

RFID read and write device PC Development Guide C#

Contents

1. Foreword

- 1.1. Overview
- 1.2. Applicable equipment
- 1.3. Copyright description
- 1.4. Basic process of reading and writing

2. Quickly get started

3. Connection description

- 3.1.RS232 connection
- 3.2.RS485 connection
- 3.3.TCP client connection
- 3.4.USB-HID connection
- 3.5.TCP server monitoring
- 3.6.Close the connection

4. Event description

- 4.1.ISO18000-6C Label report event
Report
- 4.2.ISO18000-6C label report to the end of the incident
- 4.3.ISO18000-6b Label report event
Report
- 4.4.ISO18000-6b Tags reported the end
- 4.5.GB national standard label report incident
Report
- 4.6.GB national standard label report to the end of the event
- 4.7.GJB National Army standard label report incident
Report the object
- 4.8.GJB National Army standard labels reported the end of the incident
- 4.9.GPITriggering starting events
Report the object
- 4.10. GP | Trigger the end event
Report the object
- 4.11.TCP connect to disconnect
- 4.12.usb-HID disconnect
- 4.13.TCP connection event
Report the object

5. Message configuration and query instructions

- 5.1. Send synchronous message
 - Code example 1
 - Code example 2
 - Code is not examples 3

6. News

- 6.1. Configuration and management of reader
 - 6.1.1. Restart the reader

- 6.1.2. Configuration and query serial parameters
- 6.1.3. Configure GPO status parameters
- 6.1.4. Query GPI status parameters
- 6.1.5. Configuration and query GPI trigger parameters
- 6.1.6. Query baseband software version
- 6.1.7. Query read and write information information
- 6.1.8. Query reader RFID ability
- 6.1.9. Get the label cache data
- 6.1.10. Clear the label cache data
- 6.1.11. Beeurist control

6.2. RFID configuration and operation

- 6.2.1. Stop instruction
- 6.2.2. Configuration and query reader power
- 6.2.3. Configuration and query reader operating frequency band
- 6.2.4. Configuration and query EPC baseband parameters
- 6.2.5. Configuration and query GB baseband parameters
- 6.2.6. Configuration and query label upload parameters
- 6.2.7. Read EPC tag
- 6.2.8. Write EPC tag
- 6.2.9. Lock EPC tag
- 6.2.10. Activated EPC label
- 6.2.11. Read 6B label
- 6.2.12. Write 6B label
- 6.2.13. 6B label lock
- 6.2.14. 6B label lock inquiries
- 6.2.15. Reading GB tag
- 6.2.16. Write GB tag
- 6.2.17. Lock GB tag
- 6.2.18. Activated GB label.
- 6.2.19. Read the GJB label
- 6.2.20. Write GJB tag
- 6.2.21. Lock GJB label
- 6.2.22. Activated GJB label.

7. Parameter description

- 7.1.1. 6C/GB/GJB label selection parameters
- 7.1.2. 6C label read TD parameters
- 7.1.3. 6C label reading user data area parameters
- 7.1.4. 6B Tag Reading User Data Area Parameters
- 7.1.5. GB label reading user data area parameters
- 7.1.6. GJB label read user data area parameters

8. Appendix 1

The frequency band list supported by the reader

9. Appendix 2

1.Foreword

1.1.Overview

In order to facilitate secondary development, we provide a function library that can run on the .NET platform. The library is written and encapsulated into a standard DLL library "Greamrapi.dll", which supports .NETFRAMEWORK2.0 and above versions.

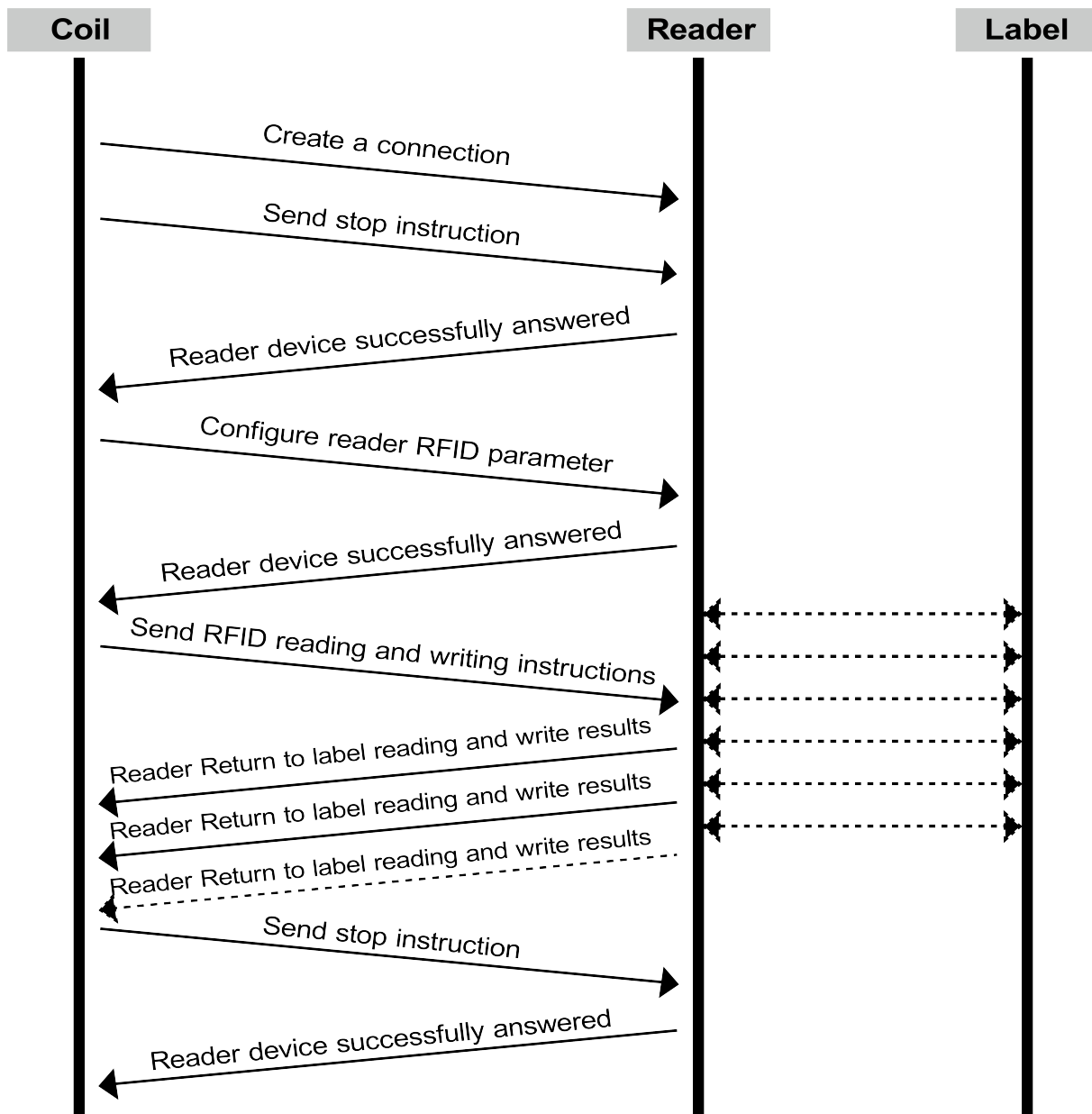
1.2.Applicable equipment

Functional module	Applicable device type
Reader configuration and management	R8008、R8004
RFID configuration and operation	UHF device full model

1.3.Copyright explanation

All the contents of the document, including text and pictures are original. The company reserves the right to investigate its legal responsibility for unauthorized use of unauthorized use in commercial users. Without authorization, users shall not add, modify, delete the content of this document without authorization, and shall not spread them in the form of networks and discs. If you violate, the consequences are at your own risk.

