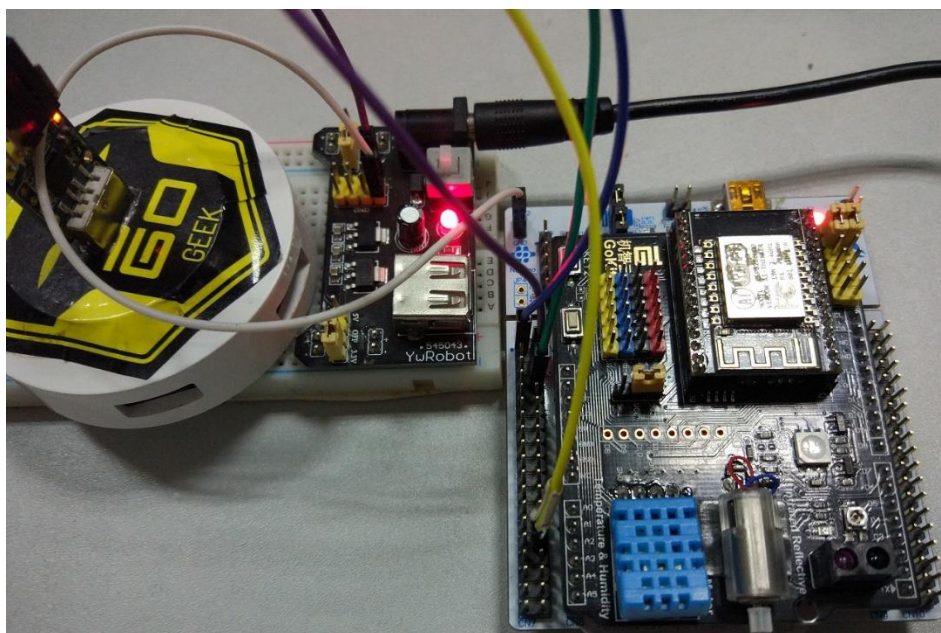


一、概览

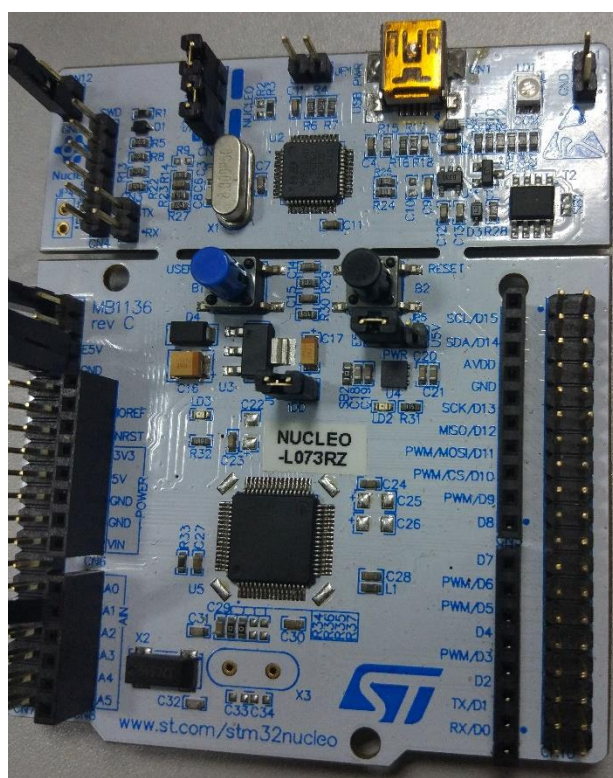


利全球领先的用第三方云平台-机智云，打造了基于超低功耗 Nucleo-STM32L073 的物联网应用系统，实现的功能有：

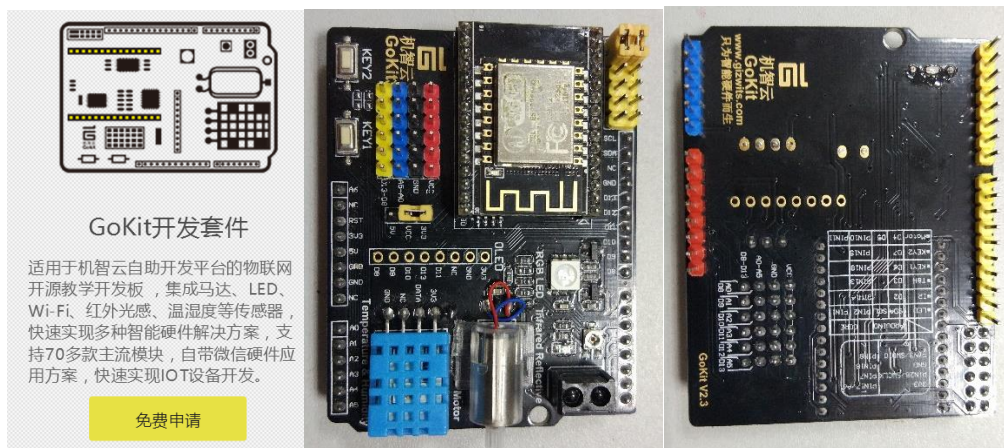
- a) 远程手机 APP 控制 RGB LED 的颜色（红、绿、蓝）；
- b) 自带 3 种混合色，黄色、紫色、粉色，一键切换所需颜色；
- c) 远程手机 APP 控制直流电机的正反转；
- d) 远程手机 APP 显示温湿度；
- e) 红外传感器障碍物检测；



