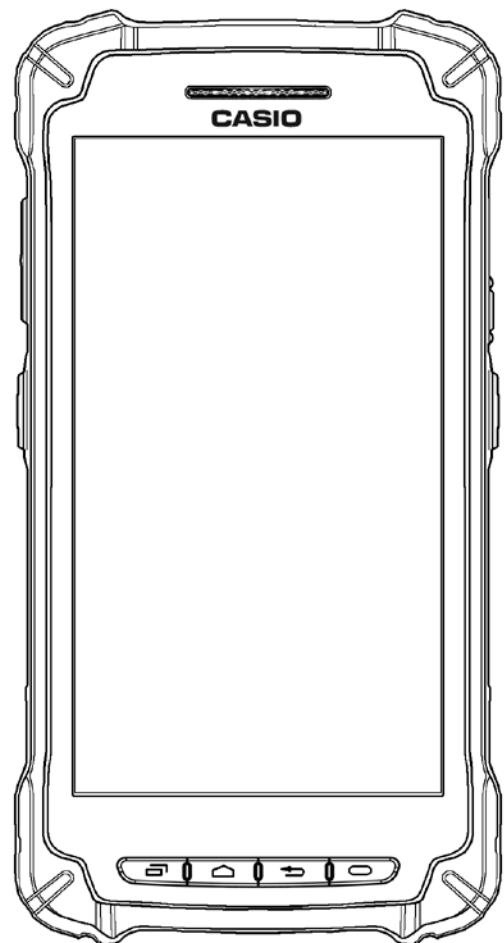




IT-G400 Series

Android 6.0 Device Library Manual

This document explains how to use the IT-G400 device library.



No part of this document may be produced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of CASIO Computer Co., Ltd. in Tokyo Japan. Information in this document is subject to change without advance notice.

CASIO Computer Co., Ltd. makes no representations or warranties with respect to the contents or use of this manual and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

© 2018 CASIO Computer Co.,Ltd.

- The Bluetooth word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by CASIO COMPUTER CO.,LTD. is under license. Other trademarks and trade names are those of their respective owners.
- Wi-Fi is a registered trademark of Wi-Fi Alliance.
- Android, Android Wear, Google, Google Play, Google Now and other marks are trademarks of Google Inc.
- Other company, product and service names used in this manual also may be trademarks or registered trademarks of others.

- table of contents -

1. Overview	4
1.1 Purpose	4
1.2 System Architecture	5
1.3 Application Development	6
1.3.1 Required privileges for applications	6
1.3.2 API Level	6
1.3.3 Access from multipul applications	7
2. Library Structure	8
3. System	9
3.1 Overview	9
3.1.1 Member list	9
3.2 SystemLibrary	10
3.2.1 getCASIOSerial	10
3.2.2 getModelName	10
4. Key	11
4.1 Overview	11
4.1.1 List of Member Functions	12
4.2 KeyLibrary	13
4.2.1 setUserKeyCode	13
4.2.2 getUserKeyCode	13
4.2.3 setDefaultKeyCode	14
4.2.4 setLaunchApplication	14
4.2.5 getLaunchApplication	15
4.2.6 ApplicationInfo class	15
4.2.7 clearLaunchApplication	16
4.3 Constants	17
4.3.1 List of Constants	17
5. Barcode Scanner	18
5.1 Overview	18
5.1.1 List of Member Functions	19
5.2 ScannerLibrary	21
5.2.1 openScanner	21
5.2.2 closeScanner	21
5.2.3 isScannerOpen	21
5.2.4 setDefaultAll	22
5.2.5 getAPIVersion	22
5.2.6 getModuleVersion	22
5.2.7 getScanResult	23

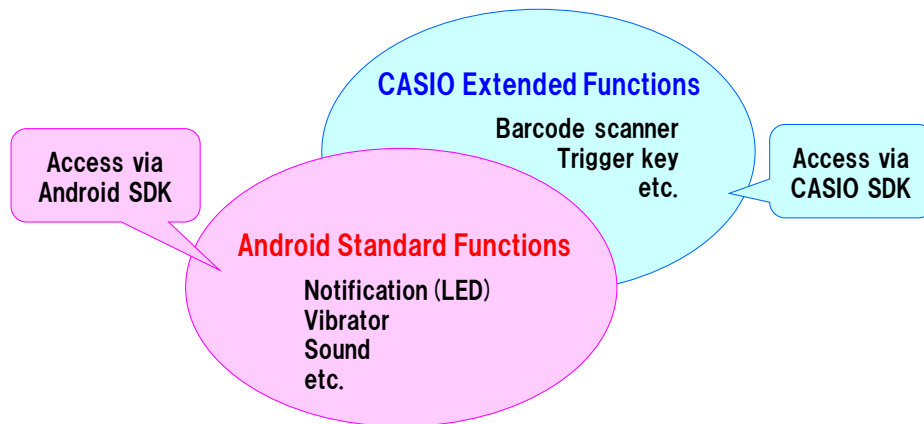
5.2.8	ScanResult class	23
5.2.9	setNotificationLED	24
5.2.10	getNotificationLED	24
5.2.11	setNotificationVibrator	25
5.2.12	getNotificationVibrator	25
5.2.13	setNotificationSound	26
5.2.14	getNotificationSound	26
5.2.15	setLightMode	27
5.2.16	getLightMode	27
5.2.17	getImageDataSize	28
5.2.18	captureImage	28
5.2.19	getStreamDataSize	29
5.2.20	startStream	29
5.2.21	readStream	30
5.2.22	stopStream	30
5.2.23	setSymbologyEnable	31
5.2.24	getSymbologyEnable	31
5.2.25	getSymbologyMaxDefault	32
5.2.26	getSymbologyMinDefault	32
5.2.27	setSymbologyMax	33
5.2.28	getSymbologyMax	33
5.2.29	setSymbologyMin	34
5.2.30	getSymbologyMin	34
5.2.31	setSymbologyProperty	35
5.2.32	getSymbologyProperty	36
5.2.33	setOutputType	37
5.2.34	getOutputType	37
5.2.35	setSuffix	38
5.2.36	getSuffix	38
5.2.37	setInverseMode	39
5.2.38	getInverseMode	39
5.2.39	setTriggerKeyEnable	40
5.2.40	getTriggerKeyEnable	40
5.2.41	setTriggerKeyMode	41
5.2.42	getTriggerKeyMode	41
5.2.43	setTriggerKeyTimeout	42
5.2.44	getTriggerKeyTimeout	42
5.2.45	setTriggerKeyOn	43
5.2.46	setScannerAPO	44
5.2.47	getScannerAPO	44
5.2.48	setCenteringWindow	45
5.2.49	getCenteringWindow	46

5.2.50	setDetectionAreaSize	46
5.2.51	getDetectionAreaSize	47
5.2.52	setInternalParameter	47
5.2.53	getInternalParameter	47
5.3	CONSTANTS	48
5.3.1	List of constants	48
5.3.2	Code identification table	53
5.3.3	Reading digits	55
5.3.4	Property of barcode	56
5.4	Basic flow of scanning application	59
5.4.1	Barcode reading procedure	59
5.4.2	Streaming flow	60
5.4.3	Shooting still images	61

1. Overview

1.1 Purpose

CASIO Android-equipped handy terminal for rugged smart device (hereinafter referred to as CASIO rugged smart device) has extended functions for business such as barcode scanner and trigger key, in addition to Android standard function.



This device library (hereinafter referred to as this library) is used to call the CASIO extension function.

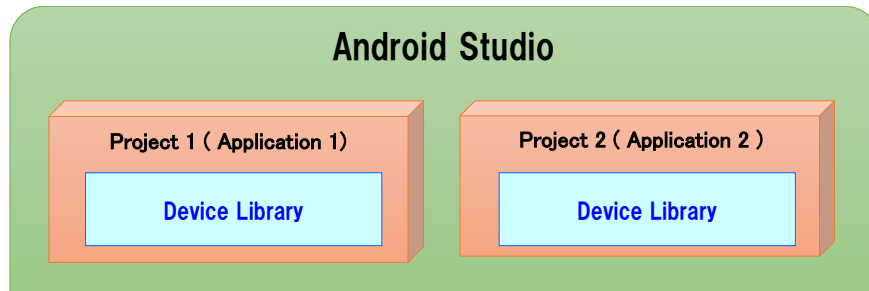
This library is defined as the standard specification of future CASIO rugged smart devices. However, there are cases where future models are not equipped with the same device. Read the explanation of each chapter for the method of developing a common application that will operate in the future CASIO rugged smart device.

The target of this library is a Java application that runs on Android.

1.2 System Architecture

This library is provided in the AAR (Android Archive) file format used in Android Studio. Import this library for each project created in Android Studio.

File name : DeviceLibrary.aar



1.3 Application Development

1.3.1 Required privileges for applications

Android permission necessary to use this library is unnecessary unless it is specified in the individual function.

1.3.2 API Level

When creating Android projects, need to specify "minimum API level" and "target API level". The minimum API level is a setting related to build and installation and depends on this library. The target API level is a setting related to the operation of Android API and does not depend on this library.

This library is created as minimum API level 17 considering portability of existing applications. When creating a project, specify the following API level.

API Level	Settings
Minimum API level	17 to 23
Target API level	Application developer should set appropriate level.

