

1.1

Function : *OpenReader*

Description: To create a connection with the reader before control it.

Function Call:

int RFID_OpenReader(char Com[])

Parameter:

Com : The reader's device name

1.2

Function : *CloseReader*

Description: To finish controlling the reader

Function Call:

int RFID_CloseReader()

1.3

Function : *GetFWVersion*

Description: Get the firmware version

Function Call:

int RFID_FWVersion(char OutVersion[])

Parameter:

OutVersion: Get reader's firmware version

1.4

Function : *APIVersion*

Description: To get the library version

Function Call:

int RFID_GetAPIVersionString(char OutVersion[])

Parameter:

OutVersion : library version

1.5

Function : *SetWorkingType*

Description: This API change the reader working type with different card type and this should be called before read the card

Function Call:

int RFID_WorkingType(int type,int Half_Power)

Parameter:

type: Reader *Working Type*

0: 15693

1: 14443A

2: 14443B

3: Felica

Half_Power : Set Reader output power

0: Full_Power(200 mW)

1: Half_Power(100 mW)

1.6

Function : *AntennaContro*

Description: Enable/Disable antenna

Function Call:

int RFID_AntennaControl(int Select)

Parameter:

Select: Antenna Contro

1: Enable Antenna

0: Disable Antenna

2 : 15693 Mode

2.1

Function : *Inventory15693*

Description: Set the card to StayQuiet mode and return the card ID

Function Call:

int Inventory15693(char flag[],char AFI[],char OutUID[])

Parameter:

Flag : Set *Tag Flags*. (Please refer to section 5.1)

AFI : Application Family Identifier parameter, please refer to ISO15693 document

OutUID : Receive the tag ID

2.2

Function : *ReadSingleBlock*

Description: Read the block data from the specific ISO15693 tag

Function Call:

**Int ReadSingleBlock(char block,char flag[],char
AddressUID[],char OutData[])**

Parameter:

Block : The block which you want to read(00 ~ FF)

Flag : Set *Flags*.(Please refer to section 5.1)

AddressUID : Tag UID(Please refer to section 5.1)

OutData : Receive the block data

2.3

Function : *WriteSingleBlock*

Description: Write the block data to the specific ISO15693 tag

Function Call:

**int WriteSingleBlock(char block,char flag[],char
AddressUID[],char Data[])**

Parameter:

Block : The block which you want to write(00 ~ FF)

Flag : Set *Flags*.(Please refer to section 5.1)

AddressUID : Tag UID(Please refer to section 5.1)

Data: The data which you want to write

2.4

Function : ISO15693LockBlock

Description: Lock ISO15693 tag block

Function Call:

int RFID_ISO15693LockBlock(char flag[], char block[])

Parameter:

Flag : Set *Flags*. (Please refer to section 5.1)

Block : The block which you want to lock

3 : 14443A Mode

3.1

Function : GetUid14443A

Description: Get the tag ID(14443A)

Function Call:

int RFID_Get14443AUID(char OutUID[])

Parameter:

OutUID : Receive the tag Uid

3.2

Function : ReadMifareBlock

Description: Read the single block data

Function Call:

int ReadMifareBlock(char Key[], char block[], char OutData[])

Parameter:

Key : Authentication KEY value of the card

block: The block which you want to read

OutData : Receive the data

3.3

Function : WriteMifareBlock

Description: Write data to the single block

Function Call:

int WriteMifareBlock(char Key[],char block[], char Data[])

Parameter:

Key : Authentication KEY value of the card

block: The block which you want to write

Data : The data which you want to write

3.4

Function : ReadUltraLightBlock

Description: Read the single block data(**UltraLight**)

Function Call:

int ReadUltraLightBlock(char block[],char OutData[])

Parameter:

block: The block which you want to read

OutData : Receive the data

3.5

Function : WriteUltraLightBlock

Description: Write data to the single block(**UltraLight**)

Function Call:

int WriteUltraLightBlock(char block[],char Data[])

Parameter:

block: The block which you want to write

Data : The data which you want to write

4 : 14443B Mode

4.1

Function : GetUid14443B

Description: get the 14443B tag ID

Function Call:

int GetUid14443B(int type,char OutUID[])

Parameter:

Type :

01 : 14443B

02 : SR176

03 : SR1X4K

OutUID : Receive the tag **UID**

4.2

Function : Get SRIX4K ChipID

Description: Get SRIX4K Chip ID

Function Call:

int SRIX4KChipID(char OutID[])

Parameter:

OutID : SRIX4K Chip ID

4.3

Function : Get SRIX4K Block Data

Description: Read the single block data

Function Call:

int SRIX4KReadBlock(char block,char OutData[])

Parameter:

block: The Block which you want to Read

OutData : Receive the data

4.4

Function : Write Data To SRIX4K Block

Description: Write data to the single block

Function Call:

int SRIX4KWriteBlock(char block,char Data[])

Parameter:

block: The Block which you want to Write

Data : The data which you want to write

5.1 :

Request Flags Bits 1 to 4
(Ref.: ISO 15693-3:2000(E), Section 7.3.1 Table 3, Page 9)

Bit	Flag Name	Value	Description
b1	Subcarrier flag	0	A single subcarrier is used by the tag.
		1	Two subcarriers are used by the tag.
b2	Data rate flag	0	Low data rate
		1	High data rate
b3	Inventory flag	0	Flags 5 to 8 meaning in following tables (points to table 4 in ISO 15693-3 protocol)
		1	Flags 5 to 8 meaning in following tables (points to table 5 in ISO 15693-3 protocol)
b4	Protocol extension flag	0	No protocol format extension
		1	Protocol format is extended. Reserved for future use.

Request Flags Bits 5 to 8 when inventory flag IS NOT set
(Ref.: ISO 15693-3:2000(E), Section 7.3.1 Table 4, Page 10)

Bit	Flag Name	Value	Description
b5	Select flag	0	Request executed by any tag according to the setting of <i>Address</i> flag.
		1	Request executed only by tag in selected state. The <i>Address</i> flag is set to 0 and the UID field is not included in the request.
b6	Address_flag	0	Request is not addressed. UID field is not included. It can be executed by any tag.
		1	Request is addressed. UID field is included. It is executed only by the tag whose UID matches the UID specified in the request.
b7	Option_flag	0	Meaning is defined by the command description. It is set to 0 if not otherwise defined by the command.
		1	Meaning is defined by the command description.
b8	RFU	0	Reserved for future use

Request Flags Bits 5 to 8 when inventory flag IS set
(Ref.: ISO 15693-3:2000(E), Section 7.3.1 Table 5, Page 10)

Bit	Flag Name	Value	Description
b5	AFI_flag	0	AFI field is not present.
		1	AFI field is present.
b6	Nb_slots_flag	0	16 slots
		1	1 slot
b7	Option_flag	0	Meaning is defined by the request description. It is set to 0 if not otherwise defined by the request.
		1	Meaning is defined by the request description.
b8	RFU	0	Reserved for future use