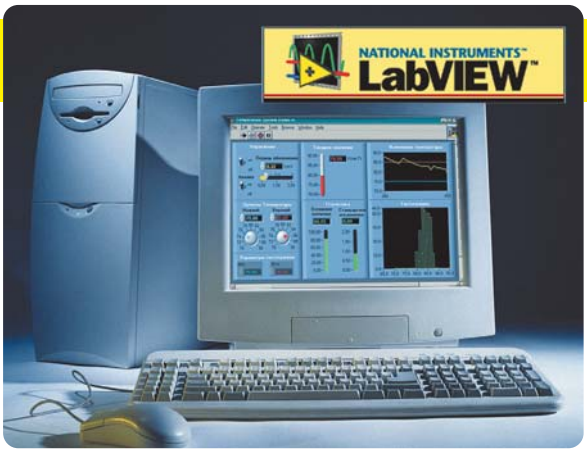
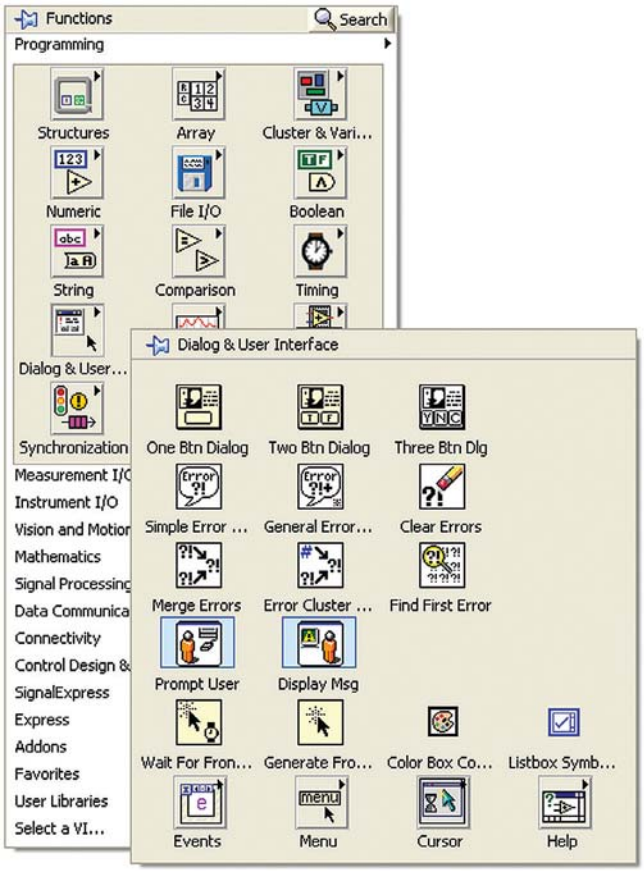
Учебный практикум по LabVIEW

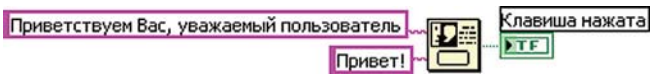


Одним из параметров оценки качества любого программного обеспечения является удобство интерфейса пользователя – достаточная наглядность. Интерфейс качественной программы должен быть понятен интуитивно, по крайней мере, насколько это возможно. Даже в меру "просвещенный" пользователь не обязательно должен "лезть" в "Help". Надписи и картинки в окне открытой программы должны наталкивать его на правильные действия. Вот об этом и пойдет речь в текущем практикуме.

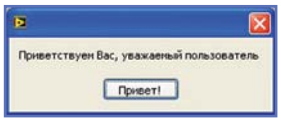
Очень важным элементом контроля над работой любой программы являются диалоговые окна. Они могут содержать подсказки к вопросам о том, какое действие необходимо выполнить (с возможностью последующего выбора) тут же в окне. С рассмотрения функций работы с диалоговыми окнами давайте и начнем. Функции эти находятся в подменю ***Dialog & User Interface***:



Очевидно, что простейшее диалоговое окно представляет собой прямоугольник с текстовым сообщением и кнопкой подтверждения. Функция ***One Button Dialog*** (Диалоговое окно с одной клавишей) как раз и реализует данное "общение":



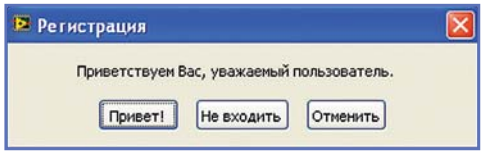
Входы слева - выводимое сообщение и надпись на кнопке подтверждения. Выводимую на кнопку надпись (в данном примере - "Привет!") можно и не задавать, тогда пользователь будет наблюдать традиционное "OK". А вот к выводимому сообщению данные действия не применимы - над кнопкой всегда должно быть написано что-то определенное.



