

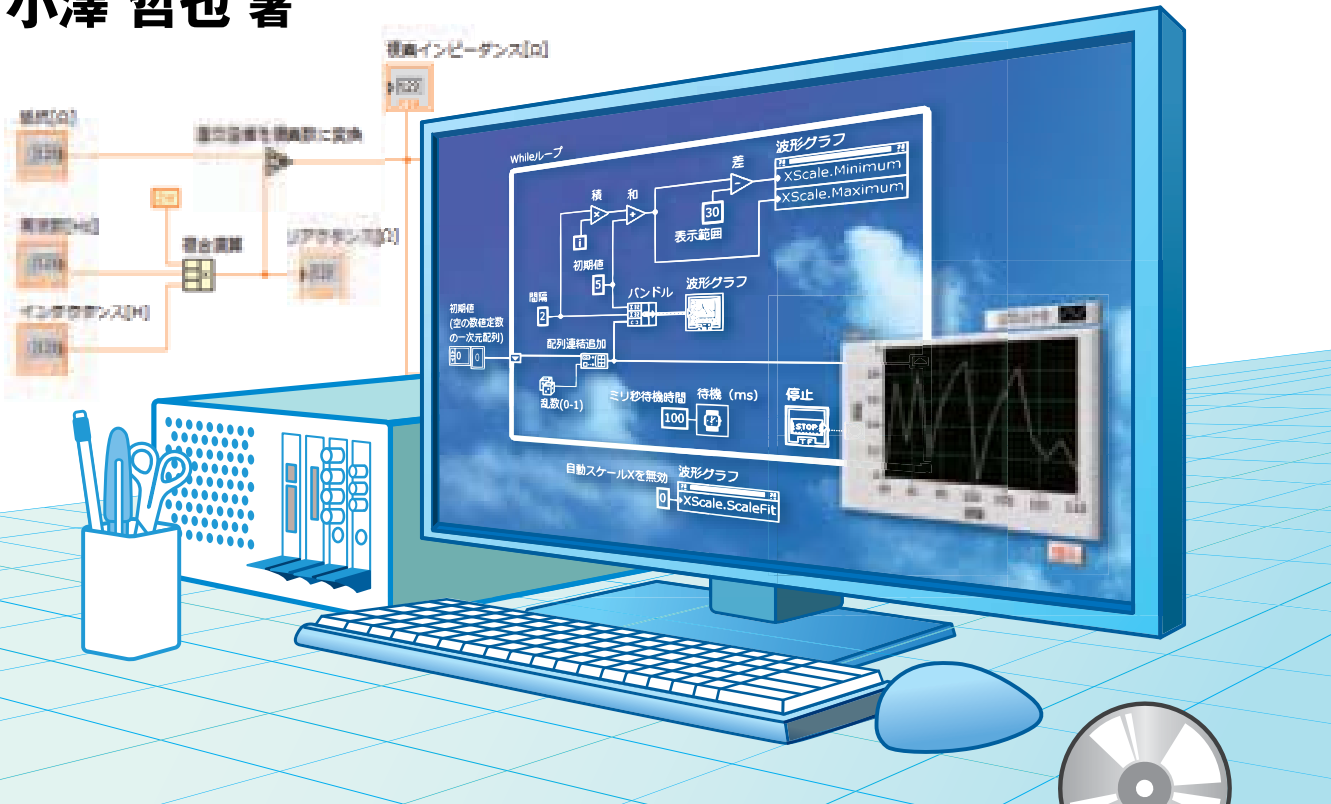
波形表示/データ保存の方法から命令や関数の使い方まで
パソコン計測制御ソフトウェア

LabVIEW

見本

リファレンス・ブック

小澤 哲也 著



CQ出版社



本書解説の全プログラム、章末問題の
全解答、講義用スライドPDFを収録



1-1 LabVIEWについて

LabVIEWとは、アメリカ合衆国テキサス州オースチン市に本社を持つナショナルインスツルメンツ社の計測制御用に特化したプログラミングソフトウェアであり、計測制御用ソフトウェアの一つとしてLabVIEWは広く知られています。

計測器と言えば、図1-1-1に示すような装置を思い浮かべることでしょう。このような計測器は、従来は手で操作して使用していましたが、研究開発や生産現場において、仕事の効率を上げたり特別な性能を引き出ししたりするためには、コンピュータ制御による計測器の自動化機能があると便利です。

LabVIEWには、この自動化のためのプログラミング機能があります。この機能に対応した計測デバイスと組み合わせることで、各自の目標に合わせた計測自動制御システムを構築することができます。

LabVIEWのプログラミング方法は、英語の文字を列挙する従来型のプログラミング方法とは大きく異なり、図1-1-2に示すように、必要な計算をワイヤで配線し、データを流すことで実現されます。そのため、従来型のプログラミング言語のように英語を書き込んでいく方法に苦手意識があるユーザーであっても、LabVIEWは開発しやすい環境であり、これが世界中に普及した理由とも言えます。LabVIEWが開発された経緯や用途に関するの詳細は、CQ出版社「バーチャル計測器LabVIEW」をご覧ください。

