

进入 `/sys/class/gpio`，可以看到以下内容：

```
1 root@okmx8mm:~# cd /sys/class/gpio
2 root@okmx8mm:/sys/class/gpio# ls
3 export  gpiochip0    gpiochip32  gpiochip96
4 gpio118  gpiochip128  gpiochip64  unexport
```

- 其中 `export` 用于导出GPIO到用户空间，例如导出编号为128 (GPIO5_I000)：

```
1 | echo 128 > export
```

当前目录下会生成 `gpio128` 目录，当系统重启时，导出的IO口会消失，所以每次使用时，最好先导出IO口。

- `unexport` 用于取消导出的GPIO，例如：

```
1 | echo 128 > unexport
```

这儿我需要使用 `GPIO5_I000` 和 `GPIO5_I001`，所以需要运行以下指令：

```
1 | echo 128 > export
2 | echo 129 > export
```

进入其中一个目录可以看到：

```
1 | root@okmx8mm:/sys/class/gpio/gpio128# ls
2 | active_low  device  direction  edge  power  subsystem  uevent  value
```

其中 `direction` 用于设置IO方向 (in, out)，`edge` 用于设置中断模式 (none、rising、falling、both)，`value` 用于读取IO电平，具体内容见

<https://www.kernel.org/doc/Documentation/gpio/sysfs.txt>。

这儿我把 `GPIO5_I000` 设置为中断输入模式，`GPIO5_I001` 设置为输出模式：

```
1 | root@okmx8mm:/sys/class/gpio/gpio128# echo "in" > direction
2 | root@okmx8mm:/sys/class/gpio/gpio128# echo "rising" > edge
3 | root@okmx8mm:/sys/class/gpio/gpio128# cd ../gpio129
4 | root@okmx8mm:/sys/class/gpio/gpio129# echo "out" > direction
```

测试时可以使用 `echo 1 > value` 设置 `GPIO5_I001` 输出高电平；`echo 0 > value` 设置 `GPIO5_I001` 输出低电平。

最后配置结果如下：

```
1 | root@okmx8mm:/sys/class/gpio/gpio129# cat /sys/kernel/debug/gpio
2 | gpiochip0: GPIOs 0-31, parent: platform/30200000.gpio, 30200000.gpio:
3 |
```

```

4 gpiochip1: GPIOs 32-63, parent: platform/30210000.gpio, 30210000.gpio:
5   gpio-38 (          |?              ) out hi
6   gpio-44 (          |cd             ) in  hi IRQ
7   gpio-51 (          |VSD_3V3        ) out lo
8
9
10  gpiochip2: GPIOs 64-95, parent: platform/30220000.gpio, 30220000.gpio:
11   gpio-80 (          |status          ) out hi
12   gpio-83 (          |usb_otg1_vbus   ) out lo
13   gpio-89 (          |WLAN_EN         ) out lo
14
15
16  gpiochip3: GPIOs 96-127, parent: platform/30230000.gpio, 30230000.gpio:
17   gpio-118 (         |headphone detect ) in  lo IRQ
18   gpio-124 (         |GPIO Key HOME   ) in  hi IRQ
19   gpio-127 (         |GPIO Key UP     ) in  hi IRQ
20
21
22  gpiochip4: GPIOs 128-159, parent: platform/30240000.gpio, 30240000.gpio:
23   gpio-128 (         |sysfs           ) in  lo IRQ
24   gpio-129 (         |sysfs           ) out lo
25   gpio-130 (         |GPIO Key DOWN   ) in  hi IRQ
26   gpio-137 (         |spi_imx         ) out hi
27   gpio-141 (         |spi_imx         ) out hi

```

3、测试

使用epoll读取 `GPIO5_I000` 中断, 并翻转 `GPIO5_I001`, 完整代码如下:

```

1  //
2  // Created by txfly on 2020/8/25.
3  //
4
5
6  #include <stdlib.h>
7  #include <unistd.h>
8  #include <fcntl.h>
9  #include <stdio.h>
10 #include <sys/epoll.h>
11
12 int main(int argc, char *argv[]) {
13     // int fd = open("/sys/class/gpio/export", O_WRONLY);
14     // if (fd == -1){
15     //     write(fd, "128", 3);
16     //     write(fd, "129", 3);
17     // }
18     // close(fd);
19     // 前面手动导出过, 这儿就不做处理了
20
21
22     // 中断输入
23     int trigger = open("/sys/class/gpio/gpio128/value", O_RDWR | O_NONBLOCK);
24     if (trigger == -1) {
25         printf("Fail to open GPIO5_I000\n");
26         exit(1);
27     }
28
29     // LED
30     int led = open("/sys/class/gpio/gpio129/value", O_WRONLY);
31     if (led == -1) {
32         printf("Fail to open GPIO5_I001\n");
33         exit(1);
34     }
35
36     // epoll
37     int epfd = epoll_create(1);
38     struct epoll_event ev, events;
39     ev.events = EPOLLPRI;
40
41

```

```
42 ev.data.fd = trigger;
43 int n = epoll_ctl(epfd, EPOLL_CTL_ADD, trigger, &ev);
44 if (n == -1) {
45     printf("Fail to add a file descriptor to the interface. \n");
46     exit(1);
47 }
48
49 char buf = 0;
50 while (1) {
51     n = epoll_wait(epfd, &events, 1, -1);
52     if (n > 0) {
53         lseek(trigger, 0, SEEK_SET);
54         n = read(trigger, &buf, 1);
55         printf("read, buf=%c\tlen=%d\n", buf, n);
56         if (buf == '0') {
57             write(led, "0", 1);
58         } else {
59             write(led, "1", 1);
60         }
61     } else {
62         break;
63     }
64 }
65 close(led);
66 close(trigger);
67 close(epfd);
68 }
```

版权声明:本文为「txfly」的原创文章, 遵循 CC 4.0 BY-SA 版权协议, 转载请附上原文出处链接及本声明。

原文链接:<https://www.jianshu.com/p/0f702330bdc8>



1人点赞>



日记本



更多精彩内容, 就在简书APP

"赠人玫瑰, 手留余香."

赞赏支持 还没有人赞赏, 支持一下



txfly

总资产109 共写了1.3W字 获得80个赞 共20个粉丝

关注



txfly

关注

总资产109

CMAKE_GNUtoMS:将GNU格式库转换为MSVC格式

阅读 97

Linux下使用mmap(/dev/mem)操作GPIO

阅读 1,243

写下你的评论...



评论0



赞1

...