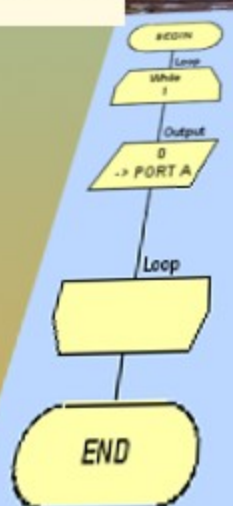
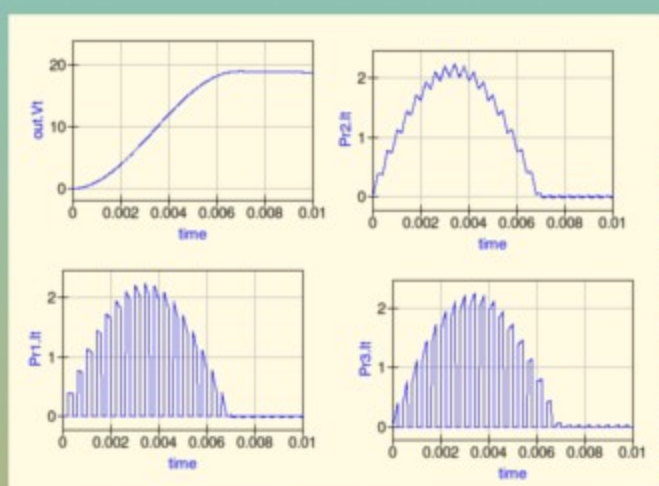


В.Н. Гололобов

Как пройти к электронике?



Москва 2008

Оглавление

Предисловие.....	3
Ток и напряжение.....	4
Переменные напряжения или сигналы в электрических цепях.....	17
Небольшой рассказ об усилителе и его характеристиках.....	28
Немного о генераторах.....	42
Немного о цифровой технике.....	52
А что такое микроконтроллер?.....	59
Мультивибратор при первом знакомстве с микроконтроллером.....	70
Устройства питания и преобразователи напряжения.....	89
О программе Qucs и других программах.....	96
Заключение.....	106

Предисловие

Как пройти к «Электронике»?

Этот вопрос озадачил меня много лет назад, когда я наконец собрался посмотреть недавно открывшийся в Москве новый магазин. Я шел по Ленинскому проспекту от старого магазина к новому, думая уж не помню о чем, и первым желанием было махнуть рукой вперед, но сомнения удержали меня от этого «жеста доброй воли», я сообразил, что нужно спросить, а какой из двух магазинов нужен? Но, пока я предавался сомнениям и размышлениям, кто-то из прохожих взялся объяснять, а я отправился дальше.

Сегодня, изредка общаясь с начинающими любителями, я пытаюсь понять, как им лучше пройти к электронике? Бывалые радиолюбители не сомневаются, что после чтения их любимых книг, взяв паяльник в руки, начинающий любитель готов познать все радости этого действительно увлекательнейшего хобби.

Я и согласен с ними, и не согласен. Даже специалист, читая порою весьма сбивчивое описание, не способен в полной мере оценить все преимущества и недостатки схемы, хотя опыт помогает ему разобраться в пригодности схемы к его целям. А что говорить о начинающих, для которых цель далеко не всегда ясна? Мне кажется, что современными технологиями, компьютерным моделированием, как удачным инструментом, дополняющим макетную плату и приборы, не следует пренебрегать.

На вопрос, как пройти к электронике, можно сказать, что есть много путей, и один из них — попытаться вспомнить, что рассказывал преподаватель на уроках физики в школе об электричестве.

Или, давайте вместе вспомним это.

Ток и напряжение

Пожалуй, это все, что нам понадобится вспомнить из школьного курса физики. Во всяком случае, пока. Если вас интересует теория, или если вы почувствуете дискомфорт от недостатка точных определений и описаний процессов, протекающих в электрических цепях, то можно углубиться в чтение учебников по ТОЭ (теоретические основы электротехники), физике (например, физике твердого тела или физике полупроводников). А если вы решите выбрать электронику в качестве сферы деятельности, то вы пойдете в учебное заведение по этому профилю и все предметы изучите (надеюсь, что так) досконально.

Электрический ток, протекающий по компонентам любой электрической схемы, суть всего, что связано с электроникой, электрикой или радиотехникой. В любом описании схемы вы встретите упоминание о токе в несколько миллиампер или ампер. Вы можете измерить ток в электрической цепи с помощью мультиметра. В действительности, когда мы привычно рассуждаем о токе, мы имеем в виду силу тока. Вот определение, которое я выписал, чтобы поговорить о токе.

Сила тока в проводнике - скалярная величина, численно равная заряду, протекающему в единицу времени через сечение проводника: $I = \delta q / \delta t$.

О чем говорит это определение? В первую очередь о том, что по проводнику «протекает» заряд или, другими словами, по проводнику движутся заряженные частицы. Если проводник — металл, то это, скорее всего, электроны. Если проводник — электролит, то это, вероятно, ионы. И т.д. Действительно, электрический ток — это направленное движение электрических зарядов. Направленное, чтобы отличать его от хаотического и беспорядочного, не образующего электрического тока, но которое есть в металлах, прекрасных проводниках электрического тока. И можно догадаться, что есть нечто, направляющее движение зарядов. Называется это нечто источником тока или источником электродвижущей силы (ЭДС). Из физики мы знаем, что заряд, помещенный в электростатическое поле, будет двигаться под действием этого поля. А это позволяет предположить, что источник ЭДС создает в проводнике поле, которое и заставит хаотически движущиеся электроны двигаться упорядоченно, направленно. Математическое описание полей тесно связано с понятием потенциала, которое для электростатического поля определяется (я выпишу это определение) так:

Потенциал электростатического поля — скалярная величина, равная отношению потенциальной энергии заряда в поле к этому заряду: $\varphi = W/q = \text{const}$, — энергетическая характеристика поля в данной точке. Потенциал не зависит от величины заряда, помещенного в это поле.

Так как потенциальная энергия зависит от выбора системы координат, то и потенциал определяется с точностью до постоянной. За точку отсчета потенциала выбирают в зависимости от задачи: потенциал Земли, потенциал бесконечно удаленной точки поля или потенциал отрицательной пластины конденсатора.

Потенциал численно равен работе поля по перемещению единичного положительного заряда из данной точки электрического поля в бесконечность. В системе единиц СИ потенциал измеряется в вольтах: 1 Вольт = 1 Джоуль/1 Кулон.

На практике чаще имеют дело с величиной, называемой напряжением:

Напряжение — разность значений потенциала в начальной и конечной точках траектории. К понятию разности потенциалов приводит рассмотрение работы:

$$A = - (W_2 - W_1) = - (\varphi_2 - \varphi_1)q = - q\Delta\varphi$$

$$U = \varphi_1 - \varphi_2 = - \Delta\varphi = A/q$$

Напряжение численно равно работе электростатического поля при перемещении единичного положительного заряда вдоль силовых линий этого поля. Разность потенциалов (напряжение) не зависит от выбора системы координат.

Я думаю, этих определений пока будет достаточно.

Вернемся к электрическому току. Для того, чтобы по электрической цепи протекал ток, нам нужен источник ЭДС, он же источник питания. Это может быть химический источник тока (батарейка, аккумулятор), это может быть генератор (на электростанции), это может быть фотоэлемент (солнечная батарея) и т.д.

Небольшое отступление: я для иллюстрации сказанного буду использовать программу Qucs. Установленная на моем компьютере операционная система Linux прекрасно работает с этой бесплатной программой. Но вы можете использовать кроме нее (есть версия для Windows) такие прекрасные программы как Multisim, Micro-Cap или Proteus (они, кстати, вполне удачно могут работать в Linux), или любую другую, которая вам кажется интереснее или доступнее.

Так что же с электрическим током?

Не будет большой беды, если в начале освоения электроники мы будем под источником тока подразумевать батарейку, как в фонарике, пульте управления телевизором или плеере. Величина ЭДС измеряется в вольтах, а батарейки в силу ряда причин вы чаще встретите с ЭДС в 1.5, 4.5 или 9 В, чем с другими значениями. Так что, если в примерах мы используем батарейку с напряжением в 10 В, то это только для удобства рассуждений, хотя получить требуемую величину источника питания не составит особого труда от сетевого блока питания, но о нем мы поговорим позже. На электрических схемах батарейка обозначается в виде двух параллельных линий, одна из которых короче и толще, а другая длиннее и тоньше. Чтобы подчеркнуть большее напряжение, чем 1.5 В, таких линий может быть несколько. Вообще, электрическая схема — это записанное графическим языком описание устройства. Каждый элемент схемы имеет свое назначение и изображается рисунком, принятым для его обозначения. По мере рассказа станет ясно, какие элементы электрических схем применяются на практике, как они обозначаются на схеме и для чего служат.

Самую простую электрическую схему (или электрическую цепь) мы получим, если присоединим к батарейке проводник (провод). В такой цепи под действием ЭДС будет протекать постоянный ток. Проводник удобно характеризовать величиной, которая называется сопротивлением (сопротивление протеканию электрического тока). Эта величина для проводника определяется его длиной, толщиной и материалом, из которого сделан проводник. Свойства материала характеризуется удельным сопротивлением, а толщина площадью сечения. Таким образом, сопротивление проводника можно записать в виде:

$$R = \rho(l/S), \text{ где } \rho — \text{удельное сопротивление, } S — \text{площадь сечения, } l — \text{длина.}$$

При протекании тока по проводнику, имеющему сопротивление, на его концах есть разность потенциалов, которую называют падением напряжения (или напряжением). Между током, напряжением и сопротивлением существует простая взаимосвязь, открытая Омом и названная в его честь «Закон Ома»:

$$I = U/R, \text{ где } I — \text{ток, } U — \text{напряжение, } R — \text{сопротивление.}$$

Для простейшей схемы, о которой мы говорили, справедливо и очевидно, что ЭДС источника тока и падение напряжения на проводнике равны. Если обобщить это

утверждение, то можно получить еще один из законов, открытых Кирхгофом:

Алгебраическая сумма напряжений в замкнутом контуре равна алгебраической сумме ЭДС.

Есть еще один закон Кирхгофа, показывающий, что ток не исчезает бесследно, если электрическая цепь разветвляется:

Ток, входящий в точку ветвления электрической цепи, равен сумме токов в ветвях.

И, конечно, это же утверждение справедливо для точки соединения ветвей, где втекающий ток будет равен сумме токов в ветвях.

Принято в этой части рассказа о токах и напряжениях использовать простые (или сложные) схемы с резисторами и источниками питания. Но при этом возникает соблазн игнорировать простые соотношения, считая, что в недрах современных «чипов» все описывается другими законами. Чтобы показать, что это не так, я хочу иллюстрировать все вышесказанное на примере типового простейшего усилительного каскада на транзисторе.

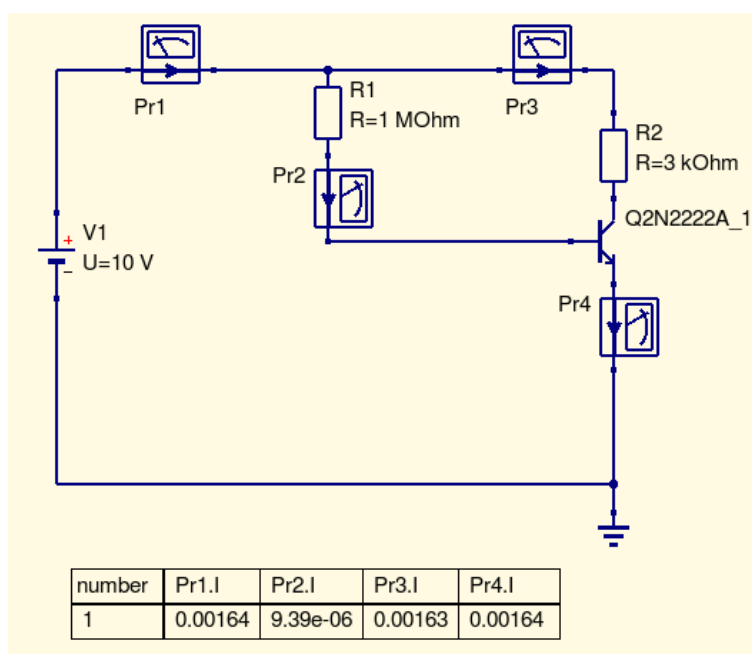


Рис. 1.1. Измерения по постоянному току в транзисторном каскаде

Некоторые пояснения. За техническое направление тока принято считать протекание тока от плюса источника к минусу. В данном случае источник тока (ЭДС) V1 с напряжением 10 В. От его плюса ток течет к прибору (амперметру) Pr1, после которого разветвляется. Часть тока протекает через резистор R1 и прибор Pr2, а часть через Pr3 и резистор R2. В транзисторе Q2N222A эти токи соединяются, чтобы «вытечь» из эмиттера транзистора и через Pr4 отправиться к минусу источника тока. Резисторы R1 и R2 — это компоненты электрической схемы, которые в первом приближении можно рассматривать, как проводники, обладающие сопротивлением в 1 мегаом и 3 килоома, согласно схеме. Конечно проводники с таким сопротивлением получить из обычных электрических проводов сложно. Поэтому резисторы с разным сопротивлением изготавливают специальным образом. Но сути это не меняет. Транзистор. Транзистор — полупроводниковый прибор, входящий в состав очень многих и аналоговых, и цифровых микросхем. В данном случае нарисован биполярный транзистор (есть еще полевые транзисторы), имеющий три вывода: сверху — коллектор, снизу — эмиттер, сбоку — база.

Ток, показываемый амперметром Pr1, равен 1.64 мА (миллиампер, одна тысячная ампера) и делится на базовый Pr2 и коллекторный Pr3 токи. Эти токи соответственно равны 0.01 мА и 1.63 мА. Именно об этом говорит закон Кирхгофа: $1.64 = 1.63 + 0.01$. И соединяясь в транзисторе эти токи в ветвях цепи образуют: $1.63 + 0.01 = 1.64$ (мА), что и показывает амперметр Pr4.

И еще один опыт с этой же схемой:

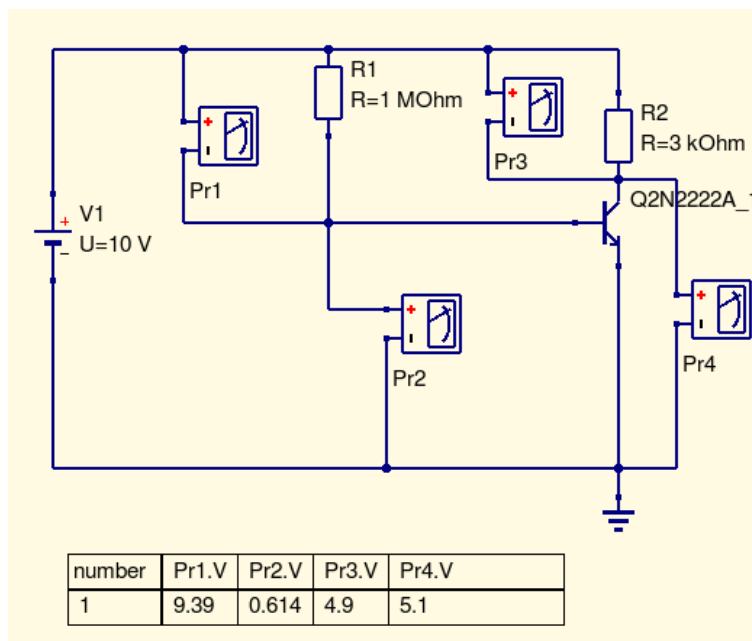


Рис. 1.2. Измерение напряжений в транзисторном каскаде

Вместо амперметров измерения мы проводим с помощью вольтметров — приборов для измерения напряжения, в данном случае постоянного напряжения. Другой закон Кирхгофа гласит, что сумма падений напряжений равна ЭДС. Проверим: $9.39 + 0.61 = 10$ (показания Pr1 и Pr2), $4.9 + 5.1 = 10$ (показания вольтметров Pr3 и Pr4).

А теперь, забегаю вперед, я расскажу, как бы я рассчитывал (навскидку) такой усилительный каскад по постоянному току, то есть, выбирал бы работу каскада по постоянному току. Обычно, и довольно часто, режим работы выбирается таким, чтобы на коллекторе транзистора в статическом режиме (без сигнала) было напряжение равное половине питающего напряжения. Резистор в цепи коллектора (его называют еще сопротивлением нагрузки каскада) выбирается из разных соображений, я выбрал его величину 3 кОм. Половина питающего напряжения это 5 В. Вспомнив закон Ома, я могу вычислить нужный мне ток коллектора: $5 : 3000 = 0.00166$ (А). Биполярный транзистор имеет такое свойство, что его ток коллектора равен току базы, умноженному на некоторую постоянную, которую называют статическим коэффициентом усиления (по току в схеме с общим эмиттером). Для данного транзистора типовое значение этого параметра равно 200. Таким образом, ток базы будет в 200 раз меньше, чем ток коллектора: $1.66 \text{ (мА)} : 200 = 0.0083$. Еще одно свойство биполярного транзистора — напряжение база-эмиттер невелико, и, как правило, не больше 1 В. Считая, что все напряжение ЭДС приложено к резистору R1, а ток через него (это ток базы транзистора) равен 0.0083 мА, я могу определить величину этого резистора: $10 \text{ (В)} : 0.0083 \text{ (мА)} = 1.2 \text{ МОм}$.

Даже такой очень приближенный расчет, использующий всего три закона электротехники позволяет определить значения близкие к реальным. Более точные расчеты при

проектировании транзисторных усилителей используют эти же законы, но учитывают и возможный разброс значений, и влияние температуры и т.д.

Некоторое недоверие может вызывать численное расхождение в показаниях приборов на рисунке 1.1. Это, скорее всего, результат округлений при численных расчетах. Но на практике следует иметь ввиду, что любой прибор имеет точность измерений, обусловленную при производстве. И показания прибора могут быть ошибочны в пределах этой точности. Например, если прибор измеряет с точностью $\pm 10\%$, то при измерении 10 В вы можете, и это правильно, увидеть значения в диапазоне от 9 до 11 вольт. Это обязательно нужно помнить, чтобы не ломать голову, пытаясь понять, куда что подевалось?

При измерениях на постоянном токе в простых случаях можно легко определить значения токов и напряжений, используя три закона электротехники. Но в более сложных случаях следует прибегать к специальным методам расчетов, иначе результат совсем не очевиден. Вот совсем несложная электрическая цепь:

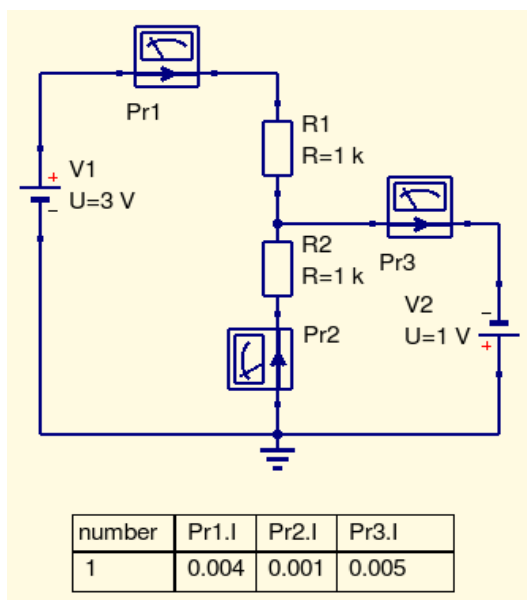


Рис. 1.3. Измерения в цепи с двумя источниками ЭДС

Если вам не составляет труда определить токи, если вас не смущают показания амперметров Pr1, Pr2 и Pr3, представленные в таблице, то вы счастливчик. Иначе можно применить, например, метод контурных токов. Вот, что написано об этом методе в курсе теоретических основ электротехники у Бессонова (когда-то я учился по этому учебнику и мне удобно пользоваться им):

При расчете методом контурных токов полагают, что в каждом независимом контуре схемы течет свой контурный ток. Уравнения составляют относительно контурных токов, после чего определяют токи ветвей через контурные токи...

Для нарисованной выше схемы можно выделить два контура: V1R1R2 и V2R2. Обозначив контурные токи I_{k1} и I_{k2} , можно написать два уравнения для этих двух неизвестных.

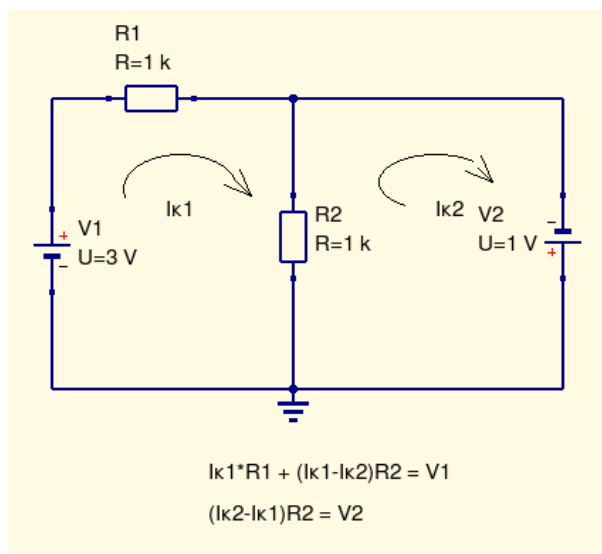


Рис. 1.4. Контурные токи и уравнения для них

Решив систему из двух уравнений с двумя неизвестными, можно убедиться, что токи соответствуют полученным с помощью программы Qucs. Этого и следовало, впрочем, ожидать. В практике мне редко приходилось решать подобные задачи, но иногда, производя измерения, сталкиваешься с результатами, которые заставляют сомневаться, и чтобы развеять сомнения можно рассчитать, а еще лучше проверить их с помощью компьютера в программе Qucs или аналогичной.

До сих пор речь шла о постоянном токе. Чем переменный ток отличается от постоянного?

Конечно, в первую очередь тем, что он меняется. Он меняется либо по величине, либо по направлению, либо и по величине, и по направлению. На практике чаще всего приходится иметь дело с периодически меняющимся током (или напряжением), что означает, что есть некоторый период времени, когда ток (или напряжение) меняется по какому-то закону, а затем закон изменения тока повторяется в промежутки времени равные этому периоду.

Один из самых простых законов изменения тока — синусоидальный. Из математики мы знаем, что функция синуса периодическая, то есть, когда мы говорим о синусоидальном переменном токе, то подразумеваем его периодичность. Есть и еще одна причина, отчего при испытаниях электрических схем очень охотно используют генераторы синусоидального напряжения. Дело в том, что многие периодические переменные токи, описываемые очень сложными функциями, можно представить в виде совокупности синусоидальных токов. Если быть точным, то совокупности синусоидальных и косинусоидальных токов с частотами кратными основной частоте, и амплитудами, определяемыми видом исходного переменного сигнала. Такое представление сложной функции с помощью функций синуса и косинуса называют разложением функции в гармонический ряд Фурье. Позже можно провести некоторые опыты в программе Qucs, чтобы посмотреть, как это выглядит (или может выглядеть) на экране осциллографа. Однако вернемся к переменному току.

Когда в школьные годы я пытался разобраться в работе схем, то меня часто смущало одно обстоятельство. Достаточно стандартное построение усилителей (ламповых или транзисторных) — это связь между каскадами с помощью конденсатора. Конденсатор, как я уже знал, не пропускает постоянный ток. Действительно, любой конденсатор можно представить в конечном счете, как две проводящие пластины, между которыми есть слой изолятора. Изолятор не позволяет постоянному току протекать через конденсатор. Но, и это я

тоже знал, переменный ток протекает через конденсатор. И смущало меня то, что в схемах усилителей не было источников питания переменного тока, только постоянного. Причиной моей ошибки, вернее, моих затруднений, было мое неверное представление о переменном токе, как токе обязательно меняющем направление. Достаточно, чтобы ток периодически изменялся по величине, чтобы он стал переменным. А разделительный конденсатор, конденсатор, соединяющий два каскада усиления, реагирует именно на изменения тока (или напряжения), пропуская изменяющийся по величине ток, как если бы он менялся и по направлению.

Самый простой способ получить переменный ток — это подключить батарейку к резистору через выключатель. Если в равные промежутки времени включать и выключать ток, то через резистор будет протекать переменный ток.

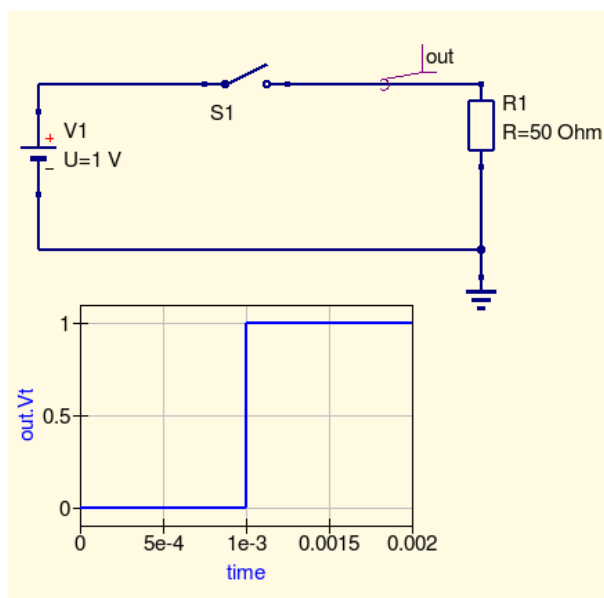


Рис. 1.5. Получение переменного тока из постоянного

На рисунке изображена схема опыта по получению переменного тока с помощью батарейки. Ниже под схемой график напряжения на резисторе, из которого видно, что в течение первой 1 мС (миллисекунда) выключатель S1 разомкнут, и напряжение на резисторе равно нулю, а в течение следующей миллисекунды ключ замкнут, напряжение на резисторе (а значит, вспомним закон Ома, и ток через него...) равно напряжению источника питания. Повторяя включение и выключение в равные промежутки времени мы получим простейший вид переменного тока (простейший только в плане получения его с помощью батарейки). А если вам такой опыт покажется слишком «детским», то обратитесь к современным блокам питания, где происходит (хотя и несколько сложнее) то же самое.

Программа Qucs, как и схожие с ней, позволяет наблюдать, что происходит в цепи, изображенной на рисунке 1.5, если мы разорвем цепь с помощью конденсатора. Включим в нашу экспериментальную цепь конденсатор после выключателя S1 и повторим опыт. Сразу оговорюсь, что резистор R2 сопротивлением в 1000 МОм необходим только для нормальной работы симулятора, хотя далеко не факт, что реальное сопротивление изолятора в конденсаторах не имеет такой же величины. Только идеальный конденсатор имеет диэлектрик с бесконечно большим сопротивлением, изоляторы между обкладками реальных конденсаторов имеют очень большое, но не бесконечно большое сопротивление. Однако в данном случае резистор R2 атрибут не столько приближения к реальному устройству конденсатора, сколько элемент, помогающий симулятору при вычислениях. Все элементы

схемы выбраны достаточно случайно, но вы можете повторить этот опыт с другими элементами, чтобы сравнить результаты. И еще, метка «наблюдения», обозначенная, как out, соответствует точке, к которой подключается щуп осциллографа для наблюдения за напряжением.

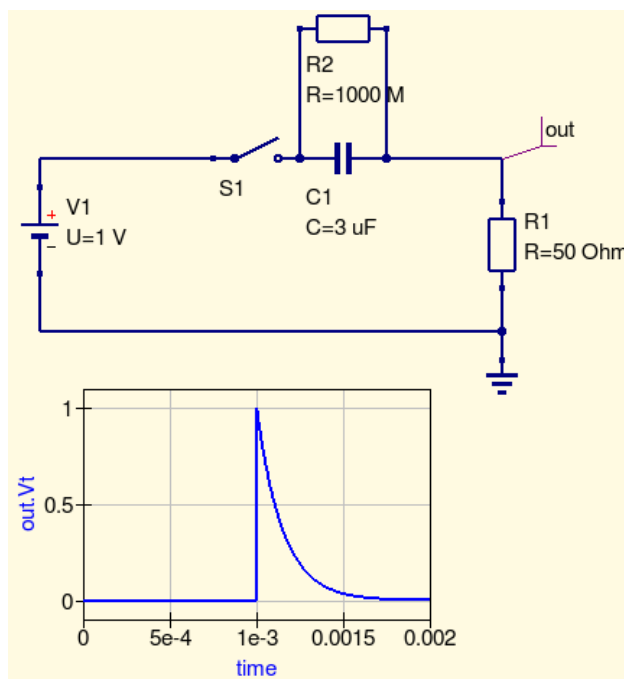


Рис. 1.6. Конденсатор в цепи постоянного тока при подключении источника

На графике отображается напряжение на резисторе R1, которое (по закону Ома) соответствует току через этот резистор, который, в свою очередь, равен току через конденсатор C1. Из графика видно, что сразу после замыкания выключателя S1 ток через конденсатор протекает максимальный, определяемый величиной ЭДС источника V1 (1 вольт) и величиной сопротивления R1 (50 Ом), а затем ток уменьшается и к концу 2 мс почти равен нулю. Подобную реакцию конденсатора на изменение напряжения можно называть переходным процессом или процессом заряда конденсатора. Действительно, если после включения ключа S1 мы не будем его размыкать, то конденсатор зарядится через резистор R1 до напряжения источника V1, в чем можно убедиться на практике, подключив конденсатор к батарейке и измерив напряжение на нем после этого вольтметром.

Но, может быть, конденсатор пропускает только очень резкие изменения напряжения — в нашем опыте напряжение в первую миллисекунду было равно нулю, а затем сразу стало равно напряжению источника питания, а более медленные изменения напряжения конденсатор не пропустит?

Перед тем, как провести следующий эксперимент, я хочу несколько изменить схему: идея использовать батарейку и выключатель для получения прямоугольных импульсов (именно так можно назвать то, что мы получили), может быть, и не плоха, но обычно для этой цели используют генератор прямоугольных сигналов (или источник напряжения прямоугольной формы). Поэтому выбросим из схемы выключатель S1 и батарейку V1, которую заменим генератором. Заодно я изменю величину сопротивления резистора R1 и уберу резистор R2, который симулятору программы Qucs теперь не нужен.

На рисунке ниже верхний график отображает сигнал (напряжение), получаемый от

генератора (метка этого сигнала input), а нижний график — напряжение на резисторе R1. Я оставил время наблюдения равным 2 мС, как в предыдущих опытах, но это время можно выбрать любым, поскольку генератор производит импульсы постоянно, пока включен. Лишь бы выбранное время наблюдения было удобным для этого наблюдения. При слишком большом времени наблюдения сигнал от генератора превратится в невыразительную частую сетку или сплошную цветную полосу. Таким же станет и наблюдаемое на резисторе R1 напряжение.

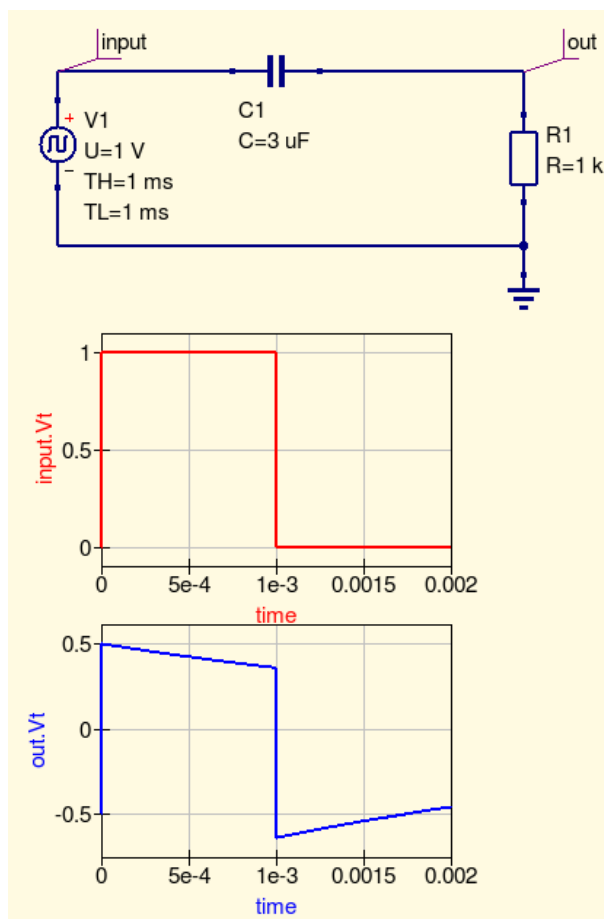


Рис. 1.7. Прохождение прямоугольного напряжения через конденсатор

Кстати, обратите внимание, что напряжение на резисторе меняет не только величину, но и полярность, то есть, ток меняет направление. В интервал времени, когда конденсатор заряжается, ток течет в одном направлении, а в следующий период конденсатор разряжается, и ток течет в обратном направлении.

Но сейчас меня больше интересует другое — замедлить изменение напряжения и посмотреть, что произойдет с током через конденсатор. Напомню, что напряжение на резисторе R1 отображает поведение тока через конденсатор.

То, как быстро изменяется напряжение у генератора прямоугольных импульсов, определяется фронтами импульса. Точнее временем нарастания и спада напряжения. В программе Qucs я могу изменить эти параметры (время переднего и заднего фронта импульса), а в результате замены, выбрав их равными времени самого импульса (или половине периода сигнала), я превращаю прямоугольные импульсы в треугольные. Теперь

напряжение меняется не сразу, резким скачком, а плавно нарастает, то есть, меняется существенно медленнее, чем раньше. Посмотрим, изменит ли это что-нибудь? Может быть, теперь конденсатор перестанет пропускать ток, как он не пропускает постоянный ток? Я не помню, когда я пользовался генератором треугольных импульсов в последний раз, если пользовался, и не могу сослаться на то, что видел результат на экране осциллографа, и пока это только предположение. Достаточно часто предположения остаются предположениями. В этом нет ничего страшного, просто нужно еще раз подумать, полистать учебник, внимательнее присмотреться к схеме, может быть, что-то посчитать.

Но сейчас нужды в этом нет, в программе можно быстро сделать все нужные изменения и увидеть результат.

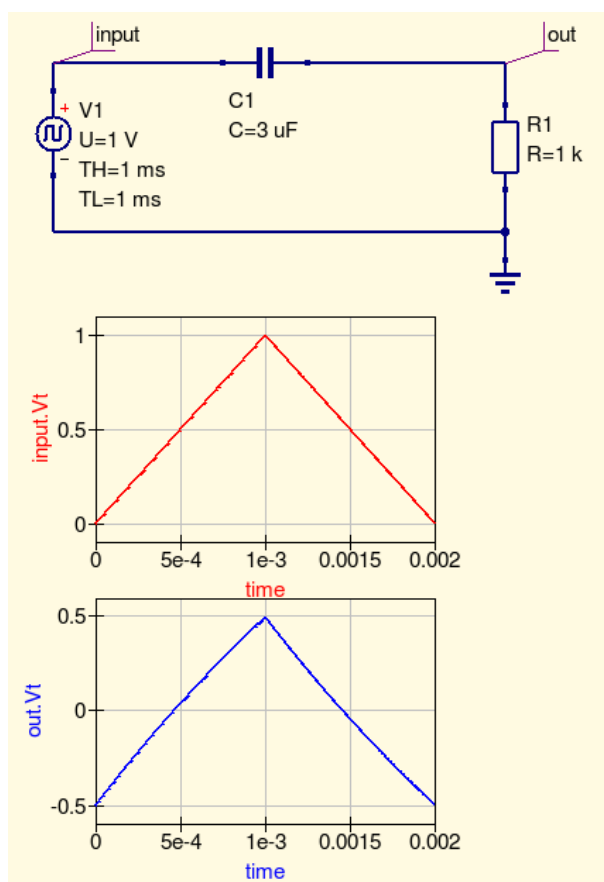


Рис. 1.8. Прохождение треугольных импульсов через конденсатор

Я рад, что не ошибся. Можно проверить еще одно, я не скажу предположение, поскольку это-то я наблюдал на экране осциллографа неоднократно — прохождение синусоидального переменного напряжения через конденсатор.

Конденсатор по отношению к переменному синусоидальному напряжению ведет себя почти как резистор, сопротивление которого зависит от емкости конденсатора и частоты. Вычислить величину этого сопротивления можно по формуле:

$X_c = 1/2\pi fC$, где f — частота в Гц, C — емкость в фарадах, а X_c — сопротивление в омах.

Сопротивление конденсатора переменному току иногда называют фиктивным, иногда реактивным. Для этого есть разные причины. Мне хотелось бы отметить одну особенность поведения конденсатора в сравнении с обычным резистором.

В следующих двух опытах я хочу заменить генератор прямоугольного напряжения синусоидальным генератором, в первом из опытов использовать конденсатор, а во втором заменить его резистором с сопротивлением близким к эквивалентному. После этих двух опытов я поясню, что я хотел отметить в поведении конденсатора и резистора, но прежде напомним, что фаза колебания — понятие относящееся к состоянию движения колеблющегося объекта. Более точное определение:

Фаза колебаний определяет величину смещения в момент времени t , начальная фаза определяет величину смещения в момент начала отсчета времени ($t = 0$).

Или иначе

Фаза колебаний — это угловая мера времени, прошедшего от начала колебаний.

В сущности эти же понятия переносятся и на переменный ток.

Итак, первый опыт с конденсатором, второй с эквивалентным резистором.

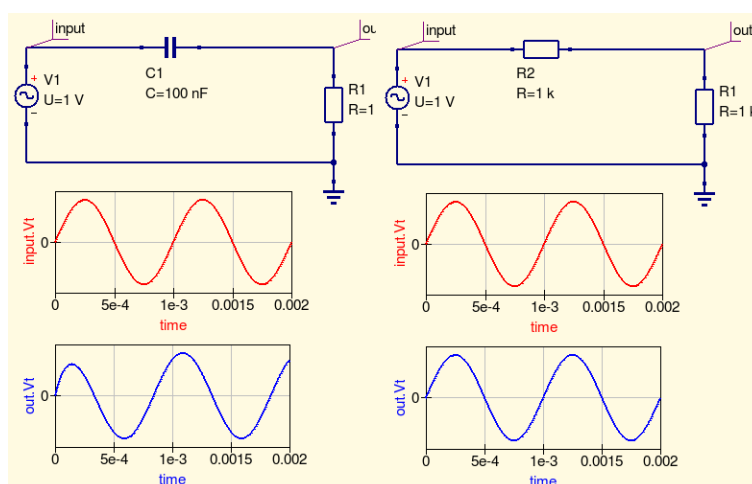


Рис. 1.9. Конденсатор и резистор в цепи синусоидального переменного напряжения

Верхние графики на рисунке — это напряжение, а нижние — ток. Для резистора справедливо, что напряжение и ток через резистор всегда находятся в фазе (в одной фазе колебания), тогда как ток через конденсатор опережает по фазе напряжение на нем. В момент времени, соответствующий 1 мС ток (на нижнем графике) почти достигает максимального значения, тогда как напряжение (верхний график) проходит через нулевое значение. Сопротивление резистора называют активным сопротивлением, а конденсатора — реактивным. И есть еще один компонент электрических цепей, имеющий реактивное сопротивление. Это индуктивность (катушка индуктивности). Ее реактивное сопротивление можно вычислить по формуле:

$$X_L = 2\pi fL, \text{ где } f \text{ — частота в Гц (герц), } L \text{ — индуктивность в Гн (генри).}$$

Для индуктивности справедливо, что ток через индуктивность отстает по фазе от напряжения на ней. Эти два реактивных сопротивления, величина которого зависит от частоты переменного тока, имеют интересную особенность, когда соединяются вместе. Рассмотрим, например, параллельное соединение конденсатора и катушки индуктивности.

Есть частота, на которой реактивное сопротивление конденсатора равно реактивному

сопротивлению индуктивности. Эту частоту называют резонансной частотой, а параллельно соединенные конденсатор и индуктивность называют колебательным контуром.

