Newland Android PDA

UHF Module Developer Handbook

Revision History

Version	Description	Date
V1.0.0	Initial release (UHF module detection function included).	June 18, 2019

Table of Contents

About This Manual

This manual is developer guide for PDA UHF module, which allows third-party applications to interact with UHF module. It is only available to Newland Android Portable Data Collectors (hereinafter referred to **“the terminal”**) with UHF capability.

Development Process

1. Importing SDK

Import SDK JAR file into Android development environment like Eclips. Add the JAR file to libs folder and load it into the compile environment.

2. Detecting Module

Power on the terminal to auto-detect UHF module and obtain module model number. You can try detecting it manually if automatic detection fails.

3. Powering on Module

Power on UHF module before operations. Default configuration parameters are automatically initialized when power on for the first time and saved parameters initialized subsequently.

4. Operating Module Functions

Power on UHF module to operate the following functions:

- Inventory Trigger
- Inventory Notification
- Access Control (Read, write, lock and kill tags)
- UHF Settings

How to Develop Functions

1. Module Detection

Description: Auto-detect UHF module and obtain its model number.

Interface definition:

Class：com.nlscan.android.uhf.UHFManager	
Return Value	Method
static UHFManager	**getInstance()** To get instance
UHFModuleInfo	**loadUHFModule ()** To detect UHF module and obtain its model number (Power on the terminal to auto-detect UHF module and obtain module model number. Detect it manually if automatic detection fails. It is time consuming process and asynchronous execution is recommended.） Return value: UHF Module Info (Null: Fail to detect module)

2. UHF Module On

Description: Auto-detect UHF module and obtain its model number.

Interface definition:

Class: com.nlscan.android.uhf.UHFManager	
Return Value	Method
static UHFManager	**getInstance()** To get instance
UHFReader.READER_STATE	**powerOn()** To power on UHF Module （Power-on process is time consuming. Asynchronous execution is recommended and wait for return state to prevent main thread blocked.）
boolean	**isPowerOn()** Power on or not?

Example:

//Get UHFManager instance

UHFManager mUHFMgr = UHFManager.getInstance();

//Power on

UHFReader.READER_STATE er = mUHFMgr.powerOn();

if(er == UHFReader.READER_STATE. OK_ERR) //success

.....

3. Inventory Trigger

Description: Set inventory operation triggers including SCAN key, left/right SCAN key, background SCAN key.

- **SCAN Key:** Use the Scan key on the front panel of the terminal as inventory trigger.
- **Left/Right SCAN key:** Use the Scan keys on the left/right side of the terminal as inventory trigger.
- **Background SCAN key:** Use the trigger on the pistol grip attached to the terminal as inventory trigger.

Interface definition:

Class: com.nlscan.android.uhf.UHFManager	
Return Value	Method
static UHFManager	**getInstance()** To get instance
boolean	**setTrigger(String trigger,boolean on)** To set trigger Parameters: trigger identifier Main SCAN key UHFCommonParams.TRIGGER_MODE.TRIGGER_MODE_MAIN Left/Right SCAN key UHFCommonParams.TRIGGER_MODE.TRIGGER_MODE_LEFT UHFCommonParams.TRIGGER_MODE.TRIGGER_MODE_RIGHT Trigger on the pistol grip UHFCommonParams.TRIGGER_MODE.TRIGGER_MODE_BACK Parameters: on true: enable, false: disable Return value: true: set successfully, false: set failed
boolean	**isTriggerOn(String trigger)** Is trigger on? Parameters: trigger identifier Main SCAN key UHFCommonParams.TRIGGER_MODE.TRIGGER_MODE_MAIN Left/Right SCAN key UHFCommonParams.TRIGGER_MODE.TRIGGER_MODE_LEFT UHFCommonParams.TRIGGER_MODE.TRIGGER_MODE_RIGHT Trigger on the pistol grip UHFCommonParams.TRIGGER_MODE.TRIGGER_MODE_BACK Return value: true: enable, false: disable

Example:

//get UHFManager instance

UHFManagermUHFMgr = UHFManager.getInstance();

//--set trigger

```
booleanbBackScan = true;
booleansuc =
 mUHFMgr.setTrigger(UHFCommonParams.TRIGGER_MODE.TRIGGER_MODE_BACK,
bBackScan);
```

//----get isTriggerOn

```
booleanbackTriggerEnable = mUHFMgr.
    isTriggerOn(UHFCommonParams.TRIGGER_MODE.TRIGGER_MODE_BACK);
```

4. Inventory Notification

Description: Use sound and/or vibration to indicate a good tag read.

