

В.Н. Гололобов

Эссе. Proteus в Linux и в Windows 8.1



Москва - 2014

Оглавление

Proteus 7	3
Аналоговые схемы.....	3
Цифровые схемы	8
Другие схемы	10
Резюме	11
Proteus 8 первая часть.....	13
Proteus 8 продолжение.....	22
Резюме	31

Proteus 7

Аналоговые схемы

Как и многие программы, ISIS после установки создаёт папку с примерами. Об обилии примеров вы можете судить по рисунку ниже:

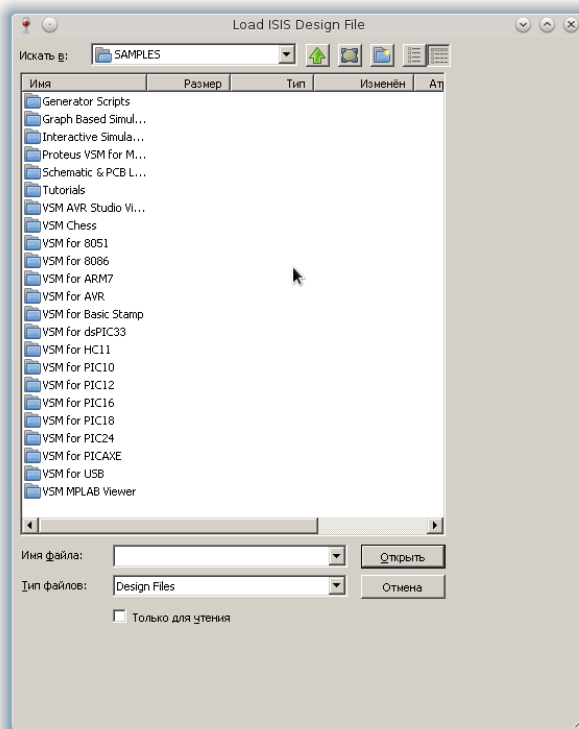


Рис. 1.1. Папки с примерами из Proteus

В части аналоговых устройств вы можете убедиться, что программа позволяет выполнить все виды моделирования, присущие другим программам такого плана: посмотреть сигнал на выходе схемы, построить частотные характеристики, проверить поведение устройства при изменении каких-то параметров.

Для работы с аналоговыми устройствами программа предлагает разные инструменты. Хорошо развитые графические средства дают полное представление о свойствах, например, усилителя, которые позже вы можете сравнить с реальными данными, производя «натурные» испытания.

Если вам привычнее проверять вид сигнала с помощью виртуального осциллографа, вы можете использовать четырёхканальный осциллограф из набора виртуальных приборов ISIS.

Однако удобнее, всё-таки работать с графиками. Это положение дел соответствует использованию современных записывающих осциллографов, когда можно наблюдать события реального времени, но осознать происходящее помогает запись данных.

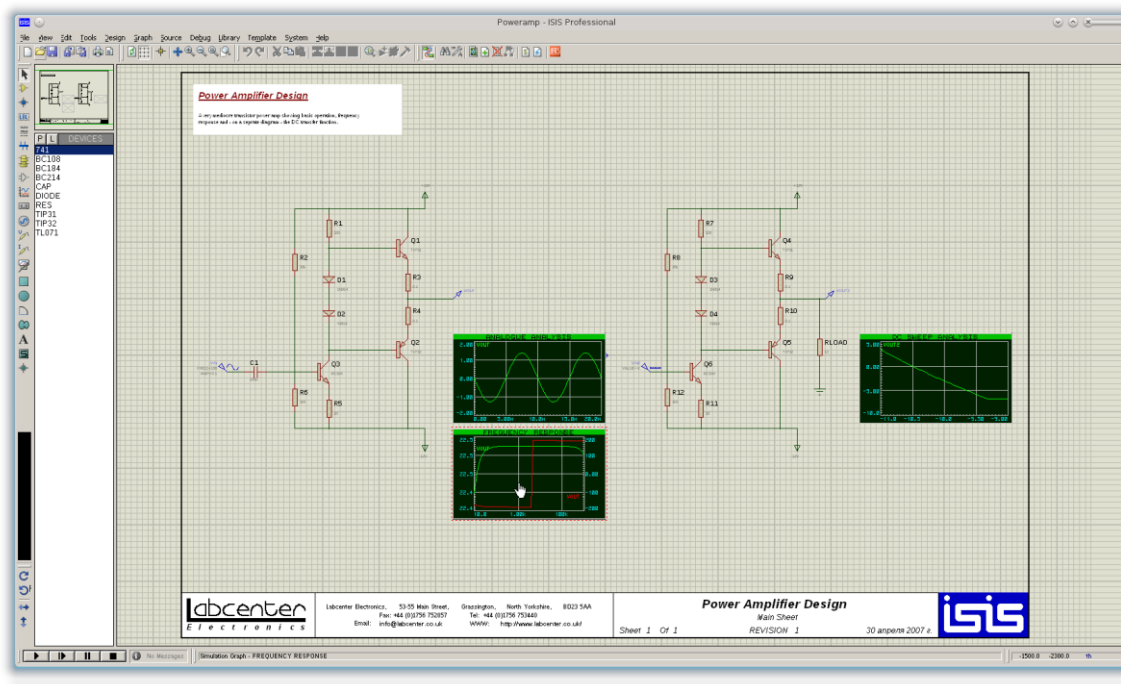


Рис. 1.2. Моделирование выходного каскада

Многие начинающие любители считают, что схема, содержащая большое количество элементов — это настоящая схема, а другие схемы «для маленьких». Но вспомните, некогда усилитель имел сотни элементов и в графическом, и в реальном представлении. Сегодня его заменяет одна микросхема, изображённая в графическом виде как квадратик, а в физическом виде представляющая почти такой же квадратик из пластика.

Я считаю, что начинающим любителям стоит больше озаботиться не сложностью схем, а ясным пониманием работы самых простых электрических цепей. Вот один из примеров, которые я имею в виду:

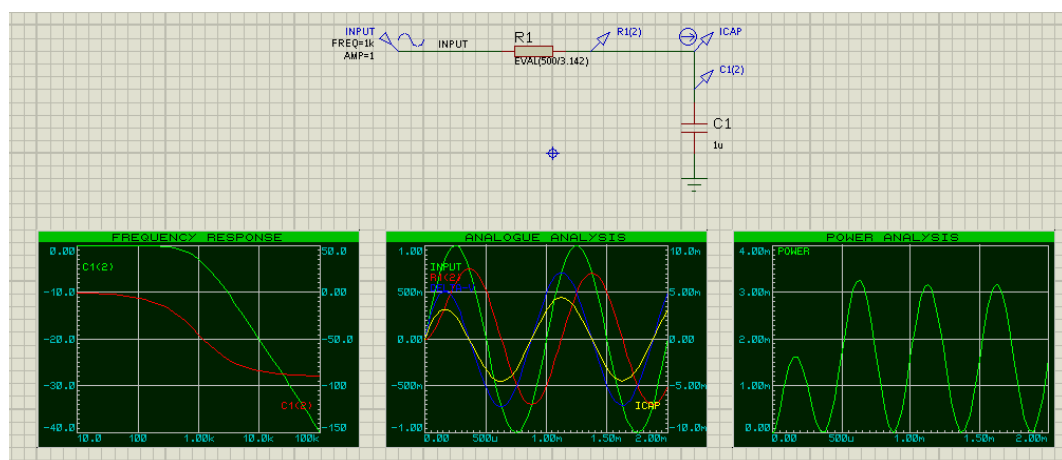


Рис. 1.3. Исследование RC цепи в ISIS

Приведённый пример ярко показывает те большие возможности, что заложены в программе для любознательного человека, которого интересует электроника.

И ещё один пример, который я сам люблю приводить, если заходит разговор о нелинейных искажениях.

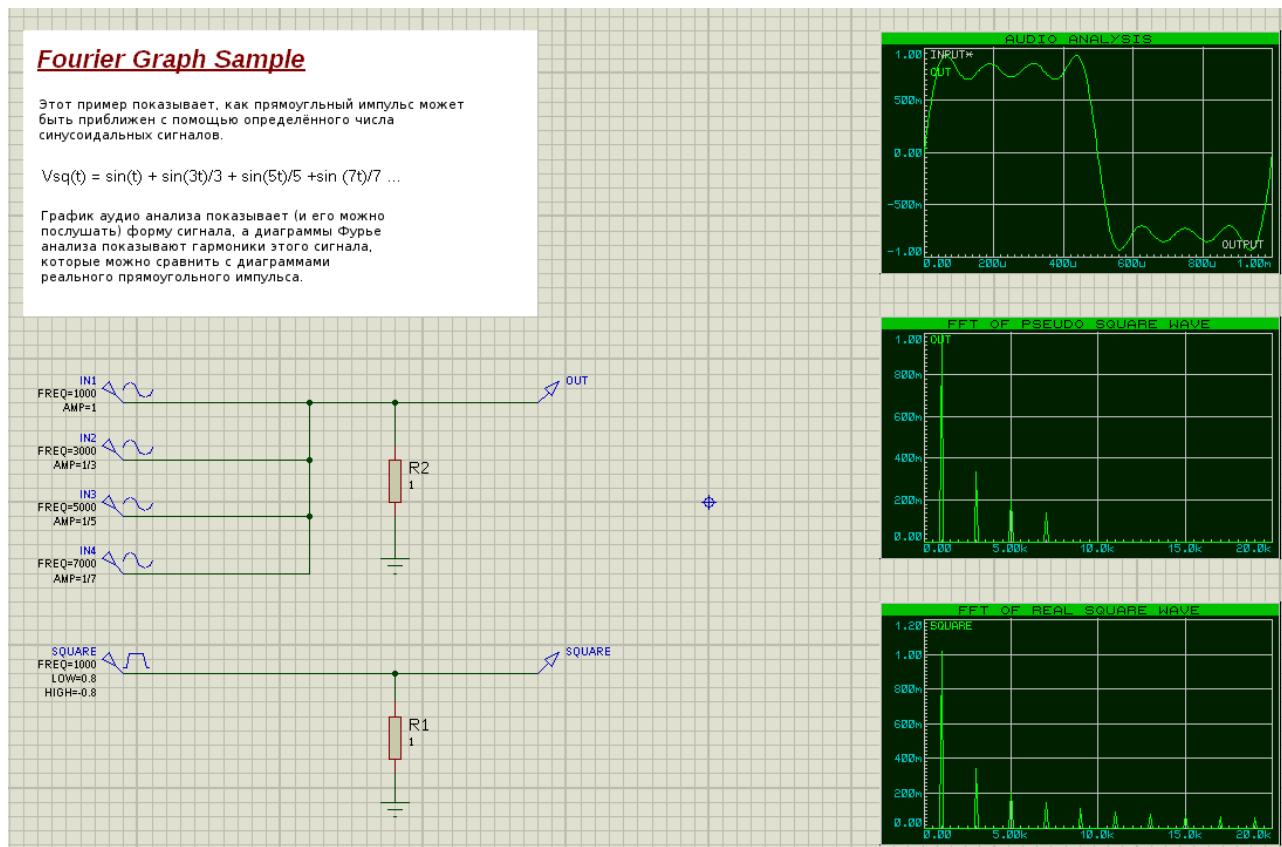


Рис. 1.4. Фурье анализ в примере из папки Samples

Я перевёл примечание к этому примеру, поскольку и для меня математическая очевидность преобразований Фурье в годы учёбы плохо сочеталась позже с электроникой, когда пришлось работать с реальными устройствами.

Как вы могли заметить, последние два примера почти не содержат привычных элементов любой электрической схемы; нет ни транзисторов, ни микросхем, но как много интересного можно узнать, экспериментируя в программе Proteus с такими простыми компонентами. Более того, вы научитесь работать с графическими инструментами исследования схем, которые тоже требуют ознакомления. Попробуя применить эти инструменты к сложной схеме, вы можете столкнуться с ошибкой в изображении схемы, которую трудно увидеть сразу, а отнести отрицательный опыт к плохой работе программы. Такое тоже бывает не так уж и редко.

Программа Proteus подобно другим программам такого рода вполне может применяться и в производственных условиях, и в условиях учебного процесса. В последнем случае возможности программы позволяют применить её даже в школе на уроках физики. Этому свидетельство примеры анимации, которые вы найдёте в той же папке Samples.

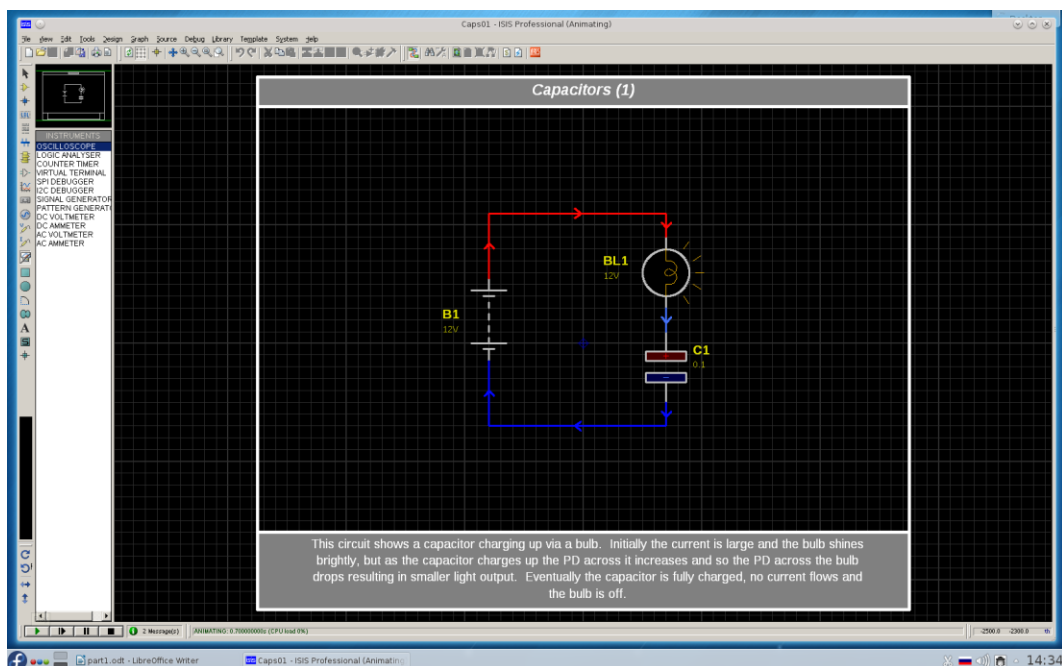


Рис. 1.5. Пример иллюстративной анимации электрической цепи

В программу Proteus включено много современных компонентов, в чём можно убедиться, открыв менеджер элементов схемы. Конечно, следует проверить, будет ли работать каждый из этих элементов, но перечень, конечно, впечатляет.

