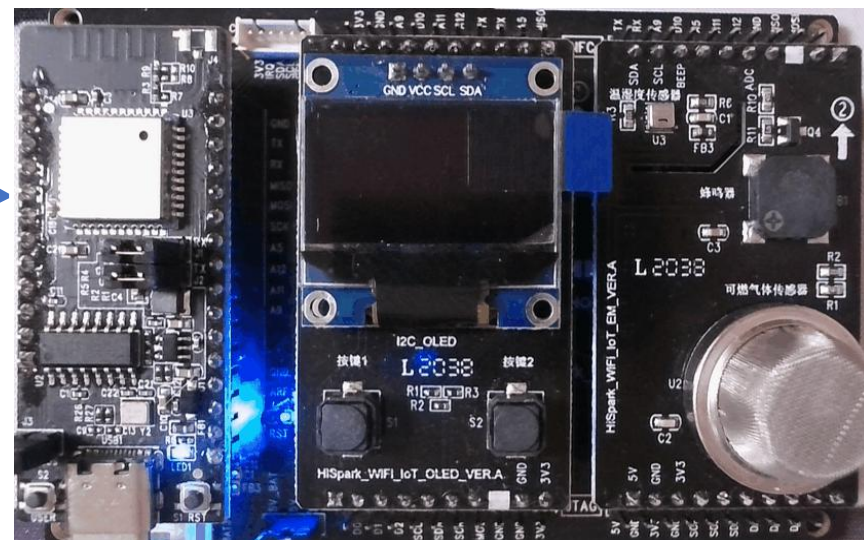


# Harmony OS 智能硬件 入门系列课程 <快速上手>

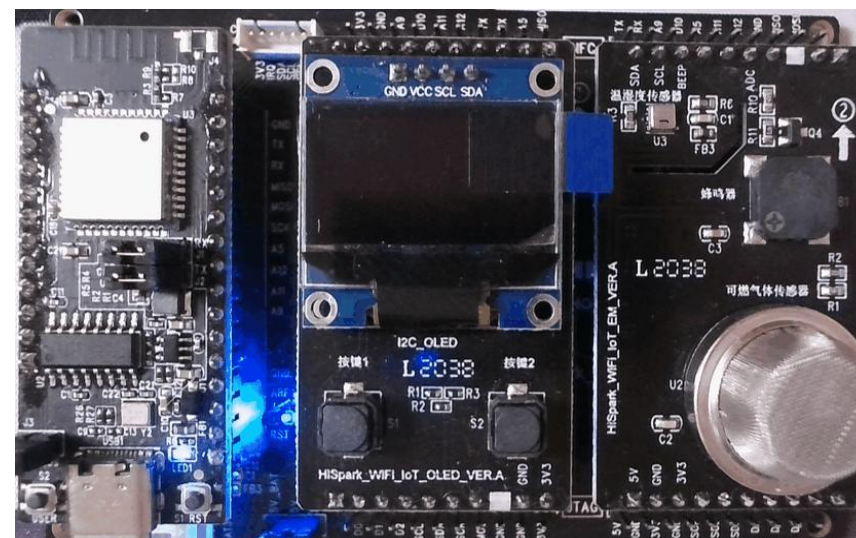
快速掌握Hi3861开发板基础开发技巧

## 第 4 讲：Hi3861外设——环境监测板开发



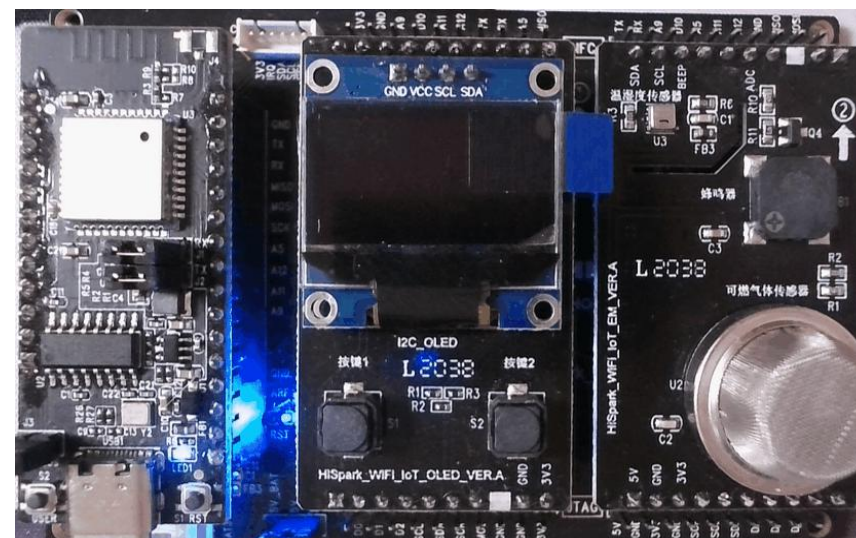
# 本讲内容

- 第1节：AHT20温湿度传感器数据读取
- 第2节：MQ-2燃气传感器数据读取
- 第3节：OLED显示数据
- 第4节：综合案例讲解



# 本讲目标

- 1、掌握Hi3861上AHT20温湿度传感器使用方法
- 2、掌握Hi3861上MQ-2燃气传感器使用方法
- 3、掌握Hi3861上OLED显示数据使用使用步骤
- 4、理解综合案例