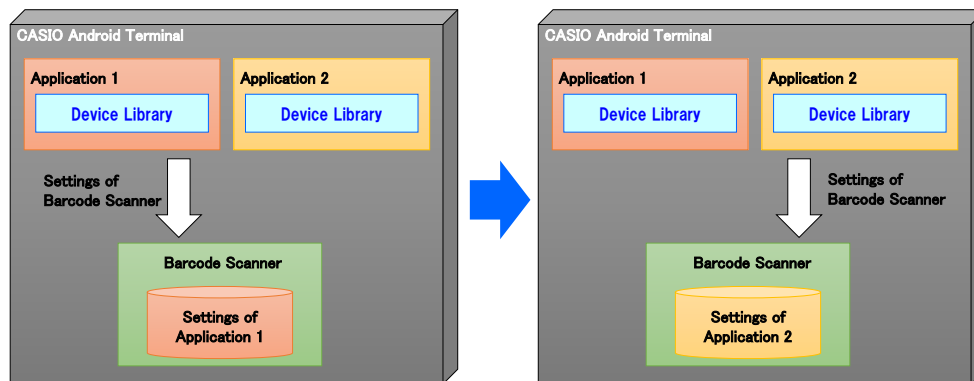
Cautions!

- If you set the project API level to 16 or less, a build error occurs.
- If you set the API level of the project to 24 or more, it can not be installed on the IT-G400.

1.3.3 Access from multipul applications

By using this library, you can change the barcode scanner, key code setting and so on. This library can be called from multiple applications at the same time, but the contents set later will always be valid.

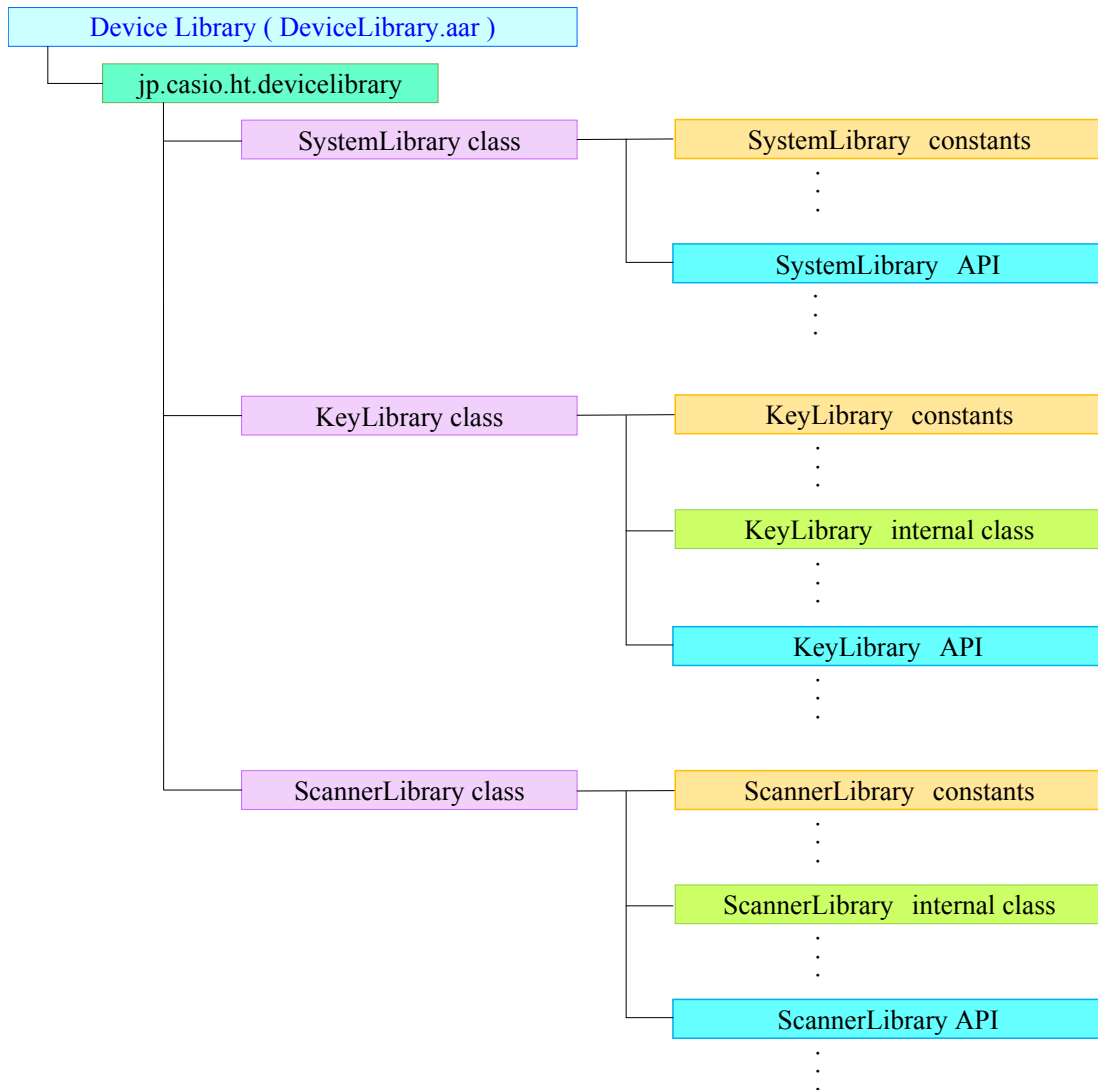
For example, if the APP-2 further changes the setting after the APP-1 changes the setting of the barcode scanner, the setting of the barcode scanner becomes the content set by the APP-2.



The contents set in this library are retained even when the terminal is restarted. For details on how to restore the settings, read the explanation of each chapter.

2. Library Structure

This library consists of the following classes.



Class name	Description
SystemLibrary	Class controlling system settings
KeyLibrary	Class controlling keys
ScannerLibrary	Class controlling barcode scanner

3. System

3.1 Overview

SystemLibrary class is used to get and set system information.

To use the SystemLibrary class in the source code, import the class then create the instance as follows.

Import : import jp.casio.ht.devicelibrary.SystemLibrary

Instance : SystemLibrary mSystemLibrary = new SystemLibrary();

3.1.1 Member list

System information acquisition function	
getCASIOSerial	Get serial ID of terminal

3.2 SystemLibrary

3.2.1 getCASIOSerial

String getCASIOSerial (void)

Function:

Get the barcode serial on the back of the terminal. This serial is for CASIO to manage the terminal.

Use the Android build serial for licensing management of applications such as copy protection. The Android build serial can be obtained by SERIAL of the member of the android.os.Build class.

For the Android standard keycode, refer to the Android.os.Build class from the Android developer site reference.

Parameter:

void None

Return value:

String Barcode serial

Returns barcode serial on success and null on failure.

3.2.2 getModelName

String getModelName (void)

Function:

Get the model name (e.g. IT-G400-C21M) of the device.

Parameter:

void None

Return value:

String Model name

Returns barcode serial on success and null on failure.

4. Key

4.1 Overview

The KeyLibrary class is used to change the key code generated when a key is pressed, or to register an application to be launched when a key is pressed.

Functions	Description
Keycode setting	Set an arbitrary key code for each key.
Keycode disabling	By setting a blank key for each key, make the key inactive.
Register app to key	Register an application to be launched to key.

The settable keys are as follows. Please specify the setting target key with KEYID of each key. If an unsupported KEYID is specified, each function returns `ERROR_NOTSUPPORTED` error.

Key	IT-G400
Right Trigger key	yes
Left Trigger key	yes
App Switch key	no
Back key	no
Function key	yes
VolumeUp key	no
VolumeDown key	no
Home key	no

yes : supported, no : not supported

The contents set using this class affect the entire system and it is effective until you clear the setting or initialize the terminal. When you change the setting from multiple applications, the last change made is valid.

To use the KeyLibrary class in your application, import the class and create the instance as follows:

```
Import : import jp.casio.ht.devicelibrary.KeyLibrary  
Instance : KeyLibrary mKeyLibrary = new KeyLibrary();
```

4.1.1 List of Member Functions

Functions for Keycode settings	
setUserKeyCode	Set key code to the specified key.
getUserKeyCode	Acquires key code of the specified key.
setDefaultKeyCode	Return the key code of the specified key to default.
Function for Application settings	
setLaunchApplication	Assign the application to be activated to the specified key.
getLaunchApplication	Acquires information on the application assigned to the specified key.
clearLaunchApplication	Clears the application assigned to the specified key.

4.2 KeyLibrary

4.2.1 setUserKeyCode

int setUserKeyCode(int keyID , int KeyCode)

Functions:

Set the key code specified by KeyCode to the key specified by keyID.

Parameters:

int keyID KEYID

Specify KEYID of setting target key

For details of KEYID of each key, see "4.3 Constants (p.17)".

int KeyCode Keycode

Specify key code to set. To set the Android standard key code, specify key code of "Android's KeyEvent class".

For the details, refer to the `Android.view.KeyEvent` class from the Android Developer Site Reference.

To use the key code added by CASIO such as the trigger key, specify the key code of "KeyLibrary class"

For details of "4.3 Constants (p.17)".

Return value:

It returns SUCCESS on success, and returns an error code on failure.

For details of the return value, see "4.3 Constants (p.17)".

4.2.2 getUserKeyCode

int getUserKeyCode(int keyID)

Functions:

Get the key code assigned to the key specified by keyID.

Parameters:

int keyID KEYID

Specify KEYID of target key.

For details of KEYID of each key, see "4.3 Constants (p.17)".

Return value:

It returns keycode on success, and returns an error code on failure.

For details of the return value, see "4.3 Constants (p.17)".

4.2.3 setDefaultKeyCode

int setDefaultKeyCode(int keyID)

Functions:

Return key code assignment of the key specified by keyID to default.

