MR6100 Serial Reader SDK Function

DLL file name：MR6100Api.dll

1.Function

1-1. Reader manage function

TcpConnectReader

Function describe	public int TcpConnectReader(string ip, int port)
Function	Reader connection via TCP/IP
Parameter	parameter　Parameter signficance, scope ip　Reader ethernet addesss，Is an IP address of the string format effective representation, Must be connected with the computer in the same network segment (default："192.168.1.200") port　Reader port number(range 1~65535)
Response	Response value：2001 success

Example:	If(TcpConnectReader ("192.168.1.200",100)==2001)
	　MessageBox("success");
	Else
	　MessageBox("fail");

OpenCommPort

Function Describe	public int OpenCommPort(string strPort, int nBoud)
Function	Initialize reader port connection and configure READER baudrate parameter
Parameter	Parameter　Parameter signficance, scope strPort　Communication serial port no.:(COM1、COM2、COM3……) nBoud　Serial port baudrate(9600、19200、38400、57600、

	115200)
Response	Response value：2001 success

Example	If(OpenCommPort（“COM1”,9600)==2001)
	MessageBox("success");
	Else
	MessageBox("fail");

TcpCloseConnect

Function Describe	public int TcpCloseConnect()
Function	Shut off reader port, disconnect the connection
Parameter	N/A
Response	Response value：2001 success

Example	If(TcpCloseConnect ()==2001)
	MessageBox("success");
	Else
	MessageBox("fail");

CloseCommPort

Function Describe	public void CloseCommPort()
Function	Shut off reader port, disconnect the connection
Parameter	N/A
Response	N/A
Example	CloseCommPort ();

SetBaudRate

Function Describe	public int SetBaudRate(int nReaderAddr, int nBaudRate)
Function	Set reader baudrate

Parameter	Parameter	Parameter signficance, scope
	nBaudRate	Serial port communication baudrate are 9600、19200、38400、57600 and 115200 (or 0,1,2,3,4)
	nReaderAddr	nReaderAddr：Reader address，for fixed reader compose RS485 network use，default 0XFF（handheld reader and module not applicable）
Response	8FD4 return value：2001 is success	

Example	If(SetBaudRate(255, 115200)==2001)
	MessageBox("success");
	Else
	MessageBox("fail");	

ResetReader

Function Describe	public int ResetReader(int readerAddr)
Function	Reader reset
Parameter	readerAddr　Reader address，for fixed reader compose RS485 network use，default 0XFF（handheld reader and module not applicable）
Response	Response value：2001 success

Example	If(ResetReader(255)==2001)
	MessageBox("success");
	Else
	MessageBox("fail");

ResetParameter

Function Describe	public int ResetParameter (int readerAddr)
Function	Reset parameter of reader
Parameter	readerAddr　Reader address，for fixed reader compose RS485 network use，default 0XFF（handheld reader and module not applicable）
Response	Reponse value：2001 success

Example	If(ResetParameter (255)==2001)
	MessageBox("success");
	Else
	MessageBox("fail");

GetFirmwareVersion

Function Describe	public int GetFirmwareVersion(int readerAddr,ref byte v1, ref byte v2)
Function	Get reader firmware version
Parameter	parameter: Parameter signficance, scope readerAddr: Reader address v1: Firmware high byte v2: Firmware low byte
Response	Response value：2001 success

Example	If(GetFirmwareVersion(255,ref v1,ref v2)==2001)
	MessageBox("success");
	Else
	MessageBox("fail");

SetRf

Function Describe	public int SetRf(int readerAddr, int power1, int power2, int power3, int power4)
Function	Set Reader RF power and frequency
Parameter	Paramter: Parameter signficance, scope readerAddr: Reader address Power1-Power4: Power value, value 0~31，equal to 0~31dBm. For separate antenna 1~4
Response	Response value：2001 success

Example	If(SetRf(255,0, 27,30,30) == 2001)
	MessageBox("success");
	Else
	MessageBox("fail");

GetRf

Function Describe	public int GetRf(int readerAddr,ref int[] power)
Function	Get Reader current power and frequency
Parameter	Paramter Parameter signficance, scope readerAddr Reader address power Get antenna power value，0~31，0~31dBm
Response	Response value：2001 success

Example	If(GetRf(255,ref power) == 2001)
	MessageBox("success");
	Else
	MessageBox("fail");

SetAnt

Function Describe	public int SetAnt(int readerAddr, byte Antenna)
Function	Set reader antenna number open state
Parameter	Paramter Parameter signficance, scope readerAddr Reader address Antenna Working antenna，show by mask. Low 4 bit for separate four antenna open or not, 1 means open，0 means not open；high 4 bit make no sense
Response	Response value：2001 success

Example	If(SetAnt (ReaderAddr, ant) == 2001)
	MessageBox("success");
	Else
	MessageBox("fail");

GetAnt

Function Describe	public int GetAnt(int ReaderAddr, ref byte workAnt, ref byte

	antState)
Function	Get reader current available open antenna number and antenna connection status
Parameter	Paramter — Parameter signficance, scope ReaderAddr — Reader address，for fixed reader compose RS485 network use，default 0XFF（handheld reader and module not applicable） workAnt — The opening of the antenna current state, show by Mask antState — The current actual antenna available，1 means able to use，0 means antenna not connect or not match
Response	Response value：2001 success

Example	If(GetAnt (ReaderAddr,ref workAnt，ref antState))== 2001)
	MessageBox("success");
	Else
	MessageBox("fail");

SetFrequency

Function Describe	public int SetFrequency (int ReaderAddr, int freqNum, int[] points)
Function	Set reader frequency parameter
Parameter	Paramter — Parameter signficance, scope ReaderAddr — Reader address，for fixed reader compose RS485 network use，default 0XFF（handheld reader and module not applicable） freqNum — Frequency point，if not 0，frequency is all point of Freq points；if Freq num is 0，one byte of Freq points means frequency area type, see below： 0：China 1：America 2：Europe points — Selection frequency range is 900～930MHz，each 250kHz for stepping frequency point index

Response	Response value：2001 success

Example	If(SetFrequency (ReaderAddr,freqNum，points))== 2001)
	MessageBox("success");
	Else
	MessageBox("fail");

GetFrequency

Function Describe	public int GetFrequency (int ReaderAddr,ref int freqNum,ref int[] points)
Function	Get reader RF Frequency
Parameter	Paramter: Parameter signficance, scope ReaderAddr: Reader address，for fixed reader compose RS485 network use，default 0XFF（handheld reader and module not applicable） freqNum: Frequency point，if not 0，frequency is all point of Freq points；if Freq num is 0，one byte of Freq points means frequency area type, see below： 0：China 1：America 2：Europe points: Selection frequency range is 900～930MHz，each 250kHz for stepping frequency point index
Response	Response value：2001 success

Example	If(GetFrequency (ReaderAddr,ref freqNum，ref points))== 2001)
	MessageBox("success");
	Else
	MessageBox("fail");

GetFastTagMode

Function Describe	public int GetFastTagMode (int ReaderAddr, ref int mode)
Function	Get reader Fast Tag Mode
Parameter	Paramter Parameter signficance, scope ReaderAddr Reader address , for fixed reader compose RS485 network use , default 0XFF (handheld reader and module not applicable) mode Mode: 0 is single card (also for few cards) fast tag mode , if not 0, is batch/bulk card mode
Response	Response value：2001 success

Example	If(GetFastTagMode (addr,ref mode)== 2001)
	MessageBox("success");
	Else
	MessageBox("fail");

SetFastTagMode

Function Describe	public int SetFastTagMode (int ReaderAddr, int mode)
Function	Set reader Fast Tag Mode
Parameter	Paramter Parameter signficance, scope ReaderAddr Reader address , for fixed reader compose RS485 network use , default 0XFF (handheld reader and module not applicable) mode Mode: 0 is single card (also for few cards) fast tag mode. If not 0, is batch/bulk card mode
Response	Response value：2001 success

Example	If(SetFastTagMode (addr, mode)== 2001)
	MessageBox("success");
	Else
	MessageBox("fail");

SetTestMode

Function Describe	public int SetTestMode (int ReaderAddr, int mode)
Function	Set reader Test Mode
Parameter	Paramter Parameter signficance, scope ReaderAddr Reader address , for fixed reader

		compose RS485 network use , default 0XFF (handheld reader and module not applicable)
	mode	00 is open power amplifier; 01 is close power amplifier; 02 is antenna calibration , An antenna calibration in four completely disconnected use
Response	Response value：2001 success	

Example	If(SetTestMode (addr, mode)== 2001)
	MessageBox("success");
	Else
	MessageBox("fail");	

QueryIDCount

Function Describe	public int QueryIDCount (int ReaderAddr, ref byte tagCount)
Function	Query ID count in buffer area
Parameter	Paramter: Parameter signficance, scope ReaderAddr: Reader address ,for fixed reader compose RS485 network use , default 0XFF (handheld reader and module not applicable) tagCount: Query ID count
Response	Response value：2001 success

Example	If(QueryIDCount (addr,ref count)== 2001)
	MessageBox("success");
	Else
	MessageBox("fail");

SetOutPort

Function Describe	public int SetOutPort(int ReaderAddr, byte port_num, byte level)

Function	Set Reader output port high/low level
Paramter	Paramter | Parameter signficance, scope ReaderAddr | Reader address，for fixed reader compose RS485 network use，default 0XFF（handheld reader and module not applicable） port_num | IO port serial number：0~1； Relay： 02 level | Output level：0 is low level，1 is high level
Response	Response value：2001 success

Example	If(SetOutPort(255，0,1)==2001)
	MessageBox("success");
	Else
	MessageBox("fail");

BuzzerLEDON

Function Describe	public int BuzzerLEDON(int ReaderAddr)
Function	Buzzer/LED ON
Parameter	Paramter | Parameter signficance, scope ReaderAddr | Reader address，for fixed reader compose RS485 network use，default 0XFF（handheld reader and module not applicable）
Response	Response value：2001 success

Example	If(BuzzerLEDON (255)==2001)
	MessageBox("success");
	Else
	MessageBox("fail");

