



UG20000

BBS Reader iOS MyKad Programming API User Guide

Rev 1.7 — Aug 5, 2025

Document information

Information	Content
Keywords	BBS, Bluetooth Connection, Linker Key, Terminal Key, AES-128
Abstract	This iOS SDK user manual provides the tools and frameworks necessary for developing applications for Apple iOS and iPad OS operating system.



Revision History

Revision history

Rev	Date	Description
1.0	Aug 10, 2024	1 st release of document on iOS SDK API of BBS Reader
1.1	Sept 3, 2024	<i>Add MyKid functions</i>
1.2	Oct 19, 2024	<i>Add MyKid JPN4 functions</i>
1.3	Feb 25, 2025	<i>Update GetBatteryLevel() description.</i>
1.4	June 8, 2025	<i>Update GetDeviceUUID() description.</i>
1.5	July 12, 2025	<i>Update ConnectSCCard() description.</i>
1.6	July 23, 2025	<i>Update user guide with additional Xcode Project setup details</i>
1.7	Aug 5, 2025	<i>Add Get MyKadFingerPrint() function.</i>

TABLE OF CONTENTS**Table of Contents**

1	Introduction	7
2	System Requirement	7
2.1	iOS SDK Folder Structure	7
2.2	Framework and Library Files	7
2.3	XCode Project Setup	8
2.3.1	Declare the delegate inside the ViewController.h	8
2.3.2	Delegate Initialization	8
2.3.3	Other Linker Flag under Build Settings	8
2.3.4	Link Binary with Libraries under Build Phases	9
2.3.5	Enable Bluetooth Permission in the Xcode project setting	9
2.3.6	Callback must be implemented to fulfil the Delegate requirements of SDK library.	9
3	Bluetooth Function Descriptions.....	10
3.1	Bluetooth API	10
3.1.1	(id) init	10
3.1.2	(void) BLEStartScan: (int) iTimeout	10
3.1.3	(int) IsDeviceConnect	10
3.1.4	(void) disconnectBTDevice	10
3.1.5	(void) didConnectReader: (int *)iConnectionStatus	11
3.1.6	(void) didBLEPowerOnState	11
3.1.7	(void) ConnectUUID: (NSString *)sUUID :(int)iTimeout	11
3.2	Device API	11
3.2.1	(NSString *) GetDeviceUUID	11
3.2.2	(NSString *) GetSDKVersion	12
3.2.3	(unsigned int) GetFWVersion: (uint_8 *)fwData :(int *)fwLen	12
3.2.4	(unsigned int) GetTID: (uint_8 *)tidData	12
3.2.5	(int) GetBatteryLevel	12
3.2.6	(unsigned int) Buzzer_Beep	13
4	Biometric Function description	14
4.1	Biometric Connection API.....	14
4.1.1	(unsigned int) btConnectBio	14
4.1.2	(unsigned int) btDisconnectBio.....	14
4.2	Biometric Callback function	14
4.2.1	(void) VerifyStatusCallBack :(int *)Status: (int)iLen: (u_char *)data	14
4.3	Biometric Verification API	15

4.3.1	(unsigned int) btBioVerifyRightNoAsync :(int) timeout :(int **)iMatchStatus	15
4.3.2	(unsigned int) btBioVerifyLeftNoAsync :(int) timeout :(int **)iMatchStatus	15
4.3.3	(unsigned int) btBioVerifyBothFPNoAsync :(int) timeout :(int **)iMatchStatus	16
4.3.4	(unsigned int) btBioVerifyPKBothFPNoAsync :(uint8_t)templateType :(uint8_t *)TemplateData1 :(int)iTemplateLength1 :(uint8_t *)TemplateData2 :(int)iTemplateLength2 :(int)iTimeout :(int **)iMatchStatus	16
5	<i>Smartcard Functions descriptions</i>	17
5.1	Generic Smartcard Functions	17
5.1.1	(unsigned int) InitReader	17
5.1.2	(unsigned int) CloseReader	17
5.1.3	(unsigned int) DisconnectCard	17
5.1.4	(unsigned int) getSlotStatus: (int *)iCardStatus	17
5.1.5	(unsigned int) getCardType: (int *)iCardStatus	18
5.2	MyKad Functions	18
5.2.1	(unsigned int) ConnectCard	18
5.2.2	(unsigned int) MyKadOriName :(NSData **) outdata	18
5.2.3	(unsigned int) MyKadGMPCName :(NSData **) outdata	18
5.2.4	(unsigned int) MyKadKPTShortName :(NSData **) outdata	18
5.2.5	(unsigned int) MyKadIDNum :(NSData **) outdata	19
5.2.6	(unsigned int) MyKadOldIDNum :(NSData **) outdata	19
5.2.7	(unsigned int) btMyKadBirthDate :(NSData **) outdata	19
5.2.8	(unsigned int) btMyKadBirthPlace :(NSData **) outdata	19
5.2.9	(unsigned int) btMyKadGender :(NSData **) outdata	19
5.2.10	(unsigned int) MyKadAdd1 :(NSData **) outdata	20
5.2.11	(unsigned int) MyKadAdd2 :(NSData **) outdata	20
5.2.12	(unsigned int) MyKadAdd3 :(NSData **) outdata	20
5.2.13	(unsigned int) MyKadCitizenship :(NSData **) outdata	20
5.2.14	(unsigned int) MyKadCity :(NSData **) outdata	20
5.2.15	(unsigned int) MyKadIssuedDate :(NSData **) outdata	21
5.2.16	(unsigned int) MyKadState :(NSData **) outdata	21
5.2.17	(unsigned int) MyKadPostcode :(NSData **) outdata	21
5.2.18	(unsigned int) MyKadRace :(NSData **) outdata	21
5.2.19	(unsigned int) MyKadReligion :(NSData **) outdata	21
5.2.20	(unsigned int) ReadFingerPrint :(NSData **) outpk1 :(NSData **) outpk2	22
5.2.21	(unsigned int) GetMyKadFingerPrint	22
5.2.22	(unsigned int) MyKadImage :(NSData **) outdata	22
5.2.23	(unsigned int) MyKadEMOrigin :(NSData **) outdata	22
5.2.24	(unsigned int) MyKadRJ :(NSData **) outdata	22
5.2.25	(unsigned int) MyKadKT :(NSData **) outdata	23
5.2.26	(unsigned int) MyKadEOtherID :(NSData **) outdata	23
5.2.27	(unsigned int) MyKadOtherIDCategory :(NSData **) outdata	23
5.2.28	(unsigned int) MyKadCardVer :(NSData **) outdata	23
5.2.29	(unsigned int) MyKadGreenCardExp :(NSData **) outdata	23
5.2.30	(unsigned int) MyKadGreenCardNationality :(NSData **) outdata	23

