

DT-X400 Series

Android 8.1 Device Library Manual

This document explains how to use the DT-X400 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 LLC.
- 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. Permissions	6
1.3.2. API Level	6
1.3.3. Access from multiple 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	13
4.2.4. setFnUserKeyCode	14
4.2.5. getFnUserKeyCode	14
4.2.6. setFnDefaultKeyCode	14
4.2.7. setLaunchApplication	15
4.2.8. getLaunchApplication	15
4.2.9. clearLaunchApplication	16
4.2.10. ApplicationInfo class	16
4.2.11. setFnLaunchApplication	17
4.2.12. getFnLaunchApplication	17
4.2.13. clearFnLaunchApplication	18
4.3. Constants	19
4.3.1. List of Constants	19
5. Barcode Scanner	21
5.1. Overview	21
5.1.1. List of Member Functions	22
5.2. ScannerLibrary	24
5.2.1. openScanner	24

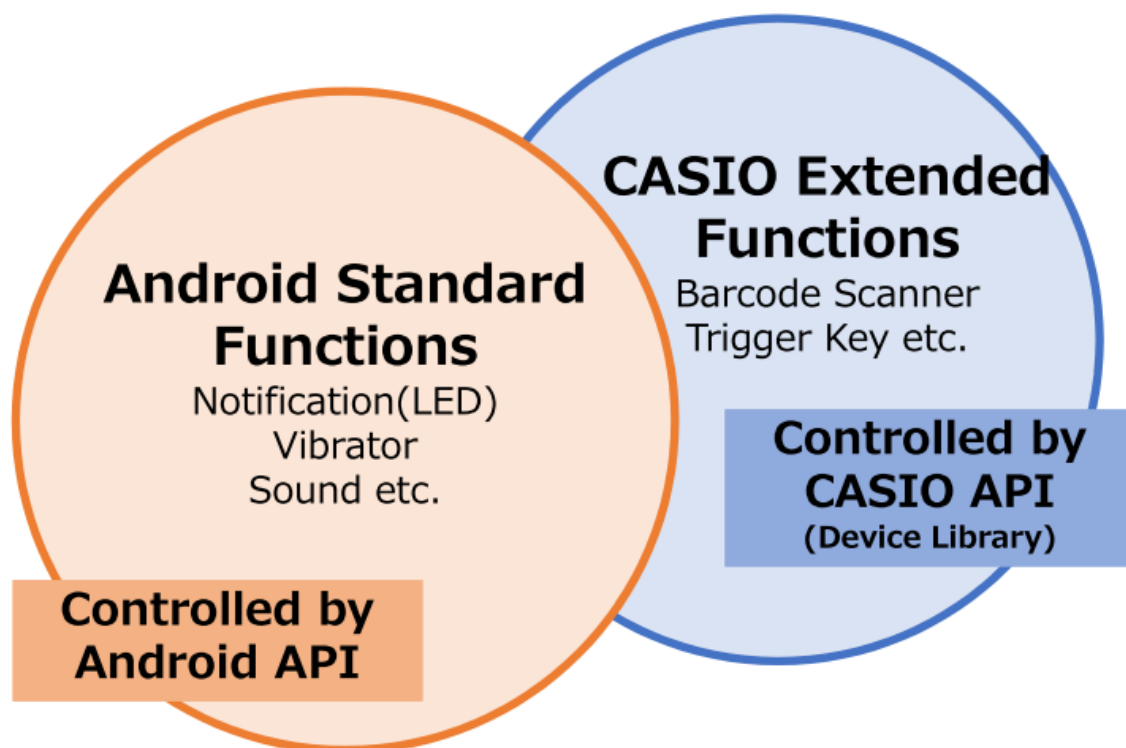
5.2.2. closeScanner	24
5.2.3. isScannerOpen	24
5.2.4. setDefaultAll	25
5.2.5. getAPIVersion	25
5.2.6. getModuleVersion	25
5.2.7. getScanResult	26
5.2.8. ScanResult class	26
5.2.9. setNotificationLED	27
5.2.10. getNotificationLED	27
5.2.11. setNotificationVibrator	28
5.2.12. getNotificationVibrator	28
5.2.13. setNotificationSound	29
5.2.14. getNotificationSound	29
5.2.15. setLightMode	30
5.2.16. getLightMode	30
5.2.17. getImageDataSize	31
5.2.18. captureImage	31
5.2.19. getStreamDataSize	31
5.2.20. startStream	32
5.2.21. readStream	32
5.2.22. stopStream	32
5.2.23. setSymbologyEnable	33
5.2.24. getSymbologyEnable	33
5.2.25. getSymbologyMaxDefault	34
5.2.26. getSymbologyMinDefault	34
5.2.27. setSymbologyMax	35
5.2.28. getSymbologyMax	35
5.2.29. setSymbologyMin	36
5.2.30. getSymbologyMin	36
5.2.31. setSymbologyProperty	37
5.2.32. getSymbologyProperty	37
5.2.33. setOutputType	38
5.2.34. getOutputType	38
5.2.35. setSuffix	39
5.2.36. getSuffix	39
5.2.37. setInverseMode	40
5.2.38. getInverseMode	40
5.2.39. setTriggerKeyEnable	41
5.2.40. getTriggerKeyEnable	41
5.2.41. setTriggerKeyMode	42
5.2.42. getTriggerKeyMode	42
5.2.43. setNumberOfBarcodes	43

5.2.44. getNumberOfBarcodes	43
5.2.45. setDelimiter	44
5.2.46. getDelimiter	44
5.2.47. setTriggerKeyTimeout	45
5.2.48. getTriggerKeyTimeout	45
5.2.49. setTriggerKeyOn	45
5.2.50. setScannerAPO	46
5.2.51. getScannerAPO	46
5.2.52. setCenteringWindow	47
5.2.53. getCenteringWindow	48
5.2.54. setDetectionAreaSize	48
5.2.55. getDetectionAreaSize	49
5.2.56. setLaserSwingWidth	49
5.2.57. getLaserSwingWidth	49
5.2.58. setLaserHighlightMode	50
5.2.59. getLaserHighlightMode	50
5.2.60. setInternalParameter	51
5.2.61. setInternalParameter	51
5.2.62. getInternalParameter	51
5.3. CONSTANTS	52
5.3.1. List of constants	52
5.3.2. Code identification table	56
5.3.3. Reading digits	58
5.3.4. Property of barcode	60
5.4. Basic flow of scanning application	68
5.4.1. Barcode scanning flow	68
5.4.2. Streaming flow	69
5.4.3. Image capturing flow	70

1. Overview

1.1. Purpose

CASIO rugged smart handheld terminal (hereinafter referred to as CASIO rugged smart devices) running the Android operating system. CASIO rugged smart devices 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 control the CASIO extended function.



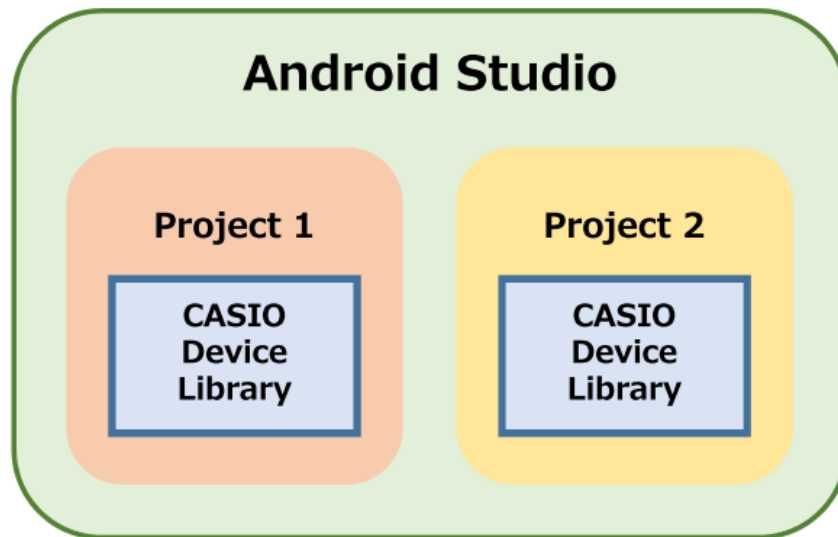
This library is defined as the standard specification of future CASIO rugged smart devices. However, depending on the model, mounted devices may be different. Read the explanation of each chapter for the method of developing a common application that will operate in the future CASIO rugged smart device.

