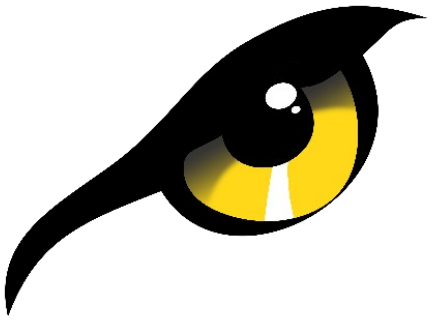
HawkEye Thai National ID Card

NALib SDK

Developer's Manual

for Android

R&D Computer System Co., Ltd.



NALib_Manual_Eng_R260520.odt

Table of Contents

Computer Software Developer License Agreement i
Part List of HawkEye NALib SDK 1
NALib Features 1
Application Development Steps 2
System Implementation After Application Development 3
Application Workflow 4
NALib API 5
- setListenerNA 5
- openLibNA 5
- closeLibNA 6
- getReaderListNA 6
- selectReaderNA 8
- deselectReaderNA 8
- connectCardNA 8
- disconnectCardNA 8
- getNIDNumberNA 8
- getNIDTextNA 9
- getNIDPhotoNA 11
- getCardStatusNA 11
- getRidNA 11
- getSoftwareInfoNA 11
- getLicenseInfoNA 12
- updateLicenseFileNA 12
- setPermissionsNA 12
- getReaderInfoNA 13
- getEmvTextCA 13
- Return Code Table 15

NADemo Usage 16
NASample And Source Code For Application Development 20
DLS License File Management For File Licensing Readers 29
Applications Development For Virtual Licensing Readers 30
App Permissions Requesting 31
Card Readers Connecting Conditions of NALib 33
Pop-up Window for Requesting USB Reader Permission Inside and Outside the Application 34
Card Readers Usage 36
NALib SDK New Features 42
FAQs About HawkEye NALib SDK 44

Computer Software Developer License Agreement

Please read the terms of this Agreement prior to entering into this Agreement by enabling this Computer Software.

After you have read the terms of this Agreement, if you don't contractually agree, please return the Computer Software that is on the disc or any other type of media, including documents and other attachments (if applicable) to the place where you received them within 7 days (seven days) from the date of receiving the Computer Software. If you received the Computer Software by downloading or copying, you must destroy or delete the Computer Software and all the copies from the hard disk, network data storage units, data storage units in the Internet network and other storage media immediately.

Once you have installed or use this Computer Software, even just one part, you agree to be bound as a licensee to use the Computer Software pursuant to this Agreement. The following in this Agreement is called "Licensee"

R&D Computer System Co., Ltd. is located at 48/20 Ratchadapisek Soi 20, Ratchadapisek Road, Sam Sen Nok, Huai Khwang, Bangkok, Thailand, called "Licensor".

Licensor and Licensee agree to enter into an Agreement with the following terms:

1. Permission to use Computer Software.

1.1 Licensor owns copyright of Thai ID Card Development Kit for Android (TDA), HawkEye TDA, NALib, NADemo, NASample, TDAService SDK, TDAService, TDA Java archive files, TDASample, TDADemo, sample programs source code and APK files, other computer programs and the documents specified in this Agreement ("Computer Software").

1.2 The Licensor agrees to grant the Licensee the use of the Computer Software described in this Agreement in the following manner:

- Licensee may use this Computer Software for development of the application programs.
- For the developed application programs which any portion of this Computer Software is included, merged, linked or ran, the application programs are considered part of this Computer Software. And the Licensee must treat the application programs as well as this Computer Software.
- Licensee will only use this Computer Software and application programs for normal business operations of Licensee.
- Licensee will only use this Computer Software with a smart card reader provided or designated by Licensor.
- There will be no more than 1 person (one person) per license (one right) is using Computer Software at the same time.
- This Agreement does not grant the Licensee to obtain the source code of the Computer Software, exclude sample programs.

2. Permit Period

2.1 Licensee will use this Computer Software from the date of commencement of the Agreement until terminated or until the Licensor terminates the license.

3. The rights and obligations of the Licensee.

3.1 The Licensee has the right to use the Computer Software and documents as specified in this Agreement.

3.2 The Licensee does not have the right to make, rent, loan, or resell any Computer Software or document or any rights granted to it under this Agreement.

3.3 In case the Licensee has developed an application program, Licensee may only distribute TDAService and TDA library files (JAR, AAR) to third parties in the manner permitted by licensing. And the Licensee must deliver this Agreement in the form of a paper document or computer file attached to the application program that are delivered to the other party. And

anyone else who receives the application program files will be required to be a Licensee of Computer Software under this Agreement.

3.4 The Licensee has the right to make a copy of Computer Software (one set) to prevent loss during the use.

3.5 The Licensee agrees not to alter, modify, delete, destroy, damage, or obscure the name of the Computer Software, name of the copyright owner or trademark, whether intentionally or negligently.

3.6 The Licensee will handle Computer Software, documents and copies by the best method. And will not allow anyone to access, use, study, copy or do any one of the Computer Software, documents and copies.

4. Non-reverse engineering and others.

Licensee will not do or try to do, on their own or authorize, refrain from assigning or encouraging others to do any of the following: modify, adapt, damage, reverse engineer, translate, reverse translate, decompile, disassemble, system analyze, black box test, decrypt, byte code debug, binary code debug, rebuild, deconstruct, sniff, simulate, emulate and any other actions that produce the same effect, no matter what the name is called, to the components of Computer Software, firmware, protocols, user interfaces, application programming interfaces, hardware and components.

5. Return Computer Software.

In case of Agreement terminated, Licensee must:

- Returns the Computer Software, documents and all copies to the Licensor immediately.
- Destroys or deletes Computer Software and all the copies from the hard disk, network data storage units, data storage units in the Internet network and other storage media immediately.

6. Trade Secrets

Licensee acknowledges that all information, technical information, procedures and processes relating to this Computer Software are the trade secrets of the Licensor. Licensee will not disclose any of these trade secrets to third parties and will try the best to protect these trade secrets.

7. Terms of Use

- Licensee will not use Computer Software for illegal processes.
- Licensee will serves or use this Computer Software for card holder only.
- Licensee will not read the data from the card without the permission of card holder.
- Licensee will not disclose or distribute or share card’s photos and text data to others without the permission of card holder.
- Strict compliance with the Personal Data Protection Act and related laws is required.
- Any damages caused by this Computer Software or the information got from this Computer Software, Licensee must be liable for all, including the damages of civil law and others. Licensor will not be held liable to and will not accept any liability, obligation or responsibility to all damages.

HawkEye Thai National ID Card NALib SDK Developer's Manual for Android

R&D Computer System Co., Ltd.

TDA NALib SDK is the software for Thai national ID card reading. It's an important part of Thai ID Card Development Kit for Android products.

1. Part List of HawkEye NALib SDK

1.1 One smart card reader, USB or Bluetooth type depends on the model.

1.2 NALib SDK (downloadable from Internet) that include:

- A simple application file (APK) and source code of NASample. This application is an easy sample for developers.
- The application file (APK) and source code of NADemo. This application will demonstrate the API of NALib for complex application development.
- NALib Android Archive file (NALib.aar) for building the application.
- This SDK manual.

2. NALib Features

2.1 Uses for smart card readers that support HawkEye NALib:

- USB readers: such as TDA3310M2, TDA39VC.
- Bluetooth 3.0 readers: such as TDA301BT.
- Bluetooth 4.0 Low Energy (BLE0) readers: such as TDA301BL, TDA301BLM, TDAi301BL and TDA301BLM.
- Bluetooth 4.0 Low Energy (BLE1) readers: such as TDA3901BE.

2.2 Supports both types of reader licensing:

- File Licensing (FL) readers which need a license file to operate such as TDA3310C2, TDA3310M2, TDA301BT and TDA3901BE.
- Virtual Licensing (VL) readers which don't need license file such as TDA301VC, TDAi301VC, TDA39VC, TDA39VM ,TDA301BL and TDA301BLM.

2.3 Supports Java and others with Android Studio.

2.4 Supports Kotlin, React Native and Flutter.

2.5 Supports Android 7.0 and higher versions.

2.6 Reads Thai national ID smart card of Thai Department of Provincial Administration (DOPA) as well.

2.7 It can read text information from bank cards, such as credit and debit cards (only with compatible card readers).

3. Application Development Steps

3.1 Buy one set of HawkEye products. It should be a Bluetooth reader that can be used for developing both Bluetooth and USB reader types.

- The Bluetooth reader is normally used wirelessly via Bluetooth. But the developer can purchase an OTG cable and connect it to the existing USB cable in the TDA package. Then plug into an Android device to develop NALib SDK in form of a USB reader.

3.2 Download the NALib SDK from the company web site, in HawkEye TDA product page.

3.3 Install NADemo application from NADemo.apk in SDK into a smartphone. Then try each NALib API command from the NADemo screen.

3.4 Study source code of NASample and NADemo.

3.5 Create a new project in Android Studio 3.2+.

3.6 Import NALib Android Archive file (NALib.aar) into the project.

3.7 Start application development by calling NALib.aar with NALib API.

3.8 Test the developed application.

3.9 Build an application file (APK) and distribute it to customers or upload it to Google Play.

Note

Developers should have some experience of Android application development before using the NALib SDK.

4. System Implementation After Application Development

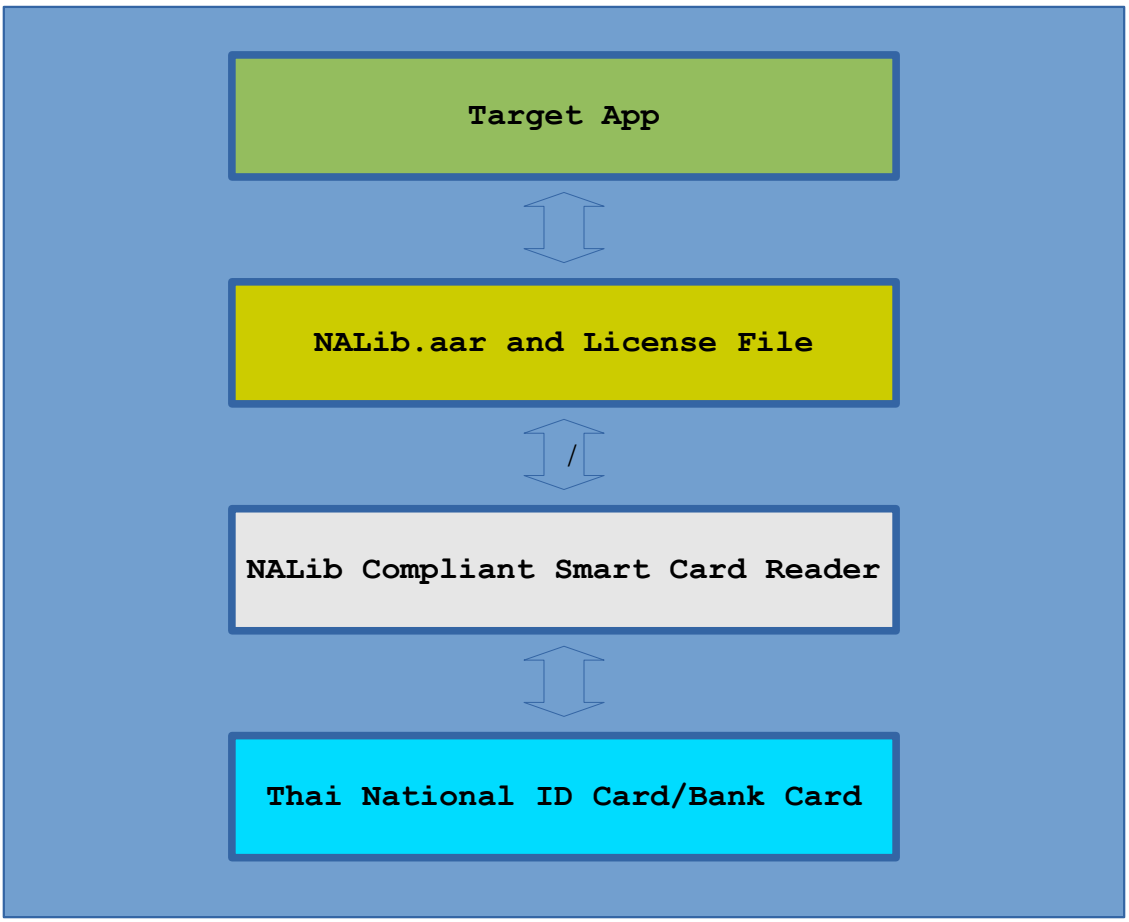
4.1 Let users purchase HawkEye products. Choose TDA301BL for Bluetooth (wireless) readers or choose TDA39VC, TDA3310C2 if require USB type readers.

4.2 For Bluetooth 3.0 readers, users have to pair the reader with Android system first. Go to the **Settings** menu, tap **Bluetooth** then **Scan** and **Pair**. However, for BLE (Bluetooth 4.0) reader, pairing is no need.

4.3 Install the developed application (Target App) from Google Play or from the APK file.

4.4 Let users start using the Target App.

5. Application Workflow



Target App

It is the developing application for reading Thai ID card.

