



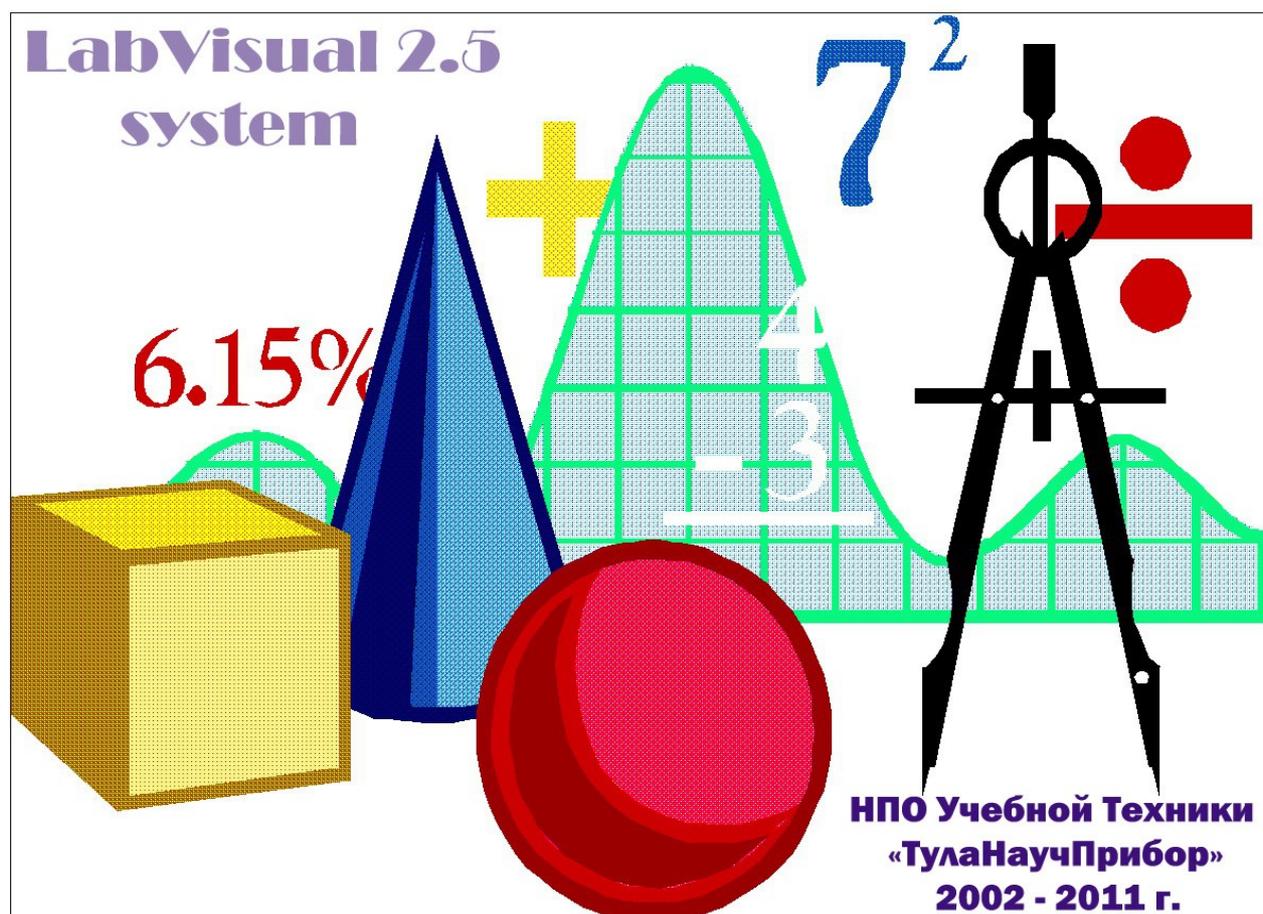
НПО УЧЕБНОЙ ТЕХНИКИ «ТУЛАНАУЧПРИБОР»

АВТОМАТИЗИРОВАННАЯ СРЕДА УЧЕБНОГО

LabVisual

ЛАБОРАТОРНОГО ПРАКТИКУМА

**РУКОВОДСТВО ПО ЭКСПЛУАТАЦИИ.
ОСНОВНЫЕ ПРАВИЛА РАБОТЫ.**



г. Тула, 2002 – 2011 г.

1. ПРОГРАММНАЯ ЧАСТЬ.

1.1. Свободное программное обеспечение.

Автоматизированная среда лабораторного практикума LabVisual, а также большинство компонент среды (**но не все компоненты**), в том числе драйвера уровня системы: USB драйвер, интерфейс RS232 работы с COM – портами ПК (персонального компьютера), интерфейсы работы со звуковой картой ПК и др.) и программы оболочки для работы с экспериментальными учебными установками являются свободным программным обеспечением.

Свободное программное обеспечение — широкий спектр программных решений, в которых права пользователя («свободы») на неограниченную установку, запуск, а также свободное использование, изучение, распространение и изменение (совершенствование) программ защищены юридически авторскими правами при помощи свободных лицензий. Среда LabVisual распространяется по стандартной лицензии GNU GPL v2, русский текст которой можно найти в этом руководстве в соответствующем разделе.

Для того чтобы сохранить модель научного сотрудничества между разработчиками, необходимо было обеспечить, чтобы исходные тексты программ, написанных разработчиками, оставались доступными для чтения и критики всему научному сообществу с сохранением авторства произведений. Для этого было сформулировано понятие свободного программного обеспечения, в котором отразились принципы открытой разработки программ в научном сообществе, сложившемся в американских университетах в 1970-е годы. Эти критерии оговаривают те права, которые авторы свободных программ передают любому пользователю:

- Программу можно свободно использовать с любой целью (*«нулевая свобода»*).
- Можно изучать, как программа работает, и адаптировать её для своих целей (*«первая свобода»*). Условием этого является доступность исходного текста программы.
- Можно свободно распространять копии программы— в помощь товарищу (*«вторая свобода»*).
- Программу можно свободно улучшать и публиковать свою улучшенную версию — с тем, чтобы принести пользу всему сообществу (*«третья свобода»*). Условием этой третьей свободы является доступность исходного текста программы и возможность внесения в него модификаций и исправлений.

Возможность исправления ошибок и улучшения программ— самая важная особенность свободного и открытого программного обеспечения, что просто невозможно для пользователей закрытых частных программ даже при обнаружении в них ошибок и дефектов, количество которых, как правило, неизвестно никому.

Только удовлетворяющая всем четырём перечисленным принципам

программа может считаться свободной программой, то есть гарантированно открытой и доступной для модернизации и исправления ошибок и дефектов, и не имеющей ограничений на использование и распространение.

1.2. Свободные операционные системы для персональных компьютеров на базе ядра GNU/Linux.

Linux (произносится «лiнукс», другие названия см. ниже)— общее название Unix-подобных операционных систем на основе одноимённого ядра и собранных для него библиотек и системных программ, разработанных в рамках проекта GNU.

Linux работает на множестве архитектур процессора таких как Intel x86, x86-64, PowerPC, ARM, Alpha AXP, Sun SPARC, Motorola 68000, Hitachi SuperH, IBM S/390, MIPS, HP PA-RISC, AXIS CRIS, Renesas M32R, Atmel AVR32, Renesas H8/300, NEC V850, Tensilica Xtensa и многих других.

В отличие от большинства других операционных систем, Linux не имеет единой «официальной» комплектации. Вместо этого Linux поставляется в большом количестве так называемых дистрибутивов, в которых ядро Linux соединяется с утилитами GNU и другими прикладными программами (например, X.org), делающими её полноценной многофункциональной операционной средой.

Наиболее известными дистрибутивами Linux являются Arch Linux, CentOS, Debian, Fedora, Gentoo, Mandriva, Mint, openSUSE, Red Hat, Slackware, Ubuntu.

Российские дистрибутивы— ALT Linux, ASPLinux, Calculate Linux, НауЛинукс, AgiliaLinux (ранее MOPSLinux), Runtu и Linux XP.

В отличие от коммерческих систем, таких как Microsoft Windows или Mac OS X, Linux не имеет географического центра разработки. Нет и организации, которая владела бы этой системой; нет даже единого координационного центра. Программы для Linux— результат работы тысяч проектов. Некоторые из этих проектов централизованы, некоторые сосредоточены в фирмах. Многие проекты объединяют программистов со всего света, которые знакомы только по переписке. Создать свой проект или присоединиться к уже существующему может любой и, в случае успеха, результаты работы станут известны миллионам пользователей. Пользователи принимают участие в тестировании свободных программ, общаются с разработчиками напрямую, что позволяет быстро находить и исправлять ошибки и реализовывать новые возможности.

История развития UNIX-систем. Linux является UNIX-совместимой, однако основывается на собственном исходном коде

Именно такая гибкая и динамичная система разработки, невозможная для проектов с закрытым кодом, определяет исключительную экономическую эффективность Linux. Низкая стоимость свободных разработок, отлаженные механизмы тестирования и распространения, привлечение людей из разных стран, обладающих разным видением проблем, защита кода лицензией GPL—

всё это стало причиной успеха свободных программ.

Конечно, такая высокая эффективность разработки не могла не заинтересовать крупные фирмы, которые стали открывать свои проекты. Так появились Mozilla (Netscape, AOL), OpenOffice.org (ORACLE), свободный клон InterBase (Borland)— Firebird, SAP DB (SAP). IBM способствовала переносу Linux на свои мейнфреймы.

С другой стороны, открытый код значительно снижает себестоимость разработки закрытых систем для Linux и позволяет снизить цену решения для пользователя. Вот почему Linux стала платформой, часто рекомендуемой для таких продуктов, как СУБД Oracle, DB2, Informix, SyBase, SAP R3, Domino.

Большинство пользователей для установки Linux используют дистрибутивы. Дистрибутив— это не просто набор программ, а ряд решений для разных задач пользователей, объединённых едиными системами установки, управления и обновления пакетов, настройки и поддержки.

Самые распространённые в мире дистрибутивы:

- [Ubuntu](#)— дистрибутив, основанный на [Debian](#) и быстро завоевавший популярность. Поддерживается сообществом, разрабатывается [Canonical Ltd](#). Основная сборка ориентирована на лёгкость в освоении и использовании, при этом существуют серверная и минимальная сборки.
- [Linux Mint](#)— дистрибутив, основанный на Ubuntu и полностью с ним совместимый, включающий в себя по умолчанию [Java](#), [Adobe Flash](#) и многое другое.
- [openSUSE](#)— дистрибутив, разрабатываемый сообществом при поддержке компании [Novell](#). Отличается удобством в настройке и обслуживании благодаря использованию утилиты [YaST](#).
- [Fedora](#)— поддерживается сообществом и корпорацией RedHat, предшествует выпускам коммерческой версии [RHEL](#).
- [Debian](#)— дистрибутив, разрабатываемый обширным сообществом разработчиков. Служит основой для создания множества других дистрибутивов. Отличается строгим подходом к включению несвободного ПО.
- [Mandriva Linux](#)— французско-бразильский дистрибутив, объединение бывших Mandrake и [Conectiva \(англ.\)](#).
- [Slackware](#)— один из старейших дистрибутивов, отличается консервативным подходом в разработке и использовании.
- [Gentoo](#)— дистрибутив, полностью собираемый из исходных кодов. Позволяет очень гибко настраивать конечную систему и оптимизировать производительность, поэтому часто называет себя мета-дистрибутивом. Ориентирован на экспертов и опытных пользователей.

Помимо перечисленных, существует множество других дистрибутивов, как базирующихся на перечисленных, так и созданных с нуля и зачастую предназначенных для выполнения ограниченного количества задач.

Каждый из них имеет свою концепцию, свой набор пакетов, свои достоинства и недостатки. Ни один не может удовлетворить всех пользователей, а потому рядом с лидерами благополучно существуют другие фирмы и объединения программистов, предлагающие свои решения, свои дистрибутивы, свои услуги. Существует множество LiveCD (запуск системы с CD – ROM даже в отсутствии жесткого диска ПК), построенных на основе Linux, например, [Knoppix](#). LiveCD позволяет запускать Linux непосредственно с компакт-диска, без установки на жёсткий диск.

1.3. ОС Ubuntu (Kubuntu) на базе ядра GNU/Linux.

Kubuntu — свободно распространяемая операционная система (ОС) основанная на ядре GNU/Linux.

Kubuntu разрабатывается сообществом при поддержке и спонсорской помощи Canonical Ltd. и является видоизменённой версией дистрибутива Ubuntu. Главное отличие этих двух дистрибутивов заключается в используемой по умолчанию среде рабочего стола, в Kubuntu это — KDE в Ubuntu – GNOME. Это различие не принципиально, фактически это одна и та же операционная система.

Kubuntu включает в себя более 1 000 единиц программного обеспечения и имеет доступ к более чем 25 000 различных приложений.

Kubuntu содержит все необходимые для работы программы от обработки текстов и электронных таблиц до приложений для работы в Интернете, веб-сервера, программ для работы с электронной почтой, языков программирования, и многое другое.

Официальный сайт проекта <http://kubuntu.org>

Русские сайты проекта <http://kubuntu.ru> <http://ubuntu.ru/>

Kubuntu использует прочную базу [Ubuntu](#) плюс последние версии [KDE](#). Мы участвуем в сообществе Убунту и используем её инфраструктуру и поддержку.

- Kubuntu всегда будет бесплатной, мы делаем наши самые лучшие работы доступными для всех на тех же свободных условиях.
- Kubuntu содержит самые лучшие переводы и доступную инфраструктуру, включающую в себя множество свободного программного обеспечения.
- Kubuntu полностью привержена принципам свободного программного обеспечения с открытыми исходными кодами, и мы призываем людей к использованию свободного и открытого программного обеспечения.

Какие плюсы имеет Кубунту перед множеством других дистрибутивов?

- Прогрессивная и удобная система управления пакетами (APT).
- Наличие LiveCD.
- Простая установка. Как в графическом LiveCD, так и в альтернативном текстовом режиме.
- Дистрибутив ориентирован на конечного пользователя. В нем легко

- разобраться даже ни разу не работавшему с Linux пользователю.
- Поддерживает большое количество всевозможного оборудования.
 - Тщательно подобранное программное обеспечение, входящее в дистрибутив.
 - Высокая интеграция программного обеспечения и, как следствие, высокая производительность системы в целом.
 - Большие репозитории программ, доступных для установки.
 - Мощная коммерческая поддержка проекта.
 - Постоянное совершенствование, выход нового релиза каждые 6 месяцев.

Для тех, кто желает сосредоточиться именно на том, для чего они используют компьютер, а не на используемом программном или аппаратном обеспечении, ищет возможность сделать работу более быстрой и продуктивной - Kubuntu лучший выбор.

Во время установки Kubuntu происходит автоматическая установка тщательно подобранного набора программного обеспечения для быстрого и комфортного начала работы на компьютере.

Если Вы планируете использовать Ваш компьютер для работы в Интернете, обработки текстов и работы с офисными приложениями, мультимедиа и т.д., Kubuntu имеет великое множество приложений, для решения различных задач.

Также на Kubuntu можно запускать приложения разработанные для других операционных систем, например, приложения для Mac OS и Windows.

Ниже приводится общий перечень эквивалентных программ в Kubuntu.

Браузер Internet Explorer или Safari:

Konqueror - браузер, входящий в базовую установку Kubuntu. Также в Kubuntu доступны для установки и использования [Firefox](#) и [Opera](#).

Microsoft Outlook, Outlook Express, и Apple's Mail:

Kontact - эквивалент Microsoft Outlook и Apple's Mail. Если весь функционал Kontact Вам не требуется, и Вы использовали Microsoft Outlook Express, то Вы можете использовать **KMail**. Mozilla's **Thunderbird** так же доступен для установки.

Microsoft Word:

OpenOffice.org Writer - редактор документов, входит в базовую установку Kubuntu.

Microsoft Excel:

OpenOffice.org Calc - редактор электронных таблиц, входит в базовую установку Kubuntu.

Microsoft PowerPoint:

OpenOffice.org Impress - редактор презентаций, входит в базовую установку Kubuntu.

Windows Media Player, Winamp и iTunes:

Amarok - аудио проигрыватель для Kubuntu. **Kaffeine** - видео

проигрыватель. Входят в базовую установку Kubuntu.

AIM, MSN, Yahoo, Jabber, ICQ, и другие:

Kopete - клиент для обмена мгновенными сообщениями. Входит в базовую установку Kubuntu.

Так же для установки доступны другие клиенты, такие как: sim, gaim, pidgin, licq, qutIm (подобен Qip).

mIRC и другие IRC-клиенты:

Konversation входит в базовую установку Kubuntu.

WinZIP или WinRAR:

Ark - архиватор, входит в базовую установку Kubuntu.

CD/DVD запись:

K3b входит в базовую установку Kubuntu.

Acrobat Reader, WinDjView:

Okular входит в базовую установку Kubuntu.

Delphi 5/6:

FPC & Lazarus ставятся из репозитория (не самая последняя версия).

Простой редактор графики, типа MS Paint:

Kolourpaint ставится из репозитория.

Простой редактор текста, типа Notepad:

Kate входит в базовую установку Kubuntu.

Профессиональный редактор растровой графики, типа Adobe Photoshop:

Krita, Gimp ставятся из репозитория.

Профессиональный редактор векторной графики, типа Corel Draw:

Karbon, Inkscape ставятся из репозитория.

Торрент-клиент, типа uTorrent:

Ktorrent ставится из репозитория.

Клиент пиринговой сети e2k, типа Emule:

Amule ставится из репозитория.

Программа для удобного чтения книг, типа Ice Book Reader:

Fbreader ставится из репозитория (не самая свежая версия).

Графический двухпанельный менеджер файлов, типа Total Commander:

Krusader ставится из репозитория.

Специальная программа для тестирования монитора, типа Nokia Monitor Test:

Screentest ставится из репозитория.

Редактор нот и табулатур, типа Guitar Pro:

KGuitar, TuxGuitar ставятся из репозитория.

Студийный аудиоредактор, типа Adobe Audition, Sony SoundForge, CoolEdit Pro:

Audacity ставится из репозитория.

1.4 Установка ОС Kubuntu на ПК.

Дистрибутив Kubuntu (рабочий стол KDE) более близок к операционной системе Windows потому он и предлагается для использования. Все программное обеспечение и сам дистрибутив Kubuntu абсолютно бесплатны и нет необходимости платить кому-либо лицензионные отчисления.

Желательные системные требования для системы Kubuntu:

Процессор с частотой не менее 1000 МГц,

объем оперативной памяти 512 Мб,

видеокарта с объемом памяти 16 Мб.

Для того, чтобы начать установку Kubuntu необходимо установить в BIOS'е загрузку с CD и вставив диск с Kubuntu перезагрузиться. Вход в BIOS ПК осуществляется нажатием клавиши F2 (в некоторых модификациях BIOS для входа в меню следует нажимать клавишу DEL).

Настоятельно рекомендуется ознакомиться с документацией к материнской плате Вашего компьютера, из которой Вы и сможете узнать об основных функциях BIOS и о том, как изменить параметры загрузки.



Рис. 1.1

Начав загрузку с диска, Вы увидите меню параметров загрузки (рис. 1.1), но уже не BIOS, а Kubuntu. Если Вы ничего не предпримите то загрузка Kubuntu начнется автоматически через определенный промежуток времени.

Используя меню измените язык интерфейса с английского на родной в данном случае русский. Для этого нам достаточно нажать клавишу F2 и из появившегося списка выбрать то, что нам подходит. Для принятия изменений просто нажимаем клавишу Enter.

Установить подходящее разрешение экрана. Для этого нажмите клавишу F4 и в появившемся меню выбираем подходящий вариант. Не забудьте нажать Enter. Если Вы не знаете какой вариант наиболее подходящий для Вас, Вы можете испробовать разные. В случае неудачи Вы сможете легко произвести перезагрузку. Но все же желательно знать необходимые параметры, которые Вы можете уточнить, сверившись с документацией к Вашему монитору, если Вы по какой либо причине не хотите рисковать, то можете просто пропустить этот шаг. Теперь можно приступить непосредственно к запуску Kubuntu, для этого просто нажимаем клавишу Enter.

Обратите внимание, что DVD диск с ОС Kubuntu, поставляемый в комплекте с персональным компьютером и учебными установками содержит и другие пункты меню. Отметим возможность установки ОС в «текстовом режиме», при этом во время процесса установки функции графической подсистемы не используются, в то же время, этот режим предоставляет немного больше возможностей и рекомендуется для достаточно опытных пользователей. Поэтому далее рассмотрим вариант графической установки.

Наберитесь терпения, загрузка может потребовать немного времени, все зависит от конкретной конфигурации оборудования, количества оперативной памяти (кстати, ОС Linux предъявляет большие требования к объему оперативной памяти, нежели к мощности процессора) и т. д.

Дождавшись окончания загрузки Вы увидите перед собой рабочий стол Kubuntu. Вы уже можете использовать Linux просто работая с диска (режим Live CD). В современных версиях установщика сразу после загрузки вы увидите меню с предложением «Попробовать Kubuntu» либо «Установить Kubuntu на жесткий диск». При этом первый вариант означает запуск ОС в режиме LiveCD.

После загрузки LiveCD вы увидите рабочий стол с ярлыком Install. Кликните мышью по ярлыку и начните установку ОС. Проявляется окно выбора страны рис 1.2:



Рис. 1.2

Российская федерация в этом списке присутствует. Далее необходимо выбрать город рис. 1.3.

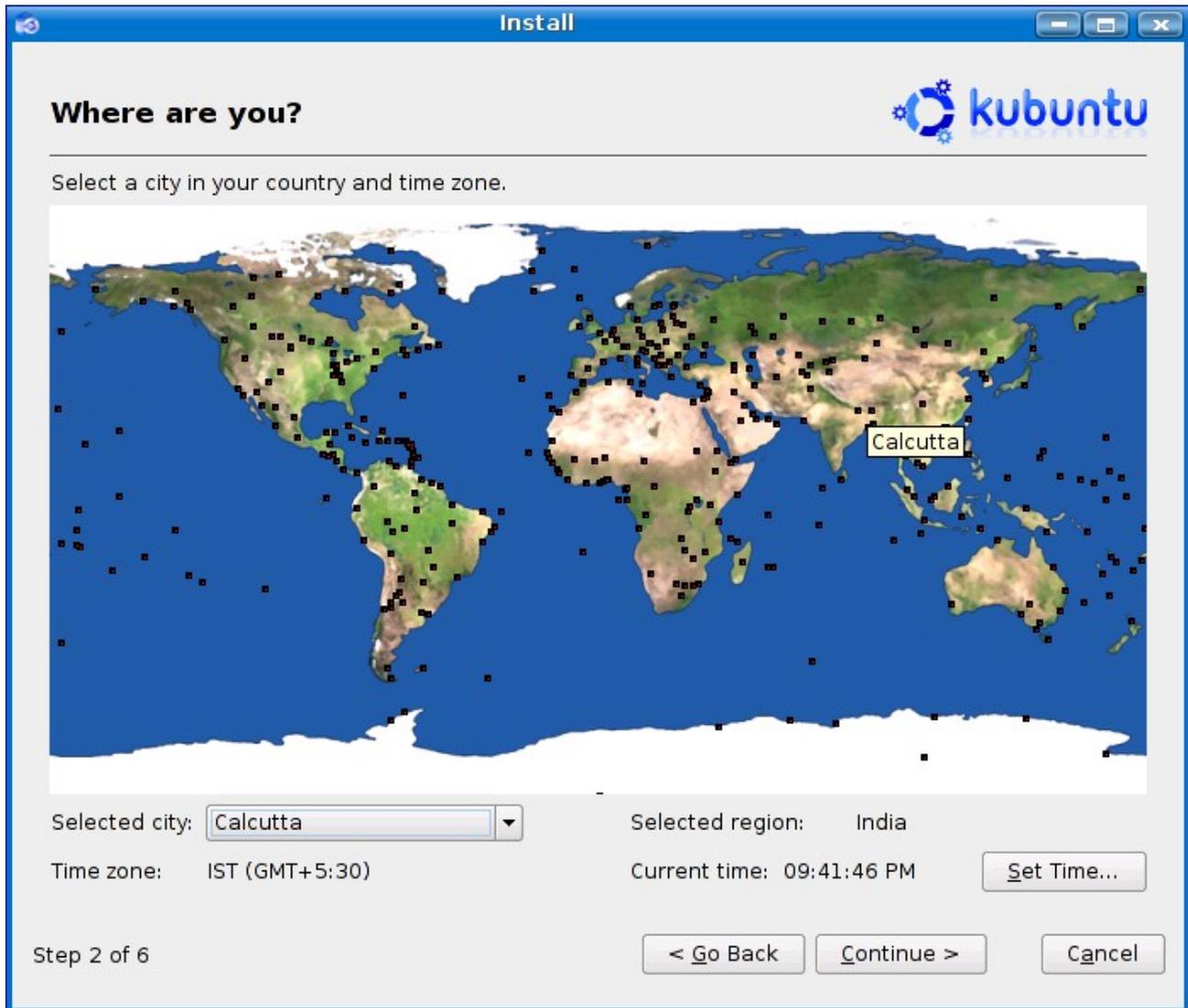


Рис. 1.3

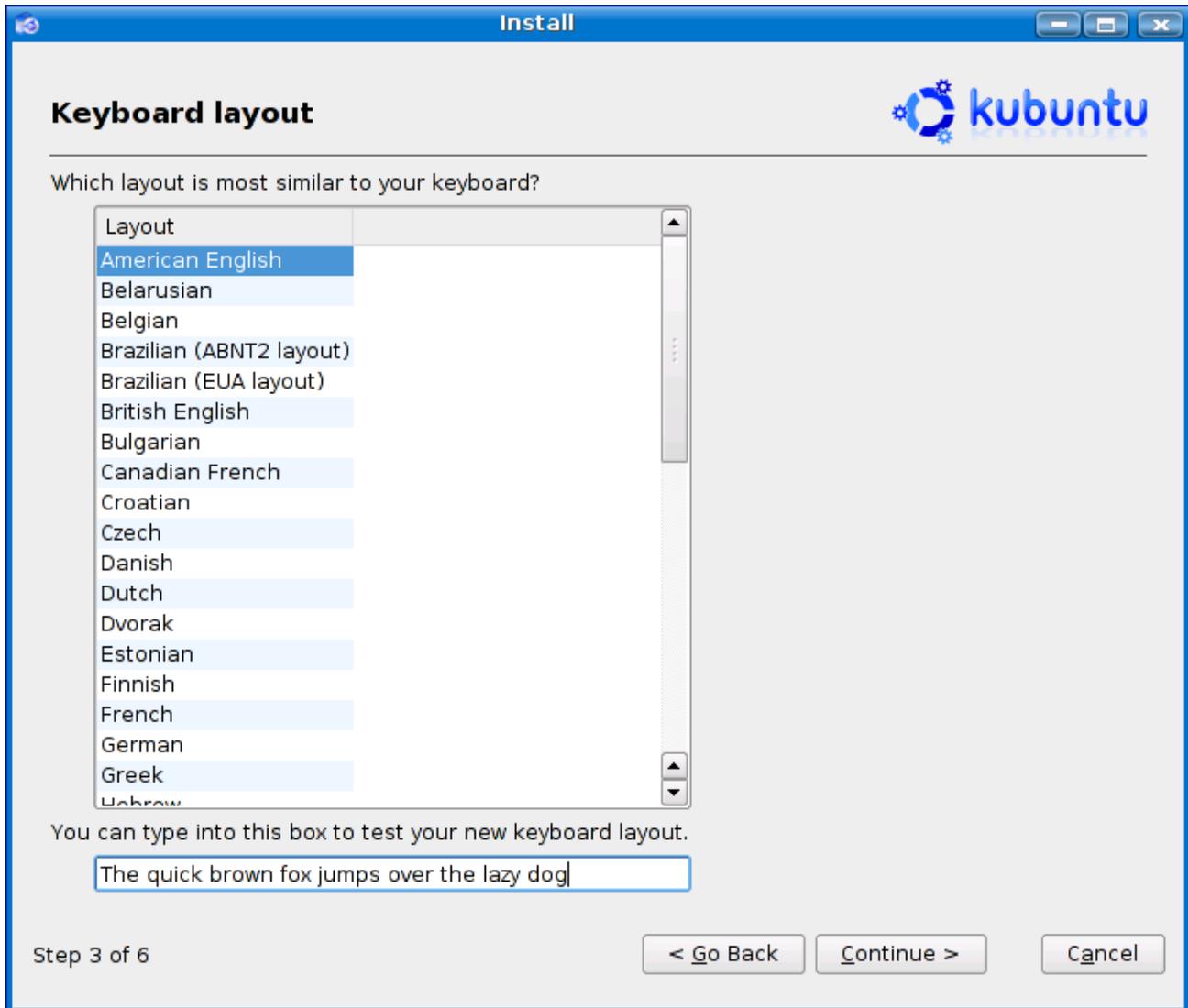


Рис. 1.4

После этого выбирается язык ввода и раскладка клавиатуры. Необходимо выбрать Russian (winkeys) рис. 1.4.

