

# Моделирование систем Цифровой Обработки Сигналов в среде LabVIEW

Круглов Евгений Владимирович, аспирант МИФИ

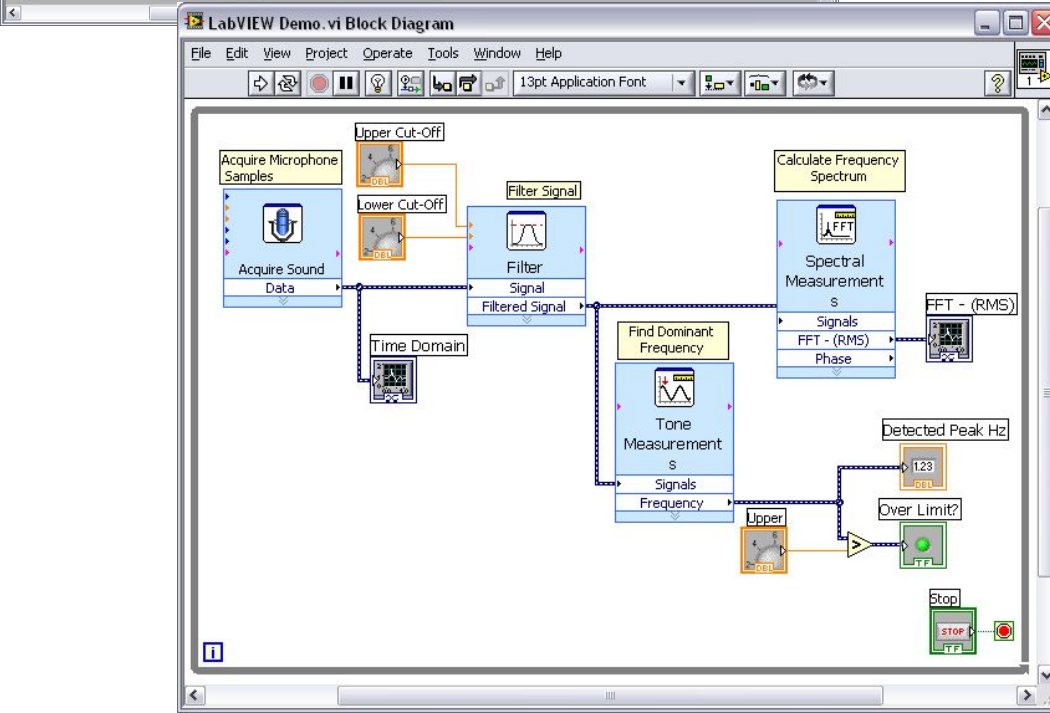
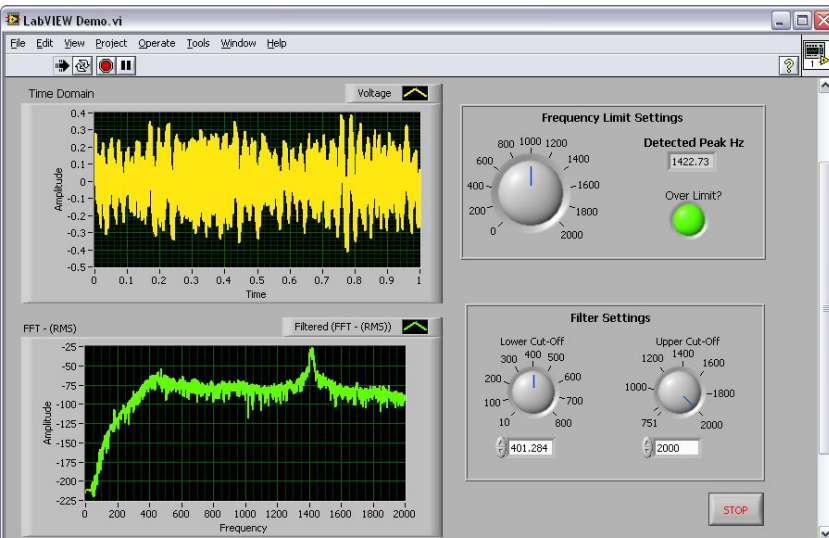
Решетов Владимир Николаевич, к.ф.-м. н. доцент МИФИ.

Москва 2008

# Введение в LabVIEW

Графическая среда программирования

для инженеров и ученых



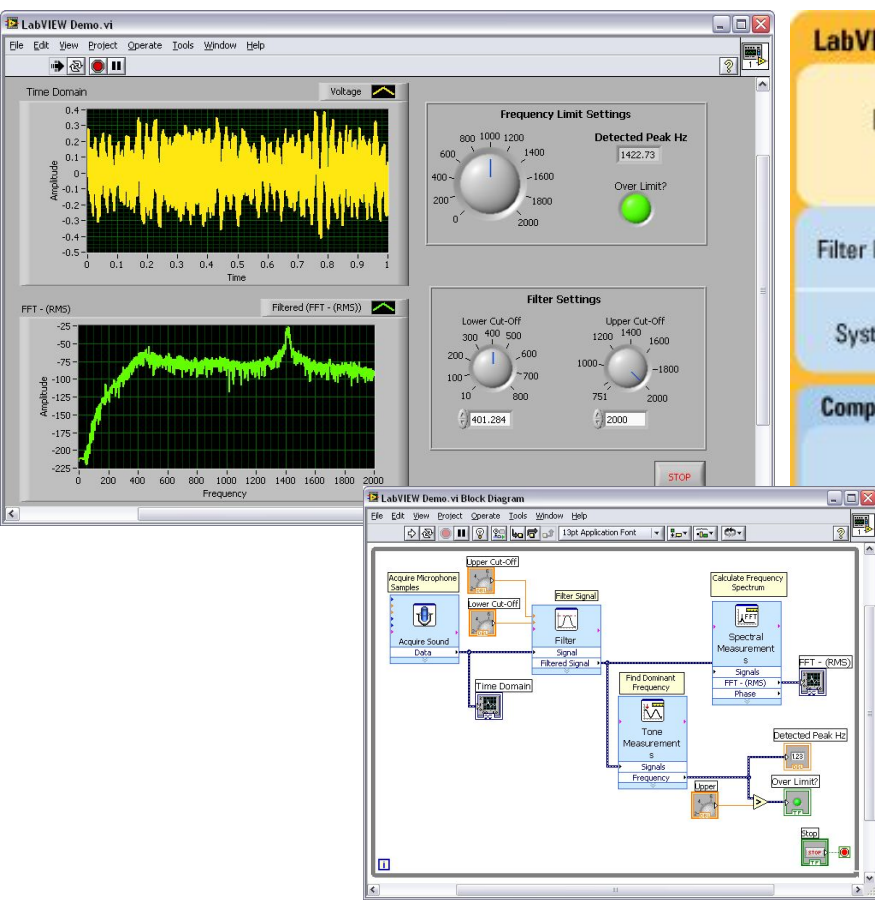
# LabVIEW™ 8

# Решаемые задачи

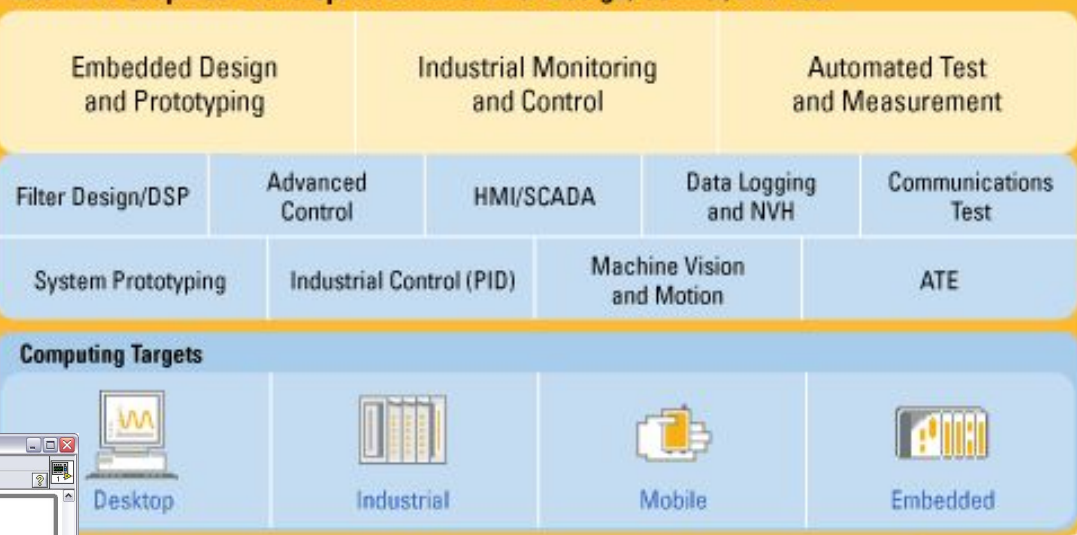
- Научиться основам графического программирования
- Возможность применения LabVIEW для создания собственных приложений
  - Сбор, анализ, отображение и сохранение данных
  - Поиск и использование функций анализа и обработки
  - Использование различных типов данных
  - Запись и печать результатов

# Графическая среда разработки LabVIEW

- Среда графического программирования
- Исполнение кода на различных ОС и устройствах
- Широкий диапазон приложений



## LabVIEW Graphical Development Platform for Design, Control, and Test



# Приложения на базе виртуальных приборов

- **Разработка**

- Обработка сигналов и видеоизображений
- Встраиваемые программируемые системы
  - (PC, DSP, FPGA, Microcontroller)
- Построение прототипов и моделирование
- Многое другое

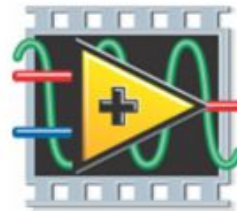
- **Управление**

- Автоматическое управление динамическими системами
- Мехатроника и робототехника
- Многое другое

- **Измерения**

- Электроника и Электротехника
- Основы измерительных технологий
- Многое другое

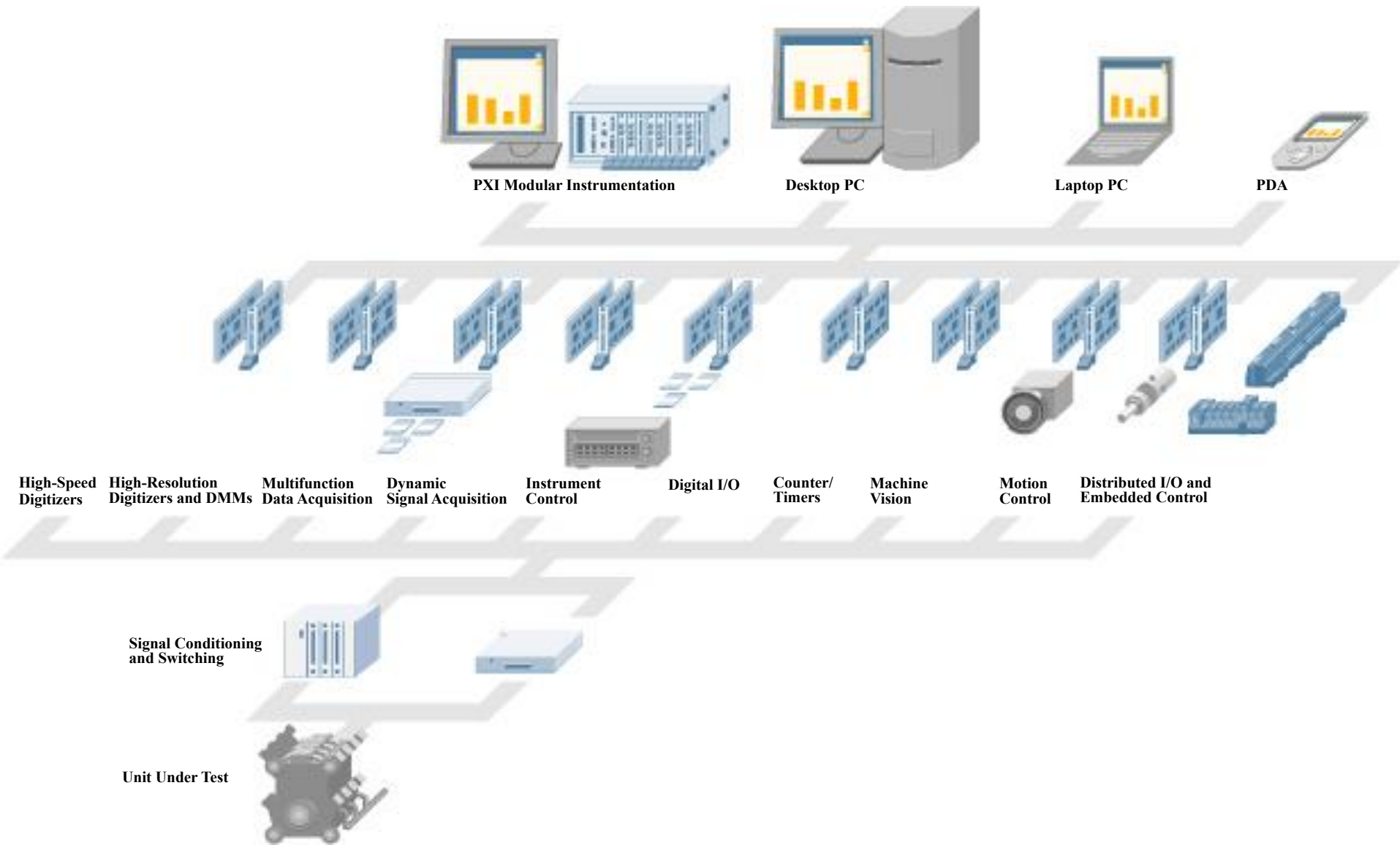
*Единая платформа разработки*



NATIONAL INSTRUMENTS

**LabVIEW™**

# Технологии NI – Интегрированная аппаратная платформа



# Среда LabVIEW

## A. Сбор данных при помощи ПК

- Data Acquisition Devices
  - NI-DAQ
  - Симулятор сбора данных
  - Звуковая карта

## B. Среда графического программирования LabVIEW

Лицевая панель / Блок диаграмма

- Палитра функций и инструментов

## C. Компоненты приложений в LabVIEW

Создание под ВП

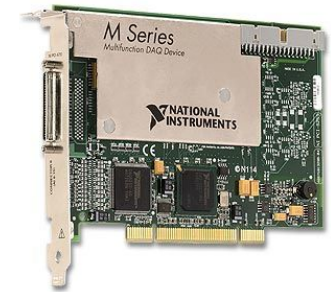
- Графическое построение кода

## D. Дополнительное контекстное меню помощи

- Поиск функций
- Особенности работы в LabVIEW

# Настройка оборудования

- Устройства сбора данных (DAQ)
  - Карты USB, PCI, or PXI Device
  - Configured in MAX
- Симулятор устройства (DAQ)
  - Симуляция DAQ устройства
  - Configured in MAX
- Звуковая карта
  - Встроенная во многие ПК





# Технические характеристики устройств

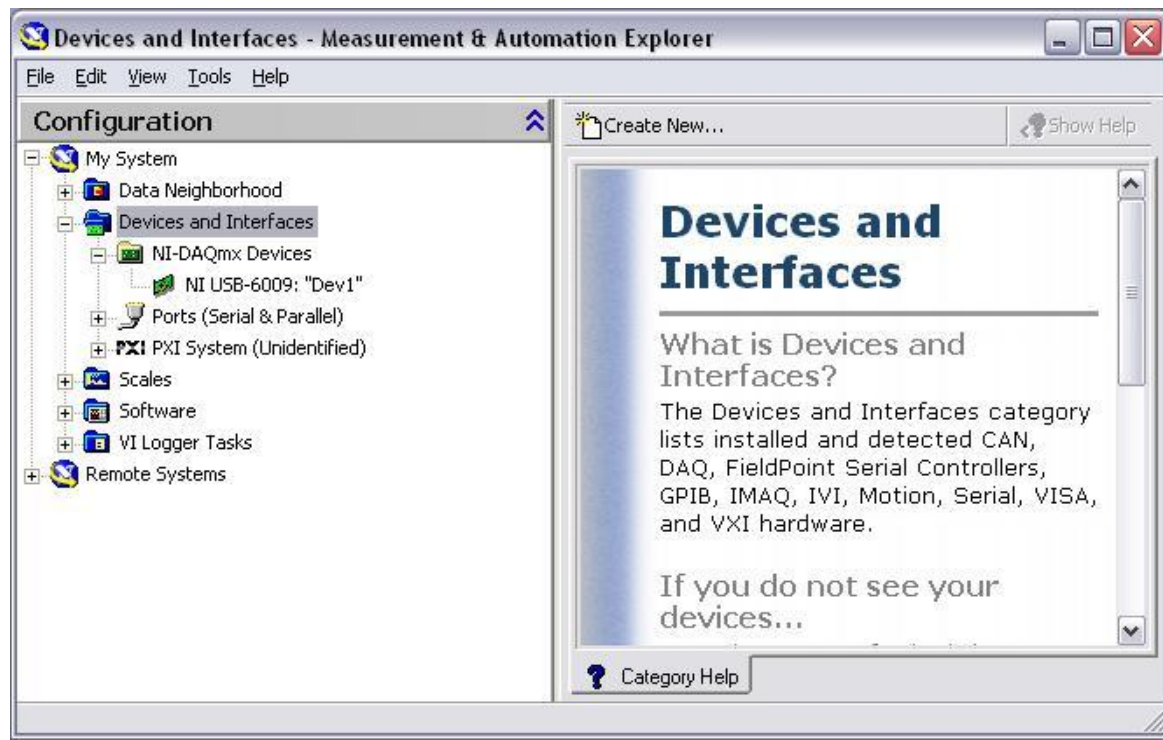


	<b>Звуковая карта*</b>	<b>NI USB DAQ</b>	<b>NI PCI DAQ</b>	<b>Приборы*</b>
<b>AI полоса пр</b>	<b>8–44 KS/s</b>	<b>10–200 KS/s</b>	<b>250 K–1.2 Ms/s</b>	<b>20kS/s–2 GS/s</b>
<b>Точность</b>	<b>12–16 bit</b>	<b>12–16 bit</b>	<b>14–18 bit</b>	<b>12–24 bit</b>
<b>Портатив.</b>	<b>x</b>	<b>x</b>	<b>—</b>	<b>some</b>
<b>AI каналы</b>	<b>2</b>	<b>8–16</b>	<b>16–80</b>	<b>2</b>
<b>АО каналы</b>	<b>2</b>	<b>1–2</b>	<b>2–4</b>	<b>0</b>
<b>AC or DC</b>	<b>AC</b>	<b>AC/DC</b>	<b>AC/DC</b>	<b>AC/DC</b>
<b>Запуск</b>	<b>—</b>	<b>x</b>	<b>x</b>	<b>x</b>
<b>Калибровка</b>	<b>—</b>	<b>x</b>	<b>x</b>	<b>x</b>

# Что такое MAX?

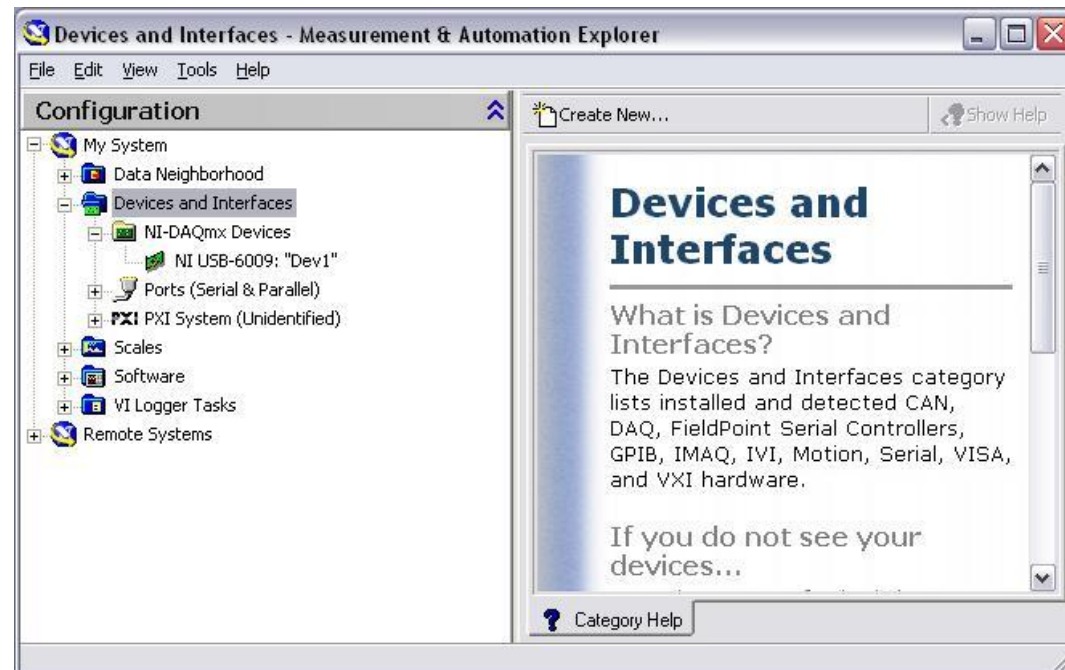
- MAX - Measurement & Automation Explorer.
- MAX конфигурирует и отображает все устройства National Instruments- DAQ, PCI/PXI instruments, GPIB, IMAQ, IVI, Motion, VISA, and VXI devices.
- Окно для конфигурации и тестирования .