NALib.aar and License File

NALib.aar is the NALib Android Archive file of NALib SDK. Developers have to import this file into the project. This library will support NALib API commands from Target App and executing them, such as reads data from Thai ID card.

License file is a file that collects the S/N of registered Thai ID card reader. NALib will check S/N of the working reader before processing the command. If the working reader is not registered, NALib will deny that command.

Note: License file is no need for VL readers.

NALib Compliant Smart Card Reader

It is the smart card reader that comes with a HawkEye SDK package which has already been registered. The unregistered reader will not work with NALib.

Thai National ID Card/Bank Card

It is the Thai national ID smart card or bank card.

6. NALib API

NALib API Commands

Command	Description
setListenerNA	Setup ResponseListener for each API command that Callback the results.
openLibNA	Start using the NALib command and assign the license file location.
closeLibNA	Stop using and clear memory.
getReaderListNA	Get reader name list from system.
selectReaderNA	Select a working reader.
deselectReaderNA	Deselect the current working reader.
connectCardNA	Connect the card that inserted in the reader.
disconnectCardNA	Disconnect the current card.
getNIDNumberNA	Read Thai national ID number (13-digit) from card.
getNIDTextNA	Read all text data from Thai national ID card.
getNIDPhotoNA	Read portrait photo from Thai national ID card.
getCardStatusNA	Get card status of the working reader, is absent or present.
getRidNA	Get unique Reader ID from the reader.
getSoftwareInfoNA	Get NALib library information.
getLicenseInfoNA	Get license information.
updateLicenseFileNA	Update the license file.
setPermissionsNA	Set permissions for NALib.
getReaderInfoNA	Get selected reader information.
getEmvTextCA	Read text data from bank card.

6.1 `void setListenerNA(ResponseListener listener)`

Description Setup ResponseListener for each API command that Callback the results.

Input Parameter **`listener`** for binding with the interface of each API command that Callback the results.

Note Must call this command first and before any other command, one time only.

6.2 `int openLibNA(String licenseFile)`

Description Start using NALib command and assign the license file location.

Input Parameter for FL+VL mode, **`licenseFile`** is the location of the license file, should be **rdnidlib.dls.** May be **`""`** (zero-length string) if used for VL only mode.

Return Value 0: Success, -1: Error

Callback Listener **`onOpenLibNA(int result)`**

Output to Listener **`result`** is a Return Code of **`openLibNA`**, see details in the **Return Code Table**.

6.3 `int closeLibNA()`

Description Stop using the library and clear memory.

Input Parameter None.

Return Value Return Code, see details in **Return Code Table**.

6.4 `int getReaderListNA(int listOption)`

Description Get reader name list from system.

Input Parameter **`listOption:`** Operating option bits for getting method. Defined Bit D7 to Bit D0 as shown in the below table.

D7	D6	D5	D4	D3	D2	D1	D0
NA_POPUP	NA_FIRST	NA_RSVD5	NA_SCAN	NA_BLE1	NA_BLE0	NA_BT	NA_USB
0x80	0x40	Reserved	0x10	0x08	0x04	0x02	0x01
128	64	Reserved	16	8	4	2	1

NA_USB (Bit D0, value 0x01) Getting USB reader name in the list

0 = Exclude USB readers

1 = Include USB readers

NA_BT (Bit D1, value 0x02) Getting Bluetooth 3 reader name (TDA301BT, TDAi301BT) in the list

0 = Exclude Bluetooth readers (TDA301BT, TDAi301BT)

1 = Include Bluetooth readers

NA_BLE0 (Bit D2, value 0x04) Getting BLE0 reader name (TDA301BL, TDAi301BL, TDA301BLM and TDAi301BLM) in the list

0 = Exclude BLE0 readers

1 = Include BLE0 readers

NA_BLE1 (Bit D3, value 0x08) Getting BLE1 reader name (TDA3901BE) in the list

0 = Exclude BLE1 readers

1 = Include BLE1 readers

NA_SCAN (Bit D4, value 0x10) Getting Bluetooth and BLE readers condition.

0 = Don’t scan Bluetooth readers from the air, get reader name from Android’s phone Paired List. This option does NOT apply for BLE readers.

1 = Scan and find actual operating Bluetooth 3 and BLE readers in the air.

NA_RSVD5 (Bit D5, value 0x20) Reserved for future use, always 0

NA_FIRST (Bit D6, value 0x40) Stop getting reader names condition.

0 = Find and get reader names until time out.

1 = Stop getting reader name after found the first reader. Not suitable for BLE readers.

NA_POPUP (Bit D7, value 0x80) Interaction mode

0 = Silent finding.

1 = Pop up scanning window while finding the readers.

Return Value	0: Success, -1: Error
Callback Listener	**`onGetReaderListNA(ArrayList<String> readerList, int result)`**
Output to Listener	**`readerList`** is an array of Thai ID card reader list. The first found reader will be in **`readerList[0]`**. The second will be in **`readerList[1]`** and so on. The name of each reader is String type. **`result`** is the number of readers found. It will be Return Code of **`getReaderListNA`** if negative, see details in the **Return Code Table**.
Note	If NA_BT or NA_BLE0 or NA_BLE1 (bit D3, D2, D1) are set to 1, then the Location Permission must be allowed. Example of recommended **`listOption`** value: 0x01 (NA_USB) For the application that doesn't want to use Bluetooth reader. 0x03 (NA_USB + NA_BT) For general use case. It will get reader names from the current attached USB reader plus Bluetooth reader name from the phone paired list. This is the fastest way to get reader names. 0xD3 (NA_USB + NA_BT + NA_SCAN + NA_FIRST + NA_POPUP) This option is the best for actual scanning Bluetooth and USB reader. It will immediately return after found the first reader. 0x93 (NA_USB + NA_BT + NA_SCAN + NA_POPUP) This option will scan only the operating **Bluetooth** readers in the air, USB readers and return the completed list of readers to the caller. Users may select any reader that shown on the screen. 0x95 (NA_USB + NA_BLE0 + NA_SCAN + NA_POPUP) This option will scan only the operating **BLE0** readers in the air, USB readers and return the completed list of readers to the caller. Users may select any reader that shown on the screen. 0x97 (NA_USB + NA_BT + NA_BLE0 + NA_SCAN + NA_POPUP) This option will scan all operating **Bluetooth and BLE0** readers in the air, USB readers and return the completed list of readers to the caller. Users may select any reader that shown on the screen. 0x9F (NA_USB + NA_BT + NA_BLE0 + NA_BLE1 + NA_SCAN + NA_POPUP) This option will scan all operating **Bluetooth BLE0 and BLE1** readers in the air, USB

readers and return the completed list of readers to the caller. Users may select any reader that shown on the screen.

6.5 `int selectReaderNA(String reader)`

Description	Select a working reader.
Input Parameter	**`reader`** is the reader name that wants to use.
Return Value	0: Success, -1: Error
Callback Listener	**`onSelectReaderNA(int result)`**
Output to Listener	**`result`** is a Return Code of **`selectReaderNA`**, see details in **Return Code Table**.
Note	The selected Bluetooth 3.0 and BLE0 reader will automatically turn off after selecting another reader.

6.6 `int deselectReaderNA()`

Description	Deselect the current working reader.
Input Parameter	None.
Return Value	Return Code, see details in **Return Code Table**.
Note	The selected Bluetooth and BLE0 reader will automatically turn off after deselecting. You can change the new reader by using **`selectReaderNA`** only, **`deselectReaderNA`** is no need.

6.7 `int connectCardNA()`

Description	Connect the inserted card in the reader.
Input Parameter	None.
Return Value	0 means the inserted card is a Thai national identity card. 1 means the inserted card is another type of card other than a Thai national identity card. Other value is a Return Code, see details in **Return Code Table** .

6.8 `int disconnectCardNA()`

Description	Disconnect the current card and clear memory
Input Parameter	None.
Return Value	Return Code, see details in **Return Code Table**.

6.9 `int getNIDNumberNA()`

Description	Read Thai national ID number (13-digit) from card.
Input Parameter	None.
Return Value	0: Success, -1: Error
Callback Listener	**`onGetNIDNumberNA(String cardData, int result)`**
Output to Listener	**`cardData`** is Thai national ID number.

result is a Return Code of **getNIDNumberNA**, see details in **Return Code Table**.

6.10 int getNIDTextNA()
int getNIDTextNA(int getTextOption)

Description	Read text data from Thai national ID card.
Input Parameter	No input parameter in case of normal text data reading from Thai ID card. Include the **getTextOption** in case of additional AText data needed. 0 = No additional data 1 = Also read the AText follow the normal data
Return Value	0: Success, -1: Error
Callback Listener	**onGetNIDTextNA(String cardData, int result)**
Output to Listener	**result** is a Return Code of **getNIDTextNA**, see details in **Return Code Table**. **cardData** is text data read from Thai national ID card. The data format of **cardData** is: "NID number#Thai title#Thai first name#Thai middle name#Thai last name#English title# English first name#English middle name#English last name#Number#Moo#Trok#Soi# Thanon#Tambon-Khwaeng#Amphoe-Khet#Changwat#Gender#Birth date#Issue place#Issue date#Expiry date#Issue number" In case of **getTextOption = 1** , these additional AText data will be added, follow the Issue number: "#BP1/Request Number#Issuer Code#Structure Version#Card Type#Title Flag"

NID Text Table

	Field	Text size	Note
1	NID number	13 characters	Example: "1234567890123"
2	Thai title	103 characters maximum	
3	Thai first name		
4	Thai middle name		
5	Thai last name		
6	English title	103 characters maximum	
7	English first name		
8	English middle name		
9	English last name		
10	Number	167 characters maximum	
11	Moo		
12	Trok		

	Field	Text size	Note
13	Soi		
14	Thanon		
15	Tambon/Khwaeng		
16	Amphoe/Khet		
17	Changwat		
18	Gender	1 character	“1”=Male, “2”=Female
19	Birth date	8 characters	YYYYMMDD (Buddhist Era) DD = “00” if the day is unknown. MM = “00” if the month is unknown.
20	Issue place	100 characters maximum	
21	Issue date	8 characters	YYYYMMDD (Buddhist Era)
22	Expiry date	8 characters	YYYYMMDD (Buddhist Era) “99999999” for life long card.
23	Issue Number	14 characters	May be not available in some card versions.
Additional AText data when **`getTextOption = 1`**			
24	BP1/Request Number	20 characters	“12345678901/12345678” The first 11 characters are the BP1 Number. The last 8 characters are Request Number. The slash (/) as a separator.
25	Issuer Code	13 characters	“1234567890123”
26	Structure Version	4 characters	The card data structure version, such as “0003” or “0004”.
27	Card Type	2 characters	“01” 01 is Thai ID card.
28	Title Flag	1 character	Space = No title flag “0” = Normal title “1” = Royal Thai navy ranks (ร.น.) “2” = Priest’s ranks

Data Example (No **`getTextOption`**)

“3650800011234#นาย#สมชาย##สามรักสกุล#Mr.#Somchai##Samruksakul#123/45#หมู่ที่ 12####ตำบลบึงพระ#อำเภอเมืองพิษณุโลก#จังหวัดพิษณุโลก#1#25240119#เทศบาลนครพิษณุโลก#25521026#25590118#12341212345678”

Data Example (**`getTextOption = 1`**)

"3650800011234#นาย#สมชาย##สามรักสกุล#Mr.#Somchai##Samruksakul#123/45#หมู่ที่12####ตำบลบึงพระ#อำเภอเมืองพิษณุโลก#จังหวัดพิษณุโลก#1#25240119#เทศบาลนครพิษณุโลก#25521026#25590118#12341212345678**#12345678901/12345678#1234567890123#0004#01#0**"

Note In case of **`getTextOption = 1`** the total items may be changed in the future.

6.11 `int getNIDPhotoNA()`

Description Read portrait photo from Thai national ID card.

Input Parameter None.

Return Value 0: Success, -1: Error

Callback Listener **`onGetNIDPhotoNA(byte[] cardData, int result)`**

Output to Listener **`result`** is a Return Code of **`getNIDPhotoNA`**, see details in **Return Code Table**.

`cardData` is photo image data read from Thai national ID card. A readable image is a byte array of 5120 bytes in JPEG format. The image size is 297x355 pixels or 148x178 pixels.

6.12 `int getCardStatusNA()`

Description Get card status of the working reader is absent or present.

Input Parameter None.

Return Value 1: Card present (card inserted).

-16: Card absent (card removed)

Other value is a Return Code, see details in **Return Code Table**.

6.13 `int getRidNA(byte[] rid)`

Description Get unique Reader ID from the reader.

Input Parameter **`rid`** is a 256-byte array, free allocated space.

Output Parameter **`rid`** is binary data of Reader ID.

Return Value Positive value is the number of bytes of Reader ID data.

Negative value is a Return Code, see details in **Return Code Table**.

Note The reader must be selected by **`selectReaderNA`** before using this command.

6.14 `int getSoftwareInfoNA(String[] softwareInfo)`

Description Get NALib library information.

Input Parameter None.

