

PW-A / PAR-A SERIES

Windows API・USB デバイスドライバ Ver. 3.10



＜ソフトウェア使用許諾契約＞

1. 権利の許諾

当社はお客様に対して、本使用許諾契約に同意いただいてダウンロード可能となるソフトウェア及びその関連資料(以下「本ソフトウェア」といいます)に関し、以下の権利を許諾します。

- (a) お客様は、本ソフトウェアに対応する当社製品を利用する目的で本ソフトウェアを使用することができます。
- (b) お客様は、本ソフトウェアを複製し、1台以上のコンピュータ上で使用することができます。

2. 追加許諾条項

本ソフトウェアを定められた目的に従って使用した結果、作成された各種のファイルは、お客様の著作物となります。

3. 著作権

本ソフトウェア及びその複製物の著作権は当社又は当社が認めた者が有するものであり、日本国著作権法及び国際条約によって保護されています。本使用許諾契約に基づき、お客様が本ソフトウェアを複製する場合は、ダウンロードされた本ソフトウェアに付されていたものと同じの著作権表示がなされることを要します。

4. 禁止事項

本ソフトウェアがソースコードで提供される場合、お客様は、ソースコードを改変したものを当社製として第三者に配布することはできません。

5. 無保証

当社は、本ソフトウェアがお客様特定の目的のために適切であること、もしくは有用であること、又は本ソフトウェアに瑕疵がないこと、その他本ソフトウェアに関していかなる保証もいたしません。

6. 免責

当社は、いかなる場合においても、本ソフトウェアの使用又は使用不能から生ずるいかなる損害(事業利益の損害、事業の中断、事業情報の損失、又はその他金銭的損害)に関して、一切責任を負いません。

7. 契約の解除

お客様が本使用許諾契約に違反した場合、当社は本使用許諾契約を解除することができます。その場合、お客様は本ソフトウェアを一切使用しないものとします。

＜サポート＞

本ソフトウェアの不具合等のお問い合わせは、当社サービスまでお願いいたします。

＜保証範囲＞

本ソフトウェアのダウンロード・インストールはお客様の責任においておこなっていただきます。また本ソフトウェアは、予告せず改良、変更することがあります。

＜著作権者＞

各ソフトウェアの著作権は、当社に帰属します。

＜各社商標＞

TEXIO は当社の産業用電子機器における製品ブランドです。また、本説明書に記載されている会社名および商品名は、それぞれの国と地域における各社および各団体の商標または登録商標です。

目 次

ソフトウェア使用許諾契約

第 1 章 概 要	1
第 2 章 動作環境	1
第 3 章 API のインストール	1
第 4 章 USB デバイスドライバのインストール	1
4-1. インストールについて	1
4-2. 再インストールについて	2
第 5 章 PC との接続方法	2
5-1. GP-IB による接続	2
5-2. USB による接続	2
5-3. RS-232C による接続	2
5-4. アドレスについて	2
5-5. アドレスの設定方法	3
5-6. アドレスの設定範囲	3
5-7. 利用上の注意について	3
5-8. 用語説明	4
5-9. 接続図	4
5-9-1. IF-41GU GP-IB 接続図	4
5-9-2. IF-41GU/USB USB 接続図	4
5-9-3. IF-41RS RS-232C 接続図	5
5-9-4. IF-41GU/USB ローカルバスの接続	5
第 6 章 関数一覧表	6
第 7 章 初期化関数	7
7-1. デバイスハンドルのオープン	7
7-2. デバイスハンドルのクローズ	7
7-3. タイムアウト時間の設定	8
7-4. リフレッシュ	8
7-5. 電源モデル名の取得	8
第 8 章 電源制御関数	9
8-1. プリセット・メモリへの電圧値設定	9
8-2. プリセット・メモリからの電圧値取得	9
8-3. プリセットメモリへの電流値設定	9
8-4. プリセットメモリからの電流値取得	10
8-5. メインアウトプットの状態設定	10
8-6. メインアウトプットの状態取得	10
8-7. デイレイ機能の設定	10

8-8. ディレイ機能の状態取得	11
8-9. アウトプットセレクトの設定	11
8-10. アウトプットセレクトの状態取得	11
8-11. トラッキング機能の設定	11
8-12. トラッキング機能の状態取得	12
8-13. トラッキングモードの設定	12
8-14. トラッキングモードの状態取得	12
8-15. トラッキングチャンネルの設定	12
8-16. トラッキングチャンネルの状態取得	13
8-17. トラッキングデータの設定	13
8-18. トラッキングデータ取得	13
8-19. ディレイ時間の設定	14
8-20. ディレイ時間設定の取得	14
8-21. ディスプレイ内容の設定	14
8-22. ディスプレイ内容の状態取得	15
8-23. プリセット番号の選択	15
8-24. プリセット番号の取得	15
8-25. 出力モニタ値の取得	15
8-26. システムアドレスの取得	16
8-27. ローカル設定	16
8-28. ローカルロックアウトの設定	16
8-29. データバックアップ	16
8-30. サービスリクエストの設定	17
8-31. 全てのプリセット値取得(1)	17
8-32. 全てのプリセット値取得(2)	18
第 9 章 コマンド送受信関数	19
9-1. 送受信コマンドについて	19
9-2. コマンド送信	19
9-3. 応答受信	19
第 10 章 利用上の注意について	20
第 11 章 電源制御用通信コマンド	21

第1章 概要

本ソフトウェアはPW-Aシリーズ、PAR-AシリーズのWindowsAPIを提供します。パーソナルコンピュータ(以下、PC)により、GP-IB、USBまたはRS-232C経由で制御するためのアプリケーション作成に利用できます。PCを使用して電源をコントロールするためには、Windowsが動作するためのPCと専用のインタフェースユニット(IF-40シリーズ)が必要です。

第2章 動作環境

- ・IF-40シリーズを装着したPW-AシリーズおよびPAR-Aシリーズ
- ・マイクロソフト社のWindows XP SP3(32bit)、VISTA(32bit)、Windows7以降(32bit / 64bit)が動作するPC。

対応言語として、Visual Basic6.0、Visual Basic for Applications、Visual C++6.0(C,C++)、VisualBasic.2005、VisualC++.2005、VisualC#.2005以後を想定しています。また、Windows7以降(64bit)で利用できる言語はVisualBasic2008、Visual C++2008、Visual C#2008以後を想定しています。

GP-IBについては、National Instruments社製のGP-IBドライバ、GP-IBカード、NI488.2M Software Ver2.8.1以降についてのみ対応しております。他社のGP-IBドライバではAPIは動作しません。

USBについては、当社のUSBデバイスドライバに対応しております。本ドライバはUSBのプラグアンドプレイに完全に適合しておりません。電源本体の電源を途中でON・OFFしたり、追加・削除した場合の動作は保証できませんのでご注意ください。また、USBのインタフェースはGP-IBと比較して手軽なだけ電氣的に弱く、ノイズや振動のある環境に適しておりません、十分なノイズ対策と検証を行った上でご使用ください。

電源を複数のプログラム、プロセス、スレッドから同時にコントロールする場合は通信が重ならないように十分検証をおこなった上でご使用ください。

第3章 APIのインストール

- ・PCへのログインは必ず管理者(administrator)としてください。
- ・適当なフォルダにダウンロードしたファイルを展開し、APIフォルダにあるSetup32.exe(32bit用)／Setup64.exe(64bit用)を右クリックで管理者として実行し、インストールしてください。
- ・USBを使用する場合は続けてDriverフォルダにあるSetup*.exeを実行し、デバイスドライバをインストールしてください。APIの利用方法はサンプルプログラムが用意しておりますのでこちらを参照してください。
- ・OSによってはインストール時に設定または続行の許可のメッセージが出ますが、そのまま続行してください。
- ・インストール中にフォルダの指定が出る場合がありますが、インストール用のフォルダは作成されません、ファイルは全てシステムフォルダにコピーされます。
- ・.netFramework4 ライブラリとVC++ランタイムライブラリのインストールが自動で行われます。システムと競合してエラーが表示され、インストールできない場合はライブラリを一旦削除してからインストールするかTEXIO_API*.msiを実行してください。
- ・再インストールは環境に応じてTEXIO_API32.msiまたは、TEXIO_API64.msiを実行して下さい。

