



Физический факультет МГУ

Кафедра общей физики

Практикум

Введение в технику эксперимента

**Физические измерения в системе
LabVIEW**

МОСКВА 2011

Оглавление

Цель задачи	2
1. Что такое LabVIEW	2
2. Виртуальные приборы в LabVIEW	3
3. Измерительный DAQ-модуль	5
Выполнение задачи	9
Упражнение 1. Первоначальное знакомство с техникой работы в среде LabVIEW	10
Упражнение 2. Виртуальный вольтметр	12
Упражнение 3. Виртуальный омметр	18
Упражнение 4. Программирование расчета случайных погрешностей вольтметра	23
Упражнение 5. Программирование расчета случайных и приборных погрешностей омметра	25
Примерные вопросы к допуску	26
Приложение. Элементы программирования в системе LabVIEW	26
Числовые функции.....	27
Узел Формулы.....	27
Преобразование типов данных.....	28
Литература	28

Физические измерения в системе LabVIEW

Цель задачи.

Создать измерительные приборы: вольтметр, амперметр и омметр на базе персонального компьютера. Провести измерения напряжения, силы тока и сопротивления, выполнить статистическую обработку результатов измерений.

Познакомиться с системой программирования LabVIEW и принципом работы измерительного аналого-цифрового преобразователя (АЦП).

1. Что такое LabVIEW

В настоящее время персональный компьютер является обязательной принадлежностью практически каждой физической лаборатории. Он широко используется как средство обработки и накопления результатов измерений, а также для управления экспериментальными установками в реальном времени. Кроме этого, с добавлением специальной внутренней платы или внешнего модуля сбора данных, сам компьютер можно превратить в многофункциональный измерительный прибор, способный во многих случаях заменить традиционные приборы – вольтметры, амперметры, осциллографы, генераторы сигналов и т.д.

Для программирования таких устройств можно использовать универсальные языки программирования: C, Pascal и другие, но это довольно трудоемкая задача и требует высокой квалификации пользователя. Чтобы облегчить программирование измерительных устройств, были созданы специализированные системы, основанные на принципах визуального графического программирования, содержащие большой набор готовых программных модулей, для упрощенного программирования многих операций, специфичных для задач автоматизации научных измерений. Система по умолчанию сама выполняет многие функции, обычно лежащие на программисте, освобождая пользователя от вникания в низкоуровневые детали работы программы.

Одной из распространенных систем такого рода является LabVIEW (Laboratory Virtual Instruments Engineering Workbench), разработанная фирмой National Instruments (США). Эта система широко используется в учебном процессе во многих ведущих университетах мира (в области автоматизации экспериментальных установок, в практикумах, для моделирования, создания компьютерных демонстраций и т.д.). В фундаментальной науке она используется в передовых научных центрах, например CERN, Lawrence Livermore, Oak ridge и многих других, а также в высокотехнологичных научно-промышленных корпорациях (например, NASA, Jet

propulsion Laboratory). Среду LabVIEW используют при разработке приложений для измерения и сбора данных, управления измерительными приборами, анализа данных измерений и составления отчетов.

В целом, LabVIEW является гибкой системой, использующей современные технологии программирования, например, ActiveX и др. Программы LabVIEW имеют также возможность подключения внешних динамических библиотек (DLL), а виртуальные приборы LabVIEW можно скомпилировать в виде DLL, доступных другим приложениям, созданным в других языках программирования. LabVIEW поддерживает сетевые протоколы TCP/IP и UDP, что обеспечивает дистанционный доступ к виртуальным приборам через локальную сеть и Интернет.

Данная задача не предполагает систематического изучения системы и языка программирования LabVIEW и основывается на самых минимальных сведениях, необходимых только для выполнения данной конкретной задачи. В целом, язык LabVIEW обладает богатыми возможностями, сравнимыми с универсальными языками программирования, и его изучение требует серьезного учебного курса. А типовые виртуальные измерительные приборы система LabVIEW позволяет создавать достаточно просто.

Программа LabVIEW не имеет русификации. Работа с ней требует некоторого минимального знания компьютерного английского языка, чтобы понимать меню, диалоги, контекстные подсказки и информационные материалы (*help*). Дополнительная литература по LabVIEW на русском языке для желающих глубже изучить систему, указана в конце описания. Для подготовки к работе достаточно данного описания.

2. Виртуальные приборы в LabVIEW.

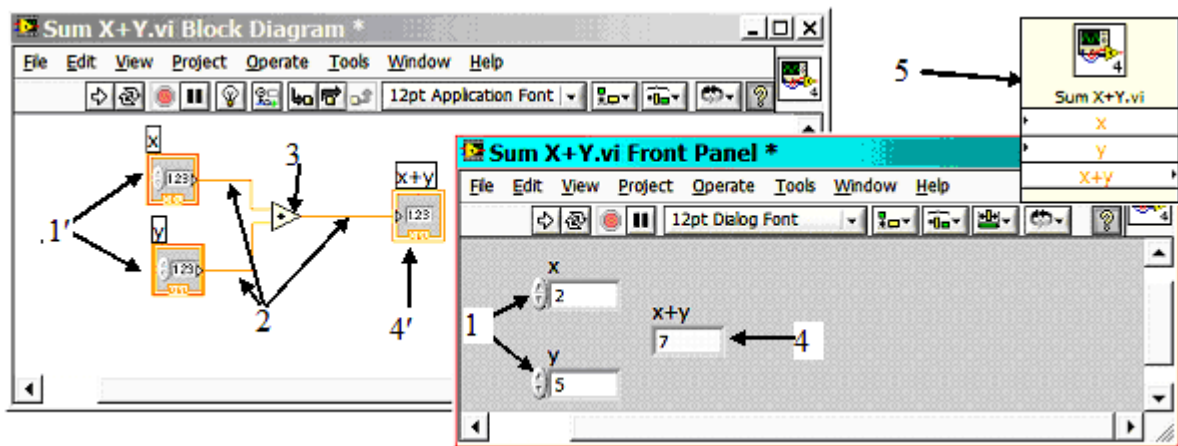
Программу LabVIEW принято называть **Виртуальным прибором (ВП) /Virtual Instrument (VI)**, поскольку в измерительных системах она является программным аналогом измерительного прибора (реально существующего или специально созданного). На LabVIEW можно писать и другие программы, не связанные с экспериментом, например, для решения дифференциальных уравнений, интегрирования и т.д., но математические программы LabVIEW больше ориентированы на обработку потока экспериментальных данных, в том числе в режиме реального времени.

Файлы ВП имеют расширение **vi**. Каждый ВП состоит из двух неразделимых взаимосвязанных частей (см. *рис. 1*).

1. **Лицевая Панель / Front Panel**, является интерфейсом пользователя, т.е. содержит средства ввода с экрана монитора исходных данных и управления программой в ходе ее выполнения – **Элементы управления/Controls**, и средства вывода информации – **Индикаторы /Indicators** – цифровые и

графические. Для удобства пользования виртуальным прибором в LabVIEW предусмотрен большой набор элементов управления, имитирующих привычные элементы управления реальных приборов – переключатели, тумблеры, сигнальные светодиоды, шкалы со стрелками и т.д. При желании лицевую панель ВП можно сделать в виде точной копии передней панели реального прибора – вольтметра, осциллографа, генератора сигналов и т.д..

2. **Блок-схема /Block diagram** в графической форме описывает алгоритм работы ВП. Эта часть содержит **терминалы** управления / индикации и функциональные **узлы**, входные и выходные поля которых соединяются между собой проводниками передачи данных (рис.1).



- 1 Элементы управления /controls (ввод чисел) 2 Проводники данных 3 Функциональный узел 4 Индикатор цифровой
1' Иконки терминалов вводимых данных 4' Иконка индикатора

Рис.1 Лицевая панель, блок-схема и представление в виде узла (5) для субВП, выполняющего сложение двух вводимых чисел x и y

LabVIEW имеет иерархическую структуру ВП. Созданный виртуальный прибор можно оформить в качестве подпрограммы (**SubVI / СубВП**), при этом он получает вид иконки с полями ввода/вывода. Количество уровней в иерархии не ограничено.

Иконки терминалов управления и индикации на блок-схеме являются "отражениями" соответствующих элементов управления и индикации лицевой панели. Терминалы – это порты ввода / вывода, осуществляющие обмен данными между лицевой панелью и блок-схемой. Цвет рамки терминала отражает тип данных, которые он принимает.

Узлы – это объекты на блок-схеме, которые имеют одно или более полей ввода/вывода данных и выполняют алгоритмические операции. Они аналогичны операторам, функциям и подпрограммам текстовых языков программирования. Узлы подразделяются на встроенные функциональные узлы и ВП, входящие в дистрибутив программного пакета LabVIEW, а также собственные СубВП пользователя, в том числе формульные структуры, содержащие текст программ на языках высокого уровня, например, С.

Данные между функциональными узлами передаются при помощи проводников и изображаются разноцветными линиями различной толщины. Тип линии однозначно соответствует типу данных (см. приложение: Элементы программирования в системе LabVIEW).

Программирование в языке LabVIEW в основном сводится к размещению на рабочем поле иконок нужных узлов и ВП и соединению их полей входа / выхода проводниками данных. Все функциональные узлы и ВП, доступные LabVIEW, вызываются с помощью иерархической системы меню-палитр. Для вызова корневого меню **Functions** достаточно щелкнуть правой кнопкой мыши по любому пустому месту на поле **Блок-схемы**. Например, узлы **Numeric**, осуществляющие основные математические операции, находясь на палитре **Functions >> Programming** (Рис. 6). Там же имеются терминалы ввода констант и основные математические константы (числа π , e и др.).

3. Измерительный DAQ-модуль.

Система LabVIEW может работать с автономными измерительными приборами, имеющими цифровой выход для сопряжения с компьютером. Связь компьютера с приборами может осуществляться через специальную плату контроллера канала общего пользования (КОП или GPIB), или же через стандартный параллельный или последовательный порт компьютера. Практически для каждого современного измерительного прибора имеются специализированные программы-драйверы для подключения к LabVIEW. Кроме этого, для LabVIEW созданы многочисленные собственные измерительные устройства сбора данных, как внутренние, устанавливаемые в системный блок компьютера в шину PCI, так и внешние, подключаемые через шину USB. Устройства сбора данных называются **DAQ**-модулями (сокращение от *Data Acquisition* = получение данных).

Принцип работы АЦП DAQ-модуля.