BuzzerLEDOFF

Function Describe	public int BuzzerLEDOFF(int ReaderAddr)
Function	BUZZER/LED OFF
Parameter	Paramter | Parameter signficance, scope ReaderAddr | Reader address ,for fixed reader compose RS485 network use ,default 0XFF(handheld reader and module not applicable)
Response	Response value：2001 success

Example	If(BuzzerLEDOFF (255)==2001)
	MessageBox("success");
	Else
	MessageBox("fail");

GetBuzzerLED

Function Describe	public int GetBuzzerLED(int ReaderAddr,ref byte state)
Function	Get Buzzer/LED status
Parameter	Paramter: Parameter signficance, scope ReaderAddr: Reader address ,for fixed reader compose RS485 network use ,default 0XFF(handheld reader and module not applicable) state: Get buzzer/LED status
Response	Response value：2001 success

Example	If(GetBuzzerLED(255，state)==2001)
	MessageBox("success");
	Else
	MessageBox("fail");

GetTcpParameter

Function Describe	public int GetTcpParameter(int readerAddr,ref string strIP, ref string strMark, ref string strGate, ref int nTcpPort)
Function	Get Reader TCP Parameter(IP address、subnet mask、default gateway、TCP port)
Parameter	Paramter: Parameter signficance, scope readerAddr: Reader address , for fixed reader compose RS485 network use , default 0XFF (handheld reader and module not applicable) strIP: IP address , Is an IP address of the string format effective representation, Must be connected with the computer in the same network segment (default："192.168.1.200") strMark: Mask address , Is an subnet mask of the string format effective representation.

	strGate	Gateway address, Is gateway of the string format effective representation
	nTcpPort	Reader port number(Range 1~65535)
Response	Response value：2001 success	

Example	If(GetTcpParameter (255，ref strIP ,ref strMark ,ref strGate,ref nTcpPort)==2001)
	MessageBox(“success”);
	Else
	MessageBox(“fail”);	

SetTcpParameter

Function Describe	public int SetTcpParameter(int readerAddr,string strIP, string strMark, string strGate, int nTcpPort)	
Function	Set Reader TCP Parameter(IP address、subnet mask、default gateway、TCP port)	
Parameter	Paramter	Parameter signficance, scope
	readerAddr	Reader address，for fixed reader compose RS485 network use，default 0XFF（handheld reader and module not applicable）
	strIP	IP address，Is an IP address of the string format effective representation, Must be connected with the computer in the same network segment（default："192.168.1.200"）
	strMark	Mask address，Is an subnet mask of the string format effective representation.
	strGate	Gateway address, Is gateway of the string format effective representation
	nTcpPort	Reader port number(Range 1~65535)
Response	Response value：2001 success	

Example	If(SetNetSetting(255，strIP，strMark，strGate, nTcpPort)==2001)
	MessageBox("success");
	else
	MessageBox("fail");	

SetMacAddress

Function Describe	public int SetMacAddress(int ReaderAddr, string[] strMacAddr)
Function	Set reader MAC address
Paramter	Paramter: Parameter signficance, scope readerAddr: Reader address，for fixed reader compose RS485 network use，default 0XFF（handheld reader and module not applicable） strMacAddr: The MAC address need set
Response	Response value：2001 success

Example	If(SetMacAddress(255，strMacAddr)==2001)
	MessageBox("success");
	else
	MessageBox("fail");

GetMacAddress

Function Describe	public int GetMacAddress(int ReaderAddr,ref string[] strMacAddr)
Function	Get reader MAC address
Parameter	Paramter: Parameter signficance, scope readerAddr: Reader address，for fixed reader compose RS485 network use，default 0XFF（handheld reader and module not applicable） strMacAddr: Get MAC address
Response	Response value：2001 success

Example	If(GetMacAddress(255，ref strMacAddr)==2001)
	MessageBox("succes");
	else
	MessageBox("fail");

SetSerialNo

Function Describe	public int SetSerialNo(int ReaderAddr, string[] strSerialNo)

Function	Set reader serial number
Parameter	Paramter　Parameter signficance, scope readerAddr　Reader address ,for fixed reader compose RS485 network use , default 0XFF (handheld reader and module not applicable) strSerialNo　the serial number need set
Response	Response value：2001 success

Example	If(SetSerialNo (255 , strSerialNo)==2001)
	MessageBox("success");
	else
	MessageBox("fail");

GetSerialNo

Function Describe	public int GetSerialNo(int ReaderAddr, string[] strSerialNo)
Function	Get reader serial number
Parameter	Paramter　Parameter signficance, scope readerAddr　Reader address , for fixed reader compose RS485 network use , default 0XFF (handheld reader and module not applicable) strSerialNo　The seril number get
Response	Response value：2001 success

Example	If(GetSerialNo (255 , ref strSerialNo)==2001)
	MessageBox("success");
	else
	MessageBox("fail");

1-2.ISO18000-6B Tag operation function

IsoMultiTagIdentify

Function Describe	public int IsoMultiTagIdentify(int readerAddr,ref byte[,] tag_buf, ref byte tag_cnt)
Function	ISO18000-6B Multi tag identify
Parameter	Paramter　Parameter signficance, scope readerAddr　Reader address , for fixed reader compose RS485

		network use , default 0XFF (handheld reader and module not applicable)
	tag_buf	Identified data (Including tag data , antenna number etc)
	tag_cnt	Tag count
Response	Response value：2001 success	

Example	if(IsoMultiTagIdentify(255,ref tag_buf,ref tag_cnt)==2001)
	MessagBox(“success”);
	else
	MessagBox(“fail”);	

IsoMultiTagRead

Function Describe	public int IsoMultiTagRead(int ReaderAddr, int startAddr, ref byte[,] tag_buf, ref int tag_cnt, ref int getCount)	
Function	ISO18000-6B Multi tag Read	
Parameter	Paramter	Parameter signficance, scope
	readerAddr	Reader address , for fixed reader compose RS485 network use , default 0XFF (handheld reader and module not applicable)
	startAddr	
	tag_buf	Read data (including tag data, antenna number etc.)
	tag_cnt	Read the tag count
	getCount	Get the tag count
Response	Response value：2001 success	

Example	if(IsoMultiTagRead (255,startAddr,ref tag_buf,ref tag_cnt,ref getCount)==2001)
	MessagBox(“success”);
	else
	MessagBox(“fail”);	

IsoReadWithID

Function Describe	public int IsoReadWithID(int ReaderAddr, byte[] byTagID, byte byAddress, ref byte[] byLabelData, ref byte byAntenna)
Function	Read specified ID number data, specifies the address of 8 bytes per read data
Parameter	Paramter: Parameter signficance, scope ReaderAddr: Reader address byTagID: Specified tag TID byAddress: Read start address byLabelData: Read tag data byAntenna: Antenna number
Response	Response value：2001 success

Example	if(IsoReadWithID(255，byTagID, byAddress, ref byLabelData, ref byAntenna)==2001)
	MessagBox(“success”);
	else
	MessagBox(“fail”);
	}

IsoWriteWithID

Function Describe	public int IsoWriteWithID(int readerAddr,byte[] byTagID, byte byAddress, byte byValue)
Function	Write data to the specified ID tag
Parameter	Paramter: Parameter signficance, scope readerAddr: Reader address byTagID: Specified tag TID byAddress: Start address for write data byValue: Data for write tag
Response	Response value：2001 success

Example

```
if(IsoWriteWithID(255,byTagID, byAddress, byValue)==2001)
    MessageBox( “success” );
else
    MessageBox( “fail” );
```

IsoLockWithID

Function Describe	public int IsoLockWithID(int ReaderAddr, byte[] byTagID, byte byAddress)
Function	Specified ID number tag Lock
Parameter	Paramter: Parameter signficance, scope ReaderAddr: Reader address byTagID: Specified tag TID byAddress: Start address for write data
Response	Response value：2001 success

Example	`if(IsoWriteWithID(255 , byTagID, byAddress)==2001)`
	`  MessageBox( “success” );`
	`else`
	`  MessageBox( “fail” );`

IsoRead

Function Describe	public int IsoRead(int readerAddr,byte addr, ref byte[] value)
Function	Read data from 6B tag
Parameter	Paramter: Parameter signficance, scope readerAddr: Reader address addr: Read start address value: Get the read data
Response	Response value：2001 success

Example	`if(IsoRead (255,addr, value)==2001)`
	`  MessageBox( “success” );`
	`else`
	`  MessageBox( “fail” );`

IsoWrite

Function Describe	public int IsoWrite(int readerAddr,byte addr, byte value)
Function	Write the data to 6B tag
Parameter	Paramter Parameter signficance, scope readerAddr Reader address addr Start address of write data value Data of write to tag
Response	Response value：2001 success

Example	if(IsoWrite (255 , addr, value)==2001)
	MessageBox("success");
	else
	MessageBox("fail");

IsoLock

Function Describe	public int IsoLock((int readerAddr,byte addr)
Function	Lock to the specified 6B tag address , once locked , this address cannot be unlock
Parameter	Paramter Parameter signficance, scope readerAddr Reader address addr Address of lock
Response	Response value：2001 success

Example	if(IsoLock (255 , addr))==2001)
	MessageBox("success");
	else
	MessageBox("fail");

IsoQueryLock

Function Describe	public int IsoQueryLock(int ReaderAddr, byte addr, ref byte lstate)

Function	Iso18000 query lock
Parameter	Paramter Parameter signficance, scope readerAddr Reader address addr Address of lock lstate Lock status
Response	Response value：2001 success

Example	if(IsoQueryLock (255，addr，ref lstate))==2001)
	MessageBox(“success”);
	else
	MessageBox(“fail”);

IsoQueryLockWithUID

Function Describe	public int IsoQueryLock(int ReaderAddr, byte[] byTagID,byte addr, ref byte lstate)
Function	Iso18000 to specified ID tag lock query
Parameter	Paramter Parameter signficance, scope readerAddr Reader address byTagID Specified tag ID addr Address of lock lstate Lock status
Response	Response value：2001 success

Example	if(IsoQueryLock (255，id,addr,ref lstate))==2001)
	MessageBox(“success”);
	else
	MessageBox(“fail”);

1-3. EPC GEN2 tag operation function

EpcMultiTagIdentify

Function Describe	public int EpcMultiTagIdentify(int readerAddr,ref byte[,] tag_buf, ref byte tag_cnt, ref byte tag_flag)

