

ТЕХНИЧЕСКОЕ ОПИСАНИЕ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ
ИСТОЧНИКА СМЕЩЕНИЯ VIASUNIT
(Rev 1.1)

ЗАО "СКОНТЕЛ"

19 ноября 2010 г.

Оглавление

Введение.	3
Hardware setup.	4
Драйвер.	4
Установка драйвера.	4
Использование программы ControlCenter.exe.	5
Описание элементов управления.	5
Описание пунктов меню.	7
Использование программы Measure.exe.	10
Описание пунктов меню и кнопок панели инструментов.	10
Описание диалогового окна Options.	13
Элемент дерева ресурсов Devices.	13
Элемент дерева ресурсов Windows.	15
Элемент дерева ресурсов Filters.	15
Элемент дерева ресурсов Internal Variables.	16
Элемент дерева ресурсов External Variables.	17
Создание нового рабочего окна.	19
Настройка источника смещения при помощи программы Prefences.exe.	27
Шаг 1 - Инициализация.	27
Шаг 2 - Конфигурирование устройства.	29
Шаг 3 - Калибровка устройства.	30
Калибровка источника смещения.	31
Калибровка печки.	33
Калибровка датчика давления.	33
Калибровка термометра.	34
Шаг 4 - Сохранение данных конфигурации устройства.	35
Пакет компонентов для работы в среде LabView.	36
Использование динамической библиотеки данных biasbox.dll.	38
Описание функций.	38
Описание типов данных и констант.	40
История версий.	42

Введение.

Источник смещения компании «Сконтел» предназначен для управления однофотонными детекторами, а также болометрическими детекторами в составе таких приборов как single photon registration system for visible and near infrared ranges, cryostat-based fast receivers for far and mid-infrared ranges.

Программное обеспечение позволяет управлять источниками смещения, а также мощностью печки, используемой при работе с болометрическим детектором. Помимо этого существует возможность сбора данных (значения тока, напряжения детекторов, давление, температура, напряжение печки). Для управления и сбора данных используется 2 программы: **ControlCenter.exe** и **Measure.exe**. Первая предназначена для управления источником смещения, вторая - для сбора данных и отображения их в графическом виде. Для написания собственных приложений, позволяющих комбинировать разнообразные приборы в комплект программного обеспечения входит пример управления источником смещения выполненный в среде LabView. В комплект программного обеспечения также входит динамическая библиотека **biasbox.dll**, с помощью которой возможно использование источника смещения в таких средах разработки программного обеспечения как VB, VC, C Builder, Delphi и прочее. Помимо этого в комплект программного обеспечения включена программа необходимая для настройки источника смещения: **Prefences.exe**. Представленное техническое описание посвящено подробному описанию установки и использования выше перечисленного программного обеспечения.

Hardware setup.

Драйвер.

Для установки коробки смещения необходимо первоначально скачать последнюю версию драйвера, которую можно найти пройдя по ссылке: <http://www.ftdichip.com/Drivers/D2XX.htm> (и выбрать драйвер FT232R для вашей операционной системы. На январь 2010 года был доступен драйвер версии 2.06.00 для операционных систем Windows Server 2008 R2, Windows 7, Windows 7 x64, Windows Server 2008, Windows Server 2008 x64, Windows Vista, Windows Vista x64, Windows XP, Windows XP x64, Windows 2000, Windows Server 2003, Windows Server 2003 x64. Для других операционных систем программное обеспечение не тестировалось. При использовании других операционных систем о результате работы просьба сообщить производителям по электронной почте scontel@scontel.ru.

Установка драйвера.

Процесс установки драйвера для различных операционных систем подробно описан на странице <http://www.ftdichip.com/Documents/InstallGuides.htm>.

Использование программы ControlCenter.exe.

Для управления и отображения данных с источника смещения необходимы 2 программы: **ControlCenter.exe** и **Measure.exe**. Стоит отметить, что использование программного обеспечения не отключает возможность управления источником смещения с помощью ручек и кнопок, находящихся на самой приборной панели. При запуске программы появляется окно, в котором перечислены все доступные устройства (рис. 1). При появлении этого окна пользователь должен выбрать инструмент, с которым он собирается работать. Одновременно компьютер может работать с несколькими источниками смещения.

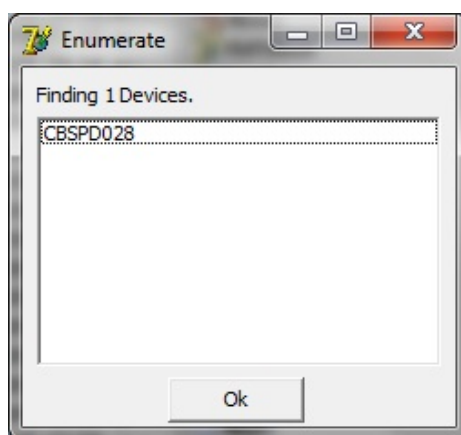


Рис. 1: Окно выбора рабочего инструмента.

Общий вид рабочего окна программы **ControlCenter.exe** представлен на рисунке 2.

Описание элементов управления.

Устройства управления и индикации источником смещения сгруппированы в блоки (**Channel1**, **Channel2**, **P&T**, **Heater**). Изготавливаемый нами источник смещения универсальный и позволяет работать как с болометрическими системами, так и с однофотонными приемными системами. Программа для компьютера также позволяет работать с обоими типами приемных систем. Включение и отключение блоков производится автоматически при запуске программы. Стоит особо отметить, что в случае одноканальных систем используется **Channel2**.

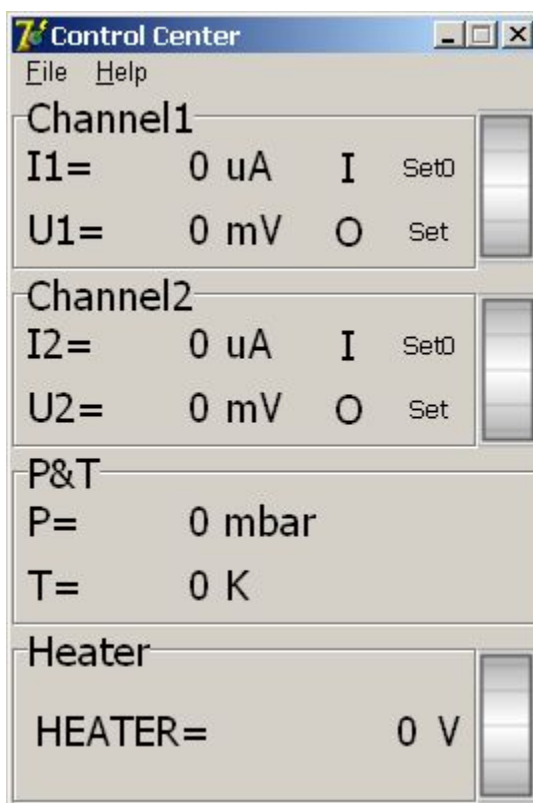


Рис. 2: Общий вид рабочего окна программы **ControlCenter.exe**

Блок управления каналом представляет собой 2 метки, отображающих значение тока и напряжения смещения, а также набор кнопок и колесико прокрутки.

Кнопка **I/U**, предназначена для выбора режима стабилизации источника смещения и одновременно для отображения режима стабилизации (при нажатии на кнопку **I**, режим стабилизации меняется на стабилизацию напряжения, а надпись на кнопке меняется на **U** и наоборот).

Кнопка **O/S** предназначена для включения - снятия закоротки. При этом надпись на кнопке показывает текущее состояние: **O** - соответствует открытому концу источника смещения, **S** - закороченному.

Кнопка **Set0** предназначена для установки нулевой величины смещения. В случае когда выбран режим стабилизации тока величина тока должна сброситься в 0, иначе напряжение будет сброшено в 0. Ноль тока и напряжения могут не совпадать, поскольку сам приемник обладает низкой температурой и различного рода термоэлектрические эффекты могут приводить к постоянному смещению ноля, в любом случае отображаемые на индикаторе значения тока и напряжения соответствуют истинным источником тока и напряжения на выходе источника смещения.

Кнопка **Set** предназначена для установки произвольного значения тока или напряжения в зависимости от выбранного режима стабилизации. При нажатии на эту кнопку появляется диалоговое окно (рис. 3), в котором можно указать величину тока - напряжения и при нажатии на кнопку **ОК**, это значение будет установлено.

Справа от блока управления каналом находится элемент управления - колесико прокрутки. При наведении на него курсора мыши и вращении колесика прокрутки

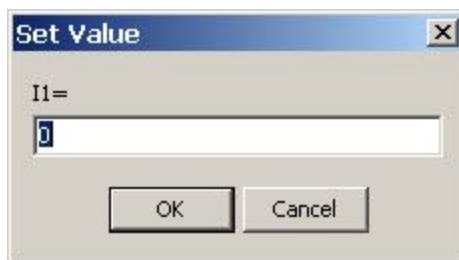


Рис. 3: Окно установки значения тока/напряжения.

мышью значения тока-напряжения (в зависимости от выбранного режима стабилизации) увеличиваются или уменьшаются (в зависимости от направления вращения колесика мыши). При нажатии клавиши клавиатуры **Ctrl** скорость прокрутки увеличивается.

Блок **P&T** предназначен для индикации значений тока и напряжения. Используется этот блок только при работе с однофотонными детекторами для контроля температуры и давления в случае когда происходит откачка паров гелия.

Блок **Heater** предназначен для индикации и управления напряжением печки. Используется только в системах с болометрическим детектором. Управление напряжением производится при помощи колесика прокрутки, расположенного справа от блока **Heater** аналогично тому как это делается для блоков источника смещения.

Описание пунктов меню.

Пункт меню **File>Window** предназначен для выбора программы осуществляющей вывод и накопление данных. При выборе этого пункта меню открывается диалоговое окно (рис. 4). В строке ввода диалогового окна необходимо полностью или частично указать заголовок окна-приемника. Представленный пример предназначен для использования программы совместно с программой **Measure.exe**. Также информацию можно передавать и в свое приложение. Информация о значении тока, напряжения, давления, температуры и напряжения печки передается строковой переменной в виде 'U1=Value'#9'I1=Value'#9'U2=Value'#9'I2=Value'#9'P=Value'#9'T=Value'#9'HEATER=Value'#9'BATP=Value'#9'BATN=Value'#9'Serial_Number:I1=Value'#9'Serial_Number:U1=Value'#9'Serial_Number:I2=Value'#9'Serial_Number:U2=Value'#9'Serial_Number:P=Value'#9'Serial_Number:T=Value'#9'Serial_Number:HEATER=Value'#9'Serial_Number:BATP=Value'#9'Serial_Number:BATN=Value', где Value - значение величины, #9 - разделительный символ (табулятор), Serial_Number - серийный номер устройства.

Ниже приводится пример приема данных в своем приложении, написанный на языке Delphi2009 (стоит помнить, что в этой версии среды программирования строки переведены в формат Unicode).



Рис. 4: Вид окна выбора импортера данных.

```

Uses Windows;
...
TFmMain = class(TForm)
...
Procedure ReceiveMessage (var Msg: TMessage); message
WM_COPYDATA;
...
end
procedure TfmMain.ReceiveMessage (var Msg: TMessage);
var
pcd : PCopyDataStruct;
Pch : Pchar;
Str : string;
Len : integer;
begin
pcd := PCopyDataStruct(Msg.LParam);
Pch := Pchar(pcd.lpData);
Str := Pch;
Len := (pcd.cbData-1) div 2;
delete(str, len+1, length(str)-len);
end;

```

Пункт меню **File>Sweep** предназначен для осуществления развертки тока-напряжения (в зависимости от выбранного режима стабилизации). При нажатии на этот пункт меню возникает диалоговое окно (рис. 5).

При помощи радиокнопок **Channel1** и **Channel2** выбирается канал, на котором будет произведена развертка тока-напряжения. В строке выбора **Start** указывается начальное значение развертываемой величины, в строке **Stop** - конечное, строка **Step** - показывает шаг перестройки. При нажатии на кнопку старт начинается развертка тока-напряжения. При этом диалоговое окно **Sweep** скрывается. Другие элементы управления программы остаются недоступными до окончания процесса развертки. Данная функция предназначена для автоматического построения вольт амперных характеристик.

Пункт меню **File>Exit** предназначен для выхода из программы.

Пункт меню **Help>About** выводит информацию о программе и производителе. Закрывать появившееся окно можно кликнув мышкой на любой точке возникшего окна.

