

Instrukcja obsługi

Instrukcja obsługi interfejsu API
(C#)

HD-RD06-UHF

Spis treści

Specyfikacje:.....	Błąd! Nie zdefiniowano zakładki.
Zawartość zestawu:.....	Błąd! Nie zdefiniowano zakładki.
Instalacja:	Błąd! Nie zdefiniowano zakładki.
Jak otworzyć szufladę:.....	Błąd! Nie zdefiniowano zakładki.
Wyjmowanie szuflady:.....	Błąd! Nie zdefiniowano zakładki.
Konserwacja:	Błąd! Nie zdefiniowano zakładki.

1. Podłącz / zamknij czytnik

1.1 Pobierz obiekt połączenia (Utwórz)

Definicja	<i>public static SReader Create(string uriString)</i>
Wyjaśnić	Pobierz obiekt połączenia
Parametr	Format <i>właściwości uriString</i> : <i>sld:///COMX</i> lub <i>tcp://IP</i> Wyjaśnić : <i>sld:///COMX</i> to połączenie dla portu szeregowego, <i>COMX</i> (<i>Jednostka Połączenia</i> to numer portu szeregowego <i>TCP://IP</i> to połączenie TCP/IP, <i>IP</i> to adres IP czytnika
Wrócić	<i>Obiekt czytnika</i> został zwrócony pomyślnie
Przykład	Port szeregowy: <i>SReader reader = SReader.create("sld:///com4");</i> RJ45: <i>SReader reader = SReader.create("tcp://192.168.1.136");</i>

1.2 Podłączanie czytnika (Połącz)

Definicja	<i>public abstract void Connect();</i>
Wyjaśnić	Podłączanie czytnika
Parametr	/
Przykład	<i>czytnik. Połącz();</i>

1.3 Wyłącz czytnik (ShutDown)

Definicja	<i>publiczny abstrakcja void ShutDown();</i>
Wyjaśnić	Wyłącz czytnik
Parametr	/
Przykład	<i>czytnik. Zamykanie();</i>

2. Instrukcja korzystania z zajęć

2.1 Gen2.InventoryValue

Opis parametrów	<i>bool isTID</i> . czytaj TID lub nie. Jeśli to prawda, oznacza to liczenie TID; <i>bool isTarget</i> :wyznacz cel lub nie, <i>ant,scanTime</i> ; <i>int adrTID</i> .Przeczytaj początkowy adres słowa tid;
	<i>int lenTID</i> :Odczytaj długość TID, zakres 0-15, jednostka: słowo; <i>int target</i> :0x00 dla A, 0x01 dla B, inne wartości zarezerwowane; <i>int ant</i> . Określ numer anteny roboczej, 0x80 anteny 1 0x81 antena 2, antena 0x82 3, antena 0x83 4; <i>int scanTime</i> :Dla tego czasu odpytywania czytnik ustawi maksymalny czas odpowiedzi jako czas skanowania * 100ms; <i>int q</i> . Zakres wartości Q wynosi 0-15, wartość początkowa wynosi 4; <i>int session</i> :0x00 dla S0, 0x01 dla S1, 0x02 dla S2, 0x0 dla S3;
Uwagi	Należy określić dwa parametry: wartość Q i wartość sesji; <i>AdrTID</i> i <i>lenTID</i> muszą istnieć, gdy <i>isTID</i> jest prawdziwe;

	Gdy parametr <code>isTarget</code> ma wartość <code>true</code> , <i>musi istnieć element target, ant i scantime</i> ;
--	--

2.2 Gen2.Select (Generacja2).

Opis parametrów	<i>Gen2.Bank bank.tag</i> obszar pamięci <i>int</i> <i>maskAdr.Adres</i> początkowy, <i>bity, int</i> <i>maskLen.Długość</i> filtrowanych danych, <i>bity</i> , <i>byte[] MaskData</i> : Wypełnij 0, jeśli przefiltrowane dane nie są wystarczające;
-----------------	--

