1

2

TPS550 TPS580 TPS586 Software Delevoping SDK Guidance

|  |  |
| --- | --- |
| Revise date： | 2014-06-24 |
| Revise numver： | V1.01 |

[Telepower Communication Co., Ltd](http://www.baidu.com/link?url=bo0dM8KSrwYt-ttT0WT-7j7j5uahFyZWiZ1z2adwhrLiSUix2PX5JUJvFPPLPDPX)

**Record of changes**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number | Release Date | Episode | authorize | examine | Compile |
| **1** | **2014-06-24** | **First issue** |  |  | **Haixin Lin** |
| **2** | **2014-12-30** | **Update interface of printer and NFC** |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Catalog**

1.To build the developing environment 1

1.Download and install the JDK 1

2.Download and install the Eclipse 1

3.Download and install the ADT plug-in 2

4.Download and update the Android SDK 3

2.API 4

1.Interface of Telpo SDK 4

(1) Class definition of Telpo SDK exception operation 4

(2) Class definition of Telpo SDK exception operation 5

(2) Interface smart card/PSAM card 5

 open 7

 close 8

 isOpen 8

 getDeviceId 8

 power 9

 getAtr 9

 setProtocol 10

 getProtocol 10

 getState 10

 transmit 11

(3) Interface of second generation ID card reader 12

 open 12

 checkIdCard 13

 getIdCardImage 13

 decodeIdCardImage 14

 close 14

(4) Operating interface of magnetic card 14

 open 14

 close 15

 check 15

(5) Operating interface of NFC/RFID 15

 openReader 16

 closeReader 16

 selectCard 16

 haltCard 17

 resetReader 17

 command 18

 m1Authority 18

 m1Read 19

 m1Write 19

 m1Add 20

 m1Sub 21

 getReaderInfo 21

(6) Interface of thermal printer 22

 start 23

 reset 24

 walkPaper 24

 stop 24

 checkStatus 24

 setFontSize 25

 enlargeFontSize 25

 setHighlight 26

 setGray 26

 setAlgin 27

 addString 27

 clearString 28

 printString 28

 printStringAndWalk 28

 setLineSpace 29

 printBarcode 29

 printLogo 30

 printLogo 31

 printLogo 31

2. Telpo SDK modules 32

3、 Dot matrix printer external for printing invoice 35

3. Develop with the SDK 37

1. Create an Android programming 37

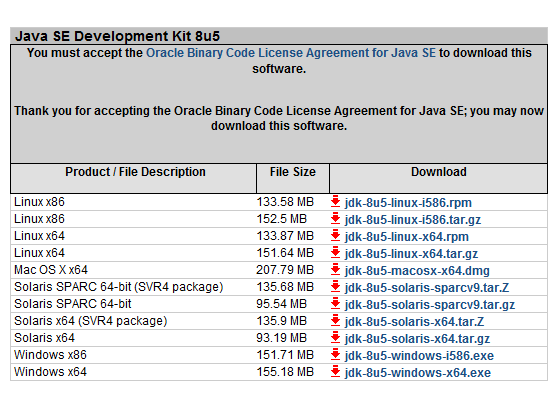
2.Import SDK 39

# 1.To build the developing environment

The OS of Telpo Product TPS550 is Android 4.4,Full support of developing of APPS based on Android SDK.We will show you how to build the environment for developing Android as below.So far, the Android ADE support the OS such as Windows, Linux, and Mac. We recommend you to use Eclipse, ADT, and Android SDK to develop. Here is the ADE:

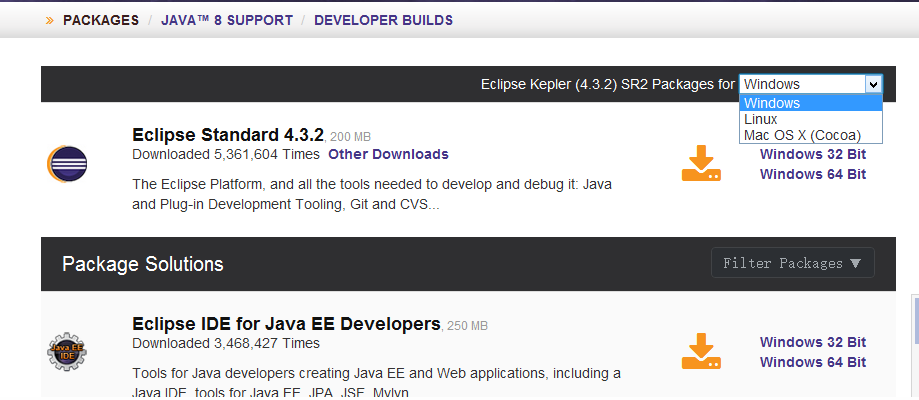
## 1.Download and install the JDK

Open this link: <http://www.oracle.com/technetwork/java/javase/downloads/index.html> , and select the right OS to download and install the JDK. The newest version is 1.8.



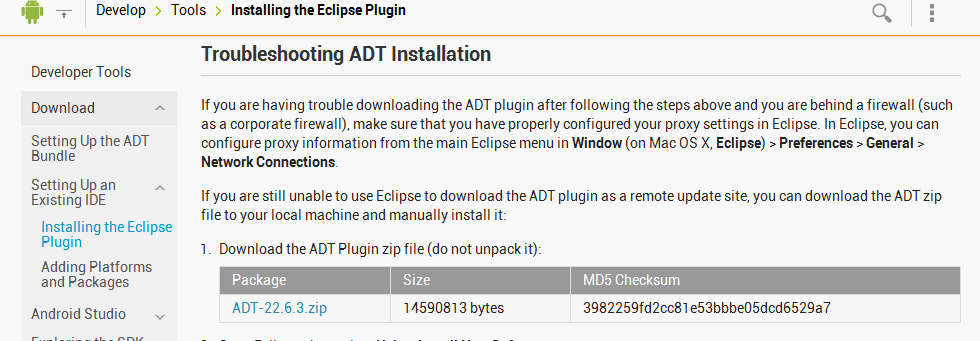
## 2.Download and install the Eclipse

Open this link: <http://www.eclipse.org/downloads/> and select the right OS to download and install it. The newest version is 4.3.2.

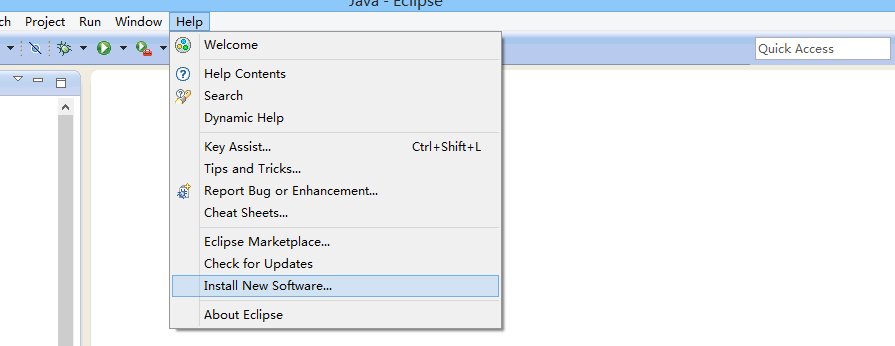


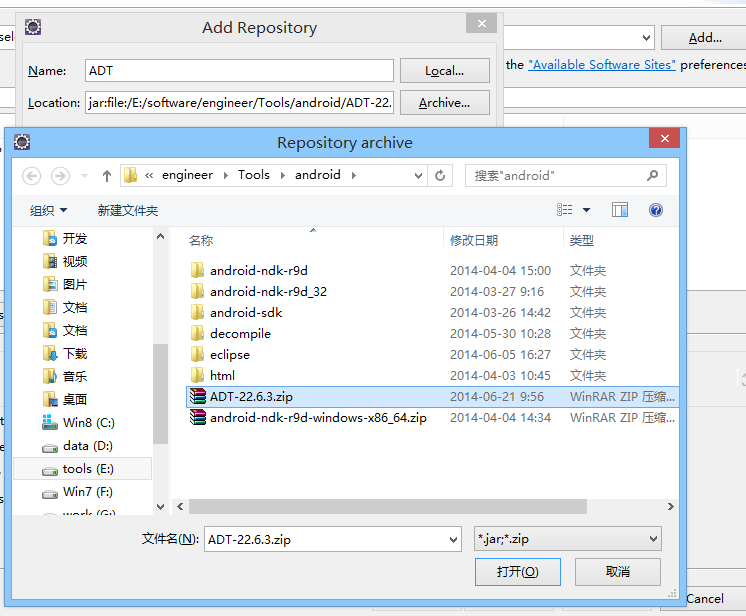
## 3.Download and install the ADT plug-in

ADT is one of the Eclipse plug-in, it is used to connect Android SDK. Open this link: <http://developer.android.com/sdk/index.html> and select the right OS to download this plug-in.