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 for Android Studio. Import this library for each project.

File name: DeviceLibrary.aar



1.3. Application Development

1.3.1. Permissions

Usually, the Android permission is not required to use this library. If the permission requirement is described in the individual function, follow that.

1.3.2. API Level

To create Android projects, it is necessary to specify "minimum API level" and "target API level". Since the minimum API level depends on this library, build and installation will fail when if it is not set correctly. Since the target API level does not depend on this library, set it appropriately according to the Android API to be used.

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 27
Target API level	Application developer should set appropriate level.

Cautions!

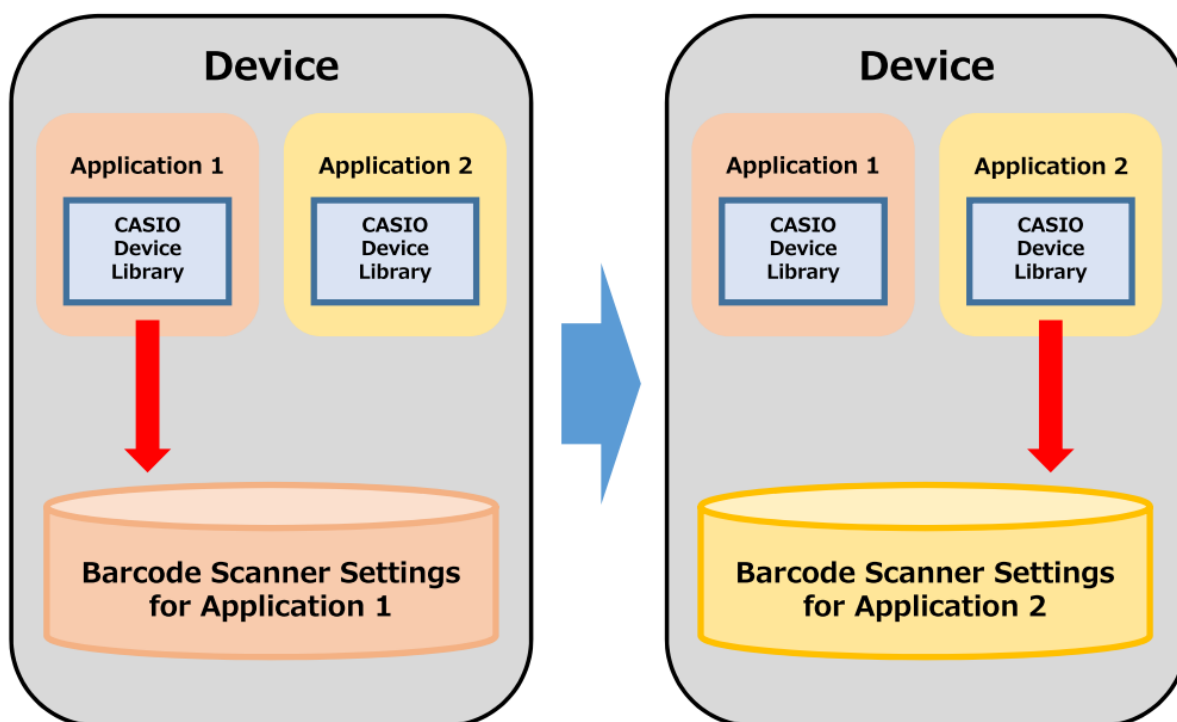
If you set the project API level to 16 or less, a build error occurs.

If you set the API level of the project to 28 or more, it can not be installed on the DT-X400.

1.3.3. Access from multiple applications

By using this library, you can change the settings of CASIO extended function. If multiple applications are change the settings, the later changing will be applied.

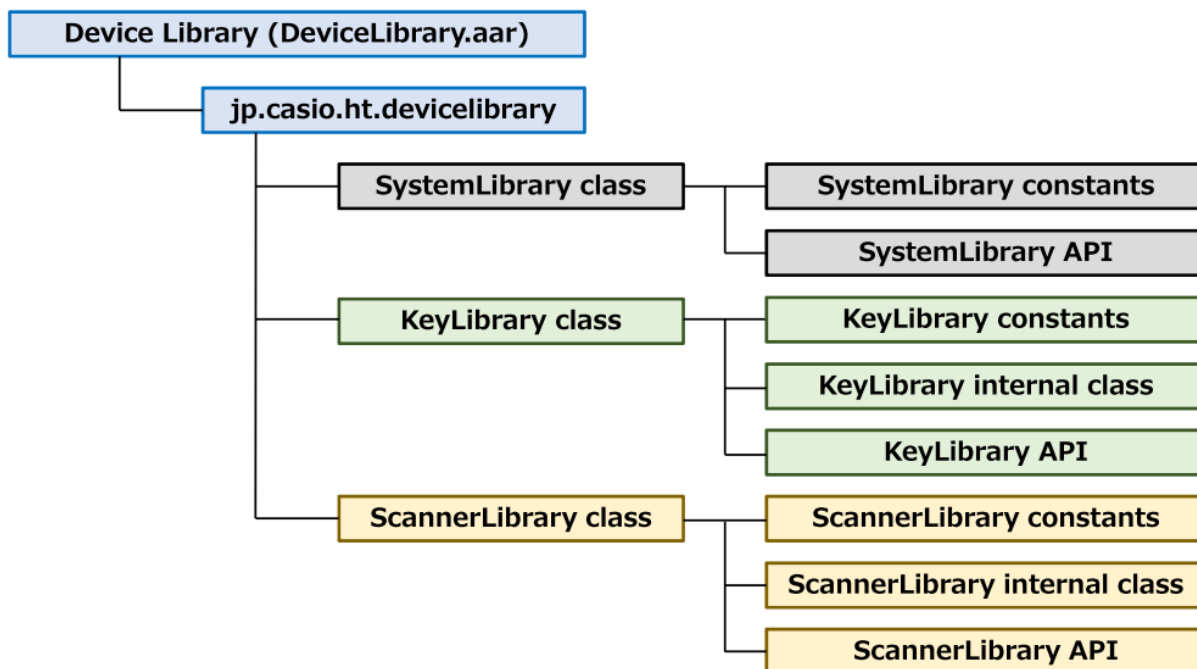
For example, if the Application 2 changes the barcode scanner settings after the Application 1 changes, the setting of the barcode scanner becomes the settings of Application 2.



The changed setting is retained even when the terminal is restarted. Refer to the explanation of each chapter about how to restore the settings.

2. Library Structure

This library consists of the following classes.



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

3. System

3.1. Overview

SystemLibrary class is used to get and set system information.

To use the SystemLibrary class in the project, import the class and 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 the barcode serial number
getModelName	Get the model name

3.2. SystemLibrary

3.2.1. getCASIOSerial

String getCASIOSerial (void)

Function:

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

The Android build serial can be obtained by SERIAL member of the android.os.Build class. Refer to the Android developer site about Android build serial.
The barcode serial number acquired with this function does not necessarily match the Android build serial.

Parameter:

void None

Return value:

String Barcode serial
Returns barcode serial number on success and null on failure.

3.2.2. getModelName

String getModelName (void)

Function:

Get the model name (e.g. DT-X400-10) of the device.
You can determine the installed device in the terminal from this model name.
For the relation between model name and installed device, refer to "1.1 Model Configuration" in the hardware manual.

Parameter:

void None

Return value:

String Model name
Returns Model name on success and null on failure.

4. Key

4.1. Overview

The KeyLibrary class is used to change the key code generated when the hardware key is pushed, or to register the application launched when the hardware key is pushed.

Functions	Description
Key code setting	Set the key code for the key.
Key code disabling	By setting the key code to blank, the key does not generate key code. It can be used to invalidate the key.
Register application	Register the launched application to the key.