Найдите иконку  
на Рабочем столе



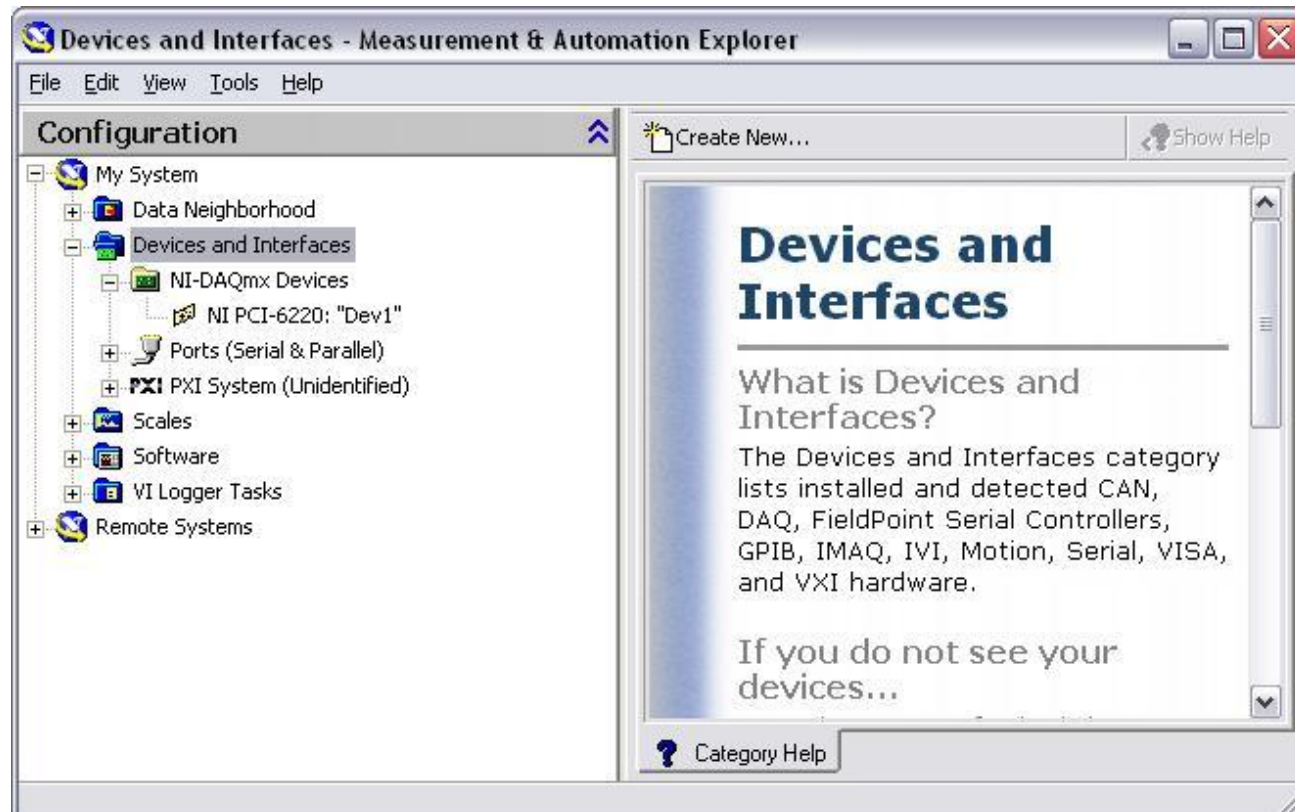
# Настройка устройств

- Используйте MAX для :
  - Конфигурирования и тестирования Вашей карты сбора данных (DAQ)



# Настройка устройств

- Используйте MAX для :
  - Конфигурирования и тестирования Симулятора карты сбора данных (DAQ)

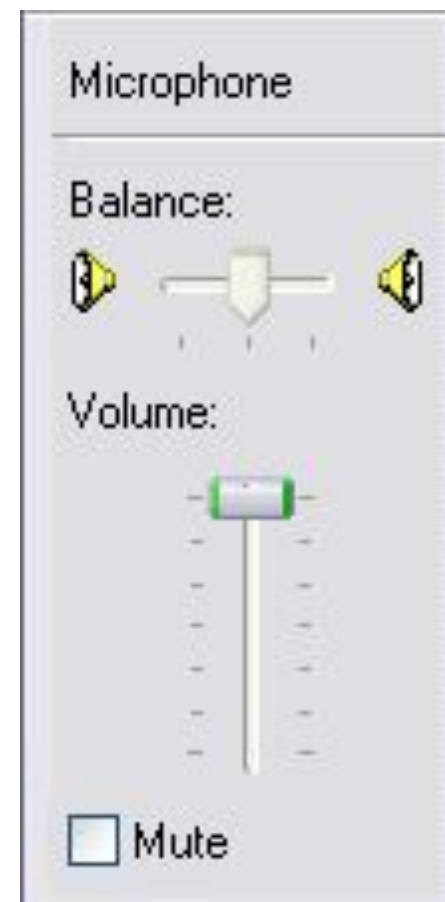


# Настройка устройств

- Используйте Windows для :
  - Тестирования Звуковой карты



Разблокируйте  
микрофон



# Откройте и запустите LabVIEW

Пуск » Программы » National Instruments LabVIEW



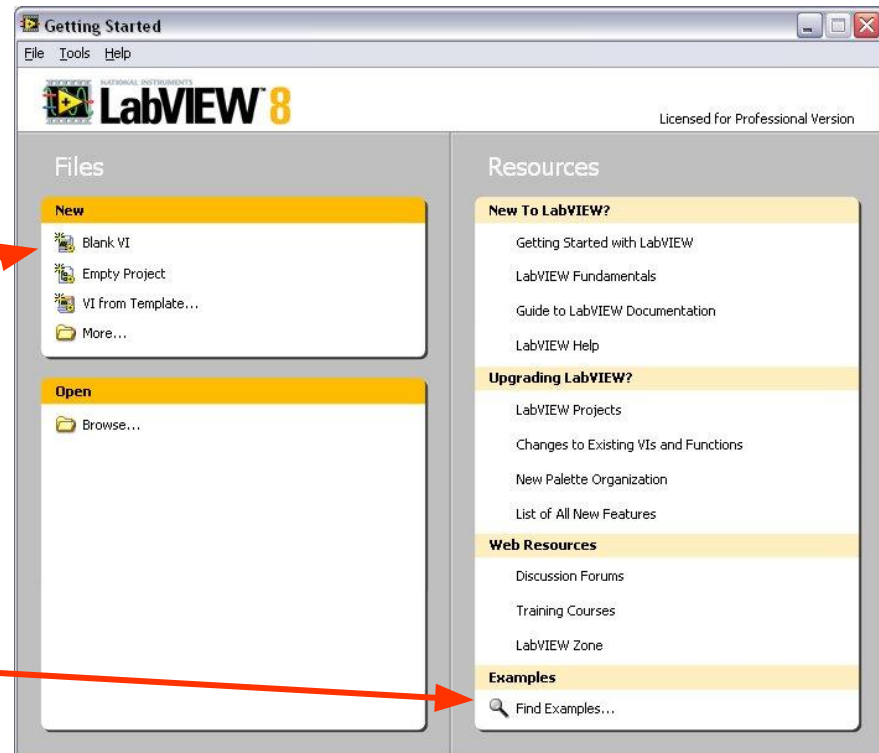
National Instruments LabVIEW 8.0

Startup Screen:

Начать с нового VI:  
НОВЫЙ VI

ИЛИ

Начать с примера:  
Examples » Find  
Examples...



# Программа в LabVIEW называется Виртуальным прибором (ВП)

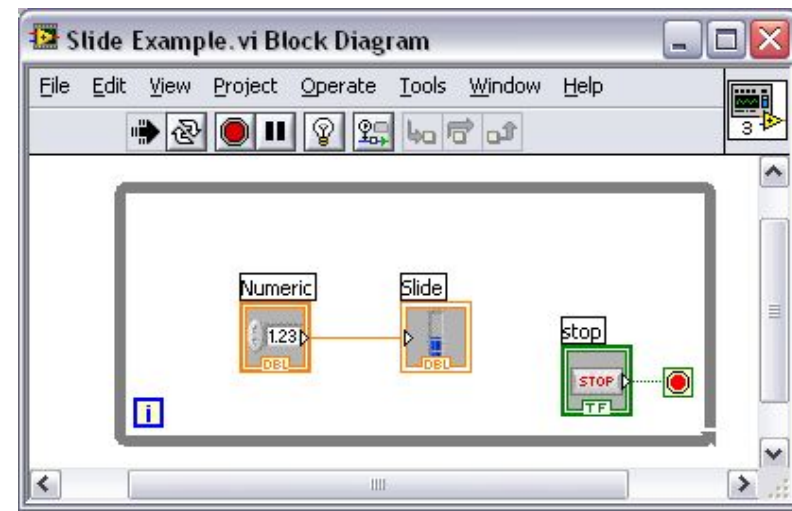
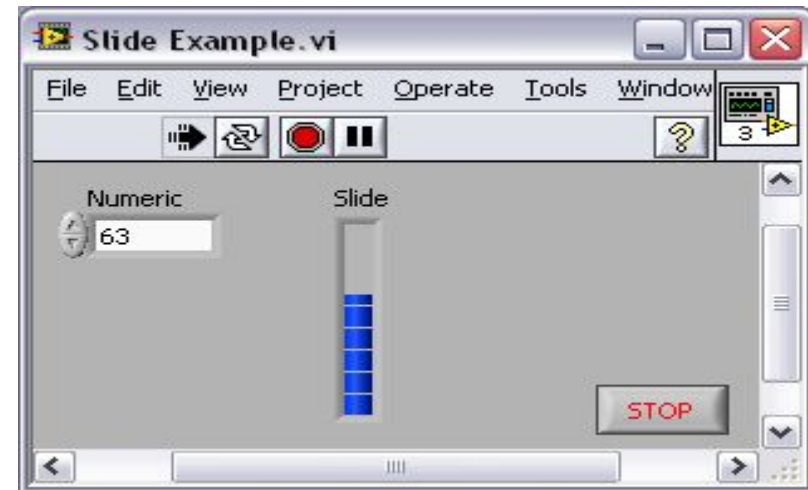
Каждый ВП имеет 2 окна

## Лицевая панель

- Интерфейс пользователя
  - Элементы управления = Входы
  - Индикаторы = Выходы

## Блок диаграмма

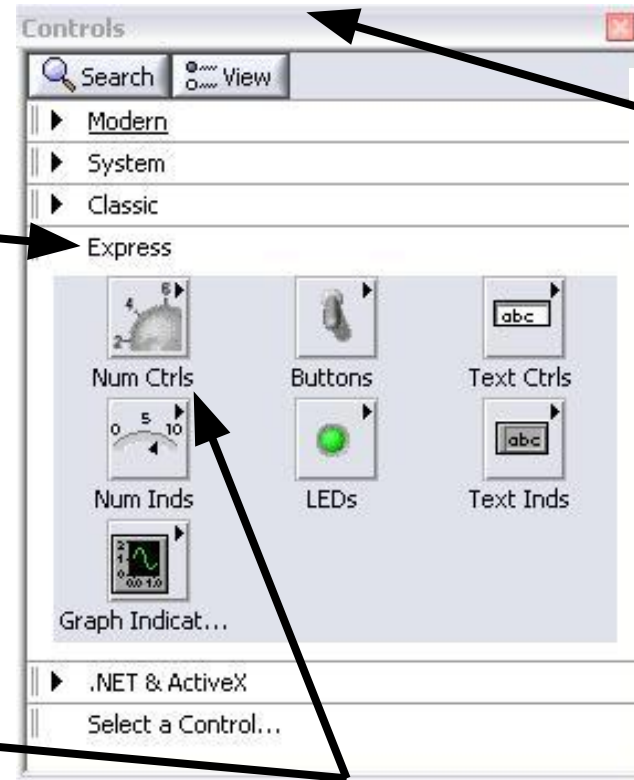
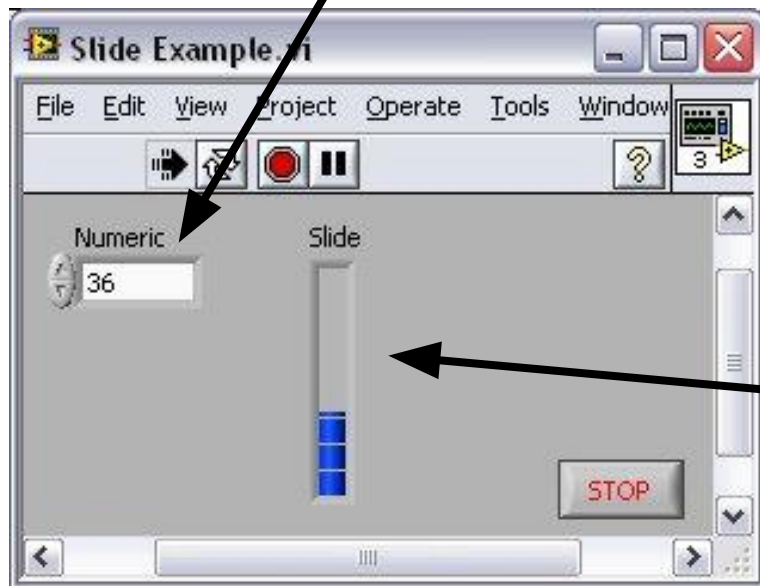
- Графический код
  - Данные передаются по проводникам от элементов управления, через функции, к индикаторам





# Элементы управления и индикаторы

**Числовые элементы  
управления:**

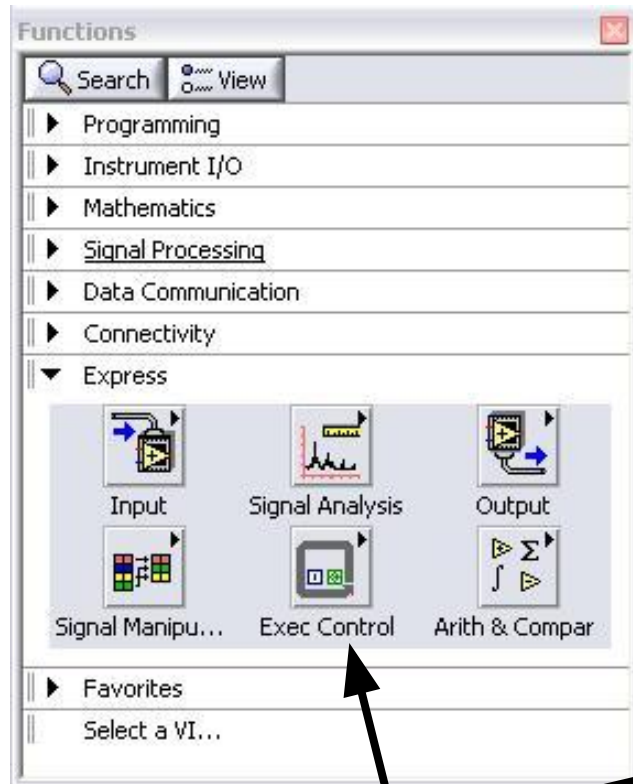


**Вид  
палитры**

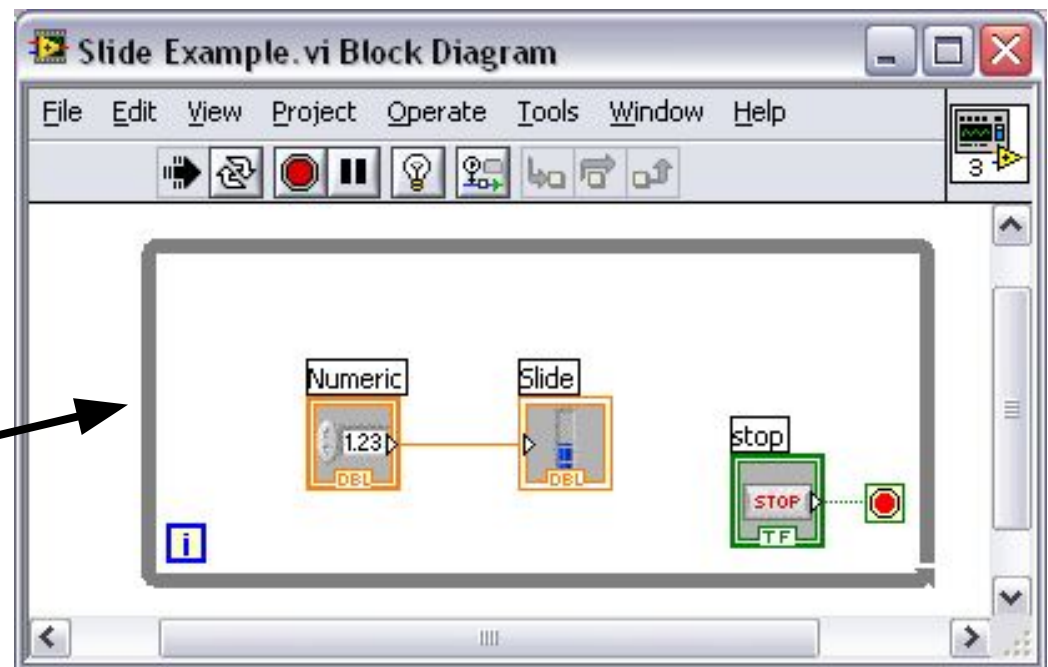
**Числовые или слайдовые:  
индикаторы**



# Палитра всех функций



Размещаются на блок- диаграмме

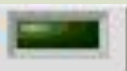


Структуры:  
Цикл по условию

# Палитра инструментов



- Рекомендуется: Включить опцию «Автоматический выбор инструментов»
- Палитра может быть доступна на лицевой панели и на блок диаграмме



**Automatic Selection Tool**

Доступные инструменты опции «Автоматический инструментов выбор»

:



**Operating Tool**



**Positioning/Resizing Tool**



**Labeling Tool**

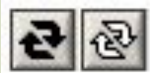


**Wiring Tool**

# Управление программой и блок-диаграммы



**Кнопка запуска**



**Кнопка запуска в непрерывном режиме**



**Экстренная остановка**

## Дополнительные кнопки управления на блок-диаграмме



**Отладочный индикатор**



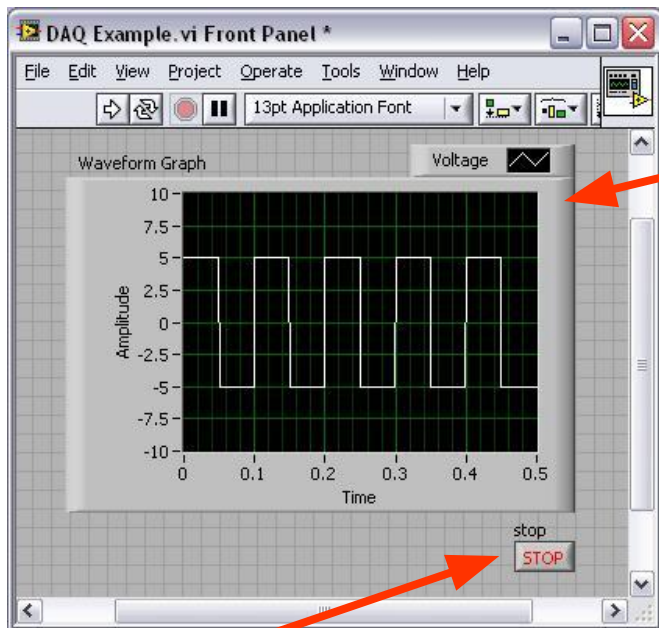
**Retain Wire Values Button**



**Кнопки пошагового доступа в узлы кода**

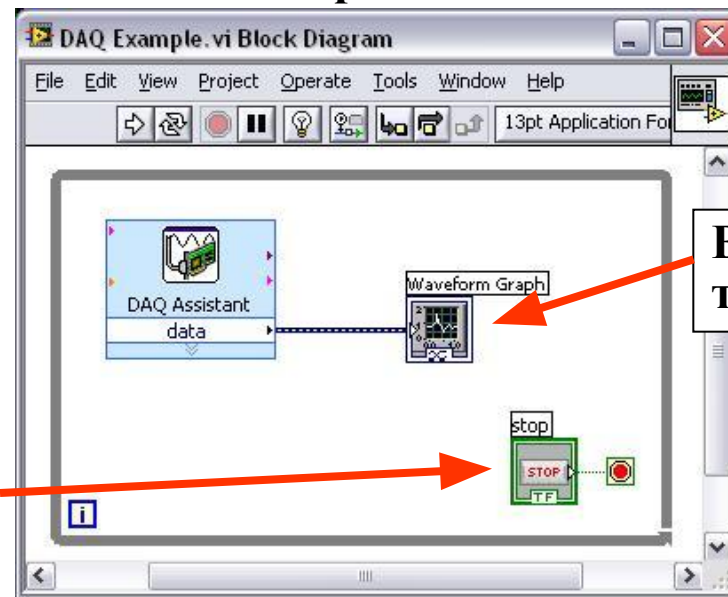
# Создание ВП

## Окно лицевой панели



Графический индикатор

## Блок- диаграмма



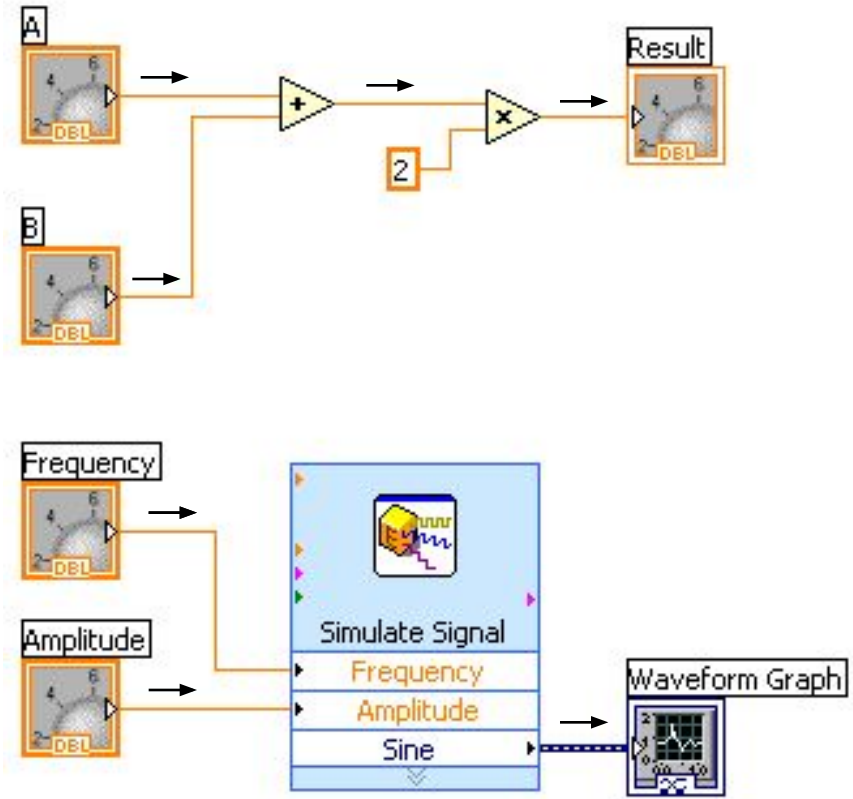
Выходной терминал

Логический эл. управления

Входной терминал

# Поточный метод программирования

- Исполнение кода
  - Зависит от потока данных
  - Код на блок-диаграмме **не** исполняется с лева на право
- Код исполняется по мере поступления всех данных на все входные терминалы
- Отображение происходит по мере поступления всех данных на все выходные терминалы.



