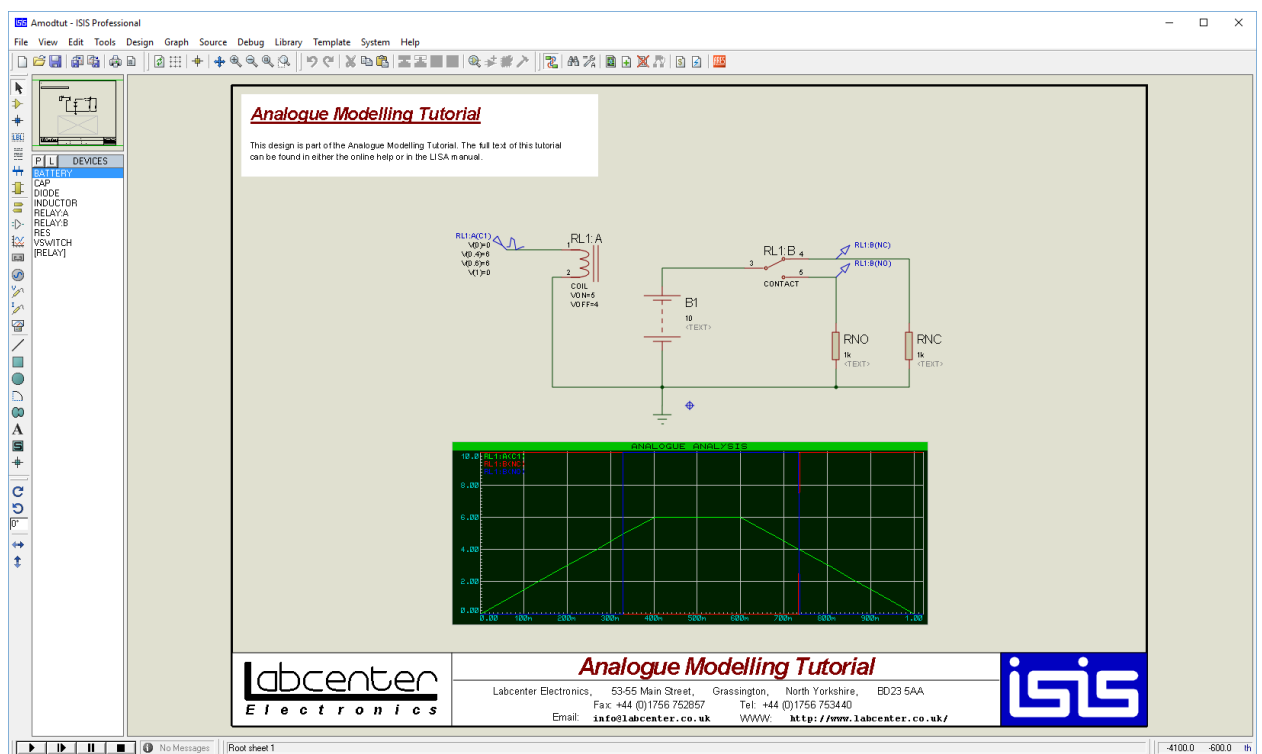


В.Н. Гололобов

Proteus. По мотивам примеров к программе.



Москва - 2016

Оглавление

Диод (файл sweep.dsn в папке Graph Based Simulation) и RC цепь	3
Выводы и дополнения	11
Моделирование RC цепи	13
Выходные характеристики транзистора (Transfer.dsn)	18
Как получены графики в этом примере.....	18
Зачем нужны статические характеристики	21
Ещё раз RC цепь как фильтр (файл Lpf.dsn).....	25
Преобразователь напряжения (файл Boost.dsn v.8)	30
Цифровые схемы (файл Counter5.dsn)	33
Моделирование смешанных схем (файлы ADC0808, DAC0808).....	35
Пример готового устройства с разводкой платы (файл Dbell.dsn).....	37
Работа с микроконтроллерами в ISIS (файл extram.dsn)	39
Виртуальные приборы (файлы DemoBoard.dsn, Poweramp.dsn)	43
Наглядные пособия (файлы AC03.dsn, Mvcr09.dsn)	45
Выводы	46
P.S. Так получилось.....	46

Довольно давно я написал небольшой рассказ о программе ISIS (часть Proteus), когда встретил на одном из форумов утверждение, что программа хорошо работает только с цифровыми компонентами. Это утверждение мне показалось несправедливым по отношению к программе, которая вполне успешно работает и с аналоговыми схемами. Позже я использовал в своих рассказах эту программу для иллюстрации того, о чём идёт речь, но не обращался к рассказу о программе по двум причинам.

Первая причина в том, что на форуме kazus.ru автор под ником Halex07 публиковал подробный рассказ о программе: <http://kazus.ru/forums/showthread.php?t=13198>

Вторая причина – программа коммерческая и не совсем дешёвая.

Вместе с тем, интерес к программе значительный, и начинающие любители не обойдут её вниманием. Поэтому я хотел бы посоветовать им: разберите примеры, написанные авторами программы для иллюстрации возможностей программы. Можно, например, воспользоваться демо-версией программы для самообразования. Итак.

Диод (файл sweep.dsn в папке Graph Based Simulation) и RC цепь

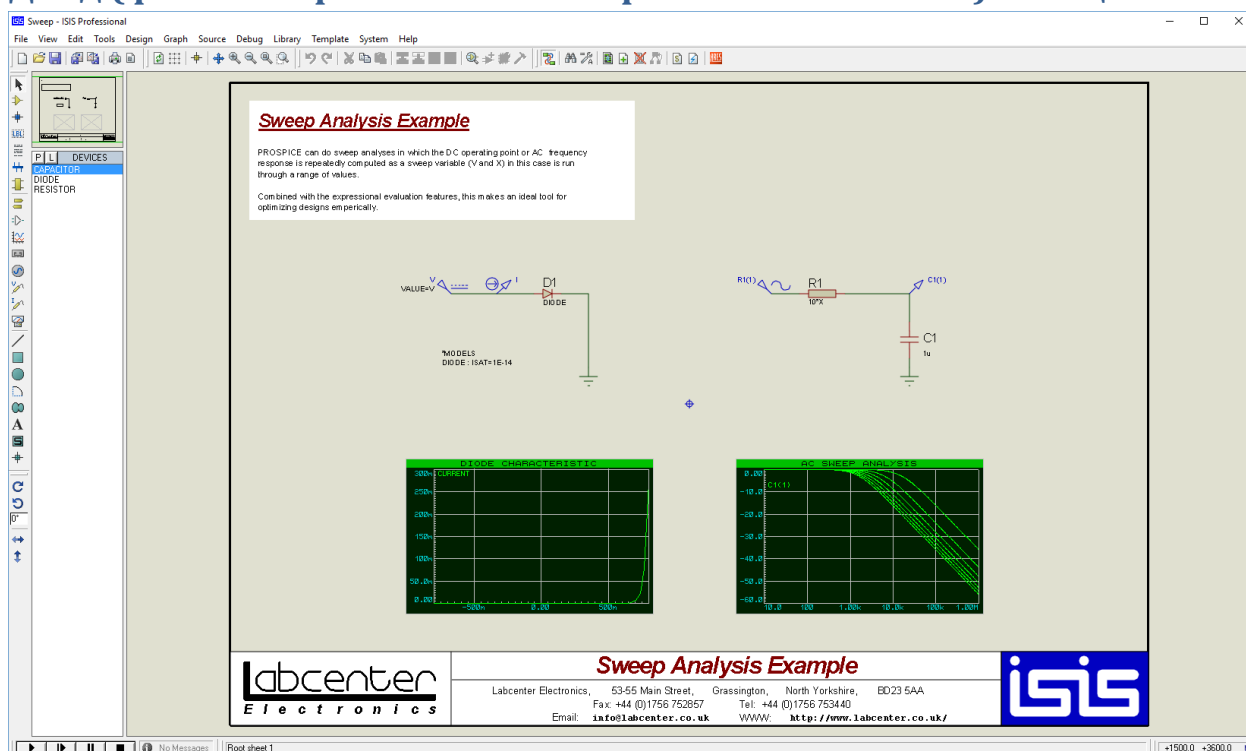
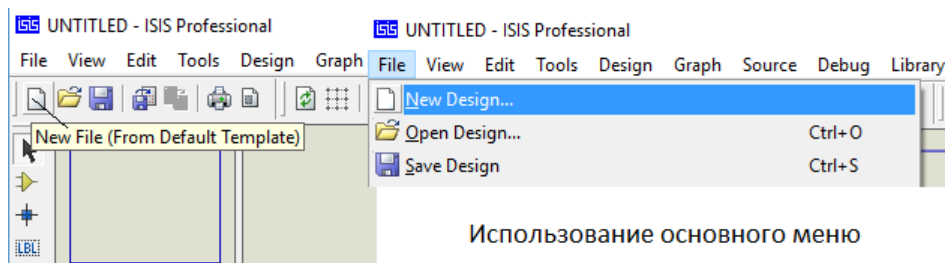


Рис. 1.1. Моделирование вольтамперной характеристики диода

В седьмой версии программы примеры можно найти в разделе Help->Sample Design. Предложение открыть эту директорию с примерами появляется при запуске программы.

Полупроводниковый диод одним из первых нашёл широкое применение в электронике. Диоды часто применяют при выпрямлении переменного напряжения, о чём все знают, но начинающие часто забывают о том, что при маленьких напряжениях...

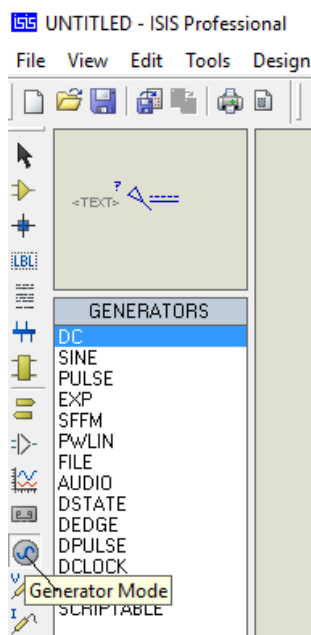
Что происходит с диодом при маленьких напряжениях, мы разберём, используя пример из набора Sample Design. Можно воспользоваться готовым примером, но полезнее будет повторить его самостоятельно. Если вы открыли папку с примерами, где нашли указанный выше файл моделирования, то вам нужно создать новый файл, используя иконку на инструментальной панели или основное меню:



Иконка на панели

Рис. 1.2. Создание нового файла в ISIS

Рабочее поле программы в новом файле пока пусто, нам предстоит заполнить его необходимыми компонентами. И начнём мы с добавления генератора постоянного напряжения.



Иконка на левой инструментальной панели, отмеченная на рисунке, переводит менеджер компонентов (окошко рядом с панелью) в режим выбора генератора.

Как видно на рисунке, можно использовать множество разных генераторов: и постоянного напряжения, и синусоидальных, и импульсных и т.д. Моделирование работы любой электрической цепи редко обходится без соответствующего генератора, как и в данном случае.

Щелчком левой клавиши мышки по DC мы выбираем этот генератор. Переводим курсор мышки в рабочее поле, где ещё раз щёлкаем левой клавишей мышки. Курсор меняет свой вид, приобретая тот, что показан в верхнем окне. Перемещая курсор в нужное нам место на рабочем поле, мы можем отказаться от этого компонента, если нажмём клавишу **Esc** на клавиатуре, либо можем вставить генератор в чертёж ещё одним щелчком левой клавиши мышки.

Рис. 1.3. Добавление генератора постоянного напряжения (DC)

Чтобы повторить схему из примера, нам нужен диод. Для добавления диода мы воспользуемся иконкой на левой инструментальной панели и кнопкой обращения к библиотеке компонентов.

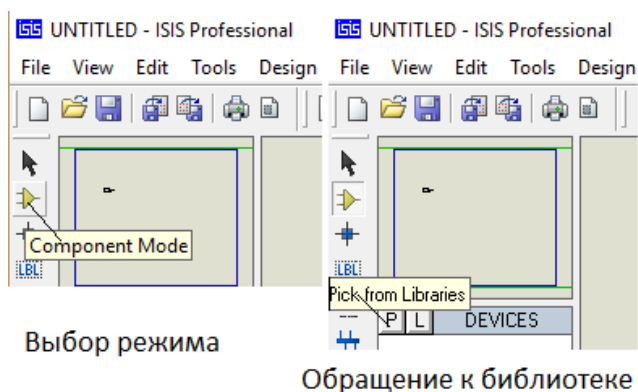


Рис. 1.4. Доступ к библиотеке компонентов в ISIS

Иконка режима выбора компонентов имеет графическое изображение операционного усилителя, чтобы легче было запомнить, что это доступ к компонентам электрической схемы, а кнопка с буквой «Р» открывает окно менеджера всех доступных для использования элементов, список которых вы видите в левом окне:

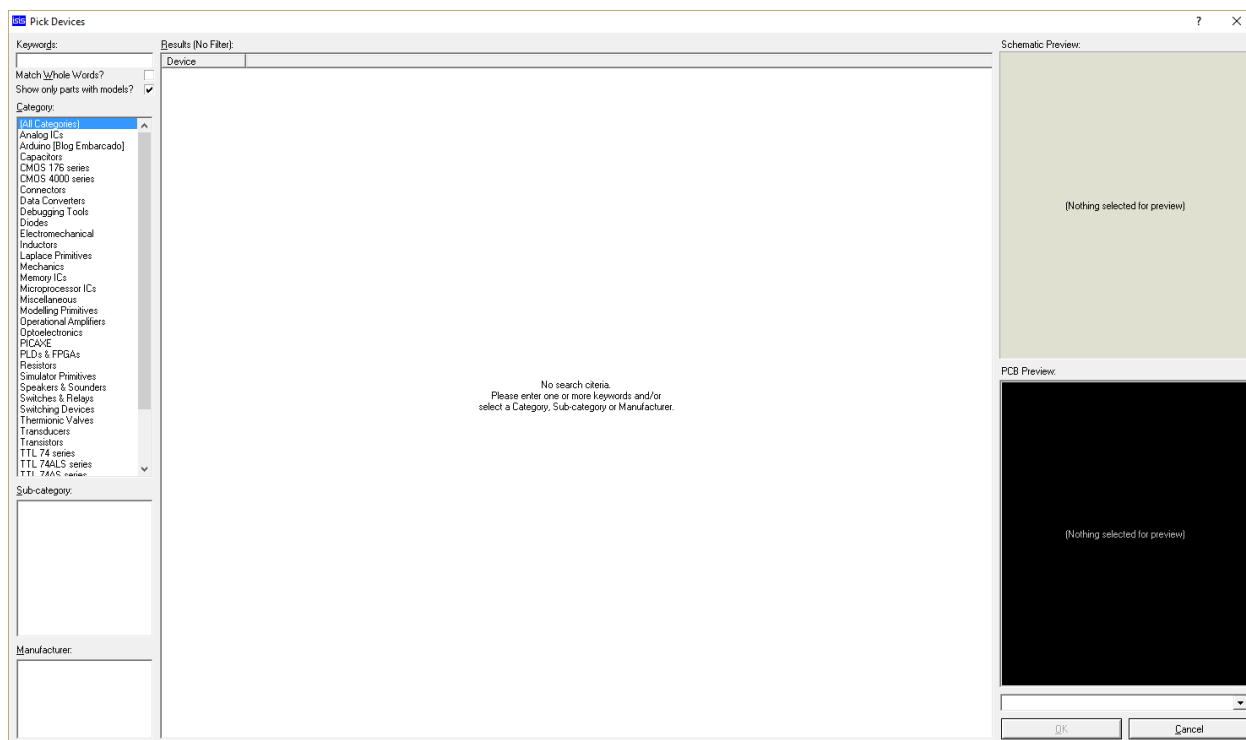


Рис. 1.5. Окно менеджера выбора компонентов

Все компоненты собраны в категории: аналоговые микросхемы, конденсаторы, цифровые микросхемы серии 4000 и т.д. Выберем категорию диодов:

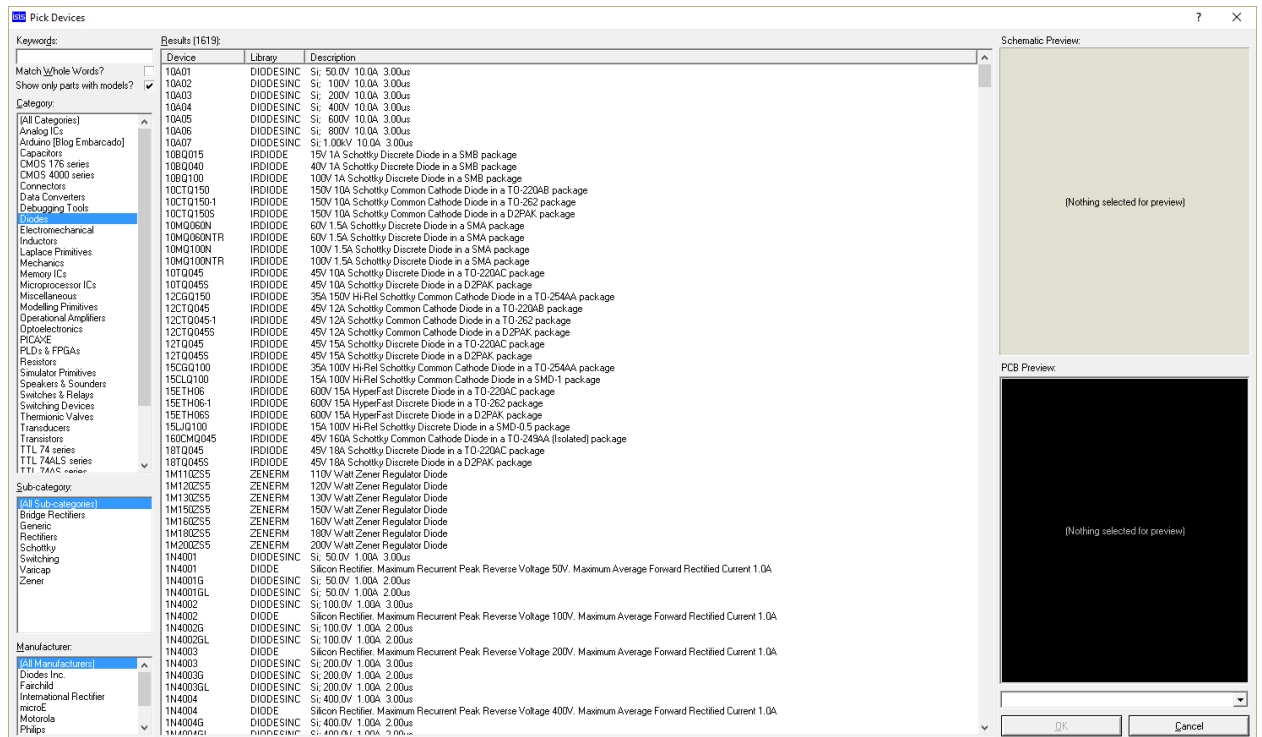


Рис. 1.6. Список диодов в категории Diodes

В большом окне справа появляются все диоды, которые можно использовать (или почти все), список которых так велик, что не умещается в окне – используйте полосу прокрутки справа, чтобы просмотреть весь список. Но, если вы знаете, какой диод вам нужен, вы можете ввести модель в окошко поиска:

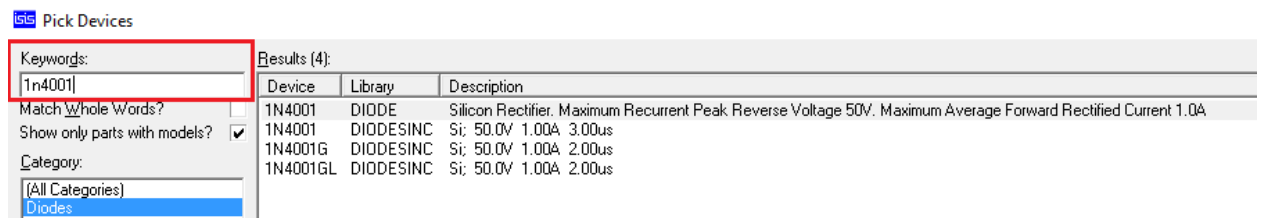


Рис. 1.7. Поиск конкретной модели компонента

Такой приём часто оказывается более удобным при поиске, чем просмотр всех компонентов. Выделив нужный компонент щелчком левой клавиши мышки, нажмите кнопку ОК, чтобы вернуться к чертежу. Курсор при этом приобретает вид диода:

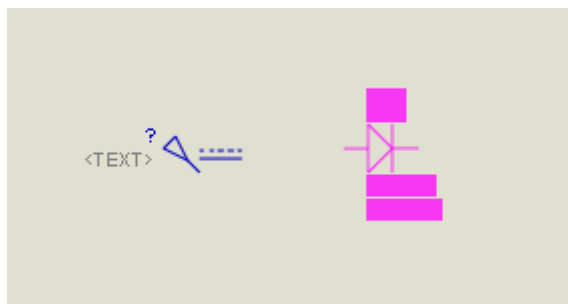


Рис. 1.8. Вид курсора после выбора конкретной модели диода

Щёлкните левой клавишей мышки в нужном месте рабочего поля. Компонент останется на чертеже. Если место выбрано неудачно, используйте возможность для перемещения компонента. Вначале щёлкните по иконке с нарисованным курсором мышки на левой инструментальной панели, затем, щёлкнув левой клавишей мышки над компонентом, скажем, слева, удерживая левую клавишу, проведите вправо вниз, образовав выделение компонента:

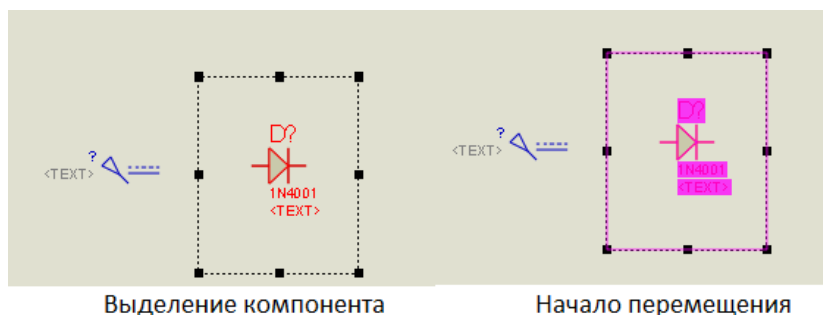


Рис. 1.9. Выделение компонента и начало перемещения

Поместив курсор мышки на выделение, нажмите левую клавишу мышки и, удерживая её, переместите компонент в нужное место. Последний компонент, который нам нужен – это земля (общий провод, относительно которого, как правило, проводят измерения). Щёлкните правой клавишей мышки в свободном месте рабочего поля программы и из выпадающего меню выберите GROUND.

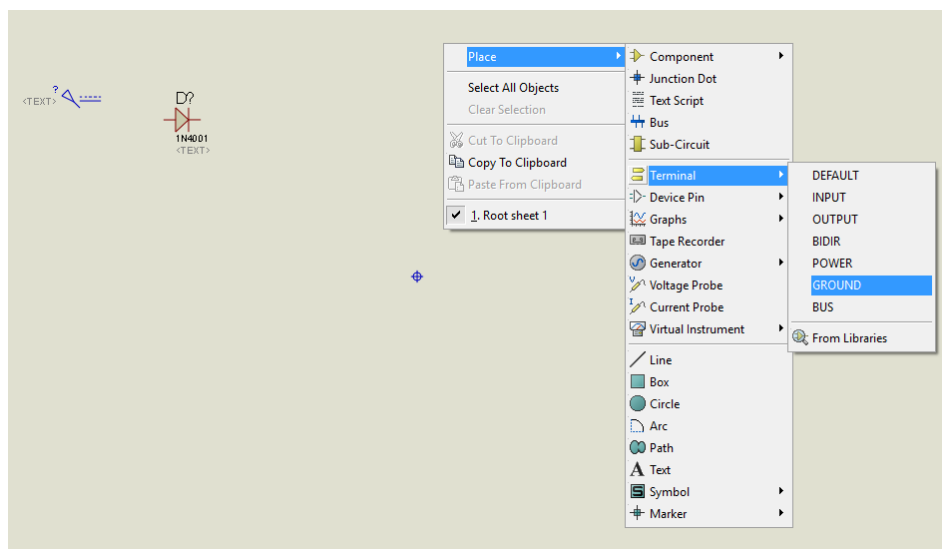


Рис. 1.10. Добавление земли к чертежу

Этот компонент, как и остальные, появится в виде изменившего свой вид курсора, который можно переместить в нужное место, где щёлкнуть левой клавишей мышки.

Если теперь подвести курсор к выводам компонентов, у них появится красная точка, а курсор мышки превратится в карандаш. Щёлкните левой клавишей мышки по точке соединения, переместите курсор к другой точке, где щёлкните повторно, чтобы соединить цепь.