Дело в том, что если подать в этот контур возмущающее воздействие (импульс напряжения), то в контуре возникают колебания. Мы видели, подключая конденсатор к батарейке, что он заряжается. Аналогично, если мы подадим импульс напряжения, то зарядим конденсатор. Но в колебательном контуре он начнет не просто разряжаться через индуктивность, а перезаряжаться, успевая сделать это несколько раз, пока не потеряет всю энергию заряда.

Формулу для вычисления резонансной частоты легко получить из равенства $X_C = X_L$:

$$f_{рез} = 1/2\pi\sqrt{LC}$$

Кроме параллельного соединения конденсатора и индуктивности, мы можем соединить их последовательно, но и в этом случае будет резонансная частота, будет последовательный колебательный контур. Для параллельного колебательного контура справедливо, что на частоте резонанса его сопротивление максимально, для последовательного — минимально. Свойства параллельного и последовательного колебательных фильтров часто используют в разного рода фильтрах. Самым примечательным, в этом отношении может быть входной колебательный контур радиоприемника. Конденсатор у него делают переменным, но при любом его значении контур имеет вполне определенную резонансную частоту, выделяя из множества частот передающих радиостанций, волны которых принимает антенна, одну частоту, одну радиостанцию.

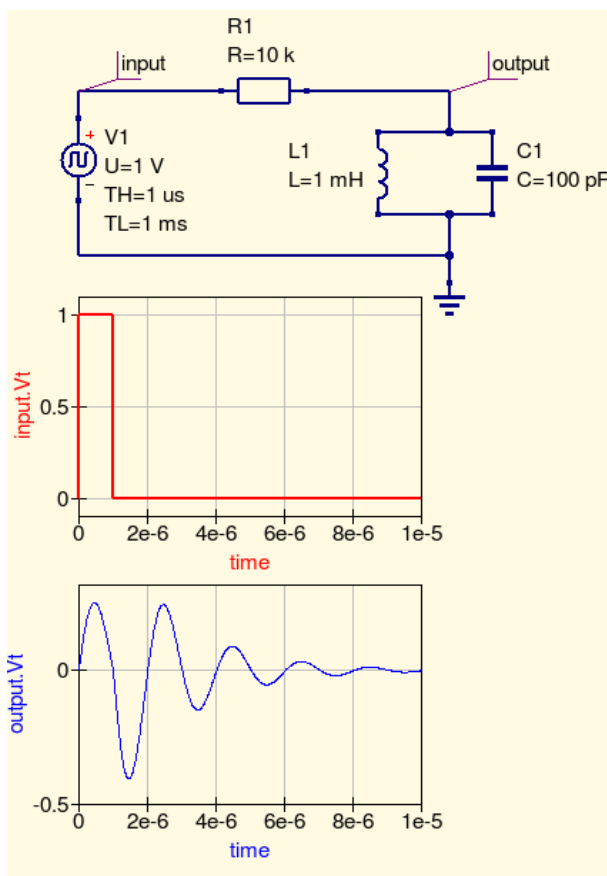


Рис. 1.10. Возникновение колебаний в параллельном колебательном контуре

На нижнем графике видно, как после перехода возбуждающего импульса в ноль на выходе

колебательного контура (точка наблюдения output) продолжают затухающие колебания.

Если говорить о соединении (параллельном и последовательном) однотипных элементов электрических схем: резисторов, конденсаторов, индуктивностей, — то можно запомнить простые правила. Для резистора величина обратная сопротивлению называется проводимостью. При последовательном соединении резисторов складываются их сопротивления. При параллельном соединении резисторов складываются их проводимости.

$$R_{\text{послед}} = R1 + R2$$

$$1/R_{\text{паралл}} = 1/R1 + 1/R2$$

Для конденсаторов — при параллельном их соединении складываются их емкости, при последовательном их соединении складываются величины, обратные емкостям.

$$C_{\text{паралл}} = C1 + C2$$

$$1/C_{\text{послед}} = 1/C1 + 1/C2$$

При последовательном соединении индуктивностей складываются их индуктивности, при параллельном — величины обратные индуктивности.

$$L_{\text{послед}} = L1 + L2$$

$$1/L_{\text{паралл}} = 1/L1 + 1/L2$$

В реальных схемах редко применяют параллельное включение однотипных элементов, но часто можно встретить параллельно включенные резистор и конденсатор. В описании схемы можно прочесть, что конденсатор «шунтирует» резистор. Это справедливо для переменного напряжения начиная с некоторой частоты и выше. Сопротивление конденсатора с ростом частоты уменьшается, чем и достигается эффект шунтирования. Зачем применяют такое включение? Для правильной работы схемы на постоянном токе необходим резистор с определенным сопротивлением. Но при этом усиление каскада может быть меньше, чем при отсутствии резистора. Чтобы увеличить усиление каскада (для полезного сигнала переменного напряжения) резистор шунтируют конденсатором, подбирая его емкость таким образом, чтобы эффект достигался, начиная с некоторой частоты и выше. Позже, когда мы поговорим об усилителях, можно будет показать влияние шунтирующего конденсатора, а сейчас... я часто употреблял слово сигнал, рассказывая о переменном токе. А что это такое?

Переменные напряжения или сигналы в электрических цепях

К сожалению в учебнике Гоноровского «Радиотехнические цепи и сигналы», по которому я некогда учился, которым пользуюсь при необходимости, определения сигнала я не нашел. Пришлось взять это определение из Wiki-педии:

Сигнал — изменение физической величины, передающее информацию, кодированную определенным способом.

Я тоже склонен придавать слову сигнал относительно всего, что касается электрических цепей, смысл: «несущий информацию».

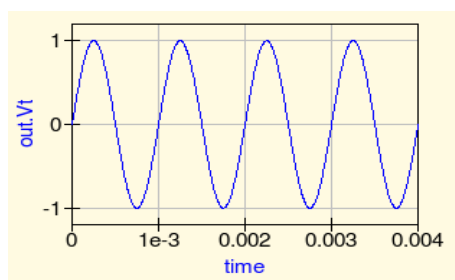


Рис. 2.1. Переменное напряжение частотой 1 кГц от генератора

В этом смысле синусоидальное напряжение, изображенное на рисунке выше, не слишком подходит к определению сигнала, хотя получено оно на выходе генератора низкочастотных сигналов. А, с другой стороны, если я назову его испытательным сигналом, то ситуация несколько изменится. В качестве испытательного сигнала, подключив его к операционному усилителю, мы можем получить информацию о работе усилителя.

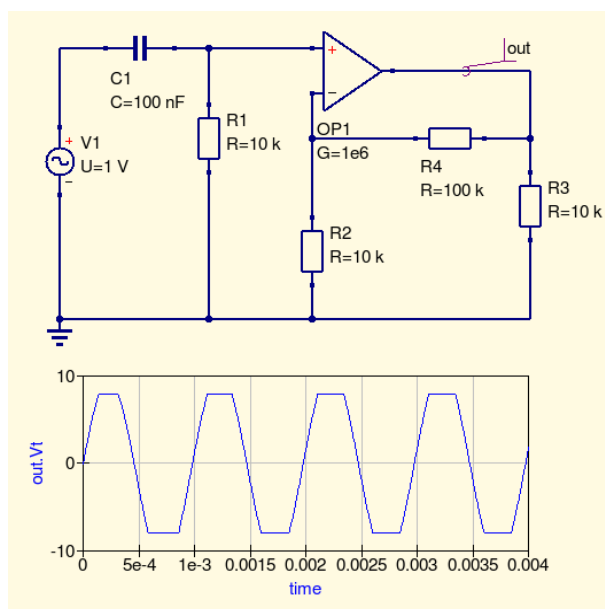


Рис. 2.2. Прохождение испытательного сигнала через усилитель

Наблюдая сигнал на выходе усилителя, можно сделать вывод, что либо следует уменьшить напряжение на входе, либо увеличить напряжение питания усилителя, иначе исходный синусоидальный сигнал, а значит и реальный более сложный сигнал, будет искажаться. Испытательный синусоидальный сигнал принес информацию о правильности работы нашего

испытываемого устройства. Если же мы испытываем проектируемое устройство, то он принесет информацию о правильности расчетов, выбора режима работы, готовности проекта к воплощению в жизнь.

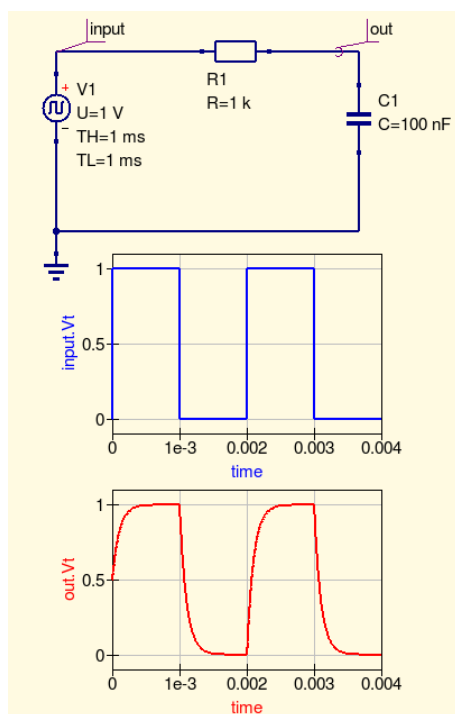


Рис. 2.3. Прохождение прямоугольного испытательного сигнала через RC цепь

Используя прямоугольный испытательный сигнал, легко заметить наличие в устройстве RC цепей типа, изображенного на рисунке, и которые называются «интегрирующими». Можно возразить, зачем нужен испытательный сигнал, если можно увидеть эту цепь на плате устройства? Не всегда так. Иногда подобные интегрирующие цепи являются «свойством» электронного компонента, их не включают в схему намеренно, их не устанавливают на плату сознательно, но они присутствуют и влияют на свойства реальных сигналов в электронном устройстве. Чтобы оценить меру этого влияния, используют испытательные сигналы.

Вообще, сигналы в реальных электронных устройствах могут быть достаточно сложными. Соответственно, сложными могут быть и испытательные сигналы, как, например, испытательные сигналы для проверки и настройки телевизоров.

Далеко не всякий радиолюбитель может позволить себе покупку всех необходимых ему приборов. Особенно начинающий. Если опытный радиолюбитель, столкнувшись с проблемой нехватки оборудования, может самостоятельно изготовить недостающее, то начинающему радиолюбителю я бы советовал использовать для знакомства с электроникой, радиотехникой и электрикой программы САПР. Практически все они имеют в арсенале программы большое количество источников напряжения: от источников постоянного и переменного синусоидального напряжения, до источников сигналов сложной формы. К таким сигналам я отнес бы амплитудно-модулированные сигналы.

Некогда амплитудно-модулированные сигналы использовались всеми вещательными радиостанциями. Теперь им на смену пришла частотная модуляция, но амплитудная модуляция сохраняет свое значение при вещании на длинных и средних волнах, при формировании телевизионного видеосигнала.

Даже для того, чтобы понять работу детекторного приемника, лучше воспользоваться

генератором амплитудно-модулированного сигнала. Программа Qucs имеет в наборе источников напряжения источник с АМ-модуляцией. Изменяя свойства низкочастотного сигнала и генератора амплитудно-модулированного сигнала, можно наблюдать на графиках (соответствующих наблюдению на экране осциллографа), как меняется вид конечного сигнала при разной «глубине» модуляции, при разных частотах сигнала и несущей частоты.

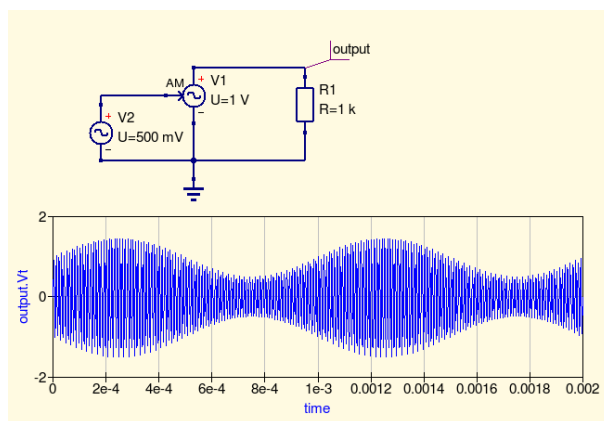


Рис. 2.4. Применение источника АМ-сигнала

Если вместо резистора R1 подключить схему детекторного приемника, то можно наблюдать, что происходит с сигналом радиостанции, принятой антенной радиоприемника. Сигнал радиостанции, когда передают последние известия или музыку, конечно будет очень отличаться от того, что на рисунке, но в те моменты, когда радиостанция передает сигналы точного времени, они будут выглядеть именно так.

Читая вопросы, которые начинающие радиолюбители задают на любительских форумах, я вижу, как быстро меняется все области электроники: недоступные еще лет 10-15 для любого интересующего микроконтроллеры, сегодня интересуют начинающих не более, чем транзисторы или операционные усилители. Я не думаю, что сегодня начинающие радиолюбители собирают детекторные приемники, хотя жаль — сущность радиолюбительства не в достижении некоторого, пусть и достойного, результата, а в том удовольствии, которое можно получить на пути к поставленной цели. Это и удовольствие от работы руками, это и удовольствие от работы головой, это и удовольствие от поиска и чтения книг по предмету, от разумного планирования работы, от осуществления экспериментов, которые будут только этапами на пути к окончательной разработке. Не знаю, как вам, мне поиск решения всегда доставлял большее удовольствие, чем получение окончательного решения. Достигнув цели, если это не относится к работе, ты понимаешь, что завершился замечательный отрезок жизни, наполненный сомнениями, размышлениями, надеждами и разочарованиями, а из подобных отрезков и складывается вся жизнь. Чем ярче эти отрезки, тем полнее и насыщеннее ваша жизнь.

Часто начинающие радиолюбители спешат найти нечто главное, что сразу позволило бы им быть «на ты» с электроникой. Как ненужное, они отбрасывают простые схемы, пытаются повторить самые «крутые» решения; как скучные предметы, они обходят школьные знания из курса физики; они не хотят «корпеть» над схемой, пытаются понять ее работу или причину ее неправильной работы — в результате они обкрадывают себя, превращая увлечение в разновидность профессиональной работы, но без должной подготовки и с «недолжными» результатами.

Я не призываю начинающих радиолюбителей к крайнему педантизму — нельзя заикливаться на доскональном изучении всех элементов электрической схемы, всегда

следует ограничивать этот процесс временными или смысловыми рамками, но и не задаться вопросами: «Как это устроено? Почему так? Зачем нужны эти элементы?», — не задаться этими вопросами, значит потерять многое и даже не из области знания предмета, а из удовольствия узнать что-то новое о вещах, подчас, привычных.

Но вернемся, пожалуй, к сигналам, а скорее, к тому, как наблюдать и как интерпретировать увиденное в электрических цепях. Для придания предметности рассуждениям воспользуемся конкретной схемой достаточно распространенного функционального узла — усилительного каскада на транзисторе.

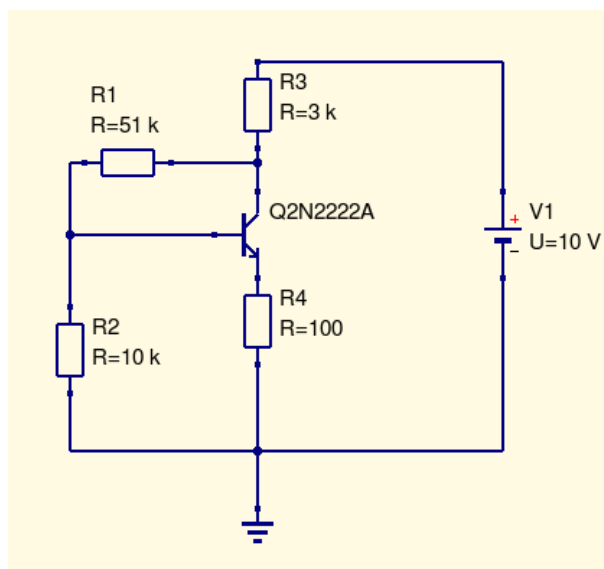


Рис. 2.5. Усилитель на транзисторе с общим эмиттером

Какими сигналами можно (или следует) воспользоваться для исследования устройства?

Давайте, поступим так — я забуду, все, что знаю об этом устройстве, и все, что знаю вообще из этой области, кроме того, что уже рассказал выше. В этом смысле мне пока ясно не так много — прямоугольники с обозначениями R1, R2 и т.д., несомненно, резисторы. Сопротивление каждого из них проставлено ниже. К слову, в документации схема может не иметь такого вида: каждый резистор будет обозначен, но сопротивление каждого из них будет проставлено в дополнительной спецификации, как и марка транзистора, и тип источника питания.

Сопротивление, обозначенное как 3 к, должно обозначать 3 килоома. Префикс к — кило — часто употребляется там, где следует добавить множитель 1000 к основным единицам. Это и киловольт, и килогерц, и килоом. Если префикс М, то это мега, множитель 1000000, тогда как м, будет означать милли — 0.001: мА (миллиампер), мС (миллисекунда) и т.д.

Из знакомых мне элементов схемы я вижу только V1 — источник ЭДС, батарейку с напряжением 10 В. Что такое транзистор, я пока не знаю. И я вижу только две тропинки от этого камня на распутии — первая с указателем «Потом, когда-нибудь разберемся», и вторая «Узнай о транзисторе все». Возможно, здесь наши пути разойдутся. Одни из вас отправятся по первому пути, другие по второму, тогда как я намерен пройти по второму пути ровно столько, сколько буду видеть первый путь. Как только он начнет теряться из вида, я перейду на него. Может кого-то из вас и встречу по дороге.

Итак, что мне интересно узнать о транзисторе?

Конечно, я не против бы узнать, что у него внутри, и как это все обустроено, но хватит ли

у меня терпения и времени окунуться в физику полупроводников, почитать что-нибудь из квантовой физики? При всем моем к ним уважении ни терпения, ни времени не хватит. Поэтому я хотел бы ограничиться знанием того, как ведет себя транзистор в данной схеме, когда меняются значения резисторов. И даже, скорее не это, а то, как реагирует транзистор на изменение входного тока. Схема, показанная на рисунке, это схема включения транзистора с общим эмиттером, то есть, эмиттер — общий вывод транзистора для входного и выходного сигналов. Скорее всего, входным током будет ток базы, а выходным ток коллектора. Поэтому для следующего эксперимента я хотел бы воспользоваться испытательным сигналом, который просто увеличивал бы мне напряжение до некоторой величины, скажем, в 2 вольта, а я посмотрел бы, как меняется ток коллектора. Из схемы я для этого эксперимента удалю все резисторы, назначения которых не понимаю, оставлю только резистор в цепи коллектора транзистора и добавлю резистор в цепь базы. На всякий случай. Преимущество программы — я не боюсь сжечь транзистор, даже если ошибусь, но привычку добавлять резистор «на всякий случай» я хочу вырабатывать у себя с самого начала — она может быть полезна, когда я буду экспериментировать с макетной платой и реальными компонентами.

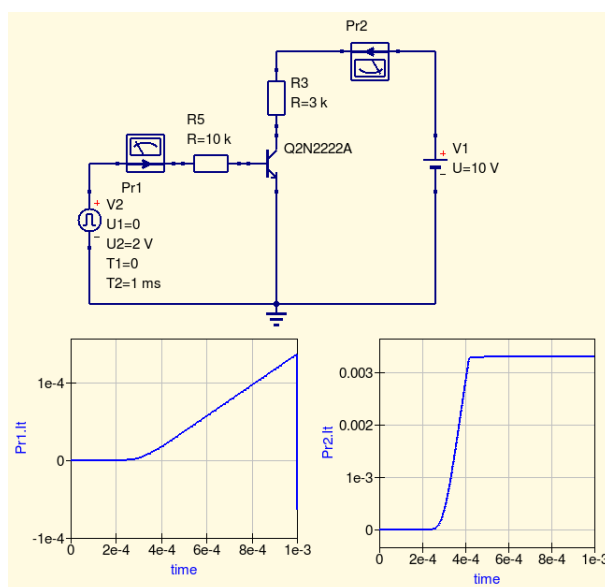


Рис. 2.6. Исследование транзистора с помощью линейно нарастающего напряжения

Источник линейно нарастающего напряжения от 0 до 2 вольт в программе Qucs я могу получить из источника импульсного напряжения, если задам в свойствах импульса время нарастания фронта, равное времени окончания импульса. Измерители тока Pr1 и Pr2, спасибо программе, позволят мне наблюдать за изменением токов.

Что можно сказать по поводу графиков на рисунке? Во-первых, до момента времени 0.3 миллисекунды ничего интересного не происходит. И ничего интересного, во-вторых, не происходит с коллекторным током после 0.4 миллисекунд (график справа). Если понаблюдать за изменением напряжения источника сигнала V2 в этом интервале времени, то можно выяснить, что напряжение меняется, примерно, от 0.5 до 0.9 вольт. Чтобы придать графикам больше выразительности, в программе можно соответственно изменить значения начального и конечного напряжений импульса источника V2.

Кстати, такой эксперимент можно проделать и с реальными компонентами на макетной плате, но не думаю, что в вашей лаборатории есть специализированный прибор, а получить результаты средствами, обычно доступными в любительских условиях, займет и больше времени, и потребует от вас сосредоточенности, чтобы не ошибиться, используя один

мультиметр при определении токов в двух разных цепях, и не испортить прибор, но, вместе с тем, опыты с реальными компонентами убеждают больше, чем работа с программой, а работа с реальными приборами вырабатывает привычку проверять правильность включения до подачи напряжений на схему. Так что, как всегда, в любом варианте есть свои преимущества, и свои недостатки, а принимать решение вам. Лично я предпочел бы довести все эксперименты до конца за компьютером, а когда результаты будут в полной мере осмысленны, перенести опыт на макетную плату. И не обязательно в качестве источника V2 использовать испытательный сигнал от генератора, вполне можно обойтись батарейкой с напряжением 1.5 В, двумя резисторами и потенциометром, который менял бы напряжение от 0.5 до 0.9 В. Как выбрать сопротивления резисторов и потенциометра? Примените закон Ома, проверьте это в программе, а когда все станет ясно, перенесите конструкцию на макетную плату.

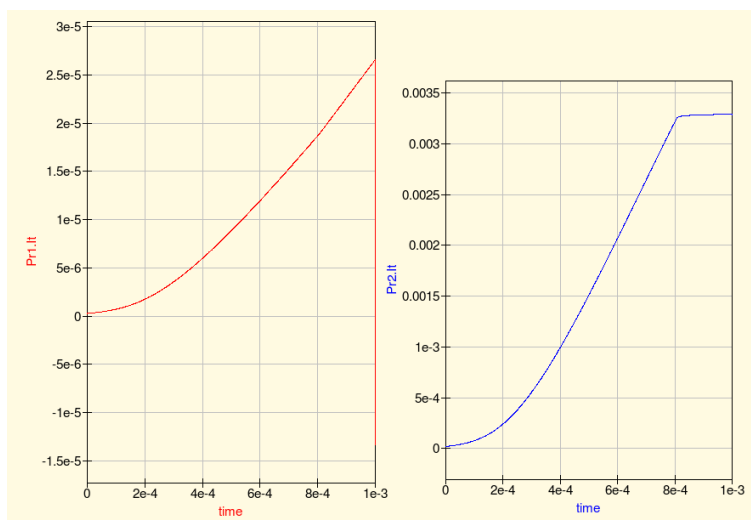


Рис. 2.7. Графики изменения токов базы и коллектора в интервале $V2 = 0.5 \div 0.9$ В

Выберем точку на графике, соответствующую времени 0.4 мС. Базовый ток, согласно первому графику, равен 6 мкА (микроампер), а ток коллектора (из второго графика) равен 1000 мкА (1 мА). Их отношение, 166, называется статическим коэффициентом передачи тока или статическим коэффициентом усиления по току. Эту величину мы используем, когда рассчитываем каскад по постоянному току.

Если мы возьмем приращение базового тока, соответствующее интервалу времени от 0.4 до 0.6 мС. Приращение тока базы составит 11 мкА, при этом приращение тока коллектора будет 1000 мкА (1 мА), а отношение приращения коллекторного тока к приращению базового тока, которое можно назвать динамическим коэффициентом усиления тока, 90. То есть, изменения базового тока будут вызывать в 90 раз большее изменение коллекторного тока.

Если заменить измерители тока в схеме на рисунке 2.6 измерителями напряжения, первый из которых будет подключен к базе и эмиттеру, а второй к коллектору и эмиттеру, то мы получим графики изменения напряжений.

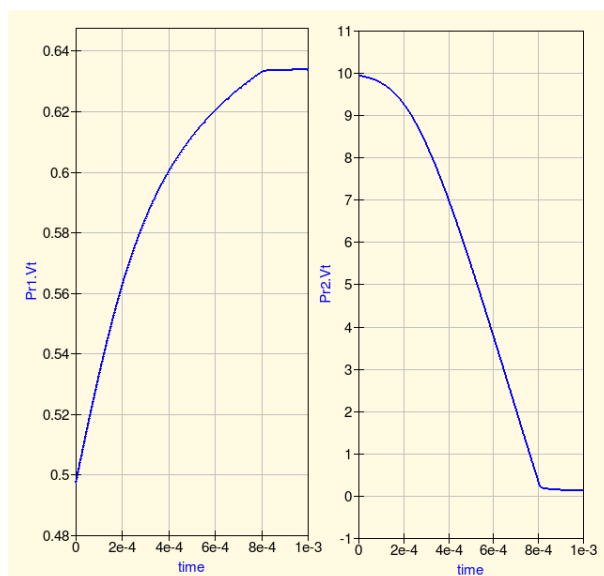


Рис. 2.8. Графики изменения базового и коллекторного напряжений

Выбрав на левом графике (входное напряжение) изменение напряжения в том же интервале времени 0.4 мс-0.6 мс, равное 0.02 В (или 20 милливольт), определив изменение напряжения в этом же интервале на правом графике (выходное напряжение), равное 3.5 В (7 — 3.5), мы можем вычислить их отношение $3500/20 = 175$, имеющее смысл коэффициента усиления по напряжению.

В программе Qucs можно проверить, насколько этот результат совпадает с усилением по напряжению в схеме усилителя на одном транзисторе при подаче испытательного синусоидального сигнала 1 мВ на вход.

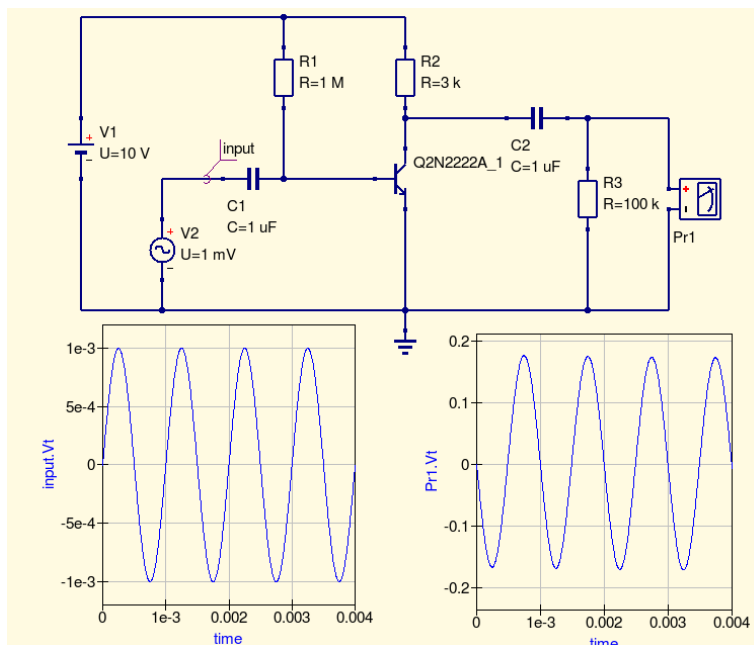


Рис. 2.9. Испытание усилителя с помощью генератора

Взяв отношение амплитуд на полученных графиках, мы получим отношение 170 мВ/1 мВ, равное 170. В данном случае совпадение очень хорошее. Если бы мы выбрали другой

интервал времени, то совпадение получилось бы хуже. Но, если обратиться к рисункам 1.1 и 1.2, то можно заметить, что ток коллектора и напряжение база-эмиттер, достаточно хорошо соотносятся именно с выбранным интервалом времени.

И еще одно: на графиках рисунка 2.8 видно, что когда напряжение на базе растет (входное напряжение), напряжение на коллекторе падает; на диаграммах напряжения рисунка 2.9 это тоже хорошо заметно. О таком поведении входного и выходного напряжения говорят, что они противофазны. Этот момент оказывается важным, когда мы организуем в схеме обратную связь.

Если подать часть выходного напряжения на вход, то мы получим обратную связь — связь выхода со входом. В зависимости от того, будет ли эта часть выходного напряжения противофазна входному напряжению, и будет вычитаться из входного, или будет синфазна с входным напряжением, складываясь с ним, мы получим отрицательную обратную связь в первом случае и положительную обратную связь во втором.

Посмотрим, для чего нужны резисторы R1 и R4 на рисунке 2.5.

Конечно, резистор R1 служит в первую очередь для создания тока базы, задающего режим работы усилительного каскада по постоянному току. Но, если в схеме рисунка 2.9 аналогичный резистор подключен к источнику питания (R1), то в схеме рисунка 2.5 резистор R1 подключен к выходу усилительного каскада. Вместе со входным сопротивлением каскада он образует делитель для выходного напряжения и часть выходного напряжения алгебраически складывается со входным. Как мы видели напряжение на коллекторе противофазно входному, значит, обратная связь будет отрицательной.

За счет того, что часть выходного сигнала вычитается из входного усиление каскада уменьшается. Казалось бы, зачем нам уменьшать усиление каскада? При вычитании напряжения во входной сигнал вносятся (частично) все «примеси», которые создаются усилителем, но отсутствуют в исходном, входном сигнале. А значит, и выходной сигнал будет содержать эти «примеси» в меньшем количестве. Ради этого, уменьшения вредных искажений исходного сигнала, применяют отрицательную обратную связь.

И еще. Часть выходного сигнала подается на вход параллельно входному сигналу, поэтому в данном случае говорят о параллельной обратной связи.

Посмотрим, как соотносятся в части фазы входное напряжение, выходное и напряжение на резисторе R4.

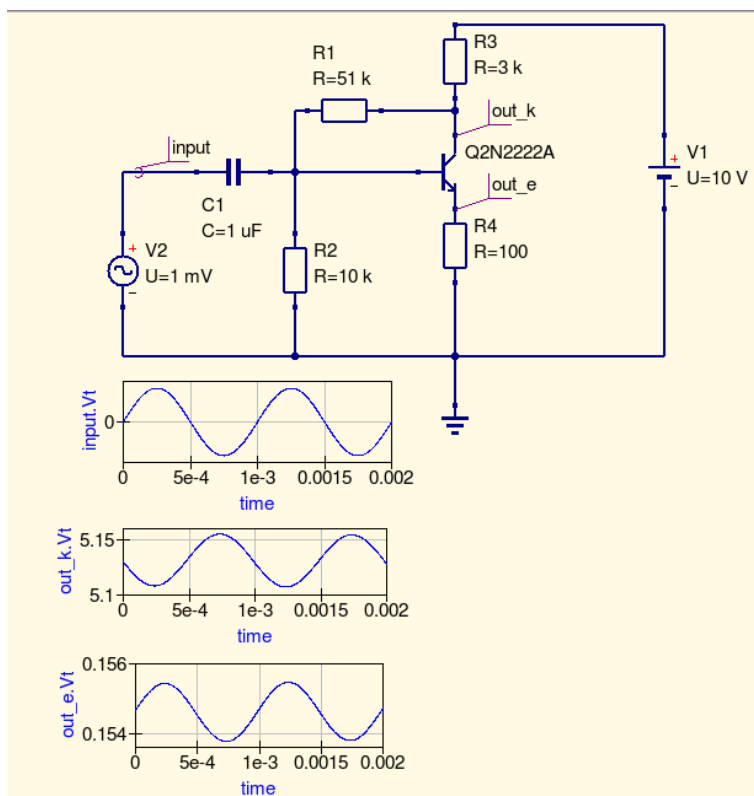


Рис. 2.10. Переменные напряжения в разных точках схемы усилителя

Как и на предыдущем рисунке напряжение на коллекторе (out_k) противофазно входному (input), а напряжение на резисторе R4 (out_e) синфазно входному. Если бы мы подавали обратную связь с этого резистора, то получили бы положительную обратную связь.

Как формируется напряжение на резисторе R4? Входной ток (базовый) складывается с коллекторным током (напомню, в технике принято направление тока от плюса к минусу), в результате получается ток эмиттера, создающий падение напряжения на резисторе R4. Но вклад базового тока, который в 50-100 раз меньше коллекторного, в этом формировании мал, так что напряжение, в основном, формируется выходным, коллекторным током. А напряжение между базой и эмиттером, которое и будет усиливаться, есть разница между напряжением источника сигнала и напряжением на резисторе R4:

$$V2 = U_{Б-Э} + U_{R4}; U_{Б-Э} = V2 - U_{R4}$$

Таким образом, резистор R4 — резистор отрицательной обратной связи, но теперь последовательной (для источника сигнала входное сопротивление и сопротивление R4 включены последовательно).

Очень часто в описаниях схем можно встретить выражения вида «сигнал с коллектора транзистора через конденсатор...». О каком сигнале идет речь? В первую очередь о реальном сигнале, который будет присутствовать в реальном устройстве. Но это в полной мере можно отнести к испытательному сигналу, который может быть подан на устройство с целью получения информации о его правильной работе, или получения информации о свойствах устройства. И очень часто, хотя речь идет о переменном напряжении в схеме, говорят о сигнале.

На практике существующие сигналы в реальном устройстве или прохождение испытательных сигналов через устройство наблюдают с помощью осциллографа — этот

прибор показывает на экране форму сигнала; как правило, он имеет сетку, помогающую оценить величину сигнала и его временные параметры.

Кроме усиления сложных и простых сигналов в электрических цепях, используются схемы разного рода преобразования сигналов, изменяющихся по одному закону, в сигналы, изменяющиеся по другому закону, то есть, преобразование одних сигналов в другие. Мы уже видели на рисунке 2.4 вид преобразованного сигнала с помощью амплитудной модуляции информационным сигналом высокочастотного сигнала, который в виде радиоволны отправится к радиоприемнику. Такой прием преобразования сигнала позволяет при меньших мощностях передатчика увеличить дальность передачи информации.

В радиоприемнике, в колебательном контуре подобном изображенному на рисунке 1.10, настроенном на частоту (ее называют несущей частотой) радиостанции, этот сигнал выделяется из других сигналов и вновь преобразуется. В простейшем случае детектируется. Самый простой радиоприемник поэтому называется детекторным. Преобразователь в данном случае — это диод. Добавим к схеме на рисунке 2.4 полупроводниковый диод.

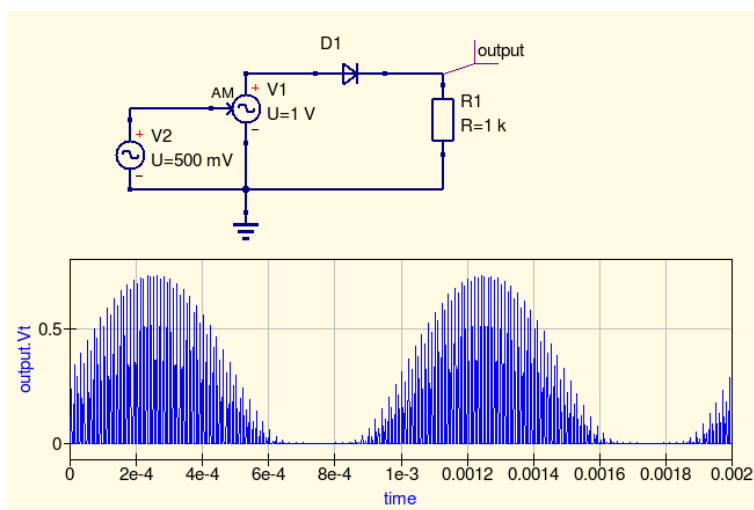


Рис. 2.11. Детектирование амплитудно-модулированного сигнала

Если теперь провести плавную линию, огибающую высокочастотный сигнал, то мы должны увидеть информационный сигнал (точнее испытательный сигнал), формируемый генератором V2. Для выделения огибающей можно воспользоваться частотными свойствами конденсатора. Его сопротивление на низкой частоте, а испытательный сигнал от V2 низкочастотный, достаточно велико и не должно влиять на напряжение output, которое падает на резисторе нагрузки R1. Но сопротивление на частоте несущей, а эта частота много больше, будет очень невелико — конденсатор как бы «шунтирует» высокочастотный сигнал, пропуская его через себя и не давая ему создать напряжение на резисторе нагрузки.

Добавив в схему конденсатор мы получим в результате на сопротивлении нагрузки R1 наш информационный сигнал:

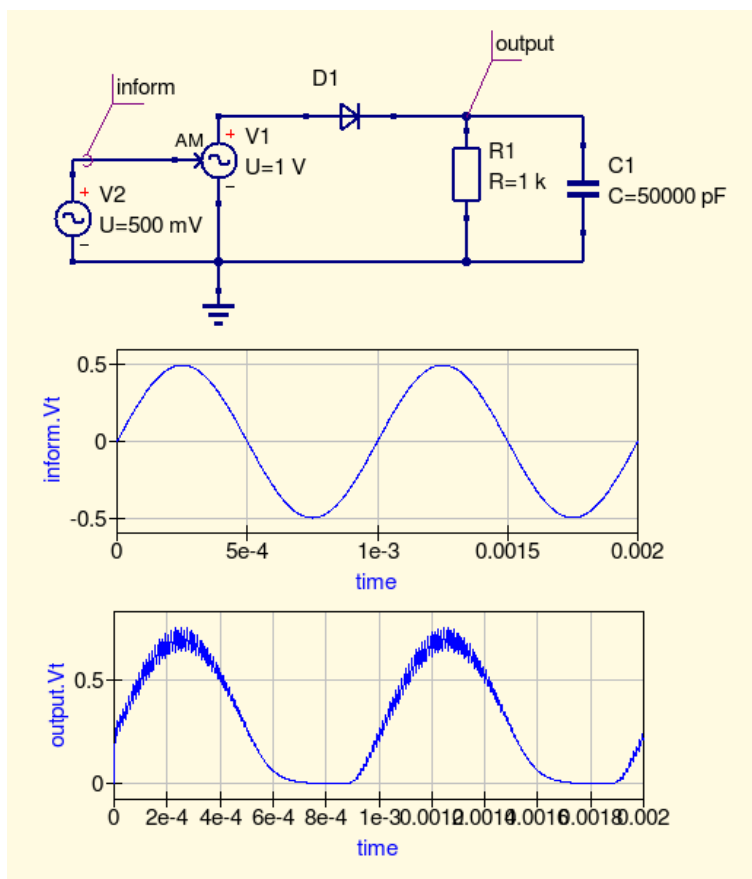


Рис. 2.12. Выделение информационного сигнала в детекторном приемнике

Исходный сигнал **inform** и полученный нами **output** изменяются по одному закону, это синусоидальный сигнал, и отличаются незначительными искажениями преобразованного сигнала. При правильных преобразованиях этого искажения может и не быть. В реальном детекторном приемнике в качестве нагрузки **R1** использовались высокоомные наушники.

Очень часто в качестве составляющих узлов в радиотехнике, электронике, автоматике используются усилители.

Небольшой рассказ об усилителе и его характеристиках

Собственно, этот рассказ уже начался с появлением на сцене транзистора. В усилителях чаще всего используют активные электрические компоненты: электронные лампы, транзисторы и т.д. Эти активные элементы используют энергию источника ЭДС (источника питания) для того, чтобы усилить сигнал, то есть, сохраняя закон изменения переменного напряжения сигнала увеличить его мощность, усиливая либо ток, либо напряжение, либо и то, и другое.

Современная электроника все чаще использует для этих целей микросхемы — готовые функциональные узлы, имеющие заданные параметры и построенные из большого количества элементов, скрытых внутри корпуса микросхемы. Но, чтобы лучше понять назначение параметров микросхем, полезно разобраться с простым усилительным каскадом. Все эксперименты мы проведем с каскадом на транзисторе, изображенном на рисунке 2.10.

