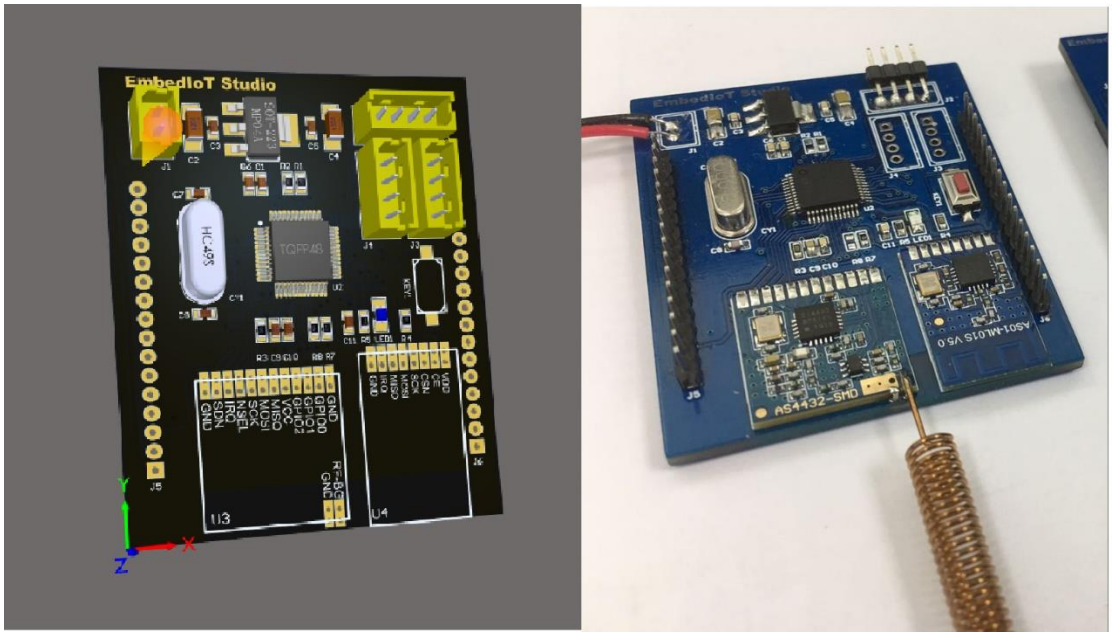
嵌入式物联网应用开发 – 无线收发模块

上一篇文章讲述了 ARM-Linux 中控扩展模块的开发设计，上一篇文章的具体内容，请参考以下链接：

开源 | 嵌入式物联网项目开发 - ARM 中控扩展模块

为了提高硬件模块的可重用性，对于终端设备模块的硬件，采用了无线收发模块+传感执行模块的方式进行设计。即一个无线收发模块可以跟不同的传感器模块或执行器模块配合使用，只需要烧录不同的设备端固件即可。无线收发模块如下图所示。

