

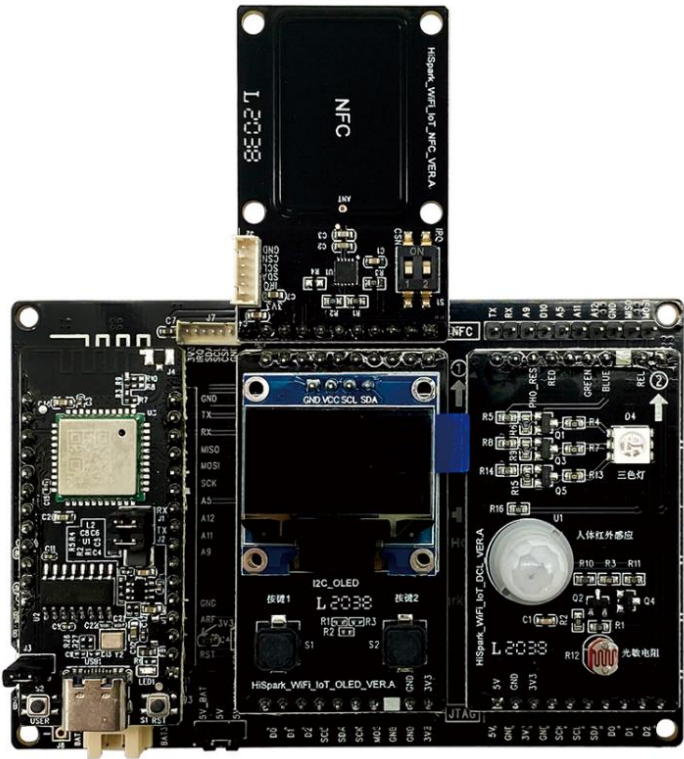
GPIO口基础与应用

内容

- Pegasus智能家居开发套件
- 实验2-1 点亮LEDgn软件架构
- 外设驱动框架
- 实验2-2 按键控制LED

一、Pegasus智能家居开发套件

• 外观与参数



产品参数	
SoC	<ul style="list-style-type: none">■ 基于海思Hi3861V100高度集成的2.4GHz WiFi芯片■ 内部集成高性能32bit微处理器、硬件安全引擎以及丰富的外设接口
操作系统	<ul style="list-style-type: none">■ 支持OpenHarmony、LiteOS和第三方组件，可与华为Hi-Link协同
通信能力	<ul style="list-style-type: none">■ 支持复杂环境下TPC、自动速率、弱干扰免疫等可靠性通信算法
组网能力	<ul style="list-style-type: none">■ 支持256节点Mesh组网■ 支持20MHz标准带宽和5MHz/10MHz窄带宽，提供最大72.2Mbit/s物理层速率
网络能力	<ul style="list-style-type: none">■ 支持IPv4 /IPv6 /DHCPv4 /DHCPv6 Client /Server/DNS Client /mDNS /CoAP /MQTT /HTTP /JSON■ 集成IEEE 802.11b/g/n基带和RF电路■ WiFi基带支持正交频分复用（OFDM）
安全能力	<ul style="list-style-type: none">■ 支持AES128/256加解密/HASH-SHA256/HMAC_SHA256 /RSA■ 支持ECC签名校验算法■ 真随机数生成，满足FIPS140-2随机测试标准■ 支持TLS/DTLS加速■ 内部集成EFUSE、安全存储、安全启动、硬件ID■ 集成MPU特性，支持内存隔离特性

一、Pegasus智能家居开发套件

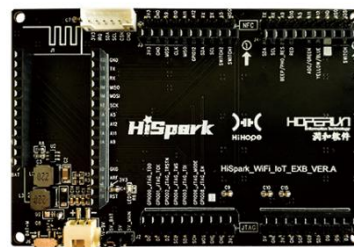
• 硬件组成



(尺寸：53mm*22.8mm)

