

# 机智云 - 设备串口通讯协议 (v4.0.0)

产品名称: IOTKIT

生成日期: 2015-01-19

## 目录

1. 设备通讯信息
  - 1.1 MCU与WIFI模组串口连接要求
2. 命令格式
3. 约定
4. 命令列表
  - 4.1 WiFi模组请求设备信息
  - 4.2 WiFi模组与设备MCU的心跳
  - 4.3 设备MCU通知WiFi模组进入配置模式
  - 4.4 设备MCU重置WiFi模组
  - 4.5 WiFi模组向设备MCU通知WiFi模组工作状态的变化
  - 4.6 WiFi模组请求重启MCU
  - 4.7 非法消息通知
  - 4.8 WiFi模组读取设备的当前状态
  - 4.9 设备MCU向WiFi模组主动上报当前状态
  - 4.10 WiFi模组控制设备

## 1. 设备通讯信息

### 1.1 MCU与WIFI模组串口连接要求

通讯方式: UART

波特率: 9600

数据位: 8

奇偶校验: 无

停止位: 1

数据流控: 无

给WIFI模组供电电压: 3.3v, 电流(max): 150mA

如需MCU升级等高级功能, 请和Gizwits联系。

## 2. 命令格式

header(2B)=0xFFFF, len(2B), cmd(1B), sn(1B), flags(2B), payload(xB), checksum(1B)

## 3. 约定

- 包头(header)固定为0xFFFF
- 长度(len)是指从cmd开始到整个数据包结束所占用的字节数
- 多于一个字节的整型数字以大端字节序编码
- 消息序号(sn)由发送方给出,接收方响应命令时需把消息序号返回给发送方
- 校验和(checksum)的计算方式为把数据包按字节求和得出的结果对256求余
- 除“非法消息通知”外的命令都带有确认,如在200毫秒内没有收到接收方的响应,发送方应重发,最多重发3次。

## 4. 命令列表

### 4.1 WiFi模组请求设备信息

WiFi模组发送:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	checksum (1B)
0xFFFF	0x0005	0x01	0x##	0x0000	0x##

设备MCU回复:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	protocol_ver (8B)
0xFFFF	0x0047	0x02	0x##	0x0000	0x3030303030303034

p0_ver (8B)	hard_ver (8B)	soft_ver (8B)	product_key (32B)	bindable_timeout (2B)	checksum (1B)
0x3030303030303032	硬件版本号	软件版本号	产品标识码	绑定超时(秒)	0x##

注:

绑定超时(bindable\_timeout)的值为0时,表示设备随时可在局域网被绑定;当值大于零时,表示当按下绑定按钮后,用户必须在该时间范围内完成绑定操作。

### 4.2 WiFi模组与设备MCU的心跳

WiFi模组发送:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	checksum (1B)
0xFFFF	0x0005	0x07	0x##	0x0000	0x##

设备MCU回复:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	checksum (1B)
0xFFFF	0x0005	0x08	0x##	0x0000	0x##

注:

当设备MCU在180秒内没有收到WiFi模组的心跳请求,则通过硬件引脚重启WiFi模组。

### 4.3 设备MCU通知WiFi模组进入配置模式

设备MCU发送:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	config_method (1B)	checksum (1B)
0xFFFF	0x0006	0x09	0x##	0x0000	配置方式	0x##

注:

配置方式(config\_method)是指使用何种方法配置WiFi模组加入网络, 可以选择以下的值:

1: SoftAp

2: Air Link

其它的值为保留值。

WiFi模组回复:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	checksum (1B)
0xFFFF	0x0005	0x0A	0x##	0x0000	0x##

注:

当WiFi模组收到进入配置模式的指令后, 需要判断当前是否已成功连接上无线路由器, 如成功, 则进让设备处于可绑定模式, 否则让设备进入对应的SoftAP或AirLink等OnBoarding配置方式。

#### 4.4 设备MCU重置WiFi模组

设备MCU发送:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	checksum (1B)
0xFFFF	0x0005	0x0B	0x##	0x0000	0x##

WiFi模组回复:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	checksum (1B)
0xFFFF	0x0005	0x0C	0x##	0x0000	0x##

注:

被重置后的WiFi模组需要重新配置与绑定。

#### 4.5 WiFi模组向设备MCU通知WiFi模组工作状态的变化

WiFi模组发送:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	wifi_status (2B)	checksum (1B)
0xFFFF	0x0007	0x0D	0x##	0x0000	WiFi状态	0x##

注:

1. WiFi状态(wifi\_status)用两个字节描述, 从右向左依次是bit0, bit1, ... bit15;

bit0: 是否开启SoftAP模式, 0: 关闭, 1: 开启;

bit1: 是否开启Station模式, 0: 关闭, 1: 开启;

bit2: 是否开启配置模式, 0: 关闭, 1: 开启;

bit3: 是否开启绑定模式, 0: 关闭, 1: 开启;

bit4: WiFi模组是否成功连接路由器, 0: 未连接, 1: 连接;

bit5: WiFi模组是否成功连接云端, 0: 未连接, 1: 连接;

bit6 - bit8: 仅当WiFi模组已成功连接路由器(请看上第4位)时值才有效, 三个位合起来表示一个整型值, 值范围为0~7, 表示WiFi模组当前连接AP的信号强度(RSSI), 0为最低, 7为最高;

bit9: 是否有已绑定的手机上线, 0: 没有, 1: 有;

bit10 - bit15: 预留。

2. WiFi模组在当状态发生了变化后立刻通知设备MCU, 同时每隔10分钟也会定期向设备MCU发送状态。

设备MCU回复:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	checksum (1B)
0xFFFF	0x0005	0x0E	0x##	0x0000	0x##

#### 4.6 WiFi模组请求重启MCU

WiFi模组发送:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	checksum (1B)
0xFFFF	0x0005	0x0F	0x##	0x0000	0x##

设备MCU回复:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	checksum (1B)
0xFFFF	0x0005	0x10	0x##	0x0000	0x##

注:

为了避免WiFi模组没有收到确认而重发指令而造成MCU多次重启, 故MCU回复WiFi模组后需等待600毫秒再进行重启。

#### 4.7 非法消息通知

WiFi模组回应MCU对应包序号的数据包非法:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	error_code (1B)	checksum (1B)
0xFFFF	0x0006	0x11	0x##	0x0000	错误码	0x##

MCU回应WiFi模组对应包序号的数据包非法:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	error_code (1B)	checksum (1B)
0xFFFF	0x0006	0x12	0x##	0x0000	错误码	0x##

注:

错误码 (error\_code)可为以下的值:

- 1: 校验和错误

2: 命令不可识别

3: 其它错误

0和4~255保留

#### 4.8 WiFi模组读取设备的当前状态

WiFi模组发送:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	action (1B)	checksum (1B)
0xFFFF	0x0006	0x03	0x##	0x0000	0x02	0x##

设备MCU回复:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	action (1B)	dev_status (8B)	checksum (1B)
0xFFFF	0x000E	0x04	0x##	0x0000	0x03	设备状态	0x##

注:

设备状态(dev\_status)使用一个或多个字节表示。例如数据包为0xFE FE FE 00 0A 01 64 64 时, 其格式为:

字节序	位序	数据内容	说明
byte0		0xFE	dp1, 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
byte1		0xFE	dp2, 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	dp3, 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 byte4		0x00 0A	dp4, 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
byte5	bit7 bit6 . . . bit1 bit0	0b00000001	dp7, 类型为bool, 值为true: 字段bit0, 字段值为0b1;

byte6		0x64	dp5, 类型为uint8, 字段值为100; 实际值计算公式 $y=1.000000*x+(0.000000)$ x最小值为0, 最大值为100
byte7		0x64	dp6, 类型为uint8, 字段值为100; 实际值计算公式 $y=1.000000*x+(0.000000)$ x最小值为0, 最大值为100

#### 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 (8B)	checksum (1B)
0xFFFF	0x000E	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0xFE FE FE 00 0A 01 64 64 时, 其格式为:

字节序	位序	数据内容	说明
byte0		0xFE	dp1, 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
byte1		0xFE	dp2, 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	dp3, 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 byte4		0x00 0A	dp4, 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
byte5	bit7 bit6 . . . bit1 bit0	0b00000001	dp7, 类型为bool, 值为true: 字段bit0, 字段值为0b1;

byte6		0x64	dp5, 类型为uint8, 字段值为100; 实际值计算公式 $y=1.000000*x+(0.000000)$ x最小值为0, 最大值为100
byte7		0x64	dp6, 类型为uint8, 字段值为100; 实际值计算公式 $y=1.000000*x+(0.000000)$ x最小值为0, 最大值为100

2. 关于发送频率。当设备MCU收到WiFi模组控制产生的状态变化, 设备MCU应立刻主动上报当前状态, 发送频率不受限制。但如设备的状态的变化是由于用户触发或环境变化所产生的, 其发送的频率不能快于2秒每次。

3. 设备MCU需要每隔10分钟定期主动上报当前状态。

WiFi模组回复:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	checksum (1B)
0xFFFF	0x0005	0x06	0x##	0x0000	0x##

#### 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 (5B)	checksum (1B)
0xFFFF	0x000C	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dp1

bit1: 设置dp2

bit2: 设置dp3

bit3: 设置dp4

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0xFE FE FE 00 0A 时, 其格式为:

字节序	bit序	数据内容	说明
byte0		0xFE	dp1, 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
byte1		0xFE	dp2, 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	dp3, 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 byte4	0x00 0A	dp4, 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
----------------	---------	---

设备MCU回复:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	checksum (1B)
0xFFFF	0x0005	0x04	0x##	0x0000	0x##