LabVIEWには、下記に記す三つのグレードが用意されています。

- ベース・パッケージ：計測器制御用の関数と四則演算などの比較的簡単なツールのみ
- 開発システム：上記に加え、フーリエ変換などの関数が使用可能
- プロフェッショナル開発システム：PID制御などの特殊な機能以外はすべて使用可能

なお、紹介した三つのグレードの内容は現時点のもので、最新の情報は、日本ナショナルインス



図1-1-1
計測器の例

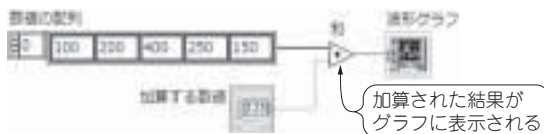
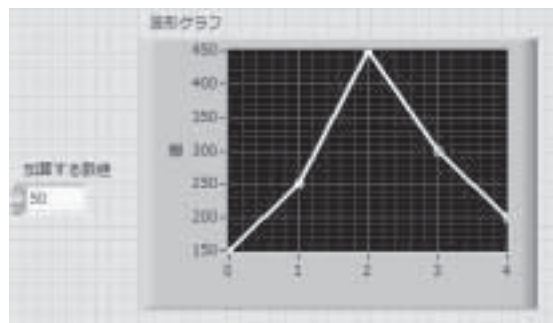


図1-1-2 LabVIEWプログラミングの例





ポイント1 筆者の研究現場

図1-1-3に示す装置は、PXIとよばれるWindows搭載の工業用コンピュータで計測自動制御されており、個々の機器はすべてマウスのクリックで動かすことができる状態になっています。マウスのクリックで自動的に動き出し、手動操作では何週間もかかるような作業を1時間程度で終わらせることができます。



図1-1-3
研究現場の計測自動
制御システムの例



ポイント2 LabVIEWで○○を計測できますか？

LabVIEWで○○を計測できますか？と問われることがあります。LabVIEWは、計測器に対してデータを取りなさいという命令を送ったり、得られたデータを解析したり、計測されたデータを保存するという機能に特化したプログラミング環境です。そのため、○○という現象を捕えることができる計測器があれば、○○を計測できるということになります。

LabVIEWは計測をする電子機器ではなく、計測制御用に適したプログラミングソフトウェアです。ソフトウェアを使うことで何が便利になるのかということ念頭に入れながら、計測自動制御システムの構築を計画するようにしてください。

ツルメンツのホームページ(<http://www.ni.com/jp>)で確認してください。

また、学生が自宅のコンピュータで学習するときに使用できる学生パッケージが安価で販売されています。本書を自宅で学習するには最適なパッケージです。詳しくは、(<http://www.ni.com/jp/academic>)をご覧ください。

※この学生パッケージは、研究などの業務目的での使用は禁止されています。

LabVIEWとは、どのようなソフトウェアなのか試してみたい場合は、日本ナショナルインスツルメンツ社のホームページから、評価版LabVIEWをダウンロードまたはDVD-ROM版として無償で入手することができます。使用期限は一か月間程度(LabVIEW2012の場合は最長45日間)に限られています。その間に本書を読みながら学習するには十分な期間です。次の節では、LabVIEWのインストール作業から説明を開始します。



1-4 制御器パレット

概要

LabVIEWでデータを入力したり、計算結果を表示したりするときは、制御器パレットから必要なオブジェクトを呼び出して使います。

課題

フロントパネル上で右クリックすると、図1-4-1に示すような制御器パレットが現れます。または、図1-4-2に示すようにメニューの表示から制御器パレットを選択しても、制御器パレットを呼び出せます。ユーザの設定によっては、最初から画面に見える設定にしてある場合もあります。

制御器パレット上でマウスを移動させると、いろいろなオブジェクトがあることがわかります。例えば、波形グラフは、図1-4-3に示すように「制御器パレット→Expressパレット→グラフ表示器パレット→波形グラフ」にあります。

