



**Учреждение Российской академии наук
Институт физики твердого тела РАН**

Лаборатории Сверхпроводимости, Сектор Нанолитографии.

Практическое пособие по разделу:
**Автоматизация и визуализация физического
эксперимента в среде LabVIEW.**

Василий Сергеевич Столяров

Основные цели практикума.

- *Изучение RS-232 и GPIB интерфейсов. Организация подключения ПК к экспериментальным приборам, имеющим интерфейс RS-232 и/или GPIB, обращение к приборам при помощи терминалов LabVIEW. Типы данных используемые в LabVIEW.*
- *Инструментальный набор, Библиотеки терминалов и функций LabVIEW. Алгоритмы периодического опроса приборов и последовательностей, логические элементы, вывод данных на график и запись данных в текстовый файл, решение уравнений.*
- *Разработка и реализация алгоритмов передачи управляющих команд прибору. Получение вольт-амперных характеристик с реальных физических объектов. Исследования электронно-транспортных свойств по 4х точечному методу. Калибровка термометра.*

Введение.

В настоящее время, любой физический эксперимент управляется и/или контролируется при помощи персональных компьютеров (ПК). Такой симбиоз позволяет проводить эксперименты на новом, высокопроизводительном уровне, с высокой точностью воспроизведения процессов. За счет увеличения производительности ПК, стало возможным, в кратчайшие сроки, снимать и обрабатывать огромные массивы данных полученных с неограниченного числа датчиков, практически мгновенно. Высокие скорости дают возможность разрабатывать новые быстродействующие кибернетические системы.

Целью данного практического курса является прививание молодому поколению понимания, что ПК не только увлекательный мультимедийный «гаджет», но и профессиональная система сбора и обработки данных, способная сильно облегчить нелегкий путь к познанию физических свойств мироздания, а значит и управлению ими.

На протяжении семестра мы рассмотрим несколько основных типов «взаимодействия» экспериментального оборудования (датчиков/передатчиков) с ПК, а также продемонстрируем и отработаем типовые методы автоматизации физических экспериментов на базе наиболее распространенного аппаратно программного комплекса (Laboratry Virtual Instrument Engineering Workbench) LabVIEW.

Среда программирования LabVIEW не похожа на другие языки программирования. С её помощью создается некий виртуальный инструмент, предназначенный для моделирования процессов и управления аппаратными средствами исследования физических объектов. Несомненно, преимуществом LabVIEW является реализация графического программирования, что дает возможность быстрого его изучения, не требующего знаний специальных команд, функций и операторов. По сути, необходимо только разработать алгоритм решения той или иной физической задачи и в виде блок-схемы изобразить его в специальном графическом редакторе LabVIEW. При инсталляции специального драйвера National Instrument DAQ, LabVIEW получает возможность общаться со всеми LPT и COM портами ПК, а также с подключенными к нему GPIB устройствами.

После последовательно, практического изучения методов и средств ведения физических экспериментов, будет предложено проведение полноценных экспериментов таких как: калибровка термометров; исследование электронно-транспортных свойств металлической пленки 4-х точечным методом; снятие вольт-амперных характеристик диода, резистора.

Работа 1.: Типы подключений внешних устройств к ПК, обращение к устройствам в среде LabVIEW.

Цель: Изучить RS-232 и GPIB интерфейсы. Организовать подключение ПК к экспериментальным приборам, имеющим интерфейс RS-232 и/или GPIB, научиться обращаться к приборам при помощи терминалов LabVIEW. Изучить типы данных используемые в LabVIEW.

Введение.

В настоящее время большое количество экспериментальных (лабораторных) приборов снабжены интерфейсами RS-232 или IEEE-488 (GPIB).

RS-232 — интерфейс передачи информации между двумя устройствами. Информация передается по проводам с уровнями сигналов, отличающимися от стандартных 5 В, для обеспечения большей устойчивости к помехам. Логическому "0" соответствует положительное напряжение (от +5 до +15 В для передатчика), а логической "1" отрицательное (от -5 до -15 В для передатчика). Приемник воспринимает сигналы от +3 до +25 В для логического "0", и от -3 до -25 В для логической "1". Асинхронная передача данных осуществляется с установленной скоростью при синхронизации уровнем сигнала стартового импульса. Интерфейс RS-232-C был разработан для простого применения, однозначно определяемого по его названию: «Интерфейс между терминальным оборудованием и связным оборудованием с обменом по последовательному двоичному коду». Чаще всего используется в промышленном и узкоспециальном оборудовании, встраиваемых устройствах. Иногда присутствует на современных персональных компьютерах.



Строго говоря, RS-232 — это название стандарта (RS - recommended standard - рекомендованный стандарт, 232 - его номер), описывающего интерфейс для соединения компьютера и устройства передачи данных. Стандарт был разработан достаточно давно, тем не менее, большинство людей по-прежнему использует название RS-232, которое накрепко приросло к самому интерфейсу.

И так, интерфейс RS-232 обеспечивает соединение двух устройств, одно из которых называется DTE (Data Terminal Equipment) — ООД (Оконечное Оборудование Данных), второе — DCE (Data Communications Equipment) — ОПД (Оборудование Передачи Данных).

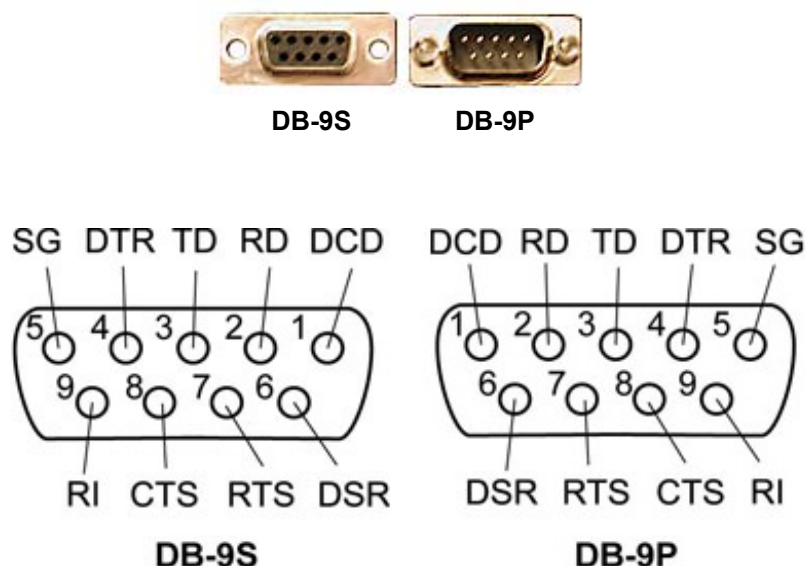
Как правило, DTE (ООД) — это компьютер, а DCE (ОПД) — это физический прибор, хотя RS-232 использовался и для подключения к компьютеру периферийных устройств (мышь, принтер, модем), и для соединения с другим компьютером или контроллером.

Важно запомнить эти обозначения (DTE и DCE). Они используются в названиях сигналов интерфейса и помогают разобраться с описанием конкретной реализации.

Типы разъемов.

Изначально стандарт описывал применение 25-контактного соединителя, типа DB25. DTE-устройство должно оснащаться вилкой (male - "папа"), DCE-устройство — розеткой (female - "мама"). Позднее, с появлением IBM PC, стали использовать усеченный вариант интерфейса и 9-контактные соединители DB9, наиболее распространенные в настоящее время.

Ниже приведен внешний вид и описание контактов разъема **DB-9**:



Распайка RS-232

В приведенной ниже таблице показано назначение контактов 9-контактного соединителя DB9. Таблица показывает распайку вилки оборудования обработки данных (DTE), например, ПК. Розетка устройства передачи данных (DCE) распаяна так, что два разъема стыкуются напрямую, или через кабель, распаянный "контакт в контакт".