二、硬件介绍



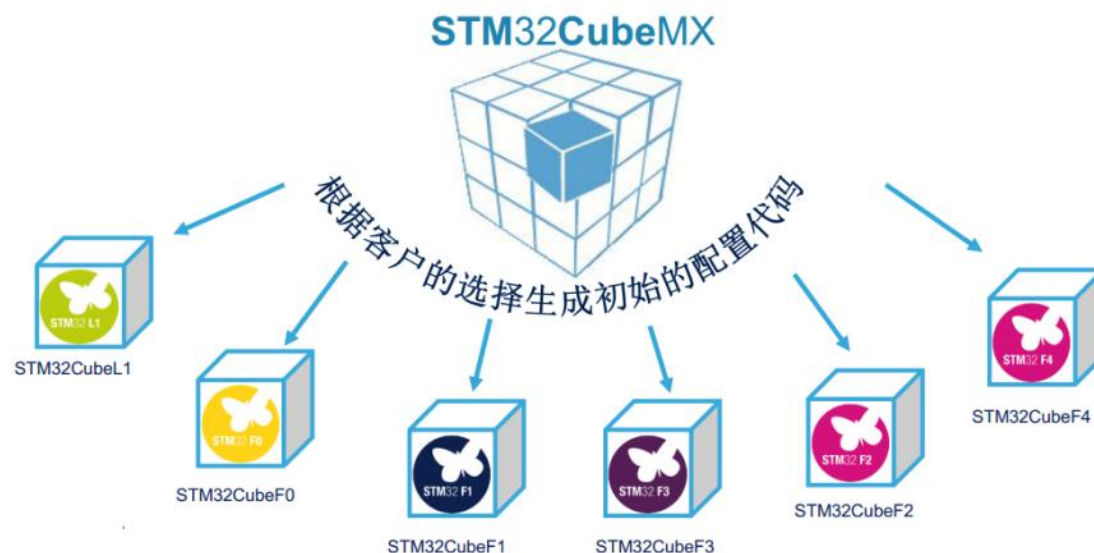
NUCLEO-L073RZ 是 ST 公司推出的一款针对 STM32L0 系列设计的 Cortex-M0 开发板，支持 mbed，兼容 Arduino，同时还提供 ST Morpho 扩展排针，可连接微控制器的所有周边外设。开发板基于 STM32L073RZT6 设计，开发板还集成了 ST-LINK/V2-1 仿真下载器（但仅对外提供 SWD 接口），免除您另外采购仿真器或下载器的麻烦。并且具备 Arduino 接口，可接入 Arduino 巨大生态系统的各种 Shield 扩展板，让您能够轻松快速增加特殊功能。



机智云是智能硬件时代专为硬件提供后台支持的云服务平台，服务的内容主要包括了统计分析、数据安全、远程管理、软件升级等。GoKit 是机智云推出的一套智能硬件开发板，Gokit 扩展板采用 Arduino UNO 接口，板载 ESP8266-12F WiFi 模块、温湿度传感器 DHT11、2 个按键、障碍物红外检测模块、微型直流电机、RGB 5050 全彩 LED。

三、软件系统

开发工具：STM32CubeMX 和 IAR for ARM，IAR 不做介绍，这里介绍下 ST 推广的全新开发工具 STM32CubeMX。



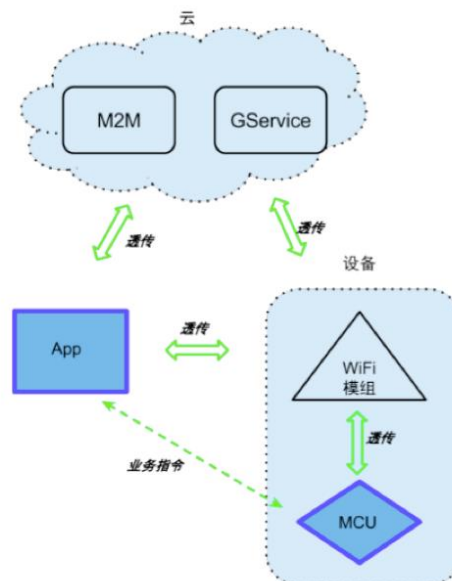
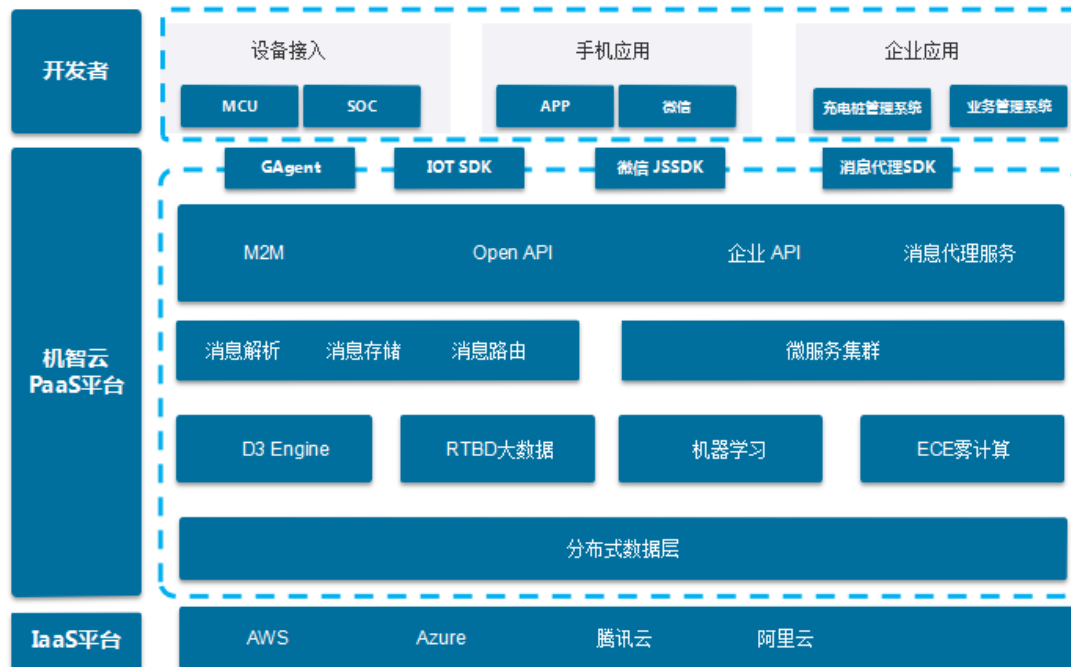
STM32CubeMX 是 ST 意法半导体近几年来大力推荐的 STM32 芯片图形化配置工具，允许用户使用图形化向导生成 C 初始化代码，可以大大减轻开发工作，时间和费用。