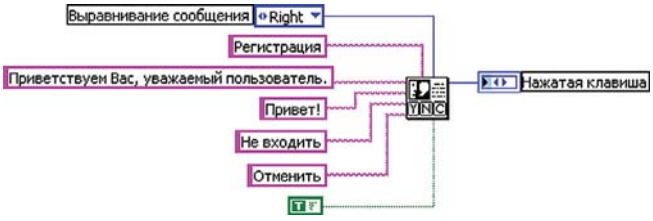
Следует заметить, что единственный выход данной функции - значение логического типа, которое показывает, была нажата клавиша или нет.

Функции ***Two Button Dialog*** (Диалоговое окно с двумя клавишами) была описана в одном из уроков по LabVIEW (ПИКАД №1-2005). Она является более "продвинутой" по сравнению с функцией однокнопочного окна, так как позволяет пользователю сделать выбор. Выход функции (данные, какая из кнопок была нажата, тип, естественно, булевский) можно использовать при последующей работе программы.

Диалоговое окно с тремя клавишами ***Three Button Dialog*** предназначено для расширения возможностей пользовательского выбора. Результат работы этой функции может быть, например, таким:

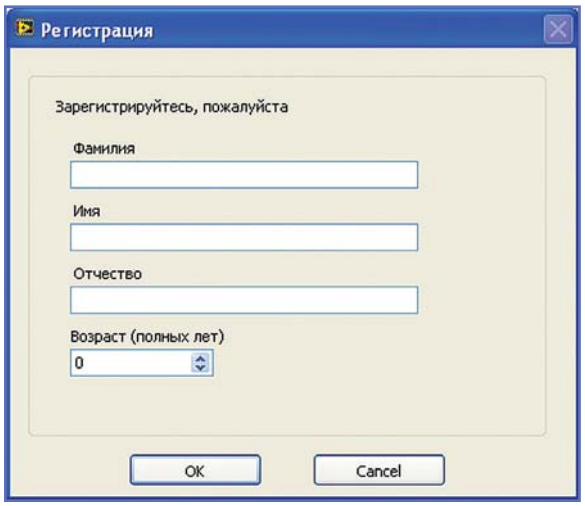


При этом на блок-диаграмме Вы можете задать все выводимые тексты и надписи, а также некоторые другие данные для настройки:

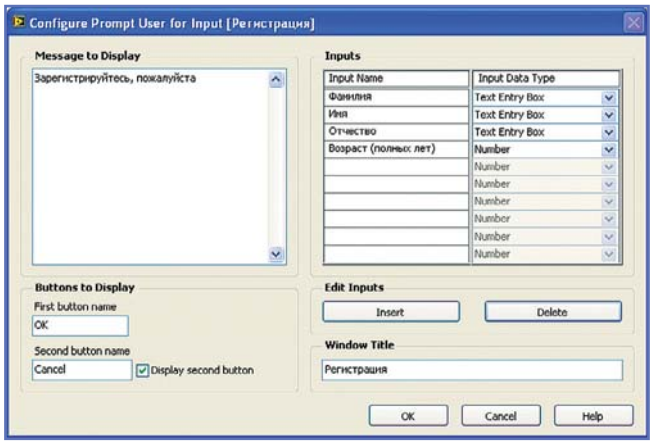


Вход "Выравнивание сообщения" используется, если выводимый текст не вмещается в одну строку окна либо состоит из нескольких абзацев, разделенных нажатием Enter. Выравнивание данного текста выполняется аналогично соответствующей процедуре в Microsoft Word.

