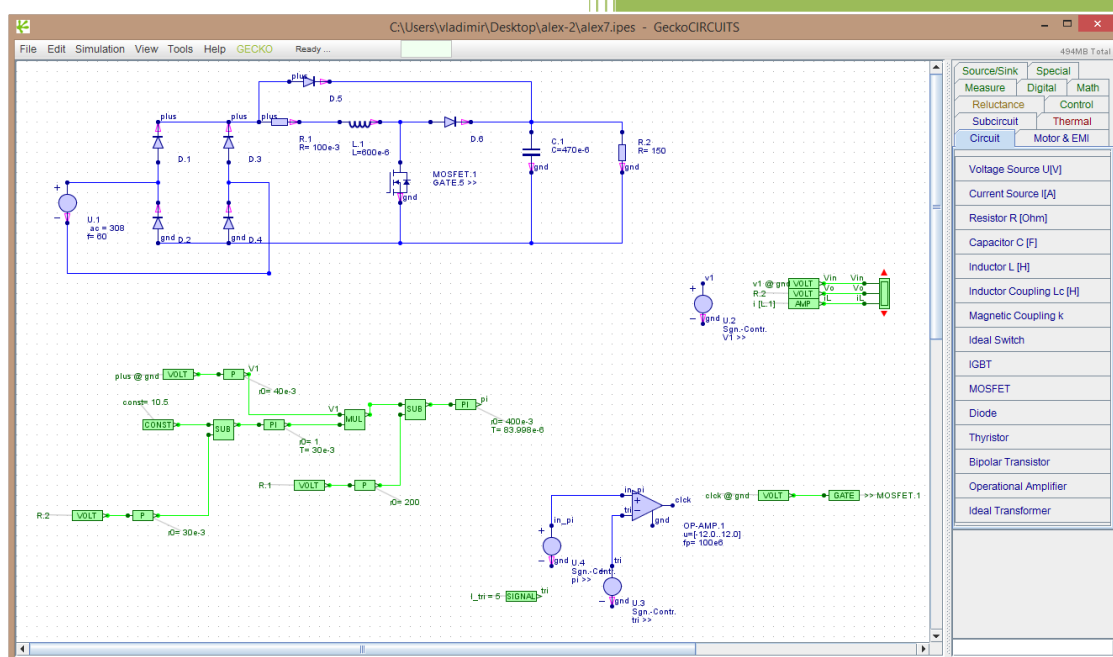




В.Н. Гололобов

Конвертеры в программе GecoCircuits



Vladimir Gololobov

Москва - 2014

Конвертеры в программе GeckoCIRCUITS

Оглавление

Предыстория.....	2
Ввод схемы.....	5
Моделирование работы схемы.....	14
Как работает повышающий конвертер	17
Ещё немного о программе GeckoCIRCUITS.....	25
Заключение	35

Предыстория

Мой давний знакомый часто работает с конвертерами, рассчитывая и моделируя работу схемы, а время от времени привлекает моё внимание к разным схемам конвертеров. Он же даёт ссылки на бесплатные программы, которые (хотя бы в принципе) могут моделировать конвертеры.

В этот раз он предложил мне рассмотреть такую схему конвертера:

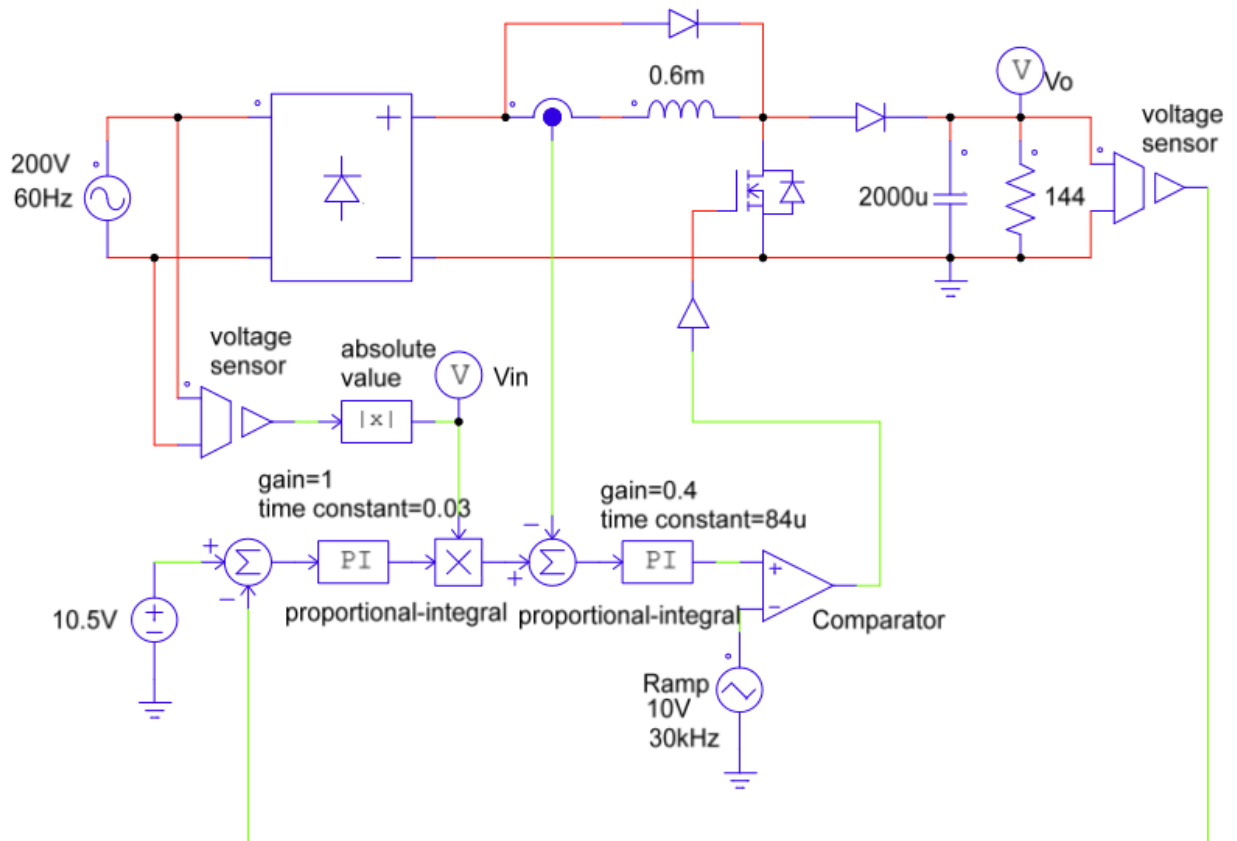


Рис. 1. Схема конвертера

Мне не довелось разрабатывать схемы конвертеров, поэтому особого интереса они у меня, признаюсь, не вызывают, исключая готовые изделия, которыми мы все пользуемся.

Он же дал ссылку на бесплатную программу, предложив смоделировать работу конвертера в этой программе симуляции:

<http://www.gecko-simulations.com/geckocircuits.html>

Программа, насколько я понял, создана в Швейцарском университете, не требует установки, но наличия Java версии не ниже 1.6. После распаковки в Windows 8 достаточно обычного двойного щелчка по исполняемому файлу на языке Ява:

Конвертеры в программе GeckoCIRCUITS

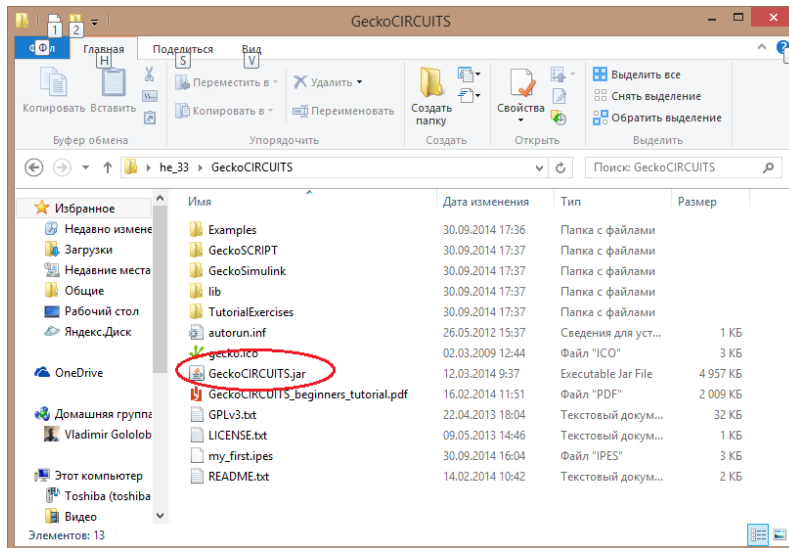


Рис. 2. Папка с программой GeckoCIRCUITS

По опыту предыдущих наших бесед я знаю, что не всегда удаётся запустить моделирование в программе, хотя обусловлено это может быть отсутствием особого моего интереса к этому процессу. Но меня всегда интересует, а не будет ли работать программа в Linux. Этот раз не стал исключением. В Fedora 18 программа запускается, хотя немного странно. Требуется запустить её из терминала (в свойствах я отметил, что это исполняемая программа, но не помогло). Не исключаю, что и здесь требуется некоторое «шаманство», но... Не слишком обременительно запустить программу из папки, помещённой в твою домашнюю директорию:

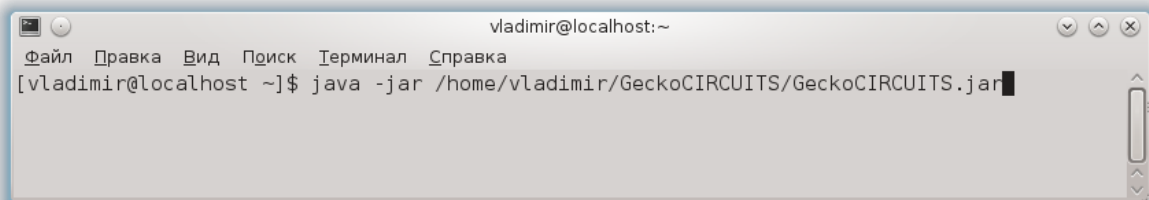


Рис. 3. Запуск программы из терминала в Linux

Терминал в Linux имеет свойство сохранять последние команды при повторном запуске даже после выключения компьютера, достаточно запустить терминальную программу и нажать на курсорной клавиатуре стрелку вверх.

Программа запускается, равно как и в Windows, давая основное рабочее окно:

Конвертеры в программе GeckoCIRCUITS

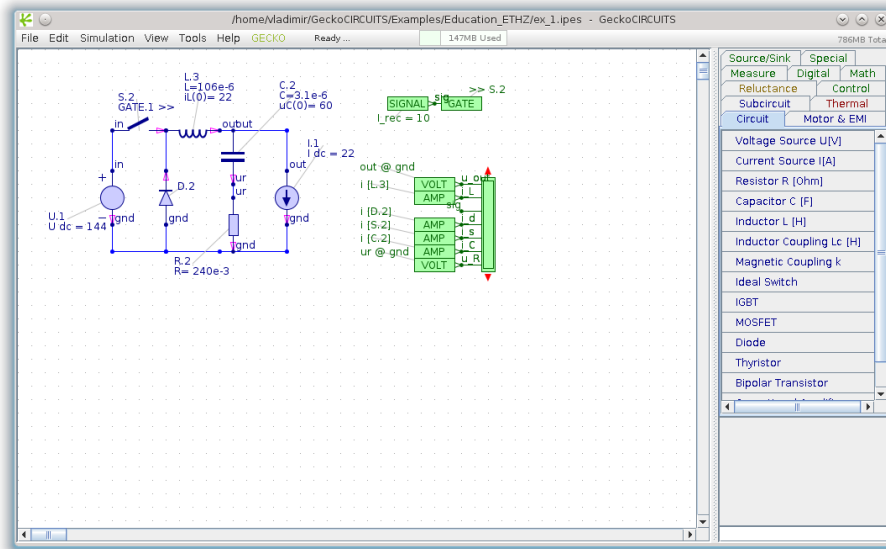


Рис. 4. Программа GeckoCIRCUIT в Fedora 18

Программа приходит с примерами и обучающими документами, которые можно найти в папке с программой. И при запуске программы она открывает ранее открытый файл. На рисунке выше открылся файл из набора примеров, который я открывал ранее. И можно убедиться, что моделирование работает.

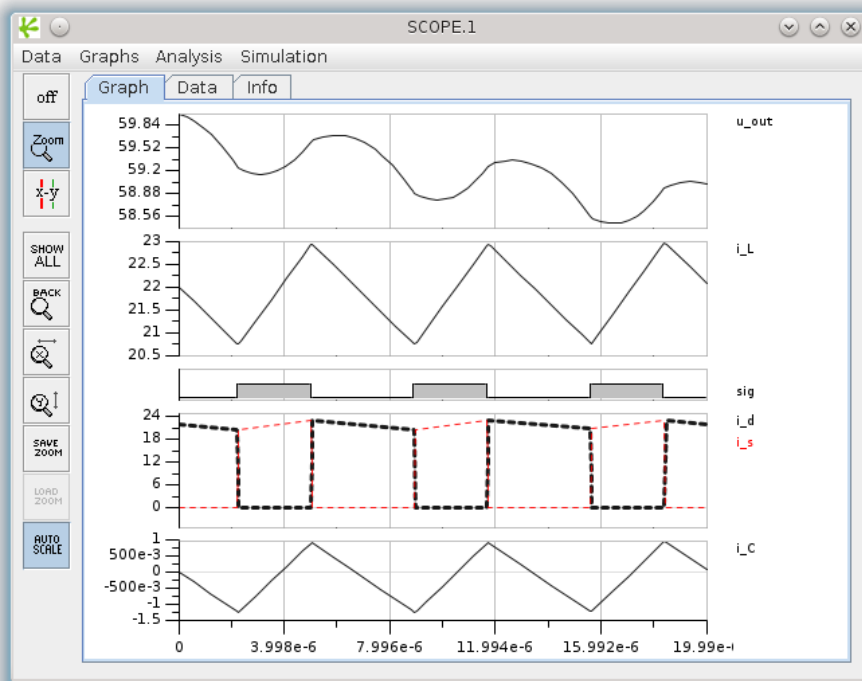


Рис. 5. Диаграммы моделирования конвертера

Получение диаграмм (осциллограмм) и есть суть работы с программами моделирования.

Ввод схемы

В папке с программой я не нашёл руководства пользователя, но есть то, что называется «Быстрый старт». Повторив всё описанное на примере сборки схемы, можно приступить к процессу самостоятельно. Итак:

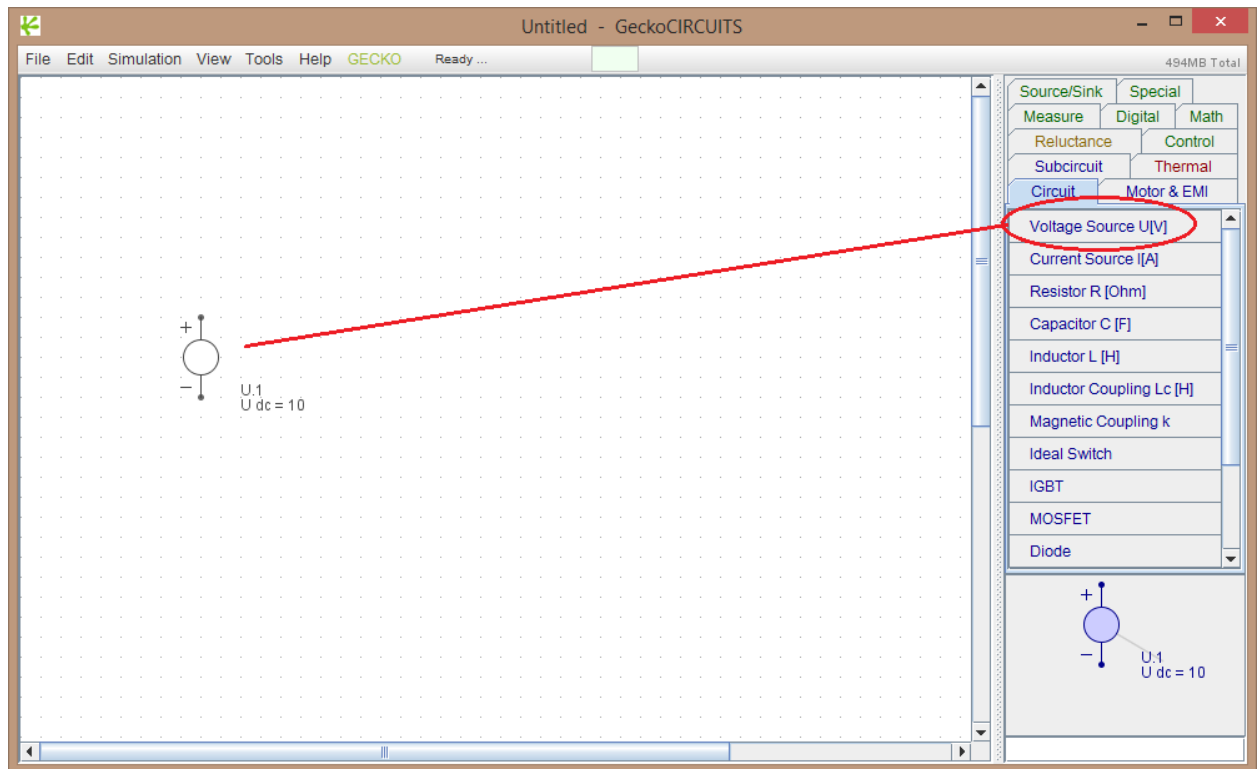


Рис. 1.1. Установка источника питания

На закладках в правом окне легко найти все категории компонентов программы. Выбрав компонент щелчком левой клавиши мышки, его можно перенести в нужное место на чертеже. При перемещении компонент движется вслед за курсором мышки в виде, показанном в нижней части правого окна. В нужном месте чертежа достаточно повторно щёлкнуть левой клавишей мышки, чтобы компонент, сменив цвет на синий, остался неподвижен. Любой, кто работал с другими программами моделирования электрических цепей, знаком с этой процедурой.

