

§2-I-1 LabVIEW の基礎実験 (Fundamental Experiment of LabVIEW)

1. 実験の狙い

本実験では簡単な例題を作成することで、LabVIEW というプログラミング言語によるソフトウェアの作成手順を学ぶ。本実験では LabVIEW でのプログラミング方法を理解し、計測機器を制御できるようになるために、以下の項目を理解することを目的としている。

- ① LabVIEW によるグラフィカルプログラミングの方法
- ② 計測機器との通信方法

2. 解説

LabVIEW

LabVIEW はグラフィカルプログラミング形式であり、文字（コマンド）によるプログラムと全く異なっている。プログラムは、オブジェクト間を結線（ワイヤリング）することにより作成する。また、LabVIEW で作成したプログラムは、VI (VI: Virtual Instruments: 仮想機器) と呼ばれる（拡張子も vi）。これは、オシロスコープなど測定器の操作パネルをイメージしたものであり、プログラムの開発作業は、まず、ウィンドウ（フロントパネル）上にノブ、スイッチ（入力器）、ディスプレイ（表示器）などを配置し、別のウィンドウ（ブロックダイアグラム）でそれらを結線することにより行われる。ただし、必要によりプログラムの一部を C 言語により記述することも可能である。

GP-IB インターフェース

GP-IB (General Purpose Interface Bus) は、コンピュータと計測器とのインターフェイスとして開発されたものであり、デジタイザチェーン接続された 15 台までの測定器に対し、コンピュータで制御・計測が可能となる。またデータ転送は 8bit の双方向パラレル通信にて行われるため、高速なデータ転送が可能である（現在では 8MB/sec）。

GP-IB では、1 台のトーカー（データ送信側）から複数台のリスナ（データ受信側）にデータ送信が可能であり、また受信処理速度が異なるリスナが混在していても問題なく送受信できるという特徴を有する。従って、データ転送は最も遅いリスナの受信速度に合わせて実施される。これを実現するシステムが 3 線ハンドシェイクである。3 線ハンドシェイクは、トーカーからリスナへ送信ことを示す（送信データが正当である）信号 (DAV: Data Valid)、リスナが受信未完了である信号 (NDAC: Not Data Accepted)、および受信準備未完了である信号 (NRFD: Not Ready For Data) の 3 信号を用いる（各々専用ラインを使用）ことにより、データ送受信の状態を確認するシステムである。各信号線は負論理（NRFD の場合、受信準備完了で High レベル）で、かつ、各装置の NDAC、NRFD はワイヤード OR 接続されていることから、全ての装置のデータ受信準備が可能となった時点で NRFD は High となる。よって、この状態を確認した後に次の送信を開始すれば全てのリスナに確実なデータ

送信が可能になる。

GP-IB は 8 本の双方向データライン (DIO1～DIO8), および 3 本のハンドシェークラインの他, 以下に示す 5 本の管理ライン (ATN, IFC, REN, SQR, EOI) を用いて (計 16 本) 各装置を効率的に制御している。

- ATN (Attention) : データラインのデータが, 装置制御データ (コマンド・モード) か測定データ (データ・モード) かを示す信号
- IFC (Interface Clear) : 装置を初期状態にするための信号
- REN (Remote Enable) : 各装置に対しリモート制御を可能とするための信号
- SQR (Service Request) : 装置が何らかのサービスを要求する際に使用する信号
- EOI (End Of Identify) : 転送データの終了を示す信号。

GP-IB 上では複数の計測器を接続可能であることから, 各装置に番号 (GP-IB アドレス) を定義する必要がある。GP-IB アドレスは 0 から 31 までの範囲で選択可能であり, 各装置で任意に設定できる。ただし, 1 つの測定システムにおいてアドレスは固有である必要があり (重複不可), また一般的に制御コンピュータのアドレスは 0 とすることが多い。

3. 実験の予習 *予習レポートとして整理して提出すること

予習1: 下記の用語について, 意味を調査し理解を進めておく。

(LabVIEW における言葉として調査してくること)

黒字は必ず。その他は余裕があれば。

- | | |
|-------------------|------------------------|
| a. LabVIEW | b. 自動計測 |
| c. ビジュアルプログラミング言語 | (グラフィカルプログラミング言語) |
| d. GP-IB | e. RS232C |
| f. USB2.0, USB3.0 | l. フローチャート |
| g. パラレル通信 | h. シリアル通信 |
| i. フロントパネル※1 | j. ブロックダイアグラム※1 |
| k. VI (仮想機器) | l. 発振器 (ファンクションジェネレータ) |
| m. デジタルマルチメータ | |

予習2: 作成する各プログラムについてコピー&ペーストできそうなところ (プログラム内容が同じところ) をピックアップしてくること。(レポートに記載する必要はない。ノートなどにまとめ, 実験時に利用すること。)

予習3: 例題 8 では RC 回路 (4 ページの図 1) の入力電圧, 出力電圧の周波数特性を測定する。そこで LTSpice を用いて周波数に対する入力電圧, 出力電圧の特性をシミュレートせよ。グラフの縦軸は [V] にすること。

※この時, 2 つの組合せをそれぞれ行うこと。(詳細は 4 ページを参照のこと)

4. 実験

4.1 実験の概要

別資料の手順を確認しながら、例題 1～例題 6 のプログラムを作成し、以下の項目が実現できることを確認する。

- 数値の入出力方法 (グラフ表示を含む)
- 数値演算方法 (C プログラミングによる手法を含む)
- 繰り返し演算方法 (For ループ・While ループ、スフとレジスタ)
- テキストファイル出力方法

次に、別資料の手順を確認しながら、例題 7, 8 のプログラムを作成し GP-IB インターフェースを用いた測定器の制御および計測データの入力方法の基礎について確認をする。

4.2 実験で使用する機材

- ・ パーソナルコンピュータ
- ・ 発振器 (WF1973)
- ・ デジタルマルチメータ (VOAC-7520H)

実験室の PC には（各自が持参した）USB を接続しないこと

4.3 実験終了後の確認事項

実験終了後 TA に以下の項目を確認してもらうこと。

- RC 回路の特性結果の確認
周波数特性（入力電圧，出力電圧）が正しく取れているか
- 各ファイルが全てあるか
vi ファイル，JPG，HTML ファイル
- 上記確認後デスクトップ（および PC 内）に実験データファイルが残っていないこと。
実験データは各自 Moodle 上で管理すること。

4.4 実験手順

[1] 実験の進め方

～14:00	手順 1：PC を起動し，LabVIEW のソフトウェアを起動する。 手順 2：例題 1 の実施 [資料 p. 2] 結果を画像として印刷し，VI ファイルと画像ファイルを保存する。 ★以下の例題でも同様にファイルを保存する。 手順 3：例題 2 の実施 [資料 p. 4]
～14:20	手順 4：例題 4 の実施 [資料 p. 6] 手順 5：例題 5 の実施 [資料 p. 7]
～15:00	手順 6：計測器（WF1973）との通信確認 [資料 pp. 9-10] 例題 7 の実施 [資料 p. 11-13] ※測定器の接続を十分に確認すること。
～16:00	手順 7：計測器（VOAC-7520H）との通信確認 [資料 pp. 14-15] 例題 8 の実施 [資料 pp. 16-18] ※測定用の回路を TA から受け取ること ※必ず回路と測定器を接続してから測定を行うこと。 手順 8：得られた情報から読みとれることをグループで話し合う 以下時間に余裕があれば行うこと。 手順 9：例題 3 の実施 [資料 p. 5] 手順 10：例題 6 の実施 [資料 p. 8]
～16:25	手順 11：TA による試問

[2] 各プログラムの詳細

- 各プログラムの詳細は Moodle 上に上がっているので，実験前に各自が確認をしておくこと。
- 例題 8 のプログラムを使用して，図 1 の RC 回路における入力電圧 v_{in} ，出力電圧 v_{out} 各々の周波数特性（100 Hz～100kHz）を測定し，フィルター特性（ v_{out} / v_{in} の周波数特性）を評価せよ。なお $R = 51 \Omega$, $C = 1\mu\text{F}$ の場合， $R = 5.1 \text{ k}\Omega$, $C = 0.01\mu\text{F}$ の場合の 2 種類について測定を行うこと。