Function	EPC GEN2 Multi Tag Identify
Parameter	Paramter Parameter signficance, scope readerAddr Reader address tag_buf Identified data (including tag data,antenna number etc.) tag_cnt Tag count tag_flag Status indicate，0 is read success,1 is read fail
Response	Response value：2001 success

Example	if(Gen2MultiTagIdentify(255，ref tag_buf, ref tag_cnt, ref tag_flag)==2001) MessagBox(“success”);
	else
	MessagBox(“fail”);

EpcRead

Function Describe	public int EpcRead(int readerAddr,e membank, byte wordptr, byte wordcnt, ref byte[] value)
Function	In specified address, read specified length data
Parameter	Paramter Parameter signficance, scope readerAddr Reader address membank Read area ,kill password and access password area is 0，EPC code area 1，USER area is 3 area wordptr Start address wordcnt Read data length，unit is word value Read the tag data
Response	Response value：2001 success

Example	if(EpcRead (255, membank, wordptr, wordcnt, ref value)==2001)
	MessageBox(“success”);
	else
	MessageBox(“fail”);

EpcWrite

Function Describe	public int EpcWrite(int ReaderAddr, byte membank, byte

	wordptr, ushort value)
Function	in specified address, write specified data, one time only write one word
Parameter	Paramter: Parameter signficance, scope ReaderAddr: Reader address membank: Read area，kill password and access password area is 0，EPC code area is 1，USER area is 3 area wordptr: Start address value: Write data in tag
Response	Response value：2001 success

Example	if(EpcWrite (255，membank, wordptr, value)==2001)
	MessageBox(“success”);
	else
	MessageBox(“fail”);

Gen2MultiTagWrite

Function Describe	public int Gen2MultiTagWrite(int ReaderAddr, int membank, int wordaddr, int wordLen, string strValue, ref int writeCount)
Function	EPC Gen2 Multi tag write
Parameter	Paramter: Parameter signficance, scope ReaderAddr: Reader address membank: Read area ,kill password and access password area is 0，EPC code area is 1，USER area is 3 area wordaddr: Start address wordLen: Write data length strValue: Write character string writeCount: Write success tag count
Response	Response value：2001 success

Example

```
if(EpcWrite (255 , membank, wordaddr , wordLen , strValue , ref
writecount )==2001)
    MessageBox( “success” );
else
    MessageBox( “fail” );
```

Gen2MultiTagRead

Function Describe	public int Gen2MultiTagRead(int ReaderAddr, byte MembankMask, byte ResWordPtr, byte ResWordCnt, byte EpcWordPtr, byte EpcWordCnt, byte TidWordPtr, byte TidWordCnt, byte UserWordPtr, byte UserWordCnt, ref int ReadCnt)

Function

EPC Gen2 Multi Tag Read

Parameter

Paramter	Parameter signficance, scope
ReaderAddr	Reader address
MembankMask	Storage area selection, show by mask。From 1 to 4 respectively indicated Reserve area、EPC area、TID area and USER area
ResWordPtr	Reserve area read word address
ResWordCnt	Read word count
EpcWordPtr	EPC area read start address
EpcWordCnt	Read word count
TidWordPtr	TID area read start address
TidWordCnt	Read word count
UserWordPtr	User area read start address
UserWordCnt	Read word count
ReadCnt	Read data count

Response

Response value：2001 success

Example

```
if(EpcWrite (255 , MembankMask, ResWordPtr,
ResWordCnt, EpcWordPtr, EpcWordCnt, TidWordPtr,
```

```
TidWordCnt, UserWordPtr, UserWordCnt, ref
ReadCnt)==2001)
    MessageBox( “success” );
else
    MessageBox( “fail” );
```

Gen2SecLock

Functions describtion	public int Gen2SecLock(int ReaderAddr, uint AccPassWord, byte Membank, byte Level)
Functions	EPC GEN2 lock tag，Make EPC tag can be safe write with right password
Parameter	parameter: Parameter meaning, range ReaderAddr: Reader address AccPassWord: Tag passwords area Membank: Lock and write area，0 is reserve area，1 is EPC，2is TID，3 is uear area Level: The level of safe lock（lock level，0 is no lock，1no lock permanent，2safe lock， 3 full lock。）
Returned value	Returned value：2001 means succeed。

Example	if(Gen2SecLock(255, AccPassWord, Membank, Level)==2001)
	MessageBox(“succeed”);
	else
	MessageBox(“fail”);

EpcLockTag

Functions discribtion	public int EpcLockTag(int readerAddr,byte MemBank)
Function	EPC GEN2 lock tag，Make EPC tag can be safe write with right password

Parameter	parameter	Parameter meaning, range
	readerAddr	Reader address
	Membank	byMemBank lock and write area，0 is reserve area，1 is EPC，2 is TID，3 is user area
Ruturned value	Returned value：2001 means is secceed	

Example	if(EpcLockTag (255，MemBank)==2001)
	MessageBox(“succeed”);
	else
	MessageBox(“fail”);	

EpcInitEpc

Functions description	public int EpcInitEpc(int readerAddr,byte bit_cnt)
Function	Init EPC length，default init is96 bit (6 word)，and voluation completely is 0

Parameter	parameter	Parameter meaning, range
	readerAddr	Reader address
	bit_cnt	Init data length
Returned value	Returned value：2001means succeed	

Example	if(EpcInitEpc (255，bit_cnt)==2001)
	MessageBox(“succeed”);
	else
	MessageBox(“fail”);	

Gen2SecWrite

Functions description	public int Gen2SecWrite(int ReaderAddr, uint AccPassWord, byte Membank, byte WordAddr, ushort Value)
Function	EPC GEN2 tag data sale write

Parameter 参数	parameter	Parameter meaning,range
	readerAddr	Reader address
	AccPassWord	Tag password area
	Membank	byMemBank lock and write area，0 is reserve area，1 is EPC，2 is TID，3 is User area
	WordAddr	Safe write start address
	Value	Safe write data
Returned value	Returned value：2001 means succeed	

Example	if(Gen2SecWrite(255, AccPassWord, Membank, WordAddr, Value)==2001)
	MessageBox(“secceed”);
	else
	MessageBox(“fail”);	

Gen2SecRead

Functions describtion	public int Gen2SecRead(int ReaderAddr, uint AccPassWord, byte Membank, byte WordAddr, byte WordCnt, ref byte[] value)
Function	EPC GEN2 tag data safe write
Parameter	Parameter: Parameter meaning. range readerAddr: Reader address AccPassWord: tag password data Membank: byMemBank lock and write area 域，0 is resever area，1 is EPC，2 is TID，3 is User area WordAddr: Safe read start address WordCnt: safe read data length value: tag value
Returned value	Returned value：2001means succeed。

Example	if(Gen2SecRead(255, AccPassWord, Membank, WordAddr, WordCnt, ref value)==2001)
	MessageBox(“succeed”);
	else
	MessageBox(“fail”);

Gen2SelectConfig

Functions describtion	public int Gen2SelectConfig(int ReaderAddr, int Action, int Membank, int wordAddr, int wordCnt, string[] words)
Function	GEN2 config tag select parameter
Parameter	parameter: Parameter meaning, range readerAddr: Reader address Action: 0 is select mached，1 is select unmach Membank: byMemBank mached area，0 is reserve area，1 is EPC，2 is TID，3 is user area WordAddr: Select data start address WordCnt: select data length words: select data
Returned value	Returned value：2001 means succeed

Example	if(Gen2SelectConfig(255, action, Membank, WordAddr, WordCnt, words)==2001)
	MessageBox(“succeed”);
	else
	MessageBox(“fail”);

Gen2SetAccPwd

Functions description	public int Gen2SetAccPwd(uint AccPassWord)
Function	EPC GEN2 tag password setting
Parameter	parameter: Parameter meaning, range AccPassWord: Password data
Returned value	Returned value：2001 means succeed
Example	if(Gen2SecRead(AccPassWord)==2001) MessageBox(“secceed”);

```
else
    MessageBox( “fail” );
```

Gen2KillTag

Functions description	public int Gen2KillTag(int ReaderAddr, uint AccPassWord)
Function	EPC GEN2 kill tag
Parameter	parameter Parameter meaning, range ReaderAddr Reader address AccPassWord Kill tag password
Returned value	Returned value：2001 means succeed。

Example	if(Gen2KillTag (255，AccPassWord)==2001)
	MessageBox(“succeed”);
	else
	MessageBox(“fail”);

1-4.SO18000-6B tag data processing function ClearIdBuf

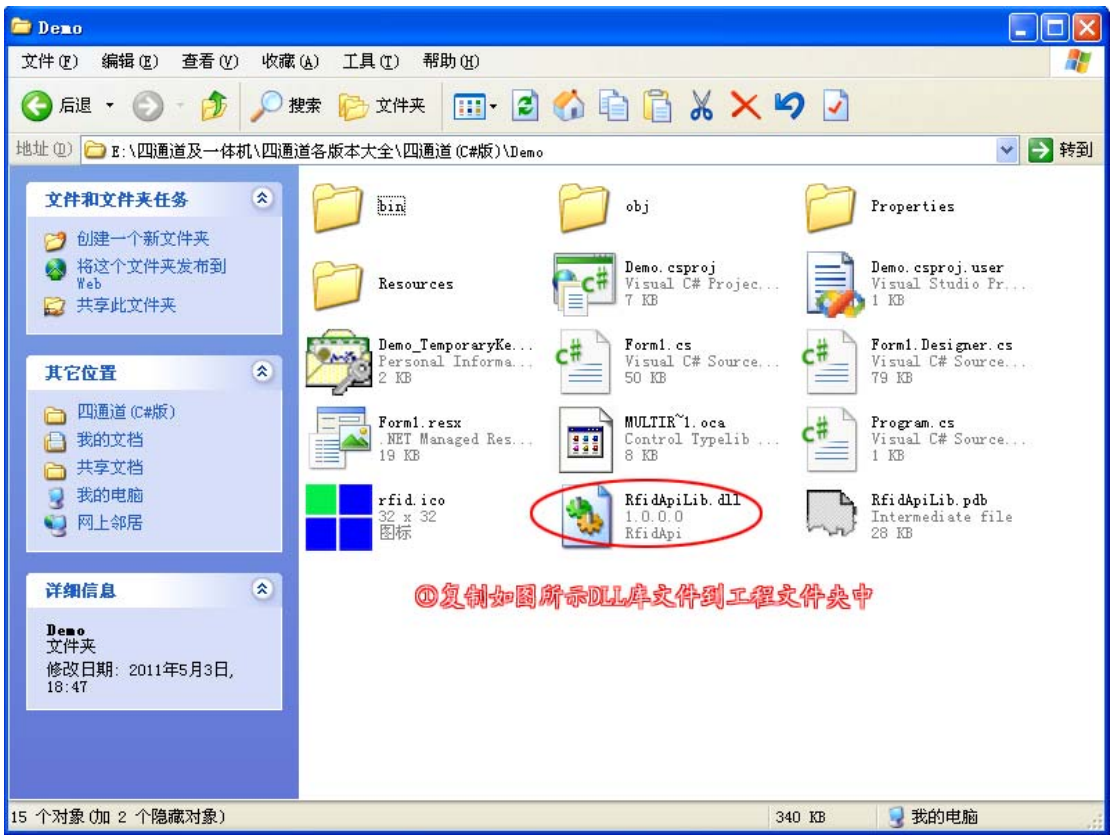
Function description	public int ClearIdBuf(int readerAddr)
Function	Clear reader buffer(tag data), can use this function before multi-tag identify
Parameter	readerAddr Reader address
Returned value	Returned value：2001 means succeed