主板

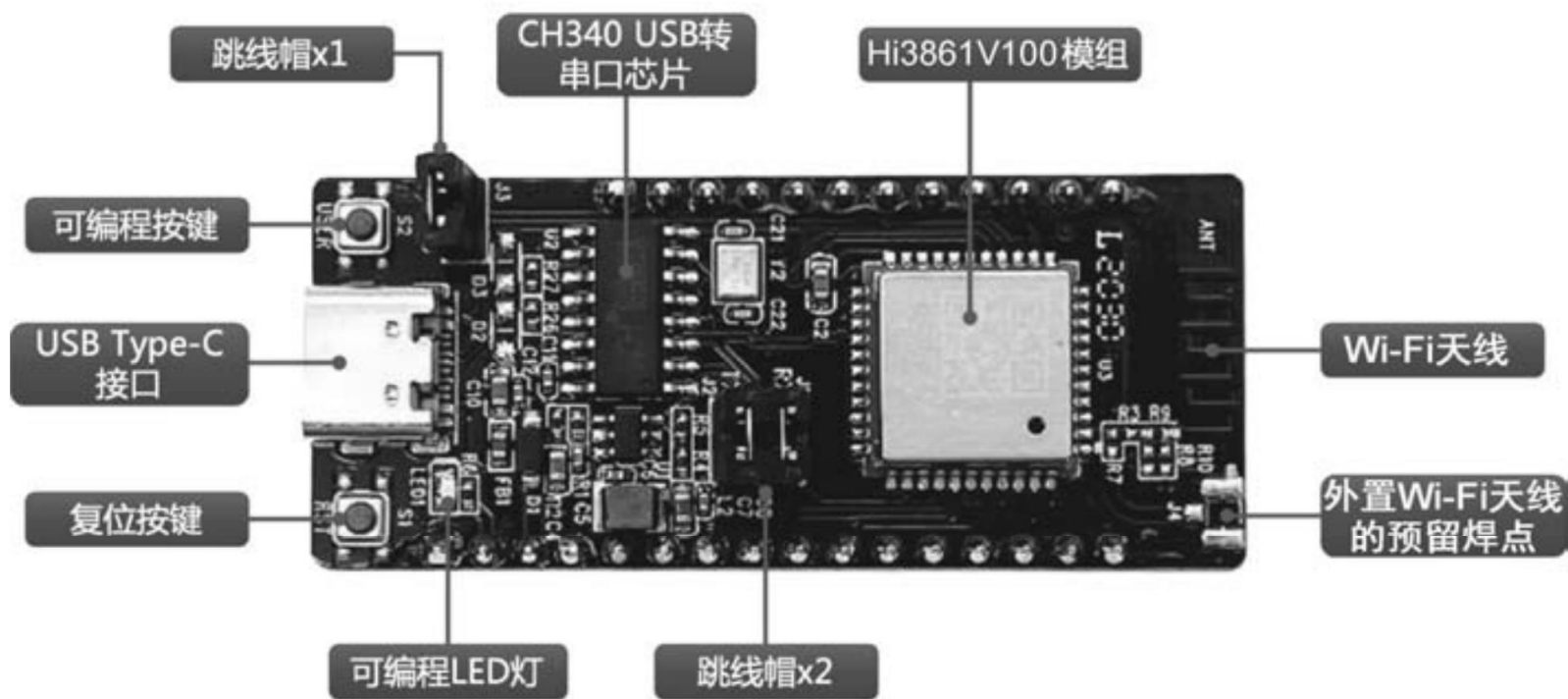
- Type-C型USB接口
- 板载2.4G WiFi天线
- 丰富的管脚功能



(尺寸：100mm*66.6mm)

通用底板

- 专用主板的插槽
- 配备多种传感器板接口
- 具有丰富的丝印接口说明，降低开发难度
- 预留丰富的插槽、卡位、排针，提升开发自主性
- 板载锂电池接口
- 智能小车基础底板方案



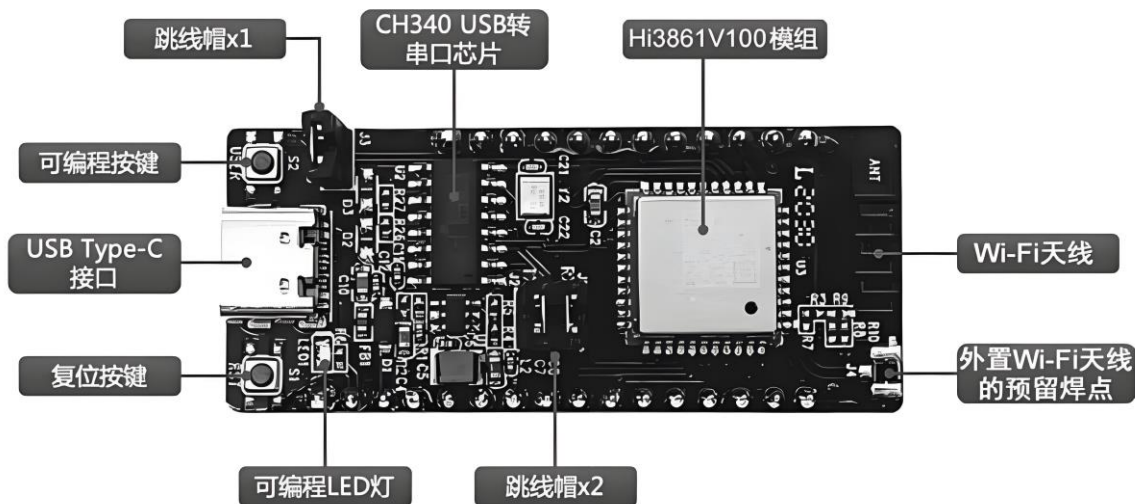
一、Pegasus智能家居开发套件

• 硬件组成

扩展板		
显示板 (尺寸: 50.85mm*30mm)		<ul style="list-style-type: none">■ 配备0.96英寸OLED显示屏■ I²C通信方式■ 搭配两个功能自定义用户按键
NFC板 (尺寸: 42.55mm*75.55mm)		<ul style="list-style-type: none">■ I²C通信方式■ 板子NFC大线圈，灵敏度高，响应速度快，具有良好的用户体验
环境监测板 (尺寸: 50.85mm*30mm)		<ul style="list-style-type: none">■ 温湿度一体传感器■ 可燃气体传感器■ 一枚功能用户自定义蜂鸣器
红绿灯板 (尺寸: 50.85mm*30mm)		<ul style="list-style-type: none">■ 红、黄、绿，三颗三色LED，便于模拟信号灯■ 一个功能用户自定义按键■ 一枚功能用户自定义蜂鸣器
炫彩灯板 (尺寸: 50.85mm*30mm)		<ul style="list-style-type: none">■ 一颗三色LED，通过三个GPIO控制■ 一枚功能用户自定义蜂鸣器■ 光照传感器
机器人板 (尺寸: 50.85mm*30mm)		<ul style="list-style-type: none">■ 具有丰富的机器人（智能小车）驱动以及接口，包括寻迹传感器接口/超声波传感器接口/电机接口/舵机接口/串口通信接口/I²C接口等

