

**DR. BOB DAVIDOV**

## **Среда разработки интерфейсов и программных модулей систем управления LabVIEW**

*Цель работы:* освоение правил работы в среде LabVIEW.

*Задача работы:* построение интерфейса пользователя LabVIEW.

*Приборы и принадлежности:* Персональный компьютер, LabVIEW.

### **ВВЕДЕНИЕ**

Среду MatLAB, разработанную специалистами по автоматизации изначально для решения задач анализа и синтеза систем управления хорошо дополняет LabVIEW как среда для построения графических и аппаратных интерфейсов и систем с известными алгоритмами управления. Постепенно LabVIEW включает в себя все больше методов хорошо работающих в MatLAB, таких, например, как оптимизация регулятора, идентификация, и др. Многие компании в том числе и National Instruments – разработчик LabVIEW, предлагают различные контроллеры, устройства ввода-вывода, и другие устройства работающие совместно с LabVIEW. Понимание особенностей LabVIEW позволяет расширить возможности разработчика традиционно использующего MatLAB для решения задач управления.

### **ОБЩИЕ СВЕДЕНИЯ**

LabVIEW - среда графического программирования (последовательного соединения функциональных блоков на блок-диаграмме) на языке «G» фирмы National Instruments. LabVIEW используется для быстрого создания комплексных приложений в задачах измерения, тестирования, управления, автоматизации научного эксперимента.

Используемый в LabVIEW графический язык программирования 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.

Программа LabVIEW называется виртуальным прибором имеет расширение VI (Virtual Instrument) и состоит из двух частей:

- лицевой панели, описывающей внешний интерфейс прибора и
- блок-диаграммы, описывающей логику работы прибора.

Лицевая панель виртуального прибора (см. пример на Рис. 3) может включать средства ввода-вывода: кнопки, переключатели, индикаторы, табло, графопостроители и т.п. а также визуальные средства построения интерфейса.

Блок-диаграмма (см. пример на Рис. 3) содержит функциональные узлы передачи и обработки данных. Узлы обработки данных помимо функциональных блоков содержат блоки задающие последовательность обработки данных: **If**, **For**, **While** и т. п.

Виртуальный прибор может использоваться для построения других виртуальных приборов.

## **Интерфейсы LabVIEW**

LabVIEW поддерживают работу с интерфейсами RS-232/422/485, CAN, GPIB-488, TCP/IP и др.

## **EXE формат виртуального прибора**

Виртуальный прибор \*.vi не работает без среды LabVIEW, однако его можно перевести в исполняемый \*.exe файл при помощи специального компонента LabVIEW **Application Builder**. Для работы таких exe программ требуется бесплатно распространяемый компонент **LabVIEW Runtime Engine** и, при необходимости, драйверы используемых внешних устройств.

Автономно выполняющуюся EXE программу виртуальных прибора можно создать при помощи компоненты фирмы National Instruments **LabWindows/CVI** в которой реализована концепция не графического, а визуального программирования. Лицевая панель виртуального прибора формируется так же, как и в LabVIEW, а алгоритм работы прибора программируется на языке Си.

## **Работа в реальном времени**

Для работы с LabVIEW в реальном времени необходимо установить специальную библиотеку **NI Real-Time** и использовать ее элементы.

Специализированное измерительное оборудование фирмы National Instruments содержит встроенный микроконтроллер на котором выполняется специальное приложение обеспечивающее работу LabVIEW прибора в жёстком реальном времени на обычном персональном компьютере.

## **Measurement & Automation Explorer (MAX)**

Все конфигурационные параметры устройств и каналов передачи данных задаются в пакете MAX, который считывает записи менеджера устройств в системном реестре Windows и присваивает каждому устройству сбора данных (DAQ-устройство компании NI) логический номер, который используется в LabVIEW в качестве ссылки на устройство.

Примечание. Таким устройством может быть интерфейсная карта NI PCI-6014, которая не только совместима с LabVIEW но и имеет драйвера в MatLAB для совместной работы **simulink** модели с картой PCI-6014 в реальном времени.

Открывается MAX через список программ компьютера:  
или из меню LabVIEW: Tools > Measurement & Automation Explorer.

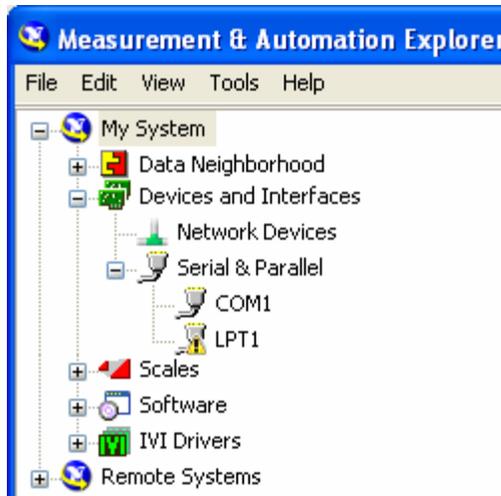


Рис. 1. Стартовое окно MAX

### Создание или открытие VI или проекта

После запуска LabVIEW на экране открывается окно Getting Started. Это окно следует использовать для создания новых VI и проектов, выбора из последних открытых файлов LabVIEW, выбора шаблонов и примеров.

Повторно открыть окно можно командой “меню > View > Getting Started Window”.

Проекты (.lvproj), предназначены для группирования файлов LabVIEW, а также файлов других типов для построения приложений и совместно используемых библиотек.

### Лицевая панель

Окно лицевой панели является пользовательским интерфейсом виртуального прибора (ВП) Virtual Instruments (VI). На Рис. 2 приведен пример окна лицевой панели вместе с органами управления (ручки, кнопки, переключатели, движки, ...) и индикаторами (экран осциллографа, стрелочный и цифровой индикаторы, лампочка, ...), которые являются соответственно устройствами ввода и вывода VI прибора.