Первое, на что хотелось бы получить ответ, а зачем нужен усилитель? Иногда он и не нужен. Но, представьте себе, что вы хотите озвучить выступление оратора. Он на трибуне, а рядом, на площадке перед трибуной, собралось много народа. Самый простой электродинамический микрофон может развить напряжение 0.5-1 мВ, при этом его сопротивление примерно 250 Ом. Если вы подключите проводами такой микрофон к громкоговорителю, то едва ли услышите что-то. Хотя бы из-за того, что сопротивление громкоговорителя обычно 4-8 Ом. Посмотрим, что у нас получится, если мы используем синусоидальный источник сигнала, добавим к нему два резистора, сопротивление которых будут повторять сопротивление микрофона и громкоговорителя.

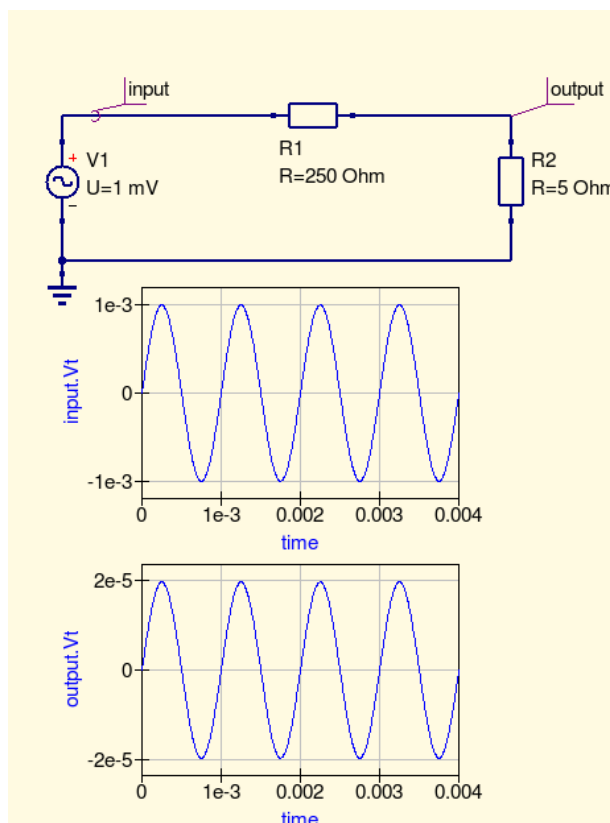


Рис. 3.1. Модель подключения микрофона к громкоговорителю

Амплитуда сигнала от микрофона (input) 1 мВ, а на громкоговоритель (output) приходит только 20 мкВ (микровольт). И без того слабенький сигнал ослаб еще в 50 раз. Но, может

быть, 20 мкВ достаточно?

Отвлечемся немного от усиления и усилителей и вспомним, что кроме множества электронных устройств: компьютеров, телевизоров, радиоприемников и т.п., — в нашем доме электрический ток выполняет не менее важную работу, освещая наш дом, обогревая его, когда не хватает тепла от батарей центрального отопления, и помогает, во многих домах, готовить еду, крутит мотор стиральной машины и заставляет компрессор холодильника работать, чтобы сохранять продукты долгое время.

Электрический ток выполняет работу, а эффективность этой работы можно оценить по мощности. Если посмотреть на лампочку накаливания, дающую свет, то на ней можно прочесть 220 В, 40 Вт (ватт). 40 Вт — это мощность, которую потребляет спираль лампочки накаливания от бытовой электросети с напряжением 220 вольт и частотой 50 Гц. Обычная лампочка накаливания в 40 Вт не зальет вашу комнату ослепительным светом. А мощность сигнала, подведенного к громкоговорителю мы можем рассчитать по очень простой формуле:

$$P = U \cdot I,$$

где P мощность в Вт, если напряжение U в вольтах, а ток I в амперах.

Подсчет, вообще говоря, верен для постоянного тока. А переменный ток? Он же все время меняется. Чтобы не усложнять жизнь, а можно для каждого мгновения рассчитать мгновенные значения напряжений и токов, получив мгновенные значения мощности, так вот, чтобы этого не делать используют действующее значение переменного напряжения или тока. Оно в 1.41 раз (корень квадратный из двух) меньше амплитудного значения переменного напряжения. И 220 В в розетке — это действующее значение переменного напряжения. Действующее значение, по определению, это такое напряжение переменного тока, которое оказывает такое же действие, что и постоянное напряжение:

Действующее значение переменного тока численно равно такому постоянному току, при котором за время равное одному периоду в проводнике с сопротивлением R выделяется такое же количество тепловой энергии, как и при переменном токе.

Например, лампочка, если на нее подать постоянное напряжение равное действующему значению переменного напряжения будет светиться так же, как и при прохождении переменного тока. Используя действующие значения, можно рассчитывать мощность по вышеприведенной формуле.

И расчет мощности, с учетом того, что ток определится по закону Ома делением 20 мкВ на 5 Ом, дает столь малую величину, что ее трудно сравнивать с мощностью усилителя, который вы используете, когда слушаете музыку. По этой причине вернемся к усилителю.

Когда вы выбираете усилитель для своих целей, вас может интересовать ряд свойств усилителя: входное сопротивление (мы видели, что сопротивление источника и сопротивление приемника сигнала может образовать делитель напряжения); полоса частот (для звукового диапазона обычно не более 20-20000 Гц, а зачастую и меньше); коэффициент нелинейных искажений; ток потребления и т.д. Какой смысл и какое значение имеют многие параметры усилителя рассмотрим, как я говорил выше, на примере схемы однокаскадного транзисторного усилителя, о котором можно сказать, что транзистор включен по схеме с общим эмиттером, а каскад имеет две цепи отрицательной обратной связи — параллельной через резистор $R1$ и последовательной через резистор $R4$. При проектировании усилителя все значения его элементов рассчитываются, но мы используем те значения, что есть, поскольку у нас нет каких-либо требований к этому усилителю.

Если вы захотите повторить схему усилителя на макетной плате, а это было бы очень полезно, то в качестве транзистора можно взять любой типа КТ315, КТ3102 или даже любой

(типа n-p-n), что есть под рукой. Он может быть и типа p-n-p, но тогда следует изменить напряжение питания, и следует иметь в виду, что каждый транзистор имеет свои параметры, которые могут изменить характеристики усилителя.

Первое, что мы определим, это влияние входного сопротивления каскада, следующего за этим, в предположении, что он существует. В качестве этого входного сопротивления мы используем резистор. Меняя величину сопротивления этого резистора мы наглядно сможем увидеть, что входное сопротивление имеет значение.

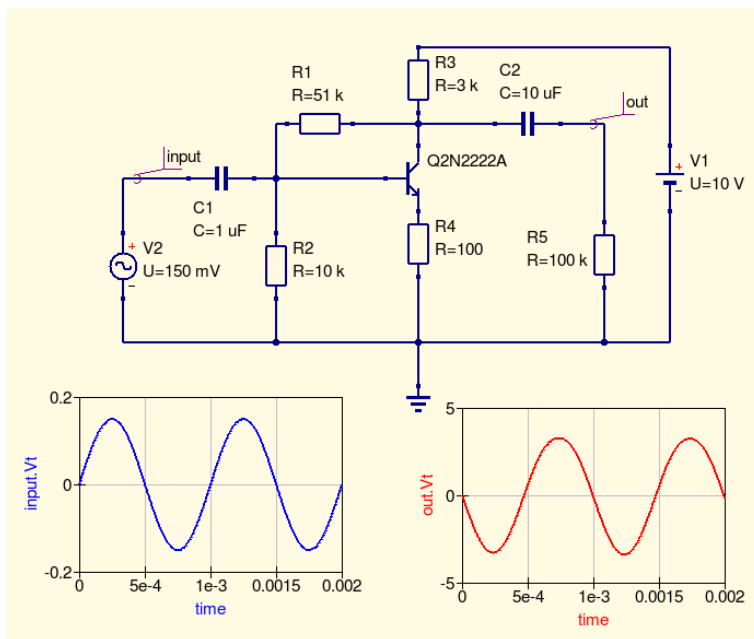


Рис. 3.2. Схема для экспериментов с усилителем

Я намерено выбрал сопротивление R5 достаточно большим, 100 килоом, чтобы, изменив его значение на 1 кОм, показать, как изменится выходное напряжение.

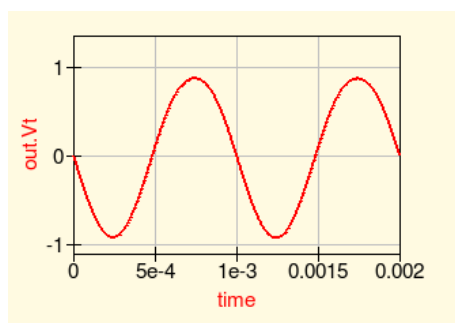


Рис. 3.3. Выходное напряжение при входном сопротивлении следующего каскада 1 кОм

Низкое входное сопротивление может стать причиной не только уменьшения уровня выходного напряжения, но и других неприятностей, связанных с перегрузкой источника сигнала. Например, это может вызывать увеличение нелинейных искажений. Нелинейные искажения усилителя один из важных параметров во многих случаях, в частности в звуковоспроизведении. Наблюдать эти искажения на экране осциллографа можно только в случае весьма значительных искажений, поэтому для оценки этого параметра применяют специальные приборы — измеритель коэффициента нелинейных искажений или анализатор

спектра.

Что такое нелинейные искажения? Хороший усилитель должен на выходе повторять закон изменения входного сигнала. Единственное, что он должен сделать в плане преобразования входного сигнала — это усилить его, то есть, масштабировать. Такие преобразования называются линейными. Но реальный усилитель может несколько исказить закон изменения входного сигнала, и такое преобразование называется нелинейным.

Каждый звук можно представить в виде основного и призвуков, которые определяют «тембр», придают звуку определенную окраску. Благодаря этим дополнительным звукам мы отличаем звучание скрипки от звучания флейты. Можно представить такой звук, как комбинацию основного тона и гармонических составляющих, звуков с частотой кратной основной частоте и значительно меньшими амплитудами, чем основного звука. При нарушении линейности преобразования в усилителе амплитуды гармоник могут усиливаться с другим коэффициентом, чем основной тон. В результате в звук вносятся искажения, которые и называются нелинейными.

Для испытания усилителя применяют генератор синусоидального напряжения. Вот как выглядит испытательный синусоидальный сигнал, если к основной частоте 1 кГц примешался, из-за нелинейных искажений, синусоидальный сигнал с частотой 5 кГц и амплитудой в 10 раз меньшей, чем основная частота.

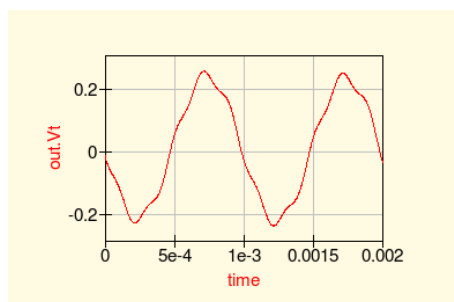


Рис. 3.4. Нелинейные искажения синусоидального сигнала

Для оценки нелинейных искажений используют относительную величину (в процентах), называемую коэффициентом нелинейных искажений:

$$K_{\Gamma} = 100 \sqrt{(A_2^2 + A_3^2 + \dots) / (A_1^2 + A_2^2 + A_3^2 + \dots)}$$

Здесь под знаком радикала в числителе сумма квадратов амплитуд всех гармоник основной частоты, а в знаменателе сумма квадратов амплитуд и гармоник, и основной частоты. Таким образом, коэффициент гармоник представляет собой отношение эффективного значения гармоник к эффективному значению полного выходного сигнала, выраженное в процентах. На практике измеряемые значения не превосходят нескольких процентов и выражение может быть упрощено:

$$K_{\Gamma} = (100 \sqrt{(A_2^2 + A_3^2)}) / A_1$$

То есть, выраженное в процентах отношение корня квадратного из суммы квадратов амплитуд второй и третьей гармоники к амплитуде основного сигнала.

Как правило, наибольшие искажения вносят оконечные каскады усилителей мощности, но правильное построение усилителя требует внимания к каждому каскаду. И не следует относить вопрос возникновения нелинейных искажений только к усилителям звуковых частот, где искажения воспринимаются, как хрипы и неприятные призвуки, в измерительных схемах искажения могут приводить к неверным показаниям, а в радиопередатчиках создавать

помехи окружающим.

Не менее важной характеристикой усилителя является полоса пропускания. Это полоса частот от нижней рабочей частоты до верхней, которая оценивается на уровне — 3 дБ.

Децибелы (дБ) — относительные величины. Для напряжения $K = 20\lg(U_{\text{вых}}/U_{\text{вх}})$ будет коэффициентом усиления, выраженным в децибелах. Операции с децибелами иногда удобнее, так усиление нескольких каскадов будет равно сумме их коэффициентов усиления по напряжению, выраженных в децибелах, а не произведению обычных коэффициентов усиления.

Для определения полосы рабочих частот производят измерения выходного напряжения на ряде частот при постоянном значении входного напряжения. По полученным точкам можно построить характеристику, которая называется амплитудно-частотной. Получить такую характеристику можно и с помощью специального прибора, построенного на базе генератора качающейся частоты. Очень удобно рассматривать амплитудно-частотные характеристики в программах.

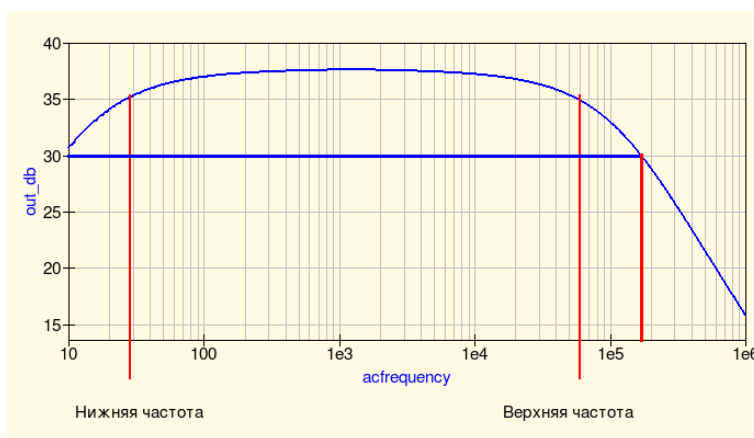


Рис. 3.5. Амплитудно-частотная характеристика каскада по схеме рис. 3.2

Правда транзистор в схеме я заменил более низкочастотным. За верхней рабочей частотой спад амплитудно-частотной характеристики происходит со скоростью 20 децибел на декаду. Если сравнить полученную амплитудно-частотную характеристику с показанной ниже:

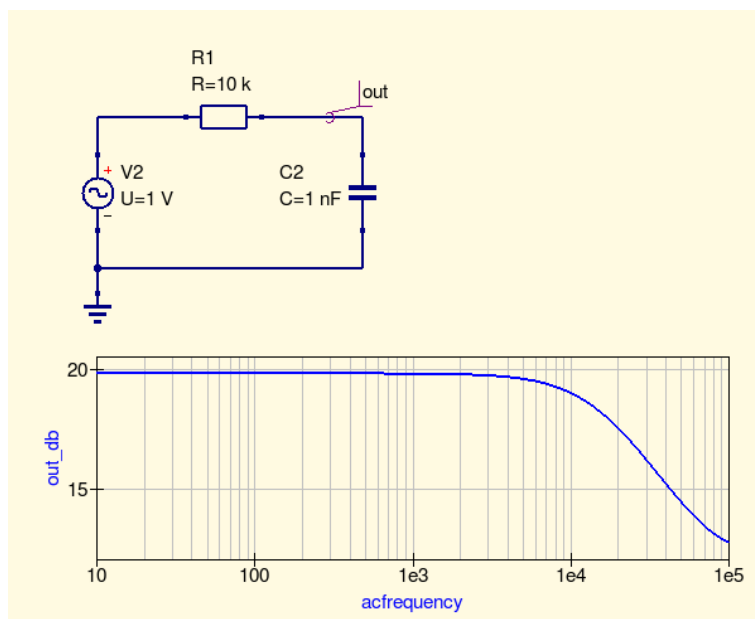


Рис. 3.6. Амплитудно-частотная характеристика интегрирующей RC цепи

то можно заметить большое сходство в области «верхней рабочей частоты». И эта характеристика имеет спад выше некоторой частоты (ее еще называют частотой среза) со скоростью 20 дБ/дек. Вместе с тем, мы знаем, что конденсатор, как реактивный элемент, имеет свойство сдвигать фазу между током и напряжением. Применительно к усилительному каскаду можно сказать, что на верхней граничной частоте фаза выходного напряжения сдвигается относительно входного на 45 градусов, и этот сдвиг продолжается с ростом частоты до 90°. Если усилитель имеет два каскада усиления с верхними граничными частотами f_1 и f_2 , то за частотой f_2 спад амплитудно-частотной характеристики происходит со скоростью 40 дБ/дек, а сдвиг фаз может достигать 180°, при этом коэффициент усиления усилителя на этой частоте может быть больше единицы. А это означает, что при введении общей отрицательной обратной связи, обратная связь на этой частоте превратится из-за сдвига фаз в положительную, а усилитель превратится в генератор.

Анализ амплитудно-частотной и фазочастотной характеристик усилителя позволяет принять правильное решение о глубине отрицательной обратной связи, обеспечивающей достаточную устойчивость усилителя к самовозбуждению. Глубина отрицательной обратной связи, в сущности, это то, насколько уменьшится усиление по напряжению после введения обратной связи, выраженное, например, в децибелах. Если на рисунке 3.5 мы проведем линию параллельно оси частот на уровне 30 дБ, то можем сказать, что получим новую амплитудно-частотную характеристику усилителя с глубиной отрицательной обратной связи 8 дБ. При этом верхняя граничная частота увеличится, снизятся нелинейные искажения. В однокаскадном усилителе мы можем ввести очень глубокую отрицательную обратную связь, не опасаясь, что усилитель потеряет устойчивость. Но в усилителе имеющем несколько каскадов усиления вопрос о допустимой глубине общей обратной связи должен решаться на основании частотных характеристик усилителя.

Как усилитель с положительной обратной связью превращается в генератор, можно посмотреть на примере операционного усилителя:

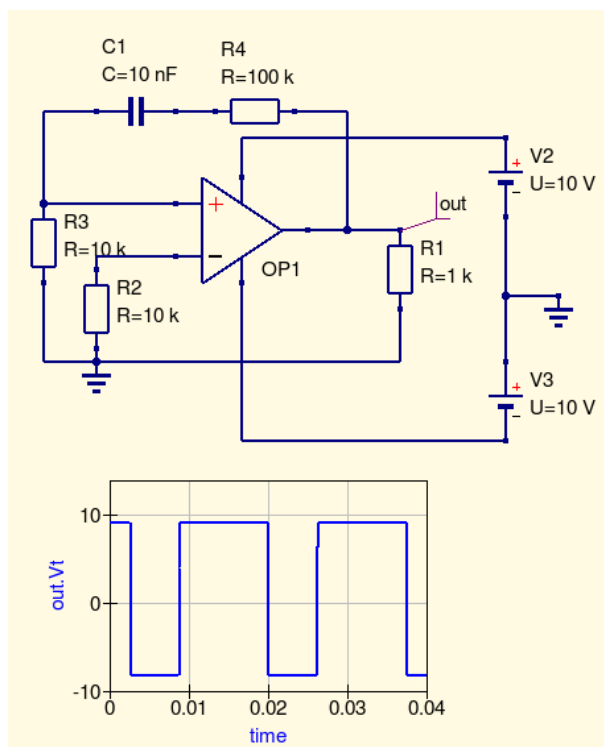


Рис. 3.7. Операционный усилитель с положительной обратной связью

Еще одной важной характеристикой усилителя является отношение сигнал-шум. Любой сигнал на выходе усилителя естественным образом ограничен напряжением питания и не может превышать некоторой максимальной величины. С другой стороны, усилитель усиливает не только полезный сигнал, но и шумы, обычно возникающие во входных каскадах усилителей: это и шумы транзисторов, и шумы резисторов. Вспомним, что ток — это направленное движение зарядов, но на фоне беспорядочного их движения. Это-то беспорядочное движение зарядов и создает шумы. Их можно представить, как сигналы с небольшой амплитудой в широком спектре частот, от самых низких до самых высоких. Сигналы от источника, имеющие амплитуду, сравнимую с амплитудой этих шумов, будут плохо разбираться, будут «тонуть» в шуме.

Ответственными за параметр сигнал-шум, как правило, становятся первые каскады усилителя. В старых аудио схемах можно встретить рекомендации по использованию малошумящих транзисторов. Что это за транзисторы?

В первую очередь, транзисторы с нормированным шумовым параметром. То есть, любой транзистор этого типа имеет шумы не более, чем обозначено в паспорте. Но это не значит, что транзисторы с ненормированными шумами не могут обеспечить отношение сигнал-шум меньше, чем малошумящие транзисторы. И еще один фактор влияет на выбор транзисторов для входных каскадов при построении звуковых устройств — изменение статического коэффициента усиления транзистора при уменьшении тока коллектора, от которого этот параметр зависит. Для разных сопротивлений источника сигнала оптимальный ток коллектора, при котором отношение сигнал-шум максимально, оказывается разным, но имеет тенденцию к малым токам коллектора. При выборе транзистора в этом случае предпочтительнее использовать транзисторы с очень большим значением статического коэффициента усиления по току — они сохраняют достаточно большое значение даже при очень малых токах коллектора.

Повторяя готовую схему, можно столкнуться с ситуацией, когда правильно «спаянная»

схема имеет значение сигнал-шум больше, чем указано в описании схемы. Есть две причины, вызывающие подобный эффект: в описании могли забыть указать, что измерение шумов произведены с использованием «взвешивающего фильтра», фильтра, учитывающего восприятием человека помех. Измерения с фильтром и без него могут сильно разниться. А второй фактор конструктивный. Шумы входного каскада обусловлены не только внутренними причинами, но и влиянием окружения — наводки от сети, от силовых трансформаторов, от радио и телевизионных передатчиков, радиотелефонов и т.д. Эти влияния следует учитывать, правильно располагая входной каскад, защищая его экраном, если это необходимо и правильно конструируя силовые трансформаторы.

Наконец, следует учитывать еще один фактор. Многие начинающие радиолюбители стремятся (начитавшись сообщений в форумах, наслушавшись разговоров приятелей) создать усилитель очень большой мощности. Если они все сделали правильно и получили очень качественный усилитель большой мощности, им не следует забывать, что их труд в некоторой мере пропал даром. Я не говорю о том, что для проверки такого усилителя радиолюбителю нужно иметь очень качественную измерительную аппаратуру, без которой нельзя с уверенностью определить, достигнута ли цель. Я хочу сказать, что реальное прослушивание, скорее всего, будет происходить при значительно меньшей громкости, чем может обеспечить подобный усилитель. Уменьшение громкости — это уменьшение напряжения сигнала на выходе усилителя, тогда как шумы остаются прежними. Только по этой причине отношение сигнал-шум будет меньше, а это не самый маловажный параметр.

Влияние всех параметров усилителя на конечный результат важно не только в области аудиотехники, это важно и при создании собственных измерительных приборов, и в автоматике, и в радиотехнике. Мне кажется, что начинающему радиолюбителю, выбирая собственную конструкцию, следует больше внимания уделить ясному пониманию схемы, а не ее «крутизне», полному пониманию назначения каждого элемента схемы и его влияния на работу устройства, а не погоне за наилучшими параметрами. И в этом смысле, даже если программа не позволяет полностью моделировать работу схемы, рассмотрение функциональных узлов с помощью программы сослужит хорошую службу до того, как будут куплены дорогостоящие детали, как будет потрачено много времени на сборку устройства. И, если что-то получится «не так», легче будет разобраться, что именно «не так».

Транзисторные усилители, а биполярный транзистор имеет три вывода: эмиттер, база, коллектор, — могут выполняться по схеме с общим эмиттером (самая распространенная), с общей базой и общим коллектором. Это подразумевает, какой вывод транзистора будет общим для входного и выходного сигнала. Каждый из вариантов имеет свои преимущества, так схема с общим коллектором имеет самое большое входное сопротивление, но и свои недостатки, например, схема с общим коллектором, ее еще называют эмиттерный повторитель, имеет коэффициент усиления по напряжению меньше единицы.

Иногда название эмиттерный повторитель смущает в сочетании с включением с общим коллектором. Видимо, имеется в виду, что усилитель повторяет (без усиления) напряжение сигнала на нагрузке, включенной в эмиттер транзистора. Отсюда и «эмиттерный повторитель», но схема включения остается с общим коллектором. Начинающего может смущать, что по виду схемы коллектор, как будто, и не общий вывод для входного и выходного напряжения.

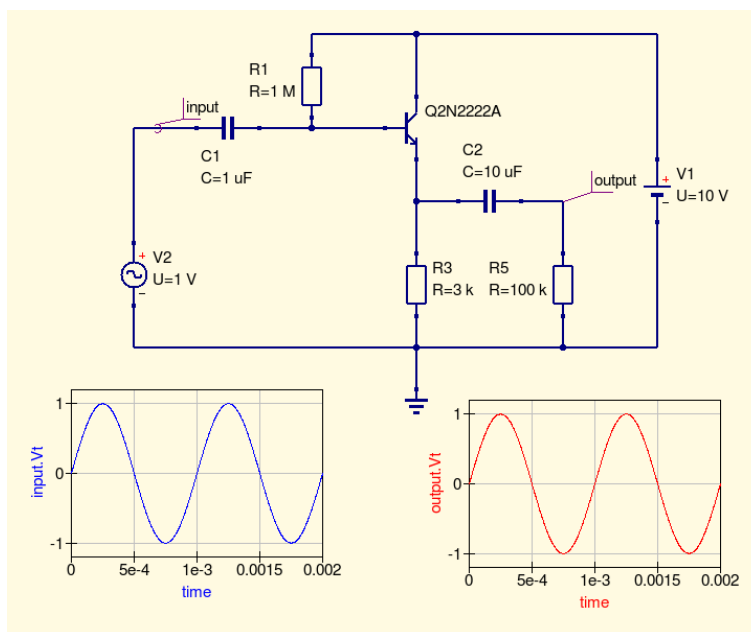


Рис. 3.8. Схема с общим коллектором

Дело в том, что источник питания для сигнала имеет столь малое сопротивление, что можно его закоротить (но только для рассмотрения схемы), а тогда общий провод будет подключен к коллектору транзистора. Иногда при питании от одного источника между каскадами добавляют резистор, понижающий напряжение питания для каскада, включенного после этого резистора. В этом случае обязательно добавляют конденсатор большой емкости с тем, чтобы его сопротивление на частоте сигнала (если полезные сигналы в полосе частот от f_n до f_v , то для частоты f_n — нижней рабочей частоты) было очень небольшим.

Любой реальный источник питания можно представить в виде идеального источника ЭДС и внутреннего сопротивления. Это внутреннее сопротивление очень небольшое, но разное для разных источников питания. Именно оно ограничивает ток, когда мы проверяем батарейку с помощью амперметра, не пора ли ее заменить? Многие батарейки могут показывать нормальное напряжение, скажем, 1.5 V , но работать будут плохо. Обычная практика — переключить мультиметр в режим измерения тока, выбрав наибольший предел (на моем мультиметре это 10 A), и, кратковременно подключив мультиметр к батарейке, проверить ток. Для современных «свежих» батареек он может быть в несколько ампер. Если бы мы имели дело с идеальным источником, то ток (по закону Ома) определялся бы сопротивлением мультиметра, которое весьма невелико, а ток составлял бы десятки и сотни ампер. Кстати, такой ток способны обеспечить автомобильные аккумуляторы, очень большой ток могут отдавать аккумуляторы мобильных телефонов.

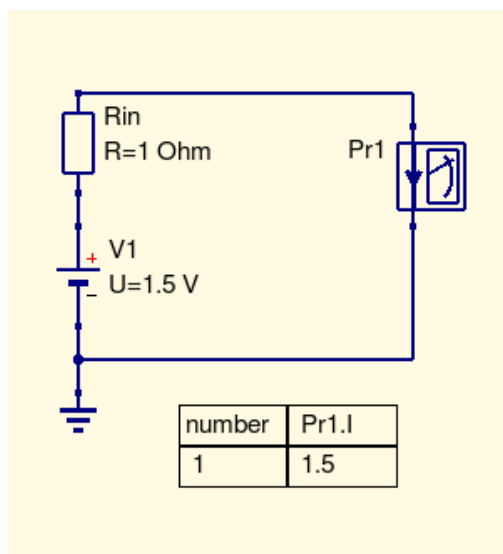


Рис. 3.10. Проверка батарейки «на пригодность»

В этом примере внутреннее сопротивление батарейки 1 Ом, определяющее ток при проверке 1.5 А. По мере старения (разряда) батарейки ее внутреннее сопротивление может возрастать, а возросшее сопротивление изменяет режим работы усилительного каскада. Если подобное происходит в многокаскадном усилителе, например, из-за старения электролитических конденсаторов (высыхания электролита), то это может стать причиной появления паразитной (не учтенной) обратной связи и привести к потере устойчивости усилителя, к самовозбуждению. Явление достаточно распространенное, отчего опытные радиолюбители всегда советуют при ремонте электронных устройств обязательно проверить электролитические конденсаторы после проверки наличия питающего напряжения.

Иногда большое внутреннее сопротивление источника может оказаться полезным. Такой источник называют источником постоянного тока (обычная батарейка — источник постоянного напряжения), что подразумевает, что ток будет мало зависеть от сопротивления нагрузки в достаточно широких пределах. У идеального источника тока полностью не зависит от сопротивления нагрузки. Этот эффект применяют, например, в мультиметрах при измерении сопротивления. Если через сопротивление на выбранном пределе будет протекать ток, не зависящий от величины сопротивления, то напряжение на измеряемом сопротивлении будет прямо пропорционально величине сопротивления и сопротивление, при удачном выборе тока, можно отсчитывать по шкале вольтметра.

Многие начинающие радиолюбители, начиная осваивать работу с транзисторами, стремятся разобраться в физике процессов, происходящих в нем. Это хорошо. Но пройдет довольно много времени, прежде чем полученные знания станут помощниками в практической работе. В этом случае удобно представить себе транзистор, как разновидность резистора, управляемого сигналом, приходящим на его базу. Между током базы и током коллектора простое отношение, а каждое изменение базового тока вызывает пропорциональное изменение коллекторного тока, как бы изменяющего сопротивление резистора по закону входного сигнала. При таком подходе схему, изображенную на рисунке 3.2, можно представить в виде двух резисторов, один из которых R1 — сопротивление нагрузки, а второй R2 — переменный резистор (потенциометр), управляемый источником сигнала, роль которого будем играть мы. Вращая ручку потенциометра по определенному закону, скажем, так, чтобы входной сигнал был синусоидальным, мы будем менять общее сопротивление цепи, меняя ток в цепи, и получим в точке наблюдения output синусоидальное

напряжение. Эта примитивная модель позволяет лучше понять, что транзистор, в сущности, управляет током, полученным от источника питания $V1$, а падение напряжения на резисторе нагрузки $R1$ формирует выходной сигнал.

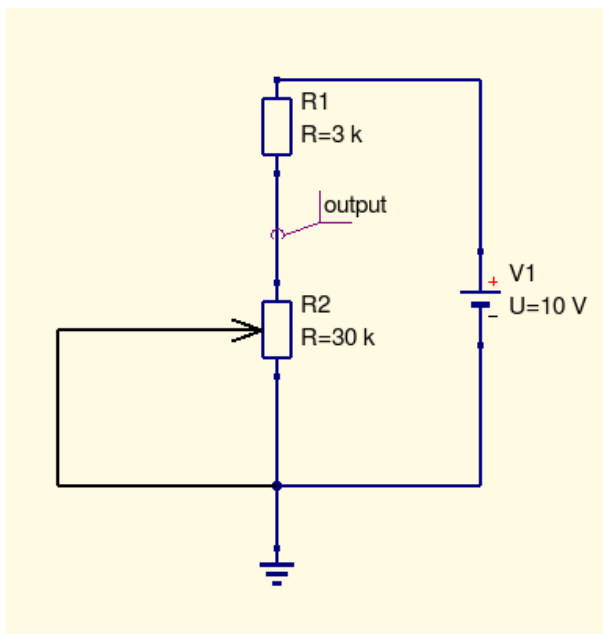


Рис. 3.10. Модель транзистора из двух резисторов

При рассмотрении такой модели яснее становится, отчего амплитуда синусоидального сигнала не может быть больше половины питающего напряжения. Ясно, что максимальный ток, протекающий через транзистор, не может быть больше, чем отношение напряжения источника питания к величине сопротивления нагрузки.

Простая модель замены транзисторов управляемыми резисторами позволяет легко понять, отчего в выходном двухтактном усилителе мощности через нагрузку протекает переменный ток, меняющий не только свою величину, но и направление.

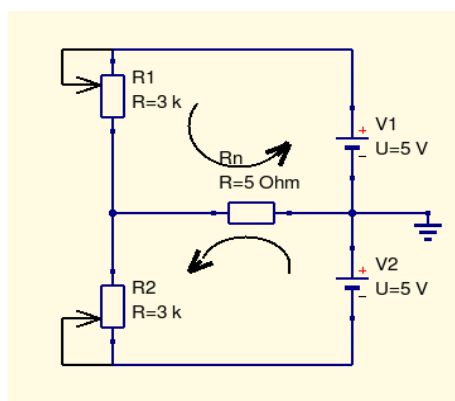


Рис. 3.11. Модель двухтактного выходного каскада

Если положить, что входной сигнал синусоидальный, что при положительной полу-волне управляется резистор $R1$, а при отрицательной $R2$, а в начальный момент при отсутствии сигнала сопротивления резисторов равны, то токи, протекающие через нагрузку взаимно компенсированы, то есть, ток через нагрузку не протекает. Изменение сопротивления $R1$ при положительной полу-волне (сопротивление резистора $R2$ остается равным 3 кОм) такое, что

закон изменения тока повторяет закон изменения входного управляющего сигнала, вызовет на сопротивлении нагрузки R_n падение напряжения под действием верхнего (на рисунке) тока, а во время действия второй полу-волны «работает» второй ток.

Менее очевидной кажется ситуация при питании двухтактного выходного каскада от одного источника питания.

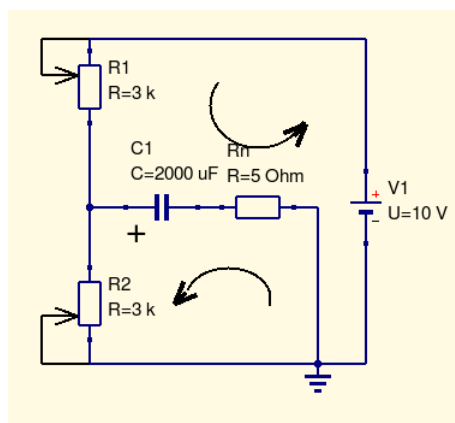


Рис. 3.12. Питание выходного каскада от одного источника

Но не следует забывать, что с изменением сопротивления $R1$ происходит заряд конденсатора $C1$, который во время работы резистора $R2$ разряжается через сопротивление нагрузки. В итоге через нагрузку протекает ток такого же характера, что и в первом случае.

Как и простыми схемами, так и простыми моделями начинающему радиолюбителю не следует пренебрегать — важна не сложность конструкции, а ясность понимания того, что происходит в ней.

Принято разделять аналоговую и цифровую электронику. Что справедливо. Но не следует забывать, что внутри цифровых микросхем транзисторы, работающие, чаще всего, как усилительные каскады. Рассмотрим такую схему:

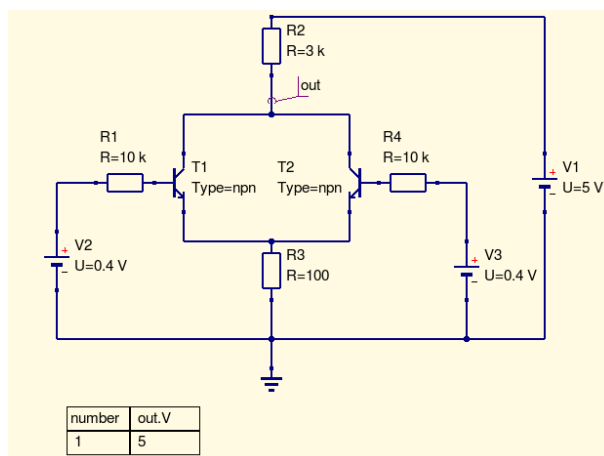


Рис. 3.13. Схема ИЛИ-НЕ при низких уровнях сигнала на входе

Напряжение в точке наблюдения, 5 вольт, соответствует уровню напряжения логической единицы (например, в транзисторно-транзисторной логике, ТТЛ), при этом на входах напряжения ($V2$, $V3$) соответствуют напряжению логического нуля. Проверим, меняя напряжения $V2$ и $V3$, что происходит в точке наблюдения.

В силу симметрии схемы входы (R1 и R2) ведут себя одинаково, нет нужды проверять это, поэтому изменим напряжение на одном из входов до 2.5 В, а затем подадим это напряжение на оба входа.

Результаты эксперимента можно свести в таблицу, которая покажет состояние (логическое) выхода от логических состояний входа. Такие таблицы в цифровой технике называются таблицами истинности.

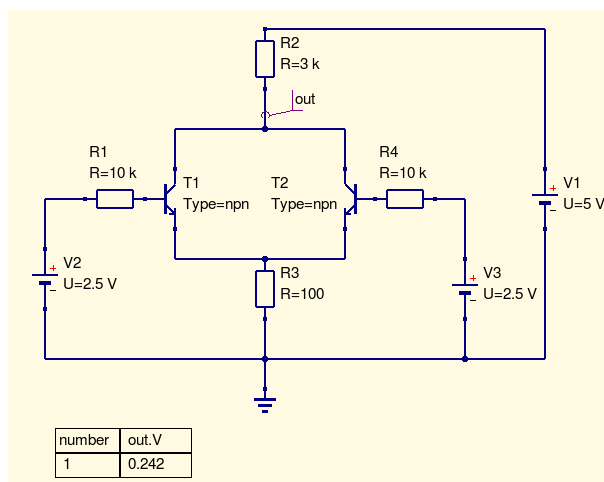
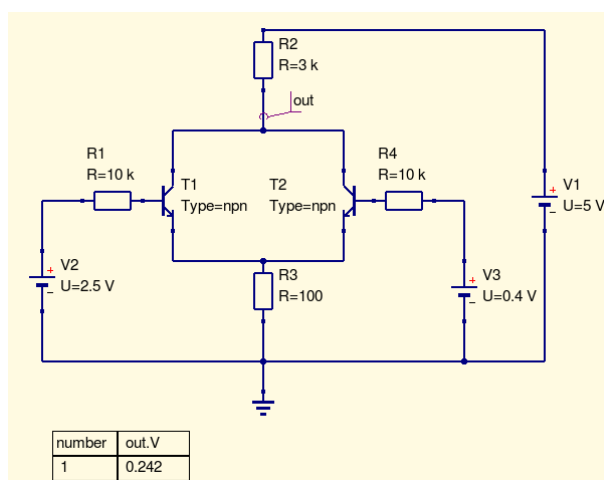


Рис. 3.14. Поведение схемы при подаче «логических» напряжений на входы

Таким образом, напряжение на выходе соответствует уровню напряжения логического нуля при подаче логической единицы ИЛИ на один вход, ИЛИ на другой вход (схема симметрична), ИЛИ на оба входа. Если бы мы инвертировали сигнал, добавив еще один каскад на транзисторе, то на выходе получили бы не логический ноль, а логическую единицу при наличии логической единицы на входах. То есть, получили бы схему ИЛИ.

Вместе с тем, позаботимся о задании начального режима работы схемы, добавив сопротивление, соединим второй вход с землей и подключим сигнал синусоидального напряжения на первый вход через конденсатор.

