[1. Intro 1](#_Toc13145277)

[2. Function Description 2](#_Toc13145278)

[2.1 Open Serial Port 2](#_Toc13145279)

[2.2 Close Serial Port: 2](#_Toc13145280)

[2.3 TCP Connection 2](#_Toc13145281)

[2.4 TCP Disconnection 2](#_Toc13145282)

[2.5 Hardware Version Acquire 3](#_Toc13145283)

[2.6 Firmware Version Acquire 3](#_Toc13145284)

[2.7 Device ID Acquire 3](#_Toc13145285)

[2.8 Setup Output Power 3](#_Toc13145286)

[2.9 Current Output Power Acquire 4](#_Toc13145287)

[2.10 Frequency Fixing Setup 4](#_Toc13145288)

[2.11 Current Fixed Frequency Acquire 4](#_Toc13145289)

[2.12 Setup Gen2 Parameter 5](#_Toc13145290)

[2.13 Current Gen2 Parameter Setup Acquire 7](#_Toc13145291)

[2.14 CW Setup 10](#_Toc13145292)

[2.15 Current CW Setup Acquire 10](#_Toc13145293)

[2.16 Antenna Setup 10](#_Toc13145294)

[2.17 Acquire Antenna Current Setup of Device 11](#_Toc13145295)

[2.18 Frequency Area Setup 11](#_Toc13145296)

[2.19 Acquire Frequency Area 11](#_Toc13145297)

[2.20 Acquire Frequency Area 12](#_Toc13145298)

[2.21 Setup Temperature Protection 12](#_Toc13145299)

[2.22 Acquire Temperature Protection Infor 12](#_Toc13145300)

[2.23 Setup Antenna Working Time 12](#_Toc13145301)

[2.24 Acquire Antenna Work Time 13](#_Toc13145302)

[2.25 Setup RF link 13](#_Toc13145303)

[2.26 Acquire RF Link Setup 14](#_Toc13145304)

[2.27 Setup FastID Function 14](#_Toc13145305)

[2.28 Acquire FastID Status 14](#_Toc13145306)

[2.29 Setup Tagfocus Function 15](#_Toc13145307)

[2.30 Acquire Tagfocus Status 15](#_Toc13145308)

[2.31 Reset 15](#_Toc13145309)

[2.32 Tag Filtering Setup 15](#_Toc13145310)

[2.33 EPC+TID, EPC+TID+USER Mode Setup 16](#_Toc13145311)

[2.34 Read EPC+TID, EPC+TID+USER Mode 16](#_Toc13145312)

[2.35 Restore Factory Data 17](#_Toc13145313)

[2.36 Read Tag (Single) 18](#_Toc13145314)

[2.37 Read Tag (Multiple) 18](#_Toc13145315)

[2.38 Stop Multiple Tags Reading 18](#_Toc13145316)

[2.39 Get Data of Multiple Tags Reading 18](#_Toc13145317)

[2.40 Read Data Areas 19](#_Toc13145318)

[2.41 Write Data Areas 20](#_Toc13145319)

[2.42 LOCK Tag 20](#_Toc13145320)

[2.43 KILL Tag 21](#_Toc13145321)

[2.44 Block Write Data 22](#_Toc13145322)

[2.45 Block Erase Data 22](#_Toc13145323)

[2.46 QT Parameter Setup 23](#_Toc13145324)

[2.47 Get QT Parameter 24](#_Toc13145325)

[2.48 QT Tag Reading 25](#_Toc13145326)

[2.49 QT Tag Writing 26](#_Toc13145327)

[2.50 Block Permalock 27](#_Toc13145328)

[2.51 Setup Protocol Type 28](#_Toc13145329)

[2.52 Get Protocol Type 28](#_Toc13145330)

[2.53 LOCK Tag (GB) 28](#_Toc13145331)

[2.54 Relay and IO Control Output 29](#_Toc13145332)

[2.55 Get Output Setup Status of Relay and IO Control 30](#_Toc13145333)

# Introduction

This document is used to explain functions in the UHFAPI.dll library. By using this DLL library, users could develop applications under the Windows system.