Весьма полезным средством пользовательского интерфейса является экспресс-VI Prompt ***User for Input*** (Приглашение пользователя для входа). С его помощью можно реализовать процедуру заполнения анкет, создав перечень необходимых полей:

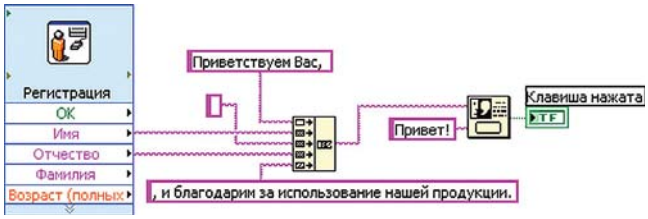


Панель конфигурирования данного экспресс-VI выглядит следующим образом (настройки соответствуют изображенному выше бланку):



Как видим, сообщение необходимо ввести в поле ***Message to Display*** - очень напоминает ранее описанные функции. Названия и количество (максимум - две) клавиш выбираются в поле ***Button to Display***. Одним из главных полей настройки является ***Inputs*** - специальный "отдел" для выбора количества заполняемых полей анкеты, их названий и типа данных. Данные могут быть как строкового типа, так и численного, и логического (в последнем случае пользователь видит соответствующий "флажок"). Заголовок окна можно задать в поле ***Window Title*** (оно расположено в правой нижней части).

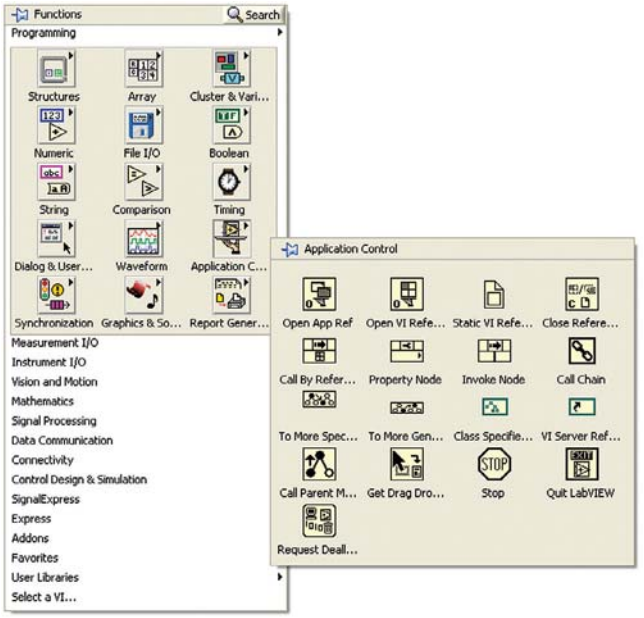
Пример совместного использования ***User for Input*** и ***One Button Dialog*** представлен ниже.



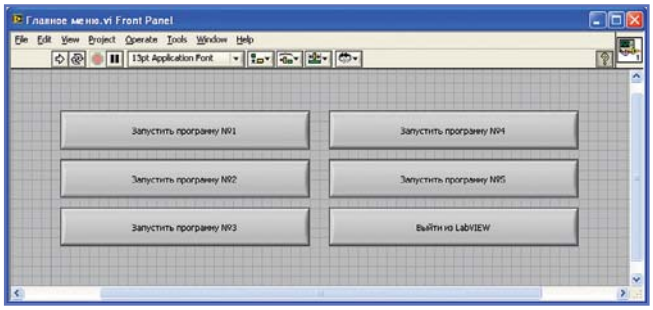
Можно в любой момент добавлять и удалять поля анкеты. При этом вид экспресс-VI на блок-диаграмме соответствующим образом меняется. Например, можно создать в экспресс-VI выход "Пароль" и впоследствии присоединить к одному из входов компаратора на блок-диаграмме, создав тем самым проверку введенного пароля. Если пароль неверный, то дальнейшая часть приложения не будет запущена. Можно таким же образом задействовать в сравнении и имя пользователя (логин). Этот вариант позволит реализовать список паролей (для каждого пользователя - свой пароль). Всевозможные компараторы уже были рассмотрены (ПИКАД №2-2008), а посему надо считать, что Ваши знания позволят выбрать подходящие для этих целей функции сравнения.

Наконец, рассмотрим организацию интерфейса пользователя при работе с несколькими VI, написанными и сохраненными в различных VI-файлах. Для удобства предположим (и гарантируем), что все файлы расположены в одной папке в памяти ПК. Так вот, чтобы не описывать в руководстве пользователя, какие из них и в какой очередности нужно запускать, в LabVIEW есть возможность создания отдельного VI-меню для их запуска.

