

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 导入微信宠物屋工程配置文件
- 5 移植微信宠物屋驱动代码
- 6 编译烧录固件并进行测试
- 7 相关说明

前序

本文主要介绍如何使用 机智云代码自动生成工具 结合 **STM32CubeMX**，完成 微信宠物屋 的快速开发。

机智云 [下载中心](#) 提供了STM32 版的信宠物屋 测试固件 以及 **STM32CubeMX** 版的 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 **6 编译烧录固件并进行测试** 一节。GoKit MCU示（**STM32CubeMX** 版）例工程下载后如图所示：

GoKit_MCU_STM32_V03010101测试固件及开发资源2017121209 >		
名称	修改日期	
固件	2017/7/28 15:46	
驱动库代码_CubeMX版	2017/12/12 9:58	
驱动库代码_标准外设库V3.5.0版	2017/6/12 17:40	
DIY微信宠物屋_MCU版.pdf	2017/7/28 15:36	
DIY微信宠物屋_STM32CubeMX版.pdf	2017/12/7 17:32	
STM32F103C8x.ioc	2017/12/6 15:20	

接下来是 微信宠物屋 的快速开发介绍

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”

产品列表 / 创建新产品

产品分类：

可穿戴产品及智能硬件

其他

产品名称：

微信宠物屋

技术方案：

Wi-Fi/移动网络方案

蓝牙方案

网关方案

云端

Wi-Fi/3G等移动网络

设备

手机

蓝牙

设备

云端

Wi-Fi/3G等移动网络

网关

子设备

子设备

子设备

选择通讯方式：

Wi-Fi

移动网络

是否变长数据点：

保存

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

服务

统计

开发向导

定义产品功能说明

MCU 开发资源

App 开发资源

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

MCU 开发资源

App 开发资源

去添加数据点

进入MCU开发

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》的数据点模版...

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MCU版代码

在“服务”中选择“MCU方案”，平台选择为“STM32F103C8x”，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:

2. 独立MCU方案

WiFi

MCU

传感器

传感器

传感器

SOC方案

硬件平台: 3. STM32F103C8x

Product: 4. 91614823aa 7f626203e87f

Secret:

5. 生成代码包

3) 下载生成好的代码工程压缩包并解压

MCU开发

MCU代码生成结果

硬件方案: MCU

硬件平台: ArduinoUNOR3

下载

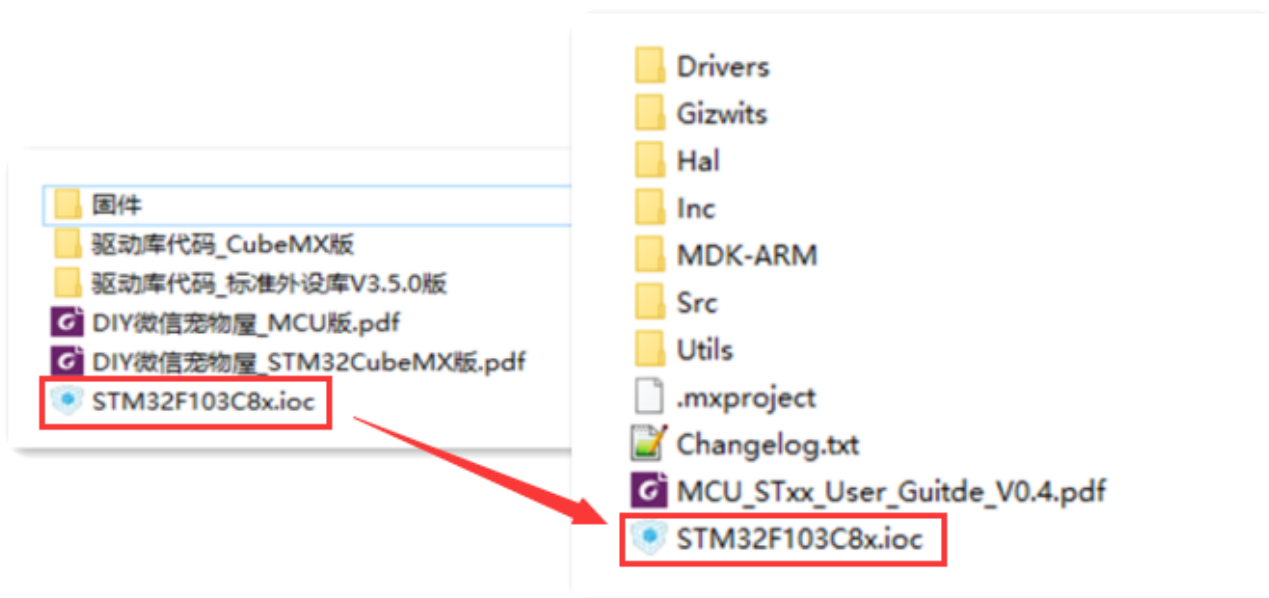
修改

Drivers	2017/12/7 15:00
Gizwits	2017/12/7 15:01
Hal	2017/12/7 14:54
Inc	2017/12/7 15:01
MDK-ARM	2017/12/7 14:54
Src	2017/12/7 14:54
Utils	2017/12/7 14:54
.mxproject	2017/11/20 11:44
Changelog.txt	2017/11/20 11:44
MCU_STxx_User_Guide_V0.4.pdf	2017/11/20 11:44
STM32F103C8x.ioc	2017/11/20 11:44