Основным элементом DAQ-модуля является аналого-цифровой преобразователь (АЦП), который преобразует величину аналогового сигнала – напряжения – в цифровой двоичный код. (Обратное преобразование циф-

рового кода в аналоговое напряжение осуществляет цифро-аналоговый преобразователь – ЦАП).

Существует несколько основных типов АЦП. В данном модуле используется **АЦП последовательного приближения**. Принцип его работы напоминает метод нахождения корней уравнения делением отрезка пополам. Сначала вырабатывается напряжение, лежащее посередине всего выбранного диапазона измерений (*рис.2*), и сравнивается с измеряемым напряжением U_x с помощью специального элемента сравнения – компаратора (блок-схема АЦП приведена на *рис.3*). Получается один двоичный разряд результата, который записывается в специальную область памяти - регистр последовательного приближения. Затем вырабатывается напряжение, лежащее посередине найденного интервала и опять сравнивается с U_x и т.д. Весь цикл измерения занимает N тактов вместо $2^N - 1$, которые бы потребовались при простом последовательном переборе всех значений. Напряжение для сравнения вырабатывается с помощью ЦАП, который управляется регистром последовательного приближения. После N тактов из этого регистра и забирается результат измерения (N бит) в двоичном коде.

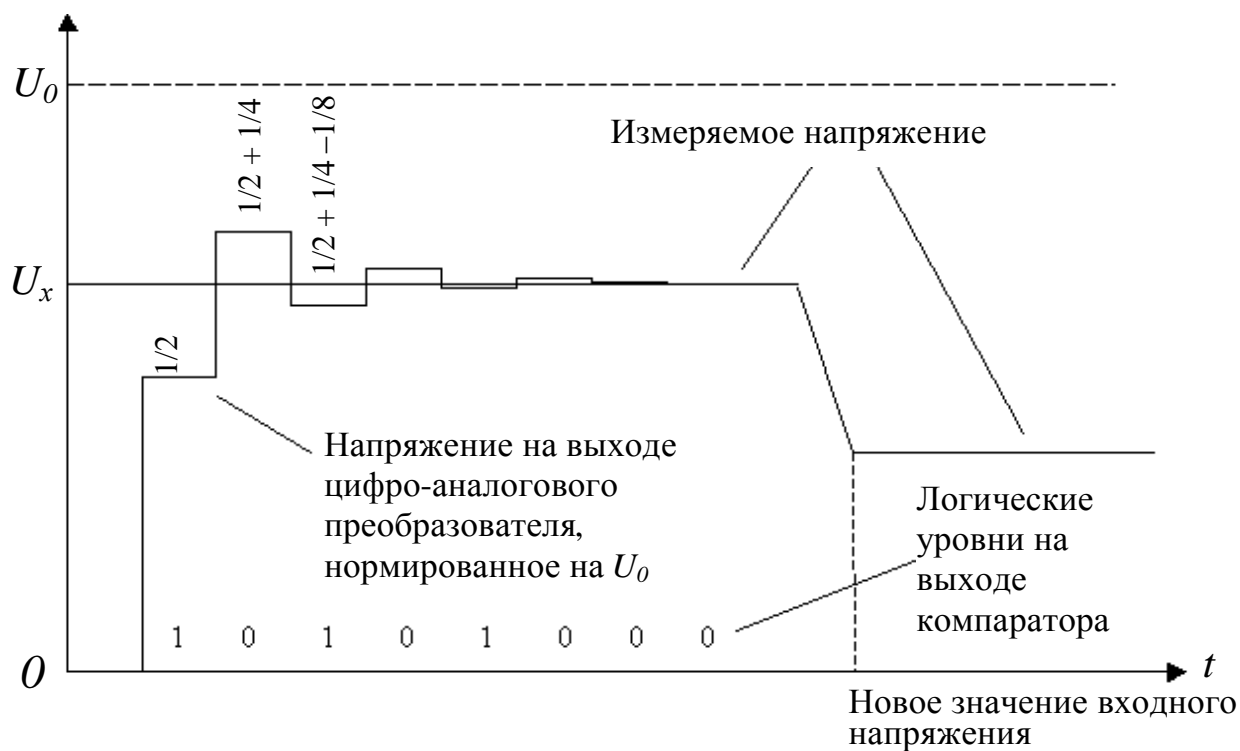


Рис.2. Временные диаграммы напряжений на входах компаратора АЦП последовательного приближения

В примере, приведенном на *рис.2*, на первом такте измеряемое напряжение U_x превышает $1/2 U_0$ и на выходе компаратора появится логическая еди-

ница. Эта единица запишется в старший разряд регистра. На втором такте напряжение на выходе ЦАП увеличится на $1/4 U_0$ и станет $3/4 U_0$. При втором измерении напряжение на выходе ЦАП превысило напряжение U_x , поэтому на выходе компаратора появится логический нуль. Этот нуль запишется во второй разряд регистра последовательного приближения, и напряжение на выходе ЦАП уменьшится на $1/8 U_0$ от предыдущего значения, став $5/8 U_0$. После N тактовых импульсов мы получим полный N -разрядный двоичный код, соответствующий входному напряжению.

АЦП последовательного приближения могут работать как в режиме однократного преобразования, так и в режиме создания непрерывного потока данных. На *рис.3* показан АЦП в режиме непрерывного преобразования, когда сигнал готовности результата запускает следующий цикл.

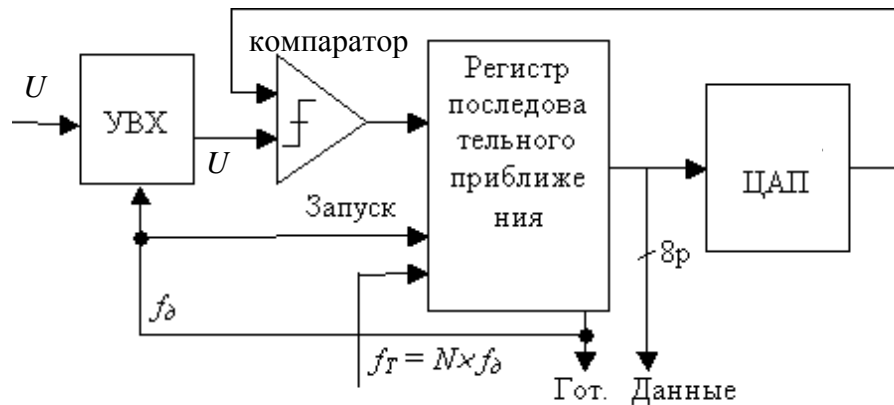


Рис.3 Структурная схема АЦП последовательного приближения.

- f_r – частота генератора для запуска тактов измерений
- Гот. – сигнал готовности результата и начала нового цикла
- Данные – N -битовое двоичное число с результатом измерения
- УВХ – блок управления входа

Основными параметрами АЦП является количество двоичных разрядов в результате – его разрядность, и максимальная частота измерений – частота дискретизации.

Используемый в данной задаче АЦП в модуле **USB 6009** имеет разрядность $N = 14$ бит, то есть весь измеряемый диапазон напряжений делится на $2^{14} = 16384$ ступенек. Таким образом, на диапазоне ± 10 В максимальная теоретическая точность составляет около 1.2 мВ. Реальная точность данного модуля, приведенная в таблице 1 (упражнение 2, стр. 17), составляет на этом диапазоне 7.73 мВ и определяется другими факторами, например, стабильностью опорного напряжения.

Частота дискретизации. Аналоговый сигнал является непрерывной функцией времени, в АЦП он преобразуется в последовательность цифровых значений. Частотой дискретизации f_d называется количество измерений, производимых за одну секунду. Для данного модуля она составляет $f_d = 48$ кГц. Если же нужно промерить форму сигнала (получить осциллограмму) хотя бы по 10 точкам на период, то максимальная частота для измеряемой синусоиды будет всего 4.8 кГц.

Существенно расширить частотный диапазон при измерении строго периодических сигналов можно с помощью стробоскопического метода. Поскольку через время, кратное периоду T , сигнал полностью повторяется, можно накопить измерения по N периодам, смещая каждый раз время измерений от начала периода на интервал $\Delta t = T/N$ и запоминая полученные значения. При этом время получения осциллограммы соответственно возрастает и составит NT .

В настоящее время созданы АЦП с гораздо более высокой частотой дискретизации вплоть до 1.5 – 2 ГГц. По этому параметру современные АЦП догнали аналоговые электроннолучевые осциллографы, которые обычно регистрируют осциллограммы сигналов до 200-500 МГц. А стробоскопические цифровые осциллографы позволяют наблюдать периодические сигналы вплоть до 100 ГГц.

В модуле **USB 6009** имеется только один АЦП, который по очереди обслуживает 8 каналов ввода, поэтому, чем больше каналов подключено, тем ниже скорость на каждом. Например, при подключении двух каналов, как в упражнении 3, максимальная частота дискретизации в каждом канале будет уже не 48 кГц, а 24 кГц.

Выполнение задачи

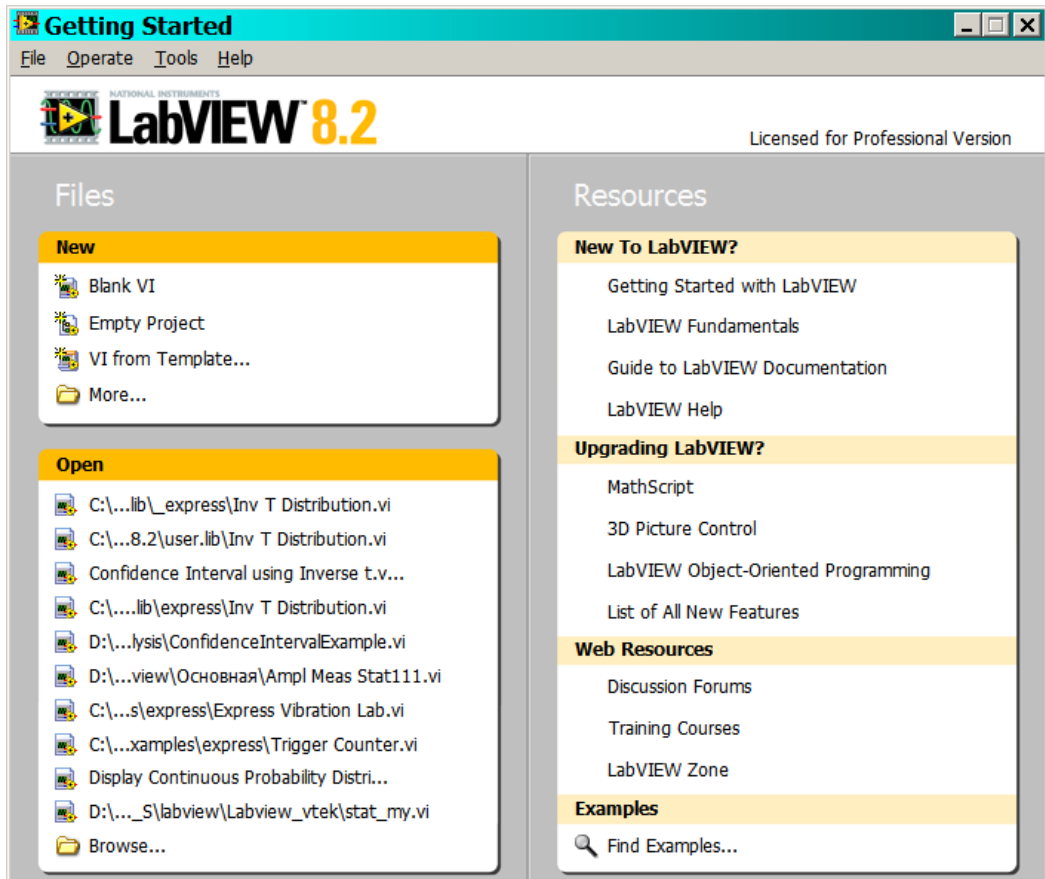



Рис.4 Стартовое окно программы

Программа запускается мышью через ярлык LabVIEW , который находится на рабочем поле Windows, а также в меню быстрого запуска в левой нижней части экрана. На экране появится стартовое окно (рис.4).

