



机智云

只为智能硬件而生的云

机智云

2005团队成立，提供互联网服务

2009年第一款互联网化的硬件产品

2010年机智云上线

2012年移动应用开发组件产品化

2013年第100款软硬件接合App上架

2014年机智云平台接入设备超过200万台

2015年开启物联网新纪元

物联网

物联网这个概念，在美国早在1999年就提出来了。当时叫传感网。

其定义是：

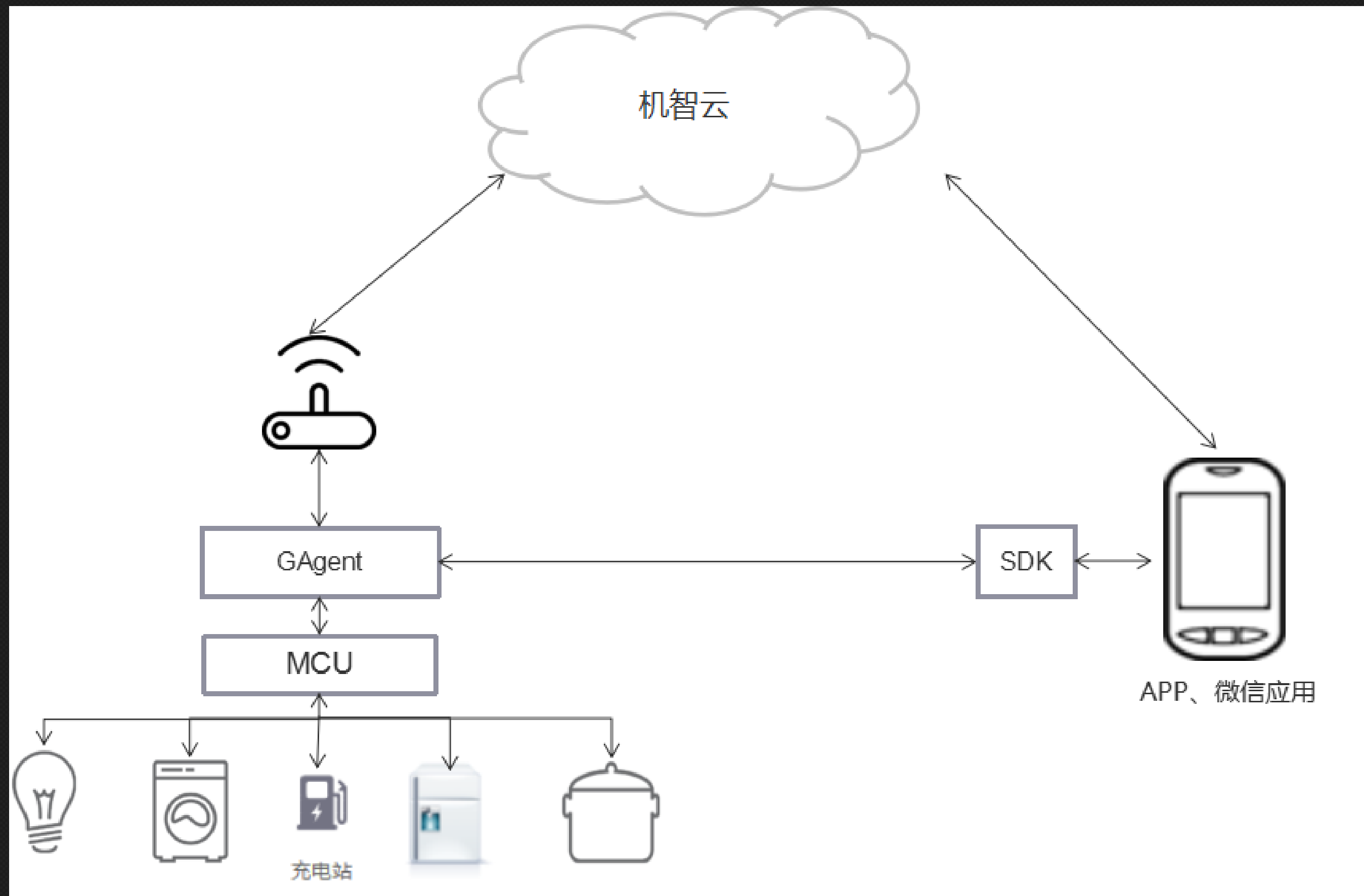
通过射频识别（RFID）、红外感应器、全球定位系统、激光扫描器等信息传感设备，按约定的协议，把任何物品与互联网相连接，进行信息交换和通信，以实现智能化识别、定位、跟踪、监控和管理的一种网络概念。

