

REX - USB220

U S B 2 . 0 G P I B C o n v e r t e r

Linux 用ライブラリ仕様書

2 0 0 5 年 1 1 月 0 7 日

Rev1.0

Revision	変更日	変更点
Rev1.0	2005年11月07日	初版

ライブラリ仕様

API 関数は、デバイスオープン・クローズに関する関数、9914 レジスタ制御関数、GPIB 機器制御関数に分類されます。以下に、関数の動作概要を示します。

- ・デバイスオープン・クローズに関する関数として下記の関数を用意しています。

USB220 を複数台接続する際に”U2”が付く関数を使用してください

関数名	動作概要
gp_attach , U2GpOpenUnit	デバイスをオープン
gp_detach , U2GpCloseUnit	デバイスをクローズ
U2GpEnumUnit	接続されている全デバイスのコンバータ ID 情報を取得

- ・USB220 のレジスタを直接制御する関数として下記の関数を用意しています。

USB220 を複数台接続する際に頭に”U2”が付く関数を使用してください

関数名	動作概要
OutPort , U2OutPort	USB220 のレジスタに書き込み
InPort , U2InPort	USB220 のレジスタから読み込み

- ・GPIB 機器の制御に関する関数として下記の関数を用意しています。

USB220 を複数台接続する際に頭に”U2”が付く関数を使用してください

関数名	動作概要
gp_init , U2Gpinit	USB220 の初期化
gp_cli , U2Gpcli	IFC ラインを TRUE にします
gp_ren , U2Gpren	REN ラインを TRUE にします
gp_clr , U2Gpclr	CLR 又は SDC コマンド送信
gp_wrt , U2Gpwrt	GPIB 機器にデータ送信
gp_red , U2Gpred	GPIB 機器からデータ受信
gp_trg , U2Gptrg	GET コマンド送信
gp_wsrg , U2Gpwsrg	指定時間 SRQ を待つ(ステータスレジスタ 1)
gp_wsrgb , U2Gpwsrgb	指定時間 SRQ を待つ(バスステータスレジスタ)
gp_rds , U2Gprds	シリアルポールを実行
gp_rds1 , U2Gprds1	シリアルポールを実行
gp_srsg , U2Gpsrsg	SRQ 割り込み
gp_lcl , U2Gplcl	GPIB 機器をローカル状態に設定

gp_llc , U2Gpllc	LL0 コマンド送信
gp_tmout , U2Gptmout	バスタイムアウト時間設定
gp_setdelay , U2Gpsetdelay	外部変数のディレイ時間設定
gp_count , U2Gpcount	受信データ数の取得
gp_delm , U2Gpdelm	デリミタの設定
gp_tfrount , U2Gptfrount	GPIB 機器にバイナリデータ送信
gp_tfrin , U2Gptfrin	GPIB 機器からバイナリデータ受信
gp_tfrinit , U2Gptfrinit	GPIB 機器からバイナリデータ受信するためのトーカ指定
gp_tfrins , U2Gptfrins	GPIB 機器からバイナリデータ受信
gp_tfrend , U2Gptfrend	GPIB 機器からバイナリデータ受信するためのトーカ解除
gp_wtb , U2Gpwtb	コマンド文字列を送信
gp_myadr , U2Gpmyadr	USB220 の機器アドレスを取得

・その他の関数として下記の関数を用意しています。

USB220 を複数台接続する際もそのまま使用できます

関数名	動作概要
gp_wait	指定時間待つ

・補助関数として下記の関数を用意しています。

関数名	動作概要
gp_srqCheck , U2GpsrqCheck	SRQ ラインの現在の状態を取得
gp_wrtcd , U2Gpwrtcd	GPIB バス上にデータ送信
gp_tfrountd , U2Gptfrountd	GPIB バス上にバイナリデータ送信
gp_redd , U2Gpredd	GPIB 上からのデータ受信
gp_redah , U2Gpredah	GPIB 機器からデータ受信
gp_redrst , U2Gpredrst	リスナ解除、RFD ホールドオフの解除
gp_findlstn , U2Gpfindlstn	リスナ機器の検出

API 関数使用上の注意

(1) 1 台のパソコンで USB220 コンバータを複数台使用する場合には、先頭が U2xxxx となる関数 (U2Gpinit など) を使用します。プログラムの始めに、指定コンバータ ID の USB220 のハンドルを関数 U2GpOpenUnit() で取得し、U2xxxx の第一引数に指定します。

(2) 1 台の USB220 コンバータで複数台の GPIB 機器 (計測器) の制御を行うには、機器アドレス間にカンマ ", " を指定します。U2xxxx 関数を使用する必要はありません。

機器アドレス指定を行う関数 gp_clr(), gp_wrt(), gp_red(), gp_trg(), gp_rds(), gp_rds1(), gp_lcl(), gp_tfROUT(), gp_tfrin(), gp_tfrinit() で使用します。

たとえば、以下のように使用してください。

```
gp_clr("3,5"); // リスナ 3 と 5 にコマンド SDC を送信します。
```

```
gp_wrt("6,20,30", "*CLS"); // リスナ 6 と 20 と 30 にデータ "*CLS" を送信します。
```

```
gp_red("3,20", buf, bufLen); /* アドレス 3 をトーカに、アドレス 20 をリスナに指定してトーカ 3 からのデータを受信します。*/
```

```
gp_rds("3,20", status_byte); /* シリアルポールを実行し、アドレス 3 と 20 にステータスバイトを問い合わせます。*/
```