Чтобы начать новую задачу, нужно нажать мышью строку Blank VI (пустой ВП), расположенную в области New. На экране появятся два пустых окна, одно из которых – лицевая панель виртуального прибора, а другое – его блок-схема (Рис.5).

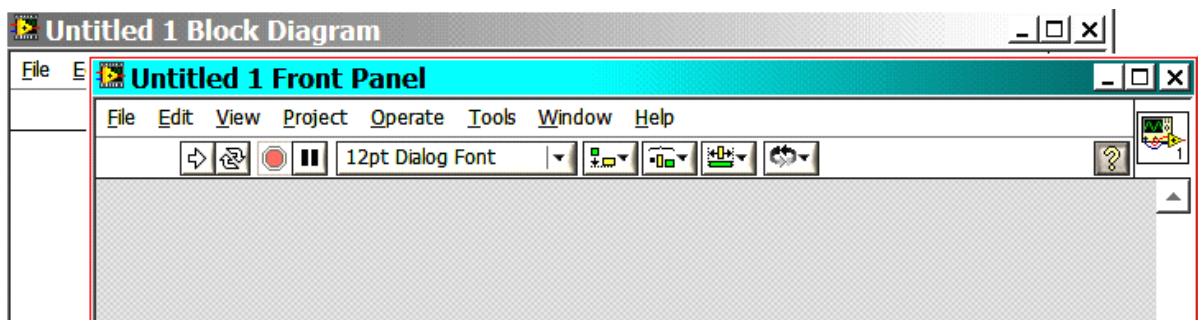


Рис.5. Окна пустого виртуального прибора

Упражнение 1. Первоначальное знакомство с техникой работы в среде LabVIEW

В данном упражнении нужно собрать схему сложения двух чисел, показанную на *рис.1*. Упражнение проводится параллельно с демонстрацией преподавателя, но можно его выполнить и самостоятельно по следующим этапам.

На блок диаграмме начнем построение ВП, который назовем "Sum X+Y". Щелчком правой кнопки мыши по полю блок-схемы вызываем палитру *Functions*. Наводим указатель мыши на нужные блоки и последовательно переходим по палитрам **Functions** >> **Programming** >> **Numeric** (*рис.6*).

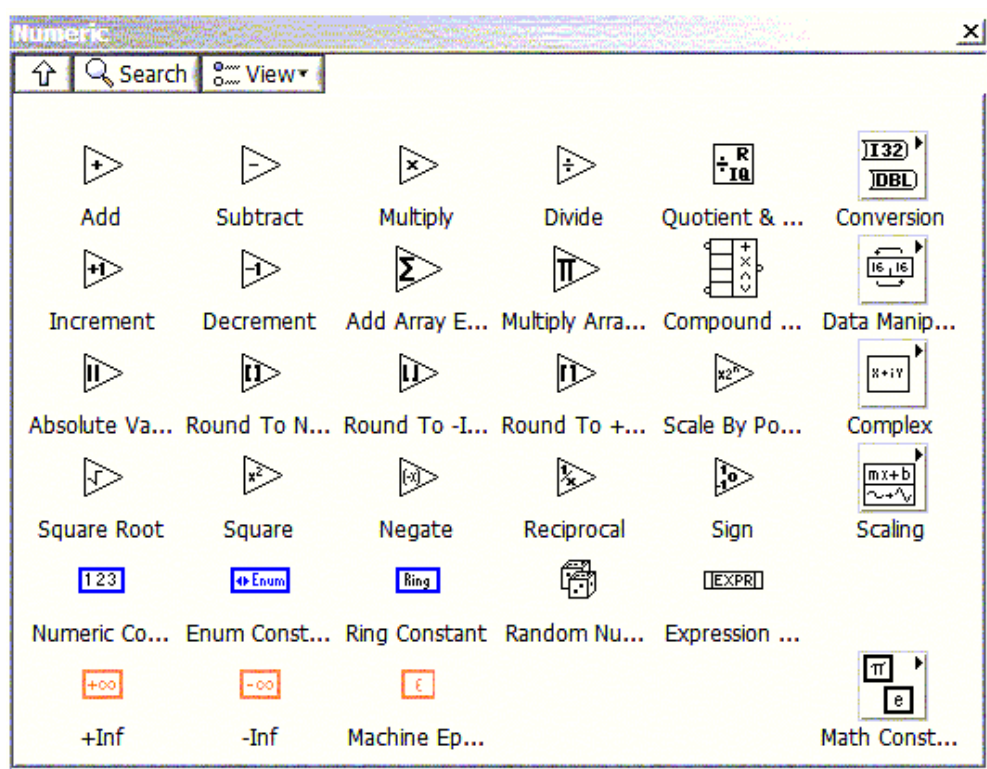




Рис.6 Палитра Numeric


На палитре **Numeric** щелчком мыши выбираем узел сложения **Add**. Далее переводим указатель мыши на блок-схему (кнопка отпущена), и, щелкнув мышью по выбранному месту блок-схемы, устанавливаем узел. Теперь нужно создать элементы ввода двух чисел и вывода результата сложения. Самый простой способ создания и присоединения стандартного терминала ввода (или вывода) к полю какого-либо узла – это поставить указатель мыши на нужное поле узла и, когда указатель примет вид соединительной катушки (**Wiring Tool**) , по щелчку правой кнопкой выбрать мышью в появившемся меню **Create** >> **Control** (или же **Create** >> **Indicator**). На

блок-схеме появится терминал именно того типа, который требует данное поле по умолчанию, а на лицевой панели возникнет элемент управления – **Control** (или, соответственно, индикатор) с тем же ярлыком. Надо создать для двух полей входа узла сложения терминалы ввода - **Control** и для поля выхода - **Indicator**.

Расположение всех элементов на блок-схеме и на лицевой панели можно изменить, выделяя их и перемещая при нажатой левой кнопке мыши.


Схема готова к работе. Перейдем на лицевую панель, нажав на блок-схеме пункт меню *Window >> Show Front Panel* (рис.5) или нажав *Ctrl+E*.

Запуск однократного выполнения ВП производится с помощью мыши кнопкой **Пуск (Run)**  на инструментальной панели, расположенной в верхней части окна программы, или же с клавиатуры, нажав *Ctrl+R*:

Если кнопка **Пуск** имеет "сломанный" вид , это значит - в программе есть ошибки. Надо нажать эту кнопку, появится окно **Error list / Список ошибок**. После исправления ошибок кнопка принимает исходный вид.

Для проверки работы программы введите на лицевой панели в окна ввода два произвольных числа и нажмите кнопку **Пуск**. На индикаторе должен появиться правильный результат сложения.

Внесите теперь ошибку в блок-схему, например, сотрите один из проводников на входе узла сложения. Для этого выделите его мышью и нажмите на клавиатуре клавишу *Delete*. Нажав **Пуск**, прочтите появившееся описание ошибки. Нажав далее кнопку **Show error**, посмотрите, какой элемент с ошибкой будет выделен на блок-схеме.

Восстановите стертый проводник. Для этого поставьте указатель мыши на выходное поле терминала ввода и когда указатель примет вид соединительной катушки (**Wiring Tool**) , протяните линию при нажатой левой кнопке мыши до входа узла сложения, после чего отпустите кнопку. Кнопка **Пуск** примет исходный вид.

Рассмотрим теперь выполнение этой же операции с применением узла Формулы (**Formula Node**), позволяющего вводить в программы LabVIEW текстовые операторы на языке C. Для этого соберем схему рис.7. Узел **Formula Node** вызывается из палитры **Functions >> Structures**. После выбора узла (щелчком мыши) нужно на блок-

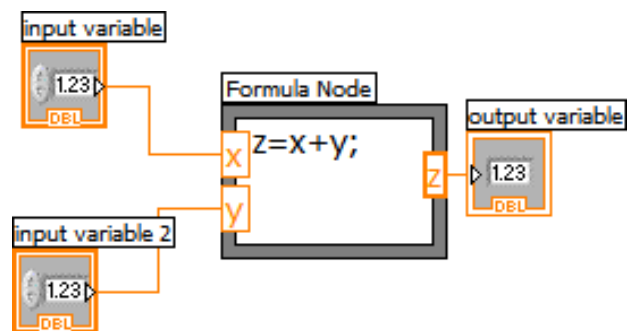


Рис. 7. Программа сложения двух чисел с использованием узла **Формулы**

диаграмме нарисовать мышью рамку, ограничивающую рабочую область этой структуры.

Чтобы создать терминалы входных и выходных данных, щелкните правой кнопкой мыши по границе узла. Далее в контекстном меню выберите пункт **Add Input** (Добавить вход) или **Add Output** (Добавить выход), а затем введите в поля данных терминалов идентификаторы переменных для входа (x, y) и выхода (z) (большие и малые буквы различаются).

Далее записываются операторы в рабочую область структуры. Каждое выражение должно заканчиваться разделителем (;). Если потребуются дополнительные переменные, кроме введенных для входа и выхода, то нужно ввести их описание по правилам языка C.