Output Parameter **`softwareInfo[0]`** is a string of library information. Each field is separated by a # character in this way:

"Library name and version#Copyright Information", for the example "NALib 0.0.1 #Copyright R&D Computer System Co., Ltd."

Return Value Return Code, see details in **Return Code Table**.

6.15 `int getLicenseInfoNA(String[] licenseInfo)`

Description	Get license file information.
Input Parameter	None.
Output Parameter	**`licenseInfo[0]`** is a string of license information. Each field is separated by a # character in this way: "License Filename#License File Date#VL Info", for the example "/storage/NASample/rdnidlib.dls#2019-12-10#V|R1.02|AR02"
Return Value	Return Code, see details in **Return Code Table**.

6.16 `int updateLicenseFileNA()`

Description	Update license file from Internet, use path file location specified by **`openLibNA`** command.
Input Parameter	None.
Return Value	0: Success, -1: Error
Callback Listener	**`onUpdateLicenseFileNA(int result)`**
Output to Listener	**`result`** is a return value of **`updateLicenseFileNA`**. If the number is 0 or positive, it means that the work was successful with the details as follows: 0, 1, 2 or 3 means the new license file has been successfully updated from the server 0, 1, 2 or 3. 100, 101, 102 or 103 means the latest license file has already been installed, updating is no need. Negative value indicates that it failed. For the examples: -15: Internet Error -18: License Update Error
Note	Call this function when the connected reader has not been registered or got error code regarding license file problems, such as Return Code = -12 or -2. Before calling this function, Android devices should be enabled the Internet connection first, otherwise, this function will return -15. This function is no need for Virtual Licensing (VL) readers which didn't use the license file.

6.17 `int setPermissionsNA(int pms)`

Description	Set permissions for NALib.
Input Parameter	**`pms`** 0: Disable USB reader in-app permission (default). 1: Enable USB reader in-app permission (**In-App**). -1: Get current permissions state.
Return Value	0: USB reader in-app permission is disabled

1: USB reader in-app permission is enabled

Negative value is a Return Code, see details in **Return Code Table**.

Note — The default value of the USB reader in-app permission is 0 or disabled. If set to 1, there will be a pop-up window asking for USB reader using permission inside the application (**In-App**).

6.18 `int getReaderInfoNA(String[] readerInfo)`

Description — Get selected reader information.

Input Parameter — None.

Output Parameter — **`raederInfo[0]`** is a string of reader information. Each field is separated by a # character in this way:

"Interface#License Type#Model#Firmware Version"

Interface	USB: USB reader.
	BLT: Bluetooth reader.
	BLE: Bluetooth Low Energy (BLE) reader
License Type	VL: Virtual Licensing reader.
	FL: File Licensing reader.
	NL: Non-License reader.
Model	Model of the reader.
Firmware Version	Reader's firmware version

For the example "USB#VL#TDAi301VC#8.00" .

Return Value — Return Code, see details in **Return Code Table**.

Note — The reader must be selected by **`selectReaderNA`** before using this command.

6.19 `int getEmvTextCA()`

Description — Read text data from a bank card.

Input Parameter — None.

Return Value — 0: Success, -1: Error

Callback Listener — **`onGetEmvTextCA(String textData, int result)`**

Output to Listener — **`result`** is a Return Code of **`getEmvTextCA`**, see details in **Return Code Table**.

`textData` is text data read from bank card.

The data format of **`textData`** is:

"PAN#Holder Name#Expiry Date#Network"

Bank Card Text Table

	Field	Text size	Note
1	Primary Account Number (PAN)	13-19 characters	Example "1234567890123456"

	Field	Text size	Note
2	Holder Name	0-26 characters	If the first name and last name are separated, the format used is “Last Name/First name”. For example, “RAKSAKUL/SOMCHAI”.
3	Expiry Date	4 characters	The format is YYMM. YY is the last two digits of the year (in the Gregorian calendar). MM is two digits of the month number. For example, “2912” means December 2029.
4	Network	Variable length	For example “Visa”.

Example “1234567890123456#RAKSAKUL/SOMCHAI#2912#Visa”

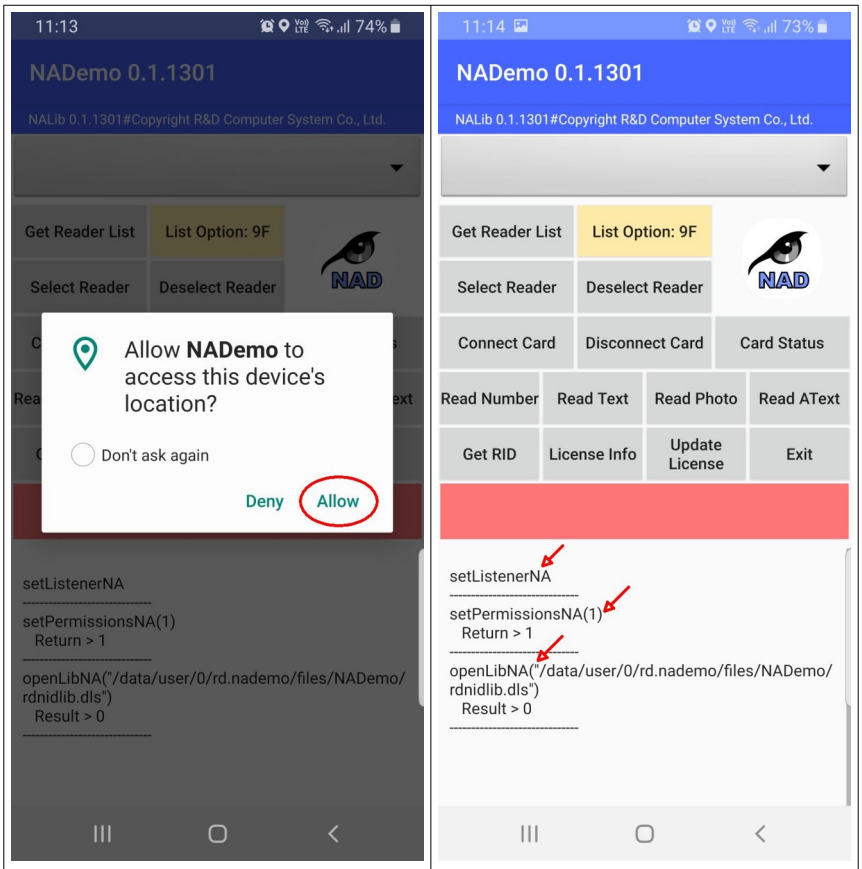
Note This command is only applicable to product models that support bank card reading, such as products in the TDC series.

6.20 Return Code Table

Return Code	Name	Descriptions
0	NA_SUCCESS	All done, no error.
-1	NA_INTERNAL_ERROR	Found internal error.
-2	NA_INVALID_LICENSE	Invalid license. Can not use this reader.
-3	NA_READER_NOT_FOUND	Smart card reader not found.
-4	NA_CONNECTION_ERROR	Card connection error.
-5	NA_GET_PHOTO_ERROR	Can't get photo data from card.
-6	NA_GET_TEXT_ERROR	Can't get text data from card.
-7	NA_INVALID_CARD	Invalid card, not supported card.
-8	NA_UNKNOWN_CARD_VERSION	Unknown Thai ID card version.
-9	NA_DISCONNECTION_ERROR	Can't disconnect the reader.
-10	NA_INIT_ERROR	Initialization error or **openLibNA** didn't call yet.
-11	NA_READER_NOT_SUPPORTED	Doesn't support this reader.
-12	NA_LICENSE_FILE_ERROR	License file error.
-13	NA_PARAMETERS_ERROR	Parameters error.
-15	NA_INTERNET_ERROR	Internet connection error.
-16	NA_CARD_NOT_FOUND	Card not found.
-17	NA_BLUETOOTH_DISABLED	Bluetooth is turned off.
-18	NA_LICENSE_UPDATE_ERROR	Can't update license file.
-19	NA_READER_SELECTION_ERROR	Reader selection error.
-20	NA_FUNCTION_NOT_SUPPORTED	Function not supported
-31	NA_STORAGE_PERMISSION_ERROR	Storage permission is error or denied.
-32	NA_LOCATION_PERMISSION_ERROR	(Bluetooth) Location permission is error or denied.
-33	NA_BLUETOOTH_PERMISSION_ERROR	Bluetooth permission is error or denied.
-41	NA_LOCATION_SERVICE_ERROR	Location Service is error or turned off.

7. NADemo Usage

- NADemo is a sample application for NALib. It's a tool for learning and testing of NALib API.
- There is the NADemo.apk file in the SDK package. Developers can install and run it immediately from an Android phone.
- Source code of NADemo is also provided in the SDK package.
- NADemo using steps:
  - At the first running NADemo should ask for location permission, choose **ALLOW** the permission for enabling Bluetooth reader usage. Error code -32 will return when using Bluetooth functions if you **DENY**.
  - At the bottom of the screen, it's a result message area. You can see result of each command here.

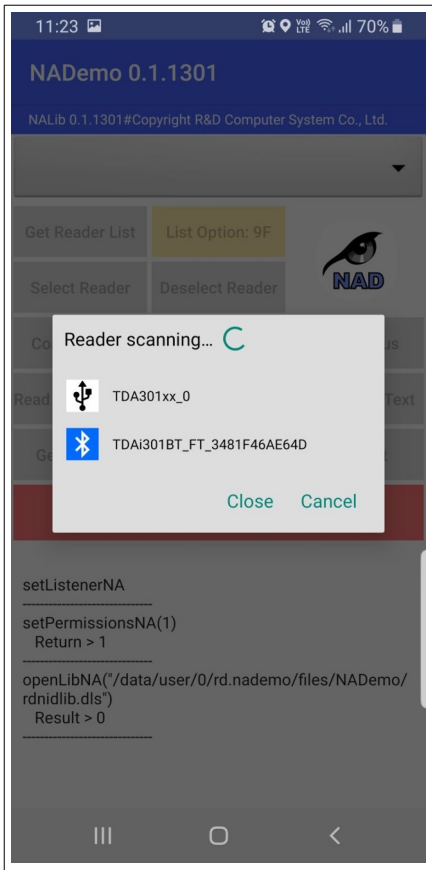


Location permission asking | Start up screen of NADemo

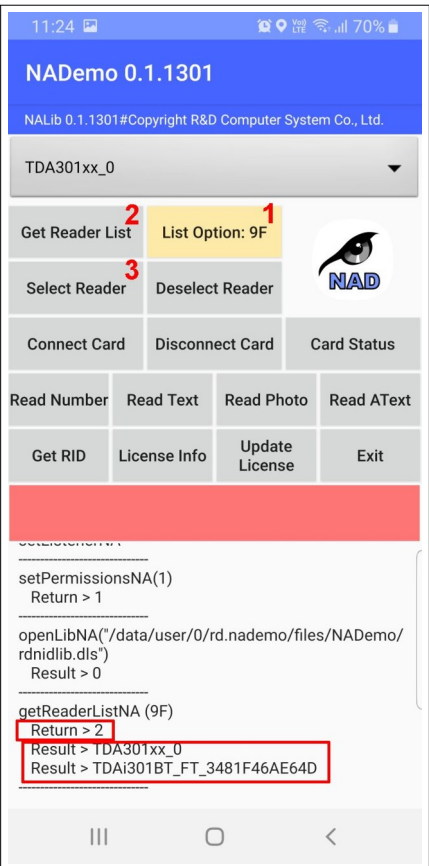
- After start up the message should show about the result of three internal commands, **`setListenerNA`**, **`setPermissionsNA(1)`** and **`openLibNA`** and the result of **`openLibNA`** should be zero (0). If it's not zero, such as -31 or -12, that means NADemo can't open and use the library. You can see the meaning of the result number in the **Return Code Table**.
- Command **`setPermissionsNA(1)`** will make NADemo use USB Reader In-App Permission (NADemo didn't use USB Reader Out-App).
- Prepare the reader.

- Case of USB reader, please plug the USB reader into a mobile phone that supports OTG.
  - If pop-up window appears and ask for application selection for USB device, tab **Back** or ignore it.
- Case of BLE reader, press the power button on the reader, wait until the blue light flashes. Then turn on the Bluetooth of the mobile phone. **Don't pair the reader.**
- Case of Bluetooth 3.0 reader, press the silver power button on the reader, wait until the blue light flashes. Then turn on the Bluetooth of the mobile phone. Pair the reader by accessing the Android system's **Settings** > **Bluetooth** > **Scan** and looking at the bottom line of the screen. Select a reader that begins with "FT_".

