

WebMan for RS-232C

Version 1.00

プログラミング解説書

The logo for TechKnowledge features a stylized 'T' in orange with a grey dot above it, followed by the word 'Tech' in orange and 'Knowledge' in grey.

TechKnowledge

目次

目次	2
はじめに	5
WEBMAN FOR RS-232C について	5
開発ライセンス	6
ランタイム・ライセンス	6
禁止事項	6
保証規定	6
プロダクト・サポート	7
販売元	9
開発元、サポート	9
インストール	10
稼働システム条件	10
インストール	10
製品モジュール一覧	11
アンインストール	11
利用方法	13
ウェブページのコンポーネント定義について	13
コンポーネントのバージョンについて	13
インターネットエクスプローラーの設定について	13
コンポーネントの参照について	15
バイナリデータの送受信について	15
漢字コードについて	16
イベントによる受信処理について	16
コンポーネントインスタンスの存在期間について	16
サンプルプログラムについて	18
プロパティリファレンス	19
BaudRate	19
ByteSize	19
CustomBaudRate	20
DTREnable	20
FlowControl	20
InBufferCount	21
IsOpen	21

NotifyReceive	21
NotifySendComplete.....	21
NullDiscard	21
Parity	22
ParityReplace	22
Port	22
Receive.....	22
ReceiveByte	23
ReceiveQSize	23
ReceiveSize.....	23
ReceiveTimeOut.....	23
RTSEnable	24
SendQSize	24
SendTimeOut.....	24
StopBits	24
メソッドリファレンス	26
ClearBreak	26
ClearDTR	26
ClearRTS	26
CloseComm.....	27
FlushComm	27
GetCTS	28
GetDSR.....	28
GetRLSD	28
GetRing	29
IsPortAvailable	29
OpenComm.....	29
Send	30
SendBreak	30
SetRTS.....	31
Sleep	31
TransmitChar	31
イベントリファレンス	33
OnCommError	33
OnReceive.....	33
OnSendComplete	34
APPENDIX-A エラーコード	35
APPENDIX-B サンプル JAVASCRIPT コード.....	37
WEBMAN FOR RS-232C 調査依頼.....	39

SAMPLE

はじめに

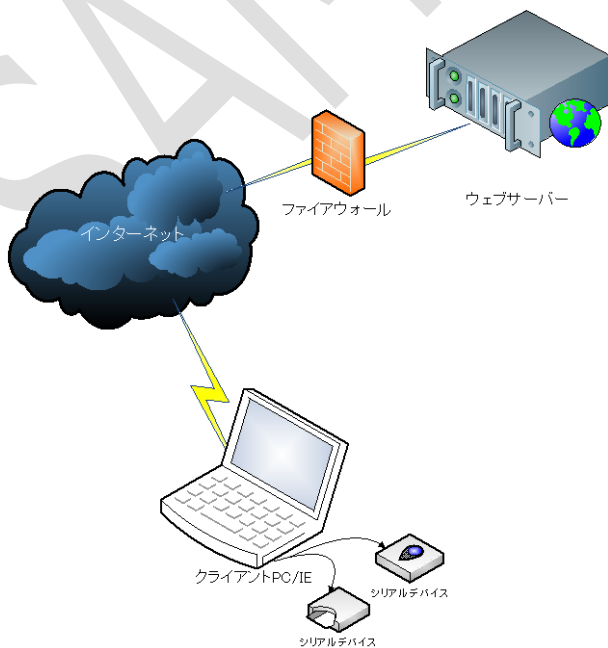
WebMan for RS-232C について

本製品はインターネットエクスプローラーで動作するJscript/VBScriptから利用可能なRS-232C通信をサポートするCOMコンポーネントです。

本製品をウェブページに設定しスクリプト言語で接続する機器に対応する通信プロトコル記述することで、クライアントPCに接続された機器のデータをウェブサーバーと送受信するシステムを作成することができます。

本マニュアルではインターネットエクスプローラのことをIEと略記する場合があります。Jscriptはjavascriptと記載します。VBScriptについての記載は割愛いたします。

以下は本製品を利用のシリアルデバイスをサポートするウェブアプリケーションのシステムイメージ図です。



開発ライセンス

開発ライセンスとは、開発者1名が1台のコンピュータ・システムで開発環境を利用することが出来る権利です。複数のコンピュータ・システムへのインストール、複数人での使用は著作権法違反となりますので、御注意ください。以下、開発者ライセンス使用権を単に使用権と記述します。

- WebMan for RS-232Cの使用権は第三者に譲渡および貸与することは出来ません。
- 使用権はWebMan for RS-232C製品パッケージを開梱したときに発効します。
- 使用権は以下のいずれかの事由が起こった場合に消滅します。
 - ① WebMan for RS-323Cに同封されているユーザー登録書を返送しない場合。
 - ② 使用規定に違反した場合。
 - ③ プログラム・ディスク、印刷物などを使用権の範囲外の目的で複製した場合。