The list of configurable keys and their support status are as follows. For the key layout and input mode, refer to "2.9.2 Hardware keyboard" in the software manual.

To specify the target key, use the KEYID of each key. If an unsupported KEYID is specified, each function returns ERROR_NOTSUPPORTED.

Trigger key setting is common to input mode of "1/A/a mode" and "Fn mode". For other key setting, separately for input mode.

Key	Input mode	
	1/A/a	Fn
Right Trigger key	Yes	
Left Trigger key	Yes	
Center Trigger key	Yes	
Back Trigger key	Yes	
0-9 key	Yes	Yes
F1 key	Yes	Yes
F2 key	Yes	Yes
F3 key	Yes	Yes
F4 key	Yes	Yes
F5(SP) key	Yes	Yes
F6(Volume Down) key	Yes	Yes
F7(Volume Up) key	Yes	Yes
F8(Mode change) key	Yes	Yes
Cursor key	Yes	Yes
CLR key	Yes	Yes
Enter key	Yes	Yes
Period key	Yes	Yes
Fn key	No	No

yes: supported, no: not supported

The settings are reflected in the entire system and are valid until clear the setting or initialize the terminal. If multiple applications are change the settings, the later changing will be applied.

To use the KeyLibrary class in the project, 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 the key code generated when the specified key is pushed in 1/A/a mode.
getUserKeyCode	Get the key code generated when the specified key is pushed in 1/A/a mode.
setDefaultKeyCode	Set the key code generated when the specified key is pushed in 1/A/a mode to default.
setFnUserKeyCode	Set the key code generated when the specified key is pushed in Fn mode.
getFnUserKeyCode	Get the key code generated when the specified key is pushed in Fn mode.
setFnDefaultKeyCode	Set the key code generated when the specified key is pushed in Fn mode to default.
Function for Application settings	
setLaunchApplication	Set the application launched when the specified key is pushed in 1/A/a mode.
getLaunchApplication	Get the application information launched when the specified key is pushed in 1/A/a mode.
clearLaunchApplication	Clear the application information launched when the specified key is pushed in 1/A/a mode.
setFnLaunchApplication	Set the application launched when the specified key is pushed in Fn mode.
getFnLaunchApplication	Get the application information launched when the specified key is pushed in Fn mode.
clearFnLaunchApplication	Clear the application information launched when the specified key is pushed in Fn mode.

4.2. KeyLibrary

4.2.1. setUserKeyCode

int setUserKeyCode(int keyID, int KeyCode)

Functions:

Set the key code generated when the specified key is pushed in 1/A/a mode.

Parameters:

int keyID KEYID

Specify the KEYID of the target key

For details of the KEYID for each key, see "4.3 Constants (p.19)".

int KeyCode key code

Specify the key code.

To set the Android standard key code, specify the key code of "Android's KeyEvent class". For the details, refer to the android.view.KeyEvent class from the Android Developer site.

To use the key code extended by CASIO such as the trigger key, specify the key code of "KeyLibrary class". For details of "4.3 Constants (p.19)".

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.19)".

4.2.2. getUserKeyCode

int getUserKeyCode(int keyID)

Functions:

Get the key code generated when the specified key is pushed in 1/A/a mode.

Parameters:

int keyID KEYID

Specify the KEYID of the target key.

For details of KEYID for each key, see "4.3 Constants (p.19)".

Return value:

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

For details of the return value, see "4.3 Constants (p.19)".

4.2.3. setDefaultKeyCode

int setDefaultKeyCode(int keyID)

Functions:

Set the key code to default generated when the specified key is pushed in 1/A/a mode.

Parameters:

int keyID KEYID

Specify the KEYID of the target key.

For details of KEYID of each key, see "4.3 Constants (p.19)".

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.19)".

4.2.4. setFnUserCode

int setFnUserCode(int keyID, int KeyCode)

Functions:

Set the key code generated when the specified key is pushed in Fn mode.

Parameters:

int keyID KEYID

Specify the KEYID of the target key.

For details of KEYID of each key, see "4.3 Constants (p.19)".

int KeyCode key code

Specify the key code.

To set the Android standard key code, specify the key code of "Android's KeyEvent class". For the details, refer to the android.view.KeyEvent class from the Android Developer site.

To use the key code extended by CASIO such as the trigger key, specify the key code of "KeyLibrary class". For details of "4.3 Constants (p.19)".

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.19)".

4.2.5. getFnUserCode

int getFnUserCode(int keyID)

Functions:

Get the key code generated when the specified key is pushed in Fn mode.

Parameters:

int keyID KEYID

Specify the KEYID of the target key.

For details of KEYID of each key, see "4.3 Constants (p.19)".

Return value:

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

For details of the return value, see "4.3 Constants (p.19)".

4.2.6. setFnDefaultKeyCode

int setFnDefaultKeyCode(int keyID)

Functions:

Set the key code to default generated when the specified key is pushed in Fn mode.

Parameters:

int keyID KEYID

Specify the KEYID of the target key.

For details of KEYID of each key, see "4.3 Constants (p.19)".

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.19)".

4.2.7. setLaunchApplication

int setLaunchApplication(int keyID, KeyLibrary.LaunchApplication launchApplication)

Functions:

Set the application launched when the specified key is pushed in 1/A/a mode.
If you set the launched application with this function, the key code is not generated when the key is pushed. If you clear the launched application, the key code is generated again.

Parameters:

int keyID KEYID

Specify the KEYID of the target key.

For details of KEYID of each key, see "4.3 Constants (p.19)".

KeyLibrary.LaunchApplication launchApplication: The launched application

For the member variables of the LaunchApplication class, refer to "4.2.10ApplicationInfo class (p.16)".

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.19)".

4.2.8. getLaunchApplication

int getLaunchApplication(int keyID, KeyLibrary.LaunchApplication launchApplication)

Functions:

Get the application information launched when the specified key is pushed in 1/A/a mode.

Parameters:

int keyID KEYID

Specify the KEYID of the target key.

For details of KEYID of each key, see "4.3 Constants (p.19)".

KeyLibrary.LaunchApplication launchApplication: The launched application

For the member variables of the LaunchApplication class, refer to "4.2.10ApplicationInfo class (p.16)".

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.19)".

4.2.9. clearLaunchApplication

`int clearLaunchApplication(int keyID)`

Functions:

Clear the application information launched when the specified key is pushed in 1/A/a mode.

Parameters:

`int keyID` KEYID

Specify the KEYID of the target key.

For details of KEYID of each key, see "4.3 Constants (p.19)".

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.19)".

4.2.10. ApplicationInfo class

`static class ApplicationInfo`

Description:

Internal class to store the launched application information.

Create an instance as follows.

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

■ Member variables:

`String packageName`

When assigning, specify the package name of the launched application.

When acquiring, the package name of the assigned 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 launched application.

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"`

4.2.11. setFnLaunchApplication

int setFnLaunchApplication(int keyID, KeyLibrary.LaunchApplication launchApplication)

Functions:

Set the application launched when the specified key is pushed in Fn mode.

If you set the application with this function, the key code is not generated when the key is pushed. If you clear the application information, the key code is generated again.

Parameters:

int keyID KEYID

Specify the KEYID of the target key.

For details of KEYID of each key, see "4.3 Constants (p.19)".

KeyLibrary.LaunchApplication launchApplication : The launched application

For the member variables of the LaunchApplication class, refer to

"4.2.10ApplicationInfo class (p.16)".

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.19)".

4.2.12. getFnLaunchApplication

int getFnLaunchApplication(int keyID, KeyLibrary.LaunchApplication launchApplication)

Functions:

Get the application information launched when the specified key is pushed in Fn mode.

Parameters:

int keyID KEYID

Specify the KEYID of the target key.

For details of KEYID of each key, see "4.3 Constants (p.19)".

KeyLibrary.LaunchApplication launchApplication: The launched application

For the member variables of the LaunchApplication class, refer to

"4.2.10ApplicationInfo class (p.16)".

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.19)".

4.2.13. clearFnLaunchApplication

int clearFnLaunchApplication(int keyID)

Functions:

Clear the application information launched when the specified key is pushed in Fn mode.

