

SDK Developer Guide For Android

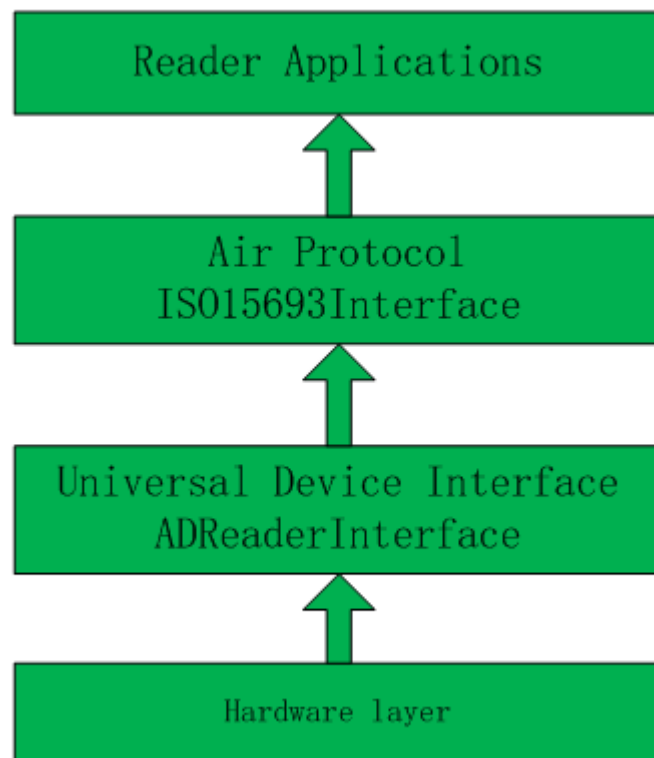
Content

1. User's guidance.....	4
1.1 Overall structure.....	4
1.2 SDK call step.....	4
1.3 Development Process.....	5
1.3.1 Process of inventory.....	5
1.3.2 Process of operation for ISO15693 tag.....	6
1.3.3 Get the scanned record for RPAN device.....	7
2. Interface of API.....	8
2.1 Generic class of reader——ADReaderInterface.....	8
2.1.1 GetPairBluetooth.....	8
2.1.2 GetSerialPortPath.....	9
2.1.3 EnumerateUsb.....	9
2.1.4 GetUsbDescription.....	9
2.1.5 HasUsbPermission.....	10
2.1.6 RequestUsbPermission.....	10
2.1.7 RDR_Open.....	11
2.1.8 RDR_Close.....	13
2.1.9 RDR_GetReaderInfor.....	14
2.1.10 RDR_OpenRFTransmitter.....	15
2.1.11 RDR_CloseRFTransmitter.....	16
2.1.12 RDR_GetRFPower.....	17
2.1.13 RDR_SetRFPower.....	18
2.1.14 RDR_LoadFactoryDefault.....	19
2.1.15 RDR_SetCommulmeTimeout.....	20
2.1.16 RDR_ResetCommulmeTimeout.....	21
2.1.17 RDR_CreateInvenParamSpecList.....	22
2.1.18 RDR_GetTagReportCount.....	23
2.1.19 RDR_TagInventory.....	24
2.1.20 RDR_GetTagDataReport.....	26
2.1.21 RPAN_ClearScanRecord.....	27
2.1.23 RPAN_GetRecord.....	29
2.1.24 RPAN_ParseRecord.....	30
2.1.25 RDR_GetReaderLastReturnError.....	31
2.1.26 RDR_GetOverflowTime.....	32
2.1.27 RDR_SetOverflowTime.....	33
2.1.28 RDR_GetAntennaInterfaceCount.....	34
2.1.29 RDR_SetAcessAntenna.....	35
2.2 Air protocol operation of ISO15693——ISO15693Interface.....	36
2.2.1 ISO15693_CreateInvenParam.....	36
2.2.2 ISO15693_ParseTagDataReport.....	38
2.2.3 ISO15693_Connect.....	39
2.2.4 ISO15693_Disconnect.....	40

2.2.5	ISO15693_ReadMultiBlocks.....	41
2.2.6	ISO15693_WriteMultipleBlocks.....	42
2.2.7	ISO15693_LockMultipleBlocks.....	43
2.2.8	ISO15693_WriteDSFID.....	44
2.2.9	ISO15693_LockDSFID.....	45
2.2.10	ISO15693_WriteAFI.....	46
2.2.11	ISO15693_LockAFI.....	47
2.2.12	ISO15693_GetSystemInfo.....	48
2.2.13	NXPICODESLI_EableEAS.....	49
2.2.13	EASNXPICODESLI_DisableEAS.....	50
2.2.14	EASNXPICODESLI_EASCheck.....	51
2.2.15	NXPICODESLI_LockEAS.....	52
2.2.15	NXPICODESLI_LockEAS.....	53
2.2.16	NXPICODESLI_GetRandomAndSetPassword.....	54
2.2.17	NXPICODESLI_WritePassword.....	55
2.2.18	NXPICODESLI_Enable64BitPwd.....	56
2.2.19	NXPICODESLI_LockPassword.....	57
2.2.20	NXPICODESLI_PasswordProtect.....	58
2.2.22	NXPICODESLI_LockPageProtection.....	61
2.3	Air protocol operation of ISO14443A——ISO14443AInterface.....	62
2.3.1	SpecAPIInvenParamISO14443A.....	62
2.3.2	ISO14443A_ParseTagDataReport.....	64
2.3.3	MFCL_Connect.....	65
2.3.4	MFCL_Authenticate.....	66
2.3.5	MFCL_ReadBlock.....	67
2.3.6	MFCL_WriteBlock.....	68
2.3.7	MFCL_FormatValueBlock.....	69
2.3.8	MFCL_Restore.....	70
2.3.8	MFCL_Increment.....	71
2.3.8	MFCL_Decrement.....	72
2.3.9	ULTRALIGHT_Connect.....	73
2.3.10	ULTRALIGHT_ReadMultiplePages.....	74
2.3.11	ULTRALIGHT_WriteMultiplePages.....	75
2.3.12	ISO14443A_Disconnect.....	76
3.	Connection string description.....	77
3.1	Outline.....	77
3.2	Connection string of Serial ports.....	78
3.3	Connection string of bluetooth.....	79
3.4	Connection string of Net.....	80
3.5	Connection string of USB.....	81
4.	RFIDLIB API error table.....	82

