＜実践報告・資料＞

GP-IB による計測・制御技術

関東職業能力開発大学校　柴田 清孝*
関東職業能力開発大学校　陣内　望

Instrumentation and Control Technique for the GP-IB Interface

Kiyotaka SHIBATA, Nozomu JINNOUCHI

要約　計測・制御用の汎用 GP-IB(General Purpose Interface Bus) インタフェースを使用して、計測器の制御、データ収集、応用プログラミングを行った。プログラミング言語には、汎用性の高い Visual Basic.NET を用いるが、Excel に内蔵されている VBA(Visual Basic for Application) にも適応できる。また、Excel のセル内に計測データを取り込み、データ処理や評価表作成を行うことも可能である。ここでは､デジタルマルチメータ (DMM) を模擬したフォームを作成し、グラフ表示などの機能を付加した。

Ⅰ　はじめに

計測・制御の分野では、計測器の制御やデータ転送のためのデジタルインタフェースとして、規格化された GP-IB インタフェースが用いられてきた。また、シリアルインタフェース (RS-232C) も手軽で低価格であるため装備されてきた。しかし、ケーブルの太さ、重さや高価格、あるいは、USB や LAN インタフェースの普及とともに過去のレガシー (Legacy) インタフェースとなっている。一方、コンピュータの環境は、オペレーティングシステム (OS) のアップデートとともに進歩し、レガシーインタフェースは装備されなくなり、GP-IB インタフェースにいたっては OS の標準外デバイスとなっている。他方で、手軽にパーソナルコンピュータ (PC) と測定器を接続し、データ収集を行いたいという要望は多く、また、古い測定器を有効活用したいという声もある。

本稿では、レガシーインタフェースである GP-IB インタフェースを用いて、計測器の制御、データ収集、応用プログラミングを行うものであり、プログラミング言語には、汎用性の高い Visual Basic.NET を用いるが、Excel に内蔵されている VBA でも使用可能である。特に、VBA 環境は、Visual Studio などの開発ツールを必要とせずに VB6 の環境で開発ができ、また、Excel のセル内に計測データを取り込み、データ処理や評価表作成を行うことも可能である。GP-IB インタフェースには、USB ポートに接続する USB-GPIB 変換インタフェース (National Instruments 社製 GPIB-USB-HS) を使用する。OS は、Windows7、WindowsXP SP3 の 2 種の環境で動作確認を行い、プログラム開発は Microsoft Visual Studio 2010 の Microsoft Visual Basic 2010 を .Net Flamework4.0 の環境 (以下 VB.NET とする) を使用して行った。なお、GP-IB インタフェース用のドライバは、National Instruments 社製の NI-488.2、NI-VISA､NI-488.2 .NET Flamework4.0 をインストールした。

Ⅱ　GP-IB インタフェース [(1)(2)(3)]

計測器同士、計測器とコンピュータの間で相互にデータ通信を行う場合、従来は、コネクタの種類・形状、ピン配置、信号の種類、論理、信号レベル、信号フォーマットなどが各社で種々多様であり、他社メーカー製品との接続はもちろん、同一メーカー製品間でも容易に接続できない場合が多く、計測部のシステム化の大きな障害や出費の要因であった。このような状況を解決するため、計測・制御用のインタフェースとして GP-IB インタフェースが IEC(International Electrotechnical Commission) や IEEE(The Institute of Electrical and Electronics Engineers, Inc.) で規格化、標準化された。GP-IB インタフェースでは、計測・制御を対象として、機器相互の入出力信号や制御法はもとより、コネクタ、ケーブル、機能までも国際的に統一されている。このインタフェースの名称は、GP-IB の他、IEEE-488、IEEE-IB、HP-IB、IEC バスなどと呼

論文受付日　H21.4.1　*現　東北職業能力開発大学校

ばれている。1975 年に IEEE488.1-1975 が認可され、その後改訂され 1987 年に IEEE488.1-1987 となった。一方で、1990 年には計測器のコマンドが統一された Standard Commands for Programmable Instruments(以下 SCPI という) 言語が制定され、1999 年に改訂された IEEE488.2-1999 規格に含まれている。また、PC の画面上で計測器を模擬表示したり、計測を行うような仮想計測器 VI(Virtual Instrument) 環境では、VISA(Virtual Instrument Software Architecture) コマンドが使用され、グラフィカルな計測・制御系の構築が可能である。