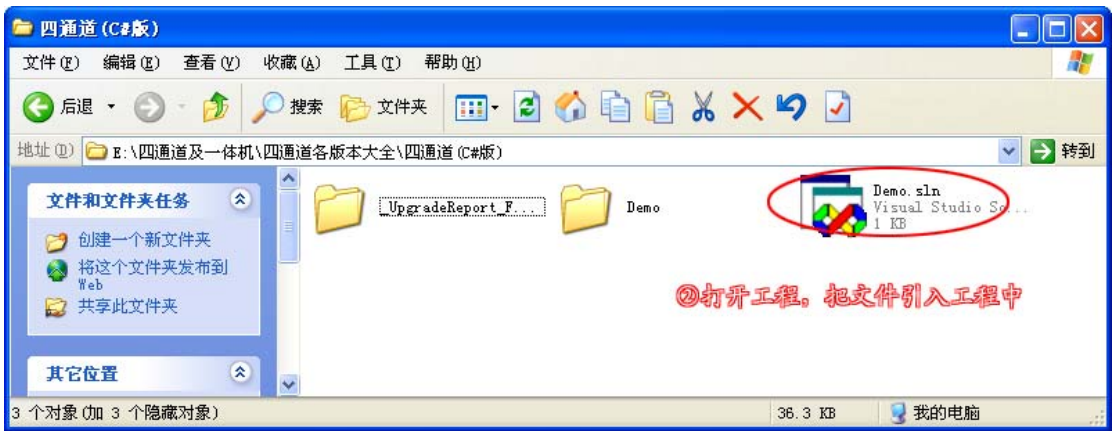
Example	if(ClearIdBuf (255)==2001)
	MessageBox(“succeed”);
	else
	MessageBox(“fail”);

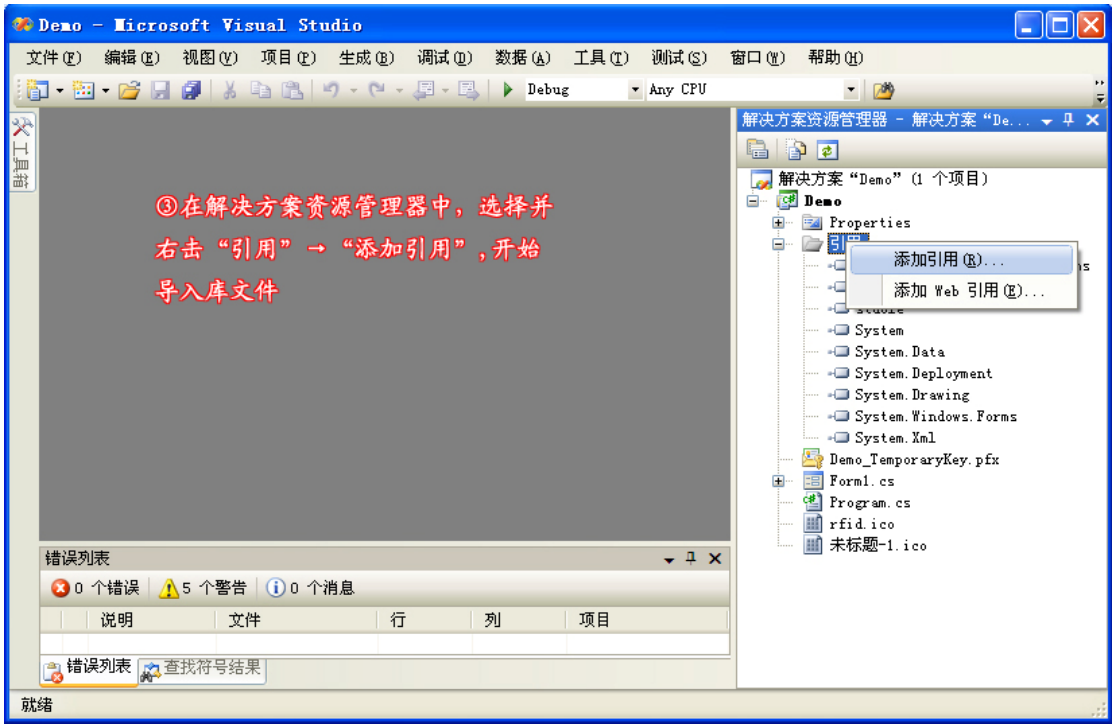
2.Source program sample

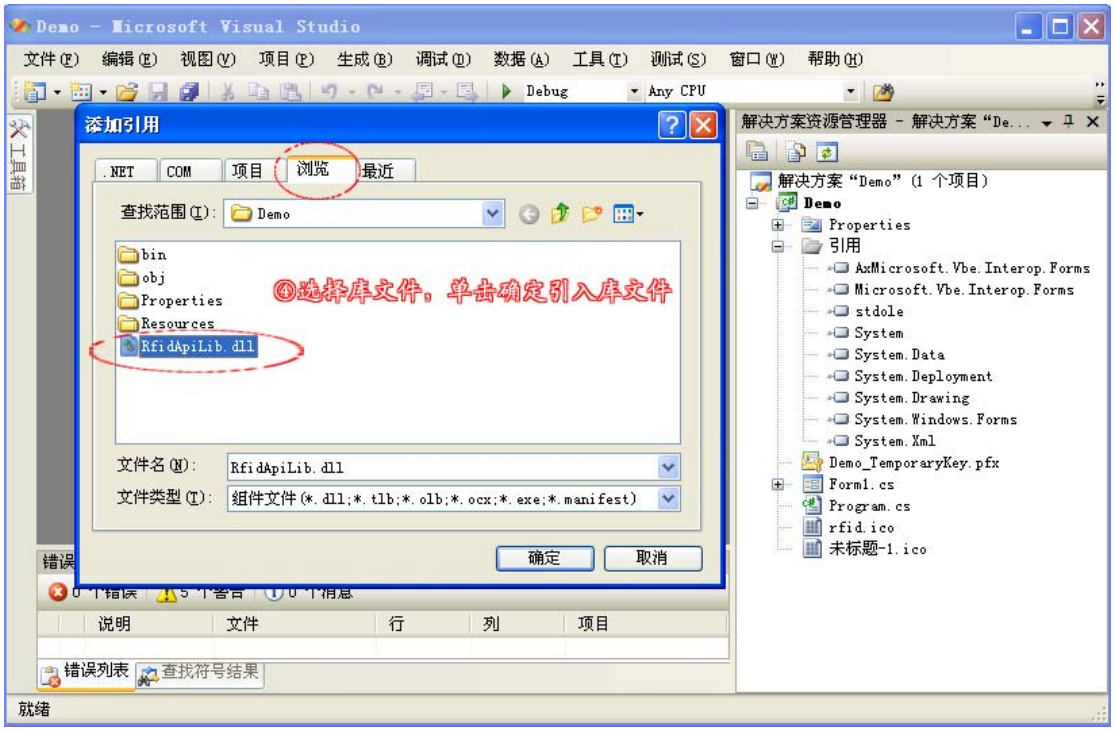
2-1．Set up a project

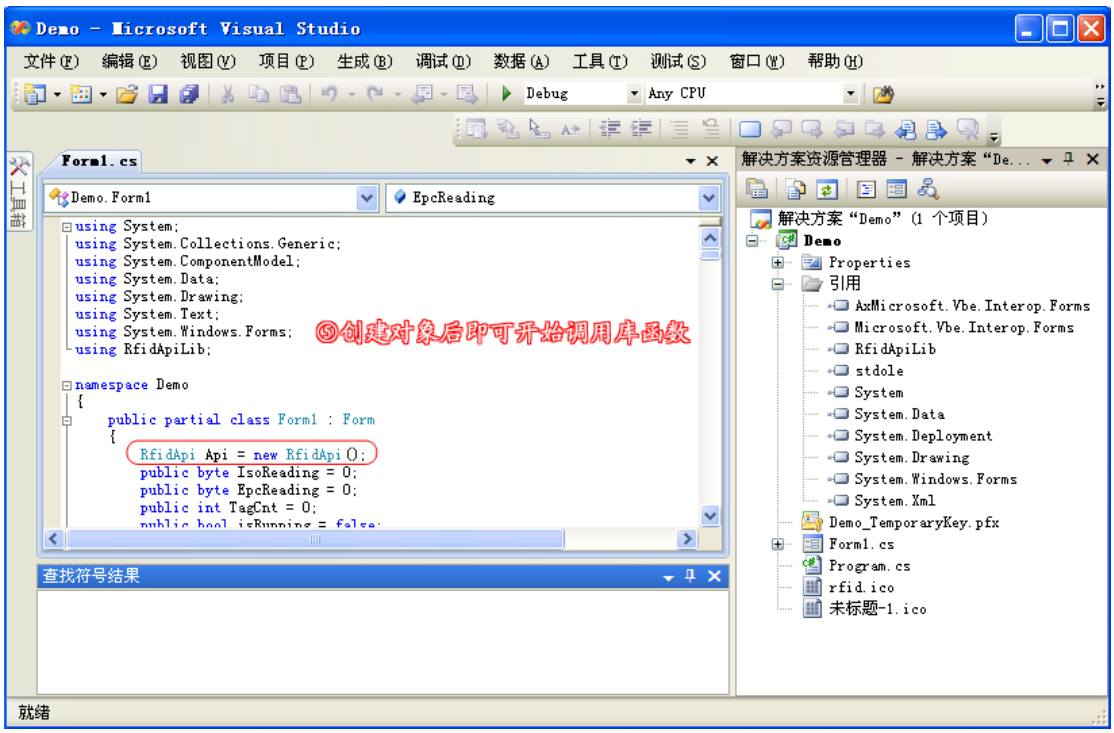
2-2．Add RfidApiLib.dll quote into project。(as shown as figure)






