STA IR 0507 E RFID Reader SDK Instruction

DLL File Name: MR6100Api.dll

1 Functions

1-1. Reader Management Functions

TcpConnectReader

Function instructions	public int TcpConnectReader(string ip, int port)	
Function	connect reader via TCP/IP	
Parameter	ip	Reader IP address is an expression of IP address's String format, the IP address has to be the same Network segment with PC (Reader's default IP is 192.168.1.200)
	port	port number range (from 1 to 65535)
return value	Success Return value is 2001	

Example	If(TcpConnectReader ("192.168.1.200",100)==2001)
	MessageBox("Success");
	Else
	MessageBox("Failed");	

OpenCommPort

Function instructions	public int OpenCommPort(string strPort, int nBoud)	
function	Initialize reader connection port and config reader parameters.	
parameter	strPort	Comm port Num. (COM1、COM2、COM3……)
	nBoud	Comm port Baudrate(9600、19200、38400、57600、115200)
return value	Success Return value is 2001	

example	If(OpenCommPort ("COM1",9600)==2001)
	MessageBox("success");
	Else
	MessageBox("failed");	

TcpCloseConnect

Function instructions	public int TcpCloseConnect()
function	Close reader TCP port, and disconnect reader and pc.
parameter	N/A
return value	Success Return value is 2001
example	If(TcpCloseConnect ()==2001) MessageBox("success");

	Else MessageBox("failed");

CloseCommPort

Function instructions	public void CloseCommPort()
function	Close reader Comm port, and disconnect reader and pc
parameter	N/A
return value	N/A
example	CloseCommPort ();

SetBaudRate

Function instructions	public int SetBaudRate(int nReaderAddr, int nBaudRate)	
function	Set reader Baudrate	
parameter	nBaudRate	Comm port communication Baudrate are: 9600、19200、38400、57600 and 115200 (or 0,1,2,3,4)
	nReaderAddr	Reader address parameter is for RS485 network establishment with fixed reader, default address is 255. (This parameter is invalid to handheld reader and module)
return value	Success 8FD4 return value is 2001	

example	If(SetBaudRate(255, 115200)==2001)
	MessageBox("success");
	Else
	MessageBox("failed");	

ResetReader

Function instructions	public int ResetReader(int readerAddr)
function	reset reader
parameter	Reader address parameter is for RS485 network establishment with fixed reader, default address is 255.(This parameter is invalid to handheld reader and module)
Return value	Success Return value is 2001

example	If(ResetReader(255)==2001)
	MessageBox("success");
	Else
	MessageBox("failed");

ResetParameter

Function instructions	public int ResetParameter (int readerAddr)
function	reset reader's all parameter
parameter	Reader address parameter is for RS485 network establishment with fixed reader, default address is 255. (This parameter is invalid to handheld reader and module)
return value	Success Return value is 2001
example	If(ResetParameter (255)==2001)

	MessageBox ('success'); Else MessageBox('failed');

GetFirmwareVersion

Function instructions	public int GetFirmwareVersion(int readerAddr,ref byte v1, ref byte v2)	
function	Get reader firmware version	
parameter	readerAddr	Reader address
	v1	firmware version's Big Endian
	v2	firmware verison's Little Endian
return value	Success Return value is 2001	

example	If(GetFirmwareVersion(255,ref v1,ref v2)==2001)
	MessageBox('success');
	Else
	MessageBox('failed');	

SetRf

Function instructions	public int SetRf(int readerAddr, int power1, int power2, int power3, int power4)	
function	set reader RF power and frequency	
parameter	readerAddr	Reader address
	Power1- Power4	RF power value (from 0-30dBm), Power1 - Power4 correspond with Antenna1 – Antenna4.
return value	Success Return value is 2001	

example	If(SetRf(255,0, 27,30,30) == 2001)
	MessageBox('success');
	Else
	MessageBox('failed');	

GetRf

Function instructions	public int GetRf(int readerAddr,ref int[] power)	
function	Get reader's RF power and frequency parameter.	
parameter	readerAddr	Reader address
	power	Get Antenna's corresponding RF power, a value of 0 to 30, correspond with 0-30dBm.
return value	Success Return value is 2001	

example	If(GetRf(255,ref power) == 2001)
	MessageBox('success');
	Else
	MessageBox('failed');	

SetAnt

Function instructions	public int SetAnt(int readerAddr, byte Antenna)

function	set antenna number's opening status	
parameter	readerAddr	Reader address
	Antenna	The working antenna, represented in mask. 4 low bits corresond with 4 antennas' status, 1 means antenna open, 0 means antenna no open, 4 high bits have not meaning.
return value	Success Return value is 2001	

example	If(SetAnt (ReaderAddr, ant) == 2001)
	MessageBox('success');
	Else
	MessageBox('failed');	

GetAnt

Function instructions	public int GetAnt(int ReaderAddr, ref byte workAnt, ref byte antState)	
function	Get opened antenna number and its connection status.	
parameter	ReaderAddr	Reader address parameter is for RS485 network establishment with fixed reader, default address is 0XFF. (This parameter is invalid to handheld reader and module).
	workAnt	The working antenna's status, represented in mask
	antState	The available antenna, '1'means available, '0' means disconnect or antenna mismatched
return value	Success Return value is 2001	

example	If(GetAnt (ReaderAddr,ref workAnt，ref antState))== 2001)
	MessageBox('success');
	Else
	MessageBox('failed');	

SetFrequency

Function instructions	public int SetFrequency (int ReaderAddr, int freqNum, int[] points)	
function	Set reader's RF parameter	
parameter	ReaderAddr	Reader address parameter is for RS485 network establishment with fixed reader, default address is 0XFF. (This parameter is invalid to handheld reader and module).
	freqNum	Frequency number, if freqNum≠0, then the frequency is the fixed frequency points, if freqNum=0, then the frequency is the regional frequency type, there are: 0: China; 1: North American; 2: European
	points	Self defined frequency range is from 900MHz to 930MHz, progressively increased by 250KHz.
return value	Success Return value is 2001	
example	If(SetFrequency (ReaderAddr,freqNum，points))== 2001)	

	MessageBox('success'); Else MessageBox('failed');

