



AsReader ASR-P52N

C# SDK 開発マニュアル

修正履歴

No.	バージョン	修正内容	日付
1	1.0	新規作成	2022/11/8

目次

概要	8
1. 開発環境の構築	9
1.1. SDK の追加	9
1.1.1. Windows デスクトップアプリケーションの新規作成	9
1.1.2. Reference を追加	9
1.2. 名前空間を参照する	12
1.3. SDK の使用方法	12
1.3.1. AsReader を接続	12
1.3.2. タグをインベントリ	13
1.3.3. AsReader を切断	13
2. AsReader クラス	14
2.1. メソッド	14
2.1.1. StartSearchDevice	14
2.1.2. StopSearchDevice	14
2.1.3. ConnectServerWithRS232	15
2.1.4. ConnectServerWithRS485	15
2.1.5. ConnectWithTCP	16
2.1.6. Disconnect	17
2.1.7. SetDelegates	17
2.1.8. GetReaderInfo	18
2.1.9. GetBaseVersion	18

2.1.10.	SetSerialParam.....	19
2.1.11.	GetSerialParam	20
2.1.12.	SetEthernetIP	20
2.1.13.	GetEthernetIP	21
2.1.14.	GetEthernetMac	22
2.1.15.	SetTcpMode	22
2.1.16.	GetTcpMode.....	23
2.1.17.	SetGPO	24
2.1.18.	GetGpiState	24
2.1.19.	SetGpiTrigger	25
2.1.20.	GetGpiTrigger	26
2.1.21.	SetReaderTime	27
2.1.22.	GetReaderTime	27
2.1.23.	SetRs485	28
2.1.24.	GetRs485	29
2.1.25.	SetBeepOnOff.....	29
2.1.26.	GetTemperatureControl	30
2.1.27.	SetEas	30
2.1.28.	GetEas	32
2.1.29.	SetAntPower.....	32
2.1.30.	GetAntPower	33
2.1.31.	SetFreqRange.....	34
2.1.32.	GetFreqRange	35
2.1.33.	SetFrequency	35
2.1.34.	GetFrequency	36

2.1.35.	SetFilter	37
2.1.36.	GetFilter	38
2.1.37.	SetEPCBaseband	39
2.1.38.	GetEPCBaseband	39
2.1.39.	SetAutoFree	40
2.1.40.	GetAutoFree	41
2.1.41.	BaseRead	41
2.1.42.	BaseInventory	43
2.1.43.	BaseStop	44
2.1.44.	WriteTag	45
2.1.45.	LockTag	46
2.1.46.	KillTag	47
2.1.47.	RestoreFactorySettings	48
2.1.48.	Restart	48
3.	AsReaderDelegate クラス	50
3.1.	デリゲート	50
3.1.1.	WhenSearchedDevices	50
3.1.2.	WhenReceivedReaderInfo	50
3.1.3.	WhenReceivedBaseVersion	51
3.1.4.	WhenReceivedSetResult	51
3.1.5.	WhenReceivedSerialParam	52
3.1.6.	WhenReceivedEthernetIP	53
3.1.7.	WhenReceivedEthernetMac	53
3.1.8.	WhenReceivedTcpMode	54
3.1.9.	WhenReceivedGPIState	54

3.1.10.	WhenReceivedGPITriggerParams	55
3.1.11.	WhenReceivedReaderTime	55
3.1.12.	WhenReceivedRs485Params.....	56
3.1.13.	WhenReceivedTemperatureControl	56
3.1.14.	WhenReceivedEas	57
3.1.15.	WhenReceivedAntPower.....	58
3.1.16.	WhenReceivedFreqRange.....	58
3.1.17.	WhenReceivedFrequency	59
3.1.18.	WhenReceivedFilter.....	59
3.1.19.	WhenReceivedEPCBaseband.....	60
3.1.20.	WhenReceivedAutoFree.....	60
3.1.21.	WhenReceivedBaseStop	61
3.1.22.	WhenReceivedTag	61
3.1.23.	WhenReceivedMagnusS3Tag	62
3.1.24.	WhenReceivedMoistureTag.....	62
3.1.25.	WhenReceivedTagReadEnd	63
3.1.26.	WhenReceivedData	63
3.1.27.	WhenDisconnect	64
3.1.28.	WhenReceivedWorkTemperature	64
3.1.29.	WhenReceivedTriggerStart.....	65
3.1.30.	WhenReceivedTriggerStop	66
4.	Types クラス.....	67
4.1.	列挙.....	67
4.1.1.	EthernetIPType.....	67
4.1.2.	CSMode	67

4.1.3.	Baudrate	67
4.1.4.	GPOStatus.....	67
4.1.5.	GPITriggerStartConditions	68
4.1.6.	GPITriggerStopConditions	68
4.1.7.	GPITriggerPorts	68
4.1.8.	OnOff	69
4.1.9.	EasMatchArea	69
4.1.10.	CloseOpen.....	69
4.1.11.	FrequencyAutoFixed.....	70
4.1.12.	InventoryMode.....	70
4.1.13.	TagArea.....	70
4.1.14.	MatchArea.....	70
4.1.15.	TIDMode	71
4.1.16.	WriteTagArea	71
4.1.17.	InventoryType.....	71
4.1.18.	ReadResult.....	71
4.1.19.	ReadEndReasonCode	72
5.	構造体.....	73
5.1.	ReaderInfomations.....	73
5.2.	EthernetIP	73
5.3.	TcpMode	74
5.4.	Rs485Param.....	74
5.5.	GPISate	74
5.6.	GPITriggerParams	75
5.7.	GPITriggerParamsSetting	76

5.8.	EasAction.....	77
5.9.	EasInfo.....	77
5.10.	EPCBaseband.....	78
5.11.	AntPower.....	79
5.12.	ReaderTime	80
5.13.	GPO	80
5.14.	Filter.....	80
5.15.	AutoFree.....	81
5.16.	Frequency	81
5.17.	ReadParameters.....	81
5.18.	InventoryParameters.....	83
5.19.	Tag	84
5.20.	WriteParameters	85
5.21.	LockParameters.....	86
5.22.	KillParameters	87
5.23.	WorkTemperature	88
6.	付録.....	89

概要

本マニュアルは SDK を使用する Windows デスクトップアプリケーション開発者向けに以下の内容を提供します。

- 開発環境の構築方法。
- SDK ライブラリの各ファンクションの説明。

開発ツール：

- Visual Studio 2019

1. 開発環境の構築

1.1. SDK の追加

1.1.1. Windows デスクトップアプリケーションの新規作成

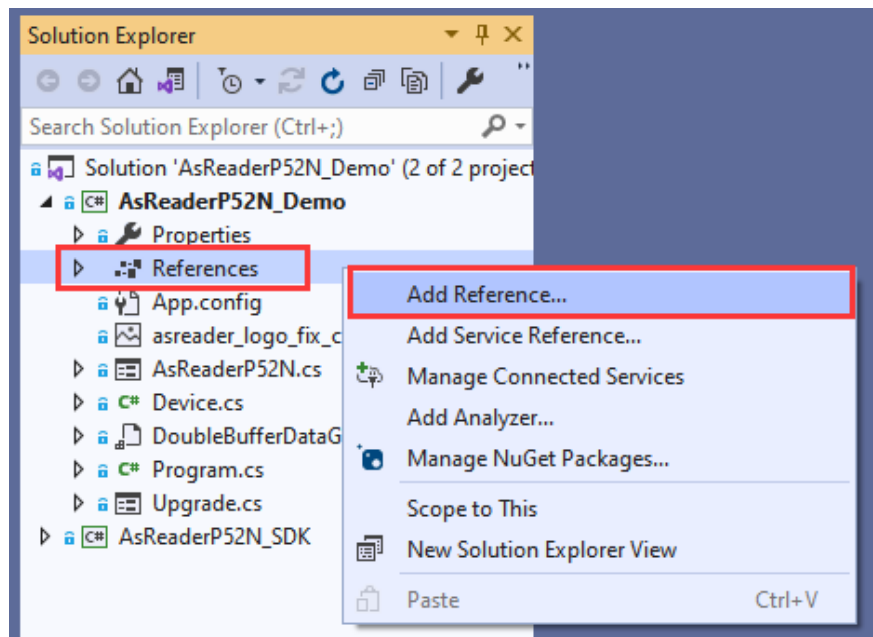
SDK をプロジェクトフォルダ内にコピーします。

開発・デバッグ時は、プログラムのカレントディレクトリの bin/Debug フォルダの下にコピーします。

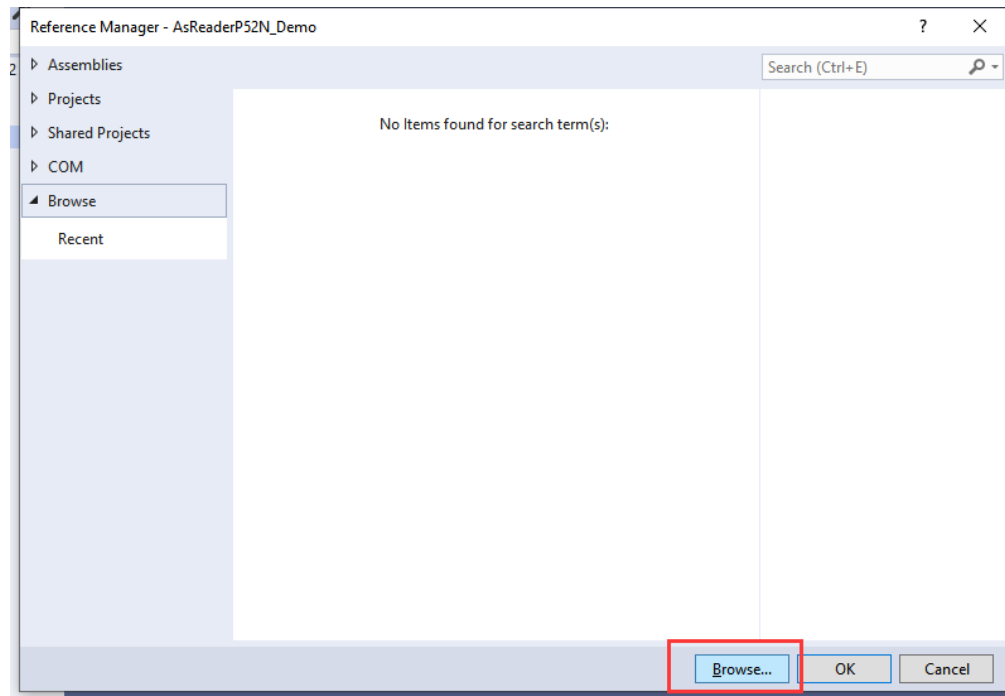
リリース時は、プログラムのカレントディレクトリの bin/Release フォルダの下にコピーします。

1.1.2. Reference を追加

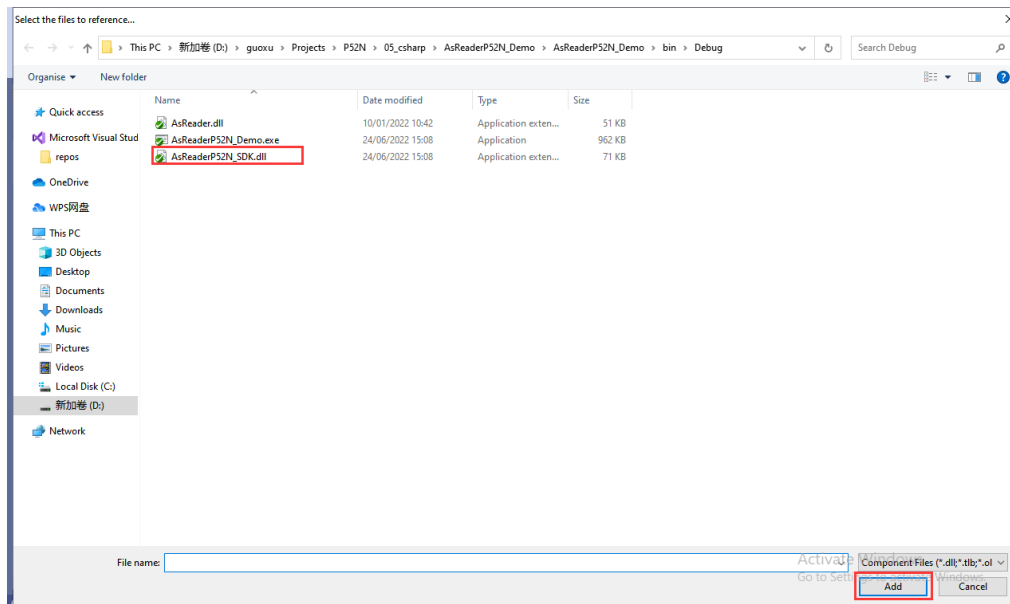
「References」を右クリックし、「Add Reference」を選択します。