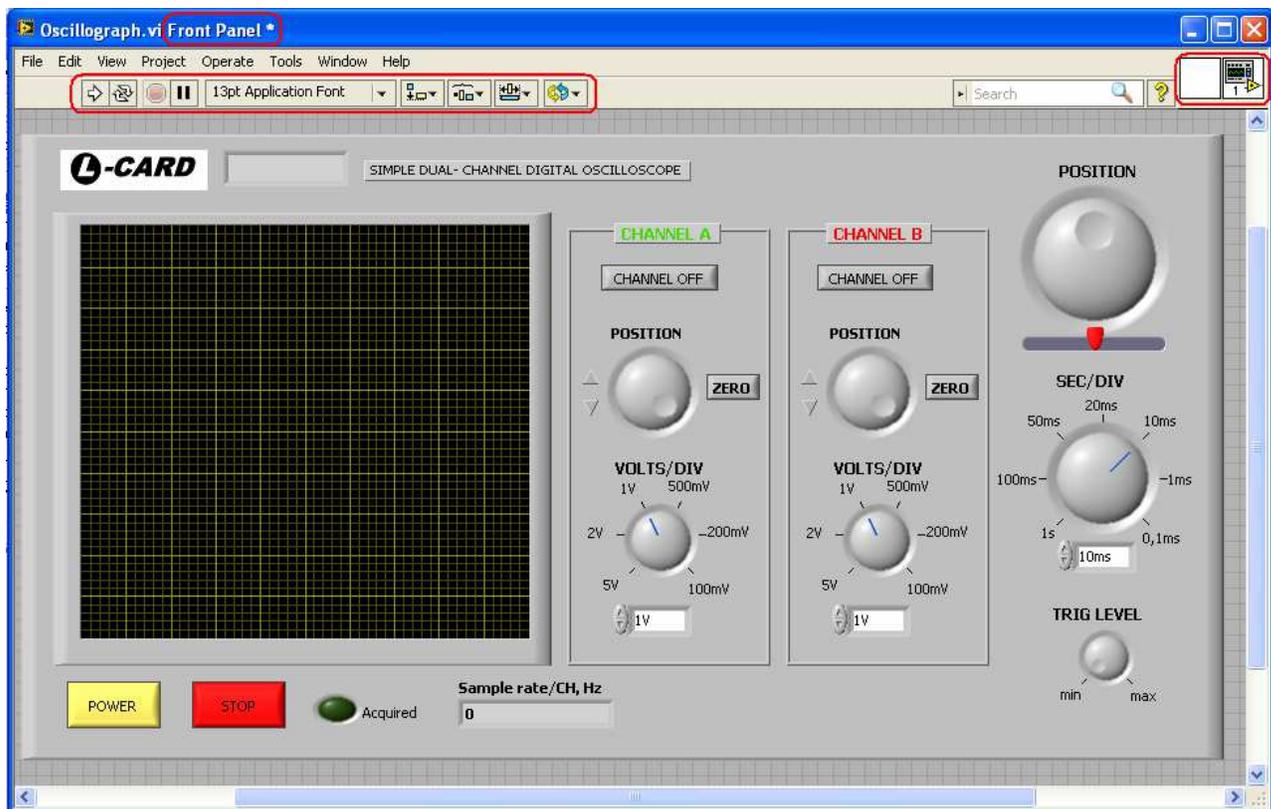


Рис. 2. Пример окна лицевой панели.

Органы управления и индикации для создания VI можно найти в палитре **Controls**. Доступ к которой осуществляется через меню > View > Controls Palette. Меню можно вызвать щелчком Правой Кнопки Мыши (ПКМ) по лицевой панели.

Все объекты LabVIEW имеют **контекстные** и всплывающие меню. Меню объекта можно вызвать щелчком ПКМ по объекту.

### Окно блок-диаграммы

При создании лицевой панели в блок-диаграмму автоматически добавляется программный код объектов лицевой панели. Блок-диаграмма - реально исполняемое приложение.

На Рис. 3 приведен пример окна блок-диаграммы. В окне блок-диаграммы содержится исходный код программы с объектами в виде

- терминалов,
- subVI,
- функций, констант (доступ: **меню > View > Functions Palette**),
- структур, а также
- проводников.

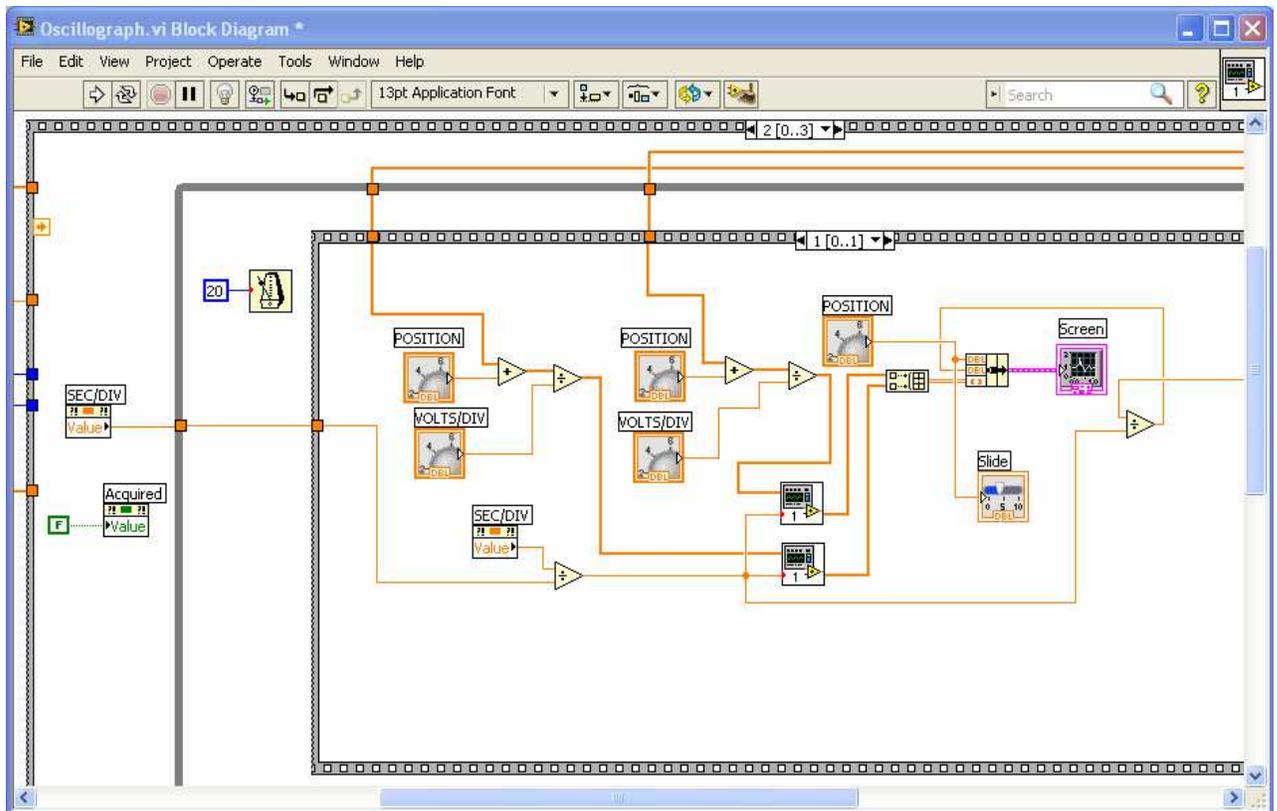


Рис. 3. Пример блок диаграммы.

**Терминалы** - это порты для обмена информацией между лицевой панелью и блок- диаграммой. Терминалы могут отображаться в виде пиктограммы или простого блока.

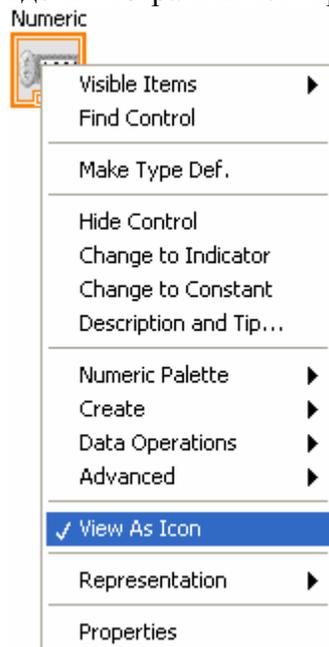


Рис. 4. Переключение внешнего вида терминала.