Parameters:

int keyID KEYID

Specify the KEYID of the target key.

For details of KEYID of each key, see "4.3 Constants (p.19)".

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.19)".

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.
CENTERTRIGGER	KEYID of Center Trigger key
BACKTRIGGER	KEYID of Back Trigger key
KEY1	KEYID of 1 key
KEY2	KEYID of 2 key
KEY3	KEYID of 3 key
KEY4	KEYID of 4 key
KEY5	KEYID of 5 key
KEY6	KEYID of 6 key
KEY7	KEYID of 7 key
KEY8	KEYID of 8 key
KEY9	KEYID of 9 key
KEY0	KEYID of 0 key
CLEAR	KEYID of CLR key
ENTER	KEYID of Enter key
PERIOD	KEYID of Period key
UP	KEYID of Up key
DOWN	KEYID of Down key
LEFT	KEYID of Left key
RIGHT	KEYID of Right key
F1	KEYID of F1 key
F2	KEYID of F2 key
F3	KEYID of F3 key
F4	KEYID of F4 key
F5	KEYID of F5 key
F6	KEYID of F6 key
F7	KEYID of F7 key
F8	KEYID of F8 key

Key code constants

Key code defined by CASIO.

Class name: KeyLibrary.CONSTANT.KEYCODE

Constant field	Description
KEYCODE_TRIGGER_RIGHT	Key code of Right Trigger key
KEYCODE_TRIGGER_LEFT	Key code of Left Trigger key
KEYCODE_TRIGGER_CENTER	Key code of Center Trigger key
KEYCODE_TRIGGER_BACK	Key code of Back Trigger key
KEYCODE_BLANK	Key code of blank. By using this key code, the key does not generate key code.

5. Barcode Scanner

5.1. Overview

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

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

To maintain compatibility of the source code, the barcode scanner library also provides deprecated 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. To avoid the settings to affect to other applications, it is necessary to reconfigure them when the application startup and active.

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

The DT-X400 has 1D scanner model and 2D scanner model. Refer to the following for the difference in function.

yes: supported, no: not supported, part: a part of function is supported

Basic functions		1D Support	2D Support
openScanner	Open the barcode scanner.	yes	yes
closeScanner	Close the barcode scanner.	yes	yes
isScannerOpen	Check the barcode scanner is opened.	yes	yes
setDefaultAll	Return all barcode scanner settings to default.	yes	yes
getAPIVersion	Get the API version.	yes	yes
getModuleVersion	Get the module version.	yes	yes
getScanResult	Get the last Scan Result.	yes	yes
Notification functions		1D Support	2D Support
setNotificationLED	Set the notification LED behavior.	yes	yes
getNotificationLED	Get the notification LED behavior.	yes	yes
setNotificationVibrator	Set the notification Vibrator behavior.	yes	yes
getNotificationVibrator	Get the notification Vibrator behavior.	yes	yes
setNotificationSound	Set the notification Sound behavior.	yes	yes
getNotificationSound	Get the notification Sound behavior.	yes	yes
Lighting functions		1D Support	2D Support
setLightMode	Set the light mode.	no	yes
getLightMode	Get the light mode.	no	yes
Image capturing functions		1D Support	2D Support
getImageDataSize	Get the size of the data required for capturing image.	no	yes
captureImage	Capture the image.	no	yes
getStreamDataSize	Get the size of data required for streaming.	no	yes
startStream	Start streaming.	no	yes
readStream	Read the stream data.	no	yes
stopStream	Stop streaming.	no	yes
Detailed setting functions		1D Support	2D Support
setSymbologyEnable	Set reading enable/disable of the specified barcode.	part	yes
getSymbologyEnable	Get reading enable/disable of the specified barcode.	part	yes
getSymbologyMaxDefault	Get the default reading maximum number of digits of the specified barcode.	part	yes
getSymbologyMinDefault	Get the default reading minimum number of digits of the specified barcode.	part	yes
setSymbologyMax	Set the reading maximum number of digits of the specified barcode.	part	yes
getSymbologyMax	Get the reading maximum number of digits of the specified barcode.	part	yes
setSymbologyMin	Set the reading minimum number of digits of the specified barcode.	part	yes
getSymbologyMin	Get the reading minimum number of digits of the specified barcode.	part	yes

setSymbologyProperty	Set the value of property setting of the specified barcode.	part	yes
getSymbologyProperty	Get the value of property setting of the specified barcode.	part	yes
setOutputType	Set the output type of the scan result.	yes	yes
getOutputType	Get the output type of the scan result.	yes	yes
setSuffix	Set the suffix type added at the end of the scan result.	yes	yes
getSuffix	Get the suffix type added at the end of the scan result.	yes	yes
setInverseMode	Set the inverse barcode reading mode.	yes	yes
getInverseMode	Get the inverse barcode reading mode.	yes	yes
setTriggerKeyEnable	Set the Trigger key enable/disable.	yes	yes
getTriggerKeyEnable	Get the Trigger key enable/disable.	yes	yes
setTriggerKeyMode	Set the Trigger key mode.	part	yes
getTriggerKeyMode	Get the Trigger key mode.	part	yes
setNumberOfBarcodes	Set the number of barcodes to be scanned in Multi-step scan or Package scan.	yes	yes
getNumberOfBarcodes	Get the number of barcodes to be scanned in Multi-step scan or Package scan.	yes	yes
setDelimiter	Set the delimiter for Package scan.	no	yes
getDelimiter	Get the delimiter for Package scan.	no	yes
setTriggerKeyTimeout	Set the Trigger key timeout.	yes	yes
getTriggerKeyTimeout	Get the Trigger key timeout.	yes	yes
setTriggerKeyOn	Control the behavior of trigger key by software.	yes	yes
setScannerAPO	Set the Auto Power Off(APO) time of the barcode scanner.	no	yes
getScannerAPO	Get the Auto Power Off(APO) time of the barcode scanner.	no	yes
setCenteringWindow	Set enable/disable of the centering window mode.	no	yes
getCenteringWindow	Get enable/disable of the centering window mode.	no	yes
setDetectionAreaSize	Set the size of barcode detection area for the centering window mode.	no	yes
getDetectionAreaSize	Get the size of barcode detection area for the centering window mode.	no	yes
setLaserSwingWidth	Set the laser swing width of 1D scanner.	yes	no
getLaserSwingWidth	Get the laser swing width of 1D scanner.	yes	no
setLaserHighlightMode	Set enable/disable of the laser highlight mode of 1D scanner.	yes	no
getLaserHighlightMode	Get enable/disable of the laser highlight mode of 1D scanner.	yes	no
setInternalParameter	Set internal parameters.	yes	yes
getInternalParameter	Get internal parameters.	yes	yes

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 the 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 the module version.

Parameter:

void None

Return value:

String Module version

Returns the module version on success and Null on failure.

5.2.7. getResult

int getResult(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 "5.2.8ScanResult class (p. 26)".

Parameter:

scanResult ScanResult class

Specify ScannerLibrary.ScanResult class that store Scan Result.

For the member variable of the ScanResult class, refer to "5.2.8ScanResult class (p. 26)".