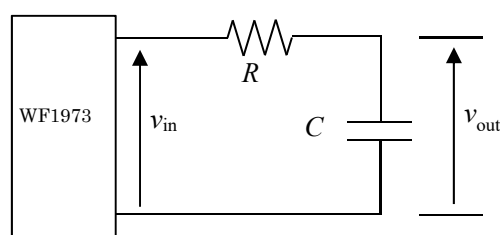


図 1 RC 回路

5. 実験レポート

検討すべき事項（考察事項）

実験レポートでは、実験結果を踏まえて検討を行い、下記の項目に関してもよく考察をすること。

- ① 各例題に対して考察せよ。
- ② 本実験で作成した例題のプログラムを参考に、これまでの手作業による計測と自動計測では得られるデータにどのような違いがあるか考察せよ。
- ③ 例題 8 の結果（各入力電圧 v_{in} 、各出力電圧 v_{out} ）について考察せよ。
回路図や数式を用いて考察すること。理論値を算出し、実験データと一つのグラフにまとめよ。
※グラフは入力電圧・出力電圧を分けて作成すること。

さらに調査すること

次回は本実験をベースに行うため、しっかりと理解した上で予習を行ってこよう。

課題

課題1： 自動計測を行う上での注意しなければならない点、メリット、デメリットを検討せよ。

※考察項目③から注意すべき項目の一つを考えだすとよい。

課題2： 自動計測に向く計測としてどのようなものがあるかを考え、説明せよ。

課題3： 例題 8 で作成したプログラムのフローチャートを示せ

課題4： 例題 8 のプログラムに機能を追加して、より測定者に扱いやすいプログラムとしたい。どのような項目が改善項目として挙げられるか、最低 2 つ以上を考え説明せよ。

6. 参考文献

- ・ Robert H. Bishop 著，尾花健一郎 訳「LabVIEW プログラミングガイド」（アスキー）
- ・ 井上泰典 著「LabVIEW グラフィカルプログラミング」（森北出版）
- ・ 鈴木昇 著「はじめてのラボビュー [基礎編]」（東京教学社）

1. プログラム作成の手順

・準備（LabVIEW の起動）

- 1) 「スタート」>「全てのプログラム」>「National Instruments LabVIEW」をクリックし、LabVIEW を起動する。
- 2) スタートアップウィンドウ > プロジェクト作成 > 「ブランク VI」を選択。これにより、「フロントパネル」「ブロックダイアグラム」の2つのウィンドウが開き、新たなプログラム作成が可能になる。なお、フロントパネルのみしか表示されない場合は、「ウィンドウ」>「ブロックダイアグラムを表示」（Ctrl+E）により、ブロックダイアグラムを表示させる。
*Ctrl+E（ショートカット）は、「フロントパネル」と「ブロックダイアグラム」とを交互にアクティブとするショートカットとしても働き、プログラム作成中多用する。
- 3) フロントパネル選択時、サブウィンドウとして「制御器」が表示されていない場合、「表示」>「制御器パレット」を選択し、「制御器」を表示しておくプログラム作成時に便利である。

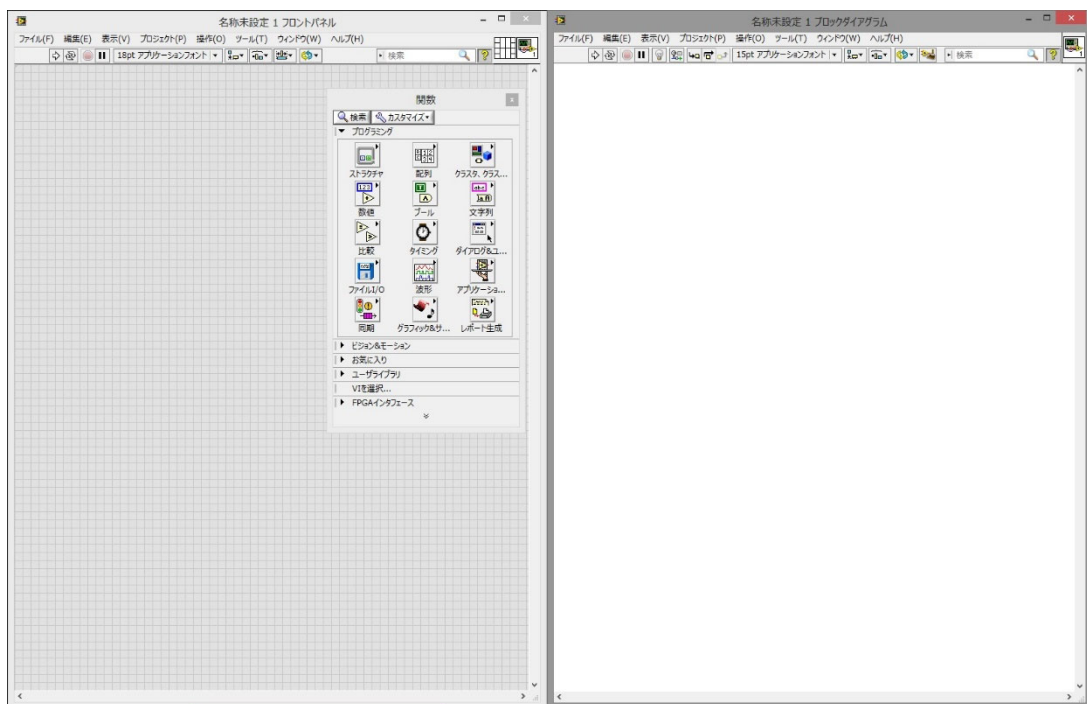


図2 「フロントパネル」と「ブロックダイアグラム」ウィンドウ。



フロントパネルをクリックすると「制御器」が、ブロックダイアグラムをクリックすると「関数」がサブウィンドウとして表示される。



※一度作成したプログラムはコピー&ペーストでどんどん利用すること。
ファイルへの出力など利用できるところをピックアップしておくこと。

[1] 例題 1：値の入力と表示 [結果は保存しない]

「ノブ」で入力したデータを「メータ」に表示するだけの簡単なプログラムを作成する。

- 1) フロントパネルを選択し、制御器パレットにおいて

Express>数値制御器>ノブ を選択し、フロントパネルに配置（ドラッグ）

- 2) Express>数値表示器>メータ を選択し、フロントパネルに配置

- 3) ブロックダイアグラムにて、「ノブの出力」と「メータ入力」を接続
(出力ノードと入力ノードで左クリック)

- 4) フロントパネルの左上、2つの矢印のアイコンをクリックすることにより、連続実行するので、マウスによりノブを自由に動かし、本プログラム動作を確認する。

- 5) 確認後、赤い丸（実行を中断）をクリックし、動作を止める。

- * 左上、右向きの矢印は「1 回実行」を意味し、実行時のノブの状態をメータに表示するとプログラムは終了する。「連続実行」を選択すると、ノブの変化に応じてメータ表示が変化する。

- 6) ファイル>保存 により所定のフォルダにプログラムを保存する。

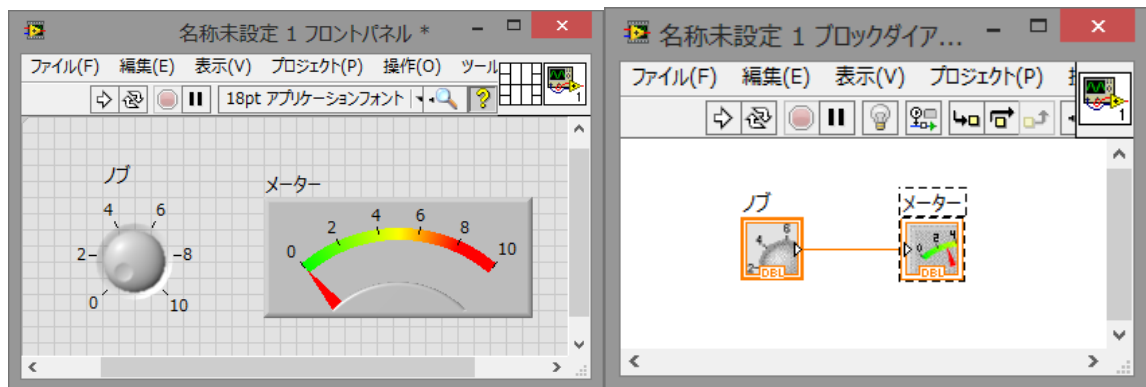


図3 例題1のフロントパネルとブロックダイアグラム

[extra1] ファイルの保存 [すべての例題]

