

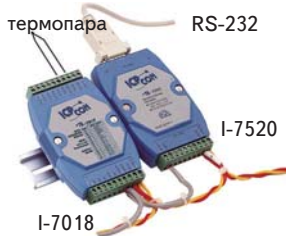
Уроки по LabVIEW

На этом уроке мы продолжаем работать с "железом". Вам предлагается построить простейшую систему сбора данных на основе популярной серии модулей i-7000.

Вы познакомитесь с программой конфигурации 7000 Utility и создадите в LabVIEW приложение, использующее функции работы с COM-портом и локальные переменные, а также создадите простейший "диалог", используя стандартные приемы, реализованные в самом LabVIEW.



Модули серии i-7000 производства компании ICP_DAS в особом представлении не нуждаются. Они использованы в сотнях реализованных проектов на многих предприятиях, в НИИ и ВУЗах Украины, и зарекомендовали себя с самой лучшей стороны. Давайте

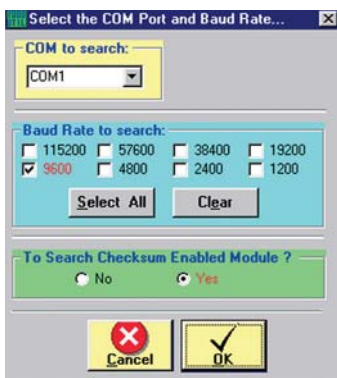


подключим один модуль АЦП для термпар i-7018 к коммуникационному порту компьютера. Модули сбора данных i-7000 имеют интерфейс RS-485, поэтому нам потребуется преобразователь интерфейса RS232/RS485, например, модуль i-7520. А ко

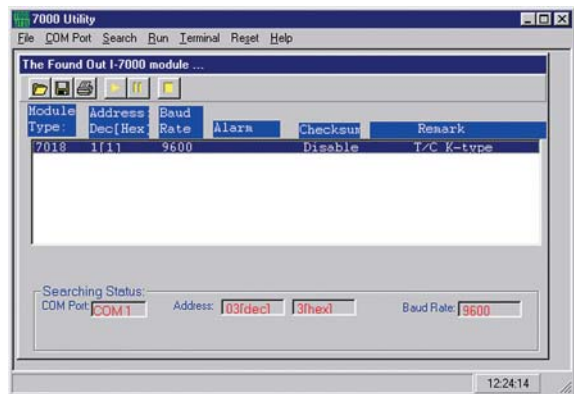
входам первого канала **Vin1+** и **Vin1-** модуля i-7018 подключим термопару. И естественно следует обеспечить питание модулей - будем использовать источник +24В фирмы MeanWell DR45-24.

Но прежде следует сконфигурировать модуль i-7018, т.е. задать адрес и скорость обмена данными в сети RS485, а также некоторые другие параметры, особенные для конкретного модуля. В соответствии с руководством пользователя переведем модуль в режим инициализации и установим необходимое программное обеспечение из двух каталогов **7000Util**, **Nap7000v**, которое находится на компакт диске ICP_DAS. Для этого запускаем из обоих каталогов файлы **setup.exe**. В первой из двух созданных директорий будет находиться утилита для конфигурирования модулей, а во второй будет создано несколько поддиректорий с драйверами (**DLL** и **LLB**-библиотеками), примерами программ и соответствующей документацией.

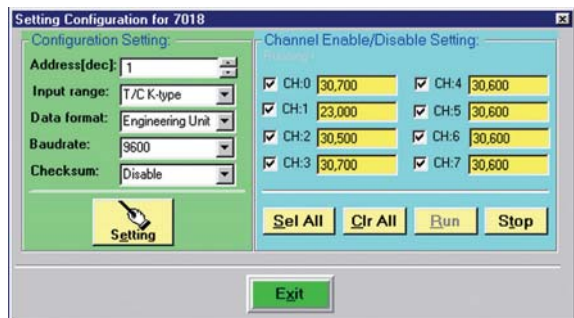
Запустите программу **7000util.exe**. Выберите в меню **COM Port** и в появившемся окне установите параметры так, как показано ниже (если, конечно, Вы не подключили модули ко второму **COM**-порту). Подтвердите свой выбор.