第4章 USBデバイスドライバのインストール

4-1.インストールについて

- ・本体の電源スイッチをOFFにしたままでUSBケーブルを動作しているPCに接続してください。
- ・PCへのログインは必ず管理者(administrator)としてください。
- ・32ビット・64ビットの環境に合わせたDriverフォルダにあるSetup32.exe(32bit用)／Setup64.exe(64bit用)を右クリックで管理者として実行しインストールしてください。途中でマイクロソフトのロゴテストに合格していないとメッセージが出る場合がありますが、そのまま続行してください。
- ・インストールが完了してから、本体の電源スイッチをONしてPCが認識するのを待ちます。接続している機器の台数や状況により時間がかかる場合があるので注意してください。
- ・認識された場合、新しいハードウェアの検出ウィザードが始まります。自動でインストールを選択し、メッセージにしたがってデバイスドライバをインストールします。途中でマイクロソフトのロゴテストに合格していないとのメッセージが出る場合がありますが、そのまま続行してください。
- ・OSによってはインストール時に設定または続行の許可のメッセージが出ますが、そのまま続行してください。
- ・PCによってはセキュリティの関係で本器が自動認識されないことがあります。この場合はコントロールパネルのデバイスマネージャーで"ほかのデバイス"にある本器を右クリックで選択し、ドライバの更新を行ないます。ドライバファイルは手動で同梱のinfファイルを指定してください。

4-2.再インストールについて

- ・本ドライバを使用している全てのデバイスとの通信が停止している事を確認してください。
- ・「コントロールパネル→プログラムの追加と削除」からドライバを削除してください。
(Windowsドライバパッケージ (WinUSB)USB Driver のうち当社が発行したもの)
- ・ドライバを Setup**.exe で再度インストールしてください。
- ・デバイスマネージャで、接続している本体が正常に認識・動作していることを確認してください。

第 5 章 PCとの接続方法

専用インタフェースユニット(以下、IF-41)を装着することで、GP-IB、USB および RS-232C により、PC から電源をコントロールできます。接続できる機種は PW-A シリーズ、PAR-A シリーズのみとなります。

IF-41 には、以下の 3 種類があります。

- ・ IF-41GU — GP-IB、USB 両方のインタフェースを備えたユニット。
- ・ IF-41USB — USB インタフェースを備えたユニット。
- ・ IF-41RS — RS-232C インタフェースを備えたユニット。

旧バージョンである IF-40 シリーズについては機種 ID が 10 以下の PW-A シリーズのみ動作をさせることが可能です、この場合は本マニュアルの IF-41 を IF-40 と読み替えて利用してください。また本体のバージョンが 11.00 以下の場合、IF-41 シリーズは IF-40 シリーズとして認識されますが、動作自体に問題はありません。

5-1. GP-IB による接続

- ・ IF-41GU 及び GP-IB ケーブルを使用して PC と接続します。
- ・ PC と直接接続可能なローカルバスマスターの台数は 14 台です。
- ・ ローカルバスを使用することにより、それぞれのローカルバスマスターに 31 台までスレーブ機を接続することが出来ます。

5-2. USB による接続

- ・ IF-41GU または、IF-41USB、及び、USB ケーブルを使用して PC と接続します。
- ・ PC と直接接続可能なローカルバスマスターの台数は 32 台です。
- ・ ローカルバスを使用することにより、それぞれのローカルバスマスターに 31 台までスレーブ機を接続可能です。

5-3. RS-232C による接続

- ・ IF-41RS 及び、RS-232C ケーブル(インターリンククロスケーブル)を使用して PC と接続します。ケーブルについては電源本体の取扱説明書を参照してください。PC と直接接続可能なローカルバスマスターの台数は 9 台です。
- ・ ローカルバスを使用することにより、それぞれのローカルバスマスターに 3 台までスレーブ機を接続可能です。
- ・ 64bit 環境では正常に通信できない場合が有ります。通信が出来ない場合は、API を使用せず直接プログラムから通信を行ってください。

5-4. アドレスについて

PC 及び、ローカルバスに接続される全ての電源は、必ず、アドレスの設定をする必要があります。アドレスにはシステムアドレスと PC アドレスの 2 種類があります。

- | | |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ・PC アドレス | — IF-41GU/USB を使用する場合に設定するアドレス。
GP-IB、または、USB で PC と接続するローカルバスマスターに、PC による識別用のパラメータとして設定する必要があります。スレーブ機をオープンする場合、当該スレーブ機が接続されているローカルバスマスターの PC アドレスを指定します。
IF-41RS を使用する場合、PC アドレスの設定は不要です。 |
| ・システムアドレス | — ローカルバスマスター及び、スレーブ機に設定されるアドレス。
ローカルバスマスターによるスレーブ機識別用のパラメータとして設定する必要があります。 |

5-5. アドレスの設定方法

以下の手順に従って、各アドレスを正しく設定してください。

- ① 電源本体の「A」KEY を押しながら電源スイッチを ON します。電流表示 LED の表示が “P-on” から “Idno” に変わるまで「A」KEY は押し続けてください。
ROM バージョンの表示に引き続いて、電流表示 LED が “SyAd”、電圧表示 LED に現在のシステムアドレスが表示されることを確認してください。
- ② ロータリーエンコーダを回転させて表示されているシステムアドレスを希望の値に変更し、MEMORY KEY を押してください。IF-41RS を使用する場合、及び、IF-41GU/USB を使用してシステムアドレスを「1」以外に設定した場合、PC アドレス設定には移行せず、ここで通常表示に戻ります。
- ③ IF-41GU/USB を使用し、システムアドレスを「1」(ローカルバスマスター)に設定した場合、引き続いて、電流表示 LED が “PcAd”、電圧表示 LED に PC アドレスが表示されます。
システムアドレス同様にロータリーエンコーダを回転させて表示されている PC アドレスを希望の値に変更し、MEMORY KEY を押してください。設定モードが終了し、通常表示に戻ります。

5-6. アドレスの設定範囲

- ・アドレスの設定可能範囲は使用する IF-41 によって異なります。下表を参照し、正しく設定してください。
- ・各アドレスの設定範囲は接続可能台数ではありません。
- ・工場出荷時またはメモリ初期化をした場合、システムアドレスおよび PC アドレスの設定は「1」となります。

使用ユニット名	接続方法	システムアドレス	PC アドレス
IF-41GU	GP-IB によるコントロール (スレーブ機)	2～32	—
	(ローカルバスマスター)	1	0～31
IF-41GU IF-41USB	USB によるコントロール (スレーブ機)	2～32	—
	(ローカルバスマスター)	1	0～31
IF-41RS	RS-232C によるコントロール (スレーブ機)	2～26	—
	(ローカルバスマスター)	1	—

5-7. 利用上の注意について

- ・ IF-41GU/USB を使用する場合、複数のローカルバスマスターに重複した PC アドレスを設定しないでください。また、同一ローカルバスに接続されているスレーブ機に重複したシステムアドレスを設定しないでください。正常に動作しません
- ・ GP-IB で使用する場合、PC アドレスの設定を 31 に設定した機器はリッスンオンリになります。
- ・ 電源本体がオーバーヒートアラーム(OHA)状態、または外部接点によりアラーム状態となっている場合、ローカル切替え、及び、各種の状態取得コマンド以外は受け付けなくなりますのでご注意ください。
- ・ ローカルバスは、バスマスター機がコマンドを解釈してから、スレーブ機にコマンドを転送します。このため実際のコマンド実行まで遅延があるため注意が必要です。また短い時間で設定変更を繰り返し行くと、本体の動作速度が低下する場合がありますのでコマンド送信間隔には十分注意してください。
- ・ PC の CPU クロックが高速な場合に、連続して設定や読み取りを行うと、本体の動作速度が遅くなる場合があります。また同じ設定のオン・オフを繰り返し行くと、処理が終わる前に次の設定が行われるため、設定が省略されて見えることがあります。特に内部のステータスやモニタデータの読み出しについては、数百 ms の読取周期がありますので注意が必要です。
- ・ 本 API は、サスペンドなどの PC の省電力モードに正式に対応しておりません。省電力モード以後の本機の動作は不定となりますので、設定には十分注意してください。