5.2.31	(unsigned int) getDetails :(struct CardInfo *) CardInfo	24
5.3	MyKid Functions	24
5.3.1	(unsigned int) getMyKidDetails :(struct MyKidInfo *) mykidInfo	24
5.3.2	(unsigned int) MyKidGetJPN1Version:(NSData **) outdata	24
5.3.3	(unsigned int) MyKidGetJPN1Size:(NSData **) outdata	24
5.3.4	(unsigned int) MyKidGetBirthCertNo:(NSData **) outdata	24
5.3.5	(unsigned int) MyKidGetIDNo:(NSData **) outdata	25
5.3.6	(unsigned int) MyKidGetName:(NSData **) outdata	25
5.3.7	(unsigned int) MyKidGetGender:(NSData **) outdata	25
5.3.8	(unsigned int) MyKidGetCitizenship:(NSData **) outdata	25
5.3.9	(unsigned int) MyKidGetBirthState:(NSData **) outdata	25
5.3.10	(unsigned int) MyKidGetAddress1:(NSData **) outdata	25
5.3.11	(unsigned int) MyKidGetAddress2:(NSData **) outdata	26
5.3.12	(unsigned int) MyKidGetAddress3:(NSData **) outdata	26
5.3.13	(unsigned int) MyKidGetPostcode:(NSData **) outdata	26
5.3.14	(unsigned int) MyKidGetCity:(NSData **) outdata	26
5.3.15	(unsigned int) MyKidGetState:(NSData **) outdata	26
5.3.16	(unsigned int) MyKidGetFatherReligion:(NSData **) outdata	26
5.3.17	(unsigned int) MyKidGetMotherReligion:(NSData **) outdata	27
5.3.18	(unsigned int) MyKidGetJPN2Version:(NSData **) outdata	27
5.3.19	(unsigned int) MyKidGetJPN2Size:(NSData **) outdata	27
5.3.20	(unsigned int) MyKidGetBirthDate:(NSData **) outdata	27
5.3.21	6.3.21 (unsigned int) MyKidGetBirthTime:(NSData **) outdata	27
5.3.22	(unsigned int) MyKidGetBirthPlace1:(NSData **) outdata	27
5.3.23	(unsigned int) MyKidGetBirthPlace2:(NSData **) outdata	28
5.3.24	(unsigned int) MyKidGetRegistrationDate:(NSData **) outdata	28
5.3.25	(unsigned int) MyKidGetNewName:(NSData **) outdata	28
5.3.26	(unsigned int) MyKidGetNewAdd1:(NSData **) outdata	28
5.3.27	(unsigned int) MyKidGetNewAdd2:(NSData **) outdata	28
5.3.28	(unsigned int) MyKidGetNewAdd3:(NSData **) outdata	28
5.3.29	(unsigned int) MyKidGetNewPostcode:(NSData **) outdata	29
5.3.30	(unsigned int) MyKidGetNewCity:(NSData **) outdata	29
5.3.31	(unsigned int) MyKidGetNewState:(NSData **) outdata	29
5.3.32	(unsigned int) MyKidGetMotherID:(NSData **) outdata	29
5.3.33	(unsigned int) MyKidGetMOtherDocNo:(NSData **) outdata	29
5.3.34	(unsigned int) MyKidGetMOtherDocType:(NSData **) outdata	29
5.3.35	(unsigned int) MyKidGetMotherName:(NSData **) outdata	30
5.3.36	(unsigned int) MyKidGetMotherDOB:(NSData **) outdata	30
5.3.37	(unsigned int) MyKidGetMotherCitizenship:(NSData **) outdata	30
5.3.38	(unsigned int) MyKidGetMotherRace:(NSData **) outdata	30
5.3.39	(unsigned int) MyKidGetFatherID:(NSData **) outdata	30
5.3.40	(unsigned int) MyKidGetFOtherDocNo:(NSData **) outdata	30
5.3.41	(unsigned int) MyKidGetFOtherDocType:(NSData **) outdata	31
5.3.42	(unsigned int) MyKidGetFatherName:(NSData **) outdata	31
5.3.43	(unsigned int) MyKidGetFatherDOB:(NSData **) outdata	31

5.3.44	(unsigned int) MyKidGetFatherCitizenship:(NSData **) outdata	31
5.3.45	(unsigned int) MyKidGetFatherRace:(NSData **) outdata	31
5.3.46	(unsigned int) MyKidGetRegistrationPlace:(NSData **) outdata	32
5.3.47	(unsigned int) MyKidGetJPN4Version:(NSData **) outdata	32
5.3.48	(unsigned int) MyKidGetJPN4Size:(NSData **) outdata	32
5.3.49	(unsigned int) MyKidGetContactAddress:(NSData **) outdata	32
5.3.50	(unsigned int) MyKidGetContactName:(NSData **) outdata	32
5.3.51	(unsigned int) MyKidGetContactTelNum:(NSData **) outdata	33
6	Appendix 1	34
6.1	Error Code	34
6.2	Enrollment and Verification status code	36
6.3	Workflow for Bluetooth Connection Setup	38
6.3.1	Workflow for First Time Bluetooth Setup	38
6.3.2	Workflow for Bluetooth Disconnection	39
6.3.3	Workflow for Non-First Bluetooth Connection	39
6.4	Workflow for MyKad Connection Setup	40
6.4.1	Workflow for block reading MyKad at once includes photo and minutiae.	40
6.4.2	Workflow for block reading MyKad in individual process	41
6.4.3	Workflow for single reading MyKad data	42
6.5	Workflow for Biometric Connection Setup	43
6.5.1	Workflow for Biometric to verify user live fingerprint against both MyKad's minutiae at once. ...	43
6.5.2	Workflow for Biometric Verification based on external minutiae	44
7	Abbreviations	45

1 Introduction

This document describes the Biometric library APIs support in iOS system via Bluetooth Low Energy (BLE) communication.

2 System Requirement

OS Supported: iOS SDK API version 10 onwards

IDE Environment: XCode

Hardware: iPhone and iPad OS operating system with BLE v4.0 onwards

2.1 iOS SDK Folder Structure

BBR10 iOS SDK release history.txt	Text file describes release history
MyKad_Programming_UserManual_iSDK_v1.5	User Guide of Biopad Programming API
BLEReaders.h	Header file
bbsrSdkPinDemo.zip	Demo Source Code
libBLEReaders.a	Library
Resources.bundle	PinPad Layout resource

2.2 Framework and Library Files

File name:

- a) libBLEReaders.a : Bluetooth Biometric Library
- b) BLEReaders.h : Library Header file
- c) Open SSL header files for cryptography
 - \third-party\OpenSSL\include\openssl\

2.3 XCode Project Setup

2.3.1 Declare the delegate inside the **ViewController.h**

```

1  //
2  //  ViewController.h
3  //  BSRGenericDemo
4  //
5  //  Created by kevin Chia on 23/7/25.
6  //
7
8  #import <UIKit/UIKit.h>
9  #import "BLEReader.h"
10
11 @interface ViewController : UIViewController<BLEReaderDelegate>{
12     BLEReader *blereader;
13 }
14
15
16 @end
17
18

```

2.3.2 Delegate Initialization

Developer must initialize the delegate object inside the **ViewController.m**,


```

blereader = [[BLEReader alloc] init];
[blereader setDelegate:self];

```

2.3.3 Other Linker Flag under Build Settings

▼ Linking - General

Setting	 bbsrDemo
Dynamic Library Install Name	
Dynamic Library Install Name Base	
Link With Standard Libraries	Yes
Mach-O Type	Executable
Other Librarian Flags	
Other Linker Flags	-ObjC
Quote Linker Arguments	Yes
Separately Edit Symbols	No

2.3.4 Link Binary with Libraries under Build Phases

▼ Link Binary With Libraries (4 items)

Name	Status
 CoreBluetooth.framework	Required
 UIKit.framework	Required
 libBLERead.a	Required
 Foundation.framework	Required

2.3.5 Enable Bluetooth Permission in the Xcode project setting

Develop needs to add the Privacy – Bluetooth Always usage Description to enable the Bluetooth Permission in the plist as below.

Privacy - Bluetooth Always Usage Description	String	App needs to use Bluetooth to exchange
--	--------	--

2.3.6 Callback must be implemented to fulfil the Delegate requirements of SDK library.

```
// Verify user status called back
- (void)VerifyStatusCallBack :(int *)status :(int)iLen :(u_char *)data
{
    unsigned int value = 0;
    value = *status;

    NSLog(@"VerifyStatusCallBack:: %d", value);
    dispatch_async(dispatch_get_main_queue(), ^{
        switch (value) {
            case BT_OK:
                NSLog(@"Verify Match OK");
                //self->_lblStatus.text = @"Verify Match OK";

                break;
            case BIO_FINGER_NOT_MATCH:
                NSLog(@"Finger Not Match");
                //self->_lblStatus.text = @"Finger Not Match";
                break;

            default:
                break;
        }
    });
}
```

3 Bluetooth Function Descriptions

3.1 Bluetooth API

3.1.1 (id) init

This function initializes the device object.

Parameter : none

Return : device object

3.1.2 (void) BLEStartScan: (int) iTimeout

This function initializes and scans for peripheral devices.

Parameter : iTimeout – A BLE master scan timeout refers to the duration for which a Blue Low Energy (BLE) central device (master) scans for advertising peripheral device before automatically stopping the scan.

Return : void

3.1.3 (int) IsDeviceConnect

This function returns the central device connection status

Parameter : none

Return : 0 - no device connection has been found
1 - device connection has established

3.1.4 (void) disconnectBTDevice

This function is to disconnect the peripheral device (BBS Reader) from central device's (iPhone or iPad) Bluetooth connection

Parameter : none

Return : none

3.1.5 (void) didConnectReader: (int *)iConnectionStatus

This function needs to be implemented to receive the BLE device connection status callback. Upon device connection has been established, central device could start to initial any function API calls to peripheral device.

Parameter : none

Return : iConnectionStatus = 0 - no device connection has been found
= 1 - client device connection has been established
= 2 - device scan has timeout

3.1.6 (void) didBLEPowerOnState

This is delegate function returns central peripheral status when its BLE power stage in at ON.

