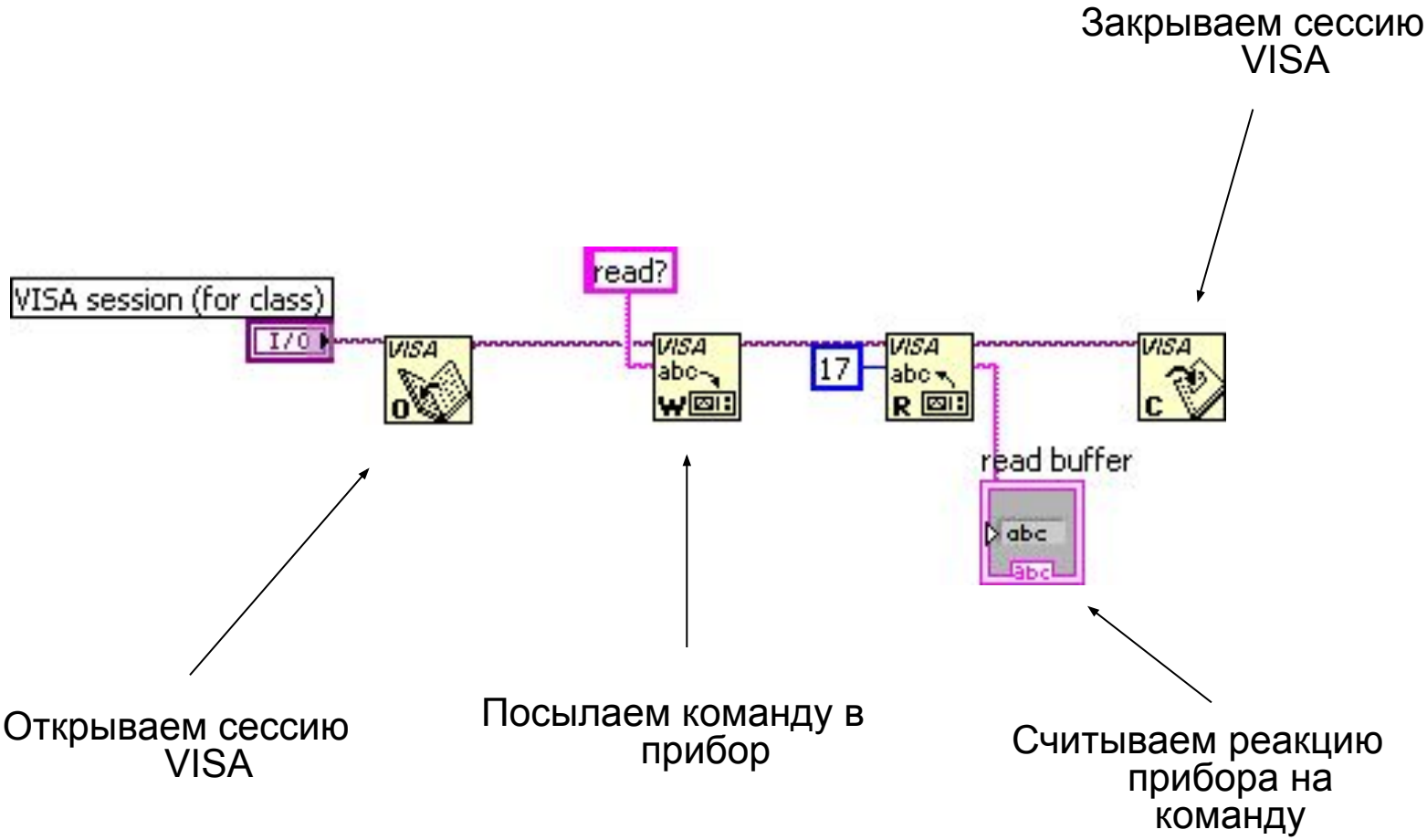
http://www.labview.webhost.ru
www.labview-ifit.narod.ru/

LABVIEW

А.Ф. Шевчун

Институт физики твердого тела РАН, Черноголовка

Проведем измерение прибором, используя язык LABVIEW.



Программа Labview – это виртуальный прибор (Virtual Instruments, VI)

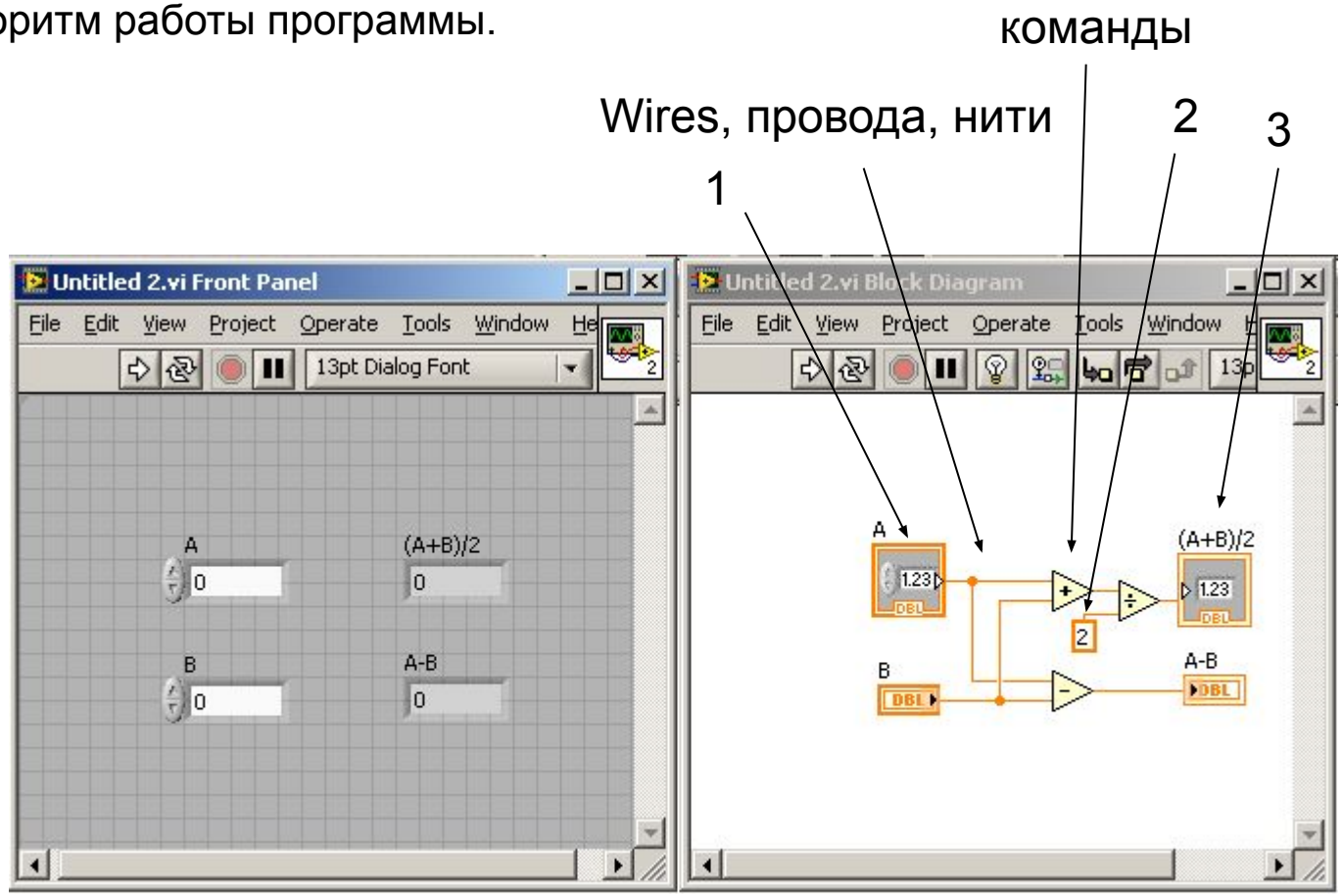
Этот виртуальный прибор состоит из «лицевой панели» (**Front Panel**) и «внутренней схемы» (**Block Diagram**).

Переход между ними через “Window>>Show Block Diagram”, или Ctrl+E, или двойной клик по переменной.

Front Panel — для элементов управления программой — кнопки, графики и т.д.

Block Diagram— алгоритм работы программы.

1 - пременная Control
2 - константа
3 - переменная Indicator



Tools Palette

Создавать, редактировать и отлаживать VI можно с помощью инструментов, расположенных на этой палитре. Палитра доступна и на лицевой панели, и на схеме.
Чтобы открыть палитру, откройте View>>Tools Palette.

Каждому действию соответствует свой вид курсора.

курсор для изменения значения переменной

курсор для выделения переменной

курсор для записи

курсор для провода (Connect Wire)

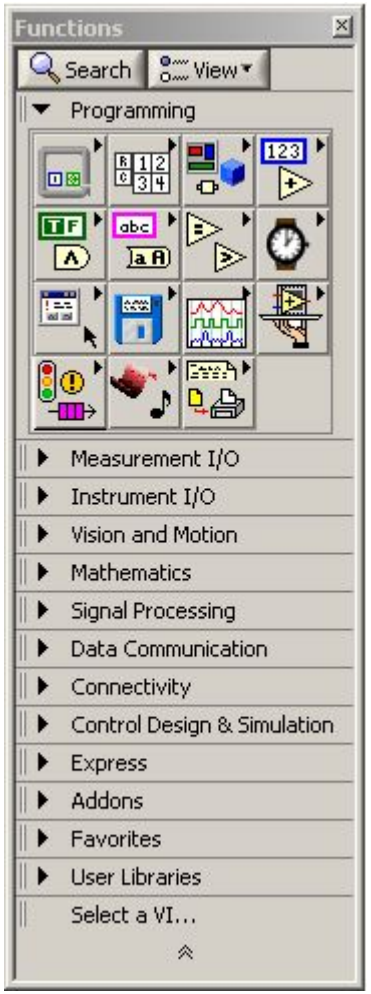
Если включен режим автоматического выбора инструментов (горит зеленая лампочка), то при наведении курсора мыши на различные объекты LabVIEW автоматически выберет нужный инструмент. Этот режим можно отключить (горячая клавиша - Shift-Tab). Если режим отключен, то выбор нужного инструмента происходит вручную на палитре, или нажатием клавиши пробел (циклический выбор).

Toolbar – запуск, циклический запуск, стоп, пауза

пошаговое исполнение программы

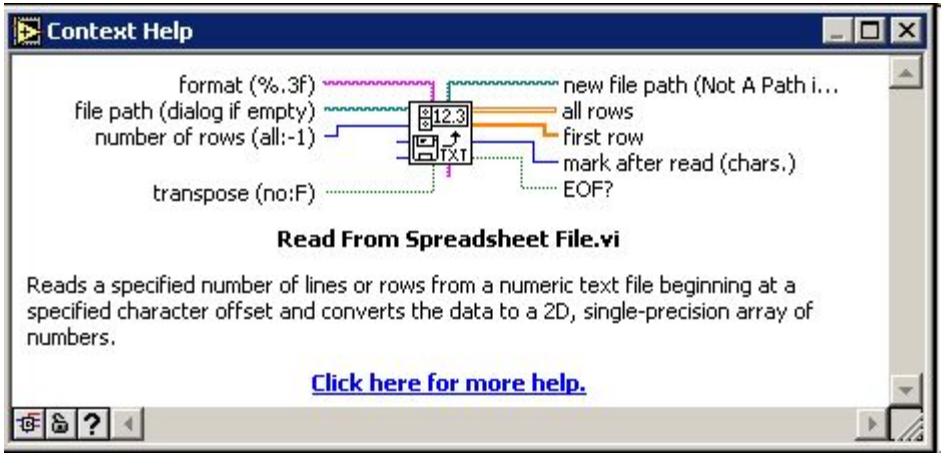
Controls Palette: позволяет задать переменные из **Front Panel**

Functions Palette: сборник команд в **Block Diagram**



Окно контекстной помощи (Context Help Window)

Чтобы сделать его видимым, нужно нажать сочетание клавиш Ctrl-H.
Показывает входящие и выходящие терминалы команды.
Подробно - "Click here for more help"



Поиск примеров (NI Example Finder)

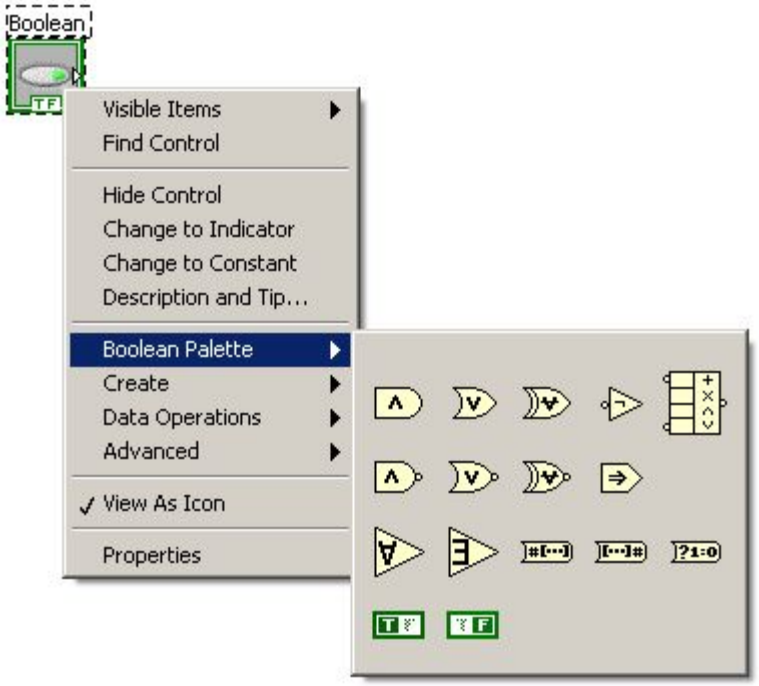
При запуске LabVIEW нажать Open->Example

Быстрые клавиши

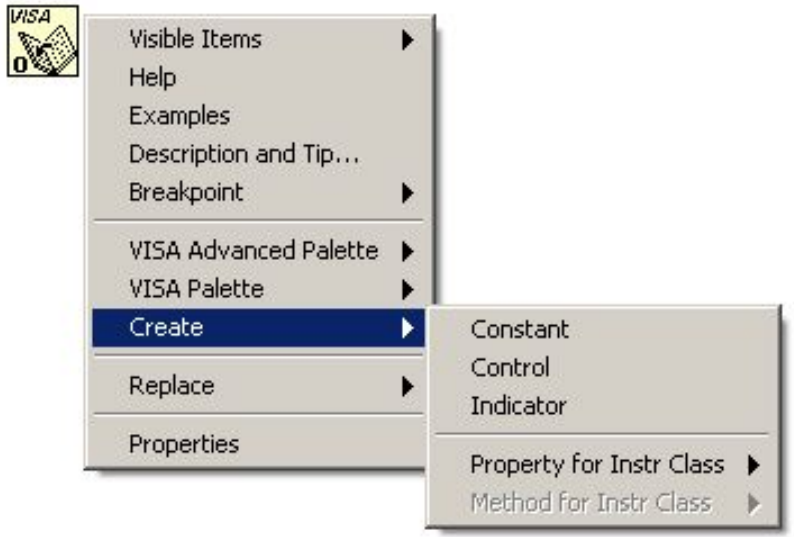
Ctrl-R - Запуск VI.
Ctrl-E - Переключение между лицевой панелью и блок-схемой.
Ctrl-H - Включает показ окна контекстной помощи.
Ctrl-B - Убирает все "испорченные" связи на блок-схеме.
Ctrl-F - Находит все VI, глобальные переменные, функции, текст или другие объекты в памяти или в определенном списке VI.