# Техника отладки

- Поиск ошибок



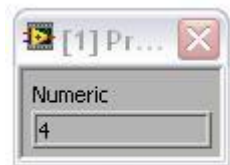
Нажать на кнопку **Запуск**  
Появится окно с описанием ошибки.

- Визуальная отладка



Нажать на кнопку **Execution Highlighting**;  
Визуальная передача данных . Значения  
отобразятся на проводниках

- Probes

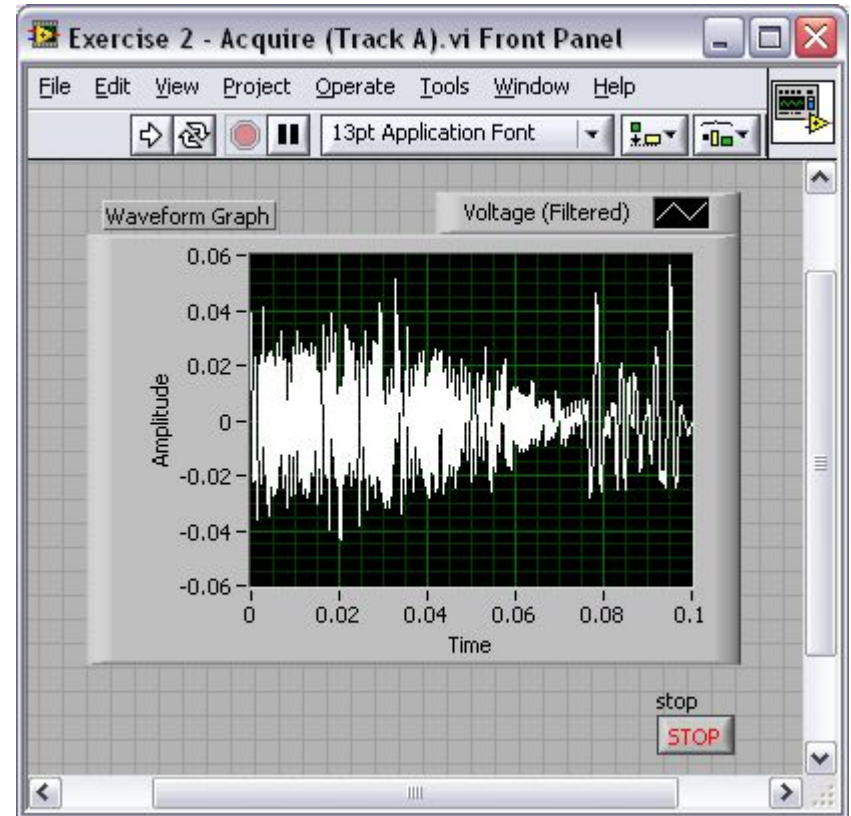


Правый кнопка мыши по проводнику для  
отображения индикатора пробника



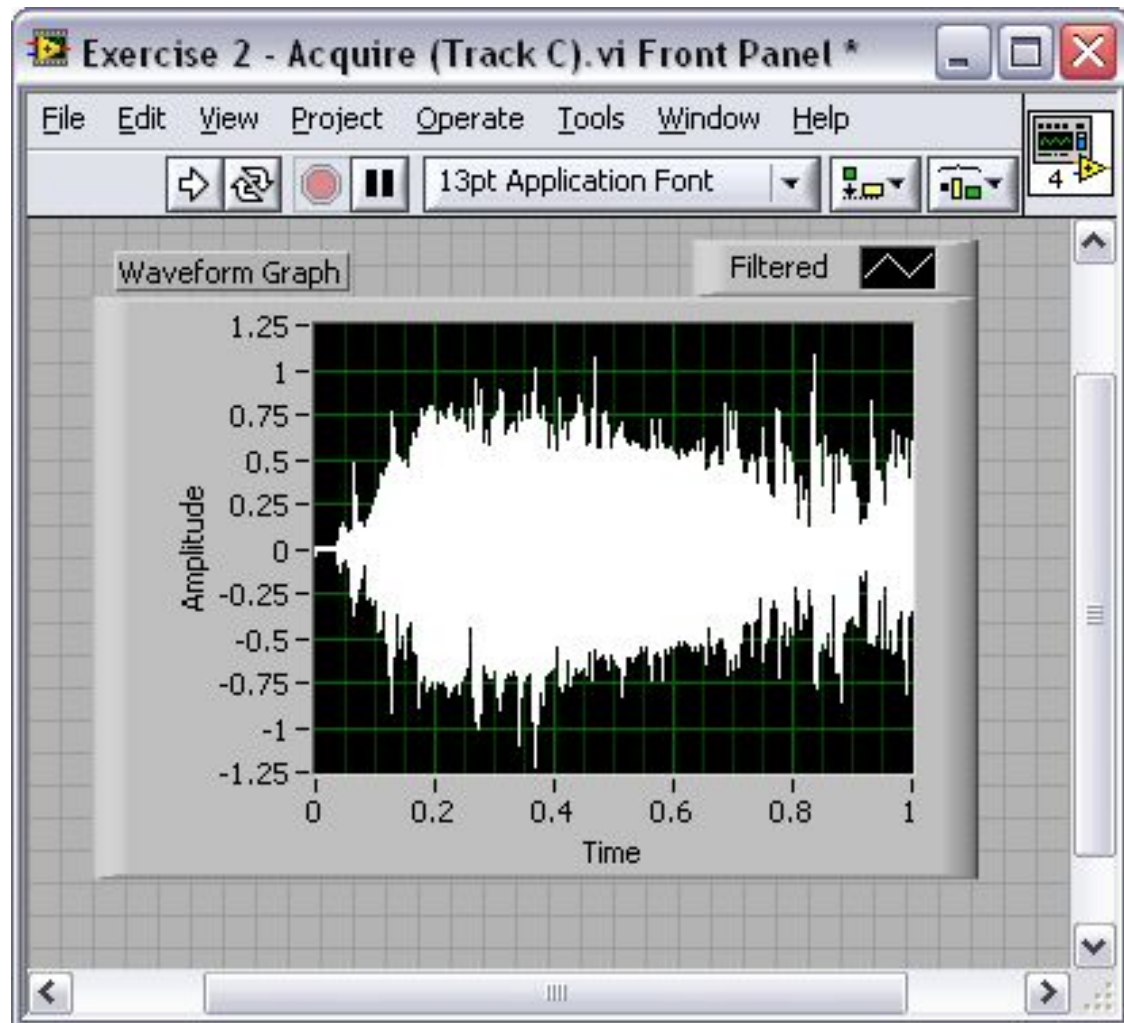
# Сбор данных с использованием DAQ

- Используя LabVIEW :
  - Оцифруем сигнал при помощи DAQ устройства



# Сбор данных со звуковой карты

- Используя LabVIEW и стандартное оборудование



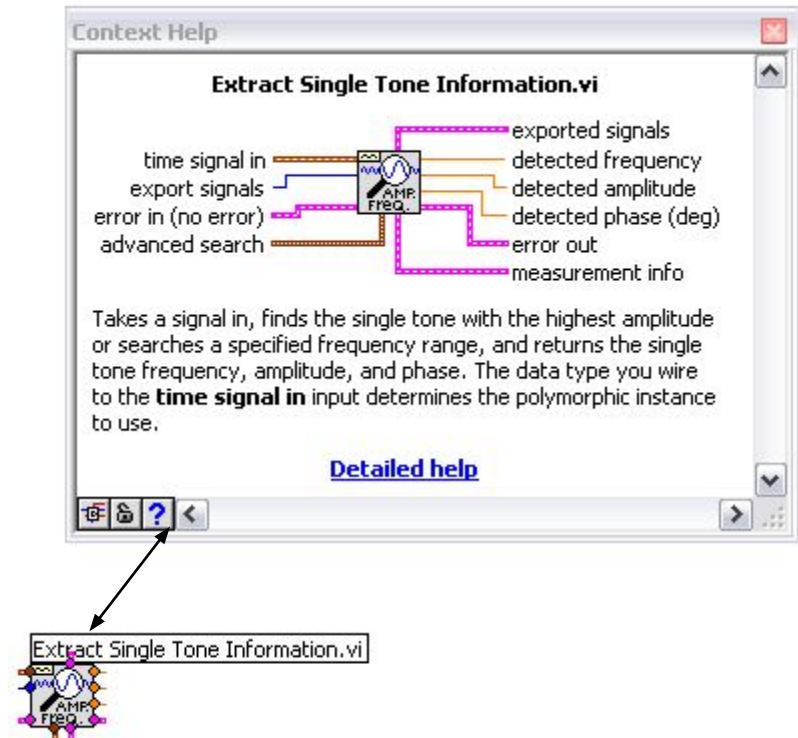


# Окно контекстной помощи

- **Help»Show Context Help**, нажать <Ctrl+N>
- Переместить курсор к объекту для обновления

## Дополнительная помощь

- Правая кнопка мыши по иконке ВП и выбрать **Help**, или
- Выбрать “**Detailed Help.**” на контекстном окне



# Горячие клавиши в LabVIEW

- <Ctrl+N> – Активация/деактивация окна Help
- <Ctrl+B> – Удаление оборванных проводников
- <Ctrl+E> – Переключение между лицевой панелью и блок -диаграммой
- <Ctrl+Z> – Отмена действий
- **Tools»Options...** – Настройки в LabVIEW
- VI Properties – конфигурация свойств ВП и т.д.

# Элементы типовой программы

## А. Циклы

- По условию
- С заданным числом итераций

## В. Функции и под ВП

- Виды функций
- Создание собственных функций (SubVI)
- Палитра функций и их поиск

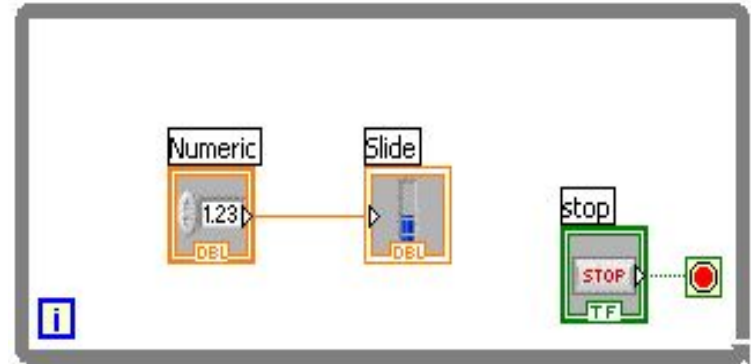
## С. Принятие решений и ввод / вывод файлов

### Структуры CASE


- Выбор
- Ввод и вывод файла

# ЦИКЛЫ

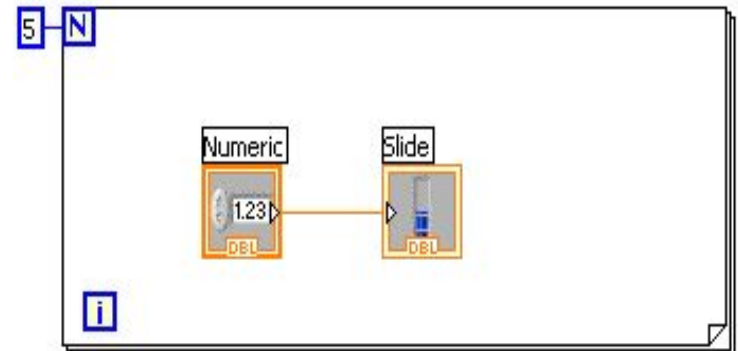
## While Loop




- По условию

- **i** Индикатор текущей итерации
- Всегда запускается как минимум один раз
- Исполняется до поступления на терминал условия выхода 

## For Loop

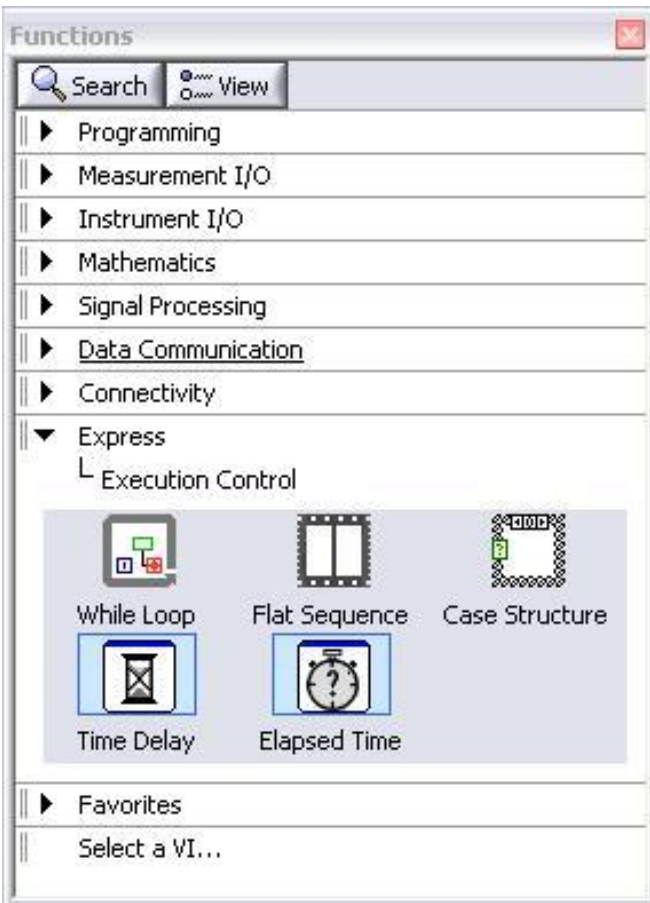


- С заданным количеством

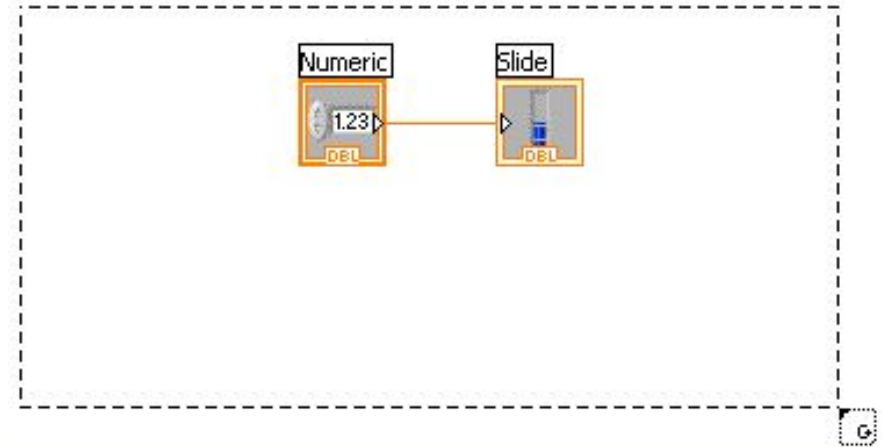
- **i** Индикатор текущей итерации
- Исполняется согласно значению счетчика **N** на терминале 

# Процесс заключения в цикл

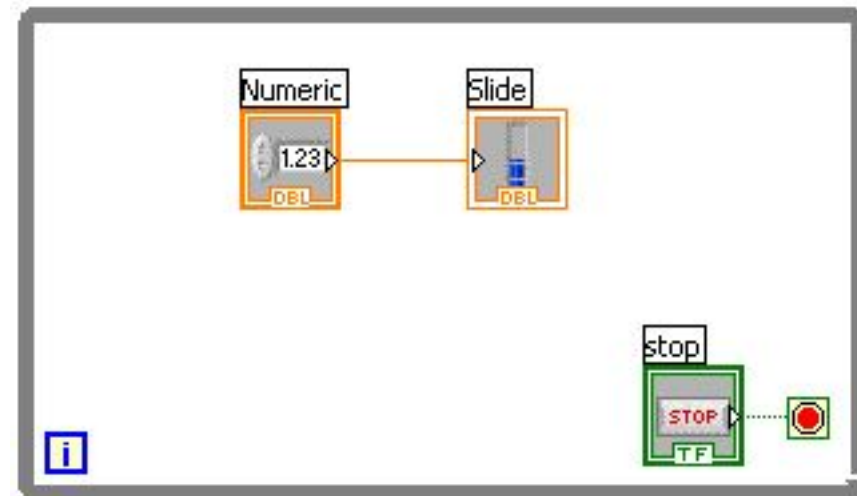
1. Выбрать из «Структуры»



2. Обвести вокруг заданного кода

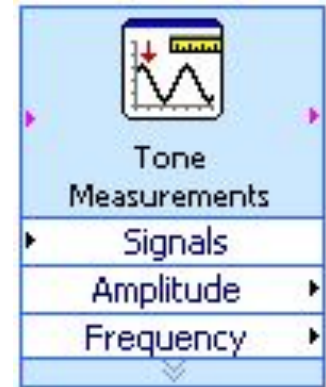


3. Или переместить код внутрь цикла и завершить соединение проводников

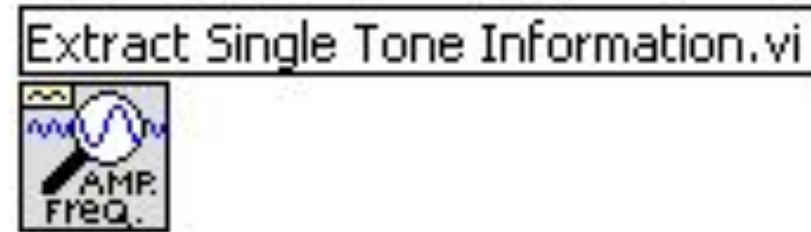


# Виды функций ( Functions Palette)

Экспресс : Функции с интерактивным окном конфигурирования (голубая иконка)



Стандартные : модульного типа ВП с возможностью переконфигурирования



Базовые : основные базовые функции LabVIEW; без лицевой панели и блок диаграммы (желтый цвет)



# Какие виды функций доступны?

## • Ввода и Вывода

## Палитра Экспресс-Функций

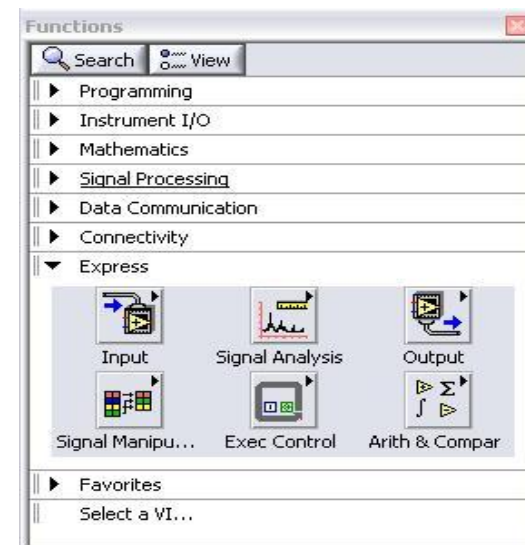
- Моделирования сбора и генерации данных
- Сбора и генерации реальных сигналов при помощи DAQ
- Управления приборами (последовательный и КОП (GPIB )
- Вызов ActiveX компонентов для связи с другими программами

## • Анализа и обработки

- Анализа и обработки различных типов сигналов
- Статистики
- Различных разделов математики
- Дополнительных временных