# 第1节： AHT20温湿度传感器数据读取

- 知识点1：板载器件介绍
- 知识点2：电路图分析
- 知识点3：代码实现温湿度读取



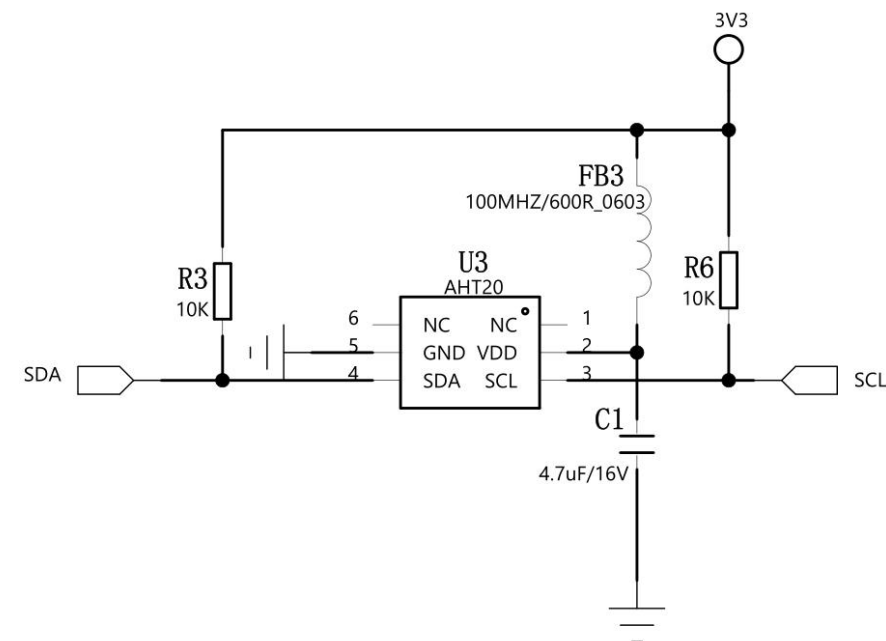
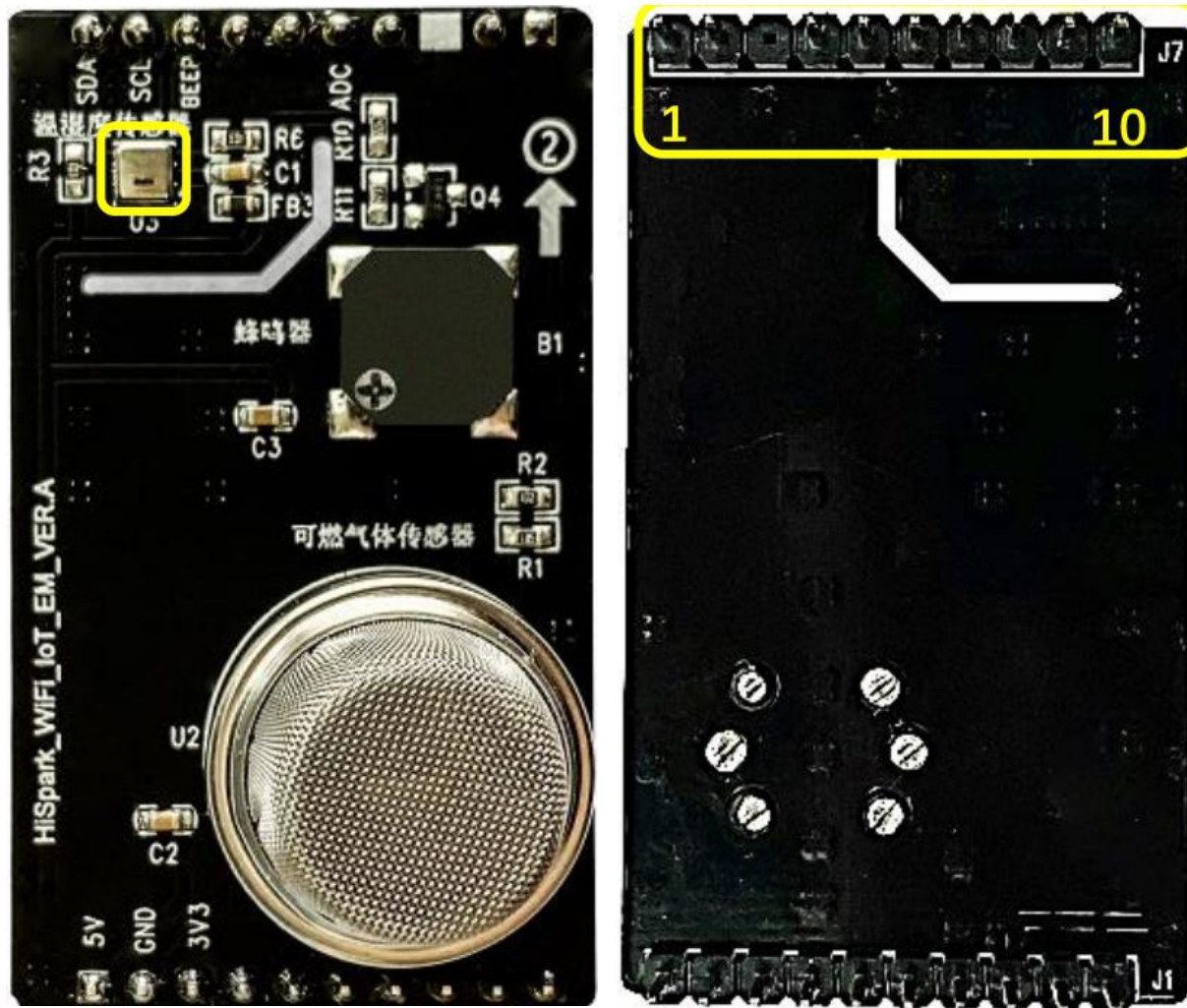


## 知识点1 【板载器件介绍】

- 型号：AHT20 温湿度传感器
- 产品特点：
  1. 测量范围：
    - 温度：-40°C to 80°C
    - 湿度：0%RH to 80%RH
  2. 相对湿度误差 $\pm 2\%RH$
  3. 温度误差 $\pm 0.3\%$

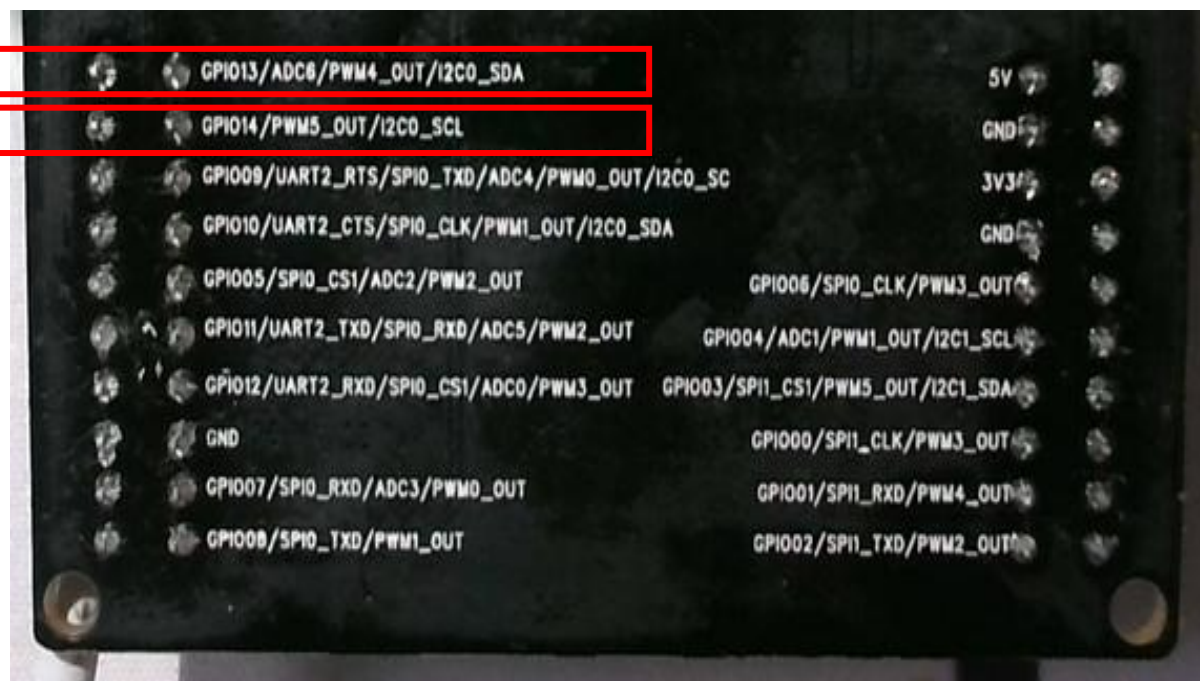


## 知识点2 【电路图分析】



- Hi3861 通过 J7 接口 SCL 引脚、SDA 引脚以 IIC 接口方式访问该传感器

## 知识点2 【电路图分析】



- GPIO13 连接SDA 引脚
- GPIO14 连接SCL 引脚

## 知识点3 【代码实现温湿度读取】

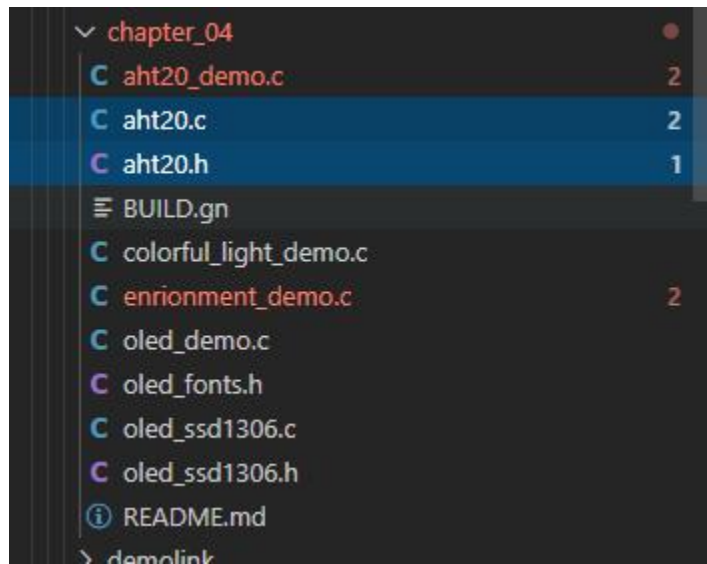
实现步骤:

1. 创建工程目录chapter\_04, 并导入接口文件aht20.c、 aht20.h
2. 创建aht20\_demo.c, 创建加入头文件
3. 初始化I2C
4. 完成温湿度数据读取功能
5. 注册自启动线程
6. 打开I2C的CONFIG选项



## 知识点3 【代码实现温湿度读取】

1. 步骤1：创建工程目录chapter\_04，并导入接口文件aht20.c、aht20.h



## 知识点3 【代码实现温湿度读取】

步骤2：创建aht20\_demo.c，创建加入头文件

<code/>

```
#include <stdio.h>
#include <stdint.h>
#include <string.h>
#include <unistd.h>
#include "ohos_init.h"
#include "cmsis_os2.h"
#include "wifiiot_i2c.h"
#include "wifiiot_gpio.h"
#include "wifiiot_gpio_ex.h"
#include "wifiiot_errno.h"
#include "aht20.h"

#define AHT20_BAUDRATE 400 * 1000
#define AHT20_I2C_IDX WIFI_IOT_I2C_IDX_0
```



## 知识点3 【代码实现温湿度读取】

步骤3：初始化I2C

**<code/>**

```
void init(void)
{
    GpioInit();
    IoSetFunc(WIFI_IOT_IO_NAME_GPIO_13, WIFI_IOT_IO_FUNC_GPIO_13_I2C0_SDA);
    IoSetFunc(WIFI_IOT_IO_NAME_GPIO_14, WIFI_IOT_IO_FUNC_GPIO_14_I2C0_SCL);
    I2cInit(AHT20_I2C_IDX, AHT20_BAUDRATE);
}
```

## 知识点3 【代码实现温湿度读取】

### 步骤4：完成温湿度数据读取功能

<code/>

```
static void AhtDemoTask(void *arg){
    (void)arg;
    uint32_t retval = 0;
    float humidity = 0.0f;
    float temperature = 0.0f;

    while (WIFI_IOT_SUCCESS != AHT20_Calibrate()){
        printf("AHT20 sensor init failed!\r\n");
        usleep(1000);
    }
    while (1){
        retval = AHT20_StartMeasure();
        if (retval != WIFI_IOT_SUCCESS){
            printf("trigger measure failed!\r\n");
        }else {
            retval = AHT20_GetMeasureResult(&temperature, &humidity);
            printf("temp: %.2f, humi: %.2f\r\n", temperature, humidity);
        }
        sleep(1);
    }
}
```



## 知识点3 【代码实现温湿度读取】

### 步骤5：注册自启动线程

<code/>

```
static void AhtDemo(void)
{
    osThreadAttr_t attr;
    init();
    attr.name = "AhtDemoTask";
    attr.attr_bits = 0U;
    attr.cb_mem = NULL;
    attr.cb_size = 0U;
    attr.stack_mem = NULL;
    attr.stack_size = 4096;
    attr.priority = osPriorityNormal;

    if (osThreadNew(AhtDemoTask, NULL, &attr) == NULL)
    {
        printf("[EnvironmentDemo] Falied to create AhtDemoTask!\n");
    }
}

APP_FEATURE_INIT(AhtDemo);
```

## 知识点3 【代码实现温湿度读取】

### 步骤6：打开I2C的CONFIG选项

- 修改vendor\hisi\hi3861\hi3861\build\config\usr\_config.mk文件中的CONFIG\_PWM\_SUPPORT行

```
# 默认情况下，hi3861_sdk中，I2C的CONFIG选项没有打开  
# CONFIG_I2C_SUPPORT is not set  
CONFIG_I2C_SUPPORT=y
```

## 知识点3 【代码实现温湿度读取】

./applications/sample/wifi-iot/app/chapter\_04/BUILD.gn文件

**<code/>**

```
static_library("sensing_demo") {  
    sources = [  
        "aht20_demo.c", "aht20.c", "oled_ssd1306.c"  
    ]  
  
    include_dirs = [  
        "../utils/native/lite/include",  
        "../kernel/liteos_m/components/cmsis/2.0",  
        "../base/iot_hardware/interfaces/kits/wifiot_lite",  
    ]  
}
```