Parameters:

int keyID KEYID

Specify the setting target key with KEYID.

For details of KEYID, refer to "4.3 Constants (p.17)".

Return value:

It returns SUCCESS on success, and returns an error code on failure.

For details of the return value, see "4.3 Constants (p.17)".

4.2.4 setLaunchApplication

int setLaunchApplication(int keyID , KeyLibrary.LaunchApplication launchApplication)

Functions:

Assign the application specified by launchApplication to the key specified by keyID.

If you assign an application with this function, the keycode assigned to the key is not issued. If you clear an assigned application, the key code assigned to the key will be issued again.

Parameters:

int keyID KEYID

Specify the setting target key with KEYID.

For details of KEYID, refer to "4.3 Constants (p.17)".

KeyLibrary.LaunchApplication launchApplication : Application to be launched

For the member variable of LaunchApplication class, refer to "4.2.6 ApplicationInfo class (p.15)".

Return value:

It returns keycode on success, and returns an error code on failure.

For details of the return value, see "4.3 Constants (p.17)".

4.2.5 getLaunchApplication

```
int getLaunchApplication(int keyID , KeyLibrary.LaunchApplication  
launchApplication )
```

Functions:

Get the application assigned to the key specified by keyID to launchApplication.
When not set, it retrieves the null character.

Parameters:

int keyID KEYID

Specify the target key by KEYID.

For details of KEYID, refer to "4.3 Constants (p.17)".

KeyLibrary.LaunchApplication launchApplication : Application assigned.

For the member variable of LaunchApplication class, refer to "4.2.6

ApplicationInfo class (p.15)".

Return value:

It returns keycode on success, and returns an error code on failure.

For details of the return value, see "4.3 Constants (p.17)".

4.2.6 ApplicationInfo class

```
static class ApplicationInfo
```

Description:

Internal class that stores application information to assign to the key.

Create an instance as follows.

```
KeyLibrary.ApplicationInfo applicationInfo = new KeyLibrary.ApplicationInfo();
```

■ Member variables:

String packageName

When assigning, specify the package name of the application to be started up.

When acquiring, the package name of the allocated application is acquired.

(Example) In case of setting "Android setting",

```
launchApplication.packageName = "com.android.settings";
```

■ Member variables:

String `activityName`

When setting, specify the activity name of the application to be assigned.

When acquiring, the activity name of the assigned application is acquired.

Activity name is specified as "package name.activity name".

(Example) In case of setting "Settings app" of Android.

```
launchApplication.activityName = "com.android.settings.Settings"
```

4.2.7 `clearLaunchApplication`

```
int clearLaunchApplication(int keyID)
```

Functions:

Clears the application assigned to the key specified by `keyID`.

Parameters:

int `keyID` `KEYID`

The key to be cleared is specified by `KEYID`.

For the key `KEYID` of each key, refer to "4.3 Constants (p.17)".

Return value:

It returns `keycode` on success, and returns an error code on failure.

For details of the return value, see "4.3 Constants (p.17)".

4.3 Constants

4.3.1 List of Constants

Return values

List of return value constants. Use to verify the result returned from member function.

Class name : KeyLibrary.CONSTANT.RETURN

Constant field	Description
SUCCESS	Finished normally.
ERROR_NOTSUPPORTED	Unsupported error. It returns in case the specified KEYID is not supported.
ERROR_FUNCTION	Internal error. It returns if an error occurred internally.

KEYID

List of KEYID set for each key.

Class name : KeyLibrary.CONSTANT.KEYID

Constant field	Description
RIGHTTRIGGER	KEYID of Right Trigger key.
LEFTTRIGGER	KEYID of Left Trigger key.
APPSWITCH	KEYID of App switch key.
BACK	KEYID of Back key
FUNCTION	KEYID of Function key
VOLUMEUP	KEYID of Volume Up key
VOLUMEDOWN	KEYID of Volume Down key

Keycode constants

Keycode defined by CASIO.

Class name : KeyLibrary.CONSTANT.KEYCODE

Constant field	Description
KEYCODE_TRIGGER_RIGHT	Keycode of Right Trigger key
KEYCODE_TRIGGER_LEFT	Keycode of Left trigger key
KEYCODE_BLANK	Keycode of black key. It is used to disable keycode of the specified key.

5. Barcode Scanner

5.1 Overview

The barcode scanner provides barcode reading (decoding), still image shooting, streaming function, etc.

The barcode scanner library is used to acquire / change barcode scanner settings.

To maintain compatibility of the source code, the barcode scanner library also provides functions that are not functionally supported, but calling this will return "unsupported error" or "parameter error". By judging these return values correctly, create compatible application.

In addition, the setting of the barcode scanner is shared by all applications. If you change the settings of the barcode scanner with multiple applications, the settings you made later are valid. When multiple applications use the barcode scanner in parallel, it is necessary to reconfigure them according to the purpose at the time of application startup and active, and be careful not to be affected by other applications.

The functions temporarily used in the application (trigger key enable / disable, trigger key mode) are restored to their default values after rebooting. To use these settings, please set it at application startup. Also, these settings will be restored to their default values even when the device is rebooting after backup / restore.

To use the ScannerLibrary class, import the class and create instance as follows.

```
Import      : import jp.casio.ht.devicelibrary.ScannerLibrary  
Instance    : ScannerLibrary mScannerLibrary = new ScannerLibrary();
```

5.1.1 List of Member Functions

Basic functions	
openScanner	Open barcode scanner
closeScanner	Close barcode scanner
isScannerOpen	Check the barcode scanner is opened.
setDefaultAll	Return all barcode scanner settings to default
getAPIVersion	Get API version
getModuleVersion	Get module version
getScanResult	Get last scan result
Notification functions	
setNotificationLED	Set notification LED behavior
getNotificationLED	Get notification LED behavior
setNotificationVibrator	Set notification Vibrator behavior
getNotificationVibrator	Get notification Vibrator behavior
setNotificationSound	Set notification sound behavior
getNotificationSound	Get notification sound behavior
Lighting functions	
setLightMode	Set Light mode
getLightMode	Get Light mode
Image capturing functions	
getImageDataSize	Get data size needed for capturing image.
captureImage	Capture image
getStreamDataSize	Get data size needed for streaming
startStream	Start streaming
readStream	Read streaming data
stopStream	Stop streaming
Detailed setting functions	
setSymbologyEnable	Set reading enable/disable of specified barcode
getSymbologyEnable	Get reading enable/disable of specified barcode
getSymbologyMaxDefault	Get default reading maximum number of digits of specified barcode
getSymbologyMinDefault	Get default reading minimum number of digits of specified barcode
setSymbologyMax	Set reading maximum number of digits of specified barcode
getSymbologyMax	Get reading maximum number of digits of specified barcode
setSymbologyMin	Set reading minimum number of digits of specified barcode
getSymbologyMin	Get reading minimum number of digits of specified barcode
setSymbologyProperty	Set value of property setting of specified barcode
getSymbologyProperty	Get value of property setting of specified barcode
setOutputType	Set output type of scan result
getOutputType	Get output type of scan result
setSuffix	Set suffix type added at the end of the scan result
getSuffix	Get suffix type added at the end of the scan result
setInverseMode	Set inverse barcode reading mode
getInverseMode	Get inverse barcode reading mode
setTriggerKeyEnable	Set trigger key enable / disable
getTriggerKeyEnable	Get trigger key enable / disable
setTriggerKeyMode	Set trigger key mode
getTriggerKeyMode	Get trigger key mode
setTriggerKeyTimeout	Set trigger key timeout
getTriggerKeyTimeout	Get trigger key timeout

setTriggerKeyOn	Software control of Trigger key behavior
setScannerAPO	Set Auto Power Off time of barcode scanner
getScannerAPO	Get Auto Power Off time of barcode scanner
setCenteringWindow	Set enable / disable centering window mode
getCenteringWindow	Get enable / disable centering window mode
setDetectionAreaSize	Set the size of Barcode detection area for the centering window mode
getDetectionAreaSize	Get the size of Barcode detection area for the centering window mode
setInternalParameter	Set internal parameters
getInternalParameter	Get internal parameters

5.2 ScannerLibrary

5.2.1 openScanner

int openScanner(void)

Function:

Open the barcode scanner.

Call this function when the application starts.

If you call Scanner Library's function before open the barcode scanner, it may not work correctly.

Parameter:

void None

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.2 closeScanner

int closeScanner(void)

Function:

Close the barcode scanner.

Call this function when the application terminates.

Parameter:

void None

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.3 isScannerOpen

boolean isScannerOpen(void)

Function:

Check the barcode scanner is opened.

Parameter:

void None

Return value:

true : The barcode scanner is opened

false : The barcode scanner is closed

5.2.4 **setDefaultAll**

int setDefaultAll(void)

Function:

Return all barcode scanner settings to default.

Parameter:

void None

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.5 **getAPIVersion**

String getAPIVersion(void)

Function:

Get API version.

Parameter:

void None

Return value:

String API version

Returns the API version on success and Null on failure.

5.2.6 **getModuleVersion**

String getModuleVersion(void)

Function:

Get module version.

Parameter:

void None

Return value:

String Module version

Returns the module version on success and Null on failure.

5.2.7 getScanResult

```
int getScanResult(ScannerLibrary.ScanResult scanResult)
```

Function:

Get the last Scan Result.

When you read multiple barcodes, you can get the final Scan Result.

When you fail scanning, you can get all of data are cleared except scan time.

When you call this function before scanning ever, you can get all of data are cleared.

