



# 鸿蒙OS设备开发教程（初级篇）

## 51CTO | HarmonyOS技术社区

轻松掌握

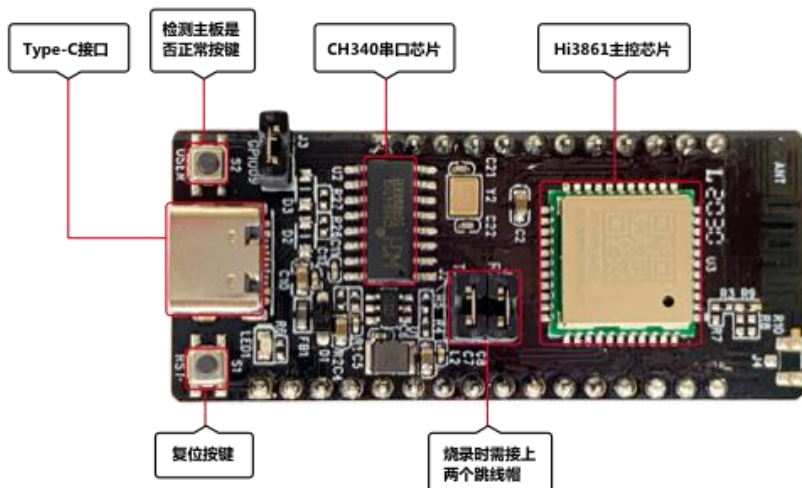
鸿蒙开发板外设控制

唐佐林

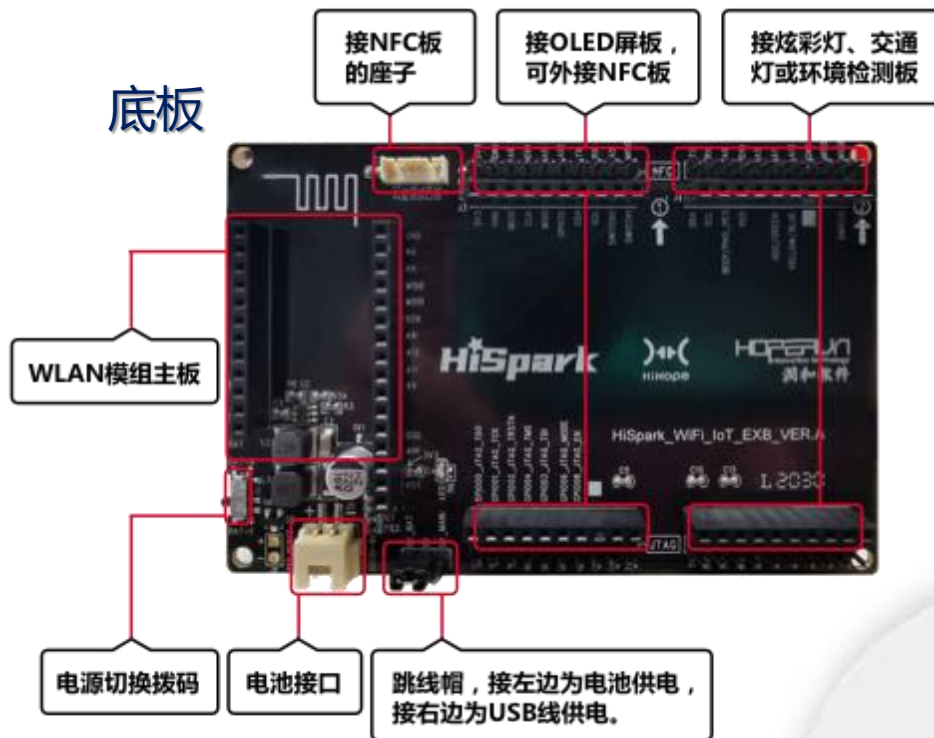
## Agenda

- Hi3861开发框架简介
- LED灯控制示例
- 按键事件响应
- 蜂鸣器控制
- Q/A

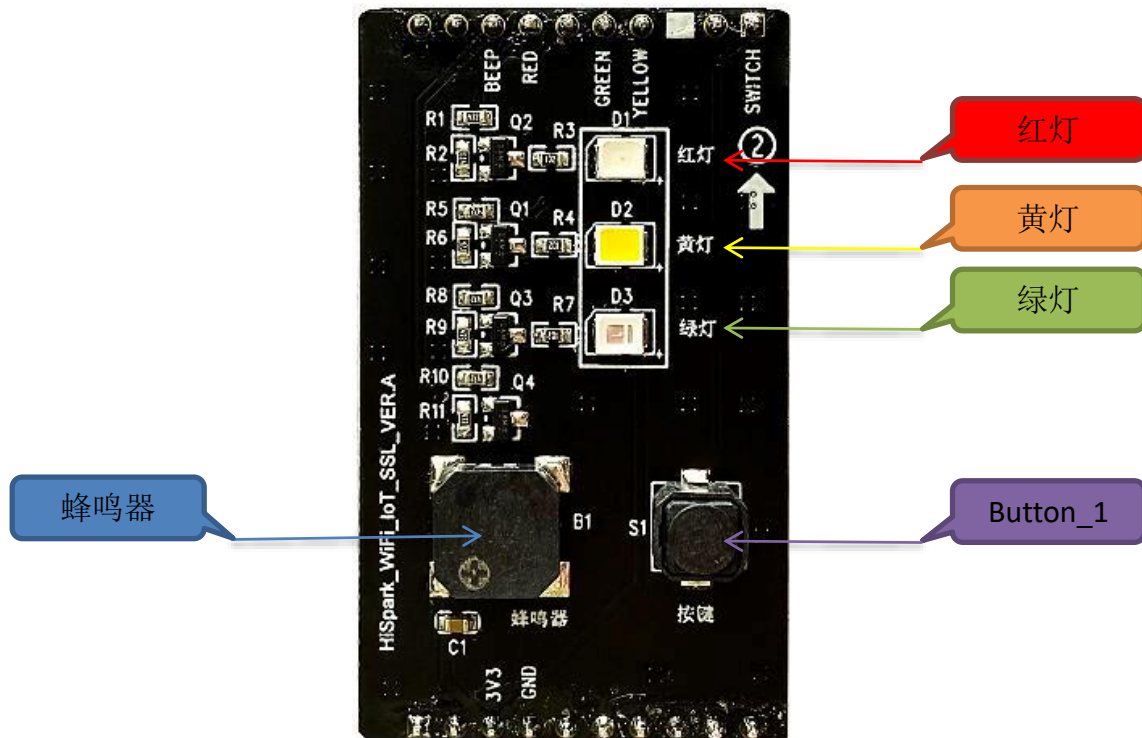
## Hi3861核心板



主板



## Hi3861外设板



## 鸿蒙设备应用开发步骤

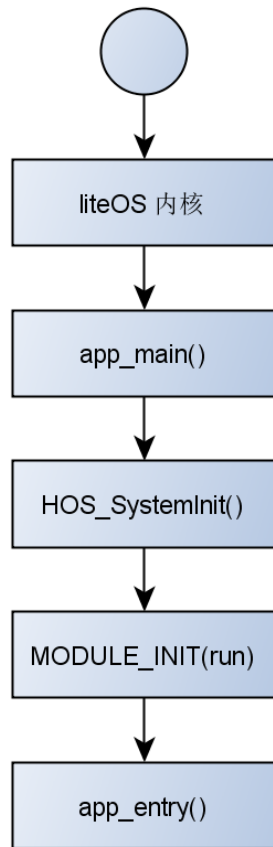
1. 新建任务目录以及源码文件 (./applications/sample/wifi-iot/app)
2. 编写任务源码编译脚本 BUILD.gn
3. 增加 app 模块的任务列表 (修改 ./applications/sample/wifi-iot/app/BUILD.gn)
4. 指定任务入口函数 (app\_entry())

```
static void HelloWorld(void)
{
    printf("[Delphi Tang] Hello World!\n");
}

SYS_RUN(HelloWorld);
```

## 鸿蒙设备应用程序启动流程

1. app\_main.c : app\_main()
2. MODULE\_INIT( run )
3. 任务入口函数 ( app\_entry() )



## 问题

`app_entry()` 是否等价于 C 语言中的 `main()` ?





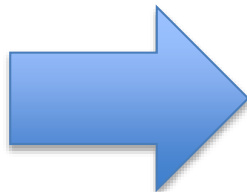
## 鸿蒙设备开发通用架构

```
static void HelloWorld_Entry(void)
{
    osThreadAttr_t attr = {0};

    printf("[HelloWorld] HelloWorld_Entry!\n");

    attr.name = "HelloWorld_Task";
    attr.attr_bits = 0U;
    attr.cb_mem = NULL;
    attr.cb_size = 0U;
    attr.stack_mem = NULL;
    attr.stack_size = 1024;
    attr.priority = osPriorityNormal;

    if (osThreadNew((osThreadFunc_t)HelloWorld_Task, NULL, &attr) == NULL)
    {
        printf("[HelloWorld] Failed to create LedTask!\n");
    }
}
```



```
static void* HelloWorld_Task(const char* arg)
{
    (void)arg;

    printf("[HelloWorld] HelloWorld_Task!\n");

    while(1)
    {
        // logic code for task
        usleep(100000);
    }

    return NULL;
}
```