Контакт Направление передачи и название сигнала

Контакт DB9	Обозначение	Описание	IN/OUT
1	DCD	Data Carrier Detected - обнаружение несущей	IN
2	RD	Receive Data - принимаемые данные	IN
3	TD	Transmit Data - передаваемые данные	OUT
4	DTR	Data Terminal Ready - готовность терминала	OUT
5	SG	Signal Ground - сигнальная земля	-
6	DSR	Data Set Ready - готовность данных	IN
7	RTS	Request To Send - запрос на передачу	OUT
8	CTS	Clear To Send - готовность к приему	IN
9	RI	Ring Indicator - индикатор вызова	IN

Вилка обозначается как **DB-9P**, а розетка - как **DB-9S**.

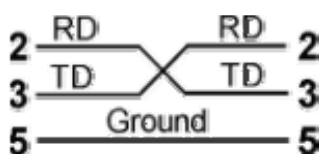
На COM - портах компьютера установлены, как правило, разъемы DB-9P. На физических приборах может оказаться разъем любого типа DB-9S или DB-9P. Обычно разводку указывают в описании к прибору.

Для передачи данных предназначены цепи RD и TD. Остальные цепи предназначены для индикации состояния устройств (DTR, DSR), управления передачей (RTS, CTS) и индикации состояния линии (CD, RI). Полный набор цепей используется только для подключения к ПК внешнего модема. В остальных случаях, например при подключении к ПК промышленного контроллера, используется ограниченный набор цепей, зависящий от аппаратной и программной реализации стыка в контроллере.

Схема кабеля RS-232.

Как было сказано выше, для соединения строго соответствующих стандарту устройств DTE и DCE нужен кабель "контакт в контакт". Для соединения двух DTE-устройств используют так называемые нуль-модемные кабели, в которых провода "перекрещиваются" в соответствии с назначением сигналов. На практике перед распайкой кабеля всегда следует разобраться с документацией на оба соединяемых устройства.

Для соединения многих устройств достаточно минимального набора цепей интерфейса RS-232: RD, TD и Signal Ground. Вот, например, схема кабеля для соединения ПК и ВЧ-Генератора CESAR 600Вт, на соединителях DB9:



Остальные цепи интерфейса в данном подключении не используются.

Длина и провод.

Стандарт определяет максимальную длину кабеля в 50 футов (примерно 15 метров) при скорости 9600 бит/с. На практике устойчивая работа может быть достигнута и при большей длине кабеля. Утверждают, что можно удвоить указанную цифру при использовании неэкранированного кабеля и упятерить ее для экранированного кабеля, а при понижении скорости вдвое предельная длина может быть увеличена примерно вдвое. Тем не менее, мы не можем ручаться за это утверждение, из-за различного уровня внешних электромагнитных помех в каждом конкретном случае.

Рекомендуется использовать кабели на основе витой пары, где каждый из сигнальных проводов свит с общим проводом. Например, для этой цели хорошо подходит кабель для прокладки локальной сети Ethernet на неэкранированных витых парах (Unshielded Twisted Pair - UTP), а лучше — на экранированных - STP. Экран кабеля рекомендуется не объединять с сигнальным общим, а подключить к металлической оболочке разъема.

Технические подробности.

Уровни сигналов

Все сигналы в интерфейсе потенциальные, с номинальными уровнями +12В и -12В относительно общего провода (Signal Ground). Логической единице соответствует уровень -12В, логическому нулю соответствует +12В.

Передача данных

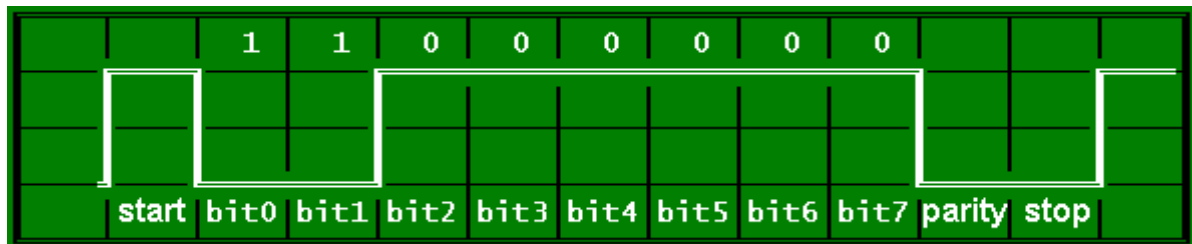
RS-232 называют последовательным интерфейсом, поскольку поток данных передается по одному проводу бит за битом. В отсутствие передачи данных линия находится в состоянии логической единицы (-12В). Скорость передачи данных стандартом не нормируется, но обычно выбирают из ряда 110, 300, 600, 1200, 2400, 4800, 9600, 19200 бит в секунду. В основном используется асинхронный режим работы, при котором данные передаются фреймами. Каждый фрейм состоит из стартового бита, битов данных, бита контроля четности (может отсутствовать), стопового бита. Биты байта данных передаются "хвостом вперед", начиная с младшего бита.

Для правильной стыковки приемопередатчики на обоих устройствах должны быть запрограммированы одинаковым образом, т.е. должны совпадать скорость, количество битов данных (7 или 8), тип контроля по четности (см. ниже), длина стопового бита (1, 1.5 или 2).

При точных расчётах времени на передачу массива байтов наряду с битами данных следует учитывать все служебные биты.

Осциллограмма

Ниже приведена "осциллограмма" одного фрейма при следующих настройках: 8 битов данных, контроль по нечетности (parity odd), 1 стоповый бит:



Стартовый бит всегда идет уровнем логического нуля, стоповый — единицей. Состояние бита паритета определяется настройкой передатчика. Бит дополняет число единичных битов данных до нечетности (parity odd), четности (parity even), может не использоваться (parity none), быть всегда единицей (mark) или нулем (space).

Перспективы

На самом деле перспектив у RS-232 нет. В настоящее время появляется всё больше компьютеров, не оснащенных этим интерфейсом. Однако в эксплуатации находится большое число устройств с интерфейсом RS-232. Для стыковки ПК с такими устройствами используют переходники USB - RS-232. После подключения такого

переходника и установки драйверов в ПК появляется виртуальный COM-порт, через который можно общаться с устройством.



IEEE-488 — спецификация международного стандарта, описывающая интерфейс подключения к шине цифровых измерительных приборов. Создана HP в конце 1960-х для использования в оборудовании для автоматических измерений (англ. *automated test equipment*, АТЕ) под названием интерфейсная шина Hewlett-Packard (англ. *Hewlett-Packard Interface Bus*, HP-IB), в 1975 стандартизирована американским Институтом инженеров электротехнической и электронной промышленности (IEEE) IEEE-488 (по номеру стандарта), и до сих пор применяется в данном качестве. IEEE-488 также известна под названием Интерфейсная шина общего назначения (англ. *General Purpose Interface Bus*, GPIB), стандарт IEC-625 (МЭК625.1), а также другими названиями. В аналогичном советском (русском) стандарте, ГОСТ 26.003-80 Система интерфейса для измерительных устройств с байт-последовательным, бит-параллельным обменом информацией. Требования к совместимости, называется [многопроходным] магистральным каналом общего пользования (КОП).

Каждое устройство на шине имеет уникальный пятибитный первичный адрес в диапазоне от 0 до 30 (таким образом, возможное количество устройств — 31). Адреса устройств не обязаны быть непрерывными, но во избежание конфликтов обязаны быть различными. Стандарт позволяет подключить до 15 устройств к одной двадцатиметровой физической шине используя для наращивания соединители цепочечного типа.