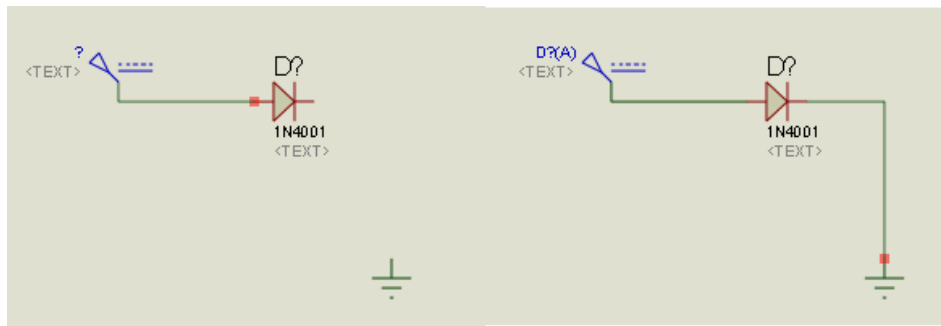
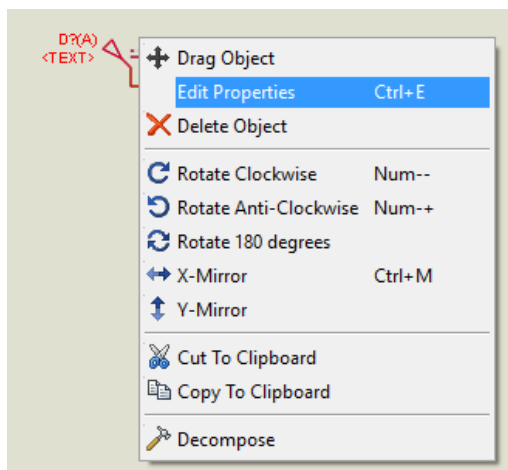


Рис. 1.11. Проведение соединений

Собрав все компоненты и соединив их в цепь, можно перейти к настройке. В данном случае можно пронумеровать компоненты и задать их свойства (там, где нужно). Начнём с генератора.

Щелчком правой клавиши мышки по графическому изображению генератора вызовем выпадающее меню, где можно выбрать раздел свойств компонента.



В выпадающем меню, как можно видеть на картинке, есть ряд полезных команд.

Так можно было раньше использовать команду перемещения Drag Object для изменения места расположения элемента.

При необходимости можно повернуть компонент командами Rotate, что часто бывает необходимо для конденсаторов и сопротивлений.

Можно также отразить компонент по вертикали или горизонтали, скопировать его или вставить.

Рис. 1.12. Выпадающее меню при щелчке по компоненту

В открывшемся диалоговом окне свойств компонента зададим ему имя V, а щёлкнув по элементу Manual Edit?, введём в качестве значения тоже V. Это понадобится позже.

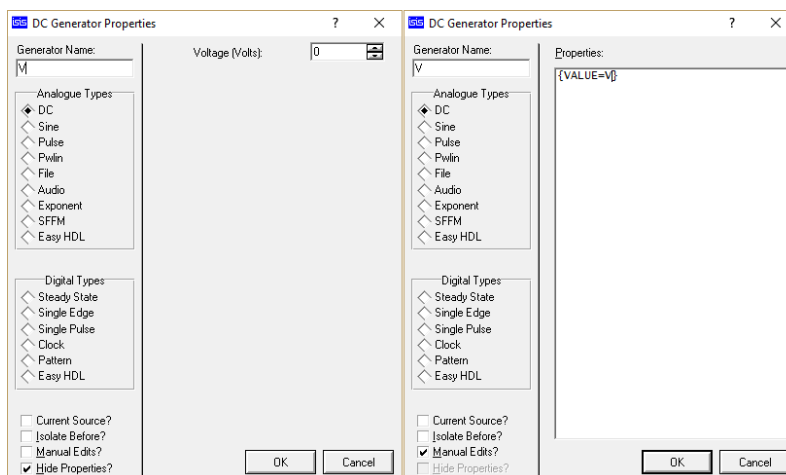


Рис. 1.13. Настройка свойств генератора

Аналогично пронумеруем диод как D1, хотя сейчас это и не обязательно, наверное.

Мы нарисовали электрическую цепь, задали нужные нам свойства для моделирования, что дальше?

Используем иконку режима создания графиков на левой инструментальной панели. И выберем график DC Sweep:

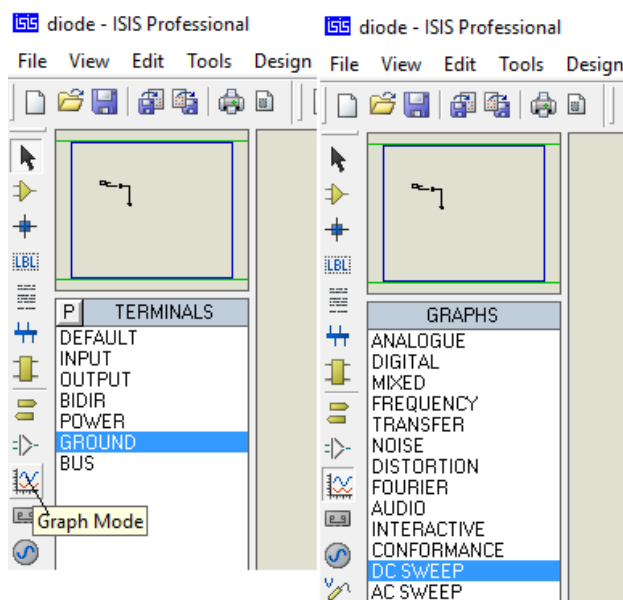


Рис. 1.14. Выбор графика

Как и с компонентами, достаточно щелчка по нужному разделу, чтобы на рабочем поле щёлкнуть левой клавишей мышки в свободной области, провести курсор к правому нижнему углу будущего графика, где щёлкнуть ещё раз, и получить:

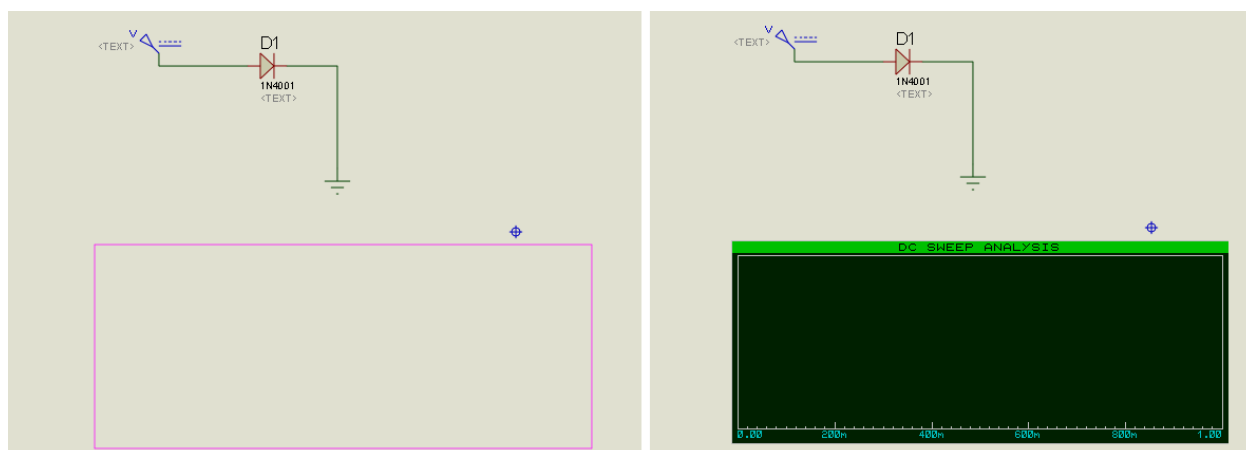


Рис. 1.15. Создание графика для результатов моделирования

Теперь нам следует задать, что мы собираемся измерять, то есть, установить пробник на наш чертёж. Мы собираемся измерять ток, варьируя напряжение от генератора постоянного напряжения, поэтому наш выбор – это пробник тока.

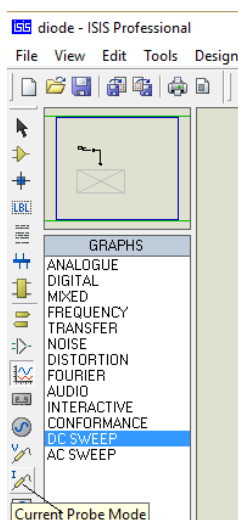


Рис. 1.16. Выбор пробника для измерения тока

Пробник, как всё остальное, переносится после щелчка левой клавишей мышки на чертёж; когда курсор попадает на соединение генератора с диодом, на проводе появляется красная точка, щёлкнув по которой можно установить пробник тока на провод.

Мы создали все необходимые компоненты для проведения опыта. Теперь мы должны настроить график. Правый щелчок мышки по графику вызывает выпадающее меню.

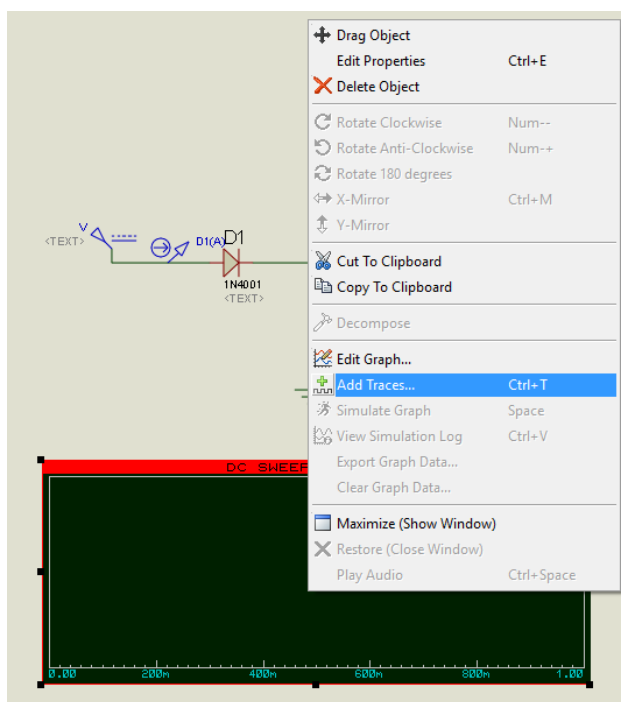


Рис. 1.17. Меню настройки будущего графика

Начнём с добавления кривой (Add Traces). В диалоговом окне введём имя будущей кривой, выберем наш пробник, зададим выражение, как показано ниже, если оно не задано автоматически.

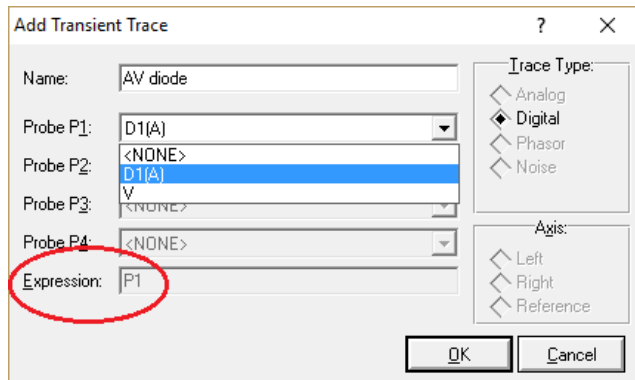


Рис. 1.18. Заполнение диалога для кривой построения

Повторим вызов выпадающего меню, но в этот раз нас будет интересовать диалог Edit Graph, который мы настроим следующим образом:

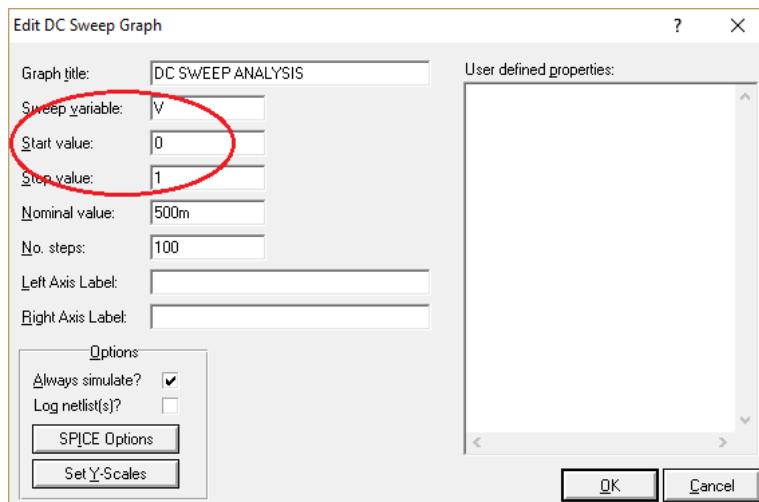


Рис. 1.19. Настройка диалога свойств графика

После нажатия на кнопку **OK** появится сообщение с предложением пересимулировать схему, нажмите кнопку **OK** на этом сообщении. В итоге вы получите график.

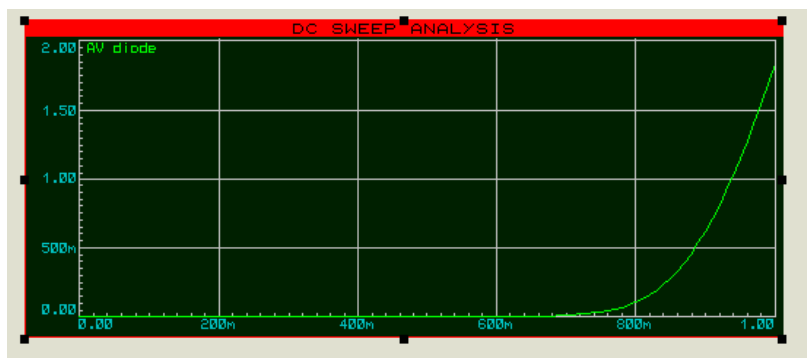


Рис. 1.20. График изменения тока через диод при изменении напряжения

Выводы и дополнения

Для диода 1N4001 ток через диод появляется только тогда, когда падение напряжения на нём больше 0.6 В, что нас и интересовало.

Если график трудно читаем, то можно развернуть его на полный экран. Из выпадающего меню (после щелчка правой клавишей мышки по графику) выберем соответствующий раздел.

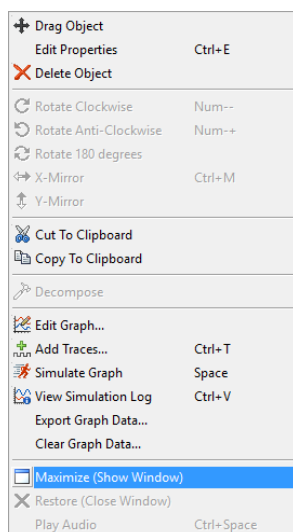


Рис. 1.21. Команда раскрытия графика на полный экран

График, раскрытый на весь экран, имеет собственное основное меню, что увеличивает возможности по его использованию. В частности, в разделе Options этого меню можно выбрать раздел Set Graph Colours, где изменить цвета графика, скажем, сделав белый фон.

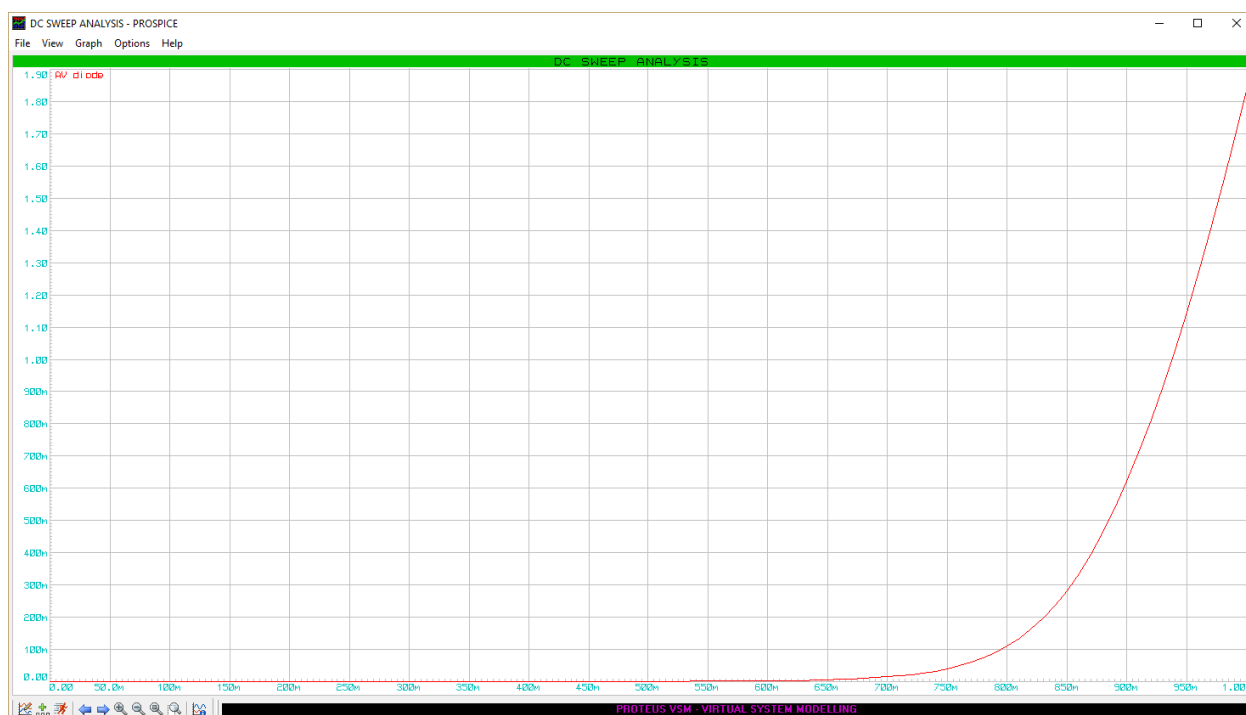


Рис. 1.22. График с изменённым фоном и цветом

График можно сохранить в нескольких графических форматах, используя File->Export Graphics.

Моделирование RC цепи

Со второй электрической цепью этого примера мы поступим иначе – выделим RC цепь, в выпадающем меню выберем Copy To Clipboard, создадим шаблон для новой схемы, щёлкнем правой клавишей мышки в свободном месте рабочего поля и вставим электрическую цепь.

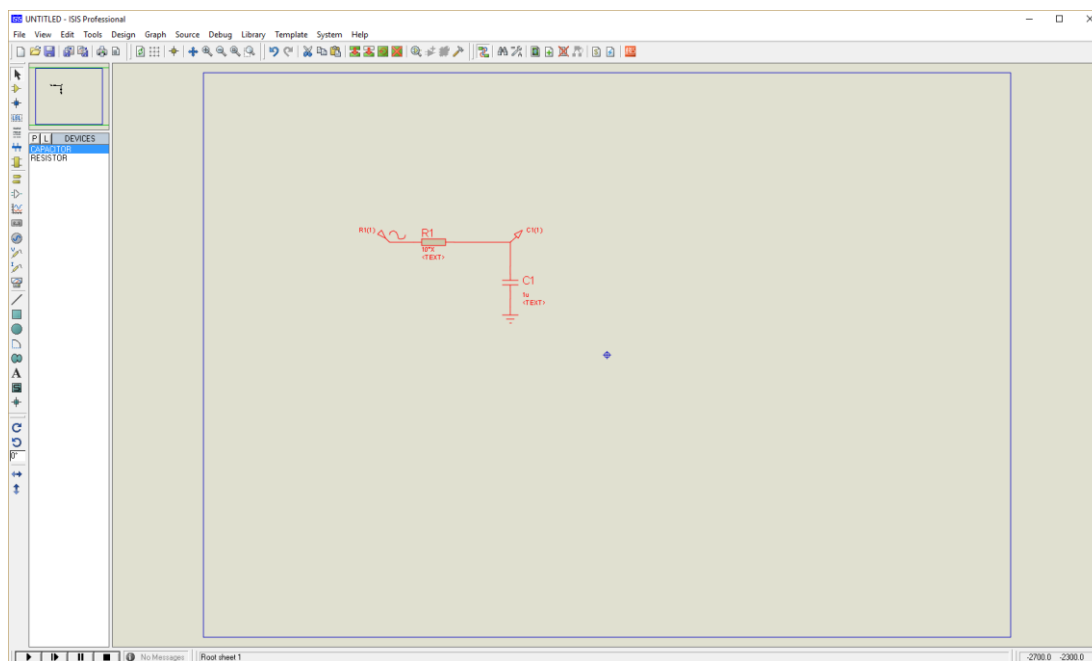


Рис. 1.23. Копирование схемы из примера

Для получения графика проделаем то же, что сделали раньше, но выберем AC Sweep. В качестве пробника в данном случае используется пробник напряжения, C1(1) на рисунке выше.

При развёртке параметра (или качании, Sweep) в предыдущем случае мы использовали напряжение генератора, задав его значение с помощью параметра V.

В данном случае, если присмотреться, величина резистора R1 задана как $10 \cdot X$. Если посмотреть на свойства графика (Edit Graph в выпадающем меню), то можно увидеть, как изменяется значение сопротивления.

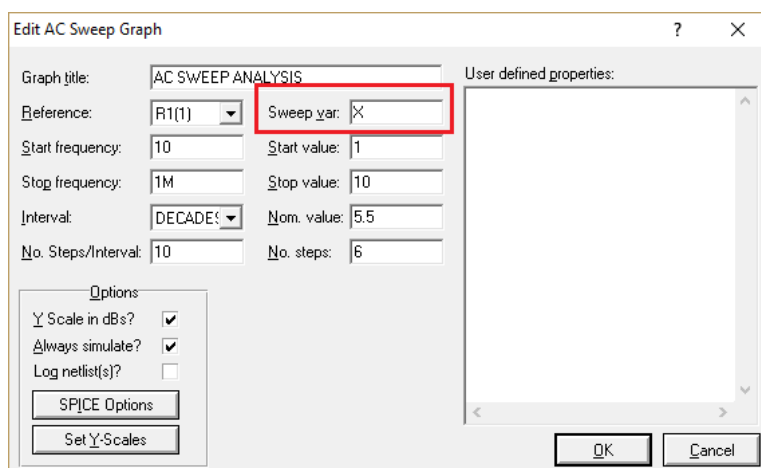


Рис. 1.24. Настройка графика

После симуляции можно наблюдать, как изменяется амплитудно-частотная характеристика RC цепи при изменении сопротивления R1.

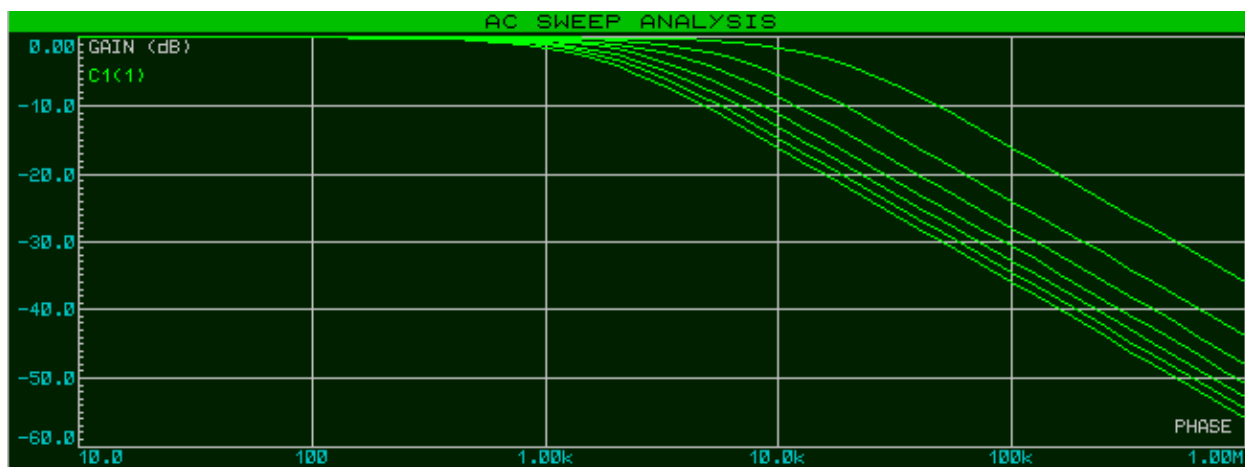


Рис. 1.25. График результата моделирования

RC цепь этого эксперимента называют интегрирующей цепью, верхняя граничная частота которой зависит от величины R и C. Вы можете повторить эксперимент, варьируя не величину сопротивления, а величину конденсатора.

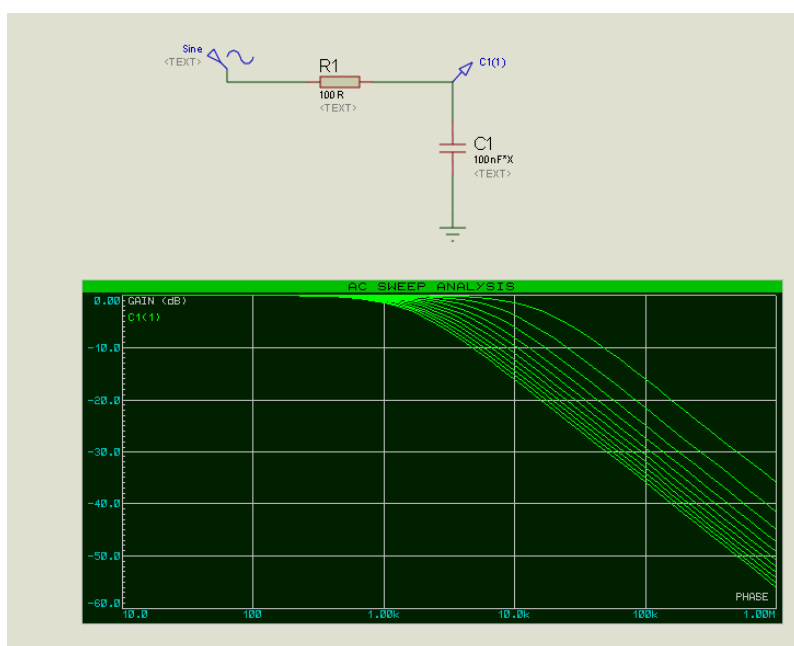


Рис. 1.26. Моделирование RC цепи при варьировании ёмкости конденсатора

Если нас интересует не развёртка параметра, а частотные характеристики RC цепи, то можно изменить значения резистора и конденсатора, сделав их постоянными величинами, и выбрать для графика изображение частотных характеристик.