GP-IB インタフェースでは、データは 8bit パラレルに伝送される。このためインタフェースバスは 8 本のデータ線、3 本のハンドシェイク線、そして、5 本の制御線を有している。また、データの送信、受信やバスの制御を行うためにインタフェースに、トーカー、リスナー、コントローラと呼ばれる機能を持たせる。トーカー (話し手に相当) は、ディジタルメーター等のデータをバスに送出する機器、リスナー (聞き手に相当) は、プリンタ、プロッタ等のバスからデータを受取る機器、コントローラ (司会者に相当) は、コンピュータ等のトーカーやリスナーの指定やバスの制御コマンドを送出する機器である。

GP-IB バス上には、上述のトーカー、リスナー、コントローラの機能を有する機器を接続するが、同時に動作するトーカーは 1 台、コントローラも 1 台となるようにしなければならない。また、バス上の機器には GP-IB アドレス (0 ～ 30) を付与し、これにより最大 31 台まで区別される。

1　GP-IB バスのコマンド、データの転送

1－1　コマンドモードとデータモード

GP-IB インタフェースのコマンドは、管理バスの 5 本の制御線 (ATN、REN、SRQ、IFC、EOI) を使用したユニラインメッセージとデータバスも使用したマルチラインメッセージに大別される。そのため、GP-IB インタフェースのバス上のデータは、GP-IB インタフェースのバスを操作するコマンドとトーカーとリスナー間で送受されるデータに大別される。それらの区別は、ATN 制御線で行われ、ATN 制御線が“L”の時をコマンドモードと呼び、データバス上のデータは GP-IB インタフェースのコマンドである。一方、ATN 制御線が“H”の時はデータモードと呼ばれ、トーカーとリスナー間でのデータ伝送状態である。

GP-IB インタフェースコマンドは、アドレスドコマンド、ユニバーサルコマンド、リスナーのアドレスを指定するリスンアドレスコマンド、トーカーのアドレスを指定するトークアドレスコマンドの他、二次コマンドと呼ばれるコマンドに大別される。アドレスドコマンドは、指定アドレスの機器に対してのみ有効なコマンドであり、ユニバーサルコマンドは、すべての機器に対して有効なコマンドである。二次コマンドは、アドレスドコマンド、ユニバーサルコマンドの一次コマンドに付随するコマンドで、二次アドレスの指定も可能である。

GP-IB バスでは、確実なデータ転送を行うため、あるいは、コントローラを介さずにトーカーとリスナー間でデータ転送するために 3 本のハンドシェイク線を使用した 3 線ハンドシェイクと呼ばれる方式でデータ伝送を行っている。

1－2　ポーリング

GP-IB インタフェースにはトーカー、リスナー側からコントローラを呼出す方法として、SRQ 制御線を用いるポーリング動作 (サービスリクエスト) が用意されており、これにより、コントローラは機器の状態をシリアルポールまたはパラレルポールと呼ばれる方法で調べることができる。SRQ 制御線は、コントローラにとっては割り込み信号となる。よって、コントローラであるコンピュータ側にサービスリクエストの割り込み処理を行うプログラムを記述することにより、ポーリングの動作が可能となる。

シリアルポールでは、サービスリクエストを受けたコントローラが 1 台ずつ、順番 (シリアル) にポーリングを行い機器のステータスバイトを読み出してサービスリクエストを出している機器を判別する。パラレルポールでは、データ線 DIO1-DIO8 にあらかじめ機器を割り当てておき、コントローラが ATN 制御線と EOI 制御線を同時に“L”にすることでパラレルポールを行う。どのデータ線が“L”かを判断して、サービスリクエストを出している機器を判別する。

1－3　データの区切り、終了

GP-IB インタフェースでは、データや機器のファンクションコードの区切りや終了を表すために、デリミッタ (Delimiter) とかターミネータ (Terminator) と呼ばれるコードをデータやコマンドに付加して伝送する。一方、GP-IB インタフェースの管理バスに EOI(End Or Identify) というデータの最後のバイト転送と同時に“L”になる制御線が用意されており、ハードウエアデリミッタとも呼ばれる。機器やコントローラ間でこれらの扱いが異なる場合、バス動作が停止するトラブルが発生することがある。

