エクセルからの GP-IB 制御

2012/2/29　ver.3

初版　2005 年 11 月 23 日

第２版　2008 年 6 月 4 日

大野泰夫

1.はじめに

計測器をプログラム制御して測定を行うことは単に作業の負担を減らすだけでなく、精度の向上や全く新しい測定を可能とする。出来合の装置とソフトで行う測定では、出来合の成果しか得られない。装置メーカー、ソフトメーカーが販売している装置やツールは誰かが初めて測定を行った残骸である。

測定をプログラム制御して行う技術は近年整備され、当初はＨＰ社（現アジレント社）のみが提供していたＨＰ－ＩＢインターフェイスがＩＥＥＥの標準規格となり、多くの測定器会社がそのインターフェースを用いた装置を作成している。現在ではアジレント社もＨＰ－ＩＢという呼称を捨ててＧＰ－ＩＢという名前を使っている。（ＧＰは General Purpose の略）最近ではＵＳＢやＬＡＮ規格の導入も始まっているが、しばらくはＧＰ－ＩＢが自動測定の標準として使われると思われる。

現実の測定やデータ整理ではエクセルでのデータ整理、解析が広く行われている。エクセルには表計算のみならずグラフの作成、ＷＯＲＤ・パワーポイントとのリンクなど実験結果を論文やスライドまで持っていくのに便利な環境がある。さらにＶＢＡ（Visual Basic for Applications）というプログラム言語を搭載しており、これを使えば複雑なデータ処理も可能となっている。

そこで、今回は測定系の制御をこのＶＢＡを用いて行うこととした。これにより、測定条件の設定をエクセルの表で指定できるようになり、また測定結果が表に直接と入り込めるようになる。

2.測定系の構成

測定系の構成は、ハードウエアとしては

（１）測定に直接用いる装置、今回はモノクロメータ付き光源と半導体パラメータアナライザ（４１５５Ｃ）を用いる、

（２）パソコン（Windows、エクセルが走るもの）、

（３）ＧＰ－ＩＢインターフェイス

である。

ＧＰ－ＩＢインターフェイスには Agilent82357A USB/GPIB Interface for Windows を用いる。

ソフトウエアはエクセルとエクセルから起動されるエクセルＶＢＡである。

GP-IB 制御に用いるソフトウエア体系には VISA(Virtual Instrument Software Architecture)と SICL((Standard Instrument Control Library)の２種がある。SICL は、以前からこれで書かれた資産を持っている場合を除いて推奨されていない。特に VXI*Plug&Play* 機器を使う場合はＶ

ＩＳＡが必須と言うことで、ここでは VISA を用いる。

図１　Agilent82357A USB/GPIB Interface for Windows

またアジレントの測定器､特に 4155C の制御では SCPI(Standard Commands for Programable Instruments)、FLEX(FAst Language for EXecution)、4145 シンタックスの３つのコマンドモードがある。4145 シンタックスは 4155C の前身の 4145Ｂで用いていたコマンドで、4145 用に開発されたソフトをそのまま使う場合に使うが 4155C の機能をフルには利用できない。SCPI と FLEX の関係は明確でないが、FLEX の方が高速ということからこちらを用いる。FLEX モードの中にさらに US と US42 のモードがある｡これも旧機種 4142B との互換性を保つためのもので、過去の資産のない大野研としては US モードを用いる。ここまでのモードに関しては、研究室としては統一した方が便利であろう。ＧＰ－ＩＢでのデータ転送にはＡＳＣＩＩとＢＩＮＡＲＹがあるがこれは状況に応じて使い分けることにする。

３．準備作業

(1)　エクセルでの VBA 練習

エクセルをオープンし、表示・ツールバーで Visual Basic にチェックを入れる。ツール・マクロ・新しいマクロの記録、などで VBA に慣れておくことが好ましい。

(2) インターフェイス機器の取り付け

(a)　パソコンへのソフトウエアのインストール。

82357A 付属の CD から PC に Agilent IO Library をインストールする。実験室にあるパソコンには既にインストールされている可能性が高い。Windows Desktop 右下の稼働中ソフトのアイコンに青い字の"IO"というアイコンがあれば既にインストールされている。(Man82357A p.11)

(b) 82357A の接続

PC の USB 端子に 82357A を接続する。USB ポートはセルフパワーである必要がある。始めは３つの LED すべてが点灯するが約 10 秒で FAIL（赤）と ACCESS（緑）は消灯し、READY（緑）のみが点灯する。(Man82357A p.21)

PC では新しい機器の接続が検知されるので指示に従いドライバーをインストールする。さらに GP-IB インターフェイスの検知が報告され､その際に VISA と SICL としての名前が表示される。必要なら名前を編集する。インターフェイスの名前はメモしておくこと。