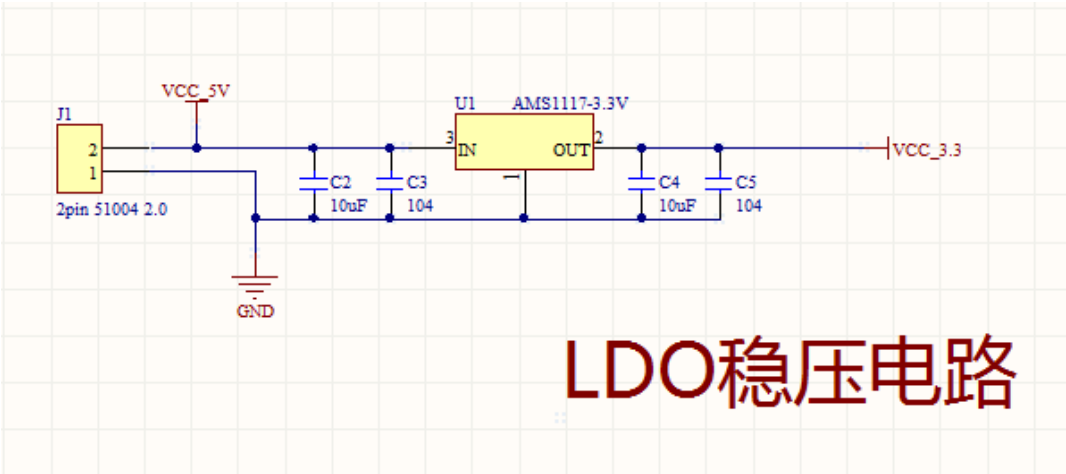


无线收发模块具有以下功能：

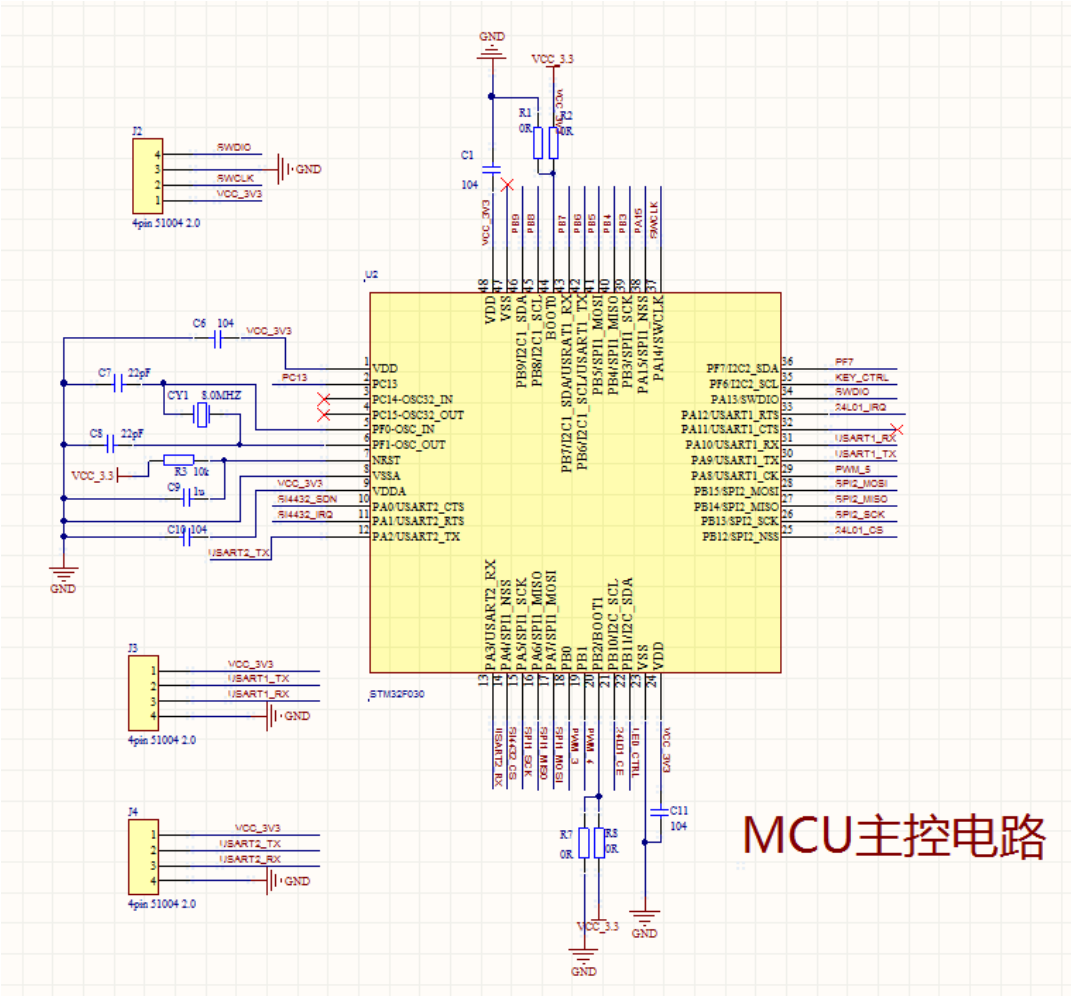
1、外部 DC-5V 电源供电。
2、具有 2.4GHz 和 433MHz 无线收发功能。
3、板载 LED 指示灯和物理按键，用来指示网络状态和重置模块。
4、网络管理功能：加入网络，退出网络，心态包维持，等等。
5、对外引出 MCU 的串口 1 和串口 2，方便外接其他串口模块。
6、对外引出 MCU 的 PWM，IIC，GPIO，外部中断接口，方便用户扩展。
7、模块尺寸：51mm * 42.5mm