2.3 Gen2.ReadData

Opis parametrów	<i>Gen2.Bank bank.tag</i> obszar pamięci; <i>int</i> <i>wordPtr.Adres</i> początkowy, słowo jednostkowe; <i>int</i> <i>len.Długość odczytu</i> , słowa jednostkowe;
-----------------	---

2.4 Gen2.Bank

Typ	Eliminacja
Wartość	<i>ZAREZERWOWANE (0x0)</i> , <i>EPC (0x1)</i> , <i>TID (0x2)</i> , <i>UŻYTKOWNIK (0x3)</i> ;

2.5 Gen2.WriteData

Opis parametrów	<i>Gen2.Bank</i> <i>bank</i> :Obszar pamięci tagów; <i>int</i> <i>WordPtr</i> :Adres początkowy, słowo jednostkowe;
	<i>int WordLen</i> :Adres początkowy, słowo jednostkowe; <i>bajt</i> [] <i>data</i> :Dane do zapisu; <i>int accessPassword</i> :Hasło dostępu, jeśli nie jest zablokowane, może mieć wartość 0;

2.6 Gen2.LockBank

Typ	Eliminacja
Wartość	<i>KILL_LOCK (0x00)</i> , Kontrola ochrony przed odczytem i zapisem hasła kill Ustawienia <i>ACCESS_LOCK (0x01)</i> , Uzyskaj dostęp do ustawień ochrony przed odczytem/zapisem hasła <i>EPC_LOCK (0x02)</i> , ustawienia ochrony przed odczytem/zapisem w obszarze pamięci EPC <i>TID_LOCK (0x03)</i> , ustawienia ochrony przed odczytem/zapisem w obszarze pamięci TID <i>USER_LOCK (0x04)</i> ; Ustawienia ochrony przed odczytem/zapisem w obszarze pamięci użytkownika

2.7 Gen2.BlockErase

Opis parametrów	<i>Gen2.Bank bank.tag</i> obszar pamięci <i>int</i> <i>wordPtr</i> :Adres
-----------------	---

	początkowy, słowo jednostki <i>int</i> <i>len</i>:Długość,słowo jednostki <i>int</i> <i>accessPassword</i>:Hasło dostępu
--	--

2.8 Informacje o czytniku

Metoda	Wyjaśnić
<i>public int VersionH</i>	Pobierz główny numer wersji
<i>public int VersionL</i>	Pobierz numer subversion
<i>public int Moc</i>	Zyskaj moc odczytu
<i>public int Scntm</i>	Pobieranie czasu zapytania
<i>public int CheckAnt</i>	Parametry detekcji anteny, 0-wyłączona detekcja anteny, 1-włączona detekcja anteny
<i>public int ant</i>	Uzyskaj informacje o konfiguracji anteny
<i>public int MaxFre</i>	Uzyskaj maksymalną częstotliwość bieżącego czytnika
<i>public int MinFre</i>	Uzyskaj minimalną częstotliwość pracy aktualnego czytnika
<i>public int getCurrentRegion()</i>	Pobieranie aktualnej częstotliwości odczytu

<i>Publiczny</i>	<i>Struna</i>	Pobierz bieżące pasmo
<i>getCurrentRegionString()</i>		czytnika i zwracany ciąg

2.9 Druga generacja.Osadzanie tagu

Opis parametrów	<pre> Publiczny int q; //Q int sesja; //0 lub 1 publiczny Gen2.Bank bankowy; Obszar odczytu publiczny int ReadAdr; //Adres początkowy publiczny int ReadLen; //Długość, w słowach publiczny int PassWord; //Hasło dostępu publiczny Wartość logiczna isTarget; publiczny int cel; //A/B publiczny int ant; publiczny int ScanTime; //Maksymalny czas odczytu anteny </pre>
Instrukcje	<pre> publiczny EmbedTagOp(int q, int session, Gen2.Bank bank, int ReadAdr, int ReadLen, int Hasło); public EmbedTagOp(int q, int session, Gen2.Bank bank, int ReadAdr, intReadLen,intPassWord,BooleanisTarget,inttar get,intant,int ScanTime); </pre>