**Функции** отображаются только на блок-диаграммах блоками с бледно-желтым фоном.

**SubVI** это VI созданные внутри другого VI. Лицевая панель subVI открывается двойным щелчком по значку subVI.

**Проводники** служат для передачи данных между объектами блок-диаграммы. В зависимости от типов данных проводники имеют различные цвета (— целые числа, — числа с плавающей точкой, ..... логические числа, — строковые данные), стили и толщины (— скаляр, — одномерный массив, — многомерный массив).

### Значок и панель подключения subVI

Значок и панель подключения позволяют использовать и просматривать VI внутри другого VI (subVI). Чтобы использовать VI в качестве subVI, его нужно снабдить значком и панелью подключения.



Значок отображается в верхнем правом углу окна лицевой панели и окна блок-диаграммы каждого VI и является графическим представлением VI. Значок "по умолчанию" содержит число, которое показывает, сколько новых VI открыто после запуска LabVIEW.



Панель подключения subVI является набором терминалов значка, которые соответствуют органам управления и индикаторам данного VI, это похоже на список параметров функции, вызываемой в текстовых языках программирования. Доступ к панели подключения осуществляется щелчком правой кнопки мыши по значку в верхнем правом углу окна лицевой панели. В окне блок-диаграммы панель подключения открыть нельзя.

### Панель инструментов окна лицевой панели

В каждом окне есть связанная с ним панель инструментов для запуска и редактирования VI.

Панель инструментов окна лицевой панели:



(Run) запускает VI или subVI. Возможные состояния этой кнопки:



означает, что никакая программа не вызывает VI, и, следовательно, он не является subVI.



выполняющийся VI является subVI,



создаваемый или редактируемый VI содержит ошибки. Чтобы открыть список ошибок (Error list) следует щелкнуть мышью по этой кнопке.



(Run Continuously) запускает VI на исполнение до момента прерывания или приостановки. Дополнительный щелчок по этой кнопке запрещает непрерывное исполнение.



(Abort Execution) немедленно останавливает VI. Если VI используется VI более высокого уровня, кнопка становится недоступной.



(Align Objects) выравнивает объекты вдоль осей, в том числе по вертикали, по верхнему краю, по левому краю и т.д., в соответствии с выбором выпадающего меню.



(Distribute Objects) выравнивает промежутки между объектами – раздвигает, сжимает и т.д., в соответствии с выбором выпадающего меню.



(Resize Objects) выравнивает размер нескольких объектов лицевой панели, в соответствии с выбором выпадающего меню



смещает выделенные перекрывающиеся объекты в направлении Move Forward, Move Backward, Move To Front и Move To Back.

### Панель инструментов окна блок-диаграммы



(Highlight Execution) анимирует исполнение блок-диаграммы при запуске VI. Показывает движение потока данных по блок-диаграмме. Дополнительный щелчок по этой кнопке запрещает анимацию потока.



(Retain Wire Values) сохраняет значения данных в проводниках в каждой точке выполняемого потока, чтобы при установке пробника на проводник можно было получать последнее значение данных, прошедшее по проводнику. Эти последние значения доступны только, если VI успешно выполнен хотя бы один раз.



(Step Into) открывает узел и приостанавливает выполнение. Повторный щелчок по кнопке Step Into, выполнит предыдущее действие и вновь приостановит выполнение subVI или структуры. Аналогично можно нажимать на клавиши **<Ctrl>** и **<стрелка вниз>**. Пошаговое выполнение VI происходит от узла к узлу. О готовности каждого узла к исполнению свидетельствует его мерцание.



(Step Over) выполняет работу узла и приостанавливается на следующем узле. Аналогично можно нажимать на клавиши **<Ctrl>** и **<стрелка вправо>**. В этом случае узел выполняется без пошагового исполнения внутри него.



(Step Out) завершает исполнение текущего узла и приостановится. Когда VI завершает исполнение, кнопка становится недоступной. Аналогично можно нажимать на клавиши **<Ctrl>** и **<стрелка вверх>**. В этом случае завершается пошаговое исполнение в текущем узле и происходит переход к следующему узлу.

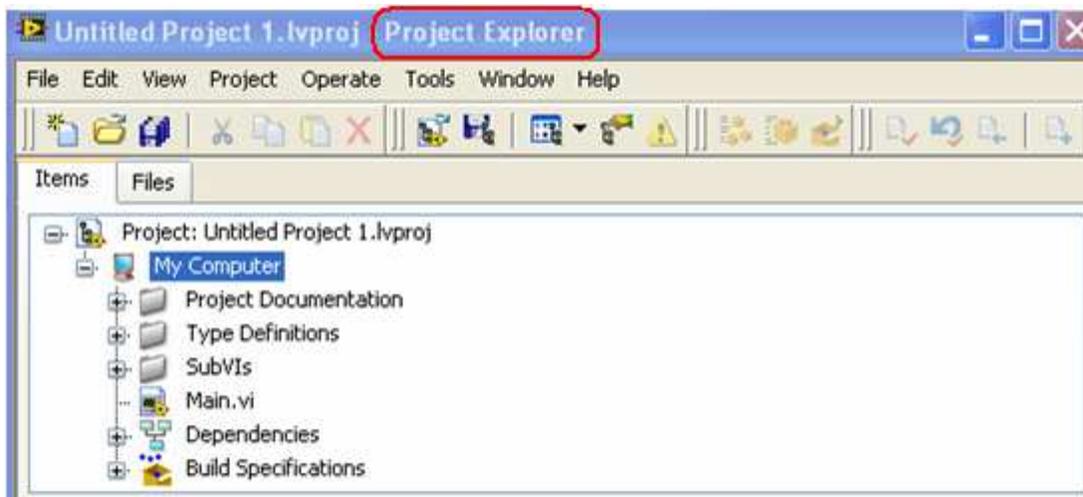


(Clean Up Diagram) автоматически перерезводит все имеющиеся проводники и упорядочивает объекты на блок- диаграмме, скомпоновав их более аккуратно. Конфигурирование параметров перекомпоновки объектов на блок-диаграмме осуществляется командой “меню > Tools > Options > Block Diagram”.

 (Warning) появляется, если в VI имеется предупреждение, а в окне Error List был установлен флажок Show Warnings. Предупреждение указывает на потенциальную проблему в блок-диаграмме, которая, однако, не приводит к прерыванию работы VI.

### Панели инструментов для проекта

Для выполнения операций в проекте LabVIEW следует использовать кнопки Standard, Project, Build и Source Control на панели инструментов. Панели инструментов доступны в верхней части окна Project Explorer (**Рис. 5**).