Рассмотрим для реализации этой задачи некоторые функции из подменю ***Application Control*** (Управление приложением):

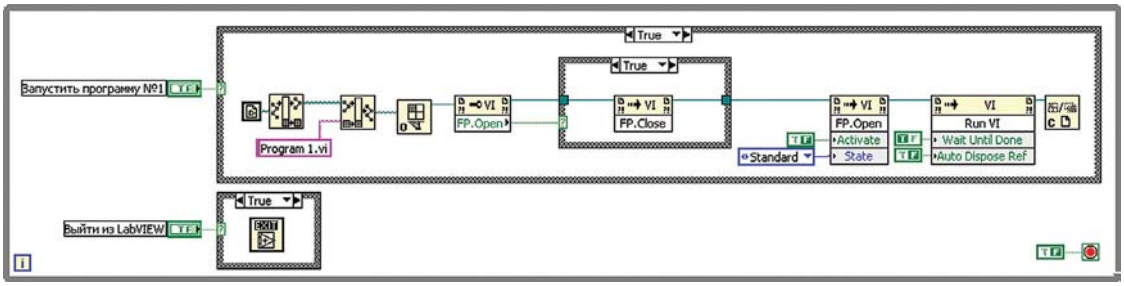


Для начала создадим в отдельном VI набор клавиш для вызова других, так называемых "этапных":

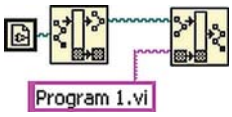


Разумеется, ничто не мешает привести в этом же окне инструкции выполнения, например, какую программу по очереди необходимо запускать. Ниже приведен фраг-

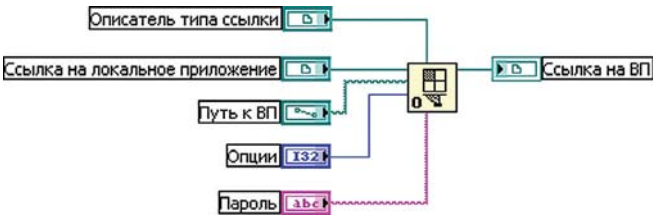
мент блок-диаграммы, с помощью которого реализуется вызов одного из "этапных" VI, и выход из этого и остальных VI:



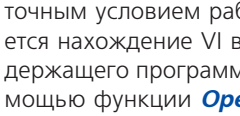
Путь к вызываемому VI, который содержит программу №1, получаем с использование функций ***Current VI's Path*** (Путь к текущему VI), ***Strip Path*** (Разобрать путь) и ***Build Path*** (Построить путь) подпалитры функций ***File I/O***. На выходе Current VI's Path - адрес нахождения VI "Главное меню" на диске, включая и его имя (непосредственно "Главное меню.vi"). Т.е. не что иное, как абсолютный путь. Функция ***Strip Path*** отсекает его последний элемент, т.е. убирает из пути название VI. А функция ***Build Path*** строит новый путь - к программе №1, который отличается от исходного лишь именем файла.



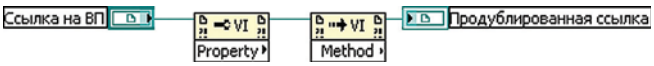
Преимущество данного подхода к определению пути заключается в адаптивности программы к перенесению набора VI в другую папку. Таким образом, Вы не используете абсолютные пути к файлам, а необходимым и достаточным условием работоспособности всей "связки" является нахождение VI в одной папке. Вызов другого VI, содержащего программу, из основного VI реализуется с помощью функции ***Open VI Reference*** (Открыть ссылку на VI) и узлов вызова и свойств. Первая загружает определенный VI в память, и возвращает - на ссылку на VI некоторые данные, используя которые можно управлять режимами работы и настройками данного VI.



Вход ***Описатель типа ссылки*** предназначен для определения типа данных, которые будут выданы функцией. Вход ***Ссылка на локальное приложение*** позволяет определить, локальная или удаленная среда LabVIEW используется. По умолчанию используется локальная среда. ***Путь к ВП*** - тот самый адрес VI в памяти, который был определен ранее. В данном случае этот вход достаточный для получении ссылки, данные на других входах задавать необязательно. Вход ***Опции*** определяет способы обработки ссылки на VI, в данном случае его использовать не требуется. ***Пароль*** - строка, содержащая пароль, установленный для защиты данного VI. Если VI не защищен, этот вход функции является необязательным (*о защите VI с помощью паролей - в заключительной части практикума*). "Антиподом" рассмотренной функции является ***Close Reference*** (Закрыть ссылку). Как видно из приведенной выше блок-диаграммы, эта функция использована по завершении работы со ссылкой, т.е., чтобы навести после себя порядок.

