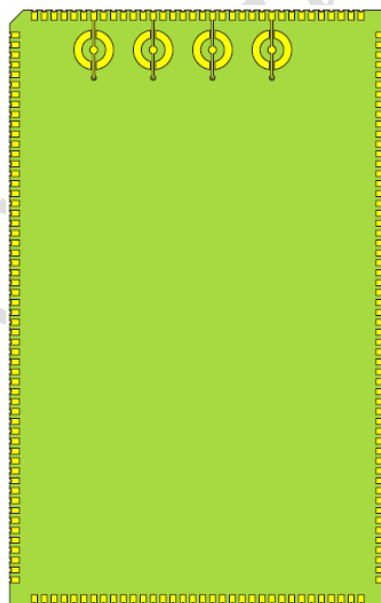


## SLM751 模块 GPIO 接口说明\_V2.5



受控文件名称	SLM751 模块 GPIO 接口说明
受控版本号	V2.5
发布机构	美格智能技术股份有限公司
发布日期	2015 年 9 月 9 日

修改记录	
V1.0	首次发布
V1.1	1: 增加[修改记录]页 2: 增加[目录]页 3: 增加[1.2.2 GPIO 作为中断的使用]小节 3: 增加[1.3 重要属性]小节 4: 修改[2.4 支持的 GPIO]内容
V2.5	修正模块图片

## 目录

1. GPIO概要 .....	4
2 支持的GPIO .....	4
3 操作/测试方法.....	4
3.1. ADB操作 .....	4
1.3 重要属性.....	5
1.3.1 中断.....	5
4. 原理图.....	6
5. 接口代码分析.....	6
5.1GPIO的一般使用流程: .....	6
5.2. 接口函数.....	7
set_gpio_export.....	7
set_gpio_unexport.....	7
set_gpio_direction.....	7
get_gpio_direction.....	7
set_gpio_val .....	8
6. 驱动代码分析.....	8
6.1. 驱动原生API .....	8
6.2.1 GPIO的一般使用流程: .....	8
6.2.2 GPIO作为中断的使用: .....	8
6.2.3 GPIO dts初始化用例: .....	9

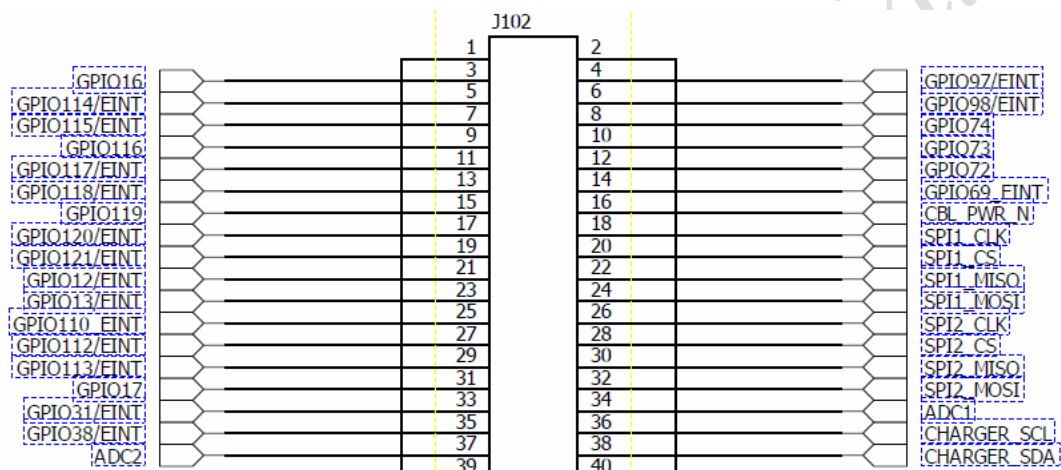
## 1. GPIO 概要

GPIO (General Purpose Input Output) 通用输入/输出, 是总线扩展器。目录/sys/class/gpio 下的每个文件夹, 包含了对应名字的 gpio 管脚的相关信息。

## 2 支持的 GPIO

支持的 gpio pin 请参照下图, 我司提供的 EVB 板背面也有丝印, 其中有/EINT 后缀说明该 gpio pin 有中断能力。如果使用我司提供的 EVB 板, 需要注意有些 gpio 已经用来连接 EVB 上的外设了, 左列只有 GPIO16、120、121、17、31、38 空余, 右列的 GPIO 都可用。