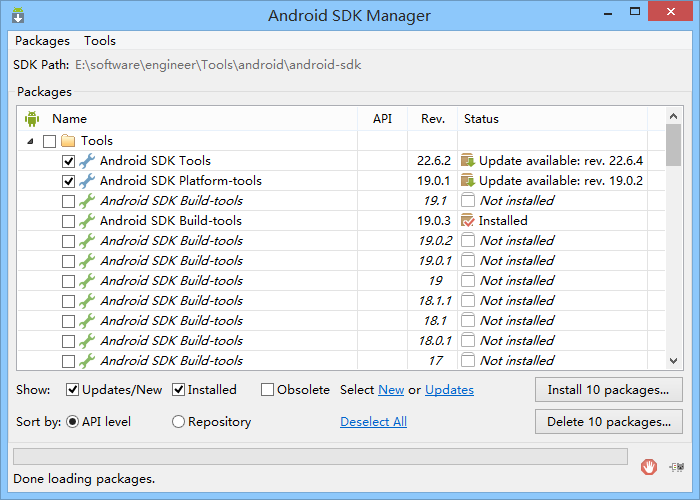
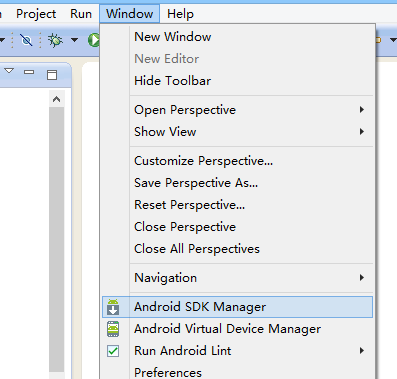
When download completed,open the Eclipse, click Help->Install New Software.Then click “Add”buttom,add this plug-in,choose the local package,click “ok”to start installing the plug-in.When installing,remember to accept the agreement.Reboot the computer when installing finishs.





## 4.Download and update the Android SDK

When installing ADT is completed, click Window->Android SDK Manager. Open SDK Manager, download the right version of SDK and related tools if needed.



# 2.API

The interfaces provided by Android SDK can be refered to the on-line API documents from GOOGLE official( <http://developer.android.com/reference/packages.html> ). Or you can download the relevant DOC documents from Android SDK Manager. It is located in “<SDK Directory>\docs\index.html”.

The SDK interfaces provided by Telpo included printer, smart card,magnetic card, contactless card, second generation id card reader, Qr code identification module, and the calling method of second generation id card OCR recognition module.

## 1.Interface of Telpo SDK

|  |  |
| --- | --- |
| **Software Package** | |
| **Software Package** | [**Annotation**](javascript:void(0);) |
| **com.telpo.tps550.api** | [Class](javascript:void(0);) [definition](javascript:void(0);) of Telpo SDK exception operation |
| **com.telpo.tps550.api.iccard** | operation of smart card reader |
| **com.telpo.tps550.api.idcard** | operation of ID card reader |
| **com.telpo.tps550.api.magnetic** | operation of magnetic card reader |
| **com.telpo.tps550.api.printer** | operation of printer |

### [Class](javascript:void(0);) [definition](javascript:void(0);) of Telpo SDK exception operation

|  |  |
| --- | --- |
| * **Outline of**[**error**](javascript:void(0);)[**exception**](javascript:void(0);) | |
| * **Outline of** [**error**](javascript:void(0);)[**exception**](javascript:void(0);) | [**Annotation**](javascript:void(0);) |
| **DeviceAlreadyOpenException** | Exception of equipment openning repeatly |
| **DeviceNotFoundException** | Exception of equipment not found |
| **DeviceNotOpenException** | Exception of equipment not power-on or equipment resources not applied |
| **DeviceOverFlowException** | Exception of insufficient equipment cache |
| **InternalErrorException** | Exception of throwing errors when internal library running |
| **TelpoException** | Exception of defining Telpo SDK |
| **TimeoutException** | Time-out |

### (2) Interface smart card/PSAM card

* The operation of smart card/PSAM card is in package” com.telpo.tps550.api.iccard”.The operating classes of smart card are defined in this package.So do the exception classes when smart card and PSAM card communicates at times.Here is the outline. Currently supported ISO7816 smart card、SLE4428 card and SLE4442 card。

|  |  |
| --- | --- |
| category | instruction |
| **CardReader** | IC card/PSAM card control base classes |
| **SmartCardReader** | Comply with ISO7816 smartcard control class, inherite from CardReader class |
| **SLE4442Reader** | SLE4442 card control class，inherite from CardReader class |
| **SLE4428Reader** | SLE4428 card control class，inherite from CardReader class |

CardReader：CardReader uses constants are defined as follows：

|  |  |
| --- | --- |
| field | |
| qualifier and type | field and instruction |
| static int | **CARD\_TYPE\_UNKNOWN**  Unknown card type |
| static int | **CARD\_TYPE\_ISO7816**  In accordance with ISO7816 smart CARDS |
| static int | **CARD\_TYPE\_SLE4442**  SLE4442卡 sle4442 card |
| static int | **CARD\_TYPE\_SLE4428**  SLE4428卡 SLE4428 Card |
| static int | **SLOT\_STATUS\_ICC\_ACTIVE**  IC card is existent and available |
| static int | **SLOT\_STATUS\_ICC\_INACTIVE**  IC card is existent but unavailable |
| static int | **SLOT\_STATUS\_ICC\_ABSENT**  Not insert IC card or card has pulled out |

Operating interface of smart card or PSAM card is as below:

#### open

public static void open(int deviceId)

//Open device of smart card of PSAM card

**Return:**

true – success

false – fail

**Throw:**

**Start from this version:**

1.03

#### close

public static void close()

Close the device of smart card or PSAM card

**Return:**

true – success

false – fail

**Start from this version:**

1.03

#### iccPowerOn

public boolean  iccOpen()

//Test reader device of smart card or PSAM card if it is opened

**Return:**

true – success

false – fail

**Start from this version:**

1.03

#### iccPoweroff

public boolean  iccpoweroff()

//Test reader device of smart card or PSAM card if it is closed

**Return:**

true – success

false – fail

**Start from this version:**

1.03

#### getICCStatus

public int getIccStatus ()

**Return:**

SLOT\_STATUS\_ICC\_ACTIVE

SLOT\_STATUS\_ICC\_INACTIVE

SLOT\_STATUS\_ICC\_ABSENT

**Start from this version:**

1.03

#### isICCPresent

public boolean isICCPresent ( )

Whether the IC/PSAM card inserted

**Return:**

true – IC card Has been inserted

false –IC card Has not been inserted or pulled out

**Start from this version:**

1.03

#### get ARTString

public String getATRString()

Acquire resetting replying

**Return:**

ATR character string

null - No reset reply

**Start from this version:**

1.03

#### get CardType

public int getCardType ()

Acquire resetting replying

**Return:**

CARD\_TYPE\_UNKNOWN

CARD\_TYPE\_ISO7816

CARD\_TYPE\_SLE4442

CARD\_TYPE\_SLE4428

**Start from this version:**

1.03

#### switchMode

public boolean switchMode (int cardType)

transform card pattern

**parameter:**

cardType – card type，Currently supports the following types

CARD\_TYPE\_ISO7816, CARD\_TYPE\_SLE4442, CARD\_TYPE\_SLE4428

**Return:**

true – success

false – fail

**Start from this version:**

1.03

[Constant](javascript:void(0);) [definition](javascript:void(0);) used by smart card module is defined as below:

|  |  |
| --- | --- |
| **Field** | |
| **Q**[**ualifier**](javascript:void(0);) **and type** | **Field and annotation** |
| static int | **PROTOCOL\_T0**  T0 protocol |
| static int | **PROTOCOL\_T1**  T1 protocol |
| static int | **PROTOCOL\_NA**  Protocol undefined |
| static int | **CARD\_POWER\_DOWN**  Card is power off |
| static int | **SLOT\_ICC**  IC card slot |
| static int | **SLOT\_PSAM1**  PSAM1 card slot |
| static int | **SLOT\_PSAM2**  PSAM2 card slot |
| static int | **SLOT\_PSAM3**  PSAM3 card slot |

