RFID MODULE

UHF RFID Module

RT400

User Manual

Version 1.0
Nov. 2014
StrongLink

CONTENT

1. GENERAL DESCRIPTION

RT400 is a high-integrated and high-performance UHF RFID module based on our advanced analog circuit design and digital signal processing technology specified for RFID. It supports ISO 18000-6C/EPC C1 GEN2 protocol and offers a much longer operation distance than HF RFID system.

RT400 supports USB virtual COM port and can be operated via COM commands easily. There are also APIs on Windows and Windows Mobile, users can develop UHF RFID application swiftly. We also offer several different kinds of UHF antenna to be cooperated with RT400.

2. FEATURES

- USB[1] Interface
- 4.5 ~ 5.5V DC power supply via USB interface
- Protocol: ISO18000-6C
- Frequency: 840MHz~960MHz
- Power consumption(RF Pout=30dBm, 50ohm load):

  Normal Mode: 160mA

  Reading Mode[2]: 460mA

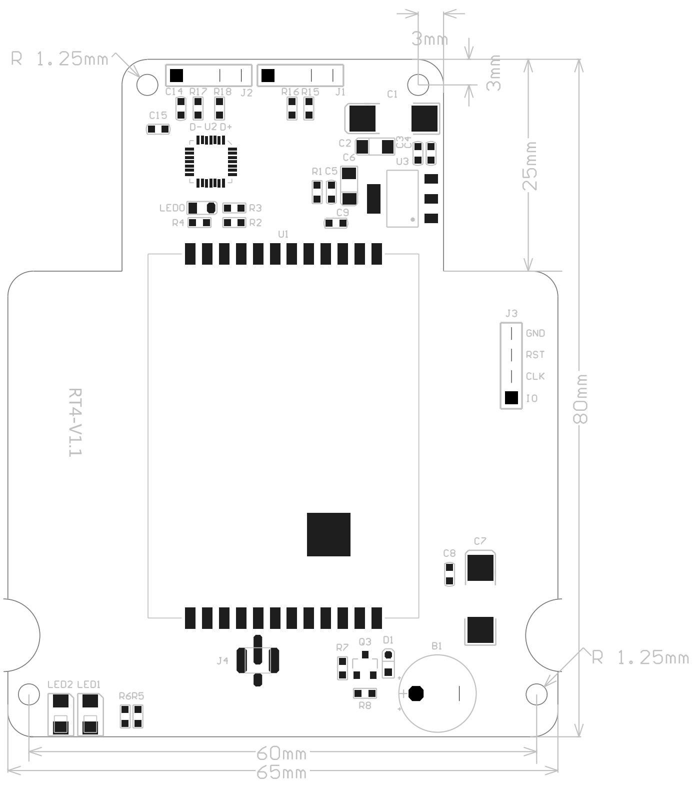
  Sleep Mode: 1.29mA

[1] This USB interface adopts USB CDC protocol as a virtual COM port.

[2] The current is different according to the different load. The current is tested under reading single tag in continuous condition.

- RF interface: IPX-female
- API compatible with Windows 32bit/64bit operation system
- Serial COM communication protocol available
- Operating Temperature Range: -20 ℃ ~ +50 ℃
- Storage Temperature Range: -40 ℃ ~ +85 ℃
- Dimension: 80 × 65 × 7 mm
- Weight: 80g(including 1.2 meters USB cable)