5-8. 用語説明

- ・ローカルバスマスター
 - － PC と接続され、スレーブ機をコントロールする電源。
 - ・スレーブ機
 - － ローカルバスを使用して接続される電源。
 - ・ローカルバス
 - － 電源間を接続するバス。
- IF-41GU/USB を装着した場合は、ツイストペアケーブルを使用します。
- IF-41RS を装着した場合は、専用接続ケーブルを使用します。
- ・マスター機
 - － ローカルバスマスターと同義。

5-9. 接続図

5-9-1. IF-41GU GP-IB 接続図

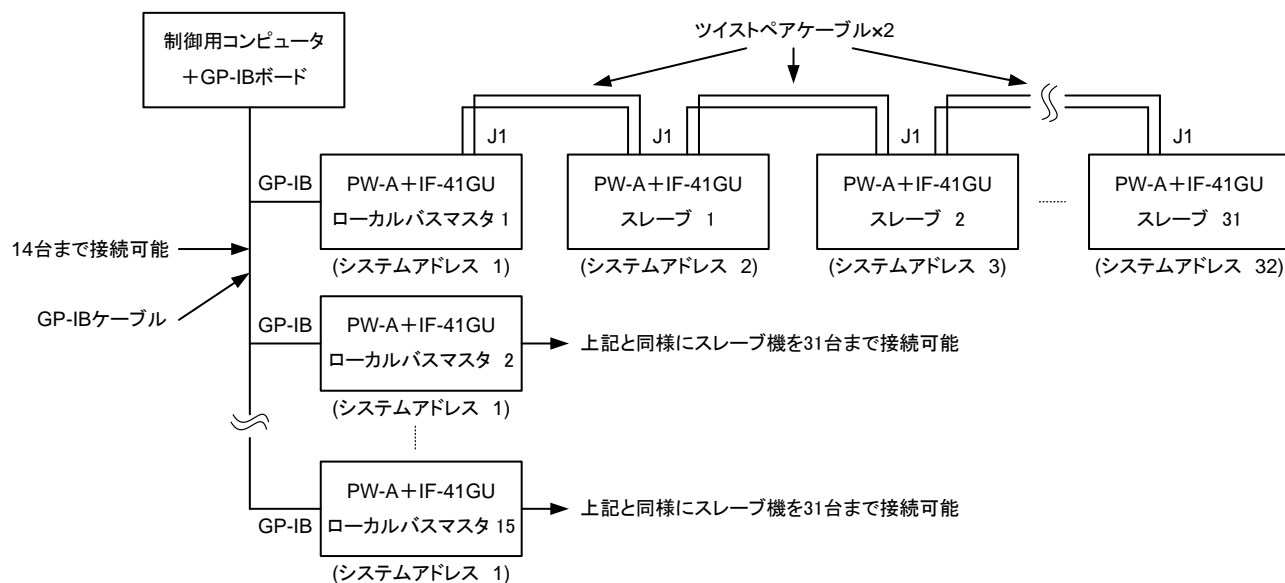


図 5-1 IF-41GU を使った GP-IB によるコントロール

5-9-2. IF-41GU/USB USB 接続図

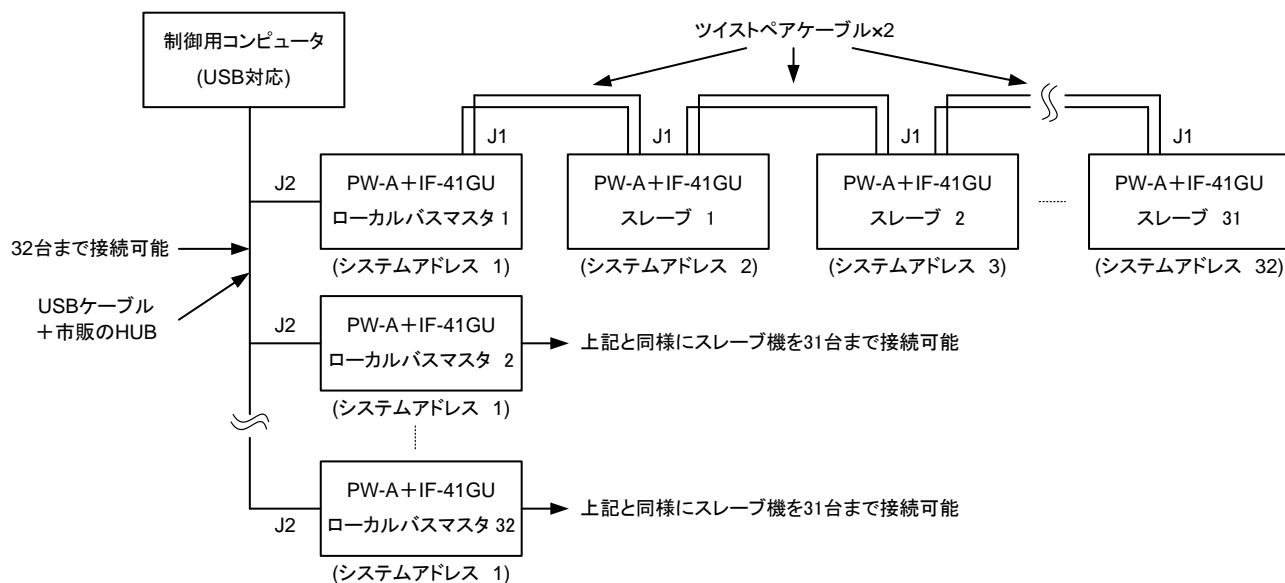


図 5-2 IF-41GU/USB を使った USB によるコントロール

5-9-3. IF-41RS RS-232C 接続図

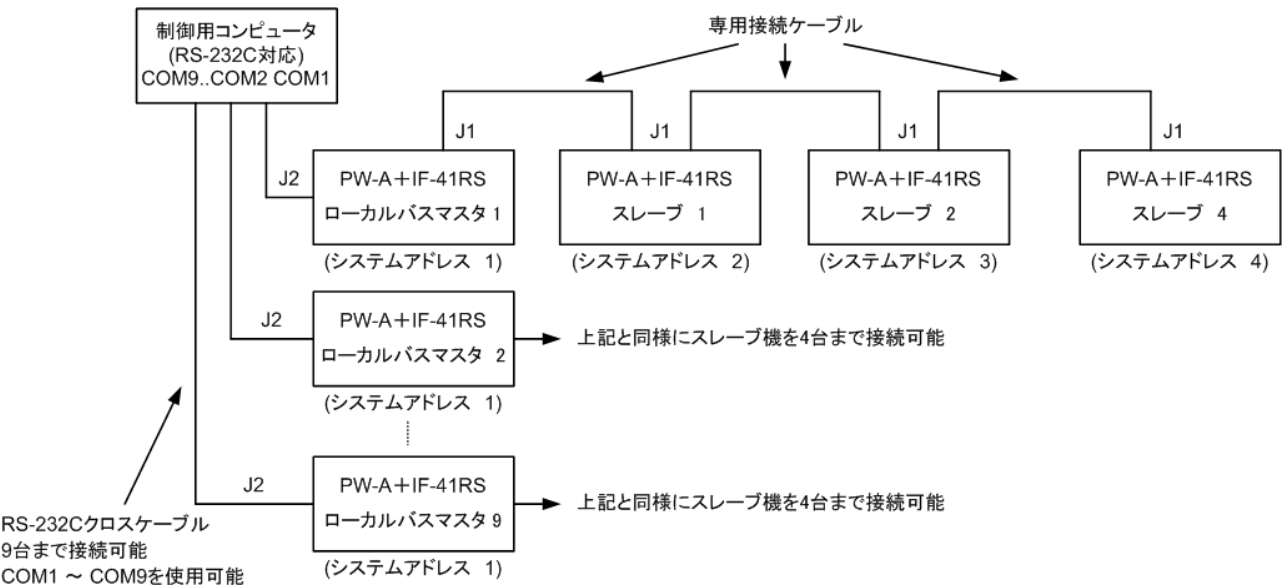


図 5-3 IF-41RS を使った RS-232C によるコントロール

5-9-4. IF-41GU/USB ローカルバスの接続

- ローカルバス接続用のツイストペアケーブルは、以下の注意点を守って接続してください。
 - ① バスの全長は、200m以内かつ 50Ω 以下としてください。
 - ② ローカルバスマスターとバスの最後に接続された IF-41GU/USB のみ、S1 を両方オンにしてください。
S1... 終端抵抗のオン/オフ切替えスイッチです。レバーを押し上げた状態が ON です。
 - ③ IF-41GU/USB 部の J1 に挿入する線材は、以下の範囲内としてください。

* 単線.....	φ0.4～φ1.2
* より線.....	0.3m ² ～1.2m ² /AWG16～22
	素線径 φ0.18 以上

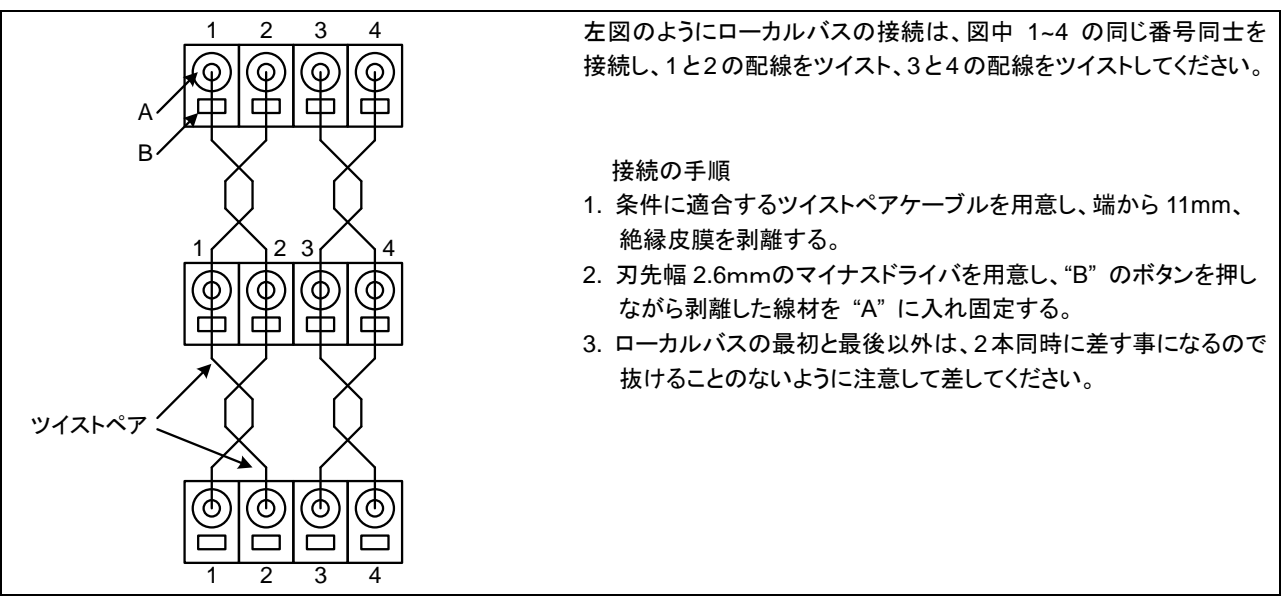


図 5-4 IF-41GU/USB ローカルバスの接続方法

第 6 章 関数一覧表

API で用意されている関数一覧

関数	内容
long TMI_HandleOpen (char &str , char &ptr)	デバイスのオープン
int TMI_HandleClose (long hID)	デバイスのクローズ
int TMI_TimeOut (long hID , int Time)	タイムアウト時間設定
int TMI_Refresh (long hID)	リフレッシュ動作
int TMI_ModelNameQ (long hID , char &Model)	モデル名の取得
int TMI_Voltage (long hID , char ch , char preset , double Voltage)	出力位置とプリセット位置を指定して電圧値を設定する
int TMI_VoltageQ (long hID , char ch , char preset , double &Voltage)	出力位置とプリセット位置を指定して電圧値を取得する
int TMI_Current (long hID , char ch , char preset , double Current)	出力位置とプリセット位置を指定して電流値を設定する
int TMI_CurrentQ (long hID , char ch , char preset , double &Current)	出力位置とプリセット位置を指定して電流値を取得する
int TMI_MainOutput (long hID , char onoff)	MainOutput の ON OFF を行う
int TMI_MainOutputQ (long hID , char &ans)	MainOutput の状態を取得する
int TMI_Delay (long hID , char onoff)	Delay の ON OFF を行う