#### SmartCardReader

Construct function

public SmartCardReader(Context context)

public SmartCardReader(Context context, int slot)

**parameter:**

context

slot –card slot，SLOT\_ICC, SLOT\_PSAM1, SLOT\_PSAM2, SLOT\_PSAM3

**Start from this version:**

1.03

#### transmit

public byte[] transmit(byte[] apdu)

send APDU

**parameter:**

apdu –going to send

APDU

**Return:**

Responsive APDU

null – fail

**throw abnormal:**

NullPointerException –throw when send empty apdu

**Start from this version:**

1.03

#### getProtocol

public int getProtocol ()

Obtainting communication protocol

**Return:**

communication protocol which the IC card use

PROTOCOL\_NA, PROTOCOL\_T0, PROTOCOL\_T1

**Start from this version:**

1.03

SLE4442Reader class operation-interface except inheriting CardReader classport，also contains the following interface：

#### SLE4442Reader

Construct function:

public SLE4442Reader(Context context)

parameter:

context

**Start from this version:**

**1.03**

#### readMainMemory

public byte[] readMainMemory(int addr, int num)

read main memory

**parameter:**

addr –initial address，scope：0~255

num –number of bytes read

**Return:**

Content read

null – fail

**Start from this version:**

1.03

#### updateMainMemory

public boolean updateMainMemory(int addr, byte[] data)

must call verification successful first，then write main memory

**parameter:**

addr –initial address，scope：0~255

data – data read，max length256 bytes

**Return:**

true – success

false – fail

**throw exception:**

InvalidParameterException –throw when address and data length are error

NullPointerException –throw when data is empty

**Start from this version:**

1.03

#### pscVerify

public boolean pscVerify(byte[] psc)

verify code successfully is valid to close or reset

**parameter:**

psc – code，fix 3 bytes

**Return:**

true – success

false – fail

**throw exception:**

InvalidParameterException –throw when psc length less than three

NullPointerException –throw when psc is empty

**Start from this version:**

1.03

#### getUserCode

public byte[] getUserCode()

read user code

**Return:**

user code

null – read fail

**Start from this version:**

1.03

#### pscModify

public boolean pscModify(byte[] pscNew)

change password，must call pscVerify to check the password before call the method until check success

**parameter:**

pscNew – new code，fix 3 bytes

**Return:**

true – success

false – fail

**throw exception:**

InvalidParameterException –throw when psc length less than three

NullPointerException –throw when psc is empty

**Start from this version:**

1.03

SLE4428Reader class operation-interface except inheriting CardReader classport，also contains the following interface：

#### SLE4428Reader

Construct function:

public SLE4428Reader(Context context)

**parameter:**

context

**Start from this version:**

1.03

#### readMainMemory

public byte[] readMainMemory(int addr, int num)

read main memory

**parameter:**

addr –initial address，scope：0~1024

num –number of bytes read

**Return:**

Content read

null – fail

**Start from this version:**

1.03

#### updateMainMemory

public boolean updateMainMemory(int addr, byte[] data)

must call verification successful first，then write main memory

**parameter:**

addr –initial address，scope：0~1020

data –data read，max length1021bytes

**Return:**

t true – success

false – fail

**throw exception:**

InvalidParameterException –throw when address and data length are error

NullPointerException –throw when data is empty

**Start from this version:**

1.03

#### pscVerify

public boolean pscVerify(byte[] psc)

verify code successfully is valid to close or reset

**parameter:**

psc –code，fix 2 bytes

**Return:**

true – success

false – fail

**throw exception:**

InvalidParameterException –throw when psc length less than three

NullPointerException –throw when psc is empty

**Start from this version:**

1.03

#### pscModify

public boolean pscModify(byte[] pscNew)

change password，must call pscVerify to check the password before call the method until check success

**parameter:**

pscNew –new code，fix 2 bytes

**Return:**

true – success

false – fail

**throw exception:**

InvalidParameterException –throw when psc length less than two

NullPointerException –throw when psc is empty

**Start from this version:**

1.03

#### getUserCode

public byte[] getUserCode()

read user code

**Return:**

user code

null –read fail

**Start from this version:**

1.03

### (3) Interface of second generation ID card reader

The operation of second generation ID card reader is in package” com.telpo.tps550.api.idcard”.The operating classes of ID card are defined in this package.So do the information class **IdentityInfo** ID card and exception classes when second generation ID card reader operating may occurs.

|  |  |  |
| --- | --- | --- |
| **Outline of class** | | |
| **Class** | | **Annotation** |
| **IdCard** | Operating class of second generation ID card reader | |
| **IdentityInfo** | Information class of ID card | |

|  |  |  |
| --- | --- | --- |
| **Outline of error exception** | | |
| **Outline of exception error** | | **Annotation** |
| **IdCardInitFailException** | Error occurs and throwed when second generation ID card reader  [initialization](javascript:void(0);) fails | |
| **IdCardNotCheckException** | Information of second generation ID card not checked | |
| **ImageDecodeException** | Picture of second generation ID card decoding fails | |

Interface of second generation ID card reader is as below:

#### open

public static void open()

Open the device of second generation ID card reader

**Throw:**

DeviceNotFoundException - Unable to find the device of second generation ID card reader

DeviceAlreadyOpenException - the device of second generation ID card reader has been [initialize](javascript:void(0);)s.

IdCardInitFailException - the device of second generation ID card reader has been [initializ](javascript:void(0);)ation fails.

**Start from this version:**

1.0

#### checkIdCard

public static IdentityInfo checkIdCard(int timeoutInMs)

Search and read the information of second generation ID card

**Parameter:**

timeoutInMs – Time of overtime，unit:ms

**Return:**

Return the information of second generation ID card

**Throw:**

DeviceNotOpenException - the device of second generation ID card reader is not opened

TimeoutException - unable to reader the information of second generation ID card reader before time-out

**Start from this version:**

1.0

#### getIdCardImage

public static byte[] getIdCardImage()

Obtaining second generation ID card’s original image byte stream

**Return:**

Second generation ID card’s original image byte stream

**Throw:**

DeviceNotOpenException - the device of second generation ID card reader is not opened

IdCardNotCheckException - the information of second generation ID card reader is not read

**Start from this version:**

1.0

#### decodeIdCardImage

public static void decodeIdCardImage(java.lang.String fileName)

Decode the second generation ID card’s images and output the BMP image

**Parameter:**

fileName – The location which the BMP images are saved when decoding finishes,such as"/mnt/sdcard/idcard.bmp"，such as"/mnt/sdcard/idcard.bmp"

**Throw:**

java.lang.NullPointerException – Location is empty

ImageDecodeException – Decoding fails

**Start from this version:**

1.0

#### close

public static void close()

Close the second generation ID card reader,release resourses

**Start from this version:**

1.0

### (4) Operating interface of magnetic card

The operation of Magnetic is in package” com.telpo.tps550.api. Magneticcard”,supporting Magnetic card standard ISO7811/2.