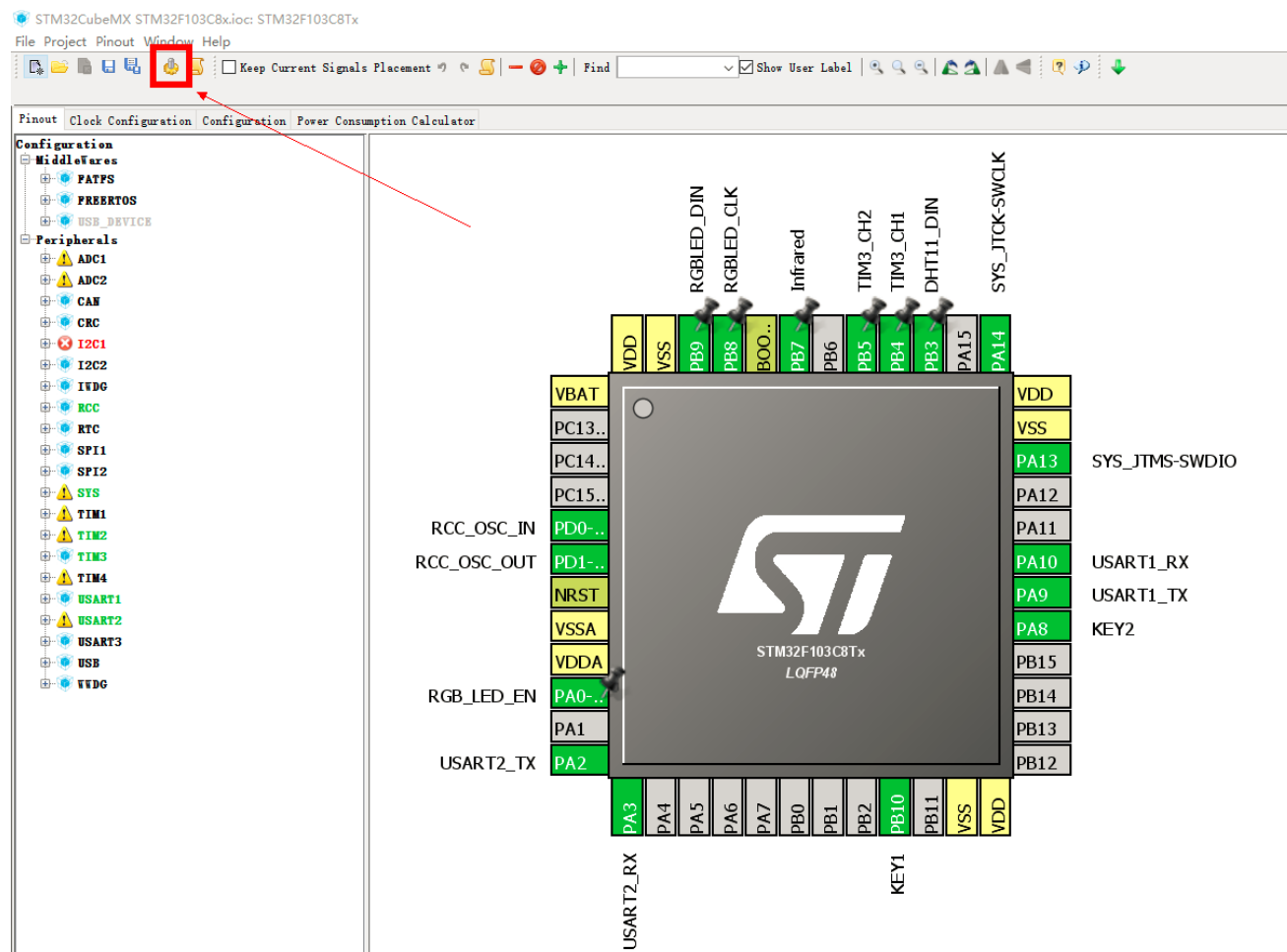
4 导入微信宠物屋工程配置文件

建议开发者提前阅读文档中心的 [《STM32CubeMX移植机智云自动生成代码详解》](#) 一文，了解STM32CubeMX的相关使用方法，本文不做过多叙述。

将前序中介绍的微信宠物屋CubeMX配置文件，替换掉自动生成代码中的原配置文件：

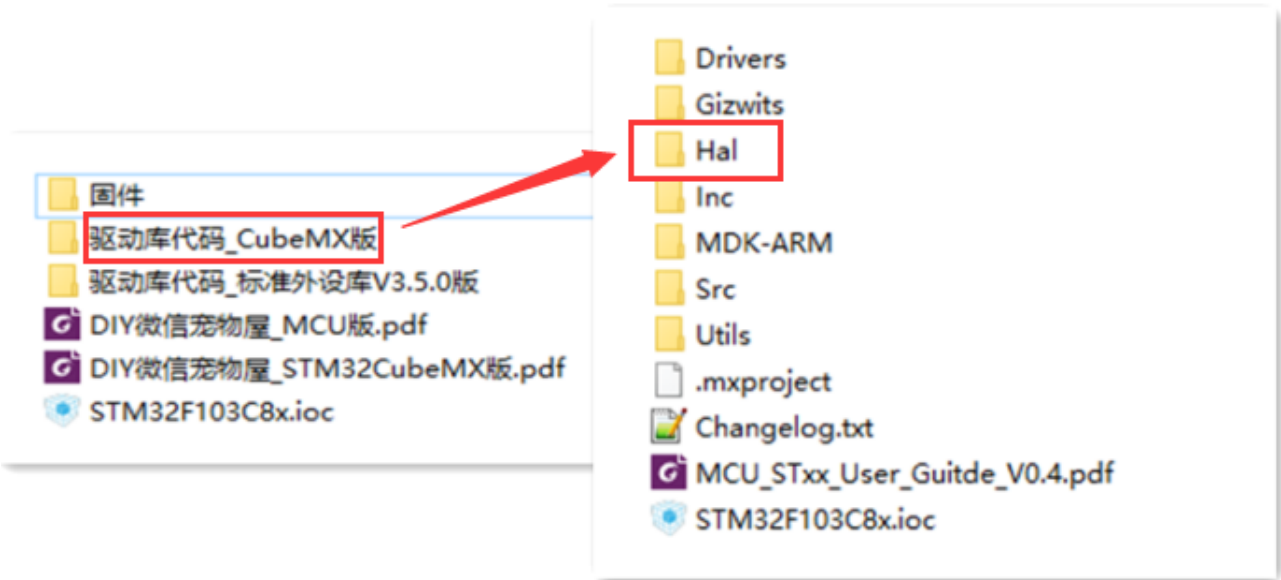


然后重新生成对应配置的STM32代码（并不会影响到原有的机智云协议处理代码）：



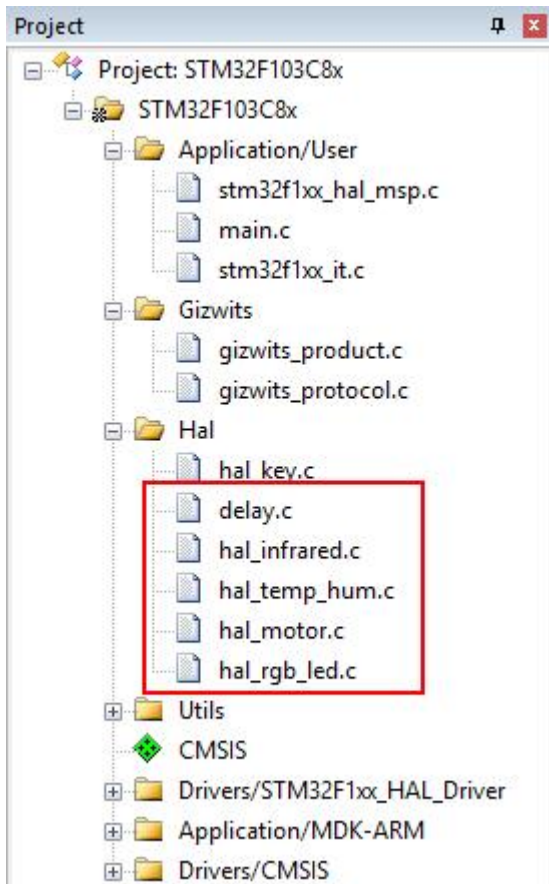
5 移植微信宠物屋驱动代码

1)将前序中介绍的 驱动库代码 文件拷贝到自动生成代码工程中的 MCU_STM32F103C8x_source\Hal 文件夹中



MCU_STM32F103C8x_source > Hal		
名称	修改日期	类型
delay.c	2017/12/1 16:20	C 文件
delay.h	2017/12/1 16:19	H 文件
hal_infrared.c	2017/12/6 15:35	C 文件
hal_infrared.h	2016/7/19 10:32	H 文件
hal_key.c	2017/11/20 11:44	C 文件
hal_key.h	2017/11/20 11:44	H 文件
hal_motor.c	2017/12/6 17:20	C 文件
hal_motor.h	2017/12/6 10:39	H 文件
hal_rgb_led.c	2017/12/6 17:19	C 文件
hal_rgb_led.h	2017/12/6 15:23	H 文件
hal_temp_hum.c	2017/12/6 17:19	C 文件
hal_temp_hum.h	2017/12/1 18:13	H 文件

2)在工程项目中国添加驱动代码的“.c”文件



3)在代码中添加相应的函数调用

在 **MCU_STM32F103C8x_source\Src\main.c** 和 **MCU_STM32F103C8x_source\Gizwits\gizwits_product.c** 文件中添加驱动库的头文件

```
#include "delay.h"
#include "hal_motor.h"
#include "hal_rgb_led.h"
#include "hal_temp_hum.h"
#include "hal_infrared.h"
```

