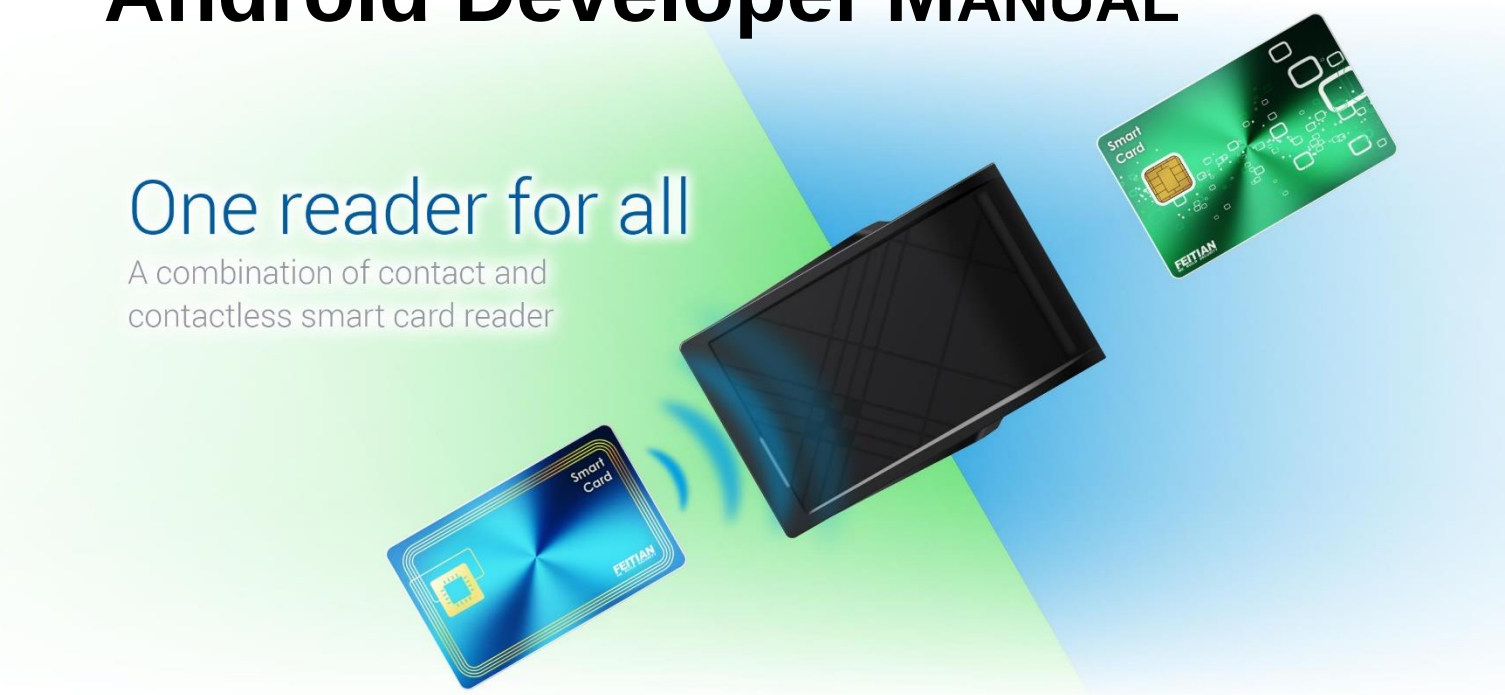
FEITIAN
WE BUILD SECURITY

R502 Card Reader
Android Developer MANUAL



Made by: Feitian Technologies

Mar, 2016

Revision Sheet

Date	Revision	Description
Dec, 2017	V1.0	Release the first version

Software Developer's Agreement

All Products of Feitian Technologies Co., Ltd. (Feitian) including, but not limited to, evaluation copies, diskettes, CD-ROMs, hardware and documentation, and all future orders, are subject to the terms of this Agreement. If you do not agree with the terms herein, please return the evaluation package to us, postage and insurance prepaid, within seven days of their receipt, and we will reimburse you the cost of the Product, less freight and reasonable handling charges.

1. Allowable Use – You may merge and link the Software with other programs for the sole purpose of protecting those programs in accordance with the usage described in the Developer's Guide. You may make archival copies of the Software.

2. Prohibited Use – The Software or hardware or any other part of the Product may not be copied, reengineered, disassembled, decompiled, revised, enhanced or otherwise modified, except as specifically allowed in item 1. You may not reverse engineer the Software or any part of the product or attempt to discover the Software's source code. You may not use the magnetic or optical media included with the Product for the purposes of transferring or storing data that was not either an original part of the Product, or a Feitian provided enhancement or upgrade to the Product.

3. Warranty – Feitian warrants that the hardware and Software storage media are substantially free from significant defects of workmanship or materials for a time period of twelve (12) months from the date of delivery of the Product to you.

4. Breach of Warranty – In the event of breach of this warranty, Feitian's sole obligation is to replace or repair, at the discretion of Feitian, any Product free of charge. Any replaced Product becomes the property of Feitian.

Warranty claims must be made in writing to Feitian during the warranty period and within fourteen (14) days after the observation of the defect. All warranty claims must be accompanied by evidence of the defect that is deemed satisfactory by Feitian. Any Products that you return to Feitian, or a Feitian authorized distributor, must be sent with freight and insurance prepaid.