Справа приведены обозначения некоторых операций в порядке старшинства. Для получения полного перечня операторов и стандартных функций используйте справку **LabVIEW Help**.

**	возведение в степень
*, /	умножение, деление
+, -	сложение, вычитание

Чтобы сохранить собранную схему, выберите пункт главного меню **File >> Save**, откройте сетевой диск Z: по адресу [Students on 'Pc1' (Z:)], создайте в папке с номером своей группы папку со своей фамилией, например, **101 >> Ivanoff** и нажмите кнопку **Save**, сохранив ВП под именем " Sum X+Y ".

Упражнение 2. Виртуальный вольтметр.

В данной задаче используется многофункциональный внешний модуль для сбора данных типа USB 6009, подключаемый к компьютеру USB-кабелем (рис.8). Модуль имеет 8 аналоговых входов для измерения напряжения в диапазонах от ± 1 до ± 20 В, 2 аналоговых выхода, 12 комбинированных цифровых входов/выходов и вход счетчика.

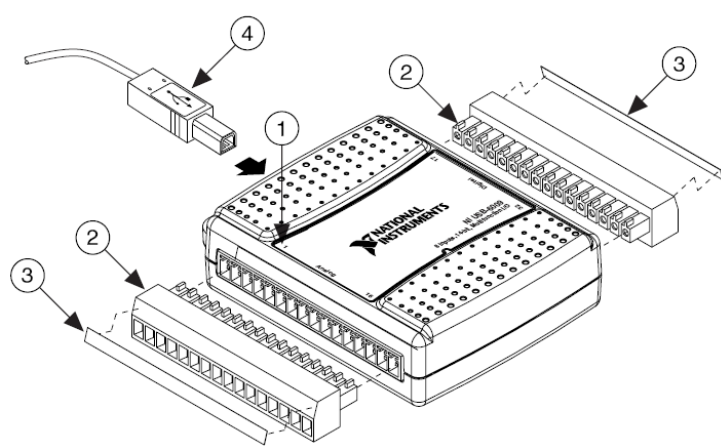


Рис.8. Измерительный модуль USB 6009

- ① DAQ модуль;
- ② Клеммные колодки;
- ③ Этикетки с обозначением сигналов;
- ④ USB кабель.

Клеммные контакты ② модуля рассчитаны на зажим проводов под винт. Для удобства пользования в данной задаче DAQ-модуль установлен внутри коммутационной коробки, и его аналоговые входы/выходы выведены от клеммных контактов на гнезда, расположенные на верхней панели коробки, куда при измерениях подключаются провода со стандартными штекерами (рис.9). Для измерения напряжений модуль имеет два режима:

1) **Single-ended mode / Несимметричный (однопроводной) режим:** измеряется напряжение между "Землей" модуля (гнезда 1, 4, 7, 10, 13 и 32, обозначенные как GND (сокращение от англ. *ground*) и соответствующими 8 клеммами входа 2, 3, 5, 6, 8, 9, 11, 12 (обозначаются **ai0 – ai7**). Этот режим можно применять, если **все** напряжения нужно измерять относительно потенциала только одной выбранной точки – "земли".

2) **Differential mode / Дифференциальный (симметричный) режим:** измеряется **разность** напряжений между гнездами 2-3, 5-6, 8-9 и 11-12 (входы, обозначенные **ai0 – ai3**). Это универсальный режим и его можно применять как измерений напряжений относительно земли, так и при измерении напряжений между любыми произвольными точками электрических схем. Этот режим и потребуется в нашей задаче.

Гнезда 11, 12 дифференциального входа **ai3** соединены внутри коробки через сопротивление 1 Ом. Этот дифференциальный вход будет использо-

National instruments Data Acquisition (DAQ) USB terminal							
Цифровые входы / выходы		Аналоговые входы / выходы		29	32	31	
№ конт. разъема	№ контакта	№ контакта	Сигнал	PFIO	Счетчик	0 V	+5 V
1	17	1	P0.0	GND	Вход	Выход	Выход
2	18	2	P0.1	GND	ai 0	ao 0	ao 1
3	19	3	P0.2	GND	ai 4	ai 3 +	ai 7
4	20	4	P0.3	GND	ai 1	ai 2 +	ai 6
5	21	5	P0.4	GND	ai 5	ai 1 -	ai 5
6	22	6	P0.5	GND	ai 2	ai 0	ai 4
7	23	7	P0.6	GND	ai 6	ai 2 -	ai 3 -
8	24	8	P0.7	GND	ai 3	ai 3 +	ai 3 -
9	25	9	P1.0	GND	ai 7	ao 0	ao 1
10	26	10	P1.1	GND	GND	GND	GND
11	27	11	P1.2	GND	GND	GND	GND
12	28	12	P1.3	GND	GND	GND	GND
13	29	13	PFI 0	GND	GND	GND	GND
14	30	14	+2.5V	GND	GND	GND	GND
15	31	15	+5 V	GND	GND	GND	GND
16	32	16	GND	GND	GND	GND	GND

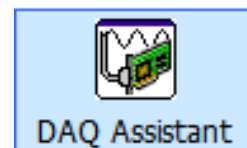
Analog input (ai) / output (ao)

Рис.9 Панель коммутационной коробки

ваться для измерения силы тока. По закону Ома измеряемое на этом входе напряжение в вольтах будет численно равно силе тока в амперах через этот резистор.

Модуль имеет также маломощный выход постоянного напряжения от порта USB с напряжением +5 В (относительно земли) с током до 200 мА (гнездо 31). Это напряжение можно использовать для питания наших измерительных схем. **Для ограничения тока короткого замыкания внутри коробки установлен резистор 27 Ом между клеммой 31 на модуле DAQ и гнездом 31 на панели.**

Для программирования работы измерительных **DAQ**-модулей в системе LabVIEW имеется большой набор готовых ВП, выполняющих определенные группы операций по обслуживанию **DAQ**-модулей. Однако стандартные задания, не требующие расширенных возможностей, можно чрезвычайно просто запрограммировать с помощью экспресс-ВП **DAQ Assistant**. Для его запуска нужно щелкнуть мышью по значку **DAQ Assistant** (находится в палитре **Functions >> Express >> Input**), а затем щелкнуть по свободному месту на блок-схеме, куда мы хотим поместить этот узел.



На блок-схеме появится иконка программы **DAQ Assistant** и диалоговое окно настройки **DAQ**-модуля, в котором надо выбрать **Analog Input (Аналоговый вход)** и затем выбрать из списка измеряемых величин **Voltage (Напряжение)**. Появится окно с названием **DAQ**-модуля, подключенного к компьютеру, и со списком имеющихся физических каналов его аналоговых входов **ai0 – ai7** (рис.10). Необходимый нам дифференциальный режим возможен только на входах **ai0 – ai3**.

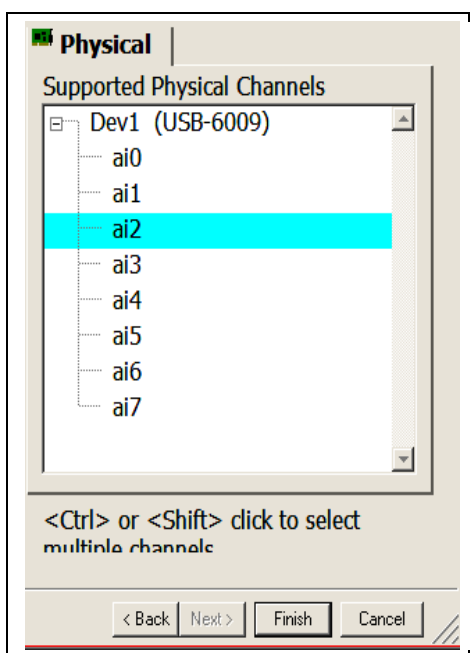



Рис.10. Диалог выбора каналов

Поскольку вход **ai3** уже зарезервирован ранее под измерение силы тока, выбираем мышью любой из остальных, например, **ai2**, и нажимаем кнопку **Finish**. Появится окно добавления и настройки каналов (рис.11).

В разделе **Signal Input Range (Диапазон входного сигнала)** поставим Max: 10, Min: -10. Стоящий по умолчанию вариант "**N samples**" (**N** измерений) в разделе **Acquisition Mode (Режим сбора данных)** обеспечит выполнение серии из **N** измерений, где величина **N** определяется окном **Samples to read** и по умолчанию равна **N = 1000**.

Для проверки работы канала нажмем кнопку **Test**  наверху окна. Поскольку выбранный вход **ai2** пока никуда не подключен, на появившейся осцилло-