1.4. Basic process of reading and writing



2. Get started quickly

```

using GDotnet.Reader.Api.DAL;
using GDotnet.Reader.Api.Protocol.Gx;
using System;
using System.Collections.Generic;
using System.Text;

// =====
// Copyright (C) 2019 SZGxwl Inc. All rights reserved.
//

```

```
//
//
//
// =====
namespace GDotnet.Reader.Api
{
    static class Example
    {
        static void Main()
        {
            GClient clientConn = new GClient();
            eConnectionAttemptEventStatusType status;
            // clientConn.OpenTcp("192.168.1.168:8160", 3000, out status)
            if (clientConn.OpenSerial("COM16:115200", 3000, out status))
            {
                // Subscribe to the incident on the label
                clientConn.OnEncapedTagEpcLog += new delegateEncapedTagEpcLog (OnEncapedTagEpcLog);
                clientConn.OnEncapedTagEpcOver += new
delegateEncapedTagEpcOver (OnEncapedTagEpcOver);

                // Stop instruction, idle state
                MsgBaseStop msgBaseStop = new MsgBaseStop();
                clientConn.SendSynMsg(msgBaseStop);
                if (0 == msgBaseStop.RtCode)
                {
                    Console.WriteLine("Stop successful.");
                }
                else { Console.WriteLine("Stop error."); }

                Console.WriteLine("Enter any character to start reading the tag.");
                Console.ReadKey();

                // 4 antenna read cards, read EPC data areas and TID data areas
                MsgBaseInventoryEpc msgBaseInventoryEpc = new MsgBaseInventoryEpc();
                msgBaseInventoryEpc.AntennaEnable = (ushort)(eAntennaNo._1 | eAntennaNo._2 |
eAntennaNo._3 | eAntennaNo._4);
                msgBaseInventoryEpc.InventoryMode = (byte)eInventoryMode.Inventory;
                msgBaseInventoryEpc.ReadTid = new ParamEpcReadTid(); // tid parameter
                msgBaseInventoryEpc.ReadTid.Mode = (byte)eParamTidMode.Auto;
                msgBaseInventoryEpc.ReadTid.Len = 6;
                clientConn.SendSynMsg(msgBaseInventoryEpc);
                if (0 == msgBaseInventoryEpc.RtCode)
                {
                    Console.WriteLine("Inventory epc successful.");
                }
            }
        }
    }
}
```

```
    }
    else { Console.WriteLine("Inventory epc error."); }
    Console.ReadKey();

    // Stop reading card, idle state
    clientConn.SendSynMsg(msgBaseStop);
    if (0 == msgBaseStop.RtCode)
    {
        Console.WriteLine("Stop successful.");
    }
    else { Console.WriteLine("Stop error."); }
}
else
{
    Console.WriteLine("Connect failure.");
}
Console.ReadKey();
}

#region API 事件

public static void OnEncapedTagEpcLog(EncapedLogBaseEpcInfo msg)
{
    // If there is obstruction inside the callback, it will affect the normal use of
    // the API
    // There are many label callbacks, please save the label data first and then do
    // business processing
    if (null != msg)
    {
        Console.WriteLine(msg.logBaseEpcInfo.ToString());
    }
}

public static void OnEncapedTagEpcOver(EncapedLogBaseEpcOver msg)
{
    if (null != msg)
    {
        Console.WriteLine("Epc log over.");
    }
}

#endregion
}
```

3.Connection description

3.1.RS232connect

Naming space	GDotnet.Reader.Api.DAL
Kind	GClient
Method	<code>public bool OpenSerial(String readerName,int timeout , out eConnectionAttemptEventStatusTypestatus)</code>
Illustrate	<code>readerName</code> :Connect string, like" COM1:115200" (" <code>String slogan: Potter rate</code> ") <code>timeout</code> :Connect confirmation timeout (millisecond), such as "1000" <code>status</code> :Connection state enumeration

3.2.RS485 grout

Naming space	GDotnet.Reader.Api.DAL
Kind	GClient
Method	<code>public bool OpenSerial485(String readerName int timeout , out eConnectionAttemptEventStatusTypestatus)</code>
Illustrate	<code>readerName</code> :Connect string, like" COM1:115200:1" (" <code>String slogan: Potter rate:485 address</code> ") <code>timeout</code> :Connect confirmation timeout (millisecond),such as" 1000" <code>status</code> :Connection state enumeration In the half -duplex mode, occasionally failure of communication is a normal phenomenon, please try it out after failure.

3.3.TCPClient connection

Naming space	GDotnet.Reader.Api.DAL
Kind	GClient

Method	<code>public bool OpenTcp(String readerName, int timeout, out eConnectionAttemptEventStatusType status)</code>
Illustrate	<p><code>readerName</code>: Connect string (such as "192.168.1.168:8160"). Reader default IP "192.168.1.168", Default port "8160"</p> <p><code>timeout</code>: Connect confirmation timeout (millisecond), such as "1000"</p> <p><code>status</code>: Connection state enumeration</p>

3.4.USB-HIDconnect

Naming space	GDotnet.Reader.Api.DAL
Kind	GClient
Method	<code>public bool OpenUsbHid(String readerName, IntPtr handle, int timeout, out eConnectionAttemptEventStatusType status)</code>
Illustrate	<p><code>readerName</code>: Connect string (pass <code>GetUsbHidList()</code> Methods to obtain a USB list).</p> <p><code>handle</code>: Handle, Can <code>IntPtr.Zero</code>.</p> <p><code>timeout</code>: Connect confirmation timeout (millisecond), such as "1000"</p> <p><code>status</code>: Connection state enumeration</p>

3.5.TCPServer monitoring

Naming space	GDotnet.Reader.Api.DAL
Kind	GServer
Method	<code>public bool Open(int port)</code>
Illustrate	<p><code>port</code>: Local interface listening to the upper machine.</p> <p>Using this method to monitor, you need to configure the UHF read-write device to "client mode".</p> <p>For the "client mode" configuration method, please refer to 《RFID Demonstration S software Operation Manual》.</p>

