

本文档中介绍的函数只支持在 **Arduino** 开发环境下使用，因为需要调用官方的驱动库。  
该开发环境的搭建请参看文档《[1\\_搭建开发环境](#)》。

该文档基于程序运行流程说明主要函数的实现，目的是帮助用户理清思路，鼓励用户对程序进行完善，并开源给更多使用者。

单设备程序和网关设备程序有很大的相似性，本文档以单设备为例进行介绍，用户可举一反三理解网关设备的程序。

**注意：在云端 esp8266 暂不支持网关形式配置，只支持单设备配置。**

限于本人技术能力有限，程序难免有漏洞，忘用户批评指正。

## 第一部分：`qiaqiaSingleDevESP8266.h` 、 `qiaqiaSingleDevESP8266.cpp` 中的函数

### 1. `void sdInit()`

该函数在 `void setup{}`中调用，配置必要的初始化参数，用户可根据情况添加自己的配置代码。`sdInit()`中一个重要操作是从设备 `flash` 中读取配置文件，该文件中的数据是启动之后实现 `wifi` 连接和 `MQTT` 连接的必要参数。

### 2. `bool sdRun()`

该函数在 `void loop{}`中调用，其功能为轮询检测设备本地 `wifi` 连接状态以及与远程服务器的连接状态。设备启动后会使用从配置文件中读取的信息自动连接，如果超时，就认为参数设置有误，需用户手动重新配置。此时，设备会进入 `Access Point` 模式（调用 `sdAPModeConfig()`）。配置同样有超时设置，如果超时，设备重启，仍使用先前的连接信息再次尝试连接，如此循环。

#### 2.1 `sdAckuseridSub()`

如果设备连接本地及云端正常，就调用 `sdAckuseridSub()`向云端服务器订阅 `ackuserid`。云端无应答信息。

#### 2.2 `sdAckuseridRegist()`

该函数紧跟着 `sdAckuseridSub()`被调用，作用是向云端推送本设备基本信息。云端会有应答信息，其内容中有云端配置好的子设备信息。应答信息是 `Json` 格式编码的，用户应了解 `Json`，便于理解程序。调试设备时，可通过串口助手查看应答信息。

#### 2.3 `pClient->loop();`

该函数是调用第三方 `MQTT` 驱动库函数，它会发送、接收组包好的 `MQTT` 协议信息（`topic+payload`）。在本函数中有解包收到数据的回调函数 `callback()`，在 `sdInit()`中已完成注

册 `sdMQTTRecv()`。

#### 2.4 `sdMQTTDataParse()`

解包收到的数据包。根据数据包内容来决定程序运行哪种功能，会调用不同的解析函数，触发不同的操作，请用户阅读程序理解。

该函数的内容通常是响应云端的各种操作，例如重启设备、清除 **wifi** 等。其中，有些功能是由回调函数实现的，在这些回调函数中可放置用户应用程序。

#### 3. `int online()` (网关设备使用)

该函数用于子设备向云端发送在线信息。通常由用户在应用程序中周期性调用，向云端表示设备还在线。

#### 4. `int sdValueChange()`

该函数用于子设备向云端推送数据点信息。

#### 5. `int sdAck()`

该函数通常与读回调函数配合使用。云端读子设备某个数据点信息，子设备需应答，就通过这个函数上传数据点信息到云端。

#### 6. `void sdAPModeConfig()`

该函数在设备连接不上 **wifi** 和云端服务器时自动调用。此时设备进入 **AP** 模式，用户可从终端设备（PC 或手机）的浏览器访问（默认 IP:192.168.4.1）进入配置页面。配置完成后点击 **OK** 退出，设备会自动重启并使用最新的连接信息尝试建立连接。

#### 7. `xxxxDecode()`

此类函数有多个，在 `sdMQTTDataParse()` 函数中被调用，针对不同类型的数据包解析出不同数据并存储。

### 第二部分：MsgHandler.h 、MsgHandler.cpp 中的函数

#### `xxxxEncode()`

此类函数是对要发送的数据进行组包，主要是 MQTT 协议中的 **topic** 和 **payload** 部分。

`topic` 和 `payload` 中的内容是由恰恰云自定义的，具体介绍请参看文档《2\_QQC\_MQTT Protocol V1.0》。云端服务器也遵循此定义，所以不建议用户修改。

### 第三部分：PubSubClient.h、PubSubClient.cpp

第三方库文件，该函数库主要包括 MQTT 协议的驱动代码，包括与服务器建立连接、订阅信息、推送信息等操作。