Не ошибается лишь тот, кто ничего не делает.

PROTEUS VSM

Система виртуального моделирования.

Наверняка многие начинающие радиолюбители сталкивались с ситуацией когда, решив собрать понравившееся и, несомненно, нужное устройства, по неопытности а может из за ошибок в схеме или же по другим «форс-мажорным» обстоятельствам ,просто-напросто сжигали с трудом приобретенные дорогие радиодетали. И скорей всего большинство, обжегшись на первых неудачах, забрасывали занятие радиоэлектроникой навсегда.

В наш век повальной компьютеризации, нашелся выход из этого тупика. Появилось огромное количество программ симуляторов, заменяющих реальные радиодетали и приборы, виртуальными моделями. Симуляторы позволяют, без сборки реального устройства, отладить работу схемы, найти ошибки, полученные на стадии проектирования , снять необходимые характеристики и многое другое.

Одна из таких программ PROTEUS VSM . Но симуляция радиоэлементов это не единственная способность программы . Proteus VSM , созданная фирмой Labcenter Electronics на основе ядра SPICE3F5 университета Berkeley, является так называемой средой сквозного проектирования .Это означает создание устройства, начиная с его графического изображения (принципиальной схемы) и заканчивая изготовления печатной платы устройства , с возможностью контроля на каждом этапе производства.

Но, не смотря на кажущуюся сложность программы, пользоваться ей смогут не только профессионалы в мире радиоэлектроники, но и новички, научившиеся, а может пока еще и нет, отличать резистор от транзистора.

В «сферу-влияний» PROTEUS VSM входят как простейшие аналоговые устройства так и сложные системы созданные на популярных ныне микроконтроллерах. Доступна огромная библиотека моделей элементов, пополнять которую может сам пользователь, естественно для этого нужно досконально знать работу элемента и уметь программировать. Возможность анимации схем позволяет программе стать прекрасным учебным пособием на уроках в школе и ВУЗ. Достаточный набор инструментов и функций, среди которых вольтметр, амперметр, осциллограф, всевозможные генераторы, способность отлаживать программное обеспечение микроконтроллеров, делают PROTEUS VSM хорошим помощником разработчику электронных устройств.

Сайт разработчиков программы http://www.labcenter.co.uk .

Программа не прихотлива к системным требованиям. Как заявлено создателями работает в Windows 98/Me/2k/XP и выше. Запускается даже на Pentium I 150 МГц. Но все-таки для комфортной работы и более полного использования возможностей программы

рекомендуется процессор не ниже 500 МГц, объем оперативной памяти 64 МБ, звуковая карта совместимая с DirectX и разрешение монитора не ниже 1024 x 768 точек.

Proteus VSM по умолчанию устанавливается в папку C:\Program Files\Labcenter Electronics\Proteus 6 Demonstration. Другие пути установки и варианты с системой стоящей на другом диске, например XP на диске D ,мной не проверялись.

Proteus VSM состоит из двух самостоятельных программ ISIS и ARES .ARES это трассировщик печатных плат с возможностью создания своих библиотек корпусов и здесь ее не будем.

Основной программой является ISIS, в ней предусмотрена горячая связь с ARES для передачи проекта для разводки платы.

При запуске программы появляется основное окно.

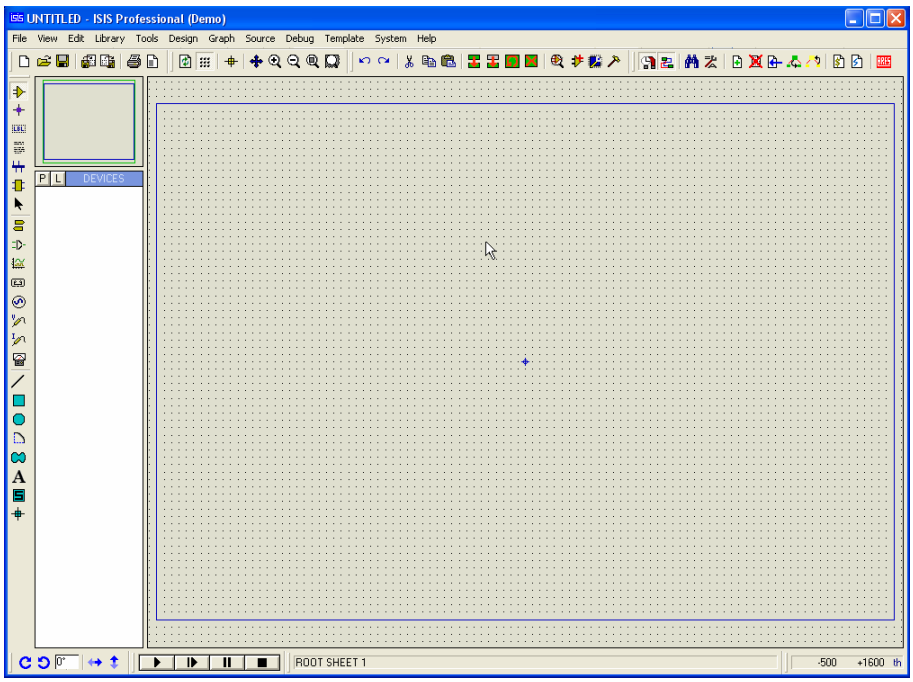
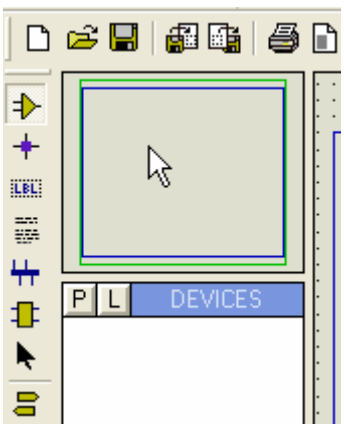


рис 01

Самое большое пространство отведено под окно редактирования EDIT WINDOW .
Именно в нем происходят все основные процессы создания, редактирования и отладки схемы устройства.

рис 02



Слева вверху маленькое окно предварительного просмотра Overview Window с его помощью можно перемещаться по окну редактирования (щелкая левой кнопкой мыши по окну предварительного просмотра, мы перемещаем окно редактирования по схеме, если конечно схема не вмещается в окно).

Перемещать окно редактирования по схеме можно еще так - удерживая нажатой кнопку SHIFT двигать курсор мыши, не нажимая ее кнопок, по окну редактирования.

Приближать и отдалять схему в окне можно соответственно кнопками F6 и F7 или же колесом мыши, F5 центрирует схему в окне а нажатие F8 подгоняет размер схемы под окно редактирование.

Под окном предварительного просмотра находится Object Selector список выбранных в данный момент компонентов, символов и других элементов. Выделенный в списке объект
отображается в окне предварительного просмотра.

Все возможные функции и инструменты Proteus VSM доступны через меню расположенное в самом верху основного окна программы, через пиктограммы находящиеся под меню и в левом углу основного окна и через горячие клавиши, которые могут переназначаться пользователем.

рис 03

В самом низу основного окна расположены: слева направо кнопки вращения и разворота объекта вокруг своей оси, панель управления интерактивной симуляцией (выглядит как магнитофонная и функции такие же: ПУСК –ПОШАГОВЫЙ РЕЖИМ – ПАУЗА-СТОП) ,

рис 03а

строка статуса (в ней отображаются :ошибки ,подсказки ,текущее состояние процесса симуляции и т.д.) и координаты курсора отображаемые в дюймах .

Для освоения основных функций программы нам понадобится «жертва».
Откроем один из уже имеющихся проектов. Выберем в меню FILE опцию LOAD DESIGN. Загрузим файл SAMPLE/ANIMATION CIRCUIT/AC01.DSN .

