

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ
УНИВЕРСИТЕТ «МИФИ»

М.П. Бежко, И.Ю. Безотосный

**ПРИМЕНЕНИЕ ЭВМ
В ЭКСПЕРИМЕНТАЛЬНЫХ
ИССЛЕДОВАНИЯХ**

Лабораторный практикум

ЧАСТЬ 1

*Рекомендовано УМО "Ядерные физика и технологии"
в качестве учебного пособия
для студентов высших учебных заведений*

Москва 2011

УДК 53.08:004.4(076.5)

ББК 22.3ся7

Б 38

Бежко М.П., Безотосный И.Ю. **ПРИМЕНЕНИЕ ЭВМ В ЭКСПЕРИМЕНТАЛЬНЫХ ИССЛЕДОВАНИЯХ** Лабораторный практикум. **Ч.1.** – М.: НИЯУ МИФИ, 2011. 76 с.

Изложены основы автоматизации эксперимента в области физики сверхпроводимости и полупроводниковых гетероструктур с использованием измерительных систем и среды программирования Agilent Technologies.

Практикум предназначен для студентов, обучающихся по специальности Физика конденсированного состояния вещества и специализирующихся по кафедре "Моделирования и физики наноструктур и сверхпроводников", а также может быть рекомендован для слушателей групп повышения квалификации специалистов в области высокотемпературной сверхпроводимости и физики наноструктур.

Подготовлено в рамках Программы создания и развития НИЯУ МИФИ.

Рецензент Руднев И.А.

ISBN 978-5-7262-1526-6

© *Национальный исследовательский ядерный университет «МИФИ», 2011*

ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ

В ходе подготовки к лабораторной работе следует:

- Внимательно прочитать описание работы (при необходимости воспользоваться списком рекомендованной литературы).
- Разобраться в порядке выполнения лабораторной работы и схеме измерения. Составить основную часть программы, необходимой в работе.
- Ответить на все контрольные вопросы, предлагаемые в работе.
- Подготовить краткое (1-2 с.) описание лабораторной работы с необходимыми схемами.
- Подготовить съемный носитель информации для записи полученных результатов.

В ходе выполнения лабораторной работе необходимо:

- Строго следовать указаниям преподавателя и инженера, ведущих лабораторную работу.
- При работе с приборами и компьютерами соблюдать технику безопасности.
- При подключении устройств и компьютера по интерфейсам GPIB, USB, LAN, RS232 убедиться, что подключаемые устройства выключены, затем произвести их коммутацию, затем убедиться, что все разъемы плотно сидят друг в друге, не выходят друг из друга под тяжестью проводов, и только затем включать всю систему вместе с компьютером. Рекомендуется закручивать специальные винты фиксации, которые имеют некоторые типы разъемов. При подключении *измерительных кабелей и схем* следует соблюдать следующие правила:

- при подключении **источников питания** - *сперва* подсоединить измерительную схему к выходу, не включая в нее образец,

затем включить источник питания, проверить, что отключены все выходы и стоят минимальные значения тока и напряжения, затем подключить кабели к образцу и *лишь затем* включать необходимый выход (канал) и устанавливать необходимые для измерений уровни тока и напряжения.

- при подключении **измерительного оборудования** – *сначала* включить прибор, удостовериться, что выбраны необходимые параметры и типы измерений (например, режим «Vac» на вольтметре) и *лишь затем* подключать к образцу измерительные кабели.

▪ Выполнить все действия, указанные в задании к работе. Действия, отмеченные знаком (*) назначаются дополнительно преподавателем, а также могут быть выполнены по желанию студента.

Оформление и сдача лабораторной работы:

➤ Распечатать указанные в задании данные и подписать их, указав Фамилию И.О. и номер группы, а также дату.

➤ Написать выводы относительно полученных результатов.

ЛАБОРАТОРНАЯ РАБОТА 1

МОДЕЛИРОВАНИЕ ЭЛЕКТРИЧЕСКИХ СИГНАЛОВ РАЗЛИЧНОЙ ФОРМЫ, МАТЕМАТИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ РЕЗУЛЬТАТОВ

Цель – изучение основных элементов и принципов работы программной среды VEE Pro; приобретение практического навыка по созданию простейшей программы.

Введение

Для создания любой автоматизированной измерительной системы необходимо составить программу, которая будет управлять процессом измерений. Среди различных программных сред, существующих на сегодняшний день, следует выделить National Instruments LabView и Agilent Technologies VEE Pro как одни из наиболее распространенных. В рамках этого курса будет рассмотрена программная среда VEE Pro 8.5. Для написания программы необходимо в первую очередь научиться пользоваться интерфейсом среды программирования и основными функциями среды программирования.

Интерфейс VEE Pro 8.5

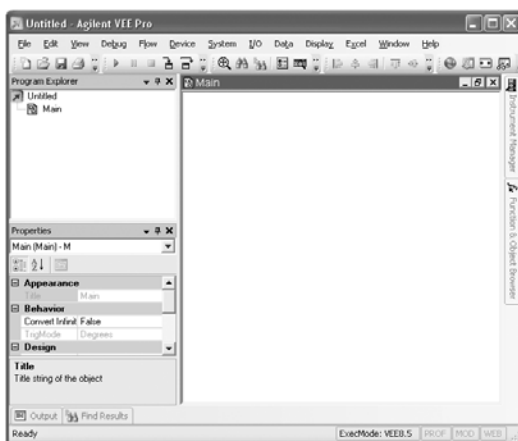


Рис. 1.1

Для подготовки лабораторных работ можно скачать полнофункциональную ознакомительную версию VEE Pro 8.5 с официального сайта Agilent Technologies (<http://www.home.agilent.com>). При запуске VEE Pro 8.5 на экране отображается окно программы рис. 1.1. Все объекты, команды и функции можно найти в главном меню сверху (File, Edit, View...). Ниже располагается панель инструментов с наиболее часто используемыми командами основного меню. Окно «Main» – основное окно, в котором и будет составляться программа. Слева вверху располагается дерево программы (Program Explorer), при помощи которого можно быстро перейти от одного объекта или функции к другому. Ниже слева располагается окно «Properties», в котором будут отображаться свойства объекта выделенного в данный момент на экране. В данный момент там отображаются свойства основного окна, которое также является объектом в программе. Справа располагаются вкладки Instrument Manager и Function & Object Browser. В первой располагается список доступных для использования в программе подключенных к компьютеру устройств. Более подробно об этом будет рассказано позднее. Во второй располагается список формул и функций, которые можно использовать в программе для различного вида преобразований данных (математических, логических, преобразования формата и т.д.).

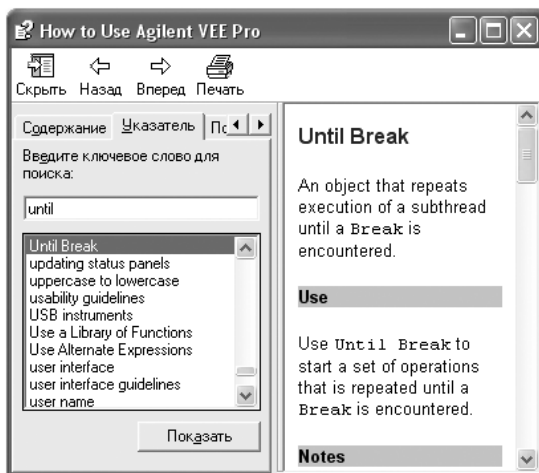


Рис. 1.2

В этой среде программирования существует руководство (How to Use Agilent VEE Pro) рис.1.2, содержащее информацию как в целом о принципах работы самой программной среды, так и об отдельных объектах. Оно написано доступным для понимания языком и рекомендуется как дополнительная литература по курсу. В нем также разобраны множество примеров для лучшего понимания материала. Для поиска информации очень удобно пользоваться «указателем». Располагается руководство в главном меню (Help → Contents and Index).

Создание простейшей программы

Рассмотрим, как необходимо создавать программу, которая будет моделировать электрические сигналы различной формы, производить с ними математические операции, предоставлять пользователю возможность изменять основные параметры и отображать результат. Разберем пример, для написания которого необходимо выполнить следующие действия:

1. Выбрать генератор сигналов (функций) из главного меню (Device→Virtual Source→Function Generator) и расположить его в свободной области основного окна.

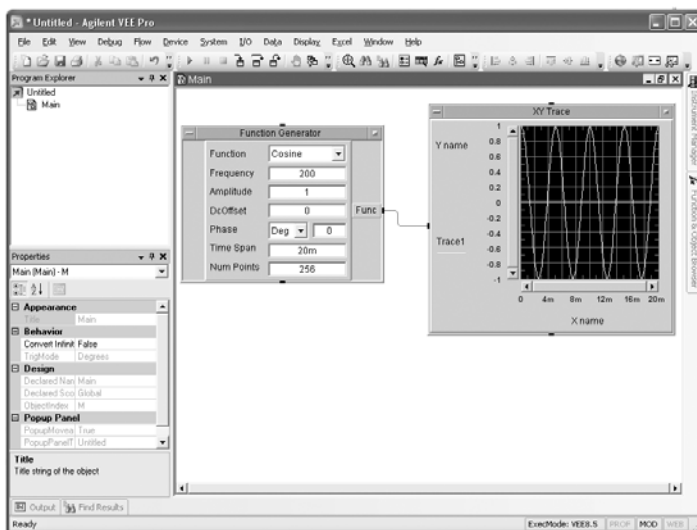


Рис. 1.3

2. Выбрать объект для отображения на экране генерируемой функции из главного меню (Display→XY Trace) и соединить между собой эти два объекта линией передачи данных. Затем запустить программу зеленой кнопкой запуска на панели инструментов или из главного меню (Debug→Run/Resume), после чего на экране появится изображение, аналогичное рис. 1.3.

3. Создать на генераторе интерфейсы для ввода частоты и амплитуды сигнала, нажав правую кнопку мыши и выбрав из меню соответствующий пункт. Затем выбрать ручки для ввода частоты и амплитуды из главного меню (Data→Continuous→Real64 Knob/Slider) и подсоединить к соответствующим терминалам. Для ручки, изменяющей частоту, установить минимальную частоту 1Гц и максимальную 10кГц, и шаг по частоте 1Гц (DetentSize в свойствах Properties). Для амплитуды оставим значения по умолчанию.

4. Создать еще один генератор и установить на нем следующие значения: форма сигнала – прямоугольник, частота «60к», амплитуда 0,1 и временная развертка (Time Span) «0,2m». Также создать формулу для перемножения двух элементов – для этого навести курсор на правую вкладку «Function & Object Browser» и выбрать Operators→Arithmetic→fx*.

5. Подключить к входным терминалам генерируемые сигналы, а результат вывести на XY Trace, предварительно отключив его от первого генератора при помощи «ножниц» (Delete line) из панели инструментов или из главного меню (Edit).

6. Установить при помощи ручки значение частоты на первом генераторе в районе 6-7 кГц, ненулевое значение амплитуды, а также проверить, чтобы значения временной развертки и количества точек (Num Points) совпадали для обоих генераторов. Выполнение последнего условия обязательно для математических преобразований, использующих одновременно два и более сигналов.

7. Запустить программу и посмотреть результат. Составленная программа будет выглядеть аналогично рис. 1.4.

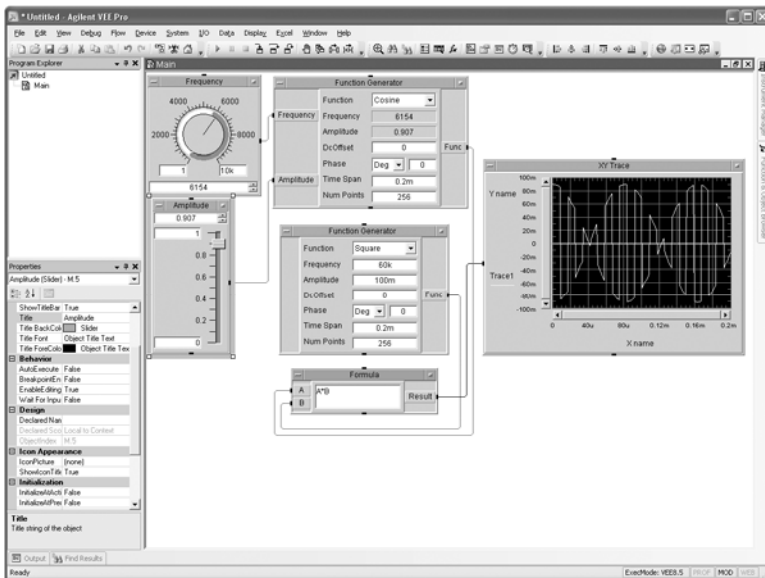


Рис. 1.4

Контрольные вопросы

1. Что такое VEE Pro 8.5?
2. Из каких основных элементов состоит интерфейс программы?
3. Как запустить справочник по программе?
4. Где находится виртуальный генератор сигналов?
5. Как запустить программу?

Задания

1. Написать программу, которая позволяет выводить на экран результат математической операции над двумя сигналами. Формы сигналов, тип математической операции и исходные параметры задаются преподавателем. Также преподавателем указываются параметры, которые должны задаваться для этих сигналов при помощи ручек (Real64).

2. Установить для ручек изменяющих частоту шаг в 1 Гц и подобрать подходящее значение временной развертки и количества

точек для генераторов в соответствии с заданной частотой таким образом, чтобы на экране XYTrace отображалось 2-3 периода конечного сигнала. Запустить программу.

3. Расположить все объекты на экране таким образом, чтобы они, а также соединяющие их линии, были четко видны. Сохранить общий вид программы после запуска при помощи кнопки PrintScreen на клавиатуре и затем распечатать.

4. (*) Дополнить программу таким образом, чтобы можно было при помощи ручки изменять количество периодов сигнала, отображаемое на экране. Пределы изменения для этой ручки должны также задаваться программой в автоматическом режиме таким образом, чтобы при выборе любого значения при помощи этой ручки не возникало ошибок в ходе выполнения программы, а на экране отображалось удобное для восприятия количество периодов.

Список литературы

1. Agilent VEE Pro Help “How To Use Agilent VEE Pro”.

ЛАБОРАТОРНАЯ РАБОТА 2

ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ VEE PRO. ОСНОВЫ ПРОГРАММНОЙ ЛОГИКИ

Цель – изучение логических элементов и принципов их работы в программной среде VEE Pro; приобретение практического навыка по созданию программы с использованием циклов.

Введение

Для того чтобы создать программу с интеллектуальными частями, которая может анализировать исходные или полученные данные и на основе этого анализа запускать ту или иную свою ветвь, необходимо научиться работать с логическими элементами. Для создания непрерывно выполняющейся части программы или программы целиком, необходимы также навыки работы с циклами.

Порядок выполнения процессов в программе. Циклы

Для начала рассмотрим подробнее, что представляют собой терминалы на любом из объектов. Практически у любого объекта есть входные и выходные терминалы последовательности запуска (рис. 2.1 Sequence In и Sequence Out). Линии, подключенные к этим терминалам, называются линиями запуска. Объект становится активным после прихода любого сигнала на входной терминал последовательности запуска (Sequence In), после того как объект завершил свою работу, он посылает запускающий сигнал с выходного терминала последовательности запуска (Sequence Out). Если к входному терминалу последовательности запуска ничего не подключено, то он запускается одновременно с запуском программы кнопкой Run/Resume. При наличии у объекта входных терминалов для управления или приема данных (рис. 2.1 Control, Data In) выполнение этого объекта не будет производиться, пока на все эти терминалы не поступят данные, т.е. после запуска объекта при помощи активации входного терминала последовательности запуска, он будет находиться в состоянии ожидания прихода данных на входные терминалы; если же все данные поступили на входные терминалы еще до активации терминала последовательности за-

пуска, то объект запустится одновременно с приходом запускающего сигнала. Исключение составляет только входной терминал XEQ (рис. 2.1). Выполнение объекта будет производиться, даже если данные не поступили, но выполнение завершится, как только данные поступят на этот терминал. При этом даже если не поступило никаких данных на входные терминалы данных и управления, но поступил сигнал на терминал XEQ, объект запустит свое выполнение и затем завершит его, пошлав данные на выходные терминалы. Это специфический терминал и встречается он только у объектов «Collector» и «Sample & Hold».

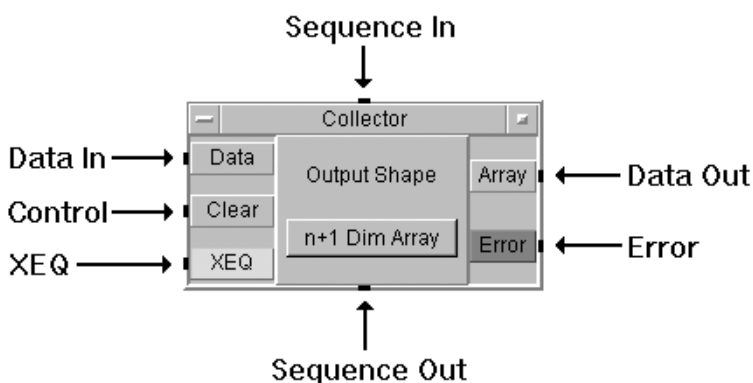


Рис. 2.1

Выходные терминалы данных и ошибки (рис. 2.1 Data Out и Error) активируются и посылают контейнер с данными сразу после завершения выполнения объекта. Выходные терминалы ошибки выдают номер ошибки в случае возникновения ошибки при выполнении объекта. Линии, по которым передаются данные, называются линиями данных и в некоторых случаях они могут использоваться и как линии запуска.