硬件设计篇

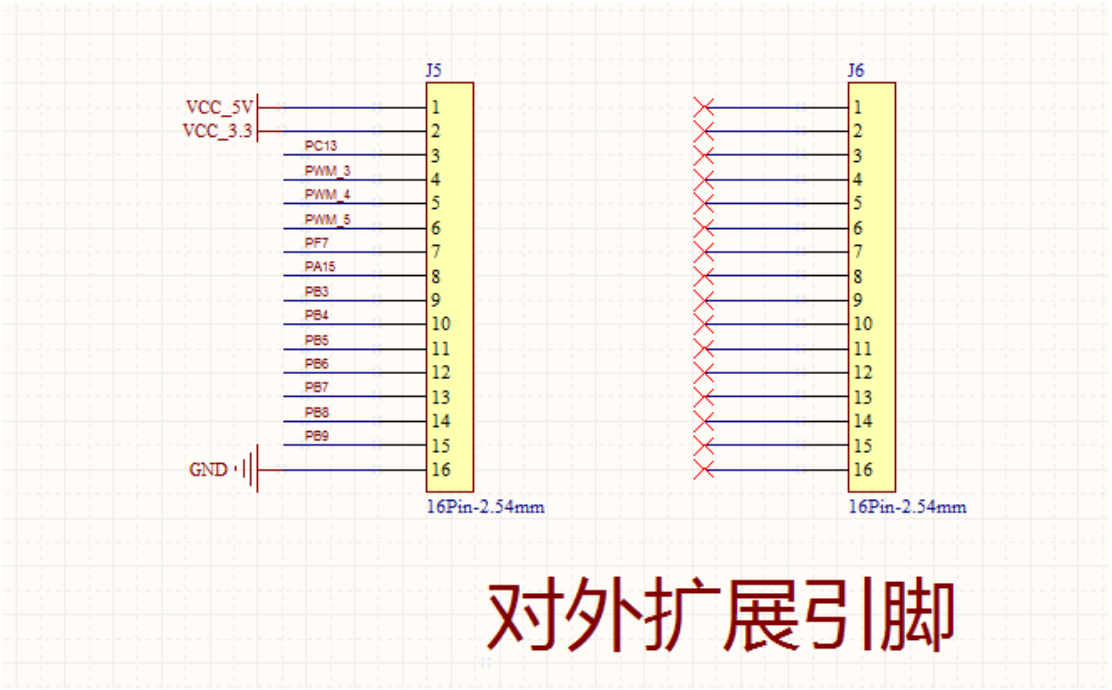
无线收发模块使用 DC-5V 进行供电，模块自带 LDO 稳压芯片，DC-5V 电源通过 LDO 稳压电路，为单片机及其他模块提供 3.3V 电源，LDO 稳压电路如下图所示。



