Содержание

Введение

1. Программный комплекс LabVIEW

2. Программирование, основанное на потоках данных

3. Генерирование детерминированных процессов в среде LabVIEW

4. Генерация детерминированных процессов

Заключение

Список литературы

Введение

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) позволяет разрабатывать прикладное программное обеспечение для организации взаимодействия с измерительной и управляющей аппаратурой, сбора, обработки и отображения информации и результатов расчетов, а также моделирования как отдельных объектов, так и автоматизированных систем в целом. Разработчиком LabVIEW является американская компания National Instruments.

В отличие от текстовых языков, таких как C, Pascal и др., где программы составляются в виде строк текста, в LabVIEW программы создаются в виде графических диаграмм, подобных обычным блок-схемам. Иногда можно создать приложение, вообще не прикасаясь к клавиатуре компьютера.

LabVIEW является открытой системой программирования и имеет встроенную поддержку всех применяемых в настоящее время программных интерфейсов, таких как Win32 DLL, COM.NET, DDE, сетевых протоколов на базе IP, DataSocket и др. В состав LabVIEW входят библиотеки управления различными аппаратными средствами и интерфейсами, такими как PCI, CompactPCI/PXI, VME, VXI, GPIB (КОП), PLC, VISA, системами технического зрения и др. Программные продукты, созданные с использованием LabVIEW, могут быть дополнены фрагментами, азработанными на традиционных языках программирования, например C/С++, Pascal, Basic, FORTRAN. И наоборот можно использовать модули, разработанные в LabVIEW в проектах, создаваемых в других системах программирования. Таким образом, LabVIEW позволяет разрабатывать практически любые приложения, взаимодействующие с любыми видами аппаратных средств, поддерживаемых операционной системой компьютера.

1. Программный комплекс LabVIEW

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) – это среда разработки и платформа для выполнения программ, созданных на графическом языке программирования «G» фирмы National Instruments (США). Первая версия LabVIEW была выпущена в 1986 году для Apple Macintosh, в настоящее существуют версии для UNIX, GNU/Linux, Mac OS и пр., а наиболее развитыми и популярными являются версии для Microsoft Windows.

LabVIEW используется в системах сбора и обработки данных, а также для управления техническими объектами и технологическими процессами. Идеологически LabVIEW очень близка к SCADA-системам, но в отличие от них в большей степени ориентирована на решение задач не столько в области АСУ ТП, сколько в области АСНИ.

2. Программирование, основанное на потоках данных

Графический язык программирования «G», используемый в LabVIEW, основан на архитектуре потоков данных. Последовательность выполнения операторов в таких языках определяется не порядком их следования (как в императивных языках программирования), а наличием данных на входах этих операторов. Операторы, не связанные по данным, выполняются параллельно в произвольном порядке.

В основе программирования в LabVIEW лежит понятие Виртуальных приборов (Virtual Instruments, VI). На лицевой панели, как и положено, располагаются элементы управления программой – кнопки, графики, выключатели и тому подобное. Блок-схема – это, по сути, и есть сама программа. При написании (а вернее создании, потому что писать приходится не так уж и много) программы используется такое понятие, как «поток данных» (Data Flow). Суть его в том, что все элементы программы (которые представлены графически) связываются между собой связями (проводами, нитками) по которым и происходит передача данных. В общем, описать это довольно сложно, лучше посмотреть на картинку, рис. 1.

Рис. 1. Простейший прибор.

Цифрами обозначены:

* Точки, элементы программы (Nodes)
* Терминалы индикаторов (Indicator Terminals)
* Связи (Wires)
* Терминалы управляющих элементов (Control Terminals)

Итак, в LabVIEW вы создаете пользовательский интерфейс (лицевую панель), с управляющими элементами и индикаторами. Управляющие элементы – это тумблеры, кнопки, поля ввода и прочие устройства ввода. Индикаторы – это графики, шкалы, лампочки, текстовые поля и тому подобное. После создания пользовательского интерфейса, вы добавляете программный код, который управляет объектами на лицевой панели. Этот код содержится в схеме (block diagram). Этот код чем-то напоминает собой блок-схему, хотя отличий много.

LabVIEW можно использовать для того, чтобы управлять различным оборудованием, таким, как, устройства сбора данных, различные датчики, устройства наблюдения, двигательные устройства (например, шаговые моторы) и тому подобное, а так же GPIB, PXI, VXI, RS-232 b RS-484 устройства. Также в LabVIEW имеются встроенные средства для подключения созданных программ к сети, используя LabVIEW Web Server и различные стандартные протоколы и средства, такие как TCP/IP и ActiveX.

