# GL-S10 BLE-Gateway User Guide CN

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 更改项 | 作者 | 日期 |
| 1.0 | 第一版 | 何丰 | 2020.04.17 |

## 简介

BLE-Gateway是一个基于C语言开发的蓝牙网关例程，使用的硬件平台是GL-S10。它实现了收发BLE数据并使用MQTT协议上传至互联网，打通了ble终端设备与云平台之间的数据通道（系统框架图如图1）。BLE支持GAP和GATT规范，即可作为master也可作为slave，用以与其他蓝牙设备连接及交换数据。该固件可以使用WiFi作为station连接上级路由，也可以通过wan口网线上网。它内置了标准的MQTT客户端，可以连接至任意标准的MQTT服务器。

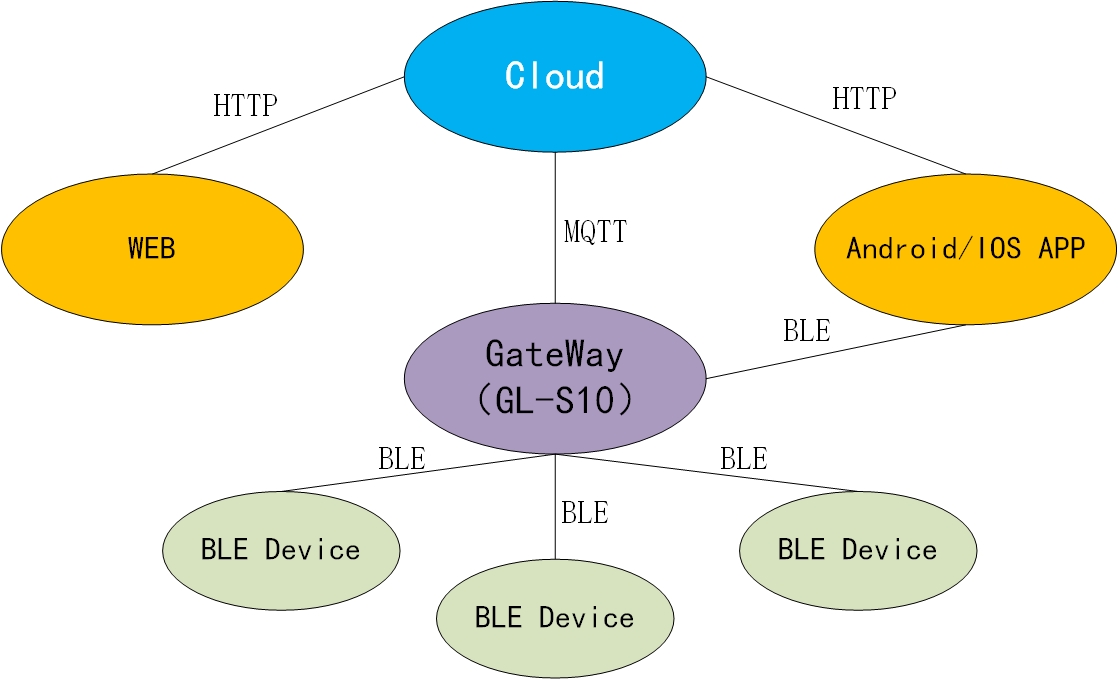


图 0‑1.1 系统通信框架

## 功能清单

### ble

本例程中ble既可作为master，也可同时作为slave。

作为master： 1. 支持发起扫描，扫描结果可传出（见[5 BLE扫描并上传至云平台](#_BLE扫描并上传至云平台)）；

2. 支持主动发起ble连接。

作为slave： 1. 支持发出广播包；

2. 支持被动ble连接，可以使用ble notify。

### 2.2 WiFi

本例程中的WiFi只能作为STA，加入上级路由的WiFi网络。用户可使用其他的ble设备通过ble API配置WiFi参数（见[附录1 – GL-S10 BLE API](#_附录1_--_BLE-Gateway-Demo)），也可配套的APP直接设置。

### 网口

本例程可以通过网口连接互联网，但网口连接与WiFi连接是互斥的，只能二选一。在连接了网口之后WiFi会自动断开；当网口连接断开，WiFi会自动连接。

### LED

GL-S10共有三个LED灯（从右往左分别是led1、led2、led3），本例程中LED灯的显示逻辑见下表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **常态** | **按键按住3-8S内** | **按键按住超过8S** | **OTA升级过程中** |
| **LED1电源灯（绿色）** | **通电亮，断电灭** | **慢闪** | **快闪** | **流水灯** |
| **LED2蓝牙灯（绿色）** | **有连接时亮， 无连接时灭** |
| **LED3网络灯（白色）** | **wifi连接时闪烁，网线连接时常亮，不联网时灭** |

### 2.5 按键

GL-S10只有一个reset按键，本例程中它的功能定义如下：

|  |  |  |
| --- | --- | --- |
|  | **按住3-8S后松开** | **按住8S以上再松开** |
| **按键功能** | **清除当前WiFi及MQTT配置， 但不会断开当前连接** | **OTA升级** |
|

### 2.6 MQTT

本例程中内置了一个标准的MQTT Client。用户可以通过ble API（见[GL-S10 BLE API](#_附录1_--_BLE-Gateway-Demo)）设置MQTT的配置，连上任意的MQTT服务器订阅收发数据。相关的使用样例见[4 MQTT收发测试](#_MQTT收发测试)

注：本例程只支持基于非加密tcp连接的MQTT

### 2.7 OTA

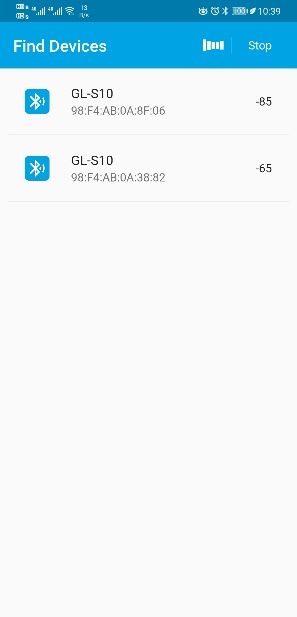
本例程支持OTA升级固件，固件中内置了指向本例程的github目录下release的链接路径。用户可以直接使用按键触发升级（见[2.5 按键](#_2.5_按键)），也可以使用ble API修改OTA的链接路径（见[附录1 – GL-S10 BLE API](#_附录1_--_BLE-Gateway-Demo)），使用自己搭建的http服务器升级成指定的固件。

## 配置APP使用指南

本例程有配套开源的配置APP，安卓用户可以直接在XXX下载apk直接安装，ios用户可以下载源码自行编译调试。

### 3.1 扫描并连接GL-S10

打开配置APP，点击右上角的scan按钮（如下图）。扫描结果已过滤非GL-S10的设备（如下图），如果当前附件有多台GL-S10，用户可根据外壳的标签二维码获取mac地址，从而选定连接哪一台GL-S10。