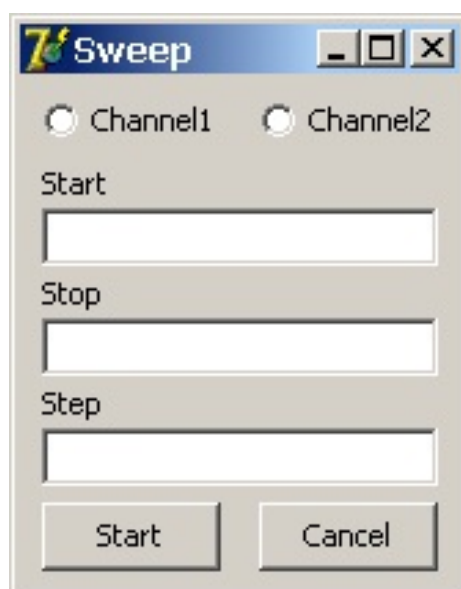



Рис. 5: Вид окна развертки тока/напряжения.


Использование программы Measure.exe.


Программа **Measure.exe** предназначена для сбора данных, отображения в графическом виде и их хранения. Изначально программа была написана для внутреннего пользования для работы с многоканальным вольтметром собственного производства "Corobo4ka". Со временем функциональность программы была расширена и оказалось, что в ряде экспериментов она удобна не только для сбора данных, но и для простейшей обработки получаемых величин. В настоящее время программа подразумевает возможность подсоединения разнообразных приборов. Ведется работа над включением в список поддерживаемых приборов и источника смещения. Рабочее окно программы представлено на рисунке 6.

Программа использует MDI технологию, при которой имеется главное окно программы и дочерние окна (далее будем называть их рабочими окнами программы). По подобной же технологии устроены такие распространенные офисные приложения как MS Word, MS Excel и другие. Материнское окно программы занимается управлением всеми инициализированными приборами, сбором данных и распределением ее между дочерними окнами, которые в свою очередь занимаются хранением и отображением информации.

Описание пунктов меню и кнопок панели инструментов.

Пункт меню **File>New**, а также кнопка  открывают диалоговое окно создания нового рабочего окна. Подробно описание особенностей работы с этим окном описано ниже в разделе "Создание нового рабочего окна".

Пункт меню **File>Open**, а также кнопка  вызывают стандартное диалоговое окно открытия файла. Файл должен иметь расширение ***.v3d**. Данные в файле хранятся в закодированном виде и открыть этот файл возможно лишь программой **Measure.exe**.

Пункт меню **File>Save**, а также кнопка  используются для сохранения данных в формате ***.v3d**. Если файл был открыт ранее, то старый файл будет перезаписан новым. В случае, когда работа идет с абсолютно новым файлом, будет вызван стандартный диалог сохранения данных, в котором пользователю будет предоставлена возможность выбрать директорию для хранения файла и его имя.

Пункт меню **File>Save as...** предназначен для вызова стандартного диалога сохранения данных, в котором пользователю будет предоставлена возможность вы-

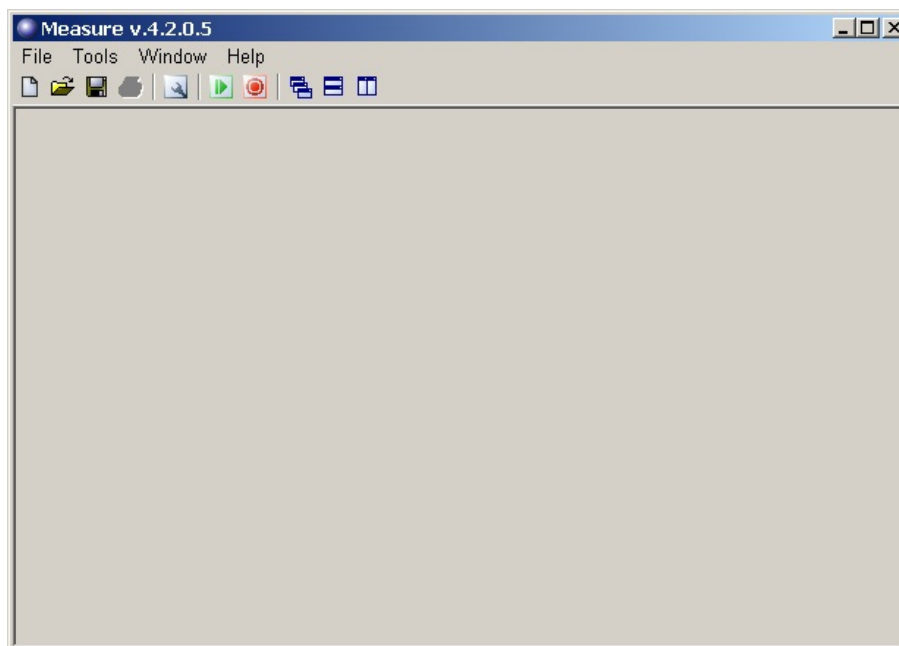






Рис. 6: Основное окно программы.


брать директорию для хранения файла и его имя.


Пункт меню **File>Print** и кнопка  вызывают стандартное диалоговое окно печати на принтере и предназначено для печати активного рабочего окна программы.


Пункт меню **File>Exit** предназначен для выхода из программы. При выходе пользователю будет представлена возможность сохранить данные.

Пункт меню **Tools>Reading**, а также кнопка  инициализируют подключенные устройства и запускают чтение данных в соответствии с настройками подключенных устройств. Повторный выбор пункта меню **Tools>Reading** или нажатие на кнопку  останавливает процесс чтения данных.

Пункт меню **Tools>Recording**, а также кнопка  начинают процесс записи данных. При этом возникает диалоговое окно, показанное на рис. 7. В диалоговом окне предлагается указать название для собираемого набора точек (далее кривой или Curve). По умолчанию предлагается в качестве названия кривой указать текущую дату и время. При нажатии на кнопку **OK** начинается запись данных во все окна, открытых для записи. При этом в этих окнах добавляется кривая с соответствующим названием. Нажатие кнопки **Cancel** отменяет процесс записи данных. Повторный выбор пункта меню **Tools>Recording** или нажатие на кнопку  останавливает процесс записи данных.

Пункт меню **Tools>Options**, а также кнопка  вызывают диалоговое окно **Options** более подробно о возможностях этого окна и работе с ним рассказано в разделе **”Описание диалогового окна Options”**.

Пункт меню **Window>Cascade**, а также кнопка  размещают все открытые рабочие окна каскадом.

Пункт меню **Window>Tile Horizontal**, а также кнопка  размещают все от-

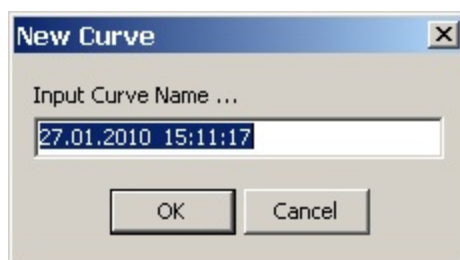



Рис. 7: Окно выбора имени кривой.

крытые рабочие окна деля рабочий экран в горизонтальном направлении.

Пункт меню **Window>Tile Vertical**, а также кнопка  размещают все открытые рабочие окна деля рабочий экран в вертикальном направлении.

Ниже разделителя в пункте меню **Windows** расположены пункты меню, соответствующие названиям всех созданных рабочих окон. При выборе пункта меню, соответствующее ему рабочее окно выйдет на передний план.

Пункт меню **Help>Help Topic** вызывает окно, в котором указаны некоторые особенности с работой программы (на русском языке).

Пункт меню **Help>Update** вызывает диалоговое окно обновления, показанное на рис. 8. При этом программа подключается к Интернету и сравнивает вашу версию программы с последней доступной версией на сайте компании "СКОНТЕЛ", при наличии новой версии появится возможность обновления программы. При нажатии на кнопку **Download**, в каталог, в котором находится **Measure.exe** будет скачана программа обновления. После чего данные собранные программой будут сохранены и основная программа закрыта. После успешной установки программы **Measure.exe** будет запущена вновь со всеми восстановленными данными. Убедительная просьба при обнаружении ошибок в работе программы сообщать об этом авторам по адресу scontel@scontel.ru. Это позволит нам создать более качественное программное обеспечение и обновление, включающее исправленные ошибки выйдет в скором времени.

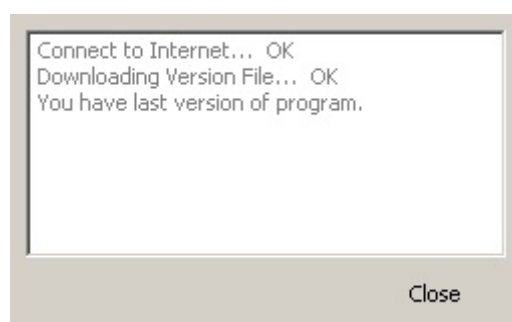



Рис. 8: Окно обновления программы.

Пункт меню **Help>About** вызывает окно, в котором содержится краткая информация об авторе, а также историю версий программы (на русском языке).

Описание диалогового окна Options.

Диалоговое окно **Options** вызывается при выборе пункта меню **Tools>Options** или при нажатии кнопки  на панели инструментов. При первом запуске программы, при условии, что рабочие окна не созданы диалоговое окно **Options** выглядит как показано на рисунке 9.

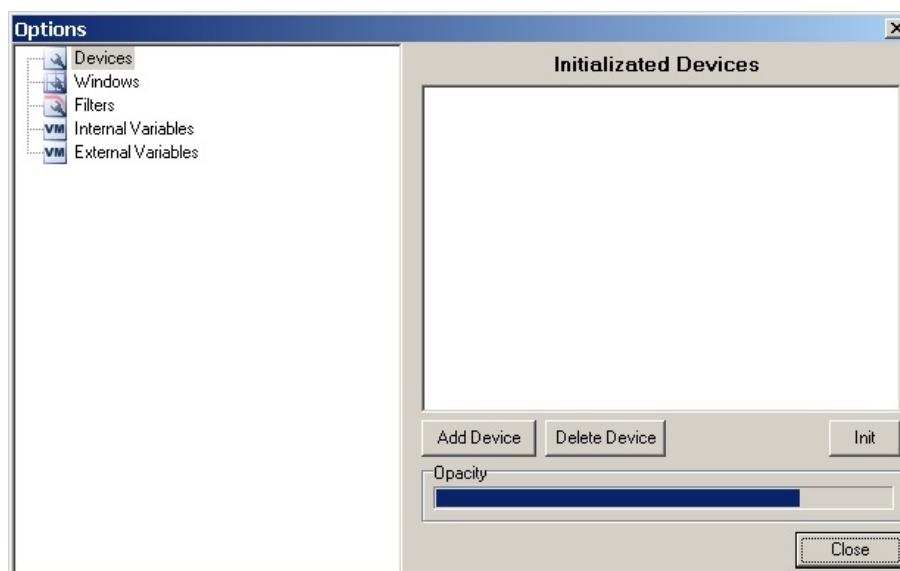


Рис. 9: Внешний вид окна **Options**.

В левой части диалогового окна **Options** располагается дерево ресурсов. Оно представлено 5 элементами: **Devices**, **Windows**, **Filters**, **Internal Variable** и **External Variable**. При выборе любого элемента дерево окна справа меняется.

Элемент дерева ресурсов Devices.

Элемент дерева **Devices** предназначен для добавления и удаления активных устройств. При нажатии на кнопку **Add Device** возникает диалоговое окно выбора устройства, в котором можно выбрать тип устройства, а также серийный номер уже подключенного устройства этого типа. Для версии программы **4.2.0.5** существует возможность добавления только устройств типа **Corobo4ka**. Для более поздних версий программы количество поддерживаемых устройств будет увеличено. При добавлении устройств появится возможность изменения настроек устройства. Устройство при этом отображается в дереве ресурсов в левой части окна. Кнопка **Delete Device** предназначена для удаления выбранного устройства из списка. Полоска прогресса в блоке **Opacity** настраивает прозрачность большинства диалоговых окон, используется в декоративных целях. Ниже описаны некоторые приборы, управление которыми реализовано к моменту написания этой инструкции.

