

API user manual (C#)

V 1.2

Catalog

| | |
|---|----|
| 1. Connect / close reader | 4 |
| 1.1 Get connection object(Create) | 4 |
| 1.2 Connecting the reader(Connect) | 4 |
| 1.3 Shut down the reader(ShutDown) | 4 |
| 2. Instructions for use of the class | 4 |
| 2.1 Gen2.InventoryValue | 4 |
| 2.2 Gen2.Select | 5 |
| 2.3 Gen2.ReadData | 5 |
| 2.4 Gen2.Bank | 5 |
| 2.5 Gen2.WriteData | 5 |
| 2.6 Gen2.LockBank | 6 |
| 2.7 Gen2.BlockErase | 6 |
| 2.8 ReaderInfo | 6 |
| 2.9 Gen2.EmbedTagOp | 7 |
| 3. Basic functions of reader | 7 |
| 3.1 Inventory | 7 |
| 3.2 Stop stocktaking | 8 |
| 3.3 Listening method registration | 8 |
| 3.4 Single tag read | 9 |
| 3.5 Write the tag | 9 |
| 3.6 Lock the tag | 10 |
| 3.7 Destroy the tag | 11 |
| 3.8 Write the EPC | 11 |
| 3.9 Block erasure | 11 |
| 3.10 Read the EPC of the single tag | 12 |
| 3.11 Set GPO output | 12 |
| 3.12 Start the monitoring of GPI | 12 |
| 3.13 Stop the monitoring of GPI | 13 |
| 3.14 EmbedRead | 13 |
| 4. Parameter setting | 13 |
| 4.1 Read power setting | 13 |
| 4.2 Write power setting | 14 |
| 4.3 Frequency band setting | 14 |
| 4.4 Scan time settings | 14 |
| 4.5 Antenna setting | 15 |
| 4.6 Antenna detection on / off | 15 |
| 4.8 DRM settings | 15 |
| 4.9 The setting of return loss | 16 |
| 4.10 Single antenna power setting | 16 |
| 5. Parameter acquisition | 16 |
| 5.1 Get the information of the reader | 16 |
| 5.2 Get the S/N number of the reader | 17 |

| | |
|--|----|
| 5.3 Acquisition of return loss | 17 |
| 5.4 Acquisition of read mode | 17 |
| 5.5 Acquisition of write power | 17 |
| 5.6 Get write times | 18 |
| 5.7 Acquisition of real-time mode parameters | 18 |
| 5.8 Acquisition of single antenna power | 18 |

1.Connect / close reader

1.1 Get connection object(Create)

| | |
|------------|--|
| Definition | <i>public static SReader Create(string uriString)</i> |
| Explain | Get connection object |
| Parameter | The format of <i>uriString</i> : <i>sld://COMX</i> or <i>tcp://IP</i> Explain: <i>sld://COMX</i> is the connection for serial port, <i>COMX</i> is the number of the serial port <i>TCP://IP</i> :is the connection of TCP/IP, <i>IP</i> is the IP address of the reader |
| Return | <i>Sreader</i> object returned successfully |
| Example | Serial port: <i>SReader reader = SReader.create("sld://com4");</i> RJ45: <i>SReader reader = SReader.create("tcp://192.168.1.136");</i> |

1.2 Connecting the reader(Connect)

| | |
|------------|--|
| Definition | <i>public abstract void Connect();</i> |
| Explain | Connecting the reader |
| Parameter | / |
| Example | <i>reader.Connect();</i> |

1.3 Shut down the reader(ShutDown)

| | |
|------------|---|
| Definition | <i>public abstract void ShutDown();</i> |
| Explain | Shut down the reader |
| Parameter | / |
| Example | <i>reader.ShutDown();</i> |

