



High-powered Handy Scanner

DENSO SP1 SDK for Android Reference Manual

Copyright © 2018 DENSO WAVE INCORPORATED
All rights reserved.

BHT, QR code, iQR code, SQRC and Qbdirect are registered trademarks of DENSO WAVE INCORPORATED.

Bluetooth® is registered trademarks of trademarks rights holders, DENSO WAVE INCORPORATED use Bluetooth® based on license.

All products and company names in this document are trademarks of their respective holders.
™, ® mark is not always specified in this document.

Contents

Chapter 0.	Introduction	4
Chapter 1.	Precaution	5
1.1.	Precautions for RF Tag Processing.....	5
1.2.	Precautions for the response from the scanner	5
1.3.	Firmware update API	5
Chapter 2.	Application Development environment.....	8
Chapter 3.	API Operaion Flow	9
3.1.	Selecting a method for connecting SP1 to a device	9
One-to-one operation		9
Many-to-many operation.....		10
Chapter 4.	Common API.....	11
4.1.	Outline	11
4.2.	Common API	12
CommScanner#upgradeFirmware		12
CommScanner#buzzer		13
CommScanner#setLED.....		14
CommScanner#getSerialNum.....		15
CommScanner#getPartNum.....		16
CommScanner#getRemainingBattery		17
CommScanner#setParams.....		18
CommScanner#saveParams		19
CommScanner#getParams.....		20
CommScanner#setBtSettings		21
CommScanner#getBtSettings		22
CommScanner#init		23
CommException#getErrorCode		24
CommScanner#getType.....		25
CommScanner#getVersion.....		26
CommScanner#addStatusListener		27
CommScanner#removeStatusListener		28
ScannerStatusListener#onScannerStatusChanged.....		29
CommStatusChangedEvent#getStatus.....		30
CommManager#addAcceptStatusListener		31
CommManager#removeAcceptStatusListener		32
CommManager#startAccept		33
CommManager#endAccept		34
ScannerAcceptStatusListener#OnScannerAppeared.....		35
CommScanner#claim		36
CommScanner#close		37
CommManager#getScanners.....		38
CommScanner#getBTAddress		39
CommScanner#getBTLocalName.....		40
CommScanner#getModel		41
CommScanner#getBarcodeScanner.....		42
CommScanner#getRFIDScanner		43
CommScanner#addKeyStatusListener		44
CommScanner#removeKeyStatusListener		45
ScannerKeyStatusListener#onScannerKeyStatusChanged		46
CommKeyStatusChangedEvent#getKeyName.....		47
CommKeyStatusChangedEvent#getKeyStatus		48
CommScanner#getCurrentKeyStatus.....		49

CommScanner#getRegion	50
Sample code 1	51
CommScannerParams class	52
CommScannerBtSettings class	54
Chapter 5. RFID	55
5.1. Outline	55
5.1.1. RFID data	55
5.1.2. Batch mode	55
5.2. RFID API	56
RFIDScanner#openInventory	56
RFIDScanner#openInventory(Index)	57
RFIDScanner#openRead(bank, addr, size, pwd)	58
RFIDScanner#openRead(bank, addr, size, pwd, UII)	59
RFIDScanner#openRead(bank, addr, size, pwd, index)	60
RFIDScanner#close	61
RFIDScanner#writeOneTag	62
RFIDScanner#lockOneTag	64
RFIDScanner#killOneTag	66
RFIDScanner#setSettings	67
RFIDScanner#getSettings	68
RFIDScanner#setAutoLinkProfile	69
RFIDScanner#getAutoLinkProfile	70
RFIDScanner#setFilter	71
RFIDScanner#clearFilter	72
RFIDScanner#clearDoubleReadingBuffer	73
RFIDScanner#setDataDelegate	74
RFIDDataDelegate#onRFIDDataReceived	75
RFIDDataReceivedEvent#getRFIDData	76
RFIDData#getData	77
RFIDData#getRSSI	78
RFIDData#getPC	79
RFIDData#getAntenna	80
RFIDData#getPolarization	81
RFIDData#getUII	82
RFIDData#getIndex	83
RFIDData#getResult	84
RFIDData#getCh	85
RFIDData#getPhase	86
RFIDScanner#getCount	87
RFIDScanner#pullData	88
RFIDScanner#setResponse	89
RFIDScanner#getResponse	90
RFIDException#getErrorCode	91
Sample code 2	93
RFIDScannerFilter Class	95
RFIDScannerSettings Class	96
RFIDScannerResponse Class	99
Chapter 6. Barcode	100
6.1. Outline	100
6.1.1. Enable Reading	100
6.1.2. Barcode Data	101
6.1.3. SQRC	101
6.2. Barcode API	102
BarcodeScanner#openReader	102

BarcodeScanner#closeReader	103
BarcodeScanner#setSettings.....	104
BarcodeScanner#getSettings.....	105
BarcodeScanner#setDataDelegate.....	106
BarcodeDataDelegate#onBarcodeDataReceived.....	107
BarcodeDataReceivedEvent#getBarcodeData	108
BarcodeData#getData	109
BarcodeData#getSymbologyDenso.....	110
BarcodeData#getSymbologyAim	111
BarcodeData#getPrivateDataLength	114
BarcodeException#getErrorCode	115
Sample code 3	116
BarcodeScannerSettings Class.....	118

Chapter 0. Introduction

This manual explains how to develop application with SP1. Android standard API can be used for functions other than unique to SP1 not described in this document.

Functions that can be confirmed with the SDK bounded library and features planned to be supported in the future are described in this manual.

Functions that will be supported in the future are described as *to be added in this document.

The contents of this document are subject to change without prior notice.

This manual is intended for programmers who are already familiar with Android and the Android Studio. For details of Android products and its development methods, or programming languages, refer to the manuals available in bookstores.

Chapter 1. Precaution

1.1. Precautions for RF Tag Processing

Since contactless communication is performed with RF tags, it is necessary to keep the following precautions in mind.

■ Reading from RF tags

- (1) If an RF tag does not respond due to its defect or it does not receive sufficient power due to wave interference, the scanner may be unable to read it even if it is located within the reading area.
- (2) If an irrelevant RF tag is located within the reading area, the scanner may read it.
- (3) When repeating the read process, previously read RF tags may be read again if they remain within the reading area.
- (4) If two or more RF tags have the same UII, they will be handled as a single RF tag.

Therefore, you may get a different result than expected if you perform the read process in a way such as to complete it for the required number of tags. To solve this problem, take the measures described below for the upper-level application as required to read from RF tags.

In cases (1) to (3):

- Construct a system capable of checking the number of read RF tags and the contents. The results of reading tags (number of RF tags and contents) should be checked by the operator. Re-read only the RF tag that could not be read or register by another means.

In case (4):

- Do NOT use RF tags having the same UUI data.

1.2. Precautions for the response from the scanner

Depending on the host device and environment, the response from the scanner may be delayed. Make sure that the scanner can communicate without problem in advance.

1.3. Firmware update API

In the case of operating with a large number of units, it is recommended to implement the SP1 firmware upgrade function (CommScanner # upgrade Firmware).

If SP1 firmware is improved, it can be rewritten without stopping the operation, and the rewriting cost can be reduced.

■ Update history

Version	Description	Supported SDK Ver.	Supported OS Ver.
1st	-		
2nd	Add Precautions [Barcode] Add description of BarcodeScannerSettings Class of QR [RFID] Add Extended power save mode of RFIDScannerSettings Class. Delete ErrorCode.SCANNER_CHARGING of getCount and pullData exception. They can work during charging. Change the condition for enables double-reading prevention during Power on.	1.0.2	1.03
3rd	Change the Sample code 2.(Delete the getCommand()) [Common API] Add CommScanner#upgradeFirmware. [RFID] Add description of Batch mode Add note on Read / Write power level(dBm) settings. Add RFIDData#getResult. Add description of writeOneTag. [Barcode] Add the SQRC cpde related Settings.	1.0.4	1.05
4th	[Common API] Add CommScanner#addKeyStatusListener Add CommScanner#removeKeyStatusListener Add ScannerKeyStatusListener#onScannerKeyStatusChanged Add CommKeyStatusChangedEvent#getKeyName Add CommKeyStatusChangedEvent#getKeyStatus Add CommScanner#getCurrentKeyStatus	1.0.5	1.06
5th	[RFID] Add RFIDData#getCh Add RFIDData#getPhase Add RFIDScanner#setResponse Add RFIDScanner#getResponse Add RFIDScannerResponse Class [Common API] Add CommScanner#getRegion Enable CommScanner#setBtSettings, CommScanner#getBtSettings Add ReConnect.ANY of CommScannerParamas Class Add settings of ComScannerBtSettings Class	1.0.6	1.11

Version	Description	Supported SDK Ver.	Supported OS Ver.
6th	[API Operation Flow] Create new [RFID] Add description of setFilter Add description of getRSSI Add description of getUII Add sample code of writeOneTag [Barcode] Add mirrorReflection of BarcodeScannerSettings class Add notes of BarcodeScannerSettings class	1.0.7	1.12
7th	[Application Development environment] Change the library import method to website reference(due to Android Studio specification change) [API Operation Flow] Change Many-to-many operation sample code(Check the current value and then change to the setting). [RFID] Add RFIDScanner#setAutoLinkProfile(Enable / Disable Autopilot) Add RFIDScanner#getAutoLinkProfile(Get Autopilot setting value)	1.0.9	1.14

Chapter 2. Application Development environment

■ Application development tools

• The following development tools are required for Android application development.

- Java SE development kit (JDK, Java SE 8 and over recommend)
- Android SDK
- Android Studio

*To get the above software tools, please refer to their official sites.

*Developers are assumed to have knowledge of Android programming.

■ Import library

The procedure for importing the library may differ depending on the specifications of each version of Android Studio.

Please check the specifications of Android Studio for the procedure in your environment. (Reference)

<https://developer.android.com/studio/projects/android-library?hl=en#psd-add-library-dependency>

* The minSdkVersion of this library is 23. TargetSDKVersion is 30.

■ Add the following permissions to AndroidManifest.xml to use Bluetooth device.

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

Chapter 3. API Operaion Flow

The following describes the flow up to connection.

3.1. Selecting a method for connecting SP1 to a device

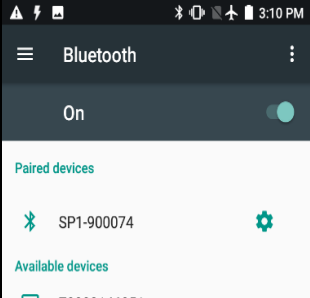
The connection method can be selected according to the on-site operation status.

One-to-one operation (One SP1 correspods to one Android device)

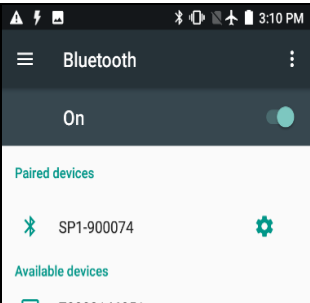
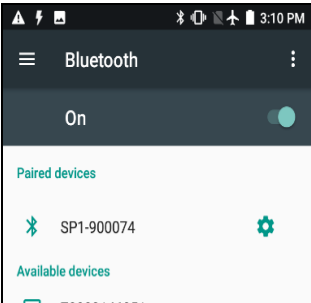
i) When using Target Setting app bundled with the SDK at the first connection

Initial connection	Connection after the initial one
Connect with Target Setting app  *Factory default SP1 Bluetooth mode is AUTO. After first connection, AUTO works as MASTER. Target Setting app pairs and connects to SP1. After that, SP1 works as MASTER (Setting value remains AUTO).	<pre>addAcceptStatusListener(listener); // add listener startAccept(); // start waiting for connection</pre> ↓ When SP1 tries to connect to an Android device OnScannerAppeared event is notified. Run the following processing in the OnScannerAppeared <pre>endAccept(); // end waiting scanner.claim(); // connection establish removeAcceptStatusListener(listener); // remove listener</pre> ↓ Connection establish

ii) When using the Android standard Bluetooth setting at the first pairing

Initial connection	Connection after the initial one
Pair with SP1 on Bluetooth Setting screen of Android standard.  ↓ <pre>// Get paired list List<CommScanner> listCommScanner; listCommScanner = CommManager.getScanners(); if (listCommScanner != null) { for (CommScanner scanner: listCommScanner) { // Check if SP1 name matches target device name if (scanner.getBTLocalName().equals(name)) { scanner.claim(); // connection establish } } }</pre> *Factory default SP1 Bluetooth mode is AUTO. After first connection, AUTO works as MASTER. getScanners and claim connect to SP1. After that, SP1 works as MASTER (Setting value remains AUTO).	<pre>addAcceptStatusListener(listener); // add listener startAccept(); // start waiting for connection</pre> ↓ When SP1 tries to connect to an Android device OnScannerAppeared event is notified. Run the following processing in the OnScannerAppeared <pre>endAccept(); // end waiting scanner.claim(); // connection establish removeAcceptStatusListener(listener); // remove listener</pre> ↓ Connection establish

Many-to-many operation (SP1 and Android devices operate without fixing their combination)

Initial connection	Connection after the initial one
Pair with SP1 on Bluetooth Setting screen of Android standard.  <pre> // Get paired list List<CommScanner> listCommScanner; listCommScanner = CommManager.getScanners(); if (listCommScanner != null) { for (CommScanner scanner: listCommScanner) { // Check if SP1 name matches target device name if (scanner.getBTLocalName().equals(name)) { scanner.claim(); // connection establish // Get SP1's Bluetooth settings CommScannerBtSettings btSet=scanner.getBtSettings(); If(btSet.mode != CommScannerBtSettings.Mode.SLAVE){ // Set SP1 bluetooth mode to SLAVE btSet.mode = CommScannerBtSettings.Mode.SLAVE; //Reflect settings to SP1 scanner.setBtSettings(btSet); } } } } </pre> *Factory default SP1 Bluetooth mode is AUTO (SLAVE). After establishing a connection, change to SLAVE mode at all times. This will work with SLAVE mode the next time connection.	Pair with SP1 on Bluetooth Setting screen of Android standard.  <pre> // Get paired list List<CommScanner> listCommScanner; listCommScanner = CommManager.getScanners(); if (listCommScanner != null) { for (CommScanner scanner: listCommScanner) { // Check if SP1 name matches target device name if (scanner.getBTLocalName().equals(name)) { scanner.claim(); // connection establish // Get SP1's Bluetooth settings CommScannerBtSettings btSet=scanner.getBtSettings(); If(btSet.mode != CommScannerBtSettings.Mode.SLAVE){ // Set SP1 bluetooth mode to SLAVE btSet.mode = CommScannerBtSettings.Mode.SLAVE; //Reflect settings to SP1 scanner.setBtSettings(btSet); } } } } </pre> *After establishing a connection, if the setting is not SLAVE mode, it is set to SLAVE mode. Because the combination is not fixed, it is difficult for the application to determine whether the connected SP1 is in the factory default state or the second or later connection.

