

前言：本文描述了如何仅使用一片 ESP8266 的 1 个 GPIO 驱动 WS2812B 型全彩 RGB LED，底层驱动参考了 <https://github.com/cnlohr/ws2812esp8266>，驱动程序还移植 Adafruit_NeoPixel 的应用库，可实现多种常用特效。在此基础上还连接到了机智云物联网平台，利用其强大的**代码自动生成工具**，简单三步即完成了一个智能灯的开发。

视频：<http://player.youku.com/player.php/sid/XMjc0MDM3NTE4MA==/v.swf>



Step1:在开发者中心新建产品；

个人项目 / **WS2812B智能灯** 选项 · 申请发布

产品信息	基本信息
基本信息	产品名称： WS2812B智能灯
数据点	产品类型： 其它/其它
虚拟设备	通讯方式： Wi-Fi
设备日志	Product Key： 3b0c24732ea84e9f9e93ce37edac1928
开发向导	Product Secret： 041b*****2bff 显示完整密钥
服务	是否变长数据点： 否
应用配置	创建时间： 2017-05-01
应用开发	更新时间： 2017-05-01
MCU开发	描述： 无
产测工具	修改
固件升级 (OTA)	
+ 添加服务	
统计	
概览	
新增上线	

Step2:添加数据点，可使用微信宠物屋 Demo 模板导入，删除不需要的即可；

产品信息

- 基本信息
- 数据点**
- 虚拟设备
- 设备日志
- 开发向导

服务

- 应用配置
- 应用开发
- MCU开发
- 产测工具
- 固件升级 (OTA)
- + 添加服务

统计

- 概览
- 新增上线
- 活跃设备
- 活跃周期

数据点

定义数据点教程

+ 新建数据点 | 管理

显示名称: 设定LED...	标识名: LED_Co...	读写类型: 可写	数据类型: 枚举	枚举范围: 0,自定义...	备注: 无				
显示名称: 设定LED...	标识名: LED_R	读写类型: 可写	数据类型: 数值	数据范围: 0 - 254	分辨率: 1	增量: 0	备注: 无		
显示名称: 设定LED...	标识名: LED_G	读写类型: 可写	数据类型: 数值	数据范围: 0 - 254	分辨率: 1	增量: 0	备注: 无		
显示名称: 设定LED...	标识名: LED_B	读写类型: 可写	数据类型: 数值	数据范围: 0 - 254	分辨率: 1	增量: 0	备注: 无		

Step3:在 MCU 开发栏目选择 SoC 方案生成代码 SDK ;

个人项目 / WS2812B智能灯

选项 | 申请发布

产品信息

- 基本信息
- 数据点
- 虚拟设备
- 设备日志
- 开发向导

服务

- 应用配置
- 应用开发
- MCU开发**

MCU开发

MCU代码生成结果

硬件方案: SoC
硬件平台: ESP8266

下载 | 修改

Step4: 下载通用版本调试 APP, 支持安卓和 IOS :

http://dev.gizwits.com/zh-cn/developer/resource/demo_app?protoc=WIFI

机智云 Gizwits

硬件接入 | 文档中心^{NEW} | 下载中心 | 社区

qldeng@gizw...

下载中心

硬件开发资源

- GoKit MCU 示例工程
- GAgent

客户端开发资源

- 开源框架
- 设备接入SDK
- 统计分析SDK

开发与调试工具

- 机智云Wi-Fi/移动通信产品调试APP
- 产测工具

机智云Wi-Fi/移动通信产品调试APP

机智云App是全球首款IoT设备通用调试工具, 根据开发者自定义的产品功能, 自动生成可响应的控制页面。开发者在机智云平台开发智能硬件时, 可以很方便地使用该App对硬件设备进行调试和验证。此App有完整的用户注册、登陆和注销流程, 并且可以完成机智云智能硬件的配置入网、设备搜索、设备绑定、设备登录、设备控制、远程控制、状态更新、本地远程切换等基本设备操作。

机智云Wi-Fi/移动通信产品调试App for iOS 1.9.2
发布时间: 2017.4.13 23:03 | [更新信息](#) | [旧版本下载](#) | [机智云 App下载](#)

机智云Wi-Fi/移动通信产品调试App for Android 1.9.2
发布时间: 2017.4.13 23:04 | [更新信息](#) | [旧版本下载](#) | [机智云 App下载](#)

Step5:把下载的 SDK 改个名字, 原来的有点长, 不方便, 我这里改为

SoC_ESP8266_WS2812B_SmartLEDs

然后添加 WS2812B 型 RGB LED 的驱动文件 (可在附件源码中找到这 2 个驱动文件):

