H-33 Scanner SDK

Android Developer’s Guide

H-33 Scanner SDK

Android Developer's Guide

Version:0.2　2019年12月発行

株式会社オプトエレクトロニクス

Contents

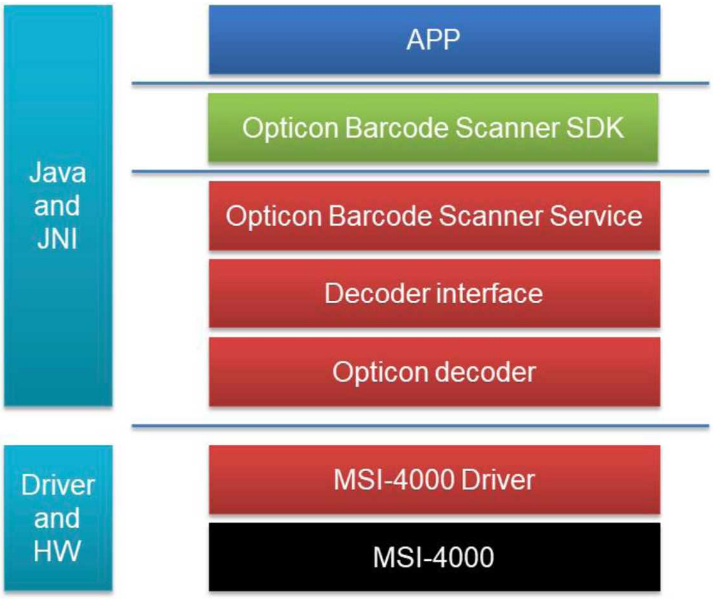
1 Introduction

This document is intended for developers to create scanner Android applications for the Opticon H-33. The application can connect through this SDK to retrieve barcodes and control scanner settings. An example on how to do this is included.

* Developers are assumed to have knowledge of Android programming.

1.1 Opticon barcode scanner SDK structure

To develop a scan application more easily, the Opticon SDK integrates the scan engine and SDK into one control method.



1.2 Development environment

1.2.1 Application development tool

The following development tools are required for Android application development.

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

- Java SE Development kit (JDK, Java SE 8 or upper are recommended)

https://www.java.com/download/

- Android Studio

https://developer.android.com/studio/

- Android SDK

[Android Studio] → [Tools] → [SDK Manager] → [Android SDK]

* To get the below software tools, please install.

-[SDK Platforms] → [Android 9.0(pie) (API Level 28)]

-[SDK Tools] → [AndroidSDK Build-Tools]

-[SDK Tools] → [Google USB Driver]

1.2.2 Application development Debug

The following preparations are required to develop / debug applications using H-33.

* The following steps were created with Android Stuido 3.5.

1. Enable USB debugging (ADB connection) of H-33.

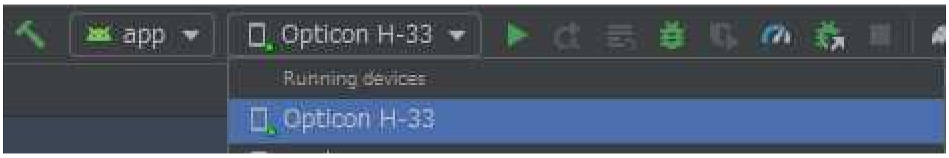
[H-33] → [Settings] → [About phone]

(1) Tap on Build number option 7 times and then a toast will be shown saying "Developer options are enabled.

(2) Turn on the check box "USB debugging" in the "Developer options".

(3) Please select "Opticon H-33"

[Android Studio] → [Run] → [Debug]



1.2.3 Procedures for using the SDK in development environments

The following must be performed before using this SDK's class library.

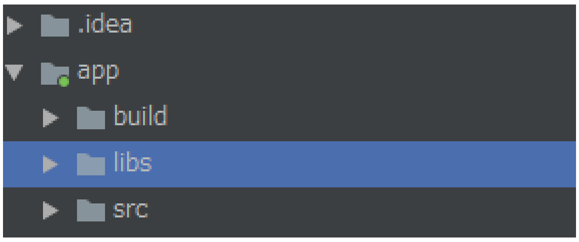
- Import [scannersdk.jar] into the development environment.

- Import the [scannersdk.jar] to Android Studio.

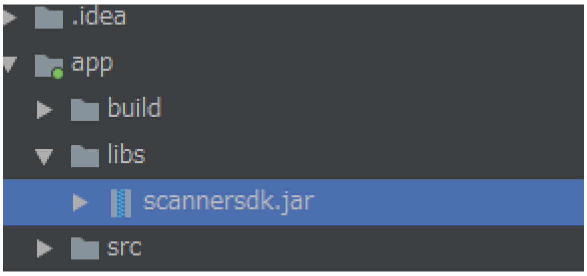
*Refer to the following website for how to import libraries.

https://developer.android.com/studio/projects/android-library

(1) Create a folder named **libs** in the Android project.

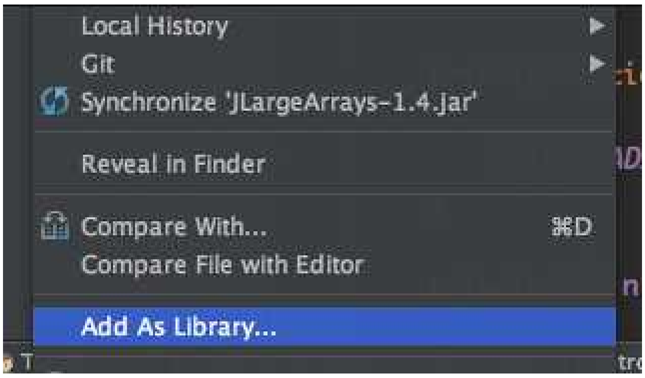


(2) Copy scannersdk.jar to the libs folder



(3) Right-click the [scannersdk.jar] in the project view and click Add As Library.

It has been added to the build.gradle file in the app.



* The minSdkVersion of this library is **<u>28.</u>**

1.3 API usage

This section uses sample code to explain how the use the SDK’s main API. The sampleSDK code, included with the SDK, will be used.

To use the barcode Library class in the source code, import the class and create an instance as follows.

Package List:

Package Name	Description
com.extbcr.scannersdk.BarcodeManager	Control the barcode behavior, like start decode.
com.extbcr.scannersdk.EventListener	Barcode service status changed, or get decoded data.
com.extbcr.scannersdk.BarcodeData	Barcode Data.
com.extbcr.scannersdk.BarcodeDataEx	An extended BarcodeData class.
com.extbcr.scannersdk.AllSettings	Setting symbology information.
com.extbcr.scannersdk.PropertyID	Property ID.
com.extbcr.scannersdk.CodeID	Barcode type

Refer to the next chapter for API details.

1.3.1 Initialization of the barcode SDK

Sample:

```
@Override
public void onResume() {
    super.onResume();
    //Must use BarcodeManager connect to service
    mBarcodeManager = new BarcodeManager(this);
    mEventListener = new EventListener() {
    //Here need to override interface function.
   };
    //Initialize and connect to Scanner service
    mBarcodeManager.init();
}
```

1.3.2 How to add events

Sample:

```
mEventListener = new EventListener() {
   //You must implement the eventlistener
};
mBarcodeManager.addListener(mEventListener);
```

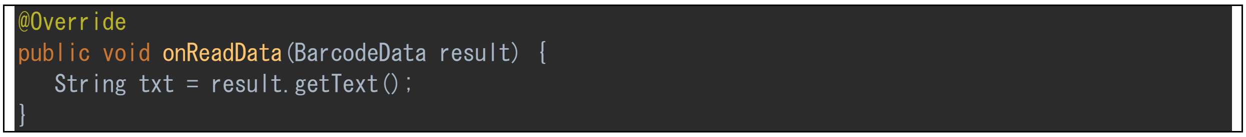
1.3.3 How to trigger the barcode module

Sample:

```
//After get onConnect callback.
mBarcodeManager.startDecode();
```

1.3.4 How to read a scanned barcode

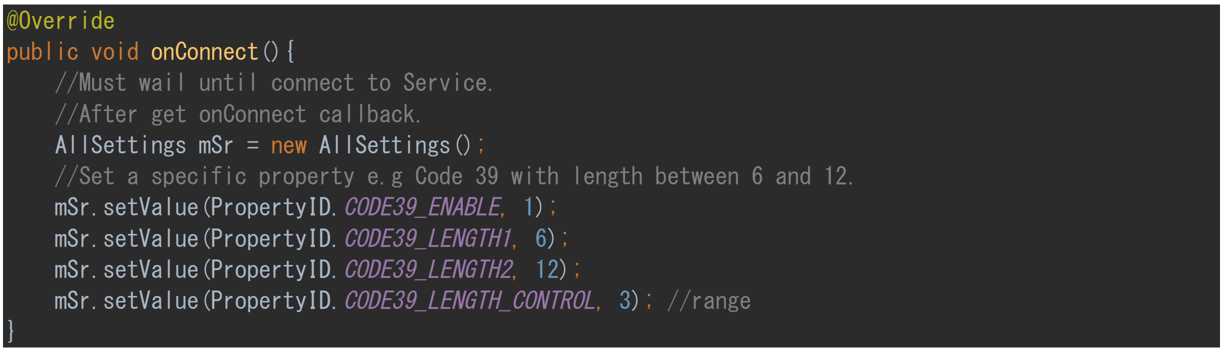
Sample:



```
@Override
public void onReadData(BarcodeData result) {
   String txt = result.getText();
}
```

1.3.5 How to change properties

Sample:



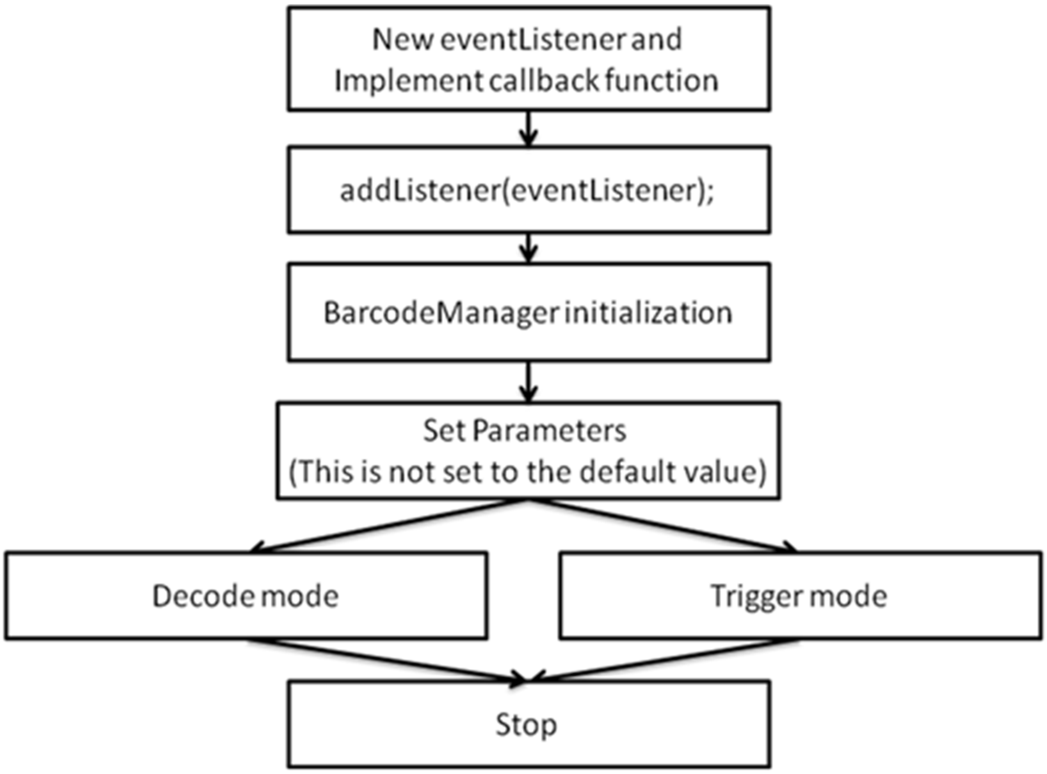
```
@Override
public void onConnect(){
    //Must wail until connect to Service.
    //After get onConnect callback.
    AllSettings mSr = new AllSettings();
    //Set a specific property e.g Code 39 with length between 6 and 12.
    mSr.setValue(PropertyID.CODE39_ENABLE, 1);
    mSr.setValue(PropertyID.CODE39_LENGTH1, 6);
    mSr.setValue(PropertyID.CODE39_LENGTH2, 12);
    mSr.setValue(PropertyID.CODE39_LENGTH_CONTROL, 3); //range
}
```

2 Opticon Barcode Scanner SDK Workflow

This section will introduce the decode workflow.

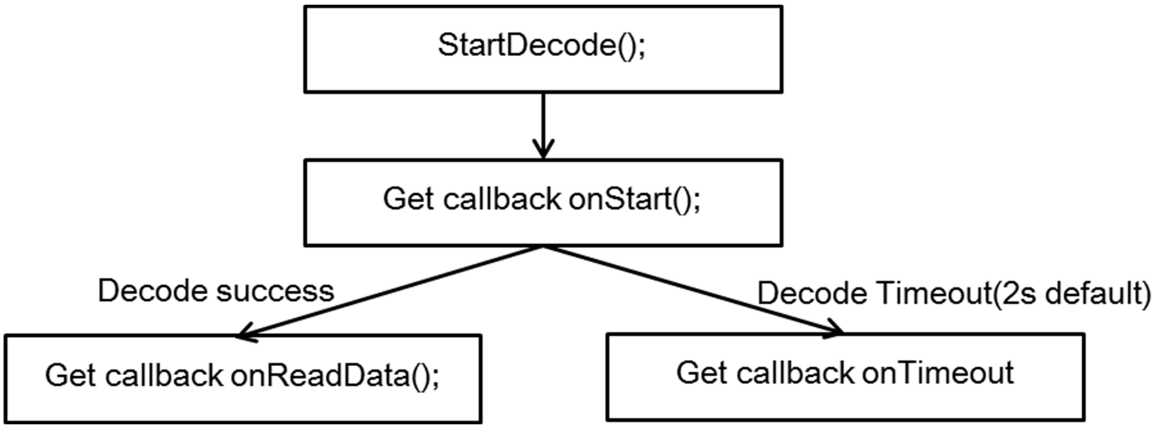
Opticon barcode scanner SDK provides two decode modes; decode mode and trigger mode.

User needs to implement callback function and initialize BarcodeManager then can start decode.


