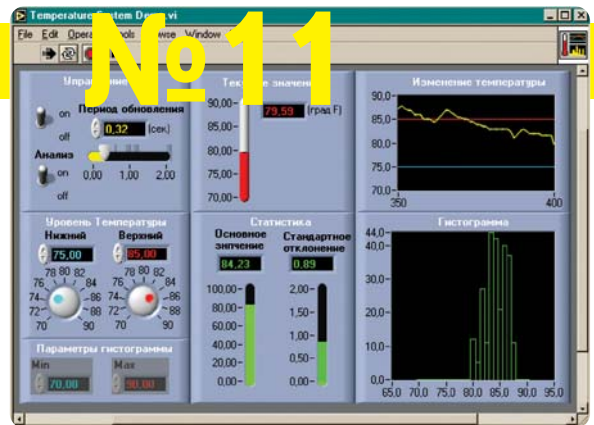


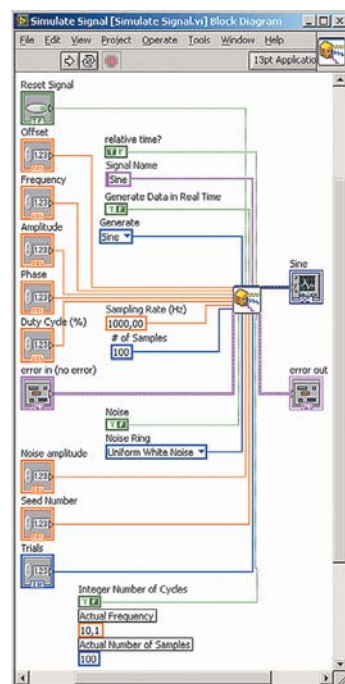
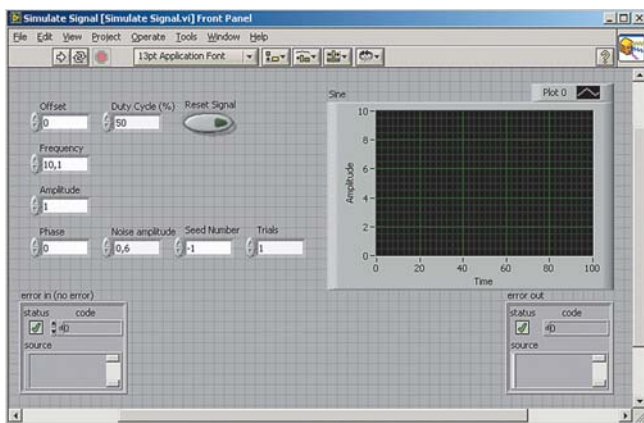
Уроки по LabVIEW

Используемые в различных программах однотипные фрагменты есть смысл оформить в виде подпрограмм. В LabVIEW это называется subVI. Наверное, у Вас уже накопилось немало полезных subVI, из которых можно сформировать целую библиотеку. Можете себе представить, а что же имеется в "загашнике" у асов LabVIEW. Очевидно, что, используя такие наработки, проекты можно "щелкать как орехи".

А почему бы не обобщить накопленный опыт. Ведь можно создать набор часто используемых инструментов, причем конфигурируемых под конкретную задачу. Это уже будет нечто другое, нежели просто библиотека subVI. Вот так и появилась в LabVIEW 7 технология ExpressVI, получившая дальнейшее развитие в новой версии - LabVIEW 8



Для решения общих измерительных задач используются Express VI, которые представляют собой узлы с минимальным числом соединений, поскольку пользователь может сконфигурировать их с помощью диалоговых окон. Каждый Express VI является самостоятельным блоком, состоящим из стандартных функций LabVIEW, и все что нужно



для его функционирования - ввести несколько нехитрых параметров и проставить нужные флажки в диалоговом окне.

Давайте рассмотрим внутреннюю структуру одного из них (на примере, **Simulate Signal**). Переходим в окно редактирования диаграмм и устанавливаем пиктограмму **Simulate Signal (Functions >> Express >> Input >> Simulate Signal)**. Как только Вы поместили пиктограмму на поле редактирования диаграмм, сразу же появляется диалоговое окно конфигурации VI. Что с ним делать, расскажем далее, а пока просто на-

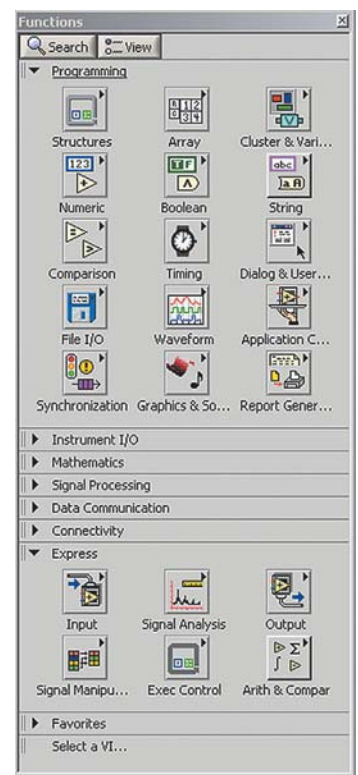
жимаем кнопку "OK".