### 3.2 查看当前设备信息

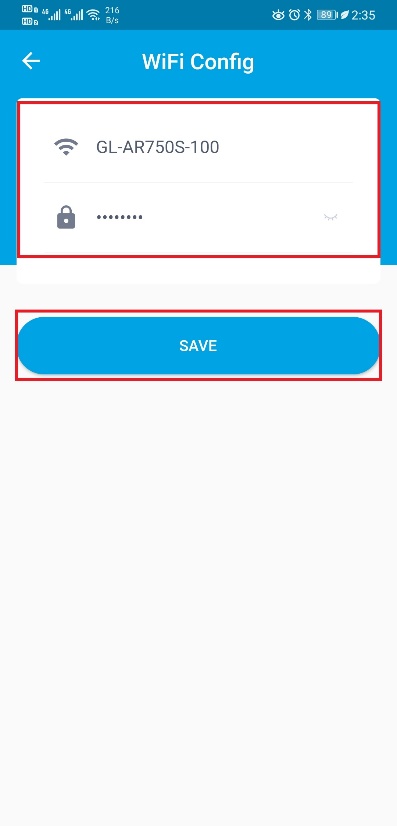
进入“Status”页面后，用户通过下拉刷新，可以获取当前设备的配置信息（如下图）。



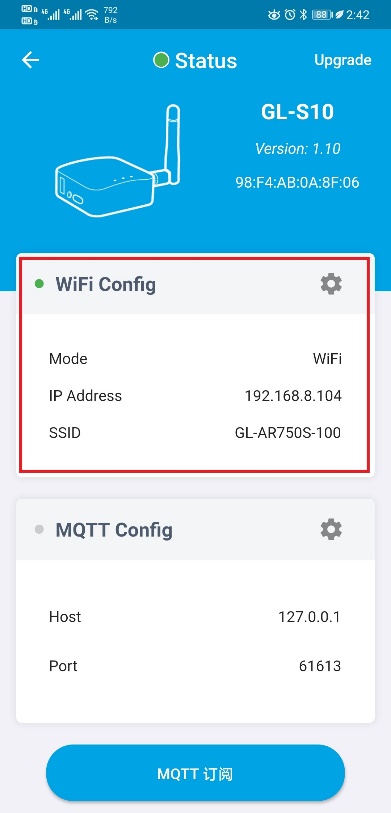
### 3.3 配置WiFi

点击“Status”页面的设置wifi按钮，进入“WiFi Config”页面。

举例：连接SSID为：GL-AR750S-100的WiFi。如下图所示：



设置完后返回“Status”页面，待GL-S10连接上WiFi后再次下拉刷新页面，可以看到WiFi Config的数据已经更新，如下图所示：

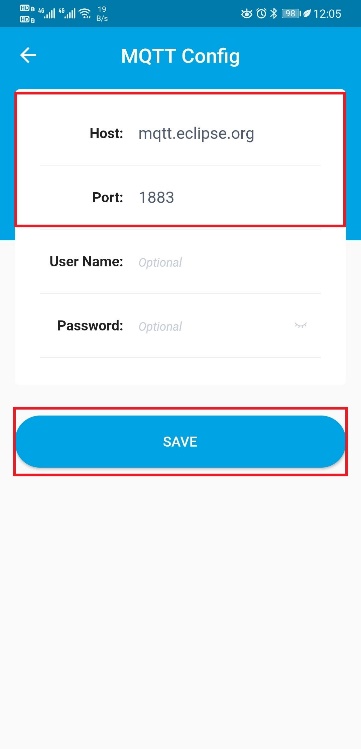


### 3.4 配置MQTT

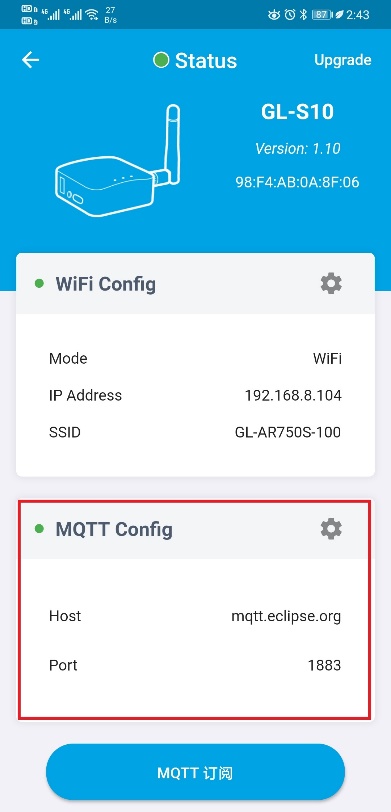
点击“Status”页面的设置MQTT按钮，进入“MQTT Config”页面。

在“HOST”栏输入需连接的MQTT Broker地址，在“PORT”栏输入端口号。如果需要用户名和密码的验证，则在“User Name”和“Password”中输入。

举例：连接地址为“mqtt.eclipse.org”，端口号为：1883，不需要用户名/密码校验的MQTT Broker（用户也可使用该服务器做调试）



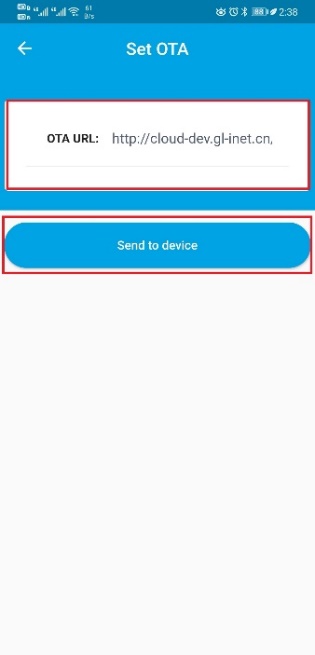
设置完后返回“Status”页面，待GL-S10连接上MQTT Broker后再次下拉刷新页面，可以看到MQTT Config的数据已经更新，如下图所示：



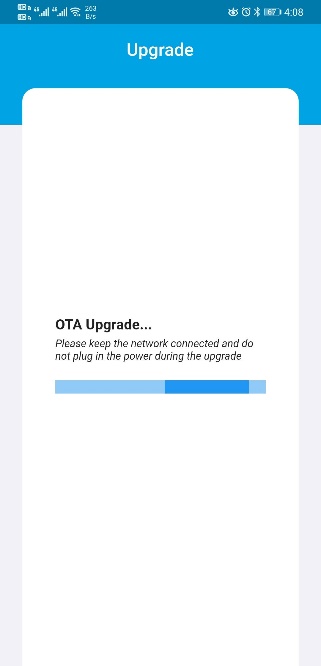
### 3.5 设置OTA

点击“Status”页面的“Upgrade”按钮，进入“Set OTA”页面。

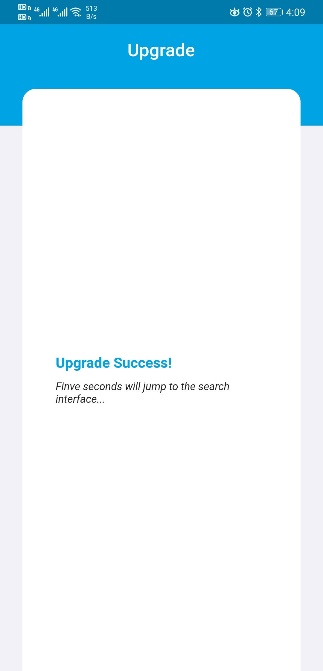
在“OTA URL”栏中输入http服务器的地址，然后点击“Send to device”。