- 下準備
 - vi ファイルや画像を保存する下準備として、デスクトップに班番号のフォルダを作成する。
ex.) 「火 01」, 「木 04」 など ← メインフォルダと呼称する。
 - 作成した班番号のフォルダの中に例題番号のフォルダを分けて管理するとよい。
ex.) 「rei01」, 「rei02」 など ← サブフォルダと呼称する。
- vi ファイルの保存
 - ファイル>保存 を選択し、上記で作成したサブフォルダに vi ファイルを保存する。
- ファイルの管理
 - メインフォルダを ZIP 化し、Moodle のプライベートフォルダに保存する。
Home>マイプロフィール>マイプライベートファイル
ドラッグアンドドロップで OK
 - ZIP 化
フォルダを右クリック>送る>圧縮 (ZIP 形式) フォルダ を選択
 - Moodle への移動は全実験終了後でよい。



[extra2] 画像の保存 [結果を保存すると書いてある例題用]

レポートにまとめるため、フロントパネル、ブロックダイアグラムともに画像として保存する方法を紹介する。

- 注意事項
 - 画像の保存は実験ごとに行うこと。実験終了後このフォルダごと消すこと。
- 画像の保存
 - ファイル>印刷 を選択
 - 該当する vi ファイルを選択し、「次へ」 をクリック
 - アイコン、説明、フロントパネル、ダイアグラム を選択、「次へ」 をクリック
 - HTML ファイル を選択、「次へ」 をクリック
 - 画像形式を「JPG (劣化)」 に変更、「保存」を選択
ファイルの保存で作成したサブフォルダに保存する。
- フロントパネルに異なる実験結果が得られるような実験では、それぞれ画像ファイルを保存すること。

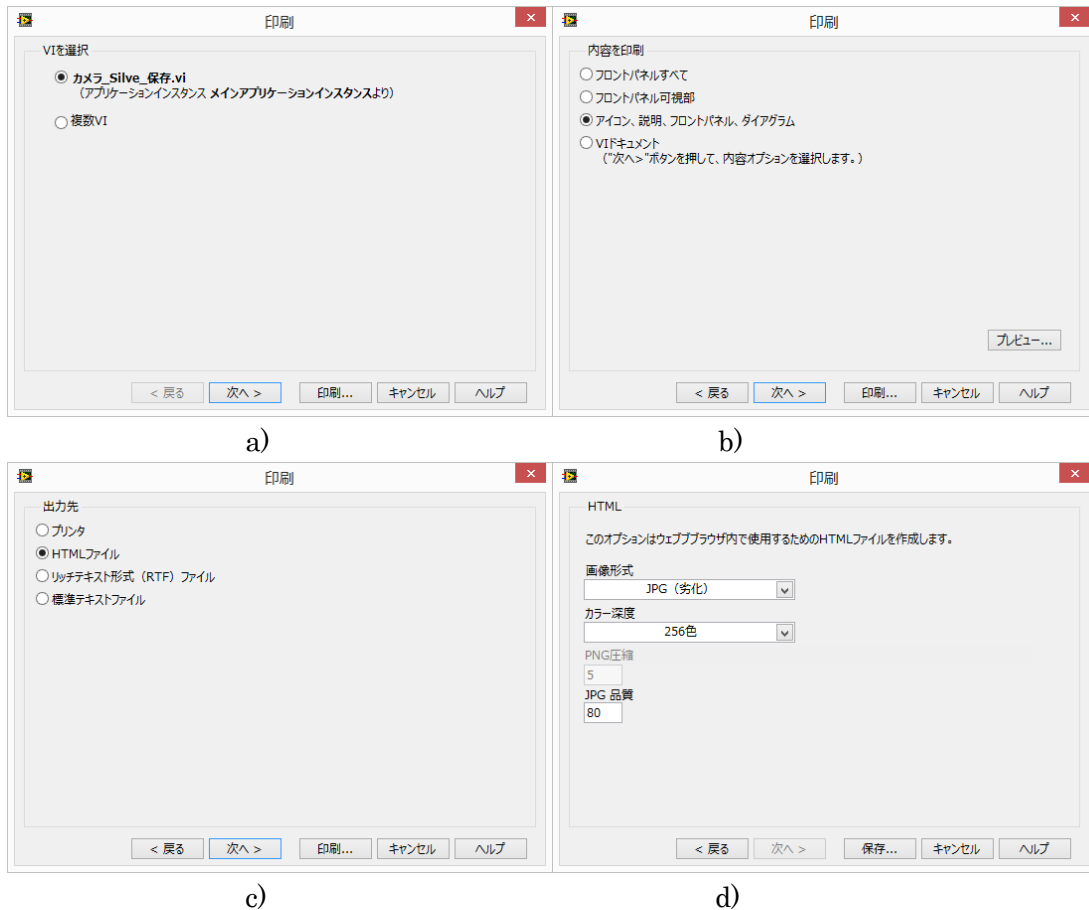


図 4 画像保存の流れ

[2] 例題 2： 入力値の 2 乗をメータに表示 **【結果を保存する】**

- 1) 例題 1 のプログラムを用いる.
- 2) ブロックダイアグラムにて, 「ノブ」「メータ」間の配線を右クリック
- 3) 挿入>数値パレット> x^2 を選択
- 4) 連続実行により動作を確認する. (ノブの値の 2 乗がメータに表示される)
- 5) メータの最大値などを変更する
フロントパネル上で メータを右クリック>プロパティ により各種設定する.
 - ・「外観」では, 指針の色, デジタル表示器の設置 などの設定可能
 - ・「スケール」では, 最小値, 最大値 などの設定可能

*設定後, メータを右クリック>データ操作>現在の値をデフォルトに設定 を選択すると, 再起動後でも現在の設定が再現される.

- 6) メータの最大値などをプログラムにて (実行中に) 変更する
 - ・ブロックダイアグラムにてメータを右クリック
 - ・作成>プロパティノード>スケール>範囲>最大値 を選択, 任意の場所に配置
 - ・配置されたアイコンを右クリック>全てを書き込みに変更
 - ・上記の入力部を右クリック>作成>制御器 により, 数値制御器を配置 (最小値についても同様に制御可能)
 - ・連続実行により動作確認

* これにより, 実行中に最大値を変更 (数値入力) できるようになる.
- 7) ファイル>別名で保存 により所定のフォルダにプログラムを保存する.

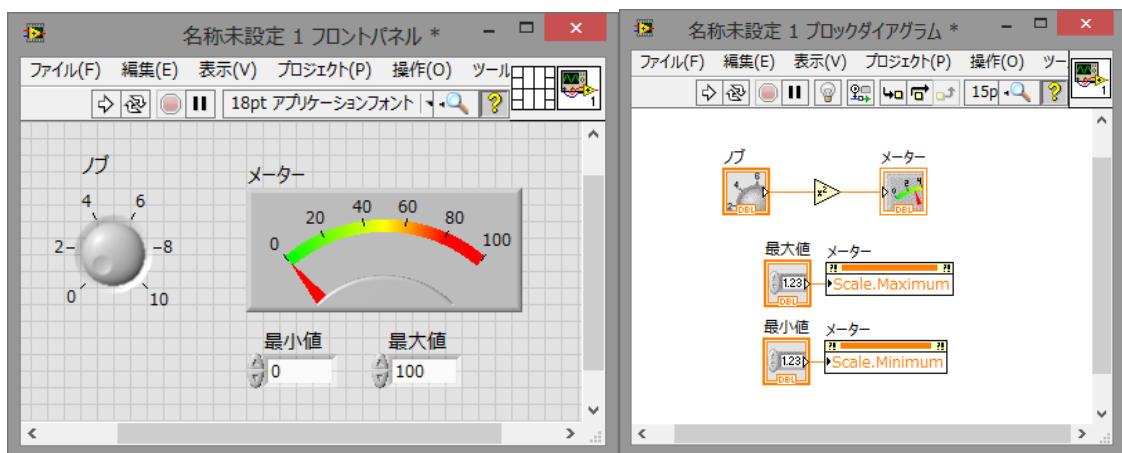


図 5 例題 2 のフロントパネルとブロックダイアグラム

ここで, LabVIEW プログラミングにおいて知っておくと便利な事項を以下に示す.

・エラー表示

1 回実行の矢印が「黒色」の状態は, 未接続部分があることを示す. なお強制的に実行すると, エラー部分が表示される.