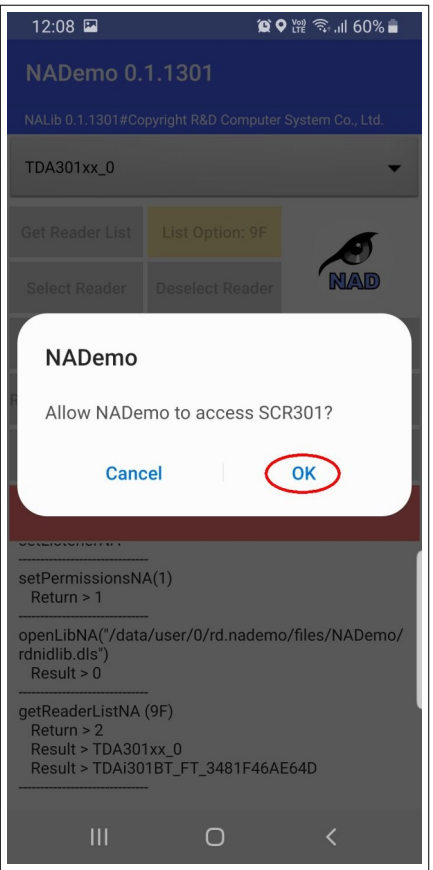
- At NADemo screen, tap the List Option button (yellow button) (1) to set the List Option value to 93 if use USB and Bluetooth reader or 9F if use USB + Bluetooth + BLE reader and then tap the Get Reader List button (2). The reader search screen will appear. Select a reader that you want to use. Or tap the OK button. If the reader is still not found, tap Re-scan to search again.
- Look at the result area. The return value is the number of readers found. (How many devices are found). Results value are the name of the readers that were found.



Get Reader List



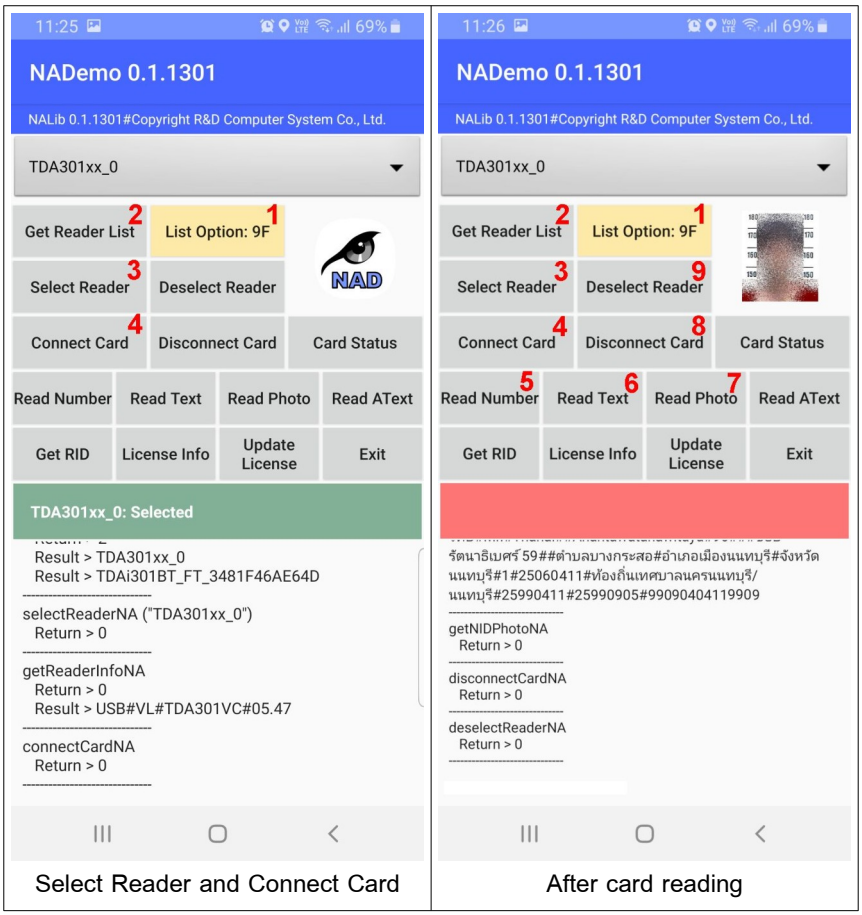
After reader list getting



In-App pop-up from Select Reader command, in case of USB reader.

- Tab the triangle arrow of the drop-down box at the top of the screen, the application will display the name of the reader received from the **`getReaderListNA`** command. Then select the reader you want to use. Usually, choose the first reader.

- Tap on the Select Reader button (3) to send **`selectReaderNA`** command with the name of reader selected in the drop-down box and view the results.
- If USB reader is selected, the app will ask for permission to access the reader, e.g. “Allow NADemo to access XXXX?” or “Allow NADemo to access XXXX?” where XXXX is the name of the system reader. (Which may not match the model name of the actual product, please ignore it) then tab OK.
- Insert Thai ID card into the reader. Then tap the Connect Card button (4) and see the results. The return code should be 0.
- Tap the Read Number button (5) and view the results.
- Tap on the Read Text button (6) and view the results.
- Tap on the Read Photo button (7) and view the results.



- After reading all data from the card, if you want to read a new card, you have to stop using the old card by tapping the Disconnect Card button then pulling the old card out and insert a new card. Tap the Connect Card button and then read that card with the Read Number, Read Text or Read Photo button as needed.
- If you want to use another reader, tap the Disconnect Card and Deselect Reader button respectively, then select a new reader from the drop-down box or Get Reader List again.
- Additional testing suggestions

- List Option values that are recommended to be tested are:
  - 93 will scan for the USB reader that is plugged in and all Bluetooth readers in the air until the timer expires. The scanning window will display and the user can choose the reader he/she wants.
  - D3 will scan for the USB reader that is plugged in and all Bluetooth reader in the air and close the window immediately after the first reader is found. If plugging in a USB reader, NADemo will immediately select a USB reader. If did not plug USB reader, it will wait for the first Bluetooth reader found.
  - 01 for projects that use only USB reader and don't want to use a Bluetooth reader at all. This option will work fastest.
  - 94 is suitable for the project that will use BLE0 reader only, no USB/Bluetooth readers.
  - 98 is for the project that allow users to use BLE1 (TDA3901BE) only.
  - 9F is for the project that allow users to use USB, Bluetooth BLE0 and BLE1, all type of readers.
- Testing the same USB reader with multiple applications
  - For In-App Permission mode, all applications can use the same reader without problem.
  - For Out-App Permission mode, one USB reader can support only one application. If you want to use this reader with another application, you have to pull the USB reader out, plug it in again and choose the correct new application name that you want.

8. NASample And Source Code For Application Development

NASample is a simple application with source code for developers to use as a template. There are source code and APK files in NALib SDK. The developers can try in Android Studio immediately.

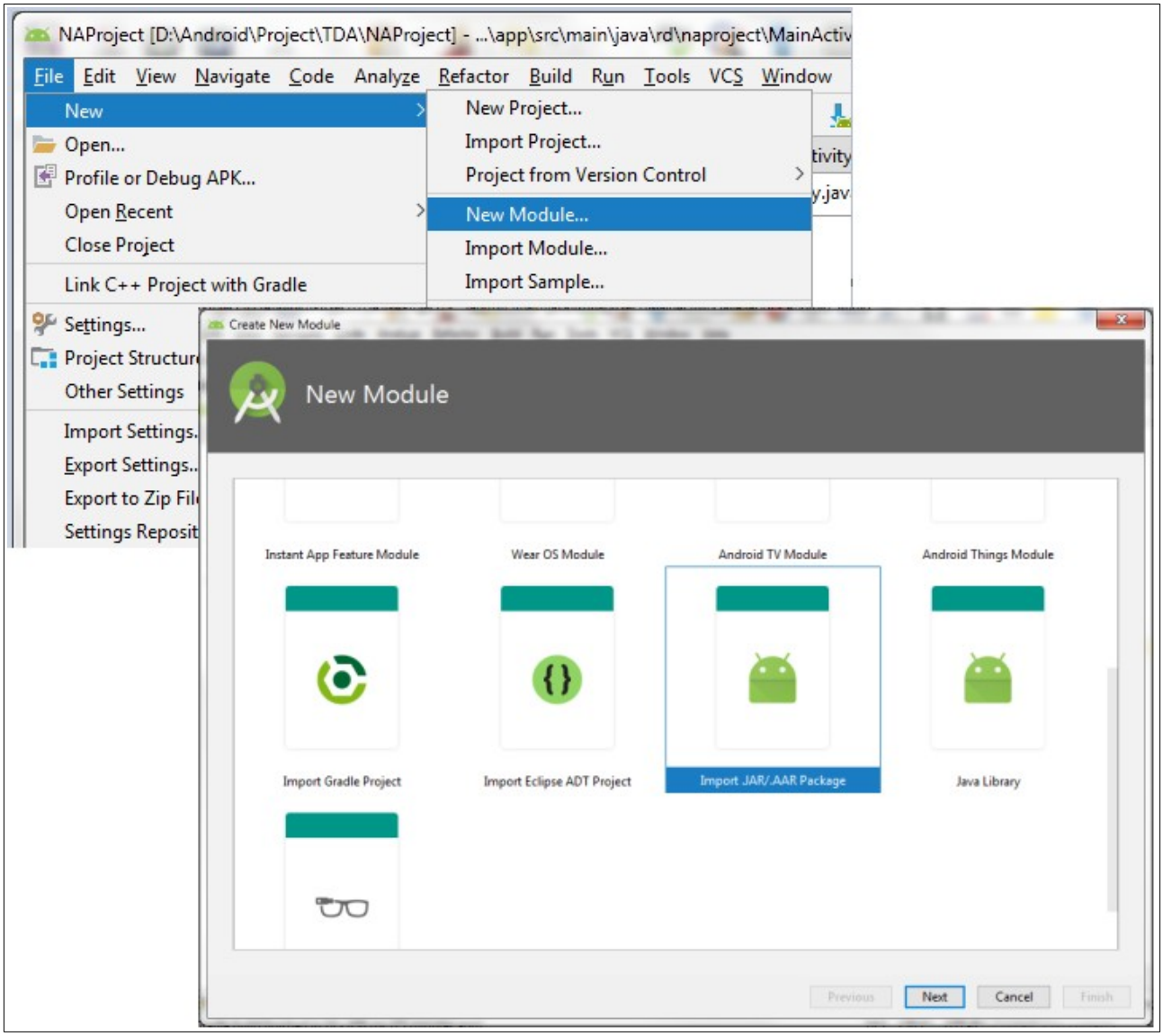
Examples of development steps and programming instructions can be found in the NASample source code. Developers can try by these steps:

8.1 Import NALib.aar

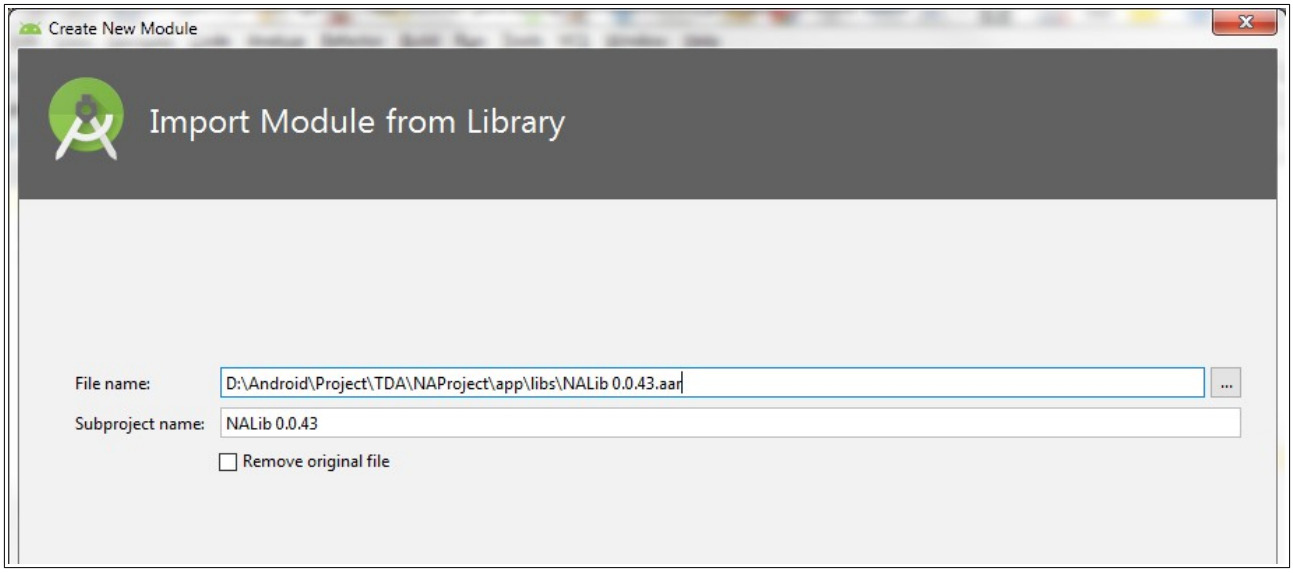
8.1.1 Create a new project in Android Studio, which should be V3.2 and up.