Чтобы раскрыть внутреннюю структуру нашего Express VI, щелкаем на нем правой кнопкой мыши. Далее, из всплывающего меню выбираем **"Open Front Panel"**, получив предупреждение о том, что при раскрытии структуры Express VI теряется возможность его конфигурирования с помощью диалогового окна, выбираем **"Convert"**. При этом цвет пиктограммы изменится из голубоватого на светло-бежевый.

Открыв окно редактирования диаграмм, Вы убедитесь в том, что Express VI представляет собой подпрограмму, состоящую из стандартных функций LabVIEW. Естественно, можно изменить структуру VI, но этого делать не будем, потому, как возможности в этой области практически не ограничены.

Кстати, Вы обратили внимание на изменение внешнего вида панели **Functions**? В LabVIEW 8, кроме привычного для нас разделения всех VI на группы (**Array**, **String** и т. д.), произведено их разбиение на более обширные классы: **Programming** (наиболее употребляемые VI при создании программы), **Instrument I/O** (VI, необходимые для сопряжения внешних устройств с ПК), **Mathematics** (функции для проведения математических вычислений), **Signal Processing** (VI для симуляции и обработки сигналов), **Data Communication** (получение, обработка и передача данных), **Connectivity** (подключение к программе динамически-компоновываемых (DLL) библиотек, регистров и т. д.), **Express** (группа, включающая в себя Express VI), **Favorites** (любимое).

Теперь, после общего ознакомления с концепцией Express VI, давайте рассмотрим группы VI, входящие в состав класса **Express**, а также самых "ярких" представите-



лей каждой группы.

Первая группа - **Input** (ввод информации). В состав данной группы входят следующие Express VI: **Instrument I/O Assistant** - применяется для соединения ПК с внешними устройствами или приборами, использующими последовательный интерфейс, GPIB или Ethernet; **Simulate Signal** - имитирует сигнал заданной формы, амплитуды и частоты; можно также добавить к сигналу один из девяти предложенных видов шумов; **Simulate Arbitrary Signal** - создает сигнал произвольной формы по заданным пользователем точкам; **Acquire Sound** - осуществляет получение данных от звукового устройства; **Read From Measurement File** - считывает данные из специальных файлов (с расширениями .lvm и .tdm), в которых хранится информация о проведенных экспериментах, а также результаты этих экспериментов (запись в эти файлы реализуется с помощью Express VI под названием **Write To Measurement File**); **Prompt User for Input** - предлагает пользователю ввести какую-либо информацию, например имя или пароль; **File Dialog** - используется для выбора существующего файла (директории) или для создания нового файла по заданному адресу.

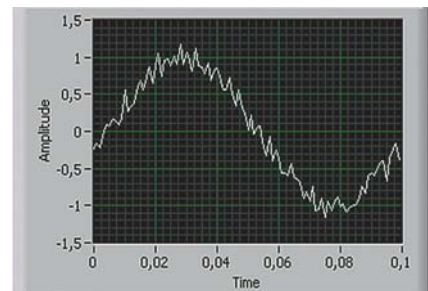
В качестве представителя этой группы рассмотрим **Simulate Signal**.

Данный Express VI имитирует различные виды сигналов (синусоидальный,



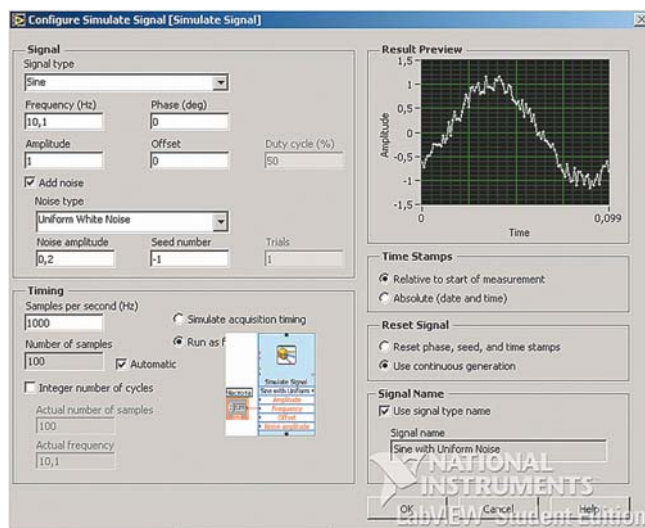
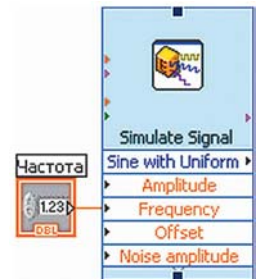
отсчетов в секунду "Samples per second" и общее количество отсчетов "Number of samples". Если же Вам нужно изменить ранее установленные параметры или значения, двойного нажатия левой кнопки мыши на иконке VI достаточно для повторного открытия диалогового окна.