3. DIMENSION DESCRIPTION

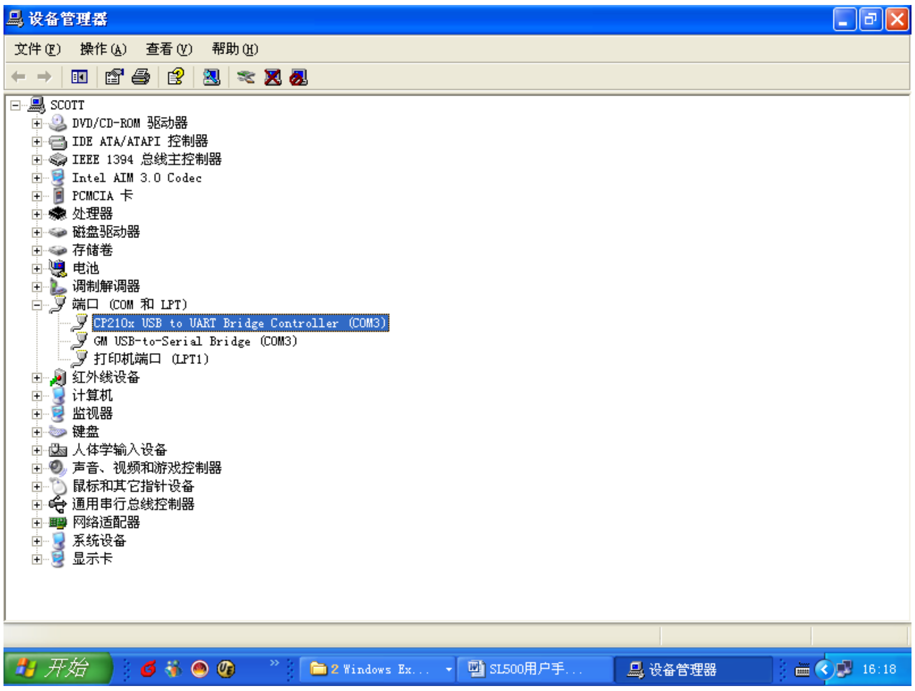


The buzzer is the highest component on the whole board and its height is 5.5mm. The thickness of PCB is 1mm. The height of the SAM slots on the bottom side of PCB is 2.5mm if there are SAM slots[3].

4. INTERFACE DESCRIPTION

RT400 realizes its USB virtual COM via USB bridge. Connect RT400 to the USB port of PC, after installing the driver it will come out a virtual COM. You can find the virtual COM number on the “Device Manager ” as follows:

[3] SAM slots are reserved for future use and its application is not included in the present version.



```
SCOTT
  DVD/CD-ROM 驱动器
  IDE ATA/ATAPI 控制器
  IEEE 1394 总线主控制器
  Intel AIM 3.0 Codec
  PCMCIA 卡
  处理器
  磁盘驱动器
  存储卷
  电池
  调制解调器
  端口 (COM 和 LPT)
    CP210x USB to UART Bridge Controller (COM3)
    GM USB-to-Serial Bridge (COM3)
    打印机端口 (LPT1)
  红外线设备
  计算机
  监视器
  键盘
  人体学输入设备
  声音、视频和游戏控制器
  鼠标和其它指针设备
  通用串行总线控制器
  网络适配器
  系统设备
  显示卡
```

The driver can be downloaded from this product web page in our website.

5. SDK INFORMATION

RT400_DEMO.exe – a full UHF function demo for RT400, it needs 1024*768 pixels at least.

RT40007_API.dll – a Windows API file from which users can call functions. Description of these functions can be found in this document.

RT400 Communication Protocol.pdf – a communication protocol description file in the level of COM interface.

6. FUNCTIONAL DESCRIPTION

6.1 COMMON FUNCTIONS

6.1.1 UhfReaderConnect ()

Description

Open the COM port and establish the connection.

Function Prototype

int WINAPI UhfReaderConnect (HANDLE &hCom, char* cPort, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE &hCom: output parameter, serial port handle, default value is NULL;

char* cPort: input parameter, serial port number;

UCHAR flagCrc: input parameter, select the baud rate and whether CRC is available,

0x00: baud rate 9600bps, CRC is unavailable;

0x01: baud rate 9600 bps, CRC is available;

0x02: baud rate 19200 bps, CRC is unavailable;

0x03: baud rate 19200 bps, CRC is available;

0x04: baud rate 57600 bps, CRC is unavailable;

0x05: baud rate 57600 bps, CRC is available;

0x06: baud rate 115200 bps, CRC is unavailable;

0x07: baud rate 115200 bps, CRC is available.

6.1.2 UhfReaderDisconnect ()

Description

Close COM port.

Function Prototype

int WINAPI UhfReaderDisconnect (HANDLE &hCom, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE &hCom: input parameter, serial port handle.

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.1.3 UhfOpenPort ()[4]

Description

Open the COM port and establish the connection.

