

*Борис Аладышкин*

# **Простой способ управлять электропотребителями по сотовому телефону**

**Пошаговая инструкция, как самостоятельно собрать и  
настроить устройство управления нагрузкой по сотовому  
телефону**

**Copyright © 2012 Борис Аладышкин**

**Оформление – Андрей Повный**

**Все права защищены**

**<http://electrik.info/>**

Один мой знакомый обратился ко мне с такой просьбой: мол, хочется, чтобы в гараже (у него там мастерская) включался камин по звонку с сотового телефона. Ни больше, ни меньше! Разговоры о том, что это легче сделать, если купить [программируемое реле](#), его не убедили. Видимо, он мог в некоторые дни там не появляться, а это при программировании реле никакому учету не поддавалось. И кручился бы там у него счетчик, пока камин грел пустое помещение.

Вообще, век сотового телефона, как и персонального компьютера, крайне недолог: новые модели появляются как грибы после дождя, а старые просто валяются дома на полке. Хотя технически вполне исправны, можно позвонить, можно использовать как будильник, но модель не престижна и показываться с таким телефоном на людях просто неудобно. Поэтому такие бросовые телефоны находят применение, и достаточно широкое, в самодельных конструкциях управления электрическими нагрузками.

В настоящее время уже выпускается промышленным способом множество всяких устройств, управляемых по стандарту GSM (Global System for Mobile Communication), - цифровой стандарт мобильной связи или всем привычный теперь мобильник. Это охранная сигнализация, различные шкафы управления промышленным оборудованием, а то и просто отдельные розетки, как показано на рисунке 1.



Рисунок1.

Конструкция с виду представляет собой обычный переходник, который вставляется в настенную розетку. Нагрузка может включаться звонком или

передачей SMS по сотовому телефону. Также возможно ручное управление с помощью двух кнопок, расположенных на передней панели. Мощность, коммутируемая такими розетками, в зависимости от модели находится в пределах 1...5 Квт, что позволяет включать практически любую нагрузку.

Также производятся многоканальные розетки, наподобие компьютерной «клапты», позволяющие управлять независимо работой нескольких нагрузок. Такие устройства являются одним из элементов «умного дома», а стало быть, цена их достаточно высока: если поискать на просторах Интернета, то цены колеблются в пределах 2000...3500 и более рублей. К подобной розетке придется покупать дополнительную SIM – карту.

Много интересной практической информации о различных современных электротехнических новинках находится здесь - <http://electrik.info/main/news/>

Естественно, у моего знакомого таких денег просто не нашлось, а телефон–то уже валялся без дела. Да, видимо, не один он такой, если в интернете выложено достаточно большое количество схем, начиная от самых простейших одноканальных, до самых сложных и серьезных, управляющих несколькими каналами.

Еще дороже, чем отдельные розетки стоят промышленные модули. В качестве примера на рисунке 2 показано предложение Интернет –магазина о продаже модуля управления ДТМФ.

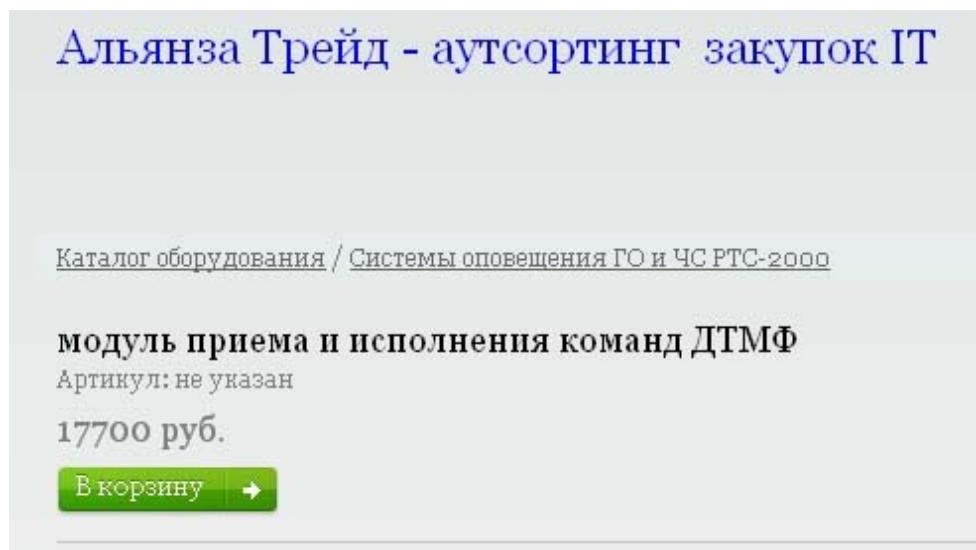


Рисунок 2.

Вот именно с этого рисунка и появилась на поверхности непонятная пока аббревиатура ДТМФ. Что это такое, посмотрим чуть ниже.

## Сигналы DTMF

В старых телефонах набор номера производился вращением диска: пальцем заводилась пружина номеронабирателя на нужное количество цифр, диск крутится назад, замыкая контакт, а в трубке прослушивались щелчки. Такой набор называли импульсным. Импульсный набор использовался и в современных аппаратах с кнопочным номеронабирателем.

В настоящее время используется, так называемый, тональный набор. Попробуйте по городскому телефону набирать номер, - в трубке прослушиваются звуки разной тональности. Это прослушиваются сигналы DTMF, - Dual-Tone Multi-Frequency, - двухтональный мультичастотный сигнал. На рисунке 3 показана таблица, по которой формируются цифры и некоторые знаки, передаваемые при наборе номера.

Частота	1209Гц	1336Гц	1477Гц	1633Гц
697Гц	1	2	3	A
770Гц	4	5	6	B
852Гц	7	8	9	C
941Гц	*/E	0	#/F	D

Рисунок 3.

Например, цифре «1» соответствует сочетание частот 697 и 1209 Гц, а цифре «9» 852 и 1477 Гц. Частоты подобраны таким образом, чтобы передаваясь вместе, они не образовали гармоник.

Для расшифровки тональных посылок существуют специализированные микросхемы - декодеры, например IL9270N, HM9270, MT8870. Это просто аналоги разных фирм. Они могут отличаться даже количеством выводов, или как теперь на иностранный манер пинов (от английского pin), но выполняют одни и те же функции.

Кроме этих специализированных декодеров сигналы DTMF могут быть расшифрованы на цифровых ЭВМ с использованием алгоритма Герцеля. Естественно, эти сигналы могут быть расшифрованы и с помощью микроконтроллеров или, как их иногда называют, встраиваемых ЭВМ.

Кроме набора телефонного номера DTMF – технология широко применяется в системах «умного дома», тревожных и охранных сигнализациях. Метки DTMF используются также в коммерческом радиовещании.

Система DTMF была разработана еще в 1961 году, но до России дошла лишь в девяностых годах прошлого века. Сначала тональный набор предоставлялся как платная услуга, и то не везде, поскольку тональный набор возможен лишь на современных цифровых телефонных станциях. А вообще, до сих пор во многих местах эксплуатируются допотопные релейные станции, которые позволяют пользоваться только импульсным набором номера.

А теперь, попробуйте проделать такой опыт: позвоните по сотовому, ну хотя бы своему коллеге по работе, ведь целый день находитесь в одной комнате. После того, как он «поднимет трубку» нажимайте на своем телефоне любые цифры: в динамике его телефона будут прослушиваться сигналы DTMF в виде коротких музыкальных звуков.

Согласно законам физики, музыкальными называются звуки, имеющие определенную частоту. Например, шум на улице, музыкальным звуком считать нельзя. Эти же звуки присутствуют и в динамике телефонной гарнитуры: дело за малым, - просто подключить дешифратор DTMF в разъем гарнитуры и вот, пожалуйста, готовое устройство управления.

### **Простые устройства управления по сотовому**

В некоторых случаях количество управляемых нагрузок всего-навсего одна, и требуется ее в любое время включить или выключить. Далее будут рассмотрены несколько простых устройств и одно из них выполнено просто на реле, причем всего на одном.

Схема опубликована на <http://dr-zeppelin.narod.ru/elektro/telefon.html>, а сама она показана на рисунке 4.

Несколько слов о работе схемы. Основой устройства является поляризованное реле. Как видно по схеме, оно имеет две катушки, включенные таким образом, что при подаче напряжения на одну катушку якорь реле притягивается к одному сердечнику, и остается в таком положении даже, если напряжения на катушке уже не будет, - там внутри реле есть магнитик.