Если при перемещении компонента в рабочее поле чертежа нажимать правую клавишу мышки, то компонент будет вращаться. Его вид в правом окне тоже повторяет это движение, а следующий выбор компонента перенесёт его в рабочее поле уже в «повёрнутом» виде. Впрочем, если вы поместили компонент на чертёж, если вас не устраивает его вид, то, выделив его щелчком левой клавиши мышки, вы можете правой клавишей мышки повернуть его так, как вам нужно.

И ещё, собрав часть схемы, вы можете пожалеть, что не сдвинули компоненты чуть правее или левее (выше или ниже). В этом случае щелчком левой клавиши мышки по свободному месту чертежа вы можете начать процесс выделения совокупности компонентов (прямоугольное выделение), удерживая левую клавишу – достаточно провести курсором мышки в другое свободное место, оранжевый прямоугольник указывает область выделения, где отпустить клавишу мышки. Выделенные компоненты меняют цвет. Щёлкните по любому из них левой клавишей мышки и, удерживая левую клавишу, переместите компоненты в другое место на чертеже.

Конвертеры в программе GeckoCIRCUITS

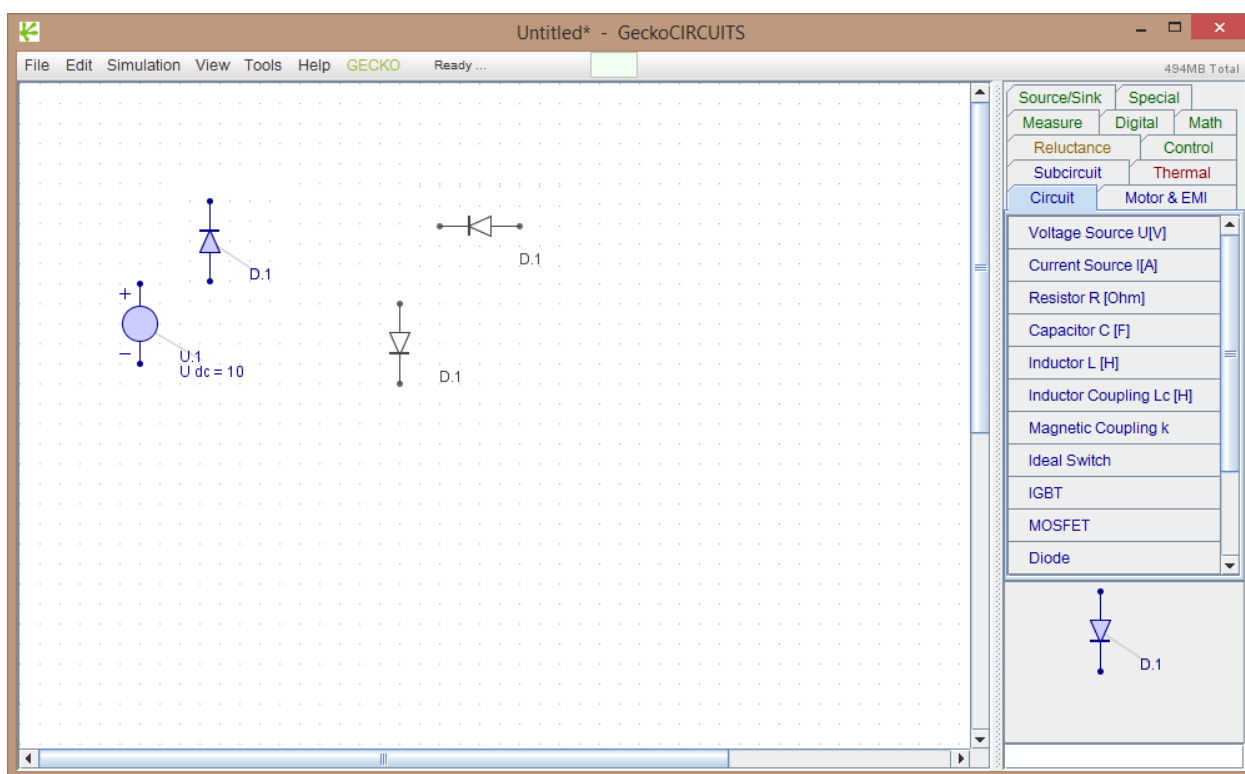


Рис. 1.2. Поворот компонента при перемещении

Надписи компонентов могут мешать, как показано слева ниже:

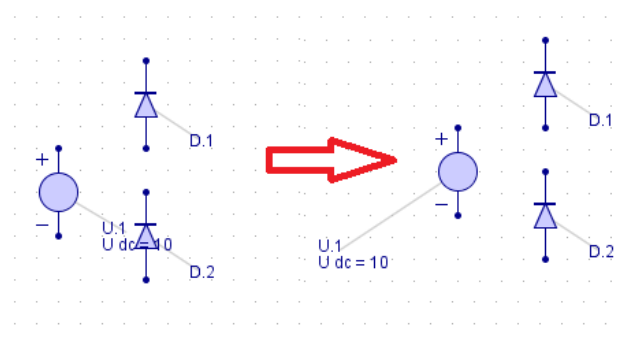


Рис. 1.3. Параметры источника перекрывают изображение диода

Достаточно щёлкнуть по надписи, как по любому компоненту, и перенести её в удобное место, где повторно щёлкнуть левой клавишей мышки.

Добавлю, что в том случае, когда вы передумали добавлять компонент на чертёж во время его переноса, вы можете отказаться от операции, нажав клавишу **Esc** на клавиатуре.

Все компоненты, как в любой другой программе, имеют свойства. Нам нужен источник напряжения частотой 60 Гц и напряжением 220 В (или 200 В, как в оригинале). Двойной щелчок по источнику напряжения даёт нам возможность выбрать вид (по умолчанию источник постоянного напряжения) и задать параметры. Закладки открывают диалоговое окно для ввода параметров.

Конвертеры в программе GeckoCIRCUITS

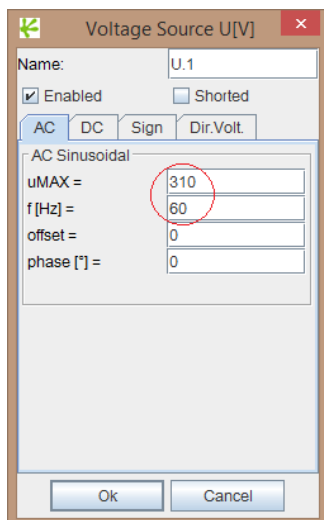


Рис. 1.4. Настройка источника питающего напряжения

Эти процедуры знакомы тем, кто работал с аналогичными программами. Но в нашем случае есть особенность. В оригинальной схеме есть источник треугольного напряжения, подключенный к инверсному входу компаратора. На закладках диалога настройки источника напряжения вы не найдёте ни прямоугольного, ни треугольного напряжения, что в других программах есть в том же диалоговом окне. Но здесь на закладке вы можете увидеть надпись *Sign*. Она относится к управляемому источнику напряжения.

Я уже говорил, что на закладках правого окна отмечены разные категории компонентов. Среди них найдём нужный компонент.

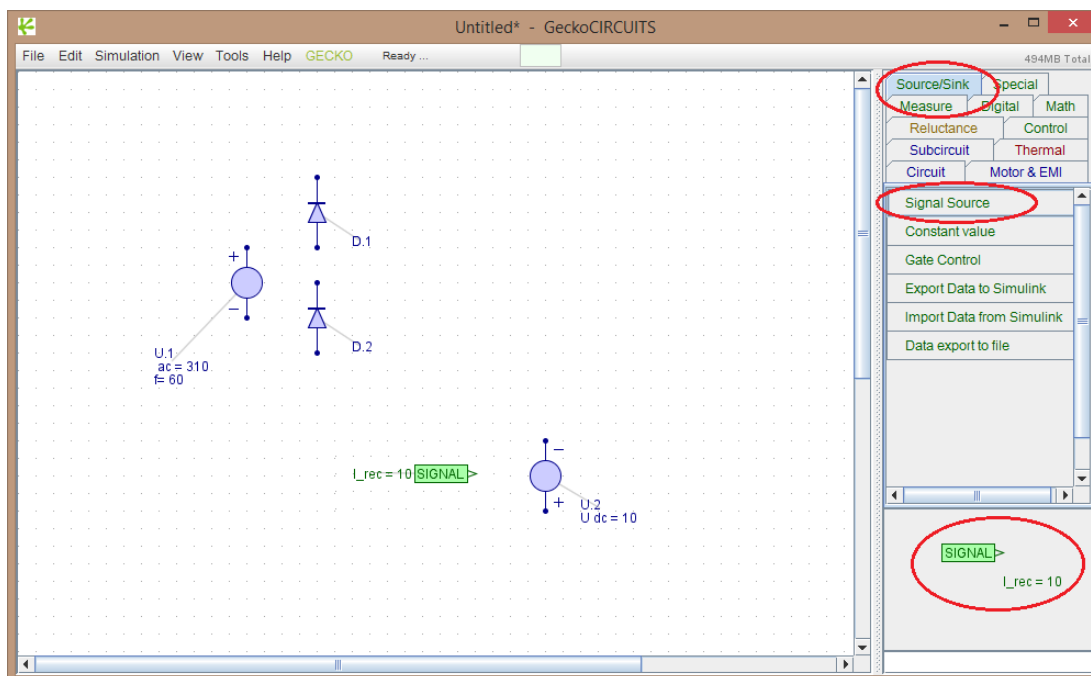


Рис. 1.5. Добавление управляющего компонента для создания источника треугольных импульсов

В диалоговом окне настройки этого управляющего компонента мы и найдём нужную форму сигнала, а попутно зададим все параметры.

Конвертеры в программе GeckoCIRCUITS

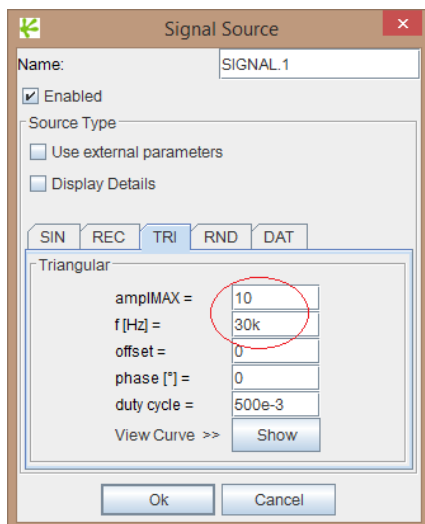


Рис. 1.6. Диалоговое окно настройки управляющего компонента

Теперь двойным щелчком левой клавиши мышки по правому краю этого компонента откроем окно маркировки узлов схемы. В этом окне присвоим имя узлу, скажем, *tri*.

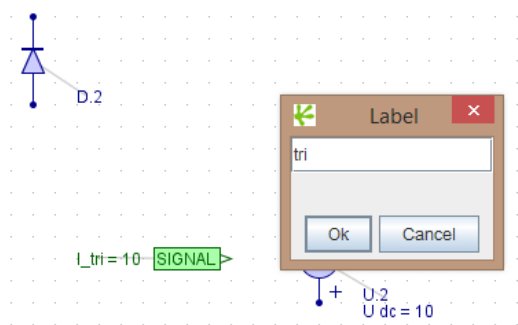


Рис. 1.7. Маркировка выхода управляющего компонента

Клавишей **OK** подтвердим свой выбор. Откроем двойным щелчком левой клавиши мышки по первому источнику диалоговое окно свойств, выберем вид *Sign*, где откроем окошко (кнопкой стрелка вниз или щелчком по окошку), в котором можно выбрать управляющий узел.

Позже мы вернёмся к подобным процедурам, поскольку мы ещё не закончили подключение источника треугольного напряжения, мало того, у нас ещё много компонентов, которые предстоит настроить. И, забегая вперёд, вы увидите изменения в описании источника.

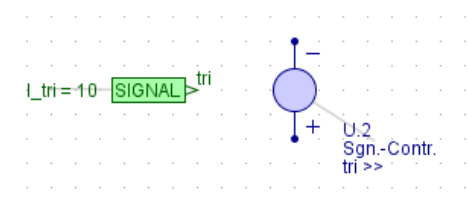


Рис. 1.8. Завершённый выбор вида сигнала источника

Конвертеры в программе GeckoCIRCUITS

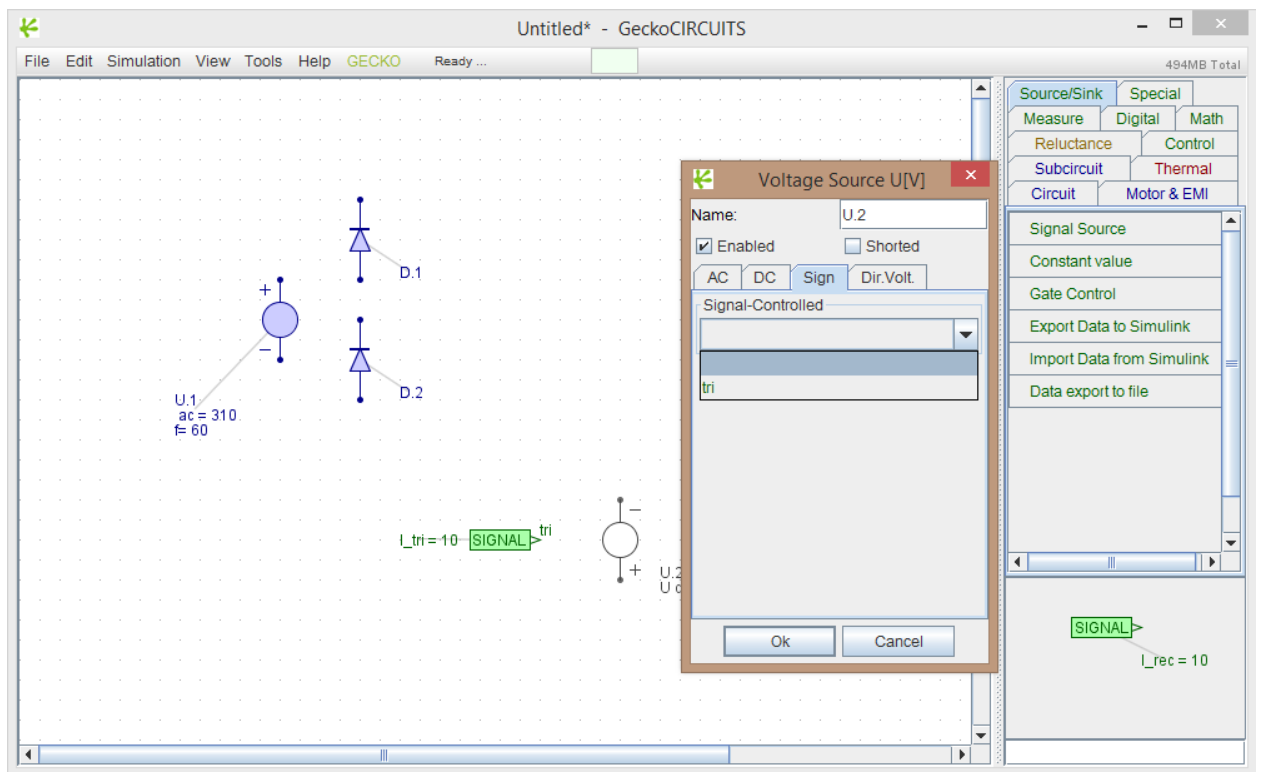


Рис. 1.9. Выбор узла управления для источника треугольного напряжения

На данный момент мы знаем, как выбрать и установить все электрические компоненты схемы. Завершив их установку, можно приступить к соединению. Для этого щёлкните правой клавишей мышки в любом месте, курсор изменит свой вид.

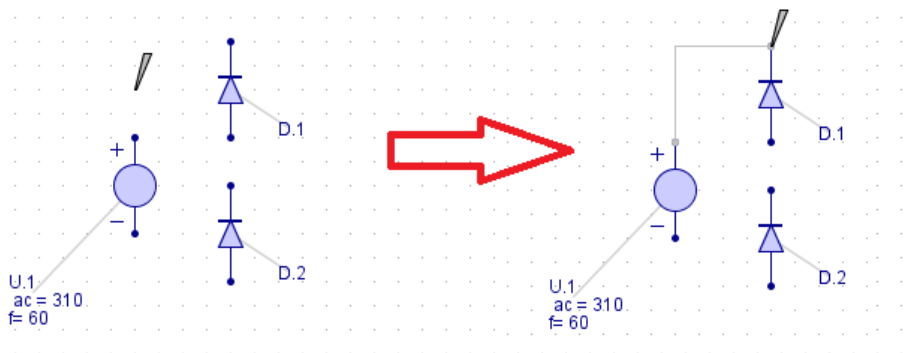


Рис. 1.10. Начало работы по соединению электрических компонентов

Подведите курсор к узлу, скажем, источника переменного напряжения, щёлкните левой клавишей; переместите курсор к другому узлу, где повторно щёлкните левой клавишей; соединение проведено. Осталось повторить эту операцию для соединения всех элементов электрической схемы. Чтобы выйти из режима проведения соединений достаточно щёлкнуть правой клавишей мышки.

Иногда узлы схемы не соединить прямой линией или нужно сделать лишний перегиб в соединении. Можно создать промежуточный узел, то есть, щёлкнуть левой клавишей мышки не в конечной точке, а на пустом месте, а затем повторить соединение, используя эту промежуточную точку. А в том случае, когда соединение совсем не «задалось», его можно удалить. Щёлкните левой

Конвертеры в программе GeckoCIRCUITS

клавишей мышки по ненужному элементу схемы или проводнику, выделив его, а затем нажмите клавишу **Delete** на клавиатуре. Этим вы удалите то, что хотите удалить. При этом, если вы перемещаете мышку после щелчка, то выделенный элемент остаётся привязан к курсору мышки, следовательно, его можно переместить в другое место.