Проделав этот несложный путь и, подключив к Express VI уже давно Вам знакомый **Waveform Graph**, увидим вполне ожидаемую картину: синусоиду с белым шумом. Вот и все - никаких трудностей. При этом код исполняемой программы будет иметь очень простой и компактный вид.

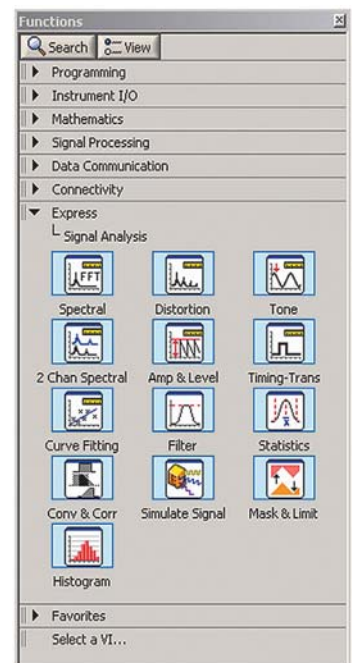


Следует также отметить, что для изменения, например, частоты сигнала нет необходимости каждый раз обращаться к блок-диаграмме и вызывать диалоговое окно конфигурации. Для этого нужно просто растянуть Express VI "вниз" и соединить регулятор (Control) со входом "frequency" нашего VI.

Вторая группа - **Signal Analysis** (обработка сигнала). В состав группы входят: **Spectral Measurements** - выполняет вычисление амплитудного спектра или спектра мощности сигнала; **Distortion Measurements** - производит вычисления искажений сигнала, такие как анализ гармонического колебания, коэффициента нелинейных искажений и отношение сигнала к шуму и искажениям; **Tone Measurements** - находит единственное гармоническое колебание с наибольшей амплитудой или производит поиск такого колебания в заданном диапазоне частот. Для найденного колебания могут быть определены частота и фаза. **Dual Channel Spectral Measurements** - выполняет спектральный анализ двух сигналов. **Amplitude and Level Measurements** - производит вычисление постоянной и переменной составляющих сигнала, максимальное и мини-

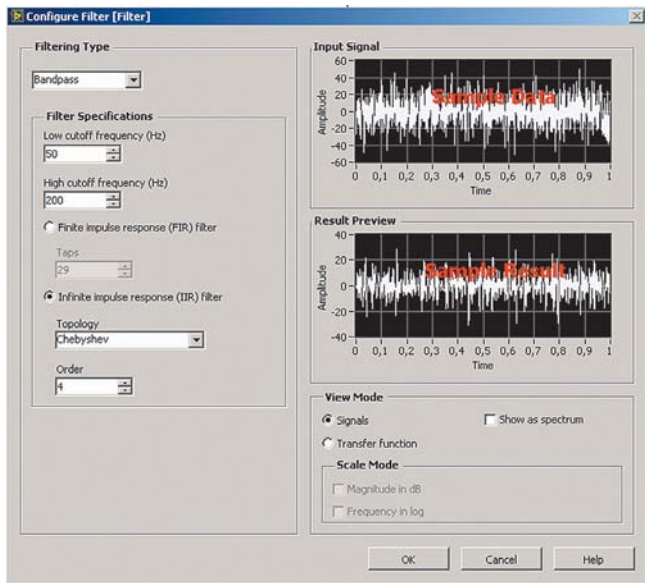
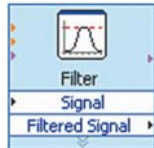


прямоугольный, треугольный, пилообразный) и шумовой сигнал. Как только Вы поместите иконку **Simulate Signal** на блок-диаграмме - сразу же "выскочит" диалоговое окно, с помощью которого можно будет задать все необходимые параметры сигнала и шума. В строке "Signal type" укажите форму сигнала, ниже его частоту, фазу, амплитуду и смещение по амплитуде. Поставив флажок "Add noise" можно к выбранному сигналу добавить шум, задав его тип и амплитуду. Также можно изменять количество



