

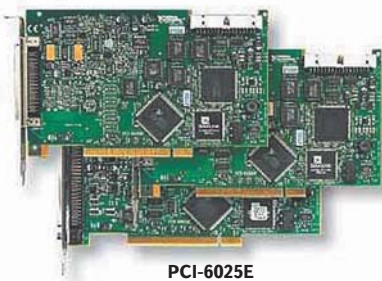
Уроки по LabVIEW

На этом уроке Вы познакомитесь с тем, как легко и просто работать с платой АЦП от National Instruments, а обладатели плат отечественного производства ADA-1292 смогут создать приложение, которое позволит ввести в компьютер реальный сигнал и выполнить спектральный анализ. Эти примеры помогут Вам освоить аппаратные средства любого уважающего себя производителя средств сбора данных для PC. Ну а тем, кому доступны только DLL-библиотеки для программирования под Windows, следует обратить особое внимание на последовательность создания иконки-функции для LabVIEW.



В предыдущих уроках инструментарий LabVIEW использовался для математического моделирования и обработки абстрактных данных. Но сила и мощь LabVIEW как раз и состоит в том, что можно так же просто создавать программы обработки и анализа данных, получаемых от реальных физических объектов. Принципиально для LabVIEW нет разницы между использованием виртуальных инструментов (функций или библиотек), которые осуществляют ввод/вывод реальных или виртуальных сигналов. Давайте создадим программы, позволяющие вводить и обрабатывать аналоговые сигналы, поступающие на один или несколько входов плат аналого-цифровых преобразователей, установленных в слоты компьютера.

Начнем с платы производства National Instruments. Пусть это будет популярный многофункциональный АЦП



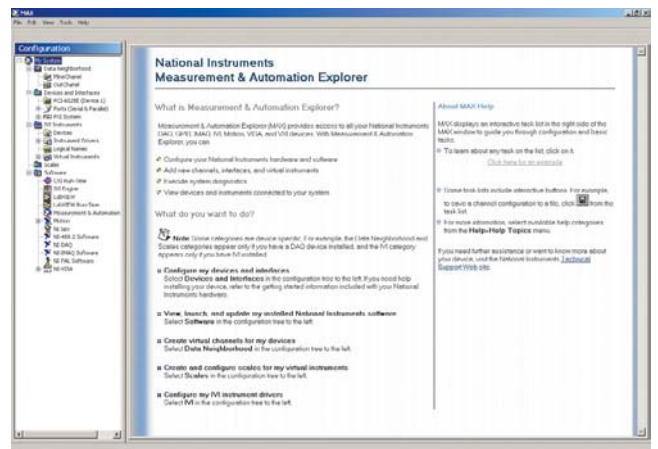
PCI-6025E

PCI-6025E. Плата имеет 16 однопроводных или 8 дифференциальных аналоговых входов с программируемым диапазоном от $\pm 0.05V$ до $\pm 10V$, АЦП 12 бит, 200 кГц, два ЦАП, 12

бит, 8 дискретных линий В/В и два таймера-счетчика по 24 бит. Неплохая плата - как и все, что производит National Instruments. Для LabVIEW эта плата как "родная" и интуитивно понятно, что никаких проблем с ней быть не должно.

Вся аппаратура, в том числе и встраиваемые платы компании National Instruments, поставляются с драйверами, собранными в единый программный пакет. Этот продукт называется NI-DAQ. Фактически NI-DAQ устанавливается по умолчанию в процессе установки LabVIEW. Поэтому все необходимые драйверы уже находятся в Вашем компьютере.

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. Он используется для конфигурации и тестирования аппаратной составляющей любого программно-технического комплекса NI.