EXCEPT AS STATED ABOVE, THERE IS NO OTHER WARRANTY OR REPRESENTATION OF THE PRODUCT, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

5. Limitation of Feitian's Liability – Feitian's entire liability to you or any other party for any cause whatsoever, whether in contract or in tort, including negligence, shall not exceed the price you paid for the unit of the Product that caused the damages or are the subject of, or indirectly related to the cause of action. In no event shall Feitian be liable for any damages caused by your failure to meet your obligations, nor for any loss of data, profit or savings, or any other consequential and incidental damages, even if Feitian has been advised of the possibility of damages, or for any claim by you based on any third-party claim.

6. Termination – This Agreement shall terminate if you fail to comply with the terms herein. Items 2, 3, 4 and 5 shall survive any termination of this Agreement.

USER'S MANUAL
TABLE OF CONTENTS

Page #

1.0 GENERAL INFORMATION

1.0 GENERAL INFORMATION

1.1 Product Introduction

R502 is a dual-interface smart card reader developed by Feitian Technologies. It is based on CCID driver. It supports not only contact cards compliant with ISO 7816 but also contactless cards compliant with ISO 14443 and contactless cards following Mifare standard. It also provides SIM card slots for many kinds of smart card applications. Moreover R502 comes with the SAM slot suitable for GSM 11.11 cards.

R502 is a terminal interface device for smart card applications and system integrations. With support for smart cards using different interfaces, R502 can be widely used in industries or applications requiring electronic payment and authentication, especially suitable for the high security fields. It is an optimal solution for authentication, e-commerce, financial organizations, access control etc.

1.2 Acronyms and Abbreviations

USB – Universal Serial Bus
CCID – (Chip Card Interface Device) Integrated Circuit(s) Card Interface Devices Specification
PCSC – (Short for "Personal Computer/Smart Card") is a specification for smart-card integration into computing environments.

2.0 DEVELOP APP ON Android

2.0 DEVELOP APP ON ANDROID

Before start development, something you need to know:

The **demo code** can find in SDK, path is
$R502_SDK_Latest\Demo\Java for Android

Notice: You will need add below information to your project.

```
<uses-feature android:name="android.hardware.usb.host" />
<uses-permission android:name="android.permission.USB_PERMISSION" />
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
```

2.1 API

2.1.1 readerFind

API: public void readerFind() throws FTException
Remark: Find device from Android, if no any device in Android, will throw exception
Parameters: N/A

2.1.2 readerOpen

API: public String[] readerOpen(Object device) throws FTException
Remark: Open card reader and return a list of reader name, if failure, will throw exception
Parameters:
Input: N/A
Output: Reader name list

2.1.3 readerClose

API: public void readerClose() throws FTException
Remark: Close reader, if failure, will throw exception
Parameters: N/A

2.1.4 readerPowerOn

API: public byte[] readerPowerOn(int index) throws FTException
Remark: Power on to card, if success will return ATR. If failure, will throw exception
Parameters:
Input: index is integer, from 0. It is index of reader name list
Output: ATR

2.1.5 readerPowerOff

API: public void readerPowerOff(int index) throws FTException
Remark: Power off the card, If failure, will throw exception
Parameters:
Input: index is integer, from 0. It is index of reader name list
Output: N/A

2.1.6 readerXfr

API: public byte[] readerXfr(int index,byte[] send) throws FTException
Remark: Through Application send data to reader, and reader got return data from card, then forward to app. If failure, will throw exception
Parameters:
Input:
Index: is integer, from 0. It is index of reader name list
Send: Send data to reader
Output: The data from card

2.1.7 readerEscape

API: public byte[] readerEscape(int index, byte[] send) throws FTException
Remark: Through Application send escape data to reader, and got return data from reader. If failure, will throw exception
Parameters:
Input:
Index: is integer, from 0. It is index of reader name list
Send: Send escape data to reader
Output: The data from reader

2.1.8 readerGetSlotStatus

API: public int readerGetSlotStatus(int index) throws FTException
Remark: Through Application send escape data to reader, and got return data from reader. If failure, will throw exception
Parameters:
Input:
Index: is integer, from 0. It is index of reader name list
Output:
com.ftsafe.DK.CARD_PRESENT_ACTIVE: Found card and powered
com.ftsafe.DK.CARD_PRESENT_INACTIVE: Found card and unpowered
com.ftsafe.DK.CARD_NO_PRESENT: No card

2.1.9 readerGetType

API: public int readerGetType() throws FTException
Remark: Get reader type. If failure, will throw exception
Parameters:
Input: N/A
Output: Return reader type, check definition com.ftsafe.DK

2.2.0 readerGetSerialNumber

API: public byte[] readerGetSerialNumber() throws FTException
Remark: Get reader serial number. If failure, will throw exception
Parameters:
Input: N/A
Output: Return reader serial number

2.2.1 readerGetSerialNumber

API: public String readerGetFirmwareVersion() throws FTException
Remark: Get reader firmware version. If failure, will throw exception
Parameters:
Input: N/A
Output: Return reader firmware version

2.2.2 readerGetUID

API: public byte[] readerGetUID() throws FTException
Remark: Get reader UID. If failure, will throw exception
Parameters:
Input: N/A
Output: Return reader UID

2.2.3 readerGetLibVersion

API: public static String readerGetLibVersion()
Remark: Get lib version. If failure, will throw exception
Parameters:
Input: N/A
Output: Return lib version