



目录

UART.....	2
EMMC/SDCARD.....	3
I2C 总线.....	4
LCD/LVDS/HDMI.....	6
声卡.....	8
RTC.....	9
SPI.....	10
CAN 总线.....	11
网卡.....	12
按键.....	14
红外.....	15
LED/GPIO.....	16

EmbedSky

天嵌科技



UART

源码路径

主要修改文件为设备树文件，驱动一般不需要修改，驱动代码位于：

文件内容	源码路径
设备树文件	arch/arm/boot/dts/imx6qdl-sabresd.dtsi [coreC 板卡]
	arch/arm/boot/dts/e9v3qdl-sabresd.dtsi [e9v3 板卡]
	arch/arm/boot/dts/e9v2qdl-sabresd.dtsi [e9v2 板卡]
驱动源码	drivers/tty/serial/imx.c drivers/tty/serial/mxs-auart.c drivers/tty/serial/serial_core.c

上面只列出主要的驱动文件且与 imx 平台相关的，UART1~UART5 对应系统中的设备节点为/dev/ttySAC0~/dev/ttySAC4。

代码简述

1. 调试串口

板子使用串口 1 作为调试串口，dtsi 将串口 1 设置为系统标准输出。

```
chosen {  
    stdout-path = &uart1;  
};
```

配置一般属性并将串口 1 的状态设置为”okay”打开模式：

```
&uart1 {  
    pinctrl-names = "default";  
    pinctrl-0 = <&pinctrl_uart1>;  
    status = "okay";  
};
```

定义配置管脚模式：

```
pinctrl_uart1: uart1grp {  
    fsl,pins = <  
        MX6QDL_PAD_SD3_DAT7__UART1_TX_DATA 0x1b0b1  
        MX6QDL_PAD_SD3_DAT6__UART1_RX_DATA 0x1b0b1  
        //MX6QDL_PAD_EIM_D20__UART1_RTS_B      0x1b0b1  
        //MX6QDL_PAD_EIM_D19__UART1_CTS_B      0x1b0b1  
    >;  
};
```

注：

1. 名字需要与 pinctrl-0 属性一致，管脚配置需要查找原理图及 imx6q-pinfunc.h(下同)。
2. 如果有其它位置定义了同一个管脚，需要将其它位置的定义注释掉(下同)。



2. 串口 2-5

串口 2-5 配置都为一样，以下以串口 2 为讲解。

配置串口一般属性并使能串口 2：

```
&uart2 {  
    pinctrl-names = "default";  
    pinctrl-0 = <&pinctrl_uart2>;  
    status = "okay";  
};
```

定义配置 uart2 管脚模式：

```
pinctrl_uart2: uart2grp {  
    fsl,pins = <  
        MX6QDL_PAD_EIM_D26__UART2_TX_DATA    0x1b0b1  
        MX6QDL_PAD_EIM_D27__UART2_RX_DATA    0x1b0b1  
        MX6QDL_PAD_EIM_D28__UART2_CTS_B 0x1b0b1  
        MX6QDL_PAD_EIM_D29__UART2_RTS_B 0x1b0b1  
    >;  
};
```

EMMC/SDCARD

源码路径

驱动一般不修改，驱动代码位于：

文件内容	源码路径
设备树文件	arch/arm/boot/dts/imx6qdl-sabresd.dtsi [coreC 板卡]
	arch/arm/boot/dts/e9v3qdl-sabresd.dtsi [e9v3 板卡]
	arch/arm/boot/dts/e9v2qdl-sabresd.dtsi [e9v2 板卡]
驱动源码	drivers/mmc/host/sdhci-esdhc-imx.c drivers/mmc/host/sdhci.c drivers/mmc/host/sdhci-pltfm.c

