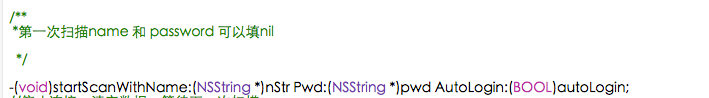
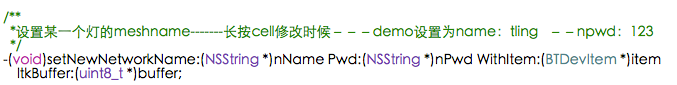
1-----加灯第一步－－－本份代码里面的AddDeviceViewController中的viewDidload方法里面调用扫描方法

在源码中有一个外接的代理方法



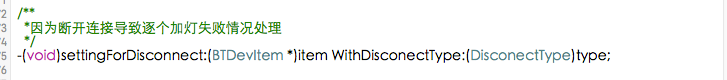
此处接收每次符合条件的BTDevItem对象，加灯过程可在此处进行－》调用

实现对单灯的设置

在每次的单灯设置完成后会调用下面的代理方法Macintosh HD:Users:telink:Desktop:屏幕快照 2015-12-30 下午7.25.01.png

在status 为上一个灯设置完成以后可以实现后一个灯的扫描设置

处理因为异常断开连接的代理方法



具体的type类型参见枚举部分－可在此处针对某一个特定设备item设置meshname后者在此处补充性的扫描，完成后面的加灯操作。

2------更新直连灯和其他灯被断电时候的重连问题

3------添加获取所有灯的状态接口，直接BTCentalmanager获取

-(void)getAllLightStatus;

4-----解决某一个灯被其他设备直连而自己的iphone无法扫描并且显示到UI界面上的问题，目前处理后UI上面显示的为整个mesh内所有的灯，包括被其他iphone设备直连

5-----解决整个app的数据源问题，前期两个数据源－1:扫描到的设备 2-notify所得到的状态补充以及对被其他iphone直连的设备补充到数据源

目前：整个app的数据源只有一个DeviceModel－具体参见代码

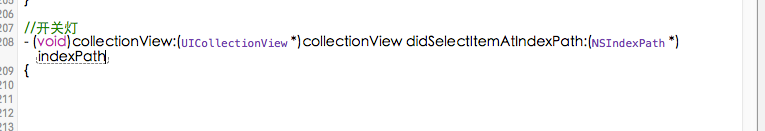
6-----处理直连灯被拔掉之后到重新连接到另外一盏灯期间的状态显示问题，处理后的效果为在此期间所有灯的状态为offline——－详细见resetStatusOfAllLight方法在BTCentraManager.m文件中的调用

7-----更改getNotify数据的次数，处理前默认为连接某一个灯以后get notify两次 处理后的效果为登录上以后每隔500ms getnotify一次， get 4 次 详细见

BTCentraManager.m文件中setNotifyOpenProSevervalTimes方法的调用

8-----修改因为数据包丢失造成灯的状态错误无法控制的问题，在每次点击灯的图标后，现在的处理方式是直接直接将灯的状态切换后发送开关灯命令，经后期的notify数据包回馈解析后校核状态问题

见：MainViewController.m文件中的

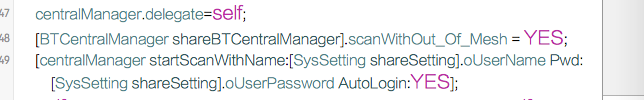


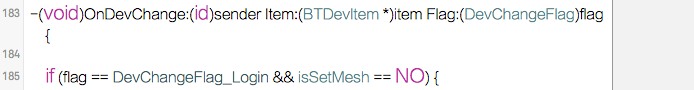
9-- 修改 了mesh内部只有一个灯的时候，灯被断开，然后上电之后的状态显示问题,设置地址之后请将所有灯一同断电一次

10-- 在单灯设置的页面添加kict\_out\_of\_mesh操作，

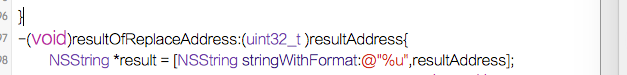
11-- 添加分配地址与加灯操作同时进行并且在加灯的过程中将out\_of\_mesh类的设备也添加到同一个mesh的操作，本版本改动主要在源码内部的扫描过滤条件的设置：scanWithOut\_of\_Mesh部分，和在connect和login的时候对out\_of\_mesh的特殊处理