For the default value, refer to "5.2.8 ScanResult class (p. 23)".

Parameter:

scanResult ScanResult class

Specify ScannerLibrary.ScanResult class that store Scan Result.

For the member variable of the ScanResult class, refer to "5.2.8 ScanResult class (p. 23)".

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: unsupported error

5.2.8 ScanResult class

```
static class ScanResult
```

Overview:

Inner class that stores the reading result.

Create an instance as follows.

```
ScannerLibrary.ScanResult mScanResult = new ScannerLibrary.ScanResult();
```

■ Member variables:

int length (default 0)	Length of barcode
int time (default 0)	Reading time
byte[] value (default null)	Scanned barcode data
byte aimID (default 0)	AIM ID of the barcode
byte aimModifier (default 0)	AIM ID modifier of the barcode
int symbologyID (default 0)	Symbol ID of the barcode
String symbologyName (default null)	Symbol name of the barcode

5.2.9 setNotificationLED

int setNotificationLED(int led)

Function:

Set notification LED behavior

Parameter:

Led - Notification LED behavior

ScannerLibrary.CONSTANT.NOTIFICATION.LED_ON: Notification LED not light up in case of success and fail scanning

ScannerLibrary.CONSTANT.NOTIFICATION.LED_OFF: Notification LED turns on BLUE in case of success scanning, and RED in case of fail scanning

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.10 getNotificationLED

int getNotificationLED(void)

Function:

Get notification LED behavior

Parameter:

void - None

Return value:

ScannerLibrary.CONSTANT.NOTIFICATION.LED_ON(default)

: Notification LED turns on BLUE in case of success scanning, and RED in case of fail scanning

ScannerLibrary.CONSTANT.NOTIFICATION.LED_OFF

: Notification LED not light up in case of success and fail scanning

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.11 setNotificationVibrator

int setNotificationVibrator(int vibrator)

Function:

Set notification Vibrator behavior

Parameter:

Vibrator – Notification vibrator behavior

ScannerLibrary.CONSTANT.NOTIFICATION.VIBRATOR_ALL_OFF

: Vibrator does not vibrate after success and fail scanning

ScannerLibrary.CONSTANT.NOTIFICATION.VIBRATOR_FAIL_ON

: Vibrator vibrates in case of fail scanning

ScannerLibrary.CONSTANT.NOTIFICATION.VIBRATOR_SUCCESS_ON

: Vibrator vibrates in case of success scanning

ScannerLibrary.CONSTANT.NOTIFICATION.VIBRATOR_ALL_ON

: Vibrator vibrates in case of success and fail scanning

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.12 getNotificationVibrator

int getNotificationVibrator(void)

Function:

Get Notification Vibrator behavior

Parameter:

Void - none

Return value:

ScannerLibrary.CONSTANT.NOTIFICATION.VIBRATOR_ALL_OFF(default)

: Vibrator does not vibrate after success and fail scanning

ScannerLibrary.CONSTANT.NOTIFICATION.VIBRATOR_FAIL_ON

: Vibrator vibrates in case of fail scanning

ScannerLibrary.CONSTANT.NOTIFICATION.VIBRATOR_SUCCESS_ON

: Vibrator vibrates in case of success scanning

ScannerLibrary.CONSTANT.NOTIFICATION.VIBRATOR_ALL_ON

: Vibrator vibrates in case of success and fail scanning

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.13 setNotificationSound

int setNotificationSound(int sound)

Function:

Set Notification Sound behavior

Parameter:

Sound - Notification sound behavior

ScannerLibrary.CONSTANT.NOTIFICATION.SOUND_ALL_OFF

: Notification sound not sound in case of success and fail scanning

ScannerLibrary.CONSTANT.NOTIFICATION.SOUND_FAIL_OFF

: Notification sound played in case of fail scanning

ScannerLibrary.CONSTANT.NOTIFICATION.SOUND_SUCCESS_ON

: Notification sound played in case of success scanning

ScannerLibrary.CONSTANT.NOTIFICATION.SOUND_ALL_ON

: Notification sound played in case of success and fail scanning

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.14 getNotificationSound

int getNotificationSound(void)

Function:

Get Notification Sound behavior

Parameter:

Void - none

Return value:

ScannerLibrary.CONSTANT.NOTIFICATION.SOUND_ALL_OFF

: Notification sound not sound in case of success and fail scanning

ScannerLibrary.CONSTANT.NOTIFICATION.SOUND_FAIL_OFF

: Notification sound played in case of fail scanning

ScannerLibrary.CONSTANT.NOTIFICATION.SOUND_SUCCESS_ON

: Notification sound played in case of success scanning

ScannerLibrary.CONSTANT.NOTIFICATION.SOUND_ALL_ON(default)

: Notification sound played in case of success and fail scanning

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.15 **setLightMode**

int setLightMode(int lightMode)

Function:

Set the light mode.

Set the action of Illumination and Aimer when you will scan, capture image, stream.

Specify ScannerLibrary.CONSTANT.LIGHT_MODE.ALL_ON when scanning.

If specify other parameters, reading performance decreased.

Parameter:

lightMode – Light mode

ScannerLibrary.CONSTANT.LIGHT_MODE.ALL_OFF

: Illumination and Aimer does not light up

ScannerLibrary.CONSTANT.LIGHT_MODE.AIMER_ON

: Aimer lights up

ScannerLibrary.CONSTANT.LIGHT_MODE.ILLUMINATION_ON

: Illumination lights up

ScannerLibrary.CONSTANT.LIGHT_MODE.ALL_ON

: Illumination and Aimer lights up

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.16 **getLightMode**

int getLightMode(void)

Function:

Get the action of Illumination and Aimer behavior when you will scan, capture image, stream.

Parameter:

Void – none

Return value:

ScannerLibrary.CONSTANT.LIGHT_MODE.ALL_OFF

: Illumination and Aimer does not light up

ScannerLibrary.CONSTANT.LIGHT_MODE.AIMER_ON

: Aimer lights up

ScannerLibrary.CONSTANT.LIGHT_MODE.ILLUMINATION_ON(default)

: Illumination lights up

ScannerLibrary.CONSTANT.LIGHT_MODE.ALL_ON

: Illumination and Aimer lights up

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.17 getImageDataSize

int getImageDataSize(void)

Function:

Get the data size needed for capturing image

Parameter:

Return value:

Return the data sizes for image capturing on success.

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.18 captureImage

int captureImage(byte[] buffer)

Function:

Capture the image

Parameter:

buffer - buffer to store image data.

Allocate the area necessary for storing image data.

For the detailt, refer to "5.2.17 getImageDataSize (p.28)"

Return value:

Return the captured image data size on success.

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

ScannerLibrary.CONSTANT.RETURN.ERROR_NOTOPENED

: Not opened error

5.2.19 **getStreamDataSize**

int getStreamDataSize(void)

Function:

Get the data size needed for streaming.

Parameter:

Return value:

Return the data sizes for streaming on.

success.ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.20 **startStream**

int startStream(void)

Function:

Start streaming.

Parameter:

void – none

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported

5.2.21 readStream

int readStream(byte[] buffer)

Function:

Read the stream data.

To realize preview, call this function continuously.

Parameter:

buffer - buffer to store stream data.

Allocate the area necessary for storing image data.

For the detail, refer to "5.2.19 getStreamDataSize (p.29)"

Return value:

Return the acquired stream data sizes on success.

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.22 stopStream

int stopStream(void)

Function:

Stop streaming.

Parameter:

void – none

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported

5.2.23 setSymbologyEnable

int setSymbologyEnable(int symbologyID, int enable)

Function:

Set reading enable/disable of the specified barcode.

Parameter:

symbologyID - Specify the symbol ID of each barcode.

For the symbol ID of each barcode, refer to "5.3.2 Code identification table (p.53)" .

When ScannerLibrary.CONSTANT.SYMBOLGY.ALL is specified, all barcodes are enabled/disabled.

enable - Reading enable/disable

ScannerLibrary.CONSTANT.SYMBOLGY_PARAMETER.ENABLE

: Read enable

ScannerLibrary.CONSTANT.SYMBOLGY_PARAMETER.DISABLE

: Read disable

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: unsupported error

5.2.24 getSymbologyEnable

int getSymbologyEnable(int symbologyID)

Function:

Get reading enable/disable of the specified barcode.

Parameter error:

symbologyID - Specify the symbol ID of each barcode.

For the symbol ID of each barcode, refer to "5.3.2 Code identification table (p.53)" .

Return value:

ScannerLibrary.CONSTANT.SYMBOLGY_PARAMETER.ENABLE

: Read enable

ScannerLibrary.CONSTANT.SYMBOLGY_PARAMETER.DISABLE

: Read disable

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.25 `getSymbologyMaxDefault`

`int getSymbologyMaxDefault(int symbologyID)`

Function:

Get default reading maximum number of digits of specified barcode.

Parameter:

`symbologyID` – Symbol ID

Specify Symbol ID of each barcode.

For the symbol ID of each barcode, refer to "5.3.2 Code identification table (p.53)" .

Return value:

Return default reading maximum number of digits on success.

If you specify barcode that can not changed number of digits (i.e. EAN13 etc.), return 0.

`ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER`

: Parameter error

`ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED`

: Unsupported error

5.2.26 `getSymbologyMinDefault`

`int getSymbologyMinDefault(int symbologyID)`

Function:

Get default reading minimum number of digits of the specified barcode.

Parameter:

`symbologyID` – Symbol ID

Specify Symbol ID of each barcode.

For the symbol ID of each barcode, refer to "5.3.2 Code identification table (p.53)" .