(3) 二次アドレスをもつ GPIB 機器の制御を行うには、一次アドレスに続いて、二次コマンド (96(0x60h)+二次アドレス) を指定します。たとえば、以下のように使用してください。

```
gp_clr("3,111"); /* 一次アドレス 3, 二次アドレス 15 のリスナにコマンド SDC を送信します。*/
```

オープンクローズ関数

<u>書式</u>	<code>int gp_attach ()</code>
機能	コンバータをオープンします。複数接続の場合は先に見つけたコンバータをオープンします。 正常にオープンされた時に返されるコンバータのハンドル(ファイルディスクリプタ)は他の関数呼び出し時の第一引数として必要となります。
引数	なし (IN)
戻値	コンバータを正常にオープンした場合はコンバータのハンドルを返します。 ID のコンバータを見つけられなかった場合、またはオープン時エラーが発生した場合は NULL が返されます。
補足	プログラム終了時、 <code>gp_detach ()</code> によりコンバータをクローズするようにしてください。

<u>書式</u>	<code>int U2GpOpenUnit(int UnitId)</code>
機能	指定の ID のコンバータをオープンします。正常にオープンされた時に返されるコンバータのハンドルは他の関数呼び出し時の第一引数として必要となります。 1 台のコンバータのみ接続の場合は、本関数を呼び出す必要はありません。
引数	UnitId (IN) オープンするコンバータ ID (工場出荷時の設定値はゼロ)
戻値	指定のコンバータを正常にオープンした場合はコンバータのハンドルを返します。 指定された ID のコンバータを見つけられなかった場合、またはオープン時エラーが発生した場合は NULL が返されます。
補足	プログラム終了時、 <code>U2GpCloseUnit ()</code> によりコンバータをクローズするようにしてください。

[書式](#) VOID **gp_detach** (int num)

機能 コンバータをクローズします。

引数 num (IN) クローズするコンバータのハンドル

戻値 ありません。

[書式](#) VOID **U2GpCloseUnit**(int hDriver)

機能 コンバータをクローズします。

引数 hDriver (IN) クローズするコンバータのハンドル

戻値 ありません。

9914 レジスタ制御関数

書式 USHORT **OutPort**(USHORT Reg, USHORT OutVal)
 USHORT **U2OutPort**(int hDriver, USHORT Reg, USHORT OutVal)

機能 GPIB コントローラのレジスタに直接、値を書き込みます。

引数 hDriver (IN) コンバータのハンドル(複数台接続時使用)
 Reg (IN) レジスタオフセット値(0~7を指定)
 OutVal (IN) レジスタに書き込む値

戻値 N 正常終了時、書き込んだ値が返されます。
 エラー時には、0xFFFFh が返されます。

書式 USHORT **InPort**(USHORT Reg)
 USHORT **U2InPort**(int hDriver, USHORT Reg)

機能 GPIB コントローラのレジスタから直接、値を読み込みます。

引数 hDriver (IN) コンバータのハンドル(複数台接続時使用)
 Reg (IN) レジスタオフセット値(0~7を指定)

戻値 N 正常終了時、読み込んだ値が返されます。
 エラー時には、0xFFFFh が返されます。

GPIB 機器制御関数

書式 `int gp_init(USHORT GpAdrs, USHORT IOBase, USHORT IrqNo)`
 `int U2Gpinit(int hDriver, USHORT GpAdrs)`

機能 USB220 の GPIB 機器アドレスをセットし、GPIB コントローラの初期化を行います。また、各パラメータ(バスタイムアウト時間, デイレイ時間, デリミタ)の初期値を設定します。GPIB 制御を行う前に必ず呼び出してください。

引数 `hDriver` (IN) コンバータのハンドル(複数台接続時使用)
 `GpAdrs` (IN) USB220 の GPIB 機器アドレス
 `IOBase` (IN) REX-5052 互換のための引数です。0 を指定してください
 `IrqNo` (IN) REX-5052 互換のための引数です。0 を指定してください

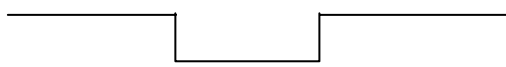
戻値 0 正常終了
 -1 ライブラリ DeviceIoControl() リクエストエラー
 60 USB220 の GPIB 機器アドレス設定エラー

書式 `int gp_cli(void)`
 `int U2Gpcli(int hDriver)`

機能 IFC ラインを TRUE にします。

引数 `hDriver` (IN) コンバータのハンドル(複数台接続時使用)

戻値 0 正常終了
 -1 ライブラリ DeviceIoControl() リクエストエラー

GPIB IFC 
バス

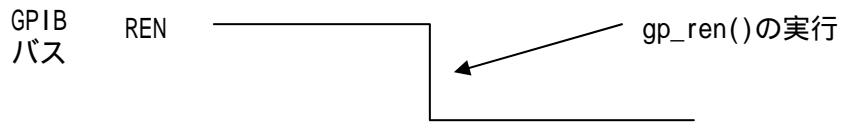
約 10 ms

書式 `int gp_ren(void)`
 `int U2Gpren(int hDriver)`

機能 REN ラインを TRUE にします。

引数 `hDriver` (IN) コンバータのハンドル(複数台接続時使用)

