

机智云平台标准接入协议

之App与云端通讯

(v4.0.4)

修订历史

版本	修订内容	修订人	修订日期
4.0.0	V4版协议	刘冠华	2014-11-19
4.0.1	删除“透传业务指令”中的“业务指令长度”长度	刘冠华	2014-12-29
4.0.2	添加“透传业务指令（带ACK）”	刘冠华	2015-04-18
4.0.3	更正：App订阅接收设备上报的实时状态的Topic应为：dev2app/<DID>/<AppClientId>而非dev2app/<DID>/#（后者会导致收到设备发给其它App的消息）	刘冠华	2015-06-19
4.0.4	修正v4.0.3的变更内容：App订阅接收设备上报的实时状态的Topic应为：dev2app/<DID>和dev2app/<DID>/<AppClientId>（前者用于接收设备的状态广播）。 规定AppClientId的字符取值范围为只能包含大小写英文字母，数字0到9，即字符[0-9a-zA-Z]。	刘冠华	2015-07-07

目录

[通讯模型](#)

[通讯流程](#)

[约定](#)

[通讯协议](#)

[用户注册](#)

[用户登陆](#)

[绑定设备](#)

[解绑设备](#)

[获取绑定列表](#)

[心跳](#)

[透传业务指令](#)

[透传业务指令（带ACK）](#)

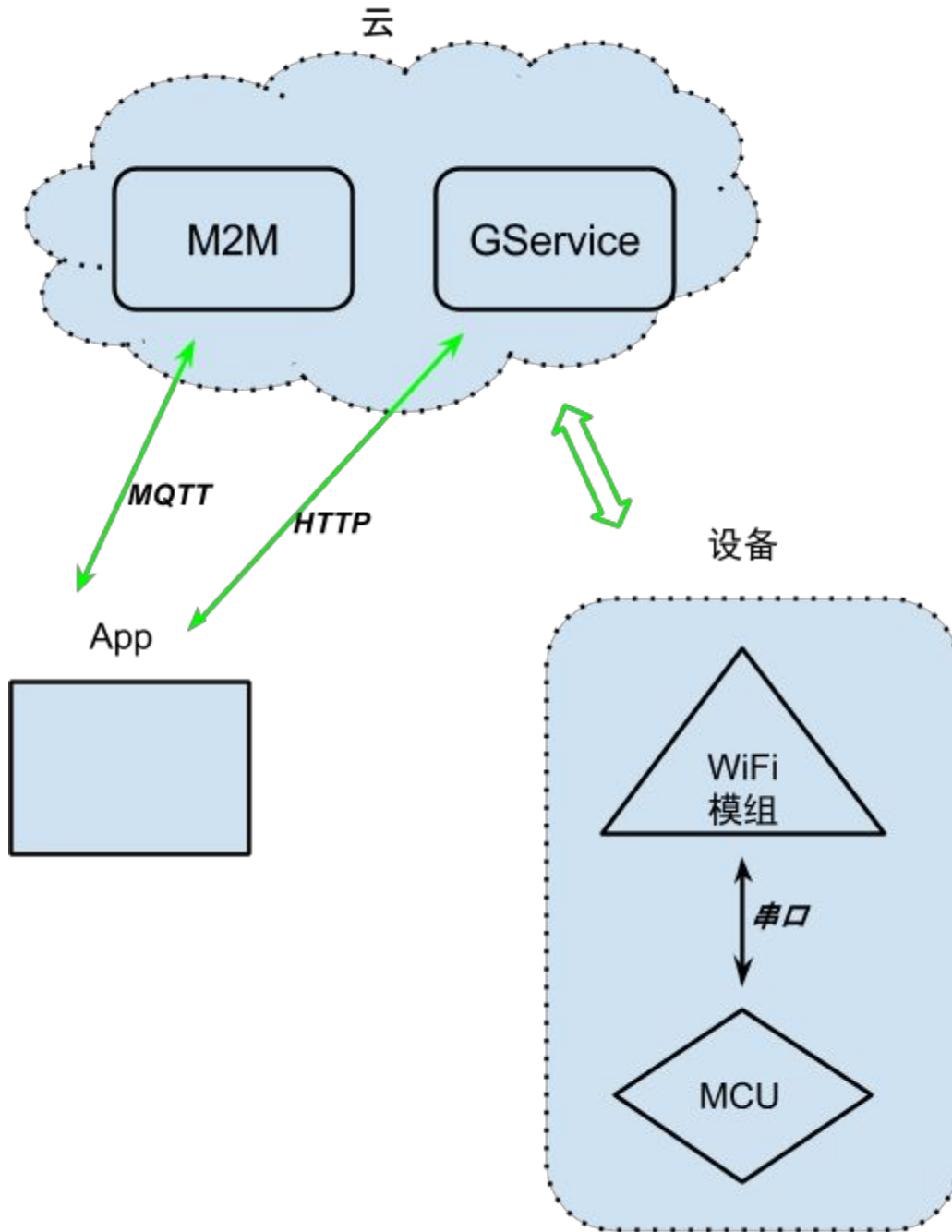
[设备在线状态变更通知](#)