8.1.2 Files > New Module…

8.1.3 Import .JAR/.AAR Package, then Next



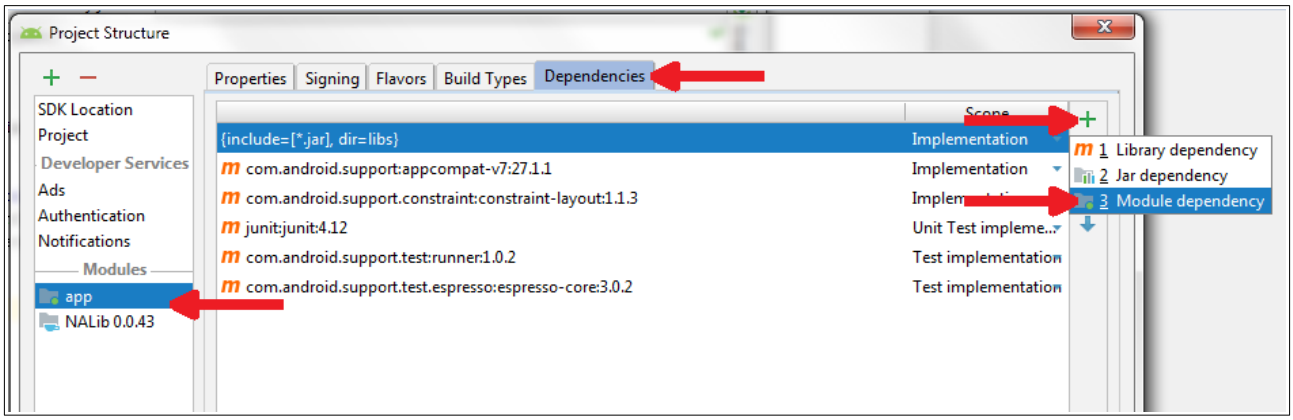
8.1.4 Specify file location of NALib.aar in File Name: then click Finish.



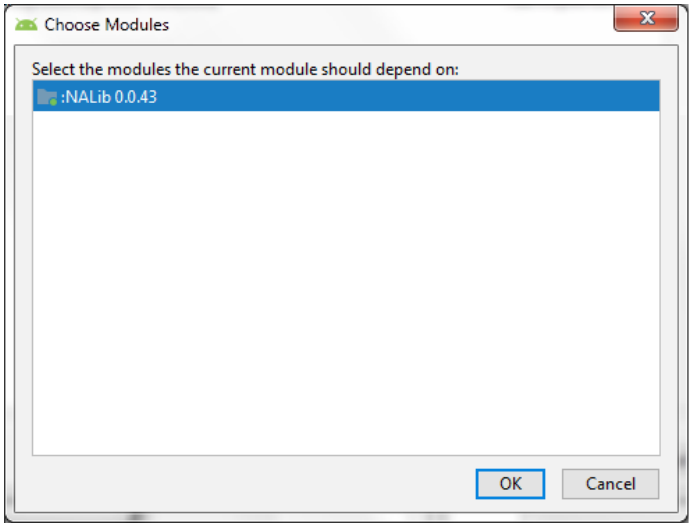
8.1.5 Click File > Project Structure…

8.1.6 Select the Main Module (app) then click the Dependencies tab.

8.1.7 Click at Plus (+) icon on the right and select 3 Module dependency.



8.1.8 Select the NALib module and click OK.

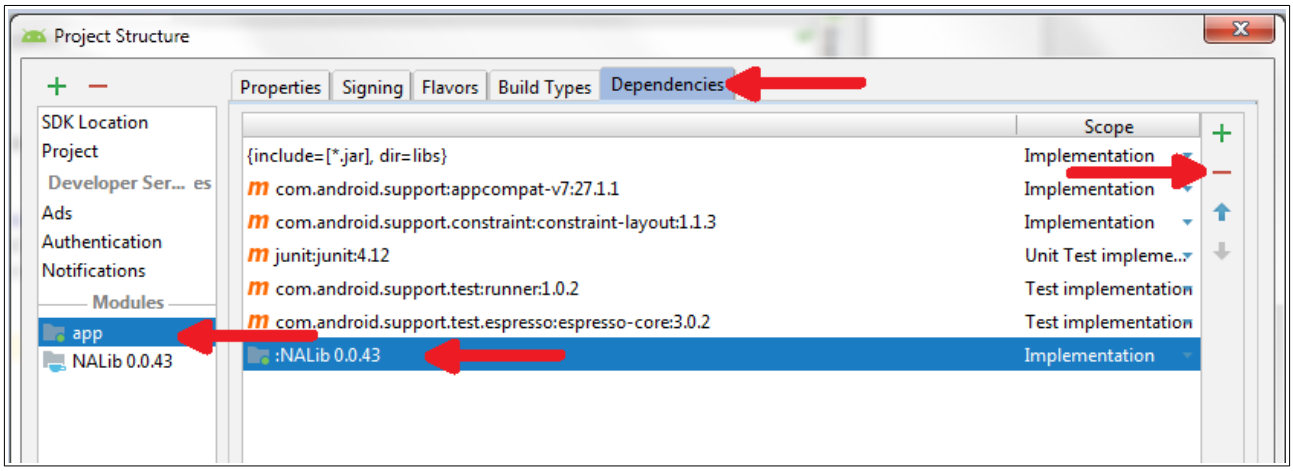


8.2 NALib.aar removing

8.2.1 File > Project Structure…

8.2.2 Select the Main Module (app) then click the Dependencies tab.

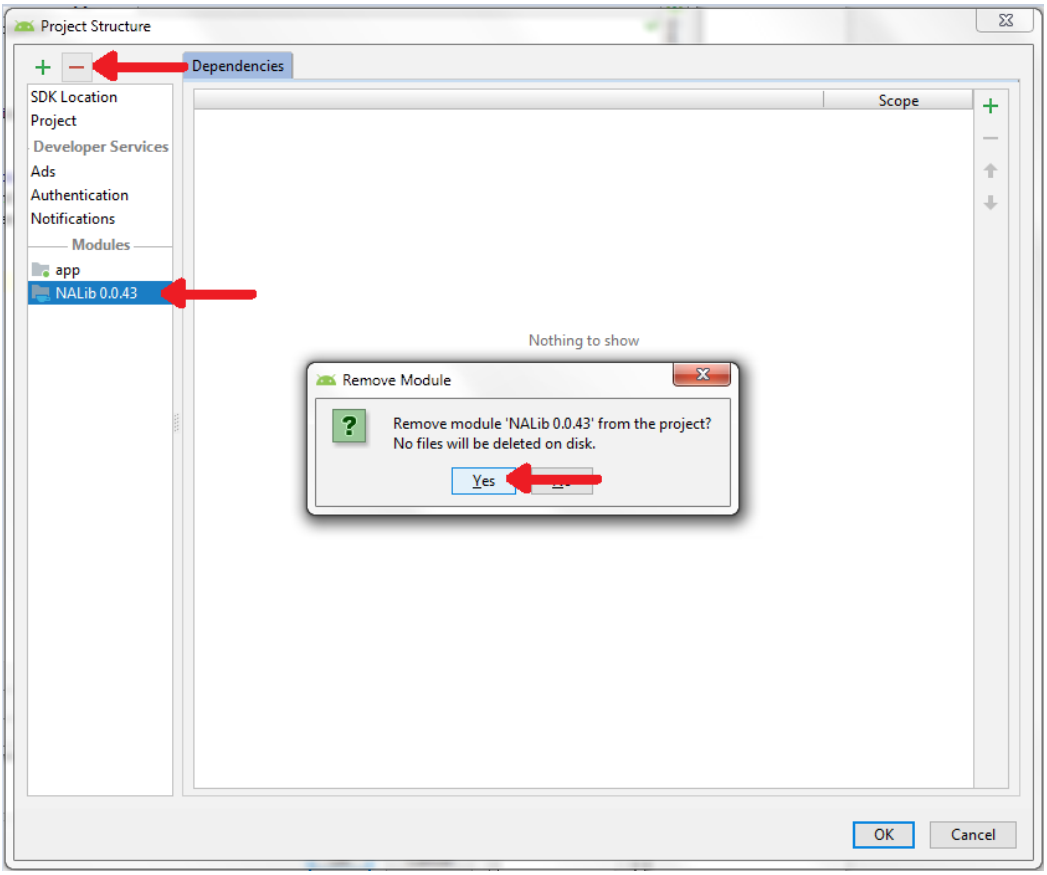
8.2.3 Select NALib module then click at Minus icon (-) on the right and OK



8.2.4 File > Project Structure…

8.2.5 Select NALib then click Minus icon (-) on the top.

8.2.6 Click Yes and OK.



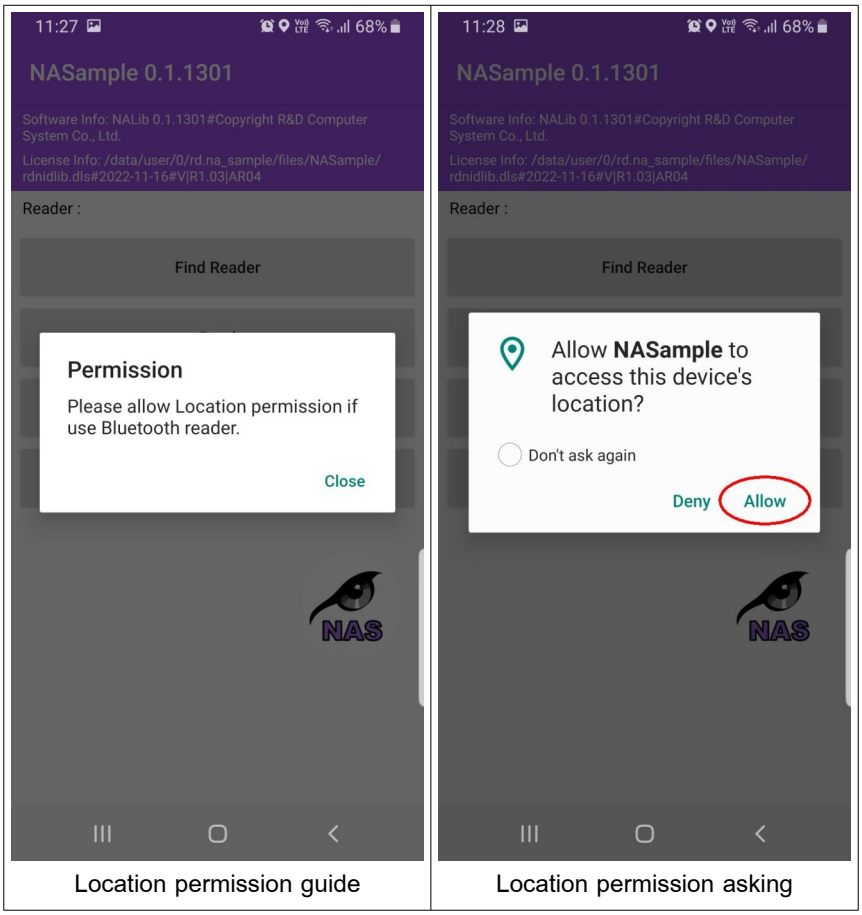
8.3 Thai ID and bank card reading steps

- **`setListenerNA`**: Bind interface with ResponseListener.
- **`setPermissionsNA(1)`**:If you want a pop-up window for USB Reader In-App Permission.
- **`openLibNA`**: Start using NALib with FL+VL mode or VL only mode.
  - Check the result from **`onOpenLibNA.`**
- **`getReaderListNA`**: Ask for reader list.
  - Get reader list from **`onGetReaderListNA.`**
- **`selectReaderNA`**: Select a reader for using.
  - Check the result from **`onSelectReaderNA.`**
- **`connectCardNA`**: Connect and start using Thai ID or bank card.
- Thai ID card reading:
  - **`getNIDTextNA`**: Read text data from Thai ID card.
    - Get text data via **`onGetNIDTextNA.`**
  - **`getNIDPhotoNA`**: Read Thai ID photo.
    - Get photo data via **`onGetNIDPhotoNA.`**
- Bank card reading:
  - **`getEmvTextCA`**: Read text data from bank card.
    - **`Get text data via onGetEmvTextCA.`**
- **`disconnectCardNA`**: Stop card connection.
- **`deselectReaderNA`**: Stop using the reader.
- **`closeLibNA`**: Stop using NALib.

8.4 NASample Operation Description

8.4.1 Application starting

- NASample will ask for location permission. Please choose **ALLOW** If you want to test a Bluetooth reader. Choosing **DENY** will disable all Bluetooth functions. If you denied this permission, you can test USB readers only.