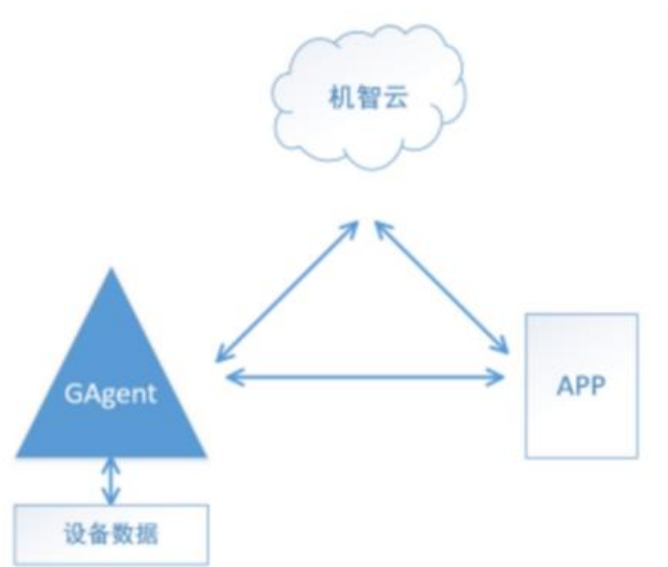
STM32CubeMX 覆盖了 STM32 全系列芯片。它具有如下特性：

- ① 直观的选择 MCU 型号，可指定系列、封装、外设数量等条件
- ② 微控制器图形化配置
- ③ 自动处理引脚冲突
- ④ 动态设置时钟树，生成系统时钟配置代码
- ⑤ 可以动态设置外围和中间件模式和初始化

- ⑥ 功耗预测
- ⑦ C 代码工程生成器覆盖了 STM32 微控制器初始化编译软件，如 IAR，KEIL，GCC。
- ⑧ 可以独立使用或者作为 Eclipse 插件使用

机智云智能设备接入原理：





硬件厂商对硬件智能化的第一步是让设备能够联网，GAgent 是机智云提供的一款兼容国内主流 Wi-Fi 模块、移动网络模块的模组系统。硬件厂商只需理解如何与 GAgent 进行通讯即可。GAgent 主要的作用是数据转发，是设备数据、机智云、应用端（APP）的数据交互桥梁。可将 GAgent 移植到 WiFi 模组、GPRS 模组、PC 端等。目前机智云提供由机智云移植的 WiFi 模组对应固件有：汉枫 LPB100、乐鑫 8266、汉枫 LPT120、高通 4004 、RealTek 8711AM 、庆科 3162 等。简单的说就是，WiFi 模块里面烧录了一个叫 GAgent 的固件，然后这个 WiFi 模块对用户的 MCU 就是串口透传了，复杂的协议、交互都给封装了。

四、云端配置



我们在机智云开发者中心 <http://dev.gizwits.com/zh-cn/developer/> 创建一个名为 NucleoL073GoKitS 的产品并添加数据点，数据点是设备产品的功能的抽象，用于描述产品功能及其参数。创建数据点后，设备与云端通讯的数据格式即可确定，设备、机智云可以相

互识别设备与机智云互联互通的数据。因为我们手机 APP 的功能需求是：

- 单独控制 RGB LED 的颜色（红、绿、蓝）；
- 自带 3 种混合色，黄色、紫色、粉色，保证能一键切换所需颜色；
- 远程手机 APP 控制直流电机的正反转；
- 远程手机 APP 显示温湿度传感器 DHT11 采集的温湿度；
- 红外传感器障碍物检测；

把以上进行抽象，依次为数值量下发控制、枚举量下发控制、数值量下发控制、数值量上报、布尔值上报。

所以建立如下数据点（这里可以利用官方提供的模板）：



创建后的数据点如下：

个人项目 / NucleoL073GoKitS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

MCU开发

应用配置

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

数据点

定义数据点教程

+ 新建数据点

管理 -

显示名称: 开启/...	标识名: LED_...	读写类型: 可写	数据类型: 布尔值	<input checked="" type="checkbox"/>	<input type="checkbox"/>
备注: 无					
显示名称: 设定L...	标识名: LED_C...	读写类型: 可写	数据类型: 枚举	<input checked="" type="checkbox"/>	<input type="checkbox"/>
枚举范围: 0.自定...					
备注: 无					
显示名称: 设定L...	标识名: LED_R	读写类型: 可写	数据类型: 数值	<input checked="" type="checkbox"/>	<input type="checkbox"/>
数据范围: 0 - 254	分辨率: 1	增量: 0			
备注: 无					
显示名称: 设定L...	标识名: LED_G	读写类型: 可写	数据类型: 数值	<input checked="" type="checkbox"/>	<input type="checkbox"/>
数据范围: 0 - 254	分辨率: 1	增量: 0			

五、MCU 端编程

在云端创建产品，建立好数据点后，即可自动生成多种方案的工程，还可以下载自动生成的手机 APP 或者进行在线虚拟调试，机智云为广大开发者提供一系列简便的开发工具：

虚拟设备

设备日志

开发向导

服务

MCU开发

应用配置

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

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。

进入MCU开发

如果想了解接入机智云的串口通信协议，可下载 [《NucleoL073GoKitS- 机智云接入串口通信协议文档》](#)。此外，也提供功能参数的 [《NucleoL073GoKitS-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。