3.6.Close the connection

Naming space	GDotnet.Reader.Api.DAL
Kind	GClient

Method	<code>public void Close()</code>
Illustrate	<p>Close and release link resources.</p> <p>Note: For linked objects that have failed, you need to actively call this method to release resources.</p>

4.Event description

4.1.ISO18000-6CLabel report

Naming space	GDotnet.Reader.Api.DAL
kind	GClient
Event	<code>public delegate EncapedTagEpcLogOnEncapedTagEpcLog;</code> <code>public delegate TagEpcLogOnTagEpcLog;</code>
Illustrate	<p><code>public delegate void delegateEncapedTagEpcLog(EncapedLogBaseEpcInfo msg);</code> <code>msg.reader</code>: Reader link string. <code>msg.serialNo</code>: The reader serial number can be automatically obtained when the reader is a client mode. The non -client mode needs to send the "Query Reader Information" message to query the "Query Reader Information"). <code>msg.logBaseEpcInfo</code>: Together.</p> <p><code>public delegate void delegateTagEpcLog(LogBaseEpcInfo msg);</code> 6C Label actively reporting incident: When the reader is a card reading state, the label information will be reported through this incident. For example, see "Quickness". <code>LogBaseEpcInfo</code>: See the "report object" for details</p>

Report

Naming space	GDotnet.Reader.Api.Protocol.Gx
Object	LogBaseEpcInfo

Property	<p> Epc: Hexades EPC Strings BEpc: EPCByte array Pc: PCvalue AntId: Antenna number Rssi: Signal strength Result: Tag reading results, 0 is successful reading, non-0 is failure 1, No response 2, CRCmistake 3, Data area is locked 4, Data area overflow 5, Access password error 6, Other label errors 7, Other reader errors Tid: Hexadent TID string BTid: TIDByte array Userdata: Hexades UserData string BUser: UserdataByte array Reserved: Hexadent retaining area string BRes: Reserved area byte group ChildAntId: Expanded antenna antenna number </p>
Illustrate	6CTags actively report parameters.

4.2.ISO18000-6C Tag report ending incident

Naming space	GDotnet.Reader.Api.Protocol.Gx
kind	GClient
Property	public delegate TagEpcOver OnTagEpcOver;
Illustrate	<p> public delegate void delegateTagEpcOver(LogBaseEpcOvermsg); 6C The label actively reports the end parameters to ensure that asynchronous news is synchronized. </p>

4.3.ISO18000-6B Label report

Naming space	GDotnet.Reader.Api.DAL
kind	GClient
Event	public delegate Tag6bLogOnTag6bLog;
Illustrate	public delegate void delegateTag6bLog(LogBase6bInfo msg); 6BLabel actively reporting incident: When the reader is a card reading state, the label information will be reported through this incident. For example, see "Quickness". LogBase6bInfo : See the "report object" for details

Report

Naming space	GDotnet.Reader.Api.Protocol.Gx
Object	LogBase6bInfo
Property	AntId : Antenna number Rssi : Signal strength Result : Tag reading results, 0 is successful reading, non -0 is failure 1, No response 2, CRCmistake 3, Other reader errors Tid : Hexadent TID string BTid : TIDByte array Userdata : Hexades UserData string BUser : UserdataByte array
Illustrate	6CTags actively report parameters.

4.4.ISO18000-6B Tag report ending incident

Naming space	GDotnet.Reader.Api.Protocol.Gx
--------------	--------------------------------

Kind	GClient
Property	<code>public</code> delegateTag6bOver OnTag6bOver;
Illustrate	<code>public delegate void</code> delegateTag6bOver(LogBase6bOvemsg); 6BTags actively report parameters.

4.5.GB Report event on the national standard label

Naming space	GDotnet.Reader.Api.DAL
Kind	GClient
Event	<code>public</code> delegateTagGb LogOnTag GbLog;
Illustrate	<code>public delegate void</code> delegateTagGbLog(LogBaseGbInfomsg); GBLabel actively reporting incident: When the reader is a card reading state, the label information will be reported through this incident. For example, see "Quickness". LogBaseGbInfo : See the "report object" for details

Report

Naming space	GDotnet.Reader.Api.Protocol.Gx
Object	LogBaseGbInfo

Property	<p>Epc: Hexades EPC Strings(Label coding area)</p> <p>BEpc: EPCByte array(Label coding area)</p> <p>Pc: PCvalue</p> <p>AntId: Antenna number</p> <p>Rssi: Signal strength</p> <p>Result: Tag reading results, 0 is successful reading, non -0 is failure1, No response</p> <p>2, CRCmistake</p> <p>3, Data area is locked</p> <p>4, Data area overflow</p> <p>5, Read password error</p> <p>6, Insufficient permissions</p> <p>7, Identify failure</p> <p>8, Other label errors</p> <p>9, Other reader errors</p> <p>Tid: Hexadent TID string(Label information area)</p> <p>BTid: TIDByte array(Label information area)</p> <p>Userdata: Hexades UserData string</p> <p>BUser: UserdataByte array</p>
Illustrate	6CTags actively report parameters.

4.6.GB National standard label report ended incident

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	GClient
Property	public delegate TagGbOver OnTagGbOver;
Illustrate	<p>public delegate void delegateTagGbOver(LogBaseGbOver msg);</p> <p>GBTags actively report parameters.</p>

4.7.GJB Reported incidents of national military standard labelling

Naming space	GDotnet.Reader.Api.DAL
--------------	------------------------

kind	GClient
Event	<code>public delegate TagGJbLog OnTagGJbLog;</code>
Illustrate	<code>public delegate void delegateTagGJbLog(LogBaseGJbInfo msg);</code> GJLabel actively reporting incident: When the reader is a card reading state, the label information will be reported through this incident. For example, see "Quickness". LogBaseGJbInfo : See the "report object" for details

Report

Naming space	GDotnet.Reader.Api.Protocol.Gx
Object	LogBaseGJbInfo
Property	<p> Epc: Hexades EPC Strings(Label coding area) BEpc: EPCByte array(Label coding area) Pc: PCvalue AntId: Antenna number Rssi: Signal strength Result: Tag reading results, 0 is successful reading, non -0 is failure 1, No response 2, CRCincorrect 3, Data area is locked 4, Data area overflow 5, Error reading password 6, Insufficient authority 7, Failure to identify 8, Other label errors 9, Other reader errors Tid: Hexadent TID string(Label Information Area) BTid: TIDByte array(Label Information Area) Userdata: Hexades UserData string BUser: UserdataByte array </p>
Illustrate	6CTags actively report parameters.