“物联网概念”是在“互联网概念”的基础上，将其用户端延伸和扩展到任何物品与物品之间，进行信息交换和通信的一种网络概念。

物联网金三角

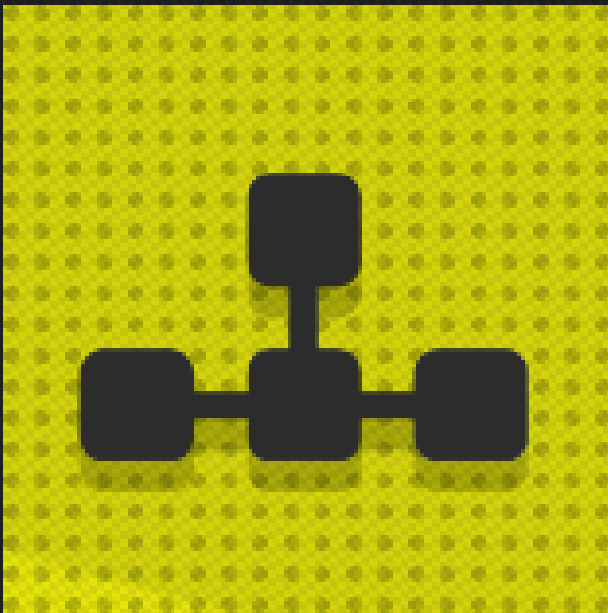


机智云

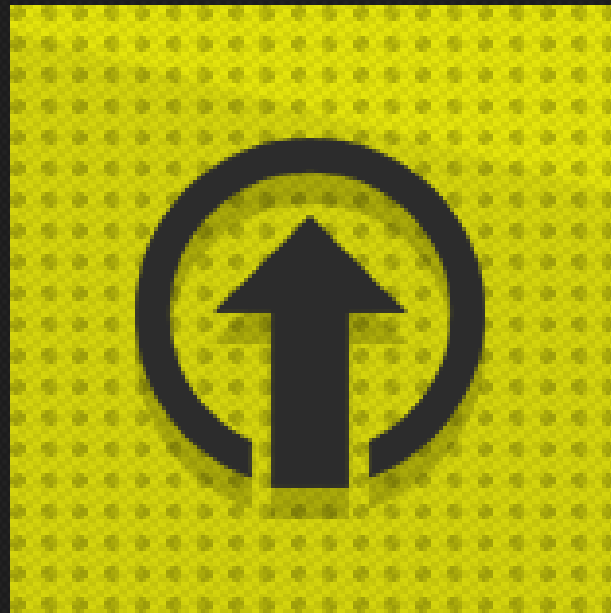


Free

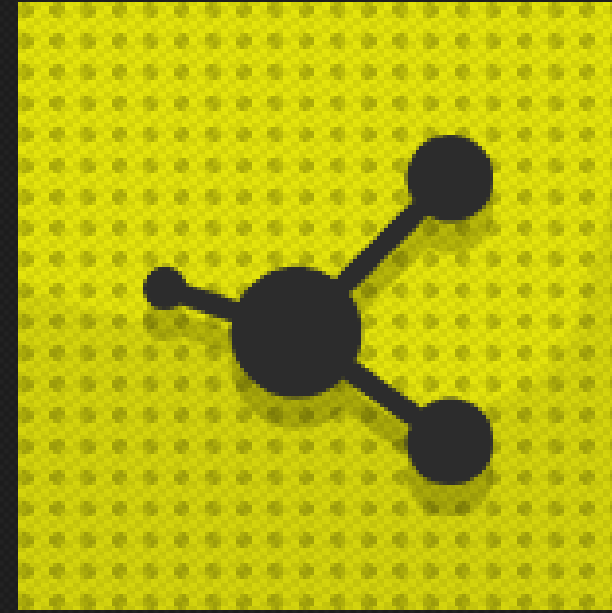
机智云 物联网开发平台



硬件接入



OTA升级

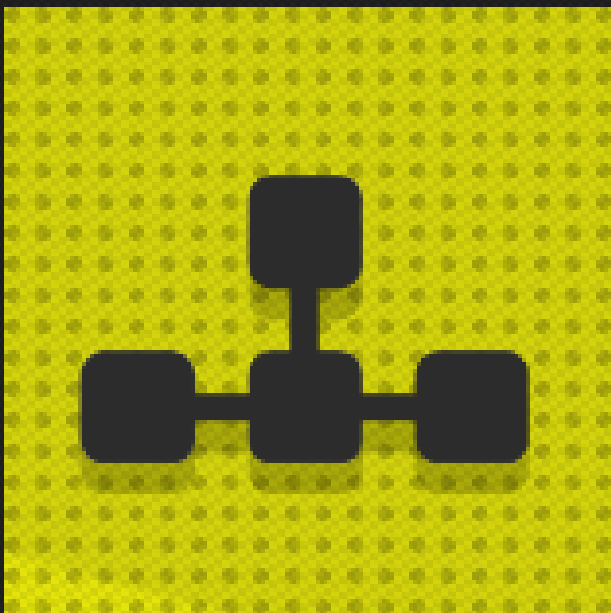


硬件社交



统计分析

M2M接入服务

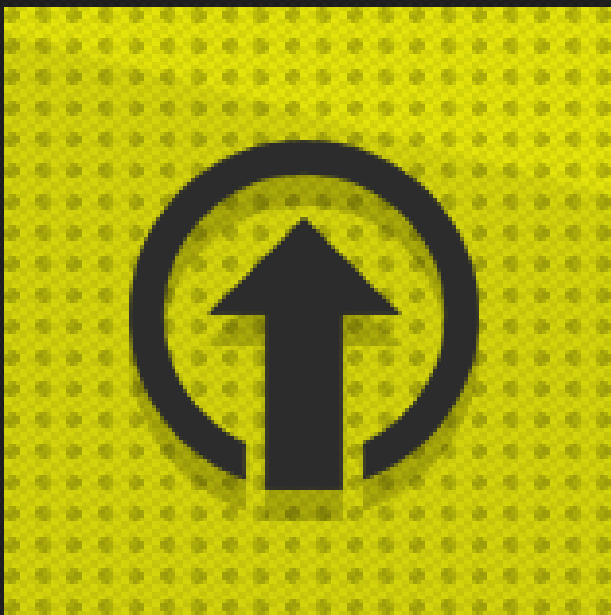


WiFi设备路由器接入(Onboarding), 远程监控和管理, 数据存储与分析等物联网基础服务

支持市面上主流的**Wi-Fi**模块, 支持所有量产的路由器接入(Onboarding), 提供高效的数据存储以及统计分析服务, 支持云端对接第三方的数据存储服务