ランタイム・ライセンス

本製品のランタイムライセンスは無償とします。

禁止事項

1. 本製品の不正複製を禁止します。
2. 本製品のリバースエンジニアリングを禁止します。
3. 本製品をラップし同種の製品を作成し販売することを禁止します。

保証規定

本製品、および付随する著作物に対して商品性及び特定の目的への適合性などについての保証を含むいかなる保証もそれを明記するしないに関わらず提供されることはありません。

本製品の著作者及び、製造、配布に関わるいかなる者も、当ソフトウェアの不具合によって発生する損害に対する責任は、それが直接的であるか間接的であるか、必然的であるか偶発的であるかに関わらず、負わないものとします。それは、その損害の可能性について、開発会社に事前に知らされていた場合でも同様です。

プロダクト・サポート

- ユーザー登録
まことにお手数ですが販売会社システム・ラボにてユーザー登録をお願いします。ユーザー登録が行われていないとお客様がユーザー・サポートが受けられない場合がございます。
- お問い合わせの方法
どうしても解決できない問題が発生した場合には、技術サポートをご利用ください。あらかじめ後ページの調査依頼書にお問い合わせ事項を記入していただき、インターネット・メールまたはファックスでお送りいただければ、折り返しご連絡をさせていただきます。本製品につきましては、複雑な内容のお問い合わせになることが多い為、電話によるユーザーサポートは実施しておりません。ご了承をお願いいたします。また、問い合わせの内容によっては、再現調査などのために、回答までに時間がかかる場合がありますので、かさねてご了承をお願いいたします。

サポートメールアドレス： support@techknowledge.co.jp

- 登録内容の変更について
転居などによるご住所や電話番号など登録内容に変更が生じた場合には、メールまたはファックスにて、販売会社システム・ラボまでご連絡をいただきますようお願いいたします。なお、電話による口頭での連絡変更は受けかねますのでよろしくお願いいたします。
- 併用される他社製品について
当社製品と併用される、他社製品の使い方等についてのご質問をお受けすることがあります。しかし、他社製品に関しましては、お答えできない場合があります。他社製品につきましては、該当開発・販売会社にご連絡ください。
- サポート対象
ご質問はご登録ユーザー様に限定させていただきます。ご登録ユー

ユーザー様以外からのご質問にはお答えできません。当ソフトウェアの料金にはご登録ユーザー様1名に限りサポート料が含まれています。

- サポート期間
製品のユーザー登録後、初回のお問い合わせから90日間は無償サポート期間とさせていただきます。また無償サポートは2件を上限とさせていただきます。無償サポート上限を超える場合には無償サポート終了以降もサポートをご希望の場合は有償サポートを承ります。有償サポートにつきましては販社システム・ラボにてお取り扱いしております。キャンペーン製品などディスカウント販売に該当する製品では無償サポート期間および回数の設定が短くなる場合がありますのであらかじめご了承ください。
- 最新版のご提供について
弊社webにて最新版の実行モジュールや技術情報、サンプル・コードの提供をしておりますのでサポートにご連絡になる前に弊社webをご参照いただけるようお願いいたします。URLは<http://www.techknowledge.co.jp>となります。
- ご質問の内容について
製品サポートは本製品に関連するご質問に限定させていただきます。

販売元

Systemlab®

(株)システムラボ

東京都杉並区上荻1丁目5番8号 直長ビル7F

電話 03-5397-7511

FAX 03-5397-7521

Internet-Mail info@systemlab.co.jp

URL www.systemlab.co.jp

開発元、サポート

TechKnowledge

(株)テクナレッジ

東京都世田谷区駒沢2丁目16番1号 サンドービル9F

電話 03-3421-7621

FAX 03-3421-6691

Internet-Mail info@techknowledge.co.jp

URL www.techknowledge.co.jp

商標登録

本マニュアルに記載される商標、登録商標は該当会社の商標または登録商標です。

インストール

WebMan for RS-232Cインストールについて説明します。

稼働システム条件

WebMan for RS-232Cは以下のOSをシステム稼働の前提とします。

- Windows Server 2008
- Windows Vista SP1以降
- Windows Server 2003
- Windows XP SP2以降

ウェブブラウザにつきましては以下をサポートします。将来のバージョンについても基本的にはサポートする方針ですが修正版での対応となる場合がございます。

- インターネットエクスプローラーバージョン 7.0
- インターネットエクスプローラーバージョン 6.0

本コンポーネントはウェブサーバーから配布することができます。ウェブサーバーにつきましては特定のウェブサーバーに依存するものではありませんが、以下で動作確認済です。