3. Podstawowe funkcje czytnika

3.1 Inwentaryzacja

Definicja	<i>public abstract void Inventory(Gen2.InventoryValue wartość, Gen2.Select filtr)</i>
Wyjaśnić	Do zliczania tagów, jeśli używane są cztery anteny, tę metodę należy wywołać cztery razy, aby zinwentaryzować wszystkie. Uwaga: tylko jedna antena działa, gdy ten interfejs jest wywoływany na raz. Na przykład, jeśli urządzenie jest ustawione z czterema antenami, jeśli wszystkie cztery anteny muszą być odpytywane raz, należy je wywołać cztery razy; Gromadzenie i wykorzystywanie danych 3.2; Ta metoda zakończy się automatycznie. Jeśli liczba tagów jest duża, zatrzymywanie będzie powolne; Jeśli chcesz szybko się zatrzymać, możesz wywołać funkcję 3.2. PS: szybkie zatrzymywanie tylko dla produktu Impinj R2000.
Parametr	Wartość: parametr zliczający, szczegóły w 2.1 Filtr: warunki filtrowania, szczegółowe informacje znajdują się w 2.2
Przykład	<i>int q = 4; int sesja = 1;</i>

	<code>Gen2.InventoryValue = NowaWartośćWynalaz sesji e wartość y ku(q,); czytnik. Inventory(wartość; null);</code>
--	---

3.2 Zatrzymaj inwentaryzację

Definicja	<i>abstrakt publiczny nieważny</i> <code>Inventory_stop();</code>
Wyjaśnić	Stosowanie we współpracy z 3.1
Parametr	/
Przykład	<code>czytnik.Inventory_stop()</code>

3.3 Rejestracja metody słuchania

Definicja	Wywołanie zwrotne danych tagu: <i>protected void OnTagRead(TagData tagData)</i> Nietypowe wywołanie zwrotne: <i>protected void OnReadException(wyjątek ReaderException)</i> Wywołanie zwrotne statusu GPI: <i>protected void OnReadGpio(Gpio_Pin pin)</i>
Wyjaśnić	Zarejestruj metodę nasłuchiwanie. Po zarejestrowaniu tej metody, wyjątki wygenerowane w danych tagu, statusie GPI i procesie odczytu tagu oddzwoni do oddzwonienia dla wychodzących. Złożone operacje nie mogą być przetwarzane w metodzie wywołania zwrotnego.
Parametr	<i>tagData</i> : Klasa danych tagu

Przykład	<pre> Listener CountMatchListener = nowy CountMatchListener(); czytelnik. TagRead += sluchacz. EtykietaCzytać; GPIOListener gpioListener = nowy GPIOListener(); Czytelnik. GpioStatus += gpioListener.GpioStatus; SReadException readException = nowy SReadException(); czytelnik. ReadException += readException.ReadException; CountMatchListener (CountMatchListener) Odniesienie do implementacji klasy jest następujące : class CountMatchListener { public void TagRead(Object sender, TagReadDataEventArgs e) { ciąg epc = e.TagData.EpcString; int ant = e.TagData.Ant; int rssi = e.TagData.Rssi; } </pre>
----------	---

	<pre> } Dokumentacja implementacji klasy SReadException jest następująca: klasa SReadException { public void ReadException(object sender, ReaderExceptionEventArgs e) { } } } Dokumentacja implementacji klasy GPIOListener jest następująca: klasa GPIOListener { public void GpioStatus(nadawca obiektu, GpioStatusEventArgs e) { Console.WriteLine("ID:{0}, Wysoki: {1}", e.Gpio_Pin.Id, e.Gpio_Pin.Wysoki); } } } </pre>
--	--

3.4 Odczyt pojedynczego tagu

Definicja	<pre> publicznyabstrakt bajt[] ReadSingleTag(bajt[] epcData, </pre>
-----------	---