- **Sound**：Use sound to indicate a good tag read.
- **Vibration**：Use vibration to indicate a good tag read.

Interface definition:

Class: com.nlscan.android.uhf.UHFManager	
Return Value	Method
static UHFManager	**getInstance()** To get instance
boolean	**setPromptSoundEnable(boolean enable)** To set sound on/off Parameters: enable/disable sound (true: enable, false: disable) Return value: true: set successfully, false: set failed
boolean	**setPromptVibrateEnable(boolean enable)** To set vibration on/off Parameters: enable/disable vibration (true: enable, false: disable) Return value: true: set successfully, false: set failed
boolean	**isPromptSoundEnable()** Enable sound or not? Return value: true: enable, false: disable
boolean	**isPromptVibrateEnable()** Enable vibration or not? Return value: true: enable, false: disable

Example:

//get UHFManager instance

UHFManagermUHFMgr = UHFManager.getInstance();

//----set inventory notification

booleansound = true;//sound

booleanvibrate = true;//vibrate

booleansuc = mUHFMgr.setPromptSoundEnable(sound);

suc = mUHFMgr.setPromptVibrateEnable(vibrate);

//----get current notification status

booleanisSound = mUHFMgr.isPromptSoundEnable();

booleanisVibrate = mUHFMgr.isPromptVibrateEnable();

5. Access Control

Description: You can read, write and lock tags.

- Read: Read data in a specified memory bank, e.g. read EPC ID in EPC memory bank.
- Write: Write data to a specified memory bank, e.g. write access password to the Reserved Memory Bank.
- Lock: Lock data in a specified memory bank. Enter access password to read or over-write data (Set access password before performing Lock operation).

  Lock options：

  - Temporary Lock: Lock the selected memory bank temporarily for reading and writing.
  - Permanent Lock: Lock the selected memory bank permanently for reading and writing.

Interface definition:

Class: com.nlscan.android.uhf.UHFManager	
Return Value	Method
static UHFManager	**getInstance()** To get instance

byte[]	**GetTagData(int bank,** **int address,** **intblkcnt,** **String hexAccesspasswd)** To read data in a specified memory bank Parameters: Types of Memory Banks(@See UHFReader.BANK_TYPE) Reserved memory bank: UHFReader.BANK_TYPE.RESERVED EPC memory bank: UHFReader.BANK_TYPE.EPC TID memory bank: UHFReader.BANK_TYPE.TID USER memory bank: UHFReader.BANK_TYPE.USER Parameter: start address, unit: block (16bit in a block) Parameter: blkcnt (Number of blocks to read) Parameter: hexAccesspasswd. If access password is required, (4-byte hexadecimal string, e.g. (0x): 0D0A0305, the length should be a multiple of two), enter it （4-byte data）. If not required, pass NULL. Return value: byte[], Byte read.
UHFReader.READER_STATE	**writeTagData(int bank,** **int address,** **byte[] data,** **String hexAccesspasswd)** To write data to each memory bank Parameters: Types of Memory Banks(@See UHFReader.BANK_TYPE) Reserved memory bank: UHFReader.BANK_TYPE.RESERVED EPC memory bank: UHFReader.BANK_TYPE.EPC TID memory bank: UHFReader.BANK_TYPE.TID USER memory bank: UHFReader.BANK_TYPE.USER Parameter: start address, unit: block (16bit in a block) Parameter: read-in data (Byte arrays) Parameter: hexAccesspasswd. If access password is required, (4-byte hexadecimal string, e.g. (0x): 0D0A0305, the length should be a multiple of two), enter it （4-byte data）. If not required, pass NULL. Return value: (@See UHFReader.READER_STATE)
UHFReader.READER_STATE	**writeTagEpcEx(byte[] epcData, String hexAccesspasswd)** To write data to EPC memory bank (EPC memory bank is usually used to store tag initialization info. Locked EPC memory bank is not writable.) Parameter: read-in epcData (Byte arrays) Parameter: hexAccesspasswd. If access password is required, (4-byte hexadecimal string, e.g. (0x): 0D0A0305, the length should be a multiple of two), enter it （4-byte data）. If not required, pass NULL. Return value: (@See UHFReader.READER_STATE)