- インターネットインフォメーションサーバー(IIS) 7.0
- インターネットインフォメーションサーバー(IIS) 6.0
- Apache 2.x

インストール

WebMan for RS-232C 開発パッケージをインストールする手順について説明します。

- ① Windows のプログラム・マネージャーまたはエクスプローラー等から Web Man のインストール CD にある setup.exe を実行します。
- ② setup.exe の質問に答えてインストール・ボタンをクリックし、メッセージに答えてディスクトを入れ替えると、自動的にインストールが終了します。インストールが正常に終了すると WebMan のプログラム・グループが作成されます。

readme.html ファイルにはマニュアルに記述されていない最新情報が記述されています。最新情報が記述される場合もありますので、必ずご一読ください。

製品モジュール一覧

Windows のインストール・ディレクトリを<windir>、WebMan 製品のインストール・ディレクトリを<instdir>とした場合に、本製品がインストールするファイルの一覧を示します。

デフォルトインストールでは<indtdir>は C:¥Program Files¥TechKnowledge¥WebMan for RS-232C となります。

モジュール名とパス	内容
<instdir>¥bin¥wmIECom.DLL	RS-232C 通信サポートコンポーネント
<instdir>¥cab¥wmIECom.CAB	上記通信コンポーネント配布用キャビネットファイル
<instdir>¥doc¥webman_serial_100.pdf	当マニュアル
<instdir>¥doc¥readme.html	README ファイル
<instdir>¥samples¥	サンプル・プログラム・ファイル

アンインストール

自動アンインストール

コントロール・パネルの「アプリケーションの追加と削除」メニューから WebMan for RS-232C を選択することでアンインストールが可能です。以下は操作手順です。

- ① 「設定」メニューから「コントロール・パネル」を選択
- ② アプリケーションの追加と削除をダブル・クリック
- ③ 「セットアップと削除」タブのリスト・ボックスから「WebMan for RS-232C」を選択
- ④ 「追加と削除」ボタンをクリック

手動アンインストール

間違えてインストールしたフォルダーを削除した場合や、上書きインストールして自動アンインストール出来なくなった場合には以下の手動でアンインストールしてください。

- ① コマンドプロンプトを起動します。
- ② インストール・ディレクトリに移動します。デフォルト・インストールでは c:¥Program Files¥techknowledge¥WebMan or RS-232C となります。
- ③ インストール・ディレクトリを削除します。
`rmdir /s "c:¥Program Files¥TechKnowledge¥WebMan for RS-232C"`
を実行してください。
- ④ WebMan メニューを削除します。ショートカットなどを作成した場合も同様に削除します。

利用方法

ウェブページのコンポーネント定義について

ウェブページ(HTML ファイル)では以下のように `object` タグで本コンポーネントを宣言します。`codebase` 属性でコンポーネントを含むキャビネットファイルを指定します。以下の例ではウェブサーバーのドキュメントルートに `wmIECom.CAB` ファイルを配置しておくことになります。

```
<object name="serial" id="serial" width="0" height="0"
classid="CLSID:7CDF124F-8ABB-42E0-8719-C4DA2A8FE1C0"
codebase="wmIECom.DLL#version=1.0.0.1">
</object>
```

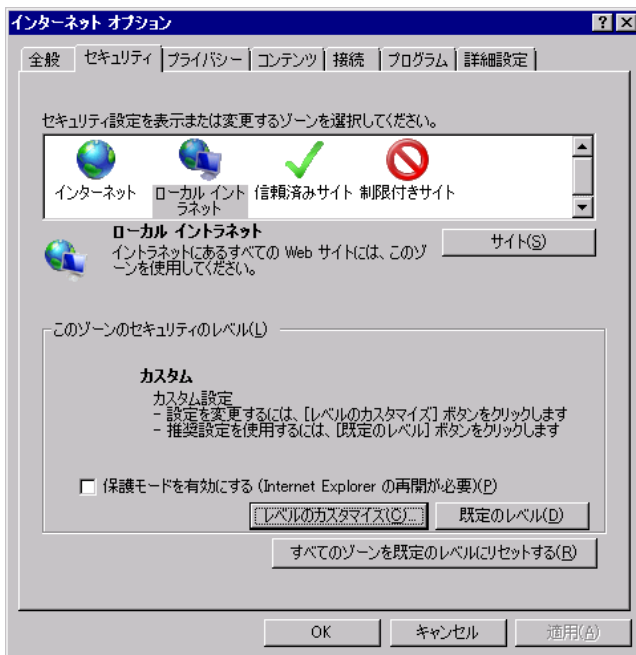
コンポーネントのバージョンについて

本製品の修正版やバージョンアップでバージョン番号が変更になる場合があります。新しいバージョンのコンポーネントをウェブページから配布する場合は `object` タグの `codebase` にあるバージョン番号を変更する必要があります。