Demo - Microsoft Visual Studio
③在解决方案资源管理器中，选择并右击“引用”→“添加引用”，开始导入库文件
添加引用(R)...
添加 Web 引用(E)...
解决方案资源管理器 - 解决方案“De...
解决方案“Demo”(1 个项目)
Demo
Properties
引用
System
System.Data
System.Deployment
System.Drawing
System.Windows.Forms
System.Xml
Demo_TemporaryKey.pfx
Form1.cs
Program.cs
rfid.ico
未标题-1.ico
错误列表
0 个错误
5 个警告
0 个消息
说明
文件
行
列
项目
错误列表
查找符号结果
就绪

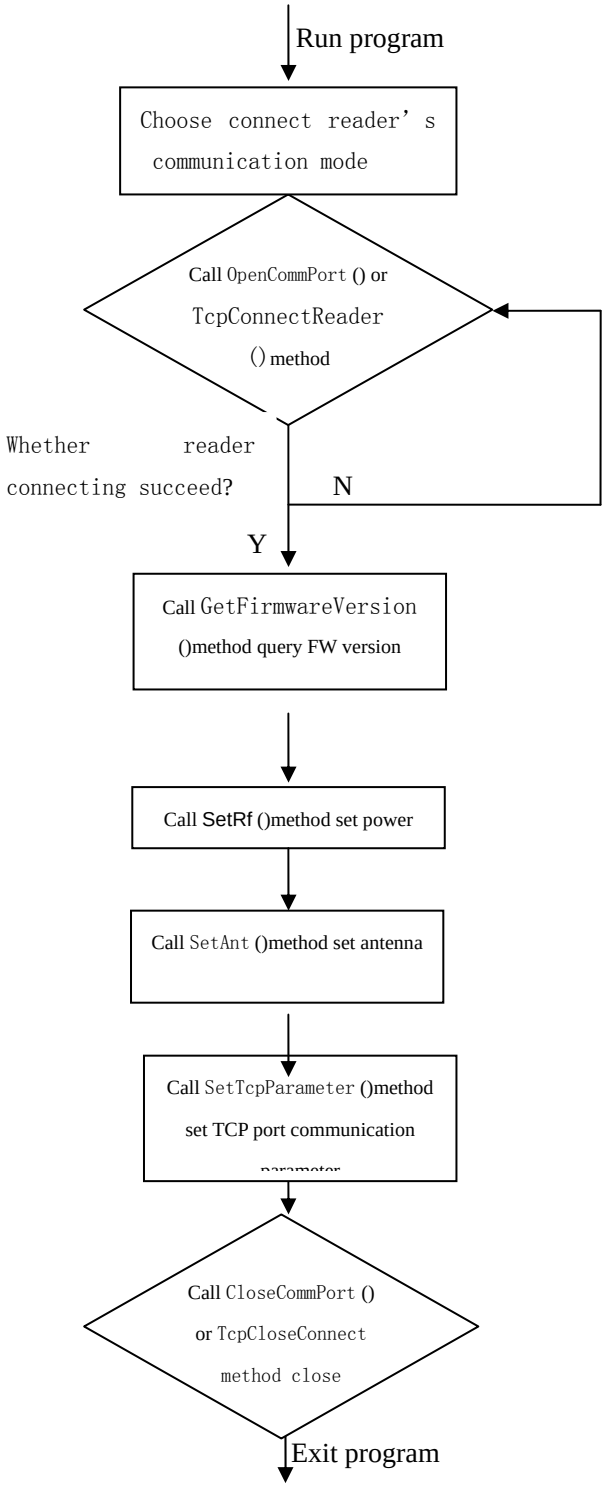

Demo - Microsoft Visual Studio
添加引用
.NET
COM
项目
浏览
最近
查找范围(I):
Demo
bin
obj
Properties
Resources
RfidApiLib.dll
④选择库文件，单击确定引入库文件
文件名(N):
RfidApiLib.dll
文件类型(T):
组件文件(*.dll;*.tlb;*.olb;*.ocx;*.exe;*.manifest)
确定
取消
解决方案资源管理器 - 解决方案“De...
解决方案“Demo”(1 个项目)
Demo
Properties
引用
AxMicrosoft.Vbe.Interop.Forms
Microsoft.Vbe.Interop.Forms
stdole
System
System.Data
System.Deployment
System.Drawing
System.Windows.Forms
System.Xml
Demo_TemporaryKey.pfx
Form1.cs
Program.cs
rfid.ico
未标题-1.ico
说明
文件
行
列
项目
错误列表
查找符号结果
就绪



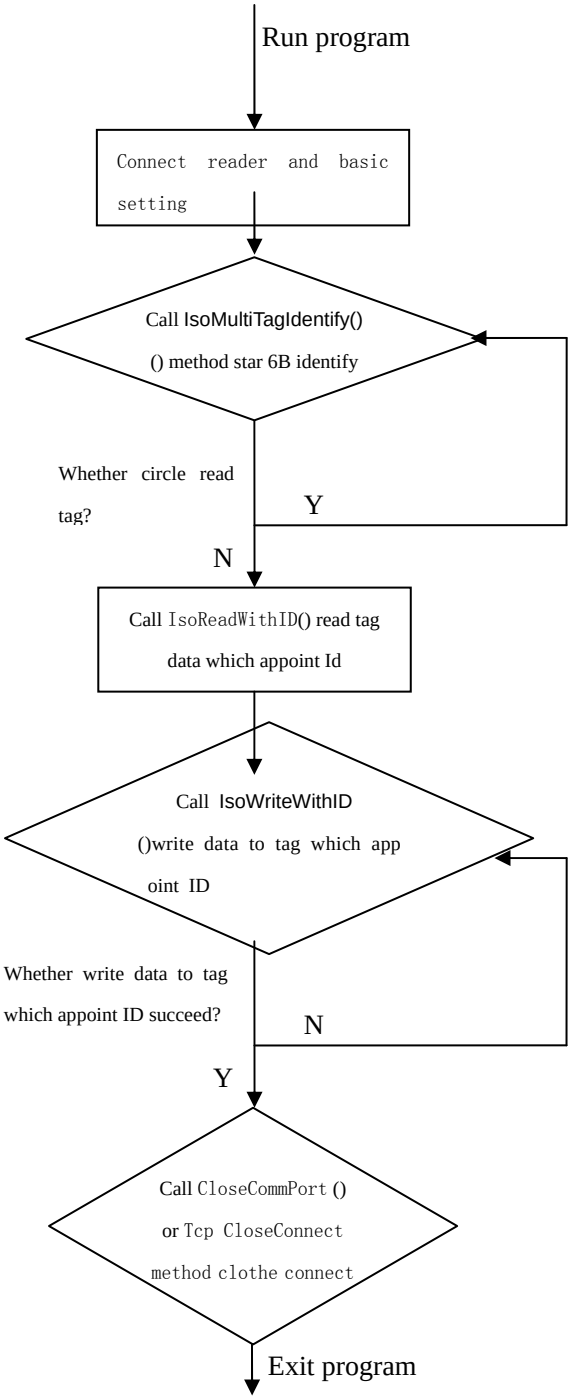
Can call functions which in dll directly after addRfidApiLib.dll quote in to project and set up a new object.

2-3 . Call library function to achieve relevant function

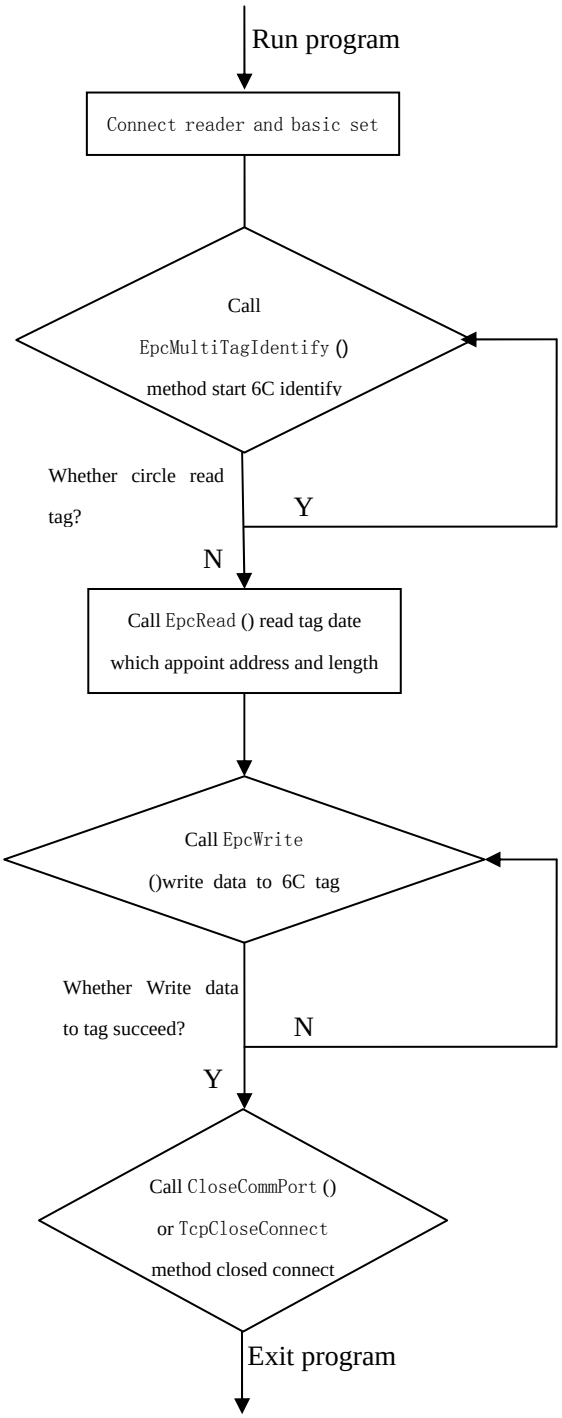
Reader basic setting and manage flow chart as follows:



ISO18000-6B tag operation flow chart as follows:



EPC GEN2 operation flow chart as follows:



(1) . Call OpenCommPort () method connect to reader。

```
int status;
byte v1 = 0;
byte v2 = 0;
string s = "";
status =
Api.OpenCommPort(cCommPort.SelectedItem.ToString(),Convert.ToInt32(cBaudrate.Text));
if (status != RfidApi.SUCCESS_RETURN)
```

```
{
    lInfo.Items.Add("Open Comm Port Failed!  ");
    return;
}
status = Api.GetFirmwareVersion(ref v1,ref v2);
if (status != RfidApi.SUCCESS_RETURN)
{
    lInfo.Items.Add("Can not connect with the reader!  ");
    Api.CloseCommPort();
    return;
}
lInfo.Items.Add("Connect the reader success! ");
s = string.Format("The reader's firmware version is:V{0:d2}.{1:d2}", v1, v2);
s = s + "   ";
lInfo.Items.Add(s);
```

(2). Call SetRf () method set power。

```
byte pwr = 0;
byte freq = 0;

int status;
pwr = (byte)(tRfPwr.Value);
freq = (byte)(cRfFreq.SelectedIndex);
status = Api.SetRf(pwr, freq);
if (status != RfidApi.SUCCESS_RETURN)
{
    lInfo.Items.Add("Set Rf settings failed!  ");
    return;
}
lInfo.Items.Add("Set Rf settings success! ");
```

(3). Call SetAnt () method set antenna。

```
byte ant_sel = 0;
byte antH = 0;
int status;

if (ant1.Checked)
    ant_sel |= 0x01;
if (ant2.Checked)
    ant_sel |= 0x02;
if (ant3.Checked)
    ant_sel |= 0x04;
if (ant4.Checked)
    ant_sel |= 0x08;
```

```
status = Api.SetAnt(ant_sel, antH);
if (status != RfidApi.SUCCESS_RETURN)
{
    lInfo.Items.Add("Set ant failed!  ");
    return;
}
lInfo.Items.Add("Set ant success! ");
```

(4) . Call SetTcpParameter () method set TCP port communication parameter

```
int status=0;
string strIp="";
string strMark = "";
string strGate = "";
int nTcpPort =0;
try {
   strIp = txtNetIP.Text;
   strMark = txtSubNet.Text;
   strGate = txtDefaultGate.Text;
   nTcpPort = int.Parse(txtTcpPort.Text);
}
catch (Exception)
{
    lInfo.Items.Add("Please input all the parameter !  ");
    return;
}
status=Api.SetTcpParameter(strIp,strMark, strGate, nTcpPort);
if (status != RfidApi.SUCCESS_RETURN) {
    lInfo.Items.Add("Setting the TcpParameter Fail,please try again.  ");
    return;
}
lInfo.Items.Add("Setting the TcpParameter successful. ");
```

(5) . Call IsoMultiTagIdentify() () method start 6B tag identify。

```
int status;
int i,j;
byte[,] IsoBuf = new byte[100,14];
byte tag_cnt = 0;
string s = "";
string s1 = "";

status = Api.IsoMultiTagIdentify(ref IsoBuf, ref tag_cnt);
if (tag_cnt > 0)
{
```

```
    for (i = 0; i < tag_cnt; i++)
    {
        s1 = string.Format("NO.{0:D}:", TagCnt);
        s1 += string.Format("[ANT{0:D}]", IsoBuf[i, 1] + 1);
        for (j = 2; j < 10; j++)
        {
            s = string.Format("{0:X2} ", IsoBuf[i, j]);
            s1 += s;
        }
        s1 = s1.Substring(0, s1.Length - 1);
        if (lInfo.Items.Count > 1000)
            lInfo.Items.Clear();
        lInfo.Items.Add(s1);
        TagCnt++;

    }
     }
```

(6). Call IsoReadWithID() read data from the tag which was appoint ID number.

```
int addr;
int len;
int i = 0;
int status = 0;
byte byAntenna = 0;
byte[] TagID = new byte[16];
byte[] value = new byte[32];
string s = "The data is:";
string s1 = "";
try
{
    addr = int.Parse(tIsoAddr.Text);
    len = int.Parse(tIsoCnt.Text);
}
catch (Exception)
{
    lInfo.Items.Add("Please input ByteAddr and ByteCnt  ");
    return;
}
string hexValues = txtTagID.Text;
string[] hexValuesSplit = hexValues.Split(' ');
try
{
    foreach (String hex in hexValuesSplit)
    {
```

```
            int x = Convert.ToInt32(hex, 16);
            TagID[i++] = (byte)x;
        }
    }
    catch (Exception)
    {
        lInfo.Items.Add("Please input Tag ID needed  ");
        return;

    }
    if (i != 8)
    {
        lInfo.Items.Add("Please input Tag ID needed  ");
        return;
    }

    for (i = 0; i < len; )
    {
        status = Api.IsoReadWithID(TagID,(byte)addr, ref value,ref byAntenna);
        if (status != RfidApi.SUCCESS_RETURN)
        {
            lInfo.Items.Add("Read failed!  ");
            return;
        }

        for (int j = 0; j < 8; j++)
        {
            s1 = string.Format("{0:X2}", value[j]);
            s += s1;
            if (i + j >= len - 1)
                break;
        }
        i += 8;
    }
    if (status == 2001)
    {
        s += "   ";
        lInfo.Items.Add("Read success! ");
        lInfo.Items.Add(s);
         }
```

（7）. IsoWriteWithID () write data to the tag which was appoint ID number。

```
int addr;
int len;
```

```
int i = 0;
int status = 0;
byte byAntenna = 0;
byte[] TagID = new byte[16];
byte[] value = new byte[32];
string s = "The data is:";
string s1 = "";
try
{
    addr = int.Parse(tIsoAddr.Text);
    len = int.Parse(tIsoCnt.Text);
}
catch (Exception)
{
    lInfo.Items.Add("Please input ByteAddr and ByteCnt  ");
    return;
}
string hexID = txtTagID.Text;
string[] hexIDSplit = hexID.Split(' ');
try
{
    foreach (String hex in hexIDSplit)
    {
        int x = Convert.ToInt32(hex, 16);
        TagID[i++] = (byte)x;
    }
}
catch (Exception)
{
    lInfo.Items.Add("Please input Tag ID needed  ");
    return;

}
string hexValues = tIsoData.Text;
string[] hexValuesSplit = hexValues.Split(' ');
try
{
    i = 0;
    foreach (String hex in hexValuesSplit)
    {
        if (hex != "")
        {
            int x = Convert.ToInt32(hex, 16);
            value[i++] = (byte)x;
```

```
            }
        }
    }
    catch (Exception)
    {
        lInfo.Items.Add("Please input data needed  ");
        return;

    }
    if (i != len)
    {
        lInfo.Items.Add("Please input data needed  ");
        return;
    }
    for (i = 0; i < len; i++)
    {
        status = Api.IsoWriteWithID(TagID,(byte)(addr + i), value[i]);
        if (status != RfidApi.SUCCESS_RETURN)
        {
            lInfo.Items.Add("Write failed!  ");
            return;
        }
    }
    lInfo.Items.Add("Write success! ");
```

(8). Call EpcMultiTagIdentify () method start 6C tag identify。

```
int status;
int i, j;
byte[,] IsoBuf = new byte[100, 14];
byte tag_cnt = 0;
string s = "";
string s1 = "";
byte tag_flag = 0;

if (!isNetConnect)
    return;
try
{
    status = Api.EpcMultiTagIdentify(ref IsoBuf, ref tag_cnt, ref tag_flag);
    if (status == 2009)
    {
        isNetConnect = false;
        return;
    }
```

```
        }
        catch (Exception ex)
        {
            System.Diagnostics.Debug.WriteLine(ex.ToString());
            isNetConnect = false;
            return;

        }
        if (tag_flag == 1)
            this.BackColor = Color.MediumBlue;
        else
            this.BackColor = Color.MidnightBlue;
        if (tag_cnt >= 100)
            return;
        if (tag_cnt > 0)
        {
            try
            {
                for (i = 0; i < tag_cnt; i++)
                {
                    s1 = string.Format("NO.{0:D}: ", TagCnt);
                    s1 += string.Format("[ANT{0:D}]", IsoBuf[i, 1]+1);
                    for (j = 2; j < 14; j++)                        {
                        s = string.Format("{0:X2} ", IsoBuf[i, j]);
                        s1 += s;
                    }
                    if (lInfo.Items.Count >= 1000)
                        lInfo.Items.Clear();

                    s1 = s1.Substring(0, s1.Length - 1);
                    lInfo.Items.Add(s1);
                    TagCnt++;

                }
            }
            catch
            {
            }
        }
```

(9). Call EpcRead () read data from the tag which was appoint address and length。

```
    int membank;
        int wordptr;
        int wordcnt;
```

```
int status = 0;
byte[] value = new byte[16];
string s = "The data is: ";
string s1 = "";
membank = cEpcMembank.SelectedIndex;
wordptr = cEpcWordptr.SelectedIndex;
wordcnt = cEpcWordcnt.SelectedIndex + 1;
status = Api.EpcRead((byte)membank, (byte)wordptr, (byte)wordcnt, ref value);
if (status != RfidApi.SUCCESS_RETURN)
{
    lInfo.Items.Add("Read failed!  ");
    return;
}
else
{
    for (int i = 0; i < wordcnt * 2; i++)
    {
        s1 = string.Format("{0:X2}", value[i]);
        s += s1;
    }
    lInfo.Items.Add("Read success! ");
    s += "   ";
    lInfo.Items.Add(s);
}
```

