

# ОГЛАВЛЕНИЕ



<b>Введение.....</b>	<b>5</b>
<b>▼ Глава 1. Основы создания приложений в Measurement Studio .....</b>	<b>8</b>
Использование программы Measurement & Automation Explorer.....	10
Задания.....	17
Наше первое приложение в среде Measurement Studio.....	23
<b>▼ Глава 2. Принципы создания приложений на базе шаблонов NI DAQ Windows .....</b>	<b>31</b>
Непрерывный режим работы NI DAQ Windows приложений.....	35
<b>▼ Глава 3. Измерение параметров сигналов в NI DAQ Windows приложениях.....</b>	<b>50</b>
Сохранение результатов измерений.....	58
Измерения временных характеристик сигналов.....	62
Использование шаблона NI Windows Application в приложениях автоматизации измерений.....	69
<b>▼ Глава 4. Прецизионные измерения параметров сигналов в приложениях NI Measurement Studio .....</b>	<b>79</b>

Ввод дискретных сигналов ..... 94

Система измерения и анализа температуры  
окружающей среды..... 105

▼ **Глава 5. Синтез сигналов в приложениях  
NI Measurement Studio ..... 111**

▼ **Глава 6. Программирование  
последовательных интерфейсов  
в приложениях NI Measurement Studio ..... 144**

Тестирование и настройка последовательных  
интерфейсов ..... 146

Передача данных через последовательный порт  
в приложениях NI VISA ..... 149

Чтение данных через последовательный интерфейс  
в приложениях использующих NI VISA..... 162

▼ **Глава 7. Сетевые коммуникации  
в приложениях NI Measurement Studio ..... 167**

# ВВЕДЕНИЕ



Пакет программ Measurement Studio фирмы National Instruments представляет собой расширение популярной среды разработки Microsoft Visual Studio набором программных утилит и библиотеками классов, которые позволяют разрабатывать системы измерения и управления при помощи популярных языков программирования, таких как Microsoft Visual Basic .NET и Visual C# .NET.

Данный программный инструмент позволяет существенно ускорить разработку и отладку приложений для систем автоматизации с помощью объектно-ориентированных методов построения аппаратных измерительных интерфейсов, программных библиотек для анализа и обработки сигналов, а также широкого спектра графических компонентов пользователя для Windows и Web-приложений.

Пакет NI Measurement Studio предоставляет разработчику широкий спектр программных средств и включает следующие библиотеки классов .NET:

- **библиотеку для анализа и обработки сигналов (Analysis).** В эту библиотеку включены классы, позволяющие выполнять цифровую обработку сигналов, их фильтрацию, цифровой синтез, а также математические функции общего назначения. Эту библиотеку можно использовать для анализа и обработки входных сигналов, поступающих в систему от датчиков и измерительных преобразователей, а также для генерации выходных сигналов для управления

различными устройствами. Кроме того, можно воспользоваться многочисленными примерами программного кода, включенного в документацию по данной библиотеке классов. Более подробно мы ознакомимся с возможностями этой библиотеки при разработке примеров приложений.

- **библиотеку Common.** Эта библиотека классов и содержит набор классов, которые обеспечивают передачу и преобразование данных между различными функциональными частями системы автоматизации. С помощью этой библиотеки осуществляется взаимодействие между блоками измерения, блоками анализа сигналов и интерфейсом пользователя приложения.
- библиотеку *Network Variable*. Разработчик может использовать эту библиотеку классов для обмена данными между сетевыми клиентами и сетевыми серверами.
- **библиотеку классов MCC-488.2.** Эта библиотека оказывается весьма полезной, если требуется создать интерфейс с устройствами и измерительными инструментами, работающими с шиной GPIB. Для использования этой библиотеки необходимо предварительно установить драйверы MCC-488.2. Для разработки интерфейса с GPIB-устройствами можно использовать классы *Device* и *Board*.
- **библиотеку Universal.** Для того, чтобы начать работу с этой библиотекой, необходимо установить *Universal Library driver*. Классы данной библиотеки используются для создания программных интерфейсов с устройствами сбора данных (DAQ). Мы будем часто обращаться к этой библиотеке в процессе разработки примеров систем измерения и контроля.
- **библиотеку Scan Components.** В эту библиотеку включены классы для чтения/записи аналоговых (непрерывных) данных по нескольким физическим каналам.
- **библиотеку пользовательских компонентов.** Многие специализированные пользовательские компоненты управления пакета *Measurement Studio* включаются в библиотеки классов *Windows Forms* и *Web Forms .NET*. Мы часто будем использовать компоненты этих библиотек для визуализации результатов измерений и генерации сигналов. Для каждого типа измерений *Measurement Studio* предоставляет

пользовательские элементы интерфейса, которые можно при необходимости размещать и совмещать произвольным образом для решения каждой конкретной задачи. Среди доступных элементов управления имеются различные кнопки, ручки, линейные и круговые регуляторы, датчики, резервуары, светодиоды и измерительные приборы. Широкий набор элементов, имеющийся в Measurement Studio, позволяет реализовать более информативное представление данных, по сравнению с традиционными приборами.

Кроме того, NI Measurement Studio включает целый ряд вспомогательных инструментов для организации обмена данными с помощью сетевых протоколов. Для реализации подобных возможностей в данный пакет включены многочисленные библиотеки классов, которые интегрируются в среду .NET. Так, например, с помощью библиотеки Network Variable разработчик может реализовать алгоритмы обмена данными между сетевыми клиентами и сетевыми серверами.

Все примеры приложений для этой книги были разработаны в среде Microsoft Visual Studio 2010 Professional с установленным пакетом NI Measurement Studio 2010. В качестве операционной системы использовалась Windows 7. В качестве аппаратного интерфейса систем автоматизации был использован модуль DAQ PCI-6221 фирмы National Instruments.