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();
```

In 1D Scanner model, the aimID and the aimModifier are always 0.

■ 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 the 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 the 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 the 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 the 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 the 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 the 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 light mode.

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 size of the data required 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 detail, refer to "5.2.17 getImageDataSize (p.31)"

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 size of data required 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 "5.2.19getStreamDataSize (p.31)"

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 "5.3.2Code identification table (p.56)" .

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

enable Reading enable/disable

ScannerLibrary.CONSTANT.SYMBOLLOGY_PARAMETER.ENABLE: Read enable

ScannerLibrary.CONSTANT.SYMBOLLOGY_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 "5.3.2Code identification table (p.56)" .

Return value:

ScannerLibrary.CONSTANT.SYMBOLLOGY_PARAMETER.ENABLE: Read enable

ScannerLibrary.CONSTANT.SYMBOLLOGY_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 the default reading maximum 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 "5.3.2Code identification table (p.56)" .

Return value:

Return default reading maximum number of digits on success.

If you specified the barcode that the number of digits can not be changed (i.e. EAN13 etc.), this function returns 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 the 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 "5.3.2Code identification table (p.56)" .

Return value:

Return default reading minimum number of digits on success.

If you specified the barcode that can not be changed the 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 "5.3.2Code identification table (p.56)" .

max reading maximum number of digits

For the number of digits that can be set, refer to "5.3.3Reading digits (p.58)".

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 the reading maximum 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 "5.3.2Code identification table (p.56)" .

Return value:

Return reading maximum number of digits on success.

If you specified the barcode that can not be changed the 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 the reading minimum number of digits of the 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 "5.3.2Code identification table (p.56)" .

min reading minimum number of digits.

For the number of digits that can be set, refer to "5.3.3Reading digits (p.58)".

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 the reading minimum number of digits of the 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 "5.3.2Code identification table (p.56)" .

Return value:

Return the reading minimum number of digits on success.

If you specified the barcode that can not be changed the 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 "5.3.2Code identification table (p.56)" .

propertyNo Property number

Specify the property number.

For the property number, refer to "5.3.4Property of barcode (p.60)".

propertySetting Setting value for property

Set the property setting value.

For the property number, refer to "5.3.4Property of barcode (p.60)".

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 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 "5.3.2Code identification table (p.56)" .

propertyNo Property number

Specify the property number.

For the property number, refer to "5.3.4Property of barcode (p.60)".

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 the 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 the suffix type added at the end of the 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 the 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 the 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 the Trigger key enable/disable.

Parameter:

trggerKeyEnable 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 the 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 the 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)

ScannerLibrary.CONSTANT.TRIGGER_MODE.MULTI_STEP: Multi-step reading

ScannerLibrary.CONSTANT.TRIGGER_MODE.PACKAGE: Package reading

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 the 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.TRIGGER_MODE.MULTI_STEP: Multi-step reading

ScannerLibrary.CONSTANT.TRIGGER_MODE.PACKAGE: Package reading

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED: Unsupported

5.2.43. setNumberOfBarcodes

`int setNumberOfBarcodes(int numberOfBarcodes)`

Function:

Set the number of Barcodes to be scanned in Multi-step scan or Package scan.

Parameter:

`numberOfBarcodes` The number of Barcodes to be scanned in Multi-step scan or Package scan.

Specify a value between 2 and 10.

Return value:

`ScannerLibrary.CONSTANT.RETURN.SUCCESS`: Success

`ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER`: Parameter error

`ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED`: Unsupported error

5.2.44. getNumberOfBarcodes

`int getNumberOfBarcodes(void)`

Function:

Get the number of barcodes to be scanned in Multi-step scan or Package scan.

Parameter:

`void` none

Return value:

Return the number of barcodes to be scanned in Multi-step scan or Package scan.

Default value is 4.

`ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED`: unsupported error

5.2.45. setDelimiter

int setDelimiter(int delimiter)

Function:

Set the delimiter for Package scan.

Parameter:

delimiter The delimiter for Package scan.

Specify a value between 0x00 and 0x7f as ASCII.

When 0x00 is set, it operates None (No delimiter).

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED: Unsupported error

5.2.46. getDelimiter

int getDelimiter(void)

Function:

Get the delimiter for Package scan.

Parameter:

void none

Return value:

Return the delimiter as ASCII for Package scan. Default value is "US(0x1f)".

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED: unsupported error

5.2.47. setTriggerKeyTimeout

int setTriggerKeyTimeout(int triggerKeyTimeout)

Function:

Set the 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

5.2.48. getTriggerKeyTimeout

int getTriggerKeyTimeout(void)

Function:

Get the 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

5.2.49. setTriggerKeyOn

int setTriggerKeyOn(int triggerKeyOn)

Function:

Control the behavior of trigger key by software.

Parameter:

trggerKeyOn status of trigger key

ScannerLibrary.CONSTANT.CONTROL.TRIGGER_OFF: Trigger key is released virtually

ScannerLibrary.CONSTANT.CONTROL.TRIGGER_ON: Trigger key is pressed virtually

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED: Unsupported

5.2.50. setScannerAPO

int setScannerAPO(int scannerAPOTime)

Function:

Set the Auto Power Off(APO) time of the barcode scanner.

Parameter:

scannerAPOTime The APO time of the barcode scanner.

Specify a value between 0 and 65535 in seconds.

When set to 0, APO is disabled.

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED: Unsupported

5.2.51. getScannerAPO

int getScannerAPO(void)

Function:

Get the Auto Power Off(APO) 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

5.2.52. setCenteringWindow

int setCenteringWindow(int centeringWindow)

Function:

Set enable/disable of the centering window mode.

Parameter:

centeringWindow Enable/disable of the centering window mode.

ScannerLibrary.CONSTANT.CENTERING_WINDOW_MODE.DISABLE: Centering window mode disable

ScannerLibrary.CONSTANT.CENTERING_WINDOW_MODE.ENABLE: Centering window mode enable

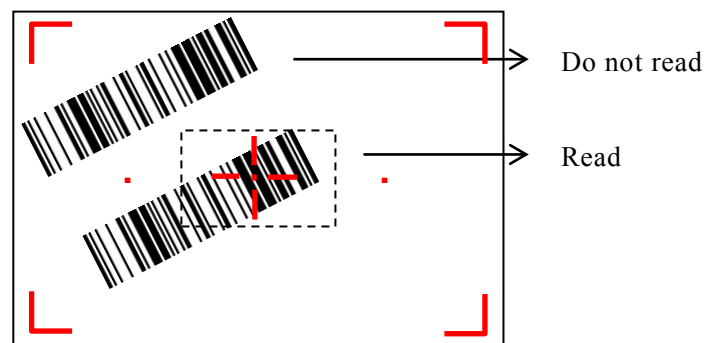
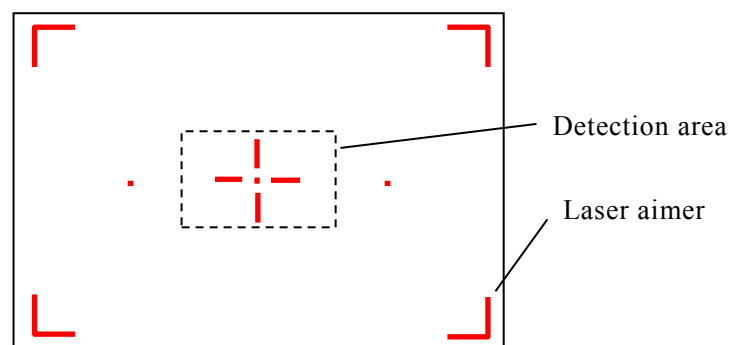
Refer to the figure below for the barcode that can be read while the center reading mode is in effect. If part of the barcode is included in the detection area, read the barcode.

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER: Parameter error

ERROR_UNSUPPORTED: unsupported error



5.2.53. getCenteringWindow

int getCenteringWindow(void)

Function:

Get enable/disable of 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

5.2.54. 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 for the centering window
mode.

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.

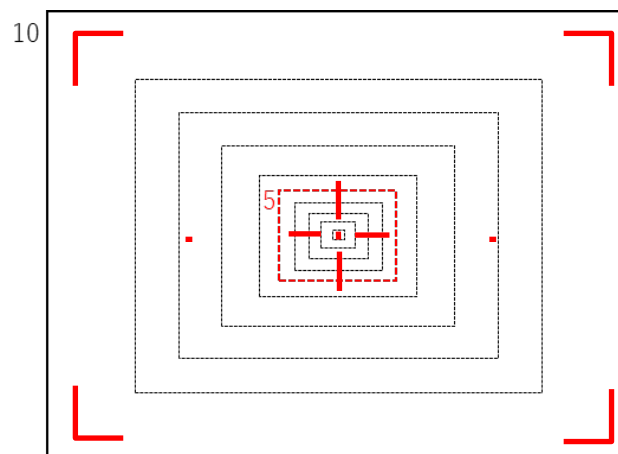
Please change the value according to the actual use environment.

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED: unsupported error



Detection area size as a guide.

5.2.55. 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

5.2.56. setLaserSwingWidth

int setLaserSwingWidth(int laserSwingWidth)

Function:

Set the laser swing width of 1D scanner.

Parameter:

laserSwingWidth swing width

ScannerLibrary.CONSTANT.SWING_WIDTH.MAX(default): Swing width MAX

ScannerLibrary.CONSTANT.SWING_WIDTH.WIDE: Swing width WIDE

ScannerLibrary.CONSTANT.SWING_WIDTH.MIDDLE: Swing width MIDDLE

ScannerLibrary.CONSTANT.SWING_WIDTH.NARROW: Swing width NARROW

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED: unsupported error

5.2.57. getLaserSwingWidth

int getLaserSwingWidth(void)

Function:

Get the laser swing width of 1D scanner.

Parameter:

void none

Return value:

ScannerLibrary.CONSTANT.SWING_WIDTH.MAX(default): Swing width MAX

ScannerLibrary.CONSTANT.SWING_WIDTH.WIDE: Swing width WIDE

ScannerLibrary.CONSTANT.SWING_WIDTH.MIDDLE: Swing width MIDDLE

ScannerLibrary.CONSTANT.SWING_WIDTH.NARROW: Swing width NARROW

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED: Unsupported error

5.2.58. setLaserHighlightMode

int setLaserHighlightMode(int enable)

Function:

Set enable/disable of the laser highlight mode of 1D scanner.

Parameter:

enable Enable/disable of the laser highlight mode

ScannerLibrary.CONSTANT.LASER_HIGHLIGHT_MODE.DISABLE (default): Laser highlight mode disable

ScannerLibrary.CONSTANT.LASER_HIGHLIGHT_MODE.ENABLE: Laser highlight mode enable

Return value:

ScannerLibrary.CONSTANT.RETURN.SUCCESS: Success

ScannerLibrary.CONSTANT.RETURN.ERROR_PARAMETER: Parameter error

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED: unsupported error

5.2.59. getLaserHighlightMode

int getLaserHighlightMode(void)

Function:

Get enable/disable of the laser highlight mode of 1D scanner.

Parameter:

void none

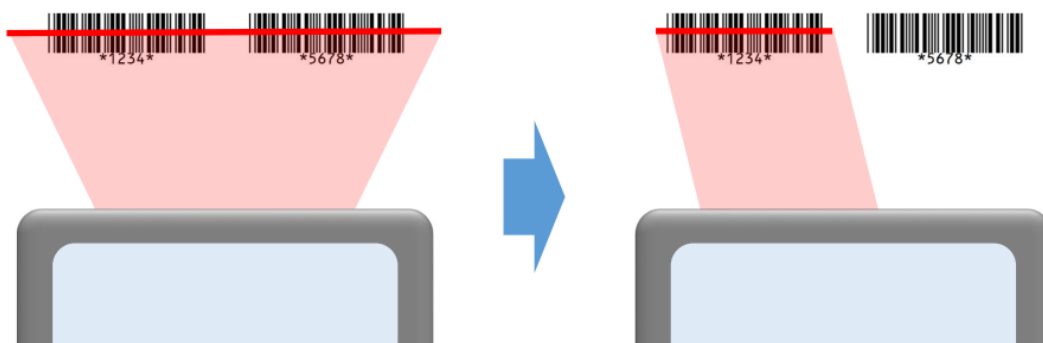
Return value:

ScannerLibrary.CONSTANT.LASER_HIGHLIGHT_MODE.DISABLE: Laser highlight mode disable

ScannerLibrary.CONSTANT.LASER_HIGHLIGHT_MODE.ENABLE: Laser highlight mode enable

ScannerLibrary.CONSTANT.RETURN.ERROR_UNSUPPORTED: Unsupported error

When highlight mode is enabled, the barcode read is emphasized. It is useful for checking the barcode read when two or more barcodes are placed nearby.



5.2.60. setInternalParameter

int setInternalParameter(int tag, int value)

Function:

Set internal parameters. Detailed specifications are not disclosed.

Parameter:

Non-disclosure

Return value:

Non-disclosure

5.2.61. setInternalParameter

int setInternalParameter(byte[] command)

Function:

Set internal parameters. Detailed specifications are not disclosed.

Parameter:

Non-disclosure

Return value:

Non-disclosure

5.2.62. 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 pressed virtually.

Symbol parameter constants

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

Class name: ScannerLibrary.CONSTANT.SYMBOLLOGY_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

Constant 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 continuously during trigger key is pressed.
MULTI_STEP	Multi-step reading. This method is for scanning a specified number of barcodes. Once scanning for the specified number of bar codes has been completed, the scanner closes and will not scan again until reopened. Also, the same bar codes that have been scanned previously cannot be scanned again.
PACKAGE	Package reading. Scanning continuously reads multiple symbols until when the Trigger key is released and then outputs a result of reading all the symbols.