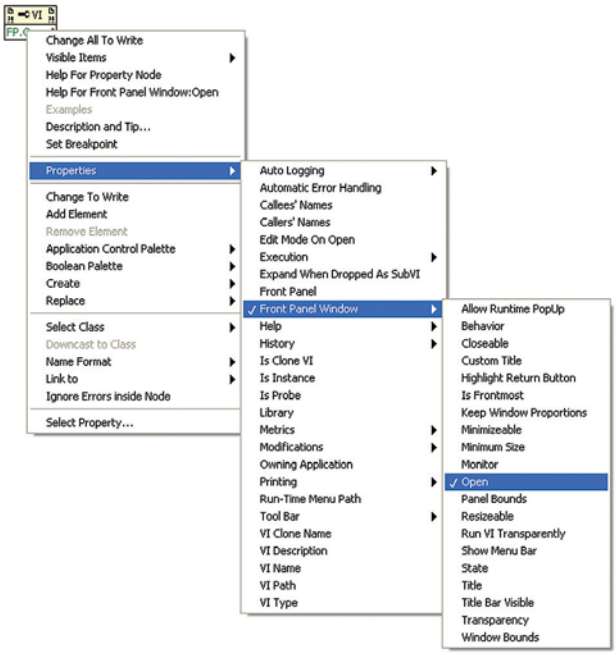


Если же Вы работаете не с VI, а с приложением, необходимо вместо функции ***Open VI Reference*** (Открыть ссылку на VI) использовать ***Open Application Reference*** (Открыть ссылку на Приложение), находящуюся в этом же подменю по соседству. Закрывается приложение той же ***Close Reference***. Далее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



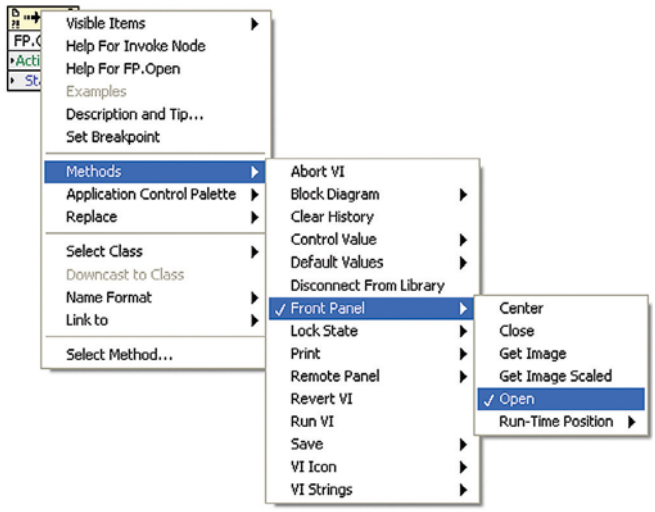
Узел свойств возвращает либо записывает свойства VI-приложения, на который указывает ссылка. Свойства - это разнообразные данные, например, открыта ли лицевая панель, каково ее название, активна ли она, и масса других. При выборе этой функции из подпалитры ***Управление приложением,*** еще не указано, какое из свойств необходимо возвратить либо записать. В этом случае в нижней части функции будет указано общее ***Property***. Выбор конкретного свойства осуществляется путем нажатия правой кнопки мыши на функции. При этом происходит появление выпадающего меню, где нужно выбрать ***Properties***.

Любознательному читателю не составит труда изучить разнообразные представленные свойства. Тем более, что из их названий интуитивно понятны их назначения.



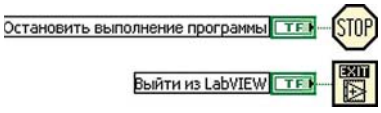
Узел вызова предназначен для осуществления какого-либо действия с VI, на который ссылается его вход. Выбор действий аналогичен соответствующей процедуре для ***Узла свойств***. Поскольку Вам необходимо открыть и запустить "этапную" программу, выбираем поочередно методы ***FP.Open*** и ***Run VI***. Следует заметить, что закрывать вызванную программу пользователь будет вручную, поэтому никаких соответствующих дейст-вий в данной блок-диаграмме не предусматриваем.

