#### 前序

- 1注册GizWits账号
- 2 定义"微信宠物屋"产品
- 3 生成微信宠物屋代码
- 4 移植微信宠物屋驱动代码
- 5 编译烧录固件并进行测试
- 6 相关说明

### 前序

机智云提供了官方 微信宠物屋 测试固件以及相应平台的驱动库文件,开发者可以直接烧录并使用机智云APP进行测试测试,烧录方式请查看本文档的 编译烧录固件并进行测试 一节。

同时机智云也为开发者提供了SOC版与MCU版的代码自动生成功能,开发者可根据此文档进行自主开发,完成属于自己的 微信宠物屋。

## 1注册GizWits账号

首先登陆机智云宫网, 注册开发者账号。

## 2 定义"微信宠物屋"产品

1)选择个人项目,点击创建新产品



2)输入相应的产品信息,注意这里的技术方案选择"WiFi/移动网络方案",通信方式选择"WiFi",最后点击"保存"



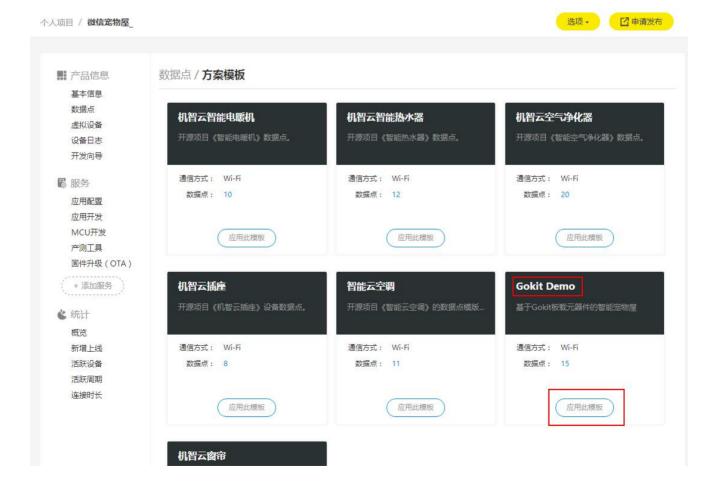
#### 3)点击"去添加数据点"



4)在"管理"中点击"选择产品数据点模板"



5)选择"Gokit Demo",并"应用此模板"



#### 注:这里会导入基于Gokit板载元器件的智能宠物屋数据点模板

#### 接下来点击"添加"



6)可以看到导入的"微信宠物屋"的相关数据点



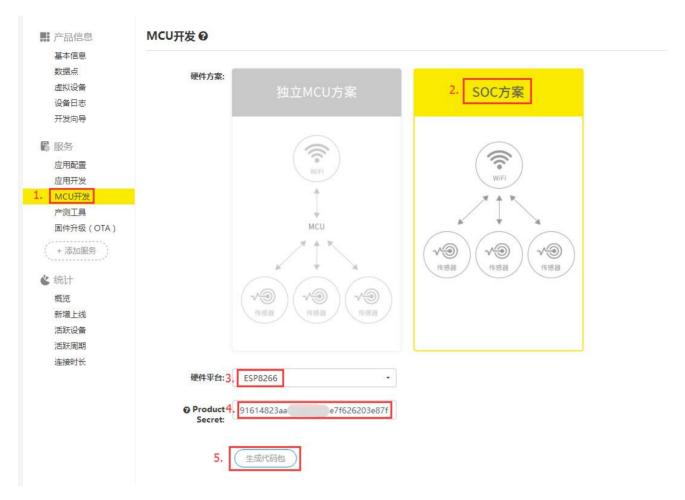
# 3 生成微信宠物屋代码

1) 首先,在使用"代码自动生成工具"前要获取产品所对应的"Product Secret" (后文简称"PS")

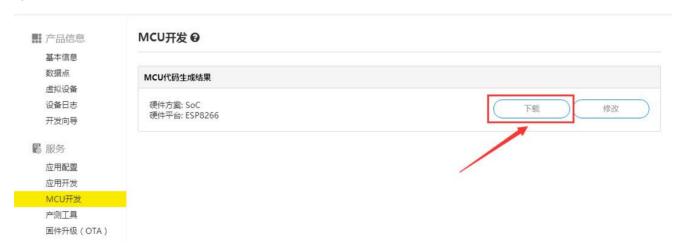


#### 2) 生成SOC版代码

在"服务"中选择"SOC方案",平台选择默认的"ESP8266",填入之前在"基本信息"中获得的 **PS**,然后点击"生成代码包",等待一段时间后便会生成相应的源码工程。