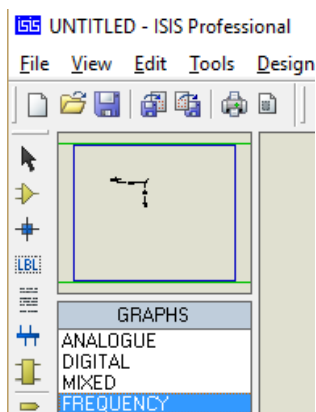


Рис. 1.27. Выбор графика для результатов моделирования

После выделения места для графика...

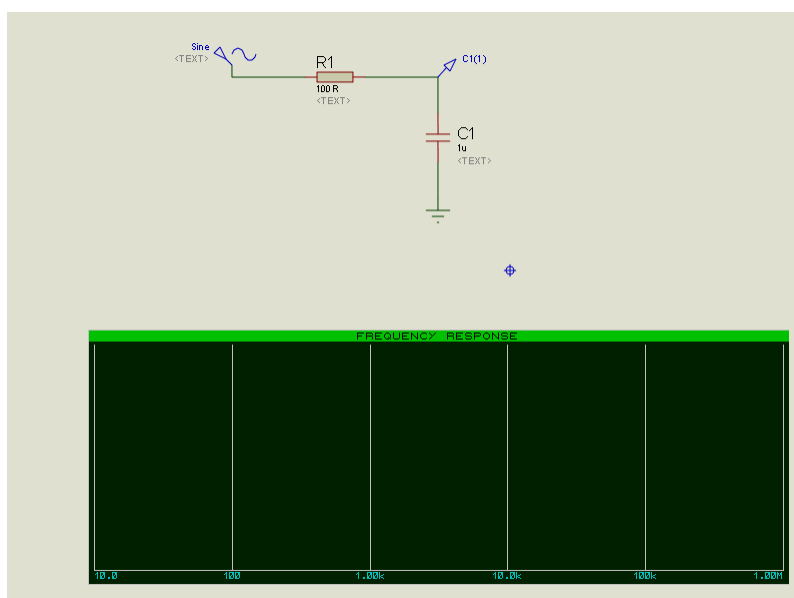


Рис. 1.28. Добавление графика к чертежу

...можно настроить график. Вначале зададим кривую для отображения на графике.

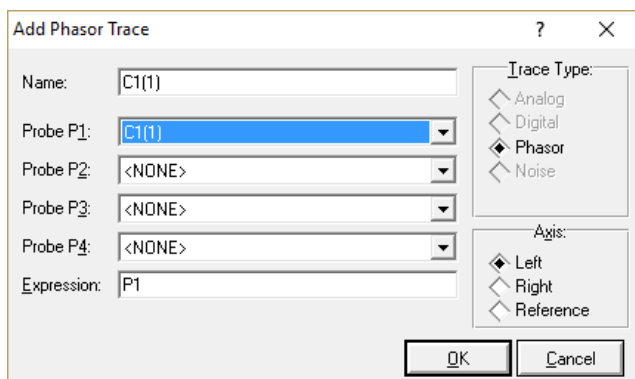


Рис. 1.29. Добавление кривой

Первый график, который мы получим после симуляции – это амплитудно-частотная характеристика RC цепи.

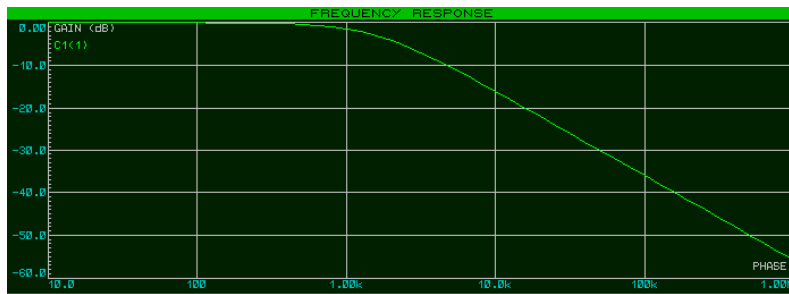


Рис. 1.30. АЧХ интегрирующей RC цепи

Зависимость передаточной характеристики от частоты изображена в относительных единицах, децибелах. Причина этого в том, что был установлен флажок в свойствах графика:

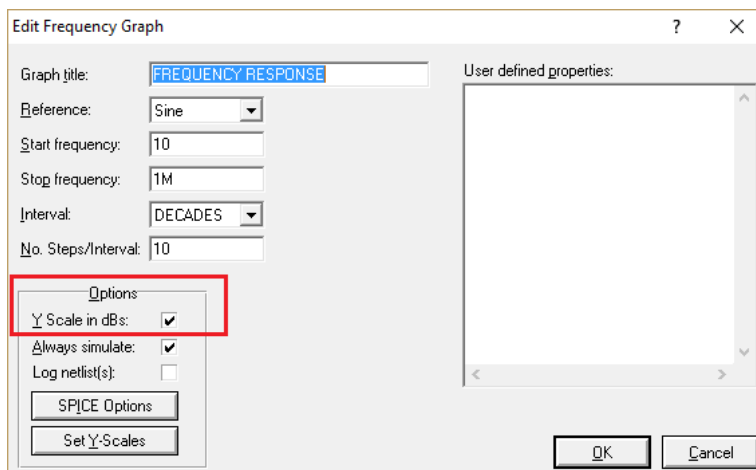


Рис. 1.31. Диалоговое окно свойств графика

Кроме этого графика мы можем получить второй график частотной характеристики – это график фазово-частотной характеристики. Для этого добавим вторую кривую, используя команду Add Traces.

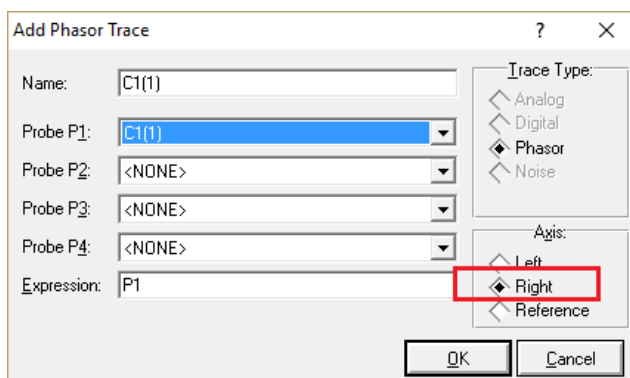


Рис. 1.32. Добавление второй кривой на график

Обратите внимание на выделенный на рисунке фрагмент, где обозначено, что будет использована правая ось. После симуляции (выпадающее меню, Simulate Graph) мы получим вторую кривую.



Рис. 1.33. Графики АЧХ и ФЧХ интегрирующей RC цепи

Граничная частота определяется по уровню -3дБ, при этом спад частотной характеристики происходит со скоростью 20 дБ/декаду. Фазово-частотная характеристика показывает, что на частоте среза фаза изменяется на 45 градусов. Это свойство изменения фазы часто используют на практике. Если развернуть график на весь экран, то можно использовать маркеры для определения величин.

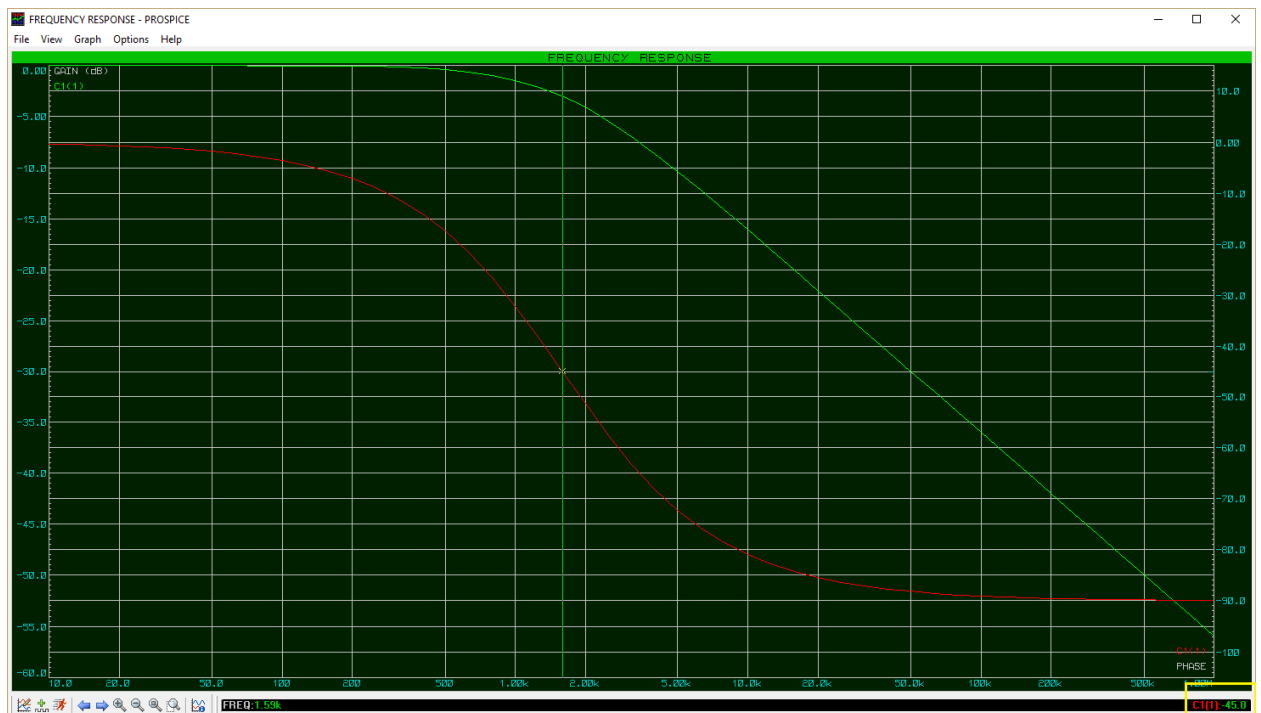


Рис. 1.34. Использование маркеров

В правом нижнем углу жёлтым отмечено значение в -45 градусов. Чтобы получить нужное значение, достаточно щёлкнуть левой клавишей мышки, указав курсором нужную кривую. Теперь, «подцепив» курсором маркер, его можно перемещать по графику.

А свойства RC цепи интересны ещё и тем, что аналогичные частотные характеристики имеет транзисторный каскад усиления с общим эмиттером. Вот сколько интересного можно найти только в одном примере.

Выходные характеристики транзистора (Transfer.dsn)

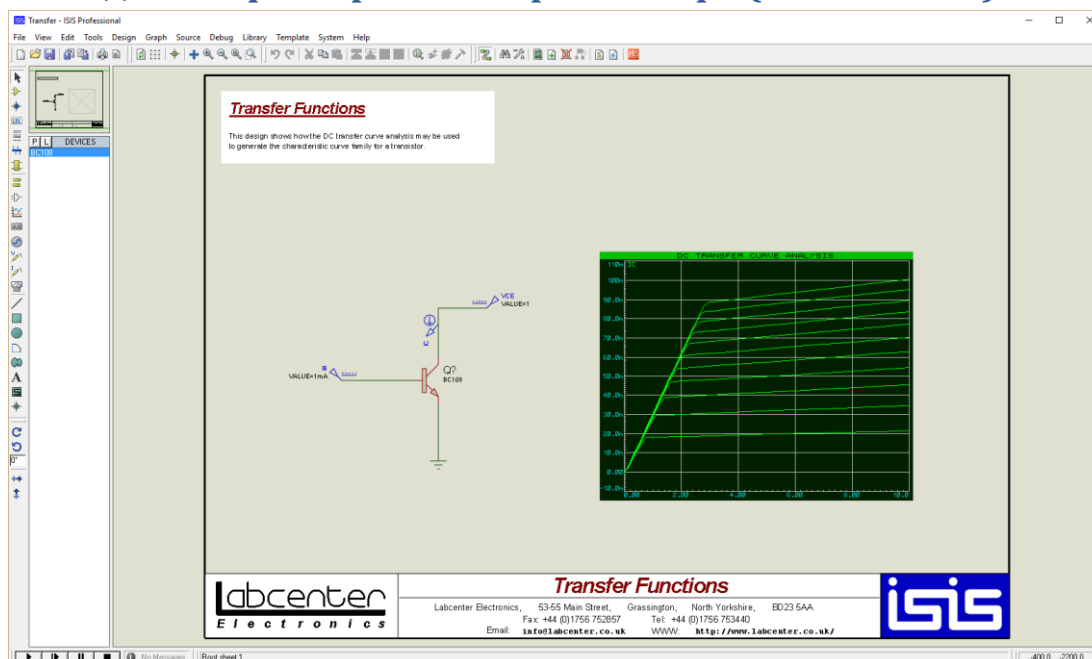


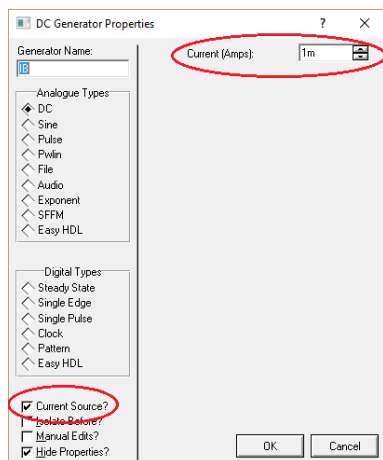
Рис. 2.1. Пример характеристик транзисторов

Не знаю, как сейчас, но когда-то в учебный курс входило проектирование транзисторных усилителей с использованием статических характеристик. Это, положим, не обеспечивало точных данных, но позволяло наглядно понять многое из «жизни транзисторов в электронике».

Подобные характеристики раньше можно было найти в справочнике по транзисторам, в более поздних изданиях, видимо, это стало не актуально, этих графиков нет. Хотя в некоторых справках к транзисторам (datasheet) их можно увидеть.

Как получены графики в этом примере

Где, а главное как, найти компоненты, изображённые на чертеже выше, мы обстоятельно рассмотрели ранее. Не будем пока повторяться. Однако отметим, что генератор IB в базе транзистора имеет следующие свойства:



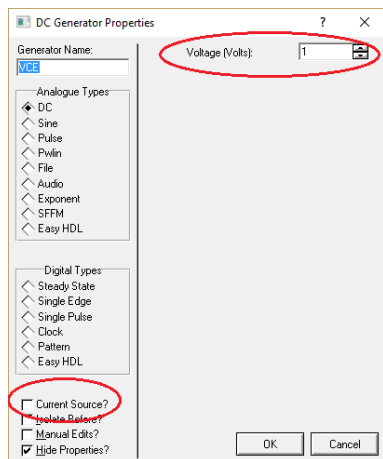
Флажок, установленный в окошке Current Source?, превращает генератор постоянного напряжения в генератор тока.

Поэтому исходное значение можно задать в виде определённого тока, а не напряжения.

Рис. 2.2. Настройки генератора IB

Используя генератор постоянного тока для питания базы транзистора, мы при моделировании избавляемся от необходимости добавлять к схеме резистор, что обычно можно наблюдать для схемы транзисторного усилителя с общим эмиттером.

Для сравнения посмотрите на свойства второго генератора (источника) VCE:



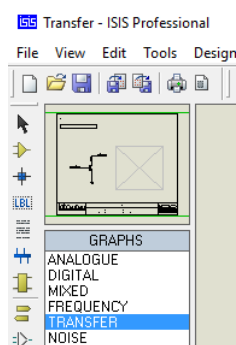
Сняв флажок в окошке Current Source?, мы превращаем источник в генератор постоянного напряжения.

Теперь исходное значение задаётся как 1 вольт.

Отметьте себе этот факт на будущее. Когда вам нужен источник тока, вы знаете, что нужно сделать!

Рис. 2.3. Настройка генератора VCE

Если в первом примере мы использовали качание (развёртку, sweep) параметра, то изменение параметров в этом примере обеспечивается типом анализа при моделировании.



Этот вид анализа относится к использованию передаточной функции.

Вот, что говорит о ней wiki: Передаточная функция — один из способов математического описания динамической системы. Используется в основном в теории управления, связи и цифровой обработке сигналов.

Не будем спорить с этим определением.

Рис. 2.4. Выбор типа анализа

Заглянем в свойства графика, используя щелчок правой клавишей мышки по графику и выбрав из выпадающего меню раздел Edit Graph:

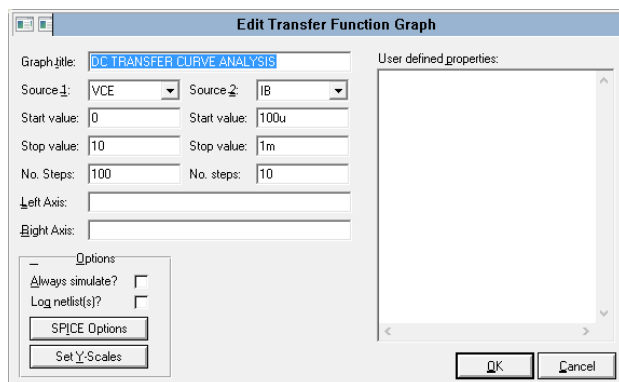


Рис. 2.5. Свойства графика transfer

Этот вид анализа позволяет использовать два источника (VCE и IB в данном случае), для которых заданы начальное и конечное значение и использовано нужное количество точек.

Но если вы удалите из примера график, вновь добавьте этот вид анализа, если настроите свойства по рисунку, приведённому выше, а затем запустите симуляцию, то...

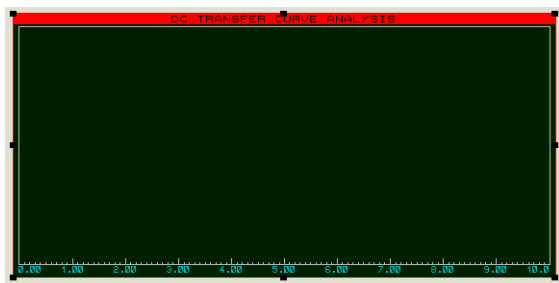


Рис. 2.6. Неудачная симуляция

Можно было бы и не показывать этот результат, которого нет. Но каждый раз, когда мы забываем что-то сделать, мы получаем похожий результат, а в неудаче склонны обвинять программу или её создателей.

Мы не указали программе, какие кривые нас интересуют. Используем пункт Add Traces выпадающего меню, чтобы указать, что нам нужно:

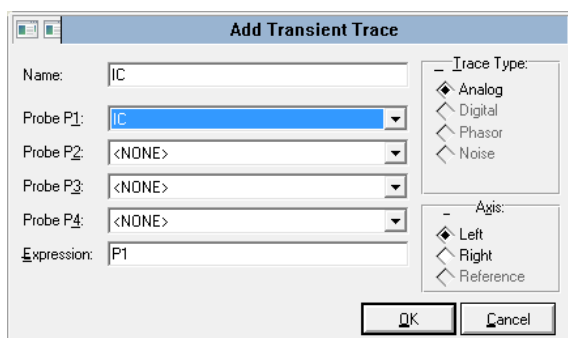


Рис. 2.7. Добавление графика в выбранный тип анализа

Нас, конечно, интересует ток коллектора IC. И теперь симуляция даёт желаемый результат:

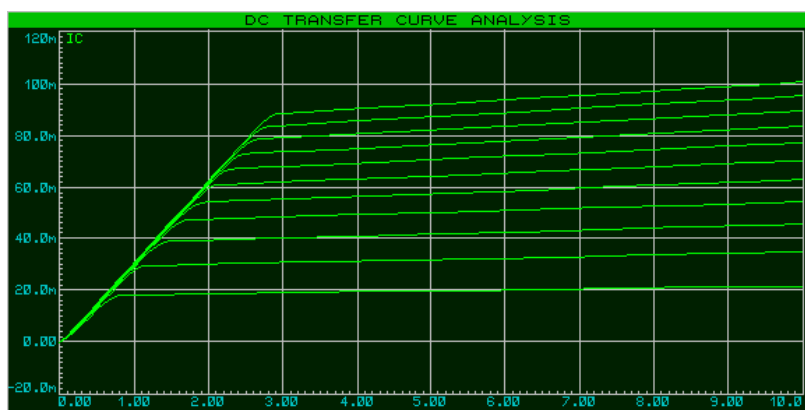


Рис. 2.8. Окончательный результат симуляции

Зачем нужны статические характеристики

Получив статические характеристики, мы получили представление о возможностях программы, что было первой целью данного эксперимента. Но что полезного можно обнаружить в этом примере?

Я, конечно, не помню, что и как мы когда-то делали в годы учёбы, но некоторое представление о пользе статических характеристик можно получить, если...

Если, используя варьирование тока базы, построить график зависимости напряжения база-эмиттер от тока базы.

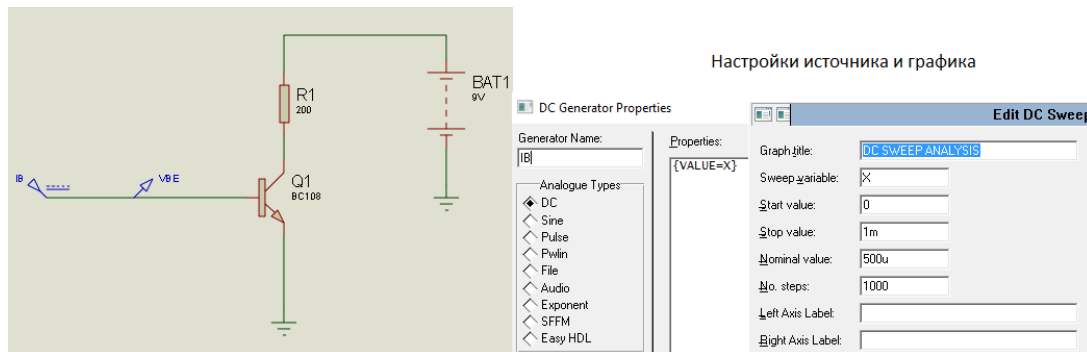


Рис. 2.9. Схема опыта для получения зависимости напряжения VBE от тока IB

Выбрав в качестве кривой для построения графика VBE, мы получим такой график:

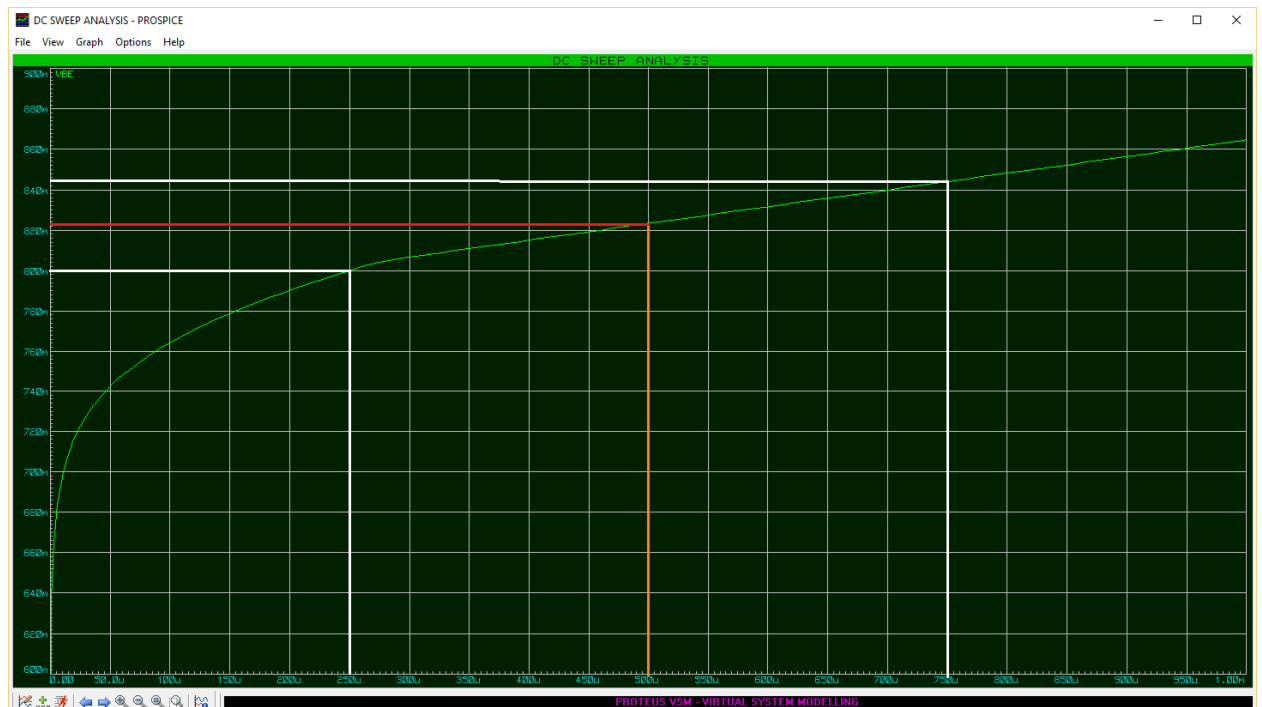


Рис. 2.10. График зависимости напряжения база-эмиттер от тока базы

Более привычным график выглядел бы как зависимость тока базы от напряжения база-эмиттер, не так ли?