Function Prototype

[4] The difference between UhfReaderconnect() and UhfOpenPort() is that the former one has shaking hands process to ensure the connetion. Actually user can use either to connect RT4 via COM port.

int WINAPI UhfOpenPort (HANDLE &hCom, char* cPort, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE &hCom: output parameter, serial port handle, default value is NULL;

char* cPort: input parameter, serial port number;

UCHAR flagCrc: input parameter, select the baud rate and whether CRC is available,

0x00: baud rate 9600bps, CRC is unavailable;

0x01: baud rate 9600 bps, CRC is available;

0x02: baud rate 19200 bps, CRC is unavailable;

0x03: baud rate 19200 bps, CRC is available;

0x04: baud rate 57600 bps, CRC is unavailable;

0x05: baud rate 57600 bps, CRC is available;

0x06: baud rate 115200 bps, CRC is unavailable;

0x07: baud rate 115200 bps, CRC is available.

6.1.4 UhfClosePort ()

Description

Close COM port.

Function Prototype

int WINAPI UhfReaderDisconnect (HANDLE &hCom, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE &hCom: input parameter, serial port handle.

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.1.5 UhfGetPaStatus ()

Description

Check the connection status.

Function Prototype

int WINAPI UhfGetPaStatus (HANDLE hCom, UCHAR* uStatus, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, serial port handle;

UCHAR* uStatus: output parameter, connection status（1byte, 0: connection succeeds; not 0: connection fails）;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.1.6 UhfGetPower ()

Description

Get RF output power.

Function Prototype

int WINAPI UhfGetPower (HANDLE hCom, UCHAR* uPower, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR* uPower: output parameter, RF output power(1 byte);

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.1.7 UhfSetPower ()

Description

Set RF output power.

Function Prototype

int WINAPI UhfSetPower (HANDLE hCom, UCHAR uOption, UCHAR uPower, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR uOption: input parameter, must be 0x01;

UCHAR uPower: input parameter, output power to be set;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.1.8 UhfGetFrequency ()

Description

Get RF frequency.

Function Prototype

int WINAPI UhfGetFrequency (HANDLE hCom, UCHAR* uFreMode, UCHAR* uFreBase, UCHAR* uBaseFre, UCHAR* uChannNum, UCHAR* uChannSpc, UCHAR* uFreHop, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR* uFreMode: output parameter, FREMODE(1 byte);

UCHAR* uFreBase: output parameter, FREBASE (1 byte);

UCHAR* uBaseFre: output parameter, BF(2 bytes);

UCHAR* uChannNum: output parameter, CN(1 byte);

UCHAR* uChannSpc: output parameter, SPC(1 byte);

UCHAR* uFreHop: output parameter, FREHOP(1 byte);

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

See to appendix A for the definition of FREMODE, FREBASE, BF, CN, SPC, FREHOP.

6.1.9 UhfSetFrequency ()

Description

Set RF frequency.

Function Prototype

int WINAPI UhfSetFrequency(HANDLE hCom, UCHAR uFreMode, UCHAR uFreBase, UCHAR* uBaseFre, UCHAR uChannNum, UCHAR uChannSpc, UCHAR uFreHop, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR uFreMode: input parameter, FREMODE(1 byte);

UCHAR uFreBase: input parameter, FREBASE(1 byte);

UCHAR* uBaseFre: input parameter, BF(2 bytes);

UCHAR uChannNum: input parameter, CN(1 byte);

UCHAR uChannSpc: input parameter, SPC(1 byte);

UCHAR uFreHop: input parameter, FREHOP(1 byte);

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.1.10 UhfGetRegister ()

Description

Get register value.

Function Prototype

int WINAPI UhfGetRegister (HANDLE hCom, int RADD, int RLEN, UCHAR* STATUS, UCHAR* REG, UCHAR flagCrc)

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

int RADD: input parameter, start address of the register;

int RLEN: input parameter, data length to be read (unit: byte);

UCHAR* STATUS: output parameter, return execution status (1 byte);

UHCAR* REG: output parameter, data of the register;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.1.11 UhfSetRegister ()

Description

Set register value.

Function Prototype

int WINAPI UhfSetRegister (HANDLE hCom, int RADD, int RLEN, UCHAR* REG, UCHAR* STATUS, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

int RADD: input parameter, start address of the register;

int RLEN: input parameter, data length to be written(unit: byte);

UHCAR* REG: input parameter, data of the register;

UCHAR* STATUS: out parameter, return execution status (1 byte);

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.1.12 UhfResetRegister ()

Description

Restore register to default value.