デリミッタには、ハードウエアデリミッタの他、CR 文字 (16 進アスキーコード表示で 0x0D)、LF 文字 (16 進アスキーコード表示で 0x0A)、CR+LF 文字 (16 進アスキーコード表示で 0x0D、0x0A)、バイト数 (バイナリー転送時)、ハードウエアデリミッタと CR 文字、ハードウエアデリミッタと CR+LF 文字の組み合わせなどのコードや方法が存在する。これらのデリミッタが混在する場合も考慮して、GP-IB バスを確実に動作させるには、ハードウエアデリミッタと CR 文字あるいは CR+LF 文字のソフトデリミッタの併用が推奨されている。すなわち、ソフトデリミッタがデータに付与されていなくてもハードウエアデリミッタによりデータ転送を終了させることができ、ハードウエアデリミッタが検出されなくてもソフトデリミッタでデータの終わりを判断できることになる。

2　GP-IB インタフェースのドライバ

GP-IB インタフェースは Windows の標準外のデバイスであり、PC から GP-IB インタフェースを操作するためには使用する GP-IB インタフェースのドライバの組込みが必要となる。

National Instruments 社の GP-IB カード、USB-GPIB 変換インタフェースでは、NI-488.2 ドライバを組み込むと、NI-VISA ドライバも同時にインストールされる。これらドライバの本体は、Windows の System フォルダ内にある“GPIB-32.dll”と“VISA32.dll”というダイナミックリンクライブラリファイルである。GP-IB インタフェースの制御をプログラムから行うには、このダイナミックリンクライブラリ内の関数を API 呼出しで実行すればよいことになる。一方、ドットネット (.NET) の環境から GP-IB インタフェースを使用するには、それぞれの .Net Flamework のバージョンに応じたフレームワークのインストールが必要となる。

Agilent (現 Keysight Technologies) 社の GP-IB カード、USB-GPIB 変換インタフェースでは、IO ライブラリをインストールすると、GP-IB ドライバと VISA ドライバがインストールされる。

3　計測器のコマンド　SCPI 言語 (4)(5)

計測器を制御するためのコマンドは、各社、各機器により異なっており、詳細は機器個別のマニュアル等で確認する必要がある。一方、計測制御用のコマンドを統一する方向として、旧 HP(Hewlett-Packard) 社が提唱し IEEE で標準化されたプログラマブル計測器用標準コマンド SCPI (スコピーあるいはスキッピー) 言語がある。これらのコマンドには、アスタリスク‘*’から始まる共通コマンドと、機器特有のサブシステムコマンドの 2 種類がある。共通コマンドは、*RST、*IDN? のようにリセット、ステータスなどの機器全体の制御に関係するコマンドである。一方、サブシステムコマンドは、機器特有の機能を設定したり、問い合せたりするために使用する。このコマンドは、ファイルシステムのようにツリー化されており、サブグループとの区分にはコロン‘:’が使われる。たとえば、OUTPUT:PON:STATE のように記述される。また、機器ステータスの読み込みや、測定データの読み取りのように機器からの応答を必要とする場合には、コマンドの後ろにクエリインジケータであるクエスチョンマーク‘?’を付加する。たとえば、

```
MEASure:VOLTage:DC?
STATus:QUEStionable:ENABle?
```

のように記述する。ここで、小文字の文字は省略可能である。

計測器等に送出するコマンド文字列は、最後に LF 文字もしくは CR ＋ L F 文字を付加し、ターミネータとする。また、計測器からの応答には、種々のフォーマットが存在しており、数値文字のみ、ヘッダー付き、バイナリなどがある。データ量が多くなければ、通常、文字 (アスキーコード) として伝送することが多い。

この SCPI 言語によるコマンドは、シリアル、パラレル、GP-IB、USB インタフェースなどの種類によらず共通のコマンド体系であり、種々の計測器を制御するために使用可能である。

Ⅲ　GP-IB プログラム作成

VB.NET から GP-IB インタフェースを使用するには、.NET Flamework4.0 とそれに対応する GP-IB ドライバの Flamework(NI-488.2 .NET Flamework4.0) がインストールされている必要がある。

はじめに、Imports 命令で National Instruments.NI4882 の NameSpace 定義を追加し、GP-IB 関係の API 関数 (サブルーチン) を使用できるようにする。コード記述画面の先頭に次例のように記述する。

```
Imports National Instruments.NI4882
```

ただし、この NameSpace を追加するには“System.web”アセンブリへの参照設定が必要である。このアセンブリは、デフォルトの VB.NET フレームワークが“.NET Framework 4 Client Profile”となっているため表示されない。そこで、対象フレームワークを“.NET Framework 4”に変更し、参照設定に“System.web”