上面只列出主要的驱动文件且与 imx 平台相关的，emmc/sdcard 对应系统中的设备节点为/dev/mmcblk*。

代码简述

修改 dtsi 文件配置 emmc 属性并使能。

```
&usdhc2 {  
    pinctrl-names = "default";  
    pinctrl-0 = <&pinctrl_usdhc2>;  
    bus-width = <4>;
```



```
cd-gpios = <&gpio1 4 GPIO_ACTIVE_LOW>;
wp-gpios = <&gpio1 2 GPIO_ACTIVE_HIGH>;
no-1-8-v;
keep-power-in-suspend;
enable-sdio-wakeup;
status = "okay";
```

};

管脚配置

```
pinctrl_usdhc2: usdhc2grp {
    fsl,pins = <
        MX6QDL_PAD_SD2_CMD__SD2_CMD      0x17059
        MX6QDL_PAD_SD2_CLK__SD2_CLK      0x10059
        MX6QDL_PAD_SD2_DAT0__SD2_DATA0    0x17059
        MX6QDL_PAD_SD2_DAT1__SD2_DATA1    0x17059
        MX6QDL_PAD_SD2_DAT2__SD2_DATA2    0x17059
        MX6QDL_PAD_SD2_DAT3__SD2_DATA3    0x17059
    >;
};
```

I2C 总线

开发板中触摸屏默认挂载到 i2c1 接口，声卡默认挂载到 i2c2，rtc 默认挂载到 i2c3。

源码路径

主要修改文件为设备树文件，驱动一般不需要修改，驱动代码位于：

文件内容	源码路径
设备树文件	arch/arm/boot/dts/imx6qdl-sabresd.dtsi [coreC 板卡]
	arch/arm/boot/dts/e9v3qdl-sabresd.dtsi [e9v3 板卡]
	arch/arm/boot/dts/e9v2qdl-sabresd.dtsi [e9v2 板卡]
I2c 驱动源码	drivers/i2c/busses/i2c-imx.c
电容触摸驱动	drivers/input/touchscreen/edt-ft5x06.module
	drivers/input/touchscreen/goodix.module (默认不开源)
电阻触摸驱动	drivers/input/touchscreen/tsc2007.module
Sgtl9000 声卡驱动	sound/soc/codecs/*
RTC 驱动	drivers/rtc/rtc-pcf8563.c

上面只列出主要的驱动文件且与 imx 平台相关的，i2c1~i2c3 对应系统中的设备节点为/dev/i2c-0~/dev/i2c-2。



代码简述

开发板中触摸屏默认挂载到 i2c1 接口，包括电阻触摸芯片 ts2007、电容触摸芯片 ft5x06、gt911（系统识别芯片会自动匹配触摸型号）。

修改 dtsi 文件配置 i2c1 属性并使能。

```
&i2c1 {
    clock-frequency = <100000>;
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_i2c1>;
    status = "okay";

    ft5x06@38 {
        compatible = "edt,edt-ft5306";    //匹配电容触摸 ft5x06 驱动
        reg = <0x38>;                    //i2c 设备地址
        interrupt-parent = <&gpio1>;
        interrupts = <9 2>;
        int-gpios = <&gpio1 9 GPIO_ACTIVE_LOW>;    //中断脚配置
        reset-gpios = <&gpio1 5 GPIO_ACTIVE_HIGH>;    //复位脚配置
    };

    tsc2007@48 {
        compatible = "ti,tsc2007";        //匹配电阻触摸 ts2007 驱动
        reg = <0x48>;                    //设备地址
        pinctrl-names = "default";
        pinctrl-0 = <&pinctrl_i2c1_tsc2007_int>;    //管脚配置
        interrupt-parent = <&gpio1>;
        interrupts = <29 8>;
        gpios = <&gpio1 29 GPIO_ACTIVE_LOW>;
        ti,x-plate-ohms = <660>;
    };

    gt9xx@5d {
        compatible = "goodix,gt911";
        reg = <0x5d>;
        interrupt-parent = <&gpio1>;
        interrupts = <9 2>;
        gpios = <&gpio1 9 GPIO_ACTIVE_LOW>;
        wakeup-gpios = <&gpio1 5 GPIO_ACTIVE_HIGH>;
    };
};

I2C 管脚定义，触摸部分的引脚定义不在此列出。
pinctrl_i2c1: i2c1grp {
```



```
fsl,pins = <
                MX6QDL_PAD_CSI0_DAT8_I2C1_SDA        0x4001b8b1
                MX6QDL_PAD_CSI0_DAT9_I2C1_SCL        0x4001b8b1
                >;
};
```