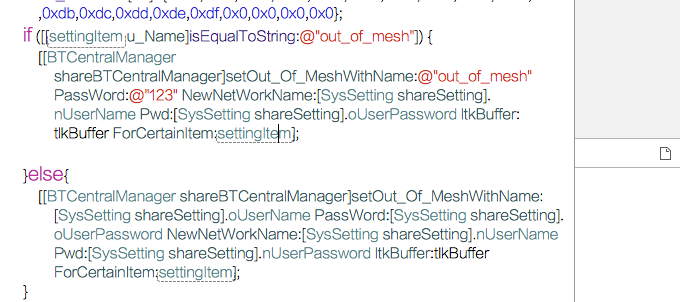
主要调用接口如下

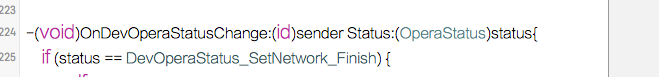
1-——

该接口开始扫描并且自动连接登录后在

中截取到已经登录成功的节点调用，其中的DevChangeFlag\_Login枚举为新加部分Macintosh HD:Users:telink:Desktop:屏幕快照 2016-01-10 上午3.46.10.png

方法对地址进行设置，分配成功与否的结果在回调方法中通过resultAddress和所设定的值是否相同判断

判断设置成功之后分类别设置out\_of\_mesh和telink\_mesh1这一类的灯的meshname，password和ltk，下面接口为新添接口

加灯设置meshname，pwd ltk如果已经设置完毕在如下回调中得到

login和修改地址，设置mehs三个部分的成功与否，本demo简单使用参数和定时器共同处理，另外加灯操作的整过过程最好先设置地址，后设置meshname，pwd等

12-- 更正因为源码内部修改mesh过程中不断主动获取notify数据的问题

13-- 修改在加灯过程中修改mesh过程中修改out\_of\_mesh的password失败的bug

14-- ota过程——只针对此demo

Macintosh HD:Users:telink:Desktop:屏幕快照 2016-01-22 下午3.34.49.png

此方法直连上某一个设备之后

Macintosh HD:Users:telink:Desktop:屏幕快照 2016-01-22 下午3.37.24.png读取当前直连灯的firmWare属性，读取后的结果以NSdata在下面

Macintosh HD:Users:telink:Desktop:屏幕快照 2016-01-22 下午3.39.17.png

中返回

在上面OnConnectionDevFirmWare中启动发送OTA数据包，调用

Macintosh HD:Users:telink:Desktop:屏幕快照 2016-01-22 下午3.41.20.png

此方法每调用一次只能发送一个20bytes的包，具体整个文件的发送参考OtaViewController.m文件

15-- ota过程——改进

用

Macintosh HD:Users:telink:Desktop:屏幕快照 2016-01-25 上午9.35.13.png

Macintosh HD:Users:telink:Desktop:屏幕快照 2016-01-25 上午9.35.46.png

和属性isAutoLogin处理过程中的指定某一个设备的连接登陆，并设置默认isAutoLogin取消自动重连机制，设置定时指定某一个设备的连接，如果tiemout就恢复自动重连机制，

替换之前冗余的接口——传入整个数据包，源码内部处理Macintosh HD:Users:telink:Desktop:屏幕快照 2016-01-22 下午3.41.20.png

改成Macintosh HD:Users:telink:Desktop:屏幕快照 2016-01-25 上午9.40.15.png

具体的分包处理放在具体控制器中

读取firmware依然在Macintosh HD:Users:telink:Desktop:屏幕快照 2016-01-22 下午3.37.24.png和

Macintosh HD:Users:telink:Desktop:屏幕快照 2016-01-22 下午3.39.17.png中组合处理

具体过程

1-——连接登陆

2-——读取firmWare

3-——发送数据包

4-——断开连接之后重新启动指定设备的连接登陆

5-——读取firmware

16-- 更改UI显示问题，取消部分不必要的扫描连接,暴露接口方便随时获取灯的状态

17-- 整个Mesh的OTA过程---只针对能够扫描到的设备

a， 获取临时数组tempArr装放已经到的设备，，，每个单灯的OTA过程以连接该tempArr的第最后一个元素开始，如果中途无法连接，或者login失败，选择重新连接tempArr的第一个元素，并且设置定时，如果tempArr的第一个元素也无法连接登录，默认处理为OTA无法进行下去，

b， 每一个设备的OTA循环结束之后，将tempArr的最后一个设备从数组中remove，并且重复a过程

c. 具体的单灯OTA过程参考 文档的15.，16部分，

17-- 添加单灯OTA每发送1kb的数据包读取一次fiirmware,发包间隔改成20ms

18--

Macintosh HD:Users:telink:Desktop:屏幕快照 2016-04-11 下午5.06.30.png接口中的otaBuffer的赋值操作的改动，保证Release出的IPA文件正常OTA