・不良ワイヤの消去

ワイヤの接続を消去した場合, 関連したワイヤに×印が付き未接続状態 (不良ワイヤ) になる. これを一括して消去する場合は, **Ctrl-B** で実行できる.





以上, 例題(1), (2)のプログラムにより, LabVIEW の最も基本となる入出力および演算の概念を学んだ. 次に, 繰り返し計算と計算結果のグラフ表示について学ぶ.

時間に余裕があれば行うこと。【結果は保存する】

[3] 例題 3: $y = 3x^2$ をグラフ表示する.

x の範囲を 0~10, 間隔 0.1 (計算点数: 101 点) で計算を行い, グラフ表示する.

繰り返し回数 101 回の For ループによりプログラミングする.

- 1) ブロックダイアグラムにて, プログラミング>ストラクチャ  > For ループ  を配置
- 2) For ループの N (繰り返し回数) を右クリック>定数を作成 を選択
定数およびワイヤーは青色で表示される. これは「整数」を意味する. ちなみに「実数」はオレンジ色で表示される.
- 3) この定数に「101」を入力 (繰り返し回数=101)
- 4) 「i」は現在の実行回数 (0~100) を示すので, For ループ内で $i \cdot 0.1$ を計算することにより x (0~10) を得る. なお, 「i」は 0 から開始されるので注意すること.
 - ・プログラミング>数値>積 を配置, 一方の入力を「i」と接続
 - ・もう一方の入力を右クリック>定数を作成 を選択
 - ・この定数は青色 (整数) なので, 定数を右クリック>表記法>DBL により実数化
*DBL は倍精度実数. SGL (単精度実数) を使用しても良い.
 - ・定数に「0.1」を入力
- 5) 4)と同様に $y = 3x^2$ を計算するよう結線する.
- 6) x, y の計算結果を For ループの壁に結線する (For ループから出力)
- 7) 6)のポイントがオープンマーク (オレンジで塗られていない) ことを確認する. オレンジで塗られたマークの場合, マークを右クリック>指標付け使用 を選択
*「指標付け使用」(オープンマーク) の場合, 出力は 101 回の計算結果を配列として出力する (1 行, 101 列). 「指標付け不使用」(クローズマーク) の場合, 最後 (101 回目) の計算値のみが出力される.
- 8) フロントパネルにて, Express>グラフ表示器  > ExpressXY グラフ  を選択, 配置.
- 9) ブロックダイアグラムにて, 6)の計算結果 (x, y) を「XY グラフ作成」の X 入力, Y 入力にそれぞれ接続する. すると, 実数配列データ (オレンジの太線) からダイナミックデータ (グラフ入力用データ) への変換が自動的に行われる.
*オレンジの細線は「実数」, オレンジの太線は「実数配列」を示す.
- 10) 「1 回実行」を行い, 動作を確認する.
- 11) ファイル>保存 により所定のフォルダにプログラムを保存する.

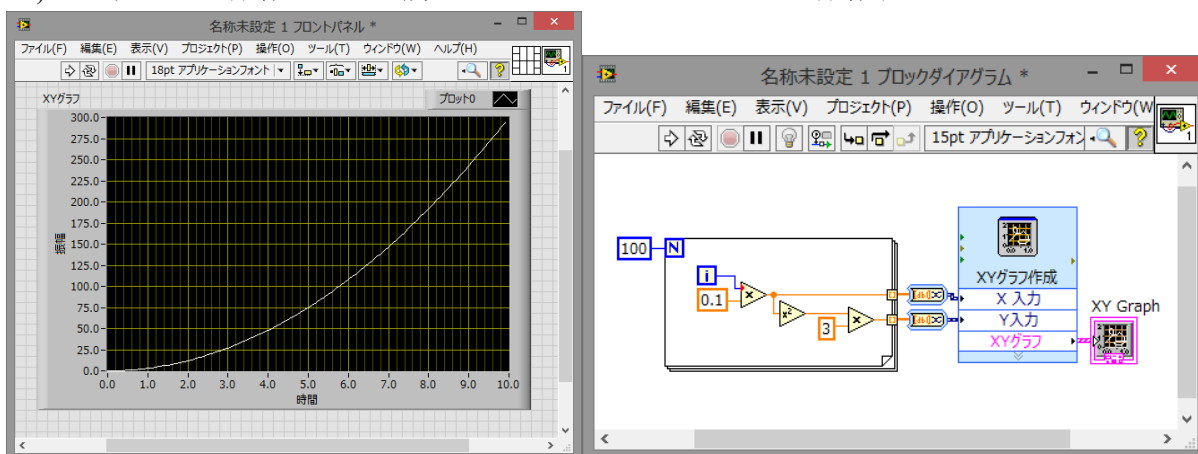




図 6 例題 3 のフロントパネルとブロックダイアグラム

[4] 例題 4：C 言語にて $y = 0.5x^3$ の計算を行なう。[結果は保存しない]

- 1) 例題 3 のプログラムを使用
- 2) プログラミング>ストラクチャ  > フォーマーノード  を For ループ内に配置
- 3) フォーマーノードの壁を右クリック>入力を追加 により，必要な入力を作成
フォーマーノードの壁を右クリック>出力を追加 により，必要な出力を作成
(入力: x, 出力: y として， $y = 0.5x^3$ の計算を行うものとする)
- 4) 入出力の関係を C 言語にてプログラム (出力しない変数でも，途中計算などで使用する変数は，必ず全て定義)
- 5) x の計算結果を，フォーマーノードの x 入力に接続. 計算結果を For ループから出力.
- 6) For ループからの出力 (配列) を，接続済みの ExpressXY グラフ(y)に 2 重接続 (既に入力されている端子に重ねて接続) すると，自動的に結合子が生成される.
- 7) 同じ x を用いる場合は，同じ配列を ExpressXY グラフ(x)に 2 重接続する.
- 8) 動作確認 (1 回実行) 後，プログラムファイル保存.

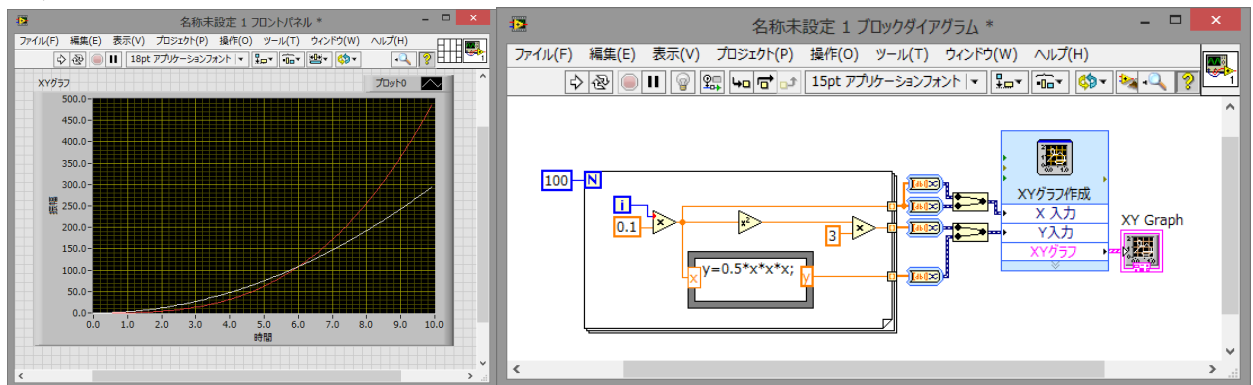


図 7 例題 4 のフロントパネルとブロックダイアグラム

[5] 例題 5： 計算結果のファイル作成 [結果を保存する]

例題 4 のプログラムの計算結果をテキストファイルとして出力する。

$y_1 = 3x^2$, $y_2 = 0.5x^3$ とし, x, y_1, y_2 の各 1 次元配列 (101 データ) を連結し, 2 次元配列とする (3 行 101 列). Excel などを使用することを考え, この行列を転置 (101 列 3 行).