LCD/LVDS/HDMI

源码路径

主要修改文件为设备树文件，驱动一般不需要修改，驱动代码位于：

文件内容	源码路径
设备树文件	arch/arm/boot/dts/imx6qdl-sabresd.dtsi [coreC 板卡]
	arch/arm/boot/dts/e9v3qdl-sabresd.dtsi [e9v3 板卡]
	arch/arm/boot/dts/e9v2qdl-sabresd.dtsi [e9v2 板卡]
驱动源码	drivers/video/fbdev/mxc/*

上面只列出主要的驱动文件且与 imx 平台相关的，显示对应系统中的设备节点为/dev/fb*。

代码简述

IMX6Q 系列平台默认为 800*480 LCD 七寸普清屏，适配天嵌大部分显示屏(不同版本系统可能不一致)，更换显示分辨率一般在 uboot 菜单命令行选择即可。如需添加天嵌以外显示屏，可使用 TQSDMaker 制卡软件制卡时进行添加，具体查看《TQSDMaker 用户使用手册 1_20190727》。

dtb 文件修改，使能。

mxcfb1 为默认显示屏，disp_dev 为默认 LCD 显示，mode_str 为默认显示屏型号，如果要修改默认显示屏，具体修改 mxcfb1 即可。

```
mxcfb1: fb@0 {
    compatible = "fsl,mxc_sdc_fb"; //匹配 LCD 驱动
    disp_dev = "lcd"; //默认为 LCD 显示
    interface_pix_fmt = "RGB24"; //LCD 显示位数为 24
    mode_str = "CLAA-WVGA"; //默认为七寸普清屏 800*480
    default_bpp = <24>;
    int_clk = <0>;
    late_init = <0>;
    status = "okay";
};
```

如果要将默认显示切换为 lvds，需要修改 uboot 源码 include/configs/mx6sabre_common.h

将

```
mxcfb0=video=mxcfb0:dev=lcd,CLAA-WVGA
```



改为

mxcfb0=video=mxcfb0:dev=ldb,1360x768@60,if=RGB666,bpp=32

同时修改 disp_dev 为 ldb 及对应参数

&ldb {

```
status = "okay";
dual-mode = <1>;           //双路 lvds，单路为 split
```

```
lvds-channel@0 {
    fsl,data-mapping = "spwg"; //数据格式
    fsl,data-width = <24>;     //数据位宽，即 8bit RGB
    status = "okay";
```

```
    display-timings {
        native-mode = <&timing0>;
        timing0: hsd100pxn1 {
            clock-frequency = <65000000>; //时钟频率
            hactive = <1280>;             //行宽----每行像素点个数
            vactive = <800>;              //屏幕高度-----屏幕的行数
            hback-porch = <20>;          //水平后沿
            hfront-porch = <20>;         //水平前沿
            vback-porch = <4>;           //垂直后沿
            vfront-porch = <4>;          //垂直前沿
            hsync-len = <10>;            //水平同步
            vsync-len = <4>;             //垂直同步
        };
    };
};
```

```
lvds-channel@1 {
    fsl,data-mapping = "spwg";
    fsl,data-width = <18>;
    primary;
    status = "okay";

    display-timings {
        native-mode = <&timing1>;
        timing1: hsd100pxn1 {
            clock-frequency = <65000000>;
            hactive = <1024>;
            vactive = <768>;
            hback-porch = <220>;
            hfront-porch = <40>;
            vback-porch = <21>;
```



```

vfront-porch = <7>;
hsync-len = <60>;
vsync-len = <10>;

};

};

};

};

