

## EasyGPIB for GP232

このマニュアルはActivecellで販売しているGPIB/  
シリアルアダプタGP232を使用した専用の機能を含  
めて解説しています。  
他のコントローラと共通の機能だけを使用する場  
合は、EasyGPIBのマニュアルを参照してください。

## EasyGPiB for GP232 とは

ご利用に当たっては「動作環境」と「考え方」の部分だけは必ず読んでください。

EasyGPiB for GP232 をダウンロードしていただきましてありがとうございました。  
以下, EasyGPiB for GP232をeg4GP232と略記します。

eg4GP232は, ActiveCell(<http://www.activecell.jp>)で販売しているGPiB/シリアルアダプタGP232用の共通化モジュールです。

ファイル名はバージョンによって異なりますが, インポートしたときのモジュール名はegDefになります。インポートの手順などは後述します。

本書はGP232専用の機能を含めて解説しております。

その機能を使用すると, 他社製のGPiBコントローラカードとの互換がなくなるので注意して下さい。

共通の機能だけを使用するときは, EasyGPiBのマニュアルを参照してください。

EasyGPiBはVisualBasicだけではなく, Office97(Windows版)以降のVBAに対応しているので, Excel97, 2000やAccess, Word, Outlookなどで使用できますが, ここではExcelで使用する手順を説明します。

木下隆

## 動作環境

OS: Windows95以降

Office: Office97以降

VB: VB4.0以降

## 特徴

- (1) Excel, Access, Word, Outlookなどのオフィスアプリケーションに付属するVBAで動作します。また, VisualBasicでも動作します。  
ただしVB.netには対応しておりません。  
VBA:Office97以降, VisualBasic:4以降
- (2) eg4GP232はVBAからWindowsの標準APIをコールしています。専用のApiやインストール作業などは一切必要ありません。
- (3) 共通化モジュールを入手すればカードメーカーによらない共通のプログラムが作成できます。  
ただし, 固有の機能を使用したときは互換がなくなります。
- (4) クラスモジュールと標準モジュールの表記方法を併用しているため, バランスが取れた直感的で活用しやすい仕様になっています。
- (5) シリアルポート制御ツールEasyCommと同時に使用できます。
- (6) コーディング中に自動的にメンバが表示されるので, 間違いが少なく効率的にプログラミングができます。
- (7) 各モジュールはコードが書かれたテキストファイルです。閲覧・変更・引用など自由に行うことができます。
- (8) デリミタの許容範囲が広いので, 誤った設定でも通信できる条件が広がりました。
- (9) 基本的な機能しかサポートしていないので, 高度なアプリケーション, または高速なアプリケーションを作成するのは困難です。
- (10) イベントには対応しておりません(すべて同期処理)。
- (11) オンリモードを持っているので, 1:1の通信ならば非常に簡単なプログラムを書くことができます。
- (12) 2本の未使用ポートを入出力I/Oビットとして利用できます。

eg4GP232利用までの手順

eg4GP232を利用するに当たって、インストールなどの作業は一切必要ありません。お使いのパソコンやExcelなどの設定を一切変更せずに、簡単に利用することができます。

必要なものは、シリアルポートがあるパソコンと、Excelなどに付属しているVBA、そしてGP232です。

GP232はUSB/シリアルアダプタでも動作しますが、USB-HUBを通す場合はUSB/シリアルアダプタの消費電流が大きく、バスパワーでは動作しないことがあるので注意してください。

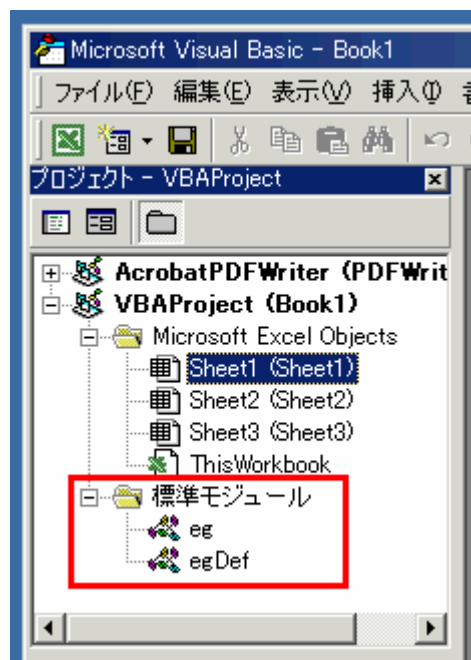
eg4GP232は主要処理を記述したモジュールと、共通化モジュールの二つのテキストファイルで構成されます。

利用までの手順は次の通りです。

- (1) 同梱されているモジュールを探しやすいフォルダ(My Documentsなど)におきます。
- (2) Excelを起動し、Alt + F11をタイプして、VBEを起動します。
- (4) Ctrl+Mをタイプします。
- (5) 同梱されているモジュールの中からeg-V122.basをダブルクリックします。
- (6) (4)、(5)を繰り返して、インタフェースの標準化モジュール  
Activecell-GP232-V123.basをインポートします。  
インポートされると、プロジェクトエクスプローラにモジュールが表示されるので確認してください。  
標準モジュールにはegとegDefの二つのモジュールがインポートされます。  
ファイル名とモジュール名は同じではありません。  
ファイル名は変更してもかまいませんが、モジュール名は変更しないでください。
- (7) インポートしたブックは名前を付けて保存してください。  
以上でモジュールのインポートは完了です。  
保存したブックを開けばeg4GP232が利用できます。  
これ以外のファイルは必要ありません。

**\* 注意**

インポートは、そのブックにモジュールを読み込む作業です。他のブックでeg4GP232を使用するときは、そのブックにインポートしなければなりません。  
インポートしてもExcel自体に組み込まれることはありません。



## 考え方

eg4GP232は機能を限定すると共に、独自の考え方を元に設計しました。  
これは特に重要なので、ここだけは必ず読んでからご使用ください。

### プログラムの共通化

eg4GP232は各社のGP232コントローラカードによらず、共通のプログラムを書くことができます。ただし、GP232固有の機能を使うと互換がなくなるので注意してください。

### アクティブアドレスという考え方

eg4GP232の各コマンドは、対象となる機器のアドレスを指定する機能があります。GP232の常識であるひとつのトーカーと複数のリスナーという概念を排除し、パソコンと相手機器とが1対1で通信するという考えのもとに構築されています。

パソコンが通信する相手のアドレスをアクティブアドレスと呼び、ActiveAddressプロパティにアドレスを代入して指定します。

<例> アドレス 2 の機器を相手機器に指定

eg.ActiveAddress = 2

一度指定すると、ActiveAddressを次に変更するまで有効で、その間に実行されるすべてのコマンドはActiveAddressで指定した機器に対して行われます。  
アクティブアドレスはeg4GP232を利用するためのもっとも重要な概念です。

### オンリモード

eg4GP232には、GP232固有の機能があります。

それはオンリモードです。

このモードはパソコンと1台の測定器を単方向で接続するために用意されたものです。

たとえば、このモードでデジタルマルチメータ(DMM)1台を接続する場合を考えてみましょう。

DMMはトークオンリモードに設定されており、測定するたびにデータを送ろうとします。gp4GP232をオンリモードで開き、読み出しを行うとDMMが出力しようとしていたデータを直ちに読み取ることができます。

また、プロッタをリスンオンリモードで接続します。

同様にオンリモードで開いて、書き込みを行うとプロッタに出力することができます。

いずれもアドレス指定を行わずに簡単にデータを取得、または出力できます。

測定器の中にはトークオンリでないと動作しないものがあります。

そのような場合は、オンリモードでデータを取得してください。

### 対応していないもの

eg4GP232は利用する立場から生まれたフリーツールです。

利用しにくい高度な機能や見かけの多機能性を強調するための機能は一切含まれていません。逆に、使いやすさのためならば、多少の機能はばっさりカットしてしまおうという考えが根底にあります。

そのため、GPiBの対応していない機能がいくつかありますのでご了承ください。

**eg4GP232はコントローラ専用です。**

スレーブに対応していないカードがあることや、機能が複雑になる点を考えてeg4GP232はコントローラ専用にしました。

ただし、GP232を使用することを前提にオンリモードを設けました。

**パラレルポートには対応しません。**

パラレルポートには対応していないコントローラカードや機器が多いので、eg4GP232も対応していません。

**イベントには対応していません。**

対応させるためには、一部のカードを切り捨てる必要があることや、Office97に非対応になるなどの理由で対応しておりません。

eg.CardOpenとeg.CardCloseは必ず実行してください。

初期設定処理と終了処理は必ず実行してください。

オンリモードのときは、eg.CardOpenではなく、egDef.GP232\_OnlyModeOpenを使用してください。

なお、カードのクローズは同じです。



(4) アクティブアドレスの指定

eg4GP232は対象となるひとつの機器と通信を行います．これから通信しようとする機器のアドレスをActiveAddressプロパティに代入します．たとえば，アドレス 9 の機器に対して送受信を行う場合は次のように書きます．

eg.ActiveAddress = 9

オンリモードでオープンしたときはActiveAddressの指定や変更をしないで下さい．

ちなみに，オンリモードではActiveAddressは-1になります．

ActiveAddressはいつでも変更できます．また読み出すこともできます．

ActiveAddressがゼロのときは，未設定として判断されます．

ですから，接続機器のアドレスをゼロに指定することはできません．

なお，コントローラ自身のアドレスはeg.CMDs.MAで取得できます．

大概是ゼロですが，ヤマトエレクトロニクスのカードでは2 1 (10進)に固定されているので注意してください．

ActiveAddressはひとつしか指定できないので，複数の機器に対してデータを同時に送信することはできません．



## (5) デリミタの設定

デリミタは必ず設定してください。

オンリモードでも指定する必要があります。

例えば、測定器がCr+Lf、パソコンがCrのみに設定されているとします。

測定器から送られてくるデータを受信すると、パソコンは最初のCrをデリミタと判断して受信を終了します。

測定器は最後のLfの送信が出来ないため、先に進めないままロックしてしまいます。

このとき、通常はタイムアウトで停止します。

次のようなケースもあります。

機器側が Cr + Lf + EOI , パソコンが Crに設定されていたとします。

機器がデータを送信し、デリミタのCrを送信した時点でパソコン側はデリミタを受信したものと判断して受信処理を終了します。

機器側では、最後のLfを未送信のまま待機します。

それ以降の状態は機器の動作によりますが、パソコンが次のデータを要求した時点で Lf + EOI が出力されるとパソコンはEOIをデリミタと判断して、空のデータを受信することになります。