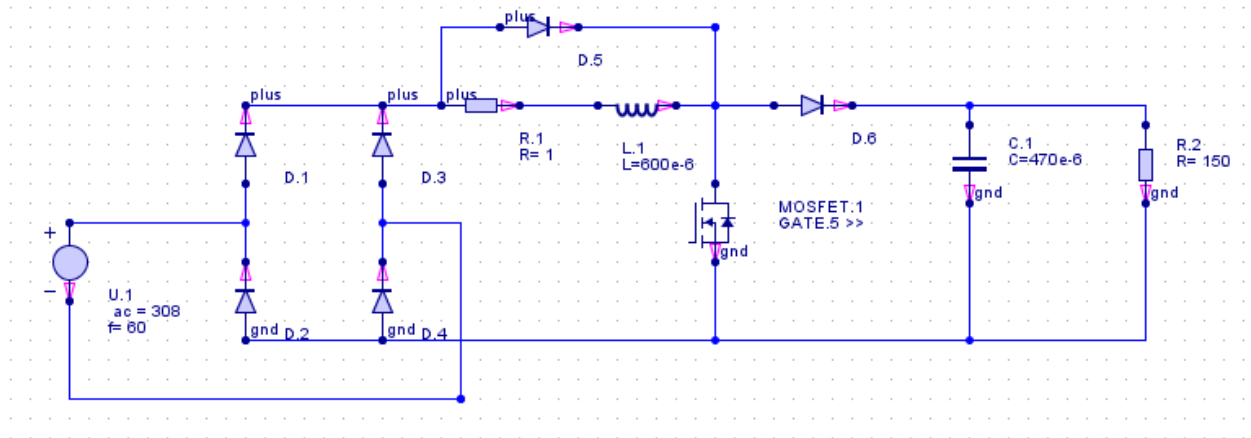


Рис. 1.11. Электрическая схема конвертера

Вы сразу можете отметить отсутствие «земли» на схеме. Но можете отметить и маркировку *gnd*. Это метка, которой мы воспользуемся дальше, и подобную метку вы можете сделать для любого узла схемы и соединения. Процедура аналогична той, что описана для маркировки выхода управляющего компонента. Заметьте, что маркировка носит относительный характер.

Кроме электрической схемы в оригинале есть схема управления. Без схемы управления электрическая схема правильно работать не будет.

Со схемой управления сразу хочу отметить, что сенсоров напряжения и тока из оригинальной схемы в данной программе я не нашёл. Но датчик напряжения, как мне пояснили, повторяет напряжение в точках, к которым он подключен, а элемент управления, обозначенный на исходной схеме как *absolute value*, даёт абсолютное значение этого напряжения. Поэтому я считаю, что можно заменить сенсор напряжения измерительным вольтметром, который есть в программе. А подключить его к выпрямителю.

Насколько правильна и правильна ли вообще подобная замена сейчас судить рано. Если результаты моделирования окажутся приемлемыми, то, вероятно, это правильно, иначе следует искать другой вариант.

Конвертеры в программе GeckoCIRCUITS

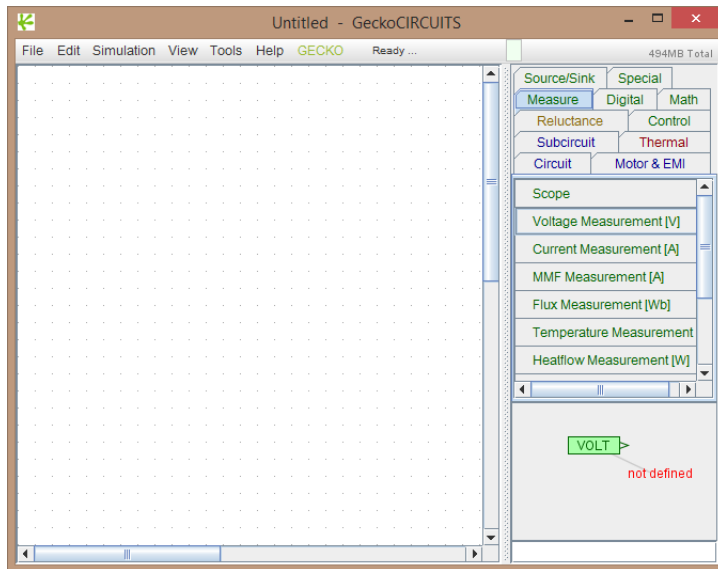


Рис. 1.12. Измерительный вольтметр на закладке *Measure*

Кстати, если кому-то интересно, можно запустить две копии программы, что я сейчас сделал. Но вернёмся к сенсору напряжения. Вольтметр нужно подключить. Для его подключения и были сделаны маркеры *plus* и *gnd*. Откройте свойства вольтметра двойным щелчком левой клавиши мышки, и вы увидите, что его можно подключить либо к узлам схемы, либо к компоненту схемы.

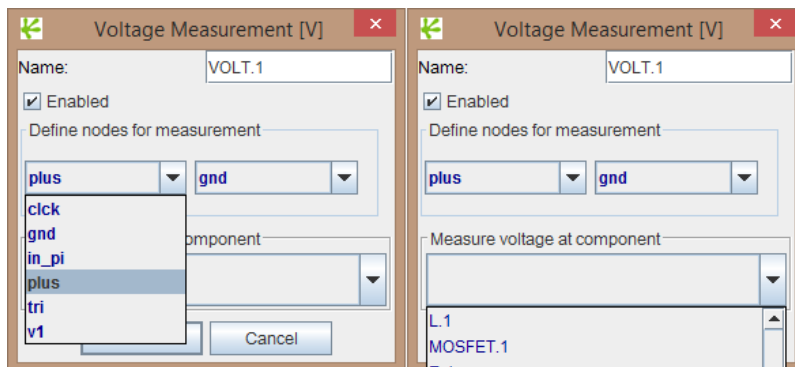


Рис. 1.13. Настройка первого датчика напряжения

При подключении к узлам схемы используются два верхних окошка, где эти узлы можно выбрать по данным им названиям, а для подключения к элементу схемы используется окошко ниже. Как мне пояснили, датчик настроен так, что он даёт напряжение меньшее, чем напряжение источника питания. Для этой цели можно использовать усилитель с усилением меньше единицы. На закладке *Control* есть компонент *GAIN*, который мы настроим, введя в диалоговом окне его свойств значение 40m. Теперь начало схемы управления выглядит так:

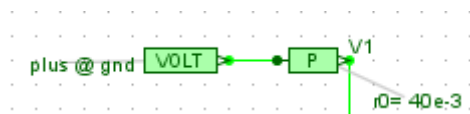


Рис. 1.14. Начальная часть схемы управления

В качестве второго сенсора напряжения, подключенного к выходному сопротивлению, используется второй вольтметр, который будет подключен к резистору R2.

Конвертеры в программе GeckoCIRCUITS

Датчик тока. Он даёт напряжение, которое пропорционально току в цепи. Не будучи полностью уверенным в своей правоте, я заменил его сопротивлением в 1 Ом и вольтметром, подключённым к этому сопротивлению.

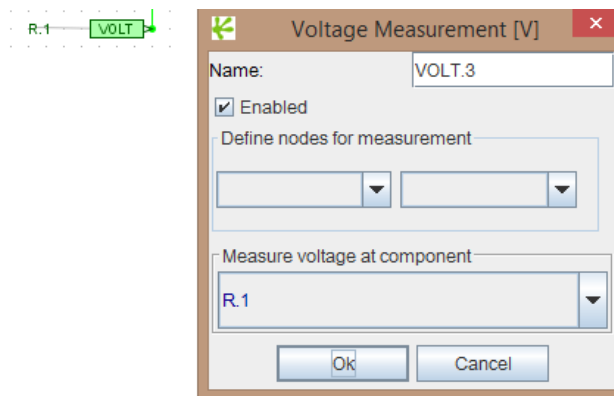


Рис. 1.15. Датчик тока

Такие компоненты, как сумматор (в схеме он вычитатель) или умножитель, есть среди компонентов закладки *Control*. Их достаточно перенести на чертёж и правильно соединить. Соединение схемы управления производится аналогично соединениям электрических элементов.

Немного меня, например, смутили элементы *PI* (пропорциональные интеграторы). Подобный интегратор, похоже, есть среди управляющих компонентов. Но без достаточной практики использования этих компонентов я не в полной мере понял их настройку, поскольку в свойствах компонента есть поясняющие формулы, которые меня и смутили.

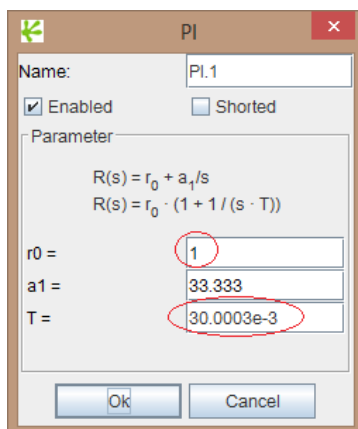


Рис. 1.16. Свойства компонента PI

Но, не мудрствуя лукаво, я использовал «r0» в качестве параметра усиления, а «Т» в качестве постоянной времени.

И ещё одно замечание. В оригинальной схеме используется источник постоянного напряжения в качестве опорного. Элементы электрические плохо сочетаются с управляющими элементами схемы, поэтому я использовал компонент *Constant value* из набора *Source/Sink* со значением 10.5.

Теперь эта часть схемы управления выглядит так:

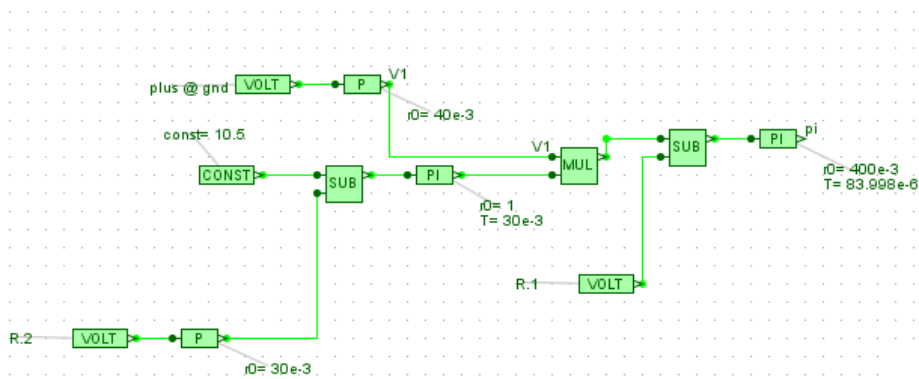


Рис. 1.17. Первая часть схемы управления

В программе я не нашёл компаратора. Использовал операционный усилитель. Его прямой вход должен подключаться к схеме управления в точке, обозначенной мной как πi (на рисунке выше). Но я уже говорил, что электрические компоненты (ОУ) плохо соединяются с управляющими. Однако в этом случае, если вспомнить, как мы сделали источник треугольного напряжения, на помощь приходит источник напряжения, управляемый напряжением. Подключив его к узлу πi , я соединю его с прямым входом ОУ. Получается такая конструкция:

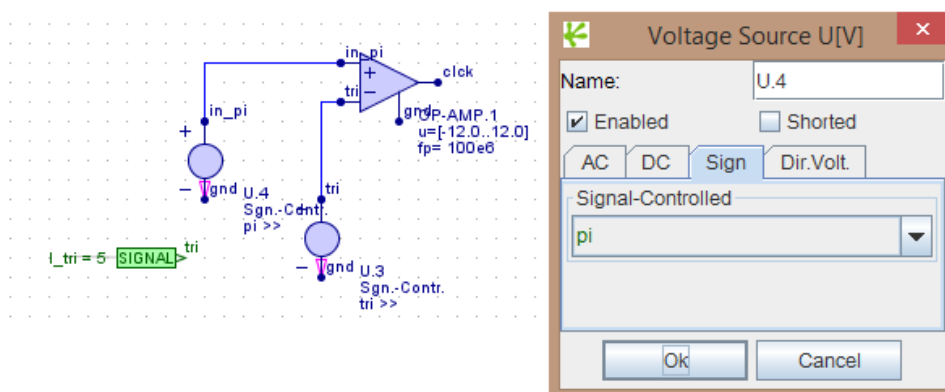


Рис. 1.18. Завершающая часть схемы управления

Справа показано диалоговое окно настройки соединения предыдущей части и источника управляемого напряжения. Работа со схемой управления почти закончена. Осталось соединить выход операционного усилителя с затвором полевого транзистора. Но, похоже, транзистор, хотя он находится в категории электрических компонентов, отчасти принадлежит к группе управления как управляемый элемент. Поэтому для оставшегося соединения я использую еще один вольтметр с настройкой подключения к выходу ОУ и элемент управления затвором, который находится в наборе *Source/Sink* и называется *Gate control*. В его свойствах выбирается транзистор.



Рис. 1.19. Соединение выхода ОУ с транзистором

Моделирование работы схемы

Чтобы моделировать работу схемы, следует увидеть осциллограммы в интересующих нас точках. Среди измерительных инструментов (*Measure*) есть осциллограф.

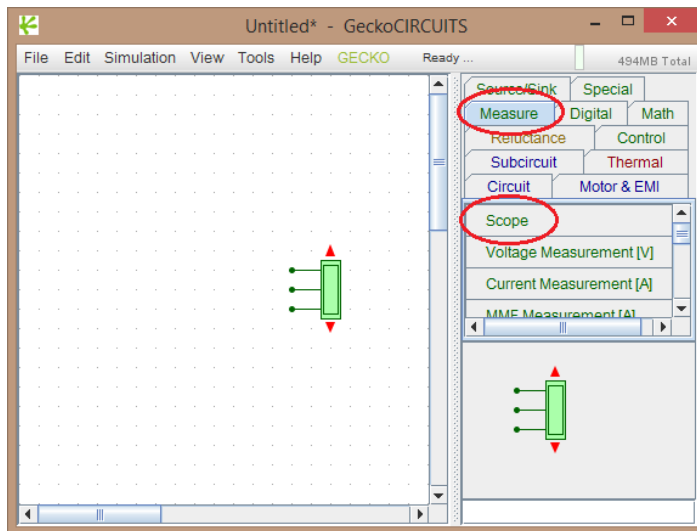


Рис. 2.1. Добавление осциллографа

Исходно можно видеть три входа, но стрелками (вверху и внизу) количество входов можно увеличить или уменьшить. Меня сейчас интересует наблюдение за напряжением на входе, на выходе и ток через индуктивность. Поэтому входы осциллографа подключаем через два вольтметра и один амперметр.

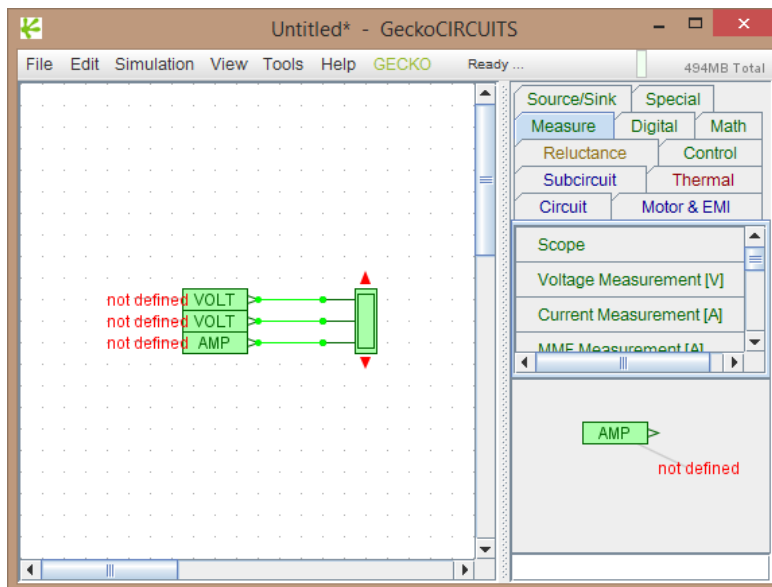


Рис. 2.2. Добавление сопряжения с точками наблюдения

Сейчас элементы сопряжения ещё не привязаны к элементам схемы, но мы выше обсудили, как это сделать. Единственное, что осталось добавить – в качестве входного выбрано напряжение, отмеченное на схеме управления как V1. На входах осциллографа следует обозначить сигналы, иначе они появятся с названиями, присвоенными им автоматически.

Конвертеры в программе GeckoCIRCUITS

Полная схема выглядит так:

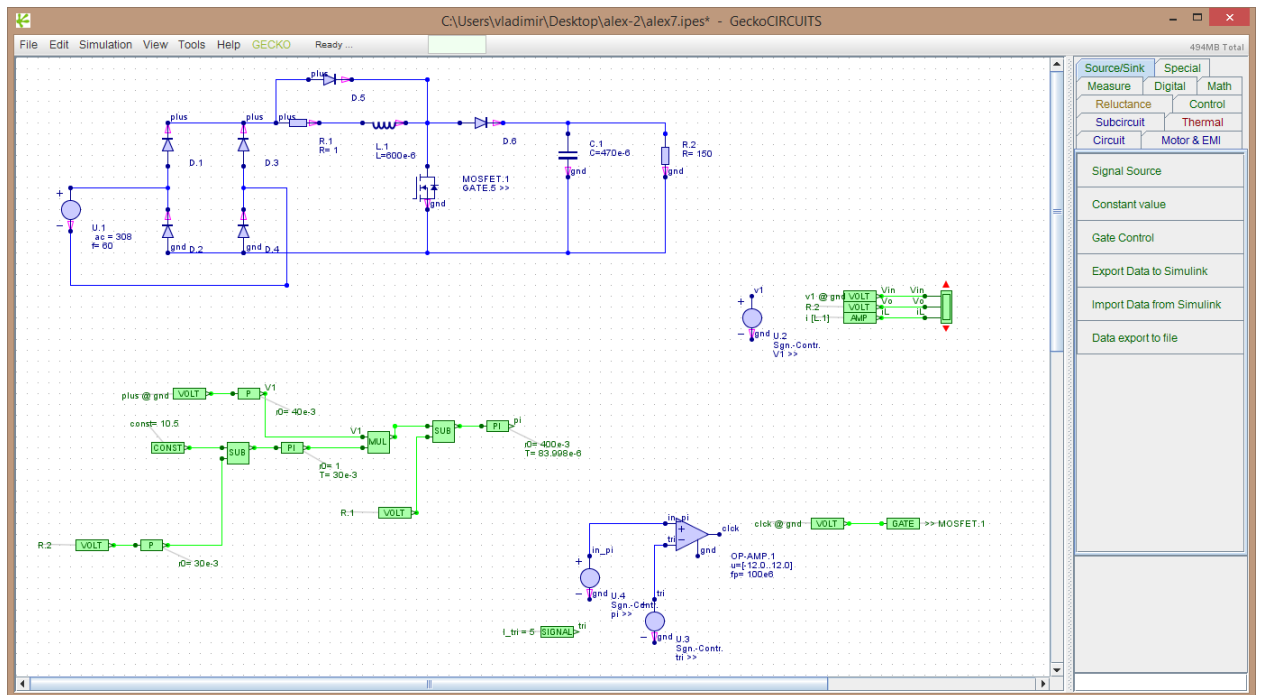


Рис. 2.3. Полная схема в программе GeckoCIRCUIT

Чтобы увидеть графику, дважды щёлкните по осциллографу.