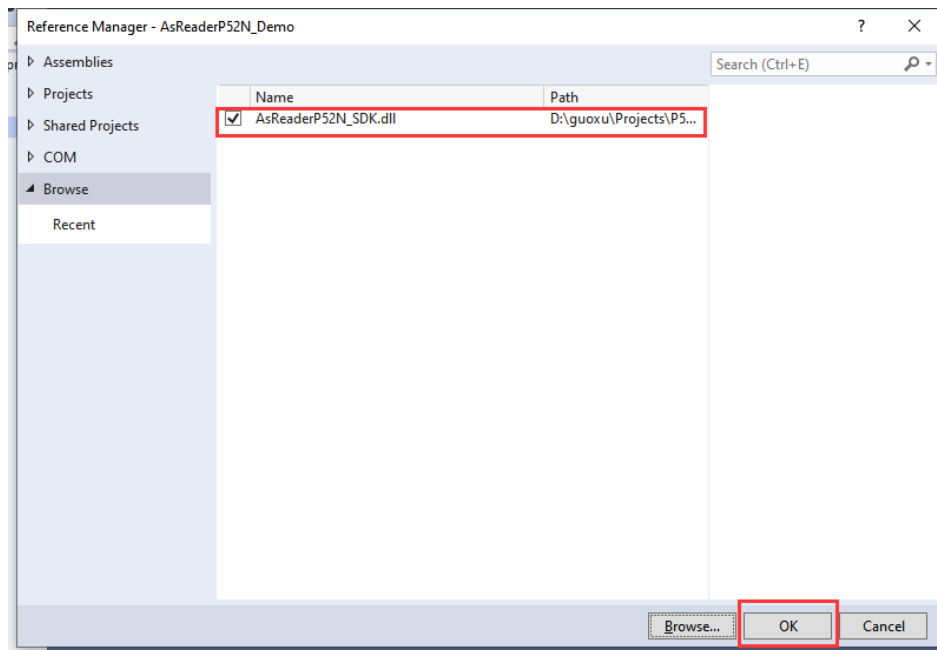
「Browse」ボタンをクリックします。



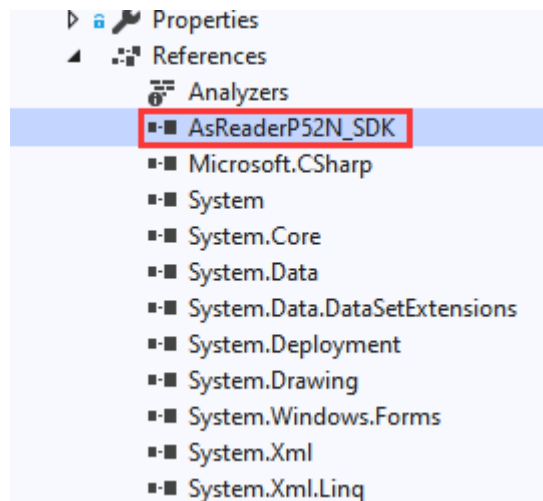
「Select the files reference」ダイアログボックスで、プロジェクトパスの下にある SDK ファイルを選択し、「Add」ボタンをクリックします。



Reference Manager 画面の「OK」ボタンをクリックすると、SDK ファイルが追加されます。



参照が正常に追加されると、参照リストに表示されます。



1.2. 名前空間を参照する

```
using AsReaderP52N;
```

1.3. SDK の使用方法

以下は AsReader クラスのメソッドを呼び出して、AsReader との接続または切断、タグの読み取り方法の使用例となります。

1.3.1. AsReader を接続

1. AsReader オブジェクトの準備

「AsReader」クラスのコンストラクタを呼び出して、AsReader のオブジェクトを取得します。（以下はサンプルコードです）

```
AsReader asreader = new AsReader();
```

2. AsReaderDelegate オブジェクトの準備

「AsReaderDelegate」クラスのデフォルトコンストラクタ関数を呼び出し、AsReaderDelegate オブジェクトを取得します。

```
AsReaderDelegate asReaderDelegate = new AsReaderDelegate();
```

3. RS232 方式で AsReader と接続

COM ポートとボーレートをパラメータにセットして asreader.ConnectServerWithRS232 メソッドを呼び出します。接続が成功すると、メソッドは 0 を返します。

```
int ret = asreader.ConnectServerWithRS232("COM1", 115200);  
if(ret == 0){  
    //接続成功の処理。  
}else{  
    //接続失敗の処理。  
}
```

1.3.2. タグをインベントリ

1. AsReaderDelegate オブジェクトのデリゲートを指定する

(1) パラメータが Tag タイプのメソッドを作成する

このメソッドは、AsReader がタグを読み取った時のコールバック処理に用いられます。パラメータ tag は tag タイプです。(5.21 参照)

```
private void ReceivedTag(Tag tag)
{
    //読み取ったタグデータを処理します。
}
```

(2) WhenReceivedTag デリゲートを指定する

1.3.2-1-(1)で作成したメソッドを AsReaderDelegate クラスの whenReceivedTag デリゲートに指定します。

```
asReaderDelegate.whenReceivedTag = ReceivedTag;
```

2. タグをインベントリ

AsReader クラスの「public UInt32 BaseInventory()」メソッドを呼び出して、タグのインベントリを開始します。タグをインベントリされると、1.3.2-1 (1) で作成されたメソッドがコールバックされ、読み取り結果が Tag タイプで渡されます。

```
//タグをインベントリ
UInt32 resultCode = asreader.BaseInventory();
```

1.3.3. AsReader を切断

AsReader クラスの「public UInt32 Disconnect()」メソッドを呼び出して、AsReader を切断します。切断が成功すると、メソッドは 0 を返します。

```
UInt32 ret = asReader.Disconnect();
if(ret == 0)
{
    //切断成功時の処理。
}else{
    //切断失敗時の処理。
}
```

2. AsReader クラス

2.1. メソッド

2.1.1. StartSearchDevice

関数名	void StartSearchDevice();			
引数	引数名	In/Out	型	説明
	-	-	-	-
戻り値	-	-	-	-
メソッドの説明： AsReader の検索を開始します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenSearchedDevices」(3.1.1 を参照) デリゲートがコールバックされ、検索された AsReader の IP 情報を返します。 呼び出し条件： なし サンプルコード： Device.StartSearchDevice();				

2.1.2. StopSearchDevice

関数名	void StopSearchDevice();			
引数	引数名	In/Out	型	説明
	-	-	-	-
戻り値	-	-	-	-
メソッドの説明： AsReader の検索を停止します。				

呼び出し条件：

なし

サンプルコード：

Device.StopSearchDevice();

2.1.3. ConnectServerWithRS232

関数名	UInt32 ConnectServerWithRS232(string comPort, int baudrate);			
引数	引数名	In/Out	型	説明
	comPort	In	string	COM ポート
	baudrate	In	int	ボーレート 9600、19200、115200、 230400、460800
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
メソッドの説明： RS232 を使用して AsReader に接続します。				
呼び出し条件： なし				
サンプルコード： //AsReader の接続（COM1 ポート、ボーレート 115200bps）。 UInt32 ret = Device.ConnectServerWithRS232("COM1", 115200);				

2.1.4. ConnectServerWithRS485

関数名	UInt32 ConnectServerWithRS485(string comPort, int baudrate, int address);			
引数	引数名	In/Out	型	説明
	comPort	In	string	COM ポート

	baudrate	In	int	ボーレート 9600、19200、115200、 230400、460800
	address	In	int	AsReader の RS485 デバイスア ドレス
返回值	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 : RS485 を使用して AsReader に接続します。</p> <p>呼び出し条件 : なし</p> <p>サンプルコード : //AsReader の接続 (COM1 ポート、ボーレート 115200bps, RS485 デバイスアドレス 1) 。 UInt32 ret = Device.ConnectServerWithRS485("COM1", 115200, 1);</p>				

2.1.5. ConnectWithTCP

関数名	UInt32 ConnectWithTCP(string ipAddress, int tcpPort);			
引数	引数名	In/Out	型	説明
	ipAddress	In	string	IP アドレス
	tcpPort	In	int	ポート
返回值	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 : TCP を使用して AsReader に接続します。</p> <p>呼び出し条件 : このメソッドは、AsReader の IP が検索できた後に呼び出すことができます。</p> <p>サンプルコード :</p>				

```
//IP アドレス 192.168.10.190 ポート 8160 に設定された AsReader に接続します。
UInt32 ret = Device.ConnectWithTCP("192.168.10.190", 8160);
```

2.1.6. Disconnect

関数名	UInt32 Disconnect();			
引数	引数名	In/Out	型	説明
	-	-	-	-
返回值	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 : AsReader との接続を切断します。</p> <p>呼び出し条件 : RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p> <p>サンプルコード : UInt32 ret = Device.Disconnect();</p>				

2.1.7. SetDelegates

関数名	void SetDelegates(AsReaderDelegate delegates);			
引数	引数名	In/Out	型	説明
	delegates	In	AsReaderDelegat e	AsReaderDelegate デリゲートオブジェクト (3 を参照)
返回值	-	-	-	-
<p>メソッドの説明 : AsReaderDelegate オブジェクトのデリゲート関数を設定します。</p> <p>呼び出し条件 : なし</p>				

サンプルコード：

```
Device.SetDelegates(delegates); // delegates は AsReaderDelegate クラスのインスタンスオブジェクトです。
```

2.1.8. GetReaderInfo

関数名	UInt32 GetReaderInfo();			
引数	引数名	In/Out	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0：成功 1：失敗

メソッドの説明：

AsReader の基本情報を取得します。

※基本情報には AsReader の製造番号、アプリバージョン、アプリのコンパイル時間、ベースバンドバージョン、システムバージョン、パワーオンタイムが含まれます。

このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedReaderInfo」（3.1.2 を参照）デリゲートがコールバックされ、AsReader の基本情報を返します。

呼び出し条件：

RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。

サンプルコード：

```
Device.GetReaderInfo();
```

2.1.9. GetBaseVersion

関数名	UInt32 GetBaseVersion();			
引数	引数名	In/Out	型	説明
	-	-	-	-

戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 : AsReader の Base バージョンを取得します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedBaseVersion」(3.1.3 を参照) デリゲートがコールバックされ、AsReader の Base バージョンを返します。</p> <p>呼び出し条件 : RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p> <p>サンプルコード : Device.GetBaseVersion();</p>				

2.1.10. SetSerialParam

関数名	UInt32 SetSerialParam(Types.Baudrate baudrate);			
引数	引数名	In/Out	型	説明
	baudrate	In	Types.Baudrate	シリアルモードの通信ボーレート 列挙型 Types.Baudrate (4.1.3 を参照)
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 : AsReader のシリアルモード通信のボーレートを設定します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedSetResult」(3.1.4 を参照) デリゲートがコールバックされ、設定した結果を返します。</p> <p>呼び出し条件 : RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p>				

サンプルコード：

```
Device.SetSerialParam(Types.Baudrate._115200); //通信ボーレートを 115200bps に設定。
```

2.1.11. GetSerialParam

関数名	UInt32 GetSerialParam();			
引数	引数名	In/Out	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0：成功 1：失敗
<p>メソッドの説明： AsReader のシリアルモードの通信ボーレートのパラメータを取得します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedSerialParam」（3.1.5 を参照）デリゲートがコールバックされ、AsReader のシリアルモードの通信ボーレートパラメータを返します。</p> <p>呼び出し条件： RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p> <p>サンプルコード： Device.GetSerialParam();</p>				

2.1.12. SetEthernetIP

関数名	UInt32 SetEthernetIP(EthernetIP ethernetIP);			
引数	引数名	In/Out	型	説明
	ethernetIP	In	EthernetIP	イーサネット IP パラメータ EthernetIP 型（5.2 を参照）
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0：成功 1：失敗
<p>メソッドの説明：</p>				

AsReader の IP アドレスのパラメータを設定します。

このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedSetResult」(3.1.4 を参照) デリゲートがコールバックされ、設定した結果を返します。

呼び出し条件：

RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。

サンプルコード：

```
EthernetIP ethernetIP = new EthernetIP()
{
    AutoOrStaticIP = (byte)Types.EthernetIPTypeAutomatic,
    IP = "192.168.1.100",
    Mask = "255.255.255.0",
    Gateway = "192.168.1.1",
    Dns1 = "0.0.0.0",
    Dns2 = "0.0.0.0",
};
Device.SetEthernetIP(ethernetIP);
```

2.1.13. GetEthernetIP

関数名	UInt32 GetEthernetIP();			
引数	引数名	In/Out	型	説明
	-	-	-	-
返回值	-	Out	UInt32	メソッド呼び出しの結果 0：成功 1：失敗

メソッドの説明：

AsReader の IP アドレスのパラメータを取得します。

このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedEthernetIP」(3.1.6 を参照) デリゲートがコールバックされ、AsReader の IP アドレスのパラメータを返します。

呼び出し条件：

RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができ

ます。

サンプルコード：

```
Device.GetEthernetIP();
```

2.1.14. GetEthernetMac

関数名	UInt32 GetEthernetMac();			
引数	引数名	In/Out	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0：成功 1：失敗

メソッドの説明：
AsReader の MAC アドレスを取得します。
このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedEthernetMac」（3.1.7 を参照）デリゲートがコールバックされ、AsReader の MAC アドレスを返します。