int TMI_DelayQ (long hID , char &ans)	Delay の状態を取得する
int TMI_OutputSel (long hID , char ch , char onoff)	OutputSelect の出力別 ON OFF を行う
int TMI_OutputSelQ (long hID , char ch , char &ans)	OutputSelect の出力別の状態を取得する
int TMI_TrackingOnOff (long hID , char onoff)	Tracking の ON OFF を行う
int TMI_TrackingOnOffQ (long hID , char &ans)	Tracking の状態を取得する
int TMI_TrackingMode (long hID , char mode)	TrackingMode を設定する
int TMI_TrackingModeQ (long hID , char &ans)	TrackingMode の状態を取得する
int TMI_TrackingGroup (long hID , char ch ,char TrackingSet)	TrackingGroup を出力別に設定する
int TMI_TrackingGroupQ (long hID , char ch ,char &ans)	TrackingGroup の出力別設定状態を取得する
int TMI_TrackingData (long hID , char ch , char va , double Data)	TrackingGroup の電圧・電流値を出力別に増減させる
int TMI_TrackingDataQ (long hID , char ch , char va , double &Data)	TrackingGroup の電圧・電流値を出力別に取得する
int TMI_DelayTime (long hID , char ch , double Data)	DELAY 時間を出力別に設定する
int TMI_DelayTimeQ (long hID , char ch ,double &Data)	DELAY 時間を出力別に取得する
int TMI_Display (long hID , char ch)	Display 表示位置を設定する
int TMI_DisplayQ (long hID , char &ans)	Display 表示位置を取得する
int TMI_Preset (long hID , char preset)	Preset 番号を設定する
int TMI_PresetQ (long hID , char &ans)	Preset 番号を取得する
int TMI_MoniDataQ (long hID , char ch ,double &Voltage , double &Current , char &cv_cc)	指定した出力の電圧・電流モニタ値と CV/CC 状態を取得する
int TMI_AdrQ (long hID , char &ans)	システムアドレスを取得する
int TMI_RemoteLocal (long hID)	リモートからローカルへ切り替える
int TMI_LocalRockOut (long hID)	ローカルロックアウト状態へ移行する
int TMI_DataBackUp (long hID)	データのバックアップを行う
int TMI_SRQEnable (long hID , char onoff)	サービスリクエストの禁止・許可の設定をする
int TMI_AllPresetQ (long hID , double &Preset)	全出力の PRESET 値の取得
int TMI_AllPresetQS (long hID , char &Preset)	全出力の PRESET 値の取得
int TMI_Out (long hID , char &str)	コントロール関数: コマンド送信
int TMI_In (long hID , char &Query)	コントロール関数: コマンド受信