UHFReader.READER_STATE	**LockTag(intlockObjects,** **intlockTypes,** **String hexAccesspasswd)** To lock data in a specified memory bank Memory areas of a tag can be locked: kill password, access password, EPC (bank 1), TID (bank 2), USER (bank 3). Set access password none-zero before data lock. Access password is required to lock data. To unlock/temporary lock/permanent lock/ data in several areas at the same time, call the function just once. Parameter: lockObject lock area, parameter (@See UHFReader.Lock_Obj) (Select several areas and use "|" to connect.) Parameter: LockType lock type, parameter (@See UHFReader.Lock_Type) （Select several types based on selected area and use "|" to connect.） Parameter: hexAccesspasswd If access password is required, (4-byte hexadecimal string, e.g. (0x): 0D0A0305, the length should be a multiple of two), enter it（4-byte data）. If not required, pass NULL. Return value: (@See UHFReader.READER_STATE)

Example:

//get UHFManager instance

UHFManagermUHFMgr = UHFManager.getInstance();

//----Read tag data

int bank = UHFReader. BANK_TYPE. EPC. value();//EPC

intstartAddr = 0;//starting address from block 0th

intblkcnt = 2 ;// read two blocks, that is to read 4 bytes.

String hexAccesspasswd = "00000001";//access password（hexadecimal string）, if no access password entered, set null.

byte[] rdata = mUHFMgr.GetTagData(bank, startAddr, blkcnt, sHexPasswd);

//----Write tag data

int bank = UHFReader. BANK_TYPE. EPC. value();//EPC

intstartAddr = 0;// starting address from block 0th

byte[]data = UHFReader.Str2Hex("27099801201000011301");//read-in data（hexadecimal change to

byte[]）

String sHexPasswd = "00000001";// access password（hexadecimal string）, if no access password entered, set null.

UHFReader.READER_STATEer=mUHFMgr.writeTagData(bank, startAddr, data,

```
                                                    sHexPasswd);
if( er == UHFReader.READER_STATE. OK_ERR)
        //success......
else
        //fail......
```

//----Directly write tag EPC data

```
byte[]data = UHFReader.Str2Hex("27099801201000011301");//data（hexadecimal change to byte[]）
String sHexPasswd = "00000001";// access password（hexadecimal string）, if no access password entered, set null.
UHFReader.READER_STATEer = mUHFMgr.writeTagEpcEx(data, sHexPasswd);
if( er == UHFReader.READER_STATE. OK_ERR)
        //success......
else
        //fail......
```

//----Lock/Unlock data

//Temporary lock EPC

```
UHFReader.Lock_Objlobj =UHFReader.Lock_Obj.LOCK_OBJECT_BANK1; //EPC
UHFReader.Lock_Typeltyp=UHFReader.Lock_Type.LOCK; //Temporary lock
String sHexPasswd = "00000001";// access password（hexadecimal string）
UHFReader.READER_STATEer = mUHFMgr.LockTag(lobj.value(), ltyp.value(),
                                                    sHexPasswd);
if( er == UHFReader.READER_STATE. OK_ERR)
        //success......
else
        //fail.....
```