Для того, чтобы отщелкнуть якорь в обратное положение, требуется подать напряжение, хотя бы импульс достаточной длительности и амплитуды, на другую катушку. Якорь останется в притянутом состоянии, даже при снятии напряжения питания. Не правда ли, очень напоминает RS – триггер?

Подробнее о назначении, устройстве и принципе действия триггеров, смотрите здесь: <http://electrik.info/main/praktika/296-logicheskie-mikrosxemy-chast-7-triggyer.html>

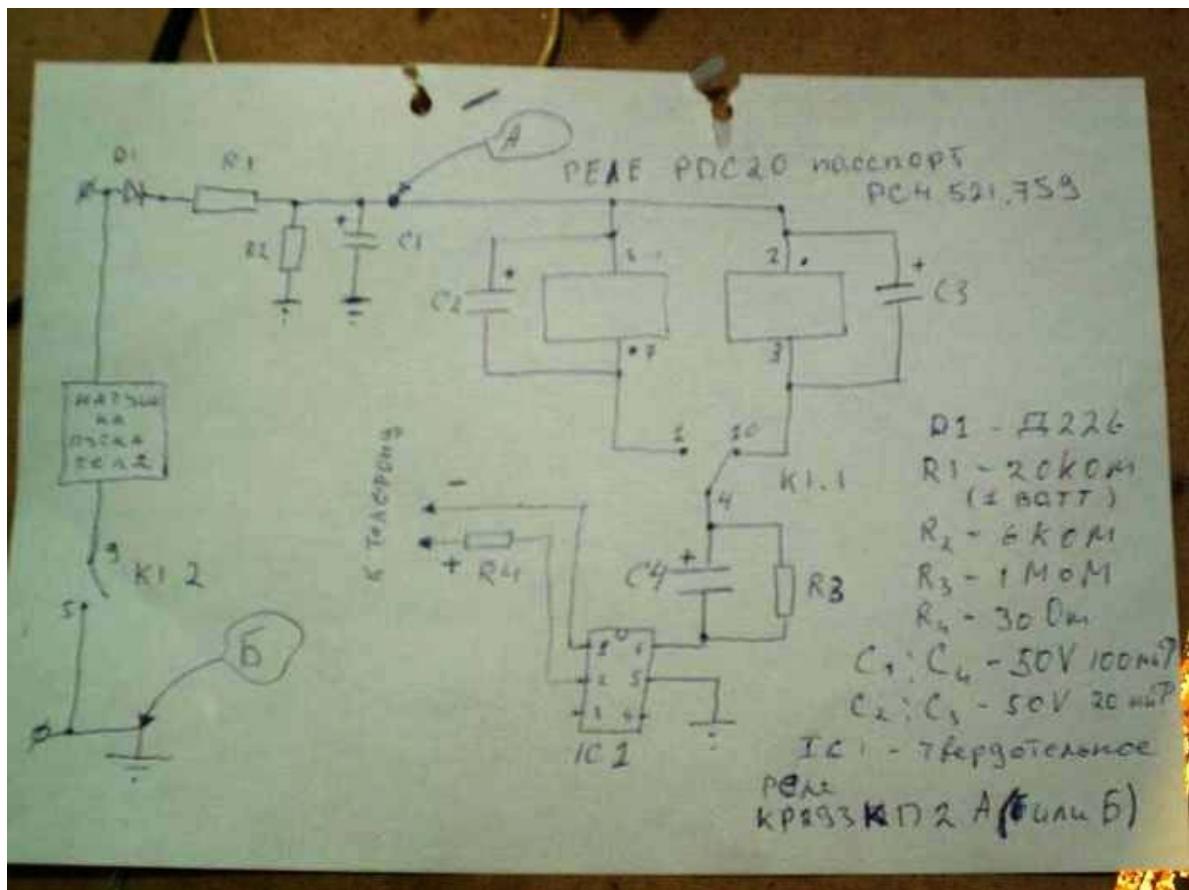


Рисунок 4.

Питание устройства осуществляется от сети, через однополупериодный выпрямитель D1, R1, R2, C1. На конденсаторе C1 получается напряжение около 24В. Конечно, так делается в нарушение всех правил техники безопасности, но автор уверяет, что если особо не наглеть, и не лезть, куда не надо, то... Ну, в общем все получится!

Телефон обязательно должен иметь виброзвонок: именно к его контактам будет присоединяться оптронное реле IC1, на схеме это резистор R4 и вывод оптрана 1. Полярность подключения указана на рисунке.

При подключении к телефону полярность напряжения на виброзвонке следует проверить с помощью мультиметра или светодиода с резистором. Когда сработает вибро, внутри оптрана открывается выходной транзистор, (выводы 5 и 6). Конденсатор C4 заряжается от источника питания через правую обмотку реле и открытый транзистор оптрана. Якорь реле переключается на левую катушку, и контактом K1.2 включает магнитный пускател, а контактом K1.1 подготавливает левую катушку к следующему переключению.

Конденсатор С4 разряжается через резистор R3 около пяти минут, за это время посылки с телефона состояния устройства изменять не будут. При всей очевидной простоте, устройство имеет один существенный недостаток: возможность достать экзотическое поляризованное реле, да еще нужного паспорта сейчас практически равна нулю. Об этом пишет даже сам автор схемы в своем описании.

Еще одно простое устройство управления показано на рисунке 5.

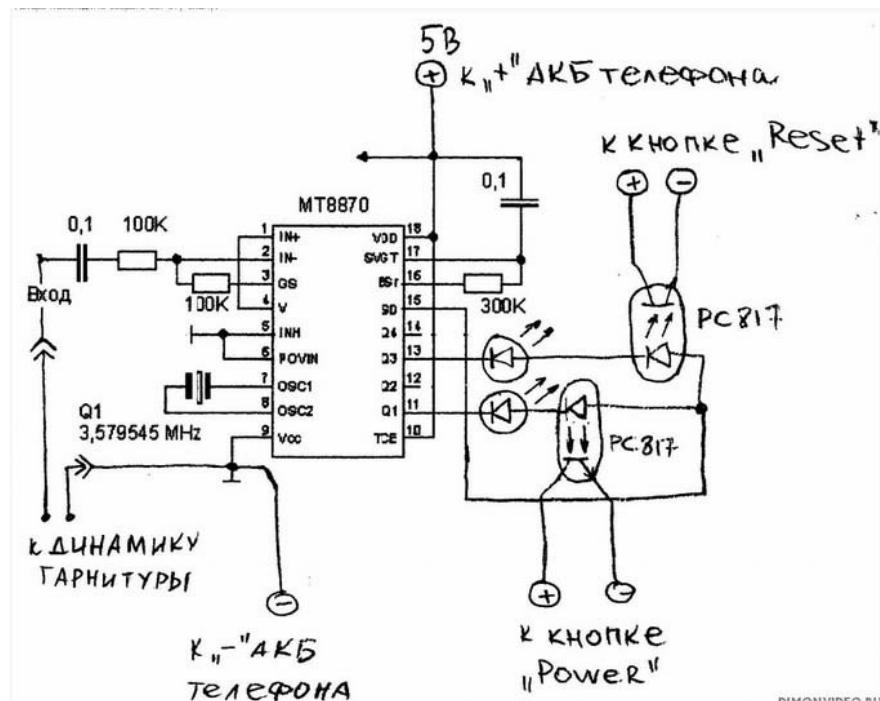


Рисунок 5.

Выполнено на специализированной микросхеме – декодере сигналов DTMF MT8870. Назначение данного устройства в авторском исполнении удаленное включение и перезагрузка компьютера. Работает устройство следующим образом. После того, как Вы позвонили на этот номер, после поднятия трубки набираете 1 или 2, что соответствует включению компьютера «POWER» либо перезагрузке «RESET».

Схема получает питание непосредственно от сотового телефона, выходные транзисторы оптронов подключены параллельно соответствующим кнопкам компьютера. Оптроны PC817 широко используются в импульсных источниках питания, - от компьютерных до зарядников мобильных телефонов.