App 开发资源

机智云根据你定义的功能参数，会自动生成对应的App安装包以及源码。除此之外，你还可以在不需要编写代码的情况下对App进行自定义。

下载 SDK

如果想进一步了解如何让用的SDK开发App，可查看 [《WiFi产品iOS SDK》](#)、[《WiFi产品Android SDK》](#)。

微信开发资源

如利用虚拟设备，在没有搭建硬件设备平台时可模拟设备与云端的交互：

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

MCU开发

应用配置

产测工具

固件升级 (OTA)

统计

概览

新增上线

活跃设备

虚拟设备



通过虚拟设备模拟真实设备上报数据的行为，可以快速验证接口功能的开发。

用您开发的app或[下载Demo App](#) 绑定虚拟设备，即可对虚拟设备进行远程控制及查看通讯日志。

[启动虚拟设备](#)

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

MCU开发

应用配置

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

在线调试 / 虚拟设备

调试提示 Product Key: XXXXXXXXXXXXXXXXXXXXXXXXXXXX [收起](#)

1、通过您开发的app或Demo app扫描二维码绑定虚拟设备。 [显示二维码](#)

2、其它绑定方式，可以通过提供指定的did与passcode绑定。本虚拟设备的did：2wfyrrgXh3kx5CPoi4dCHP，passcode：123456

模拟设备上报数据	通信日志
<div>设定LED绿色值 (LED_G) $0 = 1 * \boxed{0} + 0$</div> <div>0 <div></div> 254</div>	<div>设备上线 系统消息：成功连上云端</div> <div>2016-12-27 18:06:19</div>
<div>设定LED蓝色值 (LED_B) $0 = 1 * \boxed{0} + 0$</div> <div>0 <div></div> 254</div>	<div>正在登录服务器...</div> <div>2016-12-27 18:05:41</div>
<div>设定电机转速 (Motor_Spee... $-5 = 1 * \boxed{0} + (-5)$</div>	

在服务栏目->MCU 开发，我们选择独立 MCU+WiFi 方案：

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

MCU开发

应用配置

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

SOC方案

硬件平台: stm32f103c8x

生成代码包

点击生成代码包即可；

机智云
Gizwits

开发者中心

文档中心

下载中心

社区

qizheng@qizheng.com

2

个人项目 / NucleoL073GoKitS

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

MCU开发

应用配置

产测工具

MCU开发

MCU代码生成结果

硬件方案: mcu

硬件平台: stm32f103c8x

下载

修改

下载生成的工程，解压，其文件组织目录如下：

45f4bb6084594002b6364e5248fe080f_mcu_stm32f103c8x_20161223142829_pms3j6

名称	修改日期	类型	大小
Gizwits	2016/12/23 22:28	文件夹	
Hal	2016/12/23 22:28	文件夹	
Lib	2016/12/23 22:28	文件夹	
Project	2016/12/23 22:28	文件夹	
User	2016/12/23 22:28	文件夹	
Changelog	2016/11/30 15:09	文本文档	1 KB
MCU_STM32F103C8x_API介绍_V0.2	2016/11/30 16:13	Foxit Reader Plus P...	270 KB

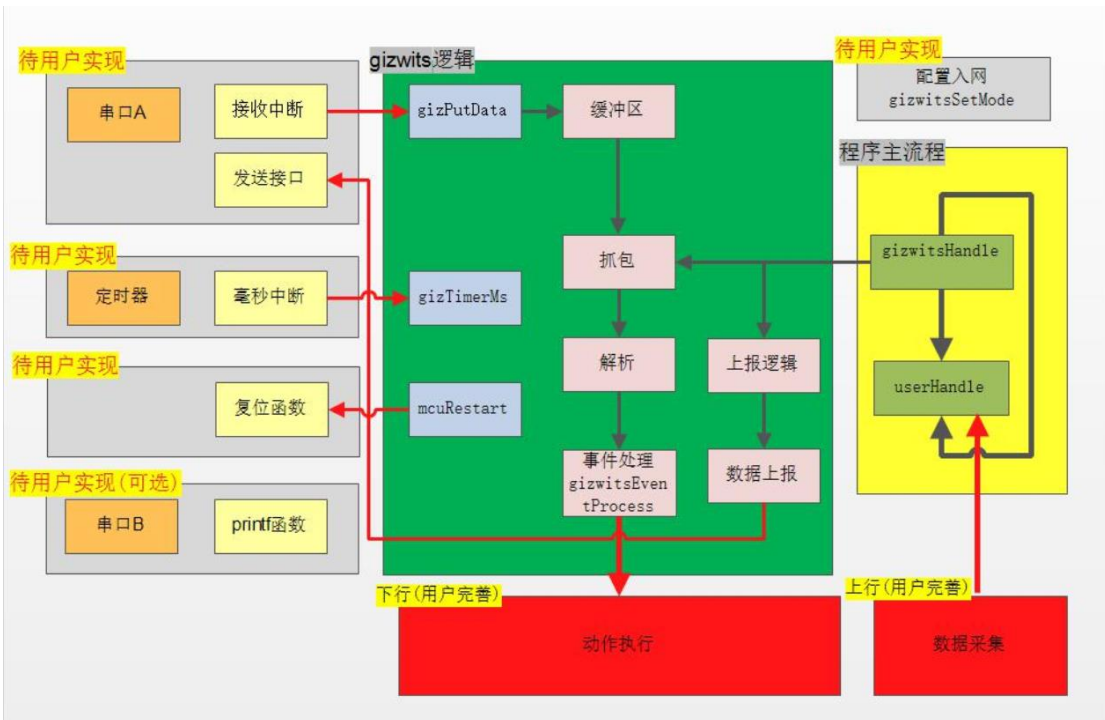
注意 Gizwits 文件夹下包含 4 个文件，这些是机智云协议层的，已包含云端创建的数据点信息并留出用户接口：