1. User's guidance

1.1 Overall structure

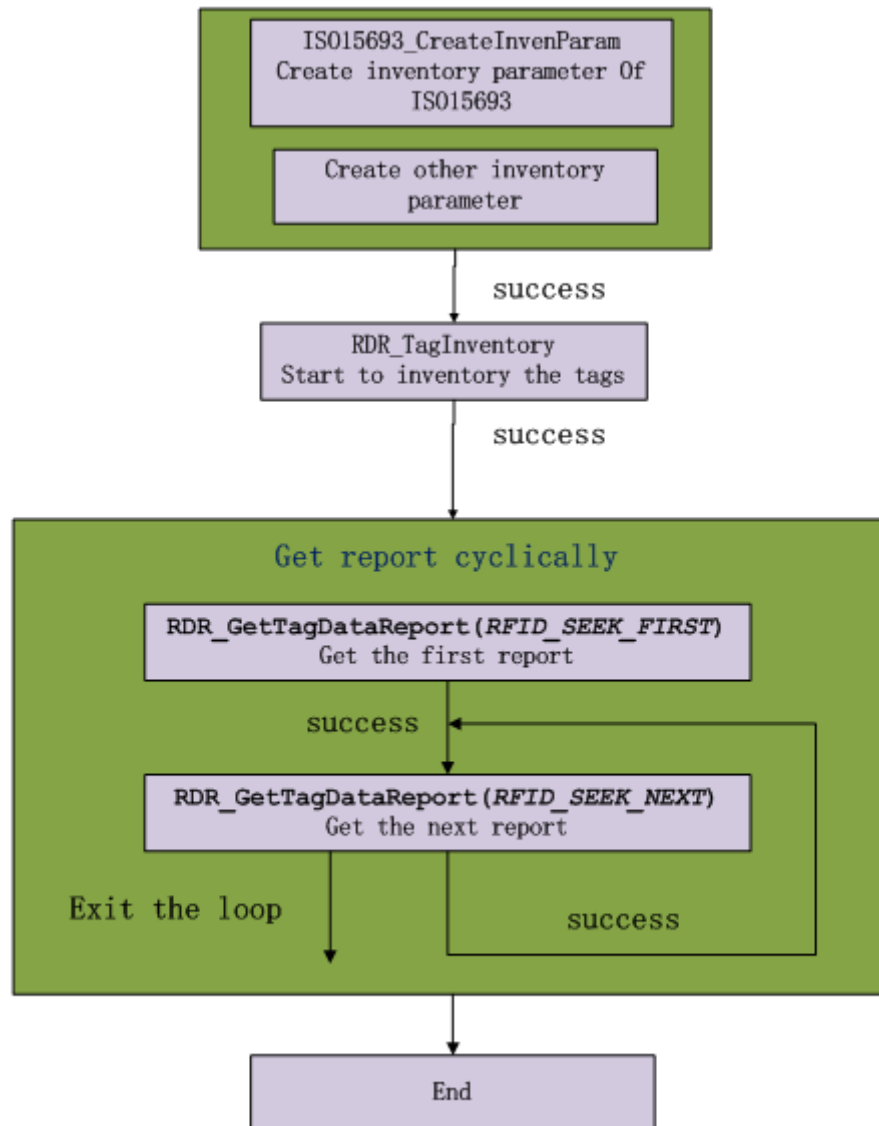


1.2 SDK call step

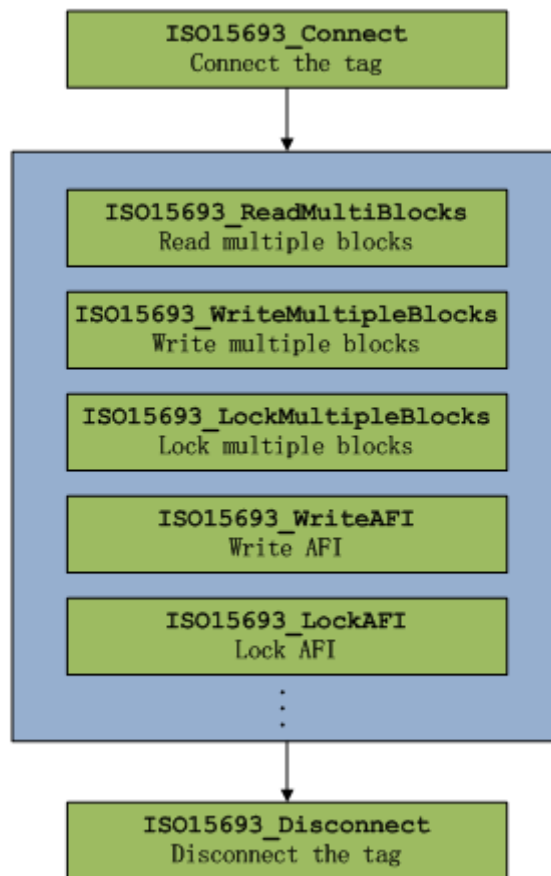
Step 1	Copy the file named “AnReaderLib.jar”, folder named “armeab” and “armeabi-v7a” to the folder named “libs” in your android project.
Step 2	Use the class called “ADReaderInterface” and “ISO15693Interface” to implement the reader function.