これをテキスト化し, タイトル (標題) を付けて出力 (保存) する。

- 1) 例題 4 のプログラム使用。
- 2) ブロックダイアグラムにて, プログラミング>配列 $\left[\begin{smallmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{smallmatrix} \right]$ >配列連結追加 $\left[\begin{smallmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{smallmatrix} \right]$ を配置
- 3) 右クリック>入力を追加 または, アイコンを下に延ばして, 3 配列入力とする
- 4) x, y_1, y_2 データ (1 次元配列) を入力 (出力は 2 次元配列となる (3 行 101 列))
*ただし, 右クリックで「入力を連結」を選択すると, 1 次元配列として追加される
- 5) プログラミング>配列 $\left[\begin{smallmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{smallmatrix} \right]$ >2D 配列転置 $\left[\begin{smallmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{smallmatrix} \right]$ を配置, 接続
*これにより, 縦方向に長い (101 行 3 列) 配列になる
- 6) プログラミング>文字列 $\left[\begin{smallmatrix} abc \\ 123 \end{smallmatrix} \right]$ >配列からスプレッドシート文字列に変換 $\left[\begin{smallmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{smallmatrix} \right]$ を配置
「形式文字列」端子を右クリック>作成>定数 で文字定数を作成, %g を入力する。
・「配列入力」に転置後の 2 次元配列入力
・「デリミタ」には, 数値間の区切り文字を入力 (無入力では Tab)
・「形式文字列」には, 数値から文字への変換方式を入力
(%g : 自動, %g : 0 の羅列削除, %d : 整数, %f : 実数 など)
- 7) プログラミング>配列 $\left[\begin{smallmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{smallmatrix} \right]$ >配列連結追加 (2 列) $\left[\begin{smallmatrix} 1 & 2 \\ 3 & 4 \end{smallmatrix} \right]$ を配置 (列タイトル作成のため)
- 8) 6) の 1 列目の要素入力を右クリック>作成>定数 で文字定数作成
 x ”Tab” y_1 ”Tab” y_2 と「タブ」が入った文字の入力可能とするため, 文字定数を右クリック>”¥”コード表示 を選択. $x¥ty_1¥ty_2$ を入力 (¥t が”Tab”を意味する).
- 6) の 2 列目にはスプレッドシート文字列 (5) の出力) を入力
- 9) プログラミング>ファイル IO $\left[\begin{smallmatrix} 1 \\ 2 \end{smallmatrix} \right]$ >テキストファイルに書き込む $\left[\begin{smallmatrix} 1 \\ 2 \end{smallmatrix} \right]$ を配置
7) の出力を接続
- 10) 実行確認 (1 回実行). 出力ファイルを”任意名.txt”で保存. Excel など内容を確認

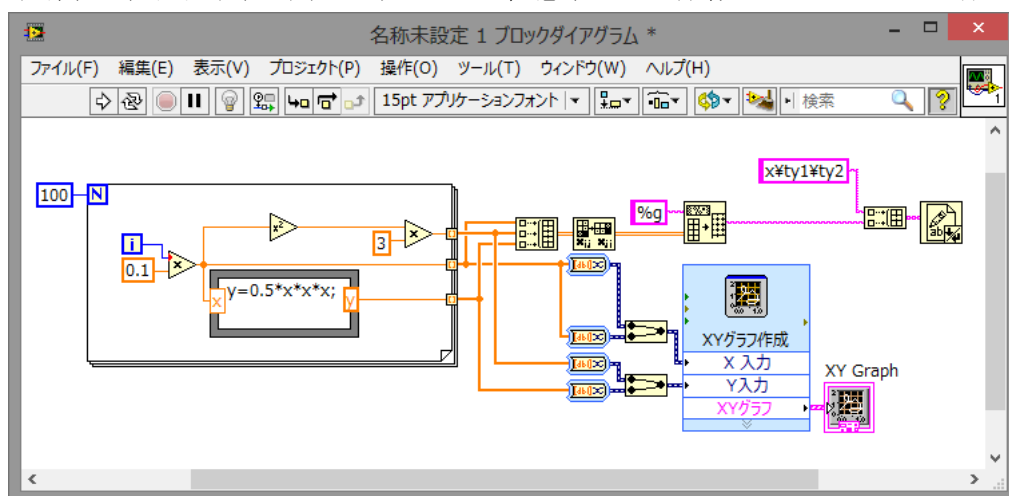










図 8 例題 5 のブロックダイアグラム

[6] 例題 6： 整数を 1 から積算し, a(入力値)以上となったときの整数, 積算値を出力する。

- 1) ブロックダイアグラムにて プログラミング>ストラクチャ  > While ループ  を配置
 - ・ i は繰り返し回数 (0 から増加) を出力
 - ・ 「赤丸」は繰り返し終了条件を入力する端子である。
- 2) プログラミング>数値  > 数値定数  を While ループ外に配置. 0 を入力 (積算初期値)
 整数にするため, 定数を右クリック>表記法>I32 で整数化
- 3) While ループ境界を右クリック>シフトレジスタを追加 を選択. 2)の定数をシフトレジスタの入力 (左側) に接続
 - ・ 本プログラムでは, 積算値 (初期値 0) に繰り返し回数 (i) を加算し, その結果に対し再度 i を加算する という作業を繰り返して積算値を算出する. よって, 現在の計算結果 (出力) を次の繰り返し (入力) にフィードバックする必要がある. これを可能とするのが「シフトレジスタ」であり, 「シフトレジスタを追加」により一対の入出力が作成される. なお, 外部から入力された値は初期値として適用される。
- 4) シフトレジスタ (入力) と繰り返し回数 (i) を加算.
 結果をシフトレジスタ (出力) に接続
- 5) フロントパネルにて, Express>数値制御器  > 数値制御器  を配置 (a 入力用)
 ブロックダイアグラムにて, 変数名を a に変更. 右クリック>表記法>I32 で整数化
- 6) ブロックダイアグラムにて, プログラミング>  比較 > 以上?  を While ループ内に配置
 4)の加算結果, 5)の入力用アイコンを入力. 出力を「赤丸」に接続
 - ・ これにより, 積算結果が a 以上となったときに, 繰り返しが終了する. また, 緑線はブール変数 (True or False) を示す。
- 7) i および 4)の積算結果を While ループの右側境界に接続 (出力).
 各々右クリック>作成>表示器 により結果を出力 (適宜名前を変更)
- 8) フロントパネルにて, 数値制御器, 数値表示器を適宜配置
- 9) 実行確認 (a に数値入力後, 1 回実行)

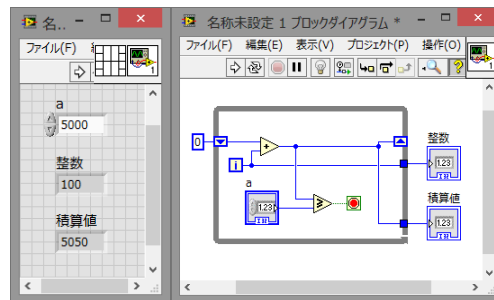


図 9 例題 6 のフロントパネルとブロックダイアグラム

以上の演習により, LabVIEW プログラミングにおける

- ・ 数値入出力の方法 (グラフ表示含む)
- ・ 数値演算方法 (C プログラミングによる手法を含む)
- ・ 繰り返し演算方法 (For ループ・While ループ, シフトレジスタ)
- ・ テキストファイル出力方法

などの基本を学んだ。

必ず確認すること。

例題 7 の準備 GP-IB による計測器制御

次に、GP-IB インターフェイスを用いた測定器の制御および計測データの入力方法の基本について演習を行う。

なお、情報工学実験で使用可能な機器は以下の通りである。


表 1 使用可能な機器

機器名	製造会社	型番	機器名	製造会社	型番
ファンクションジェネレータ	NF	WF1973	マルチメータ	IWATSU	VOAC-7520H
			直流電源	INSTEK	GPD-4303S