Многоканальный вольтметр "Corobo4ka"

При выборе устройства типа **Corobo4ka** в дереве объектов окно **Options** изменяется на показанное на рис. 10 ниже приведенное. В блоке **Channels** находятся 4 **CheckBox**, в которых выбираются необходимые для работы каналы измерительного устройства. Стоит учесть, что при уменьшении количества работающих каналов частота опроса каждого из каналов увеличивается обратно пропорционально числу работающих каналов. В блоке **Voltmeter&Spectrum** находятся элементы управления позволяющие менять настройки устройства считывания. **Time Constant** - время усреднения для работы вольтметра в режиме самописца. Это время между двумя ближайшими точками любого из каналов. Делать это время очень малым менее 20мс неправильно, поскольку это приведет к существенной нагрузке на компьютер, к тому же точность определения времени гарантированная операционной системой **Windows** порядка 10мс. **Conversion Time** - время между двумя последовательными преобразованиями. Диапазон значений этой величины 82-2600 мкс. Увеличение этой величины приводит к уменьшению шумов преобразования и снижению максимальной частоты при построении спектров. При этом максимальная частота спектра рассчитывается как $F_{max} = 1/(2 * ConversionTime * NChannel)$, где **NChannel** - количество используемых каналов. То есть при включенном одном канале и времени преобразования 1000 мс максимальная частота спектра будет равна 500 Гц.

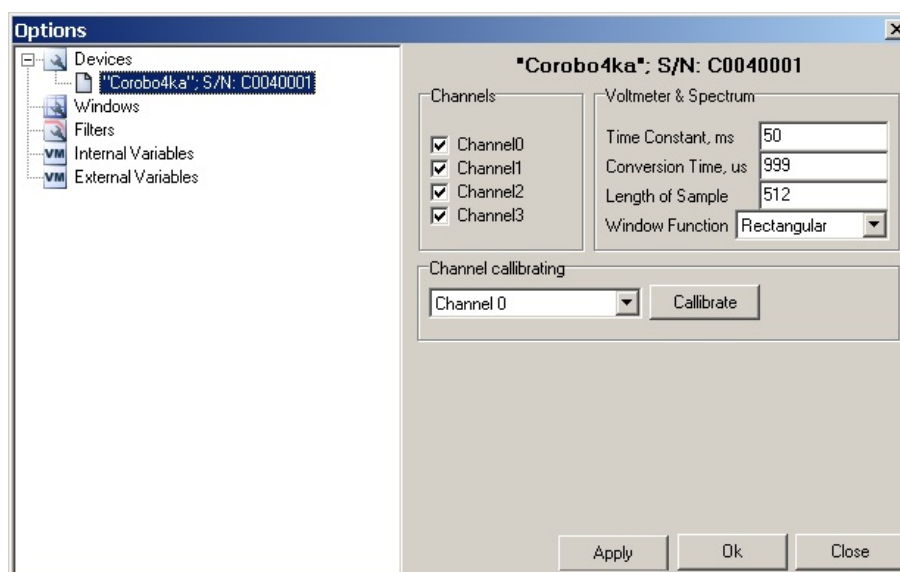


Рис. 10: Окно редактирования свойств устройства **Corobo4ka**.

Блок **Channel calibrating** предназначен для калибровки на ноль каждого из каналов вольтметра. При этом на вход вольтметра необходимо поместить закоротку либо небольшое постоянное сопротивление, в выпадающем списке выбрать соответствующий ему номер канала и нажать на кнопку **Calibrate**. Это приведет к вычислению новых калибровочных коэффициентов, хранящихся в памяти вольтметра и дальнейшему их использованию при работе программы.

Элемент дерева ресурсов Windows.

При выборе в дереве ресурсов элемента **Windows** появится окно с предложением выбрать одно из рабочих окон для редактирования его настроек. Детальное рассмотрение настроек рабочих окон будет проведено в разделе **”Создание нового рабочего окна”**.

Элемент дерева ресурсов Filters.

Этот элемент дерева ресурсов используется только совместно с многоканальным вольтметром **”Corobo4ka”** и предназначен для составления разнообразных цифровых фильтров для обработки входного сигнала. В качестве фильтра могут быть выбраны фильтр низких частот, фильтр высоких частот, полоснозаградительный или полоснопропускающий фильтры Баттерворта, а также любые последовательные комбинации этих фильтров. На рисунке 11 представлен пример окна открывающегося при выборе элемента дерева **Filters**.

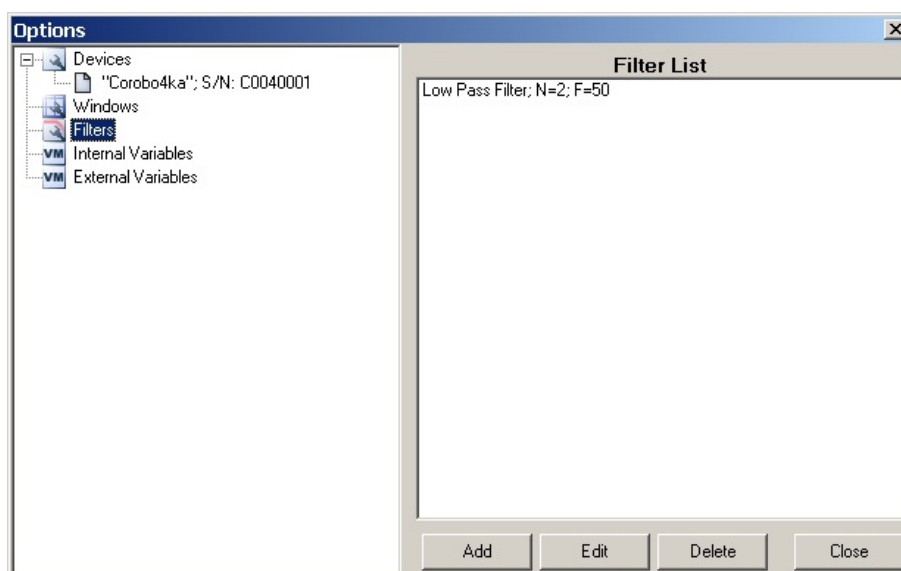


Рис. 11: Окно редактирования свойств фильтров.

В списке указывается описание включенных фильтров. При нажатии на кнопку **Add** возникает диалоговое окно создания фильтра (рис. 12). Пользователю предлагается выбрать тип фильтра, его граничные частоты, на которых коэффициент пропускания фильтра равен -3dB , а также порядок фильтра (необходимо, чтобы порядок фильтра был четным). Следует помнить, что с увеличением порядка фильтра крутизна спада увеличивается, но это приводит к увеличению сложности вычислений. Также при увеличении порядка фильтра повышаются требования к точности расчета коэффициентов фильтра, что может привести к расбалансировке и возбуждению фильтра.

При нажатии на кнопку **Edit** возникает диалоговое окно редактирования, в котором можно изменить параметры выделенного цифрового фильтра. Альтернативой для вызова окна редактирования фильтра служит двойной щелчок курсора мыши на описании фильтра в списке.



Рис. 12: Диалоговое окно добавления фильтра.

При нажатии на кнопку **Delete**, или нажатие клавиши клавиатуры **Del** выделенные фильтры удаляются.

В качестве примера сложного фильтра можно создать 2 полоснозаградительных фильтра со следующими характеристиками: для первого фильтра $F_1 = 45\text{Hz}$, $F_2 = 55\text{Hz}$, $N = 4$, для второго фильтра $F_1 = 95\text{Hz}$, $F_2 = 105\text{Hz}$, $N = 4$. Такой составной фильтр позволяет избавиться от сетевой наводки и ее второй гармоники.

Элемент дерева ресурсов Internal Variables.

При выборе в дереве ресурсов элемента **Internal Variables** появится окно показанное на рис. 13. Внутренние переменные программы используются для произведения простейших действий над всеми входными данными программы, такими как данные подключенных приборов и внешние переменные, передаваемые сторонними приложениями. Более подробно о внешних переменных и методах работы с ними будет рассказано в разделе **”Элемент дерева ресурсов External Variables”**. С внутренними переменными можно производить следующие математические действия: сумма, разность, произведение, деление, возведение в степень и операции со скобками.

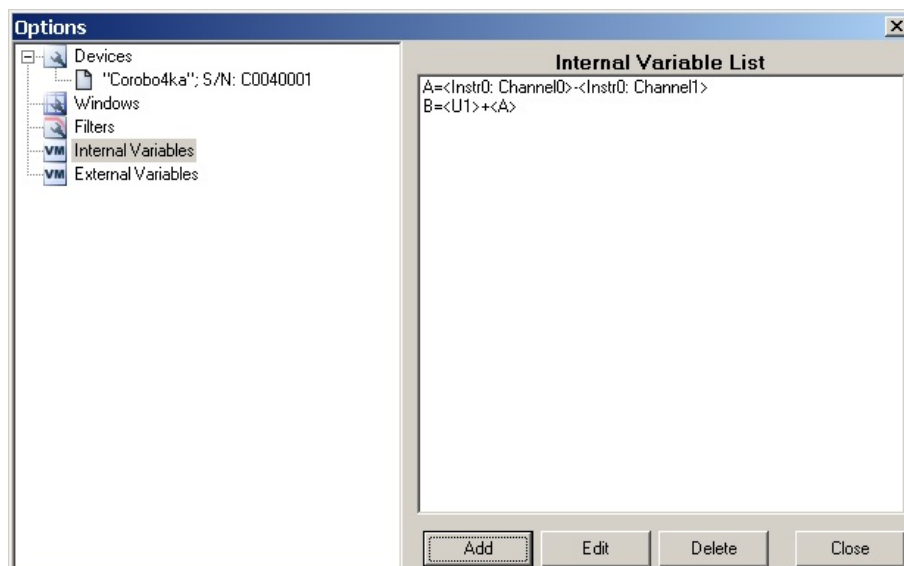


Рис. 13: Окно редактирования внутренних переменных.

При нажатии на кнопку **Add** появляется диалоговое окно создания новой пере-

менной, представленное ниже. Пользователю предоставляется возможность указать имя новой переменной и ее значение. При использовании значений какого либо канала входных данных название переменной должно стоять в скобках <> и должно полностью соответствовать названию переменной как это было бы при написании названия оси рабочего окна программы. Внешние и внутренние переменные указываются без **Ext. Variable** и **Int. Variable**.

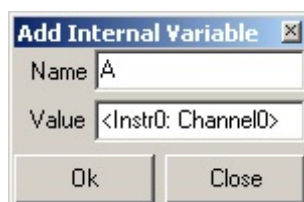


Рис. 14: Окно добавления новой внутренней переменной.

ПРИМЕР:

```
A=<Instr0: Channel1>+<Instr0: Channel2>
B=<A>+1
```

При нажатии на кнопку **Edit** возникает диалоговое окно редактирования, в котором можно изменить как имя так и значение переменной. Альтернативой для вызова окна редактирования переменной служит двойной щелчок курсора мыши на соответствующей строке в списке.

При нажатии на кнопку **Delete**, или нажатие клавиши клавиатуры **Del** выделенные переменные удаляются.

Элемент дерева ресурсов External Variables.

При выборе в дереве ресурсов элемента **External Variables** появится окно показанное на рис. 15. Внешние переменные используются для получения данных от сторонних приложений. Пример такого симбиоза демонстрирует сочетание программы с приложением **ControlCenter.exe**. Программа **ControlCenter.exe** пересылает данные для 9 переменных: I1, U1, I2, U2, HEATER, P, T, BATP, BATN.

При нажатии на кнопку **Add** появляется диалоговое окно создания новой внешней переменной, показанное на рис. 16. Пользователю предоставляется возможность указать имя новой переменной.

При нажатии на кнопку **Edit** возникает диалоговое окно редактирования, в котором можно изменить имя внешней переменной. Альтернативой для вызова окна редактирования внешней переменной служит двойной щелчок курсора мыши на соответствующей строке в списке.