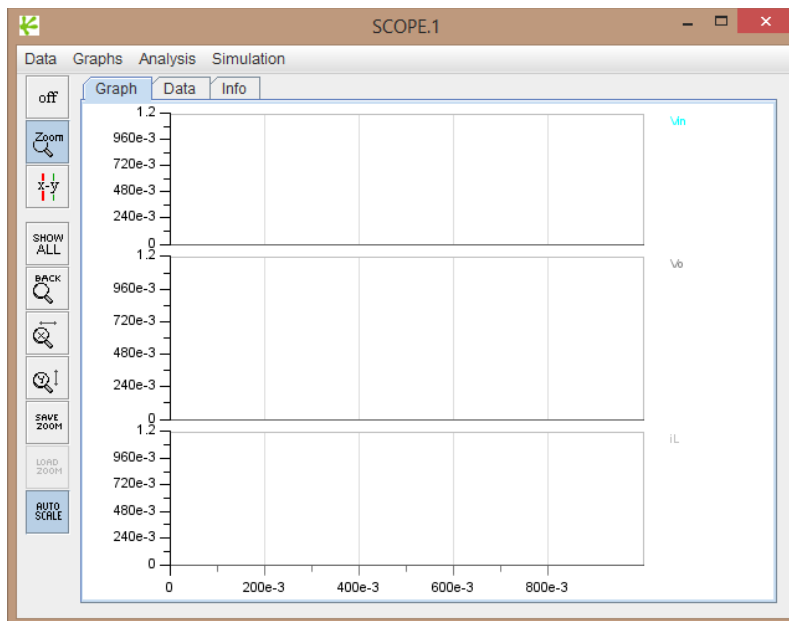


Рис. 2.4. Графическое окно программы

У графического окна есть своё меню. Так в разделе *Simulation* вы найдёте настройки параметров симуляции (например, время наблюдения) и запуск симуляции. Задав время наблюдения в 1 секунду, посмотрим на результаты предыдущей работы.

Конвертеры в программе GeckoCIRCUITS

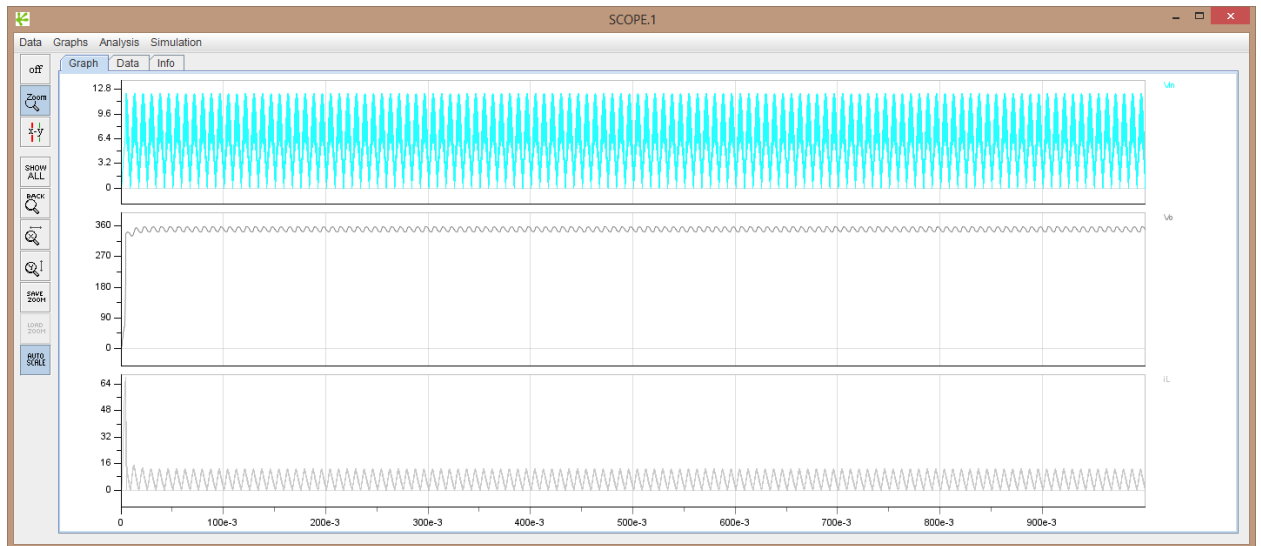


Рис. 2.5. Осциллограммы при моделировании

Почему я выбрал для наблюдения эти точки? Вот осциллограммы, полученные в другой программе, которые я и хотел сравнить с вышеприведёнными.

Ниже выходное напряжение и ток через индуктивность имеют иной порядок, чем на нашей осциллограмме.

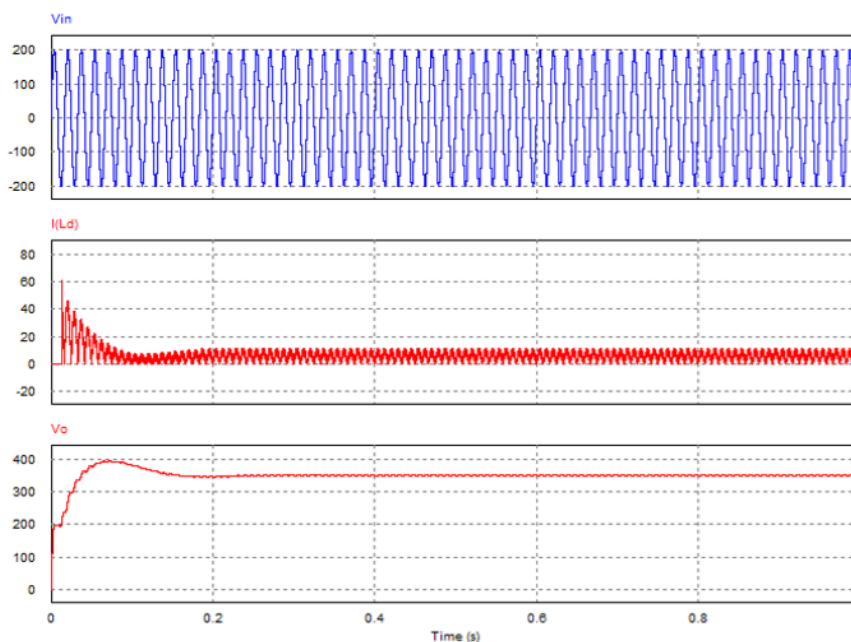


Рис. 2.6. Осциллограммы оригинальной схемы

Но согласитесь, результаты моделирования похожи. И, главное, моделирование не прервалось с сообщением, что решение расходится или матрица сингулярна.

Конвертеры в программе GeckoCIRCUITS

К концу рассказа мы ещё поговорим о том, правильно ли моделировалась схема? А сейчас я приведу осциллограммы, полученные при изменении «конструкции» датчика напряжения, при его подключении к источнику напряжения (через дополнительный мост).

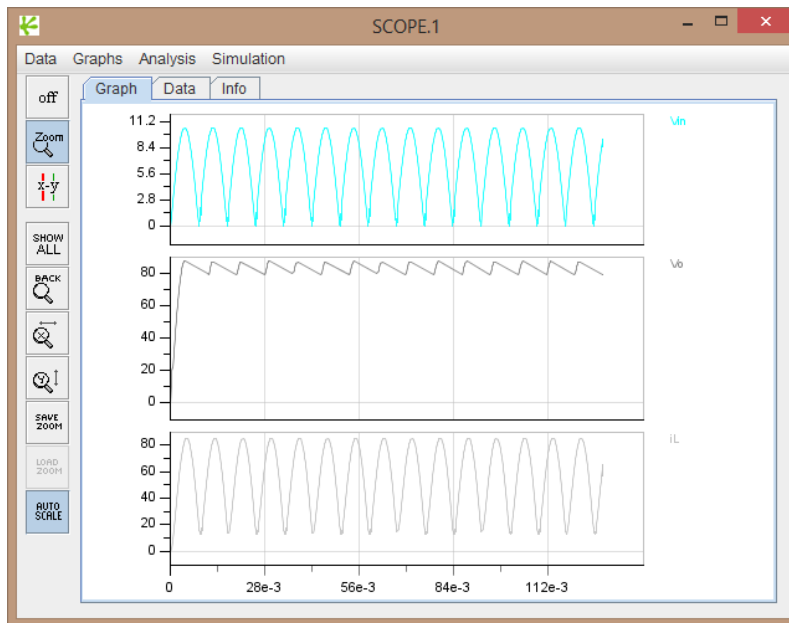


Рис. 2.7. Осциллограммы модифицированной схемы

Изменилось выходное напряжение и ток через индуктивность. В причинах этого следует, конечно, разбираться, но тогда, когда будет ясно, что моделируется схема правильно.

В сущности, выше я повторил, изменив героя повествования, то, что написано в «Быстром старте». И программа имеет ещё много возможностей, о которых я не упоминал. Чтобы узнать больше о программе, следует больше попользоваться ею. Хотя я, признаюсь, используя текстовый процессор, знаю 10-15% из его возможностей и не ощущал необходимости знать его досконально. Если что-то нужно, чего я не знаю, я ищу в справке или в Интернете, но, как правило, хватает того, чем я пользуюсь регулярно. Но это так, к слову.

А прежде, чем оценивать работу программы, следует понять...

Как работает повышающий конвертер

В Интернете можно найти много теоретических статей по работе конверторов. Я воспользуюсь несколькими статьями, чтобы проиллюстрировать их с помощью программы GeckoCIRCUITS.

Конвертеры, подобные вышеописанному, используются для импульсного преобразования напряжения с целью его повышения или понижения. Для повышающего преобразования используется сложение питающего напряжения с реактивным напряжением на индуктивности.

Вот один из примеров в папке программы:

Конвертеры в программе GeckoCIRCUITS

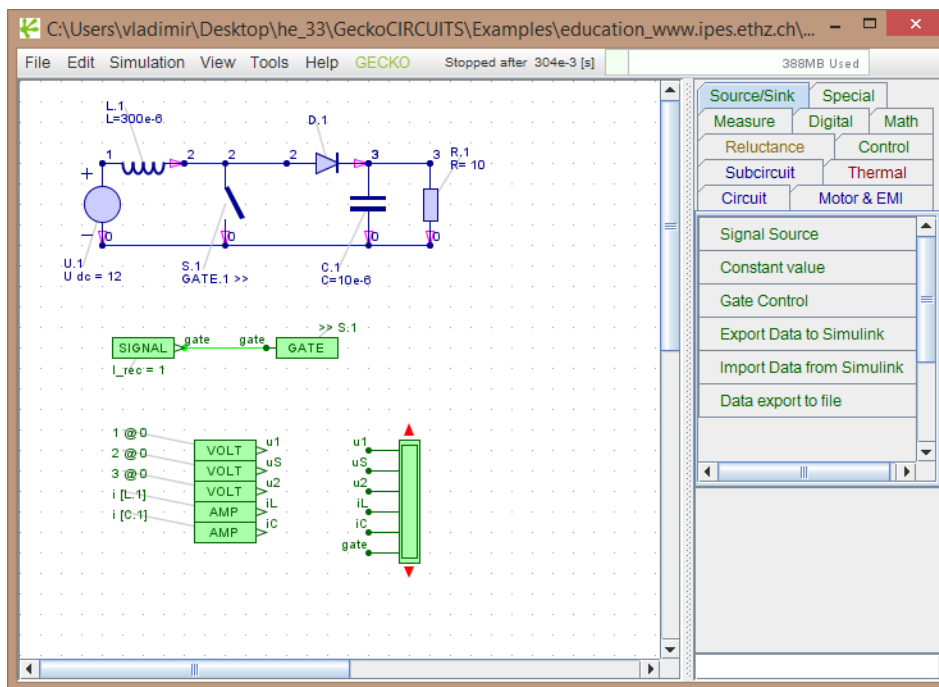


Рис. 3.1. Пример преобразователя

И осциллограммы точек наблюдения, выбранные авторами программы. Отмечу попутно, что измерительные инструменты в примере не соединяются с входами осциллографа, но отмечены одинаковыми названиями. Видимо, и такой приём удобен при работе.

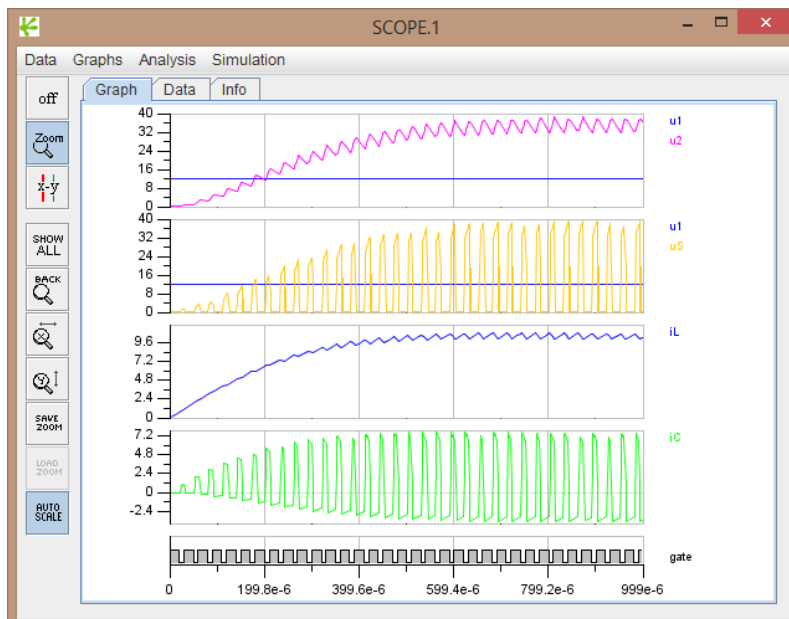


Рис. 3.2. Осциллограммы моделирования схемы

Отмеченное на верхней диаграмме напряжение u_2 – это выходное напряжение схемы, оно больше напряжения питания, то есть, конвертер повышающий. При работе ключа S_1 на индуктивности возникает эдс самоиндукции, которая и прикладывается к конденсатору C_1 совместно с питающим напряжением.

Конвертеры в программе GekoCIRCUITS

Более подробно с небольшим включением математики для установившегося режима работы это выглядит так. Положим, что ключ $S1$ замкнут. Падение напряжения на индуктивности равно по величине питающему напряжению. Напряжение и ток индуктивности связаны формулой $V1 = -L \cdot di/dt$. Интеграл от этого выражения даст значение тока через индуктивность $I = V1 \cdot t / L$. Это линейная зависимость тока от напряжения. Крутизна графика зависит от величины напряжения и индуктивности. Когда ключ размыкается, напряжение в точке соединения индуктивности и ключа будет некоторым $V2$ (практически, это выходное напряжение), а на индуктивности равным разности $V1$ и $V2$. Значит, зависимость тока от времени теперь становится $I = (V1 - V2) \cdot t / L$. Запишем время включённого ключа как T_{on} , а выключенного как T_{off} . В первый промежуток времени ток линейно нарастает, а во второй линейно спадает. Напряжение в этой точке для времени T_{on} равно нулю (ключ замкнут). За период работы ключа можно для среднего значения напряжения записать: $V2 \cdot T_{off} = V1(T_{on} + T_{off})$. То есть, для напряжения $V2 = V1(T_{on} / T_{off} + 1)$. Выходное напряжение зависит от входного напряжения и скважности управляющих импульсов. И заметьте, что оно больше входного.

Проверим это утверждение, изменив скважность управляющих импульсов. На схеме откроем свойства компонента *signal*.

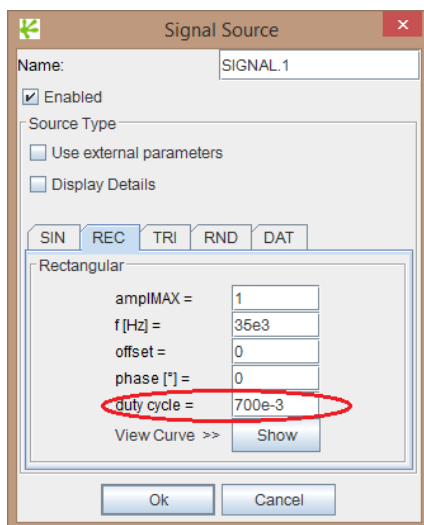


Рис. 3.3. Настройка управляющего генератора

Повторим симуляцию при параметре 700m, а затем изменим этот параметр на 300m.