# Основы создания приложений в Measurement Studio



1

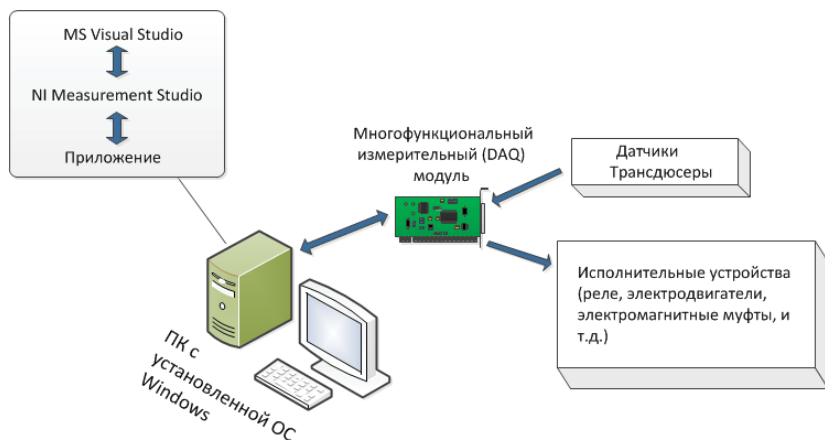
Для создания эффективных приложений, предназначенных для систем автоматизации, необходимо четко представлять себе как структуру самого приложения, так и взаимодействие программного кода системы автоматизации с аппаратными интерфейсами и операционной системой.

В данной главе мы проанализируем ключевые моменты разработки систем автоматизации и управления с использованием программной среды NI Measurement Studio. В упрощенном виде систему автоматизации на базе приложения Measurement Studio можно представить в виде схемы, показанной на рис. 1.1.

Как видно из рисунка, наша система автоматизации должна включать как минимум аппаратный измерительный интерфейс, в качестве которого можно использовать один из многочисленных многофункциональных измерительных (DAQ) модулей, выпускаемых промышленностью. В данной книге в качестве такого интерфейса мы будем использовать модуль PCI-6221 фирмы National Instruments, хотя все рассматриваемые проекты можно реализовать с минимальными изменениями, используя многочисленные DAQ устройства.

Для того, чтобы задействовать выбранный DAQ модуль, нужно установить его в слот расширения персонального компьютера и установить программное обеспечение нижнего уровня (драйверы операционной системы для устройства), а также программы конфигурирования устройства высокого уровня для настройки и проверки устройства посредством графического интерфейса пользователя. Для конфигурирования нашего DAQ модуля используется специ-

альная утилита Measurement & Automation Explorer (MAX) фирмы National Instruments – ее применение мы рассмотрим более подробно в этой главе.



**Рис. 1.1**

К модулю DAQ можно подключать внешние устройства и/или электронные схемы, осуществляющие непосредственное преобразование и передачу сигналов от измерительных датчиков в систему автоматизации и/или передачу сигналов от системы автоматизации к исполнительным устройствам. Модули DAQ можно рассматривать как интерфейсы между аппаратной и программной частью системы.

Собственно среда разработки включает в себя MS Visual Studio 2010 с установленным модулем расширения NI Measurement Studio 2010. В качестве операционной системы используется 32-разрядная ОС Windows 7, хотя все приложения можно успешно компилировать и выполнять и в более ранних ОС, таких как Windows XP или Windows Vista. В процессе инсталляции системы NI Measurement Studio в Visual Studio добавляются библиотеки классов для обработки данных с аппаратных измерительных интерфейсов, а также визуальные компоненты пользовательских интерфейсов. После успешной инсталляции Measurement Studio можно разрабатывать приложения автоматизации с использованием языка C# .NET или Visual Basic .NET.

Перед тем как начинать разработку программного обеспечения нам потребуется настроить аппаратную часть при помощи программы Measurement & Automation Explorer.

## Использование программы Measurement & Automation Explorer

В этом разделе мы рассмотрим вопросы конфигурирования аппаратных интерфейсов модулей DAQ с помощью программы Measurement & Automation Explorer (MAX). Программа MAX позволяет выполнить настройку, конфигурирование и тестирование аппаратно-программных интерфейсов, а также калибровку аппаратной части устройств DAQ поддерживаемых NI-DAQ/NI-DAQmx. С помощью MAX можно просмотреть устройства и инструменты, подсоединенные к системе.

Кроме того, программа MAX позволяет создавать именованные конфигурации системы сбора и генерации сигналов в форме «заданий» (tasks). Все приложения, разрабатываемые с среде NI Measurement Studio, используют *задание* в качестве программного интерфейса для взаимодействия с модулем DAQ. *Задание* позволяет разработчику программного обеспечения абстрагироваться от конкретной аппаратной реализации того или иного измерительного интерфейса и использовать стандартные программные методы для доступа к измерительным интерфейсам.

Все корректно установленные модули DAQ обнаруживаются и отображаются программой MAX при запуске. В нашем случае при использовании устройства NI PCI-6221 будет отображена следующая информация (рис. 1.2):

В окне конфигурации My System в категории Devices and Interfaces отображается наш модуль DAQ NI PCI-6221, который фигурирует в системе как устройство Dev1. В окне справа отображаются параметры устройства в системе Windows (Socket Number, номер слота на шине PCI и Bus Number, номер PCI-шины). Эту и более подробную информацию об устройстве можно обнаружить, вызвав консоль управления ПК (значок **Computer** → **Manage**), затем перейти в Device Management. Устройства DAQ фирмы NI в консоли управления отображаются в категории Data Acquisition Devices (рис. 1.3).