Устройство подключается к разъему гарнитуры, к выводам динамика, на которых, как было рассказано выше, появляются сигналы DTMF. Основная

нездадча этой схемы при повторении в том, что телефон при подсоединении гарнитуры должен выполнять автоподнятие трубки. А такая опция есть далеко не у всех телефонов. Более подробную информацию об этом устройстве можно получить на сайте <http://dimonvideo.ru/articles/4486>

Еще две схемы устройств управления опубликованы на сайте <http://radioelectronika.ru/>. Первая из них показана на рисунке 6.

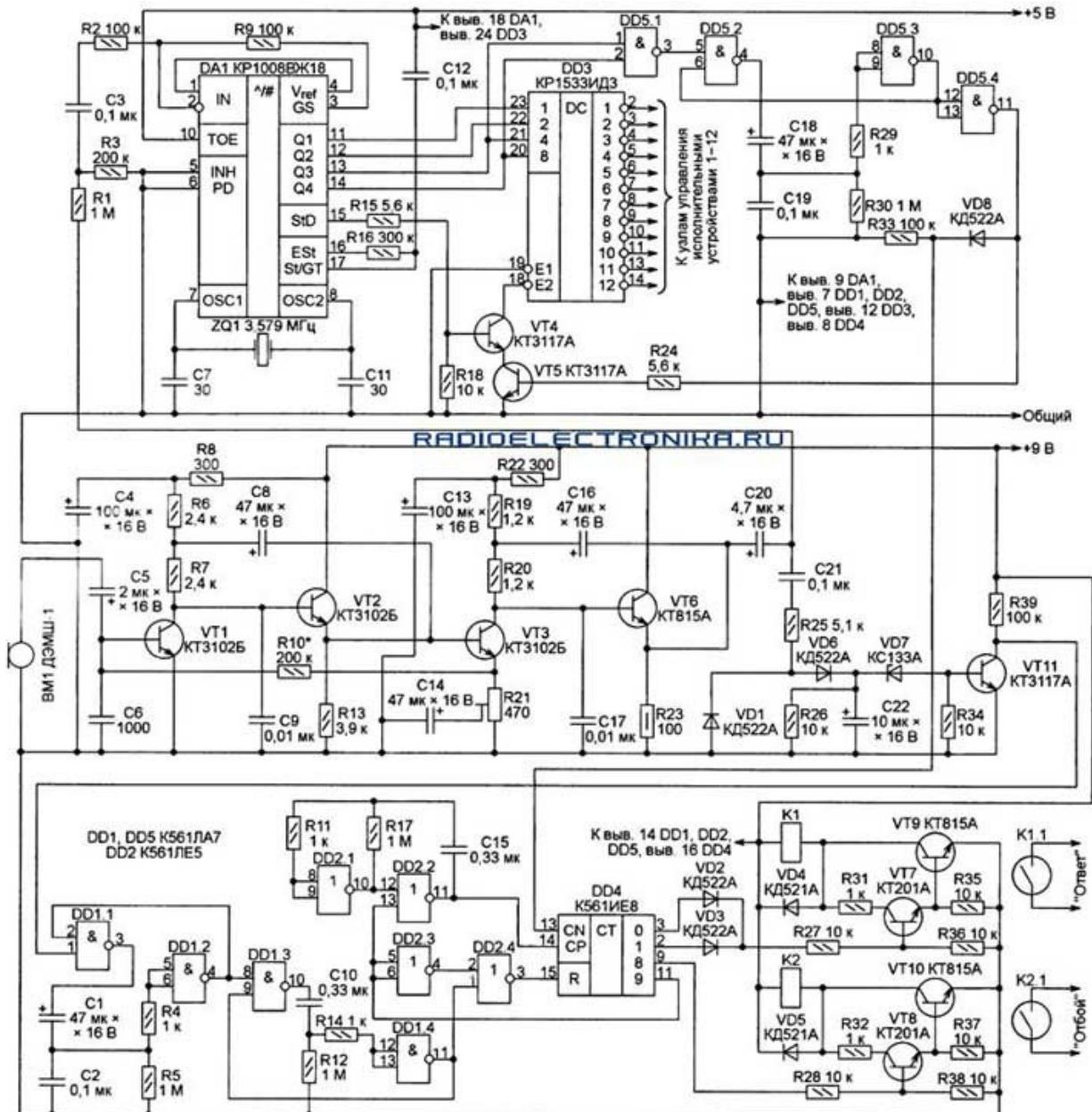


Рис. 1

Рисунок 6.

Схема выполнена на аппаратных средствах, т.е. не содержит микроконтроллеров, требующих программного обеспечения, вся логика работы достигнута за счет собственно схемы.

Звонок телефона принимается микрофоном, усиливается до нужного уровня усилителем, в результате чего срабатывает реле, контакты которого присоединены к кнопке «Ответ» (поднятие трубки). После срабатывания этого реле запускается выдержка времени около 7 секунд. Если за это время успеть нажать нужные клавиши, то сигнал DTMF будет подан на декодер DA1, выходные сигналы которого через дешифратор DD3 через реле могут подключить – отключить до 12 нагрузок. По истечении 7 секунд сработает реле «Отбой» (его контакты соединены с кнопкой « положить трубку»), для последующего управления понадобится еще один звонок. Таким образом, получается, что телефон, просто весь будет обмотан проводами: провода от реле к кнопкам да еще выход сигнала DTMF из разъема гарнитуры.

Более простая схема, имеется в виду по количеству деталей, показана на рисунке 7.

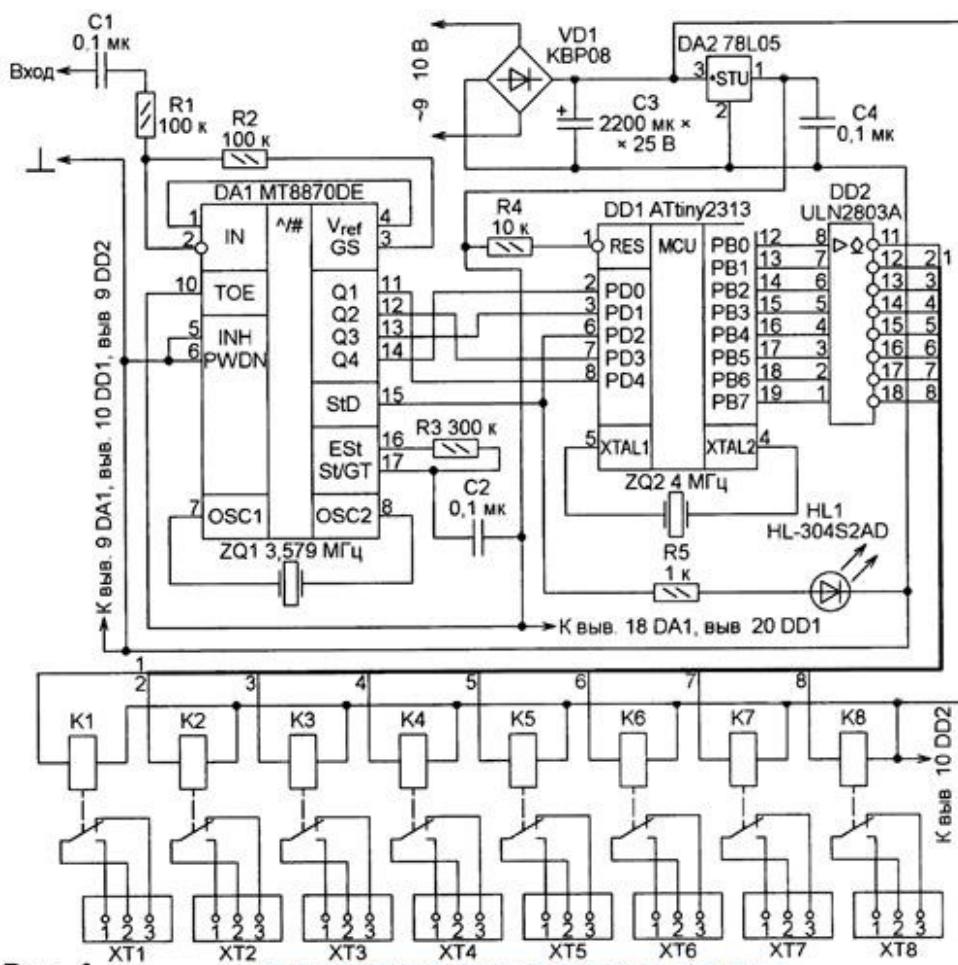


Рис. 4

Вот тут как раз используется телефон с автоподнятием трубки при подсоединенной гарнитуре, поэтому не требуется подпаиваться к кнопкам, достаточно только подключить разъем гарнитуры. Данная схема обеспечивает управление 8 нагрузками, команды управления показаны в описании схемы.