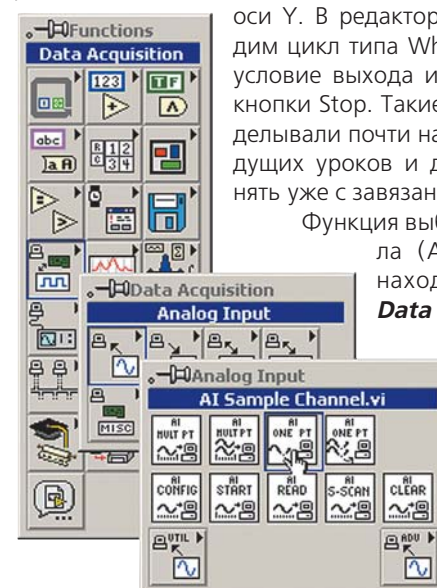


Подключим источник сигнала, например функциональный генератор (или батарейку с напряжением 3В, если это все, что имеется под рукой) к каналу "0" АЦП. Не забудем соединить все неиспользуемые каналы к контакту "общий" и проверить правильность конфигурации входных цепей (дифференциальная схема включения или схема с общим проводом).

А теперь приступим к программированию. Прежде всего, установим на интерфейсной панели графический индикатор (**Controls**) » **Graph** » **Waveform Chart** и определим опции автоматического масштабирования по оси Y. В редакторе диаграмм создадим цикл типа While Loop и зададим условие выхода из него по нажатию кнопки Stop. Такие операции Вы проделывали почти на каждом из предыдущих уроков и должны это выполнять уже с завязанными глазами.

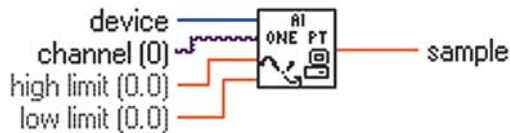
Функция выборки АЦП из канала (AI Sample Channel) находится в палитре **Data Acquisition** » **Analog Input** » **AI Sample Channel.vi**.

Стоит отметить, что если в системе установлено несколько плат фирмы NI, то каждому уст-



ройству будет присвоен свой индивидуальный номер. Этот номер будет отображен в программе MAX. Его необходимо использовать при работе с конкретными компонентами среды разработки.

Этот виртуальный инструмент позволяет делать выборку значений АЦП по одному каналу. Он имеет следующий вид:

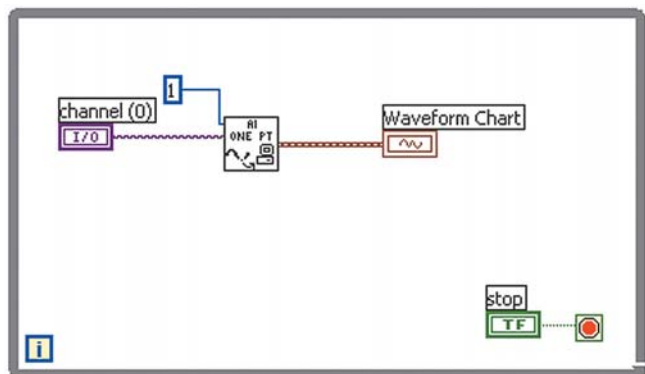
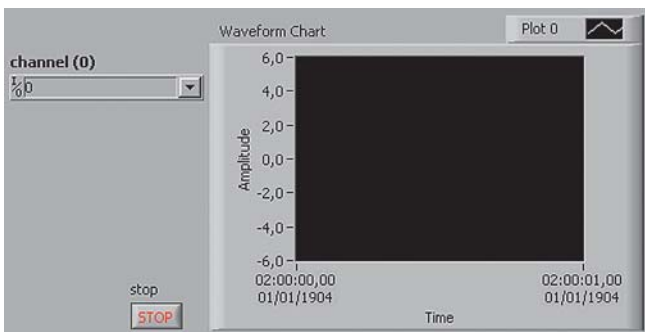


Device - это номер устройства в системе, а Channel - номер аналогового канала ввода, с которого будет осуществляться ввод сигнала.