呼び出し条件：
RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。

サンプルコード：
Device.GetEthernetMac();

2.1.15. SetTcpMode

関数名	UInt32 SetTcpMode(TcpMode tcpMode);			
引数	引数名	In/Out	型	説明
	tcpMode	In	TcpMode	TCP モードのパラメータ TcpMode 型（5.3 を参照）
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0：成功

				1 : 失敗
<p>メソッドの説明 : AsReader の TCP モード及び関連のパラメータを設定します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedSetResult」(3.1.4 を参照) デリゲートがコールバックされ、設定した結果を返します。</p> <p>呼び出し条件 : RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p> <p>サンプルコード :</p> <pre>TcpMode tcpMode = new TcpMode() { CSMode = (byte)Types.CSMode,Server, ServerPort = 8160, }; Device.SetTcpMode(tcpMode);</pre>				

2.1.16. GetTcpMode

関数名	UInt32 GetTcpMode();			
引数	引数名	In/Out	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 : AsReader の TCP モード及び関連のパラメータを取得します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedTcpMode」(3.1.8 を参照) デリゲートがコールバックされ、AsReader の TCP モード Server 及び関連パラメータを返します。</p> <p>呼び出し条件 : RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p>				

サンプルコード：

```
Device.GetTcpMode();
```

2.1.17. SetGPO

関数名	UInt32 SetGPO(GPO gpo);			
引数	引数名	In/Out	型	説明
	gpo	In	GPO	GPIO 出力ポートの電圧レベル GPO 型 (5.13 を参照)
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明： AsReader の GPIO 出力ポート電圧レベルのパラメータを設定します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedSetResult」(3.1.4 を参照) デリゲートがコールバックされ、設定した結果を返します。</p> <p>呼び出し条件： RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p> <p>サンプルコード： GPO gpo = new GPO(); gpo.GPO1 = "0";//GPO1 の出力ポート電圧レベルを Low に設定 gpo.GPO2 = "1";//GPO2 の出力ポート電圧レベルを High に設定 Device.SetGPO(gpo);</p>				

2.1.18. GetGpiState

関数名	UInt32 GetGpiState();			
引数	引数名	In/Out	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	メソッド呼び出しの結果

				0 : 成功 1 : 失敗
<p>メソッドの説明 : AsReader の GPIO 入力ポート電圧レベルのステータスを取得します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedGPIState」(3.1.9を参照) デリゲートがコールバックされ、AsReader の GPIO 入力ポート電圧レベルのステータスを返します。</p> <p>呼び出し条件 : RS232、RS485、TCPのいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p> <p>サンプルコード : Device.GetGpiState();</p>				

2.1.19. SetGpiTrigger

関数名	UInt32 SetGpiTrigger(GPITriggerParamsSetting gpiTriggerParamsSetting);			
引数	引数名	In/Out	型	説明
	gpiTriggerParamsSetting	In	GPITriggerParamsSetting	GPI 入力ポートのトリガーパラメータ GPITriggerParamsSetting 型 (5.7を参照)
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 : AsReader の GPI 入力ポートのトリガーパラメータを設定します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedSetResult」(3.1.4を参照) デリゲートがコールバックされ、設定した結果を返します。</p> <p>呼び出し条件 : RS232、RS485、TCPのいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p>				

サンプルコード :

```

GPITriggerParamsSetting gpiTriggerParamsSetting = new GPITriggerParamsSetting()
{
    port = 0,
    TriggerStartCondition = 0,
    TriggerBindCommand = "",
    TriggerStopCondition = 0,
    DelayTime = 0,
    ReportOnOff = 0,
};
Device.SetGpiTrigger(gpiTriggerParamsSetting);

```

2.1.20. GetGpiTrigger

関数名	UInt32 GetGpiTrigger(Types.GPITriggerPorts triggerPort);			
引数	引数名	In/Out	型	説明
	triggerPort	In	Types.GPITriggerPorts	GPI 入力ポートのトリガーパラメータ 列挙型 Types.GPITriggerPorts (4.1.7 を参照)
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗

メソッドの説明 :

AsReader の指定した入力ポートの GPI トリガーパラメータを取得します。

このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedGPITriggerParams」(3.1.10 を参照) デリゲートがコールバックされ、AsReader の指定した入力ポートのトリガーパラメータを返します。

呼び出し条件 :

RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。

サンプルコード :

```
//GPI ポート 0 のトリガーパラメータを取得します。
Device.GetGpiTrigger((Types.GPITriggerPorts)0);
```

2.1.21. SetReaderTime

関数名	UInt32 SetReaderTime(ReaderTime time);			
引数	引数名	In/Out	型	説明
	time	In	ReaderTime	AsReader タイムパラメータ ReaderTime 型 (5.12 を参照)
返回值	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 : AsReader のシステムタイムを設定します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedSetResult」(3.1.4 を参照) デリゲートがコールバックされ、設定した結果を返します。</p> <p>呼び出し条件 : RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p> <p>サンプルコード : <pre>ReaderTime time = new ReaderTime(); Time.UTCSecond = "2022.01.01 12:00:00", Device.SetReaderTime(time); //設定するシステム時間 2022.01.01 12:00:00。</pre> </p>				

2.1.22. GetReaderTime

関数名	UInt32 GetReaderTime();			
引数	引数名	In/Out	型	説明
	-	-	-	-
返回值	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功

				1 : 失敗
<p>メソッドの説明 : AsReader のシステムタイムを取得します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedReaderTime」(3.1.11を参照) デリゲートがコールバックされ、AsReader のシステムタイムを返します。</p> <p>呼び出し条件 : RS232、RS485、TCPのいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p> <p>サンプルコード : Device.GetReaderTime();</p>				

2.1.23. SetRs485

関数名	UInt32 SetRs485(Types.Baudrate baudrate, int address);			
引数	引数名	In/Out	型	説明
	baudrate	In	Types.Baudrate	通信ボーレート 列挙型 Types.Baudrate (4.1.3を参照)
	address	In	int	AsReader の RS485 のアドレス
返回值	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 : AsReader の RS485 の通信パラメータ (ボーレート、アドレス) を設定します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedSetResult」(3.1.4を参照) デリゲートがコールバックされ、設定した結果を返します。</p> <p>呼び出し条件 : RS232、RS485、TCPのいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p> <p>サンプルコード :</p>				

```
Device.SetRs485(Types.Baudrate._115200, 1);
```

2.1.24. GetRs485

関数名	UInt32 GetRs485();			
引数	引数名	In/Out	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 : AsReader の RS485 の通信パラメータ（ボーレート、アドレス）を取得します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedRs485Params」（3.1.12 を参照）デリゲートがコールバックされ、AsReader の通信パラメータを返します。</p> <p>呼び出し条件 : RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p> <p>サンプルコード : Device.GetRs485();</p>				

2.1.25. SetBeepOnOff

関数名	UInt32 SetBeepOnOff(int onOff);			
引数	引数名	In/Out	型	説明
	onOff	In	int	ブザーの制御 0 : ブザーオン 1 : ブザーオフ
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 :</p>				

AsReader のビーブ音をオン/オフします。

このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedSetResult」(3.1.4 を参照) デリゲートがコールバックされ、設定した結果を返します。

呼び出し条件：

RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。

サンプルコード：

```
Device.SetBeepOnOff(0); //ブザーオンに設定
```

2.1.26. GetTemperatureControl

関数名	UInt32 GetTemperatureControl();			
引数	引数名	In/Out	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗

メソッドの説明：

AsReader の温度制御パラメータを取得します。

このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedTemperatureControl」(3.1.13 を参照) デリゲートがコールバックされ、AsReader の温度制御パラメータを返します。

呼び出し条件：

RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。

サンプルコード：

```
Device.GetTemperatureControl();
```

2.1.27. SetEas

関数名	UInt32 SetEas(EasInfo info);
-----	------------------------------

引数	引数名	In/Out	型	説明
	info	In	EasInfo	アラートパラメータのマッチング EasInfo 型 (5.9 を参照)
返り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗

メソッドの説明 :

AsReader のアラートパラメータのマッチングを設定します。マッチング条件に満たしたタグを読み取ると、「マッチング成功アクション」を実行し、マッチング条件に満たさないタグを読み取ると、「マッチング失敗アクション」を実行します。

このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedSetResult」(3.1.4 を参照) デリゲートがコールバックされ、設定した結果を返します。

呼び出し条件 :

RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。

サンプルコード :

```
//アラートパラメータをマッチングする
EasInfo info = new EasInfo()
{
    AlarmOnOff = 1, //アラート機能をオン
    MatchArea = 0, //EPC エリア
    MatchStart = 32, //マッチングしようデータのスタートアドレス
    MatchData = "1234", //マッチングしようデータの内容
    Mask = "FFFF", //マスク
};
//マッチング成功アクションのパラメータ
info.EASMatchedAction = new EasAction()
{
    KeepTime = 2, //アクションのキープタイム (秒)
    GPO1 = 1, //デフォルト電圧レベル Low、出力電圧レベル High
    GPO2 = null,
    GPO3 = null,
    GPO4 = null,
};
```



```
//マッチング失敗アクションのパラメータ
info.EASUnMatchedAction = new EasAction()
{
    KeepTime = 2,//アクションのキープタイム (秒)
    GPO1 = null,
    GPO2 = 0,// デフォルト電圧レベル High、出力電圧レベル Low
    GPO3 = null,
    GPO4 = null,
};
Device.SetEas(info);
```

2.1.28. GetEas

関数名	UInt32 GetEas();			
引数	引数名	In/Out	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 : AsReader のマッチングのアラートパラメータを取得します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedEas」(3.1.14を参照) デリゲートがコールバックされ、AsReader のマッチングのアラートパラメータを返します。</p> <p>呼び出し条件 : RS232、RS485、TCPのいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p> <p>サンプルコード : Device.GetEas();</p>				

2.1.29. SetAntPower

関数名	UInt32 SetAntPower(AntPower antPower, bool powerDownSave);
-----	--

引数	引数名	In/Out	型	説明
	antPower	In	AntPower	アンテナパワーのパラメータ AntPower 型 (5.11 を参照)
	powerDownSave	In	bool	パワーダウンパラメータ保存 True : 電源切れパラメータ保存 False : 電源切れパラメータ保存しない
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 : AsReader の各アンテナポートのパワーを設定します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedSetResult」(3.1.4 を参照) デリゲートがコールバックされ、設定した結果を返します。</p> <p>呼び出し条件 : RS232、RS485、TCPのいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p> <p>サンプルコード : <pre>AntPower antPower = new AntPower(); antPower.Ant1 = 30;//アンテナ 1 パワー30dBm antPower.Ant2 = 24;//アンテナ 2 パワー24dBm Device.SetAntPower(antPower, true);</pre> </p>				

2.1.30. GetAntPower

関数名	UInt32 GetAntPower();			
引数	引数名	In/Out	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 :</p>				

AsReader の各アンテナのパワーを取得します。

このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedAntPower」(3.1.15 を参照) デリゲートがコールバックされ、AsReader の各アンテナのパワー値を返します。

呼び出し条件 :

RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。

サンプルコード :

```
Device.GetAntPower();
```

2.1.31. SetFreqRange

関数名	UInt32 SetFreqRange(int freqRangeIndex);			
引数	引数名	In/Out	型	説明
	freqRangeIndex	In	int	RF 周波数帯域 0: GB, 920MHz~925MHz 3: FCC, 902MHz~928MHz 4: ETSI, 865MHz~868MHz 10: JP, 916.8MHz~920.8MHz
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗

メソッドの説明 :

AsReader の稼働周波数帯域を設定します。

このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedSetResult」(3.1.4 を参照) デリゲートがコールバックされ、設定した結果を返します。

呼び出し条件 :

RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。

サンプルコード :

```
Device.SetFreqRange(10); //RF の周波数帯域を JP に設定する。916.8MHz~920.8MHz
```

2.1.32. GetFreqRange

関数名	UInt32 GetFreqRange();			
引数	引数名	In/Out	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 : AsReader の稼働周波数帯域を取得します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedFreqRange」(3.1.16 を参照) デリゲートがコールバックされ、AsReader の稼働周波数帯域値を返します。</p> <p>呼び出し条件 : RS232、RS485、TCPのいずれかの方式でAsReaderと接続している場合、このメソッドを呼び出すことができます。</p> <p>サンプルコード : Device.GetFreqRange();</p>				

2.1.33. SetFrequency

関数名	UInt32 SetFrequency(Types.FrequencyAutoFixed frequencyAutoFixed, byte[] pointList, bool powerDownSave);			
引数	引数名	In/Out	型	説明
	frequencyAutoFixed	In	Types.FrequencyAutoFixed	周波数の自動設定 0 : AsReader は指定された周波数リストの周波数を利用する 1 : AsReader が RF 周波数帯域以内で自動的に周波数を選択する 列挙型 Types.FrequencyAuto