Chapter 4. Common API

4.1. Outline

These API are for getting information from SP1 device, connecting to Bluetooth, setting parameters.

4.2. Common API

CommScanner#upgradeFirmware

Description

Upgrade SP1 firmware. Open the SY3 file stream.

This is synchronization function.

After normal end, SP1 will automatically reboot.

At that time the connection will be lost, please reconnect again.

Please run in a place with good communication environment.

Syntax

```
void upgradeFirmware (InputStream in);
```

Parameters

in

SP1 Firmware file stream

Return value

None

Throws

CommException

CommException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode.INVALID_COMMAND	Invalid parameters or syntax
ErrorCode.COMMUNICATION_TIMEOUT	Communication timeout
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.INVALID_MODEL	Firmware file stream does not match the model of the scanner
ErrorCode.INVALID_VERSION	Downgrade firmware
ErrorCode.INVALID_FILE	Firmware file is invalid
ErrorCode.SCANNER_WRITE_FIRMWARE	Firmware writing error occurred. SP1 automatically turns off the power
ErrorCode.SCANNER_INCOMPLETE_UPGRADE	When UPGRADE terminates with not completed due to communication error etc. SP1 automatically turns off the power.
ErrorCode.NOT_SUPPORT_COMMAND	unsupported command

CommScanner#buzzer

Description

Sound the buzzer

Syntax

```
void buzzer (  
    CommBuzzerType b_type  
);
```

Parameters

b_type

CommBuzzerType.B1

Sound the buzzer.

Sound duration depends on buzzerDuration of CommScannerParams.

Volume and Tone depends on buzzerVolume and buzzerTone of CommScannerParams.

CommBuzzerType.B2

Sound the buzzer for 120ms.

Volume and Tone depends on buzzerVolume and buzzerTone of CommScannerParams.

CommBuzzerType.B3

Sound the buzzer for 240ms.

Volume and Tone depends on buzzerVolume and buzzerTone of CommScannerParams.

Return value

None

Throws

CommException

CommException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode. INVALID_COMMAND	Invalid parameters or syntax
ErrorCode. INVALID_STATUS	Invalid status error
ErrorCode. NOT_SUPPORT_COMMAND	unsupported command

CommScanner#setLED

Description

Light the LED with designated color.

SP1 supports below combination only.

- LED1 (Power LED of SP1) supports Orange, Green, Red.
- LED2 (Pairing LED of SP1) supports Blue.
- LED3 (Indicator LED of SP1) supports Blue, Red.

Syntax

```
void setLED (  
    CommLEDTyp led_type  
    CommLEDCol color  
);
```

Parameters

led_type

CommLEDTyp.LED1

Light LED1 about 500ms.

CommLEDTyp.LED2

Light LED2 about 500ms.

CommLEDTyp.LED3

Light LED3 about 500ms.

color

CommLEDCol.RED

Light in Red.

CommLEDCol.ORANGE

Light in Orange.

CommLEDCol.BLUE

Light in Blue.

CommLEDCol.GREEN

Light in Green.

Return value

None

Throws

CommException

CommException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode. INVALID_COMMAND	Invalid parameters or syntax
ErrorCode. INVALID_STATUS	Invalid status error
ErrorCode. NOT_SUPPORT_COMMAND	unsupported command

CommScanner#getSerialNum

Description

Get serial number.

Syntax

```
String getSerialNum ();
```

Parameters

none

Return value

Serial number

Throws

This API does not throw an exception (no error occurs).

If you call this API before claim, the API returns Null.

Notes

Sample coding

CommScanner#getPartNum

Description

Get part number.

Syntax

```
String getPartNum ();
```

Parameters

none

Return value

Part number

Throws

This API does not throw an exception (no error occurs).

If you call this API before claim, the API returns Null.

Notes

Sample coding

CommScanner#getRemainingBattery

Description

Get remaining battery level.

Syntax

```
CommBattery getRemainingBattery ();
```

Parameters

none

Return value

CommBattery.UNDER10
Less than 10% of battery remaining.

CommBattery.UNDER40
Less than 40% of battery remaining.

CommBattery.40_AND_OVER
40% and over of battery remaining.

Throws

CommException
CommException#getErrorCode() can get the error.
The errors of this API are as follows.

Error label	Description
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.NOT_SUPPORT_COMMAND	unsupported command

Notes

Sample coding

CommScanner#setParams

Description

Set common parameters to SP1 device.

Syntax

```
void setParams (  
    CommScannerParams params  
);
```

Parameters

params

CommScannerParams class variable

Return value

None

Throws

NullPointerException

If params is null.

CommException

CommException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode. INVALID_COMMAND	Invalid parameters or syntax
ErrorCode. INVALID_STATUS	Invalid status error
ErrorCode. NOT_SUPPORT_COMMAND	unsupported command

Sample coding

Refer to the sample code 1.

CommScanner#saveParams

Description

Save common parameters right now.

Syntax

```
void saveParams ();
```

Parameters

None

Return value

None

Throws

CommException

CommException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.NOT_SUPPORT_COMMAND	unsupported command

Notes

Common parameters are saved automatically when Power off SP1 device.

This can save common parameters into SP1 device immediately.

CommScanner#getParams

Description

Get common parameters from the SP1 device.

Syntax

```
CommScannerParams getParams ();
```

Parameters

None

Return value

Current CommScannerParams value of SP1 device.

Throws

CommException

CommException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.NOT_SUPPORT_COMMAND	unsupported command

Sample coding

Refer to the sample code 1.

CommScanner#setBtSettings

Description

Set Bluetooth parameters of SP1 device.

Syntax

```
void setBtSettings (  
    CommScannerBtSettings bt_settings  
);
```

Parameters

bt_settings

CommScannerBtSettings class variable

Return value

None

Throws

NullPointerException
If params is null.

CommException
CommException#getErrorCode() can get the error.
The errors of this API are as follows.

Error label	Description
ErrorCode.INVALID_COMMAND	Invalid parameters or syntax
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.NOT_SUPPORT_COMMAND	unsupported command

Sample coding

CommScanner#getBtSettings

Description

Get Bluetooth parameters of SP1 device.

Syntax

```
CommScannerBtSettings getBtSettings ();
```

Parameters

None

Return value

Current CommScannerBtSettings class variable of SP1 device.

Throws

CommException

CommException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.NOT_SUPPORT_COMMAND	unsupported command

CommScanner#init

Description

initialization

Syntax

```
void init (  
    CommInitType init_type  
);
```

Parameters

init_type

CommInitType.COMM_PARAMS
Initialize common parameters.

CommInitType.BT_PARAMS
Initialize Bluetooth parameters.

CommInitType.RFID_PARAMS
Initialize RFID parameters.

CommInitType.BARCODE_PARAMS
Initialize Barcode parameters.

CommInitType.ALL_PARAMS
Initialize all parameters.

Return value

None

Throws

CommException

CommException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode.INVALID_COMMAND	Invalid parameters or syntax
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.NOT_SUPPORT_COMMAND	unsupported command

Notes

Initialization may take a few seconds.

CommException#getErrorCode

Description

Return an error code of CommException.

If an CommException is thrown, you can check the error code to determine the cause.

Syntax

```
CommErrorCode error_code getErrorCode ();
```

Parameters

None

Return value

ErrorCode. COMMUNICATION_ERROR

Communication error

ErrorCode. COMMUNICATION_TIMEOUT

Communicataion timeout

ErrorCode. SESSION_NOT_ESTABLISHED

Session establishment error

ErrorCode. HANDSHAKE_INVALID_FORMAT

Handshake format error

ErrorCode. HANDSHAKE_NO_REQUIRED_ITEM

unsupported scanner

ErrorCode. CLOSE_TIMEOUT

close timeout

ErrorCode. TOO_MUCH_COMMAND_QUEUE

Command queue overflow

ErrorCode. INVALID_COMMAND

Invalid command error

ErrorCode. INVALID_STATUS

Invalid status error(For example, some api are called before claim)

ErrorCode. INVALID_RESPONSE

Invalid response is sent by SP1 device.

ErrorCode. NOT_SUPPORT_COMMAND

Unsupported command is sent

ErrorCode. UNKNOWN

Unknown error

CommScanner#getType

Description

Get Scanner type

Syntax

```
CommScannerType getType ();
```

Parameters

None

Return value

CommScannerType.TYPE_1D
1D Barcode Scanner

CommScannerType.TYPE_2D
2D Barcode Scanner

CommScannerType.TYPE_2D_LONG
Long range 2D Barcode Scanner

CommScannerType.TYPE_UNKOWN
unknown Scanner

CommScannerType.TYPE_RFID
RFID Scanner

CommScannerType.TYPE_2D_RFID
2D Barcode and RFID Scanner

Throws

This API does not throw an exception (no error occurs).

If you call this API before claim, the API returns Null.

Sample coding

CommScanner#getVersion

Description

Get Scanner firmware version

Syntax

```
String getVersion ();
```

Parameters

None

Return value

Scanner firmware version

Throws

This API does not throw an exception (no error occurs).

If you call this API before claim, the API returns Null.

Sample coding

Refer to the sample code 1.

CommScanner#addStatusListener

Description

Register the listener used to receive scanner status. For example, Bluetooth is unconnected.

Syntax

```
void addStatusListener (ScannerStatusListener listener);
```

Parameters

listener

Listener to receive scanner status

Return value

None

Throws

This API does not throw an exception (no error occurs).

Sample coding

CommScanner#removeStatusListener

Description

Remove the listener used to receive scanner status. For example, Bluetooth is unconnected.

Syntax

```
void removeStatusListener (ScannerStatusListener listener);
```

Parameters

listener

Listener to receive scanner status

Return value

None

Throws

This API does not throw an exception (no error occurs).

Sample coding

ScannerStatusListener#onScannerStatusChanged

Description

An event handler to notify when the Bluetooth connection state between the scanner and the host device is changed.

This also notifies when Bluetooth is disconnected by out of range.

This event handler overrides ScannerStatusListener and is notified to the instance registered as listener by addStatusListener.

Syntax

```
void onScannerStatusChanged (  
    CommScanner* scanner  
    CommStatusChangedEvent* state  
);
```

Parameters

scanner

Scanner instance whose state has changed

state

State notification instance

Sample coding

CommStatusChangedEvent#getStatus

Description

get scanner status

Syntax

```
ScannerStatus getStatus ();
```

Parameters

None

Return value

ScannerStatus.CLAIMED
Scanner is claimed.

ScannerStatus.CLOSE_WAIT
Scanner status is waiting for close.

When disconnection is detected, scanner status is CLOSE_WAIT.
If scanner is this status, please call CommScanner#close.

ScannerStatus.CLOSED
Scanner is closed

ScannerStatus.UNKNOWN
Scanner status is unknown

Sample coding

CommManager#addAcceptStatusListener

Description

Register the listener used to receive connection notification from the scanner.

Syntax

```
void addAcceptStatusListener (ScannerAcceptStatusListener listener);
```

Parameters

listener

Listener to receive connection notification from the scanner.

Return value

None

Throws

This API does not throw an exception (no error occurs).

Sample coding

Refer to the sample code 1.

CommManager#removeAcceptStatusListener

Description

Remove the listener which is registered with CommManager#addAcceptStatusListener.

Syntax

```
void removeAcceptStatusListener (ScannerAcceptStatusListener listener);
```

Parameters

listener

Listener to receive connection notification from the scanner.

Return value

None

Throws

This API does not throw an exception (no error occurs).

Sample coding

Refer to the sample code 1.

CommManager#startAccept

Description

Start waiting for connection from the scanner.

Syntax

```
void startAccept ();
```

Parameters

None

Return value

None

Throws

This API does not throw an exception (no error occurs).

Sample coding

Refer to the sample code 1.

CommManager#endAccept

Description

Stop waiting for connection from the scanner.

Syntax

```
void endAccept ();
```

Parameters

None

Return value

None

Throws

This API does not throw an exception (no error occurs).

Sample coding

Refer to the sample code 1.

ScannerAcceptStatusListener#OnScannerAppeared

Description

An event handler to notify when the Bluetooth connection establishes between the scanner and the client device.

This event handler overrides `ScannerAcceptStatusListener` and is notified to the instance registered as listener by `addAcceptStatusListener`.

Syntax

```
void OnScannerAppeared (CommScanner* scanner);
```

Parameters

scanner

Connected scanner's instance

Sample coding

Refer to the sample code 1.

CommScanner#claim

Description

Start session on SP1 device by SPP and handshake

Syntax

```
void claim ();
```

Parameters

None

Return value

None

Throws

CommException

CommException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode. SESSION_NOT_ESTABLISHED	Session establishment error
ErrorCode. INVALID_STATUS	Invalid status error
ErrorCode. HANDSHAKE_INVALID_FORMAT	handshake format error
ErrorCode. HANDSHAKE_NO_REQUIRED_ITEM	unsupported scanner

Sample coding

Refer to the sample code 1.

