OneShotConfig2.0(IOS)

SDK用户手册

V1.0.0

北京联盛德微电子有限责任公司 (Winner Micro)

地址：北京市海淀区阜成路67号银都大厦18层

电话：+86-10-62161900

公司网址：[www.winnermicro.com](http://www.winnermicro.com)

文档修改记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 修订时间 | 修订记录 | 作者 | 审核 |
| V1.0.0 | 2018.09.28 | 初稿 | 王海芳 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**目录**

[1 适用范围 4](#_Toc525918436)

[2 OneShotConfig.h接口定义 4](#_Toc525918437)

[2.1 startConfig函数 4](#_Toc525918438)

[2.2 stopConfig函数 4](#_Toc525918439)

[2.3 start方法 5](#_Toc525918440)

[2.4 stop方法 5](#_Toc525918441)

[3 OneShotConfig.h中函数使用方法 6](#_Toc525918442)

[3.1 获取OneShotConfig对象实例 6](#_Toc525918443)

[3.2 启动后台线程，循环调用startConfig或调用start函数 6](#_Toc525918444)

# 适用范围

本文档设计描述了W600芯片一键配置（IOS）SDK的接口定义、使用方法等，为IPhone开发APP配置Wi-Fi设备入网提供参考。

# OneShotConfig.h接口定义

# startConfig函数

本方法开始一键配置，将用户设备连接的Wi-Fi名称和密码进行编码加密后，通过UDP组播报文发送出来。

原型：

**-(int)startConfig: (NSString\*) ssidpwd: (NSString\*) password;**

参数：

ssid：当前连接WIFI网络名称

password：用户输入的密码

返回值：

0：表示发送正常结束，如果没有配置成功，需要继续调用该方法

-1：表示由于调用stop或是stopConfig接口，中断该方法

-2：表示内部错误功能：

注意：

本方法调用了UDP socket发包接口，因此本方法需要在后台线程中调用；本方法首先发送信道同步包再发送将Wi-Fi名称和密码编码的数据包，通常在5~10秒后返回，为保证Wi-Fi设备端收全信息并正确解析，本方法需要被循环调用，直到配网成功。如果在此方法抛出OneShotException异常，表明配网失败，请通过调用stopConfig函数结束配网并释放资源。

# stopConfig函数

本方法停止一键配置，释放配置过程中的资源。

原型：

**-(void)stopConfig;**

参数：

无

返回值：

无

注意：

本方法停止一键配置并释放资源，为保证资源释放，无论配网是否成功，在结束时到必须调用此方法。

# start方法

本方法开始一键配置，将用户设备连接的Wi-Fi名称和密码进行编码加密后，通过UDP组播报文发送出来。

原型：

**-(void) start: (NSString\*)ssid key:(NSString\*)key timeout:(int) timeout;**

参数：

ssid：Wi-Fi网络名称；

key：用户设备所连接的Wi-Fi网络的密码；

timeout：超时时间，单位秒；

返回值：

无

注意：

本方法调用了UDP socket发包接口，因此本方法需要在后台线程中调用。本方法会在设定的timeout超时时间后自动返回，如果没有到超时时间即返回，表明配网过程中有异常情况终止了配网。如果想停止配网，可以通过调用stop方法终止本方法，使其直接返回。

# stop方法

本方法停止一键配置，释放配置过程中的资源。

原型：

**-(void) stop;**

参数：

无

返回值：

无

注意：

本方法停止一键配置并释放资源，为保证资源释放，无论配网是否成功，在结束时到必须调用此方法。

# OneShotConfig.h中函数使用方法

# 获取OneShotConfig对象实例

首先通过下面代码获取OneShotConfig对象实例。

OneShotConfig \* communication = [[OneShotConfig alloc] init];

# 启动后台线程，循环调用startConfig或调用start函数

然后可以参考OneShotConfig Demo App创建后台线程。

NSThread \* thread2=[[NSThread alloc] initWithTarget:**self** selector:**@selector**(sendData:) object:ti];

[thread2 start];

循环调用startConfig或调用start函数，如下代码可以参考。

-(**void**)sendData:(TaskInfo \*) ti

{

#if !USE\_TIMEOUT\_INTERFACE

**@autoreleasepool** {

**while** (1)

{

**if** ([NSThread currentThread].isCancelled)

{

[**self** performSelectorOnMainThread:**@selector**(postStop) withObject:**self** waitUntilDone:**NO**];

[communication stopConfig];

[NSThread exit];

}

**else**

{

*//配网中*

**int** status=[communication startConfig:ti.ssid pwd:ti.password];

NSLog(@"startConfig ret %d", status);

**if** ( status == -1)

{

[[NSThread currentThread] cancel];

}

}

}

}

#else

[communication start:ti.ssid key:ti.password timeout:60];

[**self** performSelectorOnMainThread:**@selector**(postStop) withObject:**self** waitUntilDone:**NO**];

[communication stop];

#endif

}