//Unlock EPC

```
UHFReader.Lock_Objlobj =UHFReader.Lock_Obj.LOCK_OBJECT_BANK1; //EPC;
UHFReader.Lock_Typeltyp=UHFReader.Lock_Type.UNLOCK; //Unlock
String sHexPasswd = "00000001";// access password（hexadecimal string）
UHFReader.READER_STATEer = mUHFMgr.LockTag(lobj.value(), ltyp.value(), sHexPasswd);
if( er == UHFReader.READER_STATE. OK_ERR)
        //success......
else
        //fail.....
```

6. Inventory

Description: Start or stop tag inventory. Keep reading and sending tag data after tag inventory starts.

Interface Definition:

- **Start/Stop Tag Inventory**

Class: com.nlscan.android.uhf.UHFManager	
Return Value	Method
static UHFManager	**getInstance()** To get instance
UHFReader.READER_STATE	**startTagInventory()** To start tag inventory Return value: (@See **UHFReader.READER_STATE**)
UHFReader.READER_STATE	**stopTagInventory()** To stop tag inventory Return value: (@See**UHFReader.READER_STATE**)

- **Get Tag Inventory Data: Register and listen for tag inventory data.**

Broadcast Action: nlscan.intent.action.uhf.ACTION_RESULT		
Extra Parameter Field	Data Type	Description
tag_info	Parcelable[]	Each element of tag arrays can be converted to object TagInfo, then get tag data like EPC ID from TagInfo.
extra_start_reading_time	long	Time to start reading（ms）: Calculate reading duration according to this start time.

Example:

//get UHFManager instance

UHFManagermUHFMgr = UHFManager.getInstance();

//----start tag inventory

UHFReader.READER_STATEer = mUHFMgr.startTagInventory();

if(er == UHFReader.READER_STATE. OK_ERR) //success

.....

else //fail

.....

```
//----stop tag inventory
UHFReader.READER_STATEer =mUHFMgr.stopTagInventory();
if( er == UHFReader.READER_STATE. OK_ERR) //success
    ......
else //fail
    ......


//----get tag inventory data
//Register and listen
IntentFilteriFilter = new IntentFilter("nlscan.intent.action.uhf.ACTION_RESULT ");
mContext.registerReceiver(mResultReceiver, iFilter);

//receiving data
privateBroadcastReceivermResultReceiver = new BroadcastReceiver() {

        @Override
        public void onReceive(Context context, Intent intent)
        {
            String action = intent.getAction();
            if(!"nlscan.intent.action.uhf.ACTION_RESULT ".equals(action))
                return ;
            //Tag arrays
            Parcelable[] tagInfos =    intent.getParcelableArrayExtra("tag_info");
            //Time to start reading
            longstartReading = intent.getLongExtra("extra_start_reading_time", 0l);
            //......
            for(Parcelable parcel : tagInfos)
            {
                TagInfotagInfo = (TagInfo)parcel;
                Log.d("TAG","Epc ID : "+UHFReader.bytes_Hexstr(tagInfo.EpcId));
            }
        }//end onReceiver
    };
```

7. Parameters Setting

Set UHF module parameters like inventory parameters, antenna power parameters, and extra parameters. Reading and writing tag performance vary with different parameters.

Different UHF models may have different parameters. Pease refer to corresponding UHF Configuration Guide.

Current model that supports UHF module: **UR90.**

Interface Definition:

Class: com.nlscan.android.uhf.UHFManager	
Return Value	Method
static UHFManager	**getInstance()** To get instance
UHFReader.READER_STATE	**setParam(String paramKey,** **String paramName,** **String sValue)** Parameters setting Parameter: paramKey function identifier (Please refer to corresponding model for parameter identifier.) Parameter: paramName parameter name (Please refer to corresponding model for parameter identifier.) Parameter: sValue, Return value: (@See **UHFReader.READER_STATE**)
Map<String ,Object>	**getAllParams()** To get all parameters from returned Map. Return value: Map<String ,Object>