**Рис. 5.** Окно Project Explorer

### Выбор инструмента

С помощью инструментов LabVIEW можно создавать, модифицировать и отлаживать VI. Инструмент это специальный режим работы курсора мыши. Режим работы курсора соответствует значку выбранного инструмента. В зависимости от текущего положения мыши LabVIEW определяет, какой выбран инструмент.

Инструмент можно выбрать в палитре Tools, которая открывается командой “меню > View > Tools Palette”.



- автоматический выбор инструмента в зависимости от положения курсора



- изменение значений элемента управления. Позволяет изменять параметры ввода лицевой панели (см. рис.3 ) при неработающем VI.

-  - выделение числового элемента управления Number of Measurements. После того, как объект выделен, его можно переместить, скопировать или удалить.
-  - ввод текста в элемент управления, редактирования текста и создания свободных меток
-  - соединение объектов на блок-диаграмме между собой
-  - доступ к контекстному меню объекта с помощью левой кнопки мыши.
-  - (Scrolling) прокрутка в окнах вместо использования линейки прокрутки.
-  - (Breakpoint) установка контрольных точек в VI, функциях, узлах, на проводниках и структурах, чтобы приостановить в этой точке выполнение VI.
-  - (Probe) создание пробников на проводниках блок-диаграммы. Этот инструмент служит для проверки промежуточных значений внутри VI, который выдает сомнительные или непредусмотренные результаты.
-  - (Color Copy) копирование цветов и вставки их в инструмент Coloring.
-  - (Coloring) раскраска объекта. Этот инструмент отображает текущие настройки цвета переднего плана и цвета фона.

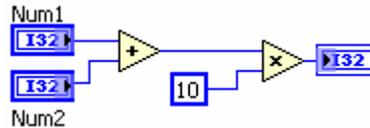
## Симуляция физических устройств NI

Можно создавать симуляторы NI-DAQmx устройств, поддерживаемых NI-DAQmx 7.4 и выше. С помощью симуляторов NI-DAQmx устройств можно опробовать изделия NI в составе ваших приложений, не имея их физически. Для симуляции необходимо поставить DAQmx на компьютер который можно бесплатно скачать с сайта NI. Симулятор устройства создается в приложении MAX. На выходе устройства при симуляции будет синус, который меняется в +/- диапазоне измерения устройства.

## Потоковое программирование

В LabVIEW VI выполняются под управлением потока данных, и узел блок-диаграммы выполняется только тогда, когда получит все требуемые входные данные. По завершении выполнения узел выдает выходные данные и передает их следующему узлу на пути распространения потока данных. Продвижение данных через узлы определяет порядок выполнения VI и функций на блок-диаграмме.

Пример потокового программирования, приведенный на рис. 6, складывает два числа и затем сумму этих чисел умножает на 10. Блок-диаграмма выполняет сначала сложение а затем умножение. Умножение не может выполняться до тех пор, пока не выполнится функция сложения, которая передаст результат блоку умножения.



**Рис. 6.** Пример программы, управляемой потоком данных: блок выполняется только тогда, когда данные доступны на всех его входах и выдает результат только после окончания выполнения.

## Модульное программирование

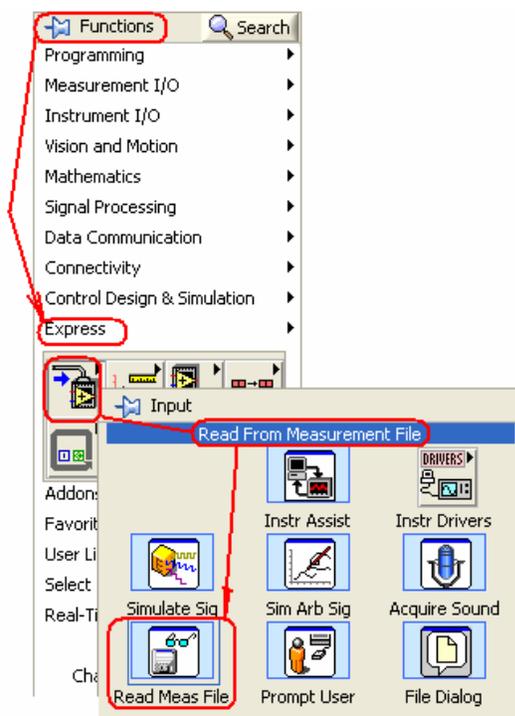
В LabVIEW модульное программирование означает создание меньших участков кода VI известных как SubVI. Это позволяет, например, большую схему сделать более компактной, заменив части схемы эквивалентными SubVI модулями, каждый из которых имеет свою переднюю панель и блок-диаграмму и отображается на схеме верхнего уровня своей пиктограммой.

Стоит отметить, что при преобразовании фрагментов VP в SubVI общая память почти не увеличивается, однако память кода верхнего уровня VP сокращается.

## Сбор и запись данных

Для решения задачи сбора данных служат следующие функции Express VI:

- DAQ Assistant  получает данные от устройства сбора данных (DAQ-устройство)
- Instrument I/O Assistant  управляет сбором данных с измерительных приборов, обычно посредством GPIB или последовательного интерфейса
- Simulate Signal и  генерирует смоделированные данные, например, синусоидальный сигнал
- Read from Measurement File  считывает файл в форматах LVM или TDM, который был создан с помощью Write To Measurement File Express VI. Read From Measurement File.



**Рис. 7.** Пример обращения к функции чтения данных из файла

В папке Function > Express > OUTPUT содержатся блоки

- Write to Measurement File Express VI – для записи данных в файл в LVM или TDMS.
- Build Text Express VI – для создания текста отображаемого на лицевой панели или для экспорта его в файл или измерительный прибор.

### **Файловый ввод-вывод**

Функции файлового ввода-вывода выполняют запись данных в файл или чтение их из файла.

Типовая операция файлового ввода-вывода включает следующие процессы:

1. Создание или открытие файла. После открытия файла его представляет уникальный идентификатор, называемый refnum (ссылка)
2. VI или функция файлового ввода-вывода читает из файла или записывает в файл.
3. Закрытие файла.

Если VI выполняет все три операции, он называется высокоуровневым VI. Однако при циклическом выводе высокоуровневые VI менее эффективны, в сравнении с низкоуровневыми VI, которые один раз создают или открывают файла, затем многократно выполняют ввод-вывод данных и завершают процесс однократным закрытием файла.