**!** GPIO97、98 会影响开机模式, 只能在开机之后使用。



## 3 操作/测试方法

### 3.1. ADB操作

751 平台 GPIO 基地址是 902, 操作 gpio 的时候需要加上基地址。因此要操作 gpio 16 的时候, 实际是操作 918, 比如 echo 918 > export

```
cd /sys/class/gpio
echo XX > export
cd gpioXX
cat direction
echo "out/in" > direction
cat value
echo "0/1" > value
cd ..
echo XX unexport
```

## 1.3 重要属性

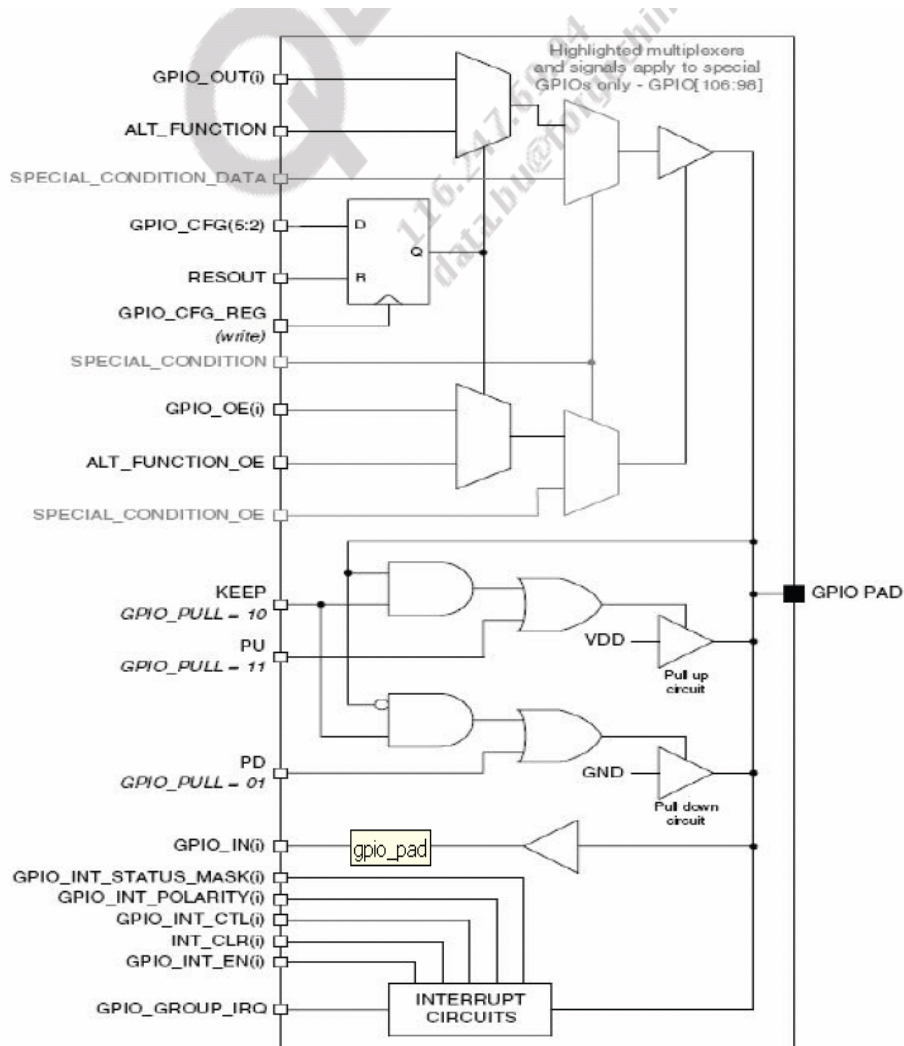
Dts 相关属性可以在 `kernel\Documentation\devicetree\bindings\gpio` 目录中搜索

### 1.3.1 中断

比如用作按键时，必须支持中断功能

MeiG Confidential

## 4. 原理图



## 5. 接口代码分析

如果不使用我司提供的源码自行开发，而是直接使用我司提供的 system.img，我司会同时提供 forgeApiTest 的源码作为客户自行定制的参考。使用 GPIO 接口可以参照 CommMethod.java 和 GpioActivity.java

