

API Development Manual:

AMTFaceDevice SDK For Android

API Version: 1.0

Doc Version: 1.0

July 2022

Thank you for choosing our product. Please read the instructions carefully before operation. Follow these instructions to ensure that the product is functioning properly. The images shown in this manual are for illustrative purposes only.



For further details, please visit our Company's website
www.armatura.us.

Copyright © 2022 ARMATURA LLC. All rights reserved.

Without the prior written consent of ARMATURA LLC no portion of this manual can be copied or forwarded in any way or form. All parts of this manual belong to ARMATURA and its subsidiaries (hereinafter the "Company" or "ARMATURA").

Trademark

ARMATURA is a registered trademark of ARMATURA LLC. Other trademarks involved in this manual are owned by their respective owners.

Disclaimer

This manual contains information on the operation and maintenance of the ARMATURA product. The copyright in all the documents, drawings, etc. in relation to the ARMATURA supplied product vests in and is the property of ARMATURA. The contents hereof should not be used or shared by the receiver with any third party without express written permission of ARMATURA.

The contents of this manual must be read as a whole before starting the operation and maintenance of the supplied product. If any of the content(s) of the manual seems unclear or incomplete, please contact ARMATURA before starting the operation and maintenance of the said product.

It is an essential pre-requisite for the satisfactory operation and maintenance that the operating and maintenance personnel are fully familiar with the design and that the said personnel have received thorough training in operating and maintaining the machine/unit/product. It is further essential for the safe operation of the machine/unit/product that personnel have read, understood, and followed the safety instructions contained in the manual.

In case of any conflict between terms and conditions of this manual and the contract specifications, drawings, instruction sheets or any other contract-related documents, the contract conditions/documents shall prevail. The contract specific conditions/documents shall apply in priority.

ARMATURA offers no warranty, guarantee, or representation regarding the completeness of any information contained in this manual or any of the amendments made thereto. ARMATURA does not extend the warranty of any kind, including, without limitation, any warranty of design, merchantability, or fitness for a particular purpose.

ARMATURA does not assume responsibility for any errors or omissions in the information or documents which are referenced by or linked to this manual. The entire risk as to the results and performance obtained from using the information is assumed by the user.

ARMATURA in no event shall be liable to the user or any third party for any incidental, consequential, indirect, special, or exemplary damages, including, without limitation, loss of business, loss of profits, business interruption, loss of business information or any pecuniary loss, arising out of, in connection with, or relating to the use of the information contained in or referenced by this manual, even if ARMATURA has been advised of the possibility of such damages.

This manual and the information contained therein may include technical, other inaccuracies, or typographical errors. ARMATURA periodically changes the information herein which will be incorporated into new additions/amendments to the manual. ARMATURA reserves the right to add, delete, amend, or modify the information contained in the manual from time to time in the form of circulars, letters, notes, etc. for better operation and safety of the machine/unit/product. The said additions or amendments are meant for improvement /better operations of the machine/unit/product and such amendments shall not give any right to claim any compensation or damages under any circumstances.

ARMATURA shall in no way be responsible (i) in case the machine/unit/product malfunctions due to any non-compliance of the instructions contained in this manual (ii) in case of operation of the machine/unit/product beyond the rate limits (iii) in case of operation of the machine and product in conditions different from the prescribed conditions of the manual. The product will be updated from time to time without prior notice. The latest operation procedures and relevant documents are available on <http://www.armatura.us>.

If there is any issue related to the product, please contact us.

ARMATURA Headquarters

Address 190 Bluegrass Valley Pkwy,
 Alpharetta, GA 30005, USA.

For business-related queries, please write to us at info@armatura.us.

To know more about our global branches, visit www.armatura.us.

About the Company

ARMATURA is a leading global developer and supplier of biometric solutions which incorporate the latest advancements in biometric hardware design, algorithm research & software development. ARMATURA holds numerous patents in the field of biometric recognition technologies. Its products are primarily used in business applications which require highly secure, accurate and fast user identification.