	<i>Gen2.ReadData readData, Gen2.Select filtr)</i>
Wyjaśniać	Odczyt obszaru pamięci pojedynczego tagu
Parametr	<i>epcData</i>: Określ numer EPC, aby odczytać tag; <i>readData</i>: Patrz 2.3, aby uzyskać szczegółowe informacje na temat parametrów odczytu klasy, <i>filtr</i>: Warunek filtra Bądź ostrożny : Epcdata i filtr muszą być jedną z dwóch opcji. Nie mogą istnieć w tym samym czasie
Przykład	<pre> <i>bajt[] EPC = nowy bajt[] {(bajty) 0x11, (bajty) 0x22, 0x22, 0x33, 0x33, 0x44, 0x00, 0s3c, 0x33, (bajty) 0xd4, (bajty) 0xdb, (bajty) 0xf4}; Gen2.reddata Reddata = Nowe Reddata (Gen2.bank.tid, 0, 6); Bajt[] dane = czytnik. ReadSingleTag(EPC, Reddata, Nal);</i> </pre>

3.5 Napisz tag

Definicja	<i>public abstract void WriteMemory(byte[] epcData, Gen2.WriteData writeData, Gen2.Select filter)</i>
Wyjaśniać	Zapis pamięci tagów
Parametr	<i>epcData</i>: Określa numer EPC znacznika zapisu; <i>writeData</i>: Patrz 2.5, aby uzyskać szczegółowe informacje na temat klasy do zapisu parametrów <i>filter</i>: Warunek filtru

	Bądź ostrożny : Epcdata i filtr muszą być jedną z dwóch opcji. Nie mogą istnieć w tym samym czasie
Przykład	<pre>byte[] newData = nowy bajt[] {0x11,0x22,0x22,0x33,0x33,0x44}; Gen2.WriteData writeData = new WriteData(Gen2.Bank.EPC, 2, 3, newData, 0); czytnik. WriteMemory(<u>epc</u>, writeData, null);</pre>

3.6 Zablokuj tag

Definicja	<pre>public abstract void LockTag(Gen2.LockBank bank, int action,int accessPassword, byte[] epcData, Gen2.Select filter)</pre>
Wyjaśnić	Blokowanie znacznika
Parametr	Gen2.LockBank bank: Szczegółowe informacje można znaleźć w 2.6 ; int action: Patrz tabela 1 poniżej; byte[] epcData: Określ EPC Gen2.Select filter: Epcdata i filtr muszą być jedną z dwóch opcji. Nie mogą istnieć w tym samym czasie

Przykład	<i>bajt[] epc = nowy bajt[] {(bajt) 0x11, (bajt) 0x22, 0x22, 0x33, 0x33, 0x44, 0x00, 0x3C, 0x33, (bajt) 0xD4, (bajt)0xDB, (bajt) 0xF4};</i> <i>Gen2.LockBank bank =</i> <i>Gen2.LockBank.EPC_LOCK; int action =</i> <i>0x02;</i> <i>int accessPassword = 0x11223344;</i> <i>czytnik.LockTag(bank, akcja,</i> <i>accessPassword, epc, null);</i>
----------	--

Tabela 1:

bank	akcja	Wyjaśnić
0x00/0x01	0x00	Niezabezpieczony, czytelny i zapisywalny
	0x01	Zawsze czytelny i zapisywalny
	0x02	Czytelne i zapisywalne za pomocą hasła
	0x03	Nigdy nie czytaj ani nie zapisuj
0x02/0x03/0x04	0x00	Niezabezpieczony zapisywalny
	0x01	Zawsze pisz
	0x02	Napisz z hasłem
	0x03	Nigdy nie pisz

3.7 Zniszcz tag

Definicja	<i>publicznyabstraktnieważnyKillTag(bajt[]</i> <i>epcData,IntkillPassword,</i>
-----------	--