Мы предполагаем усилить симметричный сигнал, поэтому выбираем линейный участок характеристики для линейного преобразования сигнала. Позже мы уточним данные, а сейчас определим изменение тока базы от 250 мкА до 750 мкА, выберем середину – 500 мкА.

Теперь построим график зависимости тока коллектора от тока базы, используя вариацию тока базы и выбрав кривую для тока коллектора.

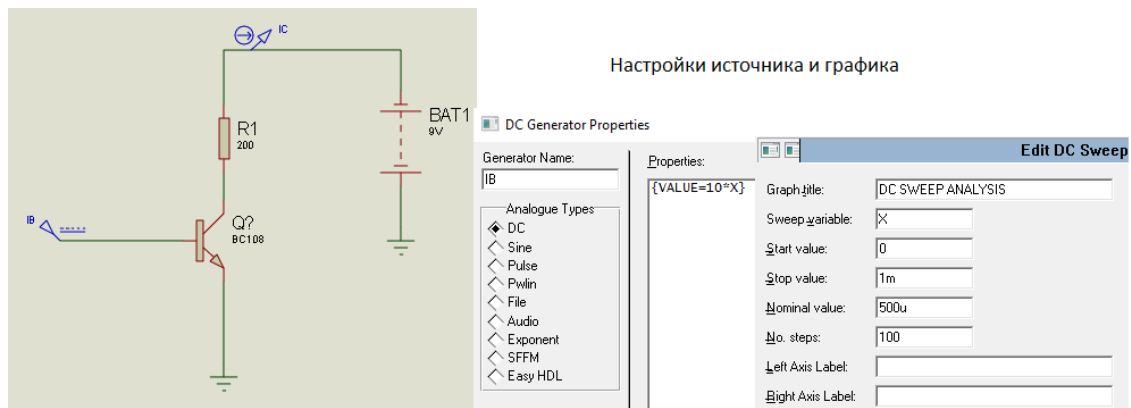


Рис. 2.11. Схема для построения зависимости тока коллектора от тока базы

График зависимости тока коллектора от тока базы в заданных пределах выглядит так:

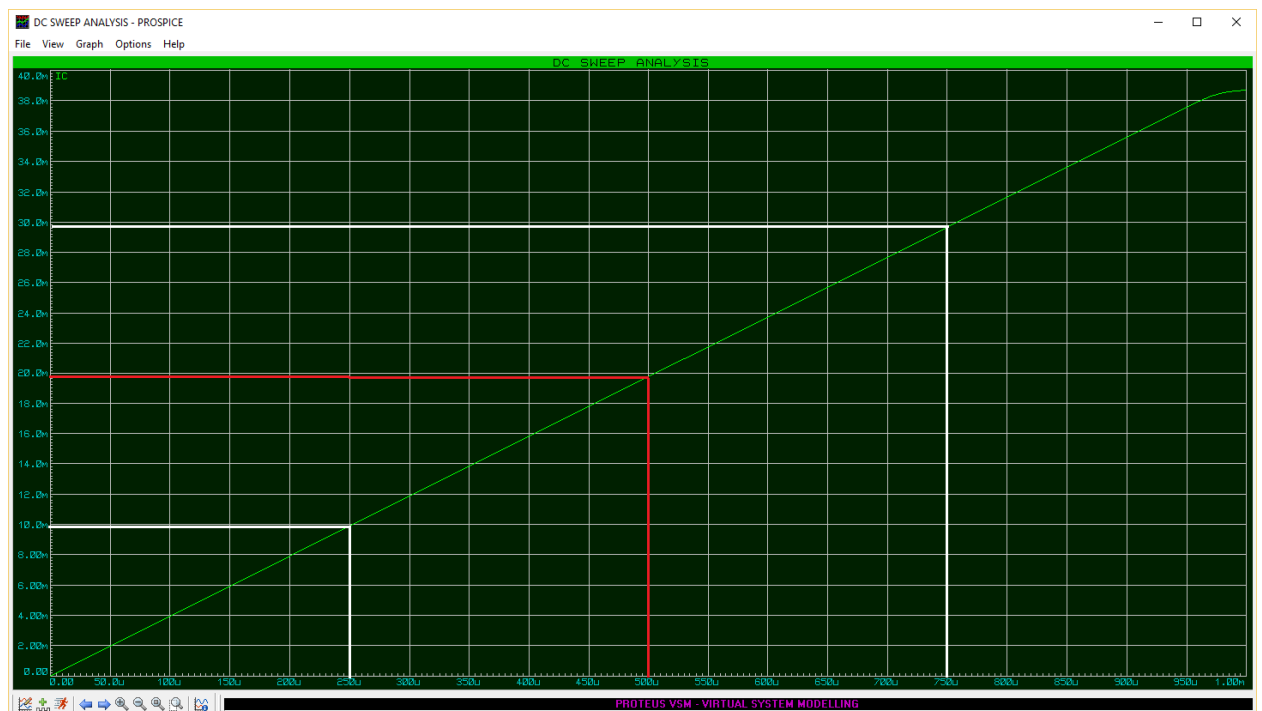


Рис. 2.12. Зависимость тока коллектора от тока базы

Изменим настройки исходной программы так, чтобы было удобно работать с графиком.

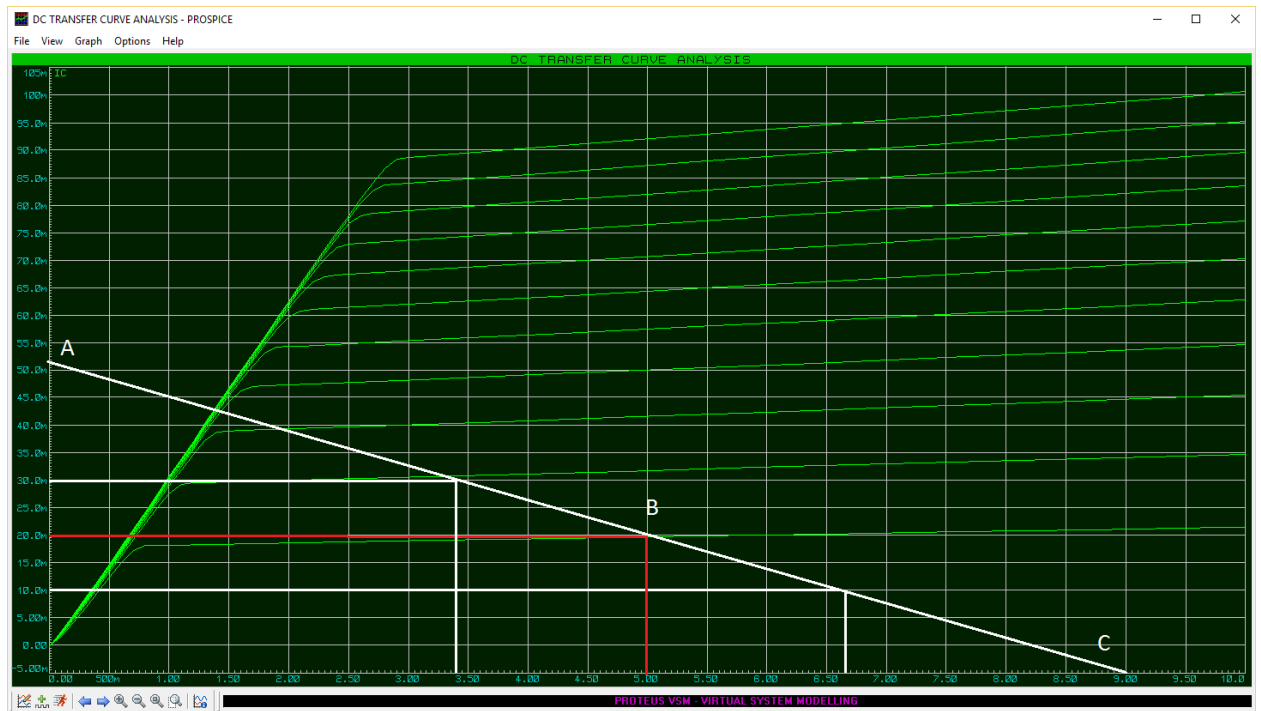


Рис. 2.13. Графики переходной характеристики

Проведём прямую от точки 20 мА на оси IC до пересечения с линией фиксированного тока базы в точке В. Соединим точку 9 В на оси VCE с точкой В и продолжим до пересечения с осью IC. Эту прямую называют нагрузочной прямой.

Я не уверен в правильности своих построений, и вам, если для вас это важно, следует прочитать об этом в каком-либо учебнике, но смысл графических построений можно проследить. Обратимся к рисунку 2.10. Амплитуда входного сигнала, V_{BE} , около 20 мВ. Амплитуда выходного сигнала (рис. 2.13) около 1.5 В. Усиление по напряжению будет равно их отношению и составит 75.

Входная статическая характеристика позволяет графическим методом определить, например, такой параметр, применяемый при расчётах с малыми сигналами, как h_{11} . В графическом варианте он означает отношение малого приращения напряжения V_{BE} к малому приращению тока базы. Для получения этого дифференциального сопротивления изменим настройки графика (рис. 2.10) так, чтобы можно было прочитать нужные значения вблизи рабочей точки 500 мкА, то есть, повторим симуляцию при изменении тока базы от 400 мкА до 600 мкА.

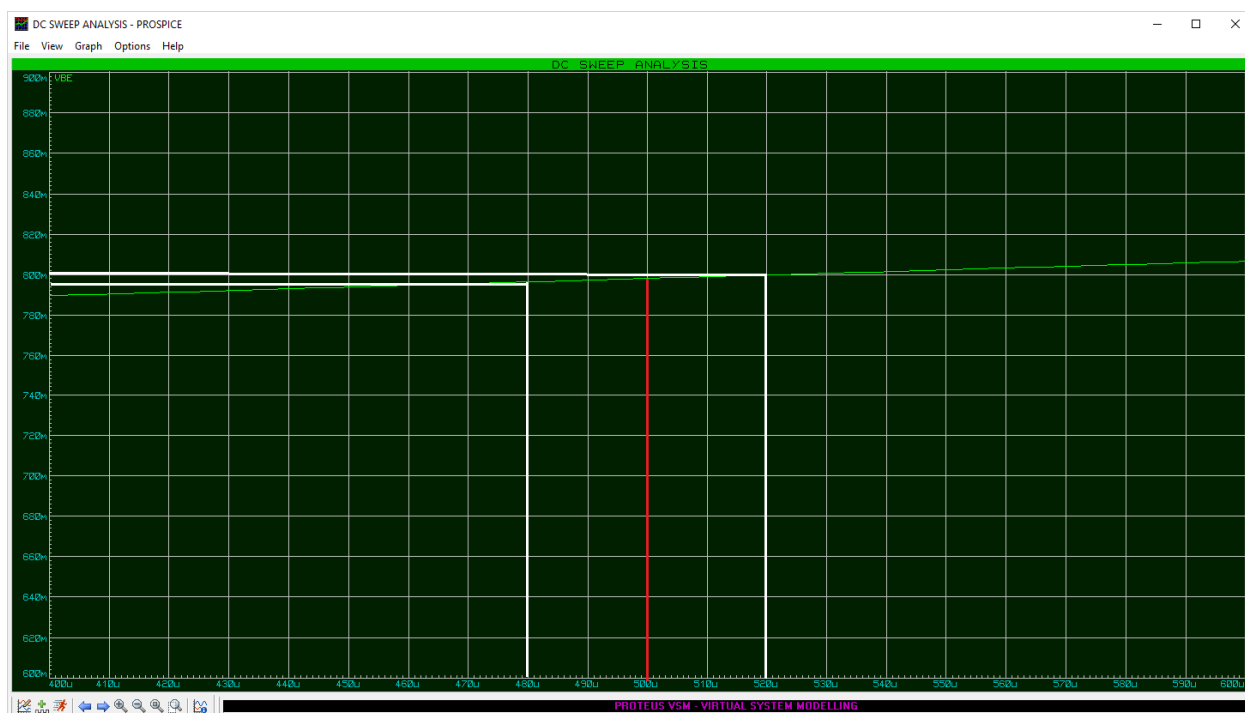


Рис. 2.14. Получение значения h_{11}

Напряжение база-эмиттер изменяется от 796 мВ до 804 мВ, то есть, изменяется на 8 мВ. При этом ток базы изменяется от 480 мкА до 520 мкА, изменяется на 40 мкА. Их отношение и будет искомым параметром h_{11} : $8/0.040 = 200 \text{ Ом}$.

Похожие построения можно провести для получения h_{21} и h_{22} . Для получения h_{12} графические методы не используют, из-за малости этого параметра им, как правило, пренебрегают. Эти параметры тоже не всегда можно найти в справочниках.

Я не думаю, что начинающие радиолюбители будут заниматься показанными выше построениями, но знать, что они возможны в Proteus, я уверен им будет полезно. Потому что, разбираясь с работой той или иной схемы, в её описании можно встретить эти понятия, а наиболее простое их понимание даёт графическое их представление.

Ещё раз RC цепь как фильтр (файл Lpf.dsn)

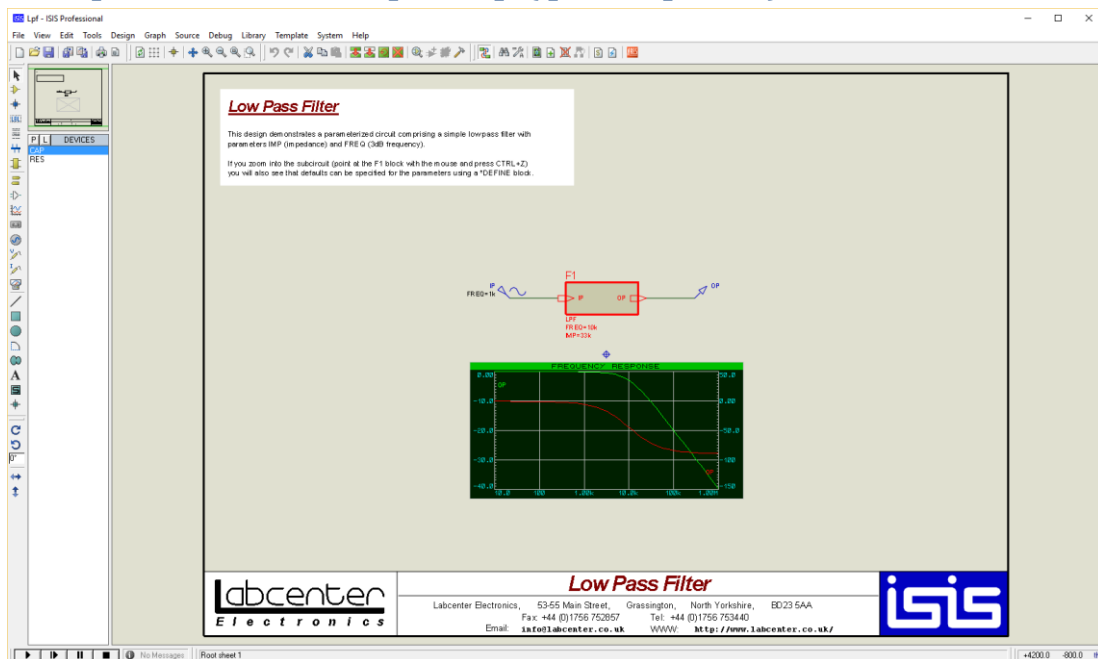


Рис. 3.1. Фильтр низких частот

Выше мы уже рассматривали интегрирующую RC цепь. Она демонстрирует простейший фильтр низких частот. Но фильтры могут быть и очень сложными, особенно тогда, когда это активный фильтр. Вместе с тем, рассматривая работу схемы, нам важно знать назначение и некоторые параметры фильтра. Если фильтр является важным, но второстепенным элементом устройства, его удобно на схеме изобразить в виде функционального блока. Что и демонстрирует этот пример, использующий встроенный в ISIS механизм подсхем.

Обозначенный на рисунке выше компонент – это RC фильтр низких частот. Чтобы посмотреть его исполнение, достаточно щёлкнуть правой клавишей мышки по элементу F1, а из выпадающего меню выбрать переход к производной схеме.

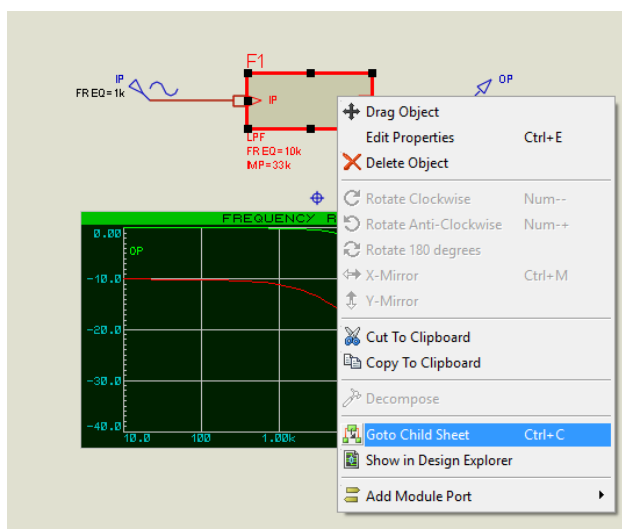


Рис. 3.2. Выбор перехода к подсхеме

Подсхема – это та же схема, в чём можно убедиться:

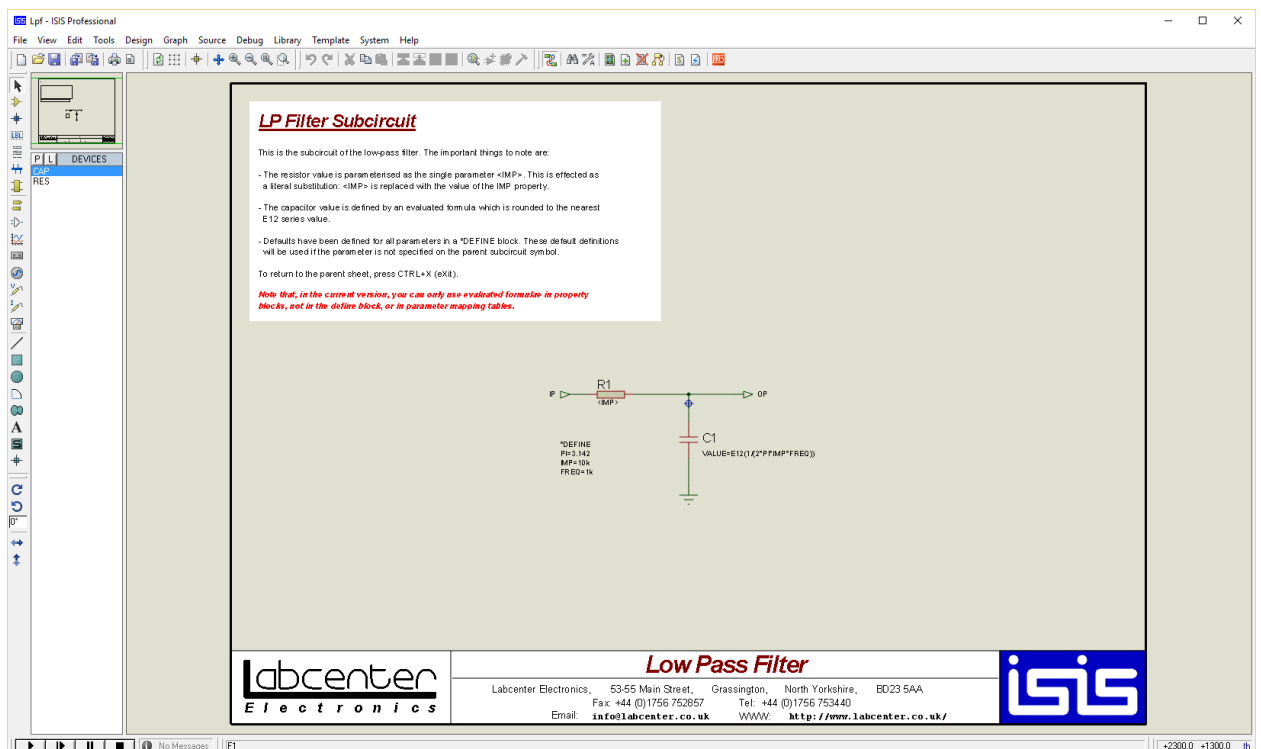


Рис. 3.3. Подсхема фильтра низких частот

Разрабатывая своё устройство, разбираясь с работой устройства по готовой схеме, вы можете уделить достаточно много внимания этому блоку, поскольку с подсхемой можно работать так же, как с любой другой схемой. А, закончив работу с этим блоком, вы можете вернуться к исходной схеме, используя основное меню:

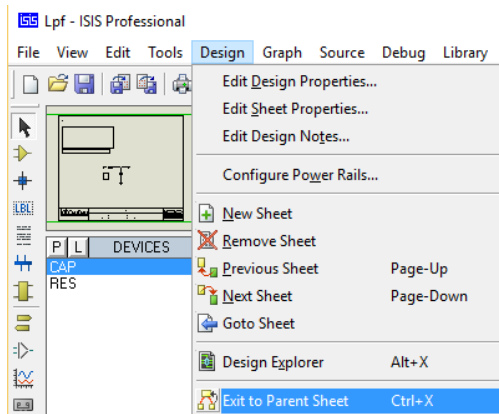


Рис. 3.4. Возвращение к основной схеме

Но как создать подсхему?

Начнём с создания новой схемы. Выберем, например, на инструментальной панели иконку перехода к режиму создания подсхемы.

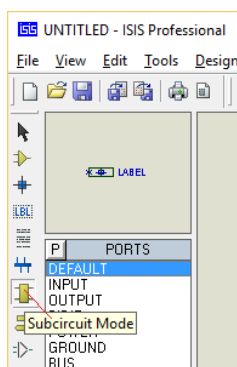


Рис. 3.5. Выбор режим работы с подсхемой

В окошке компонентов выделим DEFAULT. Теперь мы можем нарисовать прямоугольник в рабочем поле чертежа – щелчок левой клавишей мышки, задающий верхний левый угол, переход к правому нижнему углу, где повторный щелчок левой клавишей мышки. Этот прямоугольник и будет нужной нам подсхемой.

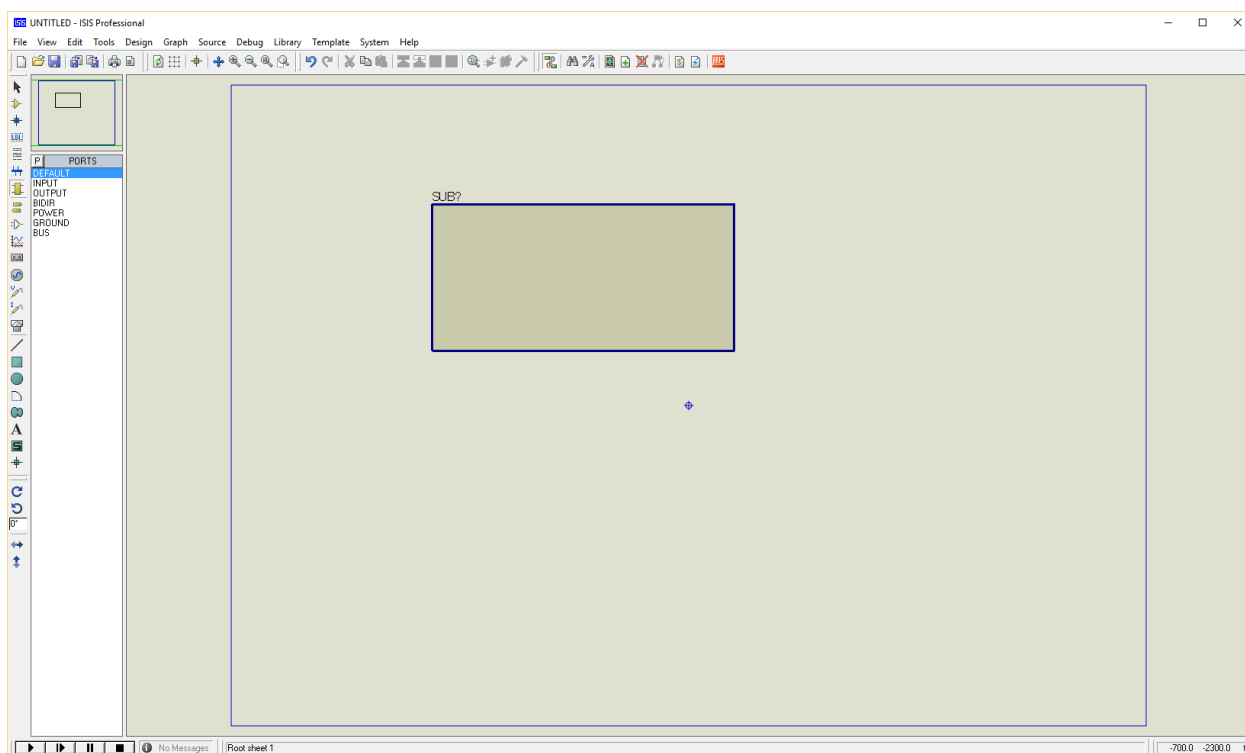


Рис. 3.6. Создание блока подсхемы на основном чертеже

Щелчок правой клавишей мышки по этому блоку и выбор перехода к подсхеме (см. выпадающее меню рис.3.2) даёт новый лист чертежа для нашей подсхемы.