1.3 Development Process

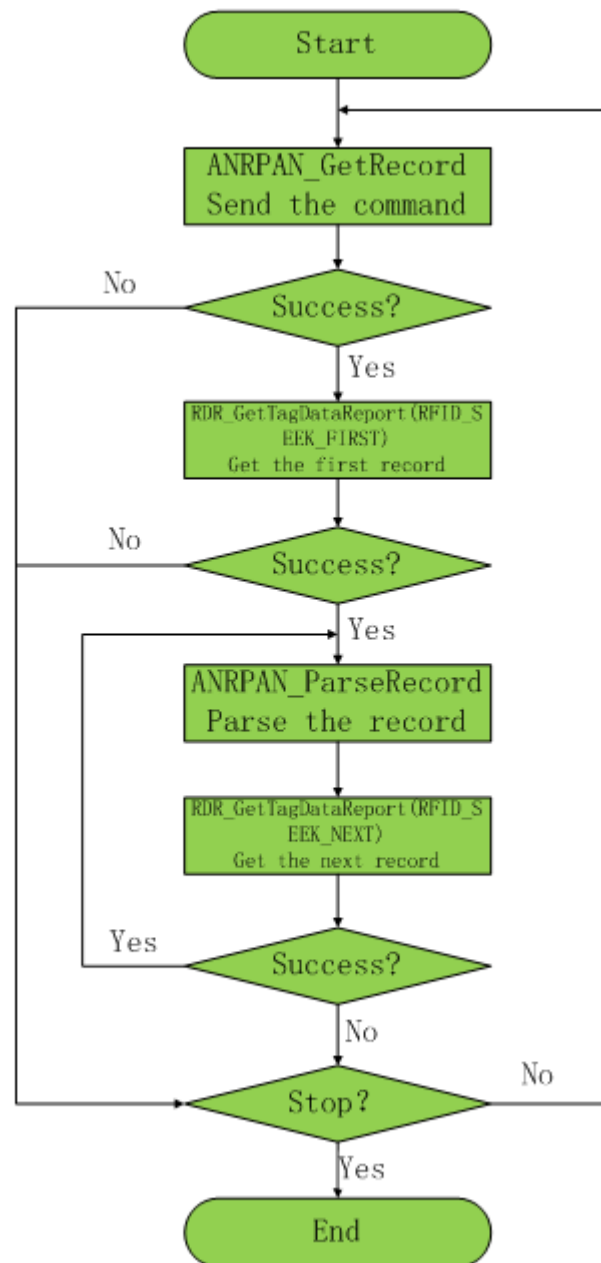
1.3.1 Process of inventory



1.3.2 Process of operation for ISO15693 tag



1.3.3 Get the scanned record for RPAN device



2. Interface of API

2.1 Generic class of reader——ADReaderInterface

2.1.1 GetPairBluetooth

Definition	Public static ArrayList<BluetoothCfg> GetPairBluetooth()
Description	Get the paired bluetooth in the device.
Parameters	No
Return	If successful, the list of bluetooth will be returned.
Remark	We can get the name and address of the bluetooth by the method “GetName” and “GetAddr” in the object of “BluetoothCfg” .

2.1.2 GetSerialPortPath

Definition	<code>public static String[] GetSerialPortPath()</code>
Description	Enumerate all serial ports on the device
Parameters	No
Return	The list of serial ports' path
Remark	

2.1.3 EnumerateUsb

Definition	<code>public static int EnumerateUsb(Context context)</code>		
Description	Enumerate all of the USB device.		
Parameters	Name	Type	Remark
	context	Context	The context.
Return	The USB device count.		
Remark			

2.1.4 GetUsbDescription

Definition	<code>public static String GetUsbDescription(int idx)</code>		
Description	Get the Description of the USB connector.		
Parameters	Name	Type	Remark
	idx	int	Index of enumeration from "EnumerateUsb".

Return	USB Description.
Remark	

2.1.5 HasUsbPermission

Definition	public static boolean HasUsbPermission(String usbDes)		
Description	Determine the permission of USB connector.		
Parameters	Name	Type	Remark
	usbDes	String	USB Description. When empty, it indicates the first enumerated USB device.
Return	true:has permission false: no permission		
Remark			

2.1.6 RequestUsbPermission

Definition	public static boolean RequestUsbPermission(String usbDes)		
Description	Request the permission of USB.		
Parameters	Name	Type	Remark

ers	usbDes	String	USB Description. When empty, it indicates the first enumerated USB device.
Return	true: success false: fail		
Remark			