また、機器がタイムアウトになり、エラーコードを送信するかもしれません。

いずれにしても正しい送受信は困難な状態になります。

例としてC R + E O I に設定するときは次のように書きます。

```
eg.Delimiter = eg.DELIMs.CR
```

このように設定すると、文字列の送信の時にC R が付加され、同時にE O I ラインがアクティブになります。

受信処理はカードによってさまざまな認識をしますが、実用上は問題にならないように考慮してあります。

受信文字列には受信したデータのデリミタとNullが削除されます。

## (6) 文字列の送受信

文字列の送受信はAsciiLineプロパティを使用します。

AsciiLineに書き込めば送信し、読み出せば受信した文字列を取得できます。

たとえば次の例はアドレス9の機器に"\*TRG"という文字列を送ったあと、その機器から送られて来た文字列を受信してA1セルに書き込みます。

```
eg.CardOpen
eg.ActiveAddress = 9
eg.Delimiter = eg.DELIMs.CrLf
eg.AsciiLine = "*TRG"
Range("A1") = eg.AsciiLine
eg.CardClose
```

## (7) エラー

eg4GP232はトラップ可能なエラーを発生しません。

エラーが発生しても何事もないかのように処理を続けますが、エラーが発生したかどうかを確認するための5個のプロパティを備えており、このプロパティを読み出すことによってエラーの状況を知ることができます。

eg.ErrorNumber	直前に発生したエラー番号で、正しく処理が行われたときはゼロにリセットされます。
eg.ErrorHold	前回発生したエラー番号で、以降処理が正しく行われても変化しないのでゼロを代入することによって強制的にリセットします。
eg.ErrorHoldText	前回発生したエラー番号に対応するメッセージを文字列で返します。
eg.LocalErrorNumber	標準化モジュールで発生したエラー番号を返します。エラー番号は各カードのapiが返す値です。
eg.LocalErrorText	標準化モジュールで直前に発生したエラー番号に対するメッセージをテキストで返します。

～ はすべてに共通ですが、あまり詳しい情報を得ることはできません。  
 , は使用するカードごとに異なりますが、発生したエラーの詳細を知ることができます。

eg.ErrorHoldは一度エラーが発生すると、次にエラーが発生するまでその値を保持します。

eg.ErrorHoldにゼロを代入して、いくつかの処理をエラーチェック無しで行った後、最後にeg.ErrorHoldをチェックすることによって、総合的なエラー発生の有無を知ることができます。

## (8) 終了処理

終了処理は必ず実行してください。

eg.CardClose

(9) プログラム例 (通常)

```
Sub GetGP()
    eg.CardOpen
    eg.ActiveAddress = 9
    eg.AsciiLine = "DL0,H0,S1,M0,PR2,F1,R0"
    Range("A1") = eg.AsciiLine
    eg.CardClose
End Sub
```

アドバンテスト製のデジタルマルチメータ(以下DMMと略記します)から測定データを取得してExcelのワークシートのA1セルに書き込むプログラムです。  
DMMはアドレスサブルモードで、アドレスを9に設定しているものとします。  
動作テストは、手持ちのR6551で行いましたが、同じコマンド体系を持つ他の機種でも動作します。

マニュアル上で調査した結果、動作すると思われる機種は次の通りです。

R6441シリーズ  
R6451シリーズ  
R6452シリーズ

プログラムの説明

**eg.CardOpen**

カードをオープンします。

eg4GP232の処理を開始する前に必ず実行する必要があります。

**eg.ActiveAddress = 9**

対称なる測定器のアドレスを指定します。

eg.ActiveAddressで指定した機器が処理の対象となります。

**eg.AsciiLine = "DL0,H0,S1,M0,PR2,F1,R0"**

測定器にコマンド文字列を送信します。

アドバンテストのコマンドの意味は次の通りです。

DL0: デリミタをCr+Lf+E0Iに設定します。

H0: 返送してくるデータの先頭にファンクションなどを示す文字列を付加しないように指定します。

S1: SRQの発信を禁止します。

M0: フリーランモードに設定します。

PR2: サンプリングレートを「中」に設定します。

F1: ファンクション1に設定します。

R6551では直流電圧測定になります。

R0: 測定レンジを指定します。

R6551ではオートレンジになります。

**Range("A1") = eg.AsciiLine**

測定器からデータを読み込み、アクティブシートのA1セルに書き込みます。

**eg.CardClose**

カードを閉じます。

処理が完了したら必ず実行する必要があります。

(10) プログラム例（GP232固有のオンリモード）

同じ機能をGP232のオンリモードを使用すると次のようになります。

```
Sub GetOnly()  
    egDef.GP232_OnlyModeOpen  
    Range("A1") = eg.AsciiLine  
    eg.CardClose  
End Sub
```

測定器はトークオンリに設定する必要があります。

測定器はトーカのままなのでパソコンからコマンドやデータを送ることはできません。そのため、測定レンジやモードは、測定器のパネルなどで手動設定する必要があります。

著作権

Excelは、米マイクロソフト社の商標です。

Copyright(c) 2002-2004 T.Kinoshita

eg4GP232はGP232ご購入者に限り無償でダウンロードし、無償で 사용할 수 있습니다. 또한, 회사의 제품 일부로서 판매할 수 있지만, eg4GP232를 단독으로 판매할 수 없습니다.

프로그램은 자유롭게閲覽, 변경, 引用할 수 있지만 저작권은木下隆가 소유하고 있으므로 良識의 범위에서 이용하십시오.

なお、慣例により、本プログラム、ブックをご利用になった結果につきましては一切責任を負わないことをご了承ください。

URL <http://www.activecell.jp>

e-mail [k2@activecell.jp](mailto:k2@activecell.jp)

木下隆

EasyGPIB for GP232  
リファレンス

## プロパティとメソッド一覧（上段は通常，下段はオンリモード）

プロパティ	書き込み	読み取り
eg.ActiveAddress	対象になる機器のアドレスを指定．	現在対象になっているアドレスを取得．
	書き込まないこと．	- 1 が返る．
eg.AsciiLine	書き込んだ文字列にデリミタを付加してActiveAddressで指定された機器に送信．	ActiveAddressで指定された機器からデリミタまでの文字列を受信して返す．
	書き込んだ文字列にデリミタを付加して送信．	デリミタが検出されるまで受信した文字列を返す．
eg.Binary	書き込んだバイナリデータをActiveAddressで指定された機器に送信．	ActiveAddressで指定された機器からバイナリデータを受信して返す．
	書き込んだバイナリデータを送信．	E0Iが検出されるまで受信したバイナリデータを返す．
eg.Cmd	書き込んだデータをATNをLowに下げてコマンドとして送信．	<書き込み専用>
	通常モードと同じ	
eg.CMDs	<読み取り専用>	GPiBのコマンドバイトを取得．
	通常モードと同じ	
eg.Delimiter	送受信のデリミタを文字列で指定．	現在設定されているデリミタを文字列で取得．
	通常モードと同じ	
eg.DELIMs	<読み取り専用>	デリミタ文字列を取得．
	通常モードと同じ	
eg.Sound	WAV形式のサウンドファイルを，非同期で再生	<書き込み専用>
	通常モードと同じ	
eg.SPoll	ActiveAddressで指定した機器にシリアルポールを実行してステータスパイトを返す．	
	実行しても何も処理を行いません．	
eg.SRQ	<読み取り専用>	バスのSRQラインの状態をTrue,Falseで取得．
	通常モードと同じ	
eg.Timeout	GPiBのタイムアウトを秒単位で指定．	現在設定されているタイムアウトを秒単位で取得．
	通常モードと同じ	

## プロパティとメソッド一覧（上段は通常，下段はオンリモード）

プロパティ	書き込み	読み取り
<a href="#">eg.ErrorNumber</a>	直前に発生したエラー番号を強制的に指定．	直前に発生したエラー番号を取得．
	通常モードと同じ	
<a href="#">eg.ErrorHold</a>	前回発生したエラー番号を強制的に指定． 使用前にゼロを書き込んでリセットす	前回発生したエラー番号を取得．
	通常モードと同じ	
<a href="#">eg.ErrorHoldText</a>	<読み取り専用>	前回発生したエラーの内容を文字列で取得．
	通常モードと同じ	
<a href="#">eg.LocalErrorNumber</a>	直前に発生したapiのエラー番号を強制的に指定．	直前に発生したapiのエラー番号を取得．
	通常モードと同じ	
<a href="#">eg.LocalErrorText</a>	<読み取り専用>	前回発生したapiエラーの内容を文字列で取得．
	通常モードと同じ	
<a href="#">eg.WAITmS</a>	指定した時間（ミリ秒単位），処理を中断	<書き込み専用>
	通常モードと同じ	
<a href="#">eg.Version</a>	<読み取り専用>	EasyGPIBと読み込まれている共通化モジュールの種類とバージョンを取得．
	通常モードと同じ	
<a href="#">egDef.GP232_10bit(BitName)</a>	True: 該当ビットにHighを出力 False: 該当ビットにLowを出力	該当ビットの入力状態を取得
	通常モードと同じ	

メソッド	動作
<a href="#">eg.CardOpen</a>	通常モードでカードを開き，変数のイニシャライズを行う．
<a href="#">egDef.GP232_OnlyModeOpen</a>	オンリモードでカードを開き，変数のイニシャライズを行う．
<a href="#">eg.CardClose</a>	カード，および変数の終了リセット処理を行う．
	通常モードと同じ



**通常モード**

eg.CardOpen でオープンしたときのモードです。  
他社製のカード用のEasyGPIBと互換があります。

**書き込み**            **整数(Integer)**  
操作の対象となる機器のアドレスを代入します。

**読み取り**           **整数(Integer)**  
現在操作の対象となっている機器のアドレスを取得します。

**解説**

EasyGPIBはデータの送受信やシリアルポールの対象となる機器のアドレスを、ひとつだけ指定できます。

そのアドレスをアクティブアドレスと呼び、ActiveAddressプロパティで指定します。たとえば9番のアドレスを持つ機器とデータの送受信を行うためには、あらかじめActiveAddressプロパティに9を代入しておかなければなりません。

```
eg.ActiveAddress = 9
```

また、ActiveAddressを読み出すことによって、現在アクティブなアドレスを知ることができます。

一時的に特定の機器に対してデータを送信するときに、現在設定されているアクティブアドレスを一時的に保存するときなどに使用します。

```
SavedAddress% = eg.ActiveAddress    ' アクティブアドレスの退避  
eg.ActiveAddress = 3                ' アクティブアドレスの一時変更  
eg.AsciiLine = "*TRG"               ' 文字列の送信  
eg.ActiveAddress = SavedAddress%    ' アクティブアドレスの復帰
```

**バスライン**

ActiveAddressは内部変数进行操作するだけなのでバスラインには影響がありません。

**使用例**

```
eg.ActiveAddress = 8                ' アドレス8を指定  
eg.AsciiLine = "E"                 ' "E"を送信します  
Range("A1") = eg.AsciiLine        ' データを受信してA1セルに書き込みます  
eg.ActiveAddress = 9                ' アドレス9を指定  
eg.AsciiLine = "E"                 ' "E"を送信します  
Range("A2") = eg.AsciiLine        ' データを受信してA2セルに書き込みます
```

