Feitian provide hardware mobile reader, the code only can using FEITIAN mobile reader, which is Feitian bR301(bluetooth smartcard reader), iPad casing reader, iDock reader.

The source code apply CCID driver, as the hardware accordance CCID standard to design, and all of these product has certified MFi(made for iPhone/iPod/iPod). bR301 and iDock(iR301) certified EMV Level 1, means can using them on payment industry. more features, please check Feitian website.

This code only using for iOS, for android and PCs, please contact ben@ftsafe.com to get full SDK.

[Supported Platforms]

PC Platform:
Windows 2000/XP/2003/2008/Vista/Windows7+ Linux, Mac OS X, UNIX
Mobile Platform:
Android(support OTG)/iOS 3.13+/Blackberry

[iR301 Product Features]

*USB 2.0 Full Speed Device
*Compliant with PC/SC, CCID Standards
*Supports ISO-7816-1/2/3 T=0 and T=1 Protocol
*Supports ISO-7816 Class A,B and C Cards
*Supports Protocol and Parameters Selection (PPS)
*Certified EMV Level 1/MFi/CE/FCC/RoHS/FIPS
*Short Circuit Protection
*Support upgrading firmware through USB Cable

[bR301 Product Features]

*Support Bluetooth 3 and Bluetooth 4, using SPP profile
*Compliant with PC/SC, CCID Standards
*Supports ISO-7816-1/2/3 T=0 and T=1 Protocol
*Supports ISO-7816 Class A,B and C Cards
*Supports Protocol and Parameters Selection (PPS)
*Certified EMV Level 1/MFi/RoHS/CE/FCC/LETIC/FIPS
*Short Circuit Protection
*Support upgrading firmware through USB Cable
*Support through bluetooth work on Mobile and PC platform
*Provide re-chargeable 800mAh battery. Running time can keep 10-11h, standard at least 100h
*Four lights to inform reader/card/battery status
*Reader support customisable

[Run environment]
Mac Pro/10.9+/xcode 5.0+

[For developer]
0. Add External Accessory Framework into your project
1. Submit to Appstore need provide your App information to Feitian, we need register to MFi product plan, then give customer PPID put into review notes. The more information, please download FAQ as well.

2. Before using this code, please confirm you have add "Supported external accessory protocols" in your Info.plist file, and add below protocol as well:

```
<array>
    <string>com.ftsafe.iR301</string>
    <string>com.ftsafe.bR301</string>
</array>
```

3. To support running in background, please confirm you have add "Required background modes" in your Info.plist file, add below strings:

```
    <string>App communicates with an accessory</string>
```

4. To monitor reader and card slot status, please set delegate in your code. The best place is set delegate before create context. Because it will list all all reader and card slot status while create context, through delegate to inform developer the status of reader and card slot.

They key code is -(void) setDelegate:(id)<ReaderInterfaceDelegate>)delegate

The second way is through SCardGetStatusChanges, this API we called ScardStatus inernal. we don't suggestion using this way.

SCARD_ABSENT -> no smartcard

SCARD_PRESENT || SCARD_SWALLOWED || SCARD_POWERED -> smartcard inserted

5. If you application support background running, to be safe, please do release context using SCardReleaseContext API when enter to background
6. If using dukpt communication, please set encryption mode before power on, using FtDukptSetEncMod API. We also apply lock mechanism, the locker will open after power on, and will unlock after power off or reader plug-out. Then can change encryption mode.

For the DUKPT(Derived unique key per transaction), we support two ways encryption(Input and output channel), single way encryption(only output channel), plain ways(without encryption). The key management we used DUKPT 2006
7. To debug your application, bR301 and iR301 using same SDK, so easy to debug with bR301 and using them together.
8. Support SLE4442 memory card, please check call_lib_sle sample code

For product specification information, please refer to FEITIAN website:

http://www.ftsafe.com/product/Smart_Reader
To buy a reader, please refer to http://www.ftsafe.com/product/reader_shop