При нажатии на кнопку **Delete**, или нажатие клавиши клавиатуры **Del** выделенные переменные удаляются. Ниже представлен пример того, как можно передать значение переменной из сторонней программы в программу **Measure.exe**. Пример написан с использованием компилятора Delphi2009. При необходимости пересылки нескольких переменных они разделяются символом табуляции.

```

Uses Windows;
...
procedure FindWindow(WindowToExport: string; var Handle: THandle);
var stop:boolean;
nstr: array [0..255] of char;
begin
Handle:=fmMain.Handle;
Stop:=False;
while (not stop) and (Handle<>0) do
begin
Handle:=GetWindow(Handle,GW_HWNDNEXT);
GetWindowText(Handle,nstr,255);
if pos(WindowToExport,strupas(nstr))<>0
then stop:=true
end;
end;
procedure SendData(Name: string; Value:real);
var Hand: THandle;
str : string;
cd : TCopyDataStruct;
begin
FindWindow('Measure',Hand);
if Hand<>0 then
begin
str:=Name+'='+floattostr(Value);
cd.cbData := length(str)*sizeof(char)+1;
cd.lpData := Pchar(str);
SendMessage(Hand, WM_COPYDATA, 0, LParam(@cd));
end;
end;

```

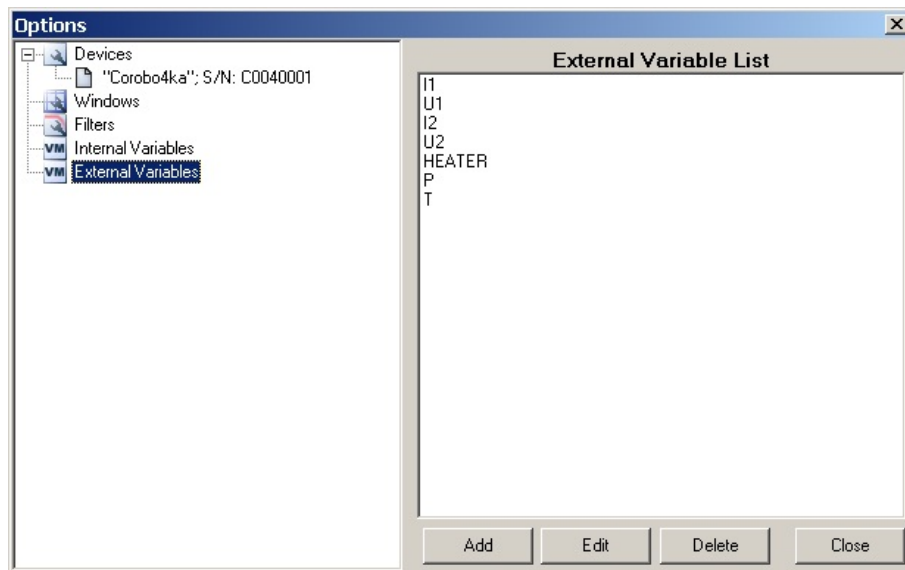


Рис. 15: Окно редактирования внешних переменных.

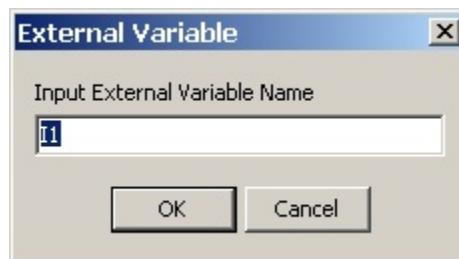



Рис. 16: Окно добавления новой внешней переменной.

Создание нового рабочего окна.

Пункт меню **File>New**, а также кнопка  открывают диалоговое окно создания нового рабочего окна, показанное на рис. 17. В верхней части диалогового окна представлена таблица с основными настройками. В строке **Dimension of Window** указывается тип создаваемого окна. При нажатии на правой части этой строки возникает выпадающий список с возможностью выбора 4 типов создаваемого окна: **2D**, **3D**, **Oscilloscope** и **Spectrum**. При выборе типа **2D** будет создано окно с двумя координатами (X (горизонтальная ось) и Y (вертикальная ось)), при выборе типа окна **3D** будет создано окно с тремя координатами (X (горизонтальная ось), Y (вертикальная ось) и ось Z, направленная перпендикулярно экрану монитора, при этом информация по этой оси отображается цветом), при выборе окна типа **Oscilloscope** будет создано двумерное окно, в котором будет строится быстрая развертка сигнала (работает только с устройством "Corobo4ka"), при выборе окна типа **Spectrum** будет создано двумерное окно, в котором будет строится спектр принимаемого сигнала (работает только с устройством "Corobo4ka"),

Строки **Axis-X**, **Axis-Y** и **Axis-Z** (последняя появляется в случае трехмерного окна) показывают значение какой переменной будет отложено по данной оси. Щел-

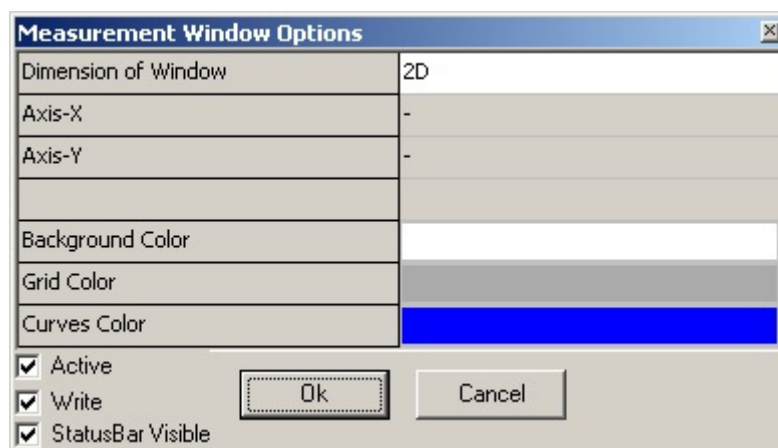


Рис. 17: Диалоговое окно создания нового рабочего окна.

чок курсора мыши на строке приводит к появлению контекстного меню (см. рис. 18), при этом у пользователя имеется возможность выбрать в качестве используемой переменной для выбранного канала любую из внутренних, внешних переменных, время или данные подсоединенных устройств.

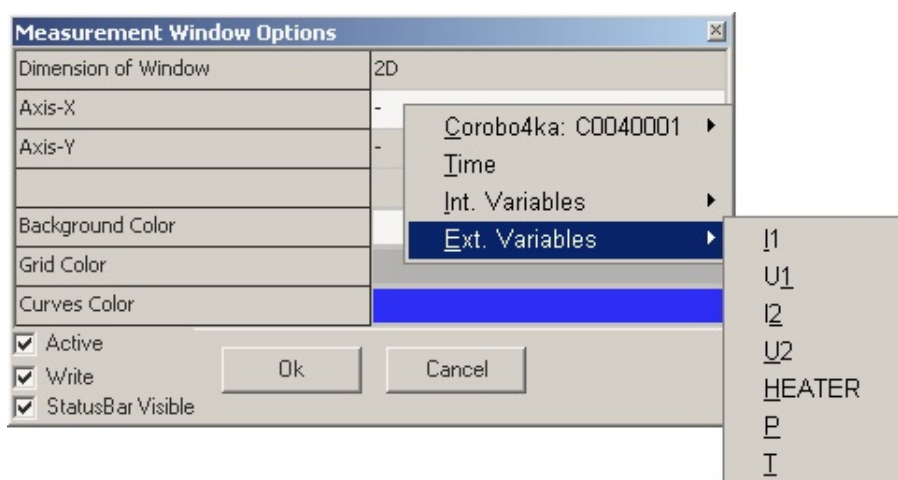



Рис. 18: Диалоговое окно создания нового рабочего окна с контекстным меню выбора переменной оси.

Поле **Background Color** отвечает за фоновый цвет создаваемого окна (по умолчанию белый). Щелчок мыши на окрашенном элементе приводит к появлению выпадающего списка, в котором можно выбрать наиболее привлекательный пользователю цвет. То же самое касается и полей **Grid Color** и **Curves Color**. Первое отвечает за цвет сетки, второе за цвет кривых, которыми будут отображаться данные. После можно будет выбрать индивидуальный цвет для каждой кривой.

В правой нижней части диалогового окна расположены 3 CheckBoxes. Галочка **Active** показывает активно ли окно и готово ли оно для приема данных. При отключении этой опции данные окном не принимаются. Галочка **Write** показывает будет ли производиться запись в это окно при выборе пункта меню **Tools>Recording**

или нажатии кнопки . Галочка **StatusBas Visible** указывает появится ли строка состояния в нижней части созданного окна. Нажатие кнопки **Ok** приводит к созданию рабочего окна, показанного на рис. 19. Для смещения сетки по горизонтали и вертикали используются клавиши клавиатуры со стрелками. Также можно зажать на колесико мышки и, передвигая ее, смещать координатную сетку. Изменять масштаб координатной сетки по горизонтали или вертикали можно нажимая клавиши клавиатуры со стрелками при нажатой клавише **Shift**.

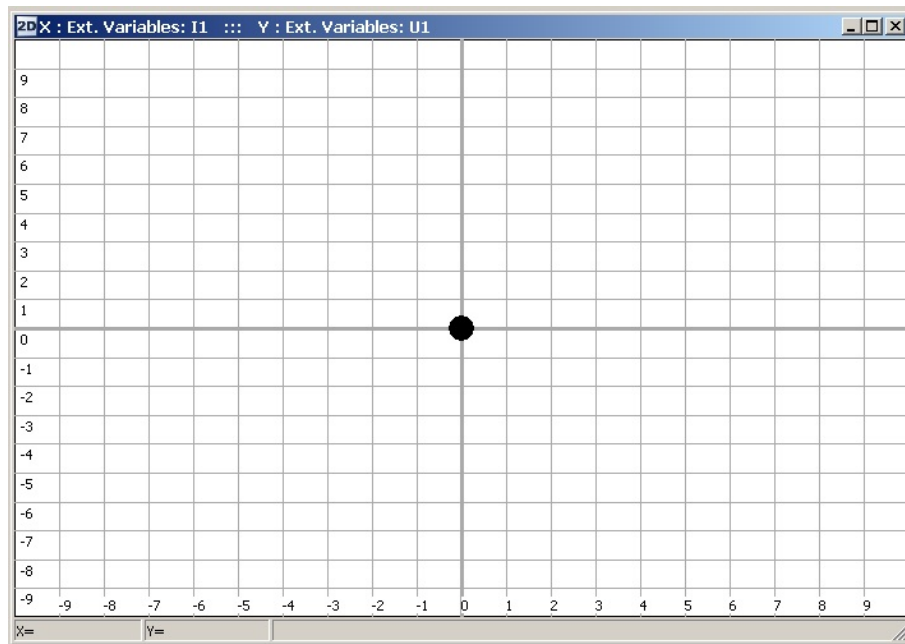


Рис. 19: Вид рабочего двумерного окна.

В нижней части окна находится строка состояний, в которой отображается текущее значение получаемых данных. Данные обновляются в момент, когда на каждую из осей координат приходит не менее одного значения. После этого счетчик значений для каждой оси сбрасывается. В центре экрана находится маркер. Маркер показывает текущее значение каждой из координат. Нажатие правой клавиши мыши на поле рабочего окна приводит к появлению контекстного меню показанному на рис. 20.

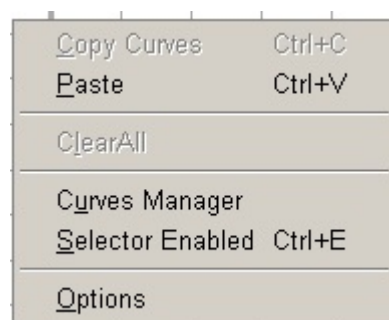


Рис. 20: Контекстное меню рабочего окна.

Пункт меню **Copy Curves** копирует данные в буфер обмена данных в собственном формате программы. Данный пункт меню предназначен для переноса данных из одного рабочего окна в другое.

Пункт меню **Paste** вставляет данные из буфера обмена данных, при этом данные могут быть как в собственном формате программы, так и в текстовом формате (при этом данные должны располагаться в 2 столбцах: первый - ось X, второй - ось Y (в случае трехмерного окна, данные должны находиться в 3 столбцах), разделенные либо пробелом, либо знаком табуляции, в конце каждой строке должен стоять символ переноса строки).

Пункт меню **Clear All** удаляет все существующие кривые.

Пункт меню **CurvesManager** открывает окно диспетчера кривых, показанное на рис. 21. В списке слева перечислены все существующие кривые. Символы левее названия кривой означают следующее: первая пиктограмма - цвет кривой (по щелчку курсором мыши на этой пиктограмме появляется контекстное меню, в котором можно выбрать цвет кривой), вторая пиктограмма - символ, обозначающий видимость точек кривой (если кружок желтый, то точки видимы, черный кружок означает невидимость точек кривой), щелчок курсором мыши на этой пиктограмме меняет состояние видимости кривой на противоположное, последняя пиктограмма - символ, показывающий соединены ли точки линией друг с другом (символ с маленьким кружочком означает, что линии не соединяются друг с другом), щелчок курсором мыши на этой пиктограмме меняет состояние соединенности точек кривой на противоположное. Правее названия кривых указаны радиус точек (R) и ширина линии, соединяющих точки (W). Справа по центру расположены 3 строки ввода, позволяющие изменить название кривой, радиус точек, и толщину линии.