一、Pegasus智能家居开发套件

• 核心板



- Hi3861V100 模组
- Wi-Fi 天线
- 外置 Wi-Fi 天线的预留焊点
- 两个跳线帽
- 一个可编程 LED 灯
- 一个复位按键
- 一个 USB Type-C 接口
- 一个可编程按键
- 一个跳线帽
- CH340USB转串口芯片

一、Pegasus智能家居开发套件

- Hi3861V100 模组

- Hi3861V100 模组是搭载 Hi3861V100 芯片的物联网 Wi - Fi 模组
- 高度集成 2.4GHz SoC WiFi 芯片, 符合 802.11 b/g/n 协议
- 具备 32 位高能效的 RISC-V 指令集架构
- 最大工作频率能够达到 160MHz
- 拥有 352KB 的 SRAM, MB 的 Flash 容量
- 15 个通用输入/输出
- 7 路模数转换器 (Analog to Digital Converter, ADC) 输入
- 6路脉宽调制 (Pulse Width Modulation, PWM) 输出
- 3 个通用异步收发器 (Universal Asynchronous Receiver & Transmitter, UART) 接口
- 1 个串行外设接口 (Synchronous Peripheral Interface, SPI)

一、Pegasus智能家居开发套件

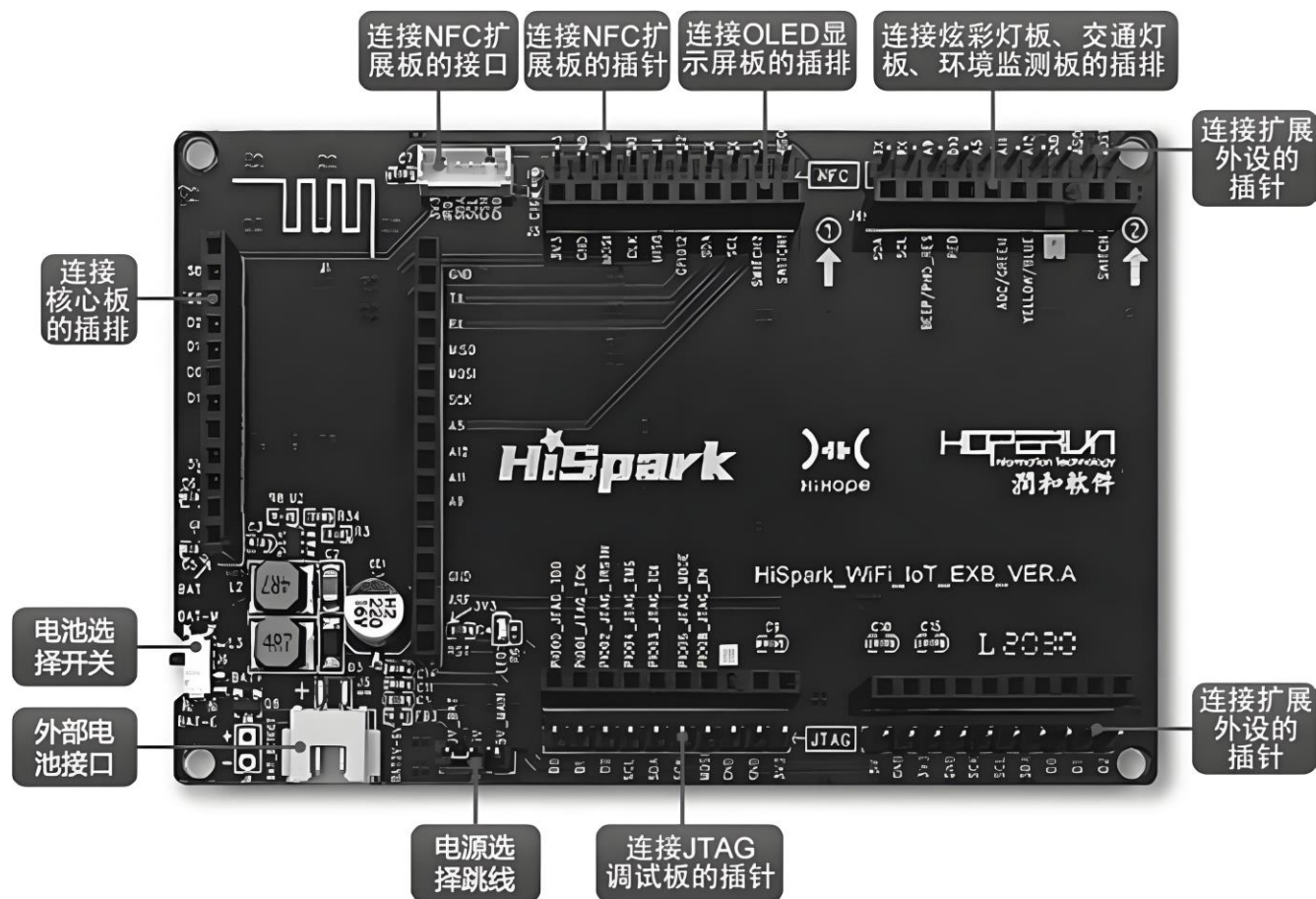
- CH340 USB 转串口芯片
 - 经典的串口调试芯片
 - 实现 UART 接口和 USB Type-C 接口间的信号转换
- USB Type-C 接口
 - 为核心板及整个套件进行供电
 - 连接到电脑的 USB 接口，进行串口调试和系统烧录
- 复位按键
 - 位按键被标记为“RST”，也就是 RESET
 - 它可以触发主控芯片的 CPU 硬件复位，使得程序重新开始执行
- 可编程按键
 - 被标记为“USER”，用于程序的按键输入