また図1-4-4に示すように、制御器パレットの下端をクリックすると、制御器パレットが長く展開されて、すべてのオブジェクトを呼び出せるようになります。

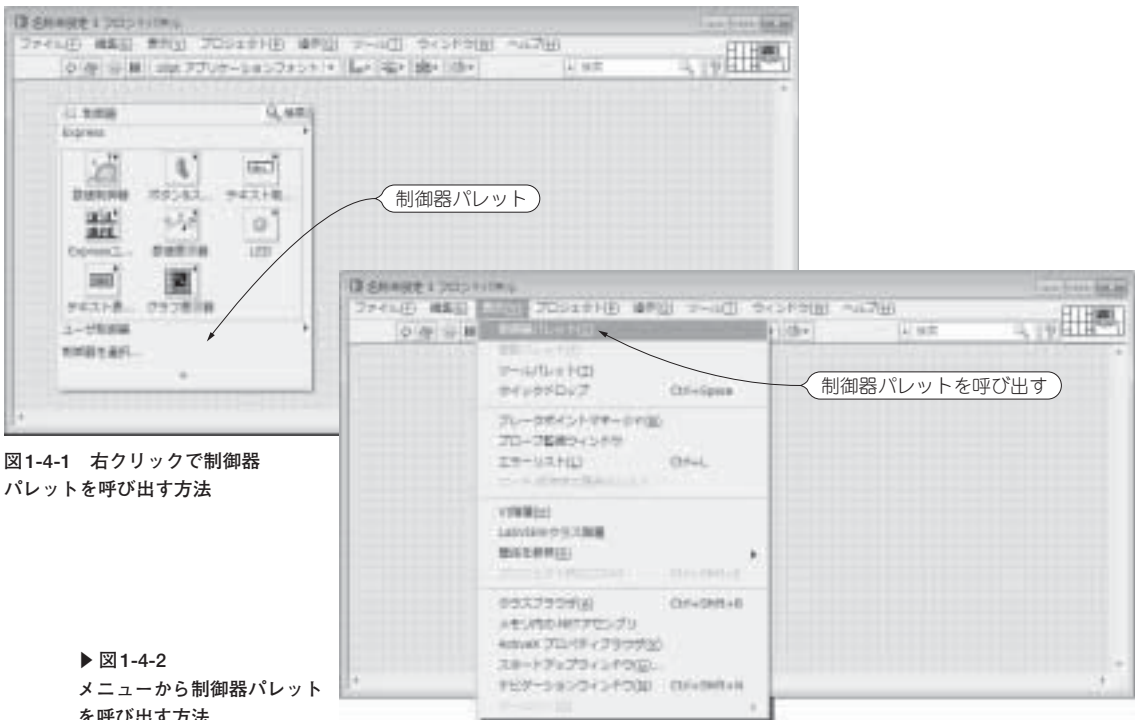


図1-4-1 右クリックで制御器パレットを呼び出す方法

▶ 図1-4-2
メニューから制御器パレットを呼び出す方法



ポイント4 パレットの項目の有無について

図1-4-4の展開した制御器パレットには、制御系設計&シミュレーションという項目が見えます。これはLabVIEWのバージョンやグレードによっては表示されない場合もあります。LabVIEWにはアドオンツールなどの追加機能があり、それらをインストールすると、制御器パレットから選べるオブジェクトの項目が増えるようになっていきます。

