

▼ Embedded Binaries

Add embedded binaries here

+ -

▼ Linked Frameworks and Libraries

Name	Status
 libSmartLinkStaticLib.a	Required ↕
+ -	

必须用IOS设备来运行(Must use the read IOS device to run the APP)

示例代码 Demo Code

在自己的工程文件内调倒入 Import the the following header file in project

```
#import <SmartLinkLib/SmartlinkLib.h>
```

```
-(void)viewDidLoad{
    smtlk = HFSmartLink sharedInstance];//获取单例 Get instance
    smtlk.isConfigOneDevice = true;//设置为多设备配置还是单个配置 Set the single or multiple device config
    /*your code*/
}

/*
    设置开始的按键 Set the start button
    isconnecting 表示是否正在进行配置
*/
- (IBAction)connectPress:(id)sender {
    NSString * pswdStr = self.pswd.text;
    self.progress.progress = 0.0;
    if(!isconnecting){
        [smtlk startWithKey:pswdStr processblock:^(NSInteger process) {
            self.progress.progress = process/18.0;
        } successBlock:^(HFSmartLinkDeviceInfo *dev) {
            [self showAlertWithMsg:[NSString stringWithFormat:@"%s:%s",dev.mac,dev.ip] title:@"OK"];
        } failBlock:^(NSString *failmsg) {
            [self showAlertWithMsg:failmsg title:@"error"];
        } endBlock:^(NSDictionary *deviceDic) {
            isconnecting = false;
            [self.connectBtn setTitle:@"connect" forState:UIControlStateNormal];
        }];
        isconnecting = true;
        [self.connectBtn setTitle:@"connecting" forState:UIControlStateNormal];
    }else{
        [smtlk stopWithBlock:^(NSString *stopMsg, BOOL isOk) {
            if(isOk){
                isconnecting = false;
                [self.connectBtn setTitle:@"connect" forState:UIControlStateNormal];
            }
        }];
    }
}
```

```

        [self showAlertWithMsg:stopMsg title:@"OK"];
    }else{
        [self showAlertWithMsg:stopMsg title:@"error"];
    }
}];
}
}
}

```

接口注释 Interface comment

```

/**
 * 设置进度Block          Set process block
 *
 * @param process 0~18 的整数    process 0~18 value
 */
typedef void(^SmartLinkProcessBlock)(NSInteger process);
/**
 * 设置成功以后的Block          Set smartlink success block
 *
 * @param dev
 */
typedef void(^SmartLinkSuccessBlock)(HFSmartLinkDeviceInfo *dev);
/**
 * 设置失败的信息          Set smartlink fail information
 *
 * @param failmsg 失败信息
 */
typedef void(^SmartLinkFailBlock)(NSString * failmsg);
/**
 * 用户手动停掉的block          Set smartlink stop block
 *
 * @param stopMsg 停止的信息    Stop information
 * @param isOk    是否停止成功    Is stop ok
 */
typedef void(^SmartLinkStopBlock)(NSString *stopMsg,B00L isOk);
/**
 * 关闭服务的Block          Close smartlink service block
 *
 * @param closeMsg 关闭的信息
 * @param isOK    是否关闭成功
 */
typedef void(^SmartLinkCloseBlock)(NSString * closeMsg,B00L isOK);
/**
 * 发现设备的block          Find smartlink config OK device block
 *
 * @param deviceDic 发现的设备    Deice information
 */
typedef void(^SmartLinkEndblock)(NSDictionary * deviceDic);

@interface HFSmartLink : NSObject
/**
 * 是否配置单个设备，或者多个设备 默认false    Config single or multiple device, false by default
 */
@property (nonatomic) B00L isConfigOneDevice;
/**
 * 配置信息发送完成以后，等待搜索设备的时间 second 默认15
 */ Wait times for sending the smartlink UDP broadcase data. Default is 15 seconds.
@property (nonatomic) NSInteger waitTimers;
/**

```

```

* 获取smartlink 的单例      Get smartlink instance
*
*  @return 返回smartlink的单例
*/
+(instancetype)shareInstence;
/**
*  开始配置 block不能为nil      Start config. block should be be nil
*
*  @param key      路由器密码      router password
*  @param pblock 进度block      process block
*  @param sblock 成功block      success block
*  @param fblock 失败block      fail block
*  @param eblock 结束block      end block
*/
-(void)startWithKey:(NSString*)key processblock:(SmartLinkProcessBlock)pblock successBlock:
(SmartLinkSuccessBlock)sblock failBlock:(SmartLinkFailBlock)fblock endBlock:(SmartLinkEndblock)eblock;
/**
*  停止配置      Stop config
*
*  @param block 停止配置的block
*/
-(void)stopWithBlock:(SmartLinkStopBlock)block;
/**
*  关闭整个Smartlink服务，再次调用的时候必须 从头开始 初始化。
*  Stop the smartlink total service, must initialize from the head before start smartlink again..
*  @param block 关闭服务block
*/
-(void)closeWithBlock:(SmartLinkCloseBlock)block;
@end

```