Добавим два порта, используя, скажем, выпадающее меню, вызванное щелчком правой клавиши мыши в любом свободном месте:

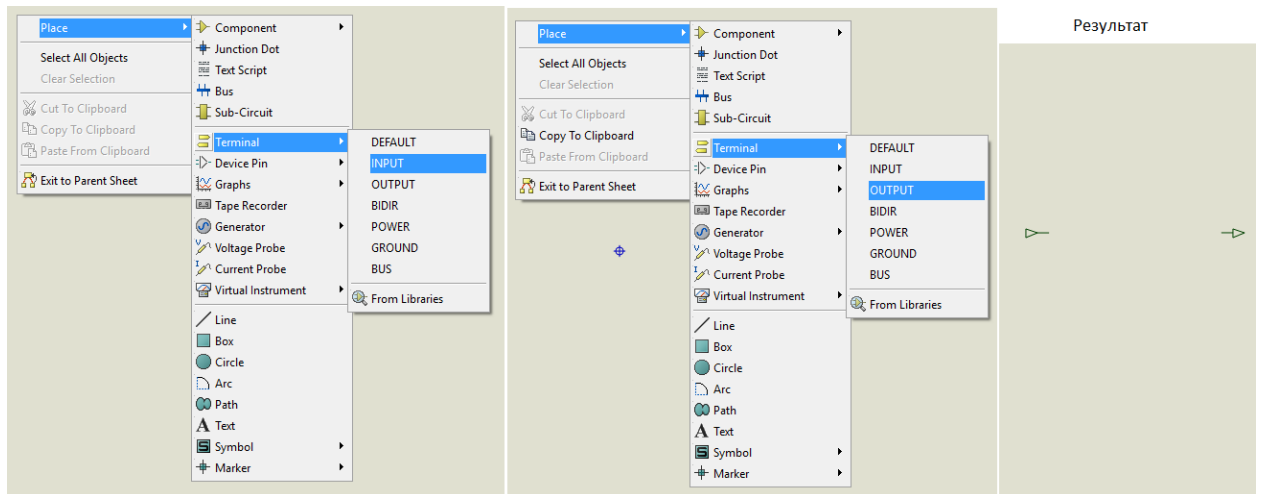


Рис. 3.7. Добавление портов входа и выхода на подсхеме

Щелчком правой клавиши мышки по этим портам откроем выпадающее меню, из которого выберем свойства компонента. В свойствах компонента обозначим входной порт PI, а выходной PO.



Рис. 3.8. Обозначение портов в диалогах свойств

Добавим в схему два резистора по 1 кОм, получив делитель напряжения.

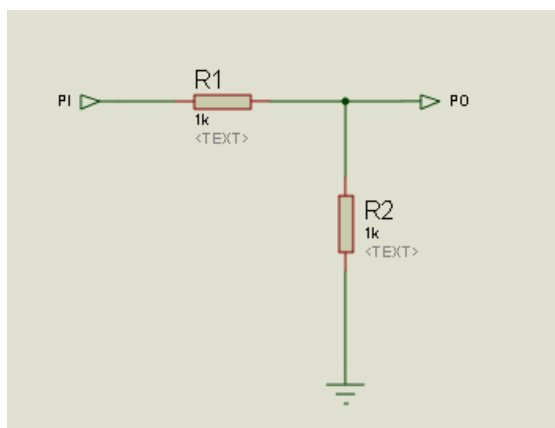


Рис. 3.9. Завершение создания подсхемы

Настало время вернуться к основной схеме, используя основное меню, как мы сделали на рис. 3.4.

По возвращении мы должны добавить к нашей подсхеме, которую обозначим с помощью диалога свойств как SUB1:

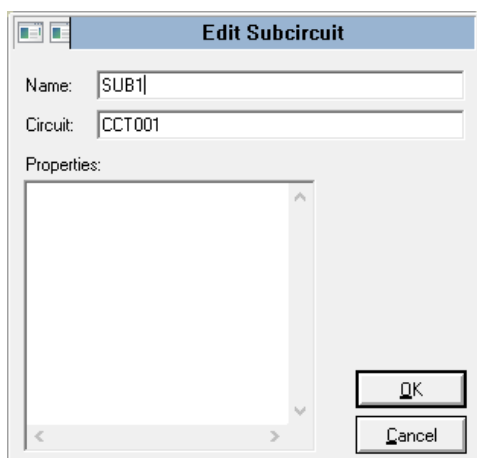


Рис. 3.10. Диалоговое окно свойств подсхемы

Обратите внимание, что ссылка на подсхему уже есть. Выберем иконку режима работы с подсхемой, выделим в окне компонентов элемент INPUT, переместим курсор мышки к левой границе подсхемы, где увидим на конце курсора маленький крестик, и щёлкнем левой клавишей мышки, добавляя вход в подсхему. Аналогично, выбрав элемент OUTPUT, добавим выход на правую границу подсхемы. Теперь мы можем подключить источник постоянного напряжения 10 В к входу, а к выходу вольтметр. И, наконец, запустить моделирование...

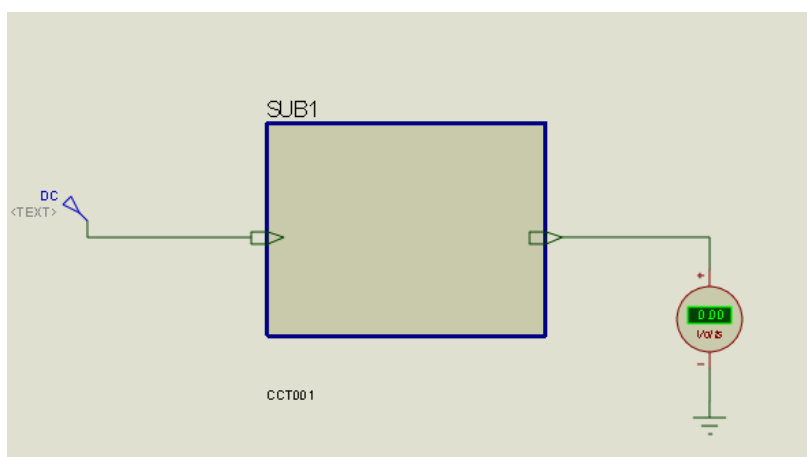


Рис. 3.11. Первое моделирование схемы

...чтобы убедиться – первый блин комом. Что не так?

Конечно, мы добавили вход и выход, но не связали их с входом и выходом подсхемы. Открыв свойства этих элементов, обозначим их так же, как они обозначены на подсхеме. Теперь моделирование даёт нужный результат.

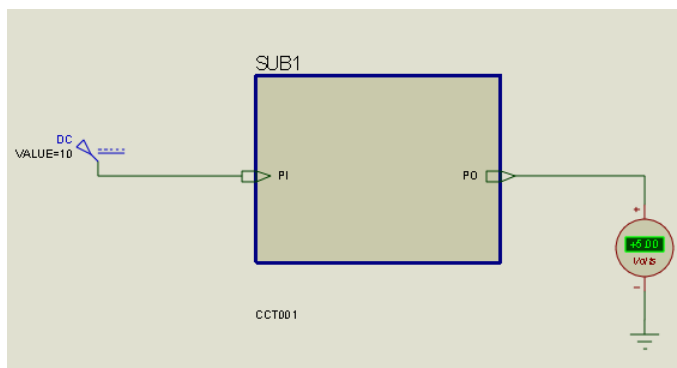


Рис. 3.12. Моделирование схемы после исправления недочётов

Конечно, можно доверять моему утверждению об удобстве использования подсхем, но можно и не доверять. Я хочу привести пример в защиту своего мнения.

Вы нашли схему прибора, работу которого проверили с помощью программы ISIS. Всё работает, всё хорошо. Но... ох, уж эти НО!

Питание устройства от 12 В удобно в машине, где вы можете подключить его к аккумулятору, но носить в походе с собой аккумулятор... вам эта идея не нравится. Однако вы знаете, что есть преобразователи, которые позволяют вам получить 12 В от батарейки.

Подобное устройство вы можете добавить в качестве подсхемы к схеме своего устройства. Подсхему отладить и проверить, а потом проверить ещё раз и работу всего устройства.

Преобразователь напряжения (файл Boost.dsn v.8)

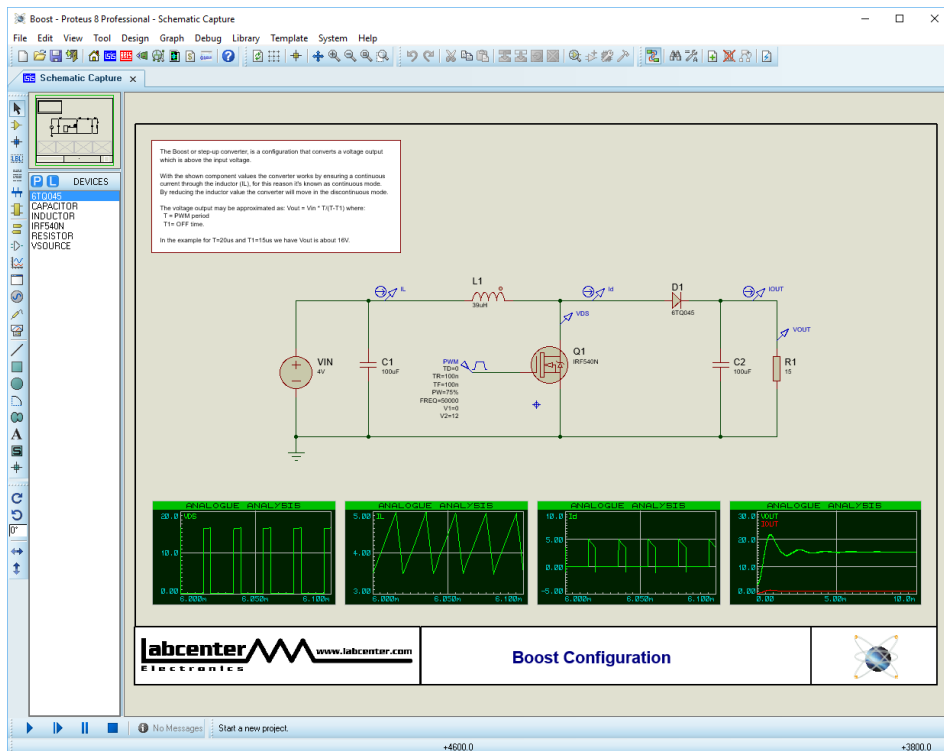


Рис. 4.1. Пример преобразователя напряжения

Это пример из восьмой версии программы Proteus, но ничто не мешает собрать эту схему в седьмой версии, чтобы проверить работу преобразователя.

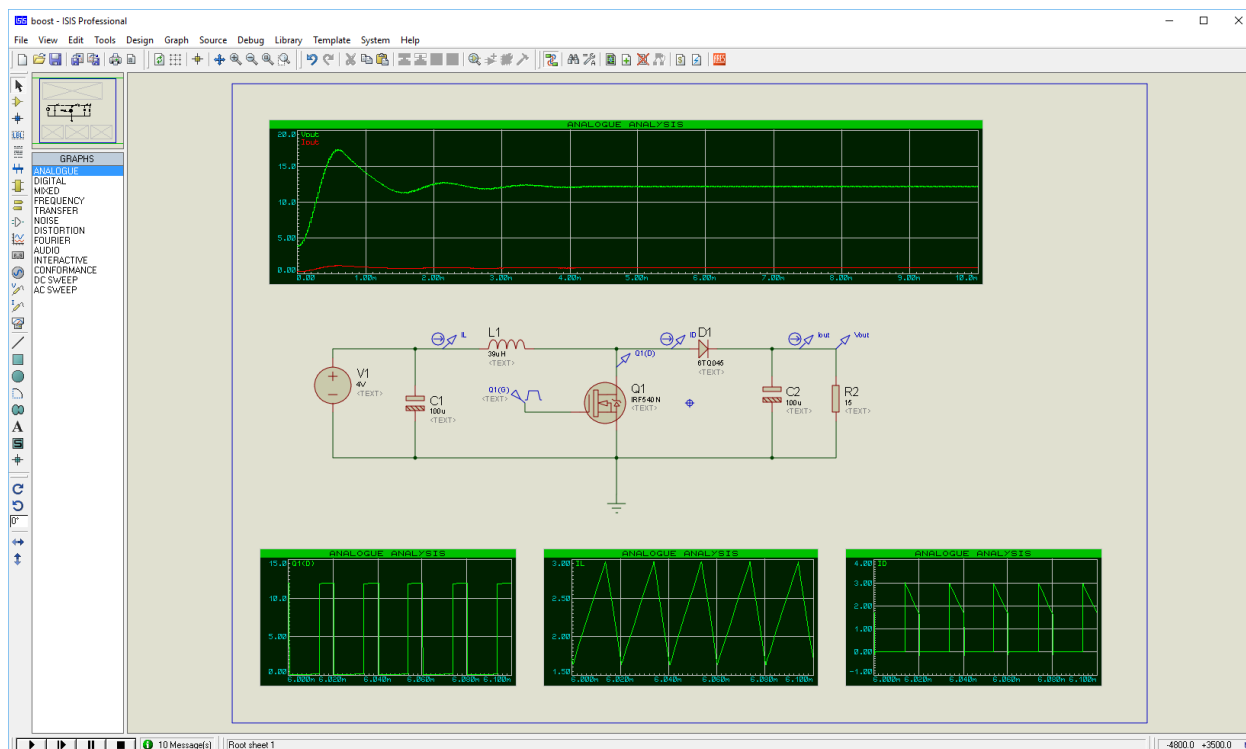


Рис. 4.2. Вариант преобразователя в седьмой версии

Согласитесь, что такой блок питания удобнее сделать в виде подсхемы, чтобы проверить и отладить схему.

Начать работу со схемой можно с выбора исходного источника питания, батарейки. В приведённом примере использовано напряжения 4 В. У вас есть выбор, например, между 1,5 - 3 В и 9 В. Есть расчётные формулы для разработки преобразователя, но можно попытаться опытным путём переделать схему. Начнём с выяснения того, как работает преобразователь.

Когда транзистор Q1 полностью открыт (первый нижний график, время 6-6.015 мс), ток через индуктивность L1 линейно нарастает (следующий график). Когда транзистор Q1 закрывается, на индуктивности появляется противоЭДС, которая складывается с напряжением на V1, формируя импульс с амплитудой значительно больше, чем напряжение батарейки V1. Этими импульсами напряжения заряжается конденсатор C2 до напряжения 12 В.

От чего зависит напряжение противоЭДС, формирующее нужное нам напряжение?

Уменьшим напряжение V1 до 3 вольт, изменим скважность управляющих импульсов на затворе Q1 от генератора Q1(G). Первый опыт при скважности 60%, второй при скважности 80%.

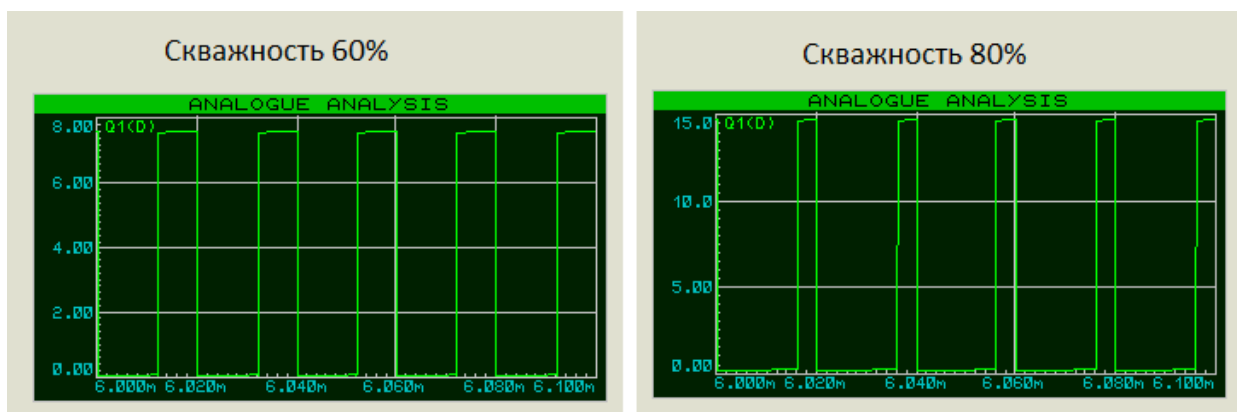


Рис. 4.3. Изменение амплитуды импульсов при изменении скважности управляющих импульсов

Скважность импульсов влияет на амплитуду импульсов, заряжающих конденсатор C2. Оставим скважность импульсов задающего генератора неизменной, но изменим время нарастания и спада импульсов (фронты импульсов).

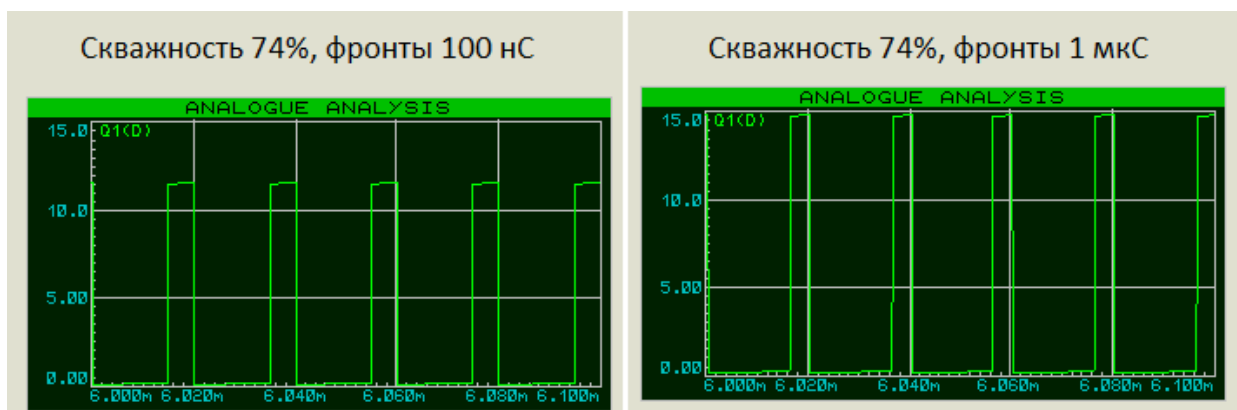


Рис. 4.4. Амплитуды импульсов при изменении скорости нарастания импульсов управления

И, наконец, посмотрим, как влияет сопротивление нагрузки на выходное напряжение.

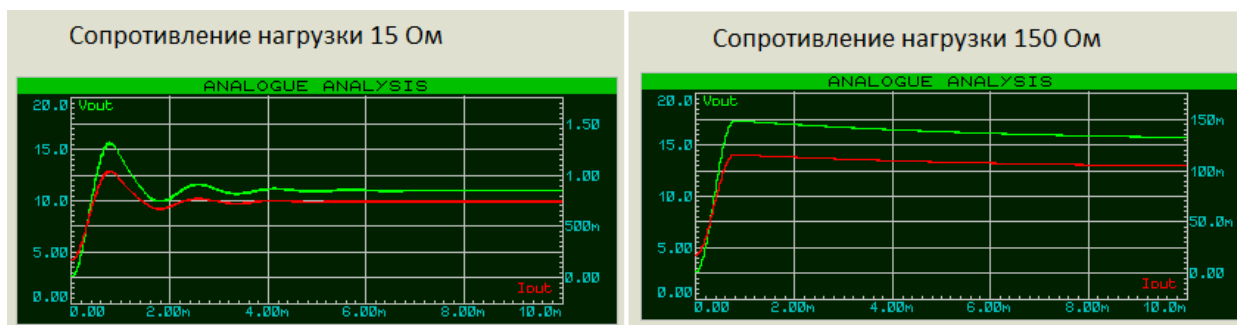


Рис. 4.5. Изменение выходного напряжения при изменении нагрузки

Нижний график – это ток в нагрузке, а правая вертикальная ось показывает величину тока.

В качестве задающего генератора Q1(G) можно использовать таймер 555, который тоже изобразится в виде схемы генератора импульсов с регулируемой скважностью.

Вы, я полагаю, не будете в подобной ситуации создавать преобразователь из отдельных компонентов, а используете подходящую микросхему. Но, согласитесь, что использование подсхемы в данной ситуации – это наиболее разумный подход, о чём, собственно, и речь.

Цифровые схемы (файл Counter5.dsn)

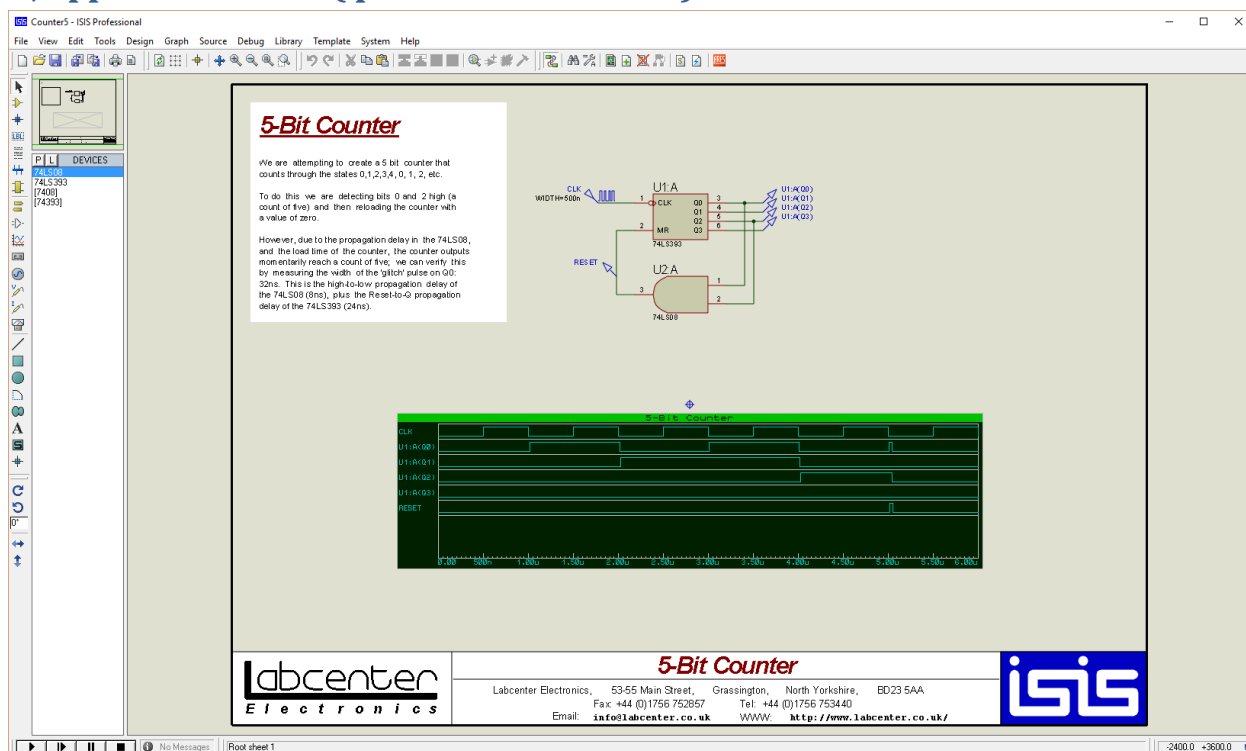


Рис. 5.1. Схема счётчика до 4

Кроме моделирования аналоговых схем программа ISIS позволяет проверить работу и цифровых устройств. Моделирование цифровых и аналоговых устройств различно, поэтому вы можете увидеть в разделах генераторов и графиков компоненты для моделирования цифровых устройств.