4.8.GJB National Military Labelling Reporting Closing Event

Naming space	GDotnet.Reader.Api.Protocol.Gx
kind	GClient
Property	public delegate TagGb Over OnTagGJbOver;
Illustrate	<pre>public delegate void delegateTagGJbOver(LogBase GJbOvermsg);</pre> <p>GBTags actively report parameters.</p>

4.9.TLTagged reported incidents

Naming space	GDotnet.Reader.Api.DAL
kind	GClient
Event	<pre>public delegate EncapedTagTILog OnEncapedTagTILog;</pre> <pre>public delegate TagTILog OnTagTILog;</pre>
Illustrate	<pre>public delegate void delegateEncapedTagTILog(EncapedLogBaseTIIInfo msg);</pre> <p>msg.reader: Read-write link string. msg.serialNo: Serial number of the reader/writer (can be obtained automatically when the reader/writer is in client mode, non-client mode requires the caller to actively send a query 'query reader/writer information' message). msg.logBaseTIIInfo: <i>ibid</i>. <pre>public delegate void delegateTagTILog(LogBaseTIIInfo msg);</pre> <p>6CLabel actively reporting incident: When the reader is a card reading state, the label information will be reported through this incident. For example, see "Quickness". LogBaseTIIInfo: See the "report object" for details</p> </p>

Report

Naming space	GDotnet.Reader.Api.Protocol.Gx
Object	LogBaseTIIInfo

Property	<p> Epc: Hexades EPC Strings BEpc: EPCByte array Pc: PCvalue AntId: Antenna number Rssi: Signal strength Result: Tag reading results, 0 is successful reading, non -0 is failure 1, No response 2, CRCincorrect 3, Data area is locked 4, Data area overflow 5, Access password error 6, Other label errors 7, Other reader errors ViewData: Railway data conversion display, hexadecimal TID string, 6bit data convert ed to 8bit ascii code data via EPC data; Industry data parses itself; </p>
Illustrate	TLTags actively report parameters.

4.10.TL Tagged on to report the end of the incident

Naming space	GDotnet.Reader.Api.Protocol.Gx
kind	GClient
Property	public delegateTagTIOver OnTagTIOver;
Illustrate	<pre>public delegate void delegateTagTIOver(LogBaseTIOver msg);</pre> <p>TLTags actively report parameters.</p>

4.11.GPItrigger a start event

Naming space	GDotnet.Reader.Api.DAL
kind	GClient
Event	public delegateGpiStart OnGpiStart;
Illustrate	<pre>public delegate void delegateGpiStart (LogBaseGpiStart msg);</pre> <p>GPITrigger Start Active Reporting Event: this event is reported when the GPI reaches the configured start condition. LogBaseGpiStart: See the "report object" for details</p>

Report

Naming space	GDotnet.Reader.Api.Protocol.Gx
Object	LogBaseGpiStart
Property	<p>GpiPort: GPI port number (indexed from 0) Level: Level state, 0 low 1 high TriggerTime: Trigger time</p>
Illustrate	GPI Trigger to start actively reporting parameters.

4.12.GPI trigger an end event

Naming space	GDotnet.Reader.Api.DAL
kind	GClient
Event	public delegateGpiOver OnGpiOver;
Illustrate	<pre>public delegate void delegateGpiOver(LogBaseGpiOvermsg);</pre> <p>GPITrigger Start Active Reporting Event: this event is reported when the GPI reaches the configured end condition. LogBaseGpiOver: See the "report object" for details</p>

Report

Naming space	GDotnet.Reader.Api.Protocol.Gx
Object	LogBaseGpiOver
Property	<p>GpiPort: GPI port number (indexed from 0)</p> <p>Level: Level state, 0 low 1 high</p> <p>TriggerTime: Trigger time</p>
Illustrate	GPI Trigger end active reporting parameters.

4.13.GPI Status Reporting Events

Naming space	GDotnet.Reader.Api.DAL
Kind	GClient
Event	public delegateGpiAll OnGpiAll;
Illustrate	<p>public delegate void delegateGpiAll(LogBaseGpiAll msg);</p> <p>This message can be proactively reported when the reader needs to proactively upload all GPI status.</p> <p>LogBaseGpiAll: See the "report object" for details</p>

Report

Naming space	GDotnet.Reader.Api.Protocol.Gx
Object	LogBaseGpiAll
Property	<p>gpiAllState: (Index starts at 0)</p> <p>GPI1-bit0: 0 low 1 high</p> <p>GPI2-bit1: 0 low 1 high</p> <p>GPI3-bit2: 0 low 1 high</p> <p>GPI4-bit3: 0 low 1 high</p> <p>GPI5-bit4: 0 low 1 high</p> <p>-----</p>

Illustrate	All GPI status active reporting parameters.
------------	---

4.14.TCP Connection Disconnection Event

Naming space	GDotnet.Reader.Api.DAL
Kind	GClient
Event	<code>public delegateTcpDisconnected OnTcpDisconnected;</code>
Illustrate	<code>public delegate void delegateTcpDisconnected(StringreaderName);</code> Illustrate: <ul style="list-style-type: none"> ➤ The connection is in TCP and this event is reported when the remote connection is actively disconnected or when there is a physical layer anomaly. ➤ After this event is reported, the connection object needs to be released by the host computer (caller), otherwise the event will be reported in a loop until the connection object is released (Close). ➤ Whether to reconnect this remote device or not is controlled autonomously by the host computer (caller) in order to meet different needs. <code>readerName</code> : Connection object name.

4.15.USB-HID disconnection event

Naming space	GDotnet.Reader.Api.DAL
Kind	GClient
Event	<code>public delegateUsbHidRemovedOnUsbHidRemoved;</code>
Illustrate	<code>public delegate void delegateUsbHidRemoved(String readerName);</code> Illustrate: <ul style="list-style-type: none"> ➤ The connection is in USB-HID and this event is reported when the physical layer of the person is abnormal. ➤ After this event is reported, the connection object needs to be released by the host computer (caller), otherwise the event will be reported in a loop until the connection object is released (Close).Whether to reconnect this remote device or not is controlled autonomously by the host computer (caller) in order to meet different needs. <code>readerName</code> : Connection object name.

4.16.TCP Connection Events

Naming space	GDotnet.Reader.Api.DAL
Kind	GServer
Event	public delegateGClientConnected OnClientConnected;
Illustrate	public delegate void delegateGClientConnected(GClient client); TCPbusy listening, This event will be triggered when the remote read/write device actively connects to the host computer. GClient: See the "report object" for details.

Report

Naming space	GDotnet.Reader.Api.DAL
Object	GClient
Property	not have
Illustrate	Illustrate: This connection object is identical to the other actively connected objects and is used in exactly the same way.