2. Instructions for use of the class

2.1 Gen2.InventoryValue

| | |
|-----------------------|--|
| Parameter description | <i>bool isTID</i> : read TID or not. If it is true, it means counting TID; <i>bool isTarget</i> : appoint the Target or not,ant,scanTime; <i>int adrTID</i> : Read the starting word address of tid; |
|-----------------------|--|

| | |
|---------|--|
| | <p><i>int lenTID</i>: Read the length of TID, range 0-15, unit: word;</p> <p><i>int target</i>: 0x00 for A, 0x01 for B, other values reserved;</p> <p><i>int ant</i>: Specify the working antenna number, 0x80 antenna 1, 0x81 antenna 2, 0x82 antenna 3, 0x83 antenna 4;</p> <p><i>int scanTime</i>: For this polling time, the reader will set the maximum response time as $scantime * 100ms$;</p> <p><i>int q</i>: Q value range is 0-15, initial value is 4;</p> <p><i>int session</i>: 0x00 for S0, 0x01 for S1, 0x02 for S2, 0x0 for S3;</p> |
| Remarks | <p>Two parameters must be specified: Q value and session value;</p> <p><i>AdrTID</i> and <i>lenTID</i> must exist when <i>isTID</i> is true;</p> <p>When <i>isTarget</i> is true, <i>target</i>, <i>ant</i> and <i>scantime</i> must exist;</p> |

2.2 Gen2.Select

| | |
|-----------------------|---|
| Parameter description | <p><i>Gen2.Bank bank</i>: tag memory area</p> <p><i>int maskAdr</i>: Starting address, <i>bits</i>;</p> <p><i>int maskLen</i>: Length of filtered data, <i>bits</i>;</p> <p><i>byte[] MaskData</i> : Fill 0 if the filtered data is not enough;</p> |
|-----------------------|---|

2.3 Gen2.ReadData

| | |
|-----------------------|--|
| Parameter description | <p><i>Gen2.Bank bank</i>: tag memory area;</p> <p><i>int wordPtr</i>: Starting address, unit word;</p> <p><i>int len</i>: Read length, unit words;</p> |
|-----------------------|--|

2.4 Gen2.Bank

| | |
|-------|---|
| Type | Enum enumeration |
| Value | <p><i>RESERVED</i> (0x0),</p> <p><i>EPC</i> (0x1),</p> <p><i>TID</i> (0x2),</p> <p><i>USER</i> (0x3);</p> |

2.5 Gen2.WriteData

| | |
|-----------------------|--|
| Parameter description | <p><i>Gen2.Bank bank</i>: Tag memory area;</p> <p><i>int WordPtr</i>: Starting address, unit word;</p> |
|-----------------------|--|

| | |
|--|--|
| | <i>int WordLen</i> : Starting address, unit word; <i>byte[] data</i> : Data to write; <i>int accessPassword</i> : Access password, if not locked can be 0; |
|--|--|

2.6 Gen2.LockBank

| Type | Enum enumeration |
|-------|---|
| Value | <i>KILL_LOCK (0x00)</i> , //Control kill password read-write protection settings <i>ACCESS_LOCK (0x01)</i> , //Access password read / write protection settings <i>EPC_LOCK (0x02)</i> , // EPC storage area read / write protection settings <i>TID_LOCK (0x03)</i> , //TID storage area read / write protection settings <i>USER_LOCK (0x04)</i> ; //User storage area read / write protection settings |

2.7 Gen2.BlockErase

| | |
|-----------------------|--|
| Parameter description | <i>Gen2.Bank bank</i> : tag memory area <i>int wordPtr</i> : Starting address, unit word <i>int len</i> : Length, unit word <i>int accessPassword</i> : Access password |
|-----------------------|--|

2.8 ReaderInfo

| Method | Explain |
|---|---|
| <i>public int VersionH</i> | Get major version number |
| <i>public int VersionL</i> | Get subversion number |
| <i>public int Power</i> | Get read power |
| <i>public int Scntm</i> | Get query time |
| <i>public int CheckAnt</i> | Antenna detection parameters, 0-turn off antenna detection, 1-turn on antenna detection |
| <i>public int Ant</i> | Get antenna configuration information |
| <i>public int MaxFre</i> | Get the maximum frequency of the current reader |
| <i>public int MinFre</i> | Get the minimum operating frequency of the current reader |
| <i>public int getCurrentRegion()</i> | Get current reader frequency |
| <i>Public String getCurrentRegionString()</i> | Get the current reader band and return string |