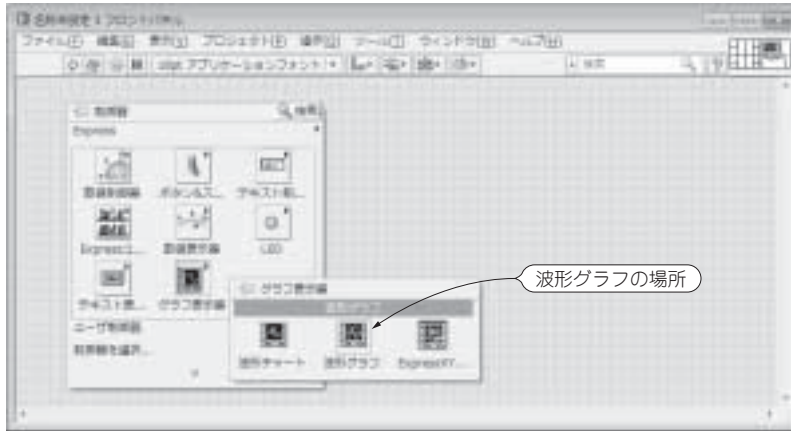


図1-4-3 波形グラフの場所

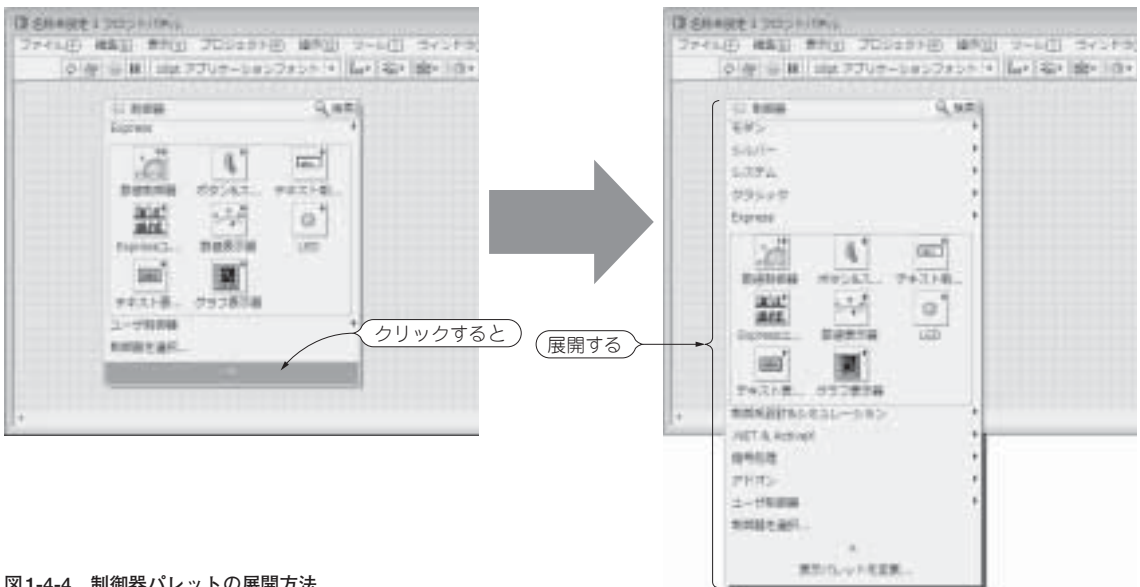


図1-4-4 制御器パレットの展開方法

概要

簡単なLabVIEWプログラムを作成し、制御器パレットや関数パレット、ツールパレットの使い方に慣れていきます。

課題

図1-8-1に示すようなフロントパネルとブロックダイアグラムを作成して、乱数を波形チャートに表示してみましょう。

ブロックダイアグラム上にあるサイコロマークの乱数関数は、図1-8-2に示すようにブロックダイアグラム上で右クリックして現れる関数パレットから「関数パレット→Expressパレット→演算&比較パレット→Express数値パレット→乱数」にあるので、マウス操作でブロックダイアグラム上に置いてください。

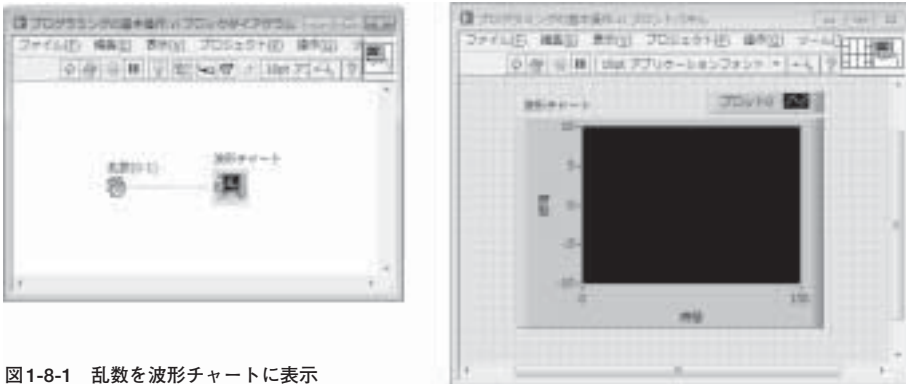


図1-8-1 乱数を波形チャートに表示

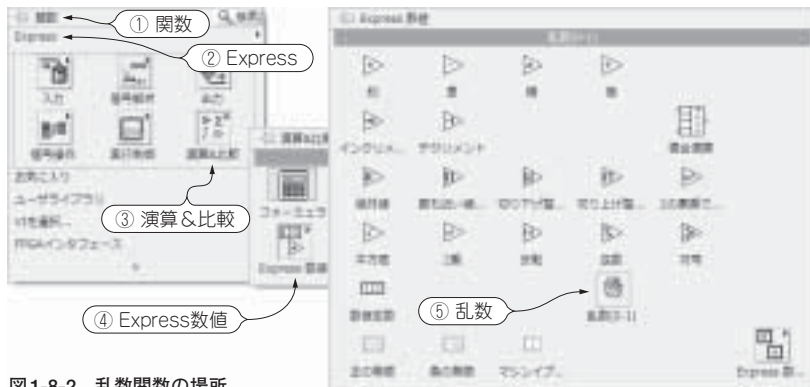


図1-8-2 乱数関数の場所



ポイント5 ラベルは端子上で右クリックして現れるメニューから作成・表示できる

なお、同じフロントパネル、またはブロックダイアグラム上に、同じ名前のラベルを付けてもプログラムの動作に支障はありませんが、プログラムを見直すときにわかりにくくなるので、できるだけ同じ名前を使用しないようにしましょう。

もし、操作方法を間違えてしまったときは、メニューバーの「編集→取り消し」を使用します。または、キーボードのCtrlキーを押しながらアルファベットのZキーを押しても同様です。

図1-8-1の乱数関数には「乱数 (0-1)」というラベルが付いています。ラベルを付ける場合には図1-8-3に示すように関数上にマウスを重ねて右クリックして現れるメニューから「表示→ラベル」を選択すればよいです。

フロントパネル上にある波形チャートは、図1-8-4に示すようにフロントパネル上で右クリックして現れる制御器パレットから「制御器パレット→Expressパレット→グラフ表示パレット→波形チャート」にあるので、マウス操作でフロントパネル上に置いてください。

フロントパネルに波形チャートをおくと、ブロックダイアグラムには図1-8-5に示すように波形チャートに対応した端子が現れます。この端子に乱数関数から発生した数値を渡すと、波形チャートに乱数が表示されることになります。

乱数関数と波形チャートの端子は、ツールパレットのワイヤリングツールまたは自動選択ツールを使用して、図1-8-5に示すようにマウス操作で配線してください。配線は、図1-8-1とまったく同じでなくとも、乱数関数と波形チャートの端子間が配線されていれば、問題がありません。

図1-8-1のプログラムが完成したら、忘れずにプログラムファイルを保存してください。プログラムファイルの保存方法はメニューバーの「ファイル→保存」を使用します。または、キーボードのCtrlキーを押しながらアルファベットのSキーを押しても同様です。例えば「図1-8-1.VI」などのように、自分でわかりやすい名前をつけて保存してください。実行方法は、次のページで説明します。