CommScanner#close

Description

Close the session

Syntax

```
void close ();
```

Parameters

None

Return value

None

Throws

CommException

CommException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode. INVALID_STATUS	Invalid status error
ErrorCode. CLOSE_TIMEOUT	close timeout

Sample coding

Refer to the sample code 1.

CommManager#getScanners

Description

Get paired list on Android device.

Syntax

```
static List<CommScanner> getScanners ();
```

Parameters

None

Return value

Paired commScanner instance list

If no paired list, null return.

Throws

This API does not throw an exception (no error occurs).

Sample coding

Refer to the sample code 1.

CommScanner#getBTAddress

Description

Get Bluetooth Address of the scanner

Syntax

```
String getBTAddress ();
```

Parameters

None

Return value

Bluetooth Address

Throws

This API does not throw an exception (no error occurs).

Sample coding

CommScanner#getBTLocalName

Description

Get Bluetooth Local Name of the scanner

Syntax

```
String getBTLocalName ();
```

Parameters

None

Return value

None

Throws

This API does not throw an exception (no error occurs).

Sample coding

Refer to the sample code 1.

CommScanner#getModel

Description

Get model name of scanner. This model name is SP1-QUBi.

Syntax

```
String getModel ();
```

Parameters

None

Return value

model name

Throws

This API does not throw an exception (no error occurs).

If you call this API before claim, the API returns Null.

CommScanner#getBarcodeScanner

Description

Get BarcodeScanner instance

Syntax

```
BarcodeScanner getBarcodeScanner ();
```

Parameters

None

Return value

BarcodeScanner instance which is created by CommScanner#claim

Throws

This API does not throw an exception (no error occurs).

If you call this API before claim, the API returns Null.

Sample coding

```
CommScanner[0].getBarcodeScanner.openReader();
```

And refer to the sample code 1.

CommScanner#getRFIDScanner

Description

Get RFIDScanner instance

Syntax

```
RFIDScanner getRFIDScanner ();
```

Parameters

None

Return value

RFIDScanner instance which is created by CommScanner#claim

Throws

CommException

Refer to the description of CommException#getErrorCode().

Sample coding

```
CommScanner[0].getRFIDScanner.openInventory();
```

And refer to the sample code 1.

CommScanner#addKeyStatusListener

Description

Register the listener used to receive key change notification from the scanner.

Syntax

```
void addKeyStatusListener (ScannerKeyStatusListener listener);
```

Parameters

listener

Listener to receive scanner key status

Return value

None

Throws

This API does not throw an exception (no error occurs).

CommScanner#removeKeyStatusListener

Description

Remove the listener which is registered with CommManager# addKeyStatusListener.

Syntax

```
void removeKeyStatusListener (ScannerKeyStatusListener listener);
```

Parameters

listener

Listener to receive scanner key status

Return value

None

Throws

This API does not throw an exception (no error occurs).

ScannerKeyStatusListener#onScannerKeyStatusChanged

Description

An event handler to notify when the key status is changed.

This notifies when key is pressed or released.

This event handler overrides ScannerKeyStatusListener and is notified to the instance registered as listener by addKeyStatusListener.

Syntax

```
void onScannerKeyStatusChanged (
    CommScanner* scanner
    CommKeyStatusChangedEvent* keystate
);
```

Parameters

scanner

Scanner instance whose key status has changed

keystate

Key status notification instance

Return value

None

CommKeyStatusChangedEvent#getKeyName

Description

Get name of key which status is changed.

Syntax

```
ScannerKeyName getKeyName ();
```

Parameters

None

Return value

ScannerKeyName.TRIGGER

Trigger key status is changed.

ScannerKeyName.UNKNOWN

Unkonwn key status is changed.

CommKeyStatusChangedEvent#getKeyStatus

Description

Get key state which is pressed or released.

Syntax

```
ScannerKeyStatus getKeyStatus ();
```

Parameters

None

Return value

ScannerKeyStatus.RELEASE
Key is released.

ScannerKeyStatus.PRESS
Key is pressed.

CommScanner#getCurrentKeyStatus

Description

Get the current status of the key.
This is synchronization function.

Syntax

```
ScannerKeyStatus getCurrentKeyStatus (ScannerKeyName name);
```

Parameters

name

Name of the key to be acquired.

SP1 can get only trigger key status.

Return value

ScannerKeyStatus.RELEASE
Key is released.

ScannerKeyStatus.PRESS
Key is pressed.

Throws

CommException

CommException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode.INVALID_COMMAND	Invalid parameters or syntax
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.NOT_SUPPORT_COMMAND	unsupported command

CommScanner#getRegion

Description

Get the country code of the scanner. If the EU model, "EU" can be gotten.

Syntax

```
String getRegion();
```

Parameters

None

Return value

Country code

Throws

This API does not throw an exception (no error occurs).

If you call this API before claim, the API returns Null.

Notes

Country code	Country
AU	Australia
CN	China
EU	Europe
HK	Hong Kong, Singapore, Thailand
ID	Indonesia
IL	Israel
IN	India
JP	Japan
KR	Korea
MX	Mexico
MY	Malaysia
PH	Philippines
RU	Russia
TW	Taiwan
US	USA, Canada
VN	Vietnam

Sample code 1

```
public class ExecuteRFIDActivity extends AppCompatActivity implements ScannerAcceptStatusListener {

    private RFIDScanner mRfidScanner = null;

    private TextView textGetVersion = null;
    int count = 0;
    String data;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_execute_connect);

        // Start accept
        CommManager.addAcceptStatusListener(this);
        CommManager.startAccept();

    }

    @Override
    public void OnScannerAppeared(CommScanner commScanner) {

        Log.d("ExecuteConnectActivity", " OnScannerAppeared ");
        Log.d("ExecuteConnectActivity", "BT Address=" + commScanner.getBTAddress());

        //Establish connection
        commScanner.claim();

        // Stop accept
        CommManager.endAccept();
        CommManager.removeAcceptStatusListener(this);

        // for disconnect notify
        commScanner.addStatusListener(this);

    }

}
```

CommScannerParams class

Description

Class to store the settings related with common parameters.

(✓:Support)

Property : description	Setting value		Initial value	SP1
CommScannerParams : Related to buzzer, power				
powerSave	true	Power save mode		✓
	false	normal mode	○	✓
ponBuzzer : sound the buzzer when power-on	true	Enabled	○	
	false	Disabled		
batteryConf : Confirm battery charge remaining by push power button.	true	Enabled	○	
	false	Disabled		
buzzerVolume	BuzzerVolume.LOW	Low volume		✓
	BuzzerVolume.MIDDLE	Middle volume		✓
	BuzzerVolume.LOUD	Loud volume	○	✓
buzzerTone :Except power-on and warning buzzer	BuzzerTone.LOW	Low		✓
	BuzzerTone.MEDIUM	Medium		✓
	BuzzerTone.HIGH	High	○	✓
buzzerDuration :Except power-on and warning buzzer	BuzzerDuration.SHORT	Short		✓
	BuzzerDuration.MIDDLE	Middle	○	✓
	BuzzerDuration.LONG	Long		✓
CommScannerParams.autopoweroff : Related to autopower settings				
settings	true	Enabled Auto Power off	○	✓
	false	Disabled Auto Power off		✓
duration	0x5– 0x280	Auto power off time(minute)	0x3C (60min)	✓
CommScannerParams.btbutton : Related to buletooth disconnection				
disconnectPermit : Disconnect by hold down pairing button	true	Enabled	○	✓
	false	Disabled		✓
reConnect :Bluetooth wait settings when disconnected	ReConnect.REPAIRING	Waiting for connection by the pairing button		✓
	ReConnect.WAITHOST	Waiting for connection from host	○	✓
	ReConnect.ANY	Waiting for connection every time		✓
CommScannerParams.notification.sound : Related to notification sound				
buzzer :Sound the buzzer when reading RFID or Barcode	Buzzer.ENABLE	Enabled	○	✓
	Buzzer.DISABLE	Disabled		✓
CommScannerParams.notification.led : Related to notification LED				
led :Light the LED when reading RFID or Barcode	true	Enabled	○	✓
	false	Disabled		✓

[Remarks]

reConnect setting value and status after disconnection event

Disconnection event reConnect setting	Disconnect by pairing button	Switch mode by Mode button. (In case Mode.AUTO)	Close by API (CommScanner#close)	Other (Moving out of service area, etc)
REPAIRING	Disconnecting	Waiting for connection	Disconnecting	Disconnecting
WAITHOST	Disconnecting	Waiting for connection	Disconnecting	Waiting for connection
ANY	Disconnecting	Waiting for connection	Waiting for connection	Waiting for connection

CommScannerBTSettings class

Description

Class to store the settings related with Bluetooth

(✓:Support)

Property : description	Setting value		Initial value	SP1
CommScannerBTSettings				
mode : master or slave or auto	Mode.MASTER	Master		✓
	Mode.SLAVE	Slave		✓
	Mode.AUTO	Auto mode	○	✓
CommScannerBTSettings.master : Related to master				
connectedTo : how connect	ConnectedToBtAddress	Connect with Bluetooth address	○	
	ConnectedToLocalName	Connect with Bluetooth Local name		
btAddress	Bluetooth address in hex 12 characters of 0-9,A-F	Bluetooth address of Master connection	"FFFFFFFFFFFFFF"	✓
localName	Local name in variable-length string from 1 to 16 characters, alphanumeric or symbol	Bluetooth Local name of Master connection	"FFFFFFFFFFFFFF"	
pincode	Specify PIN code in less than 8 disit of alphanumeric or symbol	PIN code of Master connection	"1234" (4 digit)	
tryingTime	TryingTime.30SEC	Master try time 30 seconds	○	✓
	TryingTime.NONE	No timeout		✓
CommScannerBTSettings.slave : Related to slave				
waitingTime	WaitingTime.2MIN	Slave wait time 2 minutes	○	✓
	WaitingTime.4MIN	Slave wait time 4 minutes		✓
	WaitingTime.10MIN	Slave wait time 10 minutes		✓
	WaitingTime.30MIN	Slave wait time 30 minutes		✓
	WaitingTime.NONE	No timeout		✓
stealth	true	stealth mode		
	false	normal mode	○	
pincode	PIN code in less than 8 disit of alphanumeric or symbol	PIN code of Slave connection	"1234" (4 digit)	

[Remarks]

Each BTSettings mode is as specified below.

AUTO

For the first time, SP1 waits for host connection as Bluetooth slave mode.

After connected, SP1 will memorize the Bluetooth address of the connected device.

Next time, SP1 will try to connect to it as Bluetooth master mode.

SP1 can be switched the master / slave manually by pressing the MODE button for 3 seconds or longer.

MASTER

Bluetooth master mode.

SP1 will try to connect to the device by btAddress always.

Master / slave switching by the MODE button is disabled.

SLAVE

Bluetooth slave mode.

SP1 waits for connection from host always.

Master / slave switching by the MODE button is disabled.

Chapter 5. RFID

5.1. Outline

These API are for reading or writing RF tags and setting RFID reading parameters.

5.1.1. RFID data

RFID data is stored to List of RFIDData class.

You can use the following method to get the information of the RFID data stored in RFIDData class.

-RFIDData#getData: get RFID data

* To get the length of RFID data, use the length method shown as below.
e.g.: `int rfidDataLength = data.length();`

5.1.2. Batch mode

SP1 continues to read RF tags as batch mode after unexpected Bluetooth disconnection occurs during reading.

RF tags data during batch mode are stored in the internal memory.

Connecting with the host device, SP1 stop the reading.

RF tags data which is stored in the internal memory can be gotten by the method below.

-RFIDScanner#getCount: Get the number of tags read during batch mode

-RFIDScanner#pullData: Get tag data during batch mode

5.2. RFID API

RFIDScanner#openInventory

Description

Enable the scanner to Inventory according to the settings set by the RFIDScanner#setSettings method.
For settings that can be set up, refer to RFIDScannerSettings class.
This cannot be opened at the same time as the Barcode.

Syntax

```
void openInventory ();
```

Parameters

None

Return value

None

Throws

RFIDException
RFIDException#getErrorCode() can get the error.
The errors of this API are as follows.

Error label	Description
ErrorCode.COMMUNICATION_TIMEOUT	Communication timeout
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.NOT_SUPPORT_COMMAND	unsupported command
ErrorCode.SCANNER_CHARGING	Error during charging

Sample coding

Refer to the sample code 2.

RFIDScanner#openInventory(Index)

Description

Enable the scanner to Inventory according to the settings set by the RFIDScanner#setSettings method. For settings that can be set up, refer to RFIDScannerSettings class.

After sending all stored data during Batch mode, Data read by Inventory will continue to be sent.

This cannot be opened at the same time as the Barcode.

Syntax

```
void openInventory (int index);
```

Parameters

index

Normally, use the value of index + 1 that can be acquired by getIndex before disconnection.

Return value

None

Throws

RFIDException

RFIDException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode.COMMUNICATION_TIMEOUT	Communication timeout
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.NOT_SUPPORT_COMMAND	unsupported command
ErrorCode.SCANNER_CHARGING	Error during charging
ErrorCode.INVALID_COMMAND	Invalid parameters or syntax

Sample coding

RFIDScanner#openRead(bank, addr, size, pwd)

Description

Enable the scanner to Read according to the settings set by the RFIDScanner#setSettings method.

For settings that can be set up, refer to RFIDScannerSettings class.

SP1 read any RF tags.

This cannot be opened at the same time as the Barcode.

Syntax

```
void openRead (RFIDBank bank, short addr, short size, byte[] pwd);
```

Parameters

bank

RFIDBank.RESERVED Reserved bank

RFIDBank.U11 U11 bank

RFIDBank.TID TID bank

RFIDBank.USER User bank

addr

Reading start position (It can be designated only multiples of 2).