GetFrequency

Function instructions	public int GetFrequency (int ReaderAddr,ref int freqNum,ref int[] points)	
function	Get reader frequency parameter	
parameter	ReaderAddr	Reader address parameter is for RS485 network establishment with fixed reader, default address is 0XFF. (This parameter is invalid to handheld reader and module)
	freqNum	Frequency number, if freqNum≠0, then the frequency is the fixed frequency points, if freqNum=0, then the frequency is the regional frequency type, there are: 0: China; 1: North American; 2: European
	points	Self defined frequency range is from 900MHz to 930MHz, progressively increased by 250KHz.
return value	Success Return value is 2001	

example	If(GetFrequency (ReaderAddr,ref freqNum , ref points))== 2001)
	MessageBox('success');
	Else
	MessageBox('failed');	

GetFastTagMode

Function instructions	public int GetFastTagMode (int ReaderAddr, ref int mode)	
function	Get Reader carrier offset strategy	
parameter	ReaderAddr	Reader address parameter is for RS485 network establishment with fixed reader, default address is 0XFF.(This parameter is invalid to handheld reader and module)
	mode	If mode is '0', it's favor speed reading mode for single (few) tag fast reading, if mode≠0 , it's favor quantity reading mode for multi tag reading.
return value	Success Return value is 2001	

example	If(GetFastTagMode (addr,ref mode)== 2001)
	MessageBox('success');
	Else
	MessageBox('failed');	

SetFastTagMode

Function instructions	public int SetFastTagMode (int ReaderAddr, int mode)

function	Set Reader carrier offset strategy	
Parameter	ReaderAddr	Reader address parameter is for RS485 network establishment with fixed reader, default address is 0XFF. (This parameter is invalid to handheld reader and module).
	mode	If mode is '0', it's favor speed reading mode for single (few) tag fast reading, if mode≠0 , it's favor quantity reading mode for multi tag reading.
return value	Success Return value is 2001	

example	If(SetFastTagMode (addr, mode)== 2001)
	MessageBox('success');
	Else
	MessageBox('failed');	

SetTestMode

Function instructions	public int SetTestMode (int ReaderAddr, int mode)	
function	Set Reader carrier offset strategy	
parameter	ReaderAddr	Reader address parameter is for RS485 network establishment with fixed reader, default address is 0XFF. (This parameter is invalid to handheld reader and module).
	mode	00' open Amplifier '01' close Amplifier '02' Antenna adjustment, Antenna adjustment is begin when 4 antennas are disconnected.
return value	Success Return value is 2001	

example	If(SetTestMode (addr, mode)== 2001)
	MessageBox('success');
	Else
	MessageBox('failed');	

QueryIDCount

Function instructions	public int QueryIDCount (int ReaderAddr, ref byte tagCount)	
function	Query Tag count in reader buffer	
parameter	ReaderAddr	Reader address parameter is for RS485 network establishment with fixed reader, default address is 0XFF. (This parameter is invalid to handheld reader and module).
	tagCount	queried tag count
return value	Success Return value is 2001	

example	If(QueryIDCount (addr,ref count)== 2001)
	MessageBox('success');
	Else
	MessageBox('failed');	

SetOutPort

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

instructions		
function	set reader output port's High or low level voltage	
parameter	ReaderAddr	Reader address parameter is for RS485 network establishment with fixed reader, default address is 0XFF. (This parameter is invalid to handheld reader and module).
	port_num	Port_num 00-01 are IO port, Port_num 02 is relay output port
	level	output voltage level, '0' is low level voltage, '1'is high level voltage.
return value	Success Return value is 2001	

example	If(SetOutPort(255，0,1)==2001)
	MessageBox('success');
	Else
	MessageBox('failed');	

BuzzerLEDON

Function instructions	public int BuzzerLEDON(int ReaderAddr)	
function	turn on Buzzer or LED	
parameter	ReaderAddr	Reader address parameter is for RS485 network establishment with fixed reader, default address is 0XFF. (This parameter is invalid to handheld reader and module).
return value	Success Return value is 2001	

example	If(BuzzerLEDON (255)==2001)
	MessageBox('success');
	Else
	MessageBox('failed');	

BuzzerLEDOFF

Function instructions	public int BuzzerLEDOFF(int ReaderAddr)	
function	turn off Buzzer or LED	
parameter	ReaderAddr	Reader address parameter is for RS485 network establishment with fixed reader, default address is 0XFF. (This parameter is invalid to handheld reader and module).
return value	Success Return value is 2001	

example	If(BuzzerLEDOFF (255)==2001)
	MessageBox('success');
	Else
	MessageBox('failed');	

GetBuzzerLED

Function instructions	public int GetBuzzerLED(int ReaderAddr,ref byte state)	
function	Get Buzzer or LED	
parameter	ReaderAddr	Reader address parameter is for RS485 network establishment with fixed reader, default address is 0XFF. (This parameter is invalid to handheld reader and module).
	state	the Buzzer and LED status information

return value	Success Return value is 2001

example	If(GetBuzzerLED(255，state)==2001)
	MessageBox('success');
	Else
	MessageBox('failed');

GetTcpParameter

Function instructions	public int GetTcpParameter(int readerAddr,ref string strIP, ref string strMark, ref string strGate, ref int nTcpPort)	
function	Get reader TCP/IP port's IP address, subnet mask, Gateway, TCP/IP port.	
parameter	readerAddr	Reader address parameter is for RS485 network establishment with fixed reader, default address is 0XFF. (This parameter is invalid to handheld reader and module).
	strIP	Reader IP address is an expression of IP address's String format, the IP address has to be the same Network segment with PC (Reader's default IP is 192.168.1.200)
	strMark	Subnet mask address, subnet mask address is an expression of subnet mask address's string format
	strGate	Gateway address, gateway address is an expression of gateway address'string format.
	nTcpPort	TCP/IP port number (range: from1-65535)
return value	Success Return value is 2001	

example	If(GetTcpParameter (255，ref strIP ,ref strMark ,ref strGate,ref nTcpPort)==2001)
	MessageBox('success');
	Else
	MessageBox('failed');	

SetTcpParameter

Function instructions	public int SetTcpParameter(int readerAddr,string strIP, string strMark, string strGate, int nTcpPort)	
function	Set reader TCP/IP port's IP address, subnet mask, Gateway, TCP/IP port.	
parameter	readerAddr	Reader address parameter is for RS485 network establishment with fixed reader, default address is 0XFF. (This parameter is invalid to handheld reader and module).
	strIP	Reader IP address is an expression of IP address's String format, the IP address has to be the same Network segment with PC (Reader's default IP is 192.168.1.200)
	strMark	Subnet mask address, subnet mask address is an expression of subnet mask address's string format
	strGate	Gateway address, gateway address is an expression of gateway address'string format.
	nTcpPort	TCP/IP port number (range: from1-65535)
return value	Success Return value is 2001	

example	If(SetNetSetting(255，strIP , strMark , strGate, nTcpPort)==2001)
	MessageBox('success');
	else	

