

OpenHarmonyOS E53模块开发-人体检测



目 录

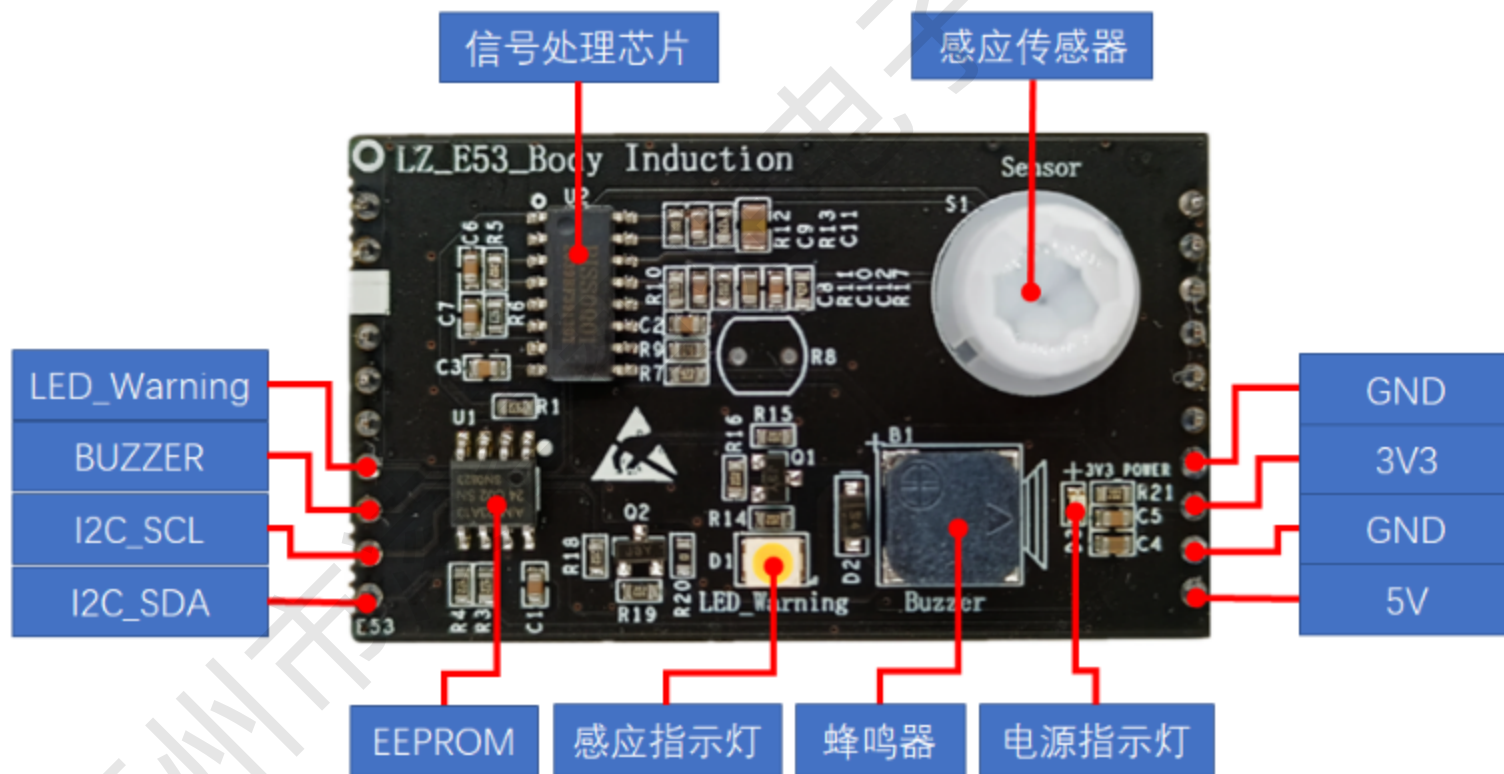
CONTENTS

- [01] 硬件设计
- [02] 工作原理
- [03] API分析
- [04] 实例分析

01

硬件设计

资源介绍



01

硬件设计



02

工作原理

人体检测模块通过检测BISS0001控制灯的输出信号，当检测到高电平时就认为是有物体靠近，通过pwm控制蜂鸣器告警。

03

API分析

```
void e53_bi_init(void);
```

该函数主要功能是E53人体检测模块初始化，包括初始化PWM、灯光开关状态检测GPIO。

```
void buzzer_set_status(SWITCH_STATUS_ENUM status)
```

该函数主要功能是E53人体检测模块蜂鸣器控制功能。其中参数status 为ON开启蜂鸣器，为OFF关闭蜂鸣器。

04

实例分析

1、打开sdk下面路径的文件

```
vendor/lockzhiner/rk2206/samples/c5_e53_body_induction/e53_body_induction_example.c
```

2、创建任务

在e53_bi_example函数中，创建的一个线程e53_bi_thread。

```
void e53_bi_example()
{
    unsigned int ret = LOS_OK;
    unsigned int thread_id;
    TSK_INIT_PARAM_S task = {0};

    task.pfnTaskEntry = (TSK_ENTRY_FUNC)e53_bi_thread;
    task.uwStackSize = 10240;
    task.pcName = "e53_bi_thread";
```

```
    task.usTaskPrio = 24;
    ret = LOS_TaskCreate(&thread_id, &task);
    if (ret != LOS_OK)
    {
        printf("Falied to create e53_bi_thread ret:0x%x\n",
ret);
        return;
    }
}
```

在void e53_bi_thread函数中，初始化GPIO作为输入，初始化PWM作为蜂鸣器控制源。采用轮询的方式读取GPIO电平值，当读取的GPIO值为高电平时，认为有物体靠近，开启蜂鸣器警告；当读取的GPIO值为低电平时，则为没有物体靠近，关闭蜂鸣器警告。

```
void e53_bi_thread()
{
    unsigned int ret;
    LzGpioValue val = LZGPIO_LEVEL_LOW, val_last = LZGPIO_LEVEL_LOW;
    e53_bi_init();
    while (1)
    {
        ret = LzGpioGetVal(GPIO0_PA5, &val);
```

```
        if (ret != LZ_HARDWARE_SUCCESS)
        {
            printf("get gpio value failed ret:%d\n", ret);
        }
        if (val_last != val)
        {
            if (val == LZGPIO_LEVEL_HIGH)
            {
                buzzer_set_status(ON);
                printf("buzzer on\n");
                LOS_Msleep(1000);
                buzzer_set_status(OFF);
                printf("buzzer off\n");
            }
            val_last = val;
        }
        LOS_Msleep(1000);
    }
}
```


3、修改编译脚本

修改 `vendor/lockzhiner/rk2206/sample` 路径下 `BUILD.gn` 文件，指定 `e53_bi_example` 参与编译。

```
"/c5_e53_body_induction:e53_bi_example",
```

修改 `device/lockzhiner/rk2206/sdk_liteos` 路径下 `Makefile` 文件，添加 `-le53_bi_example` 参与编译。

```
hardware_LIBS = -lhal_iohardware -lhardware -le53_bi_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

```
buzzer on
```

```
buzzer off
```

```
.....
```

谢谢聆听

单击此处添加副标题内容