2.1.7 RDR_Open

Definition	public int RDR_Open(String conStr)		
Description	Open the reader		
Parameters	Name	Type	Remark
	conStr	String	See “Connection string description”
Return	See API error list		
Remark	<p>1. If connected by bluetooth, you must add the following text to “AndroidManifest.xml” :</p> <pre><uses-permission android:name="android.permission.BLUETOOTH_ADMIN" /> <uses-permission android:name="android.permission.BLUETOOTH" /></pre> <p>2. If connected by net, you must add the following text to “AndroidManifest.xml” :</p> <pre><uses-permission android:name="android.permission.ACCESS_WIFI_STATE" /> <uses-permission android:name="android.permission.INTERNET" /></pre>		

	<p>3. List the usb devices first if connected by usb.</p> <p>4. The android system' s version must be more than 5.1.1 if connected by usb.</p>
--	--

2.1.8 RDR_Close

Definition	<code>public int RDR_Close()</code>
Description	Close the reader
Parameters	
Return	See API error list
Remark	No

2.1.9 RDR_GetReaderInfor

Definition	public int RDR_GetReaderInfor(StringBuffer buffer)		
Description	Get the information of the reader		
Parameters	Name	Type	Remark
	buffer	StringBuffer	String buffer used for saving the information of the reader.Format: “Firmware version;Device type;Device serial number ” .The three entry separated by “;” .
Return	See API error list		
Remark			

2.1.10 RDR_OpenRFTransmitter

Definition	<code>public int RDR_OpenRFTransmitter()</code>
Description	Open RF transmitter.
Parameters	
Return	See API error list
Remark	

2.1.11 RDR_CloseRFTransmitter

Definition	<code>public int RDR_CloseRFTransmitter()</code>
Description	Close RF transmitter.
Parameters	
Return	See API error list
Remark	

2.1.12 RDR_GetRFPower

Definition	public int RDR_GetRFPower(Byte mTime)		
Description	Get RF power		
Parameters	Name	Type	Remark
	mTime	Byte	Power index.
Return	See API error list		
Remark			

2.1.13 RDR_SetRFPower

Definition	public int RDR_SetRFPower(byte index)		
Description	Set RF power		
Parameters	Name	Type	Remark
	index	byte	Power index.
Return	See API error list		
Remark			

2. 1. 14 RDR_LoadFactoryDefault

Definition	<code>public int RDR_LoadFactoryDefault()</code>
Description	Restore factory settings
Parameters	
Return	See API error list
Remark	

2. 1. 15 RDR_SetCommulmmeTimeout

Definition	<code>public int RDR_SetCommulmmeTimeout()</code>
Description	Let the API which communicate with the reader timeout immediately.
Parameters	
Return	See API error list
Remark	

2. 1. 16 RDR_ResetCommuImmeTimeout

Definition	public int RDR_ResetCommuImmeTimeout()
Description	We use the API of “API RDR_SetCommuImmeTimeout” to let the thread be timeout immediately. When finished, we must use this API to reset the Application. Otherwise, the next operation will return the error code which equal to -5.
Parameters	No
Return	See API error list
Remark	

2.1.17 RDR_CreateInvenParamSpecList

Definition	Public static Object RDR_CreateInvenParamSpecList()
Description	Create air protocol parameter list data node.
Parameters	
Return	If successful, the handle of data node is returned.
Remark	

2. 1. 18 RDR_GetTagReportCount

Definition	public int RDR_GetTagReportCount()
Description	Get the report count of inventory
Parameters	
Return	The report count.
Remark	

2. 1. 19 RDR_TagInventory

Definit ion	<pre>public int RDR_TagInventory(byte AIType,byte AntennaIDs[],int mTimeout,Object InvenParamSpecList)</pre>		
Descrip tion	<p>Look for tags.The API supports multi-Antenna Interface and multi air interface protocol.The information of tags save in the data node.When finish,we can use the API of RDR_GetTagDataReport to get the information of the tag.</p>		
Paramet ers	Name	Type	Mark
	AIType	byte	<p>1:New query. Before inventory,all of the tags is in ready state.</p> <p>2:Continue query. The tags can not be read when the tag is in quiet state.</p>
	AntennaIDs	Byte[]	The antenna list.If AntennaCount equal to 0, the value can be NULL.
	mTimeout	int	Timeout(ms) 。 If equal to 0,use the default timeout.

	InvenParamSpecList	Object	Air interface protocol parameter list. Create by RDR_CreateInvenParamSpecList.
Return	See API error list		

2. 1. 20 RDR_GetTagDataReport

Definit ion	public Object RDR_GetTagDataReport(byte seek)		
Descrip tion	Get the data report node of inventory tags.		
Paramet ers	Name	Type	Remark
	seek	byte	The cursor position 1:The firt report 2:The next report 3:The last report
Return	The node of current data reports.		
Remark			

2. 1. 21 RPAN_ClearScanRecord

Definition	public int RPAN_ClearScanRecord()
Description	Clear the scan record saved in the RAM.
Parameters	No
Return	See API error list
Remark	Only use for the device of RPAN

2. 1. 22 RPAN_SetTime

Definit ion	public int RPAN_SetTime(int year, int month, int day, int hour, int min,int sec)		
Descrip tion	Set the time of device		
Paramet ers	Name	Type	Remark
	year	int	Year
	month	int	Month
	day	int	Date
	hour	int	Hour
	min	int	min
	sec	int	sec
Return	See API error list		
Remark			

2. 1. 23 RPAN_GetRecord

Definit ion	public int RPAN_GetRecord(byte flg)		
Descrip tion	Get the scan record		
Paramet ers	Name	Type	Remark
	flg	int	Flag. When the previous collection was failed, the value is 0x00. Otherwise, the value is 0x01.
Return	See API error list		
Remark			