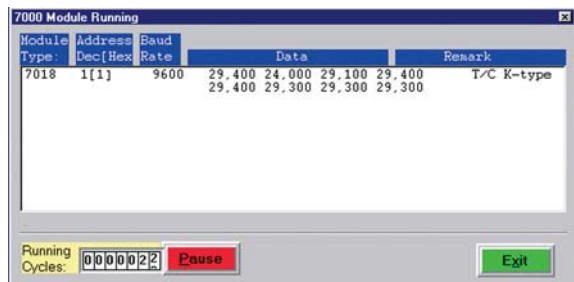
В основном окне утилиты запускаем режим автоматического поиска и не сомневайтесь, если конечно Вы все сделали правильно, что программа определит - к **COM**-порту подключен именно модуль i-7018.



Далее следует дважды щелкнуть левой клавишей мыши на выделенном синей полосой модуле i-7018. Появится окно для установки необходимых дополнительных параметров:



Установите тип термопары в окне **Input Range** и введите остальные параметры, после чего нажмите **Setting** для подтверждения установок. Затем выберите пункт меню **Run**. Появится следующее окно:



В поле **Data** мы видим восемь значений, поскольку модуль имеет 8 входных каналов. Так как термопара подключена к первому каналу (каналы пронумерованы "0", "1", .. "7"), то значение, соответствующее нашему пре-

образованию равно 24.000 °C, т.е. в помещении жарковато и надо бы включить кондиционер. Проредав процедуру конфигурирования модуля, необходимо сохранить файл, содержащий выполненные настройки, выключить питание и перевести модуль в основной режим работы.

Если в системе имеется несколько модулей В/В серии i-7000, то описанную процедуру инициализации следует проделать с каждым модулем в отдельности. Не забудьте, что скорость обмена следует установить одинаковую для всех модулей, а адреса, естественно, разные. И только после этого можно включать модули в единую систему.

Давайте напишем программу в LabVIEW, которая будет осуществлять ввод измеряемой температуры и отображать ее мгновенные значения на индикаторе в виде термометра, а также строить график зависимости температуры от времени.



В этой программе будут использоваться два новых интерфейсных элемента: элемент управления **(Controls) » Dialog Controls » Dialog Tab Control** и элемент индикации в виде термометра **(Controls) » Numeric » Thermometer**.

Создайте новое приложение. Установите на интерфейсную панель **Dialog Tab Control**. Переименуйте закладки так, как это показано ниже. Когда активной является первая закладка, установите два элемента декорации **Controls » Decorations » Thick Lowered Box**, где в дальнейшем будут размещены элементы управления и индикации для работы с последовательным портом.

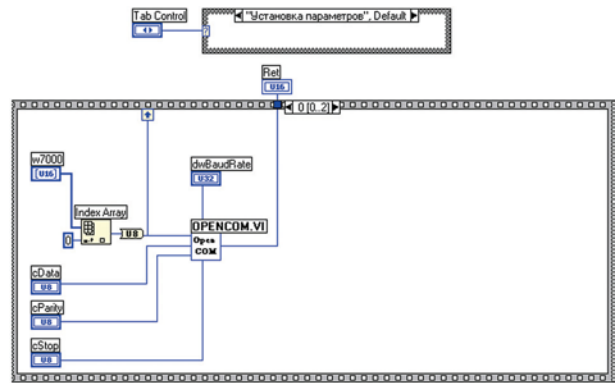
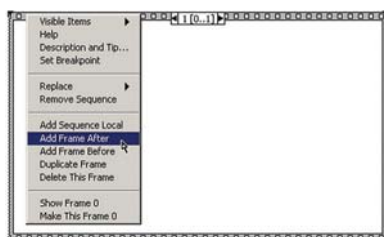


Далее следует переключиться в окно редактирования диаграмм и для Tab-элемента создать пустую CASE-структуру.

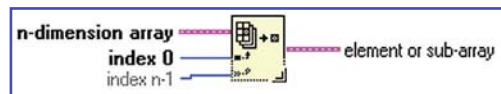
Устанавливаем структуру "Последовательность" (покадровое выполнение фрагмента программы) и создаем три кадра - 0..2.

В нулевой кадр помещаем иконку функцию для "открытия" (конфигурирования) последовательного канала. Она по умолчанию расположена в папке **C:\DAQPro\NAP7000V\7000LV\Uart**.

Подводя указатель мыши в виде катушки с проводом ко входам и выходам установленной функции, создаем элементы управления и индикаторы в соответствии с диаграммой и интерфейсной частью программы для первой закладки. При этом для создания массива используем дополнительные функции.

