**UnifingerUI for Android**

**using SFM-SDK**

**Table of Contents**

**1. Introduction**…..……………………………………………………………………………………………………… *1*

**2. Enviroment** ..………………………………………………………………………………………………………… *1*

2.1 Compile environment………………………………………………………………………………………...… *1*

2.2 Test environment……………………………………………………………………………………………….… *1*

**3. License**………………………………………………………………………………………………………….……… *2*

**4. Implement app**……………………………………………………………………………………………….…… *2*

4.1 Implement write communication function…..…………………………………………………….…… *2*

4.2 Implement read communication function……………………..………………………………….…… *3*

4.3 Implement write callback function………………………………………………………………….….… *3*

4.4 Implement read callback function………………………………………………………………….….… *4*

4.5 Connect to communication function with SDK using JNI……………………………………...… *4*

**5.SFM-SDK**……………………………………………………………………………………………………….….… *4*

5.1 How to call SFM-SDK function…………………………………………………………………….…..… *5*

5.2 Example android project………………………………………………………………………………...… *6*

1. **Introduction**

This android app uses USB serial communication using FTDI driver. But you can connect any other communication ways.(Within the limits that the module supports.) You just have to connect the SDK callback function to the communication function on Android(See the text for details). Android sfm-sdk was made by C language and compiled using NDK. UnifingerUI app is using SFM-SDK in JNI. This app version is beta 1.2(using SDK beta 1.0) and we checked about some SDK functions were working well. (you can see list of working functions identified in SDK)

1. **Environment**
   1. **Compile environment**
      * IDE : Android studio version -2.1.2
      * NDK : NDK version - r10e
      * SDK : Compile SDK version 22
   2. **Test environment**
      * APK Install device : Samsung Galaxy J5(On5) prime(android 6.0.1)
      * Cables : USB To RS232 cable : KW835(FTDI) / USB Mini 5P OTG cable
      * Test module and EVK : SFM6020,SFM5020 / SFM EVK



Application working picture

1. **License**

This android app uses USB serial communication using FTDI driver. This driver license is available on the FTDI website at http://www.ftdichip.com/Drivers/FTDriverLicenceTermsSummary.htm. Suprema has ownership this software(program source code and the production outputs) and the used SFM SDK(except FTDI library). So you must give any other recipients of the Work or Derivative Works a copy of this License. And You must cause any modified files to carry prominent notices stating that You changed the files.

1. **Implement app**

You have to implement read communication and write communication to use SFM-SDK command in android. For example we implement read/write communication USB serial communication(using FTDI driver). If you use other communication or not use FTDI driver, you must create the function form like us(refer to below).

* 1. **Implement write communication function.**



Write communication function

Write communication function has to return the integer value that is a number to trans success data size. And we have to change write data type string to byte array(if you do not want the byte type, you can change it to the appropriate type for writing the data.)

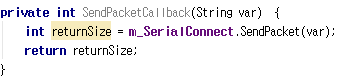
* 1. **Implement read communication function.**



Read communication function

Read communication function has to return the integer value that is a number to read success data size. And we implement two parameters. One is the wanted read data size(int size). And other is data buffer to input the really read data(byte [] read\_buffer). If you read data using read communication then you copy the data to read\_buffer(parameter).

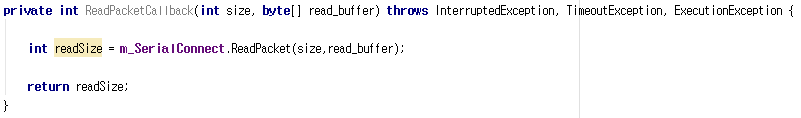
* 1. **Implement write callback function.**



Write callback function

We implemented the callback function calling write communication function. if you want to make callback by the write communication function then you can do that.

* 1. **Implement read callback function.**



Read callback function

We implemented the callback function calling read communication function. if you want to make callback by the read communication function then you can do that.

