宠物屋产品设备端开发指南

GizWits

修订记录

修改时间	修改内容	版本	修改人	备注
2014-10-28	创建	0.9.0	GizWits	
2014-11-13	1、 增加新增协议命令示例（0B、0C、0D、0E、0F、10、11、12） 2、 标识 cmd 和 p0_cmd	0.9.1	GizWits	
2015/4/22	1、修改 0B 命令的注释	0.9.2	Sean	

1.产品信息

1.1.设备识别码

本品设备识别码是：6f3074fe43894547a4f1314bd7e3ae0b； 设备识别码是区分设备类型的唯一标识，不同功能的产品需要申请不同的设备识别码。

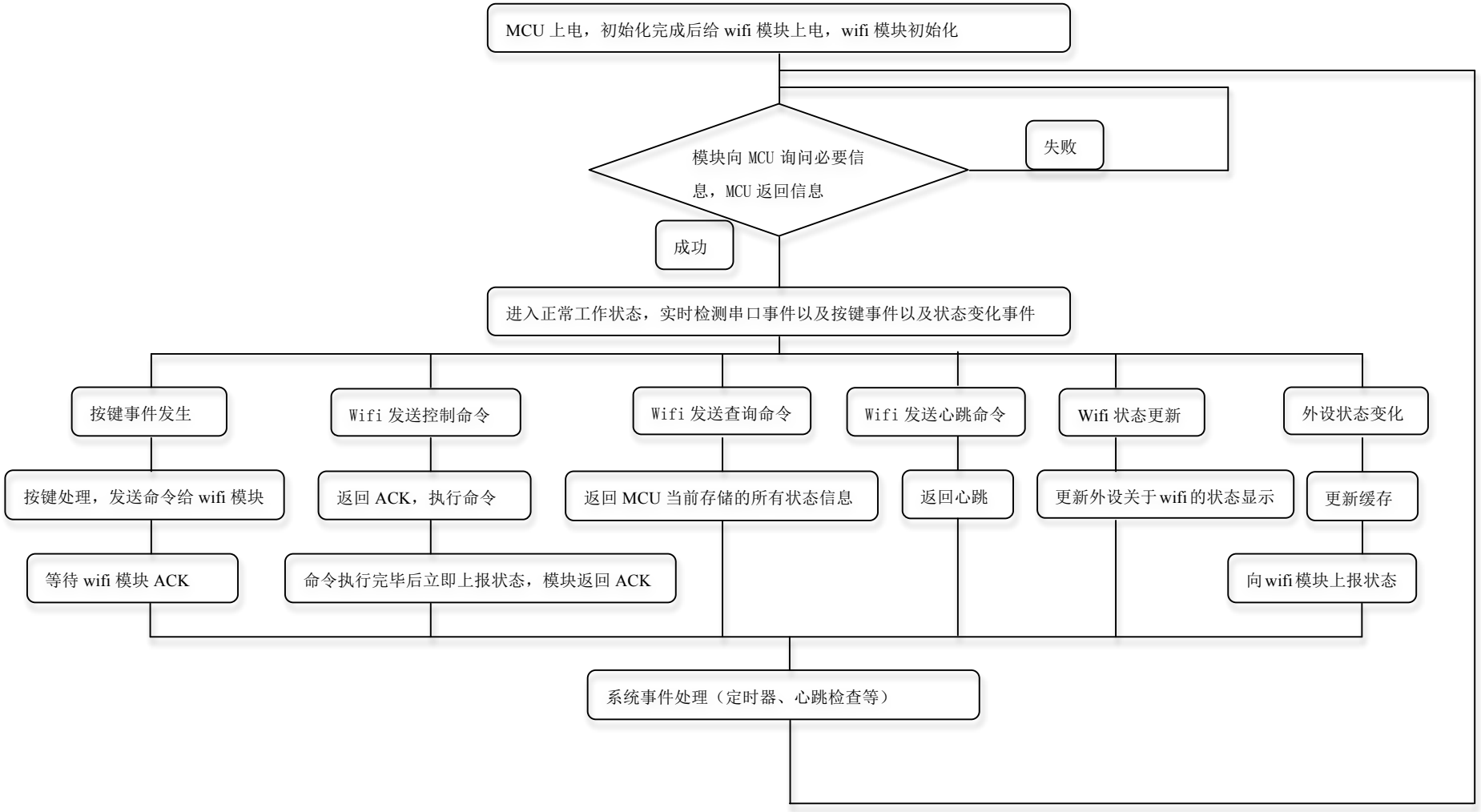
1.2.功能规格表

序号	名　称	方向	类型	长度	作 用 说 明	备　注
1	开启/关闭 红色灯	读写	bit	1	0：关闭， 1：开启，控制红色灯的开关	
2	设定 LED 组合颜色	读写	bit	2	四个枚举值，00：自定义、01：黄色、10：紫色、11：粉色	选定自定义时，可以自定义 RGB 值； 选定非自定义时，无法自定义 RGB 值，如果自定义 RGB，MCU 返回无效命令；
3	设定 LED 红色值	读写	uchar	1	0～254，单独设置 LED 的红色值	
4	设定 LED 绿色值	读写	uchar	1	0～254，单独设置 LED 的绿色值	
5	设定 LED 蓝色值	读写	uchar	1	0～254，单独设置 LED 的蓝色值	
6	设定电机转速	读写	short	2	0～10，0～4 复转，0 转速最快；5 停止；6～10 正转，10 转速最快；	
7	环境温度	只读（环境）	uchar	1	0～200，修正参数：实际温度值＝1*上报温度-13；比如 mcu 上报温度 30，实际温度＝1*30-13=17 度；	13 这个数值后期可能会调整
8	环境湿度	只读（环境）	uchar	1	0～100，无修正	
9	红外探测	只读（环境）	bit	1	0：探测到无障碍；1：探测到有障碍	
10	报警 1	只读（报警）	bit	1	长按 Key3 键 4 秒触发	模拟报警 1
11	报警 2	只读（报警）	bit	1	长按 Key4 键 4 秒触发	模拟报警 2