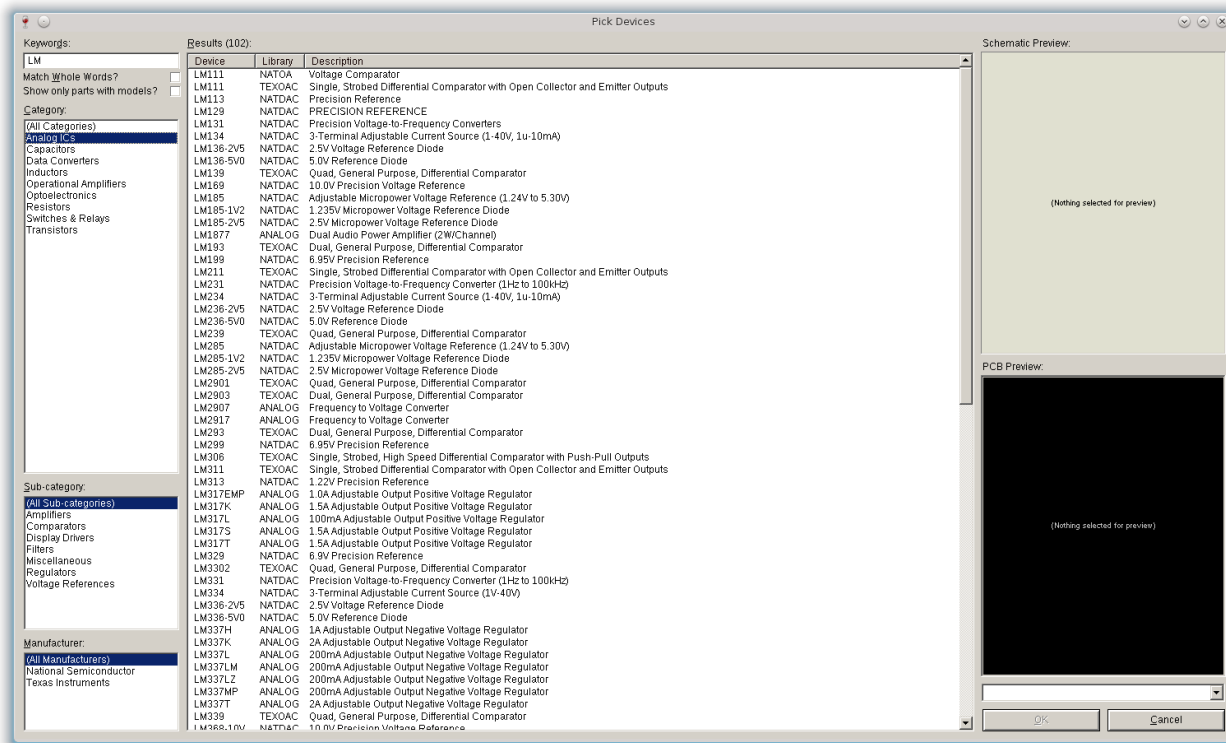


Рис. 1.6. Перечень доступных элементов для добавления в рабочее поле программы

Кроме того, вы можете самостоятельно, это нужно уметь делать, добавить нужный элемент, имеющий Spice-модель. О чём свидетельствует один из примеров.

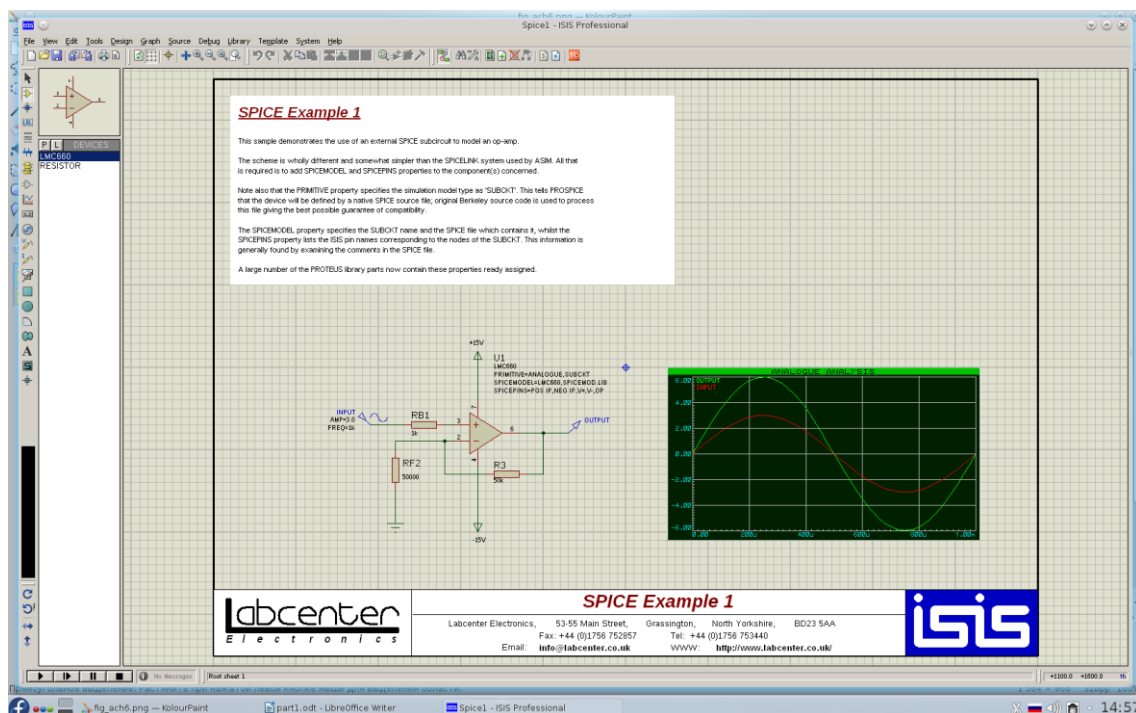


Рис. 1.7. Использование Spice-модели

Со многими активными компонентами электрической схемы в программе очень удобно работать, что особенно привлекательно для начинающих. Так переключателем можно управлять с помощью мышки: достаточно подвести курсор, он изменит свой вид, и щёлкнуть левой клавишей мышки.

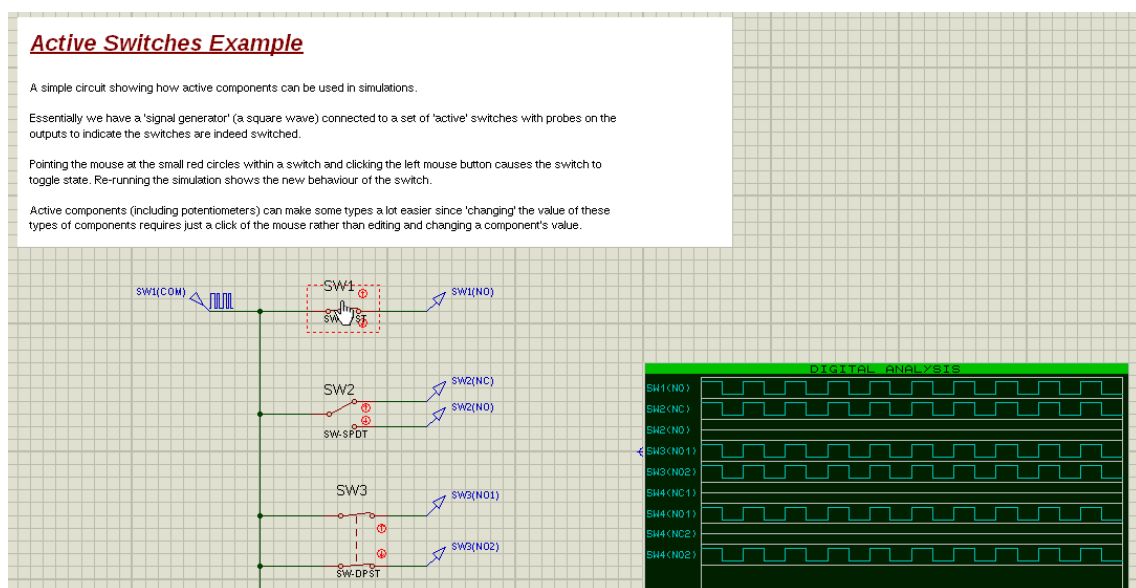


Рис. 1.8. Управление переключателем с помощью мышки

Цифровые схемы

Помимо аналоговых устройств: усилителей, фильтров, стабилизаторов блоков питания, - вы можете в программе ISIS работать и с цифровыми схемами. Микросхемы таких популярных серий как 74 и 4000 представлены достаточно полно.

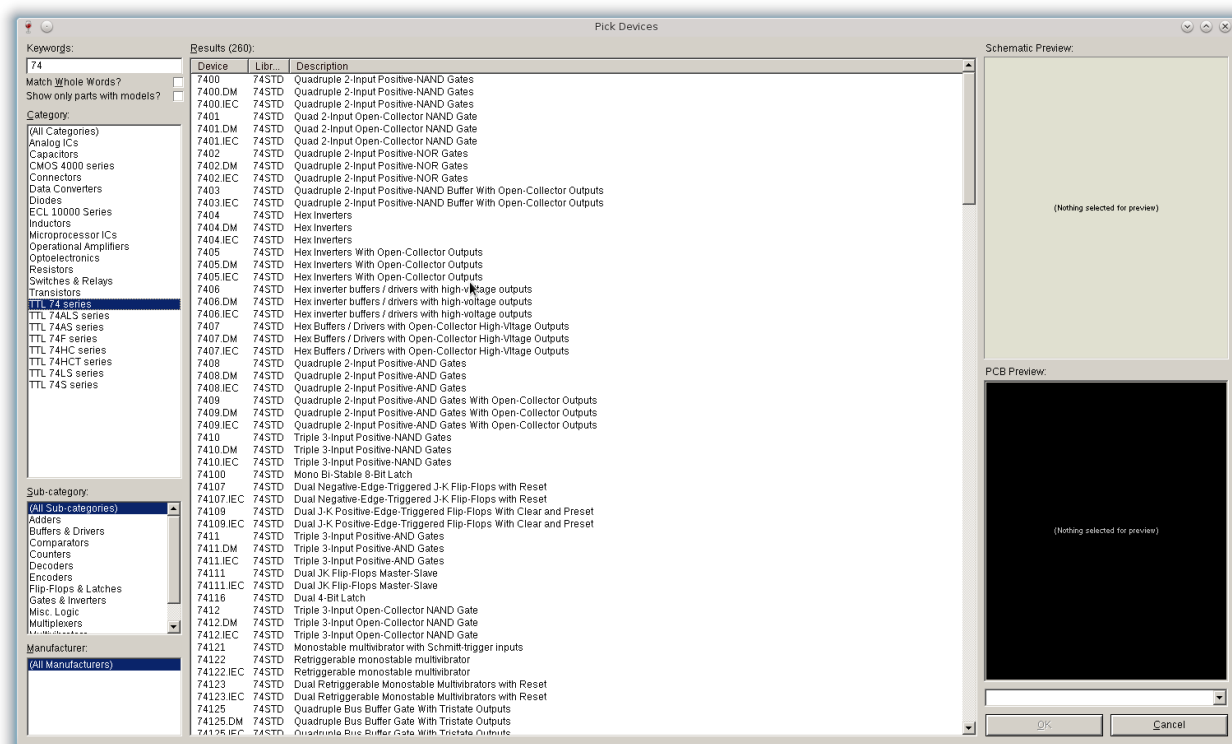


Рис. 1.9. Набор микросхем TTL серии 74

Для графического отображения результатов эксперимента служат цифровые инструменты, поскольку диаграммы для цифровых устройств могут иметь свои особенности.

Как правило, все сигналы в цифровых устройствах имеют фиксированную амплитуду, которая зависит от типа используемых цифровых микросхем, а наибольший интерес представляют временные соотношения между импульсами. Поэтому моделирование работы цифровых устройств, конечно, имеет отличие от моделирования аналоговых схем.

И вновь повторяюсь, ясное понимание работы простых цифровых схем важнее слепого повторения даже самых сложных из них. Потратив больше времени на эксперименты с небольшим набором элементов, вы сэкономите его тогда, когда будете работать с более сложными устройствами и приборами.

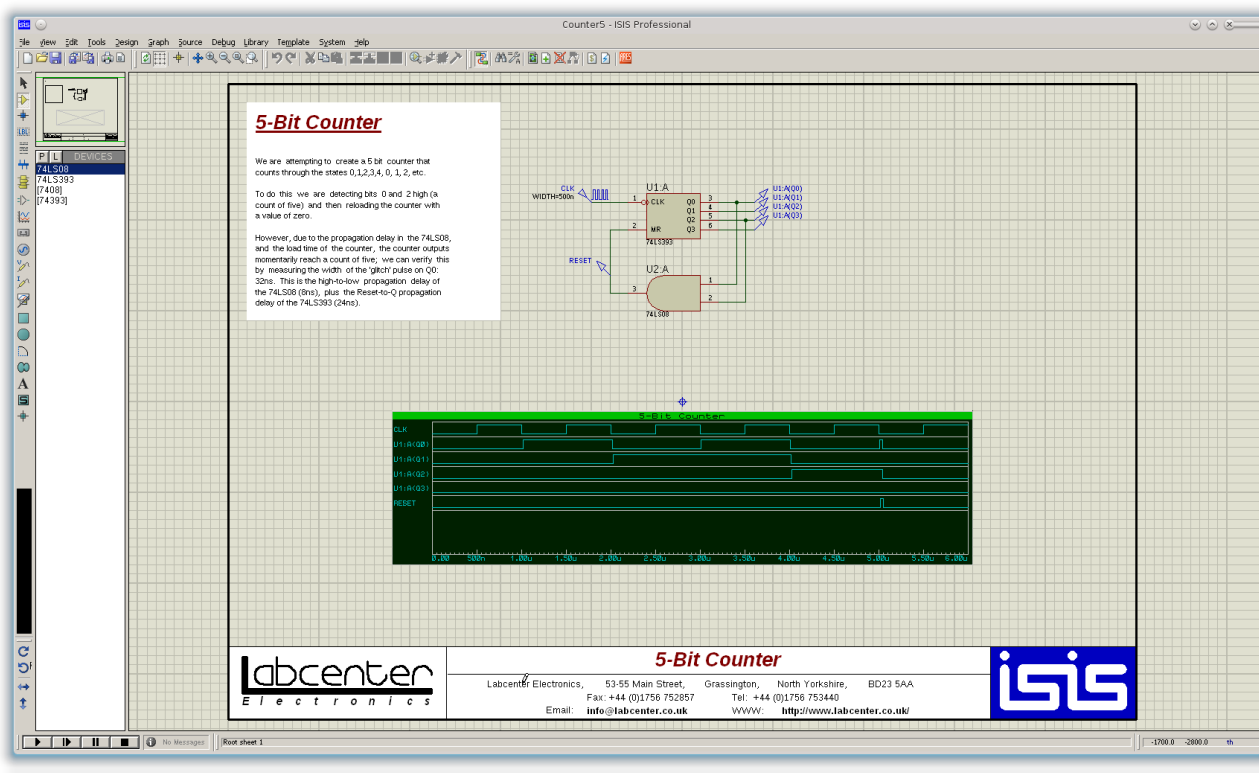


Рис. 1.10. Один из примеров цифровой схемы

Для описания работы цифровых схем при моделировании часто используют специальные языки, например, VHDL. Программа ISIS позволяет сделать модификацию, используя эти средства.

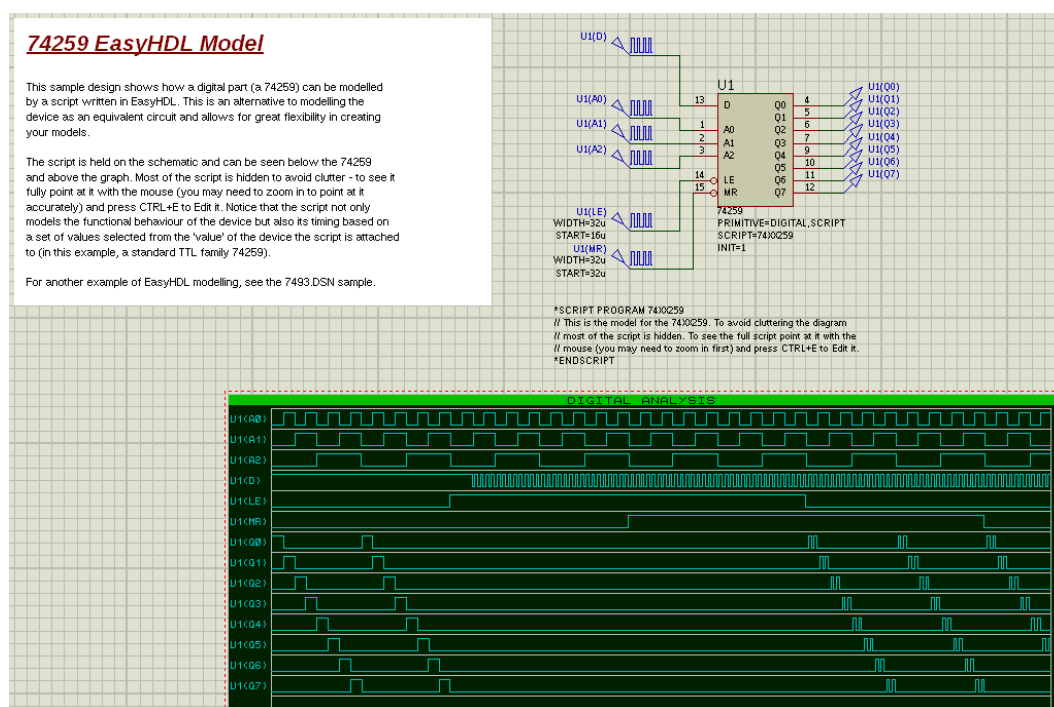


Рис. 1.11. Пример применения описания VHDL

Конечно, эта возможность может не интересовать начинающих, но не всегда же им оставаться начинающими!