Задача №1

Правая кнопка мыши в Block Diagram

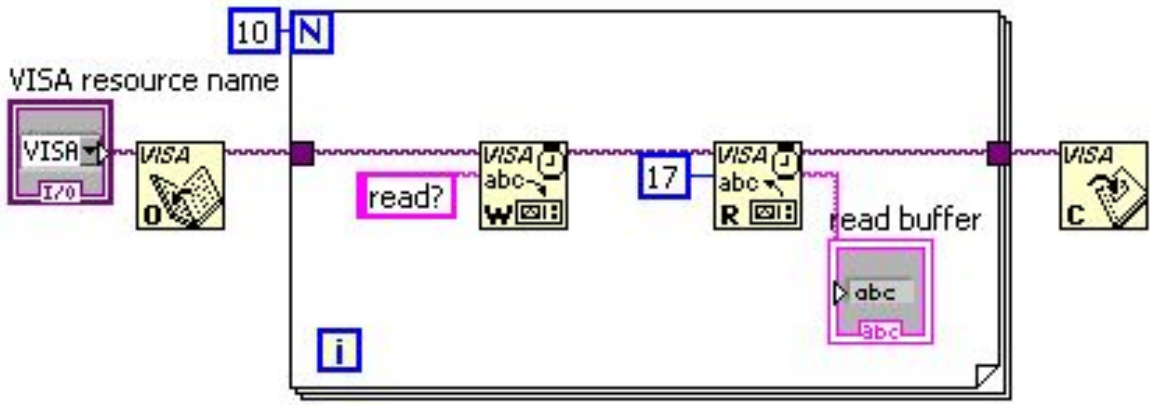


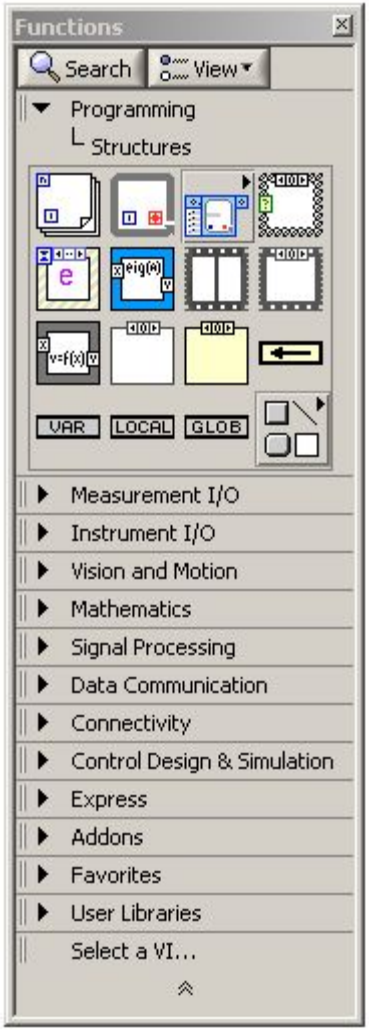
Позволяет добавлять команды без Functions Palette



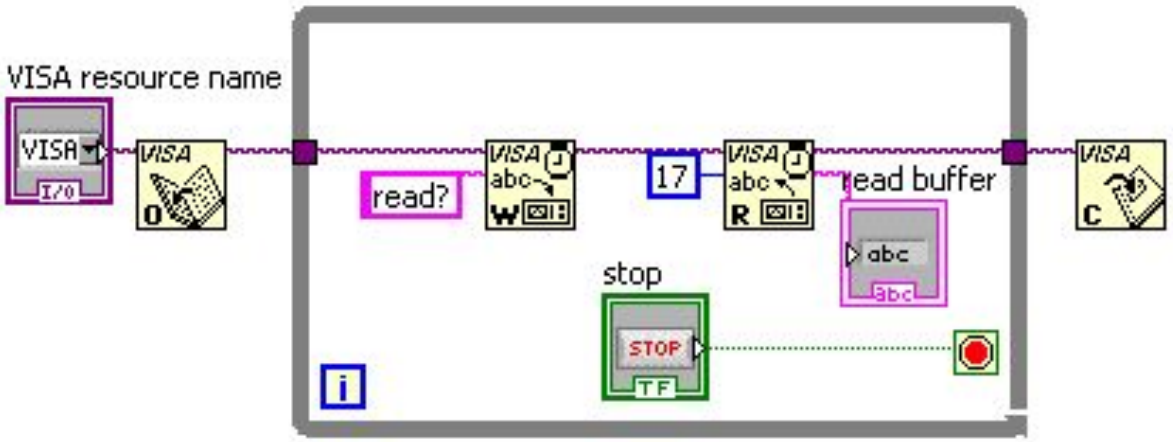
В режиме курсора Connect Wire позволяет создавать для терминалов соответствующие переменные

Цикл For

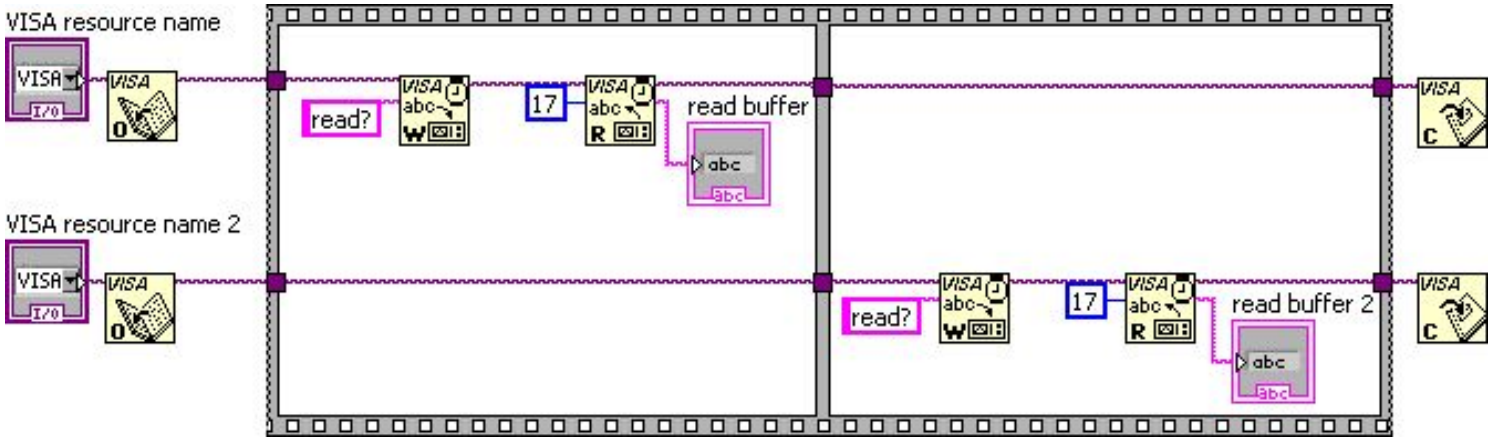


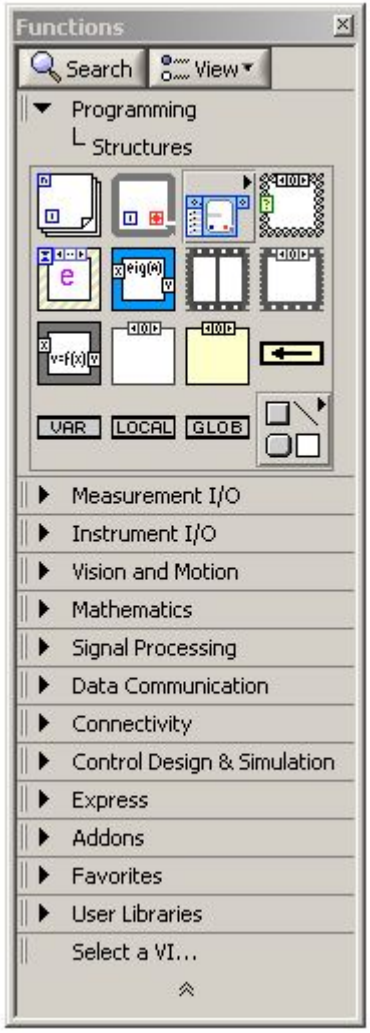


Цикл While

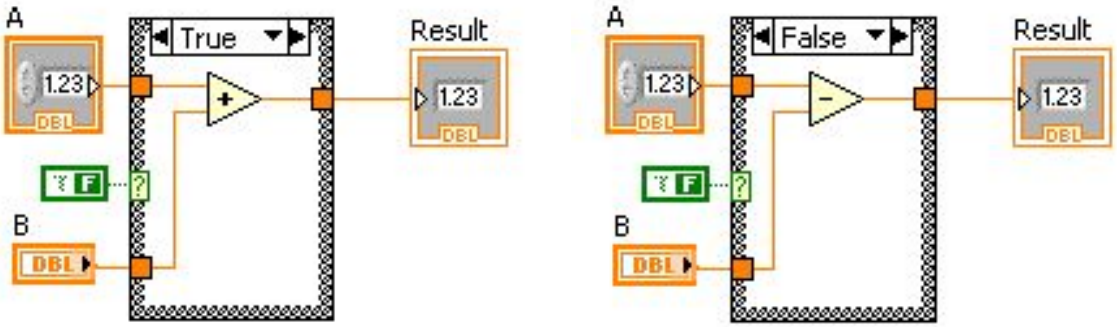


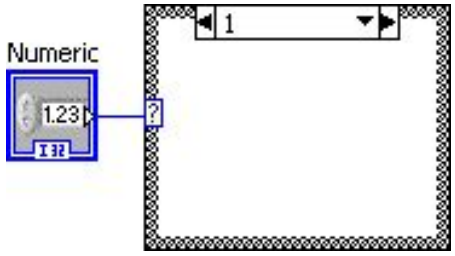
Sequence Structure (Последовательность)





Case Structure (Структура выбора)



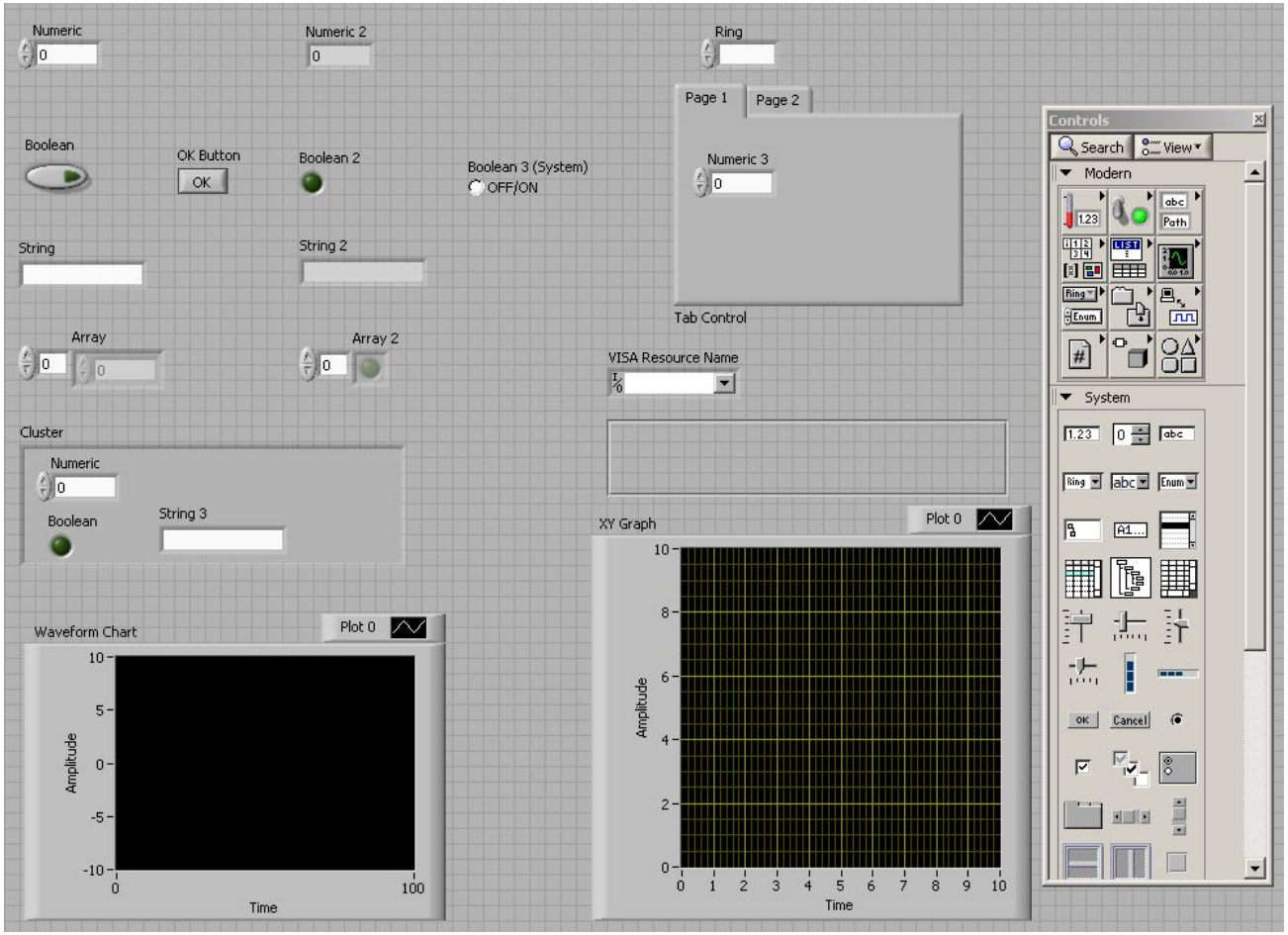


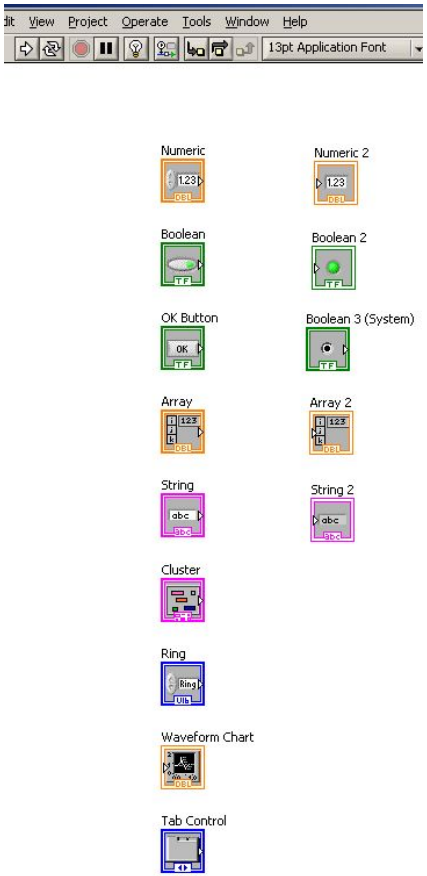
Задачи №2-5

Переменные, которые можно создать в Controls Palette

Переменные могут быть в двух состояниях: Control, Indicator.
Но это деление условное.