	<i>Gen2.Select filter)</i>
Wyjaśnić	Zniszcz tag, nieprawidłowy tag po zniszczeniu
Parametr	<i>byte[]</i> <i>epcData</i> : Określ EPC <i>int</i> <i>killPassword</i> : Zniszcz hasło <i>Gen2.Select filter</i> : Epcdata i filtr muszą być jedną z dwóch opcji. Nie mogą istnieć w tym samym czasie
Przykład	<i>byte[] epc = nowy bajt[] {(bajt) 0x11,</i> <i>(bajt) 0x22, 0x22, 0x33, 0x33, 0x44, 0x00,</i> <i>0x3C, 0x33, (bajt) 0xD4, (bajt) 0xDB, (bajt)</i> <i>0xF4}; int killPassword = 0x11223344;</i> <i>czytelnik. KillTag(epc, killPassword, null);</i>

3.8 Napisz EPC

Definicja	<i>public abstract void</i> <i>WriteEPC(Gen2.WriteData writeData)</i>
Wyjaśnić	Napisz tylko EPC, w polu anteny może istnieć tylko jeden tag
Parametr	<i>writeData</i> : Szczegółowe informacje można znaleźć w 2.5
Przykład	<i>byte[] newData =</i> <i>{0x11, 0x22, 0x22, 0x33}; int</i> <i>len = 2;</i> <i>int accessPassword = 0;</i>

	<i>Gen2.WriteData data = <u>new</u> <u>WriteData(len, data, accessPassword)</u>; czytnik. <u>WriteEPC(zapis danych)</u>;</i>
--	--

3.9 Block erasure

Definicja	<i>publiczny abstrakt nieważny BlockErase(<u>Gen2.BlockErase blok</u> <u>byte[] epcData, Gen2.Select filter</u>)</i>
Wyjaśnić	Block erasure
Parametr	<i>Blok: Patrz 2.7, aby uzyskać szczegółowe informacje; epcData: Określ EPC; filter: Epcdata i filter muszą być jedną z dwóch opcji. Nie mogą istnieć w tym samym czasie</i>
Przykład	<i>byte[] epc = nowy bajt[] {(bajt) 0x11, (bajt) 0x22, 0x22, 0x33, 0x33, 0x44, 0x00, 0x3C, 0x33, (bajt) 0xD4, (bajt) 0xDB, (bajt) 0xF4}; Gen2.Bank bank = Gen2.Bank.UŻYTKOWNIK; int słowoPtr = 0; int len = 4;</i>
	<i>int accessPassword = 0; Gen2.BlockErase block = <u>nowy</u> <u>BlockErase(bank, wordPtr, len,</u> <u>accesePassword)</u>; czytelnik. <u>BlockErase(blok, epc, null)</u>;</i>

3.10 Przeczytaj EPC pojedynczego tagu

Definicja	<i>abstrakt publiczny TagData ReadSingleEPC()</i>
Wyjaśnić	Odczytywanie świadectwa charakterystyki energetycznej pojedynczego tagu
Parametr	/
Przykład	<u>TagData tagData = czytnik. ReadSingleEPC();</u>

3.11 Ustaw wyjście GPO

Definicja	<i>public abstract void setGPO(int id, int high, int time)</i>
Wyjaśnić	Ustaw wysoki i niski poziom wyjściowy GPO, ta metoda ma zastosowanie tylko do Połączenie TCP / IP
Parametr	<i>id</i> : 1-GPO1, 2-GPO2, 3-GPO3 <i>wysoki</i> : 0-niski poziom, 1-wysoki poziom <i>czas</i> : jednostka: sekunda, 0 oznacza niski poziom wyjściowy przez cały czas; inne wartości wskazują czas trwania wysokiego poziomu, gdy czas trwania wysokiego poziomu dobiegnie końca, czytnik automatycznie pociągnie niski poziom
Przykład	<i>int id</i> <i>= 0;</i> <i>int</i>

	<i>wysoki</i> <i>= 1; int</i> <i>czas =</i> <i>2;</i> <i>reader.setGPO(identyfikator, wysoki, czas);</i>
--	--