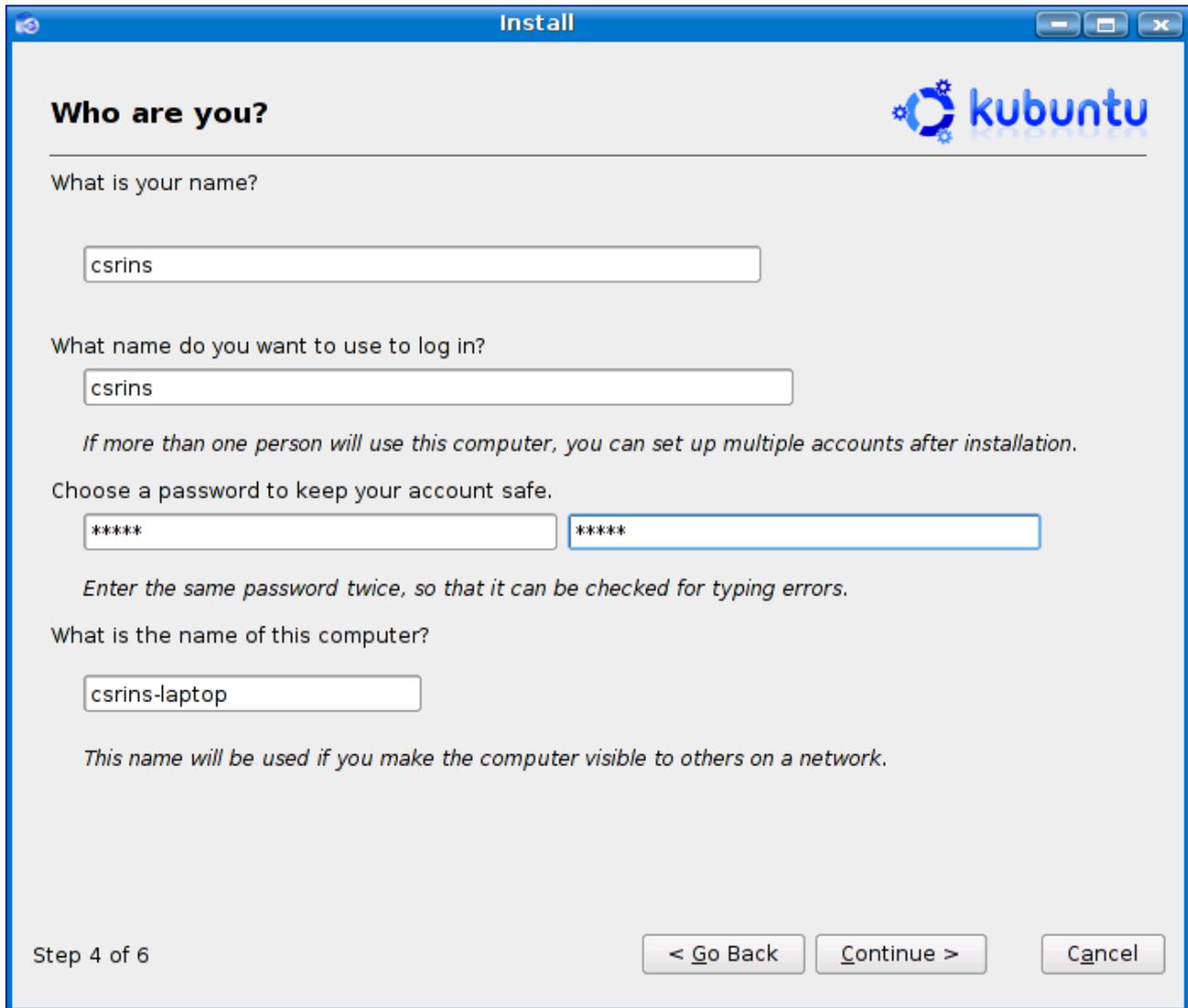


Рис. 1.5

После этого вводится информация о первом пользователе, пароле и имени компьютера рис. 1.5.

Следующее окно рис. 1.6 касается разбивки жесткого диска. Состоит из 3-х пунктов:

1. Изменить размер существующего раздела (если он есть). Индикатор с бегунком показывает новый объем раздела, который уменьшается
2. Стереть весь диск и использовать его полностью под Kubuntu
3. Разбить жесткий диск вручную.

Начинающим пользователям рекомендуется использовать первый или второй из предложенных пунктов. При выборе первого пункта предоставляется возможность использования на компьютере двух (и более) операционных систем. Установщик автоматически создаст в конце установки меню выбора операционной системы при последующей загрузке.

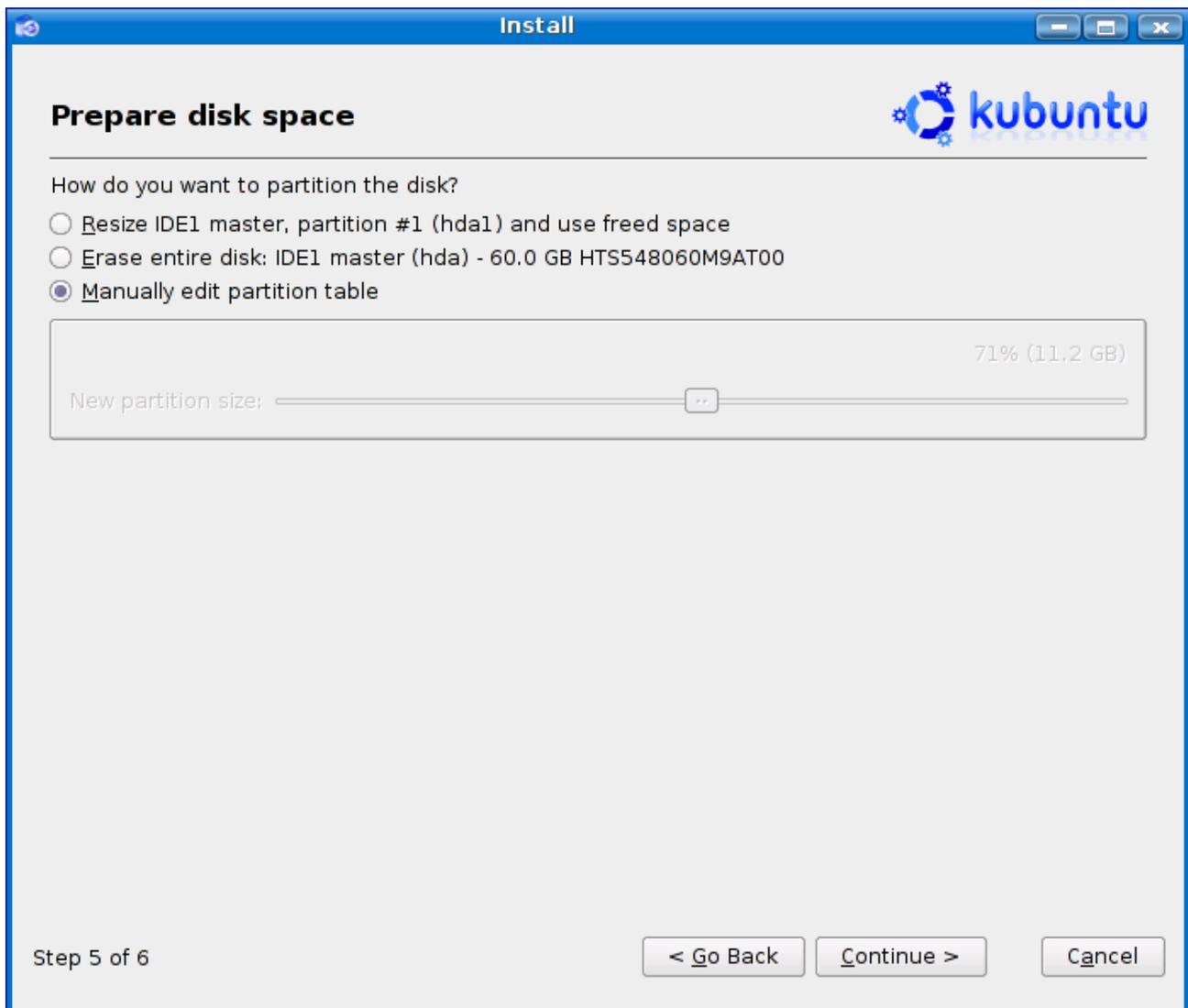


Рис. 1.6

Помните резервное копирование данных один из лучших на сегодняшний день способов защитить свои данные от порчи/потери. **Если Вы хотите разметить диск вручную то Вы должны быть уверены в том что Вы делаете в противном случае Вы рискуете потерять все данные, хранящиеся на жестком диске компьютера.**

При разметке диска вручную укажите точку монтирования корневой файловой системы (обозначается «/»), в качестве используемой файловой системы мы рекомендуем выбрать ext3. При размере оперативной памяти меньше 2 Гб, рекомендуется создать раздел «подкачки» - SWAP. В данном случае термин SWAP означает один из способов представления виртуальной памяти, при этом подразумевается, что данные из виртуальной памяти частично располагаются на внешних запоминающих устройствах (жестком диске).

Последнее окно установки содержит все собранную от пользователя информацию и запрос на ее подтверждение рис. 1.7.

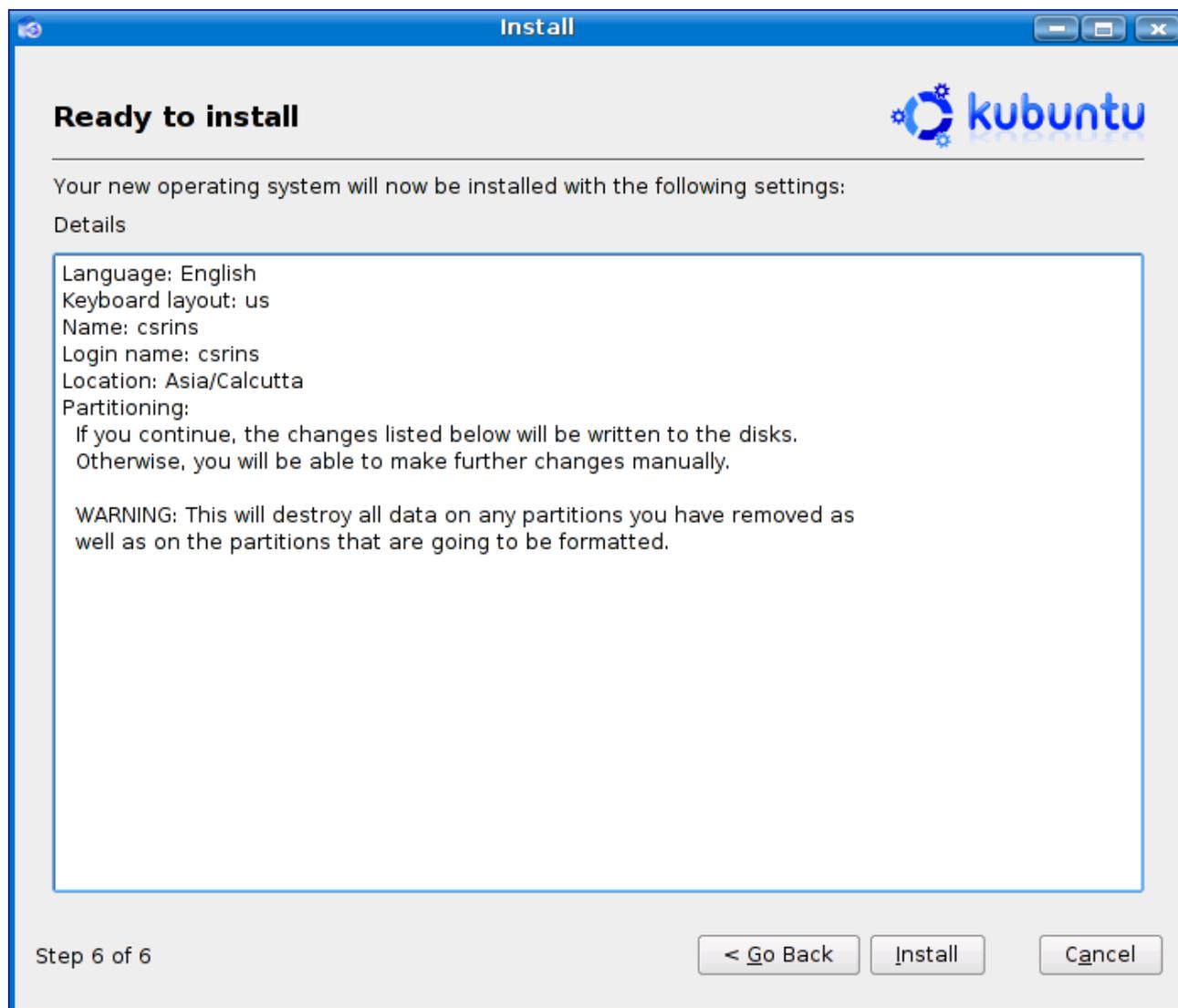


Рис. 1.7

По окончании установки в появившемся меню необходимо нажать кнопку «Restart now», изъять из привода компьютера установочный компакт-диск и перезагрузить компьютер. На этом установка Kubuntu завершена.

1.5. Kubuntu ОС. Краткое руководство пользователя.

Установленной системы в базовой конфигурации, процесс инсталляции которой описан в предыдущем пункте вполне достаточно для последующего развертывания комплекта программ LabVisual.

Однако кратко опишем основные приемы работы с операционной системой.

В Kubuntu можно выполнять задачи как из графического пользовательского интерфейса (GUI) с помощью меню, так и из командной строки (CLI) в текстовом режиме. В Kubuntu терминал с командной строкой называется Konsole (Консоль) и запускается из: К-меню -> Система -> Терминал Konsole. "К" или "К-меню" - это нижняя левая (или верхняя левая) кнопка, похожая на кнопку "Пуск" в Microsoft Windows®.

В настоящем руководстве текст, оформленный как этот, необходимо вводить или копировать в терминал Konsole.

- Многие изменения в операционной системе может сделать только пользователь с правами администратора. `sudo` временно поднимает права пользователя до уровня администратора (например, при установке программ или осуществлении изменений в системе). Пример:

```
sudo bash
```

- Необходимо использовать `kdesudo` вместо `sudo` при открытии приложений с графическим интерфейсом, таких как графический текстовый редактор `kate`, с помощью диалога "Выполнить команду" или из командной строки. Пример:

```
kdesudo kate /etc/apt/sources.list
```

- Команда `man` используется для вызова справки по указанной команде. Например, `man sudo` покажет описание команды `sudo`.
- Хотя с помощью `apt-get` и `aptitude` можно быстро установить пакеты программ, также можно использовать и KPackageKit (или диспетчер пакетов Adept или Synaptic) - графический способ установки программ. Большинство пакетов (но не все), доступных для установки с помощью `apt-get`, также есть и в KPackageKit (или в Adept или Synaptic). Если в настоящем руководстве написано

```
sudo apt-get install package
```

- ,то можно будет найти пакет `package` в KPackageKit (или в Adept, или в Synaptic) и установить его оттуда.
- Во многих инструкциях используется текстовый редактор `nano`, который есть во всех релизах на базе Linux. Однако, в Kubuntu часто удобнее

использовать вместо него текстовый редактор kate.

- Для 64-разрядной версии следует заменять все "i386" на "amd64".

Определение используемой версии Kubuntu

Следует ввести в Konsole:

```
lsb_release -a
```

Определение используемой версии ядра

```
uname -r
```

Добавление дополнительных репозитариев в Kubuntu

Пакеты программ доступны для свободного скачивания с нескольких сайтов, имеющих стандартизованную структуру, которые называются репозитариями. Существуют репозитарии, официально утверждённые и контролируемые сообществом разработчиков Kubuntu и Ubuntu, также есть другие репозитарии, которые являются независимыми и не имеющими официального одобрения и контроля (должны использоваться с осторожностью).

Типы репозитариев

Существует четыре типа главных репозитариев пакетов в Kubuntu:

- main (основной) - поддерживается и контролируется компанией Canonical, является главной частью дистрибутива.
- restricted (ограниченный) - ПО, которое распространяется не по лицензии GPL или подобной ей лицензии и поддерживается и контролируется компанией Canonical.
- universe (вселенский) - ПО, которое распространяется по лицензии GPL или подобной ей лицензии и поддерживается пользователями.
- multiverse (мультивселенский) - ПО, которое распространяется не по лицензии GPL или подобной ей лицензии и поддерживается пользователями.

Также имеются следующие дополнительные типы репозитариев:

- karmic-updates - обновления для официальных пакетов.
- karmic-backports - текущие версии ПО из Karmic+1 ("Светящейся рыси"), которые были портированы обратно на "Кармическую коалу".
- karmic-proposed - предлагаемые обновления и изменения (новейшие).

Сторонние репозитарии

Разработчики ПО часто создают собственные репозитарии, из которых пакеты программ могут скачиваться и устанавливаться непосредственно на компьютер пользователя (если [добавить данный репозитарий](#) в список). Многие из таких

сторонних репозитариев и пакетов программ никогда не проверялись сообществом (K)Ubuntu и Debian и могут представлять собой угрозу для безопасности компьютера пользователя. Троянские программы, программы, создающие лазейки, и прочее вредоносное ПО могут находиться в любом неконтролируемом репозитарии. Для использования репозитариев, не одобренных сообществом (K)ubuntu или Debian, необходимо быть абсолютно уверенным в этом сайте, прежде чем разрешать этот репозитарий и устанавливать пакеты оттуда.

Добавление репозитариев с помощью программы Package Manager

Это рекомендованный способ.

- К-меню -> Система -> Настройки системы -> Добавление и удаление программ -> Настройки -> Редактирование источников получения ПО.
- Тут можно включить репозитарии для ПО Kubuntu и сторонних приложений.
- Для сторонних программ следует выбрать Добавить -> и ввести адрес репозитария. Формат будет похож на следующий:

```
deb http://archive.ubuntu.com/ubuntu/ karmic main restricted
deb-src http://archive.ubuntu.com/ubuntu/ karmic main restricted
```

- Пример: для добавления репозитария Medibuntu, нужно добавить:


```
deb http://packages.medibuntu.org/ karmic free non-free
```
- Скачать ключ репозитария в папку.
 - Пример: ключ Medibuntu можно скачать с


```
http://packages.medibuntu.org/medibuntu-key.gpg
```
- Затем добавить ключ из:

К-меню -> Система -> Настройки системы -> Добавление и удаление программ -> Настройки -> Редактирование источников получения ПО -> Аутентификация -> Импортировать файл ключа...

 - В качестве другого варианта можно вручную добавить ключ с помощью командной строки из терминала Konsole. См. подраздел "[Добавление ключей репозитариев](#)".
- Обновить список пакетами из нового репозитария:

К-меню -> Система -> Настройки системы -> Добавление и удаление программ -> Обновления ПО -> Обновить

Добавление репозитариев вручную

Это связано с определённым риском. Стандартный `sources.list` следует изменять в Kubuntu только тогда, когда есть понимание последствий осуществляемых действий. Одновременное использование нескольких репозитариев может привести к сбоям в работе системы.

- Создать резервную копию текущего списка источников.

```
sudo cp -p /etc/apt/sources.list /etc/apt/sources.list_backup
```

Примечание: `sudo` - выполнение программы с правами суперпользователя, `cp` - команда копирования, `-p` - перезаписывать файл, если он уже существует, без дополнительного подтверждения.

- Отредактировать список источников:

```
sudo nano /etc/apt/sources.list
```

или используя графический редактор:

```
kdesu kate /etc/apt/sources.list
```

Примечание: для того, чтобы использовать локальное зеркало репозитария, необходимо добавить "xx." перед `archive.ubuntu.com`, где xx = код страны. Пример:

```
deb http://gb.archive.ubuntu.com/ubuntu karmic main restricted universe
multiverse
```

указывает на репозитарий для Великобритании (gb).

- Ниже приведён образец `sources.list`. В конце были добавлены репозитарии для Medibuntu и Google:

```
#deb cdrom:[Kubuntu 9.10 _Karmic Koala_ - Release amd64 (20091027)]/
karmic main restricted
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://ru.archive.ubuntu.com/ubuntu/ karmic main restricted
deb-src http://ru.archive.ubuntu.com/ubuntu/ karmic main restricted
## Major bug fix updates produced after the final release of the
## distribution.
deb http://ru.archive.ubuntu.com/ubuntu/ karmic-updates main restricted
deb-src http://ru.archive.ubuntu.com/ubuntu/ karmic-updates main
restricted
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the
Ubuntu
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
deb http://ru.archive.ubuntu.com/ubuntu/ karmic universe
deb-src http://ru.archive.ubuntu.com/ubuntu/ karmic universe
deb http://ru.archive.ubuntu.com/ubuntu/ karmic-updates universe
deb-src http://ru.archive.ubuntu.com/ubuntu/ karmic-updates universe
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the
```

```

Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as
to
## your rights to use the software. Also, please note that software in
## multiverse WILL NOT receive any review or updates from the Ubuntu
## security team.
deb http://ru.archive.ubuntu.com/ubuntu/ karmic multiverse
deb-src http://ru.archive.ubuntu.com/ubuntu/ karmic multiverse
deb http://ru.archive.ubuntu.com/ubuntu/ karmic-updates multiverse
deb-src http://ru.archive.ubuntu.com/ubuntu/ karmic-updates multiverse
## Uncomment the following two lines to add software from the 'backports'
## repository.
## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it
includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any
review
## or updates from the Ubuntu security team.
# deb http://ru.archive.ubuntu.com/ubuntu/ karmic-backports main
restricted universe multiverse
# deb-src http://ru.archive.ubuntu.com/ubuntu/ karmic-backports main
restricted universe multiverse
## Uncomment the following two lines to add software from Canonical's
## 'partner' repository.
## This software is not part of Ubuntu, but is offered by Canonical and
the
## respective vendors as a service to Ubuntu users.
# deb http://archive.canonical.com/ubuntu karmic partner
# deb-src http://archive.canonical.com/ubuntu karmic partner
deb http://security.ubuntu.com/ubuntu karmic-security main restricted
deb-src http://security.ubuntu.com/ubuntu karmic-security main restricted
deb http://security.ubuntu.com/ubuntu karmic-security universe
deb-src http://security.ubuntu.com/ubuntu karmic-security universe
deb http://security.ubuntu.com/ubuntu karmic-security multiverse
deb-src http://security.ubuntu.com/ubuntu karmic-security multiverse
## Medibuntu - Ubuntu 9.10 "Karmic Koala"
## Please report any bug on https://bugs.launchpad.net/medibuntu/
deb http://packages.medibuntu.org/ karmic free non-free
deb-src http://packages.medibuntu.org/ karmic free non-free
# Google software repository
deb http://dl.google.com/linux/deb/ stable non-free

```

- Скачать и добавить ключи репозитариев в хранилище ключей. См. подраздел "[Добавление ключей репозитариев](#)".
- Обновить список пакетами из новых репозитариев:

```
sudo apt-get update
```

Добавление ключей репозитариев

- Скачать gpg-ключи репозитариев и сразу добавить их в хранилище ключей репозитариев:
 - Пример: для получения и добавления ключа к репозитарию Medibuntu:

```
wget --quiet http://packages.medibuntu.org/medibuntu-key.gpg -O - |
```

```
sudo apt-key add -
```

- Пример: для получения и добавления ключа к репозитарию Google:

```
wget --quiet https://dl-ssl.google.com/linux/linux_signing_key.pub  
-O - | sudo apt-key add -
```

Примечание: `wget` - скачивает файл, находящийся в сети; `--quiet` - не выводить сообщения; `-O -` - показать скачанный файл на экране.

Символ конвейера `|` используется для передачи результата выполнения предшествующей команды в качестве входных данных для последующей команды (т.е. для команды `apt-key`, которая добавляет ключ в хранилище).

- Как вариант (видимо, более простой), можно использовать команду `apt-key` отдельно:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys КЛЮЧ
```

, где КЛЮЧ - код ключа, который был показан как результат команды `apt-get`, например `EF4186FE247510BE`.

Установка и обновление пакетов программ в Kubuntu Основные сведения о программе Apt и пакетах программ

- Прочитать раздел "[Добавление дополнительных репозитариев в Kubuntu](#)"

Большинство новых пользователей используют [KPackageKit](#) или [диспетчер пакетов Adept](#) для установки пакетов. Нижеприведённые указания даны для установки пакетов из [командной строки терминала Konsole](#).

- Установить пакеты:

```
sudo apt-get install название-пакета
```

- Пример:

```
sudo apt-get install mpd sbackup
```

- Удалить пакеты:

```
sudo apt-get remove название-пакета
```

- Удалить все зависимые пакеты:

```
sudo apt-get autoremove
```

- Пример:

```
sudo apt-get remove mpd sbackup
```

- Найти пакеты:

```
sudo apt-cache search <ключевые слова>
```

- Примеры:

```
sudo apt-cache search Музыка MP3
sudo apt-cache search "Текстовый редактор"
```

- Обновить базы данных пакетов в программе Apt после [добавления или удаления репозитариев](#):

```
sudo apt-get update
```

- Установить все имеющиеся новые версии пакетов:

```
sudo apt-get upgrade
```

- Модернизация дистрибутива, например версии 9.04 до 9.10:

```
sudo apt-get dist-upgrade
```

Установка пакетов с расширением .deb

В Ubuntu/Kubuntu используются пакеты Debian (.deb). Можно установить на систему любой такой пакет. deb-файлы, в принципе, могут устанавливаться из файлового менеджера (Konqueror или Dolphin) простым нажатием на них кнопкой мыши, поскольку они сразу привязаны к установщику, используемому в Kubuntu по умолчанию. Эти указания даны для установки пакетов из командной строки терминала Konsole.

- Установить скачанный пакет Debian (Ubuntu или Kubuntu) с расширением .deb:

```
sudo dpkg -i название-пакета.deb
```

- Удалить скачанный пакет Debian (Ubuntu или Kubuntu) с расширением .deb:

```
sudo dpkg -r название-пакета
```

- Переконфигурировать или исправить установленный пакет Debian (Ubuntu или Kubuntu) с расширением .deb:

```
sudo dpkg-reconfigure название-пакета
```

- Пример:

```
sudo dpkg-reconfigure mpd
```

Работа с архивами Tar/GZip и Tar/Bzip2

Архивы Tar/GZip имеют расширение `.tar.gz`, а архивы Tar/Bzip2 - `.tar.bz2`. Bzip2 - более новый и эффективный метод сжатия. Эти файлы, в принципе, автоматически распаковываются простым нажатием на них кнопкой мыши, поскольку они сразу привязаны к соответствующим утилитам архивирования, используемым в Kubuntu по умолчанию. Эти указания даны для установки пакетов из командной строки терминала Konsole.

- Распаковать:

```
tar xvf название-пакета.tar.gz
```

Примечание: `tar` - это приложение, которое распаковывает файлы из архивов и, если это необходимо, разжимает их. `-x` означает распаковать; `-v` означает выводить информацию о распаковываемых файлах; `-f` указывает на файл, который нужно распаковать.

- Разжать файлы `.gz`:

```
gunzip файл.gz
```

- Разжать файлы `.bz2`:

```
bunzip2 файл.bz2
```

Примечание: также можно сначала разжать пакет с помощью команды `gunzip` (для `.gz`) или `bunzip2` (для `.bz2`), получив файл `.tar`. Потом он распаковывается с помощью команды `tar`.

- Создать архив `.gz`:

```
tar cvfz название-пакета.tar.gz папка
```

- Создать архив `.bz2`:

```
tar cvfj название-пакета.tar.bz2 папка
```

Установка пакетов из исходного кода

- Убедиться, что все необходимые инструменты (т.е. библиотеки, компиляторы, заголовочные файлы) установлены:

```
sudo apt-get install build-essential
sudo apt-get install linux-headers-`uname -r`
```

Примечание: функция `uname -r` возвращает версию текущего используемого ядра

- Распаковать файлы с исходным кодом из архива:

```
tar xvf архив-с-исходным-кодом.tar.gz
```

- Собрать пакет, используя сценарий пакета (в данном случае сценарий `configure`), скомпилировать пакет (`make`) и установить скомпилированный пакет на систему (`make install`):

```
cd /путь/к/распакованным/файлам-с-исходным-кодом
./configure
sudo make
sudo make install
```

Примечание: добавляя `./` перед именем файла в текущей папке, можно выполнить файл как приложение, даже если он находится не в одной из папок, в которых осуществляется поиск программы для введённого имени команды. Если выскакивает ошибка "в доступе отказано", то файл не помечен как исполняемый. Для исправления надо ввести:

```
sudo chmod +x имя-файла
```

Пример: В вышеприведённых указаниях `configure` - это сценарий для командного интерпретатора для сборки пакета из исходного кода. Чтобы убедиться, что он является исполняемым, нужно ввести:

```
sudo chmod +x configure
```

Создание пакета `.deb` из файлов с исходным кодом

Если сборка из исходного кода прошла успешно, то можно сделать пакет Debian (Ubuntu или Kubuntu) с расширением `.deb` для дальнейшего использования:

- Установить инструменты для создания пакетов:

```
sudo apt-get install checkinstall
```

- Пересобрать пакет с помощью `checkinstall`:

```
cd /путь/к/распакованному/пакету
./configure
sudo make
sudo checkinstall
```

- Сохранить получившийся файл `.deb` для дальнейшего использования. Он позднее может устанавливаться так:

```
sudo dpkg -i название-пакета.deb
```

Примечание: Это были основные указания, которые не всегда могут работать. Для некоторых пакетов требуется, чтобы были установлены зависимые пакеты и заданы дополнительные параметры прежде чем удастся успешно собрать их.

Программа Aptitude

Программа Aptitude - это диспетчер пакетов, работающий в терминальном режиме, который можно использовать вместо apt-get. Она отмечает пакеты, которые устанавливаются автоматически, и удаляет их, если ни один пакет больше не зависит от них. Это позволяет легко удалять пакеты полностью. Для использования программы Aptitude нужно заменять apt-get на aptitude в командной строке. Примеры:

```
sudo aptitude install название-пакета
sudo aptitude remove название-пакета
sudo aptitude update
sudo aptitude upgrade
```

Для использования графического пользовательского интерфейса ncurses необходимо ввести:

```
sudo aptitude
```

Терминал Yakuake

Yakuake - это стильный терминал с командной строкой, который можно использовать вместо Konsole.