RFID tag can be accessed by WORD.

size

Reading size (It can be designated only multiples of 2).

RFID tag can be accessed by WORD.

pwd

access password

Return value

None

Throws

RFIDException

RFIDException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode.COMMUNICATION_TIMEOUT	Communication timeout
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.NOT_SUPPORT_COMMAND	unsupported command
ErrorCode.SCANNER_CHARGING	Error during charging
ErrorCode.INVALID_COMMAND	Invalid parameters or syntax

Sample coding

RFIDScanner#openRead(bank, addr, size, pwd, UII)

Description

Enable the scanner to Read according to the settings set by the RFIDScanner#setSettings method.
For settings that can be set up, refer to RFIDScannerSettings class.
This cannot be opened at the same time as the Barcode.

Syntax

```
void openRead (RFIDBank bank, short addr, short size, byte[] pwd, byte[] UII);
```

Parameters

bank

RFIDBank.RESERVED Reserved bank
RFIDBank.UII UII bank
RFIDBank.TID TID bank
RFIDBank.USER User bank

addr

Reading start position (It can be designated only multiples of 2).
RFID tag can be accessed by WORD.

size

Reading size (It can be designated only multiples of 2).
RFID tag can be accessed by WORD.

pwd

access password

UII

UII designed Read.

Return value

None

Throws

RFIDException

RFIDException#getErrorCode() can get the error.
The errors of this API are as follows.

Error label	Description
ErrorCode.COMMUNICATION_TIMEOUT	Communication timeout
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.NOT_SUPPORT_COMMAND	unsupported command
ErrorCode.SCANNER_CHARGING	Error during charging
ErrorCode.INVALID_COMMAND	Invalid parameters or syntax

Sample coding

RFIDScanner#openRead(bank, addr, size, pwd, index)

Description

Enable the scanner to Read according to the settings set by the RFIDScanner#setSettings method.

For settings that can be set up, refer to RFIDScannerSettings class.

After sending all stored data during Batch mode, Data read by openRead will continue to be sent.

This cannot be opened at the same time as the Barcode.

Syntax

```
void openRead (RFIDBank bank, short addr, short size, byte[] pwd, int index);
```

Parameters

bank

RFIDBank.RESERVED	Reserved bank
RFIDBank.UID	UID bank
RFIDBank.TID	TID bank
RFIDBank.USER	User bank

addr

Reading start position (It can be designated only multiples of 2).

RFID tag can be accessed by WORD.

size

Reading size (It can be designated only multiples of 2).

RFID tag can be accessed by WORD.

pwd

access password

index

Normally, use the value of index + 1 that can be acquired by getIndex before disconnection.

Return value

None

Throws

RFIDException

RFIDException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode.COMMUNICATION_TIMEOUT	Communication timeout
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.NOT_SUPPORT_COMMAND	unsupported command
ErrorCode.SCANNER_CHARGING	Error during charging
ErrorCode.INVALID_COMMAND	Invalid parameters or syntax

RFIDScanner#close

Description

Close openInventory or openRead

Be sure to call this method after openInventory or openRead finish, so that other applications can use the RFID feature.

Syntax

```
void close ();
```

Parameters

None

Return value

None

Throws

RFIDException

RFIDException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode. COMMUNICATION_TIMEOUT	Communication timeout
ErrorCode. INVALID_STATUS	Invalid status error
ErrorCode. NOT_SUPPORT_COMMAND	unsupported command

Sample coding

And refer to the sample code 2.

RFIDScanner#writeOneTag

Description

Write one tag with specified UUI.

This is synchronization function. If an error occurs, an exception is thrown.

Trigger mode needs to be AUTO_OFF or CONTINUOUS2.

This cannot be used when the barcode is open.

Syntax

```
void writeOneTag (  
    RFIDBank bank, short addr, short size, byte[] pwd,  
    byte[] data, byte[] UUI, long timeout  
);
```

Parameters

bank

RFIDBank.RESERVED	Reserved bank
RFIDBank.UUI	UUI bank
RFIDBank.TID	TID bank
RFIDBank.USER	User bank

addr

Writing start position (Multiples of 2 can only be designated.)

RFID tag can be accessed by WORD.

This is the address from the start position of each bank.

size

Writing size

pwd

Access password

data

Writing data (Multiples of 2 can only be designated.)

RFID tag can be accessed by WORD.

UUI

Target tag's UUI

timeout

Timeout value (millisecond)

Return value

None

Throws

RFIDException

RFIDException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode.COMMUNICATION_TIMEOUT	Communication timeout
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.NOT_SUPPORT_COMMAND	Unsupported command
ErrorCode.SCANNER_CHARGING	Error while charging
ErrorCode.INVALID_COMMAND	Invalid parameters or syntax
ErrorCode.ACCESS_OUT_OF_RANGE_MEMORY	Access out of range memory
ErrorCode.SCANNER_ACCESS_LOCK_MEMORY	Access lock memory
ErrorCode.SCANNER_POWER_SHORTAGE_OF_WRITE_MEMORY	Writing power is not enough
ErrorCode.SCANNER_WRITE_LOCK_KILL_TIMEOUT	Write or Lock or Kill's timeout
ErrorCode.SCANNER_WRITE_LOCK_KILL_UNKNOWN	Write or Lock or Kill unknown error
ErrorCode.SCANNER_TRIGGER_MODE	Trigger mode error (Implement WRITE or LOCK or KILL in unsupported trigger mode)

Sample coding

```
//Change 11111111111111111111111111111111 to 22222222222222222222222222222222.
```

```
byte[] pass = {0,0,0,0}; // No authentication
```

```
byte[] uii = {0x11,0x11,0x11,0x11,0x11,0x11,0x11,0x11,0x11,0x11,0x11,0x11};
```

```
byte[] after_uui = {0x22,0x22,0x22,0x22,0x22,0x22,0x22,0x22,0x22,0x22,0x22,0x22};
```

```
short size = (short)after_uui.length;
```

```
short addr = 4; // UUI BANK 0 to 3 bytes are CRC and PC area. EPC area from 4th byte.
```

```
try {
```

```
    // Write UUI bank from uui to after_uui with 10 seconds timeout
```

```
    scanner.getRFIDScanner().writeOneTag(RFIDScannerSettings.RFIDBank.UUI,addr,size,pass,after_uui,uii,10000);
```

```
} catch (Exception e) {
```

```
    e.printStackTrace();
```

```
}
```

RFIDScanner#lockOneTag

Description

Lock one tag with Ull is specified

This is Synchronization function. If error occurs, an exception is thrown.

This cannot be used when the barcode is open.

Syntax

```
void lockOneTag (byte target, RFIDLock type, byte[] pwd, byte[] Ull, long timeout);
```

Parameters

target

Lock target

Specify by bit field (It can be set multiple things at a time)

01 : Kill password

02 : Access password

10 : Ull Bank

20 : TID Bank

40 : User Bank

type

RFIDLock.UNLOCK

Unlock

RFIDLock.LOCK

Lock

RFIDBank.PERMANENT_UNLOCK

permanent Unlock

RFIDBank.PERMANENT_LOCK

permanent Lock

pwd

Access password

Ull

Ull of Lock target

timeout

Timeout value(msec)

Return value

None

Throws

RFIDException

RFIDException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode.COMMUNICATION_TIMEOUT	Communication timeout
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.NOT_SUPPORT_COMMAND	Unsupported command
ErrorCode.SCANNER_CHARGING	Error while charging
ErrorCode.INVALID_COMMAND	Invalid parameters or syntax
ErrorCode.SCANNER_ACCESS_LOCK_MEMORY	Access lock memory
ErrorCode.SCANNER_POWER_SHORTAGE_OF_WRITE_MEMORY	Writing power is not enough
ErrorCode.SCANNER_WRITE_LOCK_KILL_TIMEOUT	Write or Lock or Kill's timeout
ErrorCode.SCANNER_WRITE_LOCK_KILL_UNKNOWN	Write or Lock or Kill unknown error
ErrorCode.SCANNER_TRIGGER_MODE	Trigger mode error (Implement WRITE or LOCK or KILL in unsupported trigger mode)

RFIDScanner#killOneTag

Description

Kill one tag with UII is specified

This is Synchronization function. If error occurs, an exception is thrown.

This cannot be used when the barcode is open.

Syntax

```
void killOneTag (byte[] pwd, byte[] UII, long timeout);
```

Parameters

pwd

Kill password

UII

UII of Kill target

timeout

Timeout value(msec)

Return value

None

Throws

RFIDException

RFIDException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode.COMMUNICATION_TIMEOUT	Communication timeout
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.NOT_SUPPORT_COMMAND	unsupported command
ErrorCode.SCANNER_CHARGING	Error during charging
ErrorCode.INVALID_COMMAND	Invalid parameters or syntax
ErrorCode.SCANNER_POWER_SHORTAGE_OF_WRITE_MEMORY	Writing power is not enough
ErrorCode.SCANNER_WRITE_LOCK_KILL_TIMEOUT	Write or Lock or Kill's timeout
ErrorCode.SCANNER_WRITE_LOCK_KILL_UNKNOWN	Write or Lock or Kill unknown error
ErrorCode.SCANNER_TRIGGER_MODE	Trigger mode error

RFIDScanner#setSettings

Description

This method specifies the settings of RFID.

Syntax

```
void setSettings (  
    RFIDScannerSettings settings  
);
```

Parameters

settings

RFIDScannerSettings class variable set with RFID settings

Return value

None

Throws

NullPointerException
If *settings* is null.

RFIDException
RFIDException#getErrorCode() can get the error.
The errors of this API are as follows.

Error label	Description
ErrorCode. INVALID_COMMAND	Invalid parameters or syntax
ErrorCode. INVALID_STATUS	Invalid status error
ErrorCode. NOT_SUPPORT_COMMAND	unsupported command

Sample coding

Refer to the sample code 1.

Notes

Do not use this method during RFID reading.
This method will take effect the next time the RFID is read.

RFIDScanner#getSettings

Description

Get the current settings related with RF tags reading

Syntax

```
RFIDScannerSettings getSettings ();
```

Parameters

None

Return value

RFIDScannerSettings the current settings of RF tags reading

Throws

RFIDException

RFIDException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.NOT_SUPPORT_COMMAND	unsupported command

RFIDScanner#setAutoLinkProfile

Description

Enables / disables AutoLink Profile (Autopilot).

Inventory is performed with automatically changing the Link Profile.

When enabled, it will switch automatically without using the LinkProfile setting value of the RFID ScannerSettings class.

When disabled, it operates according to the LinkProfile setting value of the RFID ScannerSettings class.

The initial value is disable.

Syntax

```
void setAutoLinkProfile(boolean auto);
```

Parameters

auto

Enable AutoLinkProfile (Autopilot) if true.

Disable AutoLinkProfile (Autopilot) if false.

Return value

None

Throws

RFIDException

RFIDException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.NOT_SUPPORT_COMMAND	unsupported command

Notes

This setting will be reflected in the next Inventory operation.

This setting is valid only for Inventory operation.

If this setting is enabled on Europe models, the output is limited to 24 dBm.

RFIDScanner#getAutoLinkProfile

Description

Get the setting value of whether the AutoLink Profile (Autopilot) operation is enabled or disabled.

Syntax

```
boolean getAutoLinkProfile();
```

Parameters

None

Return value

Enable AutoLinkProfile (Autopilot) if true.

Disable AutoLinkProfile (Autopilot) if false.

Throws

RFIDException

RFIDException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.NOT_SUPPORT_COMMAND	unsupported command

RFIDScanner#setFilter

Description

Set filter of Reading RF tags. Up to six filters can be set.

Syntax

```
void setFilter (  
    RFIDScannerFilter filter[ ], RFIDLogicalOpe l_ope  
);
```

Parameters

filter

RFIDScannerFilter class list set with filter settings

l_ope

Combined condition of filter

RFIDLogicalOpe.AND and

RFIDLogicalOpe.OR or

Return value

None

Throws

NullPointerException

If *settings* is null.

RFIDException

RFIDException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode. INVALID_COMMAND	Invalid parameters or syntax
ErrorCode. INVALID_STATUS	Invalid status error
ErrorCode. NOT_SUPPORT_COMMAND	unsupported command

Sample coding

Notes

Do not use this method during RFID reading.

This method will take effect the next time the RFID is read.

UII bank includes PC and CRC area. Therefore, when applying a filter to EPC, add 0x20 to bitOffset.

RFIDScanner#clearFilter

Description

Clear Filter setting.

Syntax

```
void clearFilter ();
```

Parameters

None

Return value

None

Throws

RFIDException

RFIDException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.NOT_SUPPORT_COMMAND	unsupported command

Sample coding

RFIDScanner#clearDoubleReadingBuffer

Description

Clear double reading buffer

Syntax

```
void clearDoubleReadingBuffer ();
```

Parameters

None

Return value

None

Throws

RFIDException

RFIDException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.NOT_SUPPORT_COMMAND	unsupported command

RFIDScanner#setDataDelegate

Description

Register the delegate used to receive read data when the RF tags reading is completed.
If you need to remove the registered delegate, please call setDataDelegate(null).

Syntax

```
void setDataDelegate (RFIDDataDelegate delegate);
```

Parameters

delegate

delegate to receive data read

Return value

None

Throws

This API does not throw an exception (no error occurs).

Sample coding

Refer to the sample code 2.

RFIDDataDelegate#onRFIDDataReceived

Description

Event handler that starts listening by RFIDScanner#openInventory or RFIDScanner#openRead or RFIDScanner#pullData and notifies it when data is recieved from the scanner.

This event handler overrides RFIDDataDelegate and is notified to the instance registered with setDataDelegate.

