**Hekr-SDK 使用示例**

**以无级调光灯项目为案例**

**准备工作**

* 嵌入式开发环境
* 源码获取
* ESP wifi模块
* 无级调光控制板
* Hekr APP
* 家用路由器

**开发步骤**

* 使用SDK编译烧录
* 测试SDK工作正常
* 编写无级调光灯控制程序
* 与ESP wifi模块对接
* APP配置模块上网
* 测试APP无级调光

**问题反馈**

**版本修订**

1. **准备工作**

* 嵌入式开发环境：

安装 STVD 开发软件 编译器 PC串口驱动 串口调试助手

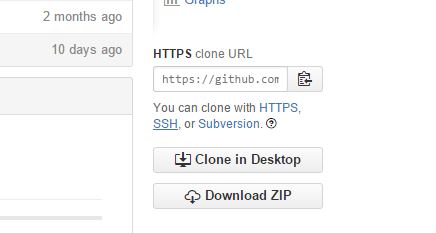
开发工具：USB转TTL驱动 ST-LINK 下载器

* 源码获取

Hekr-MCU-SDK源码获取

使用git：<https://github.com/HEKR-Cloud/HEKR-MCU-SDK.git>

如果没有git 账号可以选择 页面左侧Download ZIP会将源码打包下载下来



* ESP wifi模块
* 无级调光控制板



* Hekr APP

扫描右侧二维码 可以下载Hekr APP

安装对应安卓或者ios的APP

注册登录 APP 获取对应功能

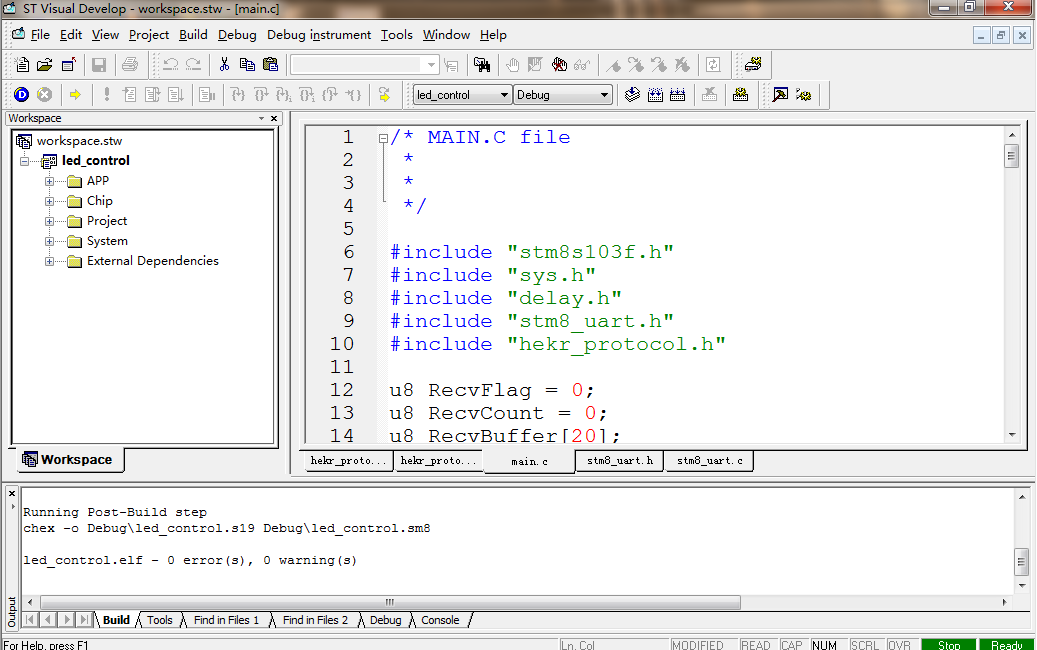
* 家用路由器

需要知道路由器无线的名字(最好是英文SSID)和密码，不能在酒店，机场，餐厅等地方使用，要确保设备登录上网时，不会被弹窗输入账号密码等。

1. **开发步骤**

* 使用SDK编译烧录

用STVD打开源码中的HekrProtocol 工程设置好编译器路径，选择好芯片型号编译入下图，此处出错一般为编译器路径设置或者芯片选择问题，调整对应配置即可通过编译。

