WM_W60X_OneShotConfig2.0(IOS)SDK 用户手册

V1.1

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

文档修改记录

版本	修订时间	修订记录	作者	审核
V1.0	2018.09.28	初稿	Wanghf	
V1.1	2018.12.13	配网协议支持 W600/W601，文件更名为 W60X	Cuiyc	

目录

1 适用范围

本文档设计描述了 W60X 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。

2 OneShotConfig.h 接口定义

2.1 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。

2.2 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。

2.3 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。

2.4 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。

3 OneShotConfig.h 中函数使用方法

3.1 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];
```

3.2 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
}
```