* 1. **Connect to communication function with SDK using JNI.**

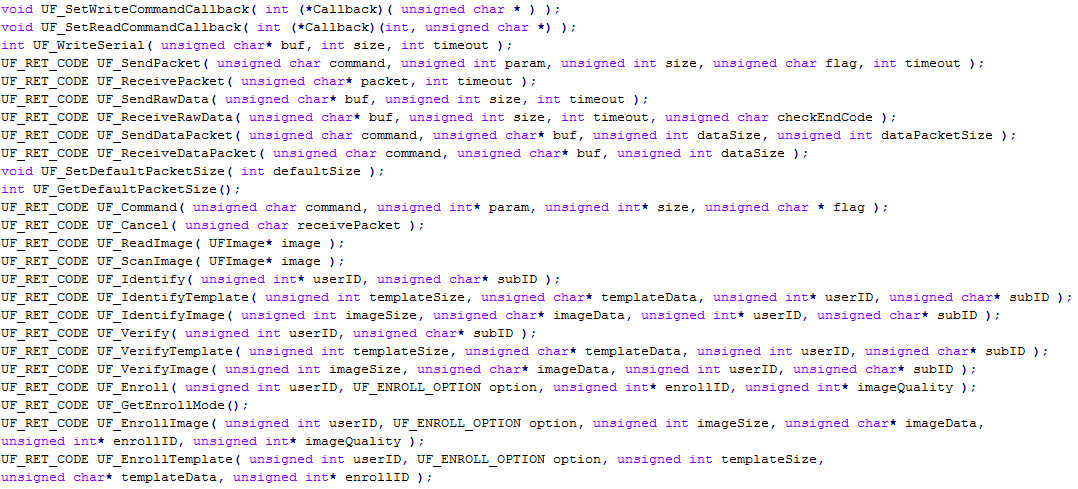


JNI functions to set communication with SDK

The functions used above is JNI functions to connect with SDK for communication. First, you have to let JNI know class name that is implemented callback function. Input the class name including package name to SetCommandClassName function. Next, Input the each function name(read/write call back function name) to SetReadCallbackFuctionName/SetWriteCallbackFuctionName function.

1. **SFM-SDK**

Android SFM-SDK file name is a libSFM\_SDK\_android.so. Unfortunately, we didn't check to work all library functions. This version is beta 1.1. But we guess that most of the functions work well. You can check functions that is supported to SDK in UF\_API.h. And You can see a list of functions whose behavior has been confirmed below.

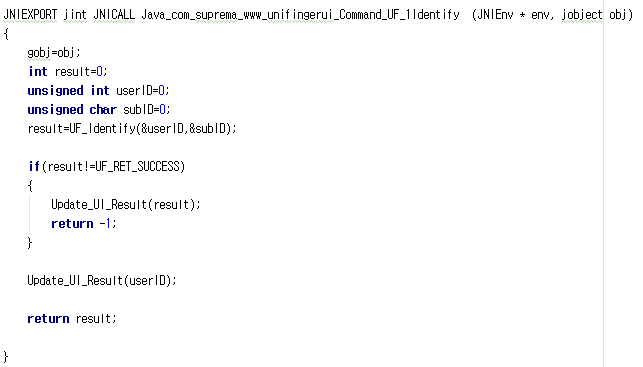


* 1. **How to call SFM-SDK function.**

SFM-SDK is built C language. So, you can call SFM-SDK function in JNI. First, check the form of the function in UF\_API.h. And make the native function in java class. Last, Call the SFM-SDK function in native function. We attached one example below.