**注意**

ActiveAddressは必ず指定してください。

ActiveAddressがゼロのときは未設定として解釈されるので、ゼロ以外の値で使用してください。

一度指定すれば、次に指定するまで有効です。

デリミタの設定はActiveAddressの指定とは連動しません。

異なったデリミタの複数の機器と通信するときは、その都度Delimiterプロパティを切り替えてください。

## &lt; 例 1 &gt;

```
eg.ActiveAddress = 2
eg.Delimiter = eg.DELIMs,CrLf      アドレス2の機器はCr+Lf+Eoi
eg.AsciiLine = "*Trg"
eg.ActiveAddress = 3
eg.Delimiter = eg.DELIMs,LfNoEoi   アドレス3の機器はLfのみ
eg.AsciiLine = "*Trg"
```

## &lt; 例 2 &gt;

アドバンテストのデジタルマルチメータ6451のファンクションを設定します。  
アドレスは 2 に設定されているものとします。

**オンリモード**

egDef.GP232\_OnlyModeOpen でオープンしたときのモードです。  
他社製のカードとの互換はありません。

書き込み          整数(Integer)

オンリモードではアドレスをしてできません。

書き込みは行わないでください。

読み取り          整数(Integer)

- 1 が返ります。

**解説**

オンリモードはアドレス指定ができません。

そのため、ActiveAddressへの書き込みは行わないで下さい。

システム内ではオンリモードでのエラー回避のためにActiveAddressを-1に設定しています。

変更すると正しく処理できなくなるので絶対に書き込まないで下さい。

**通常モード**

eg. CardOpen でオープンしたときのモードです。  
他社製のカード用のEasyGPIBと互換があります。

書き込み          文字列(String)

ActiveAddressプロパティで指定された機器へデリミタを付加して代入した文字列を送信します。

読み取り          文字列(String)

ActiveAddressプロパティで指定された機器から、デリミタを受信するまでの文字列を返します。ただしデリミタは削除されます。

**解説**

ActiveAddressプロパティで指定された機器に対して、文字列を送受信するプロパティです。

書き込み処理では書き込んだ文字列にデリミタを付加して送信し、読み取り処理ではActiveAddressから送られてきた文字列を受信して返します。

受信文字にはデリミタは含まれません。

送受信はバッファを通して行われるため、あらかじめ十分なサイズのバッファを確保しておく必要があります。

バッファについてはCardOpenメソッドを参照してください。

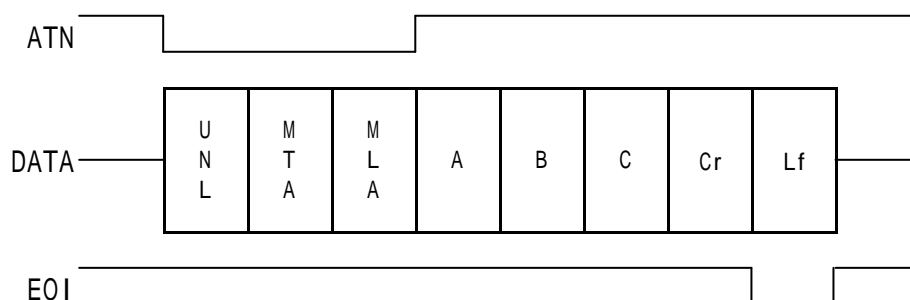
パソコンと接続機器のデリミタの設定が異なっていると、いくつかの問題が発生します。

たとえば、相手機器のデリミタがC rのみでE O Iは対応していない場合、パソコンのデリミタがL fのみに設定されているとパソコンは相手機器のデリミタ(L f)を待ち続けるのでタイムアウトが発生します。

相手機器とパソコンのデリミタは合わせるように設定してください。

**バスライン**

- AsciiLineプロパティに"ABC"を書き込んだとき  
(ActiveAddressは9、デリミタはCrLfに指定されているものとします)



使用するカードによっては異なったコマンドが出力されるものがあります。

## 使用例

```

Sub GetAscii()
    eg.CardOpen          ' カードのオープン
    eg.Delimiter = eg.DELIMs.CrLf ' デリミタの設定
    eg.ActiveAddress = 9   ' アクティブアドレスの設定
    eg.AsciiLine = "F1,R0,M1" ' ファンクションなどの送信
    eg.AsciiLine = "E"      ' トリガ
    Range("A1") = eg.AsciiLine ' 受信とセルへの書込み
    eg.CardClose          ' カードのクローズ
End Sub

```

## 注意

あらかじめ、ActiveAddressを必ず指定してください。

バイナリデータを送受信するときはBinaryプロパティを使用してください。

AsciiLineプロパティの書込みと読出しはUniCodeで行われますが、 GPIBバスで入出力されるコードはANSIに変換されます。

末尾のデリミタ文字とNullは削除されます。

## オンリモード

egDef.GP232\_OnlyModeOpen でオープンしたときのモードです。

他社製のカードとの互換はありません。

書き込み          文字列(String)

デリミタを付加して代入した文字列を送信します。

読み取り          文字列(String)

デリミタを受信するまでの文字列を返します。ただしデリミタは削除されます。

## 解説

文字列の送信、及び受信を行うプロパティです。

書き込み処理では書き込んだ文字列にデリミタを付加して送信し、読み取り処理ではデリミタが検出されるまで文字列を受信して返します。

受信文字にはデリミタは含まれません。

送受信はバッファを通して行われるため、あらかじめ十分なサイズのバッファを確保しておく必要があります。

バッファについてはGP232\_OnlyModeOpenメソッドを参照してください。

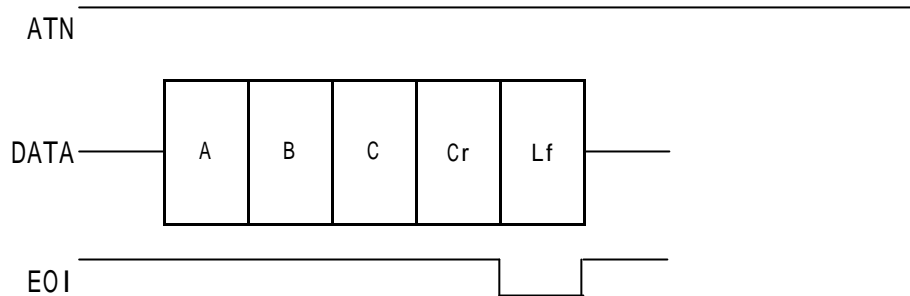
パソコンと接続機器のデリミタの設定が異なっていると、いくつかの問題が発生します。

たとえば、相手機器のデリミタがC rのみでE O Iは対応していない場合、パソコンのデリミタがL fのみに設定されているとパソコンは相手機器のデリミタ(L f)を待ち続けるのでタイムアウトが発生します。

相手機器とパソコンのデリミタは合わせるように設定してください。

## バスライン

- AsciiLineプロパティに"ABC"を書き込んだとき  
(デリミタはCrLfに指定されているものとします)



## 使用例

```
Sub GetAsciiOnly()
    egDef.GP232_OnlyModeOpen      ' カードのオープン
    eg.Delimiter = eg.DELIMs.CrLf ' デリミタの設定
    Range("A1") = eg.AsciiLine    ' 受信とセルへの書込み
    eg.CardClose                  ' カードのクローズ
End Sub
```

## 注意

ActiveAddressは変更しないで下さい。

バイナリデータを送受信するときはBinaryプロパティを使用してください。

AsciiLineプロパティの書込みと読出しはUniCodeで行われますが、GPIBバスで入出力されるコードはANSIに変換されます。

末尾のデリミタ文字とNullは削除されます。

## 通常モード

eg.CardOpen でオープンしたときのモードです。  
他社製のカード用のEasyGPIBと互換があります。

書き込み          バイト型(Byte)，整数(Integer)，長整数(Long)，単精度浮動小数点(Single)，倍精度浮動小数点(Double)  
上記型の配列

Binaryプロパティにバイナリデータを代入すると，ActiveAddressプロパティで指定された機器へバイナリデータを送信します。

読み取り          バイト型配列( Byte() )  
ActiveAddressプロパティで指定された機器から，バイナリデータを受信して返します。

## 解説

ActiveAddressプロパティで指定された機器に対して，バイナリデータを送受信するプロパティです。

書き込み，読み出し共にデリミタの設定は無視され，EOIのみが有効になります。送受信はバッファを通して行われるため，あらかじめ十分なサイズのバッファを確保しておく必要があります。

書き込むことができるデータ型は5種類ありますが，いずれも1バイトの範囲，0～255を超えるとエラーになります。

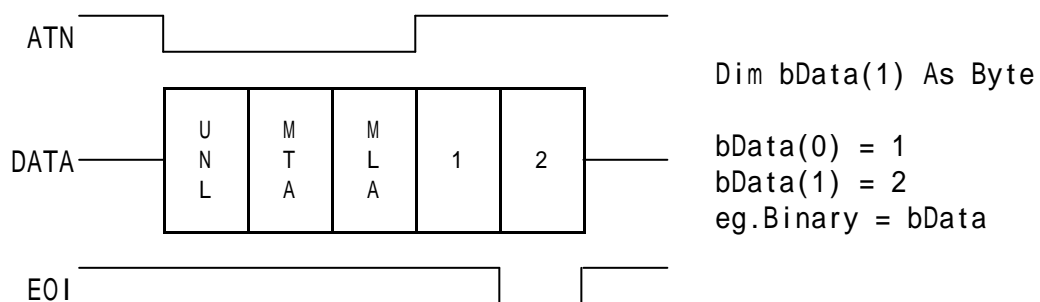
複数バイトのバイナリデータを送信するときは，配列を使います。

また，受信データはバイト配列として取得します。

1バイトのバイナリデータであってもバイト配列で返ります。

## バスライン

- ・ BinaryプロパティにbData()を書き込んだとき  
(ActiveAddressは9，デリミタはCrLf，bData(0)=1,bData(1)=2とします)