# Function Description

## Open Serial Port

|  |  |
| --- | --- |
| Function | int ComOpen (int port); |
| Description | Open serial port. Default: 115200 baud rate, 8-bit, 1 stop bit, no check bit, no hardware control. |
| Parameter | port: serial port number |
| Return | 0: success others: failure |

## Close Serial Port:

|  |  |
| --- | --- |
| Function | void ClosePort (); |
| Description | Close serial port |
| Parameter | NULL |
| Return | NULL |

## TCP Connection

|  |  |
| --- | --- |
| Function | int TCPConnect (const char \* hostaddr, int hostport); |
| Description | TCP connection |
| Parameter | hostaddr: host address  hostport: host serial port number |
| Return | 0: success other: failure |

## TCP Disconnection

|  |  |
| --- | --- |
| Function | void TCPDisconnect (); |
| Description | TCP disconnection |
| Parameter | NULL |
| Return | NULL |

## Hardware Version Acquire

|  |  |
| --- | --- |
| Function | int UHFGetHardwareVersion (unsigned char \*version); |
| Description | Acquire hardware version |
| Parameter | version:  version[0], length  version[1...x], version number |
| Return | 0: success other: failure |

## Firmware Version Acquire

|  |  |
| --- | --- |
| Function | int UHFGetSoftwareVersion (unsigned char \*version); |
| Description | Acquire firmware version |
| Parameter | version:  version[0], length  version[1...x], version number |
| Return | 0: Success Others: Failure |

## Device ID Acquire

|  |  |
| --- | --- |
| Function | int UHFGetDeviceID (unsigned int \*id); |
| Description | Acquire device ID |
| Parameter | id: device ID (integer ID number) |
| Return | 0: Success Others: Failure |

## Setup Output Power

|  |  |
| --- | --- |
| Function | int UHFSetPower ( unsigned char saveflag, unsigned char uPower); |
| Description | Setup output power depends on different antenna |
| Parameter | saveflag: 1, power fail save  0, power fail unsaving  uPower: power (DB) |
| Return | 0: Success Others: Failure |

## Current Output Power Acquire

|  |  |
| --- | --- |
| Function | int UHFGetPower (unsigned char \*uPower); |
| Description | Acquire current output power |
| Parameter | uPower: power (DB) |
| Return | 0: Success Others: Failure |

## Frequency Fixing Setup

|  |  |
| --- | --- |
| Function | int UHFSetJumpFrequency ( unsigned char nums, unsigned int \*Freqbuf); |
| Description | Setup fixed frequency, 1 frequency point supported. Deafault frequency fixing point is 1, Freq (1) means frequency point, unit is KHz. |
| Parameter | nums: frequency points number  Freqbuf: frequency points group (integer)  (for example: 920125, 921250......) |
| Return | 0: Success Others: Failure |

## Current Fixed Frequency Acquire

|  |  |
| --- | --- |
| Function | int UHFGetJumpFrequency ( unsigned int \*Freqbuf); |
| Description | Acquire current device fixed frequency and frequency fixing list |
| Parameter | Freqbuf[0]: frequency fixing number  Freqbuf[1...x]: frequency fixing points (integer)  (fox example: 920125, 921250......) |
| Return | 0: Success Others: Failure |

## Setup Gen2 Parameter

|  |  |
| --- | --- |
| Function | int UHFSetGen2 (unsigned char Target,  unsigned char Action, unsigned char T,  unsigned char Q, unsigned char StartQ,  unsigned char MinQ, unsigned char MaxQ,  unsigned char D, unsigned char C,  unsigned char P, unsigned char Sel,  unsigned char Session, unsigned char G,  unsigned char LF); |
| Description | Setup Gen2 parameter |
| Parameter | Target: Target parameter of select command   |  |  | | --- | --- | | 0 | S0 | | 1 | S1 | | 2 | S2 | | 3 | S3 | | 4 | SL |   Action: Action parameter of select command   |  |  |  | | --- | --- | --- | |  | Matching | Non-Matching | | 0 | assert SL or inventoried →A | de-assert SL or inventoried → B | | 1 | assert SL or inventoried → A | do nothing | | 2 | do nothing | de-assert SL or inventoried → B | | 3 | negate SL or (A → B, B → A) | do nothing | | 4 | de-assert SL or inventoried → B | de-assert SL or inventoried → A | | 5 | de-assert SL or inventoried → B | do nothing | | 6 | do nothing | de-assert SL or inventoried → A | | 7 | do nothing | negate SL or (A → B, B → A) |   T: Truncate parameter of select   |  |  | | --- | --- | | 0 | Disable truncation | | 1 | Enable truncation |   Q:   |  |  | | --- | --- | | 0 | Fixed Q algorithm | | 1 | Dynamic Q algorithm |   Note: Under fixed Q algorithm, Q needs to be fixed as startQ, ignore MinQ and MaxQ.  startQ: 0~15  MinQ: 0~15  MaxQ: 0~15  D: DR parameter of query command   |  |  | | --- | --- | | 0 | 8 | | 1 | 64/3 |   C: M parameter of query command   |  |  | | --- | --- | | 0 | FM0 | | 1 | Miller2 | | 2 | Miller4 | | 3 | Miller8 |   P: TRext parameter of query command   |  |  | | --- | --- | | 0 | No pilot | | 1 | Use pilot |   Sel: sel parameter of query command   |  |  | | --- | --- | | 0 | All | | 1 | All | | 2 | ~SL | | 3 | SL |   Session: Session parameter of query command   |  |  | | --- | --- | | 0 | S0 | | 1 | S1 | | 2 | S2 | | 3 | S3 |   G: Target parameter of query command   |  |  | | --- | --- | | 0 | A | | 1 | B |   LF: Link Frequency setup:   |  |  | | --- | --- | | 0 | 40KHz | | 1 | 160KHz | | 2 | 200KHz | | 3 | 250KHz | | 4 | 300KHz | | 5 | 320KHz | | 6 | 400KHz | | 7 | 640KHz | |
| Return | 0: Success Others: Failure |