Используя LabVIEW, можно создавать приложения для тестирования и измерений, сбора данных, управления различными внешними устройствами, генерации отчетов. Так же можно создать независимые исполняемые файлы и библиотеки функций, такие как DLL, так как LabVIEW – это полноценный 32-битный компилятор.

3. Генерирование детерминированных процессов в среде LabVIEW

Среда разработки виртуальных измерительных приборов LabVIEW (Laboratory Virtual Instrument Engineering Workbench) разработана фирмой National Instruments, лидером по производству контрольно-измерительной аппаратуры. Первая версия появилась в 1986 г. как результат поисков путей сокращения времени программирования измерительных приборов. Пакет быстро приобрел популярность и постоянно совершенствовался. В 2006 г. вышла его восьмая версия LabVIEW 8.2 и в 2007 г. – LabVIEW 8.5. Пакет LabVIEW содержит полный набор приборов для сбора, обработки (анализа), представления и хранения данных.

В среде LabVIEW создается так называемый виртуальный прибор (ВП) (Virtual Instrument – VI) – программный аналог реального физического прибора. Как и реальный прибор, ВП имеет лицевую панель, на которую выведены все органы управления и индикации. Проектирование ВП сводится к разработке лицевой панели и блок-схемы программы. Для этого используются два окна: окно Front Panel (FP) для размещения лицевой панели и окно Block Diagram (BD) для размещения блок-схемы программы (рис. 2).

Рис. 2

Оба окна имеют стандартную форму Windows. В верхней части каждого окна располагается полоса главного меню с названиями меню, как стандартными для Windows: File, Edit и Help, так и специфическими для LabVIEW: Operate, Tools, Browse, Window, содержание которых будет раскрываться по мере выполнения лабораторных работ. Ниже полосы главного меню размещена полоса инструментальной панели, служащая для запуска, отладки и редактирования ВП.

Вывод на лицевую панель всех элементов управления и индикации производится в активном окне FP с помощью палитры Controls (Управление) (рис. 3).

Палитра Controls вызывается щелчком правой кнопки мыши (ПКМ) в любом месте окна. Ее можно открыть также, выбрав Show Controls Palette в меню Window. Поиск нужного элемента осуществляется переходом в палитре на нижние уровни. Найденный элемент (точнее, его иконка) выделяется щелчком левой кнопки мыши (ЛКМ), переносится на FP и помещается на ней еще одним щелчком ЛКМ (можно и по другому: нажать ЛКМ – перенести – отпустить ЛКМ). После выбора элемента палитра исчезает, если ее не прикрепить, щелкнув ЛКМ по значку кнопки в левом верхнем углу палитры. Одновременно с выводом на лицевую панель элемента управления или индикации появляется соответствующий ему функциональный элемент в окне BD.

Рис. 3

Блок-схема программы ВП размещается в окне BD. Она составляется из функциональных элементов (узлов) с помощью палитры Functions (Функции). Палитра Functions вызывается в активном окне BD щелчком ПКМ. Поиск нужных узлов и размещение их в окне BD производится так же, как и для палитры Controls. Соединение узлов между собой осуществляется с помощью инструмента - “катушка”.

Инструменты в LabVIEW обладают очень удобным свойством – изменяться в зависимости от того, к чему они подводятся. Достаточно подвести указатель мыши, каким бы он ни был, к терминалу узла, как он превращается в “катушку”, а терминал начинает мерцать. Нажатием ЛКМ осуществляется подсоединение к этому выводу. Далее при нажатой кнопке проводится линия до той точки, с которой нужно соединить (она тоже должна мерцать при приближении “катушки”). Если линию нужно где-то зафиксировать, чтобы схема читалась лучше, то следует отпустить кнопку, снова нажать и продолжить соединение. Пока соединение не произведено, линия будет пунктирной. Если соединение произведено правильно, то линия после отпускания ЛКМ примет вид и цвет, соответствующий типу передаваемых данных. Для удаления линии связи нужно выделить ее инструментом – “стрелка” (щелчок ЛКМ для выделения прямолинейного участка, двойной щелчок ЛКМ для выделения всей линии) и нажать клавишу Delete.