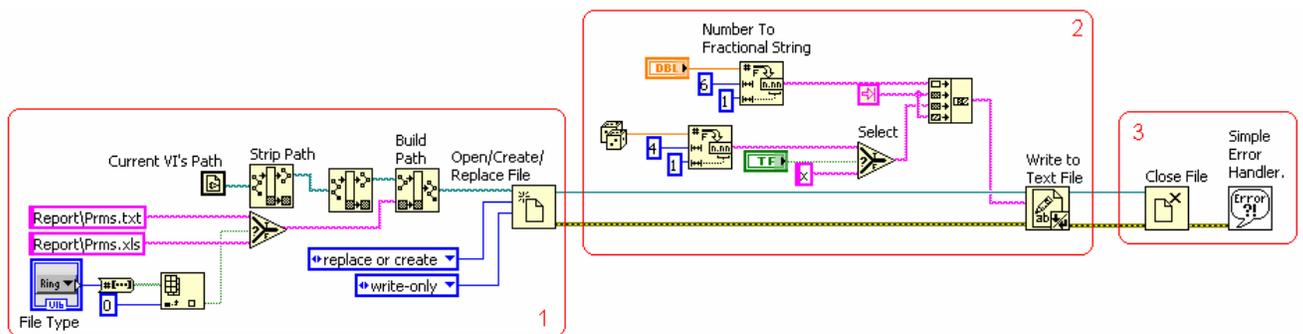
### **Форматы файлов**

В LabVIEW можно использовать или создавать файлы в следующих форматах: двоичные, ASCII, LVM и TDMS.

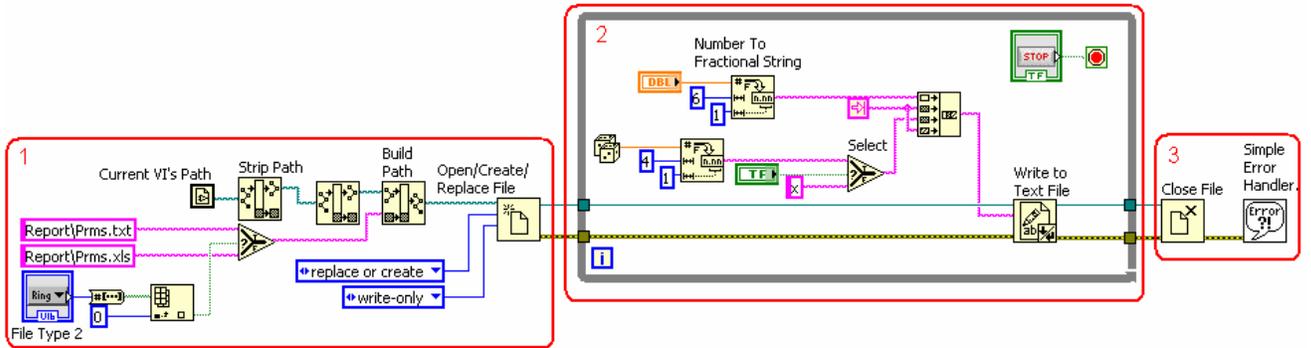
- Binary—двоичные файлы - базовый формат для всех прочих форматов.
- ASCII—Файл ASCII представляет собой особый тип двоичного файла, который является стандартом для большинства программ. Он состоит из набора ASCII-кодов. Файлы ASCII также называются текстовыми файлами.
- LVM—Файл данных измерений в LabVIEW (.lvm) – текстовый файл с табуляцией в качестве разделителей, который можно открыть в табличном или текстовом редакторе. Файл .lvm включает информацию о данных, например, дату и время их генерации. Этот формат – специальный тип ASCII-файла, созданный для LabVIEW.
- TDMS—Этот формат - специальный тип двоичного файла, созданный для продуктов National Instruments. Он состоит из двух отдельных файлов – двоичного файла, где хранятся данные и записанные свойства данных, и двоичного файла индексов, который обеспечивает сводную информацию обо всех атрибутах и указателях в двоичном файле.

**Write to Spreadsheet File** (запись в файл электронных таблиц) — преобразует двумерный или одномерный массив чисел двойной точности в текстовую строку и записывает строку в новый ASCII-файл, или добавляет строку в существующий. При этом данные можно транспонировать. VI открывает или создает файл до записи в него, а после записи закрывает. Можно использовать этот VI для создания текстового файла, читаемого большинством табличных приложений.

**Write to Measurement File** (Запись в файл измерений) — Экспресс-VI, который записывает данные в текстовый (.lvm) или двоичный (.tdms) файл измерений. Можно задать метод сохранения, формат файла (.lvm или .tdms), тип заголовка и разделитель.



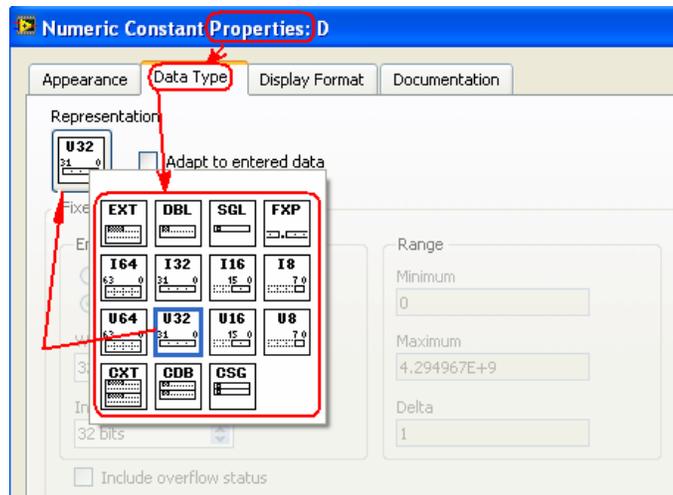
**Рис. 8.** Пример записи на диск не в режиме потока. Здесь открытие/создание/замена файла (1), запись (2) и закрытие файла (3) происходят однократно.



**Рис. 9.** Пример потоковой записи на диск. Здесь перед циклической записью (2) файл один раз открывается/создается/замещается (1) и после записи один раз закрывается (3).

### Числовые типы данных

Чтобы изменить формат представления числа, щелкните правой кнопкой мыши по элементу управления, индикатору или константе и выберите команду Representation или Properties > Data Type.

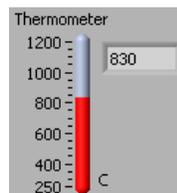


### Связываемые данные

Данные можно группировать в массивы и кластеры. Массивы объединяют данные одного типа, а кластеры – разных типов.

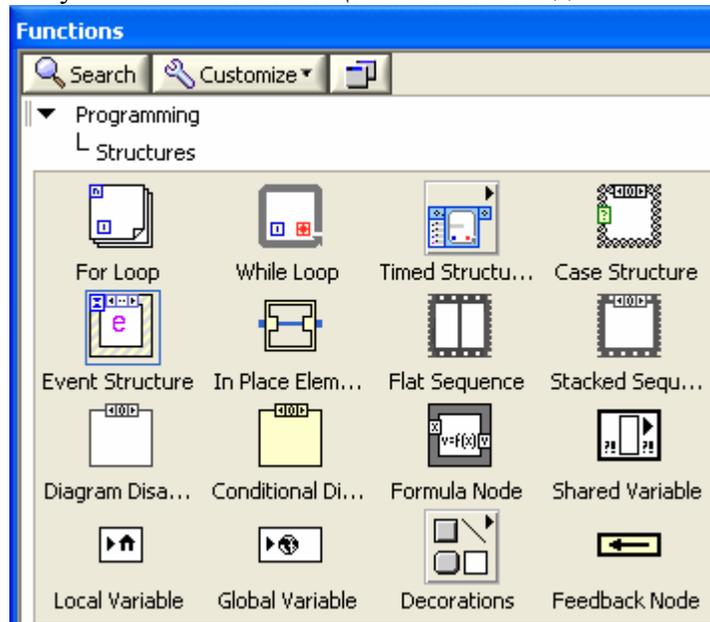
### Шкалы

В LabVIEW можно создавать пользовательские шкалы, например, для перевода выходного значения датчика в единицы измеряемого параметра.