[设置设备的日志级别与指示灯开关](#)

[设备向云端及App发送日志](#)

1. 通讯模型

App与云端通讯采用MQTT和HTTP两种方式。HTTP方式主要是提供用户注册，设备绑定等接口。而MQTT用于接收设备上报的实时状态及向设备发送远程控制指令。MQTT是建立在TCP基础上的长连接通讯协议，详细请参考文档《[MQTT_V3.1_Protocol_Specific.pdf](#)》。



2. 通讯流程

App与云端的大循环通讯主要包括以下的通讯过程。

- 用户注册。
- 用户登陆。
- 绑定设备。绑定设备后才能控制设备。
- 获取绑定设备列表。每次打开App后可以从云端获取设备列表并让用户选择操控的设备。
- 控制设备与接收设备状态上报。App主动与M2M服务器建立长连接并保持该连接，通过该连接向设备发送控制指令和接收设备上报的实时状态。

3. 约定

3.1. 协议阅读说明

- 可变长度：由 1 ~ 4 个字节(B)表示本可变长度字段后一直到数据包结尾的字节数。编码解码方式请参考《[MQTT_V3.1_Protocol_Specific.pdf](#)》第6页Remaining Length。
- 长度：没有注明为可变长度的都为普通长度，一般由一个(1B)或两个字节(2B)组成。若多于一个字节组成，采用大端编码方式，即高字节在前，低字节在后。

3.2. 云端HTTP GService服务地址

- 服务器地址为：api.gizwits.com
- 服务端口号为：80

3.3. 云端M2M服务器地址

- 服务器地址为：m2m.gizwits.com
- 服务端口号为：1883(TCP)或8883(SSL)

3.4. App与M2M服务器建立MQTT连接后，需订阅下以的主题以接收M2M服务器下发的消息

- 接收服务器的消息响应：[ser2cli_res/<AppClientId>](#) (<AppClientId>指代App的当前MQTT Client Identifier)
- 接收设备上报的实时状态：[dev2app/<DID>](#)和[dev2app/<DID>/<AppClientId>](#) (<DID>指代设备的DID，<AppClientId>指代App的当前MQTT Client Identifier)

4. 通讯协议

4.1. 用户注册

App ⇒ GService, HTTP。

请见<http://docs.gizwits.apiary.io/reference/api-for-apps/users/create-user>。

4.2. 用户登陆

用户登陆M2M服务器分两个步骤完成。先通过GService HTTP获得用户的uid和token，再以appid，uid和token登陆M2M服务器。

设备 ⇒ GService, HTTP。获得用户的uid和token。

请见<http://docs.gizwits.apiary.io/reference/api-for-apps/login/login>。

设备 ⇒ M2M服务器, TCP, MQTT CONNECT。凭appid, uid和token登陆M2M服务器。参数Client Identifier填App随机生成的以"usr"开头的AppClientId(总长度不超过23个字节, 且只能包含大小写英文字母, 数字0到9, 即字符[0-9a-zA-Z]), 参数User Name填设备"2\$<appid>\$<uid>"(例如appid为"123",uid为"abc"则User Name为"2\$123\$abc"), 参数Password填token。Keep Alive timer的最大有效值为600秒, 超出该最大有效值, M2M服务器以最大有效值处理。Keep Alive timer的值建议设为120秒。