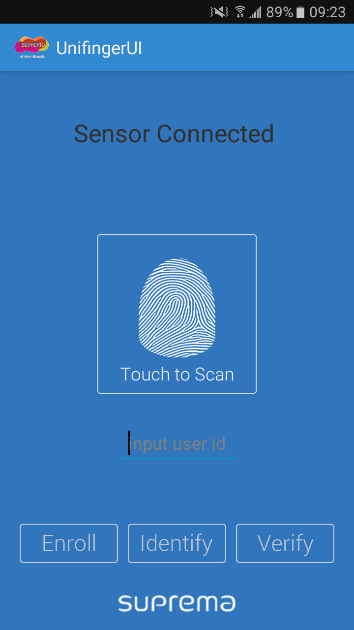
Native function in java class.



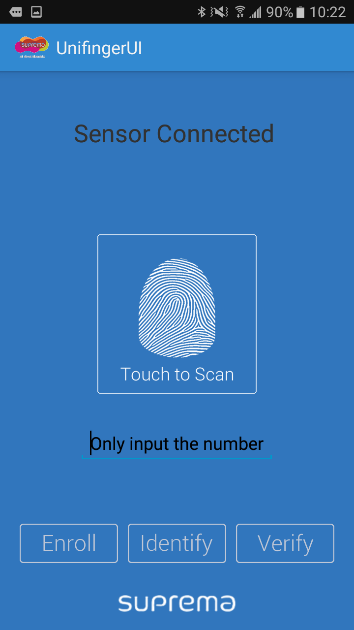
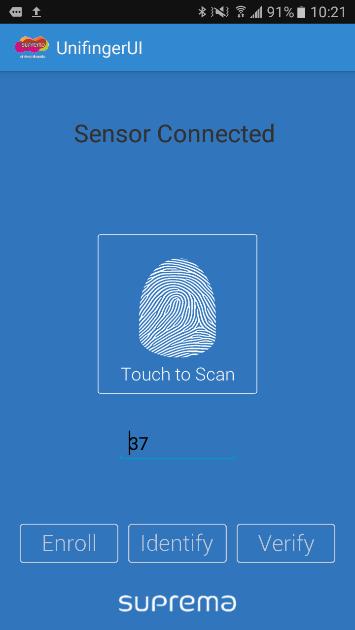
Call SFM-SDK function in Native function.

* 1. **Example android project.**

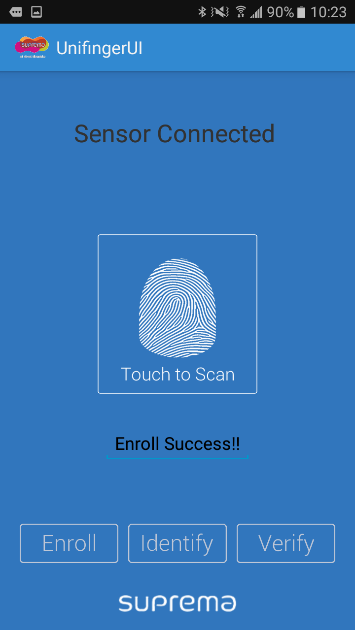
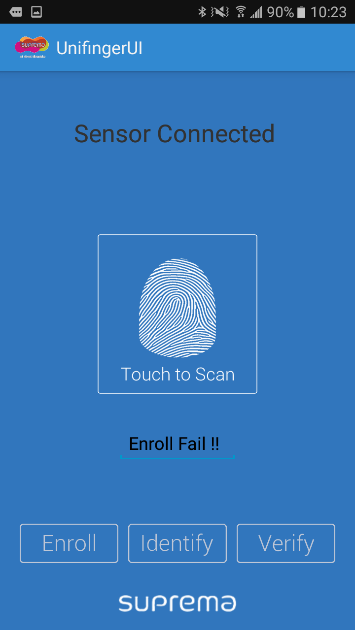
We provide an example project. But, it is not perfectly. We used callable for reading packet. but you maybe need to use thread(runnable) or you maybe need to call SFM-SDK function by the thread. You have to implement the program for your situation. We attached the picture of the sample program as shown below.