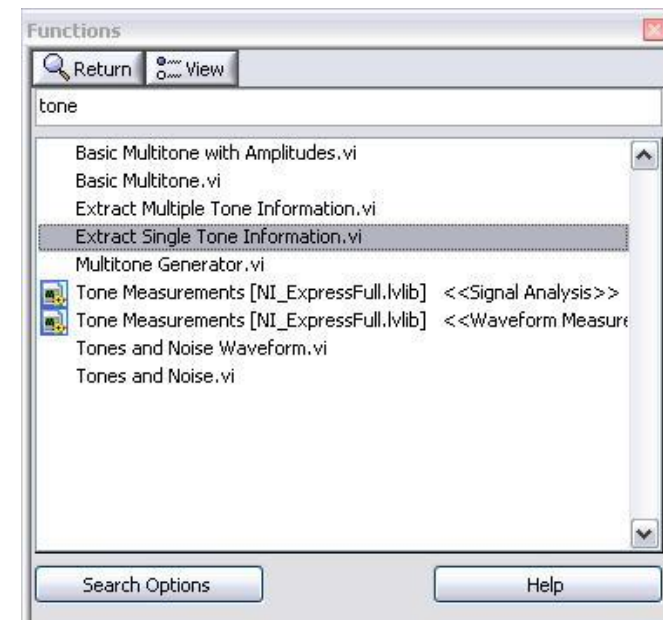
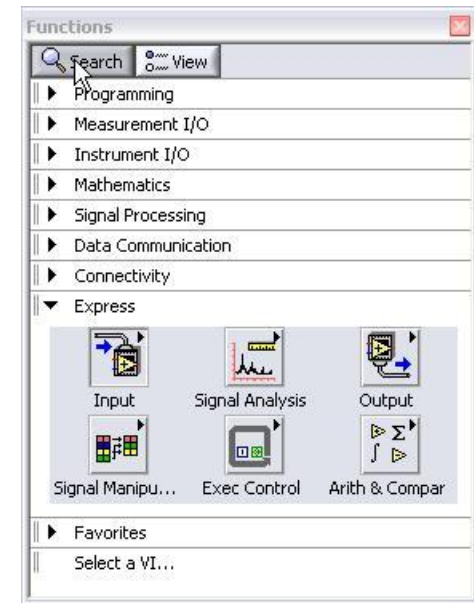
## • Сохранения и записи

- Ввода и вывода файлов



# Как найти функции в палитрах?

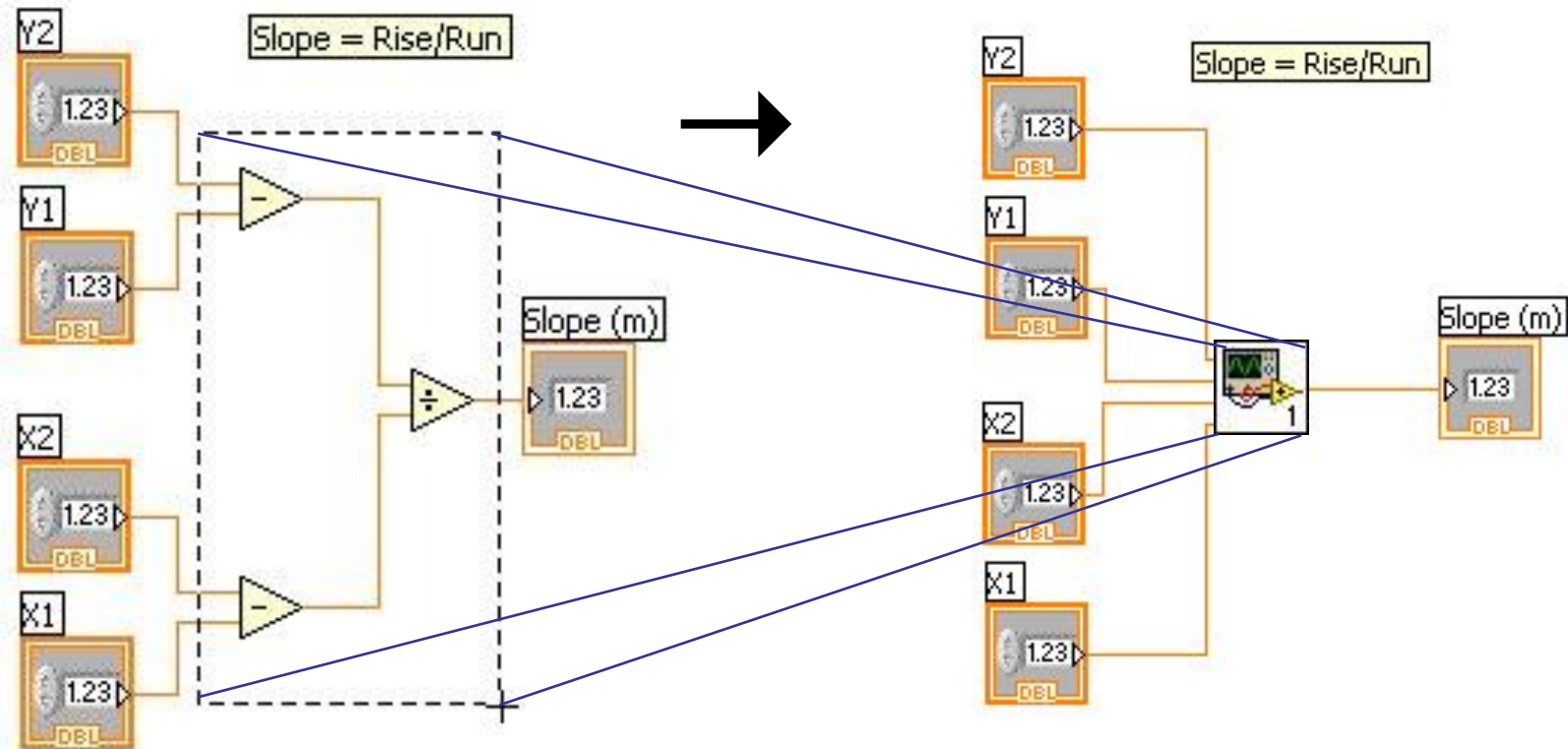
- Палитры содержат сотни функций
- Нажать кнопку «search» на строке палитры и ввести название функции
- Нажать и выбрать нужную функцию из списка
- Дважды нажать на иконку функции для отображения подпалитры
- Перетащить нужную функцию на блок-диаграмму





# Создание под ВП (SubVI)

- Выделить код на блок-диаграмме для создания под ВП
- Выбрать из Меню **Edit** «**Create SubVI**»



# Порядок работы функций в ВП

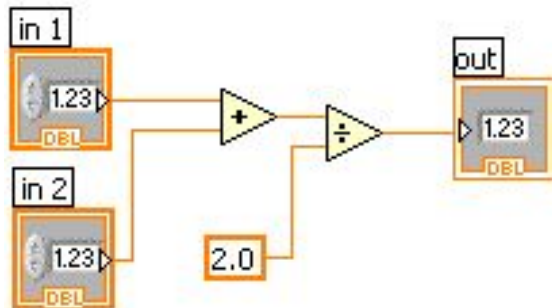
## Function Pseudo Code

```
function average (in1, in2, out)
{
  out = (in1 + in2)/2.0;
}
```

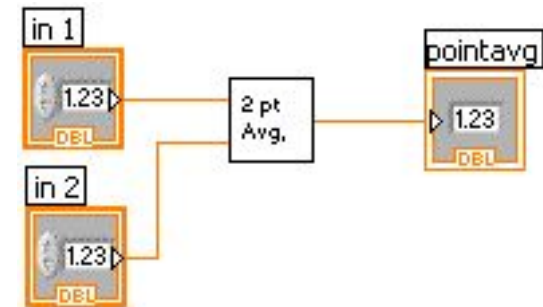
## Calling Program Pseudo Code

```
main
{
  average (in1, in2, pointavg)
}
```

## Под ВП на блок- диаграмме

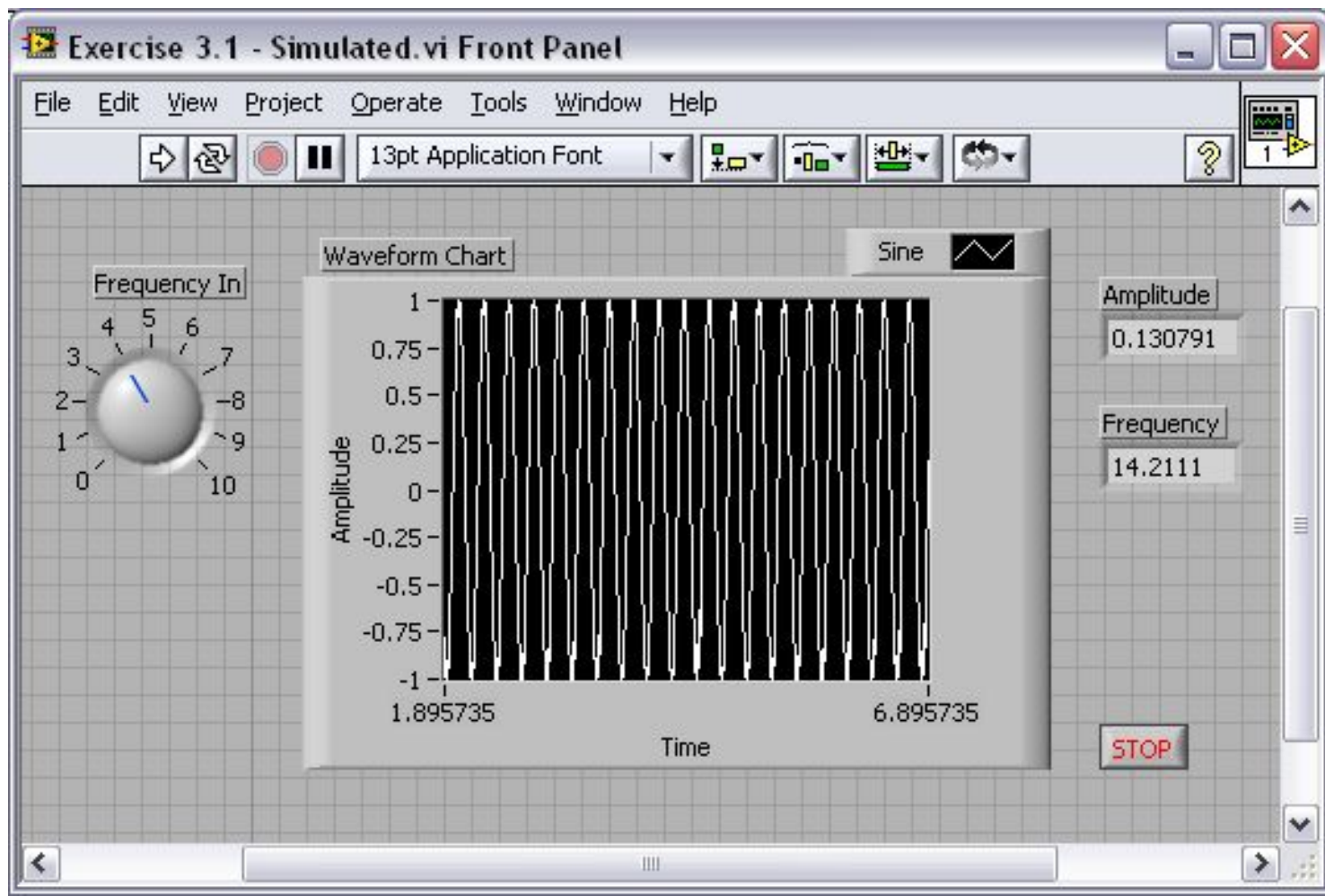


## Вызов ВП на блок- диаграмму



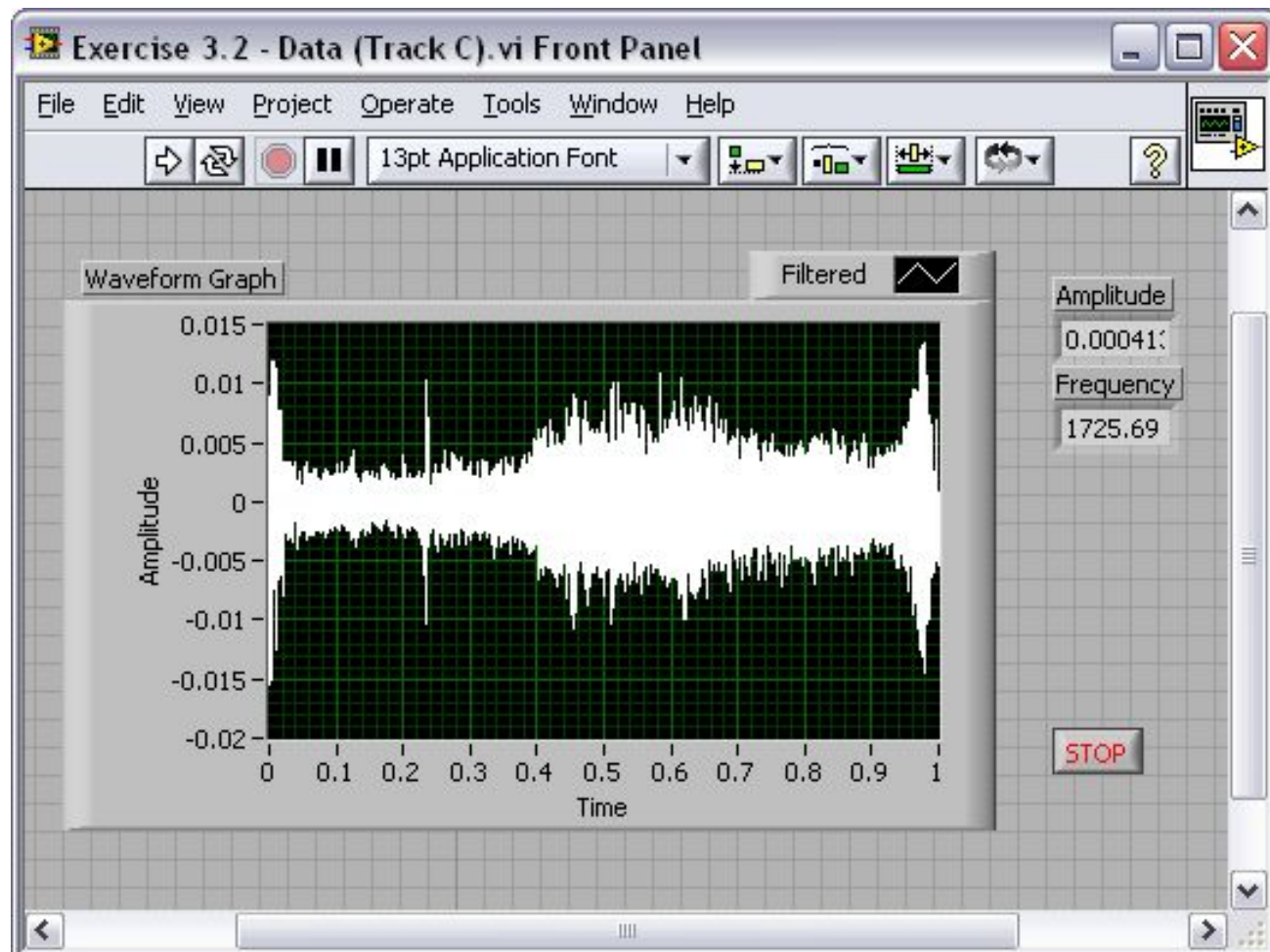
# Анализ данных

- Используем функции Экспресс для: генерации и отображения амплитуды и частоты сигнала



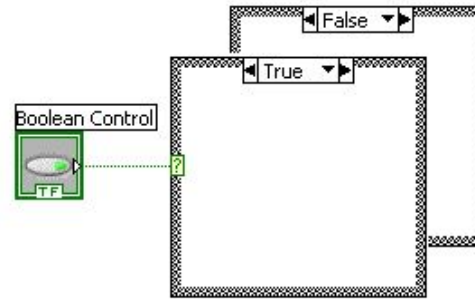
# Анализ данных

- Используем функцию Экспресс для:
  - Сбора и отображения амплитуды и частоты сигнала

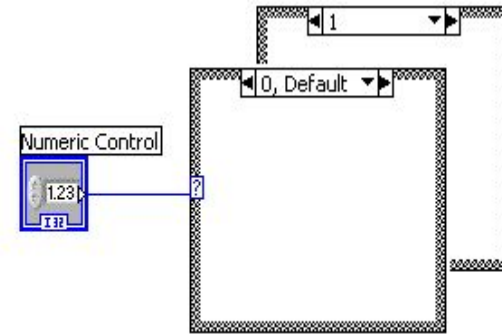


# Принятие решений в LabVIEW

## 1. Структуры «Case»

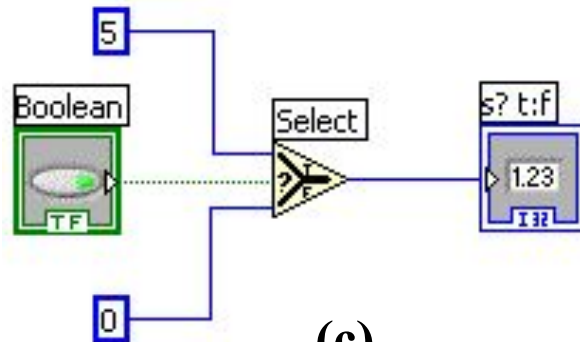


(a)



(b)

## 2. Функция «Выбор»

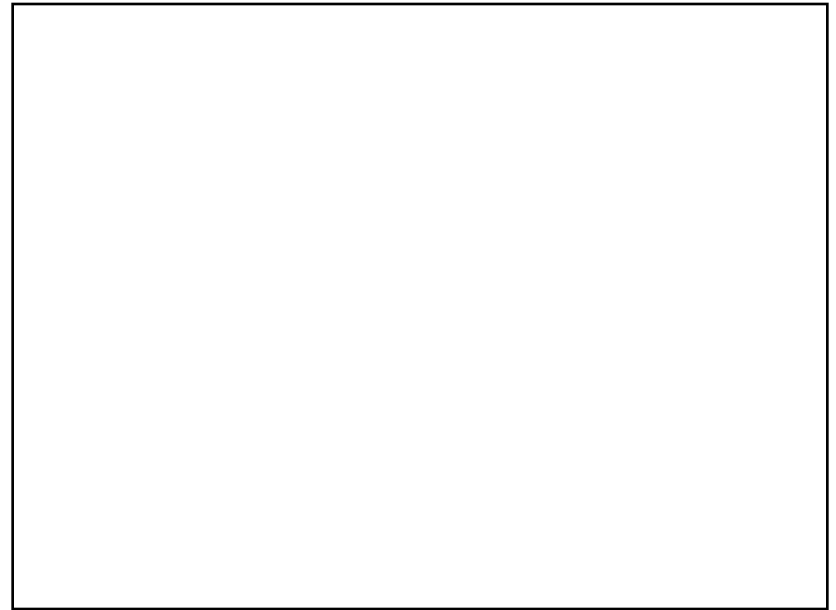
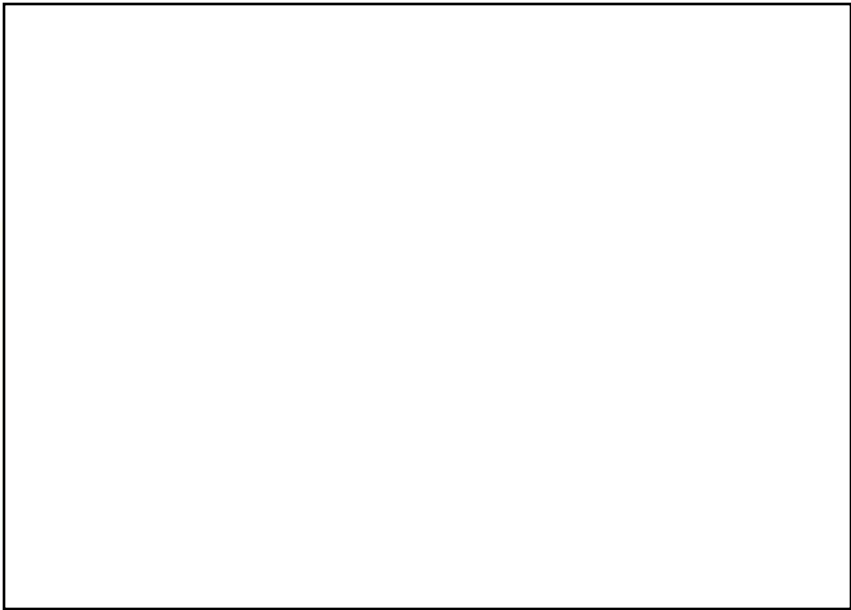


(c)