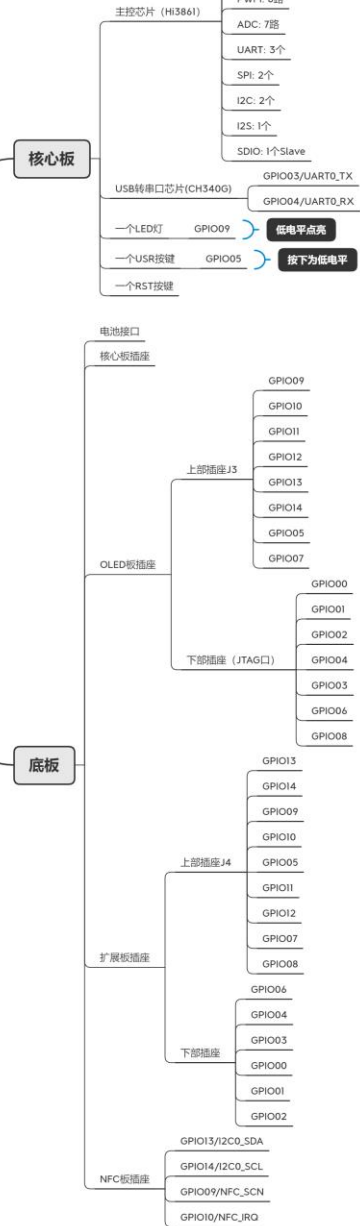
一、Pegasus智能家居开发套件

- 可编程 LED 灯
 - 被标记为“LED1”，用于显示程序的运行时状态
 - 可以编写程序控制它的点亮或者熄灭
- 两组跳线帽
 - 右侧的两个跳线帽分别被标记为 RX 和 TX，分别用于连接主控芯片 UART 接口的TX 和 RX 引脚与 CH340 USB 转串口芯片的 RX 和 TX 引脚
 - 左侧的一个跳线帽被标记为 GPIO-09，用于连接主控芯片和可编程 LED 灯

一、Pegasus智能家居开发套件

- 底板

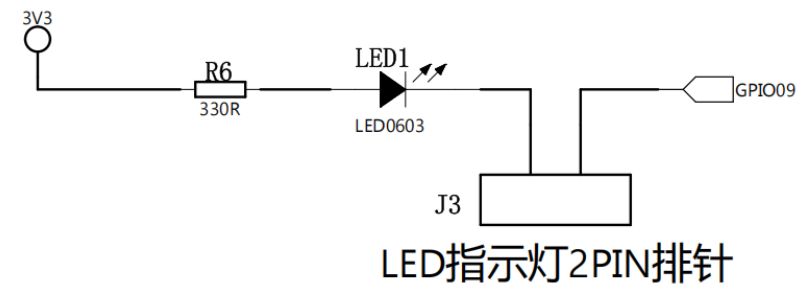
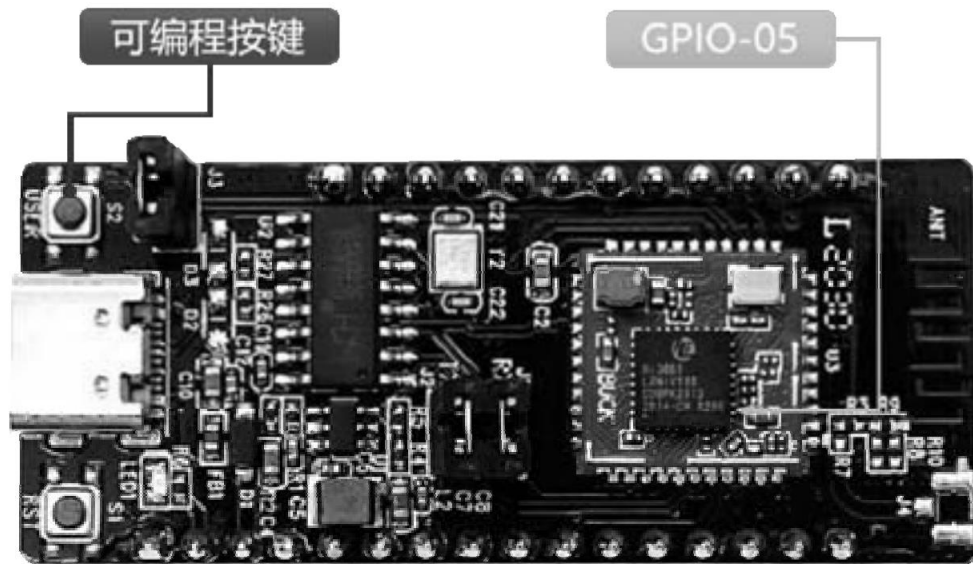
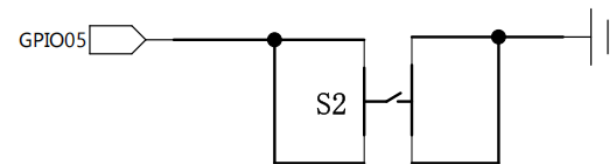