```
sudo apt-get install yakuake
```

Запускается так: К-меню -> Приложения → Система -> Yakuake

Появляется на экране при нажатии на клавишу F12.

Файловый менеджер Krusader

Krusader - **Krusader** — развитый двухпанельный файловый менеджер для оболочки KDE. Изначально разрабатывался для Linux, но теперь доступен для некоторых других платформ

```
sudo apt-get install krusader
```

Запускается так: К-меню -> Приложения → Служебное -> Krusader

либо из консоли командой:

```
krusader
```

1.6. Установка и настройка среды LabVisual в ОС Kubuntu Linux.

Эмулятор программного и аппаратного кода VirtualBox.

Среда автоматизированного лабораторного практикума LabVisual является кросс-платформенным программным продуктом. Это означает, что окружение LabVisual может успешно работать как на компьютерах под управлением ОС Windows так и на компьютерах под управлением ОС Linux при помощи эмулятора среды окружения VirtualBox. На прилагаемых дисках в соответствующих папках содержатся сборки программ для установки и работы в этих операционных системах.

ПЭВМ поставляется в комплекте с установленным и настроенным программным обеспечением (ПО), пользователю как правило не требуется вмешиваться в работу устройств и ПО. ОС семейства Linux являются мощными, удобными и защищенными от сбоев ОС, однако при возникновении ошибок и нестандартных ситуаций может возникнуть необходимость в переустановке ОС, эмулятора программного кода VirtualBox и среды LabVisual.

Далее приведен порядок действий по установке и настройке ПО для работы с установкой на базе ОС Linux дистрибутива Kubuntu. Установка самой ОС была рассмотрена в пп. 1.4 — 1.5. Во время установки пакетов предполагается наличие подключения к сети Интернет. Имя созданного при установке пользователя предполагается `pankov`. Имя домашней директории, соответственно `/home/pankov`.

После установки ОС Ubuntu на компьютер необходимо установить и настроить программный комплекс LabVisual для работы. Для этого в среде ОС Linux необходимо установить эмулятор программного кода - среду VirtualBox.

В ОС Kubuntu необходимые приложения как правило устанавливаются несколькими способами:

1. Установка из специальных хранилищ ПО в интернете либо из хранилища на локальном диске. Данные сервера ПО называются репозитории. Список репозиторий, доступных в данной системе можно посмотреть в файле `/etc/apt/sources.list`. Для визуального перемещения по каталогам, просмотра и запуска файлов можно использовать графический двухпанельный файловый менеджер Krusader.

2. Установка программ из пакетов `*.deb`. В консоли следует перейти в папку (например, команда `cd /home/pankov/имя_папки` и нажать ENTER) содержащую пакет `name.deb` (`name` - имя файла) и набрать `sudo dpkg -i name.deb`. Если в системе установлены все необходимые зависимости для работы пакета, то он успешно установится.

3. Установка ПО из исходников. Требуется компиляции исходных текстов программы.

Перед установкой пакета VirtualBox необходимо выяснить текущую архитектуру операционной системы, установленной на ПК. Сделать это можно, набрав в консоли команду:

```
uname -a
```

В результате выполнения команды, в текстовом выводе командной строки будут указаны различные параметры системы, в том числе архитектура ОС. Для 32-х битных систем вывод командной строки имеет примерно следующий вид:

```
Linux server 2.6.31-20-generic #58-Ubuntu SMP Fri Mar 12 05:23:09 UTC 2010 i686
GNU/Linux
```

Значение `i686 (i386, i586)` указывает на то, что на ПК установлена 32 — х битная ОС.

Для 64-х битных систем вывод командной строки имеет примерно следующий вид:

```
Linux ivan-desktop 2.6.32-25-generic #45-Ubuntu SMP Sat Oct 16 19:52:42 UTC 2010
x86_64 GNU/Linux
```

Значение `x86_64 (amd64)` указывает на то, что на ПК установлена 64 — х битная ОС.

Наиболее свежую версию пакета VirtualBox можно найти на сайте <http://www.virtualbox.org/wiki/Downloads> Для обычной работы с ПК под ОС Linux следует выбрать VirtualBox for Linux Hosts, соответствующую архитектуру установленной ОС (i686 или amd64) и пакет *.deb для ОС Ubuntu.

Перед установкой пакета VirtualBox, следует установить заголовки ядра ОС командой в командной строке:

```
sudo apt-get install linux-headers-`uname -r`
```

и средства компиляции (пакет build-essential):

```
sudo apt-get install build-essential
```

Затем, скачанный вручную пакет VirtualBox необходимо установить от имени суперпользователя root командой в консоли `sudo dpkg -i name.deb` (в качестве name использовать имя конкретного пакета, команда требует ввода пароля, который не отображается). Для удаления старой версии пакета можно использовать команду `sudo dpkg -r name`. Реальное имя пакета name, подлежащего удалению, включающее номер версии, можно узнать командой `dpkg -S VirtualBox`.

Для работы USB подсистемы следует скачать с официального сайта <http://www.virtualbox.org/wiki/Downloads> пакет расширений **VirtualBox Oracle VM VirtualBox Extension Pack**.

После установки необходимо запустить эмулятор VirtualBox. Запуск эмулятора осуществляется командой `VirtualBox` из консоли. Либо:

К-меню -> Приложения → Система -> VirtualBox

Для корректной работы USB протокола необходимо установить в VirtualBox загруженный ранее плагин **VirtualBox Oracle VM VirtualBox Extension Pack**. Для этого в главном окне эмулятора VirtualBox выбрать ФАЙЛ-> НАСТРОЙКИ -> ПЛАГИНЫ и нажать кнопку «Добавить Плагин» рис. 1.8 и выбрать находящийся на жестком диске загруженный файл плагина.

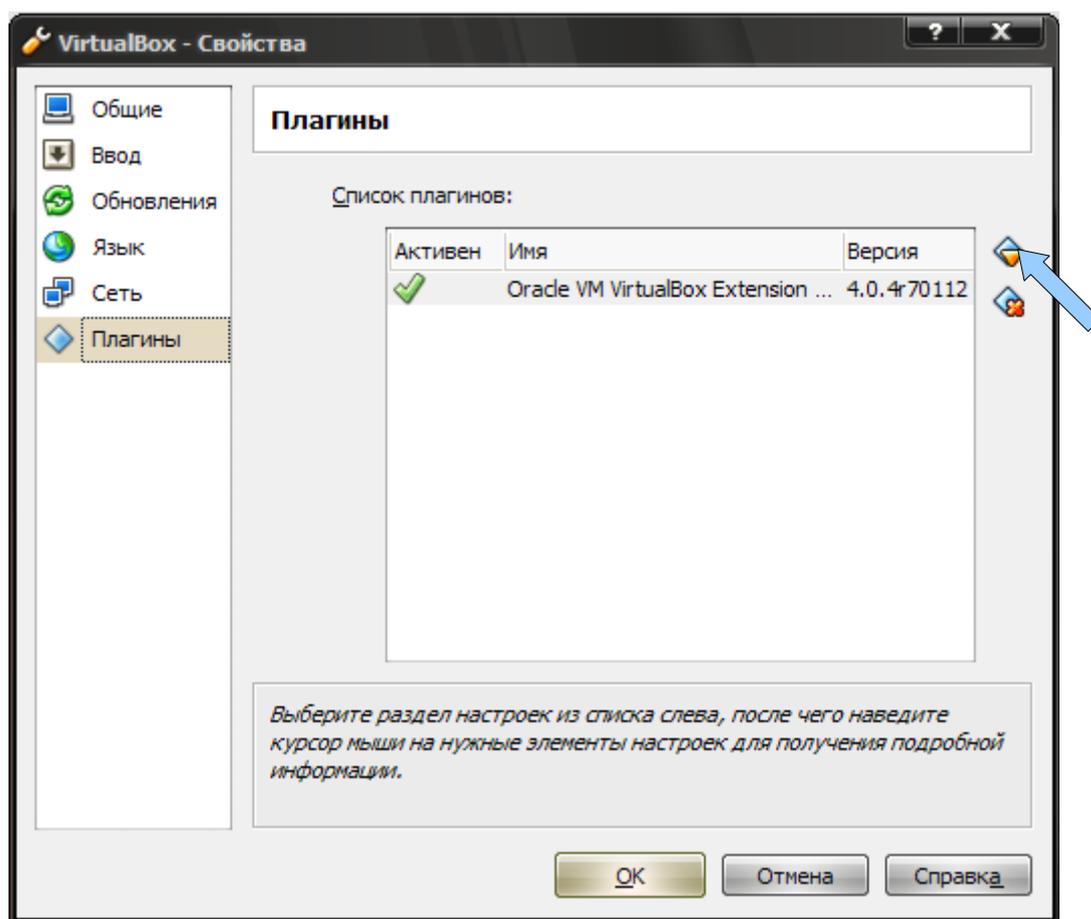


Рис. 1.8

Далее следует скопировать настроенный виртуальный жесткий диск `LabVisual.dvi` на реальный жесткий диск компьютера с DVD диска, поставляемого в комплекте с учебной установкой (рекомендуется виртуальные носители помещать в директорию `/home/pankov/.VirtualBox/HardDisks` либо в `/home/pankov/.VirtualBox/VDI` либо — непосредственно в домашнюю директор `/home/pankov/`), обычно для этого требуется около 5 — 7 Гб свободного места. Копирование также удобно осуществлять в файловом менеджере `Krusader`.

Затем необходимо создать новую виртуальную машину, нажав кнопку СОЗДАТЬ главного окна VirtualBox рис. 1.9. В качестве типа ОС рекомендуется

выбирать Windows XP, название установить — LabVisual. Как правило, для успешной работы среды следует выделить не менее 256 Мб оперативной памяти (ОЗУ).

На запрос о выборе жесткого диска, следует выбрать «СУЩЕСТВУЮЩИЙ» и указать скопированный вами на жесткий диск с прилагаемого DVD диска файл LabVisual.vdi. Если появится необходимость сменить виртуальный диск, можно в свойствах виртуальной машины выбрать пункт «Жесткие диски» и из списка доступных жестких дисков выбрать другой либо создать новый.

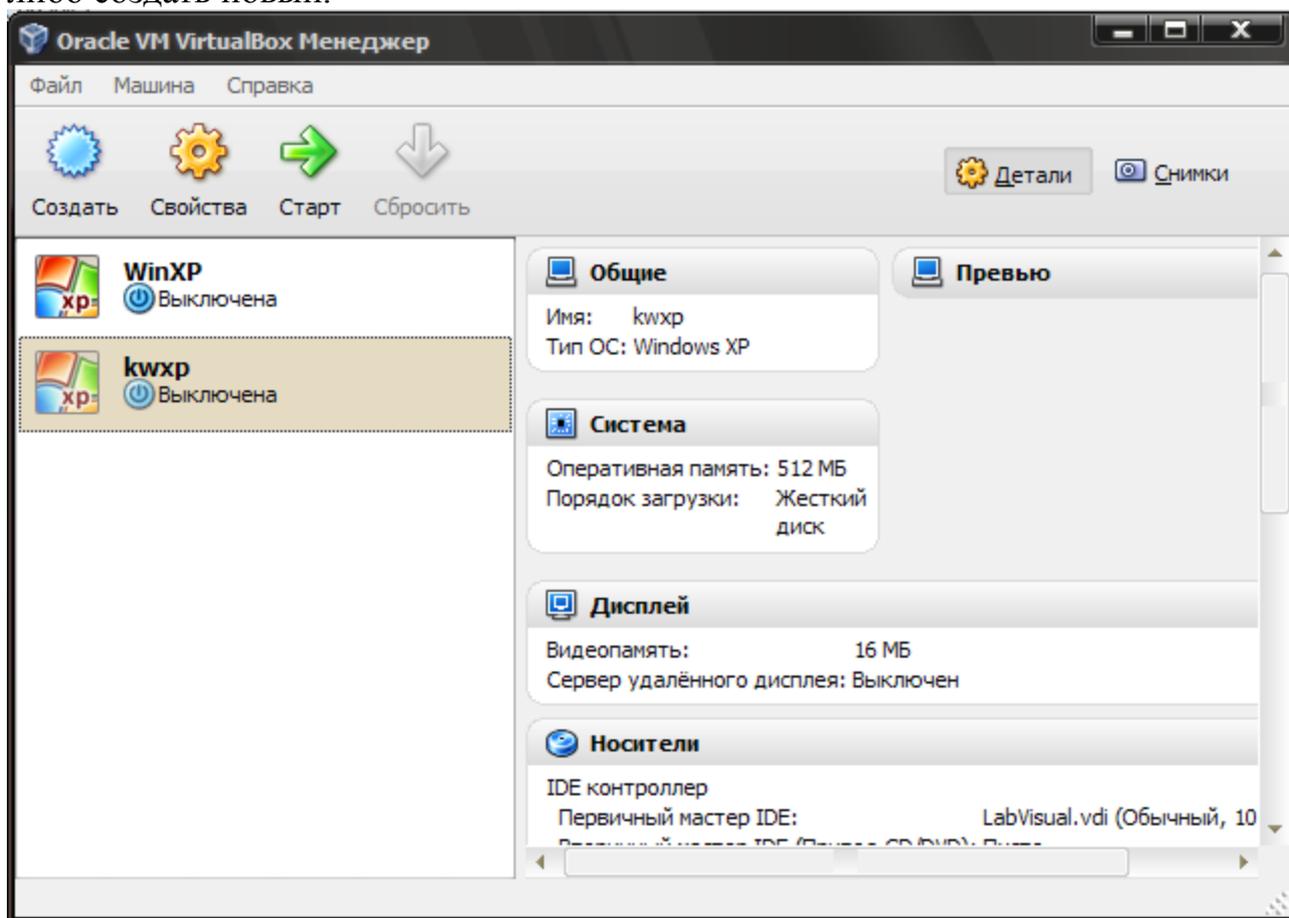


Рис. 1.9

После создания виртуальной среды, её надо предварительно сконфигурировать. Для этого в свойствах виртуальной машины LabVisual рис.1.10 — рис. 1.11:

СВОЙСТВА → **СИСТЕМА** → **МАТЕРИНСКАЯ ПЛАТА** установить тип чипсета ICH9 и поставить галочку «**ВКЛЮЧИТЬ IO APIC**».

ОСНОВНАЯ ПАМЯТЬ установить в диапазоне 256-512 Мб

На вкладке **ПРОЦЕССОР** поставить галочку «**ВКЛЮЧИТЬ PAE/NX**»

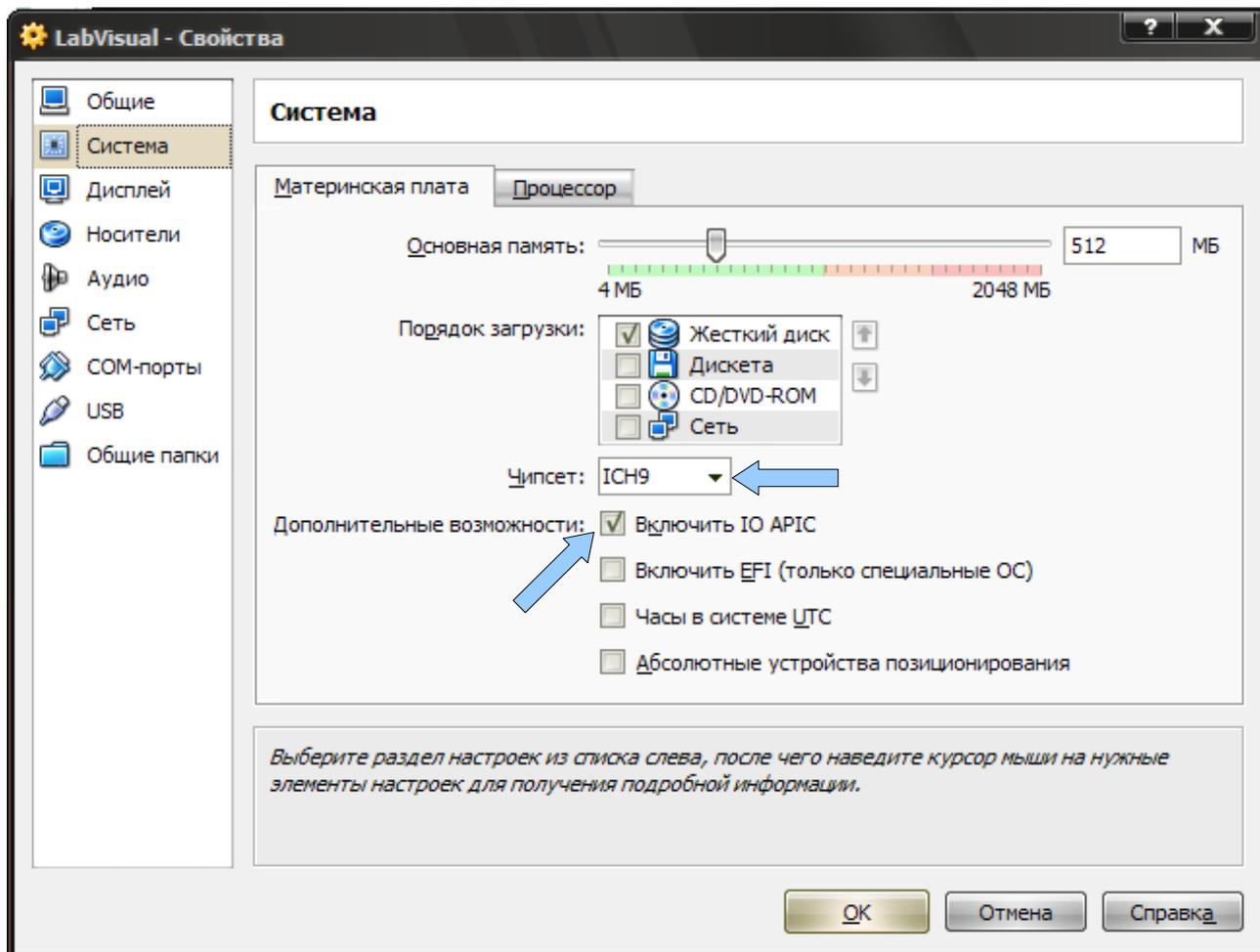


Рис. 1.10

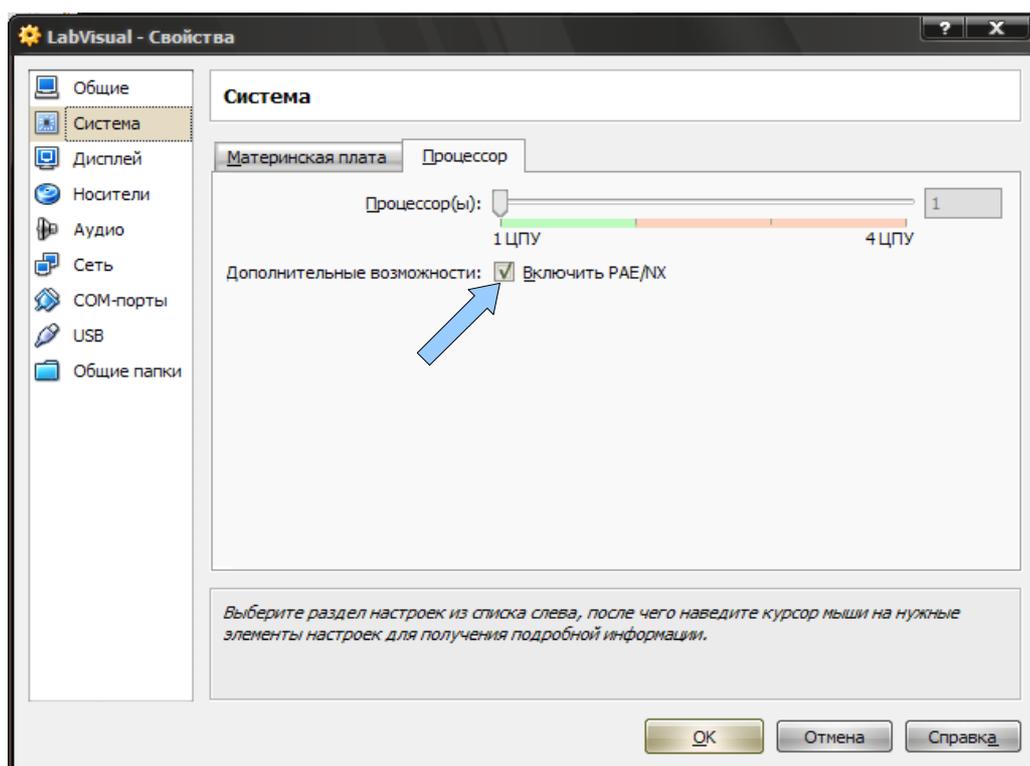


Рис. 1.11

СВОЙСТВА → СИСТЕМА → ДИСПЛЕЙ → ВИДЕОПАМЯТЬ установить значение видеопамати в пределах 64 — 128 Мб.

СВОЙСТВА → СИСТЕМА → АУДИО установить галочку «ВКЛЮЧИТЬ АУДИО» и выбрать в качестве аудиоконтроллера ICH AC97 либо Intel HD Audio рис. 1.12. В качестве аудиодрайвера установить OSS либо Pulse-Audio.

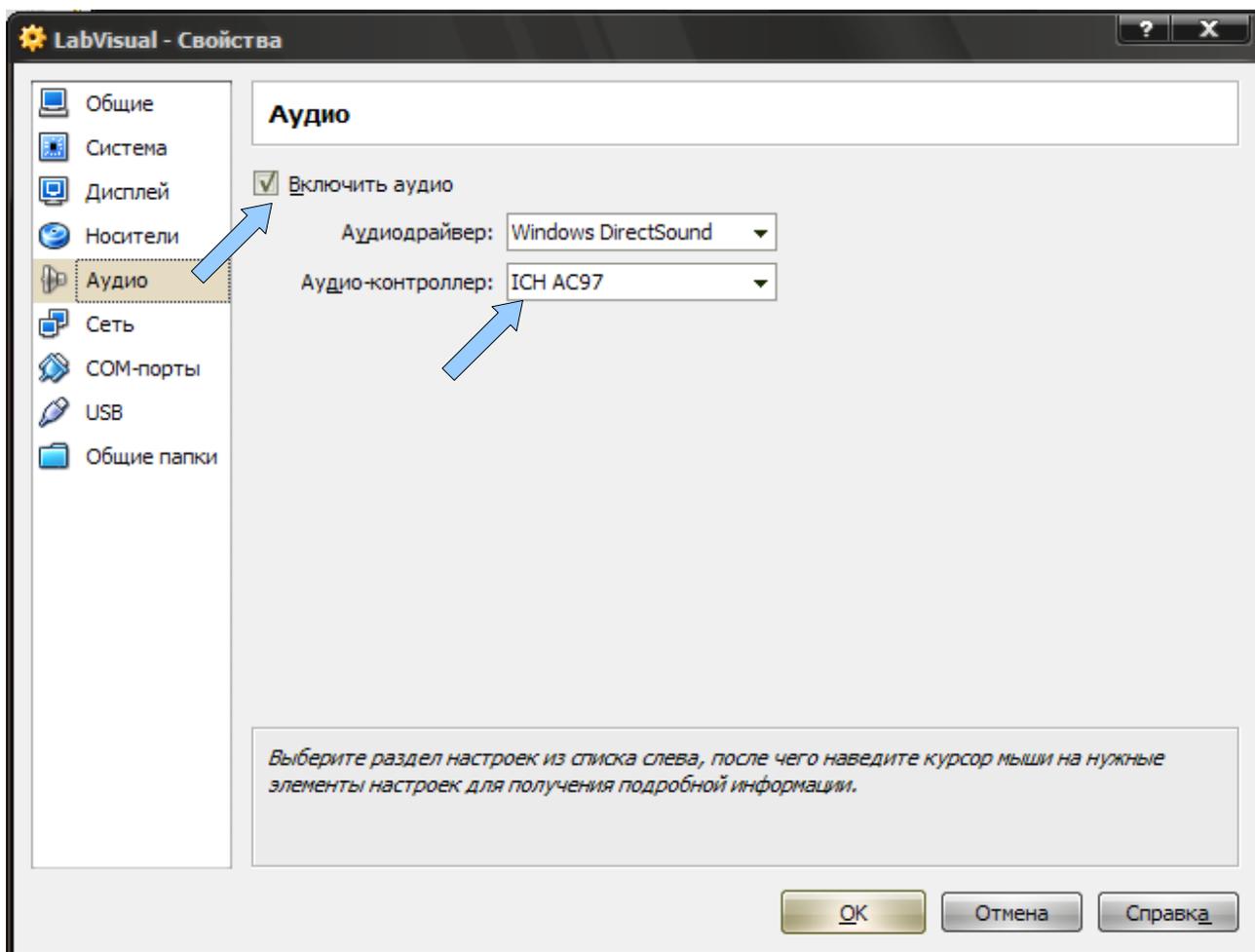


Рис. 1.12

Для работы USB следует правильным образом сконфигурировать USB подсистему. Для чего следует зайти в утилиту «Управление пользователями» системного меню Kubuntu рис. 1.13:

К-меню → Параметры системы → Управление пользователями щелкнуть мышью по вашей учетной записи и нажать кнопку «ИЗМЕНИТЬ». После чего следует добавить себя в группу vboxusers, установив галочку у соответствующей группы.

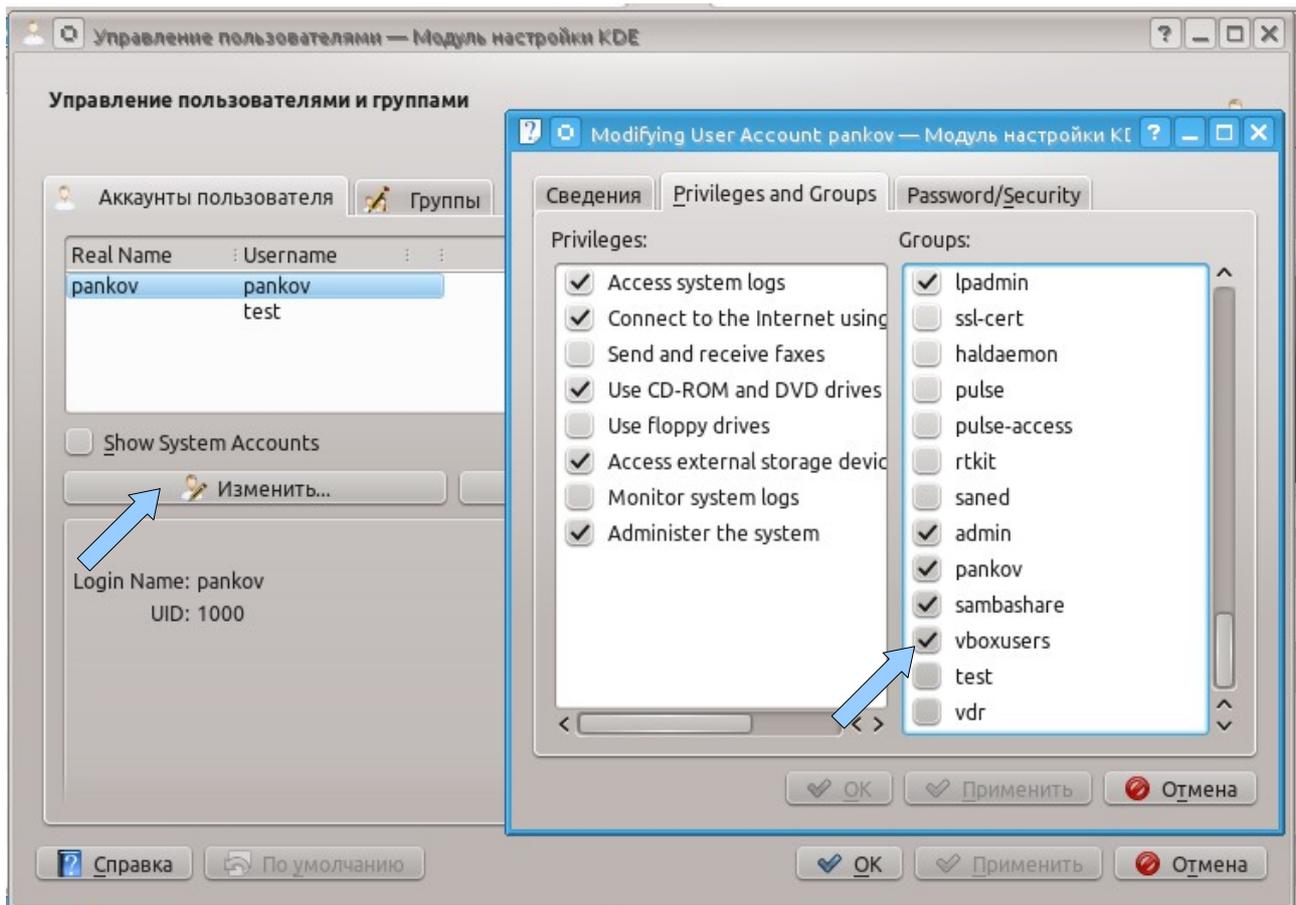


Рис. 1.13

Затем необходимо подключить учебную установку к USB порту ПК и перезагрузить компьютер:

К-меню → Выход → Перезагрузить.

После загрузки машины следует включить USB контроллер в свойствах виртуальной машины LabVisual:

МАШИНА-->СВОЙСТВА-->USB поставить галочку «Включить контроллер USB» и «Включить контроллер USB 2.0».

Затем необходимо добавить в список доступных устройств учебную установку, для чего нужно нажать кнопку «Добавить из устройства» (см. рис. 1.14) и выбрать «TULANAUCHPRIBOR ... », после этого в поле «Фильтры устройств USB» появится необходимое устройство.

Для работы программы-осциллографа на АЦП звуковой карты нужно предварительно настроить линейный вход Line In в операционной системе Kubuntu, для чего следует воспользоваться значком громкости (утилита kmix) на панели уведомлений рис. 1.15. Во всплывающем окне нажать кнопку «МИКСЕР», перейти на вкладку «Устройства захвата звука» («Input») и установить ручки в средние положения. Конкретный вид окна управления Kmix может отличаться от приведенных в зависимости от типа используемой звуковой карты и чипсета ПК.

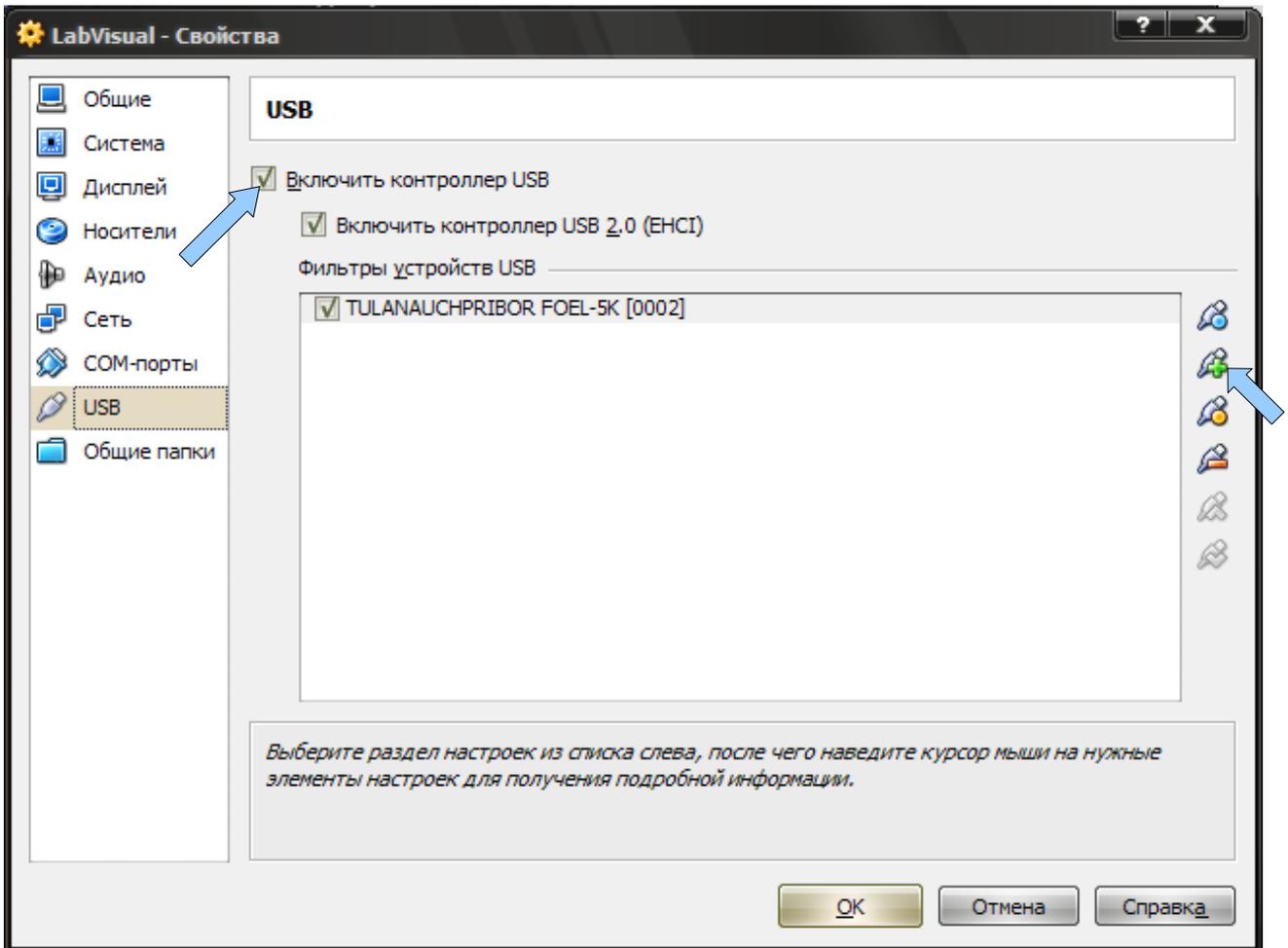


Рис. 1.14

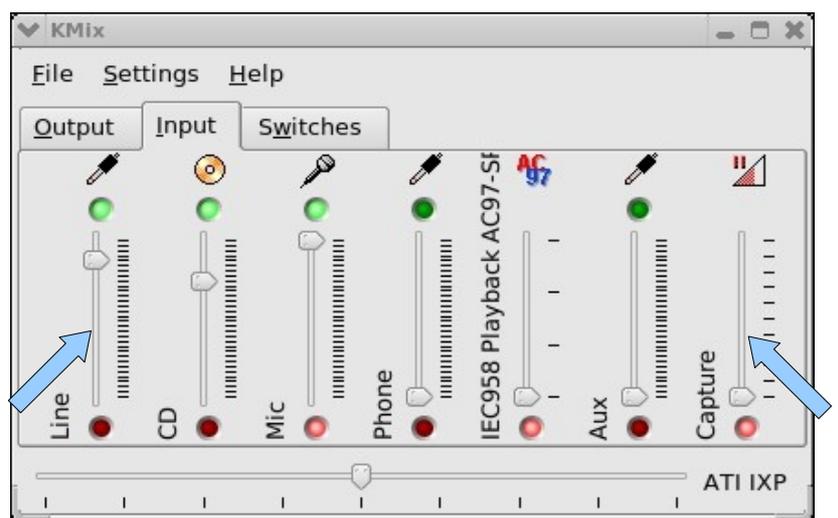


Рис. 1.15

Для обмена данными со средой LabVisual в ОС Linux необходимо настроить так называемую «ОБЩУЮ ПАПКУ». Удобнее всего сделать это следующим образом. Запустите файловый менеджер Krusader:

К-меню -> Приложения → Службное -> Krusader

Перейдите в домашнюю директорию `/home/pankov` и создайте там папку с произвольным именем, например VB (клавиша F7 в файловом менеджере).

Запустите VirtualBox, в котором из списка доступных сред для эмуляции следует выбрать настроенную вами LabVisual и нажать кнопку «СТАРТ». Дождитесь загрузки среды (~1 минута). Затем в главном меню среды выберите:

УСТРОЙСТВА → ОБЩИЕ ПАПКИ, щелкните кнопку «Добавить общую папку» и укажите путь к вашей папке (в нашем примере `/home/pankov/VB`). Установите галочки «Создать постоянную папку» и «Авто-подключение» и нажмите ОК рис. 1.15-1. Теперь вашей папке `/home/pankov/VB` в среде LabVisual присвоен соответствующий диск (как правило Z:\). Изменяя в среде содержание диска Z:\ (создавая и удаляя файлы, папки), вы одновременно таким же образом изменяете содержание вашей «ОБЩЕЙ ПАПКИ» в системе Kubuntu `/home/pankov/VB`

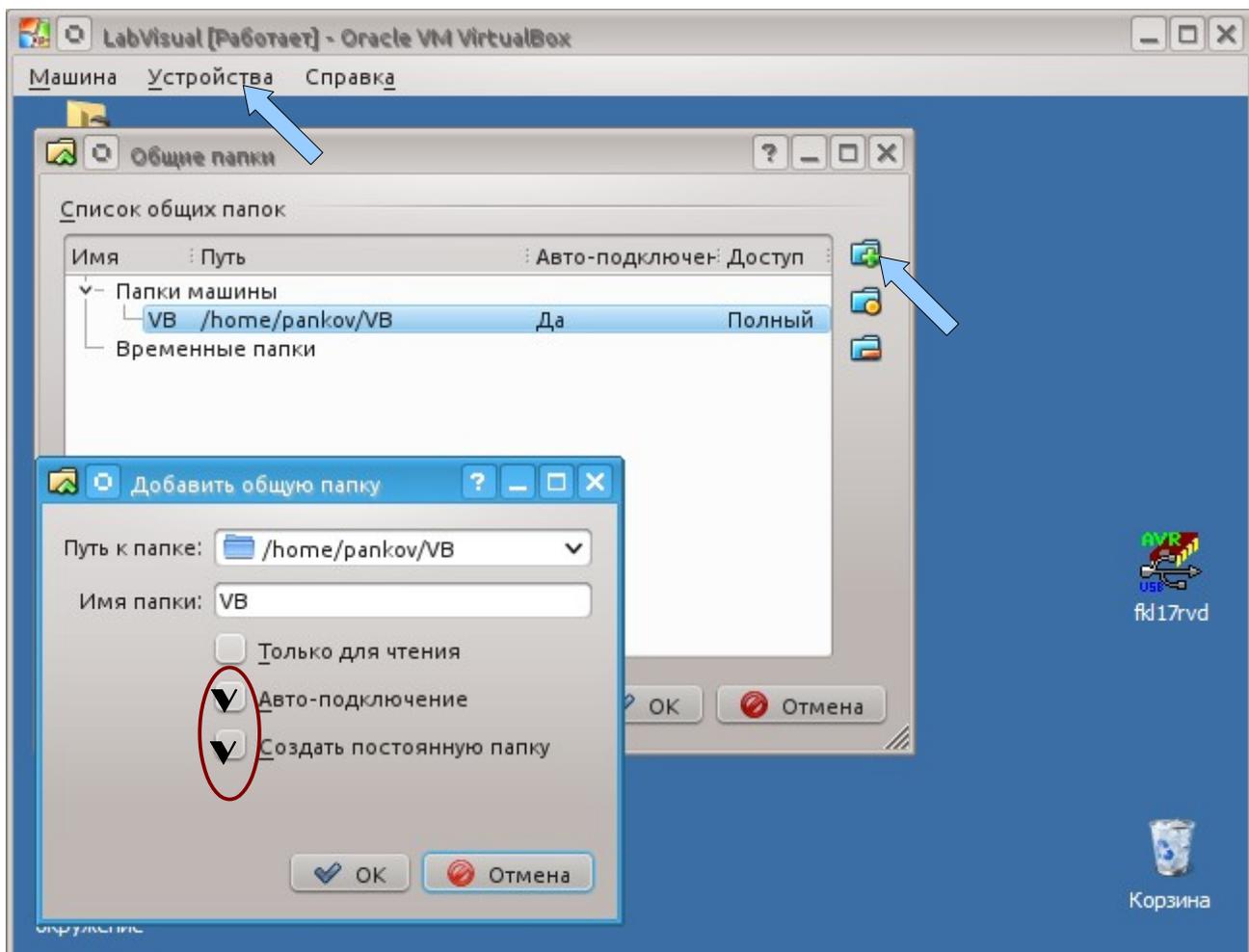


Рис. 1.15-1

1.7. Работа среды LabVisual в ОС Kubuntu Linux.

Для корректной работы рекомендуется подключать учебные установки к USB порту ПК перед включением и загрузкой ПЭВМ, или, в крайнем случае, перед стартом программы VirtualBox (LabVisual) в ОС Kubuntu. Это желательно для правильного старта подсистемы USB.

В комплекте с лабораторной установкой поставляется ПЭВМ под управлением ОС Linux с предустановленным дистрибутивом Ubuntu и установленным и настроенным программным обеспечением. Для входа в компьютер под обычным пользователем следует использовать:

ЛОГИН (login) pankov

ПАРОЛЬ (password) pankov

Для выполнения административных задач (установка, настройка ПО, добавление пользователей) в ОС Linux по умолчанию используется суперпользователь root с паролем (в данной настроенной ПЭВМ пароль pankov):

ЛОГИН (login) root

ПАРОЛЬ (password) pankov

В графическую оболочку по умолчанию вход root запрещен, везде где это необходимо рекомендуется использовать команду sudo имя_команды.

После входа в графическую среду окружения Linux следует запустить программу LabVisual, щелкнув мышью по символьной ссылке, расположенной на рабочем столе.

Среду можно запустить и непосредственно через эмулятор VirtualBox, набрав в консоли команду:

```
VirtualBox
```

и нажав ENTER (**вызов консоли уакуаке клавиша F12**) рис. 1.16. После запуска среды таким способом сначала появляется главное окно эмулятора VirtualBox (рис. 1.17) в котором из списка доступных сред для эмуляции следует выбрать LabVisual и нажать кнопку «СТАРТ».

После загрузки программной среды (~ 1 мин.) автоматически запуститься программа оболочка LabVisual для работы с экспериментальной установкой. Если программа не запустилась автоматически, на открывшемся виртуальном рабочем столе следует дважды щелкнуть левой кнопкой мыши на ярлык LabVisual. При этом должно открыться главное окно программы-оболочки LabVisual для работы с экспериментальной установкой.

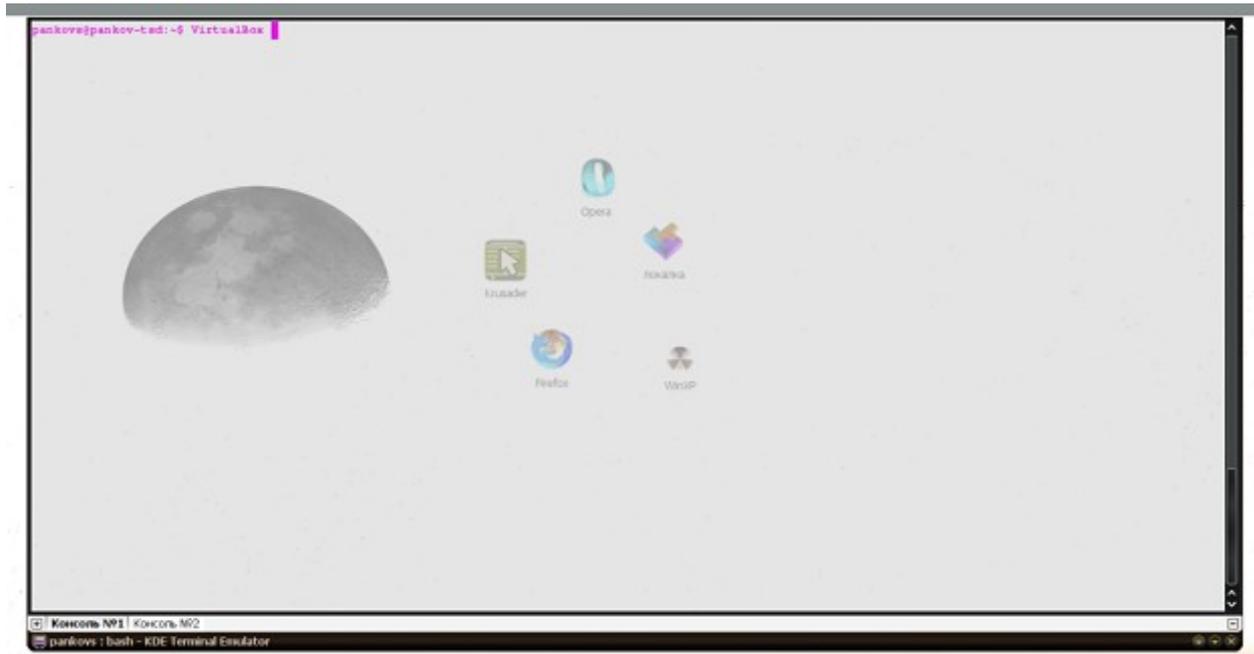


Рис. 1.16

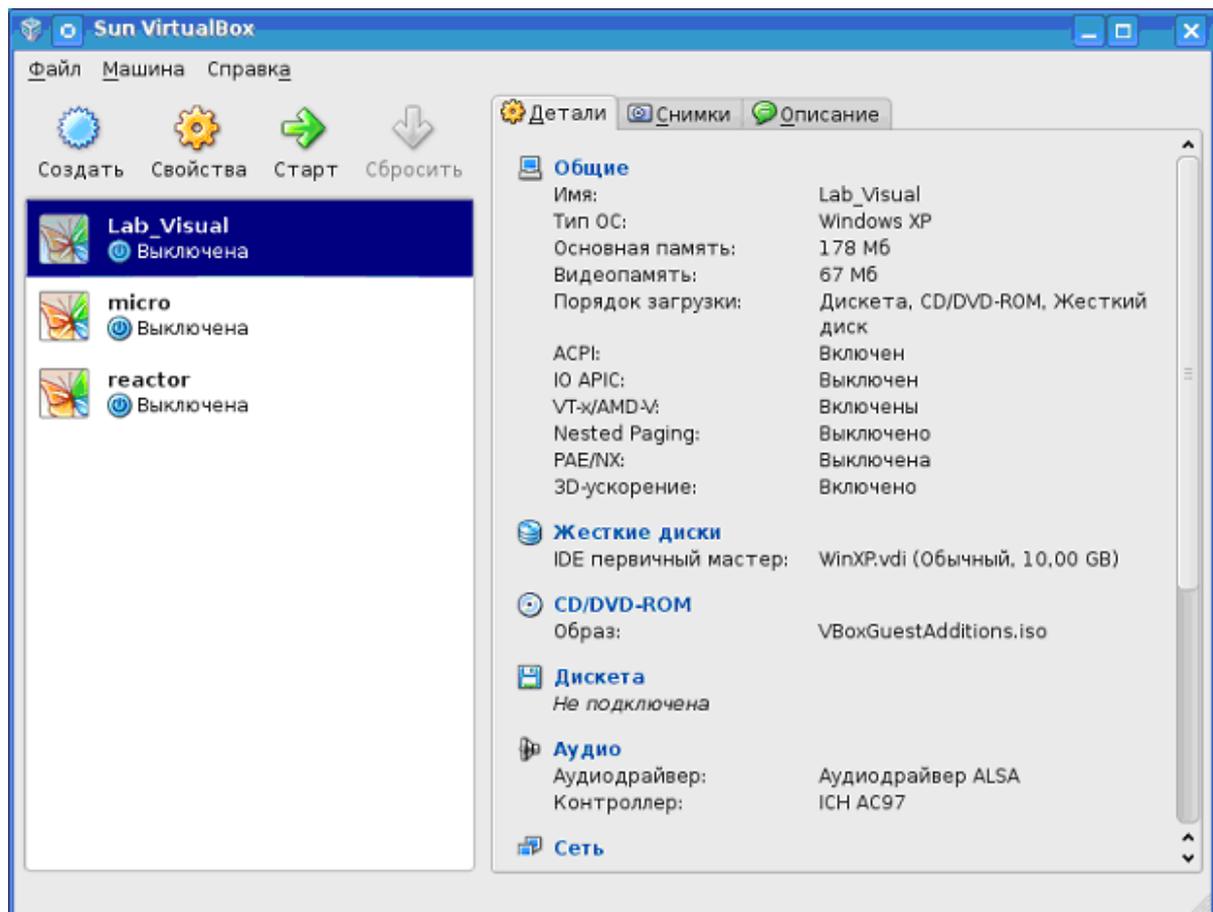


Рис. 1.17

Программа LabVisual имеет интуитивно понятный, дружелюбный пользователю интерфейс. Главное окно программы, а также входящие в среду компоненты отличаются в зависимости от используемой учебной установки. На рис. 1.18 — 1.19 приведены некоторые скриншоты компонентов среды для работы с различными лабораторными установками производства НПО «ТулаНаучПрибор».

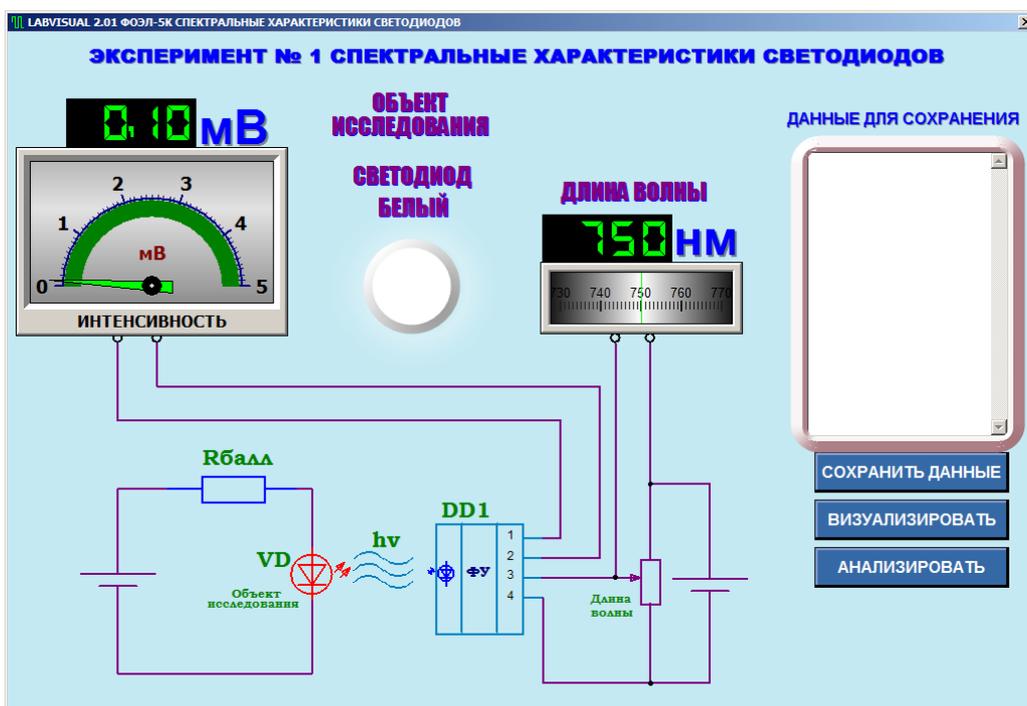
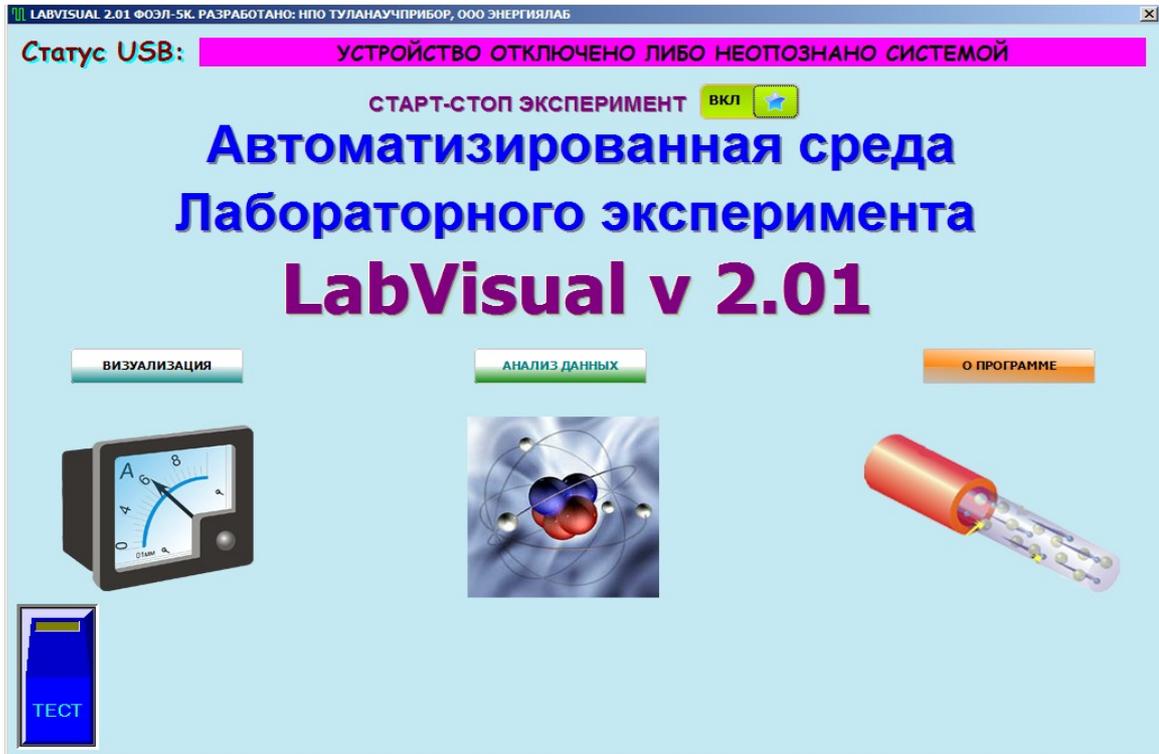


Рис. 1.18

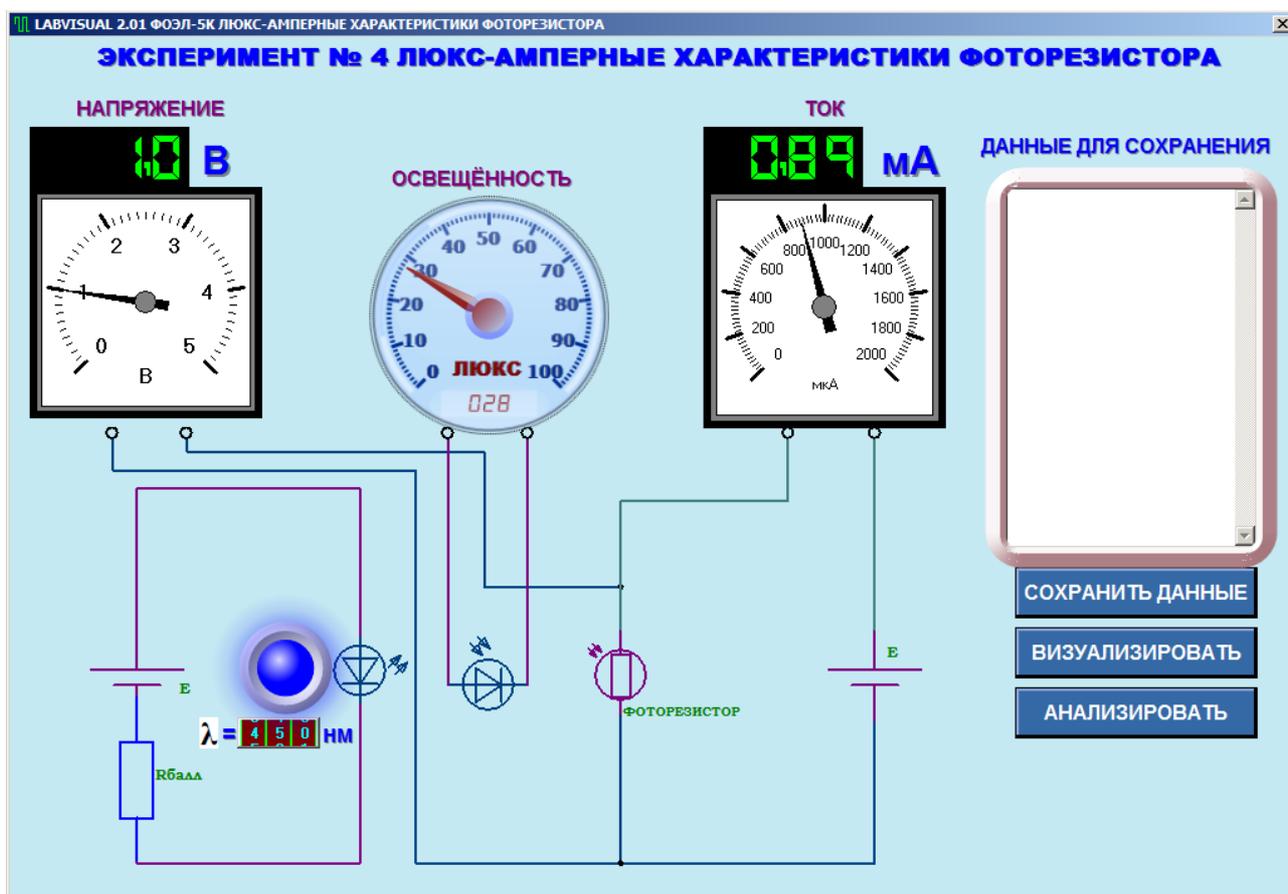


Рис. 1.19

Программа LabVisual предусматривает возможность сохранения избранного набора данных файл для последующей визуализации и, при необходимости, обработки. Данные, которые необходимо сохранить, вводятся в специальное поле «ДАнные ДЛЯ СОХРАНЕНИЯ».

При работе в ОС Linux в эмуляторе программного кода VirtualBox данные рекомендуется сохранять в файл с произвольным именем на виртуальный диск Z:\, последующий доступ к этому виртуальному диску из операционной системы осуществляется как к папке /home/имя_пользователя/VB, где имя_пользователя по умолчанию pankov. Для обзора папок рекомендуется пользоваться программой Krusader.

Для визуализации и предварительного анализа сохраненных данных можно использовать компонент LabVisual Data Analyzer, вызываемый нажатием кнопки «ВИЗУАЛИЗИРОВАТЬ» из главного окна программы-оболочки и из любой подпрограммы рис. 1.20.

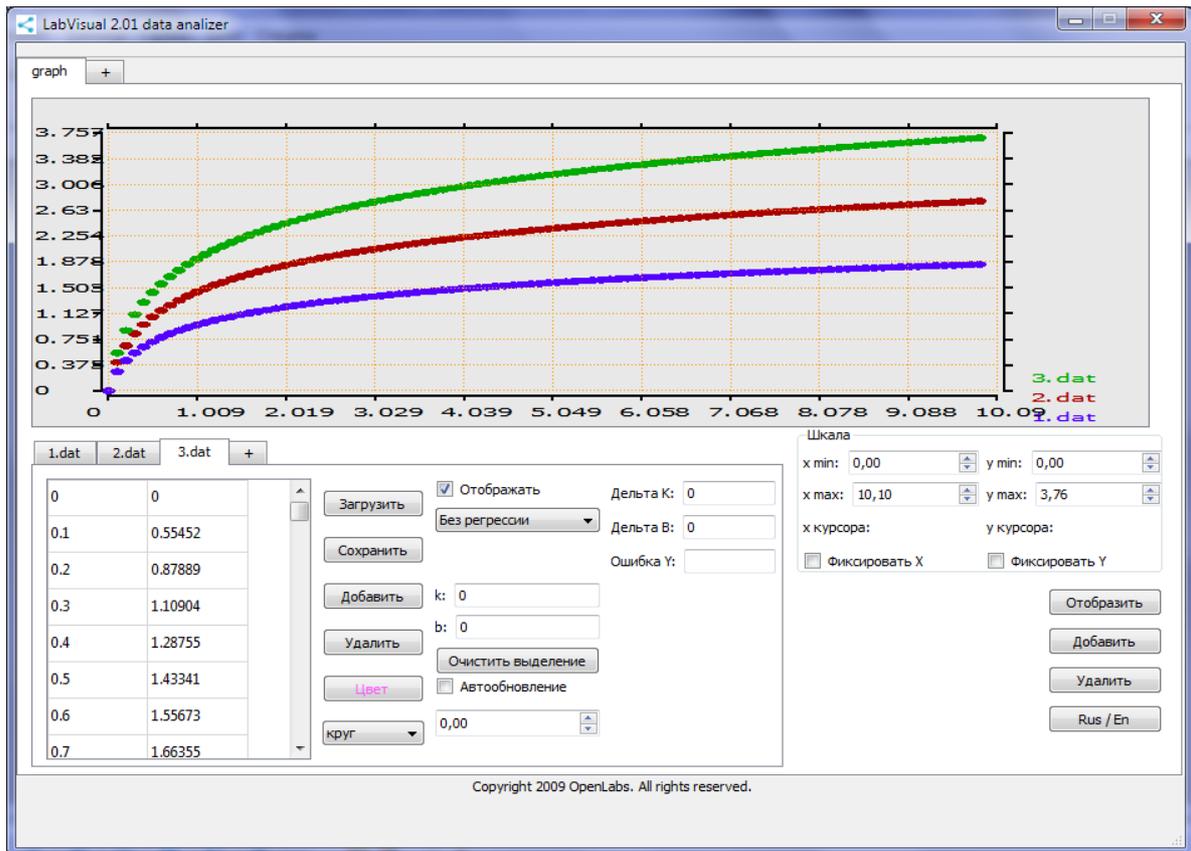


Рис. 1.20

Для загрузки какого-либо файла данных в компоненте LabVisual Data Analyzer служит кнопка «ЗАГРУЗИТЬ», для отображения загруженных данных, а также после каких-либо внесенных изменений в график (цвет, символы для отображения и т. д.) следует нажимать кнопку «ОТОБРАЗИТЬ» для перерисовки. Открыв несколько вкладок данных нажатием на вкладку «+» и, загрузив в каждую вкладку данные из сохраненных файлов данных, можно отобразить на рабочем поле семейство характеристик, так как это представлено на рис. 1.20. Вкладка «+» для графиков служит для создания нескольких рабочих областей графиков, в каждую из которых также можно загрузить данные.

Для анализа экспериментальных данных и построения графиков также можно воспользоваться специальным компонентом LabVisual, вызываемым нажатием кнопки «АНАЛИЗИРОВАТЬ» рис. 21. Внешний вид компонента для анализа данных также может варьироваться в зависимости от используемого прибора. В целом, компонент для анализа данных обладает следующими функциями: импорт-экспорт данных, построение графиков кривых данных, обработка экспериментальных данных расширенным набором функциональных зависимостей (линейная, экспоненциальная, Гауссиан и др.).

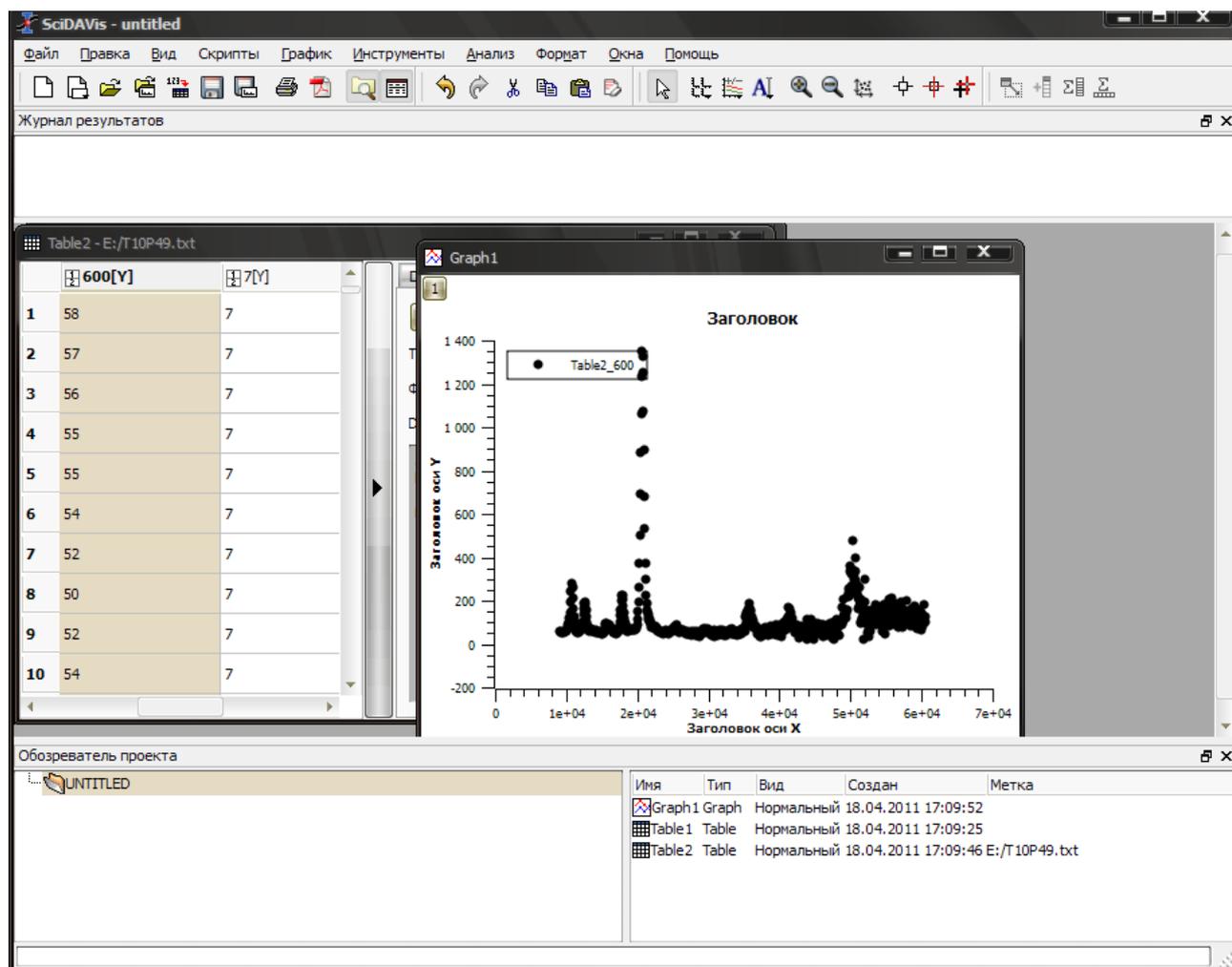


Рис. 1.21

ВНИМАНИЕ! Для работы компонента MagicPlot для анализа данных необходимо установить последнюю версию исполняющей среды Java 6 (Runtime библиотеки виртуальной машины Java).

1.8. Установка и настройка среды LabVisual в ОС Windows.

Для работы учебной установки в среде Windows необходимо установить соответствующий драйвер USB устройства, поставляемый в комплекте на CD диске. Установка драйвера устройства производится стандартным образом средствами операционной системы. Установка драйверов возможно в ручном и в автоматическом режиме.

Для установки драйверов в ручном режиме необходимо подключить учебную установку к USB порту компьютера, при этом должен запускаться мастер установки нового оборудования Windows рис. 1.22. На запрос о подключении к web узлу следует ответить отрицательно и нажать кнопку далее. В следующем диалоговом окне выбрать пункт «Установка из указанного места» и нажать кнопку далее, затем установить в качестве места поиска папку, содержащую драйвер (папка на CD диске, либо папка на жестком диске со скопированной программой). На предупреждение об отсутствии цифровой подписи ответить «ВСЁ РАВНО ПРОДОЛЖИТЬ УСТАНОВКУ». По окончании установки драйвера нажать кнопку «ГОТОВО». После инсталляции драйвера учебная установка готова к работе.

Для установки драйвера в автоматическом режиме необходимо предварительно отключить от USB-порта ПЭВМ учебную установку и затем запустить программу «LabVisual DriverInstaller». На предупреждение об отсутствии цифровой подписи ответить «ВСЁ РАВНО ПРОДОЛЖИТЬ УСТАНОВКУ». После инсталляции драйвера, следует подключить лабораторный прибор к USB – порту ПК. На запрос о подключении к web узлу (если таковой отобразится, как на рис. 1.22), следует ответить отрицательно и нажать кнопку «ДАЛЕЕ». В следующем диалоговом окне (если таковое отобразилось) выбрать пункт «Автоматическая установка», а на следующее предупреждение об отсутствии цифровой подписи на используемый драйвер ответить «ВСЁ РАВНО ПРОДОЛЖИТЬ УСТАНОВКУ». После выполненных действий учебная установка полностью готова к работе.

После успешной инсталляции драйвера для работы с учебными установками, следует установить в систему программное обеспечение среды LabVisual для Windows.

Установка ПО среды LabVisual может производиться как при помощи стандартного бинарного *.exe исполняемого файла-установщика, так и из исходных кодов с предварительной компиляцией среды специальной утилитой-компилятором.

Для большинства обычных пользователей установку ПО среды LabVisual рекомендуется производить при помощи стандартного исполняемого *.exe файла-установщика. Для чего следует запустить данный файл (обычно инсталляционные пакеты имеют название Setup.exe, Install.exe и т. п.) и следовать инструкциям мастера установки ПО. В процессе установки среды имеется возможность выбрать папку для установки и некоторые дополнительные параметры рис. 1.23 (создание ярлыков, создание группы программ в меню ПУСК-ПРОГРАММЫ и т. п.). По окончании установки программа готова к работе.

ВНИМАНИЕ! Для работы компонента MagicPlot для анализа данных необходимо установить последнюю версию исполняющей среды Java 6 (Runtime библиотеки виртуальной машины Java).

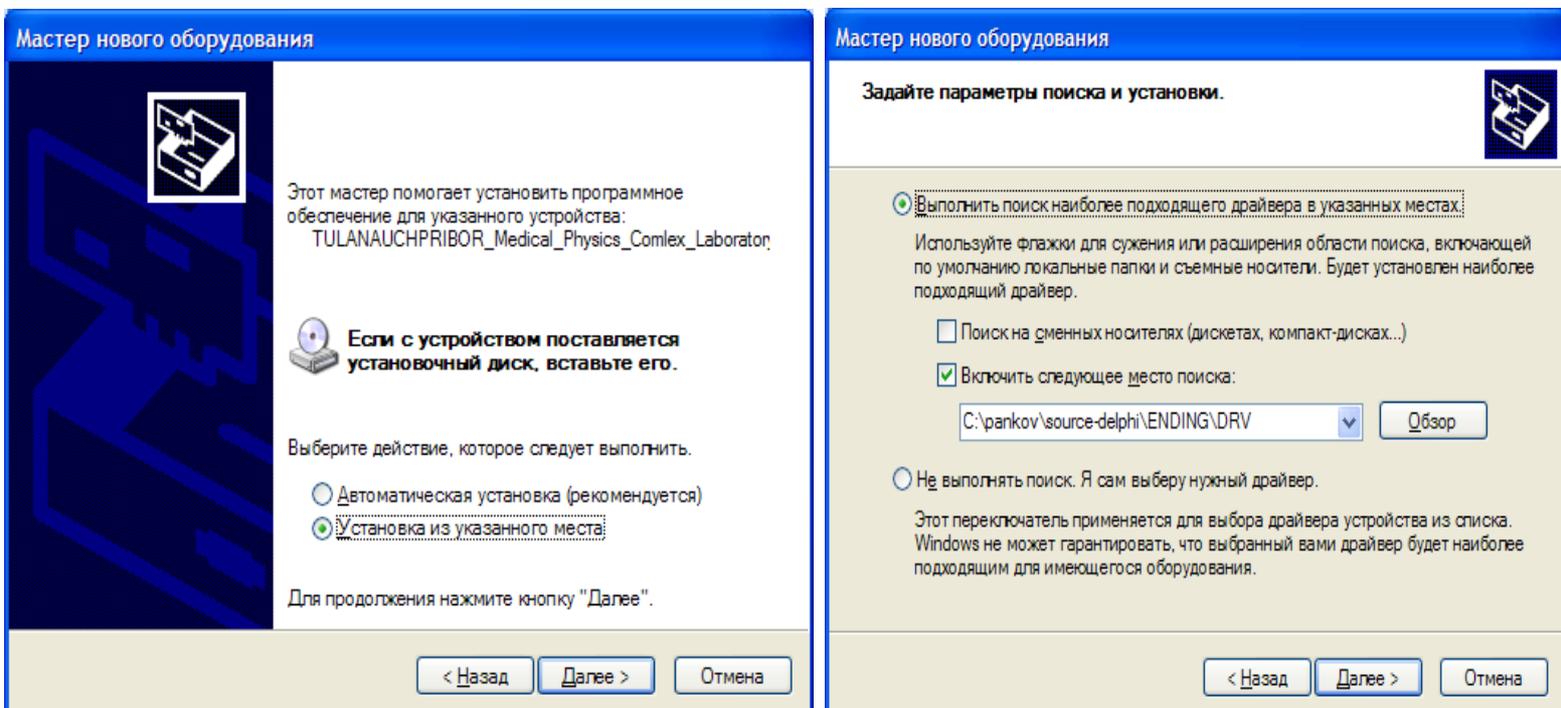
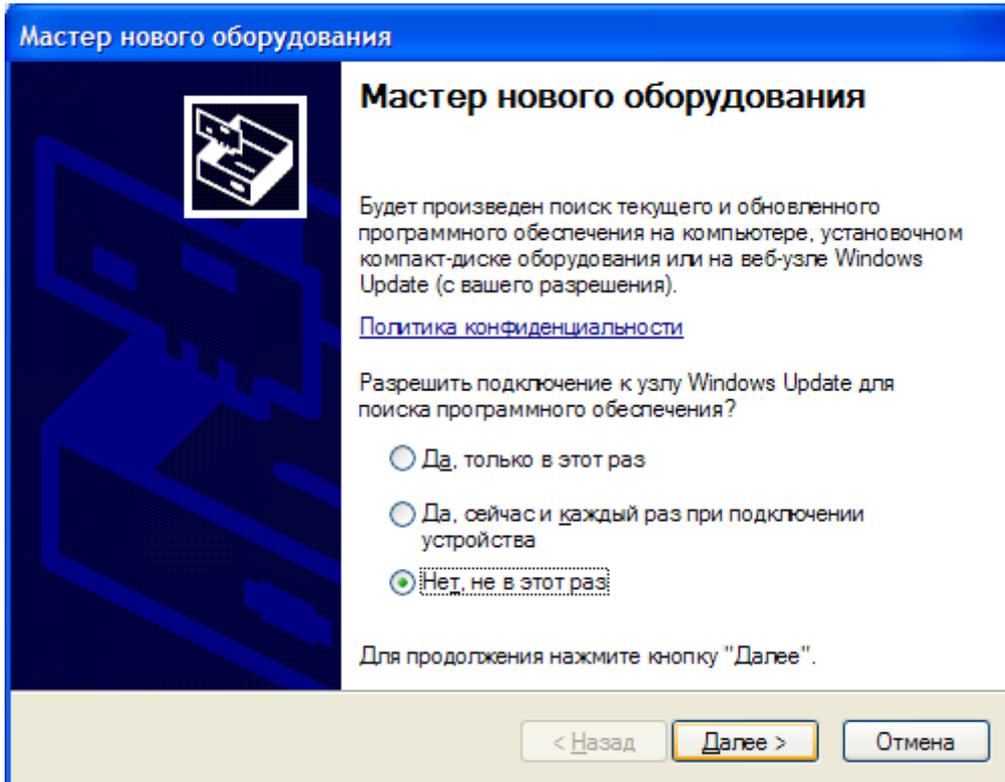


Рис. 1.22

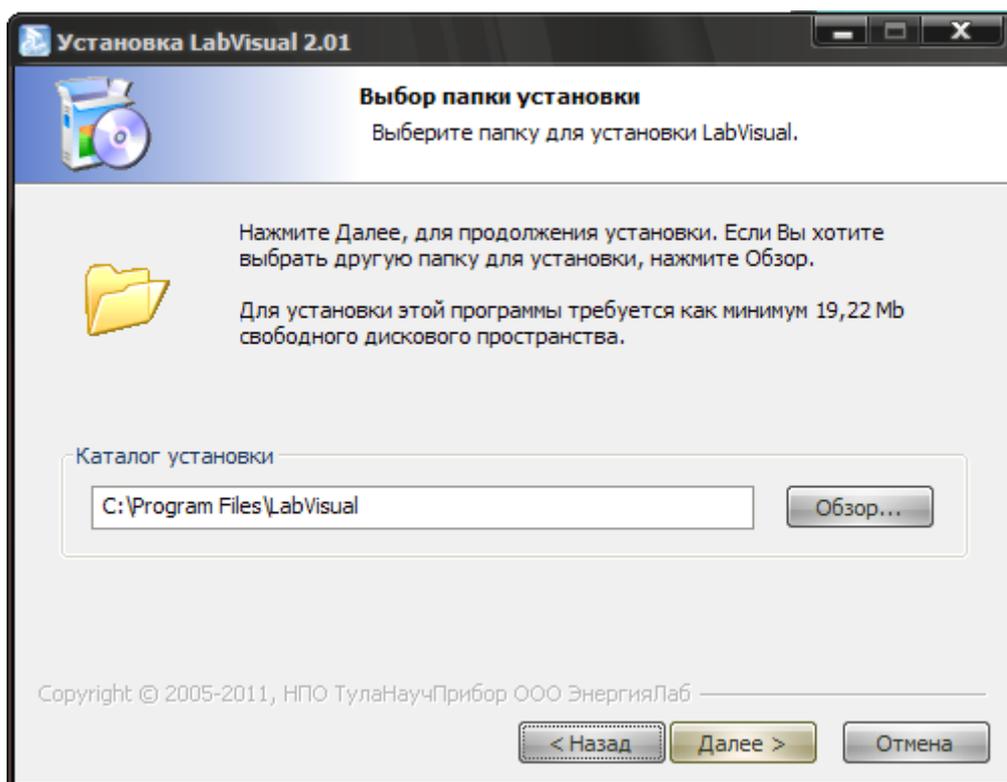


Рис. 1.23

Для большей совместимости со всеми типами архитектуры ПК, а также при необходимости изменения исходного кода среды, необходимо воспользоваться установкой среды LabVisual из исходных кодов с помощью компиляции утилитой-компилятором LabVisual_COMPILE_ENVIROMENT. Компилятор представляет собой стандартное приложение Windows рис. 1.24. Перед началом компиляции следует выбрать одну из опций компиляции:

полная компиляция — перекомпилирование всех компонент среды, в том числе сторонних используемых компонент;

стандартная компиляция — перекомпилирование только основных компонент среды;

компиляция только редактируемых модулей — перекомпиляция только тех модулей среды, исходный код которых был отредактирован пользователем.

Нажатие кнопки «КОМПИЛИРОВАТЬ» запускает компиляцию с соответствующими выбранными опциями. После этого программа переходит в режим командной строки с отображением процесса сборки приложения. По окончании процедуры сборки среды следует ввести путь для установки (копирования скомпилированных исполняемых файлов) и нажать кнопку «УСТАНОВИТЬ». При этом новые версии исполняемых файлов будут скопированы в указанную вами папку.

Обычно полные исходные коды среды с исходными кодами всех используемых компонент доступны в папке \COMPILE_ENVIROMENT\FULL_SOURCE\ на прилагаемом DVD-диске.

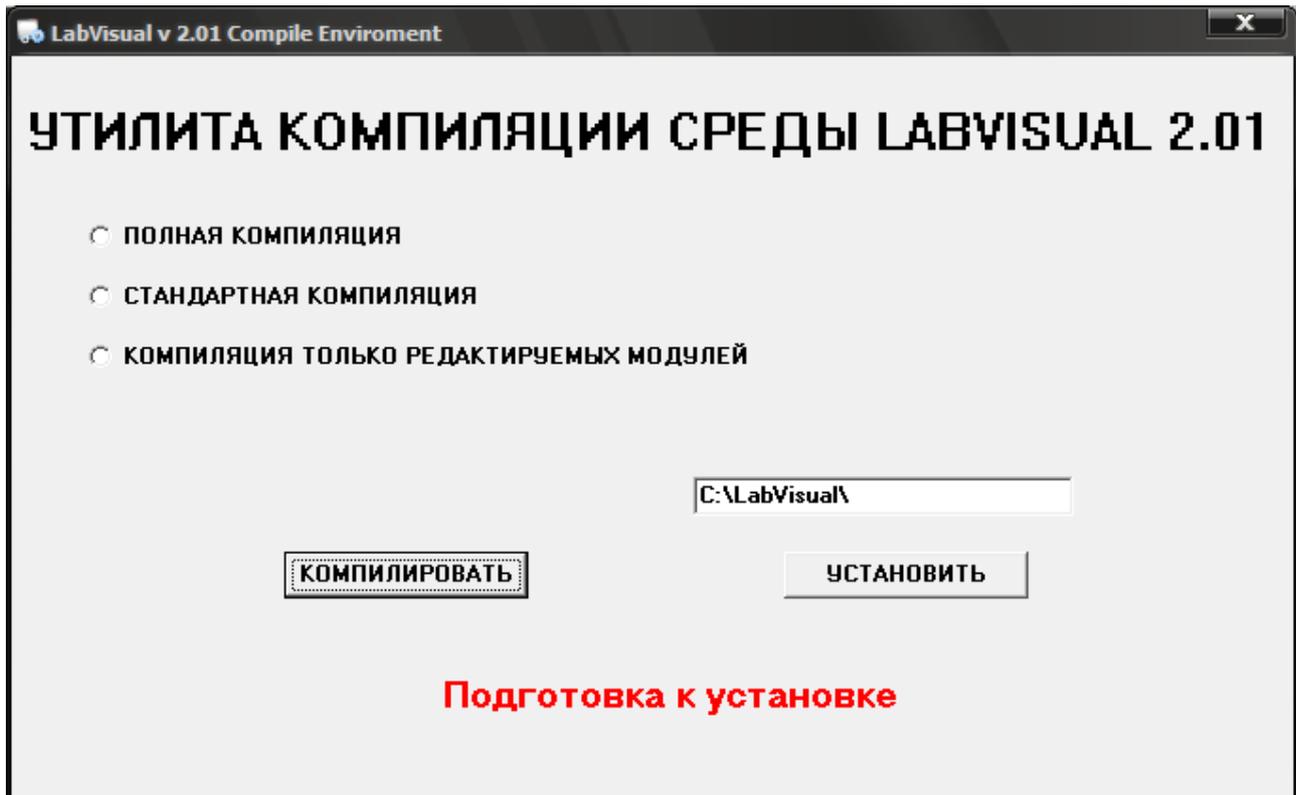


Рис. 1.24

1.9. Работа среды LabVisual в ОС Windows.

Работа программы-оболочки LabVisual в среде Windows ничем не отличается от описанной выше работы в среде ОС Linux, описанной в п. 1.7. Необходимость в использовании эмулятора программного кода VirtualBox в данном случае отпадает. При этом достаточно установить пакет программ и драйверов для работы с учебной установкой с поставляемого в комплекте с прибором CD диска в любую папку на жесткий диск, в которую доступна запись согласно п. 1.8 настоящего руководства и создать ярлык на рабочем столе для быстрого доступа к программе LabVisual.

Для корректной работы компоненты осциллографа на АЦП звуковой карты ПК, перед началом работы следует настроить линейный вход компьютера. При этом следует включить линейный вход звуковой карты в системе и настроить усиление Line In средствами операционной системы. Процедура настройки зависит от звуковой карты, в общем случае следует два раза щелкнуть правой кнопкой мыши на значок «ЗВУК», обычно располагающийся в системной области уведомлений и выбрать ПАРАМЕТРЫ → СВОЙСТВА → Устройства Записи. Для некоторых моделей звуковых карт регулировка линейного входа возможна с помощью программ, поставляемых в комплекте с драйверами для звуковой карты (например встроенные звуковые карты Realtek) рис. 1.25 — 1.26.

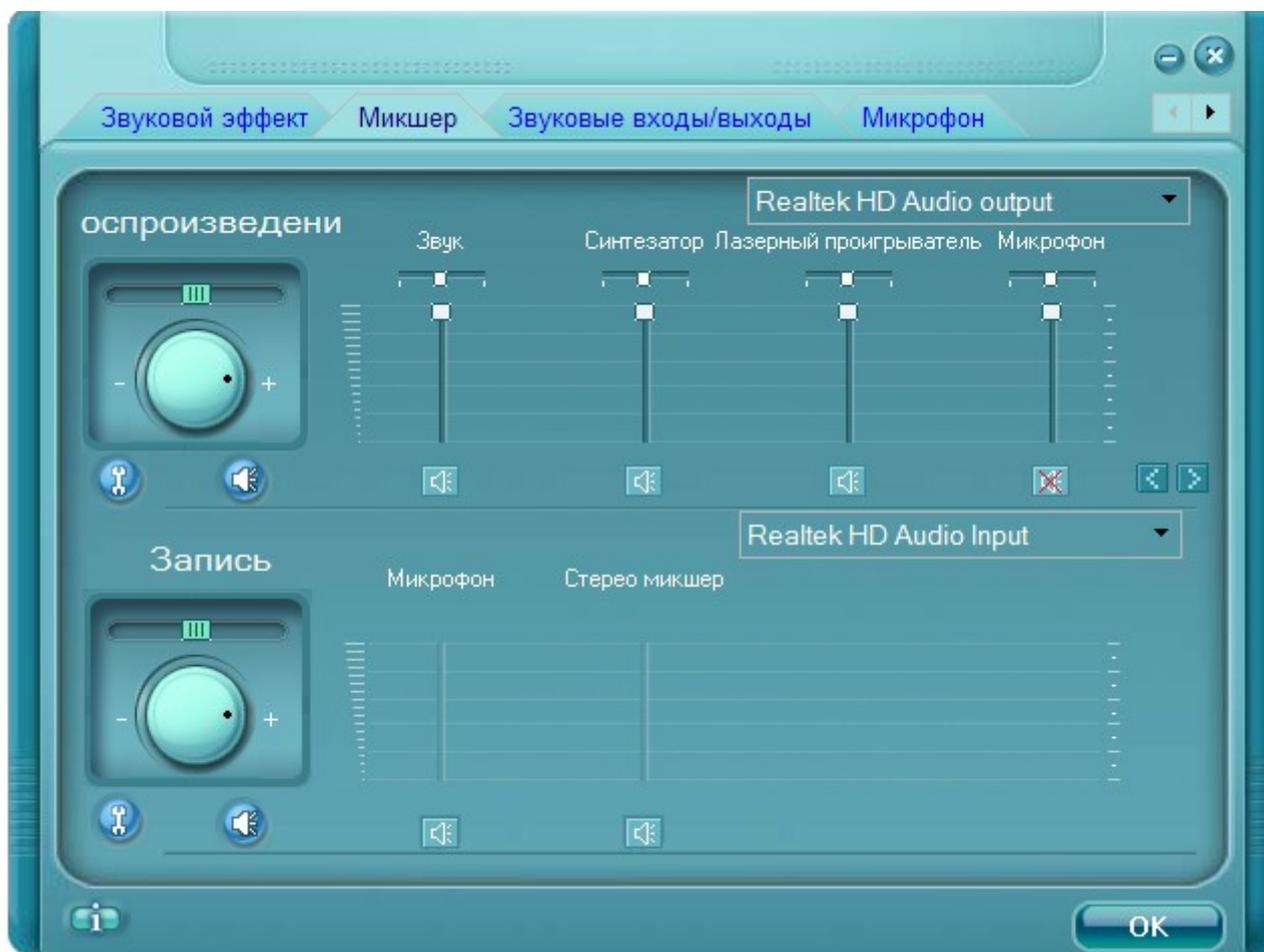


Рис. 1.25

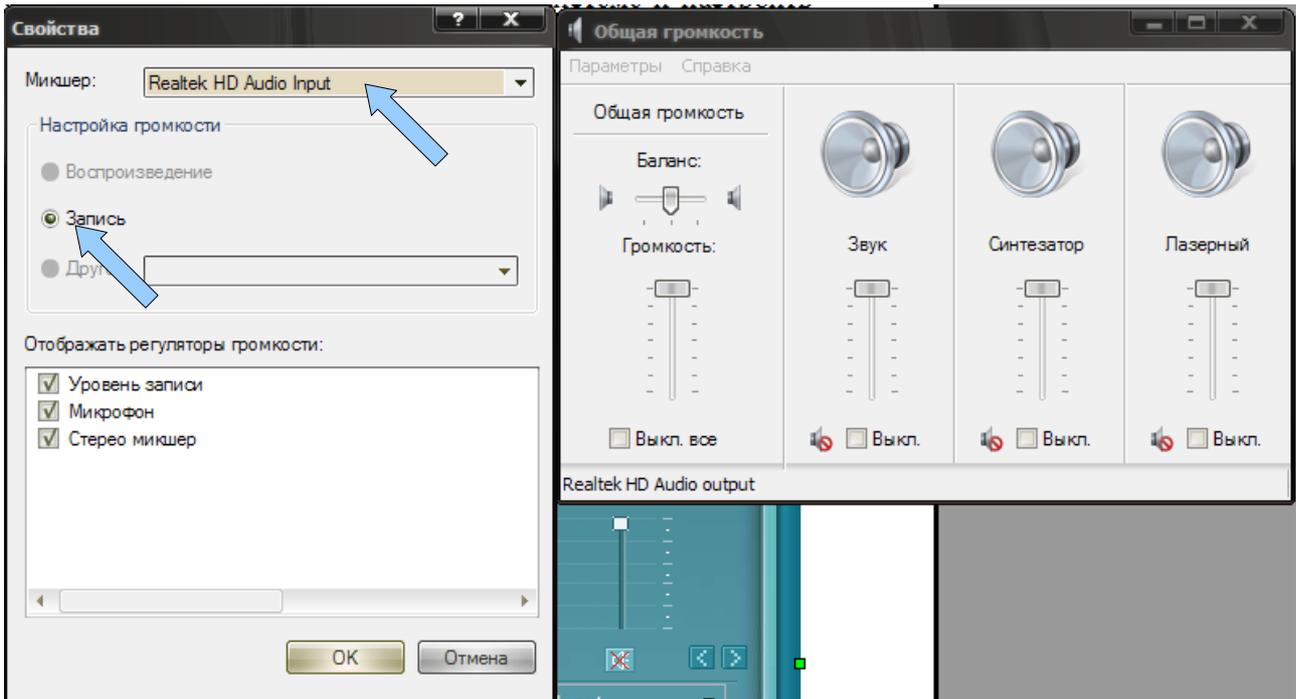


Рис. 1.26

2. АППАРАТНАЯ ЧАСТЬ. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ УНИВЕРСАЛЬНОЙ ПОСЛЕДОВАТЕЛЬНОЙ ШИНЫ (USB)

2.1 Введение.

Универсальная последовательная шина (USB) стала чрезвычайно популярной за счет предоставления ряда удобств конечным пользователям, например, функция "Plug and Play", которая позволяет идентифицировать подключенное устройство без необходимости рестарта компьютера. Однако для разработчиков интегрировать USB-интерфейс в свои проекты оказалось более сложным по сравнению, например, с интерфейсом RS232. Кроме этого, на стороне ПК также должен быть предусмотрен специальный драйвер устройства. Как следствие, интерфейс RS232 остается очень популярным среди производителей конечных систем. Данный интерфейс хорошо изучен и в достаточной мере поддерживается операционной системой. Однако, как правило, в современных ПК физический порт RS232 не устанавливается и замещается на порты USB.

Реализовать интерфейс USB во внешнем устройстве можно двумя способами:

1. С помощью микроконтроллера, у которого интерфейс USB реализован аппаратно. В этом случае необходимо знать, как работает USB и в соответствии с этим написать программу для микроконтроллера. Кроме того, если операционная система не поддерживает стандартные классы USB, то необходимо написать драйвер для компьютера. Основным недостатком данного способа - ограниченная доступность таких микроконтроллеров и их более высокая стоимость по сравнению с обычными микроконтроллерами, которые используются для связи через "RS232".

2. Использование универсального преобразователя интерфейсов: USB и любого другого. В качестве другого интерфейса обычно используется RS232, 8-разрядная шина данных или шина TWI. В этом случае разработка специальной прошивки не потребуется, нет надобности знать, как работает USB и нет необходимости написания драйвера, т.к. производитель преобразователя предлагает свой драйвер. Недостаток - более высокая стоимость завершённой системы, а также повышенные габариты готового изделия. Решение, рассматриваемое в данном документе, основывается на использовании недорогого микроконтроллера и программной эмуляции USB-протокола в микроконтроллере. Основная проблема такого подхода заключается в сложности достижения высокой скорости. Шина USB достаточно быстродействующая: режим низкой скорости - 1.5 Мбит/сек, режим полной скорости - 12 Мбит/сек, режим высокой скорости - 480 Мбит/сек. Микроконтроллеры AVR полностью отвечают требованиям для низкоскоростного режима USB. Данное решение не рекомендуется использовать для более высоких скоростей USB.

2.2. Принцип действия.

Детальная информация по физической USB-связи может быть найдена на сайте www.usb.org. Однако для начинающих данная документация может оказаться очень сложной. Более доступно информация по USB-связи изложена в документе "USB in a Nutshell. Making Sense of the USB Standard" ("О USB в двух словах. Пойми смысл стандарта USB"), написанного Крэйгом Пикоком [2].

В данных "Рекомендациях..." объяснения ограничиваются описанием программы устройства. Физический интерфейс USB состоит из четырех проводников: 2 для питания внешнего устройства (VCC и GND) и 2 сигнальных проводника (DATA+ и DATA-). Через проводники питания передается постоянное напряжение приблизительно 5В с нагрузочной способностью максимум 500 мА. Микроконтроллер AVR питается через выводы Vcc и GND. Сигнальные проводники называются DATA+ и DATA- и управляют связью между главным (компьютер) и устройством. Сигналы в этих проводниках являются двунаправленными. Уровни напряжения - дифференциальные: когда DATA+ имеет высокий уровень, тогда DATA- находится на низком уровне. Однако, имеются некоторые случаи, когда DATA+ и DATA- имеют один и тот же уровень, например, при EOP (конец пакета).

Таким образом, программа, отвечающая за реализацию протокола USB, должна контролировать данные сигналы или управлять ими.

В соответствии со стандартом USB высокий уровень на сигнальных проводниках должен составлять 3,0...3,6В, при этом, напряжение питания Vcc шины USB, поступающее от главного (компьютера) составляет 4.4...5.25В. Таким образом, если микроконтроллер запитывается непосредственно от шины USB, то линии данных должны пройти через каскад преобразования уровней для компенсации уровней дифференциального напряжения. Другим решением может быть использование стабилизатора напряжения, который понизит напряжение Vcc до уровня 3.3В, при этом, микроконтроллер будет работать с этим пониженным напряжением и, соответственно, генерировать пониженные уровни напряжений.

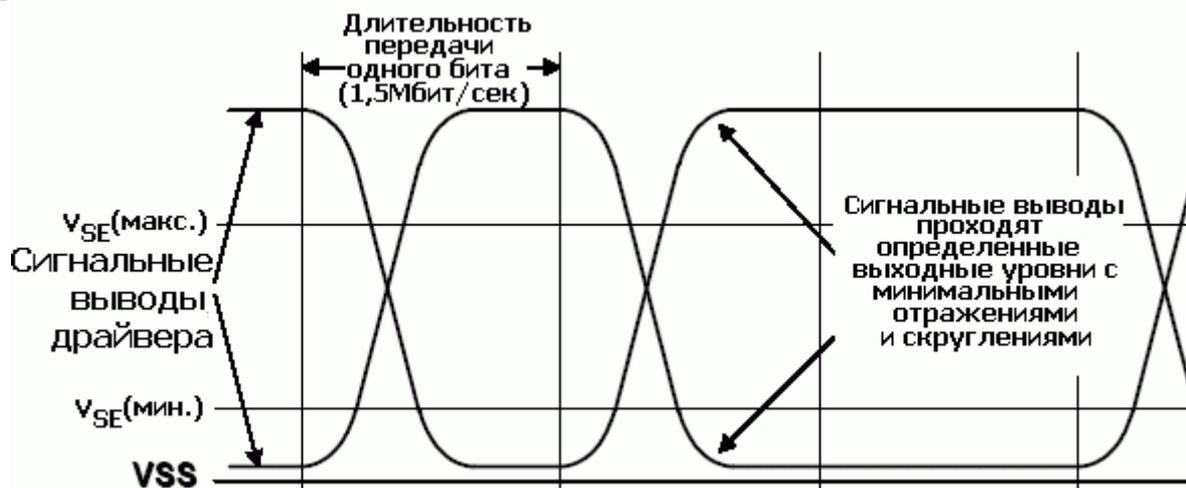


Рис. 2.1. Осциллограммы сигналов низкоскоростного драйвера

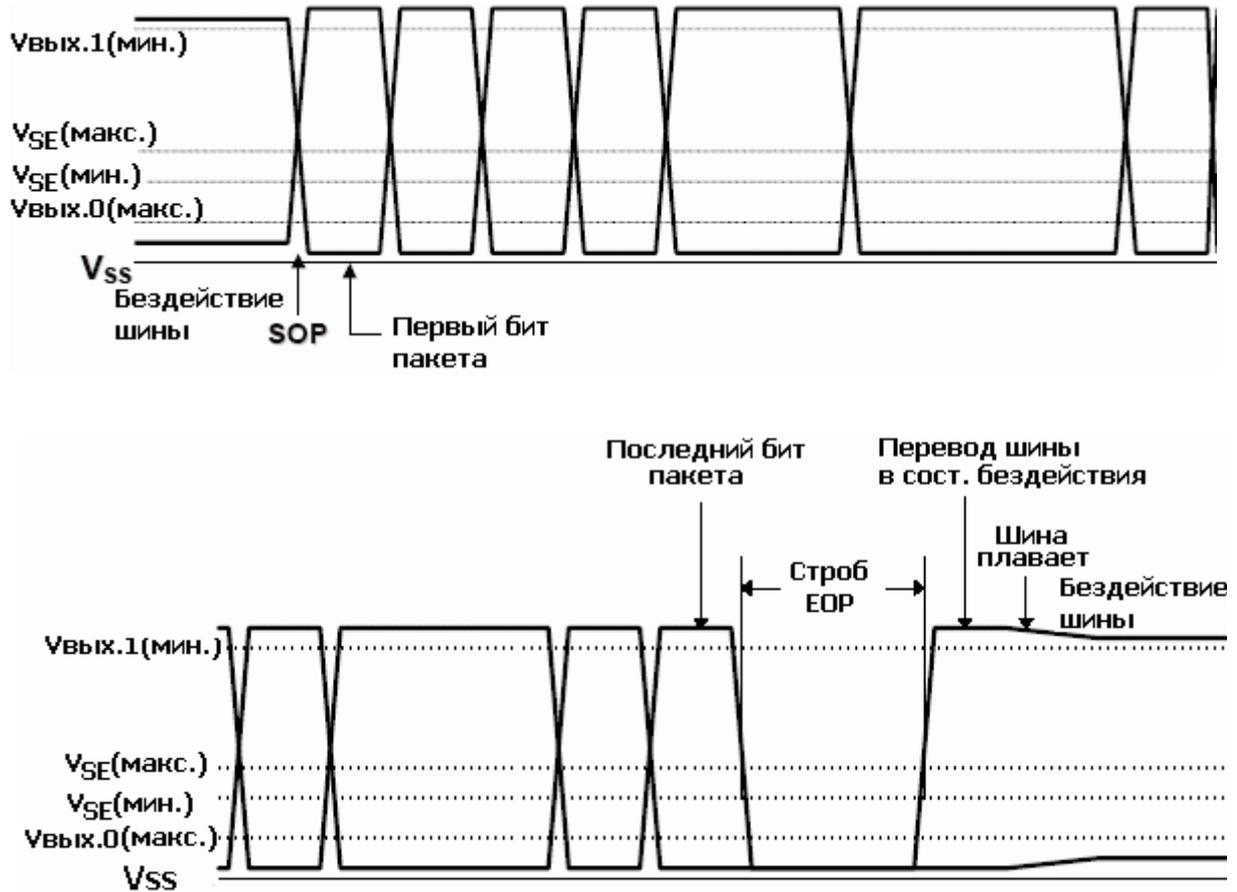


Рис. 2.2. Уровни напряжений при передаче пакетов.

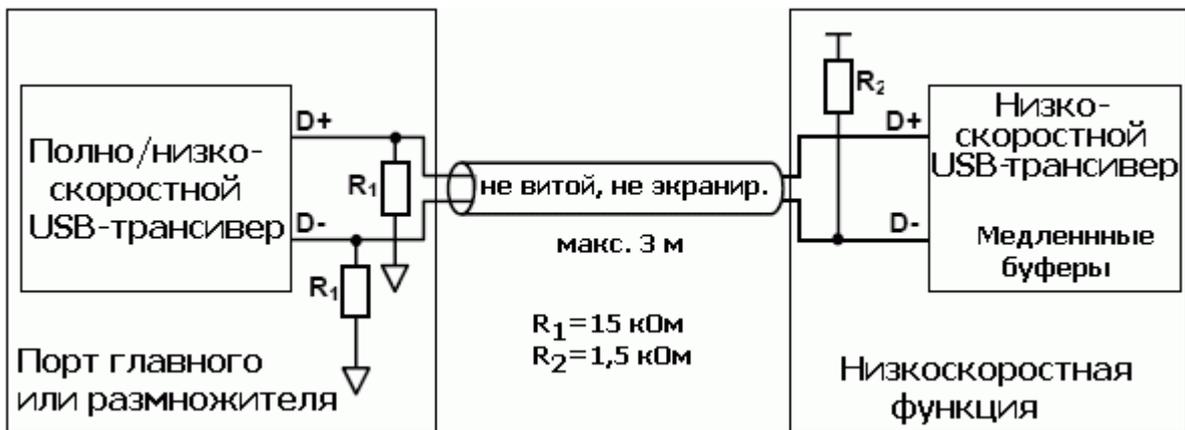


Рис. 2.3. Подключения кабеля и резисторов к низкоскоростному устройству

Принцип детекции подключения и отключения USB-устройства основан на контроле сопротивления линии USB. У низкоскоростных USB-устройств необходим подтягивающий резистор между сигналом DATA- и V_{cc} . У полноскоростных устройств данный резистор подключается к DATA+.

Определяя, на какой линии подключен подтягивающий резистор, главный компьютер определяет какое новое устройство подключено к линии USB.

После определения нового устройства главный начинает связь в соответствии с физическим протоколом USB. Протокол USB, в отличие от УАПП, основан на синхронной передаче данных. Синхронизация передатчика и приемника необходима для осуществления связи. Синхронизация выполняется путем передачи небольшого заголовка "образцовая синхронизация", который предшествует передаче данных. Данный заголовок представляет собой прямоугольные импульсы (101010), за ними передаются два 0, а затем данные.



Рис. 2.4. Образцовая синхронизация.