2.9 Gen2.EmbedTagOp

| | |
|-----------------------|--|
| Parameter description | <pre>public int q; //Q public int session; //0 or 1 public Gen2.Bank bank; //Read area public int ReadAdr; //Start address public int ReadLen; //Length, in words public int PassWord; //Access password public Boolean isTarget; public int Target; //A/B public int ant; public int ScanTime; //Maximum reading time of an antenna</pre> |
| Instructions | <pre>public EmbedTagOp(int q, int session, Gen2.Bank bank, int ReadAdr, int ReadLen, int PassWord); public EmbedTagOp(int q, int session, Gen2.Bank bank, int ReadAdr, int ReadLen, int PassWord, Boolean isTarget, int target, int ant, int ScanTime);</pre> |

3. Basic functions of reader

3.1 Inventory

| | |
|------------|--|
| Definition | <i>public abstract void Inventory(Gen2.InventoryValue value, Gen2.Select filter)</i> |
| Explain | <p>For tag counting, if four antennas are used, this method needs to be called four times to inventory all. Remark: only one antenna works when this interface is called at a time. For example, if the device is set with four antennas, if all four antennas need to be polled once, it needs to be called four times;</p> <p>Data acquisition and use 3.2;</p> <p>This method will end automatically. If the number of tags is large, the stopping will be slow;</p> <p>If you need to stopping quickly, you can call the 3.2 function. PS: quick stopping only for Impinj R2000 product.</p> |
| Parameter | <p>Value: counting parameter, see 2.1 for details</p> <p>Filter: filter conditions, see 2.2 for details</p> |
| Example | <pre>int q= 4; int session = 1;</pre> |

| | |
|--|--|
| | <code>Gen2.InventoryValue value = new InventoryValue(q, session); reader.Inventory(value,null);</code> |
|--|--|

3.2 Stop stocktaking

| | |
|------------|---|
| Definition | <code>public abstract void Inventory_stop();</code> |
| Explain | Use in cooperation with 3.1 |
| Parameter | / |
| Example | <code>reader.Inventory_stop()</code> |

3.3 Listening method registration

| | |
|------------|---|
| Definition | Tag data callback: <code>protected void OnTagRead(TagData tagData)</code> Abnormal callback: <code>protected void OnReadException(ReaderException exception)</code> GPI status callback: <code>protected void OnReadGpio(Gpio_Pin pin)</code> |
| Explain | Register the listening method. After registering this method, the exceptions generated in the tag data, GPI status, and tag reading process will call the callback for outgoing. Complex operations cannot be processed in the callback method. |
| Parameter | <code>tagData</code> : Tag data class |
| Example | <pre>CountMatchListener listener = new CountMatchListener(); reader.TagRead += listener.TagRead; GPIOListener gpioListener = new GPIOListener(); reader.GpioStatus += gpioListener.GpioStatus; SReadException readException = new SReadException(); reader.ReadException += readException.ReadException; //CountMatchListener Class implementation reference is as follows: class CountMatchListener { public void TagRead(Object sender, TagReadDataEventArgs e) { string epc = e.TagData.EpcString; int ant = e.TagData.Ant; int rssi = e.TagData.Rssi; } }</pre> |


```

}

SReadException Class implementation reference is as follows:
class SReadException
{
public void ReadException(object sender, ReaderExceptionEventArgs
e)
{
}
}

GPIOListener Class implementation reference is as follows:
class GPIOListener
{
    public void GpioStatus(object sender, GpioStatusEventArgs e)
    {
        Console.WriteLine("ID:{0}, High: {1}", e.Gpio_Pin.Id,
e.Gpio_Pin.High);
    }
}