## 鸿蒙应用外设控制步骤 (LED灯控制)

1. 查原理图, 找到外设对应的 GPIO 引脚
2. 通过系统 API 重新定义 GPIO 引脚的功能, 并设置数据传输方向
3. 编写业务逻辑代码通过 GPIO 控制外设 (发送命令或接收数据)



## 鸿蒙应用外设控制步骤

### GPIO 引脚功能设置:

// 复用引脚为 GPIO

```
IoSetFunc(WIFI_IOT_IO_NAME_GPIO_10, WIFI_IOT_IO_FUNC_GPIO_10_GPIO);
```

// 设置为输出

```
GpioSetDir(WIFI_IOT_IO_NAME_GPIO_10, WIFI_IOT_GPIO_DIR_OUT);
```

### GPIO 引脚上的数据设置:

```
GpioSetOutputVal(WIFI_IOT_IO_NAME_GPIO_10, 0);
```

```
usleep(300000);
```

```
GpioSetOutputVal(WIFI_IOT_IO_NAME_GPIO_10, 1);
```

```
usleep(300000);
```

## 鸿蒙应用外设控制步骤（按键编程）

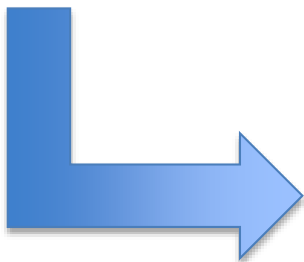
1. 查原理图，找到按键对应的 GPIO 引脚
2. 通过系统 API 重新定义 GPIO 引脚的功能，设置数据传输方向以及回调函数
3. 编写业务逻辑代码处理按键事件（按下事件 或 释放事件）



## 鸿蒙应用按键编程步骤

### GPIO 引脚功能设置

```
//复用引脚为 GPIO
IoSetFunc(WIFI_IOT_IO_NAME_GPIO_8, WIFI_IOT_IO_FUNC_GPIO_8_GPIO);
//设置为输入
GpioSetDir(WIFI_IOT_IO_NAME_GPIO_8, WIFI_IOT_GPIO_DIR_IN);
IoSetPull(WIFI_IOT_IO_NAME_GPIO_8, WIFI_IOT_IO_PULL_UP);
GpioRegisterIsrFunc(WIFI_IOT_IO_NAME_GPIO_8,
                    WIFI_IOT_INT_TYPE_EDGE,
                    WIFI_IOT_GPIO_EDGE_FALL_LEVEL_LOW,
                    OnButtonPressed, NULL);
```



### 按键回调函数

```
static void OnButtonPressed(char* arg)
{
    (void)arg;

    printf("[HelloWorld] OnButtonPressed()\n");
}
```

## 鸿蒙应用外设控制步骤（蜂鸣器控制）

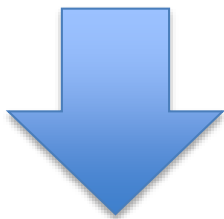
1. 配置使用蜂鸣器：CONFIG\_PWM\_SUPPORT=y (usr\_config.mk)
2. 关联 GPIO 引脚到蜂鸣器：WIFI\_IOT\_IO\_FUNC\_GPIO\_9\_PWM0\_OUT
3. 设置 GPIO 引脚为数据输出
4. 调用 Pwmlnit(), PwmStart(), PwmStop()



## 鸿蒙应用蜂鸣器编程步骤

### GPIO 引脚功能设置

```
//复用引脚为 GPIO
IoSetFunc(WIFI_IOT_IO_NAME_GPIO_9, WIFI_IOT_IO_FUNC_GPIO_9_PWM0_OUT);
//设置为输入
GpioSetDir(WIFI_IOT_IO_NAME_GPIO_9, WIFI_IOT_GPIO_DIR_OUT);
PwmInit(WIFI_IOT_PWM_PORT_PWM0);
```



### 蜂鸣器控制

```
if( BtnPressed )
|   PwmStart(WIFI_IOT_PWM_PORT_PWM0, 5 * 1000, 40 * 1000);
else
|   PwmStop(WIFI_IOT_PWM_PORT_PWM0);
```





# THANKS



扫码了解更多

版权声明，本文档全部内容版权归成都狄泰未来科技有限公司所有。