## 知识点3 【代码实现温湿度读取】

./applications/sample/wifi-iot/app/BUILD.gn文件

**<code/>**

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

lite_component("app") {
  features = [
    # "chapter_03:chapter_03_demo",
    "chapter_04:sensing_demo"
  ]
}
```



## 知识点3 【代码实现温湿度读取】

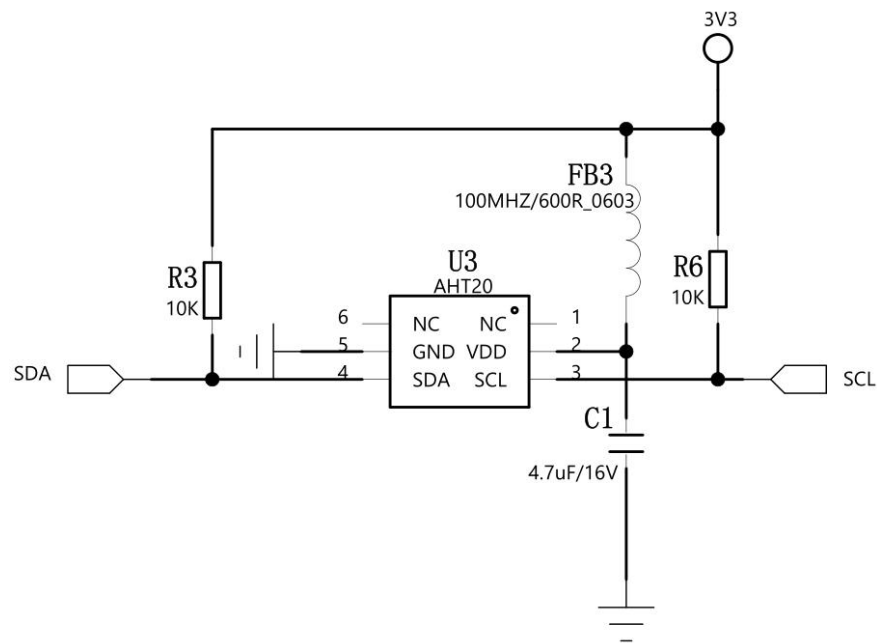
操作演示



## 本节小结

本讲所学知识点有：

- 知识点1：板载器件介绍
- 知识点2：电路图分析
- 知识点3：代码实现温湿度读取



## 第2节：MQ-2燃气传感器数据读取

- 知识点1：板载器件介绍
- 知识点2：电路图分析
- 知识点3：代码实现可燃气体浓度值读取



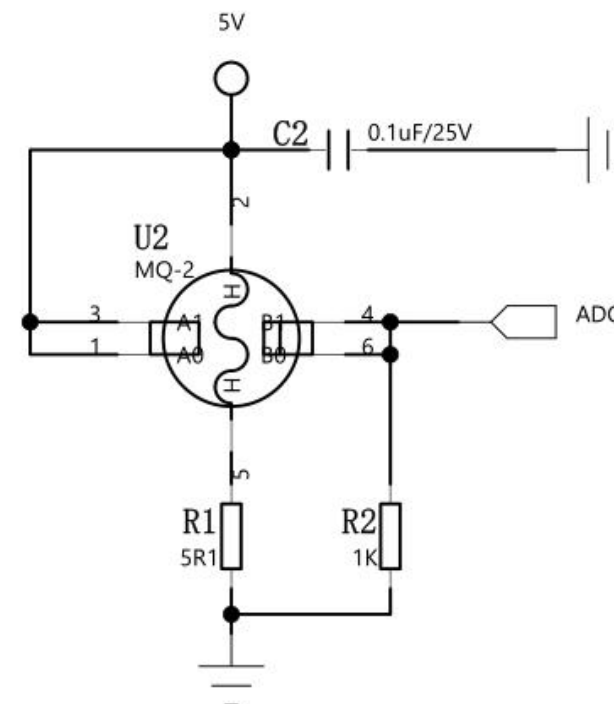
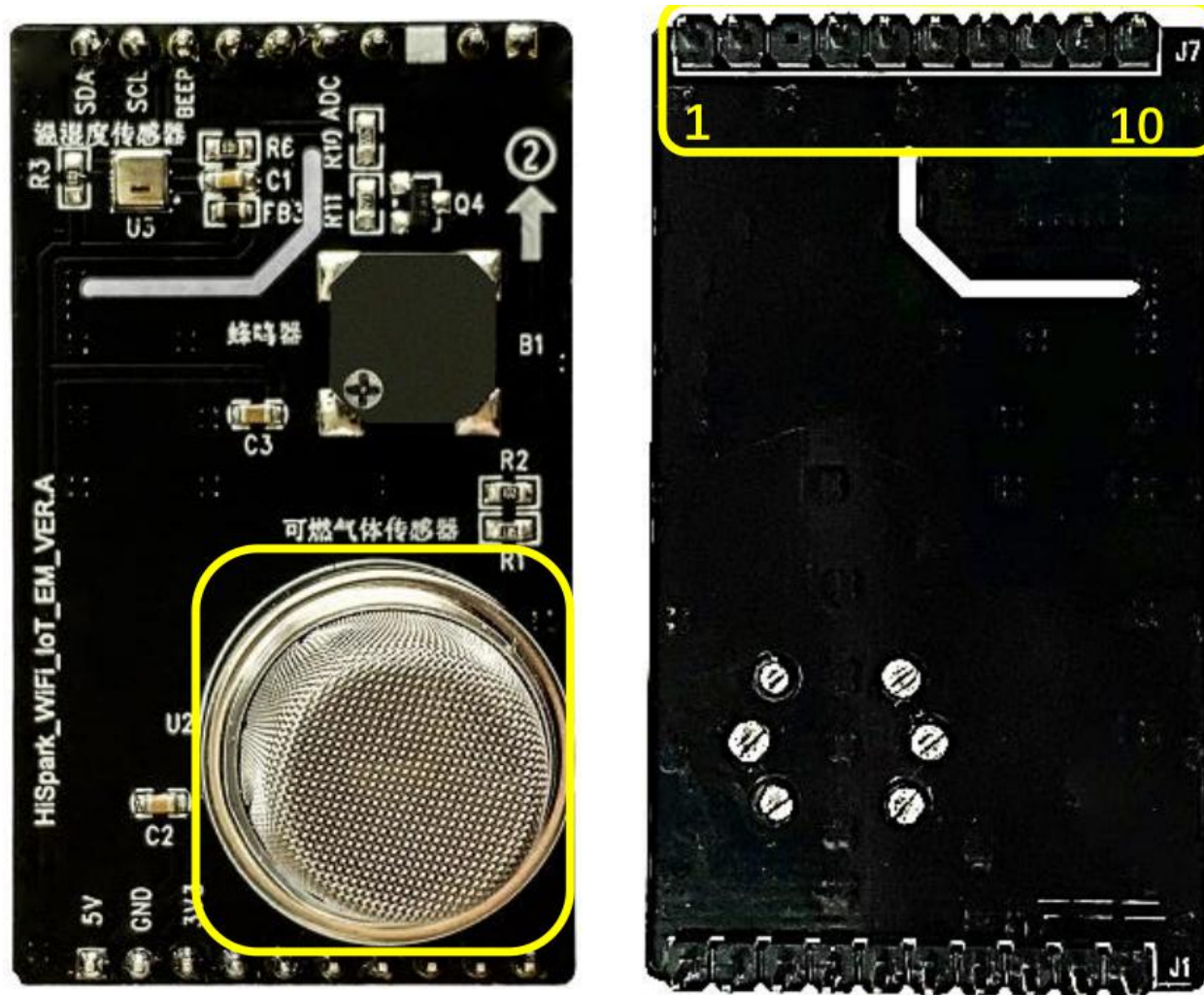
## 知识点1 【板载器件介绍】

- 型号：半导体可燃气体传感器 MQ-2
- 产品特点：
  1. MQ-2 气体传感器所使用的气敏材料是在清洁空气中导电率较低的二氧化锡 ( $\text{SnO}_2$ )。当传感器所处环境中存在可燃气体时，传感器的电导率随空气中可燃气体浓度的增加而增大。
  2. 使用简单的电路即可将电导率的变化转化为与该气体浓度相对应的输出信号。
  3. MQ-2 气体传感器对丙烷、烟雾的灵敏度高，对天然气和其他可燃蒸汽的检测也很理想
- 检测气体：可燃气体、烟雾
- 检测浓度：300-10000ppm(1ppm=1立方厘米/1立方米)
- 控制方式：本传感器通过 J7 接口 ADC 引脚上报检测结果

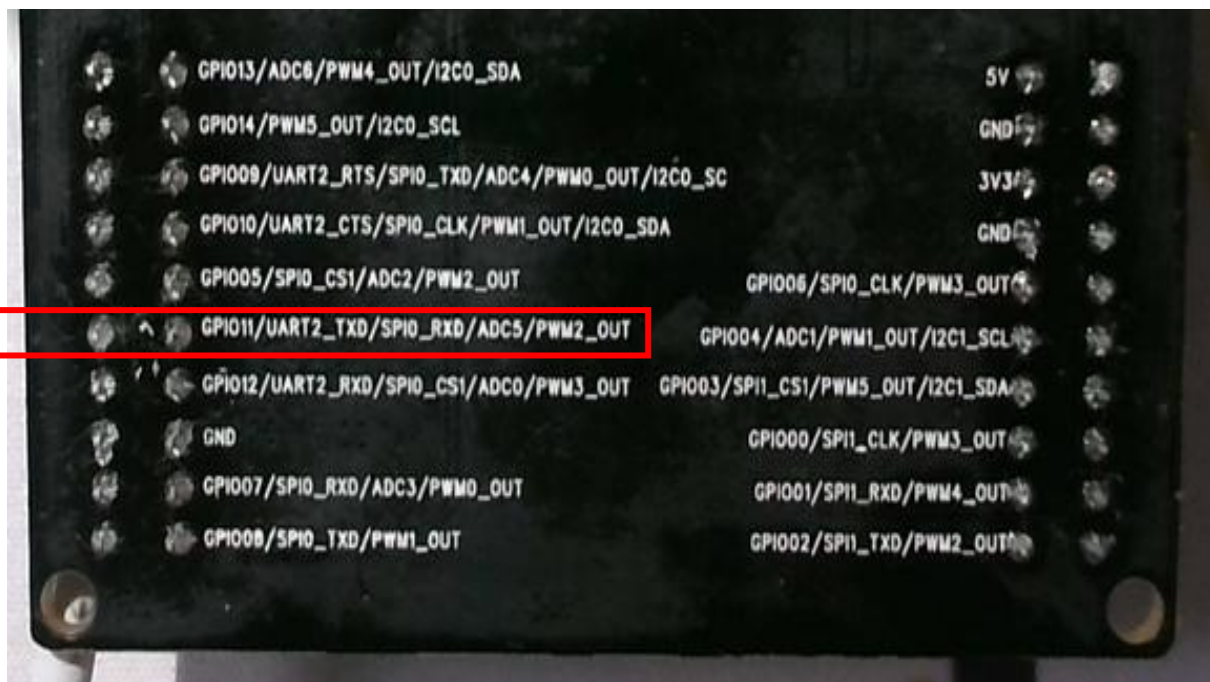




## 知识点2 【电路图分析】



- Hi3861 通过 J7 接口ADC引脚访问该传感器



- ## ■ ADC5与ADC相连

## 知识点3 【代码实现可燃气体浓度值读取】

实现步骤:

1. 创建mq2\_demo.c, 加入头文件
2. 完成可燃气体浓度值读取功能
3. 注册自启动线程

## 知识点3 【代码实现可燃气体浓度值读取】

步骤1：创建mq2\_demo.c，加入头文件

**<code/>**