High limit - максимальный ожидаемый уровень входного сигнала (по умолчанию "0"). Ограничение вводится в единицах результата считывания АЦП. **Low limit** - минимальный ожидаемый уровень входного сигнала (по умолчанию тоже "0"). При вводе "0" система использует ограничения входного сигнала, установленные в программе MAX. Ограничение вводится в единицах результата считывания АЦП.

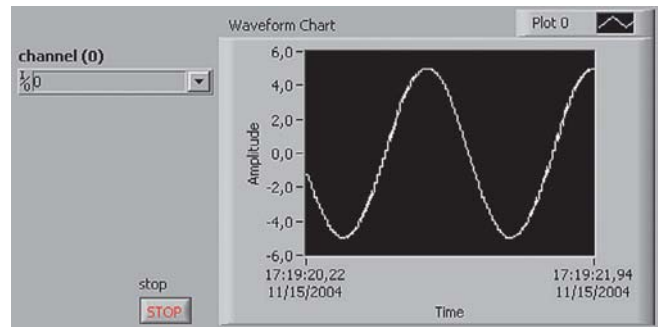
Sample - результат считывания АЦП по заданному каналу.

Для использования **AI Sample Channel.vi** минимум, что необходимо сделать, так это задать номер устройства и номер канала, из которого будет производиться выборка значений. Остальное мы уже умеем делать в LabVIEW.



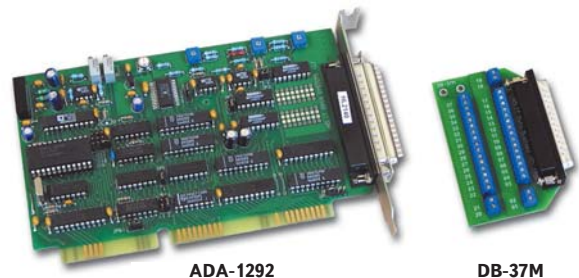
Вот и все. Готово. Запускаем программу. На вход канала "0" платы PCI-6025E подадим синусоидальный сигнал амплитудой 5В частотой 1000 Гц, который на экране монитора и наблюдаем:

Покрутим ручки "Амплитуда", "Частота" - на экране РС сигнал сжимается-расжимается по оси X и автоматически изменяется масштаб по оси Y. Действительно, как и было обещано, - по problems.



К сожалению, далеко не каждый сможет практически реализовать приведенный простенький пример, ведь не у всех найдется АЦП от National Instruments. Не беда, возьмем что-либо более доступное - плату отечественного производства ADA-1292 ("ХОЛИТ Дэйта Системс", Украина).

Модель ADA-1292 выполнена в стандарте IBM PC AT и устанавливается в любой из свободных ISA-слотов. ADA-1292 является функционально полным комплексом, включающим в себя многоканальный 12-ти разрядный



ADA-1292

DB-37M

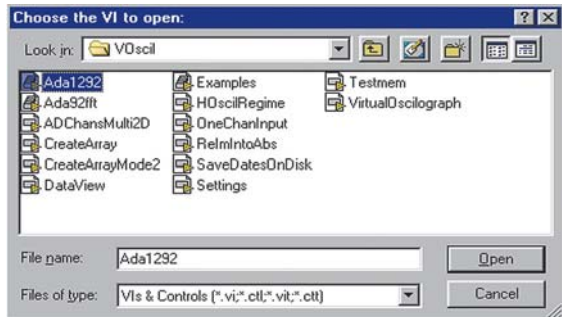
АЦП с программируемым входным диапазоном и частотой преобразования до 200 КГц на один канал, 12-ти разрядный ЦАП, 8 дискретных входов и 8 выходов в уровнях ТТЛ. Генерация прерываний и внутренняя синхронизация обеспечивается тремя 16-ти разрядными таймерами с опорной частотой 8 МГц. Предусмотрены различные типы синхронизации АЦП и таймера. Устанавливаем плату ADA-1292 в свободный слот.