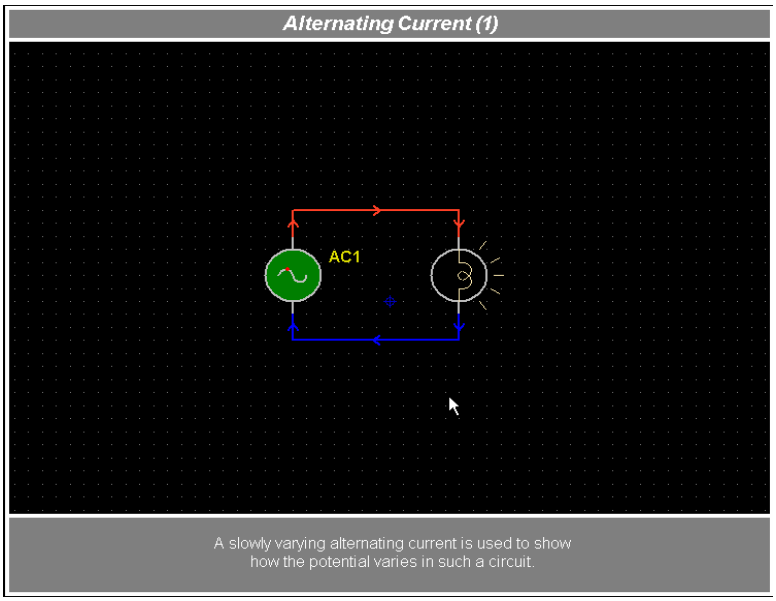


рис 04

Запустим проект, нажав на панели кнопку ПУСК.

рис 05

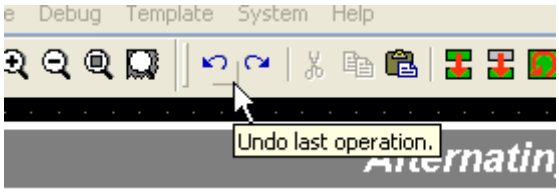
Эта схема демонстрирует действие переменного тока в цепи. Частота генератора занижена до 0.5 Гц для наглядности.

Цвет и яркость проводов определяют полярность и уровень напряжение, стрелки направление тока. Красная точка на изображение генератора показывает текущее «положение» синусоиды.

Для того чтобы манипулировать объектами их нужно сначала выделить, это можно сделать только на остановленном проекте. Для выделения одного объекта надо щелкнуть по нему правой кнопкой мыши. Для выделения группы можно либо удерживая CTRL последовательно щелкать правой кнопкой по всем объектам либо удерживая правую кнопку протащить область выделения по необходимым объектам. Выделять объекты надо осторожно, повторный щелчок правой кнопкой мыши по выделенному объекту удалит его, (удалить выделенные объекты можно еще, нажав кнопку DELETE) .

Но это не страшно отменить последние и все предыдущие действия по порядку можно с помощью кнопок отмены (UNDO, REDO).

рис 06



Кнопки отмены действуют как назад, по хронологии так и вперед.

Выделенные объекты можно перемещать по схеме, ухватив их левой кнопкой мыши

передвинув в нужное место отпустить кнопку. А с помощью этих кнопок производятся групповые операции с выделенными объектами. По порядку: копирование, перемещение, поворот и удаление.

рис 07

Потренируетесь на этом проекте. Не бойтесь «испортить» чего-либо.
Изменения не внесутся в проект пока вы, не сохраните его.

Откройте следующий проект Diode07.DSN находящийся в той же папке, предыдущий закроется, автоматически спросив вас «не желаете ли сохранить изменения». Ответьте, нет. Запустите проект.

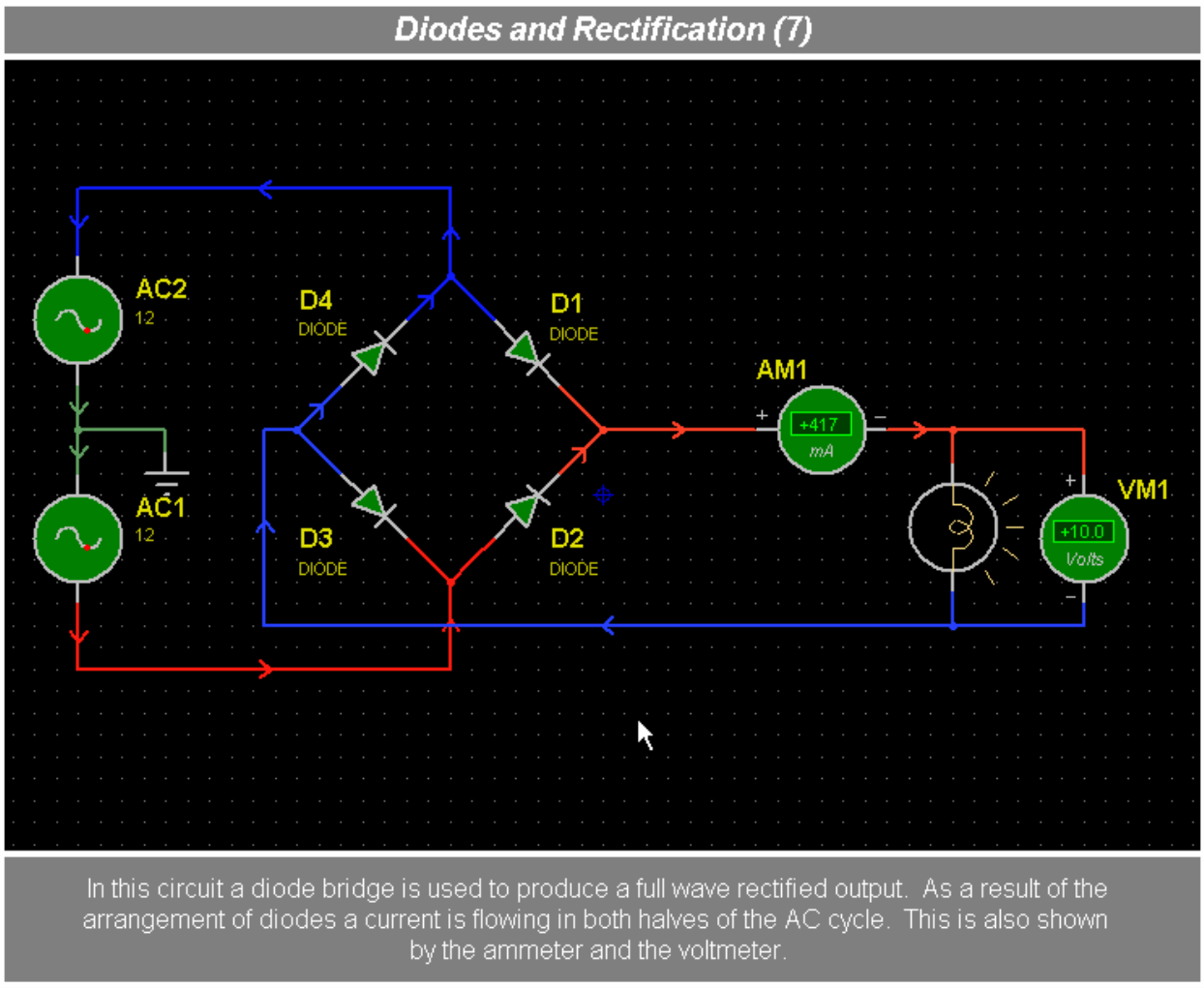


рис 08

Проект иллюстрирует работу двухполупериодного выпрямителя, проще сказать, диодного моста. Хорошо видны все процессы, проходящие в схеме. Так же как и в предыдущем проекте, частота генераторов снижена. Переделаем схему в реальную. Нам нужна частота 50 Гц. Для этого нам придется отредактировать свойства генераторов. Чтобы открыть окно редактирования компонента, нужно либо выделив компонент

щелкнуть по нему левой кнопкой мыши либо поместив на него курсор, не нажимая кнопок мыши, нажать CTRL + E . Откроем окно редактирования.

