

Accessing GPIO From User Space

GPIO Sysfs Interface for Userspace

Biyong SUN
23, DEC 2015



EXTERNAL USE



SECURE CONNECTIONS
FOR A SMARTER WORLD

GPIO Sysfs Interface for Userspace

The details in the following documents:

[rel_imx_3.14.38_6ul_ga/kernel_imx/Documentation/gpio/sysfs.txt](#)

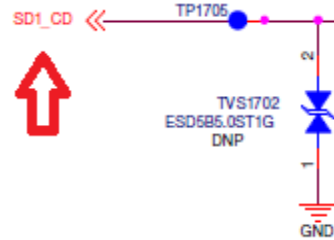
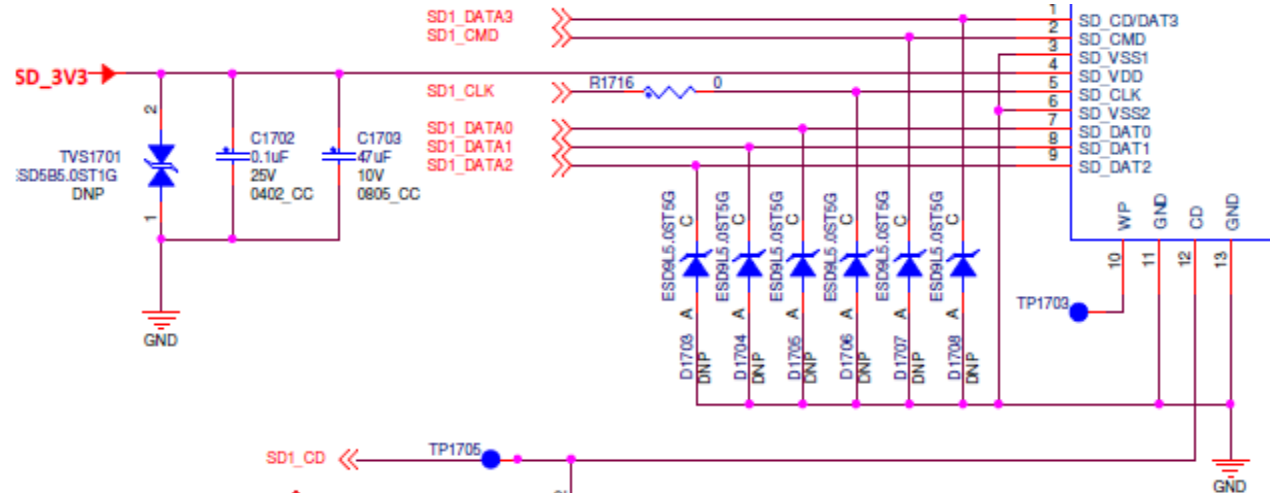
[rel_imx_3.14.28_1.1.0_ga/kernel_imx/Documentation/gpio/sysfs.txt](#)

[rel_imx_3.10.53_1.1.0_ga/ernel_imx/Documentation/gpio.txt](#)