» 45f4bb6084594002b6364e5248fe080f_mcu_stm32f103c8x_20161223142829_pms3j6 » Gizwits			
名称	修改日期	类型	大小
 gizwits_product	2016/12/23 14:28	C 文件	9 KB
 gizwits_product	2016/11/30 10:09	H 文件	4 KB
 gizwits_protocol	2016/12/23 14:28	C 文件	51 KB
 gizwits_protocol	2016/12/23 14:28	H 文件	23 KB

另外需要注意 User 目录下的 main.c 文件，这里包含了程序框架。

└─ Gizwits	//机智云协议层,包括通信协议和用户事件处理
└─┬─ gizwits_product.c	
└─┬─ gizwits_product.h	
└─┬─ gizwits_protocol.c	
└─┬─ gizwits_protocol.h	
└─ Hal	//基础驱动层,包括按键、串口(打印)、看门狗
└─┬─ Hal_key	
└─┬─┬─ hal_key.c	
└─┬─┬─ hal_key.h	
└─┬─ Hal_Uart	
└─┬─┬─ hal_uart.c	
└─┬─┬─ hal_uart.h	
└─┬─ Hal_Watchdog	
└─┬─┬─ hal_watchdog.c	
└─┬─┬─ hal_watchdog.h	
└─ Lib	//STM32F10x_StdPeriph_Lib_V3.5.0官方库
└─┬─ CMSIS	
└─┬─ STM32F10x_StdPeriph_Driver	
└─ Project	//工程文件管理层
└─┬─ mcu_stm32f103x.uvopt	
└─┬─ mcu_stm32f103x.uvproj	
└─ User	//APP层,main为程序入口函数
└─┬─ main.c	
└─┬─ stm32f10x_conf.h	
└─┬─ stm32f10x_it.c	
└─┬─ stm32f10x_it.h	

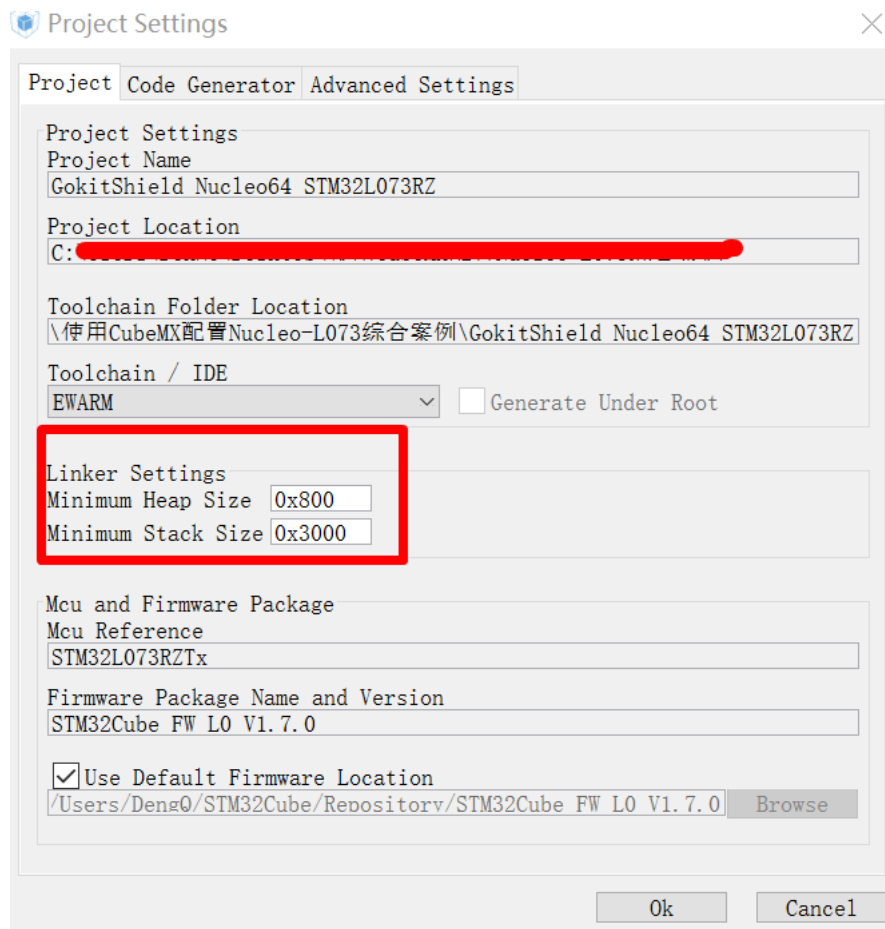
这个工程（基于 STM32 标准库函数且 IDE 为 Keil for ARM）只是一个框架，但包含了和机智云通信协议层，用户只需要添加自己的驱动代码和业务逻辑，以本例来说就是我们需要实现 RGB LED 的驱动，直流电机的驱动，温湿度传感器的驱动，红外传感器的驱动，以及收到手机 APP 发的指令后干什么，软件逻辑如下：



所以我们只需简单的移植一下，通过此图知道要把任意 MCU 通过 WIFI 模块连上机智云，只需实现 3 个简单的接口函数：串口发送、中断接收 1 字节数据，定时器 1ms 中断。