Определено три различных типа устройств, которые могут быть подключены к шине: «listener» (слушатель), «talker» (спикер) и/или контроллер (точнее, устройства могут находиться в состоянии «listener» либо «talker» либо быть типа «контроллер»). Устройство в состоянии «listener» считывает сообщения с шины; устройство в состоянии «talker» посылает сообщения на шину. В каждый конкретный момент времени в состоянии «talker» может быть одно и только одно устройство, в то время как в состоянии «listener» может быть произвольное количество устройств. Контроллер выполняет функции арбитра и определяет, какие из устройств в данный момент находятся в состоянии «talker» и «listener». К шине может быть одновременно подключено несколько контроллеров. В этом случае один из контроллеров (как правило, расположенный на интерфейсной карте GPIB) является ответственным контроллером (Controller-in-Charge, CIC) и делегирует по мере надобности свои функции другим контроллерам. Элемент управления и функции передачи данных логически отдельные; диспетчер может обратиться к одному устройству как «болтуну» (*talker*) и к одному или более устройствам как к «слушателям» (*listeners*) без необходимости участвовать в передаче данных. Это даёт возможность совместно использовать одну и ту же шину для множества контроллеров. В любое данное время, только одно шинное устройство может быть активно как ответственный контроллер.

Данные передаются по шине во время трёхфазной процедуры установления соединения *готовность / доступность / приём*, логике в которой самое медленное участвующее устройство определяет скорость транзакции. Максимальная скорость передачи данных составляла 1 МБ/сек в оригинальном издании стандарта и была увеличена до 8 МБ/сек в расширениях стандарта.

Электрически IEEE-488 восьмибитная параллельная шина, содержащая шестнадцать сигнальных линий (восемь двусторонних используются для передачи данных, три — для установки соединения, пять — для управления шиной) плюс восемь — обратные провода для земли.

Все сигнальные линии используют отрицательную логику: наибольшее положительное напряжение интерпретируется как логический «0», а наибольшее отрицательное — как логическая «1». Линии данных (DIO) пронумерованы от 1 до 8, а **линии данных (ЛД)** в ГОСТ от 0 до 7.

Пять линий управления интерфейсом сообщают устройствам, присоединенным к шине, какие действия предпринимать, в каком режиме находиться и как реагировать на команды GPIB.

Команды

Команды GPIB всегда передаются с использованием классического протокола IEEE-488.1. Стандарт задает формат команд, посылаемых инструментам, и формат и кодировку откликов. Команды, как правило, являются аббревиатурами соответствующих слов английского языка. Команды-запросы снабжаются на конце вопросительным знаком. Все обязательные команды префиксируются астериском (*). Стандарт определяет минимальный набор возможностей, которыми должен обладать каждый инструмент, а именно: принимать и передавать данные, посылать запрос на обслуживание и реагировать на сигнал «Очистить Интерфейс». Все команды и большинство данных используют 7-битный набор [ASCII](#), в котором 8 бит не используется или используется для четности.

Для получения информации от устройств, подключенных к шине, и переконфигурации шины контроллер посылает команды пяти классов: «Uniline» («однобитная»), «Universal Multiline» («многобитная общего назначения»), «Address Multiline» («многобитная адресная»), «Talk Address Group Multiline» («многобитная групповая адресная передающая») и «Listen Address Group Multiline» («многобитная групповая адресная приемная»). <http://www.ni.com/pdf/manuals/370428c.pdf> полное описание по IEEE 488

Интерфейсы RS-232 и GPIB используют набор драйверов для устройств, с помощью которых осуществляется взаимодействие данного устройства с компьютером на все более высоком уровне, заканчивая вашей программой в LabVIEW. Эти драйверы входят в комплект поставки LabVIEW, а также их можно скачать с сайта National Instruments по ссылке <http://ni.com/support/gpib/versions.htm>. При работе в LabVIEW с любым оборудованием GPIB нужно следить, чтобы была установлена последняя версия драйверов. (это предотвратит многие проблемы).

Программирование на языке LabVIEW происходит в поле двух синхронизированных виртуальных окон, в которых закладывается тело программы.

Передняя панель (Front panel) — предназначена для создания внешнего образа виртуального инструмента. Эта панель может содержать различные циферблаты, отображающие исследуемые или вводимые показания, графики всевозможных зависимостей, переключатели, регуляторы сигналов, цветовые решения для внешнего вида инструмента.

Блок диаграмм (Block Diagram) — на этой панели располагаются синхронизированные с элементами на передней панели, терминалы. Это пиктограммы

представляющие собой разъемы для соединения с другими инструментами и функциями, в любых разрешенных комбинациях.

Для того, чтобы внести изменения на «Передней панели» необходимо вызвать Window\Tools Palette и Window\Controls Palette, после чего появятся панели с набором графических инструментов и со всевозможными видами визуализаторов терминалов соответственно.

При работе с панелью «Блок диаграмм» необходимо вызвать Window\Tools Palette и Window\Function Palette. Наиболее полный набор функций и терминалов находится в окне Window\Function Palette\All Function, здесь можно найти всевозможные функции и операции для преобразования, сравнения, перемещения, считывания, передачи и сохранения, исследуемого массива данных.

Для того чтобы опрашивать приборы используются специальные терминалы (VISA Read для COM или Receive для GPIB устройств), которые считывают число байтов с прибора или интерфейса используемого этими терминалами, а затем записывают эти данные в буфер для дальнейшей работы. Для управления приборами используются терминалы VISA Write или Send для COM и GPIB соответственно.

После того как иконка терминала оказалась в окне диаграмм, при наведении курсора мыши на нее, можно увидеть разноцветные каналы выходящие из терминала, это каналы, по которым терминал получает установочные параметры и передает или принимает данные. Канал с названием VISA resource name предназначен для определения источника данных для COM устройств, а канал address определяет статический адрес устройства подключенного по GPIB.

Выполнение работы

Во-первых, необходимо проверить наличие необходимых драйверов NI для общения программы LabVIEW с COM портами ПК и с установленными в него GPIB платами сделать «Print Screen» и вставит картинку в отчет. Для этого необходимо установить и загрузить программу «Measurement & Automation» и изучить директорию My System\Device and Interfaces\... Затем убедиться в подключении прибора к одному из COM портов или GPIB устройству ПК. Описать в отчете тип подключения, номер порта (или название устройства), распайку контактной группы для данного вида подключения со схематическим изображением.

Следующая задача будет направлена на изучение терминалов общения LabVIEW, с прибором. Для этого при входе в среду LabVIEW необходимо вызвать редактор виртуальных инструментов (New Virtual Instruments (VI)). После чего, в окне Block Diagram\...\All Functions\Structures выбрать необходимую для работы структуру программы. После выбора структуры нужно вызвать окно ...\All Functions\Instrument I/O\VISA в случае работы с COM портами или ...\All Functions\Instrument I/O\GPIB\GPIB 488.2 в случае работы с GPIB устройствами.

Для того чтобы опросить прибор используется терминал «VISA Read» для COM портов или Receive для GPIB. Перенесите нужный терминал в окно диаграмм и установите канал «VISA resource name» (или адрес («address») опрашиваемого прибора для Receive терминала). Для этого правой кнопкой кликните по нему и выберете в выпадающем меню Create\Constant, затем выберите из появившегося списка источник соответствующий вашему подключению или впишите руками адрес GPIB прибора. Для VISA Read и Receive необходимо также устанавливать количество считываемых байт, а к каналу «read buffer» (или «data string» для GPIB) применить в выпадающем меню Create\Indicator. Для отображения полученных данных. Оформить переднюю панель (Front panel) на свой вкус (не повторяться). Каждый шаг подробно описать в отчете с использованием «Print Screen» изображений.