図1-8-3 ラベルの付け方

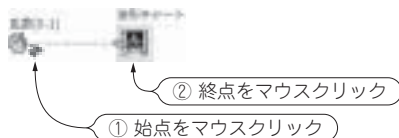


図1-8-5 マウスで配線する方法

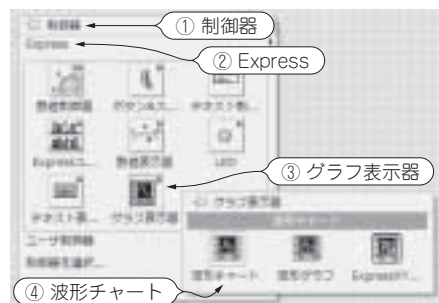


図1-8-4 波形チャートの場所

概要

ブールはTRUE (真) かFALSE (偽) のどちらかの値しかとらない単純な値ですが、これをプログラム内で使うためには論理演算の知識が必須です。ここではブールと論理演算について学びます。

課題

図2-9-1は、ブロックダイアグラム上で右クリックして現れる関数パレットの「関数パレット→Expressパレット→演算&比較パレット→Expressブールパレット」を示しており、LabVIEWで使用できる論理演算は8種類あります。LabVIEWで使用する論理演算は、基本的に**And関数**、**Or関数**、**Not関数**、**Not Or関数**の4種類です。これらについてまとめた論理演算の結果を図2-9-2に示します。**And関数**はすべての入力がTRUEならば出力がTRUEになります。**Or関数**は入力のいずれかがTRUEであれば出力がTRUEになります。**Not関数**は入力がTRUEであればFALSEを出力し、入力がFALSEであればTRUEを出力します。**Not Or関数**はOrの出力にNotを加えた動作であり、入力のいずれかがTRUEであれば出力がFALSEになります。

図2-9-3に示すようなブロックダイアグラムとフロントパネルを作成して、論理演算の取り扱いを学びます。新規にプログラムを作成するので、メニューバーの「ファイル→新規VI」を使用して、新規VI

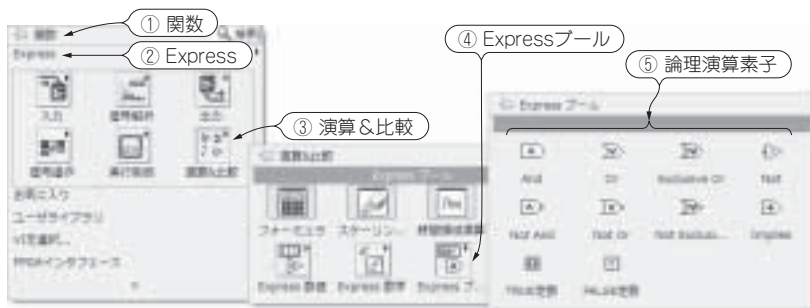


図2-9-1 Expressブールパレットと8種類の論理演算

入力A	入力B	Andの出力	Orの出力	Not Orの出力	入力C	Notの出力
FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE
FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE
TRUE	TRUE	TRUE	TRUE	FALSE		

図2-9-2 And関数・Or関数・Not関数・Not Or関数の論理演算の特性



ポイント16 ブールを普通の四則演算関数で計算するには？

ブールはTRUE(真)かFALSE(偽)のどちらかの値をとるため、このままでは四則演算できません。ブールから(0, 1)に変換関数を使用して、TRUEを数字の1, FALSEを数字の0へ変換すれば、四則演算ができます。ブールから(0, 1)に変換関数は、図2-9-5に示すように「関数パレット→プログラミングパレット→ブールパレット→ブールから(0, 1)に変換」にあります。



図2-9-5
ブールから(0,1)に変換関数の場所

を開いてください。

ブロックダイアグラム上のFALSE定数とTRUE定数とAnd関数とOr関数は、図2-9-1に示すようにブロックダイアグラム上で右クリックして現れる関数パレットから「関数パレット→Expressパレット→演算&比較パレット→Expressブールパレット」内にあります。

And関数およびOr関数につながっている表示器は、図2-9-4に示すように各関数の右半分側にマウスを重ね、右クリックして現れるメニューから「作成→表示器」を選択して作成できます。

完成したら、実行ボタンを押してみてください。左ページの論理演算の特性と同じように振る舞うことを確認してください。なお、ブロックダイアグラム中のTRUE定数は、指ツールでクリックするとFALSE定数に変更できます。同様に、FALSE定数はTRUE定数に変更できます。

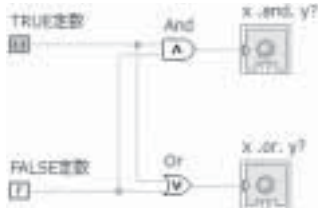


図2-9-3 ブールと論理演算

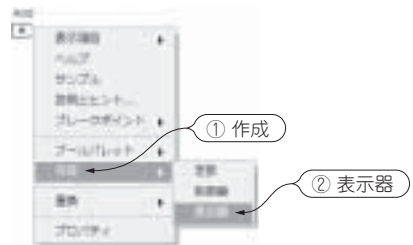
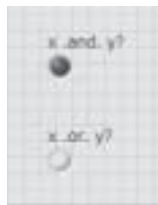