В результате мы получим обычный усилительный каскад. Это превращение цифровой микросхемы в усилитель возможно не только на схеме. Его вполне можно осуществить на практике. Так широко распространенная микросхема К155ЛА3 может быть превращена в

четыре усилителя с коэффициентом усиления по напряжению в несколько десятков раз. Конечно, это не совсем «штатный» режим работы микросхемы, параметры полученного усилителя не наилучшие, и сегодня, когда в продаже есть множество микросхем усилителей, нужды в таких превращениях нет, но хотелось подчеркнуть, что усилители остаются очень важными компонентами любого электронного устройства.

Начиная знакомство с электроникой, очень полезно познакомиться с работой усилителя, даже в том случае, если вы предполагаете заниматься цифровой техникой или устройствами автоматики. Формально не обязательно знать, как устроена цифровая микросхема или аналого-цифровой преобразователь для работы с цифровыми устройствами, но на практике, чем яснее представляешь поведение устройства, обусловленное его построением, тем лучше понимаешь природу возникающих трудностей или возможных проблем.

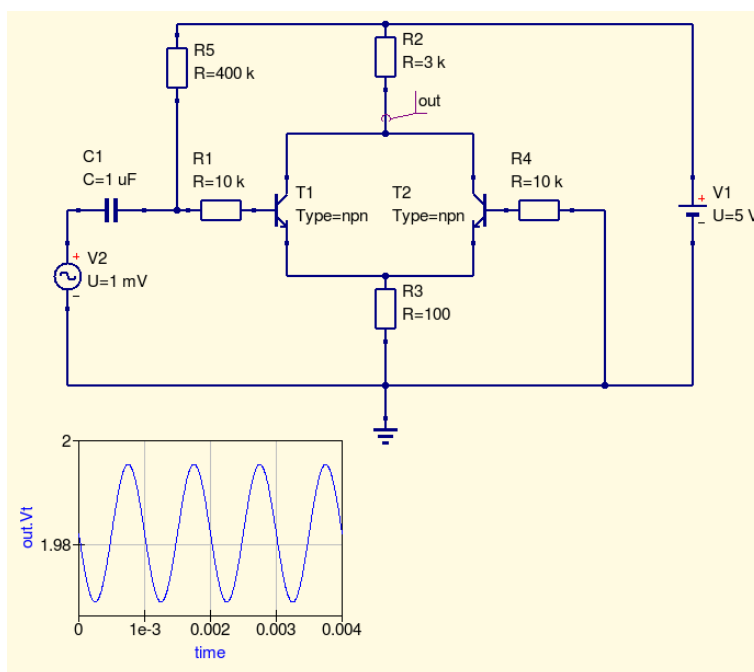


Рис. 3.15. Превращение предыдущей схемы в обычный усилительный каскад

Когда мы говорили об амплитудно-частотной характеристике усилителя, на рисунке 3.6 появлялась АЧХ интегрирующей RC цепочки. Она объясняла природу поведения усилителя на верхней рабочей частоте. В отношении цифровых устройств, где усилительные каскады имеют те же свойства, можно сказать, что сигнал на выходе, после появления напряжения высокого уровня на входе, появится не раньше, чем конденсатор зарядится через резистор, а для цифровой микросхемы время задержки сигнала на выходе играет важную роль. Логика работы схемы может быть правильной, а из-за временных задержек схема может не работать.

Немного о генераторах

Когда мы говорили об усилителях, то я упоминал, что отрицательная обратная связь при определенных условиях может превратиться в положительную и превратить усилитель в генератор. Тогда генератор был «тенью злодея».

Однако генераторы очень нужные и полезные устройства. Они необходимы и в радиотехнике, где они используются для создания несущей частоты радиосигнала, используются в качестве гетеродинов в приемниках; они необходимы, как источники испытательных сигналов при налаживании и проверке устройств; они управляют работой сложных цифровых устройств и служат «сердцем» современных устройств питания бытовой техники.

Самый простой генератор можно получить из колебательного контура, о котором мы говорили, транзистора, о котором мы тоже говорили, и дополнительной катушки, намотанной поверх катушки колебательного контура, и создающей положительную обратную связь в усилительном каскаде, о которых, положительной обратной связи и усилителе, мы тоже говорили.

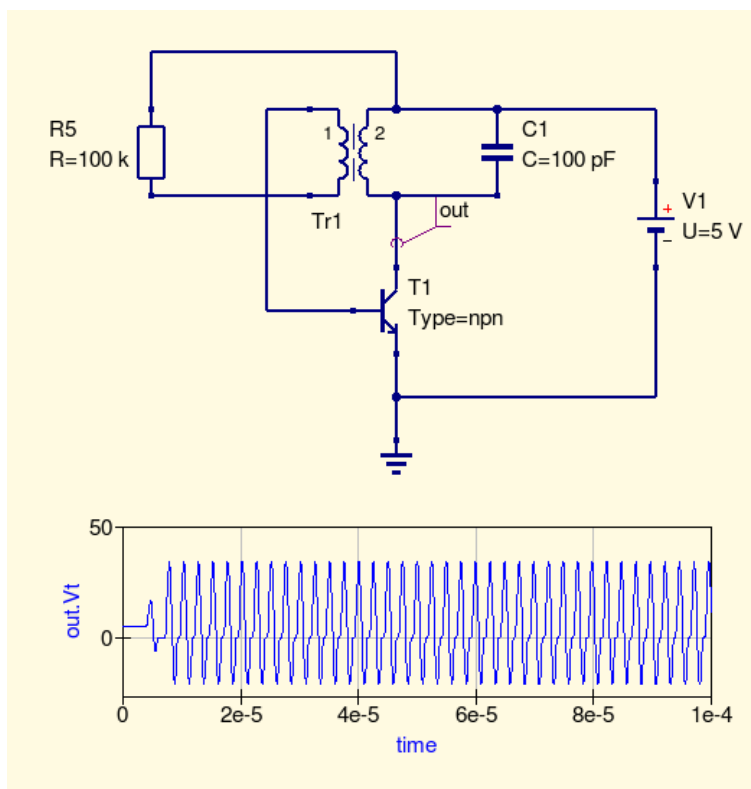


Рис. 4.1. Простейший генератор на транзисторе

Генератор такого типа можно встретить в радиоприемнике, где он играет роль гетеродина — конденсатор C1, вторая секция конденсатора переменной емкости, первая секция служит для настройки на радиостанцию, перестраивается при настройке. Изменение емкости приводит к изменению частоты генерации.

Поскольку при этом изменяется частота настройки, разница между частотой входного сигнала и частотой гетеродина поддерживается постоянной, образуя промежуточную частоту (несущую частоту амплитудно-модулированного сигнала), которая и усиливается далее в

каскадах усиления промежуточной частоты.

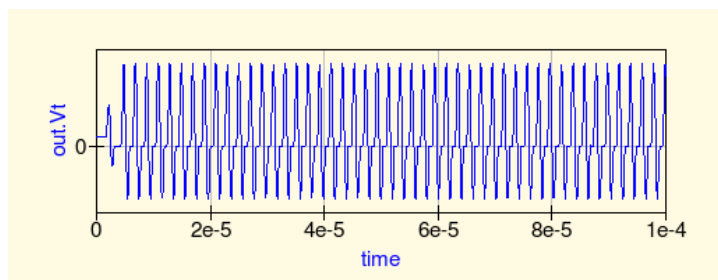


Рис. 4.2. Изменение частоты генерации при $C1 = 50$ пФ

При испытаниях усилителей звуковой частоты используют низкочастотные генераторы. В любительской практике подобные генераторы создают, как правило, на основе схемы с «мостом Вина».

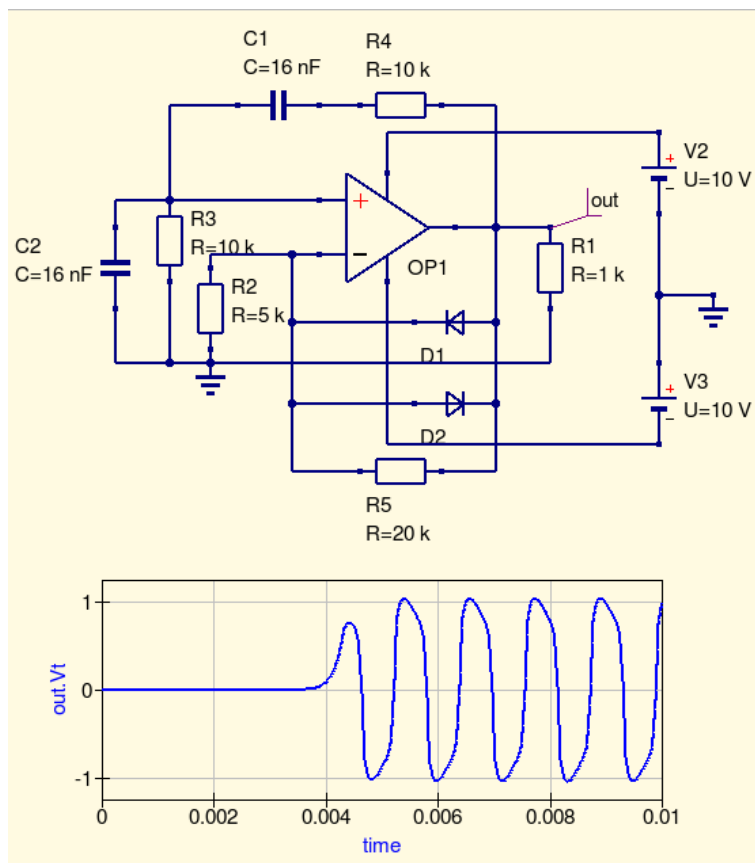


Рис. 4.3. Низкочастотный генератор с мостом Вина

Эта схема очень похожа на ту, что изображена на рисунке 3.7. Положительная (частотно-зависимая) обратная связь с элементами $R4C1R3$ создает предпосылки для возникновения незатухающих колебаний (самовозбуждения усилителя), а добавленные элементы $C2$, $D1$, $D2$ и $R5$ служат для превращения прямоугольных импульсов в синусоидальные (или очень на них похожие). На практике вместо цепи $D1D2R5$ применяют специальные терморезисторы. Сама цепь образует отрицательную обратную связь, а терморезистор ограничивает за счет отрицательной обратной связи амплитуду выходного сигнала. При этом чем больше амплитуда сигнала на выходе, тем сильнее разогревается терморезистор, и тем меньше его сопротивление, а, значит, возрастает глубина отрицательной обратной связи, уменьшается

амплитуда сигнала на выходе. Комбинация положительной и отрицательной обратной связи превращает генератор прямоугольных импульсов в генератор синусоиды.

Замечу, что модель в программе Qucs заработала не сразу, пришлось подбирать элементы схемы, чтобы получить показанный выше результат. Но, если вы думаете, что собирая на макетной плате схему генератора, вы сразу добьетесь желаемого, то, поверьте, это не так. Повторяя готовые схемы, следует быть готовым к тому, что схема не заработает сразу. Можно самому пересчитать схему и выбрать элементы на основе расчета, но и в этом случае, возможно, придется подгонять их «по месту».

Умение рассчитывать схемы и знание теории очень полезны в практике. Достигается это чтением литературы по предмету, экспериментами и практикой. Я редко сталкиваюсь с радиотехникой на практике, хотя сегодня многие устройства используют радиоканал — и телефонные аппараты, и пульты управления и радиомикрофоны. А вспомнил я о радиотехнике в связи с таким часто употребляемым термином по отношению к генератору, как «емкостная трехточка». Термин мне знаком, но существо предмета я забыл давным-давно. И вспомнить, в чем там дело, пришлось, когда я попытался описать высокочастотный генератор. Помогла найденная в Интернете статья Андреевской Т.М. Приведу свое описание, сделанное на основе этой статьи.

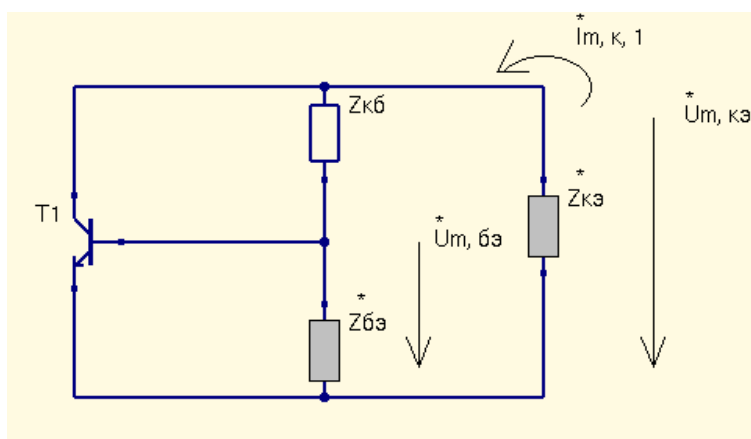


Рис. 4.4. Обобщенная схема (из статьи Андреевской)

Если выделенные заливкой элементы $Z_{кз}$ и $Z_{бэ}$ — емкости, а $Z_{кб}$ — индуктивность (все эти величины комплексные), то схема называется емкостной трехточкой, если наоборот, то индуктивной; и в первом приближении элементы принимаются чисто реактивными: $Z_c = -jX_c$, $Z_L = jX_L$. Здесь $X_c = 1/2\pi fC$, а $X_L = 2\pi fL$. При определенных условиях, баланса амплитуд и фаз на частоте генерации, в схеме возникают колебания.

Схемы подобных генераторов применяют в радиотехнике очень часто. И очень часто при повторении схемы можно столкнуться с тем, что она не хочет работать. Проверить, работает ли схема, даже при наличии осциллографа порой затруднительно — многие осциллографы работают до частоты в 10-20 МГц, а современные радиоустройства чаще используют частоты в 300-900 МГц. Ясное понимание работы схемы в подобных случаях — самый надежный помощник.

Сознаюсь, что до чтения вышеупомянутой статьи я попытался моделировать схему генератора в программе Qucs. Не получилось. И не подумайте, что я не пробовал это в MicroCap, Multisim или Proteus. Прекрасные, к слову программы, но дорогие. Не получилось. И только после чтения статьи, рассчитав элементы схемы, я получил результат с помощью программы Qucs. Из этой истории можно сделать вывод — не следует сразу ругать

программу, она может и не быть виновата в том, что вы не до конца понимаете, что и как следует сделать.

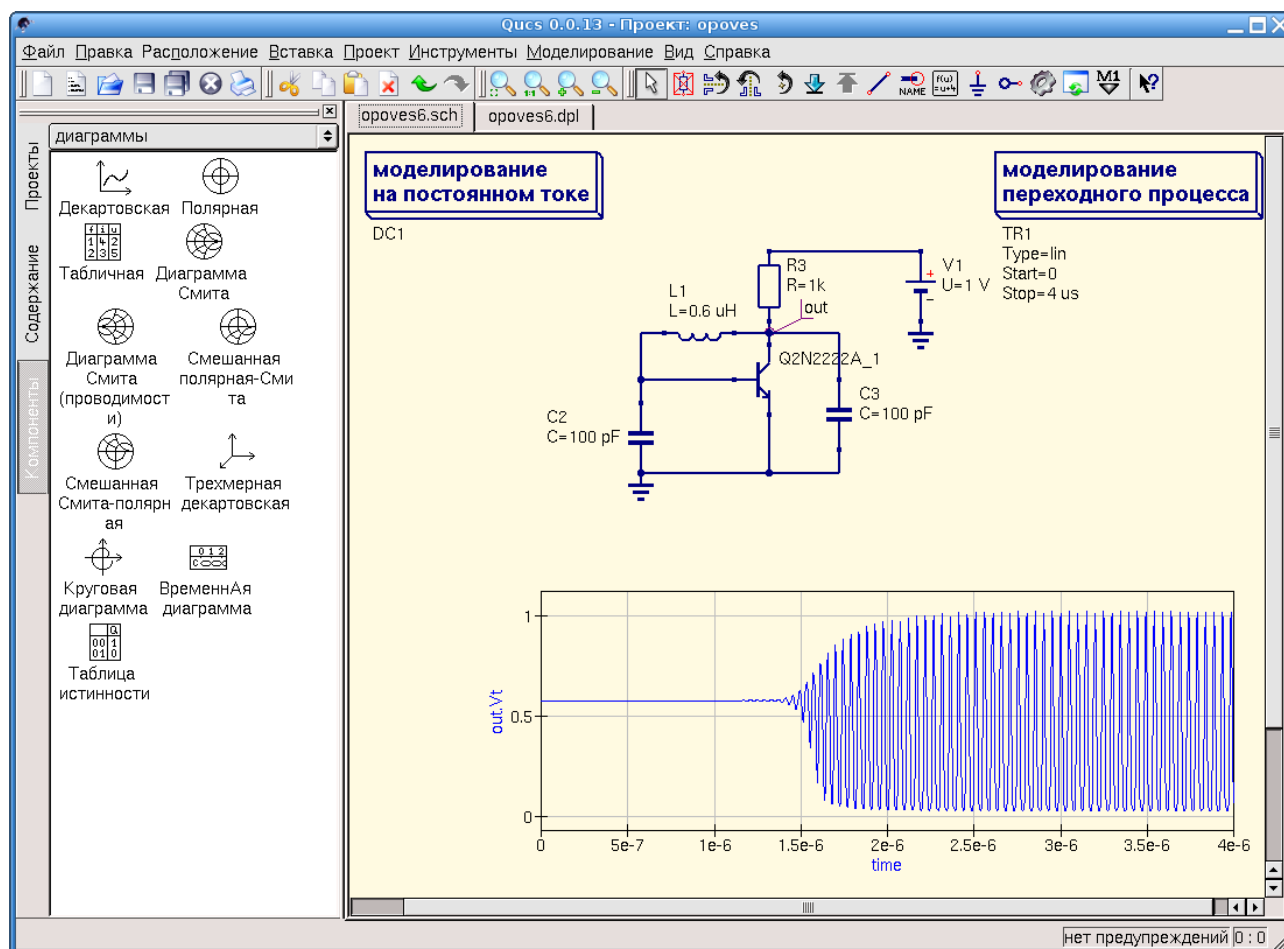


Рис. 4.5. Генератор по схеме емкостной трехточки

Если в радиоустройствах часто применяют генераторы такого типа, используя кварцевый резонатор для стабилизации частоты, то в других устройствах можно встретить, например, мультивибраторы. Основой может быть микросхема таймера 555, цифровая микросхема или схема на транзисторах. Вспомнив о программе Proteus, я вспомнил и о примерах для этой программы, среди которых мне больше всего понравились анимированные схемы. Очень наглядная иллюстрация многих процессов, происходящих в электрических цепях: в полной мере проявляется истина, что лучше один раз увидеть, чем сто раз услышать.

Мультивибраторы могут быть симметричными или нет, но достаточно легко настраиваются и повторяются. Импульсы, которые они генерируют, прямоугольные, скважность импульсов определяется элементами схемы. Симметричные мультивибраторы, обычно, генерируют меандр — прямоугольные импульсы равной длительности (равные времена высокого и низкого уровня).

Схему симметричного мультивибратора на транзисторах из примеров программы Proteus я хочу привести. Обычно при рассмотрении работы такой схемы из-за ее симметрии возникает

вопрос — а почему схема начинает работать? В первую очередь, потому что не бывает абсолютно одинаковых элементов, схема, симметричная на бумаге, в реальном исполнении чуть-чуть не симметрична, по причине чего равновесие нарушается в самом начале работы схемы, один из транзисторов оказывается полностью включен, что приводит к выключению второго транзистора.

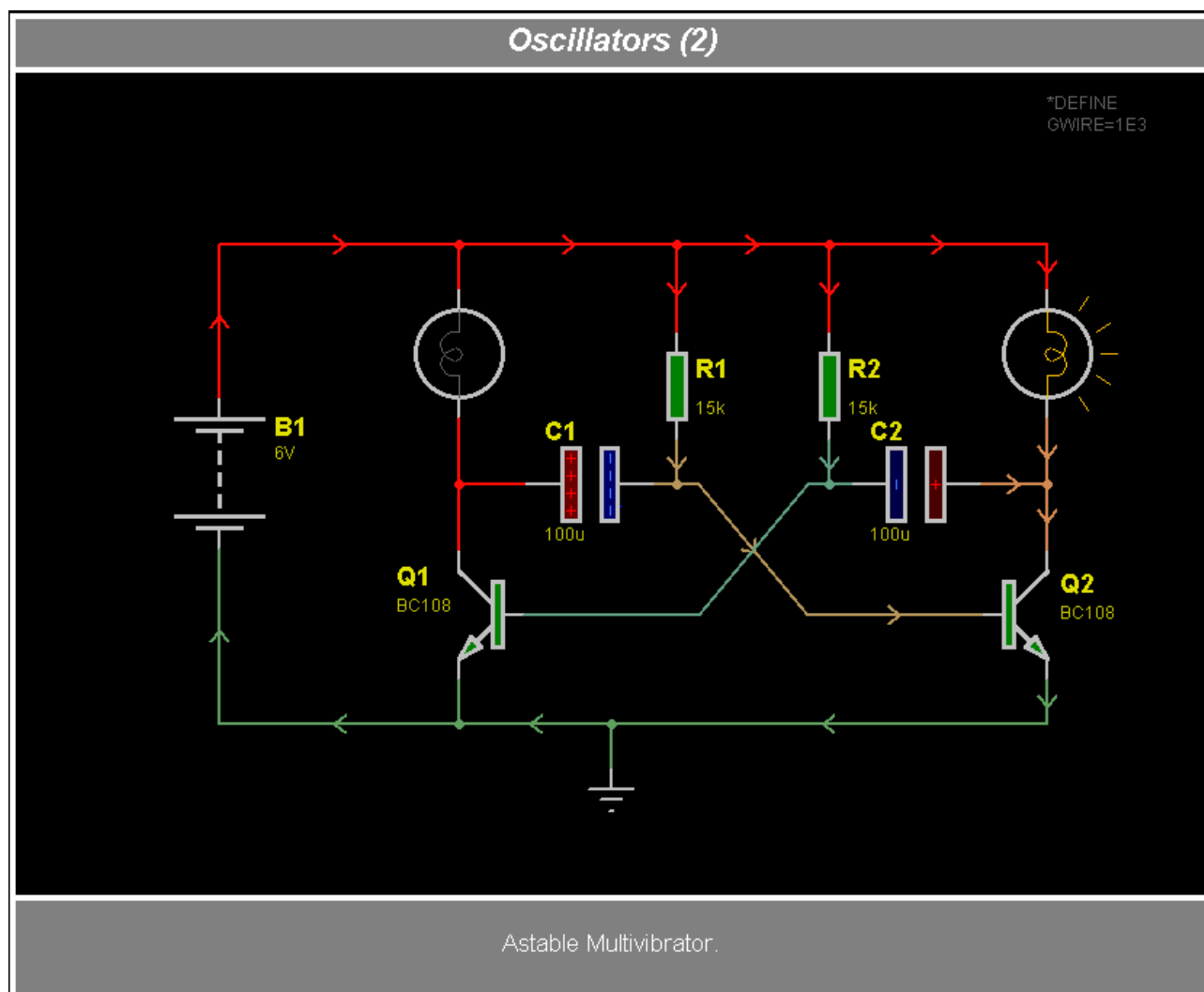


Рис. 4.6. Схема симметричного мультивибратора из примеров Proteus

В этой стадии эмуляции схемы транзистор Q2 полностью включен, а конденсатор C2, правая пластина которого подключена (через транзистор Q2) к эмиттеру транзистора Q1, а левая, отрицательно заряженная, к базе этого транзистора, полностью закрывает транзистор Q1 (напомню, что это транзистор n-p-n, базовый переход которого открывается положительным относительно эмиттера напряжением). Конденсатор C2 через резистор R2 будет перезаряжаться, а когда на его левой пластине образуется положительное напряжение больше 0.5-0.7 В, транзистор Q1 откроется, и теперь уже конденсатор C1 закроет транзистор Q2 полностью. Эти процессы будут повторяться многократно. При равенстве резисторов R1 и R2, конденсаторов C1 и C2 полученные импульсы будут равнодлительны.

И в этом случае можно рассматривать транзисторы Q1 и Q2, как управляемые резисторы, принимающие значения 0 и, скажем, 100 кОм. Когда «резистор» Q1 = 0, конденсатор C1 переключает «резистор» Q2 в состояние 100 кОм, и наоборот.

Однако, если мы перерисуем схему в виде функциональной, не забывая, что транзисторы (R1Q2 и R2Q1) — это усилительные каскады, то мы получим два каскада усиления, охваченные обратной связью, а вспомнив, что выход транзисторного каскада противофазен входу, при двух каскадах обратная связь получается положительной.

Можно посмотреть на схему и с такой точки зрения. Не правда ли?

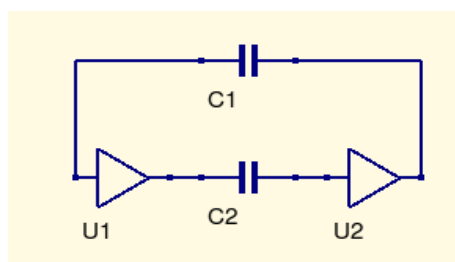


Рис. 4.7. Функциональная схема мультивибратора

И ничто не мешает нам изобразить схему несколько иначе.

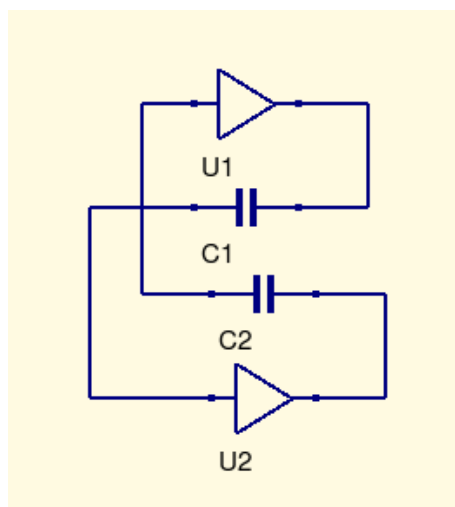


Рис. 4.8. Другой вид функциональной схемы

А, заменив конденсаторы C1 и C2, мы можем получить (почти) схему еще одного часто встречающегося в электронике устройства, которое называется триггером. При подаче импульса на один из входов триггер переходит в состояние, когда один из транзисторов включен, а второй выключен, и сохраняет это состояние до прихода импульса на второй вход.

Триггеры очень часто используются в цифровой технике. Схема RS-триггера на двух вентилях схема И-НЕ выглядит так:

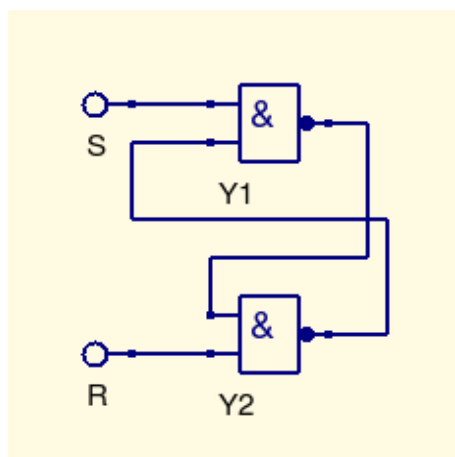


Рис. 4.9. Схема RS-триггера

В цифровой технике RS-триггер изображается в виде функционального цифрового узла с двумя входами S — установка, R — сброс, и двумя выходами Q и \overline{Q} . Временные диаграммы и таблицу истинности этого триггера можно посмотреть в программе Qucs, где Y1 — RS-триггер, а S1 и S2 — генераторы импульсов, работающие в противофазе.

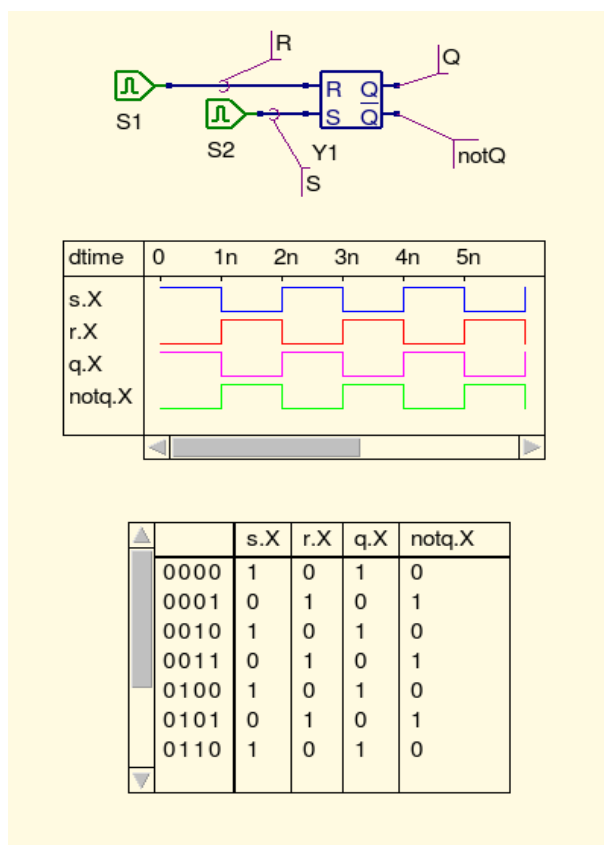


Рис. 4.10. Работа RS-триггера в программе Qucs

Рассматривая временную диаграмму или таблицу истинности, можно заметить, что когда на входе R (сброс) установлен логический ноль (в данном случае низкий уровень напряжения), а на входе S (установка) появляется логическая единица, то выход Q принимает значение логической единицы, а выход \overline{Q} (метка notQ) логического нуля. И наоборот.

Чтобы не быть голословным в отношении триггера, построенного на транзисторах, приведу схему, в которой транзисторы T1 и T2 сохраняют свое состояние до прихода управляющего импульса от генератора V2. Справа диаграмма этого управляющего импульса (метка input). Диаграммы выходов out1 и out2 показывают переход триггера из одного состояния в другое, которое триггер сохранит до прихода импульса на второй вход, если на первом импульса не будет. Состояние, когда на оба входа приходят управляющие импульсы, скорее всего, будет не определенным. Во всяком случае для идеального устройства, в реальных устройствах оно может измениться из-за разности в длительности управляющих импульсов или может сохранять предыдущие значения выходов, как сделано в некоторых сериях цифровых микросхем.

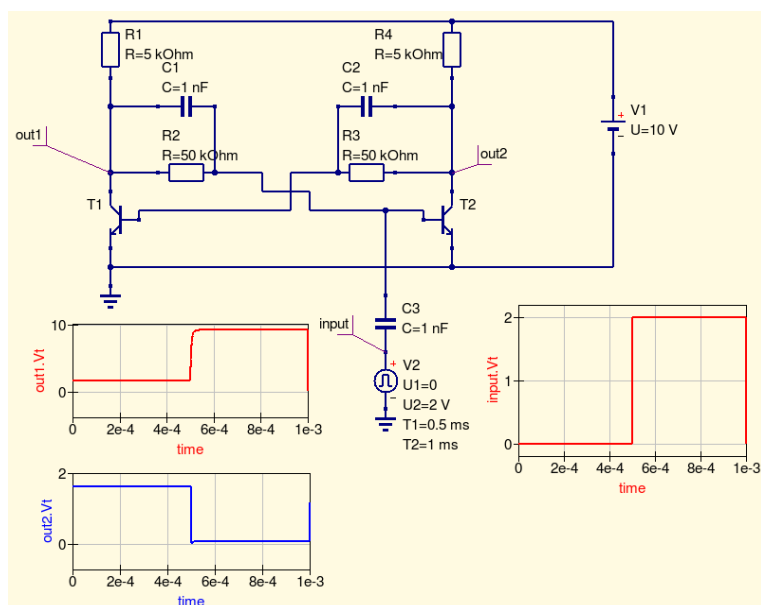


Рис. 4.11. Схема триггера на транзисторах

В цифровых схемах часто, там где не требуется очень стабильная частота, строят генераторы, используя цифровые вентили, например, инверторы. На выходе такого генератора прямоугольные импульсы имеют не лучшую форму. Чтобы привести форму импульсов к виду, достойному названия прямоугольные, используют триггер на выходе генератора. Форма импульсов значительно улучшается.

«Классическая» схема синхрогенератора на цифровых микросхемах:

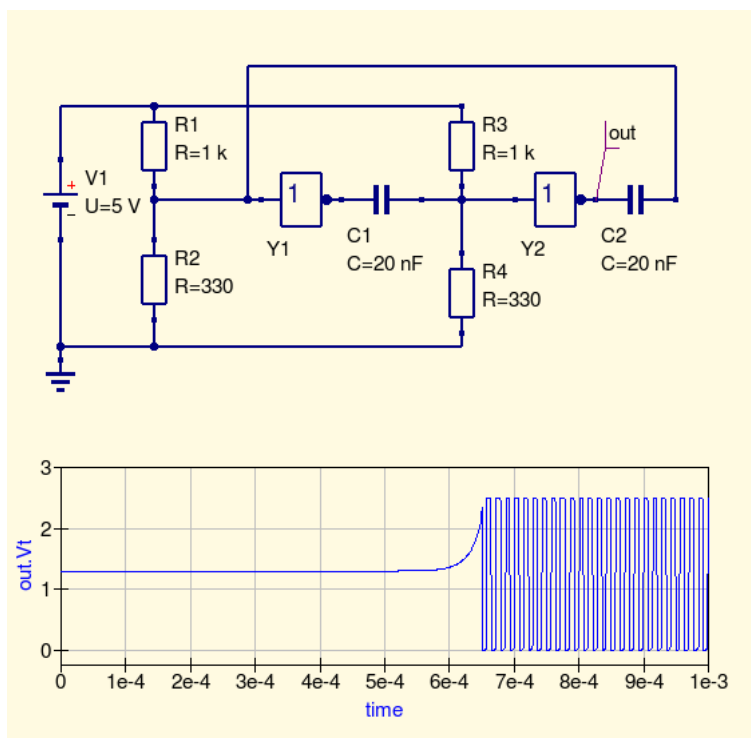


Рис. 4.12. Синхрогенератор

Заметьте, что резисторы R1 и R2 (R3 и R4) переводят цифровой вентиль с точки зрения цифровой техники в неопределенное состояние, устанавливая на входе напряжение порядка 1.5 В и превращая вентиль в усилитель. И еще, обычно считается, что симуляция цифровых устройств в программах EDA происходит только на уровне логики, но в данном случае с точки зрения логики, мне кажется, генератор не должен существовать вовсе. То есть, программа работает, все-таки, ближе к физическому уровню схемы. Выход второго вентиля (на его выходе метка наблюдения out) в начальный момент времени находится на уровне (неопределенном в логике) 1.2 В, а затем начинается процесс колебаний между 0 и 2.5 В.

Качество программ, в части их сближения с реальными устройствами, из года в год улучшается. И им можно в большей мере доверять, особенно это касается программ для профессионального использования.

Вот, как выглядит сигнал на входе первого вентиля:

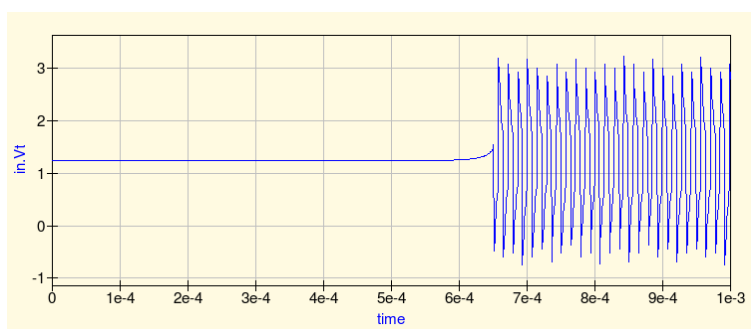


Рис. 4.13. Сигнал на входе вентиля

И, поскольку речь зашла о цифровой технике, пора немного познакомиться с ней. Но прежде мне хочется привести еще одну схему.

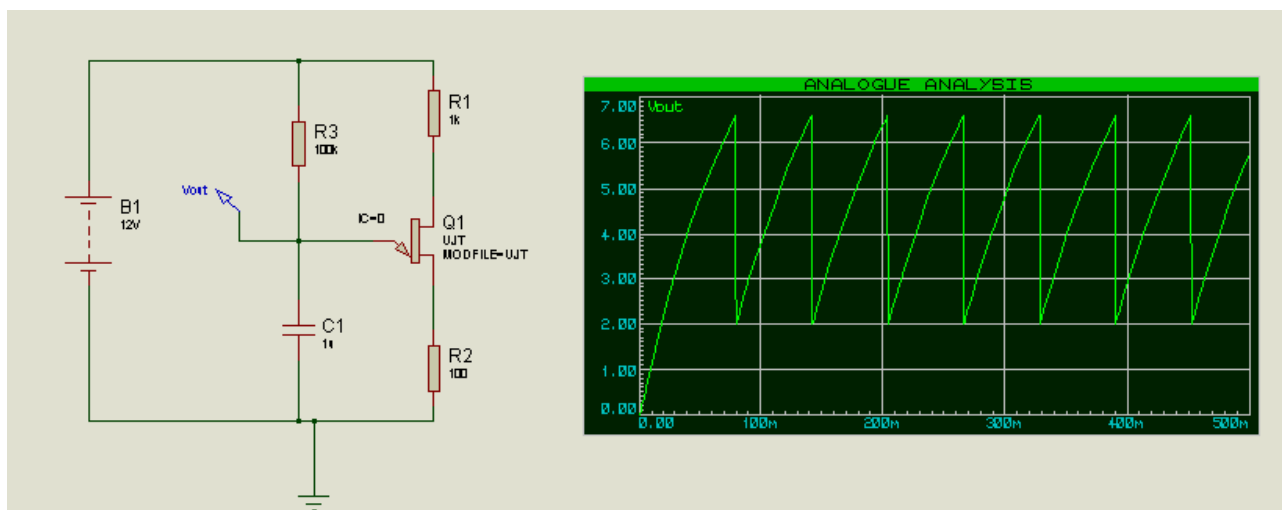


Рис. 4.14. Генератор на однопереходном транзисторе

Это схема тоже из примеров программы Proteus. А мне хотелось показать многообразие схем генераторов, которые находят применение в электронике. У каждой из схем есть своя область применения, есть свои преимущества и недостатки. Не думаю, что можно найти наилучшую схему, но всегда можно найти самую подходящую для вашей цели. Нужно только поискать.

Немного о цифровой технике

Появлению большого количества разновидностей цифровых микросхем предшествовала упорная работа по созданию цифровых вычислительных машин. Вычисления, расчеты — это то, для чего создавался ваш компьютер.

Любое число можно использовать в разных представлениях: двоичном, восьмеричном, десятичном, шестнадцатеричном и т.д. При этом по виду числа даже не скажешь, что это за число, например, 100 — это может быть десятичное число сто, а может быть двоичное 4 или шестнадцатеричное 256. Мы привыкли к числам с основанием 10, но для цифровых вычислительных машин оказалось удобнее использовать числа с основанием 2. То есть, числа, записываемые с помощью двух цифр 0 и 1. С точки зрения электроники этим числам можно сопоставить два напряжения: низкого уровня, скажем 0, и высокого, например, 5 В. Эти два напряжения прекрасно распознаются электронными компонентами, трудно ошибиться, что на входе или выходе устройства ноль или единица. А все операции с числами, вне зависимости от основания, остаются такими же: сложение, вычитание, умножение, деление... Мало того, в математике есть раздел математической логики, который оперирует с двумя понятиями «истинно» и «ложно». А это столь похоже на 0 и 1, что для логических операций можно использовать цифровые микросхемы. Эти понятия — вычисление и логические операции так давно смешались в области цифровой техники, что таблица состояний выходов устройства в зависимости от состояния входов давно называется таблицей «истинности», а о работе цифровых схем говорят, как о логике работы схемы.

Математически можно показать, что с помощью схем И, ИЛИ и инвертора НЕ, можно построить все то многообразие цифровых устройств, включая арифметико-логические, которые сегодня есть в каждой серии цифровых микросхем. Работа этих простейших составляющих, их еще называют вентилями, описывается с помощью таблиц истинности. Посмотрим, что это за таблицы на простых примерах.