戻値 0 正常終了
 -1 ライブラリ `DeviceIoControl()` リクエストエラー



書式

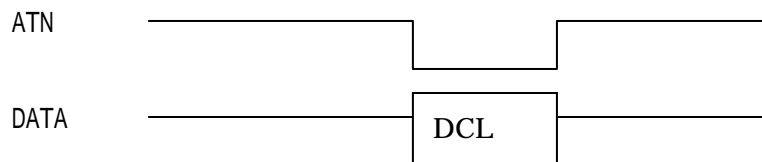
```
int gp_clr(PCHAR adrs)
int U2Gpclr(int hDriver, PCHAR adrs)
```

機能 クリアコマンド(DCL 又は SDC)を送信します。引数 adrs に機器アドレスを指定しない場合は DCL コマンドを、指定する場合は SDC コマンドを送信します。

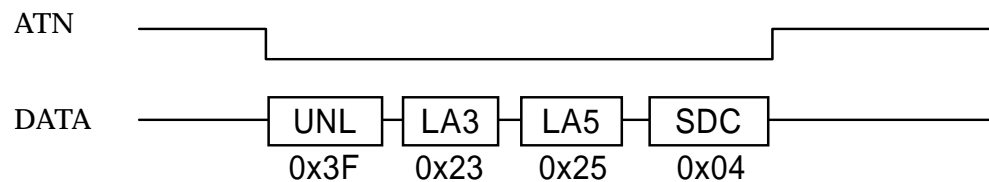
引数 hDriver (IN) コンバータのハンドル(複数台接続時使用)
adrs (IN) GPIB 機器アドレス

戻値 0 正常終了
-1 ライブラリ DeviceIoControl()リクエストエラー
53 GPIB バスタイムアウト
63 GPIB 機器アドレス設定エラー
-5, -6, -7 USB 転送時エラー

補足 機器アドレスの指定が無い場合は、GPIB 上の全機器に対して DCL(Device Clear)コマンドを送信します。
(使用例) gp_clr("");



0x14
機器アドレスの指定がある場合は、指定の機器に対して SDC(Selected Device Clear)コマンドを送信します。
(使用例) gp_clr("3,5");



書式

int **gp_wrt**(PCHAR adrs, PCHAR buf)

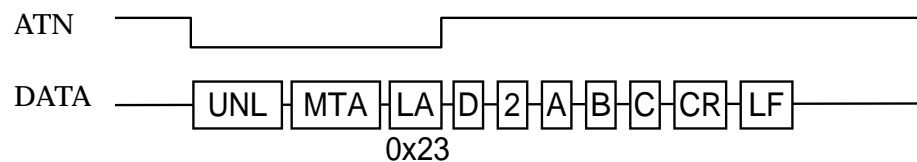
int **U2Gpwrt**(int hDriver, PCHAR adrs, PCHAR buf)

機能 引数 adrs で指定した GPIB 機器に対してデータ送信します。デリミタ指定関数 gp_delm および U2Gpdelm で指定されたデリミタを送信データに自動的に付加して送信を行います。

引数 hDriver (IN) コンバータのハンドル(複数台接続時使用)
adrs (IN) GPIB 機器アドレス
buf (IN) 送信文字列を格納するバッファアドレス

戻値 0 正常終了
-1 ライブラリ DeviceIoControl()リクエストエラー
53 GPIB バスタイムアウト
63 GPIB 機器アドレス設定エラー
64 送信データ設定エラー
-5, -6, -7 USB 転送時エラー

補足 機器アドレス 3 にアスキーデータ "D2ABC" を送信する場合の例
(使用例) gp_wrt("3", "D2ABC");



書式

int **gp_red**(PCHAR adrs, PCHAR buf, int bufLen)

int **U2Gpred**(int hDriver, PCHAR adrs, PCHAR buf, int bufLen)

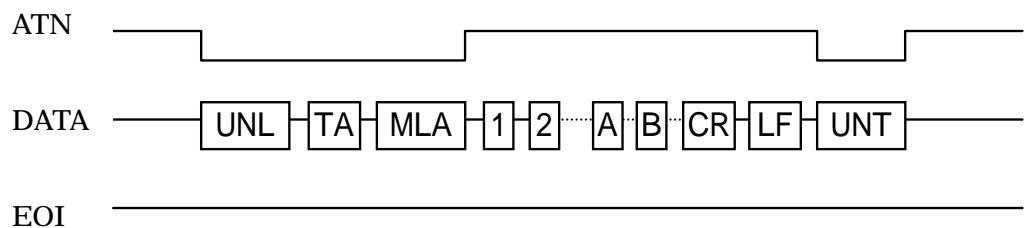
機能 引数 adrs で指定した GPIB 機器をトーカーに指定し、データの受信を行います。デリミタ指定関数 gp_delm および U2Gpdelm で指定されたデリミタ(もしくは EOI)を受信するかバスタイムアウトになるまで制御を返しません。

注)アプリケーションにはデリミタコードを返しません。

引数 hDriver (IN) コンバータのハンドル(複数台接続時使用)
adrs (IN) GPIB 機器アドレス
buf (OUT) 受信文字列を格納するバッファアドレス
bufLen (IN) 受信バッファのサイズ

戻値 0 正常終了
-1 ライブラリ DeviceIoControl()リクエストエラー
53 GPIB バスタイムアウト
61 バッファオーバーフロー(デリミタ受信しないまま、サイズ分を受信)
63 GPIB 機器アドレス設定エラー
64 受信バッファサイズ設定エラー
-5, -6, -7 USB 転送時エラー