Function Prototype

int WINAPI UhfResetRegister (HANDLE hCom, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.1.13 UhfSaveRegister ()

Description

Save the register value.[5]

Function Prototype

int WINAPI UhfSaveRegister (HANDLE hCom, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.1.14 UhfGetVersion ()

Description

Get hardware serial number and firmware serial number.

Function Prototype

[5] If not save, any change of register value will be lost when power off.

int WINAPI UhfGetVersion (HANDLE hCom, UCHAR* uSerial, UCHAR* uVersion, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR* uSerial: output parameter, hardware serial number(6 bytes);

UCHAR* uVersion: output parameter, firmware serial number(3 bytes);

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.1.15 UhfGetReaderUID ()

Description

Get module UID information.

Function Prototype

int WINAPI UhfGetReaderUID (HANDLE hCom, UCHAR* uUid, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR* uUid: output parameter, module UID information(12 bytes);

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.1.16 UhfEnterSleepMode ()

Description

Enter sleep mode.

Function Prototype

int WINAPI UhfEnterSleepMode (HANDLE hCom, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.1.17 UhfStopOperation ()

Description

Stop executing operation.

Function Prototype

int WINAPI UhfStopOperation (HANDLE hCom, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.2 TAG INVENTORY FUNCTIONS

6.2.1 UhfStartInventory ()

Description

Start tag inventory loop.

Function Prototype

int WINAPI UhfStartInventory (HANDLE hCom, UCHAR flagAnti, UCHAR initQ, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR flagAnti: input parameter, anti-collision on/off(1: anti-collision inventory;0: single tag inventory);

UCHAR initQ: input parameter, initial Q value for anti-collision, available when flagAnti is 1;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

Call UhfStartInventory() to start tag inventory loop. When flagAnti=1, anti-collision function turns on to detect multiple tags. When flagAnti=0, anti-collision function turns off to detect single tag. UhfStartInventory() can be combined with UhfReadInventory(), UhfStopOperation() functions. UhfStartInventory() just starts the loop, updating data(UII) to the buffer. UhfReadInventory() is called to read data(UII) from the buffer not from tags. When the loop starts, the module only responses UhfStopOperation() – stops the loop.

6.2.2 UhfReadInventory ()

Description

Get UII of the tag.

Function Prototype

int WINAPI UhfReadInventory (HANDLE hCom, UCHAR* uLenUii, UCHAR* uUii);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR* uLenUii: output parameter, length of UII, 1 byte;

UCHAR* uUii: output parameter, UII, 66 bytes at least.

6.2.3 UhfInventorySingleTag ()[6]

Description

[6] This function conflicts with tag inventory loop. User need stop tag inventory loop then run this function.

Single inventory, get UII of the single tag.

Function Prototype

int WINAPI UhfInventorySingleTag (HANDLE hCom, UCHAR* uLenUii, UCHAR* uUii , UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR* uLenUii: output parameter, length of UII, 1 byte;

UCHAR* uUii: output parameter, UII, 66 bytes at least;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.3 TAG DATA OPERATION FUNCTIONS

6.3.1 UhfReadDataByEPC ()

Description

Read tag data (indicating tag UII[7]).

Function Prototype

int WINAPI UhfReadDataByEPC (HANDLE hCom, UCHAR* uAccessPwd, UCHAR uBank, UCHAR* uPtr, UCHAR uCnt, UCHAR* uUii, UCHAR* uReadData, UCHAR* uErrorCode, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR* uAccessPwd: input parameter, access password (4 bytes, 0x00000000 if no access

[7] Select the tag with indicating UII.

password);

UCHAR uBank: input parameter, memory zone option;

UCHAR* uPtr: input parameter, the starting offset address;

UCHAR uCnt: input parameter, length of data to be read(unit: word, 1 word = 2 bytes, can not be assigned zero);

UCHAR* uUii: input parameter, tag's UII;

UCHAR* uReadData: output parameter, tag data, at least uCnt*2 bytes;

UCHAR* uErrorCode: output parameter, error code (1 byte), valid when this function fails and uErrorCode doesn't equal to 0xFF.

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.3.2 UhfReadDataFromSingleTag ()

Description

Read tag data (not indicating tag UII[8]).