```

3.4 Single tag read

| | |
|------------|---|
| Definition | <i>public abstract byte[] ReadSingleTag(byte[] epcData, Gen2.ReadData readData, Gen2.Select filter)</i> |
| Explain | Read single tag memory area |
| Parameter | <i>epcData</i> : Specify the EPC number to read the tag; <i>readData</i> : Refer to 2.3 for details of the class reading parameters <i>filter</i> : Filter condition Be careful : Epcdata and filter must be one of two choices. They cannot exist at the same time |
| Example | <i>byte[] epc = new byte[] {(byte) 0x11, (byte) 0x22, 0x22, 0x33, 0x33, 0x44, 0x00, 0x3C, 0x33, (byte) 0xD4, (byte)0xDB, (byte) 0xF4};</i> <i>Gen2.ReadData readData = new ReadData(Gen2.Bank.TID, 0, 6);</i> <i>byte[] data = reader.ReadSingleTag(epc, readData, null);</i> |

3.5 Write the tag

| | |
|------------|--|
| Definition | <i>public abstract void WriteMemory(byte[] epcData, Gen2.WriteData writeData, Gen2.Select filter)</i> |
|------------|--|

| | |
|-----------|---|
| Explain | Write tag memory |
| Parameter | <p><i>epcData</i>: Specifies the EPC number of the write tag; <i>writeData</i>: Refer to 2.5 for details of the class to write parameters <i>filter</i>: Filter condition</p> <p>Be careful : Epcdata and filter must be one of two choices. They cannot exist at the same time</p> |
| Example | <pre>byte[] newData = new byte[] {0x11,0x22,0x22,0x33,0x33,0x44}; Gen2.WriteData writeData = new WriteData(Gen2.Bank.EPC, 2, 3, newData, 0); reader.WriteMemory(epc, writeData, null);</pre> |

3.6 Lock the tag

| | |
|------------|--|
| Definition | <i>public abstract void LockTag(Gen2.LockBank bank, int action,int accessPassword, byte[] epcData, Gen2.Select filter)</i> |
| Explain | Lock the tag |
| Parameter | <p>Gen2.LockBank bank: Refer to 2.6 for details ; int action: Refer to table 1 below; byte[] epcData: Specify EPC Gen2.Select filter: Epcdata and filter must be one of two choices. They cannot exist at the same time</p> |
| Example | <pre>byte[] epc = new byte[] {(byte) 0x11, (byte) 0x22, 0x22, 0x33, 0x33, 0x44, 0x00, 0x3C, 0x33, (byte) 0xD4, (byte)0xDB, (byte) 0xF4}; Gen2.LockBank bank = Gen2.LockBank.EPC_LOCK; int action = 0x02; int accessPassword = 0x11223344; reader.LockTag(bank, action, accessPassword, epc, null);</pre> |

Table 1:

| bank | action | Explain |
|----------------|--------|-------------------------------------|
| 0x00/0x01 | 0x00 | Unprotected readable and writable |
| | 0x01 | Always readable and writable |
| | 0x02 | Readable and writable with password |
| | 0x03 | Never read or write |
| 0x02/0x03/0x04 | 0x00 | Unprotected writable |
| | 0x01 | Always write |
| | 0x02 | Write with password |
| | 0x03 | Never write |

3.7 Destroy the tag

| | |
|------------|---|
| Definition | <i>public abstract void KillTag(byte[] epcData, int killPassword, Gen2.Select filter)</i> |
| Explain | Destroy tag, invalid tag after destruction |
| Parameter | <i>byte[] epcData</i> : Specify EPC <i>int killPassword</i> : Destroy password <i>Gen2.Select filter</i> : Epcdata and filter must be one of two choices. They cannot exist at the same time |
| Example | <pre><i>byte[] epc = new byte[] {(byte) 0x11, (byte) 0x22, 0x22, 0x33, 0x33, 0x44, 0x00, 0x3C, 0x33, (byte) 0xD4, (byte)0xDB, (byte) 0xF4};</i> <i>int killPassword = 0x11223344;</i> <i>reader.KillTag(epc, killPassword, null);</i></pre> |

