

# 机智云 - 设备串口通讯协议 (v4.0.9)

---

产品名称: IOT\_DEMO

生成日期: 2016-11-21

## 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](#)
  - [4.11 MCU请求WiFi模组进入产测模式](#)
  - [4.12 MCU通知WiFi模组进入可绑定模式](#)
  - [4.13 MCU请求获取网络时间](#)
  - [4.14 大数据下发: 数据发起者请求向数据接收者发送大数据](#)
  - [4.15 大数据下发: 数据接收者告知数据发起者可以开始发送数据](#)
  - [4.16 大数据下发: 数据发送者向数据接收者下发数据分片](#)
  - [4.17 大数据下发: 数据发起者向数据接收者通知取消数据下发](#)
  - [4.18 MCU获取通讯模组的信息](#)
  - [4.19 MCU请求通讯模组进行事务处理](#)

## 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
- 包长度(len)是指从命令开始一直到校验和的字节长度(包括命令和校验和)。因为包头为固定0xFFFF, 对于发送方, 如检测到出现0xFF的数据内容, 需要在0xFF后添加0x55。对于接收方, 如检测到非包头部分出现0xFF, 需要把紧跟其后的0x55移除。
- 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从长度位开始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	0x004f	0x02	0x##	0x0000	0x3030303030303034

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

设备属性: 从右向左编号成bit0~bit63。bit0=1表示设备是中控设备。bit1~bit63预留。

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##

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 - bit7:预留;

bit8 - bit10: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;

bit11:是否有已绑定的手机上线, 0:没有, 1:有;

bit12:是否处于产测模式中, 0:否, 1:是;

bit13 - 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 (32B)	checksum (1B)
0xFFFF	0x0026	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

0x01 02 03 ... 04 05 时，其格式为：

字节序	位序	数据内容	说明
byte0 byte1 byte2 . . . byte30 byte31		0x01 02 03 ... 04 05	Packet，类型为binary，字段长度为32，值为[1, 2, 3 ... 4, 5]

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 (32B)	checksum (1B)
0xFFFF	0x0026	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

0x01 02 03 ... 04 05 时，其格式为：

字节序	位序	数据内容	说明
byte0 byte1 byte2 . . . byte30 byte31		0x01 02 03 ... 04 05	Packet，类型为binary，字段长度为32，值为[1, 2, 3 ... 4, 5]

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6秒每次。建议按需上报, 有特殊上报需求请联系机智云。
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 (32B)	checksum (1B)
0xFFFF	0x0027	0x03	0x##	0x0000	0x01	是否设置标志位	设置数据值	0x##

注:

- 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Packet
- 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 0x01 02 03 ... 04 05 时, 其格式为:

字节序	bit序	数据内容	说明
byte0 byte1 byte2 . . . byte30 byte31		0x01 02 03 ... 04 05	Packet, 类型为binary, 字段长度为32, 值为[1, 2, 3 ... 4, 5]

设备MCU回复:

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

重要说明:无论设备的状态是否发生变化, MCU需要立即上报一次最新的设备状态, 格式和流程参见4.9部分。

#### 4.11 MCU请求WiFi模组进入产测模式¶

设备MCU发送:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	checksum (1B)
FF FF	0x05	0x13	0x##	0x0000	0x##

WiFi模组回复:

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	checksum (1B)
-------------	----------	----------	---------	------------	---------------

FF FF	0x05	0x14	0x##	0x0000	0x##
-------	------	------	------	--------	------

4.12 MCU通知WiFi模组进入可绑定模式

设备MCU发送：

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	checksum (1B)
FF FF	0x05	0x15	0x##	0x0000	0x##

WiFi模组回复：

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	checksum (1B)
FF FF	0x05	0x16	0x##	0x0000	0x##

注：

可绑定的时间由“获取设备信息”时指定。当可绑定的时间不为0时,设备上电后,自动在可绑定时间的秒数内会处于可绑定模式。

4.13 MCU请求获取网络时间

设备MCU发送：

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	checksum (1B)
FF FF	0x05	0x17	0x##	0x0000	0x##