Function Prototype

int WINAPI UhfReadDataFromSingleTag (HANDLE hCom, UCHAR* uAccessPwd, UCHAR uBank, UCHAR* uPtr, UCHAR uCnt, UCHAR* uReadData, UCHAR* uUii, UCHAR* uLenUii, UCHAR* uErrorCode, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: Parameter Defination, COM port handle;

UCHAR* uAccessPwd: input parameter, access password (4 bytes, 0x00000000 if no access password);

UCHAR uBank: input parameter, memory zone option;

UCHAR* uPtr: input parameter, the starting offset address;

UCHAR uCnt: input parameter, length of data to be read(unit: word, 1 word = 2 bytes, can not

[8] This function must run in the condition that there is only one single tag in the detection range.

be assigned zero);

UCHAR* uReadData: output parameter, tag data, at least uCnt*2 bytes;

UCHAR* uUii: output parameter, tag's UII;

UCHAR* uLenUii : output parameter, length of tag's UII;

UCHAR* uErrorCode: output parameter, error code (1 byte), valid when this function fails and uErrorCode doesn't equal to 0xFF.

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.3.3 UhfReadMaxDataByEPC ()

Description

Read all data from selected memory zone(indicating UII).

Function Prototype

int WINAPI UhfReadMaxDataByEPC (HANDLE hCom, UCHAR* uAccessPwd, UCHAR uBank, UCHAR* uPtr, UCHAR* uUii, UCHAR* Data_len, UCHAR* uReadData, UCHAR* uErrorCode, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR* uAccessPwd: input parameter, access password (4 bytes, 0x00000000 if no access password);

UCHAR uBank: input parameter, memory zone option;

UCHAR* uPtr: input parameter, the starting offset address;

UCHAR* uUii: input parameter, tag's UII;

UCHAR* Data_len: output parameter, length of data to be read, unit; byte;

UCHAR* uReadData: output parameter, tag data;

UCHAR* uErrorCode: output parameter, error code (1 byte), valid when this function fails and uErrorCode doesn't equal to 0xFF.

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

Attention,

The module supports maximum 233 bytes memory zone and if the memory zone has more bytes than that, this function return failure.

6.3.4 UhfReadMaxDataFromSingleTag ()

Description

Read all data from selected memory zone(not indicating UII)..

Function Prototype

int WINAPI UhfReadMaxDataFromSingleTag (HANDLE hCom, UCHAR* uAccessPwd, UCHAR uBank, UCHAR* uPtr, UCHAR* Data_len, UCHAR* uReadData, UCHAR* uUii, UCHAR* uLenUii, UCHAR* uErrorCode, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR* uAccessPwd: input parameter, access password (4 bytes, 0x00000000 if no access password);

UCHAR uBank: input parameter, memory zone option;

UCHAR* uPtr: input parameter, the starting offset address;

UCHAR* Data_len: output parameter, length of data to be read, unit: byte;

UCHAR* uReadData: output parameter, tag data;

UCHAR* uUii: output parameter, tag’s UII;

UCHAR* uLenUii: output parameter, length of tag’s UII, unit: byte;

UCHAR* uErrorCode: output parameter, error code (1 byte), valid when this function fails and uErrorCode doesn’t equal to 0xFF;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

Attention,

The module supports maximum 233 bytes memory zone and if the memory zone has more bytes than that, this function return failure.

6.3.5 UhfWriteDataByEPC ()

Description

Write one word(2 bytes) into tag(indicating UII).

Function Prototype

int WINAPI UhfWriteDataByEPC (HANDLE hCom, UCHAR* uAccessPwd, UCHAR uBank, UCHAR* uPtr, UCHAR uCnt, UCHAR* uUii, UCHAR* uWriteData, UCHAR* uErrorCode, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR* uAccessPwd: input parameter, access password (4 bytes, 0x00000000 if no access password);

UCHAR uBank: input parameter, memory zone option;

UCHAR* uPtr: input parameter, the starting offset address;

UCHAR uCnt: input parameter, length of data to be written(uCnt must equal to 1);

UCHAR* uUii: input parameter, tag's UII;

UCHAR* uWriteData: input parameter, data to be written;

UCHAR* uErrorCode: output parameter, error code (1 byte), valid when this function fails and uErrorCode doesn't equal to 0xFF;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.3.6 UhfWriteDataToSingleTag ()

Description

Write one word(2 bytes) into tag(not indicating UII).