通过机智云后台制定设备通信协议, 直接生成产品开发所需的移动应用**SDK**、**Demo App**、设备模拟器等, 让你的设备快速实现与云端、手机端的连接与远程控制。

OTA升级服务

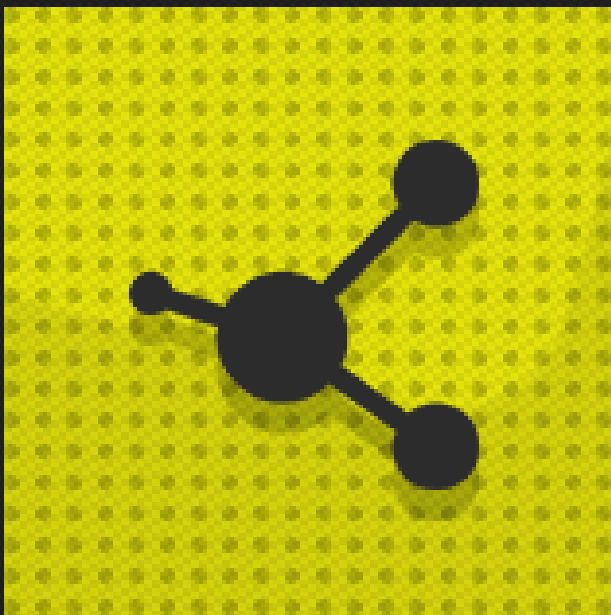


满足设备固件远程升级，支持多种定向升级策略。解决设备出厂后的持续升级需求

针对每一次设备固件升级，可设定按批次、按地域、按原固件版本等条件进行远程推送，也可以设定一个时间段进行定时升级，让固件升级的同时不影响用户使用。

随时跟踪固件推送情况，可随时停止推送，也可以对同一个固件版本进行多次推送。如果发现新固件版本有问题也可以及时的重新推送原来的版本。

硬件社交化服务



支持WiFi和蓝牙设备的微信客户端接入，定制微信硬件接口通信协议以及云端服务器实现

WiFi设备通过机智云的设备虚拟化服务，可以轻松创建轻应用并接入社交网络，发挥创意即可让设备在社交网络中进行快速的传播。

蓝牙设备使用最新的微信硬件接口，通过机智云后台数据处理，无需app即可实现智能硬件社交化。例如，你可以跟不在一个城市的好友一起练习高尔夫球，创建比赛邀请你的朋友参与