Parameter : none

Return : none

3.1.7 (void) ConnectUUID: (NSString *)sUUID :(int)iTimeout

This function is to establishes central device Bluetooth connection with peripheral device via assigned UUID.

Parameter : (NSString *) sUUID – unique identifier for each peripheral.

: (int) iTimeout – A BLE master scan timeout refers to the duration for which a Blue Low Energy (BLE) central device (master) scans for advertising peripheral device before automatically stopping the scan.

Return : none

3.2 Device API

3.2.1 (NSString *) GetDeviceUUID

This function is to return BLE peripheral device's UUID.

Parameter : none

Return : Peripheral device UUID string

3.2.2 (NSString *) GetSDKVersion

This function returns the string of SDK version

Parameter : none

Return : string of SDK version

3.2.3 (unsigned int) GetFWVersion: (uint_8 *)fwData :(int *)fwLen

This function returns the peripheral device of BBS Reader's firmware version

Parameter : (uint_8 *)fwData – pointer for out buffer data of firmware version

: (int *)fwLen – pointer for out buffer length of firmware version

Return : BBS Reader firmware version data

3.2.4 (unsigned int) GetTID: (uint_8 *)tidData

This function returns the BBS Reader Terminal ID (TID)

Parameter : (uint_8 *)tidData – pointer for out buffer data of TID

Return : Reader TID data

3.2.5 (int) GetBatteryLevel

This function returns the current BBS Reader's battery data in integer value.

Parameter : none

Return : FW version < V3.0.0

return value: 0, 10, 20, 40, 60, 80, 100

: FW version > V3.0.0

return value : 0 - 100%

: 255(0xFF) means error battery reading.

3.2.6 (unsigned int) Buzzer_Beep

This function is to trigger buzzer sound from BBS Reader.

Parameter : none

Return : refer to Error Code in Appendix 1.

4 Biometric Function description

4.1 Biometric Connection API

4.1.1 (unsigned int) btConnectBio

This function is to trigger BBS Reader to power on Biometric fingerprint scanner and to establish the connection.

Parameter : none

Return : FW version < V3.0.0,
return fails if battery level < 20%
: FW version > V3.0.0,
return fails if battery level < 1%

4.1.2 (unsigned int) btDisconnectBio

This function is to trigger BBS Reader to power off the Biometric fingerprint scanner.

Parameter : none

Return : refer to Error Code in Appendix 1.

4.2 Biometric Callback function

4.2.1 (void) VerifyStatusCallBack :(int *)Status: (int)iLen: (u_char *)data

This is Biometric API callback function that it receives return status of the biometrics enrollment or verification API. This callback is a must implementation to avoid compilation error.

Parameter : (int) iStatus - refer to Error Code in Appendix
: (int) iLen – Length of the return data, if any
: (u_char*) data – return data

Return : refer to Error Code in Appendix 1.

4.3 Biometric Verification API

4.3.1 (unsigned int) btBioVerifyRightNoAsync :(int) timeout :(int **iMatchStatus

This function captures a user live fingerprint, and it matches with the MyKad Right thumb minutiae without asynchronous messages.

Perquisition : app must call the MyKadPKRight() or getMinutaie() API.

Parameter : (int) timeout (IN) – duration time in second to wait for user places finger on the sensor before its timeout. 0 value means wait forever.

: (int **) iMatchStatus (OUT)

0 = finger has matched,

-8 = finger does not match

Return : refer to Error Code in Appendix 1.

4.3.2 (unsigned int) btBioVerifyLeftNoAsync :(int) timeout :(int **iMatchStatus

This function captures a user live fingerprint, and it matches with the MyKad Left thumb minutiae without asynchronous messages.

Perquisition : app must call the MyKadPKLeft() or getMinutaie() API.

Parameter : (int) timeout (IN) – duration time in second to wait for user places finger on the sensor before its timeout. 0 value means wait forever.

: (int **) iMatchStatus (OUT)

0 = finger has matched,

-8 = finger does not match

Return : refer to Error Code in Appendix 1.

4.3.3 (unsigned int) btBioVerifyBothFPNoAsync :(int) timeout :(int **iMatchStatus

This function captures a user live fingerprint, and it matches with both MyKad minutiae at once with asynchronous messages.

Perquisition : app must call the getMinutiae() API.

Parameter : (int) timeout (IN) – duration time in second to wait for user places finger on the sensor before its timeout. 0 value means wait forever.

: (int **) iMatchStatus (OUT)

0 = finger has matched,

-8 = finger does not match

Return : refer to Error Code in Appendix 1.

4.3.4 (unsigned int)

btBioVerifyPKBothFPNoAsync :(uint8_t)templateType :(uint8_t *)TemplateData1 :(int)iTemplateLength1 :(uint8_t *)TemplateData2 :(int)iTemplateLength2 :(int)iTimeout :(int **)iMatchStatus

This function captures a user live fingerprint, and it matches with provided both minutiae from the TemplateData1 and TemplateData2 parameter of API at once. Asynchronous messages have been enabled.

Parameter : (uint8_t) templateType (IN): Export fingerprint minutiae template format. (refer to Appendix)

: (uint8_t *) TemplateData1 (IN): input fingerprint minutiae data

: (int) iTemplateLength1 (IN): input fingerprint minutiae length

: (uint8_t *) TemplateData2 (IN): input fingerprint minutiae data

: (int) iTemplateLength2 (IN): input fingerprint minutiae length

: (int) iTimeout (IN): To specify the wait time of the fingerprint sensor

: (int **) iMatchStatus (OUT)

0 = finger has matched,

-8 = finger does not match

Return : refer to Error Code in Appendix 1.

5 Smartcard Functions descriptions

5.1 Generic Smartcard Functions

5.1.1 (unsigned int) InitReader

This function is to trigger power on the BBS Reader smartcard reader.

Parameter : none

Return : refer to error code

5.1.2 (unsigned int) CloseReader

This function is to trigger power off the BBS Reader smartcard reader.

Parameter : none

Return : refer to error code

5.1.3 (unsigned int) DisconnectCard

This function is to trigger BBS Reader to power off the inserted smartcard based on the slot number.

Parameter : none

Return : refer to error code

5.1.4 (unsigned int) getSlotStatus: (int *)iCardStatus

This function is to get the current connected slot of smartcard from the BBS Reader.

Parameter : none

Return : 0 – Card absents

: 1 – Card present

5.1.5 (unsigned int) getCardType: (int *)iCardStatus

This function is to get current connected Mykad card type from BBS Reader.

Parameter : none

Return : 0 – MyKad
: 1 – EBA MyKad
: 2 – MyKid

5.2 MyKad Functions

5.2.1 (unsigned int) ConnectCard

This function is to trigger BBS Reader to power on the inserted smartcard specifically on slot 0. This API will send Select MyKad AID after MyKad has been power on successfully. MyKad must be inserted into the smartcard connector before calling this function.

Perquisition: For MyKad only

Parameter : none

Return : refer to error code

5.2.2 (unsigned int) MyKadOriName :(NSData **) outdata

This function gets MyKad original name up to size of 150 bytes.

Parameter : (NSData **) outData (OUT) - original name of the cardholder (In ASCII character).

Return : refer to error code

5.2.3 (unsigned int) MyKadGMPCName :(NSData **) outdata

This function gets MyKad GMPC name up to size of 80 bytes.

Parameter : (NSData **) outData (OUT) - GMPC name of the cardholder (In ASCII character).

Return : refer to error code

5.2.4 (unsigned int) MyKadKPTShortName :(NSData **) outdata

This function gets MyKad KPT short name up to size of 40 bytes.

Parameter : (NSData **) outData (OUT) - KPT short name of the cardholder (In ASCII character).

Return : refer to error code

5.2.5 (unsigned int) MyKadIDNum :(NSData **) outdata

This function gets MyKad new ID number up to size of 13 bytes.

Parameter : (NSData **) outData (OUT) - ID number of cardholder (In ASCII character).

Return : refer to error code

5.2.6 (unsigned int) MyKadOldIDNum :(NSData **) outdata

This function gets MyKad ID number assigned prior to year 1990 with length up to size of 8 bytes.

Parameter : (NSData **) outData (OUT) - old ID no. of cardholder (In ASCII character).

Return : refer to error

5.2.7 (unsigned int) btMyKadBirthDate :(NSData **) outdata

This function gets MyKad Date of birth with size of 4 bytes. The DoB has formatted as: YYYYMMDD

Parameter : (NSData **) outData (OUT) - Birth date of the cardholder (In Integer value).