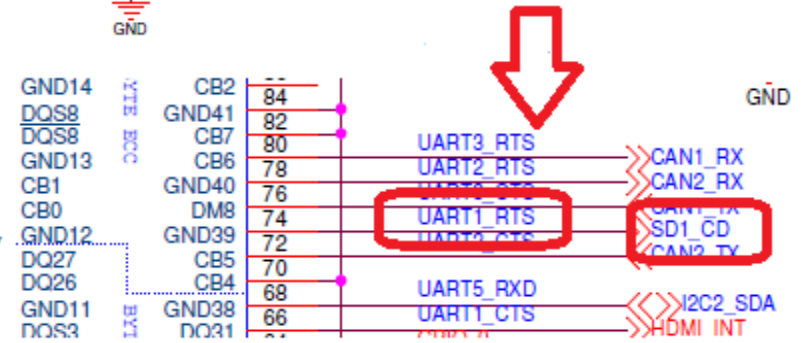
i.MX6UL evk board example

cd pin on SD1(usdhc1) as example (GPIO1, bit 19) for this test.

```
&usdhc1 {
    pinctrl-names = "default", "sta
    pinctrl-0 = <&pinctrl_usdhc1>;
    pinctrl-1 = <&pinctrl_usdhc1_10
    pinctrl-2 = <&pinctrl_usdhc1_20
    cd-gpios = <&gpio1 19 0>;
    keep-power-in-suspend,
    enable-sdio-wakeup;
    vmmc-supply = <&reg_sd1_vmmc>;
    status = "okay";
};
```



```
&iomuxc {
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_hog_1>;
    imx6ul-evk {
        pinctrl_hog_1: hoggrp-1 {
            fsl,pins = <
                MX6UL_PAD_LCD_RESET_WDOG1_WDOG_ANY 0x30b0
                MX6UL_PAD_UART1_RTS_B_GPIO1_IO19 0x17059 /* SD1 CD */
                MX6UL_PAD_GPIO1_IO05_USDHC1_VSELECT 0x17059 /* SD1 VSELECT */
                MX6UL_PAD_GPIO1_IO09_GPIO1_IO09 0x17059 /* SD1 RESET */
            >;
        };
    };
};
```



i.MX6UL evk board example(Cont.)

Currently the CD pin(GPIO1, bit 19) belongs to the usdhc1.
Cannot export to system. It will get busy error.

```
root@imx6ulevk:~# echo 19 > /sys/class/gpio/export  
-sh: echo: write error: Device or resource busy
```

But still can check the GPIO status by debugfs

i.MX6UL evk board example(Cont.)

Check the CD status by Command: `cat /sys/kernel/debug/gpio` in debugfs.

```
1 SDCRAD plug out:
2
3 root@imx6ulevk:~# cat /sys/kernel/debug/gpio
4 GPIOs 0-31, platform/209c000.gpio, 209c000.gpio:
5 gpio-3 (tsc_X- ) in hi
6 gpio-9 (VSD_3V3 ) out lo
7 gpio-19 (2190000.usdhc cd ) in hi
8
9 GPIOs 32-63, platform/20a0000.gpio, 20a0000.gpio:
10
11 GPIOs 64-95, platform/20a4000.gpio, 20a4000.gpio:
12
13 GPIOs 96-127, platform/20a8000.gpio, 20a8000.gpio:
14
```

```
1 SDCRAD plug in:
2
3 root@imx6ulevk:~# cat /sys/kernel/debug/gpio
4 GPIOs 0-31, platform/209c000.gpio, 209c000.gpio:
5 gpio-3 (tsc_X- ) in hi
6 gpio-9 (VSD_3V3 ) out lo
7 gpio-19 (2190000.usdhc cd ) in lo
8
9 GPIOs 32-63, platform/20a0000.gpio, 20a0000.gpio:
10
11 GPIOs 64-95, platform/20a4000.gpio, 20a4000.gpio:
12
13 GPIOs 96-127, platform/20a8000.gpio, 20a8000.gpio:
14
```



i.MX6UL evk board example(Cont.)

Remove the GPIO1, bit 19 from usdhc1 by modifying the device tree.
Comment the line in **RED**. Then re-compile to generate a new dtb file.

```
314 &usdhc1 {
315     pinctrl-names = "default", "state_100mhz", "state_200mhz";
316     pinctrl-0 = <&pinctrl_usdhc1>;
317     pinctrl-1 = <&pinctrl_usdhc1_100mhz>;
318     pinctrl-2 = <&pinctrl_usdhc1_200mhz>;
319 //     cd-gpios = <&gpio1 19 0>;
320     keep-power-in-suspend;
321     enable-sdio-wakeup;
322     vmmc-supply = <&reg_sd1_vmmc>;
323     status = "okay";
324 };
```

i.MX6UL evk board example(Cont.)

Use the new dtb to boot the board and check the GPIO should **NOT** be shown in the debugfs.

```
root@imx6ulevk:~# cat /sys/kernel/debug/gpio
GPIOs 0-31, platform/209c000.gpio, 209c000.gpio:
gpio-3  (tsc_X-          ) in hi
gpio-9  (VSD_3V3         ) out lo
```

```
GPIOs 32-63, platform/20a0000.gpio, 20a0000.gpio:
```

```
1
2 root@imx6ulevk:~# cat /sys/kernel/debug/gpio
3 GPIOs 0-31, platform/209c000.gpio, 209c000.gpio:
4 gpio-3  (tsc_X-          ) in hi
5 gpio-9  (VSD_3V3         ) out lo
6 gpio-19 (2190000.usdhc cd    ) in lo
7
8 GPIOs 32-63, platform/20a0000.gpio, 20a0000.gpio:
9
```

i.MX6UL evk board example(Cont.)