1	GoKit V2.3 Shield			NUCLEO-L073RZ		
2	板卡丝印	Arduino UNO标准接口	功能说明	板卡丝印	MCU引脚	功能说明
3	A6	IOREF	无任何连接	NC	IOREF	
4	NC	RESET	无任何连接	RESET	NRST	
5	3V3	3.3V	3.3V电源用于板载IC	3.3V		3.3V电源输出
6	5V	5V	5V电源用于RGB LED	5V		5V电源输出
7	GND	GND		GND		
8	GND	GND		GND		
9	NC	VIN	无任何连接	VIN		
10	A0	A0	RGB LED电源开关，高电平使能	A0	PA0	
11	A1	A1		A1	PA1	
12	A2	A2		A2	PA4	
13	A3	A3		A3	PB0	
14	A4	A4		A4	PC1	
15	A5	A5		A5	PC0	
16	SCL	SCL/D15	RGB LED驱动IC的时钟线	SCL/D15	PB8	
17	SDA	SDA/D14	RGB LED驱动IC的数据线	SDA/D14	PB9	
18	NC	AREF	无任何连接	AVDD		
19	GND	GND		GND		
20	D13	D13		D13	PA5	用户绿色LD2,高电平点亮
21	D12	D12		D12	PA6	
22	D11	D11		D11	PA7	
23	D10	D10		D10	PB6	
24	D9	D9		D9	PC7	
25	D8	D8		D8	PA9	

如上图根据 Nucleo-STM32L073 和机智云 Gokit V2.3 扩展板原理图，确定了引脚分配。这里由于板载 ST-LINK 虚拟串口和 Arduino UNO 接口的串口有冲突，需要对 Nucleo 做点手术：

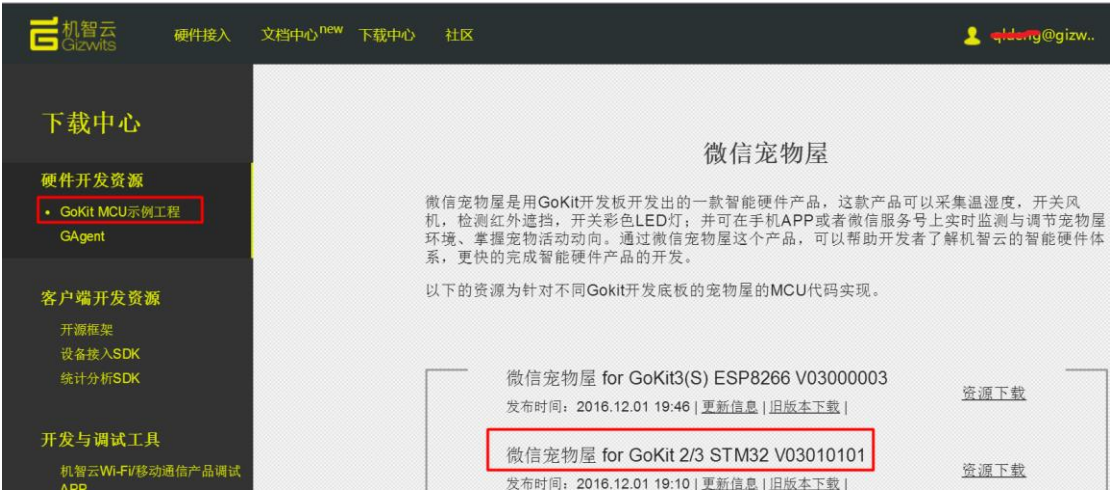


再把云端生成工程的 GizWits 目录拷贝到 IAR 工程中，移植硬件驱动即可。

GokitShield_Nucleo64_STM32L073RZ			
名称	修改日期	类型	大小
Drivers	2016/12/23 22:24	文件夹	
EWARM	2016/12/25 17:32	文件夹	
Gizwits	2016/12/24 21:47	文件夹	
Hal	2016/12/24 20:21	文件夹	
Inc	2016/12/25 17:32	文件夹	
Src	2016/12/25 17:32	文件夹	
.mxproject	2016/12/25 17:32	MXPROJECT 文件	7 KB
GokitShield_Nucleo64_STM32L073RZ	2016/12/27 20:25	STM32CubeMX	8 KB
GokitShield_Nucleo64_STM32L073RZ	2016/12/27 20:25	Foxit Reader Plus P...	178 KB
GokitShield_Nucleo64_STM32L073RZ	2016/12/27 20:25	文本文档	2 KB
GokitShield_Nucleo64_STM32L073RZ > Gizwits			
名称	修改日期	类型	大小
gizwits_product	2016/12/24 21:47	C 文件	8 KB
gizwits_product	2016/12/24 21:43	H 文件	2 KB
gizwits_protocol	2016/12/24 20:20	C 文件	53 KB
gizwits_protocol	2016/12/24 20:45	H 文件	23 KB