Return : refer to error code

5.2.8 (unsigned int) btMyKadBirthPlace :(NSData **) outdata

This function gets MyKad place of birth with size of 25 bytes.

Parameter : (NSData **) outData (OUT) - Birthplace of the cardholder (In ASCII character).

Return : refer to error code

5.2.9 (unsigned int) btMyKadGender :(NSData **) outdata

This function gets MyKad gender with size of 1 byte.

Parameter : (NSData **)outData (OUT) - Gender of the cardholder (In ASCII character).

L = Lelaki

P = Perempuan

Return : refer to error code

5.2.10 (unsigned int) MyKadAdd1 :(NSData **) outdata

This function gets MyKad first line of address with size of 30 bytes.

Parameter : (NSData **) outData (OUT) - address line 1 (In ASCII character).

Return : refer to error code

5.2.11 (unsigned int) MyKadAdd2 :(NSData **) outdata

This function gets Mykad second line of address with size of 30 bytes.

Parameter : (NSData **) outData (OUT) - address line 2 (In ASCII character).

Return : refer to error code

5.2.12 (unsigned int) MyKadAdd3 :(NSData **) outdata

This function gets MyKad third line of address with size of 30 bytes.

Parameter : (NSData **) outData (OUT) - address line 3 (In ASCII character).

Return : refer to error code

5.2.13 (unsigned int) MyKadCitizenship :(NSData **) outdata

This function gets Mykad Citizenship status with size of 18 bytes.

Parameter : (NSData **) outData(OUT) - citizenship of the cardholder (In ASCII character).

Return : refer to error code

5.2.14 (unsigned int) MyKadCity :(NSData **) outdata

This function gets MyKad city with size of 25 bytes.

Parameter : (NSData **) outData (OUT) - city of the address (In ASCII character).

Return : refer to error code

5.2.15 (unsigned int) MyKadIssuedDate :(NSData **) outdata

This function gets MyKad date of card personalized with size of 4 bytes and it based on format as: YYYYMMDD

Parameter : (NSData **) outData (OUT) - card issue date (In Integer value).

Return : refer to error code

5.2.16 (unsigned int) MyKadState :(NSData **) outdata

This function gets Mykad state with size of 30 bytes.

Parameter : (NSData **) outData (OUT) - State of the address (In ASCII character).

Return : refer to error code

5.2.17 (unsigned int) MyKadPostcode :(NSData **) outdata

This function gets MyKad postcode of address with size of 3 bytes (5 digits).

Parameter : (NSData **) outData (OUT) - postcode of the address (In Integer value).

Return : refer to error code

5.2.18 (unsigned int) MyKadRace :(NSData **) outdata

This function gets MyKad racial origin with size of 25 bytes.

Parameter : (NSData **) outData (OUT)- race of the cardholder (In ASCII character).

Return : refer to error code

5.2.19 (unsigned int) MyKadReligion :(NSData **) outdata

This function gets MyKad religion with size of 11 bytes.

Parameter : (NSData **) outData (OUT) - religion of the cardholder (In ASCII character).

Return : refer to error code

5.2.20 (unsigned int) ReadFingerPrint :(NSData **) outpk1 :(NSData **) outpk2

This function gets MyKad right and left minutiae with size of 512 bytes for each.

Parameter : (NSData **) outpk1 (OUT)- right minutiae sized 512 bytes

: (NSData **) outpk2 (OUT)- left minutiae sized 512 bytes

Return : refer to error code

5.2.21 (unsigned int) GetMyKadFingerPrint

This function read MyKad right and left minutiae with size of 512 bytes for each.

Parameter : none

Return : refer to error code

5.2.22 (unsigned int) MyKadImage :(NSData **) outdata

This function gets MyKad 24-bit color photo in jpeg format. MyKad Image binary size is up to 4000 bytes.

Parameter : (NSData **) outData (OUT) - image binary from card.

Return : refer to error code

5.2.23 (unsigned int) MyKadEMOrigin :(NSData **) outdata

This function gets MyKad East Malaysian origin with size of 1 byte.

Parameter : (NSData **) outData (OUT) - East Malaysian origin (In ASCII character).

Return : refer to error code

5.2.24 (unsigned int) MyKadRJ :(NSData **) outdata

This function gets MyKad RJ with size of 2 bytes.

Parameter : (NSData **) outData (OUT) - RJ of the cardholder (In ASCII character).

Return : refer to error code

5.2.25 (unsigned int) MyKadKT :(NSData **) outdata

This function gets MyKad KT with size of 2 bytes.

Parameter : (NSData **) outData (OUT) - KT of the cardholder (In ASCII character).

Return : refer to error code

5.2.26 (unsigned int) MyKadEOtherID :(NSData **) outdata

This function gets MyKad other ID with size of 11 bytes.

Parameter : (NSData **) outData (OUT) – Other ID of the cardholder (In ASCII character).

Return : refer to error code

5.2.27 (unsigned int) MyKadOtherIDCategory :(NSData **) outdata

This function gets MyKad other ID category with size of 1 byte.

Parameter : (NSData **) outData (OUT) – Other ID category of the cardholder (In ASCII character).

Return : refer to error code

5.2.28 (unsigned int) MyKadCardVer :(NSData **) outdata

This function gets MyKad card version with size of 1 byte.

Parameter : (NSData **) outData (OUT) – Card version of MyKad (In ASCII character).

Return : refer to error code

5.2.29 (unsigned int) MyKadGreenCardExp :(NSData **) outdata

This function gets MyKad green card expiry date size of 4 bytes.

Parameter : (NSData **) outData (OUT) – Green Card expiry date (In ASCII character).

Return : refer to error code

5.2.30 (unsigned int) MyKadGreenCardNationality :(NSData **) outdata

This function gets MyKad green card nationality with size of 20 bytes.

Parameter : (NSData **) outData (OUT) – Green card nationality (In ASCII character).

Return : refer to error code

5.2.31 (unsigned int) getDetails :(struct CardInfo *) CardInfo

This function retrieves all MyKad data and returns in a data structure.

Parameter : (struct CardInfo *) CardInfo (OUT) – MyKad data fields.

Return : refer to error code

5.3 MyKid Functions

5.3.1 (unsigned int) getMyKidDetails :(struct MyKidInfo *) mykidInfo

This function retrieves all MyKid data and returns in a structure.

Parameter : (struct MyKidInfo *) mykidInfo (OUT) – MyKid data fields.

Return : refer to error code.

5.3.2 (unsigned int) MyKidGetJPN1Version:(NSData **) outdata

This function retrieves MyKid JPN1 version.

Parameter : (NSData **) outData (OUT) – JPN1 version number (In Numeric).

Return : refer to error code.

5.3.3 (unsigned int) MyKidGetJPN1Size:(NSData **) outdata

This function retrieves MyKid JPN1.

Parameter : (NSData **) outData (OUT) – JPN1 data size (In Numeric).

Return : refer to error code.

5.3.4 (unsigned int) MyKidGetBirthCertNo:(NSData **) outdata

This function retrieves MyKid birth cert number.

Parameter : (NSData **) outData (OUT) – Childbirth certificate number (In Alphanumeric).

Return : refer to error code.

5.3.5 (unsigned int) MyKidGetIDNo:(NSData **) outdata

This function retrieves MyKid Identity Card number.

Parameter : (NSData **) outData (OUT) – Child Identity card number (In Alphanumeric).

Return : refer to error code.

5.3.6 (unsigned int) MyKidGetName:(NSData **) outdata

This function retrieves MyKid name of child.

Parameter : (NSData **) outData (OUT) – name of child (In Alphanumeric).

Return : refer to error code.

5.3.7 (unsigned int) MyKidGetGender:(NSData **) outdata

This function retrieves MyKid gender of child.

Parameter : (NSData **) outData (OUT) – gender of child (In Alphanumeric).

Return : refer to error code.

5.3.8 (unsigned int) MyKidGetCitizenship:(NSData **) outdata

This function retrieves MyKid citizenship of child.

Parameter : (NSData **) outData (OUT) – citizenship of child (In Alphanumeric).

Return : refer to error code.

5.3.9 (unsigned int) MyKidGetBirthState:(NSData **) outdata

This function retrieves MyKid state where the child was born.

Parameter : (NSData **) outData (OUT) – state where the child was born (In Alphanumeric).

Return : refer to error code.

5.3.10 (unsigned int) MyKidGetAddress1:(NSData **) outdata

This function retrieves MyKid first line of address.