(10) . Call EpcWrite () write data to 6C tag。

```
ushort[] value = new ushort[16];
        int i = 0;
        byte membank;
        byte wordptr;
        byte wordcnt;
        int status;
        string hexValues;

        membank = (byte)(cEpcMembank.SelectedIndex);
        wordptr = (byte)(cEpcWordptr.SelectedIndex);
        wordcnt = (byte)(cEpcWordcnt.SelectedIndex+1);

        hexValues = tEpcData.Text;
        string[] hexValuesSplit = hexValues.Split(' ');
        {
            foreach (String hex in hexValuesSplit)
            {
                if (hex != "")
```

```
            {
                int x = Convert.ToInt32(hex, 16);
                value[i++] = (ushort)x;
            }
        }
    }
    if (i != wordcnt)
    {
        lInfo.Items.Add("Please input data needed  ");
        return;
    }
    for(byte j = 0; j < wordcnt; j++)
    {
        status = Api.EpcWrite(membank,(byte)(wordptr+j),value[j]);
        if (status != RfidApi.SUCCESS_RETURN)
        {
            lInfo.Items.Add("Write failed!  ");
            return;
        }
    }
    lInfo.Items.Add("Write success! ");
```

(11). Call Gen2SecLock () method to lock EPC GEN2 tag，make EPC tag can be safe write with right password。

```
byte membank;
    byte pwdLevel;

    int status = 0;
    byte[] value = new byte[16];

    string s = "";
    if (tEpcAccess.TextLength != 8)
    {
        lInfo.Items.Add("Access Password length not enough  ");
        return;
    }
    uint unAccPwd;
    switch(cEpcMembank.SelectedIndex)
    {
        case 0:
            membank=3;
            break;
        case 1:
```

```
            membank=2;
            break;
        case 2:
            membank=1;
            break;
        case 3:
            membank=0;
            break;
        default:
            membank=2;
            break;
    }
    pwdLevel = (byte)(cmbLevel.SelectedIndex);

    unAccPwd = Convert.ToUInt32(tEpcAccess.Text, 16);
    status = Api.Gen2SecLock(unAccPwd, membank,pwdLevel);
    if (status != RfidApi.SUCCESS_RETURN)
    {
        lInfo.Items.Add("Lock EPC tag failed!  ");
        return;
    }
    else
    {
        lInfo.Items.Add("Lock EPC tag success! ");
        lInfo.Items.Add(s);
    }
```

(12) . Call Gen2SecWrite() method to safe write data in EPC GEN2 tag。

```
ushort[] value = new ushort[16];
    int i = 0;
    byte membank;
    byte wordptr;
    byte wordcnt;
    int status;
    string hexValues;

    membank = (byte)(cEpcMembank.SelectedIndex);
    wordptr = (byte)(cEpcWordptr.SelectedIndex);
    wordcnt = (byte)(cEpcWordcnt.SelectedIndex + 1);
    if (tEpcAccess.TextLength != 8)
    {
        lInfo.Items.Add("Access Password length not enough  ");
        return;
    }
```

```
uint unAccPwd;
unAccPwd = Convert.ToUInt32(tEpcAccess.Text, 16);

hexValues = tEpcData.Text;
string[] hexValuesSplit = hexValues.Split(' ');
foreach (String hex in hexValuesSplit)
{
    // Convert the number expressed in base-16 to an integer.
    if (hex != "")
    {
        int x = Convert.ToInt32(hex, 16);
        value[i++] = (ushort)x;
    }
}
if (i != wordcnt)
{
    lInfo.Items.Add("Please input data needed  ");
    return;
}
for (byte j = 0; j < wordcnt; j++)
{
    status = Api.Gen2SecWrite(unAccPwd,membank, (byte)(wordptr + j), value[j]);
    if (status != RfidApi.SUCCESS_RETURN)
    {
        lInfo.Items.Add("Write failed!  ");
        return;
    }
}
lInfo.Items.Add("Write success! ");
```

(13). Call Gen2KillTag () method to kill tag。

```
int status = 0;
byte[] value = new byte[16];

string s = "";
if (tEpcAccess.TextLength != 8)
{
    lInfo.Items.Add("Access Password length not enough  ");
    return;
}
uint unAccPwd;
unAccPwd = Convert.ToUInt32(tEpcAccess.Text, 16);
status = Api.Gen2KillTag(unAccPwd);
if (status != RfidApi.SUCCESS_RETURN)
```

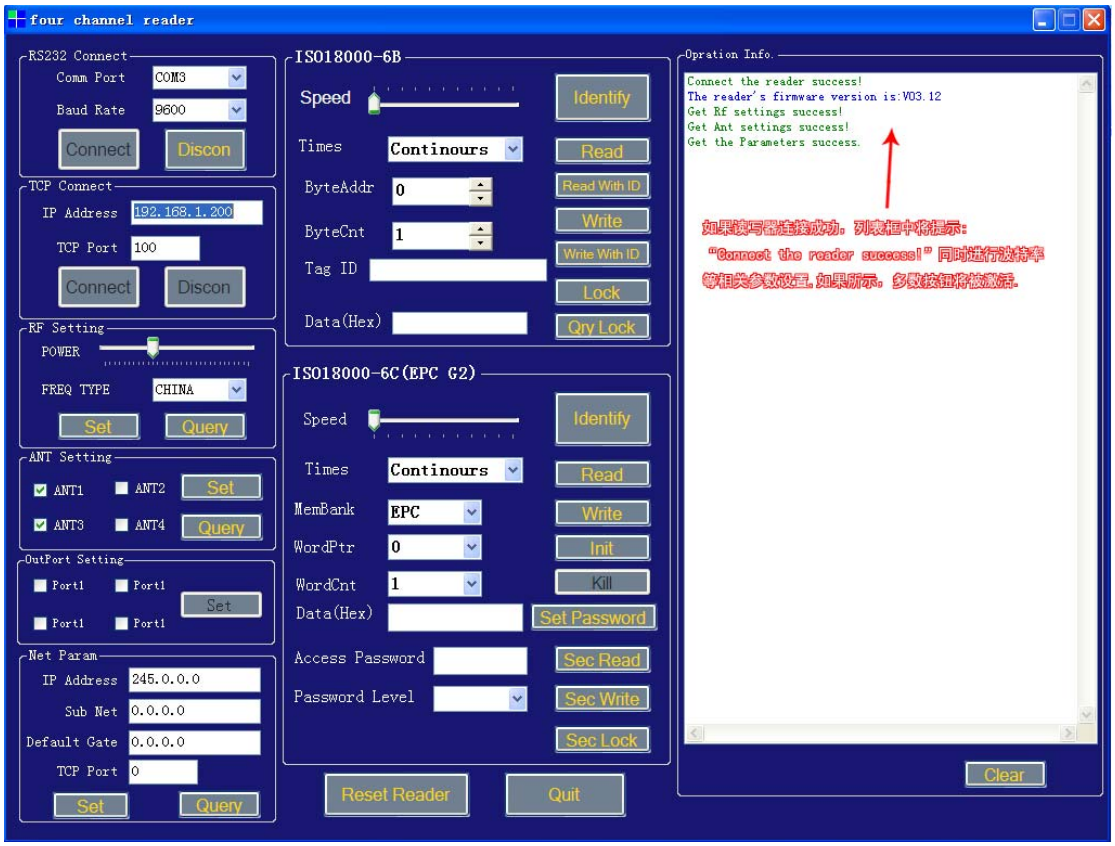
```
{
    lInfo.Items.Add("Set Password failed!  ");
    return;
}
else
{
    lInfo.Items.Add("Set Password success! ");
    lInfo.Items.Add(s);
}
```

3. Operation method and steps of presentation software

3-1．Start demo software

Click on the software icon 

, start the software , click on the "Connect" to connect to the reader ,if success tips will be shown like figure 3-1 , user can query the current antenna parameters by clicking on **【Query】**, and user can set the antenna transmission power according to actual need by clicking on **【Set】**, user can change the frequency of the antenna from 0dBm to 30dBm according to the actual demand。

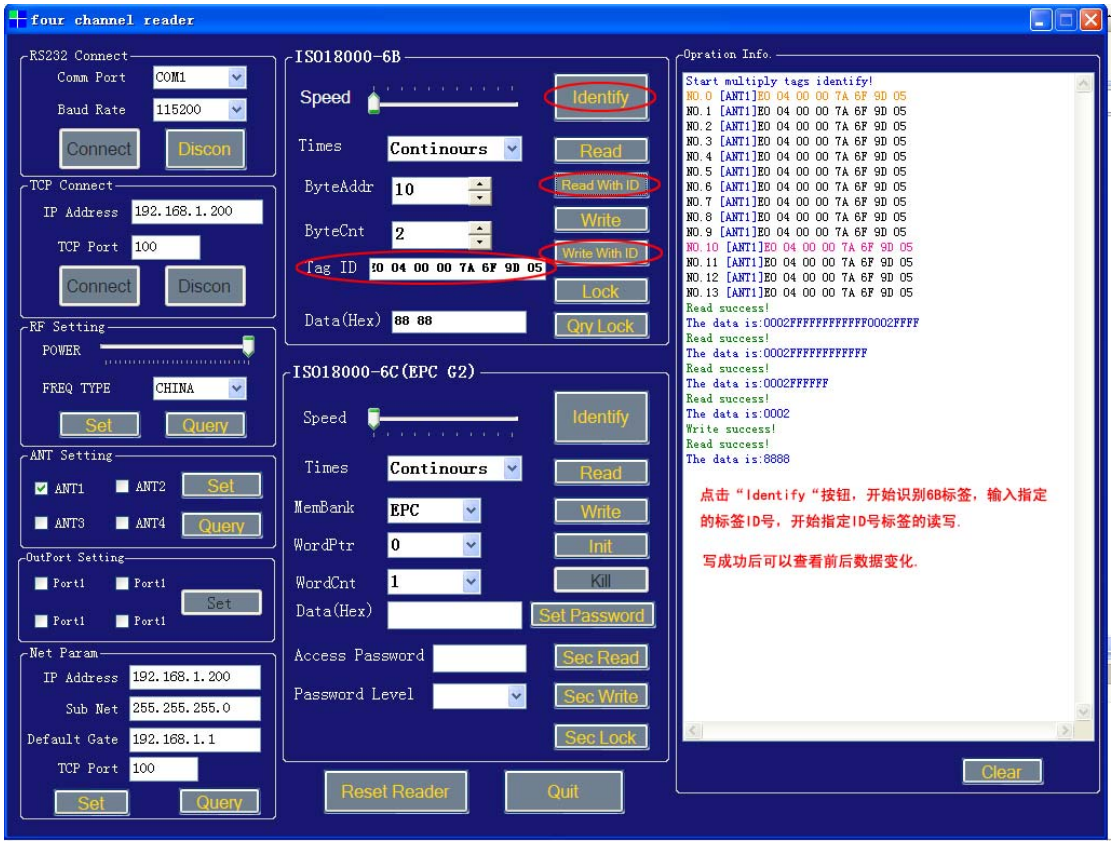


3-1

3-2．ISO18000-6B tag identify

Choose【**ISO18000-6B**】group，click on【**Identify**】，start to identify 6B tags，tags ID number will be displayed in the list，as shown in figure 3-2，user can write data and read data to target tag in bytes；

Notice：Writing data and reading data are effective for single tag only, you must stop identifying tags when you do this.