GPIB
バス



書式

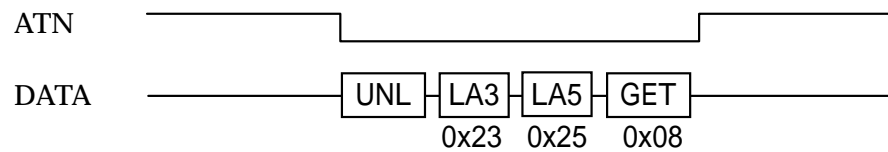
```
int gp_trg(PCHAR adrs)
int U2Gptrg(int hDriver, PCHAR adrs)
```

機能 トリガコマンド (GET) を送信します。

引数 hDriver (IN) コンバータのハンドル(複数台接続時使用)
adrs (IN) GPIB 機器アドレス

戻値 0 正常終了
-1 ライブラリ DeviceIoControl()リクエストエラー
53 GPIB バスタイムアウト
63 GPIB 機器アドレス設定エラー
-5, -6, -7 USB 転送時エラー

補足 機器アドレス 3 と 5 に GET(Group Execute Trigger)コマンドを送信する場合の例
(使用例) gp_trg("3,5");



書式 `int gp_wsrq(int WaitSecTime)`
 `int U2Gpwsrq(int hDriver, int WaitSecTime)`

機能 指定された時間、SRQ が発行されるのを待ちます。(インタラプトステータスレジスタ 1 をリード) SRQ を受信した場合、直ちに制御を返します。

引数 `hDriver` (IN) コンバータのハンドル(複数台接続時使用)
 `WaitSecTime` (IN) SRQ を待つ時間(秒単位で指定)

戻値 0 SRQ 受信
 -1 タイムアウト(SRQ 未受信)

書式 `int gp_wsrgb(int WaitSecTime)`
 `int U2Gpwsrgb(int hDriver, int WaitSecTime)`

機能 指定された時間、SRQ が発行されるのを待ちます。(バスステータスレジスタをリード) SRQ を受信した場合、直ちに制御を返します。

引数 `hDriver` (IN) コンバータのハンドル(複数台接続時使用)
 `WaitSecTime` (IN) SRQ を待つ時間(秒単位で指定)

戻値 0 SRQ 受信
 -1 タイムアウト(SRQ 未受信)

書式

int **gp_rds**(PCHAR adrs, PCHAR status_byte)

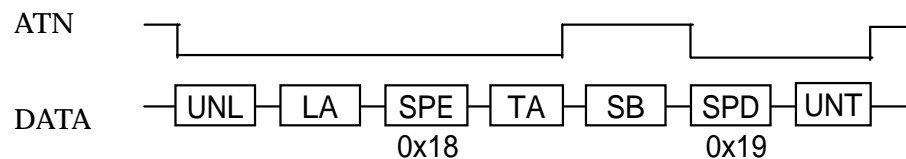
int **U2Gprds**(int hDriver, PCHAR adrs, PCHAR status_byte)

機能 シリアルポールを実行し、ステータスバイトを取得します。

引数 hDriver (IN) コンバータのハンドル(複数台接続時使用)
adrs (IN) GPIB 機器アドレス
status_byte (OUT) ステータスバイトを受け取るための配列。接続機器台数分以上の配列を確保してその先頭アドレスを指定します。

戻値 0 正常終了
-1 ライブラリ DeviceIoControl()リクエストエラー
53 GPIB バスタイムアウト
63 GPIB 機器アドレス設定エラー
-5, -6, -7 USB 転送時エラー

GPIB
バス



SB : ステータスバイト
SPE : シリアルポールイネーブル
SPD : シリアルポールディセーブル

書式

int **gp_rds1**(PCHAR adrs, PCHAR status_byte)

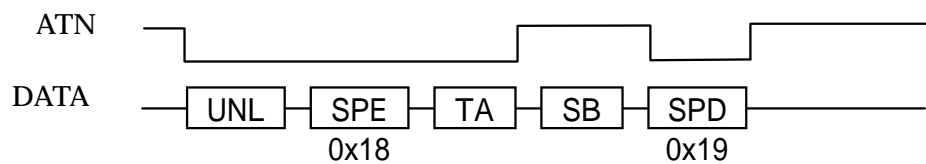
int **U2Gprds1**(int hDriver, PCHAR adrs, PCHAR status_byte)

機能 シリアルポールを実行し、ステータスバイトを取得します。
gp_rds(U2Gprds)との違いは最後に UNT(Untalk)コマンドを送信しない点です。

引数 hDriver (IN) コンバータのハンドル(複数台接続時使用)
adrs (IN) GPIB 機器アドレス
status_byte (OUT) ステータスバイトを受け取るための配列。接続機器台数分以上の配列を確保してその先頭アドレスを指定します。

戻値 0 正常終了
-1 ライブラリ DeviceIoControl()リクエストエラー
53 GPIB バスタイムアウト
63 GPIB 機器アドレス設定エラー
-5, -6, -7 USB 転送時エラー

GPIB
バス



SB : ステータスバイト
SPE : シリアルポールイネーブル
SPD : シリアルポールディセーブル

書式

int **gp_srq**(int SrqMode, PAPIFUNC pFunc)

int **U2Gpsrq**(int hDriver, int SrqMode, PAPIFUNC pFunc)

機能 SRQ 割り込みの実行および解除を行います。

引数 hDriver (IN) コンバータのハンドル(複数台接続時使用)
SrqMode (IN) モードフラグ(0:解除フラグ, 1:実行フラグ)
pFunc (IN) コールバック関数のポインタ

戻値 0 正常終了
-1 ライブラリ DeviceIoControl()リクエストエラー
-4 スレッドクリエートエラー
71 SRQ 割り込みは実行されている(実行時のエラー)

