

SL110 SDK Guidance For IOS

Version 1.0

June 24, 2014

StrongLink

Contents

1.Introduction.....	3
2.Interface Definitions.....	3
2.1.IvrJackService	3
2.1.1 uid	错误! 未定义书签。
2.1.2 open	3
2.1.3 close	3
2.1.4 readTag.....	3
2.1.5 getTagUID.....	4
2.1.6 tagAuthenticate(select sector).....	4
2.1.7 tagReadData	4
2.1.8 tagReadBlockData	4
2.1.9 tagWriteData	4
2.1.10 return value	5
2.2 IvrJackAdapter	5
2.2.1 onConnect.....	5
2.2.2 onDisconnect	5
2.2.3 onStatusChange.....	5
2.3 IvrJackStatus	6
3.Call procedures	6
3.1 Interface calling	6
3.2 Tag operation	6

1.Introduction

LiblvrJack.a supports interfaces for the platform of iOS 4.3 or above. Major interfaces are as followings:

- a) IvrJackService — Communication Controlling Interface
- b) IvrJackDelegate — Event Interface
- c) IvrJackStatus — State

Attention: It is needed to quote AudioToolbox.framework、AVFoundation.framework、MediaPlayer.framework in the project.

Refer to the event of iReader.

2.Interface Definitions

2.1 IvrJackService

2.1.1 open

Definition: -(void) open;

Explanation: Start the server for the reader.

2.1.2 close

Definition: -(void) close;

Explanation: Close the server for SL110 reader.

2.1.3 didEnterBackground

Definition: -(void) didEnterBackground;

Explanation: Call the method to inform the server of handling background when application is entering background.

2.1.4 didBecomeActive

Definition: -(void) didBecomeActive;

Explanation: Call the method to inform the server of handling activation when activating application.

2.1.5 readTag

Definition: -(int) readTag;

Explanation: Open tag detection.

Return: 0 represents success, else failure.

2.1.6 getTagUID(former version)

Definition: -(int) getTagUID(UInt8[])ids cardType(UInt8*)cardType;

Explanation: Get the UID of tag.

Param return: Return UID when successing.

Return: 0 represents success, else failure.

2.1.7 getTagUID

Definition: `-(int) getTagUID:(unsigned int*)uid;`

Explanation: Get the UID of tag.

Param return: Return UID when successing,ids[0] is the length of UID, the others are data.

Return: 0 represents success, else failure.

2.1.8 tagAuthenticate

Definition: `-(int) tagAuthenticate(UInt8)sector pwds(UInt8[])pwds;`

Explanation: Verify the key of sector.

Parameter: a) sector: sector number.

b) pwds: 6-byte key of the sector (Default: key A).

Return: 0 represents success, else failure.

2.1.9 tagReadData

Definition: `-(int) tagReadData(UInt8)block data(UInt8[])data;`

Explanation: Read the tag data.

Parameter: a) block: the block number(0-3).

b) data :16-byte block data.

Return: 0 represents success, else failure.

2.1.10 tagReadBlockData

Definition : `-(int) tagReadBlockData:(UInt8)sector1 block1:(UInt8)block1
sector2:(UInt8)sector2 block2:(UInt8)block2 pwds:(UInt8[])pwds
data:(UInt8[])data;`

Explanation: Read the data of two blocks.

Parameter: a) sector1: the sector number(0-31).

b) block1: the block number of the sector 1(0-3).

c) sector2: the sector number(0-31).

d) block2: the block number of the sector 2(0-3).

e) pwds: 6-byte key of the sector (Default: key A).

f) data: data of the sectors.

Return: 0 represents success, else failure.

2.1.11 tagWriteData

Definition: `-(int) tagWriteData:(UInt8)block data:(UInt8[])data;`

Explanation: Write the tag data.

Parameter: a) block: the block number(0-3).

b) data :16-byte data to be written (Default: key A).

Return: 0 represents success, else failure.

2.1.12 tagReadDataUL

Definition: -(int) tagReadDataUL:(UInt8)block length(UInt8)length data(UInt8)data;

Explanation: Read the data of Ultraright.

Parameter: a) block: the starting block.

b) length: the number of block (1-4),4 blocks can be read at the most.

c) data: the data of blocks, equals length*4 .

Return: 0 represents success, else failure.

2.1.13 tagWriteDataUL

Definition: -(int) tagWriteDataUL:(UInt8)block length(UInt8)length data:(UInt8[])data;

Explanation: Write the Ultraright data.

Parameter: a) block: the starting block.

b) length: the number of block (1-4),4 blocks can be read at the most.

c) data: the data of blocks, equals length*4 .

Return: 0 represents success, else failure.

2.1.14 return value

a) 0: success.

b) 1: communication failure.

c) 2: unknown error.

d) -1: battery insufficiency.

e) -2: no device

2.2 IvrJackAdapter

2.2.1 readerConnect

Definition: -(void) readerConnect(NSString*) deviceSN;

Explanation: Device connects successfully.

Parameter: a) deviceSN: the serial-number of device.

2.2.2 readerDisconnect

Definition: -(void) readerDisconnect;

Explanation: Trigger occurs when device disconnects.

2.2.3 readerStatusChange

Definition: void readerStatusChange(IvrJackStatus) status;

Explanation: The device state changes.

Parameter: a) status: the device state.

2.3 IvrJackStatus

- a) ijsDetecting: Detecting device.
- b) ijsRecognized: Having detected device.
- c) ijsUnRecognized: No device detected.
- d) ijsPlugout: The device is plugged out.

3.Call procedures

3.1 Interface calling

- a) Copy libIvrJack.a to the project directory.
- b) Include libIvrJack.a in the project directory.
- c) Import the following frameworks :

AudioToolbox.framework;

AVFoundation.framework;

MediaPlayer.framework.

- d) Others refer to examples.

Note : The interface implement using Objective-C++, so make sure that there is a file at least whose suffix is .mm. Another way is appointing the building method in project property, namely setting Xcode's Project->Edit Active Target->Build->GCC4.2-Language->Compile Sources As as "Objective-C++". You can change .m to .mm of any file.

3.2 Tag operation

- a) Call the method of readTag.
- b) Call the method of getTagUID.
- c) Call the method of tagAuthenticate(optional).
- d) Call the method of tagReadData(optional).
- e) Call the method of tagWriteData(optional).