#### open

public static void open()

Open the magnetic head and initializes

**Throw:**

DeviceNotFoundException - Unable to find magnetic head

DeviceAlreadyOpenException - Magnetic head is opened

**Start from this version:**

1.0

#### close

public static void close()

Close the magnetic head, release the resources

**Start from this version:**

1.0

#### check

public static java.lang.String[] check(int timeout)

throws[TelpoException](file:///C:\\Documents%20and%20Settings\\telpo\\%E6%A1%8C%E9%9D%A2\\TelpoAPI\\doc2\\com\\telpo\\tps550\\api\\TelpoException.html" \o "com.telpo.tps550.api中的类)

Return Start to read the magnetic card,ready to swipe the card,return if it succeeds

****Parameter:****

timeout - Time of overtime，unit:ms

****Return:****

Return Return a string array which length is 3,expresses the information of three magnetic track

**Throw:**

[DeviceNotOpenException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\DeviceNotOpenException.html) - Throw Unable to open the magnetic head

[TimeoutException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\TimeoutException.html) - Throw Not swipe card before time-out

****Start from this version:****

1.0

### (5) Operating interface of NFC/RFID

The operation of contactless card is in package” com.telpo.tps550.api. iccard”.

|  |  |
| --- | --- |
| **Field** | |
| **Qualifier and type** | **Field and annotation** |
| static int | [PICC\_M1\_TYPE\_A](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\iccard\Picc.html#PICC_M1_TYPE_A)  Secret key A |
| static int | [PICC\_M1\_TYPE\_B](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\iccard\Picc.html#PICC_M1_TYPE_B)  Secret key B |

Main interface defined：

#### openReader

public static void openReader() throws[TelpoException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\TelpoException.html)

Open contactless card reader

****Throw:****

[InternalErrorException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\InternalErrorException.html) - Throw Bottom layer operation failure

[CommunicationErrorException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\iccard\CommunicationErrorException.html) - Throw Communication error

[TimeoutException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\TimeoutException.html) - Time-out

****Start from this version:****

1.0

#### closeReader

public static void closeReader()

Close card reader

#### selectCard

public static int selectCard(byte[] sn) throws [TelpoException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\TelpoException.html)

choose the card and obtain the card serial number

****Parameter:****

sn,serial number cache to receive -

****Return:****the length of card serial number

**Throw:**

java.lang.IllegalArgumentException – Parameter is empty

[InternalErrorException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\InternalErrorException.html) - Bottom layer operation failure

[CommunicationErrorException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\iccard\CommunicationErrorException.html) - hrow Communication error

[DeviceNotOpenException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\DeviceNotOpenException.html) - Card reader is not opened

[TimeoutException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\TimeoutException.html) - Time-out

****Start from this version:****

1.0

#### haltCard

public static void haltCard()throws [TelpoException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\TelpoException.html)

Hang up the card,make current activated card into halt situation.

**Throw:**

[InternalErrorException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\InternalErrorException.html) - Throw Bottom layer operation failure

[CommunicationErrorException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\iccard\CommunicationErrorException.html) - Throw Throw Communication error

[DeviceNotOpenException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\DeviceNotOpenException.html) - Throw Card reader is not opened

[TimeoutException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\TimeoutException.html) - Throw Time-out

****Start from this version:****

1.0

#### resetReader

public static void resetReader()throws [TelpoException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\TelpoException.html)

Reset radio frequency,all cards in antenna area will be made into Idle situation.

**Throw:**

[InternalErrorException](file:///C:\\Documents%20and%20Settings\\telpo\\%E6%A1%8C%E9%9D%A2\\TelpoAPI\\doc2\\com\\telpo\\tps550\\api\\InternalErrorException.html" \o "com.telpo.tps550.api中的类) - Throw Bottom layer operation failure

[CommunicationErrorException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\iccard\CommunicationErrorException.html) - Throw Throw Communication error

[DeviceNotOpenException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\DeviceNotOpenException.html) - Throw Card reader is not opened

[TimeoutException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\TimeoutException.html) - Throw Time-out

[TelpoException](file:///C:\\Documents%20and%20Settings\\telpo\\%E6%A1%8C%E9%9D%A2\\TelpoAPI\\doc2\\com\\telpo\\tps550\\api\\TelpoException.html" \o "com.telpo.tps550.api中的类)

****Start from this version:****

1.0

#### command

public static int command(byte[] snd, int sndLen, byte[] rcv)

throws[TelpoException](file:///C:\\Documents%20and%20Settings\\telpo\\%E6%A1%8C%E9%9D%A2\\TelpoAPI\\doc2\\com\\telpo\\tps550\\api\\TelpoException.html" \o "com.telpo.tps550.api中的类)

APDU command transmission

****Parameter:****

snd, – APDU order to send

sndLen,– The length of APDU order to send

rcv, –The received result returned by APDU order

****Return:****

The length of the received result returned by APDU order

**Throw:**

java.lang.IllegalArgumentException – Parameter is empty

[InternalErrorException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\InternalErrorException.html) - Bottom layer operation failure

[CommunicationErrorException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\iccard\CommunicationErrorException.html) - Communication error

[DeviceNotOpenException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\DeviceNotOpenException.html) - Card reader is not opened

[TimeoutException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\TimeoutException.html) - Time-out

****Start from this version:****

1.0

#### m1Authority

public static void m1Authority(int type, int sectorNo, byte[] password)

throws[TelpoException](file:///C:\\Documents%20and%20Settings\\telpo\\%E6%A1%8C%E9%9D%A2\\TelpoAPI\\doc2\\com\\telpo\\tps550\\api\\TelpoException.html" \o "com.telpo.tps550.api中的类)

Testing secret key in M1 card sections

****Parameter:****

type,secret key type，such as{@link - PICC\_M1\_TYPE\_A} or PICC\_M1\_TYPE\_B

sectorNo,section number -

password,six-byte secret key -

**Throw:**

java.lang.IllegalArgumentException - Parameter is empty

[InternalErrorException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\InternalErrorException.html) - Bottom layer operation failure

[CommunicationErrorException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\iccard\CommunicationErrorException.html) - Communication error

[DeviceNotOpenException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\DeviceNotOpenException.html) - Card reader is not opened

[AuthorityFailException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\iccard\AuthorityFailException.html) –Testing secret key failure

[TimeoutException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\TimeoutException.html) - Time-out

****Start from this version:****

1.0

#### m1Read

public static void m1Read(int blockNo, byte[] buffer)

throws[TelpoException](file:///C:\\Documents%20and%20Settings\\telpo\\%E6%A1%8C%E9%9D%A2\\TelpoAPI\\doc2\\com\\telpo\\tps550\\api\\TelpoException.html" \o "com.telpo.tps550.api中的类)

M1 card reading operation

****Parameter:****

blockNo,Block number -

buffer,16-byte receiving cache -

**Throw:**

java.lang.IllegalArgumentException - Parameter is empty

[InternalErrorException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\InternalErrorException.html) - Bottom layer operation failure

[CommunicationErrorException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\iccard\CommunicationErrorException.html) - Communication error

[DeviceNotOpenException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\DeviceNotOpenException.html) - Card reader is not opened

[NoAuthorityCardException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\iccard\NoAuthorityCardException.html) – Secret key is not tested

[TimeoutException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\TimeoutException.html) – Time-out

****Start from this version:****

1.0

#### m1Write

public static void m1Write(int blockNo, byte[] buffer)

throws[TelpoException](file:///C:\\Documents%20and%20Settings\\telpo\\%E6%A1%8C%E9%9D%A2\\TelpoAPI\\doc2\\com\\telpo\\tps550\\api\\TelpoException.html" \o "com.telpo.tps550.api中的类)

M1 card reading operation

****Parameter:****

blockNo,Block number -

buffer, 16-byte cache to write -

**Throw:**

java.lang.IllegalArgumentException - Parameter is empty

[InternalErrorException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\InternalErrorException.html) - Bottom layer operation failure

[CommunicationErrorException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\iccard\CommunicationErrorException.html) - Communication error

[DeviceNotOpenException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\DeviceNotOpenException.html) - Card reader is not opened

[NoAuthorityCardException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\iccard\NoAuthorityCardException.html) – Secret key is not tested

[TimeoutException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\TimeoutException.html) – Time-out

****Start from this version:****

1.0

#### m1Add

public static void m1Add(int blockNo, byte[] value)

throws[TelpoException](file:///C:\\Documents%20and%20Settings\\telpo\\%E6%A1%8C%E9%9D%A2\\TelpoAPI\\doc2\\com\\telpo\\tps550\\api\\TelpoException.html" \o "com.telpo.tps550.api中的类)

M1card arithmetic addition

****Parameter:****

blockNo,Block number -

buffer,4-byte data -

**Throw:**

java.lang.IllegalArgumentException - Parameter is empty

[InternalErrorException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\InternalErrorException.html) - Bottom layer operation failure

[CommunicationErrorException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\iccard\CommunicationErrorException.html) - Communication error

[DeviceNotOpenException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\DeviceNotOpenException.html) - Card reader is not opened

[NoAuthorityCardException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\iccard\NoAuthorityCardException.html) – Secret key is not tested

[TimeoutException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\TimeoutException.html) – Time-out

****Start from this version:****

1.0

#### m1Sub

public static void m1Sub(int blockNo,byte[] value)

throws[TelpoException](file:///C:\\Documents%20and%20Settings\\telpo\\%E6%A1%8C%E9%9D%A2\\TelpoAPI\\doc2\\com\\telpo\\tps550\\api\\TelpoException.html" \o "com.telpo.tps550.api中的类)