使用するカードによっては異なったコマンドが出力されるものがあります。

## 使用例

次のコードはテクトロニクス製ストレージオッシロTDS210と接続してGPIBを通して画面キャプチャを画像として取り込むものです。

< 条件 >

機器アドレス：12

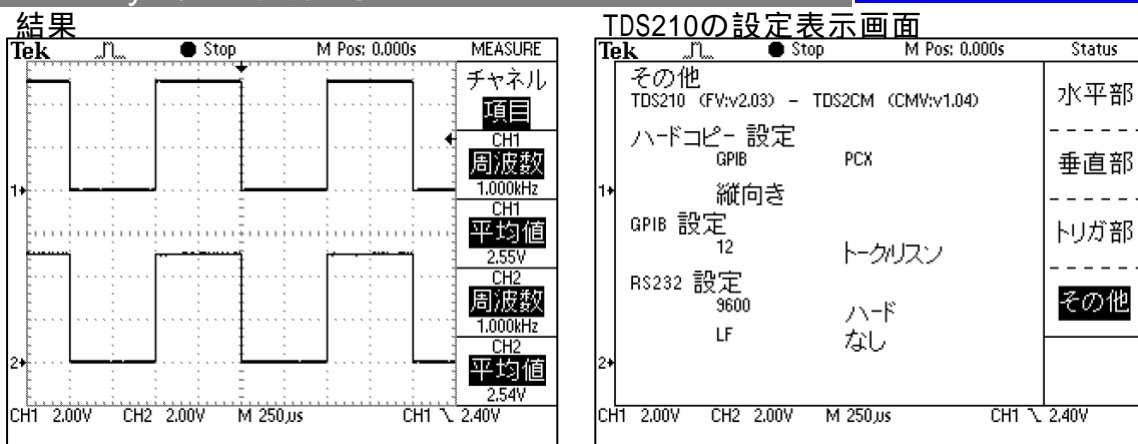
TDS210のハードコピー通信ポート：GPIBに設定

TDS210の出力形式：PCXに設定

```
Sub GetCapture()  
    ' バイナリデータの受信用配列を定義します  
    Dim bData() As Byte  
    ' カードをオープンします  
    ' 送受信バッファは100kB確保します  
    eg.CardOpen 100 & * 1024 &  
    ' コマンド送受信用デリミタを設定します  
    eg.Delimiter = eg.DELIMs.LF  
    ' 機器アドレスを指定します  
    eg.ActiveAddress = 12  
    ' 画面コピーコマンドを送ります  
    eg.AsciiLine = "hardcopy start"  
    ' 全データを受信します。  
    bData = eg.Binary  
    ' ブックと同じフォルダにtmp>pcxというファイル名でセーブします  
    Open ThisWorkbook.Path & "¥tmp.pcx" For Binary Access Write As #1  
    Put #1, , bData  
    Close #1  
    ' ファイルを開いてシートに貼り付けます  
    ActiveSheet.Pictures.Insert(ThisWorkbook.Path & "¥tmp.pcx").Select  
    ' 貼り付けた画像は大きいのでサイズ変更と縁取りを無効にします。  
    With Selection.ShapeRange  
        .ScaleWidth 0.5, msoFalse, msoScaleFromTopLeft  
        .ScaleHeight 0.5, msoFalse, msoScaleFromTopLeft  
        .Line.Weight = 1#  
        .Line.Visible = msoTrue  
    End With  
    ' カードを閉じます  
    eg.CardClose  
End Sub
```

## Binary プロパティ

[プロパティ一覧へ戻る](#)



### 注意

あらかじめ、ActiveAddressを必ず指定してください。

Ascii文字列を送受信するときは、AsciiLineプロパティを使ってください。

### オンリモード

egDef.GP232\_OnlyModeOpen でオープンしたときのモードです。

他社製のカードとの互換はありません。

### 書き込み

バイト型(Byte)、整数(Integer)、長整数(Long)、単精度浮動小数点(Single)、倍精度浮動小数点(Double)  
上記型の配列

Binaryプロパティにバイナリデータを代入すると、直ちに送信します。

### 読み取り

バイト型配列( Byte() )

E0Iを検出するまでバイナリデータを受信して返します。

### 解説

オンリモードで接続された機器に対して、バイナリデータを送受信するプロパティです。

書き込み、読み出し共にデリミタの設定は無視され、E0Iのみが有効になります。送受信はバッファを通して行われるため、あらかじめ十分なサイズのバッファを確保しておく必要があります。

書き込むことができるデータ型は5種類ありますが、いずれも1バイトの範囲、0～255を超えるとエラーになります。

複数バイトのバイナリデータを送信するときは、配列を使います。

また、受信データはバイト配列として取得します。

1バイトのバイナリデータであってもバイト配列で返ります。

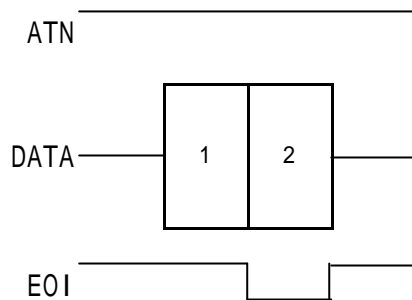


## Binary プロパティ

[プロパティ一覧へ戻る](#)

### バスライン

- ・ Binary プロパティに bData() を書き込んだとき  
(デリミタは CrLf , bData(0)=1, bData(1)=2 とします )



```
Dim bData(1) As Byte
bData(0) = 1
bData(1) = 2
egDef.GP232_OnlyModeOpen
eg.Binary = bData
```

### 注意

Ascii 文字列を送受信するときは , AsciiLine プロパティ を使ってください .

## 通常モード

eg.CardOpen でオープンしたときのモードです。  
他社製のカード用のEasyGPIBと互換があります。

## 書き込み専用

文字列型(String), バイト型(Byte), 整数(Integer), 長整数(Long), 単精度浮動小数点(Single), 倍精度浮動小数点(Double)  
上記型の配列

Cmdプロパティにバイナリデータを代入すると, ATNをLowに下げてバスコマンドとして送信します。

## 解説

ATNをLowにして, 代入した値をコマンドとして送信するプロパティです。  
配列を代入することによって, コマンド列を送信できますが, 配列の各要素は1~127の範囲になければなりません。  
配列の場合は要素の数のバイト数が出力されます。

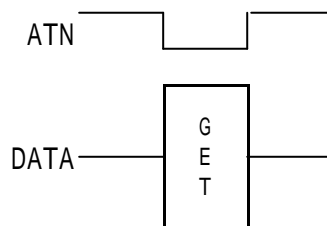
内部定数のCMDsプロパティを使うと, わかりやすいコードが書けます。

以下のコードは内部定数CMDsを使った例です。

```
eg.Cmd = eg.CMDs.DCL      ' (= &h14) DeviceClear  
eg.Cmd = eg.CMDs.GET      ' (= &h08) GroupExecuteTrigger  
eg.Cmd = eg.CMDs.GTL      ' (= &h01) GoToLocal  
eg.Cmd = eg.CMDs.LLO      ' (= &h11) LocalLockOut  
eg.Cmd = eg.CMDs.SDC      ' (= &h04) SelectedDeviceClear  
eg.Cmd = eg.CMDs.UNL      ' (= &h3F) UnListen  
eg.Cmd = eg.CMDs.UNT      ' (= &h5F) UnTalk
```

## バスライン

- ・ Cmdプロパティにeg.CMDs.GETを書き込んだとき



## 使用例

次のコードは、アドレス9,10,11の機器に対してGETコマンドを同時に送信します。  
EasyGPIBは同時にリスナ指定ができませんが、Cmdプロパティを使うことによって可能になります。

```
Dim bCmd(5) As Byte          ' コマンド送信用配列の定義
eg.Card = eg.CARDs.Yamato_PCI33 ' 使用するカードの指定
bCmd(0) = eg.CMDs.UNT        ' コマンドの準備
bCmd(1) = eg.CMDs.UNL
bCmd(2) = &H29               ' MLA 9
bCmd(3) = &H2A               ' MLA 10
bCmd(4) = &H2B               ' MLA 11
bCmd(5) = eg.CMDs.GET
eg.Cmd = bCmd                ' 送信
```

## 注意

あまり使われないコマンドですが、TCT(&h9)というコマンドを送信するときは十分に注意が必要です。  
このコマンドを送信すると、私が動作確認した5種類のカードでは、次のような3種類の動作になりました。

何事も無かったかのように継続する。

TCTを送信したあと、継続処理を放棄してスレーブモードに移行する。

TCTを送信したあと、継続処理があるときは停止状態になり、タイムアウトでエラー終了する。

いずれも、GPBの動作としてはありあるものですが、TCTはマルチコントローラで使用するコマンドで、EasyCGPIBではサポートしておりません。  
ですから、使用しないことをお勧めいたします。

## オンリモード

egDef.GP232\_OnlyModeOpen でオープンしたときのモードです。  
他社製のカードとの互換はありません。

## 解説

オンリモードでは使用しませんが、機能として残っているため使用することができます。  
動作は通常モードと同じです。

**通常モード**

eg.CardOpen でオープンしたときのモードです。  
他社製のカード用のEasyGPIBと互換があります。

読み取り専用 文字列(String)  
バスラインコマンドを数値として返すプロパティです。

**解説**

CMDsはバスラインコマンドに対応する数値を返すプロパティです。  
Cmdプロパティと共に使用します。

**CARDsプロパティと返す値**

プロパティ	返す値(16進)
eg.CMDs.GTL	(01)h
eg.CMDs.SDC	(04)h
eg.CMDs.GET	(08)h
eg.CMDs.LLO	(11)h
eg.CMDs.DCL	(14)h
eg.CMDs.UNL	(3F)h
eg.CMDs.UNT	(5F)h
eg.CMDs.MA	コントローラのアドレス
eg.CMDs.MLA	コントローラのマイ・リスナ・アドレス
eg.CMDs.MTA	コントローラのマイ・トーク・アドレス
eg.CMDs.SPE	(18)h
eg.CMDs.SPD	(18)h

**バスライン**

変化しません。

**使用例**

```
eg.Cmd = eg.CMDs.DCL      ' (= &h14) DeviceClear  
eg.Cmd = eg.CMDs.GET      ' (= &h08) GroupExecuteTrigger  
eg.Cmd = eg.CMDs.GTL      ' (= &h01) GoToLocal  
eg.Cmd = eg.CMDs.LLO      ' (= &h11) LocalLockOut
```

**オンリモード**

egDef.GP232\_OnlyModeOpen でオープンしたときのモードです。  
他社製のカードとの互換はありません。

**解説**

動作は通常モードと同じです。

## 通常モード

eg.CardOpen でオープンしたときのモードです。  
他社製のカード用のEasyGPIBと互換があります。

書き込み          文字列(String)

Ascii文字列の送受信で使用するデリミタを設定します。

読み取り          文字列(String)

現在設定されているデリミタを文字列で取得します。

## 解説

AsciiLineプロパティによる文字列の送信で、末尾に付加されるデリミタを設定するプロパティです。

内部定数のDELIMsプロパティを使うと、わかりやすいデリミタ指定ができます。  
代入する文字列には大文字・小文字の区別はありません。

デリミタを示す文字列	DELIMsを使った書き方	対応するデリミタ
CR	eg.DELIMs.CR	CR + EOI
CRLF	eg.DELIMs.CrLf	CR + LF + EOI
LF	eg.DELIMs.LF	LF + EOI
EOI	eg.DELIMs.Eoi	EOIのみ
CRLFNOEOI	eg.DELIMs.CrLfNoEoi	CR + LF (EOIなし)
CRNOEOI	eg.DELIMs.CrNoEoi	CR (EOIなし)
LFNOEOI	eg.DELIMs.LfNoEoi	LF (EOIなし)

受信時は、Delimiterプロパティの設定を拡張してデリミタを判断します。

Delimiterプロパティの設定と、受信時にデリミタとして判断する内容は次のようになります。

Delimiterの設定	デリミタとして判断する内容
CR	Cr または EOI
CRLF	LF または EOI
LF	LF または EOI
EOI	LF または EOI
CRLFNOEOI	LF または EOI
CRNOEOI	Cr または EOI
LFNOEOI	LF または EOI

## バスライン

変化しません。

### 使用例

eg.Delimiter = eg.DELIMs.Cr ' デリミタをCr/Lfに設定します .