ここでは long:32bit、int:16bit、double:64bit となっています。定義は開発環境に応じて変更して下さい。

VC++より API を利用する場合は、DLL インポート用のサンプルを利用します。サンプルでは関数名が “TMI_” をとった名前で登録してあります。

VisualBasic6.0 より API を利用する場合は、関数定義が Declare.bas にありますので参照してください。

また VisualBasic2005 以後の場合は、Declare.vb に関数定義があります。

第7章 初期化関数

7-1. デバイスハンドルのオープン

long TMI_HandleOpen(char &model , char &set)

関数名 : TMI_HandleOpen

引 数 : char &model: 電源名指定の文字列: "PW-A"

char &set: IF 種類、PC アドレス、システムアドレスを指定する文字列
(「IF 種類」+ ":" +「PC アドレス」+ ":" +「システムアドレス」)

IF 種類:	GP-IB の場合	— DEV0～DEV3
	USB の場合	— USB
	RS-232C の場	— COM1～COM9

戻り値 : ID 番号。エラー時は負数が戻ります。

解 説 : 電源名 (PW-A)、IF の種類、PC アドレス、システムアドレスを指定して、GP-IB ボード、USB ポート、RS-232C の通信ポートを OPEN し、デバイスハンドルのオープンを行います。

デバイスハンドルのオープン後は、取得した ID 番号で電源情報を管理します。

注 意 : GP-IB の場合、Bord No.の、0～3 まで対応しています。

RS-232C の場合、通信ポートの、COM1～COM9 まで対応しています。

RS-232C の場合、PC アドレスの指定は無効です。常に「0」としてください。

高速で OPEN/CLOSE を繰り返したり、OPEN 中に本体の電源を切ったりすると、その後の通信が不安定になりますので注意が必要です。

書式例(VB):

GP-IB の BordNo. = 0, PC アドレス = 1, システムアドレス = 1 のオープン

hID = TMI_HandleOpen("PW-A", "DEV0:1:1")

USB の PC アドレス = 2, システムアドレス = 3 のオープン

hID = TMI_HandleOpen("PW-A", "USB:2:3")

RS-232C の COM1、システムアドレス = 1 のオープン

hID = TMI_HandleOpen("PW-A", "COM1:0:1")

7-2. デバイスハンドルのクローズ

int TMI_HandleClose(long hID)

関数名 : TMI_HandleClose

引 数 : long hID: ID 番号

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上

解 説 : GP-IB ボードの場合はオフラインに、USB ポート、RS-232C 通信ポートの場合は当該ポートをクローズし、指定されたデバイスハンドルのクローズします。

デバイスハンドルのオープンとクローズは必ず 1 対 1 で行ってください。

書式例(VB):

i = TMI_HandleClose(hID)

指定したデバイスハンドルのクローズします。

7-3. タイムアウト時間の設定

int TMI_TimeOut(long hID, int Time)

関数名 : TMI_TimeOut

引 数 : long hID: ID 番号

int Time: タイムアウト設定値

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上

解 説 : GP-IB の場合: タイムアウト時間を指定された時間コードで設定します。

何も設定しない場合は、DEFAULT 値=T1S(1 秒)となります。

USB、RS-232C の場合: タイムアウト時間を直接設定します。

0 で無限大、1-65535 まで 1 秒単位で指定します。

注 意 : GP-IB の場合 NI 488.2SoftWare で指定された時間コードを入力してください。

ローカルバスを利用している場合は応答までの時間が長いいため十分な時間(5 秒以上)に設定してください。

書式例(VB):

指定した電源の通信ポートに対して、タイムアウト時間を 1 秒に設定します。

i = TMI_TimeOut(hID, 11) (GP-IB の場合)

i = TMI_TimeOut(hID, 1) (USB、RS-232C の場合)

7-4. リフレッシュ

int TMI_Refresh(long hID)

関数名 : TMI_Refresh

引 数 : long hID: ID 番号

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上

解 説 : 通信バッファのリフレッシュ動作を行います。

書式例(VB):

i = TMI_Refresh(hID)

7-5. 電源モデル名の取得

int TMI_ModelNameQ(long hID, char &Model)

関数名 : TMI_ModelNameQ

引 数 : long hID: ID 番号

char &Model: モデル名格納用バッファのアドレス。

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上

解 説 : 電源のモデル名を取得します。

書式例(VB):

Dim Model As String

Model = String\$(20, 0)

i = TMI_ModelNameQ(hID, Model)

モデル名を取得します。

第 8 章 電源制御関数

8-1. プリセット・メモリへの電圧値設定

```
int TMI_Voltage( long hID , char ch , char preset , double voltage )
```

関数名 : TMI_Voltage

引 数 : long hID: ID 番号

char ch: 出力 CH 位置 A-D(1-4)

char preset: PRESET 番号(1-4)

double voltage: 電圧設定値

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、
出力範囲外=-3、成功=0 以上

解 説 : 指定した出力の CH,PRESET 番号に電圧値設定します。

注 意 : 設定した値は、バックアップメモリには書き込まれません。
必要に応じて、データのバックアップを実行してください。

書式例(VB):

```
i = TMI_Voltage( hID, 1, 1, 5.00 )
```

出力 CHA、PRESET1 の電圧値を 5.00V に設定します。

8-2. プリセット・メモリからの電圧値取得

```
int TMI_VoltageQ( long hID , char ch , char preset , double &voltage )
```

関数名 : TMI_VoltageQ

引 数 : long hID: ID 番号

char CH: 出力 CH 位置 A-D(1-4)

char preset: PRESET 番号(1-4)

double &voltage: 電圧値取得用バッファのアドレス

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、
出力範囲外=-3、成功=0 以上

解 説 : 指定した出力 CH, PRESET 番号の電圧値を取得します。

書式例(VB):

```
i = TMI_VoltageQ( hID, 1, 1, voltage )
```

出力 CHA,PRESET1 の電圧設定値を取得します。

8-3. プリセットメモリへの電流値設定

```
int TMI_Current( long hID , char ch , char preset , double current )
```

関数名 : TMI_Current

引 数 : long hID: ID 番号

char ch: 出力 CH 位置 A-D(1-4)

char preset: PRESET 番号(1-4)

double current: 電流設定値

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、
出力範囲外=-3、成功=0 以上

解 説 : 指定した出力 CH,PRESET 番号に電流値を設定します。

注 意 : 設定した値は、バックアップメモリには書き込まれません。
必要に応じて、データのバックアップを実行してください。

書式例(VB)

```
i = TMI_Current( hID, 1, 1, 1.00 )
```

出力 CHA,PRESET1 の電流値を 1.00A に設定します。

8-4. プリセットメモリからの電流値取得

int TMI_CurrentQ(long hID , char ch , char preset , double ¤t)

関数名 : TMI_VoltageQ

引 数 : long hID: ID 番号

char CH: 出力 CH 位置 A-D(1-4)

char preset: PRESET 番号(1-4)

double ¤t: 電流設定値格納用バッファのアドレス

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、
出力範囲外=-3、成功=0 以上

解 説 : 指定した出力 CH, PRESET 番号の電流値を取得します。

書式例(VB):

i = TMI_CurrentQ(hID, 1, 1, current)

出力 CHA,PRESET1 の電流設定値を取得します。

8-5. メインアウトプットの状態設定

int TMI_MainOutput(long hID , char onoff)

関数名 : TMI_MainOutput

引 数 : long hID: ID 番号

char onoff: メインアウトプットの設定 (ON = 1, OFF = 0)

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上

解 説 : メインアウトプットの状態設定(ON/OFF)を行います。

書式例(VB):

i = TMI_MainOutput(hID, 1)