Конвертеры в программе GeckoCIRCUITS

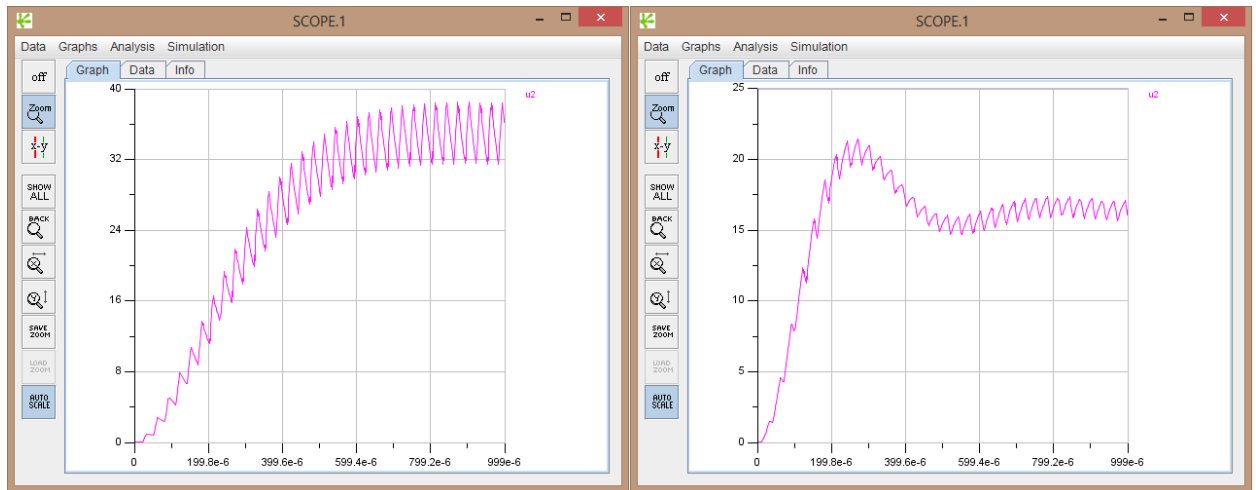


Рис. 3.4. Выходное напряжение при двух значения скважности управляющих импульсов

Изменим входное напряжение, скажем, задав 20 В.

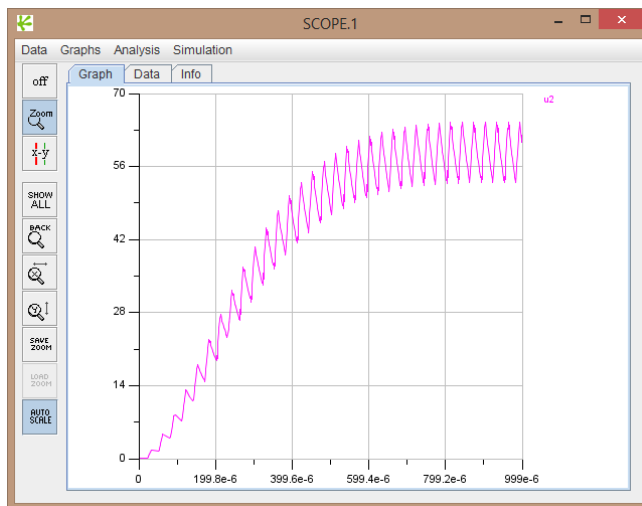


Рис. 3.5. Выходное напряжение при напряжении источника питания 20 В

Рассмотрим ещё один параметр схемы – ток через индуктивность. Вот установившийся режим:

Конвертеры в программе GeckoCIRCUITS

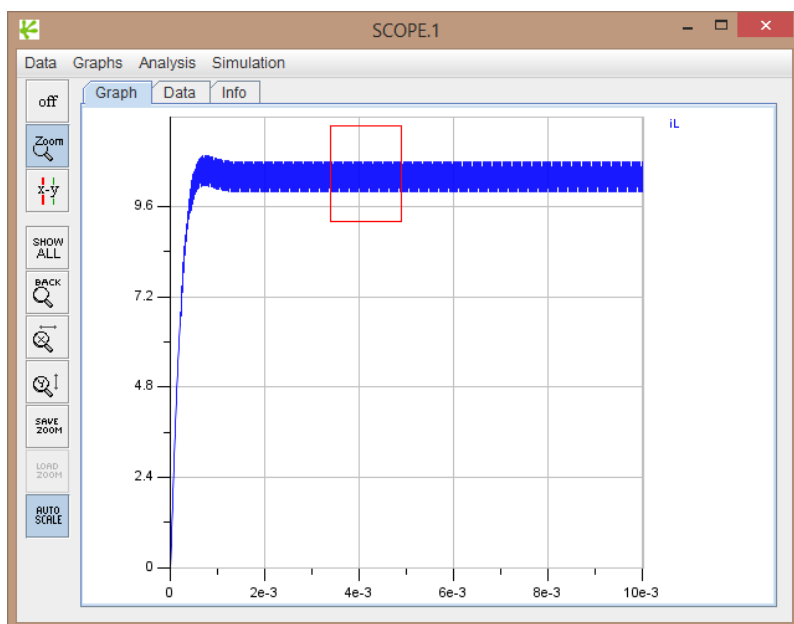


Рис. 3.6. Осциллограмма тока через индуктивность

Графическое окно позволяет рассмотреть отдельные области осциллограммы. Достаточно мышкой выделить прямоугольную область. Вид графика изменится:

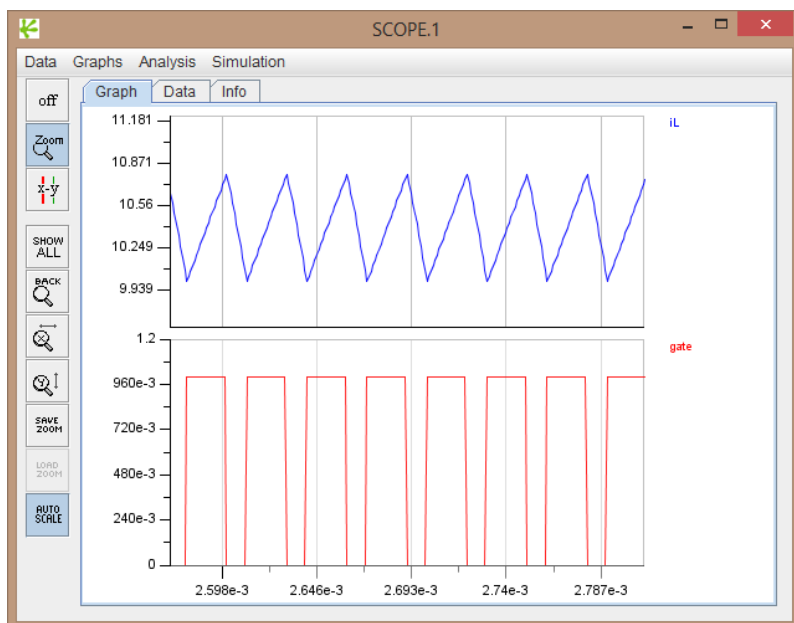


Рис. 3.7. Выделенная область графика

Для возвращения к первоначальному виду графиков используйте кнопку **BACK**.

Напомню, что графическое окно имеет своё меню. Используя раздел *Graphs*, можно удалить или добавить графики точек наблюдения, используя кнопки **Add graph** и **Delete graph**. В верхней части этого диалога все наблюдаемые величины.

Конвертеры в программе GescoCIRCUITS

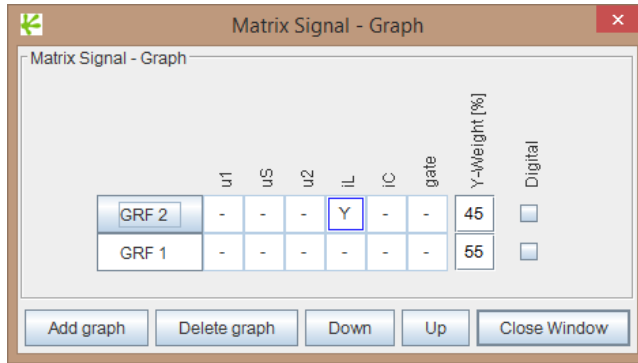


Рис. 3.8. Управление графиками в графическом окне

Вернёмся к току. Ток линейно нарастает до максимального значения I_{max} и спадает до минимального I_{min} . Под диаграммой тока представлена диаграмма управления ключом S1. Ток нарастает за время T_{on} , а спадает за время T_{off} .

Средний ток от источника (входной) можно записать так: $I_{in} = (I_{max} + I_{min})/2$. Для выходного тока справедливо $I_{out} \cdot (T_{on} + T_{off}) = T_{off} \cdot (I_{max} + I_{min})/2$, то есть:

$I_{out} = (I_{max} + I_{min}) / (2 \cdot (1 + T_{on}/T_{off}))$, а между токами и напряжениями (входными и выходными) есть связь: $I_{out}/I_{in} = V_{in}/V_{out}$. В итоге можно для максимального и минимального токов записать:

$I_{max} = I_{in} + 1/2 \cdot T_{on} \cdot V_{in}/V_{out}$, $I_{min} = I_{in} - 1/2 \cdot T_{on} \cdot V_{in}/V_{out}$, а с учётом того, что $I_{in} = I_{out} \cdot V_{out}/V_{in}$, $T_{on} = 1/f - T_{off}$, можно записать:

$$I_{max} = I_{out} \cdot \frac{V_{out}}{V_{in}} + \frac{1}{2fL} \cdot V_{in} \cdot \left(1 - \frac{V_{in}}{V_{out}}\right)$$

$$I_{min} = I_{out} \cdot \frac{V_{out}}{V_{in}} - \frac{1}{2fL} \cdot V_{in} \cdot \left(1 - \frac{V_{in}}{V_{out}}\right)$$

Этим формулам можно придать более удобный вид, если обозначить $V_{in}/V_{out} = k$.

$$I_{max} = \frac{I_{out}}{k} + \frac{V_{out}}{2fL} \cdot k \cdot (1 - k)$$

$$I_{min} = \frac{I_{out}}{k} - \frac{V_{out}}{2fL} \cdot k \cdot (1 - k)$$

Мы ранее меняли выходное напряжение, изменив скважность управляющих импульсов, посмотрим, изменились ли в этом эксперименте максимальное и минимальное значения тока.

Конвертеры в программе GeckoCIRCUITS

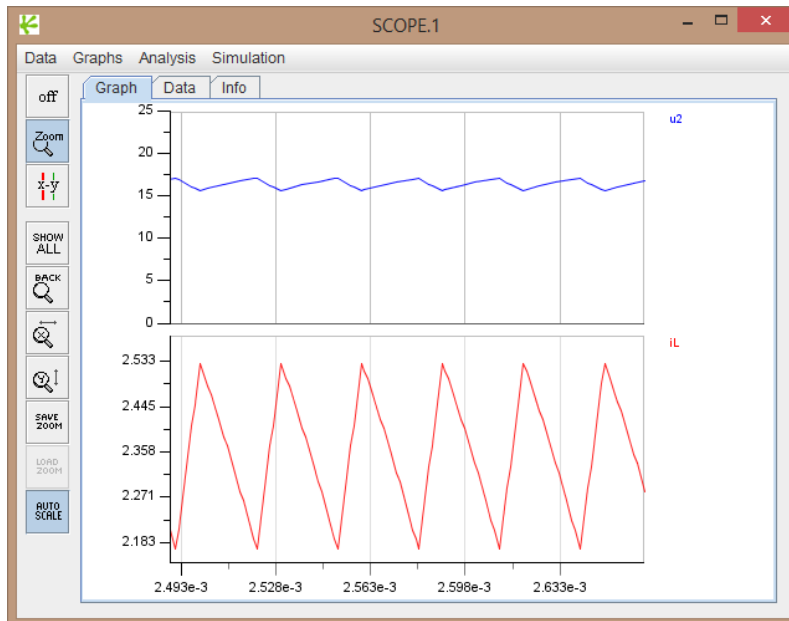


Рис. 3.9. Максимальный и минимальный ток при другом значении V_{out}

Значение минимального тока (и максимального) зависит от выходного тока. Выше мы видели, что значение минимального тока не достигает нуля. Такой режим работы называют режимом непрерывного тока. Но последовательно уменьшая выходной ток (увеличивая сопротивление нагрузки), мы можем достичь ситуации, когда выходной ток в некоторые моменты равен нулю. Такой режим работы называют режимом с прерывистым током.

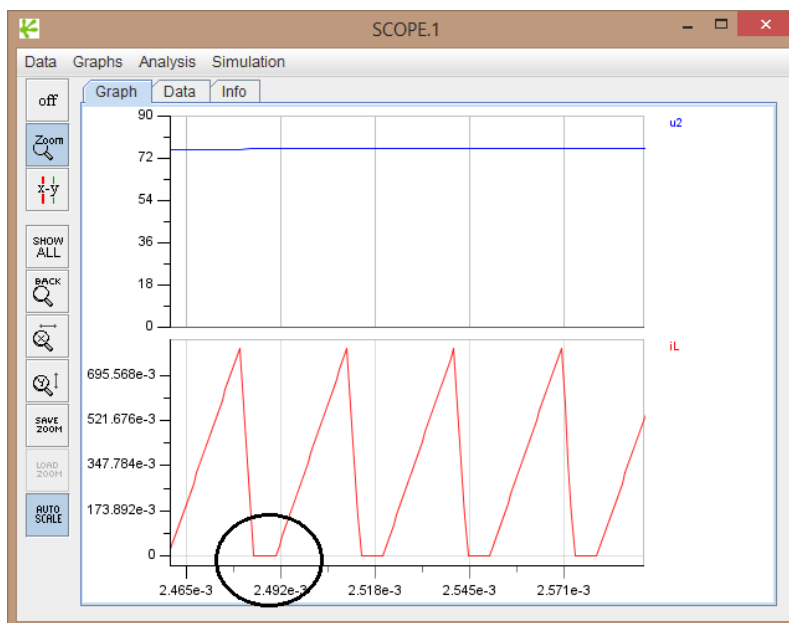


Рис. 3.10. Режим работы с прерывистым током

Многих может интересовать, как рассчитать преобразователь? В этой программе не помощник. Вот ссылка на сайт, где вы найдёте рекомендации по расчёту: <http://radioham.ru/teory/boost2.htm>

А проверить результат расчёта можно в программе.

Ещё немного о программе GeckoCIRCUITS

Посмотрим, что в основном меню программы.

Раздел *File*.

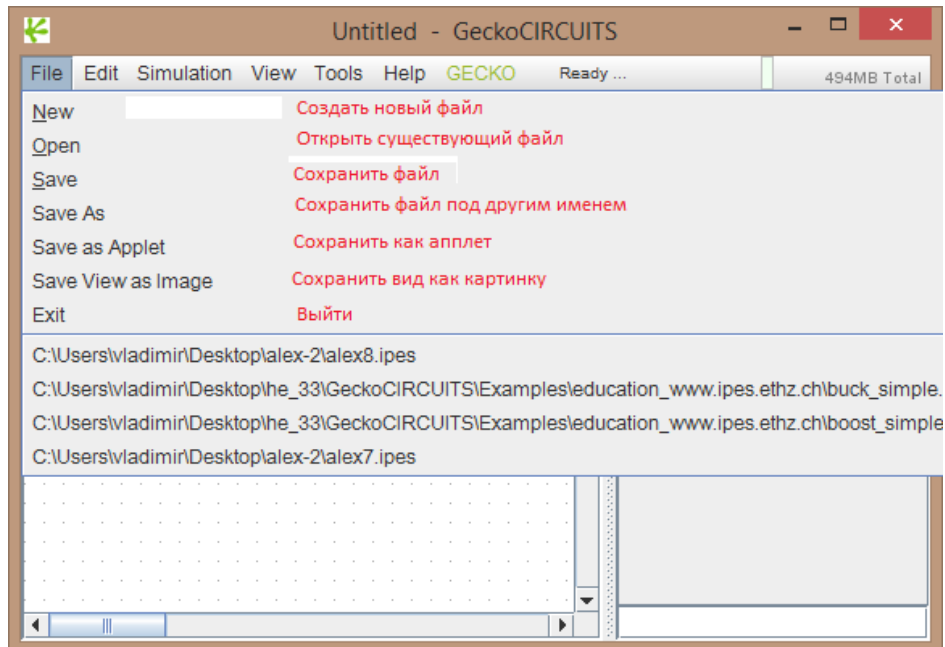


Рис. 4.1. Работа с файлами

При сохранении схемы в виде картинки можно выбрать формат изображения и место его хранения. Выбор места сохранения и имени файла – щелчок по папке справа.

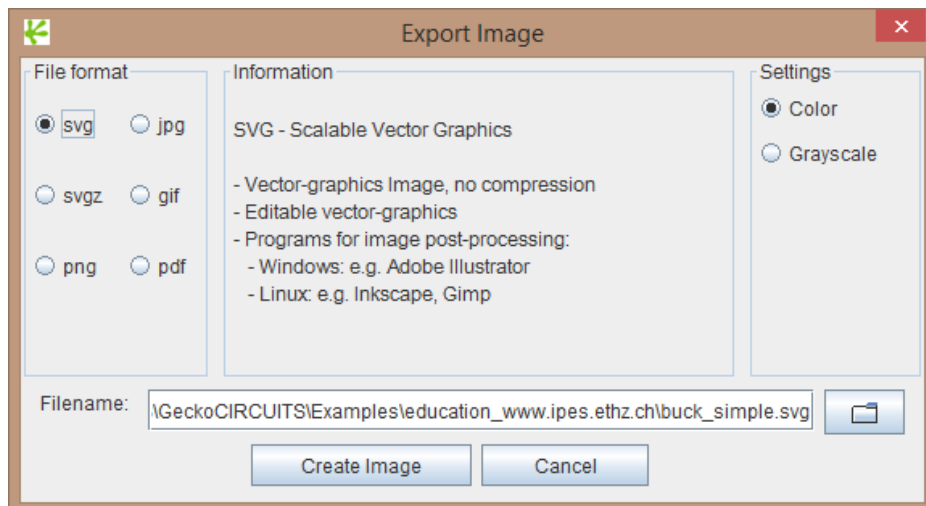


Рис. 4.2. Сохранение изображения

Раздел редактирования *Edit*. Многие команды можно выполнить и другими средствами.