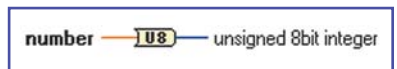


Функция **Index Array** позволяет осуществить выборку скалярного элемента из **n**-мерного массива, **n-dimension array** – входной **n**-мерный массив, **index 0... index n-1** – индексы элемента **n**-мерного массива, а функция **I8** преоб-

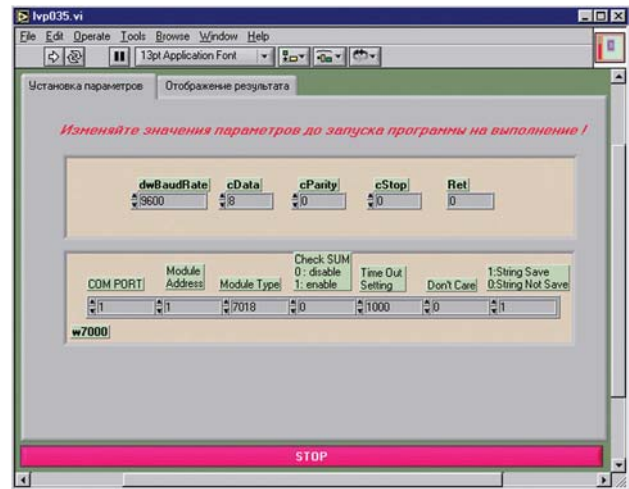


разует входное число в 8-битное целое в диапазоне от -127 до +128.

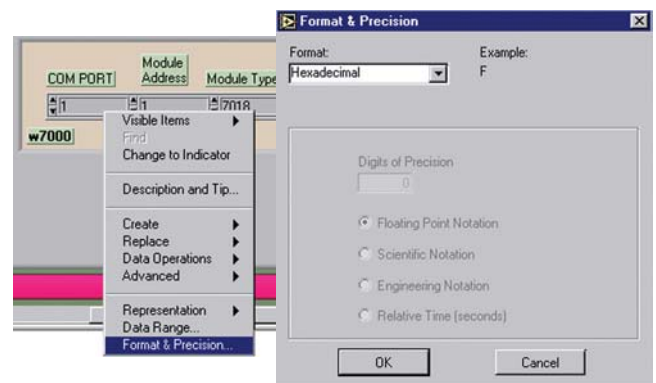
Подпишите все элементы в соответствии с рисунком.



Создав массив элементов конфигурирования **w7000**, измените тип представления данных его элемен-

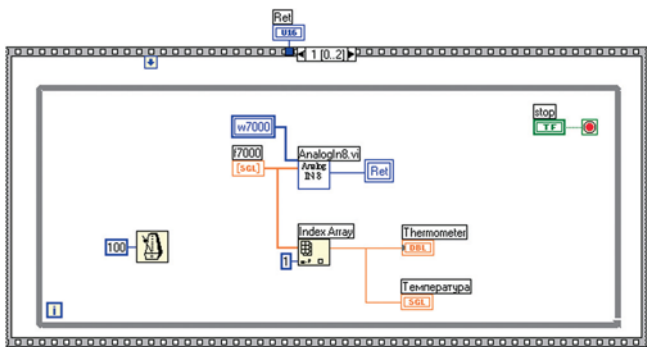


тов, преобразовав его из десятичного в шестнадцатеричный. Для этого нужно перейти в интерфейсную часть программы, подвести указатель мыши к одному из элементов массива, и нажать правую кнопку.



В появившемся меню выберите пункт, определяющий формат и точность данных (**Format & Precision**). Появилось окно, где и следует установить шестнадцатеричную систему представления данных.

Переходим в окно редактирования диаграмм и включаем первый кадр. Устанавливаем функцию для ввода аналогового сигнала для 8-и каналов и переходим в интерфейсную часть программы.



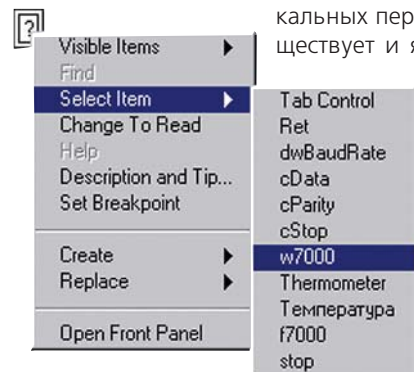
Переключаемся в окно, соответствующее второй закладке. Размещаем термометр и графический индикатор для поточечного построения зависимости. Создаем в интерфейсном окне или в окне редактирования диаграмм кнопку "STOP" для завершения работы с программой. Она же будет определять выход из цикла.

