

OpenHarmony E53模块开发-智能手势



目 录

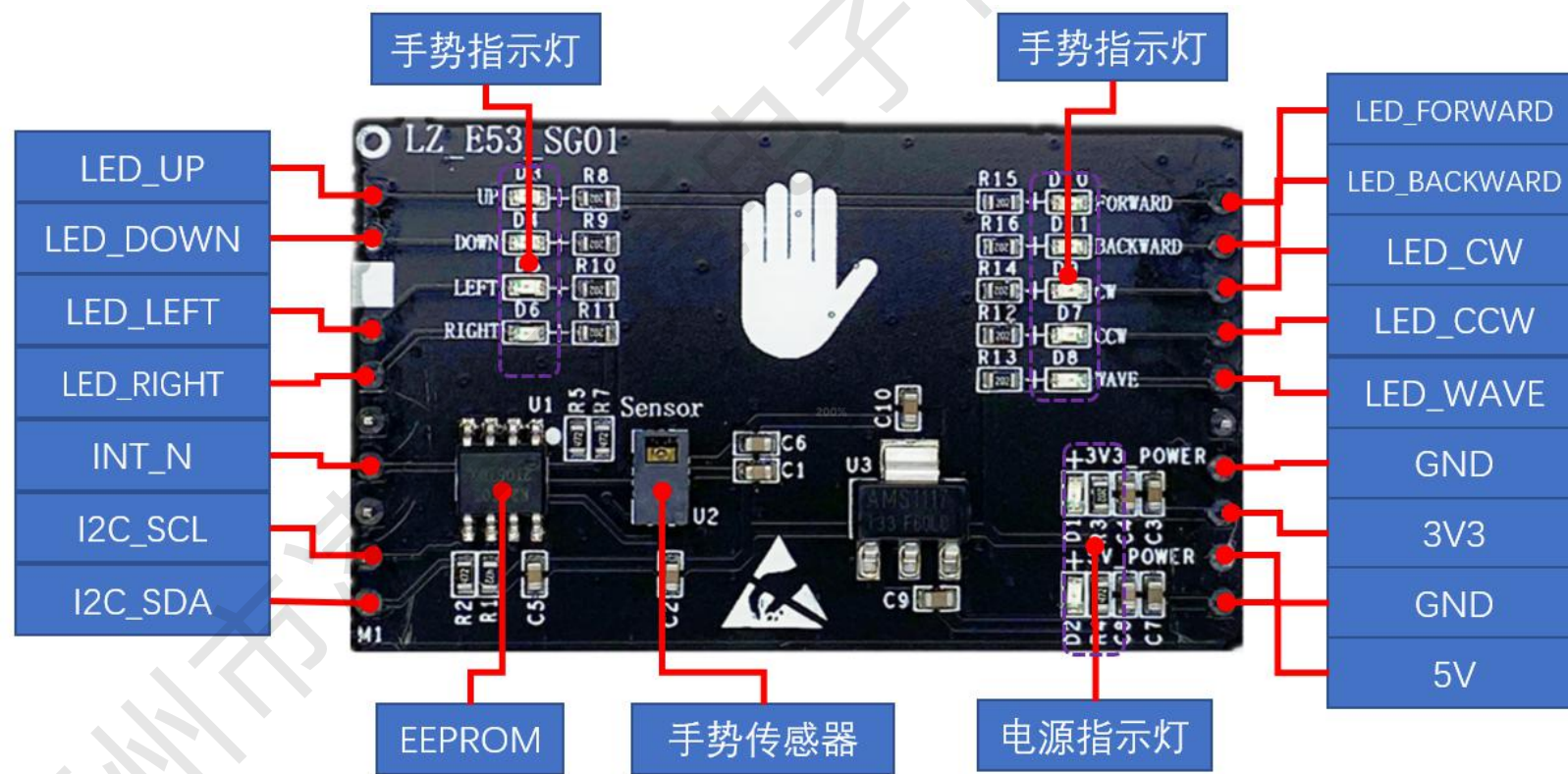
C O N T E N T S

- [01] 硬件设计
- [02] 硬件连接
- [03] API分析
- [04] 实例分析

01

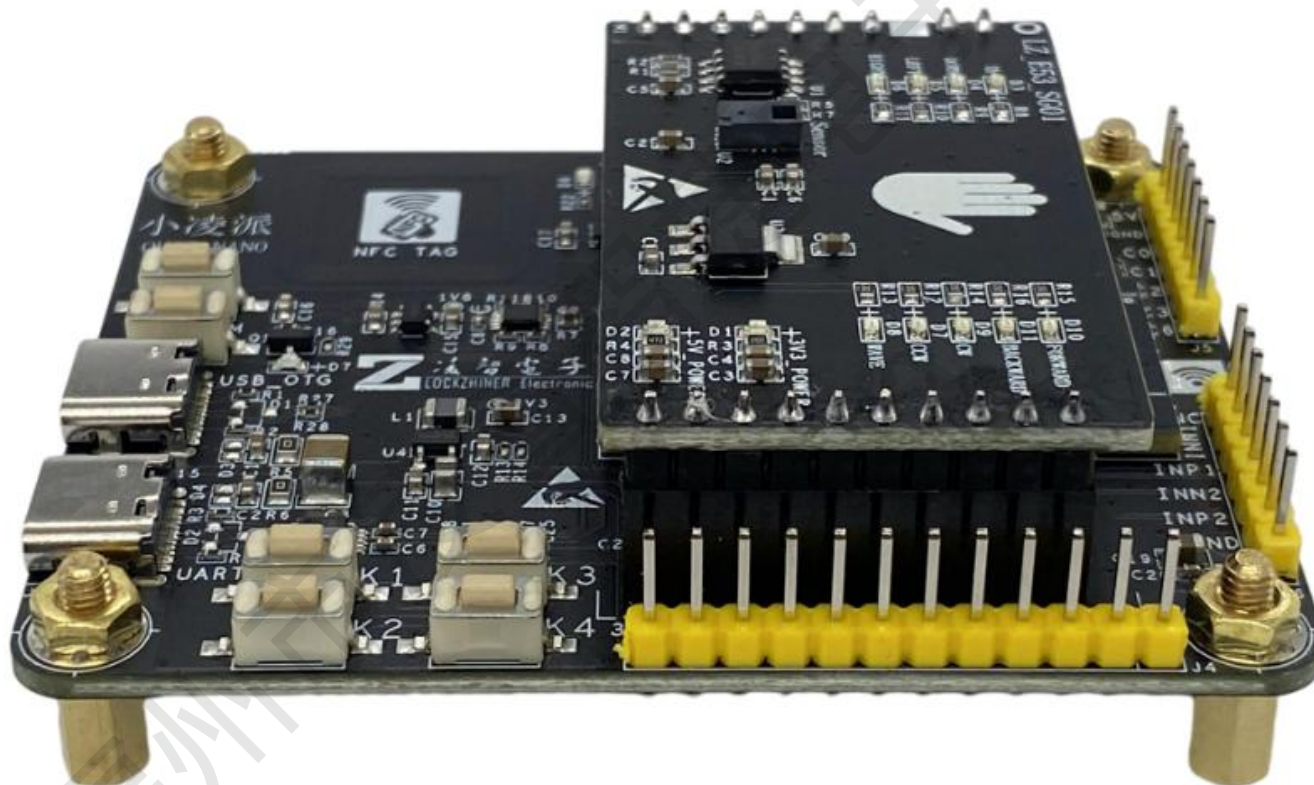
硬件设计

资源介绍



02

硬件连接



03

API分析

```
unsigned int e53_gs_init();
```

该函数主要功能是E53智能手势模块初始化，包括初始化各个LED灯的GPIO引脚，初始化i2c总线以及手势识别芯片（即PAJ7620）配置初始化。

```
unsigned int e53_gs_get_gesture_state(unsigned short *flag);
```

该函数主要功能是E53智能手势模块获取手势状态，参数flag为获取当前手势，返回值1为获取成功，0为失败。

03

API分析

```
void e53_gs_led_up_set(unsigned char is_on);  
void e53_gs_led_down_set(unsigned char is_on);  
void e53_gs_led_left_set(unsigned char is_on);  
void e53_gs_led_right_set(unsigned char is_on);  
void e53_gs_led_forward_set(unsigned char is_on);  
void e53_gs_led_backward_set(unsigned char is_on);  
void e53_gs_led_cw_set(unsigned char is_on);  
void e53_gs_led_ccw_set(unsigned char is_on);  
void e53_gs_led_wave_set(unsigned char is_on);
```