# ФАЙЛОВЫЙ ВВОД/ВЫВОД

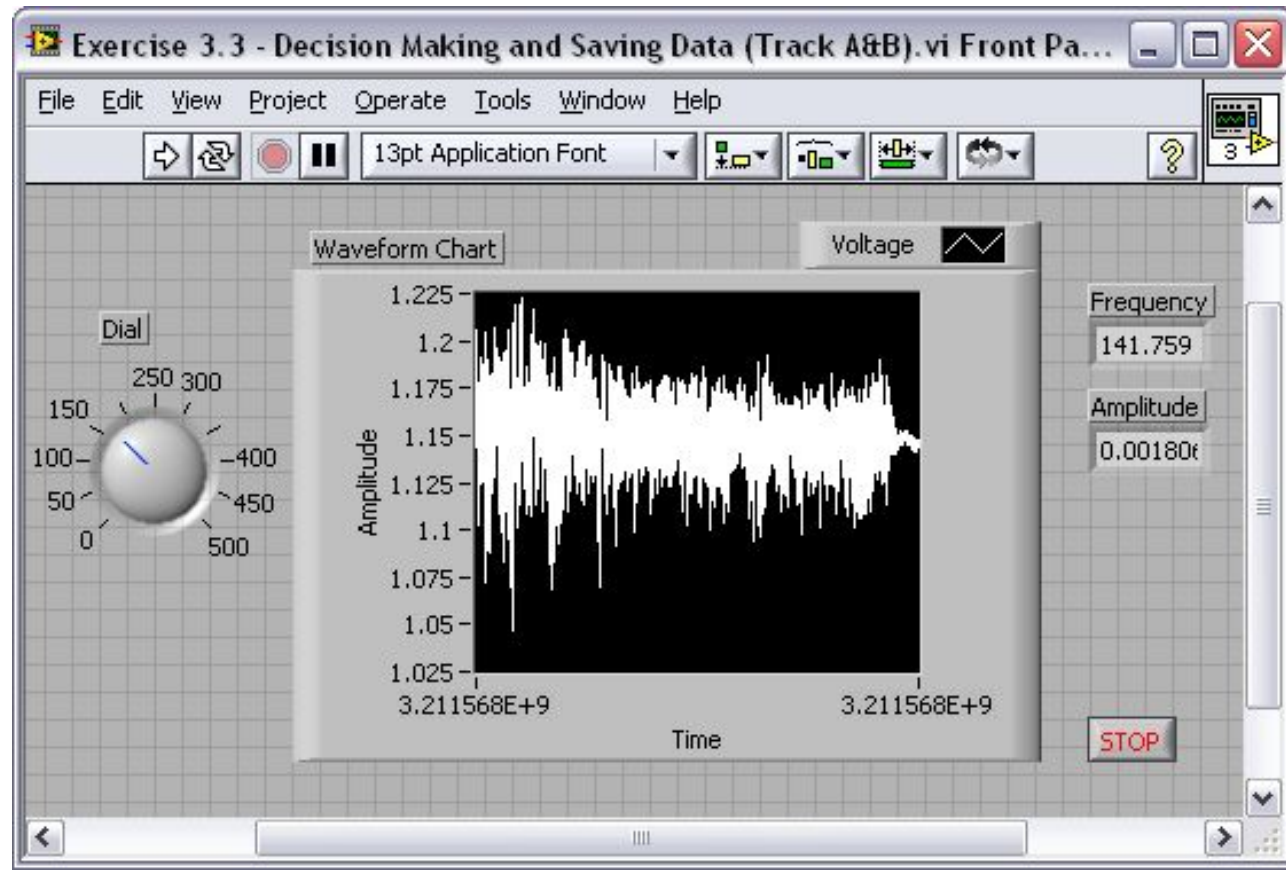
**File I/O** – передача данных в/из файлов

- Файлы могут быть в виде бинарных, тестовых или таблиц
- Запись/Чтение LabVIEW Measurements file (\*.lvm)

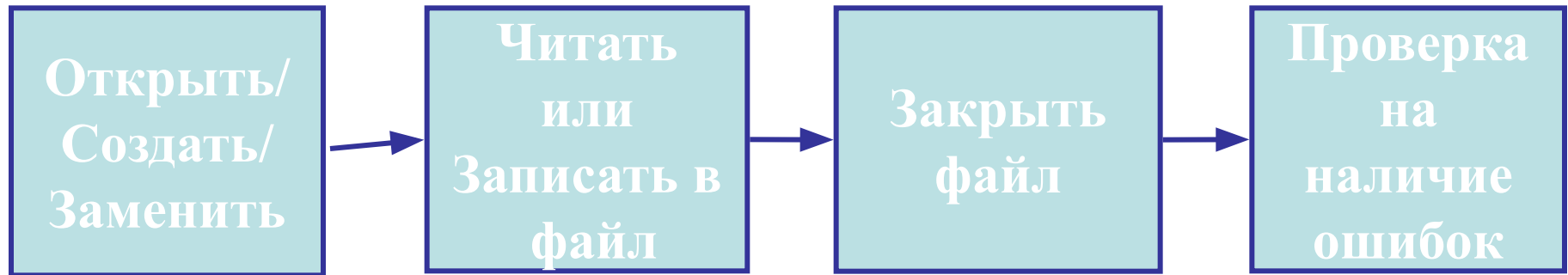


# Принятие решений и сохранение данных

- Используем структуру «case» для:
  - Сохранения данных при выполнении определенного условия



# Запись/чтение в /из файла



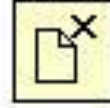
Open/Create/Replace File



Read from Text File



Close File



Simple Error Handler.vi



Write to Text File





# Отображение результатов

## А. Отображение данных на лицевой панели

- Элементы управления и индикаторы
- Графики
- Временные циклы

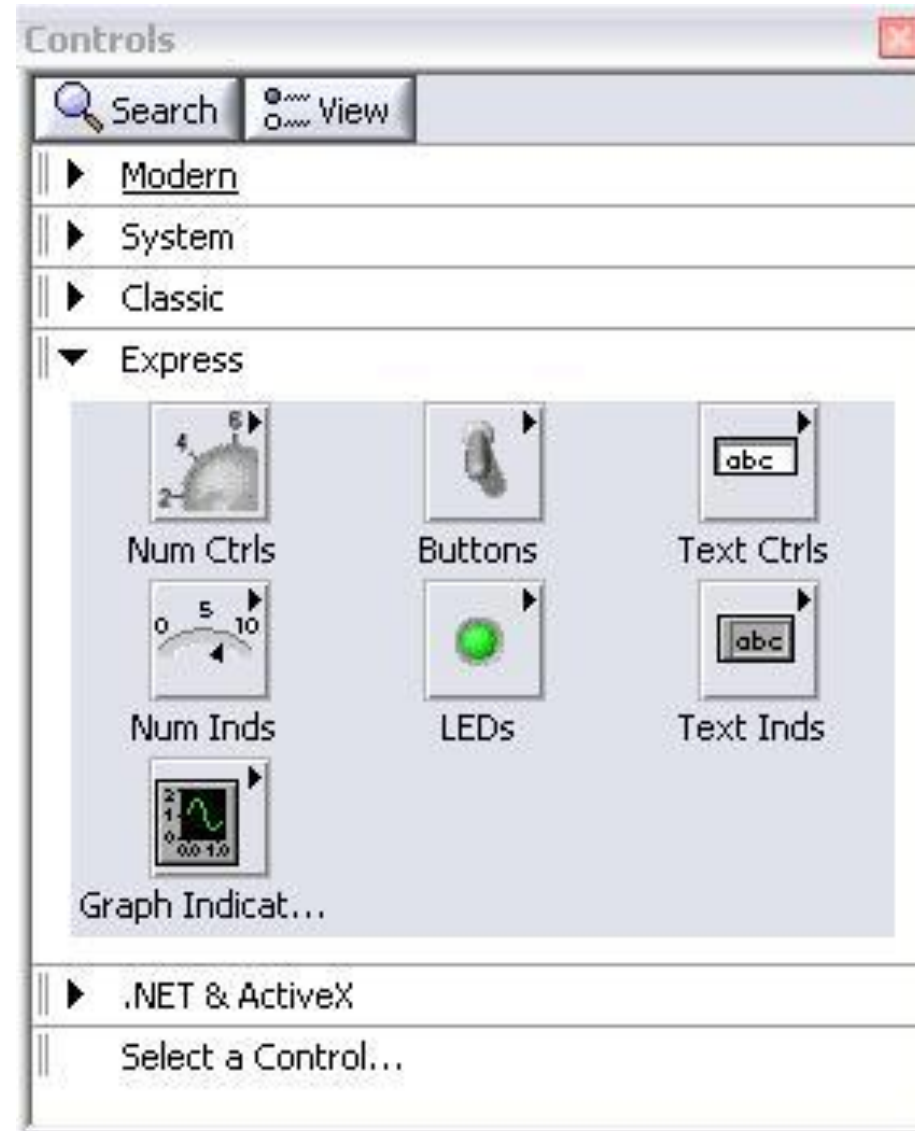
## В. Обработка сигналов

- MathScript ( математические скрипты)
- Массивы
- Кластеры
- Элементы сигнала

# Типы элементов управления и индикаторов

- **Числовые типы данных**
  - Ввод и отображение числовых данных
  - Слайдеры и ручки
- **Логические типы данных**
  - переключатели и диоды
- **Массивы и матрицы**
  - Числовые
  - Графики диаграмм
  - Графики осциллограмм
  - Графики интенсивности
  - Графики объемных изображений, поверхностей и моделей
- **Декоративные панели**
  - Переключаемые панели
  - Стрелки
- **Другие элементы и типы данных**
  - Строчные и текстовые данные
  - Картинки/отображение фотографий
  - Компоненты ActiveX

## Палитра элементов

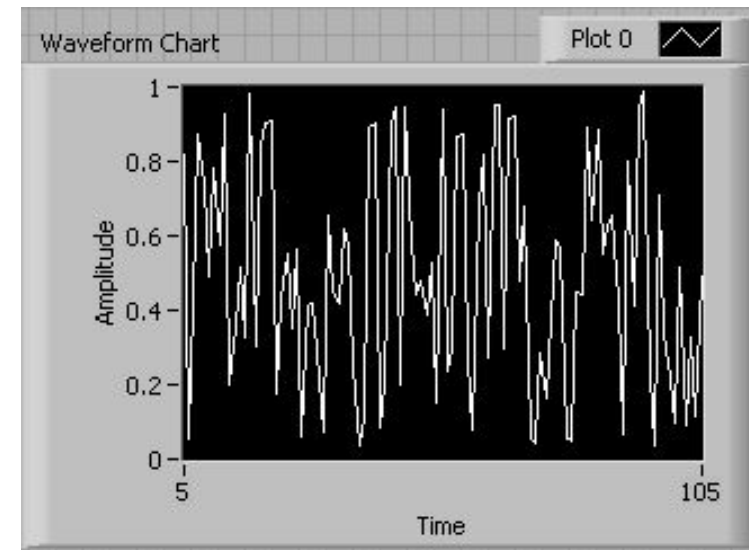
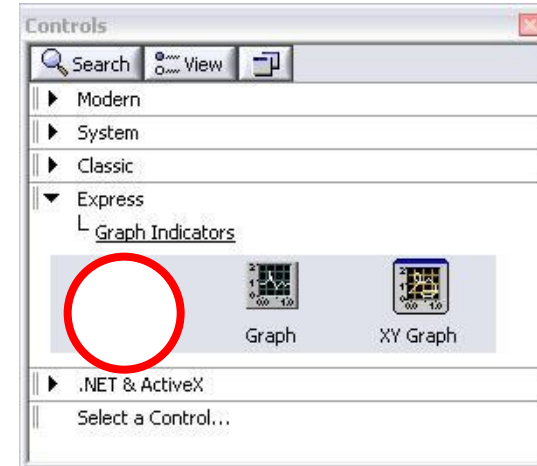
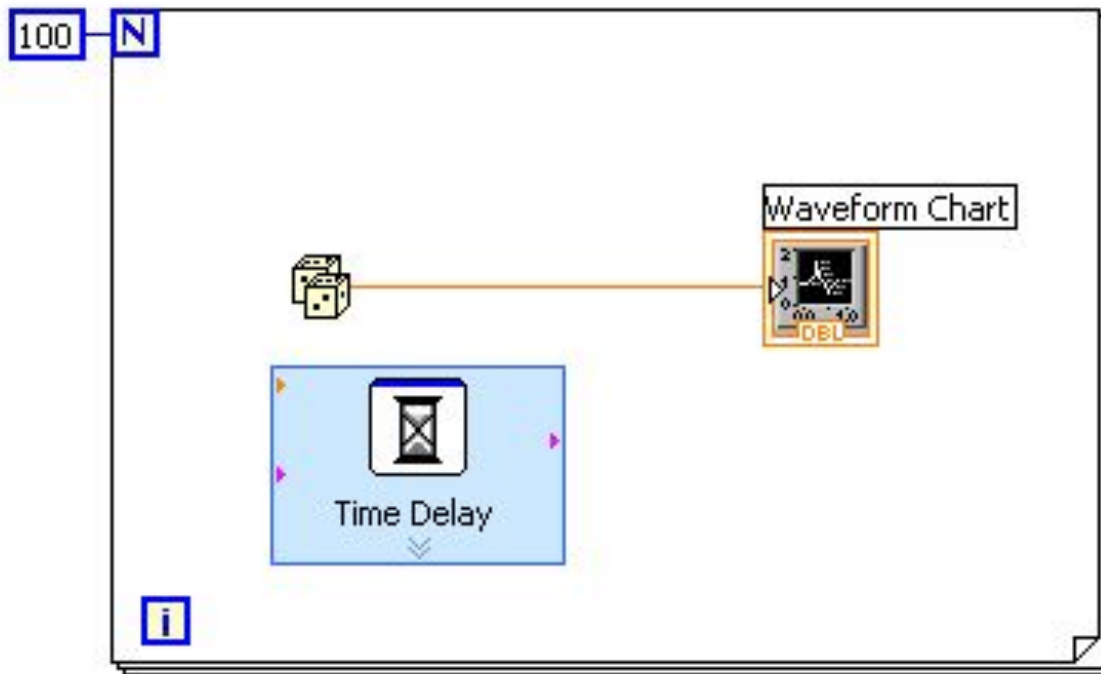


# Графики диаграмм

Графики диаграмм – Специальный числовой индикатор, отображающий собранные данные во времени

- Обнов. термин

Функции»Экспресс»Graph Indicators» Chart

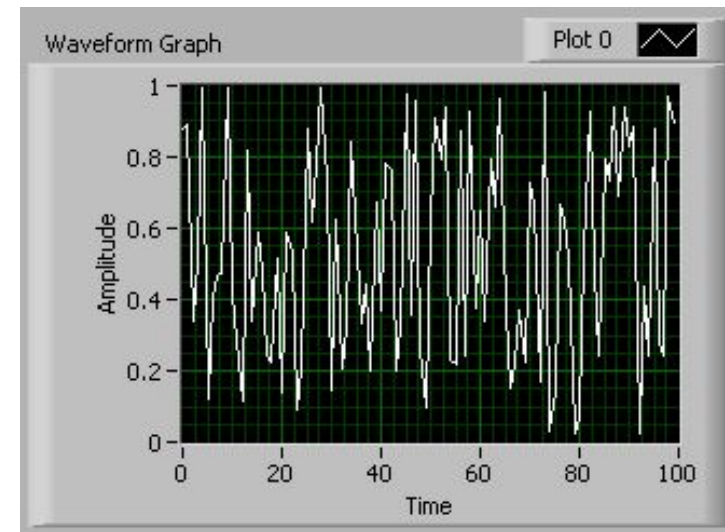
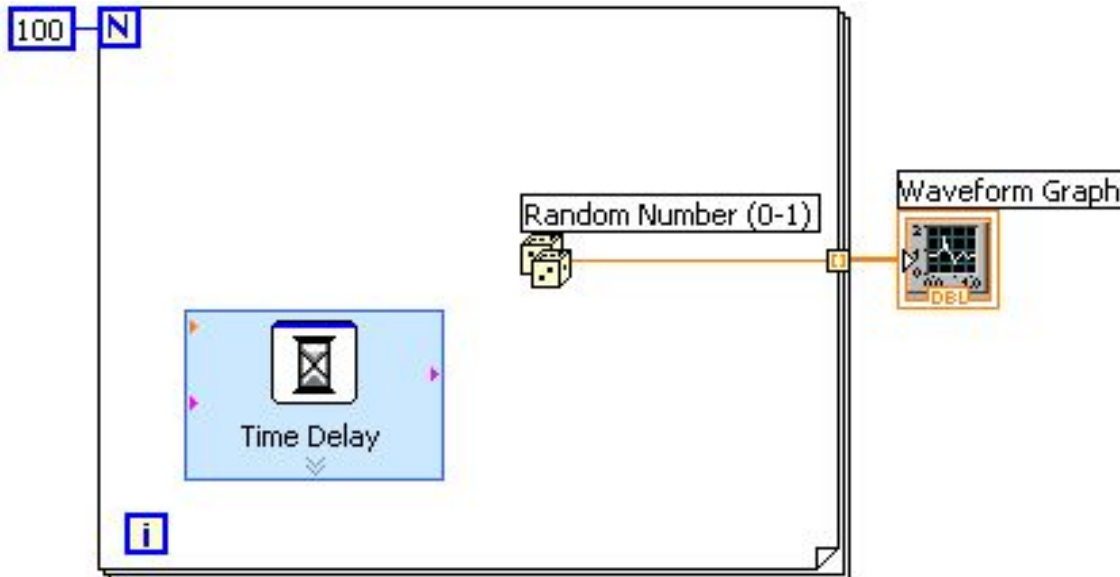
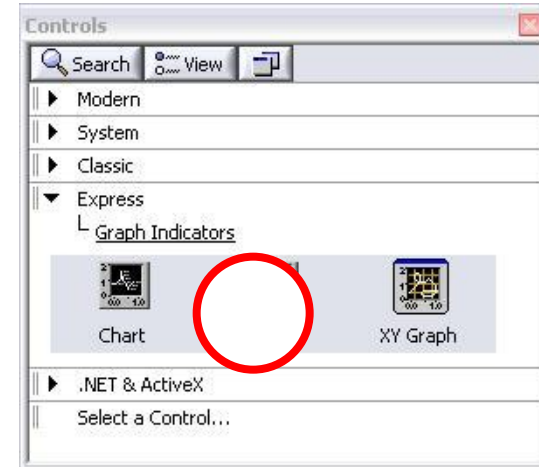


# Графики осциллограмм

Графики осциллограмм – специальный числовой индикатор

- Обновление буферизация
- Цикл мс

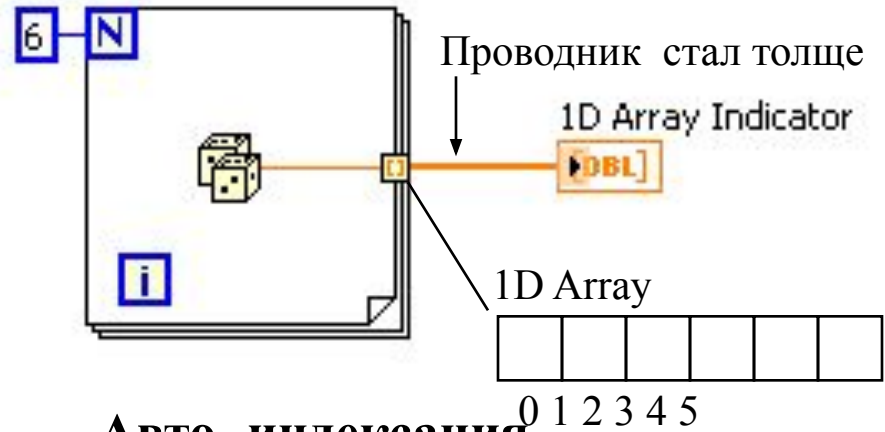
Functions»Express»Graph Indicators»Graph



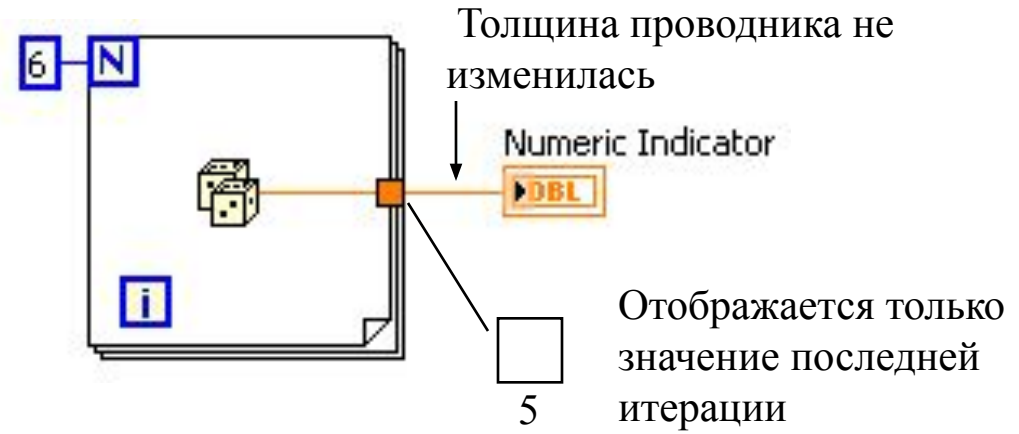
# Создание массивов с использованием Циклов

- Циклы могут аккумулировать массивы на границах (автоиндексация)
- В цикле For Loops автоиндексация по умолчанию
- Правая кнопка мыши по границе цикла активирует/деактивирует индексацию

## Авто-индексация включена

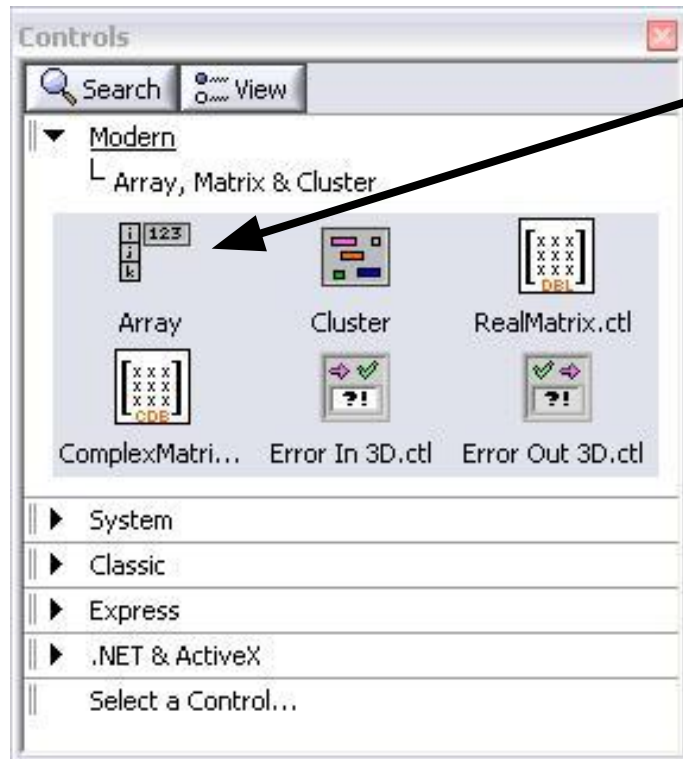


## Авто-индексация отключена



# Создание массива

Выбираем с палитры **Controls»Modern»Array, Matrix, and Cluster** иконку контейнера массива.