	MessageBox('failed');

SetMacAddress

Function instructions	public int SetMacAddress(int ReaderAddr, string[] strMacAddr)	
function	Set reader's MAC address	
parameter	readerAddr	Reader address parameter is for RS485 network establishment with fixed reader, default address is 0XFF. (This parameter is invalid to handheld reader and module).
	strMacAddr	The MAC address which need to set
return value	Success Return value is 2001	

example	If(SetMacAddress(255，strMacAddr)==2001)
	MessageBox('success');
	else
	MessageBox('failed');	

GetMacAddress

Function instructions	public int GetMacAddress(int ReaderAddr,ref string[] strMacAddr)	
function	Get reader's MAC address	
parameter	readerAddr	Reader address parameter is for RS485 network establishment with fixed reader, default address is 0XFF. (This parameter is invalid to handheld reader and module).
	strMacAddr	The MAC address which is get.
return value	Success Return value is 2001	

example	If(GetMacAddress(255，ref strMacAddr)==2001)
	MessageBox('success');
	else
	MessageBox('failed');	

SetSerialNo

Function instructions	public int SetSerialNo(int ReaderAddr, string[] strSerialNo)	
function	Set reader's serial number	
parameter	readerAddr	Reader address parameter is for RS485 network establishment with fixed reader, default address is 0XFF. (This parameter is invalid to handheld reader and module).
	strSerialNo	The serial number which is to be set
return value	Success Return value is 2001	

example	If(SetSerialNo (255，strSerialNo)==2001)
	MessageBox('success');
	else
	MessageBox('failed');	

GetSerialNo

Function instructions	public int GetSerialNo(int ReaderAddr, string[] strSerialNo)
function	Get reader's serial number

parameter	readerAddr	Reader address parameter is for RS485 network establishment with fixed reader, default address is 0XFF. (This parameter is invalid to handheld reader and module).
	strSerialNo	the serial number which is get
return value	Success Return value is 2001	

example	If(GetSerialNo (255，ref strSerialNo)==2001)
	MessageBox('success');
	else
	MessageBox('failed');	

1-2. ISO18000-6B Tag Functions

IsoMultiTagIdentify

Function instructions	public int IsoMultiTagIdentify(int readerAddr,ref byte[,] tag_buf, ref byte tag_cnt)	
function	ISO18000-6B multi tag identify	
parameter	readerAddr	Reader address parameter is for RS485 network establishment with fixed reader, default address is 0XFF. (This parameter is invalid to handheld reader and module).
	tag_buf	Identified tag data (includes: tag value, antenna No.)
	tag_cnt	tag number
return value	Success Return value is 2001	

example	if(IsoMultiTagIdentify(255,ref tag_buf,ref tag_cnt)==2001)
	MessagBox('success');
	else
	MessagBox('failed');	

IsoMultiTagRead

Function instructions	public int IsoMultiTagRead(int ReaderAddr, int startAddr, ref byte[,] tag_buf, ref int tag_cnt, ref int getCount)	
function	Read ISO18000-6B tag's user value	
parameter	readerAddr	Reader address parameter is for RS485 network establishment with fixed reader, default address is 0XFF. (This parameter is invalid to handheld reader and module).
	startAddr	
	tag_buf	Identified tag data (includes: tag value, antenna No.)
	tag_cnt	Identified tag number
	getCount	Get tag number
return value	Success Return value is 2001	

example	if(IsoMultiTagRead (255,startAddr,ref tag_buf,ref tag_cnt,ref getCount)==2001)
	MessagBox('success');
	else
	MessagBox('failed');	

IsoReadWithID

Function instructions	public int IsoReadWithID(int ReaderAddr, byte[] byTagID, byte byAddress, ref byte[] byLabelData, ref byte byAntenna)

function	Read specified tag ID's data, read 8bytes on the specified address.	
parameter	ReaderAddr	Reader address
	byTagID	specified tag TID
	byAddress	read starting address
	byLabelData	read tag data
	byAntenna	antenna number
return value	Success Return value is 2001	

example	if(IsoReadWithID(255，byTagID, byAddress, ref byLabelData, ref byAntenna)==2001)
	MessagBox('success');
	else
	MessagBox('failed');
	}	

IsoWriteWithID

Function instructions	public int IsoWriteWithID(int readerAddr,byte[] byTagID, byte byAddress, byte byValue)	
function	write tag data with Tag ID	
parameter	readerAddr	Reader address
	byTagID	specified tag TID
	byAddress	the starting address of encoding tag
	byValue	encoding value
return value	Success Return value is 2001	

example	if(IsoWriteWithID(255,byTagID, byAddress, byValue)==2001)
	MessageBox('success');
	else
	MessageBox('failed');	

IsoLockWithID

Function instructions	public int IsoLockWithID(int ReaderAddr, byte[] byTagID, byte byAddress)	
function	Lock tag with tag ID	
parameter	ReaderAddr	Reader address
	byTagID	the specified tag TID
	byAddress	the starting address of encoding value
return value	Success Return value is 2001	

example	if(IsoWriteWithID(255，byTagID, byAddress)==2001)
	MessageBox('success');
	else
	MessageBox('failed');	

IsoRead

Function instructions	public int IsoRead(int readerAddr,byte addr, ref byte[] value)
function	Read ISO18000-6B tag

parameter	readerAddr	Reader address
	addr	Starting read address
	value	tag value
return value	Success Return value is 2001	

example	if(IsoRead (255,addr, value)==2001)
	MessageBox('success');
	else
	MessageBox('failed');	

IsoWrite

Function instructions	public int IsoWrite(int readerAddr,byte addr, byte value)	
function	Write ISO18000-6B tag	
parameter	readerAddr	Reader address
	addr	the starting address of encoding
	value	encoding value
return value	Success Return value is 2001	

example	if(IsoWrite (255，addr, value)==2001)
	MessageBox('success');
	else
	MessageBox('failed');	

IsoLock

Function instructions	public int IsoLock((int readerAddr,byte addr)	
function	to lock ISO18000-6B tag's specific address, once locked, the address cannot be unlocked	
parameter	readerAddr	Reader address
	addr	the locking address
return value	Success Return value is 2001	

example	if(IsoLock (255，addr))==2001)
	MessageBox('success');
	else
	MessageBox('failed');	