書式

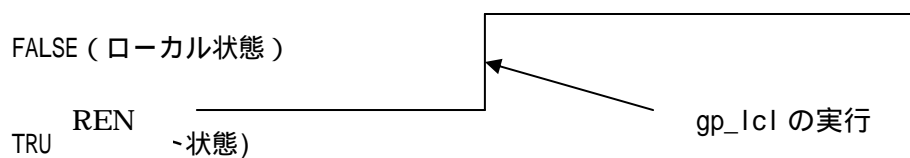
int **gp_lcl**(PCHAR adrs)
int **U2Gplcl**(int hDriver, PCHAR adrs)

機能 GPIB 機器をローカル状態に設定します。

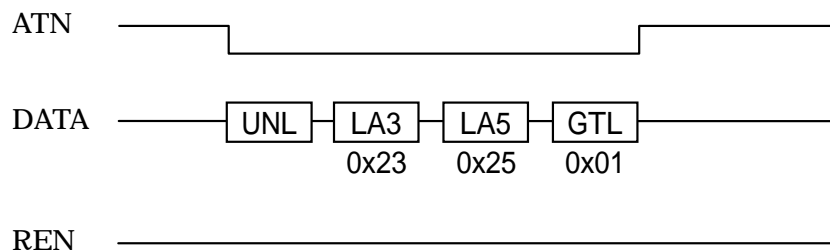
引数 hDriver (IN) コンバータのハンドル(複数台接続時使用)
adrs (IN) GPIB 機器アドレス

戻値 0 正常終了
-1 ライブラリ DeviceIoControl()リクエストエラー
53 GPIB バスタイムアウト
63 GPIB 機器アドレス設定エラー
-5, -6, -7 USB 転送時エラー

補足 機器アドレスの指定が無い場合は、REN ラインを High(FALSE)にします。
(使用例) gp_lcl("");



機器アドレスの指定がある場合は、指定の機器に対して GTL(Go To Local)コマンドを送信します。
(使用例) gp_lcl("3,5");



書式 `int gp_setdelay(int DelayTime)`
 `int U2Gpsetdelay(int hDriver, int DelayTime)`

機能 ATN ラインを TRUE 又は FALSE にする際のディレイ時間を設定します。コマンド送信時に GPIB タイムアウトとなる場合に調整します。初期値は 0usec です。

引数 `hDriver` (IN) コンバータのハンドル(複数台接続時使用)
 `DelayTime` (IN) ディレイ時間(マイクロ秒単位で指定)

戻値 0 正常終了
 -1 ライブラリ DeviceIoControl()リクエストエラー

補足 初期設定(0 マイクロ秒)は `gp_init()` で行いますので、本関数呼び出しは、`gp_init()` の後に行ってください。設定可能な最長ディレイ時間は 65500 マイクロ秒です。

書式 `int gp_count(void)`
 `int U2Gpcount(int hDriver)`

機能 GPIB 機器からの受信データ数または GPIB 機器へ送信完了したデータ数を取得します。関数 `gp_red(U2Gpred)`, `gp_tfrin(U2Gptfrin)`, `gp_tfrins(U2Gptfrins)` `gp_wrt(U2Gpwrt)`, `gp_tfrout(U2Gptfrou)` の後に呼び出すことで、実際にハンドシェイクが完了したデータ数を知ることができます。

引数 `hDriver` (IN) コンバータのハンドル(複数台接続時使用)

戻値 N 受信データ数または送信データ数が返されます。

補足 デリミタコードのカウンタは行いません。

書式

int **gp_delm**(PCHAR mode, int dlm)

int **U2Gpdelm**(int hDriver, PCHAR mode, int dlm)

機能 送信時(gp_wrt および U2Gpwrt)、受信時(gp_red および U2Gpred)のデリミタの設定を行ないます。初期設定では送信時デリミタは CR+LF、受信時デリミタは LF(0x0A)となっています。

引数

hDriver	(IN)	コンバータのハンドル(複数台接続時使用)
mode	(IN)	“ l ” で受信時、“ t ” で送信時の設定を行います。 “ b ” で受信時・送信時の設定を行います。
dlm	(IN)	デリミタコードを指定します。

戻値

0	正常終了
-1	ライブラリ DeviceIoControl()リクエストエラー

補足 ・初期設定は gp_init()で行いますので、本関数呼び出しは、gp_init()の後に行ってください。

・デリミタコード dlm については以下のような設定を行います。

(送信時) : mode = “ t ” での設定

Bit6~Bit0 の 7bit でデリミタコードを設定します。Bit7 を 1 に設定すると EOI を出力し、全ての bit を 0(dlm=0)にすると、CR+LF(0x0D + 0x0A)が設定されます。

(受信時) : mode = “ l ” での設定

Bit7~Bit0 の 8bit でデリミタコードを設定します。EOI 検出時は常にデリミタとして扱い、データ受信を終了します。

(送信・受信時) : mode = “ b ” での設定

dlm の値を以下のように設定することで、送信・受信時の設定を同時に行います。“ t ”