Free

统计分析服务



设备为智能硬件定制的统计分析服务。只需添加一行代码的集成工作，多维度的设备使用和用户行为数据就能自动统计出来

特色一：设备上线情况统计

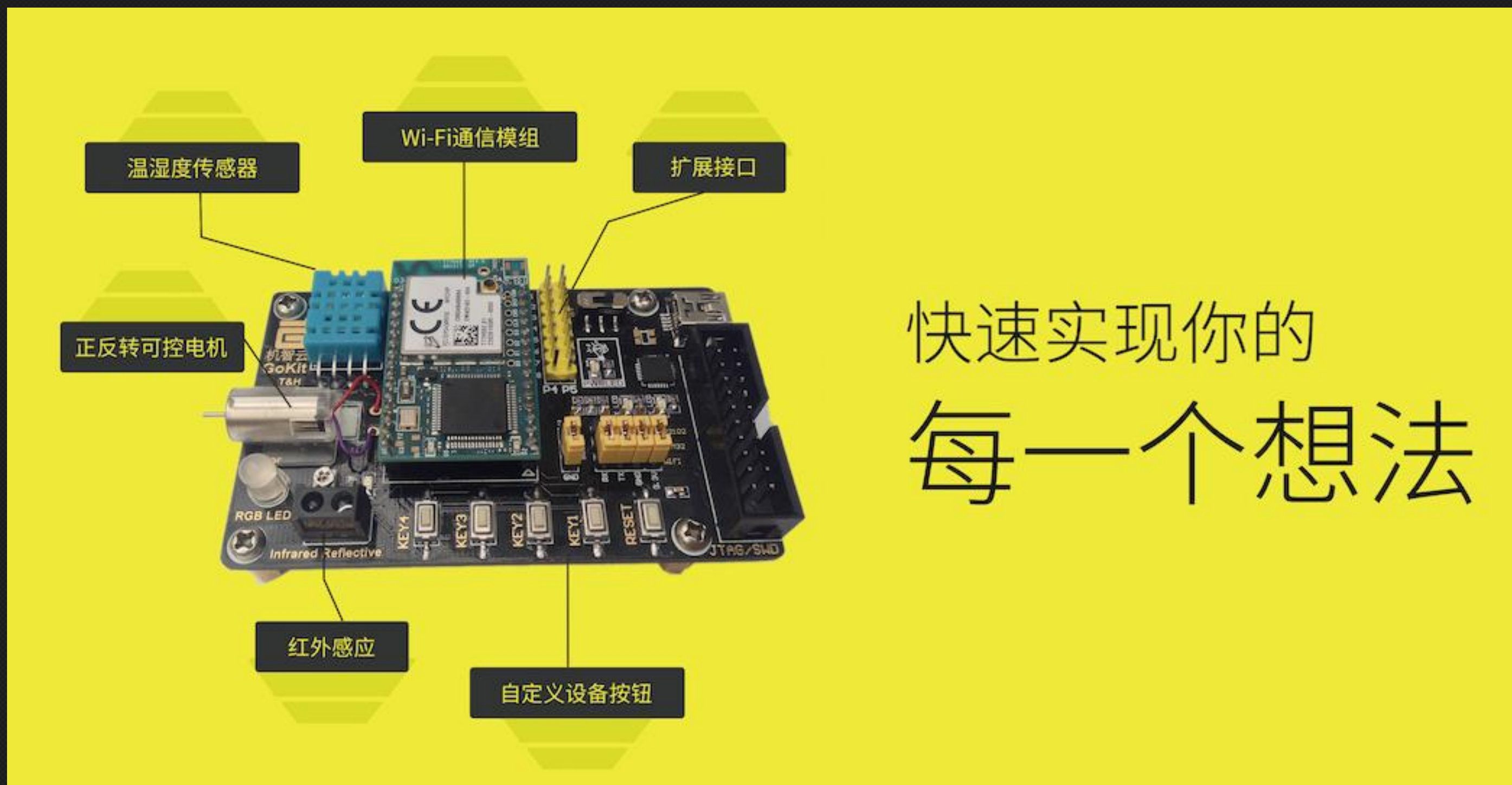