New eventListener and Implement callback function
addListener(eventListener);
BarcodeManager initialization
Set Parameters (This is not set to the default value)
Decode mode
Trigger mode
Stop

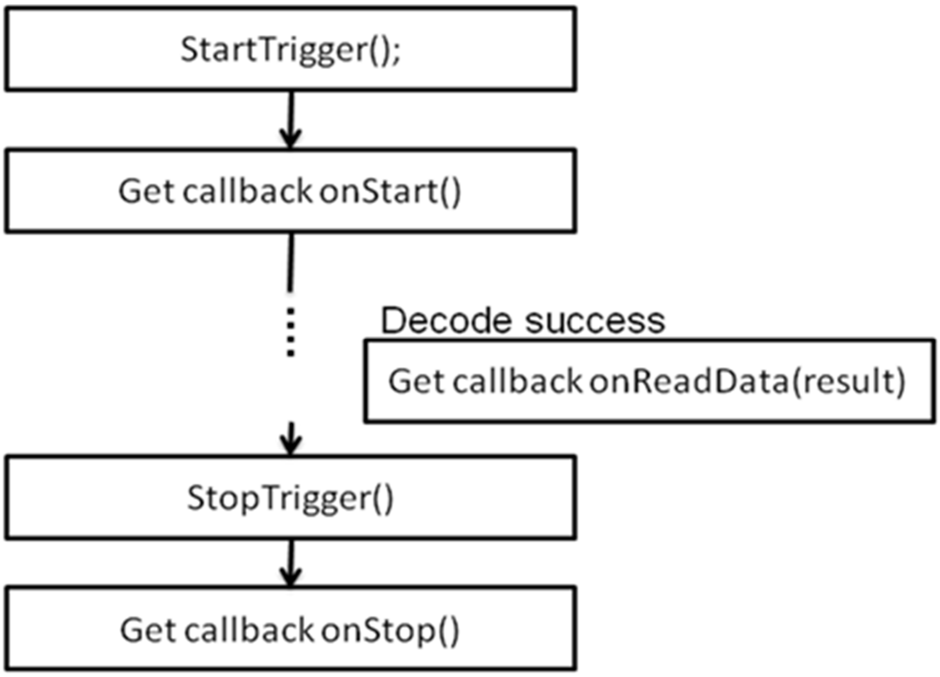
2.1 **Decode mode**

Decode mode only scan once until timeout, the default timeout is 2 sec. If decode process is not successful within timeout period, it will auto stop and onTimeout callback function is called. You can also manual stop by call stopDecode() function.



2.2 **Trigger Mode**

When trigger mode is started it will keep decoding until stopTrigger() is called. On every successful decode the onReadData() callback function is called.

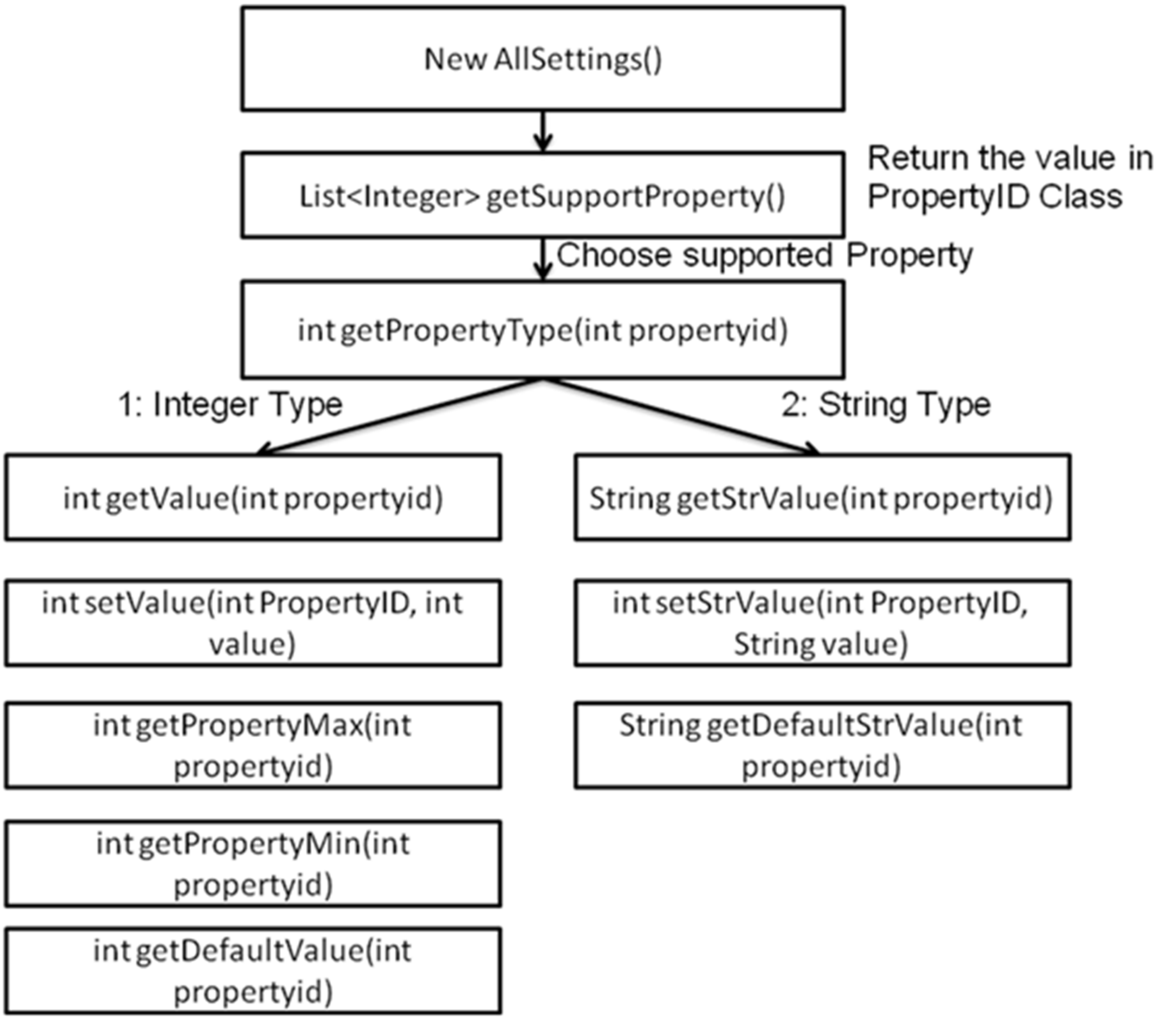


2.3 Setting Parameters

All barcode settings need to set by the AllSettings class, the detail API showed below.

When you use the settings function, you need to init the BarcodeManager.

Until the Service callback onConnect() is called, you can start setting.


New AllSettings()
List<Integer> getSupportProperty()
Return the value in PropertyID Class
Choose supported Property
int getPropertyType(int propertyid)
1: Integer Type
2: String Type
int getValue(int propertyid)
int setValue(int PropertyID, int value)
int getPropertyMax(int propertyid)
int getPropertyMin(int propertyid)
int getDefaultValue(int propertyid)
String getStrValue(int propertyid)
int setStrValue(int PropertyID, String value)
String getDefaultStrValue(int propertyid)

3 Opticon Barcode Scanner SDK Class

This chapter will introduce the Opticon Barcode scanner SDK class.

3.1 BarcodeManager

Control the barcode behavior, like start decode.

```
Class BarcodeManager{
  void startDecode();
  void stopDecode();
  void startTrigger();
  void stopTrigger();
  void addListener(EventListener listener);
  void removeListener();
  void enableAllSymbologies();
  void disableAllSymbologies();
  void resetAllSymbologies();
  void init();
  void deinit();
  void takeSnapshot(int subSampling, int bitPerPixel, int imageType, int jpegQuality);
  int backupCurrentAllSettings(String outPutFile);
  int restoreCurrentAllSettings(String inPutFile)
  BarcodeDataEx getBarcodeDataEx ();
}
```

Note: must use BarcodeManager connect to service.

3.1.1 startDecode

Description	Start decode function. It will auto stop while decode success or timeout. Need to linked with stopDecode().
Syntax	void startDecode();
Parameters	void
Return value	void
Exception	None
Sample coding	mBarcodeManager.startDecode(); Please refer to 4.2 ScanBarcode sample for details.

3.1.2 stopDecode

Description	Stop decode function. and Stop decode before timeout Need to linked with startDecode().
Syntax	void stopDecode();
Parameters	void
Return value	void
Exception	None
Sample coding	mBarcodeManager.stopDecode(); Please refer to 4.2 ScanBarcode sample for details.

3.1.3 startTrigger

Description	Start infinite decode. If decode success, will restart decode.

	Need to linked with stopTrigger().
Syntax	void startTrigger();
Parameters	void
Return value	void
Exception	None
Sample coding	mBarcodeManager.startTrigger(); Please refer to 4.2 ScanBarcode sample for details.

3.1.4 stopTrigger

Description	Stop trigger mode. Need to linked with startTrigger().
Syntax	void stopTrigger();
Parameters	void
Return value	void
Exception	None
Sample coding	mBarcodeManager.stopTrigger();Please refer to 4.2 ScanBarcode sample for details.

3.1.5 addListener(EventListener listener)

Description	Add event callback. When service had status changed, it will callback. Better setup before calling init() function or set it immediately after call init().
Syntax	void addListener(EventListener listener);
Parameters	EventListener
Return value	void
Exception	None

Sample coding	//The eventlistner need to be implemented first.
	mBarcodeManager.addListener(this);
	Please refer to 4.2 ScanBarcode sample for details.

3.1.6 removeListener

Description	Remove current event callback.
Syntax	void removeListener();
Parameters	void
Return value	void
Exception	None

Sample coding	mBarcodeManager.removeListener(); Please refer to 4.2 ScanBarcode sample for details.

3.1.7 enableAllSymbologies

Description	Enable all supported symbologies. *Except Add-on code and OCR code.
Syntax	void enableAllSymbologies();
Parameters	void
Return value	void
Exception	None
Sample coding	mBarcodeManager.enableAllSymbologies();

3.1.8 disableAllSymbologies

Description	Disable all supported symbologies.
Syntax	void disableAllSymbologies();
Parameters	void
Return value	void
Exception	None
Sample coding	mBarcodeManager.disableAllSymbologies ();

3.1.9 resetAllSymbologies

Description	Reset all settings of supported symbologies to default.
Syntax	void resetAllSymbologies();
Parameters	None
Return value	None
Exception	None
Sample coding	mBarcodeManager.resetAllSymbologies();

3.1.10 init

Description	Init the BarcodeManager and connect to service.
Syntax	void init();
Parameters	void
Return value	void
Exception	None

Sample coding	mBarcodeManager.init(); Please refer to 4.2 ScanBarcode sample for details.

3.1.11 **deinit**

Description	Release the connection.
Syntax	void deinit();
Parameters	void
Return value	void
Exception	None
Sample coding	mBarcodeManager.deinit();Please refer to 4.2 ScanBarcode sample for details.

3.1.12 **takeSnapshot**

Description	Use scanner to take snapshot.
Syntax	void takeSnapshot(int subSampling, int bitPerPixel, int imageType, int jpegQuality);
Parameters	int subSampling: 1, 2, 4, 8 int bitPerPixel: 1, 4, 8, 10 int imageType: Jpeg(1), Bmp(3) int jpegQuality: 5~100 (%)
Return value	void
Exception	None
Sample coding	mBarcodeManager.takeSnapshot(1, 8, 3, 65);

3.1.13 **backupCurrentAllSettings**

Description	Backup current setting to outPutFile.
Syntax	int backupCurrentAllSettings(String outPutFile)
Parameters	int
Return value	If backup fail will return -1. If success will return 0.
Exception	None
Sample coding	

3.1.14 **restoreCurrentAllSettings**

Description	Restore setting from inPutFile.
Syntax	int restoreCurrentAllSettings(String inPutFile)

Parameters	int
Return value	If restore fail will return -1. If restore will return 0.
Exception	None
Sample coding	

3.1.15 getBarcodeDataEx

Description	Use to get last barcode data with more information.
Syntax	BarcodeDataEx getBarcodeDataEx ()
Parameters	void
Return value	BarcodeDataEx
Exception	None
Sample coding	

3.2 EventListener

This section will introduce the listener and callback function. User need to register listener and override the callback function.

```
  interface EventListener{
    void onReadData(BarcodeData result);
    void onStart();
    void onStop();
    void onTimeout();
    void onConnect();
    void onDisconnect();
    void onImgBuffer(byte[] imgdata, int type);
}
```

3.2.1 onReadData

Description	Decode success will get this callback, it can get the decode data in result.
Syntax	void onReadData(BarcodeData result);
Parameters	BarcodeData result
Return value	void
Exception	None
Sample coding	@Override void onReadData(BarcodeData result) {String txt = result.getText();} Please refer to 4.2 ScanBarcode sample for details.

3.2.2 onStart

Description	StartDecode() or StartTrigger() success will callback this function.
Syntax	void onStart();
Parameters	void
Return value	void
Exception	None
Sample coding	@Override void onStart() { } Please refer to 4.2 ScanBarcode sample for details.

3.2.3 onStop

Description	StopDecode() or StopTrigger() success will callback this function.
Syntax	void onStop();
Parameters	void
Return value	void
Exception	None
Sample coding	@Override void onStop() { } Please refer to 4.2 ScanBarcode sample for details.

3.2.4 onTimeout

Description	This callback will be called when timeout expires and the scanner was not able to make a successful decode. Default timeout setting is 2 sec.
Syntax	void onTimeout();
Parameters	void
Return value	void
Exception	None
Sample coding	@Override void onTimeout() { } Please refer to 4.2 ScanBarcode sample for details.

3.2.5 onConnect

Description	ScannerSDK will callback this function after connecting to ScannerServer.
Syntax	void onConnect();
Parameters	void
Return value	void

Exception	None
Sample coding	@Override void onConnect() { } Please refer to 4.2 ScanBarcode sample for details.

3.2.6 onDisconnect

Description	ScannerSDK will callback this function after disconnecting from ScannerServer.
Syntax	void onDisconnect();
Parameters	void
Return value	void
Exception	None
Sample coding	@Override void onDisconnect() { } Please refer to 4.2 ScanBarcode sample for details.

3.2.7 onImgBuffer

Description	ScannerSDK will callback this function after snapshot success.
Syntax	void onImgBuffer();
Parameters	byte[] imgdata, int type type: RAW(0), JPEG(1), BMP(3)
Return value	void
Exception	None

Sample coding	@Override
	void onImgBuffer(byte[] imgdata, int type) {
	if(type == 3){
	Bitmap bmp = BitmapFactory.decodeByteArray(imgdata, 0,
	imgdata.length);mImageView.setImageBitmap(bmp);}
	}
	Please refer to 4.2 ScanBarcode sample for details.

3.3 BarcodeData

This section will introduce the decode success result passed from onReadData callback function.

```
  class BarcodeData{
            String getText();
            CodeID getCodeID();
            byte[] getRawData();
  }
```

3.3.1 getText

Description	Get string of decode result.
Syntax	String getText();
Parameters	void
Return value	String.
Exception	None
Sample coding	@Override void onReadData(BarcodeData result) {String txt = result.getText();} Please refer to 4.2 ScanBarcode sample for details.