Конвертеры в программе GeckoCIRCUITS

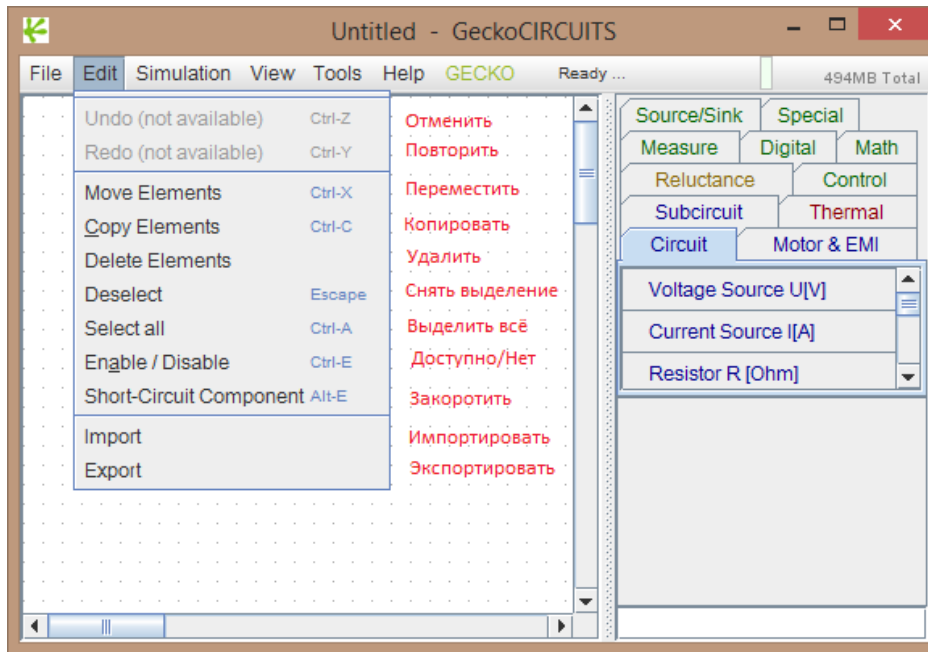


Рис. 4.3. Редактирование

Для выполнения команды *Move Elements* следует выполнить прямоугольное выделение на схеме, а затем выполнить команду. Выделенная часть схемы оказывается привязана к курсору мышки, и её легко переместить в другую часть чертежа, где достаточно щелчка левой клавиши мышки.

Аналогичная процедура для команд копирования и удаления. Определённый интерес представляет ряд команд. Рассмотрим команду *Short-Circuit Component* на простом примере.

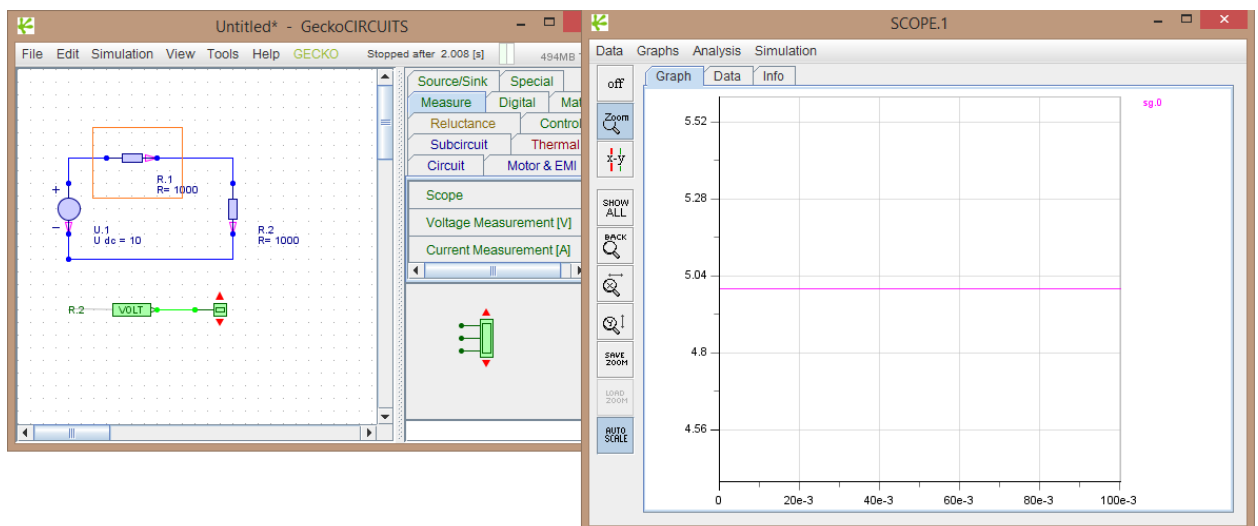


Рис. 4.4. Напряжение на резисторе R2

Теперь выделим (прямоугольным выделением, как показано выше) резистор R1 и используем команду. Выполним симуляцию, чтобы увидеть, что резистор R1 действительно закоротен.

Конвертеры в программе GeckoCIRCUITS

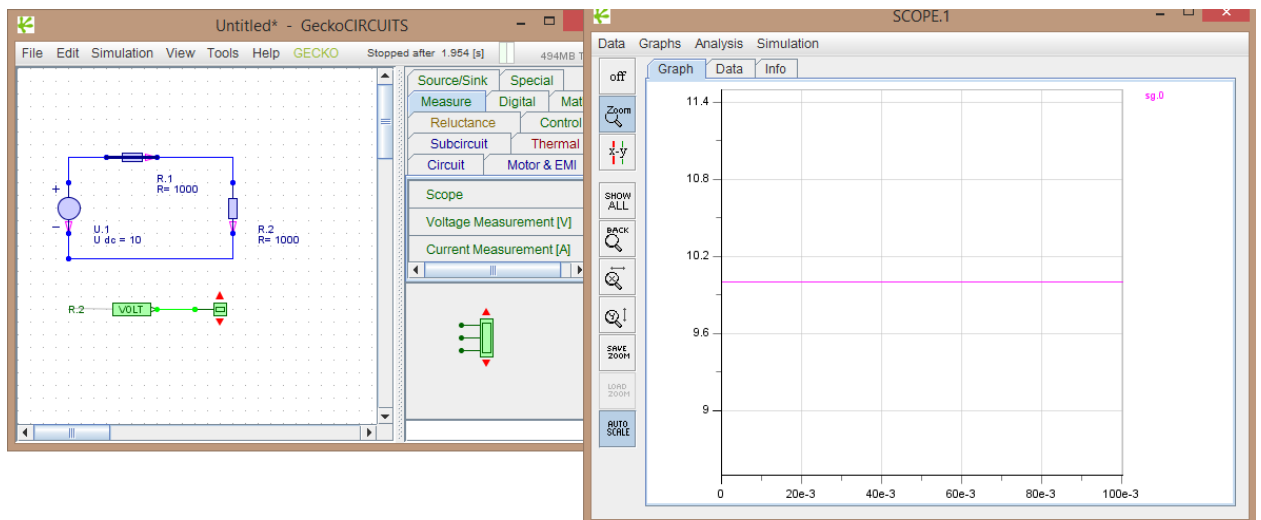


Рис. 4.5. Повторное моделирование

Чтобы снять короткое замыкание с компонента, достаточно повторно выделить компонент и выполнить ту же команду.

Ещё одна команда *Enable/Disable*. Посмотрим на действие команды с такой схемой.

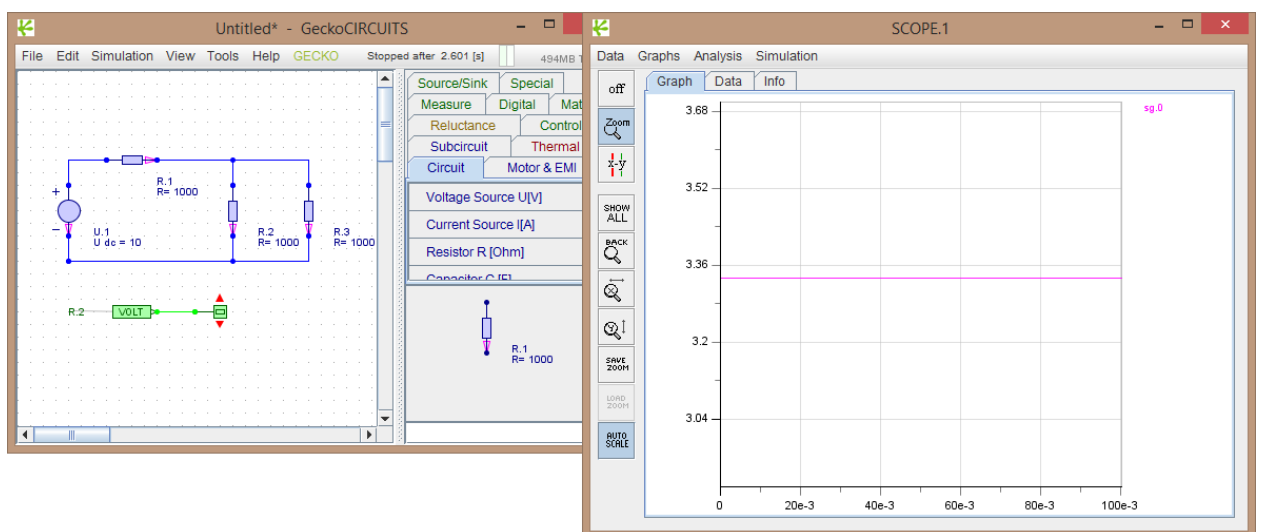


Рис. 4.6. Напряжение на резисторах R_2 и R_3

Выделим резистор R_3 (прямоугольным выделением) и применим команду *Enable/Disable*, щёлкнув по ней левой клавишей мышки в списке команд раздела Edit.

Конвертеры в программе GeckoCIRCUITS

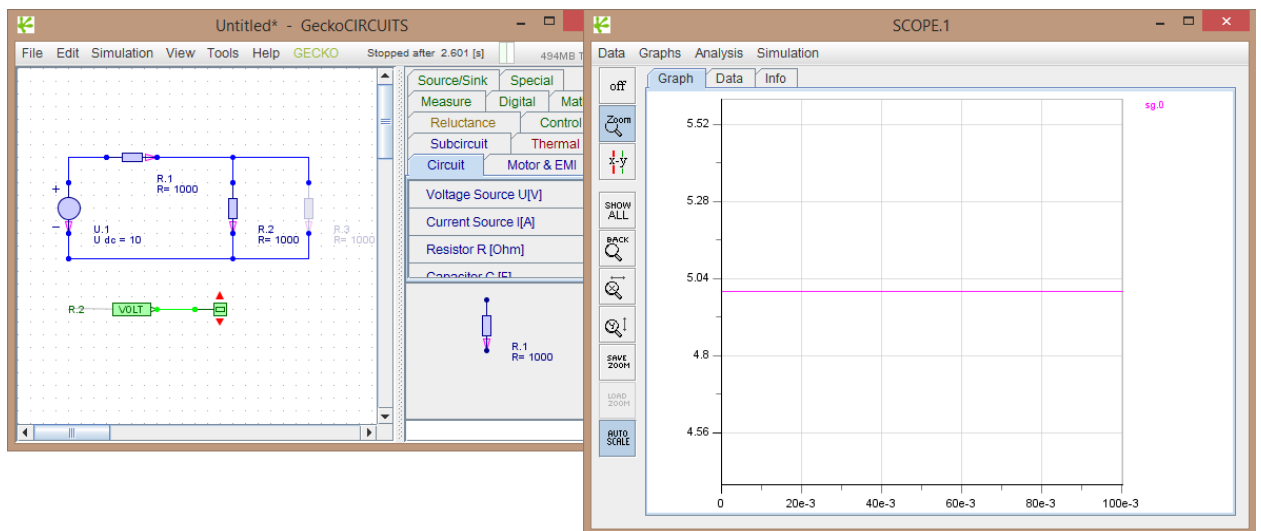


Рис. 4.7. Выполнение моделирования после операции с R3

Для восстановления «существования» резистора R3 достаточно повторить описанные выше операции.

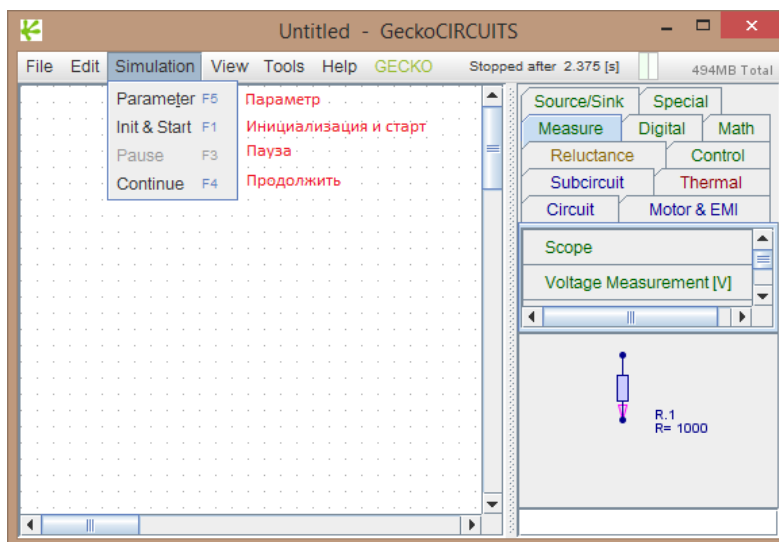
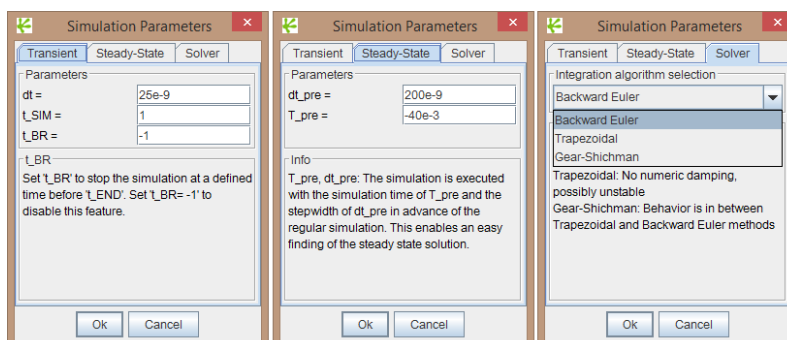


Рис. 4.8. Список команд раздела симуляции

Последние две команды относятся к процессу симуляции, который можно остановить на время и продолжить его выполнение. Первый в списке раздела *Parameter*. Откроем диалоговое окно.



Конвертеры в программе GeckoCIRCUITS

Рис. 4.9. Диалоговое окно настроек параметров симуляции

Все три закладки показаны на рисунке. На первой задаётся шаг по времени и время симуляции. На второй параметры установившегося состояния, а последняя закладка даёт возможность выбрать метод решения уравнений.

Init&Start. Инициализация схемы и запуск моделирования. Касательно тех схем, о которых говорилось выше, вначале следует двойным щелчком по компоненту *Scope* открыть графическое окно, в котором есть собственное меню и свой раздел *Simulation*. Если в нём щёлкнуть мышкой по этой команде, то начинается моделирование.

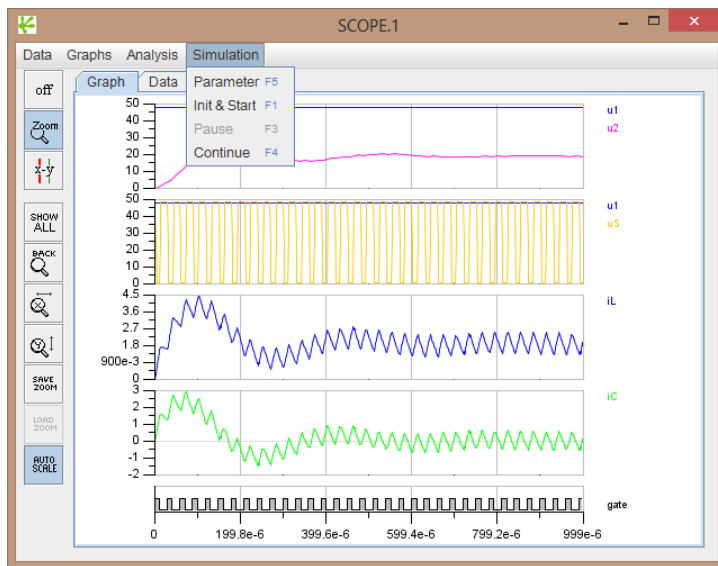
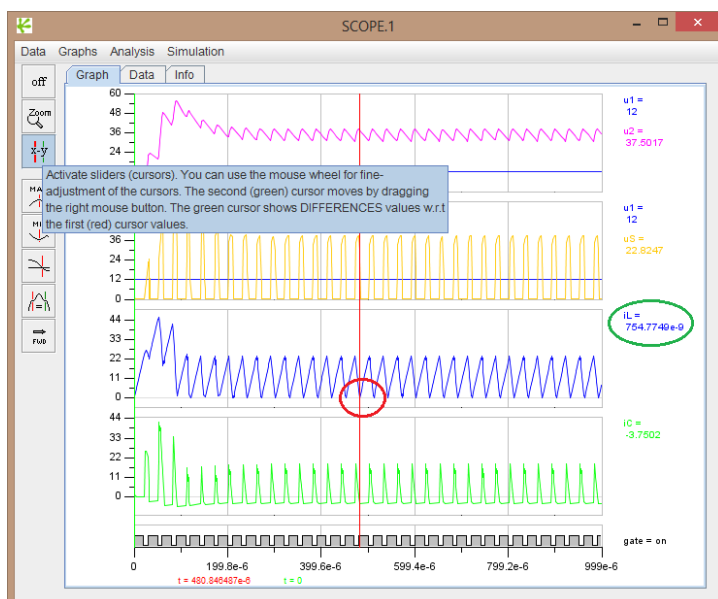


Рис. 4.10. Раздел симуляции в окне графики

И пока мы в графическом окне, напомним, что можно применить маркеры для получения значений. В предыдущем разделе мы упоминали режим работы с прерывистым током. Вспомним, что максимальное и минимальное значение зависят и от величины индуктивности. Уменьшим значение до появления прерывистого тока.