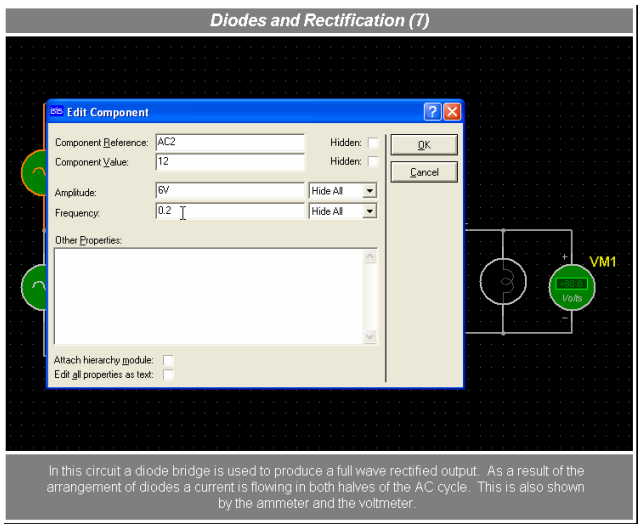


рис 09

Внесем в поле Frequency (Частота) 50. Закройте окно, нажав OK.Измените частоту и для второго генератора. Запустите проект .Да ,что то получилось не то .

Добавим в схему конденсатор.

Тот, что находится в списке ,CAPACITOR придется заменить. Дело в том, что проект существует с самой первой версии программы, модель разработчики изменили, а внести изменения в схему не потрудились. Такого рода «жучков» в старых проектах достаточно. Ну не чего страшного!

Все элементы находятся, как на складе, в библиотеке компонентов.

Чтобы попасть на этот «склад» перейдем в режим COMPONENT (компоненты), нажав соответствующую пиктограмму.

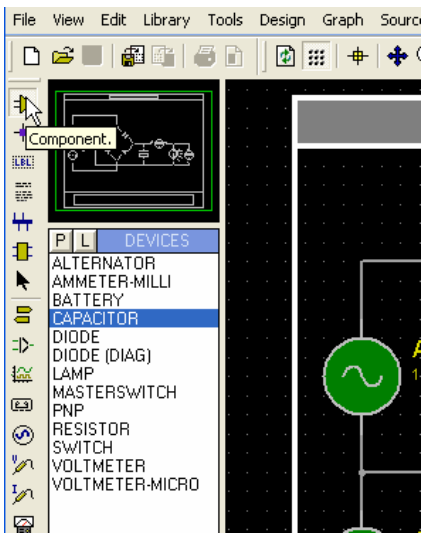


рис 10

Теперь либо щелкнув по пиктограмме P (Pick devices) либо дважды щелкнув левой кнопкой в поле выбора компонентов Object Selector , мы попадем на «склад».

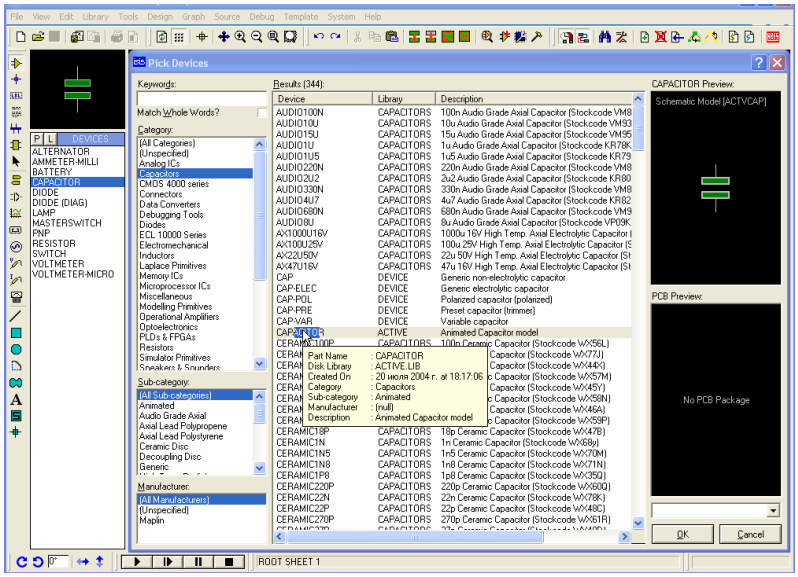


рис 11

Компоненты можно выбирать по категориям Category , подкатегориям-Sub category , по производителю Manufacturer или же искать по ключевым словам Keywords. Выберем CAPACITOR библиотеки ACTIV . Дважды щелкнем по строке с названием объекта, подтверждая выбор компонента.

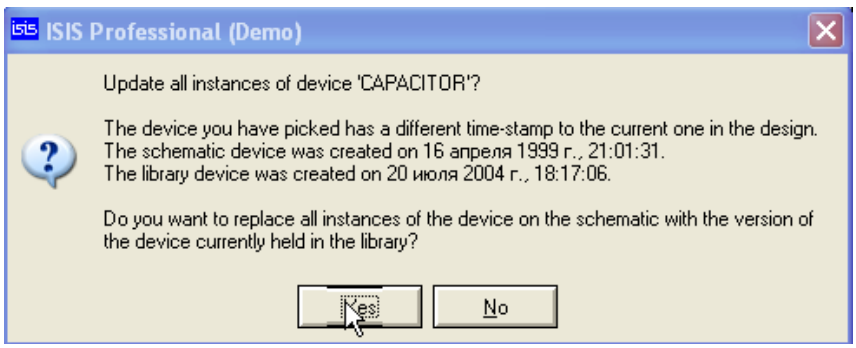
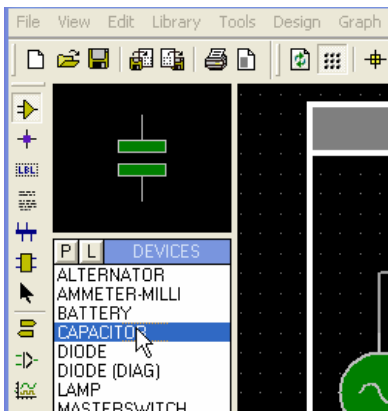


рис 12

Программа спросит, желаете ли вы заменить существующий в списке схемы компонент попутно сообщив, что компонент библиотеки и схемы имеют разное время создания, библиотечный свежее. Ответьте да. Закройте «склад» нажав OK или же закрыв окно. Выберете компонент, в списке щелкнув по нему левой кнопкой. Изображение конденсатора появиться в окне предпросмотра. Если необходимо разверните его как вам нужно.



Поместим конденсатор после диодного моста, просто щелкнув там левой кнопкой.