M1card arithmetic subductions

****Parameter:****

blockNo,Block number -

buffer,4-byte data -

**Throw:**

java.lang.IllegalArgumentException - Parameter is empty

[InternalErrorException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\InternalErrorException.html) - Bottom layer operation failure

[CommunicationErrorException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\iccard\CommunicationErrorException.html) - Communication error

[DeviceNotOpenException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\DeviceNotOpenException.html) - Card reader is not opened

[NoAuthorityCardException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\iccard\NoAuthorityCardException.html) – Secret key is not tested

[TimeoutException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\TimeoutException.html) – Time-out

[TelpoException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\TelpoException.html)

****Start from this version:****

1.0

#### getReaderInfo

public static void getReaderInfo(byte[] result)

throws[TelpoException](file:///C:\\Documents%20and%20Settings\\telpo\\%E6%A1%8C%E9%9D%A2\\TelpoAPI\\doc2\\com\\telpo\\tps550\\api\\TelpoException.html" \o "com.telpo.tps550.api中的类)

obtain reader information

****Parameter:****

result,2-byte reader information cache to receive -

**Throw:**

java.lang.IllegalArgumentException - Parameter is empty

[InternalErrorException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\InternalErrorException.html) - Bottom layer operation failure

[CommunicationErrorException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\iccard\CommunicationErrorException.html) - Communication error

[DeviceNotOpenException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\DeviceNotOpenException.html) - Card reader is not opened

[TimeoutException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\TimeoutException.html) – Time-out

[TelpoException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\TelpoException.html)

****Start from this version:****

1.0

### (6) Interface of thermal printer

The operation of thermal printer is in package” com.telpo.tps550.api. printer”,including operating class ThermalPrinter and abnormal class definition.

|  |  |
| --- | --- |
| **Outline of class** | |
| **Class** | **Annotation** |
| **ThermalPrinter** | Class Control Class of Telpo thermal printer |

|  |  |
| --- | --- |
| **Outline of Abnormal error** | |
| **Abnormal error** | **Annotation** |
| **NoPaperException** | Abnormal of printer paper out |
| **OverHeatException** | Abnormal of printer overheating |

Constant defined by ThermalPrinter is as below:

|  |  |
| --- | --- |
| **Field** | |
| **Qualifier and type** | **Field and annotation** |
| static int | **ALGIN\_LEFT**  Left algin |
| static int | **ALGIN\_MIDDLE**  Middle algin |
| static int | **ALGIN\_RIGHT**  Right algin |
| static int | **DIRECTION\_BACK**  Direction of paper travelling:back |
| static int | **DIRECTION\_FORWORD**  Direction of paper travelling:forward |
| static int | **STATUS\_NO\_PAPER**  Printer’s paper out |
| static int | **STATUS\_OK**  Printer is ready to work |
| static int | **STATUS\_OVER\_FLOW**  Printer’s cache is full |
| static int | **STATUS\_OVER\_HEAT**  Printer engine is overheating |
| static int | **STATUS\_UNKNOWN**  The status of printer is unknown |
| static int | **WALK\_DOTLINE**  paper skipping by point row |
| static int | **WALK\_LINE**  paper skipping by character row |

Main interface include：

#### start

public static void start()throws [TelpoException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\TelpoException.html)

Apply for printer resources，operation of printer is after applying. When it is done ,call stop()to release printer.

**Throw:**

[DeviceAlreadyOpenException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\DeviceAlreadyOpenException.html) - Re-apply when resources are already applied

[DeviceNotFoundException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\DeviceNotFoundException.html) - Device is not found

****Start from this version:****

1.0

#### reset

public static void reset()

Reset the printer, restore all setting to default,clear cache

**Throw:**

DeviceNotOpenException - Device or resource is not applied or opened

**Start from this version:**

1.0

#### walkPaper

public static void walkPaper(int line)

Paper skipping

**Parameter:**

line – Number of point row,>0

**Throw:**

java.lang.IllegalArgumentException - Whenthe number of point row is less or equal to zero

DeviceNotOpenException - Device or resource is not applied or opened

DeviceOverFlowException - Cache is full during printing

NoPaperException - Paper out during printing

**Start from this version:**

1.0

#### stop

public static void stop()

Release printer,this must been done if you do not use printer

****Start from this version:****

1.0

#### checkStatus

public static int checkStatus()

Testing status of printer

**Return:**

Value of return:  
STATUS\_OK  
STATUS\_NO\_PAPER  
STATUS\_OVER\_HEAT  
STATUS\_OVER\_FLOW  
STATUS\_UNKNOWN

**Throw:**

DeviceNotOpenException - Device or resource is not applied or opened

**Start from this version:**

**1.0**

#### setFontSize

public static void setFontSize(int type)

throws[TelpoException](file:///C:\\Documents%20and%20Settings\\telpo\\%E6%A1%8C%E9%9D%A2\\TelpoAPI\\doc2\\com\\telpo\\tps550\\api\\TelpoException.html" \o "com.telpo.tps550.api中的类)

Setfont

****Parameter:****

type - Font-size,1-2 The default is font 1

For ASC, font 1 size is 8\*162 while font 2 size is 12\*24

For DBCS, font 1 size is 8\*162 while font 2 size is 12\*24

**Throw:**

java.lang.IllegalArgumentException - Occurs when multiple is less than 1 or more than 2

[DeviceNotOpenException](file:///C:\Documents%20and%20Settings\telpo\%E6%A1%8C%E9%9D%A2\TelpoAPI\doc2\com\telpo\tps550\api\DeviceNotOpenException.html) - Device or resource is not applied or opened

****Start from this version:****

1.0

#### enlargeFontSize

public static void enlargeFontSize(int widthMultiple,int heightMultiple)

Increase the printing fonts’ width and height

**Parameter:**

widthMultiple - 1-2times,original status of printer is 1

heightMultiple - 1-2times,original status of printer is 1

**Throw:**

java.lang.IllegalArgumentException - Occurs when multiple is less than 1 or more than 2

DeviceNotOpenException - Device or resource is not applied or opened

**Start from this version:**

1.0

#### setHighlight

public static void setHighlight(boolean mode)

Set if black and write reversed or not

**Parameter:**

mode - true, Allow printing with black and write reversed

false, forbid printing with black and write reversed

**Throw:**

DeviceNotOpenException - Device or resource is not applied or opened

**Start from this version:**

1.0

#### setGray

public static void setGray(int level)

setting printing gray value

**Parameter:**

level - 1-20, default is 8

**Throw:**

java.lang.IllegalArgumentException – occurs when gray value is lessthan 1 or bigger than 20

DeviceNotOpenException - Device or resource is not applied or opened

**Start from this version:**

1.0

#### setAlgin

public static void setAlgin(int mode)

setting printing alginment

**Parameter:**

mode - alginment  
ALGIN\_LEFT  left algin  
ALGIN\_MIDDLE  middle algin  
ALGIN\_RIGHT  right algin

**Throw:**

java.lang.IllegalArgumentException – Occurs when alignment Parametermode is illegal

DeviceNotOpenException - Device or resource is not applied or opened

**Start from this version:**

1.0

#### addString

public static void addString(java.lang.String content)

Add printing content

**Parameter:**

content - Content to print

**Throw:**

DeviceNotOpenException - Device or resource is not applied or opened

java.lang.NullPointerException - \ Parametercontent is empty

**Start from this version:**

1.0

#### clearString

public static void clearString()

Clear all printing content already added

**Throw:**

DeviceNotOpenException - Device or resource is not applied or opened

**Start from this version:**

1.0

#### printString

public static void printString()

Start printing, do not paper skip when finished

**Throw:**

DeviceNotOpenException - Device or resource is not applied or opened

DeviceOverFlowException - Occurs when cache is full during printing

NoPaperException - Occurs when paper out during printing

**Start from this version:**

1.0

#### printStringAndWalk

public static void printStringAndWalk(int direction,

int mode,

int lines)