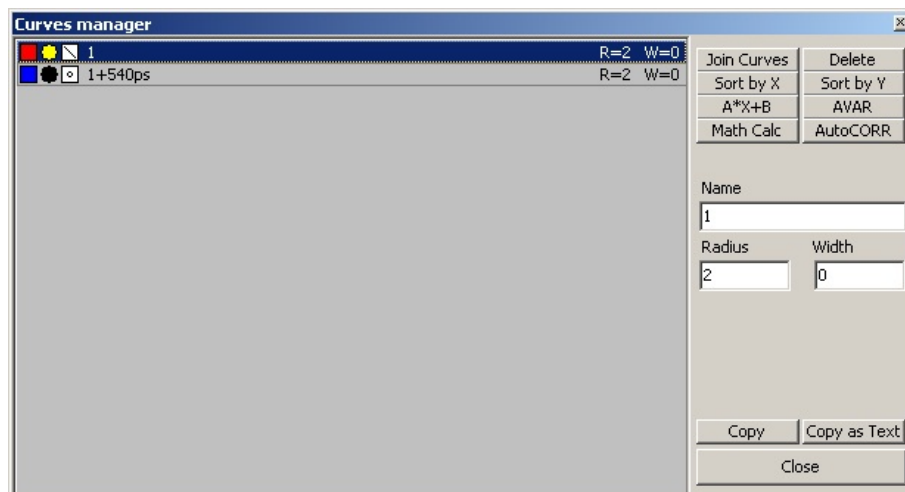


Рис. 21: Окно менеджера кривых двумерного окна.

В верхней правой части окна находятся кнопки позволяющие производить простые действия с кривыми. Нажатие кнопки **Join Curves** приводит к объединению выделенных кривых в одну, при этом все свойства (цвет, видимость, и т.д.) остаются такими же как и у первой из выделенных в списке кривых.

Нажатие кнопки **Delete** или клавиши **Del** на клавиатуре приводит к удалению выделенных кривых.

Нажатие кнопки **Sort by X** приводит к сортировке кривой по возрастанию координаты X.

Нажатие кнопки **Sort by Y** приводит к сортировке кривой по возрастанию координаты Y.

Кнопка **A*X+B** предназначена для проведения линейных трансформаций кривых. При этом открывается диалоговое окно, показанное на рис. 22. Введя коэффициенты в строки ввода и нажав на кнопку **Do It**, можно провести линейную трансформацию выделенных кривых.

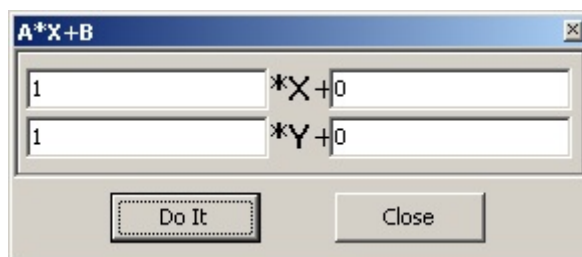


Рис. 22: Диалоговое окно, появляющееся при нажатии на кнопку **A*X+B**.

Следующие функции достаточно специфичны и вряд ли будут иметь широкое практическое применение. Кнопка **AVAR** строит вариацию Алана от выделенных кривых. **Math Calc** производит вычисление среднего значения и среднеквадратичного отклонения ординат выделенных кривых. **AutoCORR** строит автокорреляционную функцию.

В нижней части окна расположены кнопки **Copy** и **Copy as Text**. Первая предназначена для копирования данных выделенных кривых в буфер обмена данных в собственном формате. Вторая кнопка предназначена для копирования данных в буфер обмена данных в текстовом формате с возможностью последующей вставки в текстовый редактор. Между этими двумя процессами существует большая разница во времени выполнения команд в пользу копирования в собственном формате.

Окно диспетчера кривых для трехмерных графиков несколько отличается. Оно представлено на рисунке 23. Данные в трехмерных окнах могут отображаться двумя способами - градиентная раскраска (при этом цвет точек плавно меняется от черного до цвета кривой, обозначенного самой левой пиктограммой левее названия кривой, черный цвет соответствует минимальному значению, цвет кривой - максимальному), либо раскраска в цветах радуги (при этом фиолетовому цвету соответствует минимальное значение, красному - максимальное). Выбор способа представления данных производится по щелчку на самой правой пиктограмме левее имени кривой. Третья слева пиктограмма отвечает за отображение точек, если она принимает вид квадрата, то точки отображаются прямоугольниками, со сторонами RX и RY соответственно. В случае, когда эта пиктограмма принимает вид кружка, точки отображаются в виде кругов с радиусом R. В строках ввода в правой части окна можно изменять параметры кривой.

Назначение кнопок в правой части окна такое же как и для диспетчера кривых двумерного рабочего окна. Появившаяся кнопка **Gauss** позволяет делать гауссово размытие кривой. При нажатии этой кнопки возникает диалоговое окно (рис. 24).

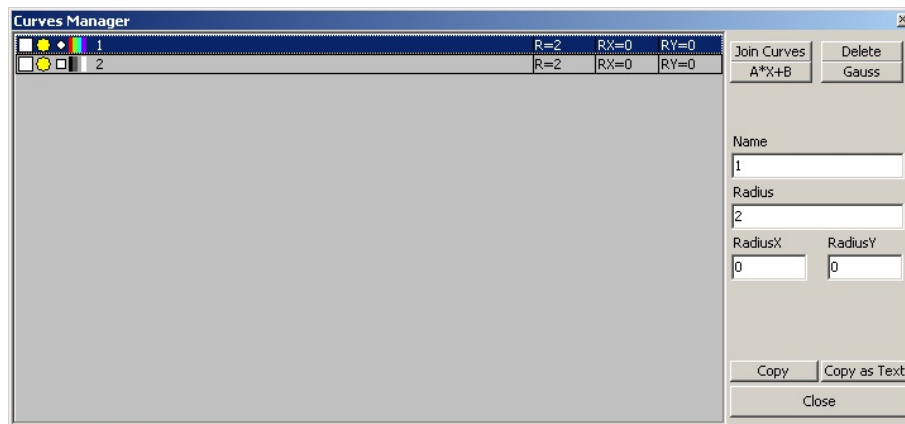
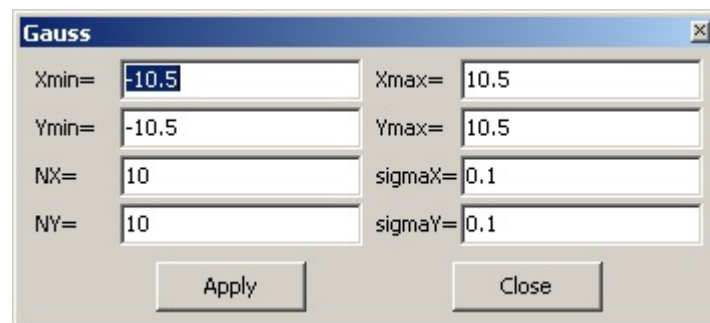



Рис. 23: Окно менеджера кривых трехмерного окна.

Рис. 24: Диалоговое окно, появляющееся при нажатии на кнопку **Gauss**.

Границы в которых происходит построение размытого изображения определяется полями Xmin, Xmax, Ymin, Ymax, при этом при размытии учитываются и точки кривой вне выбранного прямоугольника. NX и NY определяют количество точек по каждой из сторон прямоугольника, в котором происходит построение гауссова размытия. При этом общее число точек, в которых будет произведено вычисление равняется $NX \cdot NY$. sigmaX и sigmaY - параметры размытия по каждой из осей. Чем больше этот параметр, тем более размытым получится результат. Стоит выбирать эти значения порядка среднего расстояния между точками кривой. По нажатию кнопки **Apply** происходит вычисление матрицы, соответствующей размытому изображению. Результат представляется в виде кривой. Использование этой функции удобно, когда имеется ряд дискретных точек, а необходимо получить непрерывное изображение.

Вернемся к контекстному меню рабочего окна. При нажатии на пункт **Selector Enabled** контекстное меню принимает вид, показанный на рис. 25.

При этом появляется возможность выбора точек прямоугольником при нажатой левой клавиши мыши. Далее можно удалить точки внутри прямоугольника (Пункт меню **Delete**) или вне прямоугольника (пункт меню **Delete Outside**).

Пункт меню **Options** соответствует выбору пункта меню **Tools>Options** или нажатию на кнопку , с последующим выбором активного окна. При этом внешний вид диалогового окна **Options** становится таким как показано на рисунке 26.

Блок **Axes** позволяет выбрать источник данных для соответствующей оси ра-

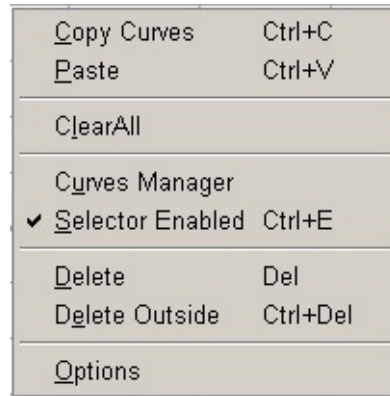


Рис. 25: Контекстное меню рабочего окна.

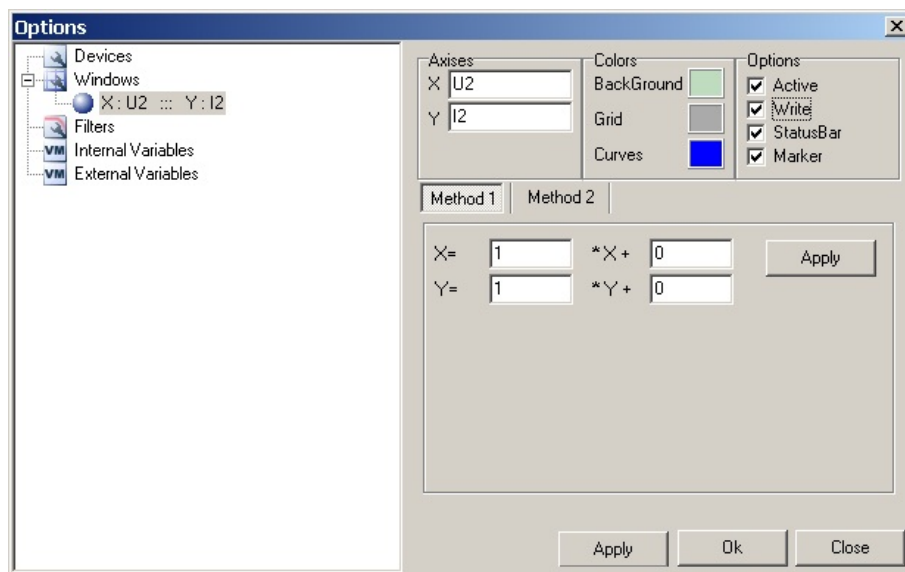



Рис. 26: Окно редактирования рабочего окна.

бочего окна. Щелчок курсором мыши на строках ввода в этом блоке приводит к появлению контекстного меню, аналогичного тому, которое появляется в диалоговом окне создания нового окна.

Блок **Colors** позволяет выбрать цветовую гамму для рабочего окна. Щелчок курсором мыши на цветовой панели вблизи надписей **BackGround**, **Grid** и **Curves** приводит к появлению контекстного меню с возможностью выбора цвета. Выбранный цвет будет установлен как основной для фона рабочего окна, координатной сетки и цвета создаваемых кривых соответственно.

Блок **Options** позволяет изменять следующие свойства рабочего окна. Галочка **Active** показывает активно ли окно и готово ли оно для приема данных. При отключении этой опции данные окном не принимаются. Галочка **Write** показывает будет ли производиться запись в это окно при выборе пункта меню **Tools>Recording** или нажатии кнопки . Галочка **StatusBas** указывает появится ли строка состояния в нижней части созданного окна. Галочка **Marker** указывает на видимость маркера в

рабочем окне программы.

Все свойства рабочего окна меняются только после нажатия на кнопку **Apply** или **Ok** в нижней части диалогового окна.

В центральной части окна находится блок линейной трансформации входных данных. В программе осуществлены 2 способа линейной трансформации входных данных. Первый способ, реализованный на вкладке **Method1**, подразумевает использование формулы $X' = A \times X + B$ для каждой из осей, где коэффициенты A и B задаются пользователем. Второй способ, реализованный на вкладке **Method2**, связывает пару точек нетрансформированных данных с парой точек выходных данных. Изменения вступают в силу только после нажатия на кнопку **Apply** в этом блоке. Для трехмерных окон в этом блоке также указывается минимальное и максимальное значения оси Z.

ПРИМЕР:

Линейная трансформация вида $1000 \cdot X + 0$ вторым методом будет записана как

Xold1 = 0

Xnew1=0

Xold2 = 1

Xnew2=1000

Настройка источника смещения при помощи программы Prefences.exe.

