

## 灯接口

1. 库文件 libjnielc.so 放到工程目录下。
2. 接口文件 jnielc.java, 注意此文件的包名不能改, 必须是 com.example.elcapi, 否则会调用不到, 所以 Android studio 工程里面, 要建好这个包名, 然后把此文件放进去。
3. 由于灯接口支持 RGB 调色功能, 所以调用红绿蓝单色的话, 需要将其他颜色关闭。

### 3.1 红

```
jnielc.seekstart();
jnielc.ledseek(seek_red, 15);
jnielc.ledseek(seek_green, 0);
jnielc.ledseek(seek_blue, 0);
jnielc.seekstop();
```

### 3.2 绿

```
jnielc.seekstart();
jnielc.ledseek(seek_red, 0);
jnielc.ledseek(seek_green, 15);
jnielc.ledseek(seek_blue, 0);
jnielc.seekstop();
```

### 3.3 蓝

```
jnielc.seekstart();
jnielc.ledseek(seek_red, 0);
jnielc.ledseek(seek_green, 0);
jnielc.ledseek(seek_blue, 15);
jnielc.seekstop();
```

## 4. 具体使用如下

### 1、控制红灯的接口

```
jnielc.ledseek(seek_red, progress);
```

其中参数 seek\_red 是控制标志, 定义为

```
private static final int seek_red=0x1;
```

参数 `progress` 是灯的亮度等级，范围是 0 到 15

例如

```
jnielc.seekstart();  
jnielc.ledseek(seek_red, progress: 10);  
jnielc.seekstop();
```

就表示亮红灯，亮度等级是 10

注意先调用

```
jnielc.seekstart();
```

然后才调用

```
jnielc.ledseek(seek_red, progress: 10);
```

最后调用

```
jnielc.seekstop();
```

## 2、控制绿灯的接口

```
jnielc.ledseek(seek_green, progress);
```

其中参数 `seek_green` 是控制标志，定义为

```
private static final int seek_green=0xa2;
```

参数 `progress` 是灯的亮度等级，范围是 0 到 15

例如

```
jnielc.seekstart();  
jnielc.ledseek(seek_green, progress: 10);  
jnielc.seekstop();
```

就表示亮绿灯，亮度等级是 10

注意先调用

```
jnielc.seekstart();
```

然后才调用

```
jnielc.ledseek(seek_green, progress: 10);
```

最后调用

```
jnielc.seekstop();
```

### 3、控制蓝灯的接口

```
jnielc.ledseek(seek_blue, progress);
```

其中参数 seek\_blue 是控制标志，定义为

```
private static final int seek_blue=0xa3;
```

参数 progress 是灯的亮度等级，范围是 0 到 15

例如

```
jnielc.seekstart();  
jnielc.ledseek(seek_blue, progress: 10);  
jnielc.seekstop();
```

就表示亮蓝灯，亮度等级是 10

注意先调用

```
jnielc.seekstart();
```

然后才调用

```
jnielc.ledseek(seek_blue, progress: 10);
```

最后调用

```
jnielc.seekstop();
```

### 继电器调用

- 1、库文件 libgpio\_control.so 放到工程目录下。
- 2、接口文件 gpioJni.java，注意此文件的包名不能改，必须是 com.sys.gpio，否则会调用不到，所以 Android studio 工程里面，要建好这个包名，然后把此文件放进去。
- 3、继电器 2（IN2）：

gpioJni.ioctl\_gpio(2, 0, 1); 设置高电平

gpioJni.ioctl\_gpio(2, 0, 0); 设置低电平

gpioJni.ioctl\_gpio(2, 1, 1); 获取电平状态返回值为 0 1

#### 4、继电器 1 (IN1) :

```
gpioJni.ioctl_gpio(3, 0, 1);  
gpioJni.ioctl_gpio(3, 0, 0);  
gpioJni.ioctl_gpio(3, 1, 1)
```

### IO 口调用

- 1、库文件 libgpio\_control.so 放到工程目录下。
- 2、接口文件 gpioJni.java, 注意此文件的包名不能改, 必须是 com.sys.gpio, 否则会调用不到, 所以 Android studio 工程里面, 要建好这个包名, 然后把此文件放进去。

#### 3、IO-1

```
gpioJni.ioctl_gpio(0, 0, 1); 设置高电平  
gpioJni.ioctl_gpio(0, 0, 0); 设置低电平  
gpioJni.ioctl_gpio(0, 1, 1); 获取电平状态返回值为 0 1
```

#### 4、IO-2

```
gpioJni.ioctl_gpio(1, 0, 1); 设置高电平  
gpioJni.ioctl_gpio(1, 0, 0); 设置低电平  
gpioJni.ioctl_gpio(1, 1, 1); 获取电平状态返回值为 0 1
```

### 温湿度

- 1、目前温湿度是按照 getevent 获取数据
- 2、温度: event5(带光感的机器是 event7)

3、湿度：event4(带光感的机器是event6)

```
C:\Users\ELC>adb shell
rk3566_t:/ $ su
rk3566_t:/ # getevent
add device 1: /dev/input/event4
    name:      "sun-ths"
add device 2: /dev/input/event5
    name:      "sun-hum"
add device 3: /dev/input/event2
    name:      "goodix-ts"
add device 4: /dev/input/event0
    name:      "fdd70030.pwm"
add device 5: /dev/input/event3
    name:      "adc-keys"
add device 6: /dev/input/event1
    name:      "rk805 pwrkey"
/dev/input/event5: 0003 001d 0000002d
/dev/input/event5: 0000 0000 00000000
^C
```

4、

5、使用 VirtualTerminal 启线程读返回值

```
VirtualTerminal mVirtualTerminal = new
```

```
VirtualTerminal("1", "getevent -l", ".");
```

```
mVirtualTerminal.setListener ();
```

```
C:\Users\ELC>adb shell
rk3566_t:/ $ su
rk3566_t:/ # getevent
add device 1: /dev/input/event4
    name:      "sun-ths"
add device 2: /dev/input/event5
    name:      "sun-hum"
add device 3: /dev/input/event2
    name:      "goodix-ts"
add device 4: /dev/input/event0
    name:      "fdd70030.pwm"
add device 5: /dev/input/event3
    name:      "adc-keys"
add device 6: /dev/input/event1
    name:      "rk805 pwrkey"
/dev/input/event5: 0003 001d 0000002d
/dev/input/event5: 0000 0000 00000000
^C
130|rk3566_t:/ #
```

6、

7、获取的值为 16 进制数值，需要自行转换一下。