Теперь рассмотрим принцип действия цикла (рис. 2.2 For Count). После выполнения части программы, идущей перед циклом, и запуска цикла, он начинает генерировать запускающие импульсы и посылать их на свой выходной терминал. Эти импульсы запускают часть программы, находящуюся внутри цикла. Если выбран тип цикла For Count, как в нашем случае на рис. 2.2, то после выполнения заданного числа итераций цикл прекратит свою работу и

будет выполняться часть программы, следующая за циклом. В качестве выходных данных (запускающих импульсов) объект For Count посылает номер итерации.

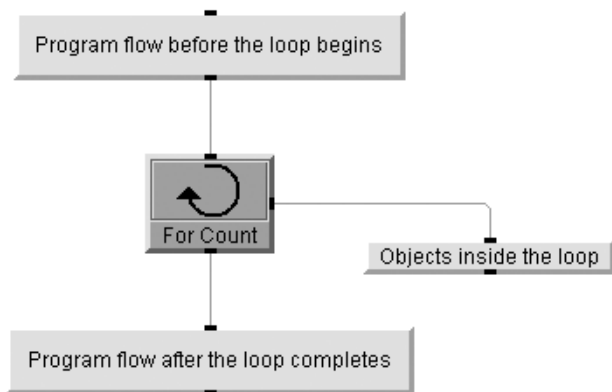


Рис. 2.2

Следует отметить, что линии запуска не всегда необходимы. Например, если от одного объекта к другому идут линии данных (рис. 2.3), то чтобы они выполнялись последовательно, линия запуска между ними не является необходимой.

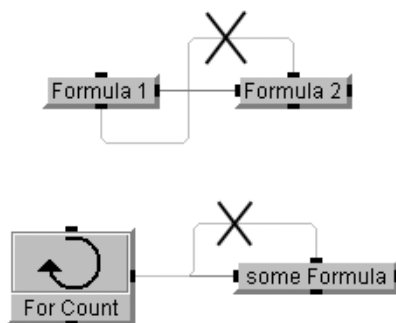


Рис. 2.3

Если от одного объекта отходит несколько параллельных ветвей (по линиям данных, или линиям запуска, или по тем и другим), то для определения конкретного порядка выполнения объектов по-

требуются дополнительные линии последовательности запуска, несмотря на присутствие линий данных.

Создание программы с использованием логических элементов и циклов

Рассмотрим создание программы, содержащей циклы и элементы программной логики на примере, для выполнения которого необходимо произвести следующие действия:

1. Пользуясь изложенным ранее материалом создать программу, отображающую сгенерированный сигнал, получить на экране изображение, аналогичное рис. 2.4.

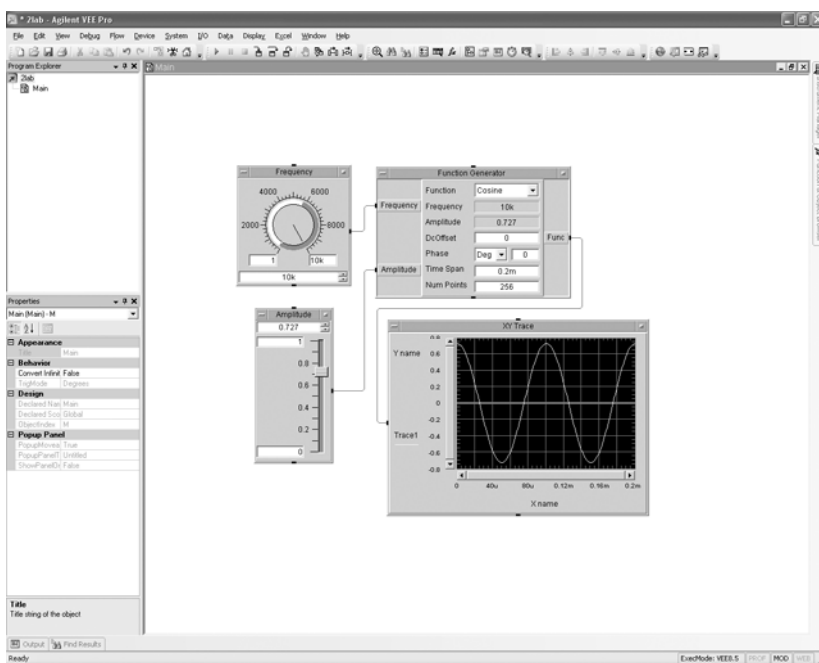


Рис. 2.4

2. Выбрать цикл **Until Break** из главного меню (**Flow**→**Repeat**→**Until Break**) и соединить его выходной терминал с входным терминалом запуска одной из ручек, изменяющих параметры генератора. Затем соединить выходной терминал запуска

этой ручки с входным терминалом запуска второй ручки, как показано на рис. 2.5. Запустить программу, и убедиться, что она работает непрерывно, и изменение параметров генератора при помощи ручек отображается на дисплее XYTrace в реальном времени.

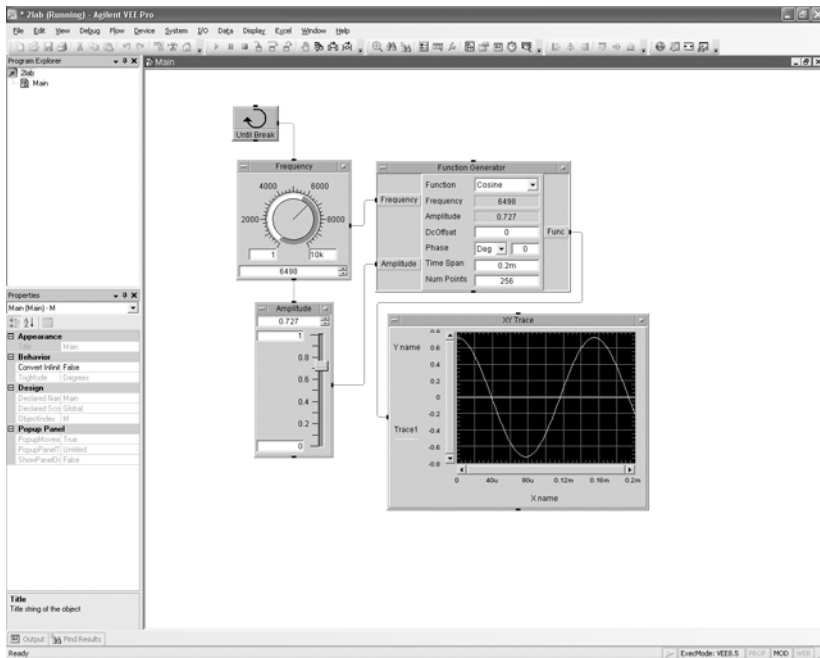


Рис. 2.5

3. Отсоединить элемент Until Break и поставить на его место цикл For Count из главного меню (Flow→Repeat→For Count). Создать входной терминал данных у объекта For Count. Выбрать постоянную Int32 из главного меню (Data→Constant→Int32) и поместить ее рядом с объектом For Count. Соединить эти два объекта линией данных и задать начальное значение постоянной, равное 100. Выбрать задержку (Delay) из главного меню (Flow→Delay) и расположить ее в самом конце ветви программы, находящейся внутри цикла For Count, соединив ее входной терминал запуска с выходным терминалом запуска последнего объекта (XYTrace). Также создать у объекта задержки входной терминал данных. Выбрать постоянную Real64 из главного меню (Data→Constant→Int32) и

разместить ее рядом с постоянной Int32. Соединить эту постоянную и задержку линией данных и задать исходное значение постоянной 0,1. Получить, после вышеописанных действий, схему программы, аналогичную изображенной на рис. 2.6. Запустить программу и убедиться что все работает без ошибок и можно в реальном времени изменять параметры генератора в течение примерно 10 секунд с момента запуска программы.

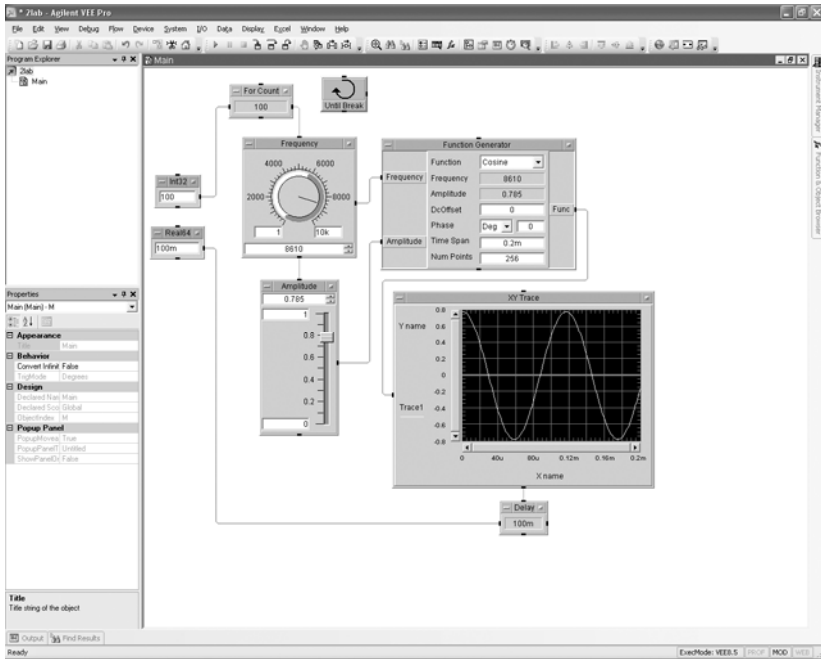


Рис. 2.6

4. Выбрать кнопку «ОК» из главного меню (Flow→Confirm(OK)) и расположить ее над постоянными. Расположить объект Until Break над кнопкой «ОК», и включить кнопку и постоянные в этот цикл, совершив действия, аналогичные описанным в пункте 2, а затем выходной терминал запуска последней постоянной соединить с входным терминалом запуска объекта For Count, также включив его в цикл Until Break. Выбрать объект Message Box из главного меню (Data→Dialog Box→Message Box) и соединить его входной терминал запуска с выходным терминалом

запуска цикла For Count. В поле Buttons выбрать «Yes No», в поле Symbol выбрать «Question», а вместо сообщения «User Message» написать «Continue?». Выбрать из главного меню объект Break (Flow→Repeat→Break), поместить его рядом с выходными терминалами объекта Message Box и соединить выходной терминал «No» с его входным терминалом запуска. После выполнения всех вышеописанных действий получить на экране схему программы, аналогичную изображенной на рисунке 2.7.

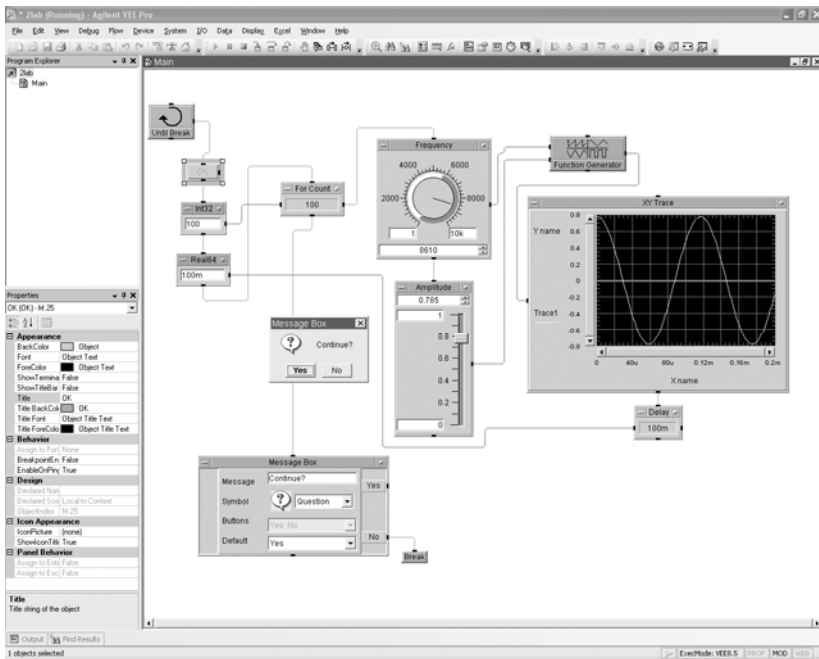


Рис. 2.7

5. Запустить программу и убедиться, что все работает без ошибок. При запуске программы можно изменить исходные параметры (константы) и затем, нажав кнопку «ОК» запустить программу дальше. В ходе выполнения программы можно изменять параметры генератора функций и видеть результат изменения через интервалы времени, задаваемые задержкой. После выполнения программы выскакивает диалоговое окно с вопросом и при поло-

жительном ответе программа запускается заново, а при отрицательном - завершает свое выполнение.

Контрольные вопросы

1. Что такое цикл?
2. Какие существуют типы терминалов у объекта?
3. Как организован порядок выполнения программы в среде VEE Pro?
4. Где можно выбрать цикл?
5. Где можно выбрать постоянные?

Задания

1. Составить программу, которая позволяет выводить на экран смоделированный при помощи генератора сигнал и в реальном времени изменять параметры генератора, такие как частота и амплитуда. Параметры временной развертки выбрать таким образом, чтобы на экране XYTrace отображалось 2-3 периода конечного сигнала. Запустить программу и проверить ее работоспособность.

2. Дополнить программу таким образом, чтобы непрерывное выполнение программы происходило в течение заданного в программе времени (допускается некоторая погрешность по времени, возникающая вследствие конечного времени выполнения части программы до объекта задержки), и это время должно вводиться после запуска программы кнопкой Run/Resume. При завершении программы должно выводиться диалоговое окно, в котором можно выбрать - запустить программу заново или завершить.

3. Сохранить программу на собственный носитель информации и распечатать экран программы, скомпоновав элементы таким образом, чтобы они были четко видны на экране.

Список литературы

1. Agilent VEE Pro Help “How To Use Agilent VEE Pro”.

ЛАБОРАТОРНАЯ РАБОТА 3

МАССИВЫ ДАННЫХ. МОДЕЛИРОВАНИЕ ЭКСПЕРИМЕНТА

Цель – изучение двумерных и одномерных массивов, принципов их работы в программной среде VEE Pro; приобретение практического навыка по созданию программы с использованием массивов данных.

Введение

Для создания любой автоматизированной системы измерения параметров или системы задания параметров в определенной последовательности необходимы навыки работы с массивами данных. Чаще всего массивы используются для записи результатов измерений или для задания параметров измерения, изначально записанных в массив.

Массивы данных в VEE Pro

В среде VEE Pro массив можно задать несколькими способами:

1. Задать его так же как постоянную, выбрав в главном меню Data→Constant→Int32 Array/Int64 Array/Real64 Array (рис.3.1) и затем задав с клавиатуры необходимые значения.

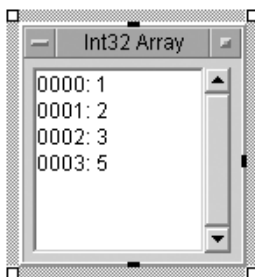


Рис. 3.1

Таким образом, мы получим одномерный массив данных с изначально заданными значениями. Такой массив полезен, если нужно

задать несколько никак не связанных между собой чисел. Большое число точек задавать в таком массиве неудобно.

2. Задать массив с необходимым числом элементов и мерностью, все элементы которого равны исходно заданному значению (Init Value), например, 0 (рис. 3.2). В одномерном случае можно задать равномерное (Lin Ramp) или логарифмическое (Log Ramp) заполнение массива от значения одного числа до значения другого (рис. 3.3).

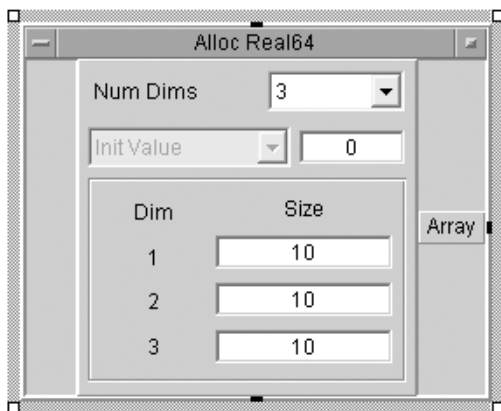


Рис. 3.2

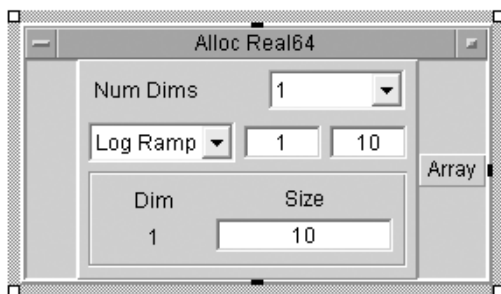


Рис. 3.3

Такие массивы очень удобны, когда мы заранее определяем количество измерений, которые будут проведены в ходе программы. В этом случае просто создаем массив, заполненный нулями, а затем записываем в него результаты измерений. Также такие массивы удобны, когда необходимо задавать на приборах какое-либо

последовательно возрастающее или убывающее значение заданное количество раз, например, постепенно возрастающее напряжение на источнике питания. В этом случае заполняем одномерный массив равномерным или логарифмическим распределением значений – от заданного минимального до заданного максимального значения. Различные типы таких массивов, в зависимости от формата данных, находятся в главном меню (Data→Allocate Array→).

