



UHF Demo API

C9 Red UHF

iDTRONIC GmbH
Ludwig-Reichling-Straße 4
67059 Ludwigshafen
Germany/Deutschland

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

Issue 1.01
– 07. October 2025 –

Subject to alteration without prior notice.
© Copyright iDTRONIC GmbH 2025
Printed in Germany

Contents

1	Library Import	4
2	Library Methods	5
2.1	getInstance.....	5
2.2	getHardware	5
2.3	asyncStartReading.....	5
2.4	asyncStopReading	5
2.5	setInventoryFilter	5
2.6	setCancelInventoryFilter	5
2.7	tagInventoryReadTime	5
2.8	tagEpcTidInventoryByTimer	6
2.9	stopTagInventory	6
2.10	tagInventoryByTimer()	6
2.11	getTagData	6
2.12	getTagDataByFilter	6
2.13	writeTagData.....	7
2.14	writeTagDataByFilter.....	7
2.15	writeTagEPC	7
2.16	writeTagEPCByFilter	7
2.17	lockTag	8
2.18	lockTagByFilter	8
2.19	killTag.....	8
2.20	killTagByFilter.....	8
2.21	setRegion	9
2.22	getRegion	9
2.23	getFrequencyPoints.....	9
2.24	setFrequencyPoints	9
2.25	setPower	9
2.26	getPower	9
2.27	setFastMode	9
2.28	getTemperature	9
2.29	close	10

1 Library Import

To use the libraries in a project, copy the library files to the directories as shown in the figure below. The `targetSdkVersion` (parameter in the module's `build.gradle` file) should not be lower than 26.

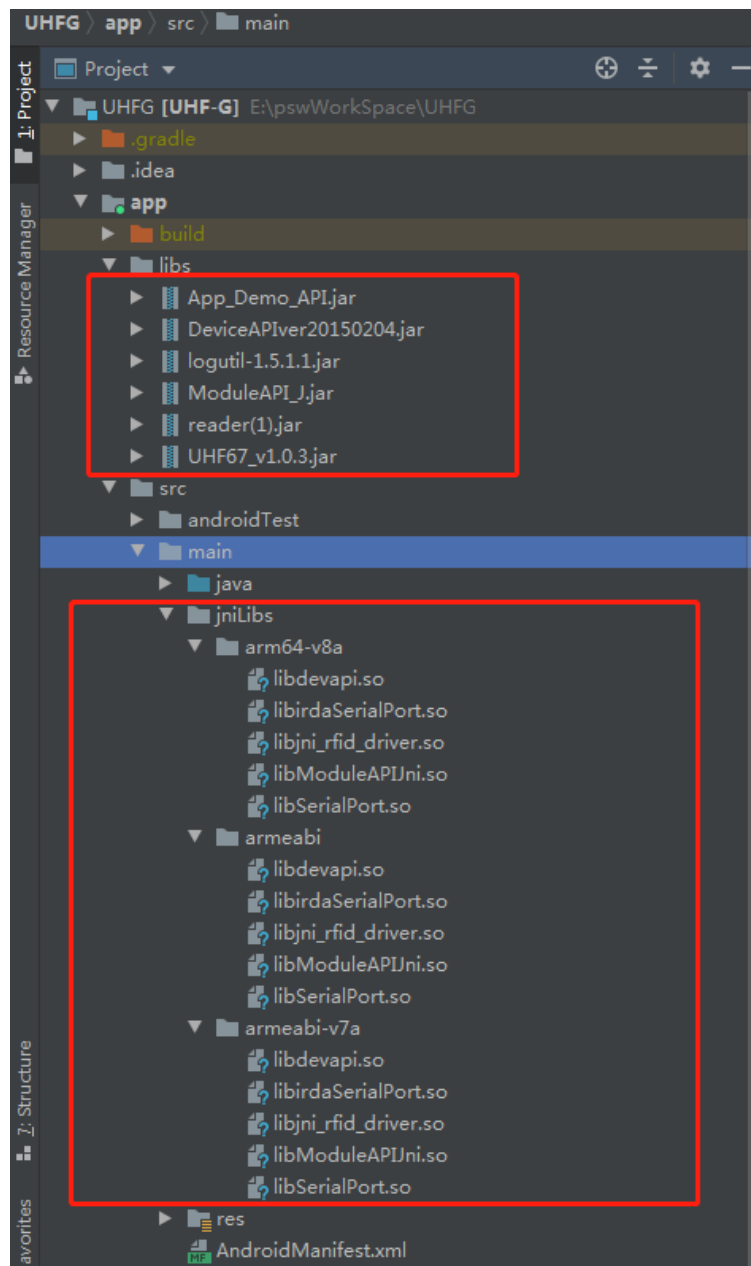


Figure 1: Screenshot Android Studio; Project library includes

2 Library Methods

The methods of the UHFManager class are the following:

2.1 getInstance

Method	UHFManager getInstance()
Description	Get a UHF class instance and open the hardware device
Parameters	Null
Return	UHFManager instance UHF class instance

2.2 getHardware

Method	String getHardware()
Description	Get hardware version information
Parameters	Null
Return	Hardware information string; if Null, the execution failed

2.3 asyncStartReading

Method	READER_ERR asyncStartReading()
Description	Start multi-mode inventory
Parameters	Null
Return	READER_ERR.MT_OK_ERR: success; else error message

2.4 asyncStopReading

Method	READER_ERR asyncStopReading()
Description	Stop multi-mode inventory
Parameters	Null
Return	READER_ERR.MT_OK_ERR: success; else error message

2.5 setInventoryFilter

Method	boolean setInventoryFilter(byte[] fdata, int fbank, int fstartaddr, boolean matching)	
Description	Set tag inventory filter	
Parameters	byte[] fdata	filter data
	int fbank	Filter data bank: 1 = EPC, 2 = TID, 3 = USER
	int fstartaddr	Start address (in words = 2 bytes)
	boolean matching	Match filter or not
Return	True: operation successful	

2.6 setCancelInventoryFilter

Method	boolean setCancelInventoryFilter()
Description	Remove the tag inventory filter that was set with setInventoryFilter()
Parameters	Null
Return	True: operation successful

2.7 tagInventoryReadTime

Method	List<TAGINFO> tagInventoryRealTime()
---------------	--------------------------------------

Description	Get list of tags, use after asyncStartReading()
Parameters	Null
Return	List<TAGINFO>: list of tags, else Null

2.8 tagEpcTidInventoryByTimer

Method	List<TAGINFO> tagEpcTidInventoryByTimer()
Description	get tags EPC and TID list
Parameters	Null
Return	List<TAGINFO>: list of tags, else Null

2.9 stopTagInventory

Method	boolean stopTagInventory()
Description	Used to stop inventory
Parameters	Null
Return	True: operation successful

2.10 tagInventoryByTimer()

Method	List<TAGINFO> tagInventoryByTimer(short readtime)
Description	Timed inventory
Parameters	Short readtime Time of single inventory (in ms)
Return	List<TAGINFO>: list of tags, else Null

2.11 getTagData

Method	READER_ERR getTagData(int mbank, int startaddr, int len, byte[] rdata, byte[] password, short timeout)												
Description	Get tag data / read tags												
Parameters	<table border="1"> <tr> <td>int mbank</td><td>tag area to read, 0 = RESERVED, 1 = EPC, 2 = TID, 3 = USER</td></tr> <tr> <td>int startaddr</td><td>Start address to read (in words = 2 bytes)</td></tr> <tr> <td>int len</td><td>Data length to read</td></tr> <tr> <td>byte[] rdata</td><td>Used to save the read data, length is len</td></tr> <tr> <td>byte[] password</td><td>Access password (4 bytes)</td></tr> <tr> <td>short timeout</td><td>Read timeout (in ms)</td></tr> </table>	int mbank	tag area to read, 0 = RESERVED, 1 = EPC, 2 = TID, 3 = USER	int startaddr	Start address to read (in words = 2 bytes)	int len	Data length to read	byte[] rdata	Used to save the read data, length is len	byte[] password	Access password (4 bytes)	short timeout	Read timeout (in ms)
int mbank	tag area to read, 0 = RESERVED, 1 = EPC, 2 = TID, 3 = USER												
int startaddr	Start address to read (in words = 2 bytes)												
int len	Data length to read												
byte[] rdata	Used to save the read data, length is len												
byte[] password	Access password (4 bytes)												
short timeout	Read timeout (in ms)												
Return	READER_ERR.MT_OK_ERR: operation successful; else error message												