После изучения операции считывания данных с приборов, нужно вызвать терминалы управления приборами «VISA Write» или «Send». Используя опыт работы с терминалами чтения данных, вывести на переднюю панель окна для ввода команд в буфер и имени ресурса. Используя структуру Stacked Sequence установить последовательность работы с прибором – «записал - считал- записал- считал» команды использовать в соответствии с вариантом. Все окна ввода и индикации подписать, оформить в индивидуальный, презентабельный вид.

Варианты

1	1. Установить число считываемых бит 5 2. Отправить команду «READ?» на устройство, используя переднюю панель. 3. Используя терминалы ... \All Functions\String\String/Number Conversion\... вставить в десятичную строку данных число полученное при решении уравнения $y=kx+b$ на позицию 10 вывести окна ввода индикации на переднюю панель. (k, x, b- переменные задаваемые во фронтальном окне)	
2	1. Установить число считываемых бит 8 2. Отправить команду «READ:VDC» на устройство, используя переднюю панель. 3. Используя терминалы ... \All Functions\String\String/Number Conversion\... вставить в строку шириной 10 число полученное при решении уравнения $y=x^2$ с точностью до 3х знаков после запятой. (x- переменная задаваемая во фронтальном окне)	
3	1. Установить число считываемых бит 7 2. Отправить команду «Measurement ON» на устройство используя переднюю панель. 3. Используя терминалы ... \All Functions\String\... записать из строки данных «Current 4.567 mA» число в подстроку.	
4	1. Установить число считываемых бит 1 2. Отправить команду «Measurement OFF» на устройство, используя переднюю панель. 3. Используя терминалы ... \All Functions\String\... записать число 3.765A в строку данных «Current mA».	
5	1. Установить число считываемых бит 4 2. Отправить команду «SET:ADR 15» на устройство, используя переднюю панель. 3. Используя терминалы ... \All Functions\String\... записать из строки данных «Voltage 7.678 mV» число в Вольтах, в отдельную подстроку.	
6	1. Установить число считываемых бит 2 2. Отправить команду «SET:Source:Current» на устройство, используя переднюю панель. 3. Используя терминалы ... \All Functions\String\... записать значения 1.2A и 2.56B в строку данных «Current mA, VoltagemV».	
7	1. Установить число считываемых бит 0 2. Отправить команду «RES*» на устройство, используя переднюю панель. 3. Используя терминалы ... \All Functions\String\... записать из строки данных «Voltage 12 mV» первое слово, в отдельную подстроку.	

Используйте Help в выпадающем меню.

Каждый шаг подробно описать в отчете с использованием «Print Screen» изображений.

В заключении указать вид подключения, имя ресурса и операции, которые проводились с приборами и программой.

Работа 2. Периодический опрос прибора в LabVIEW.

Цель работы: разработать и реализовать алгоритм периодического опроса прибора n раз с выводом данных на график и с записью в текстовый файл (реализовать алгоритмы для n -бесконечности и n -конечно).

Введение.

Наиболее часто встречающиеся задачи в физическом эксперименте обычно связаны с изучением меняющегося во времени значения напряжения, тока или сопротивления. Для того чтобы отслеживать всякого рода изменения исследуемой величины, необходимо обеспечить периодическое считывание этих значений.

Перед формированием блок-схемы программы необходимо разработать ее алгоритм, т.е. последовательность производимых операций. Для реализации различных последовательностей, не связанных между собой программные блоки LabVIEW, включают в специальные, подходящие для разработанного алгоритма структуры. В LabVIEW существует 5 типов структур для реализации последовательных и циклических алгоритмов.

Sequence Structure (последовательная структура) – это структура предназначена для пошагового выполнения подпрограмм. Чтобы реализовать последовательное выполнение подпрограмм, их помещают в соответствующие окна –страницы, которым присваивается порядковый номер в последовательности выполнения, начиная от 0. Последняя страница, является стоповой для выполнения всей последовательности.

Case Structure (структура с выбором) – структура имеет два окна, которые управляются внешним логическим элементом. В роле такого элемента может выступать обычный переключатель на передней панели или результат сравнения численных значений. Т.е. выполняется либо подпрограмма, помещенная в первое окно, либо другая, помещенная во второе окно.

For Loop (цикл определенный) – циклическое выполнение подпрограммы конечное количество раз.

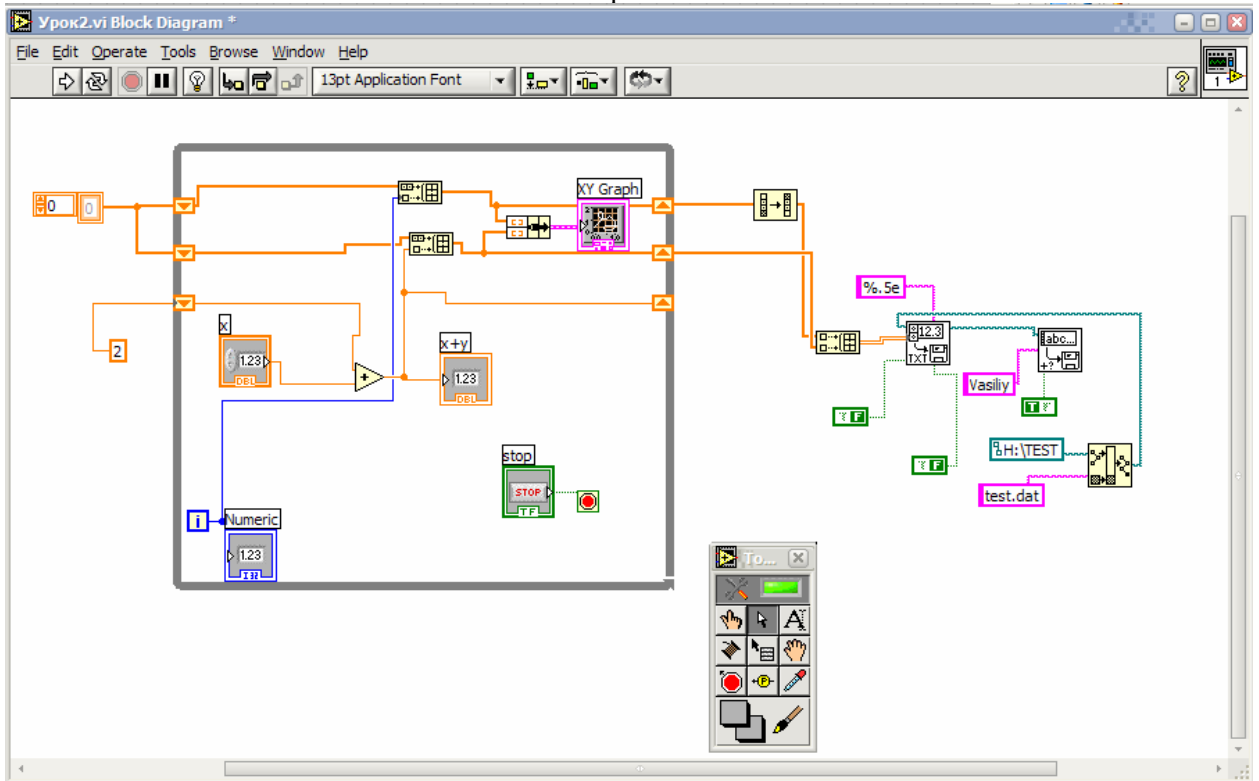
While Loop (цикл по условию) – это оператор цикла, в котором количество итераций заранее не известно, а имеется дополнительное условие выхода из цикла.