到此完成了机智云协议层的拷贝，下面拷贝硬件驱动层：

我们在机智云官网下载微信宠物屋源码：



解压，找到 Hal 文件夹，如下图，里面包含了外设驱动源码，但是这些是基于 STD 库函数且 MCU 是 STM32F103C8T6：

gokit_mcu_stm32_V03010101_2016120118 > Hal

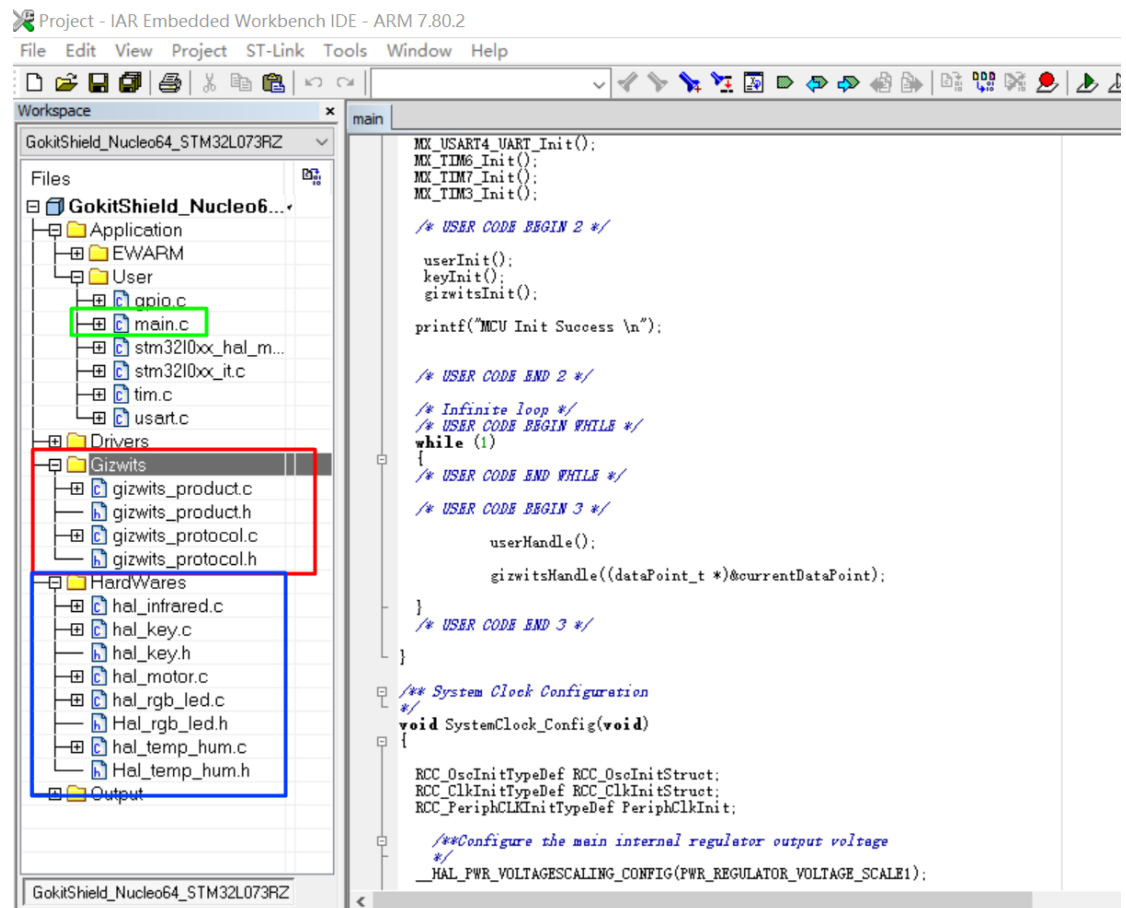
名称	修改日期	类型
Hal_infrared	2016/12/23 22:37	文件夹
Hal_key	2016/12/23 22:37	文件夹
Hal_led	2016/12/23 22:37	文件夹
hal_motor	2016/12/23 22:37	文件夹
Hal_rgb_led	2016/12/23 22:37	文件夹
hal_temp_hum	2016/12/23 22:37	文件夹
Hal_Usart	2016/12/23 22:37	文件夹
Hal_Watchdog	2016/12/23 22:37	文件夹

而我们 STM32CubeMX 软件生成的是基于 HAL 库函数且 MCU 是 STM32L073RZ，所以需要修改这些硬件驱动，移植这里就不介绍了，大家下载附件查看即可，我这里把这些驱动文件.c 和.h 放在一个文件夹 Hal:

GokitShield_Nucleo64_STM32L073RZ > Hal

名称	修改日期	类型	大小
hal_infrared	2016/12/24 18:05	C 文件	1 KB
Hal_infrared	2016/12/24 18:05	H 文件	1 KB
hal_key	2016/12/24 20:21	C 文件	8 KB
hal_key	2016/12/24 18:02	H 文件	2 KB
hal_motor	2016/11/7 14:21	C 文件	2 KB
Hal_motor	2016/12/24 18:02	H 文件	1 KB
hal_rgb_led	2016/9/26 8:42	C 文件	3 KB
Hal_rgb_led	2016/12/24 18:01	H 文件	1 KB
hal_temp_hum	2016/12/24 18:34	C 文件	6 KB
Hal_temp_hum	2016/12/24 18:14	H 文件	1 KB

如下图 IAR 工程，红框是机智云协议层，蓝框是硬件驱动层。



需要修改的 3 个接口和需要添加的用户逻辑位于 gizwits_product.c, 前文提到的三个接口函数是定时器 1ms 中断, 串口发送和中断接收如下:

Workspace

GokitShield_Nucleo64_STM32L073RZ

Files

- GokitShield_Nucleo64...
- Application
 - EWARM
 - User
 - gpio.c
 - main.c
 - stm32l0xx_hal_m...
 - stm32l0xx_it.c
 - tim.c
 - usart.c
- Drivers
 - Gizwits
 - gizwits_product.c
 - gizwits_product.h
 - gizwits_protocol.c
 - gizwits_protocol.h
 - HardWares
 - hal_infrared.c
 - hal_key.c
 - hal_key.h
 - hal_motor.c
 - hal_rgb_led.c
 - hal_rgb_led.h
 - hal_temp_hum.c
 - hal_temp_hum.h
 - Output