И, наконец, создадим элемент индикации (массив) **f7000**. Он находится слева от иконки. Поскольку он соответствует

получаемому результату ввода аналоговых сигналов, а не является элементом индикации, его следует сделать невидимым в интерфейсной части программы. Для этого необходимо в окне редактирования диаграмм подвести указатель мыши к соответствующей иконке и нажать правую клавишу мыши. В появившемся окне следует выбрать опцию, которая "прячет" индикатор - **Hide Indicator**.

Как и в любом другом языке программирования, в LabVIEW существуют как глобальные, так и локальные переменные. Первые используются для передачи данных между различными виртуальными инструментами (программами, оформленными в виде отдельных модулей), а вторые - для передачи внутри текущей программы. На последнем типе остановимся более подробно.

В LabVIEW передача значений осуществляется благодаря связям, которые как раз и выступают в роли локальных переменных. Однако существует и явное представление

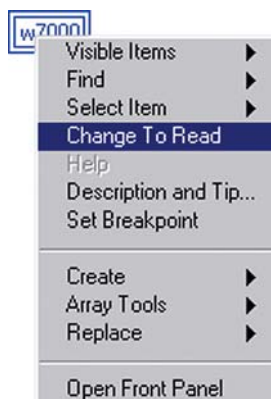


локальной переменной в виде отдельной иконки (терминала) окна редактирования диаграмм. Можно, как записывать данные в переменную, так и читать их из нее.

Создадим локальную переменную для параметра **w7000**.

Находясь в окне редактирования диаграмм, создайте новую локальную переменную. Для этого необходимо выбрать (**Functions**) » **Structures** » **Local Variable**.

Далее подводим указатель мыши ко вновь созданной иконке и нажимаем правую клавишу мыши. Выбираем необходимый элемент, к которому будет "привязана" эта локальная переменная.



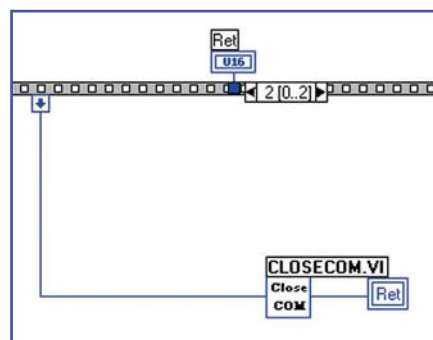
Теперь необходимо "сказать" переменной, что она "будет читать" данные - **Change To Read**.

И следующий шаг - создание локальной переменной для возвращаемого параметра **Ret**, в которую будут записываться данные.

Создаем локальную переменную на запись для возвращаемого параметра **Ret**.

Необходимо также выделить для массива его первый элемент, который соответствует значению температуры, измеряемой с помощью термометра, подключенной к первому каналу. Выполняем все необходимые соединения.

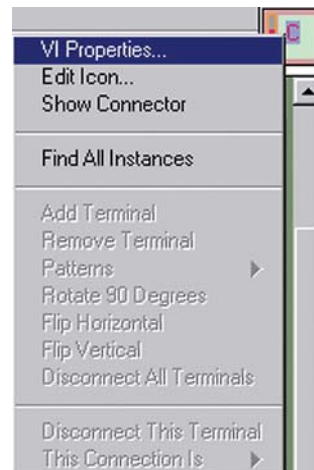
Согласно руководству пользователя для модуля i-7018, в режиме одноканального ввода выполняется 10 измерений в секунду. Исходя из этого и устанавливаем задержку в цикле, равную 100 мс.

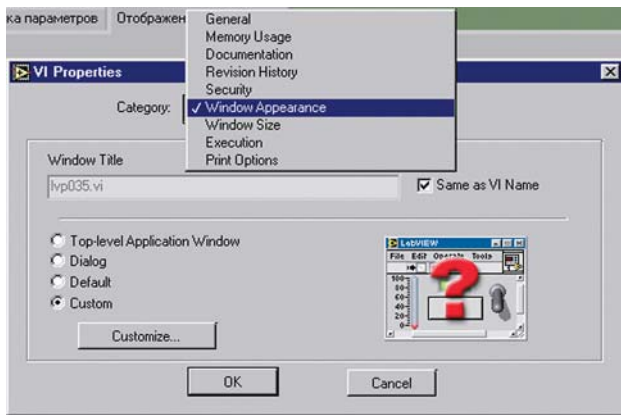


Переключаемся во второй, последний кадр. В нем устанавливаем соответственно функцию для завершения работы с последовательным портом и еще одну локальную переменную.