2. 1. 24 RPAN_ParseRecord

Definit ion	public byte[] RPAN_ParseRecord(Object hReport)		
Descrip tion	Parse the scan record.		
Paramet ers	Name	Type	Remak
	hReport	Object	The record handle. Create by “RDR_GetTagDataReport ” .
Return	The scanned report.		
Remark			

2. 1. 25 RDR_GetReaderLastError

Definition	public int RDR_GetReaderLastError()		
Description	If the error code is -17 when operating the reader, it shows that the hardware operation is failed and we can get this error code by using this API.		
Parameters	Name	Type	Remark
Return	The error code by hardware returned.		
Remark			

2. 1. 26 RDR_GetOverflowTime

Definit ion	public int RDR_GetOverflowTime(Integer mTime)		
Descrip tion	Get the overflow time of the reader.		
Paramet ers	Name	Type	Remark
	mTime	Integer	The object used for saved the overflow time.Unit:100ms
Return	The error code by hardware returned.		
Remark			

2.1.27 RDR_SetOverflowTime

Definit ion	public int RDR_GetOverflowTime(int mTime)		
Descrip tion	Set the overflow time of the reader.		
Paramet ers	Name	Type	Remark
	mTime	mTime	Overflow time. Unit:100ms
Return	The error code by hardware returned.		
Remark			

2. 1. 28 RDR_GetAntennaInterfaceCount

Definit ion	public int RDR_GetAntennaInterfaceCount()		
Descrip tion	Get the count of antenna interface of reader.		
Paramet ers	Name	Type	Remark
Return	The count of antenna interface.		
Remark			

2. 1. 29 RDR_SetAccessAntenna

Definit ion	public int RDR_SetAccessAntenna(byte AntennaID)		
Descrip tion	For the reader supporting multiple antennas, if you want to read and write the tags, the antenna inductive zone with the tags need to be selected via the API.		
Paramet ers	Name	Type	Remark
	AntennaID	byte	ID number of the antenna interface.
Return	The error code by hardware returned.		
Remark			

2.2 Air protocol operation of ISO15693 — —

ISO15693Interface

2.2.1 ISO15693_CreateInvenParam

Definit ion	public static Object ISO15693_CreateInvenParam(Object hInvenParamSpecList, byte AntennaID, Boolean en_afi,byte afi, byte slot_type)		
Descrip tion	Create the node of inventory for ISO15693 protocol. Use for the API of “RDR_TagInventory”		
Paramet ers	Name	Type	Remark
	hInvenParam SpecList	Object	The handle of inventory Parameters list.Create by the API of “ RDR_CreateInvenParam SpecList” .
	AntennaID	byte	The antenna ID.If equal to 0,All of the antenna will read the tags which supported ISO15693 protocol.
	en_afi	Boolean	If match the AFI value.
	afi	byte	AFI value.
	slot_type	byte	ISO15693 Inventory slot

			time。 0: By default;1: 1 slot time;16: 16 slot time
Return	If success, returns data node handle.		
Remark			

2.2.2 IS015693_ParseTagDataReport

Definit ion	public static int IS015693_ParseTagDataReport(Object hTagReport, IS015693Tag tagData)		
Descrip tion	Analytical the tags information which IS015693 inventory. The data node get from the API of “RDR_GetTagDataReport” .		
Paramet ers	Name	Type	Remark
	hTagReport	Object	The data node.
	tagData	IS015693Tag	The object used for saved the report information.
Return	See API error list.		
Remark	<p>The information of IS015693Tag:</p> <pre>public class IS015693Tag { public long aip_id = 0;//Air protocol type ID. public long tag_id = 0;//The tag type ID. public long ant_id = 0;//The antenna ID. public byte dsfid = 0;//DSFID public byte uid[] = new byte[8];//UID }</pre>		

2.2.3 IS015693_Connect

Definition	public int IS015693_Connect(ADReaderInterface hr, long tagType,byte address_mode, byte uid[])		
Description	Connect the tag of IS015693		
Parameters	Name	Type	Remark
	hr	ADReaderInterface	The object of the reader.
	tagType	long	The tag ID.1:NXP ICODE SLI tag.
	address_mode	byte	Address mode. 0:No address mode.No need to set the tag uid. 1:Address mode.Must set the tag uid .
	uid	byte[]	The tag UID
Return	See API error list.		
Remark			

2.2.4 IS015693_Disconnect

Definition	public int IS015693_Disconnect()		
Description	Disconnect the tag		
Parameters	Name	Type	Remark
Return	See API error list.		
Remark			

2. 2. 5 ISO15693_ReadMultiBlocks

Definit ion	public int ISO15693_ReadMultiBlocks(boolean readSecSta, int blkAddr, int numOfBlksToRead, Integer numOfBlksRead, byte bufBlocks[], Long bytesBlkDatRead)		
Descrip tion	Read multi blocks		
Paramet ers	Name	Type	Remark
	readSecSta	boolean	Whether read security status.
	blkAddr	int	The block address.
	numOfBlksTo Read	int	The block count.
	numOfBlksRe ad	Integer	The block number which have been read.
	bufBlocks	byte[]	String buffer used for saving the block data.
	bytesBlkDat Read	Long	The size written in the string buffer.
Return	See API error list.		
Remark			