IsoQueryLock

Function instructions	public int IsoQueryLock(int ReaderAddr, byte addr, ref byte lstate)	
function	Query ISO18000-6B tag lock	
parameter	readerAddr	Reader address
	addr	The locking address
	lstate	lock status
return value	Success Return value is 2001	

example	if(IsoQueryLock (255，addr，ref lstate))==2001)
	MessageBox('success');
	else
	MessageBox('failed');

IsoQueryLockWithUID

Function instructions	public int IsoQueryLock(int ReaderAddr, byte[] byTagID,byte addr, ref byte lstate)	
function	Query ISO tag lock with UID	
parameter	readerAddr	Reader address
	byTagID	Tag UID
	addr	the locking address
	lstate	Lock status
return value	Success Return value is 2001	

example	if(IsoQueryLock (255，id,addr,ref lstate))==2001)
	MessageBox('success');
	else
	MessageBox('failed');	

1-3. EPC GEN2 Tag Functions

EpcMultiTagIdentify

Function instructions	public int EpcMultiTagIdentify(int readerAddr,ref byte[,] tag_buf, ref byte tag_cnt, ref byte tag_flag)	
function	EPC Gen2 Multi tag Identify	
parameter	readerAddr	Reader address
	tag_buf	Identified tag data (includes tag value, antenna number)
	tag_cnt	tag number
	tag_flag	read status: '0' read tag success, '1' read tag failed
return value	Success Return value is 2001	

example	if(Gen2MultiTagIdentify(255，ref tag_buf, ref tag_cnt, ref tag_flag)==2001) MessagBox('success');
	else
	MessagBox('failed');	

EpcRead

Function instructions	public int EpcRead(int readerAddr,e membank, byte wordptr, byte wordcnt, ref byte[] value)	
function	Continuous read specified word address tag.	
parameter	readerAddr	Reader address
	membank	Kill password and access password membank---'0'; EPC Membank----'1'; USER Membank---'3'
	wordptr	start address
	wordcnt	Word count, unit: word.
	value	tag value
return value	Success Return value is 2001	

example	if(EpcRead (255, membank, wordptr, wordcnt, ref value)==2001)
	MessageBox('success');
	else
	MessageBox('failed');

EpcWrite

Function instructions	public int EpcWrite(int ReaderAddr, byte membank, byte wordptr, ushort value)	
function	continuously encoding a specific value to a specific address, encode one word only at a time.	
parameter	ReaderAddr	Reader address
	membank	Kill password and access password membank---'0'; EPC Membank----'1'; USER Membank---'3'
	wordptr	the starting address
	value	encoding tag value
return value	Success Return value is 2001	

example	if(EpcWrite (255，membank, wordptr, value)==2001)
	MessageBox('success');
	else
	MessageBox('failed');	

Gen2MultiTagWrite

Function instructions	public int Gen2MultiTagWrite(int ReaderAddr, int membank, int wordaddr, int wordLen, string strValue, ref int writeCount)	
function	EPC Gen2 Multi tag write	
parameter	ReaderAddr	Reader address
	membank	Kill password and access password membank---'0'; EPC Membank----'1'; USER Membank---'3'
	wordaddr	The starting address
	wordLen	encoding length
	strValue	string value
	writeCount	Succeeded encoding tag number
return value	Success Return value is 2001	

example	if(EpcWrite (255，membank, wordaddr，wordLen，strValue，ref writecount)==2001)
	MessageBox('success');
	else
	MessageBox('failed');	

Gen2MultiTagRead

Function instructions	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	ReaderAddr	Reader address
	MembankMask	Membank selection, display in mask, from 1 to 4 correspond with Reserve Membank, EPC Membank, TID Membank, USER Membank.
	ResWordPtr	read reserve Membank word address
	ResWordCnt	read reserve Membank word count
	EpcWordPtr	Read EPC Membank word address
	EpcWordCnt	EPC word Count
	TidWordPtr	Read TID Word address
	TidWordCnt	TID Membank word count
	UserWordPtr	Read USER Membank word address
	UserWordCnt	USER Membank word count
	ReadCnt	Read word count set
return value	Success Return value is 2001	

example	if(EpcWrite (255 ,MembankMask, ResWordPtr, ResWordCnt, EpcWordPtr, EpcWordCnt, TidWordPtr, TidWordCnt, UserWordPtr, UserWordCnt, ref ReadCnt)==2001)
	MessageBox('success');
	else
	MessageBox('failed');	

Gen2SecLock

Function instructions	public int Gen2SecLock(int ReaderAddr, uint AccPassWord, byte Membank, byte Level)	
function	Lock EPC Gen2 tag. Only can encode epc tag with correct password.	
parameter	ReaderAddr	Reader address
	AccPassWord	Access Password Membank value
	Membank	To be locked membank, '0'--- reverse membank, '1'---EPC Membank; '2'--- TID Membank; '3'--- User Membank.
	Level	Lock level ('0'--- unlock; '1'---unlock forever; '2'---secure lock; '3'--- lock forever.)
return value	Success Return value is 2001	

example	if(Gen2SecLock(255, AccPassWord, Membank, Level)==2001)
	MessageBox('success');
	else
	MessageBox('failed');	

EpcLockTag

Function instructions	public int EpcLockTag(int readerAddr,byte MemBank)	
function	Lock EPC Gen2 tag. Only can encode epc tag with correct password	
parameter	readerAddr	Reader address
	Membank	byMemBank: write value to the locked membank, '0'---Reserve Membank; '1'---EPC membank; '2'---TID

		membank; ‘3’---USER membank.
return value	Success Return value is 2001	

example	if(EpcLockTag (255，MemBank)==2001)
	MessageBox(‘success’);
	else
	MessageBox(‘failed’);	

EpcInitEpc

Function instructions	public int EpcInitEpc(int readerAddr,byte bit_cnt)	
function	Initialize the length of EPC Membank, the default epc length is 96bits (6words), and the default value is all ‘0’	
parameter	readerAddr	Reader address
	bit_cnt	initialize the data length
return value	Success Return value is 2001	

example	if(EpcInitEpc (255，bit_cnt)==2001)
	MessageBox(‘success’);
	else
	MessageBox(‘failed’);	