## Current Gen2 Parameter Setup Acquire

|  |  |
| --- | --- |
| Function | int UHFGetGen2 (unsigned char \*Target,  unsigned char \*Action, unsigned char \*T,  unsigned char \*Q, unsigned char \*StartQ,  unsigned char \*MinQ, unsigned char \*MaxQ,  unsigned char \*D, unsigned char \*Coding,  unsigned char \*P, unsigned char \*Sel,  unsigned char \*Session, unsigned char \*G,  unsigned char \*LF); |
| Description | Acquire Gen2 parameter |
| Parameter | Target: Target parameter of select command   |  |  | | --- | --- | | 0 | S0 | | 1 | S1 | | 2 | S2 | | 3 | S3 | | 4 | SL |   Action: Action parameter of select   |  |  |  | | --- | --- | --- | |  | Matching | Non-Matching | | 0 | assert SL or inventoried →A | de-assert SL or inventoried → B | | 1 | assert SL or inventoried → A | do nothing | | 2 | do nothing | de-assert SL or inventoried → B | | 3 | negate SL or (A → B, B → A) | do nothing | | 4 | de-assert SL or inventoried → B | de-assert SL or inventoried → A | | 5 | de-assert SL or inventoried → B | do nothing | | 6 | do nothing | de-assert SL or inventoried → A | | 7 | do nothing | negate SL or (A → B, B → A) |   T: Truncate parameter of select command   |  |  | | --- | --- | | 0 | Disable truncation | | 1 | Enable truncation |   Q:   |  |  | | --- | --- | | 0 | Fixed Q algorithm | | 1 | Dynamic Q algorithm |   Note: In fixed Q algorithm, Q needs to be fixed as StartQ, ignore MinQ and MaxQ.  startQ: 0~15  MinQ: 0~15  MaxQ: 0~15  D: DR parameter of query command   |  |  | | --- | --- | | 0 | 8 | | 1 | 64/3 |   C: M parameter of query command   |  |  | | --- | --- | | 0 | FM0 | | 1 | Miller2 | | 2 | Miller4 | | 3 | Miller8 |   P: TRext parameter of query command   |  |  | | --- | --- | | 0 | No pilot | | 1 | Use pilot |   Sel: sel parameter of query command   |  |  | | --- | --- | | 0 | All | | 1 | All | | 2 | ~SL | | 3 | SL |   Session: session parameter of query command   |  |  | | --- | --- | | 0 | S0 | | 1 | S1 | | 2 | S2 | | 3 | S3 |   G: Target parameter of query command   |  |  | | --- | --- | | 0 | A | | 1 | B |   LF: Link Frequency Setup:   |  |  | | --- | --- | | 0 | 40KHz | | 1 | 160KHz | | 2 | 200KHz | | 3 | 250KHz | | 4 | 300KHz | | 5 | 320KHz | | 6 | 400KHz | | 7 | 640KHz | |
| Return | 0: Success Others: Failure |

## CW Setup

|  |  |
| --- | --- |
| Function | int UHFSetCW ( unsigned char flag); |
| Description | CW setup |
| Parameter | flag: 1, open CW  0, close CW |
| Return | 0: Success Others: Failure |

## Current CW Setup Acquire

|  |  |
| --- | --- |
| Function | int UHFGetCW ( unsigned char \*flag); |
| Description | CW status acquire |
| Parameter | flag: 1, Open CW  0, Close CW |
| Return | 0: Success Others: Failure |