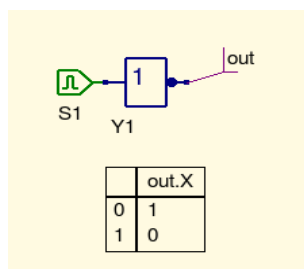


Рис. 5.1. Таблица истинности для инвертора

На рисунке: S1 — генератор цифровых сигналов (вообще-то прямоугольных импульсов с амплитудой логической единицы), Y1 — инвертор, имеющий один вход и один выход (с меткой out), а в таблице показано, какое (логическое) значение принимает выход инвертора, когда на его входе «0» и «1». Для инвертора можно встретить и другое графическое обозначение:

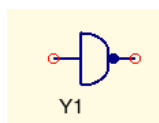


Рис. 5.2. Другое изображение инвертора

Таблицы истинности для схем И и ИЛИ, и их графическое изображение:

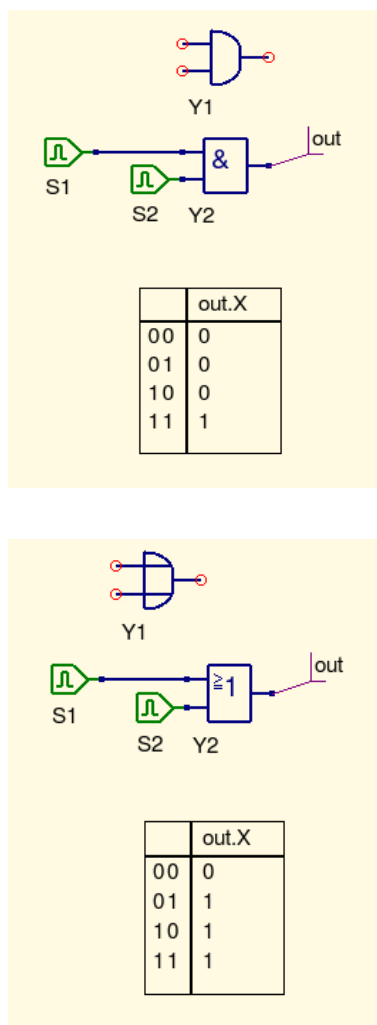


Рис. 5.3. Таблицы истинности и графика вентилях И и ИЛИ

Из таблиц истинности понятно, что для схемы И значение логической единицы на выходе появляется только тогда, когда на всех входах (в данном случае двух входах, а их может быть больше) появляются логические единицы. А для схемы ИЛИ достаточно единицы на любом из входов (01 в таблице означает, что на первом входе 0, на втором единица).

Хороший курс лекций по цифровой технике можно найти в Интернете по адресу (надеюсь, этот ресурс останется в Интернете долго):

<http://www.bashedu.ru/wsap/rusvsapelnikov.html>

А хорошее описание работы цифровых микросхем в книге В.Л. Шило «Популярные цифровые микросхемы».

Если не пытаться построить из набора микросхем компьютер, то очень полезными для практического применения оказываются микросхемы триггеров. Триггеры-защелки позволяют запомнить данные (в какой-то момент времени), чтобы позже прочитать эти данные. Посмотрим, как это можно сделать с помощью D-триггера. Чуть позже я хотел бы рассказать о счетчиках. В программе Qucs нет готовых моделей цифровых счетчиков, но это легко можно восполнить с помощью тех же D-триггеров.

Итак, как D-триггер может запомнить данные?

В схеме добавлен генератор S2, который дает один импульс сброса, чтобы состояние триггера стало определенным до начала работы устройства. Если посмотреть на диаграмму, то видно, что частота на выходе D-триггера вдвое ниже частоты сигнала синхрогенератора. Соединяя выход D-триггера со входом синхронизации следующего триггера, включенного аналогично, и создав цепочку из четырех триггеров, мы получим конструкцию, которая будет соответствовать двоичному счетчику. Счетчик будет на четырех выходах отображать число полученных от синхрогенератора импульсов в двоичном представлении. И каждый выход станет источником сигнала, частота которого вдвое ниже предыдущего.

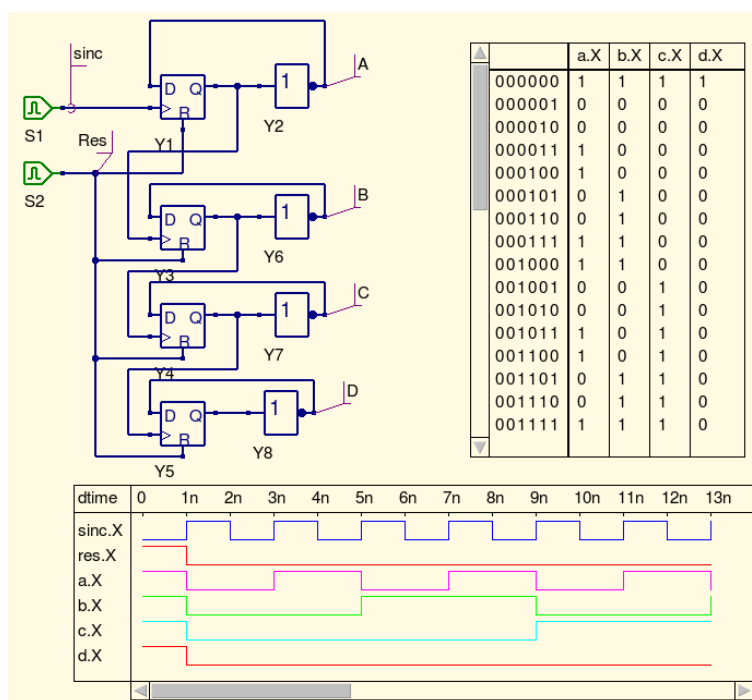


Рис. 5.6. Двоичный счетчик на основе D-триггеров

В каждой из серий цифровых микросхем можно найти много модификаций счетчиков. Подсчет числа импульсов может быть полезен в тех случаях, когда вам нужно знать число каких-либо событий, если события можно представить (с помощью датчика) импульсами. Скажем, такой счетчик может стать функциональным узлом автоответчика — счетчик будет подсчитывать количество звонков и после третьего (или пятого) звонка включаться, если трубка не была снята с аппарата.

Используя счетчик, можно быстро сделать простой генератор испытательных сигналов. Выбрав частоту синхрогенератора 6 кГц, получим сигналы с частотой 3 кГц, 1,5 кГц и т.д. И, хотя сигнал не будет синусоидальным, с помощью такого генератора можно проверить и оценить работу усилителя низкой частоты.

Предположим, что можем включать цифровой счетчик и останавливать его, когда нам потребуется, а подсчитанное количество импульсов от синхрогенератора будет запоминаться. Включая счетчик с приходом исследуемого прямоугольного импульса и останавливая его работу, когда импульс меняет высокий уровень на низкий, мы получим количество эталонных (от синхрогенератора, частота которого нам известна) импульсов, укладываемых в длительность высокого состояния исследуемого импульса, а, значит, можем определить временную длительность этого импульса, или для периодических импульсов определить длительность периода, то есть, частоту следования импульсов — на основе счетчика мы получили частотомер.

Описание многообразия цифровых схем и всех аспектов из работы заняло бы ни одну книгу, как оно и есть в действительности, я уверен, что уже написанные книги не утратили своего значения — нет смысла повторять написанное в них. Однако хотелось бы упомянуть об одном электронном устройстве, которое в последние десятилетия приобретает все большую значимость в силу все большего распространения. Если в вашем мобильном телефоне есть встроенный диктофон, то ваш мобильный телефон имеет аналого-цифровой преобразователь (АЦП или ADC).

Идея, заложенная в работу этого устройства, проста — производя измерения амплитуды сигнала (переменного напряжения) в каждое мгновение, мы получим таблицу чисел, по которой сможем впоследствии восстановить исходный сигнал. Единственный недостаток — количество чисел в таблице будет бесконечно. Но... на помощь приходит математика. Есть теорема, показывающая, что для сохранения информации при таком преобразовании достаточно конечного числа измерений.

Я не думаю, что любителю есть смысл самостоятельно делать АЦП, достаточно готовых микросхем, чтобы удовлетворить все практические нужды, но, мне кажется, что интересно было бы рассмотреть некоторые схемы, позволяющие понять, как можно осуществить, хотя бы в принципе, подобное преобразование. Сначала вернемся на некоторое время к аналоговым устройствам.

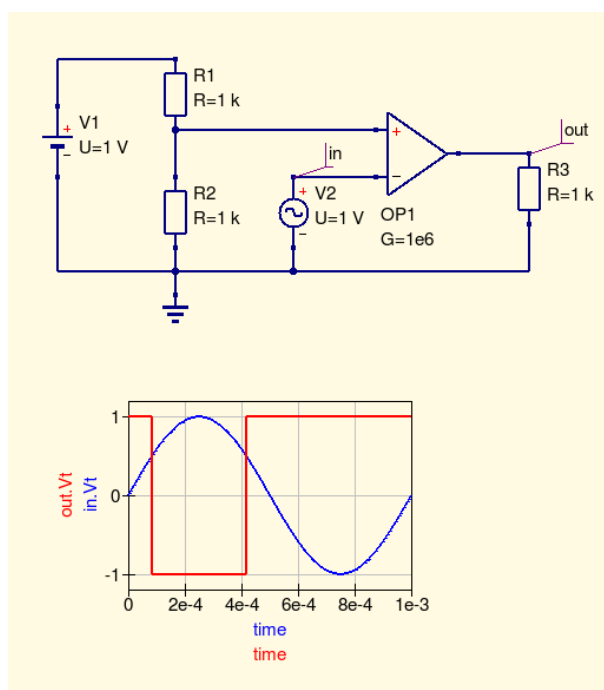


Рис. 5.7. Принцип работы компаратора

Компаратор — сравнитель, достаточно часто применяется на практике. На рисунке изображено устройство, скорее поясняющее работу компаратора, чем схема для повторения. На один из входов операционного усилителя OP1 (прямой вход) подано постоянное напряжение, снимаемое с делителя $R1R2$. Благодаря этому напряжению в начальный момент времени на выходе операционного усилителя максимальное напряжение, близкое к напряжению питания (в данном случае напряжение питания выбрано $\pm 1\text{ V}$).

На второй вход операционного усилителя (инверсный) подан исследуемый сигнал от источника $V2$ (синусоидальное напряжение 1 кГц , амплитуда 1 В). Пока амплитуда синусоидального сигнала не превысит опорного напряжения 0.5 В , снимаемого с делителя

R1R2, выходное напряжение сохраняется. Но, как только амплитуда сигнала от генератора V2 превысит это напряжение, хотя бы на несколько милливольт, напряжение изменится — для операционного усилителя с коэффициентом усиления в миллион ($G=1e6$ на рисунке) при умножении 1 мВ мы получим значение большее напряжения питания, а полярность входного напряжения изменилась на противоположную. Как видно из диаграммы, напряжение на выходе операционного усилителя резко принимает максимальное отрицательное значение.

Вот такой переход (фронт полученного прямоугольного импульса на выходе ОУ) мы можем использовать. Мы уже говорили, что счетчик может измерять промежутки времени. Предположим, что мы запустили счетчик в начальный момент времени, а фронтом импульса от ОУ остановим работу счетчика. Мы можем прочитать и запомнить на выходе счетчика полученное число. Будем считать его амплитудой исследуемого сигнала, равной 0.5 В.

Производя аналогичные действия требуемое количество раз, мы получим нужную нам таблицу чисел, произведя «оцифровку» аналогового сигнала. По этой таблице устройство обратного преобразования ЦАП (цифро-аналоговый преобразователь) воспроизведет исходный сигнал. Что и происходит в CD-проигрывателе, когда он считывает зашифрованные на диске числа и подает их на ЦАП, а мы слушаем через громкоговоритель музыку.

Есть разные методы построения АЦП и ЦАП, есть множество микросхем, выполняющих эту работу, и с каждым днем все чаще этот принцип используется в современной электронике — музыкальные проигрыватели, телефонные аппараты, измерительные приборы и датчики, бытовая техника и промышленное оборудование... Много, где сегодня применяют АЦП. Но вернемся к цифровой технике.

Мне хотелось бы, прежде чем переходить к следующей теме, привести один пример, который я нашел где-то в книге о микропроцессорах.

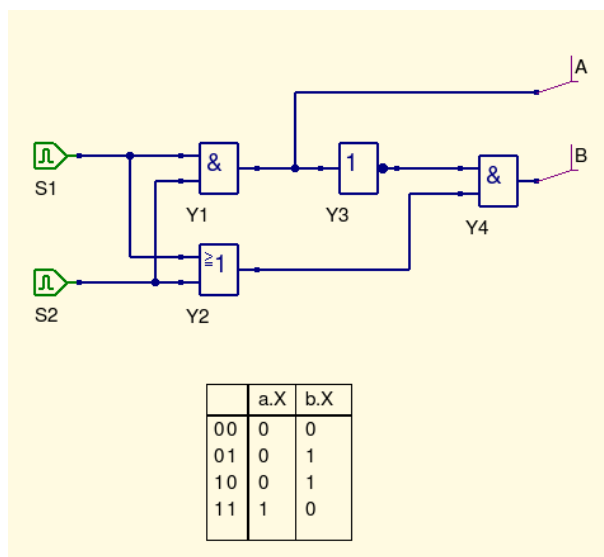


Рис. 5.8. Сложение двух чисел

Схема складывает два двоичных числа, которые генерируют два источника S1 и S2. Результат сложения появляется на выходах АВ. Из таблицы истинности видно, что когда на единица только на одном из входов, на выходе получается результат в виде 01, и только при двух единицах на входах на выходе 10 — двоичное представление числа 2. В устройстве применены простые вентили И, ИЛИ, НЕ. На основе такой схемы можно построить микросхему, которая складывала бы числа наперед заданной разрядности. Аналогично строятся схемы для вычитания, умножения и деления чисел. А на их основе создают

арифметико-логические устройства (АЛУ), которые становятся сердцем микропроцессоров.

Об одной разновидности таких устройств, микроконтроллерах, пойдет речь дальше.

А что такое микроконтроллер?

Можно начать рассказ о микроконтроллерах с рассмотрения архитектуры контроллеров разных производителей. Но я не очень люблю, хотя и понимаю важность терминов, терминологию, особенно заимствованную из других областей знаний.

Можно начать рассказ о микроконтроллерах с рассказа об устройстве микропроцессоров. Но для такого рассказа недостаточно одной главы.

А можно начать с рассказа о языках программирования и программах для работы с микроконтроллерами. Без этих составляющих работа с микроконтроллерами возможна только при условии, что микроконтроллер уже запрограммирован и его нужно только вписать в готовую схему.

Но и рассказ о языках программирования, а для написания кода программ используется не один язык, с учетом особенностей разных компиляторов займет слишком много места. Да и книг и по архитектуре микроконтроллеров, и о микропроцессорах, а, тем более, по программированию написано достаточно.

Однажды, замечу, что я не программист и не хочу им казаться, однажды я обнаружил, что в программе KTechlab, она существует только для Linux, можно написать программу для микроконтроллера без знания какого-либо языка программирования. И из этой же программы запрограммировать микроконтроллер, используя самодельный программатор. Мне показалось, что подобный подход самый удобный для начинающих любителей. Пока не возникнут серьезные сомнения в правильности этой точки зрения, я буду придерживаться позиции — чем проще, тем лучше. Поскольку многие любители покупают компьютер с установленной на нем операционной системой Windows, я начну рассказ, используя программу FlowCode. Эта программа работает в Linux с помощью Wine, по крайней мере версии 1.0. Одним из достоинств программы — наличие версии и для PIC, и для AVR контроллеров. Мало того, можно импортировать программы, написанные для AVR в программу для PIC контроллеров. И, думаю, все, что написано ниже для PIC контроллеров, вполне можно применить для AVR контроллеров.

Основная особенность микроконтроллеров в написании программы, без которой микроконтроллер, купленный в магазине, остается беспомощен. Можно, правда, используя программу для работы с программатором, который тоже может быть куплен в магазине, «зашить» в него готовый загружаемый файл, благо готовых решений в Интернете можно найти множество. Но при таком подходе, когда микроконтроллер перестает быть беспомощным, беспомощным оказывается радиолюбитель.

Преодолеть эту беспомощность я предлагаю при посредстве программы FlowCode.

Ни одна из программ, когда либо написанных, не начиналась с написания кода программы на каком-либо языке или без использования языка программирования. Любая программа начинается с того, что автор старается понять, что и как должна делать программа. Кто-то мысленно и быстро проделывает эту работу, кто-то без спешки проделывает это на бумаге, многие сегодня делают это за компьютером.

Если не вдаваться в детали, то результатом подобных размышлений, в том или ином виде, будет алгоритм работы программы. Для записи алгоритма давно создан специальный графический язык, подобный языку записи электрических схем. Как и электрическая схема, соединяющая элементы цепи, так и эта программная схема соединяет программные элементы. Вот почему я считаю такой подход легче должен восприниматься радиолюбителями. Они давно привыкли, что многие функции в электрических схемах

осуществляют микросхемы — на бумаге квадратики, связанные с другими квадратиками, не более того. И программа для микроконтроллера не отличается во многом от привычного вида схемы.

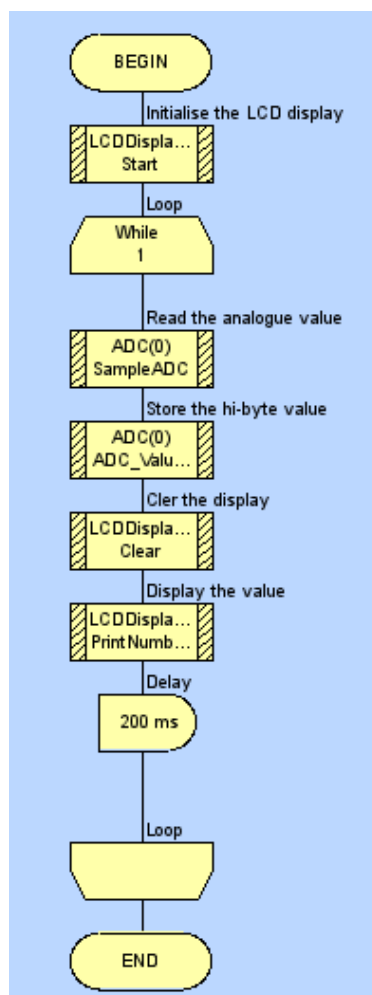


Рис. 6.1. Вид программы для микроконтроллера

Я уже делал попытки рассказать, как работать с этой программой, но впоследствии убедился, что многие вещи столь очевидные для меня, оказываются камнем преткновения для начинающих. Попробую сделать еще одну попытку, и пусть читающих этот рассказ не обидит отсутствие тех подробностей, которые они ищут (я их мог «проморгать»), и не обидит наличие тех подробностей, которые им очевидны (потерпите).

Прежде, чем перейти к рассказу о программе, я хочу сказать, что, скорее всего, за рамками этой истории окажутся такие аспекты, как работа с прерываниями, встроенными АЦП и USART, компараторами и таймерами, RF-модулями. Я уверен, что, начав освоение микроконтроллера, начинающий любитель не остановится на достигнутом и найдет, как использовать все возможности контроллеров.

Но сейчас мы будем смотреть на микроконтроллер, как на маленький компьютер с процессором внутри, огороженном устройствами, которые называются портами ввода-вывода. Когда мы включаем компьютер, мы мало думаем о том, что там внутри, и как оно устроено. Не будем этого делать и в отношении микроконтроллера. Для нас важно, что выводы портов могут по нашему желанию (это достигается при написании программы) служить цифровыми входами или выходами. На цифровые входы мы можем подать уровни

логического нуля или логической единицы, по нашему желанию (программному) на выходах появятся уровни логических нулей или логических единиц. Даже при таком, казалось бы, скромном использовании микроконтроллера можно создать много интересных схем.

Установка программы FlowCode в демо-версии сводится к обычным операциям установки любой программы в Windows, а в Linux при установленной программе Wine потребует либо точно таких же шагов, либо, скажем, запуска установки с помощью менеджера файлов Windows, это зависит от Linux дистрибутива. Если вы предполагаете использовать демо-версию, то на этом установка закончена, если используете какой-либо из вариантов регистрации, то соответствующие шаги будут описаны в приложении. Что можно увидеть при первом запуске программы?

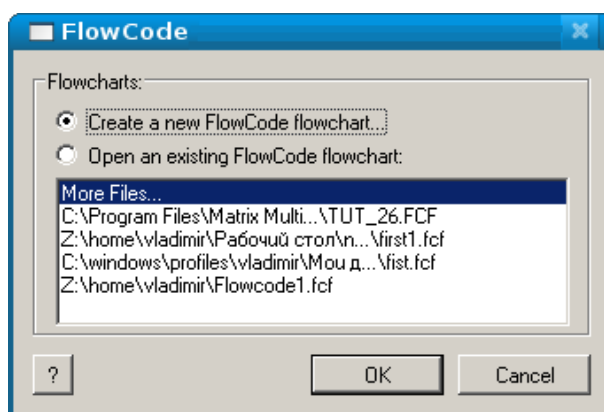


Рис.6.2. Диалоговое окно выбора проекта в FlowCode

В этом диалоговом окне можно создать новый проект (*Create a new FlowCode flowchart...*) или открыть уже существующий, выбрав его из списка ниже, или, используя опцию *Open an existing FlowCode flowchart*, открыть уже существующий проект для его модификации или использования.

Если вы выбираете новый проект, то в следующем диалоговом окне вам будет предложено выбрать модель микропроцессора из списка доступных.

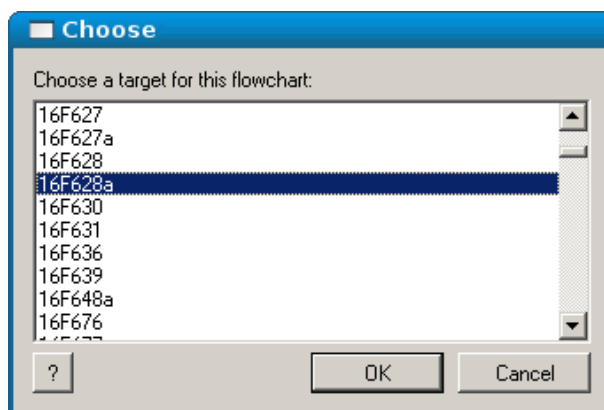


Рис. 6.3. Диалоговое окно выбора модели контроллера

Щелкнув по выбранной модели и кнопке **ОК**, вы попадаете в рабочее окно программы.

Естественно, что всю центральную область занимает графический редактор, в котором будет «написана» программа. Естественно, что интерфейс программы достаточно стандартен для всех программ Windows, и, кстати, для многих программ Linux, если вы никогда не

видели этой операционной системы. Основное меню в верхней части окна, инструментальная панель работы с файлами чуть ниже и две инструментальных панели в левой части рабочего окна программы. Обе эти панели можно сместить в рабочую область окна, если вам это понравится больше, а я их сместил, чтобы иметь возможность рассказать о компонентах этих панелей. Закончив рассказ, верну их на место. В правом верхнем углу, заметьте, появляется внешний вид выбранной микросхемы с обозначением выводов, что бывает весьма полезно при работе с программой.

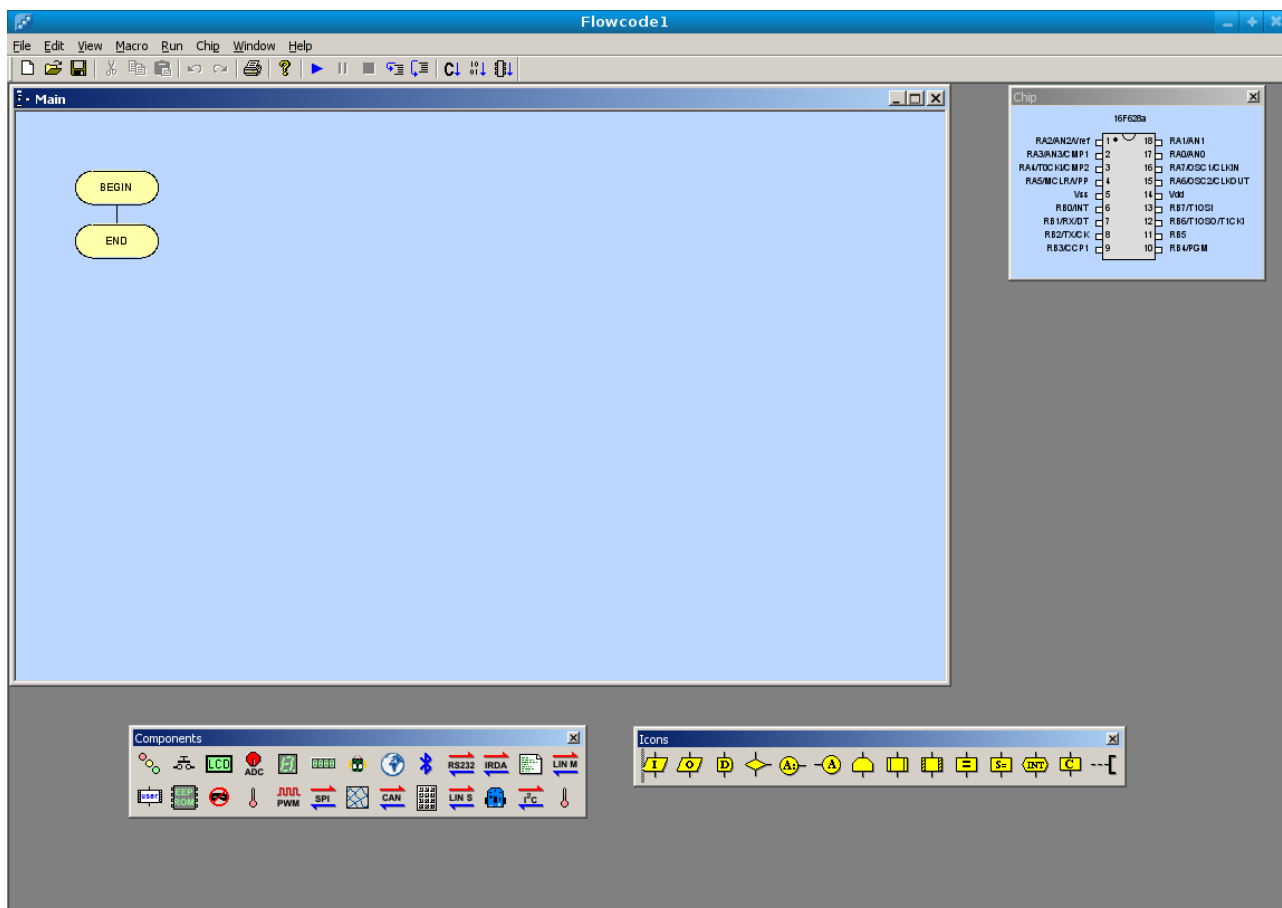


Рис. 6.4. Рабочее окно программы FlowCode

В окне графического редактора программы уже есть заготовка будущей программы: BEGIN-END. Если не ошибаюсь, в языке Pascal такая конструкция называется операторными скобками, в которую заключается и вся программа, и процедуры и функции языка. Позже подобную конструкцию мы увидим в цикле, но позже, а сейчас обратимся к основному меню:

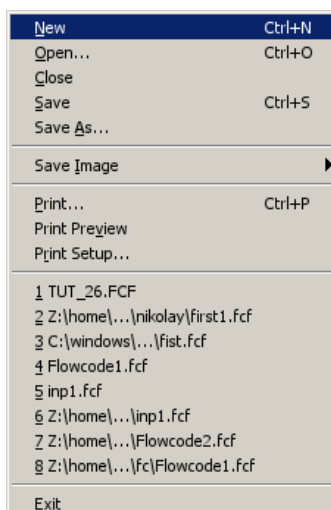


Рис. 6.5. Меню File основного меню

Стандартный набор операций с файлами: *New* — создать новый файл; *Open...* — открыть уже существующий; *Close* — закрыть файл; *Save* — сохранить файл; *Save As...* - сохранить как..., удобно на тот случай, если вы хотите сохранить файл под другим именем или в другой папке; следующий пункт выпадающего меню *Save Image*, позволяет сохранить вам программу в виде картинки в двух графических форматах, которые вы выбираете в открывающемся меню, если курсор наведен на этот пункт; пункты меню, начиная с *Print...*, относятся к выводу на печать, последний из них позволяет настроить принтер, а *Print Preview* предварительно увидеть, как будет выглядеть вывод на печать. Завершают этот список перечень недавно открывавшихся файлов и выход из программы *Exit*.

Следующий пункт основного меню *Edit* — редактирование.

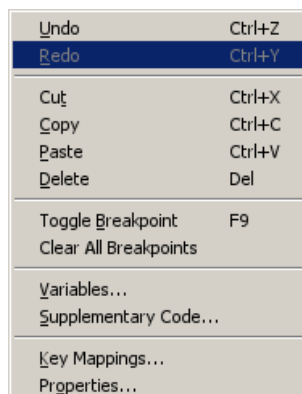


Рис. 6.6. Выпадающее меню редактирования (основное меню)

Я хочу подчеркнуть, что это вид выпадающего меню из списка основного меню, поскольку, щелкнув правой клавишей мышки в окне графического редактора, в зависимости от места, по которому вы щелкните, вы получите выпадающие меню для работы с элементами графики.

В этом меню первая половина относится к обычным операциям с объектами любого редактора: *Undo* и *Redo* — две взаимно противоположные команды по отмене последнего действия и возвращения его; *Cut* — вырезать; *Copy* — копировать; *Paste* — вставить; *Delete* — удалить. Два следующих пункта относятся к режиму отладки программы: *Toggle*

Breakpoint — переключение точки останова (точка останова позволяет вам остановить выполнение программы при отладке, чтобы, например, посмотреть значения переменных), если точка останова была включена (задана), то она выключится, и наоборот; *Clear All Breakpoints* — очистит вашу программу от всех точек останова, которые вы задавали.

Следующий пункт в меню *Variables...* открывает диалоговое окно, в котором вы можете определить переменные вашей программы, если они вам нужны. Программы всегда полны переменных, с которыми они и работают, посмотрим, как много переменных вам потребуется для написания вашей первой программы.

Следующий пункт *Supplementary Code...* (дополнительный код) позволяет при желании (или необходимости) добавить код к графической программе, открывая окно диалога, где есть поле для объявления функции и поле для реализации этой дополнительной функции, что соответствует языковым конструкциям с разделением *Definitions and function declarations* (определение и объявление функции) и *Function implementations* (реализация функции).

В соответствующих окнах мини-редакторов можно ввести текст, который после нажатия клавиши **ОК** позволит использовать эту дополнительную функцию. Возможность добавить нужные коды в программу значительно расширяет функциональность FlowCode для профессионалов, но и любителям, уже освоившимся в программе, достаточно хорошо владеющим языком программирования, это дополнение не покажется лишним.

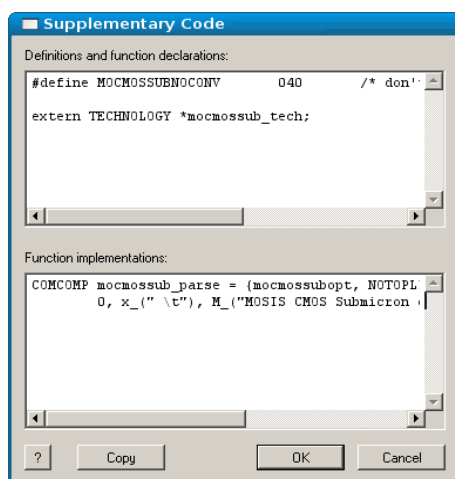


Рис. 6.7. Диалоговое окно добавления функций

Все выпадающие меню в программе «чувствительны» к происходящему в окне редактирования. И пока вы не добавите в программу клавиатуру (Keypad) из набора дополнительных элементов, следующий пункт *Key Mappings...* (карта соответствий) не будет активен. Но с появлением клавиатуры в проекте вы можете изменить свойства этого дополнительного элемента.

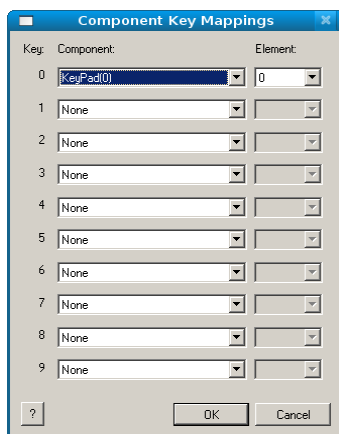
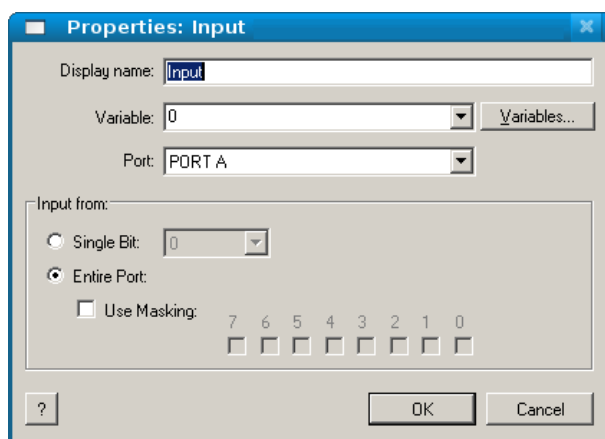


Рис. 6.8. Окно работы со свойствами клавиатуры

Аналогично поведение и последнего элемента списка из набора возможностей редактирования — *Properties...* (свойства). Чтобы «оживить» его, вам следует выделить любой элемент программы. Тогда, щелкнув по этому пункту меню, вы попадете в окно диалога свойств выбранного элемента. Вид этого диалога и предоставляемые пользователю возможности зависят от выделенного элемента. Так для элемента *input* (вход) можно определить его состояние, можно задать привязку к переменной и к порту, выбрать работу с одним битом или всем портом, задать маску, поставив галочку в поле *Use Masking*.

Каждый элемент программы в FlowCode имеет свой набор свойств, назначение и смысл которых яснее всего проявляются тогда, когда вы начинаете работать над своей программой. У вас обязательно появятся вопросы вида — как сделать так, чтобы в конкретном месте программы проверить состояние только одного бита порта, а не всех вводов? Например, когда вы дальше в программе хотите использовать ветвление, зависящее от состояния одного бита. Очень часто это проверка состояния «флага» (одного бита переменной или регистра, меняющего свое значение по окончании процесса или по результату операции).

Рис. 6.9. Диалоговое окно свойств элемента программы *input*

Продвигаясь дальше по основному меню, вы можете открыть выпадающее меню *View* (вид).

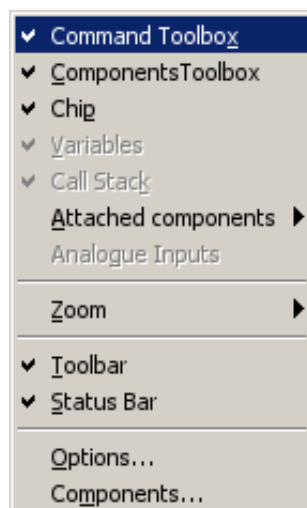
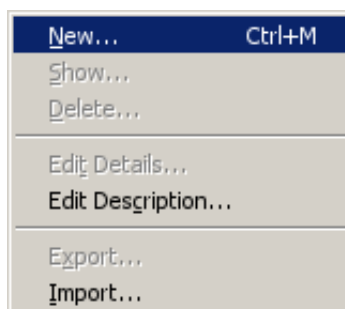


Рис. 6.10. Меню вид

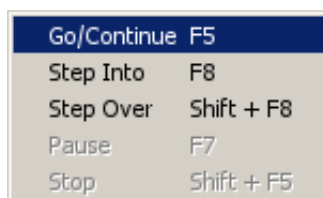
«Галочки» слева показывают, какие компоненты вы желаете видеть при работе: *Command Toolbox* (инструментальное меню команд), *Components Toolbox* (инструментальное меню компонент), *Chip* (контроллер, с которым вы работаете), *Variables* (переменные, если они есть), *Call Stack* (стек вызовов подпрограмм), *Attached components* (присоединенные компоненты, если вы добавили клавиатуру или светодиоды), *Analogue Inputs* (аналоговые входы). Присоединенные компоненты появляются в виде дополнительного списка, на что указывает стрелка справа от пункта меню. Как и дополнительный список масштаба изображения *Zoom*, который вы можете выбрать, увеличить или уменьшить, привести к заполнению экрана или ширине окна редактирования.

Вы можете также включить или выключить *Toolbar* (основное инструментальное меню) и *Status Bar* (строку состояния). Вы можете изменить свойства проекта, такие как цвет рисунка и фона, шрифт, используя раздел *Options...*, или выбрать из предлагаемого списка, какие компоненты вам нужны в настоящее время для работы — *Components...* Все это позволяет вам выбрать комфортный в вашей работе вид программы: только нужные компоненты, цветовую гамму, возможность легко читать все надписи в элементах программы и т.д. Не забывайте только, что, выключив из списка компонентов что-то, вы не увидите этот компонент при следующем запуске программы, и, если сегодня он вам не нужен, то завтра может понадобиться.

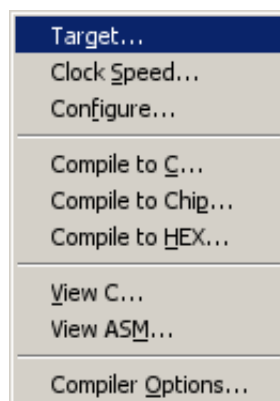
Следующие пункты основного меню я опишу кратко. Тому несколько причин — многое из этого вам понадобится не сразу, а, когда понадобится, вы будете разбираться в среде программирования микроконтроллеров FlowCode не хуже меня; рассказ о перечне пунктов меню к тому времени, когда вам понадобится воспользоваться чем-то, забудется, и легче самому сообразить, чем найти в тексте описания; да и не интересно начинать работу с программой, если ты о ней все уже знаешь, гораздо приятнее делать «свои маленькие открытия».

Рис. 6.11. Меню работы с макросами *Macro*

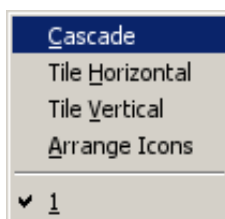
New... (новый макрос), *Show...* (показать макрос), *Delete...* (удалить), *Edit Details...* (редактировать детали), *Edit Description...* (редактировать описание, которое очень полезно делать), *Export...* (экспортировать), *Import...* (импортировать).

Рис. 6.12. Меню отладки *Run*

Go/Continue (запуск/продолжение), *Step Into* (шаг внутрь, например, функции), *Step Over* (шаг через, например, условие), *Pause* (пауза), *Stop* (стоп).

Рис. 6.13. Меню работы с контроллером *Chip*

Target... (выбор модели), *Clock Speed...* (частота тактового генератора, зависящая от требуемой), *Configure...* (слово конфигурации), *Compile to C...* (транслировать на Си), *Compile to Chip...* (транслировать все), *Compile to Hex...* (получить hex-файл загрузки), *View C...* (просмотр Си кода), *View ASM...* (просмотр ассемблерного кода), *Compiler Options...* (опции компиляции).

Рис. 6.14. Меню работы с окнами *Window*

Черепицей, горизонтально, вертикально, упорядочить иконки — вот, что можно сделать с окнами.

Последний пункт основного меню *Help*, на тот случай, если есть желание или необходимость обратиться к руководству по работе с программой.

В нижней части окна (справа) на рис. 6.4, а при запуске программы левая инструментальная панель — это панель команд.

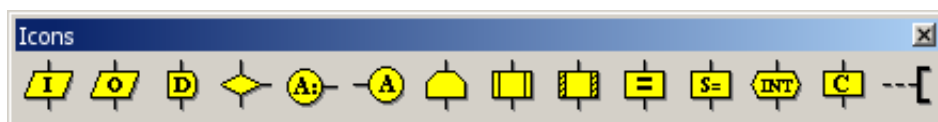


Рис. 6.15. Инструментальная панель команд

Перечень представленных команд (слева-направо на рисунке, сверху-вниз при запуске):

Input (ввод), *Output* (вывод), *Delay* (пауза), *Decision* (ветвление), *Connection Point* (две точки соединения), *Loop* (цикл), *Macro* (макрос), *Component Macro* (макрос компонента, добавленного в программу), *Calculation* (вычисление), *String Manipulation* (строковые операции), *Interrupt* (прерывание), *C Code* (блок кода на языке Си), *Comment* (комментарий).