Syntax

```
void onRFIDDataReceived (
    CommScanner* scanner,
    RFIDDataReceivedEvent* rfidEvent
);
```

Parameters

scanner

Connected Scanner class instance

rfidEvent

Event class instance

Return value

None

Sample coding

Refer to the sample code 2

RFIDDataReceivedEvent#getRFIDData

Description

Return the RFID data read as ArrayList<RFIDData>

PC, RSSI, UUI, Antenna, Polarization, Readout data can be acquired.

Syntax

```
List<RFIDData> getRFIDData ();
```

Parameters

None

Return value

List<RFIDData>

Sample coding

Refer to the sample code 2.

RFIDData#getData

Description

Get read out RFID tag's data (Binary)

Syntax

```
byte[] getData ();
```

Parameters

None

Return value

Readout data specified by openRead

Throws

This API does not throw an exception (no error occurs).

Sample coding

Refer to the sample code 2.

RFIDData#getRSSI

Description

Get RSSI value when Tag is read

Syntax

```
int getRSSI ();
```

Parameters

None

Return value

RSSI value when reading tags

This returns RSSI value obtained by multiplying the actual data by 10.

For example: This API returns -605 for -60.5dBm.

Throws

This API does not throw an exception (no error occurs).

Sample coding

Refer to the sample code 2.

RFIDData#getPC

Description

Get PC value when Tag is read

Syntax

```
int getPC ();
```

Parameters

None

Return value

PC

Throws

This API does not throw an exception (no error occurs).

Sample coding

Refer to the sample code 2.

RFIDData#getAntenna

Description

Get antenna information when Tag is read.

*SP doesn't support this method. SP1 always returns same value.

Syntax

```
byte getAntenna ();
```

Parameters

None

Return value

Antenna information

Throws

This API does not throw an exception (no error occurs).

Sample coding

Refer to the sample code 2.

RFIDData#getPolarization

Description

Get polarization information when Tag is read.

Syntax

```
byte getPolarization ();
```

Parameters

None

Return value

Polarization information

Throws

This API does not throw an exception (no error occurs).

Sample coding

Refer to the sample code 2.

RFIDData#getUII

Description

Get read tag's UII

Syntax

```
byte[] getUII ();
```

Parameters

None

Return value

UII data

* In the case of a tag that implements XPC,
this API returns the value of XPC is added to the first 2 bytes.

For example:

If the following values can be obtained with this API when reading tags implementing XPC,
The 0080 of the first two bytes is XPC,
and the subsequent 123456789ABCDEF012345678 is the UII value.

0080123456789ABCDEF012345678

Throws

This API does not throw an exception (no error occurs).

Sample coding

Refer to the sample code 2.

RFIDData#getIndex

Description

Get read tag's Index value. From 0 to 79999. If index value is 79999, next index value is 0.

For each reading tag, SP1 increments this value and sends it with RFID data.

It is used to notify SP1 how much data was received before disconnection.

Syntax

```
int getIndex ();
```

Parameters

None

Return value

Index value

Throws

This API does not throw an exception (no error occurs).

Sample coding

RFIDData#getResult

Description

Get the result as to whether tag reading was done normally.

This function is used when openRead.

Syntax

```
RFIDResult getResult ();
```

Parameters

None

Return value

RFIDResult.OK
Normal end

RFIDResult.ERROR_MEM_RANGE
Access out of range memory

RFIDResult.ERROR_LOCK_MEM_ACCESS
Access lock memory

RFIDResult.ERROR_LACK_OF_POWER
Power is not enough

RFIDResult.ERROR_OTHER
Other error

Throws

This API does not throw an exception (no error occurs).

Sample coding

RFIDData#getCh

Description

Get channel index when Tag is read.

It is necessary to set in RFIDScanner#setResponse.

If not set, it will be undefined value.

Syntax

```
byte getCh ();
```

Parameters

None

Return value

Channel index when Tag is read.

For Euro

INDEX	CH
0	4
1	7
2	10
3	13

For North America

INDEX	CH
0	1
1	2
2	3
3	4
...	...
49	50

For Israel

INDEX	CH
0	1
1	2
2	3
3	4
4	5

Throws

This API does not throw an exception (no error occurs).

RFIDData#getPhase

Description

Get phase difference value when Tag is read.

It is necessary to set in RFIDScanner#setResponse.

If not set, it will be undefined value.

Syntax

```
byte getPhase ();
```

Parameters

None

Return value

Get phase difference value when Tag is read.

Throws

This API does not throw an exception (no error occurs).

RFIDScanner#getCount

Description

Get the number of tags read from the index during Batch mode.

Syntax

```
int getCount(int index);
```

Parameters

index

Normally, use the value of index + 1 that can be acquired by getIndex before disconnection.

Return value

The number of tags read from the index during Batch mode.

Throws

RFIDException

RFIDException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode.INVALID_STATUS	Invalid status error
ErrorCode.NOT_SUPPORT_COMMAND	unsupported command

RFIDScanner#pullData

Description

Get tag data during batch mode without reading.

Data is sent to delegate registered with RFIDScanner#setDataDelegate.

Data is sent asynchronously in the same way as RFIDScanner#openInventory.

Syntax

```
void pullData(int index);
```

Parameters

index

Normally, use the value of index + 1 that can be acquired by getIndex before disconnection.

Return value

None

Throws

RFIDException

RFIDException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode. COMMUNICATION_TIMEOUT	Communication timeout
ErrorCode. INVALID_COMMAND	Invalid parameters or syntax
ErrorCode. INVALID_STATUS	Invalid status error
ErrorCode. NOT_SUPPORT_COMMAND	unsupported command

RFIDScanner#setResponse

Description

Set the acquisition of information at RFID reading.

Syntax

```
void setResponse (  
    RFIDScannerResponse response  
);
```

Parameters

response

RFIDScannerResponse class variable

Return value

None

Throws

NullPointerException
If *settings* is null.

RFIDException
RFIDException#getErrorCode() can get the error.
The errors of this API are as follows.

Error label	Description
ErrorCode. INVALID_COMMAND	Invalid parameters or syntax
ErrorCode. INVALID_STATUS	Invalid status error
ErrorCode. NOT_SUPPORT_COMMAND	unsupported command

Notes

Do not use this method during RFID reading.
This method will take effect the next time the RFID is read.

RFIDScanner#getResponse

Description

Get the acquisition of information at RFID reading.

Syntax

```
RFIDScannerResponse getResponse ();
```

Parameters

None

Return value

RFIDScannerResponse the current acquisition of information of RF tags reading

Throws

RFIDException

RFIDException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode. INVALID_STATUS	Invalid status error
ErrorCode. NOT_SUPPORT_COMMAND	unsupported command

RFIDException#getErrorCode

Description

Return an error code of RFIDException.

If an RFIDException is thrown, you can check the error code to determine the cause.

Syntax

```
ErrorCode getErrorCode ();
```

Parameters

None

Return value

ErrorCode. COMMUNICATION_ERROR

Communication error

ErrorCode. COMMUNICATION_TIMEOUT

Communication timeout

ErrorCode. TOO_MUCH_COMMAND_QUEUE

Command queue overflow

ErrorCode. INVALID_COMMAND

Invalid command error

ErrorCode. INVALID_STATUS

Invalid status error(For example, close are called before openInventory)

ErrorCode. INVALID_RESPONSE

Invalid response is sent by SP1 device.

ErrorCode. NOT_SUPPORT_COMMAND

Unsupported command is sent

ErrorCode. UNKNOWN

Unknown error

ErrorCode.ACCESS_OUT_OF_RANGE_MEMORY

Access out of range memory

ErrorCode.SCANNER_ACCESS_LOCK_MEMORY

Access lock memory

ErrorCode.SCANNER_POWER_SHORTAGE_OF_WRITE_MEMORY

Writing power is not enough

ErrorCode.SCANNER_WRITE_LOCK_KILL_TIMEOUT

Write or Lock or Kill's timeout

ErrorCode.SCANNER_WRITE_LOCK_KILL_UNKNOWN

Write or Lock or Kill unknown error

ErrorCode.SCANNER_CHARGING

Error during charging(RF tags can not be read during charging)

ErrorCode.SCANNER_TRIGGER_MODE

Trigger mode error(Write, Lock, Kill unsupport some trigger mode)

ErrorCode. SCANNER_TEMP_ERROR

Temperature error. After a while, write E-Paper Tag again

ErrorCode.EXECUTE

ROM save failed

ErrorCode. SCANNER_EPD_LOST

The target E-Paper Tag is lost during the writing process

ErrorCode. SCANNER_EPD_CHARGE_TIMEOUT

Timeout of power supply to E-Paper Tag

ErrorCode. SCANNER_EPD_OPERATION_TIMEOUT

E-Paper Tag processing timeout

Sample code 2

```
public class ExecuteRFIDActivity extends AppCompatActivity implements RFIDDataDelegate {

    private CommScanner mScanner = null;
    private RFIDScanner mRfidScanner = null;

    private TextView textGetVersion = null;
    int count = 0;
    String data;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_execute_rfid);

        textGetVersion = (TextView) findViewById(R.id.get_version);

        // get paired list
        List<CommScanner> scanner_list = CommManager.getScanners();

        for (int i = 0; i < scanner_list.size(); ++i) {
            Log.d("ExecuteRFIDActivity", "BT Address=" + scanner_list.get(i).getBTAddress());
            Log.d("ExecuteRFIDActivity", "BT LC=" + scanner_list.get(i).getBTLocalName());

            // select Bluetooth local name
            if (scanner_list.get(i).getBTLocalName().equals("SP1-800001")) {
                mScanner = scanner_list.get(i);
            }
        }

        // connect SP1-800001
        mScanner.claim();

        textGetVersion.setText(mScanner.getVersion());
    }

    public void onBtnClick(View v) {
        switch (v.getId()) {
            case R.id.demo_start_button:
                if (mScanner != null) {

                    CommScannerParams commScannerParams = new CommScannerParams();
                    commScannerParams.buzzerVolume = CommScannerParams.BuzzerVolume.LOW;

                    if (mRfidScanner == null) {
                        mRfidScanner = mScanner.getRFIDScanner();
                    }
                    mRfidScanner.setDataDelegate(this);
                }

                if (mRfidScanner != null) {

                    // get current settings from SP1
                    RFIDScannerSettings rfidScannerSettings = mRfidScanner.getSettings();

                    rfidScannerSettings.scan.triggerMode = RFIDScannerSettings.Scan.TriggerMode.AUTO_OFF;

                    mRfidScanner.setSettings(rfidScannerSettings);

                    // start Inventory
                    mRfidScanner.openInventory();
                }
                break;
            }
    }
}
```

```

        case R.id.demo_stop_button:
            if (mRfidScanner != null) {

                mRfidScanner.setDataDelegate(null);

                mRfidScanner.close();
            }
            break;
    }
}

@Override
public void onRFIDDataReceived(CommScanner commScanner, RFIDDataReceivedEvent rfidDataReceivedEvent) {
    Log.d("ExecuteRFIDActivity", "OnRFIDDataReceived");
    count++;
    StringBuilder stringBuilder = new StringBuilder();
    stringBuilder.append("count" + count + "\n");

    List<RFIDData> rfidDataList = rfidDataReceivedEvent.getRFIDData();
    for (RFIDData rfidData : rfidDataList) {
        stringBuilder.append("data:" + byteToString(rfidData.getData()) + "\n");
        stringBuilder.append("PC:" + rfidData.getPC() + "\n");
        stringBuilder.append("Polarization:" + rfidData.getPolarization() + "\n");
        stringBuilder.append("RSSI:" + rfidData.getRSSI() + "\n");
        stringBuilder.append("Ull:" + byteToString(rfidData.getUll()) + "\n");
    }
    data = stringBuilder.toString();
    Log.d("ExecuteRFIDActivity", "data=" + data);
}

private String byteToString(byte[] bytes){
    StringBuilder stringBuilder = new StringBuilder();
    for(byte b : bytes){
        stringBuilder.append(String.format("%02X", b));
    }
    return stringBuilder.toString();
}

private byte[] stringToByte(String hex) {
    byte[] bytes = new byte[hex.length() / 2];
    for (int index = 0; index < bytes.length; index++) {
        bytes[index] = (byte) Integer.parseInt(hex.substring(index * 2, (index + 1) * 2), 16);
    }
    return bytes;
}
}

```

RFIDScannerFilter Class

Description

Filter setting class

Property : description	Setting value		Initial value	SP1
RFIDScannerFilter				
bank :target bank	Bank.UII	UII bank	–	✓
	Bank.TID	TID bank	–	✓
	Bank.USER	user bank	–	✓
bitOffset	"00000000" - "0007FFFF" (HEX)	Bit offset	–	✓
bitLength	"01" - "FF" (HEX)	Effective bit length from the least significant bit of the filter data	–	✓
filterData	Max 255bit byte[]	Filter data (Use the least significant bit to the number of effective bits)	–	✓

RFIDScannerSettings Class

Description

Class to store the settings related with RFID

(✓:Support)