19

-(void)sendPack:(NSData \*)data {}中做一个是否登录状态来判断写入buffer的type

其中API

-(void)writeValue:(CBCharacteristic \*)characteristic Buffer:(uint8\_t \*)buffer Len:(int)len

改成了，添加了一个参数

-(void)writeValue:(CBCharacteristic \*)characteristic Buffer:(uint8\_t \*)buffer Len:(int)len response:(CBCharacteristicWriteType)type

其中

-(void)setNewNetworkDataPro{}/

-(void)setNotifyOpenPro{} 和

-(void)loginWithPwd:(NSString \*)pStr{} 3个中写入buffer时均时需要CBCharacteristicWriteWithResponse

而在

-(void)sendCommand:(uint8\_t \*)cmd Len:(int)len{}中写入buffer时则传入参数为CBCharacteristicWriteWithoutResponse

20

改变方法实现，增加对连续指令做延时过滤处理

- (void)sendCommand:(uint8\_t \*)cmd Len:(int)len;

21

去掉ota压包（app写数据）的interval时间（改为0），read间隔128个字节；其中开关灯／rgb／亮度相关命令，设置取样（条件300ms），其他命令不采样，并且相邻命令执行时间为间隔大于等于300ms；

4月24日修改 修改人->石晴露

22

1.优化SDK里面的命令delay

2.增加testlog 信息，宏（kTestLog）控制是否出现

3.增加SDK各阶段超时回调API：

- (void)loginTimeout:(TimeoutType)type;

4.增加扫描到登录特征值时，可供登录回调

- (void)scanedLoginCharacteristic;

5月22日修改 修改人->石晴露

23.

//修复手机修改时间错误造成的命令延时执行错误的问题；



24

加灯完成后，立刻ota，ota失败的问题：

Class BTCentralManager属性 operaType由于只有自动登录(改成DevOperaType\_AutoLogin)和修改mesh(改成DevOperaType\_Set)时会去修改，而且登录时会根据此operaType判断是否是DevOperaType\_Set，如果是，则去 "配置网络"，因此 在加灯过程中会调整成 DevOperaType\_Set，当不经过自动登录的修正直接去做OTA时，ota发包过程也会发配置网络数据，造成ota中断；

在Class BTCentralManager .m文件中1130行中加入 "\_operaType = DevOperaType\_Normal" 修正该值

25.

12月29日修改 修改人->石晴露

1、demo app 添加二维码分享功能

2、SDK中不同工作阶段添加error code上报内容

- (void)stateCodeAndErrorCodeAnasisly:(TimeoutType)type

3、调整部分代码实现，如发包function的实现

-(void)sendPack:(NSData \*)data;

26.

2月6日修改 修改人->石晴露

1、-(void)sendPack:(NSData \*)data;

12月29日调整过该方法的实现，但是在最后一个长度不是16时，计算存在问题，现修正；如果没有替换成该方法，不需要调整，以前的没有问题

2、调整下面接口的实现，适配外界对不同productid做处理时的情况

-(void)setOut\_Of\_MeshWithName:(NSString \*)addName PassWord:(NSString \*)addPassWord NewNetWorkName:(NSString \*)nName Pwd:(NSString \*)nPwd ltkBuffer:(uint8\_t \*)buffer ForCertainItem:(BTDevItem \*)item；

-(void)connectWithItem:(BTDevItem \*)cItem；

-(void)setNewNetworkName:(NSString \*)nName Pwd:(NSString \*)nPwd WithItem:(BTDevItem \*)item ltkBuffer:(uint8\_t \*)buffer;

- (void)peripheral:(CBPeripheral \*)peripheral didWriteValueForCharacteristic:(CBCharacteristic \*)characteristic error:(NSError \*)error；

27.

3月30日修改 修改人->石晴露

增加ota模式选择，加灯模式选择。Mesh加灯实现，开发者可以参考代码文件“ MeshAddVC.m”。

28.

4月26日修改 修改人->梁家誌

OTA流程修改：在APP连接并login成功后，延时3S再发送OTA数据包。

修改原因：为了提高手机的兼容性，iOS与安卓统一这样修改。

29.

4月26日修改 修改人->梁家誌

完善meshOTA功能，并把meshOTA相关的逻辑代码抽离出来，形成新的meshOTA管理类。meshOTA逻辑修改：在完成单灯OTA后，不再发送meshOTAStart接口。MeshOTA的接口在代码文件“ MeshOTAManager.h”，接口的使用，开发者可以参考代码文件“ MeshOTAVC.m”。

30.