3.12 Rozpocznij monitorowanie GPI

Definicja	<i>abstrakt publiczny nieważny</i> <i>Gpi_Listeners_start();</i>
Wyjaśnić	Uruchom metodę monitorowania stanu GPI, która ma zastosowanie tylko do Połączenia TCP / IP
Parametr	/
Przykład	<u><i>czytelnika.Gpi_Listeners_start();</i></u> Odniesienie do zwracanych danych GPI 3.2

3.13 Zatrzymaj monitorowanie GPI

Definicja	abstrakt publiczny nieważny <i>Gpi_Listeners_stop();</i>
Wyjaśnić	Zatrzymaj nasłuchiwanie stanu GPI, ta metoda ma zastosowanie tylko do połączeń TCP / IP
Parametr	/
Przykład	<i>czytelnika.Gpi_Listeners_stop();</i>

3.14 OsadźCzytaj

Definicja	public abstract void Inventory_mix(Gen2.EmbedTagOp embed, Gen2.Select filter)
Wyjaśnić	Gen2.EmbedTagOp: 2.9

Parametr	
Przykład	Czytaj TID <pre> Int q = 4; Int sesja = 1; Gen2.Bank bank = Gen2.Bank.TID; Int readAdr = 0; Int readlen = 6; Int passWord = 0; Int ant = 1; Int czas1 = 10; Gen2.EmbedTagOp embedTagOp = new Gen2.EmbedTagOp(q, sesja, bank, readAdr, readlen, passWord, true, 0, ant, time1); </pre>

4. Ustawianie parametrów

4.1 Odczytaj ustawienie mocy

Definicja	public abstract void SetReaderPower(int power)
Wyjaśnić	Odczytaj ustawienie zasilania
Parametr	moc : Zakres mocy czytników biurkowych, zintegrowany czytnik 6db wynosi 0-26 (te czytniki Moc zapisu i moc odczytu są ustawione równomiernie i nie można ich ustawić osobno); Zakres mocy czytnika czytnika 4-kanalowego, czytnika 8-kanalowego, czytnika 16-kanalowego i czytnika 32-kanalowego, zintegrowanego czytnika 9db wynosi 0-30

Przykład	czytnik. SetReaderPower(30);
----------	------------------------------

4.2 Ustawienie mocy zapisu

Definicja	<i>public abstract void SetWritePower(int włącz, int moc)</i>
Wyjaśnić	Ustawienie mocy zapisu, tylko czytnik 4-kanalowy, czytnik 8-kanalowy, czytnik 16-kanalowy i czytnik 32-kanalowy, zintegrowane czytniki 9db obsługują tę funkcję
Parametr	<i>Włącz</i> : włącz ustawienie mocy zapisu, 0-wył., 1-wł.; <i>Moc</i> : moc 0 ~ 30;
Przykład	<i>czytnik. SetWritePower(1, 30);</i>

4.3 Ustawienie pasma częstotliwości

Definicja	<i>public abstract void SetRegion(Region regionu)</i>
Wyjaśnić	Ustawienie pasma częstotliwości
Parametr	region:Pasma częstotliwości, patrz tabela 2 poniżej
Przykład	<i>Region region = Region.Chinese2;</i> <i>czytnik. SetRegion(region);</i>

Tabela 2

Typ	Eliminacja	Wyjaśnić
wartość	<i>Język chiński2</i>	<i>Chiny 920 ~ 925 MHz</i>
	<i>NAM</i>	Stany Zjednoczone 902 ~ 925 MHz
	<i>Koreański</i>	
	<i>UE</i>	Europejski

	<i>Ukraina</i>	
	<i>Peru</i>	
	<i>Język chiński</i>	
	<i>UE3</i>	
	<i>US3 powiedział:</i>	
	<i>Tajwan</i>	