Parameter : (NSData **) outData (OUT) – first line of address (In Alphanumeric).

Return : refer to error code.

5.3.11 (unsigned int) MyKidGetAddress2:(NSData **) outdata

This function retrieves MyKid second line of address.

Parameter : (NSData **) outData (OUT) – second line of address (In Alphanumeric).

Return : refer to error code.

5.3.12 (unsigned int) MyKidGetAddress3:(NSData **) outdata

This function retrieves MyKid third line of address.

Parameter : (NSData **) outData (OUT) – third line of address (In Alphanumeric).

Return : refer to error code.

5.3.13 (unsigned int) MyKidGetPostcode:(NSData **) outdata

This function retrieves MyKid postcode of address.

Parameter : (NSData **) outData (OUT) – postcode of address (In Numeric).

Return : refer to error code.

5.3.14 (unsigned int) MyKidGetCity:(NSData **) outdata

This function retrieves MyKid city of address.

Parameter : (NSData **) outData (OUT) – city of address (In Alphanumeric).

Return : refer to error code.

5.3.15 (unsigned int) MyKidGetState:(NSData **) outdata

This function retrieves MyKid state of address.

Parameter : (NSData **) outData (OUT) – state of address (In Alphanumeric).

Return : refer to error code.

5.3.16 (unsigned int) MyKidGetFatherReligion:(NSData **) outdata

This function retrieves MyKid father's religion.

Parameter : (NSData **) outData (OUT) – father's religion (In Alphanumeric).

Return : refer to error code.

5.3.17 (unsigned int) MyKidGetMotherReligion:(NSData **) outdata

This function retrieves MyKid mother's religion.

Parameter : (NSData **) outData (OUT) – mother's religion (In Alphanumeric).

Return : refer to error code.

5.3.18 (unsigned int) MyKidGetJPN2Version:(NSData **) outdata

This function retrieves MyKid JPN2 version number.

Parameter : (NSData **) outData (OUT) – JPN2 version number (In Numeric).

Return : refer to error code.

5.3.19 (unsigned int) MyKidGetJPN2Size:(NSData **) outdata

This function retrieves MyKid JPN2 data size.

Parameter : (NSData **) outData (OUT) – JPN2 data size (In Numeric).

Return : refer to error code.

5.3.20 (unsigned int) MyKidGetBirthDate:(NSData **) outdata

This function retrieves MyKid date child was born.

Parameter : (NSData **) outData (OUT) – date child was born (In Numeric YYYYMMDD).

Return : refer to error code.

5.3.21 6.3.21 (unsigned int) MyKidGetBirthTime:(NSData **) outdata

This function retrieves MyKid time child was born.

Parameter : (NSData **) outData (OUT) – time child was born (In Numeric HHMMSS).

Return : refer to error code.

5.3.22 (unsigned int) MyKidGetBirthPlace1:(NSData **) outdata

This function retrieves MyKid first line of location of birth.

Parameter : (NSData **) outData (OUT) – first line of location of birth (In Alphanumeric).

Return : refer to error code.

5.3.23 (unsigned int) MyKidGetBirthPlace2:(NSData **) outdata

This function retrieves MyKid second line of location of birth.

Parameter : (NSData **) outData (OUT) – second line of location of birth (In Alphanumeric).

Return : refer to error code.

5.3.24 (unsigned int) MyKidGetRegistrationDate:(NSData **) outdata

This function retrieves MyKid date of birth registration.

Parameter : (NSData **) outData (OUT) – date of birth registration (In Numeric YYYYMMDD).

Return : refer to error code.

5.3.25 (unsigned int) MyKidGetNewName:(NSData **) outdata

This function retrieves MyKid new or current name of child.

Parameter : (NSData **) outData (OUT) – new/current name of child. (In Alphanumeric).

Return : refer to error code.

5.3.26 (unsigned int) MyKidGetNewAdd1:(NSData **) outdata

This function retrieves MyKid first line of new or current address.

Parameter : (NSData **) outData (OUT) – first line of new/current address. (In Alphanumeric).

Return : refer to error code.

5.3.27 (unsigned int) MyKidGetNewAdd2:(NSData **) outdata

This function retrieves MyKid second line of new or current address.

Parameter : (NSData **) outData (OUT) – second line of new/current address. (In Alphanumeric).

Return : refer to error code.

5.3.28 (unsigned int) MyKidGetNewAdd3:(NSData **) outdata

This function retrieves MyKid third line of new/current address.

Parameter : (NSData **) outData (OUT) – third line of new/current address. (In Alphanumeric).

Return : refer to error code.

5.3.29 (unsigned int) MyKidGetNewPostcode:(NSData **) outdata

This function retrieves MyKid postcode of new or current address.

Parameter : (NSData **) outData (OUT) – postcode of new/current address. (In Numeric).

Return : refer to error code.

5.3.30 (unsigned int) MyKidGetNewCity:(NSData **) outdata

This function retrieves MyKid city of new or current address.

Parameter : (NSData **) outData (OUT) – city of new/current address. (In Alphanumeric).

Return : refer to error code.

5.3.31 (unsigned int) MyKidGetNewState:(NSData **) outdata

This function retrieves MyKid state of new or current address.

Parameter : (NSData **) outData (OUT) – state of new/current address. (In Alphanumeric).

Return : refer to error code.

5.3.32 (unsigned int) MyKidGetMotherID:(NSData **) outdata

This function retrieves MyKid mother's identity card number.

Parameter : (NSData **) outData (OUT) – mother's identity card number (In Alphanumeric).

Return : refer to error code.

5.3.33 (unsigned int) MyKidGetMOtherDocNo:(NSData **) outdata

This function retrieves MyKid mother's other Doc number.

Parameter : (NSData **) outData (OUT) – mother's other ID number (In Alphanumeric).

Return : refer to error code.

5.3.34 (unsigned int) MyKidGetMOtherDocType:(NSData **) outdata

This function retrieves MyKid category of mother's other Doc type.

Parameter : (NSData **) outData (OUT) – category of mother's other ID (In Alphanumeric).

Return : refer to error code.

5.3.35 (unsigned int) MyKidGetMotherName:(NSData **) outdata

This function retrieves MyKid mother's name.

Parameter : (NSData **) outData (OUT) – mother's name (In Alphanumeric).

Return : refer to error code.

5.3.36 (unsigned int) MyKidGetMotherDOB:(NSData **) outdata

This function retrieves MyKid mother's date of birth.

Parameter : (NSData **) outData (OUT) – mother's date of birth (In Numeric YYYYMMDD).

Return : refer to error code.

5.3.37 (unsigned int) MyKidGetMotherCitizenship:(NSData **) outdata

This function retrieves MyKid mother's citizenship.

Parameter : (NSData **) outData (OUT) – mother's citizenship (In Alphanumeric).

Return : refer to error code.

5.3.38 (unsigned int) MyKidGetMotherRace:(NSData **) outdata

This function retrieves MyKid mother's race.

Parameter : (NSData **) outData (OUT) – mother's race (In Alphanumeric).

Return : refer to error code.

5.3.39 (unsigned int) MyKidGetFatherID:(NSData **) outdata

This function retrieves MyKid father's identity card number.

Parameter : (NSData **) outData (OUT) – father's identity card number (In Alphanumeric).

Return : refer to error code.

5.3.40 (unsigned int) MyKidGetFOtherDocNo:(NSData **) outdata

This function retrieves MyKid Father's other Doc number

Parameter : (NSData **) outData (OUT) – father's other ID number (In Alphanumeric).

Return : refer to error code.

5.3.41 (unsigned int) MyKidGetFOtherDocType:(NSData **) outdata

This function retrieves MyKid category of father's other doc type.

Parameter : (NSData **) outData (OUT) – category of father's other ID (In Alphanumeric).

Return : refer to error code.

5.3.42 (unsigned int) MyKidGetFatherName:(NSData **) outdata

This function retrieves MyKid father's name.

Parameter : (NSData **) outData (OUT) – father's name (In Alphanumeric).

Return : refer to error code.

5.3.43 (unsigned int) MyKidGetFatherDOB:(NSData **) outdata

This function retrieves MyKid father's date of birth.

Parameter : (NSData **) outData (OUT) – father's date of birth (In Numeric YYYYMMDD).

Return : refer to error code.

5.3.44 (unsigned int) MyKidGetFatherCitizenship:(NSData **) outdata

This function retrieves MyKid father's citizenship.