## Antenna Setup

|  |  |
| --- | --- |
| Function | int UHFSetANT ( unsigned char saveflag,  unsigned char \*buf); |
| Description | Antenna setup, setup antenna of device, default antenna port is 1, cannot setup other antenna number. |
| Parameter | saveflag: 1, power-fail saving  0, power-fail unsaving  buf: 2bytes, 16bits in total,  If bit is 1, select according antenna.  If bit value is 0, deselect according antenna.  buf[0]: Ant16~Ant9  buf[1]: Ant8~Ant1 |
| Return | 0: Success Others: Failure |

## Acquire Antenna Current Setup of Device

|  |  |
| --- | --- |
| Function | int UHFGetANT ( unsigned char \*buf); |
| Description | Acquire antenna setup information and antenna number of current device. |
| Parameter | buf: 2 bytes, 16 bits, if one bit value is 1, select according antenna. If one bit value is 0, deselect according antenna.  buf[0]: Ant16~Ant9  buf[1]: Ant8~Ant1 |
| Return | 0: Success Others: Failure |

## Frequency Area Setup

|  |  |
| --- | --- |
| Function | int UHFSetRegion ( unsigned char saveflag,  unsigned char region); |
| Description | Frequency area setup |
| Parameter | saveflag: 1, power-fail saving  0, power-fail unsaving  region:   |  |  | | --- | --- | | 0x01 | China1 | | 0x02 | China2 | | 0x04 | Europe | | 0x08 | USA | | 0x16 | Korea | | 0x32 | Japan | |
| Return | 0: Success Others: Failure |

## Acquire Frequency Area

|  |  |
| --- | --- |
| Function | int UHFGetRegion ( unsigned char \*region); |
| Description | Acquire frequency area setup |
| Parameter | region:   |  |  | | --- | --- | | 0x01 | China1 | | 0x02 | China2 | | 0x04 | Europe | | 0x08 | USA | | 0x16 | Korea | | 0x32 | Japan | |
| Return | 0: Success Others: Failure |

## Acquire Frequency Area

|  |  |
| --- | --- |
| Function | int UHFGetTemperature ( unsigned int \*temperature); |
| Description | Acquire current temp. of device |
| Parameter | temperature: temp.value (integer) |
| Return | 0: Success Others: Failure |

## Setup Temperature Protection

|  |  |
| --- | --- |
| Function | int UHFSetTemperatureProtect ( unsigned char flag); |
| Description | Setup Temp. protection |
| Parameter | flag: 1, Temp. protection  0, No temp. protection |
| Return | 0: Success Others: Failure |

## Acquire Temperature Protection Infor

|  |  |
| --- | --- |
| Function | int UHFGetTemperatureProtect ( unsigned char \*flag); |
| Description | Acquire temp. protection infor |
| Parameter | flag: 1, temp.protection  0, No temp.protection |
| Return | 0: Success Others: Failure |

## Setup Antenna Working Time

|  |  |
| --- | --- |
| Function | int UHFSetANTWorkTime (unsigned char antnum,  unsigned char saveflag, unsigned int WorkTime); |
| Description | Setup antenna working time |
| Parameter | antnum: antenna number, Range: 1-16  saveflag: 1, power-fail saving  0, power-fail unsaving  WorkTime: work time, unit: ms, range: 10-65535ms |
| Return | 0: Success Others: Failure |

## Acquire Antenna Work Time

|  |  |
| --- | --- |
| Function | int UHFGetANTWorkTime (unsigned char antnum,  unsigned int \*WorkTime); |
| Description | Acquire antenna work time |
| Parameter | antnum: antenna number, range: 1-16  Note: single antenna module, acquire antenna 1 only, multiple-antenna module will support acquisition of other antenna.  WorkTime: work time, unit: ms, Range: 10-65535ms |
| Return | 0: Success Others: Failure |

## Setup RF link

|  |  |
| --- | --- |
| Function | int UHFSetRFLink ( unsigned char saveflag,  unsigned char mode); |
| Description | Setup RF link |
| Parameter | saveflag: 1, power-fail saving  0, power-fail unsaving  mode: 0: DSB\_ASK/FM0/40KHZ  1: PR\_ASK/Miller4/250KHZ  2: PR\_ASK/Miller4/300KHZ  3: DSB\_ASK/FM0/400KHZ |
| Return | 0: Success Others: Failure |

## Acquire RF Link Setup

|  |  |
| --- | --- |
| Function | int UHFGetRFLink (unsigned char\* uMode); |
| Description | Acquire RF link setup |
| Parameter | uMode: 0: DSB\_ASK/FM0/40KHZ  1: PR\_ASK/Miller4/250KHZ  2: PR\_ASK/Miller4/300KHZ  3: DSB\_ASK/FM0/400KHZ |
| Return | 0: Success Others: Failure |