```
#include <stdio.h>
#include <stdint.h>
#include <string.h>
#include <unistd.h>
```

```
#include "ohos_init.h"
#include "cmsis_os2.h"
#include "wifiiot_gpio.h"
#include "wifiiot_gpio_ex.h"
#include "wifiiot_adc.h"
#include "wifiiot_errno.h"
```

```
#define GAS_SENSOR_CHAN_NAME WIFI_IOT_ADC_CHANNEL_5
```



## 知识点3 【代码实现可燃气体浓度值读取】

步骤2：完成可燃气体浓度值读取功能

<code/>

```
static void Mq2DemoTask(void *arg)
{
    (void)arg;
    GpioInit();
    while (1)
    {
        unsigned short data = 0;
        if (AdcRead(GAS_SENSOR_CHAN_NAME, &data, WIFI_IOT_ADC_EQU_MODEL_4, WIFI_IOT_ADC_CUR_BAIS_DEFAULT, 0) ==
WIFI_IOT_SUCCESS)
        {
            printf("%d\n", data);
        }

        sleep(1);
    }
}
```

## 知识点3 【代码实现可燃气体浓度值读取】

### 步骤3：注册自启动线程

<code/>

```
static void Mq2Demo(void)
{
    osThreadAttr_t attr;
    attr.name = "Mq2DemoTask";
    attr.attr_bits = 0U;
    attr.cb_mem = NULL;
    attr.cb_size = 0U;
    attr.stack_mem = NULL;
    attr.stack_size = 4096;
    attr.priority = osPriorityNormal;
    if (osThreadNew(Mq2DemoTask, NULL, &attr) == NULL)
    {
        printf("[EnvironmentDemo] Falied to create EnvironmentTask!\n");
    }
}

APP_FEATURE_INIT(Mq2Demo);
```



## 知识点3 【代码实现可燃气体浓度值读取】

./applications/sample/wifi-iot/app/chapter\_04/BUILD.gn文件

**<code/>**

```
static_library("sensing_demo") {
    sources = [
        # "aht20_demo.c", "aht20.c",
        "mq2_demo.c"
    ]

    include_dirs = [
        "../utils/native/lite/include",
        "../kernel/liteos_m/components/cmsis/2.0",
        "../base/iot_hardware/interfaces/kits/wifi_iot_lite",
    ]
}
```



## 知识点3 【代码实现可燃气体浓度值读取】

./applications/sample/wifi-iot/app/BUILD.gn文件

**<code/>**

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

lite_component("app") {
  features = [
    # "chapter_03:chapter_03_demo",
    "chapter_04:sensing_demo"
  ]
}
```