メインアウトプットをONにします。

8-6. メインアウトプットの状態取得

int TMI_MainOutputQ(long hID , char &ans)

関数名 : TMI_MainOutputQ

引 数 : long hID: ID 番号

char &ans: メインアウトプット状態取得バッファのアドレス (ON = 1, OFF = 0)

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上

解 説 : メインアウトプットの状態を取得します。

書式例(VB):

i = TMI_MainOutputQ(hID, ans)

指定した電源のメインアウトプットの状態を取得します。

8-7. デイレイ機能の設定

int TMI_Delay(long hID , char onoff)

関数名 : TMI_Delay

引 数 : long hID: ID 番号

char onoff: デイレイ機能の設定(ON = 1,OFF = 0)

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上

解 説 : デイレイ機能の設定(ON/OFF)を行います。

注 意 : PAR-A ではエラーとなります。

書式例(VB):

i = TMI_Delay(hID, 1)

デイレイ機能をONに設定します。

8-8. ディレイ機能の状態取得

```
int TMI_DelayQ( long hID , char &ans )
```

関数名 : TMI_DelayQ

引 数 : long hID: ID 番号

char &ans: ディレイ機能の状態取得バッファのアドレス、(ON = 1, OFF = 0)

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上

解 説 : ディレイ機能の状態(ON/OFF)を取得します。

注 意 : PAR-A ではエラーとなります。

書式例(VB):

```
i = TMI_DelayQ( hID, ans )
```

ディレイ機能の状態を取得します。

8-9. アウトプットセレクトの設定

```
int TMI_OutputSel( long hID , char ch , char onoff )
```

関数名 : TMI_OutputSel

引 数 : long hID: ID 番号

char ch: 出力 CH 位置 A-D(1-4)

char onoff: アウトプットセレクトの設定(ON = 1, OFF = 0)

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、

出力範囲外=-3、成功=0 以上

解説 : 指定した出力 CH のアウトプットセレクト設定(ON/OFF)を行います。

書式例(VB):

```
i = TMI_OutputSel( hID, 1, 1 )
```

出力 CHA のアウトプットセレクトを ON に設定します。

8-10. アウトプットセレクトの状態取得

```
int TMI_OutputSelQ( long hID , char ch , char &ans )
```

関数名 : TMI_OutputSelQ

引 数 : long hID: ID 番号

char ch: 出力 CH 位置 A-D(1-4)

char &ans: アウトプットセレクト状態取得バッファのアドレス (ON = 1, OFF = 0)

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、

出力範囲外=-3、成功=0 以上

解 説 : 指定した出力 CH のアウトプットセレクトの状態(ON/OFF)を取得します。

書式例(VB):

```
i = TMI_OutputSelQ( hID, 1, ans )
```

出力 CHA のアウトプットセレクト状態を取得します。

8-11. トラッキング機能の設定

```
int TMI_TrackingOnOff( long hID , char onoff )
```

関数名 : TMI_TrackingOnOff

引 数 : long hID: ID 番号

char onoff: トラッキング機能の設定(ON = 1, OFF = 0)

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上

解 説 : トラッキング機能の設定(ON/OFF)を行います。

注 意 : PAR-A ではエラーとなります。

書式例(VB):

```
i = TMI_TrackingOnOff( hID, 1 )
```

トラッキング機能を ON に設定します。

8-12. トラッキング機能の状態取得

int TMI_TrackingOnOffQ(long hID, char &ans)

関数名 : TMI_TrackingOnOffQ

引 数 : long hID: ID 番号

char &ans: トラッキング機能の状態取得バッファのアドレス (ON = 1, OFF = 0)

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上

解 説 : トラッキング機能の状態を取得します。

注 意 : PAR-A ではエラーとなります。

書式例(VB):

i = TMI_TrackingOnOffQ(hID, ans)

トラッキング機能の状態を取得します。

8-13. トラッキングモードの設定

int TMI_TrackingMode(long hID , char mode)

関数名 : TMI_TrackingMode

引 数 : long hID: ID 番号

char mode: トラッキングモードの設定(絶対値 = 0、相対値 = 1)

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上

解 説 : トラッキングモードの設定(絶対値/相対値)を行います。

注 意 : PAR-A ではエラーとなります。

書式例(VB):

i = TMI_TrackingMode(hID, 0)

トラッキングモードを「絶対値」に設定します。

8-14. トラッキングモードの状態取得

int TMI_TrackingModeQ(long hID , char &ans)

関数名 : TMI_TrackingModeQ

引 数 : long hID: ID 番号

char &ans: トラッキングモードの状態取得バッファのアドレス (絶対値 = 0, 相対値 = 1)

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上

解 説 : トラッキングモードの状態を取得します。

注 意 : PAR-A ではエラーとなります。

書式例(VB):

i = TMI_TrackingModeQ(hID, ans)

トラッキングモードの状態を取得します。

8-15. トラッキングチャンネルの設定

int TMI_TrackingGroup(long hID , char ch , char set)

関数名 : TMI_TrackingGroup

引 数 : long hID: ID 番号

char ch: 出力 CH 位置 A-D(1-4)

char set: トラッキング設定(OFF = 0, +方向 = 1, -方向 = 2)

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、

出力範囲外=-3、成功=0 以上

解 説 : トラッキングチャンネルの設定を行います。

注 意 : PAR-A ではエラーとなります。

書式例(VB):

i = TMI_TrackingGroup(hID, 1, 1)

出力 CHA を+方向のトラッキングに設定する。

8-16. トラッキングチャンネルの状態取得

int TMI_TrackingGroupQ(long hID , char ch , char &ans)

関数名 : TMI_TrackingGroupQ

引数 : long hID: ID 番号

char ch: 出力 CH 位置 A-D(1-4)

char &ans: トラッキングチャンネル状態取得バッファのアドレス
(OFF = 0 , +方向 = 1 , -方向 = 2)

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、
出力範囲外=-3、成功=0 以上

解説 : トラッキングチャンネルの状態を取得します。

注意 : PAR-A ではエラーとなります。

書式例(VB):

i = TMI_TrackingGroupQ(hID, 1, ans)

出力 CHA についてトラッキングチャンネルの状態を取得します。

8-17. トラッキングデータの設定

int TMI_TrackingData(long hID , char ch , char va , double data)

関数名 : TMI_TrackingData

引数 : long hID: ID 番号

char ch: 出力 CH 位置 A-D(1-4)

char va: 設定データ種別(電圧指定 = 0、電流指定 = 1)

double data: トラッキング設定データ

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、
出力範囲外=-3、成功=0 以上

解説 : 指定したトラッキングチャンネルの設定状態に従って、現在の電圧または、電流値を指定したデータ分、増減させます。

データの増減は、トラッキングモードの設定により、絶対値または相対値で行われます。

注意 : 指定した CH のトラッキング設定が OFF の場合、その CH の設定値のみ、指定した設定データによって増減します。
PAR-A ではエラーとなります

書式例(VB):

絶対値の場合。

i = TMI_TrackingData(hID, 1, 0, +5.00)

トラッキングの電圧値を+5.00V 変化させます。

相対値の場合。

i = TMI_TrackingData(hID, 1, 0, +50.0)

トラッキングの電圧値を+50.0%増加させます。

8-18. トラッキングデータ取得

int TMI_TrackingDataQ(long hID , char ch , char va , double &data)

関数名 : TMI_TrackingDataQ

引数 : long hID: ID 番号

char ch: 出力 CH 位置 A-D(1-4)

char va: 取得データ種別(電圧指定 = 0、電流指定 = 1)

double &data: トラッキングデータ取得バッファのアドレス

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、
出力範囲外=-3、成功=0 以上

解説 : 指定した CH の指定した設定データを取得します。

注意 : PAR-A ではエラーとなります。

書式例(VB):

i = TMI_TrackingDataQ(hID, 1, 0, data)

出力 CHA の電圧値を取得します。

8-19. デレイ時間の設定

int TMI_DelayTime(long hID , char ch , double data)