				Fixed (4.1.11 を参照)
	pointList	In	byte[]	周波数リスト 周波数を任意で設定した い場合に稼働周波数を指 定する 周波数リストは稼働帯域 のチャンネル番号リストにな る 周波数範囲は 1～ 50
	powerDownSave	In	bool	パラメータ保存 0 : 一時的にパラメータ 保存 1 : 恒久的にパラメータを 保存
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗

メソッドの説明 :

AsReader の稼働周波数を設定します。

このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedSetResult」(3.1.4 を参照) デリゲートがコールバックされ、設定した結果を返します。

呼び出し条件 :

RS232、RS485、TCPのいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。

サンプルコード :

```
byte[] pointList = new byte[] {0}; // 周波数リストの一番目選択肢を選択する
Device.SetFrequency(Types.FrequencyAutoFixed.Fixed_frequency, pointList, true);
```

2.1.34. GetFrequency

関数名	UInt32 GetFrequency();			
引数	引数名	In/Ou	型	説明

		t		
	-	-	-	-
返り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 : AsReader の稼働周波数を取得します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedFrequency」(3.1.17 を参照) デリゲートがコールバックされ、AsReader の稼働周波数を返します。</p> <p>呼び出し条件 : RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p> <p>サンプルコード : Device.GetFrequency();</p>				

2.1.35. SetFilter

関数名	UInt32 SetFilter(ushort repeatTime, int rssiMax);			
引数	引数名	In/Out	型	説明
	repeatTime	In	ushort	同一タグの読み取り間隔 0~65535 タイム単位 : 10ms 設定した秒数ごとに同じタグの読み取り結果が通知される
	rssiMax	In	int	RSSI 閾値 0~110 タグの RSSI 値が閾値より低い場合、タグデータが廃棄される
返り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗

メソッドの説明：

AsReader がタグを読み取った結果をフィルタリングするルールを設定します。

このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedSetResult」(3.1.4 を参照) デリゲートがコールバックされ、設定した結果を返します。

呼び出し条件：

RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。

サンプルコード：

```
Device.SetFilter(0, -30);
```

2.1.36. GetFilter

関数名	UInt32 GetFilter();			
引数	引数名	In/Out	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗

メソッドの説明：

設定した AsReader の読取結果をフィルターするパラメータを取得します。

このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedFilter」(3.1.18 を参照) デリゲートがコールバックされ、AsReader がフィルタリングの設定パラメータを返します。

呼び出し条件：

RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。

サンプルコード：

```
Device.GetFilter();
```

2.1.37. SetEPCBaseband

関数名	UInt32 SetEPCBaseband(EPCBaseband epcBaseband);			
引数	引数名	In/Out	型	説明
	epcBaseband	In	EPCBaseband	ベースバンドパラメータ EPC ベースバンド型 (5.10 を参照)
返り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 : AsReader が使用する Baseband パラメータを設定します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedSetResult」(3.1.4 を参照) デリゲートがコールバックされ、設定した結果を返します。</p> <p>呼び出し条件 : RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p> <p>サンプルコード :</p> <pre>EPCBaseband epcBaseband = new EPCBaseband() { EPCSpeed = 1, //Dense mode. Session = 0, //Session0 QValue = 0, //デフォルト Q 値 SearchType = 2, // Flag A と Flag B を交代でインベントリ }; Device.SetEPCBaseband(epcBaseband);</pre>				

2.1.38. GetEPCBaseband

関数名	UInt32 GetEPCBaseband();			
引数	引数名	In/Out	型	説明
	-	-	-	-

戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 : AsReader が使用するベースバンドのパラメータを取得します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedEPCBaseband」(3.1.19 を参照) デリゲートがコールバックされ、AsReader が使用する Baseband パラメータを返します。</p> <p>呼び出し条件 : RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p> <p>サンプルコード : Device.GetEPCBaseband();</p>				

2.1.39. SetAutoFree

関数名	UInt32 SetAutoFree(AutoFree autoFree);			
引数	引数名	In/Out	型	説明
	autoFree	In	AutoFree	RF タグ読取モードパラメータ AutoFree 型 (5.15 を参照)
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 : AsReader の RF タグ読取モードの読取間隔を設定します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedSetResult」(3.1.4 を参照) デリゲートがコールバックされ、設定した結果を返します。</p> <p>呼び出し条件 : RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p> <p>サンプルコード :</p>				

```

AutoFree autoFree = new AutoFree()
{
    OpenClose = 1, // 1 : タグ読取モードの読取間隔を有効。
    FreeTime = 2, // アイドルタイム、単位 : 10ms。
};
Device.SetAutoFree(autoFree);

```

2.1.40. GetAutoFree

関数名	UInt32 GetAutoFree();			
引数	引数名	In/Out	型	説明
	-	-	-	-
返回值	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 : AsReader の RF タグ読取モードの読取間隔を取得します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedAutoFree」(3.1. 20 を参照) デリゲートがコールバックされ、AsReader の RF タグ読取モードの読取間隔を返します。</p> <p>呼び出し条件 : RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p> <p>サンプルコード : Device.GetAutoFree();</p>				

2.1.41. BaseRead

関数名	UInt32 BaseRead(ReadParameters readParameters);			
引数	引数名	In/Out	型	説明
	readParameters	In	ReadParameters	タグのパラメータを読取 ReadParameters 型 (5.17)

				を参照)
返り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗

メソッドの説明 :

AsReader の読取パラメータを設定して、タグの読取を開始します。

- 1) このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedSetResult」(3.1.4 を参照) デリゲートがコールバックされ、設定した結果を返します。
- 2) このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedTag」(3.1.22 を参照) デリゲートがコールバックされ、タグデータを返します。
- 3) このメソッドを呼び出した後、AsReader の温度が設定した温度の閾値を超えた場合、AsReaderDelegate クラスの「WhenReceivedWorkTemperature」(3.1.28 を参照) デリゲートがコールバックされ、AsReader の現在の稼働温度を返します。

呼び出し条件 :

RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。

サンプルコード :

```
ReadParameters readEpc = new ReadParameters()
{
    Ant1 = true, //アンテナ 1 を有効にする
    Ant2 = false,
    Ant3 = false,
    Ant4 = false,
    Ant5 = false,
    Ant6 = false,
    Ant7 = false,
    Ant8 = false,
    ReadTagArea = Types.TagArea.EPC, //EPC エリアを読取
    StartAddress = 2, //読み取りエリアのスタートアドレスは 2word
    ReadDataLength = 2, //読み取りデータの長さは 2word
    Password = "00000000", //タグのアクセスパスワード
    MatchArea = Types.MatchArea.EPC, //EPC エリアとマッチング
    MatchStartAddress = 2, //マッチングする対象データのスタートアドレスは 2word
    MatchData = "1111222233334444", //マッチングのデータ
};
```

```
Device.BaseRead(readEpc);
```

2.1.42. BaseInventory

関数名	UInt32 BaseInventory(InventoryParameters inventoryParameters);			
引数	引数名	In/Out	型	説明
	inventoryParameters	In	InventoryParameters	タグのパラメータをインベントリ InventoryParameters 型 (5.18 を参照)
返回值	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗

メソッドの説明 :

AsReader のインベントリのパラメータを設定して、タグデータのインベントリを開始します。

- 1) このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedSetResult」(3.1.4 を参照) デリゲートがコールバックされ、設定した結果を返します。
- 2) このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedTag」(3.1.22 を参照) デリゲートがコールバックされ、タグデータを返します。
- 3) このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedMagnusS3Tag」(3.1.23 を参照) デリゲートがコールバックされ、温度タグデータを返します。
- 4) このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedMoistureTag」(3.1.24 を参照) デリゲートがコールバックされ、湿度タグデータを返します。
- 5) このメソッドを呼び出した後、AsReader の温度が設定した温度の閾値を超えた場合、AsReaderDelegate クラスの「WhenReceivedWorkTemperature」(3.1.28 を参照) デリゲートがコールバックされ、AsReader の現在の稼働温度を返します。

呼び出し条件 :

RS232、RS485、TCPのいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。

サンプルコード :

```
InventoryParameters inventoryParameters= new InventoryParameters()
{
    Ant1 = true, //アンテナ 1 を有効にする
```

```

Ant2 = false,
Ant3 = false,
Ant4 = false,
Ant5 = false,
Ant6 = false,
Ant7 = false,
Ant8 = false,
SingleOrRepeat = Types.InventoryMode.Repeat,//連続インベントリ
InventoryType= Types.InventoryType.EPC,//EPC タグのインベントリ
ReadTIDArea= true,//TID エリアのデータをインベントリ
ReadUserArea= true,//ユーザーエリアのデータをインベントリ
ReadReservedArea= true,// Reserved エリアのデータをインベントリ
ReadEPCBank= true,//EPCBank エリアのデータをインベントリ
};
Device.BaseInventory(inventoryParameters);

```

2.1.43. BaseStop

関数名	UInt32 BaseStop();			
引数	引数名	In/Out	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗

メソッドの説明 :

AsReader のタグの読取を停止します。

このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedBaseStop」(3.1. 21 を参照) デリゲートがコールバックされ、AsReader はタグを読取停止の結果を返します。

呼び出し条件 :

RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。

サンプルコード :

```
Device.BaseStop();
```

2.1.44. WriteTag

関数名	UInt32 WriteTag(WriteParameters writeParameters);			
引数	引数名	In/Out	型	説明
	writeParameters	In	WriteParameters	タグの書き込みのパラメータ WriteParameters 型 (5.20 を参照)
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗

メソッドの説明 :

AsReader のタグを書き込みを実行します

このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedSetResult」(3.1.4 を参照) デリゲートがコールバックされ、タグの書き込み結果を返します。

呼び出し条件 :

RS232、RS485、TCPのいずれかの方式でAsReaderと接続している場合、このメソッドを呼び出すことができます。

サンプルコード :

```
WriteParameters writeEPC = new WriteParameters()
{
    Ant1 = true, //アンテナ 1 を有効にする
    Ant2 = false,
    Ant3 = false,
    Ant4 = false,
    Ant5 = false,
    Ant6 = false,
    Ant7 = false,
    Ant8 = false,
    DataArea = Types.WriteTagArea.EPC, //タグの EPC エリアに書き込み
    WordStartAddress = 2, //書き込み対象エリアのスタートアドレス
}
```

```

WriteData = "22222222",//書き込みデータ
Password = "00000000",//タグのアクセスパスワード
MatchArea = Types.MatchArea.EPC,//EPC エリアとマッチング
MatchStartAddress = 2,//マッチングする対象データのスタートアドレスは 2word
MatchData = "11111111">//マッチングデータ
};
Device.WriteTag(writeEPC);

```

2.1.45. LockTag

関数名	UInt32 LockTag(LockParameters lockParameters);			
引数	引数名	In/Out	型	説明
	lockParameters	In	LockParameters	タグのロック/アンロックのパラメータ LockParameters 型 (5.21を参照)
返回值	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗

メソッドの説明 :

AsReader のタグのロック/アンロックを実行します。

このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedSetResult」(3.1.4を参照) デリゲートがコールバックされ、タグのロック/アンロック結果を返します。

呼び出し条件 :

RS232、RS485、TCPのいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。

サンプルコード :

```

WriteParameters writeEPC = new WriteParameters()
{
    Ant1 = true,//アンテナ 1 を有効にする
    Ant2 = false,
    Ant3 = false,
    Ant4 = false,
    Ant5 = false,

```

```

    Ant6 = false,
    Ant7 = false,
    Ant8 = false,
    LockArea= 2,//ロックエリア : EPC
    LockType= 1,//ロックタイプ : ロック
    Password = "22222222",//タグのアクセスパスワード
    MatchArea = Types.MatchArea.EPC,//EPC エリアとマッチング
    MatchStartAddress = 2,//マッチングする対象データのスタートアドレスは 2word
    MatchData = "11111111">//マッチングデータ
};
Device.LockTag(lockParameters);

```

2.1.46. KillTag

関数名	UInt32 KillTag(KillParameters killParameters);			
引数	引数名	In/Out	型	説明
	killParameters	In	KillParameters	タグのキルのパラメータ KillParameters 型 (5.22 を参照)
返回值	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 : AsReader のタグのキルを実行します。 このメソッドを呼び出した後、AsReaderDelegate クラスの「WhenReceivedSetResult」(3.1.4 を参照) デリゲートがコールバックされ、タグのキル結果を返します。</p> <p>呼び出し条件 : RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p> <p>サンプルコード : KillParameters killParameters = new KillParameters() { Ant1 = true,//アンテナ 1 を有効にする</p>				


```

    Ant2 = false,
    Ant3 = false,
    Ant4 = false,
    Ant5 = false,
    Ant6 = false,
    Ant7 = false,
    Ant8 = false,
    Password = "11111111",//キルパスワード
    MatchArea = Types.MatchArea.EPC,//EPC エリアとマッチング
    MatchStartAddress = 2,//マッチングする対象データのスタートアドレスは 2word
    MatchData = "11111111">//マッチングデータ
};
Device.KillTag(killParameters);