Gen2SecWrite

Function instructions	public int Gen2SecWrite(int ReaderAddr, uint AccPassWord, byte Membank, byte WordAddr, ushort Value)	
function	EPC Gen2 tag secure write	
parameter	readerAddr	Reader address
	AccPassWord	Access password value
	Membank	byMemBank: write value to the locked membank, ‘0’---Reserve Membank; ‘1’---EPC membank; ‘2’---TID membank; ‘3’---USER membank.
	WordAddr	the starting word address of secure write
	Value	secure write value
return value	Success Return value is 2001	

example	if(Gen2SecWrite(255, AccPassWord, Membank, WordAddr, Value)==2001)
	MessageBox(‘success’);
	else
	MessageBox(‘failed’);	

Gen2SecRead

Function instructions	public int Gen2SecRead(int ReaderAddr, uint AccPassWord, byte Membank, byte WordAddr, byte WordCnt, ref byte[] value)	
function	EPC Gen2 tag secure read	
parameter	readerAddr	Reader address
	AccPassWord	Access password value
	Membank	byMemBank: write value to the locked membank,

		'0'---Reserve Membank; '1'---EPC membank; '2'---TID membank; '3'---USER membank.
	WordAddr	The starting word address of secure read
	WordCnt	The word count of secure read
	value	secure read value
return value	Success Return value is 2001	

example	if(Gen2SecRead(255, AccPassWord, Membank, WordAddr, WordCnt, ref value)==2001)
	MessageBox('success');
	else
	MessageBox('failed');	

Gen2SelectConfig

Function instructions	public int Gen2SelectConfig(int ReaderAddr, int Action, int Membank, int wordAddr, int wordCnt, string[] words)	
function	EPC Gen2 tag reading selection feature configuration	
parameter	readerAddr	Reader address
	Action	'0'---matched selection; '1'---mismatched selection
	Membank	byMemBank: write value to the locked membank, '0'---Reserve Membank; '1'---EPC membank; '2'---TID membank; '3'---USER membank.
	WordAddr	the starting address of selected word
	WordCnt	the word count of selection
	words	The word value of selection
return value	Success Return value is 2001	

example	if(Gen2SelectConfig(255, action, Membank, WordAddr, WordCnt, words)==2001)
	MessageBox('success');
	else
	MessageBox('failed');	

Gen2SetAccPwd

Function instructions	public int Gen2SetAccPwd(uint AccPassWord)	
function	Set EPC Gen2 tag access password	
parameter	AccPassWord	Access password value
return value	Success Return value is 2001	

example	if(Gen2SecRead(AccPassWord)==2001)
	MessageBox('success');
	else
	MessageBox('failed');	

Gen2KillTag

Function instructions	public int Gen2KillTag(int ReaderAddr, uint AccPassWord)

function	Kill EPC Gen2 tag.	
parameter	ReaderAddr	Reader address
	AccPassWord	kill epc gen2 tag's value
return value	Success Return value is 2001	

example	if(Gen2KillTag (255，AccPassWord)==2001)
	MessageBox('success');
	else
	MessageBox('failed');	

1-4. ISO18000-6B tag data management function

ClearIdBuf

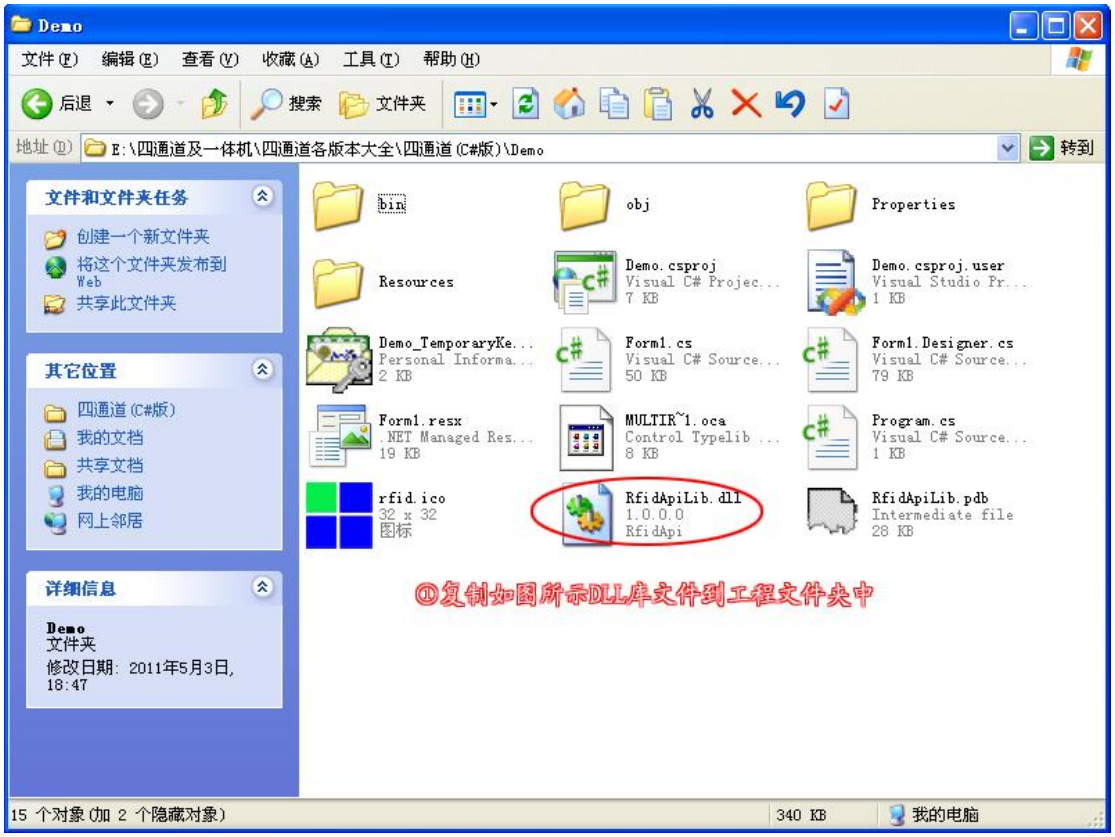
Function instructions	public int ClearIdBuf(int readerAddr)
function	Clear buffer data, used before the command of multi tag identify.
parameter	readerAddr: Reader address
return value	Success Return value is 2001