12	LED 故障	只读（故障）	bit	1	短按 Key3 键触发	短按 Key3 时，LED 和电机故障同时触发
13	电机故障	只读（故障）	bit	1	短按 Key3 键触发	短按 Key3 时，LED 和电机故障同时触发
14	温湿度传感器故障	只读（故障）	bit	1	短按 Key4 键触发	短按 Key4 时，LED 和电机故障同时触发
15	红外传感器故障	只读（故障）	bit	1	短按 Key4 键触发	短按 Key4 时，LED 和电机故障同时触发

1.3.模组交互功能表

序号	名　称	内容	备　注
1	MCU Reset wifi 模组	当 MCU 需要重启 wifi 模块的时候（比如心跳超时），可以发送此命令	参见示例 22
2	通知模组进入 Air Link 配置模式	在设备上触发某个按键，发送此命令让 wifi 模块进入 Air Link 配置模式	参见示例 23
3	通知模组进入 SoftAP 配置模式	在设备上触发某个按键，发送此命令让 wifi 模块进入 SoftAP 配置模式	参见示例 24
4	获取 MCU 信息	Wifi 模块获取 MCU 的各种版本等信息	参见示例 1、2
5	心跳	Wifi 模块发起，mcu 回应，心跳超时后，mcu 重启 wifi 模块	参见示例 18、19
6	Wifi 模块重启 MCU	当 wifi 模块升级 mcu 等场景，发此命令重启 mcu	参见示例 25
7	无效命令	当校验码等错误时，需要返回此命令	参见示例 20、21

2. 产品（设备端）工作流程


MCU 上电，初始化完成后给 wifi 模块上电，wifi 模块初始化
模块向 MCU 询问必要信息，MCU 返回信息
失败
成功
进入正常工作状态，实时检测串口事件以及按键事件以及状态变化事件
按键事件发生
Wifi 发送控制命令
Wifi 发送查询命令
Wifi 发送心跳命令
Wifi 状态更新
外设状态变化
按键处理，发送命令给 wifi 模块
返回 ACK，执行命令
返回 MCU 当前存储的所有状态信息
返回心跳
更新外设关于 wifi 的状态显示
更新缓存
等待 wifi 模块 ACK
命令执行完毕后立即上报状态，模块返回 ACK
向 wifi 模块上报状态
系统事件处理（定时器、心跳检查等）