## Setup FastID Function

|  |  |
| --- | --- |
| Function | int UHFSetFastID (unsigned char flag); |
| Description | Setup FastID function |
| Parameter | flag: 1, ON  0, OFF |
| Return | 0: Success Others: Failure |

## Acquire FastID Status

|  |  |
| --- | --- |
| Function | int UHFGetFastID (unsigned char \*flag); |
| Description | Acquire FastID function infor |
| Parameter | flag: 1, ON  0, OFF |
| Return | 0: Success Others: Failure |

## Setup Tagfocus Function

|  |  |
| --- | --- |
| Function | int UHFSetTagfocus (unsigned char flag); |
| Description | Setup Tagfocus function |
| Parameter | flag: 1, ON  0, OFF |
| Return | 0: Success Others: Failure |

## Acquire Tagfocus Status

|  |  |
| --- | --- |
| Function | int UHFGetTagfocus (unsigned char \*flag); |
| Description | Acquire Tagfocus infor |
| Parameter | flag: 1, ON  0, OFF |
| Return | 0: Success Others: Failure |

## Reset

|  |  |
| --- | --- |
| Function | int UHFSetSoftReset (void); |
| Description | Software reset |
| Parameter | NULL |
| Return | 0: Success Others: Failure |

## Tag Filtering Setup

|  |  |
| --- | --- |
| Function | int UHFSetFilter (unsigned char saveflag,  unsigned char bank, unsigned int startaddr,  unsigned int datalen, unsigned char \*databuf); |
| Description | Tag filtering setup, tag group range selection during tag filtering process. |
| Parameter | saveflag: 1, power-fail saving  0, power-fail unsaving  bank: 0x01: EPC , 0x02: TID, 0x03: USR  startaddr: start address, Unit: byte  datalen: data length, Unit: byte  databuf: data |
| Return | 0: Success Others: Failure |

## EPC+TID, EPC+TID+USER Mode Setup

|  |  |
| --- | --- |
| Function | int UHFSetEPCTIDMode (unsigned char saveflag,  unsigned char mode); |
| Description | Setup EPC and TID mode |
| Parameter | saveflag: 1, power-fail saving  0, power-fail unsaving  mode:  0x00, means OFF；  0x01, means switch ON EPC+TID mode (default address: 0x00, length: 6 words ) ；  0x02, means switch ON EPC+TID+USER mode |
| Return | 0: Success Others: Failure |

## Read EPC+TID, EPC+TID+USER Mode

|  |  |
| --- | --- |
| Function | int UHFGetEPCTIDMode (unsigned char \*mode); |
| Description | Acquire EPC and TID mode |
| Parameter | mode:  0x00, OFF  0x01, switch ON EPC+TID mode (default address 0x00, length is 6 words)  0x02, switch ON EPC+TID+USER mode |
| Return | 0: Success Others: Failure |

## Restore Factory Data

|  |  |
| --- | --- |
| Function | int UHFSetDefaultMode (); |
| Description | Restore factory data |
| Parameter | NULL |
| Return | 0: Success Others: Failure |

Note: Send “Restore factory data”to restore default settings of reader. After restore factory data, reader will auto-reset by itself. All settings in list below will change:

|  |  |  |
| --- | --- | --- |
| Setup Parameters | After Restore Factory Data | Comments |
| Transmit Power | 30dBm |  |
| Temp.protection Setup | ON |  |
| RF link | PR \_ASK /Miller4/ 250KHz |  |
| Buzzer | OFF |  |
| FastID | OFF |  |
| TagFocus | OFF |  |
| Baud Rate | 115200 |  |
| Dual and Single | Dual |  |
| Filter Setup | Filter Data Length=0 | Disable Filter function in searching tags |
| EPC and TID Mode | OFF |  |

## Read Tag (Single)

|  |  |
| --- | --- |
| Function | int UHFInventorySingle (unsigned char\* uLenUii,  unsigned char\* uUii ); |
| Description | Read tag by single way |
| Parameter | uLenUii: uii length  uUii: uii data |
| Return | 0: Success Others: Failure |

## Read Tag (Multiple)

|  |  |
| --- | --- |
| Function | int UHFInventory (); |
| Description | Read tags continuously. Note: during multiple tag reading process, reader will not implement other commands, if user needs to execute other commands, stop command needs to be sent to stop multiple tags reading. |
| Parameter | NULL |
| Return | 0: Success Others: Failure |