3.3.2 getCodeID

Description	Get code type of decode result.
Syntax	CodeID getCodeID();
Parameters	void
Return value	CodeID.
Exception	None

Sample coding	@Override void onReadData(BarcodeData result) {
	CodeID type = result.getCodeID();}
	Please refer to 4.2 ScanBarcode sample for details.

3.3.3 getRawData

Description	Get byte array data of decode result.
Syntax	byte[] getRawData();
Parameters	void
Return value	byte[]
Exception	None

Sample coding	@Override void onReadData(BarcodeData result) {
	byte[] a = result.getRawData();}
	Please refer to 4.2 ScanBarcode sample for details.

3.4 BarcodeDataEx

The BarcodeDataEx class is not used as a callback argument and is only created when the user gets the raw data. Add getBarcodeDataEx () to BarcodeManager class so that the user can get it.

```
class BarcodeDataEx  {
        public byte[] getRawData() ;
        public String getText() ;
        public int getCodeID() ;
        public int[] getSymbolAreaX() ;
        public int[] getSymbolAreaY() ;
        public int getOpticonId() ;
        public string getAimId();
        public int getImageType() ;
        public byte[] getImageRaw();
       public Bitmap getBitmapImage ();
 }
```

3.4.1 getRawData

Description	Get byte array data of decode result.
Syntax	byte[] getRawData ();
Parameters	void
Return value	Byte data of decoded barcode.
Exception	None
Sample coding	

3.4.2 getText

Description	Get string of decode result.
Syntax	String getText ();
Parameters	void
Return value	Byte data of decoded barcode.
Exception	None
Sample coding	

3.4.3 getCodeID

Description	Get code type of decode result.
Syntax	int getCodeID ();
Parameters	void
Return value	Code type of decoded barcode.
Exception	None
Sample coding	

3.4.4 getSymbolAreaX

Description	Get the X coordinate of barcode.
Syntax	int[] getSymbolAreaX();
Parameters	void
Return value	X coordinate of decoded barcode.
Exception	None
Sample coding	

3.4.5 getSymbolAreaY

Description	Get the Y coordinate of barcode.
Syntax	int[] getSymbolAreaY();
Parameters	void
Return value	Y coordinate of decoded barcode.
Exception	None
Sample coding	

3.4.6 getOpticonId

Description	Get the opticon ID.
Syntax	int getOpticonId ();
Parameters	void
Return value	id number.
Exception	None
Sample coding	

3.4.7 getAimId

Description	Get the AIM ID.
Syntax	String getAimId ();
Parameters	void
Return value	string AIM ID.
Exception	None

Sample coding	

3.4.8 getImageType

Description	Get the image type.
Syntax	int getImageType ();
Parameters	void
Return value	Barcode image type, 0: RAW, 1: Jpeg, 3: Bmp.
Exception	None
Sample coding	

3.4.9 getImageRaw

Description	Get the image data.
Syntax	byte[] getImageRaw ();
Parameters	void
Return value	Barcode image.
Exception	None
Sample coding	

3.4.10 getBitmapImage

Description	Get the bitmap of barcode image.
Syntax	Bitmap getBitmapImage ();
Parameters	void
Return value	bitmap.
Exception	None
Sample coding	

3.5 AllSettings

```
class AllSettings{
        List<Integer> getSupportProperty();
        int getPropertyType(int propertyId);
```

```
            int getPropertyMax(int propertyId);
            int getPropertyMin(int propertyId);
            int getValue(int propertyId);
            String getStrValue(int propertyId);
            int setValue(int propertyId, int value);
            int setStrValue(int propertyId, String value);
            int getDefaultValue(int propertyId);
           String getDefaultStrValue(int propertyId);
}
```

Note:

must use BarcodeManager connect to service.

If type is int, use setValue function.

If type is String, use setStrValue function.

3.5.1 getSupportProperty

Description	Return supported property in PropertyID class.
Syntax	List<Integer> getSupportProperty();
Parameters	void
Return value	List<Integer>
Exception	None
Sample coding	List<Integer> list = mAllsettings.getSupportProperty(); Please refer to 4.3 Barcode Setting Sample for details.

3.5.2 getPropertyType

Description	Get property type
Syntax	int getPropertyType(int propertyId);
Parameters	propertyId
Return value	1: Integer type, 2: String type
Exception	None

Sample coding	// code39_enable is int type, type = 1.
	int type = mAllsettings.getPropertyType(PropertyID.CODE39_ENABLE);
	Please refer to 4.3 Barcode Setting Sample for details.

3.5.3 getValue

Description	Get value of Property, the Property type should be integer type.
Syntax	int getValue(int propertyId);
Parameters	propertyId

Return value	Return value of propetry. If property is String type, will get exception.
Exception	If property is String type, will get cast exception.
Sample coding	int enable = mAllsettings.getValue(PropertyID.CODE39_ENABLE); Please refer to 4.3 Barcode Setting Sample for details.

3.5.4 getPropertyMax

Description	Get maximum value of Property, the Property type should be integer type.
Syntax	int getPropertyMax(int propertyId);
Parameters	propertyId
Return value	Return maximumvalue of Property. If property does not exist, it will return -1.
Exception	None

Sample coding	// code39_enable max is 1, means enable code39.
	int max = mAllsettings.getPropertyMax(PropertyID.CODE39_ENABLE);
	Please refer to 4.3 Barcode Setting Sample for details.

3.5.5 getPropertyMin

Description	Get minimum value of Property, the Property type should be integer type.
Syntax	int getPropertyMin(int propertyId);
Parameters	propertyId
Return value	Return minimumvalue of Property. If property does not exist, it will return -1.
Exception	None

Sample coding	// code39_enable min is 0, means disable code39.
	int type = mAllsettings.getPropertyMin(PropertyID.CODE39_ENABLE);
	Please refer to 4.3 Barcode Setting Sample for details.

3.5.6 setValue

Description	Set value of Property, the Property type should be integer type.
Syntax	int setValue(int propertyId, int value);
Parameters	propertyId, value
Return value	Return 0. If value is not in range of max and min, it will be ignored.
Exception	None

Sample coding	// means enable code39.
	mAllsettings.setValue(PropertyID.CODE39_ENABLE, 1);

	Please refer to 4.3 Barcode Setting Sample for details.

3.5.7 getStrValue

Description	Get value of Property, the Property type should be String type.
Syntax	String getStrValue(int propertyId)
Parameters	propertyId
Return value	Return String value. If property is integer type, it will return string result. If unknown propertyId is used, it will return "".
Exception	None
Sample coding	String value = mAllsettings.getStrValue(PropertyID.DECODE_COMMAND); Please refer to 4.3 Barcode Setting Sample for details.

3.5.8 setStrValue

Description	Set value of Property, the Property type should be String type.
Syntax	int setStrValue(int propertyId, String value);
Parameters	propertyId, value
Return value	Return 0 If property is integer type, it still return 0.
Exception	None
Sample coding	int r = mAllsettings.setStrValue(PropertyID.DECODE_COMMAND, "RY$1"); Please refer to 4.3 Barcode Setting Sample for details.

3.5.9 getDefaultValue

Description	Get default value of Property, the Property type should be integer type.
Syntax	getDefaultValue(int propertyId);
Parameters	propertyId
Return value	Return default value. If property is string type, it still return 0.
Exception	None

Sample coding	// code39_enable default is 1(enable).
	int value = mAllsettings.getDefaultValue(PropertyID.CODE39_ENABLE);
	Please refer to 4.3 Barcode Setting Sample for details.

3.5.10 getDefaultStrValue

Description	Get default value of Property, the Property type should be String type.
Syntax	String getDefaultStrValue(int propertyId);
Parameters	propertyId
Return value	Return String value If property is integer type, it will return "".
Exception	None
Sample coding	String value = mAllsettings.getStrValue(PropertyID.DECODE_COMMAND); Please refer to 4.3 Barcode Setting Sample for details.