В комплект программного обеспечения, поставляемого с платой, входят динамическая библиотека для Windows **WI1292.dll** и библиотека **ada1292.lib**, включающая в себя все необходимые виртуальные инструменты (функции) для работы с "железом" в LabVIEW. И для написания первой программы, которая позволит последовательно, отсчет за отсчетом, вводить значения напряжения на входе выбранного канала, будем использовать функцию (VI) этой библиотеки **ADChan.vi**.

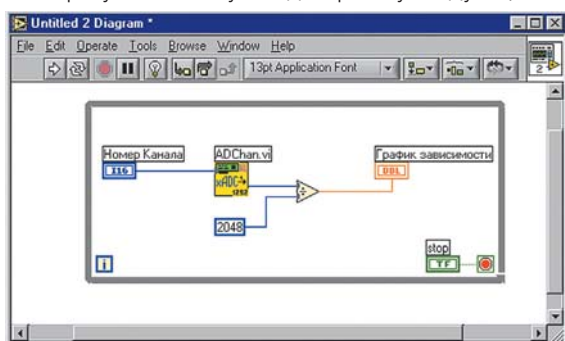
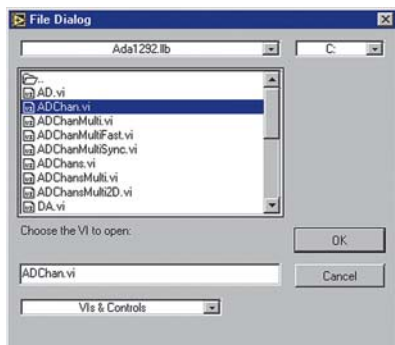
Напишем первую программу. Создаем новое приложение. Устанавливаем на интерфейсную панель элемент управления (**Controls**) > **Numeric** > **Digital Control**, определяющий номер канала, с которого будет выполняться ввод сигнала. Далее необходимо изменить тип данных на целочисленный. Для этого нажмем правой клавишей мыши на объекте, и в выпадающем меню выберем пункт **Representation**, а в нем - **Word**. Устанавливаем элемент поточечного отображения сигнала в виде графика с автоматическим масштабированием по оси Y. И, завершая первый этап программирования, помещаем схему в цикл с условием по нажатию кнопки stop. Т.е. почти все, как и в случае с американской платой.

Следующий этап - самое интересное и главное в этом уроке. Переключимся в окно редактирования диаграмм и будем выполнять установку иконки (виртуальный

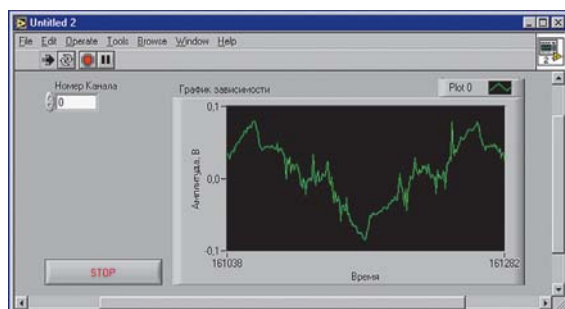
инструмент), который позволяет считывать значения амплитуды входного сигнала для заданного канала. Для этого необходимо на функциональной панели выбрать **(Functions)** » **Select a VI...** В появившемся окне сначала выбираем библиотеку **ada1292.lib**, а уже в ней функцию **ADChan.vi**. Результат - иконка на диаграмме.