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"

Swing width constant

List of swing width constants. Set swing width of laser.

Class name: ScannerLibrary.CONSTANT.SWING_WIDTH

Constant field	Description
MAX	Swing width MAX
WIDE	Swing width WIDE
MIDDLE	Swing width MIDDLE
NARROW	Swing width NARROW

Highlight mode constant

List of highlight mode constants. Enable / disable highlight mode of laser.

Class name: ScannerLibrary.CONSTANT.LASER_HIGHLIGHT_MODE

Constant field	Description
DISABLE	Disable "Highlight mode"
ENABLE	Enable "Highlight 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

1D Scanner Reading digit

barcode name	Default		Configurable	
	Minimum digits	Maximum digits	Minimum digits	Maximum digits
Codabar	2	60	2	60
Code128 (a)	1	80	1	80
Code32	-	-	-	-
Code39	2	48	1	48
Code93	1	80 (b)	1	80 (b)
EAN13	-	-	-	-
EAN8	-	-	-	-
GS1 128(EAN128) (a)	-	-	-	-
GS1 DataBar(RSS)	-	-	- (c)	- (c)
ITF (Interleaved 2 of 5)	4	80	2	80
MSI	4	48	4	48
UPC-A	-	-	-	-
UPC-E0 / UPC-E1	-	-	-	-

(a): Reading digits of Code128 and GS1 128 are shared. To specify the number of digits of GS1 128, set the number of digits of Code128.

(b): Maximum 76 digits can be read. 80 digit can be set for application compatibility.

(c): Maximum 74 digits can be read. 80 digit can be set for application compatibility.

2D Scanner Reading digit

barcode name	Default		Configurable	
	Minimum digits	Maximum digits	Minimum digits	Maximum digits
Aztec	1	3832	1	3832
Codabar	2	60	2	60
Codablock F	1	2048	1	2048
Code128	1	80	1	80
Code32	-	-	-	-
Code39	2	48	1	48
Code93	1	80	1	80
Composite	1	300	1	300
DataMatrix	1	3166	1	3166
EAN13	-	-	-	-
EAN8	-	-	-	-
GS1 128(EAN128)	1	80	1	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

Property of barcode vary depending on the barcode scanner model. Refer to the table below for the settable functions.

Property of barcode for 1D scanner

Barcode name	Property number	Property name	Possible values
Codabar	0	CLSI Editing	0: Disable CLSI Ending. (default) 1: Enable CLSI Ending. When enabled and read a 14-character Codabar, strips the Start/Stop characters and inserts a space(0x20) after 1st, 5th, 10th characters.
	1	Output Start/Stop Character	0: Start/Stop code is not output. (Default) 1: Start/Stop code is output.
	2	Upper Case Start/Stop Character	Controlling Start/Stop Character to be uppercase / lowercase. When property number 2 is set to 1(enable), the Start/Stop Character becomes uppercase. (default)
	3	Lower Case Start/Stop Character	When property number 3 is set to 1(enable), the Start/Stop Character becomes lowercase.
	4	Check Digit Calculate	0: Read regardless of presence or absence of check character (default) 1: Only bar codes with check characters are read.
Code128	-	-	-
Code39	0	Convert Code 39 to Code 32	Make the settings for Code32 format. When property number 0 is set to 1(enable), Code39 barcode created in Code32 format is read as Code32. Default is 0(disable).
	1	Code 32 Prefix	When property number 0 is set to 1(enable) and property number 1 is set to 1(enable), insert a "A" to the reading result. Default is 0(disable).
	2	Check Digit Calculate	0: Read regardless of presence or absence of check character (default) 1: Only bar codes with check characters are read.
	3	Output Check Digit	0: Check character is not output. (Default) 1: Check character is output.
	4	Enable Code 39 Full ASCII	0: Full ASCII conversion is not performed. (Default) 1: Full ASCII conversion is performed.
Code93	-	-	-
EAN13	0 ※1	UPC/EAN Supplemental	0: Read the barcode regardless of add-on presence. The add-on part is not output. (default) 1: Read the barcode regardless of add-on presence. The add-on part is output.

	1 ※1	UPC/EAN Security Level 0	<p>Select the security level according to the barcode quality.</p> <p>When property number 1 is set to 1(enable), output the reading result as it is. (default)</p> <p>When property number 2 is set to 1(enable), output the reading result if the same reading results twice.</p> <p>When property number 3 is set to 1(enable), output the reading result if the same reading results three times.</p> <p>When property number 4 is set to 1(enable), output the reading result if the same reading results four times.</p>
	2 ※1	UPC/EAN Security Level 1	
	3 ※1	UPC/EAN Security Level 2	
	4 ※1	UPC/EAN Security Level 3	
EAN8	0	EAN Zero Extend	<p>Make the settings for EAN13 format.</p> <p>When property number 0 is set to 1(enable), output the 13-character reading result as it is. (default)</p> <p>When property number 0 is set to 1(enable) and property number 1 is set to 1(enable), the symbologyID and the symbologyName acquired at the time of reading success are those of EAN13. Default is 0 (disable).</p>
	1	Convert EAN-8 to EAN-13	
	2 ※1	UPC/EAN Supplemental	<p>0: Read the barcode regardless of add-on presence. The add-on part is not output. (default)</p> <p>1: Read the barcode regardless of add-on presence. The add-on part is output.</p>
	3 ※1	UPC/EAN Security Level 0	<p>Select the security level according to the barcode quality.</p> <p>When property number 3 is set to 1(enable), output the reading result as it is. (default)</p> <p>When property number 4 is set to 1(enable), output the reading result if the same reading results twice.</p> <p>When property number 5 is set to 1(enable), output the reading result if the same reading results three times.</p> <p>When property number 6 is set to 1(enable), output the reading result if the same reading results four times.</p>
	4 ※1	UPC/EAN Security Level 1	
	5 ※1	UPC/EAN Security Level 2	
	6 ※1	UPC/EAN Security Level 3	
GS1 128(EAN128)	-	-	-

GS1 DataBar(RSS)	0	Convert GS1 DataBar to UPC/EAN	0: When read GS1 DataBar, output the reading result as it is. (default) 1: When read GS1 DataBar, output the result in UPC/EAN format. If the character following the first "010", remove the first three characters and read as EAN13. If there are two or more and five or less consecutive "0" of the next character of the first "01", remove the first four characters and read as UPC-A. In other cases, remove the first three characters, recalculate the check digit, and read as EAN13.