Formula Node (формульный узел) – поле для ввода формулы которая будет вычислена по определенным правилам.

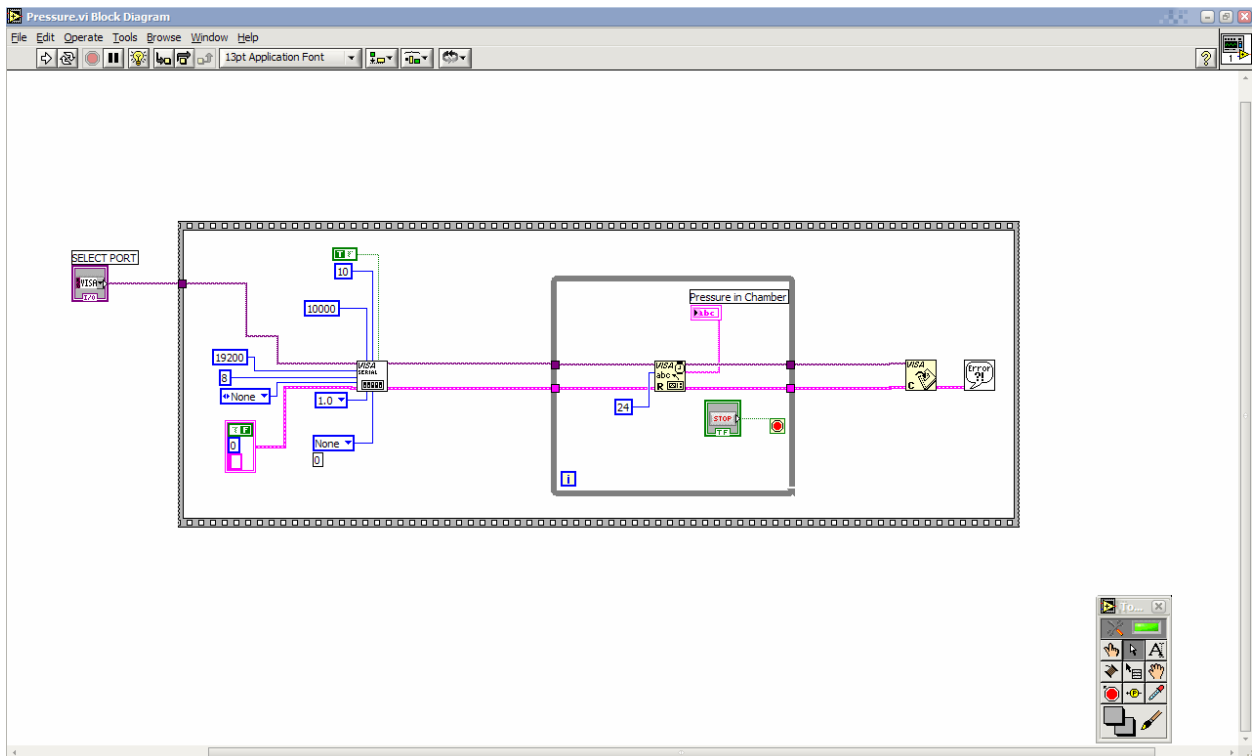
После того, как программная последовательность сбора данных реализована, возникает необходимость вывода данных в виде графика и записи их в соответствующий *.txt файл, для дальнейшей численной обработки. Для записи используется набор функций `Diagram\All Functions\File I/O...`, который позволяют создать текстовый файл данных в нужной папке и записать в него 2D или 1D массив данных, а также добавить необходимую текстовую информацию. Для графической визуализации результатов в реальном времени используется выпадающее меню при нажатии правой кнопки мыши на передней панели `Controls\Graph Indicators...`, после выбора необходимого макета графического представления в блоке диаграмм появится соответствующий терминал. Для подготовки одномерных (1D) массивов данных к выводу на графический терминал используются функции работы с кластерами `Diagram\All Functions\Cluster...`, при помощи которых 1D массивы загоняются в пакеты.

При необходимости установки задержек, при опросе приборов или моделировании используют терминалы `Diagram\All Functions\Time & Dialog...`

Пример реализации оператора **While Loop** с выводом данных на график и записью в файл.



Пример реализации оператора **While Loop** для опроса адреса.



Выполнение работы

Используя структуры **While Loop** или **For Loop** (Diagram\All Functions\Structures), Выполнить расчет уравнений соответствующих вашему варианту, результаты расчета и количество выполненных шагов выводить на переднюю панель. Обеспечить цикл функцией «пауза», позволяющей остановить расчет в любой момент, а затем продолжить выполнение программы. Для запуска обновляемых в каждом следующем шагу данных в цикл, необходимо установить в выпадающем на узлах связи меню (правой кнопкой мыши) «Replace with shift register». Результаты решения уравнений вывести в виде 2D графика. Для этого, полученные значения необходимо записать в одномерный массив (с установленной «нулевой» размерностью), используя функцию ... \All Functions\Array\Build Array..., а затем объединить в двумерный массив значения координат по осям (x, y) используя функцию... \All Functions\Cluster\Build Cluster Array.... полученные результаты записать в *.txt файл. В файле конце текста данных вставить свои ФИО и дату.

По аналогии разработать цикл опроса прибора, заменив модельное уравнение, на терминалы VISA Read и Resave.

Оформить отчет с поэтапными «Print Screen» изображениями программ и описанием операций.

Варианты

1	1. Установить количество опросов (циклов) $n=10$ 2. Установить паузу между циклами $t=1$ сек. 3. Уравнение для моделирования: $y=x$, если $y>150$ остановить счет 4. Заменить уравнение на терминал опроса прибора из указанного адреса и вывести из строки полученных данных 3 элемент.	
2	1. Установить количество опросов (циклов) не больше $n=20$, если больше остановить. 2. Установить паузу между циклами $t=2$ сек. 3. Уравнение для моделирования: $y=3x$. 4. Заменить уравнение на терминал опроса прибора из указанного адреса, вывести на график и записать в файл из строки полученных данных 4 элемент	
3	1. Установить количество опросов (циклов) не больше $n=23$, если больше остановить. 2. Установить паузу между циклами $t=1.5$ сек. 3. Уравнение для моделирования: $y=4x$, если $x>40$ остановить счет. 4. Заменить уравнение на терминал опроса прибора из указанного адреса, вывести на график и записать в файл из строки полученных данных 5 элемент	
4	1. Установить количество опросов (циклов) не больше $n=45$. 2. Установить паузу между циклами $t=1.3$ сек. 3. Уравнение для моделирования: $y=x+2$, x - переменные $z>33$ остановить счет. 4. Заменить уравнение на терминал опроса прибора из указанного адреса, вывести на график и записать в файл из строки полученных данных 6 элемент	
5	1. Установить количество опросов (циклов) не больше $n=50$. 2. Установить паузу между циклами $t=0.1$ сек. 3. Уравнение для моделирования: $y=2x+3$, x - переменные $x>100$ остановить счет. 4. Заменить уравнение на терминал опроса прибора из указанного адреса, вывести на график и записать в файл из строки полученных данных 7 элемент	
6	1. Установить количество опросов (циклов) не больше $n=50$. 2. Установить паузу между циклами $t=0.3$ сек. 3. Уравнение для моделирования: $y=x^3$, x - переменные $y>100$ остановить счет. 4. Заменить уравнение на терминал опроса прибора из указанного адреса, вывести на график и записать в файл из строки полученных данных 8 элемент	

7	<ol style="list-style-type: none">1. Установить количество опросов (циклов) не больше $n=50$.2. Установить паузу между циклами $t=0.8$сек.3. Уравнение для моделирования: $y=x^3$, x - переменные $y>100$ остановить счет.4. Заменить уравнение на терминал опроса прибора из указанного адреса, вывести на график и записать в файл из строки полученных данных 9 элемент.	
---	--	--

Каждый шаг подробно описать в отчете с использованием «Print Screen» изображений.