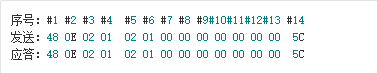


使用ST-LINK把程序烧录进去，测试烧写OK

测试SDK正常使用

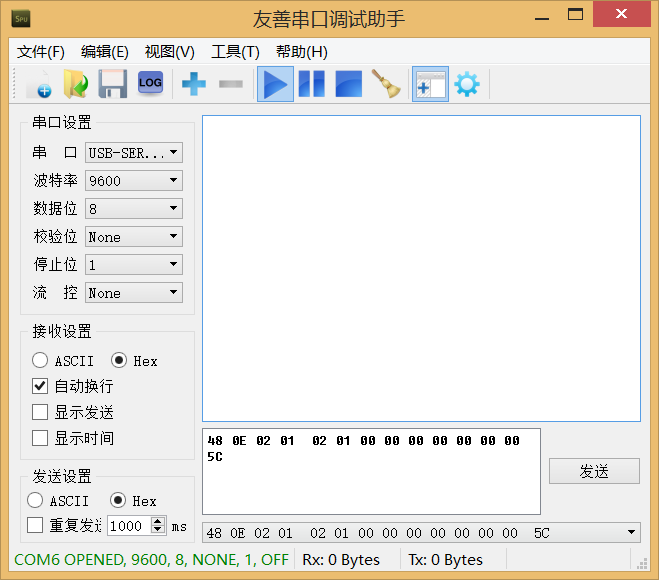
测试协议地址 <http://docs.hekr.me/protocol/>

打开串口调试助手 将PC串口和无级调光板上单片机串口相连 设置好对应参数，打开串口，对SDK进行测试。

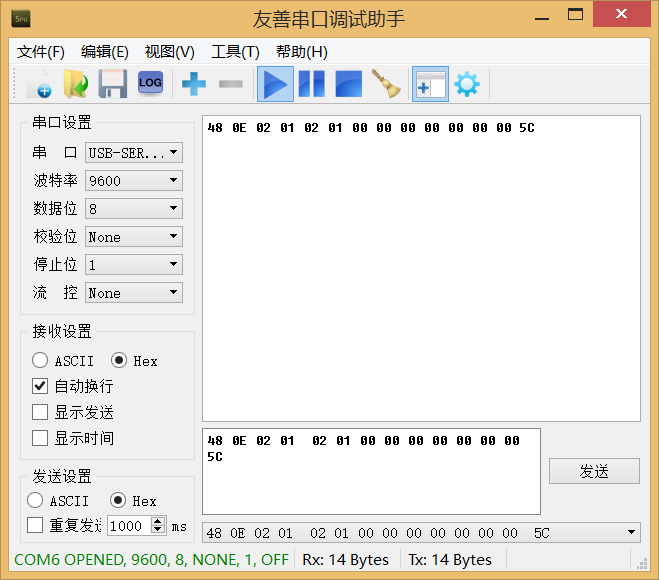


发送测试帧

48 0E 02 01 02 01 00 00 00 00 00 00 00 5C

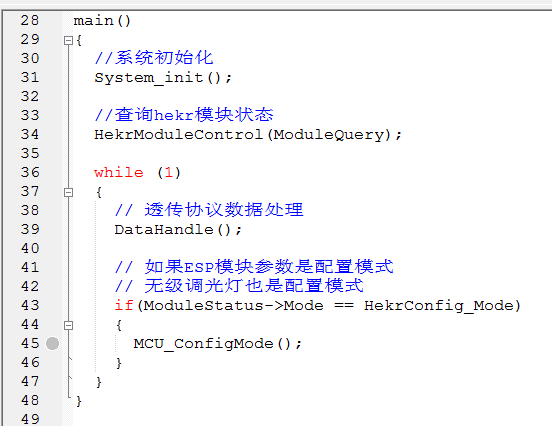


如果得到如下结果表示SDK 工作正常，直接获取的SDK源码除了打出这一帧，还会有把命令帧打出 此帧的命令码是02，即5C后多一个 02.



* 编写无级调光灯控制程序

无级调光灯程序主结构，主函数处理接受的数据，执行对应步骤具体调光由定时器控制



上电调用HEKR API 查询当前wifi模块状态。

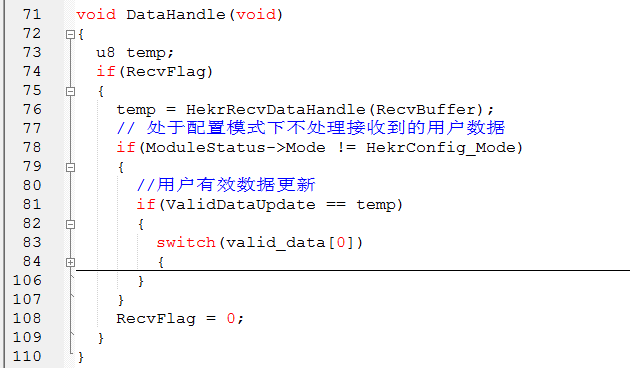
串口接收到数据后由DataHandle(); 函数处理 该函数与SDK里的主函数基本结构一样，相较SDK添加如下的判断语句是，根据无级调光灯需求所加 wifi模块配置时灯也有特定模式，且不接受正常用户命令，直到配置完成。

if(ModuleStatus->Mode != HekrConfig\_Mode)

调用HekrRecvDataHandle(); 函数后因为需求中没有在运行过程中对wifi模块状态的要求 所以不需要处理以下返回值

HekrModuleStateUpdate

返回值为ValidDataUpdate即用户数据发送更新



使用前定义了全局变量UserValidLen 代表我实际需要的帧长度

同时修改SDK中的头文件里的长度宏 为所需长度

此处长度过大会造成数据溢出

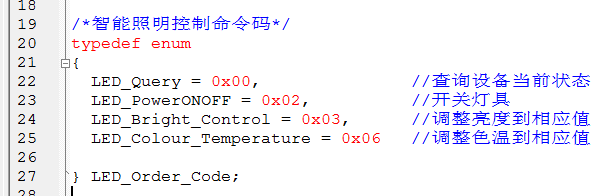
#define USER\_MAX\_LEN 0x20u

分析用户帧处理数据的方法

根据测试协议valid\_data[0]的值为用户命令值

根据协议中的命令码码值 编写函数，配置相关参数执行相关函数

编写对应函数 已经调光控制程序之后即可实现串口控制的无级调光灯的功能





配置模式 设计的需求中对 wifi模块的配置命令是采用开关电源4次发出

设计代码如下



调用HEKR API 发出配置模式试wifi模块进入配置模式

同时无级调光灯会以呼吸状态回应直到配置完成

此时可以使用APP进行配置绑定

* 与ESP wifi模块对接

把无级调光板和ESP wifi模块 串口对接

给ESP模块供电

* APP配置模块上网

首先需要让 ESP wifi模块 进入配置模式 让其能连入路由器上，连续开关无级调光板电源4次，无级调光灯进入呼吸状态，此时打开手机APP，按步骤配置ESP模块使其连入路由器 完成配置。

* 测试APP无级调光