```

2.1.47. RestoreFactorySettings

関数名	UInt32 RestoreFactorySettings();			
引数	引数名	In/Out	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
<p>メソッドの説明 : AsReader を工場出荷時の設定に戻します。</p> <p>呼び出し条件 : RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。</p> <p>サンプルコード : Device.RestoreFactorySettings();</p>				

2.1.48. Restart

関数名	UInt32 Restart();
-----	-------------------

引数	引数名	In/Out	型	説明
	-	-	-	-
返回值	-	Out	UInt32	メソッド呼び出しの結果 0 : 成功 1 : 失敗
メソッドの説明 : AsReader を再起動します。				
呼び出し条件 : RS232、RS485、TCP のいずれかの方式で AsReader と接続している場合、このメソッドを呼び出すことができます。				
サンプルコード : Device.Restart();				

3. AsReaderDelegate クラス

3.1. デリゲート

3.1.1. WhenSearchedDevices

デリゲート名	delegate void WhenSearchedDevices(string ip);			
引数	引数名	In/Out	型	説明
	ip	In	string	AsReader のネットワーク IP アドレス
戻り値	-	-	-	-
<p>デリゲートの説明 : メソッド StartSearchDevice (2.1.1 を参照) を実行後に、本デリゲートをコールバックします。</p> <p>サンプルコード : <pre>WhenSearchedDevices whenSearchedDevices = SearchedDevices; void SearchedDevices(string ip){ //検索された AsReader の IP アドレスを受け取ります。 } whenSearchedDevices(ip);</pre></p>				

3.1.2. WhenReceivedReaderInfo

デリゲート名	delegate void WhenReceivedReaderInfo(ReaderInfomations readerInfo);			
引数	引数名	In/Out	型	説明
	readerInfo	In	ReaderInfomations	AsReader の基本情報 ReaderInfomations 型 (5.1 を参照)
戻り値	-	-	-	-
<p>デリゲートの説明 : メソッド GetReaderInfo (2.1.6 を参照) を実行後に、本デリゲートをコールバックします。</p>				

サンプルコード：

```

WhenReceivedReaderInfo whenReceivedReaderInfo = ReceivedReaderInfo;
void ReceivedReaderInfo(ReaderInfomations readerInfo){
    //AsReader の基本情報を受け取ります。
}
whenReceivedReaderInfo(readerInfo);

```

3.1.3. WhenReceivedBaseVersion

デリゲート名	delegate void WhenReceivedBaseVersion(string version);			
引数	引数名	In/Out	型	説明
	version	In	string	AsReader の Base バージョン
返り値	-	-	-	-
デリゲートの説明：				
メソッド GetBaseVersion（2.1.7 を参照）を実行後に、本デリゲートをコールバックします。				
サンプルコード：				
<pre> WhenReceivedBaseVersion whenReceivedBaseVersion = ReceivedBaseVersion; void ReceivedBaseVersion(string version){ //AsReader の Base バージョンを受け取ります。 } whenReceivedBaseVersion(version); </pre>				

3.1.4. WhenReceivedSetResult

デリゲート名	delegate void WhenReceivedSetResult(string setItemName, int resultCode, string ErrMsg);			
引数	引数名	In/Out	型	説明
	setItemName	In	string	設定の項目名
	resultCode	In	int	設定結果
	ErrMsg	In	string	設定結果の説明
返り値	-	-	-	-

デリゲートの説明：

メソッド SetSerialParam (2.1.10 を参照)、SetEthernetIP (2.1.12 を参照)、SetTcpMode (2.1.15 を参照)、SetGPO (2.1.17 を参照)、SetGpiTrigger (2.1.19 を参照)、SetReaderTime (2.1.21 を参照)、SetRs485 (2.1.23 を参照)、SetBeepOnOff (2.1.25 を参照)、SetBeep (2.1.26 を参照)、SetEas (2.1.27 を参照)、SetAntPower (2.1.29 を参照)、SetFreqRange (2.1.31 を参照)、SetFrequency (2.1.33 を参照)、SetFilter (2.1.35 を参照)、SetEPCBaseband (2.1.37 を参照)、SetAutoFree (2.1.39 を参照)、BaseRead (2.1.41 を参照)、BaseInventory (2.1.42 を参照)、WriteTag (2.1.44 を参照)、LockTag (2.1.45 を参照)、KillTag (2.1.46 を参照) のいずれを実行後に、本デリゲートをコールバックします。

サンプルコード：

```
WhenReceivedSetResult whenReceivedSetResult = ReceivedSetResult;
void ReceivedSetResult(string setItemName, int resultCode, string ErrMsg){
    //設定した結果を受け取ります。
}
whenReceivedSetResult(setItemName, resultCode, ErrMsg);
```

3.1.5. WhenReceivedSerialParam

デリゲート名	delegate void WhenReceivedSerialParam(int serialParam);			
引数	引数名	In/Out	型	説明
	serialParam	In	int	AsReader シリアルモード通信のボーレート 0:9600 bps, 1:19200 bps, 2:115200 bps, 3:230400 bps, 4:460800 bps
返り値	-	-	-	-

デリゲートの説明：

メソッド GetSerialParam (2.1.9 を参照) を実行後に、本デリゲートをコールバックします。

サンプルコード：

```
WhenReceivedSerialParam whenReceivedSerialParam= ReceivedSerialParam;
void ReceivedSerialParam(int serialParam){
    //AsReader シリアルモード通信ボーレートを受け取ります。
}
}
```

```
whenReceivedSerialParam(serialParam);
```

3.1.6. WhenReceivedEthernetIP

デリゲート名	delegate void WhenReceivedEthernetIP(EthernetIP ethernetIP);			
引数	引数名	In/Out	型	説明
	ethernetIP	In	EthernetIP	AsReader の IP アドレスパラメータ EthernetIP 型 (5.2 を参照)
戻り値	-	-	-	-
<p>デリゲートの説明 : メソッド GetEthernetIP (2.1.11 を参照) を実行後に、本デリゲートをコールバックします。</p> <p>サンプルコード : <pre>WhenReceivedEthernetIP whenReceivedEthernetIP= ReceivedEthernetIP; void ReceivedEthernetIP(EthernetIP ethernetIP) { //AsReader の IP アドレスパラメータを受け取ります。 } whenReceivedEthernetIP(ethernetIP);</pre></p>				

3.1.7. WhenReceivedEthernetMac

デリゲート名	delegate void WhenReceivedEthernetMac(string ethernetMac);			
引数	引数名	In/Out	型	説明
	ethernetMac	In	string	AsReader の MAC アドレス (フォーマットは標準の MAC アドレスになる)
戻り値	-	-	-	-
<p>デリゲートの説明 : メソッド GetEthernetMac (2.1.12 を参照) を実行後に、本デリゲートをコールバックします。</p> <p>サンプルコード : <pre>WhenReceivedEthernetMac whenReceivedEthernetMac = ReceivedEthernetMac; void ReceivedEthernetMac(string ethernetMac)</pre></p>				

```
{
    //AsReader の MAC アドレスを受け取ります。
}
whenReceivedEthernetMac(ethernetMac);
```

3.1.8. WhenReceivedTcpMode

デリゲート名	delegate void WhenReceivedTcpMode(TcpMode tcpMode);			
引数	引数名	In/Out	型	説明
	tcpMode	In	TcpMode	AsReader の TCP モード及び関連のパラメータ TcpMode 型 (5.3 を参照)
戻り値	-	-	-	-
<p>デリゲートの説明： メソッド GetTcpMode (2.1.16 を参照) を実行後に、本デリゲートをコールバックします。</p> <p>サンプルコード： <pre>WhenReceivedTcpMode whenReceivedTcpMode = ReceivedTcpMode; void ReceivedTcpMode(TcpMode tcpMode){ // AsReader の TCP モード及び関連のパラメータを受け取ります。 } whenReceivedTcpMode (tcpMode);</pre></p>				

3.1.9. WhenReceivedGPIState

デリゲート名	delegate void WhenReceivedGPIState(GPIState gpiState);			
引数	引数名	In/Out	型	説明
	gpiState	In	GPIState	AsReader の GPIO 入力ポート電圧レベルのステータス GPIState 型 (5.5 を参照)
戻り値	-	-	-	-
<p>デリゲートの説明： メソッド GetGpiState (2.1.18 を参照) を実行後に、本デリゲートをコールバックします。</p>				

サンプルコード :

```

WhenReceivedGPIState whenReceivedGPIState = ReceivedGPIState;
void ReceivedGPIState(GPIState gpiState){
    //AsReader の GPIO 入力ポート電圧レベルのステータスを受け取ります。
}
whenReceivedGPIState(gpiState);

```

3.1.10. WhenReceivedGPITriggerParams

デリゲート名	delegate void WhenReceivedGPITriggerParams(GPITriggerParams triggerParams);			
引数	引数名	In/Out	型	説明
	triggerParams	In	GPITriggerParams	AsReader は指定した入力ポート GPI のトリガーパラメータ GPITriggerParams 型 (5.6 を参照)
戻り値	-	-	-	-

デリゲートの説明 :

メソッド GetGpiTrigger (2.1.20 を参照) を実行後に、本デリゲートをコールバックします。

サンプルコード :

```

WhenReceivedGPITriggerParams whenReceivedGPITriggerParams =
ReceivedGPITriggerParams;
void ReceivedGPITriggerParams(GPITriggerParams triggerParams){
    //AsReader は指定した入力ポート GPI のトリガーパラメータを受け取ります。
}
whenReceivedGPITriggerParams(triggerParams);

```

3.1.11. WhenReceivedReaderTime

デリゲート名	delegate void WhenReceivedReaderTime(string readerTime);			
引数	引数名	In/Out	型	説明

	readerTime	In	string	AsReader のシステムタイム
返り値	-	-	-	-
<p>デリゲートの説明 : メソッド GetReaderTime (2.1.22 を参照) を実行後に、本デリゲートをコールバックします。</p> <p>サンプルコード : WhenReceivedReaderTime whenReceivedReaderTime = ReceivedReaderTime; void ReceivedReaderTime(string readerTime){ //AsReader のシステムタイムを受け取ります。 } whenReceivedReaderTime(readerTime);</p>				

3.1.12. WhenReceivedRs485Params

デリゲート名	delegate void WhenReceivedRs485Params(Rs485Param rs485Param);			
引数	引数名	In/Out	型	説明
	rs485Param	In	Rs485Param	AsReader の通信パラメータ Rs485Param 型 (5.4 を参照)
返り値	-	-	-	-
<p>デリゲートの説明 : メソッド GetRs485 (2.1.24 を参照) を実行後に、本デリゲートをコールバックします。</p> <p>サンプルコード : WhenReceivedRs485Params whenReceivedRs485Params = ReceivedRs485Params; void ReceivedRs485Params(Rs485Param rs485Param){ //AsReader の通信パラメータを受け取ります。 } whenReceivedRs485Params(rs485Param);</p>				

3.1.13. WhenReceivedTemperatureControl

デリゲート名	delegate void WhenReceivedTemperatureControl(string currentTemperature);
--------	--

引数	引数名	In/Out	型	説明
	currentTemperature	In	string	AsReader 現在の温度
返り値	-	-	-	-

デリゲートの説明：
メソッド GetTemperatureControl (2.1.26 を参照) を実行後に、本デリゲートをコールバックします。

サンプルコード：

```
WhenReceivedTemperatureControl whenReceivedTemperatureControl=
ReceivedTemperatureControl;
void ReceivedTemperatureControl(string currentTemperature)
{
    // AsReader 現在の温度を受け取ります。
}
whenReceivedTemperatureControl(currentTemperature);
```

3.1.14. WhenReceivedEas

デリゲート名	delegate void WhenReceivedEas(EasInfo eas);			
引数	引数名	In/Out	型	説明
	eas	In	EasInfo	AsReader のマッチングのアラートパラメータ EasInfo 型 (5.9 を参照)
返り値	-	-	-	-

デリゲートの説明：
メソッド GetEas (2.1.28 を参照) を実行後に、本デリゲートをコールバックします。

サンプルコード：

```
WhenReceivedEas whenReceivedEas = ReceivedEas;
void ReceivedEas(EasInfo eas){
    //AsReader のマッチングのアラートパラメータを受け取ります。
}
whenReceivedEas(eas);
```

3.1.15. WhenReceivedAntPower

デリゲート名	delegate void WhenReceivedAntPower(AntPower antPower);			
引数	引数名	In/Out	型	説明
	antPower	In	AntPower	AsReader の各アンテナのパワー AntPower 型 (5.11 を参照)
返り値	-	-	-	-
<p>デリゲートの説明 : メソッド GetAntPower (2.1.30 を参照) を実行後に、本デリゲートをコールバックします。</p> <p>サンプルコード : <pre>WhenReceivedAntPower whenReceivedAntPower = ReceivedAntPower; void ReceivedAntPower(AntPower antPower){ //AsReader の各アンテナポートのパワーを受け取ります。 } whenReceivedAntPower(antPower);</pre> </p>				