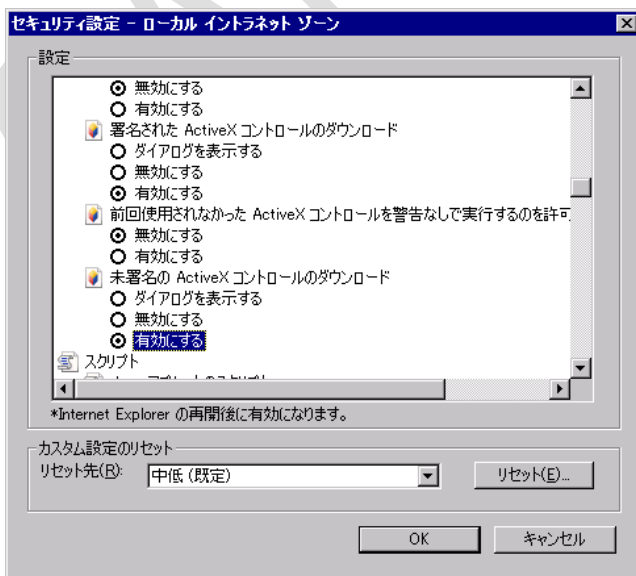
インターネットエクスプローラーの設定について

インターネットエクスプローラーが本コンポーネントをダウンロード、インストールし、ウェブページ上で動作するには、インターネットエクスプローラーの詳細設定で「未署名の ActiveX コントロールのダウンロード」を有効にします。

設定方法はインターネットエクスプローラーのメニュー「ツール」から「インターネットオプション」を選択し、「セキュリティ」タブを選択します。



ゾーンを選択して「レベルのカスタマイズ」ボタンをクリックします。



「未署名の ActiveX コントロールのダウンロード」を有効にします。通常のイントラネットでの利用であれば「前回使用されなかった ActiveX を警告なしで実行するのを

許可」も有効にすることをお勧めします。

コンポーネントの参照について

object タグの name 属性で指定した名前で javascript からコンポーネントのインスタンスを参照することができます。

```
var serial = document.forms[0].serial;
```

以下はモデムに初期化文字列を送るサンプルコードです。

```
var serial = document.forms[0].serial;
serial.Port = 1;
serial.Send("ATZ\r\n");
```

バイナリデータの送受信について

バイナリデータは javascript では文字列にバイナリデータを記述することができます。以下はバイナリデータを含むデータを送信するサンプルコードです。STX,ETX に囲まれた COMMAND=2 という文字列を送信します。

```
var serial = document.forms[0].serial;
var rc = serial.Send("¥0x02COMMAND=2¥0x3"); // STX
```

次にバイナリデータを含む文字列は string.charCodeAt() により処理することができます。以下は受信した文字列の先頭にパケットのバイト数が設定されるようなプロトコルにおけるサンプルコードです。

```
var i;
var len = serial.ReceiveByte(data); // 先頭 1 バイトを読みます。
serial.ReceiveSize = len; // 読み込むバイトサイズを指定。
var data = serial.Receive; // 長さ指定されたデータを読みます。

for (i = 0; i < data.length; i++) {
    document.forms[0].result.value += data.charCodeAt(i).toString(16);
}
```

バイナリゼロデータについても、送受信処理可能であることを確認済です。

漢字コードについて

インターネットエクスプローラの JScript では内部漢字コードは Unicode です。通常のシリアル通信で動作する機器では漢字コードは SHIFT-JIS コードが一般的ですので、本製品では漢字コードは SHIFT-JIS でシリアル通信を行います。(本コンポーネント内で UNICODE と SHIFT-JIS のコード変換をしています)

イベントによる受信処理について

インターネットエクスプローラの Jscript ではコンポーネントのイベントを受け取ることができます。最初に本コンポーネントにデータ受信をイベントで通知する設定は通信ポートを開く前に NotifyReceive プロパティを true に設定することで指定します。以下はサンプルコードです。

```
var serial = document.forms[0].serial;
serial.Port = 3;
serial.NotifyReceive = true;
var rc = serial.OpenComm();
if (rc != 0) {
    alert('COM ポートを利用できません' + rc.toString());
}
```

受信イベントは以下のように記述します。

```
<script language="javascript" type="text/javascript" for="serial"
event="OnReceive">
var serial = document.forms[0].serial;
alert(serial.Receive);
</script>
```