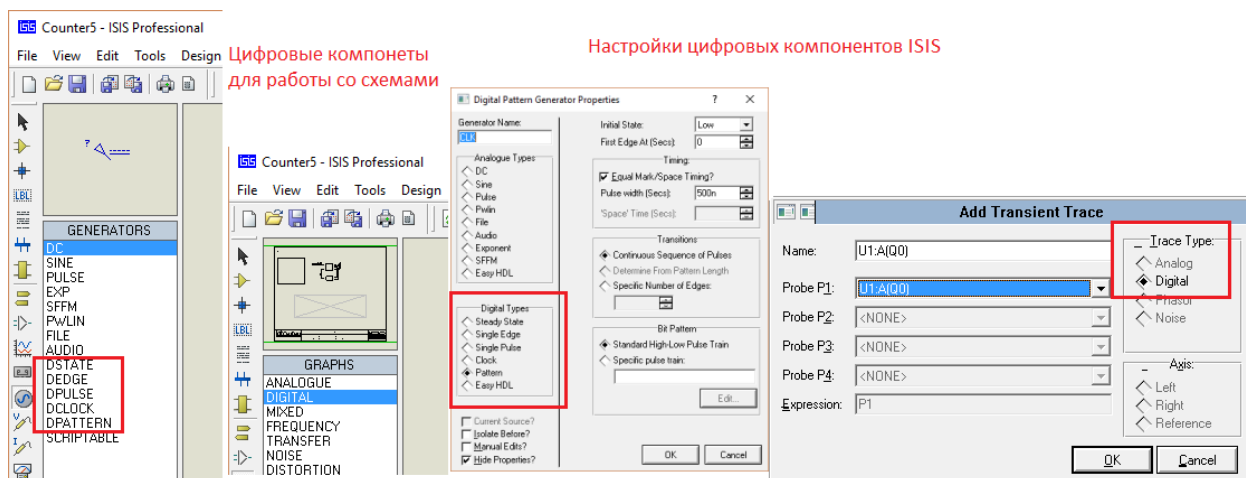


Рис. 5.2. Цифровые компоненты и настройки при работе с цифровыми устройствами

При добавлении графиков, отображающих работу цифровых компонентов, операцию добавления графика следует повторить несколько раз для нескольких выходов.

Приведённый пример показывает принцип получения счётчика для счёта до четырёх в последовательности 0, 1, 2, 3, 4.

Каждый импульс на входе увеличивает двоичное число на выходе. Когда на выходе появляется число 5, элемент «И», U2A, формирует импульс сброса счётчика.

Многие радиолюбители предпочитают строить схемы для моделирования из моделей конкретных транзисторов или микросхем. Думаю, им понравится тот факт, что программа поддерживает наиболее популярные цифровые серии микросхем.

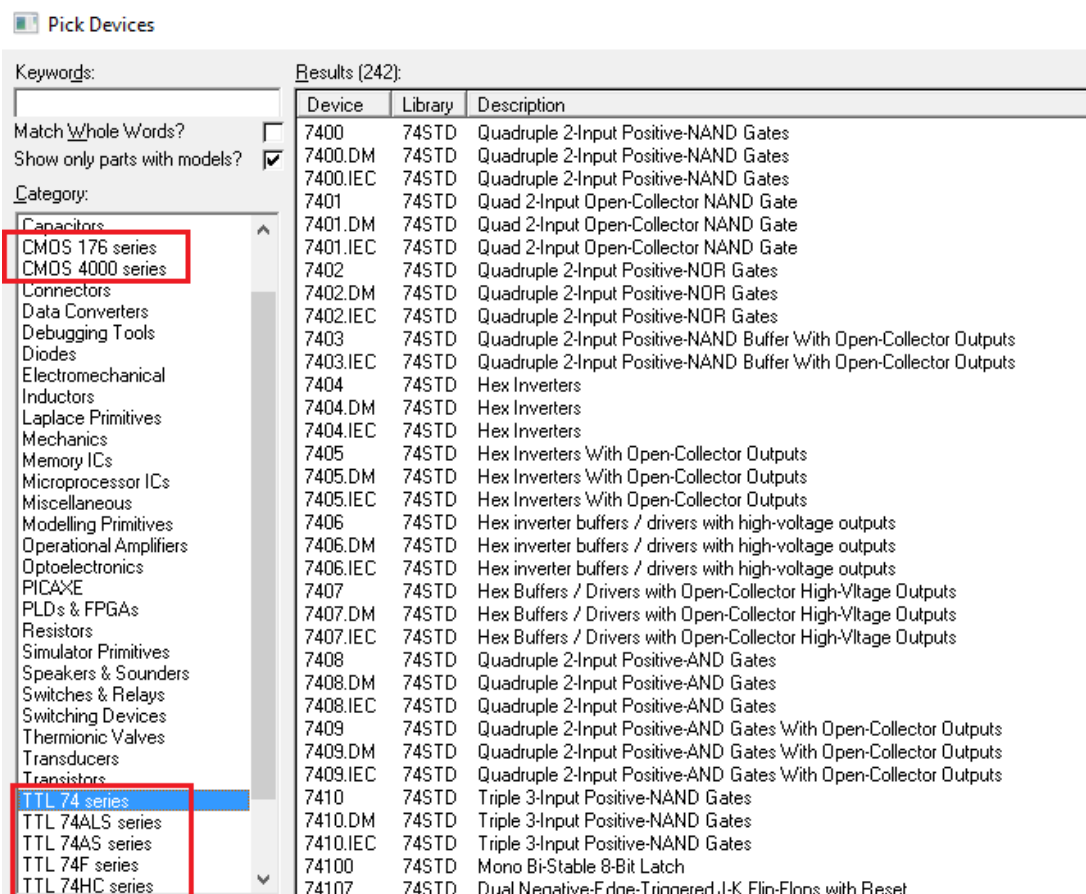


Рис. 5.3. Серии цифровых микросхем в ISIS

Хорошее соответствие отечественным сериям цифровых микросхем оставляет любителям только необходимость найти это соответствие.

Цифровые микросхемы могут использоваться самостоятельно для создания чисто цифровых устройств, но могут стать элементами «обвязки», что очень важно при работе с моделированием смешанных схем, где есть и аналоговые, и цифровые компоненты, или тогда, когда устройство базируется на микроконтроллере.

Моделирование смешанных схем (файлы ADC0808, DAC0808)

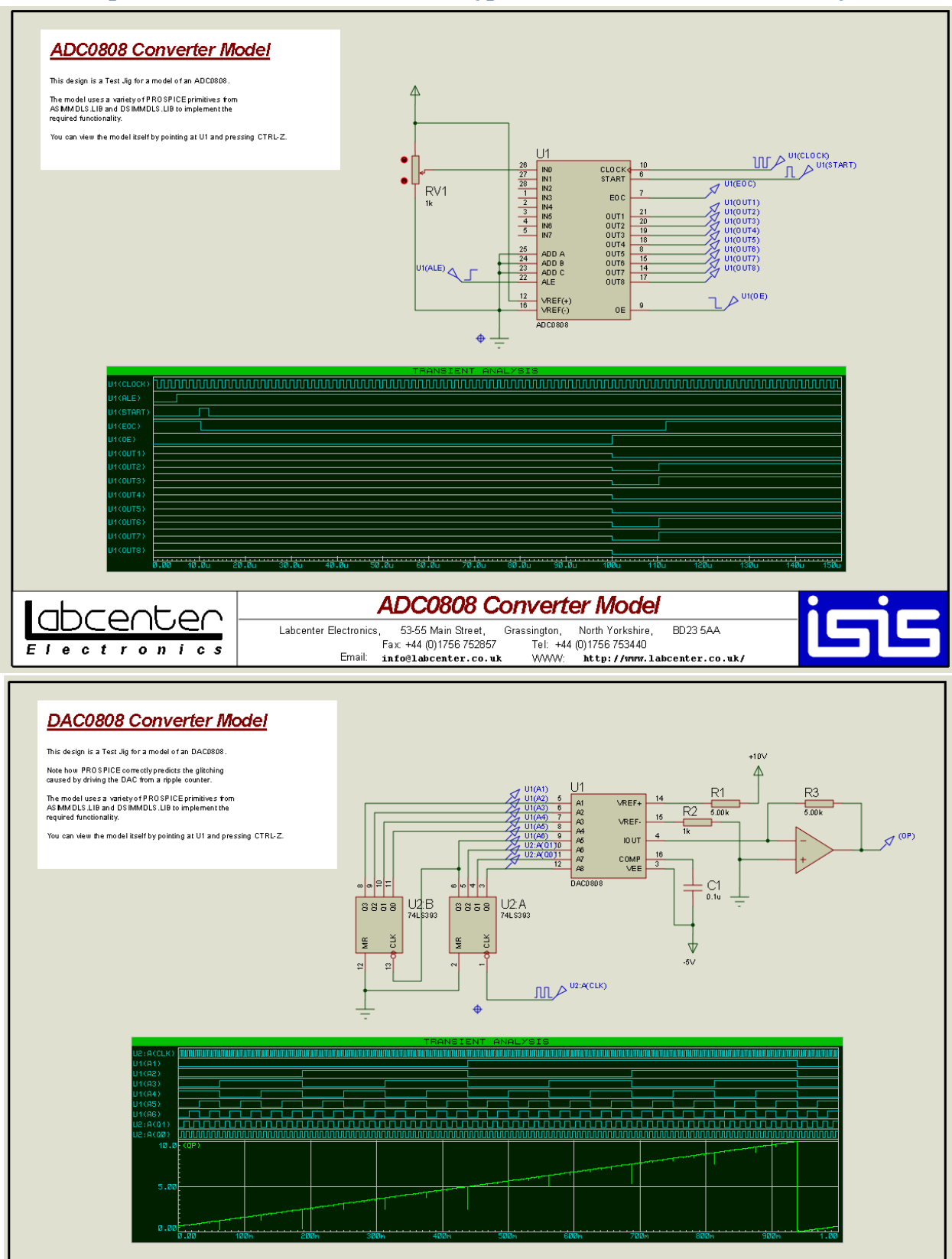


Рис. 6.1. Примеры АЦП и ЦАП

Кому-то может показаться, что приведённые примеры слишком просты, слишком примитивны показанные схемы, а результаты моделирования очевидны. Но...

Во-первых, использование программы в целях образования или самообразования – это рассмотрение принципов работы устройств, а не сложных схем.

Во-вторых, любая разработка устройств разбивается на ряд модулей, с которыми удобнее работать отдельно от всей схемы.

И, наконец, описание сложной схемы интересно только тому, кто эту схему собирается повторять. Занимая много места, это описание будет не нужно остальным. А в том, что моделировать можно и более сложные схемы вы можете убедиться на примере следующей схемы (файл Mixed.dsn):

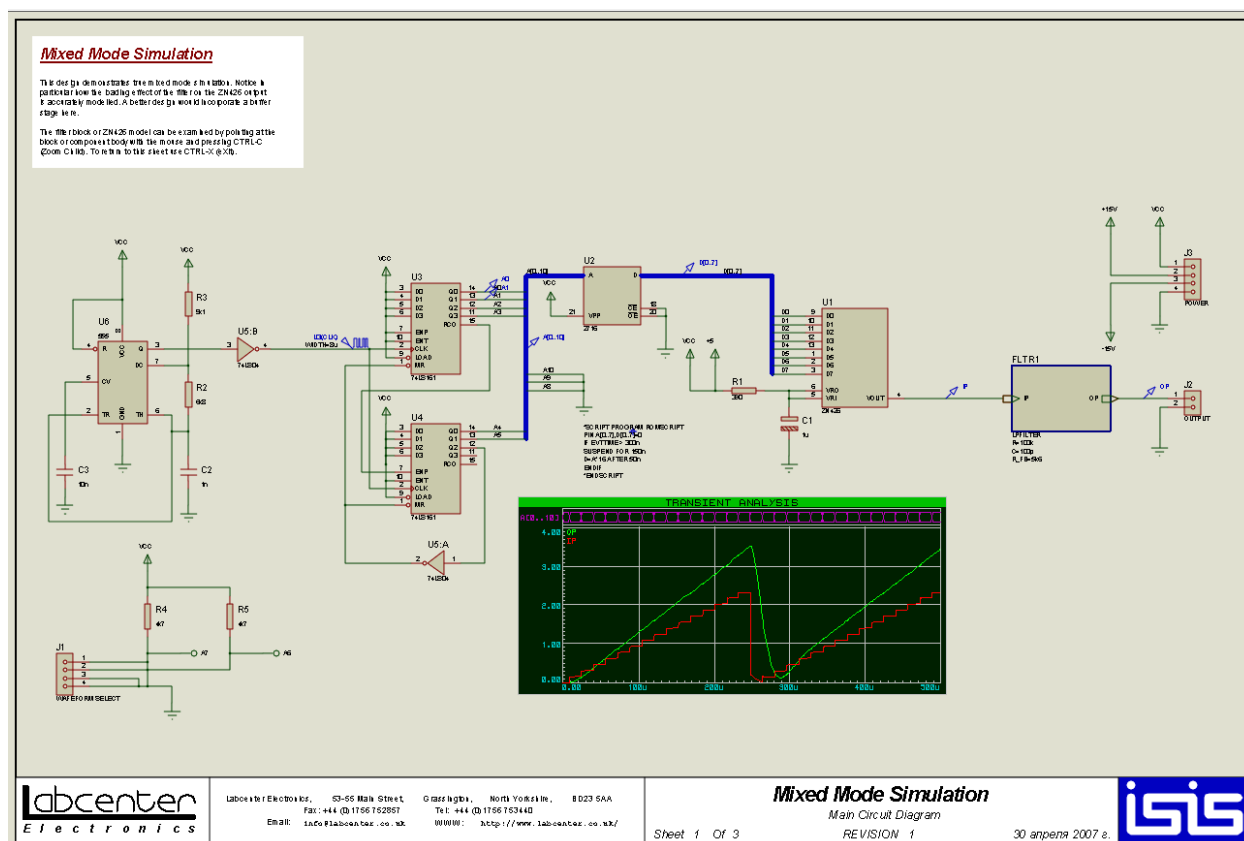


Рис. 6.2. Моделирование схемы, содержащей подсхемы

Наличие конкретных моделей микросхем, как мне кажется, в первую очередь связано с тем, что программа Proteus предназначена для сквозного проектирования. То есть, в ISIS можно собрать и отладить схему устройства, а в ARES, вторая часть программы, осуществить разводку печатной платы, получив всё необходимое для производства готовой платы.

Возможно, я этого не знаю, на крупных производствах применяют более мощные программы для сквозного проектирования, но и для любителей, и для студентов, и для небольших фирм использование программы Proteus может оказаться более удобным, а, главное, более дешёвым вариантом разработки.

Пример готового устройства с разводкой платы (файл Dbell.dsn)

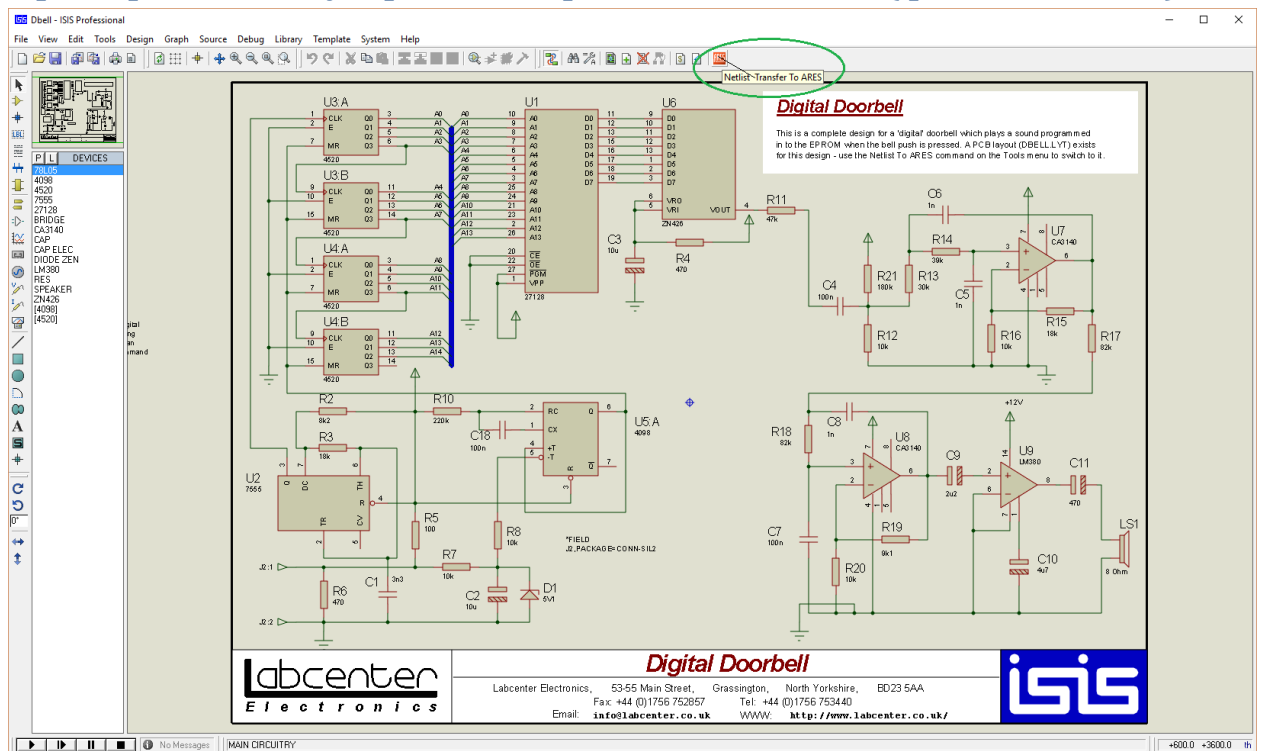


Рис. 7.1. Схема готового устройства

Создав и проверив схему устройства, можно использовать иконку на основной инструментальной панели (выделено на рисунке), чтобы перейти к разводке печатной платы. Сгенерировав netlist, программа открывает рабочее поле редактора ARES.

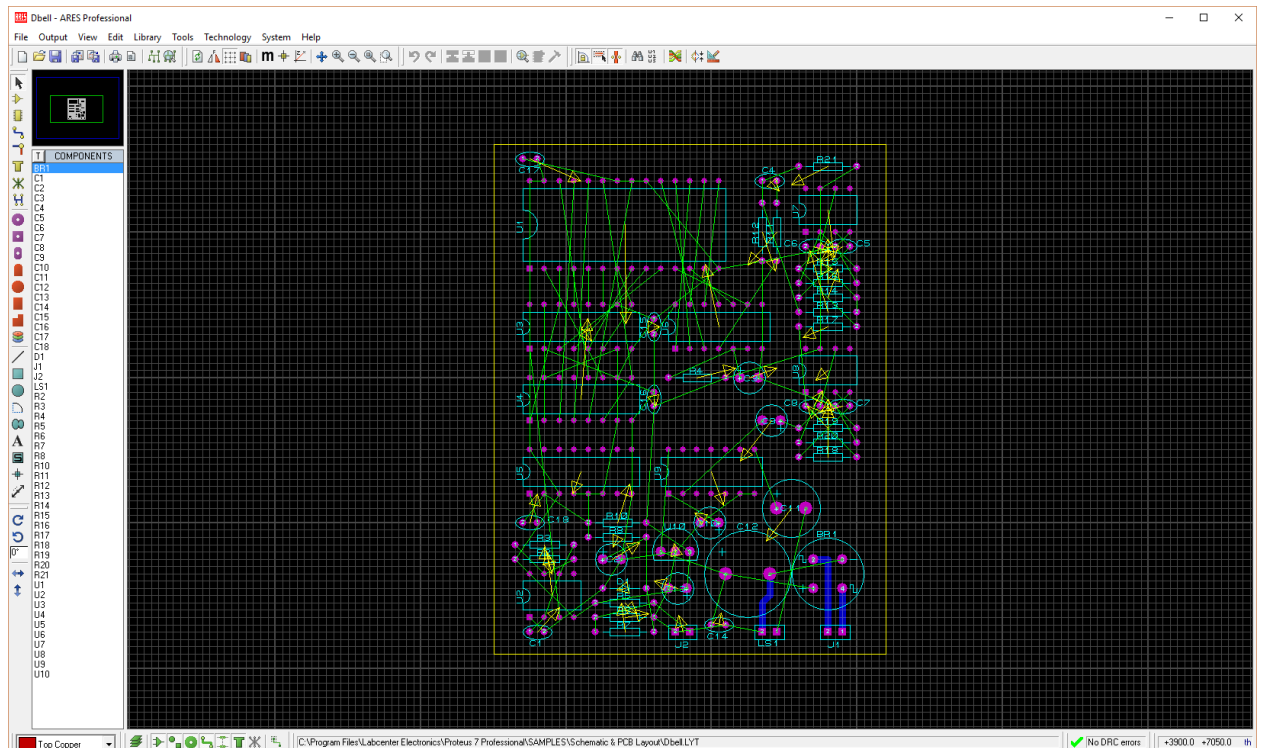


Рис. 7.2. Печатная плата в программе ARES

О том, как работать с программой ARES, следует почитать руководство к программе. На форуме kazus.ru можно поискать перевод руководства. Но следует помнить, что создание печатной платы – это работа не менее сложная, чем создание схемы. Как и другие аналогичные программы, ARES облегчает эту работу:

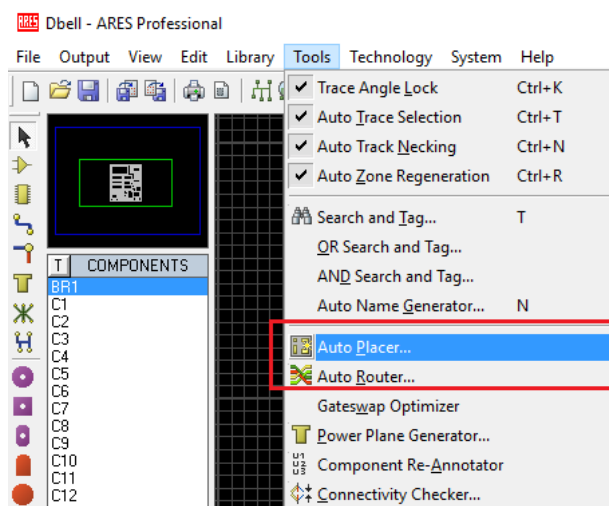


Рис. 7.3. Автоматическое размещение компонентов и автоматическая разводка платы

Более того, при разводке платы важно увидеть конечный результат – плату с установленными компонентами. В последнее время программы разводки снабжают подпрограммой, которая позволяет это сделать:

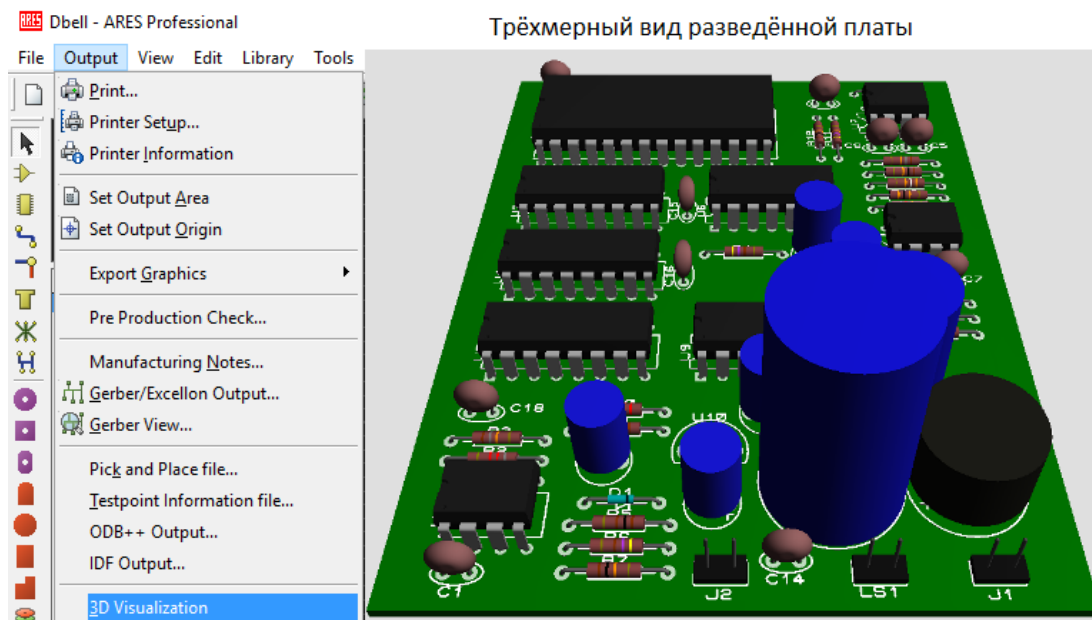


Рис. 7.4. Переход к трёхмерному виду платы

В разделе SAMPLES->Schematics & PCB Layout вы найдёте много примеров работы по разводке печатных плат. Но обычно я не советую для простых единичных любительских устройств использовать печатные платы. Достаточно смонтировать устройство на макетной плате. Вместе с тем, очень полезно, используя программу разводки, получить монтажную схему. Это уменьшит

вероятность ошибок при монтаже и позволит сделать монтажную схему удобную для понимания и отладки устройства.

Однако, если вы решили создать более или менее сложное устройство с микроконтроллером, если планируете использовать поверхностный монтаж, то, скорее всего, печатная плата лучшее решение.

Одной из самых привлекательных сторон программы Proteus я считаю возможность моделирования устройств на базе микроконтроллеров.