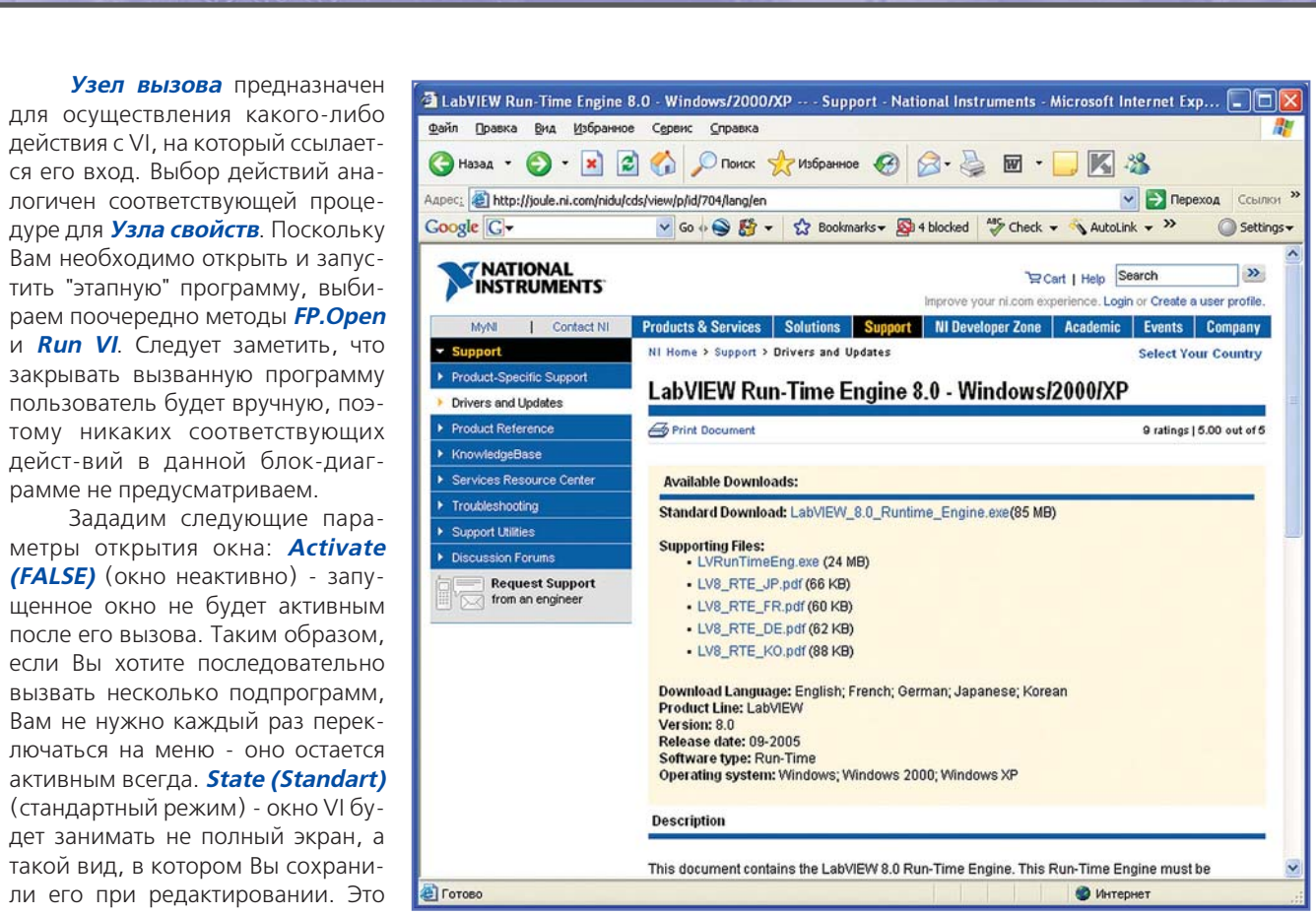
Зададим следующие параметры открытия окна: ***Activate (FALSE)*** (окно неактивно) - запущенное окно не будет активным после его вызова. Таким образом, если Вы хотите последовательно вызвать несколько подпрограмм, Вам не нужно каждый раз переключаться на меню - оно остается активным всегда. ***State (Standart)*** (стандартный режим) - окно VI будет занимать не полный экран, а такой вид, в котором Вы сохранили его при редактировании. Это касается и размера окна, и его положения среди других окон. Теперь при редатировании



каждой подпрограммы Вам нужно учесть это и расположить их лицевые панели по соседству и без накладок.