5.Message Configuration and Query Instructions

5.1.Send a synchronised message

Naming space	GDotnet.Reader.Api.DAL
Kind	GClient
Method	public void SendSynMsg(Message msg)
Method1	public void SendSynMsg(Message msg,inttimeout)
Method2	public void SendSynMsgRetry(Messagemsg, int timeout, intretry)

Return value	<p>msg.RtCode: Message return code 0 for operation success, non-0 operation failure.</p> <p>msg.RtMsg: Reason for operation failure.</p>
Illustrate	<p>Send synchronous message, see code example.</p> <p><u>Tip: Messages such as 'Reader Configuration and Management' and 'RFID Configuration and Operation' are sent via this method.</u></p>

code example1

```
// Stop command, idle state
MsgBaseStop msgBaseStop = new MsgBaseStop();
clientConn.SendSynMsg(msgBaseStop);
if (0 == msgBaseStop.RtCode)
{
    Console.WriteLine("Stop successful.");
}
else { Console.WriteLine("Stop error."); }
```

code example2

```
// For power configuration, set all 4 antennas to 30dBm.
MsgBaseSetPower msgBaseSetPower = new MsgBaseSetPower();
msgBaseSetPower.DicPower = new Dictionary<byte, byte>()
{
    {1, 30},
    {2, 30},
    {3, 30},
    {4, 30}
};
clientConn.SendSynMsg(msgBaseSetPower);
if (0 == msgBaseSetPower.RtCode)
{
    Console.WriteLine("Power configuration successful.");
}
else { Console.WriteLine("Power configuration error."); }
```

code example3

```
if (null != this.clientConn)
{
    // Query Antenna Power
    MsgBaseGetPower msg = new MsgBaseGetPower();
    this.clientConn.SendSynMsg(msg);
}
```

```
if (0 == msg.RtCode && null != msg.DicPower)
{
    foreach (var item in msg.DicPower)
    {
        switch (item.Key)
        {
            case 1:
            {
                cmbAnt1.SelectedIndex = item.Value;
            }
            break;
            case 2:
            {
                cmbAnt2.SelectedIndex = item.Value;
            }
            break;
            case 3:
            {
                cmbAnt3.SelectedIndex = item.Value;
            }
            break;
            case 4:
            {
                cmbAnt4.SelectedIndex = item.Value;
            }
            break;
            default:
                break;
        }
    }
}
```

6.Message Description

6.1.Reader Configuration and Management

6.1.1.Reboot the reader

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	MsgAppReset
Property	not have
Illustrate	Device reboot message, typically executed after modifying a configuration that requires a reboot to take effect.

6.1.2.Configuring and Querying Serial Port Parameters

Naming space	GDotnet.Reader.Api.Protocol.Gx
Configuration class	MsgAppSetSerialParam
Query class	MsgAppGetSerialParam
Property	BaudrateIndex :baud rate index (0, 9600bps;1, 19200bps;2, 115200bps;3, 230400 bps;4, 460800bps)
Illustrate	<p>Configure the device serial port parameters.</p> <p>Note: This configuration needs to be modified when the device is idle (i.e. the cyclic card reading state cannot change the configuration). Persistent configuration, power off to save, please do not do repeat configuration operation, need to be frequently configured, please query the check first;</p>

6.1.3. Configuring GPO State Parameters

Naming space	GDotnet.Reader.Api.Protocol.Gx
Configuration class	MsgAppSetGpo
Property	<p> Gpo1:0 (Low, relay disconnected) 1 (High, relay closed) Gpo2:0 (Low, relay disconnected) 1 (High, relay closed) Gpo3:0 (Low, relay disconnected) 1 (High, relay closed) Gpo4:0 (Low, relay disconnected) 1 (High, relay closed) </p>
Illustrate	<p>Configure the device GPO parameters.</p> <p>Note: For GPOs that do not need to control the state, no assignment is required.</p>

6.1.4. Query GPI status parameters

Naming space	GDotnet.Reader.Api.Protocol.Gx
Query class	MsgAppGetGpiState
Property	<p>DicGpiState:Level state of the corresponding GPI(Dictionary<byte,byte>,key: GPIindex number, value: flatten Status (0 low, 1 high))</p>
Illustrate	<p>Queries the device GPI status.</p> <p>Note: Index numbers start at 1.</p>

6.1.5. Configuring and Querying GPI Trigger Parameters

Naming space	GDotnet.Reader.Api.Protocol.Gx
Configuration class	MsgAppSetGpiTrigger
Query class	MsgAppGetGpiTrigger

Property	<p>GpiPort:GPI port number, indexed from 0</p> <p>TriggerStart:Trigger on (0 trigger off, 1 low trigger, 2 high trigger, 3 rising edge trigger, 4 falling edge trigger, 5 any edge trigger)</p> <p>TriggerCommand:Trigger bind command (Hex, can be null)</p> <p>BtriggerCommand:Trigger binding command (Byte[], can be empty)</p> <p>TriggerOver:Trigger stop (0 no stop, 1 low level trigger, 2 high level trigger, 3 rising edge trigger, 4 falling edge trigger, 5 any edge trigger, 6 delay stop)</p> <p>OverDelayTime:Delayed stop time (effective only if the stop condition is ‘ delayed stop’)</p> <p>LevelUploadSwitch:IO level change upload switch when triggered without stopping (0 no upload, 1 upload)</p>
Illustrate	<p>(Persistent Configuration, Power-Off Saving)Configure the device GPI trigger parameters.</p> <p>Note: This configuration needs to be modified when the device is idle (i.e. the cyclic card reading state cannot change the configuration).</p> <p>Persistent configuration, power off to save, please do not do repeat configuration operation, need to be frequently configured, please query the check first;</p>

6.1.6.Query baseband software version

Naming space	GDotnet.Reader.Api.Protocol.Gx
Query class	MsgAppGetBaseVersion
Property	Version :Baseband software version number (such as:“ 0.1.0.0 ”)
Illustrate	No have

6.1.7.Query read-write information

Naming space	GDotnet.Reader.Api.Protocol.Gx
Query class	MsgAppGetReaderInfo

Property	Imei :Read/write water number PowerOnTime : power-on time BaseBuildDate : Baseband compilation time AppVersion : Application software version (such as: " 0.1.0.0") App BuildDate : Application compilation time SystemVersion : Operating system version
Illustrate	No have.

6.1.8.Enquire about reader RFID capabilities

Naming space	GDotnet.Reader.Api.Protocol.Gx
Query class	MsgBaseGetCapabilities
Property	MaxPower :Maximum Supported Power MinPower : Minimum Supported Power AntennaCount : Number of antennas FrequencyArray : the list of supported frequency bands, 0, national standard 920~925MHz 1, national standard 840~845MHz 2, national standard 840~845MHz and 920~925MHz 3, FCC, 902~928MHz 4, ETSI, 866~868MHz ProtocolArray : List of supported protocols, 0, ISO18000-6C/EPCC1G2 1, ISO18000-6B 2, national standard GB/T29768-2013 3, national military standards GJB7383.1-2011
Illustrate	No have

6.1.9. Getting tag cache data

Naming space	GDotnet.Reader.Api.Protocol.Gx
Query class	MsgAppGetCache
Property	RtCode is cached when it is 0 and not cached when it is 1.
Illustrate	<ul style="list-style-type: none"> ➤ To get the cached tag data, you need to enable the ' TCP break transmission' function in ' Reader Management Software' -> ' Device Configuration' . ➤ Offline cached data, which needs to be actively fetched via this message, and tagged data is reported via the tag up event, accompanied by the time the tag was actually read.

6.1.10. Empty tag cache data

Naming space	GDotnet.Reader.Api.Protocol.Gx
Query class	MsgAppClearCache
Property	No have
Illustrate	Empty the offline cache of tag data.