特色二：设备活跃周期统计

特色三：客户端（app）使用分析

特色四：地域分布统计

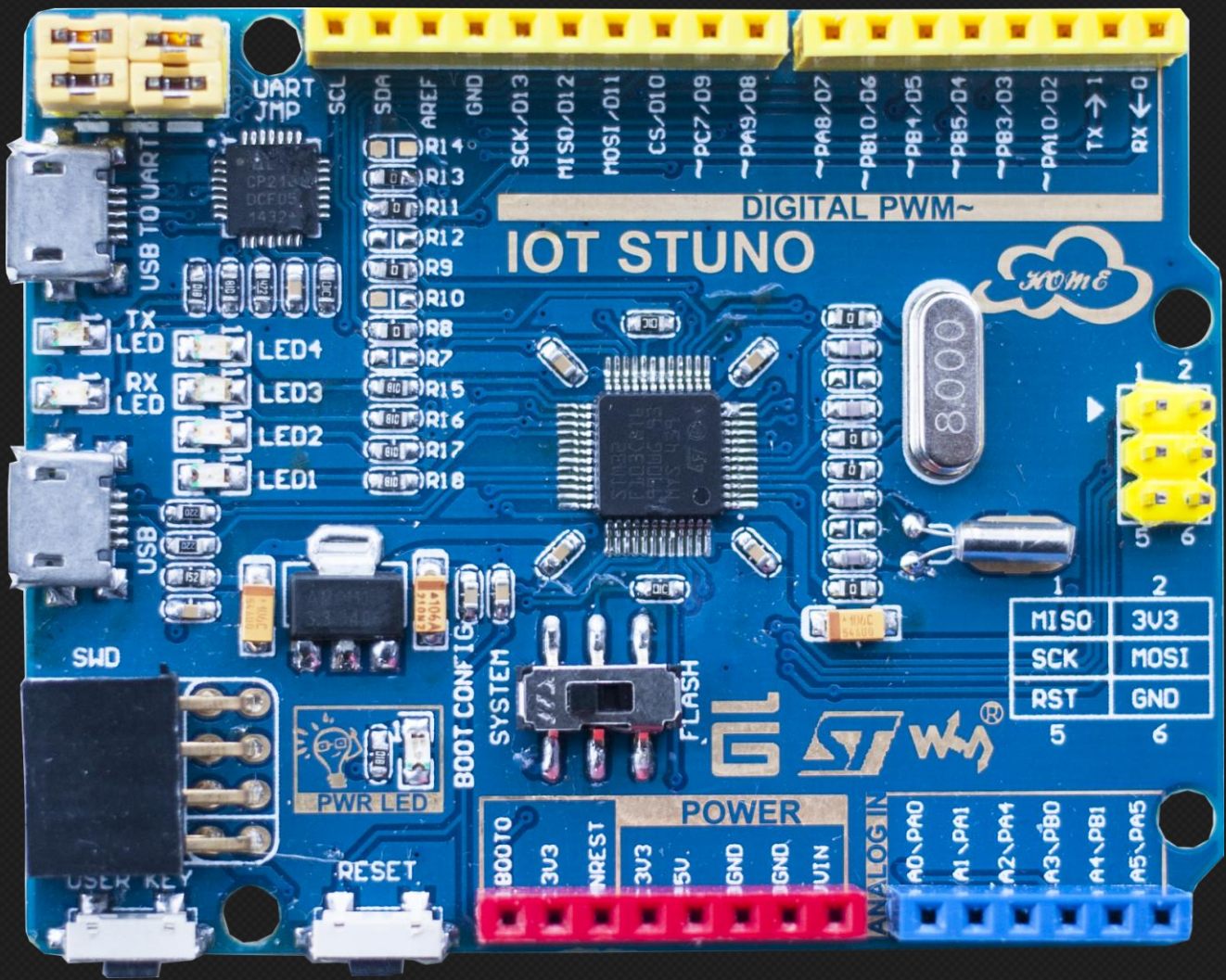
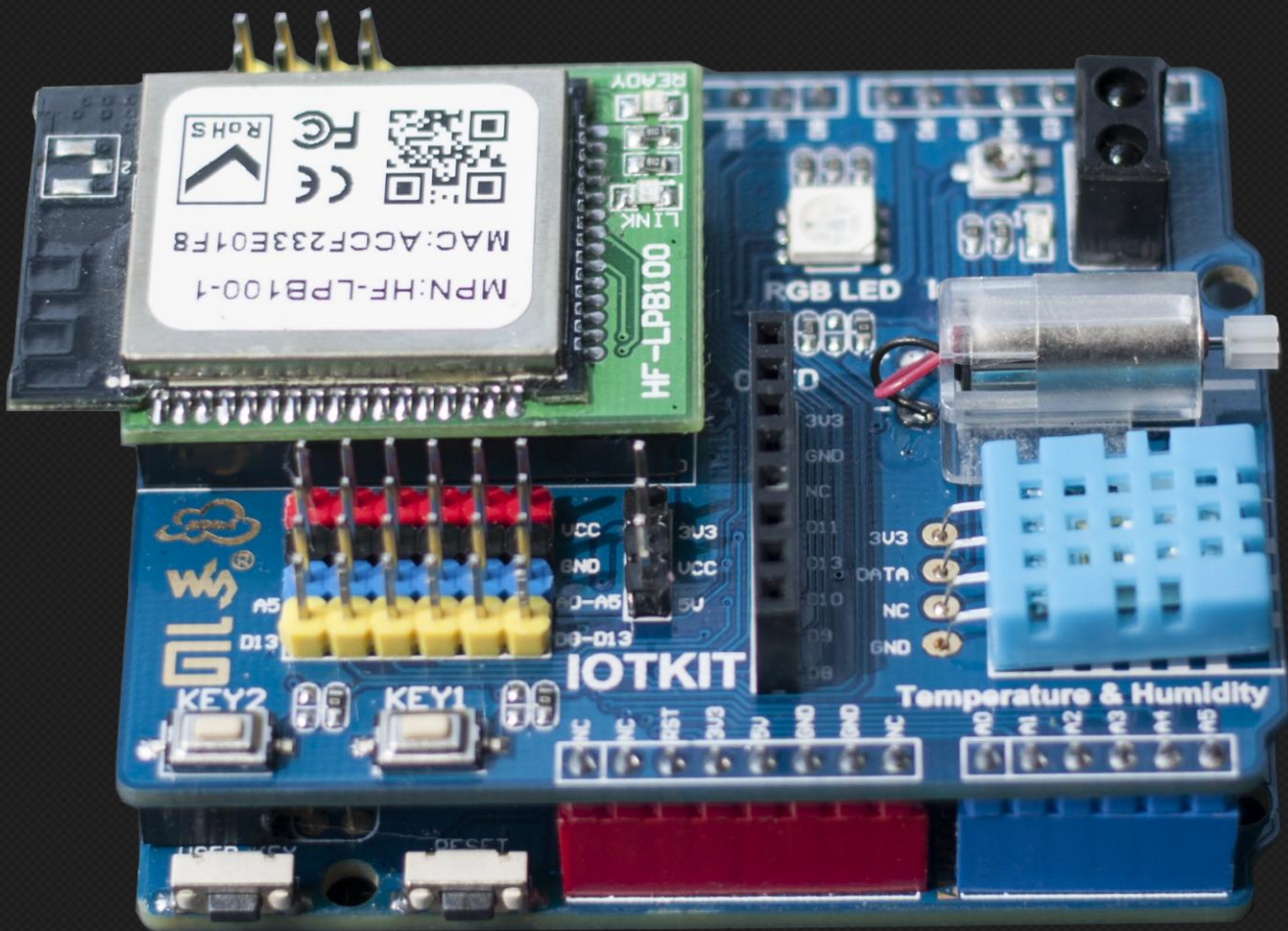
Gokit