3. 约定

3.1. 流程约定

1) MCU 先上电，初始化完成后，给模块上电；
2) 模块初始化；
3) 模块向 MCU 询问必要信息，MCU 返回信息（见协议举例）；
4) 进入正常工作循环；
   a) 模块给 MCU 下发控制命令（见协议举例）；
   b) MCU 返回确认，表示收到命令，正在执行（见协议举例）；
   c) 执行完新控制命令后，无论状态是否发生变化，MCU 都需要通知模块最新状态（见协议举例）；
   d) 若 MCU 检测到环境属性变化或者用户在设备上按键引起的状态变化，MCU 需要通知模块最新状态，但是其发送的频率不能快于 2 秒每次（见协议举例）；
   e) 若环境状态一直不变化，MCU 需要每隔 10 分钟定期主动上报当前状态（见协议举例）；
   f) 模块会向 MCU 发送心跳，MCU 收到后按照格式返回即可（见协议举例）；MCU 连续 180 秒收不到模块的数据，即可认为模块异常，可以给模块重新上电；

3.2. 通讯协议约定

命令格式：header(2B)=0xFFFF, len(2B), cmd(1B), sn(1B), flags(2B),DATA(XB)，checksum(1B)

说明：

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) 命令字（cmd）表示具体的命令含义，详见协议举例；
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) 标志位（flag），本产品填写默认 0；
6) p0 数据区（DATA）,详细参见 p0 数据区约定；
7) 检验和(checksum)的计算方式为从 len～DATA，按字节求和；

8) 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；
9) 多于一个字节的整型数字以大端字节序编码（网络字节序）；
10) 数字均用 16 进制表示；

3.3. p0 数据区约定

1) 模块向 MUC 发送控制命令时携带 p0 命令和命令标志位以及可写数据区，即：

   p0 命令(1B)，命令标志位(1B)，开启/关闭红色灯+设定 LED 组合颜色(1B),设定 LED 红色值(1B),设定 LED 绿色值(1B),设定 LED 蓝色值(1B),设定电机转速(2B)。
2) MCU 主动发送状态时或者回复 wifi 模块的状态查询时携带 p0 命令和完整数据区，即：

   p0 命令(1B),开启/关闭红色灯+设定 LED 组合颜色(1B),设定 LED 红色值(1B),设定 LED 绿色值(1B),设定 LED 蓝色值(1B),设定电机转速(2B),红外探测(1B),环境温度(1B),环境湿度(1B),报警码(1B),故障码(1B)。
3) 数据区会自动合并布尔和枚举变量，且有严格的顺序，不可任意改变。

3.4. 包头排重约定

原则：我们的包头是两个连续的FF FF，如果此包中还有某字节出现FF，仅在传输和接收的时候处理，其他环节按正常数据处理；
举例：

1) 某设备上报状态帧：FF FF 00 15 05 03 00 00 04 01 01 02 03 01 00 00 00 32 FF 20 00 03 7D， 除包头外，出现了FF；
2) 在程序内部，作为正常的数据去处理；
3) 当需要传输时，将除包头外的FF后，增加一个55字节，其他不变；
4) 将上述数据处理成：FF FF 00 15 05 03 00 00 04 01 01 02 03 01 00 00 00 32 FF 55 20 00 03 7D，长度不变，校验码不变；
5) 接收方在接收过程中，如果收到字节是FF，及判断第二个字节是否也是FF，如果是FF，表示一个新包，按照新包处理；
6) 如果第二个字节是55，直接丢弃，不算接收长度，继续接收下一个字节；
7) 直到按照长度接收完，或者碰到下一个连续的FF FF；