请见《[MQTT_V3.1_Protocol_Specific.pdf](#)》第15页3.1. CONNECT。

M2M服务器 ⇒ 设备, TCP, MQTT CONNACK。

请见《[MQTT_V3.1_Protocol_Specific.pdf](#)》第18页3.2. CONNACK。

4.3. 绑定设备

App ⇒ GService, HTTP。

请见<http://docs.gizwits.apiary.io/reference/api-for-apps/bindings/bind-devices>。

4.4. 解绑设备

App ⇒ GService, HTTP。

请见<http://docs.gizwits.apiary.io/reference/api-for-apps/bindings/unbind-devices>。

4.5. 获取绑定列表

App ⇒ GService, HTTP。

请见<http://docs.gizwits.apiary.io/reference/api-for-apps/bindings/get-bound-devices>。

4.6. 心跳

App (以下简称客户端) 与服务器建立连接后, 需要定期向M2M服务器发送心跳, M2M服务器收到客户端的心跳请求后回复响应。

客户端 ⇒ M2M服务器, TCP, MQTT PINGREQ。客户端向M2M服务器发送心跳请求。

请见《[MQTT_V3.1_Protocol_Specific.pdf](#)》第33页3.12. PINGREQ。

M2M服务器 ⇒ 客户端, TCP, MQTT PINGRESP。M2M服务器回应心跳请求。

请见《[MQTT_V3.1_Protocol_Specific.pdf](#)》第34页3.13. PINGRESP。

客户端向M2M服务器发送心跳请求的最快频率为180秒内60次，心跳频率快于此值M2M服务器会主动断开与客户端的连接。建议客户端60秒发一次心跳。

客户端如连续2两次没有收到M2M服务器的心跳响应，可以认为与M2M服务器的连接已断开，尝试重连。

关于客户端与M2M服务器的重连策略。当客户端与服务器的连接异常断开后，客户端会重连服务器。第一次与服务器的连接断开后，马上重连，如连接不上，10秒后重连，如还是连接不上，20秒后重连，依此类推，每次重连失败后都延长10秒后再重连。如客户端重启后，重连间隔从头从0秒计起。

4.7. 透传业务指令

App登陆M2M服务器后，App与设备相互间可以透过M2M服务器发送和接收业务指令，业务指令视具体的产品定制。

App透过M2M服务器向设备传送业务指令，App ⇒ M2M服务器，MQTT PUBLISH。主题为 `app2dev/<DID>/<AppClientId>` (<DID>指设备的DID，<AppClientId>指发出消息的App的当前MQTT Client Identifier)。

序号	字段名称	字节长度(B)	内容说明
1	固定包头	4	0x00000003
2	可变长度	1~4	len(Flag...业务指令)
3	Flags	1	0x00
4	命令字	2	0x0090
5	业务指令	max 65535	业务指令

App接收设备透过M2M服务器向App传送业务指令，M2M服务器 ⇒ App，MQTT PUBLISH。主题为 `dev2app/<DID>/<AppClientId>`，如设备想向所有与其有绑定关系的App传送业务指令，则主题为 `dev2app/<DID>` (<DID>指设备的DID，<AppClientId>指接收消息的App的当前MQTT Client Identifier)。

序号	字段名称	字节长度(B)	内容说明
1	固定包头	4	0x00000003
2	可变长度	1~4	len(Flag...业务指令)
3	Flags	1	0x00

4	命令字	2	0x0091
5	业务指令	max 65535	业务指令

4.8. 透传业务指令（带ACK）

App登陆M2M服务器后，App与设备相互间可以透过M2M服务器发送和接收业务指令，业务指令视具体的产品定制。

发送需要ACK的业务指令，App ⇒ M2M服务器 或 M2M服务器 ⇒ App，MQTT PUBLISH。如是从App发往设备则主题为 `app2dev/<DID>/<AppClientId>`，如是由设备发往App则主题为 `dev2app/<DID>` 或 `dev2app/<DID>/<AppClientId>`（<DID>指设备的DID，<AppClientId>指发出消息的App的当前MQTT Client Identifier）。

序号	字段名称	字节长度(B)	内容说明
1	固定包头	4	0x00000003
2	可变长度	1~4	len(Flag...业务指令)
3	Flag	1	0x00
4	命令字	2	0x0093
5	包序号(sn)	4	消息的序号，用于消息的ACK
6	业务指令	max 65535	业务指令