4.4 Ustawienia czasu skanowania

Definicja	<i>public abstract void SetScanTime(int time)</i>
Wyjaśnić	Moduł czasu zapytania
Parametr	<i>Czas</i> : zakres wartości wynosi 0 ~ 255; czas zapytania czytnika wynosi 0 * 100ms ~ 255 * 100ms; Jeśli wartość wynosi 0, nie ma górnego limitu dla zapytania
	czas czytnika i nie zakończy się, dopóki wszystkie tagi nie zostaną odpytane;
Przykład	<i>czas int = 20;</i> <i>czytnik. SetScanTime(czas);</i>

4.5 Ustawienia anteny

Definicja	<i>public abstract void SetAntenna(int[] antList)</i>
Wyjaśnić	ustawienie anteny, tylko czytnik 4-kanalowy, czytnik 8-kanalowy, czytnik 16-kanalowy i czytnik 32-kanalowy, zintegrowane czytniki 9db obsługują tę funkcję
Parametr	<i>antList</i> : Lista anten

Przykład	<code>iint[] antList = {1,2,3,4} //Ustaw czytnik anteny1,2,3,4 . UstawAntena(antList);</code>
----------	---

4.6 Wykrywanie anteny włączone / wyłączzone

Definicja	<code>public abstract void SetCheckAnt(int wartość)</code>
Wyjaśnić	wykrywanie anteny włącza / wyłącza, tylko czytnik 4-kanałowy, czytnik 8-kanałowy, czytnik 16-kanałowy i czytnik 32-kanałowy, zintegrowane czytniki 9db obsługują tę funkcję.
Parametr	<i>Wartość:</i> 0-off, 1-on
Przykład	<code>int wartość = 1;</code> <code>czytnik. SetCheckAnt(wartość);</code>

4.7 Ustawianie czasów przepisywania

Definicja	<code>public abstract void SetRetryTimes(int wartość)</code>
Wyjaśnić	Liczba ponownych zapisów tagu, gdy zapis tagu nie powiedzie się, tylko czytnik 4-kanałowy, czytnik 8-kanałowy, czytnik 16-kanałowy i czytnik 32-kanałowy, zintegrowane czytniki 9db obsługują tę funkcję.
Parametr	<i>Wartość:</i> zakres wartości 0-7;
Przykład	<code>int wartość = 3;</code> <code>czytnik. SetRetryTimes(wartość);</code>

4.8 Ustawienia DRM

Definicja	<code>public abstract void SetDRM(bool enable)</code>
Wyjaśnić	Włącz lub wyłącz funkcję DRM, tylko czytnik 4-kanałowy, 8-kanałowy

	Czytnik, czytnik 16-kanalowy i czytnik 32-kanalowy, zintegrowane czytniki 9dB obsługują tę funkcję.
Parametr	<i>enable: false</i> - zamyka DRM, <i>true</i> - włącza DRM
Przykład	<i>czytnik. SetDRM(prawda);</i>

4.9 Ustawienie tłumienia odbiciowego

Definicja	<i>public abstract void SetReturnLoss(int wartość)</i>
Wyjaśnić	Ustaw próg strat powrotnych, tylko czytnik 4-kanalowy, czytnik 8-kanalowy, czytnik 16-kanalowy i czytnik 32-kanalowy, zintegrowane czytniki 9db obsługują tę funkcję.
Parametr	Wartość : zakres wartości 0 ~ 20;
Przykład	<i>int wartość = 20;</i> <i>czytnik. SetReturnLoss(wartość);</i>

4.10 Ustawienie mocy pojedynczej anteny

Definicja	<i>public abstract void SetPortReadPower(int[] antPower)</i>
Wyjaśnić	Moc każdej anteny jest ustawiana osobno, tylko czytnik 4-kanalowy, czytnik 8-kanalowy, czytnik 16-kanalowy i czytnik 32-kanalowy, zintegrowane czytniki 9db obsługują tę funkcję. Jeśli czytnik ma 4 kanały, długość tablicy wynosi 4; Długość tablicy 8 kanałów wynosi 8; Długość tablicy 16-kanalowej wynosi 16.
Parametr	<i>antPower</i> : Tablica mocy