3. Использовать объект Collector (рис. 3.4), расположенный в главном меню (Data→Collector). Этот объект собирает данные, поступающие на вход данных, и помещает каждый вновь пришедший контейнер с данными в новую ячейку. Он продолжает собирать данные, пока не поступит какой либо сигнал (контейнер) на вход XEQ, после чего он выдает на выходе массив с числом элементов, равным количеству контейнеров с данными, пришедшими на вход. Если форма данных на выходе (Output Shape) установлена в положение «n+1 Dim Array», то при поступлении на вход n-мерных массивов они будут последовательно записываться в ячейки коллектора, и на выходе мы получим n+1-мерный массив. Если же приходят обычные данные (0-мерные) то на выходе будет одномерный массив.

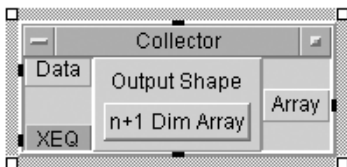


Рис. 3.4

Если форма данных на выходе установлена в положение «1Dim Array», то на выходе будет всегда одномерный массив, даже при поступлении на вход n-мерных массивов. В этом случае он преобразует n-мерный массив в одномерный.

Чтобы извлечь n-й {(n,m)-й, (n,m,k)-й...} элемент любого массива, можно воспользоваться следующей формулой из бокового меню (Function & Object Browser→Operators→Miscellaneous→fx a[2]) рис. 3.5. Для изменения n-го {(n,m)-го, (n,m,k)-го...} элемента

любого массива, можно воспользоваться другой формулой из бокового меню (Function & Object Browser → Operators → Assignment → fx $a[2]=b$) рис. 3.6. В случае, изображенном на этих рисунках, мы будем изменять 2-й элемент одномерного массива «a». Если же мы хотим изменить и узнать значение i -ого элемента, когда номер i задается вне этой формулы, то следует в поле формулы написать « $a[i]$ » (« $a[i]=b$ ») и создать входной терминал « i » у этой формулы. Также можно писать эти формулы вручную в поле объекта Formula из главного меню (Device→Formula), создавая все необходимые терминалы.

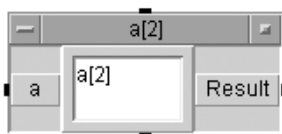


Рис. 3.5

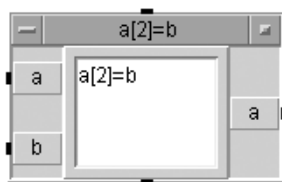


Рис. 3.6

Создание программы, моделирующей эксперимент, с использованием массивов данных

Рассмотрим создание программы, моделирующей эксперимент, в котором необходимо задать определенное число точек с определенными начальными параметрами и получить в итоге массив с экспериментальными результатами, на примере, для выполнения которого необходимо произвести следующие действия:

1. Создать цикл For Count и константу для задания количества итераций в этом цикле, пользуясь методикой, описанной в пункте 3 работы №2. Создать одномерный массив исходных значений Real64, который будет равномерно заполняться значениями от минимального до максимального. Создать на нем терминалы для за-

дания минимального (From) и максимального (Thru) значений, а также числа элементов массива (Dim Size), создать для задания максимального и минимального значений константы Real64 и подключить их к соответствующим терминалам массива. В свойствах Properties констант изменить их имена таким образом, чтобы было понятно, какая константа какому значению соответствует. Константу, задающую число итераций назвать «N Points» и подключить к терминалу массива, задающему количество элементов. Установить следующие исходные значения констант: NPoints = 100, Min = 1, Max = 10.

2. Выбрать генератор случайных чисел из бокового меню (Function & Object Browser→Built-In Functions→Probability & Statistics→fx random). Удалить входные терминалы и задать в формуле вместо «low» и «high» следующие значения: 0 и 1 соответственно. Включить генератор в цикл For Count.

3. Создать формулу для извлечения i -го элемента массива ($a[i]$) и разместить ее ниже генератора случайных чисел. В этой формуле добавить входной терминал (B) и дополнить формулу сложением с B: « $a[i]+B$ ». Затем к терминалу «a» подключить массив Real64, к терминалу «B» - выходной терминал генератора случайных чисел, к терминалу i – выходной терминал цикла For Count. Включить полученную формулу в цикл For Count, подключив ее входной терминал запуска к выходному терминалу запуска генератора.

4. Создать объект отображения сигнала XYTrace и подключить к нему выходной терминал нашей формулы. Скомпоновать объекты в программе аналогично рис. 3.7 и запустить программу. Удостовериться, что все собрано правильно и работает без ошибок, при возникновении ошибок проверить написание этой части программы и исправить ошибки.

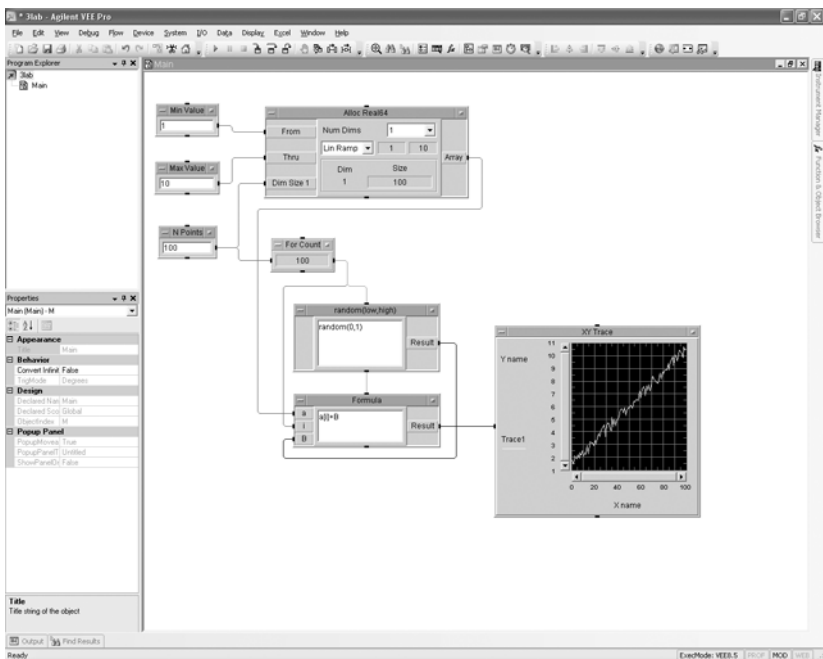


Рис. 3.7

5. Создать объект Collector и поместить его ниже формулы. Выходной терминал формулы подключить к входному терминалу данных коллектора, а выходной терминал последовательности запуска объекта For Count подсоединить к терминалу XEQ. Выбрать объект отображения AlphaNumeric из главного меню (Display→AlphaNumeric), расположить его на свободном пространстве и подключить к нему выходной терминал коллектора (Collector).

6. Аналогично пункту 4 работы №2, создать цикл для интеллектуального перезапуска программы в зависимости от выбранного варианта ответа на соответствующий вопрос, расположенный в конце программы. Создать управляющий терминал Clear для объекта XYTrace, щелкнув на объекте правой кнопкой и выбрав из меню соответствующий пункт (Add Terminal→Control Input...→Clear), и включить этот терминал в цикл Until Break, проведя линию прямо от объекта цикл.

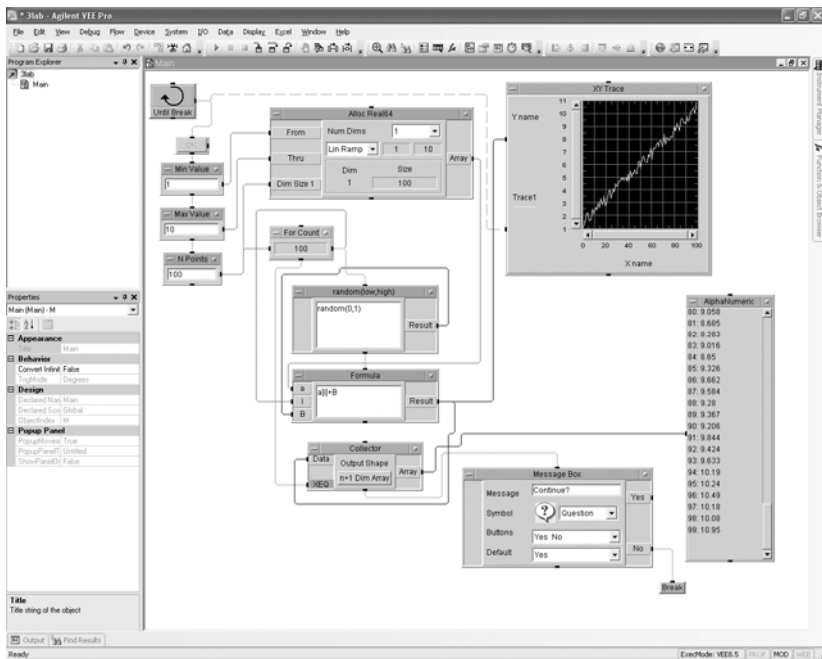


Рис. 3.8

7. Убедиться, что написанная программа аналогична программе, изображенной на рис. 3.8, и запустить ее. Убедиться, что программа работает без ошибок, проверив все функции программы. При возникновении ошибок исправить их, пользуясь вышеуказанным материалом.

Контрольные вопросы

1. Что такое массив, и какие бывают типы массивов в VEE Pro?
2. Как создать трехмерный массив?
3. Как, не выполняя дополнительных действий, заполнить одномерный массив постепенно убывающими значениями?
4. Как изменить элемент массива с номером (n,m)?
5. Как получить генерацию в «классическом» режиме?
6. Объяснить принцип работы коллектора (Collector).

Задания

1. Создать программу, которая моделировала бы получение массива данных, состоящего из заданного в программе числа экспериментальных точек с задающимися в этой же программе начальными данными, аналогично вышерассмотренному примеру.

2. Сохранить программу на собственный носитель информации и распечатать экран программы, скомпоновав элементы таким образом, чтобы они были четко видны на экране.

3. (*) Создать программу, аналогичную вышерассмотренному примеру, но на выходе которой будет двумерный массив данных, второе измерение которого должно содержать 2 элемента, первым из которых будет являться само «экспериментальное» значение, а вторым будет отклонение в процентах этого значения от исходного, заданного в программе для данной точки.

Список литературы

1. Agilent VEE Pro Help “How To Use Agilent VEE Pro”.

ЛАБОРАТОРНАЯ РАБОТА 4

КОММУТАЦИЯ ОБОРУДОВАНИЯ, УСТАНОВКА И КОНФИГУРИРОВАНИЕ ДРАЙВЕРОВ

Цель – приобретение практических навыков работы с оборудованием, имеющим интерфейсы для приема и передачи команд и данных; изучение основных интерфейсов и стандартов.

Введение

Для создания любой автоматизированной системы необходимо знать, как произвести коммутацию между компьютером и управляемыми устройствами, а также как правильно организовать работу приборов после выполнения коммутации.

Интерфейсы для подключения оборудования. Драйвера устройств

На сегодняшний день существует множество различных интерфейсов для подключения оборудования: GPIB, USB, LAN, Firewire, RS-232, VMEbus, VXI и т.д. Наиболее распространенными в современных устройствах являются GPIB, USB и LAN. При коммутации устройств по управляющим интерфейсам необходимо всегда проверять адреса устройств на соответствующих шинах во избежание конфликтов при обмене информацией по шине.

Для передачи данных по интерфейсам существуют различные стандарты, концепции и наборы команд и драйверов, такие как VISA, VI, SCPI, LXI, VMI и т.д. Рассмотрим те из них, которые наиболее часто используются для взаимосвязи с устройствами, подключенными при помощи GPIB, USB или LAN:

1. VISA

Virtual Instrument Software Architecture (VISA) (в переводе: Программная архитектура виртуального инструмента) – всемирно используемый стандарт операций ввода/вывода информации для тестового и измерительного оборудования. VISA является про-

мышленным стандартом, который используют многие компании, производящие тестовое и измерительное оборудование.

Этот стандарт содержит в себе набор спецификаций (определений) для коммуникации с различными ресурсами (часто, но не всегда являющимися аппаратными устройствами) через интерфейсы ввода/вывода, используемые в тестовом и измерительном оборудовании, такие как GPIB и VXI. Также он содержит некоторые спецификации для протоколов тестового и измерительного оборудования, использующих стандартные компьютерные интерфейсы ввода/вывода, такие как VXI-11 (через LAN(TCP/IP)) и USBTMC (через USB).

Библиотека VISA стандартизировала представление своей работы в нескольких механизмах повторного использования программного обеспечения, включая C API, представленный Windows DLL (visa32.dll) и по технологии Microsoft COM.

VISA был разработан компанией National Instruments для автоматизации измерений с помощью так называемых виртуальных инструментов (VI). Используется в основном в таких программных продуктах компании, как LabVIEW, LabWindows и Measurement Studio.

Ныне существующий стандарт VISA поддерживается компанией IVI Foundation (<http://www.ivifoundation.org>) и содержит множество спецификаций.

На практике VISA представляет собой набор драйверов для операций ввода/вывода с использованием команд SCPI, VXIplug&play, IVI-COM и некоторых других.

2. Virtual Instrumentation и Virtual Instruments (VI)

Virtual Instrumentation (В переводе – Виртуальное Оборудование) – концепция использования настраиваемого программного обеспечения и модульных аппаратных средств измерения, для создания определенных систем измерения, называемых виртуальными инструментами (VI).

Традиционные системы аппаратного оборудования составлены из заранее определенных аппаратных компонентов, таких как

цифровые мультиметры и осциллографы, которые являются неизменными по своему назначению, выполняемой функции и принципам работы (при условии, что в сам прибор не будут внесены изменения). Из-за такой жестко закрепленной за прибором функции эти системы менее гибкие, чем системы виртуального оборудования. Основная разница между аппаратным оборудованием и виртуальным оборудованием в том, что в последнем используется программное обеспечение, позволяющее заменить большое количество аппаратных средств (оборудования). То есть, используя эту концепцию можно заменить программным обеспечением множество сложных и дорогих аппаратных средств при помощи стандартной компьютерной техники. Например, на базе аппаратного аналого-цифрового преобразователя (АЦП) можно создать виртуальный осциллограф, анализатор спектра, вольтметр или специфический виртуальный инструмент для выполнения измерений.

Другими словами, в соответствии с этой концепцией система организуется в виде программной модели или моделей некоторого реально существующего или гипотетического прибора, причём программно реализуются не только средства управления (рукоятки, кнопки, лампочки и т. п.), но и логика работы прибора.

Связь программы с виртуальными инструментами осуществляется через интерфейсы и драйвера VISA.

3. Команды SCPI

Standard Commands for Programmable Instruments (SCPI в переводе: стандартные команды для программируемых устройств) – произносятся как «скиппи» – команды низкого уровня, которые передаются напрямую на управляемое устройство в виде строк формата ASCII. Вследствие такого принципа работы время отправки команды минимально короткое при затрате минимального количества системных ресурсов. Ответ от устройства приходит также в формате строки ASCII. В некоторых случаях, например, при передаче большого массива данных с устройства, данные могут передаваться в двоичном формате.

Этот стандарт определяет основной синтаксис, структуру команд и форматы данных для использования с любым инструментом (устройством). Он вводит универсальный тип команд, который может быть использован для управления любым устройством. Эти команды подразделяются на подгруппы. Существуют команды, стандартные для любых устройств («*IDN», «*RST» и т.д.), а также специфические команды для каждого устройства. Полный список таких команд всегда прилагается в специальном руководстве к устройству. Команды, как правило, имеют полное название и сокращенное, оба этих названия можно использовать при написании команды, например:

«SYSTEM:COMMUNICATE:SERIAL:BAUD 2400» устанавливает значение 2400 baud для интерфейса RS-232, находящегося на устройстве. Эту же команду можно написать в виде

«SYST:COMM:SER:BAUD 2400».

Для большинства команд можно посылать запрос, в ответ на который устройство выдаст установленное на данный момент значение запрашиваемого параметра. Для посылы такого запроса необходимо набрать команду, а вместо параметра ввести вопросительный знак («?»), например:

«SYSTEM:COMMUNICATE:SERIAL:BAUD?».

Изначально команды SCPI были разработаны для интерфейсов GPIB, но они могут использоваться и с другими интерфейсами – RS-232, USB, VXIbus и т.д.

4. LAN eXtensions for Instrumentation (LXI)

LAN eXtensions for Instrumentation (LXI) (в переводе Сетевые Расширения для Оборудования) – стандарт, разработанный LXI Консорциумом, определяющий протоколы коммуникации для оборудования и систем сбора данных, использующих Ethernet. Этот стандарт дополняет существующие стандарты для тестового и измерительного оборудования, такие как GPIB и PXI.

Платформа оборудования LXI позволяет объединять оборудование, имеющее интерфейсы Ethernet со всемирной сетью и использовать его в качестве компонентов автоматизированных изме-

рительных систем. LXI-совместимые системы имеют размеры и преимущества модульных систем и при этом не имеют дорогостоящих компонентов коммутации, таких как GPIB кабели, карты и адаптеры. Также LXI системы достаточно гибкие в использовании и можно легко подсоединять их компоненты к любому компьютеру или даже объединять компоненты (приборы) в одноранговую сеть (peer-to-peer) без применения дополнительных устройств. IP адрес устройства, как правило, можно задавать как на самом устройстве, так и с подключенного к нему компьютера.

Для обращения к подключаемому устройству, согласно этому стандарту, используются специальные драйвера или веб-интерфейсы. Этот стандарт также удовлетворяет стандарту VISA, который поддерживает операции ввода/вывода через протоколы TCP/IP. Соответственно, в большинстве случаев можно использовать команды SCPI для обращения к аппаратному устройству через интерфейс виртуального инструмента, созданный в системе.