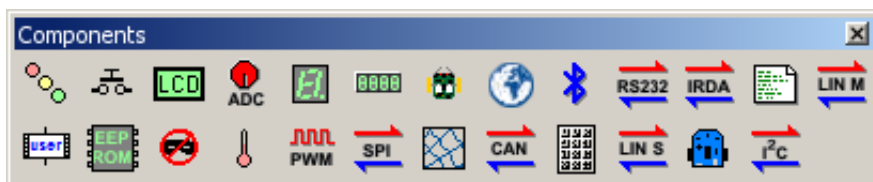


Рис. 6.16. Инструментальная панель добавочных компонентов

Компоненты (слева-направо):

LEDs (светодиоды), *Switches* (переключатели), *LCDDisplay* (жидкокристаллический дисплей), *ADC* (АЦП, если есть порт АЦП), *LED7Seg1* (семисегментный индикатор), *LED7Seg4* (блок из 4х семисегментных индикаторов), *Buggy* (компонент игрушки), далее несколько стандартных интерфейсов *TCP_IP*, *Bluetooth*, *RS232*, *IrDA*, *AddDefines* (добавить определения), *LinMaster* (ведущий в локальной сети), *Custom* (заказной компонент), *EEPROM* (перепрограммируемая память), *Alarm* (охранное устройство), *Thermometer* (термометр), *PWM* (шиотно-импульсный модулятор), *SPI* (последовательный внешний интерфейс), *WebServer* (web-сервер), *CAN* (сеть абонентского доступа), *KeyPad* (клавиатура), *LinSlave* (ведомый в локальной сети), *FormulaFlowCode* (компонент игры), *I2C* (шина связи между ИС).

Столь беглого описания программы, конечно, мало, чтобы считать себя экспертом в работе с ней, и сам я с ней не работаю, но, давайте, посмотрим, как начать работать с программой,

возобновив разговор с того места, где мы увидели рабочее окно программы (рис. 6.4.). С программой поставляются очень полезные примеры, которые я не хочу воспроизводить, но покажу выдержки из моей переписки с интернет-знакомым Николаем.

Вот, что он пишет: *«А чтобы начать написание программы, нужно собрать какую то схему, например, мультивибратор, чтобы в последствии можно было проверить на деле, что получается. Если можете предложить такую схемку, чтобы мы вместе могли бы следить за ситуацией, буду рад».*

Первоначально я намеревался ответить, что схему предложить не могу. Это так. Но затем я подумал, что и мне, как «железячнику», понятие схемы ближе, да и сколько раз я подчеркивал, что работа с FlowCode похожа на использование микросхем при разработке устройства.

Вот, что получилось из моего ответа.

Мультивибратор при первом знакомстве с микроконтроллером

Мультивибратор — устройство, которое на своем выходе (пусть он будет один) периодически устанавливает высокий и низкий уровень. Поэтому от микроконтроллера нам потребуется один выход, на котором мы будем выставить высокий и низкий уровень. Если прочитать справочные данные выбранного микроконтроллера, то можно заметить, что при возможности произвольно использовать выводы портов для входа и выхода, некоторые выводы, при разумном подходе, лучше оставить для их функционального использования, а нужный выбрать из оставшихся. Но это совсем не обязательно. На первом шаге не будем на это обращать внимания. Выберем вывод, например, RA0 (я использую PIC16F628A), но при переносе на макетную плату, а к этому выводу мы подключим светодиод, при переносе на макетную плату постараемся «на всякий случай» включить светодиод так, чтобы не повредить вывод микросхемы, тем более, что светодиоду достаточно для свечения 5-10 мА.

Что нам нужно от программы? Или как, в самом простом виде, выглядит программа?

Включить RA0, пауза, выключить RA0, пауза.

Но это один период колебаний мультивибратора, то есть, пока он вибратор, а не мультивибратор. Чтобы он стал мульти-, эту программу следует «зациклить», нужен бесконечный цикл, в котором она будет повторяться.

Бесконечный цикл (Включить RA0, пауза, выключить RA0, пауза).

Все. Программа готова. Если паузы одинаковы, получим меандр. Если хотим менять скважность или частоту, поменяем длительность пауз.

Берем программу FlowCode (запускаем, выбираем *Create New File*, выбираем *device* (модель) из списка, pic16f628a), получаем:

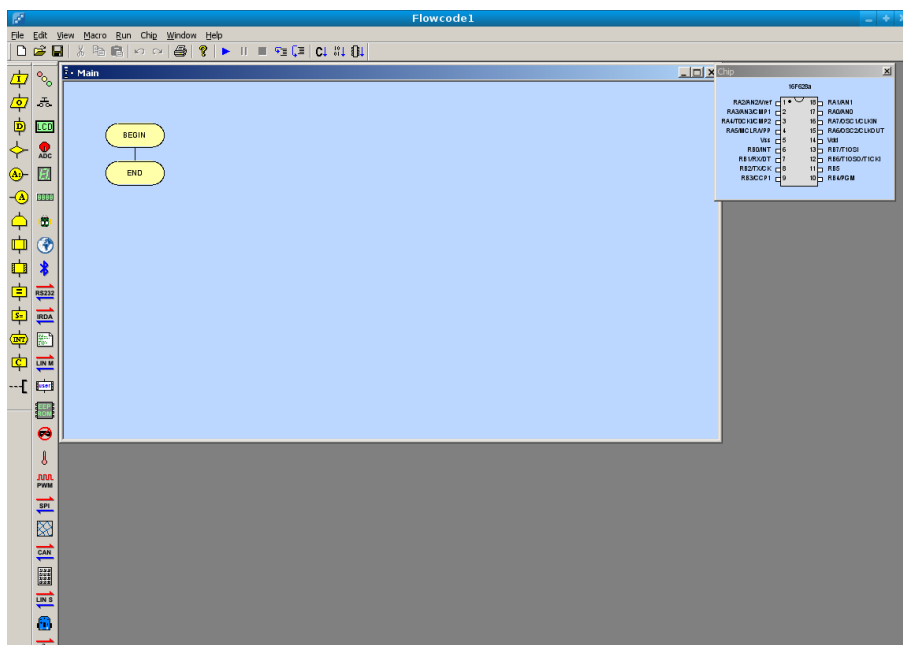


Рис. 7.1. Создание рабочего проекта в FlowCode

BEGIN-END в желтых овалах во многих языках программирования называют операторными скобками. В такие скобки заключают процедуры, функции и всю программу, например, в классическом Pascal. И наша программа будет заключена в эти скобки.

Поскольку вся программа должна находиться внутри бесконечного цикла, первое, что следует добавить в программу — цикл.

Для этих целей служит крайняя инструментальная панель слева — панель команд. Если медленно перемещать курсор по иконкам, делая остановки на каждой из них, то появляется подсказка по командам. Найдем команду цикла (*Loop*). Нажав левую клавишу мышки, когда курсор над иконкой *Loop*, и не отпуская клавиши «перенесем» команду к линии, соединяющей BEGIN-END, где и отпустим клавишу мышки. В итоге получим такой вид программы:

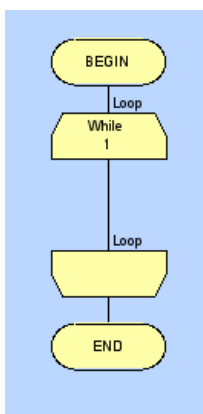


Рис. 7.2. Вид программы с циклом

Теперь добавим «Включить RA0», то есть выход RA0 установить в единицу.

На той же инструментальной панели команд найдем иконку *Output* (выход). И, как и в первый раз, «перетащим ее внутрь цикла. Получим:

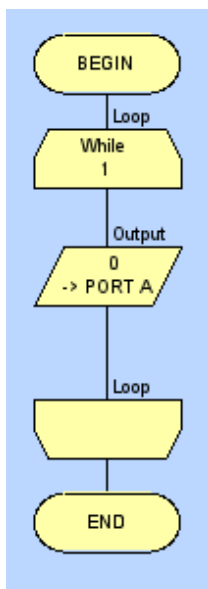


Рис. 7.3. Вид программы с «выходом» RA0

Но наш выход не настроен, вернее, не установлен в «1». Двойным щелчком по элементу *Output* (0 -> PORT A) открываем диалоговое окно.

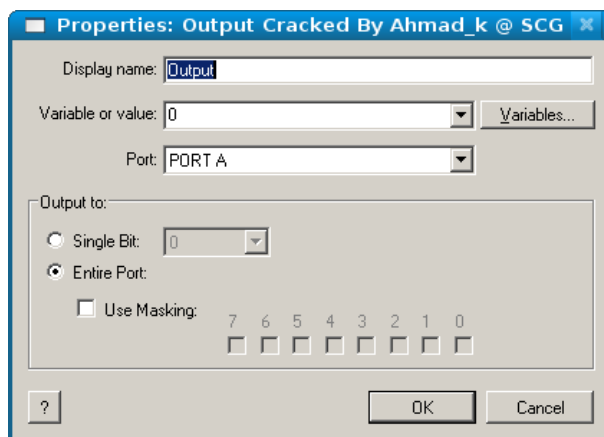


Рис. 7.4. Диалоговое окно свойств порта А

Не будем пока обращать внимание на все элементы окна, но изменим значение «0» в окне *Variable or value* (переменная или значение) на «1». Нажмем на клавишу **ОК** и отметим, что желтый параллелограмм *Output* немного изменился.

Теперь на той же панели команд ищем паузу (*Delay*), которую «перетаскиваем» чуть ниже предыдущей команды.

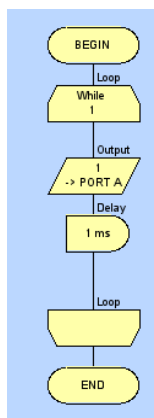


Рис. 7.5. Добавление паузы в программу

Двойным щелчком по этому элементу открываем окно свойств паузы, поскольку 1ms, скорее всего, означает 1 миллисекунда.

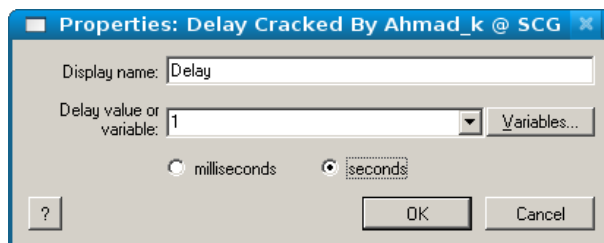


Рис. 7.6. Диалоговое окно свойств паузы

Нас устроит значение 1, но не устроит единица измерения, что мы исправим, щелкнув в поле *seconds*. Клавиша **ОК**, и изменение вида в значке паузы.

Мы сделали паузу, пора скушать «твикс».

В основном меню программы выбираем *File*, выбираем *Save As...* и сохраняем плоды трудов под именем *multi* (меняем уже заготовленное *Flowcode1*, оставляя расширение *.fcf*). Сохранять файлы удобнее всего в папке, например, *work*, которую лучше создать, скажем, в корневой директории. Некоторые программы очень чувствительны к длине пути, очень многие не любят, когда встречаются на своем пути что-то «русское», то есть, что-то написанное кириллицей, как «Мои документы», и, хотя предлагают сохранить в папке «Мои документы», сами потом найти ничего не могут.

Итак, сохранили. Теперь можно доделать программу, повторив добавление выхода, где уже не надо ничего менять, открывая диалог свойств, и повторив добавление паузы, где миллисекунды не следует забыть заменить секундами. В конечном счете программа приобретает вид:

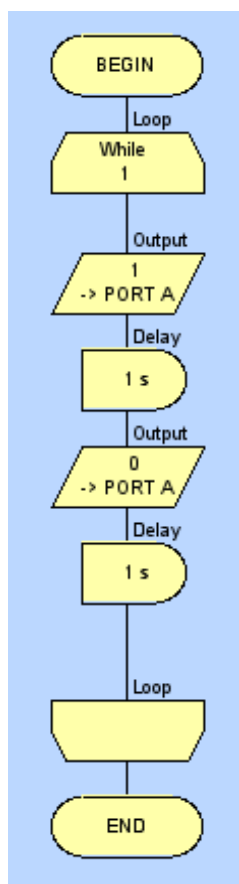


Рис. 7.7. Окончательный вид программы мультивибратора

Собственно, поскольку программа готова, ее можно транслировать в файл для загрузки с помощью программы программатора (*icprog* или другой) или, если подключен программатор, работающий с *FlowCode*, то прямо из среды программирования. Но для начала ее можно проверить (и отладить, если что-то не так). Для отладки нам понадобится, так удобнее, внешний компонент в виде светодиода.

Для внешних компонентов есть инструментальная панель (правее панели команд), где самая верхняя иконка — это набор светодиодов для всего порта. Достаточно щелкнуть по иконке, чтобы в окне программы появились светодиоды.

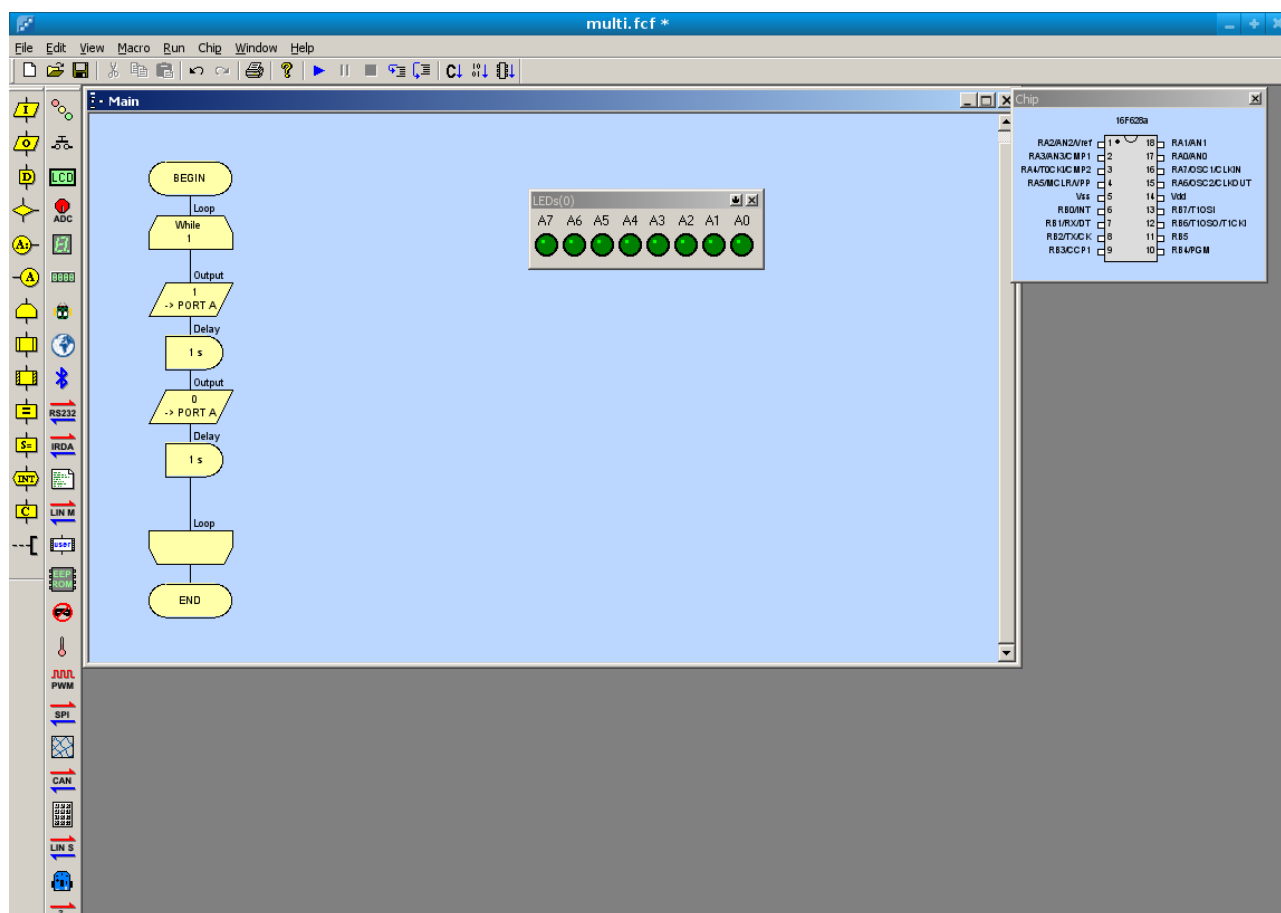


Рис. 7.8. Программа с добавленными светодиодами

Под основным меню программы есть основная инструментальная панель. Правую ее часть (после иконки со знаком вопроса) занимают элементы отладки программы, которые начинаются с клавиши *Play* (или *Run* ►), нажав на которую можно увидеть, как мигает светодиод. Это все?

Или почти все. В основном меню есть пункт *Chip* с разделом *Clock Speed...* Открыв (нажав) диалоговое окно этого раздела, можно задать тактовую частоту микроконтроллера, которую я выбираю равной 4 МГц.

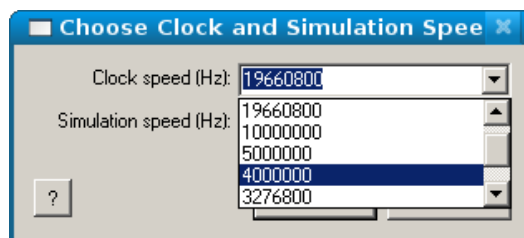


Рис. 7.9. Диалоговое окно выбора тактовой частоты контроллера

Почему 4 МГц? У микроконтроллера PIC16F628A есть возможность использовать внутренний тактовый генератор, а не внешнюю RC цепь или внешний кварцевый резонатор. Для нужд задачи этого более, чем достаточно, и не требует дополнительных элементов. А, чтобы использовать внутренний тактовый генератор, следует задать определенное слово конфигурации. Это слово конфигурации записывается в контроллер при программировании и

не меняется при выполнении программы контроллера, оно определяет ряд свойств контроллера, словом, его конфигурацию. Разумно поискать в разделе *Chip* и подразделе *Configure...* основного меню возможность определить это слово. Я хочу сделать его равным 3F18h (шестнадцатеричное число). Для работы со словом конфигурации (есть еще название фьюзы) служит устанавливаемый при установке программы дополнительный элемент PPPv3. До этого момента я работал с FlowCode в Linux, для разнообразия перейду в Windows.

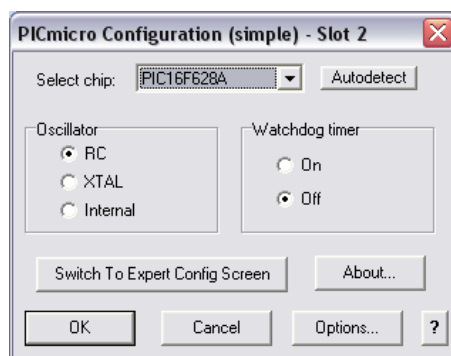


Рис. 7.10. Диалоговое окно задания конфигурации контроллера

Поскольку я хочу быть уверен, что все биты слова конфигурации установлены в соответствии с моими желаниями, я нажимаю клавишу **Switch To Expert Config Screen** (переключение к экспертному виду экрана).

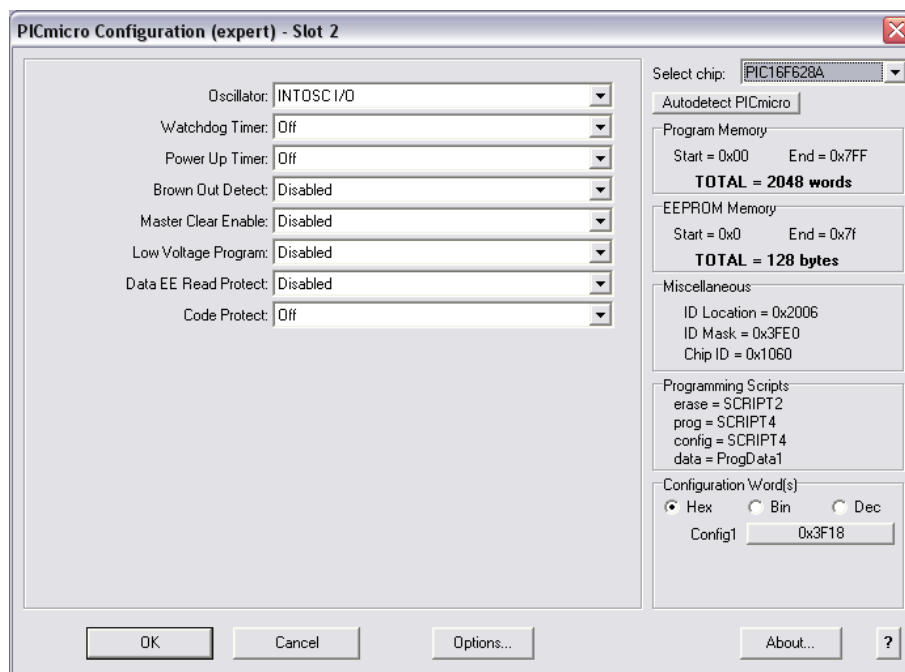


Рис. 7.11. Диалоговое окно полного управления конфигурацией

Я установил все биты слова конфигурации, как намеревался. В правой нижней части на клавише, подписанной как Config1 видно слово конфигурации **0x3F18** (0x — еще одно обозначение шестнадцатеричных чисел). Если нажать на клавишу, то появляется окно, в котором можно ввести это число без установки бит.

Назначение и смысл этих бит, назначение настройки конфигурации можно прочесть в справочной документации к конкретной модели микроконтроллера. Количество и назначение

бит конфигурации (в слове конфигурации, которое для PIC16F628A записывается только при программировании по адресу 2007h) зависит от конкретной модели и производителя микросхемы.

Последнее, что хотелось бы проверить, работает ли программа после всех дополнительных изменений. Клавиша ► позволяет убедиться, что все работает.

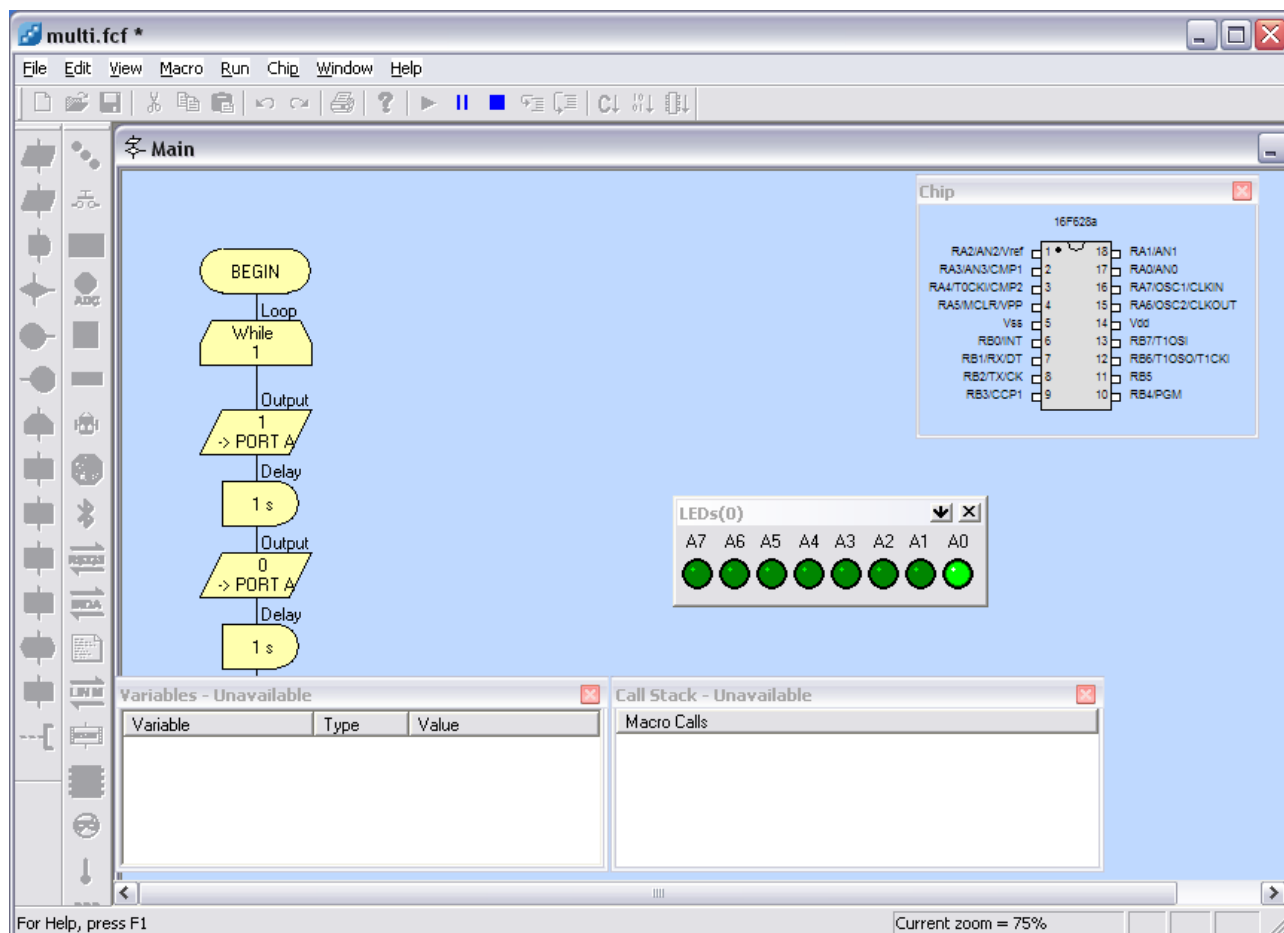


Рис. 7.12. Повторная проверка работы программы

В этом месте можно было бы транслировать программу в файл для загрузки в микроконтроллер, hex-файл. Для этого в основном меню *Chip* подраздел *Compile to HEX...* и служит. После загрузки программы в контроллер, его можно было бы перенести на макетную плату, подпаять светодиод (с резистором?), подключить питание и посмотреть на результат воочию. Но мне интересно, как выглядит эта программа в виде, написанном на языке Си.

Чуть выше команды компиляции в файл загрузки есть команда *Compile to C...*, которая должна создать файл с программой, написанной на языке Си. Нажав этот раздел меню, после появления сообщения об удачной компиляции, я могу перейти в папку, где расположен проект и найти там этот файл. Или я могу в том же пункте основного меню выбрать подраздел *View C...* После нажатия мышкой на этот раздел можно в блокноте (текстовый редактор Windows) увидеть текст программы на языке Си.

```

/*****
/**
/** File name:   Z:\home\vladimir\Рабочий стол\multi\multi.c
/** Generated by: Flowcode v3.2.2.40

```

```
/** Date:      Thursday, August 14, 2008 17:09:00
/** Licence:   free
/** Registered to: vladimir
/*******
```

```
#define MX_PIC
//Defines for microcontroller
#define P16F628A
и т.д.
```

Основная часть программы находится под объявлением void main():

```
#define MX_EE
#define MX_EE_TYPE1
#define MX_EE_SIZE 128
#define MX_UART
#define MX_UART_B
#define MX_UART_TX 2
#define MX_UART_RX 1
#define MX_PWM
#define MX_PWM_CNT 1
#define MX_PWM_TRIS1 trisb
#define MX_PWM_1 3

#include <system.h>
#pragma CLOCK_FREQ 4000000
#pragma DATA 0x2007, 0x3f18
#include "C:\Program Files\Matrix Multimedia\Flowcode V3\FCD\internals.h"

void main()
{
    cmcon = 0x07;

    while( 1 )
    {
        trisa = 0x00;
        porta = 1;

        delay_s(1);

        trisa = 0x00;
        porta = 0;

        delay_s(1);
    }
}
```

Две фигурные скобки в самом начале и в самом конце — это те самые BEGIN-END, что были раньше. Другие две — операторные скобки цикла While, внутри которого те же составляющие из установки битов порта в высокое состояние и низкое, разбитые паузами delay. Вид программы на языке Си мало отличается от вида графического или того, что написано обычными словами. Добавленная команда `cmcon = 0x07;` указывает, что выводы порта A не будут использоваться для работы со встроенными компараторами. Команда `trisa = 0x00;` указывает, что порт A будет использован для вывода.

Этот файл можно использовать, например, с компилятором `picc-lite`, если заменить нижний регистр на верхний при написании команд, например, `COMCON = 0x07;`

При использовании другого компилятора, как `SDCC`, возможно, придется внести еще какие-то изменения.

Использование раздела *Compile to HEX...* приведет к появлению в папке проекта файла программы на языке ассемблер, hex-файла и ряда вспомогательных файлов для отладки. Чтобы сравнить, насколько сложнее выглядит код на языке ассемблера, можно взглянуть на соответствующий код файла на языке ассемблера:

```

        include "P16F628A.inc"
; Heap block 0, size:95 (0x000000A0 - 0x000000FE)
__HEAP_BLOCK0_BANK      EQU      0x00000001
__HEAP_BLOCK0_START_OFFSET EQU      0x00000020
__HEAP_BLOCK0_END_OFFSET EQU      0x0000007E
; Heap block 1, size:75 (0x00000025 - 0x0000006F)
__HEAP_BLOCK1_BANK      EQU      0x00000000
__HEAP_BLOCK1_START_OFFSET EQU      0x00000025
__HEAP_BLOCK1_END_OFFSET EQU      0x0000006F
; Heap block 2, size:0 (0x00000000 - 0x00000000)
__HEAP_BLOCK2_BANK      EQU      0x00000000
__HEAP_BLOCK2_START_OFFSET EQU      0x00000000
__HEAP_BLOCK2_END_OFFSET EQU      0x00000000
; Heap block 3, size:0 (0x00000000 - 0x00000000)
__HEAP_BLOCK3_BANK      EQU      0x00000000
__HEAP_BLOCK3_START_OFFSET EQU      0x00000000
__HEAP_BLOCK3_END_OFFSET EQU      0x00000000
gbl_status               EQU      0x00000003 ; bytes:1
gbl_indf                 EQU      0x00000000 ; bytes:1
gbl_tmr0                 EQU      0x00000001 ; bytes:1
gbl_pcl                  EQU      0x00000002 ; bytes:1
gbl_fsr                  EQU      0x00000004 ; bytes:1
gbl_porta                EQU      0x00000005 ; bytes:1
gbl_portb                EQU      0x00000006 ; bytes:1
gbl_pclath               EQU      0x0000000A ; bytes:1
gbl_intcon               EQU      0x0000000B ; bytes:1
gbl_pir1                 EQU      0x0000000C ; bytes:1
gbl_tmr1l                EQU      0x0000000E ; bytes:1
gbl_tmr1h                EQU      0x0000000F ; bytes:1
gbl_t1con                EQU      0x00000010 ; bytes:1
gbl_tmr2                 EQU      0x00000011 ; bytes:1
gbl_t2con                EQU      0x00000012 ; bytes:1
gbl_ccpr1l               EQU      0x00000015 ; bytes:1
gbl_ccpr1h               EQU      0x00000016 ; bytes:1
gbl_ccplcon              EQU      0x00000017 ; bytes:1
gbl_rcsta                EQU      0x00000018 ; bytes:1
gbl_txreg                EQU      0x00000019 ; bytes:1
gbl_rcreg                EQU      0x0000001A ; bytes:1
gbl_cmcon                EQU      0x0000001F ; bytes:1
gbl_option_reg           EQU      0x00000081 ; bytes:1
gbl_trisa                EQU      0x00000085 ; bytes:1
gbl_trisb                EQU      0x00000086 ; bytes:1
gbl_piel                 EQU      0x0000008C ; bytes:1
gbl_pcon                 EQU      0x0000008E ; bytes:1
gbl_pr2                  EQU      0x00000092 ; bytes:1
gbl_txsta                EQU      0x00000098 ; bytes:1
gbl_spbrg                EQU      0x00000099 ; bytes:1
gbl_eedata               EQU      0x0000009A ; bytes:1
gbl_eeadr                EQU      0x0000009B ; bytes:1
gbl_eecon1               EQU      0x0000009C ; bytes:1
gbl_eecon2               EQU      0x0000009D ; bytes:1
gbl_vrcon                EQU      0x0000009F ; bytes:1
delay_ms_00000_arg_del   EQU      0x00000024 ; bytes:1
delay_s_00000_arg_del    EQU      0x00000023 ; bytes:1

```

```

Int1Context          EQU      0x0000007F ; bytes:1
Int1BContext         EQU      0x00000020 ; bytes:3
    ORG 0x00000000
    GOTO _startup
    ORG 0x00000004
    MOVWF Int1Context
    SWAPF STATUS, W
    BCF STATUS, RP0
    BCF STATUS, RP1
    MOVWF Int1BContext
    SWAPF PCLATH, W
    MOVWF Int1BContext+D'1'
    SWAPF FSR, W
    MOVWF Int1BContext+D'2'
    BCF PCLATH, 3
    BCF PCLATH, 4
    GOTO interrupt
    ORG 0x00000010
delay_ms_00000
; { delay_ms ; function begin
    MOVF delay_ms_00000_arg_del, F
    BTFSS STATUS, Z
    GOTO label4026531857
    RETURN
label4026531857
    MOVLW 0xF9
label4026531858
    ADDLW 0xFF
    BTFSS STATUS, Z
    GOTO label4026531858
    NOP
    DECFSZ delay_ms_00000_arg_del, F
    GOTO label4026531857
    RETURN
; } delay_ms function end

    ORG 0x0000001C
delay_s_00000
; { delay_s ; function begin
label4026531863
    MOVLW 0xFA
    MOVWF delay_ms_00000_arg_del
    CALL delay_ms_00000
    MOVLW 0xFA
    MOVWF delay_ms_00000_arg_del
    CALL delay_ms_00000
    MOVLW 0xFA
    MOVWF delay_ms_00000_arg_del
    CALL delay_ms_00000
    MOVLW 0xFA
    MOVWF delay_ms_00000_arg_del
    CALL delay_ms_00000
    DECFSZ delay_s_00000_arg_del, F
    GOTO label4026531863
    RETURN
; } delay_s function end

    ORG 0x0000002B
main
; { main ; function begin
    MOVLW 0x07

```

```
        BCF STATUS, RP0
        BCF STATUS, RP1
        MOVWF gbl_cmcon
        MOVLW 0xC0
        BSF STATUS, RP0
        MOVWF gbl_option_reg
label268439613
        BSF STATUS, RP0
        CLRF gbl_trisa
        MOVLW 0x01
        BCF STATUS, RP0
        MOVWF gbl_porta
        MOVWF delay_s_00000_arg_del
        CALL delay_s_00000
        BSF STATUS, RP0
        CLRF gbl_trisa
        BCF STATUS, RP0
        CLRF gbl_porta
        MOVLW 0x01
        MOVWF delay_s_00000_arg_del
        CALL delay_s_00000
        GOTO label268439613
; } main function end

        ORG 0x00000041
_startup
        BCF PCLATH, 3
        BCF PCLATH, 4
        GOTO main
        ORG 0x00000044

        ORG 0x00002007
        DW 0x3F18
        END
```

Этот файл можно попытаться использовать в программе MPLAB.

Для этого запустим программу, запустим мастера создания новых проектов Project-Project Wizard...

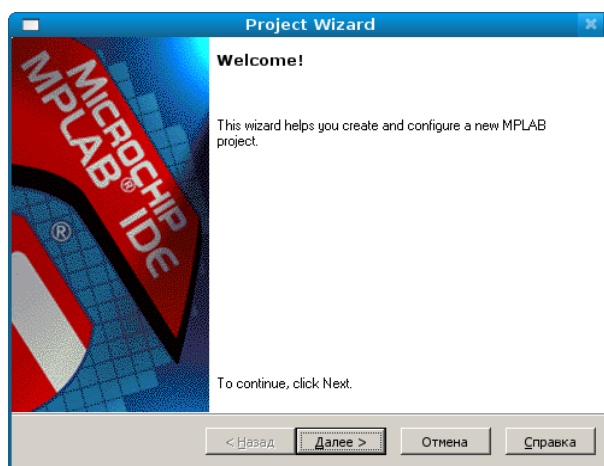


Рис. 7.13. Запуск мастера создания новых проектов

Клавиша **Далее** (Next) позволяет переместиться по всем этапам создания нового проекта от выбора микроконтроллера (Device), через выбор транслятора с языка программирования:

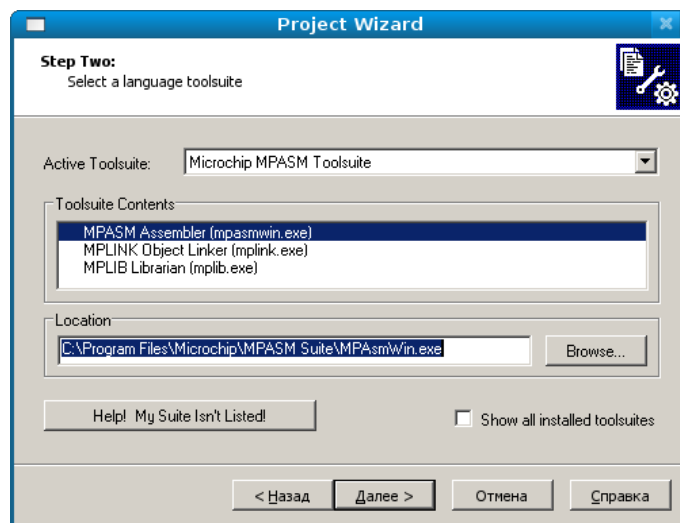


Рис. 7.14. Выбор транслятора для проекта

задание имени проекта и места расположения проекта.

Можно создать новый файл во вновь созданном проекте *File-New*. В этот файл можно скопировать текст, созданный программой FlowCode, и сохранить его (*File-Save As...*), выбрав имя файла, например, совпадающее с именем проекта, в той же папке, где лежит проект, и расширением .asm.

После выхода из MPLAB и подтверждения сохранения вида рабочего поля проекта (при необходимости), при следующей загрузке программы следует открыть проект *Project-Open*, в менеджере проекта:

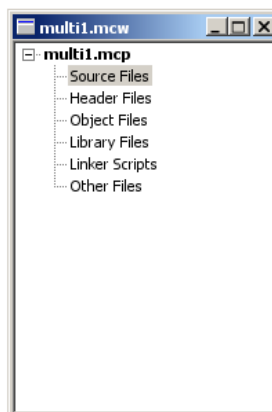


Рис. 7.15. Менеджер проекта в MPLAB

щелкнуть правой клавишей мышки по разделу *Source Files* и выбрать в появившемся меню *Add Files...* Указав недавно созданный файл (имя_файла.asm), мы получим добавленный в проект файл исходного кода.

Этот файл можно транслировать *Project-Build All*. После удачной трансляции, задав необходимые средства отладки *Debugger-Select Tool-MPLAB SIM*, можно запустить отладку программы *Debugger-Animate*.