※直流電源を使用する場合は基本的に「PSS-3203」を使用すること。

ファンクションジェネレータ NF 社製 WF1973 の制御

最初に、ファンクションジェネレータ WF1973 を用いて、計測器の接続 および GP-IB 接続状態の確認の手法について解説する。

- 1) GP-IB アドレス・デリミタの設定 (WF1973) (設定されているので行わない.)
 - ・ WF1973 の電源を ON にする。
 - ・ [MENU]キーを押し、ポップアップ Menu の[4: Utility]をモディファイノブを回して選択し[Enter]キーを押しす。
 - ・ [Utility]リスト中の[Remote]をモディファイノブを回し選択し、[Setup]にフォーカスがあることを確認して、[Enter]キーを押す。
 - ・ [Interface]を “GPIB” とし、GPIB Address を所望の番号とする。
本書では WF1973 の GP-IB アドレスを 2 (工場出荷時設定) に設定したとして説明
 - ・ 再度[MENU]キーを押して、ポップアップ Menu の[1: Oscillator] モードに戻しておく。
- 2) 測定器の接続
 - ・ WF1973 の電源を OFF にする。
 - ・ PC の USB 端子に GP-IB インターフェイス (GPIB-USB-HS または GPIB-USB-B) を接続し、GP-IB 端子を測定器 (ここでは、ファンクションジェネレータ WF1973) に接続する。
 - ・ WF1973 の電源を ON にする。
- 3) GP-IB 接続状態の確認
 - ・ Windows デスクトップの “NI MAX”  を実行 (ダブルクリック)
(NI MAX : Measurement & Automation)
 - ・ 左側の「構成」において、「デバイスとインターフェイス」をダブルクリック
 - ・ GPIB0 (“インターフェイス機器名”) を左クリック
 - ・ 右側 Window 上部の「計測器をスキャン」を左クリック
 - ・ 下側 Window に、WF1973 が現れることを確認し、これをダブルクリック
 - ・ プライマリアドレスが 1)で設定したアドレスであることを確認
 - ・ 以上により、GP-IB 接続が成立していることが確認できる。
- 4) GP-IB による WF1973 との通信
 - ・ 右側 Window 上部の「計測器と通信する」を左クリック
 - ・ 「文字列を送信」欄の “*IDN?” を削除し、:FREQ 2e3 と入力し、「書き込み」をクリック。
この時、WF1973 の周波数表示が 2.0 kHz となることを確認する。
 - ・ 同様に、:VOLT 0.5 と入力し「書き込み」。これにより振幅が 500 mV に変化する。

- ・ 同様に、:FUNC SQU と入力し「書き込み」。これにより波形が方形波に変化する。
- ・ ここで、:FREQ? と入力し「書き込み」後、「読み取り」をクリック。この時、WF1973 に表示されている周波数が「受信した文字列」に表示されることを確認する。

以上の操作により、測定器が GP-IB 接続され、かつ PC から制御可能であることが確認できた。また、4)で使用した、:FREQ 2e3, :VOLT 0.5, :FUNC SQU などが WF1973 に対する GP-IB コマンドである。以下に WF1973 の主な GP-IB コマンドを示す。

WF1973 の主な GP-IB コマンド

出力波形設定 :FUNC <出力波形>

:FUNC:SHAP SIN	正弦波
:FUNC:SHAP SQU	方形波
:FUNC:SHAP PULS	パルス波
:FUNC:SHAP RAMP	ランプ波
:FUNC:SHAP DC	直流

※ex) 「SOUR1:FUNC SIN」・・・チャンネル1の信号を正弦波にする

振幅設定 :VOLT ## <##に数値を入力, 指数表示 (1.00e-1 など) も可>

設定値は 50Ω 終端時の Peak to Peak 値となる (DC の場合は 50Ω 終端時の電圧値)。

:AMPL 5.00 と設定した場合、

50Ω 終端時： ±2.5V の信号が出力。直流の場合は、+5V の電圧が出力。

開放終端時： ±5.0V の信号が出力。直流の場合は、+10V の電圧が出力。

(マルチメータなどを直接接続した場合)

出力インピーダンス = 50Ω なので測定時には注意が必要となる。

★出力インピーダンスの設定

図 10 の「High-Z」と表示してある部分を選択し、「Output Load」とポップアップメニューが出てきたら、現在のインピーダンスの設定値 (50Ω or High-Z or Variable)を選択し、所望の値 (本実験では 50Ω) に変更をする。

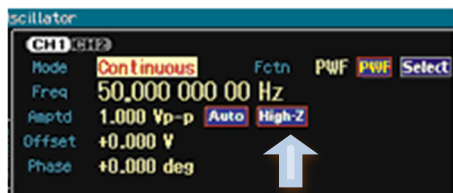


図 10 出力インピーダンスの設定



周波数設定 :FREQ ## <##に数値を入力, 指数表示 (2.00e3 など) も可>

出力設定 :OUTP ON で出力 ON. :OUTP OFF で出力 OFF.

[7] 例題 7: WF1973 に対する周波数制御プログラムの例 [結果を保存する]





振幅 1 V_{p-p} (マルチメータ接続時など開放終端時は 2 V_{p-p}) の正弦波を, 100Hz ~ 100 kHz の範囲で変化させるプログラムを作成する. なお, 周波数 1 桁 (1 decade) 当たり対数軸で等間隔に 5 分割 (総周波数ポイント = $5 \times 3 + 1 = 16$ ポイント) するものとし, 周波数変化後 10 秒間保持するものとする. この時, 1~10 kHz における設定周波数は以下の通りとなる.

1 kHz (10^3), 1.58 kHz ($10^{3.2}$), 2.51 kHz ($10^{3.4}$), 3.98 kHz ($10^{3.6}$), 6.31 kHz ($10^{3.8}$), 10 kHz (10^4)

- 1) フロントパネルにて, Express > 数値制御器  > 数値制御器  を配置 (GP-IB アドレス入力用)

変数名を WF1973 GP-IB Add.に変更. 右クリック > 表記法 > I32 で整数化

*なお GP-IB アドレスのデフォルト値を 2 にしたい場合 (プログラム立ち上げ時に 2 に設定) は, 2 を入力後, 制御器を右クリック > データ操作 > 現在の値をデフォルト設定にする を選択する.



- 2) ブロックダイアグラムにて, 計測器 I/O > GPIB  > GPIB 488.2 より, IFC , REM , CLR  アイコンを各々配置. 各命令の内容を以下に示す.





IFC (InterFace Clear): 接続された全装置の GP-IB インターフェイスを初期化する

REM (Enable Remote): 接続装置をリモート状態 (GP-IB 制御可能状態) にする




CLR (Device CLear List): 接続装置を初期化する

なお, REM, CLR は, 対象装置 (GP-IB アドレス) を指定して行い, また複数機器に対して同時に行えるよう, GP-IB アドレスを 1 次元配列として指定する (対象装置が 1 台の場合でも, 配列化する必要がある).

- 3) ブロックダイアグラムにて, プログラミング > 配列  > 配列連結追加  を配置. 1) の出力 (GP-IB アドレス) を入力. 出力を, REM, CLR の「アドレスリスト」に入力.
- 4) IFC の「エラー出力」を REM の「エラー入力」に, REM の「エラー出力」を CLR の「エラー入力」に各々接続. これにより, IFC→REM→CLR とエラーを確認しながら順次処理が進む.
- 5) 機器初期化に必要な待ち時間を設ける
機器初期化には 100~300 ms ほどの時間が必要 (機器に依存) であり, 初期化終了前に制御コマンドを入力するとエラーになる. ここでは, 2500 ms の待ち時間を設けるものとする.

- ・ プログラミング > ストラクチャ  > フラットシーケンスストラクチャ  を配置
 - ・ CLR の「エラー出力」をフラットシーケンスストラクチャに入力し, そのまま出力.
 - ・ プログラミング > タイミング  > 待機  をフラットシーケンスストラクチャ内に配置
 - ・ 「待機」の入力を右クリック > 作成 > 定数 で定数作成, 定数を 2500 とする.
- これにより, 2500 ms 待機後に次の GP-IB 処理が開始される.