（3）測定機器の接続

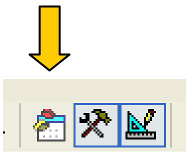
GP-IB ケーブルを接続する。機器側をつなぎ、次に 82357A 側につなぐ。
デスクトップ右下タスクバーの IO アイコンから VISA Assistant を起動する。
接続された装置が表示されているはずである。装置を選び、下記を実行する。

5 Select the **Formatted I/O** Tab.

6 Select the **IEEE 488.2** button.

7 Click the ***IDN?** button.

この操作を各装置に実行する。これが済むと、ダイアログボックスからコマンドを入力して装置を動かすことが可能となる。



（4） VISA モジュールのインストール

エクセルの VBA 編集ページを開く。
ファイル・ファイルのインポートでC:¥Program Files¥VISA¥winnt¥includeにあるvisa32.bas を指定する。これでVISAの関数が使えるようになる。
マニュアルvisa.pdf p.23～による注意事項。

- viPrintf,viScanf, viQueryfは使えない。代わりにviVPrintf, viVScanf, viVQueryf を用いる。
- VISA関数に対しValiant型変数を用いてはならない。これを防ぐため、プログラム冒頭で Option Explicitを宣言し、全変数を宣言して用いると良い。
- viVPrintf, viVscanf and viVqueryf を使う場合%記号で始まるformat指定は１個のみ。複数の場合はコマンドを分けること。
- viVScanf,viVQueryf に文字変数を使う場合は固定長の文字変数を使うこと。(Dim strVal as String * 40)

（5） プログラム例

Option Explicit (変数宣言しないと変数が使えないという面倒な点もあるが、訳の分からない変数が増えたり、変数のスペルミスによるバグを防げる。C言語では入れるのが標準で、原則入れること。)

’’

’ idn.bas・・・・以下５行はコメント文

```
’ This example program queries a GPIB device for an identification
’ string and prints the results. Note that you may have to change the
’ VISA Interface Name and address for your device from "GPIB0" and "22",
’ respectively.
’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’
Sub Main()
Dim defrm As Long ’Session to Default Resource Manager
Dim vi As Long ’Session to instrument
Dim strRes As String * 200 ’Fixed length string to hold results
’ Open the default resource manager session
Call viOpenDefaultRM(defrm)
’ Open the session to the resource
’ The "GPIB0" parameter is the VISA Interface name to a GPIB
’ instrument as defined in
’ Start | Programs | Agilent IO Libraries | IO Config
’ Change this name to what you have defined your VISA Interface.
’ "GPIB0::22::INSTR" is the address string for the device.
’ this address will be the same as seen in:
’ Start | Programs | Agilent IO Libraries | VISA Assistant
’ after the VISA Interface Name is defined in IO Config)
Call viOpen(defrm, "GPIB0::22::INSTR", 0, 0, vi)
’ Initialize device
Call viVPrintf(vi, "*RST" + Chr$(10), 0)
’ Ask for the device’s *IDN string.
Call viVPrintf(vi, "*IDN?" + Chr$(10), 0)
’ Read the results as a string.
Call viVScanf(vi, "%t", strRes)
’ Display the results
MsgBox "Result is: " + strRes, vbOKOnly, "*IDN? Result"
’ Close the vi session and the resource manager session
Call viClose(vi)
Call viClose(defrm)
End Sub
```

4155Cの制御

本体

メインスイッチ　オン

ボタンキーSYSTEM→画面キーmiscellaneous→NOT SYSTEM CONTROLLER,GPIBアドレス

*RST　：装置初期化
CN　1,2,3,4：SMU　1,2,3,4をオン
MM　1,1,2,3,4:　SMU1,2,3,4をスポットモード（この順に出力）
FMT 1:アスキー/ヘッダー付き
SLI k：積分時間　k=1 short, 2=Medium, 3=Long
AV k,0：k　平均化サンプル数が１０　０はダミー
DV n(SMU), 0(auto), Voltage, I_Comp, 0(Comp Abs)
DI
XE：測定の実行
RMD?：測定データのバッファへの転送
*STB：シリアルポール
　Bit1　1→0（測定完了）、Bit3　0→1（Query 回答あり）、Bit4 0→1（Messageあり）
BC：バッファークリア

計算途中の経過表示

（動いているかどうか、どの辺を走っているか判らなくなってしまうとき）

見たいところに、Application.StatusBar =を置く。

Application.StatusBar = Format(col) + "/" + Format(i)

＝の後ろは string です。

セルや図の書き換えを一時停止。

セルの書き換えや津の書き換えには時間を取られるので

計算が終るまで書き換えない用にする。

書き換え停止：Application.ScreenUpdating = False

解除：Application.ScreenUpdating = True

最後は必ず、解除すること。

セルデータの一括書き換え

旧エクセルでは VBA プログラムからのセルへの書き換えに時間がかかる。これを逃げるため、貼り付けるセルと同じサイズのアレイを作り、下記のようにして書き換える。

(例)

Dim data(10,20) as variant

・・・

data(i,j)でデータをセットする。

・・・

Range(cell(x,y),cell(x+10,y+20))=data

で書き込む。