



**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

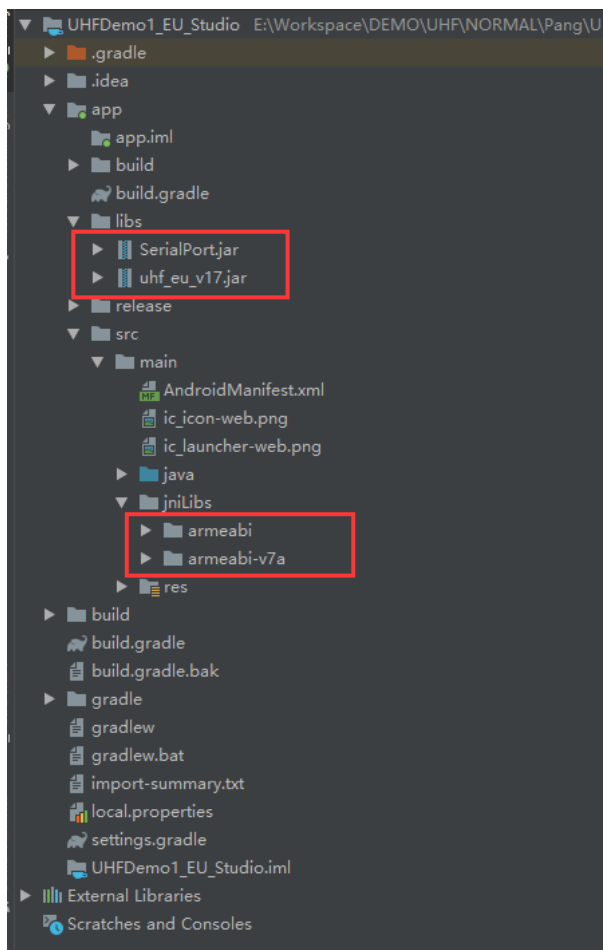
Issue 0.4
– 3. December 2019 –

Subject to alteration without prior notice.
© Copyright iDTRONIC GmbH 2019
Printed in Germany

Contents

1	Copy .jar and .so to your project:	4
2	Call UhfReader.class methods:	5
2.1	UHF-ISO1800-6C	5
2.1.1	getInstance()	5
2.1.2	getFirmware()	5
2.1.3	setOutputPower()	6
2.1.4	inventoryRealTime()	6
2.1.5	selectEPC()	6
2.1.6	readFrom6C()	6
2.1.7	writeTo6C()	7
2.1.8	setWorkArea().....	7
2.1.9	getFrequency()	7
2.1.10	setFrequency()	8
2.1.11	setAutoFre ()	8
2.1.12	lock6C ()	8
2.1.13	close().....	9
3	Error Codes.....	10

1 Copy .jar and .so to your project:



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

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

2 Call UhfReader.class methods :

1.1. UHF-ISO1800-6C UhfReader

2.1.1	getInstance()
2.1.2	getFirmware()
2.1.3	setOutputPower()
2.1.4	inventoryRealTime()
2.1.5	selectEPC()
2.1.6	readFrom6C()
2.1.7	writeTo6C()
2.1.8	setWorkArea()
2.1.9	getFrequency()
2.1.10	setFrequency()
2.1.11	setAutoFre()
2.1.12	lock6C()
2.1.13	close()

2.1 UHF-ISO1800-6C

2.1.1 getInstance()

Function	UhfManager getInstance()
Description	Get Uhf manager instance and open device
Parameter	void
Return	Uhf manager instance

2.1.2 getFirmware()

Function	byte[] getFirmware()
Description	Get firmvare version
Parameter	void
Return	Version information

2.1.3 setOutputPower()

Function	boolean setOutputPower(int value)
Description	Set antenna power gain (The module will automatically set the power to the maximum each time it is powered on)
Parameter	Power value
Return	True: set success False: set failure

Note: Scope of value, 16 - 30.

2.1.4 inventoryRealTime()

Function	List<byte[]> inventoryRealTime()
Description	Real-time get uhf epc list
Parameter	void
Return	Epc list

2.1.5 selectEPC()

Function	void selectEPC(byte[] epc)
Description	Select epc(Before you can read or write a label, you need to call this method and select the label)
Parameter	epc
Return	void

2.1.6 readFrom6C()

Function	byte[] readFrom6C(int memBank, int startAddr, int length, byte[] accessPassword)
Description	Read tag data
Parameter	Membank, 0--reserve, 1--epc, 2--tid, 3--user start address, unit: word read length, unit: word password, access password
Return	Tag data

2.1.7 writeTo6C()

Function	boolean writeTo6C(byte[] password, int memBank, int startAddr, int dataLen, byte[] data)
Description	Write data into iso-1800-6c tag
Parameter	Password, access password Membank, 0--reserve, 1--epc, 2--tid, 3--user start address, unit: word data length, unit: word data
Return	True: write success False: write failure

2.1.8 setWorkArea()

Function	int setWorkArea(int area)
Description	Set device work area
Parameter	Work area 1->China2, 2->USA, 3->Europe, 4->China1, 6->Korea
Return	0:set success Other: set failure

Note: American standard by default. Generally do not need to be modified.

2.1.9 getFrequency()

Function	int getFrequency()
Description	Get module work frequency
Parameter	void
Return	Frequency

2.1.10 setFrequency()

Function	int setFrequency(int startFrequency, int freqSpace, int freqQuality)
Description	Set frequency(Not recommended) Before calling this function, you need to call setWorkArea first, set the working area, and then set the frequency.
Parameter	Start Frequency, the starting frequency frequency Space, frequency interval frequency Quality, frequency points
Return	0:set success Other: set failure

Note:

Chinese band2: $F_s = 920.125 + N * 0.25$ (MHz), $N \in [0, 19]$.

US band: $F_s = 902.75 + N * 0.5$ (MHz), $N \in [0, 49]$.

Korean band: $F_s = 917.1 + N * 0.2$ (MHz), $N \in [0, 31]$.

EU band: $F_s = 865.1 + N * 0.2$ (MHz), $N \in [0, 14]$.

Chinese band1: $F_s = 840.125 + N * 0.25$ (MHz), $N \in [0, 19]$.

2.1.11 setAutoFre ()

Function	boolean setAutoFre(boolean isOpen)
Description	Enable or disable FHSS mode
Parameter	isOpen : true--open FHSS false--close FHSS
Return	true success false set failure

2.1.12 lock6C ()

Function	boolean lock6C(byte[] password, int memBank, int lockType)
Description	Lock the tag.
Parameter	password, access password memBank, lock area 0-- kill password, 1-- access password, 2--epc, 3--tid, 4--user lockType, 0—open, 1—open forever, 2—lock, 3—lock forever
Return	0:set success Other: set fail

Note: 1. After permanent opening and permanent locking, the tag will not be able to change the lock type.

2. To do the locking, you need to lock the kill password and access password areas first.

2.1.13 close()

Function	void close()
Description	Close device
Parameter	void
Return	void

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