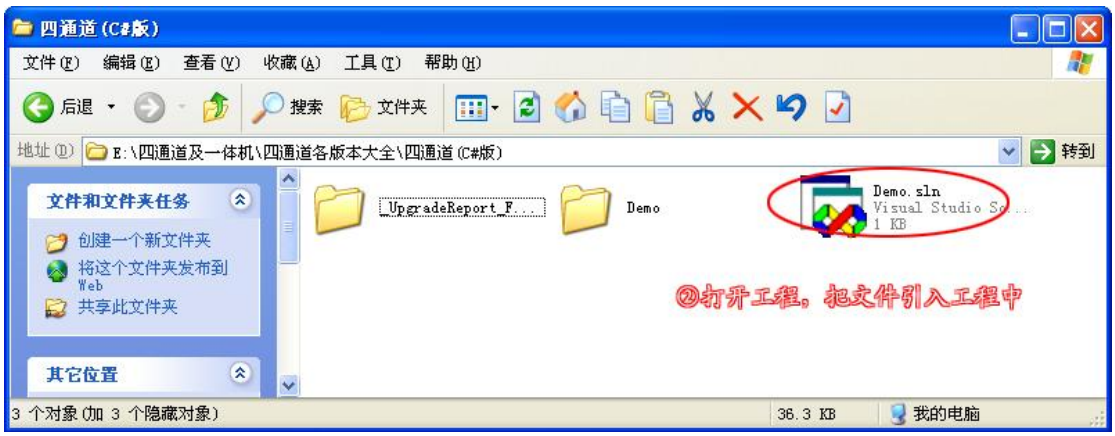
example	if(ClearIdBuf (255)==2001)
	MessageBox('success');
	else
	MessageBox('failed');