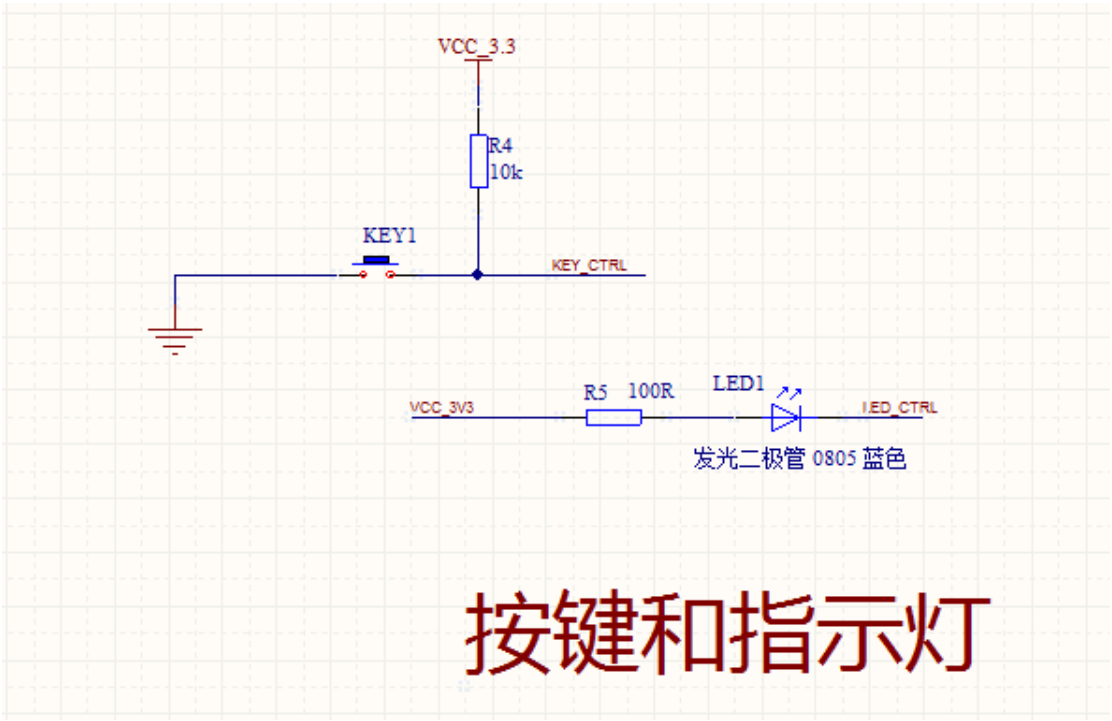
无线收发模块采用 STM32F030C8T6 作为主控芯片，这款芯片是 Cortex-M0 内核，主频 48MHz，芯片自带 64KB 可编程 Flash 和 8KB RAM，外设资源丰富，MCU 控制电路如下图所示。



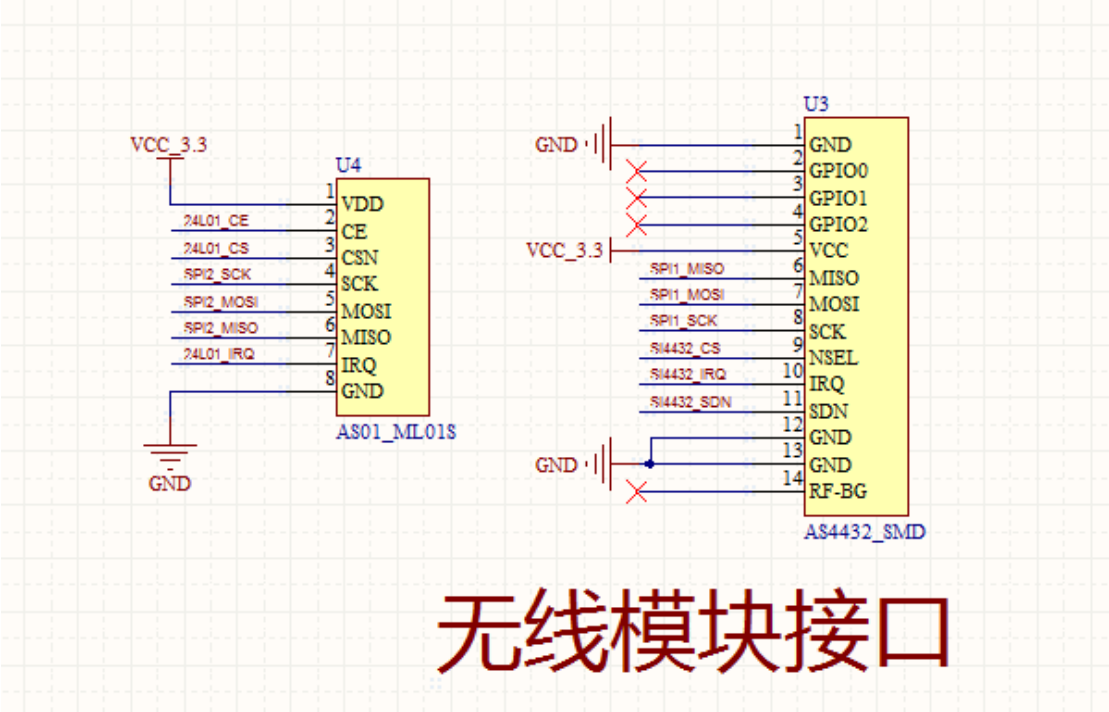
无线收发模块使用排母端子的方式对外引出 MCU3 路 PWM 引脚，一组 IIC 引脚，多组外部中断引脚和通用 GPIO 引脚。其中一边的排母端子作为固定模块使用，并没有芯片引脚连接，对外扩展引脚原理图如下图所示。