3.6 PropertyID

```
public class PropertyID {
        public static final int SCANKEY_ENABLE                        = 0x0000;
        public static final int INPUTKEY_ENABLE                       = 0x0002;
        public static final int INPUTKEY_TYPE                         = 0x000A;
        public static final int GOODREAD_ENABLE                       = 0x0010;
        public static final int GOODREAD_LED_ENABLE                   = 0x0012;
        public static final int GOODREAD_VIBRATOR_ENABLE              = 0x0013;
        public static final int GOODREAD_AUDIO_ENABLE                 = 0x0014;
        public static final int GOODREAD_BEEPPER_FRQ                  = 0x0015;
        public static final int GOODREAD_BEEPPER_DURATION             = 0x0016;
        public static final int GOODREAD_AIMING                       = 0x0018;

        public static final int FORMATING_PREFIX                      = 0x0021;
        public static final int FORMATING_SUFFIX                      = 0x0022;
        public static final int FORMATING_REMOVE_NONPRINT             = 0x0024;
        public static final int FORMATING_NONPRINT_READABLE           = 0x0025;
        public static final int FORMATING_GS_REPLACE                  = 0x0026;
        public static final int FORMATING_OPTICON_ID                  = 0x0027;
        public static final int FORMATING_AIM_ID                      = 0x0028;
        public static final int FORMATING_STRING_ENODING_SET          = 0x0029;
        public static final int WEDGE_INTENT_ENABLE                   = 0x0030;
        public static final int WEDGE_INTENT_DELIVERY_MODE            = 0x0031;
        public static final int WEDGE_INTENT_ACTION                   = 0x0032;
        public static final int WEDGE_INTENT_CATEGORY                 = 0x0033;
        public static final int WEDGE_INTENT_BARCODETYPE              = 0x0034;
        public static final int WEDGE_INTENT_BARCODEDATA           = 0x0035;
        public static final int WEDGE_INTENT_PACKAGE_NAME          = 0x0036;

        public static final int GOODREAD_LED_DURATION              = 0x0040;
        public static final int GOODREAD_LED_INTERVAL              = 0x0041;
        public static final int GOODREAD_LED_TIMES                 = 0x0042;
        public static final int GOODREAD_LED_COLOR                 = 0x0043;
        public static final int READTIMEOUT_LED_ENABLE             = 0x0044;
        public static final int READTIMEOUT_LED_DURATION           = 0x0045;
        public static final int READTIMEOUT_LED_INTERVAL           = 0x0046;
        public static final int READTIMEOUT_LED_TIMES              = 0x0047;
        public static final int READTIMEOUT_LED_COLOR              = 0x0048;

        public static final int GOODREAD_VIBRATOR_DURATION         = 0x0049;
        public static final int GOODREAD_VIBRATOR_INTERVAL         = 0x004a;
        public static final int GOODREAD_VIBRATOR_TIMES            = 0x004b;
        public static final int READTIMEOUT_VIBRATOR_ENABLE        = 0x004c;
        public static final int READTIMEOUT_VIBRATOR_DURATION      = 0x004d;
        public static final int READTIMEOUT_VIBRATOR_INTERVAL      = 0x004e;
        public static final int READTIMEOUT_VIBRATOR_TIMES         = 0x004f;

        public static final int GOODREAD_BEEPPER_INTERVAL          = 0x0050;
        public static final int GOODREAD_BEEPPER_TIMES             = 0x0051;
        public static final int READTIMEOUT_AUDIO_ENABLE           = 0x0054;
        public static final int READTIMEOUT_BEEPPER_FRQ            = 0x0055;
        public static final int READTIMEOUT_BEEPPER_DURATION       = 0x0056;
        public static final int READTIMEOUT_BEEPPER_INTERVAL       = 0x0057;
        public static final int READTIMEOUT_BEEPPER_TIMES          = 0x0058;

        public static final int READ_MODE                          = 0x0071;
        public static final int READ_TIME                          = 0x0072;
        public static final int MARGIN_CHECK                       = 0x0073;
        public static final int REDUNDANCY                         = 0x0074;
        public static final int NEGATIVE_BARCODE                   = 0x0075;
        public static final int TRIGGER_REPEAT                     = 0x0076;
        public static final int DECODE_COMMAND                     = 0x0077;
        public static final int CENTRAL_READING_AREA               = 0x0078;
        public static final int MAX_UNIQUE_READ                    = 0x0079;
        public static final int MULTIPLE_READ_RESET_TIME           = 0x007A;
        public static final int CENTRAL_READING                 = 0x0004;
        public static final int AIMER_MODE                      = 0x0008;
        public static final int CAMERA_ILLUMINATION_MODE        = 0x0009;

        public static final int DATA_EDIT_PROGRAMMINIG_INDEX    = 0x0080;
        public static final int DATA_EDIT_PROGRAMMINIG_CUT      = 0x0081;
        public static final int DATA_EDIT_PROGRAMMINIG_PASTE    = 0x0082;

        public static final int CODE39_ENABLE                   = 0x0100;
        public static final int CODE39_LENGTH1                  = 0x0101;
        public static final int CODE39_LENGTH2                  = 0x0102;
        public static final int CODE39_LENGTH_CONTROL           = 0x0103;
        public static final int CODE39_MODE                     = 0x0121;
        public static final int CODE39_CONCATENATION_ENABLE     = 0x0122;
        public static final int CODE39_STSP_TRANSMISSION        = 0x0123;
        public static final int CODE39_CHECK_DIGIT              = 0x0104;
        public static final int CODE39_CD_TRANSMISSION          = 0x0105;

        public static final int INDUSTRIAL_2OF5_ENABLE          = 0x0200;
        public static final int INDUSTRIAL_2OF5_LENGTH1         = 0x0201;
        public static final int INDUSTRIAL_2OF5_LENGTH2         = 0x0202;
        public static final int INDUSTRIAL_2OF5_LENGTH_CONTROL  = 0x0203;

        public static final int MATRIX_2OF5_ENABLE              = 0x0300;
        public static final int MATRIX_2OF5_LENGTH1             = 0x0301;
        public static final int MATRIX_2OF5_LENGTH2             = 0x0302;
        public static final int MATRIX_2OF5_LENGTH_CONTROL      = 0x0303;

        public static final int INTERLEAVED_2OF5_ENABLE         = 0x0400;
        public static final int INTERLEAVED_2OF5_LENGTH1        = 0x0401;
        public static final int INTERLEAVED_2OF5_LENGTH2        = 0x0402;
        public static final int INTERLEAVED_2OF5_LENGTH_CONTROL = 0x0403;

        public static final int CODABAR_ENABLE                  = 0x0500;
        public static final int CODABAR_LENGTH1                 = 0x0501;
        public static final int CODABAR_LENGTH2                 = 0x0502;
        public static final int CODABAR_LENGTH_CONTROL          = 0x0503;
        public static final int CODABAR_MODE                    = 0x0521;
        public static final int CODABAR_SPACE_INSERTION         = 0x0505;
        public static final int CODABAR_CHECK_DIGIT             = 0x0504;
        public static final int CODABAR_CD_TRANSMISSION         = 0x0507;
        public static final int CODABAR_STSP_TRANSMISSION       = 0x0506;

        public static final int CODE93_ENABLE                   = 0x0600;
        public static final int CODE93_LENGTH1                  = 0x0601;
        public static final int CODE93_LENGTH2                  = 0x0602;
        public static final int CODE93_LENGTH_CONTROL           = 0x0603;
        public static final int CODE93_CHECK_DIGIT              = 0x0604;

        public static final int CODE128_ENABLE                  = 0x0700;
        public static final int CODE128_LENGTH1                 = 0x0701;
        public static final int CODE128_LENGTH2                 = 0x0702;
        public static final int CODE128_LENGTH_CONTROL          = 0x0703;
        public static final int CODE128_CONCATENATION_ENABLE    = 0x0722;

        public static final int UPCA_ENABLE                     = 0x0800;
        public static final int UPCA_LEADING_ZERO               = 0x0803;
        public static final int UPCA_CD_TRANSMISSION            = 0x0802;
        public static final int UPCA_EXT_ENABLE_2_DIGIT         = 0x0821;
        public static final int UPCA_EXT_ENABLE_5_DIGIT         = 0x0822;
        public static final int UPCA_EXT_ENABLE_EXT_DIGIT       = 0x0823;

        public static final int UPCE_ENABLE                     = 0x0900;
        public static final int UPCE_LEADING_ZERO               = 0x0903;
        public static final int UPCE_CD_TRANSMISSION            = 0x0902;
        public static final int UPCE_EXT_ENABLE_2_DIGIT         = 0x0921;
        public static final int UPCE_EXT_ENABLE_5_DIGIT         = 0x0922;
        public static final int UPCE_EXT_ENABLE_EXT_DIGIT       = 0x0923;

        public static final int EAN13_ENABLE                    = 0x0a00;
        public static final int EAN13_CD_TRANSMISSION           = 0x0a01;
        public static final int EAN13_EXT_ENABLE_2_DIGIT        = 0x0a21;
        public static final int EAN13_EXT_ENABLE_5_DIGIT        = 0x0a22;
        public static final int EAN13_EXT_ENABLE_EXT_DIGIT      = 0x0a23;

        public static final int EAN8_ENABLE                     = 0x0b00;
        public static final int EAN8_CD_TRANSMISSION            = 0x0b01;
        public static final int EAN8_EXT_ENABLE_2_DIGIT              = 0x0b21;
        public static final int EAN8_EXT_ENABLE_5_DIGIT              = 0x0b22;
        public static final int EAN8_EXT_ENABLE_EXT_DIGIT            = 0x0b23;

        public static final int MSI_PLESSEY_ENABLE                   = 0x0c00;
        public static final int MSI_PLESSEY_LENGTH1                  = 0x0c01;
        public static final int MSI_PLESSEY_LENGTH2                  = 0x0c02;
        public static final int MSI_PLESSEY_LENGTH_CONTROL           = 0x0c03;
        public static final int MSI_PLESSEY_CHECK_DIGIT              = 0x0c04;
        public static final int MSI_PLESSEY_CD_TRANSMISSION          = 0x0c06;

        public static final int GS1_DATABAR_ENABLE                   = 0x0d00;
        public static final int GS1_DATABAR_LENGTH1                  = 0x0d01;
        public static final int GS1_DATABAR_LENGTH2                  = 0x0d02;
        public static final int GS1_DATABAR_LENGTH_CONTROL           = 0x0d03;
        public static final int GS1_DATABAR_AI_TRANSMISSION          = 0x0d21;
        public static final int GS1_DATABAR_CD_TRANSMISSION          = 0x0d22;
        public static final int GS1_DATABAR_LIMITED_ENABLE           = 0x0e00;
        public static final int GS1_DATABAR_EXPANDED_ENABLE          = 0x0f00;

        public static final int PDF417_ENABLE                        = 0x1000;
        public static final int PDF417_LENGTH1                       = 0x1001;
        public static final int PDF417_LENGTH2                       = 0x1002;
        public static final int PDF417_LENGTH_CONTROL                = 0x1003;

        public static final int DATAMATRIX_LENGTH1                   = 0x1101;
        public static final int DATAMATRIX_LENGTH2                   = 0x1102;
        public static final int DATAMATRIX_LENGTH_CONTROL            = 0x1103;
        public static final int DATAMATRIX_ECC200_ENABLE             = 0x1121;

        public static final int MAXICODE_ENABLE                      = 0x1200;
        public static final int MAXICODE_LENGTH1                     = 0x1201;
        public static final int MAXICODE_LENGTH2                     = 0x1202;
        public static final int MAXICODE_LENGTH_CONTROL              = 0x1203;

        public static final int TRIOPTIC_ENABLE                      = 0x1300;

        public static final int MICRO_PDF417_ENABLE                  = 0x1500;
        public static final int MICRO_PDF417_LENGTH1                 = 0x1501;
        public static final int MICRO_PDF417_LENGTH2                = 0x1502;
        public static final int MICRO_PDF417_LENGTH_CONTROL         = 0x1503;

        public static final int QR_CODE_ENABLE                      = 0x1600;
        public static final int QR_CODE_LENGTH1                     = 0x1601;
        public static final int QR_CODE_LENGTH2                     = 0x1602;
        public static final int QR_CODE_LENGTH_CONTROL              = 0x1603;

        public static final int AZTEC_ENABLE                        = 0x1700;
        public static final int AZTEC_LENGTH1                       = 0x1701;
        public static final int AZTEC_LENGTH2                       = 0x1702;
        public static final int AZTEC_LENGTH_CONTROL                = 0x1703;
        public static final int AZTEC_RUNES_ENABLE                  = 0x1721;

        public static final int PLANET_ENABLE                       = 0x1800;

        public static final int POSTNET_ENABLE                      = 0x1900;

        public static final int MAILMARK_4_STATE_POSTAL_ENABLE      = 0x1a00;
        public static final int INTELLIGENT_MAIL_BARCODE_ENABLE     = 0x1a21;

        public static final int UK_POSTAL_ENABLE                    = 0x1b00;
        public static final int AUSTRALIAN_POSTAL_ENABLE            = 0x1c00;
        public static final int NETHERLANDS_KIX_CODE_ENABLE         = 0x1d00;
        public static final int JPN_POSTAL_ENABLE                   = 0x1e00;

        public static final int GS1_128_LENGTH1                     = 0x1f01;
        public static final int GS1_128_LENGTH2                     = 0x1f02;
        public static final int GS1_128_LENGTH_CONTROL              = 0x1f03;

        public static final int MICRO_QR_CODE_ENABLE                = 0x2000;

        public static final int KOREAN_POSTAL_AUTHORITY_ENABLE      = 0x2200;
        public static final int KOREAN_POSTAL_AUTHORITY_CD_TRANSMISSION= 0x2221;

        public static final int CODE11_ENABLE                       = 0x2300;
        public static final int CODE11_LENGTH1                      = 0x2301;
        public static final int CODE11_LENGTH2                      = 0x2302;
        public static final int CODE11_LENGTH_CONTROL               = 0x2303;
        public static final int CODE11_CHECK_DIGIT              = 0x2304;
        public static final int CODE11_CD_TRANSMISSION          = 0x2305;

        public static final int CODE_2OF5_CHECK_DIGIT           = 0x3221;
        public static final int CODE_2OF5_CD_TRANSMISSION       = 0x3222;

        public static final int SCODE_ENABLE                    = 0x3300;
        public static final int SCODE_LENGTH1                   = 0x3301;
        public static final int SCODE_LENGTH2                   = 0x3302;
        public static final int SCODE_LENGTH_CONTROL            = 0x3303;

        public static final int CHINESE_POST_MATRIX_2OF5_ENABLE = 0x3400;

        public static final int IATA_ENABLE                     = 0x3500;
        public static final int IATA_LENGTH1                    = 0x3501;
        public static final int IATA_LENGTH2                    = 0x3502;
        public static final int IATA_LENGTH_CONTROL             = 0x3503;
        public static final int IATA_CHECK_DIGIT                = 0x3504;
        public static final int IATA_CD_TRANSMISSION            = 0x3505;

        public static final int TELEPEN_ENABLE                  = 0x3600;
        public static final int TELEPEN_LENGTH1                 = 0x3601;
        public static final int TELEPEN_LENGTH2                 = 0x3602;
        public static final int TELEPEN_LENGTH_CONTROL          = 0x3603;
        public static final int TELEPEN_MODE                    = 0x3604;

        public static final int UK_PLESSEY_ENABLE               = 0x3700;
        public static final int UK_PLESSEY_LENGTH1              = 0x3701;
        public static final int UK_PLESSEY_LENGTH2              = 0x3702;
        public static final int UK_PLESSEY_LENGTH_CONTROL       = 0x3703;
        public static final int UK_PLESSEY_CD_TRANSMISSION      = 0x3704;

        public static final int CODABLOCK_F_ENABLE              = 0x3800;
        public static final int CODABLOCK_F_LENGTH1             = 0x3801;
        public static final int CODABLOCK_F_LENGTH2             = 0x3802;
        public static final int CODABLOCK_F_LENGTH_CONTROL      = 0x3803;

        public static final int COMPOSITE_GS1_ENABLE            = 0x3900;
        public static final int COMPOSITE_LENGTH1               = 0x3901;
```

```
    public static final int COMPOSITE_LENGTH2                   = 0x3902;
    public static final int COMPOSITE_LENGTH_CONTROL            = 0x3903;
    public static final int COMPOSITE_EAN_UPA_ENABLE            = 0x3904;
    public static final int COMPOSITE_LINK_FLAG                 = 0x3905;
    public static final int COMPOSITE_OUTPUT_MODE               = 0x3906;

    public static final int CHINESE_SENSIBLE_CODE_ENABLE        = 0x3a00;
    public static final int CHINESE_SENSIBLE_CODE_LENGTH1       = 0x3a01;
    public static final int CHINESE_SENSIBLE_CODE_LENGTH2       = 0x3a02;
    public static final int CHINESE_SENSIBLE_CODE_LENGTH_CONTROL= 0x3a03;

    public static final int OCR_PASSPORTS                       = 0x3b01;
    public static final int OCR_VISAS_A                         = 0x3b02;
    public static final int OCR_VISAS_B                         = 0x3b03;
    public static final int OCR_TRAVEL_DOCUMENTS_1              = 0x3b04;
    public static final int OCR_TRAVEL_DOCUMENTS_2              = 0x3b05;
    public static final int OCR_ISBN                            = 0x3b06;
    public static final int OCR_JAPAN_BOOK_PRICE                = 0x3b07;
    public static final int OCR_JAPANESE_DRIVING_LICENSE        = 0x3b08;
    public static final int OCR_JAPANESE_PRIVATE_NUMBER         = 0x3b09;

    public static final int OCR_FREE_LINE                       = 0x3c00;
    public static final int OCR_FREE_LINE_LENGTH1               = 0x3c01;
    public static final int OCR_FREE_LINE_LENGTH2               = 0x3c02;
    public static final int OCR_FREE_LINE_LENGTH_CONTROL        = 0x3c03;
    public static final int OCR_FREE_LINE_FONT_TYPE             = 0x3c04;

    public static final int DOTCODE_ENABLE                      = 0x3f00;
    public static final int DOTCODE_LENGTH1                     = 0x3f01;
    public static final int DOTCODE_LENGTH2                     = 0x3f02;
    public static final int DOTCODE_LENGTH_CONTROL              = 0x3f03;

}
```

PropertyID	**Description**
SCANKEY_ENABLE	Enable the scankey to scan. enable(1), disable(0)
INPUTKEY_ENABLE	Scan result change to input event. enable(1), disable(0)
INPUTKEY_TYPE	The type of input event.

	Clipboard(0), Input Event(1) 0. is use Input Method Service to send data, like copy and paste. 1. is use Input keyevent to send data.
GOODREAD_ENABLE	Enable Good read function. enable(1), disable(0)
GOODREAD_LED_ENABLE	Good read function, LED will affect between scan. enable(1), disable(0)
GOODREAD_VIBRATOR_ENABLE	Good read function, vibrator will run after decode success. enable(1), disable(0)
GOODREAD_AUDIO_ENABLE	Good read function, it will beep after decode success. enable(1), disable(0)
GOODREAD_BEEPPER_FRQ	Beeper audio frequency. 0~4000 Hz
GOODREAD_BEEPPER_DURATION	Beeper Duration of Good read. 20~5000 ms
GOODREAD_AIMING	Scan success will have aiming signal.
FORMATING_PREFIX	Prefix of the decode result. Special ASCII symbol: [SOH], [STX], [ETX], [EOT], [ENQ], [ACK], [BEL], [BS], [HT], [LF], [VT], [FF], [CR], [SO], [SI], [DLE], [DC1], [DC2], [DC3], [DC4], [NAK], [SYN], [ETB], [CAN], [EM], [SUB], [ESC], [FS], [GS], [RS], [US], [DEL]
FORMATING_SUFFIX	Suffix of the decode result. Special ASCII symbol: [SOH], [STX], [ETX], [EOT], [ENQ], [ACK], [BEL], [BS], [HT], [LF], [VT], [FF], [CR], [SO], [SI], [DLE], [DC1], [DC2], [DC3], [DC4], [NAK], [SYN], [ETB], [CAN], [EM], [SUB], [ESC], [FS], [GS], [RS], [US], [DEL]
FORMATING_REMOVE_NONPRINT	Remove the-non print words of decode result. enable(1), disable(0)

FORMATING_NONPRINT_READABLE	Change the non-print word to readable words. enable(1), disable(0)
FORMATING_GS_REPLACE	Replace GS word for the word you want. Special ASCII symbol: [SOH], [STX], [ETX], [EOT], [ENQ], [ACK], [BEL], [BS], [HT], [LF], [VT], [FF], [CR], [SO], [SI], [DLE], [DC1], [DC2], [DC3], [DC4], [NAK], [SYN], [ETB], [CAN], [EM], [SUB], [ESC], [FS], [GS], [RS], [US], [DEL]
FORMATING_OPTICON_ID	Add Opticon ID before decode result. enable(1), disable(0)
FORMATING_AIM_ID	Add AIM ID before decode result. enable(1), disable(0)
FORMATING_STRING_ENODING_SET	Change string encording set. "Windows Latin(Cp1252)", "UTF8(UTF8)", "Big5, Traditional Chinese(Big5)", "GBK, Simplified Chinese(GBK)", "Shift-JIS, Japanese(Shift_JIS)"
WEDGE_INTENT_ENABLE	Enable the intent to send the decode result. enable(1), disable(0)
WEDGE_INTENT_DELIVERY_MODE	Three mode, Send Intent via StartActivity = 0. Send Intent via startService = 1 Broadcast intent = 2
WEDGE_INTENT_ACTION	Intent action
WEDGE_INTENT_CATEGORY	Intent category
WEDGE_INTENT_BARCODETYPE	Intent extra data to send decode type
WEDGE_INTENT_BARCODEDATA	Intent extra data to send decode data
WEDGE_INTENT_PACKAGE_NAME	Intent package name
GOODREAD_LED_DURATION	Led Duration.
GOODREAD_LED_INTERVAL	Interval of led.
GOODREAD_LED_TIMES	How many times to bright led.
GOODREAD_LED_COLOR	Set led color of good read, from 0 to 0xFFFFFF, 0xFF0000 is red color, 0x00FF00 is green color, 0x0000FF is blue color.
READTIMEOUT_LED_ENABLE	Enable Led when read timeout.
READTIMEOUT_LED_DURATION	Read timeout led duration.