Но эти схемы вовсе не те, которые в начале статьи были названы самыми сложными и серьезными. Есть такие, которые вместо старого сотового телефона используют встраиваемый модуль GSM SIM300D. Его цена 4200 руб., правда он уже снят с производства. Именно в этот модуль и вставляется SIM – карта. Схема и описание устройства показаны на сайте <http://electromost.com/>.

Если у кого-то хватило терпения дочитать до этого места, то он уже, наверно, задумывается: «Так а что же автор, обещал показать свою собственную схему, и где же она?»

### **Описание принципиальной схемы включения камина по телефону**

Как было сказано в начале первой части книги, меня попросили сделать устройство для включения камина в гараже с помощью сотового телефона. Для ознакомления с вопросом было просмотрено некоторое количество подобных схем, (а вдруг удастся что-нибудь просто повторить чужое, уже готовое) обзор которых приведен выше.

Некоторые схемы обладали избыточными возможностями, ведь речь шла об управлении всего лишь одной нагрузкой, а не восемью или хотя бы четырьмя. Отпугнули также схемы, содержащие большое количество деталей, как, например показанная на рисунке 6. Другие требовали телефон с автоподнятием трубки при подключении к разъему гарнитуры.

Одна схема, содержала всего один транзистор с датчиком в виде катушки индуктивности и микроконтроллер. Микроконтроллер в той схеме выступал, кажется, только в роли цифрового фильтра, защищал от ложных срабатываний от помех.

Назначение схемы было просто в перезагрузке удаленного компьютера – сервера. При зависании просто нажимали кнопочку на телефоне, компьютер перезагружался и продолжал работать. Решено было идти примерно по такому пути, а именно, получать сигнал с телефона подобным образом, не путаясь в проводах, не подключаясь к телефону.

Собственно любая схема имеет свои достоинства и недостатки, поэтому не стоит думать, что разработка собственной схемы явилась в результате мыслей достаточно крамольных: «Мол наизобретали там не пойми чего, а все оказывается

так просто и моя схема будет лучше, чем у всех». Вовсе нет, таких мыслей не было, просто решено было сделать устройство из того, что уже имелось в наличии.

После нескольких опытов с индуктивным датчиком (до звукового сигнала вызова телефон излучает импульсы, которые сразу включают устройство), было решено использовать просто микрофон от телефонного аппарата, благо неисправный аппарат оказался под рукой. Все достаточно просто: зазвучал сигнал вызова, микрофон усилил, получили требуемый сигнал управления, запустили контроллер, включили нагрузку.

Принципиальная схема устройства управления показана на рисунке 8.

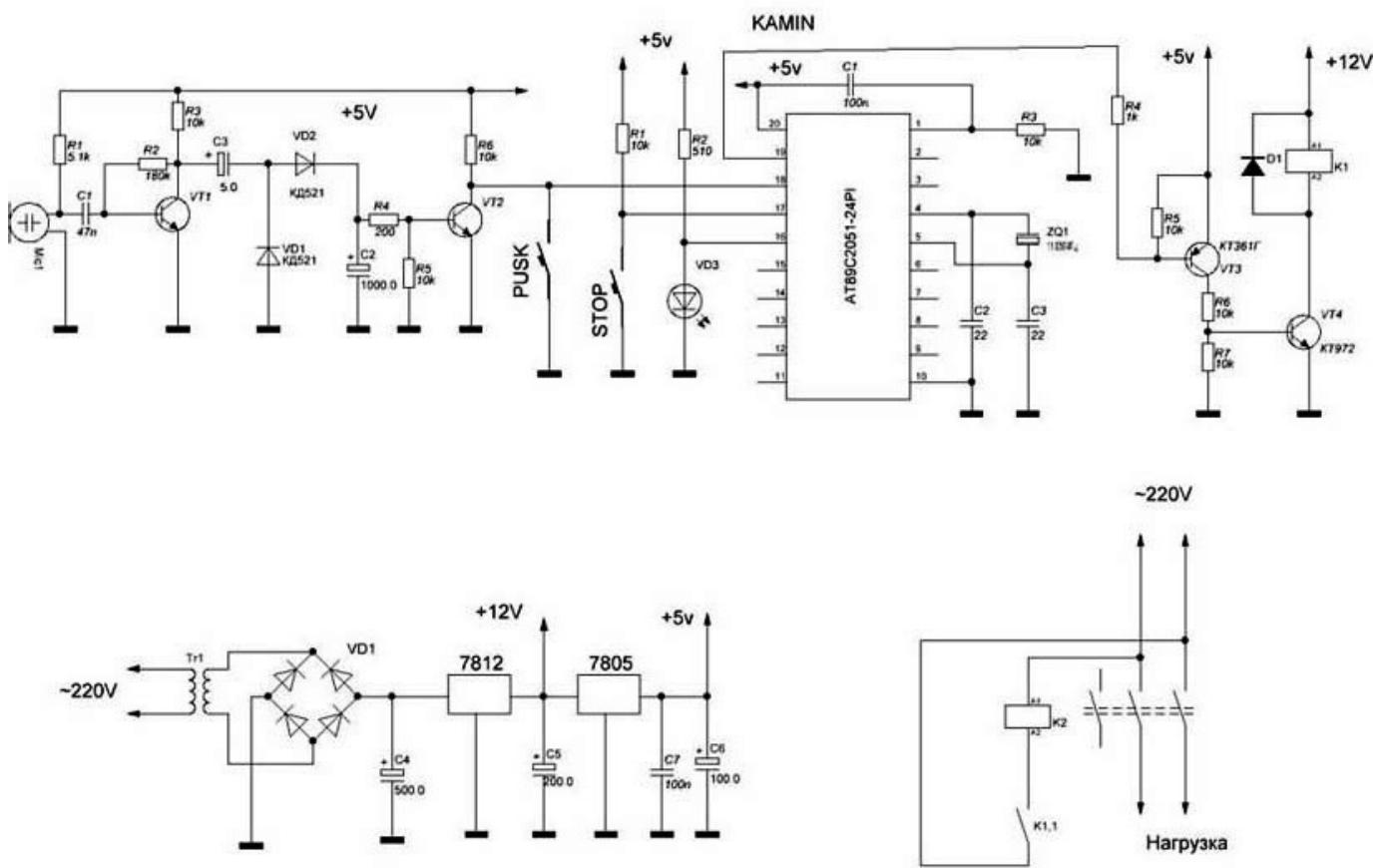


Рисунок 8. Схема блока управления камином

На схеме можно рассмотреть несколько узлов. На транзисторах  $VT_1$ ,  $VT_2$  выполнен входной усилитель - формирователь сигнала  $PUSK$ .

AT89C2051-24PI – управляющий микроконтроллер, выходной узел выполнен на транзисторах  $VT_3$ ,  $VT_4$  и реле  $K_1$ .

Силовая часть выполнена на магнитном пускателе, обозначенном на схеме K2. Также показан блок питания выполненный на интегральных стабилизаторах 78XX. Далее будет несколько подробней описана работа составных частей.

### **Формирователь входного сигнала**

На транзисторах VT1, VT2 (соответственно КТ3102Б и С9014) выполнен формирователь сигнала включения камина. Транзистор VT1 является предварительным усилителем сигнала микрофона.

С помощью резистора R2 на коллекторе транзистора VT1 устанавливается напряжение равное половине напряжения источника питания. Значение резистора R2 указано приблизительно, более точно подбирается при настройке в зависимости от коэффициента усиления транзистора VT1.

Усиленный транзистором VT1 звуковой сигнал с микрофона Mc1 через конденсатор C3 поступает на выпрямитель, выполненный на диодах VD1, VD2 по схеме с удвоением напряжения. Напряжение с этого выпрямителя заряжает конденсатор C3, которое (напряжение) открывает транзистор VT2.