## Stop Multiple Tags Reading

|  |  |
| --- | --- |
| Function | int UHFStopGet (); |
| Description | Stop multiple tags reading |
| Parameter | NULL |
| Return | 0: Success Others: Failure |

## Get Data of Multiple Tags Reading

|  |  |
| --- | --- |
| Function | int UHF\_GetReceived\_EX (int\* uLenUii,  unsigned char\* uUii); |
| Description | Get data of multiple tags |
| Parameter | uLenUii: uii length  uUii: uii data |
| Return | 0: Success Others: Failure |

## Read Data Areas

|  |  |
| --- | --- |
| Function | int UHFReadData (unsigned char\* uAccessPwd,  unsigned char FilterBank, unsigned int FilterStartaddr, unsigned int FilterLen, unsigned char \*FilterData,  unsigned char uBank, unsigned int uPtr,  unsigned int uCnt, unsigned char\* uReadDatabuf,  unsigned int\* uReadDataLen); |
| Description | Read data areas of tag, read data of specific data area of tag. |
| Parameter | uAccessPwd: 4 bytes password  FilterBank: enable bank number filtering, 1: EPC, 2: TID, 3: USR  FilterStartaddr: enable initial address of filtering, unit: bit  FilterLen: enable data length filtering, unit: bit  FilterData: enable data filtering  uBank: Read band of data  0x00: Reserved , 0x01: EPC, 0x02: TID, 0x03: USR  uPtr: Read initial address of data, unit: word  uCnt: Read data length, unit: word  uReadDatabuf: Data has been read  uReadDataLen: Data length has been read |
| Return | 0: Success Others: Failure |

## Write Data Areas

|  |  |
| --- | --- |
| Function | int UHFWriteData (unsigned char\* uAccessPwd,  unsigned char FilterBank, unsigned int FilterStartaddr, unsigned int FilterLen, unsigned char \*FilterData, unsigned char uBank, unsigned int uPtr,  unsigned char uCnt, unsigned char \*uWriteDatabuf); |
| Description | Write data area of tag, to write data into specific storage area |
| Parameter | uAccessPwd: 4 byte password  FilterBank: Enable band number of filter, 1: EPC, 2: TID, 3: USR  FilterStartaddr: Enable initial address of filter, unit: bit  FilterLen: Enable filtering length , unit: bit  FilterData: Enable data filtering  uBank: Write data bank:  0x00: Reserved , 0x01: EPC, 0x02: TID, 0x03: USR  uPtr: Write initial address of data, unit: word  uCnt: Write data length, unit: word  uWriteDatabuf: Data has been written |
| Return | 0: Success Others: Failure |

## LOCK Tag

|  |  |
| --- | --- |
| Function | bool UHFLockTag (unsigned char\* uAccessPwd,  unsigned char FilterBank, unsigned int FilterStartaddr, unsigned int FilterLen, unsigned char \*FilterData,  unsigned char \*lockbuf); |
| Description | LOCK tag, lock memory bank of tag |
| Parameter | uAccessPwd: 4 byte password  FilterBank: enable bank number of filtering, 1: EPC, 2: TID, 3: USR  FilterStartaddr: enable initial address of filtering, unit: bit  FilterLen: enable length of filtering, unit: bit  FilterData: enable data filtering  lockbuf: 3 byte, higher than 4bit (invalid), 0-9bit: Action,  10-19 bit: Mask |
| Return | 0: Success Others: Failure |

## KILL Tag

|  |  |
| --- | --- |
| Function | int UHFKillTag (unsigned char\* uAccessPwd,  unsigned char FilterBank, unsigned int FilterStartaddr, unsigned int FilterLen, unsigned char \*FilterData); |
| Description | KILL tag |
| Parameter | uAccessPwd: 4 byte password  FilterBank: enable bank number of filtering, 1: EPC, 2: TID, 3: USR  FilterStartaddr: enable initial address of filtering, unit: bit  FilterLen: enable filtering length, unit: bit  FilterData: enable data filtering |
| Return | 0: Success Others: Failure |

## Block Write Data

|  |  |
| --- | --- |
| Function | int UHFBlockWriteData (unsigned char\* uAccessPwd, unsigned char FilterBank, unsigned int FilterStartaddr, unsigned int FilterLen, unsigned char \*FilterData,  unsigned char uBank, unsigned int uPtr,  unsigned int uCnt, unsigned char \*uWriteDatabuf); |
| Description | Block Write: Specific data length to specific address of tag |
| Parameter | uAccessPwd: 4byte password  FilterBank: enable bank number of filtering, 1: EPC, 2: TID, 3: USR  FilterStartaddr: enable initial address of filtering, unit: bit  FilterLen: enable data length of filtering, unit: bit  FilterData: enable data filtering  uBank: Write bank of data:  0x00: Reserved , 0x01: EPC, 0x02: TID, 0x03: USR  uPtr: Write initial address of data, unit: word  uCnt: Write length of data, unit: word  uWriteDatabuf: Data has been written |
| Return | 0: Success Others: Failure |

