

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

相关链接及反馈

版本修订

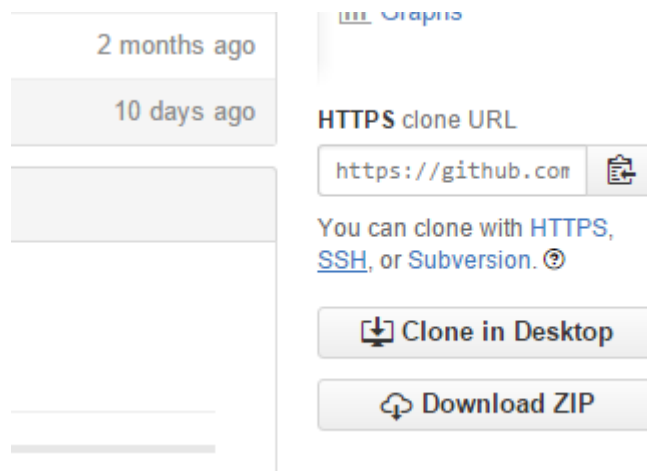
1. 准备工作

- 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



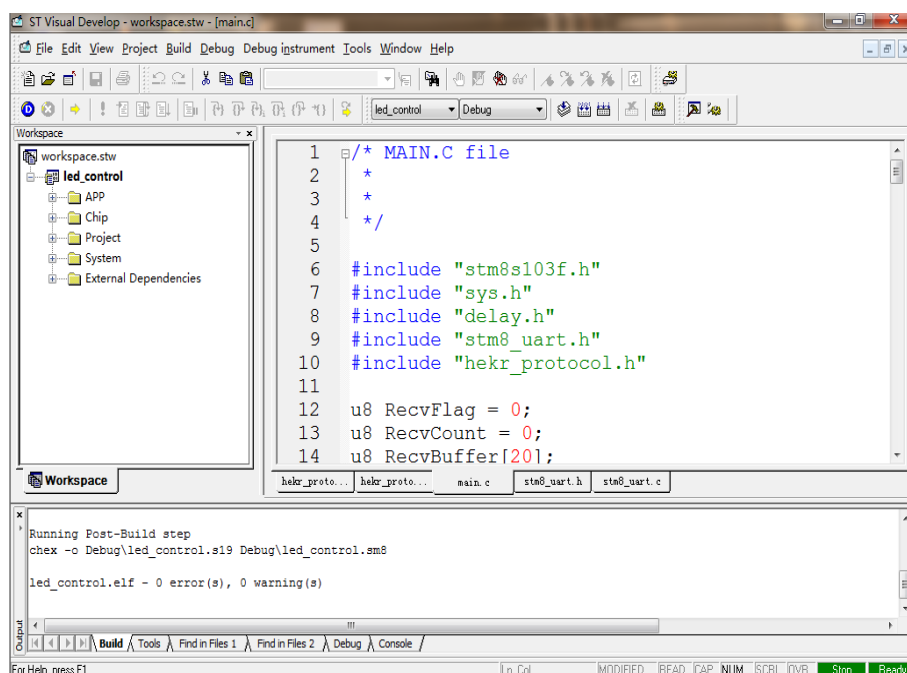
- 家用路由器

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。

2. 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

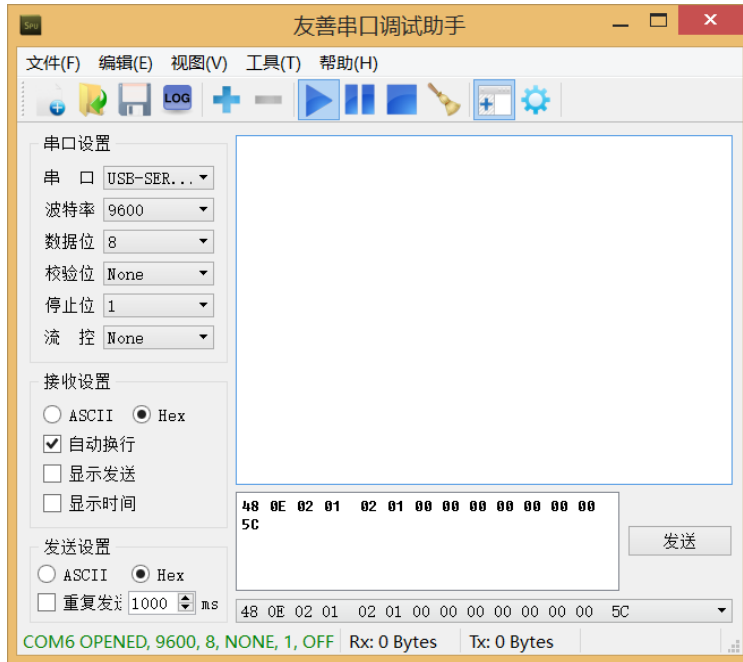
测试协议地址 <http://docs.hekr.me/protocol/>

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。

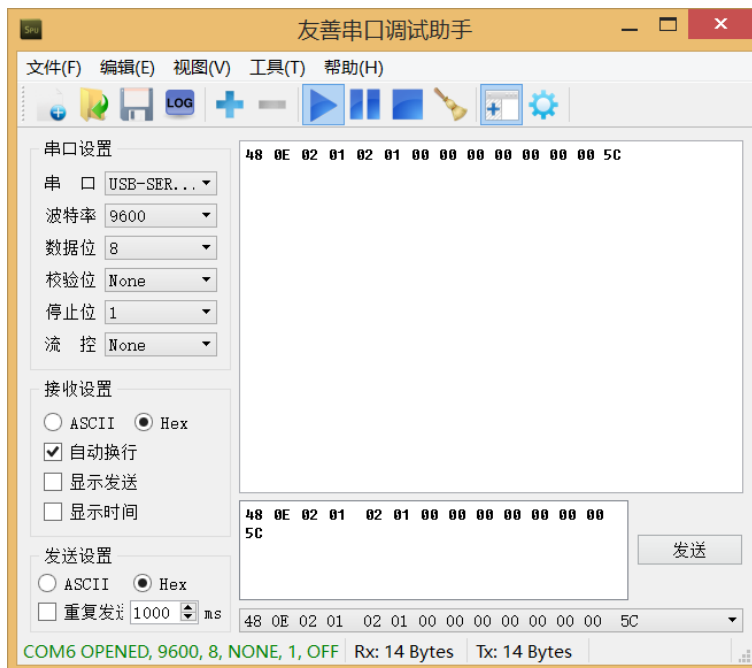
```
序号: #1 #2 #3 #4 #5 #6 #7 #8 #9#10#11#12#13 #14  
发送: 48 0E 02 01 02 01 00 00 00 00 00 00 00 5C  
应答: 48 0E 02 01 02 01 00 00 00 00 00 00 00 5C
```

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

```
28 main()
29 {
30     //系统初始化
31     System_init();
32
33     //查询hekr模块状态
34     HekrModuleControl(ModuleQuery);
35
36     while (1)
37     {
38         // 透传协议数据处理
39         DataHandle();
40
41         // 如果ESP模块参数是配置模式
42         // 无级调光灯也是配置模式
43         if(ModuleStatus->Mode == HekrConfig_Mode)
44         {
45             MCU_ConfigMode();
46         }
47     }
48 }
49
```

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

```
71 void DataHandle(void)
72 {
73     u8 temp;
74     if(RecvFlag)
75     {
76         temp = HekrRecvDataHandle(RecvBuffer);
77         // 处于配置模式下不处理接收到的用户数据
78         if(ModuleStatus->Mode != HekrConfig_Mode)
79         {
80             //用户有效数据更新
81             if(ValidDataUpdate == temp)
82             {
83                 switch(valid_data[0])
84                 {
85                     // ...
86                 }
87             }
88         }
89         RecvFlag = 0;
90     }
91 }
92
```