Следует обратить внимание на емкость конденсатора C3. Она выбрана достаточно большой – 1000 мкФ, что позволило эффективно защититься от помех: транзистор VT2 открывается настолько медленно, что до появления на его выходе уровня логического нуля (именно этот уровень нужен для запуска контроллера) приходится ждать, пока из трубки телефона не раздастся 3 гудка вызова. Если ограничиться одним гудком, то транзистор открыться не успеет, а конденсатор C3 просто разрядится, подготовившись, тем самым, к следующему звонку.

Собственно вот это явилось ответом на вопрос, поверьте, такие доброжелатели нашлись в достаточном количестве: «А если, за воротами гаража бахахнут новогоднюю петарду?» Да, пожалуйста, хоть десять! Но, даже если и это не поможет, и камин все-таки включится, то через два часа произойдет автоматическое отключение, ведь микроконтроллер будет над этим делом думать не переставая. Как он об этом думает, будет рассказано несколько позднее. К тому же чувствительность микрофона с усилителем на одном транзисторе такова, что для срабатывания формирования приходится телефон прикладывать к микрофону почти вплотную.

Если набраться терпения, и дождаться четвертого гудка вызова (проверено при настройке), транзистор VT2 откроется и замкнет кнопку PUSK. В результате этого на выводе микроконтроллера 19, обозначенном на схеме как KAMIN, (такое же обозначение этого сигнала и в программе) появится сигнал низкого уровня,

который через транзисторы VT, VT4 включит реле K1. Реле K1 своим контактом K1.1 включит пускателя K2, а уже он включит нагрузку, в данном случае камин.

Схема включения реле K1 выполнена на транзисторах VT, VT4. Кто-то может спросить, почему на двух транзисторах, не достаточно ли было одного? Все дело в том, что на выводах микроконтроллера устанавливается высокий уровень (вспомните логические микросхемы!) на время импульса сброса, и остается таковым после его окончания. Поэтому необходимо включать исполнительное устройство, в нашем случае реле K1, низким логическим уровнем, который будет сформирован управляющей программой.

Мощность камина 2Квт, что предполагает ток через него около 10А, поэтому контактная группа пускателя должна быть выбрана с запасом, выдерживать ток, как минимум, в два раза больший.

В описываемом варианте удалось приобрести пускатели с контактами на 25А. На схеме просто не показан, но в реальной конструкции установлен автоматический выключатель на 16А. Мало ли, какие могут случиться казусы.

Если снова посмотреть на схему, то можно увидеть, что управление нагрузкой возможно просто кнопками. Также на схеме можно видеть светодиод. Когда камин находится во включенном состоянии, то светодиод подмигивает каждую секунду, наподобие часов.

Когда камин выключится, светодиод светится непрерывно, говоря о том, что питание на устройство подано. Таким образом, светодиод выполняет сразу две функции: индикации питания или индикации нормальной работы. Если он погашен, то что-то неисправно.

## **Блок питания**

Выполнен на интегральных стабилизаторах 7812 и 7805, на напряжения 12 и 5В соответственно, и никаких особенностей не имеет. Трансформатор применен от блока питания кассетного магнитофона, хотя может быть и менее мощным.

## **Что такое микроконтроллер**

Тут может возникнуть вопрос: микропроцессор и микроконтроллер это просто разное название одного и того же устройства, или это все-таки разные вещи?

Микропроцессор это центральное устройство любой ЭВМ, выполненный по интегральной технологии. Само название говорит о том, что именно в нем

происходят вычислительные процессы. Чтобы из него получилась ЭВМ, пусть даже не очень современная и мощная (вспомните любительские конструкции Радио-86 или Синклер), его надо дополнить внешними устройствами. В первую очередь это оперативная память и порты ввода вывода информации.

Микроконтроллер имеет внутри себя процессор, оперативную память, память программ, а кроме этого целый набор периферийных устройств, которые превращают процессор в полнофункциональную ЭВМ. По старой терминологии советских времен подобные устройства назывались Однокристальными Микро ЭВМ. Но советская вычислительная техника, как известно, зашла в тупик, а вместе с ней и ОМЭВМ.

Зарубежная же вычислительная техника на месте не стояла, поэтому ОМЭВМ стали называться контроллерами (от англ. Control – управлять, управление). И в самом деле, контроллеры оказались весьма пригодны для управления различной техникой, даже не очень сложной. После такого небольшого отступления вернемся к нашему устройству управления камином в гараже.

Почему был выбран именно этот микроконтроллер? Потому, что с другими я просто не работал, а этот уже был под рукой, к нему уже есть среда программирования, и программатор, есть несколько готовых программ, переделав которые, можно добиться требуемого результата.

Микроконтроллер AT89C2051-24PI относится к славному семейству i8051, разработанному фирмой INTEL в 1980 году. Конструкция оказалась настолько удачной, что до сих пор на базе ядра i8051 разными компаниями выпускается свыше 200 модификаций контроллеров с различными техническими параметрами. Самое примечательное в этом семействе, что система команд, созданная в 1980 году так до сих пор не претерпела ни одного дополнения или изменения, чего нельзя сказать о других семействах, хотя и появлялись новые модели.

Некоторая практика работы с этими МК показывает, что они никогда не зависают, работают очень стабильно, даже круглосуточно. Поэтому и был выбран AT89C2051-24PI. Последняя буква в этом обозначении говорит о том, что это микросхема индустриального исполнения, с рабочим диапазоном температур - 40 С ... + 85 С, что вполне подходит для задуманного применения.

Микросхема выполнена в корпусе DIP, о чем говорит буква Р, а число 24 указывает максимальную частоту кварцевого резонатора. Вообще – то эта микросхема, как и все остальные в семействе i8051 является статической: ее рабочая частота начинается от нуля Гц. Просто вместо кварца можно поставить

кнопку, и нажимать ее рукой. Хоть и очень медленно, но верно, контроллер будет выполнять записанную в памяти программу.

## **Алгоритм управления камином**

Как уже упоминалось, алгоритм управления задуман простейший: одной кнопкой включили, - другой выключили. Если выключить забыли, то выключится само через заданное время.

Казалось бы все просто, можно применить обычное реле времени, да только где его такое взять? Ответ достаточно простой: такое реле времени без всяких трудностей можно сделать на микроконтроллере, написав соответствующую программу.

## **Коротко о микроконтроллерах**

В короткой статье много не расскажешь, поэтому упомним вскользь лишь о тех узлах контроллера, которые будут использоваться в данном проекте. Почему, и в каком проекте?

Часто программное обеспечение для микроконтроллеров состоит из нескольких файлов – исходников, которые объединяются в одно целое с помощью проекта, поэтому вся программа проектом и называется. Это очень удобно, какие-то файлы можно добавлять, заменять другими или просто удалять. Возможно даже такое сочетание, - основная программа написана на языке высокого уровня СИ, а отдельные фрагменты, в виде вставок, написаны на Ассемблере.

Наш проект состоит из двух файлов: собственно программы main.c и заголовочного файла typedef.h. Оба файла очень маленькие, можно было все написать в одном, но поскольку они писались не заново, а просто методом урезания того, что уже было, то вполне возможен и такой вариант.

Более подробно о том, что такое микроконтроллеры и как они используются, читайте здесь - <http://electrik.info/main/automation/549-chto-takoe-mikrokontrollyer-naznachenie-ustroystvo-princip-raboty-soft.html>

## **Какие устройства микроконтроллера используются в этом проекте**

Классический контроллер семейства 8051 имеет 4 двунаправленных восьмиразрядных порта ввода – вывода. Каждый порт может использоваться как параллельный 8-ми разрядный, а также каждый разряд может использоваться отдельно, сам по себе. Такая возможность позволяет подключать отдельные

кнопки, светодиодные индикаторы, управлять работой реле. Ниже будет приведен текст исходной программы, и на его основе показано, как это делается.

## Порты ввода - вывода

Четыре 8-ми разрядных порта содержат контроллеры, выпускающиеся в 40 выводном корпусе, например, AT89S51. Но столько выводов требуется не всегда, поэтому выпускаются контроллеры в 20-ти выводном корпусе. Архитектура их та же как у «сороканожек», только во внешний мир выведены всего два порта P1 и P3. На рисунках 9 и 10 показана цоколевка AT89S51 и AT89C2051-24PI соответственно.