```

声卡

声卡默认挂载到 i2c2，声卡芯片为 sgtl5000。声卡通过 i2s 传输数据，i2c 配置寄存器。

源码路径

主要修改文件为设备树文件，驱动一般不需要修改，驱动代码位于：

文件内容	源码路径
设备树文件	arch/arm/boot/dts/imx6qdl-sabresd.dtsi [coreC 板卡]
	arch/arm/boot/dts/e9v3qdl-sabresd.dtsi [e9v3 板卡]
	arch/arm/boot/dts/e9v2qdl-sabresd.dtsi [e9v2 板卡]
驱动源码	sound/soc/codecs/*

上面只列出主要的驱动文件且与 imx 平台相关的，sgtl5000 对应系统中的设备节点为/dev/snd/*。

代码简述

```

&i2c2 {
    clock-frequency = <100000>;
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_i2c2>;
    status = "okay";

    codec: sgtl5000@0a {
        compatible = "fsl,sgtl5000";           //声卡驱动匹配
        reg = <0x0a>;                          //i2c 设备地址
        clocks = <&clks 201>;
        micbias-resistor-k-ohms = <1>;
        micbias-voltage-m-volts = <3000>;
        VDDA-supply = <&reg_3p3v>;
        VDDIO-supply = <&reg_3p3v>;
    };
    .....

```



RTC

开发板中没有使用芯片内部原生的 rtc，而是使用了 pcf8563 RTC 芯片。通过 i2c 总线与系统相连。

源码路径

主要修改文件为设备树文件，驱动一般不需要修改，驱动代码位于：

文件内容	源码路径
设备树文件	arch/arm/boot/dts/imx6qdl-sabresd.dtsi [coreC 板卡]
	arch/arm/boot/dts/e9v3qdl-sabresd.dtsi [e9v3 板卡]
	arch/arm/boot/dts/e9v2qdl-sabresd.dtsi [e9v2 板卡]
驱动源码	drivers/rtc/rtc-pcf8563.c

上面只列出主要的驱动文件且与 imx 平台相关的，rtc 对应的系统设备节点为/dev/rtc0。

代码简述

Rtc 芯片默认挂载到 i2c3，使能即可。

```
&i2c3 {
    clock-frequency = <100000>;    //i2c 时钟频率
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_i2c3>;
    status = "okay";

    pcf8563@51 {                    //rtc 设备
        compatible = "nxp,pcf8563"; //匹配 pcf8563 驱动
        reg = <0x51>;               //i2c 设备地址
        interrupt-parent = <&gpio6>;
        interrupts = <31 2>;
    };
};
```

管脚配置：

```
pinctrl_i2c3: i2c3grp {
    fsl,pins = <
        MX6QDL_PAD_EIM_D17__I2C3_SCL    0x4001b8b1
        MX6QDL_PAD_EIM_D18__I2C3_SDA    0x4001b8b1
    >;
};
```



SPI

在开发板中只将 spi 接口引出，驱动为总线添加了一下 spidev 的虚拟设备，以便应用直接访问 spi 总线。

源码路径

主要修改文件为设备树文件，驱动一般不需要修改，驱动代码位于：

文件内容	源码路径
设备树文件	arch/arm/boot/dts/imx6qdl-sabresd.dtsi [coreC 板卡]
	arch/arm/boot/dts/e9v3qdl-sabresd.dtsi [e9v3 板卡]
	arch/arm/boot/dts/e9v2qdl-sabresd.dtsi [e9v2 板卡]
spi 总线驱动源码	drivers/spi/spi-imx.c
虚拟设备	drivers/spi/spidev.c