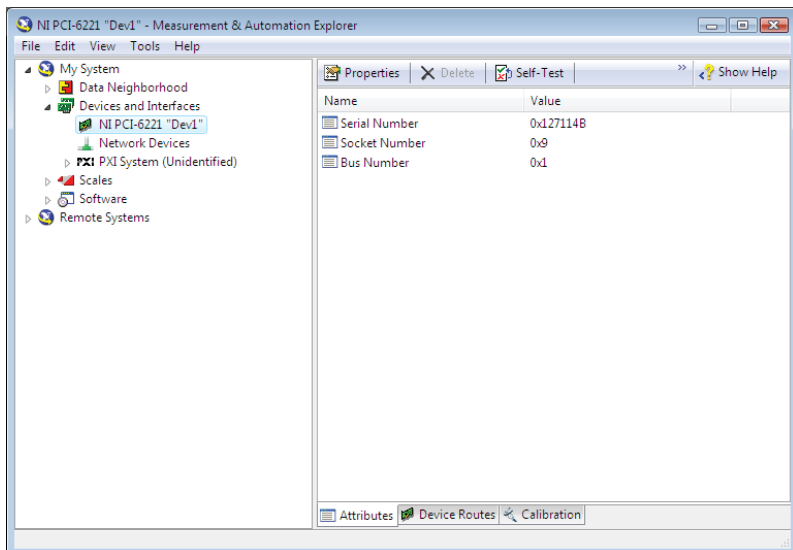


Рис. 1.2

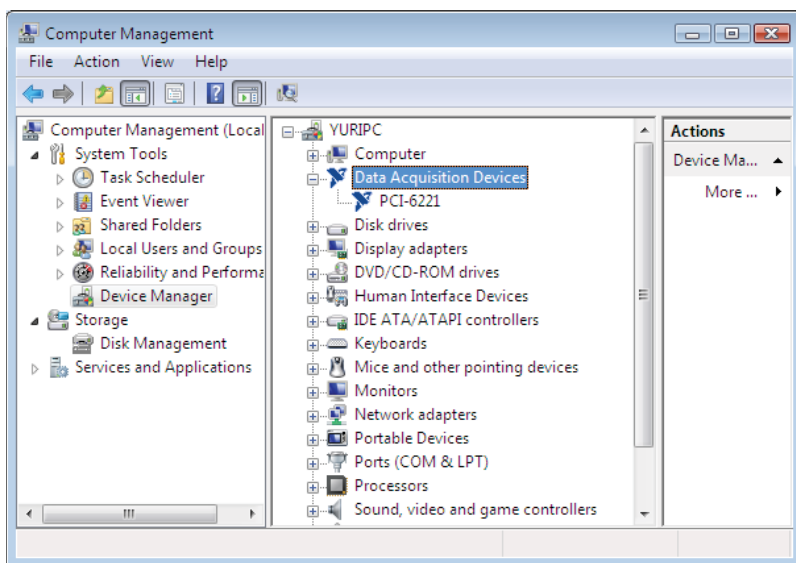


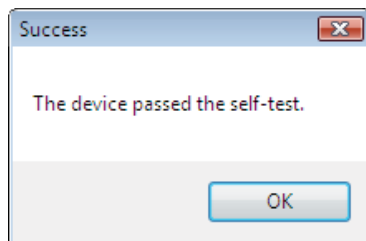
Рис. 1.3

После выбора устройства DAQ (PCI-6221, в нашем конкретном случае) можно просмотреть о нем детальную информацию и, кроме того, обновить при необходимости драйвер. Конфигурация устройства в Windows должна быть проанализирована в первую очередь, после установки программного обеспечения нижнего уровня (драйверов устройств). Программа MAX использует системные настройки устройства для конфигурирования DAQ-устройства на высоком уровне.

После того, как устройство DAQ появилось в списке сконфигурированных устройств MAX, желательно выполнить диагностику работоспособности устройства, выбрав закладку Self-Test в правой панели окна MAX. В процессе самотестирования выполняется проверка конфигурации аппаратных узлов устройства и доступ к внутренним регистрам управления/данных/состояния. Следует отметить, что в процессе тестирования не проверяются калибровочные данные устройства – используются заводские значения. Для точной калибровки устройства должна быть выполнена специальная процедура, предусмотренная в программе MAX.

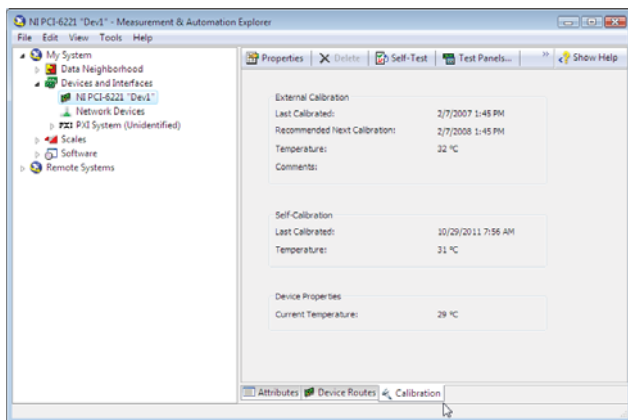
Если тестирование устройства прошло успешно, то на экран дисплея будет выдано соответствующее сообщение (рис. 1.4).

Перед использованием модуля DAQ в системе управления/измерения нужно обязательно провести калибровку устройства. Процесс калибровки нужен для того, чтобы, во-



**Рис. 1.4**

первых, проверить точность измерений, выполненных устройством DAQ, и, во-вторых, настроить модуль для работы с минимальной погрешностью. Перед началом калибровки желательно просмотреть данные по последней калибровке (если устройство выпущено недавно, то это может быть заводская калибровка). Кроме того, что очень важно, с течением времени параметры любого электронного устройства обычно «уходят» в сторону от своих номинальных значений. По этой причине желательно периодически выполнять повторную калибровку модуля DAQ, о чем будет указано на странице своей Calibration. Саму страницу Calibration можно выбрать, щелкнув на закладке Calibration справа внизу (рис. 1.5).

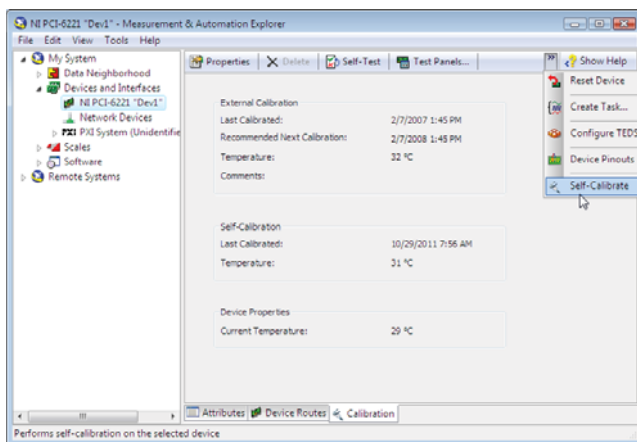
**Рис. 1.5**

На странице Calibration указывают дату выполнения заводской калибровки (External Calibration) и рекомендуемую дату для ее повторного выполнения. Если устройство работает в составе системы, то следует выполнить самокалибровку системы.

