

TJDBluetoothSDK开发手册

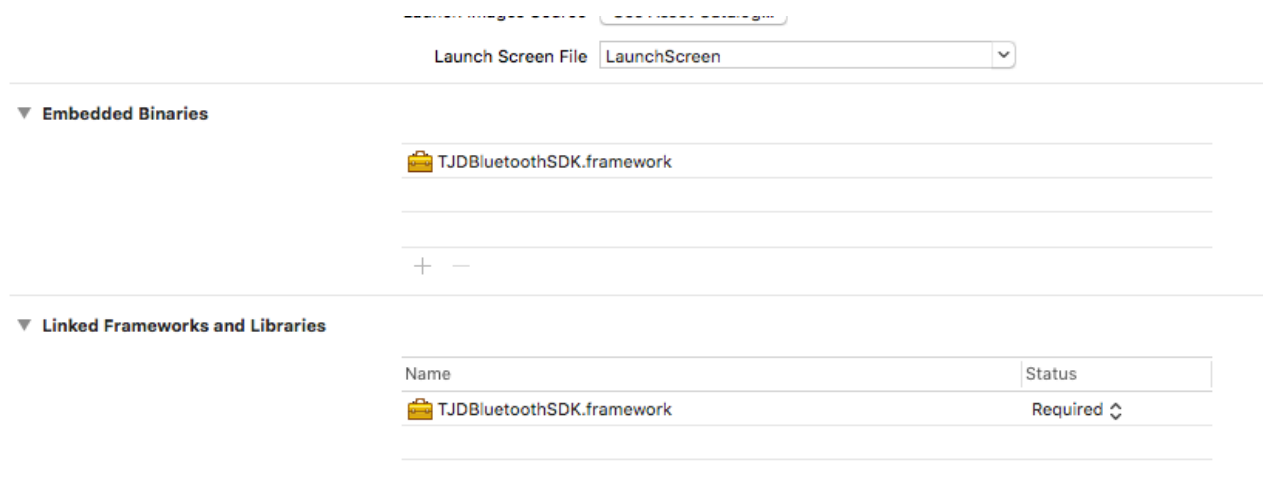
备注（Note）：只支持真机运行（Only support real machine operation）

1.swift开发，直接拖入项目，oc项目，请先创建任意一个swift文件，根据xcode提示创建一个oc桥接文件，桥接文件生成后可干掉swift文件。

1.swift development, drag directly into the project, oc project, please create any swift file first, create an oc bridge file according to the xcode prompt, and save the swift file after the bridge file is generated.

2.找到 General选项，往下拉，找到Embedded Binaries，将SDK文件拖入即可。

2. Find the General option, pull down, find the Embedded Binaries, and drag the SDK file into it.



3.若导入了预编译文件中，请到 Build Settings中找到 Always Embed Swift Standard Libraries，并设置为YES

3. If you imported the precompiled file, go to Build Settings and find Always Embed Swift Standard Libraries and set it to YES.

1.Import header file. `#import <TJDBluetoothSDK/TJDBluetoothSDK.h>` Macro defines management instance for easy operation 宏定义管理实例，方便操作

```
#define bleSelf [WUBleManager shared]
```

2. 创建管理实例，会回调手机蓝牙状态

2. Create a management instance, which will call back the Bluetooth status of the phone.

```
[bleSelf setUpManager];
```

3. 所有回调都通过通知发出。

WUBleManagerNotifyKeys 该类是基本蓝牙通知key

3. All callbacks are sent by notice.

WUBleManagerNotifyKeys This class is the basic Bluetooth notification key

WristbandNotifyKeys 该类是蓝牙数据处理返回通知类，基本上都在子线程中发出，除了“readyToWrite”

WristbandNotifyKeys This class is a Bluetooth data processing return notification class, which is basically issued in a child thread, except for "readyToWrite"

4. WUBleModel 该类存储蓝牙实例信息，具体看属性注释。

4. WUBleModel This class stores Bluetooth instance information, see the property comments.

5. 其他的再联系我。

5. Contact me again.