3.8 Write the EPC

| | |
|------------|--|
| Definition | <i>public abstract void WriteEPC(Gen2.WriteData writeData)</i> |
| Explain | Write EPC only, only one tag can exist in the antenna field |
| Parameter | <i>writeData</i> : Refer to 2.5 for details |
| Example | <pre><i>byte[] newData = {0x11,0x22,0x22,0x33};</i> <i>int len = 2;</i> <i>int accessPassword = 0;</i> <i>Gen2.WriteData data = new WriteData(len, data, accessPassword);</i> <i>reader.WriteEPC(writeData);</i></pre> |

3.9 Block erasure

| | |
|------------|---|
| Definition | <i>public abstract void BlockErase(Gen2.BlockErase block, byte[] epcData, Gen2.Select filter)</i> |
| Explain | Block erasure |
| Parameter | <i>Block</i> : Refer to 2.7 for details; <i>epcData</i> : Specify EPC; <i>filter</i> : Epcdata and filter must be one of two choices. They cannot exist at the same time |
| Example | <pre><i>byte[] epc = new byte[] {(byte) 0x11, (byte) 0x22, 0x22, 0x33, 0x33, 0x44, 0x00, 0x3C, 0x33, (byte) 0xD4, (byte)0xDB, (byte) 0xF4};</i> <i>Gen2.Bank bank = Gen2.Bank.USER;</i> <i>int wordPtr = 0;</i> <i>int len = 4;</i></pre> |

| | |
|--|--|
| | <pre>int accessPassword = 0; Gen2.BlockErase block = <u>new BlockErase(bank, wordPtr, len, accessPassword);</u> reader.BlockErase(block, epc, null);</pre> |
|--|--|

3.10 Read the EPC of the single tag

| | |
|------------|--|
| Definition | <i>public abstract TagData ReadSingleEPC()</i> |
| Explain | Read the EPC of the single tag |
| Parameter | / |
| Example | <u>TagData tagData = reader.ReadSingleEPC();</u> |

3.11 Set GPO output

| | |
|------------|--|
| Definition | <i>public abstract void setGPO(int id, int high, int time)</i> |
| Explain | Set GPO output high and low level, this method is only applicable to TCP / IP connection |
| Parameter | <i>id</i> : 1-GPO1,2-GPO2,3-GPO3 <i>high</i> : 0-low level, 1-high level <i>time</i> : unit: second, 0 indicates the output low level all the time; other values indicate the duration of high level, when the duration of high level ends, the reader will automatically pull the low level |
| Example | <pre>int id = 0; int high = 1; int time = 2; reader.setGPO(id, high, time);</pre> |

3.12 Start the monitoring of GPI

| | |
|------------|--|
| Definition | <i>public abstract void Gpi_Listeners_start();</i> |
| Explain | Start the GPI status monitoring method, which is only applicable to TCP / IP connections |
| Parameter | / |
| Example | <pre><u>reader.Gpi_Listeners_start();</u> //GPI data return reference 3.2</pre> |

3.13 Stop the monitoring of GPI

| | |
|------------|---|
| Definition | <code>public abstract void Gpi_Listeners_stop();</code> |
| Explain | Stop GPI status listening, this method is only applicable to TCP / IP connections |
| Parameter | / |
| Example | <code>reader.Gpi_Listeners_stop();</code> |

3.14 EmbedRead

| | |
|------------|---|
| Definition | <code>public abstract void Invenry_mix(Gen2.EmbedTagOp embed, Gen2.Select filter)</code> |
| Explain | Gen2.EmbedTagOp: 2.9 |
| Parameter | |
| Example | <pre>//Read TID Int q = 4; Int session = 1; Gen2.Bank bank = Gen2.Bank.TID; Int raedAdr = 0; Int readlen = 6; Int passWord = 0; Int ant = 1; Int time1 = 10; Gen2.EmbedTagOp embedTagOp = new Gen2.EmbedTagOp(q, session, bank, readAdr, readlen, passWord, true, 0, ant, time1);</pre> |