- 6) 出力波形, 出力振幅, 初期周波数の設定, 波形出力





- ・ 計測器 I/O > GPIB  > GPIB 488.2  > SEND (送信)  を配置
- ・ 5) の出力を「エラー入力」に接続
- ・ 1) の出力を「アドレス」に入力 (3) の配列ではない)












- ・ 「データ文字列」を右クリック>作成>定数 で以下の文字定数を別々に作成



SOUR1:FUNC SIN (または、:FUNC SIN でもよい)

:VOLT 1.0

:FREQ 1.0e2

:OUTP ON
- ・ プログラミング>文字列  > 文字列連結  を配置し、左クリックで「文字列連結」を選択し、中央部分に表れる青い四角を左クリックし下方向(or 上方向)に広げ8段にする。
- ・ 上記作成した各文字定数を「SOUR1:FUNC SIN」から順に一番上から一段おきに「文字列連結」に接続する。
- ・ プログラミング>文字列  > 復帰文字定数  を配置する。「文字列連結」の抜かしていた段に「復帰文字定数」を接続する。(図11を参照のこと)

これにより、振幅1V、周波数100Hzの正弦波が出力される。
- 7) Forループ配置。繰り返し回数(N)=16に設定
- 8) 周波数の計算、および周波数制御コマンドの作成
 - ・ $i \cdot 0.2 + 2$ を計算。数学>基本&特殊関数  > 指数関数  > 10のX乗  により周波数データ計算
 - ・ プログラミング>文字列  > 文字列/数値変換  > 数値を指数文字列に変換  を配置。周波数データを「数値」に入力。「精度」に6を入力(有効数字7桁となる)。この文字列データを表示器により表示。
 - ・ プログラミング>文字列  > 文字列連結  を配置。1番目の入力を文字定数:FREQとする。(Qの後にスペース挿入)。2番目の入力に周波数データ(指数文字列)を入力
- 9) 周波数制御コマンドの送信
 - ・ 計測器 I/O>GPIO  > GPIO 488.2  > SEND (送信)  を配置(既に配置されているアイコンを、Ctrlを押しながらドラッグしても良い)
 - ・ 8)で作成した周波数制御コマンドを「データ文字列」に入力
 - ・ 1)の出力を「アドレス」に入力
 - ・ Forループに「シフトレジスタ」を作成。6)の「エラー出力」を左側のシフトレジスタ(入力)に接続。これを「エラー入力」に接続する。
 - ・ 「エラー出力」を右側のシフトレジスタ(出力)に接続。

これにより、繰り返しの度にエラー情報が更新される。(例題6参照)
- 10) プログラミング>タイミング  > 次のミリ秒まで待機  をForループ内に配置。入力に定数を作成し、値を2000とする。これにより、繰り返し間隔が2秒となる。

- これにより，出力が OFF となる.



必ず確認すること。

例題 8 の準備 マルチメータ VOAC-7520H の測定値取得


次に、マルチメータ VOAC-7520H を用いて、測定値の取得を行う。なお、WF1973 同様、GP-IB アドレスの変更が可能であるが、ここでは 9（工場出荷時設定）であるとして説明する。なお、VOAC-7520H は、電圧（交流・直流）、電流（直流・交流）、周波数、温度（熱電対使用）の測定が可能である他、独立した 2 系統の電圧測定が可能となっている。

動作確認

動作確認のため、ACV ボタンにより交流電圧入力モードとし、WF1973 の出力を入力。振幅および周波数などを適宜変更し、適切に測定が行われていることを確認する。

次に WF1973 同様、PC と VOAC-7520H とを接続し、GP-IB 接続状況を確認する。なお、WF1973 と VOAC-7520H の 2 台を GP-IB 接続する場合には、GP-IB ケーブルを使用し同時に接続する。





1) GP-IB 接続状態の確認

- ・ “NI MAX” (Measurement & Automation)  を実行
- ・ デバイスとインターフェイス > GPIB0 (“インターフェイス機器名”) を左クリック
- ・ 右側 Window 上部の「計測器をスキャン」を左クリック
- ・ 下側 Window に、IWATSU, VOAC7523H が現れることを確認し、これをダブルクリック
- ・ プライマリアドレスが 9 であることを確認

2) GP-IB による VOAC-7520H との通信

- ・ 右側 Window 上部の「計測器と通信する」を左クリック
- ・ 「文字列を送信」欄の “*IDN?” を削除し、:MAIN:DATA? と入力し「書き込み」後、「読み取り」をクリック。この時、VOAC-7520H に表示されている電圧が「受信した文字列」に表示されることを確認する。

※初期設定においては Remote I/F が None となっているため、これを GPIB と変更しなければ接続しても画面上に現れないので注意する。また、RS-232C で接続したい場合は、Remote I/F を RS232 に変更する。

※Remote I/F を変更したい場合は、SYSTEM ( + ) を選択した後、上下キー ( /  キー) でリモート制御 I/F を選択する。

以上の操作により PC との正常な接続が確認できた。以下に VOAC-7520H の主な GP-IB コマンドを示す。

VOAC-7520H の主な GP-IB コマンド

測定モード設定 :MAIN:FUNC Parameter

:MAIN:FUNC DCV	直流電圧測定	:MAIN:FUNC ACV	交流電圧測定
:MAIN:FUNC DCA	直流電流測定	:MAIN:FUNC ACA	交流電流測定
:MAIN:FUNC FREQ	周波数測定 (≤ 1 MHz)		

測定のオートレンジの ON/OFF 切り換え :MAIN:RANG:AUTO ON (or OFF)

測定レンジの設定 :MAIN:RANG:VAL Parameter

DCV

50E-3 (50 mV レンジ), 500E-3 (500 mV レンジ), 5E0 (5 V レンジ), 50E0 (50 V レンジ)
500E0 (500 V レンジ), 1E3 (1,000V レンジ)

※5 V レンジの設定：:MAIN:RANG:VAL 5E0

※測定電圧がある程度分かっているならば、レンジを設定していた方が、レンジを変化させているときの測定ミスが防げる。

ACV, DAV : 500E-3, 5E0, 50E0, 500E0, 750E0

DCA : 5E-3, 50E-3, 500E-3, 10E0

ACA, DAA : 5E-3, 50E-3, 500E-3, 10E0

測定値を出力 :MAIN:DATA?

その後、出力値を読み取るにより、最新の測定データが取得できる。

2 チャンネルを有効化 :DUAL:STAT ON

CH-B の測定モード設定 (CH-B 直流電圧測定) :SUB:FUNC BCH

CH-B 測定のオートレンジの ON/OFF 切り換え :SUB:RANG:AUTO ON (or OFF)






CH-B 測定値を出力 :SUB:MEAS? ※ 1 チャンネルも:MAIN:MEAS?を使用可, 2 チャンネル使用時は両チャンネルとも:MEAS?を使用することを推奨。





この他にも測定に応じて様々な設定を行うことはできるが、ここでは説明を省略する (詳細はマニュアル参照)。

[8] 例題 8：VOAC-7520H を用いた電圧値取得プログラムの例 **[結果を保存する]**

WF1973 によりピーク電圧 1[V] (開放終端時は 2 V) の正弦波を、100Hz ～ 100 kHz の範囲で変化(周波数 1 桁当たり対数軸等間隔に 10 分割:31 周波数)させ、周波数変化後 2 秒後に VOAC-7520H により交流電圧を測定するプログラムを作成する (測定値が安定するまで待機するため)。








また、周波数に対する測定電圧値のグラフ表示、更に周波数および測定値をテキストファイルとして保存する。(測定箇所は、実験手順を確認して結果が不足することがないようにすること。)

- 1) 例題 7 のプログラムを使用してプログラムの作成を行う。
- 2) VOAC-7520H の GP-IB アドレス入力用の数値制御器を配置
- 3) 「配列連結追加」のアイコン下側を下方方向にドラッグして 2 入力とし、WF1973 と VOAC-7520H の GP-IB アドレスを各々入力。
- 4) WF1973 の初期設定用「SEND (送信)」の後に、VOAC-7520H の初期設定用の「SEND (送信)」を追加
 - ・「エラー入力」、「エラー出力」を再配線
 - ・VOAC-7520H の GP-IB アドレスを「アドレス」に接続
 - ・「データ文字列」を右クリック>作成>定数 で文字定数作成
:MAIN:FUNC ACV
:MAIN:RANG:AUTO ON
を入力。これにより、VOAC-7520H は交流電圧測定モードとなり、連続測定を行う。
- 5) For ループの繰り返し回数 (N)=31 に設定
- 6) 周波数の計算 $i*0.2+2$ (周波数 1 桁当たり 5 ポイント) を、 $i*0.1+2$ に変更
- 7) For ループの繰り返しタイミング制御部 (10 秒待ち) を削除
- 8) 周波数変化から電圧測定までの待ち時間制御 (4 秒間)
 - ・待機周波数変化用「SEND」の後に、フラットシーケンスストラクチャ を配置
 - ・ストラクチャに SEND の「エラー出力」を通す。また、「待機」を
 - ・ストラクチャ内に「待機」を配置。待機時間を 4000 ms とする。
※4 秒にするのはレンジが変わった際の測定ミスを防ぐためである。
- 9) 1 秒待機後に「SEND (送信)」配置し、VOAC-7520H のアドレスを接続。データ文字列として :MAIN:DATA?を入力。
- 10) 計測器 I/O>GP-IB  > GP-IB 488.2  より Receive (受信)  を選択、配置。9)の後に接続。
 - ・VOAC-7520H のアドレスを接続
 - ・「カウント」端子に定数 (整数) を作成し 50 を入力 (読み取り最大文字数)
*最大文字数より大きい値を入力する必要がある。一方、この値により入力文字数を限定することもできる。
- 11) プログラミング>文字列  > スプレッドシート文字列を配列に変換  を配置
10)の「データ文字列」端子を「スプレッドシート文字列」と接続。
「形式文字列」端子に定数を作成し、“%s”を入力。
「デリミタ」端子に定数を作成し，“,” (半角カンマ) を入力。