Для полной настройки устройства используется программа **Prefences.exe**. Для использования всех функций этой программы вам понадобятся компьютер, подключенный к сети Internet (обязательно наличие одного порта USB для подключения устройства и одного COM-порта для подключения мультиметра, COM-порт может быть заменен переходником подключенным к USB-порту) с установленными на нем драйверами для микросхемы FT232R. Также для полной калибровки устройства вам понадобится мультиметр Agilent34401A, подсоединенный к COM-порту компьютера. Наличие подключения к сети Internet необходимо только для обновления программного обеспечения устройства и отправки калибровочных данных производителю. COM-порт и мультиметр Agilent34401A необходимы лишь в случае калибровки каналов и печки. Далее будет произведено пошаговое описание работы программы. При работе вы всегда можете вернуться на любой из предыдущих шагов нажав на соответствующую вкладку.

Шаг 1 - Инициализация.

При запуске программы появляется рабочее окно показанное на рис [27](#).

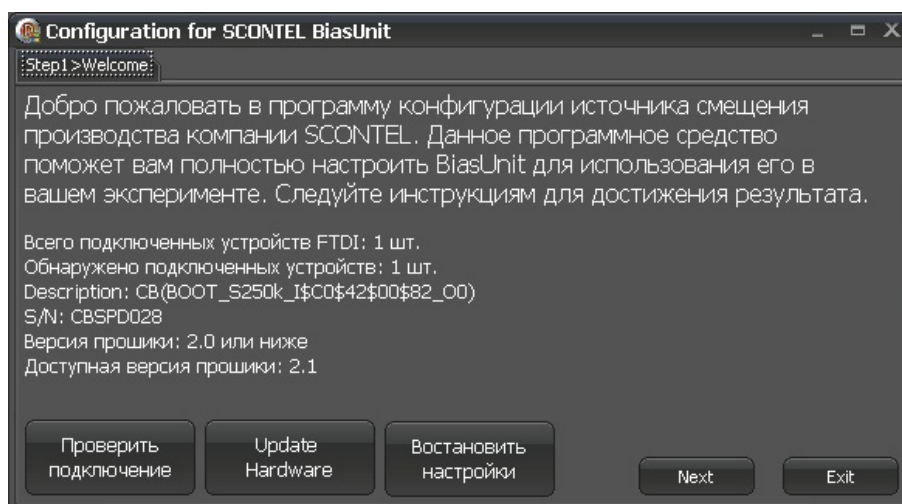


Рис. 27: Вид окна программы на шаге 1.

В окне выводится краткая информация о подключенных к компьютеру устрой-

ствах, их количестве, серийном номере и версии прошивки. При запуске программа получает информацию с сайта производителя о последней версии прошивки устройства и отображает эту информацию на экране. Правильная работа программы возможна в случае подключения только одного устройства BiasUnit к компьютеру. В противном случае пользователю будет доступна лишь кнопки **Проверить подключение** и **Exit**. При нажатии на кнопку **Проверить подключение** происходит проверка подключенных устройств к компьютеру аналогично тому, как это делается в случае начала работы программы. Кнопка **Exit** осуществляет выход из программы.

Обнаружение одного устройства FTDI и при условии не обнаружение устройств BiasUnit соответствует случаю первоначального включения устройства. В этом случае рядом с информацией о подключенных устройствах появляется кнопка **Первоначальная конфигурация FTDI**. При нажатии на эту кнопку появляется окно первоначальной конфигурации устройства (см. рис. 28), в котором нужно лишь указать серийный номер устройства и нажать на кнопку **Config**. Следует крайне внимательно подойти к этой операции. Поскольку, в случае если имеется подключенное к компьютеру устройство другого производителя, то оно может быть неправильно инициализировано и в будущем не работать.

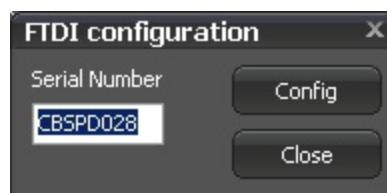


Рис. 28: Окно первоначальной конфигурации устройства.

После успешной конфигурации устройства должно произойти автоматическое переподключение устройства. В этом случае, вам возможно придется заново установить драйвера на устройство. При этом программа не будет правильно работать, пока процесс установки драйверов не завершится.

В случае успешного обнаружения BiasUnit откроется возможность использования программы. При нажатии на кнопку **Update Hardware** происходит подключение к сети Internet. Программа скачивает последнюю версию прошивки для устройства и программирует BiasUnit. Процесс полностью автоматизирован и не требует никаких действий со стороны пользователя. Все этапы процесса протоколируются на форме окна. В случае сбоя обновления программы необходимо связаться с производителем и сообщить на каком этапе происходит сбой обеспечения. В случае успеха форма загрузки примет вид, показанный на рис. 29:

Нажатие на кнопку **Восстановить настройки** открывает диалоговое окно, в котором пользователю предлагается выбрать ранее сохраненный файл настройки в формате *.bbc (старый формат файла настройки, включающий только калибровочную информацию устройства) или *.pref (файл настройки и калибровки). После выбора файла настройки автоматически загрузятся в BiasUnit. Процесс загрузки занимает некоторое время. Для того, чтобы настройки были сохранены в постоянную память устройства необходимо нажать на кнопку **SaveEEPROM**, доступную на шагах 2 и 3.

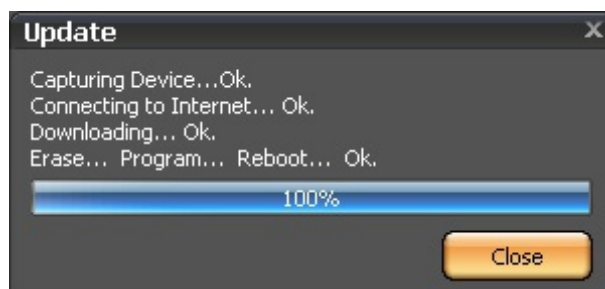


Рис. 29: Окно обновления программы **Prefences.exe**.

Нажатие на кнопку **Next** позволяет перейти к следующему шагу настройки устройства.

Шаг 2 - Конфигурирование устройства.

При переходе к шагу 2, форма приложения изменится на показанную на рис. 30.

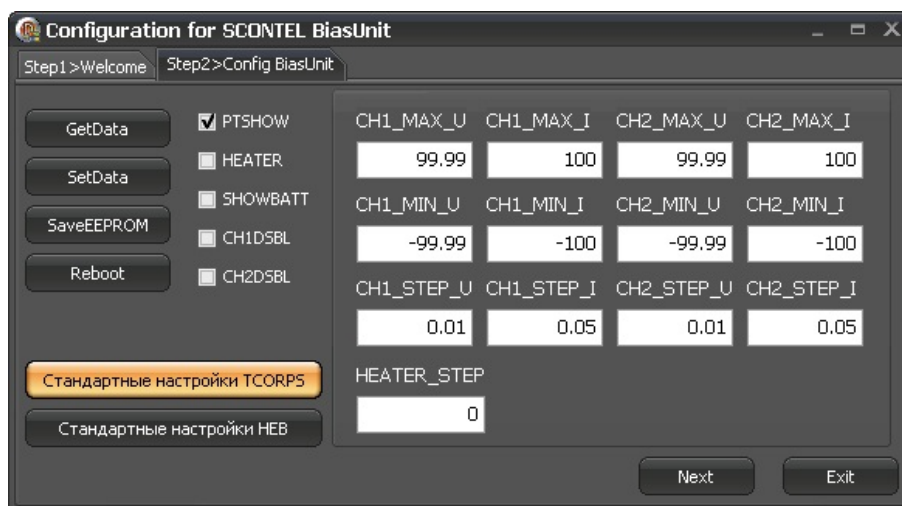


Рис. 30: Вид окна программы на шаге 2.

В левой части рабочего окна находятся поля ввода, позволяющие редактировать минимальные (CHx_MIN_U/I) и максимальные (CHx_MAX_U/I) пределы перестройки тока/напряжения для обоих каналов, шаг перестройки (CHx_STEP_U/I) тока/напряжения обоих каналов и шаг перестройки напряжения печки ($HEATER_STEP$).

Левее полей ввода находятся 5 флажков, определяющих работу устройства. Флажок **PT_SHOW** определяет необходимость измерения давления и температуры и вывода их значений на экран. При отключенном флажке АЦП1 не инициализируется. Флажок **HEATER** определяет наличие в устройстве источника смещения печки. При включении флажка **SHOWBATT** на экране отображается величина заряда аккумуляторов (используется только в источниках смещения с аккумуляторным питанием). Флажок **CH1DSBL** отключает первый канал источника смещения (при этом

АЦП2 не инициализируется). Флажок CH2DSBL отключает второй канал источника смещения (при этом АЦП3 не инициализируется).

Для получения данных о пределах, шагах перестройки включенных опциях необходимо нажать на кнопку **GetData**. При этом программа делает запрос к устройству и выводит в соответствующие поля информацию, полученную от устройства. Аналогичная операция происходит при переходе с **Шага 1** по нажатию на кнопке **Next**. Кнопка **SetData** необходима для передачи и записи данных в оперативную память устройства. Запись в постоянную память устройства всех данных производится при нажатии на кнопку **SaveEEPROM**. Изменения внешнего вида экрана источника смещения будут произведены после перезагрузки устройства. Для перезагрузки вы можете нажать на кнопку **Reboot**.

В нижнем левом углу есть кнопки для инициализации двух наиболее часто встречаемых конфигураций: двухканальный источник смещения для однофотонной системы с измерением давления и температуры и одноканальный источник смещения для болометрической системы.

Нажатие на кнопку **Next** позволяет перейти к следующему шагу настройки устройства.

Шаг 3 - Калибровка устройства.

При переходе к шагу 3, форма приложения изменится на показанную на рис. 31.

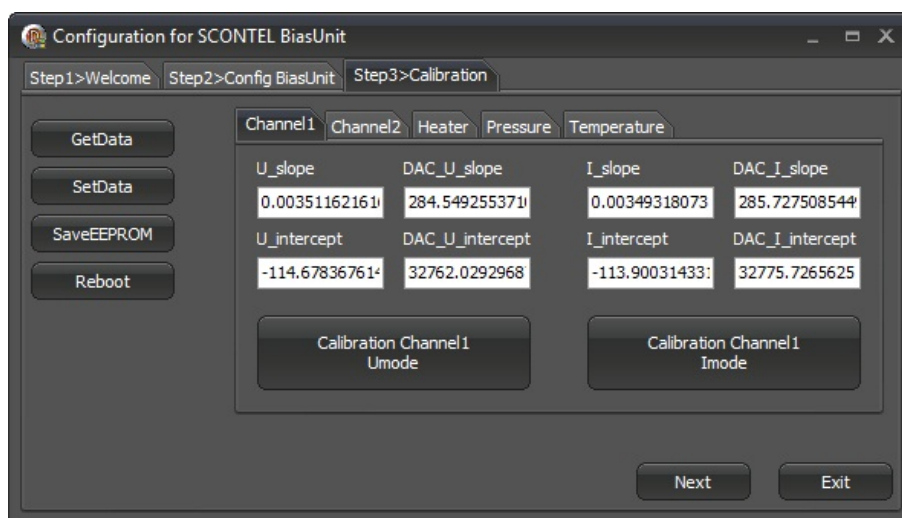


Рис. 31: Вид окна программы на шаге 3.

В правой части формы экрана находится поле с вкладками, соответствующими каждому из включенных каналов управления и индикации. В полях ввода находятся значения соответствующих калибровочных коэффициентов. Для получения калибровочных данных необходимо нажать на кнопку **GetData**. При этом программа делает запрос к устройству и выводит в соответствующие поля информацию, полученную от источника смещения. Аналогичная операция происходит при переходе с **Шага 2** по нажатию на кнопке **Next**. Кнопка **SetData** необходима для передачи и записи

калибровочных данных в оперативную память устройства. Запись в постоянную память устройства всех данных производится при нажатии на кнопку **SaveEEPROM**. Для перезагрузки устройства вы можете нажать на кнопку **Reboot**.

Калибровка источника смещения.

Для понимания смысла калибровочных коэффициентов для источника смещения следует обратиться к блок-схеме представленной на рис. 32.

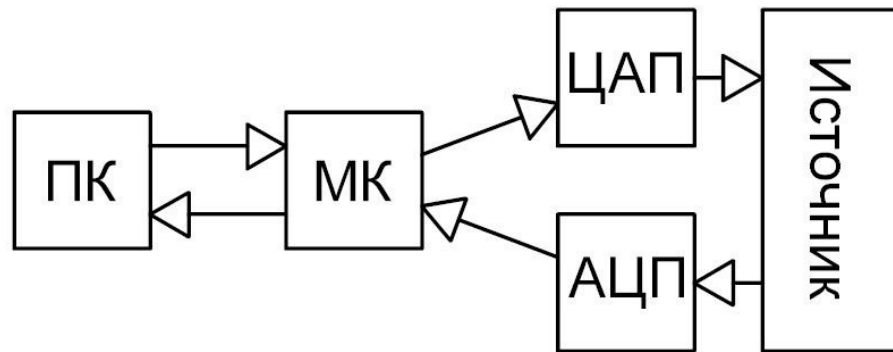


Рис. 32: Блок-схема, поясняющая смысл калибровочных коэффициентов.