Start printing and paper skipping

**Parameter:**

direction -  paper skipping direction after printing  
DIRECTION\_FORWORD, paper skipping forward  
DIRECTION\_BACK, paper skipping backward

mode – paper skipping mode   
WALK\_DOTLINE, point row  
WALK\_LINE, character row

lines – the number of paper skipping lines，1-255

**Throw:**

java.lang.IllegalArgumentException – Parameteris not correct

DeviceNotOpenException - Occurs when cache is full during printing

DeviceOverFlowException - Occurs when cache is full during printing

NoPaperException - Occurs when paper out during printing

**Start from this version:**

1.0

#### setLineSpace

public static void setLineSpace(int lineSpace)

setting linespacing

**Parameter:**

lineSpace -,0-255 Linespacing,unit point row,0-255

**Throw:**

java.lang.IllegalArgumentException – occurs when Linespacing is less than 1 or bigger than 255lineSpace

DeviceNotOpenException - Occurs when cache is full during printing

**Start from this version:**

1.0

#### printBarcode

public static void printBarcode(java.lang.String barcode)

bar code printing,default format is UPC-A

**Parameter:**

barcode - bar code string to print,such as"12345678901",format must be 11 or 12 bit number

**Throw:**

java.lang.NullPointerException – when Parameterbarcode is empty

java.lang.IllegalArgumentException – when string is not all made up ofnumbers or the length of string is not correct

DeviceNotOpenException - Occurs when cache is full during printing

DeviceOverFlowException - Occurs when cache is full during printing

NoPaperException - Occurs when paper out during printing

**Start from this version:**

1.0

#### printLogo

public static void printLogo(int width,

int height,

char[] logo)

Printing images

**Parameter:**

width - Image dot matrix width,width must less than 384

height- Image dot matrix height,height must be the [integral](javascript:void(0);) [multiple](javascript:void(0);) of eight

logo – Image dot matrix data，data arrangement must be virticle sweept,from left to right

**Throw:**

java.lang.NullPointerException – when Parameterlogo is empty

java.lang.IllegalArgumentException – when the width or height of the image is not correct

DeviceNotOpenException - Occurs when cache is full during printing

DeviceOverFlowException - Occurs when cache is full during printing

NoPaperException - Occurs when paper out during printing

**Start from this version:**

1.0

#### printLogo

public static void printLogo(Bitmap image)

throws[TelpoException](file:///C:\\Documents%20and%20Settings\\telpo\\%E6%A1%8C%E9%9D%A2\\TelpoAPI\\doc2\\com\\telpo\\tps550\\api\\TelpoException.html" \o "com.telpo.tps550.api中的类)

Printing images

****Parameter:****

image – Object Bitmap,width must less than 384, height must be the [integral](javascript:void(0);) [multiple](javascript:void(0);) of eight

**Throw:**

java.lang.NullPointerException – when Parameterimage is empty

java.lang.IllegalArgumentException – when the width or height of the image is not correct

DeviceNotOpenException - Occurs when cache is full during printing

DeviceOverFlowException - Occurs when cache is full during printing

NoPaperException - Occurs when paper out during printing

****Start from this version:****

1.0

#### printLogo

public void printLogo(Bitmap image,

int mode)

throws[TelpoException](file:///C:\\Documents%20and%20Settings\\telpo\\%E6%A1%8C%E9%9D%A2\\TelpoAPI\\doc2\\com\\telpo\\tps550\\api\\TelpoException.html" \o "com.telpo.tps550.api中的类)

Printing images

****Parameter:****

image - Object Bitmap,width must less than 384, height must be the [integral](javascript:void(0);) [multiple](javascript:void(0);) of eight

mode - alginment  
ALGIN\_LEFT  left algin  
ALGIN\_MIDDLE middle algin  
ALGIN\_RIGHT right algin

**Throw:**

java.lang.NullPointerException - Parameterimage is empty or mode is illegal

java.lang.IllegalArgumentException - when the width or height of the image is not correct

DeviceNotOpenException - Occurs when cache is full during printing

DeviceOverFlowException - Occurs when cache is full during printing

NoPaperException - Occurs when paper out during printing

****Start from this version:****

1.0

#### paperCut

public static void paperCut()

paper cut

Throws:

Device Not Open Exception - It produce When the resources or equipment are not applied or opened

Starting from the following version:

1.02

Note: This only applies to TPS617

(7) laser 2D barcode connector

laser 2D barcode connector is defined of com.telpo.tps550.api.decode,including operation class decode and exception class definition:

#### open

public static void open()

open laser 2D barcode equipment

Throws:

Exception -

Starting from the following version:

1.04

#### read

public static String read(int timeout)

Read the data and return

parameter:

timeout- unit millisecond

Throws:

Exception -

Starting from the following version:

1.04

#### read

public static String[] readDataAndType(int timeout)

Read the data and code type, data read on String[0], String[1] is code type , String[1] is equal to one, That is one dimensional code, String[1] is equal to two, That is two dimensional code

parameter

timeout- unit millisecond

Throws:

Exception -

Starting from the following version:

1.04

#### disconnect

public static void disconnect()

close laser 2D barcode equipment

Starting from the following version:

1.04

## 2. Telpo SDK modules

Speaking of the functions that are coupling highly and may need interface cooperate with codes at the same time,we encapsulate it into individual module.We provide main Activity external called,return the result in Intent after Activity finishs.

1. **Second generation ID card OCR recognizing module**
2. Module encapsulating

Module encapsulating for Second generation ID card OCR recognizing is as below:

|  |  |  |
| --- | --- | --- |
| Class name | Package name | Class name |
| Activity | com.telpo.tps550.api | com.telpo.tps550.api.ocr.IdCardOcr |

1. Parameter receiving：

Second generation ID card OCR recognizing provides recognition from front view and back view.It is the general Intent which is sent into Parameter”type”zone that distinguish front view or back view.

|  |  |  |
| --- | --- | --- |
| ParameterClass name | Parameter name | Annotation |
| boolean | type | true:front view false:back view |

1. Return the result：

The result code returned is as below after calling OCR recognizing module.

|  |  |
| --- | --- |
| Class name | Annotation |
| int | 0:success -1:failure 1:time-out |

When result code is zero,current obtainting information of ID card will be saved in serializing Class.

|  |  |  |
| --- | --- | --- |
| ParameterClass name | Parameter name | Annotation |
| com.telpo.tps550.api.idcard.IdentityInfo | idInfo | Object which passes the IntentReturn serializing Class. |

Calling as below:

|  |
| --- |
| // proceeding second generation ID card OCR recognizing from front view  Intent intent = new Intent();  intent.setClassName("com.telpo.tps550.api"," com.telpo.tps550.api.ocr.IdCardOcr");  intent.putExtra("type",true);  startActivityForResult(intent,0x123); |
| // proceeding second generation ID card OCR recognizing from front view  Intent intent = new Intent();  intent.setClassName("com.telpo.tps550.api"," com.telpo.tps550.api.ocr.IdCardOcr");  intent.putExtra("type",true);  startActivityForResult(intent,requestCode); |
| //Obtaining result  protected void onActivityResult(intrequestCode, intresultCode, Intent data) {  if (requestCode == 0x123) {  if(resultCode == 0){ //success  IdentityInfoinfo =(IdentityInfo)data.getSerializableExtra("idinfo");  }else{ // failure or other  }  }  } |

1. **QR code scanning module**
2. **Module encapsulating**

Module encapsulating is as below:

|  |  |  |
| --- | --- | --- |
| Class name | Package name | Class name |
| Activity | com.telpo.tps550.api | com.telpo.tps550.api.barcode.Capture |

1. **Return result**

Distinguish the resultCode,the QR code scanning result, which passes ActivityReturn

|  |  |
| --- | --- |
| Class name | Annotation |
| int | 0:success -1:failure 1:time-out |

If the resultCode is zero(success),the QR code information scanned will be in character string form pass the IntentReturn.

|  |  |  |
| --- | --- | --- |
| ParameterClass name | Parameter name | Annotation |
| String | qrCode | Character string which passes the IntentReturn |

Calling methods are as below：

|  |
| --- |
| //proceeding QR code scanning  Intent intent = new Intent();  intent.setClassName("com.telpo.tps550.api"," com.telpo.tps550.api.barcode.Capture");  startActivityForResult(intent,0x124); |
| //Obtain the scanning result  protected void onActivityResult(intrequestCode, intresultCode, Intent data) {  **if** (requestCode == 0x124) {  **if** (resultCode == 0) {  **if** (data != **null**) { //Success  String qrcode = data.getStringExtra("qrCode");  **return**;  }  }  }else{ //Failed  }  } |