Конвертеры в программе GeckoCIRCUITS

Рис. 4.11. Начало работы преобразователя в режиме с прерывистым током

Маркер появляется с помощью кнопки на левой инструментальной панели, которая выделена на рисунке. Перемещая маркер в нужную точку графика с помощью мышки, вы на правой панели увидите все значения переменных. Колёсико мышки поможет точнее настроить маркер. А поместив курсор мышки поверх кнопки, вы увидите подсказку. Выйти из режима можно с помощью кнопки off, верхней на левой панели. А придать панели исходный вид можно с помощью кнопки **Zoom**.

Вы можете рассчитать значение индуктивности и проверить, совпадёт ли это значение с тем, что получено при симуляции.

Раздел *View*.

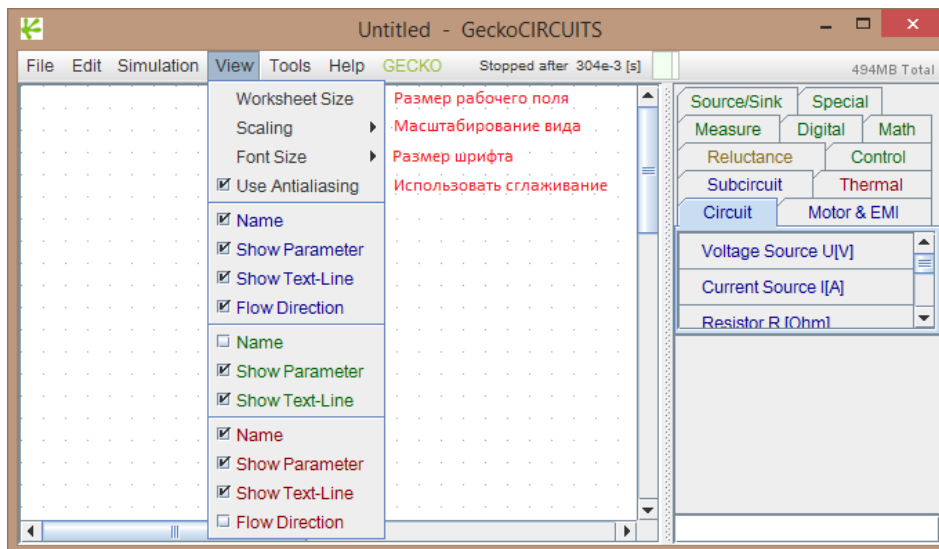


Рис. 4.12. Настройка вида рабочей области

В последних разделах можно задать или убрать, что из обозначений будет видно на чертеже.

Раздел *Tools*.

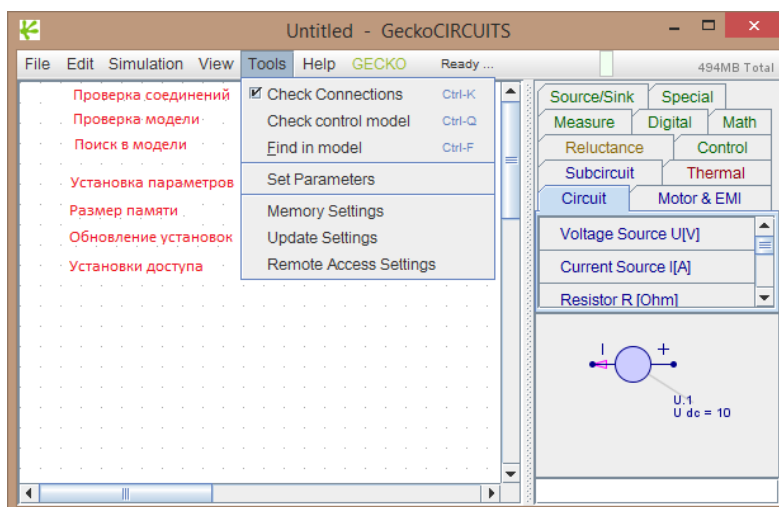


Рис. 4.13. Раздел инструментов

Конвертеры в программе GeckoCIRCUITS

Проверка соединений. После щелчка по этой команде курсор меняет вид. Щёлкните по любому узлу на схеме, чтобы увидеть:

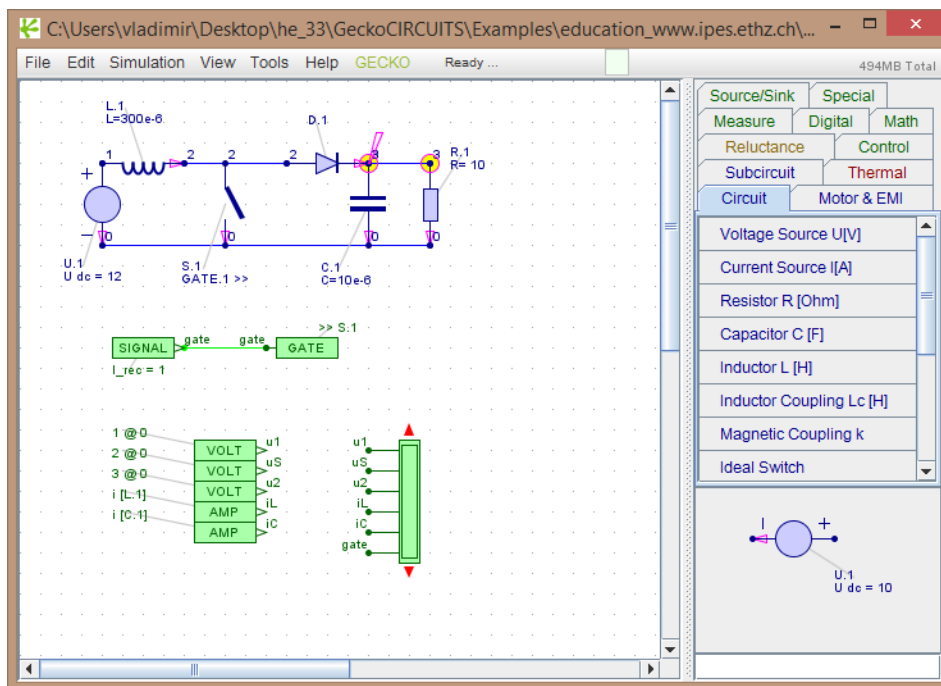


Рис. 4.14. Проверка соединений

Проверка модели даёт информацию о подключениях модели. Добавим в рабочее поле вольтметр. Выделим его прямоугольным выделением. Щёлкнем по команде *Check control model*.

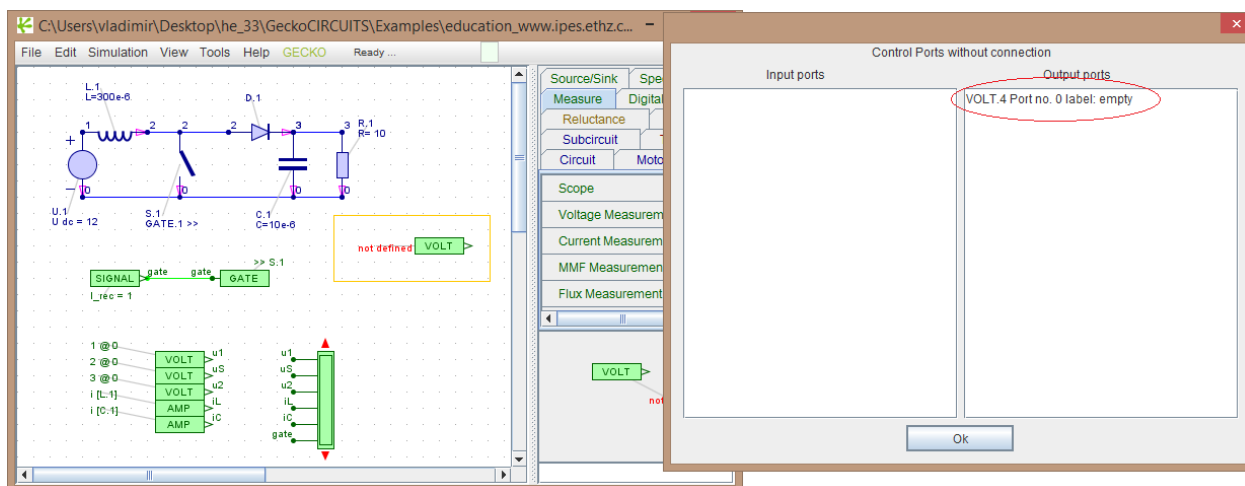


Рис. 4.15. Проверка модели

В появившемся окне справедливо отмечено, что выход вольтметра не подключен.

Поиск в модели сводится к вводу текста в окошке появившегося диалогового окна и наблюдения за схемой, где отмечается результат поиска.

Конвертеры в программе GeckoCIRCUITS

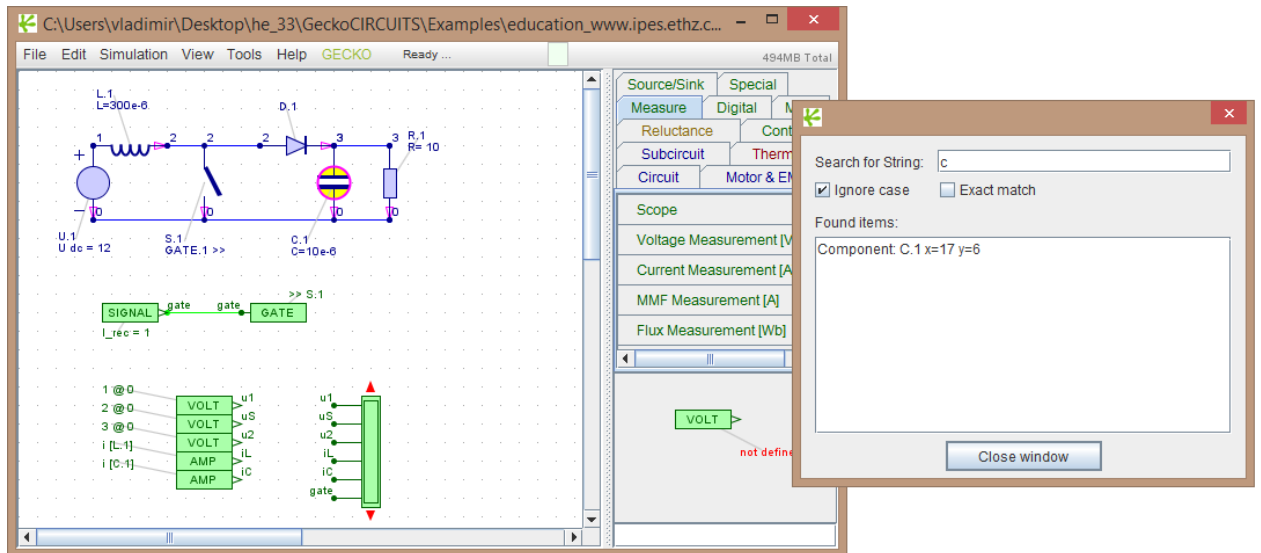


Рис. 4.16. Поиск

Об установке параметров можно прочитать на закладке *Help* окна установки.

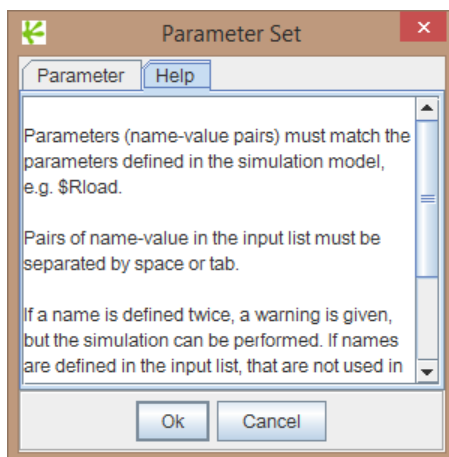


Рис. 4.17. Подсказка в окне установок параметров

Последние три команды вызывают окна настроек, включая настройки удалённого доступа.

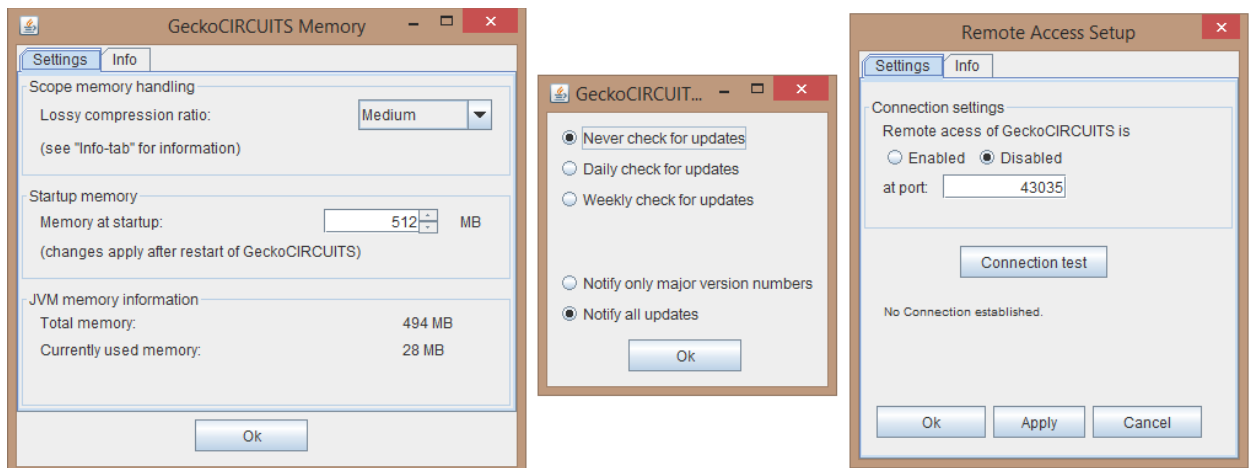


Рис. 4.18. Настройки памяти, обновление установок и настройка доступа

Конвертеры в программе GescoCIRCUITS

Последнее, что хотелось бы упомянуть – управление цветом графиков. Цвет графиков программа задаёт автоматически. Но вы можете изменить цвет, что порой очень удобно. Вернёмся к меню графического окна и выберем в разделе *Graph* пункт *Signal-Graph*.

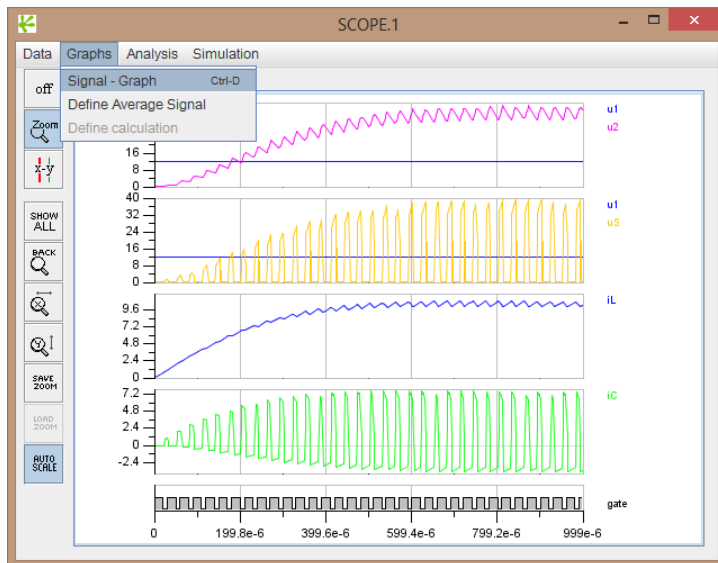


Рис. 4.19. Раздел *Graph* графического окна

В появившемся диалоге изменим жёлтый цвет графика на, скажем, чёрный, чтобы лучше видеть график. Для этого щёлкнем левой клавишей мышки по маркеру графика:

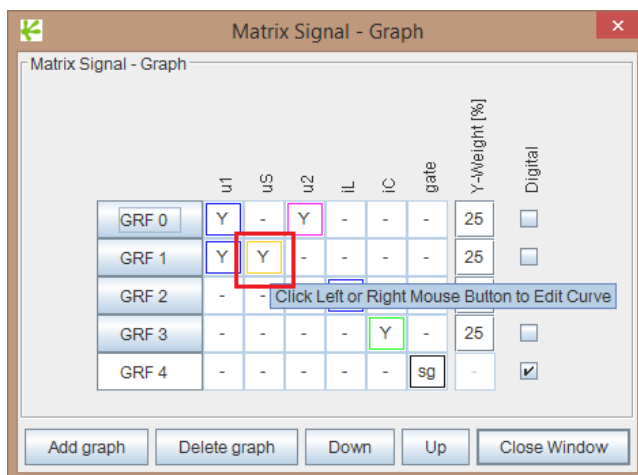


Рис. 4.20. Выбор графика для изменения цвета

В окне диалога настройки вида кривой выберем нужный цвет из выпадающего списка в окошке *Line Color*.