Parameter : (NSData **) outData (OUT) – father's citizenship (In Alphanumeric).

Return : refer to error code.

5.3.45 (unsigned int) MyKidGetFatherRace:(NSData **) outdata

This function retrieves MyKid father's race.

Parameter : (NSData **) outData (OUT) – mother's race (In Alphanumeric).

Return : refer to error code.

5.3.46 (unsigned int) MyKidGetRegistrationPlace:(NSData **) outdata

This function retrieves MyKid name of JPN branch where birth was registered.

Parameter : (NSData **) outData (OUT) – name of JPN branch where birth was registered (In Alphanumeric).

Return : refer to error code.

5.3.47 (unsigned int) MyKidGetJPN4Version:(NSData **) outdata

This function retrieves MyKid JPN4 version number.

Parameter : (NSData **) outData (OUT) – JPN4 version number (In Numeric).

Return : refer to error code.

5.3.48 (unsigned int) MyKidGetJPN4Size:(NSData **) outdata

This function retrieves MyKid JPN4 size.

Parameter : (NSData **) outData (OUT) – JPN4 size (In Numeric).

Return : refer to error code.

5.3.49 (unsigned int) MyKidGetContactAddress:(NSData **) outdata

This function retrieves MyKid address of contact person

Parameter : (NSData **) outData (OUT) – address of contact person (In Alphanumeric).

Return : refer to error code.

5.3.50 (unsigned int) MyKidGetContactName:(NSData **) outdata

This function retrieves MyKid name of contact person

Parameter : (NSData **) outData (OUT) – name of contact person (In Alphanumeric).

Return : refer to error code.

5.3.51 (unsigned int) MyKidGetContactTelNum:(NSData **) outdata

This function retrieves MyKid telephone number of contact person

Parameter : (NSData **) outData (OUT) – telephone number of contact person (In Alphanumeric).

Return : refer to error code.

6 Appendix 1

6.1 Error Code

Value	Definition	Description
Bluetooth Error Code		
0	BT_OK	No error
-100	BTERR	General Error
-101	BTERR_INVALID_PARAMETER	Invalid Parameter
-102	BTERR_NO_DEV	No BLE device
-103	BTERR_OPEN_DEV_FAIL	Connect BLE device error
-104	BTERR_DISABLED	Bluetooth disabled
-105	BTERR_NOT_CONNECTED	BLE device not connected
-106	BTERR_NO_IO_STREAM	I/O error
-107	BTERR_TIMEOUT	Timeout Error
-108	BTERR_IO	I/O operation error
-109	BTERR_NO_RESP	I/O no response
-110	BTERR_FRAME	I/O Frame error
-111	BTERR_BATT_LOW	Reader battery low
-112	BTERR_SAM_AUTH	SAM authenticate error
Biometric Error Code		
0	BIO_OK	No error
-1	BIOERR_INTERNAL	Biometric device performed an internal error
-2	BIOERR_PROTOCOLE	Internal Communication Protocol error
-3	BIOERR_CONNECT	Cannot connect biometric device
-5	BIOERR_BADPARAMETER	Invalid Parameter
-6	BIOERR_MEMORY_PC	Not enough memory
-7	BIOERR_MEMORY_DEVICE	Not enough memory for the creation of a database
-8	BIOERR_NO_HIT	Authentication or Identification failed
-9	BIOERR_STATUS	Unknown status error
-10	BIOERR_DB_FULL	The database is full
-11	BIOERR_DB_EMPTY	The Database is empty
-12	BIOERR_ALREADY_ENROLLED	User has already been enrolled
-13	BIOERR_BASE_NOT_FOUND	The specified base does not exist
-14	BIOERR_BASE_ALREADY_EXISTS	The specified base already exist

-17	BIOERR_INVALID_TEMPLATE	The template is not valid
-18	BIOERR_NOT_IMPLEMENTED	Command not yet implemented
-19	BIOERR_TIMEOUT	No response after defined time
-26	BIOERR_CMDE_ABORTED	Command has been aborted
-27	BIOERR_INVALID_PK_FORMAT	Invalid PK format
-28	BIOERR_SAME_FINGER	User gave twice the same finger
-30	BIOERR_INVALID_USER_ID	User ID is not valid
-31	BIOERR_INVALID_USER_DATA	The user data are not valid
-32	BIOERR_FIELD_INVALID	Additional field name length invalid
-33	BIOERR_USER_NOT_FOUND	User ID not found in database
-40	BIOERR_DEV_NOT_CONNECTED	Device not connected
-46	BIOERR_FFD	False Finger Detected
-47	BIOERR_MOIST_FINGER	The finger can be too moist or the scanner is wet
Smartcard Error Code		
-150	SCERR	Smart card general error
-151	SCERR_INVALID_PARAMETER	Invalid parameter input to smartcard functions
-152	SCERR_READER_NOT_AVAILABLE	Reader not available upon initialize reader
-153	SCERR_CARD_NO_RSP	Card not response during transceive
-154	SCERR_OPEN_READER_FAIL	Reader open failed during initialize
-155	SCERR_NO_DEV	No smartcard device
-156	SCERR_CARD_REMOVED	Card Removed
-157	SCERR_NO_CARD	Card Absent
-158	SCERR_DEV_NOT_CONNECT	Smart card device not connected
-159	SCERR_READER_UNAVAILABLE	No smartcard reader available
-160	SCERR_SHARING_VIOLATION	Smartcard reader sharing violation
-161	SCERR_TIMEOUT	Smartcard timeout
-162	SCERR_UNRESPONSIVE_CARD	Smartcard no response
-163	SCERR_UNSUPPORTED_CARD	Smartcard not supported
-164	SCERR_UNPOWERED_CARD	Smartcard not powered
-165	SCERR_RESET_CARD	Smartcard reset
-166	SCERR_INVALID_HANDLE	Invalid smartcard handle
-167	SCERR_AUTHENTICATION	Card Authentication failed

6.2 Enrollment and Verification status code

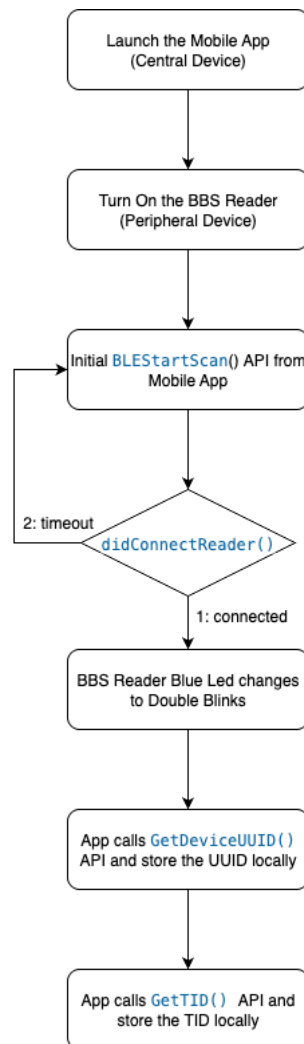
Value	Definition	Description
Biometric Status Code		
0x00	BIO_NO_FINGER	No finger
0x01	BIO_FINGER_UP	Move finger up
0x02	BIO_FINGER_DOWN	Move finger down
0x03	BIO_FINGER_LEFT	Move finger left
0x04	BIO_FINGER_RIGHT	Move finger right
0x05	BIO_FINGER_PRESS_HARDER	Press harder
0x06	BIO_FINGER_LATENT	Latent finger
0x07	BIO_FINGER_REMOVE	Remove finger
0x08	BIO_FINGER_OK	Finger Acquisition OK
0x09	BIO_FINGER_DETECTED	Finger detected
Biometric finger Matching return Code		
0	BIO_FINGER_MATCHED	Finger verification Success
-8	BIO_FINGER_NOT_MATCHED	Finger verification fail
Biometric Enrollment template format		
0x00	PK_COMP_V2	Morpho private format dedicated to terminals
0x02	PK_MAT	Morpho private format dedicated to AFIS
0x03	PK_MAT_NORM	Morpho private format
0x41	ANSI_INCITS_378_2004	ANSI INCITS 378-2004 public finger minutiae record format
0x4C	ISO_19794_2_FMR_2011	ISO/IEC 19794-2 Finger Minutiae Record version 2011
0x4D	ANSI_INCITS_378_2009	ANSI INCITS 378-2009 public finger minutiae record format
0x6C	ISO_19794_2_FMR_CS	ISO/IEC 19794-2 Finger Minutiae Card Record, Compact Size
0x6D	ISO_19794_2_FMR_NS	ISO/IEC 19794-2 Finger Minutiae Card Record, Normal Size
0x6E	ISO_19794_2_FMR	public fingerprint template format, defined by the ISO organization
0x6F	MINEX_A	public fingerprint template format, defined by the NIST organisation for MINEX testing
0x7D	DIN_V66400_FMR	public fingerprint template format, dedicated to use with Smart Cards to perform a Match On Card
0x7E	DIN_V66400_FMR_CS_AA	minutiae ordered by ascending angle