Property : description	Setting value		Initial value	SP1
RFIDScannerSettings.scan :				
triggerMode	TriggerMode.AUTO_OFF	Auto off mode		✓
	TriggerMode.MOMENTARY	Momentary mode		✓
	TriggerMode.ALTERNATE	Alternate mode		✓
	TriggerMode.CONTINUOUS1	Continuous mode 1	○	✓
	TriggerMode.CONTINUOUS2	Continuous mode 2		✓
oneShot :One shot duration	0-5	0:Auto off mode 1-5:One shot 1-5sec	○	✓
powerLevelRead : Read power	4-30	Read power level(dBm)	30	✓
powerLevelWrite : Write power	4-30	Write power level(dBm)	30	✓
channel	0x00000000 – 0x0007FFFF	For Euro settings 0x00000001 : Ch4, 0x00000002 : Ch7, 0x00000004 : Ch10, 0x00000008 : Ch13	0x0000000F (Ch4,7, 10, 13 are enabled)	✓
		For Israel settings 0x00000001 : Ch1, 0x00000002 : Ch2, 0x00000004 : Ch3, 0x00000008 : Ch4, 0x00000010 : Ch5	0x0000001F (Ch1,2, 3, 4, 5 are enabled)	
		For India settings 0x00000001 : Ch1, 0x00000002 : Ch2, 0x00000004 : Ch3	0x00000007 (Ch1,2, 3 are enabled)	
		For Japan settings 0x00000001 : Ch5, 0x00000002 : Ch11, 0x00000004: Ch17, 0x00000008 : Ch23, 0x00000010 : Ch24, 0x00000020 : Ch25	0x0000003F (Ch5,11,17, 23,24,25 are enabled)	
		For the Others settings This settings is now ignored.	0x0007FFFF	
qParam : Q param	0-7	Adjust the probability of collision of response from RF tag	4	✓
sessionFlag	SessionFlag.S0	After the RF tag communication standby state, UII of the same RF tag can be readout again.	○	✓
	SessionFlag.S1	If the specified time has elapsed after obtaining the UII, same UII RF tag can be acquired again.		✓
	SessionFlag.S2	If the specified time has elapsed after the RF tag communication standby state, same UII RF tag can be acquired again.		✓
	SessionFlag.S3	Same as S2		✓
sessionInit	true	Enabled		✓
	false	Disabled	○	✓
polarization	Polarization.V	Vertical only		✓
	Polarization.H	Only horizontal		✓
	Polarization.Both	Both vertical and horizontal	○	✓

powerSave	true	Power saving mode1		✓
	false	Normal mode (power saving prohibited)	○	✓
doubleReading	DoubleReading.Free	Double reading prevention is disabled	○	✓
	DoubleReading.PREVENT1	Double reading prevention is enabled during reading operation		✓
	DoubleReading.PREVENT2	Double reading prevention is enabled during Power on		✓
writeVeri : Write Verify	false	No Verify	○	✓
	true	With Verify		✓
linkProfile : Link Profile num	1 or 2 or 4 or 5	Link profile number For except Europe	4	✓
		For Europe	2	
powerSaveExt :Extended Power save mode	PowerSaveExt.DISABLE	Normal mode (power saving prohibited)	○	✓
	PowerSaveExt.MODE1	Power saving mode1		✓
	PowerSaveExt.MODE2	Power saving mode2		✓

Notes

Each trigger mode is as specified below.

Auto off (normal)

The RF tag communication continues for a maximum of 5 seconds after pressing the trigger. When the RF tag operation is completed before 5 seconds, it enters the standby mode. Also by releasing the trigger, it will be in a standby mode.

Auto off (one shot)

After the trigger is pressed, the RF tag communication continues for only one shot time. If the trigger is released, the RF tag communication continue until the one shot time elapses. When the RF tag operation is completed before the one shot time elapses, it enters the standby mode.

Momentary

While the trigger is pressed, the RF tag communication is enabled. Write, Lock, and Kill cannot be used in this mode.

Alternate

Every time a trigger is pressed, the RF tag communication mode and the standby mode are switched alternately. Write, Lock, and Kill cannot be used in this mode.

Continuous mode 1

From the call of the tag function control API (openInventory, etc.) to the close API, the RF tag communication continues. Write, Lock, and Kill cannot be used in this mode.

Continuous mode 2

From the call of the tag function control API (openInventory, etc.) to the close API, the RF tag communication continues. When the RF tag operation is completed, it enters the standby mode. To resume the communication, call the tag function control API, after calling the close API.

The double-reading prevention mode is as specified below.

Settings	Description
Disables double-reading prevention	Reading the same RF tag many times.
Enables double-reading prevention while reading	Continue to prevent re-reading the same RF tag during the RF tag reading. The double-reading prevention buffer is cleared when the following conditions are met. - Stop reading by trigger mode - Stop reading by RF tag control stop command - Transition to barcode mode - Start charging - Power OFF - RFIDScanner # ClearDoubleReadingBuffer
Enables double-reading prevention during Power on	Continue to prevent re-reading the same RF tag during Power on. The double-reading prevention buffer is cleared when the following conditions are met. - Power OFF - RFIDScanner # ClearDoubleReadingBuffer - Switching between RFIDScanner#OpenInventory and RFIDScanner#OpenRead - Switching between RFIDScanner#OpenRead size <= 128 and 129 <= size

Power save setting.

Ensure to use the extended power save mode. This power saving setting is left for compatibility. It may become out of support in the future.

Extended Power save setting.

The power saving effect is as follows.

Normal mode (power saving prohibited) < Power saving mode 1 < Power saving mode 2

The higher the power saving effect, the lower the reading speed.

On the relationship between power save and extended power save setting

	PowerSaveExt.DISABLE	PowerSaveExt.MODE1	PowerSaveExt.MODE2
PowerSave.false	<i>Normal mode</i>	<i>Power saving mode 1</i>	<i>Power saving mode 2</i>
PowerSave.true	<i>Power saving mode 1</i>	<i>Power saving mode 1</i>	<i>Power saving mode 2</i>

Read / Write power level settings.

Specification value is 10dBm-30dBm.

If you use between 4dBm-9dBm, check the operation (small distance).

RFIDScannerResponse Class

Description

Class to store the acquired information at RFID reading.

(✓:Support)

Property : description	Setting value		Initial value	SP1
RFIDScannerResponse:				
format : Format	Format.BINARY	Binary mode	☐	✓
	Format.TEXT	Text mode	☐	
pc :PC info	true	Enabled	☐	✓
	false	Disabled	☐	
rssi :RSSI info	true	Enabled	☐	✓
	false	Disabled	☐	
antenna :antenna info	true	Enabled	☐	✓
	false	Disabled	☐	
polarization :polarization info	true	Enabled	☐	✓
	false	Disabled	☐	
ch :ch info	true	Enabled	☐	✓
	false	Disabled	☐	✓
phase :phase info	true	Enabled	☐	✓
	false	Disabled	☐	✓

Notes

In order to acquire Ch and Phase information at RFID reading, enable ch and phase and call RFIDScanner#setResponse.

If using a setting other than the default, RFIDScanner#setResponse is needed every time connection is established.

Chapter 6. Barcode

6.1. Outline

These API are for reading Barcode and setting Barcode reading parameters.

6.1.1. Enable Reading

The BarcodeScanner#openReader() method enables the barcode device to read barcodes.

The following barcode types available in the SP1 can be specified in the BarcodeScanner#setSettings() method and the SP1 can handle one of them or their combination. For default settings and options of the following barcode types, refer to BarcodeScannerSettings class.

Code type	SP1 Support
1D code	
EAN-13 UPC-A	✓
EAN-8	✓
UPC-E	✓
ITF	✓
STF	✓
Codabar	✓
Code39	✓
Code93	✓
Code128, GS1-128 (EAN-128)	✓
MSI	
GS1 Databar	✓
GS1 Databar Limited	✓
GS1 Databar Expanded	✓
2D code	
QR Model1, 2	✓
Micro QR	✓
iQR	
Data Matrix	✓
PDF417	✓
Micro PDF417	✓
Maxi Code	✓
GS1 Composite	✓
Multi-line code	✓

6.1.2. Barcode Data

Barcode data read will be stored in the List of BarcodeData class.

The maximum value of barcode data is 8192 bytes.

You can use the following methods to get the information of the barcode data stored in

BarcodeData class

- BarcodeData#getSymbologyAim: get codes (AIM-compliant).
- BarcodeData#getSymbologyDenso: get codes (DENSO-specified)
- BarcodeData#getData: get barcode data
- * To get the length of barcode data, use the length

6.1.3. SQRC

SQRC is a QR Code that has data reading restrictions.

Reference the SQRC Setting user's manual for detail.

6.2. Barcode API

BarcodeScanner#openReader

Description

Enable the barcode scanner to read barcode according to the settings set by the BarcodeScanner#setSettings() method.

For settings that can be set up, refer to BarcodeScannerSettings class.

This cannot be opened at the same time as RFID.

Syntax

```
void openReader ();
```

Parameters

None

Return value

None

Throws

BarcodeException

BarcodeException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode. COMMUNICATION_TIMEOUT	Communication timeout
ErrorCode. INVALID_STATUS	Invalid status error
ErrorCode. NOT_SUPPORT_COMMAND	unsupported command

Sample coding

Refer to the sample code 3.

BarcodeScanner#closeReader

Description

Close barcode device and disable barcode reading.

Syntax

```
void closeReader ();
```

Parameters

None

Return value

None

Throws

BarcodeException

BarcodeException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode. COMMUNICATION_TIMEOUT	Communication timeout
ErrorCode. INVALID_STATUS	Invalid status error
ErrorCode. NOT_SUPPORT_COMMAND	unsupported command

Sample coding

Refer to the sample code 3.

BarcodeScanner#setSettings

Description

This method specifies the settings of barcode reading.

Syntax

```
void setSettings (  
    BarcodeScannerSettings settings  
);
```

Parameters

settings

BarcodeScannerSettings class variable set with barcode reading settings

Return value

None

Throws

NullPointerException
If settings is null.

BarcodeException
BarcodeException#getErrorCode() can get the error.
The errors of this API are as follows.

Error label	Description
ErrorCode. COMMUNICATION_TIMEOUT	Communication timeout
ErrorCode. INVALID_STATUS	Invalid status error
ErrorCode. NOT_SUPPORT_COMMAND	unsupported command
ErrorCode. INVALID_COMMAND	Invalid parameters or syntax

Sample coding

Refer to the sample code 3.

Notes

Setting values take effect on the barcode reading behavior after openReader() method is called next.

BarcodeScanner#getSettings

Description

Get the current settings related with barcode reading

Syntax

```
BarcodeScannerSettings getSettings ();
```

Parameters

None

Return value

BarcodeScannerSettings the current settings of barcode reading

Throws

BarcodeException

BarcodeException#getErrorCode() can get the error.

The errors of this API are as follows.

Error label	Description
ErrorCode. COMMUNICATION_TIMEOUT	Communication timeout
ErrorCode. INVALID_STATUS	Invalid status error
ErrorCode. NOT_SUPPORT_COMMAND	unsupported command

Sample coding

Refer to the sample code 3.

BarcodeScanner#setDataDelegate

Description

Register the delegate used to receive read data when the Barcode reading is completed.

If you need to remove the registered delegate, please call setDataDelegate(null).

Event handler that starts listening by BarcodeScanner#openReader and notifies it when data is received from the scanner.

This event handler overrides BarcodeDataDelegate and is notified to the instance registered with setDataDelegate.

Syntax

```
void setDataDelegate (BarcodeDataDelegate delegate);
```

Parameters

delegate

delegate to receive data read

Return value

None

Throws

This API does not throw an exception (no error occurs).

Sample coding

Refer to the sample code 3.

BarcodeDataDelegate#onBarcodeDataReceived

Description

Event handler that starts listening by BarcodeScanner#openReader and notifies it when data is recieved from the scanner.

This event handler overrides BarcodeDataDelegate and is notified to the instance registered with setDataDelegate.

Syntax

```
void onBarcodeDataReceived (  
    CommScanner* scanner,  
    BarcodeDataReceivedEvent* barcodeEvent  
);
```

Parameters

scanner

Connected Scanner class instance

barcodeEvent

Event class instance

Return value

None

Sample coding

Refer to the sample code 3

BarcodeDataReceivedEvent#getBarcodeData

Description

Return the Barcode data readout as ArrayList<BarcodeData>
Symbology Denso and Aim and Readout data can be acquired.

*Information about barcodes such as multiple line codes can also be returned.

*For details of barcode, refer to the specification of methods of BarcodeData class.

Syntax

```
List<BarcodeData> getBarcodeData ();
```

Parameters

None

Return value

List<BarcodeData>

*The List of BarcodeData class in which barcode data are stored.

Sample coding

Refer to the sample code 3.

BarcodeData#getData

Description

Return the barcode data read(Binary)

Syntax

```
byte[]  getData ();
```

Parameters

None

Return value

Byte array storing barcode data read.

Throws

This API does not throw an exception (no error occurs).

Sample coding

Refer to the sample code 3.

BarcodeData#getSymbologyDenso

Description

Return the code type of read data. The format of the code type is DENSO-SPECIFIED.

Syntax

```
String getSymbologyDenso ();
```

Parameters

None

Return value

String code type (DENSO-SPECIFIED).

Throws

This API does not throw an exception (no error occurs).

Sample coding

Refer to the sample code 3.

BarcodeData#getSymbologyAim

Description

Return the code type read. The format of the code type read is AIM USA “Guidelines on Symbology identifiers” compliant.

Syntax

```
String getSymbologyAim ();
```

Parameters

None

Return value

Code type (AIM USA “Guidelines on Symbology Identifiers” compliant).

Throws

This API does not throw an exception (no error occurs).

Sample coding

Refer to the sample code 3.

Notes

Code types returned by getSymbologyDenso() and getSymbologyAim() are listed as below.

