

[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，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[机智云官网](#)，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”

产品分类： 可穿戴产品及智能硬件 其他

产品名称：

技术方案：

Wi-Fi/移动网络方案



选择通讯方式：☒ Wi-Fi ☐ 移动网络
是否变长数据点：☐

保存

蓝牙方案



网关方案



在线咨询

3)点击“去添加数据点”

个人项目 / 微信宠物屋

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
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
连接时长

开发向导

1 定义产品功能 2 MCU开发 App/微信开发 3 功能调试 4 发布产品

定义产品功能说明

产品开发的第一步是定义产品的功能，一个数据点可以定义为产品的某个功能，如开关等。产品的数据点如何定义，请查看教程 [《如何定义数据点》](#)。

去添加数据点

MCU 开发资源

机智云根据你定义的产品数据点，会自动生成MCU串口通信代码或整个MCU工程代码，同时也有SOC方案的工程代码。

如果想了解接入机智云的串口通信协议，可下载 [《微信宠物屋 - 机智云SOC方案接入通信协议文档》](#) 或 [《微信宠物屋 - 机智云独立MCU方案接入通信协议文档》](#)。此外，也提供功能参数的 [《微信宠物屋 - 机智云接入JSON文档》](#)，此文档是对协议的格式化说明，包含每个数据点的ID、描述、数据类型、位置信息等。

进入MCU开发

App 开发资源

4)在“管理”中点击“选择产品数据点模板”

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

连接时长

数据点 ?

定义数据点教程



尚未创建产品的数据点，快去建立适合的数据点吧~

+ 新建数据点

管理

选择产品数据点模板

导入Excel

5)选择“Gokit Demo”，并“应用此模板”

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

连接时长

数据点 / 方案模板

机智云智能电暖机

开源项目《智能电暖机》数据点。

通信方式：Wi-Fi

数据点：10

应用此模板

机智云智能热水器

开源项目《智能热水器》数据点。

通信方式：Wi-Fi

数据点：12

应用此模板

机智云空气净化器

开源项目《智能空气净化器》数据点。

通信方式：Wi-Fi

数据点：20

应用此模板

机智云插座

开源项目《机智云插座》设备数据点。

通信方式：Wi-Fi

数据点：8

应用此模板

智能云空调

开源项目《智能云空调》的数据点模板...

通信方式：Wi-Fi

数据点：11

应用此模板

Gokit Demo

基于Gokit板载元器件的智能宠物屋

通信方式：Wi-Fi

数据点：15

应用此模板

机智云窗帘

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”）

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
- 连接时长

基本信息

产品名称： 微信宠物屋

产品类型： 智能家居/生活小家电/咖啡机

通讯方式： Wi-Fi

Product Key： 189655b1df9a473c8312f6792b917dc5

Product Secret： 9161*****e87f [显示完整密钥](#)

是否变长数据点： 否

设备分享功能： 未开启

创建时间： 2017-06-01

更新时间： 2017-06-01

描述： 无

[修改](#)

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。

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

1. 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

连接时长

MCU开发

硬件方案:

独立MCU方案

2. SOC方案

硬件平台: 3. ESP8266

Product Secret: 4. 91614823aa e7f626203e87f

5. 生成代码包

3) 下载生成好的代码工程压缩包并解压

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)

MCU开发

MCU代码生成结果

硬件方案: SoC

硬件平台: ESP8266

下载

修改

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

share > SOC_ESP8266_source > app > driver	
名称	修改日期
gpio16.c	2017/6/2 11:57
hal_infrared.c	2016/12/1 17:46
hal_key.c	2017/6/2 11:57
hal_motor.c	2016/12/1 17:46
hal_rgb_led.c	2016/12/1 17:46
hal_temp_hum.c	2016/12/1 17:46
hw_timer.c	2017/6/2 11:57
i2c_master.c	2017/6/2 11:57
key.c	2017/6/2 11:57
Makefile	2017/6/2 11:57
spi.c	2017/6/2 11:57
spi_overlap.c	2017/6/2 11:57
uart.c	2017/6/2 11:57