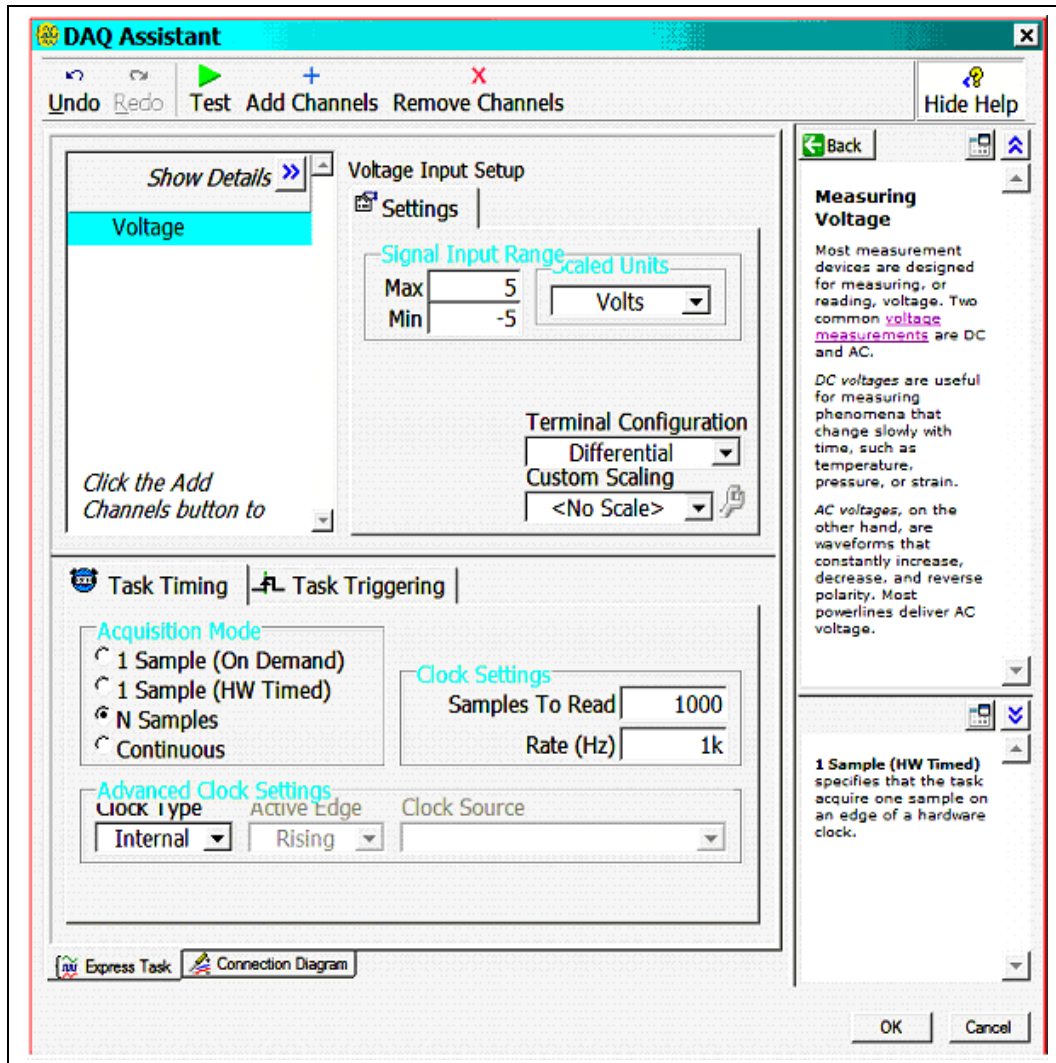


Рис.11. Диалог создания и настройки каналов. Измерение напряжения

грамме из 1000 точек будут видны лишь слабые шумы амплитудой 1-2 мВ вокруг нулевого среднего значения (рис.12). Подсоединим теперь вход **ai2** к гнездам 31 (+5 В) и любому гнезду GND (Земля). На осциллограмме средний уровень напряжения должен возрасти до 5 В.

Убедившись в работоспособности канала, закрыва-

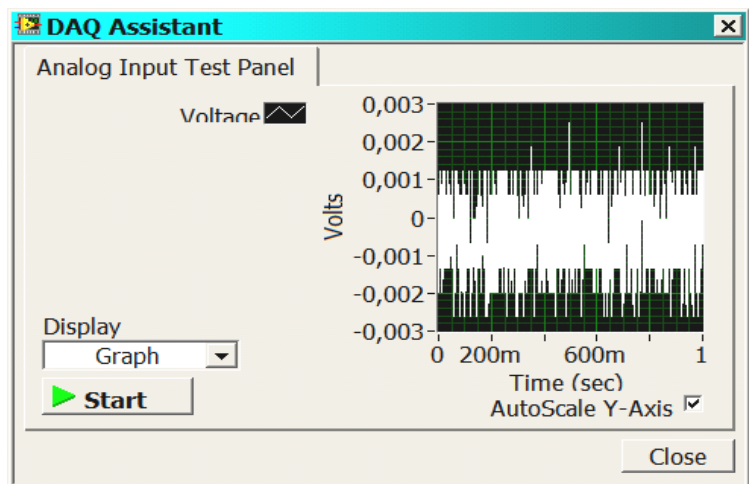


Рис.12. Окно Test. Видны шумы работающего канала

ем кнопкой **Close** окно **Test** и нажимаем кнопку **OK** (Рис.11), подтверждая выбранные параметры.

DAQ-модуль запрограммирован и готов к измерению напряжения.

Чтобы завершить создание вольтметра, осталось подключить индикатор к выходу узла **DAQ Assistant**. Предварительно надо раскрыть все поля этого узла, растянув его мышью за нижний или верхний край до полного размера (рис.13). Цифровой индикатор получаем установив указатель мыши на поле *Data* и выбрав по щелчку правой кнопки мыши **Create >> Numeric Indicator**.

Стрелочный индикатор (и все остальные, кроме стандартного цифрового) таким образом получить нельзя. Сначала надо создать этот индикатор на лицевой панели. Для этого щелчком мыши по лицевой панели вызываем

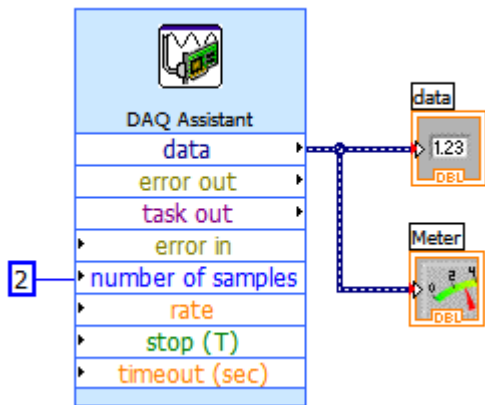


Рис.13 Вольтметр: блок-схема

палитру **Controls** и далее на палитре **Controls >> Numeric** выбираем щелчком мыши стрелочный индикатор **Meter** и устанавливаем его на лицевой панели (рис.14). При этом на блок-схеме автоматически появится его терминал, который нужно присоединить к тому же полю *Data*. Если пределы шкалы стрелочного индикатора будут отличны от требуемого диапазона 0 – 10 В, то их можно заменить прямо на самой шкале. Для этого надо двойным щелчком мыши выделить крайнее число шкалы и записать туда с клавиатуры новое значение.

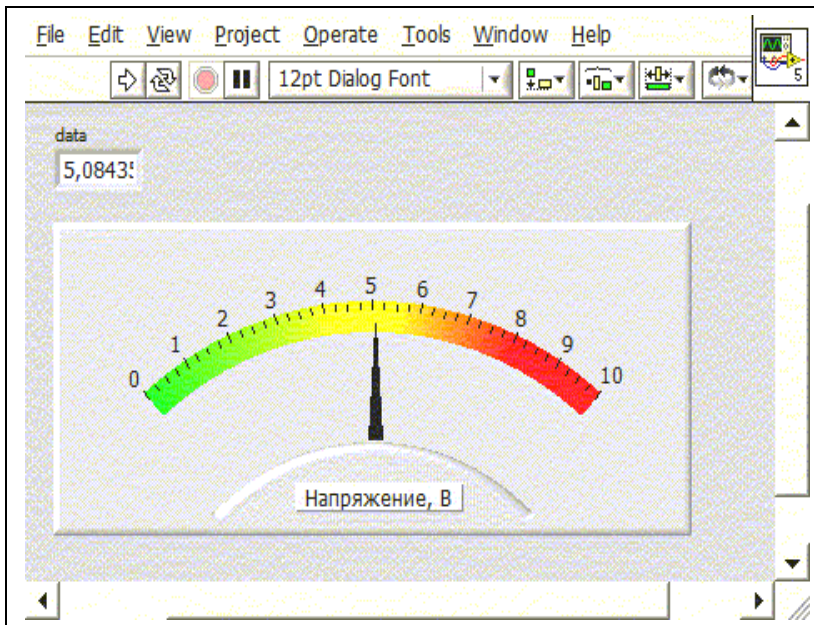



Рис.14. Вольтметр: лицевая панель

атюры новое значение. Вся шкала автоматически изменит свой масштаб в соответствии с новыми пределами.

Предусмотренные системой варианты редактирования элементов управления и отображения можно просмотреть и установить через пункт **Properties** их кон-

текстного меню, которое вызывается щелчком правой кнопки мыши по объекту. Поскольку пока нам не нужны 1000 измерений, то для проведения минимального числа измерений подадим на вход *Number of samples* число 2 (меньше нельзя по особенностям работы программы). После щелчка правой кнопкой на данном входе выбираем в меню **Create >> Constant**, и в появившемся прямоугольнике поставим 2. Окончательная схема вольтметра показана на *рис.13*.

Подключите гнездо "-" к любому гнезду GND "земля", а гнездо "+" дифференциального входа 2 (**ai2**) к гнезду +5 В. Нажмите **Пуск** . Стрелка вольтметра покажет значение, близкое к 5 вольтам, а в окне появится численное значение измеренного напряжения (*рис.14*). Зарисуйте собранную блок-схему.

Оформите лицевую панель и блок диаграмму так, чтобы было понятно, какие физические величины вы измеряете и в каких единицах. При оформлении блок-схемы и лицевой панели для нанесения дополнительных надписей и комментариев нужно дважды щелкнуть мышью по выбранному месту и ввести текст в открывшееся окно ввода. Некоторые декоративные элементы можно добавить из палитры на лицевой панели **Controls >> Modern >> Decorations**. Приведенные в описание рисунки примерно такие, как выдаются Вам программой по умолчанию. **Оформление лицевой панели и блок-диаграммы – ваша индивидуальная работа.** Общие правила для оформления блок-диаграммы: все проводники располагайте так, чтобы было четко видно, где они начинаются, где проходят и кончаются (минимум пересечений). Блоки (узлы, терминалы), выполняемые и заданные раньше, располагаются выше и левее.

Проведите 5 - 7 измерений, данные занесите в тетрадь. Представьте полученный результат с учетом приборной погрешности и случайной (шумов). Приборная погрешность DAQ-модуля USB 6009, в зависимости от выбранного вами диапазона измерений, представлена в Таблице 1.

Работу вольтметра надо показать преподавателю.

ВП «Вольтметр» сохраните в свою папку.

Таблица 1

Абсолютная погрешность DAQ-модуля USB 6009 на пределе шкалы в дифференциальном режиме, мВ

Диапазон, В	Погрешность при 25°C (мВ)
±20	14.7
±10	7.73
±5	4.28
±4	3.59
±2.5	2.56
+2	2.21
±1.25	1.70
±1	1.53

Упражнение 3. Виртуальный омметр.

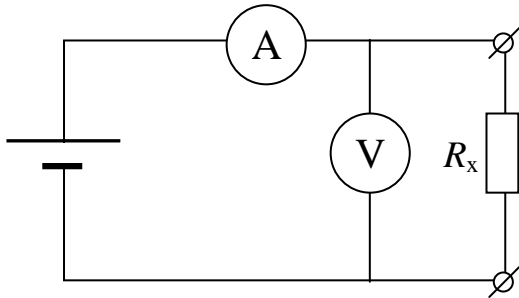


Рис.15. Омметр: Принципиальная схема

Сопротивление участка цепи рассчитывается по закону Ома $R_x = U / I$.

Поскольку вольтметр уже был создан в упражнении 2, надо дополнительно создать амперметр. Для измерения силы тока будем использовать другой канал **DAQ**-модуля (**ai3**). Измеряя падение

напряжения на резисторе сопротивлением 1 Ом, подключенном внутри платы к дифференциальному входу **ai3** (гнезда 11-12), получим значение силы тока, численно равное напряжению.