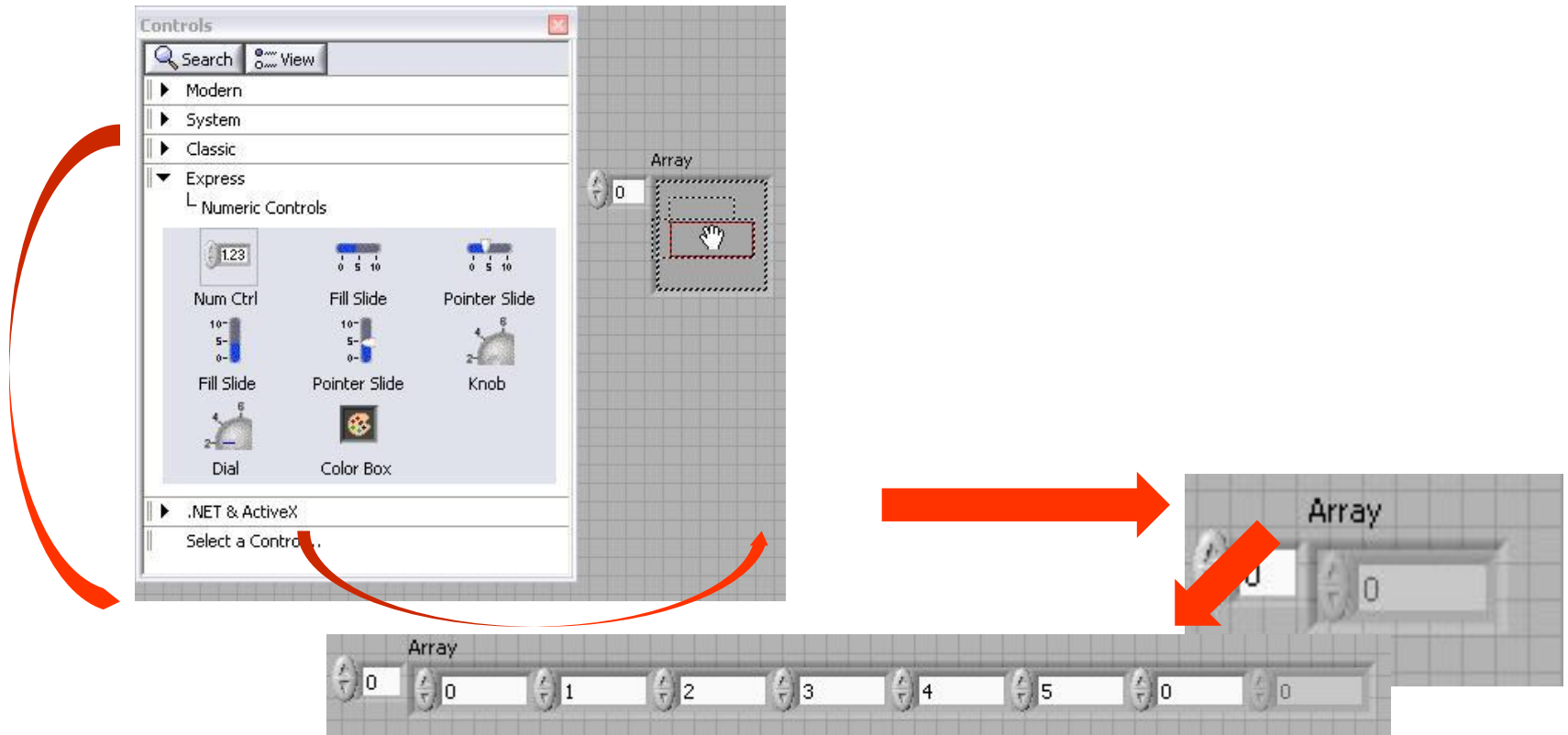


Помещаем на лицевую панель



# Создание массива

1. В контейнер массива помещаем нужный нам элемент управления или индикатор



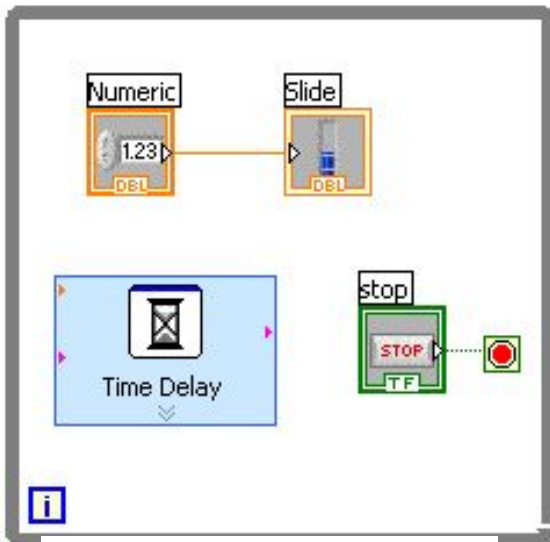
# Для чего нужен временной цикл?

## 1. Задержка времени исполнения кода

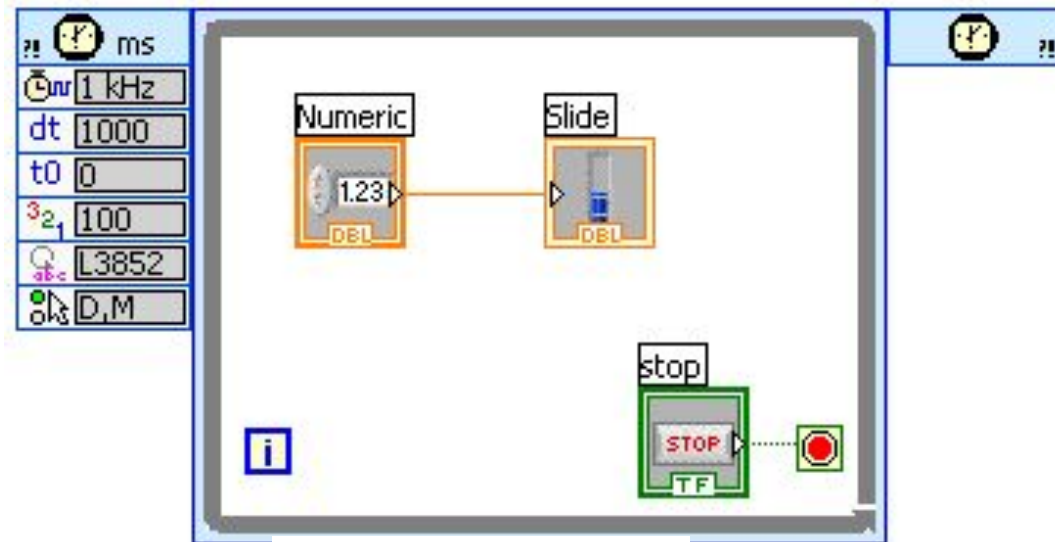
- Настройка Time Delay Express VI для работы кода с определенным промежутком времени (работа с For and While циклами).

## 2. Временной цикл

- Настройка временного цикла While loop для заданного  $dt$ .



Задержка времени



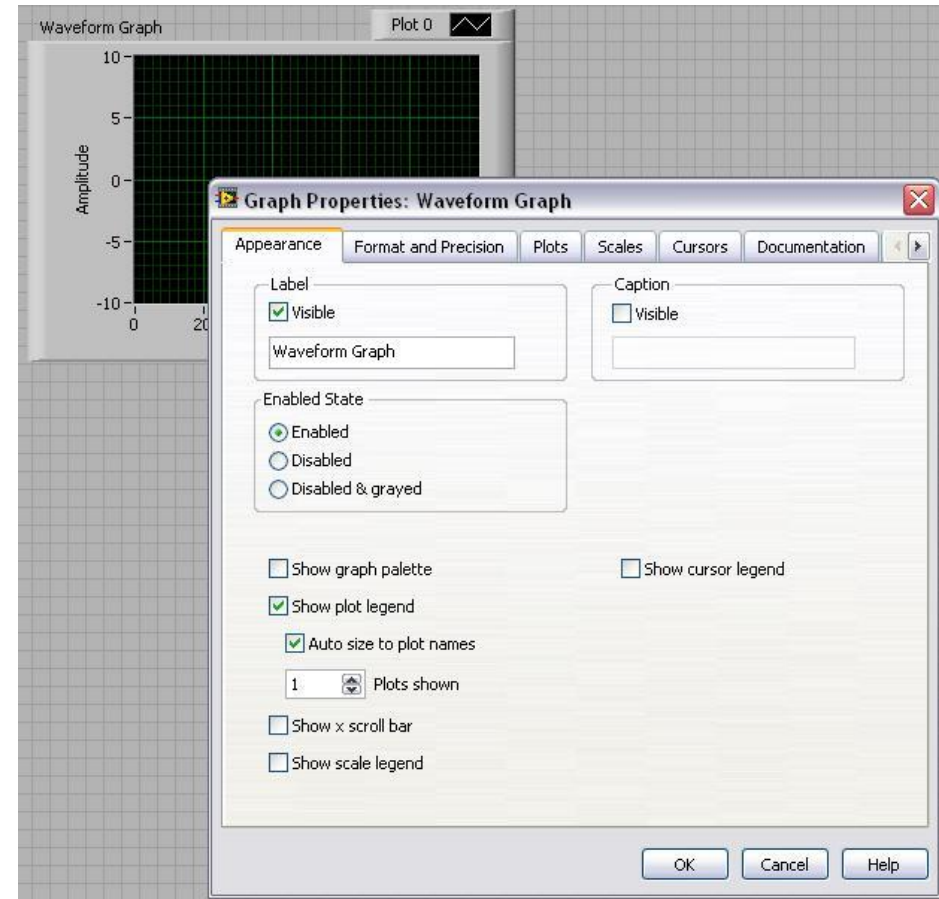
Временной цикл



# Свойства элементов управления и индикаторов

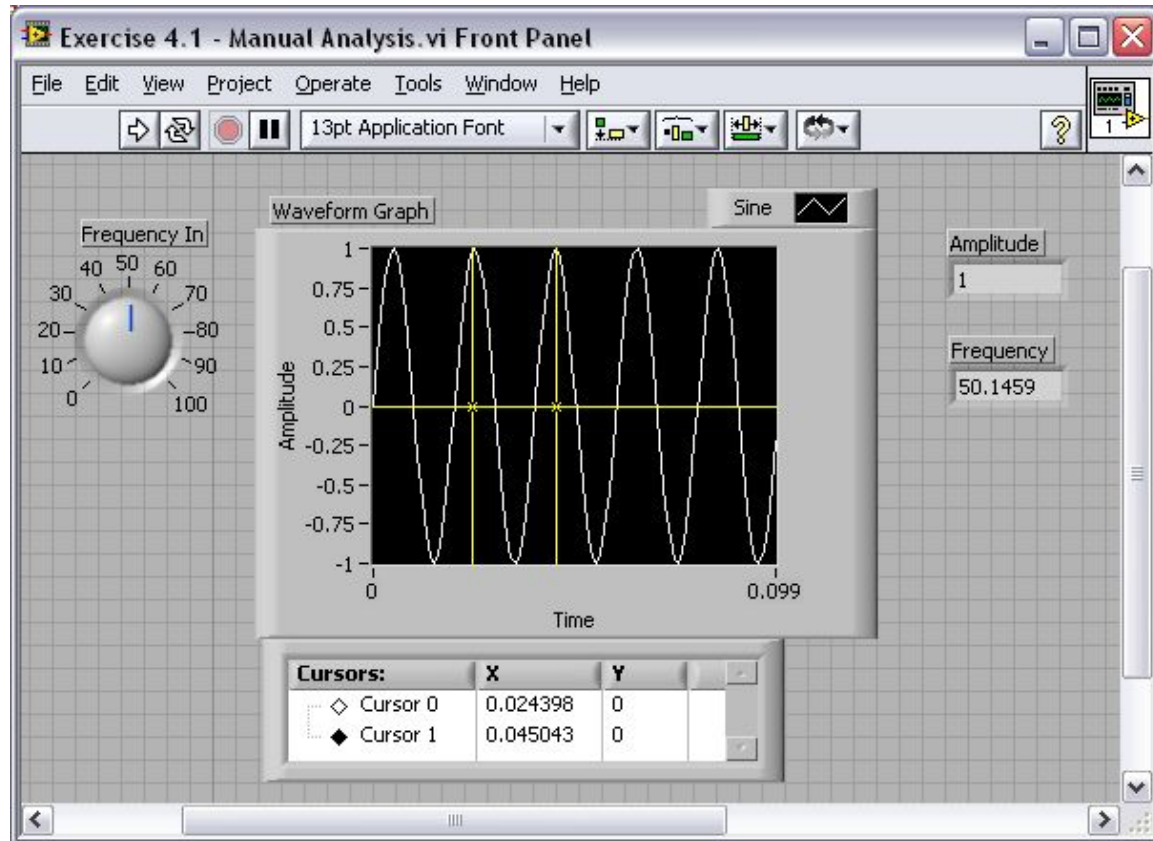
- Свойства характеризуют настройки объектов
- Щелчок правой кнопкой мыши по объекту
  - Свойства объектов:

- размер
- цвет
- настройки шкал
- цвет шкал
- курсор
- и т.д.

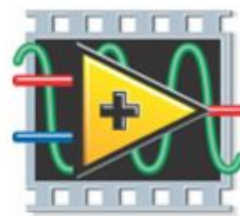


# Анализ сигнала

- Используя курсор графического индикатора:
  - Измерить частоту и амплитуду синусоиды



# Mathscript



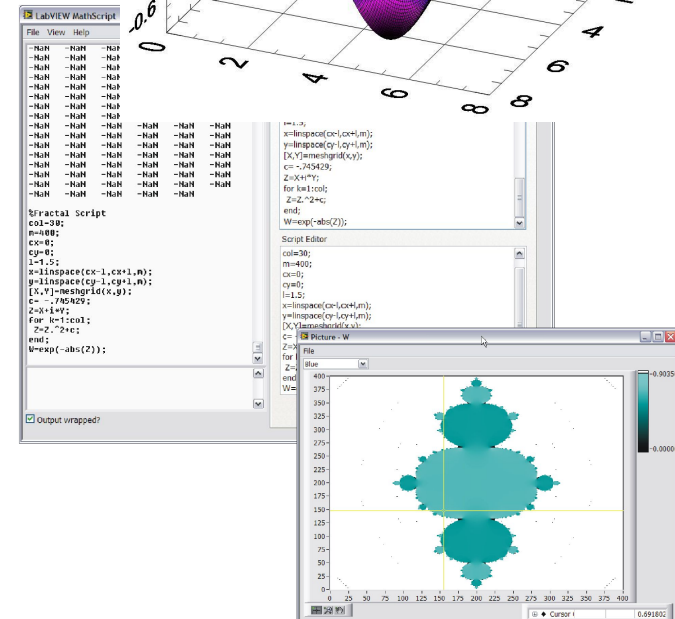
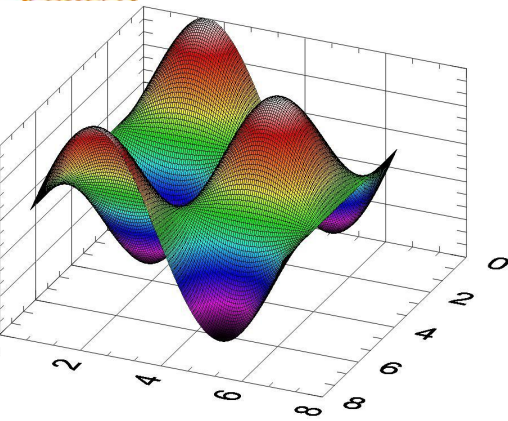
NATIONAL INSTRUMENTS  
**LabVIEW™**

## LabVIEW MathScript

- Возможности разработки приложений обработки сигналов, анализа и математических уравнений в текстовом коде среды LabVIEW
  - Более 500 встроенных функций
  - Синтаксис схожий с популярными языками математического моделирования
  - Встроенные в LabVIEW основанный на оригинальной среде математической разработки MATRIXx

MathScript Node

```
tic;  
Y=fft(X);  
PowerY=abs(Y).^2;  
toc;
```



# Математические скрипты в LabVIEW

- Интеграция существующих скриптов для быстрой разработки приложений в LabVIEW
- Интерактивность и легкость разработки,
- Разработка математических алгоритмов, обработка и анализ в одной среде LabVIEW
- Возможность интеграции математического и графического синтаксиса в одном ВП

## **Поддерживаемые интерфейсы:**

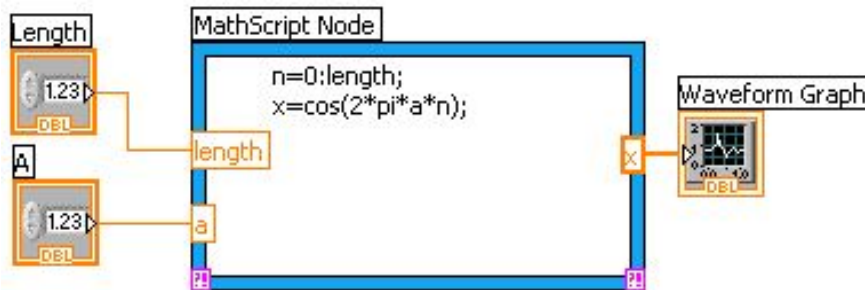
Узел MathScript script node MathSoft software

Mathematica software      MATLAB<sup>®</sup> software

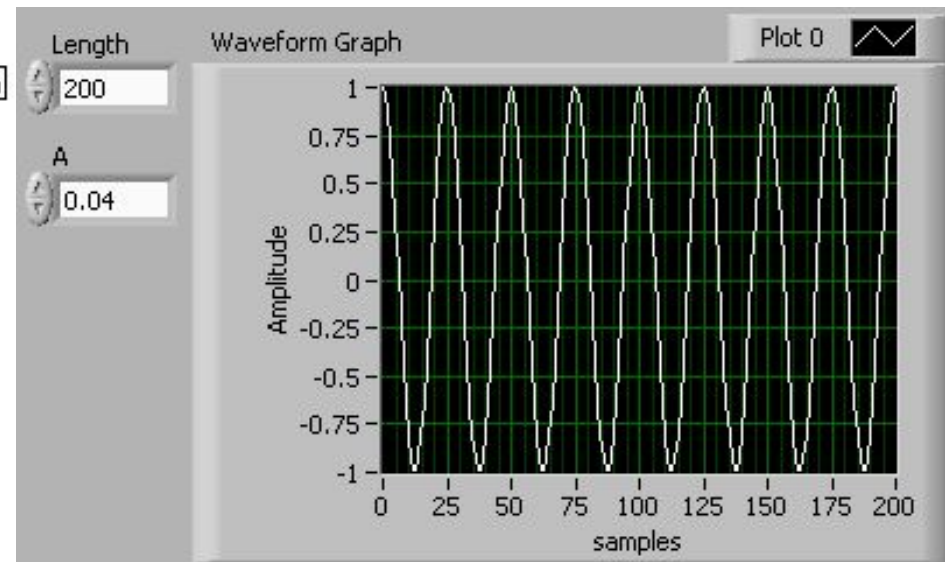
Maple software              Xmath software

# Математический код в MathScript Node

- Введение уравнений в текстовом формате
- Входные и выходные переменные на границах цикла
- Синтаксис схожий с математическими текстовыми языками
- Богатая библиотека готовых решений

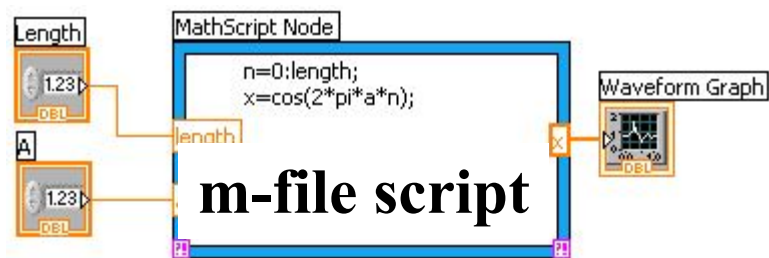


(Functions >> Programming >>  
Structures >> MathScript)



# Интерактивное окно MathScript

- Быстрая разработка и алгоритмы отладки
- Создание и редактирование переменных
- Отображение в 1D, 2D, и 3D .