Виртуальные инструменты в среде Agilent VEE

В среде Agilent VEE под виртуальным инструментом понимается не вся программа, составленная в VEE Pro (в среде LabView виртуальным инструментом называется именно вся программа), а объект, отображающийся в менеджере инструментов.

В среде Agilent VEE каждому виртуальному инструменту (ВИ) присваивается уникальный адрес VISA, который соответствует также и реальному (аппаратному) устройству. При изменении адреса VISA какого либо виртуального инструмента, этому виртуальному инструменту будет соответствовать другое аппаратное устройство, расположенное по введенному адресу.

Если мы имеем аппаратное устройство, подключенное к компьютеру одновременно через несколько интерфейсов (GPIB, USB, TCP/IP) то, при необходимости, в ходе программы можно динамически изменять VISA-адрес таким образом, чтобы обращение к прибору происходило через выбранную шину. Таким образом, будет задействован всего один виртуальный инструмент с изменяющимся адресом вместо нескольких (отдельный ВИ для каждой шины).

Например, пусть подключенному по шинам USB и GPIB аппаратному устройству соответствует некоторый виртуальный инструмент, имеющий имя «Voltmeter» и VISA-адрес «GPIB0::7::INSTR», т.е. находящийся на шине GPIB. И в ходе программы необходимо обратиться к этому аппаратному устройству по шине USB. Тогда можно изменить VISA адрес виртуального инструмента, написав в поле объекта Formula следующую команду: «IO.Voltmeter.VISAAddress="USB0::2391::1031::US432109::0::INSTR"; ». Соответственно, после выполнения этого объекта VISA адрес виртуального инструмента изменится и станет «USB0::2391::1031::US432109::0::INSTR». Адреса VISA, соответствующие аппаратному устройству на разных шинах, можно посмотреть в специальной программе «Agilent Connection Expert».

Рассмотрим немного подробнее интерфейс Agilent Connection Expert (рис. 4.1):



Рис. 4.1

Сверху располагается панель главного меню (Menu Bar), чуть ниже располагается панель инструментов (Tool Bar). Затем, чуть

ниже слева располагается панель задач (Task Guide), по центру – список устройств (Explorer Pane) и справа – свойства устройства (Properties Pane). В самом низу располагается строка статуса (Status Bar).

Подключение оборудования к компьютеру. Операции ввода/вывода в среде Agilent VEE

Рассмотрим методику подключения к компьютеру и конфигурации драйверов аппаратного и виртуального инструментов на примере подключения мультиметра через GPIB/USB контроллер (рис. 4.2) к компьютеру. Для этого необходимо выполнить следующие действия:

1. Проверить, что мультиметр отключен от сети, затем произвести коммутацию GPIB интерфейсов мультиметра и контроллера, подключаемого к компьютеру.



Рис. 4.2

2. Подключить GPIB/USB контроллер к USB интерфейсу компьютера. При необходимости установить в автоматическом режиме драйвер для этого контроллера (файл с драйвером копируется на компьютер при установке Agilent IO Libraries Suite и затем система его находит автоматически при установке). Убедиться, что на контроллере загорелся зеленый светодиод «Ready». Если этот диод не

загорелся, перегрузить систему или переустановить драйвер этого устройства, следуя рекомендациям преподавателя.

3. Включить мультиметр в сеть, затем включить его кнопкой питания. Запустить программу Agilent Connection Expert и убедиться, что подключенное устройство присутствует в списке оборудования на соответствующей шине. Если оборудование не появилось сразу, обновить конфигурацию оборудования, нажав кнопку «Refresh All», располагающуюся в самом верху списка устройств, или F5 при активном окне программы Agilent Connection Expert.

4. Выделить курсором мультиметр в списке подключенного оборудования и вызвать панель интерактивного ввода/вывода (Interactive IO) (рис. 4.3), нажав кнопку «Send commands to this instrument» на панели задач, либо выбрав эту панель из главного меню (Tools → Interactive IO).

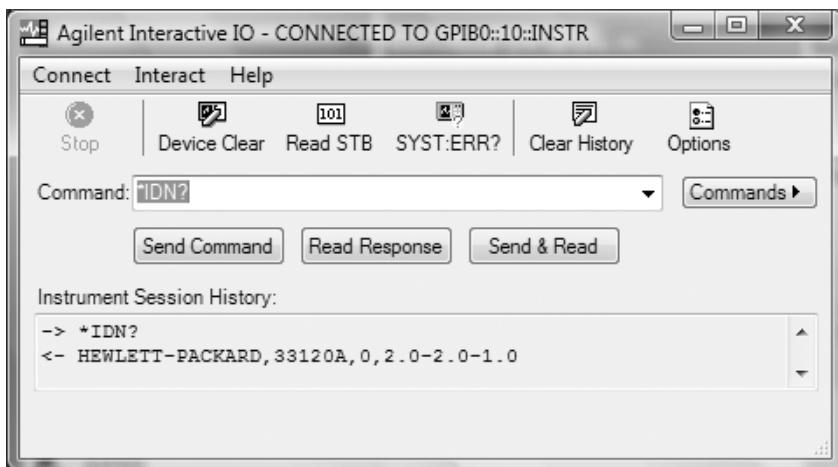


Рис. 4.3

Ввести в поле «Command» команду идентификации «*IDN?» и нажать кнопку отправки и чтения (Send & Read). Убедиться, что подключенный прибор отвечает на запрос, и закрыть программу.

5. Запустить Agilent VEE и навести курсор на боковую вкладку менеджера инструментов (Instrument Manager). В появившемся

списке отобразится последняя сохраненная конфигурация виртуальных инструментов, созданных ранее на данном компьютере. Если ранее конфигурация не создавалась, то в списке будет присутствовать только самая верхняя строка с названием текущей конфигурации (My Configuration). В любом из этих случаев выделить верхнюю строчку с названием конфигурации, однократно нажав на ней левой клавишей мыши, и затем нажать кнопку «Find», располагающуюся чуть выше. Положительно ответить на все вопросы, появляющиеся во всплывающем окне. Затем убедиться, что напротив виртуального инструмента, соответствующего подключенному устройству, появилась иконка с галочкой на зеленом фоне, означающая, что драйвера сконфигурированы правильно.

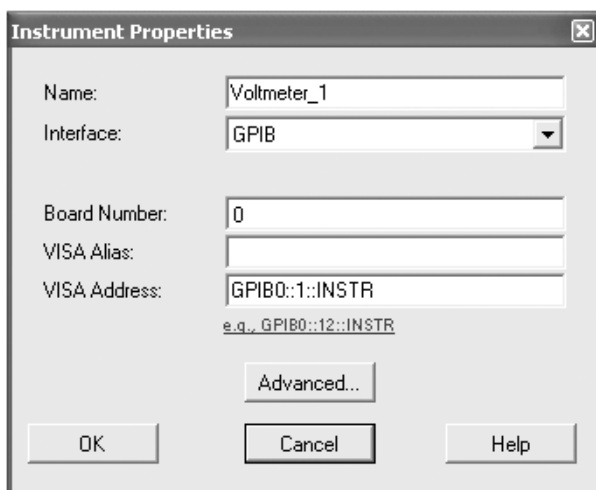


Рис. 4.4

6. Нажать правую кнопку мыши на только что сконфигурированном виртуальном инструменте в списке менеджера инструментов. В появившемся меню выбрать пункт «свойства инструмента» (Instrument Properties) и в поле «Name» появившегося окна свойств инструмента ввести новое имя инструмента латинскими буквами или цифрами без пробелов, например, как на рис. 4.4. Нажать кнопку «OK» в окне свойств инструмента.

7. Снова нажать на виртуальном инструменте правой кнопкой мыши и в появившемся меню выбрать «Создание объекта прямого ввода/вывода» (Create Direct I/O Object). Перевести курсор мыши на свободное пространство основного окна и нажать левую клавишу мыши. Появится объект прямого ввода/вывода (рис. 4.5).

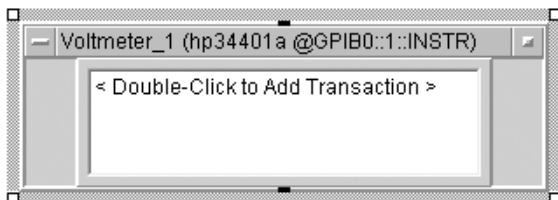


Рис. 4.5



Рис. 4.6

В поле этого объекта нажать два раза левой кнопкой мыши по надписи «Double-Click to Add Transaction» для добавления операции ввода-вывода. В появившемся окне добавления операции ввода/вывода (рис.4.6) ввести команду идентификации «*IDN», затем нажать «OK». Аналогично ввести операцию чтения установив в окне добавления операции ввода/вывода тип операции «READ» и формат данных «STRING FORMAT» (см. рис. 4.7). В поле ввода команды теперь указывается имя выходного терминала, который будет создан у объекта прямого ввода/вывода.

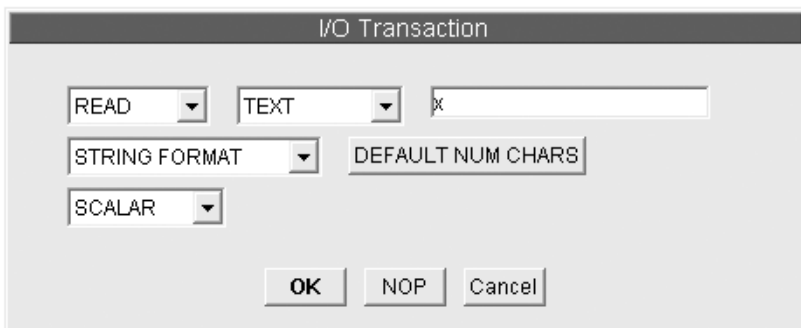


Рис. 4.7

8. Создать объект отображения AlphaNumeric и подключить его к созданному терминалу («x») объекта прямого ввода/вывода. Запустить программу и убедиться, что все работает правильно и в окне AlphaNumeric отображается ответ на запрос, то есть модель устройства, версия прошивки и т.д.

Контрольные вопросы

1. Какие существуют интерфейсы для подключения контрольно-измерительного оборудования к компьютеру?
2. Что такое стандарт и драйвера VISA?
3. Что такое виртуальный инструмент и чем он отличается от аппаратного?
4. Как найти инструмент и послать на него команду при помощи Agilent Connection Expert?
5. Как создать виртуальный инструмент и послать на него команду в среде Agilent VEE?

Задания

1. Получить указания преподавателя, – какой прибор, и по какому интерфейсу подключать к компьютеру.
2. Следуя вышеуказанной методике подключить прибор к компьютеру, обнаружить его при помощи Agilent Connection Expert, послать на него запрос «*IDN» и получить ответ. Распечатать полученное изображение на экране с ответом прибора.

3. Следуя вышеуказанной методике, создать виртуальный инструмент в среде Agilent VEE для подключенного прибора, присвоить ему имя, послать на него запрос «*IDN» и получить ответ. Распечатать полученное изображение на экране после запуска программы.

Список литературы

1. Agilent Technologies IO Libraries Suite 15.5 Connectivity Guide with Getting Started.
2. Agilent VEE Pro Help “How To Use Agilent VEE Pro”, To Change VISA Interface and Address Dynamically.
3. http://en.wikipedia.org/wiki/Virtual_instrumentation
4. http://ru.wikipedia.org/wiki/Виртуальный_прибор
5. http://en.wikipedia.org/wiki/Virtual_Instrument_Software_Architecture
6. http://en.wikipedia.org/wiki/Standard_Commands_for_Programmable_Instruments

ЛАБОРАТОРНАЯ РАБОТА 5

АВТОМАТИЗИРОВАННЫЕ ИЗМЕРЕНИЯ РАЗЛИЧНЫХ ФИЗИЧЕСКИХ ВЕЛИЧИН В РЕАЛЬНОМ ВРЕМЕНИ

Цель – приобретение практических навыков работы с контрольно-измерительным оборудованием, имеющим интерфейсы для приема и передачи команд и данных; изучение основных команд и принципов программирования контрольно-измерительного оборудования.

Введение

Для создания любой автоматизированной контрольно-измерительной системы необходимо знать, как составить программу, использующую объекты ввода/вывода для обращения к инструменту.

Создание программы для автоматизированного измерения

Рассмотрим методику создания программы в среде Agilent VEE Pro, производящей измерения при помощи подключенного к компьютеру аппаратного оборудования, а также математические преобразования результатов измерений. Для этого разберем пример с использованием системы питания Agilent N6700B, подключенной по интерфейсу GPIB к компьютеру, для выполнения которого необходимо совершить следующие действия:

1. Подключить систему Agilent N6700B к компьютеру через интерфейс GPIB и сконфигурировать драйверы, произведя действия, аналогичные изложенным в работе №4. В среде VEE Pro присвоить виртуальному инструменту, соответствующему подключенному аппаратному оборудованию, имя «PowerSupply», которое будет использовано в этой работе для краткого обозначения этого виртуального инструмента. Создать для PowerSupply объект прямого ввода/вывода, послать на него команду идентификации и получить ответ от устройства, запустив программу.

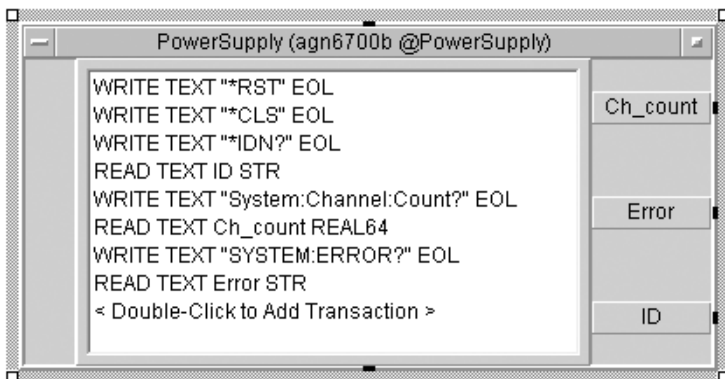


Рис. 5.1

2. Создать объект проверки устройства. Для этого выбрать пользовательский объект (UserObject) из главного меню (Device → UserObject). Зайти в него двойным нажатием левой клавиши мыши и поместить туда созданный объект прямого ввода/вывода для PowerSupply и экран отображения AlphaNumeric. Дополнить объект прямого ввода/вывода командами и интерфейсами, аналогично рис. 5.1.

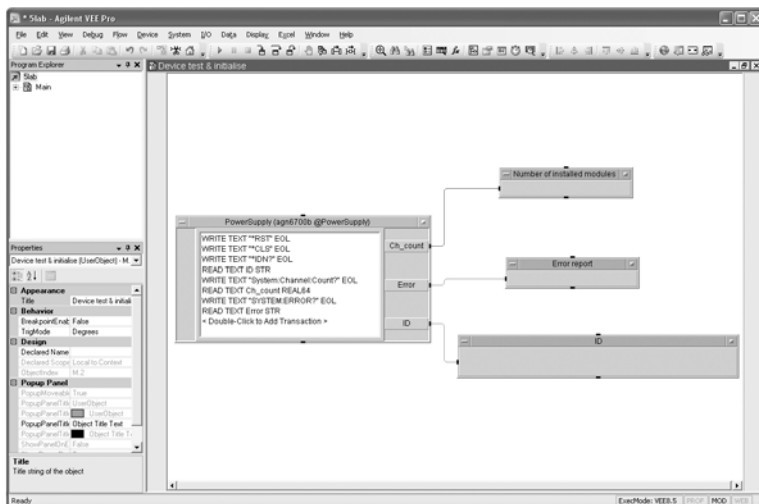


Рис. 5.2

При необходимости можно перемещать строки с командами в списке вверх или вниз, выделив необходимую строку и нажав «Ctrl+Up» или «Ctrl+Down» соответственно. Затем подключить к вновь созданным терминалам объекты AlphaNumeric и получить на экране изображение, аналогичное рис. 5.2. Закрыть этот объект.

3. Используя материал работы №3 или собственную программу, создать в основном окне часть программы, которая будет задавать массив, равномерно заполненный значениями напряжения от минимального до максимального, использовать цикл For Count с задаваемым количеством точек и позволять выбирать в конце программы – завершить ее или продолжить работу. Включить в нее ранее созданный объект проверки устройства самым первым элементом внутри цикла Until Break (рис. 5.3).

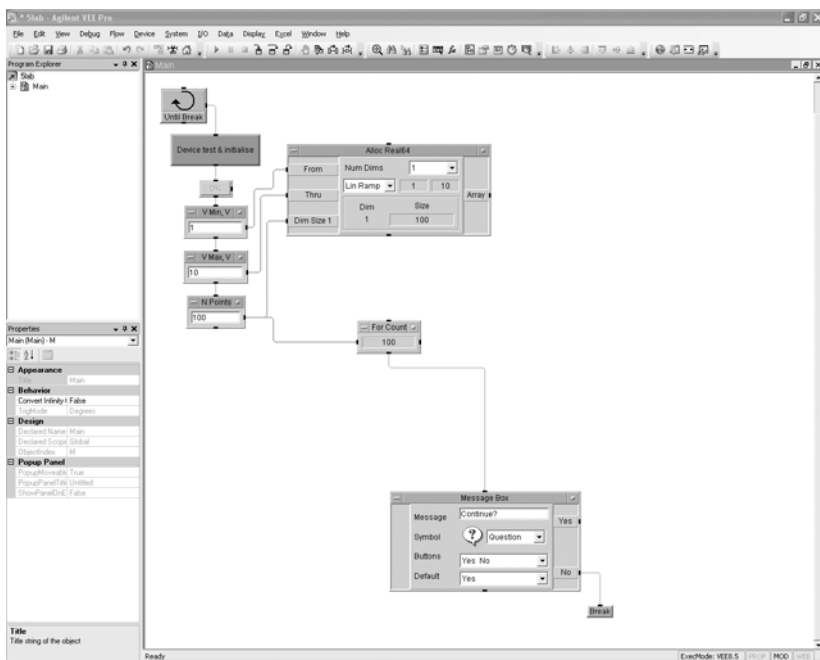


Рис. 5.3

4. Создать константу Real64 для задания ограничения максимального тока в амперах с названием «Curr Lim, A» и константу

Int16 для задания номера рабочего канала системы Agilent N6700B с названием «Channel», включить их в цикл после константы «N Points». Создать объект прямого ввода/вывода для PowerSupply, который будет задавать начальное состояние нашего прибора, ввести туда команды и создать терминалы, аналогично рис. 5.4. Подключить созданные константы к соответствующим терминалам (I – ток, N – номер канала на рис. 5.4).