図2-9-4 関数の上で右クリックして表示器を作成

概要

これまででは一次元配列について説明してきました。ここでは次元数を増やして、表計算データのような二次元配列にする方法を学びます。

課題

一次元配列を二次元配列に変更する場合は、図3-3-1に示すように一次元配列を作成したあとに、配列指標上で右クリックして現れるメニューから「次元を追加」を選択すれば二次元にすることができます。配列の見かけ上の大きさは、矢印ツールで整えます。

LabVIEWでは、配列化されていない数値のことを明示的にスカラと呼びます。図3-3-2は、スカラである普通の数値をブロックダイアグラム内で一次元配列化させ、さらに二次元配列化させる方法を示しています。

図3-3-2に示すプログラムを作成してみてください。ここで使用されている配列連結追加関数は、図3-3-3に示すようにブロックダイアグラム上で右クリックして現れる関数パレットから「関数パレット

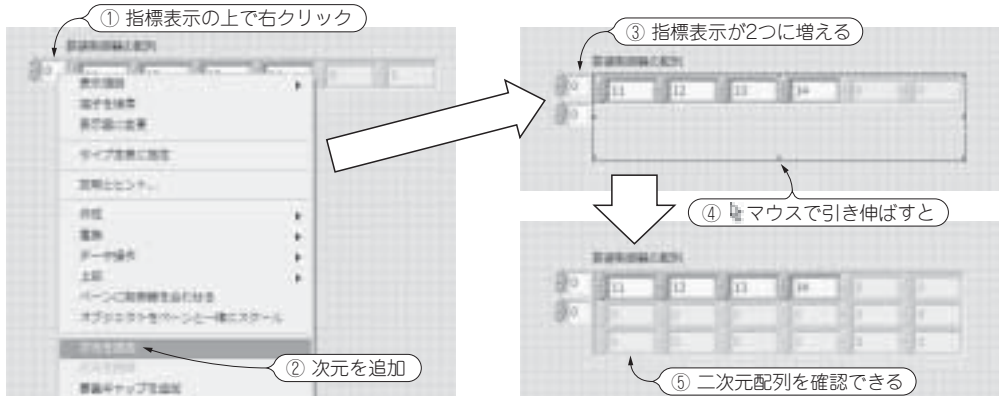


図3-3-1 配列を多次元化して、二次元配列を作成する方法



図3-3-2 配列連結追加関数による配列化と多次元化



ポイント19 さらに多次元化できますか？

配列指標上で右クリックして次元を追加する、または、**配列連結追加関数**を使用すれば、配列の次元数を三次元、四次元と増やすことが可能です。しかし、多次元化すると、配列内のどこに必要としているデータが収まっているのかが理解しにくくなるため、実際には一次元か二次元にとどめておいたほうが良いでしょう。特にプログラムを作成した本人が理解できても、それを利用する人には三次元以上に多次元化されたデータは理解しにくいものです。

→プログラミングパレット→配列パレット→配列連結追加」にあります。

図3-3-2に示す一次元配列と二次元配列のラベルがついている数値表示器の配列は、図3-3-4に示すように**配列連結追加関数**の右半分側にマウスを重ね、右クリックして現れるメニューから「作成→表示器」を選択して作成できます。

矢印ツールで、図3-3-2に示すようにフロントパネル上の配列の形を整えたら、実行してみましょう。図3-3-2は、ブロックダイアグラムの左側にある**配列連結追加関数**によって数値が一次元の配列になり、右側にある**配列連結追加関数**によって一元の配列データが二次元の配列に変換される動作となります。



図3-3-3 配列連結追加関数の場所

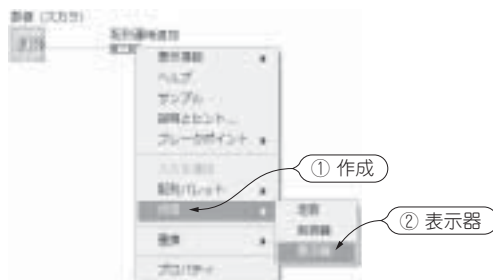


図3-3-4 配列連結追加から表示器を作成する方法

概要

これまででは、データファイルとして保存するプログラムを作成してきました。ここでは、保存したデータファイルをLabVIEWで読み取る方法について学びます。

課題

図6-5-1に示すのは、これまで保存してきたデータファイルをLabVIEWプログラムで読み取るプログラムです。以下の手順で、新規に図6-5-1のプログラムを作成してください。