二、实验2-1 点亮LED

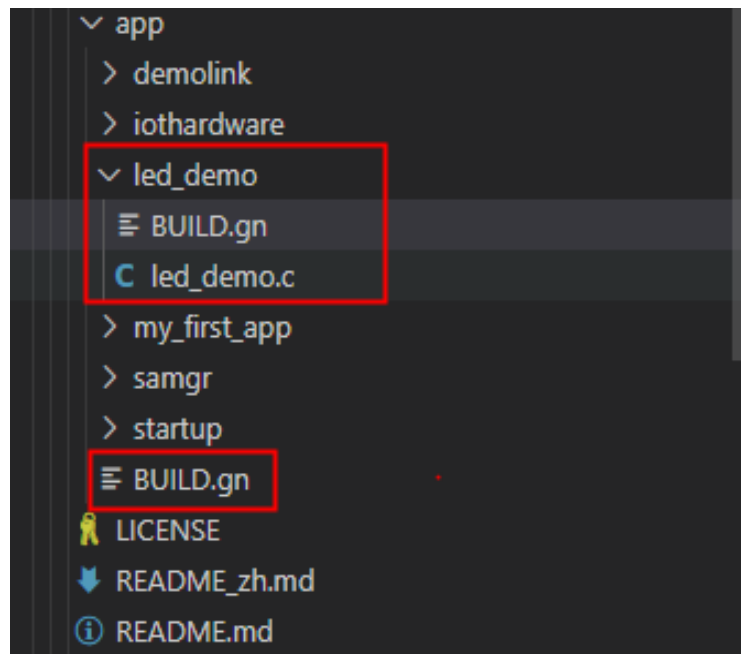
• 连接方式

- LED1正极接3.3V，负极接GPIO09
- 按键S2一端接GPIO05，另外一端接地



二、实验2-1 点亮LED

- 源码结构



二、实验2-1 点亮LED

- 第一个BUILD.gn文件内容

```
1.     static_library("led_demo") {  
2.         sources = [  
3.             "led_demo.c"  
4.         ]  
5.  
6.         include_dirs = [  
7.             "//utils/native/lite/include",  
8.             "//kernel/liteos_m/components/cmsis/2.0",  
9.             "//base/iot_hardware/peripheral/interfaces/kits",  
10.        ]  
11.    }
```

二、实验2-1 点亮LED

- 第二个BUILD.gn内容

```
import("//build/lite/config/component/lite_component.gni")

lite_component("app") {
    features = [
        "led_demo:led_demo",
    ]
}
```