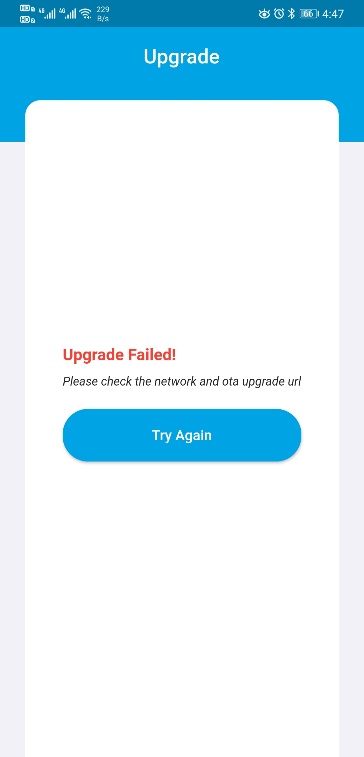


GL-S10收到该指令后会自动开始OTA升级，如下图所示：



升级完成后会返回升级结果：

成功时： 

失败时： 

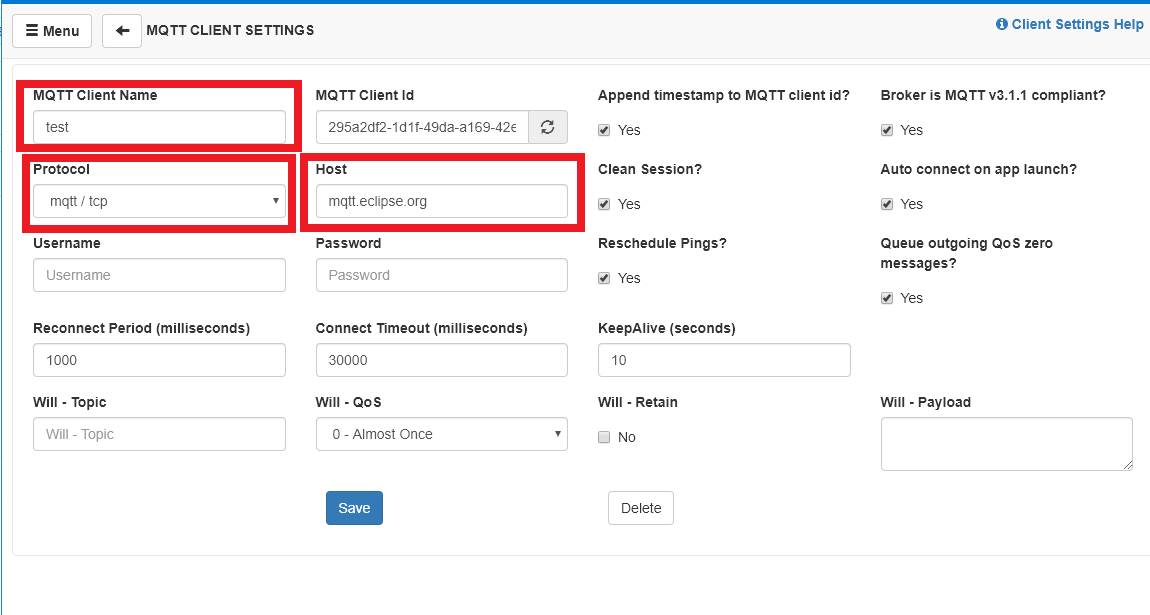
## MQTT收发测试

GL-S10上打通了ble子设备<==>网关<==>云平台的数据链路。用户可以调用ble API往指定的MQTT服务器订阅及发送数据，在以下的演示中手机充当该数据链路中的ble子设备部分。

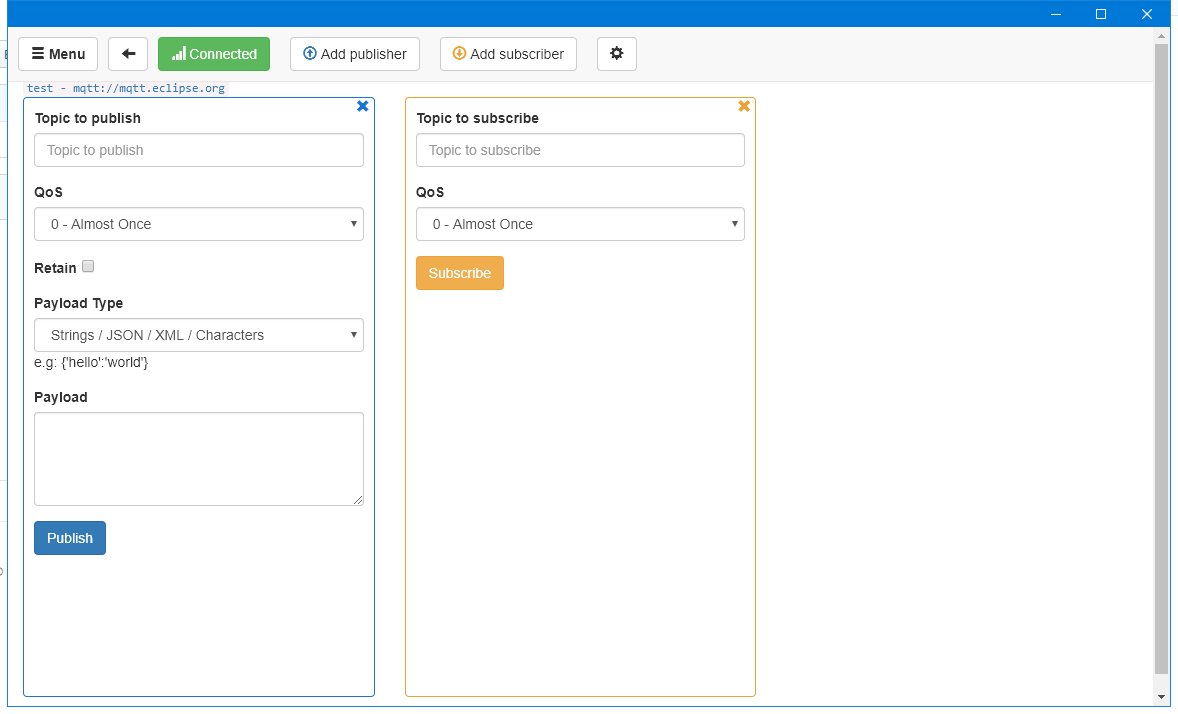
当GL-S10已连接到MQTT服务器后（配置MQTT的步骤见[3.4 配置MQTT](#_3.4_配置MQTT)），用户可以使用另外的MQTT Client来进行收发测试。本文中使用的配测MQTT Client是MQTTBox，可在<http://workswithweb.com/html/mqttbox/downloads.html>下载。

### 4.1 配置连接MQTT Broker

首先使用配置APP设置GL-S10（步骤见[3.4 配置MQTT](#_3.4_配置MQTT)），然后设置MQTTBox连接到同一个MQTT Broker（配置如下图）