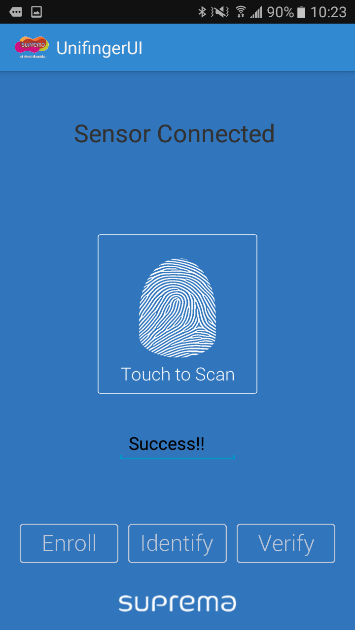
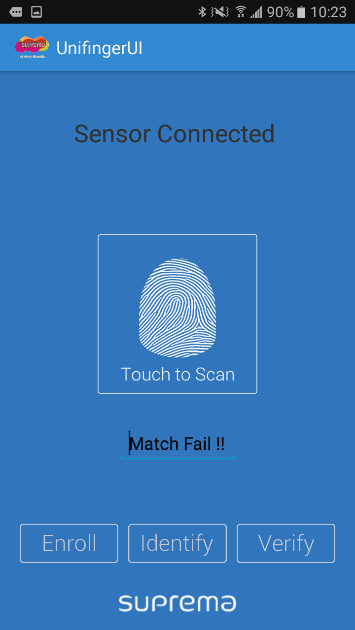
First app launch screen.



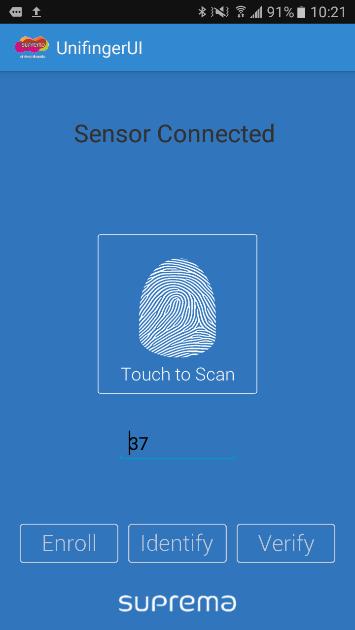
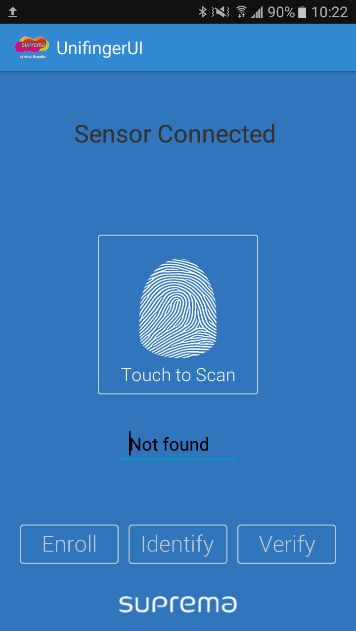
Only input the number for Enroll or verify.



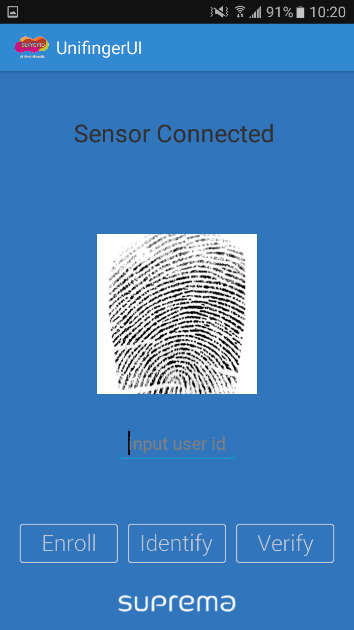
Launch screen after enroll.



Launch screen after verify.



Launch screen after identify



Launch screen after scan image.

(It may take a few seconds for the image to be transferred)