iOS SDK Development Manual

Environmental requirements 3

iOS Static Library Integration Guide 3

1.Import the SDK 3

2. Set project properties 4

3. Use the SDK in code 4

The function interface 5

Connect the lock 5

Disconnect lock 5

Get lock status 5

Get lock power 6

Unlock 6

Lock 7

Configure lock 7

Get lock record 7

Delete lock record 8

Fingerprint controller add fingerprint 8

Fingerprint controller delete fingerprint 8

Fingerprint controller binding lock 9

Add password 9

Password deletion 9

Dynamic password acquisition 10

Get synchronization time string 10

Bluetooth synchronization time 10

Upgrade lock firmware 10

Seamoon Cloud Technology Ltd.

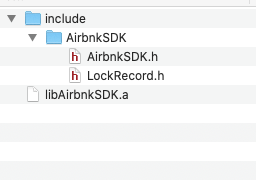
### Environmental requirements

Minimum support for iOS10.0.

### iOS Static Library Integration Guide

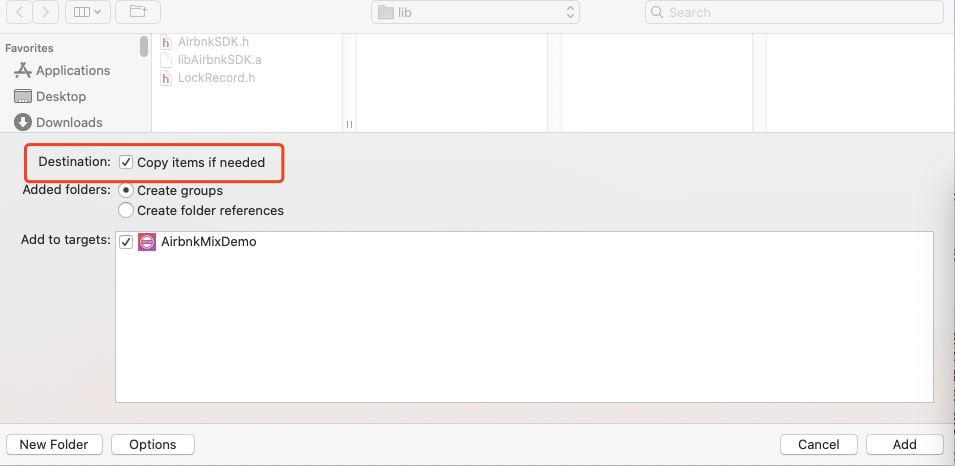
#### 1.Import the SDK

Step 1: Prepare the header file and.A file. As shown in figure:

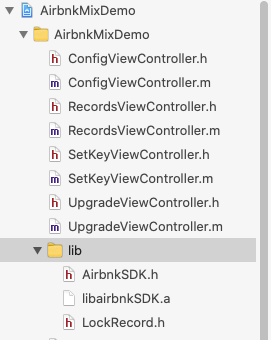


libAirbnkSDK.a is the static library, airbnksdk.h is the header file, lockrecord.h is a header file.

Step 2: Copy the file to the project directory, and check Destination. As shown in figure:

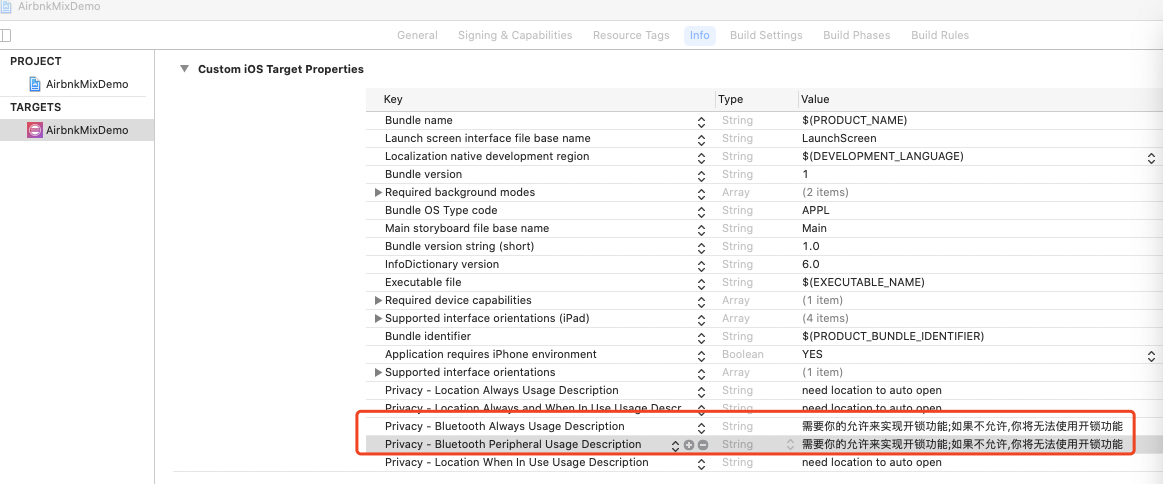


Step 3: Add SDK. The project directory is shown in the figure：



#### Set project properties

1. Bluetooth permission Settings. You need to manually add the appropriate permissions in the info.plist file of the project, as shown in the figure below：