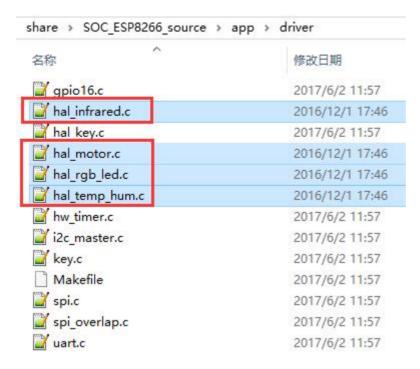


3) 下载生成好的代码工程压缩包并解压

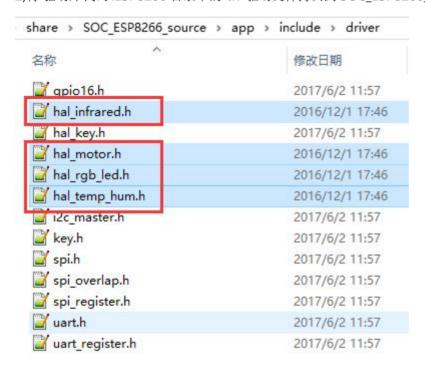


# 4 移植微信宠物屋驱动代码

1)将 微信宠物屋教程\ESP8266\驱动库代码 目录下的".c"驱动文件拷贝到 SOC\_ESP8266\_source\app\driver 目录下



2)将 驱动库代码\ESP8266 目录下的".h"驱动文件拷贝到 SOC\_ESP8266\_source\app\include\driver 目录下



3)在代码中添加相应的函数调用

在SOC\_ESP8266\_source\app\user\user\_main.c 文件中添加各驱动库的头文件

```
#include "driver/hal_infrared.h"
#include "driver/hal_motor.h"
#include "driver/hal_rgb_led.h"
#include "driver/hal_temp_hum.h"
```

在**SOC\_ESP8266\_source\app\user\user\_main.c**文件的**datapointInit( )**函数中添加各sensor的初始化

```
void datapointInit(void)
{
    os_memset((uint8_t *)&currentDataPoint, 0, sizeof(dataPoint_t));

    ///< 新添加代码: RGB LED初始化
    rgbGpioInit();
    rgbLedInit();

    ///< 新添加代码: 电机初始化
    motorInit();
    motorControl(0);

    ///< 新添加代码: 温湿度初始化
    dh11Init();

    ///< 新添加代码: 红外初始化
    irInit();
}
```

在 SOC\_ESP8266\_source\app\user\user\_main.c 文件中以及userHandle()函数中添加只读型传感器数据点相关的代码

```
void ICACHE_FLASH_ATTR userTimerFunc(void)
   uint8_t ret = 0;
   uint8_t curTemperature = 0;
   uint8_t curHumidity = 0;
   uint8 t curIr = 0;
    static uint8 t thCtime = 0;
    static uint8_t irCtime = 0;
   thCtime++;
   irCtime++;
    ///< 新添加代码: 红外传感器数据获取
   if(INF TIMEOUT < irCtime)</pre>
       irCtime = 0;
       curIr = irUpdateStatus();
        currentDataPoint.valueInfrared = curIr;
   }
    ///< 新添加代码: 温湿度传感器数据获取
    if(TH TIMEOUT < thCtime)</pre>
       thCtime = 0;
       ret = dh11Read(&curTemperature, &curHumidity);
       if(0 == ret)
           currentDataPoint.valueTemperature = curTemperature;
           currentDataPoint.valueHumidity = curHumidity;
       }
       else
           os_printf("@@@@ dh11Read error ! \n");
        }
   }
    system_os_post(USER_TASK_PRIO_0, SIG_UPGRADE_DATA, 0);
}
```

在**SOC\_ESP8266\_source\app\user\user\_main.c**文件的**key2ShortPress( )**函数与**key2LongPress( )**函数中添加 长/短按key2时的LED点亮代码

```
LOCAL void ICACHE_FLASH_ATTR key2ShortPress(void)
{
    GIZWITS_LOG("#### key2 short press, soft ap mode \n");
    rgbControl(250, 0, 0); ///< 新添加代码: Soft AP mode, RGB red gizwitsSetMode(WIFI_SOFTAP_MODE);
}
```

```
LOCAL void ICACHE_FLASH_ATTR key2LongPress(void)
{
    GIZWITS_LOG("#### key2 long press, airlink mode\n");
    rgbControl(0, 250, 0); ///< 新添加代码: AirLink mode, RGB Green gizwitsSetMode(WIFI_AIRLINK_MODE);
}
```