Значения, которые возвращает эта функция в результате аналогово-цифрового преобразования, лежат в диапазоне от -2048 до +2047. Минимальное значение соответствует значению -5В (или -10В), в зависимости от выбранного входного диапазона сигнала. А максимальное, соответственно, +5В или +10В. Поэтому для получения значения не в условных единицах измерения, а в вольтах, необходимо получаемое значение умножить на 5 или 10, и разделить на 2047. Введем соответствующие блоки математических преобразований и выполним необходимые соединения. В результате получим диаграмму следующего вида:



Не будем подавать гармонический сигнал на вход АЦП, пусть сигналом будет помеха. Запускаем программу на выполнение. В результате получится примерно так:



Созданное приложение есть смысл сохранить.

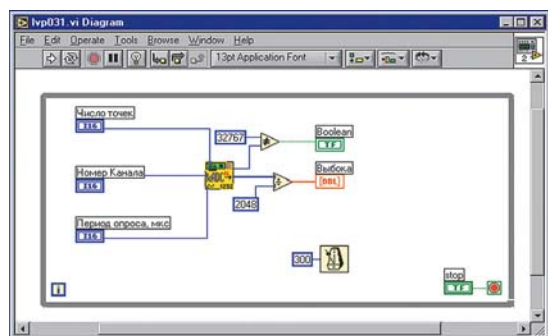
Создав такую программу буквально за считанные минуты, становится очевидным, как можно элементарно просто работать в LabVIEW не только с техникой National Instruments. Причем, и это стоит в очередной раз подчеркнуть, совсем нет необходимости досконально знать аппаратную часть платы, и тем более писать несколько десятков, сотен или даже тысяч строк кода на текст-ориентированных языках программирования высокого уровня.

Давайте напишем еще одну программу, используя библиотеку **ada1292.lib**. Пусть необходимо вводить сигнал по заданному каналу и выдавать результат в виде выборки чисел (массива). Задаваемые параметры: номер канала, число точек и период опроса в мкс. А возвращаемые значения: выходной массив точек и код ошибки (32767) при обнаружении таймаута. Для решения сформулированной задачи будем использовать функцию (VI) **ADChanMulti**, входящей в состав упомянутой библиотеки. Создаем новое приложение.

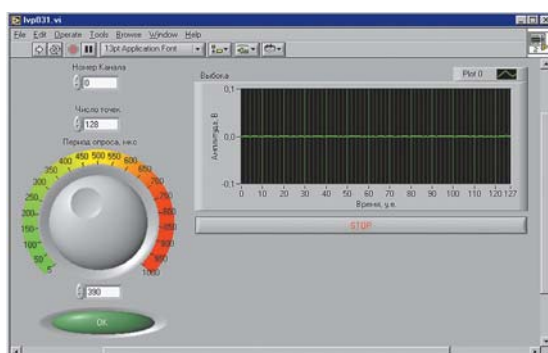
Оформим интерфейсную часть программы. Сначала установим элементы управления, определяющие номер канала, число точек и период опроса. Разместим также и элемент построения графических зависимостей: **(Controls)** » **Graph** » **Waveform Graph**. Следует отметить, что использовать следует именно **Waveform Graph**, а не **Waveform Chart**, так как строиться будет массив точек. Добавим еще и элемент индикации, который будет извещать о таймауте. Переименовываем и размещаем все элементы управления и индикации соответствующим образом и переключаемся в окно редактирования диаграмм.

Диаграмма будет выглядеть приблизительно так, как и в предыдущем примере. Установим виртуальный инструмент (иконку) для ввода массива точек. В **Functions** выбираем **Select a VI...** Опять в появившемся окне выбираем сначала библиотеку **ada1292.lib**, а в ней функцию - **ADChanMulti.vi**.

Выполним соответствующие соединения на диаграмме (аналогично предыдущему примеру). Но кроме этого добавляем еще и проверку таймаута с выводом на индикатор. В результате должна получиться приведенная ниже диаграмма:



а передняя панель, после запуска программы примет вид:



Не спешите подавать тестовый сигнал. Свободное место на интерфейсной панели оставлено не случайно, а с целью дальнейшей модернизации программы. Предлагается для вводимого сигнала выполнить цифровую обработку, а именно построить амплитудный спектр сигнала. Будем использовать алгоритм Быстрого Преобразования Фурье. Следует сразу оговорить, что с целью простоты восприятия принципов графического программирования, числа по обеим осям графика спектра не нормированы.