在 **MCU_STM32F103C8x_source\Gizwits\gizwits_product.c** 文件的 **userInit()** 函数中添加各sensor的初始化

```
void userInit(void)
{
    memset((uint8_t *)&currentDataPoint, 0, sizeof(dataPoint_t));

    delay_init(72);    // 延时 初始化
    rgbLedInit();      // RGB LED 初始化
    dht11Init();       // 温湿度初始化
    irInit();           // 红外初始化
    motorInit();        // 电机初始化
    motorStatus(0);     // 电机转速初始化
}
```

在 **MCU_STM32F103C8x_source\Gizwits\gizwits_product.c** 文件的 **userHandle()** 函数中添加只读型传感器数据点相关的代码

```

void userHandle(void)
{
    uint8_t ret = 0;
    static uint32_t thLastTimer = 0;

    ///< 新添加代码：红外传感器数据获取
    currentDataPoint.valueInfrared = irHandle();

    ///< 新添加代码：温湿度传感器数据获取
    if((gizGetTimerCount() - thLastTimer) > 2000)    //上报间隔2S
    {
        ret = dht11Read((uint8_t *)&currentDataPoint.valueTemperature,
                        (uint8_t *)&currentDataPoint.valueHumidity);

        if(ret != 0)
        {
            printf("Failed to read DHT11 [%d] \n", ret);
        }

        thLastTimer = gizGetTimerCount();
    }
}

```

在 **MCU_STM32F103C8x_source\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

```

void key2ShortPress(void)
{
    GIZWITS_LOG("KEY2 PRESS ,Soft AP mode\n");
    #if !MODULE_TYPE
    gizwitsSetMode(WIFI_SOFTAP_MODE);
    #endif

    ///

```

```

void key2LongPress(void)
{
    ///

```

在 **MCU_STM32F103C8x_source\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 gizwitsEventProcess(eventInfo_t *info, uint8_t *gizdata, uint32_t len)
{
    uint8_t i = 0;
    dataPoint_t * dataPointPtr = (dataPoint_t *)gizdata;
    moduleStatusInfo_t * wifiData = (moduleStatusInfo_t *)gizdata;
    protocolTime_t * ptime = (protocolTime_t *)gizdata;

    #if MODULE_TYPE
    gprsInfo_t * gprsInfoData = (gprsInfo_t *)gizdata;
    #else
    moduleInfo_t * ptModuleInfo = (moduleInfo_t *)gizdata;
    #endif

    if((NULL == info) || (NULL == gizdata))
    {
        return -1;
    }

    for(i = 0; i < info->num; i++)
    {
        switch(info->event[i])
        {
            case EVENT_LED_OnOff:
                currentDataPoint.valueLED_OnOff = dataPointPtr->valueLED_OnOff;
                GIZWITS_LOG("Evt: EVENT_LED_OnOff %d \n", currentDataPoint.valueLED_OnOff);
                if(0x01 == currentDataPoint.valueLED_OnOff)
                {
                    ledRgbControl(254, 0, 0);
                }
                else
                {
                    ledRgbControl(0, 0, 0);
                }
                break;

            case EVENT_LED_Color:
                currentDataPoint.valueLED_Color = dataPointPtr->valueLED_Color;
                GIZWITS_LOG("Evt: EVENT_LED_Color %d\n", currentDataPoint.valueLED_Color);
                switch(currentDataPoint.valueLED_Color)
                {
                    case LED_Color_VALUE0:
                        ledRgbControl(currentDataPoint.valueLED_R, currentDataPoint.valueLED_G,
currentDataPoint.valueLED_B);
                        break;
                    case LED_Color_VALUE1:
                        ledRgbControl(254, 254, 0);
                        break;
                    case LED_Color_VALUE2:
                        ledRgbControl(254, 0, 70);
                        break;
                    case LED_Color_VALUE3:
                        ledRgbControl(238, 30, 30);
                        break;
                }
            }
        }
    }
}

```