Example of UR90:

```
//get UHFManager instance
UHFManagermUHFMgr = UHFManager.getInstance();


//----set antenna power
JSONArrayjsAntArray = new JSONArray();
try {
    JSONObjectjobj = new JSONObject();
    jobj.put("antid", 1);//antenna ID
    jobj.put("readPower", 2700);//read power
    jobj.put("writePower", 2000);//write power
    jsAntArray.put(jobj);
} catch (Exception e) {
}
String sAntPowerValue = jsAntArray.toString();
String paramKey = "RF_ANTPOWER";
```

```
String paramName = "PARAM_RF_ANTPOWER";
UHFReader.READER_STATEEr = mUHFMgr.setParam(paramKey, paramName, sAntPowerValue);
if( er == UHFReader.READER_STATE. OK_ERR)
        //success......
else
        //fail.....
```

//----get antenna power

```
//load “reader transmit power JSONArray ”
//[{"antid":1,"readPower":2600,"writePower":2700},...] format
Map<String,Object>settingsMap = mUHFMgr.getAllParams();
String key = "RF_ANTPOWER";
String sJsonValue = (String) settingsMap.get(key);
JSONArrayjsArray = new JSONArray(sValue);
intlen = jsArray.length();
for(inti =0 ;i<len;i++)
{
        JSONObjectjobj = jsArray.optJSONObject(i);
        intantid = jobj.optInt("antid");//Antenna ID
        short readPower = (short)jobj.optInt("readPower");//read power
        short writePower = (short)jobj.optInt("writePower");//write power
}
```

Association Classes

1. UHFReader

It provides string-processing method and different types of enumeration.

Class: **com.nlscan.android.uhf.UHFReader**

Universal Method:

Return Value	Method
static String	**bytes_Hexstr(byte[] bArray)** To convert bytes to hexadecimal strings Parameter: bArray (byte array) Return Value: hexadecimal strings e.g.(0x): 0A02

static byte[]	**Str2Hex(String hexStr)** To convert hexadecimal strings to binary. This conversion only supports hexadecimal string length within 600 characters. Parameter: hexStr hexadecimal strings (The length of string should be a multiple of two, as two characters convert to one byte and each character respectively represents the first 4 bits and the last 4 bits in one byte. e.g.: (0x):0A02) Return value: byte array
static String	**Hex2Str(byte[] buf)** To convert bytes to hexadecimal strings Parameter: buf (byte buffer) Return value: hexadecimal strings, e.g.: (0x):0A02

Interfaces Calling Returned State Enumeration:

Enumeration：UHFReader.READER_STATE	
Member	Description
OK_ERR	Operate successfully.
IO_ERR	Network or serial connection failed. Power off reader and reinitialize it.
INTERNAL_DEV_ERR	Internal device issue. Power off reader and reinitialize it.
CMD_FAILED_ERR	Operation failed.
CMD_NO_TAG_ERR	No tag detected.
OP_NOT_SUPPORTED	Operation is not supported.
INVALID_PARA	Invalid parameters.
INVALID_READER_HANDLE	Invalid reader parameters.
HARDWARE_ALERT_ERR_BY_NO_ANTENNAS	No antenna detected.
HARDWARE_ALERT_ERR_BY_HIGH_TEMPERATURE	Reader temperature is too high.
HARDWARE_ALERT_ERR_BY_UNKNOWN_ERR	Unknown error.
OP_EXECING	Operation is currently executing.
UNKNOWN_READER_TYPE	Unknown reader type.

OP_INVALID	Invalid operation.

Enumeration of Tag Memory Banks:

Enumeration: UHFReader.BANK_TYPE	
Member	Description
RESERVED	Reserved memory bank (stores access password and kill password).
EPC	EPC memory bank (stores EPC code)
TID	TID memory bank (tag ID). It is globally unique serial number.
USER	USER memory bank. It is extendable and Different brand tags have different memory capacities. And many tags do not have USER memory bank.