コンポーネントインスタンスの存在期間について

シリアル通信コンポーネントはブラウザーに html ページが表示されている間だけインスタンスが存在することができます。ページにまたがってインスタンスが存在することができないため、シリアル通信も html ページ単位で処理が完結するようにウェブアプリケーションを構成する必要があります。

長時間の通信が必要な機器を運用する場合、通信結果はその都度 Ajax など html ページ遷移が発生しない方法で、ウェブサーバーにデータを送るような設計としてください。

SAMPLE

サンプルプログラムについて

本製品には以下の2つのサンプルプログラムが提供されます。

1. バーコードリーダーサンプル

STX,ETX にバーコードデータをはさんで送るタイプのバーコードリーダーに対応したサンプルコードです。OnReceive イベントで受信したデータをフォームの TextBox に設定します。

ソースファイル名: barcode.html , barcode.js

2. ターミナルサンプル

キーボードからの入力を送信し、受信データをウェブ画面のマルチラインテキストに表示します。

ソースファイル名: terminal.html termina.js

3. ポート検査サンプル

PC で利用可能な RS-232C ポートをリストするサンプルコードです。

ソースファイル名: detect.html detect.js

マイクロソフト Visual Studio 2008 では javascript のデバッグ機能が強化されたので、これらサンプルを Visual Studio のウェブプロジェクトに追加して、javascript の動作を確認されると良いかと思えます。

プロパティリファレンス

WebMan for RS-232C のプロパティについて説明をします。

BaudRate

シリアル非同期通信の速度を設定します。以下の通信速度をプロパティに設定します。プロパティのデータ型は数値型で列挙されます。以下の値が設定可能値です。デフォルト値は8で9,600bpsとなります。

プロパティ値	ボーレート(bps)
0	75
1	110
2	150
3	300
4	600
5	1200
6	2400
7	4800
8	9600
9	14400
10	19200
11	28800
12	38400
13	57600
14	115200

また、通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。

ByteSize

7ビットまたは8ビットを指定します。プロパティのデータ型は数値型です。列挙型のプロパティで以下の値を設定します。

プロパティ値	バイト・サイズ
--------	---------

0	7bit
1	8bit

また、通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。

CustomBaudRate

通信速度は通常 BaudRate で固定値を設定しますが特定の機器に接続するような場合にはこちらのプロパティで任意の通信速度を設定して通信することが可能です。通信速度の上限・下限はご利用になるパソコンのシリアル通信チップや通信ボードの仕様により異なります。当プロパティ設定が 0 の場合は BaudRate プロパティを参照して通信速度を決定します。

DTREnable

DTR ラインの制御を指定します。プロパティを True に設定すると、通信開始時に DTR ラインをイネーブル状態に設定します。また、通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。以下はサンプル・コードです。

```
Comm.DTREnable = True
```

FlowControl

フロー制御を設定します。なし、ハードウェア、XOn/XOff が設定可能です。通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。プロパティのデータ型は列挙型で以下の値を設定します。

プロパティ値	フロー制御
0	なし
1	XON/XOFF
2	ハードウェア

ハードウェアフロー制御を選択した場合、Windows API レベルで RTS(request-to-send),DTR(data-terminal-ready)フロー制御を設定し、CTS(clear-to-send)タイムアウト、DSR(data-set-ready)タイムアウトは 30ms に設定します。

InBufferCount

プロパティを参照した時点での受信バッファ内にある受信データのバイト数を返します。

IsOpen

プロパティを参照した時点でシリアルポートがオープン状態であれば True をプログラムにもどします。

NotifyReceive

このプロパティに True を設定した場合、あたりにデータを受信すると OnReceive イベントが発生します。

また、通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次のポート・オープン時に変更が反映されます。

NotifySendComplete

送信完了イベントの発生の有無を指定します。このプロパティを True に設定した場合、データの送信が完了して送信バッファが空になると OnSendComplete イベントが発生します。また、通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次のポート・オープン時に変更が反映されます。

NullDiscard

ヌル文字の処理方法を設定します。このプロパティを True に設定するとヌル文字を受信した場合は無視され、ユーザー・プログラムには返されなくなります。また、通信ポートをオープンしている状態でこのプロパティをセットした場

合、変更は有効になりません。次回のポート・オープン時に変更が反映され
ず。

Parity

パリティ・ビットを設定します。なし、偶数、奇数が選択可能です。プロパティの
データ型は列挙型です。以下の値が設定可能です。

プロパティ値	パリティ
0	なし
1	奇数
2	偶数

また、通信ポートをオープンしている状態でこのプロパティをセットした場合、
変更は有効になりません。次回のポート・オープン時に変更が反映されます。

ParityReplace

パリティ・エラーが発生した場合に置き換える文字を指定します。デフォルト
は”?”です。通信ポートをオープンしている状態でこのプロパティをセットした場
合、変更は有効になりません。次回のポート・オープン時に変更が反映されま
す。