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

```
18
19  /*智能照明控制命令码*/
20  typedef enum
21  {
22      LED_Query = 0x00,           //查询设备当前状态
23      LED_PowerONOFF = 0x02,      //开关灯具
24      LED_Bright_Control = 0x03,  //调整亮度到相应值
25      LED_Colour_Temperature = 0x06 //调整色温到相应值
26  } LED_Order_Code;
27
28
83
84  switch(valid_data[0])
85  {
86      // 查询无级调光灯状态
87      case LED_Query:
88          //保存当前数据
89          valid_data[1] = led_open_flag;
90          valid_data[3] = bright_set;
91          valid_data[4] = colour_set;
92          //上传用户数据
93          HekrValidDataUpload(UserValidLen);break;
94      // 无级调光灯状态开关控制
95      case LED_PowerONOFF:
96          LED_StateControl(valid_data[1]);break;
97      // 总亮度控制
98      case LED_Bright_Control:
99          bright_set = valid_data[3];
100         if(led_open_flag == 1)UpdateBright();
101         break;
102     // 色温控制
103     case LED_Colour_Temperature:
104         colour_set = valid_data[4];
105         if(led_open_flag == 1)UpdateBright();
106         break;
107     default:break;
108 }
```

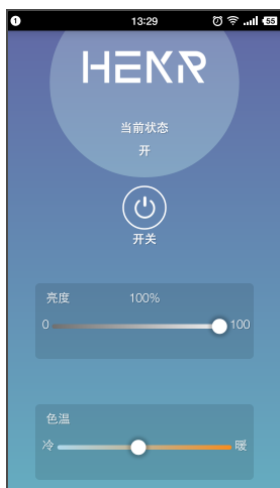
配置模式 设计的需求中对 wifi 模块的配置命令是采用开关电源 4 次发出
设计代码如下

```
33  //模式选择 设定初始值
34  switch(count)
35  {
36      case 1: bright_set = ReadEEPROM(BrightModel1);
37              colour_set = ReadEEPROM(ColourModel1);
38              break;
39      case 2: bright_set = ReadEEPROM(BrightMode2);
40              colour_set = ReadEEPROM(ColourMode2);
41              break;
42      case 3: bright_set = ReadEEPROM(BrightMode3);
43              colour_set = ReadEEPROM(ColourMode3);
44              break;
45      // 发送模块控制指令 使esp进入配置模式
46      // 同时自身也进入配置模式
47      // 恢复预设模式初值
48      case 4: HekrModuleControl(HekrConfig);
49              WriteEEPROM(BrightModel1,0x32);WriteEEPROM(ColourModel1,0x80);
50              WriteEEPROM(BrightMode2,0x32);WriteEEPROM(ColourMode2,0x00);
51              WriteEEPROM(BrightMode3,0x32);WriteEEPROM(ColourMode3,0xFF);
52              break;
53      default:
54          break;
55  }
56  UpdateBright();
```

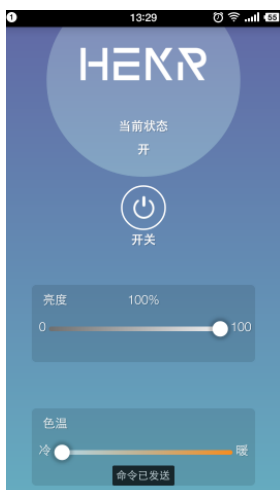
调用 HEKR API 发出配置模式试 wifi 模块进入配置模式
同时无级调光灯会以呼吸状态回应直到配置完成
此时可以使用 APP 进行配置绑定

- 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

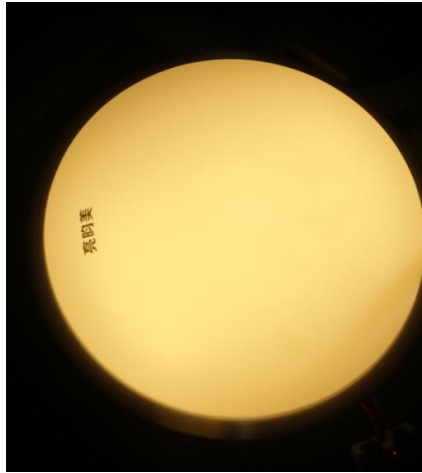
正白 100%



冷白 100%



暖白 100%



实物图



相关链接及反馈

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

无级调光示例代码也在 SDK 中。

氩氩 HEKR 模块透传协议：<http://docs.hekr.me/protocol/>

问题反馈：pengyu.zhang@hekr.me 965006619@qq.com

版本修订

2015.9.22 新建文档