```

        default:
            break;
    }
    break;

case EVENT_LED_R:
    currentDataPoint.valueLED_R = dataPointPtr->valueLED_R;
    GIZWITS_LOG("Evt:EVENT_LED_R %d\n", currentDataPoint.valueLED_R);
    ledRgbControl(currentDataPoint.valueLED_R, currentDataPoint.valueLED_G,
currentDataPoint.valueLED_B);
    break;
case EVENT_LED_G:
    currentDataPoint.valueLED_G = dataPointPtr->valueLED_G;
    GIZWITS_LOG("Evt:EVENT_LED_G %d\n", currentDataPoint.valueLED_G);
    ledRgbControl(currentDataPoint.valueLED_R, currentDataPoint.valueLED_G,
currentDataPoint.valueLED_B);
    break;
case EVENT_LED_B:
    currentDataPoint.valueLED_B = dataPointPtr->valueLED_B;
    GIZWITS_LOG("Evt:EVENT_LED_B %d\n", currentDataPoint.valueLED_B);
    ledRgbControl(currentDataPoint.valueLED_R, currentDataPoint.valueLED_G,
currentDataPoint.valueLED_B);
    break;
case EVENT_Motor_Speed:
    currentDataPoint.valueMotor_Speed = dataPointPtr->valueMotor_Speed;
    GIZWITS_LOG("Evt:EVENT_Motor_Speed %d\n", currentDataPoint.valueMotor_Speed);
    motorStatus(currentDataPoint.valueMotor_Speed);
    break;

case WIFI_SOFTAP:
    break;
case WIFI_AIRLINK:
    break;
case WIFI_STATION:
    break;
case WIFI_CON_ROUTER:
    ledRgbControl(0, 0, 0);
    break;
case WIFI_DISCON_ROUTER:

    break;
case WIFI_CON_M2M:

    break;
case WIFI_DISCON_M2M:
    break;
case WIFI_RSSI:
    GIZWITS_LOG("RSSI %d\n", wifiData->rssi);
    break;
case TRANSPARENT_DATA:
    GIZWITS_LOG("TRANSPARENT_DATA \n");
    //user handle , Fetch data from [data] , size is [len]

    break;

```

```

        case WIFI_NTP:
            GIZWITS_LOG("WIFI_NTP : [%d-%d-%d %02d:%02d:%02d][%d] \n", ptime->year, ptime->month, ptime->day, ptime->hour, ptime->minute, ptime->second, ptime->ntp);
            break;
        case MODULE_INFO:
            GIZWITS_LOG("MODULE INFO ...\n");
            #if MODULE_TYPE
            GIZWITS_LOG("GPRS MODULE ...\n");
            //Format By gprsInfo_t
            #else
            GIZWITS_LOG("WIF MODULE ...\n");
            //Format By moduleInfo_t
            GIZWITS_LOG("moduleType : [%d] \n", ptModuleInfo->moduleType);
            #endif
            break;
        default:
            break;
    }
}

return 0;
}

```

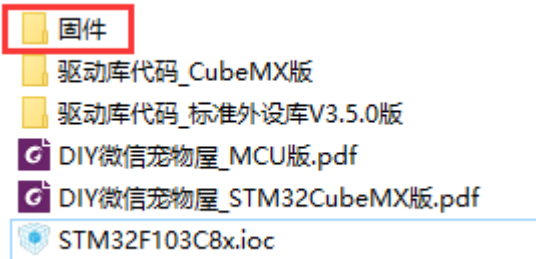
6 编译烧录固件并进行测试

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 GIZWITS website's 'Download Center' (下载中心) page. The left sidebar contains navigation links for hardware development resources, client development resources, development and testing 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 smart hardware product. Below the description, there is a table of firmware links for different GoKit development boards, with the first three rows highlighted by a red box.

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：**GoKit_MCU_STM32_XX**测试固件及开发资源**XX\固件\mcu_stm32f103c8x.hex**



MCU版的固件烧录方式可查看 [ST底板程序编译及下载教程](#) 一文。

微信宠物屋的操作方式可查看 [GoKit使用说明书](#) 一文

7 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。