В приведенной схеме сигнал о необходимом для установки значении тока-напряжения идет от компьютера (ПК или РС) в микроконтроллер (МК или МС), расположенном на основной плате источника смещения. В случае автономной работы это значение формируется в самом МК. Причем эти значения являются аналоговыми значениями, выраженными в единицах тока/напряжения. Далее МК использует линейную зависимость и высчитывает цифровой код для установки выходного тока-напряжения при помощи цифро-аналогового преобразователя (ЦАП или DAC). Формула применительно к току первого канала выглядит следующим образом:

$$I_{DAC} = DAC_I_slope_1 \times I_{MC} + DAC_I_intercept_1.$$

Значения тока/напряжения измеряются при помощи аналого-цифрового преобразователя (АЦП или ADC), на выходе которого получается цифровой код. Этот цифровой код переводится в значение тока напряжения по формуле похожей на предыдущую формулу.

$$I_{MC} = I_slope_1 \times I_{ADC} + I_intercept_1.$$

Калибровка источников смещения заключается в нахождении неизвестных коэффициентов: $DAC_X_slope_N$, $DAC_X_intercept_N$, X_slope_N , $X_intercept_N$, где $X = I, U$, а $N = 1, 2$. Для получения этих коэффициентов необходимо обязательное наличие мультиметра Agilent34401A. Начало калибровки происходит после нажатия на одну из 4-х кнопок **Calibration ChannelN Xmode**. После нажатия на эту кнопку возникает окно калибровочной утилиты (см. рис. 33). Также на весь рабочий стол раскрывается окно, в котором отображается информация о ходе калибровки в виде графика. График в процессе калибровки должен быть линейным, в противном случае следует говорить о неисправности источника смещения.

Перед начало калибровки убедитесь, что выбранный для калибровки канал источника смещения подключен к измерительному входу мультиметра Agilent34401A,

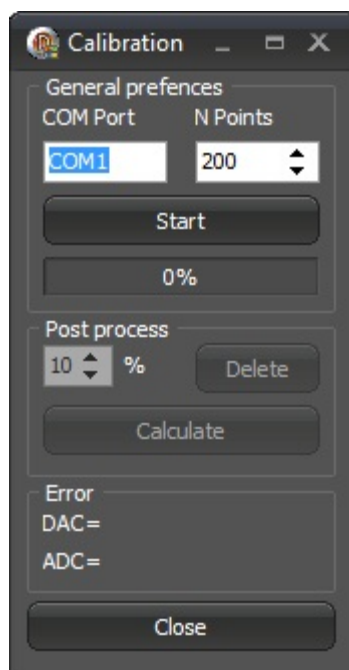


Рис. 33: Внешний вид калибровочной утилиты.

а сам мультиметр подключен к компьютеру через RS232 интерфейс. В поле редактирования **COM Port** необходимо указать название порта, к которому подключен ваш мультиметр, в поле **N Points** - количество точек, для калибровки. После выполнения подготовительных работ вы можете начать калибровку нажатием на кнопку **Start**. Прогресс калибровки отображается ниже кнопки. Сам процесс калибровки занимает длительное время, в течении которого не рекомендуется прикасаться к используемым приборам. Процесс калибровки всегда можно прервать нажав на кнопку **Stop**, появляющуюся за место кнопки **Start**.

В ходе процесса калибровки на графике будут строиться измеряемые величины. Для точек строящихся синим цветом ордината соответствует измеренному мультиметром значению, абсцисса точки соответствует цифровому коду, посылаемому на ЦАП. Для точек строящихся зеленым цветом ордината также соответствует измеряемому мультиметром значению, а абсцисса соответствует цифровому коду, измеряемому при помощи АЦП. Для смещения графика по горизонтали или вертикали удерживайте клавиши клавиатуры со стрелками. Также можно перемещать график при нажатой центральной клавиши мыши (при нажатии на колесико мыши). Для изменения масштаба по горизонтали или вертикали удерживая клавишу **Shift** нажмите соответствующую клавишу со стрелкой. Во время процесса калибровки может произойти сбой, связанный с ошибкой приема-передачи команды. В этот момент может произойти выброс, который повлияет на конечную калибровку устройства. Вероятность этого достаточно мала, но контролировать процесс все же необходимо.

После окончания процесса калибровки откроется доступ к математическим функциям обработки данных. Нажатие на кнопку **Delete** приводит к удалению наиболее удаленных от линейного закона точек. При этом удаляются точки в количестве указанном левее этой точки.

Также открыта возможность самостоятельного удаления пользователем лишних точек. Для этого необходимо в контекстном меню графического окна выбрать пункт **Selector Enabled**, далее при помощи мышки выделить точки подлежащие удалению и нажать клавишу клавиатуры **Del** или выбрать в контекстном меню пункт **Delete**. После такого удаления следует пересчитать калибровочные коэффициенты нажав на кнопку **Calculate**.

Метками **Error_DAC** и **Error_ADC** отображается среднеквадратичное отклонение от линейного закона измеренных точек. Естественно, что меньшее значение этих коэффициентов соответствует лучшей калибровке. После окончания калибровки и нажатии на кнопку **Close**, данные сохраняются в оперативной памяти устройства и занимают соответствующие поля в форме программы.

Калибровка печки.

В случае, когда в устройстве используется печка (примером работы устройства в таком режиме может быть использование его для работы с прямым детекторным приемником, в котором необходим дополнительный подогрев детектора), необходимо провести калибровку напряжения на печке. Калибровка производится аналогично описанному в разделе **Калибровка источника смещения**. В ходе калибровки часто может наблюдаться нелинейность синей кривой. После окончания калибровки обе кривые автоматически аппроксимируются прямой линией. Нелинейность синей кривой приведет к тому что шаг перестройки печки будет нелинеен. При большем значении напряжения шаг перестройки меньше. Такого рода нелинейность в эксперименте бывает более удобна, чем линейная зависимость шага перестройки. Стоит отметить, что зеленая кривая (кривая показаний АЦП) остается линейной, т.е. отображаемые на дисплее значения напряжения печки являются истинными.

Калибровка датчика давления.

Зависимость выходного напряжения датчика давления от давления линейна. Этот факт используется для калибровки датчика всего по 2 точкам. При выборе вкладки **Pressure** окно программы будет выглядеть как показано на рис. 34.

В качестве одного опорного давления удобно выбрать атмосферное давление, что соответствует на приведенной картинке точке №1. Для получения значения АЦП, измеряющего давление, в этой точке необходимо нажать на соответствующую кнопку **ReadADC**. При этом прочитанное значение АЦП должно быть равно 0 либо близко к 0. Далее датчик давления откачивается. Давление необходимо оценивать по эталонному манометру. В поле ввода соответствующему второй точке необходимо указать измеренное давление и нажать на соответствующую кнопку **ReadADC**. Полная откачка соответствует значению АЦП $\sim 50000 - 60000$. После нажатия на кнопку **Calculate** соответствующие калибровочные коэффициенты автоматически высчитываются и загружаются в устройство, а также отображаются в полях **P_slope** и **P_intercept**.

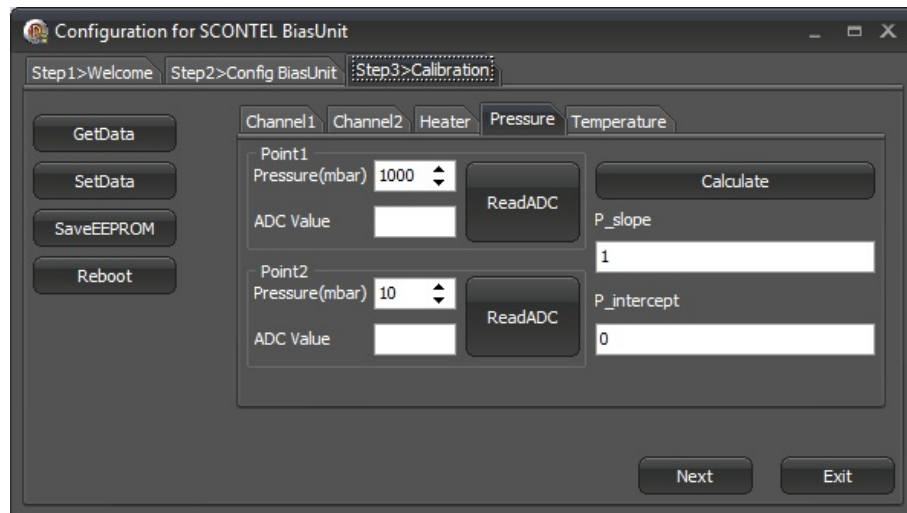


Рис. 34: Калибровка давления.

Калибровка термометра.

Калибровка термометра в памяти источника смещения представлена в виде массива из 100 значений напряжения термометра и соответствующих им значений температуры. Используемые термометры имеют одинаковые температурные зависимости. Схема источника питания и измерения напряжения термометра подразумевает наличие погрешностей, что может привести к появлению ошибки, которая может составлять несколько десятых градуса кельвина. Для учета погрешности необходимо получить поправку к величине зондирующего тока. При выборе вкладки **Temperature** окно программы будет выглядеть как показано на рис. 35.

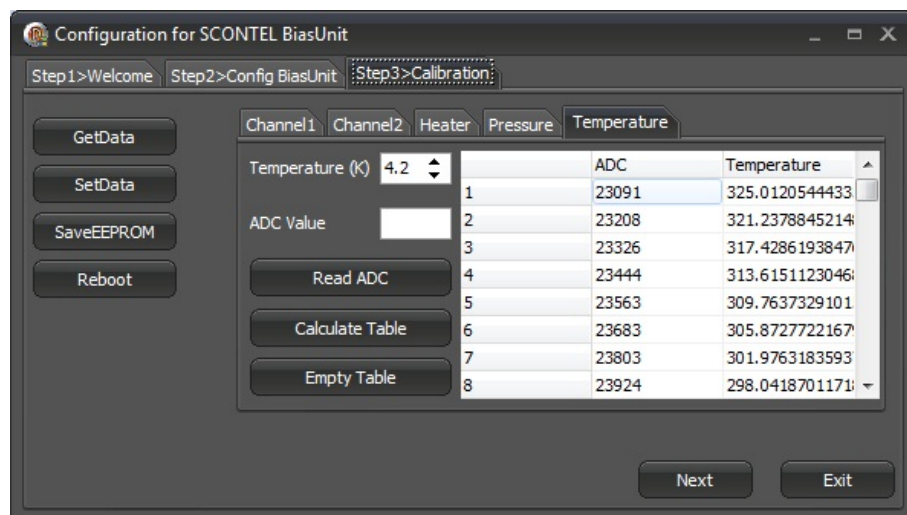


Рис. 35: Калибровка температуры.

Для калибровки мы используем среду известной температуры. В качестве такой температуры можно выбрать температуру кипения жидкого гелия (~ 4.2). По-

скольку температура кипения жидкого гелия является зависящей от атмосферного давления величиной и при изменении давления в пределах нормы может меняться более чем 0.1K, удобно выбрать в качестве калибровочной температуры λ -точку He (2.17K).

Для начала калибровки необходимо в поле **Temperature (K)** указать температуру термометра, после чего нажать на кнопку **ReadADC**. В поле **ADC Value** появится значение сопротивления термометра.

Для удобства хранения и вычисления температуры по показаниям АЦП в памяти МК строится таблица значений показаний АЦП и соответствующим им температурах. Таблица ограничена в размерах. Используется только 100 точек, при этом шаг по температуре выбран логарифмическим и покрывающим весь интересующий диапазон. Для формирования этой таблицы после введения поправки тока необходимо нажать на кнопку **Calculate Table**. Вы также можете использовать линейный закон температуры. Для формирования такой таблицы необходимо нажать на кнопку **Empty Table**. Загрузка таблицы в оперативную память устройства осуществляется по нажатию кнопки **SetData**.

Шаг 4 - Сохранение данных конфигурации устройства.

При переходе к шагу 4, форма приложения изменится на показанную на рис. 36.