ARMATURA biometric hardware and software are incorporated into the product designs of some of the world's leading suppliers of workforce management (WFM) terminals, Point-of-Sale (PoS) terminals, intercoms, electronic safes, metal key lockers, dangerous machinery, and many other products which heavily rely on correctly verifying & authenticating user's identity.

About the Manual

This manual introduces the operations of **AMTFaceDevice SDK For Android**.

All figures displayed are for illustration purposes only. Figures in this manual may not be exactly consistent with the actual products.

Document Conventions

Conventions used in this manual are listed below:

GUI Conventions

For Software	
Convention	Description
Bold font	Used to identify software interface names e.g. OK , Confirm , Cancel .
>	Multi-level menus are separated by these brackets. For example, File > Create > Folder.
For Device	
Convention	Description
< >	Button or key names for devices. For example, press <OK>.
[]	Window names, menu items, data table, and field names are inside square brackets. For example, pop up the [New User] window.
/	Multi-level menus are separated by forwarding slashes. For example, [File/Create/Folder].

Symbols






Convention	Description
	This represents a note that needs to pay more attention to.
	The general information which helps in performing the operations faster.
	The information which is significant.
	Care taken to avoid danger or mistakes.
	The statement or event that warns of something or that serves as a cautionary example.

Table of Contents

1	OVERVIEW	6
1.1	ABOUT THE SDK PRODUCT.....	6
1.2	ADVANTAGES OF THE SDK	6
2	TECHNICAL SPECIFICATIONS	7
2.1	ARCHITECTURE.....	7
2.1.1	PROJECT CONFIGURATION	8
2.2	PROGRAMMING GUIDE	8
2.2.1	MOTION DETECTION PROCESS	8
3	SDK INTERFACE DESCRIPTION	11
3.1	DEVICE INTERFACES.....	11
3.1.1	LOGHELPER.CLASS.....	11
3.2	FACE IMAGE CAPTURING INTERFACES	12
3.2.1	AMTFACECAMERAFACTORY.CLASS	12
3.2.2	AMTFACECAMERA.CLASS.....	14
3.2.3	AMTFACECAMERALISTENER.CLASS.....	22
	APPENDIX.....	24
	APPENDIX 1: PARAMETER CODE DESCRIPTION	24

1 Overview

This document will provide you with basic SDK development guide and technical background to help with better use our AMTFaceDevice SDK document. From the perspective of a developer, the key design objective of this AMTFaceDevice SDK is compatibility and the ease of execution.

This development manual contains the product development documentation for developers that describes the functions provided by the SDK and its related usage, which eases the development environment.

The following sections explains all the required information on how to perform and integrate AMTFaceDevice SDK.

1.1 About the SDK Product

The AMTFaceDevice SDK facilitates in the integration of Facial recognition and authentication of a human face. It ensures system efficiency and performance with live face recognition, simultaneous identification of multiple faces while in motion, and instant face matching. It has consistent recognition on different platforms, where it can provide different customized SDKs to distinctive markets and according to the customer needs.

The simple library components aid in supporting and enhancing the security requirements through biometric facial recognition which avoids spoofing and has been widely used in various systems, including attendance, security, video monitoring and so on.

1.2 Advantages of the SDK

- A simple algorithm for easy integration with existing device terminals.
- Thorough documentation to explain how your code works.
- Enough functionality so it adds value to other applications.
- Supports both visible and infrared light facial recognition method.
- Does not negatively impact.
- Performs well with other SDKs.

2 Technical Specifications

Development Language

This SDK provides a JAR package to support Java development.

Platform Requirements

This SDK must be used on Android 4.1 or later, which supports Android USB Host.