мальное пиковые значения, от пика до пика, среднеквадратическое значение, среднее и среднеквадратическое значения цикла. **Timing and Transition Measurements** - выполняет вычисления временных (частота, период, длительность импульса) и переходных (выброс перед фронтом импульса, выброс после фронта импульса, скорость нарастания) параметров импульсных сигналов. **Curve Fitting** - реализует сглаживание сигнала согласно выбранной модели (линейная аппроксимация, нелинейная аппроксимация, сплайн-интерполяция, полиномиальная интерполяция). **Filter** - обрабатывает сигналы, используя функции фильтров или весовых окон, при этом используются цифровые КИХ-фильтры (с конечной импульсной характеристикой) и БИХ-фильтры (с бесконечной импульсной характеристикой). **Statistics** - при помощи данного Express VI можно определить статистические параметры сигнала, начиная с минимального и максимального значений сигнала, и заканчивая такими моментами, как эксцесс и асимметрия. **Convolution and Correlation** - производит свертку (Convolution) или деконволюцию (Deconvolution) сигналов, вычисляет авто- или взаимно-корреляционные функции. Далее видим уже знакомый **Simulate Signal**, а это говорит о том, что к некоторым VI можно добраться сразу несколькими путями. **Mask and Limit Testing** - проверяет, входит ли амплитуда сигнала в заданные Вами верхнюю и нижнюю границы, при этом границы могут описываться как константами, так и различными функциональными зависимостями. И, наконец, последний Express VI этой группы - **Create Histogram**. С его помощью можно построить гистограмму входного сигнала согласно заданному количеству интервалов и граничным значениям.

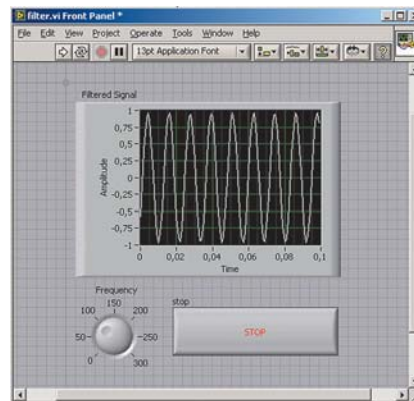
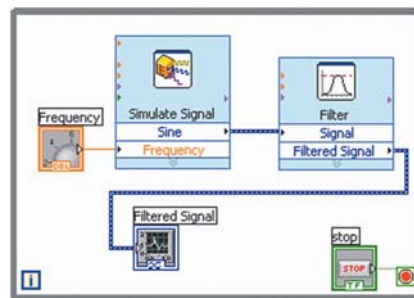
Более подробно рассмотрим Express VI, реализующий фильтрацию сигнала - **Filter**. В диалоговом окне задаем: полосовой фильтр (Bandpass), нижняя и верхняя частоты среза - соответственно 50 и 200 Гц, БИХ-фильтр (Infinite impulse response filter) Чебышева четвертого порядка.



Теперь подключим к фильтру источник синусоидального сигнала с возможностью изменения частоты от 0 до 300 Гц. Выводим результат фильтрации с помощью **Waveform Graph**. Изменяя частоту колебания, Вы увидите, как подавляются сигналы в диапазонах до 50 Гц и свы-

ше 200 Гц. Очевидно, что работать с этим Express VI предельно просто.

Третья группа - **Output** (вывод). В состав группы входят следующие Express VI: уже рассмотренный **Instrument I/O Assistant**, **Build Text** - в наиболее простом случае соединяет две текстовые величины: текст на входе + текст, заданный в диалоговом окне (если же на вход подаются не текстовые символы - Express



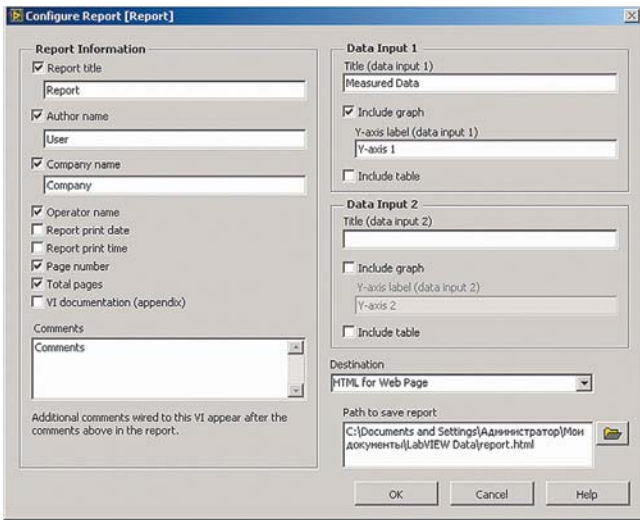
VI преобразовывает их в строку, согласно указанным Вами параметрам преобразования); **Display Message To User** - при выполнении определенных Вами условий, пользователю выводится соответствующее сообщение; **Play Waveform** - выводит сигнал на звуковоспроизводящее устройство; **Write To Measurement File** - записывает данные результатов измерения в файл с расширениями .lvm или .tdm, которые потом можно прочитать при помощи **Read From Measurement File**; **Report** - подает на печать или создает html-файл, включающий в себя, возвращенные другими VI данные, а также имя автора отчета, количество страниц и дату создания; **DIAdem Report** - дает возможность не только создавать отчеты в виде .html или .pdf, но и интерактивно анализировать и отображать данные (для правильного функционирования этого Express VI нужно установить DIAdem 9.1 или более позднюю версию на Ваш ПК).