#### Use the SDK in code

1. The SDK needs to be initialized before invoking the functional interface.

- (BOOL)application:(UIApplication \*)application didFinishLaunchingWithOptions:(NSDictionary \*)launchOptions {

    [AirbnkSDK initAirbnkSDK];

return YES;

}

1. Set up the delegate where necessary

@interface LockViewController ()<AirbnkSDKDelegate>

@end

@implementation LockViewController

- (void)viewDidLoad {

    [super viewDidLoad];

    [AirbnkSDK handleDelegate:self];

}

1. Method invocation

Get the lock status：

[AirbnkSDK getLockStatus:SNINFO];

Lock status callback：

-(void) responeseLockStatus:(LockState) state {

    if (state == LockStateLocked) {//door is locked

    } else if(state == LockStateUnLock) {//door is unlocked

    } else {//door lock not well

    }

}

### The function interface

The SDK provides the mobile Bluetooth and Airbnk lock interface. The functions include connection lock, disconnection lock, access to lock state, access to lock power, unlock, lock, configure lock information, access to lock record and so on. The SDK interacts with the lock in an asynchronous callback mode. When the SDK calls the interface, if there is lock information and status return, the result will be returned in the proxy callback method.

Sdkkey: the secret key of Airbnk lock SDK. The SDK needs to be called once during initialization.

SNINFO: Airbnk lock unlocks the key. Each lock corresponds to a different SNINFO. The SDK needs to take SNINFO parameters when it interacts with the lock.

**The connection lock method is called before any interaction with the lock.**

#### Connect the lock

+(void) connectToLock:(nullable NSString\*) sninfo;

Parameter description：

sninfo Airbnk locks decryption string

The callback function

-(void) didConnectLock:(nullable NSString\*) sninfo;

#### Disconnect lock

+(void) disconnectToLock:(nullable NSString\*) sninfo;

Parameter description：

sninfo Airbnk locks decryption string

The callback function

-(void) didDisConnectLock:(nullable NSString\*) sninfo;

#### Get lock status

+(void) getLockStatus:(nullable NSString\*) sninfo;

Parameter description：

sninfo Airbnk locks decryption string

The callback function

-(void) responeseLockStatus:(LockState) state;

Parameter description：

LockState：

LockStateLocked,

    LockStateUnLock,

    LockStateLockedOpen,

    LockStateLockedClose,

    LockStateUnLockOpen,

    LockStateUnLockClose,

    LockStateJammed

#### Get lock power

+(void) getLockVoltage:(nullable NSString\*) sninfo;

Parameter description：

sninfo Airbnk locks decryption string

The callback function

-(void) responseLockVoltage:(LockVoltage) voltage;

Parameter description：

LockVoltage：

LockVoltageHigh,

    LockVoltageMiddle,

    LockVoltageLow,

    LockVoltageEmpty

#### Unlock

+(void) unlock:(nullable NSString\*) sninfo;

Parameter description：

sninfo Airbnk locks decryption string

The callback function

-(void) responeseLockStatus:(LockState) state;

Parameter description：

LockState：

LockStateLocked,

    LockStateUnLock,

    LockStateLockedOpen,

    LockStateLockedClose,

    LockStateUnLockOpen,

    LockStateUnLockClose,

    LockStateJammed

#### Lock

+(void) lock:(nullable NSString\*) sninfo;

Parameter description：

sninfo Airbnk locks decryption string

The callback function

-(void) responeseLockStatus:(LockState) state;

Parameter description：

LockState：

LockStateLocked,

    LockStateUnLock,

    LockStateLockedOpen,

    LockStateLockedClose,

    LockStateUnLockOpen,

    LockStateUnLockClose,

    LockStateJammed

#### Configure lock

+(void) configLock:(nullable NSString\*) sninfo openDirection:(NSInteger) direction autoLock:(NSInteger) autoLock autoLockTime:(NSInteger) autoTime doorSensor:(NSInteger) doorSensor doorSensorAutoLockTime:(NSInteger) doorSensorAutoLockTime handRotation:(NSInteger) handRotation latchTime:(NSInteger) latchTime;

Parameter description：

sninfo Airbnk locks decryption string

direction direction 1 open left 2 open right

autoLock autoLock lock 1 yes 0 no

autoTime AutoTime lock time is valid when autoLock=1, per second

doorSensor DoorSensor 1 yes 0 no

doorSensorAutoLockTime enable doorsensor valid when doorSensor=1 per second

handRotation handRotation is disabled 1 disabled 0 enable

latchTime tongue lock time unit seconds, no such configuration of lock fill -1.