Здесь имеются два очень важных момента, которые следует учитывать. Как было сказано ранее, параметры любого измерительного устройства, и модуль DAQ не является здесь исключением, со временем изменяются в ту или иную сторону, что приводит к погрешностям в измерениях. Даже если устройство функционирует в одних и тех же климатических условиях (температура, влажность, давление) и при неизменных электрических параметрах питания и уровне электромагнитных помех, внутренние процессы в кристалле полупроводника приводят к постепенной деградации устройства. По этой причине через определенные промежутки времени следует выполнять самокалибровку устройства DAQ, для чего указывается дата следующей калибровки (Recommended Next Calibration).

Второй момент заключается в том, что параметры устройства DAQ изменяются при изменении климатических условий, в которых оно функционирует. Важнейшим фактором нестабильности здесь является температура окружающей среды, поскольку полупроводниковые электронные компоненты довольно существенно изменяют свои характеристики с изменением температуры. По этой причине на момент выполнения калибровки указывается температура окружающей среды.

Как видно из рис. 1.5, данный модуль был откалиброван при температурах 32° С и 31° С. Если постоянные температурные условия, при которых функционирует модуль DAQ, существенно отличаются от тех, при которых он был ранее откалиброван, то обязательно следует выполнить перекалибровку устройства независимо от того, какой период времени прошел с момента последней калибровки. Для повторной калибровки модуля DAQ следует выбрать опцию Self-Calibrate (рис. 1.6). При выполнении калибровки устройства в открывающемся окне будет отображаться отчет о состоянии процесса. После завершения калибровки окно закрывается.

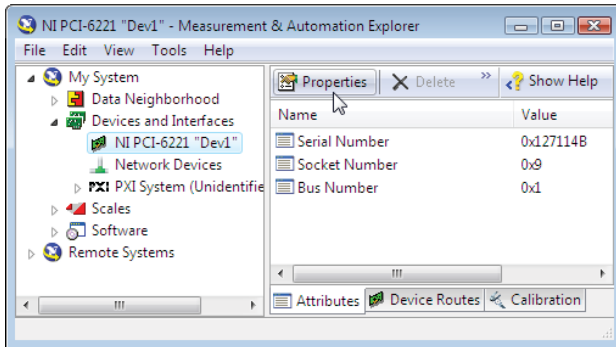


**Рис. 1.6**

Более подробную информацию, касающуюся калибровки устройств DAQ и подгонки их параметров, можно найти в руководствах пользователя (DAQ Getting Started guides for NI-DAQ 9.2). Для установки параметров «по умолчанию» служит опция Reset Device.

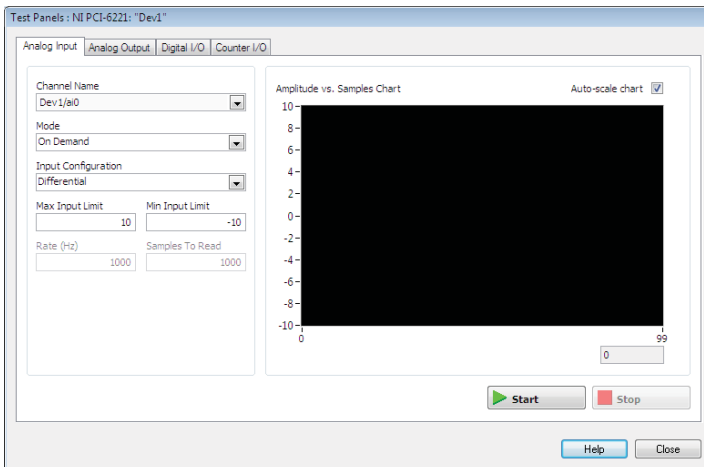
Перед началом работы с устройством в системе необходимо точно знать начальное состояние сигналов на выходах модуля DAQ в момент включения питания – это особенно важно, если к модулю подключены какие-либо исполнительные устройства. Состояние выходов модуля DAQ должно быть четко определено, в частности, при подаче питающих напряжений на шину питания устройств. По умолчанию выходы устройства DAQ переходят в высокоимпедансное состояние при включении питания, и это нужно учитывать при конфигурировании внешних устройств, подключенных к этим выходам.

Для проверки работоспособности устройства DAQ нужно в правой панели стандартного окна программы MAX выбрать опцию меню Test Panel (рис. 1.7).



**Рис. 1.7**

После появления диалогового окна Test Panels можно выбрать, сконфигурировать и запустить соответствующий тест для данного модуля DAQ. Каждый модуль имеет свой специфичный набор периферийных устройств, соответственно и опции диалогового окна будут разными для разных устройств. Например, для модуля NI PCI-6221 диалоговое окно Test Panels будет выглядеть так, как показано на рис. 1.8.

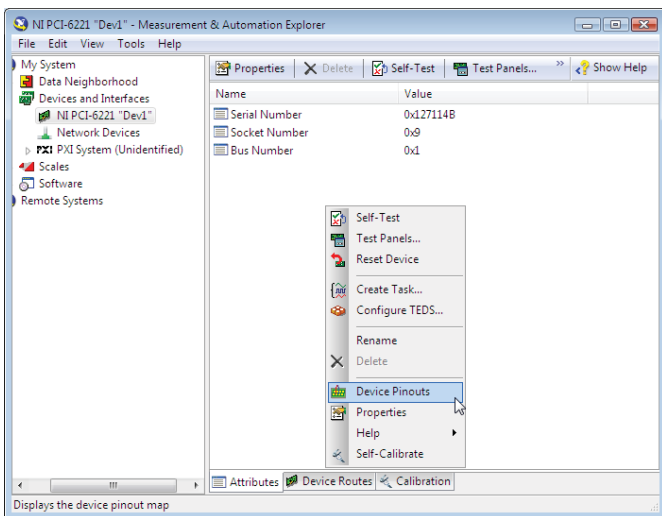


**Рис. 1.8**

Закладки диалогового окна отображают тестовые окна для всех встроенных периферийных устройств данного модуля. В нашем случае можно выполнить тестирование цифровых портов ввода/вывода (закладка Digital I/O), таймеров/счетчиков (закладка Counter I/O), физических каналов аналого-цифрового преобразователя (закладка Analog Input) и физических каналов цифро-аналогового преобразователя (закладка Analog Output). Рассмотрим процесс тестирования более подробно и начнем с цифровых каналов ввода/вывода.