を追加する。

次に、VB.NET では、GP-IB インタフェースはボードオブジェクトとして、GP-IB 機器はデバイスオブジェクトとして作成され、これらのオブジェクトを使用してプログラミング等を行う。通常のプログラミングではデバイスオブジェクトを使用し、GP-IB バスの直接制御やトーカー、リスナーのステータス確認などはボードオブジェクトを使用する。GP-IB デバイスを作成するのが、New Device 関数である。この関数の引数は、ボード ID(通常 0、複数の GP-IB インタフェースが存在するなら区別する)、接続機器の GP-IB アドレス、拡張アドレス (通常 0 でよい) である。以下は、GpibDevice という変数名で GP-IB デバイスを作成するコード例である。

```
Private GpibDevice As Device
GpibDevice = New Device(0, 27, 0)
```

なお、Device 関数には引数の異なるバリエーション (拡張アドレスが無い、タイムアウト値ありなど) を有している。

Device 関数で作成された GP-IB デバイスへのコマンドの送信 (書込み) は、GpibDevice.Write 関数を使用する。引数は、送信するコマンド+デリミッタである。デリミッタコードは、使用機器の設定を確認し付加する。以下は、SCPI コマンドの ID を読出すクエリコマンド *idn? にデリミッタコード LF 文字を付加して GP-IB デバイスに送信する例である。

```
GpibDevice.Write("*idn?" + ControlChars.Lf)
```

GP-IB デバイスからのレスポンスやデータなどの文字データは、GpibDevice.ReadString 関数を使用して受信 (読込み) する。以下は、受信文字列をテキストボックス (オブジェクト名 txtRead) のテキストプロパティに代入して表示する例である。

```
txtRead.Text = GpibDevice.ReadString()
```

テキストボックスの場合、末尾のデリミッタコードなどは表示されない。これらを表示するためにはテキスト処理が必要となる。たとえば、エスケープコードを模擬して、LF 文字を \ ¥n、CR 文字を \ ¥r で表示、変換する Function コードを用意する。以下は変換するコード例である。

```
Private Function ReplaceCommonEscapeSequences
(ByVal s As String) As String
Return s.Replace("¥n", ControlChars.Lf).Replace
("¥r", ControlChars.Cr)
  End Function
```

```
Private Function InsertCommonEscapeSequences
(ByVal s As String) As String
  Return s.Replace(ControlChars.Lf, "¥n").Replace
(ControlChars.Cr, "¥r")
  End Function
```

GP-IB デバイスの使用を終わるには、GpibDevice.Dispose 関数を実行し、デバイスと使用リソースを開放する。以下は、開放する例である。

```
GpibDevice.Dispose()
```

コンピュータから計測器を制御する場合、従来はインタフェースの初期化 (IFC コマンド) と接続機器のリセット、リモートモード化 (REN コマンド) が必要とされていたが、VB.NET では特に意識する必要はない。しかし、計測器の今の状態が不明であるため、機器設定を行う前に、SCPI コマンド *RST を送出して、インタフェースに接続されている計測器等を初期化、リセットして初期状態としてから設定を行うとよい。

その他のプロシジャーやコマンドが用意されているが、詳細はヘルプ (スタートメニュー内の National Instruments/NI-488.2/NI-488.2 .NET Flamework 4.0 Help の“Using .NET Class Library” や“Using the Measurement Studio NI-488.2 .NET Library”など) やオブジェクトブラウザーで確認するとよい。

1　テストプログラム例

GP-IB インタフェースの動作を確認するためには、先に述べた GP-IB デバイスの作成、データ送信、受信、リソースの開放のコードを記述し、正常に動作、機能しているかを調べる必要がある。そのため、ここでは、ディジタルマルチメータ DMM(Agilent 社製 HP34401A) を GP-IB インタフェースに接続し、SCPI コマンド等を送信し、その返答を表示するテストフォームを作成する。図 1 にテスト用フォームの例を示す。