Return value:

Return default reading minimum number of digits on success.

If you specify barcode that can not changed number of digits (i.e. EAN13 etc.), return 0.

`ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER`

: Parameter error

`ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED`

: Unsupport error

5.2.27 setSymbologyMax

int setSymbologyMax(int symbologyID, int max)

Function:

Set the reading maximum number of digits of the specified barcode.

Barcodes larger than the set number of digits are not read.

Parameter:

symbologyID – Symbol ID

Specify Symbol ID of each barcode.

For the symbol ID of each barcode, refer to "5.3.2 Code identification table (p.53)" .

max - reading maximum number of digits

For the number of digits that can be set, refer to "5.3.3 Reading digits (p.55)".

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.28 getSymbologyMax

int getSymbologyMax(int symbologyID)

Function:

Get reading maximum number of digits of specified barcode.

Parameter:

symbologyID – Symbol ID

Specify Symbol ID of each barcode.

For the symbol ID of each barcode, refer to "5.3.2 Code identification table (p.53)" .

Return value:

Return reading maximum number of digits on success.

If you specify barcode that can not changed number of digits (i.e. EAN13 etc.), return 0.

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.29 setSymbologyMin

int setSymbologyMin(int symbologyID, int min)

Function:

Set reading minimum number of digits of specified barcode.

Bar codes less than the set number of digits are not read.

Parameter:

symbologyID – Symbol ID

Specify Symbol ID of each barcode.

For the symbol ID of each barcode, refer to "5.3.2 Code identification table (p.53)" .

min - reading minimum number of digits.

For the number of digits that can be set, refer to "5.3.3 Reading digits (p.55)".

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.30 getSymbologyMin

int getSymbologyMin(int symbologyID)

Function:

Get reading minimum number of digits of specified barcode.

Parameter:

symbologyID – Symbol ID

Specify Symbol ID of each barcode.

When ScannerLibrary.CONSTANT.SYMBOLLOGY.ALL is specified, all barcodes are enabled/disabled.

For the symbol ID of each barcode, refer to "5.3.2 Code identification table (p.53)" .

Return value:

Return the reading minimum number of digits on success.

If you specify barcode that can not changed number of digits (i.e. EAN13 etc.), return 0.

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.31 setSymbologyProperty

int setSymbologyProperty(int symbologyID, int propertyNo, int propertySetting)

Function:

Set the value of property setting of the specified barcode.

Specify property number and set value to change.

Parameter:

symbologyID – Symbol ID

Specify Symbol ID of each barcode.

For the symbol ID of each barcode, refer to "5.3.2 Code identification table (p.53)" .

propertyNo – Property number

Specify the property number.

For the property number, refer to "5.3.4 Property of barcode (p.56)".

propertySetting – Setting value for property

Set the property setting value.

For the property number, refer to "5.3.4 Property of barcode (p.56)".

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: unsupported error

5.2.32 **getSymbologyProperty**

`int getSymbologyProperty(int symbologyID, int propertyNo)`

Function:

Get value of property setting of specified barcode.

Specify property number and set value to change.

Parameter:

`symbologyID` – Symbol ID

Specify Symbol ID of each barcode.

For the symbol ID of each barcode, refer to "5.3.2 Code identification table (p.53)" .

`propertyNo` – Property number

Specify the property number.

For the property number, refer to "5.3.4 Property of barcode (p.56)".

Return value:

Return the value of property setting on success.

`ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER`

: Parameter error

`ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED`

: Unsupported error

5.2.33 setOutputType

int setOutputType(int outputType)

Function:

Set output type of the scan result.

Parameter:

outputType - The output type of the scan result

ScannerLibrary.CONSTANT.OUTPUT.CLIP : Clipboard output

ScannerLibrary.CONSTANT.OUTPUT.KEY : Keyboard output

ScannerLibrary.CONSTANT.OUTPUT.USER : User message output

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

Remarks:

Intent issued when user message output setting is enabled is "device.common.USERMSG".

To receive a user message prepare the receiver as shown below.

<intent-filter>

<action android:name="device.common.USERMSG" />

</intent-filter>

5.2.34 getOutputType

int getOutputType(void)

Function:

Get the output type of the scan result.

Parameter:

void – none

Return value:

ScannerLibrary.CONSTANT.OUTPUT.CLIP (default)

: Clipboard output

ScannerLibrary.CONSTANT.OUTPUT.KEY

: Keyboard output

ScannerLibrary.CONSTANT.OUTPUT.USER

: User message output

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error.

5.2.35 setSuffix

int setSuffix(int suffix)

Function:

Set the suffix type added at the end of the scan result.

Parameter:

suffix - suffix to be added.

ScannerLibrary.CONSTANT.SUFFIX.NONE : no suffix

ScannerLibrary.CONSTANT.SUFFIX.LF : LF (0x0A)

ScannerLibrary.CONSTANT.SUFFIX.TAB : TAB (0x09)

ScannerLibrary.CONSTANT.SUFFIX.TAB_LF : TAB+LF (0x09, 0x0A)

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.36 getSuffix

int getSuffix(void)

Function:

Get suffix type added at the end of scan result.

Parameter:

void – none

Return value:

ScannerLibrary.CONSTANT.SUFFIX.NONE (default) : no suffix

ScannerLibrary.CONSTANT.SUFFIX.LF : LF (0x0A)

ScannerLibrary.CONSTANT.SUFFIX.TAB : TAB (0x09)

ScannerLibrary.CONSTANT.SUFFIX.TAB_LF : TAB+LF (0x09, 0x0A)

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported

5.2.37 **setInverseMode**

int setInverseMode(int inverseMode)

Function:

Set inverse barcode reading mode.

Parameter:

inverseMode - Inverse barcode reading mode.

ScannerLibrary.CONSTANT.INVERSE.DISABLE

: Only Normal barcode can read

ScannerLibrary.CONSTANT.INVERSE.ENABLE

: Only Inverse barcode can read

ScannerLibrary.CONSTANT.INVERSE.AUTO

: Normal and Inverse barcode can read

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.38 **getInverseMode**

int getInverseMode(void)

Function:

Get inverse barcode reading mode.

Parameter:

void - none

Return value:

ScannerLibrary.CONSTANT.INVERSE.DISABLE (default)

: Only Normal barcode can read

ScannerLibrary.CONSTANT.INVERSE.ENABLE

: Only Inverse barcode can read

ScannerLibrary.CONSTANT.INVERSE.AUTO

: Normal and Inverse barcode can read

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.39 setTriggerKeyEnable

int setTriggerKeyEnable(int triggerKeyEnable)

Function:

Set Trigger key enable/disable.

Parameter:

triggerKeyEnable - Trigger key enable/disable.

ScannerLibrary.CONSTANT.TRIGGERKEY.DISABLE : Trigger key disable

ScannerLibrary.CONSTANT.TRIGGERKEY.ENABLE : Trigger key enable

This setting returns to the default after rebooting. To use this setting, please set it when starting the application.

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.40 getTriggerKeyEnable

int getTriggerKeyEnable(void)

Function:

Get Trigger key enable/disable.

Parameter:

void – none

Return value:

ScannerLibrary.CONSTANT.TRIGGERKEY.DISABLE

: Trigger key disable

ScannerLibrary.CONSTANT.TRIGGERKEY.ENABLE (default)

: Trigger key enable

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.41 setTriggerKeyMode

int setTriggerKeyMode(int triggerKeyMode)

Function:

Set Trigger key mode.

Parameter:

triggerMode – Trigger key mode

ScannerLibrary.CONSTANT.TRIGGER_MODE.NORMAL

: Normal Scan (Every time the trigger key is pressed, scanning)

ScannerLibrary.CONSTANT.TRIGGER_MODE.CONTINIOUS

: Continuous reading (scanning while pressing trigger key)

This setting returns to the default after rebooting. To use this setting, please set it when starting the application.

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported

5.2.42 getTriggerKeyMode

int getTriggerKeyMode(void)

Function:

Get Trigger key mode.

Parameter:

void – none

Return value:

ScannerLibrary.CONSTANT.TRIGGER_MODE.NORMAL (default)

: Normal scan (Every time the trigger key is pressed, scanning)

ScannerLibrary.CONSTANT.TRIGGER_MODE.CONTINIOUS

: Continuous reading (scanning while pressing trigger key)

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported

5.2.43 setTriggerKeyTimeout

int setTriggerKeyTimeout(int triggerKeyTimeout)

Function:

Set Trigger key timeout.

Parameter:

trggerMode – Trigger timeout

Specify a value in milliseconds.

The setting range is 1000 to 10000.

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.44 getTriggerKeyTimeout

int getTriggerKeyTimeout(void)

Function:

Get Trigger key timeout.

Parameter:

void – none

Return value:

Return the value of Trigger key timeout on success.

Default value is 10000.

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: unsupported error

5.2.45 setTriggerKeyOn

int setTriggerKeyOn(int triggerKeyOn)

Function:

Software control of Trigger key behavior.

Parameter:

trggerKeyOn - status of trigger key

ScannerLibrary.CONSTANT.CONTROL.TRIGGER_OFF

: Trigger key is released virtually

ScannerLibrary.CONSTANT.CONTROL.TRIGGER_ON

: Trigger key is pushed virtually

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported

5.2.46 setScannerAPO

int setScannerAPO(int scannerAPOTime)

Function:

Set the Auto Power Off time of the barcode scanner.

Parameter:

scannerAPOTime – Scanner APO time

Specify a value between 0 and 65535 in seconds.

If you set 0, APO is disabled.