関数名 : TMI_DelayTime

引 数 : long hID: ID 番号

char ch: 出力 CH 位置 A-D(1-4)

double data: デレイ時間 (0.0 秒~10.0 秒 設定単位: 0.1 秒)

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、
出力範囲外=-3、成功=0 以上

解 説 : 指定した CH のデレイ時間を設定します。

注 意 : PAR-A ではエラーとなります。

書式例(VB):

i = TMI_DelayTime(hID, 1, 5.0)

出力 CHA のデレイ時間を 5.0 秒に設定します。

8-20. デレイ時間設定の取得

int TMI_DelayTimeQ(long hID , char ch , double &data)

関数名 : TMI_DelayTimeQ

引 数 : long hID: ID 番号

char ch: 出力 CH 位置 A-D(1-4)

double &data: デレイ時間取得バッファのアドレス。

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、
出力範囲外=-3、成功=0 以上

解 説 : 指定した CH のデレイ時間設定を取得します。

注 意 : PAR-A ではエラーとなります。

書式例(VB):

i = TMI_DelayTimeQ(hID, 1, data)

出力 CHA のデレイ時間設定を取得します。

8-21. ディスプレイ内容の設定

int TMI_Display(long hID , char ch)

関数名 : TMI_Display

引 数 : long hID: ID 番号

char ch: 表示出力 CH 位置 A-D(1-4)

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、
出力範囲外=-3、成功=0 以上

解 説 : LED ディスプレイに表示する出力 CH の設定を行います。

注 意 : ディスプレイ内容の設定を行った後、出力電圧、電流を取り込むまで表示が 0 になる場合があります。
表示の確認は 1 秒以上待ってから行ってください。

書式例(VB):

i = TMI_Display(hID, 1)

出力 CHA を LED ディスプレイ表示に設定します。

8-22. ディスプレイ内容の状態取得

int TMI_DisplayQ(long hID , char &ans)
関数名 : TMI_DisplayQ
引 数 : long hID: ID 番号
char &ans: ディスプレイ内容の状態取得バッファのアドレス
戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上
解 説 : LED ディスプレイ内容の表示状態を取得します。

書式例(VB):

```
i = TMI_DisplayQ( hID, ans )  
ディスプレイ内容の表示状態を取得します。
```

8-23. プリセット番号の選択

int TMI_Preset(long hID , char preset)
関数名 : TMI_Preset
引 数 : long hID: ID 番号
char preset: プリセット番号指定(1-4)
戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上
解 説 : 指定したプリセット番号を選択します。

書式例(VB):

```
i = TMI_Preset( hID, 1 )  
プリセット番号1を選択します。
```

8-24. プリセット番号の取得

int TMI_PresetQ(long hID , char &ans)
関数名 : TMI_PresetQ
引 数 : long hID: ID 番号
char &ans: プリセット番号取得バッファのアドレス(1-4)
戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上
解 説 : 選択されているプリセット番号を取得します。

書式例(VB)

```
i = TMI_PresetQ( hID, ans )  
選択されているプリセット番号を取得します。
```

8-25. 出力モニタ値の取得

int TMI_MoniDataQ(long hID , char ch , double &voltage , double ¤t ,char &cv_cc)
関数名 : TMI_MoniDataQ
引 数 : long hID: ID 番号
char ch: 出力 CH 位置 A-D(1-4)
double &voltage: モニタ電圧値取得バッファのアドレス
double ¤t: モニタ電流値取得バッファのアドレス
char &cv_cc: CV/CC 状態取得バッファのアドレス(CV = 0,CC = 1)
戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、
出力範囲外=-3、成功=0 以上
解 説 : 指定した出力 CH のモニタ電圧値、電流値、CV/CC 状態を取得します。
注 意 : モニタ値は本体表示している CH は 0.7 秒、非表示の CH は 2 秒ごとのサンプリングとなっています。設定を変更してからモニタ値の変化が読み取れるまでの時間は、負荷の状態によって 5 秒程度かかる場合がありますので応答値の扱いには注意が必要です。

書式例(VB):

```
i = TMI_MoniDataQ( hID, 1, voltage, current, cv_cc )  
指定した出力 CHA の電圧・電流モニタ値と CV,CC 状態を返します。
```

8-26. システムアドレスの取得

int TMI_AdrQ(long hID , char &ans)

関数名 : TMI_AdrQ

引 数 : long hID: ID 番号

char &ans: システムアドレス取得バッファのアドレス

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上

解 説 : 指定した ID 番号のシステムアドレスを取得します。

書式例(VB):

i = TMI_AdrQ(hID, ans)

システムアドレスを取得します。

8-27. ローカル設定

int TMI_RemoteLocal(long hID)

関数名 : TMI_RemoteLocal

引 数 : long hID: ID 番号

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上

解 説 : リモート状態の電源をローカルに切り替えます。

書式例(VB)

i = TMI_RemoteLocal(hID)

8-28. ローカルロックアウトの設定

int TMI_LocalLockOut(long hID)

関数名 : TMI_LocalLockOut

引 数 : long hID: ID 番号

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上

解 説 : 電源をローカルロックアウト状態にし、本体の LOCAL キーを無効にします。

注 意 : ローカルロックアウト状態からの復帰には、ローカル設定を実行してください。

書式例(VB):

i = TMI_LocalLockOut(hID)

8-29. データバックアップ

int TMI_DataBackUp(long hID)

関数名 : TMI_DataBackUp

引 数 : long hID: ID 番号

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上

解 説 : EEPROM に保存可能な全てのデータについてバックアップを行います。

データバックアップには約 30 秒かかります。

注 意 : データバックアップ終了後、バックアップ完了メッセージが電源から発行されます。

書式例(VB):

i = TMI_DataBackUp(hID)

Do While result <> 0

result = TMI_In(hID, buffer)

Loop

データバックアップを実行し、完了メッセージ受信まで待機する。

8-30. サービスリクエストの設定

int TMI_SRQEnable(long hID , char onoff)

関数名 : TMI_SRQEnable

引 数 : long hID: ID 番号

char onoff: サービスリクエスト設定(許可 = 1、禁止 = 0)

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上

解 説 : サービスリクエスト機能の許可／禁止の設定をします。

書式例(VB):

i = TMI_SRQEnable(hID, 1)

サービスリクエスト機能を許可します。

8-31. 全てのプリセット値取得(1)

int TMI_AllPresetQ(long hID , double &data)

関数名 : TMI_AllPresetQ

引 数 : long hID: ID 番号

double &data: プリセットデータ取得バッファ配列の先頭アドレス

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上

解 説 : 全てのプリセット値を取得します。

注 意 : 取得したデータは以下の順序でバッファ配列に格納されます。

プリセット1: CHA: 電圧値

プリセット1: CHA: 電流値

↓

プリセット 1: CHD: 電圧値

プリセット 1: CHD: 電流値

↓

プリセット 4: CHA: 電圧値

プリセット 4: CHA: 電流値

↓

プリセット 4: CHD: 電圧値

プリセット 4: CHD: 電流値

必要となるバッファ配列は以下となります。

プリセット数 x CH 数 x (電圧 + 電流) = 4 x 4 x 2 = 32

(64bit 環境では正常に通信できない場合があります。

通信できない場合は「TMI_AllPresetS」命令を使用してください。)

書式例(VB):

Dim Data(32) As Double

i = TMI_AllPresetQ(hID, Data(0))

8-32. 全てのプリセット値取得(2)

int TMI_AllPresetQS(long hID , char &data)