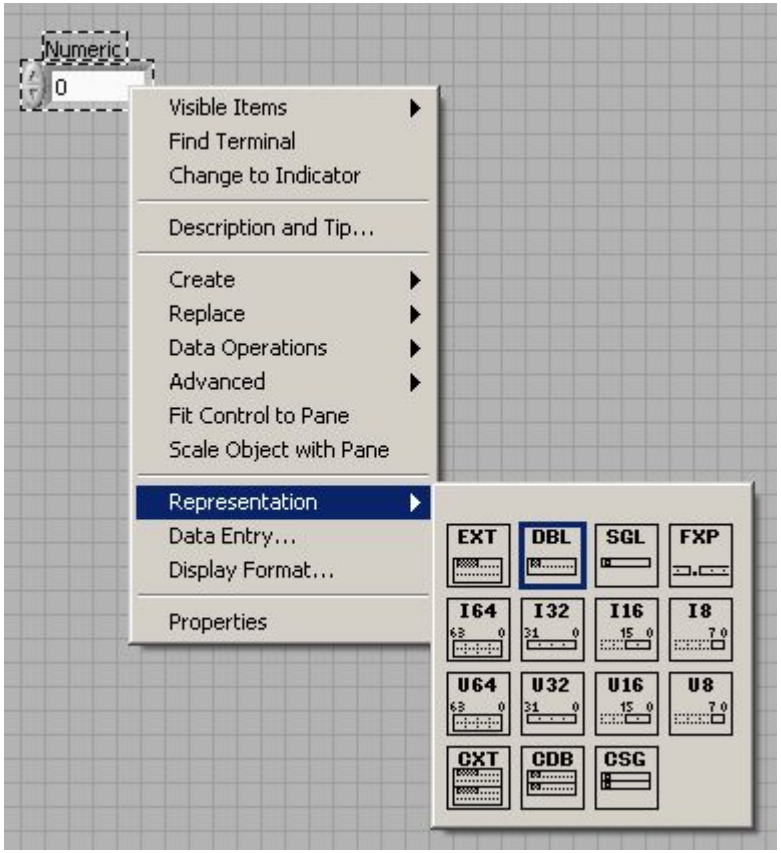
Переменные бывают разных типов: Numeric, Boolean, String, Path, Array, Cluster, Graph, Ring и т.д.
В Block Diagram от типа зависит цвет иконки переменной.



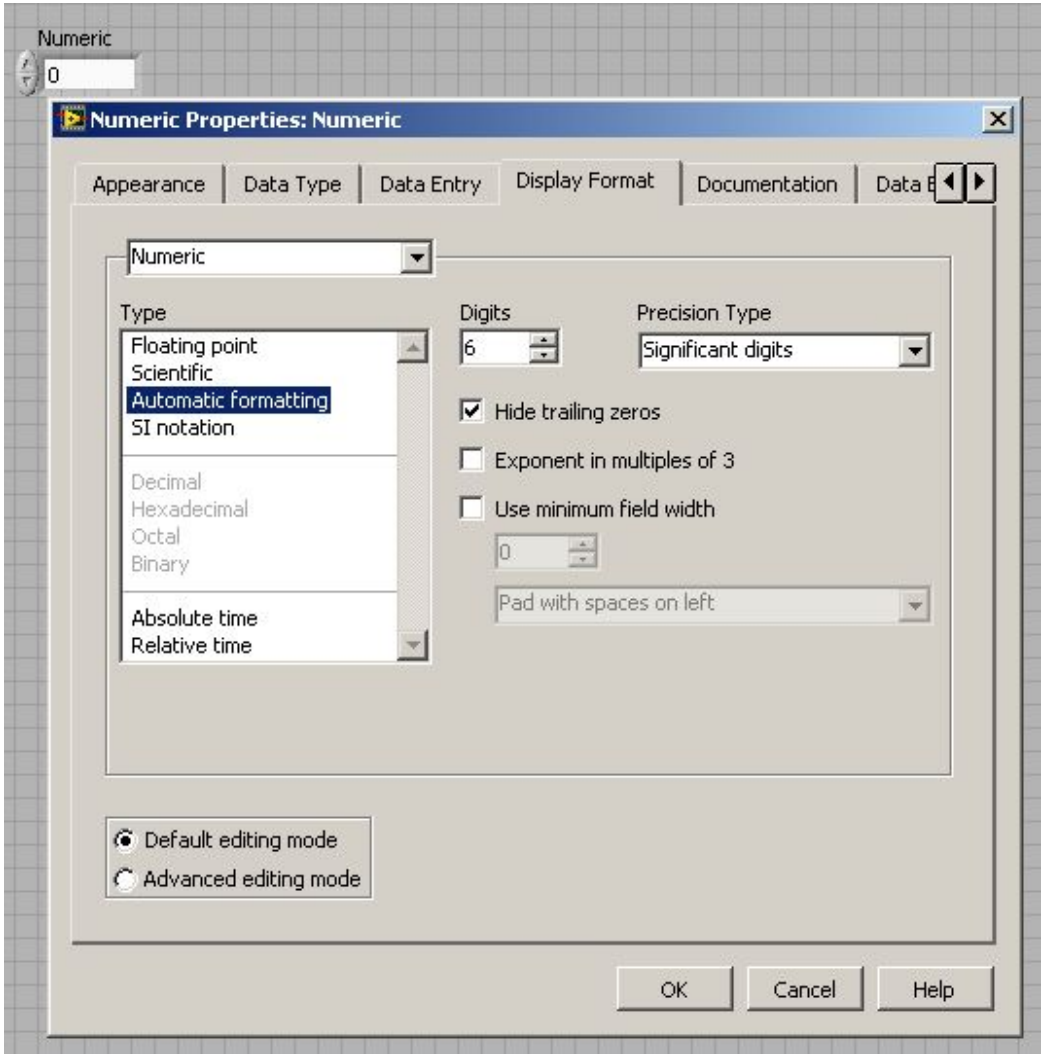


Подробно о переменной типа Numeric (правая кнопка мыши во Front Panel)

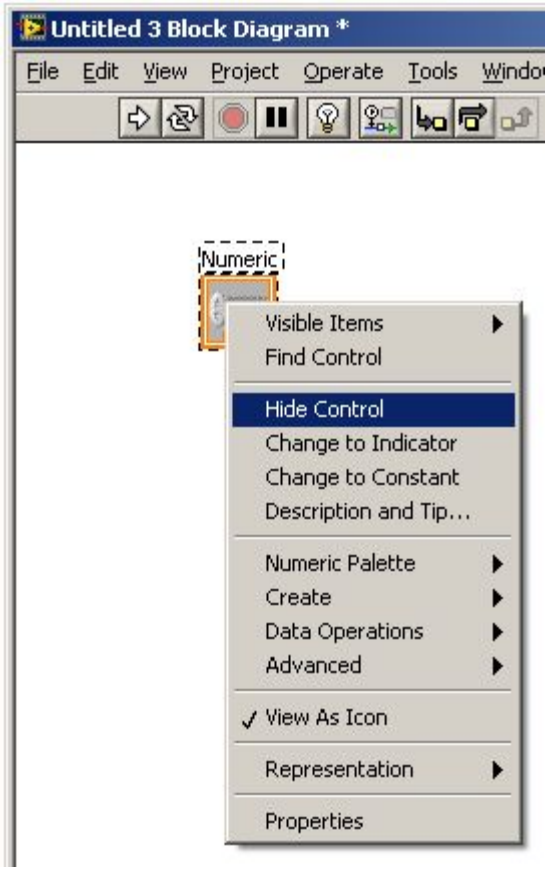
Диапазон значений переменной:



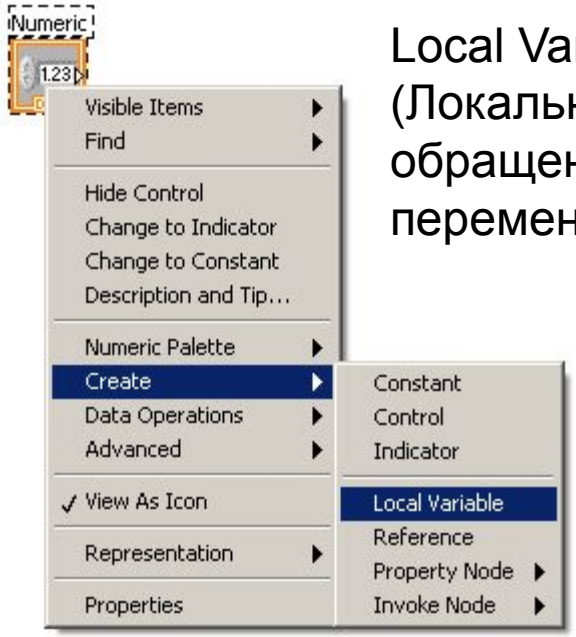
Как переменная отображается на Front Panel:



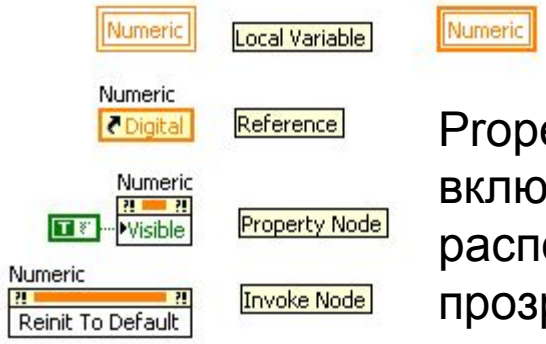
Подробно о переменной типа Numeric (правая кнопка мыши в Block Diagram)



Hide – переменная не будет видна в **Front Panel**



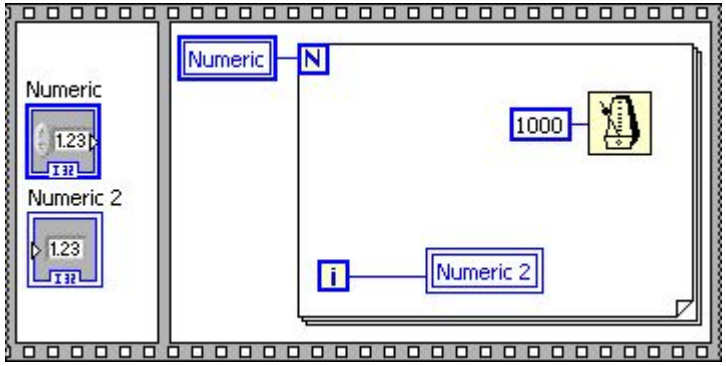
Local Variable (Локальная переменная): обращение к значению переменной

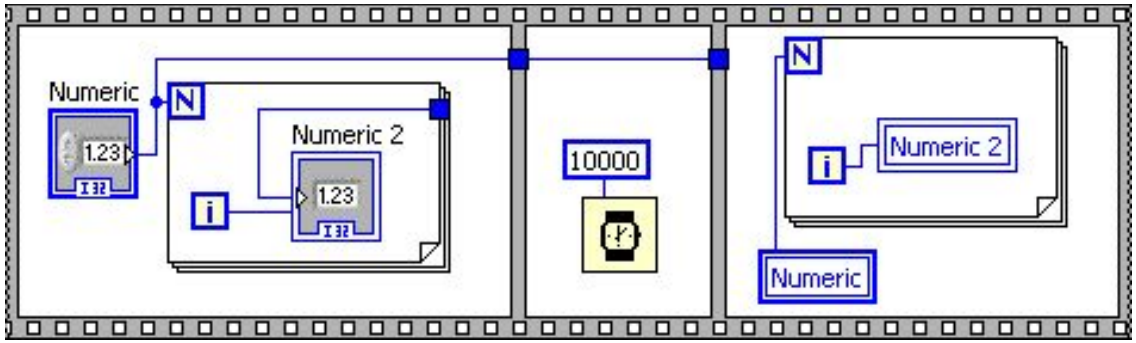


Property Node (свойства): включают цвет, расположение, размер, прозрачность переменной