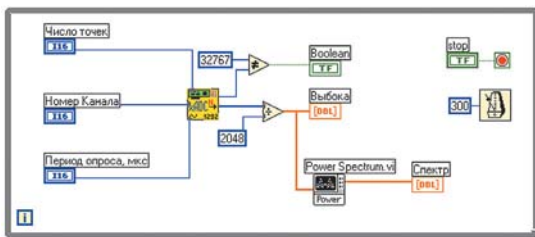
Добавим еще один элемент отображения данных в виде графика **Waveform Graph** и изменим тип графика на график в виде спектра. Для спектральной обработки установим иконку (VI), которая находится: **(Functions) » Signal Processing » Frequency Domain » Power Spectrum.vi**.

Следует отметить, что в LabVIEW существует множество других виртуальных инструментов для спектральной обработки и анализа сигналов,

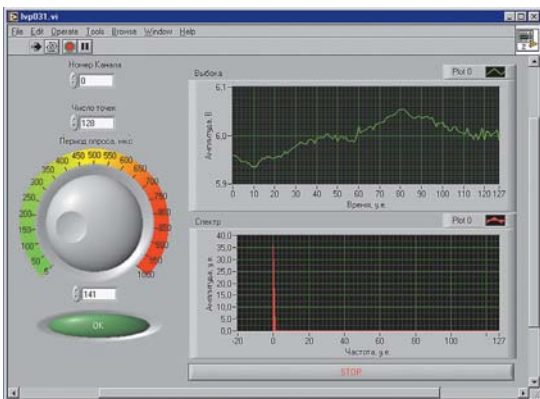


но мы используем самый простой, не требующий ввода дополнительных параметров.

Сделаем необходимые элементарные соединения. Всего один функциональный блок и две связи, и в результате получим такую диаграмму:



Можно запускать программу на выполнение:



Что в итоге мы имеем? Программу осциллографа-анализатора спектра для конкретной платы. Здорово! Не правда ли? Причем построение диаграммы и интерфейса

заяло всего несколько минут, ну может быть десятков минут. Если бы мы реализовывали эту задачу на "текстовых" языках программирования, то для этого потребовалось бы намного больше времени и сил. Кроме того, диаграмма, т.е. листинг программы, имеет очень простой вид для понимания алгоритма ввода и обработки сигнала. Недаром ведь LabVIEW создали для инженеров, а не программистов

Будем считать, что Вами и эта часть начального курса по LabVIEW успешно пройдена. А если добавить еще навыки, полученные на предыдущих уроках, то можно сказать, что Вы более не нуждаетесь в услугах программистов-профессионалов. То есть почти не нуждаетесь.

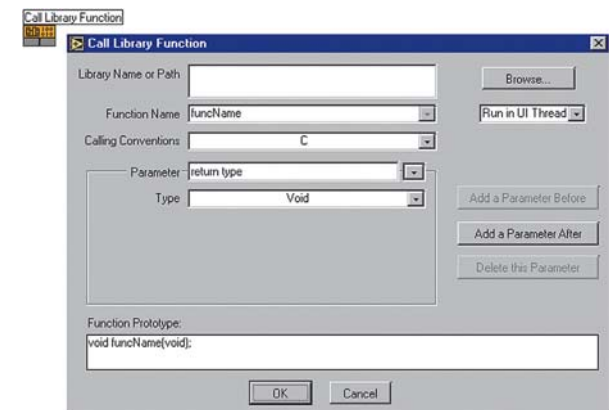
Давайте научимся использовать DLL-библиотеки, а точнее функции этой библиотеки. Зачем? Затем, что "некоторые" производители плат расширения и другого оборудования для PC не поставляют LLB-библиотеки для LabVIEW, а включают в комплект поставки только DDL-библиотеки, содержащие все необходимые функции для работы с платами в среде Windows. Такие библиотеки могут быть использованы в популярных средах программирования, таких как Borland Delphi, Borland C++ Builder, Microsoft Visual C++, Microsoft Visual Basic и других. И LabVIEW также не является исключением.