関数名 : TMI_AllPresetQS

引 数 : long hID: ID 番号

char &data: プリセットデータ取得バッファ配列の先頭アドレス

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上

解 説 : 全てのプリセット値を取得します。

注 意 : 取得したデータは以下の順序で「,」区切りで格納されます。

コマンド名

PC アドレス

↓

プリセット 4: CHA: 電圧値

プリセット 4: CHA: 電流値

↓

プリセット 4: CHD: 電圧値

プリセット 4: CHD: 電流値

↓

プリセット 1: CHA: 電圧値

プリセット 1: CHA: 電流値

↓

プリセット 1: CHD: 電圧値

プリセット 1: CHD: 電流値

↓

プリセット 2: CHA: 電圧値

プリセット 2: CHA: 電流値

↓

プリセット 2: CHD: 電圧値

プリセット 2: CHD: 電流値

↓

プリセット 3: CHA: 電圧値

プリセット 3: CHA: 電流値

↓

プリセット 3: CHD: 電圧値

プリセット 3: CHD: 電流値

書式例(VB):

Dim Data As String

Data = String\$(256, 0)

i = TMI_AllPresetQS(hID, Data)

第 9 章 コマンド送受信関数

電源制御用のコマンドを直接、送受信するための関数です。

9-1. 送受信コマンドについて

◆ 送受信で使用するコマンドの文字列については、本体の取扱説明書または巻末を参照してください。

9-2. コマンド送信

```
int TMI_Out( long hID , char &str )
```

関数名 : TMI_Out

引 数 : long hID: ID 番号

char &str: 送信コマンド文字列

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上

解 説 : コマンドを送信する。

書式例(VB):

```
i = TMI_Out( hID, "SW1" )
```

メインアウトプットを ON に設定する。

9-3. 応答受信

```
int TMI_In( long hID , char &buffer )
```

関数名 : TMI_In

引 数 : long hID: ID 番号

char &buffer: 受信応答格納バッファのアドレス

戻り値 : タイムアウトエラー=-1、オープン、クローズエラー=-2、成功=0 以上

解 説 : 電源からの応答を受信します。

注 意 : 受信バッファは、想定される応答に対して十分な領域を確保してください。

書式例(VB):

```
i = TMI_In( hID, buffer )
```

第 10 章 利用上の注意について

- ・ 複数のローカルバスマスターに重複した PC アドレスを設定しないでください。また、同一ローカルバスに接続されているスレーブ機に、重複したシステムアドレスを設定しないでください。正常に動作しません。
- ・ ローカルバスは、バスマスター機がコマンドを解釈してからスレーブ機にコマンドを転送します、このため、実際のコマンド実行まで遅延があるため注意が必要です。また、短い時間で設定変更を繰り返し行くと、本体の動作速度が低下したり、コマンドを受付けない場合がありますので、コマンド送信間隔の設定には十分注意し、タイムアウトの設定を長めにおこなってください。
- ・ ローカルバスマスターは、電源 ON 時に接続されているスレーブを確認しますので、スレーブと同時か後に電源が立ち上がるようにしてください。また、スレーブの構成やアドレス設定が変わる場合は、必ずローカルバスマスターの電源を OFF にしてください。
- ・ ローカルバス接続を行なう場合は、PC 及び OS の高速化・最適化やローカル機器の台数により、連続してコマンドを送信すると正常にコマンドを受付けない場合があります。コマンド間にウエイトを 100ms 程度入れると安定することがあります。PC の CPU、チップセット、クロックや OS のドライバ類が変更になった場合は、十分ご注意ください。
- ・ PC の CPU クロックが高速な場合に連続して設定や読み取りを行うと、本体の動作速度が遅くなる場合があります。また、同じ設定のオン・オフを繰り返して行くと、処理が終わる前に次の設定が行われるため、設定が省略されて見えることがあります。特に内部のステータスやモニタデータの読み出しについては、数百 ms の読取周期がありますので注意が必要です。
- ・ 本 API は、サスペンドなどの PC の省電力モードに対応しておりません、省電力モード以後の本機の動作は不定となりますので設定には十分注意してください。
- ・ 本 API 内部で double 型の変数を利用している値については、2 進数演算による演算誤差を含む場合があります。この場合は、最小分解能の半分の値を加算して有効桁数以後を切り捨てると正しい値を得ることができます。詳しくは利用する開発環境での数値の取り扱いを確認してください。
- ・ 電源を複数のプログラム、プロセス、スレッドから同時にコントロールする場合は、通信が重ならないように十分検証をおこなった上でご使用ください。
- ・ アプリケーションのフォルダなどに、API の "Tmi_Api.dll"がある場合は、動作がおかしくなる場合があります。フォルダから DLL を削除して API を正しくインストールしてください。
- ・ WindowsUpdate によるインストールが実行中の場合は API または USB デバイスドライバのインストールができないことがあります、WindowsUpdate の処理がすべて終了してから再度インストールをおこなってください。
- ・ 通信を行う場合に、PC および機器のグラウンド電位が異なったり接地されていない場合は、動作が不安定になります。必ず 3 芯の電源ケーブルを利用し、筐体が接地した状態でご利用ください。USB ハブを利用する場合も、できるだけ筐体グラウンドが同じ電位になるように設置してください。
- ・ モーターやインバーター、溶接機などノイズが発生している環境では、通信が不安定になります。PC、被制御機器、通信経路、商用電源がノイズの影響を受けないように設置し、十分に動作確認をした上でご利用ください。

第 11 章 電源制御用通信コマンド

API の関数を利用せずに直接制御を行う場合のコマンド一覧は以下のとおりです。

モード設定

項 目	コマンド	パラメータ
MAIN OUTPUT	SW	0/1
OUTPUT SEL CH-A	OA	0/1
OUTPUT SEL CH-B	OB	0/1
OUTPUT SEL CH-C	OC	0/1
OUTPUT SEL CH-D	OD	0/1
表示 CH 選択	DS	1/2/3/4
PRESET 呼出し	PR	1/2/3/4
ローカル指定	LC	1
ローカルロックアウト	LL	1
サービスリクエスト	SR	0/1
設定データ保存	MW	1
トラッキング ON/OFF	TO	0/1
トラッキングモード	TM	0/1
DELAY ON/OFF	DY	0/1

設定要求

項 目	コマンド	パラメータ
出力モニタ要求(整数)	ST	0
プリセット情報要求(整数)		1
ディレイ・トラッキング情報		2
機器情報要求		3
出力モニタ要求(実数)		4
プリセット情報要求(実数)		5
ID 要求	PWID	なし
被制御電源問い合わせ	PW?	なし
機種 ID	* IDN?	なし
状態応答設定	SRMODE	0(通常)/1(禁止)

電圧電流設定

プリセット	CH	電圧設定コマンド	電流設定コマンド	パラメータ
4	A	VA	AA	%%% 4 桁整数指定 または小数指定
	B	VB	AB	
	C	VC	AC	
	D	VD	AD	
1	A	VE	AE	
	B	VF	AF	
	C	VG	AG	
	D	VH	AH	
2	A	VJ	AJ	
	B	VK	AK	
	C	VL	AL	
	D	VM	AM	
3	A	VN	AN	
	B	VP	AP	
	C	VQ	AQ	
	D	VR	AR	

トラッキング設定

CH	電圧設定コマンド	電流設定コマンド	パラメータ
A	EA	IA	%%%% 4 桁整数指定
B	EB	IB	
C	EC	IC	
D	ED	ID	

トラッキングモード設定

CH	コマンド	パラメータ
A	GA	0/1/2
B	GB	
C	GC	
D	GD	

ディレイ時間設定

CH	コマンド	パラメータ
A	DA	%%%% 4 桁整数指定
B	DB	
C	DC	
D	DD	

各コマンドの利用方法は PW-A または PAR-A の本体取扱説明書を参照してください。



株式会社 テクシオ・テクノロジー

〒222-0033 横浜市港北区新横浜 2-18-13 藤和不動産新横浜ビル 7F
<http://www.texio.co.jp/>

アフターサービスに関しては下記サービスセンターへ

サービスセンター 〒222-0033 横浜市港北区新横浜 2-18-13 藤和不動産新横浜ビル 7F
TEL.045-620-2786