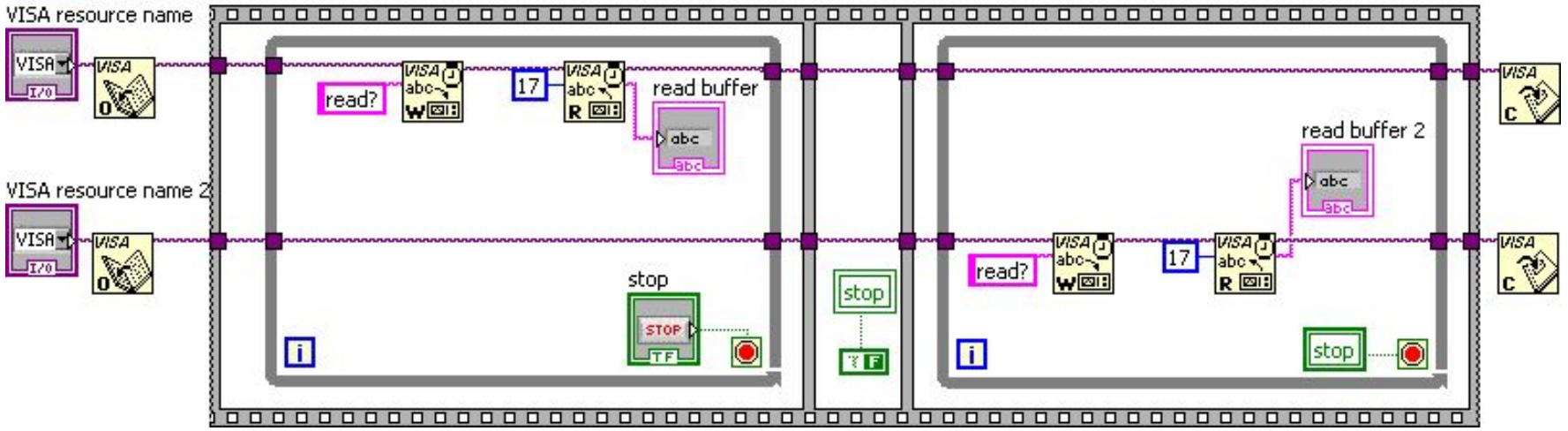
Invoke Nodes вызывает действие над объектом

Local Variable (Локальная переменная)

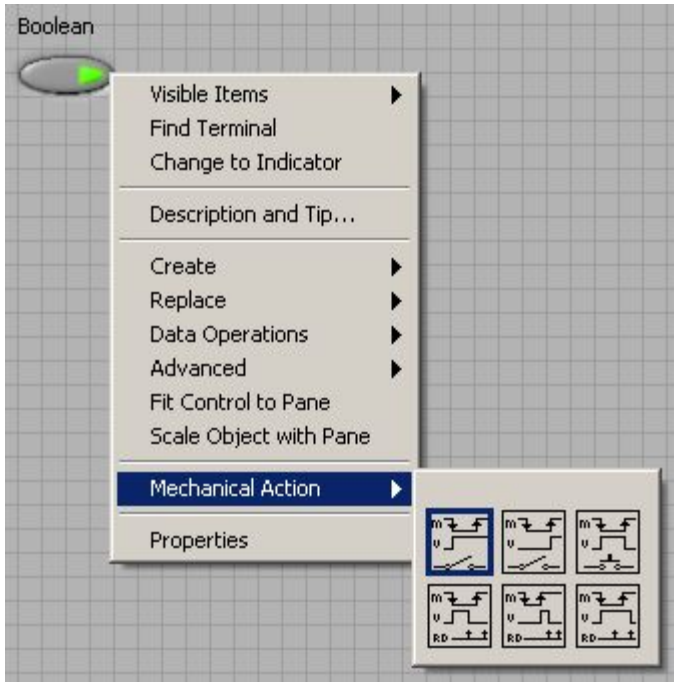




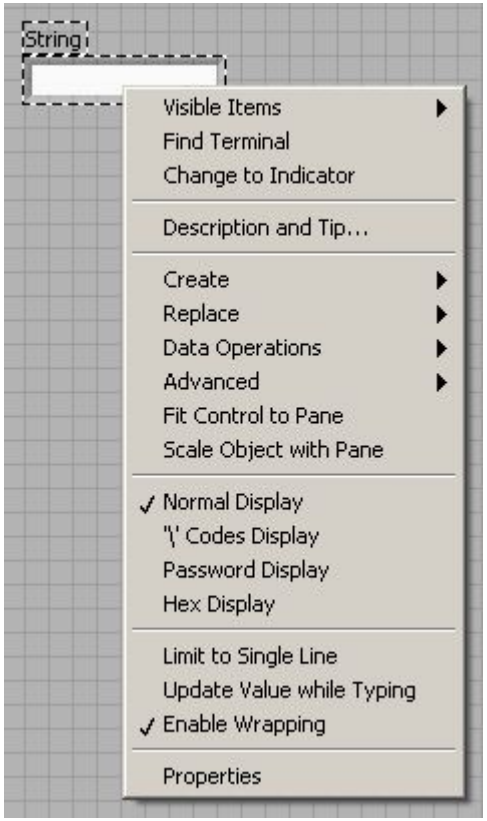
Локальная переменная позволяет в месте установки обратиться к переменной, а также считать или записать в переменную независимо от ее состояния Control или Indicator.



Подробно о переменных типа Boolean и String

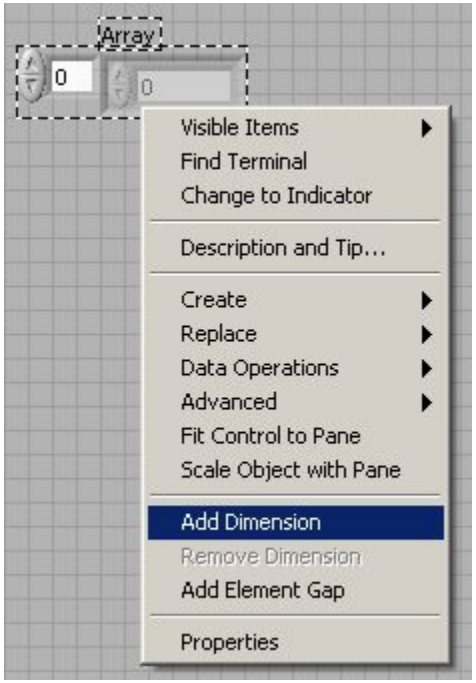


Переменная может работать в режиме переключателя или клавиши, менять свое значение в начале или конце нажатия

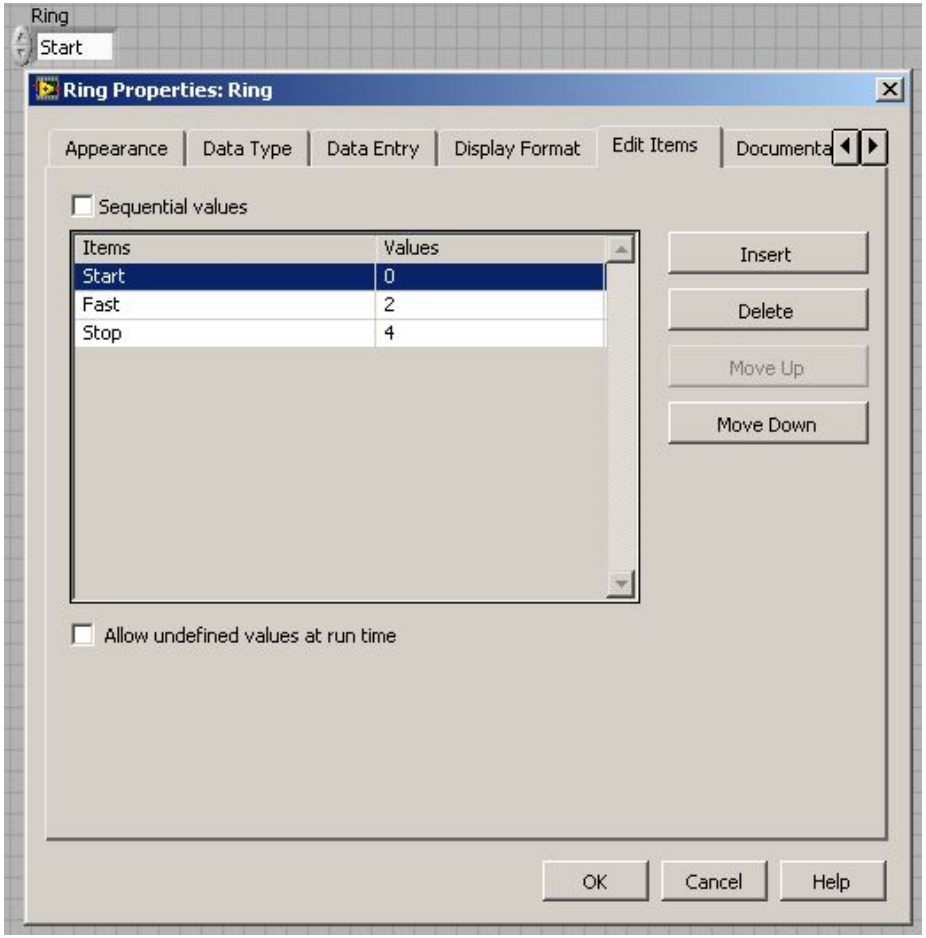


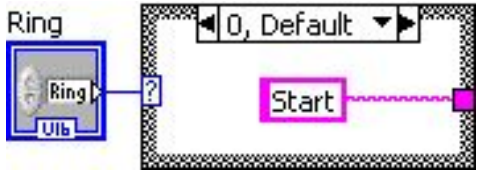
Изображение строковой переменной

Массив и изменение его размерности



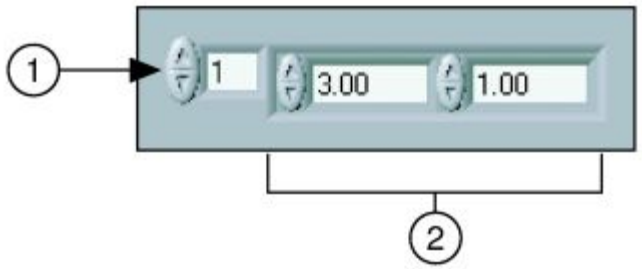
Заполнение переменной типа Ring



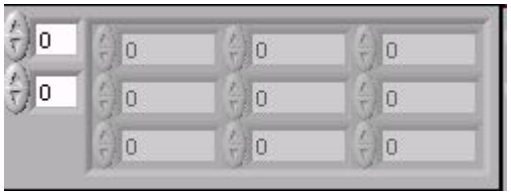


Задачи №6-9

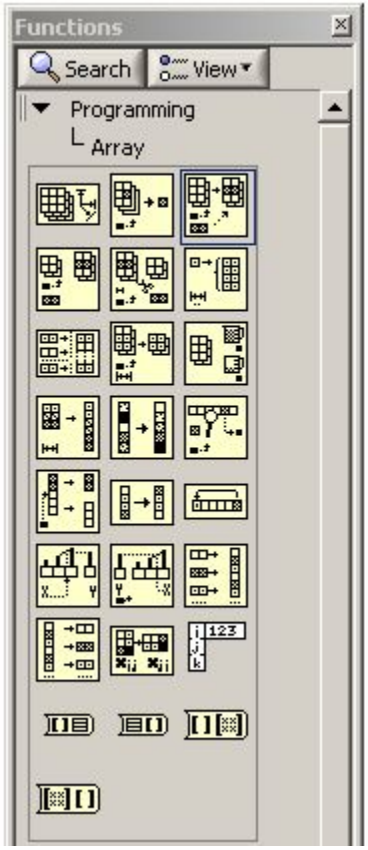
Array (Массив)

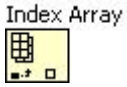


Массив



Команды для работы с массивами



Build Array		Создать массив
Array Size		Размер массива
Index Array		Извлечь элемент массива
Insert Into Array		Добавить в массив

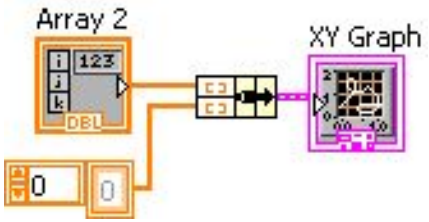
Переменная	Скалярная величина	Одномерный массив	Двумерный массив
Аналоговая			
Дискретная			
Строка			

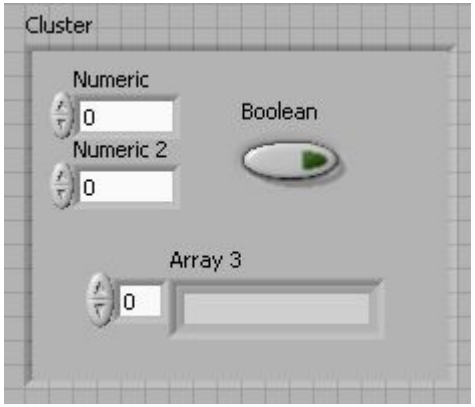
Кластер

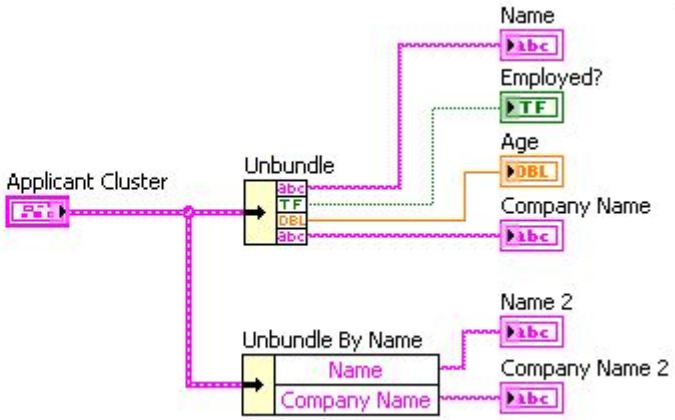
Кластер - это объединение элементов разных типов.

Кластеры используют для уменьшения количества связей на Block Diagram.