Free



Gokit 2代

Free



Free

自助开发平台--首页



硬件接入

开发文档

下载中心

视频教学

论坛

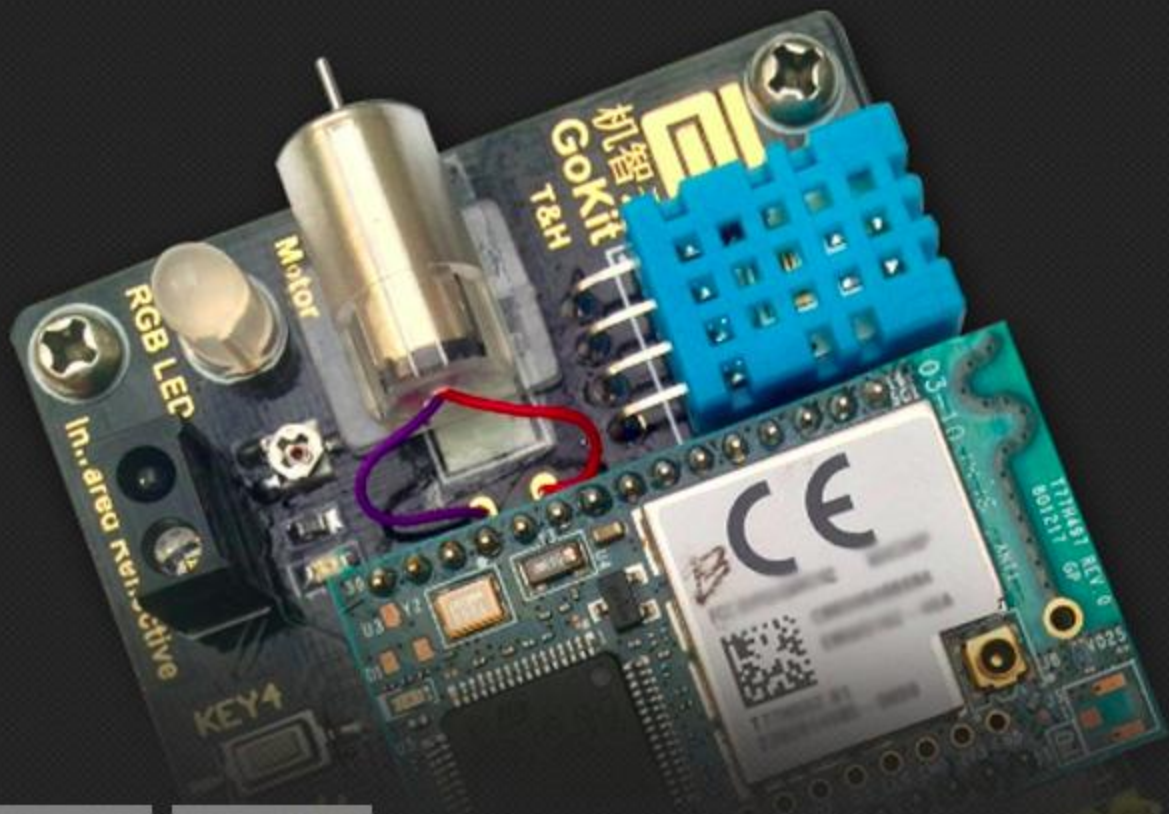
agao@gizwits...

消息



GoKit

智能设备开发套件



平台服务

从设备接入到硬件开发，从方案产出到方案推广，机智云平台为您提供最优质的智能硬件服务，机智云只为智能硬件而生。



硬件接入

物联网硬件，首先解决的就是设备接入互联网的问题。完成硬件的互联网接入之后，信息可以在产品与云端、制造商与用户之间互联互通；通过连接，产品的某些功能可以脱离物理装置，直接在“产品”云中存在。