В заключении указать вид подключения, имя ресурса и операции, которые проводились с приборами и программой. Описать алгоритм.

Работа 3. Решение заданного уравнения в среде LabVIEW и вывод данных в графическом виде.

Цель работы: научиться решать уравнения в среде LabVIEW с последующим поточечным выводом информации на график функции и записью в текстовый файл.. (стр 246)

Введение.

Данная работа направлена на получение навыков, позволяющих создавать виртуальные приборы со сложным взаимодействием отдельных подпрограмм. В ходе выполнения предстоит научиться создавать *.vi модули, выполняющие заданный алгоритм. Использование таких модулей позволит облегчить и ускорить написание более сложных алгоритмов и на каждом уровне иерархии, структура программы может быть понятной и простой.

Работа будет выполняться на примере решения сложного математического выражения, результаты которого должны быть записаны в файл при помощи предварительно созданного save.vi-файла. Решение уравнения будет происходить при помощи формульного узла (Formula Node). Формульный узел называется так же, как и другие элементы структуры. При вызове на диаграмме устанавливается рамка, внутри которой пишутся формулы некоторого набора. После ввода уравнения в окно узла необходимо кликнуть правой кнопкой мыши на рамку и добавить входную и выходную переменные. Например, для того чтобы построить программу ограничителя сигнала от -1 до +1 в окне формульного узла, необходимо записать

$y=x;$
 $y=(y>1)?1;y;$
 $y=(y<-1)?-1;y;$,

где x и y – входная и выходная переменные соответственно.

Выполнение работы.

Создать *.vi –файл, который содержит уравнение, соответствующее варианту. Для этого необходимо внести в поле формульного узла исследуемую функцию и установить входные и выходные переменные. Затем создать *.vi -файл для записи данных в текстовый файл. Результаты вычислений должны быть выведены графически на XY график.

Оформить отчет с поэтапными «Print Screen» изображениями программ и описанием операций.

Варианты

1	1. Решить уравнение $y=U\cos(wt+f)$ при помощи Formula Node. Посторить графики функций $y(t)$, $y(f)$. 2. Установить нижний и верхний пределы для y -0,5 и + 0,5. U – выполнить в виде отдельной подпрограммы, с коэффициентом усиления управляемым при помощи Pointer Slider. 3. Осуществить возможность изменения переменных в процессе запущенного цикла, не останавливая программу. Обеспечить возможность постановки цикла на паузу с выводом диалогового окна. 3. Записать переменные в виде 5и (y,U,w,t,f) столбиков в файл, название которого должно формироваться от текущего машинного времени ччммсс.txt. В конце столбцов добавить запись с ФИО исполнителей.	
2	1. Решить уравнение $y=tg(wt+f)$ при помощи Formula Node. Посторить графики функций $y(t)$, $y(f)$. 2. Установить предел для y -6, +6. w – выполнить в виде отдельной	

	<p>подпрограммы, как функцию частоты (в Гц), при помощи Pointer Slider.</p> <p>3. Осуществить возможность изменения переменных в процессе запущенного цикла, не останавливая программу. Обеспечить возможность постановки цикла на паузу с выводом диалогового окна.</p> <p>3. Записать переменные в виде 4x (y,w,t,f) столбиков в файл, название которого должно формироваться от текущего машинного времени ччммсс.txt. В конце столбцов добавить запись с ФИО исполнителей.</p>	
3	<p>1. Решить уравнение $y=\sin(x)+kx$ при помощи Formula Node. Посторить графики функций $y(x)$, $y(x,k)$.</p> <p>2. Установить предел для t 0, +1000. для k -5 , +5 ввод k– выполнить в виде отдельной подпрограммы, при помощи контроллера Pointer Slider.</p> <p>3. Осуществить возможность изменения переменных в процессе запущенного цикла, не останавливая программу. Обеспечить возможность постановки цикла на паузу с выводом диалогового окна.</p> <p>3. Записать переменные в виде 3x (y,k,x) столбиков в файл, название которого должно формироваться от текущего машинного времени ччммсс.txt. В конце столбцов добавить запись с ФИО исполнителей.</p>	
4	<p>1. Решить уравнение $y=(a+bx+cx^2+dx^3)^{0.5}$ при помощи Formula Node.</p> <p>2. Установить предел для x 0, +1000. a– выполнить в виде отдельной подпрограммы, при помощи контроллера Pointer Slider.</p> <p>3. Осуществить возможность изменения переменных в процессе запущенного цикла, не останавливая программу. Обеспечить возможность постановки цикла на паузу с выводом диалогового окна.</p> <p>3. Записать переменные в виде би (y, a, b, c, d, x) столбиков в файл, название которого должно формироваться от текущего машинного времени ччммсс.txt. В конце столбцов добавить запись с ФИО исполнителей.</p>	
5	<p>1. Решить уравнение $y=b \sin^2(a x)$ при помощи Formula Node.</p> <p>2. a и b– выполнить в виде отдельной подпрограммы, при помощи контроллера Pointer Slider.</p> <p>3. Осуществить возможность изменения переменных в процессе запущенного цикла, не останавливая программу. Обеспечить возможность постановки цикла на паузу с выводом диалогового окна.</p> <p>3. Записать переменные в виде 3x (z,x,b) столбиков в файл, название которого должно формироваться от текущего машинного времени ччммсс.txt. В конце столбцов добавить запись с ФИО исполнителей.</p>	
6	<p>1. Решить уравнение $y=\sin(wt)\cos(f)+b \cos(wt)\sin(f)$ при помощи Formula Node.</p> <p>2. управление w, t, b, f– выполнить в виде отдельной подпрограммы, при помощи контроллера Pointer Slider.</p> <p>3. Осуществить возможность изменения переменных в процессе запущенного цикла, не останавливая программу. Обеспечить возможность постановки цикла на паузу с выводом диалогового окна.</p> <p>3. Записать переменные в виде 5x (y,a,w,f,b) столбиков в файл, название которого должно формироваться от текущего машинного времени ччммсс.txt. В конце столбцов добавить запись с ФИО исполнителей.</p>	

Каждый шаг подробно описать в отчете с использованием «Print Screen» изображений.

В заключении указать вид подключения, имя ресурса и операции, которые проводились с приборами и программой.

Работа 4.. Управление прибором в среде LabVIEW.

Цель работы: разработать и реализовать алгоритм передачи управляющих команд прибору с выводом данных на график и с записью в текстовый файл.

Введение.

Чаще всего при постановке эксперимента ставится задача об измерении изменяющегося сопротивления во времени, при каком либо внешнем воздействии (принцип работы большинства датчиков), либо управление каким-либо двигателем. Для этого в измерительную схему включают прибор, питающий электрическую цепь, чтобы в последствии, другим прибором, фиксировать изменяющееся напряжение. Разберем пример управления приборами на конкретном источнике постоянного тока KEITHLEY 224, описание которого прилагается к работе.

Прибор будем подключать к PCI плате Agilent 82350B GPIB при помощи кабеля IEEE 488.

Установка платы требует отдельного внимания, так как она не является детищем фирмы производителя LabView. К платам такого рода обязательно прилагается драйвер по аналогии с NI-DAQ. Этот драйвер определяет путь к плате для LV и др. языков программирования.