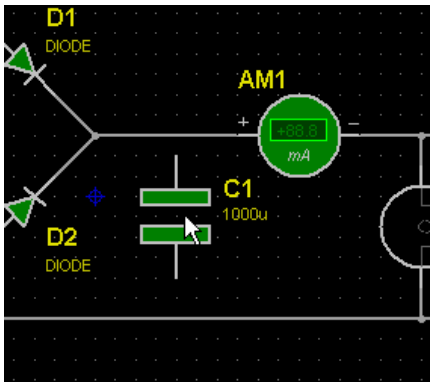
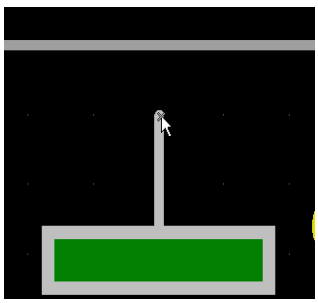
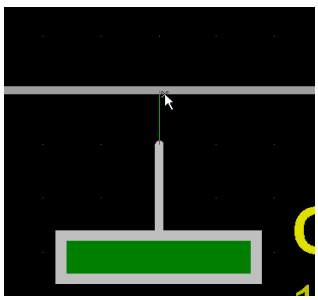
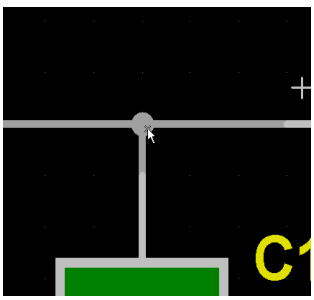


рис 14

Теперь нам надо подсоединить его к схеме. Поместите курсор на верхний вывод конденсатора, на конце курсора появится крестик, показывающий что соединение возможно. Щелкните левой кнопкой, передвиньте курсор на провод, выше конденсатора ,появится тонкая линия, показывающая возможные соединения. Когда курсор будет над проводом, вновь появится крестик. Щелкните левой кнопкой еще раз.







рисунки 15 ,16 ,17

Соедините нижний вывод сами. Измените емкость конденсатора на 500 мкФ. Запустите симуляцию. Ну, вот совсем другое дело. Количество плюсиков и минусов на обкладках конденсатора указывает на уровень заряда. Измените обратно частоту обоих генераторов на 0,2 Гц, в качестве разделителя в десятичной дроби Proteus VSM использует точку (английская раскладка клавиатуры).
Запустив, проект вы увидите процесс зарядки-разрядки конденсатора в динамике.

И так мы уже научились открывать проект, запускать его, перемещаться по схеме , манипулировать объектами, редактировать их свойства и добавлять элементы в схему. Продолжим. Научимся применять органы управления схемы, именуемыми в PROTEUS активаторами.

Откроем проект Basic07.DSN .

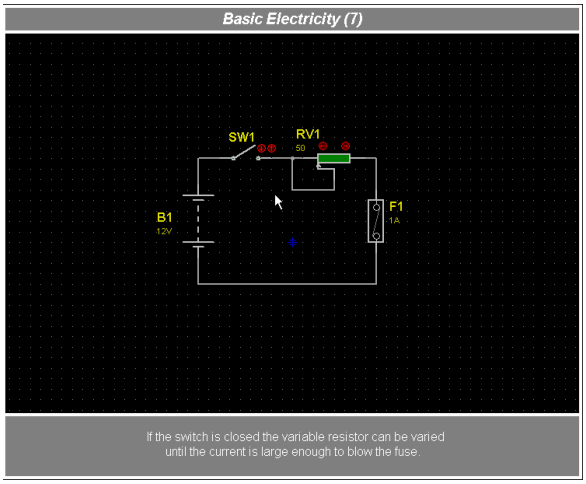


рис 18

Простейшая схема. Запустите проект. У тумблера и реостата имеются красные стрелки в кружках. Это и есть активаторы. Нажимая на них левой кнопкой можно переключать тумблер или же перемещать движок реостата, изменяя, таким образом, его сопротивление. Включите тумблер и переместите движок реостата в крайнее правое положение. Ну вот! Предохранитель сгорел. Ничего при рестарте проекта он будет целым.

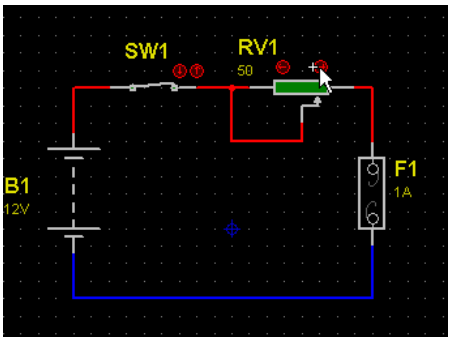


рис 19

Загрузите проект Comb01.DSN .

Это схема демонстрирует работу логического примитива И.

Запустим. Попробуем изменить логические уровни на входе, щелкая левой кнопкой по активаторам, подсоединенным к входам элемента. Ничего не получается. Симулятор сообщает Real time Simulation in progress Press ESC to Stop.

Здесь та же история что и с конденсатором. Модель LOGICSTATE изменена и в схеме не активна. Когда щелкаем по органу, симулятор, не воспринимая модель как активную, думает, что мы собираемся редактировать работающую схему и ругается. Ну что ж исправим это. Откроем библиотеку и найдем элемент LOGICSTATE (логический уровень) он находится в категории Debugging tools . Поместим его в наш список, дважды щелкнув по строке. Программа снова спросит нас о замене. Ответьте утвердительно. Закройте библиотеку, и запустите проект. Вот теперь все в порядке. Изучите работу элемента изменяя, логические уровни на его входах и сравнивая с таблицей истинности. Потренируйтесь с остальными проектами из серии Comb .

Мы овладели необходимым минимумом, настало время создания своих проектов. Создайте новый проект, используя меню FILE > NEW DESIGN. Это можно не делать, если вы только что открыли программу, так как при запуске PROTEUS автоматически

создаст новый проект с именем UNTITLED.DSN – безымянный.

Установим для удобства свои размеры листа схемы, Откроем меню SYSTEM > SET SHEET SIZE (Установить размеры листа). Выберем вариант USER – пользовательский, в окошках введем 6 in 4 in (высота и ширина в дюймах).
После этого нажмите F8 , чтобы подогнать размер листа схемы под окно редактирования.

Соберем схему согласно рисунку.

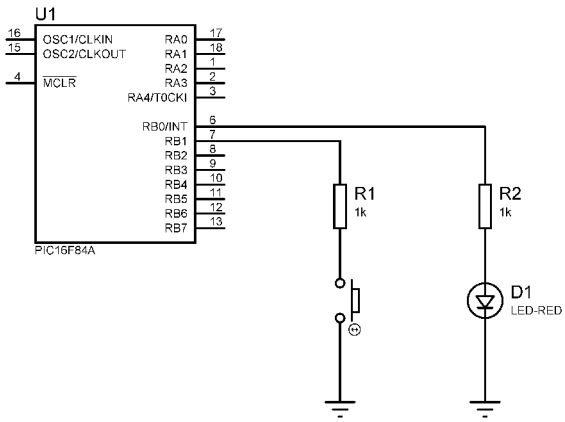
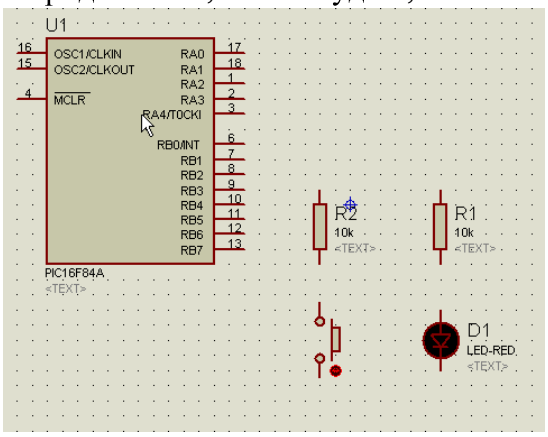


рис 20

Для начала определимся со списком деталей.
Итак нам нужны: микроконтроллер PIC16F84A 1 штука, светодиод красный 1 штука, кнопка и два резистора по 1 кОм каждый. Догадываюсь
вы уже спрашиваете,