Следует отметить, что методика тестирования, которая будет продемонстрирована для данного конкретного устройства NI PCI-6221, подойдет и для других устройств, в которых имеются аналогичные компоненты.

Перед началом тестирования модуля DAQ нам необходимо знать соответствие сигналов устройства DAQ номерам выводов на внешнем разъеме, подсоединенном к модулю, поскольку в процессе тестирования возникнет необходимость в измерениях уровней сигналов на этом разъеме. Для этого нужно щелкнуть правой кнопкой мыши на сводном поле правой панели окна MAX и выбрать опцию Device Pinouts (рис. 1.9).



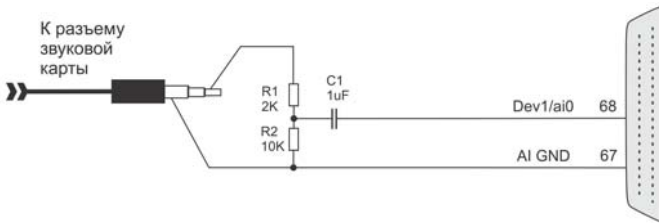
**Рис. 1.9**

После этого появится окно, в котором будут показаны выводы внешнего разъема модуля DAQ и соответствующие им сигналы.

## Задания

Следующая тема, которую мы обсудим, касается концепции заданий (tasks) – полностью определенных и сконфигурированных процедур обработки физических сигналов. Программа MAX позволяет создавать «задания» на обработку сигналов, которые могут запускаться самостоятельно в самом приложении MAX или же служить готовыми «строительными блоками» при разработке виртуальных инструментов.

Объяснить концепцию «задания» лучше всего на реальном примере. В следующем примере показано, как выполнить чтение аналогового сигнала, поступающего с линейного выхода звуковой карты персонального компьютера. Аппаратная часть задания для измерения параметров сигнала на линейном выходе звуковой карты показана на рис. 1.10.



**Рис. 1.10**

В этой схеме для измерения используется выходной сигнал одного канала звуковой карты. Измеряемый сигнал поступает на вход канала ai0 аналого-цифрового преобразователя устройства DAQ с делителя напряжения на резисторах R1R2. Для передачи только переменной составляющей аудиосигнала используется разделительный конденсатор C1. Общий провод звукового сигнала соединяется с «аналоговой землей» устройства DAQ.

Для выполнения процедуры измерения аналогового сигнала с выхода звуковой карты создадим задание (task). Выбрав пункт Data Neighborhood, щелкнем на нем правой кнопкой мыши и выберем опцию Create New... (рис. 1.11).

В раскрывшемся окне мастер проектов предложит создать один из проектов, представленных в списке (рис. 1.12). Поскольку мы создаем новое задание, то в этом списке следует выбрать пункт NI-DAQmx Task, затем перейти к следующему шагу, нажав кнопку Next.

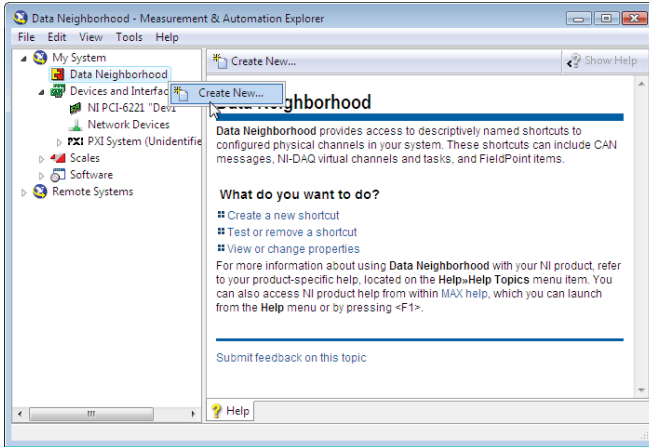


Рис. 1.11

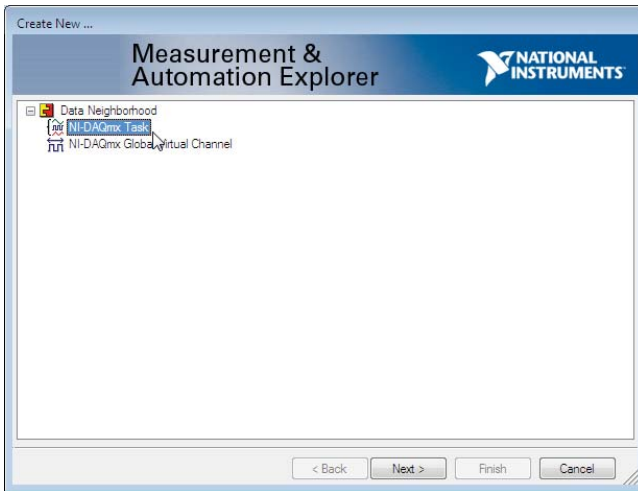


Рис. 1.12

В следующем окне нам будет предложено выбрать тип измерения для нашего задания (рис. 1.13). Нас интересует измерение временных параметров непрерывного сигнала с аудиовыхода звуковой карты, поэтому из раскрывающихся списков следует последовательно выбрать **Acquire Signals** → **Analog Input** → **Voltage**, затем продолжить конфигурирование задания, нажав кнопку **Next**.



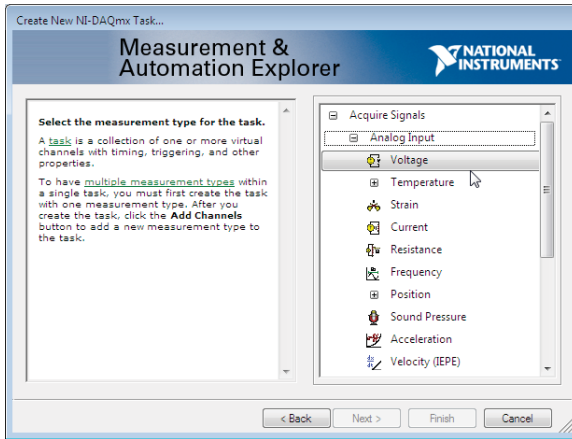


Рис. 1.13

В следующем окне (рис. 1.14) нужно выбрать физический канал аналогового ввода для приема сигнала. Для устройства NI PCI-6221 выберем канал ai0.