Другие схемы

Среди примеров, находящихся в папке Samples, я отвёл бы особое место примерам на базе таймера 555. Некогда вызвавшая всеобщее восхищение эта микросхема и по сей день не потеряла своего значения.

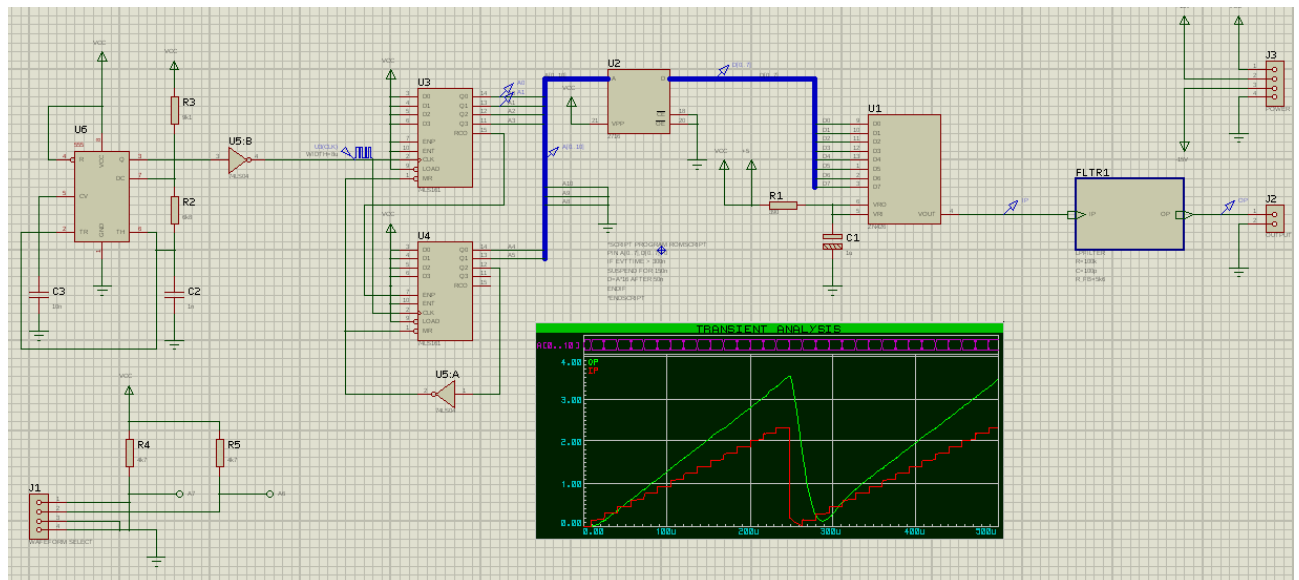


Рис. 1.12. Пример схемы с таймером 555

Такие микросхемы, как микросхемы АЦП, тоже полезно рассмотреть в программе, чтобы без проблем использовать реальное устройство в реальной схеме.

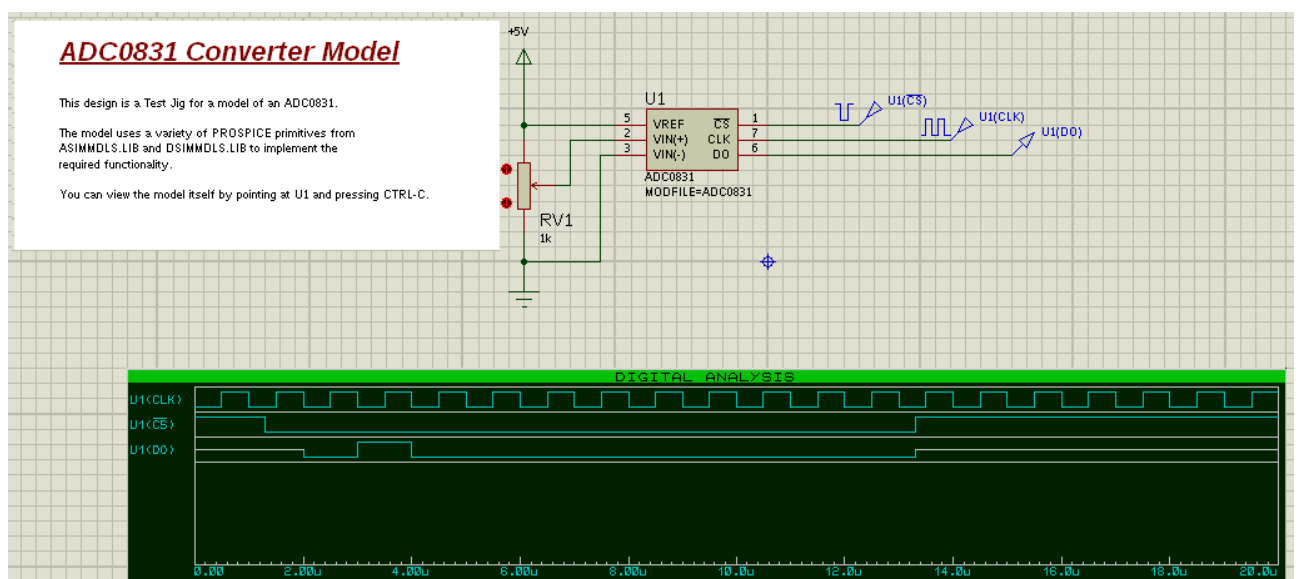


Рис. 1.13. Схема с АЦП

И не менее интересно разобраться с микросхемой ЦАП, осуществляющей обратное преобразование из цифрового представления данных в аналоговый вид.

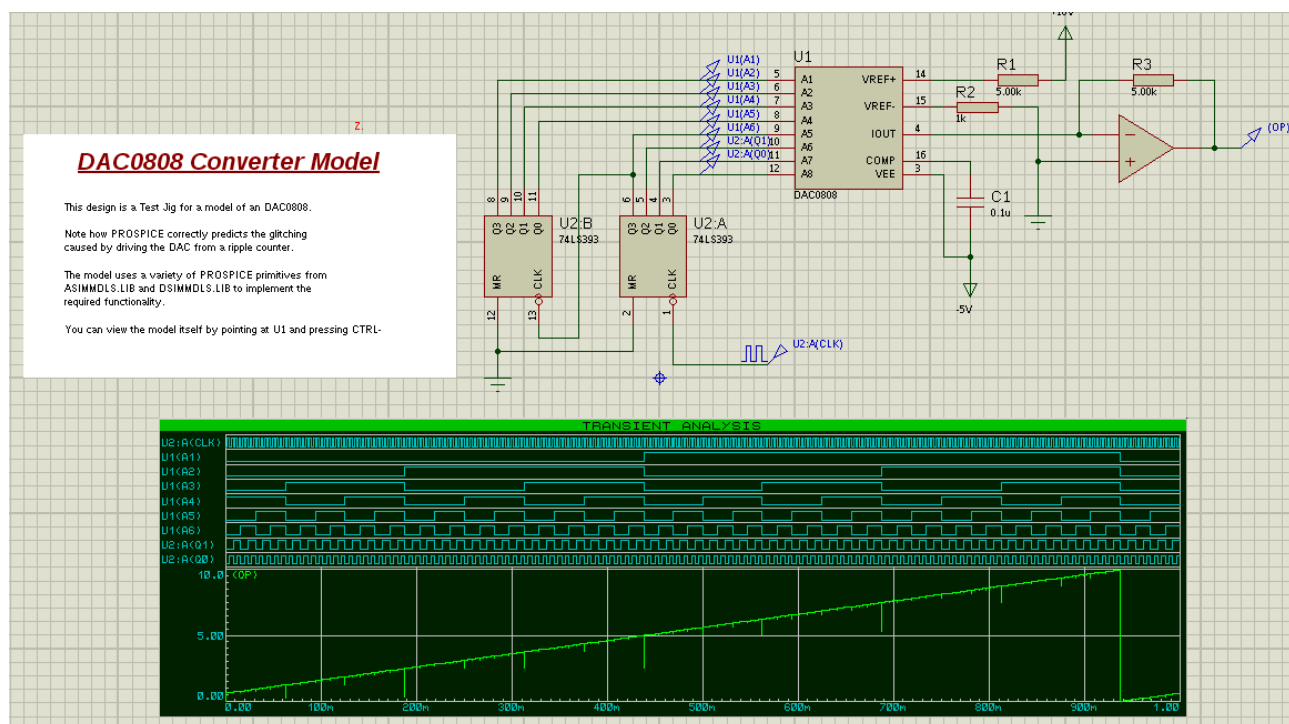


Рис. 1.14. Схема с ЦАП

Резюме

Программа ISIS, как явствует из рассмотрения примеров, пришедших с программой, не только полезна, но и достаточно интересна и для любителей, и для профессионалов. Если последним доступно больше, предприятие может позаботиться о приобретении любых программ, то любителям, конечно, приходится учитывать свои возможности. Если у вас есть возможность поработать с программой Proteus, не упускайте такую возможность.

Первое впечатление, если вы привыкли работать с другой программой, может оказаться не самым благоприятным. Но поработайте подольше, просмотрите примеры, обратите внимание на папку с обучающими программами, Tutorials.

От простых схем, которые вам не придётся рисовать самостоятельно «на первых порах», до весьма сложных, рисовать которые вам не захотелось бы, вы опробуете разные подходы, разные приёмы моделирования. После этого легче будет разобраться и с собственными схемами, которые перед пайкой на макетной плате можно проверить с помощью программы моделирования.

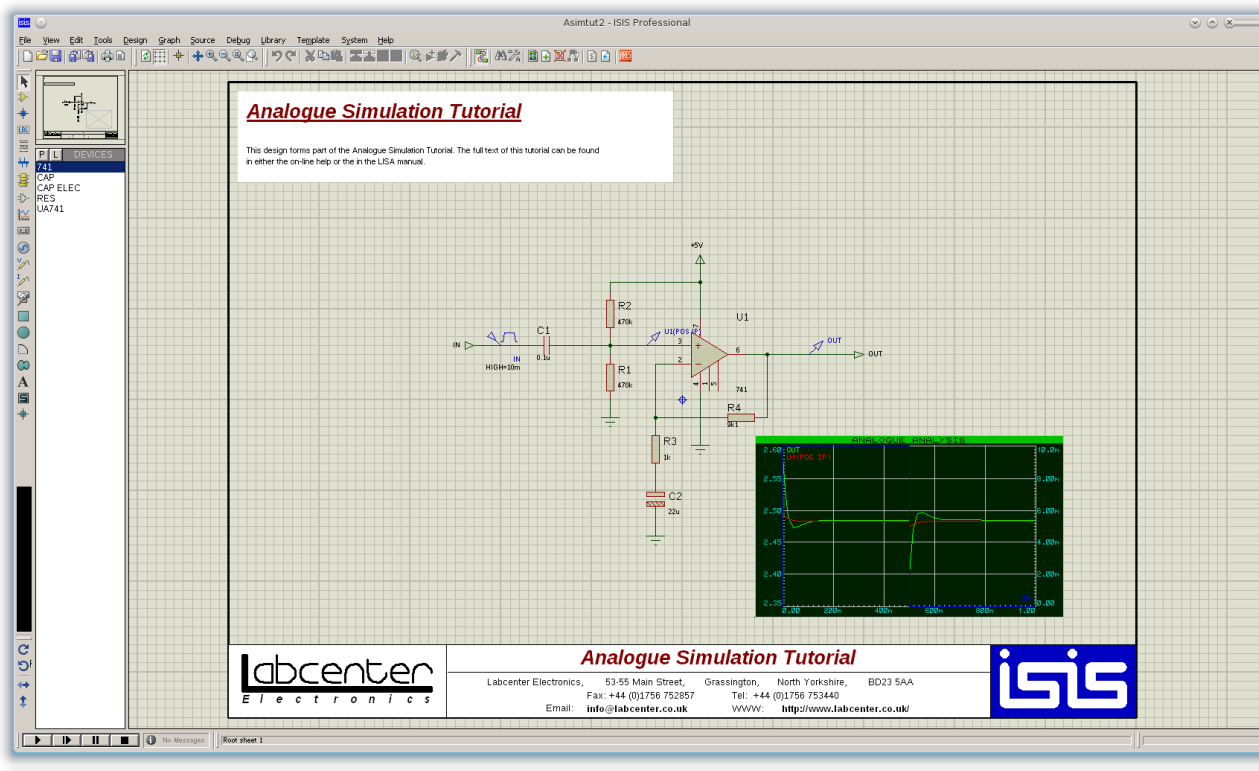


Рис. 1.15. Пример из обучающих программ

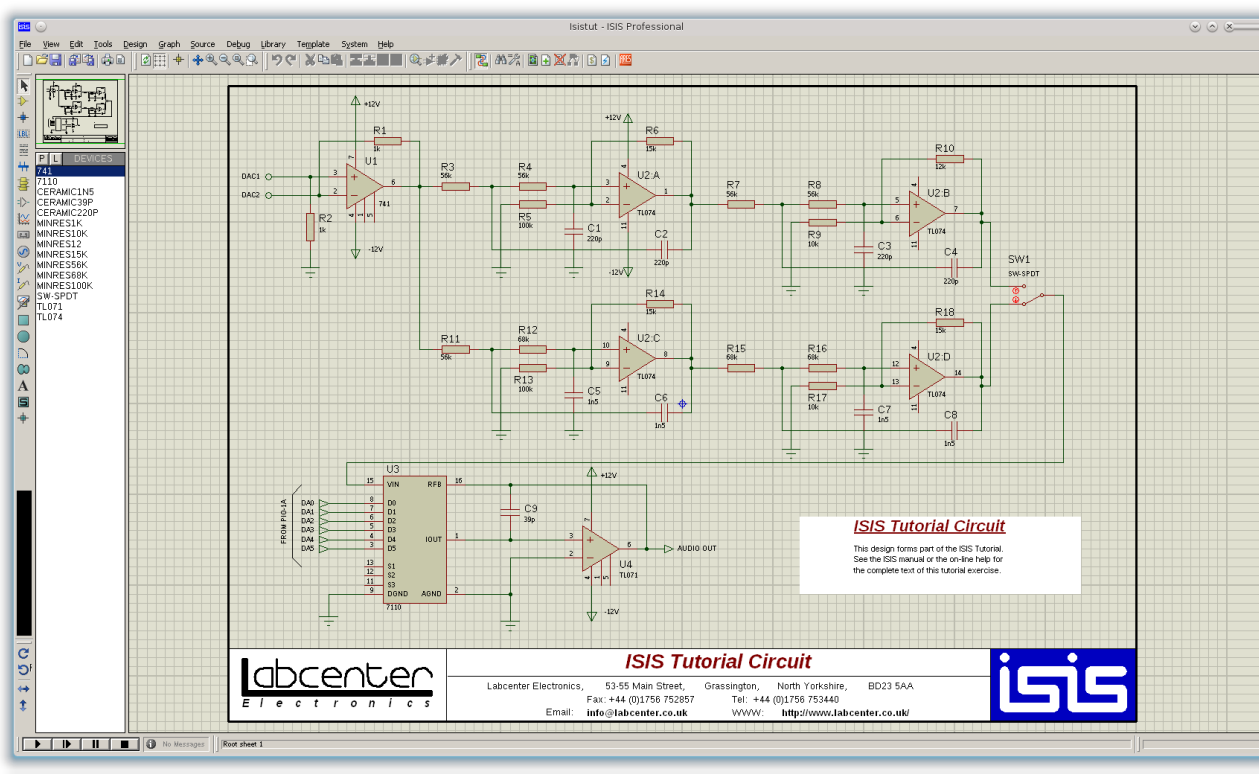


Рис. 1.16. Ещё один пример схемы обучающей части программы

Proteus 8 первая часть

Седьмая версия вполне, похоже, работоспособна в Linux. А восьмая версия не захотела устанавливаться. Возможно, если постараться, то и её можно установить, но...