Port

シリアルポートを選択します。プロパティのデータ型は整数型で 1 以上の値を
指定します。

シリアルポートをオープンしている状態でこのプロパティをセットした場合、変
更は有効になりません。次回のポート・オープン時に変更が反映されます。

Receive

シリアル通信データをこのプロパティに文字列型データとして受信します。Rec
eiveSize プロパティとの組み合わせで受信するバイト数を指定して受信するこ
とができます。文字型データですがバイナリデータを含むことができます。

ReceiveByte

シリアル通信データをこのプロパティに 1 バイトの数値型データとして受信します。

ReceiveQSize

受信キュー・サイズをバイト単位で整数で指定します。通信開始前に設定される必要があります。デフォルトは 1024 バイトです。コードで設定する場合は以下ようになります。受信・キューの最大サイズはオペレーティング・システムで用意される通信デバイス・ドライバーの仕様に依存します。また、通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次のポート・オープン時に変更が反映されます。

ReceiveSize

Receive プロパティを参照してデータを受信する場合に受信する上限のバイト数を指定することができます。受信するデータがこのサイズに満たない場合は Receive 内部でブロックされます。デフォルト値は 0 です。0 指定の場合は Receive プロパティを参照した時点で受信したすべてのデータを Receive プロパティの値として戻します。

ReceiveTimeout

RecvString メソッドで文字列を受信するとき、受信バイト数をパラメータで指定した場合に受信タイムアウトをこのプロパティで指定可能です。単位はミリ・秒(1/1000 秒)です。プロパティのデータ型は Long 型です。タイム・アウトはエラー・イベントに ERR_RECV_TIMEOUT が発生し、RecvString にはその時点まで受信した文字列が返されます。49日以上連続稼働しているパソコンでは動作しない可能性もありますので、ご注意ください。¹

¹ タイム・アウトの計測に Win32API の GetThickCount を使っています。この API は 49 日でカウントがラップするので、連続稼働はできない API です。現実にはタイムアウトで 49 日以上を指定することは希と思いますが念のため記述しています。

RTSEnable

このプロパティを True に設定すると通信開始時に RTS ラインをイネーブルにします。通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。以下はサンプル・コードです。

```
Comm.RTSEnable = True
```

SendQSize

送信キューのサイズをバイト単位で整数で指定します。デフォルトは 2048 バイトです。このプロパティは通信開始前に設定される必要があります。また、通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。

SendTimeOut

送信タイムアウトを msec 単位で指定します。通信状況に依存して正確な msec 単位でのタイムアウトができない場合もありますので、あらかじめご了承ください。プロパティの値に0を指定した場合はタイムアウトはしないで送信が完了するまで送信関係のメソッドはブロックされます。

StopBits

ストップ・ビットを設定します。1,1.5,2 ビットを設定可能です。プロパティのデータ型は short です。設定は以下の対応になります。

プロパティ値	ストップビット
0	1
1	1.5
2	2

また、通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。

SAMPLE

メソッドリファレンス

ClearBreak

書式

int ClearBreak()

戻り値

メジャー通信エラー・コード。エラー・コード一覧を参照してください。

解説

通信ポートをブレイク状態から通常の通信状態にもどします。戻り値はエラー・イベントに渡される最初のパラメータ(MajorErrorCode)と同じ値です。

ClearDTR

書式

int ClearDTR()

戻り値

メジャー通信エラー・コード。エラー・コード一覧を参照してください。

解説

DTR ラインをオフにします。戻り値はエラー・イベントに渡される最初のパラメータ(MajorErrorCode)と同じ値です。

ClearRTS

書式

int ClearRTS()

戻り値

メジャー通信エラー・コード。エラー・コード一覧を参照してください。

解説

RTS ラインをオフにします。戻り値はエラー・イベントに渡される最初のパラメータ(MajorErrorCode)と同じ値です。

CloseComm

書式

int CloseComm()

戻り値

メジャー通信エラー・コード。エラー・コード一覧を参照してください。

解説

通信ポートをクローズします。VcAutoOpen プロパティが True の場合は使用できません。以下はサンプルです。戻り値はエラー・イベントに渡される最初のパラメータ(MajorErrorCode)と同じ値です。

FlushComm

通信キュー・バッファをフラッシュします。

書式

int FlushComm(int *QueueType*)

パラメータ

メソッドの動作を指定します。以下の値が指定可能です。

値	キュー・タイプ
1	実行中の送信処理を中断します
2	実行中の受信処理を中断します
4	送信キューをクリアします
8	受信キューをクリアします

戻り値

MajorErrorCode の値を返します。

解説

通信キュー・バッファのデータを破棄します。送受信両方のキューを 1 度のメソッド呼び出しで破棄する場合は値 12 を指定します。

GetCTS

書式

int GetCTS()