无线收发模块带有一个蓝色 LED 指示灯和物理按键，蓝牙 LED 指示灯用来指示网络状态，如入网状态，掉线状态，连接状态，等等。物理按键用来重置模块的出厂配置信息，并使模块重新回到配网状态，电路原理图如下图所示。



无线收发模块使用了 MCU 的 SPI1 和 SPI2 接口跟两款无线芯片进行通信，其中，SPI1 接口连接 SI4432 这款 433MHz 无线模块，SPI2 接口连接 nRF24L01 这款 2.4GHz 无线模块，SI4432 模块和 nRF24L01 均使用焊接方式固定在无线收发模块上面，接口原理图如下所示。



以上就是 ARM 中控扩展模块的原理图描述，各个模块的原理比较简单，具体原理图和 PCB 的硬件工程可以到 GitHub 或 Gitee 上下载。

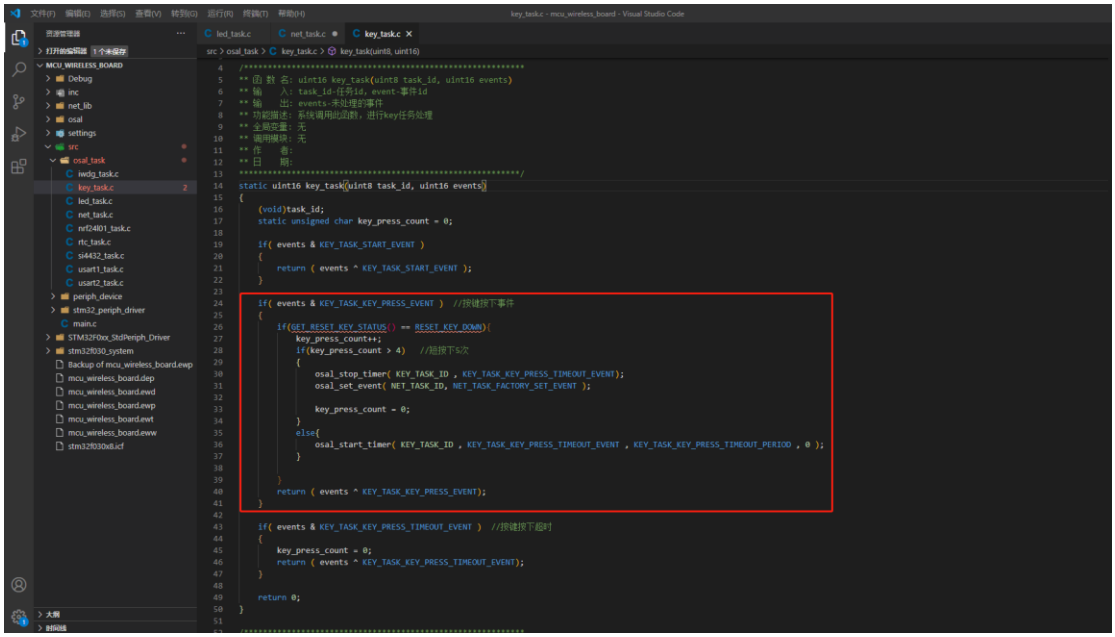
软件设计篇

无线收发模块的软件框架，是基于 OSAL 调度器进行设计的，关于 OSAL 调度器的内容，请参考以下文章。

开源 | 嵌入式物联网项目开发 - 基于任务和事件的 OSAL 调度器

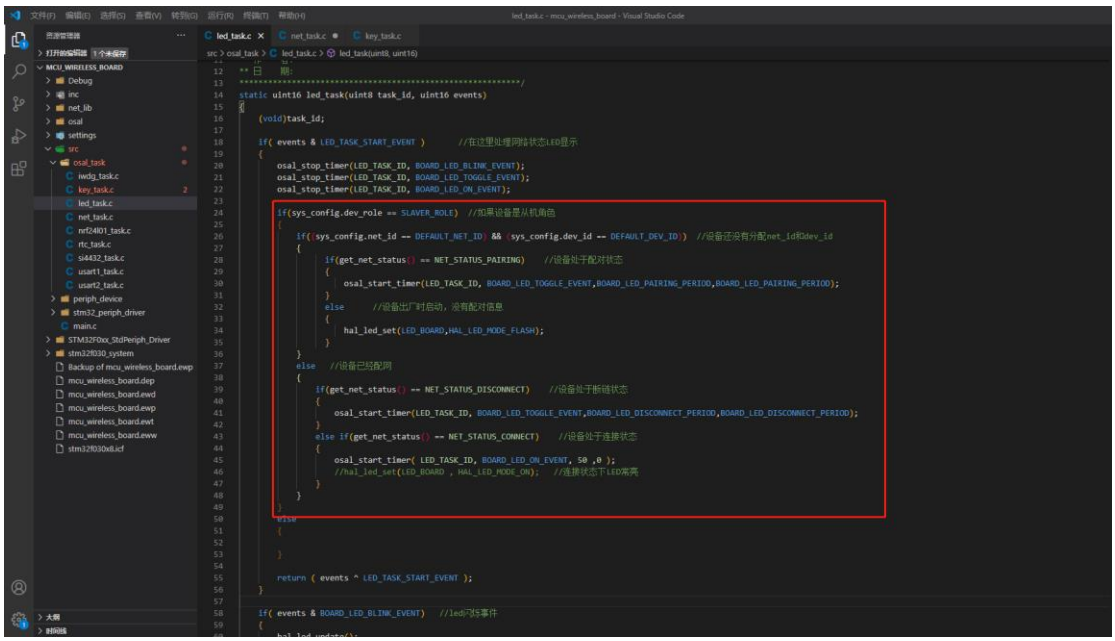
为了让无线收发模块有更好的扩展性，模块并没有添加其他关于传感器和执行器的功能，只保留了最基本的无线收发功能和网络管理功能，例如恢复出厂设置，加入网络，退出网络，心跳包维持，等等。

无线收发模块可以通过板载物理按键重置自身的网络信息，用户连续按下板子上的物理按键 5 次，无线收发模块会把当前的网络配置信息恢复为出厂设置。物理按键的操作函数在 key_task.c 文件中实现，如下图所示。