二、实验2-1 点亮LED

- led_demo.c内容

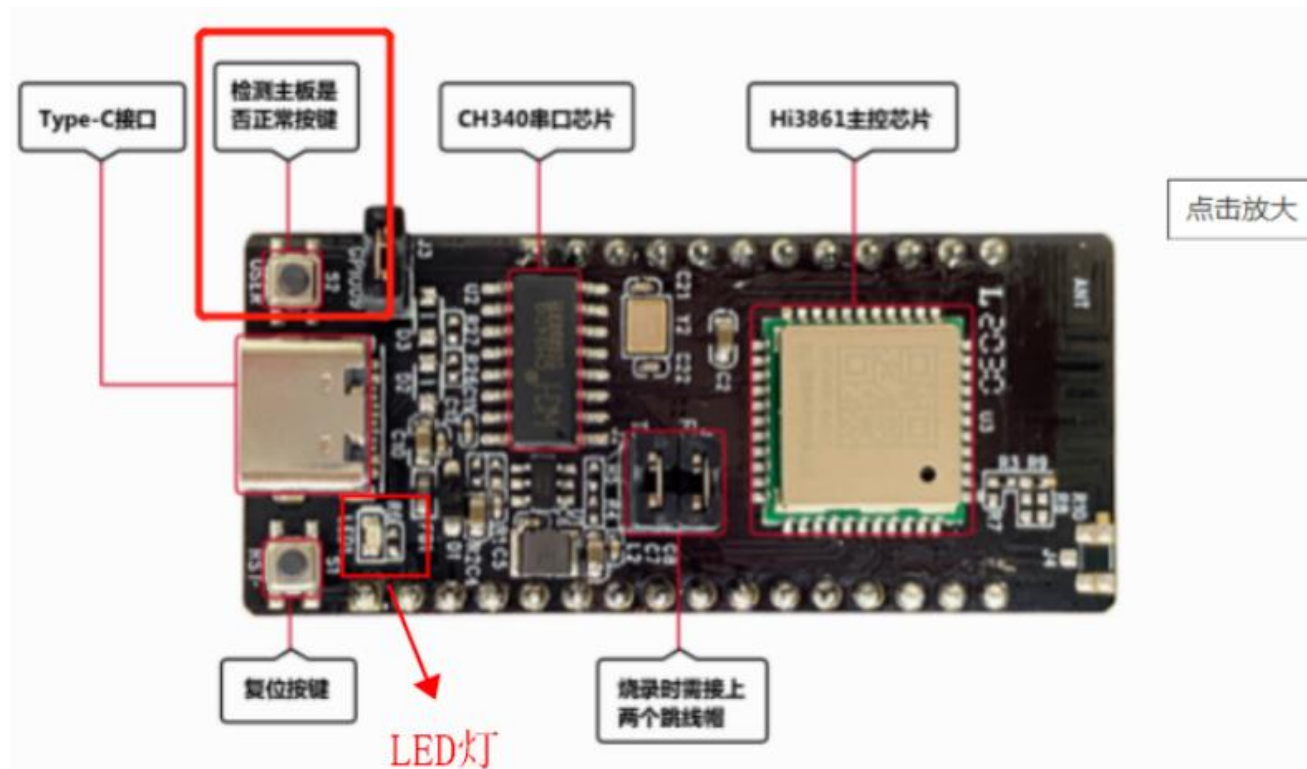
```
31. void led_demo(void)
32. {
33.     osThreadAttr_t attr;
34.
35.     attr.name = "LedTask";
36.     attr.attr_bits = 0U;
37.     attr.cb_mem = NULL;
38.     attr.cb_size = 0U;
39.     attr.stack_mem = NULL;
40.     attr.stack_size = 512;
41.     attr.priority = 26;
42.
43.     if (osThreadNew((osThreadFunc_t)LedTask, NULL, &attr) == NULL) {
44.         printf("[LedExample] Failed to create LedTask!\n");
45.     }
46.
47. }
```

```
1. #include <unistd.h>
2. #include "stdio.h"
3. #include "ohos_init.h"
4. #include "cmsis_os2.h"
5. #include "iot_gpio.h"
6.
7. #define LED_TEST_GPIO 9 // for hispark_pegasus
8.
9. void *LedTask(const char *arg)
10. {
11.     // 初始化GPIO
12.     IoTGPIOInit(LED_TEST_GPIO);
13.
14.     // 设置为输出
15.     IoTGPIOSetDir(LED_TEST_GPIO, IOT_GPIO_DIR_OUT);
16.
17.     (void)arg;
18.     while (1)
19.     {
20.         // 输出低电平
21.         IoTGPIOSetDir(LED_TEST_GPIO, 0);
22.         usleep(300000);
23.         // 输出高电平
24.         IoTGPIOSetDir(LED_TEST_GPIO, 1);
25.         usleep(300000);
26.     }
27.
28.     return NULL;
29. }
```

二、实验2-1 点亮LED

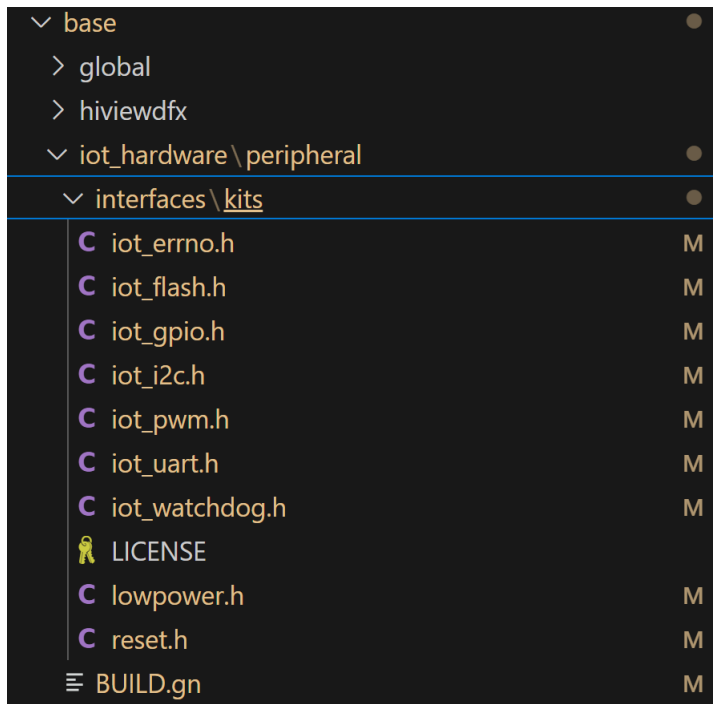
- 现象

- 编译后烧录进去，应该可以看到复位按键旁边的LED灯一直在闪烁



三、外设驱动框架

- OpenHarmony为轻量系统提供了一套简单的驱动封装接口，函数的定义相关头文件位于
“base\iot_hardware\peripheral\interfaces\kits”