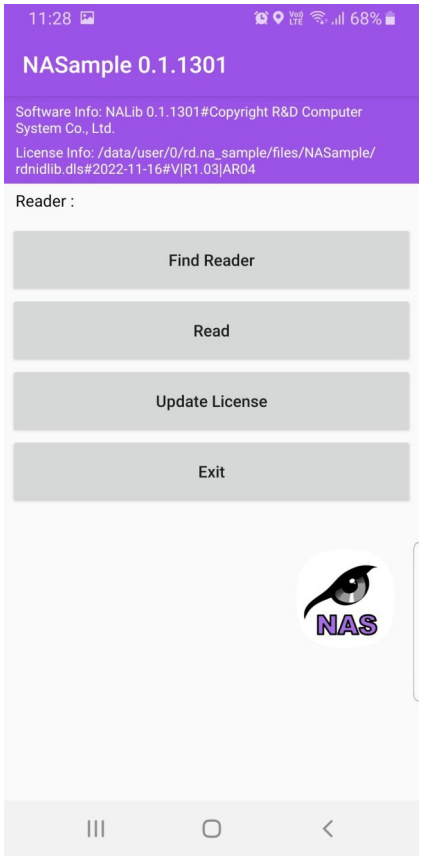


Location permission guide | Location permission asking

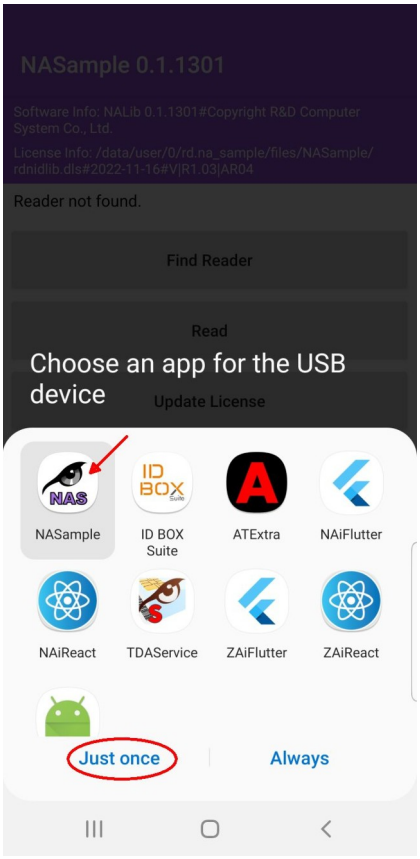
- Use **`setListenerNA`** to bind callback interface with ResponseListener.
- Use **`setPermissionsNA(1)`** to make NASample use USB Reader In-App Permission (NASample uses both USB Reader Out-App and In-App permissions).
- Start using NALib by using **`openLibNA(licenseFilename)`**. For NASample, the license file will be used to support testing with both FL and VL readers.

8.4.2 Preparing the reader for NASample testing

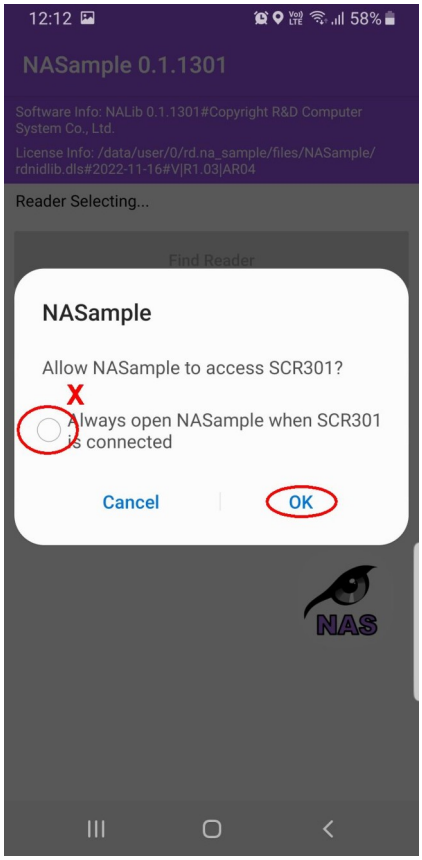
- If testing with a USB reader, plug the USB reader into an Android device that supports OTG. Wait a moment and a pop-up screen will show to choose which application the USB reader should be used for. Select at NASample (or select nothing).
  - If no screen shows, it's possible that that mobile phone not compatible with OTG. Switch to another OTG compatible mobile phone instead.
  - NASample will ask for In-App permission if users denied Out-App permission.



Startup screen of NASample



Out-App Permission pop-up after USB reader plugging. Please select NASample and tab “Just once”, not “Always”.



In-App Permission pop-up if Out-App Permission has been denied. Please don’t enable “Always open...”

- If testing with a BLE reader (e.g. TDA301BL, TDA301BLM, TDA3901BE), press the power button on the reader, wait until the blue light flashes. Then turn on the Bluetooth of the mobile phone. **Don’t pair the reader.**
- If testing with a Bluetooth reader, press the power button on the reader, wait until the blue light or green light flashes. Then turn on the Bluetooth signal of the mobile phone. Pair the phone with the reader by using the Android system's **Settings** > **Bluetooth** > **Scan** and looking at the bottom line of the screen. Select a reader name that begins with "FT_" and confirm.

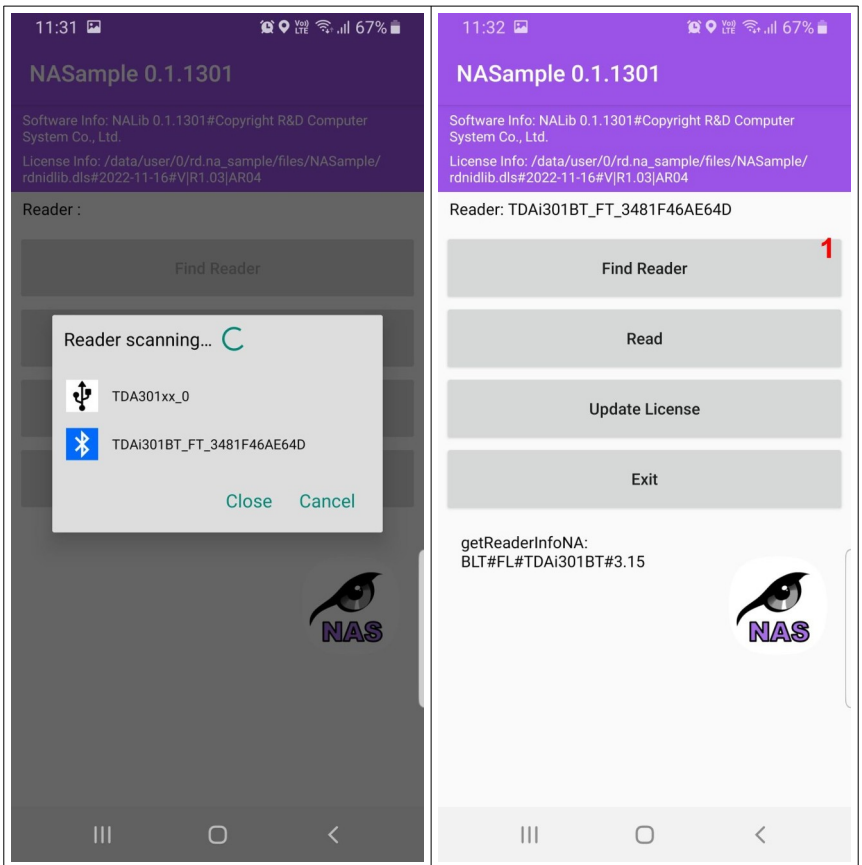
8.4.3 Reader finding

- The **Find Reader** button in NASample application is for finding or changing the new readers
- When the user touches the **Find Reader** button, use the command **`getReaderListNA(listOption)`** to request a list of plugged-in USB readers and paired Bluetooth readers. This readers finding method can be determined from

`listOption` parameter. Normally, the value is recommended to be 0x93, 0x9F or 0x01, but for NASample will use 0x9F, which means that the application will always scan all reader types.

- In case the user does not want to use the Bluetooth reader and has denied the request for Location Permission, NASample will use the value 0x81 to work instead. This will reduce the waiting time for the Bluetooth reader scanning.

- Get a list of the reader via **`onGetReaderListNA`**.
- Call **`selectReaderNA(reader)`** using the **`reader`** name of the first reader that is obtained from **`getReaderListNA`** (you may create drop-down for user to select the reader, but NASample doesn't use this method).



Reader finding | Found and select TDAi310BT

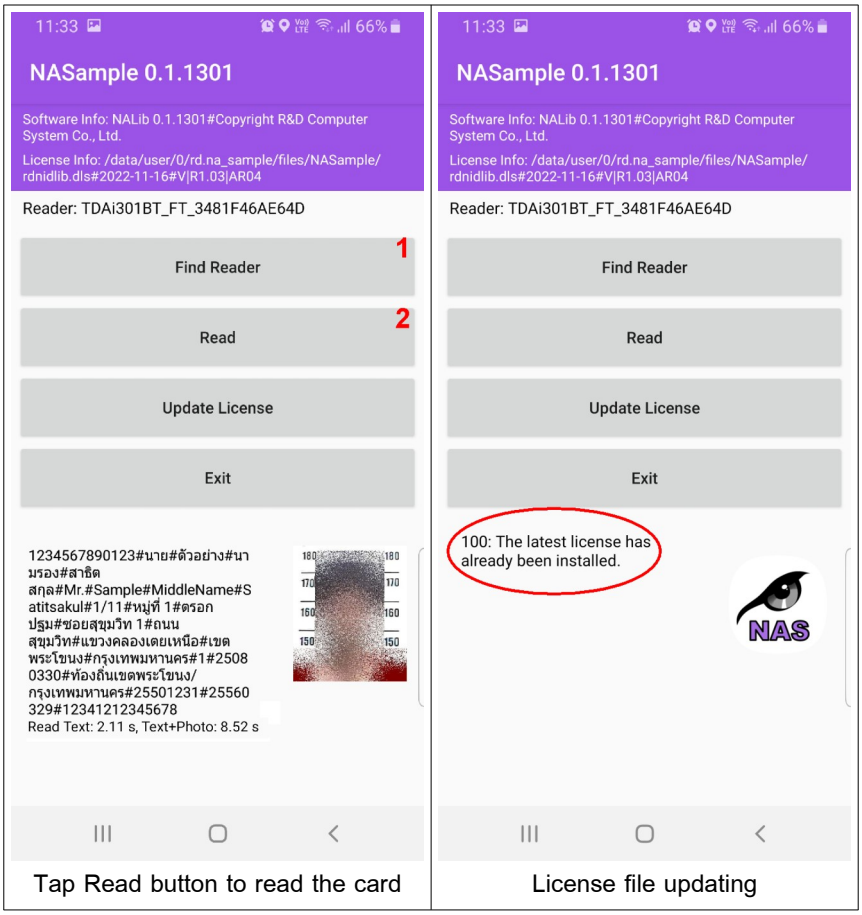
- Check the Return Code from **`onSelectReaderNA`**. If it is zero, go to next step. If it is -2 or -12, notify the user of the problem or update the license file and then call **`selectReaderNA`** again.
- Check reader type with **`getReaderInfoNA`** function if you want.

8.4.4 Thai ID card data reading

- When the user wants to read the ID card, call **`connectCardNA`** to supply power to the ID card. And start connecting with the card. If able to connect, will return the

Return Code to 0. If other values such as -16 (card has not been inserted into the reader) should inform the user of those problems.

- Use the command **`getNIDNumberNA`**, **`getNIDTextNA`** or **`getNIDPhotoNA`** to read data from Thai ID card as needed.
- After that disconnect the Thai ID card by using **`disconnectCardNA`**.



8.4.5 Updating the license file for File Licensing readers

- Use **`updateLicenseFileNA`** to download the latest version of the license file from the Internet and save as the name specified in the **`openLibNA`** function. After completion, check the Return Code from **`onUpdateLicenseFileNA`**. If the value is 0, 1, 2 or 3 that means the NALib has updated the license file to a new file. If the value is 100, 101, 102 or 103 that means the license file is already up to date. If the value is -15, it cannot be downloaded because you are unable to access the Internet or if -18, it means you cannot update because of other reasons. In these cases, -15 and -18, may require the user to enable the mobile Internet connection and try the update again.

8.4.6 Changing the reader

- User can change the reader in 2 ways, by using the previous list or calling **`getReaderListNA`** to get a new list.

- For NASample, we use **`getReaderListNA`** function through the **Find Reader** button to scan new readers.

8.4.7 When exiting the application

- Deselect the reader by **`deselectReaderNA`**.
- Always call **`closeLibNA`** to stop NALib.
- Then close the application.

9. DLS License File Management For File Licensing Readers

The license file is the most important for using NALib SDK with File Licensing (FL) readers. This file is a collection of registered and licensed readers for NALib.

The license file is no need for Virtual Licensing (VL) readers. If you want to use VL readers **only** you can skip and ignore this chapter.

The license file contains a list of readers registered for use with the NALib SDK. Therefore, even if a reader looks the same and was purchased from R&D Computer Systems, if it is an unregistered device, it will not work with the NALib SDK. Registered readers are only sold in product packages with model names beginning with TDA, TDAi, or TDC.

- HawkEye NALib license filename is rdnidlib.dls.
- In the NADemo application, the license information can be got by **`getLicenseInfoNA`** through the **License Info** button which displays the folder name, file name and date of the file.
- We can also look at the license file information in the NASample application.
- All registered readers S/N will be saved inside this license file.
- Only registered readers can be used with NALib.
- The new license file will always bigger than the old one.
- For the first run of the developed application, the developer should download the new license file from the Internet with **`updateLicenseFileNA`** command.
- In case of application software found unregistered reader from **`selectReaderNA`** command, please try to update the new license file from the Internet with **`updateLicenseFileNA`** command or ask the user to replace it with a new registered reader.
- The application should be programmed to load and update license files in a variety of ways for ease of application using.
  - After installation, we should update the new license file from the Internet immediately.
  - The application should update the new license file every time get a return code of "-2" or "-12".
  - Always put an "Update License" command button on an application screen. Then users can solve the problem by updating the license file manually by themselves.
- Updating the license file requires the Internet connection. Please ask the user to make Internet connection when found "-15" or "-18" return code from **`updateLicenseFileNA`** command.