READTIMEOUT_LED_INTERVAL	Read timeout led interval.
READTIMEOUT_LED_TIMES	Read timeout led times.
READTIMEOUT_LED_COLOR	Like GOODREAD_LED_COLOR, just for read timeout.
GOODREAD_VIBRATOR_DURATION	Vibrator duration of goodread.
GOODREAD_VIBRATOR_INTERVAL	Vibrator interval of goodread.
GOODREAD_VIBRATOR_TIMES	Vibrator times of goodread.
READTIMEOUT_VIBRATOR_ENABLE	Enable vibrator when read timeout.
READTIMEOUT_VIBRATOR_DURATION	Vibrator duration of read timeout.
READTIMEOUT_VIBRATOR_INTERVAL	Vibrator interval of read timeout.
READTIMEOUT_VIBRATOR_TIMES	Vibrator times of read timeout.
GOODREAD_BEEPPER_INTERVAL	Beeper interval of good read.
GOODREAD_BEEPPER_TIMES	Beeper times of good read.
READTIMEOUT_AUDIO_ENABLE	Enable beeper when read timeout
READTIMEOUT_BEEPPER_FRQ	Beeper frequency of read timeout.
READTIMEOUT_BEEPPER_DURATION	Beeper duration of read timeout.
READTIMEOUT_BEEPPER_INTERVAL	Beeper interval of read timeout.
READTIMEOUT_BEEPPER_TIMES	Beeper times of read timeout.
READ_MODE	READMODE_SINGLE 0 READMODE_MULTIPLE 1 READMODE_CONTINUOUS 2
READ_TIME	Definition of Read time from 0~9 second, and infinity time.
MARGIN_CHECK	From non-check, 1/7, 2/7, 3/7, 4/7, 5/7, 6/7, normal check. No check(0), 1/7(1) …….. 6/7(6), normal(7)
REDUNDANCY	From Read 1 time to Read 9 times. Read 1 time, redundancy = 0 Read 2 times, redundancy = 1 Read 3 times, redundancy = 2 Read 4 times, redundancy = 3(default) Read 5 times, redundancy = 4 Read 6 times, redundancy = 5 Read 7 times, redundancy = 6 Read 8 times, redundancy = 7 Read 9 times, redundancy = 8
NEGATIVE_BARCODE	Positive and negative bar codes. Positive(1), Negative(2), Both(3)

TRIGGER_REPEAT	Enable trigger repeat function. enable(1), disable(0)
DECODE_COMMAND	Send Command to the decode library.
CENTRAL_READING	Enable Central reading function. enable(1), disable(0)
CENTRAL_READING_AREA	Set the central reading area 0: regular area, 1: small area, 2: large area
MAX_UNIQUE_READ	Make number of reading variable. (default:20) Range of property ID is "1 to 200".
MULTIPLE_READ_RESET_TIME	Rest time for Multiple read inhibition. Range of property ID is "1 to 999".
AIMER_MODE	AIMER ON/OFF enable(1), disable(0)
CAMERA_ILLUMINATION_MODE	CAMERA_LED_MODE_ENABLE 1 CAMERA_LED_MODE_DISABLE 2 CAMERA_LED_MODE_ALTERNATIV 3 CAMERA_LED_MODE_MIRROR_REFLECTION 4
DATA_EDIT_PROGRAMMINIG_INDEX	Type of Data edit.
DATA_EDIT_PROGRAMMINIG_CUT	What string need to be cut.
DATA_EDIT_PROGRAMMINIG_PASTE	What string need to be pasted.
CODE39_ENABLE	Enable Code39 enable(1), disable(0)
CODE39_LENGTH1	Code39 Length1
CODE39_LENGTH2	Code39 Length2
CODE39_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3
CODE39_MODE	NORMAL_CODE39 1 CODE39_FULLASCII 2 CODE39_FULLASCII_IF_POSSIBLE 3 ITALIAN_PHARM_ONLY 4 ITALIAN_PHARM_IF_POSSIBLE 5
CODE39_CONCATENATION_ENABLE	Enable concatenation. enable(1), disable(0)
CODE39_STSP_TRANSMISSION	Send Start and stop charactor. enable(1), disable(0)

CODE39_CHECK_DIGIT	Enable Code39 check digit enable(1), disable(0)
CODE39_CD_TRANSMISSION	Send Code39 check digit enable(1), disable(0)
INDUSTRIAL_2OF5_ENABLE	Enable Industrial 2 of 5 enable(1), disable(0)
INDUSTRIAL_2OF5_LENGTH1	Industrial 2 of 5 Length1
INDUSTRIAL_2OF5_LENGTH2	Industrial 2 of 5 Length2
INDUSTRIAL_2OF5_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3
MATRIX_2OF5_ENABLE	Enable Matrix 2 of 5 enable(1), disable(0)
MATRIX_2OF5_LENGTH1	Matrix 2 of 5 Length1
MATRIX_2OF5_LENGTH2	Matrix 2 of 5 Length2
MATRIX_2OF5_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3
INTERLEAVED_2OF5_ENABLE	Enable Interleaved 2 of 5 enable(1), disable(0)
INTERLEAVED_2OF5_LENGTH1	Interleaved 2 of 5 Length1
INTERLEAVED_2OF5_LENGTH2	Interleaved 2 of 5 Length1
INTERLEAVED_2OF5_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3
CODABAR_ENABLE	Enable Codabar enable(1), disable(0)
CODABAR_LENGTH1	Codabar Length1
CODABAR_LENGTH2	Codabar Length2
CODABAR_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1,

	Two fixed = 2, Range = 3
CODABAR_MODE	CODABAR_NORMAL 0 ABC_CODE_ONLY 1 CX_CODE_ONLY 2 CODABAR_ABC_CX 3
CODABAR_SPACE_INSERTION	This option inserts spaces in position 2, 7, 13, of the data string for use in library systems. enable(1), disable(0)
CODABAR_CHECK_DIGIT	Enable Codabar check digit enable(1), disable(0)
CODABAR_CD_TRANSMISSION	Send Codabar check digit enable(1), disable(0)
CODABAR_STSP_TRANSMISSION	NOT_TRANSMIT_STSP 0 STSP_ABCD 1 STSP_abcd 2 STSP_ABCD_TN_E 3 STSP_abcd_tn_e 4 STSP_DC1DC2DC3DC4 5
CODE93_ENABLE	Enable Code93 enable(1), disable(0)
CODE93_LENGTH1	Code93 Length1
CODE93_LENGTH2	Code93 Length2
CODE93_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3
CODE93_CHECK_DIGIT	Enable Code93 check digit. enable(1), disable(0)
CODE128_ENABLE	Enable Code128 enable(1), disable(0)
CODE128_LENGTH1	Code128 Length1
CODE128_LENGTH2	Code128 Length2
CODE128_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3
CODE128_CONCATENATION_ENABLE	Enable Code128 concatenation.

	enable(1), disable(0)
UPCA_ENABLE	Enable UPCA enable(1), disable(0)
UPCA_LEADING_ZERO	Send leading zero. enable(1), disable(0)
UPCA_CD_TRANSMISSION	Send check digit. enable(1), disable(0)
UPCA_EXT_ENABLE_2_DIGIT	Enable UPCA add-on 2. enable(1), disable(0)
UPCA_EXT_ENABLE_5_DIGIT	Enable UPCA add-on 5. enable(1), disable(0)
UPCA_EXT_ENABLE_EXT_DIGIT	Only decode when UPCA has add-on. enable(1), disable(0)
UPCE_ENABLE	Enable UPCE enable(1), disable(0)
UPCE_LEADING_ZERO	Send leading zero enable(1), disable(0)
UPCE_CD_TRANSMISSION	Send check digit. enable(1), disable(0)
UPCE_EXT_ENABLE_2_DIGIT	Enable UPCE add-on 2 enable(1), disable(0)
UPCE_EXT_ENABLE_5_DIGIT	Enable UPCE add-on 5 enable(1), disable(0)
UPCE_EXT_ENABLE_EXT_DIGIT	Only decode when UPCE has add-on. enable(1), disable(0)
EAN13_ENABLE	Enable Ean13 enable(1), disable(0)
EAN13_CD_TRANSMISSION	Send check digit enable(1), disable(0)
EAN13_EXT_ENABLE_2_DIGIT	Enable Ean13 add-on 2 enable(1), disable(0)
EAN13_EXT_ENABLE_5_DIGIT	Enable Ean13 add-on 5 enable(1), disable(0)
EAN13_EXT_ENABLE_EXT_DIGIT	Only decode when Ean13 has add-on. enable(1), disable(0)

EAN8_ENABLE	Enable Ean8 enable(1), disable(0)
EAN8_CD_TRANSMISSION	Send check digit enable(1), disable(0)
EAN8_EXT_ENABLE_2_DIGIT	Enable Ean8 add-on 2 enable(1), disable(0)
EAN8_EXT_ENABLE_5_DIGIT	Enable Ean8 add-on 5 enable(1), disable(0)
EAN8_EXT_ENABLE_EXT_DIGIT	Only decode when Ean8 has add-on. enable(1), disable(0)
MSI_PLESSEY_ENABLE	Enable MSI/Plessey enable(1), disable(0)
MSI_PLESSEY_LENGTH1	MSI/Plessey Length1
MSI_PLESSEY_LENGTH2	MSI/Plessey Length2
MSI_PLESSEY_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3
MSI_PLESSEY_CHECK_DIGIT	NOT_CHECK_DIGIT 0 MSI_PLESSEY_1CD_MOD10 1 MSI_PLESSEY_MOD10_MOD10 2 MSI_PLESSEY_MOD10_MOD11 3 MSI_PLESSEY_MOD11_MOD10 4 MSI_PLESSEY_MOD11_MOD11 5
MSI_PLESSEY_CD_TRANSMISSION	NOT_TRANSMIT_CD 0 TRANSMIT_CD1 1 TRANSMIT_CD1_CD2 2
GS1_DATABAR_ENABLE	Enable GS1 DataBar enable(1), disable(0)
GS1_DATABAR_LENGTH1	GS1 DataBar Length1
GS1_DATABAR_LENGTH2	GS1 DataBar Length2
GS1_DATABAR_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3

GS1_DATABAR_AI_TRANSMISSION	Transmission application ID enable(1), disable(0)
GS1_DATABAR_CD_TRANSMISSION	Transmission check digit enable(1), disable(0)
GS1_DATABAR_LIMITED_ENABLE	Enable GS1 DataBar Limited enable(1), disable(0)
GS1_DATABAR_EXPANDED_ENABLE	Enable GS1 DataBar Expaned enable(1), disable(0)
PDF417_ENABLE	Enable PDF417 enable(1), disable(0)
PDF417_LENGTH1	PDF417 Length1
PDF417_LENGTH2	PDF417 Length2
PDF417_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3
DATAMATRIX_ECC200_ENABLE	Enable Datamatrix ECC200 enable(1), disable(0)
DATAMATRIX_LENGTH1	Datamatrix Length1
DATAMATRIX_LENGTH2	Datamatrix Length2
DATAMATRIX_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3
MAXICODE_ENABLE	Enable Maxicode enable(1), disable(0)
MAXICODE_LENGTH1	Maxicode Length1
MAXICODE_LENGTH2	Maxicode Length2
MAXICODE_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3

TRIOPTIC_ENABLE	Enable Trioptic enable(1), disable(0)
MICRO_PDF417_ENABLE	Enable Micropdf417 enable(1), disable(0)
MICRO_PDF417_LENGTH1	Micropdf417 Length1
MICRO_PDF417_LENGTH2	Micropdf417 Length2
MICRO_PDF417_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3
QR_CODE_ENABLE	Enable Qrcode enable(1), disable(0)
QR_CODE_LENGTH1	Qrcode Length1
QR_CODE_LENGTH2	Qrcode Length2
QR_CODE_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3
AZTEC_ENABLE	Enable Aztec enable(1), disable(0)
AZTEC_LENGTH1	Aztec Length1
AZTEC_LENGTH2	Aztec Length2
AZTEC_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3
AZTEC_RUNES_ENABLE	Enable Aztec Runes enable(1), disable(0)
PLANET_ENABLE	Enable Postal Planet enable(1), disable(0)
POSTNET_ENABLE	Enable Postal Postnet enable(1), disable(0)

MAILMARK_4_STATE_POSTAL_ENABLE	Enable Mailmark 4state postal enable(1), disable(0)
INTELLIGENT_MAIL_BARCODE_ENABLE	Enable Intelligent Mail enable(1), disable(0)
UK_POSTAL_ENABLE	Enable UK postal Mail enable(1), disable(0)
AUSTRALIAN_POSTAL_ENABLE	Enable Australia Post enable(1), disable(0)
NETHERLANDS_KIX_CODE_ENABLE	Enable KIX-code enable(1), disable(0)
JPN_POSTAL_ENABLE	Enable Japan Post enable(1), disable(0)
GS1_128_LENGTH1	GS1-128 Length1
GS1_128_LENGTH2	GS1-128 Length2
GS1_128_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3
MICRO_QR_CODE_ENABLE	Enable MicroQR code enable(1), disable(0)
KOREAN_POSTAL_AUTHORITY_ENABLE	Enable Korean postal enable(1), disable(0)
KOREAN_POSTAL_AUTHORITY_CD_TRANSMISSION	Send check digit enable(1), disable(0)
CODE11_ENABLE	Enable Code11 enable(1), disable(0)
CODE11_LENGTH1	Code11 Length1
CODE11_LENGTH2	Code11 Length2
CODE11_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3
CODE11_CHECK_DIGIT	NOT_CHECK_DIGIT 0 CODE11_CHECK_1CD 1 CODE11_CHECK_2CD 2

	CHECK_DIGIT_AUTO 3
CODE11_CD_TRANSMISSION	Send check digit enable(1), disable(0)
CODE_2OF5_CHECK_DIGIT	Enable all 2 of 5 barcode check digit enable(1), disable(0)
CODE_2OF5_CD_TRANSMISSION	Send all 2 of 5 barcode check digit enable(1), disable(0)
SCODE_ENABLE	Enalbe Scode enable(1), disable(0)
SCODE_LENGTH1	Scode Length1
SCODE_LENGTH2	Scode Length2
SCODE_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3
CHINESE_POST_MATRIX_2OF5_ENABLE	Enable Chinese post matrix 2 of 5 enable(1), disable(0)
IATA_ENABLE	Enable IATA enable(1), disable(0)
IATA_LENGTH1	IATA Length1
IATA_LENGTH2	IATA Length2
IATA_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3
IATA_CHECK_DIGIT	NOT_CHECK_DIGIT 0 IATA_CHECK_FC_SN_ONLY 1 IATA_CHECK_CPN_FC_SN 2 IATA_CHECK_CPN_AC_FC_SN 3
IATA_CD_TRANSMISSION	Send check digit enable(1), disable(0)
TELEPEN_ENABLE	Enable Telepen enable(1), disable(0)
TELEPEN_LENGTH1	Telepen Length1