2.12 getTagDataByFilter

Method	byte[] getTagDataByFilter(int mbank, int startaddr, int len, byte[] rdata, byte[] password, short timeout, byte[] fdata, int fbank, int fstartaddr, boolean matching)																		
Description	Get tag data / read tags through filter																		
Parameters	<table border="1"> <tr> <td>int mbank</td><td>tag area to read, 0 = RESERVED, 1 = EPC, 2 = TID, 3 = USER</td></tr> <tr> <td>int startaddr</td><td>Start address to read (in words = 2 bytes)</td></tr> <tr> <td>int len</td><td>Data length to read</td></tr> <tr> <td>byte[] rdata</td><td>Used to save the read data, length is len</td></tr> <tr> <td>byte[] password</td><td>Access password (4 bytes)</td></tr> <tr> <td>short timeout</td><td>Read timeout (in ms)</td></tr> <tr> <td>byte[] fdata</td><td>Filter data</td></tr> <tr> <td>int fbank</td><td>Filter bank, 1 = EPC, 2 = TID, 3 = USER</td></tr> <tr> <td>int fstartaddr</td><td>Filter start address (in words = 2 bytes)</td></tr> </table>	int mbank	tag area to read, 0 = RESERVED, 1 = EPC, 2 = TID, 3 = USER	int startaddr	Start address to read (in words = 2 bytes)	int len	Data length to read	byte[] rdata	Used to save the read data, length is len	byte[] password	Access password (4 bytes)	short timeout	Read timeout (in ms)	byte[] fdata	Filter data	int fbank	Filter bank, 1 = EPC, 2 = TID, 3 = USER	int fstartaddr	Filter start address (in words = 2 bytes)
int mbank	tag area to read, 0 = RESERVED, 1 = EPC, 2 = TID, 3 = USER																		
int startaddr	Start address to read (in words = 2 bytes)																		
int len	Data length to read																		
byte[] rdata	Used to save the read data, length is len																		
byte[] password	Access password (4 bytes)																		
short timeout	Read timeout (in ms)																		
byte[] fdata	Filter data																		
int fbank	Filter bank, 1 = EPC, 2 = TID, 3 = USER																		
int fstartaddr	Filter start address (in words = 2 bytes)																		

	boolean matching	Match filter or not
Return	Byte[]: operation successful; else Null	

2.13 writeTagData

Method	READER_ERR writeTagData(int mbank, int startaddress, byte[] data, int datalen, byte[] password, short timeout)	
Description	Write data to tags	
Parameters	int mbank	tag area to write, 0 = RESERVED, 1 = EPC, 2 = TID, 3 = USER
	int startaddress	Start address to write (in words = 2 bytes)
	byte[] data	Data to write
	int datalen	Data length to write (in words = 2 bytes)
	byte[] password	Access password (4 bytes)
	short timeout	Write timeout (in ms)
Return	READER_ERR.MT_OK_ERR: operation successful; else error message	

2.14 writeTagDataByFilter

Method	READER_ERR writeTagDataByFilter(int mbank, int startaddress, byte[] data, int datalen, byte[] password, short timeout, byte[] fdata, int fbank, int fstartaddr, boolean matching)	
Description	Write data to tags through filter	
Parameters	int mbank	tag area to read, 0 = RESERVED, 1 = EPC, 2 = TID, 3 = USER
	int startaddress	Start address to write (in words = 2 bytes)
	byte[] data	Data to write
	int datalen	Data length to write (in words = 2 bytes)
	byte[] password	Access password (4 bytes)
	short timeout	Write timeout (in ms)
	byte[] fdata	Filter data
	int fbank	Filter bank, 1 = EPC, 2 = TID, 3 = USER
	int fstartaddr	Filter start address (in words = 2 bytes)
	boolean matching	Match filter or not
Return	READER_ERR.MT_OK_ERR: operation successful; else error message	

2.15 writeTagEPC

Method	READER_ERR writeTagEPC(byte[] data, byte[] accesspwd, short timeout)	
Description	Write EPC of a tag	
Parameters	byte[] data	EPC data to write
	byte[] accesspwd	Access password (4 bytes)
	short timeout	Read timeout (in ms)
Return	READER_ERR.MT_OK_ERR: operation successful; else error message	

2.16 writeTagEPCByFilter

Method	READER_ERR writeTagEPCByFilter(byte[] data, byte[] accesspwd, short timeout, byte[] fdata, int fbank, int fstartaddr, boolean matching)	
Description	Write EPC of a tag through filter	
Parameters	byte[] data	EPC data to write
	byte[] accesspwd	Access password (4 bytes)
	short timeout	Read timeout (in ms)

	byte[] fdata	Filter data
	int fbank	Filter bank, 1 = EPC, 2 = TID, 3 = USER
	int fstartaddr	Filter start address (in words = 2 bytes)
	boolean matching	Match filter or not
Return	READER_ERR.MT_OK_ERR: operation successful; else error message	

