

内核GPIO操作库函数

gpio_request :
gpio_direction_output
gpio_direction_input
gpio_set_value
gpio_get_value
gpio_free
所需的头文件
复习结构体数组 :

内核GPIO操作库函数

GPIO作为内核资源，有唯一的软件编号。

硬件标识	软件编号
GPA0_0	S5PV210_GPA0(0)
GPC0_3	S5PV210_GPC0(3)
GPC0_4	S5PV210_GPC0(4)

gpio_request :

gpio_request(int gpio,char *name)

功能：向内核申请GPIO资源

gpio：GPIO软件编号

name：标识

gpio_direction_output

gpio_direction_output(int gpio, intvalue)

功能：设置GPIO为输出口，并且输出value值

gpio：GPIO软件编号

value：管脚状态

gpio_direction_input

gpio_direction_inputput(int gpio)

功能：设置GPIO为输入口

gpio：GPIO软件编号

gpio_set_value

gpio_set_value(int gpio, intvalue)

功能：设置GPIO的状态为value

gpio：GPIO软件编号

value：管脚状态

gpio_get_value

gpio_get_value(int gpio)

功能：获取GPIO的状态，返回值为状态信息

gpio：GPIO软件编号

gpio_free

gpio_free(int gpio);

释放GPIO资源

所需的头文件

```
#include <asm/gpio.h>
#include <plat/gpio-cfg.h>
```

OK210板上对应的LED为：

GPH2_0--GPH2_3

复习结构体数组：

```
struct led_resource {
    int gpio; //描述GPIO的软件编号
    char *name; //标签，名称
};
```

结构体数组的初始化：

```
static struct led_resource led_info[] = {
    [0] = {
        .gpio = S5PV210_GPC1(3),
        .name = "LED1"
    },
    [1] = {
        .gpio = S5PV210_GPC1(4),
        .name = "LED2"
    }
};
```

得到结构体数组的长度：

```
ARRAY_SIZE(led_info)
```

引用结构体数组的元素：

```
led_info[0].gpio  
led_info[0].name
```