戻り値

CTS ラインの状態を論理値で返します。

解説

CTS ラインの状態を取得します。一般的にはラインステータスが変った時点で OnCommNotify イベントが発生しますから、このイベント中でどのラインのステータスがどのように変化したのかを取得するためにこのメソッドを使います。

GetDSR

書式

int GetDSR()

戻り値

DSR ラインの状態を論理値で返します。

解説

DSR ラインの状態を取得します。一般的にはラインステータスが変った時点で OnCommNotify イベントが発生しますから、このイベント中でどのラインのステータスがどのように変化したのかを取得するためにこのメソッドを使います。

GetRLSD

書式

int GetRLSD()

戻り値

RLSD²の状態を論理値で返します。

解説

RLSD 信号の状態を取得します。

GetRing

書式

```
int GetRing()
```

戻り値

Ring ラインの状態を論理値で返します。

解説

Ring ラインの状態を取得します。一般的にはラインステータスが変った時点で OnCommNotify イベントが発生しますから、このイベント中でどのラインのステータスがどのように変化したのかを取得するためにこのメソッドを使います。

IsPortAvailable

書式

```
bool IsPortAvailable(int PortNo)
```

戻り値

シリアルポートがシステムで利用可能であれば True を返します。

解説

VcAutoOpen プロパティが False の場合に、コードで明示的に通信ポートをオープンする場合に使います。以下はサンプル・コードです。戻り値はエラー・イベントに渡される最初のパラメータ(MajorErrorCode)と同じ値です。

OpenComm

書式

```
int OpenComm()
```

² receive-line-single-detect

戻り値

メジャー通信エラー・コード。エラー・コード一覧を参照してください。

解説

シリアルポートをオープンし、通信可能な状態とします。

Send

書式

```
int Send(string data);
```

パラメータ

data

送信するデータ

戻り値

メジャー通信エラー・コード。エラー・コード一覧を参照してください。

解説

データを送信します。文字列型データを指定した場合は複数バイトを送ります。整数型データを指定した場合は、1 バイトのみ送信します。整数型データとして指定できるのは 0～255 の範囲となります。配列データを指定することはできません。(Jscript がメソッドにパラメータを配列として渡すことをサポートしないため)

SendBreak

書式

```
int SendBreak();
```

戻り値

メジャー通信エラー・コード。エラー・コード一覧を参照してください。

解説

通信ポートをブレイク状態にします。以下はサンプルです。ブレイク状態の解除には ClearBreak メソッドを使用します。戻り値はエラー・イベントに渡される最初のパラメータ(MajorErrorCode)と同じ値です。

SetRTS

書式

int SetRTS()

戻り値

メジャー通信エラー・コード。エラー・コード一覧を参照してください。

解説

RTS ラインをオンにします。

Sleep

書式

void Sleep(int ms)

パラメータ

ms 待ち時間(ミリ秒)

戻り値

なし。

解説

シリアル通信には直接の関係はありませんが接続機器によって必要になるスリープ処理を実行します。

TransmitChar

書式

int TransmitChar (string *aChar*)

パラメータ

送信する1文字。

戻り値

メジャー通信エラー・コード。エラー・コード一覧を参照してください。

解説

1バイトデータを送信します。指定された文字は、送信バッファの先頭に置かれます。

SAMPLE

イベントリファレンス

OnCommError

シリアル通信エラーの通知にイベントを発生させます。エラー・イベントは以下の形式です。

CommError(*MajorErrorCode* As Integer, *MinorErrorCode* As Long)

MajorErrorCode

主エラーコード。エラーの詳細は Appendix-A にあります。

MinorErrorCode

詳細情報エラー・コード。送受信時のエラーについて詳細情報が通知されます。Win32 API の GetLastError から値です。

サンプルコード

以下はウェブページ内でのエラーイベント処理サンプルです。

```
<script language="javascript" type="text/javascript"
for="serial" event="OnCommError (p1, p2)">
  handle_errors (p1, p2);
</script>
```

OnReceive

プロパティ NotifyReceive が true に設定された状態で通信した場合、データを受信すると OnReceive イベントが発生します。

サンプルコード

以下はウェブページ内でのイベント受信サンプルです。

```
<script language="javascript" type="text/javascript"
for="serial" event="OnReceive">
```

```
var serial = document.forms[0].serial;  
var data = serial.Receive;  
alert(data);  
</script>
```

OnSendComplete

プロパティ NotifySendComplete に True の値が設定されており、送信キューが空になった時に OnSendComplete イベントが発生します。