При разработке ВП требуется многократно переходить от одного окна к другому. Это можно сделать разными способами. 1) Через пункт Window главного меню. Например, при активном окне FP в ниспадающем меню Window нужно выбрать Show Block Diagram. 2) Через нижнюю строку, если до этого окно было свернуто. 3) Двойным щелчком ЛКМ на элементе управления или индикации в любом окне. Такой способ удобен, особенно, если нужно определить расположение элемента управления или индикации в другом окне, – элемент, по которому щелкнули, будет выделен. Границы выделенного элемента мерцают. Выделение можно снять, щелкнув ЛКМ в любом свободном месте окна.

4. Генерация детерминированных процессов

В пакете LabVIEW возможны три способа генерирования детерминированных процессов:

- вычисление процесса по формуле,

- формирование процесса с использованием имеющихся в LabVIEW элементарных функций

- использование имеющихся в LabVIEW генераторов детерминированных процессов.

Рассмотрим эти способы подробнее.

## Вычисление детерминированного процесса по формуле. Для этого используется структура LabVIEW “Formula Node” (Формула). Внутри этой структуры записывается формула по обычным для языка программирования высшего уровня правилам. Формул может быть несколько, каждая из них должна заканчиваться знаком “;” (точка с запятой).

В данной лабораторной работе ограничимся генерированием отрезка синусоидального процесса длиной в k периодов. Непрерывный процесс записывается так:

Y = Asin(2πft + φ) = Asin(2πt/T + φ).

Перейдем к дискретному времени, подставив в формулу t = iTд, где Тд – интервал дискретизации. Тогда

Y = Asin(2πiTд/T + φ) = Asin(2πi/N + φ),

где N = T/Tд – количество отсчетов за период синусоиды.

Изменяя i от 0 до N - 1, получим N отсчетов синусоидального процесса за период синусоиды. Чтобы сгенерировать синусоидальный процесс длительностью k периодов, нужно увеличить верхний предел i до kN – При этом в k раз увеличивается количество отсчетов. Если нужно сохранить количество отсчетов на всю реализацию неизменным при изменении количества периодов k, то увеличивают интервал дискретизации. Для такого варианта

Y = Asin(2πki/N + φ).

Здесь N равно количеству отсчетов на всю реализацию синусоидального процесса, содержащего k периодов синусоиды.

Воспользуемся именно этой формулой. Так как размерность начальной фазы φ в этой формуле – радианы, а задавать фазу удобнее в градусах, то следует перейти от радианной меры к градусной.

Y = Asin(2πki/N + φ0π/180). ( 1)

Запись этой формулы в структуре Formula Node показана на рис. 4.

Рис. 4

Имена всех входных и выходных переменных вводятся соответственно во входные и выходные терминалы. Терминалы образуются щелчком ПКМ на границе структуры в предполагаемом месте терминала и выбором команд Add Input (Добавить вход) или Add Output (Добавить выход). В появившемся зачерненном прямоугольнике записывается имя переменной.

В структуре Formula Node рассчитывается только одно значение переменной Y при заданных входных переменных. Чтобы рассчитать все N значений при изменяющемся i, нужно использовать структуру For Loop (Цикл с фиксированным числом итераций). Она эквивалентна текстовому оператору: for i = 0 to N – 1 do На рис. 5 приведена схема генерации отрезка синусоиды со всеми элементами управления и индикации. Количество рассчитываемых значений равно количеству циклов N, которое задается подачей на терминал числа итераций (прямоугольник с буквой N в левом верхнем углу структуры) соответствующего числа. Текущее число завершенных итераций цикла содержится в терминале счетчика итераций (прямоугольник с буквой i внизу слева).

Рис. 5

В LabVIEW генерируемый процесс представляет собой массив чисел. Поэтому выходной терминал структуры For Loop по умолчанию находится в состоянии формирования массива Enable Indexing (Включить индексирование). При этом выходной терминал имеет вид . Если индексирование отключено (Disable Indexing), то на выходе будет последнее рассчитанное значение. Переключить состояние терминала можно, щелкнув по нему ПКМ и выбрав соответствующую команду. Сформированный массив подается на другие узлы только после завершения цикла.

Формирование процесса с использованием имеющихся в LabVIEW элементарных функций. Для формирования процесса используется та же формула ( 1), только для выполнения математических операций сложения, вычитания, умножения, деления, а также нелинейных операций используются функциональные узлы. Они находятся в подпалитре Numeric палитры Functions. Блок-схема программы, реализуемой таким образом, показана на рис. 6.