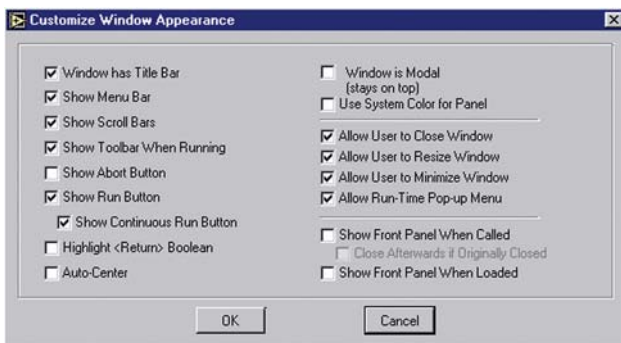
Для того чтобы завершение работы с последовательным портом проходило корректно, необходимо "убрать" кнопку останова с панели специальных клавиш. Для этого необходимо нажать на правую клавишу мыши, когда курсор находится на иконке программы, и там выбрать **VI Properties...** (Свойства виртуального инструмента).

В появившемся окне в разделе **Category** необходимо установить **Window Appearance**.



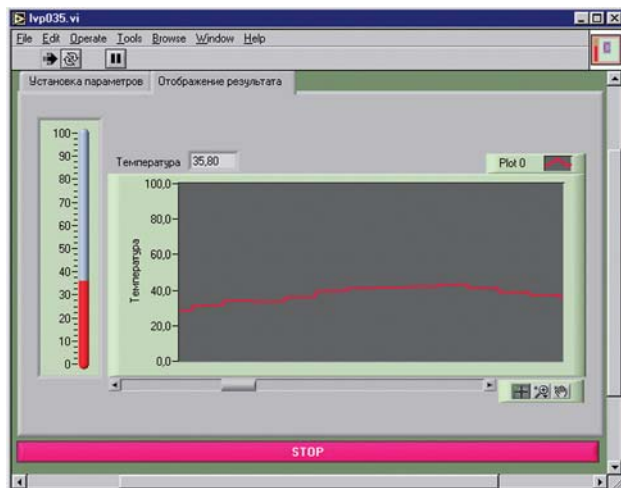


Далее, следует выбрать выборочное конфигурирование **Custom** и нажать **Customize...**



В новом появившемся окне уберите галочку, соответствующую **Show Abort Button**, и подтвердите свой выбор. Вот, кажется, и все.

Остается только запустить программу на выполнение. Возьмите термометр рукой или поднесите к ней зажигалку. А теперь можно и на экран посмотреть:

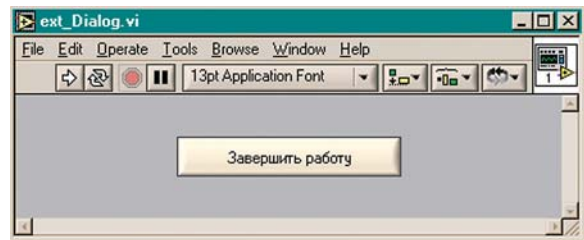


Будем надеяться, что у Вас все получилось. Иначе и быть не могло. Оказывается работать с **COM**-портом еще проще, чем с платами ввода/вывода.

Итак, успешно выполнив все примеры, рассмотренные на уроках №№ 1..6, Вы заочно прошли базовый курс LabVIEW в полном объеме и потенциально готовы к самостоятельному решению реальных задач. Возможно не все еще получается без "шпаргалок". Не отчаивайтесь! Будьте терпеливы и настойчивы в достижении поставленных целей! И удача, безусловно, придет!

Но уроки по LabVIEW на этом не заканчиваются, ведь LabVIEW - неисчерпаем. От "базового" плавно так переходим к "расширенному" курсу. И для начала - стандартные диалоговые окна. Рассмотренный пример подскажет Вам, как в дальнейшем при написании законченных программных решений можно использовать стандартные приемы, реализованные в самой среде разработки LabVIEW.

Давайте создадим "диалог", который будет выполнять запрос о завершении работы с программой.



Открываем новый проект и устанавливаем на интерфейсную панель единственную кнопку. Нажимая на нее, будет "выпадать" диалоговое окошко с запросом о завершении работы.

Для работы с диалогами в LabVIEW реализованы две функции: **One Button Dialog** и **Two Button Dialog**, которые находятся в **Functions** » **Time & Dialog**.

В примере мы будем использовать окно с двумя кнопками: подтверждения запроса и его отмены. Создадим диаграмму, как показано на приведенных ниже рисунках:

Теперь остается запустить программу на выполнение и убедиться в ее работоспособности.

Не трудно представить, насколько широким является круг задач, где может быть использован этот небольшой и простой, но в то же время очень полезный, алгоритм.