## Pin Configurations

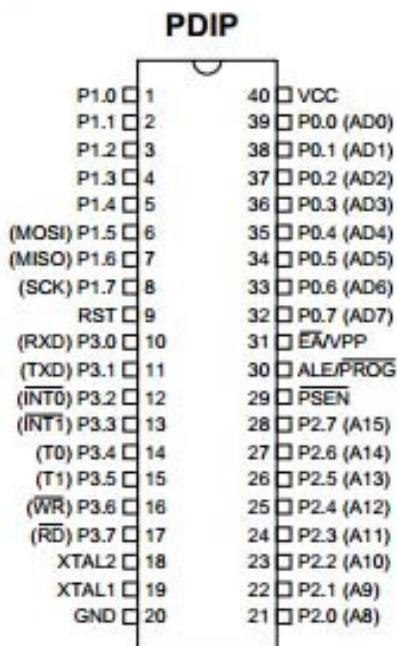


Рисунок 9.

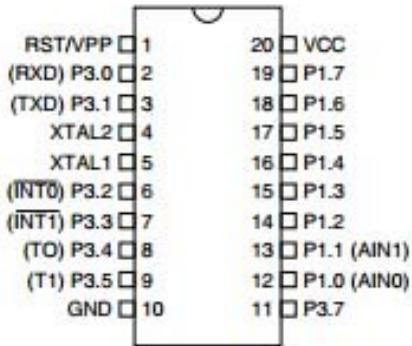


Рисунок 10.

В нашем проекте используются всего 4 вывода. Это разряды 4, 5, 6 и 7 порта P1. 4-й разряд используется для управления светодиодом индикации, 5 и 6-й кнопки управления, 7-й разряд – выход на исполнительное устройство.

Все это можно посмотреть на принципиальной схеме, показанной на рисунке 8. Таким образом, для работы схемы понадобилось всего 4 вывода микросхемы. Выводы питания, кварцевого резонатора, и RESET здесь не учитываются.

Как уже было сказано выше, получилось просто реле времени, управляемое двумя кнопками, к одной из которых «присоединен» с помощью микрофона сотовый телефон. Точнее не присоединен, а просто лежит рядом. Поэтому для отсчета интервала времени используется один из таймеров – счетчиков контроллера.

### Как работает таймер – счетчик

В режиме счетчика можно подсчитывать внешние импульсы. Этот режим нас не интересует. Режимы работы таймеров – счетчиков, как и всех остальных устройств контроллера, выбирается программно с помощью регистра специальных функций (SFR). В данном случае используется таймер - счетчик 0 (T/C0), настроенный в режим с автоперезагрузкой.

Таймер работает следующим образом. Предположим, что изначально он был обнулен. После того, как программным способом будет запущен счет импульсов, вырабатываемых специальной схемой, содержимое счетчика будет увеличиваться на единицу с приходом каждого импульса, пока не будет достигнута максимальная емкость счетчика, - во всех разрядах будут единицы. Максимальная емкость счетчика 0xFFFF или в десятичном виде 65535, поскольку таймер содержит два байта.

## Так что же считает таймер?

Тактовая частота контроллера стабилизирована кристалловым резонатором. На выполнение одной команды требуется 12 тактов задающего генератора, поэтому частота выполнения команд в 12 раз ниже частоты кристалла. Это называется машинным циклом.

При частоте кристалла 12,00МГц частота выполнения команд составляет ровно 1 МГц (период 1 микросекунда), что соответствует выполнению 1 миллиона операций в секунду. Это для тех машинных команд, которые выполняются за один машинный цикл. Некоторые команды выполняются за два машинных цикла. Вот как раз эти машинные циклы с периодом 1 микросекунда и считает таймер. По сути, таймер просто считает количество выполненных машинных команд.

С приходом последнего импульса счетчик достигает состояния 0xFFFF, а следующий счетный импульс переводит его в состояние 0000. Этот переход называется переполнением счетчика. Он, переход, устанавливает флаг прерываний по таймеру (TF0), и если прерывания разрешены, то возникает прерывание по таймеру.

При такой работе счетчика прерывание будет возникать каждые 65 535 микросекунд, или 65,535 миллисекунд, что согласитесь очень медленно. Если при каждом проходе таймер перезагружать величиной (65535 – 250) , то переполнение счетчика будет возникать каждые 250 микросекунд, вызывая тем самым подпрограмму обслуживания прерывания по таймеру 0. (На самом деле это немного не так. Таймер работает в режиме с автоперезагрузкой. Значение перезагрузки #define PERIOD (-230) хранится в старшем байте таймера и перезагружается оттуда в младший байт).

Для подсчета количества прерываний в программе служит переменная `unsigned char intcycle;`. Она увеличивается каждые 250 микросекунд. Складываясь, эти `intcycle` в количестве 4-х дают одну миллисекунду.

Любое прерывание приостанавливает выполнение основной программы и запускает подпрограмму обслуживания прерывания, в данном случае просто подсчет времени. Время учитывается с помощью структуры `time vremya;`. По окончании подпрограммы обслуживания продолжается выполнение основной программы с того места, где она была прервана.

Структура - это одна из конструкций языка СИ, не обязательно время. Это может быть, например, товар на складе. Тогда вместо миллисекунд, секунд, минут

и часов в структуре могут быть совсем другие переменные: наименование товара, артикул, цвет, размер и т.п.

Даже структур time может быть несколько: time текущее время, time время игры, time будильник, time время до конца рабочего дня. Естественно все эти имена должны вписываться в текст программы на латинице.

В нашем случае это просто отвлеченная структура time vremya. Кнопку нажали, - время пошло. Другую, - остановилось и сбросилось. Дотикало до определенного предела, время остановилось, нагрузка выключилась. Хотя именно такая структура, если дополнить схему цифровыми индикаторами, позволяет показывать время. Для этого надо дополнить программу соответствующими функциями.

## Кстати, о функциях

Любая программа на языке СИ начинается с функции под названием main, это имя функции. Все, что находится между фигурными скобками называется телом функции.

```
void main (void) {  
init();  
t0setup();  
while(1) {  
if (!PUSK) kamin_ON();  
if (!STOP) init();  
if (vremya.hour >= TIME_OFF) init();  
} // end of while (1)  
} // end of main
```

В начале функции main однократно вызываются функции init (); t0setup (); Они осуществляют сброс структуры времени и выключение камина, запрещают прерывания, сразу все EA = 0;, а также настройку таймера 0. while (1) {...тело цикла...} – бесконечный цикл. Все, что находится между фигурными скобками повторяется, пока не выключено питание.

Внутри этого цикла опрос кнопок, и проверка времени выключения. (Вот, собственно, и вся основная программа.) Это делается следующим образом. if (!PUSK) kamin\_ON(); если (условие выполняется) включить камина;. В данном случае условием является (!PUSK) - нажатие кнопки PUSK, о чем говорит восклицательный знак, означающий в языке СИ инверсию.

Если условие истинно (выражение в скобках равно нулю), то включить камин. В противном случае перейти к выполнению следующей команды. Для выключения камина кнопкой или по времени специальная функция не писалась. Достаточно просто вызвать функцию `init ()`.

Вот тут наглядный пример, какие возможности и преимущества дают функции: проще несколько раз вызвать функцию, написав в тексте программы `init ()`, чем переписывать в нескольких местах текст функции полностью.

### **Комментарии и их роль в исходном тексте программы**

Во всех учебниках по программированию говорится о том, что не надо пренебрегать написанием комментариев, просто не лениться. Дело в том, что при написании программы все кажется понятным и простым, даже если в программе всего несколько строк. Уже через две недели, месяц, все забывается напрочь, приходится разбираться, какие же мысли Вас посещали в тот момент? Иногда бывает проще написать программу заново.

В языке СИ имеется два вида комментариев: `/*ЭТО КОММЕНТАРИЙ НА НЕСКОЛЬКО СТРОК*/` позволяет разместить целый блок текста\*

```
//а это комментарий всего для одной строки  
//очень удобен при отладке программ.
```

В первой строке файла – исходника неплохо указать имя файла, например `//main.c`, `//typedef.h`, `//decoder.c`. Я специально не стал убирать «философские» размышления в самом начале файла `//main.c` о том, как начать написание этого проекта.

Вот, пожалуй, и все объяснения по поводу проекта, в следующей части статьи приводится исходный текст программы.

### **Исходный текст программы**

О нем было уже рассказано выше, текст содержит подробные комментарии, что поможет понять о чем собственно идет речь.