Open ボタンは、GP-IB デバイスを作成するコマンド

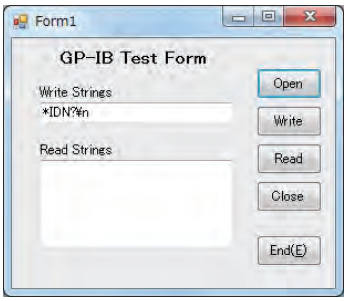

図 1　テスト用フォームの例

ボタンで、PC 側の GP-IB アドレスは 0、DMM は 27 としている。Write ボタンは、Write Strings テキストボックス内の文字列を、\ ¥ n を LF 文字に、\ ¥ r を CR 文字に変換してから GP-IB インタフェースに送信するコマンドボタンである。Read ボタンは、クエリコマンドの実行後にその応答を受信し、Read Strings テキストボックスに表示するボタンである。この場合もターミネータコードなどの LF 文字を \ ¥ n に、CR 文字を \ ¥ r に変換してから表示する。Close ボタンは、GP-IB デバイスと使用リソースの開放を行うボタンである。これらのボタンを順に実行し、動作の確認を行う。

図 2(a) は＊ IDN コマンド、図 2(b) は測定コマンド、図 2(c) は抵抗測定への切り替えの実行例である。(a) は SCPI コマンドの機器情報を問合せるクエリコマンド *IDN? を Write ボタンで送信し、Read ボタンで受信文字を表示したものである。社名、機器名、バージョン番号などが表示されている。(b) は、DMM の DC 電圧測定コマンド MEAS:DC? を実行し、測定結果を受信したものである。(c) は、DMM の測定機能を抵抗測定に切り替えるコマンド FUNC “RES” を送信したものである。クエリコマンドではないため返答はないが、このコマンドの送信で DMM の機能が抵抗測定に切り替わる。

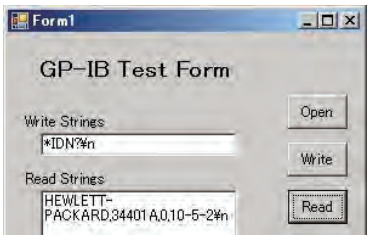

図 2(a)　*IDN コマンドの実行例

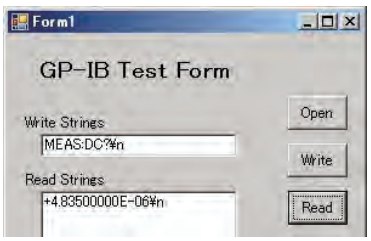

図 2(b)　測定コマンドの実行例

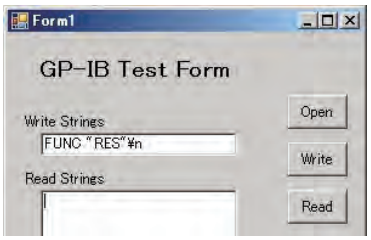

図 2(c)　抵抗測定への切替の実行例

2　ディジタルマルチメータフォームの作成

GP-IB 機器を用いた応用プログラミングの一例として、DMM フォームの作成とグラフ表示を取り上げる。はじめに、DMM を模擬したフォームを作成する。図 3 に DMM フォームの例を示す。このフォームは、タイマーコントロール Timer1 により一定時間間隔で測定を行い、測定結果を表示するものであり、測定ファンクション切り替えのボタン、表示、単位表示も付加している。

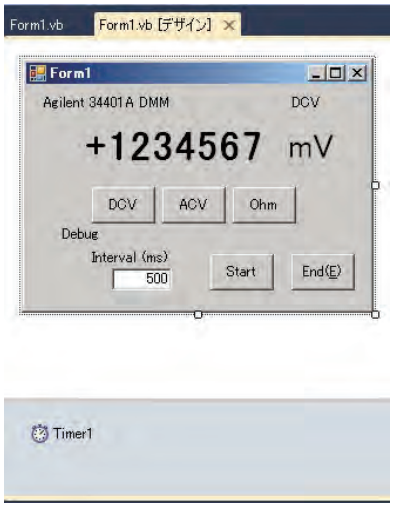

図 3　DMM フォームの例 (編集中)

Start ボタンは、Interval テキストボックスの時間に Timer1 コントロールのインターバル値を設定してスタートし、この時間間隔で DMM に測定コマンドを送信し、測定値を受信してラベルに表示するものである。このボタンはトグル動作で Stop ボタンも兼用している。

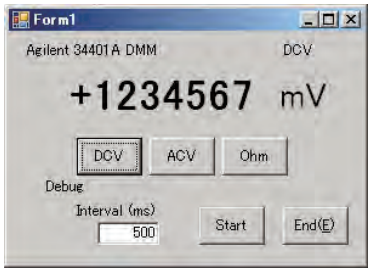

図 4　直流電圧測定の実行例

図 4 に直流電圧測定の実行例を示す。タイマーコントロールの時間精度は 20ms 程度であり、また、測定器の AD 変換速度およびインタフェースの速度、ドライバの処理速度なども影響し測定間隔はあまり早くできず、リアルタイム性には乏しい処理結果であった。