4. Parameter setting

4.1 Read power setting

| | |
|------------|---|
| Definition | <code>public abstract void SetReaderPower(int power)</code> |
| Explain | Read power setting |
| Parameter | <p>power : The power range of Desktop readers,6db integrated reader are 0-26 (these readers Write power and read power are set uniformly, and cannot be set separately) ;</p> <p>The reader power range of 4-channel reader, 8-Channel reader, 16-channel reader and 32-channel reader,9db integrated reader is 0-30</p> |
| Example | <code>reader.SetReaderPower(30);</code> |

4.2 Write power setting

| | |
|------------|---|
| Definition | <i>public abstract void SetWritePower(int enable, int power)</i> |
| Explain | Write power setting, only 4-channel reader, 8-Channel reader, 16-channel reader and 32-channel reader, 9db integrated readers support this function |
| Parameter | <i>Enable</i> : enable write power setting, 0-off, 1-on; <i>Power</i> : power 0 ~ 30; |
| Example | <i>reader. SetWritePower(1, 30);</i> |

4.3 Frequency band setting

| | |
|------------|---|
| Definition | <i>public abstract void SetRegion(Region region)</i> |
| Explain | Frequency band setting |
| Parameter | region: Frequency band, refer to table 2 below |
| Example | <i>Region region = Region.Chinese2;</i> <i>reader.SetRegion(region);</i> |

Table2

| Type | Enum enumeration | Explain |
|-------|------------------|-------------------------|
| value | <i>Chinese2</i> | <i>China 920~925MHz</i> |
| | <i>US</i> | <i>USA 902~925MHz</i> |
| | <i>Korean</i> | |
| | <i>EU</i> | <i>European</i> |
| | <i>UKraine</i> | |
| | <i>Peru</i> | |
| | <i>Chinese1</i> | |
| | <i>EU3</i> | |
| | <i>US3</i> | |
| | <i>Taiwan</i> | |

4.4 Scan time settings

| | |
|------------|--|
| Definition | <i>public abstract void SetScanTime(int time)</i> |
| Explain | Module query time |
| Parameter | <i>Time</i> : the value range is 0 ~ 255; the query time of the reader is 0 * 100ms ~ 255 * 100ms; if it is 0, there is no upper limit for the query |

| | |
|---------|--|
| | time of the reader, and it will not exit until all tags are queried; |
| Example | <i>int time = 20;</i> <i>reader.SetScanTime(time);</i> |

4.5 Antenna setting

| | |
|------------|--|
| Definition | <i>public abstract void SetAntenna(int[] antList)</i> |
| Explain | antenna setting, only 4-channel reader, 8-Channel reader, 16-channel reader and 32-channel reader,9db integrated readers support this function |
| Parameter | <i>antList</i> : Antenna list |
| Example | <i>int[] antList = {1,2,3,4} //Set the antenna1,2,3,4</i> <i>reader.SetAntenna(antList);</i> |

4.6 Antenna detection on / off

| | |
|------------|---|
| Definition | <i>public abstract void SetCheckAnt(int value)</i> |
| Explain | antenna detection on / off,only 4-channel reader, 8-Channel reader, 16-channel reader and 32-channel reader,9db integrated readers support this function. |
| Parameter | <i>Value</i> : 0-off, 1-on |
| Example | <i>int value = 1;</i> <i>reader.SetCheckAnt(value);</i> |

4.7 The setting of rewrite times

| | |
|------------|--|
| Definition | <i>public abstract void SetRetryTimes(int value)</i> |
| Explain | The number of times to write the tag again when the tag writing fails , only 4-channel reader, 8-Channel reader, 16-channel reader and 32-channel reader,9db integrated readers support this function. |
| Parameter | <i>Value</i> : Value range 0-7; |
| Example | <i>int value = 3;</i> <i>reader. SetRetryTimes(value);</i> |

4.8 DRM settings

| | |
|------------|---|
| Definition | <i>public abstract void SetDRM(bool enable)</i> |
| Explain | Enable or disable DRM function,only 4-channel reader, 8-Channel |

| | |
|-----------|---|
| | reader, 16-channel reader and 32-channel reader,9db integrated readers support this function. |
| Parameter | <i>enable</i> : <i>false</i> - close DRM, <i>true</i> - enable DRM |
| Example | <i>reader.SetDRM(true);</i> |