Нарисуйте в тетради схему *рис.15* в явном виде: представьте амперметр в виде вольтметра и дополнительного шунтирующего резистора. Запишите чему равно неизвестное сопротивление через показания двух вольтметров. Напишите, чему равна погрешность такого (косвенного) измерения.


Чтобы продолжить работу на основе уже собранного выше ВП, сохраните его в своей папке под новым именем, например, *Измерение R*, воспользовавшись пунктом меню **File >> Save as** (создаем копию).

Нам потребуется только узел **DAQ Assistant**, поэтому на блок-схеме сотрем все остальное.

Для открытия второго физического измерительного канала нужно двойным щелчком мыши по иконке **DAQ Assistant** снова открыть *Диалог создания и настройки каналов* (*рис.16*), установленный нами ранее на измерение напряжения.

Далее нажимаем  / **Добавить каналы** и в открывшемся меню выбираем **Current / Ток** (*рис.16*).

В окно **Resistor value (ohms) / Номинал резистора (Омы)** надо ввести 1 (именно такой резистор сопротивлением 1 Ом припаян к входу **ai3**), а в окне **Signal Input Range / Входной Диапазон Сигнала** установить ± 200 мА или ± 1 А. Нужно также проверить, что опция **Shunt Resistor / Шунтирующий резистор** установлена на **External / Внешний**.

Далее проверяем работоспособность обоих каналов кнопкой  **Test** и нажимаем **ОК**. Канал измерения силы тока подключен.

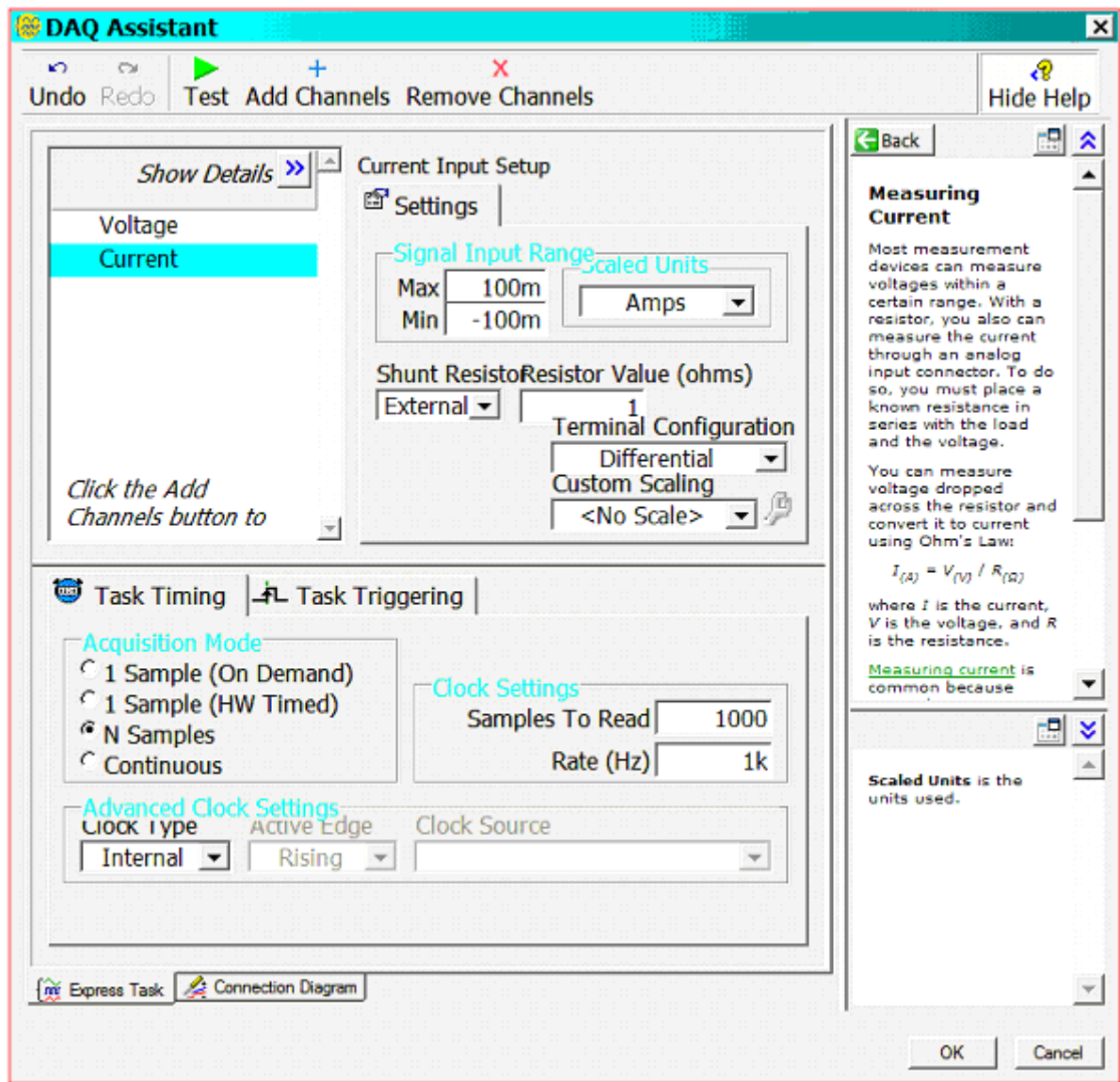
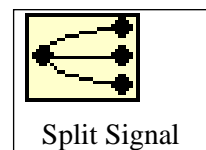


Рис.16 Диалог настройки каналов. Измерение силы тока

В модуле **USB 6009** один измерительный аналого-цифровой преобразователь по очереди обслуживает все каналы ввода. Поэтому после введения второго канала поле выхода *data* на узле **DAQ Assistant** по-прежнему остается одним, но по нему будут последовательно во времени выходить два разных цифровых динамических сигнала, относящиеся к двум установленным каналам – напряжения и тока. Для разделения этих сигналов используется узел **Split Signal** (Палитра **Functions >> Express >> Signal manipulation**).



Данный узел ставим вблизи выхода **Data** и соединяем с ним проводником. Первоначально узел **Split Signal** имеет один выход. С помощью мыши его нужно растянуть по вертикали за нижний или верхний край до получения двух выходов (Рис. 17). Верхний будет выходом напряжения, нижний – силы тока (в соответствии с порядком добавления каналов, см. рис.16).

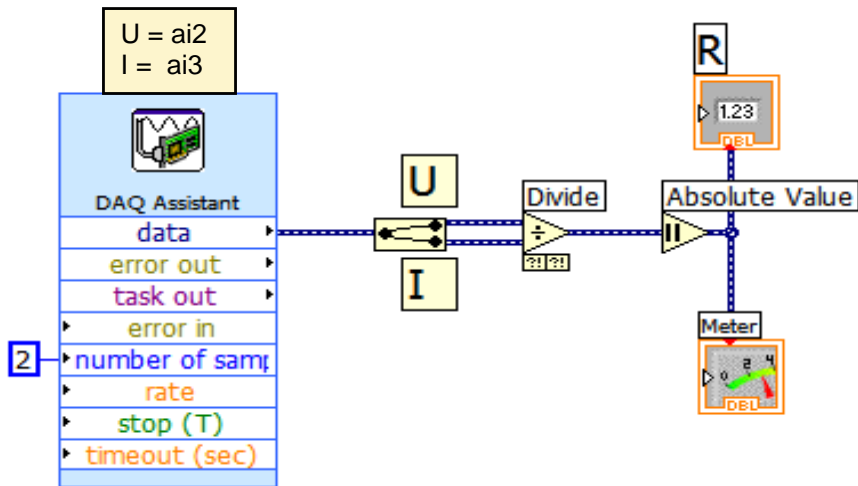


Рис.17 Омметр: блок-схема

Чтобы не забыть, какие каналы, и в каком порядке подключены, целесообразно подписать их в том же порядке над иконкой DAQ, как это сделано на рис.16. Для этого нужно дважды щелкнуть мышью над иконкой и в появившемся поле ввода набрать их названия.

Для получения величины сопротивления нужно поделить напряжение на силу тока, что выполняется узлом *Divide*.

Окончательная схема омметра показана на рис.17. На выходе установлены цифровой и стрелочный индикаторы. Поскольку полярности подключения входов могут быть разными, а величина сопротивления должна быть положительна, на выходе узла деления поставлен узел нахождения модуля *Absolute Value*, взятый, как и узел деления *Divide*, из палитры **Functions >> Numeric**.

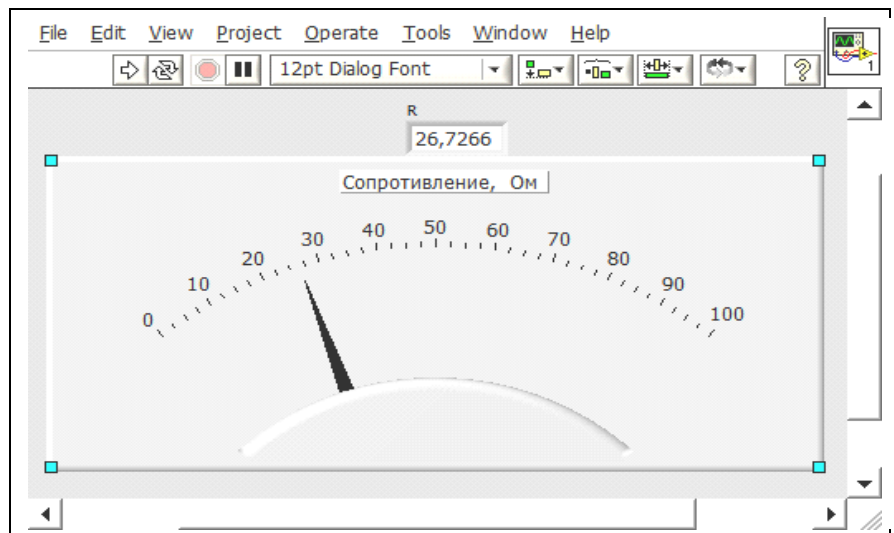


Рис.18 Омметр: лицевая панель

Примерная лицевая панель прибора показана на *рис.18*. Дополнительные поясняющие надписи на панели можно сделать указанным в Упражнении 2 способом.