5月8日修改 修改人->梁家誌

为了APP在静默meshOTA时，用户可以控制设备，demo在调用发送OTA数据包的方法- (void)sendPack:(NSData \*)data前，判断前一秒内是否有发送非OTA数据，有则延时0.3秒，再重复做判断操作。

5月8日修改 修改人->梁家誌

添加分组的控制亮度、色温、颜色功能。原色温组包-(void)setCTOfLightWithDestinationAddress:(uint32\_t)destinationAddress AndCT:(float)CT异常，色温值应该存在第12字节。

31.

5月9号修改 修改人->梁家誌

在meshAdd完成之后，统一获取设备类型、固件版本号。即在“MeshAddVC.m”line195添加[[BTCentralManager shareBTCentralManager] readFirmwareVersion];

32.

5月14日修改 修改人->梁家誌

调节亮度最低值为0，修改为1。因为在灯开的状态下，亮度范围为：1-100。

33.

5月16日修改 修改人->梁家誌

33.1.异常情况：加灯过程中，会进行修改设备短地址的操作。设备短地址修改后，蓝牙设备的广播包数据data中key“kCBAdvDataManufacturerData”对应的值会发生改变。由于iOS蓝牙框架的scan缓存的问题，在个别系统的个别类型的设备上会出现一次扫描返回同一个设备的两个广播包的情况，即：扫描蓝牙设备的代理回调中，第一次回调中的广播包数据为旧的，紧接着第二次回调中的广播包数据才是新的。

33.2修改逻辑：除了连接指定的蓝牙设备或者只扫描不添加这两种情况，其它情况下，扫描到蓝牙设备先存储到数组“discoverPeripheralArrs”中。300ms没有搜索到新的蓝牙设备或者数组存储的蓝牙设备数达到了10个，开始for循环处理数组里面保存的蓝牙数据(原来的处理逻辑，无需修改)。

33.3 参考代码：“BTCentralManager.m”line683

34.

9月6日优化33容错逻辑，更新到文档 (修改人->梁家誌)

34.1 优化后逻辑：扫描到不同的蓝牙设备则都设置一个500ms的延时，扫描到相同的蓝牙设备则后一个设备替换原来的设备且原延时不做修改。这样总体就只是慢了500ms，不会出现明显变慢的情况。

34.2 参考代码：“BTCentralManager.m”line661

35.

9月6日 普通添加流程优化 (修改人->梁家誌)

35.1优化：添加流程中，连接需要添加的前已经停止扫描，去掉修改mesh名称前多余的停止扫描操作。

36.

9月6日 正常登陆控制流程优化 (修改人->梁家誌)

36.1优化：登录成功后，为了错开发送数据包，0s、2s、4s发送setNotifyOpenPro，0.5s发送setTime，存在meshOTA功能的，1s发送readMeshOTAState。蓝牙连接异常断开时，上面定时器和延时都需要停止掉。

36.2添加设置时间API：setTime

37.

9月6日 demo代码与SDK代码分离 (修改人->梁家誌)

37.1 把demo部分代码与SDK代码分离，形成Telink的维护的静态库工程TelinkMeshLib.xcodeproj，可编译出静态库libTelinkMeshLib.a。开发者可以很方便的把静态库工程拖到直接的项目并设置头文件搜索路径就可以使用了。

37.2 添加TelinkMeshLib.xcodeproj源代码库使用说明。

38.

9月20日SDK代码添加RSSI排序处理。(修改人->梁家誌)

38.1 在34的修改基础上，优先处理500ms内RSSI较大的设备，且RSSI为正数则直接过滤掉不处理。所以，添加时优先添加RSSI信号强的设备，正常连接也优先连接RSSI信号强的。

39.

MeshOTA注意事项：

1.开始MeshOTA接口startMeshOTAWithDeviceType需要传入设备的类型和需要升级的OTA文件，通过普通模式添加的设备可以在添加过程中获取到设备类型，通过Mesh模式添加的设备，当前是没有获取到设备类型的。

40.

2019年09月24日SDK兼容iOS13系统的优化。(修改人->梁家誌)

40.1 iOS13后，系统类NSData的对象方法- (NSString \*)description;发生改变。SDK新增方法- (NSString \*)getDescriptionStringWithData:(NSData \*)data;用于替代系统方法。

41.

补充meshOTA的新增优化逻辑：(修改人->梁家誌)

41.1 新增接口从Bin文件中读取它的设备类型。

41.2 但只有一个设备时，直接走GATT的OTA，不走Mesh OTA。

41.3 为实现41.2的功能，单独抽出GATT的OTA相关文件，形成OTAManager管理类和独立的startOTA、stopOTA接口。

42.

新增宏kMacComplement用于补全蓝牙的广播MacAddress地址，默认为0xFFFF。(修改人->梁家誌)

通过方法- (NSString \*)getMacAddressFromU\_Mac;即可获取完整的六个字节的MacAddress字符串。