export i.MX6UL GPIO1 bit 19 to system:

```
root@imx6ulevk:~# ls /sys/class/gpio/  
export gpiochip0 gpiochip248 gpiochip64 unexport  
gpio132 gpiochip128 gpiochip32 gpiochip96
```

```
echo 19 > /sys/class/gpio/export
```

```
root@imx6ulevk:~# ls /sys/class/gpio/  
export gpio19 gpiochip128 gpiochip32 gpiochip96  
gpio132 gpiochip0 gpiochip248 gpiochip64 unexport
```


i.MX6UL evk board example(Cont.)

Check status:

```
1 SDCARD plug out:
2
3 root@imx6ulevk:~# cat /sys/class/gpio/gpio19/value
4 524288
5
6 root@imx6ulevk:~# /unit_tests/memtool GPIO1.DR.DR
7 SOC: i.MX6UL
8 GPIO1 Addr:0x209c000
9 GPIO1.DR Addr:0x0209C000 Value:0xF00C0318 - The 32-bit GPIO_DR register stores data
10 GPIO1.DR.DR(0..31) :0xf00c0318
11 Data bits.
12
13 root@imx6ulevk:~# cat /sys/kernel/debug/gpio
14 GPIOs 0-31, platform/209c000.gpio, 209c000.gpio:
15 gpio-3 (tsc_X- ) in hi
16 gpio-9 (VSD_3V3 ) out lo
17 gpio-19 (sysfs ) in hi
18
19 GPIOs 32-63, platform/20a0000.gpio, 20a0000.gpio:
20 ..
```

```
1 SDCARD plug in:
2
3 root@imx6ulevk:~# cat /sys/class/gpio/gpio19/value
4 0
5
6 root@imx6ulevk:~# /unit_tests/memtool GPIO1.DR.DR
7 SOC: i.MX6UL
8 GPIO1 Addr:0x209c000
9 GPIO1.DR Addr:0x0209C000 Value:0xF0040318 - The 32-bit
10 GPIO1.DR.DR(0..31) :0xf0040318
11 Data bits.
12
13 root@imx6ulevk:~# cat /sys/kernel/debug/gpio
14 GPIOs 0-31, platform/209c000.gpio, 209c000.gpio:
15 gpio-3 (tsc_X- ) in hi
16 gpio-9 (VSD_3V3 ) out lo
17 gpio-19 (sysfs ) in lo
18
19 GPIOs 32-63, platform/20a0000.gpio, 20a0000.gpio:
20 ..
```



i.MX6UL evk board example(Cont.)

GPIO output:

```
root@imx6ulevk:~# echo 19 > /sys/class/gpio/export
root@imx6ulevk:~# echo out > /sys/class/gpio/gpio19/direction
root@imx6ulevk:~# cat /sys/kernel/debug/gpio
GPIOs 0-31, platform/209c000.gpio, 209c000.gpio:
gpio-3  (tsc_X-          ) in hi
gpio-9  (VSD_3V3         ) out lo
gpio-19 (sysfs          ) out lo
```

GPIOs 32-63, platform/20a0000.gpio, 20a0000.gpio:

i.MX6UL evk board example(Cont.)

GPIO output:

```
root@imx6ulevk:~# /unit_tests/memtool GPIO1.DR.DR
```

```
SOC: i.MX6UL
```

```
GPIO1 Addr:0x209c000
```

```
GPIO1.DR Addr:0x0209C000 Value:0xF0040118 - The 32-bit GPIO_DR register stores data that is ready to be driven to the output lines.
```

```
GPIO1.DR.DR(0..31)      :0xf0040118
```

```
Data bits.
```

```
root@imx6ulevk:~# /unit_tests/memtool GPIO1.DR.DR
```

```
SOC: i.MX6UL
```

```
GPIO1 Addr:0x209c000
```

```
GPIO1.DR Addr:0x0209C000 Value:0xF00C0118 - The 32-bit GPIO_DR register stores data that is ready to be driven to the output lines.
```

```
GPIO1.DR.DR(0..31)      :0xf00c0118
```

```
Data bits.
```

i.MX6UL evk board example(Cont.)