Внешний вид программы при запуске изменился.

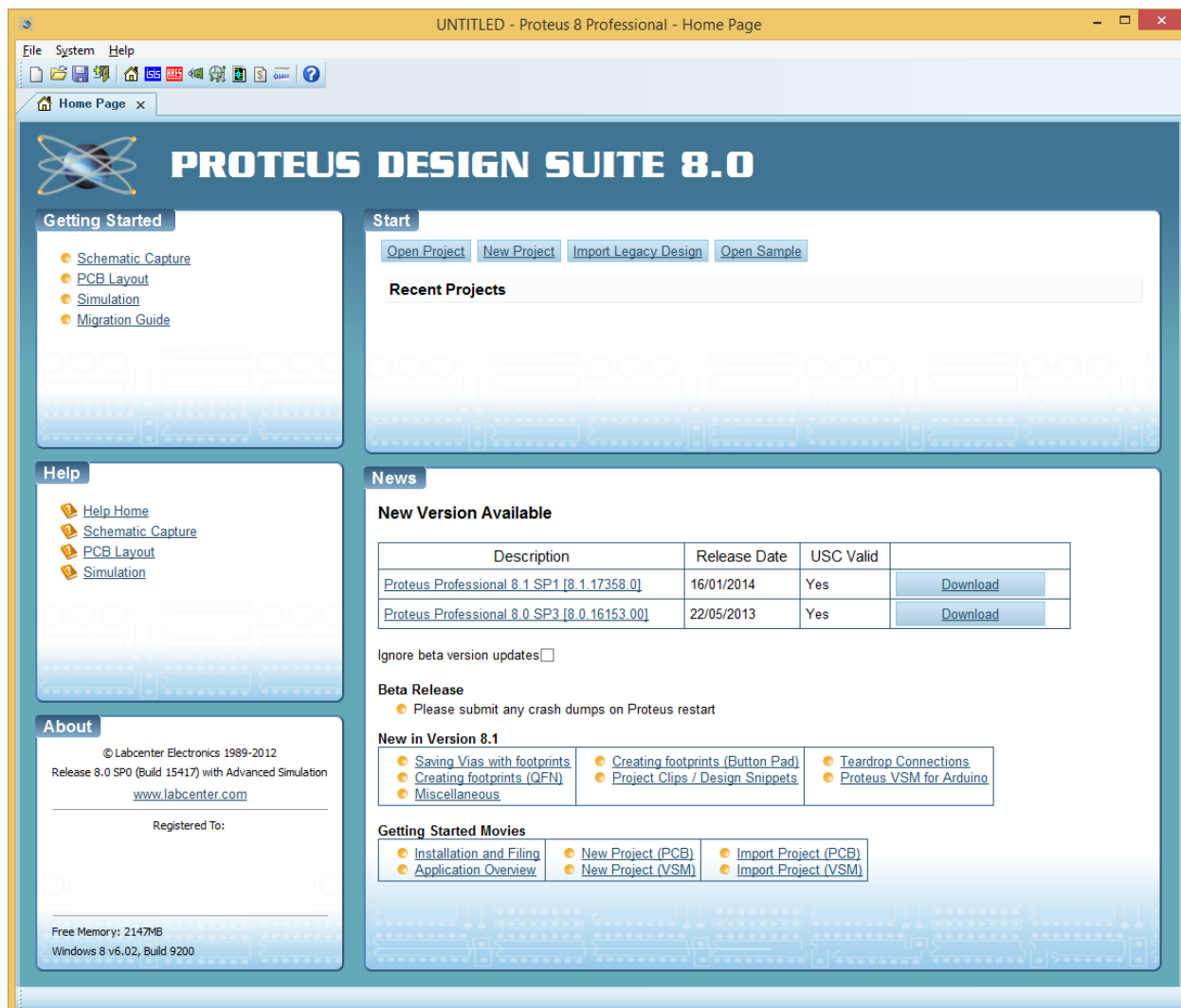


Рис. 2.1. Внешний вид диалога начала работы с программой

Скорее всего, изменений в программе больше, но сейчас меня интересует другое – работает ли программа в Windows 8.1?

В диалоге начала работы есть закладка Open Sample. Разумно начать с этого.

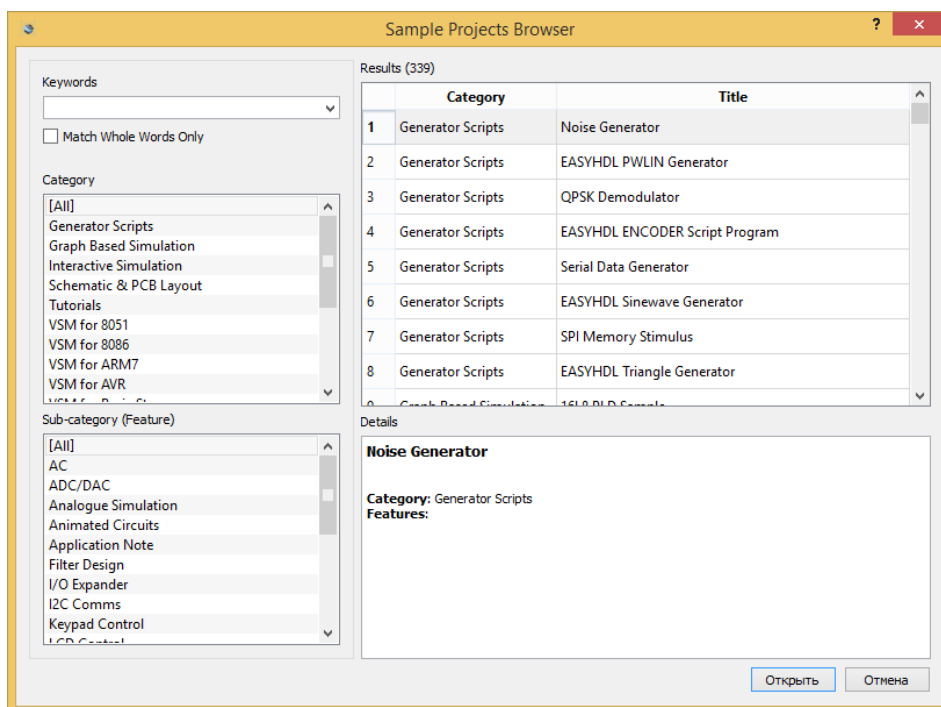


Рис. 2.2. Обзорщик проектов в папке примеров

Для пробной проверки подойдёт, наверное, любой проект. Пусть будет схема мультивибратора.

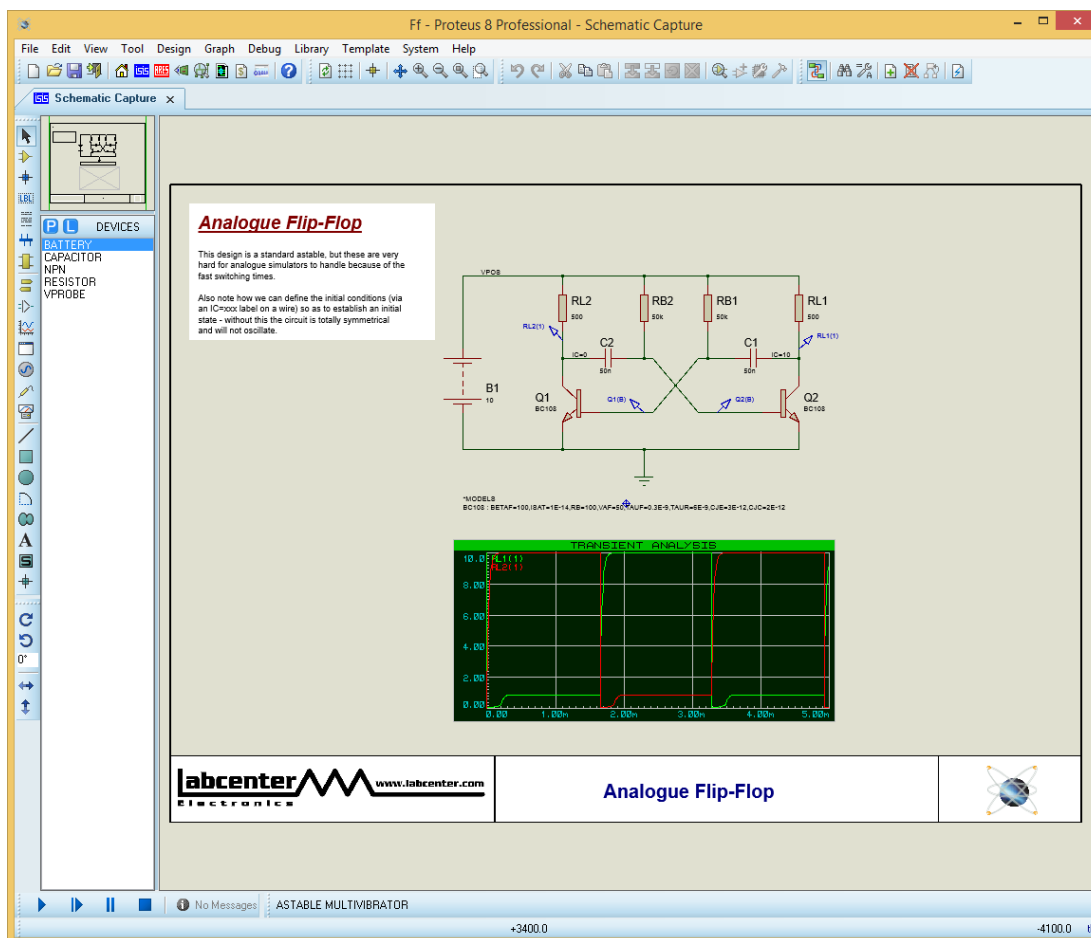


Рис. 2.3. Схема мультивибратора из набора примеров

Чтобы проверить моделирование, щёлкнем правой клавишей мышки по графику:

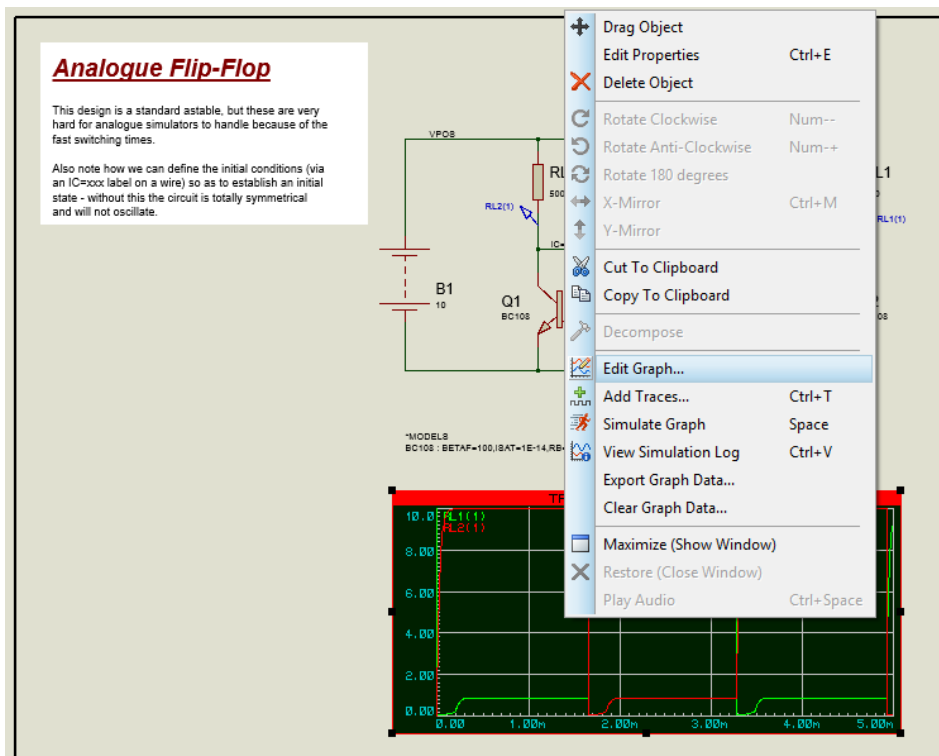


Рис. 2.4. Всплывающее меню для работы с графиком

Раздел Edit Graph позволяет изменить время симуляции, что изменит график работы устройства, если программа работает.

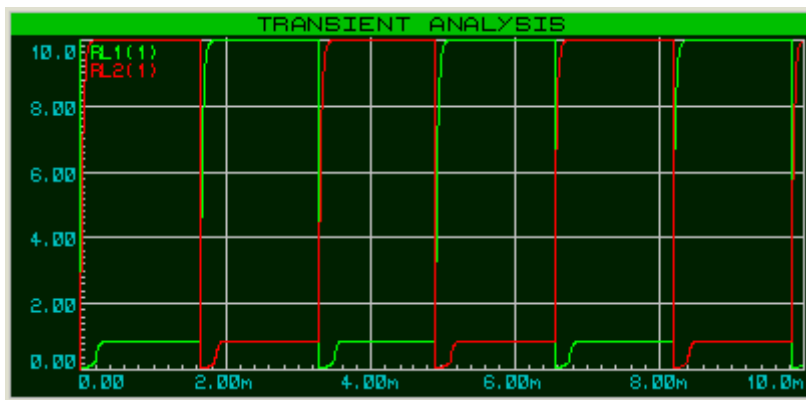


Рис. 2.5. Новый график работы мультивибратора

Ещё один способ проверить, работает ли программа, это использовать приборы, с помощью которых проверяется работа электрических схем в программе.

И есть много тех, кто предпочитает работу с виртуальным осциллографом, а не с графиками, у них есть возможность сделать это.