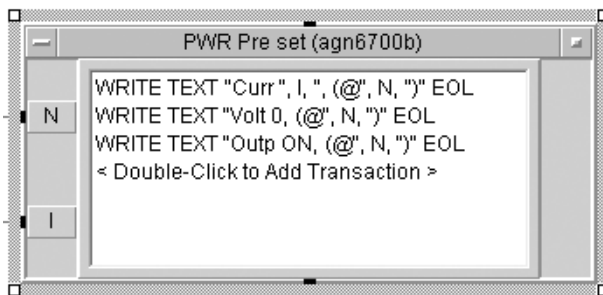


Рис. 5.4

5. Создать константу Real64 для задания задержки в секундах и назвать ее «Delay, sec», создать объект задержки и входной терминал на нем. Включить константу в цикл после «Channel» и подключить к объекту задержки. Выходной терминал последовательности запуска константы «Delay» подключить к входному терминалу последовательности запуска объекта Alloc Real64 для создания правильной последовательности выполнения программы.

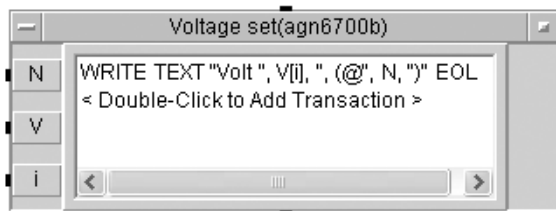


Рис. 5.5

Создать объекты прямого ввода/вывода для PowerSupply, один из которых будет задавать напряжение на выходе выбранного канала (рис. 5.5), а другой запрашивать текущие значения тока и напряжения на выбранном канале (рис. 5.6), пользуясь нижеприве-

денными рисунками. Включить созданные объекты в цикл For Count в следующей последовательности: «задание напряжения», затем «задержка», затем «измерение».

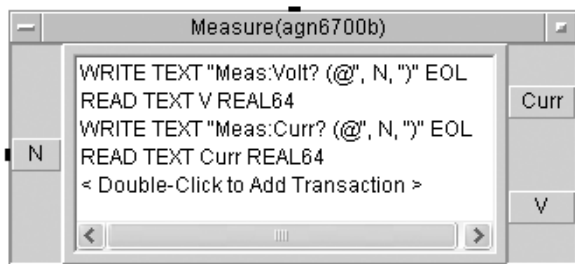


Рис. 5.6

6. Создать формулу, перемножающую два значения для вычисления мощности, выделяемой на выходе системы питания Agilent N6700B, расположить ее внутри цикла For Count последним объектом и подсоединить соответствующие терминалы. Создать график «X vs Y Plot» для графического отображения зависимости мощности от напряжения на выходе системы питания, выбрав его из главного меню (Display→ X vs Y Plot). Имя оси «X» задать как «V, Volts», имя оси «Y» как «Pwr, W». Подсоединить к терминалу оси «X» (верхний терминал) терминал «V» объекта «измерение», а к терминалу оси «Y» результат формулы. Также создать терминал управления «Clear» для графика «X vs Y Plot» и подсоединить его к выходному терминалу главного цикла (Until Break), как это было сделано в пункте 6 работы №3.

7. Соединить объекты задания массива (Alloc Real64), задания начального состояния системы питания (PWR Pre set, рис. 5.4) и цикл For Count в следующей последовательности запуска: «Alloc Real64», затем «PWR Pre set», затем For Count.

8. Создать пользовательский объект с именем «Power OFF» и поместить в него объект прямого ввода/вывода для PowerSupply, в котором ввести команду и создать терминалы в соответствии с рис. 5.7.

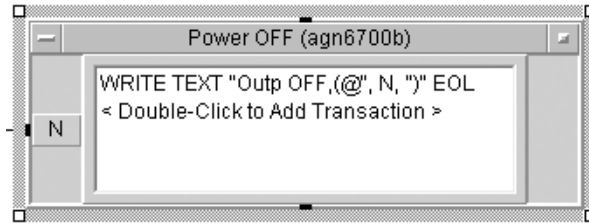


Рис. 5.7

Затем создать входной терминал в пользовательском объекте и соединить терминалы, как показано на рис. 5.8.

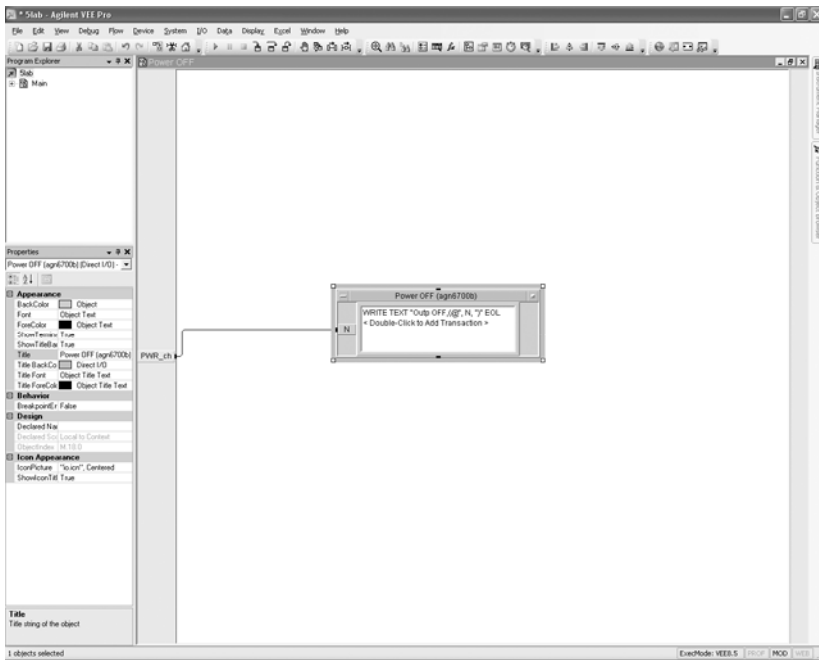


Рис. 5.8

Закрывать пользовательский объект и поместить его в основном окне, соединив его линиями запуска в таком порядке, чтобы он находился после цикла For Count, но перед диалоговым окном «Message Box».

9. Скомпоновать объекты в основном окне так, чтобы они все умещались на экране, и проверить коммутацию всех объектов. После выполнения всех вышеуказанных действий на экране будет картинка, аналогичная рис. 5.9.

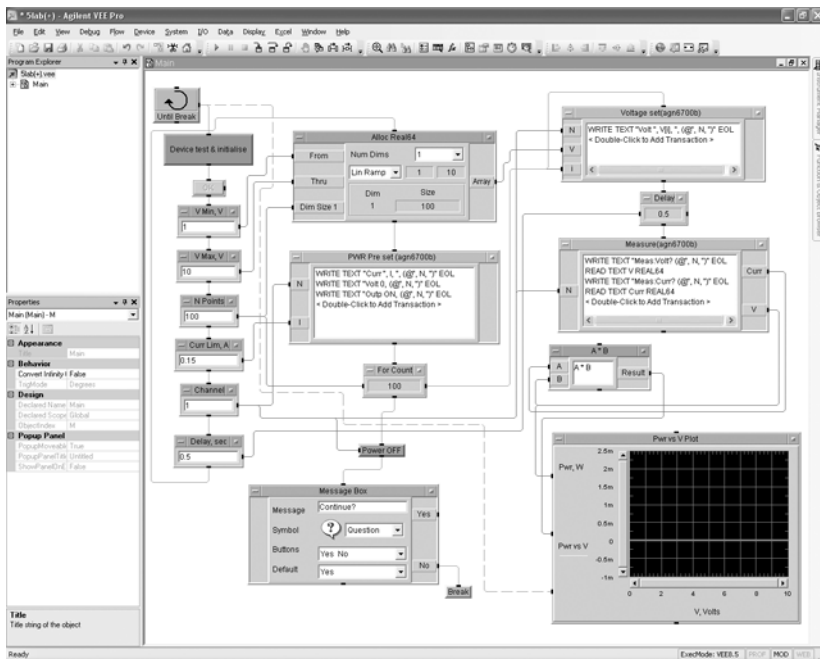


Рис. 5.9

10. Подключить к выходу первого канала системы питания сопротивление порядка 100.Ом, рассчитанное на мощность 0,25 Вт или более. Запустить программу, перед нажатием кнопки «ОК» ввести следующие исходные значения констант: VMin = 1; VMax = 10; N Points = 10; Curr Lim = 0.15; Channel = 1; Delay = 0.5. Затем нажать кнопку «ОК» и проверить работоспособность программы.

В ходе подготовки работы можно пользоваться описанием системы питания Agilent N6700B.

Контрольные вопросы

1. Как пишется стандартная команда идентификации прибора?
2. Что нужно сделать, чтобы отобразить на экране количество подключенных модулей системы питания Agilent N6700B?
3. Как вывести на экран код ошибки системы питания Agilent N6700B?
4. Как написать команды для задания и измерения напряжения с помощью системы питания Agilent N6700B?
5. Какими командами можно включать и выключать выход определенного канала системы питания Agilent N6700B?

Задания

1. Пользуясь вышеизложенным материалом написать программу, последовательно увеличивающую напряжение на выходе источника питания, измеряющую мощность, выделяемую на выходе и строящую график зависимости мощности от напряжения. Основные части программы заготовить заранее, при необходимости пользуясь описанием системы питания Agilent N6700B.
2. Получить у преподавателя образец (сопротивление) и значения констант, используемых в программе.
3. Запустить программу. После ее выполнения распечатать скриншот полученного графика и основных объектов программы. Записать программу на собственный носитель информации.

Список литературы

1. Agilent Technologies Low-Profile Modular Power System Series N6700 User's Guide.

ЛАБОРАТОРНАЯ РАБОТА 6

АВТОМАТИЗИРОВАННАЯ УСТАНОВКА ДЛЯ ИЗМЕРЕНИЯ ВОЛЬТ-АМПЕРНЫХ ХАРАКТЕРИСТИК

Цель – приобретение практических навыков создания автоматизированной измерительной системы для снятия вольт-амперных характеристик.

Введение

В большинстве автоматизированных измерительных систем используются одинаковые принципы и основные алгоритмы. Соответственно для успешного создания автоматизированных контрольно-измерительных систем для определенных задач необходимо изучить эти принципы на конкретных примерах.

Методика измерения и приемы программирования

Получение вольт-амперной характеристики (ВАХ) производится четырехконтактным методом, схема которого представлена на рис.6.1.

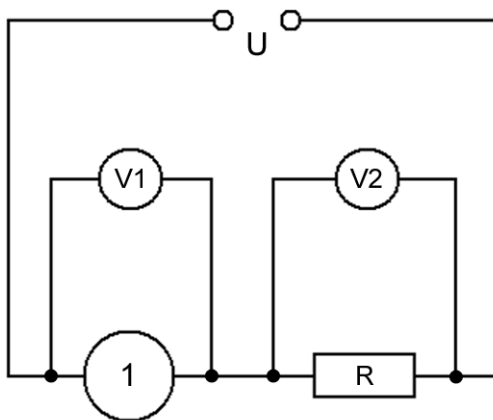


Рис. 6.1

С источника питания подается напряжение U на измерительную цепь, состоящую из образца (1) и измерительного сопротивления R , которое используется для расчета тока в цепи. Соответственно, падение напряжения на образце и измерительном сопротивлении измеряется вольтметрами $V1$ и $V2$. При такой схеме подключения ток по измерительным проводам (проводам, идущим к вольтметрам) не течет, вследствие чего отсутствуют погрешности и ошибки измерения, связанные с сопротивлением измерительных проводов и контактов. Если какая-либо из постоянных или какие-либо данные полученные в ходе программы используются во многих объектах, то вместо применения большого количества линий данных гораздо удобнее задать глобальную переменную и присвоить ей значение этих данных. Тогда можно будет вызывать ее в любой части программы или даже внутри формулы, избегая создания дополнительных линий данных и также уменьшая нагрузку на процессор.

Рассмотрим чуть подробнее использование глобальных (и локальных) переменных. Сначала необходимо декларировать (установить) эту переменную в программе. Для этого и нужно выбрать соответствующий объект из главного меню (Data→Variable→Declare Variable), рис. 6.2.

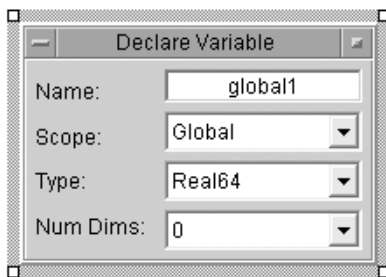


Рис. 6.2

В поле «Name» задается имя переменной (без пробелов) по которому ее можно будет найти и вызвать в программе. Далее, в поле «Scope» указывается ее тип, глобальный или локальный. Мы будем использовать только глобальные переменные. Далее задается формат данных переменной «Type» и ее мерность «Num Dims». То есть, если мы хотим использовать эту переменную, чтобы записать в нее

двумерный массив данных в формате Real32, необходимо поставить «Type»= Real32 и «Num Dims»=2.

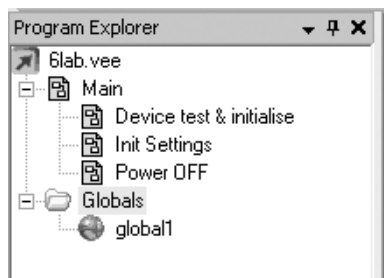


Рис. 6.3

После того как мы расположим этот объект в главном окне и зададим его имя, можно будет всегда быстро получить к нему доступ из меню дерева программы (Program Explorer), в котором появится вкладка «Globals», а в ней название нашей переменной (рис. 6.3). Сначала нужно будет присвоить значение этой переменной, а потом можно будет вызывать значение этой переменной, т.е. порядок выполнения программы должен быть составлен так, чтобы вызывались только те переменные, которым уже присвоены значения. Чтобы присвоить значение переменной или вызвать значение переменной, можно воспользоваться объектами из главного меню (Data→Variable→ Set Variable и Data→Variable→ Get Variable). Также эти объекты можно вызвать, нажав на имени переменной в дереве программы правой кнопкой мыши и в появившемся меню выбрав Generate→Set Variable или Generate→Get Variable соответственно. Можно вызывать значение переменной прямо внутри поля формулы или поля для записи команд в объекте прямого ввода/вывода. Для этого достаточно ввести имя переменной в этом поле. В поле формулы, при правильном введении части имени, появится меню с подсказкой, в котором можно выбрать необходимый объект (рис. 6.4).

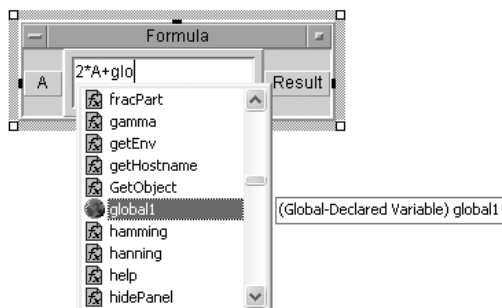


Рис. 6.4

Создание автоматизированной системы измерения

Рассмотрим методику создания измерительной системы для получения вольт-амперных характеристик на следующем примере, в котором будет использоваться среда Agilent VEE Pro, и внешний GPIB/USB контроллер и для выполнения которого необходимо совершить следующие действия:

1. Произвести коммутацию двух мультиметров и системы питания с GPIB/USB контроллером, сконфигурировать драйверы в среде Agilent VEE Pro и убедиться, что коммутация выполнена правильно.

2. Взяв за основу программу, написанную для выполнения работы №5, дополнить ее таким образом, чтобы измерения проводились только при помощи подключенных мультиметров (вместо измерений, проводимых источником питания). В данном случае они будут измерять падения напряжения на образце и измерительном сопротивлении. При необходимости воспользоваться описанием для мультиметров. Создать глобальную переменную для задания номера канала системы питания и подсоединить объект задания переменной к окну константы, куда мы будем вводить номер канала перед запуском основной части программы, как в работе №5. Имя переменной задать «Ch_N», формат данных Int16 и мерность «0». Также создать константу «R_Ref» формата Real64 для задания значения измерительного сопротивления в омах и включить ее в

главный цикл Until Break после константы Delay, задающей значение задержки.

3. Создать пользовательский объект, в котором будут происходить все начальные установки на приборах «Init Settings», а также задание массива напряжений для получения необходимого количества измерений, и переместить туда уже созданные объекты (выделив их и нажав Ctrl+X, затем Ctrl+V). Убрать в объекте прямого ввода/вывода терминал «N», а в командах поставить вместо «N» имя глобальной переменной «Ch_N». Создать в этом объекте соответствующие входные и выходные терминалы аналогично рисунку 6.5 и произвести коммутацию. Аналогично вышеуказанным действиям воспользоваться глобальной переменной во всех остальных объектах программы, где необходимо задавать номер канала системы питания.