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported

5.2.47 getScannerAPO

int getScannerAPO(void)

Function:

Get the Auto Power Off time of the barcode scanner.

Parameter:

void – none

Return value:

Return the APO time of barcode scanner.

Default value is 60.

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: unsupported error

5.2.48 setCenteringWindow

int setCenteringWindow(int centeringWindow)

Function:

Set enable/disable centering window mode.

Parameter:

centeringWindow - Enable/disable the centering window mode
ScannerLibrary.CONSTANT.CENTERING_WINDOW_MODE.DISABLE
: Centering window mode disable
ScannerLibrary.CONSTANT.CENTERING_WINDOW_MODE.ENABLE
: Centering window mode enable

Refer to the figure below for the barcode that can be read while the center reading mode is in effect. If part of the barcode is included in the detection area, read the barcode.

Returnvalue:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

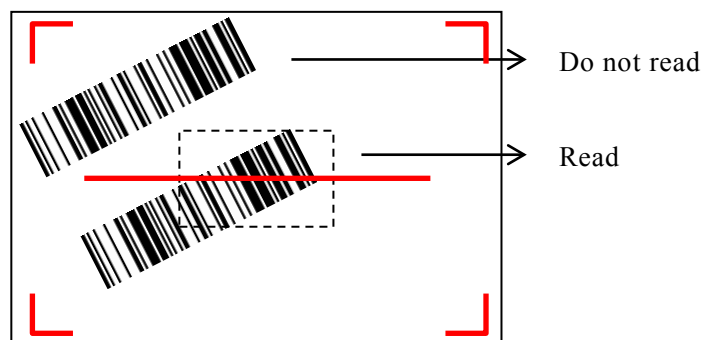
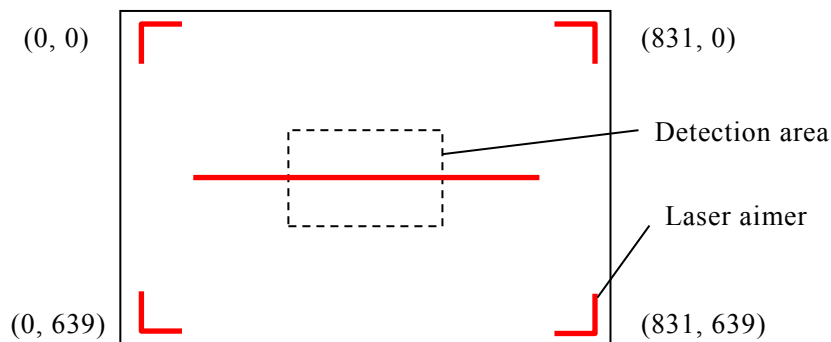
: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ERROR_UNSUPPORTED

: unsupported error



5.2.49 getCenteringWindow

int getCenteringWindow(void)

Function:

Get enable/disable the centering window mode.

Parameter:

void – none

Return value:

ScannerLibrary.CONSTANT.CENTERING_WINDOW_MODE.DISABLE (default)

: Centering window mode disable

ScannerLibrary.CONSTANT.CENTERING_WINDOW_MODE.ENABLE

: Centering window mode enable

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.50 setDetectionAreaSize

int setDetectionAreaSize(int detectionAreaSize)

Function:

Set the size of Barcode detection area for the centering window mode.

Parameter:

detectionAreaSize - The size of Barcode detection area

Specify a value between 0 and 10.

If you set smaller value, the barcode near the center is detected.

If you specified 0, the center point becomes the detection area.

Please change the value according to the actual use environment.

Returnvalue:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

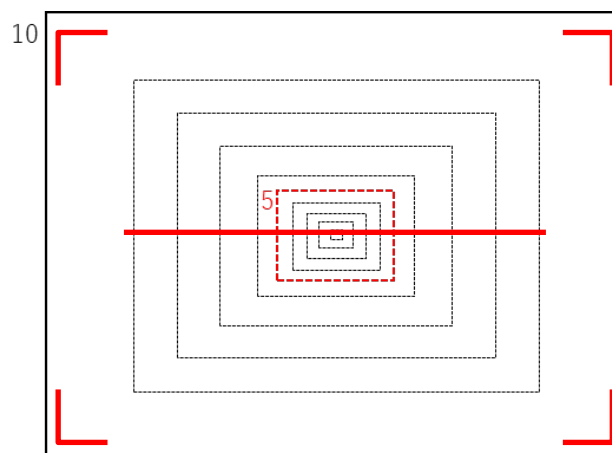
: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ERROR_UNSUPPORTED

: unsupported error



Detection area size as a guide.

5.2.51 **getDetectionAreaSize**

int getDetectionAreaSize(void)

Function:

Get the size of Barcode detection area for the centering window mode.

Parameter:

void – none

Return value:

Return the size of Barcode detection area for the centering window mode.

Default value is 5.

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED

: Unsupported error

5.2.52 **setInternalParameter**

int setInternalParameter(int tag, int value)

Function:

Set internal parameters. Detailed specifications are not disclosed.

Parameter:

Non-disclosure

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS

: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER

: Parameter error

ERROR_UNSUPPORTED

: unsupported error

5.2.53 **getInternalParameter**

int getInternalParameter(int tag)

Function:

Get internal parameters. Detailed specifications are not disclosed.

Parameter:

Non-disclosure

Return value:

Non-disclosure

5.3 CONSTANTS

Constants used with ScannerLibrary. Instance creation is not necessary.

5.3.1 List of constants

Return value constants

List of return value constants. Check execution result of member function with return value constant.

Class name : ScannerLibrary.CONSTANT.RETURN

Constant field	Description
SUCCESS	Successful completion. It returns when the function terminates normally.
ERROR_UNSUPPORTED	Unsupported error. It returns if the function is not supported.
ERROR_PARAMETER	Parameter error. It returns when the parameter specified as the argument is illegal.
ERROR_NOTOPENED	Non-open error. It returns when a function is called with the barcode scanner not open.

Notification constants

List of notification constants. Sets notification behavior when bar code is read.

Class name : ScannerLibrary.CONSTANT.NOTIFICATION

Constant field	Description
LED_OFF	Notification LED not lights up in case of success and fail scanning.
LED_ON	Notification LED turns on BLUE in case of success scanning, and RED in case of fail scanning.
VIBRATOR_ALL_OFF	Vibrator does not vibrate after success and fail scanning.
VIBRATOR_FAIL_ON	Vibrator vibrates in case of fail scanning.
VIBRATOR_SUCCESS_ON	Vibrator vibrates in case of success scanning.
VIBRATOR_ALL_ON	Vibrator vibrates in case of success and fail scanning.
SOUND_ALL_OFF	Vibrator does not vibrate after success and fail scanning.
SOUND_FAIL_ON	Notification sound sounds in case of fail scanning.
SOUND_SUCCESS_ON	Notification sound sounds in case of success scanning.
SOUND_ALL_ON	Notification sound sounds in case of success and fail scanning.

Constants of light mode

List of light mode constants. Set the illumination and the Aimer operation that light up when scanning, shooting still images and streaming.

Class name : ScannerLibrary.CONSTANT.LIGHT_MODE

Constant field	Description
ALL_OFF	Illumination and Aimer does not light up
AIMER_ON	Aimer lights up.
ILLUMINATION_ON	Illumination lights up.
ALL_ON	Illumination and Aimer lights up.

Control constants

List of control constants. Controls the trigger key press.

Class name : ScannerLibrary.CONSTANT.CONTROL

Constant field	Description
TRIGGER_OFF	Trigger key is released virtually.
TRIGGER_ON	Trigger key is pushed virtually.

Symbol parameter constants

List of symbol parameter constants. Set barcode read enable / disable.

Class name : ScannerLibrary.CONSTANT.SYMBOLOLOGY_PARAMETER

Constant field	Description
DISABLE	Read disable.
ENABLE	Read enable.

Symbol ID constants

List of symbol ID constants. Used to specify the symbol ID of each barcode.

Class name : ScannerLibrary.CONSTANT.SYMBOLOLOGY

Constant field	Description
AZTEC	Symbol ID of Aztec
CODABAR	Symbol ID of Codabar
CODABLOCKF	Symbol ID of Codablock F
CODE128	Symbol ID of Code128
CODE32	Symbol ID of Code32
CODE39	Symbol ID of Code39
CODE93	Symbol ID of Code93
COMPOSITE	Symbol ID of Composite
DATAMATRIX	Symbol ID of DataMatrix
EAN13	Symbol ID of EAN13
EAN8	Symbol ID of EAN8
GS1_128	Symbol ID of GS1 128(EAN128)
GS1_DATABAR	Symbol ID of GS1 DataBar(RSS)
HANXIN	Symbol ID of Han Xin
ITF	Symbol ID of ITF(Interleaved 2 of 5)
ISBT	Symbol ID of ISBT
MAXICODE	Symbol ID of Maxicode
MICRO_PDF	Symbol ID of Micro PDF
MSI	Symbol ID of MSI
PDF417	Symbol ID of PDF417
QR	Symbol ID of QR Code/ Micro QR Code
UPCA	Symbol ID of UPC-A
UPCE	Symbol ID of UPC-E0 / UPC-E1
ALL	All designated symbol ID. Specify when reading enable/disable for all symbology at once.

Output type constants

List of output type constants. Sets the output type of the reading result.

Class name : ScannerLibrary.CONSTANT.OUTPUT

Contant field	Description
CLIP	Clipboard output
KEY	Keyboard output.
USER	User message output

Suffix constants

List of "Suffix constants". Sets the qualifier to be appended to the end of the reading result.