6.1.11. Buzzer control

Naming space	GDotnet.Reader.Api.Protocol.Gx
Configuration class	MsgAppSetBeep
Property	<p>OnOff:0 (Buzzer stop) 1 (The buzzer sounds.)</p> <p>BeepType:0 (Buzzer sounds once) 1 (Buzzer always sounds)</p>

Illustrate	Used to operate the reader's internal buzzer.
------------	---

6.2.RFIDConfiguration and Operation

6.2.1.stop command

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	MsgBaseStop
Property	No have
Illustrate	<p>Stops all RFID operations of the reader and puts the reader into an idle state.</p> <p><u>Tip: When the reader is in the read state, all configuration messages will not be sent and a stop command must be sent.</u></p>

6.2.2.Configure and query reader power

Naming space	GDotnet.Reader.Api.Protocol.Gx
Configuration class	MsgBaseSetPower
Query class	MsgBaseGetPower
Property	<p>DicPower:Corresponding antenna power of the reader(Dictionary<byte,byte>,key: Antenna index number, value: Antenna power value)</p> <p>Flag: Configuration class-0, Simultaneous configuration of read and write power (default); 1, Configure read power only; 2, Configure write power only; Query class-0, Query read power; 1, Query write power; default query read/write power;</p> <p>IsPersistence: 0, No saving on power down; 1, Power-down save (default);</p>
Illustrate	<p>Persistent configuration, power off to save, please do not do repeat configuration operation, need to be frequently configured, please query the check first;</p> <p>Configure the reader power for each antenna port.</p>

6.2.3. Configure and query reader operating bands

Naming space	GDotnet.Reader.Api.Protocol.Gx
Configuration class	MsgBaseSetFreqRange
Query class	MsgBaseGetFreqRange
Property	FreqRangeIndex : The band index, with specific correspondences, is detailed in Appendix 1.
Illustrate	<p>Persistent configuration, power off to save, please do not do repeat configuration operation, need to be frequently configured, please query the check first;</p> <p>Used to configure the current operating band of the reader.</p>

6.2.4. Configuring and Querying EPC Baseband Parameters

Naming space	GDotnet.Reader.Api.Protocol.Gx
Configuration class	MsgBaseSetBaseband
Query class	MsgBaseGetBaseband
Property	<p>BaseSpeed: EPCBaseband rate (optional)</p> <p>QValue: Default Q value (selectable)(0~15)。</p> <p>Session: (selectable) (0, Session0; 1, Session1; 2, Session2; 3, Session3)。</p> <p>InventoryFlag: Inventory flag parameter (selectable) (0, only FlagA inventory; 1, only FlagB inventory; 2, alternate FlagA and FlagB)。</p>
Illustrate	<p>Persistent configuration, power off to save, please do not do repeat configuration operation, need to be frequently configured, please query the check first;</p> <p>Used to configure the baseband parameters used by the reader.</p>

6.2.5. Configure and query GB baseband parameters

Naming space	GDotnet.Reader.Api.Protocol.Gx
Configuration class	MsgBaseSetGbBaseband

Query class	MsgBaseGetGbBaseband
Property	<p>RateBit7:0, Tc=6.25us; 1, Tc=12.5us (selectable)。</p> <p>RateBit6:0, Ttext=0, no introduction tone (music); 1, Ttext=1, leading tone (in phonetics) (selectable)。</p> <p>RateK:reverse link factorK, 0, K=1/5; 1, K=3/7; 2, K=6/11; 3, K=1; 4, K=2/5; 5, K=6/7; 6, K=12/11; 7, K=2; 8~15, retention (selectable)。</p> <p>RateReverseCoding:reverse link encoding, 0, FM0; 1, Miller=2; 2, Miller=4; 3, Miller=8 (selectable)。</p> <p>CIN:Anti-collision parameters, continuous idle thresholds, CIN, The default value is 4 (selectable)。</p> <p>CCN:Anti-collision parameters, continuous collision threshold, CCN, The default value is 3 (selectable)。</p> <p>Session: (selectable) (0,Session0;1,Session1;2,Session2;3,Session3)。</p> <p>InventoryFlag:Inventory flag parameter (optional) (0, only FlagA inventory; 1, only FlagB inventory; 2, rotate Flag)</p>
Illustrate	(Persistent Configuration , Power-Off Saving)Used to configure the national standard baseband parameters used by the reader.

6.2.6.Configuring and Querying Label Upload Parameters

Naming space	GDotnet.Reader.Api.Protocol.Gx
Configuration class	MsgBaseSetTagLog
Query class	MsgBaseGetTagLog
Property	<p>RepeatedTime:Repeat label filter time (selectable) (Indicates that the same tag content will be uploaded only once within a read command execution cycle within the specified repeat filtering time , 0~65535, time unit: 10ms)。</p> <p>RssiTV:RSSIthresholds (selectable)(Tag data will not be uploaded and discarded if the tag RSSI value is below the threshold)。</p>
Illustrate	<p>Persistent configuration, power off to save, please do not do repeat configuration operation, need to be frequently configured, please query the check first;</p> <p>Configuration tag active upload parameters.</p>

6.2.7.Read EPC tags

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	MsgBaseInventoryEpc
Property	<p>AntennaEnable:antenna port (Using antenna enumeration, see Quick Start for details)</p> <p>InventoryMode:Continuous/single reading(0: Single reading mode, the reader reads the card as far as possible on each enabled antenna to carry out a round of card reading operation will end the card reading operation and automatically enter the idle state; 1: Continuous reading mode, the reader has been reading the card operation until the reader receives a stop command to end the reading of the card)</p> <p>Filter:Selection of reading parameters (selectable) (See parameter description for details)</p> <p>ReadTid:TID Read parameters (selectable) (See parameter description for details)</p> <p>ReadUserdata:User data area read parameters (selectable) (See parameter description for details)</p> <p>ReadReserved:Reserved area read parameters (selectable) (See parameter description for details)</p> <p>HexPassword:Access code (selectable)</p> <p>Timeout:Unit (ms)。Single reading, read the first label or timeout time stop; cycle reading, timeout time stop; (selectable) (Supported by some devices)</p>
Illustrate	It is used to configure the tag reading parameters of the reader and start the card reading operation. Any operation to read the tag data needs to get the tag EPC code first, so any card reading operation will get the EPC code.