Для поддержания синхронизации требуется передача образцовой синхронизации каждую миллисекунду в полноскоростном режиме связи, а в низкоскоростном режиме каждую миллисекунду необходимо устанавливать низкий уровень на обеих сигнальных линиях. В аппаратно-реализованном USB-приемнике данная синхронизация гарантируется цифровой ФАПЧ (фазовая автоподстройка частоты). В данной реализации период преобразования данных должен быть синхронизирован с образцовой синхронизацией, затем ожидается два нуля, а затем начинается процесс приема данных.

Прием данных должен удовлетворять требованию возможности засинхронизировать приемник и передатчик в любой момент времени. Таким образом, не разрешается передавать непрерывный поток нулей или единиц по линиям данных. Протокол USB гарантирует синхронизацию за счет заполнения битами. Это означает, что, после 6 непрерывных единиц или нулей на линиях данных, вставляется одно одиночное изменение (один бит). Сигналы по линиям USB передаются в коде NRZI. В коде NRZI каждый '0' представляется путем сдвига текущего уровня сигнала, а каждая '1' путем удержания текущего уровня. На уровне битового заполнения это означает, что каждый нулевой бит вставляется в поток логических данных после 6 непрерывных логических 1.

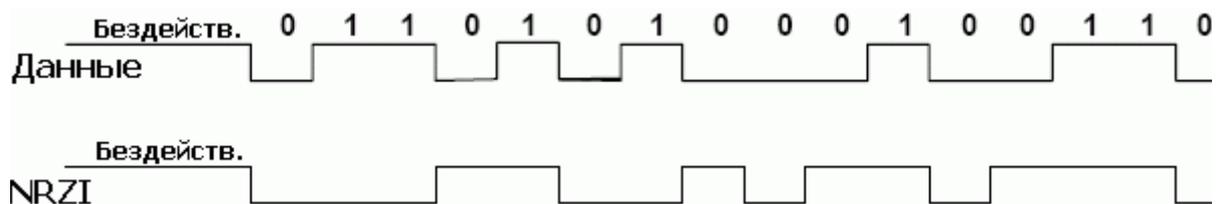


Рис. 2.5. Кодирование данных в коде NRZI.

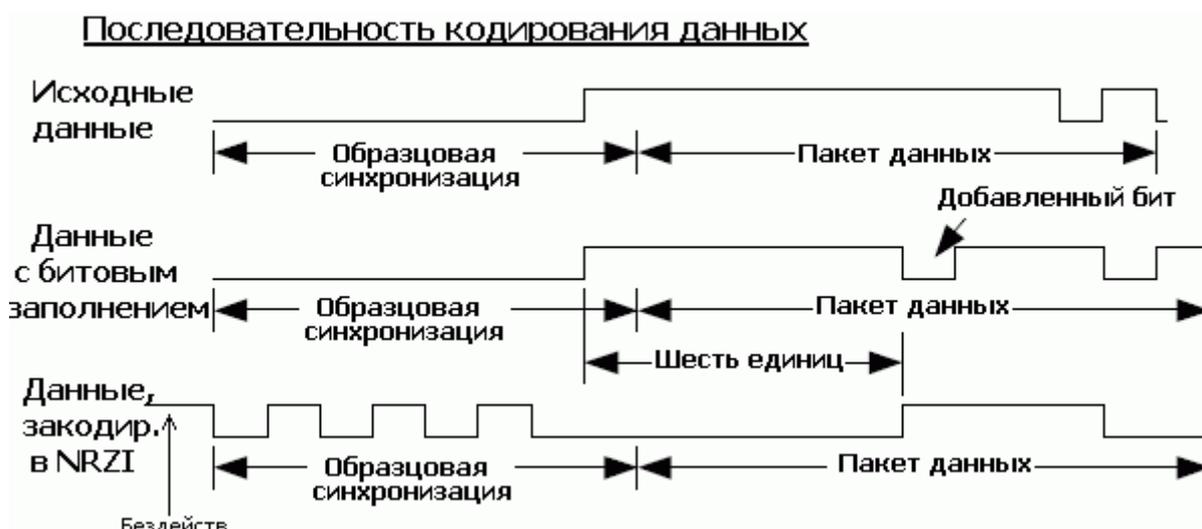


Рис. 2.6. Заполнение бит.

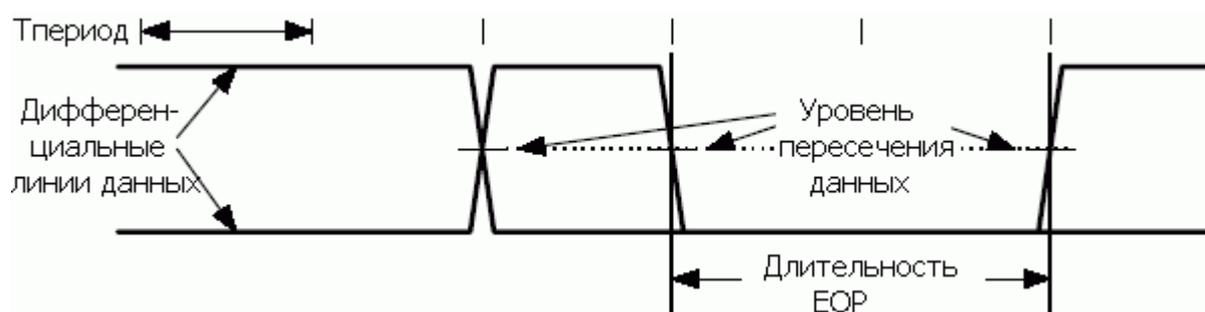


Рис. 2.7. Временная диаграмма сигнала EOP.

Уведомление об окончании передачи данных выполняется с помощью передачи сигнала "конец пакета" (EOP). EOP передается путем установки низких уровней на обеих линиях данных DATA+ и DATA-. EOP передается непродолжительное время (минимум два периода скорости данных). После этого, выполняется следующая транзакция.

Данные, которые передаются между образцовой синхронизацией и EOP, закодированы в коде NRZI. Поток данных состоит из пакетов, пакет в свою очередь состоит из нескольких полей: поле синхронизации (образцовая синхронизация), идентификатор пакета (PacketID, PID), поле адреса (ADDR), поле конечной точки (ENDP), данные и поле циклического избыточного контроля (CRC). USB подразумевает четыре типа передачи: передача управления, передача прерывания, изохронная передача и передача потока. Каждый из этих типов передач специфичен для различных требований устройства.

В данной реализации рассматривается тип передачи управления. Данный режим, как правило, используется для управления настройками устройства,

однако, может также использоваться для передачи общего назначения. Режим передачи управления должен присутствовать у каждого USB-устройства, т.к. он используется для конфигурации при подключении устройства, когда необходимо получить информацию об устройстве, установленный адрес устройства и пр. Каждая передача управления состоит из нескольких стадий: стадия установки, стадия данных и стадия статуса.

Данные по шине USB передаются пакетами, по несколько байт в каждом. Размер пакета определяется каждым устройством, но его предельный размер ограничен. Для низкоскоростных устройств максимальный размер пакета равен 8 байтам. Данный 8-байтный пакет вместе с начальным и конечным полем должны быть приняты в буфер устройства за одну USB-передачу. В аппаратно-реализованных USB-приемниках различные части передачи автоматически дешифрируются и Устройство уведомляется, когда все сообщение назначено отдельному устройству. При программной реализации USB-сообщение дешифрируется программно после приема в буфер всего сообщения. Вследствие этого возникают требования и ограничения: устройство должно иметь буфер для хранения всего USB-сообщения, иметь другой буфер для USB-передачи (подготовка данных для передачи), а также выполнять администрирование с дешифрованием и проверкой сообщений. Кроме того, необходима программа для выполнения быстрого и точного синхронного приема с физических линий в буфер и передачи из буфера на линии. Данные возможности ограничиваются быстродействием микроконтроллера и размером памяти программ/данных, т.к. программа должна быть тщательно оптимизирована. В некоторых случаях вычислительные возможности микроконтроллера очень близки к минимальным требованиям, поэтому, вся программа написана на Ассемблере.

2.3. Аппаратная реализация.

На рисунках 2.8 и 2.9 показана схема подключения микроконтроллера к шине USB. Данные схемы имеют специфичное назначение: преобразователь USB - RS232. На них также реализованы специфичные функции, такие как непосредственное управление выводом и чтение/запись ЭСППЗУ.

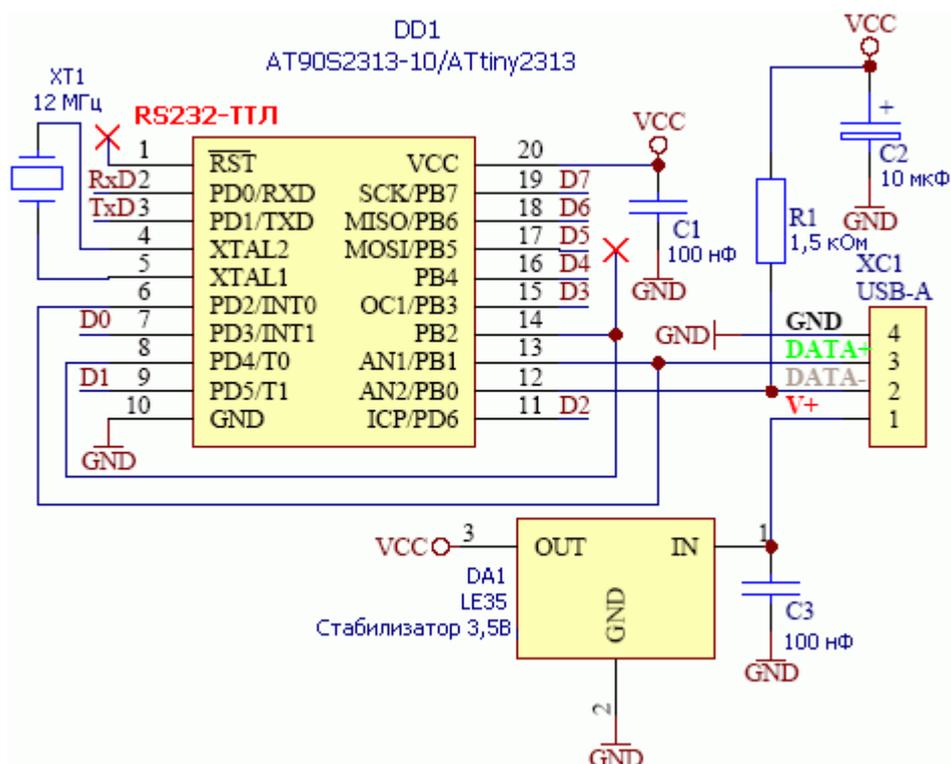


Рис. 2.8. Подключение шины USB к микроконтроллеру ATtiny2313 в качестве преобразователя USB - RS232 с 32 байтным буфером FIFO, 8-разрядным управлением вводом-выводом и 128 байтным ЭСППЗУ

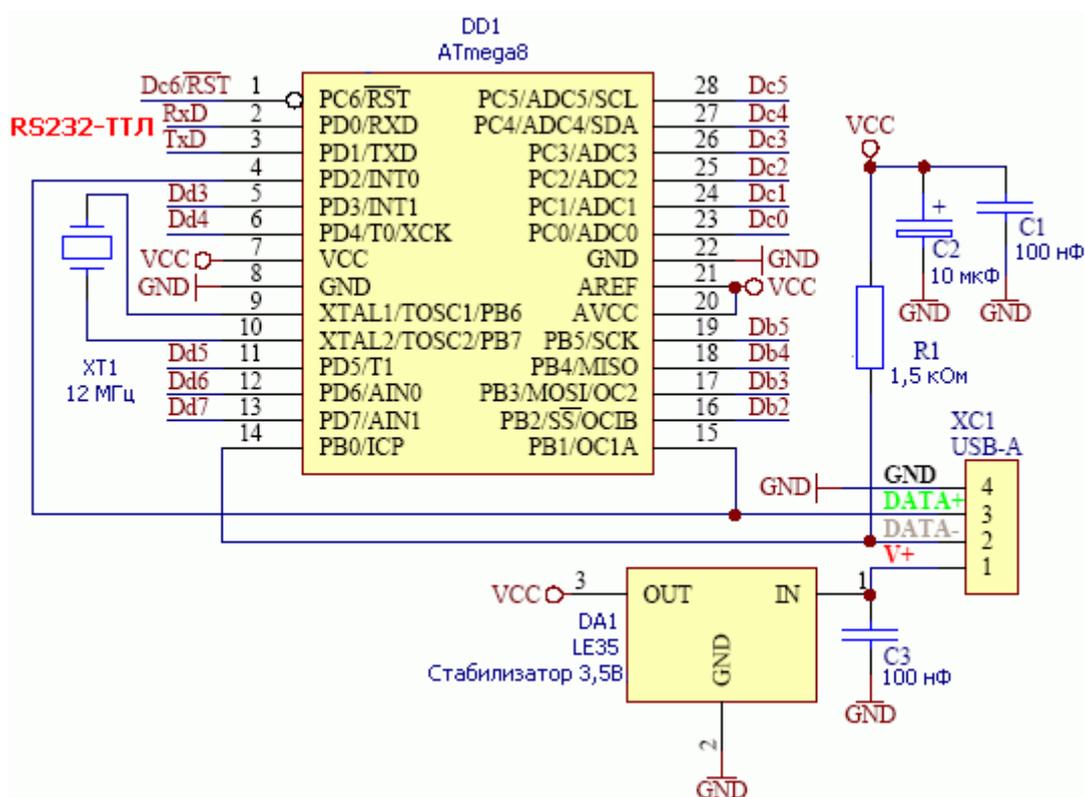


Рис. 2.9. Подключение шины USB с микроконтроллером ATmega8/48/88/168 в качестве преобразователя USB - RS232 с 800 байтным FIFO, управлением вводом-выводом и 512 байтным ЭСППЗУ

Линии данных USB, DATA- и DATA+, подключены к выводам PB0 и PB1 микроконтроллера AVR. Данное подключение нельзя изменить, т.к. в программе используются уникальности ядра AVR для быстрого приема сигнала: битовый сигнал захватывается из линий данных и младший разряд (PB0) сдвигается вправо в перенос, а затем в приемный регистр, который накапливает биты, принятых с линий данных. PB1 используется в качестве входного сигнала, поскольку у 8-выводного AT90S2323 данный ввод может использоваться в качестве входа внешнего прерывания без каких-либо внешних подключений к INT0. В других микроконтроллерах AVR необходимо внешнее подключение DATA+ к выводу INT0, что гарантирует неизменность программы при переходе между микроконтроллерами.

Для соответствующего подключения USB-устройства и сигнализации микроконтроллер AVR, работающий как низкоскоростное USB-устройство, должен иметь подтягивающий резистор 1,5 кОм на линии DATA-.

Напряжение V_{cc} , поступающее от USB-порта компьютера, может варьироваться в диапазоне 4.4...5.25В. Перед его подачей к микроконтроллеру и подтягивающему резистору сопротивлением 1,5 кОм данное напряжение стабилизируется на уровне 3.0 - 3.6В. Тип стабилизатора напряжения зависит от уровня нагрузки целевой системы. Стабилизатор напряжения должен относиться к типу стабилизаторов с малым минимальным перепадом напряжения. В схемах на рисунках 8 и 9 используется стабилизатор LE35 с номинальным выходным напряжением 3,5В. Однако можно использовать любое другое решение по стабилизации напряжения, если оно отвечает требованиям проектируемой системы. В некоторых случаях может быть вполне приемлемым параметрический стабилизатор на основе стабилитрона.

Остальные компоненты необходимы для соответствующей работы микроконтроллера: кварцевый резонатор для синхронизации микроконтроллера и конденсаторы для фильтрации питания.

Таким образом, для получения завершеного устройства, которое обеспечивает связь с компьютером через интерфейс USB, необходимо небольшое количество элементов. Для расширения функциональных возможностей могут быть добавлены дополнительные компоненты.

Инфракрасный датчик TSOP1738 может использоваться для приема инфракрасного сигнала. Преобразователь уровня TTL - RS232 MAX232 необходимо добавить при необходимости разработки преобразователя интерфейсов USB - RS232. Для управления светодиодами или дисплеем их необходимо подключить к линиям ввода-вывода непосредственно или через резисторы.

2.4. Программная реализация

Протокол USB-приема и дешифрации полностью реализован программно. Программа вначале принимает битовый поток USB в один USB-пакет во внутреннем буфере. Начало приема инициируется по внешнему прерыванию INT0, которое отвечает за образцовую синхронизацию. В процессе приема проверяется только сигнал конца пакета (определяется только EOP). Это необходимо ввиду очень высокой скорости передачи данных по шине USB. После успешного приема программа дешифрует пакеты данных и анализирует их. Во-первых, с помощью адреса анализируется, что принятый пакет относится к данному устройству. Адрес передается при каждой USB-транзакции и, следовательно, устройство знает, что переданные данные относятся именно к нему. Дешифрация USB-адреса должна выполняться очень быстро, поскольку, в случае определения действительного пакета с заданным адресом, устройство должно ответить компьютеру подтверждающим пакетом ACK. Таким образом, дешифрация адреса является критичной частью USB-ответа.

После приема битового потока мы получаем последовательность, закодированную в коде NRZI, путем поразрядного заполнения входного буфера. В процессе дешифрации мы вначале удаляем поразрядное заполнение, а затем NRZI-кодирование. Все данные изменения выполняются во втором буфере (копия приемного буфера). После дешифрации текущего пакета может приниматься новый пакет. Для данной точки, скорость дешифрации не так важна, т.к. устройство может задержать свой ответ. Если компьютер запрашивает ответ в процессе дешифрации, то устройство должно ответить незамедлительно NAK (нет подтверждения), исходя из чего компьютер поймет о неготовности устройства. Таким образом, микроконтроллер должен быть способен принимать пакеты от компьютера в процессе дешифрации, определить относится ли транзакция к устройству, а затем отправить пакет NAK, если дешифрация еще находится в процессе выполнения. В этом случае компьютер отправит запрос снова. Микроконтроллер также дешифрует основную USB-транзакцию и выполняет запрашиваемое действие; например, отправка символа по линии RS232 и ожидание завершения транзакции, а также подготовка соответствующего ответа. При выполнении данного процесса устройство будет прерываться некоторыми пакетами, поступающих от компьютера, обычно пакеты IN для получения ответа от устройства. На данные пакеты IN устройство должно ответить пакетами NAK. Если ответ готов и устройство выполнило требуемое действие, ответ должен вначале пройти через функцию вычисления и присоединения CRC, затем выполняется NRZI-кодирование, а затем поразрядное заполнение. Теперь, когда компьютер запрашивает ответ, данный битовый поток передается по линиям данных в соответствии с требованиями стандарта USB.

2.5. Описание программы микроконтроллера.

Далее описываются основные части программы для микроконтроллера. Программа разделена на блоки: процедуры прерывания, процедуры дешифрации, USB-прием, USB-передача, дешифрация запрашиваемого действия и выполнение требуемых действий.

Пользователь, при необходимости, может добавить собственные функции. В коде программы можно найти примеры по тому, как сделать специфичные пользовательские функции. Таким образом, пользователь на основе существующих встроенных функций может написать собственные расширения. Например, на основе встроенной функции непосредственного управления выводом можно добавить поддержку TWI.

Процедура обработки прерывания "EXT_INT0"

Внешнее прерывание 0 находится в активном состоянии в течение всего времени выполнения программы. Данная процедура инициирует прием данных, последовательно передаваемых по шине USB. Внешнее прерывание возникает при появлении нарастающего фронта на выводе INT0. Нарастающий фронт указывает на начало образцовой синхронизации (см. рисунок 4) и инициирует выполнение процедуры USB-приема.

Вначале процесс оцифровки данных должен быть засинхронизирован к середине бита. Это выполняется в соответствии с образцовой синхронизацией. Поскольку длительность передачи одного бита равно только 8 периодам синхронизации XTAL и возникновение прерывания может быть задержано (+/- 4 периода), то синхронизацию к фронтам образцовой синхронизации необходимо выполнить тщательно. Окончание передачи образцовой синхронизации и начало передачи бит данных определяется следующим за образцовой синхронизацией двумя битами с низким уровнем (см. рисунок 4).

После этого начинается фактическая оцифровка данных. Оцифровка выполняется по середине бита. При скорости передачи данных 1,5 Мбит/секунду (1.5МГц) и частоте синхронизации микроконтроллера 12 МГц в нашем распоряжении имеется только 8 тактов для оцифровки данных, записи результата в однобайтный буфер, сдвига данных в однобайтном буфере, проверки на прием всего байта, записи байта в статическое ОЗУ и определения окончания пакета (EOP). Это наиболее критическая часть программы; все должно быть сделано синхронно и за установленные временные рамки. После приема всего USB-пакета необходимо выполнить дешифрацию пакета. Вначале мы должны быстро определить тип пакета (SETUP, IN, OUT, DATA) и принятый USB-адрес. Быстрая дешифрация должна выполняться внутри процедуры обработки прерывания, т.к. необходимо очень быстро ответить после приема USB-пакета (устройство должно ответить пакетом ACK, если принят пакет с адресом устройства и NAK, если принят пакет, адресованный устройству, но ответ еще не готов).

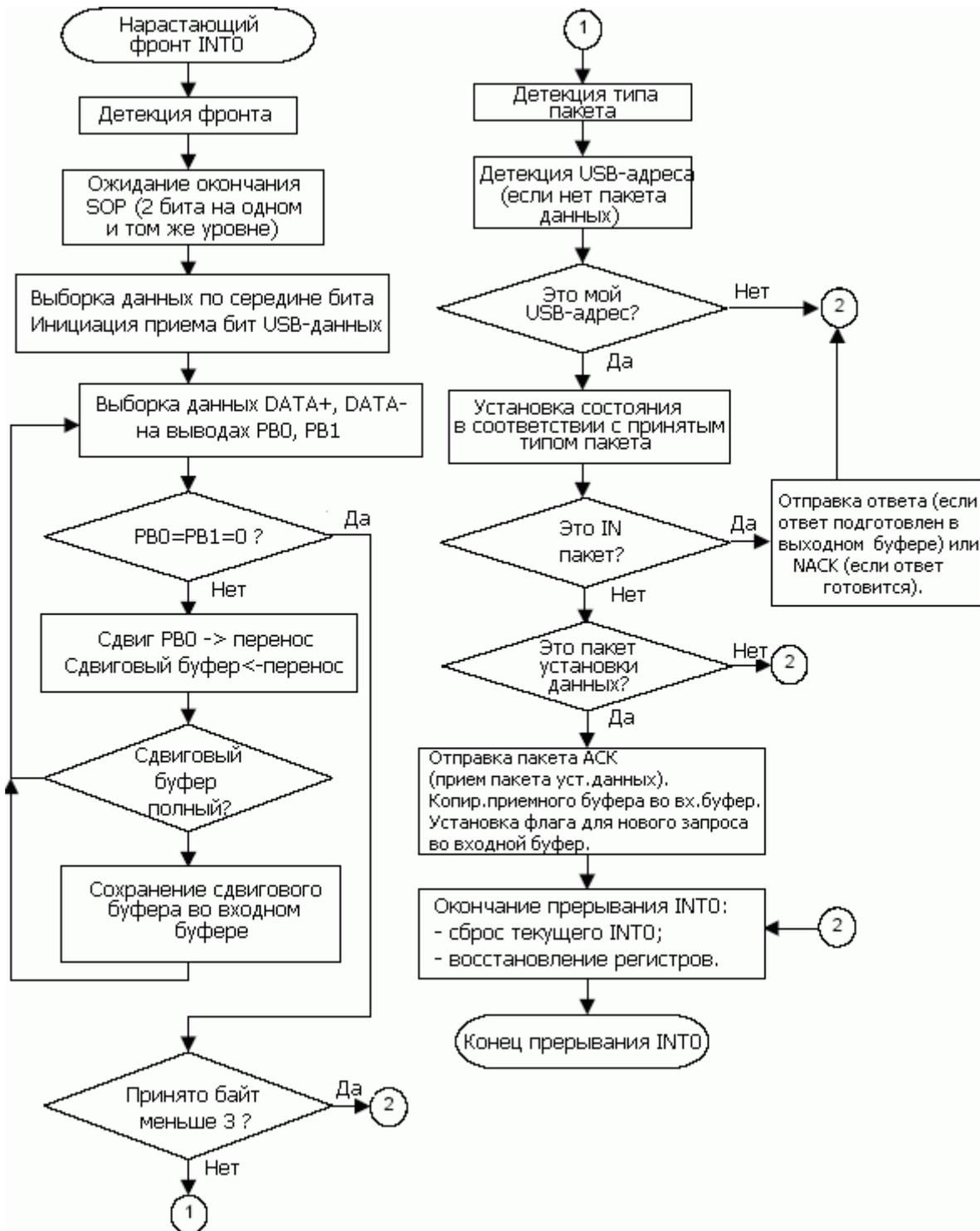


Рис. 2.10. Блок-схема процедуры приема.

В конце процедуры приема (после отправки пакета ACK/NAK) записанные в буфер данные должны быть скопированы в другой буфер, в котором выполняется дешифрация данных. Это необходимо для освобождения приемного буфера для приема нового пакета.

В процессе приема дешифрируется тип пакета и устанавливается соответствующее значение флага. Данный флаг проверяется в основном цикле

программы и в зависимости от его значения выполняется соответствующее действие и подготавливается соответствующий ответ без предъявления каких-либо требований по быстродействию микроконтроллера.

Для сохранения высокой скорости обработки внешнего прерывания INT0 оно должно оставаться активным постоянно, даже при обработке других прерываний (например, прерывание по последовательному приему). Быстрота приема в процедуре обработки прерывания INT0 очень важна, поэтому, программу необходимо оптимизировать по быстродействию и времени выполнения. Важной проблемой является оптимизация резервирования регистра в процедурах обработки прерываний.

Основной программный цикл

Основной программный цикл очень простой. В нем необходимо только проверять состояние флага действия и определять, что делать после приема данных. Кроме того, в основном цикле проверяется, сброшен ли USB-интерфейс (обе линии данных находятся в низком уровне в течение длительного времени) и, в случае определения такового состояния, выполняется переинициализация устройства. Если флаг действия активен, то вызывается соответствующее действие: дешифрация кода NRZI в пакете, удаление поразрядного заполнения и подготовка запрашиваемого ответа в буфере передатчика (с поразрядным заполнением и NRZI-кодированием). После этого активизируется флаг для сигнализации о готовности ответа к отправке.

Физическая передача выходного буфера по линиям USB выполняется в процедуре приема как ответ на пакет IN.

Ниже приведено краткое описание подпрограмм:

Reset:

Выполняет инициализацию ресурсов микроконтроллера AVR: стек, линии ввода-вывода, буферы USB, прерывания.

Main:

Основной цикл программы. Проверяет состояние флага действия; если флаг установлен, то выполняет требуемое действие. Дополнительно, данная процедура проверяет сброс линий данных USB-шины и в случае определения сброса переинициализирует USB-интерфейс.

Int0Handler:

Процедура обработки внешнего прерывания INT0. Основной процессор приема/передачи. Запись данных в буфер, определение получателя USB-пакета (USB-адреса), распознавание пакета, отправка ответа компьютеру. Является основой USB устройства.

SetMyNewUSBAddresses:

Процедура изменения USB-адреса. Адрес изменяется и кодируется в NRZI эквивалент. Это необходимо, т.к. необходимо быстро выполнить дешифрацию адреса в процессе приема USB-пакета.

FinishReceiving:

Копирует непроверенные данные из принятого USB-пакета для его дешифрации (дешифрация NRZI и поразрядного заполнения).

USBreset:

Инициализирует интерфейс USB к значениям по умолчанию (переход к состоянию, эквивалентного состоянию сразу после подачи питания).

SendPreparedUSBAnswer:

Отправляет подготовленное содержимое выходного буфера по линиям USB. NRZI-кодирование и поразрядное заполнение выполняется в процессе передачи. Пакет завершается сигналом EOP.

ToggleDATAID:

Переключает идентификатор пакета DATAID (PID) между DATA0 и DATA1 PID. Данное переключение необходимо в процессе передачи согласно техническим требованиям USB.

ComposeZeroDATA1PIDAnswer:

Составляет нулевой ответ для передачи. Нулевой ответ не содержит данных и используется в некоторых случаях в качестве ответа, когда в устройстве нет доступных дополнительных данных.

InitACKBuffer:

Инициализирует буфер в ОЗУ данными ACK (пакет подтверждения ACK). Данный буфер часто отправляется в качестве ответа, что он всегда остается готовым в памяти.

SendACK:

Передает пакет ACK по линиям USB.

InitNAKBuffer:

Инициализирует буфер в ОЗУ данными NAK (пакет нет подтверждения NAK). Данный буфер часто отправляется как ответ о готовности буфера в памяти.

SendNAK:

Передает пакет NAK по линиям USB.

ComposeSTALL:

Инициализирует буфер в ОЗУ данными STALL (пакет STALL). Данный буфер часто отправляется в качестве ответа о готовности буфера в памяти.

DecodeNRZI:

Выполняет NRZI-дешифрацию. Данные, поступающие с линий USB, закодированы в коде NRZI. Данная процедура удаляет NRZI-кодирование из данных.

BitStuff:

Удаляет/добавляет поразрядное заполнение в принятых USB-данных. Поразрядное заполнение добавляется в компьютере на аппаратном уровне в соответствии с требованиями USB, чтобы гарантировать синхронизацию оцифровки данных. Данная процедура генерирует принятые данные без поразрядного заполнения или формирует данные с поразрядным заполнением

для передачи.

ShiftInsertBuffer:

Вспомогательная процедура, которая используется при добавлении поразрядного заполнения. Добавляет один бит к выходному буферу данных и, таким образом, увеличивает длину буфера.

ShiftDeleteBuffer:

Вспомогательная процедура, которая используется для удаления поразрядного заполнения. Удаляет один бит из выходного буфера данных и, таким образом, уменьшает длину буфера.

MirrorInBufferBytes:

Изменяет порядок разрядов в байте, т.к. у принятых данных с линий USB обратный порядок (мл. разряд/ст. разряд).

CheckCRCIn:

Выполняет проверку CRC (контроль циклической избыточности) в пакете принятых данных. CRC добавляется к USB-пакету для детекции повреждения данных.

AddCRCOut:

Добавляет поле CRC в выходной пакет данных. CRC вычисляется в соответствии с техническими требованиями USB для данных USB-полей.

CheckCRC:

Вспомогательная процедура, которая используется для проверки и добавления CRC.

LoadDescriptorFromROM:

Загрузка данных из ПЗУ в выходной буфер USB (в качестве USB-ответа).

LoadDescriptorFromROMZeroInsert:

Загрузка данных из ПЗУ в выходной буфер USB (в качестве USB-ответа), но каждый четный байт записывается с нулевым значением. Используется, когда дескриптор строки запрашивается в формате UNICODE (сохранение ПЗУ).

LoadDescriptorFromSRAM:

Загрузка данных из ОЗУ в выходной буфер USB (в качестве USB-ответа).

LoadDescriptorFromEEPROM:

Загрузка данных из ЭСППЗУ в выходной буфер USB (в качестве USB-ответа).

Load[X]Descriptor:

Выполняет выбор источника ответа: ПЗУ, ОЗУ или ЭСППЗУ.

PrepareUSBOutAnswer:

Подготавливает USB-ответ в выходном буфере в соответствии с запросом от компьютера и выполняет запрашиваемое действие. Добавляет поразрядное заполнение в ответ.

PrepareUSBAnswer:

Основная процедура для выполнения запрашиваемого действия и подготовки соответствующего ответа. Процедура вначале определит, какое

действие выполнить (путем поиска номера функции во входном пакете данных), а затем выполнит запрашиваемую функцию. Параметры функции находятся во входном пакете данных.

Процедура разделена на две части:

- стандартные запросы;
- специфические запросы производителя.

Стандартные запросы необходимы в соответствии с техническими требованиями USB (SET_ADDRESS, GET_DESCRIPTOR, ...).

Специфическими запросами производителя являются запросы, которые могут получить специфические данные производителя (передача управления IN USB). Передача управления IN USB используется для связи AVR с компьютером. Разработчики могут добавить свои собственные функции и, таким образом, расширить гибкость устройства. Различные документированные встроенные функции в исходном коде могут использоваться в качестве шаблона для разработки собственных функций.

2.6. Стандартные функции USB (стандартные запросы)

ComposeGET_STATUS;
 ComposeCLEAR_FEATURE;
 ComposeSET_FEATURE;
 ComposeSET_ADDRESS;
 ComposeGET_DESCRIPTOR;
 ComposeSET_DESCRIPTOR;
 ComposeGET_CONFIGURATION;
 ComposeSET_CONFIGURATION;
 ComposeGET_INTERFACE;
 ComposeSET_INTERFACE;
 ComposeSYNCH_FRAME;

2.7. Функции USB-производителя (запросы производителя)

DoSetInfraBufferEmpty;
 DoGetInfraCode;
 DoSetDataPortDirection;
 DoGetDataPortDirection;
 DoSetOutDataPort;
 DoGetOutDataPort;
 DoGetInDataPort;
 DoEEPROMRead;
 DoEEPROMWrite;
 DoRS232Send;
 DoRS232Read;
 DoSetRS232Baud;
 DoGetRS232Baud;
 DoGetRS232Buffer;
 DoSetRS232DataBits;

DoGetRS232DataBits;
 DoSetRS232Parity;
 DoGetRS232Parity;
 DoSetRS232StopBits;
 DoGetRS232StopBits;

2.8. Структуры данных (USB-дескрипторы и строки)

DeviceDescriptor;
 ConfigDescriptor;
 LangIDStringDescriptor;
 VendorStringDescriptor;
 DevNameStringDescriptor;

2.9. Формат входного сообщения компьютера

Как сказано выше, USB-устройство использует USB передачу управления. Данный тип передачи использует формат данных, определенный в технических требованиях к USB. В документе описываются подробности передачи управления и, таким образом, того, как данное устройство связывается с компьютером. Микроконтроллер AVR использует управляющую конечную точку IN. Передача данных между компьютером и микроконтроллером AVR выполнена в соответствии с данным примером.

В дополнение к фактической передаче управления обсуждается формат поля DATA0/1. Передача управления определяет на своем этапе установки стандартный запрос длиной 8 байт.

Стандартный пакет установки используется для детекции и конфигурации устройства после подачи питания. Данный пакет использует стандартный тип запроса в поле bmRequestType (разряды D6-D5 = 0). Все остальные значения полей (bRequest, wValue, wIndex, wLength) могут быть найдены в технических требованиях USB.

Каждый пакет установки содержит 8 байт, которые используются в соответствии с описанием в таблице 1.

Таблица 1. Поля стандартных пакетов установки (передача управления)

Смещение	Поле	Размер	Значение	Описание
0	bmRequestType	1	побитовое отображение	Характеристики запроса D7 - направление данных xfer: 0 = главный устройству; 1 = устройство главному. D6..5 - Тип: 0 = стандарт; 1 = класс; 2 = производитель; 3 = зарезервировано. D4..0 - Получатель 0 = устройство; 1 = интерфейс; 2 = конечная точка; 3 = другое; 4..31 = зарезервировано.
1	bRequest	1	Значение	Специфический запрос
2	wValue	2	Значение	Поле, измеряемое в словах, которое варьируется в зависимости от запроса
4	wIndex	2	Индекс или смещение	Поле, измеряемое в словах, которое варьируется в зависимости от запроса - обычно используется для пересылки индекса или смещения
6	wLength	2	Подсчет	Количество байт для передачи, если наступила фаза данных

Таблица 2. Стандартные запросы устройства

BmRequest-Type	bRequest	wValue	wIndex	wLength	Данные
0000000B 0000001B 0000010B	CLEAR_FEATURE	Переключатель особенностей	Нуль Интерфейс Конечная точка	Нуль	Нет
1000000B	GET_CONFIGURATION	Нуль	Нуль	Один	Конфигурационное значение
1000000B	GET_DESCRIPTOR	Тип дескриптора и индекс дескриптора	Нуль или язык ID	Длина дескриптора	Дескриптор
1000001B	GET_INTERFACE	Нуль	Интерфейс	Один	Альтернативный интерфейс
1000000B 1000001B 1000010B	GET_STATUS	Нуль	Нуль Интерфейс Конечная точка	Два	Статус устройства, интерфейса или конечной точки
0000000B	SET_ADDRESS	Адрес устройства	Нуль	Нуль	Нет
0000000B	SET_CONFIGURATION	Конфигурационное значение	Нуль	Нуль	Нет
0000000B	SET_DESCRIPTOR	Тип дескриптора и индекс дескриптора	Нуль или язык ID	Длина дескриптора	Дескриптор
0000000B 0000001B 0000010B	SET_FEATURE	Переключатель особенностей	Нуль Интерфейс Конечная точка	Нуль	Нет
0000001B	SET_INTERFACE	Альтернативная установка	Интерфейс	Нуль	Нет
1000010B	SYNCH_FRAME	Нуль	Конечная точка	Два	Номер кадра

Таблица 3. Запросы производителя, используемые в программе для вызова функций

Тип запроса BmRequest-Type	Наименование запрашиваемой функции bRequest	Номер bRequest	Параметр 1wValue	Параметр 2wIndex	wLength	Данные
110xxxxxB	FNCNumberDoSetInfraBufferEmpty	1	Нет	Нет	1	Статус
110xxxxxB	FNCNumberDoGetInfraCode	2	Нет	Нет	1	Статус
110xxxxxB	FNCNumberDoSetDataPortDirection	3	DDRБ DDRC	DDRД Используемые порты	1	Статус
110xxxxxB	FNCNumberDoGetDataPortDirection	4	Нет	Нет	3	DDRБ DDRC DDRД
110xxxxxB	FNCNumberDoSetOutDataPort	5	PORTB PORTC	PORTD Используемые порты	1	Статус
110xxxxxB	FNCNumberDoGetOutDataPort	6	Нет	Нет	3	PORTB PORTC PORTD
110xxxxxB	FNCNumberDoGetInDataPort	7	Нет	Нет	3	PINB PINC PIND
110xxxxxB	FNCNumberDoEEPROMRead	8	Адрес	Нет	Длина	байты ЭСППЗУ
110xxxxxB	FNCNumberDoEEPROMWrite	9	Адрес	Значение ЭСППЗУ	1	Статус
110xxxxxB	FNCNumberDoRS232Send	10	Значение байта RS232	Нет	1	Статус
110xxxxxB	FNCNumberDoRS232Read	11	Нет	Нет	2	Статус
110xxxxxB	FNCNumberDoSetRS232Baud	12	Мл. байт скорости связи	Ст. байт скорости связи	1	Статус
110xxxxxB	FNCNumberDoGetRS232Baud	13	Нет	Нет	2	Скорость связи
110xxxxxB	FNCNumberDoGetRS232Buffer	14	Нет	Нет	Длина	Байты RS232 из FIFO
110xxxxxB	FNCNumberDoSetRS232DataBits	15	Значение бит данных	Нет	1	Статус
110xxxxxB	FNCNumberDoGetRS232DataBits	16	Нет	Нет	1	Значение бит данных
110xxxxxB	FNCNumberDoSetRS232Parity	17	Значение паритета	Нет	1	Статус
110xxxxxB	FNCNumberDoGetRS232Parity	18	Нет	Нет	1	Значение паритета
110xxxxxB	FNCNumberDoSetRS232StopBits	19	Значение стоп-бит	Нет	1	Статус
110xxxxxB	FNCNumberDoGetRS232StopBits	20	Нет	Нет	1	Значения стоп-бит

Режим передачи управления используется для пользовательской связи, реализованной в программе как специальные функции. Запрос производителя используется в поле `bmRequestType` (разряды D6-D5 = 2). Здесь все следующие поля (`bRequest`, `wValue`, `wIndex`) могут модифицироваться в соответствии с целями программиста. В нашей реализации поле `bRequest` используется для задания номера функции, а следующие поля - для задания параметров функции. Первый параметр задается в слоте `wValue`, а второй - в позиции `wIndex`.

Рассмотрим пример записи в ЭСППЗУ. В качестве номера функции выбирается `bRequest = 9`. Поле `wValue` используется для задания адреса ЭСППЗУ, а записываемое значение (данные ЭСППЗУ) указываются в поле `wIndex`. В соответствии с этим получаем следующую функцию: `EEPROMWrite(Address, Value)`.

Если пользовательских функций недостаточно, то достаточно добавить номер функции и тело требуемой функции в программу. В качестве методики можно использовать существующие встроенные функции в программе (см. исходный код).

Главный шины USB (компьютер) также связывается с устройством посредством передач управления IN. Главный отправляет устройству 8-байтный IN-пакет данных в формате определенном выше (номер функции и параметры), а затем устройство отвечает запрошенными данными. Размер запрашиваемых данных ограничен в программе в некоторых случаях до 255 байт, но основное ограничение выполняется драйвером устройства на стороне управляющего компьютера. Существующий драйвер поддерживает 8-байтную длину ответов в запросах, относящихся к типу запросов производителя.

2.10. Создание собственной программы

Пользователи могут добавлять новые функции и расширять функциональные возможности устройства. В программе имеется 3 примера добавления новых функций: `DoUserFunctionX` ($X=0,1,2$). Изучите данные примеры, чтобы понять, как добавить похожие расширенные функции. Содержимое функций зависит только от требований устройства.

Имя идентификации и устройства, представленное на стороне устройства, может быть изменено в программе микроконтроллера. Данное имя хранится в программе в виде строки и может быть заменено на любую другую строку. Однако данные имена рекомендуется изменять вместе с USB PID (идентификатор продукции) и VID (идентификатор производителя) для корректного распознавания в целевой системе. VID вместе с PID должны быть уникальными для заданного типа устройства. Таким образом, рекомендуется, при изменении функциональных возможностей устройства, изменять PID и/или VID. Идентификатор производителя зависит от производителя USB-устройства и должен быть определен USB-организацией (более подробно смотри в [1]).

Каждый производитель имеет собственный идентификатор и, следовательно, его значение не может быть изменено на другое несоответствующее значение. Однако идентификатор продукции зависит только

от выбора производителя, а назначением PID является распознавание различных устройств одного и того же производителя. В данных рекомендациях по применению установлены VID 0x03EB и PID 0x21FF (идентификатор Atmel). Нельзя использовать данный идентификатор производителя в своей целевой системе.

2.11. Программа для ПК.

Для обеспечения связи с устройством необходима некоторая программная поддержка на стороне ПК. Программа разделена на три уровня:

1. Драйвер устройства: используется для связи на низком уровне с устройством и для инсталляции в операционную систему (Windows98/ME/NT/XP).

2. DLL-библиотека: используется для инкапсуляции функций устройства и связи с драйвером устройства. DLL упрощает доступ к функциям устройства из программы пользователя. В состав библиотеки входят некоторые функции устройства и операционной системы (задачи, буферы и др.).

3. Приложение пользователя: создает пользовательский интерфейс для удобной связи между пользователем и устройством. Вызывает функции только из DLL-библиотеки.

Драйвер устройства и файлы инсталляции

Первоначально, USB-устройство подключается к компьютерному USB-порту, затем операционная система определит устройство и запросит файлы драйвера. Данный процесс называется инсталляцией устройства. Для выполнения инсталляции необходимо не только создать драйвер устройства, но также инсталляционный скрипт, в котором описывается последовательность инсталляции.

Драйвер устройства выполнен с использованием Windows2000 DDK (набор для разработки драйвера). Разработка драйвера USB основана на входящих в DDK примерах - IsoUsb. Данный драйвер изменен для организации USB-связи с AVR. В оригинальном исходном коде изменения выполнялись вокруг IOCTL-связи путем добавления или расширения, т.к. устройство связывается с компьютером посредством IOCTL-вызовов. Для уменьшения размера кода драйвера из исходного кода были удалены неиспользуемые части. Имя драйвера - "AVR309.sys". Он работает как отправитель команд USB-устройству (управляет входящей передачей). Драйвер работает под всеми версиями 32-разрядных операционных систем Windows, за исключением Win95. Инсталляционный скрипт, записанный в INF-файл, используется в процессе инсталляции устройства. В данном INF-файле описаны различные инсталляционные шаги. Файл "AVR309.inf" создан с помощью текстового редактора. Он запрашивается операционной системой в процессе инсталляции.

