

One Point Lesson

第六回「GP-IB ボードの制御をしよう」

今回は、「GP-IB ボードを制御しよう」です。この GP-IB ボードの制御は比較的多くのユーザーさんからご質問をいただいています。そこで、今回の OnePointLesson では GP-IB ボードの制御について、マニュアルだけでは分かりづらい部分をできるだけ分かりやすく説明しようと思っております。ちょっと高度なプログラム知識が必要ですが、F-BASIC では比較的簡単に GP-IB を制御することができます。

また、GP-IB の制御には他言語で書かれた DLL の呼び出しを使いますので、他言語呼び出しが分からない方も 参考に見てみてください。

では、さっそく今回の OnePointLesson を始めましょう。

今回は、統合モデルという形の GP-IB ボードを制御するプログラムを作ります。

まず「GP-IB とは何か」というところから始めることにしましょう。GP-IB とは [General Purpose Interface Bus] の略称で、正式には IEEE Std.488-1978 として規定されています。デジタルマルチメータやオシロスコープといった計測器とコンピュータを接続するインタフェースの 1 つです。コンピュータと計測器とのインタフェースでは、GP-IB ほかに、RS-232C があります。(RS-232C については、第三回の「モデムで電話をかけてみよう」参照) GP-IB は RS-232C にくらべて高速です。また、多数の相手と通信ができるという特徴を備えています。ですから GP-IB は、データ計測や FA(Factory Automation)の現場などで利用されています。

GP-IB は通常パソコンに用意されていません。ボードメーカーから発売されている GP-IB ボードを購入し、パソコンの拡張スロットか PC カードスロットに取付けなければいけません。パソコンに取り付ける GP-IB ボードは複数のメーカーから発売されています。代表的なボードメーカーとしては

ナショナルインスツルメンツ株式会社

株式会社エヌエフ回路設計ブロック

ラトックシステム株式会社

があります。もちろん他にも多くのボードメーカーから GP-IB ボードは発売されています。

次に実際に GP-IB ボード選ぶ時の注意点をあげておきます。

- ?? GP-IB ボード制御用のアプリケーションインターフェースが用意されているか？。
- ?? パソコンに用意されているバスとボードのバスが一致しているか？。
- ?? ノートパソコン、デスクトップパソコンのいずれで使用するのか？

また、F-BASIC 用インターフェースがない場合がほとんどですが、C 言語用のインターフェースがあれば、変更して制御することができます。

計測器には、GP-IB に対応しているものであれば、GP-IB コネクタが用意されています。GP-IB ボードと計測器はコネクタどうしを GP-IB ケーブルで接続します。

なお、GP-IB ボードに関して詳しくは各ボードメーカーにお問い合わせください。

One Point Lesson

第六回「GP-IB ボードの制御をしよう」

さて、いよいよ肝心のプログラムに入りましょう。

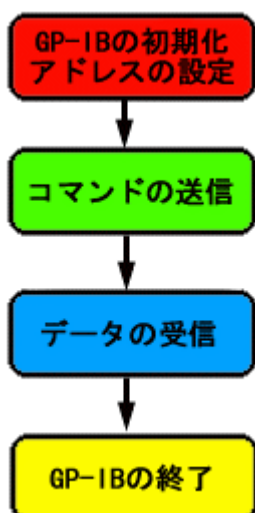
F-BASIC から GP-IB を制御するには、GP-IB ボードに対応したプログラム言語用 DLL を利用します。この時に、他言語呼び出しが必要になります。DLL は、GP-IB ボードとともにボードメーカーから提供されます。通常、GP-IB ボードのセットアップディスクによりパソコンにセットアップされます。

プログラムは、プログラム本体(GPIB.BAS)と DLL を利用するためのヘッダファイル(GPIB.BI)にわかれます。このプログラムは説明用に簡略化したものです、実際のボードには対応していません注意して下さい。

まずは、プログラム本体(GPIB.BAS)の方から説明していきます。

・制御の流れ・

GP-IB を制御するプログラムの流れは以下のようになります。



```
----- GPIB.BAS
-----

#include "GPIB.BI"
'
var BUFFER as string * 256
MYADDRESS& = 1
ADDRESS1& = 2
' 初期化、アドレスの設定
INIT MYADDRESS&
' コマンドの送信
SEND      ADDRESS1&,
"VOLTS?"
' データの受信
RECV      ADDRESS1&,
BUFFER, len(BUFFER)
print BUFFER ' データの表示
' 終了
TERM
stop
```

```
end
```

データの送信(プログラム 9 行目)

文字型データを利用します。文字型では最後にヌル文字(chr\$(0))が自動的に付加されます。

```
SEND ADDRESS1&, "VOLTS?"
```

データの受信(プログラム 3・11 行目)

文字型データを受信用バッファとして利用します。

```
' データの受信
```

```
var BUFFER as string * 256
```

```
RECV ADDRESS1&, BUFFER, len(BUFFER)
```

受信バッファは十分な領域を確保してください。受信データがバッファサイズよりも大きい場合は、データの後部が受信できなくなります。また、受信用の関数にバッファサイズを指定しない場合は、暴走する可能性があります。また、バッファサイズが足りない場合も暴走することがあります。

ヌルで終わる文字列の獲得して F-BASIC で扱うには以下のようにしてください。

```
A$=left$(BUFFER,instr(BUFFER,chr$(0))-1)
```

One Point Lesson

第六回「GP-IB ボードの制御をしよう」

次に、DLL を利用するためのヘッダファイル(GPIB.BI)について説明しましょう。

```
----- GPIB.BI -----  
  
declare sub INIT lib "GPIB.DLL" alias "Init" ( byval ADDR& )  
declare sub SEND lib "GPIB.DLL" alias "Send" ( byval ADDR&, byval DAT$ )  
declare sub RECV lib "GPIB.DLL" alias "Recv" ( byval ADDR&, byval DAT$,  
                                              byval  
DATLEN& )  
declare sub TERM lib "GPIB.DLL" alias "Term" ( )  
      (以下省略)...
```

declare 命令を使ってボードメーカーから提供されている DLL を使えるように宣言をします。GP-IB の DLL は BASIC 以外の言語で書かれていますので、DECLARE FUNCTION/SUB を使って、プロシージャの参照を宣言し、パラメータの型を指定します。

DECLARE FUNCTION 命令は関数プロシージャの参照を宣言し、パラメータおよび復帰値の型を指定します。

DECLARE SUB 命令はサブプロシージャの参照を宣言し、パラメータの型を指定します (declare 命令について詳しくは前回の「WindowsAPI を使ってみよう」をご覧ください)。

このヘッダーファイルは、プログラム本体の 1 行目(#include "GPIB.BI")で宣言されています。これによって、F-BASIC のプログラム内で GP-IB の DLL を利用できるわけです。

分かっていただけましたか？プログラム以外の知識が必要となるのでちょっと難しかったかもしれませんね。どのようなプログラムを組めば GP-IB を制御できるかは分かっていただけましたよね。実際に GP-IB ボードを制御できるサンプルプログラムをダウンロードしてお試し下さい。