4. 协议举例

序号	含义	数据	说明
1	模块向 MCU 询问必要信息 cmd: 01 p0_cmd: NULL	FF FF 00 05 01 01 00 00 07	FF FF 包头 00 05 长度，表示后面还有 5 个字节 01 wifi 要获取 mcu 基本信息的命令 01 序列号 00 00 标志位，保持 0 即可 07 校验值，从长度到校验和前的字节相加值
2	MCU 给 WIFI 模块返回基本信息 cmd: 02 p0_cmd: NULL	FF FF 00 47 02 01 00 00 30 30 30 30 30 30 30 34 30 30 30 30 30 30 30 34 30 30 30 30 30 30 30 31 30 30 30 30 30 30 30 31 36 66 33 30 37 34 66 65 34 33 38 39 34 35 34 37 61 34 66 31 33 31 34 62 64 37 65 33 61 65 30 62 00 00 F4	FF FF 包头 00 47 长度，表示后面有 71 个字节（47 为 16 进制） 02 mcu 给 wifi 返回基本信息的命令 01 序列号，需要和上述 wifi 的查询包中的 sn 相同 00 00 标志位，保持 0 即可 30 30 30 30 30 30 30 34 机智云协议版本号（本产品：字符串 00000004） 30 30 30 30 30 30 30 34 P0 协议版本号（本产品：字符串 00000004） 30 30 30 30 30 30 30 31 MCU 硬件版本号（本产品：字符串 00000001） 30 30 30 30 30 30 30 31 MCU 软件版本号（本产品：字符串 00000001） 36 66 33 30 37 34 66 65 34 33 38 39 34 35 34 37 61 34 66 31 33 31 34 62 64 37 65 33 61 65 30 62 产品的 product_key（本产品：字符串 6f3074fe43894547a4f1314bd7e3ae0b） 00 00 绑定超时时间（00 00 为可以随时绑定） F4 校验值，从长度到校验和前的字节相加值 其他：一问一答，命令回复者需要返回发起者相同的 sn
3	WIFI 读取 MCU 属性 cmd: 03 p0_cmd: 02	FF FF 00 06 03 02 00 00 02 0d	FF FF 包头 00 06 长度，后面有 6 个字节 03 wifi 向 mcu 发送数据的命令 02 序列号 00 00 标志位，保持 0 即可 02 查询状态命令 0d 校验值，从长度到校验和前的字节相加值

4	MCU 返回当前属性值 cmd: 04 p0_cmd: 03	开启/关闭红色灯+设定 LED 组合颜色(1B),设定 LED 红色值(1B),设定 LED 绿色值(1B),设定 LED 蓝色值(1B),设定电机转速(2B),红外探测(1B),环境温度(1B),环境湿度(1B),报警码(1B),故障码(1B) FF FF 00 11 04 02 00 00 03 01 AA BB CC 00 06 00 25 36 01 02 B0	FF FF 包头 00 11 长度，后面有 17 个字节 04 mcu 向 wifi 发送数据的命令 02 序列号，和 00 00 标志位，保持 0 即可 03 mcu 响应 wifi 的查询状态命令 01 二进制为 0000 0001，最右侧第 0 位，最左为第 7 位；依次是：开启/关闭红色灯（1bit）、设定 LED 组合颜色（2bit），其余预留； AA 设定 LED 红色值 BB 设定 LED 绿色值 CC 设定 LED 蓝色值 00 06 设定电机转速 25 环境温度 36 环境湿度 01 报警码,第一个报警发生 02 故障码，第二个故障发生 B0 校验值，从长度到校验和前的字节相加值 其他：一问一答，命令回复者需要返回发起者相同的 sn
5	MCU 主动上报当前的属性值 （包含故障信息） cmd: 05 p0_cmd: 04	FF FF 00 14 05 03 00 00 04 01 AA BB CC 00 06 00 25 36 01 02 B6	与第 4 条不同的地方是： 1、红色 05 表示 MCU 主动向模块发送数据； 2、蓝色 03 表示新的 sn，这次有 MCU 生成； 3、红色 04 表示 MCU 发出的是状态上报； 4、紫色的校验码不同； 5、含义参见示例 4；
6	WIFI 收到后的响应 cmd: 06 p0_cmd: NULL	FF FF 00 05 06 03 00 00 0E	1、固定格式 2、一问一答，命令回复者需要返回发起者相同的 sn；

