

# プログラミング言語 LabVIEW による GPIB 制御

安間 義和

静岡大学 電子工学研究所 技術部

## 1. はじめに

NI (ナショナルインスツルメンツ) 社のプログラミングソフトウェア LabVIEW は、関数などにあたる VI (Virtual Instruments) やさらにまとめたサブ VI をアイコンとして扱い、それらを結線や枠で囲うなどをしてプログラムを組み上げることから、視覚及び直接的な感覚でプログラミングが可能と言われるプログラム言語である。また、計測器のインターフェイスとして一般的な GPIB を使用することで、取り扱いやすいシステムが構築可能である。

## 2. 目的

作製したデバイスに数ボルトを印加させ、その時々静電容量を計測しデータ収集することで特性曲線を得る事ができ、デバイスの評価をすることが可能である。計測点が多いため、計測者の負担を軽減することに加え、計測に関わる効率化のためにプログラム構築が必要である。また、計測器が一新し、今までの他言語プログラムに大幅な変更が必要になり、今回新しく導入した LabVIEW とを組み合わせて計測する GPIB 制御プログラムの作成をする事となった。

## 3. 計測器構成

DC 電源                      横河電機製 『yokogawa 9651』  
LCR メーター                日置電機製 『LCR3532-50』 オプション 『DC バイアス電圧ユニット』  
GPIB - USB インターフェイス  
制御用ノートパソコン

## 4. LabVIEW 用ドライバ

各メーカーで販売されている計測器の多くは GPIB が搭載されており、制御を可能とする LabVIEW ドライバも多くのメーカーが VI として無償提供をしている。メーカー運営サイトよりダウンロードができ、利用可能である。

しかし、これら2機種ドライバは、販売期間が長い計測器の初期に作成されたドライバでバージョンが古く対応していなかったり、ケーブル長の指定ができない、また計測器補正の周波数指定機能が不十分などがあった。メーカーに問い合わせ開発途中の別ドライバを個別受け取り試してみたが、やはりすべては改善されていなかった。

VI は1つにまとまっている点で利点はあるが、GPIB コマンドの割り込む余地がない事もあり、新たにドライバに頼らないプログラムを作成する事となった。

## 5. 簡易プログラムの作成

### 5.1 電圧を出力させるプログラム

DC電源に電圧1[V]を出力させるのプログラムを示す。(Fig. 1)

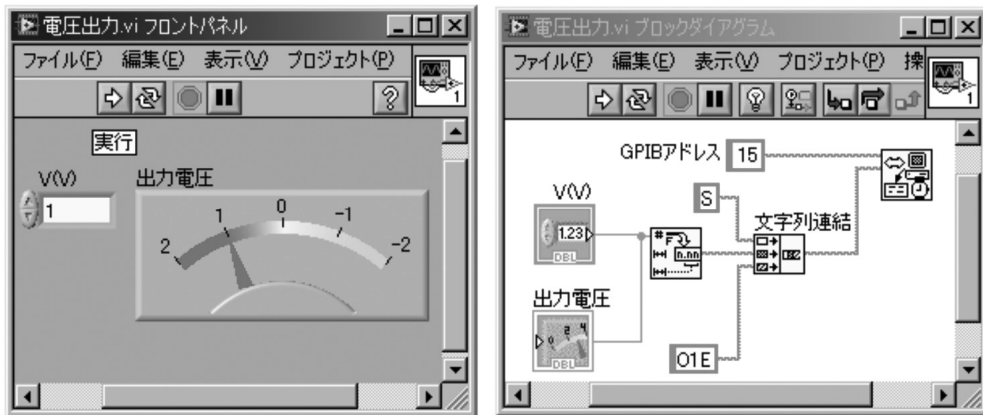


Fig. 1 電圧出力プログラム

プログラムを新規作成する際、フロントパネルとブロックダイアグラムが表示される。後者がプログラムに該当する。そこへ GPIB 書き込み関数、表示器、数値等をパレットよりドラック・ドロップにて配置し、GPIB アドレス 15 に指定した DC 電源の GPIB 書き込み関数に対し、文字列連結した GPIB コマンドの電圧レンジ S、正しい数値に変換した電圧 1 [V]、GPIB コマンドのアウトプット命令 O1E を結線する。結線はデータの種類などによって色分けされているが、モノクロ印刷なので判断はつかず説明は割愛させて頂く。この結線が間違えた場合、結線できない表示になる。また、実行も不可能な表示になり、視覚的にエラーがある状態か判断ができる。