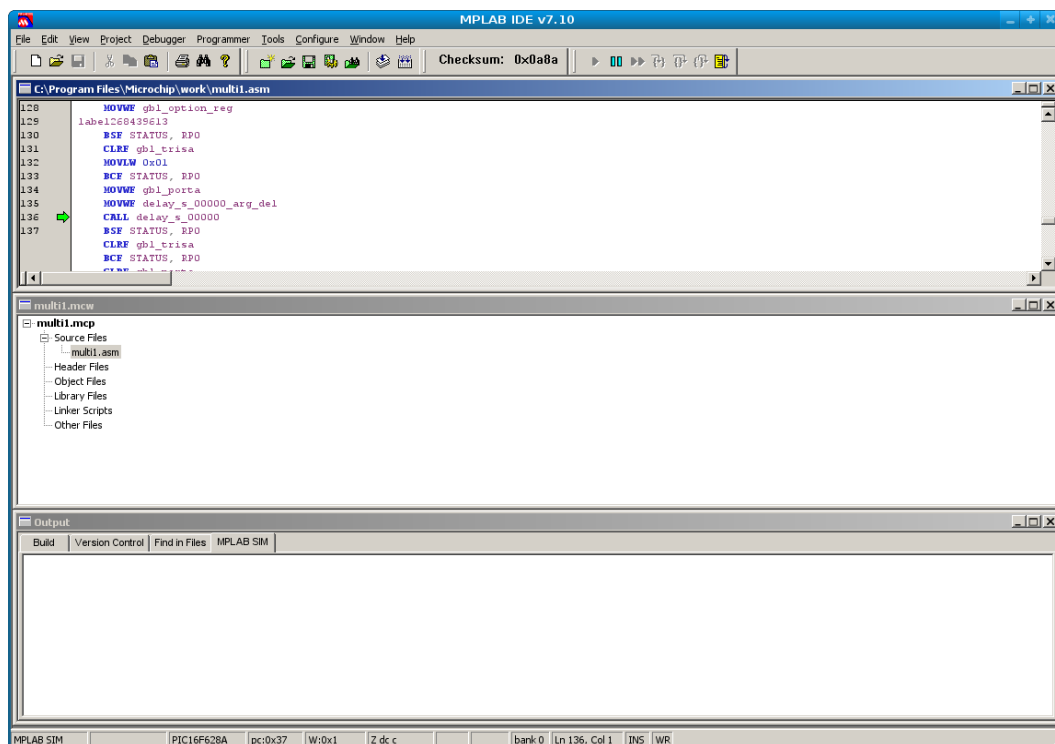


Рис. 7.16. Отладка программы в MPLAB

Тот факт, что ассемблерный код программы FlowCode оказывается вполне «подходящим» кодом для MPLAB, указывает на один и тот же компилятор (или очень схожий), используемый двумя программами.

Можно было бы мне и не упоминать о языках программирования, приводя длинные тексты программы, и о другой программе MPLAB, но, зная неумную любознательность радиолюбителей, я хотел показать, что сделав первый шаг, всегда можно сделать второй.

Разобравшись со схемой мультивибратора, мне пришлось разбираться со схемой двух мультивибраторов, работающей с разной частотой. Не самая сложная задача, но, пожалуй, есть смысл ее упомянуть. Вот мой ответ Николаю.

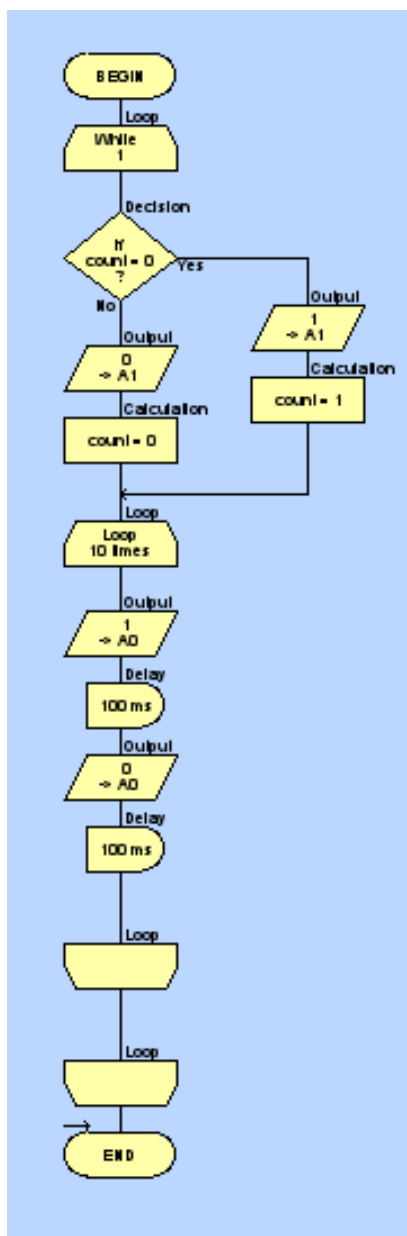
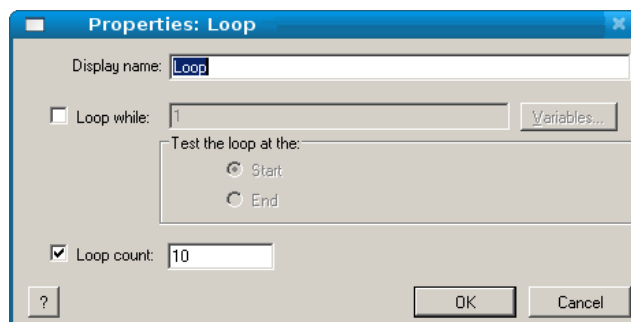


Рис. 7.17. Программа двух мультивибраторов

Внутренний цикл (*Loop*) взят из предыдущей программы, но бесконечный цикл заменен счетным, что позволяет сделать окно диалога свойств цикла. Время пауз уменьшено в десять раз (100 мС), а количество циклов до завершения выбрано равным 10 с тем, чтобы сохранить частоту мигания второго светодиода такой, как в предыдущей программе.

Рис. 7.18. Изменение свойств цикла (*Loop*)

Как можно видеть, графический элемент цикла изменил свой вид и надпись гласит «цикл 10 раз». После выполнения внутреннего цикла 10 раз программа переходит внутрь внешнего бесконечного цикла, который начинается с выяснения значения переменной *count*. Добавление переменных в программу выполняется вызовом диалогового окна менеджера переменных: *Edit-Variables...*

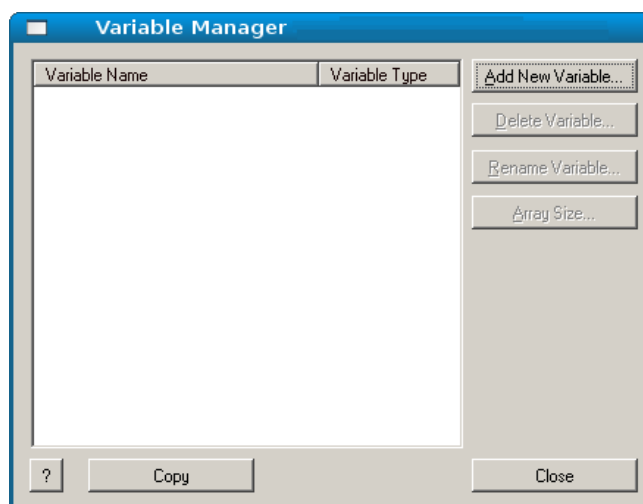


Рис. 7.19. Окно менеджера переменных

Клавиша **Add New Variable...** позволяет создать переменную в новом окне.

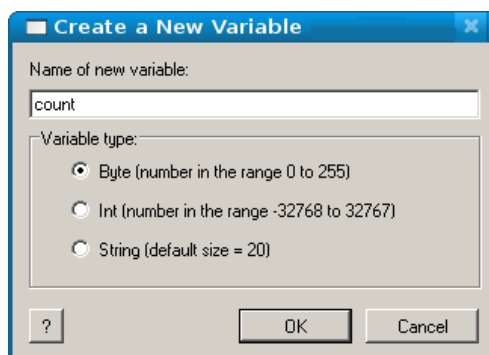


Рис. 7.20. Окно создания переменной

Выбор типа данных ограничен самым необходимым: байт, целое, строка, — но этого, как правило, вполне достаточно даже для сложных программ, если учесть, что программная память микроконтроллеров ограничена.

Проверка, возвращаясь к нашей программе, условия «равна ли переменная count нулю» вызывает ветвление программы: если да, то программа выполнится по «правому пути», где вывод RA1 порта A примет значение высокого уровня, а переменная count станет равна единице, и тогда при следующем проходе программа пойдет по пути, где ответ на проверку нулевого значения переменной count — нет. То есть вывод RA1 порта A устанавливается в низкое состояние, а переменная count становится равной нулю.

Внешний цикл бесконечный, поэтому программа будет выполнять функцию двух мультивибраторов, работающих с разной частотой.

«По науке» предыдущую программу следовало бы организовать с использованием таймера и прерывания по таймеру. Что такое прерывание?

Есть два основных приема работы процессора с внешними устройствами и событиями: «по опросу» и «по прерыванию». Получая задание, внешнее устройство снимает в регистре состояния флаг готовности. Процессор, если задание внешнему устройству передано не полностью, ожидает, когда внешнее устройство может продолжить работу, опрашивая готовность, то есть, ждет, когда внешнее устройство, освободившись, выставит флаг готовности. Как правило, процессор работает много быстрее, чем внешние устройства, и такое ожидание не рационально тратит процессорное время.

Чтобы избежать непроизводительных расходов, применяют второй метод. Получая задание от процессора, внешнее устройство начинает его выполнение, а процессор разрешает ему прерывание. Сам процессор с этого момента может заниматься выполнением другой работы. Когда внешнее устройство заканчивает выполнение части задания, оно выставляет требование прерывания. Сигнал прерывания заставляет процессор прервать свою работу, чтобы вернуться к ней позже процессор вводит все данные в стек, это адрес остановки, промежуточные результаты, и обращается к подпрограмме обработки прерывания, заданную вектором прерывания (адресом). После передачи внешнему устройству следующей «порции» задания процессор изымает из стека все данные по прерванной работе и продолжает заниматься делом до следующего требования прерывания. Такой механизм экономит время работы процессора над задачами, особенно, если внешние устройства медленные.

Использовать ли прерывания всегда? Не знаю. Если процессору нечего делать в те моменты, когда внешнее устройство выполняет задание, то я не вижу смысла в использовании прерывания. Некоторые микроконтроллеры при программировании требуют, чтобы программист отслеживал состояние стека, другие делают это автоматически. Что удобнее? Зависит, скорее всего, от конкретной ситуации, во всяком случае в этом вопросе лучше положиться на опыт программистов, работающих с микроконтроллерами. Но, думаю, не будет большого вреда, если вы напишете работающую программу не «по науке».

Я не буду описывать, как «собрана» программа на рисунке ниже, скажу только, что к первой программе добавлено прерывание по таймеру, в подпрограмму прерывания добавлена та часть, которая во второй программе «занимается» организацией работы второго мультивибратора, и, как можно видеть, ничем не отличается от фрагмента «внешнего» цикла. Я не буду уверять, что все прекрасно в этой программе, мне только хотелось сделать графический набросок к рассказу о прерывании.

Вообще, программа FlowCode имеет большие возможности, множество дополнительных компонент и длинный список микроконтроллеров. Мало того, если вы задумаете работать с контроллерами PIC и AVR, то сможете импортировать программы одного производителя в программу для работы с контроллерами другого.

Так что, описание всех возможностей программы, скорее тема для отдельной книги, а не

для небольшой статьи или главы.

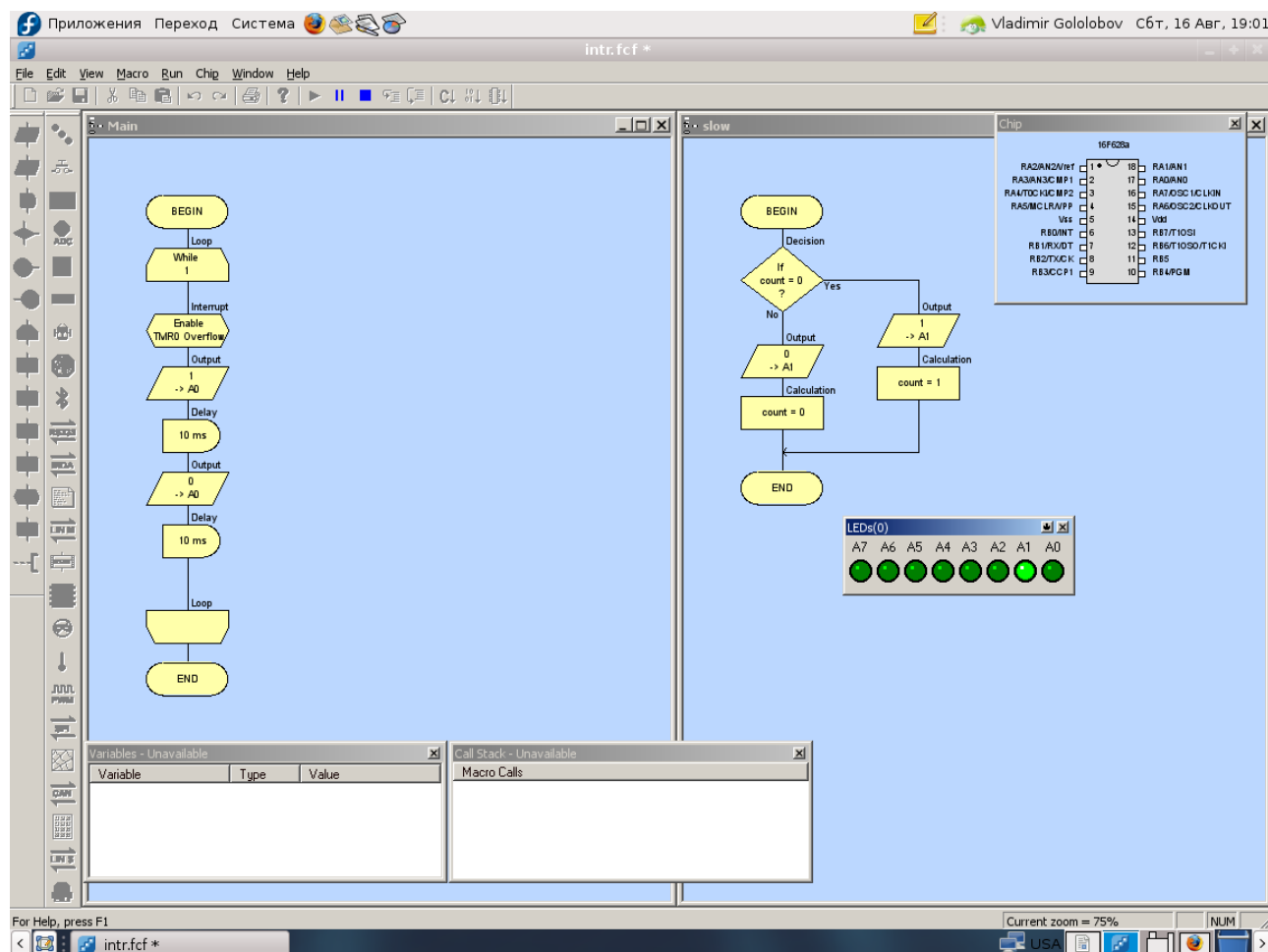


Рис. 7.21. Вид программы с использованием механизма прерывания

Завершая эту главу о микроконтроллерах и программе FlowCode, я хочу привести очень простой пример использования выводов контроллера не только для работы на «выход», но и для «опроса», скажем, датчиков. Это могут быть реальные датчики, многие из которых формируют сигнал с помощью контактов реле, это могут быть контакты клавиатуры, это могут быть с тем же успехом транзисторные ключи.

Программа, как многие написанные для микроконтроллеров, будет «крутиться» в бесконечном цикле. Первая команда — ввод состояния порта А. Это достигается добавлением внутрь цикла команды input, добавление дополнительного компонента switches и созданием переменной sw. Напомню, команды — инструментальная панель команд, переключатель — инструментальная панель компонент, а переменные — основное меню *Edit-Variables...-Add New Variable...*

В этом примере я использую весь порт А для входа, а контакты переключателя все присоединены к порту, и весь порт В я использую для выхода, к его выводам подключены светодиоды. Вместе с тем управлять я собираюсь только одним входом и одним выходом. Реальная программа может использовать единственный вывод порта А для опроса состояния контактов, при этом следующий вывод порта А можно применить для подключения светодиода. Можно, если контактов несколько написать программу, которая не только будет опрашивать состояние контактов, но и последовательность их замыкания: замыкание контактов на входе RA0, а следом RA1, могло бы означать одно направление движения,

обратная последовательность замыкания контактов — другое.

Вернемся к программе, в которой после опроса состояния входа добавим ветвление программы. Если контакты замкнуты — включим светодиод, если контакты разомкнуты, выключим его. Вид программы:

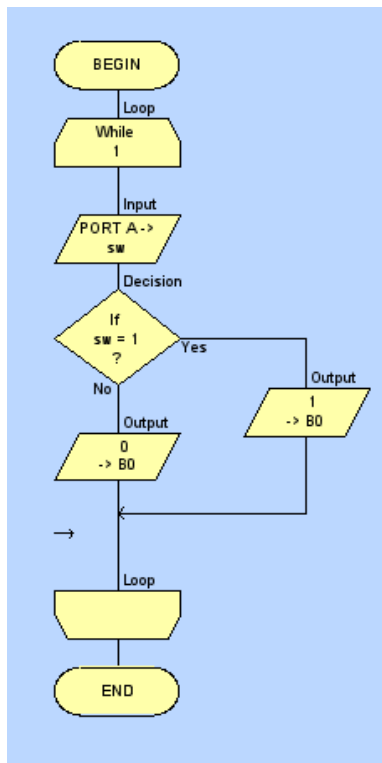


Рис. 7.22. Вид программы с использованием входов

Небольшая особенность в свойствах команды input.

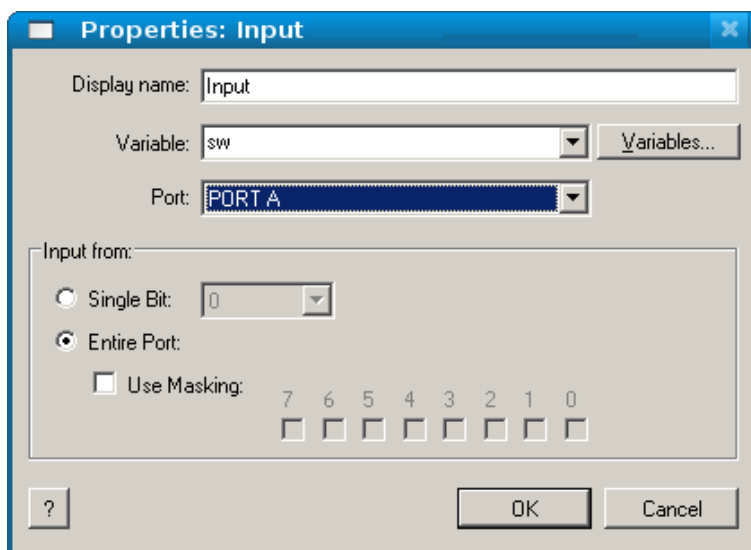


Рис. 7.23. Задание свойств элемента программы input

Используется не значение, а переменная sw.

Теперь, после запуска программы в режиме отладки, можно видеть, как при замыкании

контактов A0 переключателя (курсор мышки на этот контакт, нажать и удерживать левую клавишу), светодиод B0 загорается, а при размыкании контактов гаснет.

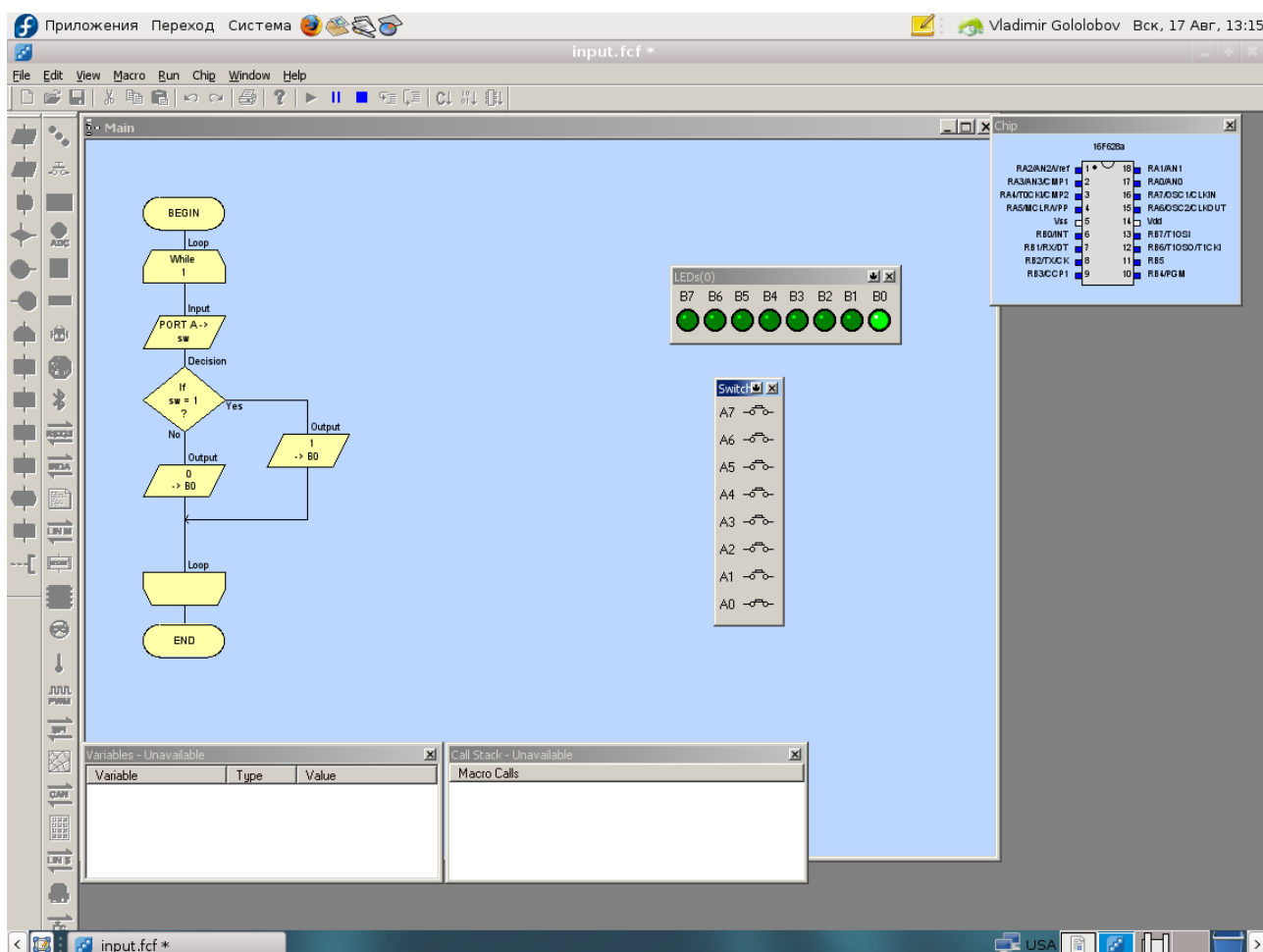


Рис. 7.24. Запуск программы использующей вход и выход контроллера

Даже начинающим последний пример может показаться верхом нелепости — соединил выключатель со светодиодом и получил результат, да и мультивибратор — одна микросхема и не нужно никаких программ. Но, если вам приходилось делать это не только на бумаге, а доводить работу до готового вида устройства, в почувствуете разницу, когда захотите менять частоту генерации мультивибратора. Используя контроллер, вы можете управлять частотой с помощью двух кнопок: увеличить частоту, уменьшить частоту. А отображать значение частоты можно на семисегментном индикаторе. Такая конструкция удобнее и проще, чем шкалы и бегунки. А как может усложниться схема, определяющая порядок замыкания контактов, я предоставляю проверить вам.

Устройства питания и преобразователи напряжения

В любительской практике не следует пренебрегать в качестве источников питания возможностью использовать батарейку не только в переносных устройствах, но и при отлаживании устройства на макетной плате. Современные батарейки с напряжением 9 В могут долго работать, отдавая достаточно большой ток. В сочетании с микросхемами стабилизаторов напряжения такое решение позволяет вам не задумываться о последствиях, когда, увлеченно меняя детали, вы забываете выключить питание схемы. Детали могут пострадать, но не вы.

Можно использовать и лабораторный блок питания, даже если это простейший сетевой адаптер с переключателем выходных напряжений. Простой блок питания легко собрать самостоятельно, используя трансформатор фабричного производства. Изготавливать трансформаторы начинающим любителям я бы не советовал, можно легко ошибиться, а результат ошибки может объявиться не сразу и в самый неподходящий момент.

Как устроен простой сетевой адаптер?

Он имеет понижающий трансформатор, выпрямитель и конденсатор.

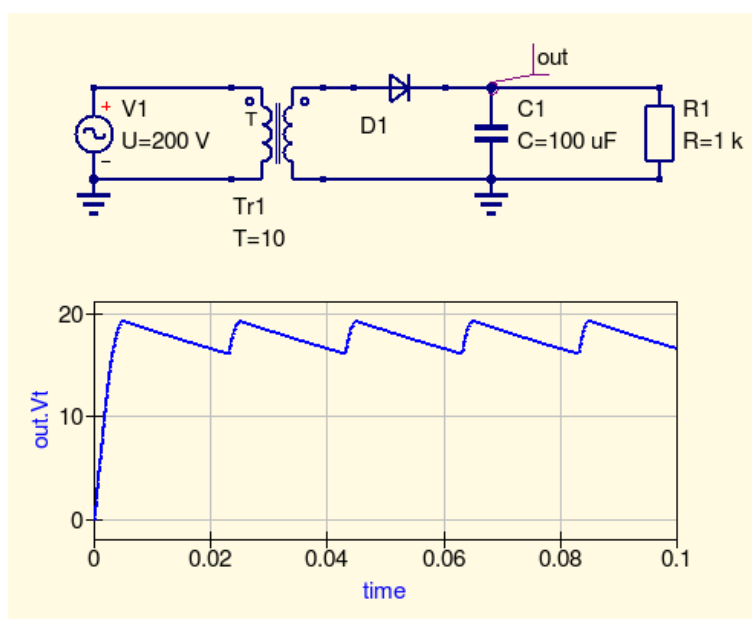


Рис. 8.1. Схема простого сетевого адаптера

Единственным отличием этой схемы от схемы купленного вами блока питания будет наличие не одного диода, а четырех, включенных по мостовой схеме, а, скорее всего, это будет диодная сборка.

Напряжение в розетке (в нашей стране) это переменное напряжение, меняющееся по величине по закону синуса и меняющееся по направлению, с действующим значением 220 В и частотой 50 Гц. Для питания многих устройств достаточно 5-12 В. Напряжения, измеряемые киловольтами, в наши дни уходят из бытовой практики с появлением жидкокристаллических телевизоров и мониторов. Не задумывались ли вы, почему в розетке не 12 В, а 220? Да еще переменное напряжение.

Проще ответить, почему переменное напряжение. В первую очередь потому, что есть такое чудесное устройство, как трансформатор. Трансформатор позволяет легко превратить

напряжение 220 В в любое другое. Две индуктивно связанные катушки трансформатора позволяют, включив первую обмотку в сеть, на второй получить напряжение меньше сетевого, если количество витков второй обмотки меньше, или более высокое напряжение, если количество витков больше. При расчетах трансформаторов широко используют такой параметр, как виток/вольт. Есть методики быстрого расчета трансформаторов, есть методики детального расчета, и опытные радиолюбители знают, как рассчитать трансформатор.

Трансформатор позволяет получить из сетевого напряжения любое заданное, это понятно. Но почему же в сети не 12 В?

Все устройства потребляют мощность. Посчитайте мощность всех лампочек в вашей квартире, добавьте мощность, потребляемую телевизорами, стиральными машинами, электрочайниками... Мощность, как произведение напряжения на ток, позволяет понять, что с уменьшением напряжения растет потребляемый ток при заданной мощности, а большой ток требует проводов большого сечения. Если бы напряжение в сети было 12 В, то толщина проводов нашей проводки в квартире была бы сравнима с толщиной труб водопровода, а к нашим домам подходили бы провода...

Вернемся к схеме сетевого адаптера. Диод, как все догадываются, выпрямляет переменное напряжение. Однако диаграмма показывает, что напряжение остается переменным. Оно меняется по величине, хотя амплитуда изменений не очень большая. Чтобы уменьшить эту амплитуду (эту часть напряжения обычно называют пульсации выпрямленного напряжения) после диода ставят «сглаживающий» конденсатор. Увеличьте емкость этого конденсатора.

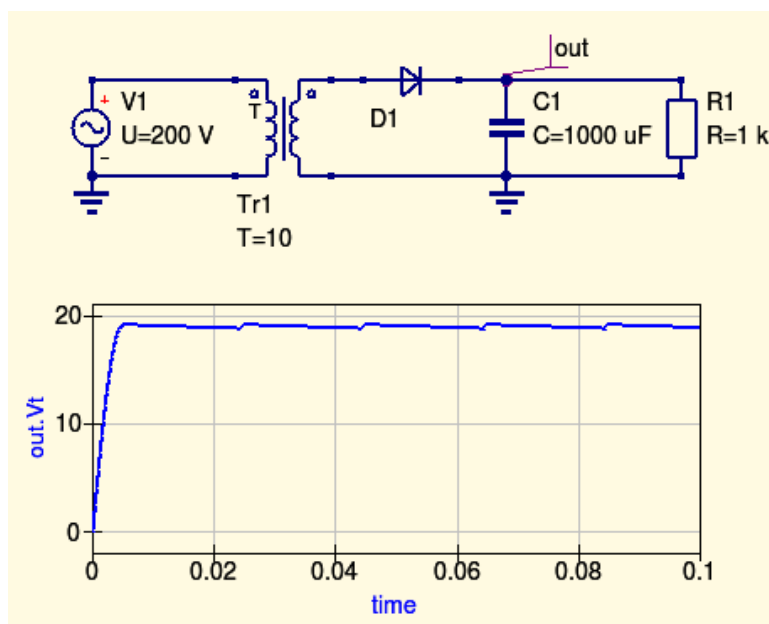


Рис.8.2. Выпрямленное напряжение при увеличении емкости конденсатора

При том токе, что потребляет сопротивление нагрузки R1, такое напряжение вполне выглядит, как постоянное. В расчетах емкость конденсатора определяют по току потребления, чем он больше, тем больше должна быть емкость. Если, конечно, нам нужно напряжение, которое имеет низкие пульсации. Для питания лампочки накаливания это совсем не обязательно, а микрофонный усилитель будет очень чувствителен к этим пульсациям.

Во многом схема сетевого адаптера, изображенная выше, идеализирована. При создании собственного блока питания необходимо учесть, что желание применить сглаживающий

конденсатор как можно большей емкости, а сейчас есть малогабаритные электролитические конденсаторы, желание получить наилучшее по качеству напряжение на выходе выпрямителя следует соизмерить с параметрами схемы.

На рисунках выше переменное напряжение от источника начинает «свое путешествие» с нулевого напряжения, а скорость изменения напряжения определяется синусоидальным законом, в реальных условиях вы никогда не знаете мгновенного напряжения в розетке при включении адаптера в сеть. Выше мы говорили о пользе испытательных сигналов. Применим испытательный сигнал к схеме адаптера, где идеальный диод я заменю реальной моделью, а сигнал, примерно, будет соответствовать ситуации, когда при подключении адаптера в сети напряжение достигает максимума.

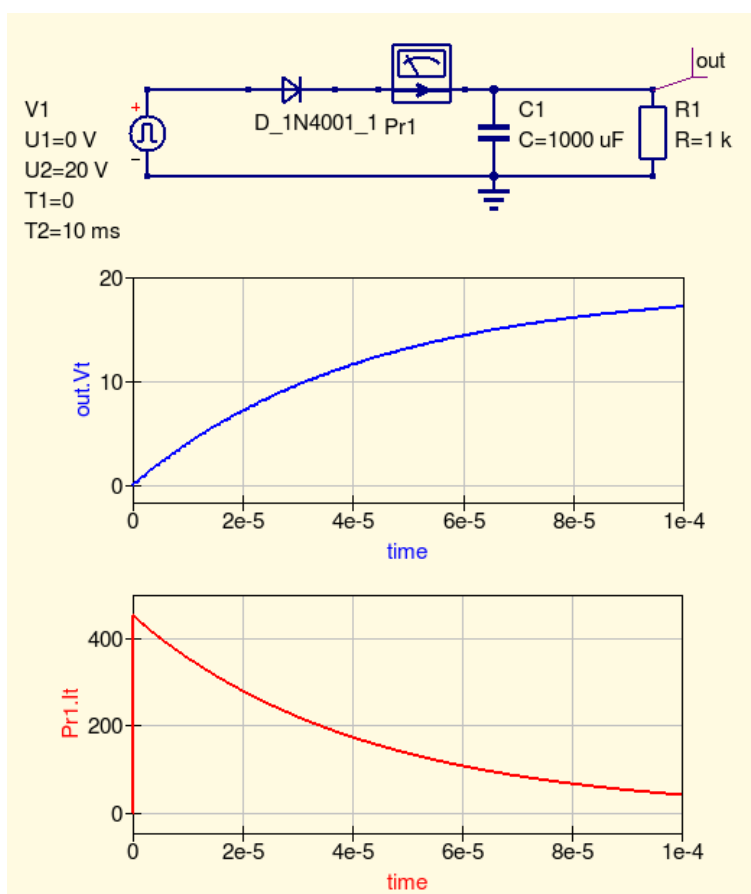


Рис. 8.3. Испытание адаптера импульсным генератором

Нижняя диаграмма показывает ток через диод (Pr1). Не думаю, что в реальной схеме он будет достигать величины в 400 А, он будет ограничен в первую очередь активным сопротивлением выходной обмотки трансформатора, но при его достаточной мощности это небольшое сопротивление, несколько ом, скажется и индуктивный характер трансформатора, но что ток может оказаться выше, чем потребляемый сопротивлением 1 кОм при напряжении 20 В, это точно. Всплеск тока будет кратковременным, его трудно увидеть с помощью мультиметра и даже осциллографом, но его приходится учитывать при выборе диода, иначе тот может «сгореть».

Еще одна особенность простейшего блока питания связана с напряжением в сети. 220 В — это расчетное напряжение в сети. Измеряя это напряжение в разное время суток вы можете обнаружить, что оно меняется от 200 В до 240 В (если электроснабжение качественное). А это означает, что на выходе блока питания вы можете наблюдать напряжение, скажем, от 18

до 22 вольт. Если ваше устройство не огорчают подобные «выходки» питающего напряжения, то и вам не о чем беспокоиться. Иначе следует стабилизировать выходное напряжение.

Простейшее устройство для стабилизации напряжения — диод. Специализированный диод, который называется стабилитрон, при обратном включении до определенного напряжения ведет себя, как любой другой диод, практически, не пропуская тока. Но при достижении напряжения, которое называют напряжением стабилизации, начинает интенсивно пропускать ток. Обычно стабилитрон включают в паре с токоограничительным резистором, на котором «падает» избыток напряжения, что позволяет поддерживать «напряжение стабилизации» почти неизменным.

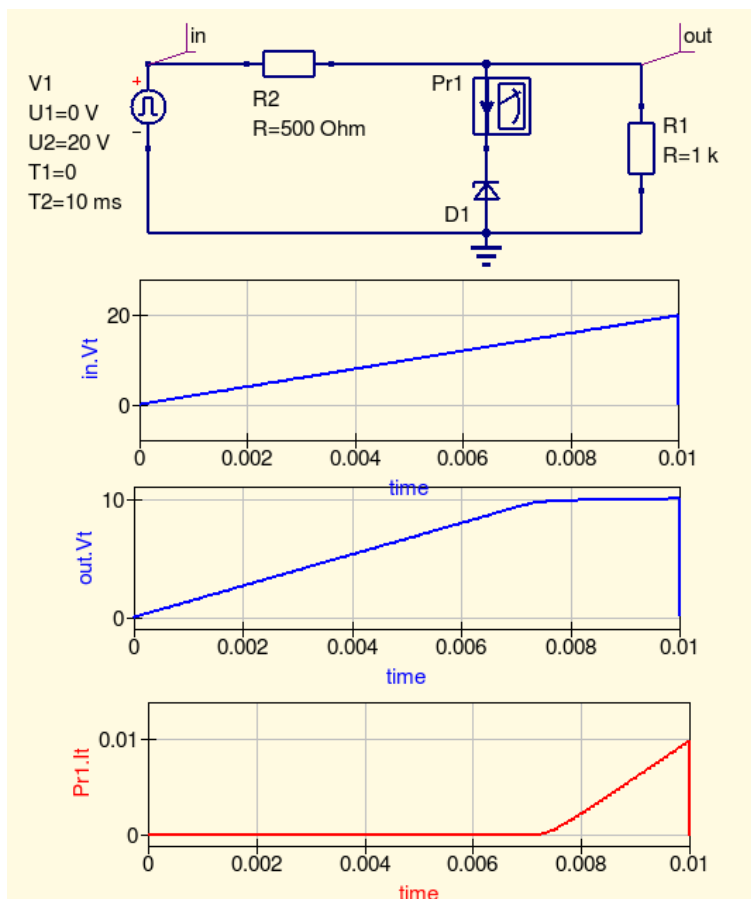


Рис. 8.4. Испытание стабилитрона плавно растущим напряжением

Программа позволяет «превратить» генератор прямоугольных импульсов в генератор треугольного испытательного сигнала, напряжение которого изменяется, как показано на верхней диаграмме. Напряжение на стабилитроне меняется соответственно следующей диаграмме, а ток через стабилитрон (Pr1) показан на нижнем графике.

В таком простейшем стабилизаторе ток через стабилитрон и ток в нагрузке соизмеримы, что ограничивает область применения простейшей схемы стабилизатора. Для нагрузки, где требуются большие токи, применяют стабилизаторы, основанные на способности транзистора усиливать ток. Стабилитрон ставят в базовой цепи транзистора, а нагрузку в коллекторную или эмиттерную цепь транзистора. Применяют и более сложные схемы, скажем, компенсационные схемы стабилизаторов. Но сегодня для радиолюбителей этот вопрос решается проще — есть микросхемы стабилизаторов напряжения, например, серии КР142ЕН (их в просторечии называют «кренками»), их следует использовать. Микросхема

позволяет получить не только напряжение на выходе, соответствующее паспортному напряжению стабилизации, но с двум-тремя дополнительными элементами и напряжения выше.

Современные блоки питания многих бытовых электронных устройств используют преобразователи напряжения. Такая конструкция позволяет либо не использовать трансформаторы, либо использовать при той же мощности трансформаторы, работающие на частоте гораздо выше 50 Гц, чьи габариты и вес, а они у трансформатора зависят от частоты, значительно меньше. Телевизоры и мониторы стали легче, зарядные устройства для мобильных телефонов вполне соответствуют им и по габаритам, и по весу.

Преобразователи напряжения удобно использовать и в переносных устройствах, если требуется напряжение большее типового напряжения батареек и аккумуляторов. Простейший преобразователь напряжения — это генератор. И выгоднее использовать генератор прямоугольных импульсов.

В чем выгода использования генератора прямоугольных импульсов?

Вот простая схема, где R2 — сопротивление нагрузки, а T1 — выходной транзистор.

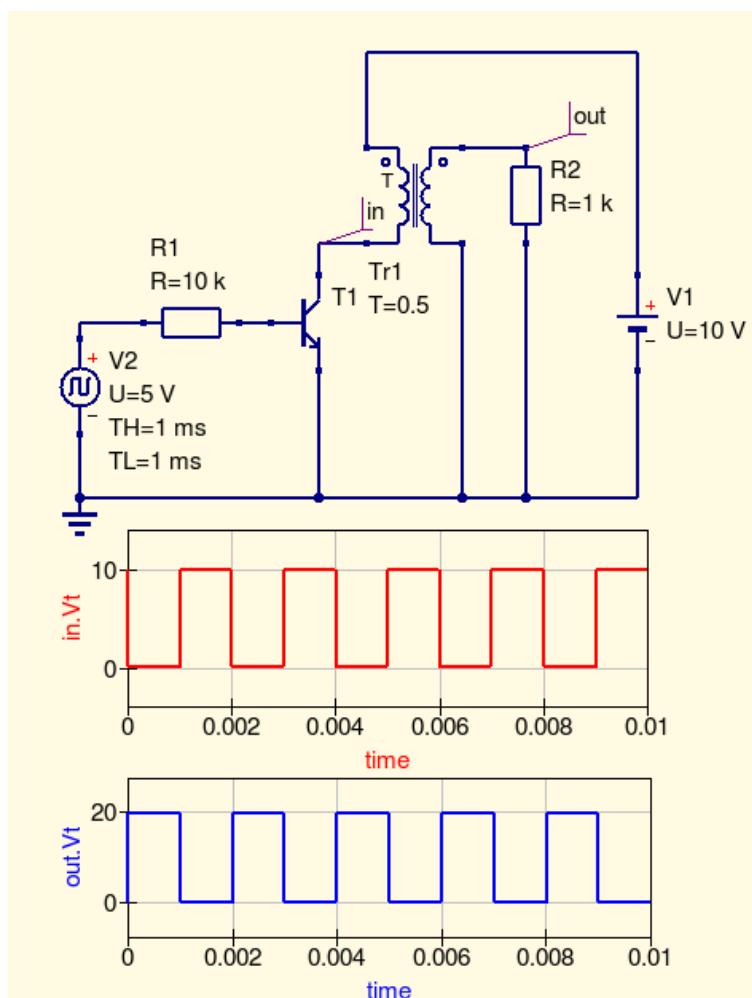


Рис. 8.5. Применение мультивибратора для преобразования напряжения

При работе на транзисторе T1 выделяется мощность в виде тепла. Это тепло нужно отводить, чтобы не перегреть транзистор, это тепло снижает коэффициент полезного действия преобразователя. На диаграмме видно, что транзистор, подключенный к

мультивибратору (источник прямоугольного напряжения V2), работает в «ключевом режиме», он либо полностью включен, либо полностью выключен (верхняя диаграмма). Когда транзистор полностью выключен, напряжение на нем большое, но ток очень мал, когда транзистор полностью включен, ток большой, но напряжение очень маленькое, то есть, в обоих случаях мощность рассеивается небольшая. Значительно больше мощности рассеивается в моменты перехода транзистора из одного состояния в другое, то есть, на фронтах импульсов. Но эта мощность существенно меньше, чем в том случае, когда генератор синусоидальный. Схемы, как правило, двухтактные, но существо дела это не меняет.