Code Type			DENSO SPECIFIED	AIM USA (*1)
QR			Q	]Qm
	Unedited split		S(*2)	S(*2)
Micro QR			Q(*2)	Q(*2)
SQRC			Q(*2)	Q(*2)
iQR			G	]Qm
	Unedited split		S(*2)	S(*2)
PDF417			Y	]L0
MaxiCode			X	]Um
Data Matrix			Z	]dm
EAN-13(JAN-13) COMPOSITE	Linear code	no add on	A	]E0
		2-digit add on		]E3
		5-digit add on		
	2D code		Y	]e0
UPC-A COMPOSITE	Linear code	no add on	A	]E0
		2-digit add on		]X3
		5-digit add on		
	2D code		Y	]e0
EAN-8(JAN-8) COMPOSITE	Linear code	no add on	B	]E4
		2-digit add on		]E5
		5-digit add on		]E6
	2D code		Y	]e0
UPC-E COMPOSITE	Linear code	no add on	C	]X0
		2-digit add on		]X3
		5-digit add on		
	2D code		Y	]e0
GS1-128 (EAN-128) COMPOSITE	Linear code		W	]e0
	2D code		Y	none
GS1 DataBar COMPOSITE	Linear code		R	]e0
	2D code		Y	none
EAN-13 (JAN-13)	no add on		A	]E0
	2-digit add on			]E3
	5-digit add on			
UPC-A	no add on		A	]X0
	2-digit add on			]X3
	5-digit add on			
EAN-8 (JAN-8)	no add on		B	]E4
	2-digit add on			]E5
	5-digit add on			]E6
UPC-E	no add on		C	]X0
	2-digit add on			]X3
	5-digit add on			
ITF			I	]Im
STF	short		H	]R0
	normal			]S0
CODABAR (NW-7)			N	]Fm
CODE-39			M	]Am
CODE-93			L	]G0
CODE-128			K	]Cm
GS1-128 (EAN-128)			W	]C1
MSI			P	]M1
GS1 DataBar			R	]e0

(*1) The suffix “m” depends on the data format of the barcode system as shown in the table below.
e.g.)]I1]:Flag Character (ASCII 93) I:Code Character (ITF) 1:Modifier Character (shown below)
For example, if the code type is set as ITF with C/D, the code mark will be “] I1”.

Code Type	Modifier Character	Description
ITF	0	Read with no check digit
	1	Read with check digit
CODE-39	0	Read with no check digit
	1	Read with check digit
CODABAR (NW-7)	0	Read with no check digit
	1	Read with check digit
CODE-128	0	FNC1 is not included in the first and second characters from the start code
	2	The second character from the start code is FNC1
QR	0	Model 1
	1	Model 2
	2	Model2 ECI is included(*3)
	3	Model 2 The first character from the start code is FNC1
	4	Model2 ECI is include The first character from the start code is FNC1(*3)
	5	Model2 The first character from the start code is FNC2(*3)
	6	Model2 ECI is include The first character from the start code is FNC2(*3)
iQR	A	FNC1 is not included in the first character from the start code
	C	The first character from the start code is FNC1
MaxiCode	0	mode4、 mode5
	1	mode2、 mode3
Data Matrix	1	ECC-200
	2	ECC-200 (The first or the fifth character from the start code is FNC1)
	3	ECC-200 (The second or the sixth character from the start code is FNC1)
	4	ECC-200 ECI is included(*3)
	5	ECC-200 EC is included (The first or the fifth character from the start code is FNC1) (*3)
	6	ECC-200 ECI is included (The second or the sixth character from the start code is FNC1) (*3)

(*2) The code type which is not compliant with AIM USA [Guidelines on Symbology Identifiers] will be treated as DENSO Specified code type.

(*3) To be added.

BarcodeData#getPrivateDataLength

Description

Return the data length of the private part of SQRC code.

Syntax

```
int getPrivateDataLength ();
```

Parameters

This API does not throw an exception (no error occurs).

Return value

The data length of the private part of SQRC code.

BarcodeException#getErrorCode

Description

Return an error code of BarcodeException.

If an BarcodeException is thrown, you can check the error code to determine the cause.

Syntax

```
ErrorCode getErrorCode ();
```

Parameters

None

Return value

ErrorCode. COMMUNICATION_ERROR

Communication error

ErrorCode. COMMUNICATION_TIMEOUT

Communication timeout

ErrorCode. TOO_MUCH_COMMAND_QUEUE

Command queue overflow

ErrorCode. INVALID_COMMAND

Invalid command error

ErrorCode. INVALID_STATUS

Invalid status error(For example, close are called before openInventory)

ErrorCode. INVALID_OPERATION

Invalid operation error

ErrorCode. INVALID_RESPONSE

Invalid response is sent by SP1 device.

ErrorCode. INVALID_RESPONSE_CODE

Invalid response code is sent by SP1 device.

ErrorCode. NOT_SUPPORT_COMMAND

Unsupported command is sent

ErrorCode. UNKNOWN

Unknown error

Sample code 3

```
public class ExecuteBarcodeActivity extends AppCompatActivity implements BarcodeDataDelegate{

    private CommScanner mScanner = null;
    private BarcodeScanner mBarcodeScanner;

    int count = 0;
    String data;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_execute_barcode);

        // get paired list
        List<CommScanner> scanner_list = CommManager.getScanners();

        for (int i = 0; i < scanner_list.size(); ++i) {
            // select Bluetooth local name
            if (scanner_list.get(i).getBTLocalName().equals("SP1-800001")) {
                mScanner = scanner_list.get(i);
            }
        }

        // connect SP1-800001
        mScanner.claim();
    }

    public void onBtnClick(View v) {
        switch (v.getId()) {
            case R.id.demo_start_button:
                if(mScanner != null){

                    mBarcodeScanner = mScanner.getBarcodeScanner();

                    mBarcodeScanner.setDataDelegate(this);

                    // get scanner settings
                    BarcodeScannerSettings settings = mBarcodeScanner.getSettings();

                    // set scanner settings
                    mBarcodeScanner.setSettings(settings);
                }
                if(mBarcodeScanner != null){

                    // open
                    mBarcodeScanner.openReader();
                }
                break;
            case R.id.demo_stop_button:
                if(mBarcodeScanner != null){

                    mBarcodeScanner.setDataDelegate(null);

                    // close
                    mBarcodeScanner.closeReader();
                }
                break;
        }
    }
}
```

```

public void onBarcodeDataReceived(CommScanner commScanner, BarcodeDataReceivedEvent
barcodeDataReceivedEvent) {
    Log.d("ExecuteBarcodeActivity", "OnBarcodeDataReceived");
    count++;
    StringBuilder stringBuilder = new StringBuilder();
    stringBuilder.append("count" + count + "\n");

    List<BarcodeData> barcodeDataList = barcodeDataReceivedEvent.getBarcodeData();
    for(BarcodeData barcodeData : barcodeDataList) {
        stringBuilder.append("Data:" + new String(barcodeData.getData()) + "\n");

        stringBuilder.append("SymbologyAim:" + barcodeData.getSymbologyAim() + "\n");

        stringBuilder.append("SymbologyDenso:" + barcodeData.getSymbologyDenso() + "\n");
    }
    data = stringBuilder.toString();
    Log.d("ExecuteBarcodeActivity", "data=" + data);
}
}

```

BarcodeScannerSettings Class

Description

Class to store the settings related with barcode reading

(✓:Support)

Property : description	Settig value		Initial Value	SP1
BarcodeScannerSettings.scan : Related to reading operation				
triggerMode	TriggerMode.AUTO_OFF	Auto OFF mode	○	✓
	TriggerMode.MOMENTARY	Momentary mode		✓
	TriggerMode.ALTERNATE	Alternate mode		✓
	TriggerMode.CONTINUOUS	Continuous reading mode		✓
	TriggerMode.TRIGGER_RELEASE	Trigger release mode		✓
lightMode	LightMode.AUTO	Auto	○	✓
	LightMode.ALWAYS_ON	Always ON		✓
	LightMode.OFF	OFF		✓
markerMode	MarkerMode.NORMAL	Normal	○	✓
	MarkerMode.AHEAD	Marker ahead		✓
	MarkerMode.OFF	Marker OFF		✓
sideLightMode	SideLightMode.ON	ON		
	SideLightMode.OFF	OFF	○	
BarcodeScannerSettings.decode : Related to decoding				
sameBarcode IntervalTime : Re-read prevention time	0~255	Time for cancelling the prevention (x100ms)	10	✓
decodeLevel : Decoding level	1~9	Level 1 to 9	4	✓
invertMode : Reversed reflectance reading	InvertMode.DISABLED	Disabled	○	✓
	InvertMode.INVERSION_ONLY	Enabled (Reverse only)		
	InvertMode.AUTO	Enabled (Automatic detection)		✓
pointScanMode : Point scanning mode	PointScanMode.DISABLED	Normal	○	✓
	PointScanMode.ENABLED	Point scanning		✓
reverseMode : Mirror image reading	ReverseMode.DISABLED	Disabled	○	✓
	ReverseMode.ENABLED	Enabled		✓
charset : Barcode encoding method	Character string * See Android Developer website.	charset specified character string	"UTF-8"	
mirrorReflection	MirrorReflection.DISABLED	Disabled		✓
	MirrorReflection.ENABLED	Enabled	○	✓

Notes

lightMode

You can set whether to turn on the light when reading barcodes.
Turning off for materials that are highly reflective may improve reading.
It is turned off at least once every few times if LightMode.AUTO.

markerMode

You can set ON / OFF of the red marker, which is a guide when reading barcodes.

decodeLevel

Set the read decode level.

A smaller value increases the success rate of reading the code, but increases the risk of misreading poor quality code. (crack, dirt, etc.)

The higher the value, the lower the success rate of reading the code, but the lower the risk of misreading.

pointScanMode

You can scan by aiming at the code at the center of the marker.

If the marker has no code or the marker cannot be detected due to extraneous light, etc, it cannot be read. This reading is valid only when the markerMode is NORMAL or AHEAD.

Property : description	Setting value	Initial value	SP1
BarcodeScannerSettings.decode.multiLineMode : Related to multi-line code reading			
enabled : Multi-line reading	true	Enabled	✓
	false	Disabled	○
BarcodeScannerSettings.decode.multiLineMode.symbology1st/2nd/3rd : Information on code reading in the 1st/2nd/3rd lines			
symbologyType : Code type	MultiLineSymbologyType.NONE	Not designated (Disabled to read)	○
	MultiLineSymbologyType.EAN13UPCA	EAN13/UPC-A	✓
	MultiLineSymbologyType.EAN8	EAN-8	✓
	MultiLineSymbologyType.UPCE	UPC-E	✓
	MultiLineSymbologyType.INTERLEAVED2OF5	Interleaved 2of5	✓
	MultiLineSymbologyType.STANDARD2OF5	(ITF)	✓
	MultiLineSymbologyType.CODABAR	Standard 2of5	✓
	MultiLineSymbologyType.CODE39	(STF)	✓
	MultiLineSymbologyType.CODE93	CODABAR (NW-7)	✓
	MultiLineSymbologyType.CODE128	CODE-39	✓
firstCharacter : Character in the first digit (Valid only when the code type is POS)	""	Not limited	○
	"?"	Not limited	✓
	"0" to "9"	Character in the first digit on the readable label	✓
secondCharacter : Character in the second digit (Valid only when the code type is POS)	"" (Blank character string)	Not limited	○
	"?"	Not limited	✓
	"0" to "9"	Character in the second digit on the readable label	✓
startStop Character : Start and stop character (Valid only when the code type is Codabar)	"" (Blank character string)	Start or stop is not limited.	○
	Character string having start character and stop character linked ex) When "AA", "AB" "?" are designated, either start or stop can be designated. ex) "A?", "?D"	Only labels with designated start/stop character are readable.	✓
lengthMin : Minimum number of digits (Valid only when the code type is other than POS)	0	No designation of minimum number of digits	○
	2 to 99 (For ITF, STF) 3 to 99 (For Codabar) 1 to 99 (Case other than the above)	Minimum number of digits	✓
	2 to 99 (For ITF, STF) 3 to 99 (For Codabar) 1 to 99 (Case other than the above)	Maximum number of digits	99
verifyCheckDegit : Check digit verification (Valid only when code type is other than POS, Code39, and Code93)	true	Enabled	✓
	false	Disabled	○

Property : description	Setting value		Initial value	SP1
BarcodeScannerSettings.decode.symbologies: Related to code enabling to read				
BarcodeScannerSettings.decode.symbologies.ean13upcA: EAN-13, UPC-A related				
enabled : Enabling to read	true	Enabling to read	○	✓
	false	Disabling to read		✓
firstCharacter : Character in the first digit	""	Not limited	○	✓
	"?"	Not limited		✓
	"0" to "9"	First digit character in the readable code		✓
secondCharacter : Character in the second digit	""	Not limited	○	✓
	"?"	Not limited		✓
	"0" to "9"	Second digit character in the readable code		✓
BarcodeScannerSettings.decode.symbologies.ean13upcA.addOn : EAN-13, UPC-A add-on related				
enabled : With add-on	true	Enabling to read		✓
	false	Disabling to read	○	✓
addOn2Digit : With 2-digit add-on	true	Enabling to read		↘
	false	Disabling to read	○	
addOn5Digit : With 5-digit add-on	true	Enabling to read		↘
	false	Disabling to read	○	
onlyWithAddOn : Only with add-on	true	Limited to code with add-on		↘
	false	Not limited to code with add-on	○	
BarcodeScannerSettings.decode.symbologies.ean8 : EAN-8 related				
enabled : Enabling to read	true	Enabling to read	○	✓
	false	Disabling to read		✓
firstCharacter : Character in the first digit	""	Not limited	○	✓
	"?"	Not limited		✓
	"0" to "9"	First digit character in the readable code		✓
secondCharacter : Character in the second digit	""	Not limited	○	✓
	"?"	Not limited		✓
	"0" to "9"	Second digit character in the readable code		✓
BarcodeScannerSettings.decode.symbologies.ean8.addOn : EAN-8 add-on related				
enabled : With add-on	true	Enabling to read		✓
	false	Disabling to read	○	✓
addOn2Digit : With 2-digit add-on	true	Enabling to read		↘
	false	Disabling to read	○	
addOn5Digit : With 5-digit add-on	true	Enabling to read		↘
	false	Disabling to read	○	
onlyWithAddOn : Only with add-on	true	Limited to code with add-on		↘
	false	Not limited to code with add-on	○	
BarcodeScannerSettings.decode.symbologies.upcE: UPC-E related				
enabled : Enabling to read	true	Enabling to read	○	✓
	false	Disabling to read		✓
firstCharacter : Character in the first digit	""	Not limited	○	✓
	"?"	Not limited		✓
	"0" to "9"	First digit character in the readable code		✓
secondCharacter : Character in the second digit	""	Not limited	○	✓
	"?"	Not limited		✓
	"0" to "9"	Second digit character in the readable code		✓
BarcodeScannerSettings.decode.symbologies.upcE.addOn : UPC-E add-on related				
enabled : With add-on	true	Enabling to read		✓
	false	Disabling to read	○	✓
addOn2Digit : With 2-digit add-on	true	Enabling to read		↘
	false	Disabling to read	○	
addOn5Digit : With 5-digit add-on	true	Enabling to read		↘
	false	Disabling to read	○	
onlyWithAddOn : Only with add-on	true	Limited to code with add-on		↘
	false	Not limited to code with add-on	○	