### 5.2 プログラムの実行・停止

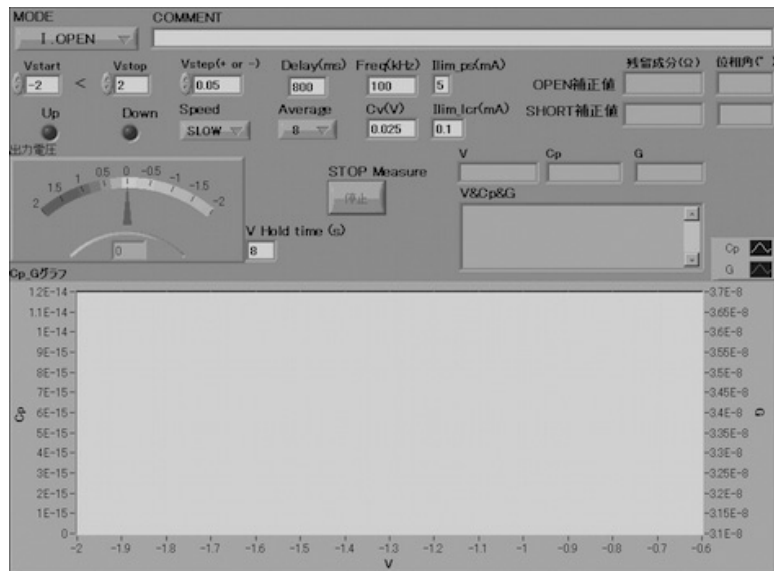
プログラムは、フロントパネルの右→(矢印)の実行アイコンをクリックすることにより実行する。この場合、1V が永久に出力される。停止は、実行アイコンの端の一時停止アイコンで行える。また、電灯のアイコンを押した状態で実行をした場合、結線上にデータが流れていく様子が実行行程として明るく描画され、プログラムの流れが判断できる。エラーがある場合は、その箇所で描画が停止し点滅する場合もある。