次の例は , デリミタをCr + Lf (EOLなし)に設定します .  
書き方は異なりますが , すべて同じ結果になります .

```
eg.Delimiter = eg.DELIMs.CrLfNoEoi  
eg.Delimiter = "CrLfNoEoi"  
eg.Delimiter = "CRLFNOEOL"
```

### 注意

BinaryプロパティはDelimiterプロパティの設定に影響されません .  
また , Binaryプロパティで送受信してもDelimiterプロパティは変化しません .

デリミタが異なる機器が接続されているときは , ActiveAddressの切り替えとともに  
Delimiterプロパティもそのつど設定してください .  
ActiveAddressプロパティ , Delimiterプロパティの処理は内部変数を変更するだけで  
Apiをコールしないので高速に処理します .

### オンリモード

egDef.GP232\_OnlyModeOpen でオープンしたときのモードです .  
他社製のカードとの互換はありません .

### 解説

動作は通常モードと同じです .

**通常モード**

eg.CardOpen でオープンしたときのモードです。  
他社製のカード用のEasyGPIBと互換があります。

読み取り専用 文字列(String)  
デリミタを指定する文字列を返すプロパティです。

**解説**

DELIMsはDelimiterプロパティを使用するときにデリミタを示す文字列を返すプロパティです。Delimiterプロパティと共に使用します。

**CARDsプロパティと返す文字列**

プロパティ	返す文字列
eg.DELIMs.Cr	"CR"
eg.DELIMs.CrLf	"CRLF"
eg.DELIMs.Eoi	"EOI"
eg.DELIMs.Lf	"LF"
eg.DELIMs.CrLfNoEoi	"CRLFNOEOI"
eg.DELIMs.CrNoEoi	"CRNOEOI"
eg.DELIMs.LfNoEoi	"LFNOEOI"

**バスライン**

変化しません。

**使用例**

eg.Delimiter = eg.DELIMs.CrLf        ' デリミタをCR + LF + EOIに設定します  
eg.Delimiter = eg.DELIMs.CrNoEoi    ' デリミタをCR (EOIなし)に設定します

**オンリモード**

egDef.GP232\_OnlyModeOpen でオープンしたときのモードです。  
他社製のカードとの互換はありません。

**解説**

動作は通常モードと同じです。

**通常モード**

eg.CardOpen でオープンしたときのモードです。  
他社製のカード用のEasyGPIBと互換があります。

書き込み専用          文字列型(String)

WAVファイルを非同期で再生します。

**解説**

WAV形式のサウンドファイル名を代入することによってサウンドを再生します。  
アラーム音の再生などに利用します。  
ファイルはフルパスで指定する必要があります。  
再生はバックグラウンドで行われるため、Soundプロパティで待たされることはありません。  
サウンドを再生している間に、再度Soundプロパティが実行されると、再生中のサウンドは中断して新たに指定したサウンドが開始します。

長さゼロの文字列を代入すると、現在再生中のサウンドを中止することができます。  
また、サウンドファイルが見つからないときは再生しないで戻り、エラーは発生しません。

**使用例**

機器から送られてくるデータをA1セルから下方向に10回取得し、終了するとチャイムを鳴らすプログラムです。  
Excelで動作します。  
ThisWorkbook.Pathは、コードが書かれているブックのフォルダのパスを示します。  
WAVファイル名はchime.wavで、ブックと同じフォルダにおきます。

```
Sub Alarm()  
    Dim i As Integer          ' カウント変数の宣言  
    eg.CardOpen               ' カードをオープンします  
    eg.ActiveAddress = 2      ' 対象となる機器のアドレスを指定します  
    eg.Delimiter = eg.DELIMs.Cr ' デリミタを指定します  
    For i = 0 To 9            ' 10回繰り返します  
        Range("A1").Offset(i, 0) = eg.AsciiLine ' データを取得してセル  
    Next i                    ' に書き込みます  
    eg.Sound = ThisWorkbook.Path & "%chimes.wav" ' 再生します  
    eg.WAITmS = 2000          ' 少し待ちます  
    eg.CardClose              ' カードを閉じます  
End Sub
```

**オンリモード**

egDef.GP232\_OnlyModeOpen でオープンしたときのモードです。  
他社製のカードとの互換はありません。

**解説**

動作は通常モードと同じです。



**通常モード**

eg.CardOpen でオープンしたときのモードです。  
他社製のカード用のEasyGPIBと互換があります。

読み取り専用          バイト型(Byte)

ActiveAddressプロパティで指定された機器に対してシリアルポールを実行してステータスバイトを返します。

**解説**

ActiveAddressプロパティで指定された機器に対してシリアルポールを実行するので、あらかじめActiveAddressを指定する必要があります。

Spollプロパティはステータスバイトを返します。

SRQプロパティがTrueになると、接続されているいずれかの機器がシリアルポールを要求しています。

シリアルポールの要因となる機器のアドレスをActiveAddressで切り替えながらSPoIIプロパティでステータスバイトを読みながら要求している機器を調べます。

[プログラム例はSRQプロパティを参照してください。](#)

**オンリモード**

egDef.GP232\_OnlyModeOpen でオープンしたときのモードです。  
他社製のカードとの互換はありません。

読み取り専用          バイト型(Byte)

何も処理を行いません。  
常にゼロを返します。

**解説**

オンリモードでオープンした場合、シリアルポールは実行できません。

強制的に実行すると何も処理を行わないでゼロを返します。

エラーは発生しません。

**通常モード**

eg. CardOpen でオープンしたときのモードです。  
他社製のカード用のEasyGPIOと互換があります。

読み取り専用      論理型 (Boolean)  
GPIOバスのSRQの状態を示します。

**解説**

GPIOバスのSRQがLowの時はTrue, Highの時はFalseを返します。  
カードによってはSRQプロパティを読むと, SRQラインの状態にかかわらずSRQプロパティがリセットされるものがあるので, SRQを読んでTrueだったら発生要因となる機器すべてに対してシリアルポールを実行してください。

シリアルポールはActiveAddressで指定した機器に対してSpollプロパティを読み出すことによって実行されます。  
読み出した値はステータスバイトなので, その7ビットを調べることによってSRQを発信しているかどうかを確認できます。  
ステータスバイトの内容は機器によって異なるので, 各ビットの意味や条件はそれぞれのマニュアルを参照してください。

**例**

アドレス1, 2の二つの機器に対してサービスリクエストを有効にして監視し, SRQがアクティブになったら各機器に対してシリアルポールを実行してSRQを出力している機器のステータスバイトを表示します。  
ワークシート上に配置した二つのボタン(開始, 終了)に, それぞれTestSerialPollとEndProgramのマクロを登録します。

開始ボタンをクリックすると二つの機器にコマンドが送信され, 繰り返し処理に入ります。  
終了ボタンをクリックすると, 繰り返し処理を抜けて終了します。

SRQが発生するとActiveAddressを切り替えながらSpollを読み出して, それぞれの機器のステータスバイトを取得します。  
ステータスバイトのビット7を調べるには, &h40とAND演算し, ゼロかどうかで判断できます。  
SRQが発生している機器はビット7がセットされるのでゼロではなくなります。

```

Dim RunFlag As Boolean      ' 動作中フラグ

' プログラムを開始する
' 開始ボタンに登録
Sub TestSerialPoll()
    Dim SaveAddress As Integer ' アクティブアドレスの一時退避
    Dim Stat(2) As Byte       ' ステータスバイト格納用配列( Stat(0)は未使用 )

    If RunFlag Then Exit Sub ' 2重起動の防止
    RunFlag = True

    ' 設定
    eg.CardOpen
    eg.Delimiter = eg.DELIMs.CrLf ' Cr + Lf + Eoi
    eg.ActiveAddress = 1
    eg.AsciiLine = "S0,M1"        ' シリアルポールイネーブル,ホールドコマンドを送る
    eg.Delimiter = eg.DELIMs.Cr   ' Cr + Eoi
    eg.ActiveAddress = 2
    eg.AsciiLine = "S0,M1"        ' シリアルポールイネーブル,ホールドコマンドを送る

    ' 繰り返し処理
    Do While RunFlag            ' RunFlagがTrueの間繰り返し返す
        DoEvents               ' 終了ボタンがクリックできるようにする
        If eg.SRQ Then
            ' シリアルポール処理
            SaveAddress = eg.ActiveAddress ' アクティブアドレスの一時退避
            eg.ActiveAddress = 1
            Stat(eg.ActiveAddress) = eg.SPoll
            eg.ActiveAddress = 2
            Stat(eg.ActiveAddress) = eg.SPoll
            eg.ActiveAddress = SaveAddress ' アクティブアドレスを戻す
            ' ...以下,シリアルポールの処理を書く
            If Stat(1) And &H40 <> 0 Then
                ' アドレス1の機器がシリアルポール要求
                MsgBox "ステータスバイト= " & Hex(Stat(1)), , "アドレス 1の機器がSRQ要求しています"
            End If
            If Stat(2) And &H40 <> 0 Then
                ' アドレス2の機器がシリアルポール要求
                MsgBox "ステータスバイト= " & Hex(Stat(2)), , "アドレス 2の機器がSRQ要求しています"
            End If
        End If
        ' ...ここに通常の処理を書く
    Loop
    ' 終了
    eg.CardClose
    MsgBox "終了しました"
End Sub

```

## SRQ プロパティ

[プロパティ一覧へ戻る](#)

' プログラムを終了させる .