4.9 The setting of return loss

| | |
|------------|--|
| Definition | <i>public abstract void SetReturnLoss(int value)</i> |
| Explain | Set return loss threshold,only 4-channel reader, 8-Channel reader, 16-channel reader and 32-channel reader,9db integrated readers support this function. |
| Parameter | Value : value range 0 ~ 20; |
| Example | <i>int value = 20;</i> <i>reader. SetReturnLoss(value);</i> |

4.10 Single antenna power setting

| | |
|------------|--|
| Definition | <i>public abstract void SetPortReadPower(int[] antPower)</i> |
| Explain | The power of each antenna is set separately, only 4-channel reader, 8-Channel reader, 16-channel reader and 32-channel reader,9db integrated readers support this function.If the reader is 4 channels, the array length is 4; the array length of 8 channels is 8; the array length of 16-channels is 16. |
| Parameter | <i>antPower</i> : Power array |
| Example | <i>int[] antPower = {30,30,30,28}; //4-channel power setting</i> <i>reader. SetPortReadPower(antPower);</i> |

5. Parameter acquisition

5.1 Get the information of the reader

| | |
|--------------|---|
| Definition | <i>public abstract ReaderInfo GetReaderInfo()</i> |
| Explain | Get basic reader information |
| Return value | <i>ReaderInfo</i> : Refer to 2.8 for details |
| Example | <i>ReaderInfo info = reader. GetReaderInfo();</i> |

5.2 Get the S/N number of the reader

| | |
|--------------|---|
| Definition | <i>public abstract string GetSerialNO()</i> |
| Explain | Get the S/N number of the reader |
| Return value | Character string |
| Example | <i>string serialNo = reader. GetSerialNO();</i> |

5.3 Acquisition of return loss

| | |
|--------------|---|
| Definition | <i>public abstract int GetReturnLoss()</i> |
| Explain | Acquisition of return loss threshold, only 4-channel reader, 8-Channel reader, 16-channel reader and 32-channel reader, 9db integrated readers support this function. |
| Return value | int |
| Example | <i>int value = reader. GetReturnLoss();</i> |

5.4 Acquisition of read mode

| | |
|--------------|--|
| Definition | <i>public abstract int GetReadMode()</i> |
| Explain | Acquisition of read mode |
| Return value | 0-answer mode, 1-real time mode |
| Example | <i>int mode = reader. GetReadMode();</i> |

5.5 Acquisition of write power

| | |
|--------------|--|
| Definition | <i>public abstract int GetWritePower()</i> |
| Explain | get write power, only 4-channel reader, 8-Channel reader, 16-channel reader and 32-channel reader, 9db integrated readers support this function. |
| Return value | int |
| Example | <i>int power = reader. GetWritePower();</i> |

5.6 Get write times

| | |
|--------------|--|
| Definition | <i>public abstract int GetRetryTimes()</i> |
| Explain | get number of rewrites, only 4-channel reader, 8-Channel reader, 16-channel reader and 32-channel reader,9db integrated readers support this function. |
| Return value | int |
| Example | <i>int times = reader. GetRetryTimes();</i> |

5.7 Acquisition of real-time mode parameters

| | |
|--------------|--|
| Definition | <i>public abstract RealTimeParam GetRealTimeParam()</i> |
| Explain | acquisition of real-time mode parameters, only 4-channel reader, 8-Channel reader, 16-channel reader and 32-channel reader,9db integrated readers support this function. |
| Return value | <i>RealTimeParam</i> : refer to 2.9 for details |
| Example | <i>RealTimeParam param = reader. GetRealTimeParam();</i> |

5.8 Acquisition of single antenna power

| | |
|--------------|--|
| Definition | <i>public abstract int[] GetPortReadPower()</i> |
| Explain | acquisition of single antenna power,only 4-channel reader, 8-Channel reader, 16-channel reader and 32-channel reader,9db integrated readers support this function. |
| Return value | int |
| Example | <i>int[] power = reader. GetPortReadPower();</i> |