Function Prototype

int WINAPI UhfWriteDataToSingleTag (HANDLE hCom, UCHAR* uAccessPwd, UCHAR uBank, UCHAR* uPtr, UCHAR uCnt, UCHAR* uWriteData, UCHAR* uUii, UCHAR* uLenUii, UCHAR* uErrorCode, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: Parameter Defination, COM port handle;

UCHAR* uAccessPwd: input parameter, access password (4 bytes, 0x00000000 if no access password);

UCHAR uBank: input parameter, memory zone option;

UCHAR* uPtr: input parameter, the starting offset address;

UCHAR uCnt: input parameter, length of data to be written(uCnt must equal to 1);

UCHAR* uWriteData: input parameter, data to be written;

UCHAR* uUii: output parameter, tag's UII;

UCHAR* uLenUii : output parameter, length of UII;

UCHAR* uErrorCode: output parameter, error code (1 byte), valid when this function fails and uErrorCode doesn't equal to 0xFF;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.3.7 UhfBlockWriteDataByEPC ()

Description

Write multiple bytes into the tag(indicating UII).

Function Prototype

int WINAPI UhfBlockWriteDataByEPC (HANDLE hCom, UCHAR* uAccessPwd, UCHAR uBank, UCHAR* uPtr, UCHAR uCnt, UCHAR* uUii, UCHAR* uWriteData, UCHAR* uErrorCode, UCHAR* uStatus, UCHAR* uWritedLen, UCHAR* RuUii, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR* uAccessPwd: input parameter, access password (4 bytes, 0x00000000 if no access password);

UCHAR uBank: input parameter, memory zone option;

UCHAR* uPtr: input parameter, the starting offset address;

UCHAR uCnt: input parameter, length of data to be written, unit: byte;

UCHAR* uUii: input parameter, tag's UII;

UCHAR* uWriteData: input parameter, data to be written;

UCHAR* uErrorCode: output parameter, error code (1 byte), valid when this function fails and uErrorCode doesn't equal to 0xFF;

UCHAR* uStatus: output parameter, operation result(1 byte);

UCHAR* uWritedLen: output parameter, length of data written successfully, unit: byte;

UCHAR* RuUii: output parameter, return tag's UII;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.3.8 UhfBlockWriteDataToSingleTag()

Description

Write multiple bytes into the tag (not indicating UII).

Function Prototype

int WINAPI UhfBlockWriteDataToSingleTag (HANDLE hCom, UCHAR* uAccessPwd, UCHAR uBank, UCHAR* uPtr, UCHAR uCnt, UCHAR* uWriteData, UCHAR* uUii, UCHAR* uLenUii, UCHAR* uErrorCode, UCHAR* uStatus,UCHAR* uWritedLen, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR* uAccessPwd: input parameter, access password (4 bytes, 0x00000000 if no access password);

UCHAR uBank: input parameter, memory zone option;

UCHAR* uPtr: input parameter, the starting offset address;

UCHAR uCnt: input parameter, length of data to be written, unit: byte;

UCHAR* uWriteData: input parameter, data to be written;

UCHAR* uUii: output parameter, tag's UII;

UCHAR* uLenUii: output parameter, length of tag's UII, unit: byte;

UCHAR* uErrorCode: output parameter, error code (1 byte), valid when this function fails and uErrorCode doesn't equal to 0xFF;

UCHAR* uStatus: output parameter, operation result(1 byte);

UCHAR* uWritedLen: output parameter, length of data written successfully, unit: byte;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.3.9 UhfEraseDataByEPC ()

Description

Erase data in the tag(indicating UII).

Function Prototype

int WINAPI UhfEraseDataByEPC (HANDLE hCom, UCHAR* uAccessPwd, UCHAR uBank, UCHAR* uPtr, UCHAR uCnt, UCHAR* uUii, UCHAR* uErrorCode, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR* uAccessPwd: input parameter, access password (4 bytes, 0x00000000 if no access password);

UCHAR uBank: input parameter, memory zone option;

UCHAR* uPtr: input parameter, the starting offset address;

UCHAR uCnt: input parameter, length of data to be erased (unit: byte);

UCHAR* uUii: input parameter, tag's UII;

UCHAR* uErrorCode: output parameter, error code (1 byte), valid when this function fails and uErrorCode doesn't equal to 0xFF;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.3.10 UhfEraseDataFromSingleTag ()

Description

Erase data in the tag(indicating UII).