2.1 Architecture

SDK File

Copy the following files in the libs directory to the app libs directory and add the jniLib dependent library to the android node in the gradle file.

```
sourceSets.main
{
    jniLibs.srcDir 'libs'
    jni.srcDirs = []
}
```

File Name	Description
amtandroidcore.jar	Java library for communication media (such as USB and serial port)
amtandroidfacecap.jar	Java library for face capturing
libamtsensorcore.so	Dynamic link library for underlying communication interfaces of the collector
libamtfacecap.so	Dynamic link library for face capturing

2.1.1 Project Configuration

USB information and permission configuration

Face recognition device: See the [Device Introduction](#) for the technical specifications.

Device Name	Vendor ID	Product ID
AMT-FAM-10	0x34c9	0x1121
AMT-FAR-10	0x34c9	0x1181

Permission configuration

See the demo or "Android USB Host Helper.md."

2.2 Programming Guide

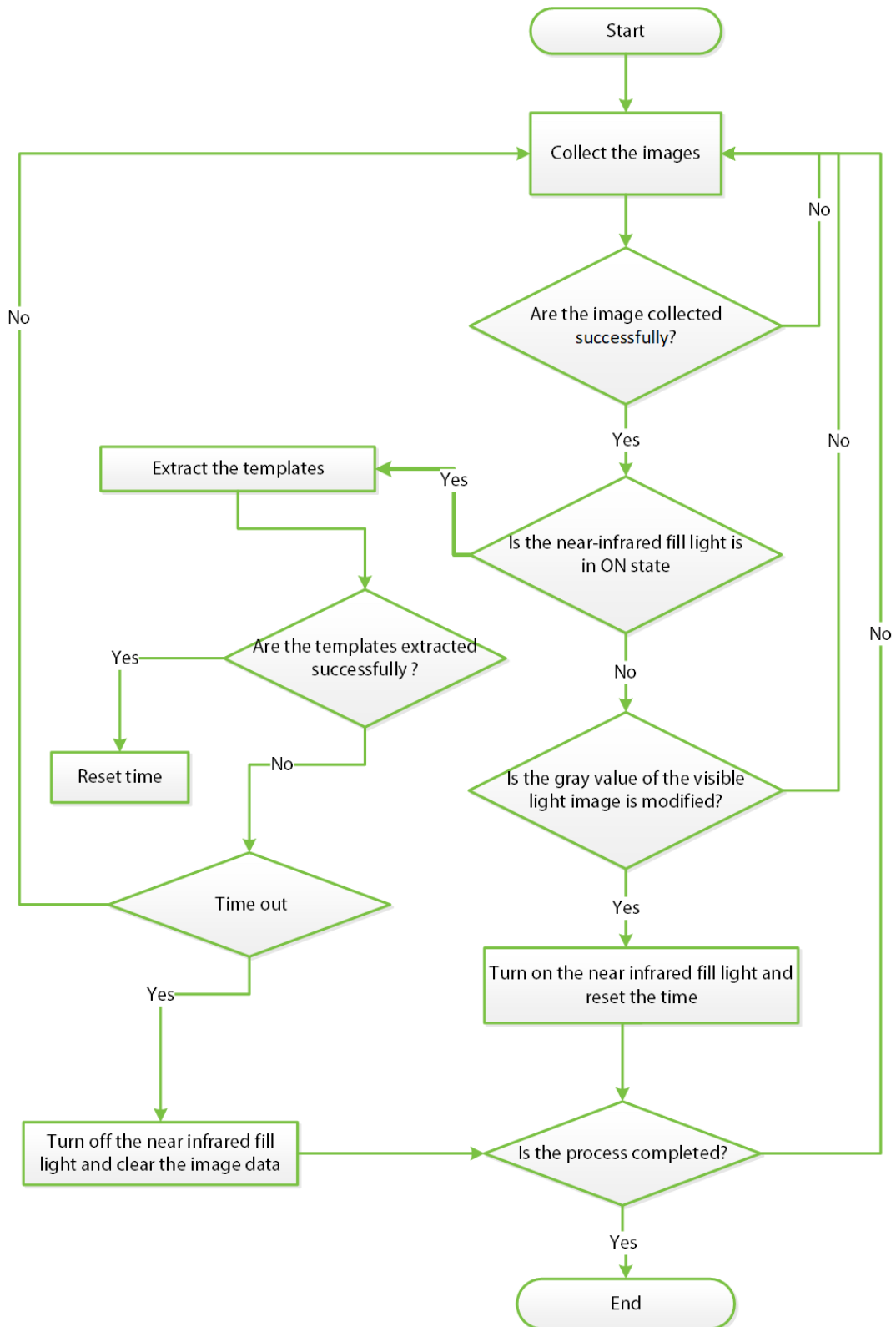
AMTFaceDevice SDK is the device SDK for both near-infrared and visible light face recognition device developed by ARMATURA LLC.