После инициации инсталляции файл драйвера копируется в операционную систему, а затем выполняются требуемые системные изменения. INF-файл гарантирует инсталляцию DLL-библиотеки в системную папку, что гарантирует

простоту ее вызова из различных приложений. Для инсталляции устройства необходимо три файла: INF-файл "AVR309.inf", драйвер "AVR309.sys" и DLL-библиотека "AVR309.dll".

Библиотека DLL

DLL-библиотека связывает с драйвером устройства и всеми функциями устройства, реализованных в этой библиотеке. Способ написания программы конечного пользователя предельно упрощен. DLL-библиотека гарантирует привилегированный доступ к устройству (упорядочивает доступ к устройству), содержит системный буфер для приема данных через RS232 и создает одну системную задачу для чтения буфера данных RS232.

Упорядочение в DLL гарантирует, что только одно приложение/задача будет связано с устройством в данное время. Это необходимо ввиду возможности наложения запросов/ответов от различных приложений в одно и то же время.

Системный буфер для приема данных через RS232 гарантирует, что принятые данные будут размещены в одном буфере, который является общим для всех приложений. В таком случае принятые данные устройством будут отправляться всем приложениям. Не стоит беспокоиться о том, что приложение примет не полностью данные из-за того, что другое приложение до этого считывало буфер данных.

Для всех приложений существует только одна системная задача, и данные будут периодически запрашиваться у устройства через RS232. В этом случае задача будет сохранять принятые данные в системный буфер. Наличие только одного системного буфера гарантирует небольшую загрузку ЦПУ (в сравнении, когда каждое приложение имеет свою собственную задачу) и упрощает сохранение данных в системный буфер. Все функции устройства определены в библиотеке DLL, а их экспорт выполняется в удобной форме: не в виде номера функции и параметров, а в виде удобного имени функции с параметрами. Некоторые функции более сложны внутренне, например, функция чтения буфера данных RS232. Таким образом, разработчики могут быстро разрабатывать свои приложения, используя только интерфейс DLL. Нет необходимости изучать функции низкого уровня устройства, т.к. DLL-библиотека разделяет уровень прикладного программирования от аппаратного уровня.

Объявления написаны для 3 наиболее популярных языков программирования: Borland Delphi, C++ (Borland или Microsoft) и Visual Basic.

Приложение конечного пользователя

Приложение конечного пользователя использует функции из DLL-библиотеки для связи с устройством. Его основным назначением является реализация дружественного графического пользовательского интерфейса (GUI).

Для написания собственной программы программистам необходимо использовать библиотеку DLL.

2.12. Погрешность генерации скорости УАПП.

Микроконтроллер использует тактовую частоту 12МГц из-за осуществления преобразования USB. Однако недостатком использования такой частоты синхронизации является наличие небольшой погрешности при генерации стандартных скоростей связи. Однако за счет высокого значения частоты синхронизации погрешность минимальна. Максимально допустимая погрешность генерации скорости связи должны быть 4%, т.к. максимальная погрешность равна отношению длительности половины бита (0.5) к максимальной длительности пакета 12 бит = 1 старт-бит + 8 бит данных + 1 бит паритета + 2 стоп-бита. Таким образом, погрешность равна $0.5/12 \cdot 100\% = 4.1\%$.

Функции в DLL автоматически вычисляют погрешность и устанавливают такую скорость связи, отклонение от стандартной которой не превышает 4%. В случае определения неподдерживаемой скорости выводится сообщение об ошибке. Однако не рекомендуется использовать скорости с погрешностью более 2%.

В таблице 4 подытоживаются погрешности в генерации стандартных скоростей при работе микроконтроллера на частоте 12 МГц.

Таблица 4. Погрешность генерации скорости связи УАПП микроконтроллера AVR (тактовая частота 12 МГц)

Стандартная скорость	Скорость в AVR	Погрешность, %
600	602	+0.33
1200	1204	+0.33
2400	2408	+0.33
4800	4808	+0.17
9600	9616	+0.17
19200	19230	+0.16
28800	28846	+0.16
38400	38462	+0.16
57600	57692	+0.16
115200	115384	+0.16

3. СИСТЕМНЫЕ ТРЕБОВАНИЯ ДЛЯ РАБОТЫ СРЕДЫ.

3.1. Системные требования при работе среды в ОС Linux (U(K)ubuntu.

Минимальные аппаратные требования, которым должна отвечать система для успешного развертывания среды LabVisual при работе под управлением ОС Kubuntu следующие:

Процессор: Pentium IV (Intel Atom 230)1.6 ГГц

L2 Кэш 512 Кб

FSB Speed 533 MHz

Оперативная память: 2048 Мб

Видеокарта: 128 Мб

Монитор: 1024x768, 16 бит

Аудиокарта: Совместимая с DirectX 9.0, имеющая линейный вход Line In, микрофонный вход Mic и линейный выход Line Out

Интерфейсные разъёмы:

4xUSB 2.0

1xPS/2 (клавиатура)

1xPS/2 (мышь)

1xVGA

1xLAN (RJ45)

Жесткий диск: 80 Гб

Необходимые системные требования:

Ядро Linux Kernel версии 2.6.35,

Окружение рабочего стола: GNOME 2.32 либо KDE4,

Основное программное обеспечение: Mozilla Firefox 3.6.10, OpenOffice.org 3.2.1, GCC 4.4.4, VirtualBox 4.0, Krusader, Konsole, должны быть установлены заголовки используемой версии ядра `linux-headers-`uname -r``, пакет разработчика `build-essential`.

Свободное место на жестком диске: 10 Гб

3.2. Системные требования при работе среды в ОС Windows.

Минимальные аппаратные требования, которым должна отвечать система для успешного развертывания среды LabVisual при работе под управлением ОС Windows следующие:

Процессор: Pentium IV (Intel Atom 230)1.6 ГГц

L2 Кэш 512 Кб

FSB Speed 533 MHz

Оперативная память: 2048 Мб

Видеокарта: 128 Мб

Монитор: 1024x768, 16 бит

Аудиокарта: Совместимая с DirectX 9.0, имеющая линейный вход Line In, микрофонный вход Mic и линейный выход Line Out

Интерфейсные разъёмы:

4xUSB 2.0

1xPS/2 (клавиатура)

1xPS/2 (мышь)

1xVGA

1xLAN (RJ45)

Жесткий диск: 80 Гб

Необходимые системные требования:

ОС не ниже Windows XP SP2 (Windows NT 5.1.2600),

Основное программное обеспечение: Mozilla Firefox 3.6.10, OpenOffice.org

3.2.1, Total Commander, драйвер USB LabVisual для работы ПЭВМ с учебными установками.

Свободное место на жестком диске: 2 Гб

4. ЛИЦЕНЗИЯ НА ИСПОЛЬЗОВАНИЕ СРЕДЫ АВТОМАТИЗИРОВАННОГО ЛАБОРАТОРНОГО ПРАКТИКУМА LabVisual.

Автоматизированная среда лабораторного практикума LabVisual

© НПО Учебной Техники «ТулаНаучПрибор», РФ, г. Тула

Данная программа является свободным программным обеспечением. Вы вправе распространять ее и/или модифицировать в соответствии с условиями версии 2 либо по вашему выбору с условиями более поздней версии Стандартной Общественной Лицензии GNU, опубликованной Free Software Foundation.

Мы распространяем данную программу в надежде на то, что она будет вам полезной, однако НЕ ПРЕДОСТАВЛЯЕМ НА НЕЕ НИКАКИХ ГАРАНТИЙ, в том числе ГАРАНТИИ ТОВАРНОГО СОСТОЯНИЯ ПРИ ПРОДАЖЕ и ПРИГОДНОСТИ ДЛЯ ИСПОЛЬЗОВАНИЯ В КОНКРЕТНЫХ ЦЕЛЯХ.

Однако, если среда LabVisual поставляется непосредственно с учебными установками нашего производства, то условия гарантии определяются заключенным договором на поставку партии продукции и прописываются в техническом паспорте на продукцию.

Для получения более подробной информации ознакомьтесь со Стандартной Общественной Лицензией GNU.

Вместе с данной программой вы должны были получить экземпляр Стандартной Общественной Лицензии GNU. Если вы его не получили, сообщите об этом в Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Среда LabVisual распространяется БЕЗ ВСЯКИХ ГАРАНТИЙ. Данная программа является свободным программным обеспечением и вы можете распространять ее в соответствии с условиями Стандартной Общественной Лицензии GNU.

НПО Учебной Техники «ТулаНаучПрибор» настоящим отказывается от всех исключительных прав на использование программы для ЭВМ "LabVisual", авторами которой являются коллектив НПО «ТулаНаучПрибор» и передает все права на использование указанной программы организации, учебному заведению либо физическому лицу, получившему копию среды LabVisual любым способом (в комплекте с учебными приборами, загрузив ПО LabVisual из сети, получив ПО LabVisual на диске от друзей и т. п.).

5. ЛИЦЕНЗИЯ GNU GENERAL PUBLIC LICENSE VERSION 2

Настоящий перевод Стандартной Общественной Лицензии GNU на русский язык не является официальным. Он не публикуется Free Software Foundation и не устанавливает имеющих юридическую силу условий для распространения программного обеспечения, которое распространяется на условиях Стандартной Общественной Лицензии GNU. Условия, имеющие юридическую силу, закреплены исключительно в аутентичном тексте Стандартной Общественной Лицензии GNU на английском языке. Я надеюсь, что настоящий перевод поможет русскоязычным пользователям лучше понять содержание Стандартной Общественной Лицензии GNU. Текст GNU GPL на английском языке вы можете прочитать [здесь](http://www.gnu.org/copyleft/gpl.html)

GNU GENERAL PUBLIC LICENSE

Версия 2, июнь 1991г.

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Каждый вправе копировать и распространять экземпляры настоящей Лицензии без внесения изменений в ее текст.

Преамбула

Большинство лицензий на программное обеспечение лишает вас права распространять и вносить изменения в это программное обеспечение. Стандартная Общественная Лицензия GNU, напротив, разработана с целью гарантировать вам право совместно использовать и вносить изменения в свободное программное обеспечение, т.е. обеспечить свободный доступ к программному обеспечению для всех пользователей. Условия настоящей Стандартной Общественной Лицензии применяются к большей части программного обеспечения Free Software Foundation, а также к любому другому программному обеспечению по желанию его автора. (К некоторому программному обеспечению Free Software Foundation применяются условия Стандартной Общественной Лицензии GNU для Библиотек). Вы также можете применять Стандартную Общественную Лицензию к разработанному вами программному обеспечению.

Говоря о свободном программном обеспечении, мы имеем в виду свободу, а не безвозмездность. Настоящая Стандартная Общественная Лицензия разработана с целью гарантировать вам право распространять экземпляры свободного программного обеспечения (и при желании получать за это вознаграждение), право получать исходный текст программного обеспечения или иметь возможность его получить, право вносить изменения в программное обеспечение или использовать его части в новом свободном программном обеспечении, а также право знать, что вы имеете все вышеперечисленные права.

Чтобы защитить ваши права, мы вводим ряд ограничений с тем, чтобы никто не имел возможности лишить вас этих прав или обратиться к вам с

предложением отказаться от этих прав. Данные ограничения налагают на вас определенные обязанности в случае, если вы распространяете экземпляры программного обеспечения или модифицируете программное обеспечение.

Например, если вы распространяете экземпляры такого программного обеспечения за плату или бесплатно, вы обязаны передать новым обладателям все права в том же объеме, в каком они принадлежат вам. Вы обязаны обеспечить получение новыми обладателями программы ее исходного текста или возможность его получить. Вы также обязаны ознакомить их с условиями настоящей Лицензии.

Для защиты ваших прав мы: (1) оставляем за собой авторские права на программное обеспечение и (2) предлагаем вам использовать настоящую Лицензию, в соответствии с условиями которой вы вправе воспроизводить, распространять и/или модифицировать программное обеспечение.

Кроме того, для защиты как нашей репутации, так и репутации других авторов программного обеспечения, мы уведомляем всех пользователей, что на данное программное обеспечение никаких гарантий не предоставляется. Те, кто приобрел программное обеспечение, с внесенными в него третьими лицами изменениями, должны знать, что они получают не оригинал, в силу чего автор оригинала не несет ответственности за ошибки в работе программного обеспечения, допущенные третьими лицами при внесении изменений.

Наконец, программное обеспечение перестает быть свободным в случае, если лицо приобретает на него исключительные права [1]. Недопустимо, чтобы лица, распространяющие свободное программное обеспечение, могли приобрести исключительные права на использование данного программного обеспечения и зарегистрировать их в Патентном ведомстве. Чтобы избежать этого, мы заявляем, что обладатель исключительных прав обязан предоставить любому лицу права на использование программного обеспечения либо не приобретать исключительных прав вообще.

Ниже изложены условия воспроизведения, распространения и модификации программного обеспечения.

Условия воспроизведения, распространения и модификации

0. Условия настоящей Лицензии применяются ко всем видам программного обеспечения или любому иному произведению, которое содержит указание правообладателя на то, что данное произведение может распространяться на условиях Стандартной Общественной Лицензии. Под термином "Программа" далее понимается любое подобное программное обеспечение или иное произведение. Под термином "произведение, производное от Программы" понимается Программа или любое иное производное произведение в соответствии с законодательством об авторском праве [2], т.е. произведение, включающее в себя Программу или ее часть, как с внесенными в ее текст изменениями, так и без них и/или переведенную на другой язык. (Здесь и далее, понятие "модификация" включает в себя понятие перевода в самом широком смысле). Каждый приобретатель экземпляра Программы именуется в

дальнейшем "Лицензиат".

Действие настоящей Лицензии не распространяется на осуществление иных прав, кроме воспроизведения, распространения и модификации программного обеспечения. Не устанавливаются ограничения на запуск Программы. Условия Лицензии распространяются на выходные данные из Программы только в том случае, если их содержание составляет произведение, производное от Программы (независимо от того, было ли такое произведение создано в результате запуска Программы). Это зависит от того, какие функции выполняет Программа.

1. Лицензиат вправе изготавливать и распространять экземпляры исходного текста Программы в том виде, в каком он его получил, без внесения в него изменений на любом носителе, при соблюдении следующих условий: на каждом экземпляре помещен знак охраны авторского права и уведомление об отсутствии гарантий; оставлены без изменений все уведомления, относящиеся к настоящей Лицензии и отсутствию гарантий; вместе с экземпляром Программы приобретателю передается копия настоящей Лицензии. Лицензиат вправе взимать плату за передачу экземпляра Программы, а также вправе за плату оказывать услуги по гарантийной поддержке Программы.

2. Лицензиат вправе модифицировать свой экземпляр или экземпляры Программы полностью или любую ее часть. Данные действия Лицензиата влекут за собой создание произведения, производного от Программы. Лицензиат вправе изготавливать и распространять экземпляры такого произведения, производного от Программы, или собственно экземпляры изменений в соответствии с пунктом 1 настоящей Лицензии при соблюдении следующих условий:

а) файлы, измененные Лицензиатом, должны содержать хорошо заметную пометку, что они были изменены, а также дату внесения изменений;