消息ACK，M2M服务器 ⇒ App 或 App ⇒ M2M服务器，MQTT PUBLISH。如是从App发往设备则主题为 `app2dev/<DID>/<AppClientId>`，如是由设备发往App则主题为 `dev2app/<DID>` 或 `dev2app/<DID>/<AppClientId>`（<DID>指设备的DID，<AppClientId>指发出消息的App的当前MQTT Client Identifier）。

序号	字段名称	字节长度(B)	内容说明
1	固定包头	4	0x00000003
2	可变长度	1~4	len(Flag...业务指令)
3	Flag	1	0x00
4	命令字	2	0x0094
5	包序号(sn)	4	对应发送包的包序号
6	业务指令	max 65535	业务指令

注意：M2M服务器不会对消息进行ACK，消息的ACK由接收端负责，即为设备或App。

4.9. 设备在线状态变更通知

当用户绑定的设备有上线或下线变化时，M2M服务器会推送设备的在线状态给App。

M2M服务器推送设备的在线状态给App，M2M服务器 ⇒ App，MQTT PUBLISH。主题为 `dev2app/<DID>` (<DID>指设备的DID)。

序号	字段名称	字节长度 (B)	内容说明
1	固定包头	4	0x00000003
2	可变长度	1~4	len(Flag...业务协议版本号)
3	Flags	1	0x00
4	命令字	2	0x0092
5	设备MAC长度	2	len(设备MAC)
6	设备MAC	max 32	设备MAC地址
7	设备Passcode长度	2	len(设备Passcode)
8	设备Passcode	max 32	设备Passcode
9	是否在线	1	0为离线，1为在线
10	通用协议版本号	4	0x00000003
11	业务协议版本号长度	2	len(业务协议版本号)
12	业务协议版本号	max 64	保留

4.10. 设置设备的日志级别与指示灯开关

App可以设置设备的日志级别与指示灯开关。

App修改设备日志级别与指示灯开关的指令，App ⇒ M2M服务器，MQTT PUBLISH。主题为 `app2dev/<DID>/<AppClientId>` (<DID>指设备的DID，<AppClientId>指发出消息的App的当前MQTT Client Identifier)。

序号	字段名称	字节长度 (B)	内容说明
1	固定包头	4	0x00000003
2	可变长度	1~4	len(Flag...设置值bit0~bit7)

3	Flags	1	0x00
4	命令字	2	0x0010
5	设置值bit8~bit15	1	bit8 ~ bit15从低位(bit)向高位排列
6	设置值bit0~bit7	1	bit0 ~ bit7从低位(bit)向高位排列，与上面的bit8 ~ bit15一共组成bit0 ~ bit15，各位的定义如下： <ul style="list-style-type: none"> • bit0: Error日志级别的开与关，0为关，1为开 • bit1: Warning日志级别的开与关，0为关，1为开 • bit2: Info日志级别的开与关，0为关，1为开 • bit3: WiFi模组所有指示灯的总开关，0为关，1为开 • bit4~bit15: 保留

设备返回操作确认，M2M服务器 ⇒ App，MQTT PUBLISH。主题为 `dev2app/<DID>/<AppClientId>` (<DID>指设备的DID，<AppClientId>指接收消息的App的当前MQTT Client Identifier)。

序号	字段名称	字节长度(B)	内容说明
1	固定包头	4	0x00000003
2	可变长度	1~4	len(Flag...命令字)
3	Flags	1	0x00
4	命令字	2	0x0011

4.11. 设备向云端及App发送日志

当设备相关的日志级别打开后，设备会透过M2M服务器发送日志给App。

设备向App发送日志，M2M服务器 ⇒ App，MQTT PUBLISH。主题为 `dev2app/<DID>` (<DID>指设备的DID)。

序号	字段名称	字节长度(B)	内容说明
1	固定包头	4	0x00000003
2	可变长度	1~4	len(Flag...产品标识码)
3	Flags	1	0x00
4	命令字	2	0x0012
5	日志级别	1	1为Error，2为Warning，3为Info，
6	标签长度	2	len(标签)

7	标签	max 256	指示日志种类与关键字
8	来源长度	2	len(来源)
9	来源	max 32	指示日志是从哪里来的
10	日志内容长度	2	len(日志内容)
11	日志内容	max 65535	日志的内容，为UTF8编码的字符