Function Prototype

int WINAPI UhfEraseDataFromSingleTag (HANDLE hCom, UCHAR* uAccessPwd, UCHAR uBank, UCHAR* uPtr, UCHAR uCnt, UCHAR* uUii, UCHAR* uErrorCode, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR* uAccessPwd: input parameter, access password (4 bytes, 0x00000000 if no access password);

UCHAR uBank: input parameter, memory zone option;

UCHAR* uPtr: input parameter, the starting offset address;

UCHAR uCnt: input parameter, length of data to be erased (unit: byte);

UCHAR* uUii: output parameter, return tag's UII;

UCHAR* uErrorCode: output parameter, error code (1 byte), valid when this function fails and uErrorCode doesn't equal to 0xFF;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.3.11 UhfLockMemByEPC ()

Description

Lock the data of memory(indicating UII).

Function Prototype

int WINAPI UhfLockMemByEPC (HANDLE hCom, UCHAR* uAccessPwd, UCHAR* uLockData, UCHAR* uUii, UCHAR* uErrorCode, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR* uAccessPwd: input parameter, access password (4 bytes, 0x00000000 if no access password);

UCHAR* uLockData: input parameter, input lockdata[9](3 bytes);

UCHAR* uUii: input parameter, tag’s UII;

UCHAR* uErrorCode: output parameter, error code (1 byte), valid when this function fails and uErrorCode doesn’t equal to 0xFF;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.3.12 UhfLockMemFromSingleTag ()

Description

Lock the data of memory(not indicating UII).

Function Prototype

int WINAPI UhfLockMemFromSingleTag (HANDLE hCom, UCHAR* uAccessPwd, UCHAR* uLockData, UCHAR* uUii, UCHAR* uErrorCode, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

[9] About how to operate lock function, refer to <18000-6C Tag Operation Manual>

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR* uAccessPwd: input parameter, access password (4 bytes, 0x00000000 if no access password);

UCHAR* uLockData: input parameter, input lockdata[10](3 bytes);

UCHAR* uUii: output parameter, return tag's UII;

UCHAR* uErrorCode: output parameter, error code (1 byte), valid when this function fails and uErrorCode doesn't equal to 0xFF;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.3.13 UhfKillTagByEPC ()

Description

Kill the tag(indicating UII).

Function Prototype

int WINAPI UhfKillTagByEPC (HANDLE hCom, UCHAR* uKillPwd, UCHAR* uUii, UCHAR* uErrorCode, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR* uAccessPwd: input parameter, kill password(4 bytes);

UCHAR* uUii: input parameter, tag's UII;

UCHAR* uErrorCode: output parameter, error code (1 byte), valid when this function fails and uErrorCode doesn't equal to 0xFF;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.3.14 UhfKillSingleTag ()

[10] About how to operate lock function, refer to <18000-6C Tag Operation Manual>

Description

Kill the tag(not indicating UII).

Function Prototype

int WINAPI UhfKillSingleTag (HANDLE hCom, UCHAR* uKillPwd, UCHAR* uUii, UCHAR* uErrorCode, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR* uAccessPwd: input parameter, kill password(4 bytes);

UCHAR* uUii: output parameter, tag's UII;

UCHAR* uErrorCode: output parameter, error code (1 byte), valid when this function fails and uErrorCode doesn't equal to 0xFF;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.3.15 UhfBlockWriteEPCByEPC()

Description

Write EPC data into UII memory (indicating UII).

Function Prototype

int WINAPI UhfBlockWriteEPCByEPC (HANDLE hCom, UCHAR* uAccessPwd, UCHAR uCnt, UCHAR* uUii, UCHAR* uWriteData, UCHAR* uErrorCode, UCHAR* uStatus, UCHAR* uWritedLen, UCHAR* RuUii, UCHAR flagCrc)

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR* uAccessPwd: input parameter, access password (4 bytes, 0x00000000 if no access password);

UCHAR uCnt: input parameter, length of data to be written;

UCHAR* uUii: input parameter, tag’s UII;

UCHAR* uWriteData: input parameter, data to be written;

UCHAR* uErrorCode: output parameter, error code (1 byte), valid when this function fails and uErrorCode doesn’t equal to 0xFF;

UCHAR* uStatus: output parameter, operation result(1 byte);

UCHAR* uWritedLen: output parameter, length of data written successfully, unit: byte;

UCHAR* RuUii: output parameter, return the tag’s UII;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

6.3.16 UhfBlockWriteEPCToSingleTag()

Description

Write EPC data into UII memory (not indicating UII).