	1	RSE Symbology enable	0: Disable reading of GS1 DataBar Expanded. 1: Enable reading of GS1 DataBar Expanded. (Default)
	2	RSL Symbology enable	0: Disable reading of GS1 DataBar Limited. 1: Enable reading of GS1 DataBar Limited. (Default)
	3	RSS Symbology enable	0: Disable reading of GS1 DataBar Omnidirectional, Truncated, Stacked, Stacked Omnidirectional. 1: Enable reading of GS1 DataBar Omnidirectional, Truncated, Stacked, Stacked Omnidirectional. (Default)
ITF (Interleaved 2 of 5)	0	No Check Digit Calculate	Select the algorithm of check digit according to the barcode. When property number 0 is set to 1(enable), does not calculate check digit. (default) When property number 1 is set to 1(enable), calculate check digit by USS(Uniform Symbology Specification modulus 10/weight 3). When property number 2 is set to 1(enable), calculate check digit by OPCC(Optical Product Code Council modulus 10/weight 2).
	1	USS Check Digit Calculate	
	2	OPCC Check Digit Calculate	
	3	Output Check Digit	0: Check character is not output. (Default) 1: Check character is output.
	4	Convert I 2 of 5 to EAN-13	0: When read ITF, output the reading result as it is. (default) 1: When read ITF, output the result in EAN13 format. 14-character ITF read as EAN13. Valid only for ITF with first character "0" and check digit.
MSI	0	One Check Digit Calculate	Select the number of check digit according to the barcode.

	1	Two Check Digit Calculate	When property number 0 is set to 1(enable), at least one check digit is always required. (default) When property number 1 is set to 1(enable), two check digits is always required.
	2	Output Check Digit	0: Check character is not output. (Default) 1: Check character is output.
	3	Mod 10/Mod 11 Two Check Digit	Select the algorithm when "Two Check Digit Calculate" is set to 1(enable). Make one of the algorithms set to 1(enable) according to the barcode. When property number 3 is set to 1(enable), calculate check digit by modulus 10/modulus 11. When property number 4 is set to 1(enable), calculate check digit by modulus 10/modulus 10. (default) When property number 5 is set to 1(enable), calculate check digit by modulus 11/modulus 10.
	4	Mod 10/Mod 10 Two Check Digit	
	5	Mod 11/Mod 10 Two Check Digit	
UPC-A	0	Output Check Digit	0: Check character is not output. (Default) 1: Check character is output.
	1	No Output Preamble Character	Controlling the character to be output when read UPC-A. Select the settings according to the reading barcode. When property number 1 is set to 1(enable), output the reading result as it is. When property number 2 is set to 1(enable), output the reading result with adding system character. (default) When property number 3 is set to 1(enable), output the reading result with adding system character and country code. The country code to be added is fixed to 0.
	2	Output System Character	
	3	Output System Character and Country Code	
	4 ※1	UPC/EAN Supplemental	0: Read the barcode regardless of add-on presence. The add-on part is not output. (default) 1: Read the barcode regardless of add-on presence. The add-on part is output.
	5 ※1	UPC/EAN Security Level 0	Select the security level according to the barcode quality. When property number 5 is set to 1(enable), output the reading result as it is. (default) When property number 6 is set to 1(enable), output the reading result if the same reading results twice. When property number 7 is set to
	6 ※1	UPC/EAN Security Level 1	
	7 ※1	UPC/EAN Security Level 2	

	8 ※1	UPC/EAN Security Level 3	1(enable), output the reading result if the same reading results three times. When property number 8 is set to 1(enable), output the reading result if the same reading results four times.
UPC-E0 / UPC-E1	0	Output Check Digit	0: Check character is not output. (Default) 1: Check character is output.
	1	No Output Preamble Character	Control the output character to be added when read UPC-E. Select the settings according to the usage environment. When property number 1 is set to 1(enable), output the reading result as it is.
	2	Output System Character	
	3	Output System Character and Country Code	When property number 2 is set to 1(enable), output the reading result with adding system character. (default) When property number 3 is set to 1(enable), output the reading result with adding system character and country code. The country code to be added is fixed to 0.
	4	Convert UPC-E to UPC-A	0: When read UPC-E, output the reading result as it is. (default) 1: When read UPC-E, output the result in UPC-A format.
	5 ※1	UPC/EAN Supplemental	0: Read the barcode regardless of add-on presence. The add-on part is not output. (default) 1: Read the barcode regardless of add-on presence. The add-on part is output.
	6 ※1	UPC/EAN Security Level 0	Select the security level according to the barcode quality. When property number 6 is set to 1(enable), output the reading result as it is. (default) When property number 7 is set to 1(enable), output the reading result if the same reading results twice. When property number 8 is set to 1(enable), output the reading result if the same reading results three times. When property number 9 is set to 1(enable), output the reading result if the same reading results four times.
	7 ※1	UPC/EAN Security Level 1	
	8 ※1	UPC/EAN Security Level 2	
	9 ※1	UPC/EAN Security Level 3	

Remarks)

※1 These settings are sheared in UPC/EAN.

Property of barcode for 2D scanner

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. (Default) 1: Outputs the check character.
	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)

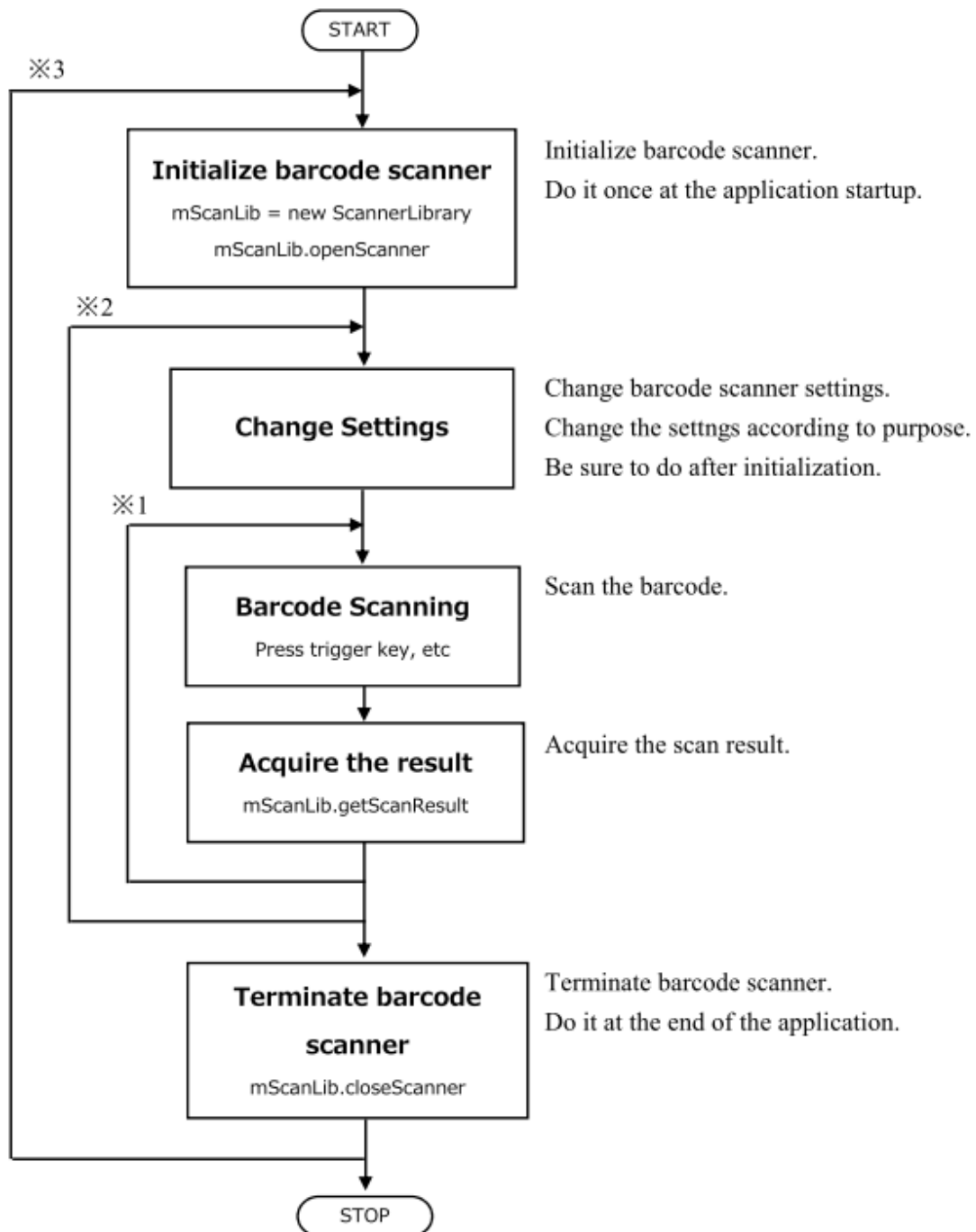
Remarks)

※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 scanning flow



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

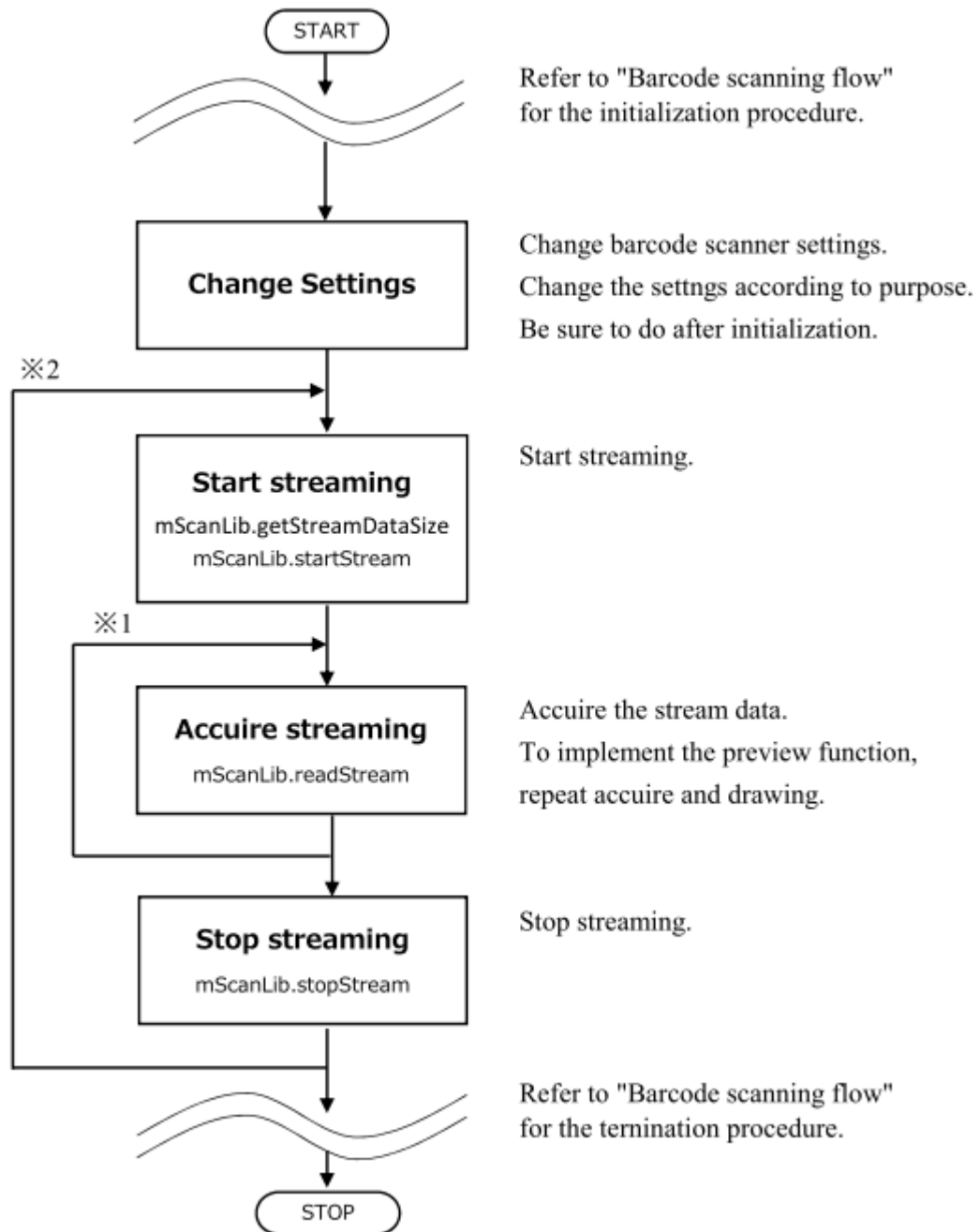
※2 Repeat this process to read barcodes with different settings.

※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, refer to "5.2.39

5.4.2. Streaming flow



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

※2 Repeat this process to resume streaming.

5.4.3. Image capturing flow