### **Программа на языке СИ**

```
//main.c
```

/\* Цель проекта: управление камином в гараже по сотовому телефону.

Алгоритм управления простейший: получив вызов по сотовому телефону, устройство просто включает камин, а по прошествии двух или трех часов просто его выключает. При этом возможно повторное включение звонком с телефона, когда все уже включено.

Даже в случае ложного включения все равно камин отключится через указанное в программе время.

В качестве датчика звонка будет использоваться просто микрофон, желательно угольный, поскольку он не реагирует на электромагнитные помехи. угольный теперь найти трудно, поэтому конденсаторный.

Желательно также предусмотреть включение - выключение просто кнопками. За основу придется брать те же гоблинские часы или биллиардный таймер. Оттуда же возьмем и заголовочный файл.

Вот, пока писал эти строки, кажется, возникли такие идеи.

1. При включении питания вызывать функцию ИНИТ. В ней сбросить структуру подсчета времени, на всякий случай, выключить камин. При этом не разрешать прерывания, чтобы не начался отсчет времени. Вполне достаточно одной структуры, просто она будет останавливаться и запускаться кнопками по мере надобности.

2. При получении импульса запуска от телефона, включить камин, и разрешить прерывания. Для ручного включения предусмотреть к этому входу просто параллельную кнопку.

3. По истечении заданного времени просто вызвать функцию ИНИТ, тем самым выключив камин и подготовив его к следующему включению.

4. Кнопка СТОП также просто вызывает функцию ИНИТ. Произойдет отключение. Если будет холодно, то можно просто нажать кнопку ПУСК или позвонить по телефону!!!

5. Светодиод W\_Led при включенном камине мигает каждую секунду. При отключенном камине светодиод светит постоянно, что говорит о наличии питания устройства в целом.

\*/

```

//Подключение заголовочных файлов
#include <reg52.h> // набор инструкций mk52
#include "typedef.h" // Мои определения структур, типов unsigned char,
unsigned int и порты

//Назначение уровней сигналов управления
#define ON 0 //Включение реле низким уровнем сигнала
#define OFF 1 //а выключение высоким.

//Таймер 0 в режиме работы с автоперезагрузкой
#define PERIOD (-230) // Задаёт период обработки прерывания по таймеру 0
(250мкс на 11.0592МГц)
//struct time vremya; // текущее время
time vremya; //МОЖНО ОБЪЯВИТЬ ЭТИ СТРУКТУРЫ И ТАК
//сама структура находится в заголовочном файле typdef.h

//Объявление переменных
unsigned char intcycle; // обновляется каждые 250мкс от 0 до 3, т.е. цикл=1мс
// и используется в таймере реального времени.
/*
//Кнопки управления пока для платы гоблинских часов, ибо там уже есть кнопки
sbit PUSK = P1^2; //Кнопка ПУСК, она же включает и от телефона
sbit STOP = P1^3; //Кнопка СТОП
sbit KAMIN = P1^0; //Сигнал включения камина.
sbit W_Led = P1^6; //Белый светодиод, мигает каждую секунду, пока камин
включен
*/
/*
//Кнопки управления для платы Устройства на базе AT89C2051
sbit PUSK = P1^6; //Кнопка ПУСК, она же включает и от телефона
sbit STOP = P1^5; //Кнопка СТОП
sbit KAMIN = P1^7; //Сигнал включения камина.
sbit W_Led = P1^4; //Светодиод, мигает каждую секунду, пока включен камин.
*/
//-----
//// Выполнение начальных условий /////
void init (void) { // инициализация всего
vremya.hour = 00; // инициализация часов 00:00:00
vremya.min = 00;

```

```

vremya.sec = 00;
vremya.ms = 000;
KAMIN = OFF; //камин выключить
W_Led = OFF; // погасить светодиод, точнее зажечь, т. к. инверсное включение
EA = 0; //пока все прерывания запретить
P1 - 0xFF; //Настроить порт P1 на ввод.
intcycle=0;
}

//включение камина
void kamin_ON (void) {
vremya.hour = 00; // инициализация часов 00:00:00
vremya.min = 00;
vremya.sec = 00;
vremya.ms = 000;
EA = 1; //разрешить все прерывания
intcycle=0;
KAMIN = ON; //камин включить
}

/*
* The following function is called from the interrupt service routine. This
* means that the code must be generated in the same Register Bank as the
* interrupt service function.
*/
*****  

*****  

* ПОДПРОГРАММА ОТРАБОТКИ ПРЕРЫВАНИЯ 0 *
* Отрабатывается каждые 250 мкс *
*****  

*****/  

void timer0 (void) interrupt 1 { /* Int Vector at 000BH, Reg Bank 1 */  

//Мигание светодиода каждую секунду
if (vremya.ms < SEK/2)
W_Led = ON;  

if (vremya.ms >= SEK/2)
W_Led = OFF;  

//А это, собственно, подсчет времени, просто часы от 00 до 24 часов

```

```

//если их показания вывести на цифровой индикатор, то получатся просто часы

if (++intcycle==4) { // 1 мс = 4 * 250 мкс
intcycle=0;
if (++vremya.ms == SEK) { // Обновление счётчика миллисекунд
vremya.ms=0;
if (++vremya.sec == 60) { // Обновление счётчика секунд
vremya.sec=0;
if (++vremya.min == 60) { // Обновление счётчика минут
vremya.min=0;
if (++vremya.hour==24) { // Обновление счётчика часов
vremya.hour=0;
} // hour
} // min
} // sec
} // ms

} // end of ++intcycle

} // end of timer0
//-----
void t0setup (void) {
// setup the timer 0 interrupt
TH0 = PERIOD; // set timer period
TL0 = PERIOD; //ПО МОЕМУ ЭТУ ВОТ СТРОКУ МОЖНО И НЕ ДЕЛАТЬ. БЕЗ
HEE РАБОТАЕТ ТАКЖЕ КАК И С НЕЙ
TMOD = TMOD | 0x02; // select mode 2 (режим с автоперезагрузкой в данном случае
PERIOD)
TR0 = 1; // start timer 0
ET0 = 1; // enable timer 0 interrupt.
}

//-----
void main (void) {
init (); // п/п инициализации
t0setup (); // п/п инициализации т/c0

while (1) { //Бесконечный цикл, чтобы программа не остановилась выполнив один
проход

if (!PUSK)
kamin_ON(); //если нажата Кнопка ПУСК ,она же включает и от телефона

```

```

if (!STOP)
init () //если нажата Кнопка СТОП

#ifndef DEBUG
if (vremya.hour >= TIME_OFF) //Если истекло заданное время, выключить через
два часа
init () //Это рабочий режим
#endif

#endif DEBUG
if (vremya.sec >= TIME_OFF) //в режиме отладки выключить через две секунды
init () //указывается в заголовочном файле typedef.h
#endif
} // end of while (1)
} // end of main

```

Заголовочный файл

```

//typedef .h

//Первоначальный вариант этого файла смотри в проекте Гоблинские часы
typedef struct { // объявление структуры time
unsigned char hour; //НО МОЖНО ОБЪЯВИТЬ И ВОТ ТАК.
unsigned char min;
unsigned char sec;
unsigned int ms;
} time;
//Чтобы войти в режим отладки следующую строку раскомментировать
//#define DEBUG

```

```

//В режиме отладки
#define DEBUG
#define SEK 1000 //длительность секунды в миллисекундах
#define TIME_OFF 10 //Время включения камина вышло
#endif

```

```

//Режим работы
#ifndef DEBUG
#define SEK 1000
#define TIME_OFF 2 //Время включения камина вышло
#endif

```

На этом исходный текст программы заканчивается. Дополнительный комментарий по поводу условия if (++intcycle==4). В скобках выполняется несколько действий: сначала при каждом вхождении в подпрограмму обслуживания прерывания от таймера увеличивается на единицу ++intcycle (двойной плюсик перед именем переменной intcycle).