把 Adafruit_NeoPixel.c 文件放到 SoC_ESP8266_WS2812B_SmartLEDs\app\driver 下

把 Adafruit_NeoPixel.h 文件放到 SoC_ESP8266_WS2812B_SmartLEDs\app\include\driver 下

本驱动提供如下功能：

```
Adafruit_NeoPixel.h *
1  #ifndef _Adafruit_NeoPixel_H
2  #define _Adafruit_NeoPixel_H
3
4  #include "c_types.h"
5  #include "user_interface.h"
6  #include "ets_sys.h"
7  #include "gpio.h"
8
9  //thanks for https://github.com/cnLohr/ws2812esp8266
10 //thanks for https://github.com/adafruit/Adafruit_NeoPixel
11
12 #define WSGPIO 0 //must use the ESP8266 GPIO0 as the pin to drive WS2812B RGB LED!!!
13 //user can change
14 #define PIXEL_MAX 7 //the total numbers of LEDs you are used in your project
15
16 //You will have to os_intr_lock();    os_intr_unlock();
17
18 void ICACHE_FLASH_ATTR ws2812_strip(uint8_t R,uint8_t G,uint8_t B, uint16_t Length);
19 void ICACHE_FLASH_ATTR setAllPixelColor(uint8_t r, uint8_t g, uint8_t b);
20 void ICACHE_FLASH_ATTR setPixelColor(uint16_t n, uint8_t r, uint8_t g, uint8_t b);
21 void SetPixelColor(uint16_t n, uint32_t c);
22 uint32_t Color(uint8_t r, uint8_t g, uint8_t b);
23 uint32_t Wheel(uint8_t WheelPos);
24 void rainbowCycle(uint8_t wait) ;
25 void theaterChase(uint32_t c, uint8_t wait);
26 void theaterChaseRainbow(uint8_t wait);
27 void colorWipe(uint32_t c, uint8_t wait);
28 void WS2812B_Test(void);
29 void ICACHE_FLASH_ATTR WS2812B_Init(void);
30 |
31 #endif
```

```
Adafruit_NeoPixel.c
203     uint16_t i=0;
204     for( i=0; i<PIXEL_MAX; i++)
205     {
206         SetPixelColor(i, c);
207         HAL_Delay(wait);
208     }
209 }
210
211 void ICACHE_FLASH_ATTR WS2812B_Init(void)
212 {
213     setAllPixelColor(0, 0, 0);
214 }
215 void WS2812B_Test(void)
216 {
217     colorWipe(Color(255, 0, 0), 50); // Red
218     colorWipe(Color(0, 255, 0), 50); // Green
219     colorWipe(Color(0, 0, 255), 50); // Blue
220     //colorWipe(strip.Color(0, 0, 0, 255), 50); // White RGBW
221     // Send a theater pixel chase in...
222     theaterChase(Color(127, 127, 127), 50); // White
223     theaterChase(Color(127, 0, 0), 50); // Red
224     theaterChase(Color(0, 0, 127), 50); // Blue
225
226     rainbow(20);
227     rainbowCycle(20);
228     theaterChaseRainbow(50);
229 }
```

Step6:用文本编辑器如 notepad++、SublimeText 等打开目录 SoC_ESP8266_WS2812B_SmartLEDs\app\user 下的 user_main.c 文件，然后做如下修改；首先添加头文件：

```
user_main.c
1  /**
2  ****
3  * @file      user_main.c
4  * @brief    SOC版 入口文件
5  * @author   Gizwits
6  * @date    2016-09-05
7  * @version  V03010201
8  * @copyright Gizwits, bigfanofIoT
9  *
10 * @note    机智云·只为智能硬件而生
11 *          Gizwits Smart Cloud for Smart Products
12 *          链接|增值|开放|中立|安全|自有|自由|生态
13 *          www.gizwits.com
14 * @For more projects, please visit: http://club.gizwits.com/forum.php
15 ****
16 #include "ets_sys.h"
17 #include "osapi.h"
18 #include "user_interface.h"
19 #include "gagent_soc.h"
20 #include "user_devicefind.h"
21 #include "user_webserver.h"
22 #include "gizwits_protocol.h"
23 #include "driver/hal_key.h"
24
25 #include "driver/Adafruit_NeoPixel.h"
26
27 #if ESP_PLATFORM
28 #include "user_esp_platform.h"
29 #endif
30
```

我们此次使用的是 Gokit3 SoC 开发板，由于驱动 LED 的 GPIO 是 GPIO0（Gokit3 SoC 开发板上对应 D6），而生成的代码包默认使用了 GPIO0 连接到了 Gokit3 SoC 版的按键 key1，因此把按键 1 相关删除，一般不需要使用产测功能，若自己使用 esp8266-12f 搭建最小系统，可保留此功能：

```
/**@name 按键相关定义
* @{
*/
#define GPIO_KEY_NUM 1 ///< 定义按键成员总数
#define KEY_1_IO_MUX PERIPHS_IO_MUX_MTMS_U ///< ESP8266 GPIO 功能
#define KEY_1_IO_NUM 14 ///< ESP8266 GPIO 编号
#define KEY_1_IO_FUNC FUNC_GPIO14 ///< ESP8266 GPIO 名称
LOCAL key_typedef_t * singleKey[GPIO_KEY_NUM]; ///< 定义单个按键成员数组指针
LOCAL keys_typedef_t keys; ///< 定义总的按键模块结构体指针
/**@} */

/** 用户区当前设备状态结构体*/
dataPoint_t currentDataPoint;

/**
* key2 按键短按处理
* @param none
* @return none
*/
LOCAL void ICACHE_FLASH_ATTR key2ShortPress(void)
{
    os_printf("#### key2 short press, soft ap mode \n");
    gizwitsSetMode(WIFI_SOFTAP_MODE);
}

/**
* key2 按键长按处理
```