TELEPEN_LENGTH2	Telepen Length2
TELEPEN_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3
TELEPEN_MODE	TELEPEN_MODE_NUMERIC 0 TELEPEN_MODE_ASCII 1
UK_PLESSEY_ENABLE	Enable UK/Plessey enable(1), disable(0)
UK_PLESSEY_LENGTH1	UK/Plessey Length1
UK_PLESSEY_LENGTH2	UK/Plessey Length2
UK_PLESSEY_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3
UK_PLESSEY_CD_TRANSMISSION	Send check digit enable(1), disable(0)
CODABLOCK_F_ENABLE	Enable Codablock F enable(1), disable(0)
CODABLOCK_F_LENGTH1	Codablock F Length1
CODABLOCK_F_LENGTH2	Codablock F Length2
CODABLOCK_F_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3
COMPOSITE_GS1_ENABLE	Enable GS1 Composite enable(1), disable(0)
COMPOSITE_LENGTH1	Composite Length1
COMPOSITE_LENGTH2	Composite Length2
COMPOSITE_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3
COMPOSITE_EAN_UPA_ENABLE	Enable Ean/UPA Composite enable(1), disable(0)
COMPOSITE_LINK_FLAG	Enable Link flag of Composite

	enable(1), disable(0)
COMPOSITE_OUTPUT_MODE	OUTPUT_MODE_1D 0 OUTPUT_MODE_2D 1 OUTPUT_MODE_1D2D 2
CHINESE_SENSIBLE_CODE_ENABLE	Enable Chinese sensible code enable(1), disable(0)
CHINESE_SENSIBLE_CODE_LENGTH1	Chinese sensible code Length1
CHINESE_SENSIBLE_CODE_LENGTH2	Chinese sensible code Length2
CHINESE_SENSIBLE_CODE_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3
OCR_PASSPORTS	Enable OCR PASSPORTS enable(1), disable(0)
OCR_VISAS_A	Enable OCR VISAS A enable(1), disable(0)
OCR_VISAS_B	Enable OCR VISAS B enable(1), disable(0)
OCR_TRAVEL_DOCUMENTS_1	Enable OCR Travel documents 1 enable(1), disable(0)
OCR_TRAVEL_DOCUMENTS_2	Enable OCR Travel documents 2 enable(1), disable(0)
OCR_ISBN	Enable OCR ISBN enable(1), disable(0)
OCR_JAPAN_BOOK_PRICE	Enable OCR Japan book price enable(1), disable(0)
OCR_JAPANESE_DRIVING_LICENSE	Enable OCR Japanese driving license enable(1), disable(0)
OCR_JAPANESE_PRIVATE_NUMBER	Enable OCR Japanese private number enable(1), disable(0)
OCR_FREE_LINE	Enable OCR Free Line. enable(1), disable(0)
OCR_FREE_LINE_LENGTH1	OCR Free Line Length1
OCR_FREE_LINE_LENGTH2	OCR Free Line Length2
OCR_FREE_LINE_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1,

	Two fixed = 2, Range = 3
OCR_FREE_LINE_FONT_TYPE	Font of free line. OCR_B(0), OCR_A(1), MS_GHOTHIC(2), MS_MINCHO(3), MICR_E13B(4)
DOTCODE_ENABLE	Enable Dot code.
DOTCODE_LENGTH1	Dot code Length1
DOTCODE_LENGTH2	Dot code Length2
DOTCODE_LENGTH_CONTROL	Control Length1 and Length2 for no check, one fixed, two fixed or Range. No check = 0, One fixed = 1, Two fixed = 2, Range = 3

3.7 CodeID

This section will introduce the code type.

```
public static final int POSTAL_PLANET         = 24;
public static final int POSTAL_4STATE         = 26;
public static final int POSTAL_ROYALMAIL      = 27;
public static final int POSTAL_AUSTRALIAN     = 28;
public static final int POSTAL_KIX            = 29;
public static final int POSTAL_JAPAN          = 30;
public static final int UPC_A                 = 64;
public static final int UPC_A_ADD2            = 65;
public static final int UPC_A_ADD5            = 66;
public static final int UPC_E                 = 67;
public static final int UPC_E_ADD2            = 68;
public static final int UPC_E_ADD5            = 69;
public static final int EAN_13                = 70;
public static final int EAN_13_ADD2           = 71;
public static final int EAN_13_ADD5           = 72;
public static final int EAN_8                 = 73;
public static final int EAN_8_ADD2            = 74;
public static final int EAN_8_ADD5            = 75;
public static final int Code39                = 76;
public static final int Code39_Full_ASCII     = 77;
public static final int Italian_Pharmaceutical = 78;
public static final int Codabar               = 79;
public static final int Codabar_ABC           = 80;
public static final int Codabar_CX            = 81;
public static final int Industrial_2of5       = 82;
public static final int Interleaved_2of5      = 83;
public static final int S_Code                = 84;
public static final int Matrix_2of5           = 85;
public static final int CHinese_Post          = 86;
public static final int IATA                  = 87;
public static final int MSI_Plessey           = 88;
public static final int Telepen               = 89;
public static final int UK_Plessey            = 90;
public static final int Code_128              = 91;
public static final int Code_93               = 92;
public static final int Code_11               = 93;
public static final int Korean_Postal          = 94;
public static final int Intelligent_Mail       = 95;
public static final int POSTNET                = 96;
public static final int GS1_Databar            = 97;
public static final int CC_A                   = 98;
public static final int CC_B                   = 99;
public static final int CC_C                   = 100;
public static final int Codablock_F            = 101;
public static final int DataMatrix             = 102;
public static final int Aztec                  = 103;
public static final int Chinese_Sensible       = 104;
public static final int QR_code                = 105;
public static final int Micro_QR_Code         = 106;
public static final int Maxi_Code              = 107;
public static final int MicroPDF417            = 108;
public static final int OCR                    = 109;
```

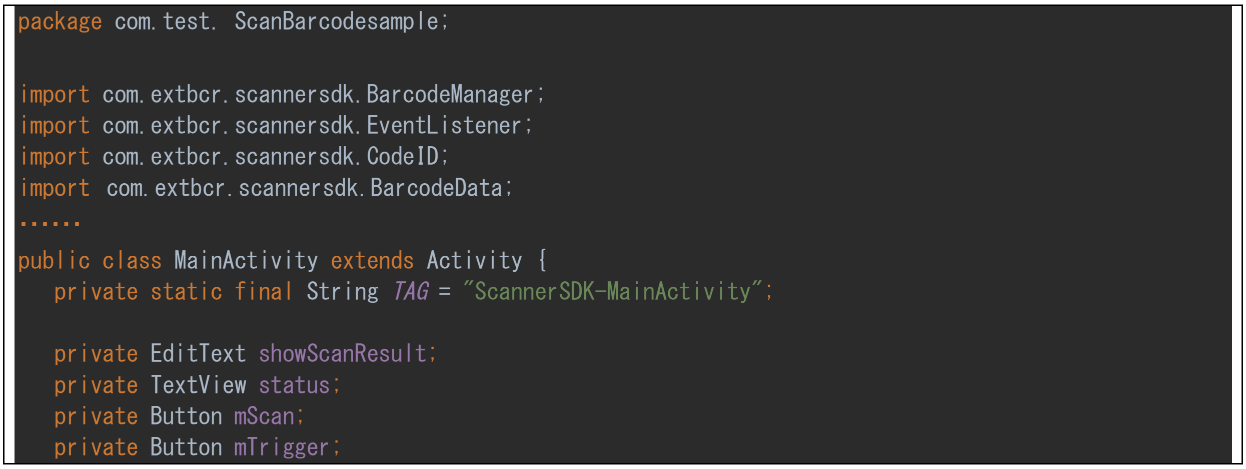
4 Sample Code

4.1 Barcode Scan App

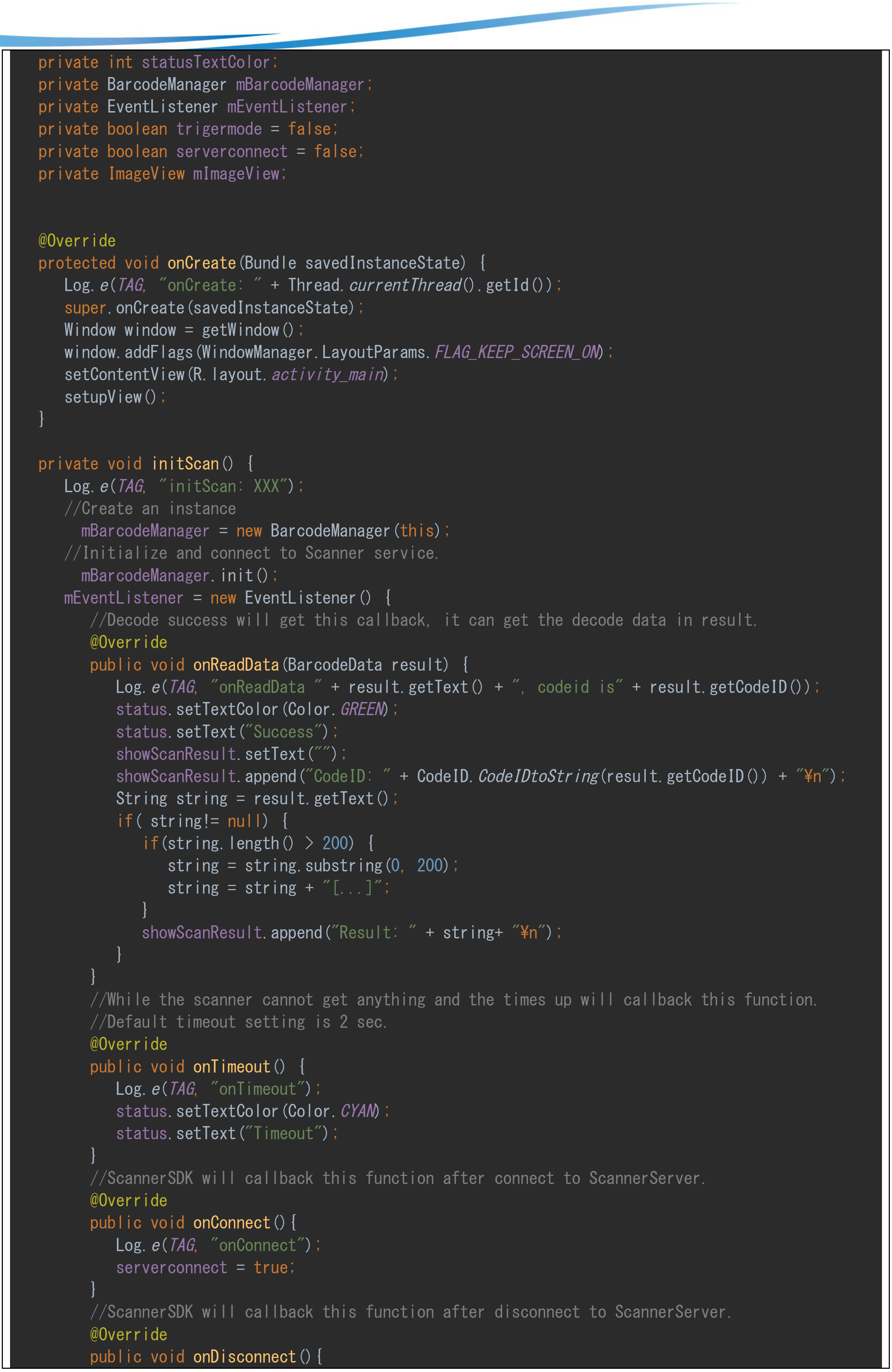
SampleSDK_H33.zip

Go to www.opticon.com/H-33 to download the zip file.