Рис. 6

Использование имеющихся в LabVIEW генераторов детерминированных процессов. В LabVIEW имеется большой набор ВП, генерирующих различные детерминированные процессы. Их можно разделить на три группы, различающиеся представлением выходного процесса и заданием его параметров. Это группы: Pattern (отрезок, образчик, шаблон), Wave (колебание, волна) и Waveform (сигнал, осциллограмма). Рассмотрим генераторы синусоидальных процессов.

ВП Sine Pattern (рис. 7) генерирует массив Sinusoidal Pattern, содержащий отрезок синусоидального сигнала. На рисунке приведены названия терминалов ВП и тип данных, подаваемых на терминалы.

Рис. 7

Значения массива рассчитываются по формуле (1). Массив задается следующими параметрами:

samples (выборки) – количество отсчетов на весь процесс (количество элементов в массиве) – N. Обозначение I32 означает, что это число целое, 32 бита;

amplitude (амплитуда) – амплитуда синусоиды – А. (DBL означает действительное число двойной точности);

phase (degrees) (фаза в градусах) – начальная фаза – φ0;

cycles (циклы) – количество периодов отрезка синусоиды – k.

ВП Sine Wave (рис. 8) генерирует массив sine wave, представляющий синусоидальное колебание. Значения массива рассчитываются по формуле

U = Asin(2πfi + φ0π/180).

Рис. 8

По сравнению с предыдущим ВП по другому задается частота и начальная фаза.

Frequency (частота) – выражается в относительных единицах как величина, обратная количеству отсчетов на период: f = k/N.

Reset phase (сбросить фазу) – логический вход, определяющий начальную фазу выходного процесса. По умолчанию на этом входе установлено состояние ИСТИНА. При этом начальная фаза синусоиды устанавливается равной значению на входе phase in (вход фазы). Если на входе Reset phase установить состояние ЛОЖЬ, то начальная фаза устанавливается равной значению фазы на выходе phase out (выход фазы) при последнем выполнении этого ВП.

На входы samples, amplitude – подается то же, что и в предыдущем ВП.

ВП Sine Waveform (рис. 9) генерирует кластер signal out, в котором вместе с массивом синусоидального сигнала содержится информация о частоте сигнала (временной шкале).

Выходной кластер задается следующими параметрами.

Offset (смещение) – постоянная составляющая сигнала.

Рис. 9

Reset signal (сбросить сигнал) – в состоянии ИСТИНА устанавливает начальную фазу синусоидального колебания равной значению на входе phase и сбрасывает отметку времени в 0. По умолчанию на этом входе установлено значение ЛОЖЬ.

Frequency (частота) – определяет частоту синусоидального колебания в Гц.

Amplitude – амплитуда.

Phase – определяет начальную фазу синусоидального колебания в градусах. По умолчанию значение равно 0.

Sampling info (информация о выборках) – кластер, содержащий следующую информацию о выборках: частота выборок Fs, равная числу выборок в секунду; по умолчанию значение равно 1000; число выборок в осциллограмме #s; по умолчанию число равно 1000.

Для визуального наблюдения генерируемых процессов используется ВП графический индикатор Graph. Если на его вход подается массив, то развертка осуществляется по элементам массива. Можно просмотреть несколько процессов одного типа, предварительно объединив их. Например, если они представлены массивами, то используется функция Build Array (объединить массивы). Объединять различные типы данных, например, массив и кластер , нельзя.

программирование данные генерирование массив

Заключение

LabVIEW обладает огромным арсеналом достоинств, таких как полноценный язык программирования, интуитивно понятный процесс графического программирования, широкие возможности сбора, обработки и анализа данных, управления приборами, генерации отчетов и обмена данных через сетевые интерфейсы.

Вместе с тем LabVIEW – очень простая и интуитивно понятная система. Неискушённый пользователь, не являясь программистом, за сравнительно короткое время (от нескольких минут до нескольких часов) способен создать сложную программу для сбора данных и управления объектами, обладающую красивым и удобным человеко-машинным интерфейсом. Например, средствами LabVIEW можно быстро превратить старый компьютер, снабжённый звуковой картой, в мощную измерительную лабораторию.

Специальный компонент LabVIEW – Application Builder, позволяет выполнять LabVIEW-программы на тех компьютерах, на которых не установлена полная среда разработки.

Список литературы

1. Н.А. Виноградова, Я.И. Листратов, Е.В. Свиридов. «Разработка прикладного программного обеспечения в среде LabVIEW». Учебное пособие – М.: Издательство МЭИ, 2005.
2. http://www.automationlabs.ru/
3. http://digital.ni.com/
4. http://www.labview.ru/
5. http://ru.wikipedia.org/