在SOC\_ESP8266\_source\app\Gizwits\gizwits\_product.c文件的gizwitsEventProcess()函数中添加可写型传感器数据点相关的代码

```
int8_t ICACHE_FLASH_ATTR gizwitsEventProcess(eventInfo_t *info, uint8_t *data, uint32_t len)
 uint8_t i = 0;
 dataPoint_t *dataPointPtr = (dataPoint_t *)data;
 moduleStatusInfo_t *wifiData = (moduleStatusInfo_t *)data;
 if((NULL == info) || (NULL == data))
 {
   return -1;
 }
 for(i=0; i<info->num; i++)
   switch(info->event[i])
     case EVENT LED OnOff:
       currentDataPoint.valueLED OnOff = dataPointPtr->valueLED OnOff;
       os printf("Evt: EVENT LED ONOFF %d \n", currentDataPoint.valueLED OnOff);
       if(0x01 == currentDataPoint.valueLED_OnOff)
           rgbControl(254, 0, 0); ///< 新添加代码: 对应开启红灯
       }
       else
           rgbControl(0, 0, 0); ///< 新添加代码:对应关闭红灯
       }
       break;
     case EVENT_LED_Color:
       currentDataPoint.valueLED Color = dataPointPtr->valueLED Color;
       os_printf("Evt: EVENT_LED_COLOR %d\n", currentDataPoint.valueLED_Color);
       switch(currentDataPoint.valueLED_Color)
       {
         case LED Color VALUE0:
rgbControl(currentDataPoint.valueLED_R,currentDataPoint.valueLED_G,currentDataPoint.valueLED_B);
           break;
         case LED Color VALUE1:
           rgbControl(254, 254, 0); ///< 新添加代码:对应LED组合颜色-黄色
         case LED Color VALUE2:
           rgbControl(254, 0, 70);
                                     ///< 新添加代码: 对应LED组合颜色-紫色
           break;
         case LED Color VALUE3:
           rgbControl(238, 30, 30); ///< 新添加代码:对应LED组合颜色-粉色
           break;
         default:
           break;
       }
       break;
     case EVENT_LED_R:
       currentDataPoint.valueLED_R = dataPointPtr->valueLED_R;
```

```
os_printf("Evt:EVENT_LED_R %d\n",currentDataPoint.valueLED_R);
        ///< 新添加代码:对应设置LED组合色
rgbControl(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);
        ///< 新添加代码:对应设置LED组合色
rgbControl(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);
       ///< 新添加代码:对应设置LED组合色
rgbControl(currentDataPoint.valueLED_R,currentDataPoint.valueLED_G,currentDataPoint.valueLED_B);
     case EVENT Motor Speed:
       currentDataPoint.valueMotor Speed = dataPointPtr->valueMotor Speed;
       os_printf("Evt:EVENT_MOTOR_SPEED %d\n",currentDataPoint.valueMotor_Speed);
       ///< 新添加代码: 对应设定电机转速
       motorControl(currentDataPoint.valueMotor Speed);
       break;
     case WIFI_SOFTAP:
       break;
     case WIFI AIRLINK:
       break;
     case WIFI STATION:
        break;
     case WIFI CON ROUTER:
                                      ///< 新添加代码: 连接路由后关闭LED灯
       rgbControl(0, 0, 0);
       break;
     case WIFI DISCON ROUTER:
        break;
     case WIFI_CON_M2M:
       break;
     case WIFI DISCON M2M:
       break;
     case WIFI RSSI:
       os_printf("RSSI %d\n", wifiData->rssi);
       break;
     case TRANSPARENT DATA:
       os printf("TRANSPARENT DATA \n");
       //user handle , Fetch data from [data] , size is [len]
       break;
     default:
       break;
   }
 system_os_post(USER_TASK_PRIO_0, SIG_UPGRADE_DATA, 0);
```

```
return 0;
}
```

### 5 编译烧录固件并进行测试

可在 下载中心 中下载对应平台的微信宠物屋官方测试固件以及驱动库文件



测试固件位于: 微信宠物屋教程\ESP8266\官方成品固件\user1.4096.new.6.bin

SOC版的固件烧录方式可查看 GoKit3(S) 开发环境搭"源码编译"和"固件下载"这两章节。

微信宠物屋的操作方式可查看 GoKit3(S)使用说明书一文

## 6相关说明

说明: SOC版与MCU版的工程环境搭建与代码细节介绍请查看<u>文档中心</u>中设备开发的**Gokit资**料一节,本章节只介绍微信宠物屋驱动程序的移植方法。