## 知识点3 【代码实现可燃气体浓度值读取】

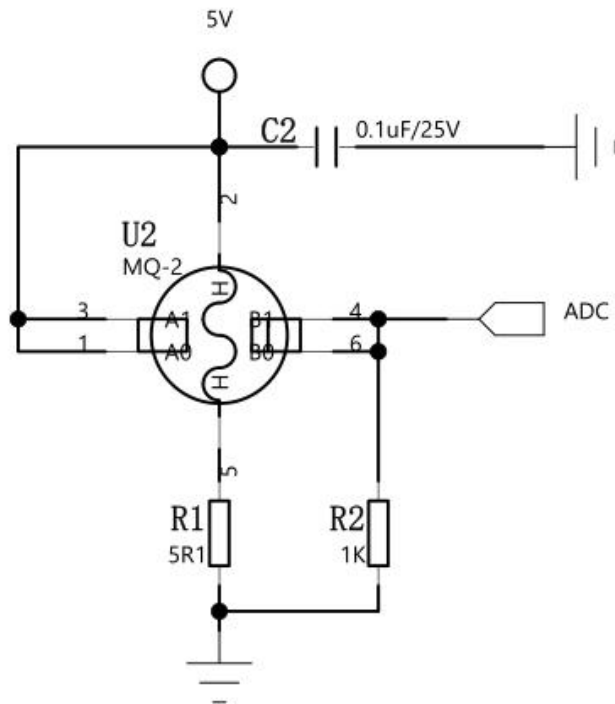
操作演示



## 本节小结

本讲所学知识点有：

- 知识点1：板载器件介绍
- 知识点2：电路图分析
- 知识点3：代码实现可燃气体浓度值读取





## 第3节：OLED显示数据

- 知识点1：板载器件介绍
- 知识点2：电路图分析
- 知识点3：代码实现信息显示

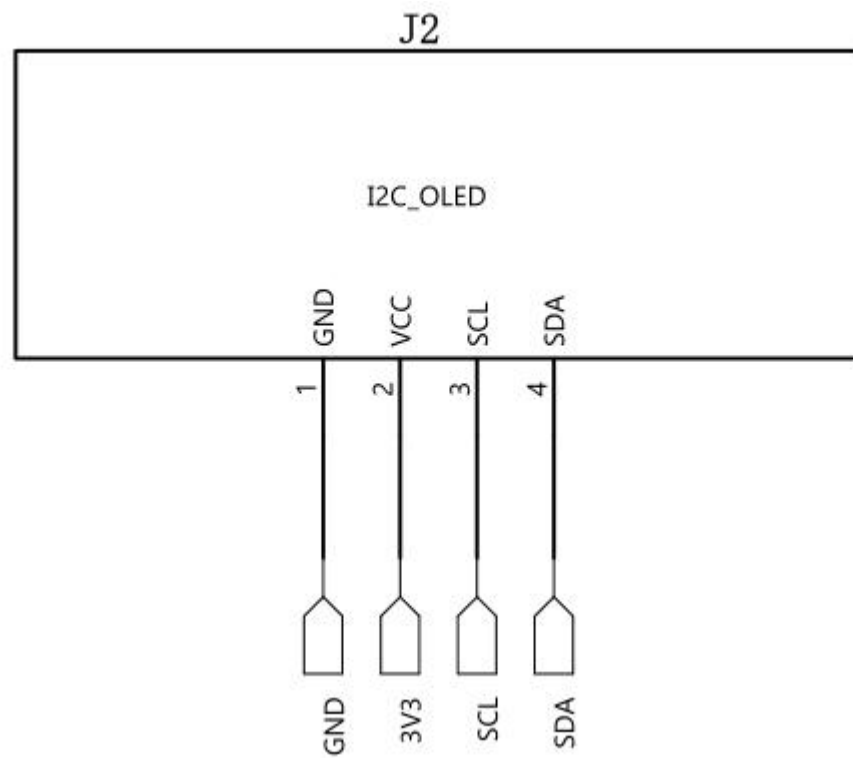
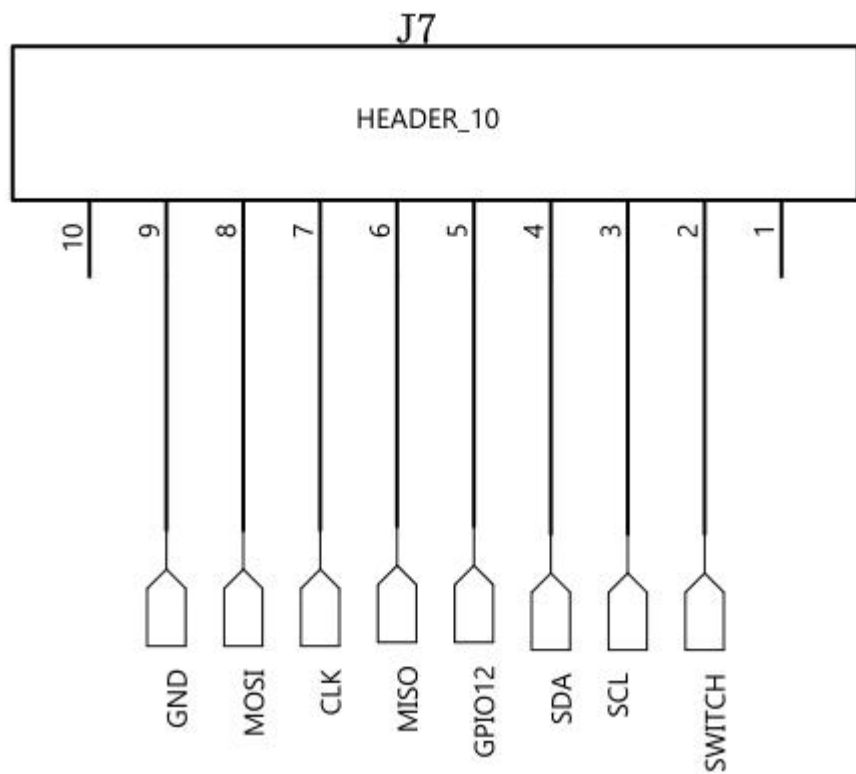


## 知识点1 【板载器件介绍】

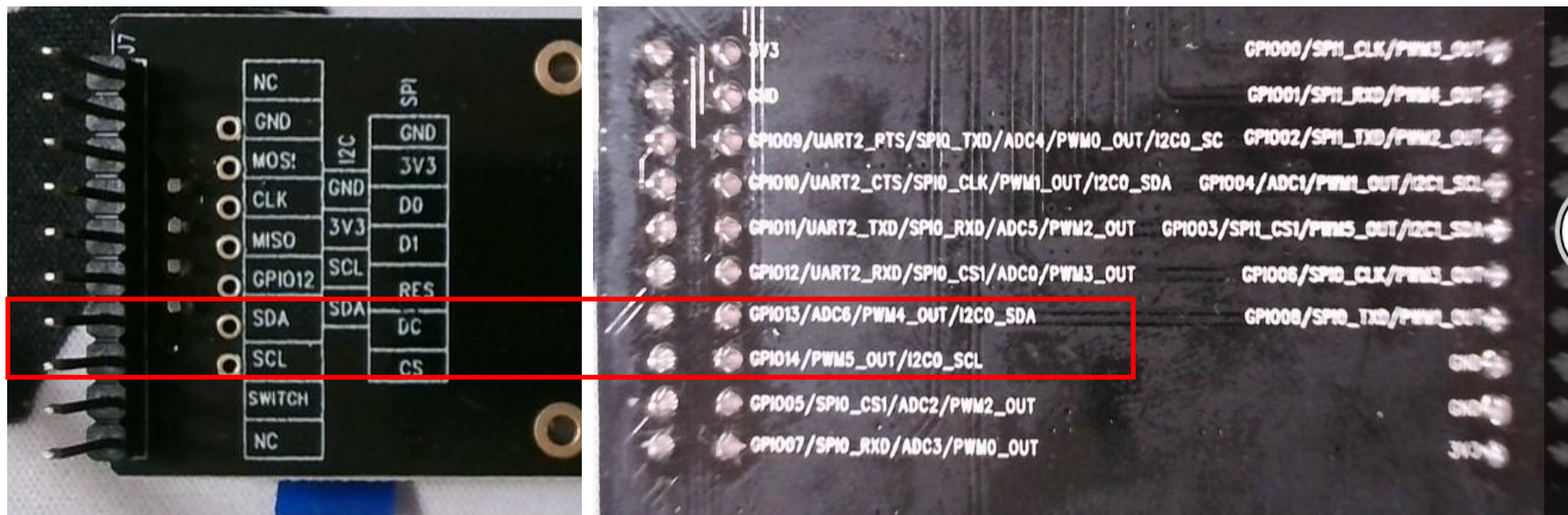
- 型号：JMD0.96 寸 4pin GND 显示屏模块
- 产品特点：
  1. 高分辨率：128\*64
  2. 广可视角度：大于 160°
  3. 低功耗：正常显示时 0.06W
  4. 宽供电范围：直流 3.3V-5V
  5. 通信方式：IIC
  6. 亮度、对比度可以通过程序指令控制
  7. 内部驱动芯片：SSD1306
- 控制方式：Hi3861 通过 J7 接口 SCL 引脚、SDA 引脚以 IIC 接口方式控制该 OLED 模块



## 知识点2 【电路图分析】



## 知识点2 【电路图分析】



- SDA复用GPIO13
- SCL复用GPIO14

## 知识点3 【代码实现信息显示】

实现步骤:

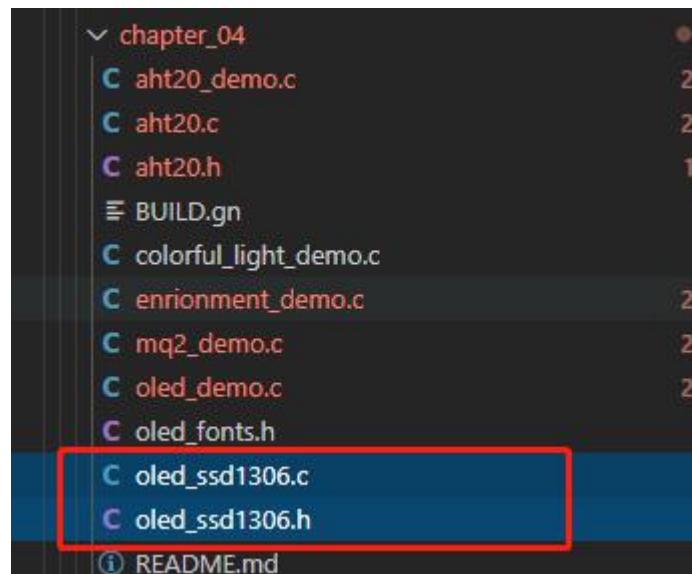
1. 导入驱动文件oled\_ssd1306.c、oled\_ssd1306.h
2. 创建oled\_demo.c, 创建初始化oled函数
3. 完成显示数据功能
4. 注册为自启动线程
5. 配置工程模块





## 知识点3 【代码实现信息显示】

步骤1：导入驱动文件oled\_ssd1306.c、oled\_ssd1306.h



## 知识点3 【代码实现信息显示】

步骤2: 创建oled\_demo.c, 创建初始化oled函数

**<code/>**

```
#include <stdio.h>
#include <unistd.h>
#include "ohos_init.h"
#include "cmsis_os2.h"
#include "wifiiot_gpio.h"
#include "wifiiot_gpio_ex.h"
#include "oled_ssd1306.h"
```

```
void init(void)
{
    GpioInit();
    OledInit();
}
```





## 知识点3 【代码实现信息显示】

步骤3：完成显示数据功能

**<code/>**

```
static void OledTask(void *arg){
    (void)arg;
    init();
    OledFillScreen(0x00);
    OledShowString(0, 0, "Hello, HarmonyOS", 1);
    sleep(1);
    for (int i = 0; i < 3; i++) {
        OledFillScreen(0x00);
        for (int y = 0; y < 8; y++){
            static const char text[] = "ABCDEFGHJKLMNOP"; // QRSTUVWXYZ
            OledShowString(0, y, text, 1);
        }
        sleep(1);
    }
}
```

## 知识点3 【代码实现信息显示】

步骤4：注册为自启动线程

**<code/>**

```
static void OledDemo(void)
{
    osThreadAttr_t attr;
    attr.name = "OledTask";
    attr.attr_bits = 0U;
    attr.cb_mem = NULL;
    attr.cb_size = 0U;
    attr.stack_mem = NULL;
    attr.stack_size = 4096;
    attr.priority = osPriorityNormal;
    if (osThreadNew(OledTask, NULL, &attr) == NULL)
    {
        printf("[OledDemo] Failed to create OledTask!\n");
    }
}
APP_FEATURE_INIT(OledDemo);}
```

## 知识点3 【代码实现信息显示】

### 步骤5：配置工程模块

➤ ./applications/sample/wifi-iot/app/chapter\_04/BUILD.gn文件

```
static_library("sensing_demo") {  
    sources = [  
        # "aht20_demo.c","aht20.c",  
        # "mq2_demo.c"  
        #"colorful_light_demo.c",  
        "oled_demo.c", "oled_ssd1306.c",  
    ]  
  
    include_dirs = [  
        "../utils/native/lite/include",  
        "../kernel/liteos_m/components/cmsis/2.0",  
        "../base/iot_hardware/interfaces/kits/wifiot_lite",  
    ]  
}
```



## 知识点3 【代码实现信息显示】

操作演示



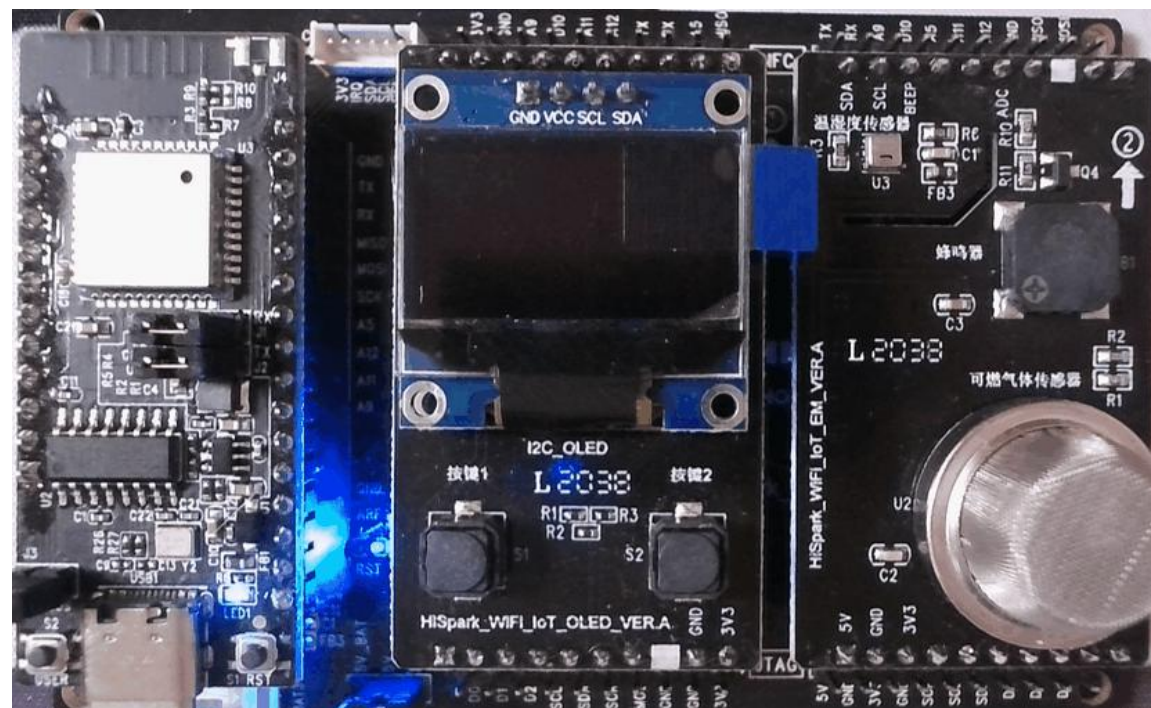
## 本节小结

本讲所学知识点有：

- 知识点1：板载器件介绍
- 知识点2：电路图分析
- 知识点3：代码实现信息显示

## 第4节：综合案例讲解

- 知识点1：案例源码
- 知识点2：案例源码分析
- 知识点3：案例效果演示



## 知识点1 【案例源码】

./applications/sample/wifi-iot/app/chapter\_04/enrionment\_demo.c

**<code/>**

```
#include <stdio.h>
#include <stdint.h>
#include <string.h>
#include <unistd.h>

#include "ohos_init.h"
#include "cmsis_os2.h"
#include "wifiiot_i2c.h"
#include "wifiiot_gpio.h"
#include "wifiiot_gpio_ex.h"
#include "wifiiot_adc.h"
#include "wifiiot_errno.h"
#include "aht20.h"
#include "oled_ssd1306.h"
```