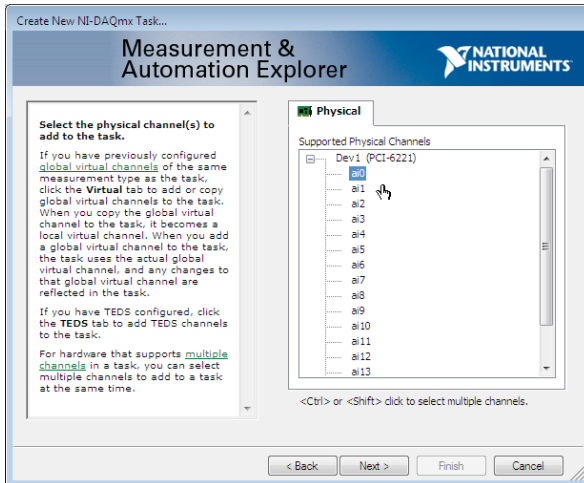


Рис. 1.14

На следующем шаге мастер проектов предложит ввести имя задания (рис. 1.15). В качестве имени по контексту задания выберем AudioInput и нажмем кнопку Finish – на этом создание задания завершается.

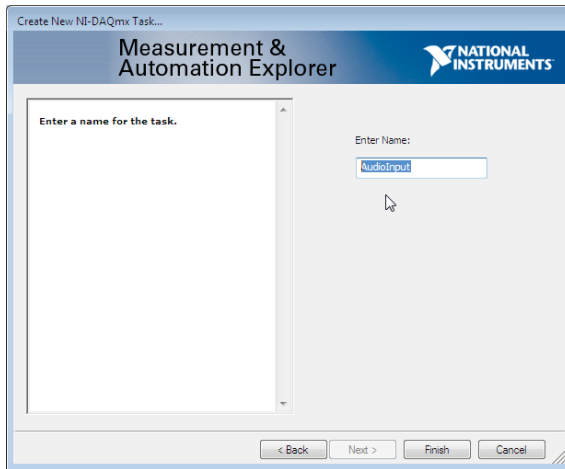


Рис. 1.15

В следующем окне (рис. 1.16), которое появляется сразу же по завершению создания задания, будут показаны различные параметры задания, которые мы можем сконфигурировать по своему усмотрению.

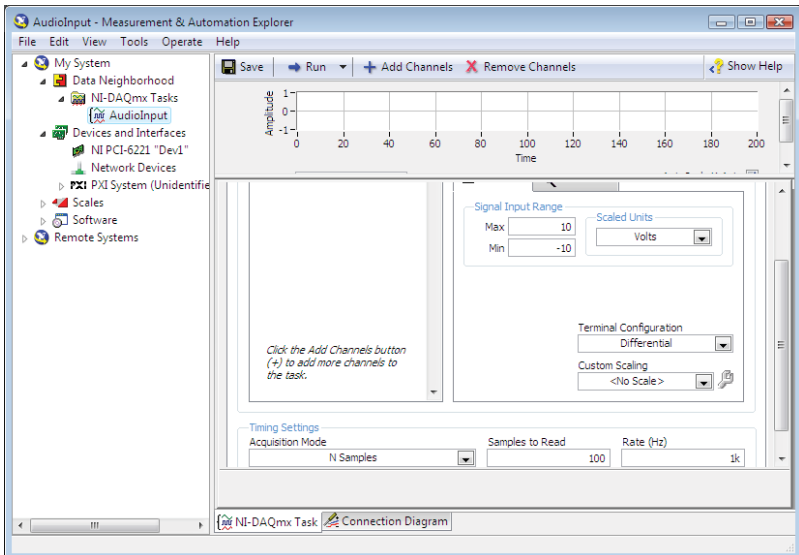
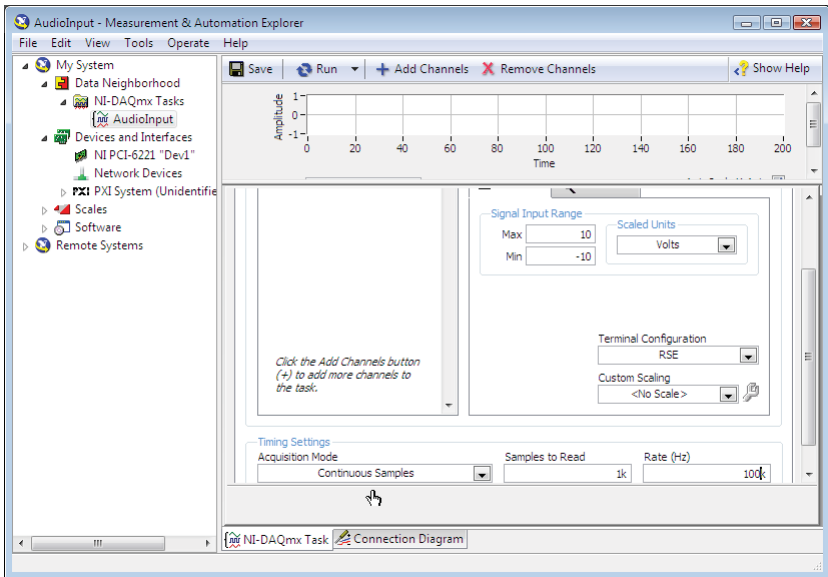


Рис. 1.16

Нам следует изменить некоторые параметры задания, чтобы адаптировать процедуру измерения к нашему конкретному случаю. Во-первых, поскольку сигнал, поступающий с линейного выхода звуковой карты, является однопроводным, то в поле параметра Terminal Configuration следует установить значение RSE. Далее, следует выбрать более высокую частоту дискретизации (Rate(Hz)) вместо значения по умолчанию, равного 1000 выборок/с. Установим новое значение частоты дискретизации, равное 100 000 выборок/с – этого будет более чем достаточно для того, чтобы перекрыть диапазон аудиочастот.