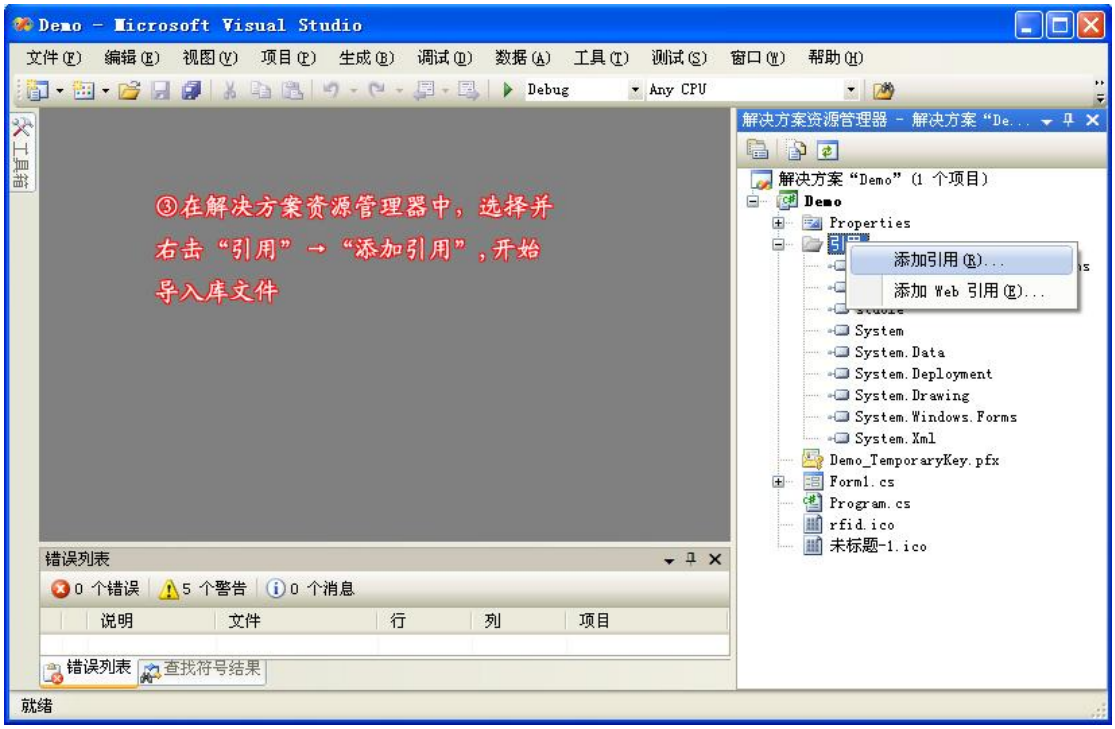
2. Samples of Demo software source code

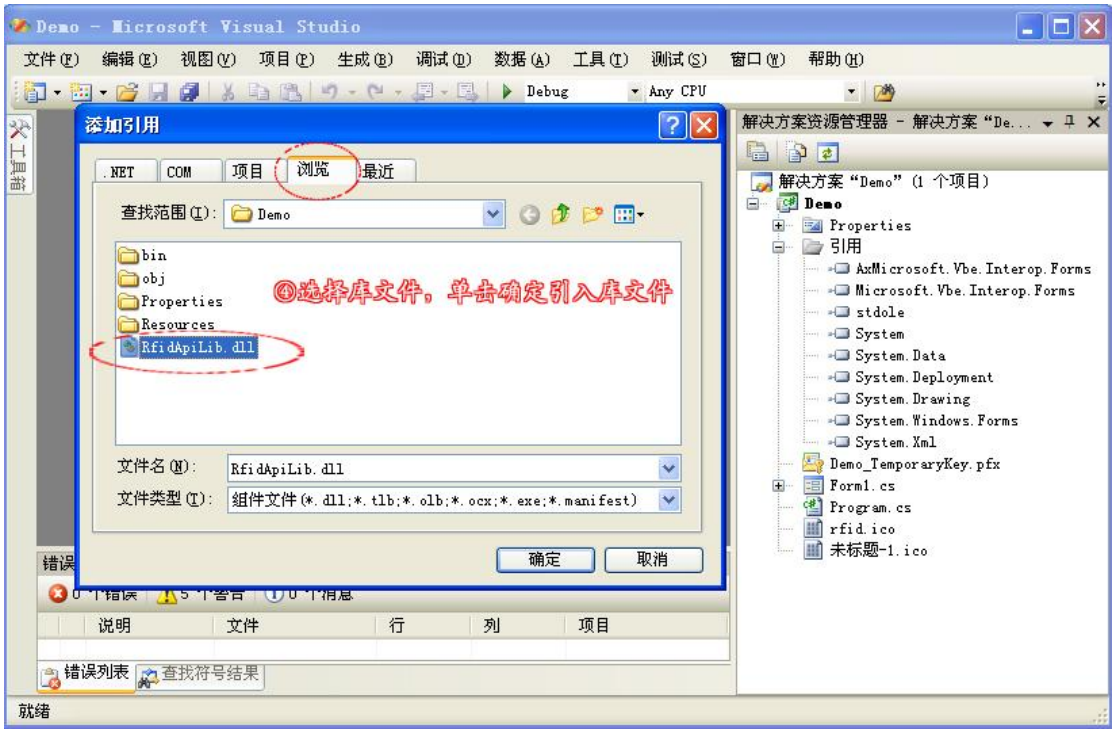
2-1. Establish project

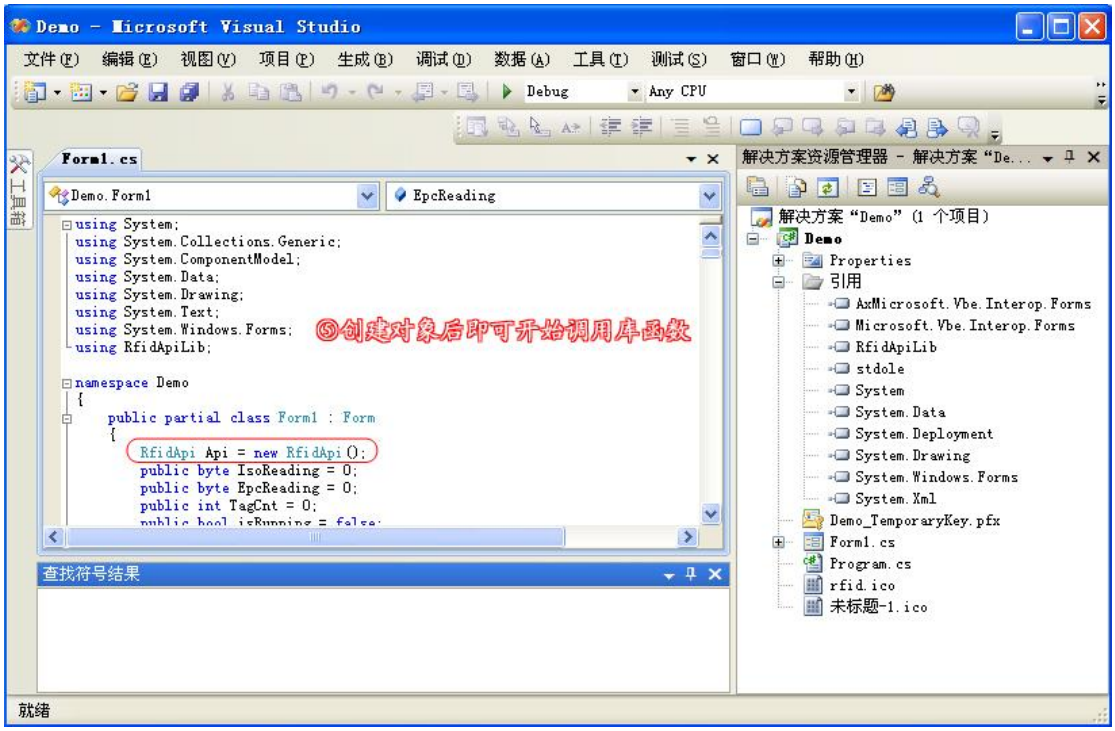
2-2. Input RfidApilib.dll library to the established project, take below image for reference






Demo - Microsoft Visual Studio
文件(F) 编辑(E) 视图(V) 项目(P) 生成(B) 调试(D) 数据(A) 工具(T) 测试(S) 窗口(W) 帮助(H)
Debug
Any CPU
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
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
添加引用(R)...
添加 Web 引用(E)...
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

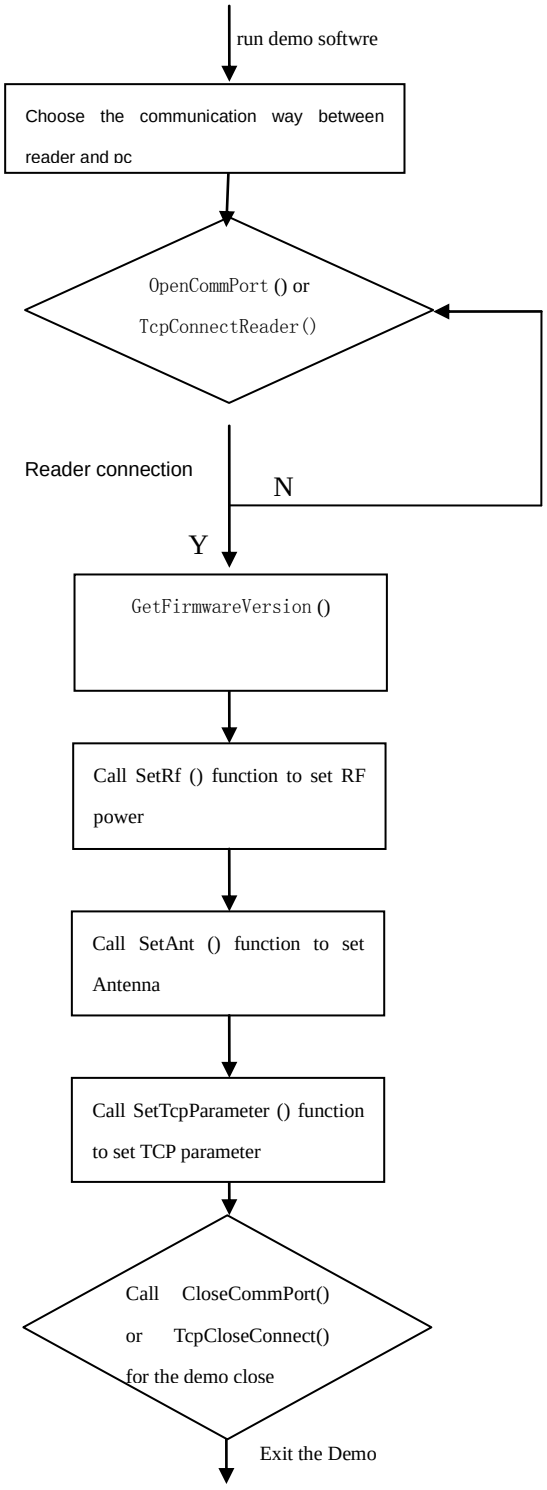

Demo - Microsoft Visual Studio
文件(F) 编辑(E) 视图(V) 项目(P) 生成(B) 调试(D) 数据(A) 工具(T) 测试(S) 窗口(W) 帮助(H)
Debug
Any CPU
添加引用
.NET
COM
项目
浏览
最近
查找范围(I):
Demo
bin
obj
Properties
Resources
RfidApiLib.dll
④选择库文件，单击确定引入库文件
文件名(N):
RfidApiLib.dll
文件类型(T):
组件文件(*.dll;*.tlb;*.olb;*.ocx;*.exe;*.manifest)
确定
取消
解决方案资源管理器 - 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



Call Dll library functions after Inputing **RfidApilib.dll** library to the established project and creating the object.