ファイルパス制御器は、図6-5-2に示すように「制御器パレット→Expressパレット→テキスト制御器パレット→ファイルパス制御器」にあります。

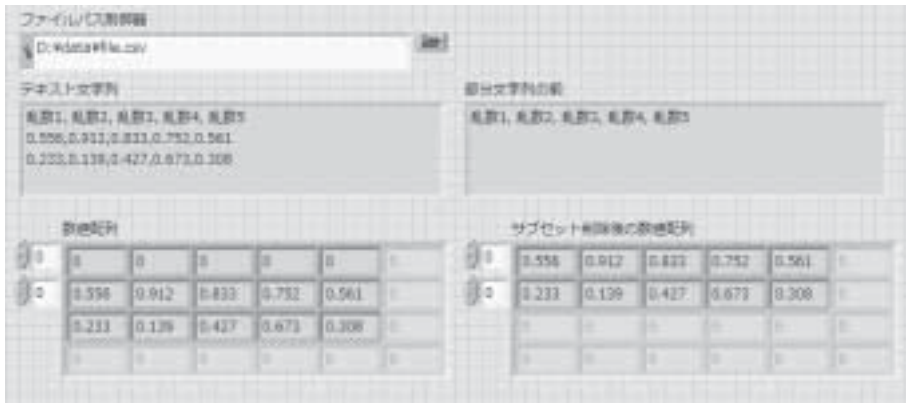


図6-5-1 データファイルを読み取る方法

テキストファイルから読み取る関数は、図6-5-3に示すように「関数パレット→プログラミングパレット→ファイルI/Oパレット→テキストファイルから読み取る」にあります。

テキストファイルから読み取る関数にワイヤで配線されているカウントというラベルが付いた数値定数は、「-1」の値を記入しておいてください。この「-1」という数値は、ファイルに含まれている文字をすべて読み込みなさいという動作を指定するものです。

パターンで一致関数および改行定数、デリミタというラベルが付いた文字列定数は、図6-5-4に示すように「関数パレット→プログラミングパレット→文字列パレット」内にあります。

Read From Spreadsheet File.vi関数は、図6-5-5に示すように「関数パレット→プログラミングパレット→ファイルI/Oパレット→スプレッドシートファイルから読み取る」にあります。

配列から削除関数は、「関数パレット→プログラミングパレット→配列パレット→配列から削除」にあります。

指標(行)というラベルが付いた数値定数は、配列から削除関数の入力端子上で右クリックして現れるメニューから「作成→定数」を選択して作成するか、「関数パレット→プログラミングパレット→数値パレット→数値定数」にある数値定数を使ってください。

テキスト文字列および部分文字列の前というラベルが付いた文字列表示器は、図6-5-6に示すように

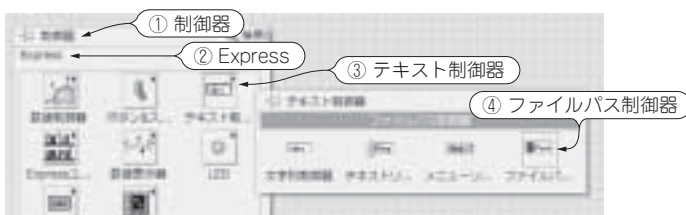


図6-5-2 ファイルパス制御器の場所



図6-5-3 テキストファイルから読み取る関数の場所

概要

5-2節のポイント38で述べたように、待機(ms)関数による時間間隔の制御は、絶対に保証されるものではありません。多少の遅れが生じます。その解決策として、ここではタイミンググループの使用方法を説明します(LabVIEW7以前には、タイミンググループ機能が備わっていない)。

課題

図7-5-1は、図5-2-1のプログラムを流用して、タイミンググループを適用したものです。次の順序にしたがって、図7-5-1のプログラムを作成してください。

図5-2-1のプログラムを流用して、図7-5-2のように、Whileループの中から待機(ms)関数を削除し、Whileループのフレーム上で右クリックして現れるメニューから「タイミンググループと置換」を選択してください。

タイミンググループに置き換えたら、図7-5-3に示すように、左側にあるdt端子の上で右クリックして現れるメニューから「作成→制御器」を選択して、dt端子に与える数値制御器(ラベルは周期)を作成し