上面只列出主要的驱动文件且与 imx 平台相关的，spi1、spi2 对应的系统设备节点为/dev/spidev0.0、/dev/spidev1.0。

代码简述

dtsi 使能

```
&ecspi1 {
    fsl,spi-num-chipselects = <1>;
    cs-gpios = <&gpio4 9 0>;           //片选引脚
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_ecspi1>;
    status = "okay";                   //disabled 表示关闭，okay 为打开

    flash: m25p80@0 {                  //spi 虚拟设备
        #address-cells = <1>;
        #size-cells = <1>;
        compatible = "st,m25p32";
        spi-max-frequency = <20000000>; //spi 最大时钟频率
        reg = <0>;                     //spi 地址
    };
};

&ecspi2 {
    fsl,spi-num-chipselects = <1>;
    cs-gpios = <&gpio5 29 GPIO_ACTIVE_HIGH>;
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_ecspi2>;
    status = "okay";
```



```
spidev0: spi@0 {
    compatible = "rohm,dh2228fv";
    reg = <0>;
    spi-max-frequency = <54000000>;
};
```

spi 管脚配置:

```
pinctrl_ecspi1: ecspi1grp {
    fsl,pins = <
        MX6QDL_PAD_KEY_COL1__ECSPI1_MISO    0x100b1
        MX6QDL_PAD_KEY_ROW0__ECSPI1_MOSI    0x100b1
        MX6QDL_PAD_KEY_COL0__ECSPI1_SCLK    0x100b1
        MX6QDL_PAD_KEY_ROW1__GPIO4_IO09    0x1b0b0
    >;
};

pinctrl_ecspi2: ecspi2grp {
    fsl,pins = <
        MX6QDL_PAD_EIM_OE__ECSPI2_MISO    0x100b1
        MX6QDL_PAD_EIM_CS1__ECSPI2_MOSI    0x100b1
        MX6QDL_PAD_EIM_CS0__ECSPI2_SCLK    0x100b1
        MX6QDL_PAD_CSI0_DAT11__GPIO5_IO29    0x80000000
    >;
};
```

CAN 总线

源码路径

主要修改文件为设备树文件，驱动一般不需要修改，驱动代码位于：

文件内容	源码路径
设备树文件	arch/arm/boot/dts/imx6qdl-sabresd.dtsi [coreC 板卡]
	arch/arm/boot/dts/e9v3qdl-sabresd.dtsi [e9v3 板卡]
	arch/arm/boot/dts/e9v2qdl-sabresd.dtsi [e9v2 板卡]
驱动源码	drivers/net/can/flexcan.c

上面只列出主要的驱动文件且与 imx 平台相关的，can 对应的系统设备节点为 can0、can1。



代码简述

dtsi 使能

```
&can1 {
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_flexcan1>;
    status = "okay";
};

&can2 {
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_flexcan2>;
    status = "okay";
};
```

管脚配置

```
pinctrl_flexcan1: flexcan1grp {
    fsl,pins = <
        MX6QDL_PAD_GPIO_7__FLEXCAN1_TX 0x80000000
        MX6QDL_PAD_GPIO_8__FLEXCAN1_RX 0x80000000
    >;
};

pinctrl_flexcan2: flexcan2grp {
    fsl,pins = <
        MX6QDL_PAD_KEY_ROW4__FLEXCAN2_RX 0x1b0b0
        MX6QDL_PAD_KEY_COL4__FLEXCAN2_TX 0x1b0b0
    >;
};
```

网卡

源码路径

主要修改文件为设备树文件，驱动一般不需要修改，驱动代码位于：

文件内容	源码路径
设备树文件	arch/arm/boot/dts/imx6qdl-sabresd.dtsi [coreC 板卡]
	arch/arm/boot/dts/e9v3qdl-sabresd.dtsi [e9v3 板卡]
	arch/arm/boot/dts/e9v2qdl-sabresd.dtsi [e9v2 板卡]
驱动源码	drivers/net/phy/*

上面只列出主要的驱动文件且与 imx 平台相关的，网卡对应系统的设备节点为 eth0。



代码简述

dtsi 使能

&fec {

```
pinctrl-names = "default";
pinctrl-0 = <&pinctrl_enet>;           //管脚配置
phy-mode = "rgmii";
phy-reset-gpios = <&gpio1 25 0>;        //网卡复位脚
fsl,magic-packet;
status = "okay";
```

};