Enumeration of Tag Lock Area:

Enumeration: UHFReader.Lock_Obj	
Member	Description
LOCK_OBJECT_KILL_PASSWORD	Lock kill password area
LOCK_OBJECT_ACCESS_PASSWD	Lock access password area
LOCK_OBJECT_BANK1	Lock EPC memory bank
LOCK_OBJECT_BANK2	Lock TID memory bank
LOCK_OBJECT_BANK3	Lock USER memory bank

Enumeration of Lock Types:

Enumeration: HFReader.Lock_Obj	
Member	Description
UNLOCK	Unlock
LOCK	Temporary lock (Available to unlock)
PERM_LOCK	Permanent lock (NOT available to unlock)

2. UHFCommonParams

It provides common configuration parameters like parameters of inventory trigger.

Class: **com.nlscan.android.uhf.UHFCommonParams**

Class of Inventory Trigger Common Parameter:

Class: UHFCommonParams.TRIGGER_MODE	
Member	Description
TRIGGER_MODE_MAIN	Trigger mode – Main SCAN key
TRIGGER_MODE_LEFT	Trigger mode – Left SCAN key
TRIGGER_MODE_RIGHT	Trigger mode – Right SCAN key
TRIGGER_MODE_BACK	Trigger mode – Background SCAN key

3. TagInfo

Class of tag info includes data read from tag like EPC ID, RSS (signal strength) and etc.

Class: com.nlscan.android.uhf.TagInfo		
Member	Type	Description
AntennaID	byte	Which antenna read the tag?
Frequency	int	Which frequency read the tag?
TimeStamp	int	Time stamp (unit ms) that identified when tag was read (Relative to the moment that command was issued.).
EmbededData	byte[]	Embedded data(extra data)
EpcId	byte[]	EPC code
PC	byte[]	PC
CRC	byte[]	CRC
ReadCnt	int	Number of times that tag has been read.
RSSI	int	Received tag signal strength

protocol	SL_TagProtocol	Tag Protocol (Enumeration)
		TagInfo.SL_TagProtocol{
		SL_TAG_PROTOCOL_NONE(0),
		SL_TAG_PROTOCOL_ISO180006B(3),
		SL_TAG_PROTOCOL_GEN2(5),
		SL_TAG_PROTOCOL_ISO180006B_UCODE(6),
		SL_TAG_PROTOCOL_IPX64(7),
		SL_TAG_PROTOCOL_IPX256(8);
		}

Newland Auto-ID Tech. Co., Ltd.
(Headquarters)
3F, Building A, No.1, Rujiang West Rd., Mawei, Fuzhou, Fujian, China 350015
Tel: +86 - (0) 591-83978605
Fax: +86 - (0) 591-83979216
E-mail: contact@nlscan.com
Web: www.newlandaidc.com

Newland Europe BV
Rolweg 25, 4104 AV Culemborg, The Netherlands
Tel: +31 (0) 345 87 00 33
Fax: +31 (0) 345 87 00 39
Email: info@newland-id.com
Web: www.newland-id.com
Tech Support: tech-support@newland-id.com

Newland North America Inc.
46559 Fremont Blvd., Fremont, CA 94538, USA
Tel: 510 490 3888
Fax: 510 490 3887
Email: info@newlandna.com
Web: www.newlandamerica.com

Newland Latin America
Tel: +1 (239) 598 0068
Fax: +1 (239) 280 1238
Email: info@newlandla.com
Web: www.newlandamerica.com

Newland Taiwan Inc.
7F-6, No. 268, Liancheng Rd., Jhonghe Dist. 235, New Taipei City, Taiwan
Tel: +886 2 7731 5388
Fax: +886 2 7731 5389
Email: info@newland-id.com.tw
Web: www.newland-id.com.tw

Newland Korea
Biz. Center Best-one, Jang-eun
Medical Plaza 6F, Bojeong-dong 1261-4,
Kihung-gu, Yongin-City, Kyunggi-do, South Korea
Tel: +82 10 8990 4838
Fax: +82 70 4369 0009
Email: th.sung@newland-id.com.tw
Web: www.newlandaidc.com/kor/