7	模块向 MCU 发送命令： 电机正传 cmd: 03 p0_cmd: 01	FF FF 00 0D 03 04 00 00 01 01 01 00 00 00 00 00 00 1F	FF FF 包头 00 0D 长度 03 wifi 向 mcu 发送数据的命令 04 序号 00 00 标志位，保持 0 即可 01 控制类命令 01 控制属性标志组合，二进制表示是 0000 0001， 最右为第 0 位，最左为第 7 位，依次表示第一个...第 7 个可写属性是否要被执行，0000 0001，表示第 1 个可写属性也就是开启/关闭红色灯要执行，其余的属性值忽略； 01 00 00 00 00 00 可写属性数据 1F 校验值，从长度到校验和前的字节相加值 说明： 1) 模块给 mcu 发送控制指令时，会含有 mcu 所有的可写属性信息，这个属性信息和 mcu 给模块上报的可写属性信息格式以及顺序完全相同（在定义产品的时候就确定了），模块可以根据需要设定控制属性标志组合，用位表示，最右为第 0 位，最左为最高位，属性顺序和位顺序对应，将想控制的属性对应的位置 1，表示要控制该属性； 2) 本产品中，可以控制的属性依次有：开启/关闭红色灯，设定 LED 组合颜色,设定 LED 红色值,设定 LED 绿色值,设定 LED 蓝色值,设定电机转速； 3) 可以单独控制一个属性，比如单独设定开启/关闭红色灯，此时为 0x01， 用位描述是 0000 0001； 4) 也可以一次控制多个属性，比如同时设定开启/关闭红色灯和设定 LED 组合颜色，此时为 03，用位描述是 0000 0011； 5) 或者同时设定所有的属性值，此时为 0x3f，用位描述是 0011 1111； 6) 此字段的长度根据定义产品时的可写属性个数自动确定，本产品共有 6 个可写属性，一个字节即可表示，若定义 9 个可写属性，则自动用 2 个字节表示，依次类推； 7）本例中，仅设定一个属性值（第一个），属性值为 01，表示要开启/关闭红色灯；
8	MCU 收到后回复模块 （表示 MCU 收到命令了，正在执行） cmd: 04 p0_cmd: NULL	FF FF 00 05 04 04 00 00 0d	FF FF 包头 00 05 长度 04 mcu 响应 wifi 的命令 04 序号，与 wifi 发出的数据包中的 sn 相同 00 00 标志位，保持 0 即可 0d 校验码
9	MCU 执行完后，无论状态是否变化，都需要上报最新的状态 cmd: 05 p0_cmd: 04	FF FF 00 14 05 03 00 00 04 01 AA BB CC 00 06 00 25 36 01 02 B6	参见示例 5
10	WIFI 收到后的响应 cmd: 06 p0_cmd: NULL	FF FF 00 05 06 03 00 00 0E	参见示例 6
11	模块向 MCU 发送命令： 设定 LED 颜色粉色 cmd: 03 p0_cmd: 01	FF FF 00 0D 03 04 00 00 01 02 06 00 00 00 00 00 00 1D	参见示例 7，后期流程参考示例 8、9、10