0x7F	ISO_19794_2_FMR_CS_AA	ISO/IEC 19794-2 Finger Minutiae Card Record, Compact Size, minutiae ordered by Ascending Angle

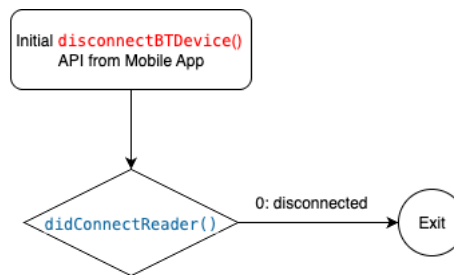
6.3 Workflow for Bluetooth Connection Setup

6.3.1 Workflow for First Time Bluetooth Setup

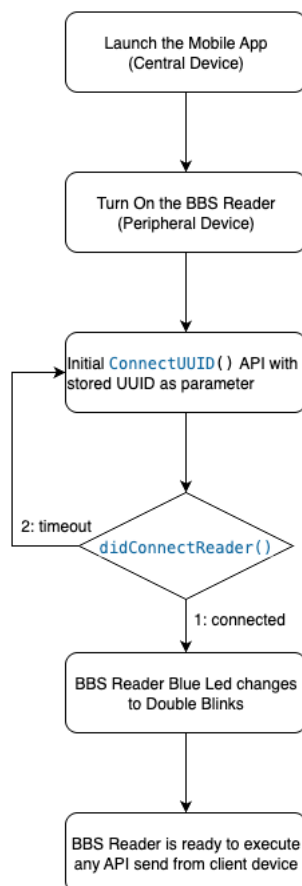
- App needs to store the assigned UUID and TID after central device has established the Bluetooth connection with peripheral device likes BBS Reader.



6.3.2 Workflow for Bluetooth Disconnection



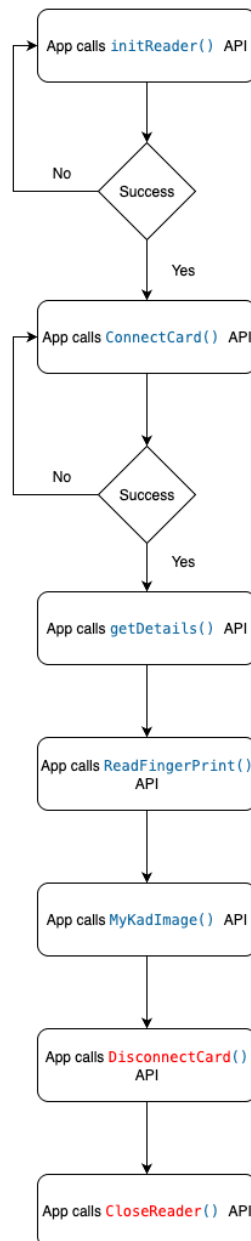
6.3.3 Workflow for Non-First Bluetooth Connection



6.4 Workflow for MyKad Connection Setup

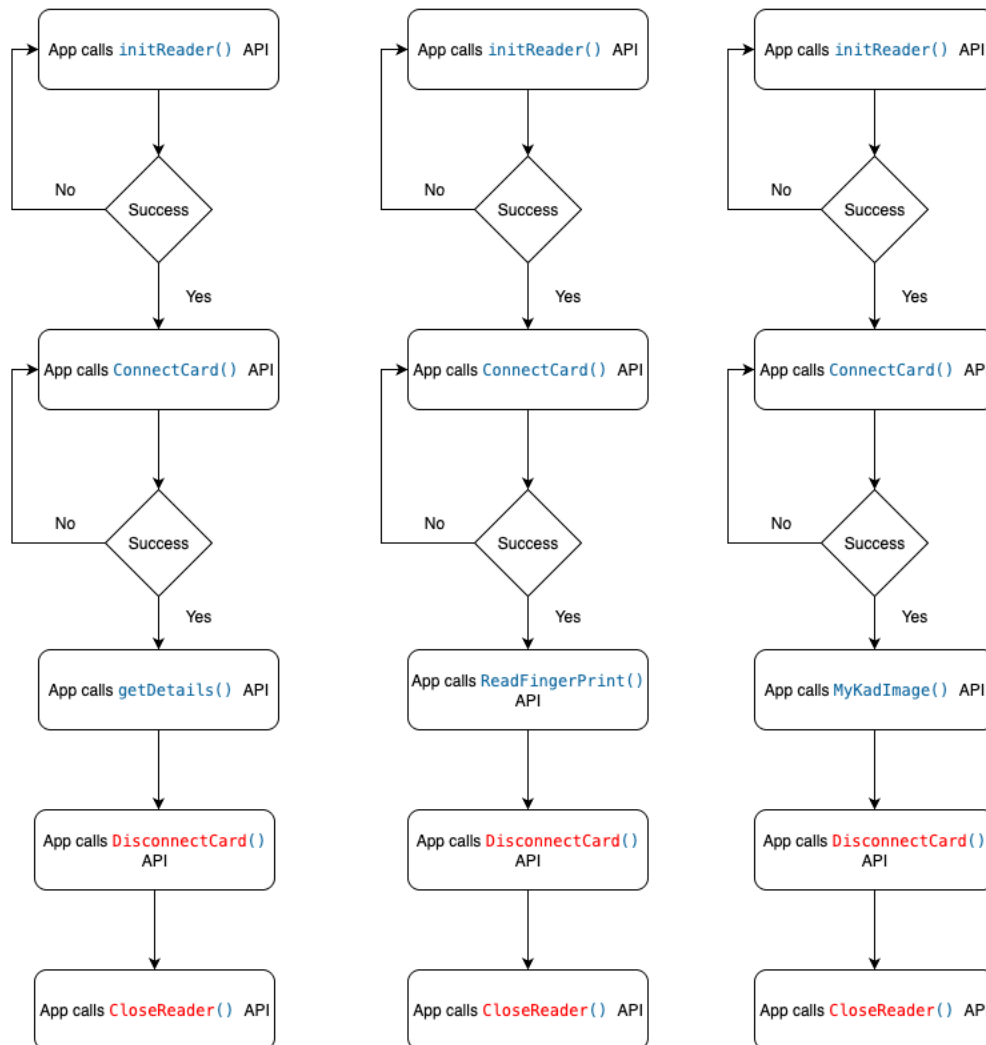
6.4.1 Workflow for block reading MyKad at once includes photo and minutiae.

- User needs to aware this process will take up to 12 seconds for BBS Reader to complete all the tasks.