### 5.1 GPIO的一般使用流程：

- 1) 确定需要操作哪个 GPIO 管脚，export
- 2) 按照需求，读写 direction 和 value
- 3) 如果不需要再操作此 gpio，那么 unexport

## 5.2. 接口函数

目前已经实现的接口有：

### set\_gpio\_export

作用等效“echo gpioPort /sys/class/gpio/export”

使用如下：

```
methodEx.loadReflectMethod(CUBIC_MANAGER_CLASS_NAME,  
                           mObjCubicManager, "set_gpio_export",  
                           new Object[] { gpioPort });
```

### set\_gpio\_unexport

作用等效“echo gpioPort /sys/class/gpio/unexport”

使用如下：

```
methodEx.loadReflectMethod(CUBIC_MANAGER_CLASS_NAME,  
                           mObjCubicManager, "set_gpio_unexport",  
                           new Object[] { gpioPort });
```

接下来的接口函数，使用到的 gpioPort 都是 export 的参数，因此如果要操作的 gpioPort 没有做过 export 操作，即/sys/class/gpio/gpioPort 目录存在，请不要做接下的 direction 和 value 的操作

### set\_gpio\_direction

作用等效“echo gpioDirection /sys/class/gpio/gpioPort/direction”

使用如下：

```
methodEx.loadReflectMethod(CUBIC_MANAGER_CLASS_NAME,  
                           mObjCubicManager, "set_gpio_direction",  
                           new Object[] { direction });
```

### get\_gpio\_direction

作用等效“cat /sys/class/gpio/gpioPort/direction”

使用如下：

```
direction = methodEx.loadReflectMethod(CUBIC_MANAGER_CLASS_NAME,  
                                       mObjCubicManager, "get_gpio_direction", null);
```

## set\_gpio\_val

作用等效“echo gpioValue /sys/class/gpio/gpioPort/value”

使用如下：

```
methodEx.loadReflectMethod(CUBIC_MANAGER_CLASS_NAME,  
                           mObjCubicManager, "set_gpio_val", new Object[] { value });
```

## 6. 驱动代码分析

### 6.1. 驱动原生API

请参照 Gpiolib.c (kernel/drivers/gpio)

比如

gpio\_export

gpio\_direction\_output

gpio\_direction\_input

gpio\_unexport

gpio\_set\_value

gpio\_get\_value

### 6.2. 驱动代码编写方法

#### 6.2.1 GPIO的一般使用流程：

- 4) 确定需要操作哪个 GPIO 管脚，export
- 5) 按照需求，读写 direction 和 value
- 6) 如果不需要再操作此 gpio，那么 unexport

#### 6.2.2 GPIO作为中断的使用：

- 1) gpio\_is\_valid
- 2) gpio\_request
- 3) gpio\_direction\_input
- 4) gpio\_to\_irq
- 5) devm\_request\_threaded\_irq、devm\_request\_irq 或者 devm\_gpio\_request\_one
- 6) enable\_irq\_wake 可选



### 6.2.3 GPIO dts初始化用例:

添加 GPIO72 的初始化一共有 2 步

#### 1、XXX pinctrl.dtsi 文件中添加 gpio 的定义:

kernel\arch\arm\boot\dts\qcom\msm8916-pinctrl.dtsi

```
&soc {
    tlmm_pinmux: pinctrl@1000000 {
        compatible = "qcom,msm-tlmm-v4";
        reg = <0x1000000 0x300000>;
        interrupts = <0 208 0>;

        /*General purpose pins*/
        gp: gp {
            qcom,pin-type-gp;
            .....
        };
        .....
        gpio_72 {
            qcom,pins = <&gp 72>;
            qcom,num-grp-pins = <1>;
            qcom,pin-func = <0>;
            label = "my_gpio";
            init_72: init_72 {
                drive-strength = <2>;
                bias-pull-up;
            };
        };
    };
};
```

具体参考 kernel\Documentation\devicetree\bindings\pinctrl\pinctrl-bindings.txt

#### 2、其他 dts 文件中添加 gpio 的使用:

kernel\arch\arm\boot\dts\qcom\msm8916-qrd-slm751-00.dts

```
my_gpio {
    pinctrl-states = "default";
    pinctrl-names = "default";
    pinctrl-0 = <&init_72>;
};
```