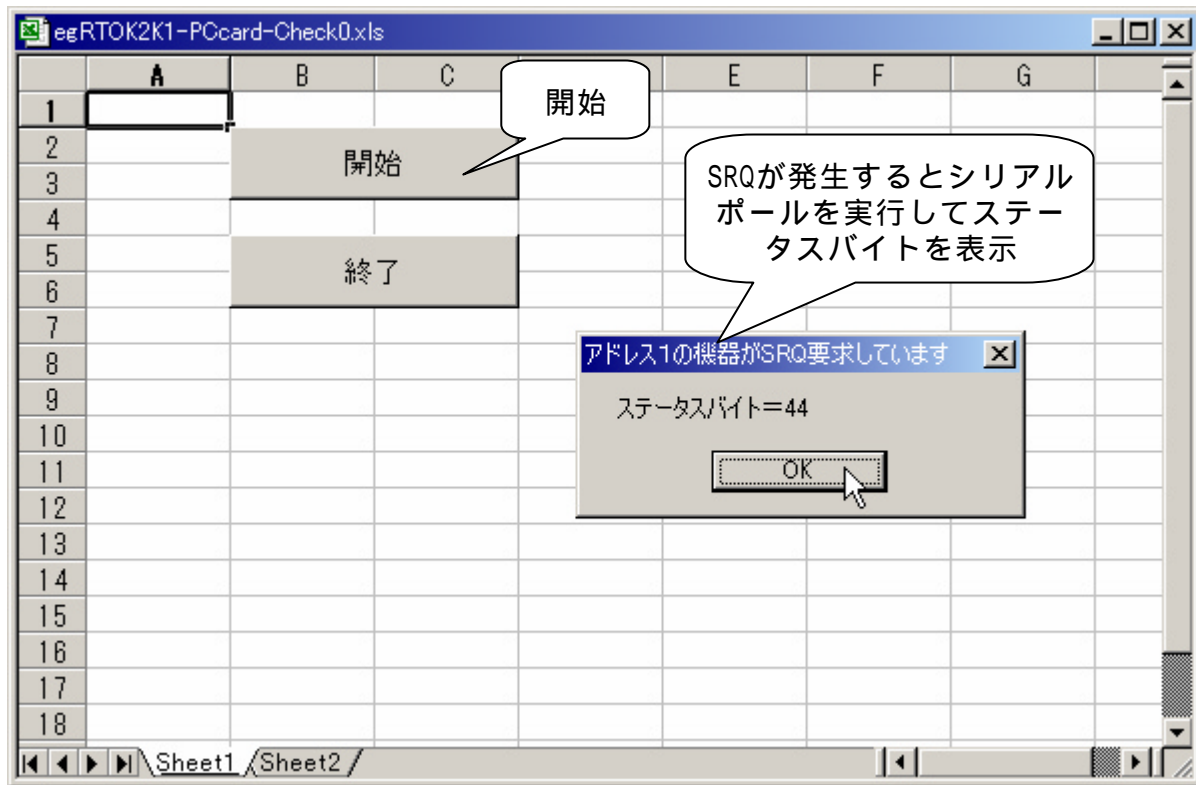
' 終了ボタンに登録

```
Sub EndProgram()
```

```
    RunFlag = False
```

```
End Sub
```

### < 実行結果 >



### オンリモード

egDef.GP232\_OnlyModeOpen でオープンしたときのモードです .  
他社製のカードとの互換はありません .

### 解説

動作は通常モードと同じです .

**通常モード**

eg.CardOpen でオープンしたときのモードです。  
他社製のカード用のEasyGPIBと互換があります。

書き込み          整数(Long)  
タイムアウトを秒単位で設定します。

読み取り          整数(Long)  
現在設定されているタイムアウト値を秒単位で取得します。

**解説**

GPIBのタイムアウトを秒単位で設定，または設定値の読み出しを行います。

**注意**

タイムアウトは正確ではありません。  
とくに，ハンドシェークのタイムアウトを監視しているカードがあるので，時間に関してはまったくあてにならないこともあります。

**オンリモード**

egDef.GP232\_OnlyModeOpen でオープンしたときのモードです。  
他社製のカードとの互換はありません。

**解説**

動作は通常モードと同じです。

### 通常モード

eg.CardOpen でオープンしたときのモードです。  
他社製のカード用のEasyGPIBと互換があります。

書き込み            長整数(Long)  
エラー番号を強制的に指定します。

読み取り            長整数(Long)  
直前に発生したエラー番号を返します。

### 解説

EasyGPIB内でエラーが発生するとErrorNumberプロパティとErrorHoldプロパティにエラー番号がセットされます。  
続けて処理が行われて正常終了するとErrorNumberはゼロにリセットされますが、ErrorHoldプロパティは変化しません。  
いずれも強制的にエラー番号を書き込むことができます。

### バスライン

変化しません。

### 使用例

```

eg.CardOpen           ' カードをオープンします
If eg.ErrorNumber <> 0 Then ' エラーをチェックします
    MsgBox "指定のカードは使用できません" ' メッセージの表示
    Exit Sub           ' 処理を中止
End If
    
```

### オンリモード

egDef.GP232\_OnlyModeOpen でオープンしたときのモードです。  
他社製のカードとの互換はありません。

### 解説

動作は通常モードと同じです。

**通常モード**

eg.CardOpen でオープンしたときのモードです。  
他社製のカード用のEasyGPIBと互換があります。

書き込み            長整数(Long)  
エラー番号を強制的に指定します。

読み取り            長整数(Long)  
最後に発生したエラー番号を取得します。

**解説**

EasyGPIB内でエラーが発生するとErrorNumberプロパティとErrorHoldプロパティにエラー番号がセットされます。  
次に処理が行われて正常終了するとErrorNumberはゼロにリセットされますが、ErrorHoldプロパティは変化しません。  
いずれも強制的にエラー番号を書き込むことができます。  
ErrorHoldを使用する前にゼロを代入してリセットしておきます。  
一連の処理が完了したら、トータル的にエラーがなかったかどうかをErrorHoldプロパティで確認します。

**バスライン**

変化しません。

**使用例**

```
' ErrorHoldプロパティをリセットします
eg.ErrorHold = 0

eg.CardOpen
eg.Delimiter = eg.DELIMs.Lf
eg.ActiveAddress = 12
If eg.ErrorHold <> 0 Then                                ' エラーをチェックします
    MsgBox "エラーが発生しました"                        ' メッセージの表示
    eg.CardClose
    Exit Sub                                              ' 処理を中止
End If
```

**オンリモード**

egDef.GP232\_OnlyModeOpen でオープンしたときのモードです。  
他社製のカードとの互換はありません。

**解説**

動作は通常モードと同じです。

**通常モード**

eg.CardOpen でオープンしたときのモードです。  
他社製のカード用のEasyGPIBと互換があります。

読み取り専用 文字列(String)  
最後に発生したエラー番号を取得します。

**解説**

EasyGPIB内でエラーが発生するとErrorNumberプロパティとErrorHoldプロパティにエラー番号がセットされます。  
次に処理が行われて正常終了するとErrorNumberはゼロにリセットされますが、ErrorHoldプロパティは変化しません。  
いずれも強制的にエラー番号を書き込むことができます。  
ErrorHoldTextプロパティは、ErrorHoldのエラー番号に対応するエラー文字列を取得します。

**バスライン**

変化しません。

**使用例**

```
' ErrorHoldプロパティをリセットします
eg.ErrorHold = 0

eg.CardOpen 500 & * 1024 &
eg.Delimiter = eg.DELIMs.Lf
eg.ActiveAddress = 12
If eg.ErrorHold <> 0 Then                                ' エラーをチェックします
    MsgBox "最後のエラー：" & eg.ErrorHoldText           ' メッセージの表示
    eg.CardClose
    Exit Sub                                              ' 処理を中止
End If
```

**オンリモード**

egDef.GP232\_OnlyModeOpen でオープンしたときのモードです。  
他社製のカードとの互換はありません。

**解説**

動作は通常モードと同じです。



### 通常モード

eg.CardOpen でオープンしたときのモードです。  
他社製のカード用のEasyGPIBと互換があります。

書き込み          長整数(Long)  
エラー番号を強制的に指定します。

読み取り          長整数(Long)  
直前に発生したapiのエラー番号を返します。

### 解説

標準化モジュール内でapiコールのエラーが発生すると、LocalErrorNumberプロパティにエラー番号がセットされます。  
この番号は各カード固有のもので、使用しているカードのマニュアルで発生したエラー番号を確認してください。  
LocalErrorNumberプロパティは、apiが正常値を戻すとゼロにリセットされます。

また、カードによっては正常動作でもゼロ以外の値を返す場合があるので、マニュアルで確認してください。  
なお、この番号はメーカー独自のものですので、デバッグ時以外は使用しないでください。EasyGPIBではErrorNumber,ErrorHoldを使用してください。

### バスライン

変化しません。

### オンリモード

egDef.GP232\_OnlyModeOpen でオープンしたときのモードです。  
他社製のカードとの互換はありません。

### 解説

動作は通常モードと同じです。

### 通常モード

eg.CardOpen でオープンしたときのモードです。  
他社製のカード用のEasyGPIBと互換があります。

読み取り専用 文字列(String)  
最後に発生したapiエラー番号に対するエラー内容を文字列で取得します。

### 解説

EasyGPIBの標準化モジュール内のapiがエラーを返すと、その番号をLocalErrorNumberに代入して処理を中止します。  
LocalErrorNumberに対するエラー内容を文字列で出力するのがLocalErrorTextプロパティです。  
なお、この番号はメーカー独自のものなので、デバッグ時以外は使用しないでください。  
EasyGPIBではErrorNumber,ErrorHoldを使用してください。

### バスライン

変化しません。

### オンリモード

egDef.GP232\_OnlyModeOpen でオープンしたときのモードです。  
他社製のカードとの互換はありません。

### 解説

動作は通常モードと同じです。

**通常モード**

eg.CardOpen でオープンしたときのモードです。  
他社製のカード用のEasyGPIBと互換があります。

指定した時間，処理を中断するプロパティです。  
時間(ミリ秒単位)は長整数(Long)型で書き込みます。

**解説**

指定された時間，処理を中断します。  
ただし，DoEventsを実行しているのでE x c e l がフリーズ状態になることはありません。時間はミリ秒単位で渡します。

Windows起動からの経過時間を取得するA P I 関数GetTickCountを使用しています。  
最大49.7日までの時間を指定できます。

**例 1**

```
ec.WAITmS = 500      '0.5秒待ちます。
```

**オンリモード**

egDef.GP232\_OnlyModeOpen でオープンしたときのモードです。  
他社製のカードとの互換はありません。

**解説**

動作は通常モードと同じです。

**通常モード**

eg.CardOpen でオープンしたときのモードです。  
他社製のカード用のEasyGPIBと互換があります。

読み取り専用 文字列型(String)  
EasyGPIBのバージョン情報を取得します。

**解説**

EasyGPIBのバージョンと読み込まれている共通化モジュールの種類とバージョンを文字列で取得する読み出し専用プロパティです。  
CardOpenが実行されていなくても動作します。

**例**

MsgBox eg.Version

**実行結果の例**

下記内容はマイナーバージョンアップなどによって変更される場合があります。

**オンリモード**

egDef.GP232\_OnlyModeOpen でオープンしたときのモードです。  
他社製のカードとの互換はありません。

**解説**

動作は通常モードと同じです。

## 通常モード

通常モードでオープンし、カードと変数の初期設定を行います。  
EasyGPIBの機能を利用する前に、必ず実行しておく必要があります。  
このメソッドは送受信時のバッファサイズを指定するパラメータを付加できます。  
省略するとデフォルト値、65536バイトが設定されます。