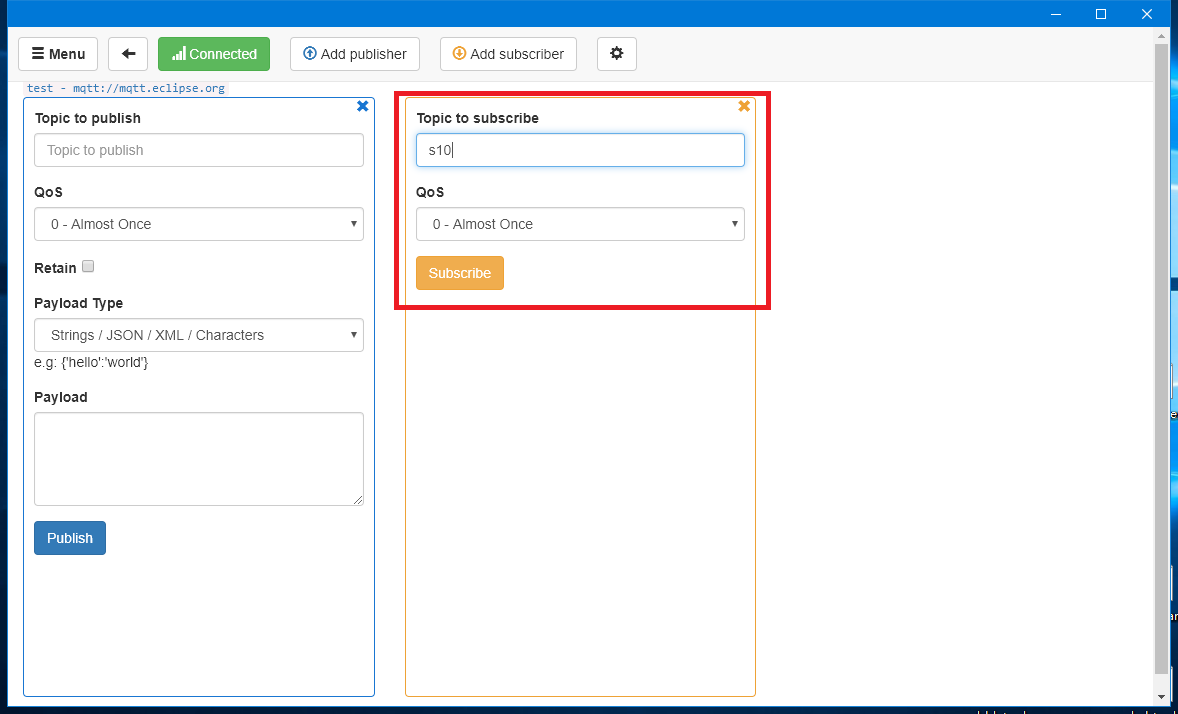


配置成功后会自动连接，连接成功如下图：

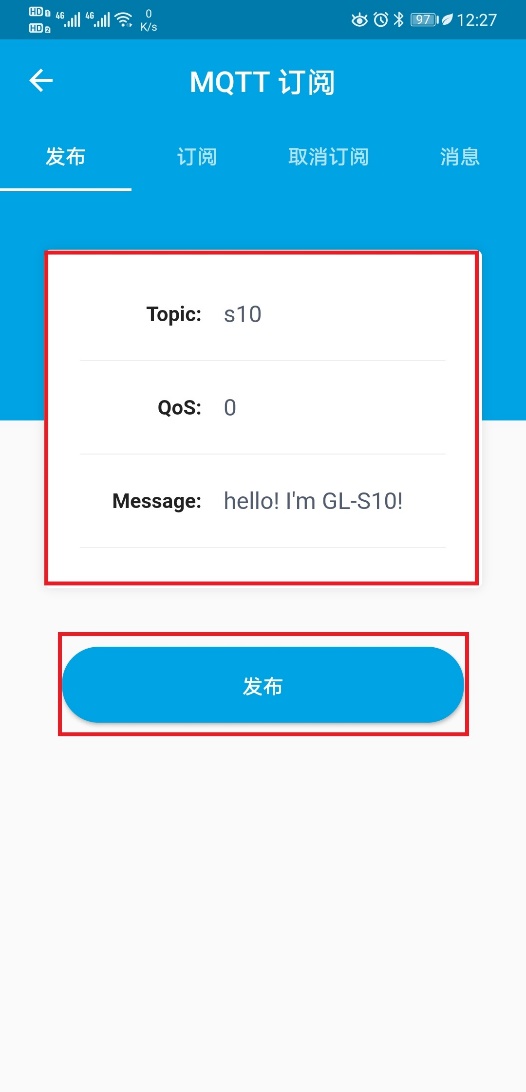


### 4.2 APP发布消息，MQTTBox接收

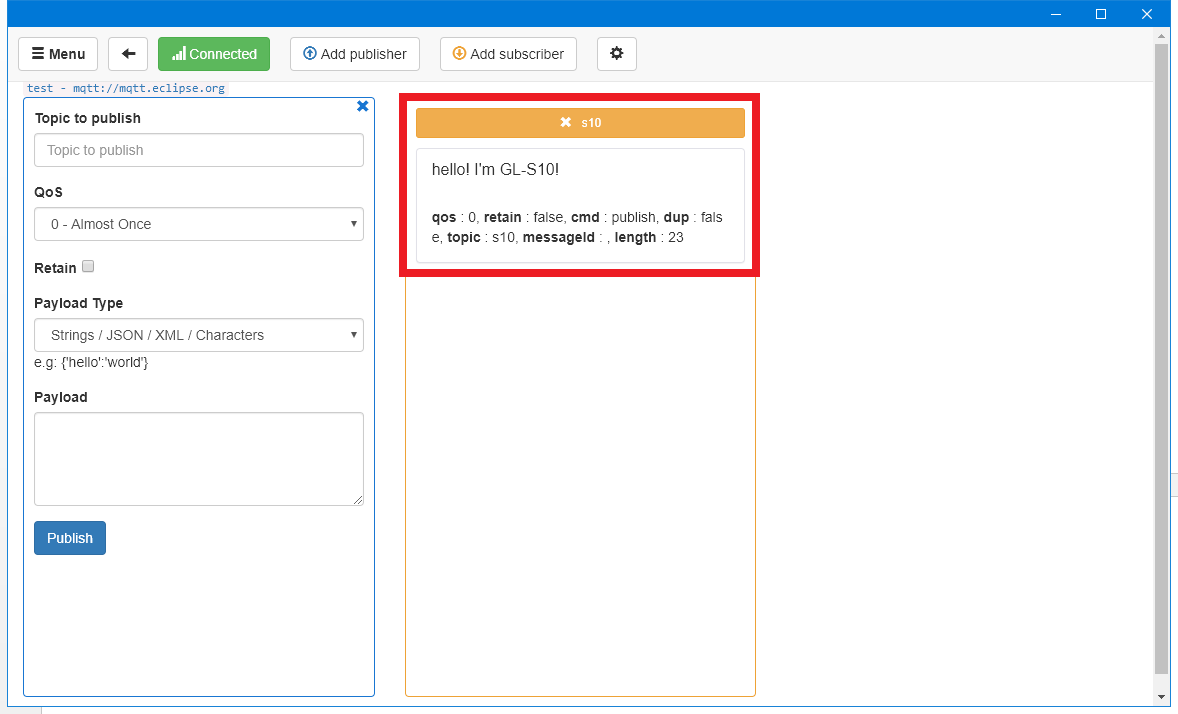
首先设置MQTTBox订阅“s10”主题，如下图：



使用配套APP，在“MQTT订阅页面”往“s10”主题发送“hello！ I’m GL-S10!”，如下图所示：