## Dot matrix printer external for printing invoice

For dot matrix printer external(USB) for printing invoice,we can operate through the operating interface provide by Android supporting for USB host.The main operating interace includes:

|  |  |  |
| --- | --- | --- |
| Class | Main method | Annotation |
| UsbManager  Class the mainly manages the USB device | getDeviceList() | Get the list of devices，Return a HashMap |
| hasPermission(UsbDevice device) | Check if your application has the authority to connect this USB device,if it can,return true, otherwise return false. |
| openDevice(UsbDevice device) | Open the USB device in order to send or receiving data from thish USB device.Return a link refered to this USB device. |
| requestPermission(UsbDevice device, PendingIntent pi) | Apply for temporary accessing anthority to USB device |
| UsbDevice  Represent a USB device class.Each device has one or more interface.Each interface has one or more node to transport data with the device. | getDeviceClass() | Return the Class type of this USB device.Represent it as a integer. |
| getDeviceId() | Return the only one ID of this device |
| getDeviceName() | Return the name of this device |
| getDeviceProtocol() | Return the protocol Class type of this device |
| getDeviceSubclass() | Return the subclass type of this device |
| getVendorId() | Return the ID of the manufacturer |
| getProductId() | Returnthe product ID |
| getInterfaceCount() | Return the number of devices |
| getInterface(int index) | Get an interface of this device |
| UsbInterface  UsbInterface is a Class indeed,not an interface | getId() | Get the ID number of this interface given |
| getInterfaceClass() | Get the Class type of this interface given |
| getInterfaceSubclass() | Get the subclass of this interface given |
| getInterfaceProtocol() | Get the protocol class type of this interface given |
| getEndpointCount() | Get the interface quantity of this interface given |
| getEndpoint(int index) | Return a UsbEndpoint,for the node accessing the interface appointed by specific index. |
| UsbEndpoint  Represent a node’Class of an interface.This Class provides some methods for you to use. | getAddress() | Get the address of that node |
| getAttributes() | Get the property of that node |
| getDirection() | Get the data transporting direction of that node |
| UsbDeviceConnection  Represent a Class connected to the USB.Through this link we can send and receive data from this USB device. | bulkTransfer(UsbEndpoint endpoint, byte[] buffer, int length, int timeout) | Proceed a large amount of data transmission through the given endpoint.The direction transporting is decided by the direction of this node.Buffer is a byte array to ben sent or received.Length is the length of this array. |
| controlTransfer(int requestType, int request, int value, int index, byte[] buffer, int length, int timeout) | This method is transporting data through Node 0.The transporting direction is decided by the Class type applying.If requestType is USB\_DIR\_OUT,than write the data.If it is USB\_DIR\_IN,than read the data. |

You can recognize the gainscha dot matrix printer(USB) by a xml docunment,defining the VID and PID in a range as below:

**<?xml version="1.0" encoding="utf-8"?>**

**<resources>**

**<usb-device vendor-id="34918" product-id="256" />**

**<usb-device vendor-id="6790" product-id="30084" />**

**<usb-device vendor-id="1137" product-id="85" />**

**<usb-device vendor-id="1659" product-id="8965" />**

**<usb-device vendor-id="1137" product-id="23" />**

**<usb-device vendor-id="1155" product-id="22304" />**

**<usb-device vendor-id="26728" product-id="512" />**

**</resources>**

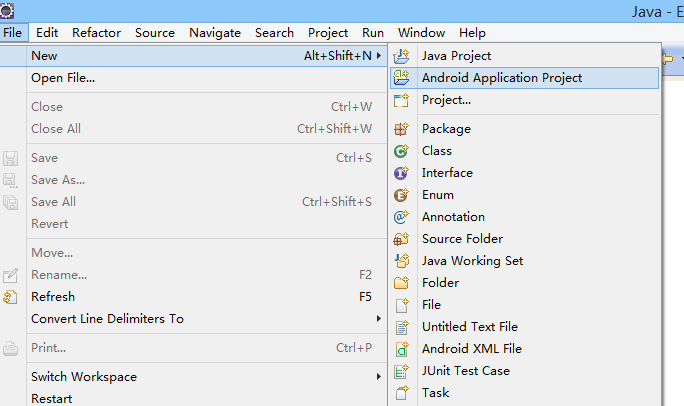
Through the operation as above, you can obtain the Usb Device Connection.You can also use the bulkTransfer to send and receive the printer’s order.Specific printer programming manual is as below:



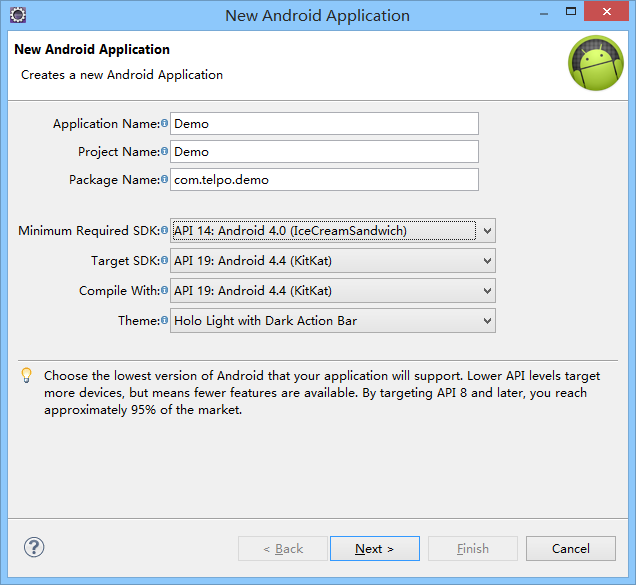
# 3. Develop with the SDK

## 1. Create an Android programming

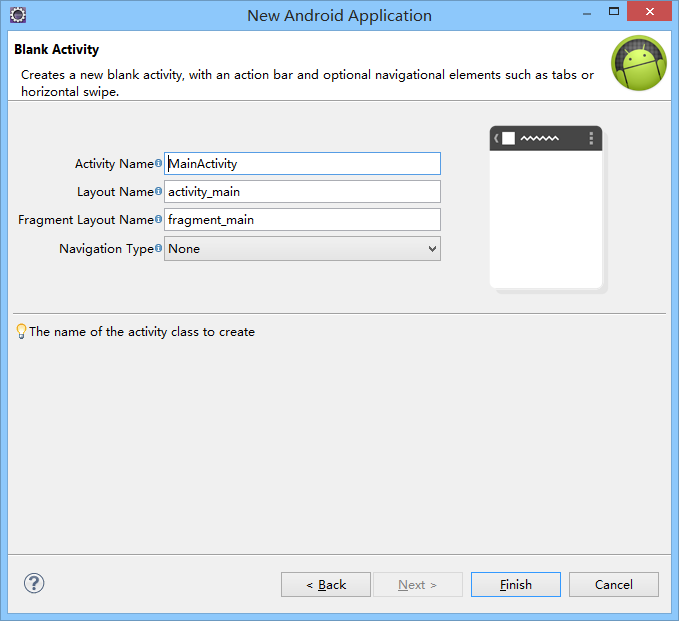
Open Eclipse->File->New->Android Appliction Project(as Graph 3.1.1)，click into “create an Android programming guidance”,input the programming name and choose the Android SDK version.The lowest SDK version choose Android 4.0,while the target SDK version choose Android 4.4(as Graph 3.1.2).Then click “next” until you see the “finish” botton and click it.The programming will be created automatically.