3.1.16. WhenReceivedFreqRange

デリゲート名	delegate void WhenReceivedFreqRange(int freqRange);			
引数	引数名	In/Out	型	説明
	freqRange	In	int	AsReader の稼働周波数帯域
返り値	-	-	-	-
<p>デリゲートの説明 : メソッド GetFreqRange (2.1.32 を参照) を実行後に、本デリゲートをコールバックします。</p> <p>サンプルコード : <pre>WhenReceivedFreqRange whenReceivedFreqRange = ReceivedFreqRange; void ReceivedFreqRange(int freqRange){ //AsReader の稼働周波数帯域を受け取ります。 } whenReceivedFreqRange(freqRange);</pre> </p>				

3.1.17. WhenReceivedFrequency

デリゲート名	delegate void WhenReceivedFrequency(Frequency frequency);			
引数	引数名	In/Out	型	説明
	frequency	In	Frequency	AsReaderの稼働周波数の設定（5.16を参照）
戻り値	-	-	-	-
<p>デリゲートの説明： メソッド GetFrequency（2.1.34を参照）を実行後に、本デリゲートをコールバックします。</p> <p>サンプルコード： <pre>WhenReceivedFrequency whenReceivedFrequency = ReceivedFrequency; void ReceivedFrequency(Frequency frequency){ //AsReaderの稼働周波数の設定を受け取ります。 } whenReceivedFrequency(frequency);</pre></p>				

3.1.18. WhenReceivedFilter

デリゲート名	delegate void WhenReceivedFilter(Filter filter);			
引数	引数名	In/Out	型	説明
	filter	In	Filter	AsReaderのタグのフィルタリングのパラメータ Filter 型（5.14を参照）
戻り値	-	-	-	-
<p>デリゲートの説明： メソッド GetFilter（2.1.36を参照）を実行後に、本デリゲートをコールバックします。</p> <p>サンプルコード： <pre>WhenReceivedFilter whenReceivedFilter= ReceivedFilter; void ReceivedFilter(Filter filter){ //AsReaderのタグのフィルタリングのパラメータを受け取ります。 } whenReceivedFilter(filter);</pre></p>				

3.1.19. WhenReceivedEPCBaseband

デリゲート名	delegate void WhenReceivedEPCBaseband(EPCBaseband epcBaseband);			
引数	引数名	In/Out	型	説明
	epcBaseband	In	EPCBaseband	AsReader が使用する Baseband パラメータ EPCBaseband 型 (5.10 を参照)
戻り値	-	-	-	-
<p>デリゲートの説明 : メソッド GetEPCBaseband (2.1.38 を参照) を実行後に、本デリゲートをコールバックします。</p> <p>サンプルコード : <pre>WhenReceivedEPCBaseband whenReceivedEPCBaseband = ReceivedEPCBaseband; void ReceivedEPCBaseband(EPCBaseband epcBaseband){ //AsReader が使用する Baseband パラメータを受け取ります。 } whenReceivedEPCBaseband(epcBaseband);</pre></p>				

3.1.20. WhenReceivedAutoFree

デリゲート名	delegate void WhenReceivedAutoFree(AutoFree autoFree);			
引数	引数名	In/Out	型	説明
	autoFree	In	AutoFree	AsReader のタグ読取モードの読取間隔 AutoFree 型 (5.15 を参照)
戻り値	-	-	-	-
<p>デリゲートの説明 : メソッド GetAutoFree (2.1.40 を参照) を実行後に、本デリゲートをコールバックします。</p> <p>サンプルコード : <pre>WhenReceivedAutoFree whenReceivedAutoFree = ReceivedAutoFree; void ReceivedAutoFree(AutoFree autoFree){ //AsReader のタグ読取モードの読取間隔を受け取ります。 } whenReceivedAutoFree(autoFree);</pre></p>				

3.1.21. WhenReceivedBaseStop

デリゲート名	delegate void WhenReceivedBaseStop(int resultCode, string msg);			
引数	引数名	In/Out	型	説明
	resultCode	In	int	AsReader の読取停止アクションの実行結果
	msg	In	string	AsReader の読取停止アクションの実行結果の説明
戻り値	-	-	-	-

デリゲートの説明：
 メソッド BaseStop (2.1.43 を参照) を実行後に、本デリゲートをコールバックします。

サンプルコード：

```
WhenReceivedBaseStop whenReceivedBaseStop = ReceivedBaseStop;
void ReceivedBaseStop(int resultCode, string msg){
    //AsReader の読取停止アクションの実行結果を受け取ります。
}
whenReceivedBaseStop(resultCode, msg);
```

3.1.22. WhenReceivedTag

デリゲート名	delegate void WhenReceivedTag(Tag tag);			
引数	引数名	In/Out	型	説明
	tag	In	Tag	AsReader のインベントリしたタグデータ Tag 型 (5.19 を参照)
戻り値	-	-	-	-

デリゲートの説明：
 メソッド BaseRead (2.1.41 を参照) を実行後に、AsReader がタグをインベントリした場合、本デリゲートをコールバックします。

サンプルコード：

```
WhenReceivedTag whenReceivedTag = ReceivedTag;
void ReceivedTag(Tag tag){
```

```
//AsReader のインベントリしたタグデータを受け取ります。
}
whenReceivedTag(tag);
```

3.1.23. WhenReceivedMagnusS3Tag

デリゲート名	delegate void WhenReceivedMagnusS3Tag(Tag tag);			
引数	引数名	In/Out	型	説明
	tag	In	Tag	AsReader のインベントリしたタグデータ Tag 型 (5.19 を参照)
戻り値	-	-	-	-

デリゲートの説明:
メソッド BaseInventory (2.1.42 を参照) を実行後に、AsReader が温度タグをインベントリした場合、本デリゲートをコールバックします。

サンプルコード:

```
WhenReceivedMagnusS3Tag whenReceivedMagnusS3Tag = ReceivedMagnusS3Tag;
void ReceivedMagnusS3Tag(Tag tag){
    //AsReader のインベントリしたタグデータを受け取ります。
}
whenReceivedMagnusS3Tag (tag);
```

3.1.24. WhenReceivedMoistureTag

デリゲート名	delegate void WhenReceivedMoistureTag(Tag tag);			
引数	引数名	In/Out	型	説明
	tag	In	Tag	AsReader のインベントリしたタグデータ Tag 型 (5.19 を参照)
戻り値	-	-	-	-

デリゲートの説明:
メソッド BaseInventory (2.1.42 を参照) を実行後に、AsReader が湿度タグをインベントリした場合、本デリゲートをコールバックします。

サンプルコード:

```

WhenReceivedMoistureTag whenReceivedMoistureTag= ReceivedMoistureTag;
void ReceivedMoistureTag(Tag tag){
    //AsReader のインベントリしたタグデータを受け取ります。
}
whenReceivedMoistureTag(tag);

```

3.1.25. WhenReceivedTagReadEnd

デリゲート名	delegate void WhenReceivedTagReadEnd(Types.ReadEndReasonCode readEndReasonCode);			
引数	引数名	In/Out	型	説明
	readEndReasonCode	In	Types.ReadEndReasonCode	AsReader のタグ読取停止の原因 列挙型 ReadEndReasonCode (4.1.19 を参照)
戻り値	-	-	-	-

デリゲートの説明:

メソッド BaseRead (2.1.41 を参照) を実行してタグを読取停止、またはメソッド BaseStop (2.1.43 を参照) や BaseInventory (2.1.42 を参照) を実行してシングル読取は停止後に、本デリゲートをコールバックします。

サンプルコード:

```

WhenReceivedTagReadEnd whenReceivedTagReadEnd= ReceivedTagReadEnd;
void ReceivedTagReadEnd(Types.ReadEndReasonCode readEndReasonCode){
    //AsReader のタグを読取停止の原因を受け取ります。
}
whenReceivedTagReadEnd(readEndReasonCode);

```

3.1.26. WhenReceivedData

デリゲート名	delegate void WhenReceivedData(byte[] data);			
引数	引数名	In/Out	型	説明
	data	In	byte[]	AsReader から受け取るデータ
戻り値	-	-	-	-

デリゲートの説明:

AsReader からデータを受け取る際に、本デリゲートをコールバックします。

サンプルコード:

```
WhenReceivedData whenReceivedData= ReceivedData;
void ReceivedData(byte[] data){
    //AsReader から返したデータを受け取ります。
}
whenReceivedData();
```

3.1.27. WhenDisconnect

デリゲート名	delegate void WhenDisconnect();			
引数	引数名	In/Out	型	説明
	-	-	-	-
戻り値	-	-	-	-

デリゲートの説明:
AsReader との接続を切断した際に、本デリゲートをコールバックします。

サンプルコード:
WhenDisconnect whenDisconnect= ConnectError;
void ConnectError(){
 //AsReader との接続が切断した際の処理を受け取ります。
}
whenDisconnect();

3.1.28. WhenReceivedWorkTemperature

デリゲート名	delegate void WhenReceivedWorkTemperature(WorkTemperature workTemperature);			
引数	引数名	In/Out	型	説明
	workTemperature	In	WorkTemperature	AsReader 現在の稼働温度 WorkTemperature 型 (5.23 を参照)

戻り値	-	-	-	-
<p>デリゲートの説明： メソッド BaseRead (2.1.41 を参照)、またはメソッド BaseInventory (2.1.42 を参照) を実行後に、本デリゲートをコールバックします。</p> <p>サンプルコード: WhenReceivedWorkTemperature whenReceivedWorkTemperature= ReceivedWorkTemperature; void ReceivedWorkTemperature(WorkTemperature workTemperature) { //AsReader の現在の稼働温度を受け取ります。 } whenReceivedWorkTemperature(workTemperature);</p>				

3.1.29. WhenReceivedTriggerStart

デリゲート名	delegate void WhenReceivedTriggerStart(Types.GPITriggerPorts gpiTriggerPorts, Types.GPOStatus gpoStatus);			
引数	引数名	In/Out	型	説明
	gpiTrigge rPorts	In	Types.GPITriggerPort s	トリガー-GPI ポート Types.GPITriggerPorts 型 (請参照 : 4.1.7)
	gpoStatu s	In	Types.GPOStatus	GPI ポート電圧レベル Types.GPOStatus 型 (4.1. 4 を参照)
戻り値	-	-	-	-
<p>デリゲートの説明： メソッド SetGpiTrigger (2.1.19 を参照) を実行し、パラメータ GPITriggerParamsSetting にある ReportOnOff を 1 に設定後に、トリガー条件に満たすと、本デリゲートをコールバックします。</p> <p>サンプルコード: WhenReceivedTriggerStart whenReceivedTriggerStart= ReceivedTriggerStart; void ReceivedTriggerStart(Types.GPITriggerPorts gpiTriggerPorts, Types.GPOStatus gpoStatus) {</p>				

```
//AsReader から返したトリガーアクションの GPI ポートと GPI ポート電圧レベルを受け取ります。
}
whenReceivedTriggerStart(gpiTriggerPorts, gpoStatus);
```

3.1.30. WhenReceivedTriggerStop

デリゲート名	delegate void WhenReceivedTriggerStop(Types.GPITriggerPorts gpiTriggerPorts, Types.GPOStatus gpoStatus);			
引数	引数名	In/Out	型	説明
	gpiTriggerPorts	In	Types.GPITriggerPorts	トリガー GPI ポート Types.GPITriggerPorts 型 (4.1.7 を参照)
	gpoStatus	In	Types.GPOStatus	GPI ポート電圧レベル Types.GPOStatus 型 (4.1.4 を参照)
戻り値	-	-	-	-

デリゲートの説明 :
 メソッド SetGpiTrigger (2.1.19 を参照) を実行し、パラメータ GPITriggerParamsSetting にある ReportOnOff を 1 に設定後、トリガー停止条件に満たすと、本デリゲートをコールバックします。

サンプルコード:

```
WhenReceivedTriggerStop whenReceivedTriggerStop= ReceivedTriggerStop;
void ReceivedTriggerStop(Types.GPITriggerPorts gpiTriggerPorts, Types.GPOStatus gpoStatus)
{
    //AsReader から返したトリガー停止の GPI ポートと GPI ポート電圧レベルを受け取ります
}
ReceivedTriggerStop(gpiTriggerPorts, gpoStatus);
```

4. Types クラス

4.1. 列挙

4.1.1. EthernetIPType

イーサネットの IP タイプ

定義	説明
Automatic	動的 IP
StaticIP	固定 IP

4.1.2. CSMode

TCP モード

定義	説明
Server	サーバーモード

4.1.3. Baudrate

ボーレート

定義	説明
_9600	9600 bps
_19200	19200 bps
_115200	115200 bps
_230400	230400 bps
_460800	460800 bps

4.1.4. GPOStatus

GPIO 出力ポートの電圧レベル

定義	説明
Low_level	電圧レベル Low
High_level	電圧レベル High

4.1.5. GPITriggerStartConditions

トリガー開始条件

定義	説明
Off	トリガーオフ
Low_level	Low 電圧レベルトリガー
High_level	High 電圧レベルトリガー
Rising_edge	アップエッジトリガー
Falling_edge	ダウンエッジトリガー
Any_edge	いずれのエッジトリガー