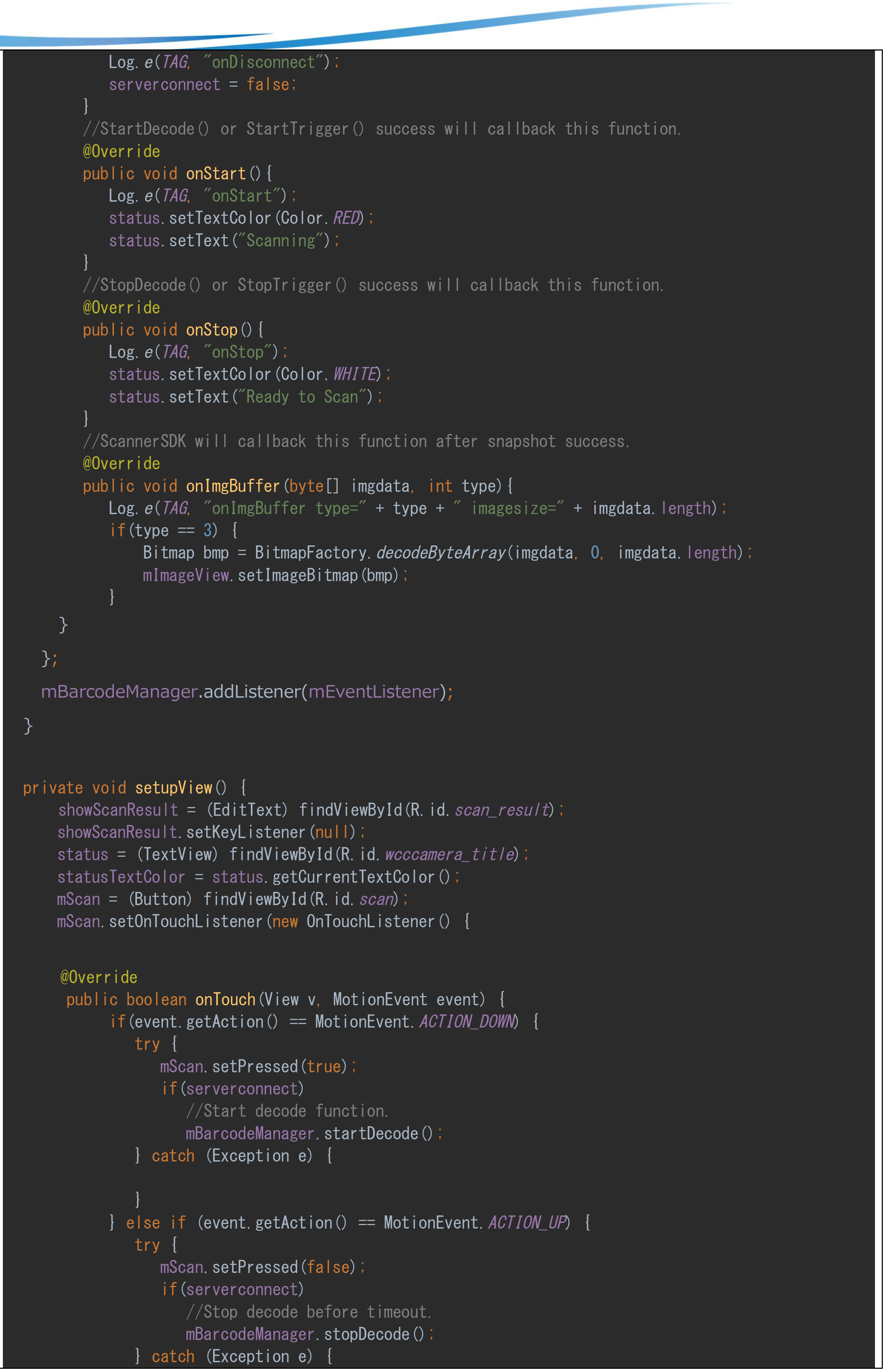
4.2 ScanBarcode sample



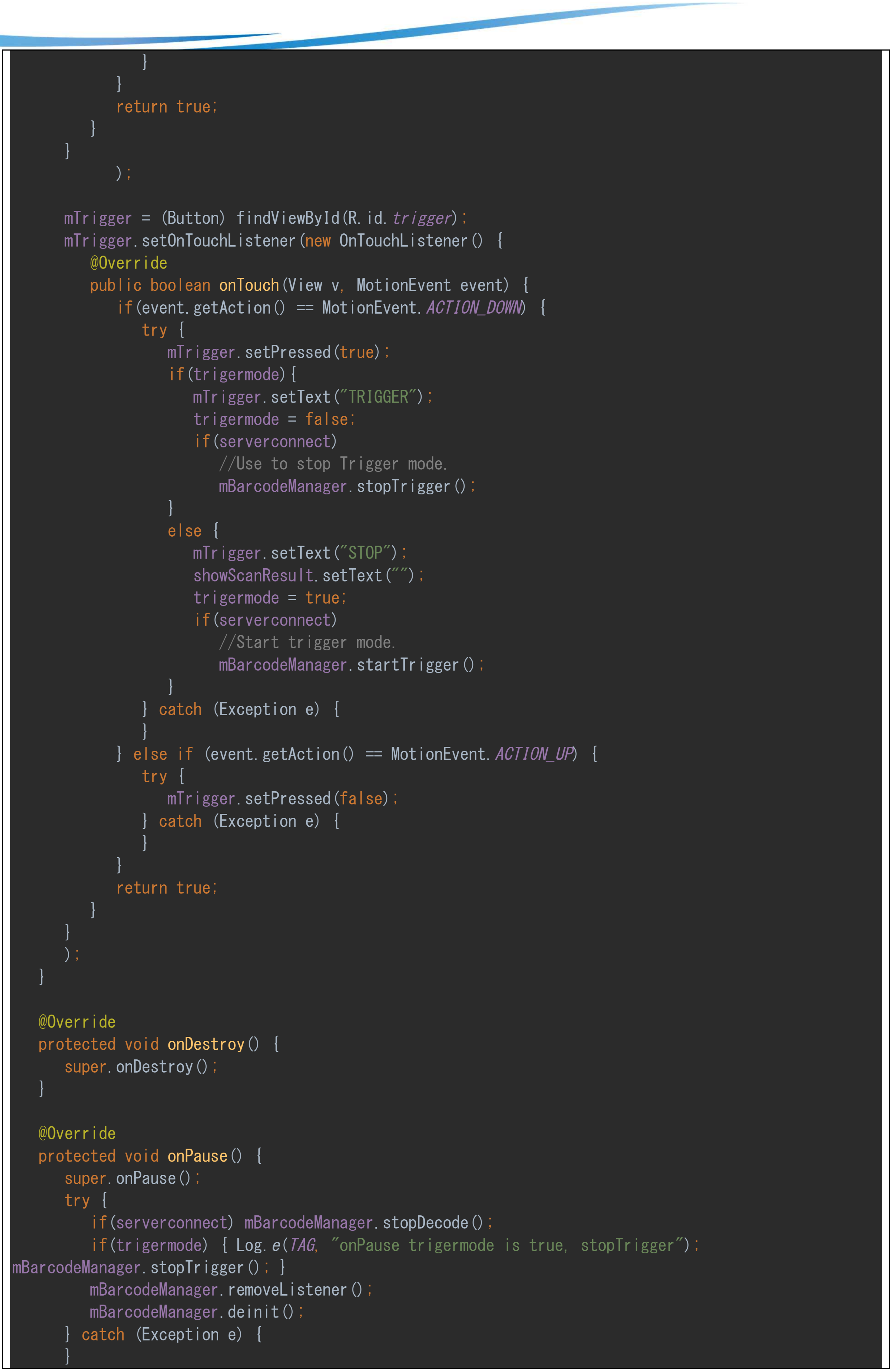
```
package com.test. ScanBarcodesample;

import com.extbcr.scannersdk.BarcodeManager;
import com.extbcr.scannersdk.EventListener;
import com.extbcr.scannersdk.CodeID;
import  com.extbcr.scannersdk.BarcodeData;
……
public class MainActivity extends Activity {
   private static final String TAG = ”ScannerSDK-MainActivity”;

   private EditText showScanResult;
   private TextView status;
   private Button mScan;
   private Button mTrigger;
```



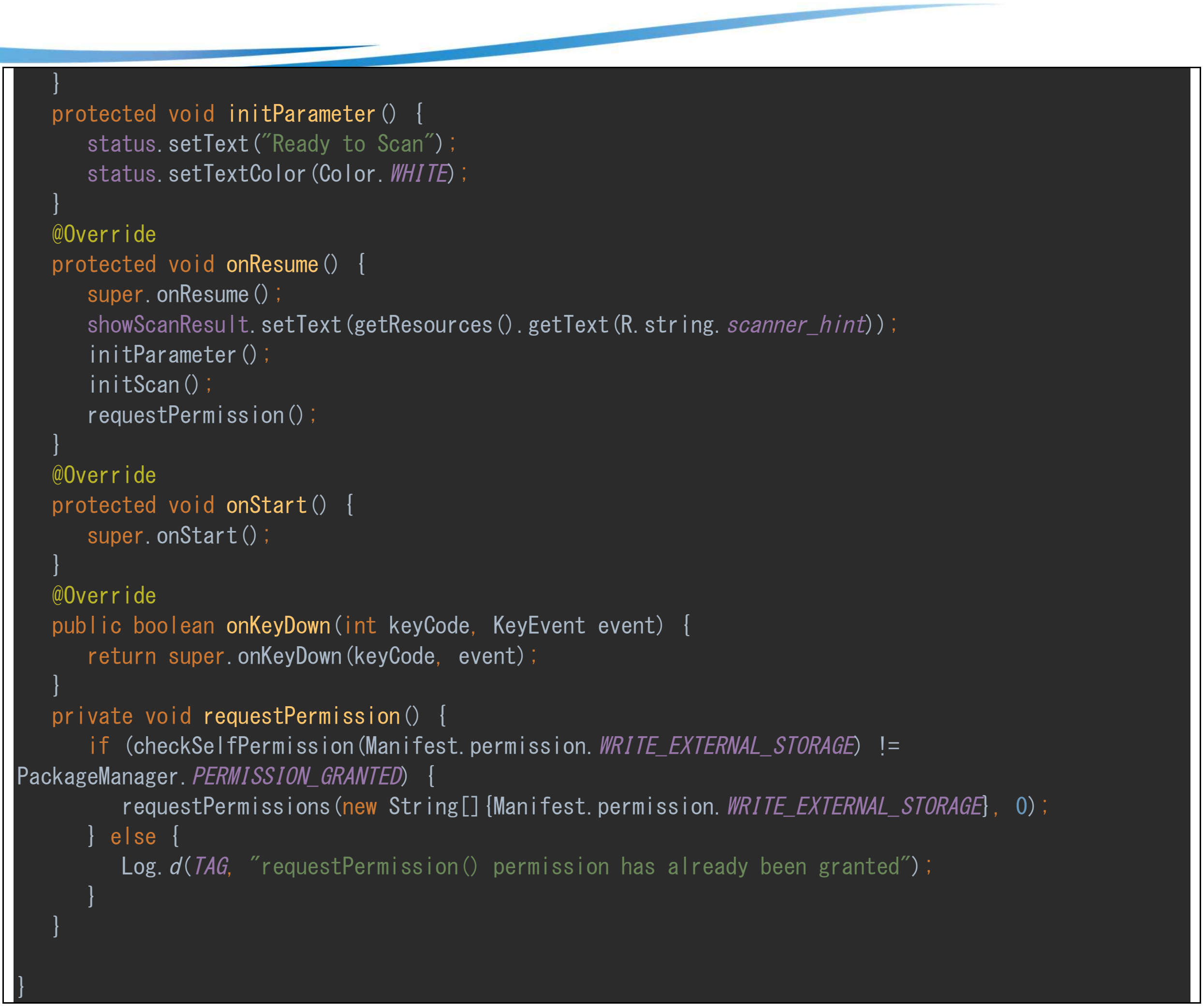
```
    private int statusTextColor;
    private BarcodeManager mBarcodeManager;
    private EventListener mEventListener;
    private boolean trigermode = false;
    private boolean serverconnect = false;
    private ImageView mImageView;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        Log.e(TAG, "onCreate: " + Thread.currentThread().getId());
        super.onCreate(savedInstanceState);
        Window window = getWindow();
        window.addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
        setContentView(R.layout.activity_main);
        setupView();
    }

    private void initScan() {
        Log.e(TAG, "initScan: XXX");
        //Create an instance
          mBarcodeManager = new BarcodeManager(this);
        //Initialize and connect to Scanner service.
          mBarcodeManager.init();
        mEventListener = new EventListener() {
            //Decode success will get this callback, it can get the decode data in result.
            @Override
            public void onReadData(BarcodeData result) {
                Log.e(TAG, "onReadData " + result.getText() + ", codeid is" + result.getCodeID());
                status.setTextColor(Color.GREEN);
                status.setText("Success");
                showScanResult.setText("");
                showScanResult.append("CodeID: " + CodeID.CodeIDtoString(result.getCodeID()) + "¥n");
                String string = result.getText();
                if( string!= null) {
                    if(string.length() > 200) {
                        string = string.substring(0, 200);
                        string = string + "[...]";
                    }
                    showScanResult.append("Result: " + string+ "¥n");
                }
            }
            //While the scanner cannot get anything and the times up will callback this function.
            //Default timeout setting is 2 sec.
            @Override
            public void onTimeout() {
                Log.e(TAG, "onTimeout");
                status.setTextColor(Color.CYAN);
                status.setText("Timeout");
            }
            //ScannerSDK will callback this function after connect to ScannerServer.
            @Override
            public void onConnect(){
                Log.e(TAG, "onConnect");
                serverconnect = true;
            }
            //ScannerSDK will callback this function after disconnect to ScannerServer.
            @Override
            public void onDisconnect(){
```



```
            Log.e(TAG, "onDisconnect");
            serverconnect = false;
        }
        //StartDecode() or StartTrigger() success will callback this function.
        @Override
        public void onStart(){
            Log.e(TAG, "onStart");
            status.setTextColor(Color.RED);
            status.setText("Scanning");
        }
        //StopDecode() or StopTrigger() success will callback this function.
        @Override
        public void onStop(){
            Log.e(TAG, "onStop");
            status.setTextColor(Color.WHITE);
            status.setText("Ready to Scan");
        }
        //ScannerSDK will callback this function after snapshot success.
        @Override
        public void onImgBuffer(byte[] imgdata, int type){
            Log.e(TAG, "onImgBuffer type=" + type + " imagesize=" + imgdata.length);
            if(type == 3) {
                Bitmap bmp = BitmapFactory.decodeByteArray(imgdata, 0, imgdata.length);
                mImageView.setImageBitmap(bmp);
            }
      }
  };
  mBarcodeManager.addListener(mEventListener);
}


private void setupView() {
    showScanResult = (EditText) findViewById(R.id.scan_result);
    showScanResult.setKeyListener(null);
    status = (TextView) findViewById(R.id.wcccamera_title);
    statusTextColor = status.getCurrentTextColor();
    mScan = (Button) findViewById(R.id.scan);
    mScan.setOnTouchListener(new OnTouchListener() {


    @Override
     public boolean onTouch(View v, MotionEvent event) {
        if(event.getAction() == MotionEvent.ACTION_DOWN) {
            try {
               mScan.setPressed(true);
               if(serverconnect)
                  //Start decode function.
                  mBarcodeManager.startDecode();
            } catch (Exception e) {

            }
        } else if (event.getAction() == MotionEvent.ACTION_UP) {
            try {
               mScan.setPressed(false);
               if(serverconnect)
                  //Stop decode before timeout.
                  mBarcodeManager.stopDecode();
            } catch (Exception e) {
```

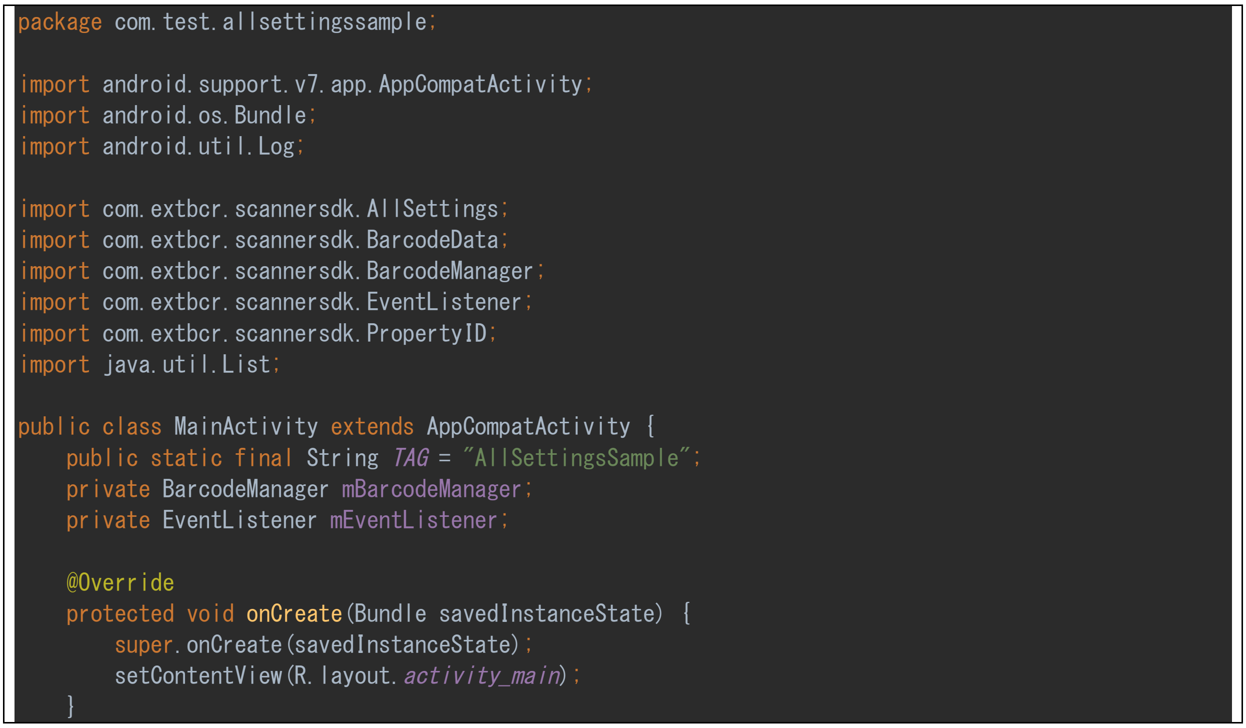


```
                    }
                }
                return true;
            }
        }
                );

        mTrigger = (Button) findViewById(R.id.trigger);
        mTrigger.setOnTouchListener(new OnTouchListener() {
            @Override
            public boolean onTouch(View v, MotionEvent event) {
                if(event.getAction() == MotionEvent.ACTION_DOWN) {
                    try {
                        mTrigger.setPressed(true);
                        if(trigermode){
                            mTrigger.setText(“TRIGGER”);
                            trigermode = false;
                            if(serverconnect)
                                //Use to stop Trigger mode.
                                mBarcodeManager.stopTrigger();
                        }
                        else {
                            mTrigger.setText(“STOP”);
                            showScanResult.setText(“”);
                            trigermode = true;
                            if(serverconnect)
                                //Start trigger mode.
                                mBarcodeManager.startTrigger();
                        }
                    } catch (Exception e) {
                    }
                } else if (event.getAction() == MotionEvent.ACTION_UP) {
                    try {
                        mTrigger.setPressed(false);
                    } catch (Exception e) {
                    }
                }
                return true;
            }
        }
        );
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
    }

    @Override
    protected void onPause() {
        super.onPause();
        try {
            if(serverconnect) mBarcodeManager.stopDecode();
            if(trigermode) { Log.e(TAG, “onPause trigermode is true, stopTrigger”);
mBarcodeManager.stopTrigger(); }
            mBarcodeManager.removeListener();
            mBarcodeManager.deinit();
        } catch (Exception e) {
        }
```