2.17 lockTag

Method	READER_ERR lockTag(Lock_Obj lockobject, Lock_Type locktype, byte[] accesspwd, short timeout)	
Description	Lock tags	
Parameters	Lock_Obj lockobject	Area to lock, please refer to the demo source code
	Lock_Type locktype	Lock type, please refer to the demo source code
	byte[] accesspwd	Access password (4 bytes)
	short timeout	Operation timeout (in ms)
Return	READER_ERR.MT_OK_ERR: operation successful; else error message	

2.18 lockTagByFilter

Method	READER_ERR lockTagByFilter(Lock_Obj lockobject, Lock_Type locktype, byte[] accesspwd, short timeout, byte[] fdata, int fbank, int fstartaddr, boolean matching)	
Description	Lock tags through filter	
Parameters	Lock_Obj lockobject	Area to lock, please refer to the demo source code
	Lock_Type locktype	Lock type, please refer to the demo source code
	byte[] accesspwd	Access password (4 bytes)
	short timeout	Operation timeout (in ms)
	byte[] fdata	Filter data
	int fbank	Filter bank, 1 = EPC, 2 = TID, 3 = USER
	int fstartaddr	Filter start address (in words = 2 bytes)
	boolean matching	Match filter or not
Return	READER_ERR.MT_OK_ERR: operation successful; else error message	

2.19 killTag

Method	READER_ERR killTag(byte[] killpasswd, short timeout)	
Description	Kill tags, killed tags cannot be read anymore, please operate with caution (password != 0 must be set)	
Parameters	byte[] killpasswd	Kill password (4 bytes; != 0)
	short timeout	Operation timeout (in ms)
Return	READER_ERR.MT_OK_ERR: operation successful; else error message	

2.20 killTagByFilter

Method	READER_ERR killTag(byte[] killpasswd, short timeout, byte[] fdata, int fbank, int fstartaddr, boolean matching)	
Description	Kill tags, killed tags cannot be read anymore, please operate with caution (password != 0 must be set)	
Parameters	byte[] killpasswd	Kill password (4 bytes; != 0)
	short timeout	Operation timeout (in ms)
	byte[] fdata	Filter data
	int fbank	Filter bank, 1 = EPC, 2 = TID, 3 = USER
	int fstartaddr	Filter start address (in words = 2 bytes)
	boolean matching	Match filter or not
Return	READER_ERR.MT_OK_ERR: operation successful; else error message	

2.21 setRegion

Method	READER_ERR setRegion(Region_Conf region)	
Description	Set the region (changes frequency band of the device to be consistent with regional regulations)	
Parameters	Region_Conf region	Frequency band region, including China, Europe, United States
Return	READER_ERR.MT_OK_ERR: operation successful; else error message (not supported)	

2.22 getRegion

Method	Region_Conf getRegion()	
Description	Get current region	
Parameters	Null	
Return	Region_Conf, please refer to demo source code	

2.23 getFrequencyPoints

Method	int[] getFrequencyPoints()	
Description	Get all frequency points of the current frequency band	
Parameters	Null	
Return	int[] frequency point array (in kHz)	

2.24 setFrequencyPoints

Method	READER_ERR setFrequencyPoints(int[] frequencyPoints)	
Description	Set frequency points, only points that match the current region are allowed	
Parameters	int[] frequencyPoints	Frequency point array (in kHz)
Return	READER_ERR.MT_OK_ERR: operation successful; else error message	

2.25 setPower

Method	READER_ERR setPower(int readPower, int writePower)	
Description	Set the RF power of the device	
Parameters	int readPower	Read power, range 5..30
	int writePower	Write power, range 5..30
Return	READER_ERR.MT_OK_ERR: operation successful; else error message	

2.26 getPower

Method	int[] getPower()	
Description	Get the RF power of the device	
Parameters	Null	
Return	int[0] read power, int[1] write power; null if the operation failed	

2.27 setFastMode

Method	READER_ERR setFastMode()	
Description	Set multi-mode /Fast-mode, the power is then at its maximum	
Parameters	Null	
Return	READER_ERR.MT_OK_ERR: operation successful; else error message	

2.28 getTemperature

Method	int getTemperature()	
---------------	----------------------	--

Description	Get the temperature of the UHF reader chip
Parameters	Null
Return	int > 0: operation successful, else fail

2.29 close

Method	boolean close()
Description	Close UHF device hardware power supply, disconnect data connection
Parameters	Null
Return	True: operation successful