## Block Erase Data

|  |  |
| --- | --- |
| Function | int UHFBlockEraseData (unsigned char\* uAccessPwd, unsigned char FilterBank, unsigned int FilterStartaddr, unsigned int FilterLen, unsigned char \*FilterData,  unsigned char uBank, unsigned int uPtr,  unsigned char uCnt); |
| Description | BlockErase: Specific length data to specific address of tag |
| Parameter | uAccessPwd: 4 byte password  FilterBank: enable bank number filtering, 1: EPC, 2: TID, 3: USR  FilterStartaddr: enable initial address of filtering, unit: bit  FilterLen: enable filtering data length, unit: bit  FilterData: enable data filtering  uBank: write band of data:  0x00: Reserved , 0x01: EPC, 0x02: TID, 0x03: USR  uPtr: write initial address of data, unit: word  uCnt: write data length, unit: word |
| Return | 0: Success Others: Failure |

## QT Parameter Setup

|  |  |
| --- | --- |
| Function | int UHFSetQT (unsigned char\* uAccessPwd,  unsigned char FilterBank, unsigned int FilterStartaddr, unsigned int FilterLen, unsigned char \*FilterData, unsigned char QTData); |
| Description | Setup QT parameters |
| Parameter | uAccessPwd: 4 bytes password  FilterBank: enable bank number filtering, 1: EPC, 2: TID, 3: USR  FilterStartaddr: enable initial address of filtering, unit: bit  FilterLen: enable filtering length, unit: bit  FilterData: enable data filtering  QTData:   |  |  | | --- | --- | | bit0 | 0: disable short distance control, 1: enable short distance control | | bit1 | 0: enable private memory map,  1: enable public memory map | | bit2~bit7 | reserved | |
| Return | 0: Success Others: Failure |

## Get QT Parameter

|  |  |
| --- | --- |
| Function | int UHFGetQT (unsigned char\* uAccessPwd,  unsigned char FilterBank, unsigned int FilterStartaddr, unsigned int FilterLen, unsigned char \*FilterData,  unsigned char \*QTData); |
| Description | Get QT parameters |
| Parameter | uAccessPwd: 4 byte passwords  FilterBank: enable bank number filtering, 1: EPC, 2: TID, 3: USR  FilterStartaddr: enable initial address filtering, unit: bit  FilterLen: enable filtering length, unit: bit  FilterData: enable data filtering  QTData:   |  |  | | --- | --- | | bit0 | 0: disable short distance control, 1: enable short distance control | | bit1 | 0: enable private Memory map,  1: enable public memory map | | bit2~bit7 | reserved | |
| Return | 0: Success Others: Failure |

## QT Tag Reading

|  |  |
| --- | --- |
| Function | int UHFReadQT (unsigned char\* uAccessPwd,  unsigned char FilterBank, unsigned int FilterStartaddr, unsigned int FilterLen, unsigned char \*FilterData,  unsigned char QTData, unsigned char uBank,  unsigned int uPtr, unsigned char uCnt,  unsigned char \*uReadDatabuf,  unsigned char \*uReadDataLen); |
| Description | Read data of private memory map by QT adjustment. After QT reading setup, tag will return memory map status before QT reading setup. |
| Parameter | uAccessPwd: 4 byte password  FilterBank: enable bank number filtering, 1: EPC, 2: TID, 3: USR  FilterStartaddr: enable initial address of filtering, unit: bit  FilterLen: enable filtering length, unit: bit  FilterData: enable data filtering  QTData:   |  |  | | --- | --- | | bit0 | 0: disable short distance control, 1: enable short distance control | | bit1~bit7 | reserved |   uBank: read bank of data  uPtr: read initial address of data, unit: word  uCnt: read data length, unit: word  uReadDatabuf: data has been read  uReadDataLen: data length has been read |
| Return | 0: Success Others: Failure |