This SDK captures both near-infrared and visible light face images. And it can be used with AMTNIRFace SDK, AMTLiveFace SDK, or with a third-party face recognition SDK to complete integrated development of a face recognition system with high accuracy and large capacity.

2.2.1 Motion Detection Process

This section describes the process of controlling the ON/OFF state of the near-infrared fill light through motion detection based on variations of the average grayscale of visible images and the near-infrared face recognition algorithm.

Process Flow:



Brief description of the process:

- After the face recognition device turns on, the face recognition application calls the **startCapture** function to capture the face images continuously.

Note: The near-infrared fill light will be in OFF state by default.

- The SDK calls the callback function **onCapture** to notify the application that the face images are captured successfully.
- The application checks the status (ON or OFF state) of the near-infrared fill light. If the near-infrared fill light is in OFF state, the application calls the **procMotionDetection** function to calculate the gray scale variation in the visible image data.
- Once the gray scale has changed, the application turns on the near-infrared fill light and resets the timeout timer.
- Then the application calls the face recognition SDK to extract the face template. If the template is extracted successfully, the application resets the timeout timer.
- If the extraction of the templates fails, the application checks whether the timeout timer is expired, (8s for example).
- And if the timeout timer is expired, the application calls the **procMotionDetection** function to reset the motion detection state and turns off the near infrared light.

3 SDK Interface Description

3.1 Device Interfaces

3.1.1 LogHelper.class

Function List

Class/Interface	Description
com.armatura.android.biometric.core.utils.LogHelper.class	Log interface class, used to set the log level
setLevel	Sets the log level

setLevel

Function Syntax

```
public static void setLevel(int level)
```

Description

Sets the log level for the SDK.

Parameters

Parameter	Description
level	In: Log.VERBOSE to Log.ASSERT.

Remarks

Click [here](#) to view the Function List.

3.2 Face Image Capturing Interfaces

3.2.1 AMTFaceCameraFactory.class

It is a factory class used to create as well as destroy the AMTFaceCamera object.

Function List

Class/Interface	Description
com.armatura.android.biometric.device.palmsensor.AMTFaceCameraFactory.class	AMTFaceCamera factory class
createAMTFaceCamera	Creates a AMTFaceCamera object
destroy	Destroys a AMTFaceCamera object

createAMTFaceCamera

Function Syntax

```
public static AMTFaceCamera createAMTFaceCamera
(
    Context context,
    TransportType transportType,
    Map<String, Object> parameters
);
```

Description

Creates a AMTFaceCamera instance

Parameters

Parameter	Description
context	In: Application context (getApplicationContext).
transportType	In: Type of the transfer protocol (TransportType.USB)
parameters	In: USB vendor ID and product ID

Returns

AMTFaceCamera object

Example

```

...
private final static int VID = 0x34c9; //AMT USB vendor ID
private final static int PID = 0x1121; //AMT-FAM-10 USB product ID
...
{
    ...
    // Start face device
    Map deviceParams = new HashMap();
    //set vid
    deviceParams.put(ParameterHelper.PARAM_KEY_VID, VID);
    //set pid
    deviceParams.put(ParameterHelper.PARAM_KEY_PID, PID);
    amtfaceCamera =
    AMTFaceCameraFactory.createAMTFaceCamera(getApplicationContext(),
    TransportType.USB, deviceParams);
    ...
}

```

Remarks

In addition to specifying the VID and PID, you can also enumerate devices with VID=0x34C9 and PID high 16 bits of 0x11 (enumSensor in demo source code)

Click [here](#) to view the Function List.

destroy**Function Syntax**

```
public static void destroy(AMTFaceCamera device)
```

Description

Destroys an object

Parameters

Parameter	Description
-----------	-------------

device	In: AMTFaceCamera object.
--------	---------------------------

Remarks

Click [here](#) to view the Function List.