В нашем случае установка выглядит следующим образом:

1. С диска установить библиотеки Agilent (все по умолчанию, полную конфигурацию).
2. В меню ПУСК –Программы\Agilent IO libraries\ IO configuration\ выбрать 82350B (PCI GPIB card) сделать Auto config.
3. В меню ПУСК\Программы\NI-Measurement & Automation. Затем в закладке Tools\NI VISA\VISA Options\VISA settings \Pasports отметить NiVISAatulip.dll поставит галочку.
4. Сохранить изменения, перезапустить программу.

Теперь GPIB карта, установлена. Вызвать VISA Assistant.

Прежде чем запускать источник тока в циклические алгоритмы, необходимо провести предварительные настройки прибора. Обычно к каждому прибору прилагается инструкция с набором управляющих команд, которыми должен пользоваться экспериментатор для установки начальных значений тока при запуске прибора (например, 0) или для установки исследуемого диапазона напряжений (например, Auto Range) и другие важные настройки. Для этого обычно выбирают **Sequence Structure**. Первое окно последовательности, предназначено для предварительной настройки приборов, а последующие уже непосредственно алгоритм программ с циклами.

Выполнение работы.

Создать новый *.vi файл, в котором написать программу позволяющую вводить значения тока в источник постоянного тока KEITHLEY 224.

Структура программы:

1. Используя алгоритм **Sequence Structure** или другую последовательность, произвести обнуление прибора. Затем, используя описание к прибору KEITHLEY 224 установить внешнее управление и автоматическое масштабирование шкалы. Также выставить необходимое начальное значение тока.
2. Используя **While Loop** осуществить ввод значений тока в прибор с возможностью изменять значение тока с передней панели в реальном времени, и

одновременно выводить в соседнее окно индикатора считанное с прибора значение.

3. Вывести на временные и фазовые графики текущее значение тока. Записать вводимые и считанные данные в файл.

Оформить отчет с поэтапными «Print Screen» изображениями программ и описанием операций.

Варианты

1	Написать программу позволяющую увеличивать значение подаваемого тока на сопротивление с шагом 1 мкА. Начиная с 0А и до 9мА (так, чтобы шаг было можно изменять во время выполнение программы).	
2	Написать программу позволяющую уменьшать значение тока подаваемого на нагрузку от 9мА до 0мкА (так, чтобы шаг было можно изменять во время выполнение программы).	
3	Написать программу, которая автоматически вводит ток с шагом 1мкА до 9мА а затем разворачивает с тем же шагом до -5мА. (так, чтобы шаг было можно изменять во время выполнение программы)	
4	Написать программу позволяющую увеличивать значение подаваемого тока на сопротивление на 1 мкА. Начиная с 73А и до 2мА (так, чтобы шаг было можно изменять во время выполнение программы).	
5	Написать программу позволяющую изменять значение подаваемого тока на сопротивление на 1 мкА в положительную или отрицательную сторону. Начиная с 0А и до +/- 9мА (так, чтобы шаг было можно изменять во время выполнение программы).	
6	Написать программу позволяющую увеличивать значение подаваемого тока на нагрузку. С возможностью разворота развертки тока. Начиная с 0А и до 9мА (так, чтобы шаг было можно изменять во время выполнение программы).	

Каждый шаг подробно описать в отчете с использованием «Print Screen» изображений.

В заключении указать вид подключения, имя ресурса и операции, которые проводились с приборами и программой.

Работа 5. Извлечение данных из текстовых файлов с таблицами и их обработка в среде LabVIEW.

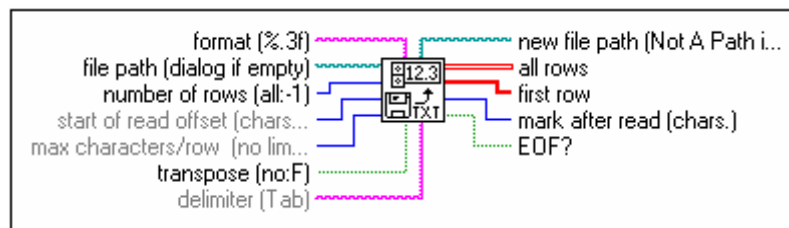
Цель работы: научиться работать с файлами данных в среде LabVIEW с последующим поточечным выводом информации на график функции и записью в текстовый файл.

Введение.

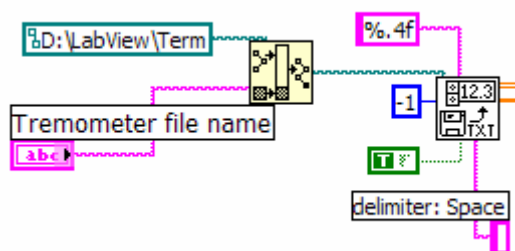
В процессе эксперимента, часто приходится производить операции сравнения измеряемой физической величины с табличными значениями, полученными теоретически или эмпирически (например, калибровочные таблицы данных всевозможных датчиков).

Допустим, у вас имеется полупроводниковый резистор, температурная зависимость сопротивления, которого описывается экспонентой, т.е. при уменьшении температуры оно растет. Такого рода резисторы обычно используют в качестве термометров во время низкотемпературных исследования до 0.3К. Для того что бы, реализовать термометр на полупроводниковом резисторе необходимо его откалибровать. Сделать это, возможно имея доступ к эталонному термометру, калибровка которого известна, или путем численного решения аналитических выражений описывающих зависимость сопротивления от температуры для конкретного резистора. Записав массив данных в виде двух столбцов R[Ом] и T[K] в calibration.txt, встает вопрос, как автоматически сравнивать значения в таблице с измеряемым сопротивлением и сопоставлять его с соответствующей температурой.