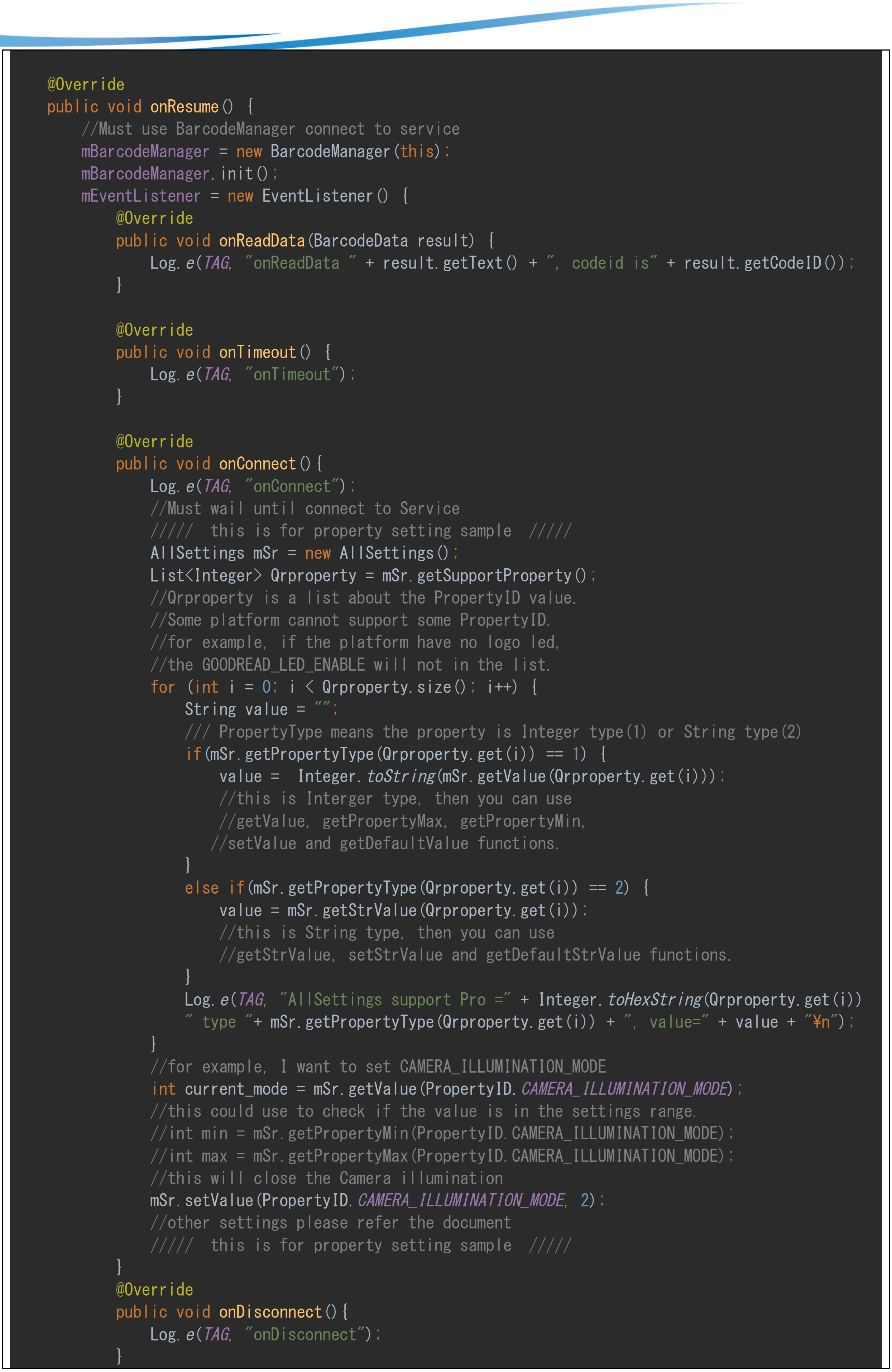


```
    }
    protected void initParameter() {
        status.setText(”Ready to Scan”);
        status.setTextColor(Color.WHITE);
    }
    @Override
    protected void onResume() {
        super.onResume();
        showScanResult.setText(getResources().getText(R.string.scanner_hint));
        initParameter();
        initScan();
        requestPermission();
    }
    @Override
    protected void onStart() {
        super.onStart();
    }
    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        return super.onKeyDown(keyCode, event);
    }
    private void requestPermission() {
        if (checkSelfPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE) !=
PackageManager.PERMISSION_GRANTED) {
            requestPermissions(new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, 0);
        } else {
            Log.d(TAG, ”requestPermission() permission has already been granted”);
        }
    }

}
```

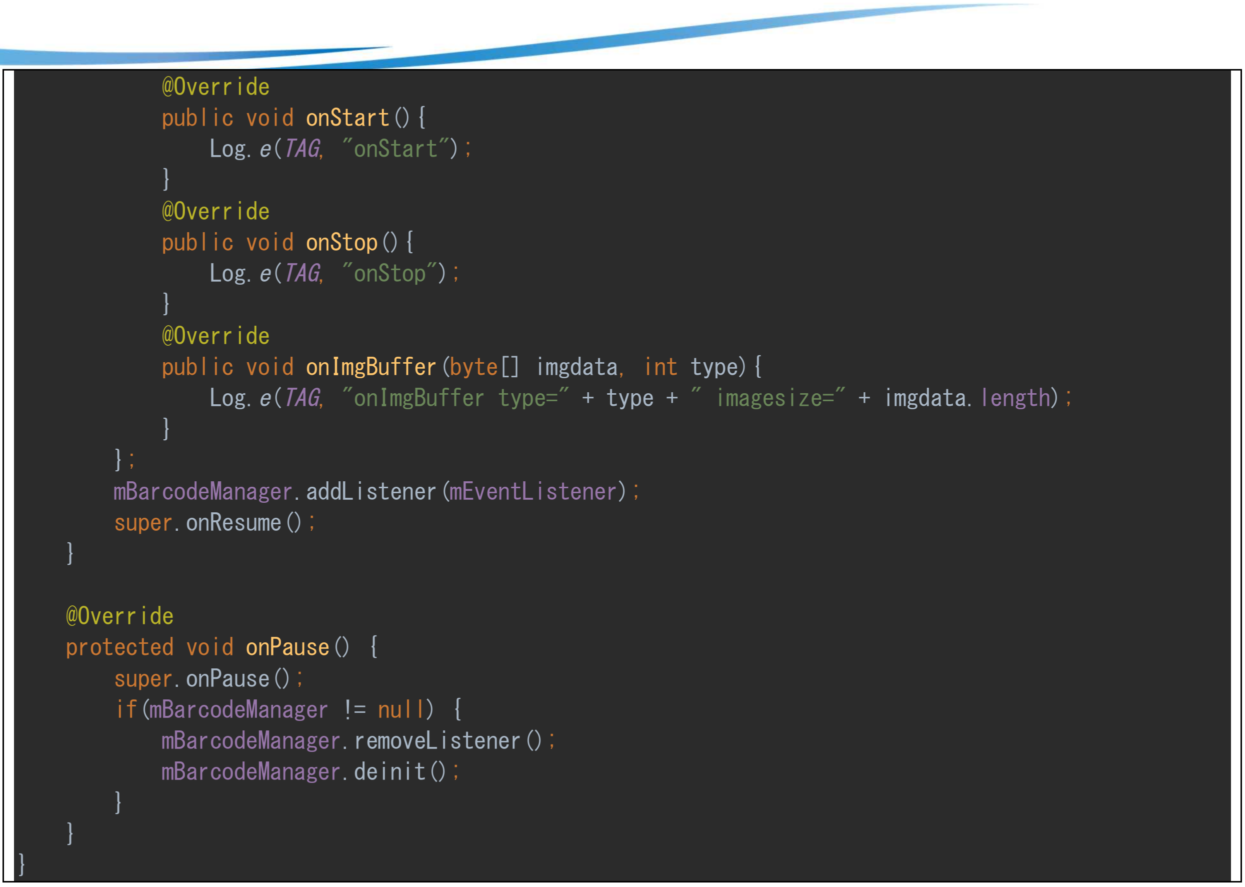
4.3 Barcode Settings Sample



```
package com.test.allsettingssample;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;

import com.extbcr.scannersdk.AllSettings;
import com.extbcr.scannersdk.BarcodeData;
import com.extbcr.scannersdk.BarcodeManager;
import com.extbcr.scannersdk.EventListener;
import com.extbcr.scannersdk.PropertyID;
import java.util.List;

public class MainActivity extends AppCompatActivity {
    public static final String TAG = ”AllSettingsSample”;
    private BarcodeManager mBarcodeManager;
    private EventListener mEventListener;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
```



```
    @Override
    public void onResume() {
        //Must use BarcodeManager connect to service
        mBarcodeManager = new BarcodeManager(this);
        mBarcodeManager.init();
        mEventListener = new EventListener() {
            @Override
            public void onReadData(BarcodeData result) {
                Log.e(TAG, "onReadData " + result.getText() + ", codeid is" + result.getCodeID());
            }

            @Override
            public void onTimeout() {
                Log.e(TAG, "onTimeout");
            }

            @Override
            public void onConnect(){
                Log.e(TAG, "onConnect");
                //Must wail until connect to Service
                /////  this is for property setting sample  /////
                AllSettings mSr = new AllSettings();
                List<Integer> Qrproperty = mSr.getSupportProperty();
                //Qrproperty is a list about the PropertyID value.
                //Some platform cannot support some PropertyID.
                //for example, if the platform have no logo led,
                //the GOODREAD_LED_ENABLE will not in the list.
                for (int i = 0; i < Qrproperty.size(); i++) {
                    String value = "";
                    /// PropertyType means the property is Integer type(1) or String type(2)
                    if(mSr.getPropertyType(Qrproperty.get(i)) == 1) {
                        value =  Integer.toString(mSr.getValue(Qrproperty.get(i)));
                        //this is Interger type, then you can use
                        //getValue, getPropertyMax, getPropertyMin,
                       //setValue and getDefaultValue functions.
                    }
                    else if(mSr.getPropertyType(Qrproperty.get(i)) == 2) {
                        value = mSr.getStrValue(Qrproperty.get(i));
                        //this is String type, then you can use
                        //getStrValue, setStrValue and getDefaultStrValue functions.
                    }
                    Log.e(TAG, "AllSettings support Pro =" + Integer.toHexString(Qrproperty.get(i))
                    " type "+ mSr.getPropertyType(Qrproperty.get(i)) + ", value=" + value + "¥n");
                }
                //for example, I want to set CAMERA_ILLUMINATION_MODE
                int current_mode = mSr.getValue(PropertyID.CAMERA_ILLUMINATION_MODE);
                //this could use to check if the value is in the settings range.
                //int min = mSr.getPropertyMin(PropertyID.CAMERA_ILLUMINATION_MODE);
                //int max = mSr.getPropertyMax(PropertyID.CAMERA_ILLUMINATION_MODE);
                //this will close the Camera illumination
                mSr.setValue(PropertyID.CAMERA_ILLUMINATION_MODE, 2);
                //other settings please refer the document
                /////  this is for property setting sample  /////
            }
            @Override
            public void onDisconnect(){
                Log.e(TAG, "onDisconnect");
            }
```



```
            @Override
            public void onStart(){
                Log.e(TAG, "onStart");
            }
            @Override
            public void onStop(){
                Log.e(TAG, "onStop");
            }
            @Override
            public void onImgBuffer(byte[] imgdata, int type){
                Log.e(TAG, "onImgBuffer type=" + type + " imagesize=" + imgdata.length);
            }
        };
        mBarcodeManager.addListener(mEventListener);
        super.onResume();
    }

    @Override
    protected void onPause() {
        super.onPause();
        if(mBarcodeManager != null) {
            mBarcodeManager.removeListener();
            mBarcodeManager.deinit();
        }
    }
}
```

5 Acronym

API - Application Programming Interface

SDK - Software Development Kit

6 Reference

Revision History

Doc ID: XXXXXXX

製品名: H-33 Scanner SDK

Version	Date	Modification
	20191023	Delete poperty ID "UPCE1_ENABLE" Add property ID of "PropertyID "MAX_UNIQUE_READ" Add property ID of "PropertyID "MULTIPLE_READ_RESET_TIME" Add to enableAllSymologies “Except Add-on code and OCR code”
	20191113	Add backup and restore functions Modify Good read and add read timeout functions Add Dot code, free line font type properties Delete KeyManager and LedManager
	20191212	Parameters (PropertyID & propertyid) unified to propertyId Delete imageType Raw(0)

H-33 Scanner SDK

Android Developer’s Guide

Version:0.2　2019 年 12 月発行

株式会社オプトエレクトロニクス

本社	〒335-0002 埼玉県蕨市塚越 4-12-17
TEL	048-446-1183
FAX	048-446-1184
E-mail	sales@opto.co.jp
URL	http://www.opto.co.jp/