Работа с микроконтроллерами в ISIS (файл extram.dsn)

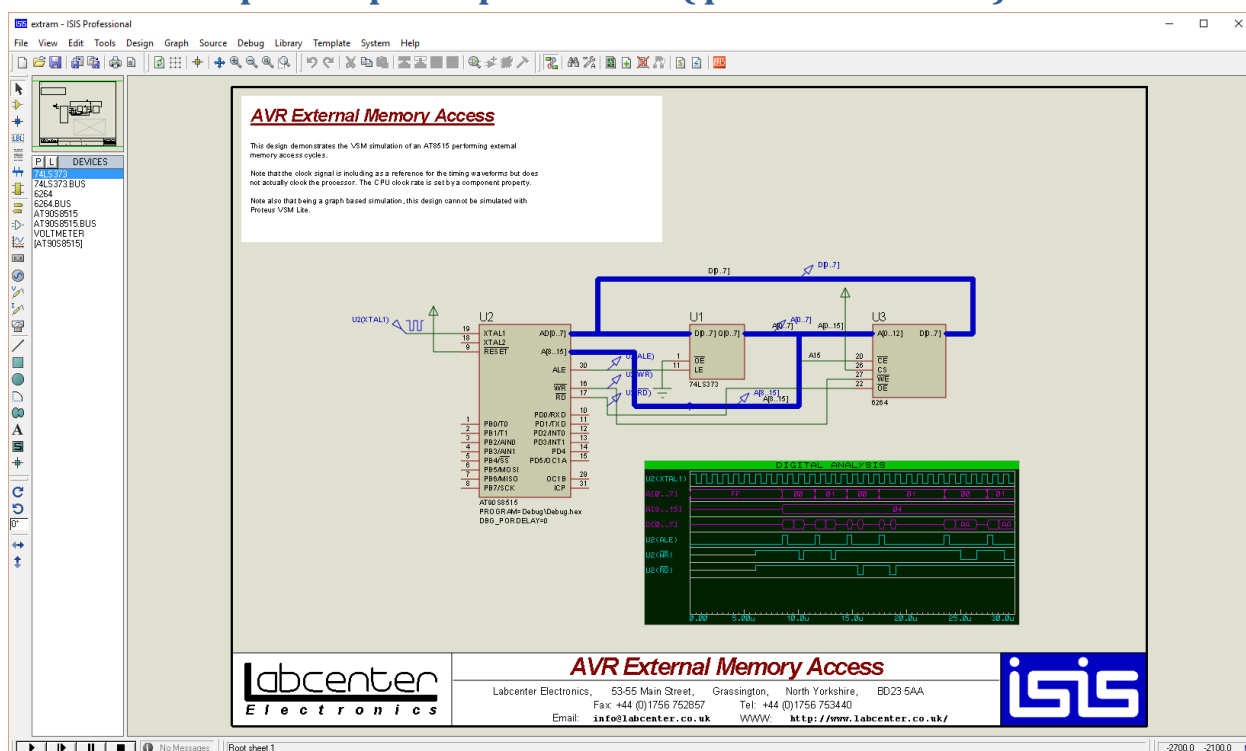


Рис. 8.1. Микроконтроллер AVR и внешняя память

Если отладка схем, как аналоговых, так и цифровых, может осуществляться на макетной плате, то отладка микроконтроллерных устройств предпочтительнее в специальных программах. Проблема в том, что на макетной плате можно отслеживать только выходы портов, но не внутреннюю работу контроллера.

Многие среды разработки программ предлагают отладку с использованием внешних компонентов. Но Proteus предлагает почти любые внешние компоненты, чем не может похвастаться модуль отладки других сред разработки.

Написать код программы, с моей точки зрения, удобнее в среде разработки от производителя. Так для микроконтроллеров AVR удобно использовать бесплатную программу Atmel Studio. Можно писать код на ассемблере или, например, на Си.

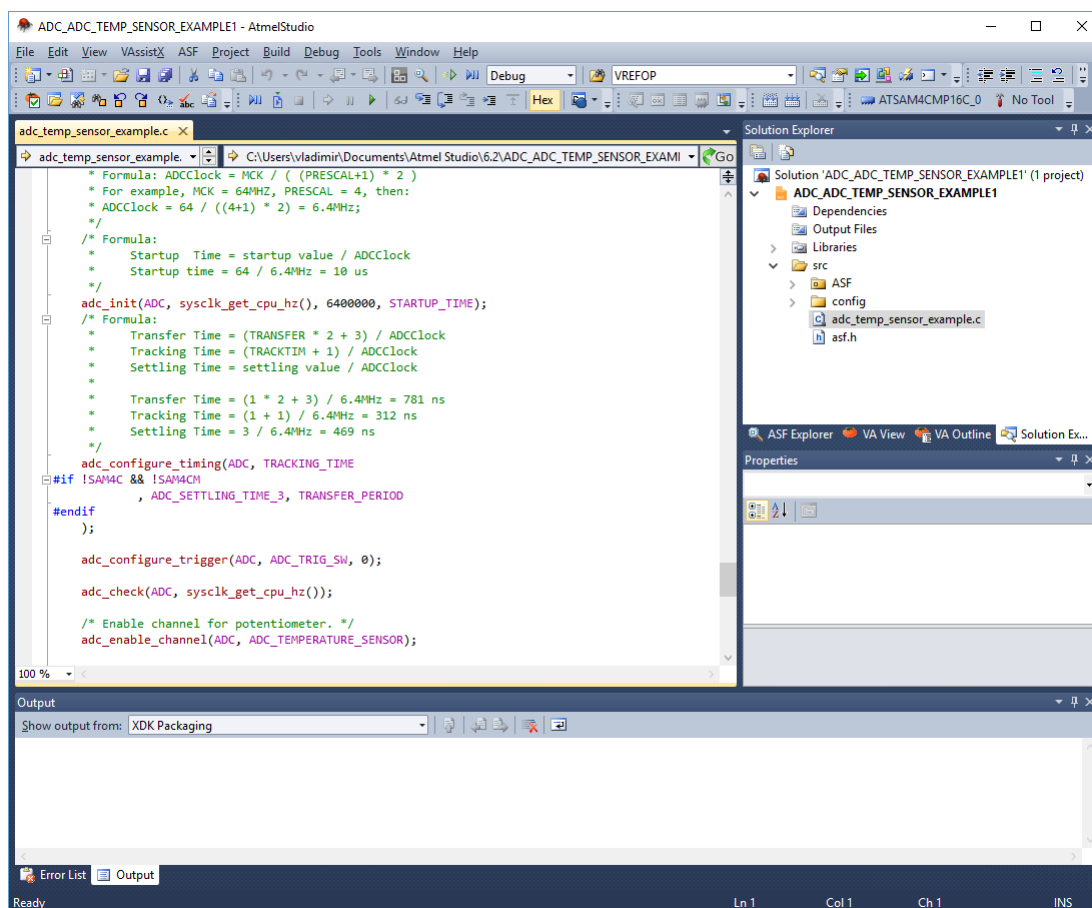


Рис. 8.2. Среда разработки программ для AVR контроллеров

Среда разработки имеет собственный отладчик, позволяющий проверить все основные моменты в работе микроконтроллера. Но, если, а это бывает часто, вы используете многочисленные внешние элементы, то после компиляции программы вы можете вставить исполняемый файл в схему с внешними компонентами, как это показано на рисунке 8.1. Для этого откройте свойства микроконтроллера:

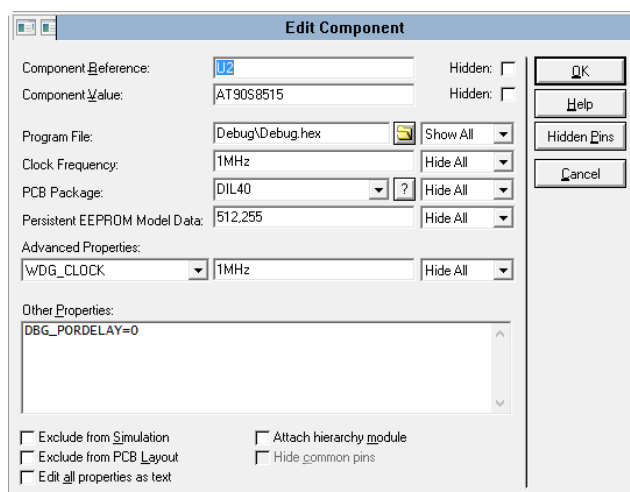


Рис. 8.3. Диалоговое окно свойств AT90S8515

В диалоге вы указываете расположение исполняемого файла и частоту тактового генератора. При отладке вы можете использовать пошаговый режим.

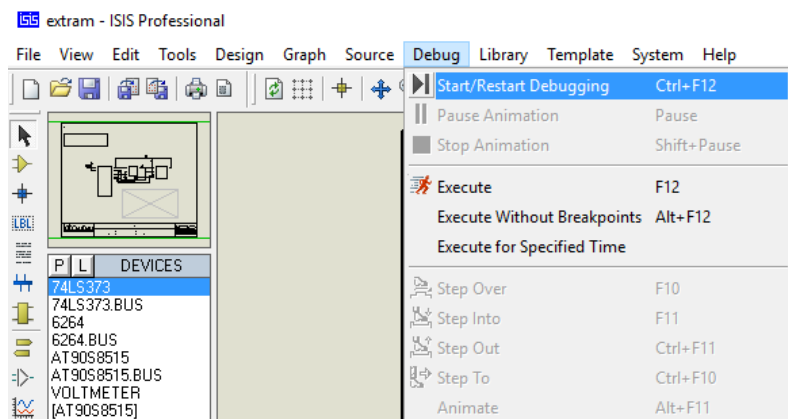


Рис. 8.4. Начало отладки программы

В дополнительном окне при пошаговой отладке можно видеть команды, которые выполняет процессор микроконтроллера.

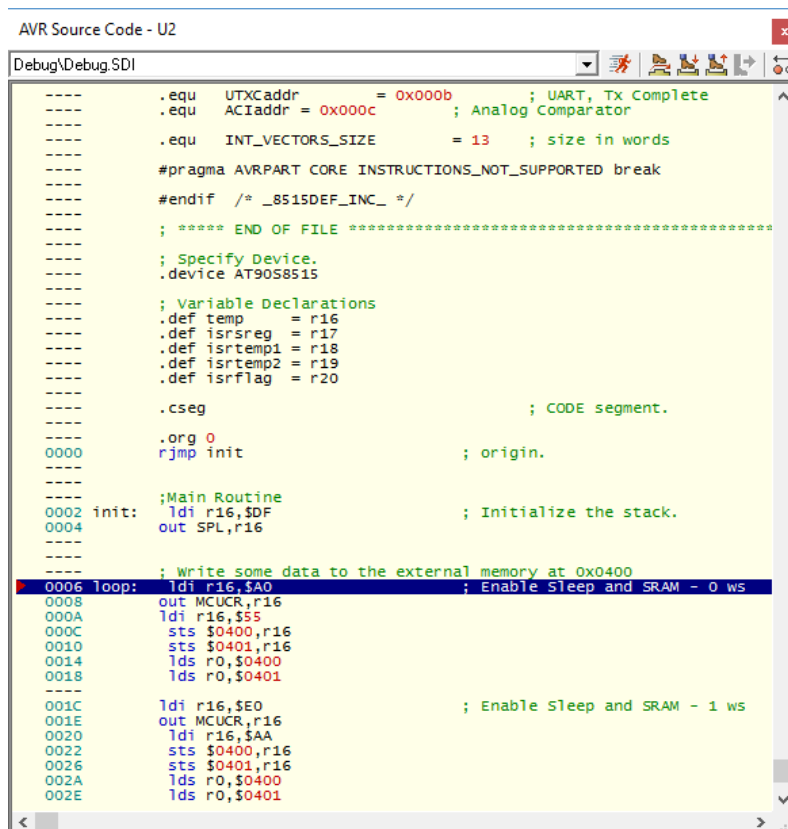


Рис. 8.5. Пошаговая отладка программы

Ещё одним достоинством программы Proteus можно считать то, что она работает и с микроконтроллерами других производителей. Например, можно написать код программы в свободной программе от производителя Microchip, которая называется MPLAB X.

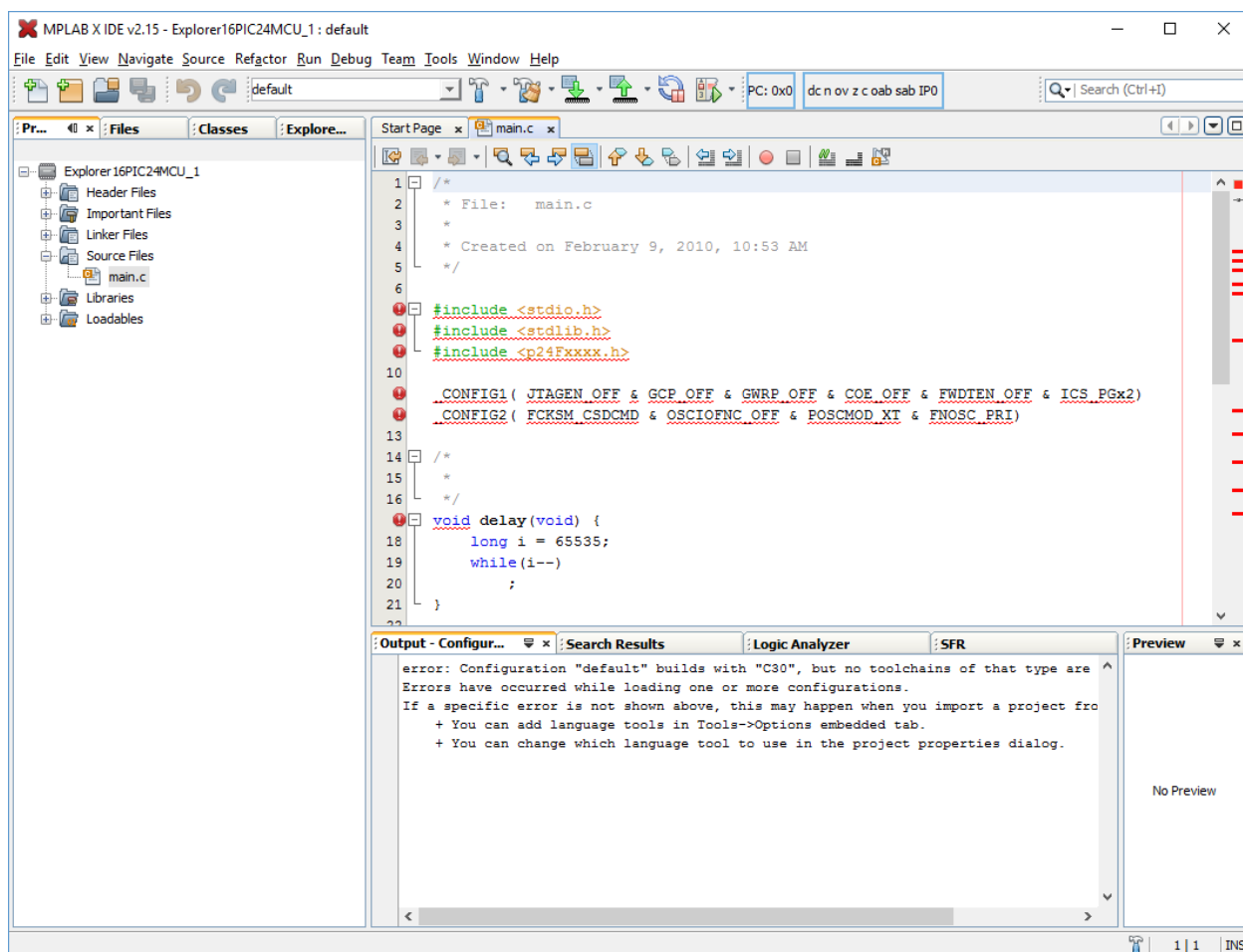


Рис. 8.6. Среда разработки PIC-контроллеров MPLABX

Как и с AVR микроконтроллером, вы можете написать код программы, отладить её, оттранслировать и провести окончательную проверку работы устройства в программе ISIS.

На рисунке 8.1. показана графическая проверка работы контроллера с внешней памятью. Это удобно, поскольку позволяет тщательно исследовать все аспекты работы, но иногда удобно использовать виртуальные приборы ISIS.

Виртуальные приборы (файлы DemoBoard.dsn, Poweramp.dsn)

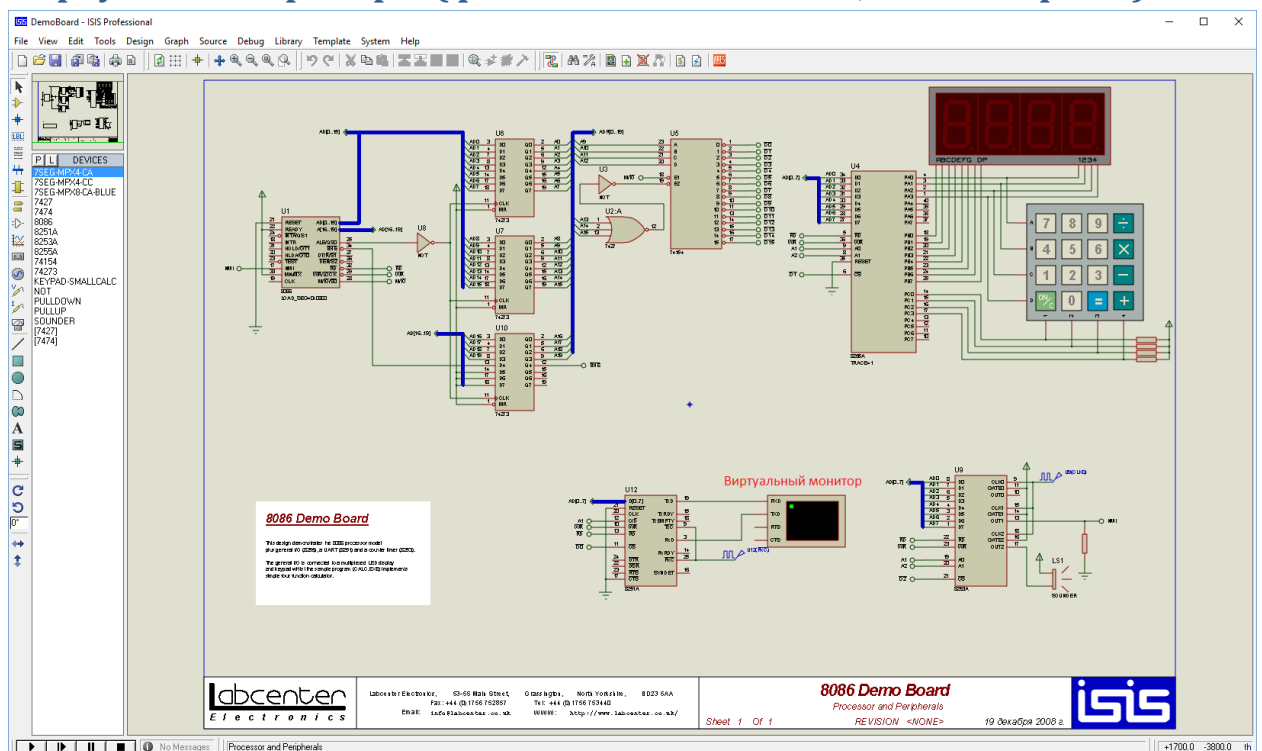


Рис. 9.1. Виртуальный монитор в ISIS

Для работы с модулем USART, встроенным в микроконтроллер, очень важно правильно настроить его, чтобы получить правильную скорость последовательного асинхронного обмена. Графическая проверка в этом случае позволяет увидеть наличие импульсов, но затрудняет проверку скорости.

Открыв диалоговое окно виртуального терминала (свойства терминала), вы можете настроить его работу:

Рис. 9.2. Диалоговое окно свойств виртуального терминала

А после запуска моделирования, когда появляется окно терминала, вы можете правым щелчком мышки вызывать выпадающее меню, где можно сделать дополнительный выбор свойств терминала.

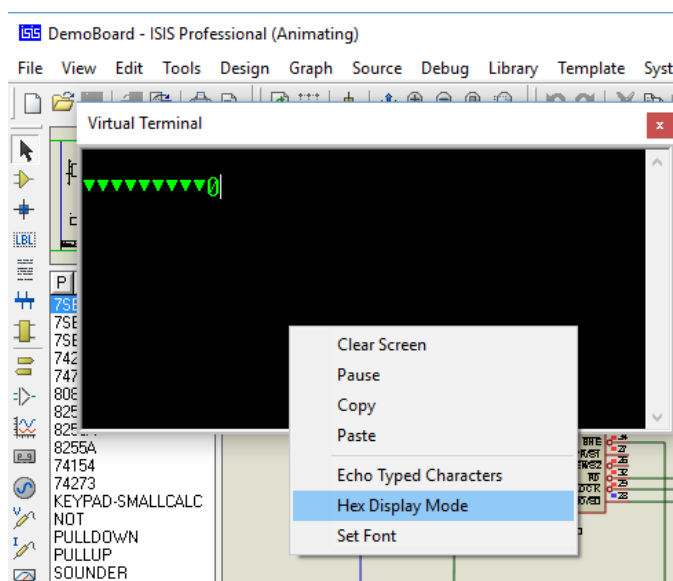


Рис. 9.3. Выпадающее меню свойств терминала

Посмотреть все доступные виртуальные приборы можно с помощью иконки на левой инструментальной панели.

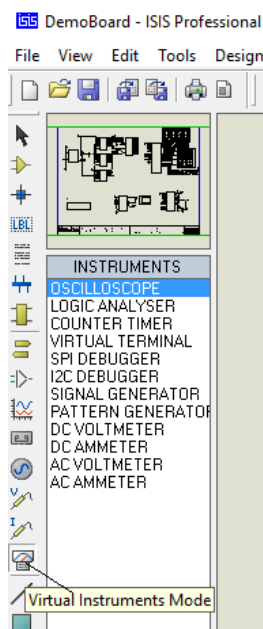


Рис. 9.4. Перечень виртуальных приборов ISIS

Работа с виртуальными приборами очень похожа на обычную отладку схемы на макетной плате. Самым востребованным прибором в этом случае служит осциллограф.

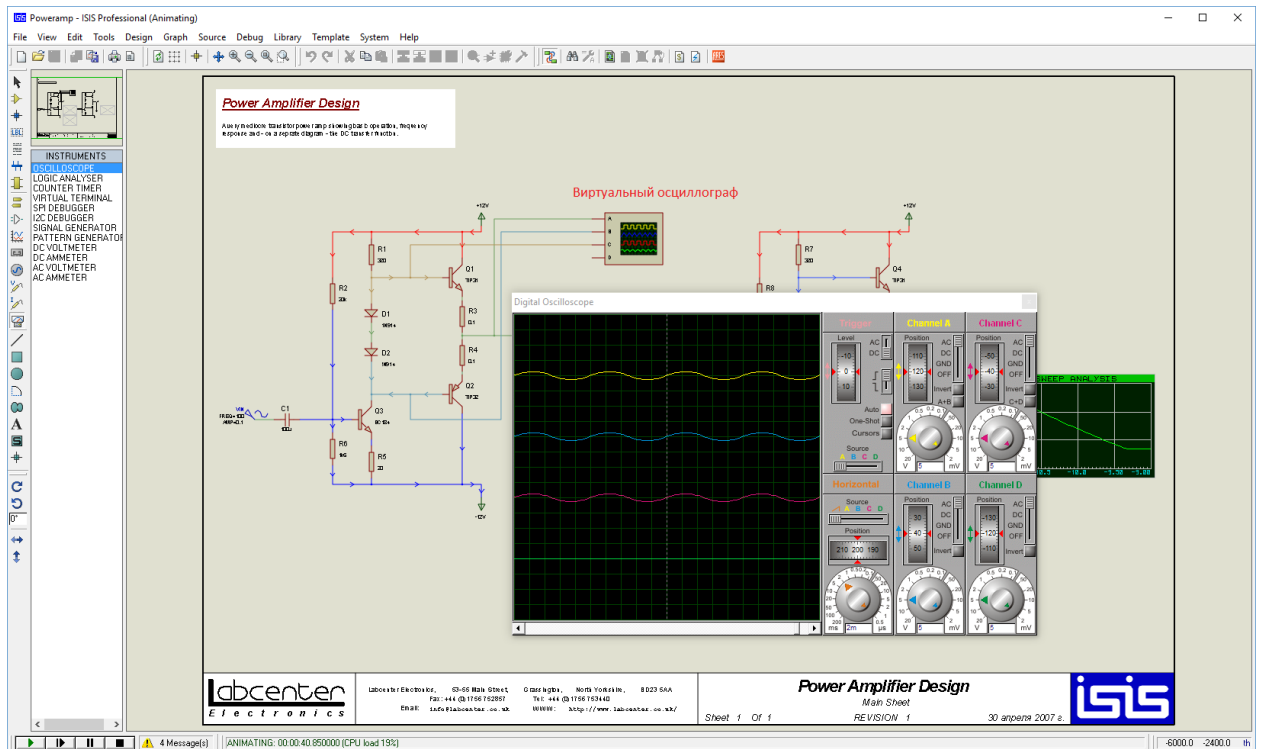


Рис. 9.5. Использование виртуального осциллографа для наладки устройства

Как у реального осциллографа, у виртуального есть ручки, управлять которыми можно с помощью курсора и левой клавиши мышки.