Скриншот интерфейса LabVIEW MathScript. Вывод данных:

Global	Local	Dimension	Type
f	f	1x1	double array
n	n	1x201	double array
	n	1x201	double array

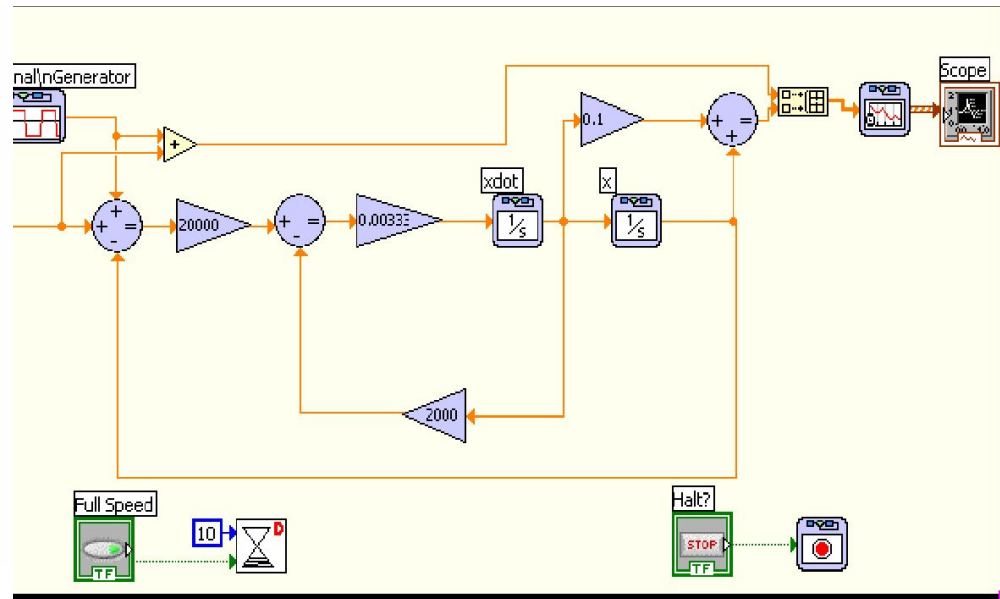
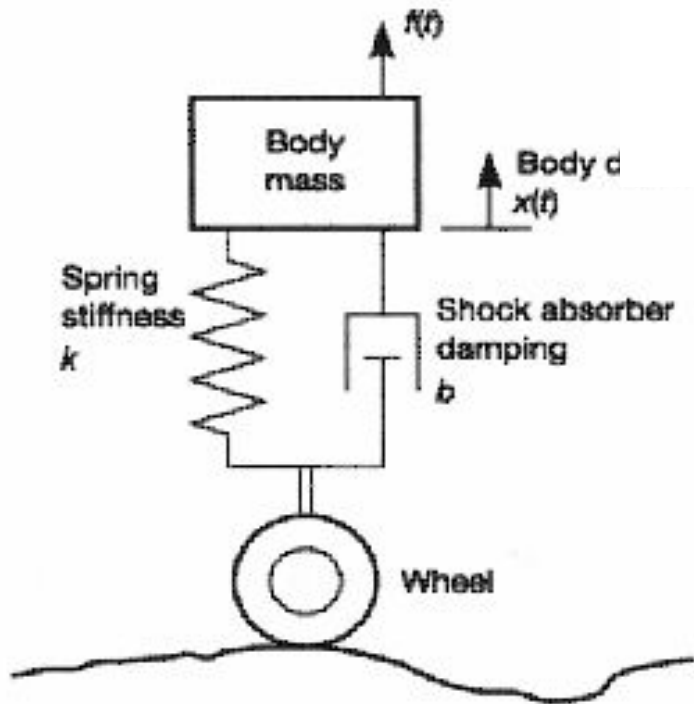
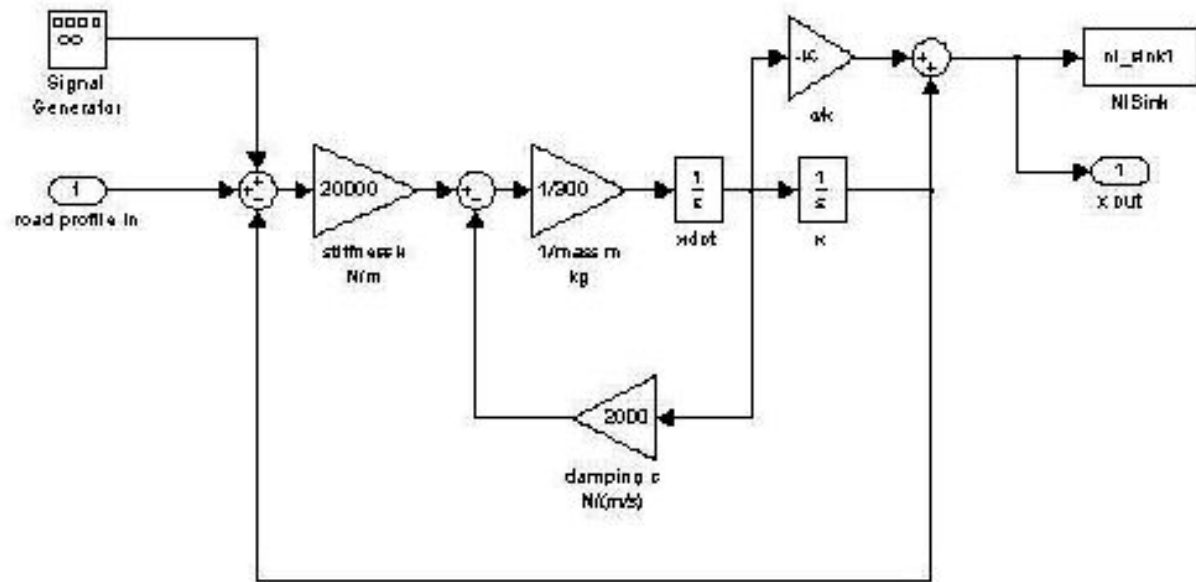
Командное окно: `x=cos(2*pi*f*n)`

Отображение в форматах

**Командное  
ОКНО**  
(LabVIEW >>Tools >> MathScript  
Window)

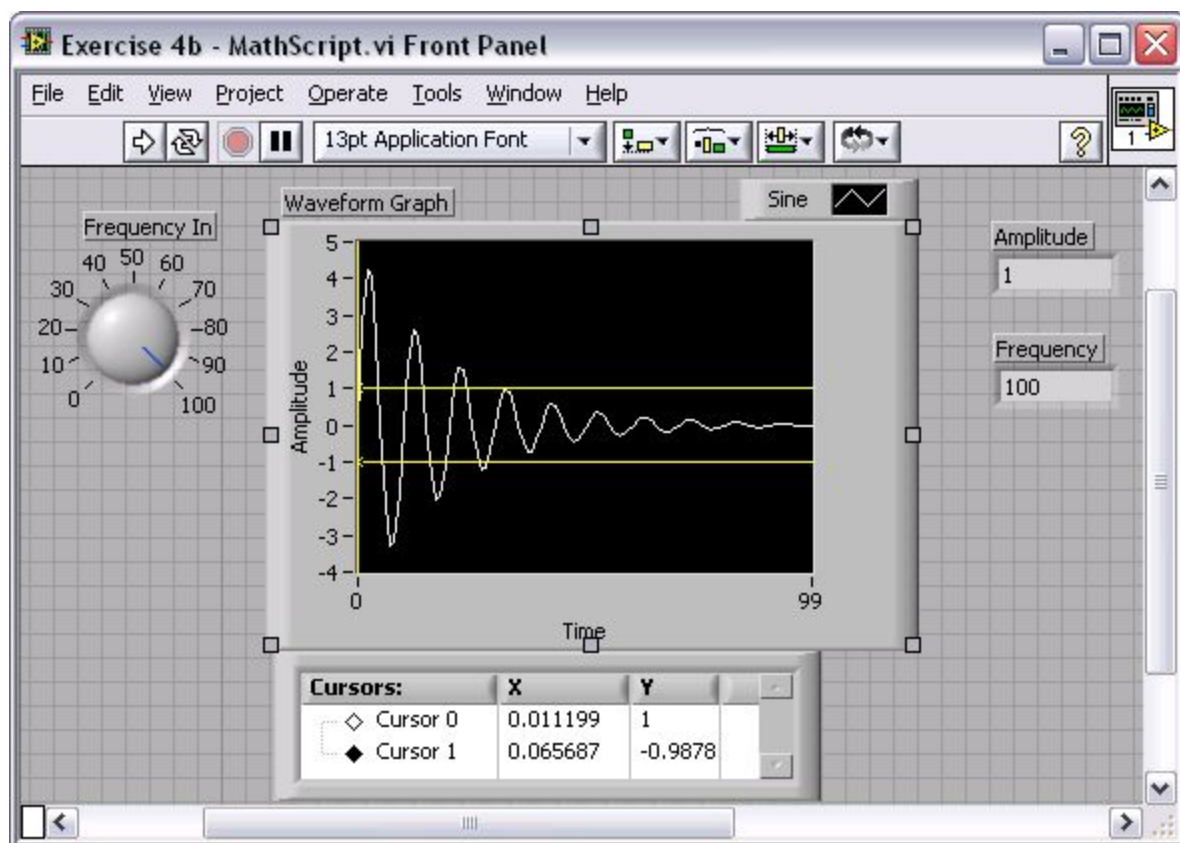


# LabVIEW и Simulink



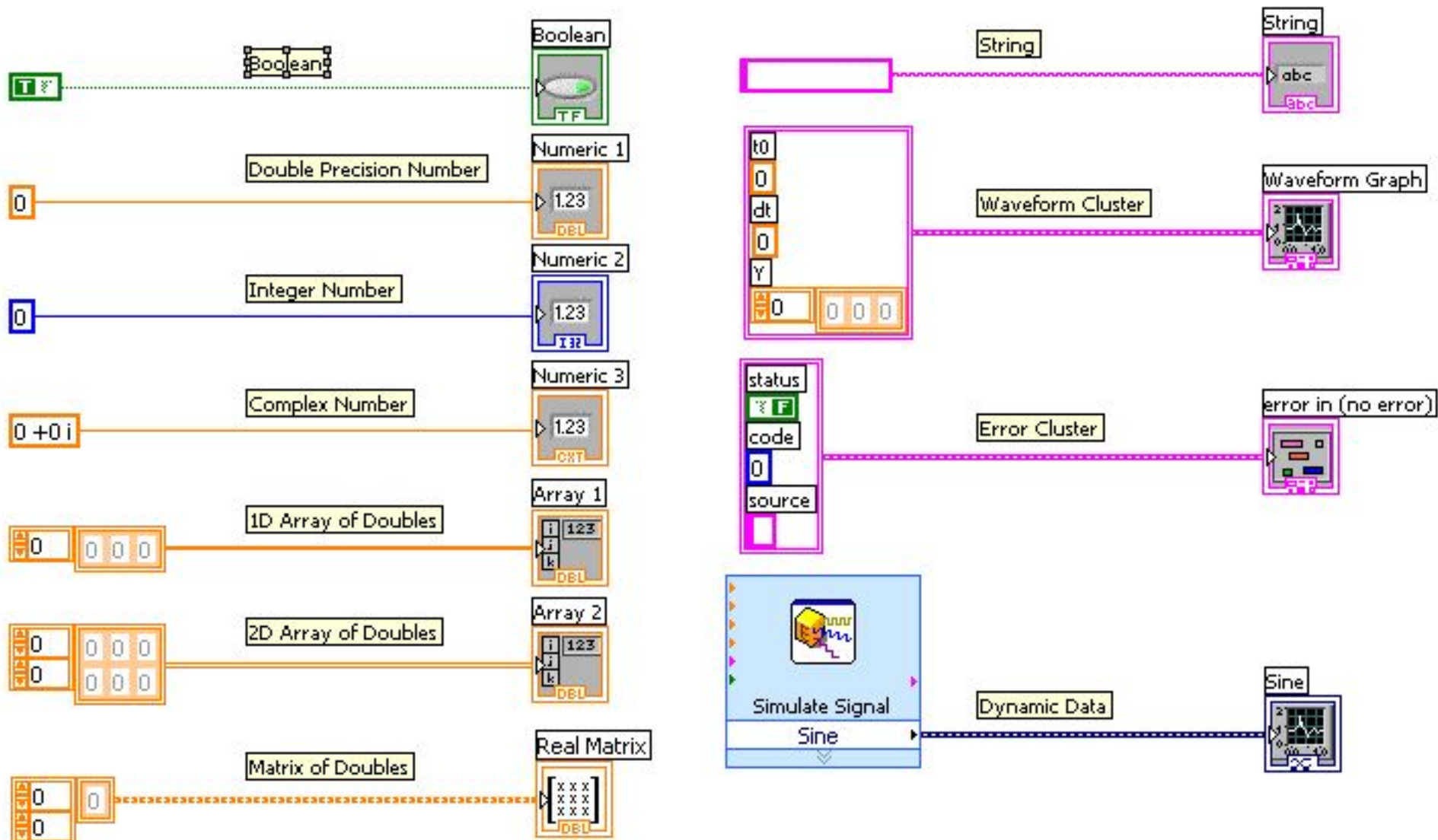
# Использование Script

Используя узел MathScript and Интерактивное Окно MathScript для обработки и сохранения данных.

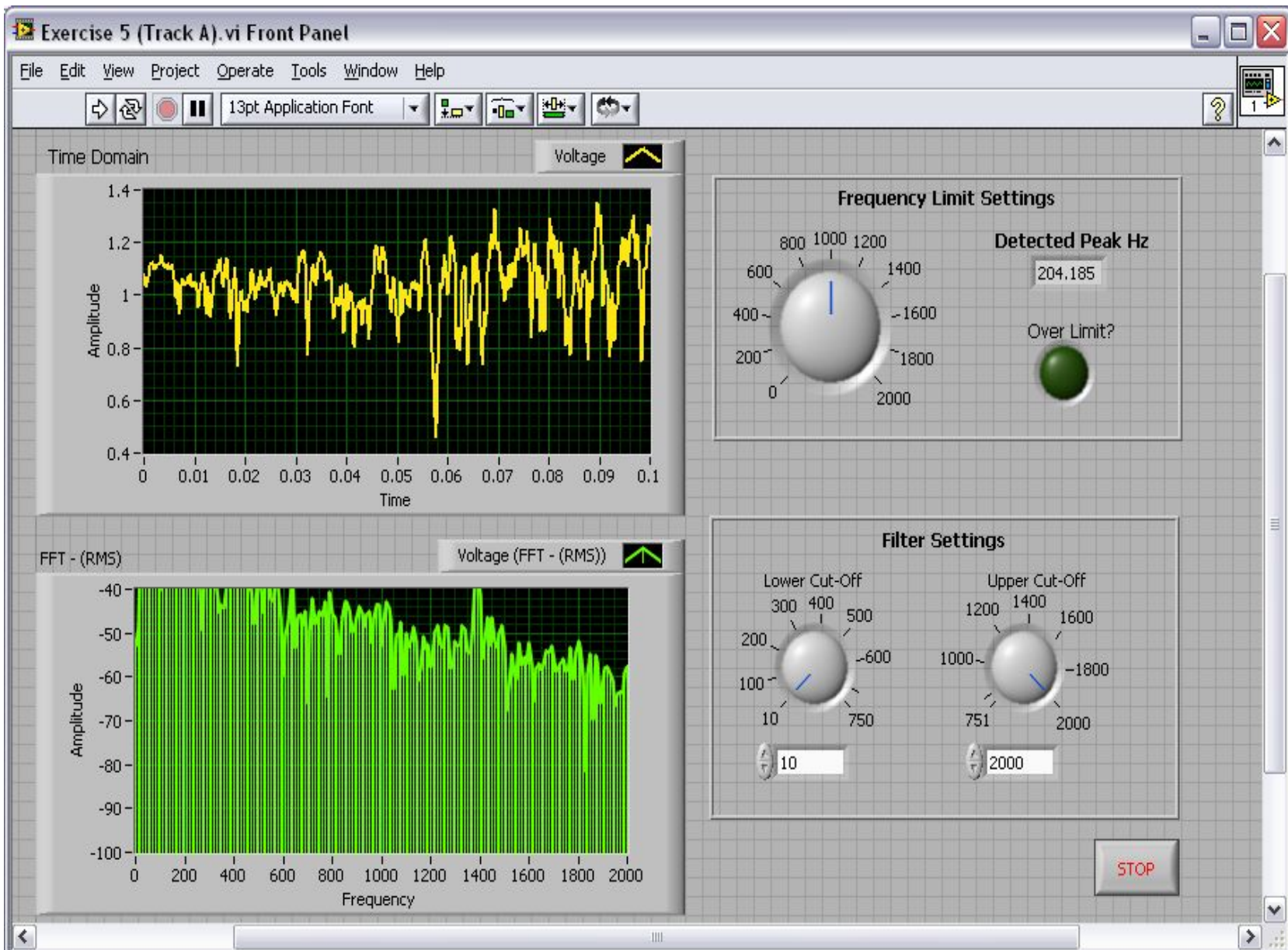




# Обзор типов данных в LabVIEW



# Комплекс возможностей



# Дополнительные возможности

## А. Дополнительные типы данных

- Кластеры

## В. Управление потоком данных

- Сдвиговые регистры
- Локальные переменные

## С. Разработка больших приложений

- Окно навигации
- Проект в LabVIEW

# Кластеры

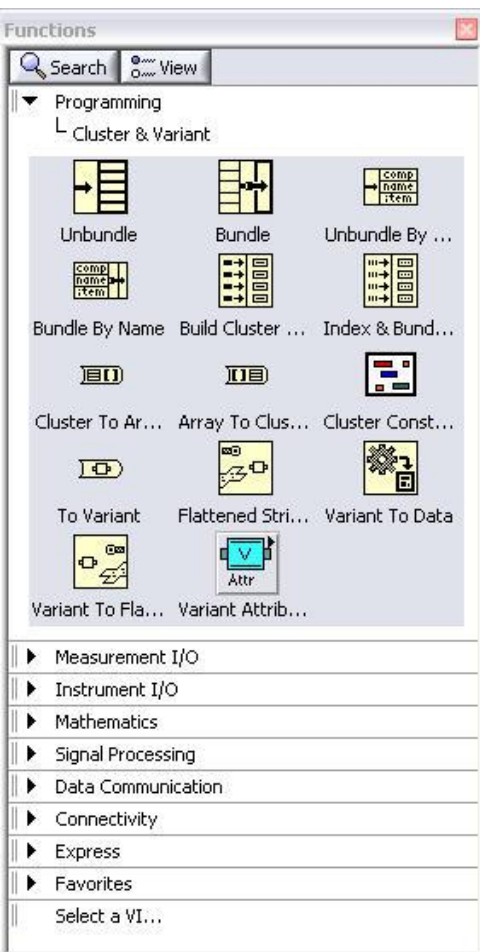
- Объединяет различные типы данных
- Аналогия с *struct* в языке C
- Могут быть группами элементов управления или индикаторов
- Схожи с многожильным проводом
- **Порядок элементов в кластере очень важен**



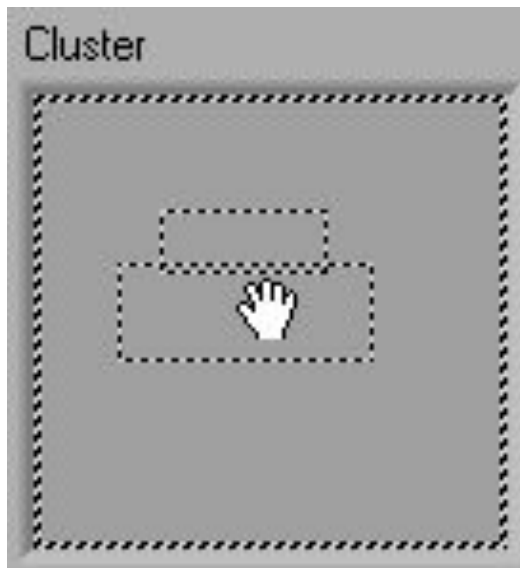
# Создание кластера

1. Выбрать контейнер кластера с палитры.

**Controls»Modern»Array, Matrix & Cluster**

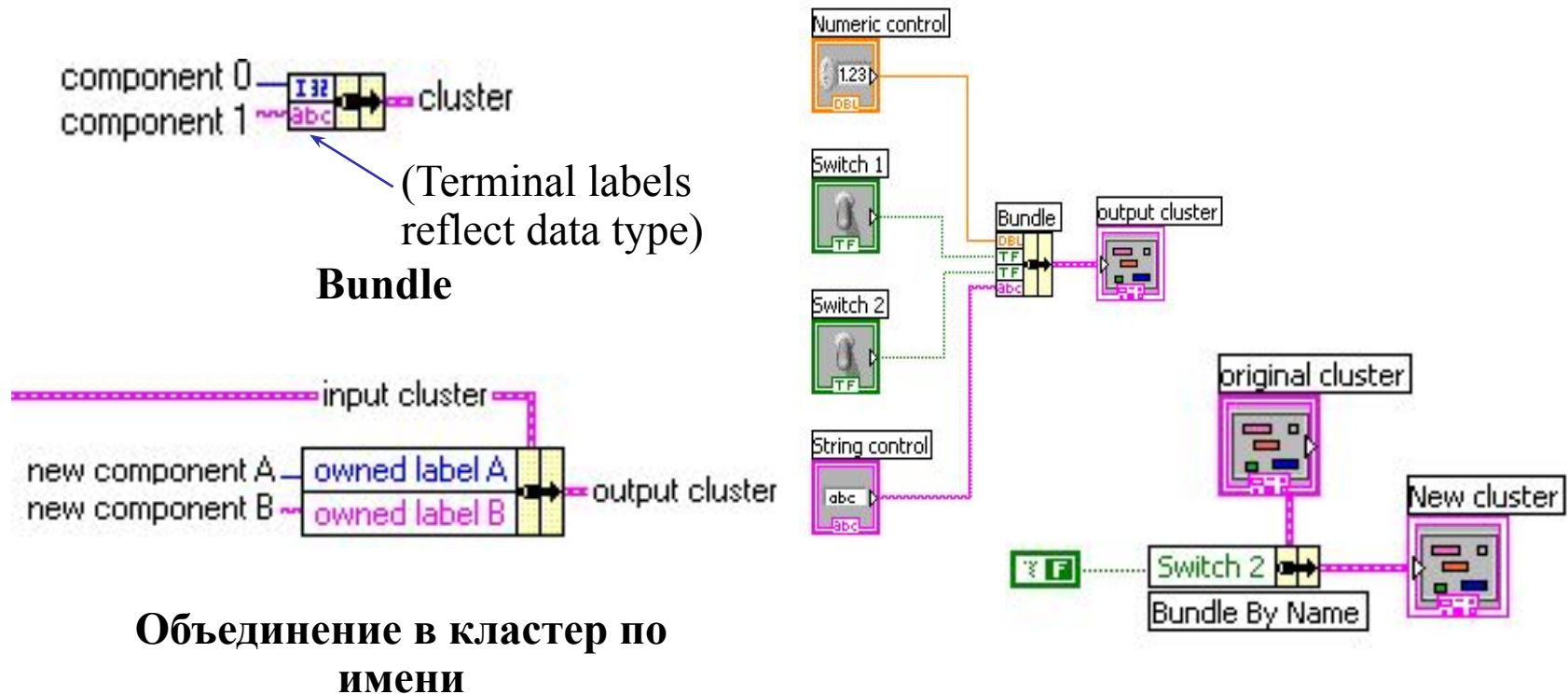


2. Вставить желаемый объект в контейнер.