然后在主函数里面调用 LED 的初始化：

```
user_main.c
197     os_printf( "epc1=0x%08x, epc2=0x%08x, epc3=0x%08x\n",
198             rtc_info->epc1, rtc_info->epc2, rtc_info->epc3);
199 }
200
201 if (system_upgrade_userbin_check() == UPGRADE_FW_BIN1)
202 {
203     os_printf( "---UPGRADE_FW_BIN1---\n");
204 }
205 else if (system_upgrade_userbin_check() == UPGRADE_FW_BIN2)
206 {
207     os_printf( "---UPGRADE_FW_BIN2---\n");
208 }
209
210 //user init
211
212 keyInit();
213
214 WS2812B_Init();
215 //WS2812B_Test();
216
217 //gizwits InitSIG_UPGRADE_DATA
218 os_memset((uint8_t *)&currentDataPoint, 0, sizeof(currentDataPoint));
219 gizwitsInit();
220
221 system_os_task(gizwitsUserTask, USER_TASK_PRIO_0, userTask, 1);
222
223 //user timer
```

Step7:打开目录 SoC_ESP8266_WS2812B_SmartLEDs\app\Gizwits 下的 gizwits_product.c 文件；

首先添加头文件：

```
gizwits_product.c
1  /**
2  ****
3  * @file      gizwits_product.c
4  * @brief     Gizwits 控制协议处理, 及平台相关的硬件初始化
5  * @author    Gizwits
6  * @date      2016-09-05
7  * @version   V03010201
8  * @copyright Gizwits, , bigfanofIoT
9  *
10 * @note      机智云. 只为智能硬件而生
11 *            Gizwits Smart Cloud for Smart Products
12 *            链接|增值|开放|中立|安全|自有|自由|生态
13 *            www.gizwits.com
14 * @For more projects, please visit: http://club.gizwits.com/forum.php
15 ****
16 #include <stdio.h>
17 #include <string.h>
18 #include "gizwits_protocol.h"
19 #include "driver/hal_key.h"
20
21 #include "driver/Adafruit_NeoPixel.h"
22
```

然后在有 user handle 注释的地方添加用户代码：

如下是 ESP8266 接收到手机 APP 指令后执行的动作；

```
gizwits_product.c
58 switch(info->event[i])
59 {
60
61     case EVENT_LED_COLOR:
62         currentDataPoint.valueLED_Color = dataPointPtr->valueLED_Color;
63         os_printf("Evt: EVENT_LED_COLOR %d\n", currentDataPoint.valueLED_Color);
64         switch(currentDataPoint.valueLED_Color)
65         {
66             case LED_COLOR_VALUE0:
67                 //user handle
68                 setAllPixelColor(currentDataPoint.valueLED_R, currentDataPoint.valueLED_G, currentDataPoint.valueLED_B);
69                 break;
70             case LED_COLOR_VALUE1:
71                 //user handle
72                 setAllPixelColor(250, 250, 0);
73                 break;
74             case LED_COLOR_VALUE2:
75                 //user handle
76                 setAllPixelColor(250, 0, 70);
77                 break;
78             case LED_COLOR_VALUE3:
79                 //user handle
80                 setAllPixelColor(140, 30, 30);
81                 break;
82             default:
83                 break;
84         }
85     }
```

```
case EVENT_LED_R:
    currentDataPoint.valueLED_R = dataPointPtr->valueLED_R;
    os_printf("Evt:EVENT_LED_R %d\n",currentDataPoint.valueLED_R);
    //user handle
    setAllPixelColor(currentDataPoint.valueLED_R, currentDataPoint.valueLED_G, currentDataPoint.valueLED_B);
    break;
case EVENT_LED_G:
    currentDataPoint.valueLED_G = dataPointPtr->valueLED_G;
    os_printf("Evt:EVENT_LED_G %d\n",currentDataPoint.valueLED_G);
    //user handle
    setAllPixelColor(currentDataPoint.valueLED_R, currentDataPoint.valueLED_G, currentDataPoint.valueLED_B);
    break;
case EVENT_LED_B:
    currentDataPoint.valueLED_B = dataPointPtr->valueLED_B;
    os_printf("Evt:EVENT_LED_B %d\n",currentDataPoint.valueLED_B);
    //user handle
    setAllPixelColor(currentDataPoint.valueLED_R, currentDataPoint.valueLED_G, currentDataPoint.valueLED_B);
    break;
```

源码开发环境搭建，固件烧录教程查看：

<http://docs.gizwits.com/zh-cn/deviceDev/WiFiSOC/GoKit3S%E4%BA%8C%E6%AC%A1%E5%BC%80%E5%8F%91.html>

工程源码：https://github.com/DengQilong/ESP8266_WS2812B_SmartLamp

原文出处：<http://club.gizwits.com/thread-6116-1-1.html>