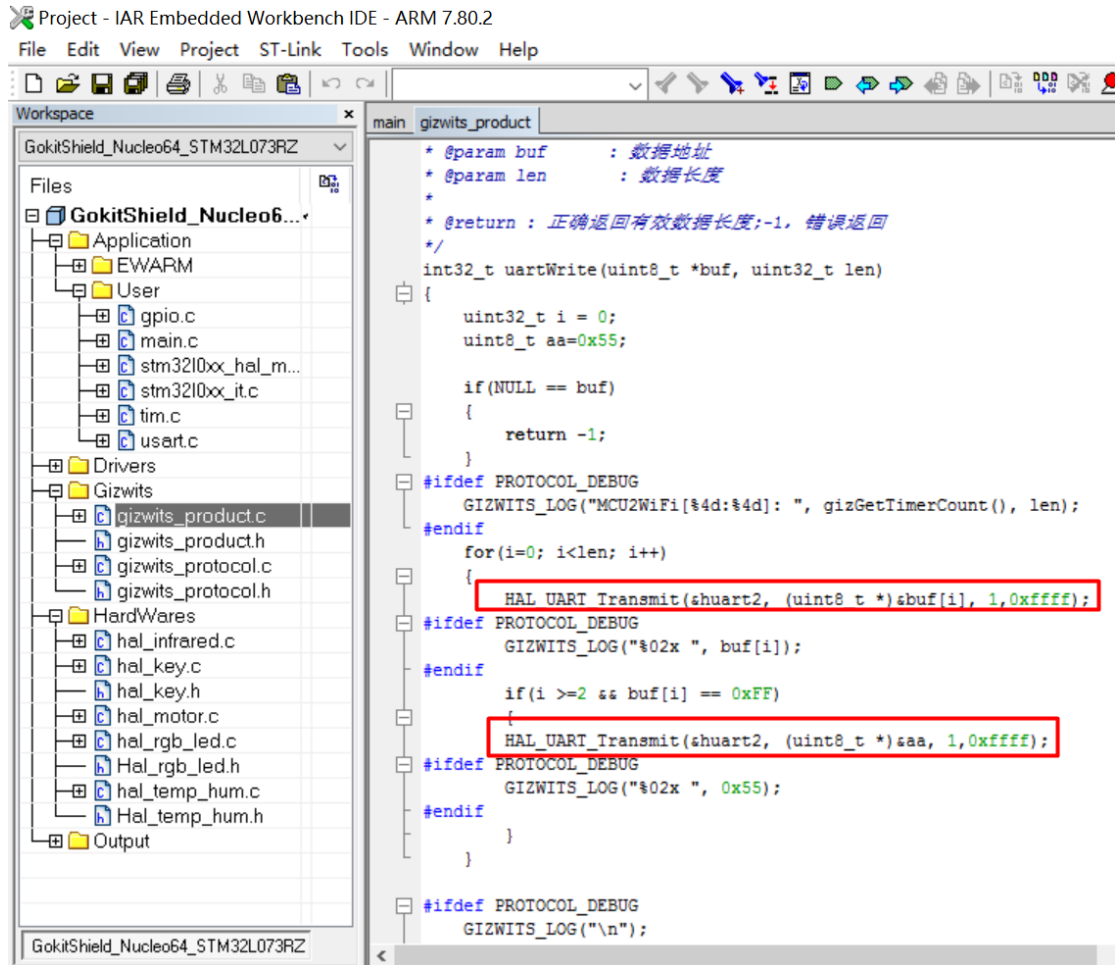
main gizwits_product

```
/**@ */  
/**  
 * @brief 定时器TIM中断处理函数  
  
 * @param none  
 * @return none  
 */  
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)  
{  
    if(htim->Instance ==TIM6) //tim6 1ms中断, 作为MCU和WIFI模组的心跳用  
    {  
        gizTimerMs();  
    }  
    if(htim->Instance ==TIM7) //tim7 1ms中断, 按键检测逻辑  
    {  
        keyHandle((keyTypeDef_t *)&keys);  
    }  
}  
/**  
 * @brief 定时器TIM定时器初始化  
  
 * @param none  
 * @return none  
 */  
void timerInit(void)  
{  
    HAL_TIM_Base_Start_IT(&htim6);  
    HAL_TIM_Base_Start_IT(&htim7);  
}  
  
/**  
 * @brief USART2串口接收中断函数
```

The screenshot shows the IAR Embedded Workbench IDE interface. On the left, the 'Workspace' pane displays a project tree for 'GokitShield_Nucleo64_STM32L073RZ'. The tree includes folders for 'Application', 'EWARM', 'User', 'Drivers', 'Gizwits', 'HardWares', and 'Output'. The 'Gizwits' folder is expanded, showing files like 'gizwits_product.c' and 'gizwits_protocol.c'. The 'main' tab is active, displaying the source code for 'gizwits_product.c'. The code is written in C and includes comments in Chinese. A red rectangle highlights a specific function definition.

```
/**
 * @brief USART2 串口接收中断函数
 * 接收功能, 用于接收与WiFi模组间的串口协议数据
 * @param none
 * @return none
 */
extern uint8_t RxData;
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *UartHandle)
{
    if(UartHandle->Instance == USART2 )
    {
        gizPutData(&RxData, 1);
        // HAL_UART_Transmit(&huart2, &RxData, 1, 0xffff);
        // LD2_Toggle();
        HAL_UART_Receive_IT(&huart2, &RxData, 1);
    }
}

/**
 * @brief 串口写操作, 发送数据到WiFi模组
 *
 * @param buf      : 数据地址
 * @param len      : 数据长度
 *
 * @return : 正确返回有效数据长度;-1, 错误返回
 */
int32_t uartWrite(uint8_t *buf, uint32_t len)
{
    uint32_t i = 0;
```



上电，打开串口调试助手可以查看日志：



其它细节不再赘述，代码修改移植细节请参考社区系列教程：

<http://club.gizwits.com/thread-3914-1-1.html>

总结：本文介绍了使用机智云自助开发平台快速创建智能产品的过程，并给出了移植机智云微信宠物屋 SDK 到 Nucleo-STM32L073

的过程。利用机智云强大的代码生成工具，我们只需简单的把用户代码填充到机智云为我们搭建的框架即可。

附演示视频链接：<http://player.youku.com/player.php/sid/XMTg4NzQ3MjM0MA==/v.swf>

完整工程源码，资料请戳：<http://club.gizwits.com/thread-4467-1-1.html>