# Функции кластеров

- Местоположение **Cluster & Variant**, подпалитра **Programming**
- Могут быть доступны при щелчке правой кнопки мыши по терминалу кластера



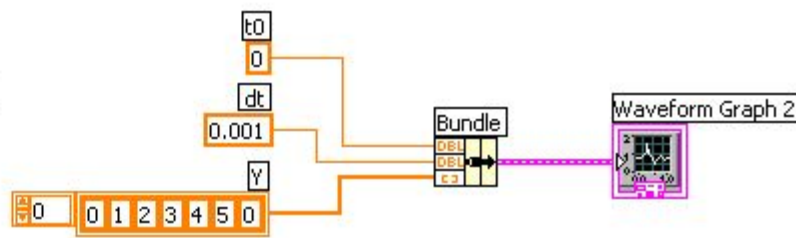
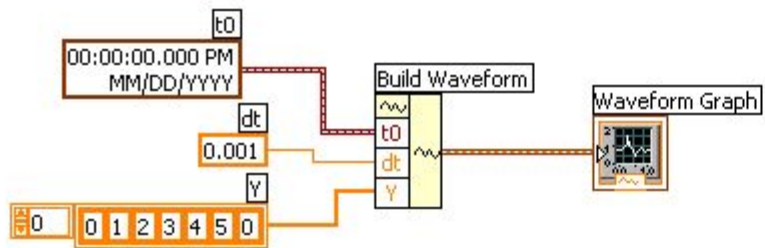
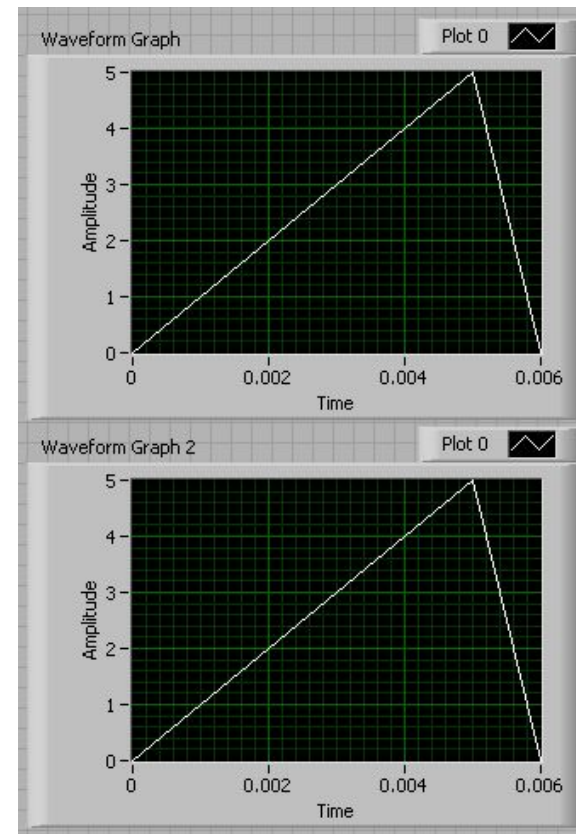


# Использование массивов и кластеров

Отображение на графике

- $t_0$  = начальное время
- $dt$  = время между отсчетами
- $Y$  = массив по оси  $Y$

Два способа создание кластера графиков

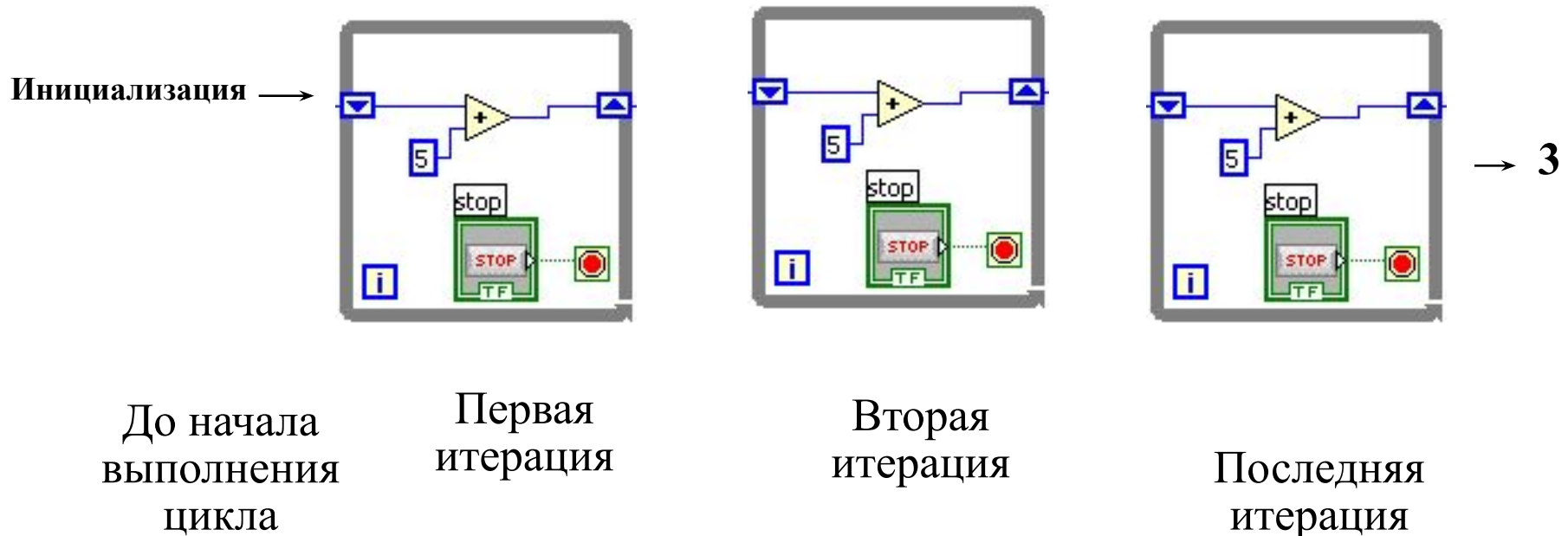


Создание графика (абсолютное время)

Кластер (относительное время)

# Сдвиговой регистр – доступ к предыдущим итерациям

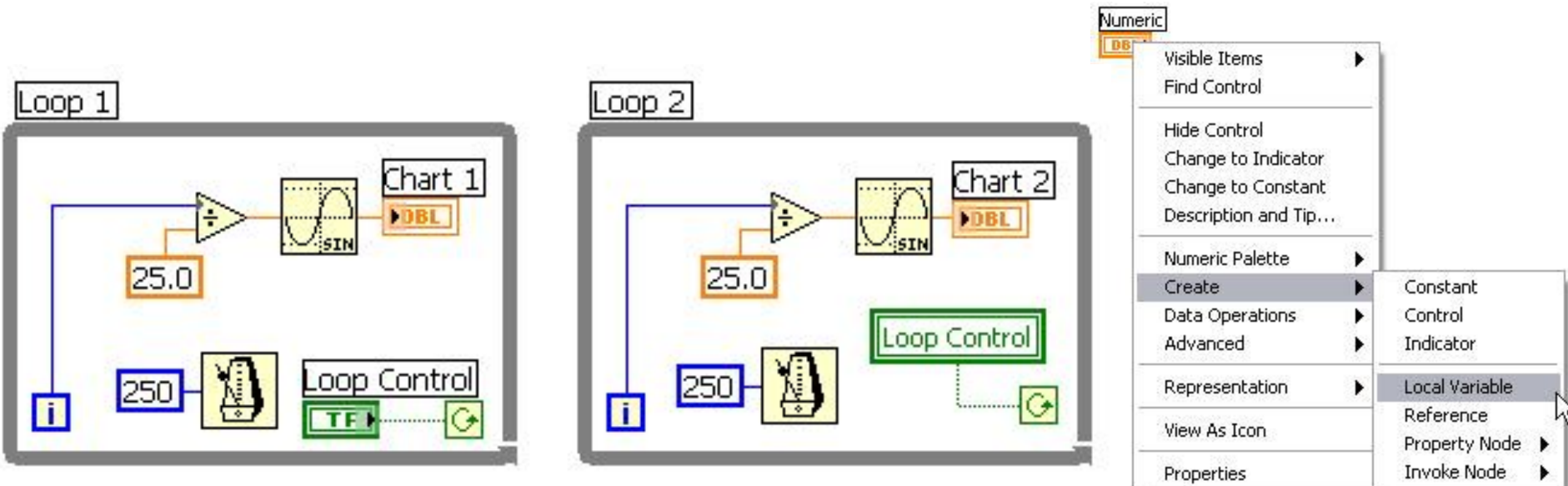
- Доступен на правой и левой границе циклов
- Правая кнопка мыши по границе и выбор из меню **Add Shift Register**
- Правый терминал сохраняет данные последней итерации
- Левый терминал сохраняет данные предыдущей итерации для начала следующей.



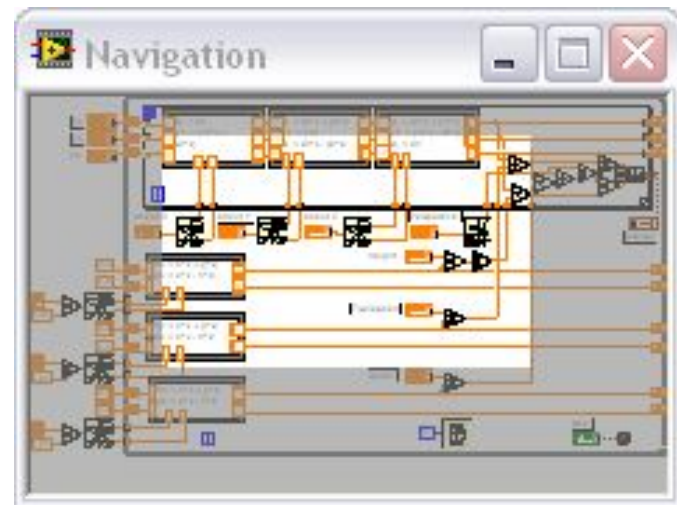
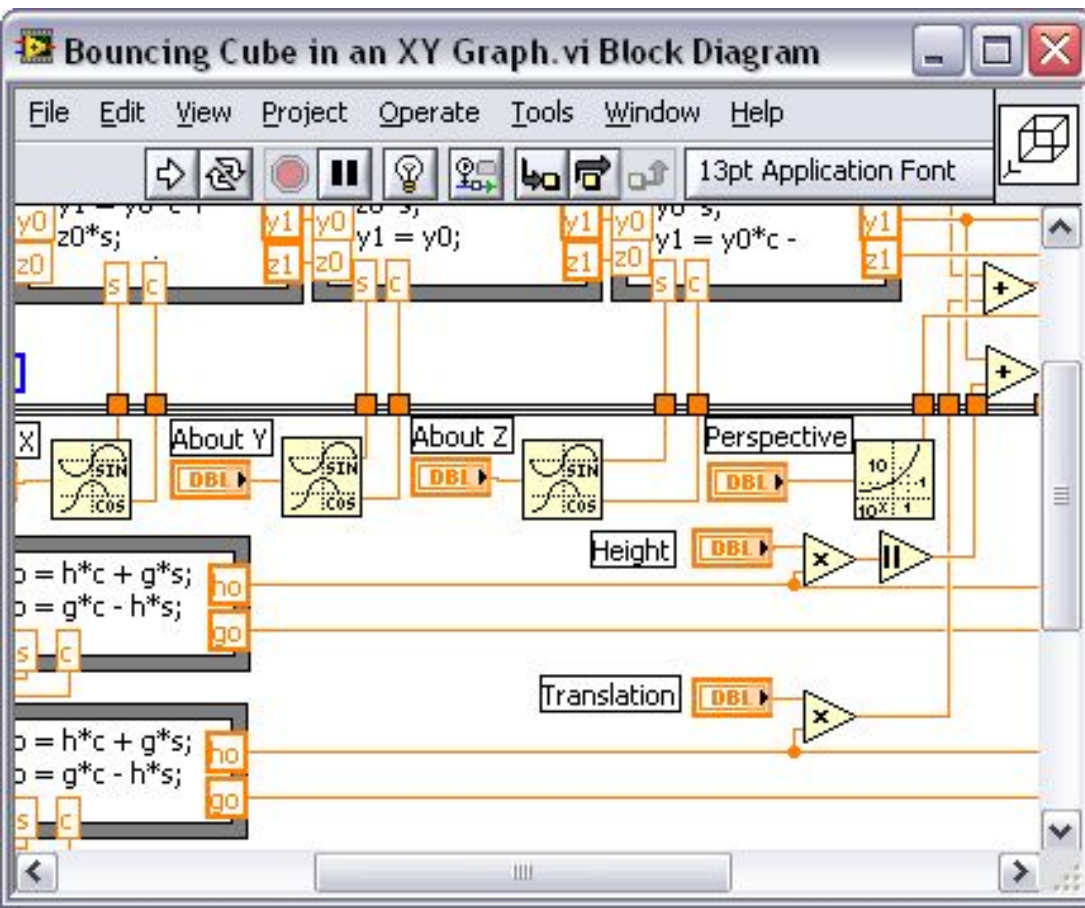


# Локальные переменные

- Локальные переменные обеспечивают передачу данных между параллельными циклами
- Значение элемента управления или индикаторы может быть считанно или записано в разных местах программы в нескольких местах



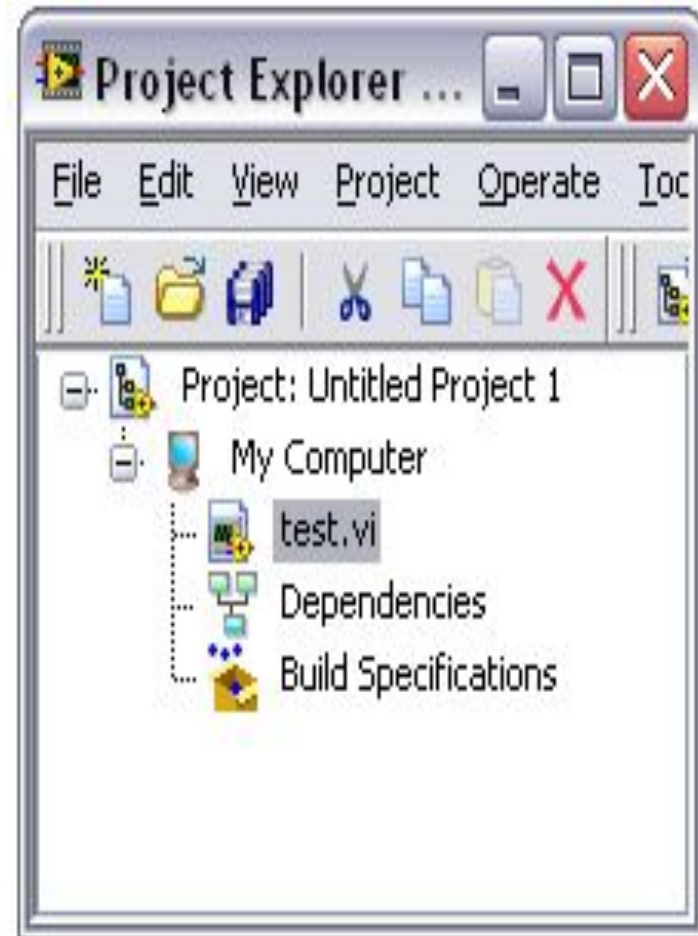
# Окно навигации в LabVIEW



- Отображает текущий участок программы
- Идеально для больших приложений

# Проекты в LabVIEW

- Группирует и организует ВП
- Управление аппаратными средствами
- Управление и загрузка на различные платформы
- Создание библиотек и исполняемых файлов
- Управление большими приложениями в LabVIEW а  
(**LabVIEW»Project»New**)



# Программа сертификации LabVIEW

## Architect

- Mastery of LabVIEW
- Expert in large application development
- Skilled in leading project teams

Certified  
LabVIEW  
Architect

## Developer

- Advanced LabVIEW knowledge and application development experience
- Project management skills

Certified LabVIEW  
Developer

## Associate Developer

- Proficiency in navigating LabVIEW environment
- Some application development experience

Certified LabVIEW Associate  
Developer

**Бесплатный экзамен LabVIEW**

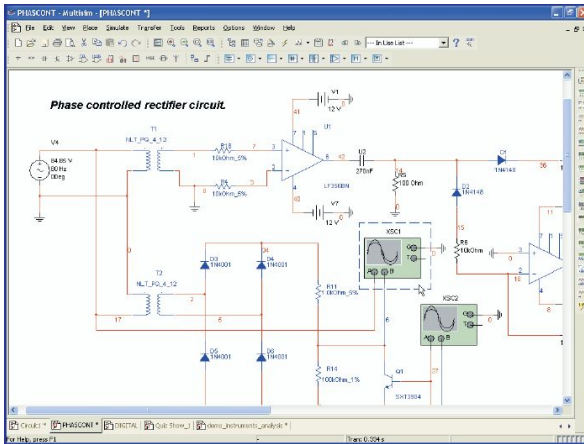
## Fundamentals Exam

- Pre-Certification Skills Test

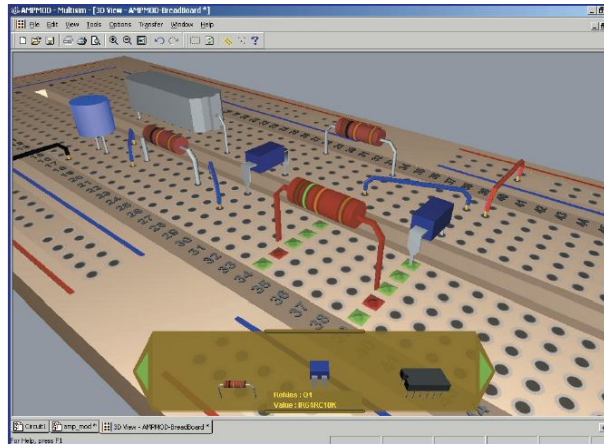


# Интеграция Multisim с LabVIEW

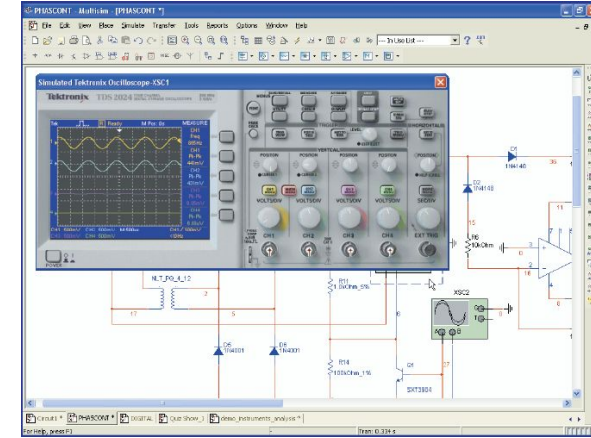
## 1. Create Schematic



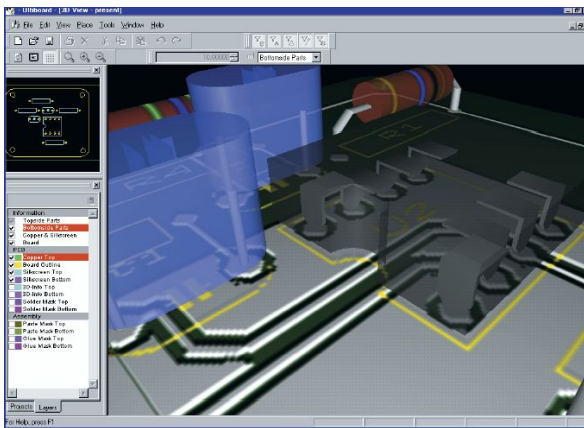
## 2. Virtual Breadboard



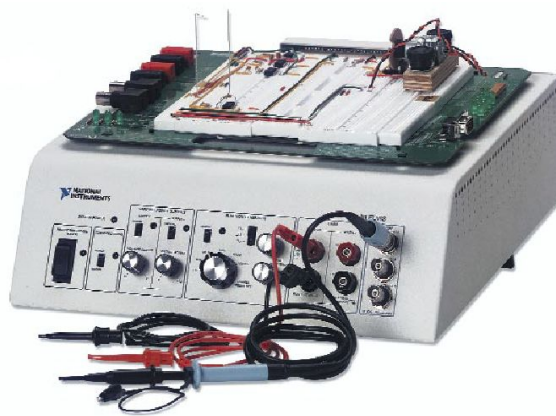
## 3. Simulate



## 4. PCB Layout



## 5. Test



## 6. Compare

