



**C4 UHF RFID Handheld
Software Development Manual
Android UHF API**

iDTRONIC GmbH
Donnersbergweg 1
67059 Ludwigshafen
Germany/Deutschland

Phone: +49 621 6690094-0
Fax: +49 621 6690094-9
E-Mail: info@idtronic.de
Web: idtronic-rfid.com

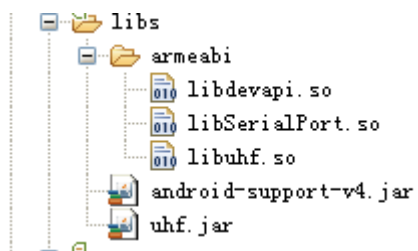
Issue 0.3
– 15. November 2017 –

Subject to alteration without prior notice.
© Copyright iDTRONIC GmbH 2017
Printed in Germany

Contents

1	Packages and libraries into engineering	4
2	Function Description	5
2.1	get UhfReader sample getInstance()	5
2.2	setOutputPower(int value)	5
2.3	Multi-label inventory inventoryMulti()	5
2.4	Stop multi-label inventory stopInventoryMulti()	5
2.5	Real-time inventory inventoryRealTime()	5
2.6	Select label selectEPC(byte[] epc)	5
2.7	Deselect the label unSelectEPC()	5
2.8	Reading the label data readFrom6C(int memBank, int startAddr, int length, byte[] accessPassword)	5
2.9	Write label data: writeTo6C(byte[] password, int memBank, int startAddr, int dataLen, byte[] data)	6
2.10	Adjust the operating frequency setFrequency(int startFrequency, int freqSpace, int freqQuality)	6
2.11	Set the region frequency setWorkArea(int area)	6
3	Error Codes	7

1 Packages and libraries into engineering



UHF Function is encapsulated in `uhf.jar` `com.android.hdhe.uhf.reader.Uhfreader`

The functions described in the following are declared in this file:

```
\SDK\UhfSDKdemo_EU\src\com\android\hdhe\uhf\reader\UhfReader.java
```

2 Function Description

2.1 get UhfReader sample getInstance()

Function: Get UhfReader instance, call UHF package instructions

Parameters: None

Returns: UhfReader instance

2.2 setOutputPower(int value)

Function: Set the transmitter function (read the label to adjust the distance)

Parameters: int value, value range of 16...26 corresponding to 16...26 dBm RF TX power

Returns: boolean flag, true set successfully, false set up failure

2.3 Multi-label inventory inventoryMulti()

Features: chip inventory multiple calls for multi-tag environment (10000 inventory is automatically stopped, and if call again)

Parameters: None

Returns: List <byte []> list, list is not null, the returned inventory list, list is null, no label or not the antenna area.

2.4 Stop multi-label inventory stopInventoryMulti()

Function: Stop multi-label inventory

Parameters: None

Returns: None

2.5 Real-time inventory inventoryRealTime()

Features: Real-time inventory, as a single instruction is executed when the cycle is called inventory can achieve multi-label inventory

Parameters: None

Returns: List <byte []> list, list is not null, the returned inventory list, list is null, no label or not the antenna area.

2.6 Select label selectEPC(byte[] epc)

Function: Select the label, read the label before, you should call.

Parameters: byte [] epc, EPC tag data

Returns: None

2.7 Deselect the label unSelectEPC()

Function: deselect the label, select the label or in the inventory should be re-called.

Parameters: None

Returns: int, for 0:00 properly executed, others wrong.

2.8 Reading the label data readFrom6C(int memBank, int startAddr, int length, byte[] accessPassword)

Function: Read the label data storage area, the data word is read out as a unit.

Parameters: int memBank store RESEVER zone 0, EPC District 1, TID District 2, USER District 3;

int startAddr starting address (not too large, depending on the size of the data area);

int length read data length, in units of word (1word = 2bytes); byte [] accessPassword password 4 bytes

Returns: byte [] data; data not null, a data area data, or null to read data failed to return an error code byte

2.9 Write label data: `writeTo6C(byte[] password, int memBank, int startAddr, int dataLen, byte[] data)`

Function: write tag data storage area (TID district can not write), write data to the word as a unit, not padded with zeros.

Parameters: `byte [] Password` Password 4 bytes

`int memBank` store RESEVER zone 0, EPC District 1, TID District 2, USER District 3;

`int startAddr` starting address (not too large, depending on the size of the data area);

`int dataLen` read data length, in units of word (1word = 2bytes); `byte [] data`

Returns: `boolean`, true success is written, false write failure

2.10 Adjust the operating frequency `setFrequency(int startFrequency, int freqSpace, int freqQuality)`

Function: Adjust the operating frequency (not recommended).

Parameters: `int startFrequency` initial frequency

`i int freqSpace` frequency interval

`int freqQuality` frequency points;

Returns: `int`, if returns 0, the setting is successful, others fail

2.11 Set the region frequency `setWorkArea(int area)`

Function: Set the regional frequency (recommended).

Parameters: `int area`;

`area = 1` China 2, 2 American Standard, 3 European standard, 4 China 1, 5 Korea

Returns: `int`, if returns 0, the setting is successful, others fail

3 Error Codes

Type	Code	Description
Command Error	0x17	Command frame instruction code error.
FHSS Fail	0x20	Search channel hopping out. All channels are occupied during that time.
Inventory Fail	0x15	Polling operation failed. No tag return data return or CRC errors.
Access Fail	0x16	Access tab fails, there may be accessed using the password is wrong.
Read Fail	0x09	Reading the label data stored several area failed. Tag does not return or return data CRC errors
Read Error	0xA0	Reading the label data storage area error. The return code obtained from 0xA0 bit or Error Code. Error code information in the table below.
Write Fail	0x10	Write tag data store several area failed. Tag does not return or return data CRC checksum error.
Write Error	0xB0	Write tag data store error. The return code obtained from 0xB0 bit or Error Code. Error code information in the table below.
Lock Fail	0x13	Lock tag data store several area failed. Tag does not return or return data CRC checksum error.
Lock Error	0xC0	Lock tag data store error. The return code obtained from 0xC0 bit or Error Code. Error code information in the table below.
Kill Fail	0x12	Inactivated label failed. Tag does not return or return data CRC checksum error.
Kill Error	0xD0	Inactivated label error. The return code obtained from 0xC0 bit or Error Code. Error code information in the table below.

NXP G2X label instructions unique error code:

Type	Code	Description
ReadProtect Fail	0x2A	ReadProtect command fails, the tag return data returns no data or CRC errors.
Reset ReadProtect Fail	0x2B	Reset ReadProtect command fails, the tag return data returns no data or CRC errors.
Change EAS Fail	0x1B	Change EAS command fails, the tag return data returns no data or CRC errors.
The error code returned by NXP unique label instructions	0xE0	The error code returned by NXP unique label instructions, the error code obtained from 0xE0 or on the label returned Error Code.

EPC Gen2 protocol label returns an error code (Tag Error Codes):

Types	Code	Error code Name	Description
Error-specific	00000000 ₂	Other error	All other errors in this table have not declared
	00000011 ₂	Memory overrun	Specify the tag data store does not exist; or the labels do not support a specified length EPC, such as XPC.
	00000100 ₂	Memory locked	Specified label data storage area is locked and / or are permanently locked, and locked state is not writable or readable
	00001011 ₂	Insufficient power	Tags do not receive enough energy to be written
Non-specific	00001111 ₂	Non-specific error	tag don't support Error-code return