6.2.8.Write EPC labels

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	MsgBaseWriteEpc
Property	<p>AntennaEnable:Antenna port (Using antenna enumeration, see Quick Start for details)</p> <p>Area: Label data area to be written(0, reserved area; 1, EPC area; 2, TID area; 3, user data area)Start: Starting address of the word to be written to the label data area</p> <p>HexWriteData: Contents of data to be written (selectable) (hexadecimal)</p> <p>BwriteData: Contents of data to be written</p> <p>Filter:Selection of reading parameters (selectable) (See parameter description for details) HexPassword:access code (selectable)</p>

Illustrate	<ul style="list-style-type: none"> ➤ The reader performs a write operation on the EPC tag, and the write operation defined in this instruction is a single operation ➤ ISO18000-6C The protocol specifies that the minimum data unit for read and write operations is a word. ➤ The EPC area consists of CRC-16 (0th word) + PC (1st word) + EPC: CRC16: 0th word, unwritable PC: The first word, the first 5 bit bits indicate the EPC word length, i.e. the PC is calculated by shifting the word length of the EPC by 11 bits to the left.
------------	--

6.2.9. Locking EPC tags

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	MsgBaseLockEpc
Property	<p>AntennaEnable: Antenna port (Using antenna enumeration, see Quick Start for details)</p> <p>Area: Tag data area to be locked (0, inactivation password area; 1, access password area; 2, EPC area; 3, TID area; 4, user data area)</p> <p>Mode: Lock Operation Type (0, unlocked; 1, locked; 2, permanently unlocked; 3, permanently locked)</p> <p>Filter: Selection of reading parameters (selectable) (See parameter description for details)</p> <p>HexPassword: access code (selectable)</p>
Illustrate	Lock or unlock operation on the tag, the operation defined by this instruction is a single operation

6.2.10. Inactivation of EPC labels

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	MsgBaseDestoryEpc
Property	<p>AntennaEnable: Antenna port (Using antenna enumeration, see Quick Start for details)</p> <p>HexPassword: Destroy password</p> <p>Filter: Selection of reading parameters (selectable) (See parameter description for details)</p>
Illustrate	Inactivation of the label, after inactivation, the label will be permanently invalidated, this operation is irreversible. The operation defined in this instruction is a single operation

6.2.11.Read 6B tags

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	MsgBaseInventory6b
Property	<p>AntennaEnable:Antenna port (Using antenna enumeration, see Quick Start for details)</p> <p>InventoryMode:Continuous/single reading(0: Single reading mode, the reader reads the card as far as possible on each enabled antenna to carry out a round of card reading operation will end the card reading operation and automatically enter the idle state; 1: Continuous reading mode, the reader has been reading the card operation until the reader receives a stop command to end the reading of the card)</p> <p>Area: Read content(0, read 6BTID only; 1, read 6BTID + user data; 2, read user data only)</p> <p>ReadUserdata:User data area read parameters (selectable) (See parameter description for details)</p> <p>HexMatchTid: TID code of the 6B tag to be matched (selectable) (hexadecimal)</p> <p>BMatchTid: TID code of the 6B tag to be matched (selectable)</p> <p>Timeout:unit (ms). Single reading, read the first label or timeout time stop; cycle reading, timeout time stop; (selectable) (Supported by some devices)</p>
Illustrate	Data read operation for ISO 18000-6B tags

6.2.12.Write 6B labels

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	MsgBaseWrite6b
Property	<p>AntennaEnable:Antenna port (Using antenna enumeration, see Quick Start for details)</p> <p>HexMatchTid: TID code of the 6B tag to be matched (selectable) (hexadecimal)</p> <p>BMatchTid: TID code of the 6B tag to be matched</p> <p>Start: Starting address of the byte to be written to the label data area</p> <p>HexWriteData: Contents of data to be written (selectable)</p> <p>BwriteData: Contents of data to be written</p>
Illustrate	Write operation to 6B tag, the write operation defined by this instruction is a single operation

6.2.13.6B tag locked

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	MsgBaseLock6b
Property	AntennaEnable :Antenna port (Using antenna enumeration, see Quick Start for details) HexMatchTid : TID code of the 6B tag to be matched (selectable) (hexadecimal) BMatchTid : TID code of the 6B tag to be matched LockIndex : Byte address of the data to be locked
Illustrate	Lock operation on 6B tag data, the operation is irrevocable and irreversible, the lock operation defined by this instruction is a single operation

6.2.14.6B Tag Lock Enquiry

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	MsgBaseLock6bGet
Property	AntennaEnable :Antenna port (Using antenna enumeration, see Quick Start for details) HexMatchTid : TID code of the 6B tag to be matched (selectable) (hexadecimal) BMatchTid : TID code of the 6B tag to be matched LockIndex : Byte address of the data to be locked
Illustrate	To query the 6B tag data lock status, the lock query operation defined in this instruction is a single operation

6.2.15.Read GB tags

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	MsgBaseInventoryGb

Property	<p>AntennaEnable:Antenna port (Using antenna enumeration, see Quick Start for details)</p> <p>InventoryMode:Continuous/single reading(0: Single reading mode, the reader reads the card as far as possible on each enabled antenna to carry out a round of card reading operation will end the card reading operation and automatically enter the idle state; 1: Continuous reading mode, the reader has been reading the card operation until the reader receives a stop command to end the reading of the card)</p> <p>Filter:Selection of reading parameters (selectable) (See parameter description for details)</p> <p>ReadTid:TIDRead parameters (selectable) (See parameter description for details)</p> <p>ReadUserdata:User data area read parameters (selectable) (See parameter description for details)</p> <p>HexPassword:Access code (selectable)</p> <p>Timeout:unit (ms)。Single reading, read the first label or timeout time stop; cycle reading, timeout time stop: (selectable) (Supported by some devices)</p>
Illustrate	Used to configure the reader's tag Read parameters and start the read operation, any read tag data operation needs to get the tag code first, so any read operation will get the tag code (EPC).

6.2.16.Write GB labels

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	MsgBaseWriteGb
Property	<p>AntennaEnable:Antenna port (Using antenna enumeration, see Quick Start for details)</p> <p>Area: Label data area to be written(0x10, Label coding area0x20, Label Safety Zone0x30~0x3F, user subarea0~15)</p> <p>Start: Starting address of the word to be written to the label data area</p> <p>HexWriteData: Contents of data to be written (selectable) (hexadecimal)</p> <p>BwriteData: Contents of data to be written</p> <p>Filter:Selection of reading parameters (selectable) (See parameter description for details)</p> <p>HexPassword:Access code (selectable)</p>
Illustrate	<ul style="list-style-type: none"> ➤ The reader performs a write operation on the GB tag, and the write operation defined by this instruction is a single operation ➤ The GB protocol specifies that the minimum data unit for a read/write operation is a word.