Количество выборок (Samples to Read) также увеличим до 1000. Наконец, изменим режим измерения (Acquisition Mode), выбрав непрерывный (Continuous Samples). После внесения необходимых изменений окно конфигурации нашего задания будет выглядеть так, как показано на рис. 1.17.

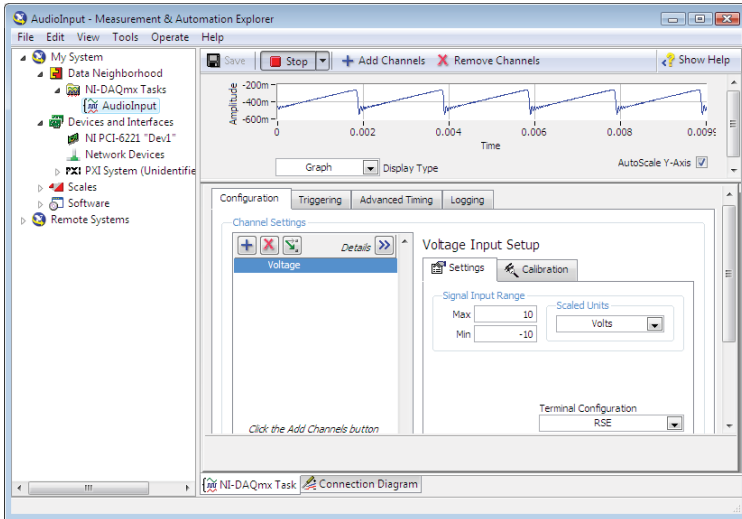


**Рис. 1.17**

Для тестирования нашего задания нужно сгенерировать какой-либо аудиосигнал (для наглядности, лучше периодической формы) с помощью одного из популярных тон-генераторов (в данном проекте используется популярная программа Audacity). Запуск задания

осуществляется при нажатии кнопки Run – при этом на графике, расположенном ниже меню, будет отображаться амплитудная характеристика измеренного сигнала.

Вот так будет выглядеть окно задания при измерении пилообразного напряжения, сгенерированного программой Audacity (рис. 1.18).



**Рис. 1.18**

Для остановки выполняющегося задания следует нажать кнопку Stop. Задание также можно сохранить для последующих измерений, нажав кнопку Save.

Задания представляют собой целиком автономные процедуры, которые позволяют выполнить то или иное измерение или сгенерировать сигнал. Тем не менее, их очень удобно использовать при разработке виртуальных инструментов. Компонент DAQ Assistant, который мы будем часто использовать в последующих проектах, позволяет задавать имя задания в качестве одного из входных параметров. Имея набор заданий, можно быстро создавать виртуальные инструменты с широким набором функций.

Следует сказать, что задания могут формироваться динамически, когда в проект включается компонент DAQ Assistant – при этом будет выполнена только часть шагов, начиная с выбора типа измерения. Дальнейшее конфигурирование задания в DAQ Assistant не будет ничем отличаться от только что описанного.

Определение и конфигурирование задания является одним из этапов создания приложения в NI Measurement Studio. При этом мастер проектов позволяет либо использовать ранее созданное с помощью утилиты MAX задание (например, как в только что изложенном примере), либо создать задание динамически («на лету») и включить его в состав проекта приложения. В последующих разработках проектов Measurement Studio мы будем использовать, как правило, динамические задания.

Перейдем к созданию нашего первого приложения автоматизации с помощью инструментальной среды NI Measurement Studio.

## Наше первое приложение в среде Measurement Studio

Перед началом работы предполагаем, что:

1. Драйверы аппаратного интерфейса DAQ установлены.
2. Программный пакет NI Measurement Studio 2010 успешно проинсталлирован в среде MS Visual Studio.

При запуске среды разработки MS Visual Studio в общем меню должна присутствовать опция Measurement Studio (рис. 1.19).

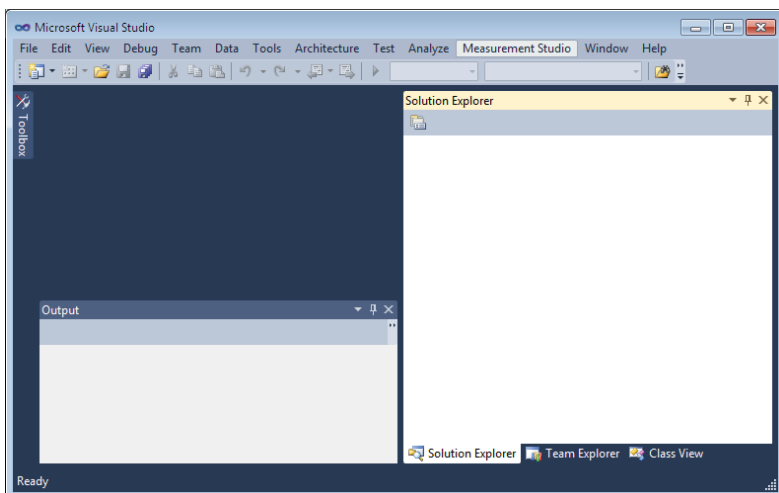


Рис. 1.19

При выборе этого пункта меню в выпадающем окне появляются опции для настройки и конфигурирования приложений Measurement Studio, а также опции для настройки самой программной среды. В нашем первом приложении мы оставим все настройки приложения установленными по умолчанию и перейдем к созданию нашего первого приложения автоматизации.

Наше приложение будет считывать сигнал от внешнего источника, поданный на физический канал 2 модуля DAQ. Аппаратная часть нашего приложения показана на рис. 1.20.

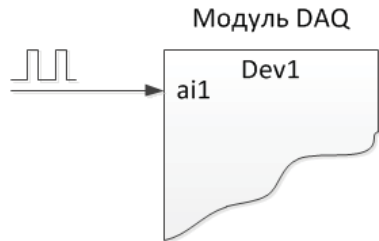


Рис. 1.20

По этой схеме прямоугольный сигнал от внешнего источника поступает на вход аналогового канала 2, обозначенного как ai1 (фирма National Instruments, как и остальные производители подобного оборудования, устанавливает нумерацию каналов, начиная с 0).