管脚配置

pinctrl_enet: enetgrp {

```
    fsl,pins = <
        MX6QDL_PAD_ENET_MDIO__ENET_MDIO      0x1b8b0
        MX6QDL_PAD_ENET_MDC__ENET_MDC         0x1b0b0
        /* AR8035 reset */
        /* AR8035 interrupt */
        /* GPIO16 -> AR8035 25MHz */
        MX6QDL_PAD_GPIO_16__ENET_REF_CLK      0xc0000000
        MX6QDL_PAD_RGMII_TXC__RGMII_TXC       0x80000000
        MX6QDL_PAD_RGMII_TD0__RGMII_TD0       0x1b030
        MX6QDL_PAD_RGMII_TD1__RGMII_TD1       0x1b030
        MX6QDL_PAD_RGMII_TD2__RGMII_TD2       0x1b030
        MX6QDL_PAD_RGMII_TD3__RGMII_TD3       0x1b030
        MX6QDL_PAD_RGMII_TX_CTL__RGMII_TX_CTL 0x1b030
        /* AR8035 CLK_25M --> ENET_REF_CLK (V22) */
        MX6QDL_PAD_ENET_REF_CLK__ENET_TX_CLK  0x0a0b1
        /* AR8035 pin strapping: IO voltage: pull up */
        MX6QDL_PAD_RGMII_RXC__RGMII_RXC       0x1b030
        /* AR8035 pin strapping: PHYADDR#0: pull down */
        MX6QDL_PAD_RGMII_RD0__RGMII_RD0       0x13030
        /* AR8035 pin strapping: PHYADDR#1: pull down */
        MX6QDL_PAD_RGMII_RD1__RGMII_RD1       0x13030
        /* AR8035 pin strapping: MODE#1: pull up */
        MX6QDL_PAD_RGMII_RD2__RGMII_RD2       0x1b030
        /* AR8035 pin strapping: MODE#3: pull up */
        MX6QDL_PAD_RGMII_RD3__RGMII_RD3       0x1b030
        /* AR8035 pin strapping: MODE#0: pull down */
        MX6QDL_PAD_RGMII_RX_CTL__RGMII_RX_CTL 0x13030
    >;
};
```



按键

TQIMX6Q 板卡默认配置有 4 个按键。

源码路径

主要修改文件为设备树文件，驱动一般不需要修改，驱动代码位于：

文件内容	源码路径
设备树文件	arch/arm/boot/dts/imx6qdl-sabresd.dtsi [coreC 板卡]
	arch/arm/boot/dts/e9v3qdl-sabresd.dtsi [e9v3 板卡]
	arch/arm/boot/dts/e9v2qdl-sabresd.dtsi [e9v2 板卡]
驱动源码	drivers/net/phy/*

上面只列出主要的驱动文件且与 imx 平台相关的，按键对应的设备节点为/dev/input/event*。

代码简述

dtsi 按键配置

```
gpio-keys {  
    compatible = "gpio-keys";  
    pinctrl-names = "default";  
    pinctrl-0 = <&pinctrl_gpio_keys>;  
  
    home {  
        label = "Home Button";  
        gpios = <&gpio1 28 GPIO_ACTIVE_LOW>;  
        gpio-key,wakeup;  
        linux,code = <KEY_HOME>;  
    };  
    enter {  
        label = "Enter Button";  
        gpios = <&gpio4 5 GPIO_ACTIVE_LOW>;  
        gpio-key,wakeup;  
        linux,code = <KEY_POWER>;  
    };  
    esc {  
        label = "Esc Button";  
        gpios = <&gpio1 30 GPIO_ACTIVE_LOW>;  
        gpio-key,wakeup;  
        linux,code = <KEY_ESC>;  
    };  
};
```



```

    back {
        label = "Back Button";
        gpios = <&gpio4 20 GPIO_ACTIVE_LOW>;
        gpio-key,wakeup;
        linux,code = <KEY_BACK>; //158
    };
};