Сборка кластера - функция Bundle

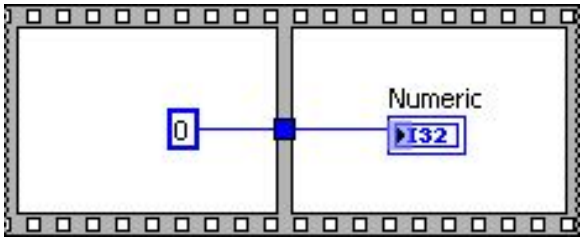




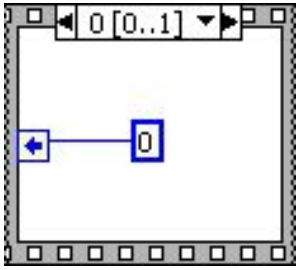


Разборка кластера:
Unbundle
Unbundle By Name

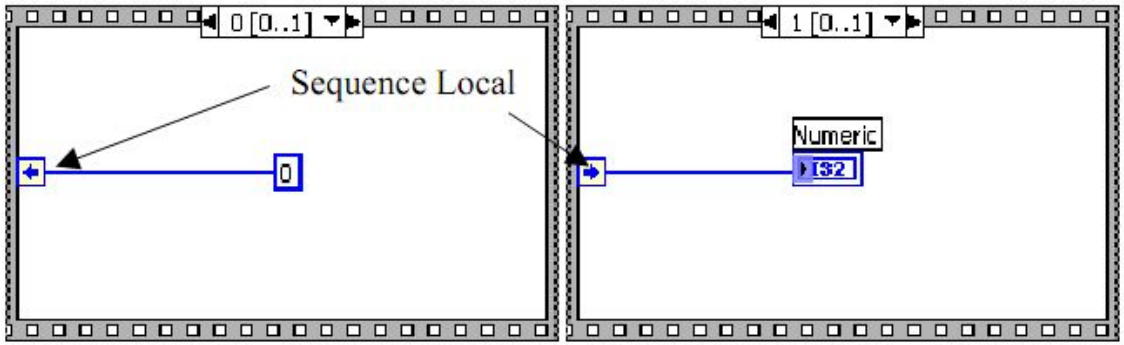
Flat Sequence Structure (Открытая последовательность)



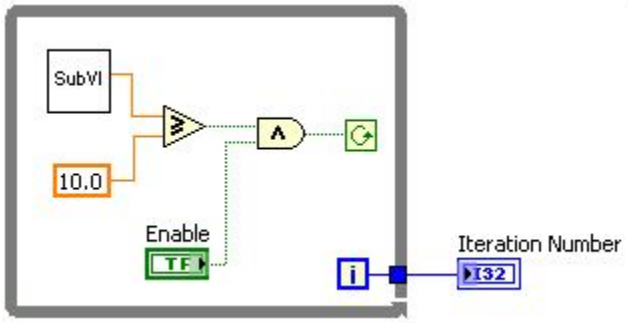
Stacked Sequence Structure (Последовательность)



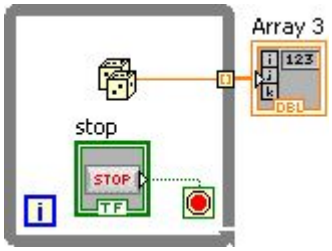
Для передачи значения из кадра в кадр приходится создавать **Sequence Local**:



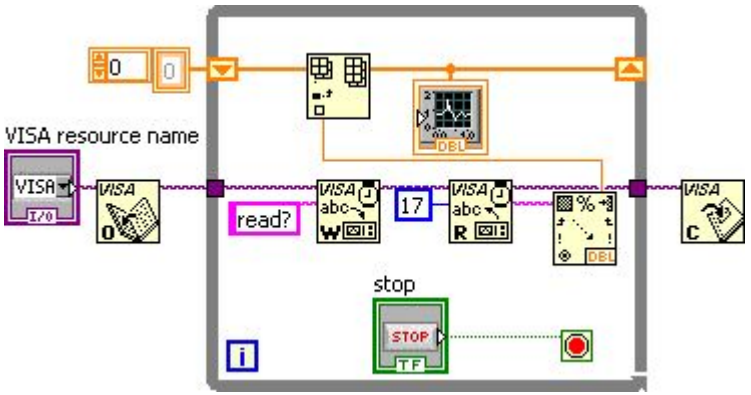
Туннели – выход из цикла



Disable indexing – индексирование запрещено.
Из цикла будет выведено последнее значение.

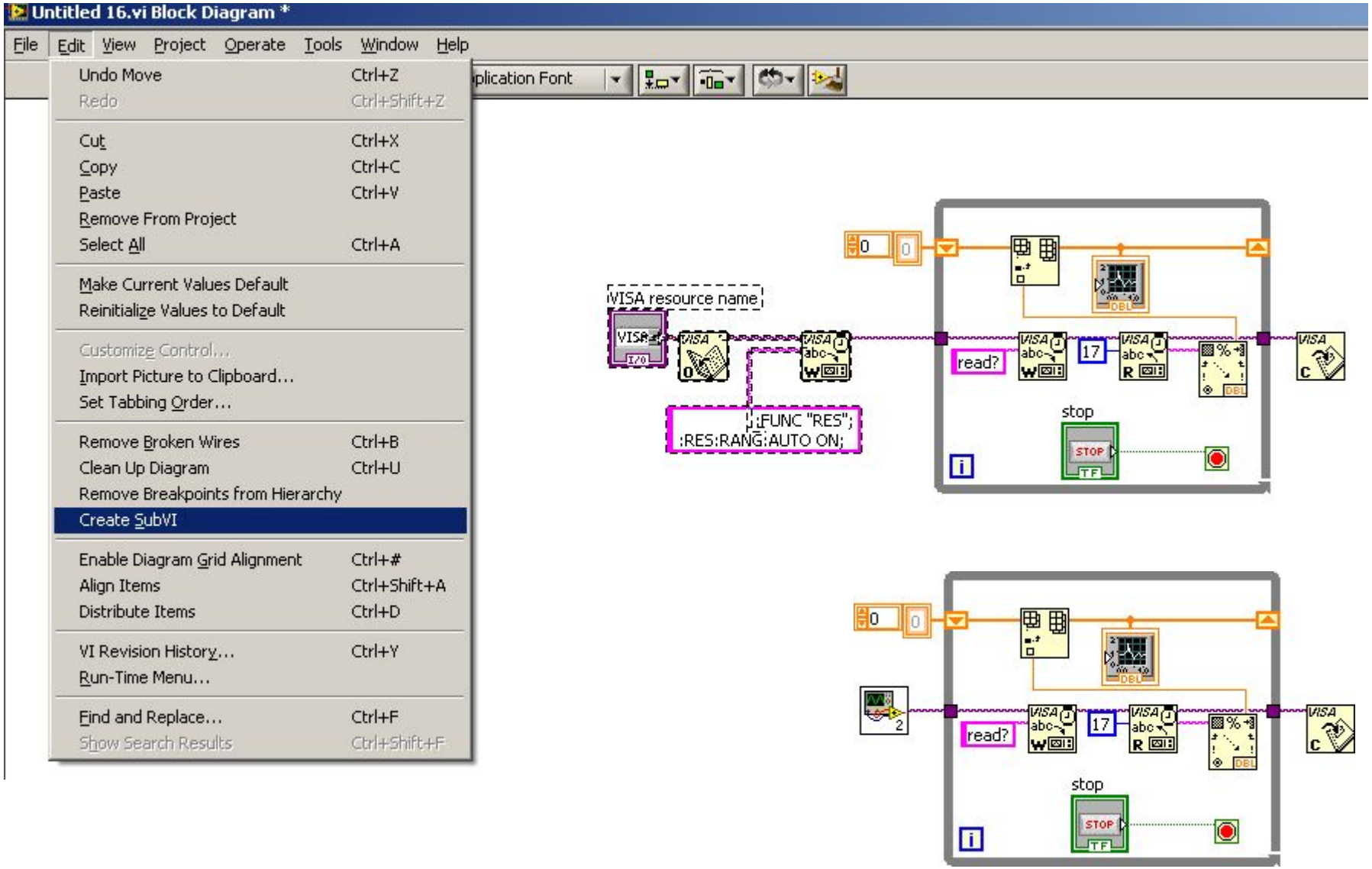


Enable indexing – индексирование разрешено.
Из цикла будут выведены все значения с созданием массива.

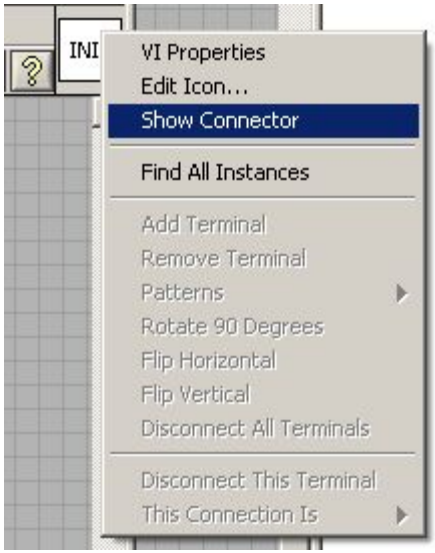


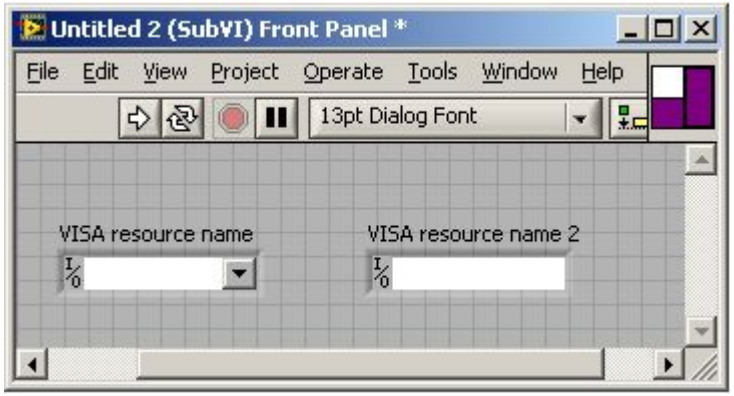
Shift Registers (**Сдвиговые регистр)**
Используются для передачи значения переменной из предыдущей итерации в следующую.