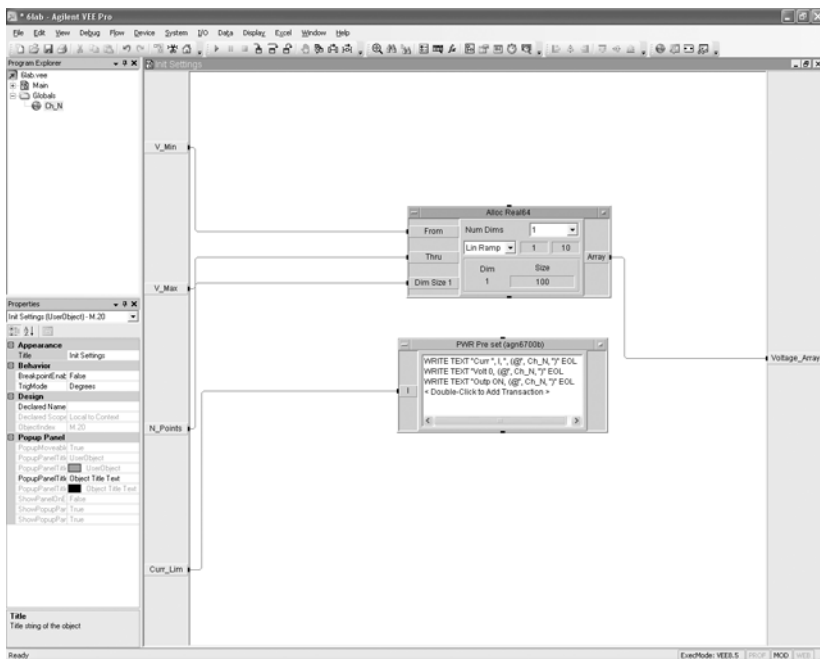


Рис. 6.5

Затем свернуть этот объект и произвести коммутацию в основном окне. Тем же методом создать пользовательский объект для проведения измерений «Measurement», в который поместить все объекты, находящиеся внутри цикла измерений (For Count) за исключением графика, аналогично рис. 6.6. Формулу изменить таким образом, чтобы в ней вычислялись значения тока исходя из известного значения сопротивления и измеренного падения напряжения на нем.

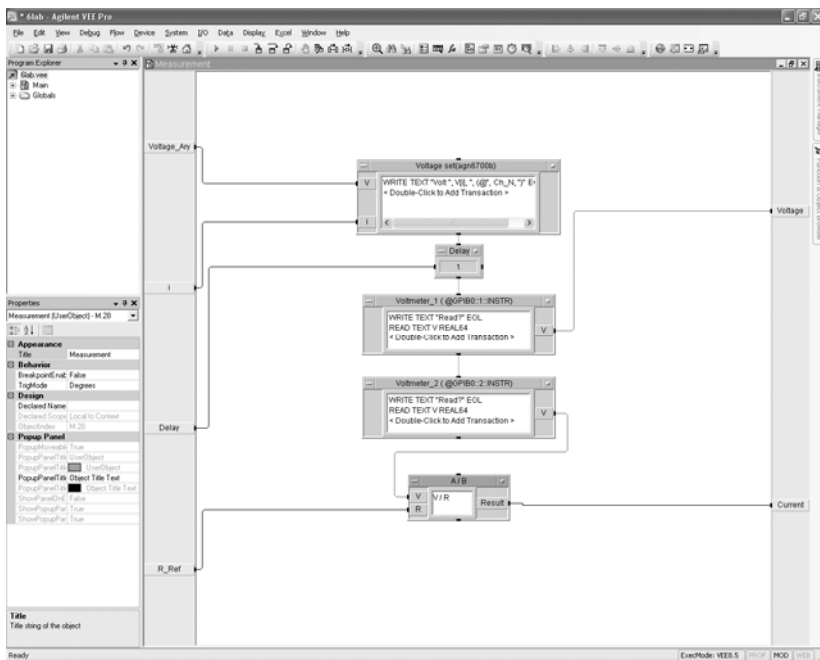


Рис. 6.6

4. Произвести коммутацию в основном окне программы и изменить названия осей на графике и его названия в соответствии с данным типом измерений и получаемых величин. После проведения всех действий на экране будет картинка, аналогичная рис. 6.7. Проверить все части программы и запустить ее в тестовом режиме (без подключения реальной измерительной схемы). При возникновении ошибок исправить их.

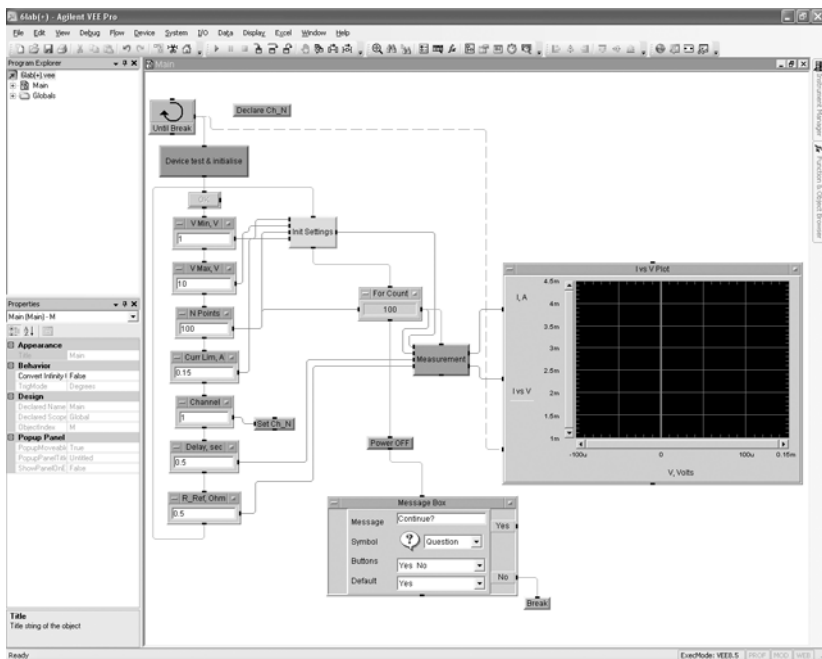


Рис. 6.7

5. Выключить систему питания. Собрать измерительную схему, согласно вышеприведенному описанию и подключить к одному из каналов системы питания. В качестве образца использовать диод, в качестве измерительного сопротивления – сопротивление порядка 100 Ом. При необходимости воспользоваться описанием системы питания Agilent N6700B. Убедиться, что все подключено корректно, затем включить систему питания и произвести измерения.

Контрольные вопросы

1. Объяснить устройство и принцип работы схемы для измерения ВАХ четырехконтактным методом.
2. Как использовать глобальные переменные в среде Agilent VEE Pro?

3. Какими способами можно оптимизировать (уменьшить) количество линий данных, а также количество отображаемых объектов в основном окне?

4. Рассказать про порядок подключения экспериментальной схемы к измерительной установке.

Задания

1. Пользуясь вышеизложенным материалом написать программу, для получения ВАХ. Основные части программы заготовить заранее.

2. Проверить программу, запустив ее в тестовом режиме.

3. Получить у преподавателя образец и измерительное сопротивление. Собрать схему для измерения ВАХ и произвести коммутацию установки.

4. Получить у преподавателя значения констант, используемых в программе.

5. Запустить программу. После ее выполнения распечатать скриншот полученного графика и основных объектов программы. Записать программу на собственный носитель информации.

Список литературы

1. Agilent Technologies Low-Profile Modular Power System Series N6700 User's Guide.

2. Agilent 34401A 6 ½ Digit Multimeter User's Guide.

3. Agilent VEE Pro Help "How To Use Agilent VEE Pro".

ЛАБОРАТОРНАЯ РАБОТА 7

АВТОМАТИЗИРОВАННАЯ УСТАНОВКА ДЛЯ ИЗМЕРЕНИЯ ВОЛЬТ-ФАРАДНЫХ ХАРАКТЕРИСТИК

Цель – приобретение практических навыков создания автоматизированной измерительной системы для снятия вольт-фарадных характеристик.

Введение

В большинстве автоматизированных измерительных систем используются одинаковые принципы и основные алгоритмы. Соответственно, для успешного создания автоматизированных контрольно-измерительных систем для определенных задач необходимо изучить эти принципы на конкретных примерах.

Методика измерения и приемы программирования

Для получения вольт-фарадных характеристик (ВФХ) существует множество методик. Рассмотрим простейшую из них, для получения которой используется только специальный измеритель емкости (LCR)-метр.

LCR-метр подключается напрямую к контактам образца при помощи специальных измерительных кабелей, по которым подается тестовый переменный сигнал с заданной частотой, а также постоянное напряжение смещения на образце. LCR-метр Agilent E4980A имеет специальный встроенный источник питания, который позволяет задавать напряжения смещения на образце в определенных пределах, величина которых зависит от опций, установленных в этом приборе. Этот встроенный источник напряжения развязан с измерительным трактом таким образом, что не оказывает влияния на результаты измерения и лишь устанавливает смещение. Если же использовать стандартный внешний источник питания, то емкость выходных элементов его схемы будет оказывать большое влияние на результаты измерений.

Если после измерений необходимо сохранить полученные данные и зависимости, то можно воспользоваться экспортом данных в MS Office Excel.

Для того чтобы построить график зависимости в MS Office Excel, данные необходимо привести к виду двумерного массива, первый элемент второго измерения которого будет соответствовать значению по оси «X», а остальные элементы – значениям по оси «Y». Таким образом, чтобы построить график ВАХ, у нас должен быть двумерный массив (\mathbf{a}), первый элемент второго измерения которого ($\mathbf{a}[i,1]$) будет содержать значение напряжения, а второй элемент ($\mathbf{a}[i,2]$) – значение тока для точки с номером i .

Для того чтобы воспользоваться библиотекой Excel и построить график в Excel, необходимо сначала вызвать эту библиотеку в среде Agilent VEE Pro при помощи специального объекта «Initialize Excel Library», затем воспользоваться объектом «Call xlLib.NewWorkbook» для создания новой книги Excel, потом воспользоваться объектом «CreateXYChart» для создания таблицы и объектом «Excel Settings» для установки параметров отображения окна Excel. После чего, если в дальнейшей части программы Excel не будет использоваться, желательно удалить библиотеку xlLib из программы, воспользовавшись объектом «Quit Excel Library». Все эти объекты можно найти в главном меню (Excel → ...) Внутри каждого объекта присутствуют дополнительные пояснения и инструкции по использованию.

Для более быстрого написания программы рекомендуется использовать специальные шаблоны, уже созданные в среде Agilent VEE Pro. Вызвать необходимый шаблон можно из главного меню (Excel → Quick Start (Short Cuts)) и далее можно дополнить его или заменить какие-либо его элементы в соответствии с задачами конкретной программы.

Рассмотрим немного подробнее объект «CreateXYChart». Сразу после создания он будет выглядеть, как на рис. 7.1. Но если воспользоваться им в таком виде на компьютере, где установлена не англоязычная версия MS Office Excel, то возникнет ошибка и график не будет построен.

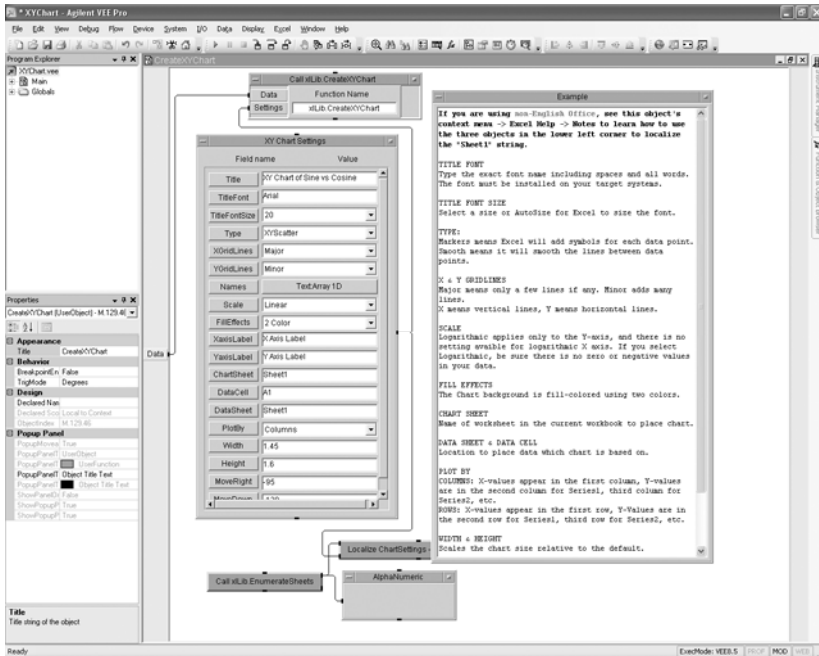


Рис. 7.1

Ошибка возникает вследствие несовпадения имен листов в конкретной версии MS Office Excel с заданными английскими именами листов в программе (Sheet1 в поле «DataSet»). Так как среда Agilent VEE Pro рассчитана на работу с английскими символами, то корректно задать имя листа на русском языке не получится. Но можно воспользоваться специальной функцией, которая создана в среде Agilent VEE Pro, и которая присутствует в объекте «CreateXYChart». Она реализована при помощи двух объектов «Call xLib.EnumerateSheets» и «Localize ChartSettings -> "Sheet1" string». Необходимо только изменить существующую в объекте «CreateXYChart» коммутацию, подсоединив выходной терминал «Localize ChartSettings -> "Sheet1" string» ко входному терминалу (Settings) объекта «Call xLib.CreateXYChart», как показано на рис.7.2.

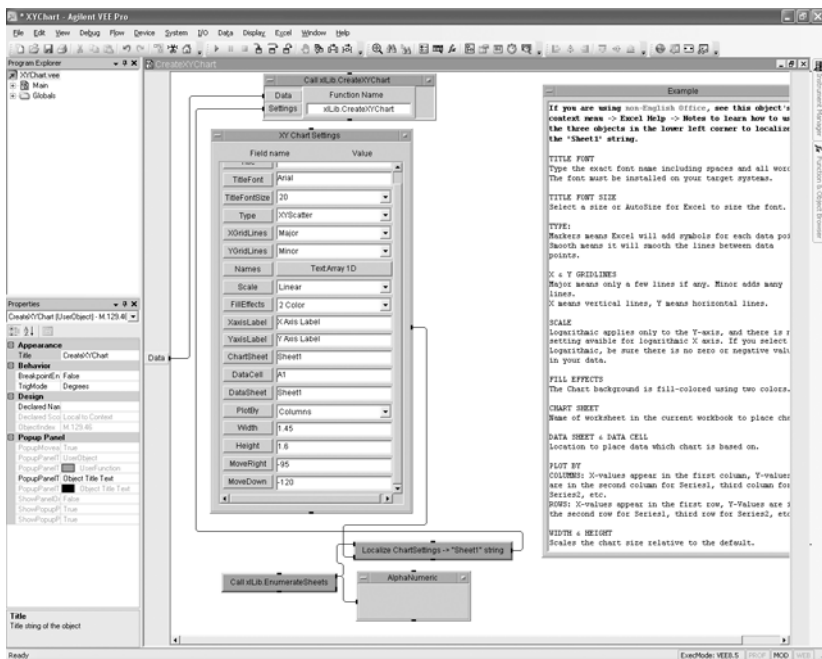


Рис. 7.2

Теперь можно использовать объект «CreateXYChart» для создания графика в MS Office Excel с любой локализацией (языком, используемым в программе).

Создание автоматизированной системы измерения

Рассмотрим методику создания измерительной системы для получения вольт-фарадных характеристик на следующем примере, в котором будет использоваться среда Agilent VEE Pro и внешний GPIB/USB контроллер и для выполнения которого необходимо совершить следующие действия:

1. Произвести коммутацию LCR-метра, мультиметра (вольтметра) и системы питания с GPIB/USB контроллером, сконфигурировать драйверы в среде Agilent VEE Pro и убедиться, что коммутация выполнена правильно.

2. Взяв за основу программу, написанную для выполнения работы №6, дополнить ее таким образом:

2.1. Удалить все объекты ввода/вывода и виртуальные инструменты в менеджере инструментов, относящиеся к мультиметрам и системе питания, использовавшимся в работе №6. Удалить пользовательский объект «Power OFF» и восстановить порядок выполнения главного цикла. Создать в менеджере инструментов виртуальный инструмент с именем «LCR» для LCR-метра и сконфигурировать драйверы устройства.

2.2. В объекте, где производятся измерения (Measurement), создать два объекта прямого ввода/вывода для LCR и поместить их до и после задержки Delay. В верхнем объекте ввода/вывода ввести команды и создать терминалы аналогично рис. 7.3, а в нижнем – как на рис.7.4.

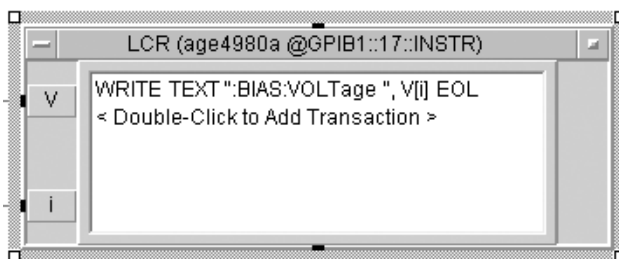


Рис. 7.3

Удалить формулу вычисления тока и терминалы объекта «Measurement», через которые задавалось значение измерительного сопротивления. Переименовать терминалы внутри объекта «Measurement» в соответствии с заданным типом измерений, произвести коммутацию объектов, после которой на экране будет изображение, аналогичное рис. 7.5.