б) при распространении или публикации Лицензиатом любого произведения, которое содержит Программу или ее часть или является производным от Программы или от ее части, Лицензиат обязан передавать права на использование данного произведения третьим лицам на условиях настоящей Лицензии, при этом Лицензиат не вправе требовать уплаты каких-либо лицензионных платежей. Распространяемое произведение лицензируется как одно целое;

с) если модифицированная Программа при запуске обычно читает команды в интерактивном режиме, Лицензиат обязан обеспечить вывод на экран дисплея или печатающее устройство сообщения, которое должно включать в себя:

знак охраны авторского права;

уведомление об отсутствии гарантий на Программу (или иное, если Лицензиат предоставляет гарантии);

указание на то, что пользователи вправе распространять экземпляры Программы в соответствии с условиями настоящей Лицензии, а также на то, каким образом пользователь может ознакомиться с текстом настоящей

Лицензии. (Исключение: если оригинальная Программа является интерактивной, но не выводит в своем обычном режиме работы сообщение такого рода, то вывод подобного сообщения произведением, производным от Программы, в этом случае не обязателен). Вышеуказанные условия применяются к модифицированному произведению, производному от Программы, в целом. В случае если отдельные части данного произведения не являются производными от Программы, являются результатом творческой деятельности и могут быть использованы как самостоятельное произведение, Лицензиат вправе распространять отдельно такое произведение на иных лицензионных условиях. В случае если Лицензиат распространяет вышеуказанные части в составе произведения, производного от Программы, то условия настоящей Лицензии применяются к произведению в целом, при этом права, приобретаемые sublicензиатами на основании Лицензии, передаются им в отношении всего произведения, включая все его части, независимо от того, кто является их авторами.

Целью настоящего пункта 2 не является заявление прав или оспаривание прав на произведение, созданное исключительно Лицензиатом. Целью настоящего пункта является обеспечение права контролировать распространение произведений, производных от Программы, и составных произведений, производных от Программы. Размещение произведения, которое не является производным от Программы, на одном устройстве для хранения информации или носителе вместе с Программой или произведением, производным от Программы, не влечет за собой распространения условий настоящей Лицензии на такое произведение.

3. Лицензиат вправе воспроизводить и распространять экземпляры Программы или произведения, которое является производным от Программы, в соответствии с пунктом 2 настоящей Лицензии, в виде объектного кода или в исполняемой форме в соответствии с условиями п.п. 1 и 2 настоящей Лицензии при соблюдении одного из перечисленных ниже условий:

а) к экземпляру должен прилагаться соответствующий полный исходный текст в машиночитаемой форме, который должен распространяться в соответствии с условиями п.п. 1 и 2 настоящей Лицензии на носителе, обычно используемом для передачи программного обеспечения, либо

б) к экземпляру должно прилагаться действительное в течение трех лет предложение в письменной форме к любому третьему лицу передать за плату, не превышающую стоимость осуществления собственно передачи, экземпляр соответствующего полного исходного текста в машиночитаемой форме в соответствии с условиями п.п. 1 и 2 настоящей Лицензии на носителе, обычно используемом для передачи программного обеспечения, либо

с) к экземпляру должна прилагаться полученная Лицензиатом информация о предложении, в соответствии с которым можно получить соответствующий исходный текст. (Данное положение применяется исключительно в том случае, если Лицензиат осуществляет некоммерческое распространение программы,

при этом программа была получена самим Лицензиатом в виде объектного кода или в исполняемой форме и сопровождалась предложением, соответствующим условиям пп.в п.3 настоящей Лицензии).

Под исходным текстом произведения понимается такая форма произведения, которая наиболее удобна для внесения изменений. Под полным исходным текстом исполняемого произведения понимается исходный текст всех составляющих произведение модулей, а также всех файлов, связанных с описанием интерфейса, и сценариев, предназначенных для управления компиляцией и установкой исполняемого произведения. Однако, в качестве особого исключения, распространяемый исходный текст может не включать того, что обычно распространяется (в виде исходного текста или в бинарной форме) с основными компонентами (компилятор, ядро и т.д.) операционной системы, в которой работает исполняемое произведение, за исключением случаев, когда исполняемое произведение сопровождается таким компонентом.

В случае если произведение в виде объектного кода или в исполняемой форме распространяется путем предоставления доступа для копирования его из определенного места, обеспечение равноценного доступа для копирования исходного текста из этого же места удовлетворяет требованиям распространения исходного текста, даже если третьи лица при этом не обязаны копировать исходный текст вместе с объектным кодом произведения.

4. Лицензиат вправе воспроизводить, модифицировать, распространять или передавать права на использование Программы только на условиях настоящей Лицензии. Любое воспроизведение, модификация, распространение или передача прав на иных условиях являются недействительными и автоматически ведут к расторжению настоящей Лицензии и прекращению всех прав Лицензиата, предоставленных ему настоящей Лицензией. При этом права третьих лиц, которым Лицензиат в соответствии с настоящей Лицензией передал экземпляры Программы или права на нее, сохраняются в силе при условии полного соблюдения ими настоящей Лицензии.

5. Лицензиат не обязан присоединяться к настоящей Лицензии, поскольку он ее не подписал. Однако только настоящая Лицензия предоставляет право распространять или модифицировать Программу или произведение, производное от Программы. Подобные действия нарушают действующее законодательство, если они не осуществляются в соответствии с настоящей Лицензией. Если Лицензиат внес изменения или осуществил распространение экземпляров Программы или произведения, производного от Программы, Лицензиат тем самым подтвердил свое присоединение к настоящей Лицензии в целом, включая условия, определяющие порядок воспроизведения, распространения или модификации Программы или произведения, производного от Программы.

6. При распространении экземпляров Программы или произведения, производного от Программы, первоначальный лицензиар автоматически передает приобретателю такого экземпляра право воспроизводить,

распространять и модифицировать Программу в соответствии с условиями настоящей Лицензии. Лицензиат не вправе ограничивать каким-либо способом осуществление приобретателями полученных ими прав. Лицензиат не несет ответственности за несоблюдение условий настоящей Лицензии третьими лицами.

7. Лицензиат не освобождается от исполнения обязательств в соответствии с настоящей Лицензией в случае, если в результате решения суда или заявления о нарушении исключительных прав или в связи с наступлением иных обстоятельств, не связанных непосредственно с нарушением исключительных прав, на Лицензиата на основании решения суда, договора или ином основании возложены обязательства, которые противоречат условиям настоящей Лицензии. В этом случае Лицензиат не вправе распространять экземпляры Программы, если он не может одновременно исполнить условия настоящей Лицензии и возложенные на него указанным выше способом обязательства. Например, если по условиям лицензионного соглашения сублицензиатам не может быть предоставлено право бесплатного распространения экземпляров Программы, которые они приобрели напрямую или через третьих лиц у Лицензиата, то в этом случае Лицензиат обязан отказаться от распространения экземпляров Программы.

Если любое положение настоящего пункта при наступлении конкретных обстоятельств будет признано недействительным или неприменимым, настоящий пункт применяется за исключением такого положения. Настоящий пункт применяется в целом при прекращении вышеуказанных обстоятельств или их отсутствии. Целью данного пункта не является принуждение Лицензиата к нарушению патента или заявления на иные права собственности или к оспариванию действительности такого заявления. Единственной целью данного пункта является защита неприкосновенности системы распространения свободного программного обеспечения, которая обеспечивается за счет общественного лицензирования. Многие люди внесли свой щедрый вклад в создание большого количества программного обеспечения, которое распространяется через данную систему в надежде на ее длительное и последовательное применение. Лицензиат не вправе вынуждать автора распространять программное обеспечение через данную систему. Право выбора системы распространения программного обеспечения принадлежит исключительно его автору.

Настоящий пункт 7 имеет целью четко определить те цели, которые преследуют все остальные положения настоящей Лицензии.

8. В том случае если распространение и/или использование Программы в отдельных государствах ограничено соглашениями в области патентных или авторских прав, первоначальный правообладатель, распространяющий Программу на условиях настоящей Лицензии, вправе ограничить территорию распространения Программы, указав только те государства, на территории которых допускается распространение Программы без ограничений,

обусловленных такими соглашениями. В этом случае такое указание в отношении территорий определенных государств признается одним из условий настоящей Лицензии.

9. Free Software Foundation может публиковать исправленные и/или новые версии настоящей Стандартной Общественной Лицензии. Такие версии могут быть дополнены различными нормами, регулирующими правоотношения, которые возникли после опубликования предыдущих версий, однако в них будут сохранены основные принципы, закрепленные в настоящей версии. Каждой версии присваивается свой собственный номер. Если указано, что Программа распространяется в соответствии с определенной версией, т.е. указан ее номер, или любой более поздней версией настоящей Лицензии, Лицензиат вправе присоединиться к любой из этих версий Лицензии, опубликованных Free Software Foundation. Если Программа не содержит такого указания на номер версии Лицензии Лицензиат вправе присоединиться к любой из версий Лицензии, опубликованных когда-либо Free Software Foundation.

10. В случае если Лицензиат намерен включить часть Программы в другое свободное программное обеспечение, которое распространяется на иных условиях, чем в настоящей Лицензии, ему следует испросить письменное разрешение на это у автора программного обеспечения. Разрешение в отношении программного обеспечения, права на которое принадлежат Free Software Foundation, следует испрашивать у Free Software Foundation. В некоторых случаях Free Software Foundation делает исключения. При принятии решения Free Software Foundation будет руководствоваться двумя целями: сохранение статуса свободного для любого произведения, производного от свободного программного обеспечения Free Software Foundation и обеспечение наиболее широкого совместного использования программного обеспечения.

ОТСУТСТВИЕ ГАРАНТИЙНЫХ ОБЯЗАТЕЛЬСТВ, ЕСЛИ ПО LAVVISUAL ПОСТАВЛЯЕТСЯ НЕ В КОМПЛЕКТЕ С УЧЕБНЫМИ УСТАНОВКАМИ ПРОИЗВОДСТВА НПО «ТУЛАНАУЧПРИБОР».

11. ПОСКОЛЬКУ НАСТОЯЩАЯ ПРОГРАММА РАСПРОСТРАНЯЕТСЯ БЕСПЛАТНО, ГАРАНТИИ НА НЕЕ НЕ ПРЕДОСТАВЛЯЮТСЯ В ТОЙ СТЕПЕНИ, В КАКОЙ ЭТО ДОПУСКАЕТСЯ ПРИМЕНИМЫМ ПРАВОМ. НАСТОЯЩАЯ ПРОГРАММА ПОСТАВЛЯЕТСЯ НА УСЛОВИЯХ "КАК ЕСТЬ". ЕСЛИ ИНОЕ НЕ УКАЗАНО В ДОГОВОРЕ В ПИСЬМЕННОЙ ФОРМЕ, АВТОР И/ИЛИ ИНОЙ ПРАВООБЛАДАТЕЛЬ НЕ ПРИНИМАЕТ НА СЕБЯ НИКАКИХ ГАРАНТИЙНЫХ ОБЯЗАТЕЛЬСТВ, КАК ЯВНО ВЫРАЖЕННЫХ, ТАК И ПОДРАЗУМЕВАЕМЫХ, В ОТНОШЕНИИ ПРОГРАММЫ, В ТОМ ЧИСЛЕ ПОДРАЗУМЕВАЕМУЮ ГАРАНТИЮ ТОВАРНОГО СОСТОЯНИЯ ПРИ ПРОДАЖЕ И ПРИГОДНОСТИ ДЛЯ ИСПОЛЬЗОВАНИЯ В КОНКРЕТНЫХ ЦЕЛЯХ, А ТАКЖЕ ЛЮБЫЕ ИНЫЕ ГАРАНТИИ. ВСЕ РИСКИ, СВЯЗАННЫЕ С КАЧЕСТВОМ И ПРОИЗВОДИТЕЛЬНОСТЬЮ ПРОГРАММЫ, НЕСЕТ ЛИЦЕНЗИАТ. В СЛУЧАЕ ЕСЛИ В ПРОГРАММЕ БУДУТ ОБНАРУЖЕНЫ НЕДОСТАТКИ, ВСЕ

РАСХОДЫ, СВЯЗАННЫЕ С ТЕХНИЧЕСКИМ ОБСЛУЖИВАНИЕМ, РЕМОНТОМ ИЛИ ИСПРАВЛЕНИЕМ ПРОГРАММЫ, НЕСЕТ ЛИЦЕНЗИАТ.

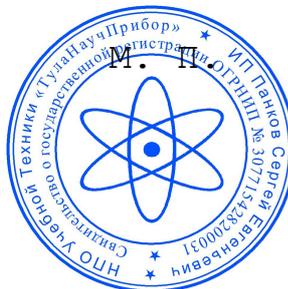
12. ЕСЛИ ИНОЕ НЕ ПРЕДУСМОТРЕНО ПРИМЕНЯЕМЫМ ПРАВОМ ИЛИ НЕ СОГЛАСОВАНО СТОРОНАМИ В ДОГОВОРЕ В ПИСЬМЕННОЙ ФОРМЕ, АВТОР И/ИЛИ ИНОЙ ПРАВООБЛАДАТЕЛЬ, КОТОРЫЙ МОДИФИЦИРУЕТ И/ИЛИ РАСПРОСТРАНЯЕТ ПРОГРАММУ НА УСЛОВИЯХ НАСТОЯЩЕЙ ЛИЦЕНЗИИ, НЕ НЕСЕТ ОТВЕТСТВЕННОСТИ ПЕРЕД ЛИЦЕНЗИАТОМ ЗА УБЫТКИ, ВКЛЮЧАЯ ОБЩИЕ, РЕАЛЬНЫЕ, ПРЕДВИДИМЫЕ И КОСВЕННЫЕ УБЫТКИ (В ТОМ ЧИСЛЕ УТРАТУ ИЛИ ИСКАЖЕНИЕ ИНФОРМАЦИИ, УБЫТКИ, ПОНЕСЕННЫЕ ЛИЦЕНЗИАТОМ ИЛИ ТРЕТЬИМИ ЛИЦАМИ, НЕВОЗМОЖНОСТЬ РАБОТЫ ПРОГРАММЫ С ЛЮБОЙ ДРУГОЙ ПРОГРАММОЙ И ИНЫЕ УБЫТКИ). АВТОР И/ИЛИ ИНОЙ ПРАВООБЛАДАТЕЛЬ В СООТВЕТСТВИИ С НАСТОЯЩИМ ПУНКТОМ НЕ НЕСУТ ОТВЕТСТВЕННОСТИ ДАЖЕ В ТОМ СЛУЧАЕ, ЕСЛИ ОНИ БЫЛИ ПРЕДУПРЕЖДЕНЫ О ВОЗМОЖНОСТИ ВОЗНИКНОВЕНИЯ ТАКИХ УБЫТКОВ.

Примечание: Обращаем внимание, что пп. 11 — 12 настоящей лицензии не распространяются на ПО для ПЭВМ LabVisual, если указанное ПО поставляется в комплекте с учебными установками производства НПО Учебной Техники «ТулаНаучПрибор». Т. о., если ПО LabVisual получено в комплекте с учебными приборами, то гарантийные обязательства предоставляются согласно соглашениям, прописанным в договоре на поставку продукции либо на разработку приборов. Гарантийные обязательства также прописаны в технических паспортах на каждую единицу продукции.



Генеральный директор НПО «ТулаНаучПрибор» _____

/Панков С. Е./



СОДЕРЖАНИЕ.

1. ПРОГРАММНАЯ ЧАСТЬ.	
1.1. Свободное программное обеспечение.....с. 2	2
1.2. Свободные операционные системы для персональных компьютеров на базе ядра GNU/Linux.с. 3	3
1.3. ОС Ubuntu (Kubuntu) на базе ядра GNU/Linux.....с. 5	5
1.4. Установка ОС Kubuntu на ПК.....с. 8	8
1.5. Kubuntu ОС. Краткое руководство пользователя.....с. 16	16
1.6. Установка и настройка среды LabVisual в ОС Kubuntu Linux. Эмулятор программного и аппаратного кода VirtualBox.....с. 26	26
1.7. Работа среды LabVisual в ОС Kubuntu Linux.....с. 35	35
1.8. Установка и настройка среды LabVisual в ОС Windows.....с. 41	41
1.9. Работа среды LabVisual в ОС Windows.....с. 45	45
2. АППАРАТНАЯ ЧАСТЬ. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ УНИВЕРСАЛЬНОЙ ПОСЛЕДОВАТЕЛЬНОЙ ШИНЫ (USB)	
2.1. Введение.....с. 47	47
2.2. Принцип действия.....с. 48	48
2.3. Аппаратная реализация.....с. 52	52
2.4. Программная реализация.....с. 55	55
2.5. Описание программы микроконтроллера.....с. 56	56
2.6. Стандартные функции USB (стандартные запросы).....с. 61	61
2.7. Функции USB-производителя (запросы производителя).....с. 61	61
2.8. Структуры данных (USB-дискрипторы и строки).....с. 62	62
2.9. Формат входного сообщения компьютера.....с. 62	62
2.10. Создание собственной программы.....с. 64	64
2.11. Программа для ПК.....с. 65	65
2.12. Погрешность генерации скорости УАПП.....с. 67	67
3. СИСТЕМНЫЕ ТРЕБОВАНИЯ ДЛЯ РАБОТЫ СРЕДЫ.	
3.1. Системные требования при работе среды в ОС Linux (U(K)ubuntu.....с. 68	68
3.2. Системные требования при работе среды в ОС Windows.....с. 69	69
4. ЛИЦЕНЗИЯ НА ИСПОЛЬЗОВАНИЕ СРЕДЫ АВТОМАТИЗИРОВАННОГО ЛАБОРАТОРНОГО ПРАКТИКУМА LabVisal.....с. 70	70
5. ЛИЦЕНЗИЯ GNU GENERAL PUBLIC LICENSE VERSION 2....с. 71	71

**ДЛЯ СВОБОДНОГО РАСПРОСТРАНЕНИЯ
НПО УЧЕБНОЙ ТЕХНИКИ «ТУЛАНАУЧПРИБОР»**