Наглядные пособия (файлы AC03.dsn, Mvcr09.dsn)

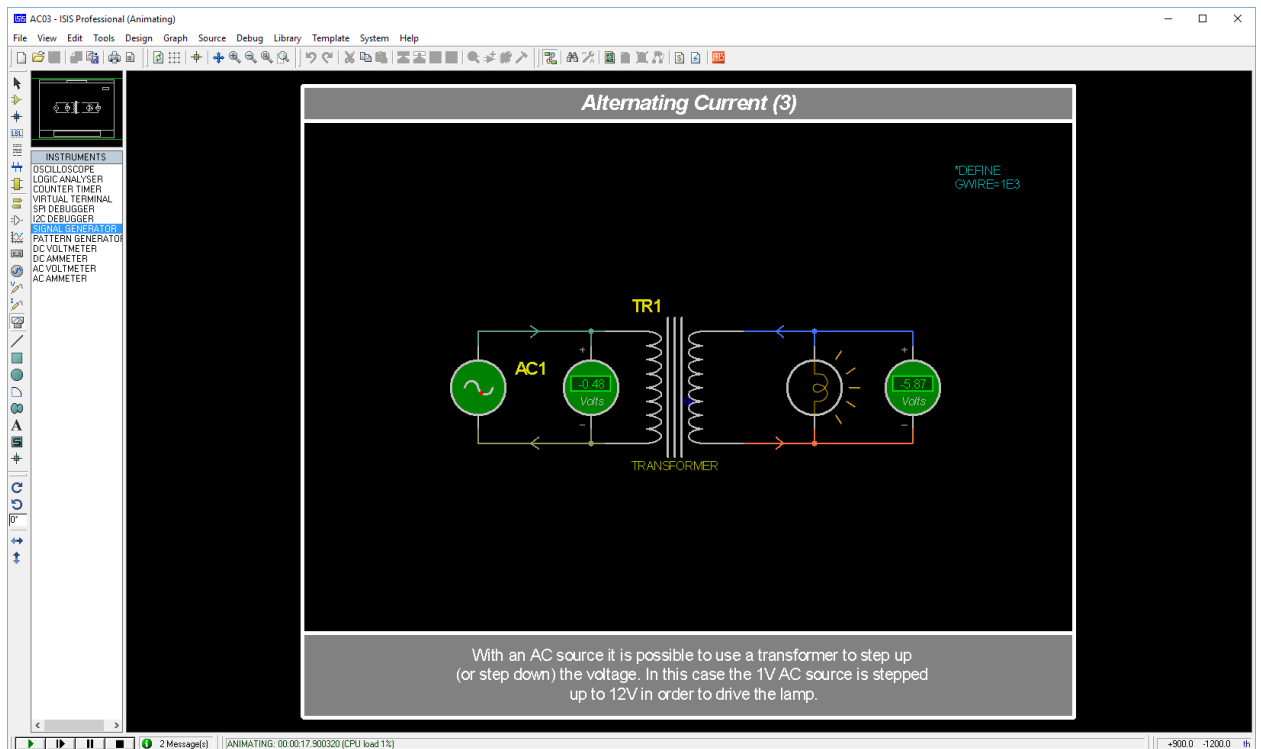


Рис. 10.1. Анимированная схема

Если стоимость программы окажется «подъёмной», например, для школы, то программу можно использовать не только в школьном радиокружке, но и на уроках физики при изучении раздела электричества. Очень наглядно выглядят анимированные схемы, которые можно найти в SAMPLES->Interactive Simulation->Animated Circuits.

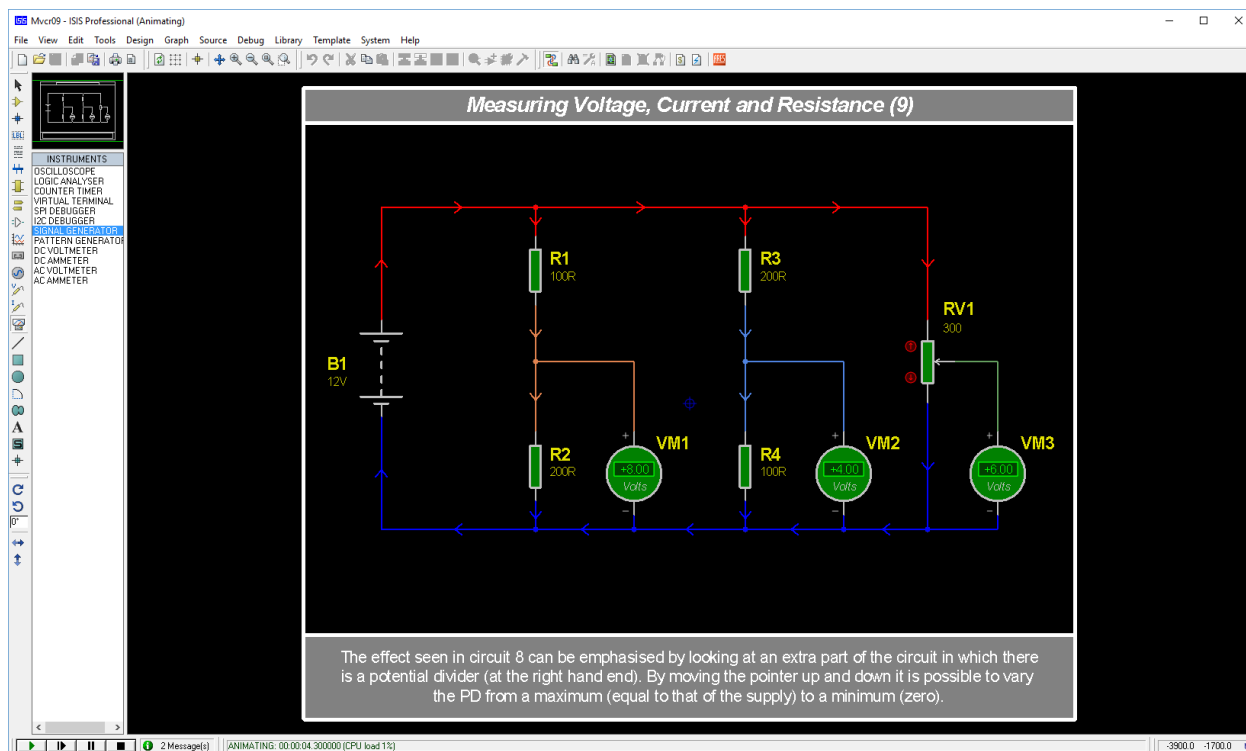


Рис. 10.2. Вольтметры и делители напряжения

Выводы

В этом кратком обзоре рассмотрены далеко не все примеры, их много. Если вас заинтересовала программа, то я советую начать ознакомление с ней с чтения руководств к программе (слегка устаревший перевод к ISIS можно найти на моём сайте <http://vgololobov.narod.ru>), а перевод к ARES поищите на форуме kazu.ru.

И я ещё раз очень советую ознакомиться с FAQ на форуме kazu.ru, созданный автором под ником Halex07. Там на форуме вы найдёте и обсуждение проблем, возникающих при работе с программой, и примеры устройств, которые созданы пользователями.

P.S. Так получилось...

...что я опростоволосился, когда рассказывал о пользе применения подсхем. В этой главе я упомянул, что в качестве задающего генератора можно использовать таймер 555. В этом месте я планировал рассказать об одном полезном приёме – копировании готовой схемы для своих нужд.

Среди примеров в разделе Graph Based Simulation есть файл Mixed.dns. Вот эта схема:

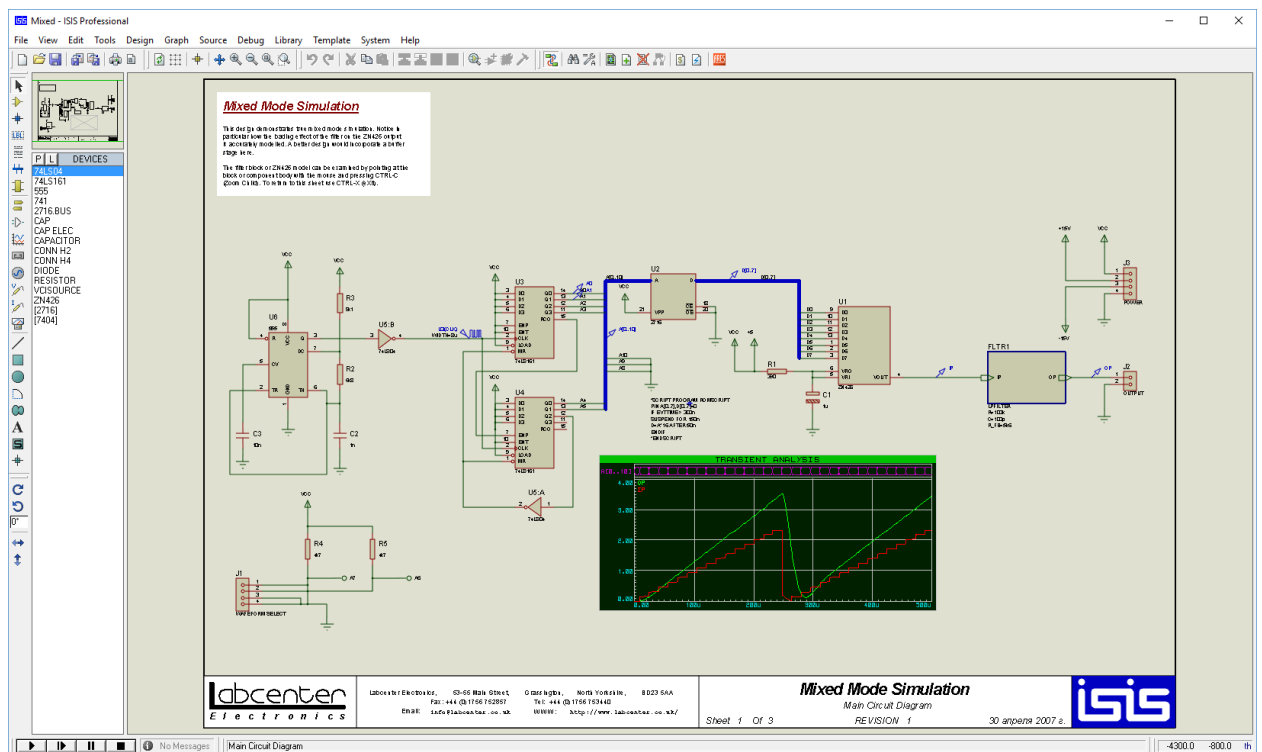


Рис. 11.1. Пример смешанной схемы

На схеме есть интересный нас генератор на таймере 555. Достаточно выделить его, щёлкнуть правой клавишей мыши по выделению и выбрать команду копирования в буфер обмена.

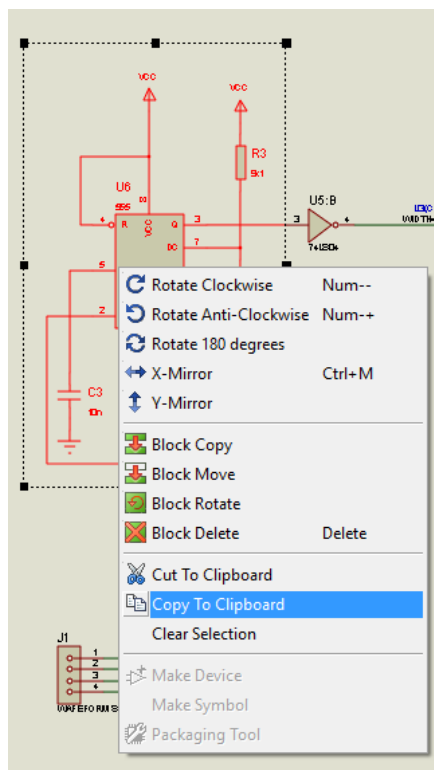


Рис. 11.2. Копирование нужной схемы

Создав новый файл (иконкой на инструментальной панели или через основное меню), вы получаете возможность вставить скопированную схему в свой файл, используя выпадающее меню, которое появляется при щелчке правой клавиши мышки по рабочему полю. В нём есть команда вставки из буфера обмена.

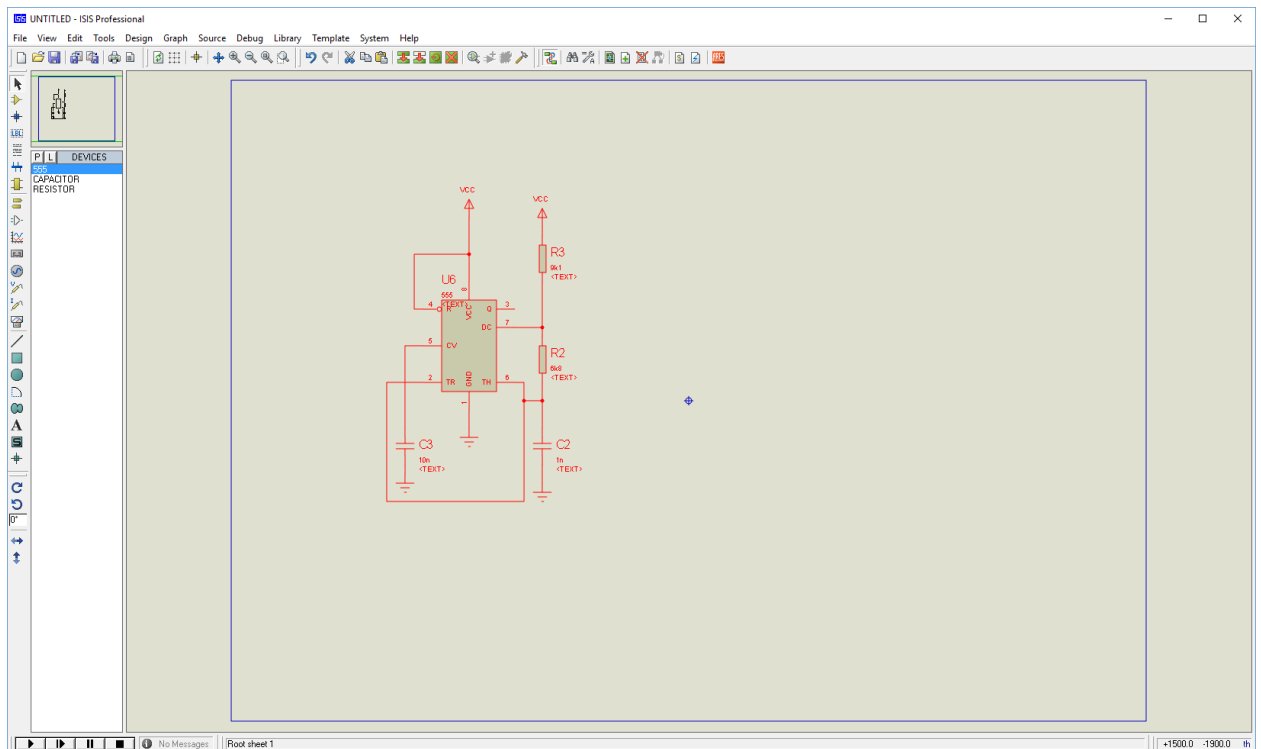


Рис. 11.3. Вставка скопированной схемы

Чтобы избежать сомнений, добавим батарейку для питания схемы. Для построения графика выхода добавим пробник напряжения (с левой инструментальной панели). Из раздела графиков (на той же панели) выберем построение аналогового графика. Добавим, как и положено, пробник $U_6(Q)$, таким он появился на схеме, в диалоговом окне выбора трассы. Оставим настройки графика (пока) без изменения и запустим симуляцию графика.

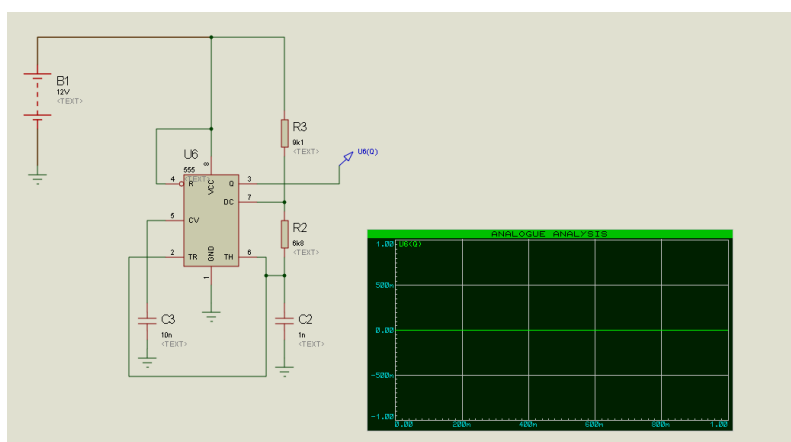


Рис. 11.4. Результат симуляции

Вот такой получился конфуз!

Я знаю, что схема генератора правильная. Я знаю, что почти все программы симуляции поддерживают работу таймера 555. Поэтому меня очень заинтересовало всё произошедшее. В чём же причина неудачи?

Давайте посмотрим свойства всех элементов схемы:

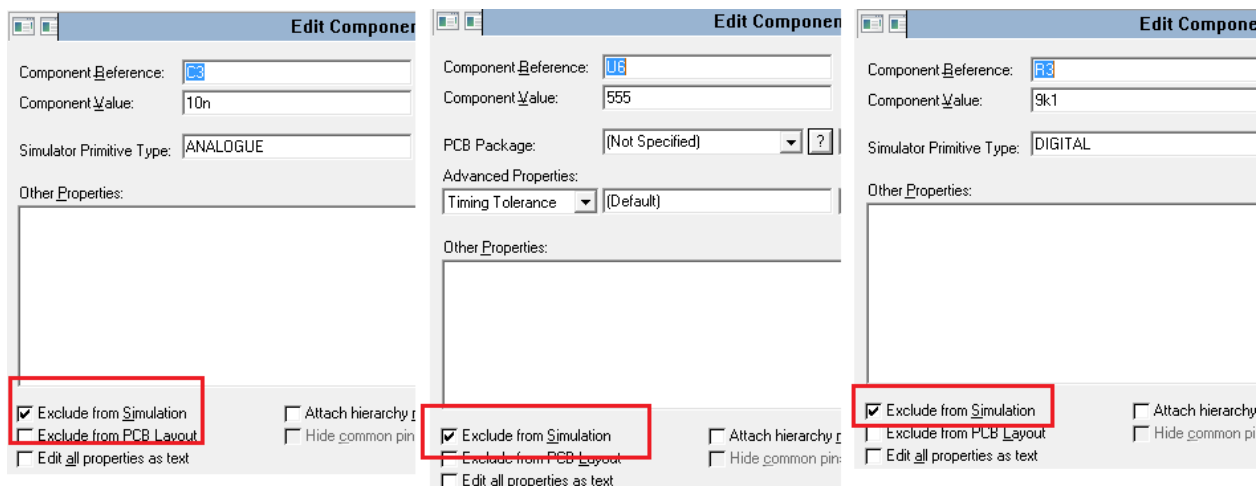


Рис. 11.5. Свойства элементов схемы

Одна из причин в том, что эти элементы исключены из симуляции. Исправим это, сняв галочки в окошках Exclude from Simulation. Но и это ещё не всё. Вот, что следует ещё сделать:

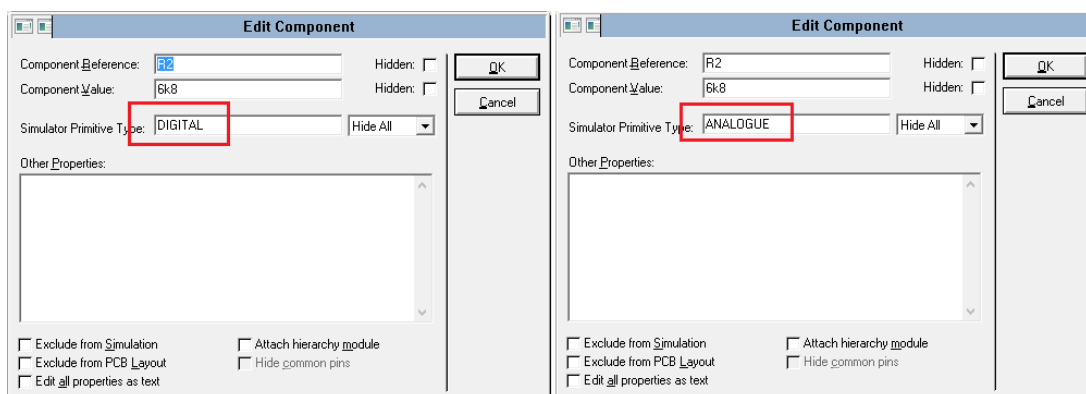


Рис. 11.6. Изменение типа элементов

Последнее, что следует исправить – это изменить время окончания симуляции:

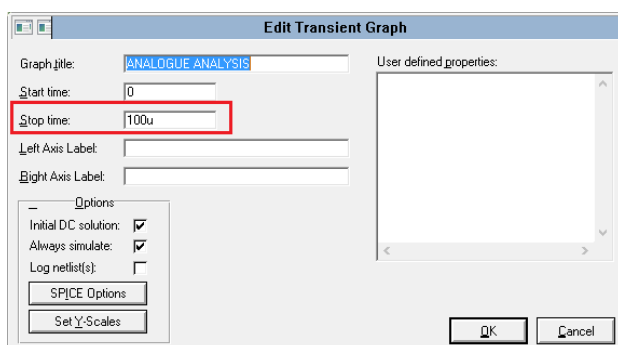


Рис. 11.7. Окончательная настройка графика

Теперь симуляция выполняется.

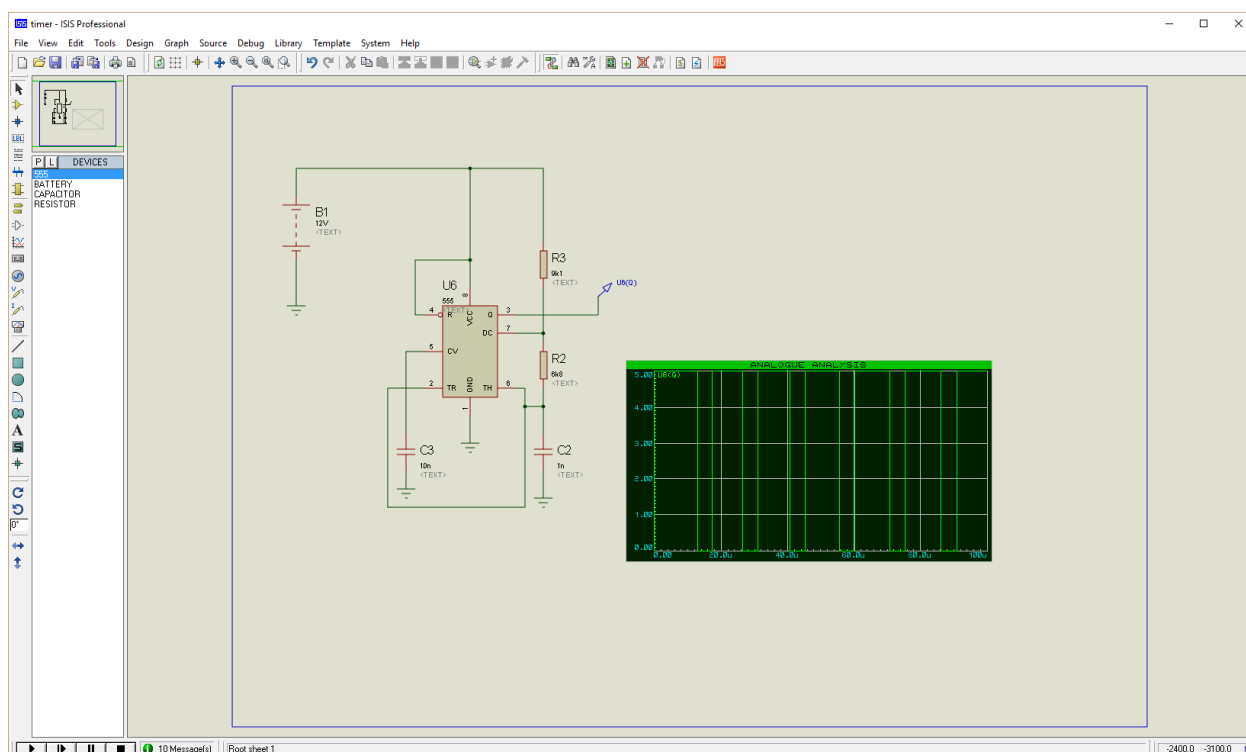


Рис. 11.8. Окончательная симуляция генератора с таймером 555

То, что амплитуда импульсов 5 В, можно отнести к свойствам модели. Можно попробовать другую модель. Но из этого случая можно сделать ряд выводов.

Во-первых, удобно использовать копирование готовых схем.

Во-вторых, копируя готовые схемы, следует тщательно разбираться со свойствами каждого элемента.

Затем, если схема большая, её моделирование может оказаться достаточно долгим и сложным. В этом случае можно исключать часть схемы из симуляции, что и было сделано в примере со смешанными элементами. Генератор на таймере 555 заменили штатным генератором импульсов, оставив его в схеме для полноты картины.

И последнее, похоже, что и резисторы с конденсаторами могут иметь разные типы для аналогового и для цифрового моделирования. Но это нужно уточнять, проверять, читая руководство и производя опыты. Для этого, отчасти, и служат примеры, созданные разработчиками программы Proteus.