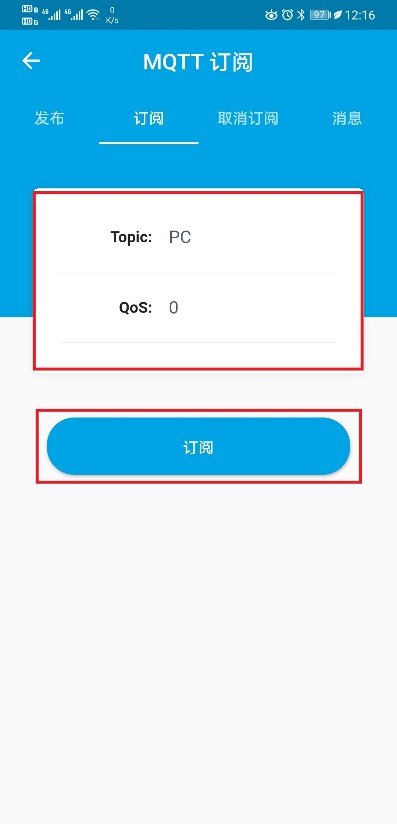


MQTTBox接收到数据，如下图所示：

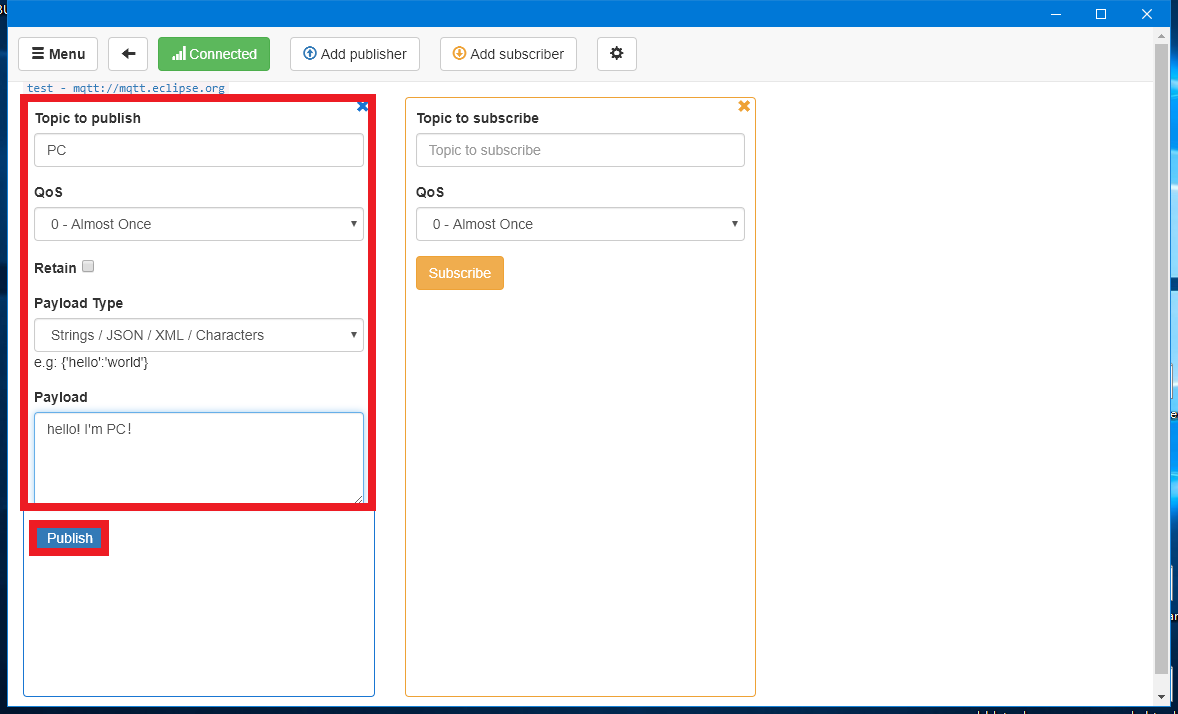


### 4.3 MQTTBox发布消息，APP接收

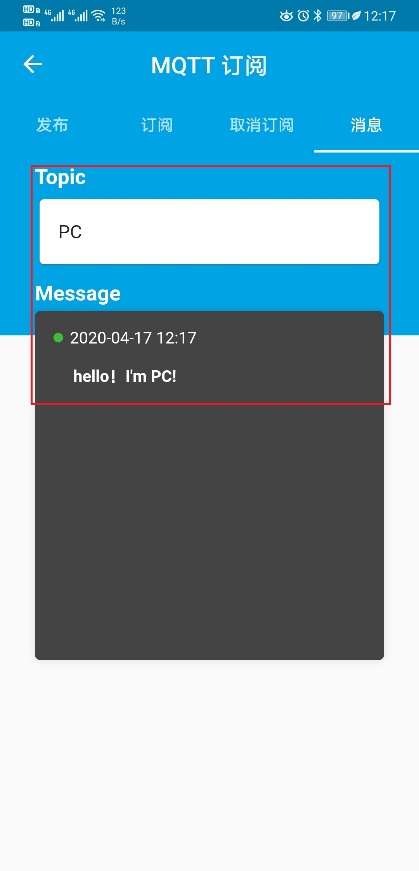
首先使用配套APP，在GL-S10上订阅“PC”主题，如下图所示：



MQTTBox往“PC”主题发送“hello！I'm PC!”