この簡易プログラムの出力電圧は、計測された電圧ではないので出力端子側で確実に電圧が印加されているかをテスターで確認した。

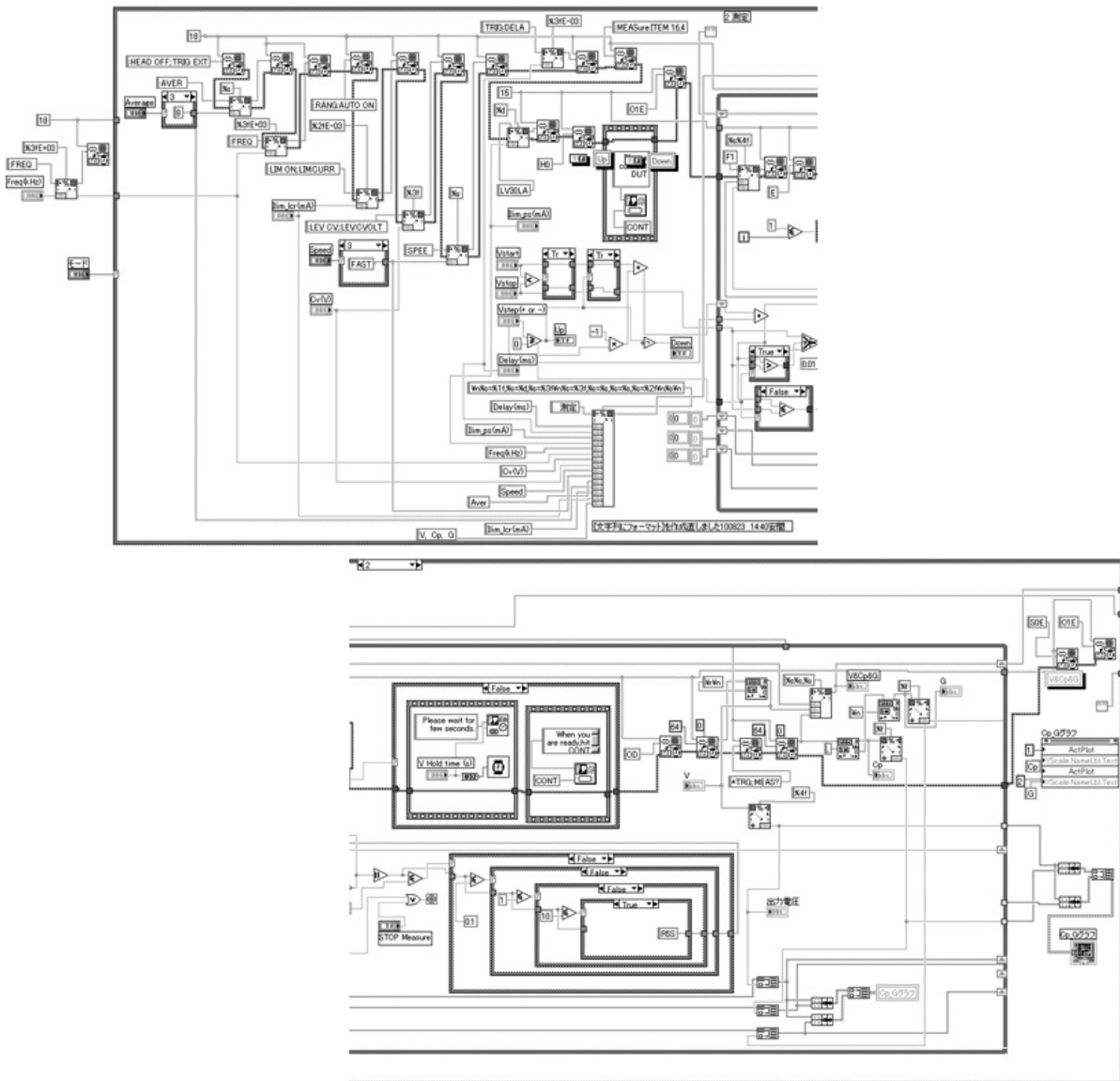
## 6. プログラム作成

### 6.1 計測プログラム

電圧を任意の範囲にスイープさせ、静電容量を計測。その値をグラフにプロットし最後にデータを保存を行う。下記に、計測プログラムのフロントパネル、ブロックダイアグラムを示す。(Fig. 2)



計測プログラムのフロントパネル



ブロックダイアグラム(中央にて分割) (Fig. 2)

## 6.2 オープン・ショート補正プログラム

計測プログラムは、計測前のオープン・ショート補正プログラムも加えた3部構成になった。ブロックダイアグラムでは3つのプログラム重なり合っていて切り替えで、表示・非表示が可能。オープン補正、ショート補正、計測の切り替えはフロントパネル左上[MODE]により行う。

## 6.3 プログラムの検証

市販品のデバイス2種類を用い、スイープ方向を反転させて得た特性曲線をメーカー値と比較することで検証を行った。また、実際作製したデバイスでも行い、こちらは過去計測した特性曲線と比較し、プログラムに不備がないか等総合的に検証した。

その結果、デバイスにかかる出力電圧に遅れがある事がわかった。原因は経由する DC バイアス電圧ユニットであった。プログラム内にその遅れを固定値として設定し修正をした。また計測器の設定 GPIB コマンドの不備をさらに追加することにより完成度を高め、再現性のある安定した結果が得られる計測プログラムとなった。

## 6.4 プログラムの改善

人為的ミスを防ぐためダイアログボックスを出現させ、作業が確実に行われたかを問いかける仕様した。また、今後計測者自ら変更修正やプログラム内容の理解ができるよう、VI 化はさけ、それにより肥大化しない様、表示器や結線の配置も工夫した。しかし、グラフ描画のサブ VI は煩雑をさけるために使用せざるを得なかった。

また、計測時に常に表示させるフロントパネルのパラメータ・グラフ配置も考慮した。

## 6.5 補正值データ表示プログラム作成

オープン・ショート補正後、その値がLCRメータの液晶画面で表示しない問題があり別途プログラムを作成した。

## 7 まとめ

- ・LabVIEW の GPIB 制御習得は参考になる書籍やサンプルプログラムが少ない
- ・他社製計測器のドライバは LabVIEW バージョンに対応しているか検証が必須
- ・プログラムがより複雑になるとブロックダイアグラムのウインドウ自体が巨大化する
- ・NI 社計測器を使用すれば、専用ドライバが充実していて GPIB 制御の概念がない
- ・初心者がプログラムに興味をもつきっかけとなるプログラム言語と感じた