В последнее время применяют конвертеры другого типа. Например, такого:

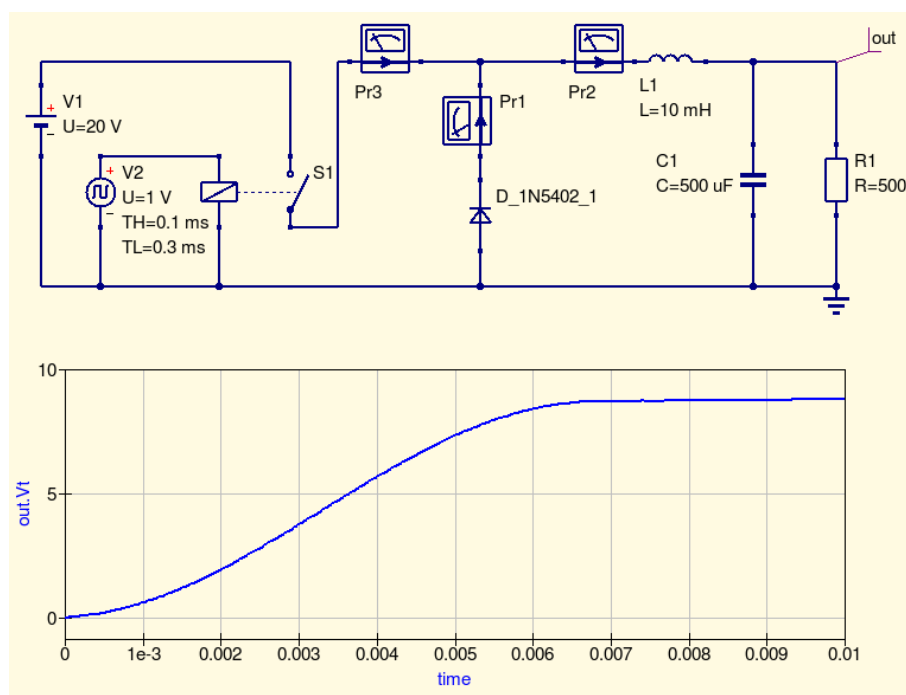


Рис. 8.6. Конвертер постоянного напряжения в постоянное

Напряжение от источника питания 20 В в данном случае преобразуется в напряжение 8 В. В зависимости от параметров этой схемы оно может быть преобразовано в большее напряжение. Реле S1 применено для моделирования процесса в Qucs вместо обычно применяемых полевых транзисторов. Правильно рассчитанный преобразователь получается достаточно легким и компактным.

Зависимость выходного напряжения от скважности импульсов генератора может использоваться для стабилизации выходного напряжения, если управлять скважностью импульсов с помощью обратной связи. Заменим параметры генератора V2, сделав TH=TL=0.2ms.

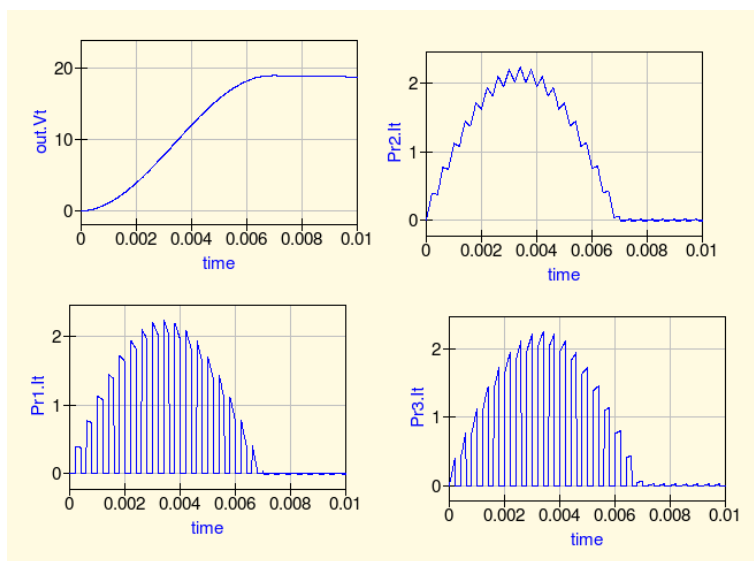


Рис. 8.7. Диаграммы работы предыдущей схемы с изменением скважности импульсов

Первая диаграмма показывает, что выходное напряжение увеличилось почти до 20 В.

Несколько видоизменив схему, мы можем получить напряжение обратной полярности, как показано ниже.

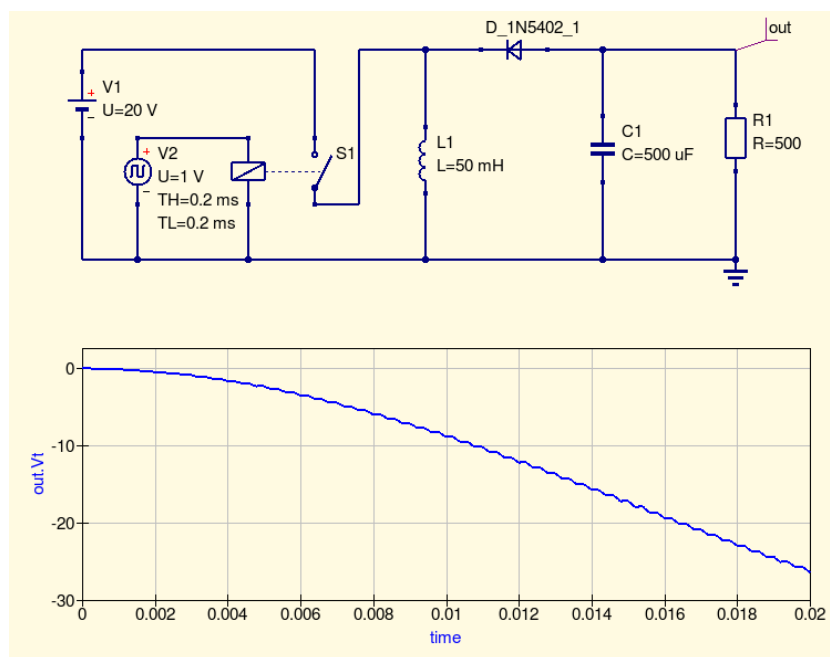


Рис. 8.8. Преобразователь с выходным напряжением обратной полярности

Источники питания — очень важный компонент любого электронного устройства. От их надежности и правильной работы зависит очень многое, поэтому при возникновении проблем в первую очередь следует убедиться в наличии и качестве питающего напряжения. Чем сложнее устроен блок питания, тем труднее без специальных приборов оценить, все ли в порядке с питанием. Начинающим любителям я бы посоветовал использовать такие источники питания, качество которых легко проверить с помощью мультиметра. Это сэкономит много времени и сил.

О программе Qucs и других программах

Я не уверен, что сегодня при разработке микропроцессора возможно, хотя бы в принципе, создать его прототип на макетной плате из транзисторов, резисторов, конденсаторов и т.п.

Но если в современных разработках важным инструментом становится компьютер, то овладеть работой с программами моделирования электрических цепей, может оказаться не менее важной задачей, чем умение читать схемы или рассчитывать их.

Любой школьный радиокружок ставит своей целью не только занять время школьников, но и дать им представление об очень важной и интересной сфере деятельности, возможно, будущей профессии сегодняшних школяров. Я не уверен, что каждая школа, хотя многие уже имеют компьютеры, может позволить себе покупку программ, позволяющих разрабатывать электронные устройства любой сложности и безупречного качества. Демонстрационные версии программ, как правило, имеют достаточно много ограничений, и бывает очень обидно, когда в процессе работы сталкиваешься с невозможностью расширить проект, который близится к завершению. Можно, конечно, вернуться к макетной плате и дорабатывать проект «живьем», но неприятный осадок останется.

Программа Qucs это полная свободно распространяемая версия, существующая как для операционной системы Linux, так и для Windows. Установка программы в Linux при наличии Интернета для многих дистрибутивов, таких как Fedora, Debian, Ubuntu, ALTLinux сводится только к выбору этой программы из списка доступных программ. Для работы в Windows потребуется загрузка основной программы и вспомогательной, позволяющей запускать программу в Windows. Файлы, необходимые для установки, доступны для скачивания даже при модемном соединении, поскольку занимают чуть больше 5 Мбайт. Для сравнения, демонстрационная версия Multisim, когда я в последний раз пользовался ею, требовала при каждом запуске докачки 50 Мбайт.

Немаловажно для начинающих то, что программа позволяет выбрать с десятков языков интерфейса и подсказки, включая русский. По мере освоения программы и накопленного опыта можно использовать средства, предоставляемые программой, для расширения, скажем, элементной базы. Многих интересует работа с отечественными деталями, а не импортными. Если у вас есть необходимые данные для создания модели, то вы можете расширить библиотеку компонент. А, если вы умеете и любите программировать, то, используя исходные коды программы, а они тоже доступны, вы можете создать свою, более мощную программу.

Все, что касается проекта, можно найти на сайте:

<http://qucs.sourceforge.net/>

У меня программа давно установлена, поэтому после первого запуска на вкладке проектов появляется длинный список этих проектов. Хранится он (напомню, что я пользуюсь Linux) в моей домашней папке в скрытых файлах. Чтобы их увидеть достаточно в пункте основного меню менеджера файлов выбрать пункт «Показывать скрытые файлы».

Просматривать файлы проектов в менеджере файлов особой нужды нет, но полезно помнить, что когда вы сохраняете свои документы, то кроме явно видимых папок есть много нужных (или нет, вам решать), скрытых от глаз в первую очередь по причине их общего количества, служебных файлов. Я не работаю в программе Qucs, но часто рассказываю о ней или об электронике с помощью этой программы. В результате накопилось множество проектов, о содержании которых я едва ли помню. Однако, когда в предыдущей главе мне понадобилась схема преобразователя, я первым делом стал перебирать все старые проекты и

схемы, нашел нужную и это избавило меня от необходимости искать оригинал и рисовать схему, подбирать параметры и т.д.

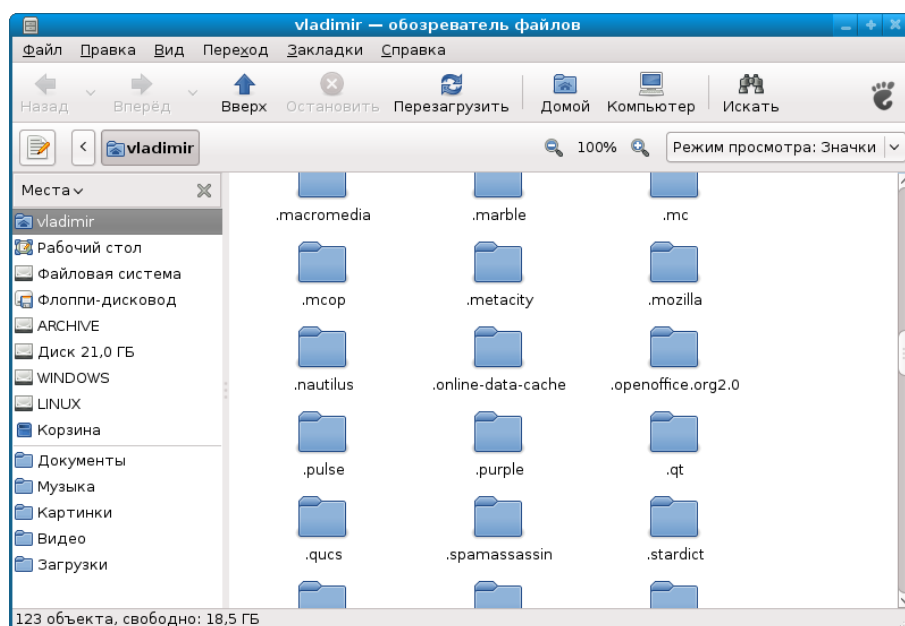


Рис. 9.1. Вид скрытой папки проектов Qucs в Nautilus

А вот как выглядит программа Qucs после запуска.

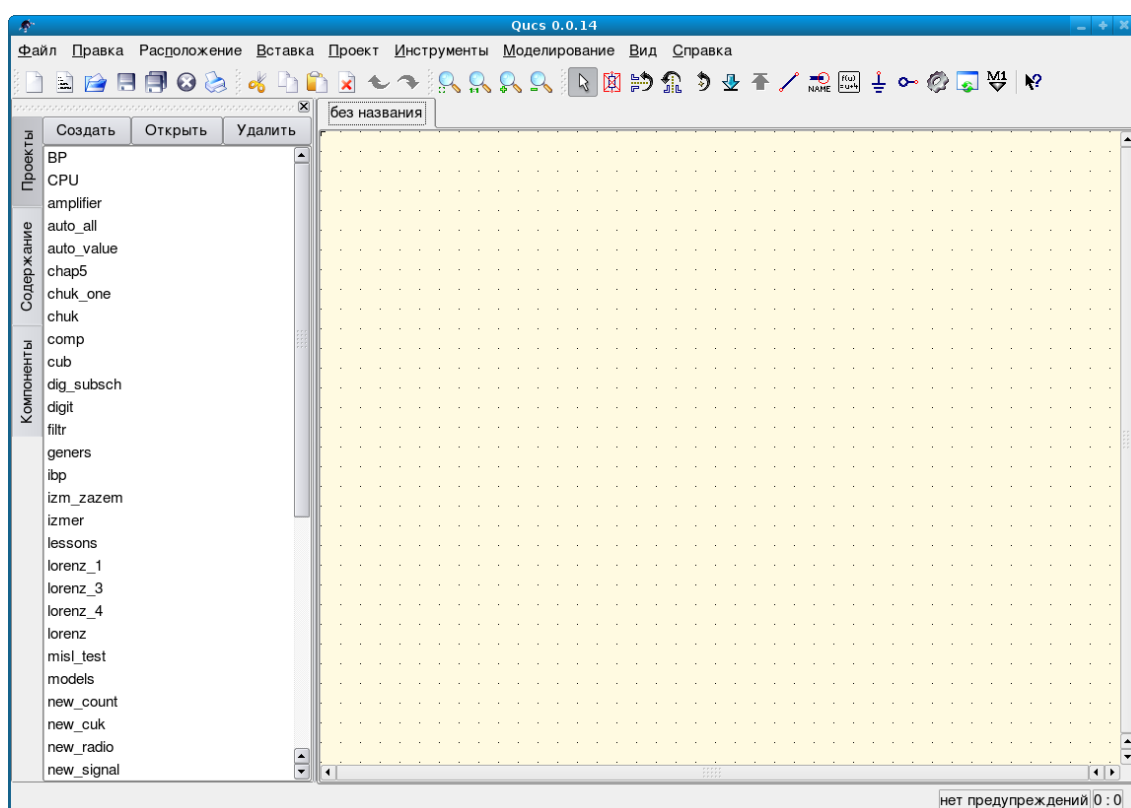


Рис. 9.2. Программа Qucs в Linux

Длинный список проектов в левом окне, открытом на закладке *Проекты*, позволяет с помощью кнопок в верхней части окна: **Создать**, **Открыть**, **Удалить**, — управлять этим

списком проектов, приводя его к нормальному виду. Для сравнения я хочу добавить вид запущенной программы в Windows.

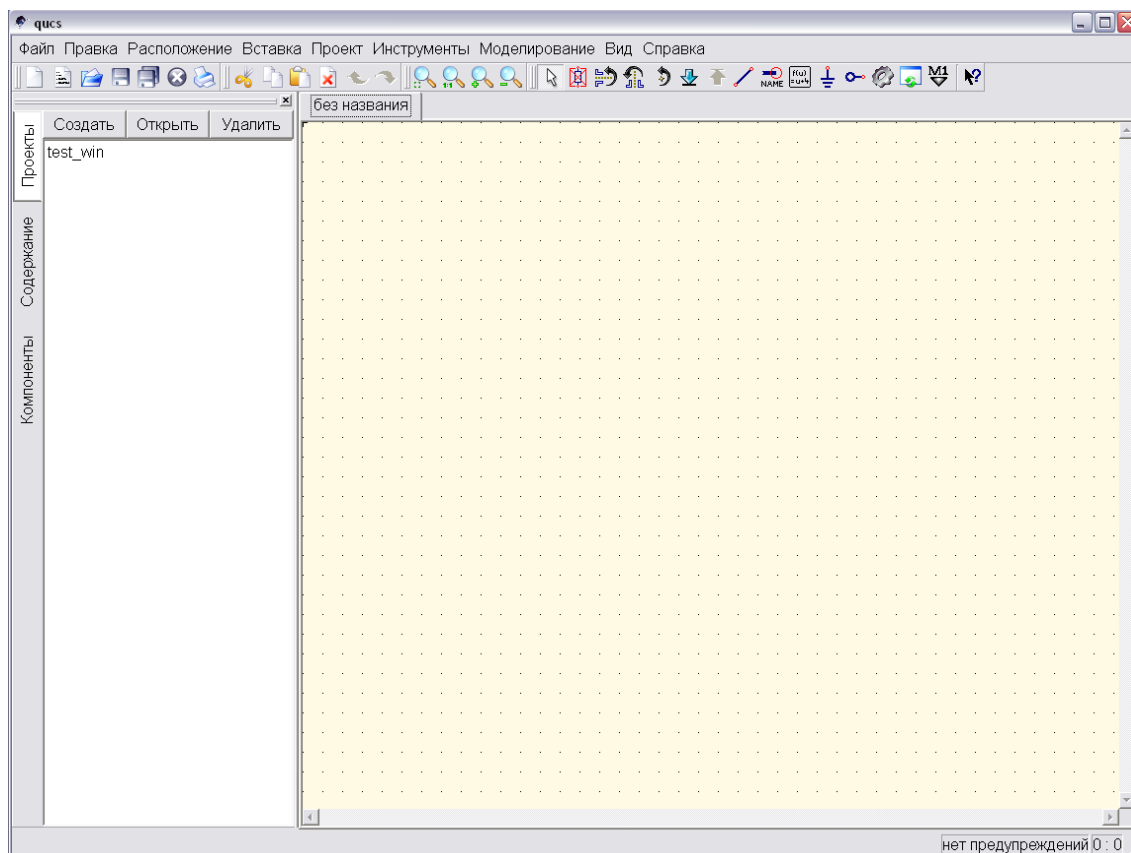


Рис. 9.3. Программа Qucs в Windows

Как видите отличий, практически, нет. Ниже вкладки *Проекты* в левом окне вкладка *Содержание*, подразумевающая содержание открытого проекта и вкладка *Компоненты*.

Если вы только установили программу, ни проектов, ни тем более их содержания пока нет. Вы можете выбрать вкладку *Компоненты* и начать собирать схему, но я советую создать проект с помощью кнопки **Создать**. Нажав на нее вы попадаете в диалоговое окно, в котором пишете имя проекта.

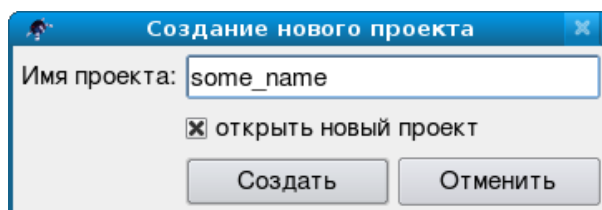


Рис. 9.4. Первый шаг при создании проекта

После того, как имя проекту придумано и вписано, кнопка **Создать** нажата, а вы готовы приступить к сборке схемы, вы оказываетесь на вкладке *Содержание*. В этом месте появятся все файлы со схемами проекта, которые вы будете использовать. В данный момент наибольший интерес представляет следующая вкладка *Компоненты*. Не только по причине отсутствия в содержании проекта хотя бы одного файла, но и по причине интереса, а что есть вообще в программе такого, с чем есть смысл попробовать работать.

Открывается вкладка на дискретных компонентах.

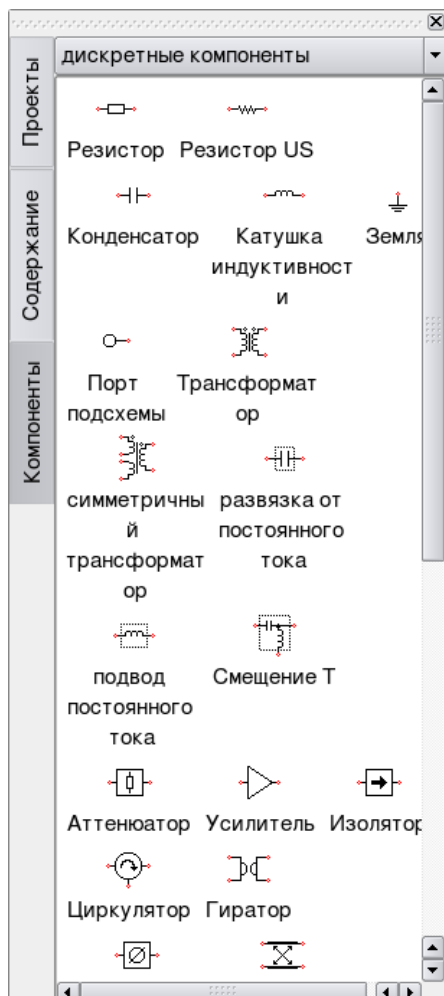


Рис. 9.5. Дискретные компоненты программы Qucs

Полоса прокрутки справа и свободное поле ниже показывают, что это далеко не все дискретные компоненты программы. Кнопка с иконкой X в правом верхнем углу подразумевает, что это окно можно закрыть, когда не хватает места для рисования схемы. Но самой интересной кнопкой я считаю кнопку с иконкой ▼ правее надписи *дискретные компоненты*. Она открывает список всех категорий компонентов программы.

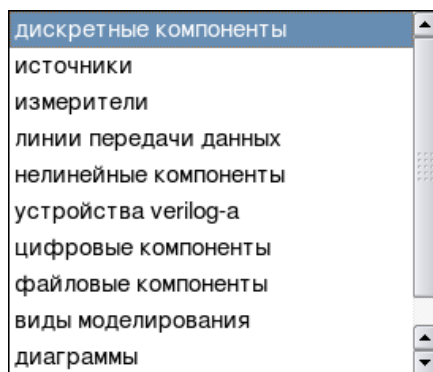


Рис. 9.6. Список категорий компонентов программы Qucs

Проект Qucs развивается, пополняется компонентами, но этот список меняется мало. Следующий в списке раздел источников не менее объемистый, чем раздел дискретных компонентов. Кроме привычных любому из нас батарейки, генератора синусоидального напряжения (источник переменного напряжения) или прямоугольных импульсов, он содержит очень полезные инструменты моделирования, такие, как источника тока, управляемые током, или источник тока, управляемый напряжением. В пользу этих компонентов можно убедиться, если обратиться к Workbook, где описано, как создать модель смешанного, аналого-цифрового компонента, таймера 555.



Рис. 9.7. Источники в программе Qucs

Любой компонент можно перенести в рабочее поле чертежа, щелкнув левой клавишей мышки по компоненту в этом окне. После перемещения курсора контур элемента помогает расположить его в нужном месте, при необходимости можно, щелкнув правой клавишей мышки, повернуть элемент, а щелчком левой клавиши мышки оставить его на чертеже. Часто количество резисторов, например, не ограничивается одним, не прерывая процесса можно расставить столько резисторов, сколько нужно. Этот механизм работает со всеми элементами раздела компоненты, но все они нужны во многих количествах. Скажем, вида моделирования бывает достаточно одного. Для прекращения повторного копирования достаточно щелкнуть по другому компоненту или нажать клавишу **Esc** на клавиатуре, или нажать на клавишу с иконкой курсора на инструментальном меню программы. Иногда клавиша **Esc** сбрасывает у меня режим установки компонента, но переводит программу в режим работы с курсором. В этом случае помогает обращение к инструментальной панели.

Содержание окна *Компоненты* может меняться не только при выборе из списка, после удачного завершения моделирования открывается перечень компонентов, относящихся к видам моделирования, чтобы вы могли выбрать, в каком виде вам нужны результаты: табличном, графическом, таблицы истинности, если моделировалась цифровая схема.

Основное меню программы начинается, как обычно принято, с команд работы с файлами.

Создать	Ctrl+N
Новый текст	Ctrl+Shift+V
Открыть...	Ctrl+O
Заккрыть	Ctrl+W
Сохранить	Ctrl+S
Сохранить все	Ctrl++
Сохранить как...	Ctrl+-
Напечатать...	Ctrl+P
Печать по размеру страницы...	Ctrl+Shift+P
Настройки документа...	Ctrl+.
Изменить обозначение схемы	F9
Настройки программы...	Ctrl+,
Выйти	Ctrl+Q

Рис. 9.8. Содержание раздела работы с файлами

Из представленных возможностей отмечу только несколько, остальные должны быть всем знакомы: *Настройки документа...*, *Изменить обозначение схемы*, *Настройки программы...*

Настройки документа.

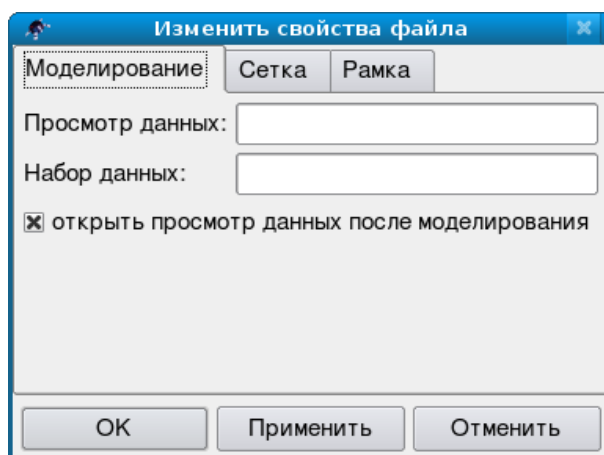


Рис. 9.9. Диалоговое окно настройки документа

Настройки, как это видно, относятся к моделированию, к настройке и видимости сетки и содержанию рамки. Если вы любите тщательную работу и распечатываете свои схемы, сопровождая их спецификацией и описанием, не лишне будет заполнить штамп, чтобы не гадать, какая из схем была исправлением предыдущей.

Я очень часто использую вставку картинок из этой программы в текст, как в этом рассказе, если не забываю, то в этом диалоге отключаю сетку.

Изменить схему. Этим можно воспользоваться при создании своего компонента после добавления всех составляющих. В итоге вы получите общепринятое изображение в виде прямоугольника с соответствующими выводами. Добавленный компонент, как подсхема, может быть добавлен в другую схему. После выбора изменения схемы окно *Компоненты* переключается к разделу *Рисунки*.

Настройки программы.

Позволяет настроить такие элементы программы, как цвет фона документа, шрифт, выбрать язык интерфейса программы из списка доступных и ряд других свойств программы. Закладка *Редактор VHDL* относится к настройкам цветовой подсветки, если текст написан на VHDL.

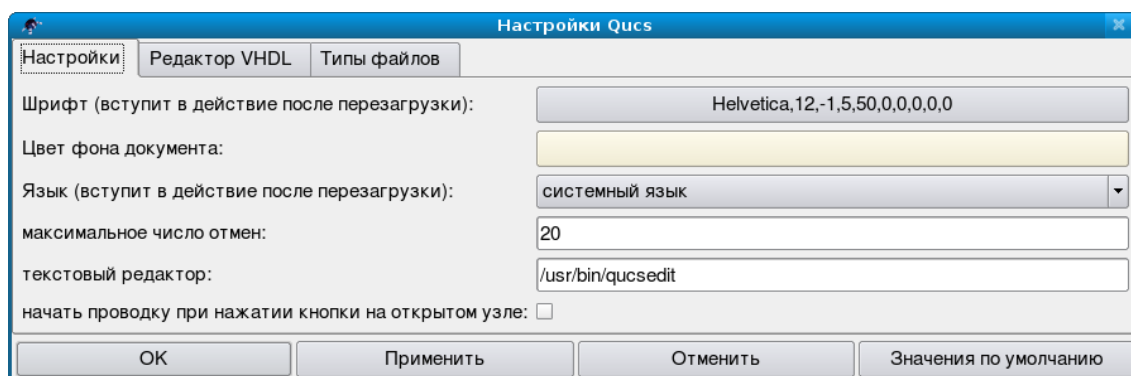


Рис. 9.10. Диалоговое окно настроек программы

Назначение разделов основного меню вполне понятно из названий, хочу отметить только, что в разделе Инструменты пункт Библиотека компонентов позволяет заменить идеальные компоненты моделями реальных.

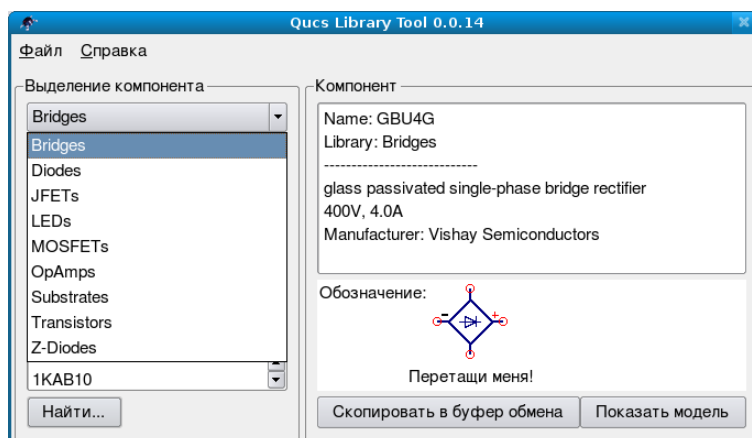


Рис. 9.11. Диалоговое окно выбора компонентов из библиотеки

Диодные мосты, транзисторы, операционные усилители и т.д. — все модели выполнены по данным реальных элементов. И последнее, что мне кажется важным для начинающих, раздел *Справка*.

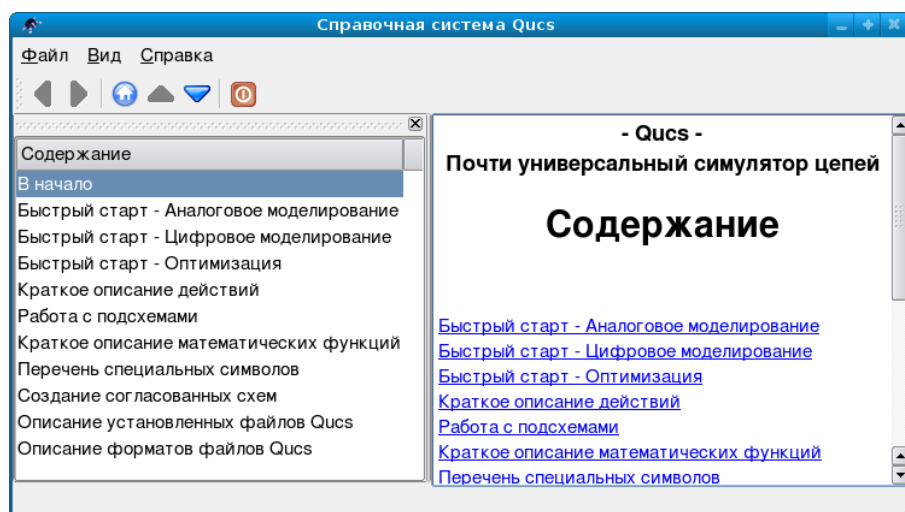


Рис. 9.12. Справка по программе Qucs

Добавлю еще, что любой компонент электрической схемы, позволяет изменить параметры, заданные по умолчанию, если двойным щелчком по элементу открыть окно его свойств. Например, свойства транзистора определяются таким длинным списком, что он не умещается полностью на экране.

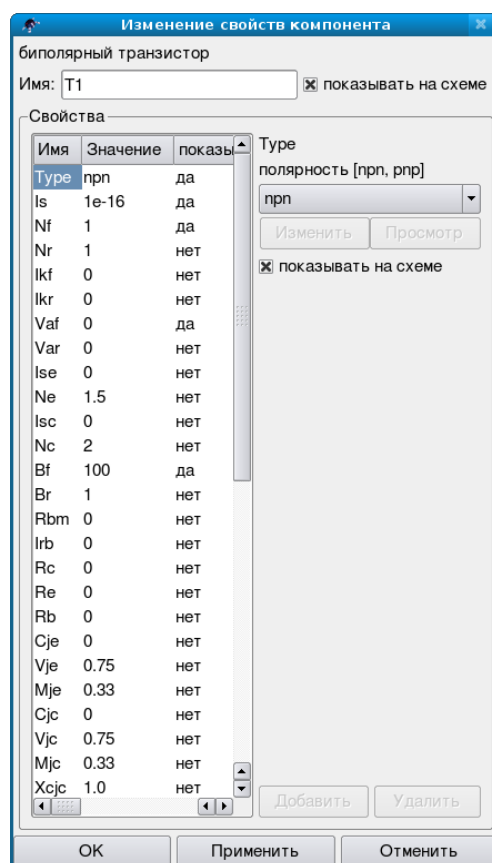


Рис. 9.13. Окно свойств транзистора

Правым щелчком мышки можно вызвать выпадающее меню, которое помогает в редактировании схемы.







Изменить обозначение схемы	F9
Настройки документа...	Ctrl+.
Переместить текст компонента	Ctrl+K
Выравнивать по сетке	Ctrl+U
 Скопировать	Ctrl+C
 Вставить	Ctrl+V
 Удалить	Del
 Повернуть	Ctrl+R
 Отобразить относительно оси X	Ctrl+J
 Отобразить относительно оси Y	Ctrl+M

Рис. 9.14. Выпадающее меню редактирования

Некоторые программы, используемые при обучении и в профессиональной практике, имеют русский интерфейс, частично или полностью. Например, хорошо русифицирована программа MicroCAP (<http://microcap.forum24.ru/>), есть отечественные аналоги оригинальных компонент. Но демонстрационная версия имеет ограничения, а полная версия достаточно дорогая.

Среди программ аналогичных Qucs, а я несколько раз делал беглый обзор таких программ, хочу отметить российскую разработку Fastmean.

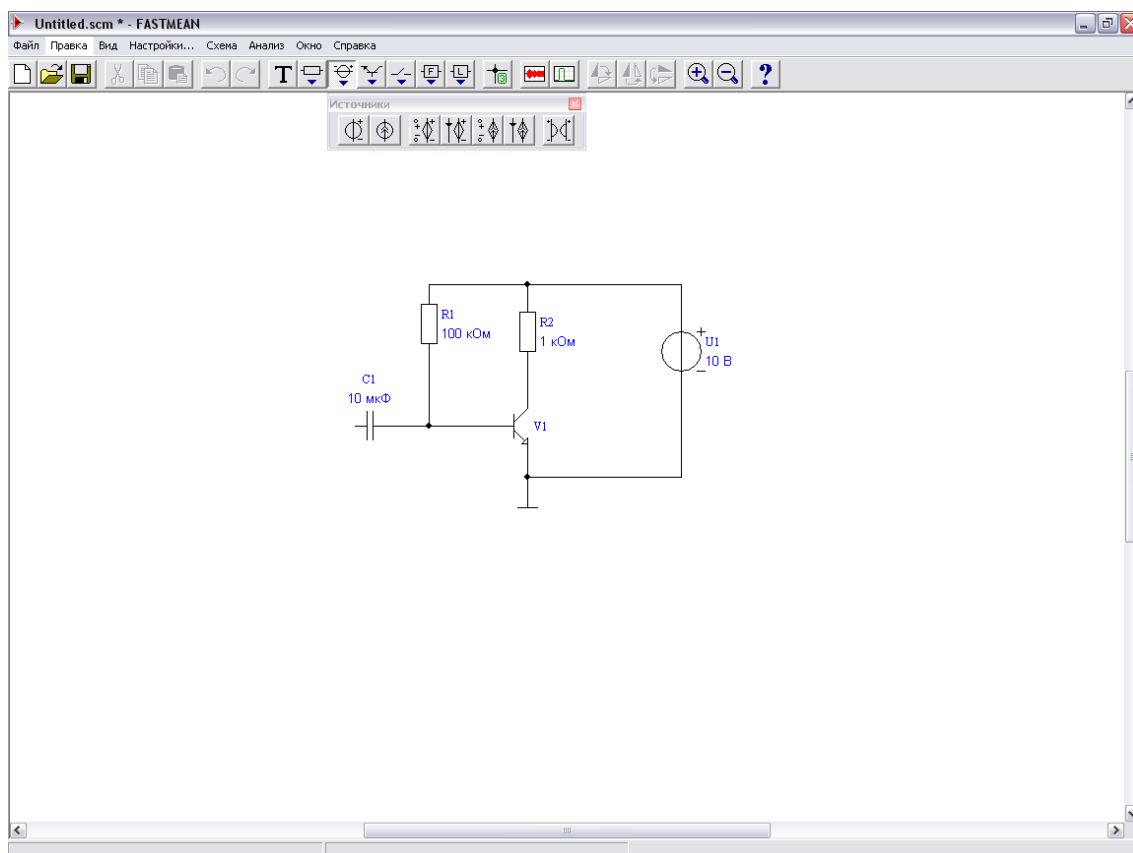


Рис. 9.15. Рабочее окно программы Fastmean

Интерфейс программы очень похож на EWB, программу многим знакомую, но интерфейс русскоязычный. Однако демонстрационная версия программы имеет ограничения, сильно

мешающие при работе со схемами, а по вопросам приобретения полной версии следует обращаться к разработчикам (<http://www.fastmean.ru/rus/>).

Заключение

Многие известные программы имеют тенденцию к интеграции всех радио и электронных компонентов в свою среду разработки. Так Proteus прекрасно может работать и с усилителем, и с микроконтроллером. Появились микроконтроллеры и в среде Multisim. Удобно ли это?

С точки зрения профессионалов, думаю специализированные программы и достаточны, и надежны: так те, кто занимается источниками питания, преобразователями и другими устройствами, связанными с переключением, предпочитают, например, SwCAD. Но радиолюбитель, лаборатория которого «воленс-ноленс» имеет и низко- и высокочастотные приборы, и цифровые генераторы, и программаторы для работы с ПЗУ и микроконтроллерами, мне кажется, должен включать в свой арсенал все программы, которые можно найти на рынке (в принципе, на любом). В отличие от профессионалов он редко специализируется на чем-либо, он не может передать решение своих проблем в соседний отдел. Для него — все средства хороши, что позволяют достичь цели.

Единственное, о чем не следует забывать радиолюбителю, достижение поставленной цели — это хорошо, но получить огромное удовольствие, а удовольствие, как основа увлечения, тоже цель, получить удовольствие можно от хорошо организованного процесса. Сам процесс в этом случае доставит много радостного и приятного. А получить какое-то устройство можно и через магазин, заодно сравнив его с изготовленным собственноручно. И далеко не всегда сравнение будет в пользу покупного.