2.2.6 ISO15693_WriteMultipleBlocks

Definit ion	public int ISO15693_WriteMultipleBlocks(int blkAddr, int numOfBlks,byte newBlksData[])		
Descrip tion	Write multi blocks		
Paramet ers	Name	Type	Remark
	blkAddr	int	The address of the block.
	numOfBlksTo Read	int	The number of the block.
	newBlksData	byte[]	New data of the block.
Return	See API error list.		
Remark			

2.2.7 IS015693_LockMultipleBlocks

Definit ion	public int IS015693_LockMultipleBlocks(int blkAddr, int numOfBlks)		
Descrip tion	Lock multi blocks		
Paramet ers	Name	Type	Remark
	blkAddr	int	The address of the block.
	numOfBlks	int	The number of the block.
Return	See API error list.		
Remark			

2.2.8 IS015693_WriteDSFID

Definit ion	public int IS015693_WriteDSFID(byte dsfid)		
Descrip tion	Write DSFID		
Paramet ers	Name	Type	Remark
	dsfid	byte	DSFID
Return	See API error list.		
Remark			

2. 2. 9 IS015693_LockDSFID

Definit ion	public int IS015693_LockDSFID()		
Descrip tion	Lock DSFID		
Paramet ers	Name	Type	Remark
Return	See API error list.		
Remark			

2.2.10 IS015693_WriteAFI

Definit ion	public int IS015693_WriteAFI(byte afi)		
Descrip tion	Write AFI		
Paramet ers	Name	Type	Remark
	afi	byte	The value of AFI
Return	See API error list.		
Remark			

2. 2. 11 IS015693_LockAFI

Definit ion	public int IS015693_LockAFI()		
Descrip tion	Lock AFI		
Paramet ers	Name	Type	Remark
Return	See API error list.		
Remark			

2. 2. 12 IS015693_GetSystemInfo

Definit ion	public int IS015693_GetSystemInfo(byte uid[], Byte dsfid,Byte afi, Long blkSize,Long numOfBlok s, Byte icRef)		
Descrip tion	Get the information of the tag.		
Paramet ers	Name	Type	Remakr
	uid[]	Byte[]	The object used for saved uid.
	dsfid	Byte	The object used for saved DSFID.
	afi	Byte	The object used for saving AFI.
	blkSize	Long	The object used for used for saving the size of single block.
	numOfBlok s	Long	The object used for saving the number of blocks.
	icRef	Byte	The object used for saving IC reference.
Return	See API error list.		

2. 2. 13 NXPCODESLI_EableEAS

Definit ion	public int NXPCODESLI_EableEAS()		
Descrip tion	Eable EAS		
Paramet ers	Name	Type	Remark
Return	See API error list.		
Remark			

2. 2. 13 EASNXPCODESLI_DisableEAS

Definit ion	public int NXPCODESLI_Disable()		
Descrip tion	Disable EAS		
Paramet ers	Name	Type	Remark
Return	See API error list.		
Remark			

2. 2. 14 EASNXPCODESLI_EASCheck

Definit ion	public int NXPCODESLI_EASCheck(Byte EASFlag)		
Descrip tion	Check the EAS		
Paramet ers	Name	Type	Remark
	EASFlag	Byte	The object used for saving the EAS status. 0:The EAS is Enable. 1:The EAS disable.
Return	See API error list.		
Remark			

2. 2. 15 NXPICODESLI_LockEAS

Definit ion	public int NXPICODESLI_LockEAS		
Descrip tion	Lock EAS		
Paramet ers	Name	Type	Remark
Return	See API error list.		
Remark			

2. 2. 15 NXPICODESLI_LockEAS

Definit ion	public int NXPICODESLI_LockEAS()		
Descrip tion	Lock EAS.		
Paramet ers	Name	Type	Remark
Return	See API error list.		
Remark			

2. 2. 16 NXPCODESLI_GetRandomAndSetPassword

Definit ion	public int NXPCODESLI_GetRandomAndSetPassword (byte pwdNo, long pwd)		
Descrip tion	Icode slix or Icode slix2 tag password authority, the function get automatically random XOR.		
Paramet ers	Name	Type	Remark
	pwdNo	byte	Password type. 0x01:read 0x02:write 0x04:private 0x08:destory 0x10:EAS/AFI
	pwd	long	4 bytes authentication password
Return	See API error list.		
Remark			

2.2.17 NXPICODESLI_WritePassword

Definit ion	public int NXPICODESLI_WritePassword(byte pwdNo, long pwd)		
Descrip tion	Modify the password. Authorize the old password before this command. The command would be successful if the password is right and unlocked.		
Paramet ers	Name	Type	Remark
	pwdNo	byte	Password type. 0x01:read 0x02:write 0x04:private 0x08:destory 0x10:EAS/AFI
	pwd	long	4 bytes authentication password
Return	See API error list.		
Remark			

2. 2. 18 NXPICODESLI_Enable64BitPwd

Definit ion	public int NXPICODESLI_Enable64BitPwd()		
Descrip tion	Enable 64-bit password.		
Paramet ers	Name	Type	Remark
Return	See API error list.		
Remark			

2. 2. 19 NXPICODESLI_LockPassword

Definit ion	public int NXPICODESLI_LockPassword(byte pwdNo)		
Descrip tion	Lock the password. The password can' t be modified once it is locked successfully. Authorize the old password before this command.		
Paramet ers	Name	Type	Remark
	pwdNo	byte	Password type. 0x01:read 0x02:write 0x04:private 0x08:destory 0x10:EAS/AFI
Return	See the API error list.		
Remark			