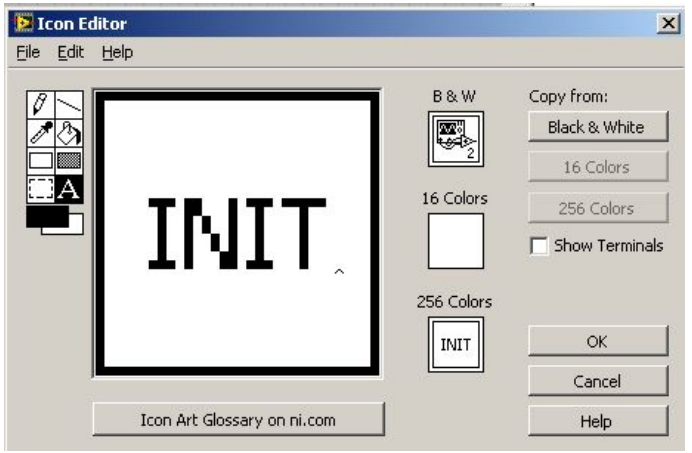
Пример: создаем SubVI

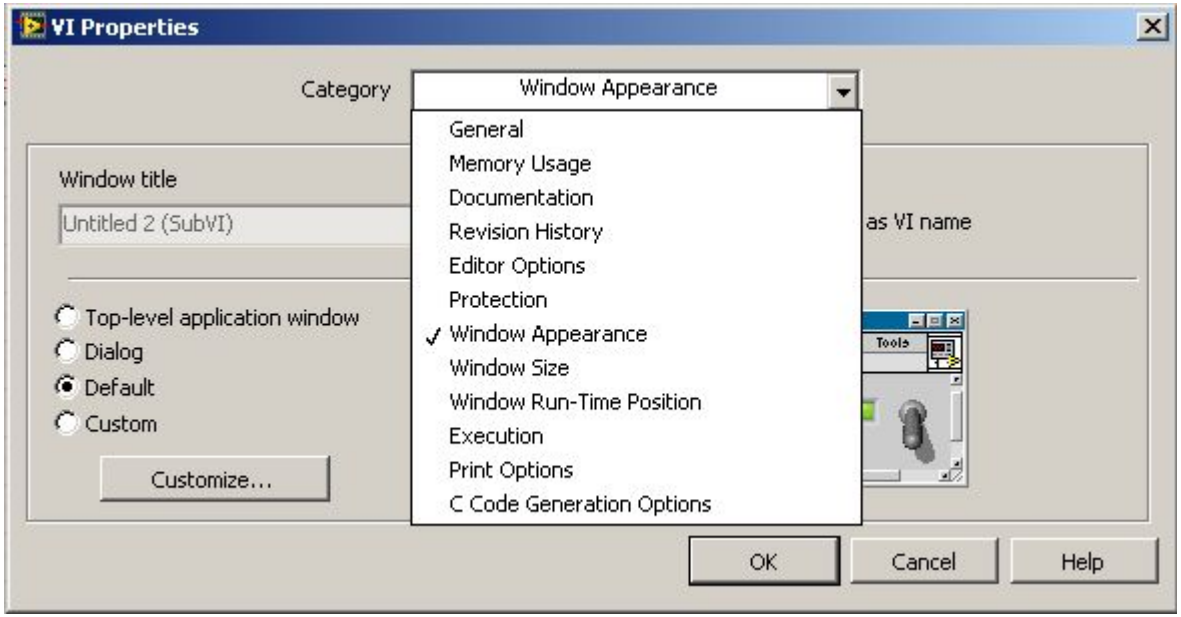


Icon and Connector Panel



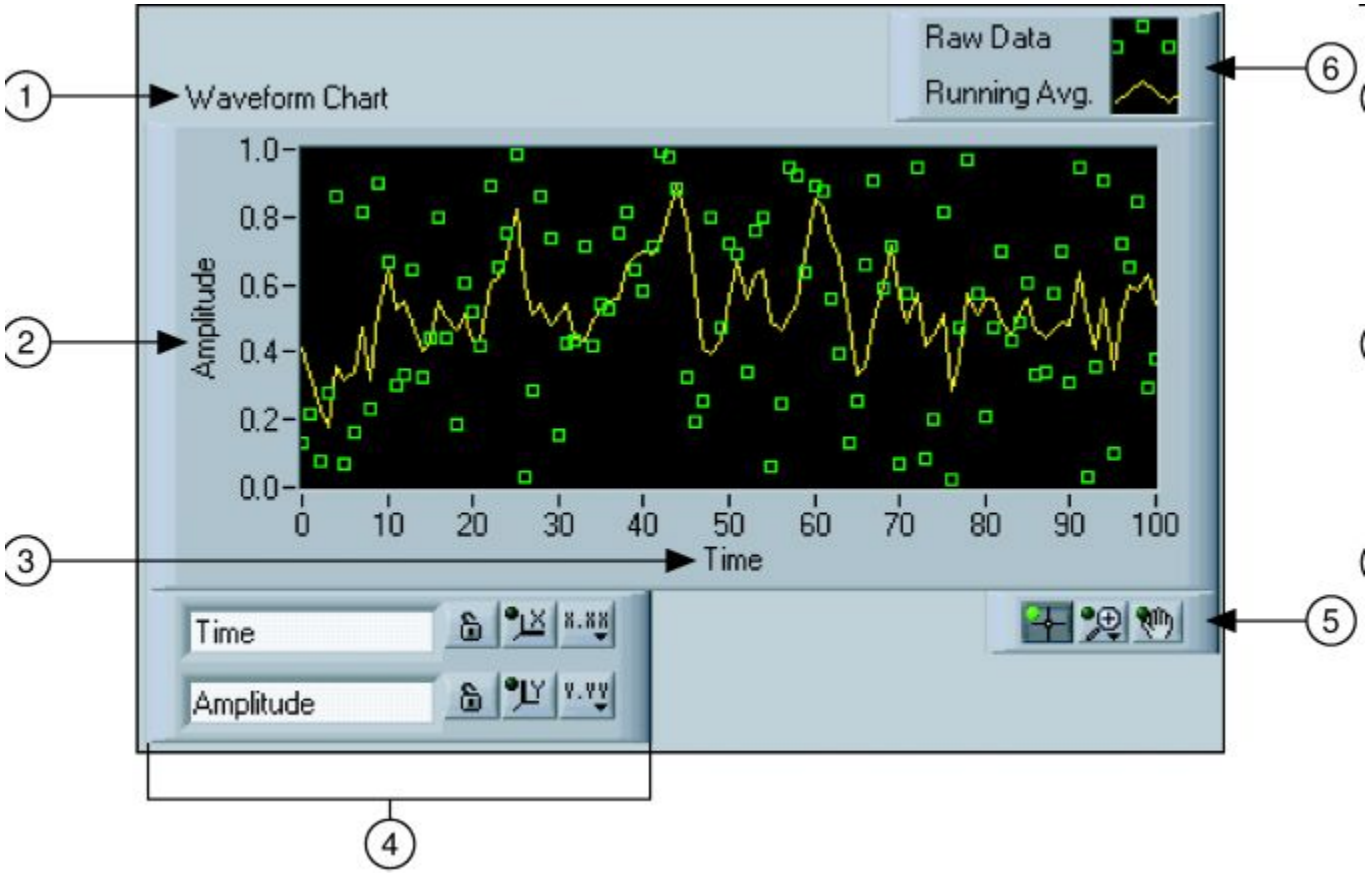






Задачи №10-12

График **Waveform Chart** – аналог самописца

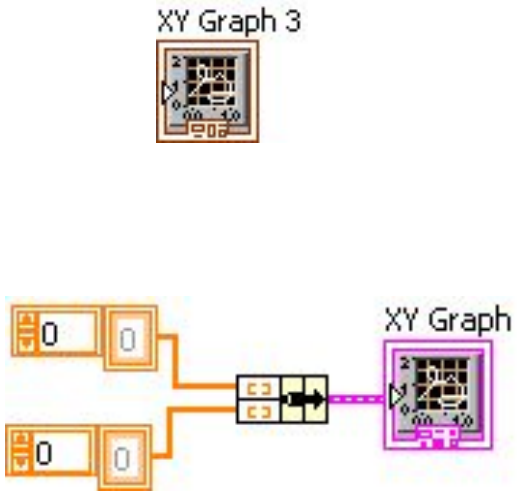


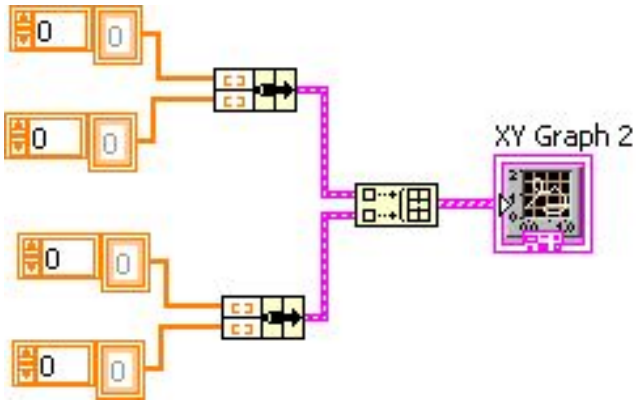
На рисунке цифрами обозначены:
1 - название графика (можно поменять на любое другое)
2,3 - названия осей. Аналогично, обозначаем как нам удобно
4 - панель управления осями (название, включение-выключение Autoscale и т.п.)
5 - панель управления изображением (двигать туда-сюда, зум и все такое)
6 - список графиков. Названия и вид отображения настраиваются по вкусу

XY Graph

Чтобы вывести один график, на вход XY Graph надо подать кластер с двумя массивами - в одном из которых значения x, а в другом - значения y, соответственно.

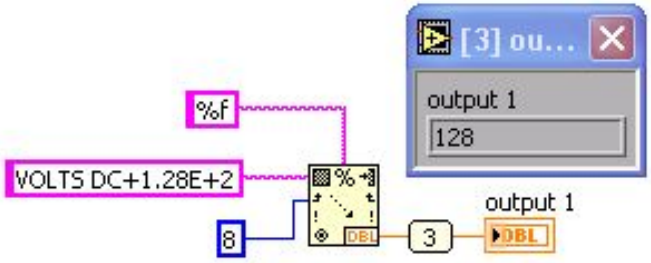
Чтобы вывести несколько графиков одновременно, нужно подать на вход массив из кластеров, в каждом из которых содержится по два массива со значениями для x и y.

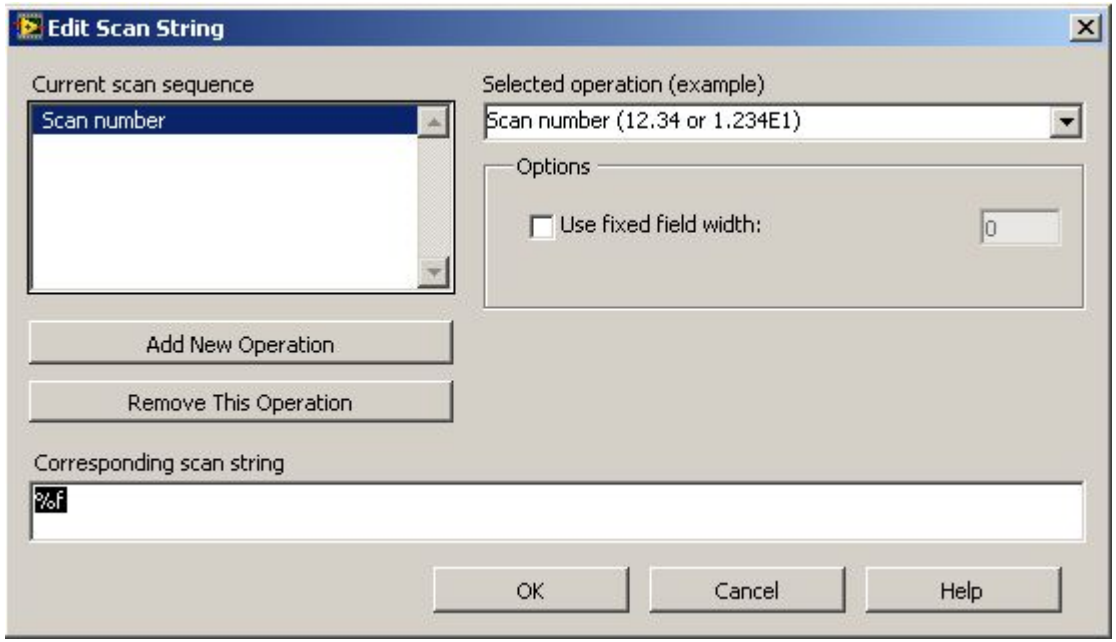


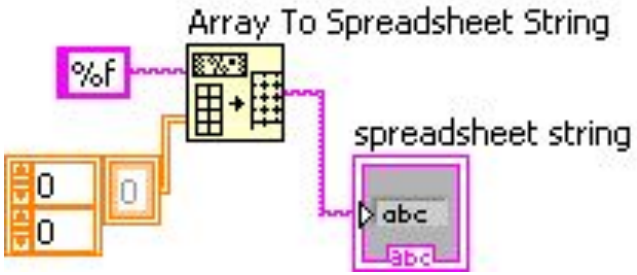


String – переменная типа строка

Scan From String
Показания приборов считываются в формате String. Эта команда переводит строку в число.

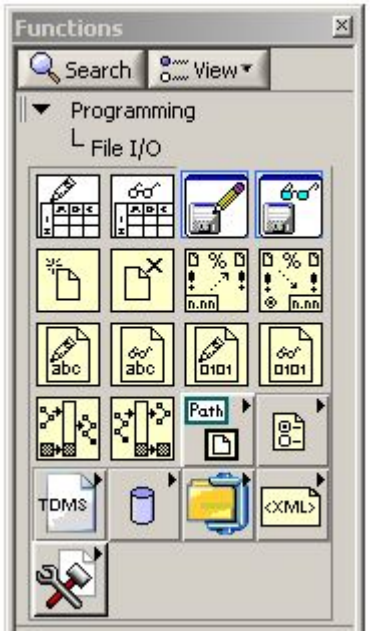


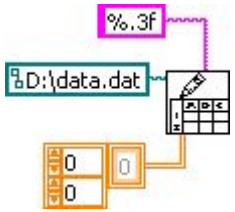




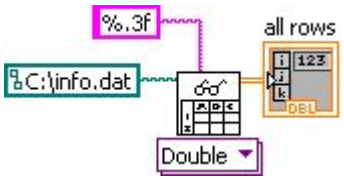
Для записи в файл нужно перевести числовой массив в String.