WiFi模组回复：

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	time (7B)	ntp_time (4B)	checksum (1B)
FF FF	0x0C	0x18	0x##	0x0000		秒	0x##

ntp\_time为1970年1月1日至今的秒数（零时区时间）。

Time用7个字节表示,0x07 DF 01 02 03 04 05格式如下：

字节序	位序	数据内容	说明
byte0 byte1		07 DF	年 (2015, 网络字节序)
byte2		01	月
byte3		02	日
byte4		03	时
byte5		04	分
byte6		05	秒

4.14 大数据下发：数据发起者请求向数据接收者发送大数据

大数据发起者发送：

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	大数据信息 (##B)	checksum (1B)
FF FF	0x##	0x19	0x##	0x0000		0x##

序号	字段名称	字节长度 (B)	内容说明
1	数据大小	4	请求传送的数据字节大小
2	数据校验码长度len	2	len(数据校验码)
3	数据校验码	len	数据校验码的内容，使用MD5校验算法

大数据接收者回复：

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	checksum (1B)
FF FF	0x05	0x1A	0x##	0x0000	0x##

4.15 大数据下发：数据接收者告知数据发起者可以开始发送数据¶

大数据发起者发送：

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	大数据信息 (##B)	checksum (1B)
FF FF	0x##	0x1B	0x##	0x0000		0x##

大数据信息：

序号	字段名称	字节长度 (B)	内容说明
1	数据校验码长度len	2	len(数据校验码)
2	数据校验码	len	向WiFi模组回传准备接收数据的数据校验码的内容
3	分片大小	2	大数据需要分片传送。由MCU指定数据分片的大小，分片大小建议设为128B

大数据发起者回复：

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	checksum (1B)
FF FF	0x05	0x1C	0x##	0x0000	0x##

4.16 大数据下发：数据发送者向数据接收者下发数据分片¶

大数据发起者发送：

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	大数据信息 (##B)	checksum (1B)
FF FF	0x##	0x1D	0x##	0x0000		0x##

大数据信息：

序号	字段名称	字节长度 (B)	内容说明
1	分片序号	2	当前数据包的分片序号，分片序号从1开始计算
2	总分片数	2	
3	分片数据内容		

大数据接收者回复：



header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	checksum (1B)
FF FF	0x05	0x1E	0x##	0x0000	0x##

4.17 大数据下发：数据发起者向数据接收者通知取消数据下发

大数据发起者发送：

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	checksum (1B)
FF FF	0x05	0x1F	0x##	0x0000	0x##

大数据接收者回复：

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	checksum (1B)
FF FF	0x05	0x20	0x##	0x0000	0x##

4.18 MCU获取通讯模组的信息

通讯模组上电后，进入正常工作模式后，MCU可以向通讯模组查询相关信息。 各产品可以根据需要判断是否支持此协议。

MCU发出：

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	type (1B)	checksum (1B)
FF FF	0x06	0x21	0x##	0x0000	0x00	0x##

WiFi模组回复：

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	WiFiInfo (65B)	checksum (1B)
FF FF	0x46	0x22	0x##	0x0000		0x##

Wifi Info：

序号	字段名称	字节长度 (B)	内容说明
1	ModuleType	1	0x01：WiFi模组
2	通用串口协议版本号	8	字符串，形如“00000004”
3	硬件版本号	8	字符串，形如“HFLPB100”
4	软件版本号	8	字符串，形如“04020100”
5	MAC	16	字符串，形如： 5CF9388AE8F0， 全大写，前对齐，后补零
6	IP	16	字符串，形如：192.168.100.254
7	设备属性	8	设备属性，预留。

2G/3G/4G模组回复：

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	2GInfo (#B)	checksum (1B)
FF FF	0x##	0x22	0x##	0x0000		0x##