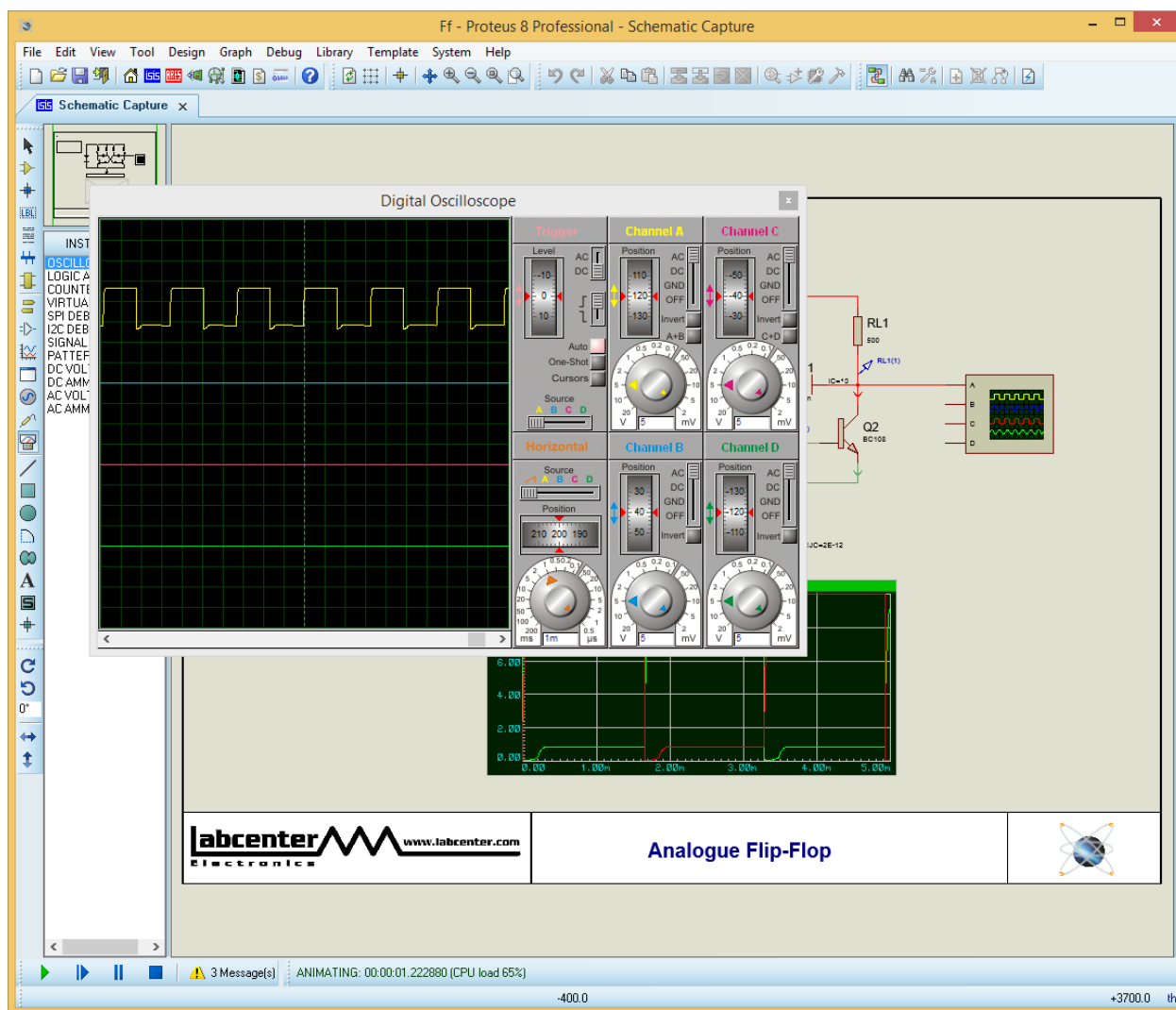


Рис. 2.6. Подключение к схеме виртуального осциллографа

Технология сквозного проектирования при разработке электронных устройств применяется довольно давно. Поэтому программы часто имеют вторую часть – разводку печатных плат (с созданием вспомогательных файлов для физического воплощения результата). Предыдущая версия тоже имела программу ARES в своем составе, но об этом я не упоминал, чтобы сделать это сейчас.

Итак. Вы нарисовали электрическую схему. Вы наладили её, используя средства моделирования программы. У вас нет сомнений в том, что схема работоспособна. Не обязательно в любительских условиях изготавливать печатную плату для единственного экземпляра – выгода использования печатной платы в серийном производстве. Но даже начинающим следует приобрести начальные навыки работы с программой ARES. Если в автоматическом режиме создать разводку печатной платы, то её можно использовать в качестве монтажной схемы, что уменьшит возможность ошибок при монтаже.

Используем схему мультивибратора для получения монтажной схемы.

Для перехода в программу разводки печатной платы можно использовать кнопку инструментальной панели.

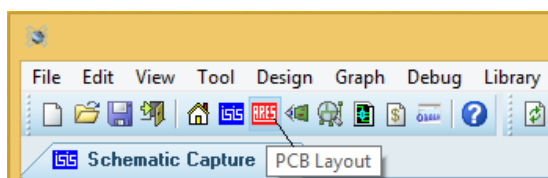


Рис. 2.7. Кнопка вызова программы ARES

Если всё в порядке, то после перехода можно увидеть перечень всех элементов схемы.

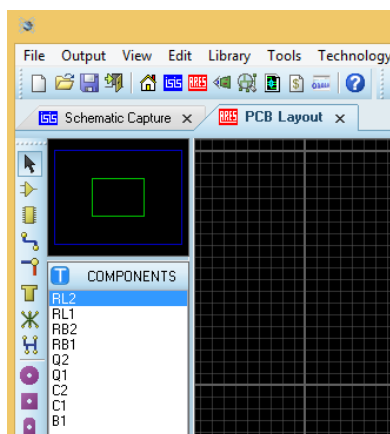


Рис. 2.8. Перечень всех монтируемых элементов схемы

Чтобы печатная плата была создана, нужно для всех элементов схемы иметь указание на корпус элемента, доступный программе ARES. Можно нарисовать схему заново, но я сделаю проще – заменю все элементы схемы на те, что имеют соответствующий корпус. Из схемы также будет исключена батарейка, её на плате не будет.

Для поиска резисторов я использую поисковую строку в менеджере компонентов (слева на инструментальной панели есть кнопка перехода в режим выбора компонентов, а в окне проекта кнопка вызова менеджера – иконка с буквой P). Для поиска резисторов можно в строку поиска ввести minres. Далее следует номинал резистора, но резистора в 500 Ом вы не найдёте, ищите резистор 470 Ом или 510 Ом. Аналогично и для поиска резистора 50 кОм используйте правильное значение. Таким образом, в последнем случае строка поиска выглядит так: minres51k.

Перед заменой рабочее поле чертежа выглядит так:

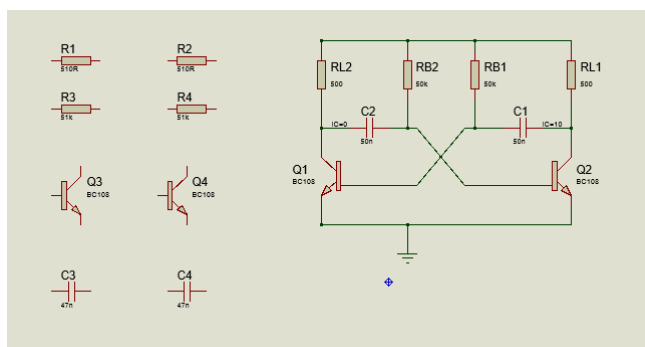


Рис. 2.9. Рабочее поле чертежа с подобранными компонентами

Произведя замену, можно упорядочить нумерацию элементов и перейти в программу ARES.

Если вы не задали размер платы, вы получите предупреждение при попытке автоматически расставить все элементы:

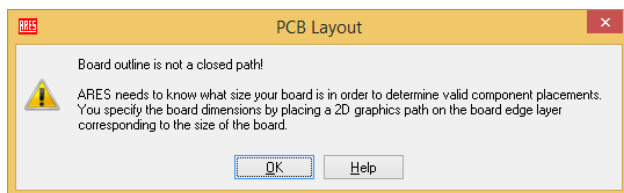


Рис. 2.10. Предупреждение, что размеры платы не выбраны

Выберите прорисовку прямоугольника на инструментальной панели графических элементов. В нижней части рабочего окна выберите Board Edge. И нарисуйте прямоугольник в поле редактора.

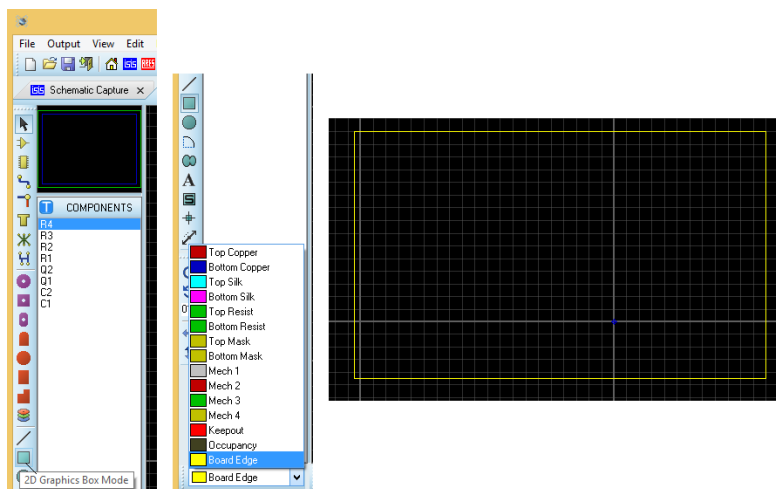


Рис. 2.11. Первые операции при создании платы

Контур платы выделится жёлтым цветом. Используя автоматическую расстановку элементов, вы получите:

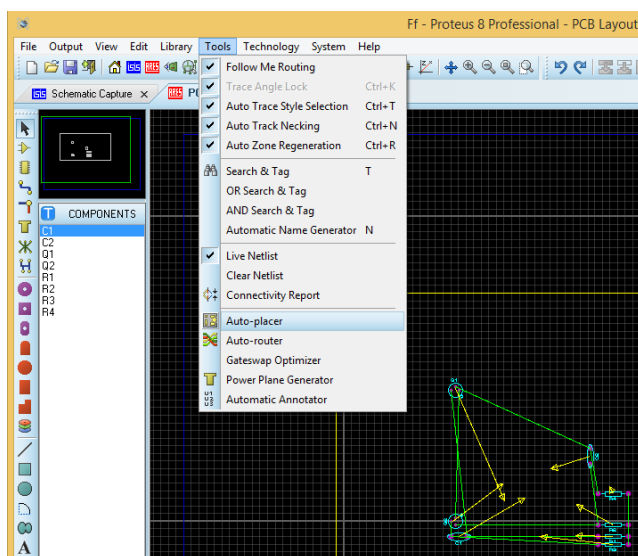


Рис. 2.12. Автоматическая расстановка элементов схемы

Используя команду Auto-router в том же меню в следующей строке, вы получите разводку печатной платы.

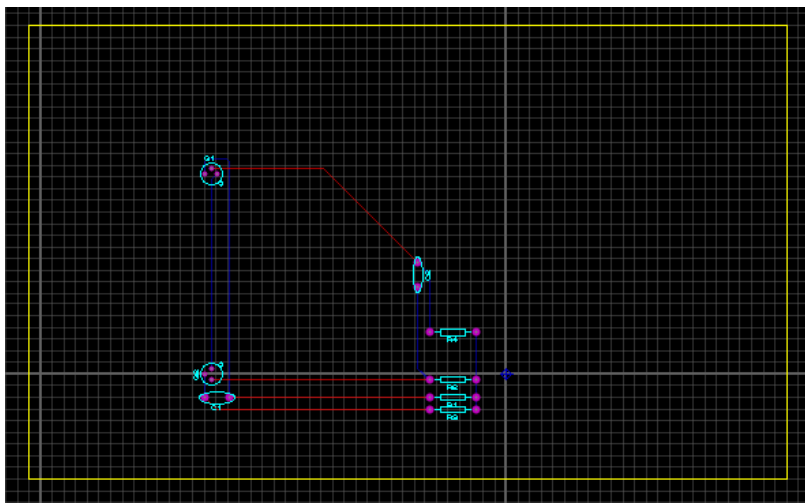


Рис. 2.13. Автоматически разведённая печатная плата

Как и во многих современных программах, вы можете посмотреть трёхмерное изображение платы.

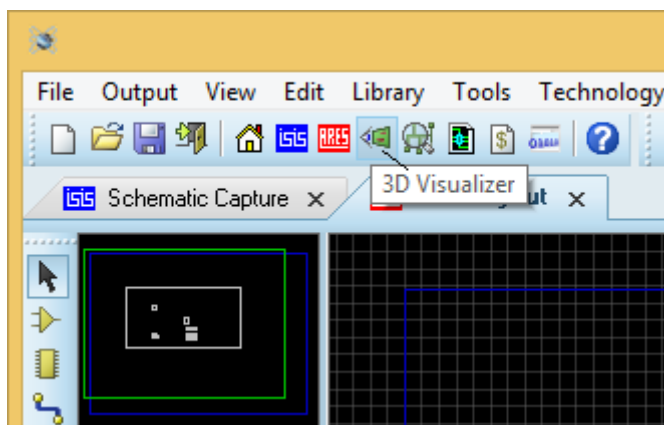


Рис. 2.14. Пространственный обозреватель

Изображение можно поворачивать, рассматривать и сложить более полное представление о полученном результате.

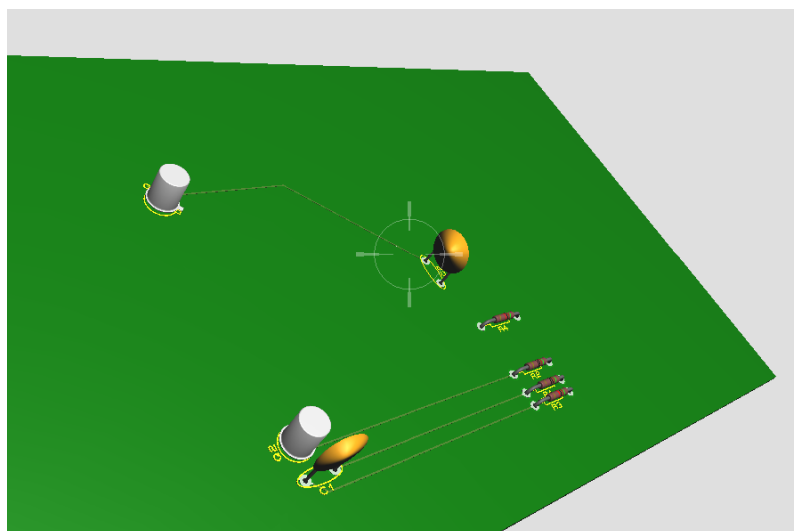


Рис. 2.15. Трёхмерный вид полученной платы

Разводка печатной платы требует знаний и опыта. Описанная выше автоматическая разводка полезна для знакомства с программой; полученные соединения в программе помогут выполнить монтаж. Но начинать монтаж, если у вас нет большого опыта в этом, лучше с макетной платы. Без опыта трудно учесть паразитные связи между элементами, которые могут «свести на нет» всю вашу работу.

Среди готовых примеров есть примеры разводки печатных плат.

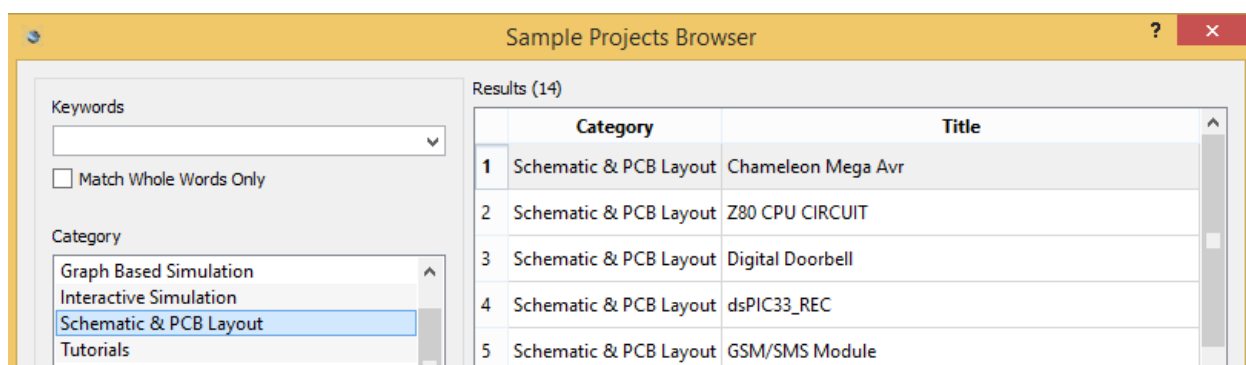


Рис. 2.16. Примеры печатных плат

Схемы в этих примерах, не в пример приведённой выше, достаточно «солидные», чтобы у вас не сложилось превратное впечатление о малых возможностях программы Proteus. Это начинать лучше с простых схем, а работать можно и со схемами, состоящими из нескольких модулей, нарисованных на нескольких листах.

Для работы со сложными схемами нужно хорошо разбираться во многом, что приходит с опытом работы, когда в сложной схеме вы готовы быстро различить хорошо известные решения.