Поскольку величины измеряемых сопротивлений могут лежать в широких пределах, можно сделать шкалу с логарифмическим масштабом с пределами индикации 0.1 – 1000. Для этого надо щелкнуть правой кнопкой по стрелочному индикатору и выбрать мышью пункт *Properties (Свойства)*. На закладке *Scale (Шкала)* надо поставить галочку в окне *Logarithmic*. Пределы измерения также можно установить здесь же в окнах *Scale range (Диапазон)*.

Созданный ВП теперь можно использовать для измерения сопротивления участков цепи в любой схеме с протекающими в ней токами, подключенной к внешнему источнику ЭДС. Для этого нужно подключить гнезда амперметра 11-12 (Дифф. Вход ai3) в разрыв измеряемой цепи, а гнезда вольтметра 8-9 (Дифф. Вход ai2) параллельно измеряемому сопротивлению и нажать **Пуск**.

В нашем случае, используя собственный источник напряжения +5 В, имеющийся на плате, можно сделать и автономный омметр. Для этого соберем схему омметра согласно *рис. 15, 19*.

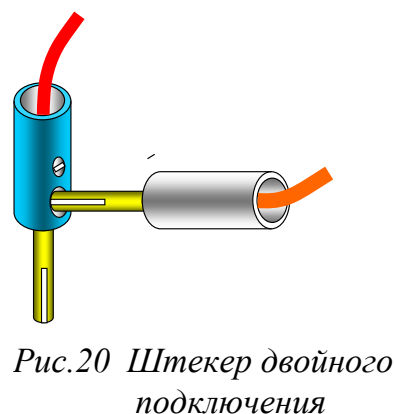
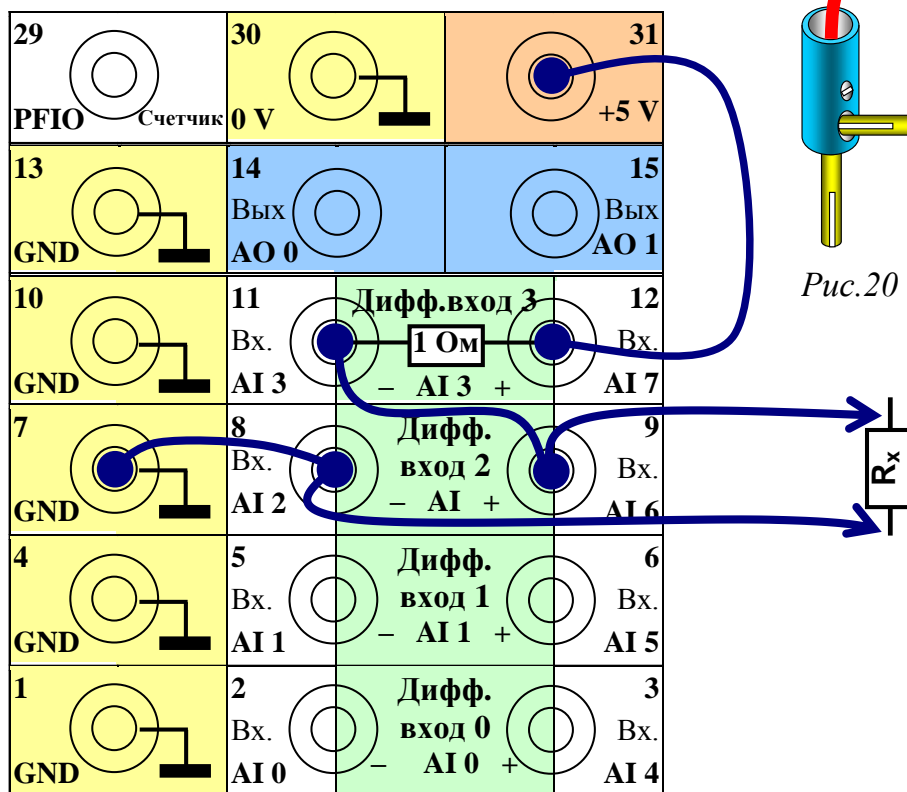


Рис.19 Омметр: Схема соединений на панели

При сборке схемы потребуется присоединение двух проводов в одну клемму (клеммы 8 и 9). Для этого используйте имеющиеся в комплекте задачи два провода со специальным штекером, в котором есть отверстие для подключения второго штекера (*рис.20*).

Задание

Нужно измерить сопротивления неизвестных резисторов, а также рассчитать максимальную приборную погрешность результатов на основании собственной точности DAQ-модуля USB 6009, приведенной в *таблице 1*. Для определения приборной погрешности для силы тока нужно учесть, что реально измеряется **напряжение** на резисторе 1 Ом, и определить, на каком из приведенных в таблице и имеющихся у модуля диапазонов напряжения работает данный канал на выбранном диапазоне тока.

Расчет погрешностей можно запрограммировать самостоятельно (упр. 5) или же провести вручную. Для ручного расчета потребуются значения напряжения и силы тока для каждого резистора, поэтому надо **добавить в схему цифровые индикаторы для U и I** и также записать их показания вместе с величиной R .

Подключая к омметру по очереди **все** имеющиеся в комплекте задачи неизвестные резисторы, расположенные на монтажной панели, запускаем однократные измерения и записываем полученные результаты в таблицу, где надо указать также **номер панели** и номера резисторов.

Замечание. На некоторых резисторах может получаться большой разброс результатов при повторении измерений. Это связано с тем, что выбранные при настройке каналов максимальные диапазоны измерения в этих случаях не подходят либо 1) из-за слишком малых токов (это будет при больших величинах сопротивления R_x), либо 2) из-за слишком малых напряжений (при малых R_x). В обоих этих случаях амплитуда шумов становится сравнима с измеряемым сигналом, что и вызывает большой случайный разброс результатов.

Наилучшим решением этой проблемы является оперативное переключение диапазонов напряжения и тока в зависимости от величины R_x , но это вызвало бы усложнение задачи. Другим вариантом, частично решающим эту проблему, является статистическое усреднение измеряемых напряжений и токов, что предусмотрено в Упражнении 5.

Упражнение 4. Программирование расчета случайных погрешностей вольтметра

Упражнения 4 и 5 выполняются самостоятельно после завершения упражнений 1-3. Они не имеют подробного описания всех выполняемых действий, как в предыдущих упражнениях, поскольку подразумевают наличие у выполняющего приобретенных навыков работы с системой.

Задание: добавить статистическую обработку для собранного в упражнении 2 вольтметра (рис.13).

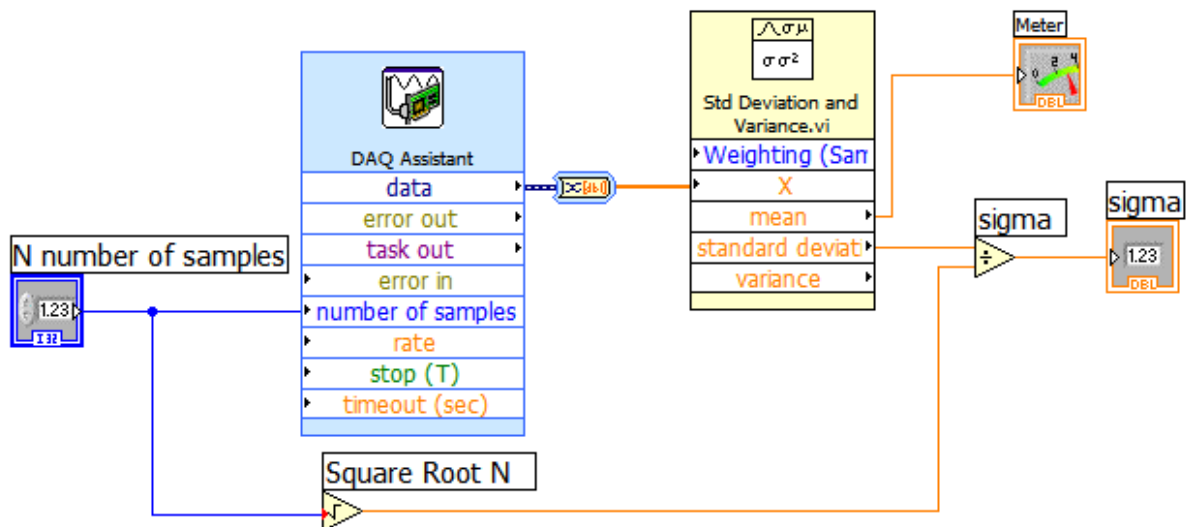


Рис.21. Вольтметр с расчетом случайных погрешностей измерения: Блок-схема

Система LabVIEW имеет несколько функциональных узлов статистической обработки данных, которые расположены на палитре **Functions >> Mathematics >> Prob & Stat**. Используем один из них – **Std Deviation & Variance**, который рассчитывает среднее арифметическое (*mean*) и стандартное отклонение (*Std Deviation*) для случайной величины, заданной массивом чисел на входе *X*.

Откройте ВП вольтметра и поместите на него узел **Std Deviation&Variance**. При установки на блок-схему этот узел имеет вид иконки без надписей на полях ввода / вывода. Чтобы раскрыть узел, нужно щелкнуть по нему правой кнопкой и в контекстном меню снять галочку "**View as Icon**", а затем растянуть узел по вертикали до полного раскрытия.

Чтобы поясняющие надписи на полях ввода / вывода не обрезались, можно в том же контекстном меню поставить галочку на пункте *Size to Text*, при этом узел увеличится по ширине.