Давайте создадим отчет при помощи **Report**. В диалоговом окне указываем название отчета, автора и компанию. Также, проставив определенные флажки в окне, включаем в доклад дополнительную информацию и комментарии.

Далее указываем путь и название файла-отчета в правом нижнем окошке.

Подключаем к нашему VI все тот же **Simulate Signal** и за-



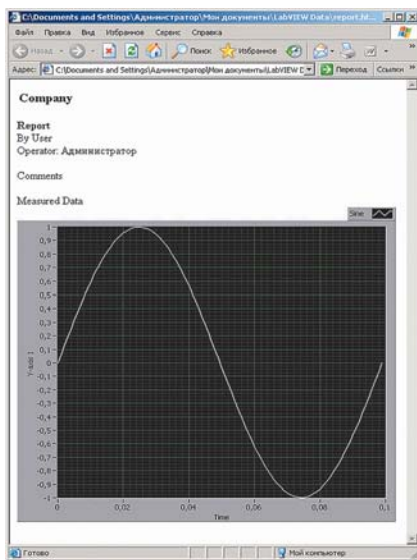


пускаем программу. Все что Вам теперь нужно сделать - открыть html-файл и посмотреть на результаты Вашей работы.

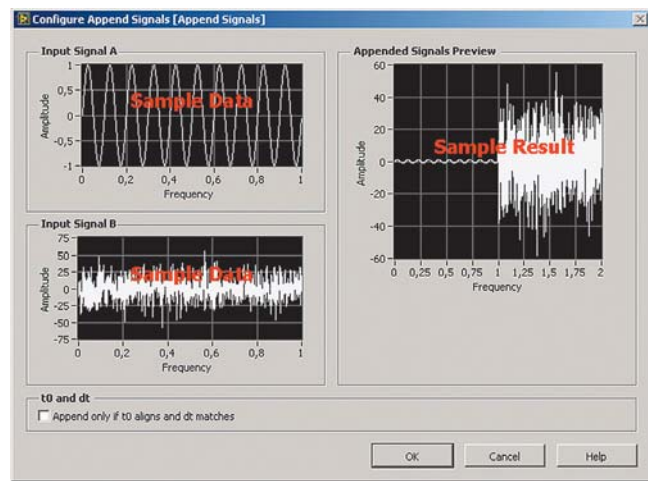
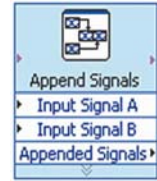
Выбрав в окне "Destination" Print to Printer, можно сразу же и напечатать данный отчет.

Четвертая группа - **Signal Manipulation**.

Перед тем, как переходить к рассмотрению VI из этой группы, обратите внимание на возможно уже знакомые Вам VI: **Merge Signals** и **Split Signals**. Первый VI реализует "слияние" нескольких сигналов в один поток. Второй - наоборот, разделяет его на исходные сигналы. Теперь перейдем непосредственно к рассмотрению Express VI. **Select Signals** - на вход принимает поток сигналов, из которых с помощью диалогового окна Вы можете выбрать только полезные для Вас; **Align and Resample** - позволяет выровнять времена запуска сигналов, а также производить их повторную дискретизацию при заданном Вами периоде дискретизации; **Collector** - выводит на **Waveform Graph** заданное Вами количество точек сигнала (или несколь-



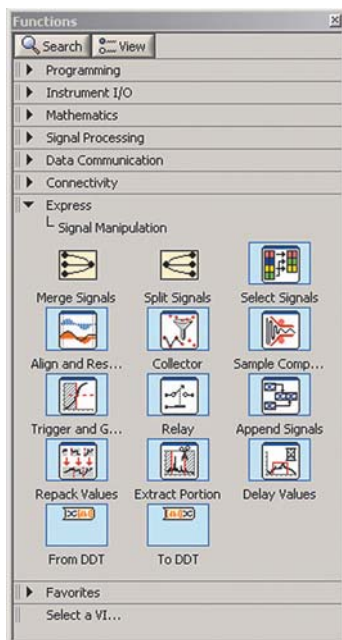
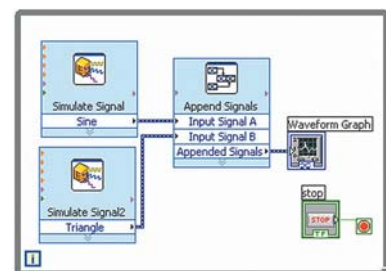
ких сигналов), при этом выводится N последних точек; **Sample Compression** - уменьшает количество выборок входного сигнала в указанное Вами целое число раз; **Trigger and Gate** - позволяет выделить часть сигнала согласно указанным Вами условиям; **Relay** - выполняет роль ключа: в замкнутом состоянии сигнал на выходе ключа равен входному, в разомкнутом возможны два варианта работы - на выходе либо нулевой сигнал, либо данные отсутствуют; **Append Signals** - позволяет совместить конец одного сигнала с началом другого (более подробно работа данного Express VI будет рассмотрена ниже); **Repack Values** - разбивает входные сигналы на "пакеты" заданного Вами размера; **Extract Portion of Signal** - вырезает полезную часть сигнала, все что для этого нужно - указать начальную и конечную точки интересующего Вас интервала; **Delay Values** - содержит данные, полученные в результате циклического выполнения определенных итераций, выдает данные после того, как операции в цикле были выполнены некоторое число раз (по умолчанию 1);