4.1.6. GPITriggerStopConditions

トリガー停止条件

定義	説明
Off	トリガーオフ
Low_level	Low 電圧レベルトリガー
High_level	High 電圧レベルトリガー
Rising_edge	アップエッジトリガー
Falling_edge	ダウンエッジトリガー
Any_edge	いずれのエッジトリガー
Delay_stop	遅延停止

4.1.7. GPITriggerPorts

GPI トリガー ポート番号

定義	説明
GPI1	GPI ポート 1
GPI2	GPI ポート 2
GPI3	GPI ポート 3
GPI4	GPI ポート 4

4.1.8. OnOff

オン/オフ

定義	説明
Off	オフ
On	オン

4.1.9. EasMatchArea

データエリアのマッチング

定義	説明
EPC	EPC エリア
TID	TID エリア
Userdata	ユーザーエリア
Reserved	リザーブドエリア

4.1.10. CloseOpen

アラートオン/オフ

定義	説明
Close	アラート機能オフ
Open	アラート機能オン

4.1.11. FrequencyAutoFixed

周波数のオート設定

定義	説明
Fixed_frequency	周波数リストに指定された周波数を使用します。
Auto_frequency	RF 周波数帯域内で自動的に周波数を選択します。

4.1.12. InventoryMode

タグの読取モード

定義	説明
Single	シングル読取
Repeat	連続読取

4.1.13. TagArea

タグエリア

定義	説明
EPC	EPC エリア
TID	TID エリア
Userdata	ユーザーエリア
Reserved	リザーブドエリア

4.1.14. MatchArea

タグエリアのマッチング

定義	説明
None	エリアをマッチングしない
EPC	EPC エリアをマッチング
TID	TID エリアをマッチング
Userdata	ユーザーエリアをマッチング

4.1.15. TIDMode

TID 読取モード

定義	説明
Adaptive	TID の読取長さを自動的に調整
FixedLength	指定された長さで TID を読取

4.1.16. WriteTagArea

タグエリアの書き込み

定義	説明
Reserved	リザーブドエリア
EPC	EPC エリア
TID	TID エリア
User	ユーザーエリア

4.1.17. InventoryType

タグタイプ

定義	説明
EPC	EPC タグ
Temperature	温度タグ
Moisture	湿度タグ

4.1.18. ReadResult

タグデータの読取結果

定義	説明
ReadSucceed	読取成功
TagNoResponse	タグからレスポンスなし
CRCError	CRC エラー

DataAreaLocked	データエリアロック済み
DataAreaOverflow	データオーバーフロー
PasswordError	アクセスパスワードエラー
OtherTagError	そのほかのタグエラー
OtherReaderError	そのほかの AsReader エラー

4.1.19. ReadEndReasonCode

読取停止原因

定義	説明
ReadSucceed	読取完了
StopCommandRceived	読取停止のコマンドを受信
HardwareFailureReadInterrupt	ハード故障で読取中止

5. 構造体

5.1. ReaderInformation

AsReader の基本情報

変数名	型	説明
ReaderSerialNo	string	AsReader の製造番号
ReaderPowerOnTime	string	AsReader の起動時間 AsReader の電源を入れてからの経過時間（単位：s）
BasebandCompilationTime	string	Baseband のコンパイル時間
ApplicationVer	string	アプリケーションのバージョン
SystemVer	string	システムバージョン
AppCompilationTime	string	アプリケーションのコンパイル時間

5.2. EthernetIP

AsReader の IP アドレスパラメータ

変数名	型	説明
AutoOrStaticIP	byte	オート/固定 IP 0：動的 IP 1：固定 IP
IP	string	AsReader の IP アドレス
Mask	string	AsReader のサブネットマスク
Gateway	string	AsReader のデフォルトゲートウェイ
Dns1	string	Dns1
Dns2	string	Dns2

5.3. TcpMode

AsReader の TCP モード及び関連のパラメータ

変数名	型	説明
CSMode	byte	TCP モード 0 : サーバモード
ServerPort	UInt16	AsReader の TCP ポート 設定可能範囲 : 1024~65535 初期値 : 8160

5.4. Rs485Param

AsReader の RS485 の通信パラメータ

変数名	型	説明
Address	byte	AsReader の RS485 アドレス 値の範囲 : 0~255
BaudrateIndex	UInt16	通信ポーレート 0 : 9600 bps 1 : 19200 bps 2 : 115200 bps 3 : 230400 bps 4 : 460800 bps

5.5. GPIState

AsReader の GPIO 入力ポート電圧レベルステータス

変数名	型	説明
GPI1	byte	GPI1 電圧レベル 0 : 電圧レベル Low 1 : 電圧レベル High
GPI2	byte	GPI2 電圧レベル 0 : 電圧レベル Low

		1 : 電圧レベル Low
GPI3	byte	GPI3 電圧レベル 0 : 電圧レベル Low 1 : 電圧レベル High
GPI4	byte	GPI4 電圧レベル 0 : 電圧レベル Low 1 : 電圧レベル High

5.6. GPITriggerParams

AsReader の GPI トリガーパラメータ

変数名	型	説明
TriggerStartCondition	byte	トリガー開始条件 0 : トリガーオフ 1 : Low 電圧レベルトリガー 2 : High 電圧レベルトリガー 3 : アップエッジトリガー 4 : ダウンエッジトリガー 5 : いずれのエッジトリガー
TriggerBindCommand	string	トリガーバンドルコマンド トリガーバンドルコマンドはフレームヘッダーと CRC16 検証フィールドを含まない
TriggerStopCondition	byte	トリガー停止条件 0 : 停止しない 1 : Low 電圧レベルトリガー 2 : High 電圧レベルトリガー 3 : アップエッジトリガー 4 : ダウンエッジトリガー 5 : いずれのエッジトリガー 6 : デレイ停止 トリガー停止条件に満たすと AsReader は停止コマンドを実行する
DelayTime	UInt16	デレイ停止タイム 単位 : 10ms

		0 の場合、デレイタイム制限なし 停止条件はデレイ停止の場合のみこのパラメータが有効
ReportOnOff	byte	トリガー停止しない時に IO 電圧レベル変更を通知するスイッチ 0 : アップロードしない 1 : アップロード

5.7. GPITriggerParamsSetting

AsReader の GPI トリガーパラメータ、設定する時に使う

変数名	型	説明
port	byte	トリガー-GPI ポート 0 : GPI1 1 : GPI2 2 : GPI3 3 : GPI4。
TriggerStartCondition	byte	トリガー開始条件 0 : トリガーオフ 1 : Low 電圧レベルトリガー 2 : High 電圧レベルトリガー 3 : アップエッジトリガー 4 : ダウンエッジトリガー 5 : いずれのエッジトリガー
TriggerBindCommand	string	トリガーバンドルコマンド トリガーバンドルコマンドはフレームヘッダーと CRC16 検証フィールドを含まない
TriggerStopCondition	byte	トリガー停止条件 0 : 停止しない 1 : Low 電圧レベルトリガー 2 : High 電圧レベルトリガー 3 : アップエッジトリガー 4 : ダウンエッジトリガー 5 : いずれのエッジトリガー 6 : デレイ停止 トリガー停止条件に満たすと AsReader は停止コマンドを実行

		する
DelayTime	UInt16	ディレイ停止タイム 単位：10ms 0の場合、ディレイタイム制限なし 停止条件はディレイ停止の場合のみこのパラメータが有効
ReportOnOff	byte	トリガー停止しない時に IO 電圧レベル変更を通知するスイッチ 0：アップロードしない 1：アップロード

5.8. EasAction

AsReader は EAS アクションのマッチングパラメータ

変数名	型	説明
KeepTime	UInt16	キープタイム 単位：秒。0x0001：アクションは 1s キープして、デフォルト電圧レベルに戻る
GPO1	string	0：デフォルト電圧レベル High、出力電圧レベル Low 1：デフォルト電圧レベル Low、出力電圧レベル High
GPO2	string	0：デフォルト電圧レベル High、出力電圧レベル Low 1：デフォルト電圧レベル Low、出力電圧レベル High
GPO3	string	0：デフォルト電圧レベル High、出力電圧レベル Low 1：デフォルト電圧レベル Low、出力電圧レベル High
GPO4	string	0：デフォルト電圧レベル High、出力電圧レベル Low 1：デフォルト電圧レベル Low、出力電圧レベル High

5.9. EasInfo

AsReader の EAS パラメータ

変数名	型	説明
AlarmOnOff	byte	アラームオン/オフ 0：アラームオフ

		1 : アラームオン
MatchArea	byte	マッチングデータエリアを指定する 0 : EPC エリア 1 : TID エリア 2 : ユーザーエリア 3 : リザーブドエリア
MatchStart	UInt16	マッチングデータのスタートアドレス
MatchData	string	マッチングデータの内容
Mask	string	マスク マッチングデータビットを指定する
EASMatchedAction	EasAction	マッチング成功アクションのパラメータ (5.8 を参照)
EASUnMatchedAction	EasAction	マッチング失敗アクションのパラメータ (5.8 を参照)

5.10. EPCBaseband

AsReader の EPC ベースバンドパラメータ

変数名	型	説明
EPCSpeed	byte	EPC ベースバンドの速度 0 : Tari=25us,FM0,LHF=40KHz 1 : Dense mode. 2 : Tari=25us,Miller4,LHF=300KHz 3 : Fast mode. 4 : Tari=20us,Miller2,LHF=320KHz 5 : Tari=6.25us,FM0,LHF=620KHz 6 : Tari=20us,Miller4,LHF=320KHz 7 : Tari=6.25us,Miller4,LHF=400KHz 10 : Tari=12.5us,Miller4,LHF=250KHz 11 : Tari=6.25us,Miller4,LHF=250KHz 12 : Tari=12.5us,Miller4,LHF=300KHz 13 : Tari=6.25us,Miller4,LHF=300KHz 14 : Auto。
QValue	byte	デフォルト : Q 値

		0~15、AsReader のイベントリスタートの時 Q 値
Session	byte	0 : Session0 1 : Session1 2 : Session2 3 : Session3。
SearchType	byte	イベントリタイプパラメータ 0 : Flag A だけイベントリ 1 : Flag B だけイベントリ 2 : Flag A と Flag B を交代でイベントリ

5.11. AntPower

AsReader の各アンテナのパワー

変数名	型	説明
Ant1	byte	アンテナポート 1 のパワー 0~30、単位 : dBm、ステップ 1dB
Ant2	byte	アンテナポート 2 のパワー 0~30、単位 : dBm、ステップ 1dB
Ant3	byte	アンテナポート 3 のパワー 0~30、単位 : dBm、ステップ 1dB
Ant4	byte	アンテナポート 4 のパワー 0~30、単位 : dBm、ステップ 1dB
Ant5	byte	アンテナポート 5 のパワー 0~30、単位 : dBm、ステップ 1dB
Ant6	byte	アンテナポート 6 のパワー 0~30、単位 : dBm、ステップ 1dB
Ant7	byte	アンテナポート 7 のパワー 0~30、単位 : dBm、ステップ 1dB
Ant8	byte	アンテナポート 8 のパワー 0~30、単位 : dBm、ステップ 1dB

5.12. ReaderTime

AsReader のシステムタイム

変数名	型	説明
UTCSecond	string	AsReader のシステムタイム タイムフォーマット : YYYY.MM.DD.hh:mm:ss

5.13. GPO

AsReader の GPIO 出力ポート電圧レベルパラメータ

変数名	型	説明
GPO1	string	0 : 出力電圧レベル Low 1 : 出力電圧レベル High
GPO2	string	0 : 出力電圧レベル Low 1 : 出力電圧レベル High
GPO3	string	0 : 出力電圧レベル Low 1 : 出力電圧レベル High
GPO4	string	0 : 出力電圧レベル Low 1 : 出力電圧レベル High

5.14. Filter

AsReader のタグ読み取りのフィルタリングパラメータ

変数名	型	説明
RepeatTime	UInt16	同一タグの読み取り間隔 0~65535、タイム単位 : 10ms
RssiMax	int	タグの RSSI 値が閾値より低い場合、タグデータが廃棄される 0~110

5.15. AutoFree

AsReader はタグ読取モードの読取間隔

変数名	型	説明
OpenClose	byte	読取間隔モードを有効にする 0 : 読取間隔モードをオフ 1 : 読取間隔モードをオン
FreeTime	UInt16	間隔タイム 読取間隔モードに入った後の間隔タイムを設定する 0~65535、単位 : 10ms

5.16. Frequency

AsReader の稼働周波数

変数名	型	説明
FreqAutoFixed	byte	周波数を自動的に選択する 0 : AsReader は指定された周波数リストの周波数を使用する 1 : AsReader は RF 周波数帯域内で自動的に周波数を選択する
PointIndexList	string	周波数リスト 周波数を任意で設定したい場合に稼働周波数を指定する 周波数リストは稼働帯域のチャンネル番号リストになる 例えば、国際標準 920~925MHz の周波数帯域に 920.625、922.375、924.375MHz を指定したい場合、 周波数リストは「0,7,15」になる 周波数は最低 1、最大 50 になる