次に、グラフ表示を追加する。先の例は、一定時間間隔で測定を行い、その結果を数字として表示するフォームであった。このフォームにリアルタイムでのグラフ表示とファイルへの入出力機能を追加したプロ

グラムを作成する。

VB.NET でのグラフィクス描画は、グラフィクスオブジェクトで行われ、フォームおよびコントロールへの描画が可能である。ここでは、フォーム上に配置したピクチャボックス内に測定データをプロットする。先のフォームを右に拡張し、そこにピクチャボックス等を配置する。図 5 にグラフ表示を追加したフォームを示す。

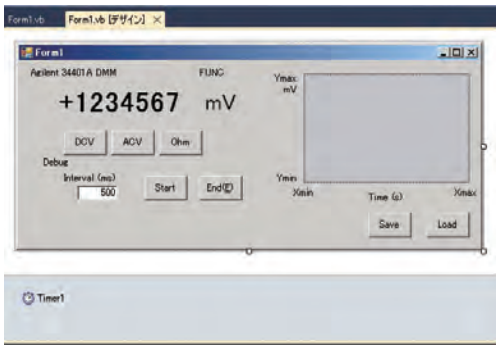

図 5　グラフ表示を追加したフォーム (編集中)

ピクチャボックスには、オシロスコープ画面を模擬したビットマップファイルを読み込みスケールとする。横軸は時間、縦軸は測定ファンクションである。測定開始時の縦軸、横軸のスケールは 0 ～ 100 としているが、オーバースケールした場合は折り返すように処理を行っている。

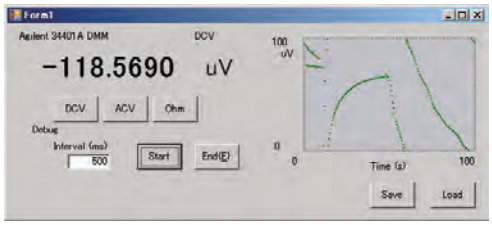

図 6(a)　直流電圧測定 (測定中)

図 6(b)　直流電圧測定 (ファイル入力時)

図 6(a) に直流電圧測定 (測定中) を示す。縦軸、横軸ともにオーバースケールするとグラフが折り返して表示されている。図 6(b) に直流電圧測定 (ファイル入力時) を示す。測定終了後にファイルにデータを保存した後、読み込んだ場合の表示である。縦軸、横軸の最小値、最大値が表示される。

Ⅳ　おわりに

計測機器の有効活用と計測制御プログラミングの技法を習得するために、GP-IB インタフェースを使用したプログラム開発を行った。開発言語には、汎用性の高い VB.NET を使用し、ディジタルマルチメータによる DMM フォーム、リアルタイムでのグラフ表示等を行った。さらに、直流電源装置を電源として使用すれば、回路の過渡現象測定、素子の VI 特性測定なども可能となろう。この他、種々の測定が自動化できると考えられる。また、Excel へのデータ取り込みや VBA による機器制御なども可能であり、手軽に計測制御プログラミングを行うことが可能である。

GP-IB インタフェースは、レガシーな汎用の計測制御インタフェースであるが、計測器からの割り込みを処理するポーリングの機能を有しており、これらをサポートするプログラムを作成することでより柔軟な応用性のある処理が可能である。一方、計測器には種々のインタフェースが装備されているものがあり、このインタフェースで何ができるかというよりは、何を接続し、何を測定したいか、どの様な制御を行いたいかというニーズを明確にし、それを種々のインタフェースとプログラムにより実現していくという、ハード面よりもソフト面の充実が必要であろう。

※商標等表記

[参考文献]

(1) 岡村廸男、IEEE-488(GP-IB) とその応用、CQ 出版、1988 年、p.15
(2) 松崎幹夫、GPIB の使い方、トランジスタ技術、12 月号、CQ 出版、1983 年、p.295
(3) 磯部俊夫、C 言語と RS-232C/GP-IB、工学図書、1988 年、p.118
(4) Hewlett-Packard、SCPI ビギナーズガイド、p.2-1
(5) Agilent Technology、HP34401A マルチメータ ユーザーズ・ガイド、p.150