Przykład	<i>int[] antPower = {30,30,30,28}; 4-kanalowy czytnik ustawień mocy . SetPortReadPower(antPower);</i>
----------	---

5. Akwizycja parametrów

5.1 Uzyskaj informacje o czytniku

Definicja	<i>abstrakt publiczny ReaderInfo GetReaderInfo()</i>
Wyjaśnić	Uzyskiwanie podstawowych informacji o czytniku
Wartość zwracana	<i>ReaderInfo</i> . Patrz 2.8, aby uzyskać szczegółowe informacje
Przykład	<i>ReaderInfo info = czytnik. GetReaderInfo()</i>

5.2 Uzyskaj numer seryjny czytnika

Definicja	<i>publiczny abstrakcyjny ciąg znaków GetSerialNO()</i>
Wyjaśnić	Uzyskaj numer seryjny czytnika
Wartość zwracana	Ciąg znaków
Przykład	<i>string serialNo = czytnik. GetSerialNO();</i>

5.3 Przejęcie straty ze strony stopy zwrotu

Definicja	<i>public abstract int GetReturnLoss()</i>
Wyjaśnić	Akwizycja progu strat powrotnych, tylko czytnik 4-kanalowy, czytnik 8-kanalowy, czytnik 16-kanalowy i czytnik 32-kanalowy, zintegrowane czytniki 9db obsługują tę funkcję.
Wartość zwracana	Int

Przykład	<i>int wartość = czytnik. Funkcja GetReturnLoss();</i>
----------	--

5.4 Akwizycja trybu odczytu

Definicja	<i>public abstract int GetReadMode()</i>
Wyjaśnić	Akwizycja trybu odczytu
Wartość zwracana	Tryb 0 odpowiedzi, 1 tryb czasu rzeczywistego
Przykład	<i>int mode = czytnik. Funkcja GetReadMode();</i>

5.5 Nabycie mocy zapisu

Definicja	<i>public abstract int GetWritePower()</i>
Wyjaśnić	uzyskać moc zapisu, tylko czytnik 4-kanalowy, czytnik 8-kanalowy, czytnik 16-kanalowy i czytnik 32-kanalowy, zintegrowane czytniki 9db obsługują tę funkcję.
Wartość zwracana	Int
Przykład	<i>int power = czytnik. GetWritePower();</i>

5.6 Pobieranie czasów zapisu

Definicja	<i>public abstract int GetRetryTimes()</i>
Wyjaśnić	uzyskać liczbę przeróbek, tylko czytnik 4-kanalowy, czytnik 8-kanalowy, czytnik 16-kanalowy i czytnik 32-kanalowy, zintegrowane czytniki 9db obsługują tę funkcję.
Wartość zwracana	Int
Przykład	<i>int razy = czytelnik. GetRetryTimes();</i>

5.7 Akwizycja parametrów trybu czasu rzeczywistego

Definicja	<i>abstrakt publiczny RealTimeParam GetRealTimeParam()</i>
Wyjaśnić	akwizycja parametrów trybu czasu rzeczywistego, tylko czytnik 4-kanalowy, czytnik 8-kanalowy, czytnik 16-kanalowy i czytnik 32-kanalowy, zintegrowane czytniki 9db obsługują tę funkcję.
Wartość zwracana	<i>RealTimeParam</i> : szczegółowe informacje można znaleźć w wersji 2.9
Przykład	<i>RealTimeParam param = czytnik. Funkcja GetRealTimeParam();</i>

5.8 Akwizycja mocy pojedynczej anteny

Definicja	<i>public abstract int[] GetPortReadPower()</i>
Wyjaśnić	pozyskiwanie mocy pojedynczej anteny, tylko czytnik 4-kanalowy, czytnik 8-kanalowy, czytnik 16-kanalowy i czytnik 32-kanalowy, zintegrowane czytniki 9db obsługują tę funkcję.
Wartość zwracana	Int
Przykład	<i>int[] power = czytnik. Funkcja GetPortReadPower();</i>