Appendix-A エラーコード

以下はエラー・イベント・プロシージャに通知される最初のパラメータ(MajorErrorCode部分)の説明です。

ERR_OPEN	100	通信ポートがオープンできません。
ERR_BUILD_DCB	101	Data Control Blockが作成できませんでした。
ERR_COMM_STATE	102	通信状態エラー。詳細はMinorErrorCodeを調べてください。
ERR_NO_MEM	103	メモリが不足しています。
ERR_BUFFER_SHORT	104	受信するキューのサイズが小さい。キュー・サイズを大きくしてください。
ERR_READ_COMM	105	通信ポートから読み込みません。詳細はMinorErrorCodeを調べてください。
ERR_WRITE_COMM	106	通信ポートに出力できません。詳細はMinorErrorCodeを調べてください。
ERR_CLEAR_BREAK	107	ブレイク状態をクリアできません。
ERR_SET_BREAK	108	ブレイク状態に移行できません。
ERR_TRANSMIT_CHAR	109	SendCharに失敗しました。
ERR_INVALID_SIZE	110	SendCharで文字列が指定されました。
ERR_NOT_OPEN	111	通信ポートがオープンしていません。
ERR_ALREADY_OPEN	112	2度通信ポートのオープンを試みました。
ERR_INVALID_DEVICE_NAME	113	通信ポートの指定が不正です。
ERR_FLUSH_COMM	114	通信キューの廃棄に失敗しました。
ERR_RECV_TIMEOUT	115	受信タイムアウト
ERR_CREATE_EVENT	116	イベントの作成に失敗しました。
ERR_RECV_LENGTH_TOO_LONG	117	受信長プロパティが受信キューサイズより大きいです。
ERR_THREAD	118	スレッドの作成に失敗しました。
ERR_CLEAR_DTR	119	DTRのクリアに失敗しました。
ERR_SET_DTR	120	DTRラインをセットできません。
ERR_CLEAR_RTS	121	RTSラインをクリアできません。
ERR_SET_RTS	122	RTSラインをセットできません。
ERR_GET_MODEM_STATUS	123	モデムの状態を取得することに失敗しました。
ERR_COMM_LINE	124	ライン状態を取得することに失敗しました。

ERR_SEND_TIMEOUT	127	送信タイムアウトです。
ERR_TYPE_INVALID	128	送信メソッドに指定したデータの型はサポートされていません。文字列、整数型のみのサポートとなります。
ERR_NO_DATA	129	メソッド等の呼び出し時点で通信バッファにはデータが存在しませんでした。

SAMPLE

Appendix-B サンプル javascript コード

以下はバーコード読み込みサンプルコードです。

```
var STX = 2;
var ETX = 3;

window.onload = function() {

    var serial = document.forms[0].serial;
    serial.Port = 3; // 接続ポートに合わせて変更してください。
    serial.NotifyReceive = true;
    serial.ReceiveTimeOut = 1000;
    var rc = serial.OpenComm();
    if (rc != 0) {
        alert('COM ポートを利用できません' + rc.toString());
        return;
    }
    document.getElementById("message").innerText = 'ポートオープンし
ました';
}

var receive_barcode = function() {

    var serial = document.forms[0].serial;
    var result = '';
    var ch = serial.ReceiveByte;

    if (serial.LastMajorErrCode != 0) {
        return;
    }

    // テキストクリア
    document.forms[0].barcode.value = '';

    // STX を読み込むまで待機。
    while (ch != STX) {
        ch = serial.ReceiveByte;
        if (serial.LastMajorErrCode != 0) {
            return;
        }
    }
}
```

```
    }  
  }  
  
  // ETX まではバーコード値。  
  ch = serial.ReceiveByte;  
  
  while (ch != ETX) {  
    result += String.fromCharCode(ch);  
    ch = serial.ReceiveByte;  
    if (serial.LastMajorErrCode != 0) {  
      return;  
    }  
  }  
  
  // 結果をテキストに設定  
  document.forms[0].barcode.value = result;  
}  
  
var handle_errors = function(majorErr, minorErr) {  
  document.getElementById("message").innerText = 'エラー' +  
  majorErr.toString();  
}
```

WebMan for RS-232C 調査依頼

日付	
会社名	
登録ユーザー名	
製品シリアル番号	
製品バージョン	
電話番号	
ファックス番号	
電子メールアドレス	
使用パソコン機種	
ホスト言語とバージョン	
OSバージョン	
開発環境バージョン	
お問合わせ内容、問題記述など、具体的に再現可能なようにご記入ください。	
添付資料	

SAMPLE

WebMan for RS-232C version 1.00
プログラミング解説書

第1版

2008年12月22日

版權・著作 株式会社テクナレッジ
Printed In Japan