- А как же кварц, конденсаторы и элементы питания?
Все это эмулируется программно, и нет необходимости добавлять их в схему. Хотя если вы будете разрабатывать проект до его логического конца, то есть до изготовления печатной платы, элементы придется добавить.
Вперед за компонентами! Откройте библиотеку компонентов. Чтобы вам долго не искать подскажу, наберите в окне KEYWORDS pic16f84a .Теперь либо дважды нажав ENTER , но тогда закроется библиотека и придется заново ее открывать, либо
дважды щелкнув левой кнопкой по строке с описанием компонента, появившейся в окне RESULTS (результат), вы переместите выбранный вами компонент в список Object Selector .Выберите, таким образом, резистор, набрав RES ,кнопку BUTTON и светодиод LED-RED .
Компоненты набираются по одному экземпляру, размножить их можно уже потом просто выбирая в списке Object Selector .Закройте библиотеку, нажав OK или же закрыв окно. Со временем к вам придет опыт, и вы будете, сами определять, какие нужны



компоненты и где они находятся. Надеюсь, все прошло без ошибок и в окне Object Selector появился список набранных нами компонентов. Если так то разместим их на схеме, щелкнув левой кнопкой сначала по названию компонента в списке, а затем в нужном нам месте на пустой пока еще схеме. Разместите и разверните, если это необходимо все компоненты. В итоге получится, что-то вроде этого.
Нам не хватает еще одного важного элемента – «земли» или «корпуса». Элементы такого типа (терминалы) выбираются в режиме INTER SHEET TERMINAL .

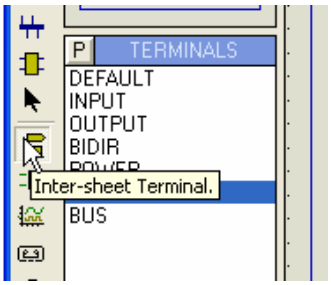


рис 22

Выберите элемент GROUND (земля) и поместите его на схеме под кнопкой и светодиодом. Теперь соедините компоненты меж собой как показано на схеме.
Как проводить соединения показывалось выше на примере конденсатора.
Измените сопротивление резисторов на 1 кОм. Также измените, тип модели резисторов на digital (цифровой), это необходимо, чтобы симулятор не тратил время на обсчет аналоговых свойств резисторов. Нам нужно только, горит или нет светодиод и нажата кнопка или нет, то есть чисто логические уровни.

Схема готова.

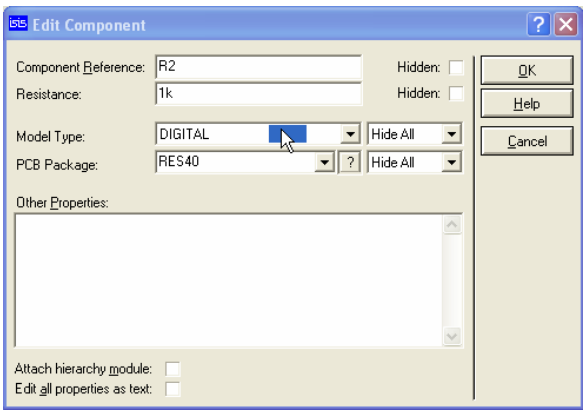


рис 23

Сохраните проект, в свою папку, чтобы не путаться, под именем LED.DSN .
Сделаем небольшое отступление. Если вы собираете схему на микроконтроллерах и хотите только отладить программу, не старайтесь точно воспроизвести схему источник, Насколько это возможно меньше применяйте аналоговые устройства или же старайтесь заменить их цифровыми примитивами. Многие модели имеют два варианта аналоговый и цифровой, например тот же резистор. Транзисторы, работающие ключами можно заменить либо инверторами, либо буферами исходя из их проводимости.
Это поможет разгрузить процессор. Конечно, PROTEUS имеет более продвинутые средства решения этой проблемы, например «магнитофон» TAPE, который позволяет

разбить схему на несколько частей записать сигналы с одной части в промежуточный файл, затем заморозить эту часть и использовать записанные сигналы для других частей, симулируя только выбранные не затрагивая остальных . Но этот вариант, если и будет рассмотрен, то только не сейчас. Так что старайтесь максимально упростить схему, тем более, если у вас не особо шустрый компьютер.

Попробуем оживить схему. Для этого нам необходим файл исходник. Среда PROTEUS поддерживает много средств разработки, среди них и HI-TECH Си компилятор и CROWHILL PIC BASIC и BASIC STAMP. И это только для микроконтроллеров фирмы MICROCHIP . Мы воспользуемся ассемблером MPASM .
В поставку PROTEUS входят компиляторы MPASM и MPASMWIN от MICROCHIP, но они старые, созданные еще в 2001г., и многие типы микроконтроллеров не поддерживают, так что придется заменить на новые.
Желательно от MPLAB версии 6.30 . В 6.50 уже нет MPASM . Я пользуюсь MPASM потому что MPASMWIN не поддерживает пути длинней 62 символа. Но MPASM , а возможно и MPASMWIN , поддерживают только формат имени файла 8.3 .Так что будьте внимательны, когда компилятор будет ругаться что не нашел файл. Надеюсь, что читатели уже имеют опыт общения с компилятором MPASM фирмы MICROCHIP .
Ну,с начнем.
Вот он, исходник. Набейте его в каком-нибудь текстовом редакторе . Я использую редактор MED, у него есть много просто необходимых функций, например создание своей схемы подсветки синтаксиса.
(Кстати к PROTEUS можно подключить внешний редактор, заменив встроенный. Для этого зайдите
в меню SOURCE пункт SETUP EXTERNAL TEXT EDITOR.
Нажмите BROWSE (просмотр) и найдите свой любимый редактор.)
Продолжим. Сохраните набитый файл, в папку с нашим проектом, под именем LED.asm.

```
        list p=16f84
        #include <p16F84A.inc>
       __CONFIG   _CP_OFF & _WDT_OFF & _PWRTE_ON &_HS_OSC


#define LED  PORTB,0
DelayL      equ 0x0C
DelayM      equ 0x0D
DelayH      equ 0x0E

            org 0h
            clrf DelayL
            clrf DelayM
            clrf DelayH
            clrf PORTA
            CLRF PORTB
            bsf STATUS,RP0
            clrf TRISA
            clrf TRISB
            bcf STATUS,RP0

start       bsf LED
```

```
            call Delay500
            bcf LED
            call Delay500
            goto start

Delay500    clrf DelayL
            clrf DelayM
            movlw 3h
            movwf DelayH
Wait1       decfsz DelayL
            goto Wait1
            decfsz DelayM
            goto Wait1
            decfsz DelayH
            goto Wait1
            return
            end
```

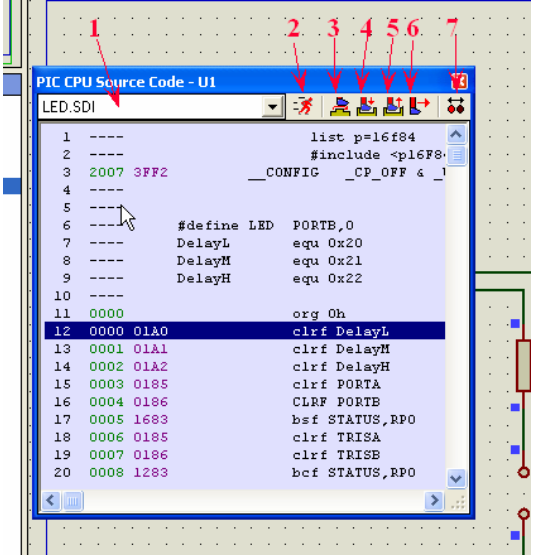
Добавим наш исходник в проект. Для этого в меню SOURCE(исходник) выберем ADD/REMOVE SOURCE FILE(добавить/удалить файл) . В появившемся окне нажмем кнопку NEW (новый). В строке SOURCE CODE FILINAME выберем наш исходник, с помощью кнопки CHANGE (сменить), а в строке CODE GENERATION TOOLS компилятор MPASM .Подтвердим наш выбор, нажав OK .
Соберем проект, то есть сотворим файл прошивки. Откроем меню SOURCE и нажмем BUILD ALL . Откроется лог компилятора, сообщая, что все в порядке если вы не допустили ошибок, или же появятся строки с ошибками.

