

Lamp interface

1. Place the library file libjnielc.so in the project directory.
2. The interface file jnielc.java, please note that the package name of this file cannot be changed, It must be the com. example. elcapi, otherwise it will not be called. Therefore, in the Android studio project, it is necessary to create this package name and then put this file in it.
3. Due to the support of RGB color adjustment function in the light interface, if calling the red, green, and blue monochrome, other colors need to be turned off.

3.1 Red

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

3.2 Green

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

3.3 Blue

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

```
jnielc.ledseek(seek_blue, 15);  
jnielc.seekstop();
```

1、Interface for controlling red lights

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

Among them, the parameter `seek_Red` is the control flag, defined as

```
private static final int seek_red=0xa1;
```

The parameter `progress` is the brightness level of the lamp, ranging from 0 to 15

For

example

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

It means the red light is on, and the brightness level is 10

Pay attention to calling first

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

Then we call

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

Last call

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

2、Interface for controlling green light

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

Among them, the parameter `seek_Green` is the control flag, defined as

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

The parameter `progress` is the brightness level of the lamp, ranging from 0 to 15

For example

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

It means the green light is on, and the brightness level is 10
Pay attention to calling first

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

Then we call

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

Last call

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

3、Interface for controlling the blue light

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

Among them, the parameter seek_Blue is a control flag defined as

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

The parameter progress is the brightness level of the lamp, ranging from 0 to 15
For example

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

It means a bright blue light with a brightness level of 10
Pay attention to calling first

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

Then we call

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

Last call

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

Relay call

1、Library file libgpio_Control. so placed in the project directory。

2、The interface file `gpioJni.java`, please note that the package name of this file cannot be changed, It must be `com.sys.gpio`, otherwise it will not be called. Therefore, in the Android studio project, it is necessary to create this package name and then put this file in it.

3、Relay 2 (IN2):

```
gpioJni.ioctl_gpio(2, 0, 1); Set High Level
```

```
gpioJni.ioctl_gpio(2, 0, 0); Set Low Level
```

```
gpioJni.ioctl_gpio(2, 1, 1); Obtain the return value of  
level status as 0 1
```

4、Relay 1 (IN1):

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

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

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

IO port call

1、Library file `libgpio_Control`. so placed in the project directory.

2、The interface file `gpioJni.java`, please note that the package name of this file cannot be changed, It must be `com.sys.gpio`, otherwise it will not be called. Therefore,

in the Android studio project, it is necessary to create this package name and then put this file in it.

3、IO-1

```
gpioJni.ioctl_gpio(0, 0, 1); Set High Level
```

```
gpioJni.ioctl_gpio(0, 0, 0); Set Low Level
```

```
gpioJni.ioctl_gpio(0, 1, 1); Obtain the return value  
of level status as 0 1
```

4、IO-2

```
gpioJni.ioctl_gpio(1, 0, 1); Set High Level
```

```
gpioJni.ioctl_gpio(1, 0, 0); Set Low Level
```

```
gpioJni.ioctl_gpio(1, 1, 1); Obtain the return value of  
level status as 0 1
```

Temperature and humidity

1、Currently, temperature and humidity are obtained based on getevent data

2、temperature: event5(The machine with light sensitivity is event 7)

3、humidity: event4(The machine with light sensitivity is event 6)

```
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、

Using VirtualTerminal to start a thread to read the return value

```
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:/ #
```

5、

6、 The obtained value is a hexadecimal value that needs to be converted by oneself.