Отдельно следует отметить функции "финиша" - ***Stop*** (Остановка выполнения программы) и ***Quit LabVIEW*** (Выход из LabVIEW). В случае их вызова происходит то же самое, что и при нажатии ***Abort Execution*** в основном меню LabVIEW либо кнопки ***Закрытие приложения*** в его правом верхнем углу. Последнее весьма удобно, если открыто много всяких VI, либо работа с ними организована через VI-меню.





Следует заметить, что входы функций по умолчанию содержат значения ***TRUE***. Таким образом, вовсе не обязательно соединять их входы с какими-либо булевскими переменными или константами.

Если в процессе работы программы LabVIEW "натолкнулось" на одну из этих функций в блок-диаграмме, то деваться уже некуда - придется либо остановить программу, либо закрыть себя.

Раз уж мы затронули тему безопасности данных в LabVIEW, описывая создание поля ***Пароль*** в ***Приглашении пользователя***, давайте рассмотрим дополнительные возможности LabVIEW для сохранения секретности написанного Вами "исходного кода". Как известно, откомпилированное приложение, созданное в LabVIEW, можно запускать на компьютерах, где LabVIEW не установлено. В этом случае необходимо предустановить специальное приложение - утилиту ***Run-Time Engine***, которая доступна для "скачивания" на сайте ni.com, причем бесплатно.

Бесспорное и очевидное преимущество этого варианта - отсутствие необходимости в еще одной лицензионной версии LabVIEW. При этом следует заметить, что блок-диаграмму программы на рабочем ПК Вы увидеть не сможете, поскольку исходный код доступен лишь в среде разработки LabVIEW, но никак не в Run-Time. Таким образом, Ваша интеллектуальная собственность в данном случае защищена.

Однако есть и другой вариант распространения программ, написанных в LabVIEW, когда передается именно исходный код (*.vi-файл). Это может быть вызвано несколькими причинами, например - необходимостью доступа к исходному коду (фактически - к блок-диаграм-

ме и свойствам элементов лицевой панели, тем самым, к ***Properties***) для его модернизации либо адаптации "на месте". При этом на "принимающем" компьютере обязательно необходим LabVIEW. Если Вы намерены установить защиту своей программы от несанкционированного копирования, возможна установка пароля на доступ к блок-диаграмме и ***Properties***.

Пароль устанавливается следующим образом. Выберите пункт ***VI Properties*** в меню ***File***, при этом появится соответствующее окошко. В перечне пунктов меню ***Category*** выберите ***Protection***. Если Вы вызываете данное свойство виртуального свойства впервые, то, естественно, никакой пароль еще не установлен и из перечня появившихся вариантов будет выбран ***Unlocked (no password)***. Выбор ***Locked (no password)*** позволяет скрыть блок-диаграмму от нежелательных пользователей, поскольку отменяется повторным выбором ***Unlocked*** здесь же.

А вот выбор ***Password-protected*** является полномасштабной защитой - при его выборе Вы вводите новый пароль (дважды), который и будет защищать блок-диаграмму и запрашиваться при каждой попытке перехода на нее. Смена пароля возможна лишь при введении старого.

Очевидно, что введение пароля в приглашении пользователя теряет смысл, если блок-диаграмма доступна - этот пользователь свободно может подсмотреть общий пароль либо список индивидуальных логинов с паролями, открыв ее. Ну, а комбинация установки паролей в приглашении пользователя и пароля на блок диаграмму позволяет надежно защитить исходный код от незванных гостей.

Материал подготовлен студентами старших курсов Национального технического Университета Украины "Киевский политехнический институт" факультета авиакосмических систем, при технической поддержке ООО "ХОЛИТ Дейта Системс" (Киев).

e-mail: info@labview.com.ua