“ l ” で設定されたデリミタは無効となります。

dlm = 0x0400 デリミタなし

dlm = 0x000D CR

dlm = 0x000A LF

dlm = 0x0200 CR+LF

dlm = 0x0C00 EOI のみ

dlm = 0x080D CR+EOI (受信時は CR もしくは EOI で受信終了)

dlm = 0x080A LF+EOI (受信時は CR もしくは EOI で受信終了)

dlm = 0x0A00 CR+LF+EOI (受信時は CR+LF もしくは EOI で受信終了)

mode= “ b ” で送信時・受信時に異なる設定を行いたい場合は gp_wrt(), gp_red()関数呼び出し直前に、本関数で再設定を行ってください。

書式

int **gp_tfROUT**(PCHAR adrs, int bufLen, PCHAR buf)

int **U2GptfROUT**(int hDriver, PCHAR adrs, int bufLen, PCHAR buf)

機能 引数 adrs で指定した GPIB 機器に対してバイナリデータを送信します。デリミタは EOI のみです。

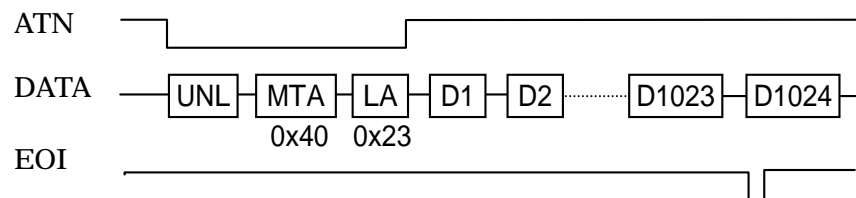
引数

hDriver	(IN)	コンバータのハンドル(複数台接続時使用)
adrs	(IN)	GPIB 機器アドレス
bufLen	(IN)	送信するデータの長さ
buf	(IN)	送信データを格納する配列の先頭アドレス。

戻値

0	正常終了
-1	ライブラリ DeviceIoControl()リクエストエラー
53	GPIB バスタイムアウト
63	GPIB 機器アドレス設定エラー
64	送信データ長設定エラー
-5, -6, -7	USB 転送時エラー

GPIB
バス



書式

int **gp_tfrin**(PCHAR adrs, int bufLen, PCHAR buf)

int **U2Gptfrin**(int hDriver, PCHAR adrs, int bufLen, PCHAR buf)

機能 引数 adrs で指定した GPIB 機器をトーカーに指定し、バイナリデータを受信します。デリミタは EOI のみです。EOI を受信するかバスタイムアウトになるまで制御を返しません。

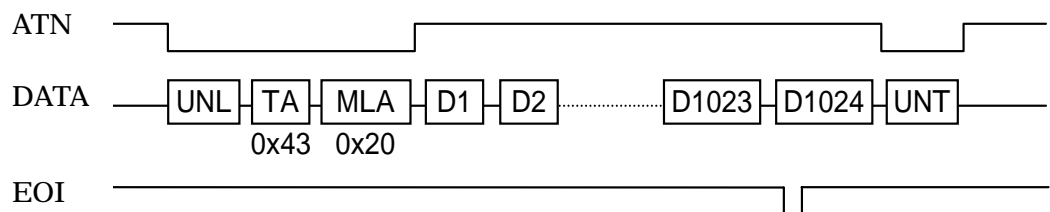
引数

hDriver	(IN)	コンバータのハンドル(複数台接続時使用)
adrs	(IN)	GPIB 機器アドレス
bufLen	(IN)	用意する配列数
buf	(OUT)	受信データを格納する配列の先頭アドレス

戻値

0	正常終了
-1	ライブラリ DeviceIoControl() リクエストエラー
53	GPIB バスタイムアウト
61	受信バッファオーバーフロー (EOI 受信しないまま、サイズ分を受信)
63	GPIB 機器アドレス設定エラー
64	受信用配列設定エラー
-5, -6, -7	USB 転送時エラー

GPIB
バス



書式

```
int gp_tfrinit(PCHAR adrs)
int U2Gptfrint(int hDriver, PCHAR adrs)
```

機能 GPIB 機器からバイナリデータを受信するためにトーカアドレスを指定します。
(gp_tfrins または U2Gptfrins, gp_tfrend または U2Gptfrend と共に使用します)

引数 hDriver (IN) コンバータのハンドル(複数台接続時使用)
adrs (IN) GPIB 機器アドレス

戻値 0 正常終了
-1 ライブラリ DeviceIoControl() リクエストエラー
53 GPIB バスタイムアウト
63 GPIB 機器アドレス設定エラー
-5, -6, -7 USB 転送時エラー

補足 受信すべきデータ数が不明な場合、関数 gp_tfrin(U2Gptfrint)の代わりに 3 つの関数 gp_tfrinit(U2Gptfrint), gp_tfrins(U2Gptfrins), gp_tfrend(U2Gptfrend)を組み合わせ使用し、データを受信することが可能です。gp_tfrins(U2Gptfrins)を繰り返して呼び出すことで、連続してデータを受信することができます。

(使用例) 機器アドレス 3 からデータを受信する場合。通常は gp_tfrins(U2Gptfrins)の戻り値より EOI 受信の有無を調べ、EOI 未受信であれば、再度を呼び出します。

```
unsigned char RxBuf[256];

gp_tfrinit("3"); // トーカ指定
gp_tfrins(256, RxBuf); // 256Byte データ受信
gp_tfrins(256, RxBuf);
gp_tfrins(256, RxBuf);
... (EOI 受信するまで繰り返し呼び出す)
gp_tfrend(); // トーカ指定解除
```

書式

int **gp_tfrins**(int bufLen, PCHAR buf)
int **U2Gptfrins**(int hDriver, int bufLen, PCHAR buf)