## 知识点1 【案例源码】

./applications/sample/wifi-iot/app/chapter\_04/enrionment\_demo.c

<code/>

```
#define GAS_SENSOR_CHAN_NAME WIFI_IOT_ADC_CHANNEL_5
#define AHT20_BAUDRATE 400 * 1000
#define AHT20_I2C_IDX WIFI_IOT_I2C_IDX_0
#define ADC_RESOLUTION 2048
```

```
uint32_t retval = 0;
float humidity = 0.0f;
float temperature = 0.0f;
unsigned short data = 0;
static char line[32] = {0};
```



## 知识点1 【案例源码】

./applications/sample/wifi-iot/app/chapter\_04/enrionment\_demo.c

**<code/>**

```
void init(void)
{
    GpioInit();
    OledInit();
    OledFillScreen(0);
    I2cInit(AHT20_I2C_IDX, AHT20_BAUDRATE);
    while (WIFI_IOT_SUCCESS != AHT20_Calibrate())
    {
        printf("AHT20 sensor init failed!\r\n");
        usleep(1000);
    }
}
```



## 知识点1 【案例源码】

./applications/sample/wifi-iot/app/chapter\_04/enrionment\_demo.c

<code/>

```
static void EnvironmentTask(void *arg)
{
    (void)arg;
    init();
    while (1){
        if (AHT20_StartMeasure() != WIFI_IOT_SUCCESS){
            printf("trigger measure failed!\r\n");
        }
        if (AHT20_GetMeasureResult(&temperature, &humidity) != WIFI_IOT_SUCCESS){
            printf("get humidity data failed!\r\n");
        }
        AdcRead(GAS_SENSOR_CHAN_NAME, &data, WIFI_IOT_ADC_EQU_MODEL_4, WIFI_IOT_ADC_CUR_BAIS_DEFAULT, 0);
        OledShowString(0, 0, "Sensor values:", 1);
        snprintf(line, sizeof(line), "temp: %.2f", temperature);
        OledShowString(0, 1, line, 1);
        snprintf(line, sizeof(line), "humi: %.2f", humidity);
        OledShowString(0, 2, line, 1);
        snprintf(line, sizeof(line), "gas: %d", data);
        OledShowString(0, 3, line, 1);
        sleep(1);
    }
}
```

## 知识点1 【案例源码】

./applications/sample/wifi-iot/app/chapter\_04/enrionment\_demo.c

<code/>

```
static void EnvironmentDemo(void)
{
    osThreadAttr_t attr;
    attr.name = "EnvironmentTask";
    attr.attr_bits = 0U;
    attr.cb_mem = NULL;
    attr.cb_size = 0U;
    attr.stack_mem = NULL;
    attr.stack_size = 4096;
    attr.priority = osPriorityNormal;

    if (osThreadNew(EnvironmentTask, NULL, &attr) == NULL)
    {
        printf("[EnvironmentDemo] Falied to create EnvironmentTask!\n");
    }
}

APP_FEATURE_INIT(EnvironmentDemo);
```



## 知识点1 【案例源码】

./applications/sample/wifi-iot/app/chapter\_04/BUILD.gn文件

**<code/>**

```
static_library("sensing_demo") {  
    sources = [  
        # "aht20_demo.c", "aht20.c",  
        # "mq2_demo.c"  
        # "oled_demo.c", "oled_ssd1306.c",  
        "environment_demo.c", "oled_ssd1306.c", "aht20.c"  
    ]  
  
    include_dirs = [  
        "//utils/native/lite/include",  
        "//kernel/liteos_m/components/cmsis/2.0",  
        "//base/iot_hardware/interfaces/kits/wifiot_lite",  
    ]  
}
```



## 知识点1 【案例源码】

./applications/sample/wifi-iot/app/BUILD.gn文件

**<code/>**

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

lite_component("app") {
  features = [
    # "chapter_03:chapter_03_demo",
    "chapter_04:sensing_demo"
  ]
}
```





## 知识点2 【案例源码分析】

代码分析流程：

1. APP\_FEATURE\_INIT(EnvironmentDemo);
2. static void EnvironmentDemo(void)
3. static void EnvironmentTask(void \*arg)

## 知识点2 【案例源码分析】

源码分析演示



## 知识点3 【案例效果演示】

操作演示



## 本节小结

本讲所学知识点有：

- 知识点1：案例源码
- 知识点2：案例源码分析
- 知识点3：案例效果演示

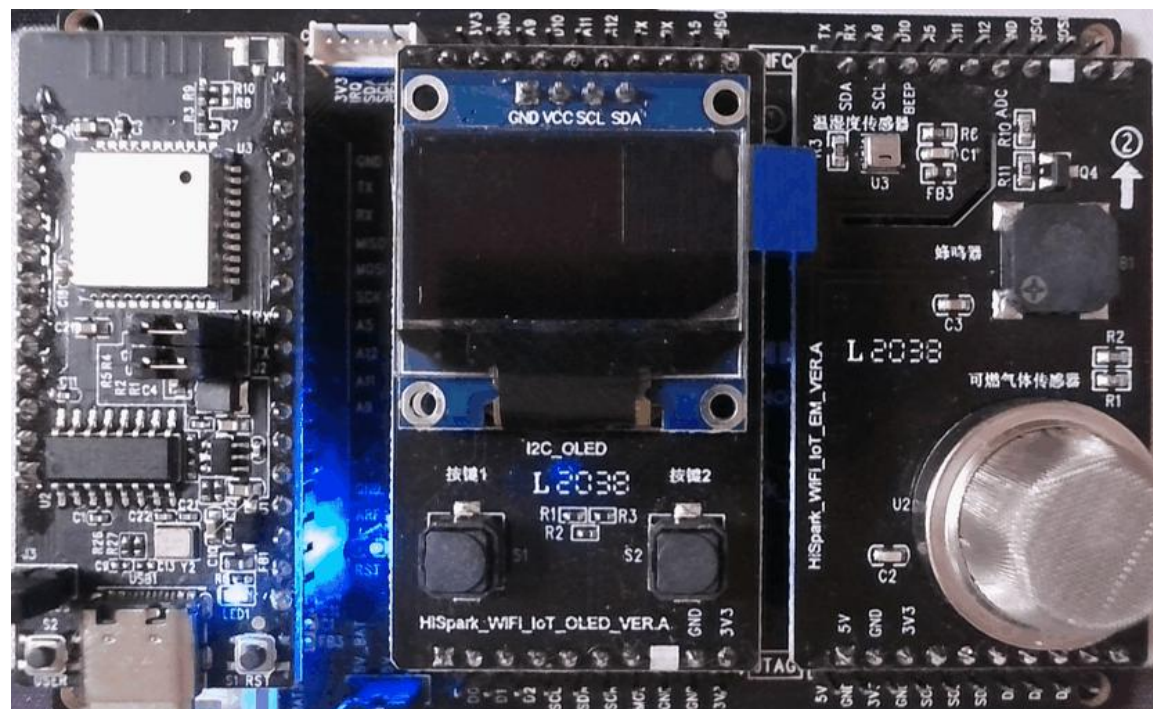
# 本讲总结

本章所学内容有：

- 第1节：AHT20温湿度传感器数据读取
- 第2节：MQ-2燃气传感器数据读取
- 第3节：OLED显示数据
- 第4节：综合案例讲解

# 任务挑战

挑战任务：在综合案例中添加蜂鸣器操作，每次数据更新，蜂鸣器响一次



# THANKS

更多学习视频，关注宅客学院.....