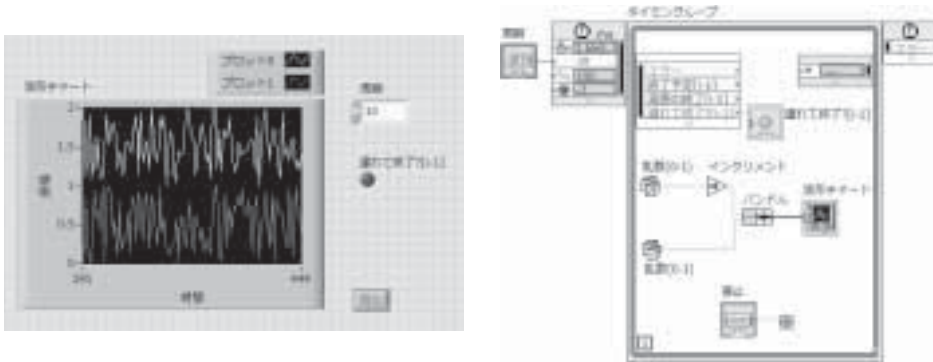


図7-5-1 タイミンググループを使用したプログラム

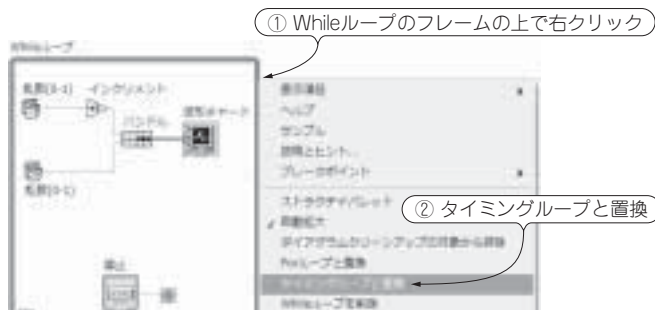


図7-5-2
タイミンググループ
と置換する方法

てください。このdt端子に入力する値がループを繰り返す時間間隔となります(単位はミリ秒)。

図7-5-4に示すように、タイミングループの内側にある端子をマウスで引き伸ばして、「遅れて終了?」の端子を増やしてください。次に、図7-5-5のように、「遅れて終了?」の端子の上で右クリックして「作成→表示器」を選択して、遅れて終了?の円LEDのブール表示器を作成してください。

プログラムが完成したら、周期というラベルがついている数値制御器に、1や1000などの値を入力して実行してください。周期を変えたときは、プログラムを再び実行してください。

ループ内の計算処理が、周期に指定した時間に間に合わないときは、遅れて終了?の円LEDが点灯します。よく確認できないときは、周期の設定を短くして、なにか他のソフトウェアを立ち上げるなどパソコンに負荷がかかる状態にすると、点灯しやすくなります。この方法によって、時間間隔に遅れが発生していないかどうかを調べることができます。

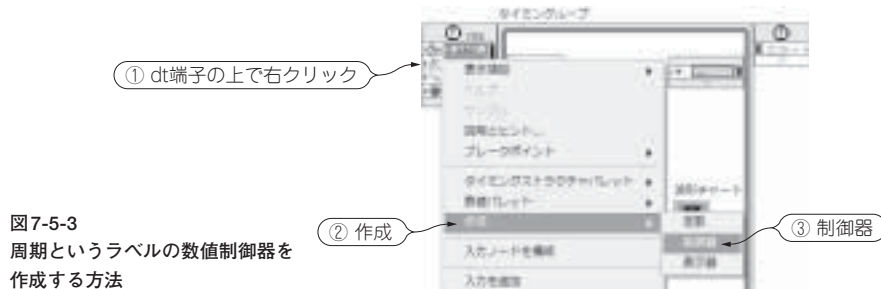


図7-5-3
周期というラベルの数値制御器を作成する方法

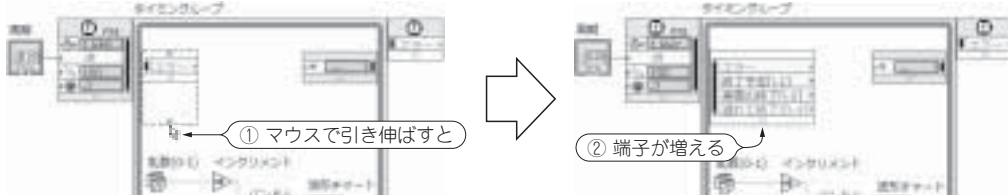


図7-5-4 周期というラベルの数値制御器を作成する方法

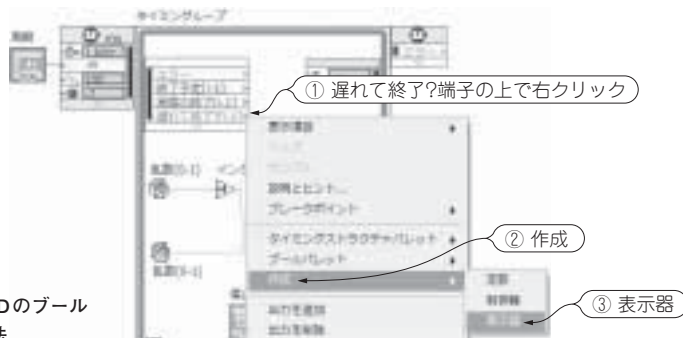


図7-5-5
遅れて終了?の円LEDのブール表示器を作成する方法

見本

ISBN978-4-7898-4095-8

C3055 ¥2600E

CQ出版社

定価：本体2,600円（税別）

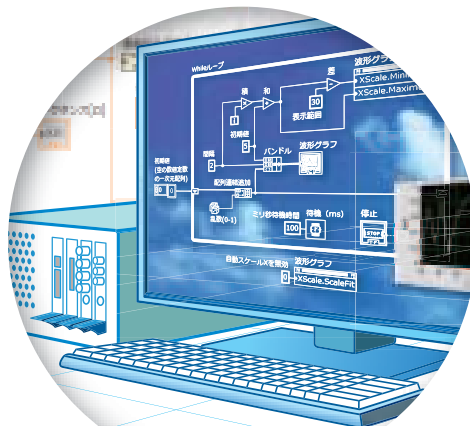


このPDFは、CQ出版社発売の「パソコン計測制御ソフトウェア LabVIEWリファレンス・ブック」の一部見本です。

内容・購入方法などにつきましては以下のホームページをご覧ください。

内容 <http://shop.cqpub.co.jp/hanbai/books/40/40951.htm>

購入方法 <http://www.cqpub.co.jp/order.htm>



パソコン計測制御ソフトウェア

LabVIEW
リファレンス・ブック