プログラミング>配列  > 配列定数  を配置、プログラミング>文字列  > 文字列定数  を作成し，“,” (半角カンマ)を入力。この時、プロパティの外観(タブ内)のラベルに「デリミタ (タブ)」と入力しておく役割がわかりやすい。この文字列定数を要素

とするため、配列定数の配列シェル内にドラッグ&ドロップする。作成された配列定数を「配列タイプ」端子に接続。

※測定器から送られてくるデータは「DCV_ __, __, __, +0.71206E+0,3770」なカンマで区切られたデータであり、手前から[測定種別, エラー / 演算情報フィールド, 統計データ種別フィールド, 測定値, タイムスタンプフィールド]となっている。必要なのは4番目測定値であるのでまず11)では各項目を配列として格納する。

- 12) プログラミング>配列  > 部分配列  を配置
11)の「配列」端子を「配列」と接続。
「指標」端子に3を入力し、「長さ」端子に1を入力する。
※ここで、測定器から送られてくるデータから測定値だけを取り出す。
- 13) プログラミング>文字列  > 文字列／数値変換  > 小数／指数文字列を数値に変換  を配置
12)の「部分配列」端子を「文字列」と接続。
- 14) プログラミング>配列  > 指標配列  を配置
13)の「数値」端子を「n次元配列」と接続。
「指標0」端子に0を入力。
この時、「要素または部分配列」端子に測定結果（数値）が出力されるので、数値表示器作成。
- 12) 周波数、測定電圧（数値）を For ループから配列として出力（指標付け使用）
- 13) 周波数 – 電圧の関係をグラフ化（例題3参照）
フロントパネルでグラフを右クリック>プロパティにて、片対数化、軸の名称変更などを適宜行う。
- 14) 周波数、測定電圧をテキストファイル出力する（例題5参照）。
- 15) 各 GP-IB アドレスを確認し、動作確認（1回実行）。出力ファイルも確認する。

実験番号：2-I-1

実験タイトル：LabVIEWの基礎実験

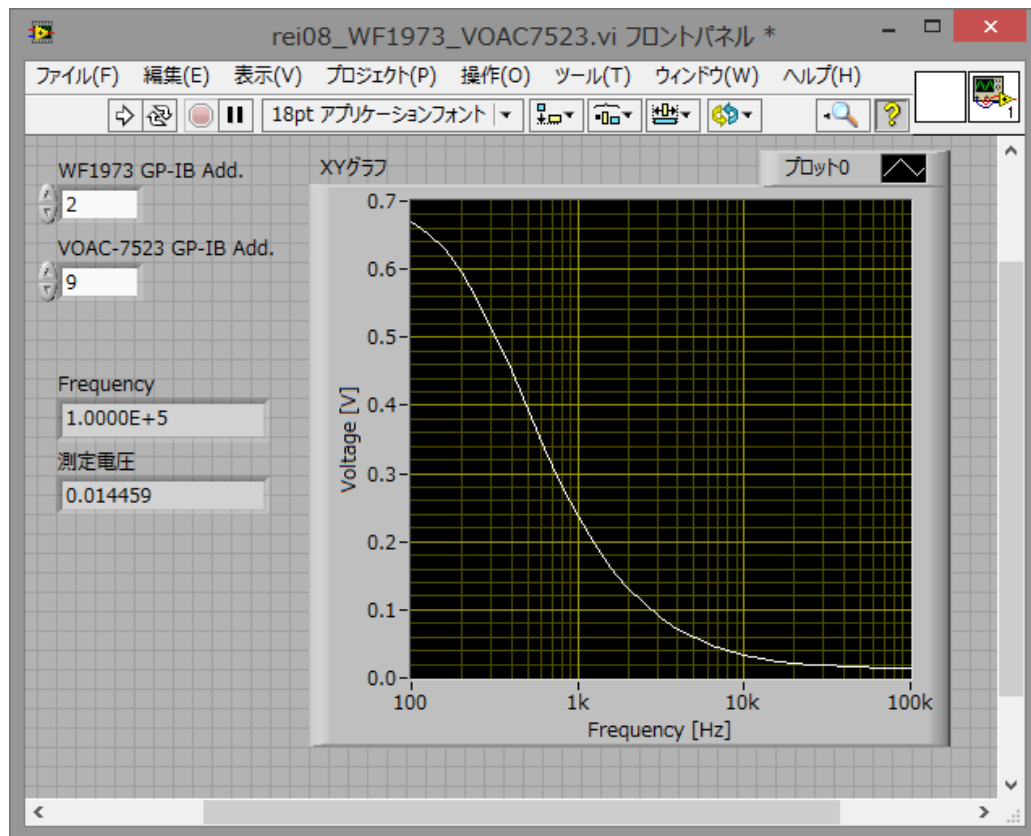


図 12 例題 8 のフロントパネル

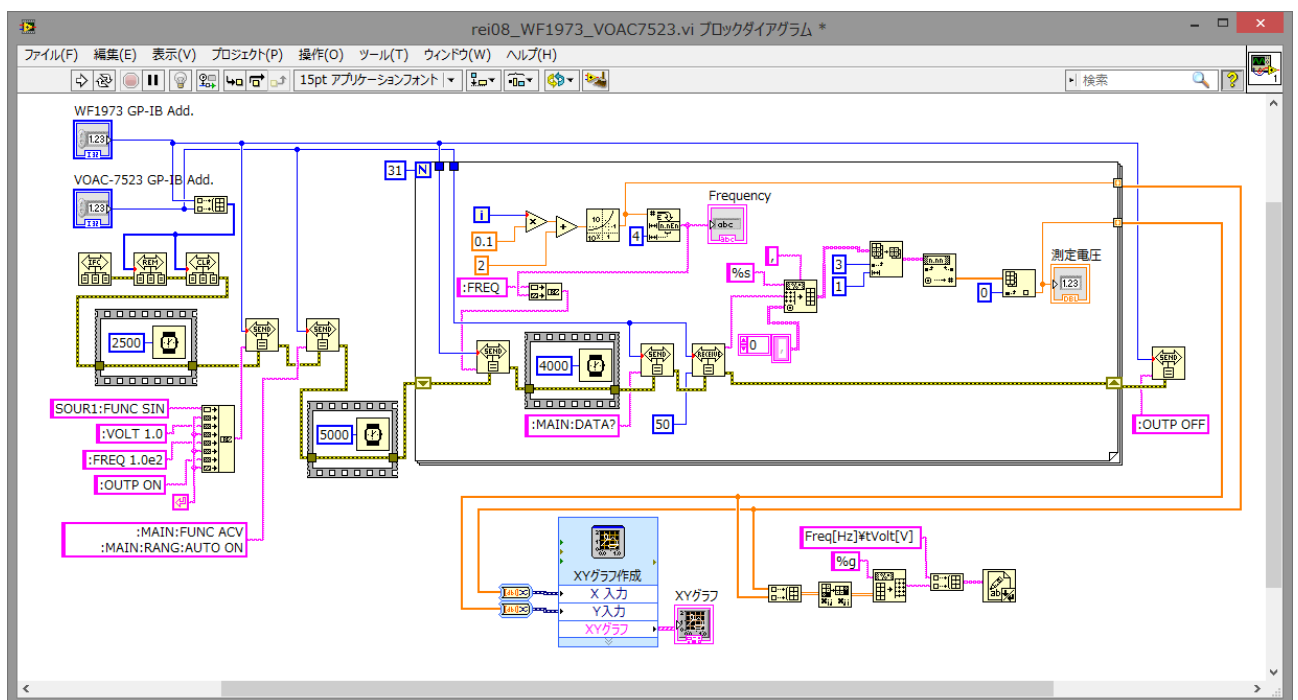


図 13 例題 8 のブロックダイアグラム