Free

自助开发平台--新建设备接入



Free

自助开发平台--编辑数据点

新设备接入向导

编辑数据点

mcu编程

设备调试

通过数据点协议的编辑，你可以制定设备每一个功能是否对外发送状态数据以及是否接受外部的控制。

通过方案模板创建

手动编辑数据点

宠物屋

狗屋和猫屋

贡献者：user@xtremeprog.com

数据点：2 查看

应用此模版

GoKit Demo 方案模板

Demo 方案

贡献者：yliu@xtremeprog.com

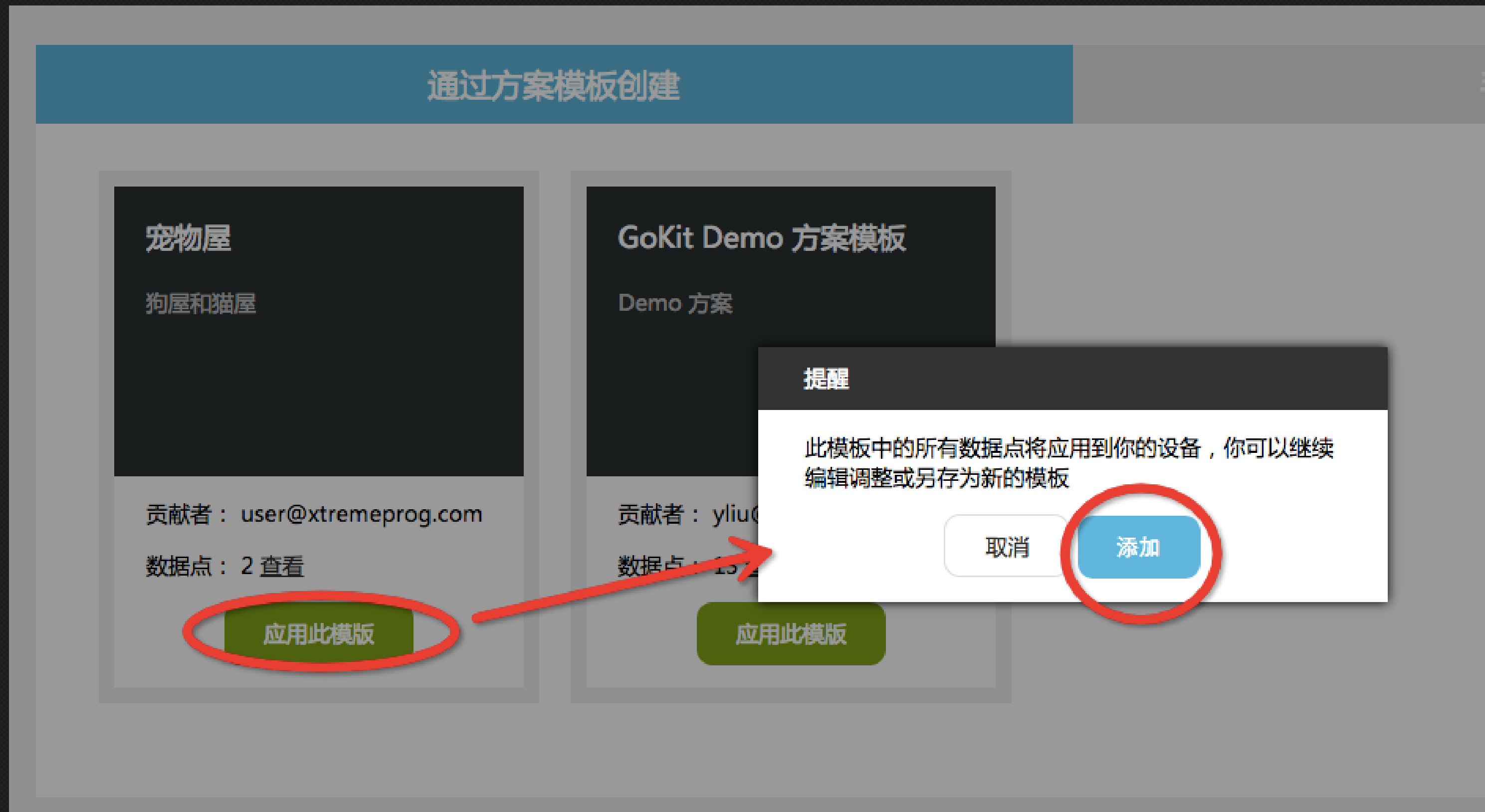
数据点：13 查看

应用此模版

 机智云
Gizwits

Free

自助开发平台--数据点模板



Free

自助开发平台--手动编辑数据点

手动编辑数据点

添加数据点

显示名称 *

滚筒速度

读写类型

可写

数据类型

数字

数据范围(x) *

1

-

5

分辨率(k) *

1

取消

添加

+ 新建数据点

编辑

删除

编辑

删除

编辑

删除

编辑

删除

编辑

删除

数据类型：数字

数据类型：数字

Free

自助开发平台--MCU编程

新设备接入向导



根据自动生成的mcu程序开发文档你可以快速的完成mcu通信的编写

现在，你可以开始编写你的mcu程序。机制云根据你前面制定的数据点生成了一份《GoKit开发套件mcu程序指南》，你可以按提示完成你的mcu开发。
你也可以直接进入下一步，通过我们的设备模拟器以及demo app验证你的数据点创建是否有效。

上一步

下一步

Free

自助开发平台--DEMO App

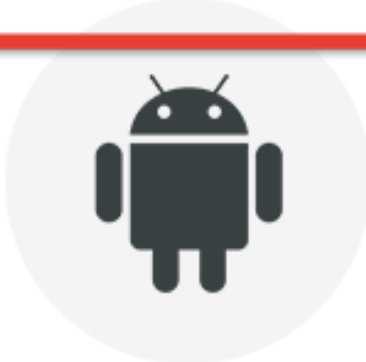
[编辑数据点](#)[mcu编程](#)[设备调试](#)

使用我们提供的设备模拟器、在线轻应用、Demo App等工具对你的设备进行验证

品开发资源



[下载SDK](#)[DEMO APP下载](#)[集成文档](#)



[下载SDK](#)[DEMO APP下载](#)[集成文档](#)

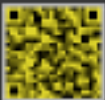
自助开发平台--虚拟设备

发布

技术支持：电话 400-6525-488 QQ: 8

产品名称：滚筒洗衣机

属性名：	设定LED颜色R值	类型：	uint8	value:	0
属性名：	设定LED颜色G值	类型：	uint8	value:	0
属性名：	设定LED颜色B值	类型：	uint8	value:	0
属性名：	设定电机转速	类型：	uint16	value:	0
属性名：	设定LED亮度	类型：	uint32	value:	0
属性名：	滚筒速度	类型：	uint32	value:	0
属性名：	环境温度	类型：	uint8	value:	0
属性名：	环境湿度	类型：	uint8	value:	0
属性名：	红外探测	类型：	uint8	value:	0
属性名：	led灯故障	类型：	bool	value:	0
属性名：	温度传感器故障	类型：	bool	value:	0
属性名：	湿度传感器故障	类型：	bool	value:	0
属性名：	电机故障	类型：	bool	value:	0

扫码控制设备

通信日志

登录成功!
正在登录服务器...

开发资源

www.gizwits.com 、文档中心、下载中心、社区等

硬件开发资源:

Gokit开发版 硬件程序、电路原理;
多款WiFi模组的固件程序;

客户端开发资源:

设备接入SDK, 接入类型有WiFi/BLE, 支持系统有iOS、Android等;
统计分析SDK;

开发与调试工具:

WiFi或移动通信产品 Demo App;

产测工具

开源项目:

10款公版开源App源码, 包括iOS、Android不同平台;
GoKit 智能宠物屋 App代码, 包括iOS、Android不同平台;
多个开源硬件项目, 有桌上足球、植物宝、智能宠物屋等;
BT/BLE设备接入参考实现源码, 包括iOS、Android不同平台;