10. Applications Development For Virtual Licensing Readers

The Virtual Licensing or VL is a new generation of Thai ID licensing. Readers with VL are no longer need license file as normal File Licensing readers. The benefits of VL are:

- No programming required to manage the license file. For example, do not need to create an Update button and do not need to prepare downloading.
- No Internet usage.
- Users no longer have to update the license file.
- Users can buy more readers immediately. Do not wait for license file updating.
- Android's Internet permission and External Storage permission are not required.
- It can be mixed-used with FL - File Licensing readers in the same application.

These are the supported VL readers of NALib 0.1.12 and up:

- TDA301VC, TDAi301VC
- TDA39VC, TDA39VC2, TDA39VM
- TDA301BL, TDAi301BL
- TDA301BLM, TDAi301BLM
- TDC39VC, TDC301VC

The guidelines for VL readers application development as follows.

- VL readers only supported by NALib.aar version 0.1.1302 or later.
- In the beginning process **`openLibNA(licenseFile)`**, set **`licenseFile`** to `""` (zero-length string). Or if you want to use it mixed with normal File Licensing readers, you can set it as the name of the license file.
- No need to create an Update License button and **`updateLicenseFileNA`** is no longer used except for normal FL readers mixed-used.
- No need to ask for permissions regarding Internet and storage from Android, unless mixed-used with normal FL readers.

11. App Permissions Requesting

NALib may need to request the following permissions to work.

- **Group 1: Bluetooth Permissions**
  - Note: If not using the Bluetooth reader in the application at all and use the USB readers only, you don't need to ask for group 1 permissions.
  - Permissions in this group consist of:
    - General Permissions
      (1) android.permission.BLUETOOTH
      (2) android.permission.BLUETOOTH_ADMIN
    - Permission for Android 11 and below
      (3) **android.permission.ACCESS_FINE_LOCATION** (Runtime)
    - Permissions for Android 12 and higher
      (4) **android.permission.BLUETOOTH_SCAN** (Runtime)
      (5) **android.permission.BLUETOOTH_CONNECT** (Runtime)

- **Group 2: Internet Permissions**
  - Note: These permissions are no need if you didn't use **updateLicenseFileNA** command.
  - Permissions in this group consist of:
    (6) android.permission.INTERNET
    (7) android.permission.ACCESS_NETWORK_STATE

- **Group 3: External Storage Permissions for license file reading and writing**
  - Case 1: Application supports VL readers only, the Storage Permission is **no need**.
  - Case 2: Application supports both VL and FL readers but save license file in application's **Private Storage** , the Storage Permission is **no need**. The NASample and NADemo use this approach.
  - Case 3: Application supports both VL and FL readers and save license file in **External Storage** , the Storage Permission is **required**.
    - Permissions for Android 10 and below:
      (8) **android.permission.READ_EXTERNAL_STORAGE** (Runtime)
      (9) **android.permission.WRITE_EXTERNAL_STORAGE** (Runtime)
    - Permission for Android 11 and higher:
      (10) **android.permission.MANAGE_EXTERNAL_STORAGE** (Runtime)

How to request the permissions

- Request all permissions used by specifying in the file AndroidManifest.xml.

- For permission (3), (4), (5), (8) and (9), which are runtime permissions, developers must write more code for permission checking and requesting consent from users at the time of use (runtime).
  - Because the permissions (8) and (9) belong to the same group, so in requesting the permission while runtime, request only one of them (8) or (9) is enough. The system will allow both permissions at the same time.
  - Therefore, requesting only (3), (4), (5) and (8) is enough in runtime.

Please preview the AndroidManifest.xml file and programming examples from the source code of the NASample or NADemo.

- For Virtual Licensing readers, the license file will not be used, so there is no need to request permission for group 2 (Internet) and group 3 (External Storage) and if it is a USB reader, then no permission is required at all.
- For group 3 (External Storage) permissions, they are need only when you want to save the license file in the External Storage. If you choose to save it in Private Storage, please don't ask for the permissions.
  - For ExternalStorage Permissions
    - Android 10 or below, please ask for (8) and (9) and add **`android:requestLegacyExternalStorage="true"`** in the **Application** tag.
    - Android 11 and higher, please ask for the permission (10).
  - The best practice is saving the license file in the Private Storage, all permissions will not need.

12. Card Readers Connecting Conditions of NALib

- The NALib is compatible with both USB and Bluetooth readers, but can only be used with one reader at a time via the **`selectReaderNA`** function.
- USB card readers should not be plugged into a mobile phone more than one device. Please don't use a USB hub because NALib may not work correctly.
- Bluetooth 3.0 readers that will be used with NALib must only be paired with the mobile phone. If not already paired, readers will not found while scanning.
- Please **Don’t Pair** a BLE reader, users can use it without pairing.

13. Pop-up Window for Requesting USB Reader Permission Inside and Outside the Application

- Usual for using a USB reader with Android application, the user must first consent or give permission. Then the application can use that reader. The request will appear in a pop-up window which can be displayed in 2 points which are:
  - Pop-up to ask immediately after plugging in the USB reader, called USB Reader Out-App Permission.
  - Pop-up to ask when the application is in use, called USB Reader In-App Permission.
- NALib supports both methods. Select and use only one or select both is possible.
- In case of using Out-App method only:
  - Advantages
    - Users will immediately know when the USB reader is plugged.
  - Disadvantages
    - If there are multiple applications that can use the same, single USB reader. The problem may be found when the wrong application is chosen while pop-up with Always option. These will prevent the pop-up from being displayed again until the user clears the default setting of the wrong application.
    - If you want to change or switch between multiple applications with the same reader, the reader must be unplugged and then plugged back again to select a new application in the pop-up window.
- In case of using In-App method only:
  - Advantages
    - There will always be a pop-up window within our application. Although user accidentally select Always option in other applications.
    - Easily switch between multiple applications with the same reader. No need to unplug the reader. In this case, all apps must use In-App permission.
  - Disadvantages
    - Not suitable for use with a kiosk. Because no one will respond to the pop-up window.
    - If there are other apps selected using the Out-App method with the same reader, the name of the app using the In-App permission will not appear in the pop-up window. Then users can't find our app.
- Using both methods Out-App and In-App together:
  - Advantages
    - When plugging in the USB reader, a pop-up window will appear immediately and will have our app name. If users choose correctly, they can continue to use. If users choose incorrectly, there will be pop-up in our app to choose again.
    - Easily switch between multiple applications with the same reader. No need to unplug the reader. In this case, all apps must use In-App permission.

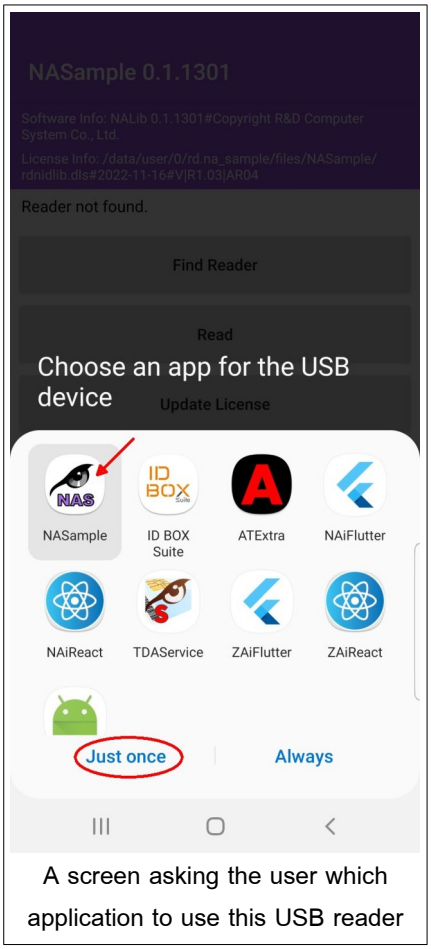
- How to set:
  - For USB Reader Out-App permission, developers must specify the activity **`rd.nalib.UsbEventReceiverActivity`** in the file AndroidManifest.xml. See the examples in the NADemo and NASample projects.
  - For USB Reader In-App permission, developers must call **`setPermissionsNA(1)`** during the first part of the application's operation. See an example from the source code of NASample.
  - Developers can try using Out-App and In-App from our sample applications.
    - NADemo is an example application that only uses In-App permission.
    - NASample is an example application that uses both Out-App and In-App permissions.
- It is recommended to develop by using both Out-App and In-App together.

14. Card Readers Usage (End Users should be notified.)

14.1 USB Card reader usage, case of using only Out-App permission.

14.1.1 Selecting Application for NALib USB Reader

When the USB reader is plugged into the port of your phone or tablet. Android system will detect the USB device and will ask the user which application to use with this USB device.



A screen asking the user which application to use this USB reader

- Please select the application that wants to use, such as NASample and then tap **Just Once**.
  - Should not select **Always** or **Remember this choice**. If selected, the next time you plug in the reader, the Android system will not let the user select other applications. It will use the selected application forever.
- If the wrong application is selected and **Just Once** is tapped, please unplug the reader and plug it again.
- If the wrong application is selected and **Always** is tapped, users have to reset the selection. Go to **Settings** > **Application Manager** or **App Info** > select the wrong selected application name > **Open by default** or **Launch by default** and tap **CLEAR DEFAULTS.** Then unplug the reader and plug it again.

14.1.2 If the USB reader is already plugged into the phone's port but no window popping up for application choosing, that maybe 3 reasons:

- The OTG system of the phone may be turned off. Access the Android's **Settings** menu and try searching for menus related to OTG activation (Oppo, Realme and Vivo mobile phone).
- That phone may not support OTG. Change the phone to a new device that supports OTG.
- Readers may be locked to use with other application, due to accidentally touching **Always** or **Remember this choice.** User has to **CLEAR DEFAULTS** of the application that is locked according to 14.1.1.

14.1.3 If the USB reader cannot be used, try unplugging the reader and plug it again then choose the correct application name.

14.2 Bluetooth 3 Card Reader Usage (for readers that have “BT” suffix model e.g. TDA301BT and TDAi301BT)

14.2.1 Bluetooth settings on smart phone

- Enable Bluetooth.
- Enable Location Service by swiping down Quick Settings from the top of your screen then tap “Location” icon as using GPS.
- Pair the reader.

14.2.2 Bluetooth 3 reader pairing (TDA301BT, TDAi301BT)

Normally, every Bluetooth 3 device to use on a mobile phone needs to be “Paired” before. The pairing between the Bluetooth card reader with “BT” suffix model and the mobile phone can do as follows:

- Install the batteries into the reader’s battery slot.
- Push the power button on the front of the Bluetooth reader. Notice there is a flashing blue light.
- On the phone or tablet, go to the Android **Settings** menu.
- Tap **Bluetooth** > **Scan** then look at the bottom of the screen, try scrolling to the bottom.
- Wait for a while, the card reader name starts with “FT_” like “FT__xxxxxxxxxxx” which “xxxxxx” will match the number or S/N on the back of the reader. Tap that reader name.
- If it can not find a reader, tap **Scan** again.

14.2.3 The Bluetooth reader should be paired with the phone before launching the application for the first time. If an application is launched before, then it will not connect to the reader.

14.2.4 Usage of TDA301BT and TDAi301BT Bluetooth reader

- Power control
  - Power on by pressing the front button, notice the blue light flashing.
  - Power off by pressing the front button 3 seconds, all LED light will turn off.
  - The reader will automatically shut down to save energy when:
    - Disconnected from the phone, turn off the Bluetooth signal of the phone or move the reader far away from the phone.
    - Didn't use the reader more than 3 minutes.
- Definition of LED lights of TDA301BT and TDAi301BT Bluetooth reader:
  - Blue: Blink 1 time: Wait for the connection.
  - Blue: Blink 2 times: Successfully connected. Can read the data.
  - Blue: Blink 3 times: Self-checking after power on and wait for the connection.
  - Blue: On: Successfully connected. Can read the data.
  - White: On or flash when the card is reading.
  - Yellow: On: Battery is weak. The reader should be charged.
  - Orange: On when charging the battery. And it will be turned off when the battery is full.
    - If the charger is plugged in but the orange light does not light up, indicating that the battery is full or almost full. No need to charge again. That is, if the charger is plugged in then the orange light is not lit, it does not mean that the reader is malfunctioning. But that means charging is no need and the reader is still working well.

14.2.5 Do not pair one Bluetooth reader with many Android devices, as it will be confusing.

14.2.6 One Android phone should be paired with only one Bluetooth reader. Always **unpair** (**Forget**) the unused reader in Android phone.

14.2.7 Bluetooth reader shutting down or Bluetooth connection dropped problems.

- This may be caused by a weak battery. (Observe the light will go out as soon as the card is read.) Test by plugging in the charger cable with the charging adapter, it will work immediately. (Do not charge by plugging into a computer or mobile phone, to be plugged into the adapter only)
- It can be caused by the Android device that turns off the Bluetooth itself when in standby mode or power save mode or security mode or when the screen is closed, depending on the user's Android device. Correct the problems by finding the way to set the Bluetooth don't turn off when entering the various modes above, or to disable the above modes.