```

管脚配置

```

pinctrl_gpio_keys: gpio_keysgroup {
    fsl,pins = <
        MX6QDL_PAD_ENET_TX_EN__GPIO1_IO28 0x1b0b0
        MX6QDL_PAD_GPIO_19__GPIO4_IO05     0x1b0b0
        MX6QDL_PAD_ENET_TXD0__GPIO1_IO30    0x1b0b0
        MX6QDL_PAD_DI0_PIN4__GPIO4_IO20     0x1b0b0
    >;
};

```

红外

源码路径

主要修改文件为设备树文件，驱动一般不需要修改，驱动代码位于：

文件内容	源码路径
设备树文件	arch/arm/boot/dts/imx6qdl-sabresd.dtsi [coreC 板卡]
	arch/arm/boot/dts/e9v3qdl-sabresd.dtsi [e9v3 板卡]
	arch/arm/boot/dts/e9v2qdl-sabresd.dtsi [e9v2 板卡]
驱动源码	drivers/input/keyboard/tq_hs0038.c

上面只列出主要的驱动文件且与 imx 平台相关的，红外对应的设备节点为/dev/input/event*。

代码简述

dtsti 使能

```

hs0038 {
    //因在驱动中固定了 gpio，因此不需要填入 gpio 号
    compatible = "hs0038";
    status = "okay";
};

```



LED/GPIO

源码路径

主要修改文件为设备树文件，驱动一般不需要修改，驱动代码位于：

文件内容	源码路径
设备树文件	arch/arm/boot/dts/imx6qdl-sabresd.dtsi [coreC 板卡]
	arch/arm/boot/dts/e9v3qdl-sabresd.dtsi [e9v3 板卡]
	arch/arm/boot/dts/e9v2qdl-sabresd.dtsi [e9v2 板卡]
驱动源码	drivers/leds/leds-gpio.c

上面只列出主要的驱动文件且与 imx 平台相关的，LED 对应的设备节点为/sys/class/leds/led*/brightness，GPIO 对应的系统设备节点为/sys/class/gpio/*。

代码简述

1. LED

LED 与 GPIO 的驱动原理一致，都是控制其高低电平。IMX6Q_coreC 板卡有 3 个 led 灯，可如下配置：

```
leds {  
    compatible = "gpio-leds";  
  
    led4 {  
        label = "led4";  
        gpios = <&gpio3 21 0>;  
        default-state = "on";  
    };  
  
    led5 {  
        label = "led5";  
        gpios = <&gpio3 22 0>;  
        default-state = "off";  
    };  
  
    led6 {  
        label = "led6";  
        gpios = <&gpio3 23 0>;  
        default-state = "on";  
    };  
};
```



LED 控制命令：

```
echo 0 > /sys/class/leds/led*/brightness //点亮 led 灯
echo 1 > /sys/class/leds/led*/brightness //点灭 led 灯
```

GPIO 也可虚拟成一个 LED 灯的方式来进行控制。

2. GPIO

如果要将一个引脚设置为普通 GPIO，在 imx6qdl-sabresd 下添加其管脚定义，同时将其他地方使用到该引脚的地方注释掉即可，下面以 EIM_D28 为例，将其配置为普通可控 GPIO：

(1) 在 imx6q-pinfunc.h 文件中搜索 EIM_D28，搜索的结果如下所示：

```
#define MX6QDL_PAD_EIM_D28__EIM_DATA28      0x0c4 0x3d8 0x000 0x0 0x0
#define MX6QDL_PAD_EIM_D28__I2C1_SDA        0x0c4 0x3d8 0x89c 0x1 0x0
#define MX6QDL_PAD_EIM_D28__ECSPI4_MOSI     0x0c4 0x3d8 0x000 0x2 0x0
#define MX6QDL_PAD_EIM_D28__IPU2_CSI1_DATA12 0x0c4 0x3d8 0x8b8 0x3 0x0
#define MX6QDL_PAD_EIM_D28__UART2_CTS_B     0x0c4 0x3d8 0x000 0x4 0x0
#define MX6QDL_PAD_EIM_D28__UART2_RTS_B     0x0c4 0x3d8 0x924 0x4 0x0
#define MX6QDL_PAD_EIM_D28__UART2_DTE_CTS_B 0x0c4 0x3d8 0x924 0x4 0x0
#define MX6QDL_PAD_EIM_D28__UART2_DTE_RTS_B 0x0c4 0x3d8 0x000 0x4 0x0
#define MX6QDL_PAD_EIM_D28__GPIO3_IO28      0x0c4 0x3d8 0x000 0x5 0x0
#define MX6QDL_PAD_EIM_D28__IPU1_EXT_TRIG   0x0c4 0x3d8 0x000 0x6 0x0
#define MX6QDL_PAD_EIM_D28__IPU1_DIO_PIN13  0x0c4 0x3d8 0x000 0x7 0x0
```