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

share > SOC_ESP8266_source > app > include > driver	
名称	修改日期
gpio16.h	2017/6/2 11:57
hal_infrared.h	2016/12/1 17:46
hal_key.h	2017/6/2 11:57
hal_motor.h	2016/12/1 17:46
hal_rgb_led.h	2016/12/1 17:46
hal_temp_hum.h	2016/12/1 17:46
i2c_master.h	2017/6/2 11:57
key.h	2017/6/2 11:57
spi.h	2017/6/2 11:57
spi_overlap.h	2017/6/2 11:57
spi_register.h	2017/6/2 11:57
uart.h	2017/6/2 11:57
uart_register.h	2017/6/2 11:57

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

#define USER_TIME_MS 100		///< 新添加代码: 更改定
时器间隔为100ms		
#define TH_TIMEOUT	(1000 / USER_TIME_MS)	///< 新添加代码: 温湿度
采集间隔为1S (1000ms)		
#define INF_TIMEOUT	(500 / USER_TIME_MS)	///< 新添加代码: 红外采
集间隔为500ms		


```

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)
    {
        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)
    {
        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
                        break;
                    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);
    break;
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:
    rgbControl(0, 0, 0);          ///< 新添加代码：连接路由后关闭LED灯
    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

The screenshot shows the 'Download Center' (下载中心) page on the Gizwits website. The left sidebar contains navigation links for hardware development resources, client development resources, development and debugging tools, and open source code. The main content area is titled '微信宠物屋' (WeChat Pet House) and provides a description of the product. Below the description, there is a table listing firmware downloads for different GoKit boards:

固件名称	发布时间	更新信息	旧版本下载	资源下载
微信宠物屋 for GoKit3(S) ESP8266 V03000003	2016.12.01 19:46	更新信息	旧版本下载	资源下载
微信宠物屋 for GoKit 2/3 STM32 V03010101	2016.12.01 19:10	更新信息	旧版本下载	资源下载
微信宠物屋 for GoKit 2 Arduino 2.3.1	2016.1.04 12:19	更新信息	旧版本下载	资源下载
微信宠物屋 for GoKit 1.0.20141116	2015.4.22 17:38	更新信息	旧版本下载	资源下载

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[文档中心](#)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。

The screenshot shows the 'Documentation Center' (文档中心) page on the Gizwits website. The top navigation bar includes links for 'Platform Overview', 'Quick Start', 'APP Development', 'WeChat Development', 'Device Development', 'Cloud Platform', and 'User Manual'. The 'Device Development' (设备开发) link is highlighted. The left sidebar contains a 'Table of Contents' (目录) with links to 'Gizwits Platform Overview', 'Gizwits Platform Architecture', 'Gizwits Advantages', and 'Gizwits Global Network Report'. The main content area is titled '平台概述' (Platform Overview) and provides a brief introduction to the Gizwits platform. A dropdown menu is open under '设备开发', showing a list of resources including 'Gokit资料' (Gokit Resources), which is highlighted. The 'Gokit资料' dropdown menu lists various documents related to the GoKit series, including 'GoKit3(S)使用说明书' (GoKit3(S) User Manual), 'GoKit3(S)开发套件介绍' (GoKit3(S) Development Kit Introduction), 'GoKit3(S)开发环境搭建' (GoKit3(S) Development Environment Setup), 'GoKit3(S)源码详解' (GoKit3(S) Source Code Detailed Explanation), 'GoKit3(V)使用说明书' (GoKit3(V) User Manual), 'GoKit3(V)开发指南' (GoKit3(V) Development Guide), 'GoKit3(V)词条管理' (GoKit3(V) Keyword Management), 'GoKit3(V)源码详解' (GoKit3(V) Source Code Detailed Explanation), and 'GoKit2 使用指南' (GoKit2 User Guide).