Конвертеры в программе GeckoCIRCUITS

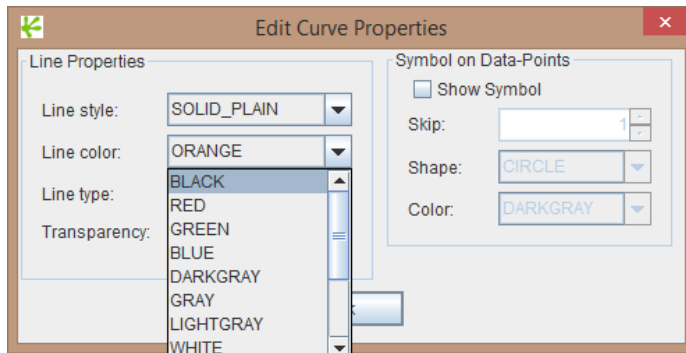


Рис. 4.21. Выбор цвета графика

Здесь же вы можете задать тип линии (*Line type*) или прозрачность (*Transparency*). А вот перечень стилей линии (*Line style*):

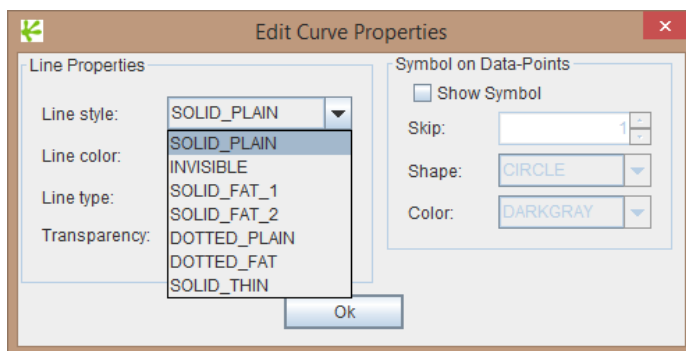


Рис. 4.22. Список стилей линии

Когда графиков много полезно разнообразить их вид, чтобы лучше понимать происходящее.

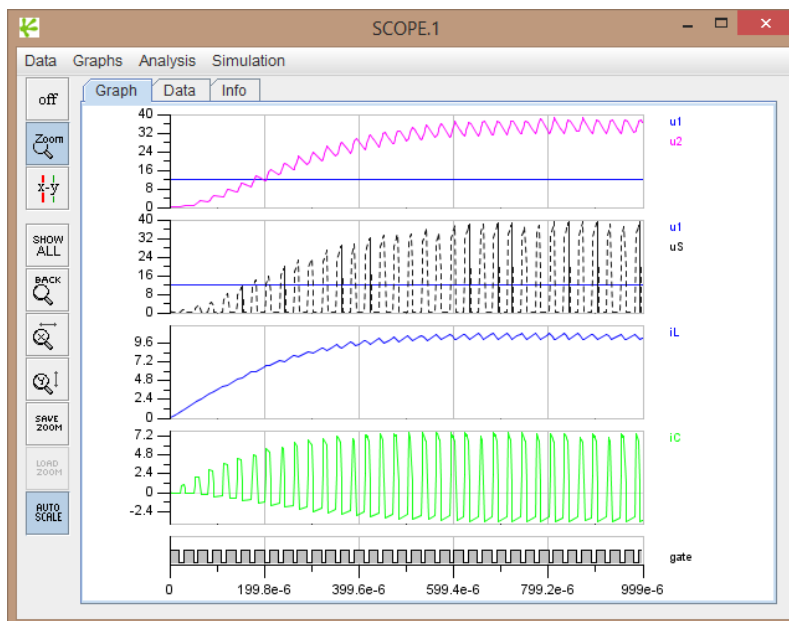


Рис. 4.23. Вид изменившегося графика

Закключение

Далеко не все программы моделирования хорошо работают с преобразователями, а импульсные преобразователи всё чаще используются в разных устройствах. Если начинающим радиолюбителям покажется, что программа не очень удобна в пользовании, то... не всегда же им быть начинающими.

Выше я обещал, что к концу рассказа вернусь к схеме, с которой всё началось. Время пришло.

Во-первых, в схеме обнаружилась опечатка.

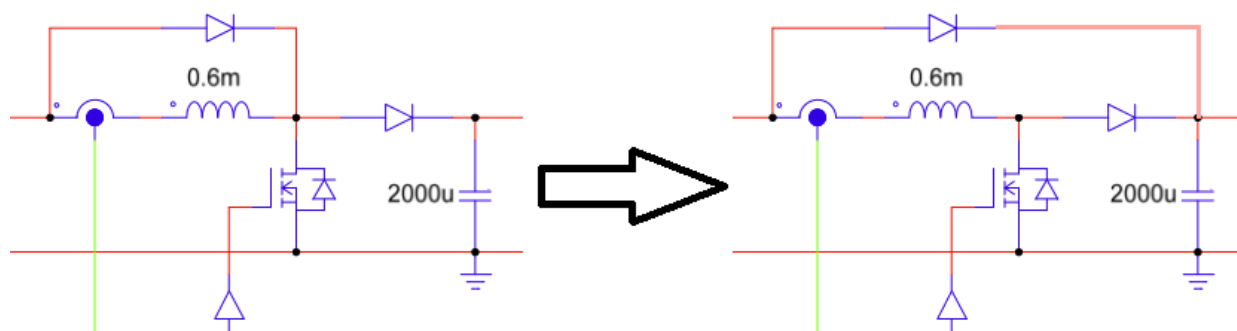


Рис. 5.1. Опечатка в исходной схеме

Опечатка обнаружилась, когда я узнал, что за схема управления применена в этом преобразователе. Меня уведомили, что это «power factor correction или PFC». Кому-то это было ясно с самого начала, а мне пришлось поискать в Интернете пояснений, чтобы понять – следует понаблюдать за напряжением и током, потребляемым схемой от источника питания:
http://www.nix.ru/support/fag/show_articles.php?number=616&faq_topics=PFC

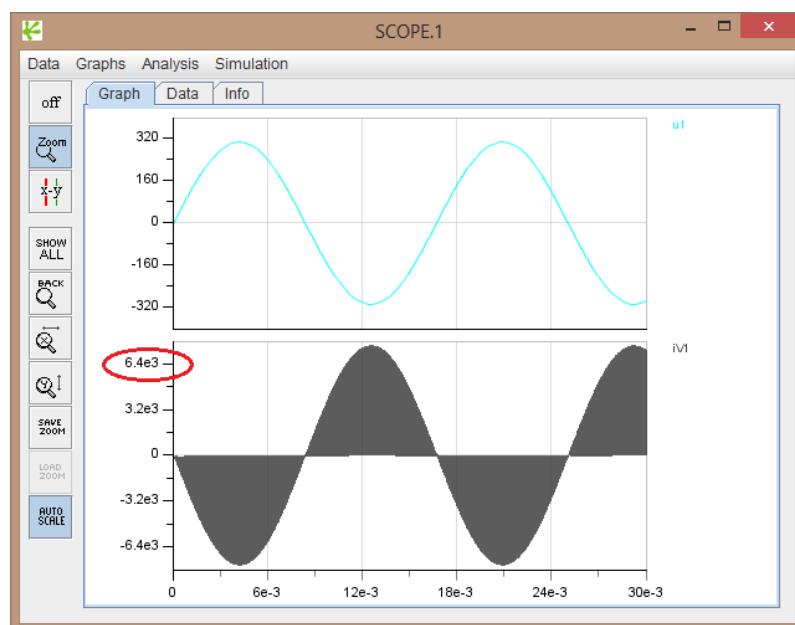


Рис. 5.2. Напряжение и ток источника питания в исходной схеме

Величина тока и привлекла внимание – если бы диоды и транзисторный ключ не были бы идеальными, если бы схема была включена на макетной плате, то они сгорели бы, поскольку ключевой транзистор закорачивает выпрямитель. Убрав диод, мы получим следующие осциллограммы.

Конвертеры в программе GeckoCIRCUITS

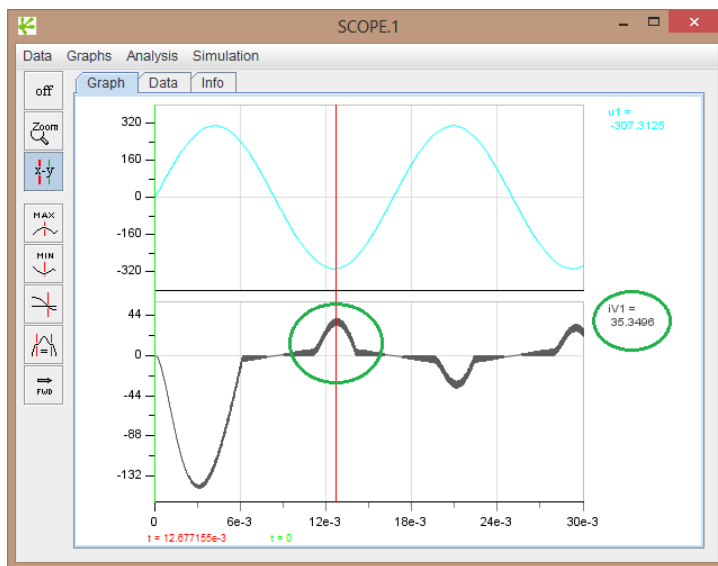


Рис. 5.3. Напряжение и ток источника без диода, вызвавшего сомнения

Характер изменения тока исправленной схемы показывает смысл PFC. Ток, потребляемый схемой от источника питания, характерен для реактивной нагрузки, что плохо для сетевого питания. Все нагрузки стараются привести к активному характеру.

Осциллограммы, приведённые выше, сделаны для схемы без управляющей части. Со схемой управления осциллограмма выглядит так:

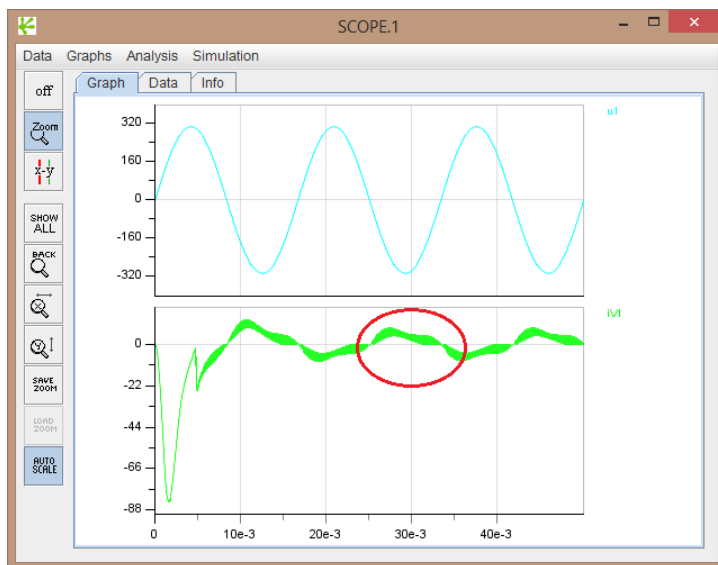


Рис. 5.4. Осциллограмма напряжения и тока со схемой управления

Потребляемый ток выглядит лучше, но не наилучшим образом. Сомнения у меня вызывает датчик тока, который я «собирал» по наитию. Попробуем увеличить сопротивление резистора датчика тока в 10 раз.

Конвертеры в программе GeckoCIRCUITS

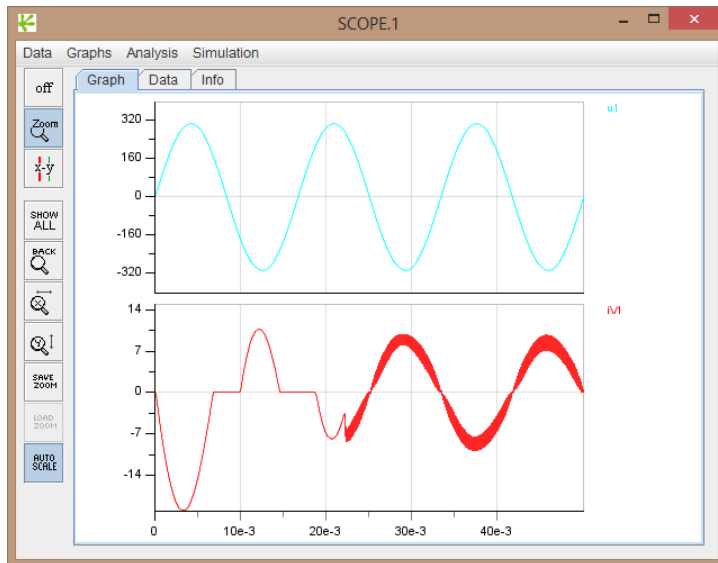


Рис. 5.5. Осциллограммы при изменённом датчике тока

Характер тока в установившемся режиме вполне удовлетворяет нашим планам, но величина резистора, наверное, слишком велика. Можно уменьшить сопротивление, а вольтметр датчика тока снабдить усилителем.

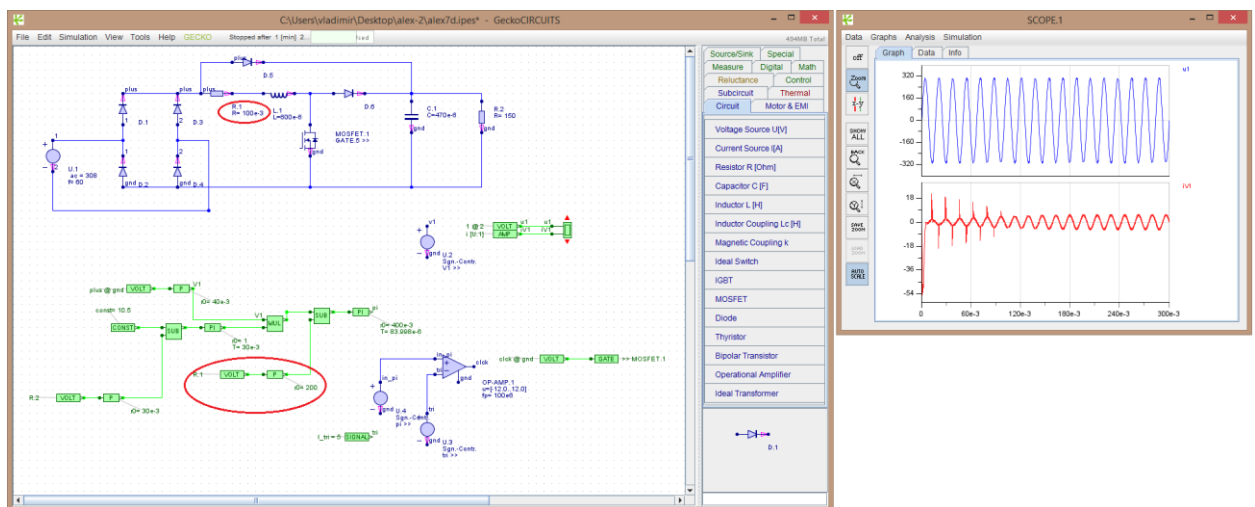


Рис. 5.6. Модификация схемы и осциллограмма входного напряжения и потребляемого тока

А завершить рассказ мне хотелось бы повторением симуляции схемы после тех исправлений, что были сделаны.

Я не берусь судить о правильности работы схемы, поскольку знаю о ней слишком мало. Я собирался проверить только возможность моделирования схемы в программе GeckoCIRCUIT. Если вас интересуют конвертеры, то найдите в примерах простейшую схему, рассчитайте её элементы, проверьте её работу в программе моделирования, соберите схему на макетной плате и сравните результаты. Во всех смыслах это будет полезно. Особенно в части доверия к вашей программе моделирования.

Конвертеры в программе GeckoCIRCUITS

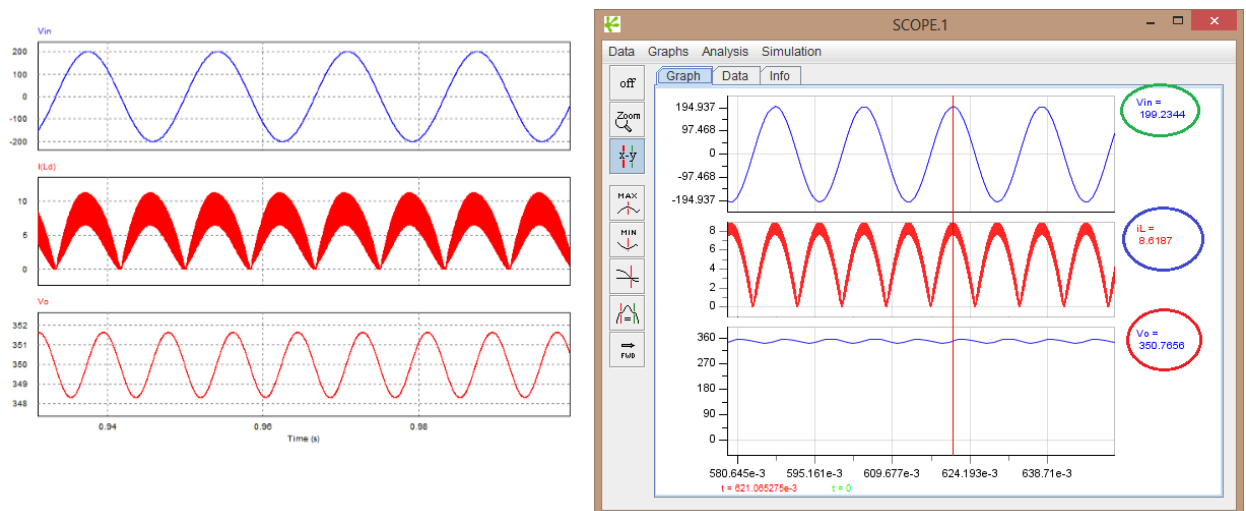


Рис. 5.7. Осциллограммы входного и выходного напряжения и тока через индуктивность L1

Согласитесь, что результаты оригинальные и полученные в GeckoCIRCUIT очень схожи.

Многие начинающие радиолюбители стараются найти подходящую схему, сделать печатную плату и спаять устройство. Но всегда могут быть досадные опечатки, как в чертеже, так и в номиналах деталей. В лучшем случае схема не будет работать, в худшем детали сгорят, оставив вас в разочаровании и недоумении. Поэтому я считаю, что даже при использовании готовой схемы следует разобраться в ней, понять принцип её работы, понять назначение каждой из деталей. Эту работу можно проделать и на макетной плате, но сегодня есть такая удобная макетная плата, как программа моделирования. Виртуальные детали не горят, а ошибки выявляются при тщательной проверке. Программы – это инструменты для овладения предметом. И как любые инструменты, программы бывают разными. Одни удобны для работы с аналоговой техникой, другие для работы с импульсными устройствами. Если у вас есть набор отвёрток, то почему не быть набору программ?