## Циклы While

Суб-диаграмма в цикле While выполняется до тех пор, пока на вход терминала условия не поступит определенное булевское значение. Цикл While находится в палитре Structures.



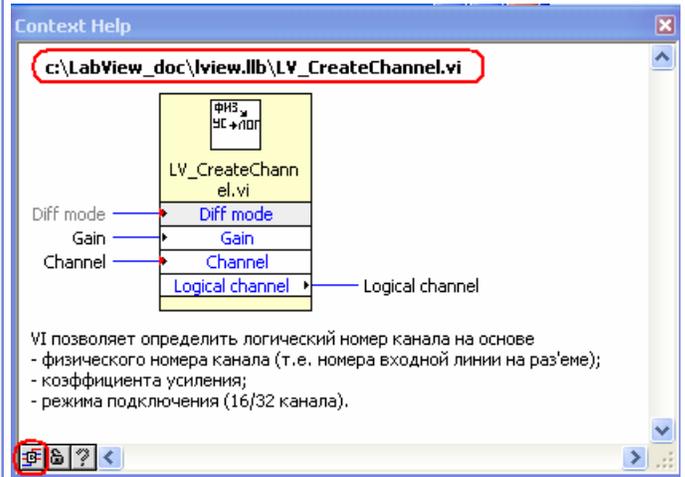
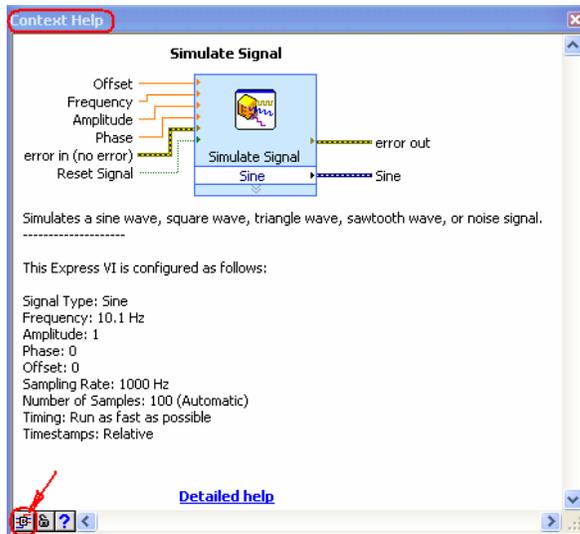
Чтобы добавить терминал условия в цикл For, щелкните правой кнопкой мыши по границе цикла и выберите из контекстного меню команду Conditional Terminal.

## Case-структуры

Case-структура состоит из двух и более суб-диаграмм или фреймов. В каждый момент времени видима только одна суб-диаграмма, и в каждый момент времени выполняется код только одного фрейма.

## Справочные утилиты LabVIEW

Когда курсор перемещается над объектами LabVIEW, основная информация о них выводится в окно Context Help, которое открывается через “меню > Help > Show Context Help” или комбинацией клавиш <Ctrl-H> или щелчком мыши по кнопке  на панели инструментов.

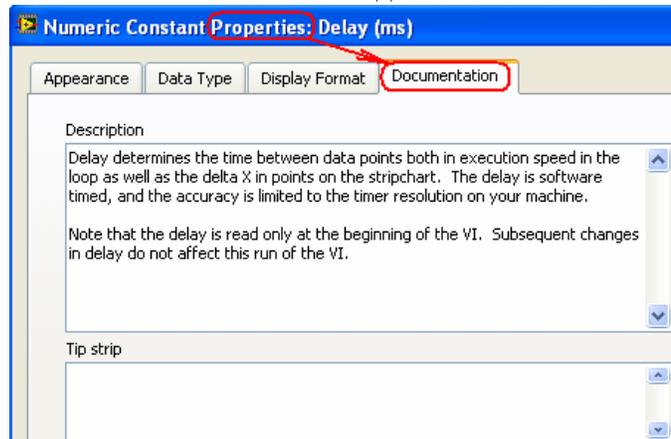


кнопка “Show Optional Terminals and Full Path”, расположенная в нижнем левом углу окна Context Help, делает видимыми дополнительные терминалы панели подключения и полный путь к VI. Дополнительные терминалы с укороченными проводниками указывают на то, что есть еще соединения. В режиме полной детализации отображаются все терминалы.

## Подсказки и описания

Подсказки – это краткие описания, которые появляются, когда вы перемещаете курсор над элементом управления или индикатором в процессе выполнения VI. Можно, например, добавить подсказку, чтобы показать, что температура выражается в градусах Цельсия или объяснить, как используется вход в реализованном алгоритме.

Чтобы создать описания VI и связать VI с HTML файлами или компилируемыми файлами справки, используйте компонент Documentation в диалоговом окне VI Properties.



## Поиск причин неисправностей VI

Предупреждения не препятствуют запуску VI в отличие от ошибок. Необходимо исправить все ошибки прежде чем VI покажет белой стрелкой запуска  готовность к работе.

Чтобы узнать, почему VI не запускается, щелкните по кнопке Run с разорванной стрелкой  или выберите команду “меню > View > Error List”. В окне Error list приведены все ошибки. Двойной щелчок по описанию ошибки выделяет область на блок-диаграмме или лицевой панели, в которой содержится ошибка.

## Техника отладки

Можно воспользоваться следующими приемами, чтобы выявить и исправить ошибки VI:

- Соединяйте входы и выходы error in и error out, расположенные внизу значков большинства



встраиваемых VI и функций.

- Чтобы увидеть все предупреждения, связанные с VI, выберите команду “меню > View > Error List” и установите флажок  Show Warnings
- Чтобы высветить весь маршрут следования проводника и убедиться, что он подключен к правильным терминалам, трижды щелкните по проводнику инструментом **Positioning** .



- Используйте окно **Context Help**  для проверки значений параметров по умолчанию для каждой функции и subVI. Если рекомендуемые или необязательные для подключения входы никуда не подключены, VI и функции получают значения по умолчанию. Например, для входа типа Boolean, если он никуда не подключен, по умолчанию может быть установлено значение TRUE.
- При исправлении ошибок используйте диалоговое окно **Find** для поиска subVI, текста и других объектов.
- Чтобы найти не подключенные subVI, выберите команду “меню > View > VI Hierarchy”. В отличие от не подключенных функций, неподключенные VI не всегда генерируют ошибки.
- Воспользуйтесь режимом подсветки выполнения , чтобы пронаблюдать продвижение данных по блок-диаграмме.
- Чтобы пронаблюдать каждое действие VI, выполняйте VI по шагам .
- Применяйте инструмент Probe  для контроля промежуточных значений данных и выходов ошибок VI и функций, особенно выполняющих операции ввода-вывода.
- Щелкните по кнопке  Retain Wire Values на панели инструментов блок-диаграммы, чтобы сохранять значения данных в проводниках для наблюдения их с помощью пробников. Это позволяет легко проверять значения данных, которые последними проходили через любой проводник.
- Для приостановки выполнения используйте контрольные точки , это позволит выполнять VI по шагам и вставлять пробники.