Function Prototype

int WINAPI UhfBlockWriteEPCToSingleTag (HANDLE hCom, UCHAR* uAccessPwd, UCHAR uCnt, UCHAR* uWriteData, UCHAR* uUii, UCHAR* uLenUii, UCHAR* uStatus, UCHAR* uErrorCode, UCHAR* uWritedLen, UCHAR flagCrc);

Return Value

1: success;

Others: failure.

Parameter Defination

HANDLE hCom: input parameter, COM port handle;

UCHAR* uAccessPwd: input parameter, access password (4 bytes, 0x00000000 if no access password);

UCHAR uCnt: input parameter, length of data to be written;

UCHAR* uUii: input parameter, tag’s UII;

UCHAR* uWriteData: input parameter, data to be written;

UCHAR* uLenUii: output parameter, return the length of tag’s UII, unit: byte;

UCHAR* uErrorCode: output parameter, error code (1 byte), valid when this function fails and uErrorCode doesn’t equal to 0xFF;

UCHAR* uStatus: output parameter, operation result(1 byte);

UCHAR* uWritedLen: output parameter, length of data written successfully, unit: byte;

UCHAR flagCrc: input parameter, whether CRC16 is available,

0: CRC is unavailable;

1: CRC is available.

Appendix A: Frequency Parameters

Parameter Name	Type	Description

bFreMode	UCHAR*	Frequency Option:
		0: 920-925MHz;
		1: 840-845MHz;
		2: ETSI;
		3: reserved;
		4: user-defined;

BFreBase	UCHAR*	Base Frequency:
		0: 50MHz;
		1: 125MHz;

bBaseFre	UCHAR*	Starting Frequency, value range: 840~960;
bChannNum	UCHAR*	The Number of Frequency Channels, value range: 1~16;

bChannSpc	UCHAR*	Frequency channel factor:
		When bFreBase = 50, value range: 1~20;
		When bFreBase = 125, value range: 1~8;

bFreHop	UCHAR*	The Way of Frequency Jump: 0: Random; 1: From high to low frequency; 2: From low to high frequency; others: Random;

The relationships between starting frequency, stop frequency, base frequency, frequency channel, frequency channel factor:

```
if(base frequency == 0)
{
    stop frequency - starting frequency = (frequency channel - 1) * frequency channel factor * 0.050
}else{
    stop frequency - starting frequency = (frequency channel - 1) * frequency channel factor * 0.125
}
```

Example: Frequency range is 920.625-924.375MHz
base frequency =1;
starting frequency decimal part is 0.625;
starting frequency integer part is 920;
starting frequency decimal part in hex: 0.625÷0.125 = 0x05;
‘920’ in binary is 11 1001 1000;

```
bBaseFre[0] = (byte)(11 1001 1000 >> 3);                  --->0x73
bBaseFre[1] = (byte)(11 1001 1000 << 5 | (0x05 & 0x1F));  --->0x05
```

```
stop frequency - starting frequency = (frequency channel - 1) * frequency channel factor * 0.125;
That is,
924.375 -920.625 = （frequency channel - 1）* frequency channel factor * 0.125;
So that,
（frequency channel -1）* frequency channel factor = 30;
And,
When “frequency channel”= 16, “frequency channel factor”= 2;

bFreMode = 4;
bFreBase = 1;
bFreBase = {0x73 , 0x05};
bChannNum = 16;
bChannSpc = 2;
bFreHop = 0;(Random frequency jump).
```

Appendix B: Tag Parameters

Parameter Type	Parameter Name	Defination
UCHAR*	uAccessPwd	Tag access password, length: 4 bytes.
UCHAR*	uKillPwd	Tag kill password, length: 4 bytes.
UCHAR	uBank	Tag memory zone: 0x00: RESERVED 0x01: UII 0x02: TID 0x03: USER
UCHAR*	uPtr	Starting offset address.
UCHAR	uCnt	Length of byte count.
UCHAR*	uUii	Tag's UII.
UCHAR*	uLenUii	Length of UII.
UCHAR*	uReadData	Data being read.
UCHAR*	uWriteData	Data to be written.
UCHAR*	uWritedLen	Length of data written successfully.
UCHAR*	uStatus	Return status, it's an array whose length =1.
UCHAR*	uErrorCode	Error code, it's an array whose length =1.