Для создания приложения Measurement Studio мы воспользуемся возможностями мастера проектов среды Visual Studio. На первом шаге нужно в меню File выбрать опцию **New...** → **Project**. В раскрывшемся окне из списка шаблонов Visual C# следует выбрать Measurement Studio и далее тип проекта NI DAQ Windows Application (рис. 1.21).

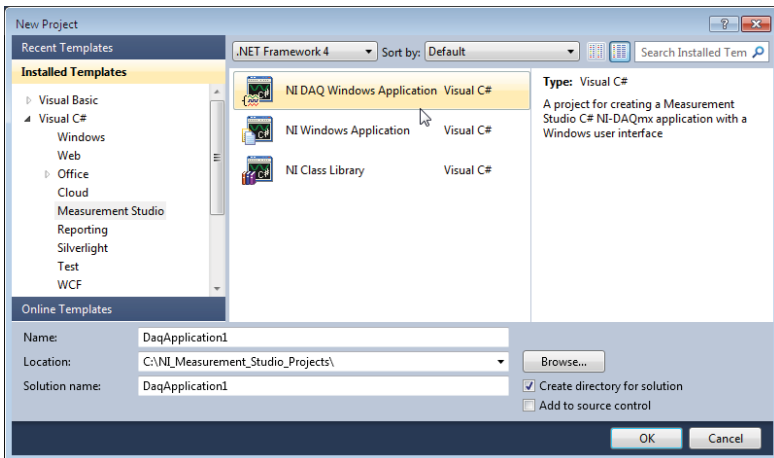
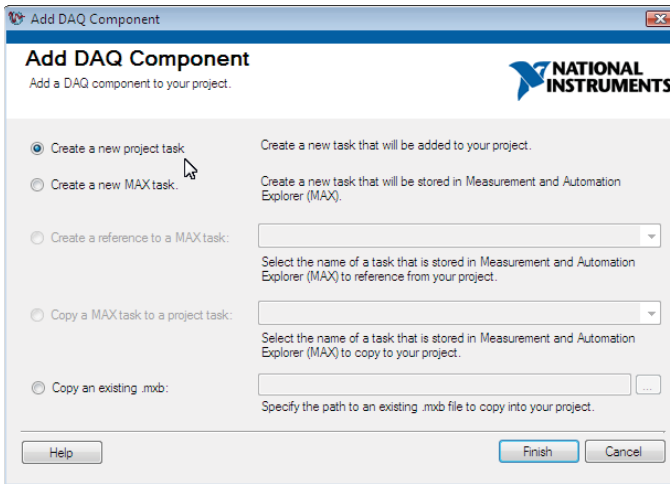


Рис. 1.21

Данный тип проекта наиболее прост в реализации и требует от разработчика минимального количества шагов, поскольку мастер проектов выполнит как конфигурирование аппаратно-программного интерфейса, так и генерацию главной формы приложения с визуальными компонентами. По умолчанию мастер проектов предлагает имя DaqApplication1 для данного проекта, с чем мы согласимся и перейдем к следующему шагу после нажатия кнопки ОК.

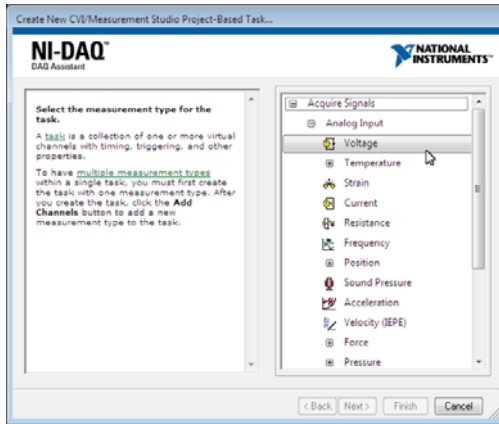


**Рис. 1.22**

На этом шаге мастер проектов предложит нам сконфигурировать *задание* для приложения. Вспомним, что задание представляет собой программный интерфейс высокого уровня между приложением пользователя и измерительным модулем. На этом шаге можно использовать одну из альтернатив: либо сконфигурировать задание непосредственно на этапе разработки проекта, либо использовать уже имеющееся задание, созданное ранее с помощью утилиты MAX. По умолчанию мастер проектов предлагает создать задание «на лету», чем мы и воспользуемся. После нажатия кнопки Finish выполняется переход к следующему этапу, на котором нужно сконфигурировать новое задание (рис. 1.23).

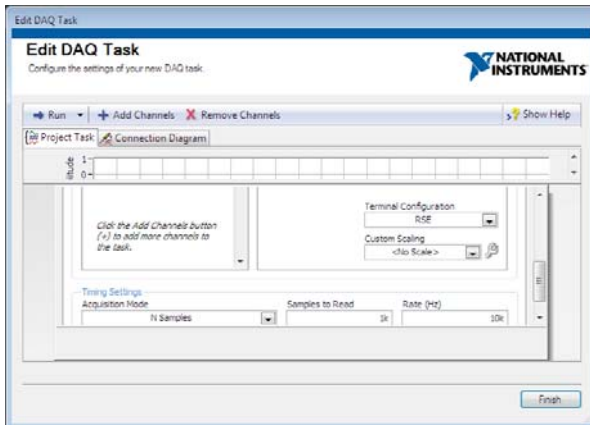
Как видно из рисунка, в процессе конфигурирования задания мастер проектов обращается к функциям приложения MAX. Поскольку наше приложение должно выполнять измерение непрерывного сигнала, то в правой панели окна DAQ Assistant следует выбрать опцию

**Acquire Signals** → **Analog Input** → **Voltage**. Все последующие шаги будут идентичны тем, которые были выполнены при настройке задания с помощью MAX ранее в этой главе.



**Рис. 1.23**

В окончательной конфигурации наше задание должно выполнять захват данных на входе физического канала ai1, настроенного на обработку однополярных сигналов (параметр RSE в настройках канала). Частоту дискретизации (sample rate) нашего канала выберем равной 10К, а количество выборок равным 1000. Все эти настройки показаны на рис. 1.24.



**Рис. 1.24**