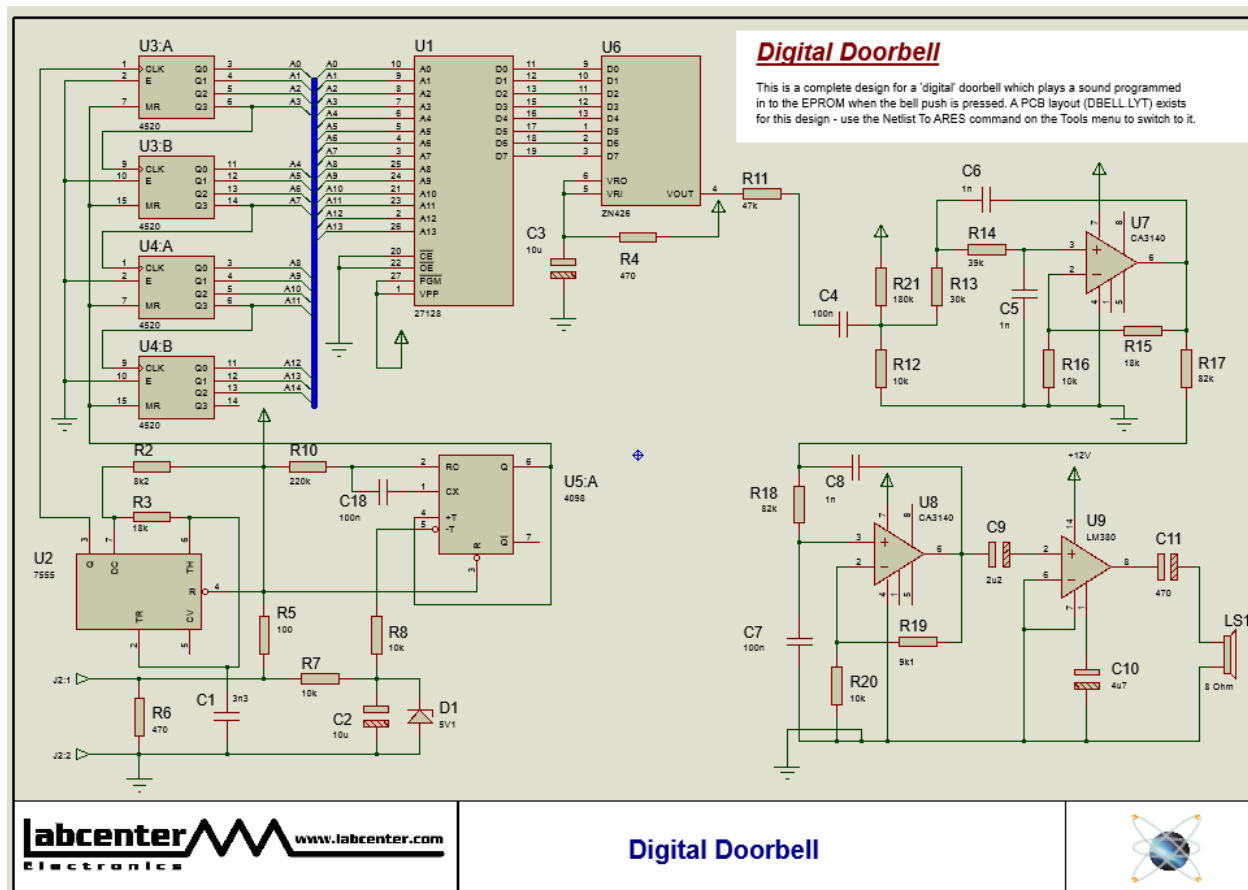


Рис. 2.17. Пример из набора примеров разводки печатных плат

А так выглядит хорошо разведённая печатная плата.

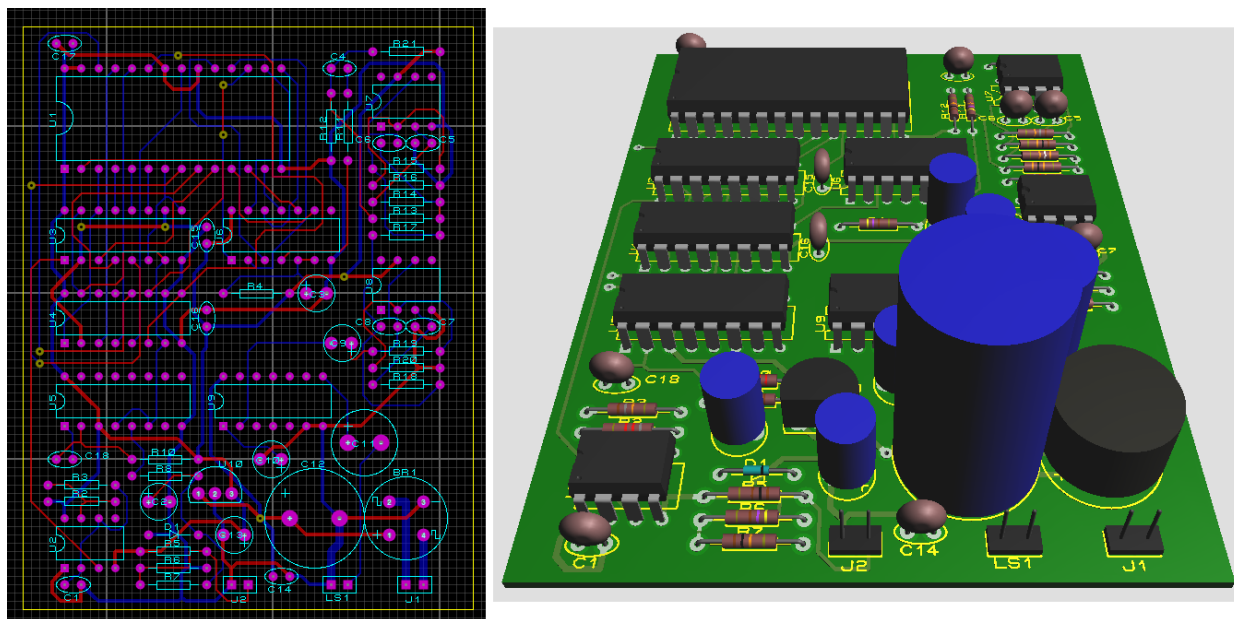


Рис. 2.18. Печатная плата из набора примеров Proteus 8

Если вас интересует разводка печатных плат в программе Proteus, поищите в Интернете, я встречал перевод руководства, надеюсь, и вы найдёте.

Proteus 8 продолжение

Особое значение программы Proteus я вижу в её особенных возможностях для работы с микроконтроллерами. Можно не выходя из программы создать устройство на базе многих типов контроллеров. Впечатляет даже список доступных моделей.

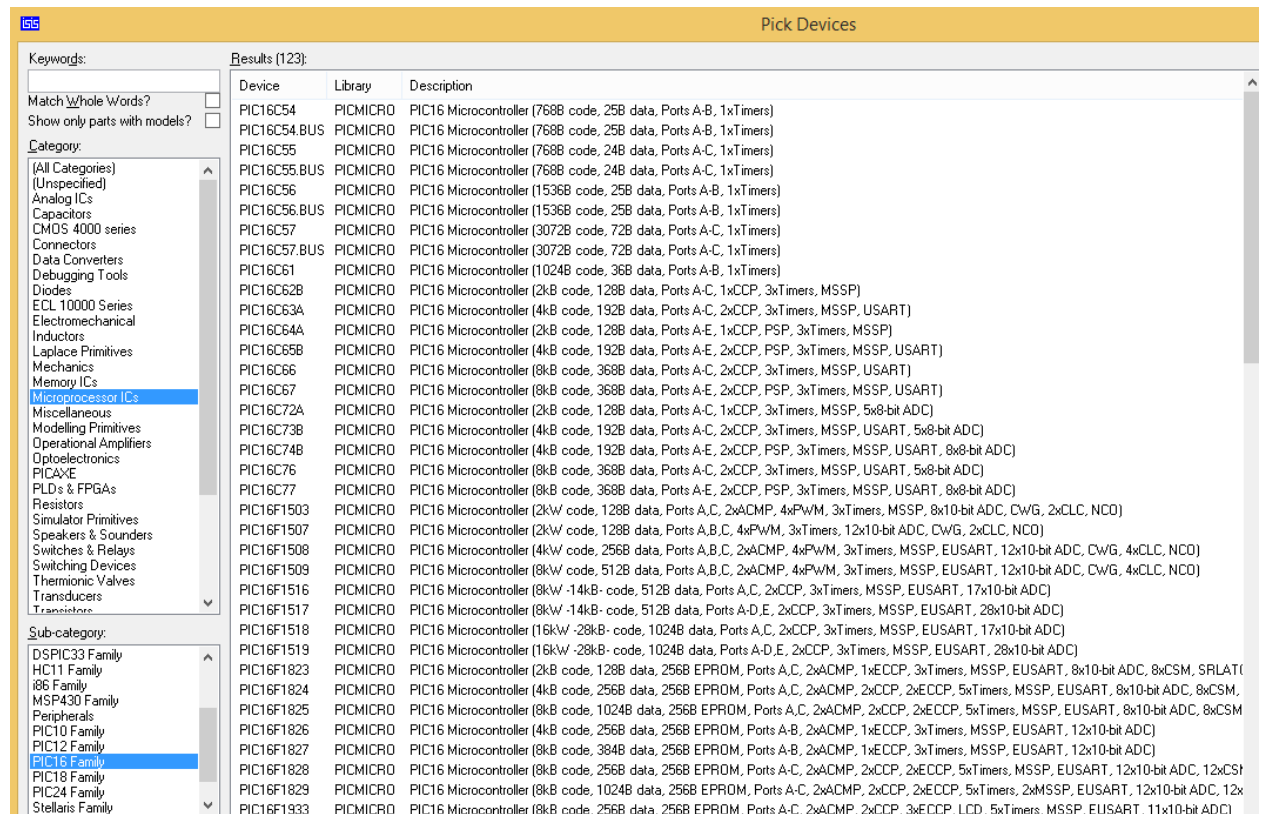


Рис. 3.1. Список модельного ряда одного из производителей микроконтроллеров

И это, как вы видите, не полный список, а только его часть.

В этой версии программы можно легко создавать код своей программы на языке высокого уровня. Хотя вы можете встретить утверждение, что код программы следует писать только на ассемблере, не верьте этому. Для работы на ассемблере нужно не только знание команд конкретной модели, не только знание архитектуры конкретной модели микроконтроллера, но иметь большой опыт создания программ на ассемблере. Это ближе к искусству, чем к ремеслу, а, значит, требует не только больше времени на обучение, но и большой любви к этому виду деятельности.

Для начинающих наибольшую трудность в использовании ассемблера будет представлять не набор команд, а связь между работой конечного устройства и этими командами. Даже ясное понимание того, что делает та или иная команда, никак не связывается с вопросом, зачем выполнять эти команды. Только долгое использование языка позволяет начинать работу с блок схемы программы, элементы которой уже и заполняются хорошо знакомыми командами.

В отличие от других программ Proteus не только имеет обширный список микроконтроллеров, не только позволяет написать код, но позволяет проверить работу схемы, когда кроме микроконтроллера устройство имеет много внешних элементов: от кнопок и светодиодов, до аналоговых, цифровых и электромеханических деталей. Вот пример такой схемы:

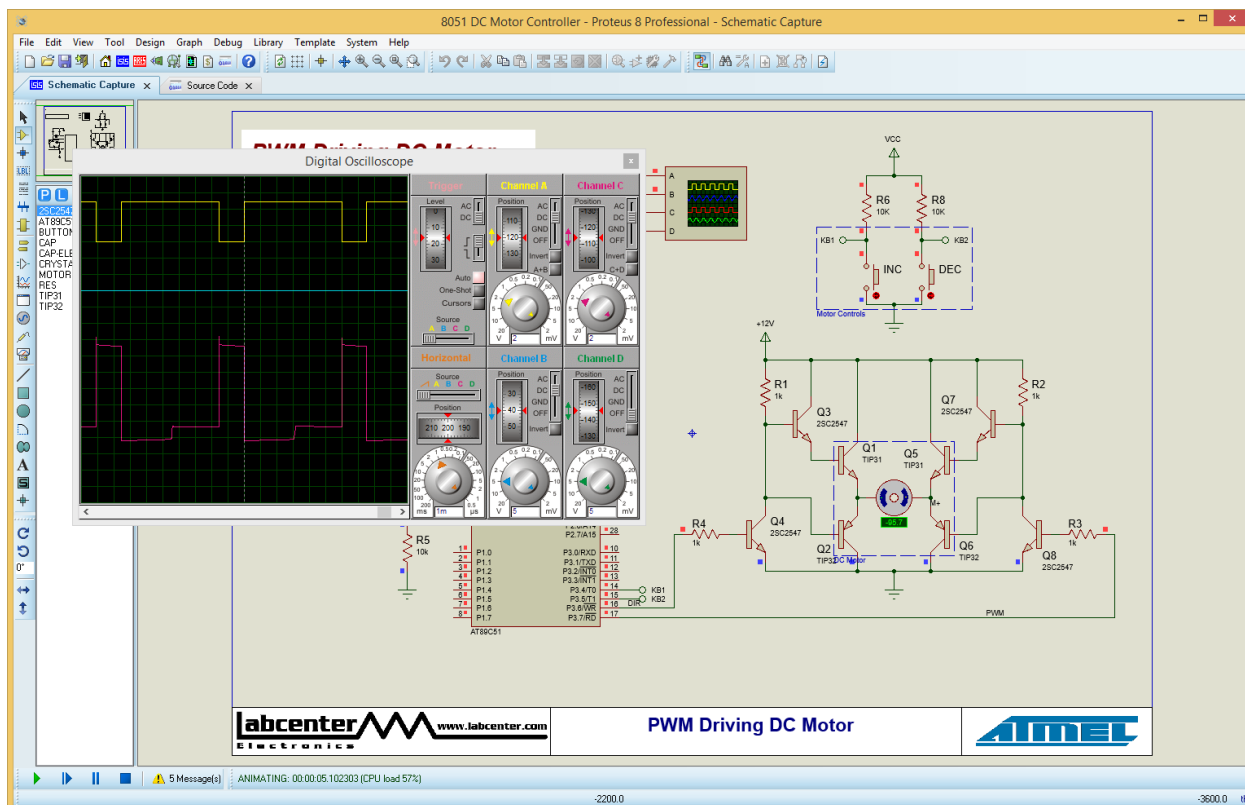
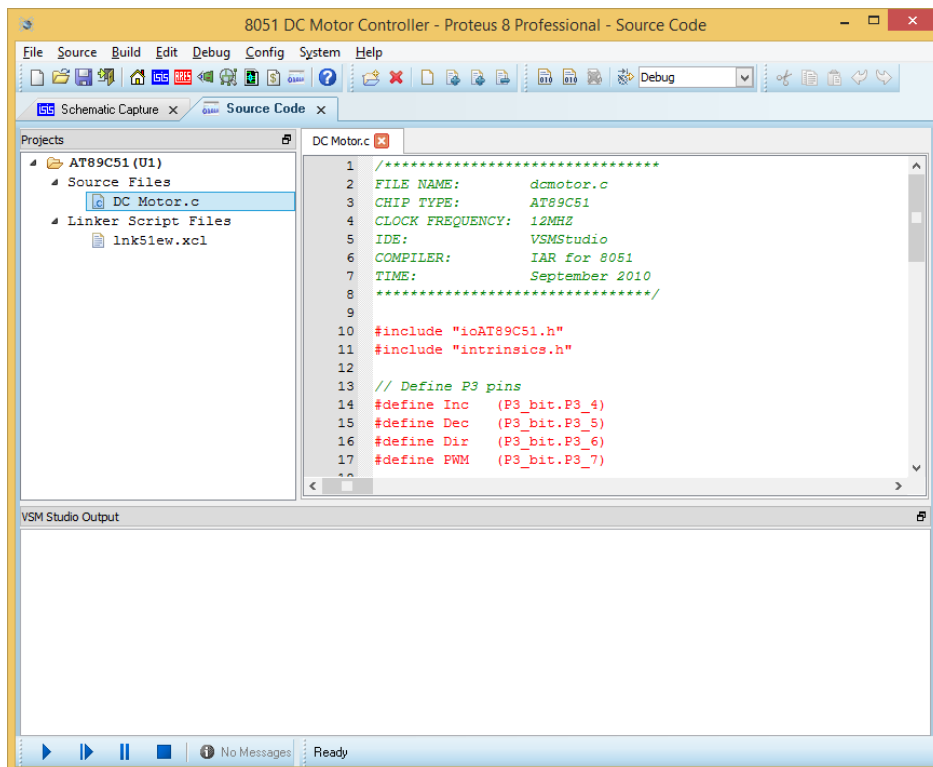


Рис. 3.2. Один из примеров схемы в папке Sample

Там же вы можете посмотреть и код программы на языке Си – закладка Source Code.