Because it is output, the debug fs can **NOT** see the changes.

Enable **SION** to loopback to system.

27.4.3.2 GPIO Write Mode

The programming sequence for driving output signals should be as follows:

1. Configure IOMUX to select GPIO mode (Via IOMUXC), also enable SION if need to read loopback pad value through PSR
2. Configure GPIO direction register to output (GPIO_GDIR[GDIR] set to 1b).
3. Write value to data register (GPIO_DR).

A pseudocode description to drive 4'b0101 on [output3:output0] is as follows:

```
// SET PADS TO GPIO MODE VIA IOMUX.
write sw_mux_ctl_pad_<output [0-3]>.mux_mode, <GPIO_MUX_MODE>
// Enable loopback so we can capture pad value into PSR in output mode
write sw_mux_ctl_pad_<output [0-3]>.sion, 1
// SET GDIR=1 TO OUTPUT BITS.
write GDIR[31:4,output3_bit,output2_bit, output1_bit, output0_bit,] 32'hxxxxxxxxF
// WRITE OUTPUT VALUE=4'b0101 TO DR.
write DR, 32'hxxxxxxxx5
// READ OUTPUT VALUE FROM PSR ONLY.
read_cmp PSR, 32'hxxxxxxxx5
```



i.MX6UL evk board example(Cont.)

```
root@imx6ulevk:~# /unit_tests/memtool IOMUXC.SW_MUX_CTL_PAD_UART1_RTS_B.SION=1
SOC: i.MX6UL
write 0x00000001 to Bit 4..4 of 0x020E0090
```

```
root@imx6ulevk:~# cat /sys/kernel/debug/gpio
GPIOs 0-31, platform/209c000.gpio, 209c000.gpio:
gpio-3 (tsc_X-          ) in hi
gpio-9 (VSD_3V3         ) out lo
gpio-19 (sysfs           ) out hi
```

GPIOs 32-63, platform/20a0000.gpio, 20a0000.gpio:

```
root@imx6ulevk:~# echo 0 > /sys/class/gpio/gpio19/value
root@imx6ulevk:~# cat /sys/kernel/debug/gpio
GPIOs 0-31, platform/209c000.gpio, 209c000.gpio:
gpio-3 (tsc_X-          ) in hi
gpio-9 (VSD_3V3         ) out lo
gpio-19 (sysfs           ) out lo
```



i.MX6UL evk board example(Cont.)

How to calculate the GPIO number?

Sysfs.txt:

`/sys/class/gpio/gpiochipN/`

"base" ... same as N, the first GPIO managed by this chip

"label" ... provided for diagnostics (not always unique)

"ngpio" ... how many GPIOs this manages (N to N + ngpio - 1)

i.MX6ul evk:

```
root@imx6ulevk:~# cat /sys/class/gpio/gpiochip0/base  
0
```

```
root@imx6ulevk:~# cat /sys/class/gpio/gpiochip0/label  
209c000.gpio
```

```
root@imx6ulevk:~# cat /sys/class/gpio/gpiochip0/ngpio  
32
```

The range of i.MX6UL GPIO1 for linux system is N to N + gpio -1; 0 to 0 + 32 -1; 0 to 31

The bit 19 for of i.MX6UL GPIO1 is 0 + 19 = 19 for linux system

The command is: `echo 19 > /sys/class/gpio/export`

Note: From the 209c000, we can know it is i.MX6UL GPIO1



i.MX6UL evk board example(Cont.)

i.MX6UL GPIO3 example:

```
root@imx6ulevk:~# cat /sys/class/gpio/gpiochip64/base  
64
```

```
root@imx6ulevk:~# cat /sys/class/gpio/gpiochip64/label  
20a4000.gpio
```

```
root@imx6ulevk:~# cat /sys/class/gpio/gpiochip0/ngpio  
32
```

i.MX6UL GPIO3 bit 5 to export to linux system:

$5 + 64 = 69$

The command line:

```
echo 69 > /sys/class/gpio/export
```



SECURE CONNECTIONS
FOR A SMARTER WORLD