機能 GPIB 機器からバイナリデータを受信します。デリミタは EOI のみです。
(gp_tfrinit または U2Gptfrinit, gp_tfrend または U2Gptfrend と共に使用します)

引数 hDriver (IN) コンバータのハンドル(複数台接続時使用)
bufLen (IN) 受信するデータの長さ
buf (OUT) 受信データを格納する配列の先頭アドレス

戻値 0 指定サイズ分のデータを受信して正常終了
24 EOI を受信して正常終了
-1 ライブラリ DeviceIoControl() リクエストエラー
53 GPIB バスタイムアウト
64 受信用配列設定エラー
-5, -6, -7 USB 転送時エラー

書式

VOID **gp_tfrend**(void)
VOID **U2Gptfrend**(int hDriver)

機能 GPIB 機器からバイナリデータを受信するために指定したトーカアドレスを解除します。
(gp_tfrinit または U2Gptfrinit, gp_tfrins または U2Gptfrins と共に使用します)

引数 hDriver (IN) コンバータのハンドル(複数台接続時使用)

戻値 ありません。

書式

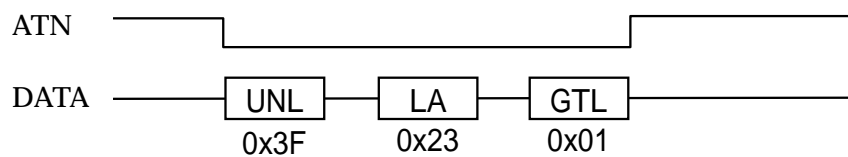
```
int gp_wtb(PCHAR buf)
int U2Gpwtb(int hDriver, PCHAR buf)
```

機能 ATN ラインを TRUE にしてコマンド文字列を送信します。コマンド文字列の最後に、データ終了を示す NULL コード (0x00) を指定してください。

引数 hDriver (IN) コンバータのハンドル(複数台接続時使用)
buf (IN) 送信文字列を格納するバッファアドレス

戻値 0 正常終了
-1 ライブラリ DeviceIoControl() リクエストエラー
53 GPIB バスタイムアウト
64 送信データ設定エラー
-5, -6, -7 USB 転送時エラー

GPIB
バス



書式

```
int gp_myadr(void)
int U2Gpmyadr(int hDriver)
```

機能 関数 gp_init(U2Gpinit) で設定された USB220 の GPIB 機器アドレスを取得します。プログラムで新たに USB220 の GPIB アドレスを知る必要が無い場合は、本関数を呼び出す必要はありません。

引数 hDriver (IN) コンバータのハンドル(複数台接続時使用)

戻値 N 正常終了時、USB220 の GPIB 機器アドレスが返されます。
-1 ライブラリ DeviceIoControl() リクエストエラー

その他の関数

書式 VOID **gp_wait**(int WaitSecTime)

機能 指定時間プログラムを停止させます。

引数 WaitSecTime (IN) プログラムを停止する時間(秒単位で指定)

戻値 ありません。

補助関数

書式 `int gp_srqCheck(void)`
 `int U2GpsrqCheck(int hDriver)`

機能 SRQ ラインの現在の状態を返します。

引数 `hDriver` (IN) コンバータのハンドル(複数台接続時使用)

戻値 1 SRQ ラインが TRUE
 0 SRQ ラインが FALSE
 -1 ライブラリ `DeviceIoControl()` リクエストエラー

書式

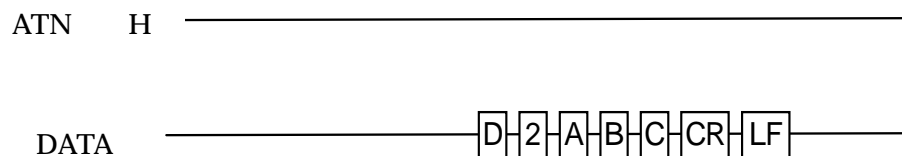
int **gp_wrt**(PCHAR buf, int bufLen)
int **U2Gpwrt**(int hDriver, PCHAR buf, int bufLen)

機能 GPIB バス上にデータ送信します。デリミタ指定関数 gp_delm および U2Gpdelm で指定されたデリミタを送信データに自動的に付加して送信を行います。

引数 hDriver (IN) コンバータのハンドル(複数台接続時使用)
buf (IN) 送信文字列を格納するバッファアドレス
bufLen (IN) 送信するデータサイズ

戻値 0 正常終了
-1 ライブラリ DeviceIoControl()リクエストエラー
53 GPIB バスタイムアウト
64 送信データ設定エラー
-5, -6, -7 USB 転送時エラー

補足 gp_wrt(U2Gpwrt)と異なる点は、データ送信前にコマンド送信をしない点です。通常、gp_wtb(U2Gpwtb)と組み合わせて使用します。



書式

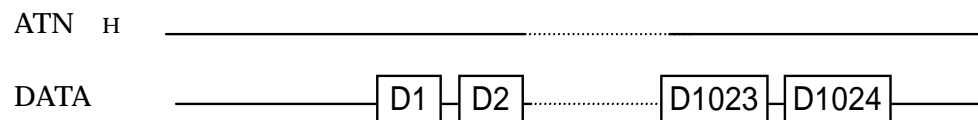
int **gp_tfroutd**(int bufLen, PCHAR buf)
int **U2Gptfroutd**(int hDriver, int bufLen, PCHAR buf)

機能 GPIB バス上にバイナリデータ送信します。

引数 hDriver (IN) コンバータのハンドル(複数台接続時使用)
bufLen (IN) 送信するデータサイズ
buf (IN) 送信データを格納する配列の先頭アドレス。