Закроем окно лога. «Прошьем» микроконтроллер. Для этого в окне свойств микроконтроллера, как открыть это окно рассказывалось выше, в строке PROGRAMM FILE выберем файл прошивки LED.HEX , который появился в нашей папке после компиляции.

В строке PROCESSOR CLOCK FREQUENCY (тактовая частота процессора) выставьте 4 МГЦ. В строке PROGRAMM CONFIGURATION WORD (слово конфигурации) выставлять ничего не надо, если соответствующие данные есть в файле-исходнике. Остальные установки также изменять пока нет необходимости.
Сохраним еще раз проект.

Посмотрим на дело рук своих, запустив проект. Замигал светодиод? Все в порядке.

Конечно, просто созерцать на мигание проку мало, поэтому посмотрим, что нам предлагает PROTEUS в качестве инструментов отладки программы и просмотра внутренностей микроконтроллера. Поставьте проект на паузу. Появится окно отладчика,если этого не произошло, то в меню DEBUG отметьте пункт PIC CPU SOURCE CODE .



На рисунке под цифрами:

1 Выбор файла отладки, если было два и более исходника, один, например, на СИ или бэйсике а другой на ассемблере.

2 Продолжить выполнение программы.

3 Шаг без входа в подпрограмму.

4 Шаг с входом в подпрограмму.

5 Исполнять код до возврата из подпрограммы. Фактически остановка происходит на следующей за RETURN командой. Использовать эту опцию можно естественно, только находясь в подпрограмме.

6 Исполнять пока не будет достигнута выделенная полосой строка (курсор).

7 Триггер точек остановки (брейк поинт), то есть вкл /выкл точку.

Щелкнув правой кнопкой по окну отладчика можно изменить его настройки.
В появившемся меню есть такие опции:

GOTO LINE перейти на линию
GOTO ADDRESS перейти на адрес
FIND найти

TOGGLE (SET/CLEAR) BREAK POINT установить / очистить точку установки
ENABLE ALL BREAK POINT разрешить все точки остановки
DISABLE ALL BREAK POINT запретить все точки (не удалить !)
CLEAR ALL BREAK POINT удалить все точки

FIX-UP BREAKPOINTS ON LOAD зафиксировать (разрешить) точки при загрузке проекта

Отметьте эту опцию.

DISPLAY LINE NUMBERS показывать номера линий
DISPLAY ADDRESSES показывать адреса команд
DISPLAY OPCODES показывать опкод команд

SET FONT выбор шрифта

Чтобы видеть символы кириллицы выберите шрифт COURIER NEW , а для того чтобы все убралось в окне, поставьте размер шрифта 8.

SET COLOR выбор цвета текста, фона и т.д.

Поставьте точку остановки на строку с номером 0. Как это сделать, думаю, вы уже сами поняли. У номера строки появится красная точка, индикатор того, что точка остановки активна. Если вместо точки окружность, то значит, установленная точка не активна. Выключите проект и снова запустите. Программа прервалась именно там, где мы поставили точку. Красный треугольник (маркер) показывает на текущую строку. Поставьте свои точки. Потренируйтесь, запуская проект и используя опции пошаговой трассировки.

Но что это нам дает? Пока ничего, просто перемещаемся по коду не видя ни спецрегистров ни регистров пользователя.

- Спокойствие, только спокойствие! - как говаривал великий Карлсон, сейчас они появятся. В том же меню DEBUG отметьте пункт PIC CPU REGISTER .Теперь при трассировке кода вы сможете наблюдать содержимое спецрегистров.

-А как же регистры пользователя, спросите вы.

Есть два способа увидеть содержимое регистров пользователя и спецрегистров одновременно, первый не очень удобный отметьте в меню DEBUG пункт PIC CPU DATA MEMORY .Не удобный, потому что отображаются они все сразу, в виде таблицы и только во время паузы или на точке остановки.

Второй более продвинутый способ это WATCH WINDOW. Отметьте соответствующий пункт в меню DEBUG . Напоминаю, что выбор всех этих окон, возможен только на, находящемся в паузе или работающем проекте. Щелкните правой кнопкой по появившемуся окну WATCH WINDOW.

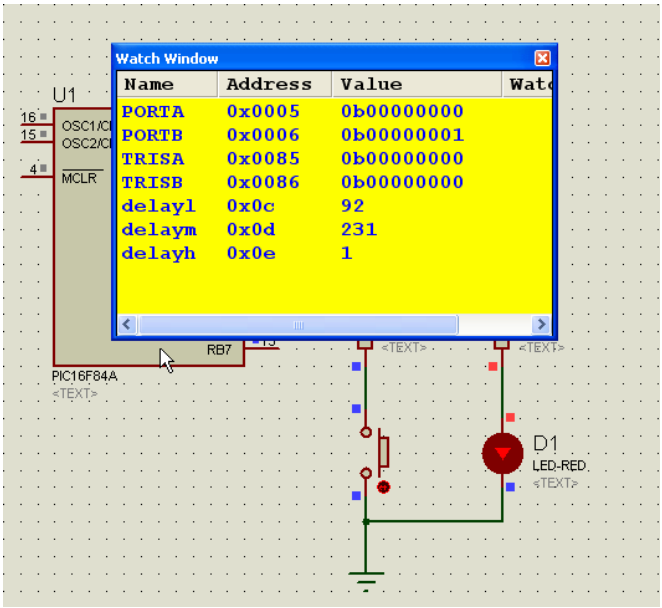
И так по пунктам:

ADD ITEMS (BY NAME)	добавить элемент по имени
ADD ITEMS (BY ADDRESSES)	добавить элемент по адресу
WATCHPOINT CONDITION	условие для остановки
SELECT ALL	выбрать все элементы
RENAME ITEM	переименовать
COPY CLIPBOARD	копировать в буфер обмена
DELETE ITEM	удалить выбранное
DATA TYPE	в каком виде представлять данные (строка, байт, слово и т.д.)
DISPLAY FORMAT	формат данных (двоичный, десятичный и т.д.)
SHOW ADDRESSES	показывать адрес
SHOW GRIDLINES	показывать сетку
SHOW WATCH EXPRESSIONS	показывать условие
MINIMUM SIZE	минимизировать размер

Щелкните по ADD ITEMS (BY NAME) , в появившемся списке выберите PORTA дважды щелкнув левой кнопкой по нему. Элемент добавится в окно WATCH WINDOW. Добавьте также PORTB , TRISA , TRISB . Когда закончите, закройте окно, нажав DONE . Добавьте DELAYL с адресом 0x0C , DELAYM 0X0D и DELAYH 0X0E .
Для этого выберете ADD ITEMS (BY ADRESSES) , в поле NAME введите имя регистра, например, DELAYL , в поле ADDRESS естественно адрес и выберите формат

USIGNED INTEGER (целое без знака). Нажмите ADD. Добавьте остальные регистры самостоятельно.
Теперь измените формат отображения данных регистров TRISA , TRISB , PORTA и PORTB на двоичный BINARY . Нам удобнее рассматривать эти регистры как отдельные биты (выводы).

Снимите все точки остановки с программы. Запустите проект.
Сейчас хорошо видно и изменение регистров в подпрограмме задержки и смена бита PORTB .