12	模块向 MCU 发送命令： 设定 LED 颜色自定义 cmd: 03 p0_cmd: 01	FF FF 00 0D 03 04 00 00 01 02 00 00 00 00 00 00 17	参见示例 7，后期流程参考示例 8、9、10；颜色自定义时，才能设定红、绿、蓝值；
13	模块向 MCU 发送命令： 设定 LED 红色值 0xCC， 绿色值 0xBB，蓝色值 0xAA cmd: 03 p0_cmd: 01	FF FF 00 0D 03 04 00 00 01 1C 00 CC BB AA 00 00 62	参见示例 7，后期流程参考示例 8、9、10
14	模块向 MCU 发送命令： 电机正转，转速为 1 挡 cmd: 03 p0_cmd: 01	FF FF 00 0D 03 04 00 00 01 20 00 00 00 00 00 06 3B	红色 20(0010 0000)表示设置第 6 个可写属性（电机），06 表示正向转速 1
15	模块向 MCU 发送命令： 电机负转，转速为 2 挡 cmd: 03 p0_cmd: 01	FF FF 00 0D 03 04 00 00 01 20 00 00 00 00 00 03 38	红色 20(0010 0000)表示设置第 6 个可写属性（电机），03 表示逆向转速 2
16	模块向 MCU 发送命令： 电机停止 cmd: 03 p0_cmd: 01	FF FF 00 0D 03 04 00 00 01 20 00 00 00 00 00 05 3A	红色 20(0010 0000)表示设置第 6 个可写属性（电机），05 表示停止
17	模块向 MCU 发送命令： 设定 LED 颜色自定义，红色值 0xCC， 绿色值 0xBB，蓝色值 0xAA， 电机正转档位 3 cmd: 03 p0_cmd: 01	FF FF 00 0D 03 04 00 00 01 3E 00 CC BB AA 00 08 8C	红色 3E(0011 1110)表示同时设置第 2、3、4、5、6 五个可写属性，即设定 LED 颜色自定义、红色值 CC、绿色值 BB、蓝色值 AA、电机转速正向 3 挡
18	模块向 MCU 发送心跳 cmd: 07 p0_cmd: NULL	FF FF 00 05 07 06 00 00 12	FF FF 包头 00 05 长度 07 模块向 MCU 发送心跳的命令 06 序号 00 00 标志位，保持 0 即可 12 校验码
19	MCU 向模块回复 cmd: 08 p0_cmd: NULL	FF FF 00 05 08 06 00 00 13	FF FF 包头 00 05 长度 08 MCU 向模块回复心跳的命令 06 序号，与发起者相同 00 00 标志位，保持 0 即可 13 校验码

20	MCU 通知模组进入 Air Link 配置模式 cmd: 09 p0_cmd: NULL	FF FF 00 06 09 03 00 00 02 14	在设备上触发某个按键，发送此命令让 wifi 模块进入 Air Link 配置模式
21	MCU 通知模组进入 SoftAP 配置模式 cmd: 09 p0_cmd: NULL	FF FF 00 06 09 08 00 00 01 18	在设备上触发某个按键，发送此命令让 wifi 模块进入 SoftAP 配置模式
22	模组回复 MCU cmd: 0A p0_cmd: NULL	FF FF 00 05 0A 08 00 00 17	模组收到进配置命令后对 MCU 的回复
23	MCU Reset wifi 模组 cmd: 0B p0_cmd: NULL	FF FF 00 05 0B 06 00 00 16	当 MCU 需要重置（reset）wifi 模块的时候（比如用户按下了重置按键），可以发送此命令
24	Wifi 模组回复 MCU cmd: 0C p0_cmd: NULL	FF FF 00 05 0C 06 00 00 17	Wifi 模组收到后，先发送此确认包，再重置
25	Wifi 模块给 Mcu 发送当前状态 cmd: 0D p0_cmd: NULL	FF FF 00 07 0D 00 00 00 00 02 16	蓝色部分是2个字节的wifi_status ，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 onboarding 模式,0:关闭,1:开启; 第 3 位:是否开启 binding 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
26	Mcu 回复 wifi 模组 cmd: 0E p0_cmd: NULL	FF FF 00 05 0E 00 00 00 13	Mcu 收到后回复确认
27	Wifi 模块重启 MCU cmd: 0F p0_cmd: NULL	FF FF 00 05 0F 01 00 00 15	Wifi 模块在升级 MCU 等场景发此命令重启 MCU
28	Mcu 回复 wifi 模组 cmd: 10 p0_cmd: NULL	FF FF 00 05 10 01 00 00 16	MCU 发送此确认后，等待 600 毫秒后重启

29	MCU 收到来自模组的不识的命令或者校验错误 cmd: 11 p0_cmd: NULL	FF FF 00 05 11 06 00 00 00 1C	收到不识别的命令或者检测出来校验码异常后，用此命令代替原来的 ACK，并赋值正确的错误类型； 1 - checksum 不正确 2 - 命令不可识别 3 - 其它 0, 4~255 保留不用
30	模组收到来自 MCU 的不识别的命令或者校验错误 cmd: 12 p0_cmd: NULL	FF FF 00 05 12 06 00 00 00 1D	收到不识别的命令或者检测出来校验码异常后，用此命令代替原来的 ACK，并赋值正确的错误类型； 错误码同上