## QT Tag Writing

|  |  |
| --- | --- |
| Function | int UHFWriteQT (unsigned char\* uAccessPwd,  unsigned char FilterBank, unsigned int FilterStartaddr, unsigned int FilterLen, unsigned char \*FilterData,  unsigned char QTData, unsigned char uBank,  unsigned int uPtr, unsigned char uCnt,  unsigned char \*uWriteDatabuf); |
| Description | Write data of private memory map by QT adjustment. After QT reading setup, tag will return memory map status before QT reading setup. |
| Parameter | uAccessPwd: 4 bytes password  FilterBank: enable filtering bank number, 1: EPC, 2: TID, 3: USR  FilterStartaddr: enable initial address of filtering, unit: bit  FilterLen: enable data length of filtering, unit: bit  FilterData: enable data filter  QTData:   |  |  | | --- | --- | | bit0 | 0 disable short distance control, 1 enable short distance control | | bit1~bit7 | reserved |   uBank: write bank of data  uPtr: write initial address of data, unit: word  uCnt: data length, unit: word  uWriteDatabuf: data needs to be written |
| Return | 0: Success Others: Failure |

## Block Permalock

|  |  |
| --- | --- |
| Function | int UHFBlockPermalock (unsigned char\* uAccessPwd, unsigned char FilterBank, unsigned int FilterStartaddr, unsigned int FilterLen, unsigned char \*FilterData,  unsigned char ReadLock, unsigned char uBank,  unsigned int uPtr, unsigned char uRange,  unsigned char \*uMaskbuf); |
| Description | BlockPermalock execution |
| Parameter | uAccessPwd: 4 bytes passwords  FilterBank: enable bank number of filtering, 1: EPC, 2: TID, 3: USR  FilterStartaddr: enable initial address of filtering, unit: bit  FilterLen: enable data length of filtering, unit: bit  FilterData: enable data filtering  ReadLock:   |  |  | | --- | --- | | bit0 | 0: Read, 1: Permalock | | bit1~bit7 | Reserved |   uBank: write bank of data  uPtr: Block initial address, unit: 16 blocks  uRange: Block range, unit: 16 blocks  uMaskbuf: mask data of block, 2 bytes, 16 bits correspond to 16 blocks for selection |
| Return | 0: Success Others: Failure |

## Setup Protocol Type

|  |  |
| --- | --- |
| Function | int UHFSetProtocolType ( unsigned char type); |
| Description | Setup protocol type |
| Parameter | type: 0x00 ISO18000-6C  0x01 GB/T 29768  0x02 GJB 7377.1 |
| Return | 0: Success Others: Failure |

## Get Protocol Type

|  |  |
| --- | --- |
| Function | int UHFGetProtocolType (unsigned char \*type); |
| Description | Get protocol type |
| Parameter | type: 0x00 ISO18000-6C  0x01 GB/T 29768  0x02 GJB 7377.1 |
| Return | 0: Success Others: Failure |

## LOCK Tag (GB)

|  |  |
| --- | --- |
| Function | int UHFGBTagLock (unsigned char\* uAccessPwd,  unsigned char FilterBank, unsigned int FilterStartaddr, unsigned int FilterLen, unsigned char \*FilterData,  unsigned char memory, unsigned char config,  unsigned char action); |
| Description | LOCK tag (GB) |
| Parameter | uAccessPwd: 4 bytes password  FilterBank: enable bank number filtering, 1: EPC, 2: TID, 3: USR  FilterStartaddr: enable initial address of filtering, unit: byte  FilterLen: enable filter length, unit: byte  FilterData: enable data filtering  Memory storage area: 0x00 information area of tag,  0x10 decoding area, 0x20 safety area,  0x3x user area  0x30-0x3F 0-5 in user area  config : 0x00 config storage area,  0x01 config safty mode  action:  config storage area: 0x00: read and write,  0x01: read only,  0x02: write only,  0x03: disable read&write  Config safety mode: 0x00: reserved, 0x01: no-need to verify, 0x02: need to verify, doesnot need safety communication, 0x03: need verify& safety communication |
| Return | 0: Success Others: Failure |

## Relay and IO Control Output

|  |  |
| --- | --- |
| Function | int UHFSetIOControl ( unsigned char output1, unsigned char output2 , unsigned char outStatus); |
| Description | Relay and IO control output setup |
| Parameter | output1: 0: low level 1: High level  output2: 0: low level 1: High level  outStatus: 0: OFF 1: ON |
| Return | 0: Success Others: Failure |

## Get Output Setup Status of Relay and IO Control

|  |  |
| --- | --- |
| Function | int UHFGetIOControl (unsigned char \*outData); |
| Description | Get output setup status of relay and IO control |
| Parameter | outData[0]: 0: low level 1: high level  outData[1]: 0: low level 1: high level |
| Return | 2: data length (success) -1: get failure |