## 解説

CardOpenメソッドは必ず実行しなければなりません。  
CardOpenメソッドを実行すると、次のような処理が行われます。

## (1) カードの初期設定

## (2) 変数のリセット

エラー変数 = ゼロ

デリミタ = Cr+Lf+E0I

デリミタは接続機器に合わせる必要があります。

デフォルトを使用せずに、eg.Delimiterで明示的に設定してください。

タイムアウト = 5秒

カードによっては、通信中にタイマーが動作するものがあります。

送受信に時間がかかる処理の前には、タイムアウトを十分に大きな値に再設定するようにしてください。

タイムアウトはeg.Timeoutで設定できます。

アクティブアドレス = 0 (未設定の状態)

操作の対象となるアクティブアドレスは必ず設定してください。

ゼロは未設定の状態を示します。

eg.ActiveAddressで、設定してから送受信を行ってください。

送受信バッファサイズの設定

バッファは、一度に送受信するデータサイズよりも、十分に大きな値に設定してください。

バッファサイズを指定しないと、デフォルト値65536バイトに設定されます。

## (3) IFC出力

## (4) RENをLOWに下げる

バッファサイズを指定するときは、eg.CardOpen のあとに半角のスペースを入れてサイズを書いてください。

eg.CardOpen BufferSize

BufferSize:長整数

< 例 >

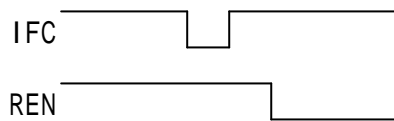
eg.CardOpen 100& \* 1024&

バッファサイズを100kBに指定します。

## CardOpen メソッド

[プロパティ一覧へ戻る](#)

バスライン



### 使用例

次の例は、アドバンテストのDMM(R6451A)に対して機種情報送信コマンド(IDN?)を送信し、返送されてきた文字列を表示するものです。DMMのアドレスは9に設定されているものとします。送受信バッファサイズは、デフォルトの65536バイトに設定されます。

Sub OpenClose()

eg.CardOpen                   最初に実行します。

eg.ActiveAddress = 9       機器のアドレスを指定します。

eg.Delimiter = eg.DELIMs.CrLf   デリミタを設定します。

eg.AsciiLine = "IDN?"       IDN?を送信します。

eg.WAITmS = 50           少し待ってから次の処理に移ります。(不要な場合もあります)

MsgBox eg.AsciiLine       受信文字列を表示します。

eg.CardClose           最後に実行します。

End Sub

< 結果 >



## オンリモード

オンリモードでオープンし、カードと変数の初期設定を行います。

GP232\_OnlyModeOpenはActiveCellで販売しているGPIB/シリアルアダプタGP232専用のモードです。

そのため、egではなくegDefに属するため、使用する場合は次のように書いてください。

```
egDef.GP232_OnlyModeOpen
```

オンリモードではアドレスの指定ができません。

EasyGPIBの機能を利用する前に、必ず実行しておく必要があります。

このメソッドは送受信時のバッファサイズを指定するパラメータを付加できます。省略するとデフォルト値、65536バイトが設定されます。

## 解説

GP232\_OnlyModeOpenメソッドは必ず実行しなければなりません。

GP232\_OnlyModeOpenメソッドを実行すると、次のような処理が行われます。

## (1) カードの初期設定

## (2) 変数のリセット

エラー変数 = ゼロ

デリミタ = Cr+Lf+EOL

デリミタは接続機器に合わせる必要があります。

デフォルトを使用せずに、eg.Delimiterで明示的に設定してください。

タイムアウト = 5秒

カードによっては、通信中にタイマーが動作するものがあります。

送受信に時間がかかる処理の前には、タイムアウトを十分に大きな値に再設定するようにしてください。

タイムアウトはeg.Timeoutで設定できます。

アクティブアドレス = - 1

アクティブアドレスは設定しないで下さい。

ActiveAddressを読み出すと-1になります。

変更しないで下さい。

送受信バッファサイズの設定

バッファは、一度に送受信するデータサイズよりも、十分に大きな値に設定してください。

バッファサイズを指定しないと、デフォルト値65536バイトに設定されます。

## (3) RENをLOWに下げる

バッファサイズを指定するときは、egDef.GP232\_OnlyModeOpen のあとに半角のスペースを入れてサイズを書いてください。

```
egDef.GP232_OnlyModeOpen BufferSize
```

BufferSize: 長整数

< 例 >

```
egDef.GP232_OnlyModeOpen 100& * 1024&
```

バッファサイズを100kBに指定します。

バスライン

IFC

REN



## オンリモード

## パラメータ

BitName

GP232で未使用の I/Oビットを文字列で指定します。

未使用ビットはRC0とRC3で、それぞれ"RC0","RC3"という文字列が対応します。

パラメータは省略できません。

書き込み          論理 (Boolean)

指定した I/Oビットを出力に設定し、Trueを書き込むとHigh、Falseを書き込むとLowを出力します。

GP232で未使用なのはRC0とRC3だけなので、それ以外のビットは制御できません。

読み取り          整数 (Long)

指定した I/Oビットを入力に設定し、入力されているロジックをHighの時はTrue、Lowの時はFalseで読み取ります。

GP232で未使用なのはRC0とRC3だけなので、それ以外のビットの状態は取得できません。

## 解説

GP232固有の機能で、標準のモジュール(eg)の機能ではなく、GP232の共通化モジュール(egDef)に含まれています。

そのため、利用するためには先頭に「egDef.」をつける必要があります。

egDef.GP232\_lobit(BitName) プロパティは、パラメータで指定したビットの操作を行います。

書き込み動作を行うと、そのビットは出力に設定され、書き込む値（論理値）によって、High、またはLowを出力します。

読み出し動作を行うと、そのビットは入力に設定され、入力状態を論理値で取得します。

操作の対象となるビットは、"RC0","RC3"のような 3文字で表します。

GP232で未使用なのは、RC0とRC3の 2 ビットだけなので、指定できるパラメータは"RC0","RC3"の 2 種類になります。

eg.CaedOpen、またはegDef.GP232\_OnlyModeOpenでカードを開くと、RC0,RC3、いずれも出力に設定され、Highを出力します。

この点には十分に注意してください。

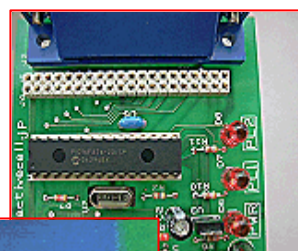
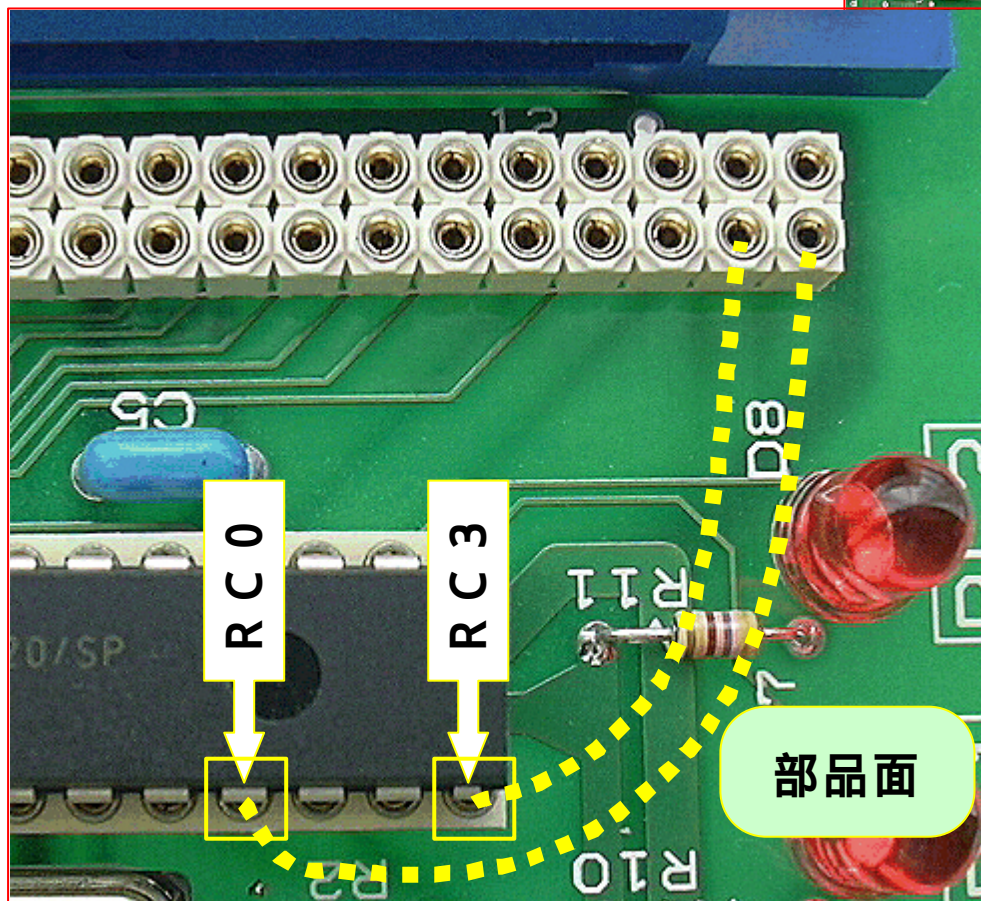
ロジック出力をRC0、またはRC3に接続してカードをオープンすると出力同士がぶつかるため、素子に損傷を与えることがあります。

これらのビットを入力として使用するには、直列に1kΩの抵抗を通すなどの対策が必要になります。

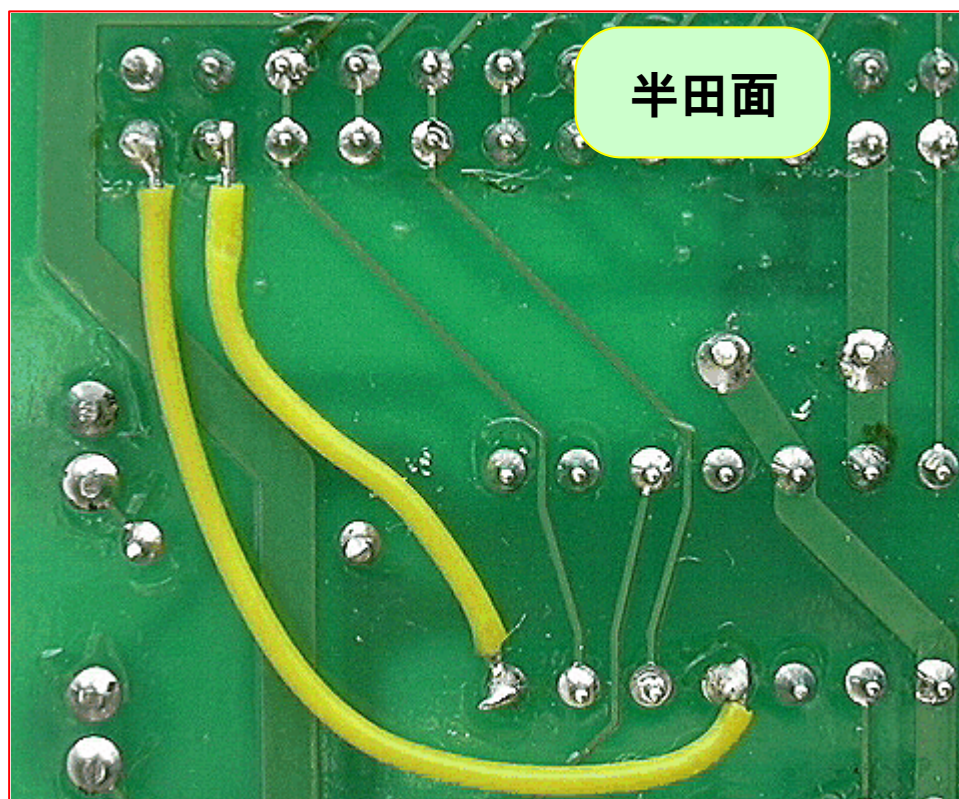
## 使用例

次の例は，アドバンテストのデジタルマルチメータ，R6551の値を監視し，上限値10.01Vを越えたときはRC3のLEDを点灯し，下限値9.99Vを下回ったときはRC0のLEDを点灯する，High-Low表示器です．