Сделаем вызов функции из DLL-библиотеки, которая позволяет поточно вводить данные по указанному каналу для платы ADA-1292. Т.е. создадим свой, аналогичный используемому ранее, виртуальный инструмент (иконку-функцию). Как уже было упомянуто, для работы с платой ADA-1292 под Windows поставляется соответствующая библиотека - **WI1292.dll**.

Пользуясь описанием на плату и библиотеку, находим, что необходимая нам функция называется **ADChan1292**. Она имеет формат: **int16 ADChan(uInt16 chan);** где параметр **chan** - номер канала.

Создаем новое приложение и переключаемся в окно редактирования диаграмм. Выполним вызов необходимой функции из DLL-библиотеки: **(Functions) » Advanced » Call Library Function**.

Щелкните дважды левой клавишей мыши на вновь



появившейся иконке. Появится форма в виде диалогового окна для вызова необходимой функции.

Далее следует ввести имя библиотеки **WI1292.dll** в поле **Library Name or Path** или найти ее с помощью клавиши **Browse**. В выпадающем списке **Function Name** следует выбрать функцию **ADChan1292**.

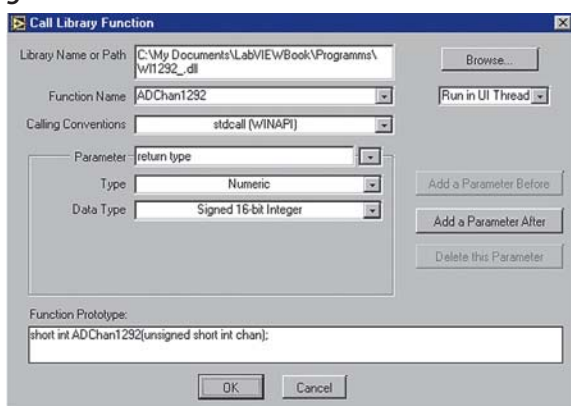
Calling Conventions следует изменить на **stdcall (WINAPI)**.

Параметр **Type** изменяем на **Numeric**.

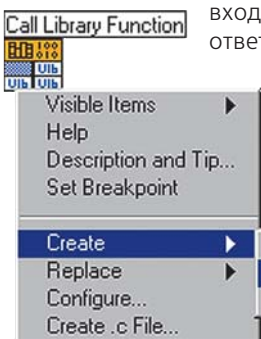
В появившемся дополнительном выпадающем списке **Type** следует выбрать **Signed 16-bit Integer**, в соответствии с описанием на библиотеку.

Для определения типа данных параметра, определяющего номер канала, необходимо нажать кнопку **Add a Parameter After**.

Переименовываем параметр в **chan** и изменяем тип представления данных в **Data Type** на **Unsigned 16-bit Integer**.

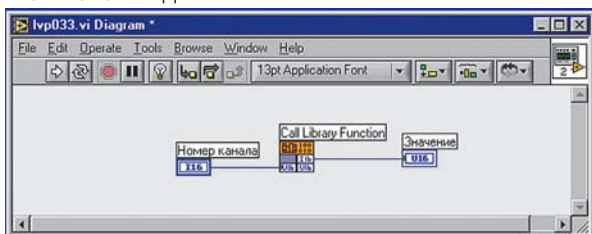


Создадим элемент управления и индикатор для входного и выходного параметров соответственно. Для этого необходимо подвести указатель мыши в виде катухки к входному параметру **chan** (слева) и выбрать создание элемента управления.