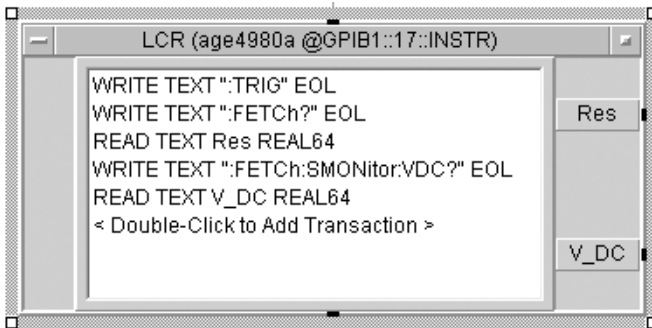


Рис. 7.4

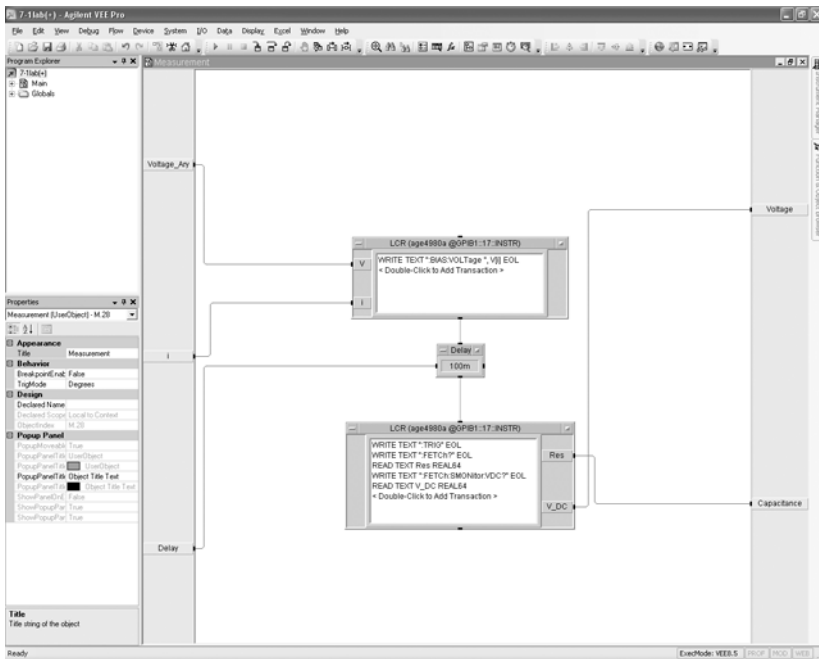


Рис. 7.5

2.3. В объекте, задающем исходные значения (Init Settings), также создать объект прямого ввода/вывода для LCR с командами и терминалами аналогично рис. 7.6. «Freq» служит для задания рабочей частоты измерительного сигнала, а «Volt» - для задания амплитуды измерительного сигнала. Создать необходимые дополни-

тельные терминалы и, при необходимости, удалить не использованные в объекте «Init Settings».

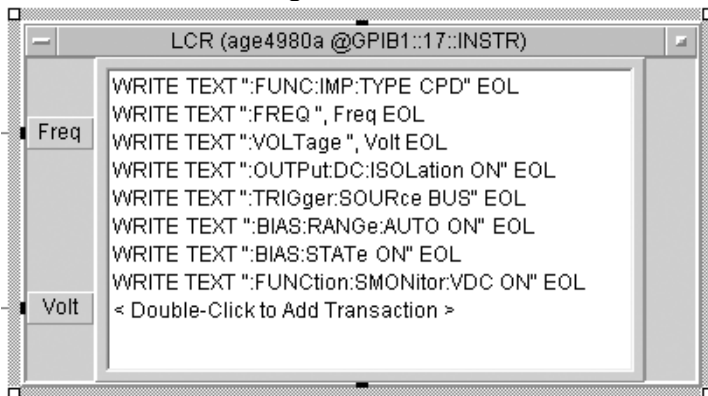


Рис. 7.6

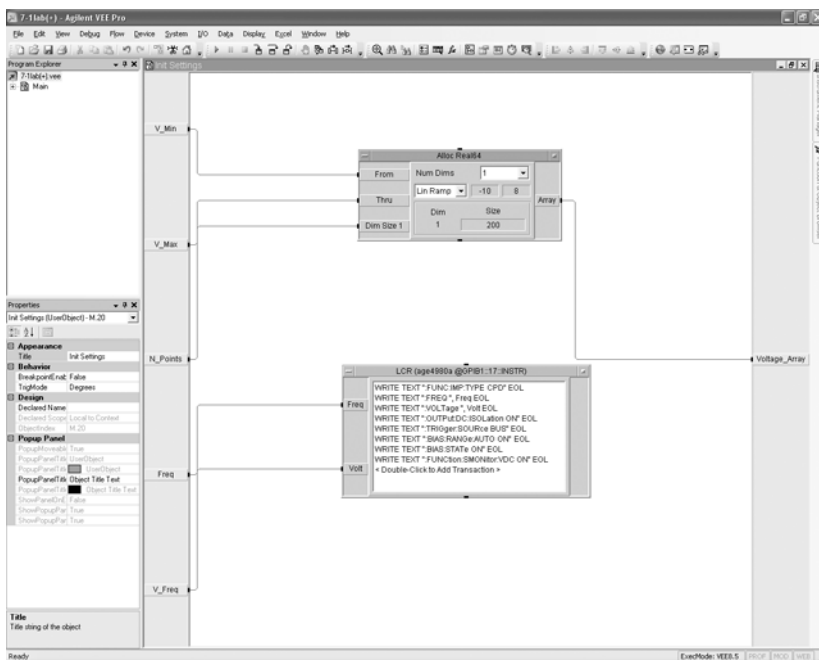


Рис. 7.7

Произвести коммутацию, получив на экране изображение, аналогичное рис. 7.7.

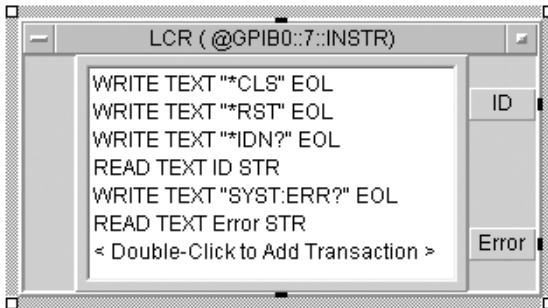


Рис. 7.8

2.4. В объекте инициализации устройств (Device Test & Initialise) аналогично создать объект прямого ввода/вывода для LCR с командами и терминалами, изображенными на рис. 7.8.

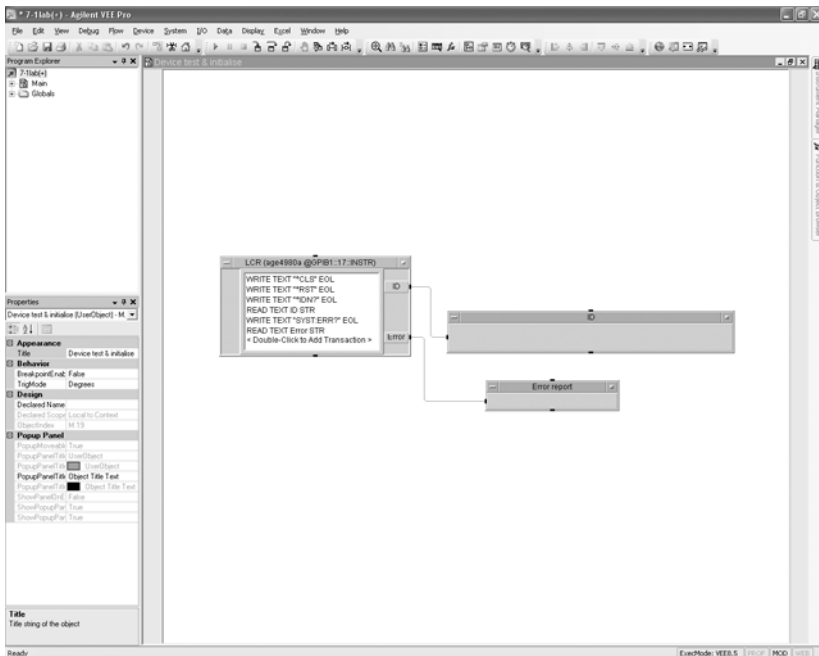


Рис. 7.9

Создать объекты AlphaNumeric для отображения информации и произвести коммутацию, как показано на рис. 7.9.

2.5. В главном окне программы удалить ненужные константы Curr Lim, Channel и R_Ref, а также все объекты переменной Ch_N. Создать константы Real64 для задания частоты в герцах и амплитуды в вольтах измерительного сигнала LCR-метра и произвести коммутацию с терминалами объекта Init Settings. Расположить эти константы внутри главного цикла Until Break после константы Delay и восстановить его последовательность выполнения линиями запуска. Изменить названия графика и осей на графике в соответствии с типом проводимых измерений, получив, в итоге, на экране изображение, аналогичное рис. 7.10. Проверить все части программы и запустить ее в тестовом режиме (без подключения реальной измерительной схемы). При возникновении ошибок исправить их.

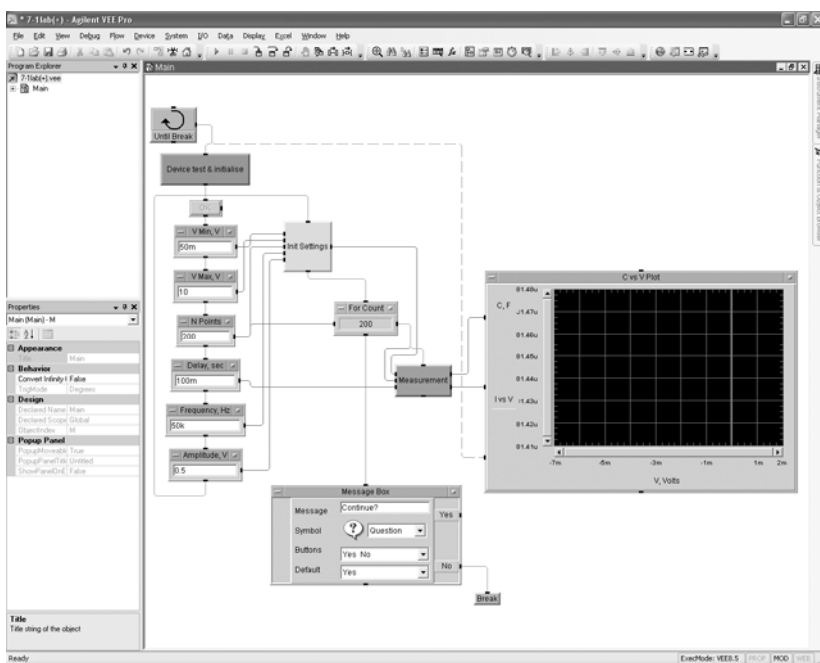


Рис. 7.10

3. Подключить образец к LCR-метру, ввести необходимые параметры измерения в поля констант и запустить программу.

Примечание: вышеописанный пример приведен для изучения основных принципов создания автоматизированной установки для измерения ВФХ. Реальная программа для измерений ВФХ и других измерений, проводимых с использованием LCR-метра, значительно более сложная и объемная и не может быть рассмотрена и создана в ходе одной лабораторной работы.

Контрольные вопросы

1. Объяснить принцип работы установки для измерения ВФХ, состоящей только из LCR-метра.
2. Как использовать MS Office Excel в среде Agilent VEE Pro?
3. Как создать график «XY Chart» в локализованной (не английской) версии MS Office Excel при помощи среды Agilent VEE Pro?
4. Какие начальные параметры необходимо задать на LCR-метре для проведения простейших измерений?

Задания

1. Пользуясь вышеизложенным материалом написать программу для получения ВФХ. Основные части программы заготовить заранее.
2. Проверить программу, запустив ее в тестовом режиме.
3. Получить у преподавателя образец. Затем собрать схему для измерения ВФХ и произвести коммутацию установки.
4. Получить у преподавателя значения констант, используемых в программе.
5. Запустить программу. После ее выполнения распечатать скриншот полученного графика и основных объектов программы. Записать программу на собственный носитель информации.
6. (*) Используя материал, изложенный в ходе этой и предыдущих работ, дополнить программу таким образом, чтобы полученная зависимость сохранялась в MS Office Excel.

Список литературы

1. Agilent E4980A Precision LCR Meter User's Guide Eighth Edition.

ЛАБОРАТОРНАЯ РАБОТА 8

АВТОМАТИЗИРОВАННАЯ УСТАНОВКА ДЛЯ ИЗМЕРЕНИЯ КРИТИЧЕСКОЙ ТЕМПЕРАТУРЫ СВЕРХПРОВОДНИКОВ

Цель – приобретение практических навыков создания автоматизированной измерительной системы с использованием криогенной техники.

Введение

В большинстве автоматизированных измерительных систем используются одинаковые принципы и основные алгоритмы. Специфическими устройствами в криогенной технике являются электронный термометр и управляемый нагреватель, зачастую объединенные в один прибор, называющийся контроллером температуры. Соответственно, для успешного создания автоматизированных контрольно-измерительных систем с использованием криогенной техники, необходимо разобрать конкретный пример с использованием контроллера температуры.

Методика измерения

Рассмотрим схему простейшей установки для измерения критической температуры сверхпроводников, рис 8.1. На образце (1), находящемся в криостате, задается ток A с источника питания, падение напряжения на образце измеряется вольтметром V , а температура при помощи полупроводникового термодатчика измеряется контроллером температуры T . Сопротивление сверхпроводников обычно достаточно большое в нормальном состоянии, поэтому, даже при низких температурах можно использовать достаточно маленькие токи для получения падения напряжения, которое легко детектировать при помощи стандартного мультиметра (десятки микровольт). При переходе в сверхпроводящее состояние падение напряжения на сверхпроводнике станет нулевым. При использовании системы питания в качестве источника тока, необходимо задать на ней напряжение, превышающее максимально возможное в эксперименте падение напряжения на образце, а затем задавать необходимые значения тока. При таком задании параметров система

будет работать в режиме ограничения по току, и мы сможем задавать любые необходимые значения тока, не изменяя значение напряжения.

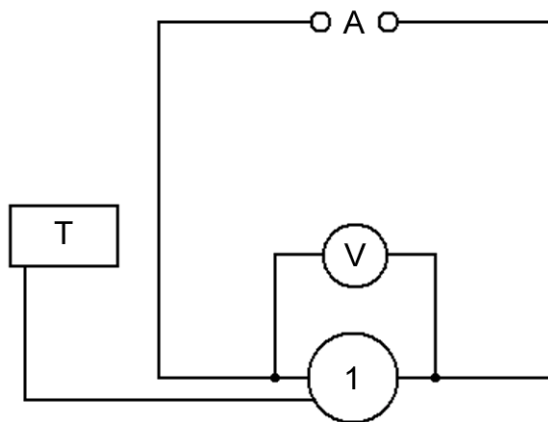


Рис. 8.1

Создание автоматизированной системы измерения

Рассмотрим методику создания измерительной системы для измерения критической температуры на следующем примере, в котором будет использоваться среда Agilent VEE Pro, и внешний GPIB/USB контроллер. Программа будет задавать определенное значение тока через сверхпроводник и снимать зависимость напряжения на образце от температуры, с заданным шагом по температуре. Для реализации этой методики и написания программы необходимо выполнить следующие действия:

1. Произвести коммутацию контроллера температуры, мультиметра (вольтметра) и системы питания с GPIB/USB контроллером, сконфигурировать драйверы в среде Agilent VEE Pro и убедиться, что коммутация выполнена правильно.

2. Взяв за основу программу, написанную для выполнения работы №6, дополнить ее таким образом:

2.1. Удалить все объекты, включая виртуальный инструмент в менеджере инструментов, относящиеся ко второму мультиметру, измерявшему падение напряжения на измерительном сопротивлении в работе №6. Удалить формулу вычисления тока и терминалы, через которые задавалось значение измерительного сопротивления. Также удалить саму константу «R_Ref». Аналогично удалить задержку (Delay) и соответствующую ей константу.

2.2. Создать в менеджере инструментов виртуальный инструмент с именем «TempController» для контроллера температуры. Затем в объекте, где производятся измерения («Measurement»), удалить объект прямого ввода/вывода для источника питания и создать объект прямого ввода/вывода для контроллера температуры. В созданном объекте ввода/вывода ввести команды, как показано на рис. 8.2, на выходе этого объекта мы будем получать значения текущей температуры.

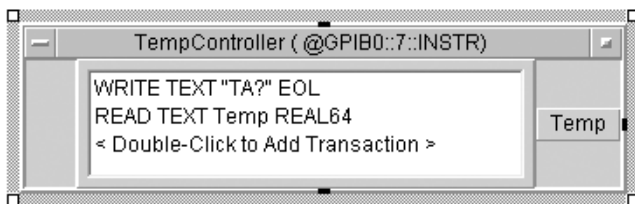


Рис. 8.2

2.3. Теперь необходимо создать объект формулы, который будет вычислять модуль разности текущей температуры и температуры последнего измерения. Для этого следует воспользоваться функцией $\text{mag}(x)$, выдающей в качестве результата реальную часть числа «x», то есть являющейся модулем для действительных чисел. То есть создать объект «Formula» из главного меню (Device→Formula), присвоив ей имя «Magnitude», и создать в ней еще один входной терминал «B». Внутри написать « $\text{mag}(A-B)$ », как показано на рис. 8.3.