Graph 3.1.1



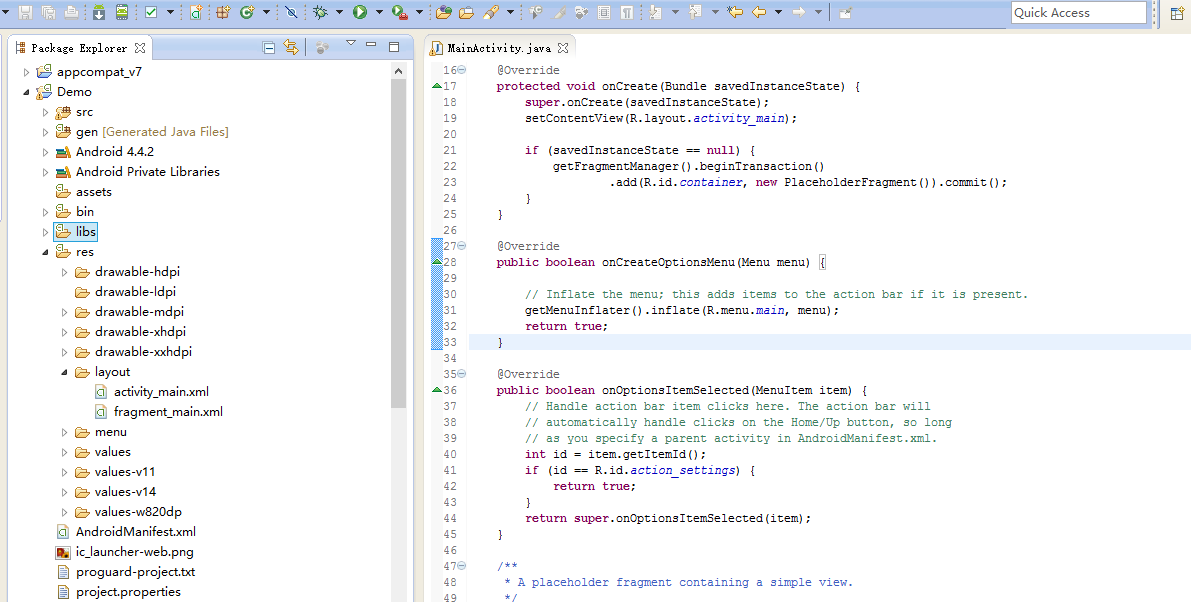
Graph 3.1.2



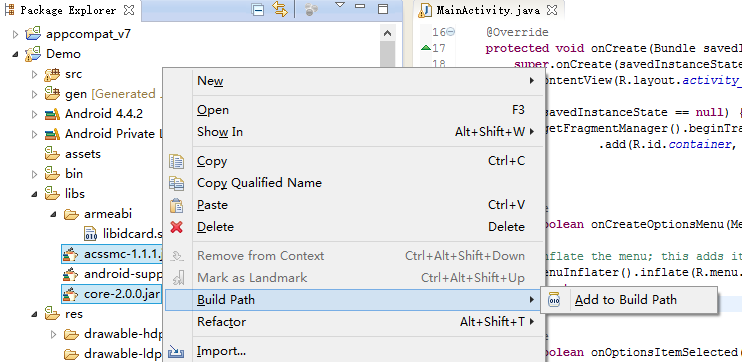
Graph 3.1.3

## 2.Import SDK

You can see the the whole contents and blank Acitivity code which generates automatically of the Android programming as below:

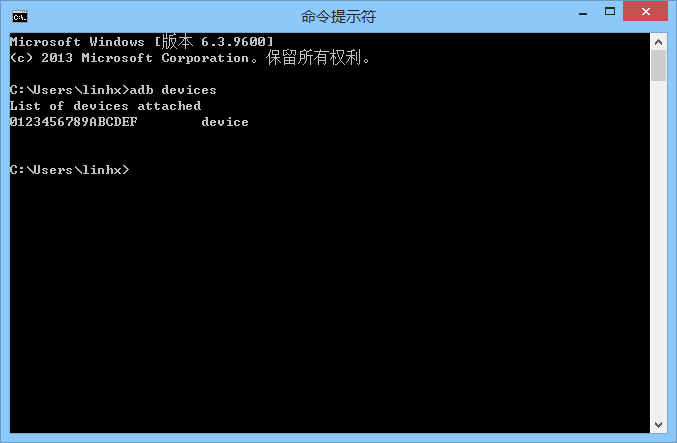


Telpo SDK provides Java library and C library.At first you can create an armeabi subdirectory under the libs directory of the programming.Then copy the jar package provided by Telpo SDK to the libs directory,and copy the .so library files to the armeabi directory.Right-click the jar package, add the compiling address as below:



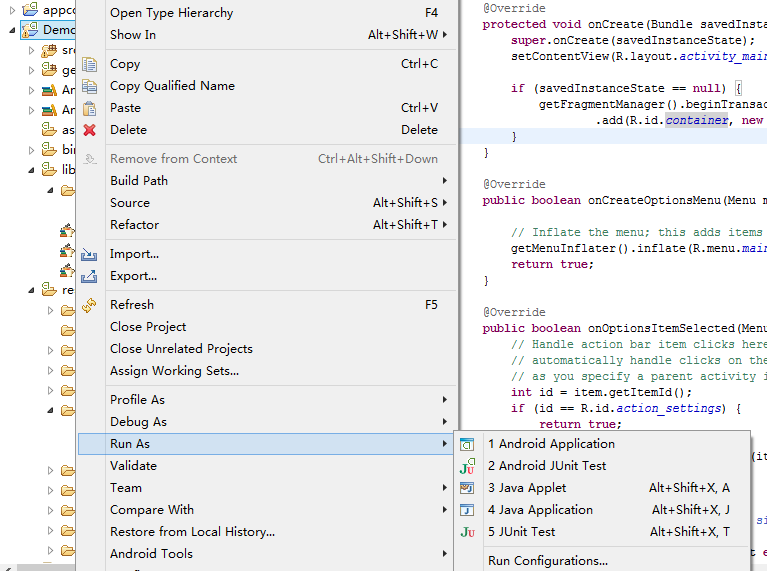
1. **Installing ADB driver**

Choose the right adb driver to install according to different OS.It supports windows XP/Vista/7/8/8.1 32/64bit OS. You can use the software such as Cellphone Assistant to install automatically.When the installing succeeds,open the serial port of command line,input”adb devices”,you can see the Android devices which are connected as below.

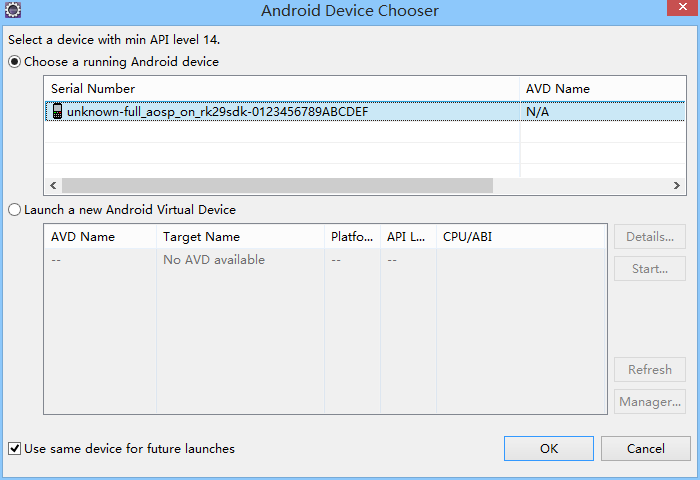


1. **Programming and running**

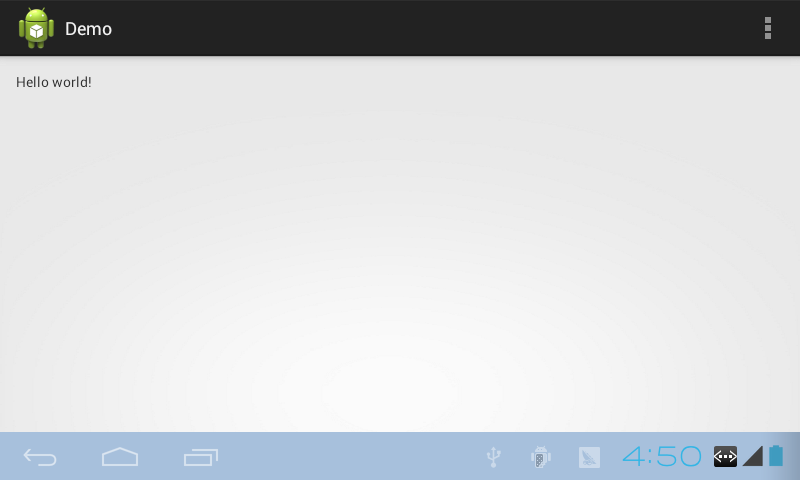
After importing the SDK, you can proceed the application programming according to the interface documents.After programming, use the USB line to connect the device.You can run or debug the applications after installing the ADB driver. Right-click project->Run As->Android Applicationy to run the application. Choose the ”Use same device for future lauches” and click “ok”,then you can run the application.



Graph 3.4.1 Run the Android application



Graph 3.4.2 Choose the device to debug



Graph 3.4.3 Interface running on the terminal of new creating programming