Файлы – запись, чтение
Write to Spreadsheet File

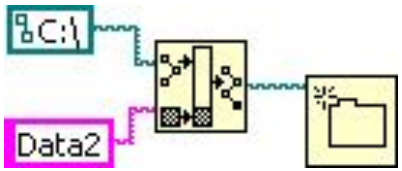




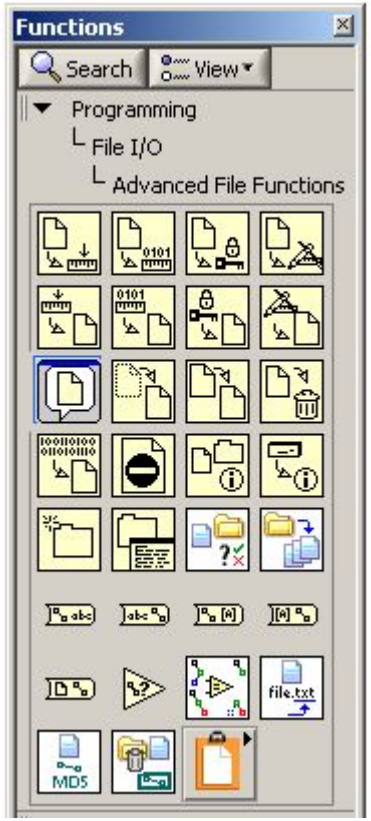
Запись в файл из массива



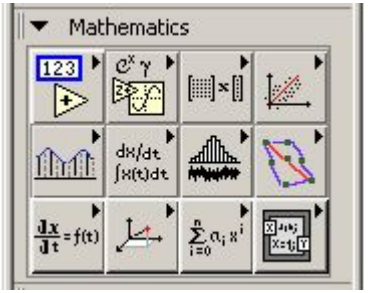
Чтение из файла в массив



Создать директорию Data2



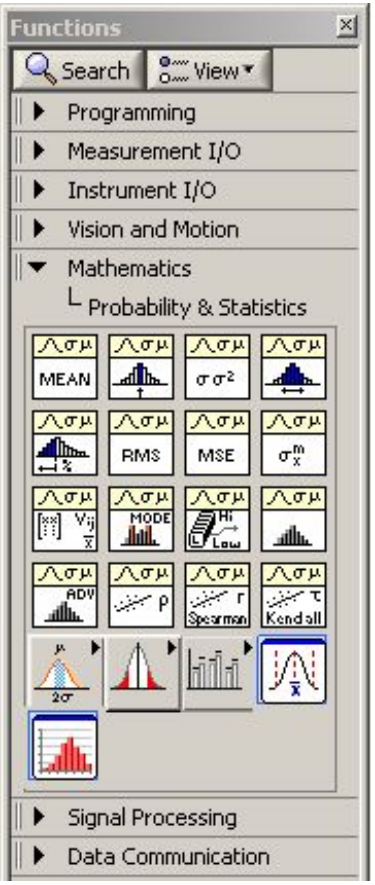
Математика и статистика

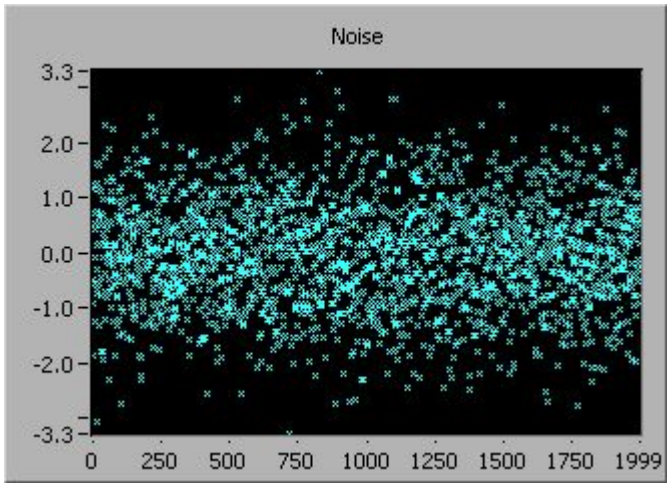


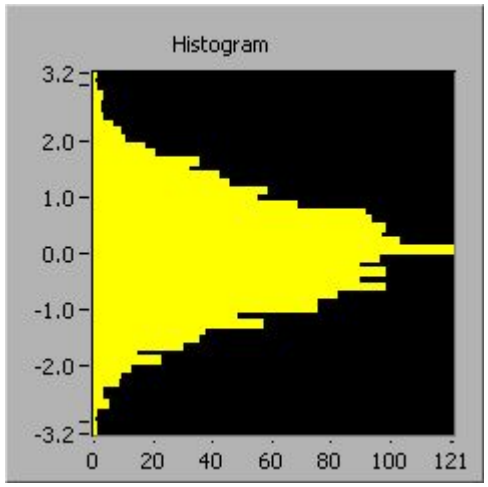
среднее значение

дисперсия

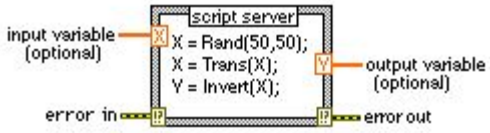
гистограмма распределения

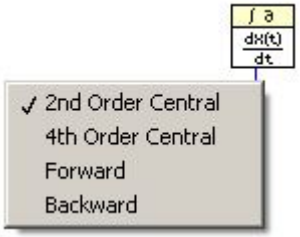






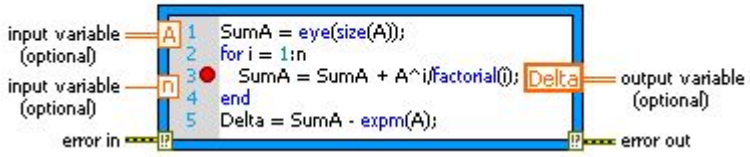
MATLAB script node





вычисление производной

MathScript Node

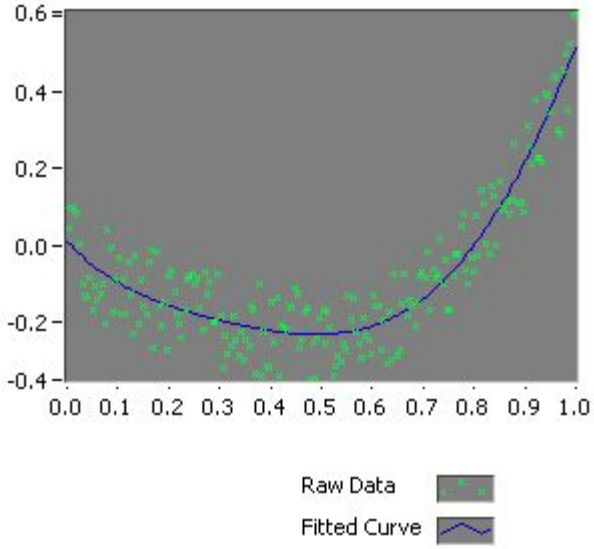


Аппроксимация (фитирование) данных

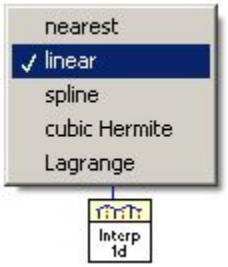
фитирование линейной функцией

фитирование полиномом

фитирование произвольной функцией

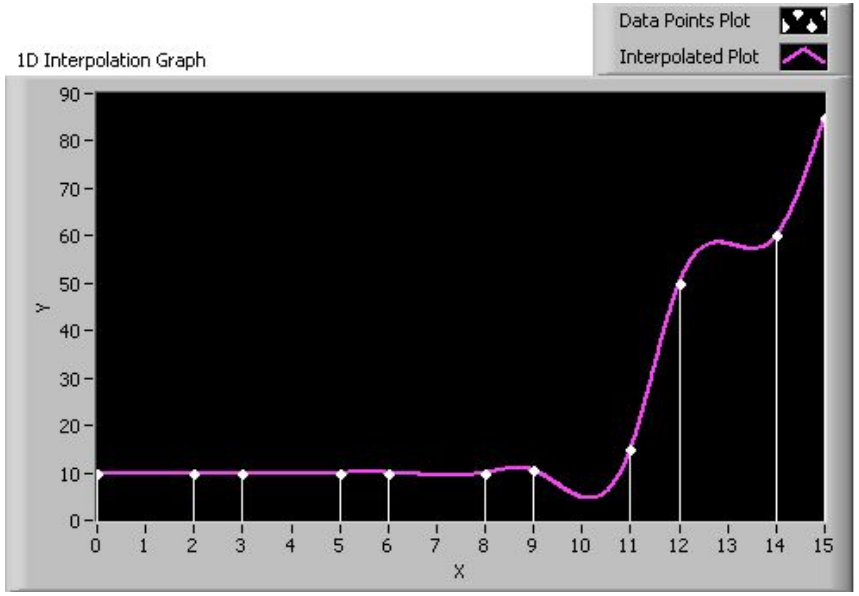


Интерполяция данных



интерполяция одномерного массива (разные методы)

интерполяция сплайном



Дискретное преобразование Фурье

Прямое преобразование:

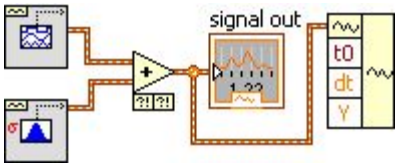
$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0, \ldots, N-1$$

Обратное преобразование:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn} \qquad n = 0, \ldots, N-1.$$

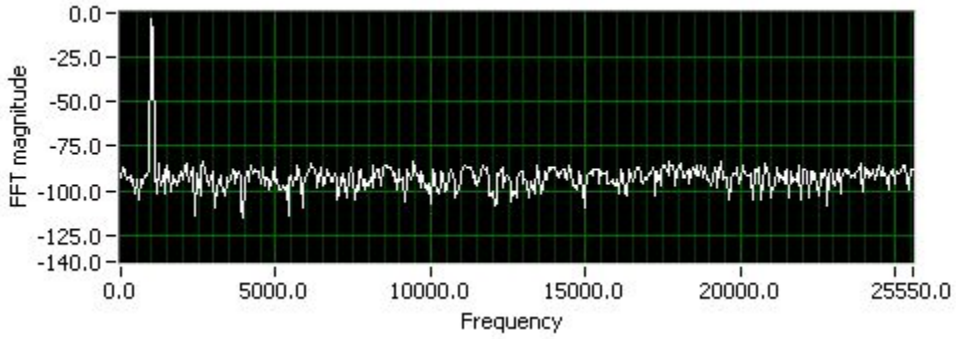
N — количество измеренных точек, а также количество компонент разложения

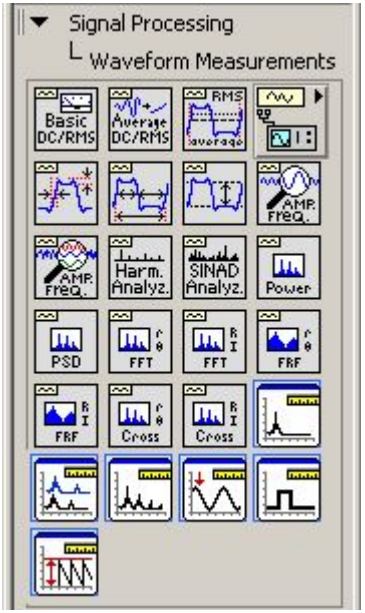
k — частота k-го элемента преобразования $\frac{k}{T}$ где T — время измерения



Переменная типа Waveform

Быстрое Фурье преобразование (FFT)





Задачи №13-18

GPIB

General Purpose Interface Bus (GPIB), описывает интерфейс для коммутации между приборами и контроллерами различных производителей. GPIB – это цифровой, параллельный 8-битный интерфейс связи, со скоростью передачи данных от 1 Мбайт/сек и выше. Шина поддерживает один контроллер (обычно компьютер) и до 14 дополнительных устройств.

Шина GPIB состоит из 24 проводов, из них 8 – линии передачи данных, 5 линий управления шиной (TN, EOI, IFC, REN, и SRQ), 3 линии квитирования (установления связи), и 8 линий земли.

VISA, Virtual Instrument Software Architecture —стандартизированный интерфейс ввода-вывода для управления приборами с персонального компьютера. Поддерживает интерфейсы GPIB, RS-232, LAN, USB.

Между NI DAQ и LabVIEW функционирует связующая программа, называемая **MAX (Measurement and Automation Explorer)** программа анализа измерений и автоматизации). MAX является программным интерфейсом Windows, который предоставляет возможность доступа ко всем платам NI.

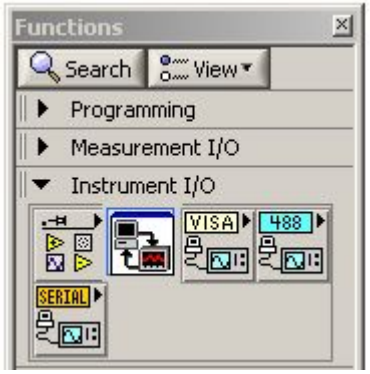
Дистанционное управление прибором

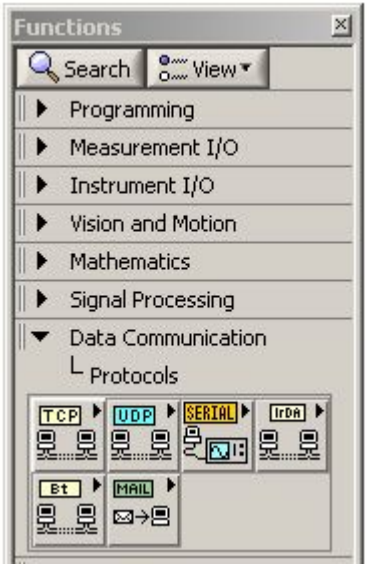
Отправить команду в прибор

Считать данные из прибора

Открыть VISA сессию

Закрыть VISA сессию



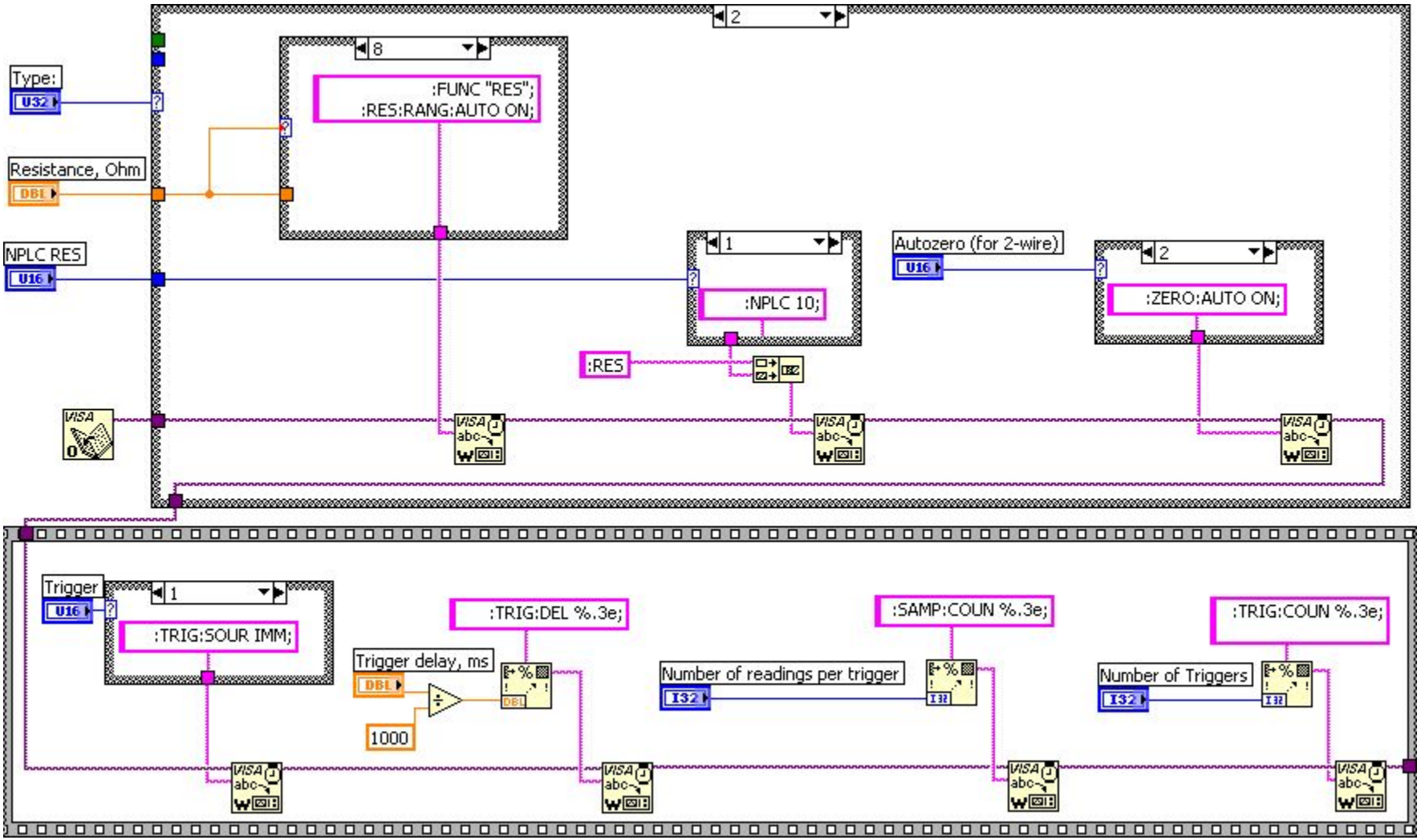


Стандартные команды GPIB

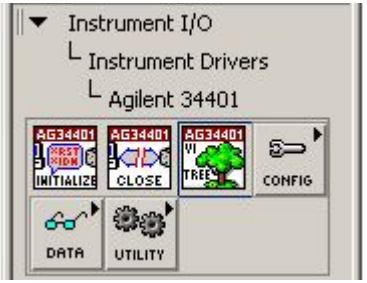
Мнемоника	Группа	Описание
***IDN?**	Сист. информация	Запрос идентификации
***RST**	Внутренние операции	Сброс
***TST?**	Внутренние операции	Провести внутреннее тестирование
*OPC	Синхронизация	Закончить операцию
***OPC?**	Синхронизация	Закончена ли операция?
*WAI	Синхронизация	Дождаться завершения операции
*CLS	Статус операции	Очистить статусный регистр
*ESE	Статус операции	Разрешить работу регистра статуса событий
*ESE?	Статус операции	Разрешена ли работа регистра статуса событий?
*ESR?	Статус операции	Запросить регистр статуса событий
*SRE	Статус операции	Разрешить посылку запросов наобслуживание
*SRE?	Статус операции	Разрешена ли послыка запросов на обслуживание?
***STB?**	Статус операции	Считать статусный байт