- The Bluetooth reader sometimes may be disconnected by interference or moving the reader far away from the phone.
- When the Bluetooth connection is off, the reader will shut itself down. Therefore, users will have to press the Bluetooth reader button to continue to use it.

14.3 Bluetooth 4.0 (BLE0) Card Reader Usage (for readers that have "BL" or "BLM" suffix model e.g. TDA301BL, TDAi301BL, TDA301BLM and TDAi301BLM)

14.3.1 Bluetooth settings on smart phone

- Enable Bluetooth.
- Enable Location Service by swiping down Quick Settings from the top of your screen then tap "Location" icon as using GPS.
- **Don't pair the reader.**

14.3.2 Definition of LED lights of BLE0 (TDA301BL, TDAi301BL, TDA301BLM and TDAi301BLM) reader:

- Blue: Blink = Wait for the connection.
- Blue: On = Successfully connected. Can read the data.
- Yellow: On or flash when the card is reading.
- Red: On when charging the battery. And it will be turned off when the battery is full.
  - If the charger is plugged in but the red light does not light up, indicating that the battery is full or almost full. No need to charge again. That is, if the charger is plugged in then the red light is not lit, it does not mean that the reader is malfunctioning. But that means charging is no need and the reader is still working well.

14.3.3 Power control for BLE0 reader

- Power on by pressing the reader's button, notice the blue light flashing.
- Power off by pressing the front button 3 seconds, all LED light will turn off.
- The reader will automatically shut down to save energy when:
  - Disconnected from the phone, turn off the Bluetooth signal of the phone or move the reader far away from the phone.
  - Didn't use the reader more than 3 minutes.

14.3.4 BLE0 reader shutting down or Bluetooth connection dropped problems.

- This may be caused by a weak battery. (Observe the light will go out as soon as the card is read.) Test by plugging in the charger cable with the charging adapter, it will work immediately. (Do not charge by plugging into a computer or mobile phone, to be plugged into the charging adapter only)

- It can be caused by the Android device that turns off the Bluetooth itself when in standby mode or power save mode or security mode or when the screen is closed, depending on the user's Android device. Correct the problems by finding the way to set the Bluetooth don't turn off when entering the various modes above, or to disable the above modes.
- The BLE reader sometimes may be disconnected by interference or moving the reader far away from the phone.
- When the Bluetooth connection is off, the reader will shut itself down. Therefore, users will have to press the BLE reader button to continue to use it.

14.4 Bluetooth 4.0 (BLE1) Card Reader Usage (for readers that have "BE" suffix model e.g. TDA3901BE)

14.4.1 Bluetooth settings on smart phone

- Enable Bluetooth.
- Enable Location Service by swiping down Quick Settings from the top of your screen then tap "Location" icon as using GPS.
- **Don't pair the reader.**

14.4.2 Definition of LED lights of BLE1 (TDA3901BE) reader:

- Blue (Left LED): Bluetooth Mode
  - Fast–Slow flash: wait for the connection.
  - Slow flash: connected, waiting for card insertion.
  - On: Successfully connected. Can read the data.
  - Fast flash: data reading.
  - Off: sleep
- Red (Middle LED): Battery Charging
  - On: charging the battery.
  - Off: the battery is full.
  - Slow flash: low battery.
- Green (Right LED): USB Mode
  - Slow flash: USB connected, waiting for card insertion.
  - On: Successfully connected. Can read the data.
  - Fast flash: data reading.

14.4.3 Power control for BLE1 reader

- Power on by sliding up the reader's switch, notice the blue light flashing.
- Power off by sliding the reader's switch to middle position (O), all LED light will turn off.

14.4.4 Sleep Mode

- The reader will automatically sleep in 60 seconds if it has not been used
- You can wake it up by
  - Inserting or removing the card.
  - Let application read data from the card.
  - Turn off and turn on the reader.

14.4.5 BLE1 reader shutting down or Bluetooth connection dropped problems.

- This may be caused by a weak battery. (Observe the light will go out as soon as the card is read.) Test by plugging in the charger cable with the charging adapter, it will work immediately. (Do not charge by plugging into a computer or mobile phone, to be plugged into the adapter only)
- It can be caused by the Android device that turns off the Bluetooth itself when in standby mode or power save mode or security mode or when the screen is closed, depending on the user's Android device. Correct the problems by finding the way to set the Bluetooth don't turn off when entering the various modes above, or to disable the above modes.
- The BLE reader sometimes may be disconnected by interference or moving the reader far away from the phone.
- When the reader is not used for a long time, the reader will stop itself (into Sleep Mode). Insert the card and order to read the data, the reader will wake up and continue working.

15. NALib SDK New Features

R260520 (NALib 0.1.1614)

1. Added support for the TDC301VC and TDC39VC readers.
2. Supports bank card reading (getEmvTextCA).

R250924 (NALib 0.1.1600)

1. Supports targeting API Level 36 (Android 16).
2. Changed page size of native libraries to 16K for better performance and compatibility requirement .
3. Fixed some issues in the library.

R250807 (NALib 0.1.1504)

1. Added support for the TDA39VC2 reader.
2. Improved reading speed for the TDA301VC reader (Firmware 2.25 and above).
3. Updated the library and sample applications to allow setting the Target API Level 35 (Android 15).
4. Improved reading of the PM0 Thai ID card.

R241108 (NALib 0.1.1414)

1. Fixed some issues in the library.
2. Removed TDA01BT reader supporting

R240404 (NALib 0.1.1401)

1. Improved the library and sample applications to support the Target API Level 34 (Android 14).
2. Fixed some issues in the library.

R231011 (NALib 0.1.1306)

1. Added support for the PDA121RS and PDA3500RS readers.

R230620 (NALib 0.1.1304)

1. Added support for the TDA01BT Bluetooth reader.

R2211239 (NALib 0.1.1302)

1. Supports TDAi301VC firmware version 8.00

R221123 (NALib 0.1.1301)

1. Improved NALib to better support React Native and Flutter in Android 13.

R221024 (NALib 0.1.13)

1. The **`getNIDTextNA`** function has been improved to be able to read 5 more data items from Thai ID cards.
2. Added support for the TDA3901BE Bluetooth reader by setting BLE1 in the **`listOption`** variable of the **`getReaderListNA`** function.
3. Improved the sample applications to support the Target API Level 33 (Android 13).

R220804 (NALib 0.1.12)

1. Improved the **`updateLicenseFileNA`** function for auto selecting multiple license file download sources.
2. Improved the **`getReaderInfoNA`** function for getting the reader's firmware version.
3. Supports TDAi301U8A firmware version 6.xx and TDAi301VC readers firmware version 8.xx.

R220520 (NALib 0.1.10)

1. Supports new Bluetooth 4.0 (BLE) readers e.g. TDA301BL, TDAi301BL, TDA301BLM and TDAi301BLM.
2. Supports the Target API Level 31 (for Android 12) and can use Private Storage instead of External Storage.
3. Removed Location Service requesting when using Bluetooth and BLE readers in Android 12.
4. Added cancel button in pop-up screen of **`getReaderListNA`** command.
5. Improved Thai ID card reading speed for TDA301VC reader.
6. Supports Thai ID card version 0004 (JC3).
7. Updated the SDK developer's manual.
8. Fixed some issues in the library and the sample applications .

FAQs About HawkEye NALib SDK

Question

What is HawkEye NALib Software Development Kit?

Answer

HawkEye Thai ID Card Development Kit for Android (NALib SDK) is a hardware and software kit for developing applications on Android to read Thai national ID cards. The kit includes a smart card reader (USB or Bluetooth), built sample applications and source code.

Question

What is the difference of TDA readers such as TDA3310M2, TDA3310C2 and TDA301BT from TRA readers and standard smart card readers?

Answer

TDA is an SDK suite for Android application development. Suitable for programmers or developers.

TRA is a Thai ID card reading solution for Android phones and tablets. The package includes both a reader and application software. It is suitable for general end-users, no programmers required.

The standard card readers, such as the bR301, SCR3500B and SCR3310M2, are just smart card readers without application software. They are suitable for organizations that already have Thai ID card applications software.

Question

Can I buy only the standard smart card reader such as bR301, SCR3500B or SCR3310M2 (hardware only products) then use it with NALib SDK?

Answer

The HawkEye NALib software development kit is only compatible with registered readers in the TDA, TDAi and TDC series product. All readers are already registered for using with the NALib SDK before selling.

For unregistered readers such as standard readers (hardware only) and TRA readers can not use with NALib SDK software.

Question

What are the application development steps for Thai ID card reading?

Answer

Do these steps:

- Buy one set of HawkEye NALib product. It should be TDA301BL that can be used for developing both Bluetooth and USB readers.
- Study source code and description of API from HawkEye NALib SDK Manual.
- Develop a new project in Android Studio and test with the reader that comes from the package.
- When the application is complete. Purchase additional HawkEye NALib products for all users.

- Install application into the Android device and give it with HawkEye NALib reader to users.

Question

Can we use the NALib SDK developed application with other brand readers or hardware-only reader?

Answer

No, because the NALib SDK is only compatible with the card reader that comes with the package. Other card readers are not compatible.

Question

How is a Virtual Licensing (VL) card reader different from a normal File Licensing reader?

Answer

The Virtual Licensing or VL is a new generation of NALib licensing. Readers with VL are no longer need license file as normal File Licensing readers. The benefits of VL are easier application development, no update license file and do not need Internet access.

In addition, the VL reader can also be used with a normal reader that uses a license file, so it can be easily migrated from the old existing application.

Question

If the NALib card reader is broken, can I transfer the license right to another unregistered reader?

Answer

If the card reader is broken or damaged, the license is considered to be terminated.

However, all readers have a warranty. If the reader is damaged during the warranty period, the company will repair the reader to a normal state for customers, free of charge.

Question

Can NALib software get online data from the Department of Provincial Administration (DOPA)'s server?

Answer

No, NALib card reader will read the data from the chip in Thai ID card only. It works offline.

Question

What are the NALib SDK terms of use?

Answer

Conditions are:

- Do not use TDA SDK for illegal processes.
- Serve or use this TDA SDK for card holder only.
- Do not read the data from the card without the permission of the card holder.
- Do not disclose or distribute or share card's photos and text data to others without the permission of the card holder.

- Any damages caused by this NALib SDK or the information got from this NALib SDK, you must be liable for all, including the damages of civil law and others. R&D Computer System Co., Ltd. will not be held liable to and will not accept any liability, obligation or responsibility to all damages.

Question

Where can I download NALib SDK?

Answer

Developers can download the SDK software from the company website. Visit the HawkEye TDA or TDC product page (such as TDA301BL) and click on the link on that page.

Question

USB reader is already plugged into the phone's port but no window popping up for application choosing, both outside the application and while using the application.

Answer

Please check these

- The OTG system of the phone may be turned off. Access the Android's Settings menu and try searching for menus related to OTG activation, then enable it. (Oppo and Vivo mobile phone).
- That phone may not support OTG. Please change the phone to a new device that supports OTG.
- Readers may be locked to use with other application, due to accidentally touching Always or Remember this choice. User has to CLEAR DEFAULTS of the application that is locked by going to **Settings** > **Application Manager** or **App Info** > select the wrong selected application name > **Open by default** or **Launch by default** and tap **CLEAR DEFAULTS**. Then unplug the reader and plug it again.

To avoid this problem, we should modify the application to have a pop-up window inside the application (USB Reader In-App Permission) with **`setPermissionsNA(1)`** command.

Question

How is the Bluetooth 4.0 (BLE) reader different from normal Bluetooth 3.0 reader?

Answer

The Bluetooth Low Energy-BLE reader (TDA301BL, TDA301BLM and TDA3901BE) is a newer version of the original “BT” model e.g. TDA301BT and has the following differences:

- BLE is a technology of Bluetooth 4.0 , while the BT version uses the technology of Bluetooth 3.0.
- BLE uses less power.
- BLE doesn't need to be paired first. It can be used immediately, but BT must be paired before using.
- BLE reads data faster.
- BLE has a higher internal security than BT.

Question

What are the different of TDA3901BE and other BLE Readers?

Answer

The TDA3901BE BLE Reader is a reader from ACS, called BLE1. Other BLE readers include the TDA301BL, TDAi301BL, TDA301BLM and TDAi301BLM (called BLE0)

- TDA3901BE has Sleep Mode. The reader will not turn off after deselected but will sleep after it is not used for more than 1 minute. User can wake it by both software and by inserting – removing the smart card. While the reader in the BLE0 group will turn off immediately after deselection or after more than 3 minutes of inactivity.
- The TDA3901BE has a higher security feature than other BLE readers because it has AES-128 data encryption embedded inside.
- The TDA3901BE is the File Licensing (FL) reader and requires a license file to run, while the BLE0 reader is the Virtual Licensing (VL) reader.

บริษัท อาร์แอนด์ดี คอมพิวเตอร์ ซิสเท็ม จำกัด
R&D Computer System Co., Ltd.
(66)2-693-1745, (66)2-694-3133
sales@rd-comp.com
https://www.rd-comp.com