- Приостановите выполнение subVI, чтобы редактировать значения элементов управления и индикаторов, или управлять количеством запусков subVI или вернуться к начальной точке выполнения subVI.
- Выясните, есть ли неопределенные данные, передаваемые через функцию или subVI. Такое часто случается с числами. Например, в какой-то момент VI может выполнить деление на ноль, которое возвращает результат Inf (бесконечность), в то время, как последующие функции или subVI ожидают числа.
- Если VI работает медленнее, чем ожидается, убедитесь в том, что отключили внутри subVI подсветку выполнения. Закройте также лицевые панели и блок-диаграммы, если их не используете, поскольку открытые окна могут повлиять на быстрдействие.
- Проверьте формат представления элементов управления и индикации, чтобы увидеть, возможно ли переполнение из-за преобразования вещественного числа в целое, или целого – в более короткое целое. Например, можно подключить 16-разрядное целое к функции, которая воспринимает только 8-разрядные целые. Т.к. функция приводит 16-разрядное целое к 8-разрядному представлению, из-за этого возможна потеря данных.
- Проверьте, нет ли у вас случайно таких циклов For Loop, которые не выполняют ни одной итерации и обрабатывают пустые массивы.
- Проверьте, правильно ли проинициализированы сдвиговые регистры, кроме случая, когда, согласно вашим намерениям, они должны хранить данные, полученные в одной итерации цикла, для последующей.
- Проверьте порядок элементов кластера в точках, соответствующих источнику и приемнику данных. Во время редактирования LabVIEW обнаруживает несоответствие размеров кластеров и типов данных, однако несоответствие элементов одного и того же типа не обнаруживается.
- Проверьте порядок выполнения узлов.
- Убедитесь в том, что VI не содержит скрытых subVI. subVI могли случайно спрятать, наложив один на другой, или уменьшив размер структуры, скрыв subVI под структурой.
- Сопоставьте список subVI, используемых VI, с результатами выполнения команд “меню > View > Browse Relationships> This VI’s SubVIs” и “View > Browse Relationships > Unopened SubVIs”, чтобы определить, имеются ли какие-нибудь лишние subVI. Откройте также окно VI Hierarchy, чтобы увидеть subVI, входящие в состав VI. Избежать некорректных результатов из-за скрытых VI поможет задание входов этих VI обязательными для подключения.

### Пошаговое выполнение

Пошаговый режим выполнения VI позволяет видеть результат каждого действия на блок-диаграмме. Кнопки панели инструментов **блок-диаграммы** для пошагового выполнения:  (Step Into),  (Step Over),  (Step Out).

Когда VI исполняется в пошаговом режиме, мерцание узлов показывает, что они готовы к выполнению. Если выполнять VI в пошаговом режиме с подсветкой (анимацией) , на значках выполняющихся subVI появляется специальный знак.

Воспользуйтесь инструментом Probe (пробником)  для контроля промежуточных значения данных, передаваемых по проводникам в процессе работы VI.

Можно контролировать промежуточные значения данных в проводнике во время работы VI с помощью пробника общего назначения , индикатора из палитры Controls , специального (Supplied) пробника, адаптируемого пользователем специального пробника или путем создания нового пробника.

Воспользуйтесь инструментом  Breakpoint, чтобы устанавливать контрольные точки на VI, узлы или проводники, на которых нужно приостановить выполнение VI.

Примечание: Для наблюдения анимационной последовательности выполнения блоков диаграммы используйте режим  (Highlight Execution) который необходимо включить перед запуском виртуального прибора.

## ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

**Задание 1.** Построение виртуального осциллографа.

1. В среде LabVIEW откройте библиотеку модуля E14-440 “[Iview.lib](#)”
2. Рассмотрите структуры виртуальных приборов библиотеки.
3. Используя пример виртуального прибора “Oscillograph.vi”, отображающего входные сигналы интерфейсной карты E14-440, разработайте двухлучевой виртуальный осциллограф отображающий сигналы смоделированные блоком “Simulate Signal”:



Осциллограф должен обеспечивать следующие функции:

- Включение/выключение луча.
- Изменение усиления по вертикали (отношением В/Дел.).
- Изменение развертки (отношением Сек/Дел.).
- Смещение луча по вертикали.
- Возвращение луча в “нулевое” положение по вертикали.

- Смещение лучей по горизонтали.
- Синхронизация развертки осциллографа (уровнем триггера).
- Выключение осциллографа.

Примечание: Пример виртуального осциллографа LabVIEW показан на Рис. 10 и Рис.11.

4. Проверьте работу осциллографа меняя формы и параметры смоделированных сигналов осциллографа.
5. Наблюдайте движение потоков данных на блок-диаграмме осциллографа. Для этого перед запуском VI включите режим  (Highlight Execution).
6. Найдите узлы блок-диаграммы соответствующие элементам лицевой панели.
7. Внося изменения в блок-диаграмму и лицевую панель осциллографа наблюдайте соответствующие изменения в работе виртуального прибора.
8. Наблюдайте работу прибора в режиме отладки: выполнение по шагам и до точек останова (breakpoints)  с выводом текущих значений заданных переменных .