Команды, исполняемые прибором

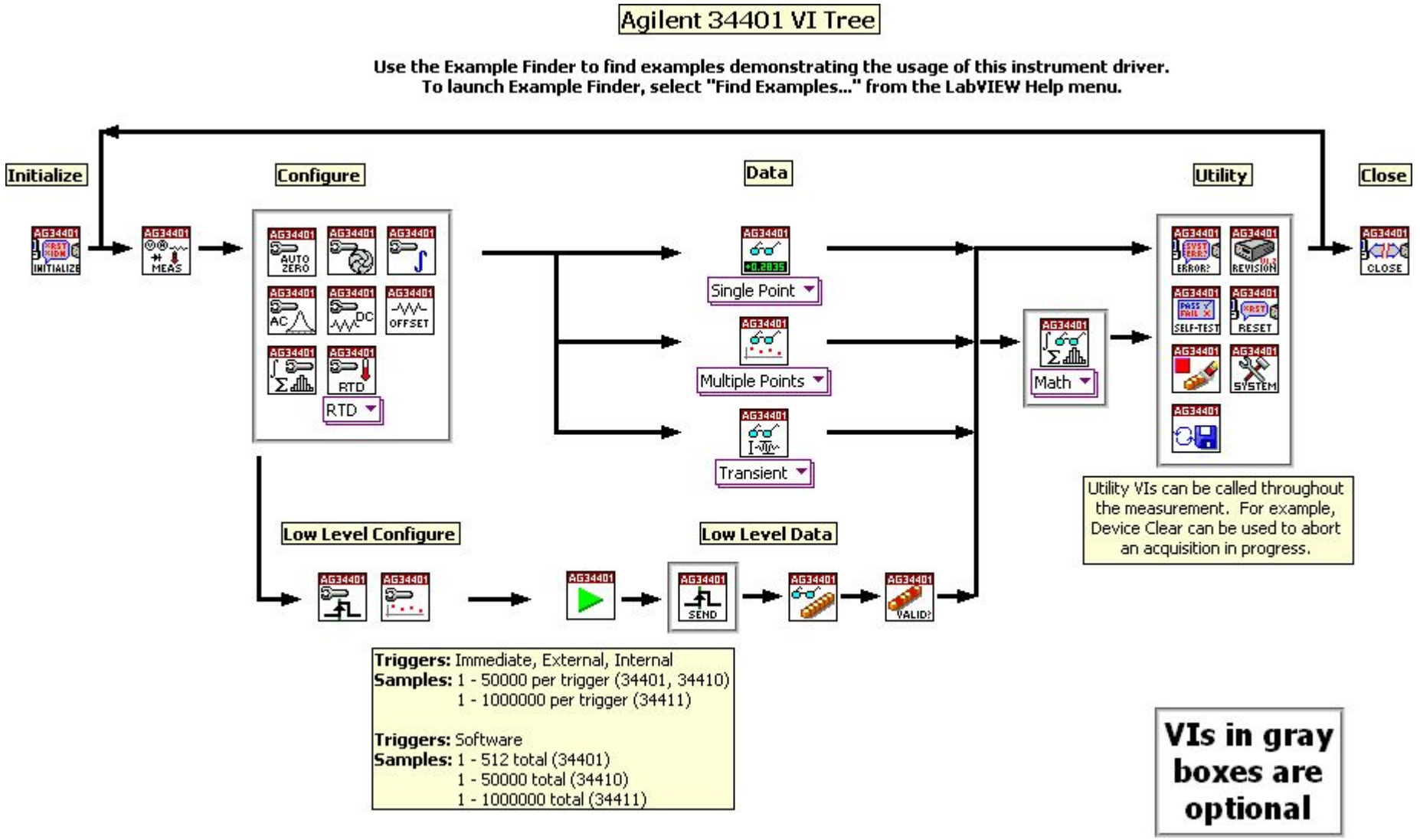
Читаем Users Guide Agilent 34401

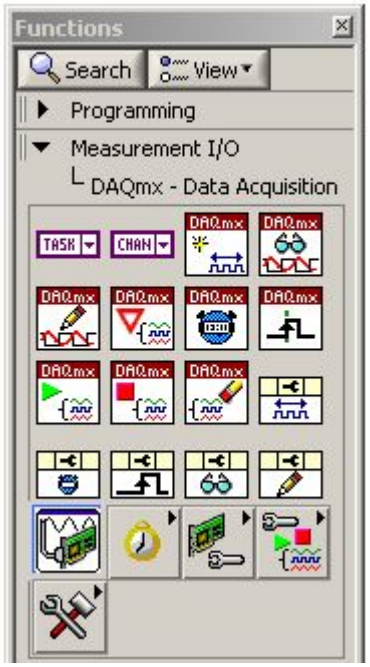


Instruments Drivers –Agilent 34401

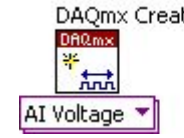


VI Tree.vi — программа с описанием команд мультиметра





Работа с платой АЦП-ЦАП в модуле DAQmx

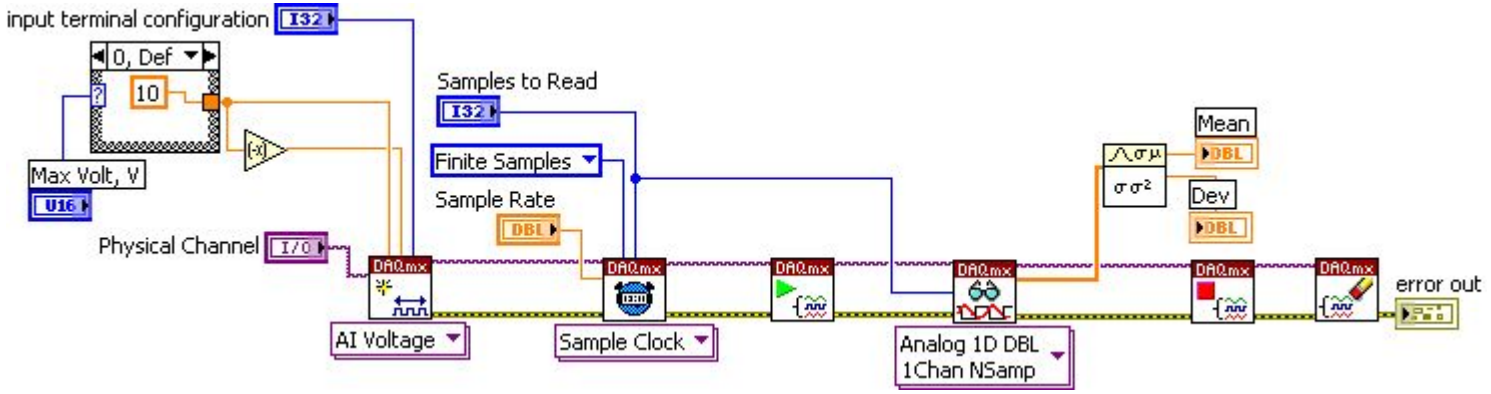


выбор параметров канала измерения

установка частоты измерения и размера буфера

триггер старта измерения

команда считывания содержания буффера



Работа с прибором без LV – стандартные команды. Подключение через IO и NI-MAX, подключение АЦП-ЦАП.

Задача по подключению прибора
Задача по подключению АЦП-ЦАП в модуле DAQmx

Параллельная работа SubVI.
Вызов SubVI программными методами.
Event и Event mashine.

Узлы вызовов (Invoke Nodes)

Следующий набор узлов, упомянутых выше – **Property Node** (Узел свойства) и **Invoke Node** (Узел метода), позволяет реализовать в LabVIEW методологию объектно-ориентированного программирования, когда программный объект описывается набором свойств и методов (а также событий). Программными объектами в LabVIEW могут быть как объекты самой среды LabVIEW, доступные через технологию VI Server (Сервер ВП), так и объекты, доступные через технологии ActiveX, .NET, DataSocket, IMAQImage, Sequrity, SharedVariable и VISA.

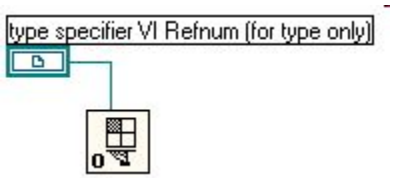
Call By Reference Node (Узел вызова по ссылке) служит для динамического вызова локального или удаленного VI по его ссылке (reference).

Call Library Function Node (Узел вызова библиотечной функции) применяется для вызова процедур, написанных на других языках и оформленных в виде внешних динамически подключаемых библиотек (DLL).

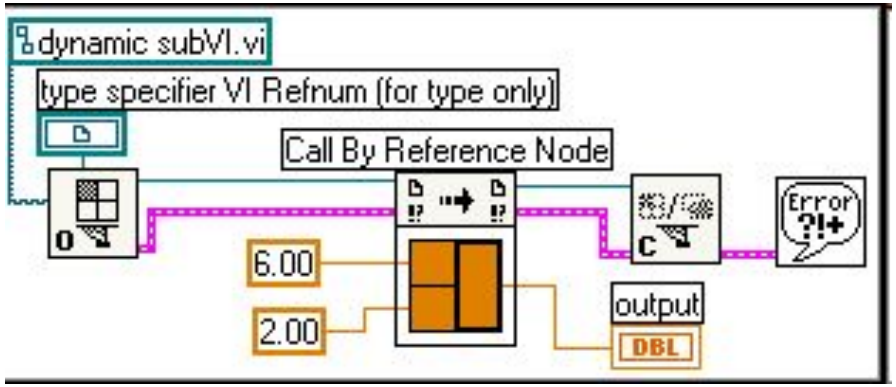
http://digital.ni.com/public.nsf/allkb/4838398F3B2798A186256B82007B4ED0

Вызов SubVI с помощью ссылки

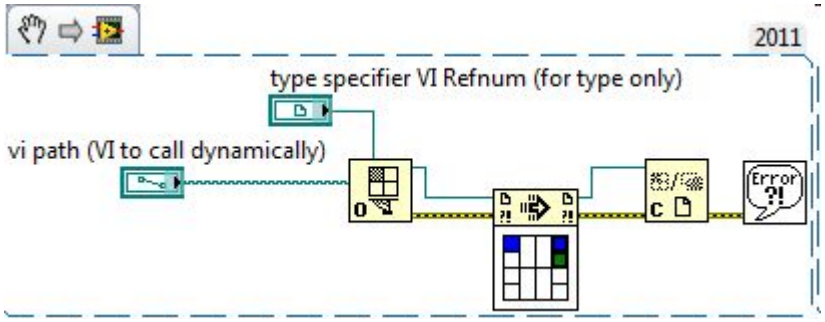
Итак: заходим в палитру Application Control, и выбираем там кубик Open VI Reference - . С помощью него будет определяться "тип" subVI, который мы вызываем. Под типом здесь подрозумевается количество входных и выходных коннекторов. Щелкаем по входу "type specifier VI refnum" правой кнопкой мыши, создаем регулятор. Затем жмем на него опять же правой кнопкой, в контекстном меню выбираем "Select VI class server"-> Browse и открываем тот файл, который будет использован в качестве образца

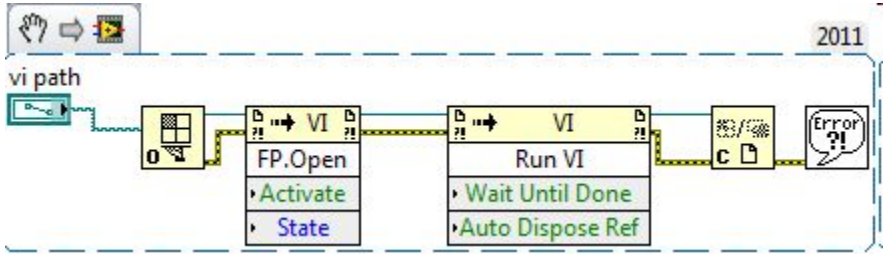


Ставим на диаграмму кубик Call By Reference Node (он находится все там же в палитре Application Control). Должно получитсья примерно вот так:

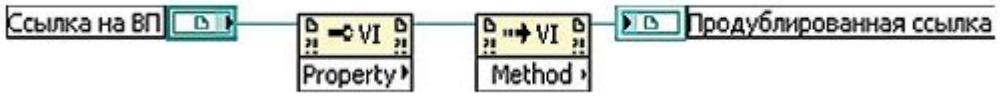


Differences Between Calling a VI with an Invoke Node and Calling a VI with a Call by Reference Node





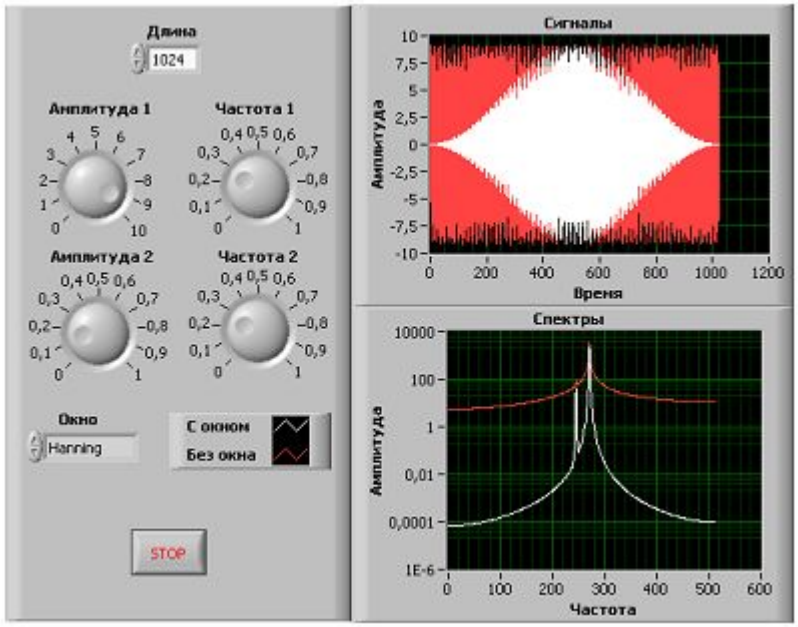
Если же Вы работаете не с VI, а с приложением, необ
ходимо вместо функции Open VI Reference (Открыть
ссылку на VI) использовать Open Application Reference
(Открыть ссылку на Приложение), находящуюся в этом же
подменю по соседству. Закрывается приложение той же
Close Reference. Далее для управление свойствами VI,
его запуска и, при необходимости, останова, следует ис
пользовать функции Property Node (Узел свойств) и
Invoke Node (Узел вызова).



Узел свойств возвращает либо записывает свойства
VI приложения, на который указывает ссылка. Свойства
это разнообразные данные, например, открыта ли лицевая
панель, каково ее название, активна ли она, и масса дру
гих. При выборе этой функции из подпалитры Управление

Использование окон при спектральном анализе сигналов (**Signal Processing Windows**). Спектральное окно – это весовая функция, на которую во временнной области умножается исследуемый сигнал перед спектральным анализом (СА). Необходимость применения окон обусловлена следующими причинами.

Во-первых, это явление «просачивания» спектральной энергии (Spectral Leakage). Его можно проиллюстрировать на примере СА отрезка синусоиды. Если в окно укладывается целое число периодов, то дискретный спектр отличен от нуля только в одной точке. Если же число периодов не целое, спектр отличен от нуля во всех точках. Нетрудно убедиться, что в этом случае на концах окна анализа в сигнале возникают разрывы. Для уменьшения их влияния сигнал умножают на функцию, спадающую на концах интервала анализа.



Во-вторых, при использовании СА для выделения слабого гармонического сигнала на фоне сильного с близкой частотой, умножение на окно позволяет добиться быстрого спада спектрального отклика при отстройке от частоты сильного сигнала.