

# Sample-机智云接入串口通信协议文档

（生成日期：2014-11-28）

产品名称：Sample

## 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状态的变化
  - 4.6 WiFi模组读取设备的当前状态
  - 4.7 设备MCU向WiFi模组主动上报当前状态
  - 4.8 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： RS232

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), params(xB), checksum(1B)
- 3. 约定
  - 1 包头(header)固定为0xFFFF
  - 2 长度(len)是指从cmd开始到整个数据包结束所占用的字节数
  - 3 多于一个字节的整型数字以大端字节序编码

- 4 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
- 5 检验和 (checksum) 的计算方式为把数据包按字节求和得出的结果取出最低字节的内容
- 6 所有发送的命令都带有确认，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)	binable_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：

1. 绑定超时 (binable\_timeout) 的值为0时，表示设备随在局域网被绑定；
- 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：

1. 当设备MCU在180秒内没有收到WiFi模组的心跳请求，应重启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：

1. 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
- 3: WPS
- 4: NFC
- 5: Apple WAC

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：

- 1. 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，从右向左依次是第0位，第1位，...第15位；
  - 第0位：是否开启SoftAP模式，0：关闭，1：开启；
  - 第1位：是否开启Station模式，0：关闭，1：开启；
  - 第2位：是否开启配置模式，0：关闭，1：开启；
  - 第3位：是否开启绑定模式，0：关闭，1：开启；
  - 第4位：WiFi模组是否成功连接路由器，0：未连接，1：连接；
  - 第5位：WiFi模组是否成功连接云端，0：未连接，1：连接；
  - 第6，7，8位：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；
  - 第9 - 15位：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 (31B)	checksum (1B)
0xFFFF	0x0025	0x04	0x##	0x0000	0x03	设备状态	0x##

注：

1. 设备状态 (dev\_status) 使用一个或多个字节表示，其格式为：

- byte0 ~ byte3 (4B), uint32: 时, 最小值0, 最大值23, 实际值的计算公式为:  $y=1.000000*x+(0.000000)$  (x为本字段的值, y为要表示的值)
- byte4 ~ byte7 (4B), uint32: 分, 最小值0, 最大值59, 实际值的计算公式为:  $y=1.000000*x+(0.000000)$  (x为本字段的值, y为要表示的值)
- byte8 ~ byte11 (4B), uint32: 秒, 最小值0, 最大值59, 实际值的计算公式为:  $y=1.000000*x+(0.000000)$  (x为本字段的值, y为要表示的值)
- byte12, bit0, bool: 系统状态
- byte13 ~ byte14 (2B), uint16: PM2.5浓度, 最小值0, 最大值9999, 实际值的计算公式为:  $y=1.000000*x+(0.000000)$  (x为本字段的值, y为要表示的值)
- byte15 ~ byte18 (4B), uint32: 甲醛浓度, 最小值0, 最大值9999, 实际值的计算公式为:  $y=1.000000*x+(0.000000)$  (x为本字段的值, y为要表示的值)
- byte19 ~ byte22 (4B), uint32: 温度, 最小值0, 最大值1099, 实际值的计算公式为:  $y=1.000000*x+(-100.000000)$  (x为本字段的值, y为要表示的值)
- byte23 ~ byte26 (4B), uint32: 湿度, 最小值0, 最大值100, 实际值的计算公式为:  $y=1.000000*x+(0.000000)$  (x为本字段的值, y为要表示的值)
- byte27 ~ byte30 (4B), uint32: 电池容量, 最小值0, 最大值100, 实际值的计算公式为:  $y=1.000000*x+(0.000000)$  (x为本字段的值, y为要表示的值)

4.7 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 (31B)	checksum (1B)
0xFFFF	0x0025	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，其格式为：

- byte0 ~ byte3 (4B), uint32: 时, 最小值0, 最大值23, 实际值的计算公式为:  $y=1.000000*x+(0.000000)$  (x为本字段的值, y为要表示的值)
- byte4 ~ byte7 (4B), uint32: 分, 最小值0, 最大值59, 实际值的计算公式为:  $y=1.000000*x+(0.000000)$  (x为本字段的值, y为要表示的值)
- byte8 ~ byte11 (4B), uint32: 秒, 最小值0, 最大值59, 实际值的计算公式为:  $y=1.000000*x+(0.000000)$  (x为本字段的值, y为要表示的值)
- byte12, bit0, bool: 系统状态
- byte13 ~ byte14 (2B), uint16: PM2.5浓度, 最小值0, 最大值9999, 实际值的计算公式为:  $y=1.000000*x+(0.000000)$  (x为本字段的值, y为要表示的值)
- byte15 ~ byte18 (4B), uint32: 甲醛浓度, 最小值0, 最大值9999, 实际值的计算公式为:  $y=1.000000*x+(0.000000)$  (x为本字段的值, y为要表示的值)
- byte19 ~ byte22 (4B), uint32: 温度, 最小值0, 最大值1099, 实际值的计算公式为:  $y=1.000000*x+(-100.000000)$  (x为本字段的值, y为要表示的值)
- byte23 ~ byte26 (4B), uint32: 湿度, 最小值0, 最大值100, 实际值的计算公式为:  $y=1.000000*x+(0.000000)$  (x为本字段的值, y为要表示的值)
- byte27 ~ byte30 (4B), uint32: 电池容量, 最小值0, 最大值100, 实际值的计算公式为:  $y=1.000000*x+(0.000000)$  (x为本字段的值, y为要表示的值)

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##

4.8 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 (12B)	checksum (1B)
0xFFFF	0x0013	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，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: 设置时
- bit1: 设置分
- bit2: 设置秒

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：

byte0 ~ byte3 (4B), uint32: 时，最小值0，最大值23，实际值的计算公式为： $y=1.000000*x+(0.000000)$  (x为本字段的值，y为要表示的值)

byte4 ~ byte7 (4B), uint32: 分，最小值0，最大值59，实际值的计算公式为： $y=1.000000*x+(0.000000)$  (x为本字段的值，y为要表示的值)

byte8 ~ byte11 (4B), uint32: 秒，最小值0，最大值59，实际值的计算公式为： $y=1.000000*x+(0.000000)$  (x为本字段的值，y为要表示的值)

设备MCU回复：

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