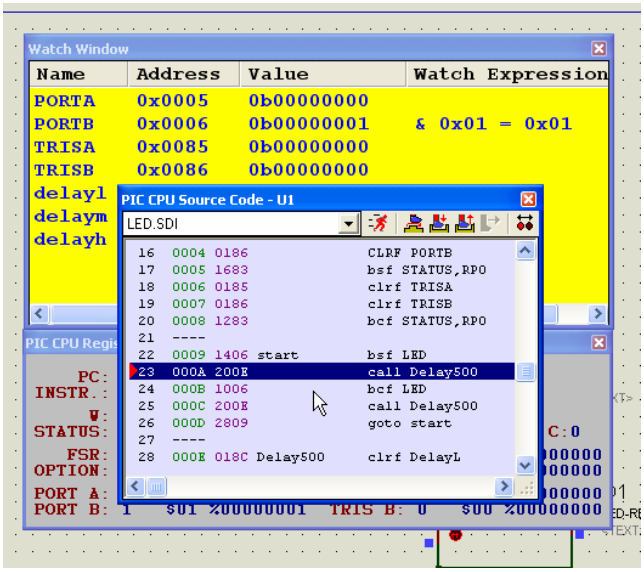


(рис prot_027)
Но и это далеко не все возможности. Добавим остановку по совпадению условия.
Выберите WATCHPOINT CONDITION в меню WATCH WINDOW.
По порядку:
Turn off(disable) watch point запретить остановку по условию
Suspend the simulation if any expression is true остановить симулятор если любое условие соблюдено .
Stop the simulation only when all expression is true остановить симулятор когда все условия соблюдены.

ITEM выбор регистра для которого применяются условие
MASK маска AND, OR , XOR и значение маски
CONDITION выбор условия (равно , больше чем , меньше чем и т.д.)
VALUE значение с которым сравнивается регистр.

Введите в поле ITEM PORTB ,в поле MASK AND 0X01 далее EQUALS(равно) 0X01. Таким образом, мы выставили условие симулятору остановиться, когда бит 0 PORTB буден равен 1. Также отметьте Suspend the simulation if any expression is true .
Запустите симулятор. Первая остановка внеплановая. Из-за того, что при старте в регистре был мусор. Продолжите симуляцию, теперь остановки будут проходить как нужно. И так, мы остановились. Посмотрите в окне отладчика, на какой команде встал симулятор.
Правильно, на CALL DELAY 500 , а предыдущая была BSF LED именно она и установила

бит 0 PORTB в 1.



Попробуйте сами ввести условие, например, остановиться, когда DELAYL равен 56
.
У нас осталась не функционирующей кнопка. Предлагаю вам самим дописать код.
И заодно пользуясь вышеописанными способами отладить программу.
Не забудьте сохранить проект.

Создайте новый проект с такими же как и у предыдущего размерами листа схемы. Наберите следующие элементы: PIC16F84 , 7SEG-MPX4CC-BLUE это семисегментный четырехразрядный индикатор с синим свечением, находится в категории OPTOELECTRONICS .

Подсоедините PORTA к катодам, а PORTB к сегментам.

Ну, как? Получилось нечто страшное.

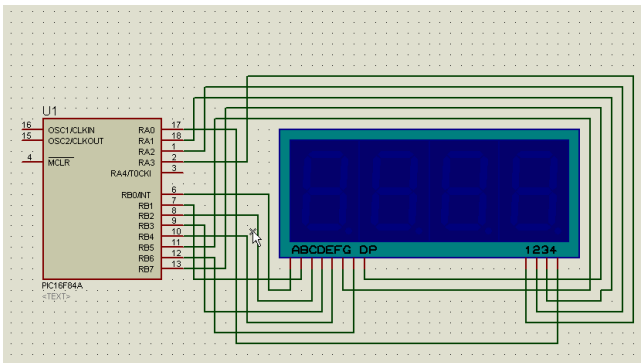


рис 29

Как же быть ? Вот если бы все провода связать в «пучок».Без проблем!
Удалим все соединения. Выберем режим BUS шина.

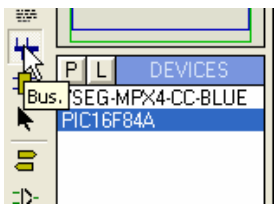


рис 30

Теперь отмечая левой кнопкой точки, проведем шину как на рисунке.
Конец шины отмечается щелчком правой кнопки.