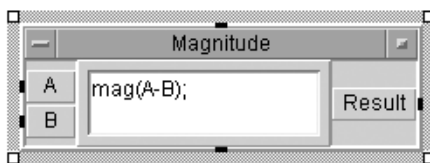


Рис. 8.3

Подключить выходной терминал объекта «TempController» к терминалу «А» формулы, создать в объекте «Measurement» входной терминал с именем «Init_Temp» и подключить его к терминалу «В» формулы.

2.4. Создать объект If/Then/Else из главного меню (Flow→If/Then/Else), который будет запускать измерение в случае, если разница температур, вычисляемая вышеописанной формулой, будет больше порогового значения (шага по температуре). Добавить этому объекту входной терминал «В» и ввести логическую формулу, как показано на рис. 8.4. Создать в объекте «Measurement» входной терминал с именем «Temp_Step». Затем подсоединить к терминалу «А» результат «Magnitude», а к терминалу «В» – терминал «Temp_Step».

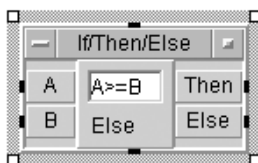


Рис. 8.4

2.5. Создать еще один объект формулы с именем «Trigger» и написать в нем просто значение входного терминала «А», то есть который не будет вносить никаких преобразований. Он будет служить в качестве триггера. Подсоединить к входному терминалу «А» выходной терминал «TempController». Создать в объекте «Measurement» выходной терминал для температуры с именем «Temperature» и подсоединить к нему выходной терминал этой формулы.

2.6. Подсоединить выходной терминал «Then» объекта If/Then/Else к входному терминалу запуска формулы Trigger, а следующим объектом в этой последовательности поставить объект прямого ввода/вывода вольтметра. Удалить все неиспользованные терминалы в объекте «Measurement» и получить на экране картинку, аналогичную рис. 8.5.

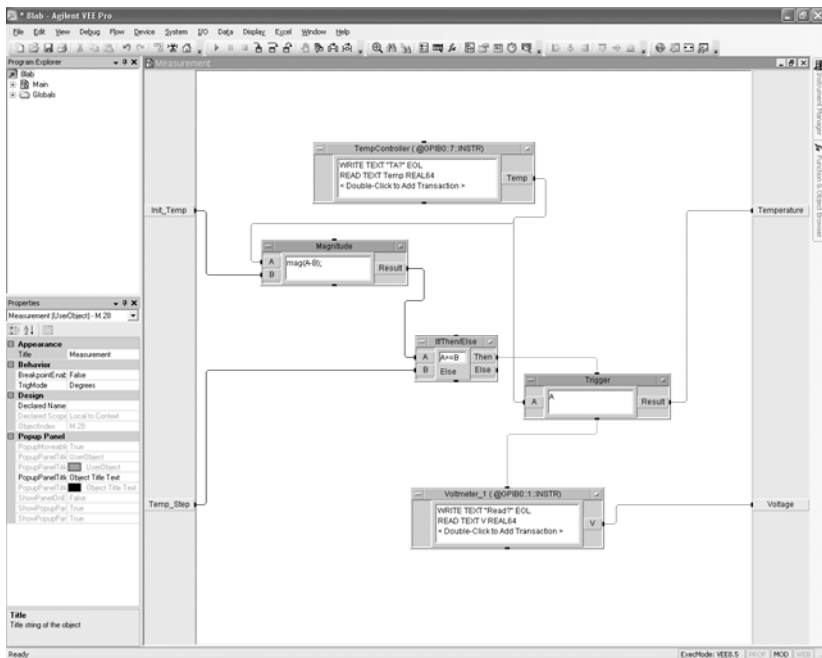


Рис. 8.5

2.7. В основном окне программы создать две константы Real64 с названиями «Init Temp, K» и «Temp Step, K» для задания начальной температуры и шага по температуре в кельвинах. Включить их в главный цикл Until Break между константой «Channel» и объектом «Init Settings». Подсоединить их к соответствующим терминалам объекта «Measurement». Значение константы «Init Temp» можно оставить равным 0, тогда первое измерение будет проводиться при той же температуре, при которой запущена программа. Включить эти константы в главный цикл программы (Until Break) после константы, задающей номер канала.

2.8. Удалить константы, задающие максимальное напряжение и количество точек. Константу, задававшую минимальное напряжение, переименовать в «Voltage, V», теперь она будет задавать исходное значение напряжения в системе питания. Создать линию запуска для корректной работы главного цикла, восстанавливающую порядок действий до удаления этих констант.

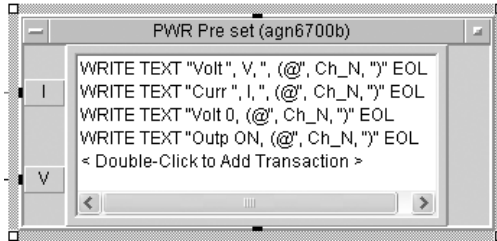


Рис. 8.6

2.9. Внутри объекта «Init Settings» удалить массив, задававший значения напряжения, а также ненужные входные терминалы, один из терминалов переименовать в «Voltage». Объект прямого ввода/вывода для источника питания дополнить, как указано на рис.8.6 для задания начального напряжения на источнике питания. После коммутации данных объект «Init Settings» должен выглядеть аналогично рис. 8.7.

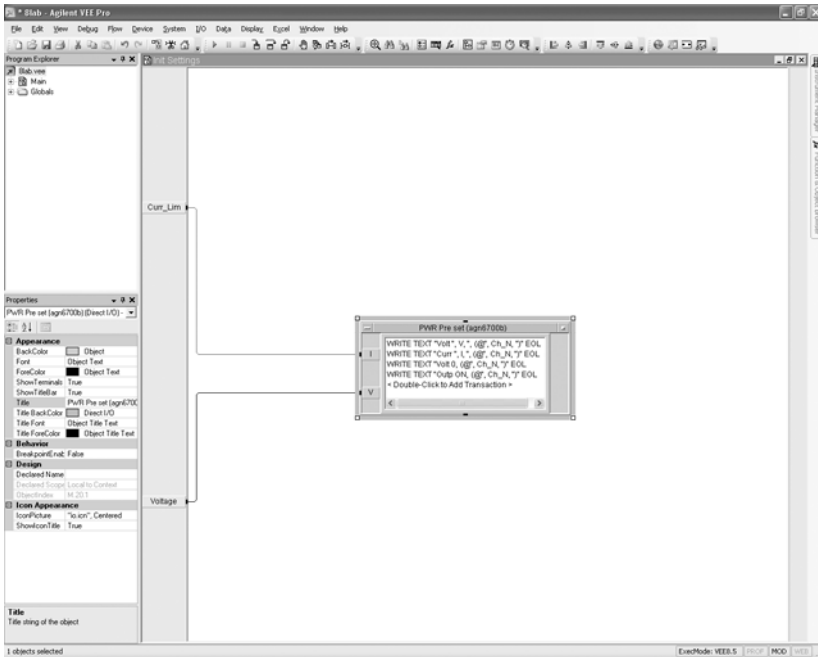


Рис. 8.7

2.10. В основном окне программы измерительный цикл изменить с For Count на Until Break и восстановить коммутацию объектов. Создать две кнопки «OK» и задать им названия «Start» и «Stop». Задать глобальную переменную с именем «Stop_TRG» и форматом данных Int16. Создать константу Int16 и поместить ее в главный цикл программы Until Break самым первым объектом. Значение константы оставить равным 0. Создать объект для задания значения переменной Stop_TRG и подсоединить к нему эту константу. Кнопку «Start» поместить внутри последовательности программы между объектами «Init Settings» и измерительным циклом Until Break. Выходной терминал последовательности запуска объекта «Init Settings» также подключить к входному терминалу последовательности запуска кнопки «Stop».

2.11. Создать пользовательский объект с названием «Stop Measurement». Внутри него создать константу Int16 и задать ей значение, равное 1. Затем создать объект для задания значения переменной «Stop_TRG» и подсоединить к нему эту константу (рис. 8.8). В основном окне программы подсоединить выходной терминал последовательности запуска кнопки «Stop» к соответствующему входному терминалу объекта «Stop Measurement».



Рис. 8.8

2.12. Внутри объекта «Measurement» создать логический элемент If/Then/Else с терминалами и логической формулой, аналогично рис. 8.9.

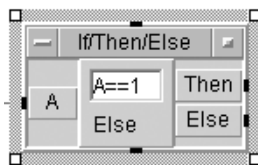


Рис. 8.9

Создать также объект получения значения переменной Stop_TRG и подсоединить это значение к входному терминалу «А» логического объекта. Создать в объекте «Measurement» выходной терминал с именем «TRG» и подсоединить к нему выходной терминал Then логического объекта. Терминал «Else» подключить к входному терминалу последовательности запуска объекта прямого ввода/вывода «TempController». Скомпоновав объекты внутри «Measurement», получить на экране картинку, аналогичную рис.8.10.

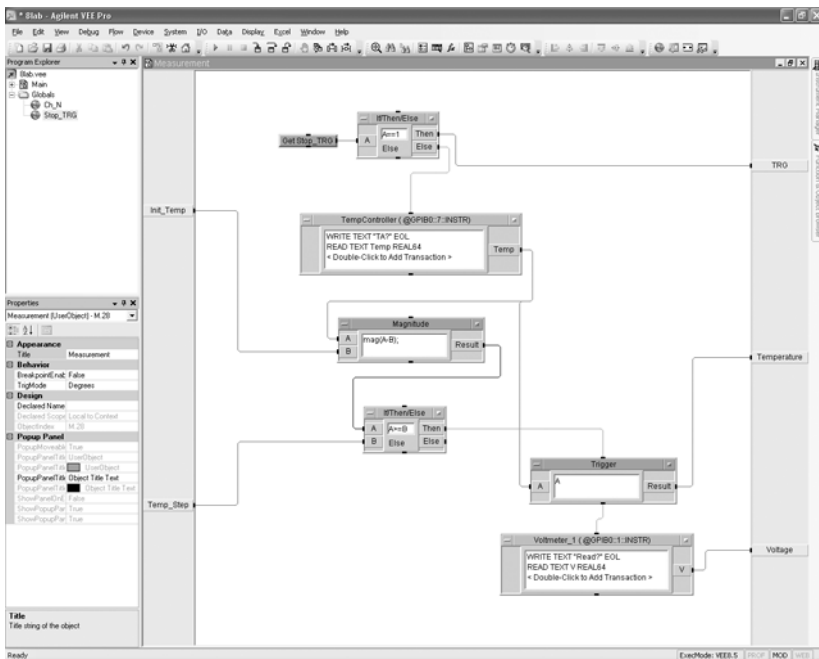


Рис. 8.10

2.13. В основном окне программы создать объект «Break» и подсоединить его к терминалу TRG объекта «Measurement». Создать объект объединения сигналов «Junction» (Flow → Junction в главном меню) и подсоединить его выходной терминал к терминалу «Init_Temp» объекта «Measurement». Затем подключить выходной терминал «Temperature» к одному из входных терминалов объекта

Junction, а к другому входному терминалу подсоединить константу «Init_Temp».

2.14. Произвести коммутацию оставшихся терминалов и изменить названия графика и осей на графике в соответствии с типом проводимых измерений (зависимость напряжения от температуры). Скомпоновать объекты программы, проверить коммутацию объектов и получить на экране изображение, аналогичное рис. 8.11. Проверить все части программы и запустить ее в тестовом режиме (без подключения реальной измерительной схемы). При возникновении ошибок исправить их.

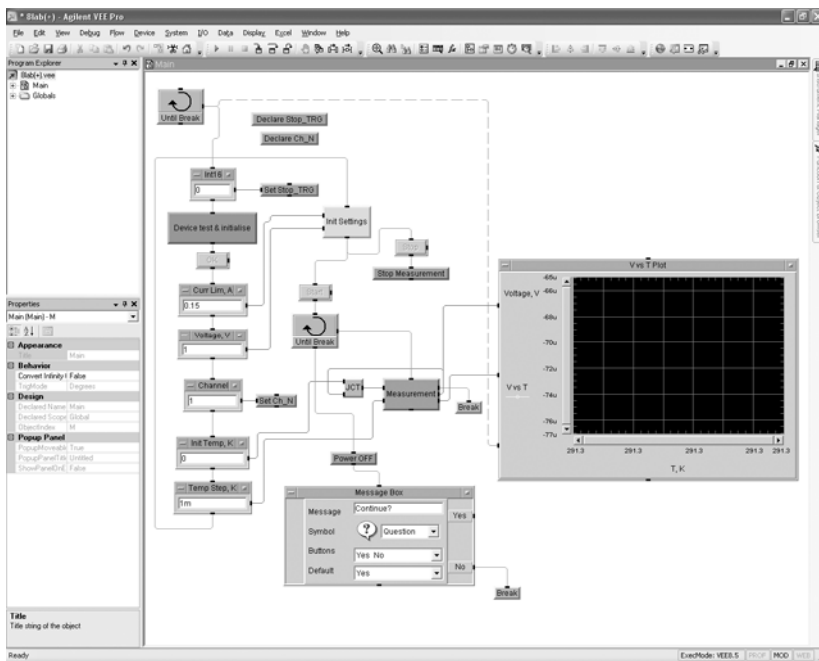


Рис. 8.11

3. Выключить систему питания. Собрать измерительную схему, согласно вышеприведенному описанию и подключить к одному из каналов системы питания. При необходимости воспользоваться описанием системы питания Agilent N6700B. Задать необходимые

значения констант. Убедиться, что все подключено корректно, затем включить систему питания и произвести измерения.

Контрольные вопросы

1. Объяснить устройство и принцип работы схемы для измерения критической температуры.
2. Как использовать систему питания в режиме ограничения (стабилизации) тока?
3. Объяснить для чего используются логические объекты If/Then/Else в вышеприведенном примере.
4. Как реализована функция запуска и остановки измерений в вышеприведенном примере?
5. Объяснить для чего используется объект Junction в вышеприведенном примере.

Задания

1. Пользуясь вышеизложенным материалом написать программу для получения зависимости напряжения на образце от температуры образца. Основные части программы заготовить заранее.
2. Проверить программу, запустив ее в тестовом режиме.
3. Собрать установку для измерения критической температуры, при необходимости используя руководства к оборудованию и указания преподавателя.
4. Получить у преподавателя значения констант, используемых в программе. Запустить программу. После ее выполнения распечатать скриншот полученного графика и основных объектов программы. Записать программу на собственный носитель информации.
5. (*) Используя материал, изложенный в ходе предыдущих работ, дополнить программу таким образом, чтобы полученная зависимость сохранялась в MS Office Excel.

Список литературы

1. Agilent Technologies Low-Profile Modular Power System Series N6700 User's Guide.
2. Operation Manual for 9700 Temperature Controller, Scientific Instruments, INC. 090-256.
3. Agilent VEE Pro Help "How To Use Agilent VEE Pro".

Содержание

Лабораторная работа 1. МОДЕЛИРОВАНИЕ ЭЛЕКТРИЧЕСКИХ СИГНАЛОВ РАЗЛИЧНОЙ ФОРМЫ, МАТЕМАТИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ РЕЗУЛЬТАТОВ.....	5
Лабораторная работа 2. ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ VEE PRO. ОСНОВЫ ПРОГРАММНОЙ ЛОГИКИ.....	11
Лабораторная работа 3. МАССИВЫ ДАННЫХ. МОДЕЛИРОВАНИЕ ЭКСПЕРИМЕНТА.....	19
Лабораторная работа 4. КОММУТАЦИЯ ОБОРУДОВАНИЯ, УСТАНОВКА И КОНФИГУРАЦИЯ ДРАЙВЕРОВ.....	27
Лабораторная работа 5. АВТОМАТИЗИРОВАННЫЕ ИЗМЕРЕНИЯ РАЗЛИЧНЫХ ФИЗИЧЕСКИХ ВЕЛИЧИН В РЕАЛЬНОМ ВРЕМЕНИ	39
Лабораторная работа 6. АВТОМАТИЗИРОВАННАЯ УСТАНОВКА ДЛЯ ИЗМЕРЕНИЯ ВОЛЬТ-АМПЕРНЫХ ХАРАКТЕРИСТИК.....	47
Лабораторная работа 7. АВТОМАТИЗИРОВАННАЯ УСТАНОВКА ДЛЯ ИЗМЕРЕНИЯ ВОЛЬТ-ФАРАДНЫХ ХАРАКТЕРИСТИК.....	55
Лабораторная работа 8. АВТОМАТИЗИРОВАННАЯ УСТАНОВКА ДЛЯ ИЗМЕРЕНИЯ КРИТИЧЕСКОЙ ТЕМПЕРАТУРЫ СВЕРХПРОВОДНИКОВ.....	65

Редактор Г.А. Петрухина

Подписано в печать 15.11.2011. Формат 60×84 1/16.

Печ.л. 4,75. Уч.-изд.л. 4,75. Тираж 100 экз.

Изд. № 3/1. Заказ № 63

Национальный исследовательский ядерный университет «МИФИ».
115409, Москва, Каширское ш., 31

ООО «Полиграфический комплекс «Курчатовский».
144000, Московская область, г. Электросталь, ул. Красная, д. 42