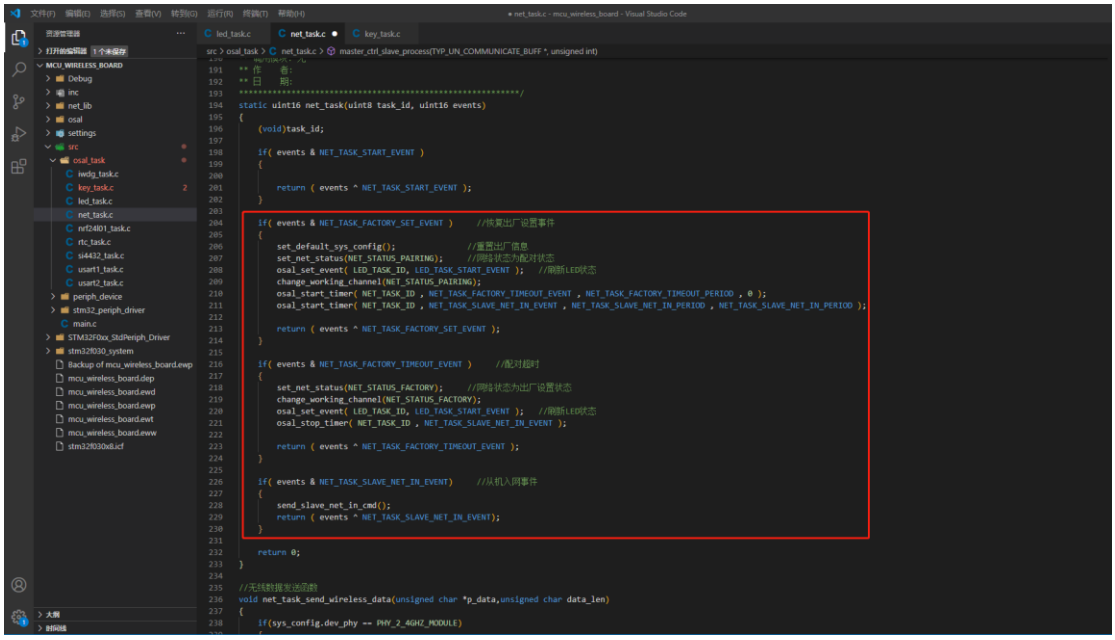


```c
/*********************************************************
** 功 能：uint16 key_task(uint8 task_id, uint16 events)
** 输    入：task_id-任务id，event-事件id
** 输    出：events-未处理的事件
** 功能描述：系统调用此函数，进行key任务处理
** 全局变量：无
** 调用模块：无
** 作    者：
** 日    期：
*********************************************************/
static uint16 key_task(uint8 task_id, uint16 events)
{
    (void)task_id;
    static unsigned char key_press_count = 0;

    if( events & KEY_TASK_START_EVENT )
    {
        return ( events ^ KEY_TASK_START_EVENT );
    }

    if( events & KEY_TASK_KEY_PRESS_EVENT )  //按键按下事件
    {
        if(GET_RESET_KEY_STATUS() == RESET_KEY_DOWN){
            key_press_count++;
            if(key_press_count > 4)   //连按下5次
            {
                osal_stop_timer( KEY_TASK_ID , KEY_TASK_KEY_PRESS_TIMEOUT_EVENT);
                osal_set_event( NET_TASK_ID, NET_TASK_FACTORY_SET_EVENT );

                key_press_count = 0;
            }
            else{
                osal_start_timer( KEY_TASK_ID , KEY_TASK_KEY_PRESS_TIMEOUT_EVENT , KEY_TASK_KEY_PRESS_TIMEOUT_PERIOD , 0 );
            }

        }
        return ( events ^ KEY_TASK_KEY_PRESS_EVENT);
    }

    if( events & KEY_TASK_KEY_PRESS_TIMEOUT_EVENT )  //按键按下超时
    {
        key_press_count = 0;
        return ( events ^ KEY_TASK_KEY_PRESS_TIMEOUT_EVENT);
    }

    return 0;
}
```

为了更好地辨别出无线收发模块当前的网络状态（如配对状态，连接状态，断连状态），模块通过板子上的一颗蓝色 LED 进行显示，当网络状态发生改变时，LED 的闪烁状态也会发生改变，在 led_task.c 文件中实现 LED 状态显示，如下图所示。



无线收发模块的网络管理功能，在 net_task.c 文件中实现，如入网操作，退出网络，心跳包维持，等等。无线收发模块通过入网操作，向主机请求网络 ID 和设备 ID，当入网成功后，主机会通过心跳包的方式维持整个网络，如下图所示。



无线收发模块在入网的时候，需要向主机（即 ARM 中控扩展模块）请求网络信息，而主机的网络信息需要在 ARM 中控上对其预先进行配置，因此在 ARM 中控上设计了一个主机参数配置界面，用于对主机参数进行配置，如下图所示。

如图所示，点击【设置主机】按钮，弹出设置主机参数界面，可以使用该界面进行参数获取和参数设置，设置完成后，点击 OK 退出界面。【主机扫描】按钮是在从机入网时使用的，点击该按钮，主机会启动扫描，进入待入网状态，等待接收从机的入网请求。

系统软件之间的通信协议和 AT 指令集，请参考如下两篇文档：《0-ARM 中控扩展模块 AT 指令集.docx》《1-嵌入式物联网系统通信协议.docx》

项目的开源地址：
https://github.com/embediot/Embedded-IoT-Project
https://gitee.com/embediot/Embedded-IoT-Project

感谢阅读！

点击这里，访问作者博客

欢迎关注【微联智控工作室】


EmbedIoT