2. 2. 20 NXPICODESLI_PasswordProtect

Definit ion	public int NXPICODESLI_PasswordProtect (byte bandType)		
Descrip tion	Enable password protection for EAS or AFI.		
Paramet ers	Name	Type	Remark
	bandType	byte	Password type. 0x00:EAS 0x01:AFI
Return	See the API error list.		
Remark			

2. 2. 21 NXPICODESLI_ProtectPage

Definit ion	public int NXPICODESLI_ProtectPage(byte PPPointer, byte protSta)		
Descrip tion	Protect the blocks. " PPPointer" is used as the demarcation. The low 4-bit data of "proSta" represents the status of the blocks which address are less than the demarcation. The high 4-bit data of "proSta" represents the status of the blocks which address are equal or greater than the demarcation.		
Paramet ers	Name	Type	Mark
	PPPpointer	byte	The value of the demarcation.
	protSta	byte	<ul style="list-style-type: none"> ● Low 4-bit data: The blocks which address are less than the demarcation. ● High 4-bit data: The blocks which address are equal or greater than the demarcation.

			<p>Description of protected status:</p> <p>0x00: public and no protection</p> <p>0x01:Read and write blocks with the reading password;</p> <p>0x02: Write blocks with the written password.</p> <p>0x03: Read blocks with the reading password and write blocks with the written password.</p> <p>If the 64-bit password is enable, the operation is protected by the reading password and written password.</p>
Return	See the API error list.		
Remark			

2. 2. 22 NXPCODESLI_LockPageProtection

Definit ion	public int NXPCODESLI_LockPageProtection(byte pageAddr)		
Descrip tion	Lock the protection status. Once locked successfully, it can' t recover to unlock status again.		
Paramet ers	Name	Type	Remark
	pageAddr	byte	The demarcation of the block.
Return	See the API error list.		
Remark			

2.3 Air protocol operation of ISO14443A — — ISO14443AInterface

2.3.1 SpecAIPInvenParamISO14443A

Definit ion	public static SpecAIPInvenParamISO14443A ISO14443A_CreateInvenParam(Object hInvenParamSpecList, byte AntennaID)		
Descrip tion	Create ISO14443A Inventory Parameter data nodes for “RDR_TagInventory” .		
Paramet ers	Name	Type	Remark
	hInvenParamSpecList	Object	The handle of Inventory parameter list created by RDR_CreateInvenParamSpecList.
	AntennaID	byte	The antenna ID, 0 means All of the antenna interface will read the tags which supported ISO14443A protocol. Other values for the corresponding antenna interface .

Return	See API error list.
Remark	

2.3.2 ISO14443A_ParseTagDataReport

Definition	<pre> public static int ISO14443A_ParseTagDataReport (Object hTagReport, ISO14443ATag tagData) </pre>		
Description	<p>Analysis ISO14443A tag data information. Obtained data node from RDR_GetTagDataReport, for the RDR_TagInventory function.</p>		
Parameters	Name	Type	Remark
	hTagReport	Object	The node handle of the label data information
	tagData	ISO14443ATag	The data of the tag.
Return	See API error list.		
Remark	<pre> ISO14443ATag: public class ISO14443ATag{ public long aip_id=0;//protocol id public long tag_id=0;//tag id public long ant_id=0;//antenna id public byte uid[]=null;//UID } </pre>		

2.3.3 MFCL_Connect

Definit ion	public int MFCL_Connect (ADReaderInterface hr, byte tagType, byte uid[])		
Descrip tion	Activate mifare classic tag		
Paramet ers	Name	Type	Remark
	hr	ADReaderInt erface	The object of the reader.
	tagType	byte	Tag Type: 0:Mifare S50 1:Mifare S70
	uid	Byte[]	Tag serial number
Return	See API error list.		
Remark			

2.3.4 MFCL_Authenticate

Definit ion	public int MFCL_Authenticate(byte blkAddr, byte keyType, byte key[])		
Descrip tion	Key Authentication		
Paramet ers	Name	Type	Remark
	blkAddr	byte	Block Address
	keyType	byte	MIFARE® classic Authentication key type. 0:key A 1:key B
	key	byte []	6 bytes of key data
Return	See API error list.		
Remark			

2.3.5 MFCL_ReadBlock

Definit ion	public int MFCL_ReadBlock(byte blkAddr, byte blkData[])		
Descrip tion	See API error list.		
Paramet ers	Name	Type	Remark
	blkAddr	byte	Block Address
	blkData	byte []	The byte buffer is used to hold the data block has read .
Return	See API error list.		
Remark			

2.3.6 MFCL_WriteBlock

Definit ion	public int MFCL_WriteBlock(byte blkAddr, byte blkData[])		
Descrip tion	Write block		
Paramet ers	Name	Type	Remark
	blkAddr	byte	Block Address
	blkData	byte []	Prepare the 16-byte block data to be written.
Return	See API error list.		
Remark			

2.3.7 MFCL_FormatValueBlock

Definit ion	public int MFCL_FormatValueBlock(byte blkAddr, long initValue)		
Descrip tion	Format the value data block.		
Paramet ers	Name	Type	Remark
	blkAddr	byte	Block Address
	initValue	long	Initialization Value
Return	See API error list.		
Remark			

2.3.8 MFCL_Restore

Definit ion	public int MFCL_Restore(byte blkAddr)		
Descrip tion	Backup amount, before using backup, the data block must first be formatted with the value block by MFCL_FormatValueBlock.		
Paramet ers	Name	Type	Remark
	blkAddr	byte	Block Address
Return	See API error list.		
Remark			