Property : description	Setting value		Initial value	SP1
BarcodeScannerSettings.decode.symbologies.itf: ITF related				
enabled : Enabling to read	true	Enabling to read	○	✓
	false	Disabling to read		✓
lengthMin : Minimum number of digits	2 to 99	Minimum number of digits	4	✓
lengthMax : Maximun number of digits	2 to 99	Maximum number of digits	99	✓
verifyCheckDigit: Check digit verification	true	Valid		✓
	false	Invalid	○	✓
BarcodeScannerSettings.decode.symbologies.stf: STF related				
enabled : Enabling to read	true	Enabling to read	○	✓
	false	Disabling to read		✓
lengthMin : Minimum number of digits	2 to 99	Minimum number of digits	3	✓
lengthMax : Maximun number of digits	2 to 99	Maximum number of digits	99	✓
verifyCheckDigit: Check digit verification	true	Valid		✓
	false	Invalid	○	✓
startStopCharacter : Start and stop character type	""	Not limited	○	
	"S"	Short		
	"N"	Normal		
BarcodeScannerSettings.decode.symbologies.codabar: Codabar related				
enabled : Enabling to read	true	Enabling to read	○	✓
	false	Disabling to read		✓
lengthMin : Minimum number of digits	3 to 99	Minimum number of digits	4	✓
lengthMax : Maximun number of digits	3 to 99	Maximum number of digits	99	✓
verifyCheckDigit: Check digit verification	true	Valid		✓
	false	Invalid	○	✓
startStopCharacter : Start and stop character type	"" (Blank character string)	Start or stop is not limited.	○	✓
	Character string having start character and stop character linked ex) When "AA", "AB", "?" are designated, either start or stop can be designated. ex) "A?", "?D"	Only labels with designated start/stop character are readable.		✓
BarcodeScannerSettings.decode.symbologies.code39: Code39 related				
enabled : Enabling to read	true	Enabling to read	○	✓
	false	Disabling to read		✓
lengthMin : Minimum number of digits	1 to 99	Minimum number of digits	1	✓
lengthMax : Maximun number of digits	1 to 99	Maximum number of digits	99	✓
verifyCheckDigit: Check digit verification	true	Valid		✓
	false	Invalid	○	✓
BarcodeScannerSettings.decode.symbologies.code93: Code93 related				
enabled : Enabling to read	true	Enabling to read	○	✓
	false	Disabling to read		✓
lengthMin : Minimum number of digits	1 to 99	Minimum number of digits	1	✓
lengthMax : Maximun number of dights	1 to 99	Maximum number of digits	99	✓

Property : description	Setting value		Initial value	SP1
BarcodeScannerSettings.decode.symbologies.code128: Code128 related				
enabled : Enabling to read	true	Enabling to read	☐	✓
	false	Disabling to read		✓
lengthMin : Minimum number of digits	1 to 99	Minimum number of digits	1	✓
lengthMax : Maximum number of digits	1 to 99	Maximum number of digits	99	✓
BarcodeScannerSettings.decode.symbologies.msi: MSI related				
enabled : Enabling to read	true	Enabling to read		
	false	Disabling to read	☐	
lengthMin : Minimum number of digits	1 to 99	Minimum number of digits	1	
lengthMax : Maximum number of digits	1 to 99	Maximum number of digits	99	
numberOfCheckDigit Verification : Number of digits to be used for check digit	1	1 digit	☐	
	2	2 digits		
BarcodeScannerSettings.decode.symbologies.gs1DataBar: GS1 Databar related				
enabled : Enabling to read	true	Enabling to read	☐	✓
	false	Disabling to read		✓
stacked : Enabling to read by stack	true	Enabling to read		✓
	false	Disabling to read	☐	✓
BarcodeScannerSettings.decode.symbologies.gs1DataBarLimited: GS1 Databar Limited related				
enabled : Enabling to read	true	Enabling to read	☐	✓
	false	Disabling to read		✓
BarcodeScannerSettings.decode.symbologies.gs1DataBarExpanded: GS1 Databar Expanded related				
enabled : Enabling to read	true	Enabling to read	☐	✓
	false	Disabling to read		✓
stacked : Enabling to read by stack	true	Enabling to read		✓
	false	Disabling to read	☐	✓
lengthMin : Minimum number of digits	1 to 99	Minimum number of digits	1	✓
lengthMax : Maximum number of digits	1 to 99	Maximum number of digits	99	✓
BarcodeScannerSettings.decode.symbologies.gs1Composite: GS1 Composite related				
enabled : Enabling to read	true	Enabling to read	☐	✓
	false	Disabling to read		✓

Property : description	Setting value		Initial value	SP1
BarcodeScannerSettings.decode.symbologies.qrCode: QR code related(*1)				
enabled : Enabling to read	true	Enabling to read	○	✓
	false	Disabling to read		✓
splitMode : Edit/non-edit mode	SplitModeQr.DISABLED	Disabling edit reading	○	✓
	SplitModeQr.EDIT	Edit mode		✓
	SplitModeQr.BATCH_EDIT	Batch edit mode		✓
	SplitModeQr.NON_EDIT	Non-edit mode		✓
BarcodeScannerSettings.decode.symbologies.qrCode.model1: QR Code model 1 code related(*1)				
enabled : Eanbling to read	true	Enabling to read	○	✓
	false	Disabling to read		✓
versionMin : Minimum code version	1 to 22	Value of minimum version	1	✓
versionMax : Maximum code version	1 to 22	Value of maximum version	22	✓
BarcodeScannerSettings.decode.symbologies.qrCode.model2: QR Code model 2 code related(*1)				
enabled : Enabling to read	true	Enabling to read	○	✓
	false	Disabling to read		✓
versionMin : Minimum code version	1 to 40	Value of minimum version	1	✓
versionMax : Maximum code version	1 to 40	Value of maximum version	40	✓
BarcodeScannerSettings.decode.symbologies.microQr: Micro QR Code related(*1)				
enabled : Enabling to read	true	Enabling to read		✓
	False	Disabling to read		✓
versionMin : Minimum code version	1 to 4	Value of minimum version	1	✓
versionMax : Maximum code version	1 to 4	Value of maximum version	4	✓
BarcodeScannerSettings.decode.symbologies.sqrc: SQRC Code related				
enabled : Enabling to read	true	Enabling to read		✓
	False	Disabling to read	○	✓
versionMin : Minimum code version	1 to 40	Value of minimum version	1	✓
versionMax : Maximum code version	1 to 40	Value of maximum version	40	✓
BarcodeScannerSettings.decode.symbologies.iqrCode: iQR Code related				
enabled : Enabling to read	true	Enabling to read		
	false	Disabling to read	○	
splitMode : Edit/non-edit mode	SplitModeliqr.DISABLED	Disabling to read	○	
	SplitModeliqr.EDIT	Edit mode		
	SplitModeliqr.NON_EDIT	Non-edit mode		
BarcodeScannerSettings.decode.symbologies.iqrCode.square : Square iQR Code related				
enabled : Enabling to read	true	Enabling to read		
	false	Disabling to read	○	
versionMin : Minimum code version	1 to 61	Value of minimum version	1	
versionMax : Maximum code version	1 to 61	Value of maximum version	61	
BarcodeScannerSettings.decode.symbologies.iqrCode.rectangle : Rectangular iQR Code related				
enabled : Enabling to read	true	Enabling to read		
	false	Disabling to read	○	
versionMin : Minimum code version	1 to 15	Value of minimum version	1	
versionMax : Maximum code version	1 to 15	Value of maximum version	15	

(*1) When BarcodeScannerSettings.decode.symbologies.qrCode.enabled = false, model1, model2, and microQr are automatically set to false.

Simultaneously with setting BarcodeScannerSettings.decode.symbologies.qrCode.enabled = true, please set model1, model2 and microQr at the same time.

Property : description	Setting value		Initial value	SP1
BarcodeScannerSettings.decode.symbologies.dataMatrix: Data Matrix Code related				
enabled	true	Enabling to read	○	✓
: Enabling to read	false	Disabling to read		✓
BarcodeScannerSettings.decode.symbologies.dataMatrix.square: Data Matrix Square Code related				
enabled	true	Enabling to read	○	✓
: Enabling to read	false	Disabling to read		✓
codeNumberMin : Minimum code number	1 to 24	Value of minimum code number	1	✓
codeNumberMax : Maximum code number	1 to 24	Value of maximum code number	24	✓
BarcodeScannerSettings.decode.symbologies.dataMatrix.rectangle: Data Matrix Rectangular Code related				
enabled	True	Enabling to read	○	✓
: Enabling to read	False	Disabling to read		✓
codeNumberMin : Minimum code number	1 to 6	Value of minimum code number	1	✓
codeNumberMax : Maximum code number	1 to 6	Value of maximum code number	6	✓
BarcodeScannerSettings.decode.symbologies.pdf417: PDF417 Code related				
enabled	True	Enabling to read	○	✓
: Enabling to read	False	Disabling to read		✓
BarcodeScannerSettings.decode.symbologies.microPdf417: Micro PDF417 Code related				
enabled	true	Enabling to read	○	✓
: Enabling to read	false	Disabling to read		✓
BarcodeScannerSettings.decode.symbologies.maxiCode: Maxi Code related				
enabled	true	Enabling to read	○	✓
: Enabling to read	false	Disabling to read		✓
BarcodeScannerSettings.decode.symbologies.plessey: Plessey related				
enabled	true	Enabling to read		
: Enabling to read	false	Disabling to read	○	
BarcodeScannerSettings.decode.symbologies.aztec: Aztec code related				
enabled	true	Enabling to read		
: Enabling to read	false	Disabling to read	○	

Property : description	Setting value		Initial value	SP1
BarcodeScannerSetting.editing : Related to edit of already read data				
BarcodeScannerSetting.editing.ean13 : Related to edit of already read EAN-13 code				
reportCheckDigit : Adding check digit	true	Add	○	✓
	false	Not add		✓
BarcodeScannerSetting.editing.upcA : Related to edit of already read UPC-A code data				
reportCheckDigit : Adding check digit	true	Add	○	✓
	false	Not add		✓
addLeadingZero : Adding top character for adjusting the number of digits of transferred code	true	Add	○	✓
	false	Not add		✓
BarcodeScannerSetting.editing.ean8: Related to edit of already read EAN-8 code data				
reportCheckDigit : Adding check digit	true	Add	○	✓
	false	Not add		✓
convertToEan13 : Conversion to EAN13	true	Convert		✓
	false	Not convert	○	✓
BarcodeScannerSetting.editing.upcE: Related to edit of already read UPC-E code data				
reportCheckDigit : Adding check digit	true	Add	○	✓
	false	Not add		✓
addLeadingZero : Adding top character for adjusting the number of digits of transferred code	true	Add		✓
	false	Not add	○	✓
convertToUpcA : Conversion to UPC-A	true	Convert		✓
	false	Not convert	○	✓
reportNumberSystem CharacterOf ConvertedUpcA : Adding number system when converted to UPC- A	true	Add	○	✓
	false	Not add		✓
BarcodeScannerSetting.editing.code39: Related to edit of already read Code39 code data				
reportCheckDigit : Adding check digit	true	Add	○	✓
	false	Not add		✓
reportStartStopCharacter : Adding start/stop character	true	Add		✓
	false	Not add	○	✓
BarcodeScannerSetting.editing.codabar: Related to edit of already read Codabar code data				
reportCheckDigit : Adding check digit	true	Add	○	✓
	false	Not add		✓
reportStartStopCharacter : Adding start/stop character	true	Add	○	✓
	false	Not add		✓
convertToUpperCase : Convert to capitals of start/stop characters	true	Convert		✓
	false	Not convert	○	✓
BarcodeScannerSetting.editing.itf: Related to edit of already read ITF code data				
reportCheckDigit : Adding check digit	true	Add	○	✓
	false	Not add		✓
BarcodeScannerSetting.editing.stf: Related to edit of already read STF code data				
reportCheckDigit : Adding check digit	true	Add	○	✓
	false	Not add		✓
BarcodeScannerSetting.editing.sqrc: Related to edit of already read SQRC code data				
correctKeyDecode : SQRC data when encryption key is correct	CorrectKeyDecode. PUBLIC_AND_PRIVATE_DATA	Public and private key	○	✓
	CorrectKeyDecode. ONLY_PRIVATE_DATA	Only private key		✓
incorrectKeyDecode : SQRC data when encryption key is incorrect	IncorrectKeyDecode.NONE	None	○	✓
	IncorrectKeyDecode. ONLY_PUBLIC_DATA	Only public data		✓

Property : description	Setting value		Initial value	SP1
BarcodeScannerSettings.PowerOffDelay :				
time :Time to off barcode scanner module after closeBarcode	Min value 0	Time to off (msec)	0	

DENSO SP1 SDK for Android Reference Manual

7th Edition. Sep 2021

DENSO WAVE INCORPORATED AUTO-ID Business Unit

No part of this publication may be reproduced in any form or by any means without permission in writing from the publisher

Specifications are subject to change without prior notice.

In no event will DENSO WAVE be liable for any direct or indirect damages resulting from the application of the information in this manual.