The callback function

-(void) responseConfigLock:(BOOL) state;

#### Get lock record

+(void) getLockRecords:(nullable NSString\*)snInfo recordType:(LockRecordType) type;

Parameter Description:

Sninfo airbnk lock decryption string

Type lock record type lockrecordtypeeeprom is with EEPROM lock

typedef NS\_ ENUM(NSInteger, LockRecordType) {

    LockRecordTypeNormal = 0,

    LockRecordTypeEEProm,

}

Callback function

-(void) responseLockRecords:(nullable NSArray<LockRecord\*>\*) records;

Return Parameter Description:

Records lock record list. Lockrecord lock record entity refers to lockrecord.h

#### Delete lock record

+(void) deleteLockRecords:(nullable NSString\*)snInfo;

Parameter Description:

Sninfo airbnk lock decryption string

Callback function

-(void) responseDeleteLockRecords:(BOOL) state;

Return Parameter Description:

state is deleted successfully

#### Fingerprint controller add fingerprint

+(void) registerFingerprint:(nullable NSString\*)snInfo;

When the fingerprint controller is connected, the fingerprint can be entered

Parameter Description:

Sninfo aibike fingerprint control decryption string

Callback function

-(void) responseRegisterStatus:(NSInteger) status progress:(NSInteger) progress fingerprintId:(NSInteger) fpId;

Return Parameter Description:

Status 0 entering fingerprint 1 fingerprint successfully 2 fingerprint failed 3 fingerprint entered

Progress entry percentage

FPID returns the fingerprint ID successfully entered

#### Fingerprint controller delete fingerprint

+(void) deleteFingerprint:(nullable NSString\*)snInfo fingerprintId:(NSInteger) fpId;

After the fingerprint controller is connected, the entered fingerprint can be deleted according to the fingerprint ID

Parameter Description:

Sninfo aibike fingerprint control decryption string

FPID fingerprint ID

Callback function

-(void) responseDeleteFingerprint:(BOOL) state;

Return Parameter Description:

state is deleted successfully

#### Fingerprint controller binding lock

+(void) bindLock:(nullable NSString\*)snInfo lockSninfo:(nullable NSString\*) lockSninfo;

When the fingerprint controller is connected, the lock can be bound to the fingerprint controller

Parameter Description:

Sninfo airbnk fingerprint control decryption string

Locksinfo is the decryption string of the bound lock

Callback function

-(void) responseBindLock:(BOOL) state;

Return Parameter Description:

state bound successfully

#### Add password

+(void) addPassword:(nullable NSString\*)snInfo pwdIndex:(NSUInteger) pwdIndex password:(nullable NSString\*) password startTime:(NSUInteger) startTime endTime:(NSUInteger) endTime;

When the controller or lock with password keyboard is connected, the fixed password can be added

Parameter Description:

Sninfo airbnk lock decryption string

Pwdindex password index, value range [0-19]

Password set password, 6-10 digit string

The unit of time for the password to take effect is minutes

Endtime password effective end time unit is minutes

Callback function

-(void) responseAddPassword:(Bool) state;

Return Parameter Description:

state is added successfully

#### Password deletion

+(void) deletePassword:(nullable NSString\*)snInfo pwdIndex:(NSUInteger) pwdIndex;

When the controller or lock with password keyboard is connected, the added fixed password can be deleted

Parameter Description:

Sninfo airbnk lock decryption string

Pwdindex password index, value range [0-19]

Callback function

-(void) responseDeletePassword:(Bool) state;

Return Parameter Description:

state is deleted successfully

#### Dynamic password acquisition

+(nullable NSString\*) getOneTimePassWord:(nullable NSString\*)snInfo;

Get a valid unlock dynamic password

Parameter Description:

Sninfo airbnk lock decryption string

Return Parameter Description:

6-bit dynamic password

#### Get synchronization time string

+(nullable NSString\*) getOneTimePassWord:(nullable NSString\*)snInfo;

When the password keyboard time is offset, input the synchronization time string to synchronize the time

Parameter Description:

Sninfo airbnk lock decryption string

Return Parameter Description:

14 bit synchronous time encryption string

#### Bluetooth synchronization time

+(void) syncTimeBle:(nullable NSString\*)snInfo;

When the controller or lock with password keyboard is connected, the time can be synchronized through Bluetooth

Parameter Description:

Sninfo airbnk lock decryption string

Callback function

-(void) responseSyncTimeBle:(Bool) state;

Return Parameter Description:

state is synchronized successfully

#### Upgrade lock firmware

+(void) upgradeFirmWare:(nullable NSData\*) firmware deviceSnInfo:(nullable NSString\*) snInfo;

Through this interface, the firmware of the lock can be upgraded

Parameter Description:

Firmware binary data

Sninfo airbnk lock decryption string

Callback function

-(void) responseUpgradeFirmWare:(BOOL) state progress:(float) progress;

Return Parameter Description:

Progress upgrade progress [0-1.0]

State is upgraded