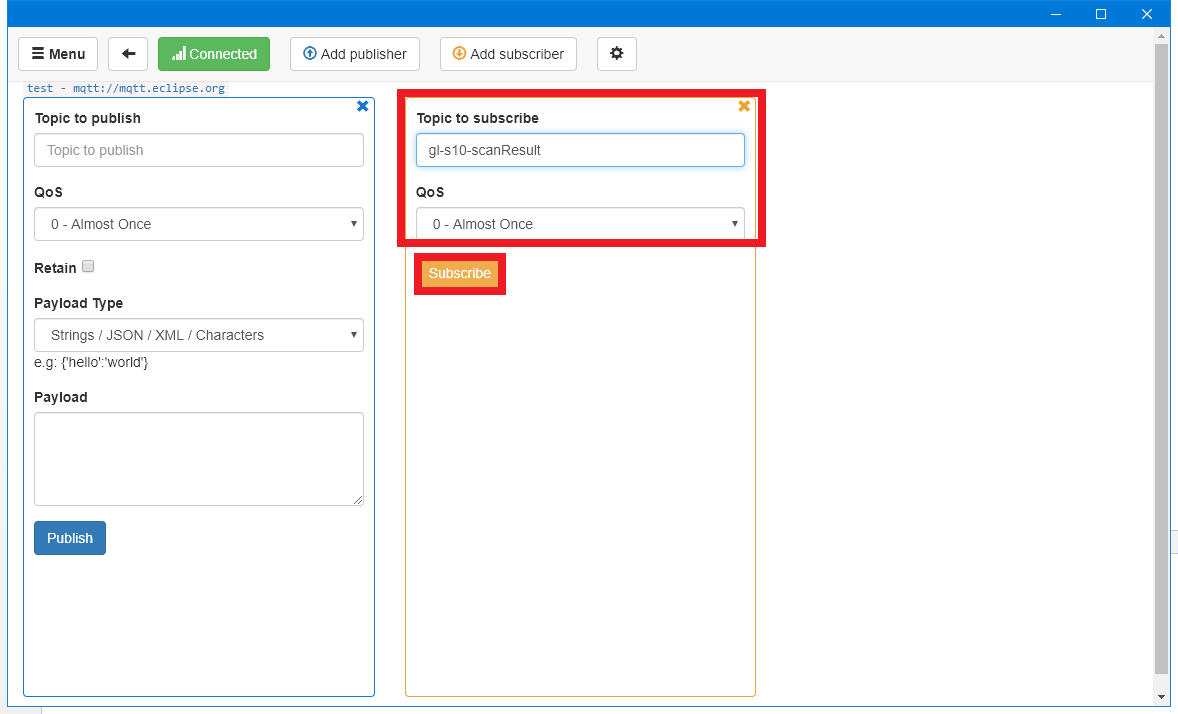


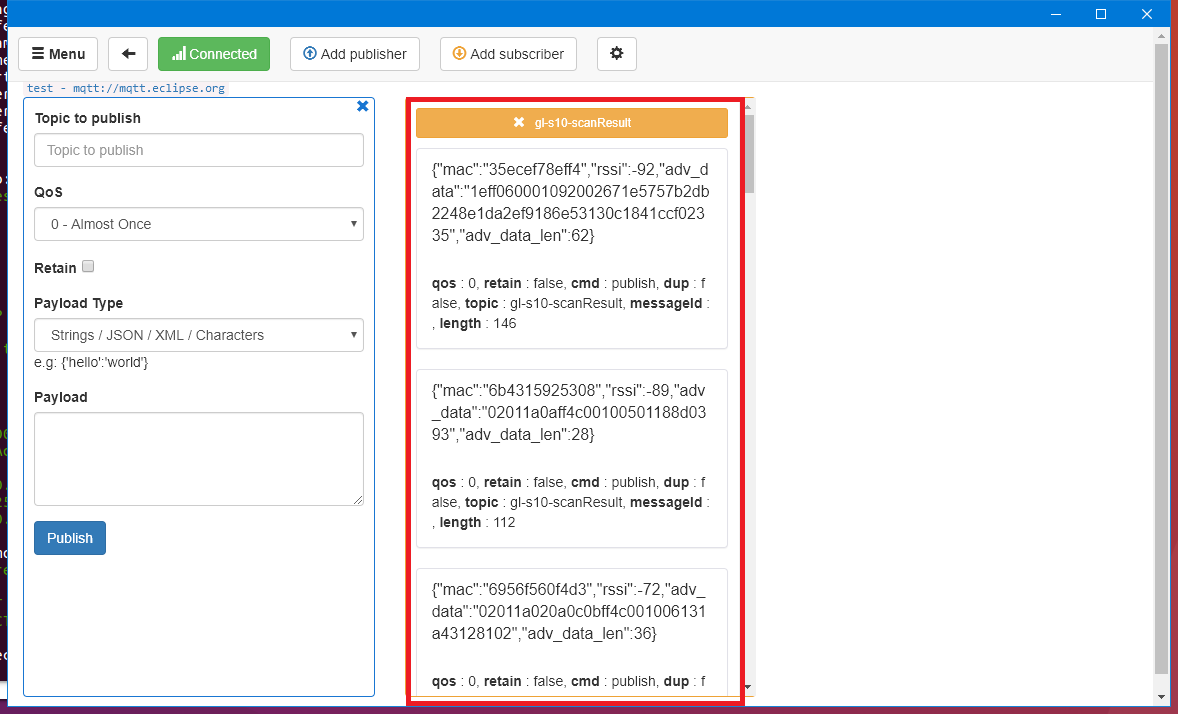
APP上接收到GL-S10传来的数据，如下图所示：



## BLE扫描并上传至云平台

本例程中，在用户联网后默认开启ble扫描（每间隔9秒扫描1秒），如果连接了MQTT服务器，扫描的结果会以JSON的格式打包上传到"gl-s10-scanResult"主题。用户可以使用另外的MQTT Client（如MQTTBox）订阅该主题，获取数据。





## 附录1 -- GL-S10 BLE API

GL-S10包含两个GATT服务。 第一个是为了接受指令和消息，服务的UUID是000000FF-0000-1000-8000-00805F9B34FB，它包含一个特征值，特征值的UUID是0000FF01-0000-1000-8000-00805F9B34FB，该特征值的读写权限只有可写。 第二个是为了发送通知，服务的UUID是000000EE-0000-1000-8000-00805F9B34FB，他也只有一个特征值，特征值的UUID是0000EE01-0000-1000-8000-00805F9B34FB。该特征值的权限是可读可通知 (read | notify)。 使能了第二个服务的notify功能，才可以获取通知。

### 指令格式

指令必须通过第一个GATT服务来发送，将要发送的指令写到第一个GATT的特征值里(服务UUID为000000FF-0000-1000-8000-00805F9B34FB，特征值UUID为0000FF01-0000-1000-8000-00805F9B34FB)即实现指令的发送。 指令有特定的格式。每条指令有四个部分构成，开始符，指令码，指令内容，结束符。其中指令内容长度不定，可以为空。

|  |  |  |  |
| --- | --- | --- | --- |
| Start Code | Length | CmdID | Payload |
| 开始符（1字节） | 包长（2字节） | 指令码（1字节） | 数据段 |

开始符是固定的1个字节 **0xFE** ；

包长是该数据包的长度（包含开始符）占2字节；

指令码的详细内容见“[1.3.1 指令汇总表](#_1.3.1_指令汇总表)”占1字节；

数据段是指令码的参数，部分指令不需要参数，这个区域就为空，详细内容见下表“[1.3.2 带参数的指令内容格式](#_1.3.2_带参数的指令内容格式)”。

**注意，整条指令必须转换成十六进制的uint8数组才可以写入GATT特征值里**

### 1.2 回应格式

每条指令执行后都会返回一个执行结果，来表明其执行是否成功。这个执行结果是通过第二个GATT服务的通知功能传递出来的（服务 UUID 000000EE-0000-1000-8000-00805F9B34FB, 特征值 UUID ：0000EE01-0000-1000-8000-00805F9B34FB）。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Start Code | Length | CmdID | result | Payload |
| 开始符（1字节） | 包长（2字节） | 指令码（1字节） | 执行结果 | 数据段 |