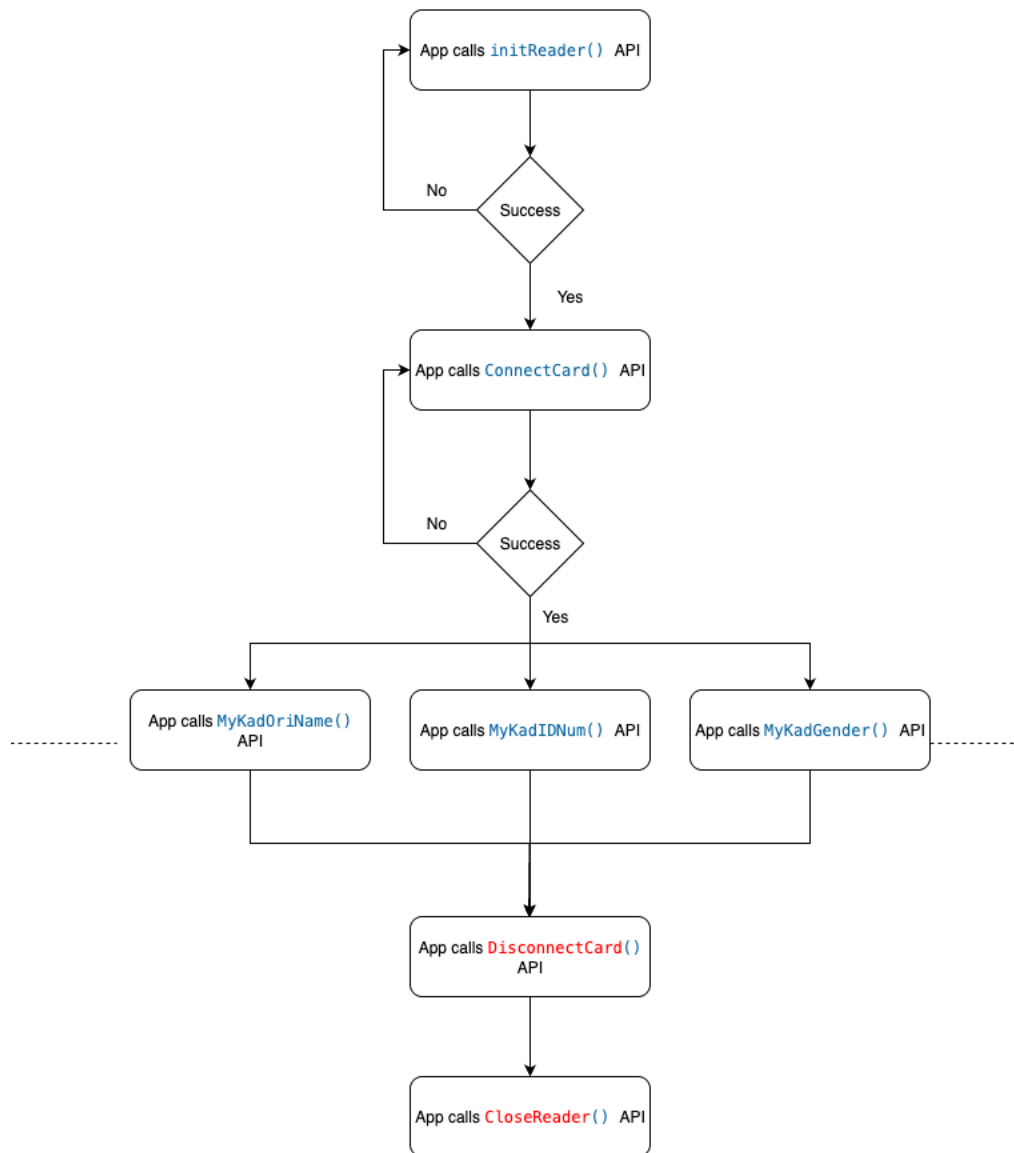


6.4.2 Workflow for block reading MyKad in individual process

- This individual process gives better timing flexibility for the app UI thread to update the current MyKad reading status.



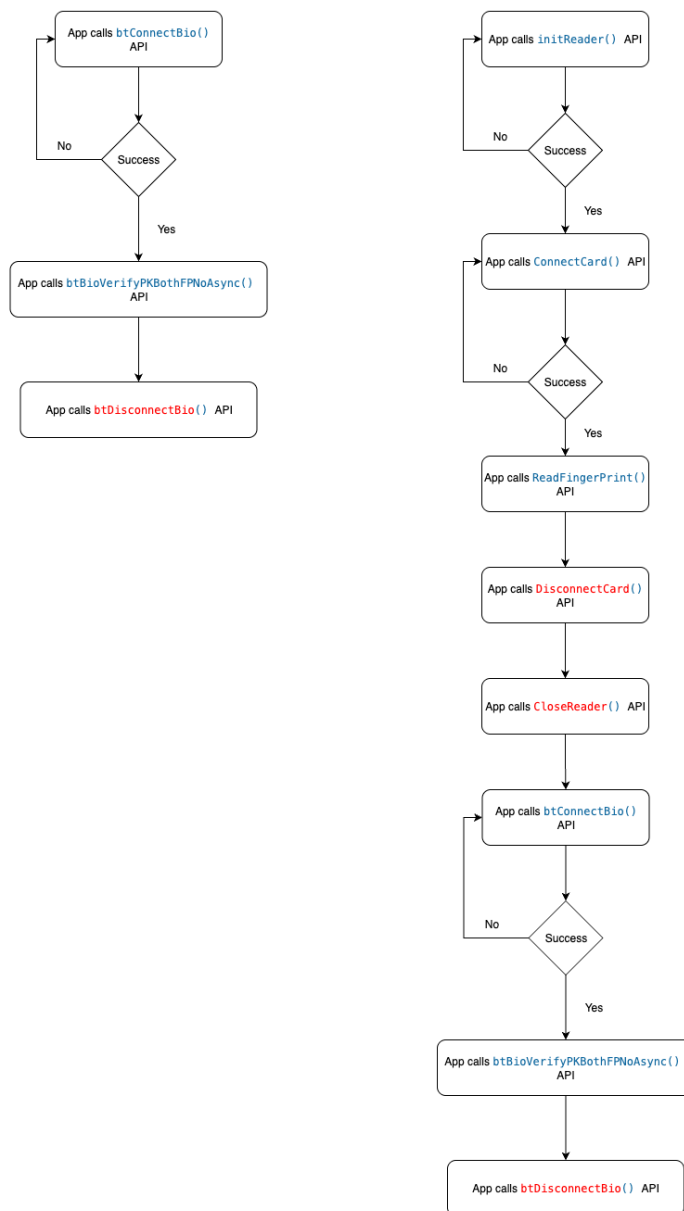
6.4.3 Workflow for single reading MyKad data



6.5 Workflow for Biometric Connection Setup

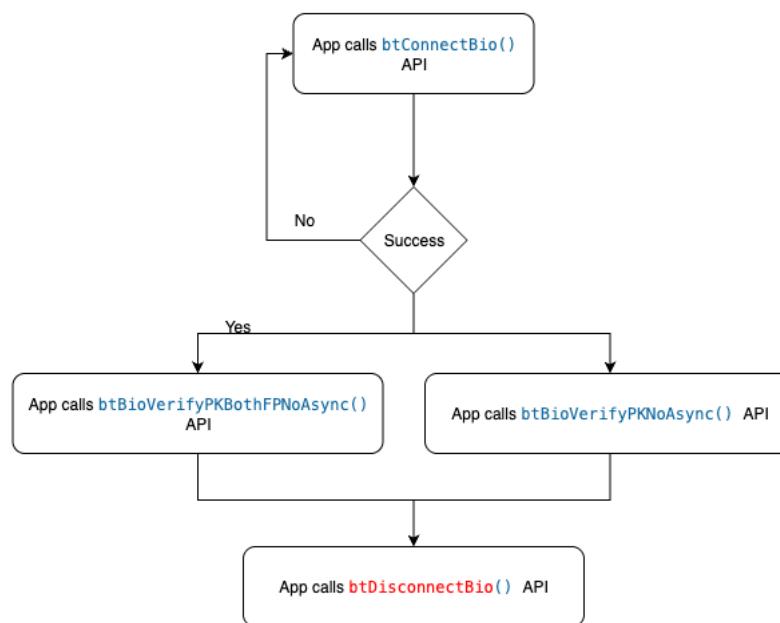
6.5.1 Workflow for Biometric to verify user live fingerprint against both MyKad's minutiae at once.

- The left-side workflow shows it involves only Biometric APIs only. However, developer must ensure the MyKad “**ReadFingerprint ()**” API has been called prior to the workflow below.
- The right-side workflow shows the complete MyKad thumbprint verification process involves MyKad minutiae reading first then follow by Biometric verification.



6.5.2 Workflow for Biometric Verification based on external minutiae

- Developer needs to know the external minutiae format such as ISO19794-2.
- Developer needs to get the complete external minutiae data via Server or Enrollment API.
- `btBioVerifyPKBothFPNoAsync` API is to verify user live fingerprint with 2 external minutiae at once.
- `btBioVerifyPKNoAsync` API is to verify user live fingerprint with single external minutiae at once



7 Abbreviations

Acronym	Description
BLE	Bluetooth Low Energy
BLE Peripheral Device	Peripheral device is a device that advertises its presence and services to other BLE devices, typically in low-power, resource-constrained environment. It acts as a server in the BLE communication framework, waiting for a central device to connect to it. Example: BBS Reader
BLE Central Device	Central device acts as a “client” in a BLE connection, actively scanning for and connecting to nearby “peripheral” devices. Its responsible for initiating communication, managing connections, and typically receiving data from peripheral. Example: Smartphones, tables
UUID	In iOS when working BLE, the Bluetooth UUID is used to identify services and characteristics where Apple provides a unique identifier for each peripheral (like a device).