从搜索可以看出，EIM_D28 对应的 GPIO 是 GPIO3_IO28 管脚

(2) 打开 arch/arm/boot/dts/目录下的 e9v2qdl-sabresd.dtsi(e9 板卡使用)，imx6qdl-sabresd.dtsi(imx6q corec 板卡使用) 文件，搜索 EIM_D28

```
pinctrl_uart2: uart2grp {
    fsl,pins = <
        MX6QDL_PAD_EIM_D26__UART2_TX_DATA 0x1b0b1
        MX6QDL_PAD_EIM_D27__UART2_RX_DATA 0x1b0b1
        MX6QDL_PAD_EIM_D28__UART2_CTS_B 0x1b0b1
        MX6QDL_PAD_EIM_D29__UART2_RTS_B 0x1b0b1
    >
}
```

可知 EIM_D28 被用于串口 2，搜索 pinctrl_uart2 将其注释掉

```
/*
&uart2 {
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_uart2>;
    status = "okay";
}
*/
```

(3) 将 EIM_D28 配置成 gpio:

搜索 imx6qdl-sabresd，在组里面添加步骤 1 搜索到的 gpio，MX6QDL_PAD_EIM_D28__GPIO3_IO28

```
imx6qdl-sabresd {
    pinctrl_hog: hoggrp {
        fsl,pins = <
            ...
            MX6QDL_PAD_EIM_D28__GPIO3_IO28 0x80000000 /*AWD EIM_D28*/
        >
    }
}
```



```
MX6QDL_PAD_EIM_D29__GPIO3_IO29 0x80000000 /*AWD EIM_D29*/  
MX6QDL_PAD_SD3_DAT2__GPIO7_IO06 0x80000000 /*AWD SD3_DAT2*/  
MX6QDL_PAD_SD3_DAT3__GPIO7_IO07 0x80000000 /*AWD SD3_DAT3*/  
MX6QDL_PAD_SD3_CMD__GPIO7_IO02 0x80000000 /*AWD SD3_CMD*/  
MX6QDL_PAD_EIM_A25__GPIO5_IO02 0x80000000 /*AWD EIM_A25*/  
MX6QDL_PAD_SD3_CLK__GPIO7_IO03 0x80000000 /*AWD SD3_CLK*/  
MX6QDL_PAD_SD3_DAT0__GPIO7_IO04 0x80000000 /*AWD SD3_DAT0*/  
MX6QDL_PAD_SD3_DAT1__GPIO7_IO05 0x80000000 /*AWD SD3_DAT1*/  
MX6QDL_PAD_SD1_DAT1__GPIO1_IO17 0x80000000 /*AWD PWM3*/  
MX6QDL_PAD_SD1_CMD__GPIO1_IO18 0x80000000 /*AWD PWM4*/
```

```
>;
```

```
};
```

然后重新编译，将修改后的设备树烧写进去即可。

引脚编号计算：

例如 GPIOx_y：引脚编号=32*(x-1)+y

应用层测试命令：

```
echo 92 > /sys/class/gpio/export  
echo "out" > /sys/class/gpio/gpio92/direction  
echo 1 > /sys/class/gpio/gpio92/value  
echo 0 > /sys/class/gpio/gpio92/value
```