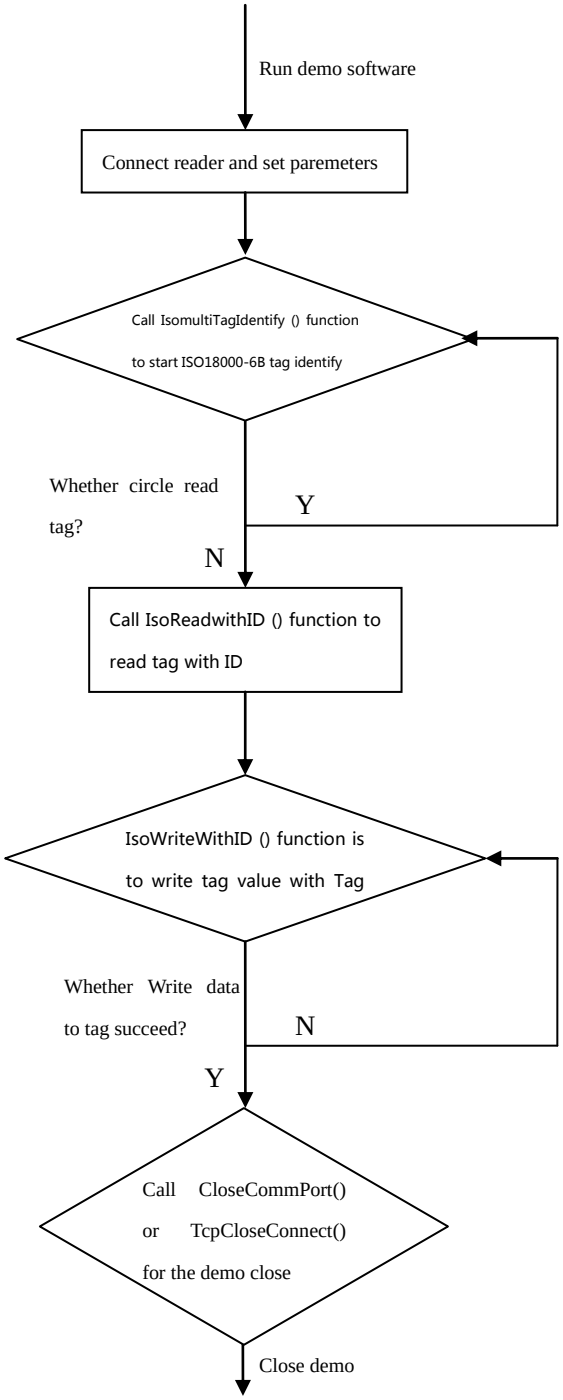
2-3. Call for functions

Take below flow chart for the reference of reader configurat ion and management

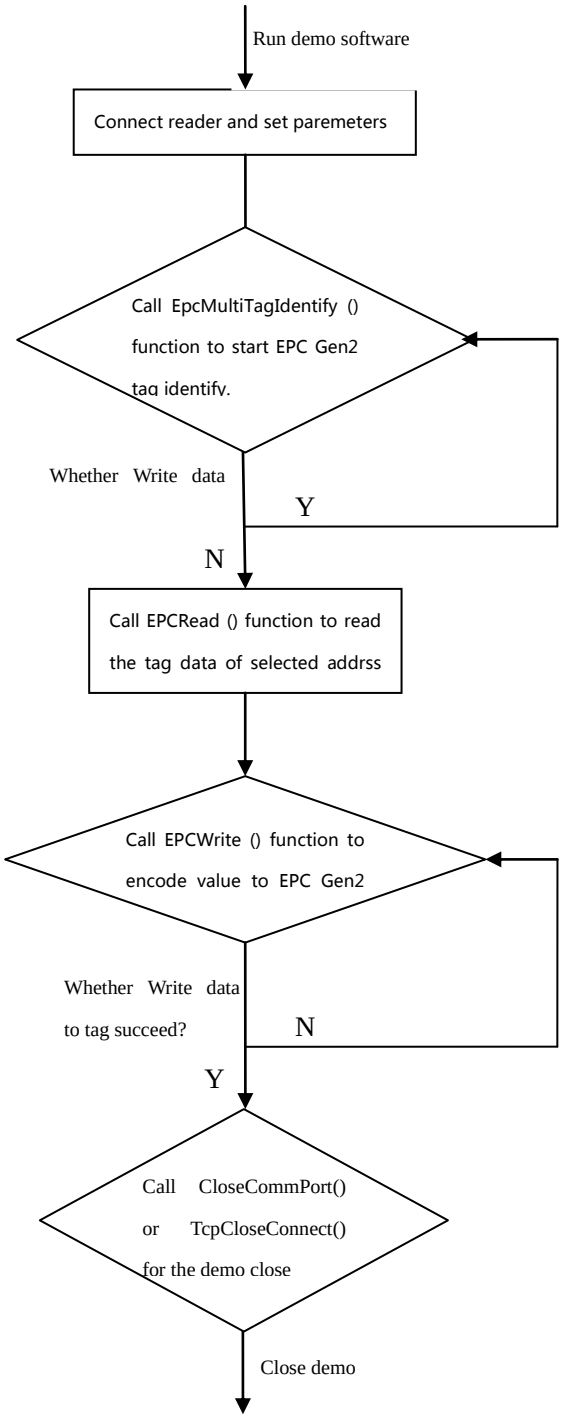


Call CloseCommport () or TcpCloseConnect () fuction to close the connection

ISO18000-6B tag working flow chart



EPC Gen2 tag working flow chart



1. Call OpenCommPort() function for reader connection

```
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 () function to set RF 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 () function to 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 () function to 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 () function to start ISO18000-6B 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 () function to read tag with ID

```
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 () function is to write tag value with Tag ID.

```
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 () function to start EPC Gen2 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 () function to read the tag data of selected addrss and selected address 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 () function to encode value to 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);

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 () function to lock EPC Gen2 tag for EPC Tag encoding with access 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 () function for EPC Gen2 tag secure write.

```
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 () function for EPC Gen2 tag destroy.

```
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);
}
```