2GInfo：

序号	字段名称	字节长度 (B)	内容说明
1	Type	1	0x02: 2G/3G/4G模组
2	通用串口协议版本号	8	字符串, 形如"00000004"
3	硬件版本号	8	字符串
4	软件版本号	8	字符串
5	设备属性	8	设备属性, 预留。
6	IMEI	16	字符串, 形如: "355065053311001"
7	IMSI	16	字符串, 形如: "355065053311001"
8	MCC移动国家码	8	字符串, 形如: "460"
9	MNC移动网络码	8	字符串, 形如: "03"
10	CellNum基站数量	1	无符号数字, 范围: 0-255
11	基站信息长度	1	无符号数字, 范围: 0-255, 目前长度固定为5
12	基站1信息	5	参见下表: 基站信息
13	.....	5	参见下表: 基站信息
14	基站n信息	5	参见下表: 基站信息

基站信息:

序号	字段名称	字节长度 (B)	内容说明
1	LAC区域ID	2	无符号数字, 范围: 0-65535
2	CellID基站ID	2	无符号数字, 范围: 0-65535
3	RSSI信号强度	1	无符号数字, 范围: 0-255

#### 4. 19 MCU请求通讯模组进行事务处理

说明:

- 1、此过程为MCU申请模组做事务处理的通用流程，一共两次交互，每次交互两次通讯，因为事务处理需要一段时间，第一个来回和第二个来回之间不可用阻塞的方式进行等待。
- 2、具体的事务处理数据，参见第4部分的事务附录。

MCU向通讯模组请求事务处理，MCU => 通讯模组。

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	info1 (#B)	checksum (1B)
FF FF	0x##	0x23	0x##	0x0000		0x##

通讯模组响应MCU，表示收到请求。

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	info2 (#B)	checksum (1B)
FF FF	0x##	0x24	0x##	0x0000		0x##

在此期间，MCU不可以进行阻塞等待，通常会有秒级的时间间隔。

通讯模组事务处理完成后，通知MCU处理结果。

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	info3 (#B)	checksum (1B)
FF FF	0x##	0x25	0x##	0x0000		0x##

MCU响应通讯模组。

header (2B)	len (2B)	cmd (1B)	sn (1B)	flags (2B)	info4 (#B)	checksum (1B)
FF FF	0x##	0x26	0x##	0x0000		0x##

事务处理一：MCU请求GAgent进行设备OTA检查

info1：MCU向通讯模组进行子设备OTA检查，MCU => 通讯模组。

序号	字段名称	字节长度 (B)	内容说明
1	SubCmd	1	0x01
2	PK	32	字符串
3	DID	32	字符串
4	硬件版本号	8	字符串
5	软件版本号	8	字符串
6	TAG	1	0：不需要GAgent比较结果，仅需要传送软件版本号和URL； 1：需要GAgent比较结果，如果需要升级，直接发送大文件

info2：空。

info3：通讯模组通知MCU OTA检查结果。

当TAG为0的时候，不需要GAgent比较结果，仅需要传送软件版本号和URL

序号	字段名称	字节长度 (B)	内容说明
1	SubCmd	1	0x02
2	SoftVersion	8	
3	URL Length	2	
4	URL	URL Length	

不判断是否需要升级，不进行大文件发送。

当TAG为1的时候，需要GAgent比较结果，如果需要升级，直接发送大文件

序号	字段名称	字节长度 (B)	内容说明
1	SubCmd	1	0x02
2	Result	1	处理结果， 0x00：不需要升级； 0x01：需要升级；

当需要升级时，模组在发送本命令并得到MCU的回复后，便立即启动大文件发送

info4：空。

事务处理二：MCU请求GAgent进行文件下载

info1：MCU向通讯模组进行文件下载，MCU => 通讯模组。

序号	字段名称	字节长度(B)	内容说明
1	SubCmd	1	0x03
2	URL Length	2	
3	URL	URL Length	

info2：空。

info3：通讯模组通知MCU OTA检查结果。

序号	字段名称	字节长度(B)	内容说明
1	SubCmd	1	0x04
2	Result	1	处理结果， 0x00：成功；0x01：失败；

当文件下载成功时，模组会立即启动大文件传输过程。

info4：空。