5.17. ReadParameters

AsReader のタグ読取パラメータ

変数名	型	説明
-----	---	----

Ant1	bool	True : アンテナ 1 を使う False : アンテナ 1 を使わない
Ant2	bool	True : アンテナ 2 を使う False : アンテナ 2 を使わない
Ant3	bool	True : アンテナ 3 を使う False : アンテナ 3 を使わない
Ant4	bool	True : アンテナ 4 を使う False : アンテナ 4 を使わない
Ant5	bool	True : アンテナ 5 を使う False : アンテナ 5 を使わない
Ant6	bool	True : アンテナ 6 を使う False : アンテナ 6 を使わない
Ant7	bool	True : アンテナ 7 を使う False : アンテナ 7 を使わない
Ant8	bool	True : アンテナ 8 を使う False : アンテナ 8 を使わない
MatchArea	Types.MatchArea	マッチング対象エリア Types.MatchArea 型 (4.1.14 を参照)
ReadTagArea	Types.TagArea	読み取り対象タグエリア Types.TagArea 型 (4.1.13 を参照)
MatchStartAddresses	int	マッチング対象のスタートアドレス (単位 : word)
MatchData	string	マッチングデータ
TIDMode	Types.TIDMode	TID 読取モード Types.TIDMode (4.1.15 を参照)
StartAddress	int	読取するデータエリアのスタートアドレス (単位 : word)
ReadDataLength	byte	読取のデータ長さ (単位 : word)
Password	string	タグのアクセスパスワード

5.18. InventoryParameters

AsReader のタグインベントリパラメータ

変数名	型	説明
Ant1	bool	True : アンテナ 1 を使う False : アンテナ 1 を使わない
Ant2	bool	True : アンテナ 2 を使う False : アンテナ 2 を使わない
Ant3	bool	True : アンテナ 3 を使う False : アンテナ 3 を使わない
Ant4	bool	True : アンテナ 4 を使う False : アンテナ 4 を使わない
Ant5	bool	True : アンテナ 5 を使う False : アンテナ 5 を使わない
Ant6	bool	True : アンテナ 6 を使う False : アンテナ 6 を使わない
Ant7	bool	True : アンテナ 7 を使う False : アンテナ 7 を使わない
Ant8	bool	True : アンテナ 8 を使う False : アンテナ 8 を使わない
SingleOrRepeat	Types.InventoryMode	連続/シングルインベントリ Types.InventoryMode 型 (4.1.12 を参照)
ReadEPCBank	bool	EPCBank エリアの読取要否 True : 読取 False : 読取しない
ReadTIDArea	bool	TID エリアの読取要否 True : 読取 False : 読取しない
ReadUserArea	bool	ユーザーエリアの読取要否 True : 読取 False : 読取しない
ReadReserved	bool	リザーブドエリアの読取要否

Area		True : 読取 False : 読取しない
InventoryType	Types.InventoryType	インベントリタグのタイプ Types.InventoryType 型 (4.1.17 を参照)

5.19. Tag

AsReader のタグデータパラメータ

変数名	型	説明
EPC	string	タグの EPC コード
PC	string	タグの PC データ
AntID	byte	読取したタグのアンテナ ID ナンバー 1 : アンテナ 1 2 : アンテナ 2 3 : アンテナ 3 4 : アンテナ 4 5 : アンテナ 5 6 : アンテナ 6 7 : アンテナ 7 8 : アンテナ 8
ReadResult	byte	タグの読取結果 (TID エリア、ユーザーエリア、リザーブドエリアを含む) 0 : 読取成功 1 : タグからレスポンスなし 2 : CRC エラー 3 : データエリアロック済み 4 : データオーバーフロー 5 : アクセスパスワードエラー 6 : そのほかのタグエラー 7 : そのほかの AsReader エラー
TID	string	タグの TID データ
UserData	string	タグのユーザーエリアデータ

ReservedData	string	タグのリザーブドエリアのデータ
Channel	UInt32	タグのチャンネル、単位： KHz
EPCBank	string	タグの EPC データ
MoistureData	Int16	タグの湿度データ
TemperatureData	double	タグの温度データ
Phase	double	タグの位相データ
Rssi	int	タグの RSSI 値

5.20. WriteParameters

AsReader のタグ書き込みパラメータ

変数名	型	説明
Ant1	bool	True : アンテナ 1 を使う False : アンテナ 1 を使わない
Ant2	bool	True : アンテナ 2 を使う False : アンテナ 2 を使わない
Ant3	bool	True : アンテナ 3 を使う False : アンテナ 3 を使わない
Ant4	bool	True : アンテナ 4 を使う False : アンテナ 4 を使わない
Ant5	bool	True : アンテナ 5 を使う False : アンテナ 5 を使わない
Ant6	bool	True : アンテナ 6 を使う False : アンテナ 6 を使わない
Ant7	bool	True : アンテナ 7 を使う False : アンテナ 7 を使わない
Ant8	bool	True : アンテナ 8 を使う False : アンテナ 8 を使わない
MatchArea	Types.MatchArea	マッチングエリア Types.MatchArea 型 (4.1.14 を参照)

MatchData	string	マッチングデータ
MatchStartAddresses	int	スタートアドレスマッチング
DataArea	Types.WriteTagArea	書き込みのデータエリア Types.WriteTagArea 型 (4.1.16 を参照)
WordStartAddress	int	書き込みのデータエリアのスタートアドレス
WriteData	string	書き込みのデータ
Password	string	タグのアクセスパスワード

5.21. LockParameters

AsReader のタグロックパラメータ

変数名	型	説明
Ant1	bool	True : アンテナ 1 を使う False : アンテナ 1 を使わない
Ant2	bool	True : アンテナ 2 を使う False : アンテナ 2 を使わない
Ant3	bool	True : アンテナ 3 を使う False : アンテナ 3 を使わない
Ant4	bool	True : アンテナ 4 を使う False : アンテナ 4 を使わない
Ant5	bool	True : アンテナ 5 を使う False : アンテナ 5 を使わない
Ant6	bool	True : アンテナ 6 を使う False : アンテナ 6 を使わない
Ant7	bool	True : アンテナ 7 を使う False : アンテナ 7 を使わない
Ant8	bool	True : アンテナ 8 を使う False : アンテナ 8 を使わない

LockArea	byte	ロックエリア 0 : キルパスワードエリア 1 : アクセスパスワードエリア 2 : EPC エリア 3 : TID エリア 4 : ユーザーエリア
LockType	byte	ロックタイプ 0 : アンロック 1 : ロック 2 : 永久アンロック 3 : 永久ロック
MatchArea	Types.MatchArea	マッチングタグのデータエリア Types.MatchArea 型 (4.1.14 を参照)
MatchData	string	マッチングデータ
MatchStartAddresses	int	スタートアドレスマッチング
Password	string	アクセスパスワード

5.22. KillParameters

AsReader のタグキルパラメータ

変数名	型	説明
Ant1	bool	True : アンテナ 1 を使う False : アンテナ 1 を使わない
Ant2	bool	True : アンテナ 2 を使う False : アンテナ 2 を使わない
Ant3	bool	True : アンテナ 3 を使う False : アンテナ 3 を使わない
Ant4	bool	True : アンテナ 4 を使う False : アンテナ 4 を使わない
Ant5	bool	True : アンテナ 5 を使う False : アンテナ 5 を使わない

Ant6	bool	True : アンテナ 6 を使う False : アンテナ 6 を使わない
Ant7	bool	True : アンテナ 7 を使う False : アンテナ 7 を使わない
Ant8	bool	True : アンテナ 8 を使う False : アンテナ 8 を使わない
MatchArea	Types.MatchArea	マッチングエリア Types.MatchArea 型 (4.1.14 を参照)
MatchData	string	マッチングデータ
MatchStartAddresses	int	スタートアドレスマッチング
Password	string	タグのアクセスパスワード

5.23. WorkTemperature

AsReader の稼働温度

変数名	型	説明
TemperatureWork	string	温度パラメータ 単位 : 0.01 度
TemperatureThreshold	string	温度の閾値 単位 : 0.01 度

6. 付録

メソッド	setItemName	resultCode	ErrMsg	説明
SetSerialParam	RS232	0	Setting succeed.	設定成功
		1	Setting Failed.This baud rate is not supported.	失敗 このボーレートをサポートしない
SetRs485	RS485	0	Setting succeed.	設定成功
		1	Other error.	その他のエラー
SetEthernetIP	Ethernet	0	Setting succeed.	設定成功
		1	Reader IP parameters error.	AsReader IP パラメータエラー
SetReaderTime	Time	0	Setting succeed.	設定成功
		1	RTC setting failed.	RTC 設定失敗
SetTcpMode	Client/Server	0	Setting succeed.	設定成功
		1	Server IP parameter error.	サーバーIP パラメータエラー
SetBeepOnOff	Beep On/Off	0	Setting succeed.	設定成功
		1	Setting failed.	設定失敗
SetGPO	GPO	0	Setting succeed.	設定成功
		1	The port parameter that reader hardware does not	AsReader はこのポートパラメータをサポートしない

			support.	
SetGpiTrigger	GPI trigger settings	0	Setting succeed.	設定成功
		1	The port parameter that reader hardware does not support.	AsReaderはこのポートパラメータをサポートしない
		2	Parameter missing.	パラメータ紛失
SetEas	EAS	0	Setting succeed.	設定成功
		1	Setting failed.	設定失敗
SetFreqRange	Frequency Range	0	Setting succeed.	設定成功
		1	The frequency range parameter that reader hardware does not support.	AsReaderはこの周波数パラメータをサポートしない
		2	Save failed.	保存失敗
SetEPCBaseband	EPC Baseband	0	Setting succeed.	設定成功
		1	The baseband that reader hardware does not support.	AsReaderはこのベースバンドをサポートしない
		2	QV error.	Qのパラメータエラー
		3	Session error.	Sessionパラメータエラー
		4	Search Type error	Sessionフラッグパラメータエラー

		5	Other parameters error.	その他のパラメータエラー
		6	Save failed.	保存失敗
SetAntPower	Ant Power	0	Setting succeed.	設定成功
		1	The port parameters that reader hardware does not support.	AsReaderはこのポートパラメータをサポートしない
		2	The power parameters that reader hardware does not support.	AsReaderはこのパワーパラメータをサポートしない
		3	Save failed.	保存失敗
SetAutoFree	Auto Free	0	Setting succeed.	設定成功
		1	Mode parameter error.	モードパラメータエラー
		2	Other parameters error.	その他のパラメータエラー
		3	Save failed.	保存失敗
SetFilter	Filter	0	Setting succeed.	設定成功
		1	Parameters error.	パラメータエラー
		2	Save failed.	保存失敗
SetFrequency	Frequency	0	Setting succeed.	設定成功

		1	The channel number is not within the current frequency band.	チャンネル番号は現在の周波数帯域内ではない
		2	Invalid frequency.	無効な周波数
		3	Other parameters error.	その他のパラメータエラー
		4	Save failed.	保存失敗
BaseRead 或 BaseInventory	Read	0	Read succeed.	読取成功
		1	Antenna parameter error.	アンテナパラメータエラー
		2	Selection parameter error.	選択パラメータエラー
		3	TID read parameters error.	TID 読取パラメータエラー
		4	User data read parameters error.	ユーザデータエリアパラメータ読取エラー
		5	Reserved data read parameters error.	リザーブドエリアパラメータ読取エラー
		6	Other parameters error.	その他のパラメータエラー
WriteTag	Write	0	Write succeed.	書き込み成功
		1	Antenna parameter error.	アンテナパラメータエラー
		2	Selection parameter error.	選択パラメータエラー

		3	Write parameter error.	パラメータ書き込みエラー
		4	CRC check error.	CRC チェックエラー
		5	Deficit power.	パワー不足
		6	Data area overflow.	データエリアオーバー
		7	Data area is locked.	データエリアがロックされた
		8	Access password error.	アクセスパスワードエラー
		9	Other tags error.	その他のタグエラー
		10	Tag missing.	タグ
		11	Send command error.	コマンド送信エラー
LockTag	Lock	0	Option succeed.	オプション成功
		1	Antenna parameter error.	アンテナパラメータエラー
		2	Selection parameter error.	選択パラメータエラー
		3	Lock parameter error.	ロックパラメータエラー
		4	CRC check error.	CRC チェックエラー
		5	Deficit power.	パワー不足
		6	Data area overflow.	データエリアオーバー
		7	Data area is locked.	データエリアがロックされた
		8	Access	アクセスパスワードエラー

			password error.	
		9	Other tags error.	その他のタグエラー
		10	Tag missing.	タグ紛失
		11	Send command error.	コマンド送信エラー
KillTag	Kill	0	Kill succeed.	キルアクション成功
		1	Antenna parameter error.	アンテナパラメータエラー
		2	Selection parameter error.	選択パラメータエラー
		3	CRC check error.	CRC チェックエラー
		4	Deficit power.	パワー不足
		5	Kill password error.	キルパスワードエラー
		6	Other tags error.	その他のタグエラー
		7	Tag missing.	タグ紛失
		8	Send command error.	コマンド送信エラー