Convert from Dynamic Data - преобразует данные динамического типа (динамически изменяющиеся) в данные другого типа: числового, булевого, в массив и т. д.; **Convert to Dynamic Data** - выполняет обратные предыдущему Express VI преобразования.

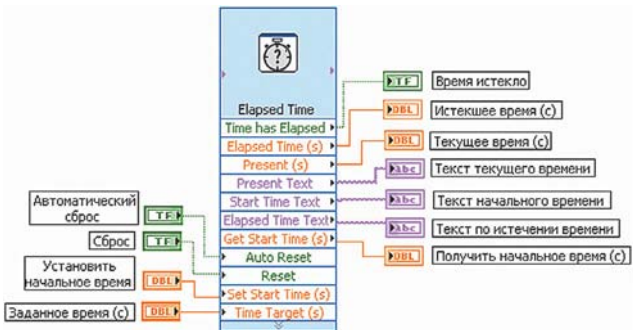
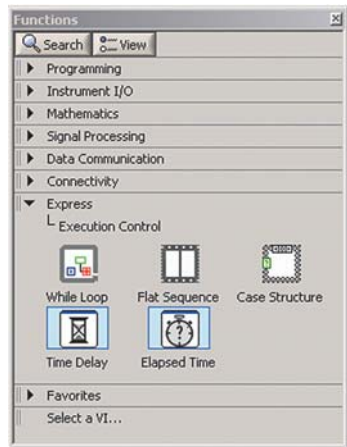
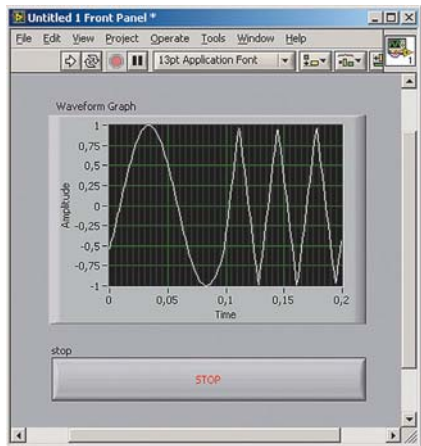
Как было обещано выше, рассмотрим **Append Signals** более детально. Итак, помещаем на поле диаграмм Express VI. В диалоговом окне видим пример совмещения сигналов. Обратите внимание на строку внизу окна. Наличие здесь "галочки" означает, что второй сигнал добавится к первому только после того, как этот первый закончится. Это означает, что в случае использования Вами **Simulate Signal** (а он может генерировать сигнал бесконечно долго), второй сигнал может и не начаться. Но LabVIEW этого не допустит. Вы сразу увидите сообщение об ошибке. Чтобы избежать всех вышеописанных неприятностей, следует просто убрать этот флажок.

Подключаем к **Append Signals** два источника сигналов и посмотрим, что получится. Для лучшей наглядности сделай-



те частоту второго сигнала в три раза большей, чем первого.

Пятая группа - **Execution Control** (контроль над выполнением). Сюда входят не только Express VI, но и некоторые другие структуры, рассмотренные в более ранних уроках: **While Loop**, **Flat Sequence Structure** и **Case Structure**. Как Вы уже знаете, именно с помощью этих структур осуществляется управление ходом выполнения и ветвление программы, именно поэтому они включены в данную группу. Рассмотрим теперь Express VI. Их в этой группе всего два: **Time Delay** (вносит временную задержку в выполнение программы) и **Elapsed Time** (сохраняет



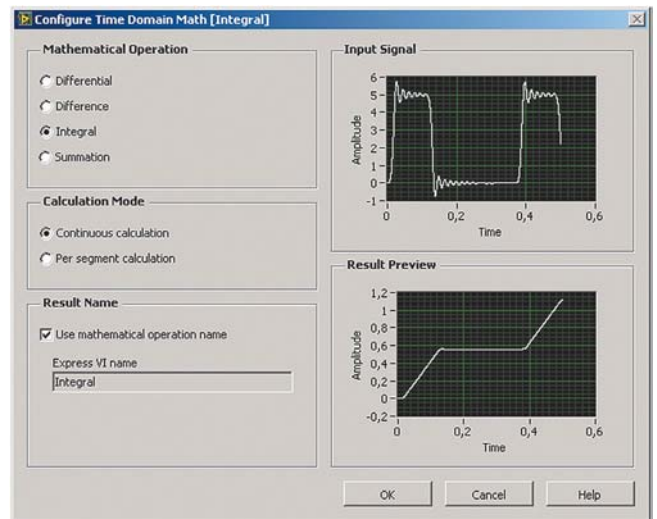
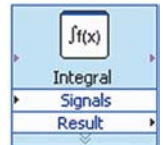
отсчеты времени для индикации момента истечения заданного интервала времени). Использовать эти функции предельно просто, но все же раскроем **Elapsed Time** более подробно.