6.2.17.Lock GB Label

Naming space	GDotnet.Reader.Api.Protocol.Gx
--------------	--------------------------------

Kind	MsgBaseLockGb
Property	<p>AntennaEnable:Antenna port (Using antenna enumeration, see Quick Start for details)</p> <p>Area: Tag data area to be locked(0x00Label information area, 0x10Label coding area, 0x20Label security area, 0x30~0x3FUser sub-area 0~15)</p> <p>Mode: Lock Operation Type 0x00, readable and writable 0x01, Readable and unwritable. 0x02, Not readable or writable. 0x03, Unreadable and unwritable. 0x11, The security mode is set to No Authentication Required; this operating area must be a label security zone. 0x12, The security mode is set to require authentication and no secure communication; this operating area must be a tag security zone. 0x13, The security mode is set to Authentication Required, Secure Communication Required; this operating area must be a tag security zone.</p> <p>Filter:Selection of reading parameters (selectable) (See parameter description for details)</p> <p>HexPassword:Access code (selectable)</p>
Illustrate	Lock or unlock operation on the tag, the operation defined by this instruction is a single operation

6.2.18.Inactivated GB labels

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	MsgBaseDestoryGb
Property	<p>AntennaEnable:Antenna port (Using antenna enumeration, see Quick Start for details)</p> <p>HexPassword:Destroy password</p> <p>Filter:Selection of reading parameters (selectable) (See parameter description for details)</p>
Illustrate	Inactivation of the label, after inactivation, the label will be permanently invalidated, this operation is irreversible. The operation defined in this instruction is a single operation

6.2.19.Read the GJB label

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	MsgBaseInventoryGJb

Property	<p>AntennaEnable:Antenna port (Using antenna enumeration, see Quick Start for details)</p> <p>InventoryMode:Continuous/single reading(0: Single reading mode, the reader reads the card as far as possible on each enabled antenna to carry out a round of card reading operation will end the card reading operation and automatically enter the idle state; 1: Continuous reading mode, the reader has been reading the card operation until the reader receives a stop command to end the reading of the card)</p> <p>Filter:Selection of reading parameters (selectable) (See parameter description for details)</p> <p>ReadTid:TIDRead parameters (selectable) (See parameter description for details)</p> <p>ReadUserdata:User data area read parameters (selectable) (See parameter description for details)</p> <p>HexPassword:Access code (selectable)</p> <p>Timeout:unit (ms)。Single reading, read the first label or timeout time stop; cycle reading, timeout time stop; (selectable) (Supported by some devices)</p>
Illustrate	<p>It is used to configure the tag reading parameters of the reader and start the reading operation. Any operation to read the tag data needs to get the tag code first, so any reading operation will get the tag code (EPC).</p>

6.2.20.Writing GJB Tags

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	MsgBaseWriteGJb
Property	<p>AntennaEnable:Antenna port (Using antenna enumeration, see Quick Start for details)</p> <p>Area: Label data area to be written(0x01Label coding area, 0x02Label security area, 0x03User data area)Start: Starting address of the word to be written to the label data area</p> <p>HexWriteData: Contents of data to be written (selectable) (hexadecimal)</p> <p>BwriteData: Contents of data to be written</p> <p>Filter:Selection of reading parameters (selectable) (See parameter description for details)</p> <p>HexPassword:Access code (selectable)</p>
Illustrate	<ul style="list-style-type: none"> ➤ The reader performs a write operation on the GJB tag, and the write operation defined by this instruction is a single operation ➤ The GJB protocol specifies that the minimum data unit for read and write operations is a word.

6.2.21.Locking GJB Tags

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	MsgBaseLockGJb
Property	<p>AntennaEnable:Antenna port (Using antenna enumeration, see Quick Start for details)</p> <p>Area: Tag data area to be locked(0x00Label information area, 0x01Label coding area, 0x02Label security area, 0x03User data area)</p> <p>Mode: Lock Operation Type 0x00, Readable and writable. 0x01, Readable and unwritable. 0x02, Not readable or writable. 0x03, Unreadable and unwritable.</p> <p>Filter:Selection of reading parameters (selectable) (See parameter description for details) HexPassword:Access code (selectable)</p>
Illustrate	Lock or unlock operation on the tag, the operation defined by this instruction is a single operation

6.2.22.Inactivation GJB labelling

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	MsgBaseDestoryGJb
Property	<p>AntennaEnable:Antenna port (Using antenna enumeration, see Quick Start for details)</p> <p>HexPassword:Destroy password</p> <p>Filter:Selection of reading parameters (selectable) (See parameter description for details)</p>
Illustrate	Inactivation of the label, after inactivation, the label will be permanently invalidated, this operation is irreversible. The operation defined in this instruction is a single operation

6.2.23.Read TL tags

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	MsgBaseInventoryTl
Property	<p>AntennaEnable:Antenna port (Using antenna enumeration, see Quick Start for details)</p> <p>InventoryMode:Continuous/single reading(0: Single reading mode, the reader reads the card as far as possible on each enabled antenna to carry out a round of card reading operation will end the card reading operation and automatically enter the idle state; 1: Continuous reading mode, the reader has been reading the card operation until the reader receives a stop command to end the reading of the card)</p> <p>Timeout:unit (ms)。Single reading, read the first label or timeout time stop; cycle reading, timeout time stop; (selectable) (Supported by some devices)</p>
Illustrate	It is used to configure the tag reading parameters of the reader and start the reading operation. Any operation to read the tag data needs to get the tag code first, so any reading operation will get the tag code (EPC).

7.Parameter description

7.1.1.6C/GB/GJB Label Selection Parameters

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	ParamEpcFilter
Property	<p>Area:Data area to be matched(1, EPC area; 2, TID area; 3, user data area) (GB label, 0x00Label information area, 0x10Label coding area, 0x20Label security area, 0x30~0x3FUser subarea 0~15)</p> <p>BitStart:Match data start bit address</p> <p>BitLength:Length of data bits to be matched</p> <p>HexData:Data content to be matched (selectable) (hexadecimal)</p> <p>BData:Data content to be matched</p>
Illustrate	<p>Optional parameters</p> <p>The first 32 bits of the EPC area are PC values, so the start bit address when distinguishing EPC is usually 32.</p>

7.1.2.6C Tag reading TID parameters

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	ParamEpcReadTid
Property	Mode : TID Read Mode Configuration, (0, TID reading length is adaptive, but the maximum length does not exceed the length defined by byte 1; 1, read TID according to the length defined by byte 1) Len : The read-write needs to read the word (word, 16bits, below) length of the TID data
Illustrate	Optional parameters

7.1.3.6C Label reads user data area parameters

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	ParamEpcReadUserdata
Property	Start : starting word address Len : Word length of user data to be read by the reader/writer
Illustrate	Optional parameters

7.1.4.6B Tag Read User Data Area Parameters

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	Param6bReadUserdata
Property	Start : User data start byte address Len : User data byte length
Illustrate	Optional parameters

7.1.5.GB Tag Read User Data Area Parameters

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	ParamGbReadUserdata
Property	ChildArea: user subarea Start: User data start byte address Len: User data byte length
Illustrate	Optional parameters

7.1.6.GJB Tag Read User Data Area Parameters

Naming space	GDotnet.Reader.Api.Protocol.Gx
Kind	ParamEpcReadUserdata
Property	Start: starting word address Len: Word length of user data to be read by the reader/writer
Illustrate	Optional parameters

8.Appendice 1

List of frequency bands supported by the reader

indexing	illustrate
0	national standard 920~925MHz
1	national standard 840~845MHz
2	national standard 840~845MHz and 920~925MHz
3	FCC, 902~928MHz
4	ETSI, 866~868MHz

9.Appendice 2