2.3.8 MFCL_Increment

Definit ion	public int MFCL_Increment(byte blkAddr, long val)		
Descrip tion	Recharge, before using recharge, the data block must first be formatted with the value block by MFCL_FormatValueBlock.		
Paramet ers	Name	Type	Remark
	blkAddr	byte	Block Address
	val	long	Recharge Quota
Return	See API error list.		
Remark			

2.3.8 MFCL_Decrement

Definit ion	public int MFCL_Decrement(byte blkAddr, long val)		
Descrip tion	Decrement, before using decrement, the data block must first be formatted with the value block by MFCL_FormatValueBlock.		
Paramet ers	Name	Type	Remark
	blkAddr	byte	Block Address
	val	long	Decrement Quota
Return	See API error list.		
Remark			

2.3.9 ULTRALIGHT_Connect

Definit ion	public int ULTRALIGHT_Connect(ADReaderInterface hr, byte uid[])		
Descrip tion	Connect Ultralight tag.		
Paramet ers	Name	Type	Remark
	hr	ADReaderInt erface	The object of the reader.
	uid	Byte[]	7-byte serial number
Return	See API error list.		
Remark			

2.3.10 ULTRALIGHT_ReadMultiplePages

Definit ion	public int ULTRALIGHT_ReadMultiplePages(long pageStart, long pageNum, byte databuf[], Integer nSize)		
Descrip tion	Read multiple pages		
Paramet ers	Name	Type	Remark
	pageStart	long	Page start, ultralight page address 0-15; ultralightC page address 0-47;
	pageNum	long	Pages amount
	databuf	Byte[]	Read the page data
	nSize	Integer	The number of bytes written to the databuf buffer;
Return	See API error list.		
Remark			

2.3.11 ULTRALIGHT_WriteMultiplePages

Definition	<pre>public int ULTRALIGHT_WriteMultiplePages(int pageStart, int pageNum, byte databuf[], int bytesToWrite)</pre>		
Description	Write multiple pages		
Parameters	Name	Type	Remark
	pageStart	int	Page start, ultralight page address 0-15; ultralightC page address 0-47;
	pageNum	int	Pages amount
	databuf	byte[]	The page data to be written
	bytesToWrite	int	The number of bytes to write
Return	See API error list.		
Remark			

2. 3. 12 IS014443A_Disconnect

Definit ion	public int IS014443A_Disconnect()		
Descrip tion	Disconnect the tag of IS014443A.		
Paramet ers	Name	Type	Remark
Return	See API error list.		
Remark			

3. Connection string description

3.1 Outline

The connection string consists of multiple fields , separated by “;”. Every field consists of the field name and the value, separated by “=” .

3.2 Connection string of Serial ports

3.2.1 example

RDType=M201;CommType=COM;ComPath=/dev/ttyMT1;Baund=38400;Frame=8E1;Addr=255

3.2.2 Description

No.	Name	Explanation	
1	RDType	The driver name	
2	CommType	Communicate type.Use “COM” here.	
		Value	Meaning
		COM	Serial ports
		BLUETOOTH	Blueth
		NET	Net
3	ComPath	The path of Serial ports, such as “/dev/ttyMT1”	
4	Baund	The baund, such as “38400”	
5	Frame	The frame	
		Value	Meaning
		8E1	
		801	
		8N1	
6	Addr	The bus address. 1-254:the address. 255:the broadcast address.	

3.3 Connection string of bluetooth

3.3.1 example

RDType=RPAN;CommType=BLUETOOTH;Name=R-PAN. D835

3.3.2 Description

No.	Name	Explanation	
1	RDType	The driver name	
2	CommType	Communicate type. Use “BLUETOOTH” here.	
		Value	Meaning
		COM	Serial ports
		BLUETOOTH	Bluetooth
		NET	Net
3	Name	The bluetooth name.	

3.4 Connection string of Net

3.4.1 example

RDType=RPAN;CommType=NET;RemoteIp=192.168.1.88;RemotePort=4800

3.4.2 Description

No.	Name	Explanation	
1	RDType	The driver name	
2	CommType	Communicate type. Use “NET” here.	
		Value	Meaning
		COM	Serial ports
		BLUETOOTH	Bluetooth
		USB	USB
		NET	Net
3	RemoteIp	The IP address	
4	RemotePort	The port	

3.5 Connection string of USB

3.5.1 example

RDType=RL8000;CommType=USB;Description=

3.4.2 Description

No.	Name	Explanation	
1	RDType	The driver name	
2	CommType	Communicate type. Use “USB” here.	
		Value	Meaning
		COM	Serial ports
		BLUETOOTH	Bluetooth
		USB	USB
		NET	Net
3	Description	USB device index, which start with 0.	

The android system' s version must be more than 5.1.1.

4. RFIDLIB API error table

Error	Description
0	No error
-1	Unknown error
-2	IO error
-3	Parameter error
-4	Parameter value error
-5	Reader respond timeout
-6	Memory allocation fail
-7	Reserved
-8	Reserved
-9	Reserved
-10	Reserved
-11	Reserved
-12	Invalid message size from reader
-13	Reserved
-14	Reserved
-15	Reserved
-16	Reserved
-17	Error from reader, can use “RDR_GetReaderLastReturnError” to get reader error code .
-18	Reserved
-19	Reserved
-20	Reserved
-21	Timeout stop trigger occur .
-22	Invalid tag command
-23	Invalid Configuration block No
-24	Reserved
-25	TCP socket error
-26	Size of input buffer too small.
-27	Reserved