3.2.2 AMTFaceCamera.class

AMTFaceCamera provides functions such as turning on and off the device, obtaining device parameters, and capturing the face images.

Function List

Class/Interface	Description
com.armatura.biometric.amtfacecamera. AMTFaceCamera.class	Face recognition device interface class
getSDKVersion	Gets the SDK version
open	Turns on the device
close	Turns off the device
getImgWidth	Gets the image width
getImgHeight	Gets the image height
getSerialNumber	Gets the serial number of the device
getFirmwareVersion	Gets the firmware version
setCaptureInterval	Sets the image capturing interval
reset/resetEx	Resets the device
setAMTFaceCameraListener	Sets the callback listener object
startCapture	Starts the image capturing process
stopCapture	Stops the image capturing process
setParameter	Sets the parameters
upgrade	Upgrades the firmware

getSDKVersion

Function Syntax

```
public static String getSDKVersion()
```

Description

Gets the SDK version

Returns

Returns the version number

Remarks

Click [here](#) to view the Function List.

open**Function Syntax**

```
public void open(int index) throws AMTFaceCameraException
```

Description

Turns on the device

Parameters

Parameter	Description
index	In: Device index, always 0.

Remarks

An exception is thrown when the function fails to execute.

Click [here](#) to view the Function List.

close**Function Syntax**

```
public void close(int index) throws AMTFaceCameraException
```

Description

Turns off the device

Parameters

Parameter	Description
index	In: Device index, always 0.

Remarks

An exception is thrown when the function fails to execute.

Click [here](#) to view the Function List.

getImgWidth**Function Syntax**

```
public int getImgWidth()
```

Description

Gets the image width

Returns

Returns the image width

Remarks

Click [here](#) to view the Function List.

getImgHeight**Function Syntax**

```
public int getImgHeight()
```

Description

Gets the image height

Returns

Returns the image height

Remarks

Click [here](#) to view the Function List.

getSerialNumber

Function Syntax

```
public String getSerialNumber()
```

Description

Gets the serial number of the device

Returns

Returns the serial number of the device

Remarks

Click [here](#) to view the Function List.

getFirmwareVersion

Function Syntax

```
public String getFirmwareVersion()
```

Description

Gets the firmware version

Returns

Returns the firmware version

Remarks

Click [here](#) to view the Function List.

setCaptureInterval

Function Syntax

```
public void setCaptureInterval(int captureInterval)
```

Description

Sets the capturing interval

Parameter

Parameter	Description
captureInterval	In: Interval time(Unit: millisecond).

Remarks

Click [here](#) to view the Function List.

reset**Function Syntax**

```
public void reset() throws AMTFaceCameraException
```

Description

Resets the device

Remarks

An exception is thrown when the function fails to execute.

Click [here](#) to view the Function List.

resetEx**Function Syntax**

```
public void resetEx(int index) throws AMTFaceCameraException
```

Description

Resets the device

Parameters

Parameter	Description
index	In: Device index, always 0.

Remarks

This method is similar to the sequence of opening, resetting, and closing functions. An exception is thrown when the function fails to execute.

Click [here](#) to view the Function List.

setAMTFaceCameraListener**Function Syntax**

```
public void setAMTFaceCameraListener(AMTFaceCameraListener listener)
```

Description

Sets the image capturing listener

Parameters

Parameter	Description
listener	In: AMTFaceCameraListener object, as described in AMTFaceCameraListener.class .