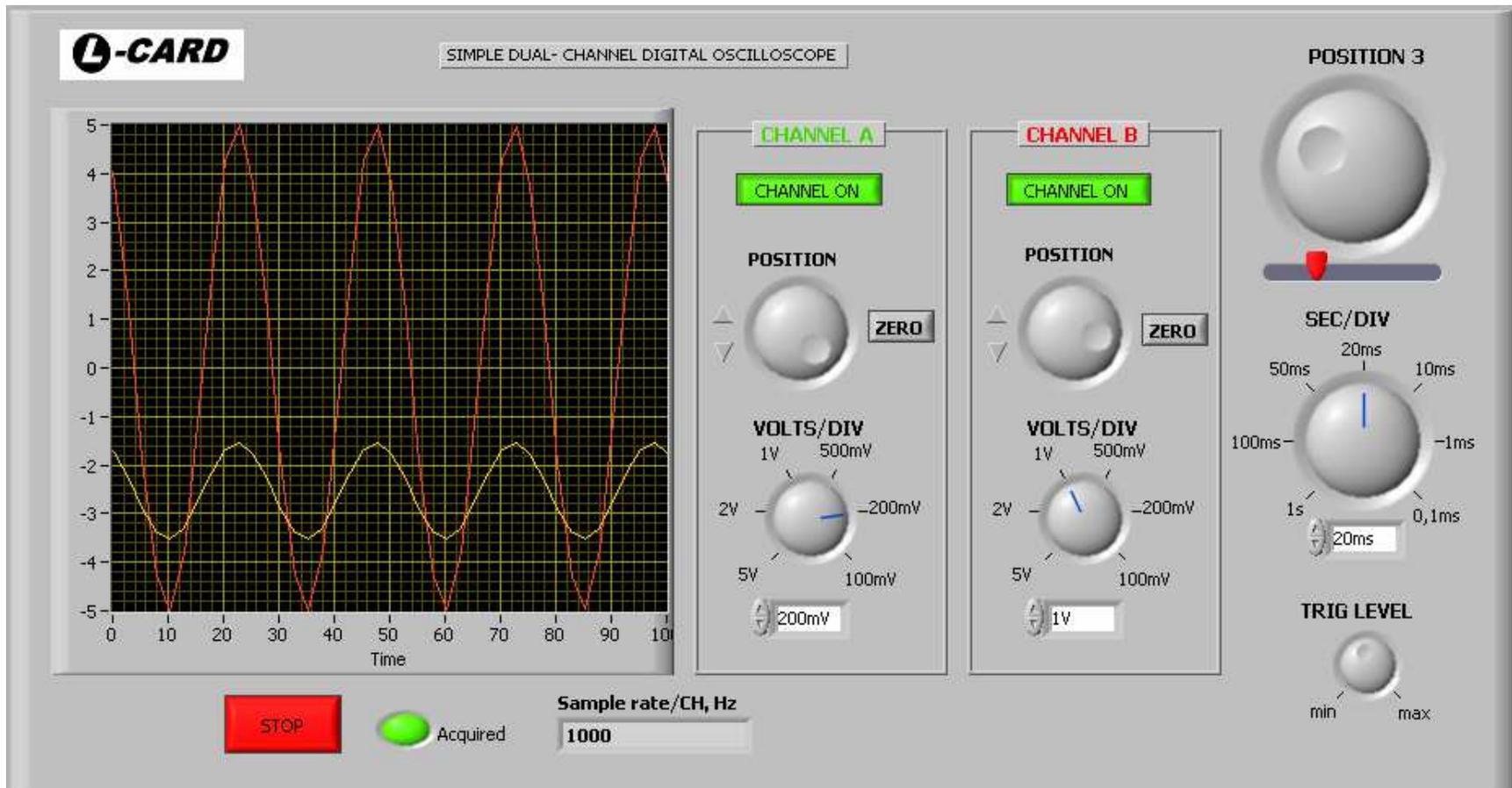


Рис. 10. Пример виртуального осциллографа.

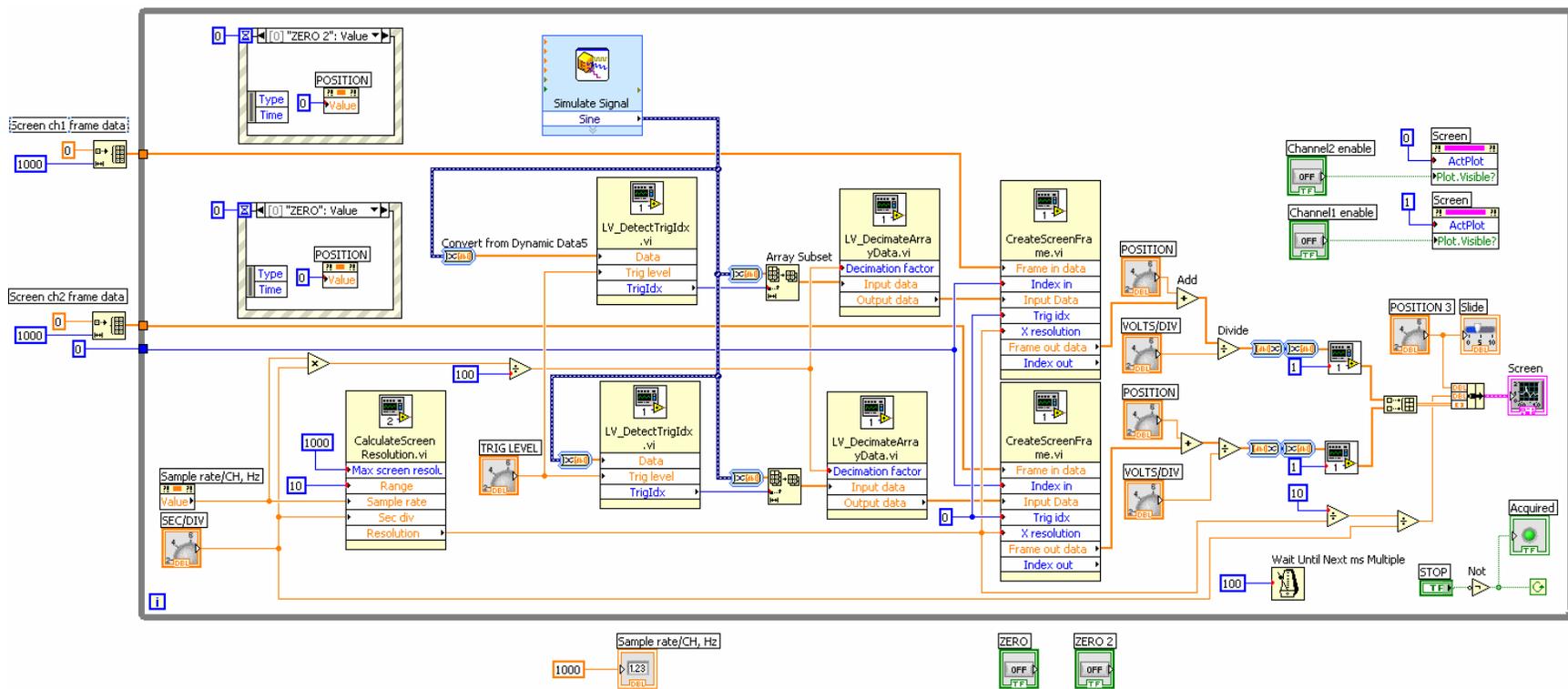


Рис. 11. Блок схема виртуального осциллографа.

**Задание 2.** Запись данных в txt и xls файлы.

1. Используя примеры (Рис. 8 и Рис.9) разработайте программы однократной и потоковой записи данных в txt и xls файлы.
2. Разработайте пример чтения файла и отображения его данных на виртуальном осциллографе.

### **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Для чего предназначен пакет LabVIEW?
2. Сравните MatLAB и LabVIEW.

### **БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

1. Википедия. LabVIEW.
2. Основы LabVIEW. Учебное пособие. Программное обеспечение курса версии 2009 г. Издание – октябрь 2009.
3. Работа с модулями фирмы L-CARD в среде LabVIEW. ЗАО Л-Кард. Современные средства измерения и контроля. [www.lcard.ru](http://www.lcard.ru)