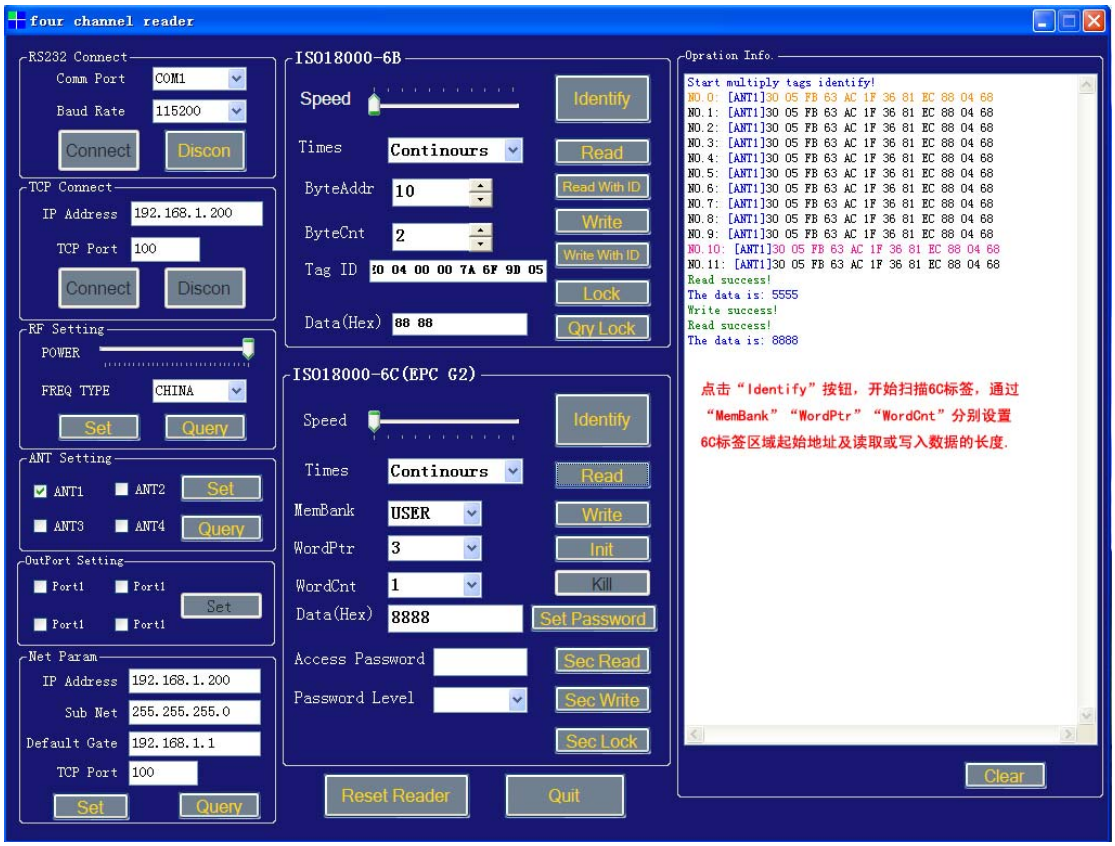
3-2

3-3．EPC Class1 GEN2 tag identify

【Write】 is scanning all tag which in setting antenna range select【EPC GEN2】, click 【Identify】, scanning EPC tag 标签，tag data (EPC code) will display on list，see pcture3-3；

【Read】is reader one tag data, attention, make sure there is only one tag on antenna range, other wise, tag reader may fail；

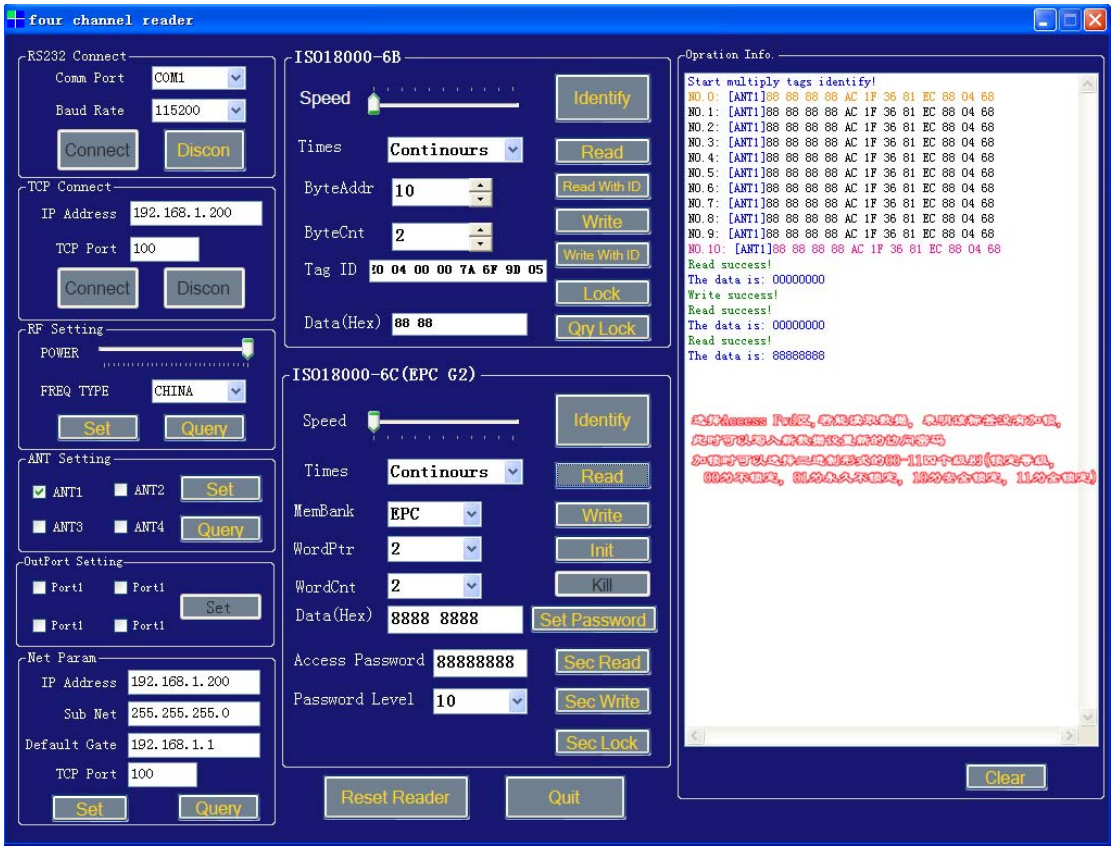
【Write】is write data into tag, click” write” button，when writeEPC, the unit is word，write length is limited，can write 1 byte every time；



3-3

【SecLock】is after set Access Pwd，to safe write EPC tag;

【SecWrite】the level of tag lock is 10，tag write should choose “Secured Password” and input the righh password can write tag success, see picture 3-4;


four channel reader
RS232 Connect
Comm Port
COM1
Baud Rate
115200
Connect
Discon
TCP Connect
IP Address
192.168.1.200
TCP Port
100
Connect
Discon
RF Setting
POWER
FREQ TYPE
CHINA
Set
Query
ANT Setting
ANT1
ANT2
ANT3
ANT4
Set
Query
OutPort Setting
Port1
Port1
Port1
Port1
Set
Net Param
IP Address
192.168.1.200
Sub Net
255.255.255.0
Default Gate
192.168.1.1
TCP Port
100
Set
Query
ISO18000-6B
Speed
Identify
Times
Continours
Read
ByteAddr
10
Read With ID
ByteCnt
2
Write
Tag ID
Write With ID
Lock
Data(Hex)
88 88
Qry Lock
ISO18000-6C(EPC G2)
Speed
Identify
Times
Continours
Read
MemBank
EPC
Write
WordPtr
2
Init
WordCnt
2
Kill
Data(Hex)
8888 8888
Set Password
Access Password
88888888
Sec Read
Password Level
10
Sec Write
Sec Lock
Reset Reader
Quit
Opration Info.
Start multiply tags identify!
NO.0: [ANT1]88 88 88 88 AC 1F 36 81 EC 88 04 68
NO.1: [ANT1]88 88 88 88 AC 1F 36 81 EC 88 04 68
NO.2: [ANT1]88 88 88 88 AC 1F 36 81 EC 88 04 68
NO.3: [ANT1]88 88 88 88 AC 1F 36 81 EC 88 04 68
NO.4: [ANT1]88 88 88 88 AC 1F 36 81 EC 88 04 68
NO.5: [ANT1]88 88 88 88 AC 1F 36 81 EC 88 04 68
NO.6: [ANT1]88 88 88 88 AC 1F 36 81 EC 88 04 68
NO.7: [ANT1]88 88 88 88 AC 1F 36 81 EC 88 04 68
NO.8: [ANT1]88 88 88 88 AC 1F 36 81 EC 88 04 68
NO.9: [ANT1]88 88 88 88 AC 1F 36 81 EC 88 04 68
NO.10: [ANT1]88 88 88 88 AC 1F 36 81 EC 88 04 68
Read success!
The data is: 00000000
Write success!
Read success!
The data is: 00000000
Read success!
The data is: 88888888
Clear