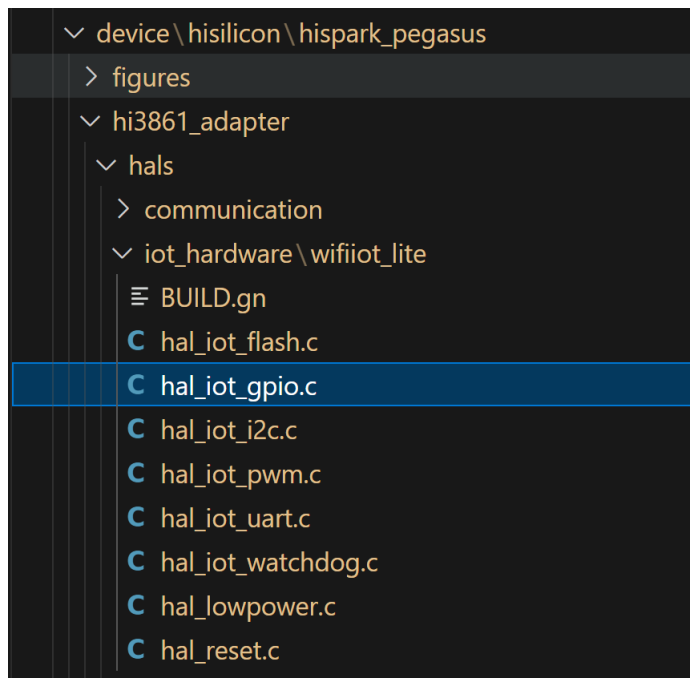


```
42  /**
43   * @brief Enumerates GPIO level values.
44   */
45  typedef enum {
46      /** Low GPIO level */
47      IOT_GPIO_VALUE0 = 0,
48      /** High GPIO level */
49      IOT_GPIO_VALUE1
50  } IotGpioValue;
51
52  /**
53   * @brief Enumerates GPIO directions.
54   */
55  typedef enum {
56      /** Input */
57      IOT_GPIO_DIR_IN = 0,
58      /** Output */
59      IOT_GPIO_DIR_OUT
60  } IotGpioDir;
```

```
88  /**
89   * @brief Initializes a GPIO device.
90   *
91   * @param id Indicates the GPIO pin number.
92   * @return Returns {@link IOT_SUCCESS} if the
93   *         returns {@link IOT_FAILURE} otherwise. For
94   * @since 2.2
95   * @version 2.2
96   */
97  unsigned int IoTGpioInit(unsigned int id);
```

三、外设驱动框架

- 具体的函数实现，需要在对应的soc中



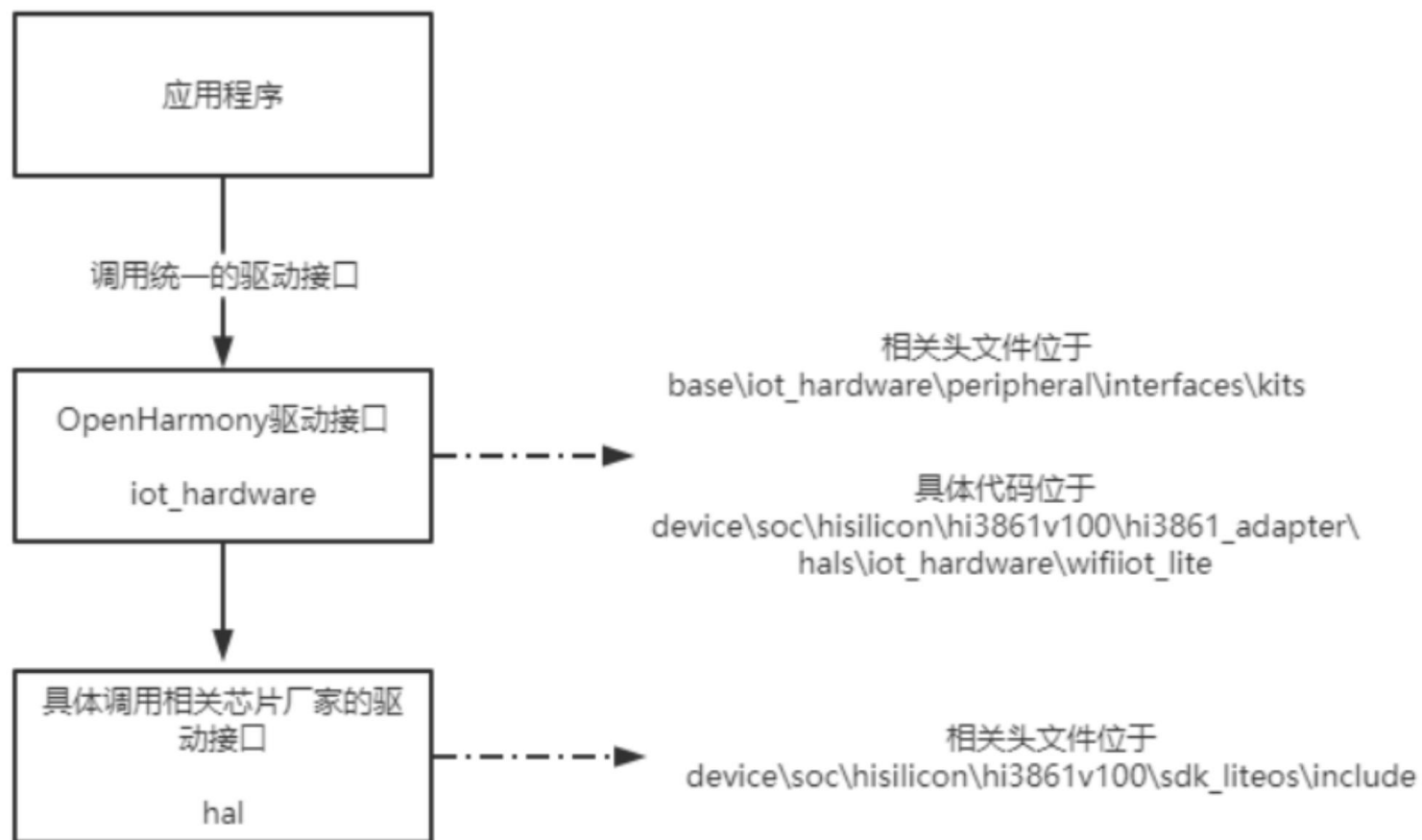
```
unsigned int IoTGPIOSetDir(unsigned int id, IotGpioDir dir)
{
    return hi_gpio_set_dir((hi_gpio_idx)id, (hi_gpio_dir)dir);
}

unsigned int IoTGPIOGetDir(unsigned int id, IotGpioDir *dir)
{
    return hi_gpio_get_dir((hi_gpio_idx)id, (hi_gpio_dir *)dir);
}

unsigned int IoTGPIOSetOutputVal(unsigned int id, IotGpioValue val)
{
    return hi_gpio_set_output_val((hi_gpio_idx)id, (hi_gpio_value)val);
}

unsigned int IoTGPIOGetOutputVal(unsigned int id, IotGpioValue *val)
{
    return hi_gpio_get_output_val((hi_gpio_idx)id, (hi_gpio_value *)val);
}
```

