Nut iOS SDK 使用说明

时间 文档版本 备注

|  |  |  |
| --- | --- | --- |
| 2016/12/29 | 0.1 | 初稿 |
|  |  |  |

**1. 开发环境**

iOS 8.0及以上

**2. 接口定义**

NTDeviceState

typedef NS\_ENUM(NSUInteger, NTDeviceState) {

NTDeviceStateDisconnected = 0,

NTDeviceStateConnecting,

NTDeviceStateConnected,

NTDeviceStateDisconnecting

};

NTDeviceDelegate

/\* It will be called after the device connected to the phone. \*/

- (void)deviceDidConnect:(NTDevice \*)device;

/\* It will be called after the device disconnected to the phone. \*/

- (void)deviceDidDisconnected:(NTDevice \*)device;

/\* It will be called if the connection attempted failed due to the error \*/

- (void)device:(NTDevice \*)device didFailedToConnect:(NSError \*)error;

/\* It will be called after you call readRSSI on a NTDevice object \*/

- (void)device:(NTDevice \*)device didUpdateRSSI:(NSNumber \*)RSSI;

/\* It will be called when you clicked the button on the device. The param numberOfClick could be 0, 1 or 2. When numberOfClick is 0, it means long press(press and hold for 2 seconds) \*/

- (void)device:(NTDevice \*)device didClicked:(NSInteger)numberOfClick;

NTDevice

/\* The delegate of the NTDevice object \*/

@property (nonatomic,weak) id <NTDeviceDelegate> delegate;

/\* The identifier is a system generated UUID for each device. See CBPeripheral's identifier for more information\*/

@property (nonatomic,readonly) NSString \*identifier;

/\* The local name advertised by the device. \*/

@property (nonatomic,readonly) NSString \*localName;

/\* The fireware version read from the device. \*/

@property (nonatomic,readonly) NSString \*firmwareVersion;

/\* The hardware version read from the device. \*/

@property (nonatomic,readonly) NSString \*hardwareVersion;

/\* Setting this property to YES will trigger auto reconnect after the device disconnected or failed to connect \*/

@property (nonatomic,assign) BOOL autoReconnect;

/\* The device connection state, see NTDeviceState emun for more information \*/

@property (nonatomic,assign) NTDeviceState state;

/\* The device's battery level, ranged from 0 to 100 \*/

@property (nonatomic,readonly) NSNumber \*battery;

/\* The unique identifier read from the device \*/

@property (nonatomic,strong,readonly) NSString \*deviceID;

/\* Used to connect to the device, if the device is already connected, this API does nothing.\*/

- (void)connect;

/\* Used to disconnect the device, or cancel a pending connection attempt \*/

- (void)cancelConnection;

/\* If the device is connected, calling this method will send a command to beep the device for the duration passed in. The maximum timeout is 30 seconds\*/

- (BOOL)beep:(BOOL)enabled withTimeOutDuration:(NSInteger)timeout;

/\* Enabling or disabling the device alarm when the device disconnected from the cell phone \*/

- (BOOL)setHardwareAlarmEnabled:(BOOL)enabled;

/\* Read the RSSI from the device, the result will be send back from delegate method -device:didUpdateRSSI: \*/

- (void)readRSSI;

NTDeviceManager

/\* The openID from the SDK provider \*/

@property (nonatomic,strong) NSString \*openID;

@property (nonatomic,weak) id<NTDeviceManagerDelegate> delegate;

/\* Returns the singleton instance \*/

+ (NTDeviceManager \*)sharedManager;

/\* Initialize the CBCentralManager. However you don't need to call this method unless you've already called destroyCentralManager. \*/

- (void)initCentralManager;

/\* Destroy the CBCentralManager object to clean up BLE objects \*/

- (void)destroyCentralManager;

/\* Start scan nut devices, optionDict accepts NTCentralManagerScanOptionAllowDuplicatesKey

\* as key and NSNumber as a boolean wrapper to disable duplicate filtering

\*/

- (void)startScanWithOptions:(NSDictionary \*)optionDict;

/\* Stop scanning devices \*/

- (void)stopScan;

/\* Retrieve the NTDevices object with identifiers. The identifiers param is an array of CBPeripheral's identifiers\*/

- (NSArray \*)retrieveDevicesWithIdentifiers:(NSArray \*)identifiers;

/\* Option key for startScanWithOptions \*/

NSString \*const NTCentralManagerScanOptionAllowDuplicatesKey;

NTDeviceManagerDelegate

/\* The method will be called when the CBCentralManagerState changes. \*/

- (void)deviceManager:(NTDeviceManager \*)manager didUpdateState:(CBCentralManagerState)state;

/\* The method will be called when a device is discovered \*/

- (void)deviceManager:(NTDeviceManager \*)manager didDiscoveredDevice:(NTDevice \*)device RSSI:(NSNumber \*)rssi;

3. 须知

请注意，为了正常使用SDK，务必在App初始化之后，配置正确的OpenID，厂家的OpenID请向SDK提供商索取。

4. 工程配置

在工程配置页Capabiliies里的Background Modes里，打开 “Use Bluetooth LE accessories”.