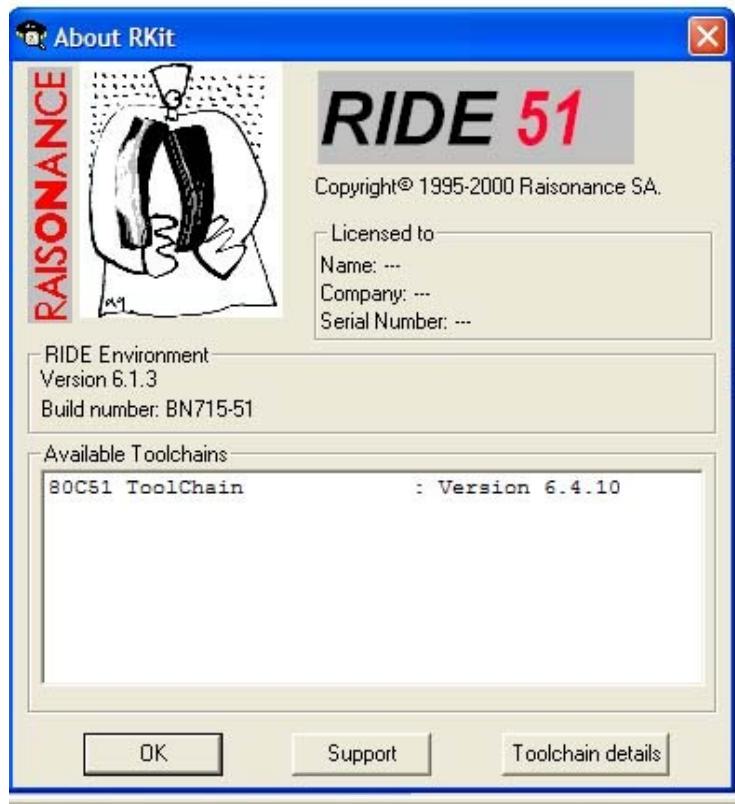
Двойной знак равенства говорит о том, что полученное значение intcycle сравнивается с константой 4. По достижении этой величины intcycle = 0; (в данном случае знак = есть операция присваивания, читается как: «Переменной intcycle присвоить числовое значение 0». Вместо константы может быть просто переменная или даже целое выражение, например intcycle = a+b;. Сначала будет вычислено значение выражения, а результат присвоен переменной intcycle).

Таким же способом подсчитываются миллисекунды, секунды, минуты и часы.

## **Среда программирования RIDE 51**

Этот рассказ был бы неполным, если хоть чуть не рассказать о среде программирования, в которой этот проект был создан. Книжка о работе со средой Getting Started with the Raisonance Development Kits содержит 128 страниц. Это среда RIDE 51 американской фирмы Raisonance. Использование ее в оценочном режиме позволяет создавать проекты с объемом машинного кода не более 4КБ, чего более чем достаточно для любительских целей и не только.

Информация о среде RIDE 51 показана на рисунке 11.



## Рисунок 11.

Среда имеет свой встроенный текстовый редактор, подсвечивающий текст программы разными цветами. Исходный текст, созданный в этом редакторе показан на рисунке 12.

```
/*-----*
*          ПОДПРОГРАММА ОТРАБОТКИ ПРЕРЫВАНИЯ 0
*          Отрабатывается каждые 250 мкс
*-----*/
void timer0 (void) interrupt 1 { /* Int Vector at 000BH, Reg Bank 1 */

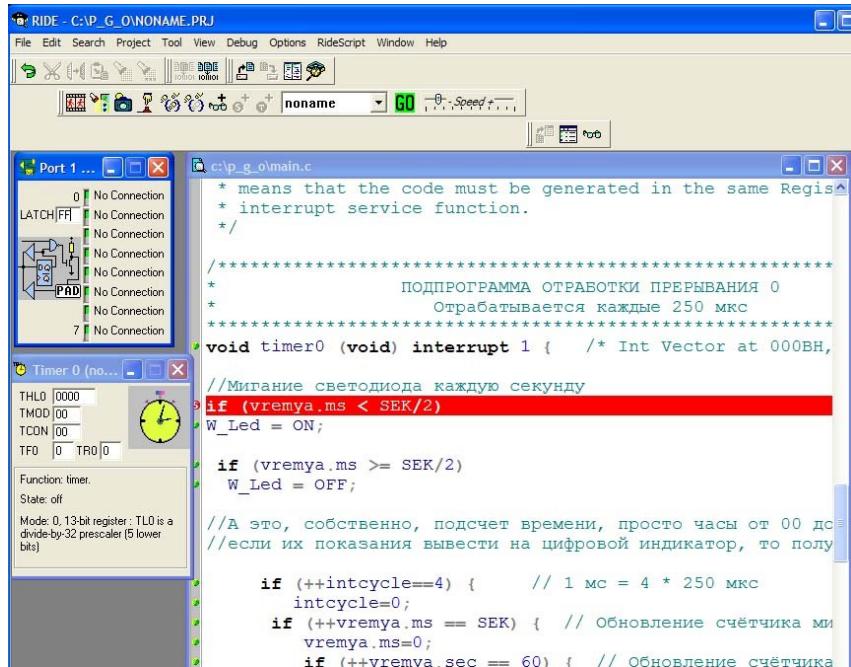
// Мигание светодиода каждую секунду
if (vremya.ms < SEK/2)
W_Led = ON;

if (vremya.ms >= SEK/2)
W_Led = OFF;

// А это, собственно, подсчет времени, просто часы от 00 до 24 часов
// если их показания вывести на цифровой индикатор, то получатся просто часы

if (++intcycle==4) { // 1 мс = 4 * 250 мкс
intcycle=0;
if (++vremya.ms == SEK) { // Обновление счётчика миллисекунд
vremya.ms=0;
if (++vremya.sec == 60) { // Обновление счётчика секунд
vremya.sec=0;
if (++vremya.min == 60) { // Обновление счётчика минут
vremya.min=0;
if (++vremya.hour==24) { // Обновление счётчика часов
vremya.hour=0;
} // hour
} // min
} // sec
}
```

RIDE 51 в своем составе имеет также превосходный симулятор – отладчик, позволяющий отлаживать написанные программы. Его скриншот показан на рисунке.



## Полезные ссылки:

Статьи про домашнюю автоматизацию на сайте Электрик Инфо –  
<http://electrik.info/main/automation/>

Булева алгебра –  
<http://electrik.info/main/fakty/229-buleva-algebra-chast-1-nemnogo-istorii.html>

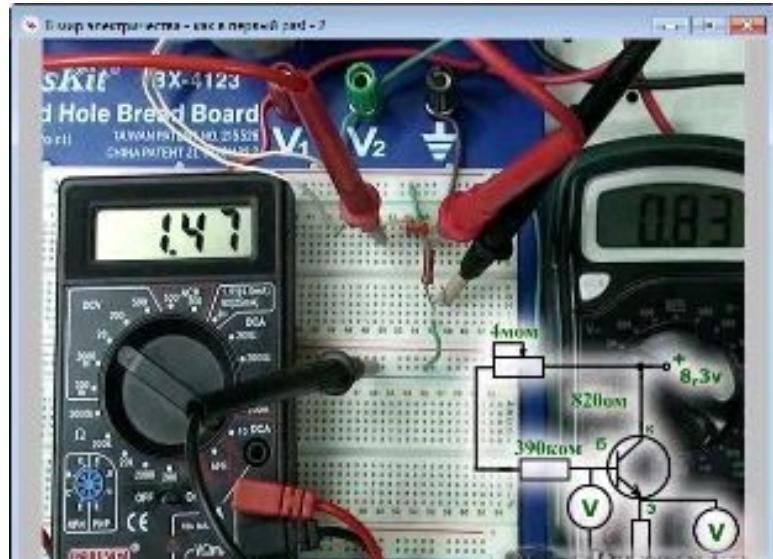
Логические микросхемы –  
<http://electrik.info/main/praktika/251-logicheskie-mikrosxemy-chast-1.html>

Как подключить нагрузку к блоку управления на микросхемах –  
<http://electrik.info/main/praktika/226-kak-podklyuchit-nagruzku-k-bloku-upravleniya-na.html>

Как можно легко управлять мощной нагрузкой переменного тока –  
<http://electrik.info/main/master/77-kak-mozhno-legko-upravlyat-moshchnoj-nagruzkoj.html>

Автоматизация освещения с использованием технологии X10 –  
<http://electrik.info/main/master/294-avtomatizaciya-osveshheniya-s-ispolzovaniem.html>

Видео уроки: <http://electrik.info/electronika.php>



В мир электричества - как в первый раз!-2. А что внутри диска?  
<http://electrik.info/main/school/607-v-mir-elektrichestva-kak-v-pervyy-raz-2.html>