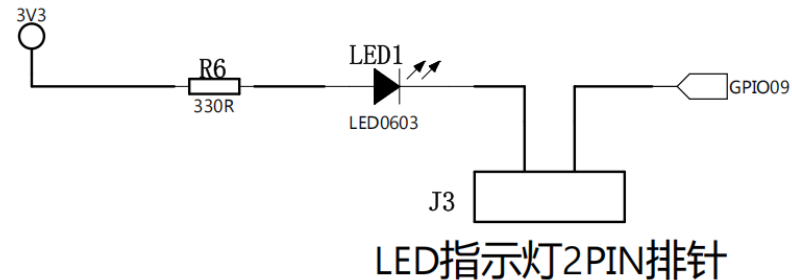
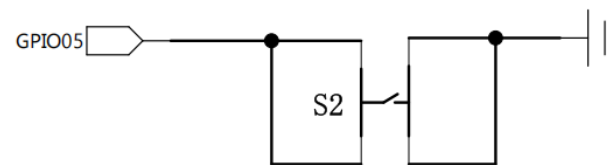
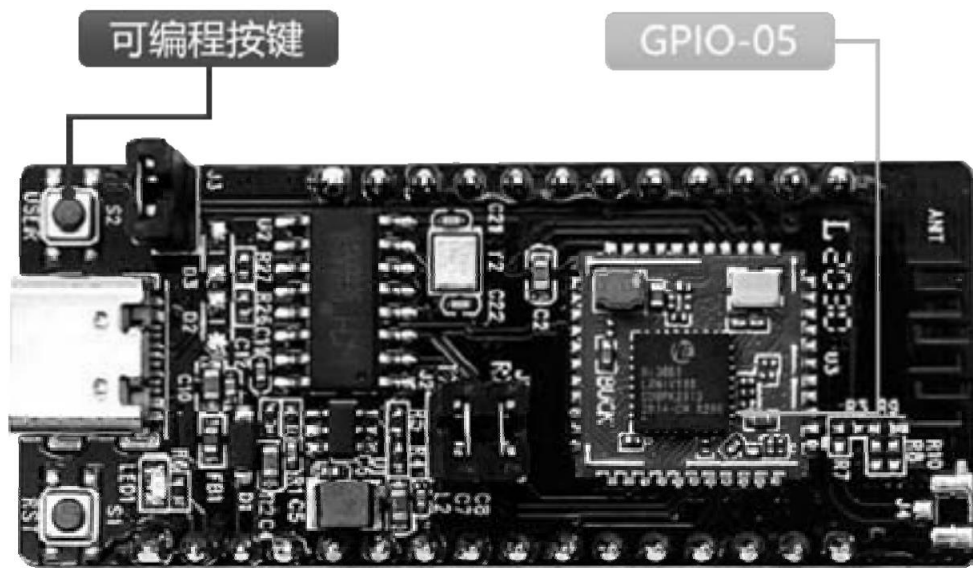
三、外设驱动框架



三、实验2-2 按键控制LED

• 连接方式

- LED1正极接3.3V，负极接GPIO09
- 按键S2一端接GPIO05，另外一端接地



四、实验2-2 按键控制LED

- 实验要求

- 当核心板上的按键S2按下时，LED灯亮，并在串口监视器上显示“HI_GPIO_VALUE_1”；当S2抬起时，LED灯灭，并在串口监视器上显示