Class name : ScannerLibrary.CONSTANT.SUFFIX

Constant field	Description
NONE	no suffix
LF	Add LF (0x0A)
TAB	Add TAB (0x09)
TAB_LF	Add TAB+LF (0x09, 0x0A)

Black-and-white inversion mode constants

List of black and white reverse mode constants. Set black-and-white inverted barcode read enable / disable.

Class name : ScannerLibrary.CONSTANT.INVERSE

Constant field	Description
DISABLE	Only normal bar code (barcode printed in black on white background) is read.
ENABLE	Only black-and-white reversed barcode (barcode printed in black on black background) is read.
AUTO	Read normal barcode and black-and-white reversing bar code.

Trigger key constants

List of Trigger key constants. Set trigger key enable / disable.

Class name : ScannerLibrary.CONSTANT.TRIGGERKEY

Constant field	Description
DISABLE	Trigger key disable
ENABLE	Trigger key enable

Trigger key mode constants

List of Trigger key mode constants. Set the action when trigger key is pressed.

Class name : ScannerLibrary.CONSTANT.TRIGGER_MODE

Constant field	Description
NORMAL	Normal reading. Scan is performed every time the trigger key is pressed
CONTINUOUS	Continuous reading. Scanning continuesly during trigger key is pressed.

Centering window mode constant

List of centering window mode constants. Enable / disable centering window mode.

Class name : ScannerLibrary.CONSTANT.CENTERING_WINDOW_MODE

Constant field	Description
DISABLE	Disable "Centering window mode"
ENABLE	Enable "Centering window mode"

5.3.2 Code identification table

Barcode name "Return value of SymbologyName"	Symbol ID	AIM ID	AIM ID Modifier
Aztec "AZTEC"	AZTEC	z	0-1
Codabar "CODABAR"	CODABAR	F	0-1
Codablock F "CODABLOCK F"	CODABLOCKF	O	0, 1, 4, 5, 6
Code128 "CODE 128"	CODE128	C	0, 1, 2, 4
Code32 "CODE 32 PHARMACEUTICAL (PARAF)"	CODE32	X	0
Code39 "CODE 39"	CODE39	A	0, 1, 3, 4, 5, 7
Code93 "CODE 93"	CODE93	G	0-9, A-Z, a-m
Composite	COMPOSITE	-	-
DataMatrix "DATAMATRIX"	DATAMATRIX	d	0-6
EAN13 "EAN 13"	EAN13	E	0-4
EAN8 "EAN 8"	EAN8	E	0-4
GS1 128(EAN128) "GS1 128(EAN 128)"	GS1_128	C	0, 1, 2, 4
GS1 DataBar(RSS) "EAN.UCC Composite / GS1-DataBar(RSS)"	GS1_DATABAR	e	0
Han Xin "HAN XIN"	HANXIN	X	0
ITF(Interleaved 2 of 5) "INTERLEAVED 2 OF 5"	ITF	I	0, 1, 3
ISBT "ISBT 128"	ISBT	C	4
Maxicode "MAXICODE"	MAXICODE	U	0-3
Micro PDF "MICRO PDF417"	MICRO_PDF	L	3-5

MSI "MSI"	MSI	M	0, 1
PDF417 "PDF417"	PDF417	L	0-2
QR Code/ Micro QR Code "QR CODE"	QR	Q	0-6
UPC-A "UPC A"	UPCA	E	0-4
UPC-E0 / UPC-E1 "UPC E" or "UPC E1"	UPCE	E	0-4

Note

The return value of symbologyName may be changed due to barcode standard modification etc. Please use the Symbol ID for the code identification of the read barcode, and use the return value of symbologyName for display and debugging purposes.

When reading the COMPOSITE code, also enable set 1D barcode placed in the lower row. When read the COMPOSITE code, return the symbol ID/AIM ID/AIM ID Modifier depend on 1D barcode placed in the lower row. Please use the length of barcode to judge that COMPOSITE code was read.

column

Reading performance is improved by limiting the barcode to be read. It is recommended to enable only the barcode which is actually read, and disable other barcodes.

5.3.3 Reading digits

barcode name	Default		Configurable	
	Minimum digits	Maximum digits	Minimum digits	Maximum digits
Aztec	1	3832	1	3832
Codabar	2	60	2	60
Codablock F	0	2048	0	2048
Code128	0	80	0	80
Code32	-	-	-	-
Code39	2	48	0	48
Code93	0	80	0	80
Composite	1	300	1	300
DataMatrix	1	3166	1	3166
EAN13	-	-	-	-
EAN8	-	-	-	-
GS1 128(EAN128)	0	80	0	80
GS1 DataBar(RSS)	1	80	1	80
Han Xin	1	6000	1	6000
ITF(Interleaved 2 of 5)	4	80	2	80
ISBT	-	-	-	-
Maxicode	1	150	1	150
Micro PDF	1	2750	1	2750
MSI	4	48	4	48
PDF417	1	2750	1	2750
QR Code/ Micro QR Code	1	7089	1	7089
UPC-A	-	-	-	-
UPC-E0 / UPC-E1	-	-	-	-

The "number of reading digits" is the number of digits that can be set in the barcode scanner, depends on the print quality of the barcode and environmental conditions whether the barcode can be read.

column

The misreading incidence decreases by limiting the reading digits. It is recommended to set the reading digits according to the barcode which is actually read.

5.3.4 Property of barcode

Barcode name	Property number	Property name	Possible values
Aztec	-	-	-
Codabar	0	Start/Stop Characters	0: No start / stop code is output. (Default) 1: Start / stop code is output.
	1	Check Character	0: Read regardless of presence or absence of check character (default) 1: Only barcodes with check characters are read.
	2	Send Check Character	0: No check character is output. (Default) 1: Outputs the check character.
Codablock F	-	-	-
Code128	-	-	-
Code32	-	-	-
Code39	0	Start/Stop Characters	0: No start / stop code is output. (Default) 1: Start / stop code is output.
	1	Full ASCII	0: Full ASCII conversion is not performed. (Default) 1: Full ASCII conversion is performed.
	2	Check Character	0: Read regardless of presence or absence of check character (default) 1: Only bar codes with check characters are read.
	3	Send Check Character	0: No check character is output. (Default) 1: Outputs the check character.
Code93	-	-	-
Composite	0	UPC Composite Codes	0: UPC / EAN Composite is not read. (Default) 1: Read UPC / EAN Composite. ※ 1
DataMatrix	-	-	-
EAN13	0	Send Check Character	0: No check character is output. 1: Outputs the check character. (Default)
	1	Addenda Required	0: It reads regardless of presence of add-on. (Default) 1: Read bar codes with 2 or 5 digit add-on only.
	2	Include Addenda Separator	0: Output bar code data and add-ons without separating them with a space. 1: Separate barcode data and add-ons with spaces. (Default)
	3	2-Digit Addenda	0: Ignore the 2 digit add-on. (Default) 1: When there is a two-digit add-on, the add-on is also read.
	4	5-Digit Addenda	0: Ignore the 5 digit add-on. (Default) 1: When there is a 5 digit add-on, the add-on is also read.

EAN8	0	Send Check Character	0: No check character is output. 1: Outputs the check character. (Default)
	1	Addenda Required	0: reads regardless of presence of add-on. (Default) 1: Read bar codes with 2 or 5 digit add-on only.
	2	Include Addenda Separator	0: Output bar code data and add-ons without separating them with a space. 1: Separate barcode data and add-ons with spaces. (Default)
	3	2-Digit Addenda	0: Ignore the 2 digit add-on. (Default) 1: When there is a two-digit add-on, the add-on is also read.
	4	5-Digit Addenda	0: Ignore the 5 digit add-on. (Default) 1: When there is a 5 digit add-on, the add-on is also read.
GS1 128(EAN128)	-	-	-
GS1 DataBar(RSS)	0	RSE Symbology enable	0: Disable reading of GS1 DataBar Expanded. 1: Enable reading of GS1 DataBar Expanded. (Default)
	1	RSL Symbology enable	0: Disable reading of GS1 DataBar Limited. 1: Enable reading of GS1 DataBar Limited. (Default)
	2	RSS Symbology enable	0: Disable reading of GS1 DataBar 14. 1: Enable reading of GS1 DataBar 14. (Default)
Han Xin	-	-	-
ITF (Interleaved 2 of 5)	0	Check Character	0: Read regardless of presence or absence of check character (default) 1: Only barcodes with check characters are read.
	1	Send Check Character	0: No check character is output. (Default) 1: Outputs the check character.
ISBT	-	-	-
Maxicode	-	-	-
Micro PDF	-	-	-
MSI	0	Send Check Character	0: No check character is output. (Default) 1: Outputs the check character.
PDF417	-	-	-
QR Code/ Micro QR Code	-	-	-
UPC-A	0	Send Check Character	0: No check character is output. 1: Outputs the check character. (Default)
	1	Addenda Required	0: Reads regardless of presence of add-on. (Default) 1: Read bar codes with 2 or 5 digit add-on only.
	2	Include Addenda Separator	0: Output bar code data and add-ons without separating them with a space. 1: Separate barcode data and add-ons with spaces. (Default)