Язык высокого уровня позволяет больше сосредоточить своё внимание на том, что должен делать микроконтроллер, а не на том, как он это будет делать. Однако начинающим следует иметь в виду, что общность языка Си – на нём пишутся приложения и для компьютера, и для микроконтроллера – не исключает того, что руководство к конкретному компилятору всегда следует держать под рукой. Разные компиляторы могут иметь различия в синтаксисе, разный набор, например, математических и логических операций. И начинающим лучше использовать среду разработки программ на языке Си, имеющую хорошую библиотеку разных функций. К созданию собственных функции можно обратиться тогда, когда будет личный опыт в написании программ. Вместе с тем, язык Си достаточно универсален, программы для одной модели микроконтроллера легко переделать для другой или для модели другого производителя.

Программа ISIS, если использовать её для проверки готовой программы, позволяет легко это сделать. Достаточно добавить нужную модель микроконтроллера, указать готовый код в диалоге настройки свойств микроконтроллера, добавить все внешние элементы, если они есть, и запустить моделирование.

Если вы уже начали работать с каким-то компилятором, вы можете проверить, есть ли он в списке тех, что могут работать в ISIS.

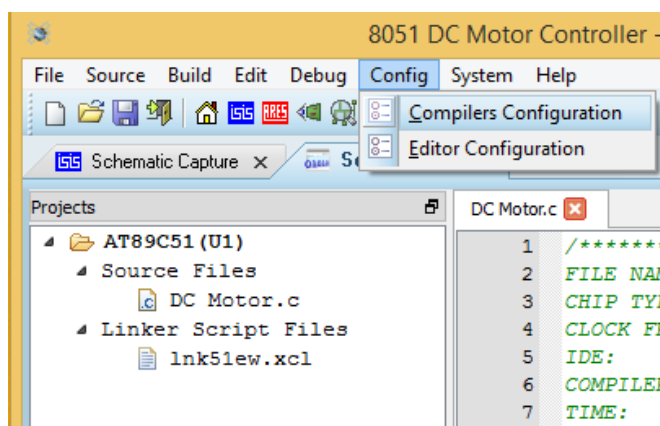


Рис. 3.4. Раздел меню со свойствами компиляторов

Этот раздел меню появляется тогда, когда вы переходите на страницу исходного кода. В диалоге, который открывается в разделе меню (Config -> Compilers Configuration), перечислены те компиляторы, которые установлены на вашем компьютере – они отмечены словом «Yes» – и те, которые могут быть загружены, например, из Интернета.

Если вы хотите попробовать полностью перейти на создание своих устройств на базе микроконтроллеров в среде Proteus, то проверьте, будет ли работать ваш любимый компилятор.

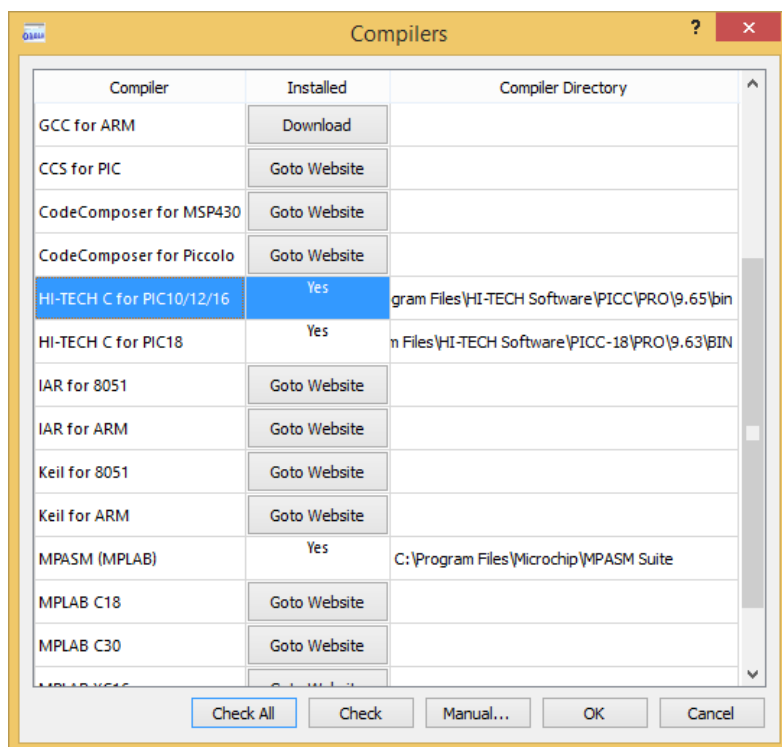


Рис. 3.5. Диалоговое окно конфигурации компиляторов

Давайте проверим, что получится, создав новый проект и создав самую простую из всех возможных программ. В первоначальном окне программы выберем New Project. Открывается диалог помощника создания нового проекта.

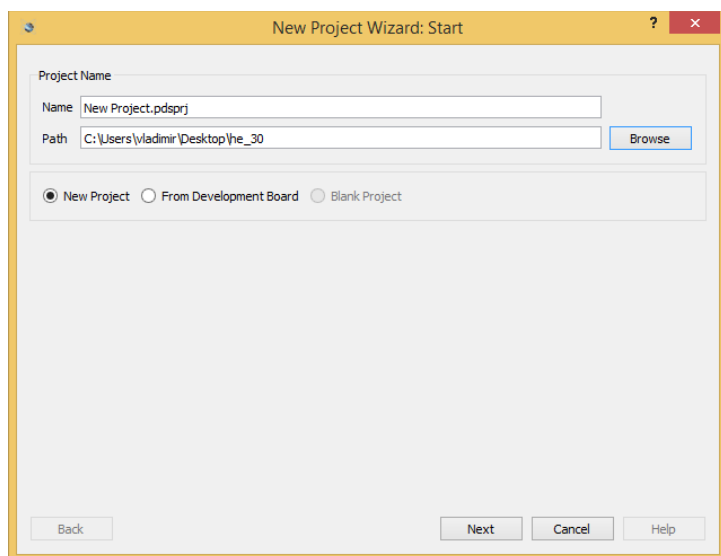


Рис. 3.6. Начало создания нового проекта

На этой странице можно задать имя проекта и указать место, где этот проект будет располагаться.

Для продолжения служит кнопка **Next**. На следующей странице можно определить, будете ли вы использовать шаблон для схемы. Например, выбрать формат чертежа, а далее, на следующей странице, определиться, нужна ли вам разводка печатной платы. Все эти действия вы выполняете с помощью мышки, выделяя нужные решения щелчком левой клавиши мышки, и переходите к

следующим свойствам проекта с помощью клавиши **Next**. На странице выбора микроконтроллера есть выбор из списка модельного ряда и списка конкретной модели, там же вы выбираете компилятор.

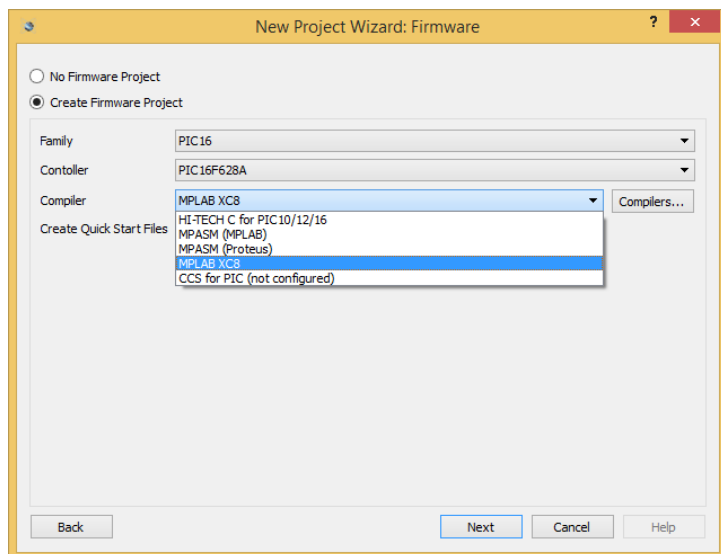


Рис. 3.7. Выбор модели микроконтроллера и языка программы

Последняя кнопка, которую вы нажмёте – это Finish. В итоге вы получите заготовку под ввод схемы и шаблон для исходного кода.

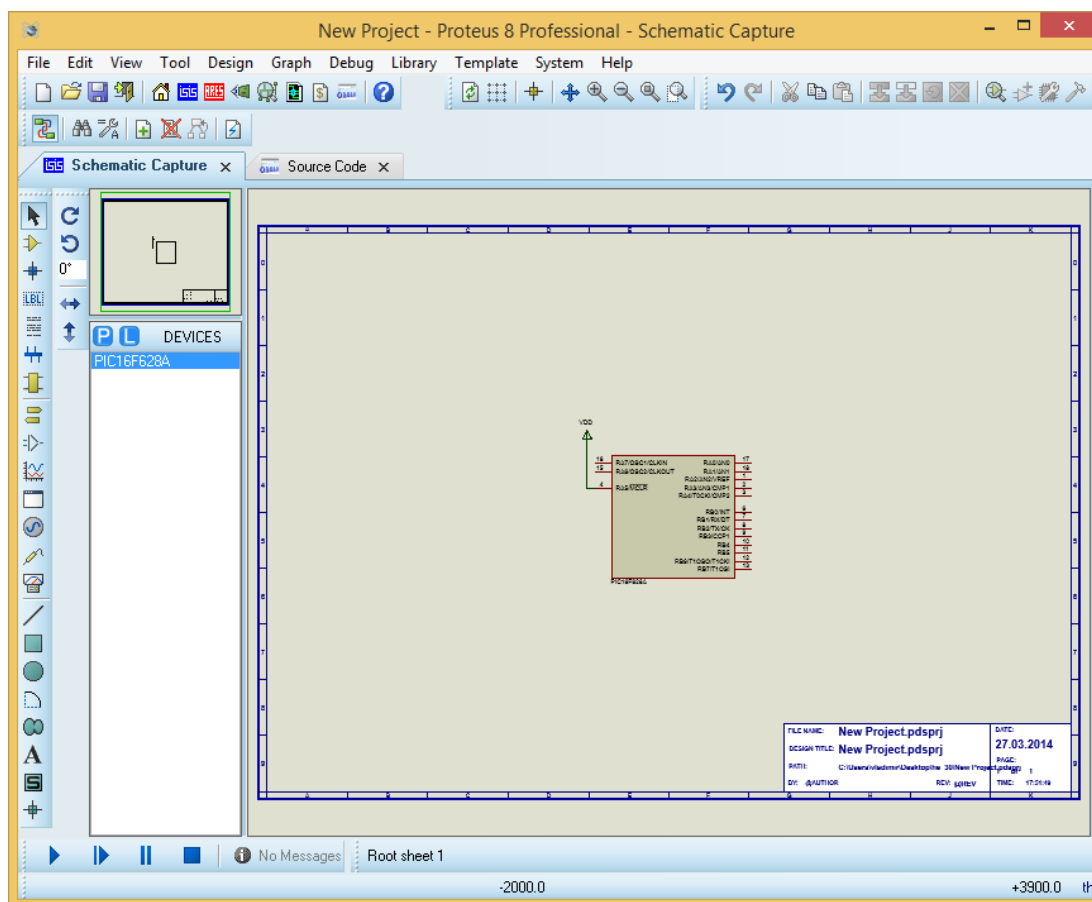


Рис. 3.8. Заготовки проекта

На крупном предприятии, наверное, свои правила оформления чертежей, но для небольшой фирмы, выпускающей, скажем, ёлочные гирлянды, можно распечатать схему из программы и подшить к документам.

Итак, переходя на страницу кода, используя готовый шаблон, запишем программу. Закончив записывать текст, используем возможность оттранслировать программу.

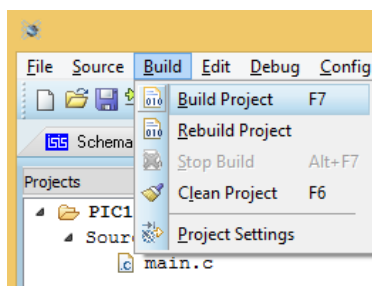


Рис. 3.9. Раздел основного меню для трансляции программы

Если в тексте программы есть ошибки, то при трансляции вы получите сообщение об ошибках.

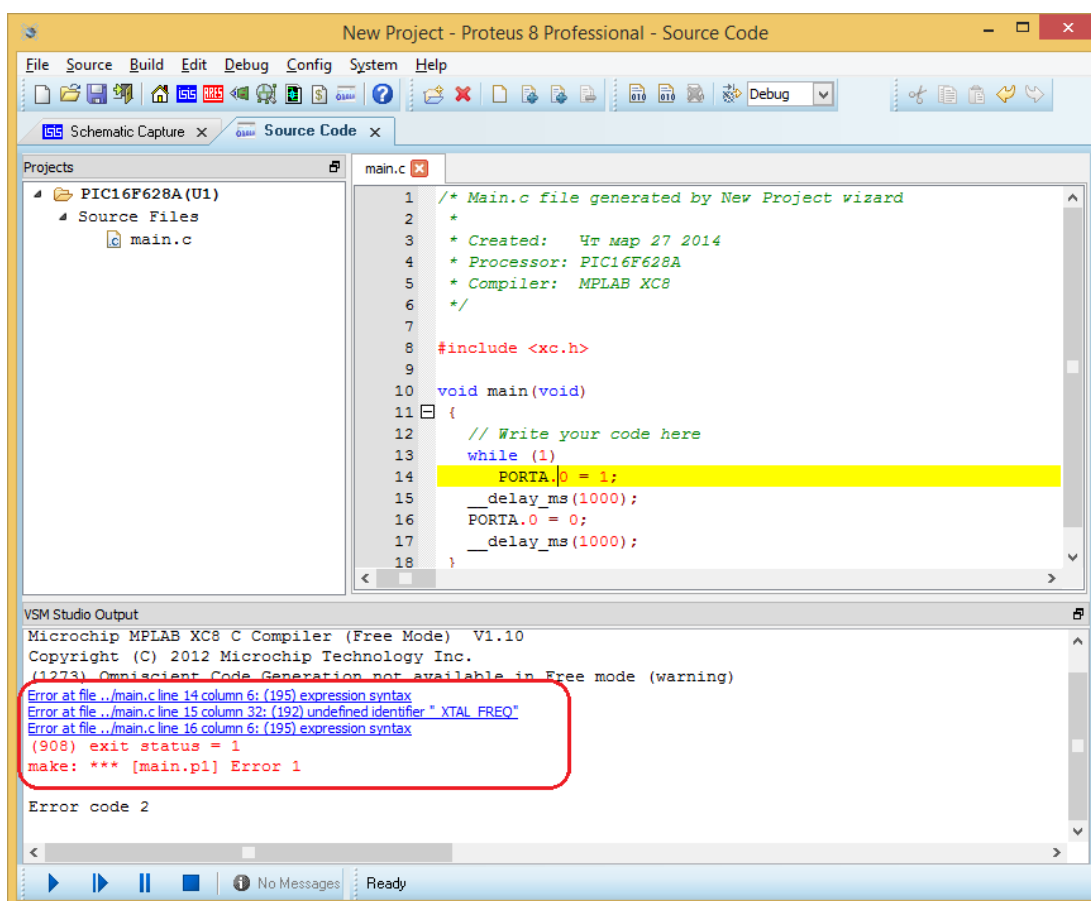


Рис. 3.10. Сообщение об ошибках

После того, как ошибки найдены, ошибки исправлены, можно повторить трансляцию, чтобы получить работающую программу. При удачной компиляции программы, вы получите сообщение о том, что проект скомпилирован успешно.