Вам следует только указать отсчитываемое время и определить, какой сброс будет использоваться: автоматический или ручной.

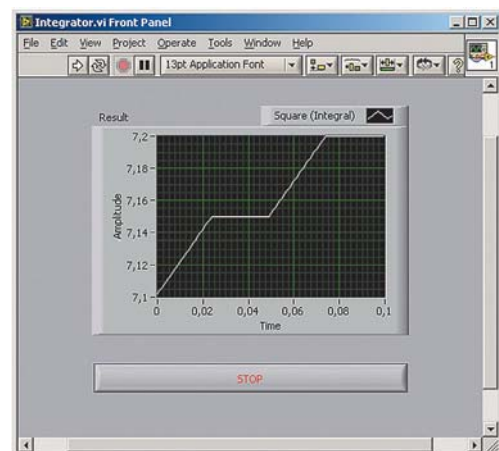
И последняя, шестая группа Express VI - **Arithmetic & Comparison** (вычисление и срав-



нение). В эту группу входят стандартные вычислительные VI, а также три Express VI. **Formula** - позволяет производить математическую обработку входных данных, обеспечиваемую базовыми научными калькуляторами; входы и выходы Express VI рассчитаны на подключение данных динамического типа; при подключении данных другого типа производится преобразование типов, в том числе и с помощью Express VI **Convert to Dynamic Data. Scaling & Mapping** - изменяет диапазон и характер отображения входных данных с помощью их масштабирования и выбора вида преобразования (линейное, логарифмическое, интерполяционное). **Time Domain Math** - выполняет одну из операций математической обработки сигналов во временной области. Рассмотрим этот VI подробнее. В диалоговом окне выбираем, например, вычисление функции интеграла для входного сигнала. При помощи **Simulate Signal** подайте на VI прямоугольный сигнал, не

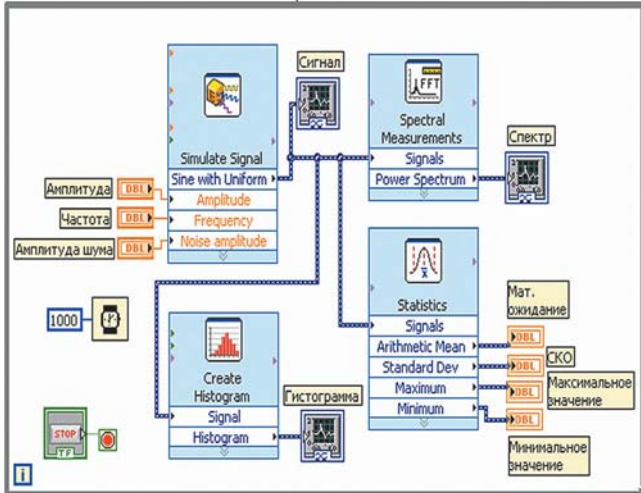


имеющий отрицательных значений. Подключите к выходу "Result" графический индикатор **Waveform Graph**, и запустите программу. Выполнив все вышесказанное, на **Waveform Graph** Вы увидите следующую картинку.

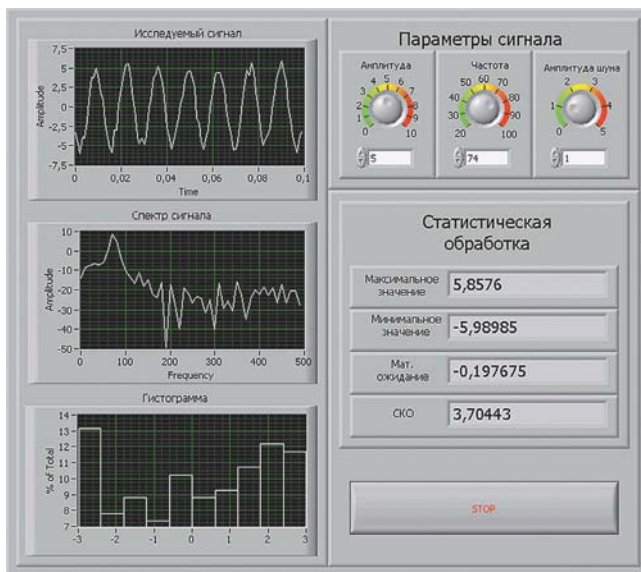


Логическим завершением этого урока будет простенькая программа, созданная при помощи нескольких Express VI.