	3	2-Digit Addenda	0: Ignore the 2 digit add-on. (Default) 1: When there is a two-digit add-on, the add-on is also read.
	4	5-Digit Addenda	0: Ignore the 5 digit add-on. (Default) 1: When there is a 5 digit add-on, the add-on is also read.
	5	Send Num Sys	0: No number system character is output. 1: Number system character is output. (Default)
	6	Convert UPC-A to EAN-13	0: UPC-A is not convert to EAN13. (Default) 1: Converts UPC-A to EAN13 format. ※2
UPC-E0 / UPC-E1	0	Send Check Character	0: No check character is output. 1: Outputs the check character. (Default)
	1	Addenda Required	0: Reads regardless of presence of add-on. (Default) 1: Read bar codes with 2 or 5 digit add-on only.
	2	Extended UPC-E	0: UPC-E is not extended to UPC-A format. (Default) 1: Extends UPC-E to UPC-A format.
	3	Include Addenda Separator	0: Output bar code data and add-ons without separating them with a space. 1: Separate barcode data and add-ons with spaces. (Default)
	4	2-Digit Addenda	0: Ignore the 2 digit add-on. (Default) 1: When there is a two-digit add-on, the add-on is also read.
	5	5-Digit Addenda	0: Ignore the 5 digit add-on. (Default) 1: When there is a 5 digit add-on, the add-on is also read.
	6	Send Num Sys	0: No number system character is output. 1: Number system character is output. (Default)

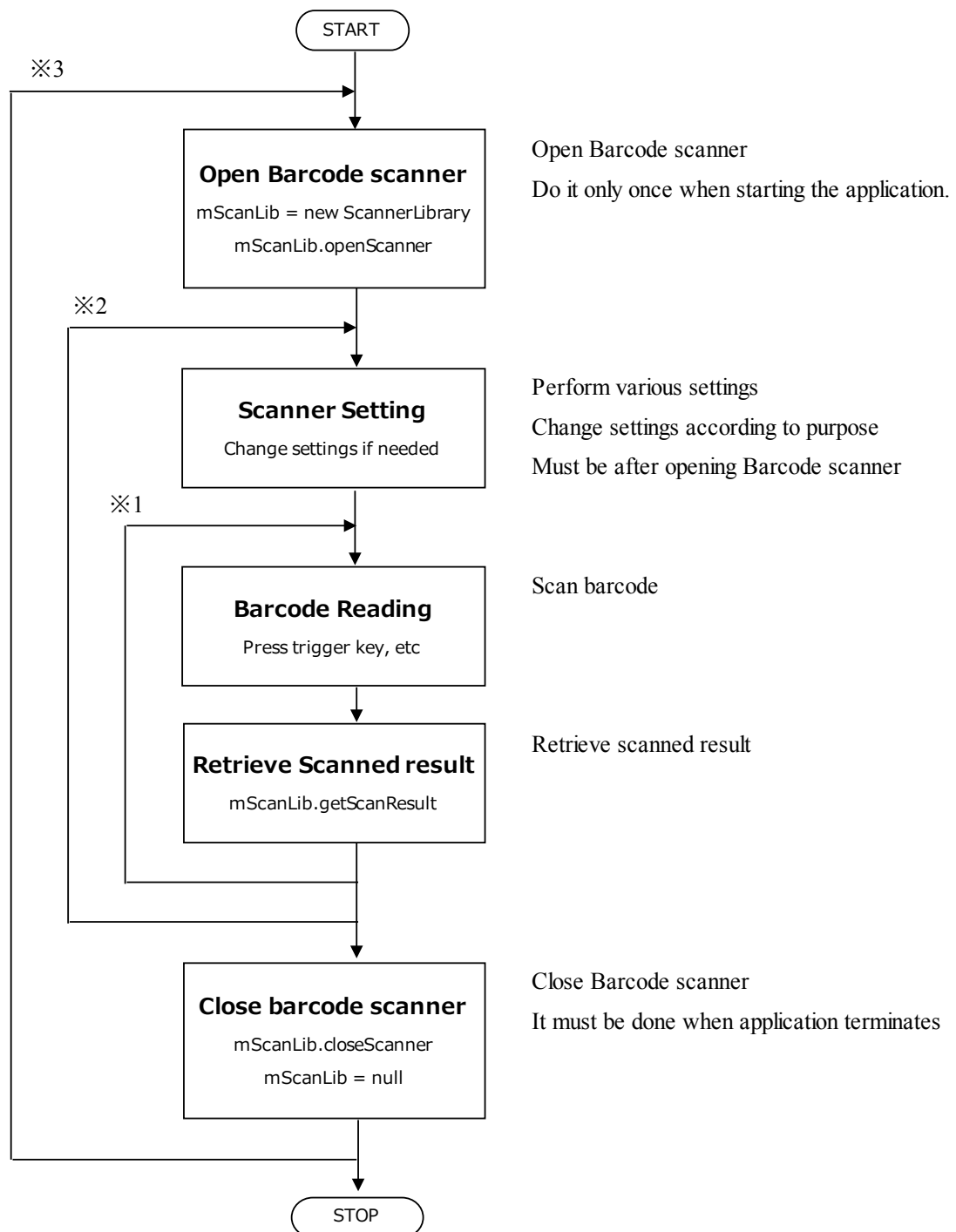
Remarks)

※1 It is recommended to set 1 when reading UPC / EAN Composite. When 0 is set, it may be read by UPC / EAN only.

※2 Please set reading enable UPC-A and EAN13.

5.4 Basic flow of scanning application

5.4.1 Barcode reading procedure



※1 Repeat this process when reading the bar code with the same settings.

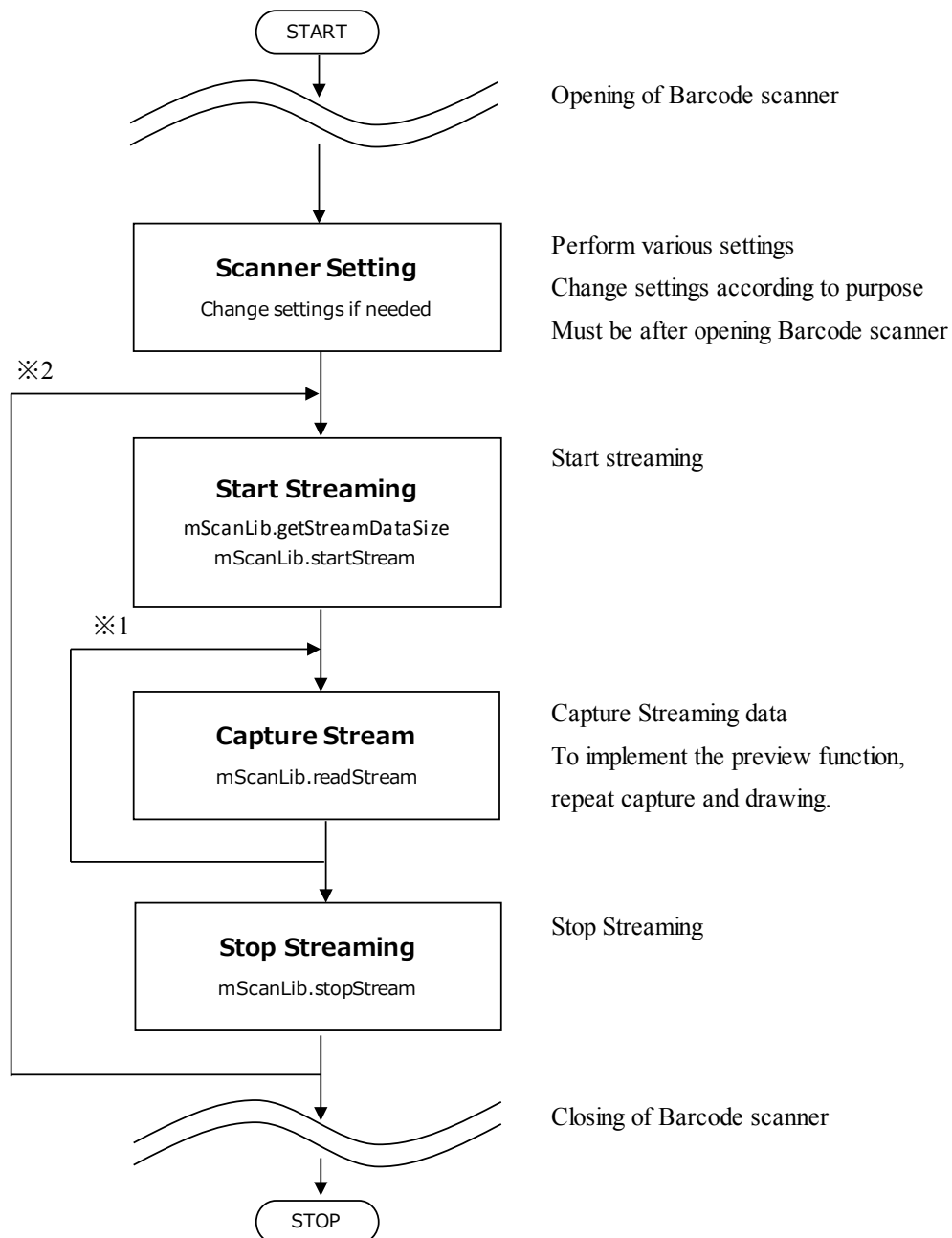
※2 Repeat this process to read bar codes with different settings.

※3 Repeat this process when turning off the power of the barcode scanner device.

Column

It takes time to initialize the barcode scanner. Therefore, if want to change use / non-use of barcode scanner at high speed, we recommend that switch enable / disable of trigger key. For the detail, refere to "5.2.39 setTriggerKeyEnable (p.40)".

5.4.2 Streaming flow



※1 To implement preview, repeatedly obtain and draw streaming data.

※2 Repeat this process to resume streaming.

5.4.3 Shooting still images