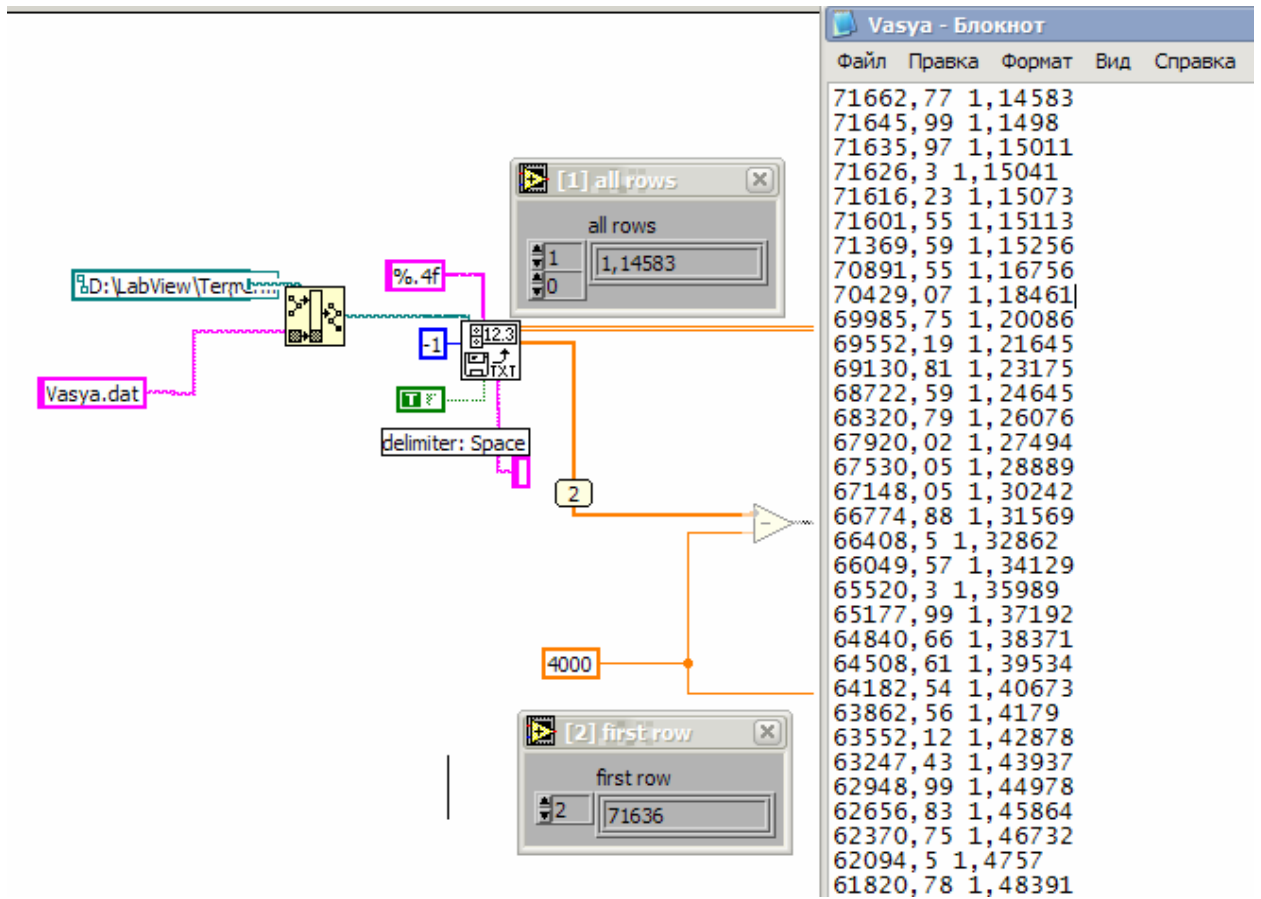
Для этого в LV существует набор специальных терминалов общения с DATA файлами. Используя эти терминалы, возможно, вытащить нужное значение из одного столбца, а затем присвоить ему соответствующее значение другого столбца. В нашем случае лучше всего для работы с нецелыми числами подходит терминал **Read From Spreadsheet File**:



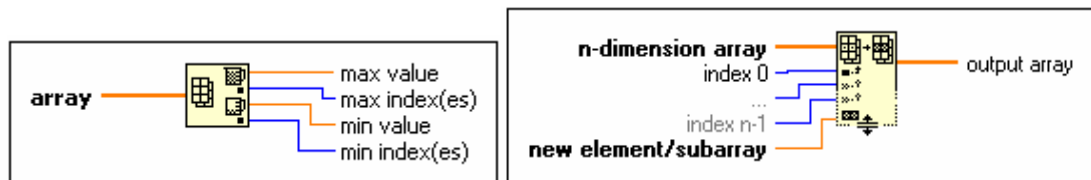
Например, приведем некоторые возможные установки для этого терминала:



При этом на выходе мы будем иметь, два массива. 2D и 1D состоящие из двух и одного столбцов соответственно.

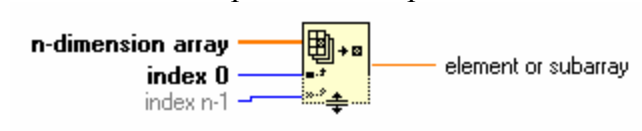


После чего вычитаем реальное, измеренное значение (4000 Ом) из всего массива. Это необходимо, чтобы найти координаты элемента с минимальным по модулю значением в столбце, и соответствующую ему температуру. Для выделения минимального или максимального элементов массива обычно используют оператор **Array Max & Min** а для сдвига можно использовать **Replace Array Subset**:



Также, необходимо произвести усреднение по двум близлежащим значениям (в случае если значение не попало на известное число), для этого необходимо выполнить смещение массива. И повторить поиск минимума.

Выбор элемента из полного массива производит терминал **Index Array**:



Выполнение Работы

Для выполнения задачи воспользоваться терминалами, упомянутыми во введении. Затем вывести на переднюю панель данные из файла в виде 1D и 2D массивов. Найти номер элемента столбца 0 соответствующий данному значению внешнего сигнала произвести сдвиг значения и выполнит расчет по формуле:

$$T = R_{\text{exp}} \frac{T_{\text{min}} - T_{\text{max}}}{R_{\text{max}} - R_{\text{min}}} + \frac{T_{\text{max}} R_{\text{max}} - T_{\text{min}} R_{\text{min}}}{R_{\text{max}} - R_{\text{min}}}$$

Вывести на переднюю панель значение температуры для заданного сопротивления. Сопротивление вводить ручкой потенциометра **Knob**. Калибровочные таблицы брать в соответствии с вариантом.

Оформить отчет с поэтапными «Print Screen» изображениями программ и описанием операций.

Варианты

1	<ol style="list-style-type: none">1. Калибровочный файл 1.dat2. Записать переменные в виде 2x (R_{exp} и T) столбиков в файл, название которого должно формироваться от текущего машинного времени ччммсс.txt. В конце столбцов добавить запись с ФИО исполнителей.3. Дописать в калибровочный файл, по запросу всплывающего меню, ФИО исполнителей.4. Построить калибровочный график. Оформить его в Origin.	
2	<ol style="list-style-type: none">1. Калибровочный файл 2.dat2. Записать переменные в виде 2x (R_{exp} и T) столбиков в файл, название которого должно формироваться от текущего машинного времени ччммсс.txt. В конце столбцов добавить запись с ФИО исполнителей.3. Дописать в калибровочный файл, по запросу всплывающего меню, ФИО исполнителей.4. Построить калибровочный график. Оформить его в Origin.	
3	<ol style="list-style-type: none">1. Калибровочный файл 3.dat2. Записать переменные в виде 2x (R_{exp} и T) столбиков в файл, название которого должно формироваться от текущего машинного времени ччммсс.txt. В конце столбцов добавить запись с ФИО исполнителей.3. Дописать в калибровочный файл, по запросу всплывающего меню, ФИО исполнителей.4. Построить калибровочный график. Оформить его в Origin.	
4	<ol style="list-style-type: none">1. Калибровочный файл 4.dat2. Записать переменные в виде 2x (R_{exp} и T) столбиков в файл, название которого должно формироваться от текущего машинного времени ччммсс.txt. В конце столбцов добавить запись с ФИО исполнителей.3. Дописать в калибровочный файл, по запросу всплывающего меню, ФИО исполнителей.4. Построить калибровочный график. Оформить его в Origin.	
5	<ol style="list-style-type: none">1. Калибровочный файл 5.dat2. Записать переменные в виде 2x (R_{exp} и T) столбиков в файл, название	

	<p>которого должно формироваться от текущего машинного времени ччммсс.txt. В конце столбцов добавить запись с ФИО исполнителей.</p> <p>3. Дописать в калибровочный файл, по запросу всплывающего меню, ФИО исполнителей.</p> <p>4. Построить калибровочный график. Оформить его в Origin.</p>	
6	<p>1. Калибровочный файл 6.dat</p> <p>2. Записать переменные в виде 2х (R_{exp} и T) столбиков в файл, название которого должно формироваться от текущего машинного времени ччммсс.txt. В конце столбцов добавить запись с ФИО исполнителей.</p> <p>3. Дописать в калибровочный файл, по запросу всплывающего меню, ФИО исполнителей.</p> <p>4. Построить калибровочный график. Оформить его в Origin.</p>	

Каждый шаг подробно описать в отчете с использованием «Print Screen» изображений.

В заключении указать вид подключения, имя ресурса и операции, которые проводились с приборами и программой.