以上函数主要功能是E53智能手势模块LED灯控制，用于显示手势状态。参数is_on为控制LED灯开关，当为1开启LED，为0关闭LED。无返回值。

04

实例分析

1、打开sdk下面路径的文件

```
vendor/lockzhiner/rk2206/samples/c6_e53_gesture_sensor/e53_gesture_sensor_example.c
```

2、创建任务

在e53_gs_example函数中，创建的一个线程e53_gs_process。

```
void e53_gs_example()
{
    unsigned int thread_id;
    TSK_INIT_PARAM_S task = {0};
    unsigned int ret = LOS_OK;
    task.pfnTaskEntry = (TSK_ENTRY_FUNC)e53_gs_process;
    task.uwStackSize = 2048;
    task.pcName = "e53 getsture sensor process";
```

```
    task.usTaskPrio = 24;
    ret = LOS_TaskCreate(&thread_id, &task);
    if (ret != LOS_OK)
    {
        printf("Falied to create Task_One ret:0x%x\n", ret);
        return;
    }
}
```


e53_gs_process函数先调用e53_gs_init()初始化智能手势模块。然后调用 e53_gs_get_gesture_state(&flag);获取手势状态，获取失败时空闲100毫秒重复执行。获取手势状态成功则根据手势状态打印结果并控制对应的LED显示。

```
void e53_gs_process(void *arg)
{
    unsigned int ret = 0;
    unsigned short flag = 0;
    e53_gs_init();
    while (1)
    {
        ret = e53_gs_get_gesture_state(&flag);
        if (ret != 0)
        {
            printf("Get Gesture Statu: 0x%x\n", flag);
```

```
        if (flag & GES_UP)
        {
            printf("\tUp\n");
        }
        if (flag & GES_DOWM)
        {
            printf("\tDown\n");
        }
        if (flag & GES_LEFT)
        {
            printf("\tLeft\n");
        }
        if (flag & GES_RIGHT)
        {
            printf("\tRight\n");
        }
    }
}
```



```
if (flag & GES_FORWARD)
{
    printf("\tForward\n");
}
if (flag & GES_BACKWARD)
{
    printf("\tBackward\n");
}
if (flag & GES_CLOCKWISE)
{
    printf("\tClockwise\n");
}
if (flag & GES_COUNT_CLOCKWISE)
{
    printf("\tCount Clockwise\n");
}
if (flag & GES_WAVE)
{
    printf("\tWave\n");
}
```

```
e53_gs_led_up_set((flag & GES_UP) ? (1) : (0));
e53_gs_led_down_set((flag & GES_DOWN) ? (1) : (0));
e53_gs_led_left_set((flag & GES_LEFT) ? (1) : (0));
e53_gs_led_right_set((flag & GES_RIGHT) ? (1) : (0));
e53_gs_led_forward_set((flag & GES_FORWARD) ? (1) : (0));
e53_gs_led_backward_set((flag & GES_BACKWARD) ? (1) : (0));
e53_gs_led_cw_set((flag & GES_CLOCKWISE) ? (1) : (0));
e53_gs_led_ccw_set((flag & GES_COUNT_CLOCKWISE) ? (1) : (0));
e53_gs_led_wave_set((flag & GES_WAVE) ? (1) : (0));
}
else
{
    /* 如果没有数据，则多等待 */
    LOS_Msleep(100);
}
}
```

3、修改编译脚本

修改 `vendor/lockzhiner/rk2206/sample` 路径下 `BUILD.gn` 文件，指定 `e53_gs_example` 参与编译。

```
"/e53_gesture_sensor:e53_gs_example",
```

修改 `device/lockzhiner/rk2206/sdk_liteos` 路径下 `Makefile` 文件，添加 `-le53_gs_example` 参与编译。

```
hardware_LIBS = -lhal_iohardware -lhardware -le53_gs_example
```

4、编译固件

```
hb set -root .
```

```
hb set
```

```
hb build -f
```

5、烧写固件

6、通过串口查看结果

运行结果

Get Gesture Statu: 0x1

Down

Get Gesture Statu: 0x5

Down

Left

.....

谢谢聆听

单击此处添加副标题内容