Free

MCU二次开发

gokit_mcu_stm32_V2.3.2.zip

-Hal

多种传感器驱动程序

-Lib

STM32开发库

-Project

工程文件

-Protocol

串口通信协议处理模块

-User

用户逻辑处理模块

README.txt

MCU二次开发-数据结构

```
gokit.h
14 #include <string.h>
15
16 __packed typedef struct →
17 {
18     →uint8_t → → → → LED_Cmd;
19     →uint8_t → → → → LED_R;
20     →uint8_t → → → → LED_G;
21     →uint8_t → → → → LED_B;
22     →MOTOR_T → → → → Motor;
23     →uint8_t → → → → Infrared;
24     →uint8_t → → → → Temperature;
25     →uint8_t → → → → Humidity;
26     →uint8_t → → → → Alert;
27     →uint8_t → → → → Fault;
28 }ReadTypeDef_t;
29
30 __packed typedef struct →
31 {
32     →uint8_t → → → → → Attr_Flags;
33     →uint8_t → → → → → LED_Cmd;
34     →uint8_t → → → → → LED_R;
35     →uint8_t → → → → → LED_G;
36     →uint8_t → → → → → LED_B;
37     →MOTOR_T → → → → → Motor;
38
39 }WriteTypeDef_t;
40
```

4.9 设备MCU向WiFi模组主动上报当前状态

设备MCU发送:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	action (1B)	dev_status (11B)	checksum (1B)
0xFFFF	0x0011	0x05	0x##	0x0000	0x04	设备状态	0x##

注:

1. 设备状态 (dev_status) 使用一个或多个字节表示。例如数据包为 0x07 FE FE FE 00 0A 01 C8 64 03 0F 时, 其格式为:

字节序	位序	数据内容	说明
-----	----	------	----

byte0	bit7 bit6 . . bit1 bit0	0b00000111	LED_OnOff, 类型为bool, 值为true; 字段bit0, 字段值为0b1; LED_Color, 类型为enum, 值为3; 字段bit2 ~ bit1, 字段值为0b11;
byte1		0xFE	LED_R, 类型为uint8, 字段值为254; 实际值计算公式: $y = 1.000000 * x + (0.000000)$ x最小值为0, 最大值为254
byte2		0xFE	LED_G, 类型为uint8, 字段值为254; 实际值计算公式: $y = 1.000000 * x + (0.000000)$ x最小值为0, 最大值为254
byte3		0xFE	LED_B, 类型为uint8, 字段值为254; 实际值计算公式: $y = 1.000000 * x + (0.000000)$ x最小值为0, 最大值为254
byte4 byte5		0x00 0A	Motor_Speed, 类型为uint16, 字段值为10; 实际值计算公式: $y = 1.000000 * x + (-5.000000)$ x最小值为0, 最大值为10

4.10 WiFi模组控制设备

WiFi模组发送:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	action (1B)	attr_flags (1B)	attr_vals (6B)	checksum (1B)
0xFFFF	0x0000	0x03	0x##	0x0000	0x01	是否设置标志位	设置数据值	0x##

注:

1. 是否设置标志位 (attr_flags) 表示相关的数据值是否为有效值, 相关的标志位为1表示值有效, 为0表示值无效。从右到左的标志位依次为:

bit0: 设置LED_OnOff

bit1: 设置LED_Color

bit2: 设置LED_R

bit3: 设置LED_G

bit4: 设置LED_B

bit5: 设置Motor_Speed

2. 设置数据值 (attr_vals) 存放数据值, 只有相关的设置标志位为1时, 数据值才有效, 例如数据包为 0x07 FE FE FE 00 0A 时, 其格式为:

字节序	bit序	数据内容	说明
byte0	bit7 bit6 . . bit1 bit0	0b00000111	LED_OnOff, 类型为bool, 值为true; 字段bit0, 字段值为0b1; LED_Color, 类型为enum, 值为3; 字段bit2 ~ bit1, 字段值为0b11;
byte1		0xFE	LED_R, 类型为uint8, 字段值为254; 实际值计算公式: $y = 1.000000 * x + (0.000000)$ x最小值为0, 最大值为254
byte2		0xFE	LED_G, 类型为uint8, 字段值为254; 实际值计算公式: $y = 1.000000 * x + (0.000000)$ x最小值为0, 最大值为254
byte3		0xFE	LED_B, 类型为uint8, 字段值为254; 实际值计算公式: $y = 1.000000 * x + (0.000000)$ x最小值为0, 最大值为254

MCU 开发资源

现在, 你可以开始开发你的 MCU 程序, 机智云根据你在上一步添加的数据点生成了 [《微信宠物屋-机智云接入串口通信协议文档》](#), 你可以按提示完成你的 MCU 开发。

此外, 机智云还自动生成了数据点的 [《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, 你可以根据json文档自动化生成MCU头文件, 甚至是APP的页面, 这将极大的简化你的开发过程。

MCU二次开发-相关函数

User/main.c line 136

```
void GizWits_ControlDeviceHandle()
```

User/main.c line 61

```
GizWits_GatherSensorData()
```

检测传感器的值并更新。

```
void GizWits_DevStatusUpgrade(uint8_t *P0_Buff, uint32_t Time,  
uint8_t flag)
```

上报数据接口，P0_Buff上报数据，Time上报间隔，flag(1为立即上报)。

其他函数

```
void KEY_Handle(void)
```

```
void GizWits_WiFiStatueHandle(uint16_t wifiStatue)
```


Thanks



机智云 (gizwits)