Remarks

Click [here](#) to view the Function List.

startCapture**Function Syntax**

```
public void startCapture() throws AMTFaceCameraException
```

Description

Starts the image capturing process ([AMTFaceCameraListener.onCapture](#) is called to notify

the application every time of the successful capturing of the images).

Remarks

An exception is thrown when the function fails to execute.

Click [here](#) to view the Function List.

stopCapture

Function Syntax

```
public void stopCapture() throws AMTFaceCameraException
```

Description

Stops the image capturing process.

Remarks

An exception is thrown when the function fails to execute.

Click [here](#) to view the Function List.

setParameter

Function Syntax

```
public boolean setParameter
(
    int code,
    byte[] value,
    int size
)
```

Description

Sets the parameters

Parameters

Parameter	Description
code	In: Parameter code.(see Parameter Code Description)

value	In: Parameter value
size	In: Length of the parameter value (Size of (int) is 4.)

Returns

Returns True / False.

Remarks

For details about int-to-byte [] conversion of the parameter value, see `ToolUtils.intToByteArray`.

Click [here](#) to view the Function List.

upgrade**Function Syntax**

```
public boolean upgrade(byte[] data, int size)
```

Description

Upgrades the firmware

Parameters

Parameter	Description
data	In: Firmware upgrade package file data
size	In: File data size

Returns

Returns True / False.

Remarks

Upgrading the firmware is expected to take about 20~30s.

Click [here](#) to view the Function List.

3.2.3 AMTFaceCameraListener.class

It is a callback listener class that notifies the application both on the successful image capturing process and the exceptions on the device.

Function List

Class/Interface	Description
com.armatura.android.biometric.device.palmsensor.AMTFaceCameraListener.class	Callback listener class
onCapture	Callback function (bitmap) that indicates a success in image capturing
onException	Callback function that indicates an exception on the device
onCapture2	Callback function that indicates a success in image capturing (RGB888 + Gray256)

onCapture

Function Syntax

```
void onCapture(Bitmap liveFace, Bitmap irFace)
```

Description

Notifies the application of the result after the image capturing succeeds.

Parameters

Parameter	Description
liveFace	In: Visible image
irFace	In: Infrared image

Remarks

Click [here](#) to view the Function List.

onException

Function Syntax

```
void onException(int count)
```

Description

Notifies the application of the exceptions on the device

Parameters

Parameter	Description
count	In: Number of exceptions

Remarks

It is recommended to call the **resetEx** in this function, and then monitor USB unplug/plug notifications to reconnect the device.

Do not call the **stopCapture** function directly in deviceException. Because **stopCapture** will wait until the thread with the deviceException ends. Therefore, it is recommended to create a new thread in order to reconnect the device

Click [here](#) to view the Function List.

onCapture2

Function Syntax

```
void onCapture2(byte[] rgb, byte[] gray)
```

Description

Notifies the application of the result after the image capturing succeeds

Parameters

Parameter	Description
rgb	In: RGB888 visible image data

gray	In: Gray256 near-infrared image data
------	--------------------------------------

Remarks

This interface is reserved and not supported currently.

Click [here](#) to view the Function List.

Appendix

Appendix 1: Parameter Code Description

Parameter code	Data type/length	Attribute	Description
10002	Int/4Bytes	W	Backlight switch control
10107	Int/4Bytes	W	Set the camera exposure value

Remarks

When setting the backlight, the parameter value types are described as follows:

Parameter Value (paramValue)	Data type/length	Description
0	Int/4Bytes	Turn off all status lights
1	Int/4Bytes	Turn on the near infrared light
2	Int/4Bytes	Turn on the visible light

When setting the camera exposure value, the parameter value is returned from the function(`AMTNIRFaceService12.extractFromBitmap/extractFromNV21/extractFromGrayscale Data`).

190 Bluegrass Valley Pkwy,

Alpharetta, GA 30005, USA

E-mail: info@armatura.us

www.armatura.us



Copyright © 2022 ARMATURA LLC. All Rights Reserved.