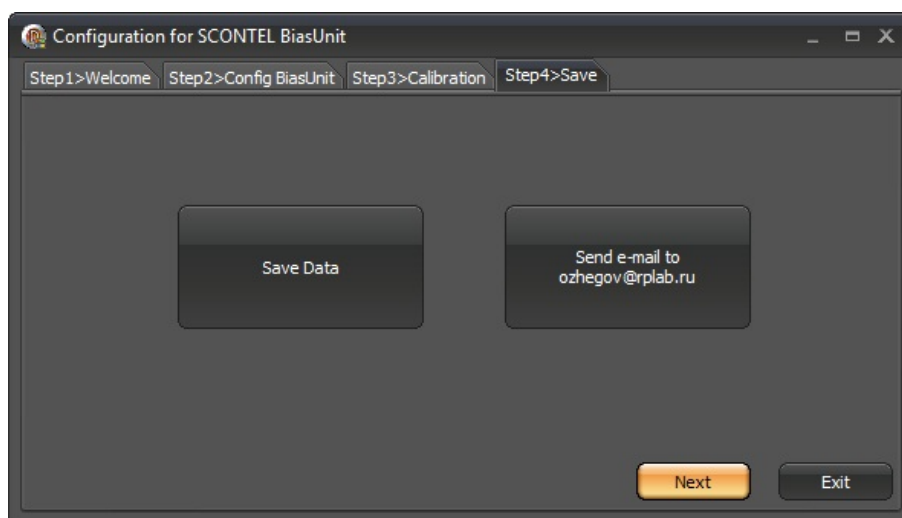


Рис. 36: Вид окна программы на шаге 4.

На этом шаге, являющимся последним, активно только 2 кнопки. При нажатии на кнопку **SaveData** пользователю предлагается выбрать место для записи файла конфигурации в удобное для него место. По нажатию на кнопку **Send e-mail** файл конфигурации будет отправлен разработчику, что позволит ему иметь полную базу данных о калибровочных данных на все произведенные устройства.

Пакет компонентов для работы в среде LabView.

Для работы в среде LabView разработан пакет компонентов виртуальных инструментов обеспечивающих управление и сбор данных источником смещения. В составе комплекта программного обеспечения subVI, являющихся оболочками к функциям динамической библиотеки biasbox.dll.


Для организации работы по принципу "потока данных", все subVI имеют входной и выходной терминал ошибки, за исключением bb_Open (имеет только выходной терминал) и bb_Close (имеет только входной терминал). Терминал ошибки является кластером со следующими полями:

ERROR - принимает значение TRUE в случае возникновения ошибки обмена.

CODE - возвращает код ошибки (см. описание biasbox.dll).

SOURCE - Возвращает текстовое название subVI вызвавшего ошибку.

Виртуальный инструмент bb_Open.vi  предназначен для инициализации интерфейса обмена данными. Имеет один выходной терминал ошибки.

Виртуальный инструмент bb_GetData.vi  предназначен для получения данных от источника смещения. Имеет входной и выходной терминал ошибки.

DATA - Массив кластеров с данными от прибора. Кластеры имеют вид:

I_x - значение тока на канале x в А.

U_x - значение напряжения на канале x в V.

P - значение давления mBar.

T - значение температуры в K.

R - сопротивление терморезистора в Ohm.


HEATER - напряжение на печке в V.

BAT_x - напряжения в mV батарей положительного (P) и отрицательного питания (N).

STATUS - Режимы стабилизации источников и нуля на выходе (см. описание на biasbox.dll).

ERROR - Ошибки в работе прибора (см. описание на biasbox.dll).

NUMBER - количество кластеров с данными во входном буфере.

Виртуальный инструмент bb_Value.vi  устанавливает значение тока или напряжения в зависимости от выбранного режима работы. Имеет входной и выходной терминал ошибки.

CHANNEL - номер канала (0 или 1).

VALUE - Значение тока или напряжения в зависимости от текущего режима стаби-

лизации.



Виртуальный инструмент `bb_SetMode.vi` устанавливает режим стабилизации тока-напряжения. Имеет входной и выходной терминал ошибки.

SET - режим стабилизации: FALSE - по напряжению; TRUE - по току.

CHANNEL - номер канала (0 или 1).



Виртуальный инструмент `bb_SetZero.vi` устанавливает нулевое значение на выбранном канале. Имеет входной и выходной терминал ошибки.

SET - установка нуля (короткое замыкание выхода источника): FALSE - выход разомкнут; TRUE - выход замкнут.

CHANNEL - номер канала (0 или 1).



Виртуальный инструмент `bb_Close.vi` предназначена для закрытия интерфейса и завершения процесса обмена данными. Имеет выходной терминал ошибки.

В комплекте поставки приведен простой пример того, как организовать работу источника смещения (файл `simple_demo.vi`)

Использование динамической библиотеки данных biasbox.dll.

Динамическая библиотека данных biasbox.dll предназначена для работы с источником смещения. При этом имеется возможность создания своих приложений с подключением функций библиотеки, предназначенных для управления и сбора данных источника смещения. Работу с устройством необходимо начать с открытия его, используя функцию `bb_Open`, завершать работу нужно закрытием устройства, используя команду `bb_Close`.

Описание функций.

Инициализации интерфейса обмена данными.

`char bb_Open(void);`

Возвращаемый параметр равен 1 в случае успешного открытия интерфейса и 0 в случае ошибки.

Закрытие интерфейса и завершение обмена данными.

`char bb_Close(void);`

Возвращаемый параметр равен 1 в случае успешного закрытия интерфейса и 0 в случае ошибки.

Получение данных.

`char bb_GetData(PMEASDATA buf);`

Входным параметром процедуры является указатель на структуру типа `MEASDATA`, описанную ниже.

Возвращаемый параметр равен 1 в случае успешного выполнения функции и 0 в случае ошибки.

Конфигурация источника смещения.

`char bb_PutData(PCONFIGDATA buf);`

Входным параметром процедуры является указатель на структуру типа `CONFIGDATA`, описанную ниже.

Возвращаемый параметр равен 1 в случае успешного выполнения функции и 0 в случае ошибки.

Получение калибровочных констант.

char bb_GetEeConst(PEECONST buf);

Входным параметром процедуры является указатель на структуру типа EEECONST, описанную ниже.

Возвращаемый параметр равен 1 в случае успешного выполнения функции и 0 в случае ошибки.

Пересылка калибровочных констант в источник смещения.

char bb_PutEeConst(PEECONST buf);

Входным параметром процедуры является указатель на структуру типа EEECONST, описанную ниже.

Возвращаемый параметр равен 1 в случае успешного выполнения функции и 0 в случае ошибки.

Получение таблицы зависимости истинной температуры от показаний датчика температуры.

char bb_GetEeTable(PUSHORT TempCode, PFLOAT TempData);

Процедура имеет два входных параметра:

PUSHORT TempCode - указатель на массив из 100 элементов с указанием кодов АЦП датчика температуры.

PFLOAT TempData - указатель на массив из 100 элементов с указанием истинных значений температуры, соответствующим кодам АЦП в массиве TempCode.

Возвращаемый параметр равен 1 в случае успешного выполнения функции и 0 в случае ошибки.

Передача таблицы зависимости истинной температуры от показаний датчика температуры.

char bb_PutEeTable(PUSHORT TempCode, PFLOAT TempData);

Процедура имеет два входных параметра:

PUSHORT TempCode - указатель на массив из 100 элементов с указанием кодов АЦП датчика температуры.

PFLOAT TempData - указатель на массив из 100 элементов с указанием истинных значений температуры, соответствующим кодам АЦП в массиве TempCode.

Возвращаемый параметр равен 1 в случае успешного выполнения функции и 0 в случае ошибки.

Сохранение текущих констант в постоянную память источника смещения.

char bb_SaveEeConst(void);

Процедура выполняется достаточно долго (порядка 10-20сек.)

Возвращаемый параметр равен 1 в случае успешного выполнения функции и 0 в случае ошибки.

Получение данных в формате LabView.

UINT bb_LVGetData(PMEASDATA buf);

Входным параметром процедуры является указатель на структуру типа MEASDATA, описанную ниже.

Возвращаемый параметр равен 1 в случае успешного выполнения функции и 0 в случае ошибки.

Конфигурация источника смещения для среды LabView.

UINT bb_LVPutData(PCONFIGDATA buf);

Входным параметром процедуры является указатель на структуру типа MEASDATA, описанную ниже.

Возвращаемый параметр равен 1 в случае успешного выполнения функции и 0 в случае ошибки.

Описание типов данных и констант.

Структуры данных, используемые в динамической библиотеке данных biasbox.dll

```
typedef struct _MEASDATA
```

```
{
float I1; //Ток первого канала в мкА
float U1; //Напряжение первого канала в мВ
float I2; //Ток второго канала в мкА
float U2; //Напряжение второго канала в мВ
float P; //Давление
float T; //Температура
float R; //Сопротивление термометра
unsigned short int BATP; //Заряд положительной батареи аккумулятора
(показываемое напряжение необходимо умножить на 10мВ)
unsigned short int BATN; //Заряд отрицательной батареи аккумулятора
(показываемое напряжение необходимо умножить на 10мВ)
float HEATER; //Напряжение печи
unsigned char STATUS; //Статусный байт (значения битов описаны ниже)
unsigned char BBEROR; //Байт ошибок (значение битов описаны ниже)
} MEASDATA, *PMEASDATA;
```

```
typedef struct _CONFIGDATA
```

```
{ float I1; //Ток первого канала в мкА
float U1; //Напряжение первого канала в мВ
float I2; //Ток второго канала в мкА
float U2; //Напряжение второго канала в мВ
float HEATER; //Напряжение печи
unsigned char STATUS; //Статусный байт (значения битов описаны ниже)
} CONFIGDATA, *PCONFIGDATA;
```

```
typedef struct _EECONST
```

```

{ float I_slope[2];
float I_int[2];
float U_slope[2];
float U_int[2]; //калибровочные значения для получения значения токов каналов в
мкА из цифровых значений показаний АЦП.
float DAC_I_slope[2];
float DAC_I_int[2];
float DAC_U_slope[2];
float DAC_U_int[2];
float R_slope;
float R_int;
float UH_slope;
float UH_int;
float DAC_UH_slope;
float DAC_UH_int;
float P_slope;
float P_int;
}EECONST, *PEECONST;

```

Биты статусного байта

```

Bit0 ENB_CH1 //КАНАЛ 1 ВКЛЮЧЕН
Bit1 ENB_CH2 //КАНАЛ 2 ВКЛЮЧЕН
Bit2 CUR_CH1 //КАНАЛ 1 В СТАБИЛИЗАЦИИ ПО ТОКУ
Bit3 CUR_CH2 //КАНАЛ 2 В СТАБИЛИЗАЦИИ ПО ТОКУ
Bit4 SHORT1 //ЗАКОРОТКА КАНАЛА 1 ВКЛЮЧЕНА
Bit5 SHORT2 //ЗАКОРОТКА КАНАЛА 2 ВКЛЮЧЕНА
Bit7 MAIN_MODE //РЕЖИМ РАБОТЫ: 1 - РАБОТА 0 - ЗАРЯД АККУМУЛЯ-
ТОРОВ

```

Биты байта ошибок

```

Bit0 ADC_1 //ADC1 не отвечает - превышено время ожидания
Bit1 ADC_2 //ADC2 не отвечает - превышено время ожидания
Bit2 ADC_3 //ADC3 не отвечает - превышено время ожидания
Bit3 BATP_ER //Ошибка зарядного устройства положительных батарей
Bit4 BATN_ER //Ошибка зарядного устройства отрицательных батарей
Bit5 BATP_CH //Положительные батарею заряжаются
Bit6 BATN_CH //Отрицательные батарею заряжаются

```

История версий.

Rev. 1.0. Первая версия технической инструкции.

Rev. 1.1. Из набора инструментов исключены программы Programmer.exe, calibr.exe, CalibrManager.exe. В этой и последующих версиях функции этих программ выполняет программа Prefences.exe.

Комплектация поставки.

Полный набор программного обеспечения для источника смещения находится в архиве **Toolset_Rev_1_1.zip**.

В каталоге **CalibrData** содержатся калибровки на устройства, созданные до выхода настоящего документа.

В каталоге **LabView-Vi** содержатся виртуальные компоненты для работы в среде LabView. Пример использования устройства рассмотрен в виртуальном инструменте **simple_demo.vi**.

Файл **biasbox.dll** - динамическая библиотека для управления устройством предназначена для разработки собственных приложений пользователем.

Файл **ControlCenter.exe** - программа предназначенная для управления и контроля показаний источника смещения.

Файл **Measure.exe** - программа предназначенная для сбора и отображения данных.

Файл **Prefences.exe** - программа предназначенная для первоначальной конфигурации и калибровки источника смещения.

Архив **CDM20602.zip** - драйвер источника смещения для операционной системы Windows OS.