GL-S10的回应格式和指令格式在大体上一致的，执行结果“00”代表执行成功；“01”代表执行失败 。

### 1.3 指令码

#### 1.3.1 指令汇总表

|  |  |  |  |
| --- | --- | --- | --- |
| 指令码 | 指令 | 需要参数？ | 描述 |
| 0x01 | [SET\_WIFI\_CONFIG](#_1.4.1_SET_WIFI_CONFIG) | Y | 设置wifi的SSID和密码 |
| 0x02 | [START\_WIFI](#_1.4.2_START_WIFI) | N | 连接wifi |
| 0x03 | [STOP\_WIFI](#_1.4.3_STOP_WIFI) | N | 断开wifi |
| 0x04 | [RESTART\_WIFI](#_1.4.4_RESTART_WIFI) | N | 重连 WiFi |
| 0x05 | [GET\_NETWORK\_STATE](#_1.4.5_GET_NETWORK_STATE) | N | 获取当前网络状态 |
| 0x06 | [CLEAR\_WIFI\_CONFIG](#_1.4.6_CLEAR_WIFI_CONFIG) | N | 清空wifi信息 |
| 0x65 | [SET\_MQTT\_URI](#_1.4.7_SET_MQTT_URI) | Y | 设置MQTT uri |
| 0x66 | [START\_MQTT](#_1.4.8_START_MQTT) | N | 连接MQTT |
| 0x67 | [STOP\_MQTT](#_1.4.9_STOP_MQTT) | N | 断开MQTT |
| 0x68 | [RESTART\_MQTT](#_1.4.10_RESTART_MQTT) | N | 重连MQTT |
| 0x69 | [GET\_MQTT\_CONFIG](#_1.4.11_GET_MQTT_CONFIG) | N | 获取当前MQTT uri |
| 0x6A | [CLEAR\_MQTT\_CONFIG](#_1.4.12_CLEAR_MQTT_CONFIG) | N | 清除MQTT配置信息 |
| 0x6B | [MQTT\_SUBSCRIBE](#_1.4.13_MQTT_SUBSCRIBE) | Y | 订阅MQTT主题 |
| 0x6C | [MQTT\_UNSUBSCRIBE](#_1.4.14_MQTT_UNSUBSCRIBE) | Y | 取消订阅MQTT主题 |
| 0x6D | [MQTT\_PUBLISH](#_1.4.15_MQTT_PUBLISH) | Y | 向MQTT服务器发布消息 |
| 0x6E | [MQTT\_SUBSCRIBE\_MES](#_1.5.1_订阅MQTT信息通知) | / | 订阅的MQTT主题有消息通知 |
| 0xF0 | [GET\_VERSION](#_1.4.16_GET_VERSION) | N | 获取当前固件版本号 |
| 0xF1 | [SET\_OTA\_URL](#_1.4.17_SET_OTA_URL) | Y | 设置OTA服务器的url |
| 0xF2 | [OTA\_STATUS](#_1.5.2_OTA状态通知) | / | OTA升级状态通知 |
| 0xFA | [RESTART](#_1.4.18_RESTART) | N | 重启设备 |

#### 1.3.2 带参数的指令内容格式

|  |  |  |
| --- | --- | --- |
| 指令码 | 操作 | 内容格式 |
| 0x01 | SET\_WIFI\_CONFIG | ssid@password |
| 0x65 | SET\_MQTT\_URI | username:password@host:port |
| 0x6B | MQTT\_SUBSCRIBE | qos@topic |
| 0x6C | MQTT\_UNSUBSCRIBE | topic |
| 0x6D | MQTT\_PUBLISH | qos@topic@data |
| 0XF1 | SET\_OTA\_URL | updateNow@url |

### 1.4 API详情

#### 1.4.1 SET\_WIFI\_CONFIG

SET\_WIFI\_CONFIG用于设置wifi配置，指令码0x01。

|  |  |  |
| --- | --- | --- |
| 指令码 | 指令 | 内容格式 |
| 0x01 | SET\_WIFI\_CONFIG | ssid@password |

举例: 要连接wifi的SSID：GL-S1300-9cc，密码：12345678

**fe 00 16 01 47 4c 2d 53 31 33 30 30 2d 39 63 63 40 31 32 33 34 35 36 37 38**

回应：

若连接成功：**fe 00 05 01 00**

若20s內未连接成功：**fe 00 05 01 01**

#### 1.4.2 START\_WIFI

START\_WIFI用于启动连接wifi，指令码0x02。

|  |  |  |
| --- | --- | --- |
| 指令码 | 指令 | 内容格式 |
| 0x02 | START\_WIFI | 无 |

举例:

**fe 00 04 02**

回应：

若启动成功：**fe 00 05 02 00**

若启动失败：**fe 00 05 02 01**

#### 1.4.3 STOP\_WIFI

STOP\_WIFI用于关闭wifi连接，指令码0x03。

|  |  |  |
| --- | --- | --- |
| 指令码 | 指令 | 内容格式 |
| 0x03 | STOP\_WIFI | 无 |

举例:

**fe 00 04 03**

回应：

若关闭成功：**fe 00 05 03 00**

若关闭失败：**fe 00 05 03 01**

#### 1.4.4 RESTART\_WIFI

RESTART\_WIFI用于重启wifi连接，指令码0x04。

|  |  |  |
| --- | --- | --- |
| 指令码 | 指令 | 内容格式 |
| 0x04 | RESTART\_WIFI | 无 |

举例:

**fe 00 04 00**

回应：

若重启成功：**fe 00 05 04 00**

若重启失败：**fe 00 05 04 01**

#### 1.4.5 GET\_NETWORK\_STATE

GET\_NETWORK\_STATE用于获取当前网络状态，指令码0x05。

|  |  |  |
| --- | --- | --- |
| 指令码 | 指令 | 内容格式 |
| 0x05 | GET\_NETWORK\_STATE | 无 |

举例:

**fe 00 04 05**

回应：

|  |  |
| --- | --- |
| data段格式 | 例子 |
| state@mode@IP\_address@ssid | 1@0@192.168.1.111@testwifi |

“state” 表示连接的状态（网络连接或者是MQTT服务器连接），1表示连接，0表示连接断开。“mode”是表示连接方式，0表示通过wifi连接，1表示通过网线连接。

**fe 00 1f 05 00 31 40 30 40 31 39 32 2e 31 36 38 2e 31 2e 31 31 31 40 74 65 73 74 77 69 66 69** ====> 获取到当前网络状态为：网络已连接，通过wifi，ip地址为192.168.1.111，连接到的wifi ssid为 testwifi。

#### 1.4.6 CLEAR\_WIFI\_CONFIG

CLEAR\_WIFI\_CONFIG用于清除当前wifi配置，指令码0x06。

|  |  |  |
| --- | --- | --- |
| 指令码 | 指令 | 内容格式 |
| 0x06 | CLEAR\_WIFI\_CONFIG | 无 |

举例:

**fe 00 04 06**

回应：

若成功：**fe 00 05 06 00**

若失败：**fe 00 05 06 01**

#### 1.4.7 SET\_MQTT\_URI

SET\_MQTT\_URI用于设置MQTT配置，指令码0x65。

|  |  |  |
| --- | --- | --- |
| 指令码 | 指令 | 内容格式 |
| 0x65 | SET\_MQTT\_URI | mqtt://username:password@host:port |

举例: 要连接的mqtt服务端host：iot.cloud.com；port：1883；用户名：admin；密码：password

**fe 00 2c 65 6d 71 74 74 3a 2f 2f 61 64 6d 69 6e 3a 70 61 73 73 77 6f 72 64 40 69 6f 74 2e 63 6c 6f 75 64 2e 63 6f 6d 3a 31 38 38 33**

回应：

若成功：**fe 00 05 65 00**

若失败：**fe 00 05 65 01**

#### 1.4.8 START\_MQTT

START\_MQTT用于启动MQTT连接，指令码0x66。

|  |  |  |
| --- | --- | --- |
| 指令码 | 指令 | 内容格式 |
| 0x66 | START\_MQTT | 无 |

举例:

**fe 00 04 66**

回应：

若成功：**fe 00 05 66 00**

若失败：**fe 00 05 66 01**

#### 1.4.9 STOP\_MQTT

STOP\_MQTT用于中断MQTT连接，指令码0x67。

|  |  |  |
| --- | --- | --- |
| 指令码 | 指令 | 内容格式 |
| 0x67 | STOP\_MQTT | 无 |

举例:

**fe 00 04 67**

回应：

若成功：**fe 00 05 67 00**

若失败：**fe 00 05 67 01**

#### 1.4.10 RESTART\_MQTT

RESTART\_MQTT用于重新连接MQTT服务器，指令码0x68。

|  |  |  |
| --- | --- | --- |
| 指令码 | 指令 | 内容格式 |
| 0x68 | RESTART\_MQTT | 无 |

举例:

**fe 00 04 68**

回应：

若成功：**fe 00 05 68 00**

若失败：**fe 00 05 68 01**

#### 1.4.11 GET\_MQTT\_CONFIG

GET\_MQTT\_CONFIG用于获取当前MQTT配置信息，指令码0x69。

|  |  |  |
| --- | --- | --- |
| 指令码 | 指令 | 内容格式 |
| 0x69 | GET\_MQTT\_CONFIG | 无 |

举例:

**fe 00 04 69**

回应：

|  |  |
| --- | --- |
| 数据段格式 | 例子 |
| state@user:password@host:port | 1@admin:password@192.168.8.1:1107 |

“state” 表示MQTT服务器连接的状态，1表示连接，0表示连接断开。“user”是用户名（没有可即为空），“password”是密码（没有即为空），“host”是MQTT服务器的域名或IP，“port”是MQTT服务器的端口。

**fe 00 26 69 00 31 40 61 64 6d 69 6e 3a 70 61 73 73 77 6f 72 64 40 31 39 32 2e 31 36 38 2e 38 2e 31 3a 31 31 30 37** ====> 当前MQTT配置为：已连接到服务器，用户名是“admin”，密码是“password”，服务器IP是“192.168.8.1”，端口是“1107”。

#### 1.4.12 CLEAR\_MQTT\_CONFIG

CLEAR\_MQTT\_CONFIG用于清除当前MQTT配置，指令码0x6a。

|  |  |  |
| --- | --- | --- |
| 指令码 | 指令 | 内容格式 |
| 0x6a | CLEAR\_MQTT\_CONFIG | 无 |

举例:

**fe 00 04 6a**

回应：

若成功：**fe 00 05 6a 00**

若失败：**fe 00 05 6a 01**

#### 1.4.13 MQTT\_SUBSCRIBE

MQTT\_SUBSCRIBE用于MQTT订阅主题消息，指令码0x6b。

|  |  |  |
| --- | --- | --- |
| 指令码 | 指令 | 内容格式 |
| 0x6b | MQTT\_SUBSCRIBE | qos@topic |

举例: 订阅一个“test”主题，qos为0

**fe 00 0a 6b 30 40 74 65 73 74**

回应：

若成功：**fe 00 05 6b 00**

若失败：**fe 00 05 6a 01**

#### 1.4.14 MQTT\_UNSUBSCRIBE

MQTT\_UNSUBSCRIBE用于MQTT取消订阅主题信息，指令码0x6c。

|  |  |  |
| --- | --- | --- |
| 指令码 | 指令 | 内容格式 |
| 0x6c | MQTT\_UNSUBSCRIBE | topic |

举例: 取消订阅一个“test”主题

**fe 00 08 6c 74 65 73 74**

回应：

若成功：**fe 00 05 6c 00**

若失败：**fe 00 05 6c 01**

#### 1.4.15 MQTT\_PUBLISH

MQTT\_PUBLISH用于MQTT发布主题消息，指令码0x6d。

|  |  |  |
| --- | --- | --- |
| 指令码 | 指令 | 内容格式 |
| 0x6d | MQTT\_PUBLISH | qos@topic@data |

举例: 往“test”主题发送一个“hello”的消息，qos为0

**fe 00 12 6c 30 40 74 65 73 74 40 74 6f 70 69 63**

回应：

若成功：**fe 00 05 6d 00**

若失败：**fe 00 05 6d 01**

#### 1.4.16 GET\_VERSION

GET\_VERSION用于获取当前固件的版本号。

|  |  |  |
| --- | --- | --- |
| 指令码 | 指令 | 内容格式 |
| 0XF0 | GET\_VERSION | 无 |

举例：

**fe 00 04 f0**

回应：

|  |  |
| --- | --- |
| 数据段格式 | 例子 |
| version | 1.00 |

“version”即是当前固件的版本号。**fe 00 00 f0 00 31 2e 30 30** ====> 当前固件版本号为1.00

#### 1.4.17 SET\_OTA\_URL

SET\_OTA\_URL用于设置OTA升级的服务器链接，并可设置是否立即OTA升级。

|  |  |  |
| --- | --- | --- |
| 指令码 | 指令 | 内容格式 |
| 0XF1 | SET\_OTA\_URL | updateNow@url |

“updateNow”即为是否立即OTA升级，占1字节，0x00为立即升级，0x01为不立即升级。

举例：设置OTA升级的服务器链接是：http://cloud-dev.gl-inet.cn/gl-s10\_app.bin，且立即OTA升级。

**fe 00 2e f1 68 74 74 70 3a 2f 2f 63 6c 6f 75 64 2d 64 65 76 2e 67 6c 2d 69 6e 65 74 2e 63 6e 2f 67 6c 2d 73 31 30 5f 61 70 70 2e 62 69 6e**

回应：

若成功：**fe 00 05 f1 00**

若失败：**fe 00 05 f1 01**

#### 1.4.18 RESTART

RESTART用于立即重启设备，该指令没有返回，一般情况下不推荐使用。

|  |  |  |
| --- | --- | --- |
| 指令码 | 指令 | 内容格式 |
| 0xF2 | RESTART | 无 |

举例：

**fe 00 04 f2**

### 1.5 主动信息通知

除了返回指令执行结果之外，还有在特定场景下触发的ble信息，也是通过ble notify的形式下发的。

#### 1.5.1 订阅MQTT信息通知

当MQTT订阅的主题有数据通知时，会通过ble传出。

|  |  |  |  |
| --- | --- | --- | --- |
| Start Code | Length | CmdID | Payload |
| 开始符（1字节） | 包长（2字节） | 指令码（0x6E） | 数据段（topic@data） |

例如, 收到通知 **fe 00 15 6e 74 65 73 74 40 48 65 6c 6c 6f 20 77 6f 72 6c 64 21** 的意思是收到MQTT服务器推送的一条来自于订阅主题“test”的消息“Hello world!”

#### 1.5.2 OTA状态通知

当OTA升级通过http拉取固件并烧录结束或者该过程中出现失败，都会有OTA状态通知通过ble传出。

|  |  |  |  |
| --- | --- | --- | --- |
| Start Code | Length | CmdID | Payload |
| 开始符（1字节） | 包长（2字节） | 指令码（0xF2） | 数据段（1字节） |

Payload段0x00表示成功，0x01表示失败

例如：

OTA升级成功，即将马上重启：**fe 00 05 f2 00**

OTA失败：**fe 00 05 f2 01**