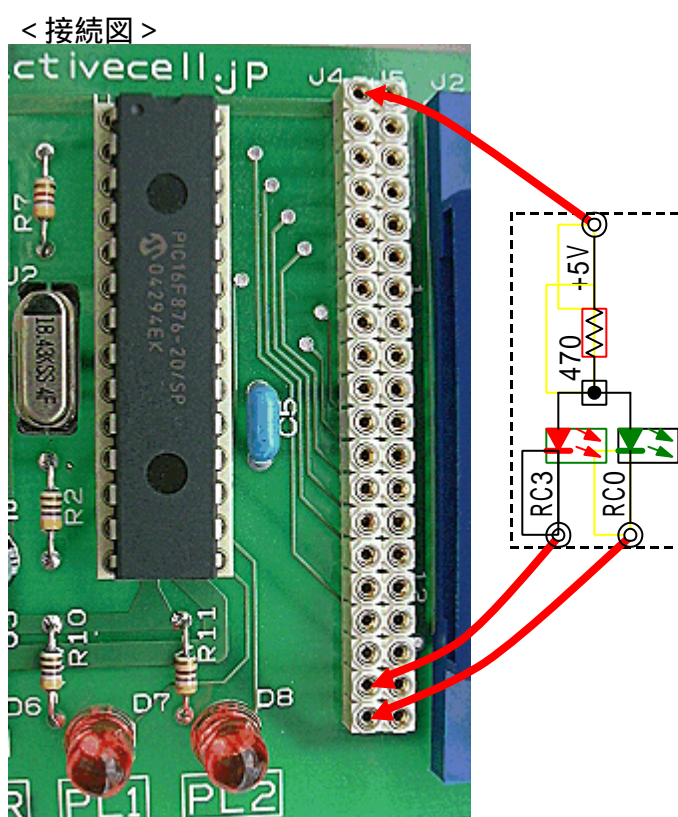
使いやすくするために，未接続のRC0とRC3をJ4のコネクタにジャンパ線で接続しました．



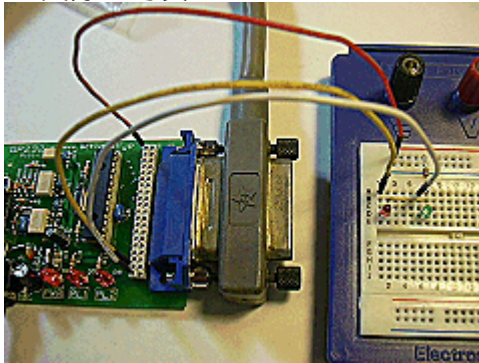




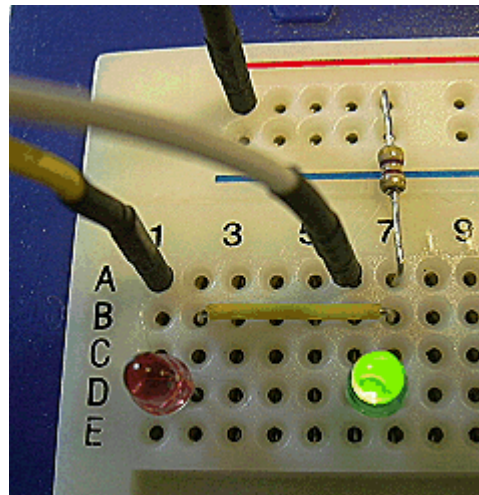
LEDと+5Vは、J4から取り出し、ブレッドボードにLEDと抵抗を実装しました。



### < 実際の写真 >



### < ブレッドボードの部品配置 >



### < リスト >

以下,LED点灯部分のみ抜粋します.

実際のワークシートは,同梱されているSample2.xlsを参照してください.

' 点滅	
egDef.GP232_IObit("RC0") = False	' 点灯
egDef.GP232_IObit("RC3") = False	
eg.WAITmS = 200	
egDef.GP232_IObit("RC0") = True	' 消灯
egDef.GP232_IObit("RC3") = True	
eg.WAITmS = 200	
egDef.GP232_IObit("RC0") = False	' 点灯
egDef.GP232_IObit("RC3") = False	
eg.WAITmS = 200	
egDef.GP232_IObit("RC0") = True	' 消灯
egDef.GP232_IObit("RC3") = True	

### エラー

LocalErrorのみ発生します.

LocalErrorNumber	
-901	ビット出力時のエラー
-902	ビット入力時のエラー

**通常モード**

eg.CardOpen でオープンしたときのモードです。  
他社製のカード用のEasyGPIBと互換があります。

カードと変数の終了処理を行います。  
EasyGPIBの使用が終了したら必ず実行してください。

**解説**

CardCloseメソッドは、プログラムの最後に必ず実行しなければなりません。  
CardCloseメソッドを実行すると、次のような処理が行われます。

- (1) カードの初期設定
- (2) 変数のリセット  
エラー変数 = ゼロ  
デリミタ = Cr+Lf+E01  
タイムアウト = 5秒  
アクティブアドレス = 0  
送受信バッファ = 6 5 5 3 6 バイト
- (3) RENをHIGHにあげる

**使用例**

[CardOpenメソッドを参照してください。](#)

**注意**

使用するカードによってはカードクローズに関するApiをサポートしていないものがあります。  
通常の使用においては問題ありませんが、PCMCIAタイプの場合は、一度開いたアプリケーションを終了しないと抜き取ることが出来ないことがあるので注意して下さい。

**< 例 >**

	PCMCIAの使用終了と抜き取り
Windows起動	可
Excel起動	可
eg.CardOpen実行	不可
eg.CardClose実行	不可
Excel終了	可

**オンリモード**

eg.Def.GP232\_OnlyModeOpen でオープンしたときのモードです。  
他社製のカードとの互換はありません。

**解説**

動作は通常モードと同じです。

## エラーコード表

番号	内容
0	正常
12010	カードオープンエラー
12020	カードクローズエラー
12030	マイアドレス取得エラー
12040	未オープン
12050	IFC出力エラー
12060	タイムアウトの設定，または読み出しエラー
12070	バッファサイズの設定，または読み出しエラー
12080	デリミタの設定，または読み出しエラー
12090	コマンド送信エラー
12100	アクティブアドレスの設定，または読み出しエラー
12110	Ascii文字列の送信，または受信エラー
12120	アクティブアドレス未設定エラー
12130	バイナリデータの送信，または受信エラー
12140	SRQの読み取りエラー
12150	シリアルボール実行エラー
12160	パラメータ許容範囲外エラー
12170	未対応のパラメータ
13000	タイムアウトが発生

## 対応機種と共通化モジュール

## 対応機種と共通化モジュール

ここに示す内容は私のパソコン環境で行った結果を示すだけで、動作を保証するものではありません。

### アクティブセル

<http://www.activecell.jp>

#### 必要なソフトウェアと対応するEasyGPIB共通化モジュール

- ・ EasyGPIBを動作させるために必要なファイル  
eg-V122.bas  
Activecell-GP232-V123.bas
- ・ 必要なソフトウェア  
ソフト名： シリアルポートのドライバはWindowsのApiを使用するので特に必要ありません。
- ・ 動作可能環境
  - O S : Windows95,Windows98,WindowsMe,WindowsNT4.0,Windows2000,WindowsXP
  - V B : VisualBasic5.0, 6.0
  - Office : 97,2000,XP,2003  
(ただしOffice2003はWindows2000,XPより前のバージョンでは動作しません)
- ・ 共通化モジュール開発環境
  - カード : GP232
  - O S : WindowsXP
  - Office : 2000
- ・ 共通化モジュール動作確認済み環境
  - カード : GP232
  - O S : Windows98,WindowsMe,WindowsXP
  - アプリ : VisualBasic5.0,Excel97,Excel2000,ExcelXP,Excel2003



## 対応機種と共通化モジュール

### カード固有の注記

プログラムがエラーなどで停止するとシリアルポートがオープンされたままになることがあります。

その場合は、一旦アプリケーションを終了する必要があります。

送受信処理を行っている間は、他の操作はできません。

専用機能 GP232\_OnlyModeOpen

Version1.23よりGP232用専用の機能を追加しました。

ただし、EasyGPIBの本体であるeg-V122.basにはまったく変更がないので、共通化モジュールだけを入れ替えれば使えるようになります。

GP232\_OnlyModeOpenは共通化モジュールegDefに含まれるため、先頭にegDefをつけてドットでつないで使用します。

使い方はeg.CardOpenとまったく同じです。

ただし、egDef.GP232\_OnlyModeOpenで開くとオンリモードになり、次のような動作になります。

- ActiveAddressの指定はできません。

- 読み出すと-1になります。

- この値は変更しないで下さい。

- シリアルポールは実行できません。

- シリアルポールを実行して処理は行いません。

- また、エラーも発生しないので注意してください。

- IFCは出力されません。

- egDef.GP232\_OnlyModeOpenを実行してもIFCパルスは出力されません。

- ただしRENはLowを出力します。

- eg.AsciiLineに書き込むと文字列を送信します。

- バスコマンドなどは一切出力されずに、直ちに文字列が送信されます。

- ただし、デリミタは付加されます。

- eg.AsciiLineを読み出すと文字列を受信します。

- バスコマンドなどは一切出力されずに、直ちに文字列の受信を開始します。

- デリミタ、またはEOIを検出すると終了します。

- eg.Binaryに書き込むとバイナリデータを送信します。

- バスコマンドなどは一切出力されずに、直ちに送信されます。

- eg.Binaryを読み出すとバイナリデータを受信します。

- バスコマンドなどは一切出力されずに、直ちに受信を開始します。
- EOIを検出すると終了します。

他のコマンドはeg.CardOpenで開いたときと同じです。



## 改版記録

[illegible]