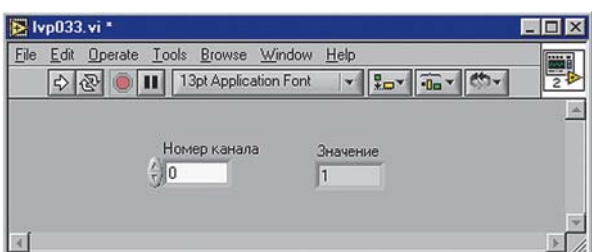


Для выходного параметра **Output Value**, создаем элемент индикации по аналогии. Переименовываем параметры. В результате диаграмма будет

иметь такой вид:

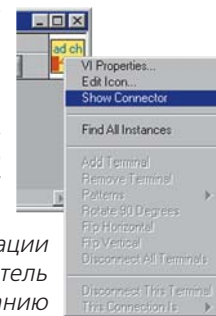


а интерфейсная панель:

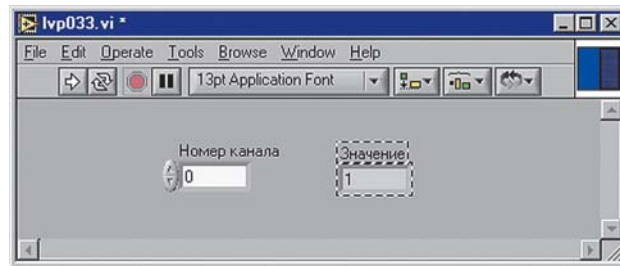


Вот так, разрабатывая приложения, можно напрямую вызывать функции из DLL-библиотеки. Конечно, это менее удобно и для этого требуется время. Поэтому лучше на основе DLL-библиотеки создать свою LLB-библиотеку, или, как минимум, сохранить используемую процедуру В/В просто в виде отдельного виртуального инструмента.

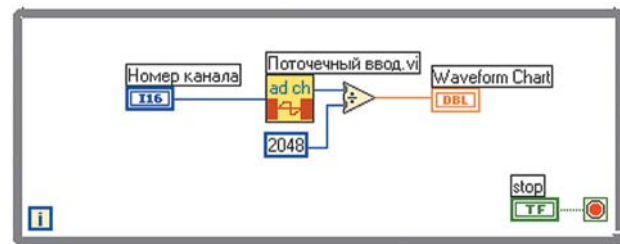
Создаем свою собственную иконку-функцию. Она позволит выполнять поточечный ввод значений через выбранный канал и в дальнейшем будет использоваться в Ваших приложениях. Для этого необходимо вспомнить как создаются в LabVIEW "подпрограммы" (ПИКАД №1, 2004, Урок №2, стр. 51,52): *...для реализации этого необходимо подвести указатель мыши к установленной по умолчанию иконке в правом верхнем углу интерфейсной панели и сделать двойной щелчок левой клавишей мыши. Появится графический редактор иконок со стандартным набором инструментов...* Вспомнили? Тогда рисуем иконку.



Устанавливаем соответствие между элементами управления и индикаторами с ячейками соединителя и сохраняем файл в своей директории (папке) под желаемым именем, например "Поточечный ввод".



Для проверки выполненных действий постройте самостоятельно диаграмму, аналогичную предыдущей, но уже используя уже вновь созданную функцию (VI):



Нет сомнений в том, что это у Вас получится. Это означает - в LabVIEW можно работать с платами для PC от любого производителя. В порядке закрепления урока попробуйте разработать драйвера для Ваших ISA/PCI-устройств под LabVIEW.

Уроки по LabVIEW

На следующем уроке:

Будут рассмотрены вопросы взаимодействия с коммуникационными портами и, прежде всего, с COM-портом. В качестве аппаратуры используем популярные модули для распределенных систем сбора данных i-7000/8000 и tetraCON. Познакомимся также и с локальными переменными.