Эта программа выполняет и статистический, и спектральный анализ сигнала одновременно. Ее описание в рамках одного урока было бы затруднительным, если бы использовались стандартные VI более старых версий LabVIEW. Но в данном случае, благодаря использованию Express VI, получилась очень компактная и легкая для понимания программа.



Как видите, кроме нескольких Express VI (*Simulate Signal, Spectral Measurement, Create Histogram, Statistics*), простейших регуляторов и индикаторов ничего использовать не пришлось.



Вам понравилось знакомство с новшествами под названием Express VI? Ответ может быть только положительный. Вы сможете писать свои гораздо более сложные по функциональности программы, но теперь уж точно их код не будет пугать ни Вас, ни Ваших сотрудников, ни клиентов.

Успехов Вам в LabVIEW 8!

Материал урока подготовлен сотрудниками "ХОЛИТ Дэйта Системс", г.Киев.

СПРАШИВАЙТЕ - ОТВЕЧАЕМ

- Чем отличается базовая (Base), полная (Full) и профессиональная (Pro) системы разработки LabVIEW? Ценой это понятно, а по существу?
- В двух словах, популярно, - "глубиной" математики
- А нельзя ли подробнее?
- Можно, проанализируйте таблицу сравнения из каталога "Measurement and Automation 2006", стр. 31

Development Tools	Pro	Full	Base
3D controls and indicators, graphs	•	•	•
Programming structures and fundamentals	•	•	•
Instrument control, data and image acquisition, motion	•	•	•
Express VIs, DAQ Assistant, ¹ Instrument I/O Assistant ¹	•	•	•
Property pages for front panel objects	•	•	•
Automatic wiring	•	•	•
Localization tools	•	•	•
Automatic instrument driver recognition and retrieval	•	•	•
Text report creation	•	•	-
HTML document publishing	•	•	-
Debugging Tools	Pro	Full	Base
Slow and pause execution using execution highlighting	•	•	•
Set breakpoints, probe values during execution	•	•	•
Step-into, step-out, step-over	•	•	•
Custom probes, conditional breakpoints	•	•	•
Automatic error handling	•	•	•
Connectivity Tools	Pro	Full	Base
Easy client/server broadcasting through shared variable	•	•	•
Built-in Web server, Web publishing tools	•	•	•
NET ¹ , XML, ActiveX, ¹ TCP/IP, UDP support	•	•	•
Infrared (IrDA) communication	•	•	•
E-mailing data from VIs	•	•	-
Remote control of applications using Web interface	•	•	-
Analysis Tools	Pro	Full	Base
Point-by-point analysis	•	•	-
Linear algebra and array operations	•	•	-
Probability and statistics	•	•	-
Curve fitting	•	•	-
Fourier, Hilbert, and other transforms	•	•	-
Amplitude, phase, power, and cross-power spectra	•	•	-
Signal and noise generation	•	•	-
Frequency and impulse response	•	•	-
Peak detection	•	•	-
DC/RMS calculation	•	•	-
Harmonic distortion analysis	•	•	-
SINAD analysis	•	•	-
Single tone analysis	•	•	-
Limit mask testing	•	•	-
IIR/FIR filters	•	•	-
Butterworth, Chebyshev, and other nonlinear filters	•	•	-
Windowing	•	•	-
Interpolation algorithms	•	•	-
Ordinary differential equations	•	•	-
Integration and differentiation	•	•	-
Gamma, Bessel, Jacobian	•	•	-
Root solving	•	•	-
The MathWorks, Inc. MATLAB [®] scripting	•	•	-
Pulse and transition measurements	•	•	-
Interactive math/algorithm development environment	•	•	-
Advanced User Interface Tools	Pro	Full	Base
Splitter panes, tree controls, subpanels	•	•	-
Fully customizable Xcontrols	•	-	-
Event-driven user interface programming	•	•	-
Custom graphics and animation, 3D graphs	•	•	-
VIs to change cursor appearance	•	•	-
Project Management Tools	Pro	Full	Base
Project Explorer window	•	•	•
Project library	•	•	•
Complexity metrics	•	-	-
Source code control integration with VSS and Perforce	•	-	-
Polymorphic VI creation	•	-	-
Software engineering with LabVIEW documentation	•	-	-
Quality and programming standards documentation	•	-	-
Deployment Tools	Pro	Full	Base
Application Builder (EXE and DLL creation)	•	-	-
MSI installer generation ¹	•	-	-

СПРАШИВАЙТЕ - ОТВЕЧАЕМ

MATLAB[®] is a registered trademark of The MathWorks, Inc.
¹ Available for Windows platforms only.