戻値 0 正常終了
-1 ライブラリ DeviceIoControl()リクエストエラー
53 GPIB バスタイムアウト
64 送信データ長設定エラー
-5, -6, -7 USB 転送時エラー

補足 gp_tfroutd(U2Gptfroutd)と異なる点は、データ送信前にコマンド送信をしない点です。
また、送信デリミタ EOI の有無は gp_delm(U2Gpdelm)の設定に従います。デリミタコードは付加しません。通常、gp_wtb(U2Gpwtb)と組み合わせて使用します。



書式

int **gp_redd**(PCHAR buf, int bufLen)

int **U2Gpredd**(int hDriver, PCHAR buf, int bufLen)

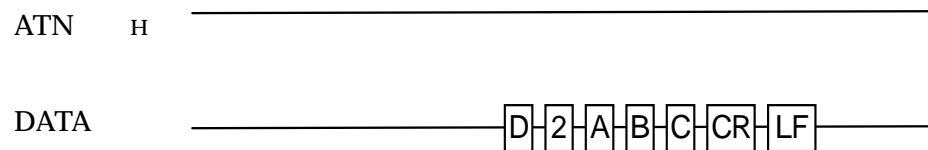
機能 GPIB 上からのデータ受信を行います。デリミタ指定関数 `gp_delm(U2Gpdelm)` で指定されたデリミタを受信するかバスタイムアウトになるまで制御を返しません。

注)アプリケーションにはデリミタコードを返しません。

引数 hDriver (IN) コンバータのハンドル(複数台接続時使用)
buf (OUT) 受信文字列を格納するバッファアドレス
bufLen (IN) 受信バッファのサイズ

戻値 0 指定サイズ分データ受信して正常終了
24 指定のデリミタを受信して正常終了
-1 ライブラリ `DeviceIoControl()` リクエストエラー
53 GPIB バスタイムアウト
64 受信バッファサイズ設定エラー
-5, -6, -7 USB 転送時エラー

補足 `gp_red(U2Gpred)` と異なる点は、データ受信前にコマンド送信をしない点と最後に UNT コマンドを送信しない点です。また、本関数から制御を戻したとき、RFD ホールドオフとなります。通常、`gp_wtb(U2Gpwtb)` と組み合わせて使用します。



書式

```
int gp_redah(PCHAR adrs, PCHAR buf, int bufLen)
int U2Gpredah(int hDriver, PCHAR adrs, PCHAR buf, int bufLen)
```

機能 引数 adrs で指定した GPIB 機器をトーカーに指定し、データの受信を行います。デリミタ指定関数 gp_delm(U2Gpdelm)で指定されたデリミタを受信するかバスタイムアウトになるまで制御を返しません。

注)アプリケーションにはデリミタコードを返しません。

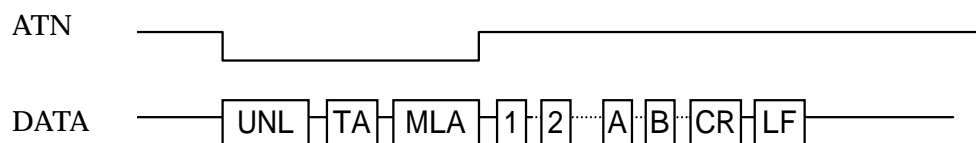
引数

hDriver	(IN)	コンバータのハンドル(複数台接続時使用)
adrs	(IN)	GPIB 機器アドレス
buf	(OUT)	受信文字列を格納するバッファアドレス
bufLen	(IN)	受信バッファのサイズ

戻値

0	正常終了
-1	ライブラリ DeviceIoControl()リクエストエラー
53	GPIB バスタイムアウト
61	バッファオーバーフロー(デリミタ受信しないまま、サイズ分を受信)
63	GPIB 機器アドレス設定エラー
64	受信用配列数設定エラー
-5, -6, -7	USB 転送時エラー

補足 gp_red(U2Gpred)と異なる点は、最後に UNT コマンドを送信しない点です。また、本関数から制御を戻したとき、RFD ホールドオフとなります。



書式 `int gp_redrst(void)`
 `int U2Gpredrst(int hDriver)`

機能 本関数は、リスナ解除、RFD ホールドオフの解除を行います。

引数 `hDriver` (IN) コンバータのハンドル(複数台接続時使用)

戻値 0 正常終了
 -1 ライブラリ `DeviceIoControl()` リクエストエラー

書式 `int gp_findlstn(PCHAR adrs, int adrsLen)`
 `int U2Gpfindlstn(int hDriver, PCHAR adrs, int adrsLen)`

機能 GPIB バスに接続されているリスナ機器を検出し、GPIB アドレスを取得します。

引数 `hDriver` (IN) コンバータのハンドル(複数台接続時使用)
 `adrs` (OUT) GPIB アドレスを格納するバッファアドレス
 `adrsLen` (IN) バッファのサイズ

戻値 0 リスナ未検出
 -1 ライブラリ `DeviceIoControl()` リクエストエラー
 53 GPIB バスタイムアウト
 64 アドレス格納用バッファサイズが正しくない
 N リスナ検出台数

補足 本関数では、取得した GPIB アドレスを ASCII データの形で `adrs` に格納します。取得した `adrs` を `gp_wrt()`、`gp_red()` 等の第一引数でそのまま使用できます。
戻り値に 64 が返る場合は、確保するバッファ `adrs` を大きめに確保してください。