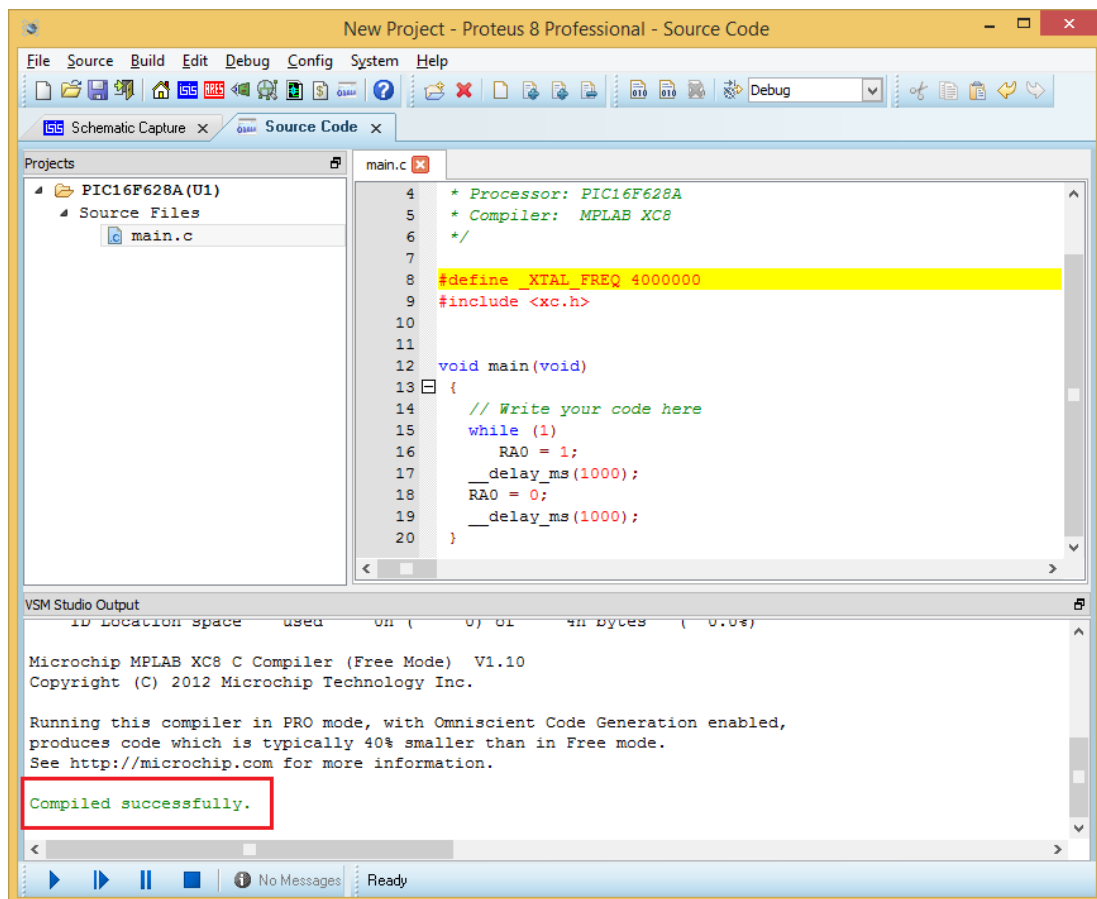


Рис. 3.11. Завершающая фаза компиляции проекта

Синтаксических ошибок проект не содержит, но и работать программа не желает. И тому есть причины. Исправим текст, добавив слово конфигурации и пару скобок.

```

#include <xc.h>
__CONFIG(FOSC_INTOSCIO & WDTE_OFF & PWRTE_OFF & MCLRE_OFF & BOREN_OFF &
LVP_OFF & CPD_OFF & CP_OFF);
#define _XTAL_FREQ 4000000

void main(void)
{
    // Write your code here
    TRISA = 0xFE;
    while (1){
        RA0 = 1;
        __delay_ms(1000);
        RA0 = 0;
        __delay_ms(1000);
    }
}

```

Теперь программа работает, в чём можно убедиться, вернувшись на страницу схемы.

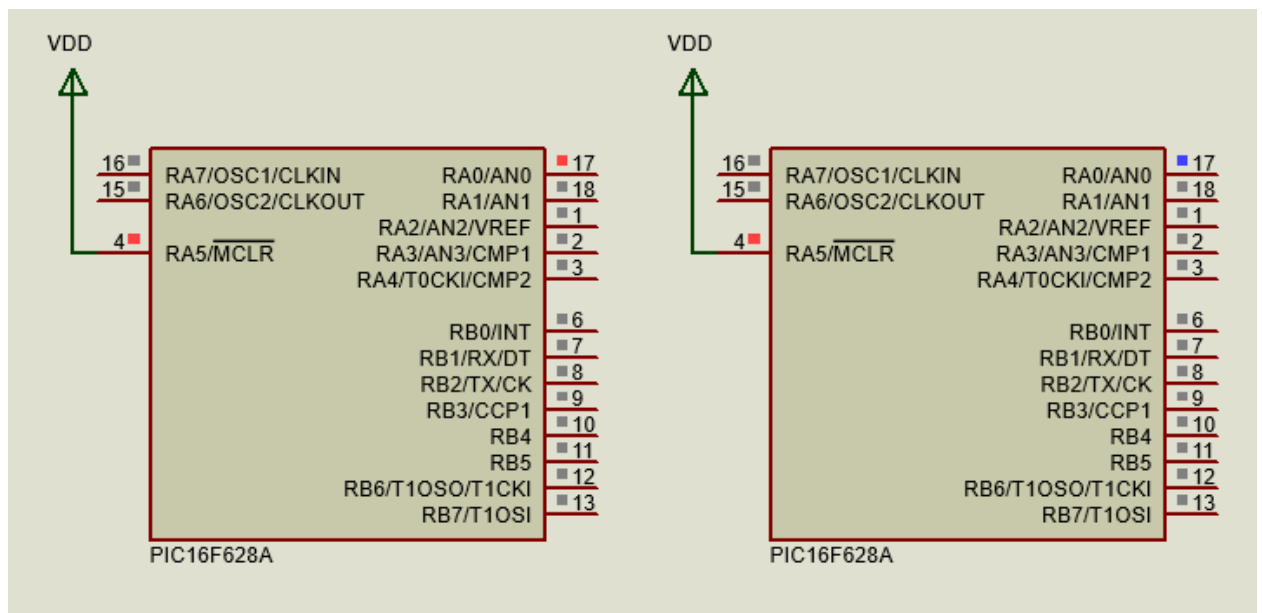


Рис. 3.12. Окончание работы над программой

В программе есть отладчик, но сказать что-то определённое... Да, есть окно наблюдения за переменными; да, можно посмотреть регистры; нет, мне не удалось запустить отладку, как и найти что-то в файле помощи о том, как это сделать.

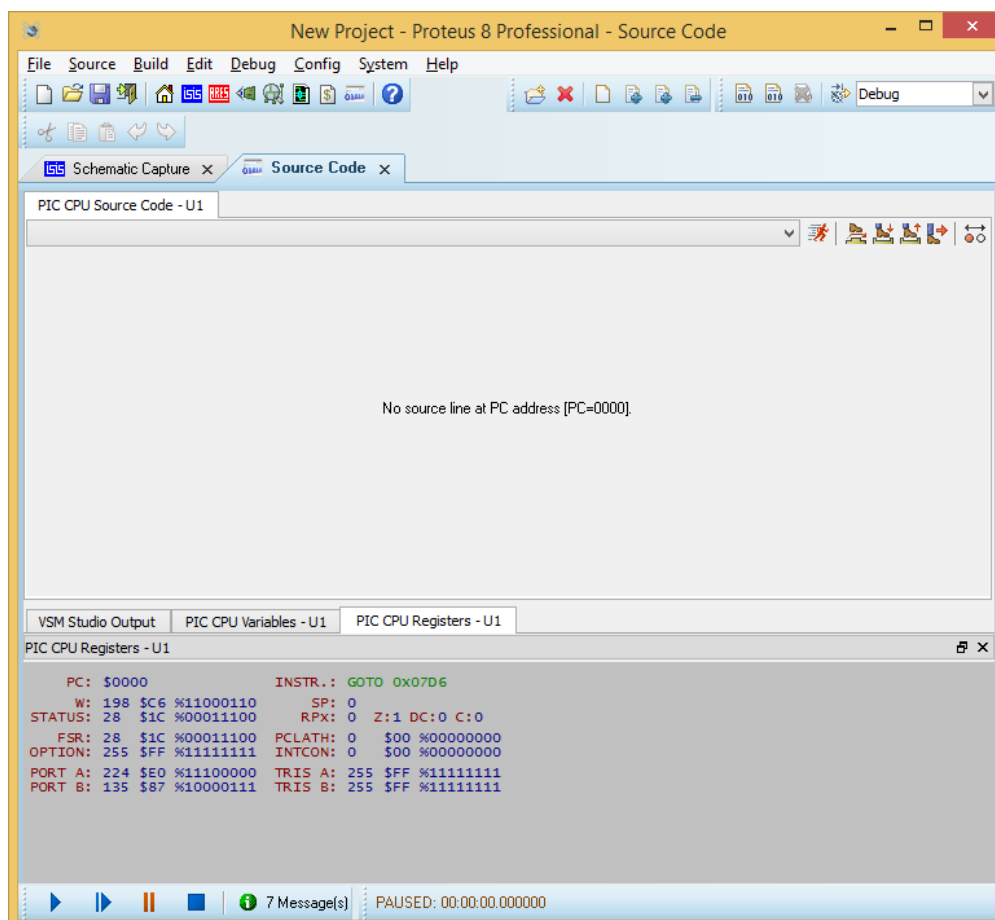


Рис. 3.13. Несостоявшийся сеанс отладки

Если мне не удалось, то это мои проблемы, не думаю, что это проблемы программы. А последнее, что интересует меня – можно ли использовать ранее созданные проекты в предыдущей версии? И не столько это интересно мне, сколько, думаю, полезно знать тем, кто пользуется программой. В одном из рассказов я использовал ISIS для моделирования работы датчика температуры. Открывая файл, я получаю сообщение:

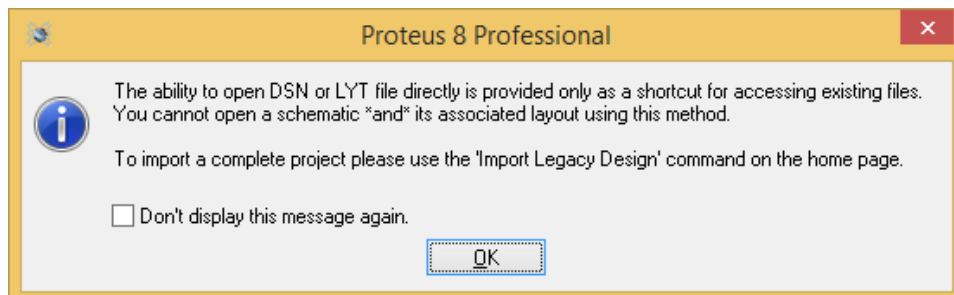


Рис. 3.14. Сообщение при открывании файла, сделанного в предыдущей версии

Суть сообщения в том, что открывая файл схемы, я не получаю полного доступа, например, к выполненной ранее (если она имела место) разводке печатной платы. Для полного доступа к проекту его следует импортировать (в меню File есть такая команда). В данный момент мой интерес проще, будет ли работать схема?

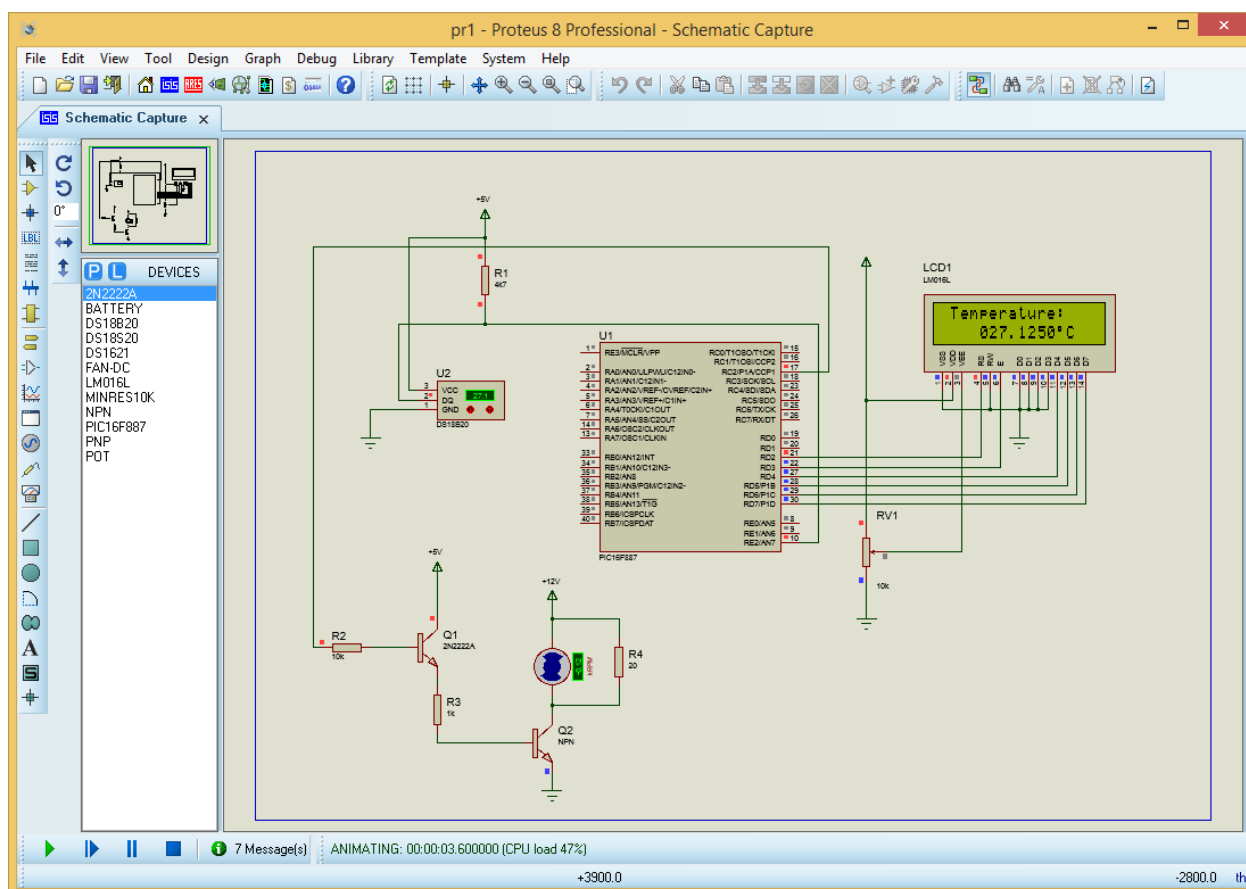


Рис. 3.15. Работа схемы, выполненной в предыдущей версии программы

Датчик температуры показывает правильное значение, вентилятор крутится, скорость вращения вентилятора, как и задумывалась, меняется при изменении температуры – схема работает. Что и требовалось проверить.

Резюме

Я не работаю с программой Proteus не потому, что она меня не устраивает, но нет задач, которые следовало бы решать с помощью программы. Если у радиолюбителя есть возможность поработать с программой, то он может определиться, устраивает ли она его. Для предприятий – учебных ли, производственных – применение программы зависит от таких факторов, как стоимость программы, опыт работы сотрудников с программой, но, и это главное, применимость программы к решению конкретных задач, стоящих перед предприятием.

Однако не так много программ из тех, что мне довелось видеть, позволяют моделировать работу устройств на базе микроконтроллеров, которые имели бы много внешних компонентов и обширный список доступных моделей и микроконтроллеров, и внешних устройств.

Для небольшой фирмы, выполняющей разработки по заказу или реализующей собственные задумки, программа может оказаться весьма полезна, а её стоимость приемлема.

Радиолюбители, которых заинтересовала программа, могут найти на моём сайте мой перевод руководства к ранней версии ISIS. Руководство вполне поможет освоить работу и с более поздними версиями:

<http://vgololobov.narod.ru>

Я не думаю, что радиолюбителей не устроит седьмая версия программы, но и радиолюбители бывают разные. Чем и славен этот любознательный народ.