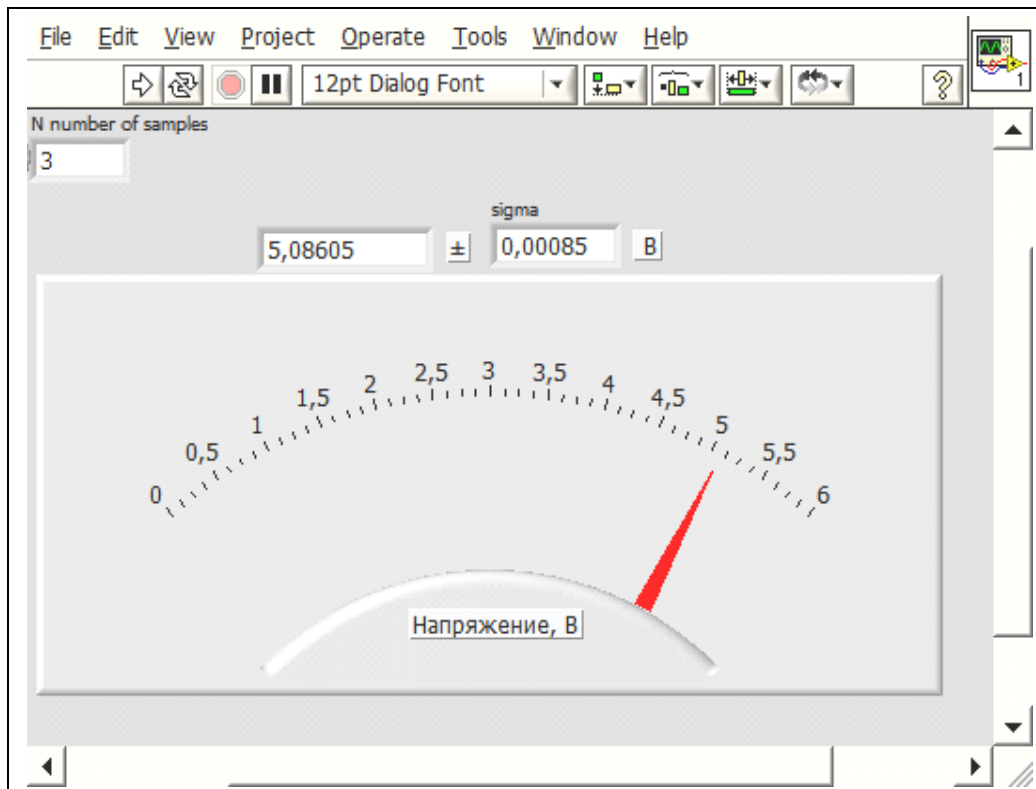



Рис.22. Вольтметр с расчетом случайных погрешностей измерения: примерный вид лицевой панели

Соберите ВП согласно рис.21, 22. Обратите внимание, что при соединении узла **DAQ Assistant** с узлом **Std Deviation&Variance** система сама поставит узел , преобразующий динамические данные, идущие от выхода *data*, в массив чисел, которого требует на входе *X* узел **Std Deviation&Variance**. Чтобы иметь возможность менять число измерений, добавим ввод целых чисел на вход **Number of samples**.

Стандартное отклонение σ для среднего арифметического рассчитывается по формуле $\sigma = s / \sqrt{N}$, где s – выборочное стандартное отклонение для случайной переменной.

Подключите вольтметр к напряжению +5 В (гнездо 31 и любое GND).

Запишите результаты измерений этого напряжения при разных значениях N – числа измерений в серии, начиная с $N = 2$ и далее 5, 10, 100... Для каждого N проведите 2-3 измерения. Представьте эти результаты графически (рис.23). Сравните получаемые погрешности с приборными погрешностями самого DAQ-модуля.

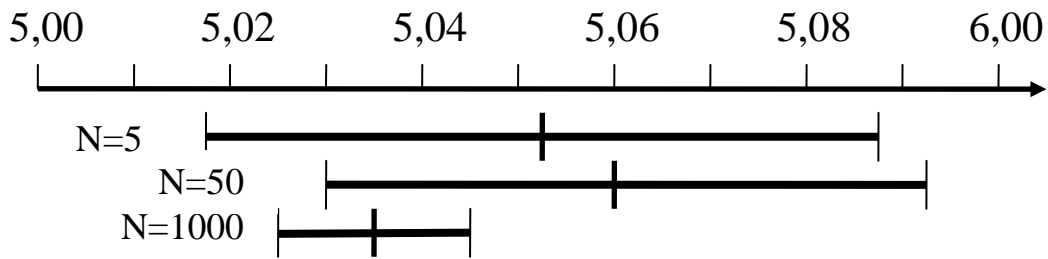


Рис.23. Результаты измерения напряжения в зависимости от числа измерений в серии.

Измеряемое напряжение содержит довольно много шумов, что мы видели на осциллограмме при проверке работы каналов в упражнении 2. Сам компьютер и **DAQ**-модуль являются достаточно "шумными" импульсными приборами. Важно отметить, что путем увеличения числа измерений можно уменьшить погрешность, вызванную случайными шумами, до любого нужного уровня, пренебрежимо малого по сравнению с абсолютной погрешностью самого прибора.

Упражнение 5. Программирование расчета случайных и приборных погрешностей омметра

1) добавить статистическую обработку для собранного в упражнении 3 омметра (рис.17). Данный ВП нужно разработать самостоятельно, зарисовать в тетради блок-схему и показать его работу преподавателю.

Замечание. Обратите внимание, что статистическое усреднение нужно проводить не для конечного результата – величины сопротивления R , а для первичных измерений – напряжения и силы тока, а погрешность для R рассчитывается потом как погрешность косвенных измерений.

2) добавить расчет приборной ошибки в соответствии с табл. 1.

При выполнении этого упражнения для расчетов погрешностей целесообразно использовать **узел Формулы**. При редактировании нужно так располагать проводники, чтобы были четко видны места их подключения, иначе при распечатке эту схему будет невозможно понять.

Примерные вопросы к допуску

- 1) Принцип работы АЦП последовательного приближения (блок-схема и временные зависимости при оцифровке напряжений).
- 2) Основные параметры АЦП – разрядность и частота дискретизации и их влияние на точность измерений и возможность наблюдения меняющихся во времени сигналов.
- 3) Особенности управления исполнением операций в языке LabVIEW.
- 4) Расчет статистической и приборной погрешности при измерении сопротивления.

При сдаче задачи нужно: знать перечисленные выше вопросы для допуска; уметь воспроизвести на компьютере виртуальные приборы, рассмотренные в задаче, а также, возможно, и другие простые виртуальные приборы по заданию преподавателя, создание которых не требует материала, выходящего за рамки данного описания.

В тетради должен быть конспект (домашняя подготовка, описание выполнения работы, обработка и расчет результатов), обработанные и полностью оформленные результаты измерений с указанием приборных и случайных погрешностей.

Представляя в конце задачи итоги работы и выводы, опишите, какие приборы вы создали: что ими можно измерять, в каких диапазонах и с какой точностью. Приведите ваши рекомендации по количеству необходимых измерений (поясните, почему именно столько).

Приложение. Элементы программирования в системе LabVIEW.

В LabVIEW блоки выполняются последовательно (поточная модель обработки данных). Этим язык LabVIEW отличается от текстовых языков программирования (C, Pascal и др.), где порядок выполнения операций задается последовательностью записи операторов. Очередность выполнения блоков определяется их зависимостью друг от друга. Если один блок нуждается в данных, которые выдаются другими блоками, то он не начнет выполняться до тех пор, пока не получит от них всех необходимых данных. Таким образом, обработка потока данных осуществляется поступлением самих данных. Если же данные образуют замкнутый цикл, это будет ошибкой.

LabVIEW может оперировать с разными типами данных, обозначаемых на схемах цветом. В данной задаче встретятся следующие типы:

Numeric – численные типы

Integer – целочисленный тип (**синий**).

Float – вещественный с плавающей запятой (**оранжевый**).

Boolean – логический тип, 0 (FALSE) или 1 (TRUE) (**зеленый**).

Array – массивы, т.е. пронумерованные наборы однотипных данных, имеет тип составляющих их элементов и принимают соответствующий им цвет.

Waveform – сигнальный тип данных – кластер, то есть набор разнотипных элементов, содержащий массив значений самих данных, начальное значение времени и интервал времени между измерениями.

Dynamic – динамический тип, отображается в виде **темно-синих** терминалов и проводников. Кроме данных самого сигнала, динамический тип содержит дополнительную информацию, например, название сигнала или дату и время его получения. Большинство ВП принимают и/или возвращают данные динамического типа. Данные динамического типа можно направлять к любому элементу отображения или полю ввода, принимающему данные численного, логического или сигнального типа.

Числовые функции

Арифметические функции являются полиморфными. Это означает, что на поля ввода этих функций могут поступать данные различных типов в разных комбинациях – скалярные величины, целые и вещественные (**Numeric**), массивы (**Array**), сигнальные типы (**Waveform**). Результатом операции будет тип с большей размерностью.

Узел Формулы

Каждый стандартный узел LabVIEW из палитры **Functions** выполняет только одну операцию, например, сложение (+), умножение (×) и т.д. Это наглядно, но в случае больших математических выражений выглядит на блок-схеме громоздко и неудобно. Поэтому в LabVIEW предусмотрена также возможность вставлять узлы-структуры с традиционными текстами программного кода на языке C, что выполняется с помощью узла **Формулы (Formula Node)**. Использовать этот узел удобно, когда математические выражения имеют много переменных, или они достаточно сложные, а также для использования уже имеющихся текстовых математических кодов, которые можно копировать и вставлять в него. В один узел можно писать много формул, задавать циклы.

Преобразование типов данных

Если при редактировании программы к полю ввода узла подключается проводник с данными, не предусмотренными для данного узла, то система LabVIEW пытается сама преобразовать их к нужному, на ее взгляд, типу, вставляя нужные узлы преобразования. В большинстве случаев ей это удается, если же нет, то появляется сообщение об ошибке.

Автоматическое преобразование типов происходит и при приведении к одному типу нескольких аргументов для числовых операций. При этом в месте соединения проводника с полем входа появляется точка с цветом, соответствующим преобразованному типу. На выходе операции получается тип с большим диапазоном значений.

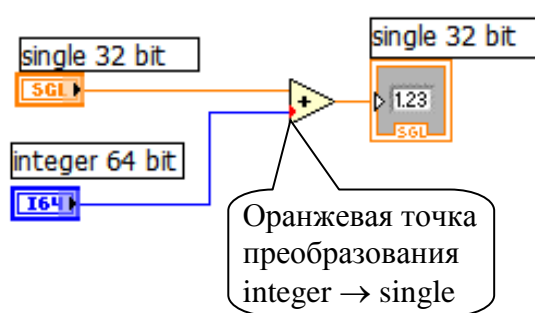


Рис.24. Пример преобразования типов данных

Например, данные типа I 64 (64-битный целочисленный со знаком) и SGL (32-битное число с плавающей запятой одинарной точности) поступают на входные терминалы функции сложения (рис.24). На выходе функции формируются данные в представлении SGL, поскольку тип I 64, несмотря на большее количество бит, имеет меньший диапазон возможных значений.

Литература

Тревис Дж. LabVIEW для всех / пер. с англ. М.: ДМК Пресс, 2004.

Суранов А.Я. LabVIEW 7: справочник по функциям. М.: ДМК Пресс, 2005.

Бутырин П.А., Васьковская Т.А., Каратаева В.В., Материкин С.В. Автоматизация физических исследований и эксперимента: компьютерные измерения и виртуальные приборы на основе LabVIEW 7. М.: ДМК Пресс, 2005

Митин И.В., Русаков В.С. Анализ и обработка экспериментальных данных. М.: Физич.ф-т. МГУ, 2002.