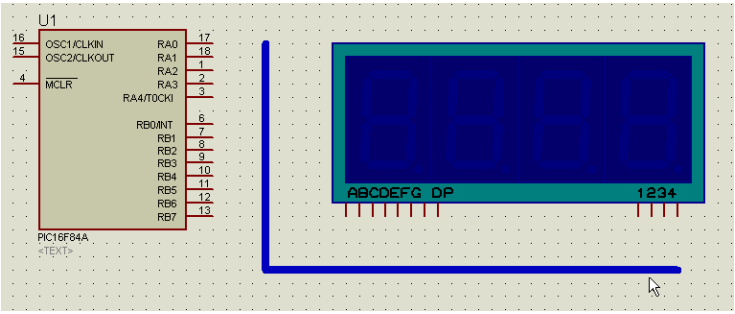


рис 31

Соедините выводы микроконтроллера и индикатора напрямую с шиной .
А как же определять, какой провод куда подключен? Для этого существуют метки.
Перейдем в режим WIRE LABLE (метка провода), щелкнув
по пиктограмме с надписью LBL.Начните с индикатора, щелкая по выводу, вы открываете окно ввода и редактирования метки. Чтобы не запутаться давайте имена
меткам в соответствии с именами выводов индикатора (A-A B-B 1-1 и т.д.)
Если не «попадаете» по проводу, увеличьте схему, покрутив колесо мыши или нажав F6.
Закончив с индикатором переходите к микроконтроллеру . Здесь уже нет необходимости вводить с клавиатуры символы . Все те символу которые вы ввели , уже присутствуют в списке. Нажмите в поле ввода кнопку выбора и вы сами все увидите.
Подключите так RA0-4…RA3-1 , RB0-A…..RB7-H(DP).
Теперь что бы убедиться, что все это работает, откомпилируйте этот исходник (MPX.ASM) и подключите полученную прошивку к проекту. Выставьте тактовую частоту 4 МГц .

```
        list    p=16F84A
        #include <p16F84A.inc>

        __CONFIG  _CP_OFF & _WDT_OFF & _PWRTE_ON &_HS_OSC


        CBLOCK  0x0C
  w_temp,status_temp,tik , scan                 ; Определяем регистры
  scantemp ,CAT_1 , CAT_2 ,CAT_3 ,CAT_4  ;  и переменные
        ENDC                    ;
OPTION_SETUP  EQU    B'00000110'          ; Резисторы подтяжки
                                          ; включены ,предделитель
                                          ;  к TMR0 1:16 , приращение
                                          ; от внутреннего источника
     ;*************************
     ;     СТАРТ (СБРОС)       *
     ;*************************

        ORG    0x000                      ; Вектор сброса
        goto   init


    ;*********************************
    ;       ПРЕРЫВАНИЕ                *
    ;                                 *
    ;*********************************
        ORG    0x004                          ; Вектор прерывания

        movwf  w_temp                         ; Сохраняем временно
        movf   STATUS,w                       ; STATUS регистр и
```

```
        movwf   status_temp                     ; аккумулятор

        INCF    tik,f                           ; Увеличим счетчик
                                                ; (пока он считает только
                                                ; прерывания)
        BCF     STATUS, RP0
        RLF     scan,W                          ; Движок мультиплексора
        BTFSS   scan,1                          ; индикатора
        MOVF    CAT_3,W                         ;
        BTFSS   scan,0                          ;
        MOVF    CAT_2,W                         ;
        BTFSS   scan,3                          ;
        MOVF    CAT_1,W                         ;
        BTFSS   scan,2                          ;
        MOVF    CAT_4,W                         ;
        MOVWF   scantemp                        ;
        CLRF    PORTB                           ;
        RLF     scan                            ;
        MOVF    scan,W                          ;

        ANDLW   0FH                             ; Маскируем 4
        MOVWF   PORTA                           ; разряд PORTA для
        MOVF    scantemp,W                      ; будущего
        MOVWF   PORTB                           ;

        movlw   0XAA                            ; В TMR0 костанту
        movwf   TMR0                            ; отсчета

        bcf     INTCON,T0IF                     ; Сбросим флаг чтобы
                                                ; при выходе из прерывания
                                                ; сразу же не попасть в него

        movf    status_temp,w                   ; Востановим сохраненые нами
        movwf   STATUS                          ; регистры и вернемся из
        swapf   w_temp,f                        ; прерывания
        swapf   w_temp,w                        ;
        retfie                                  ;


    ;*****************************************
    ;       ИНИЦИЛИЗАЦИЯ РЕГИСТРОВ          *
    ;*****************************************

init
                                        ;
        clrf    PORTA                   ;
        clrf    PORTB                   ; Очищаем спецрегистры
        clrf    TMR0                    ;
        bsf     STATUS, RP0             ;
        MOVLW   OPTION_SETUP            ;
        movwf   OPTION_REG              ;
        movlw   B'00000000'             ; и делаем все биты PORTB
        movwf   TRISB                   ; выходами
        movlw   B'11100000'             ; биты 0-4 PORTA выходы
        movwf   TRISA                   ;
        bcf     STATUS, RP0             ;
        clrf    INTCON                  ;
```

```
        movlw   0x0C                    ;
        movwf   FSR                     ;

clear_mem
                                        ; Чистим регистры
        clrf    INDF                    ; пользователя
        incf    FSR, F                  ;
        btfss   FSR, 6                  ;
        goto    clear_mem               ;
        clrf    FSR                     ;

        movlw   0EEh                    ; Грузим в scan
        MOVWF   scan                    ; первоначальное состояние
        movwf   PORTA                   ; катодов индикатора

        movlw  0XAA                     ; В TMR0 точку отсчета
        movwf  TMR0                     ;
        movlw  b'10100000'              ; Разрешаем прерывания
        movwf  INTCON                   ; от TMR0 и общее
        GOTO   MAIN                     ; и переходим на основной цикл


    ;*********************************
    ;       ОСНОВНОЙ ЦИКЛ             *
    ;*********************************


MAIN
        movlw   0x73            ; грузим значения для индикаторов
        movwf   CAT_1
        movlw   0X3F
        movwf   CAT_2
        movlw   0X78
        movwf   CAT_3
        movlw   0x6D
        movwf   CAT_4

LOOP    NOP
        NOP
        GOTO    LOOP            ;  и зацикливаемся  навсегда


        END
```

Запустите проект.
Если все в порядке на индикаторе появится надпись StOP с маленькой t .
Поставьте условие остановки WATCHPOINT CONDITION: INTCON , AND 0X04 , EQUALS , 0X04.
Теперь мы будем прерываться каждый раз, когда возникает прерывание по переполнению таймера TMR0, то есть при переходе его с FF на 00, перед самым переходом на обработку прерывания. Чтобы убедиться в этом, запустите проект. Мы встали на строке 0045 с меткой LOOP , таймер, как и следовало, ожидать, равен 0 , флаг прерывания TOIF (бит 3 INTCON) выставлен. Нажмите кнопку шаг и попадем прямиком на адрес 0004 , который как вы знаете, является вектором прерывания.

И на «сладкое» немного о инструментах.
Сначала выберем осциллограф. Включим режим VIRTUAL INSTRUMENTS и в появившемся списке выберем OSCILLOSCOPE.

Осциллограф двухканальный. Подключите канал А к катоду 1 , канал В к катоду 2. Запустите проект. Если у вас получилась другая картинка, установите ручки и переключатели в соответствии с рисунком. Поэкспериментируйте с органами управления осциллографом. Управление такое же, как и у реального. Так что, если возникнут вопросы, обращайтесь к соответствующей литературе.

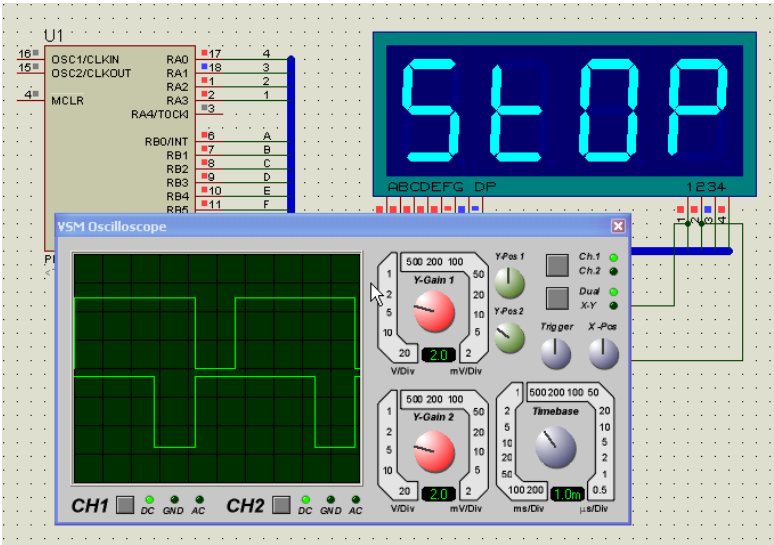


рис 32

Анализатор логических уровней.

Выберете режим инструментов.

В окне выбора щелкните по LOGIC ANALISER .Разместите и подсоедините его как показано на рисунке.

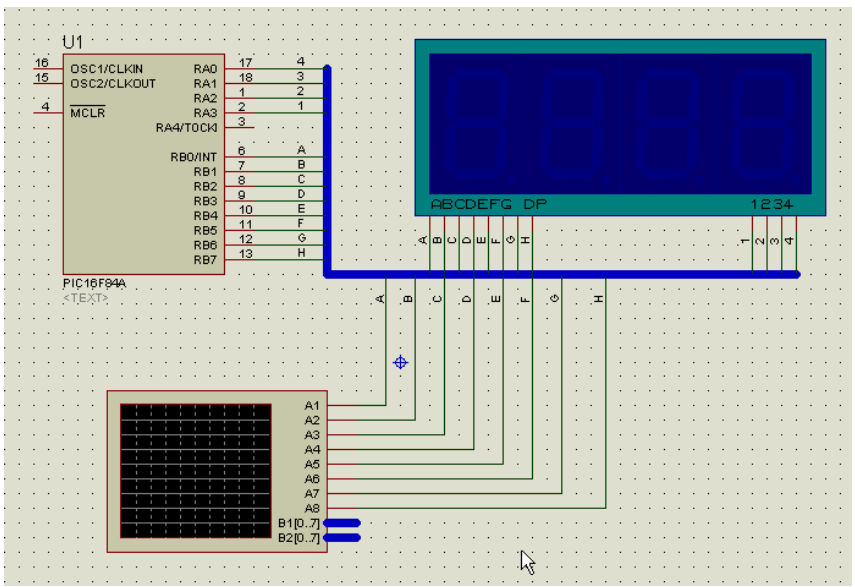


рис 33

Поставьте проект на паузу, нажав соответствующую кнопку. Настройте анализатор как показано на рисунке и запустите проект, нажав ПУСК.

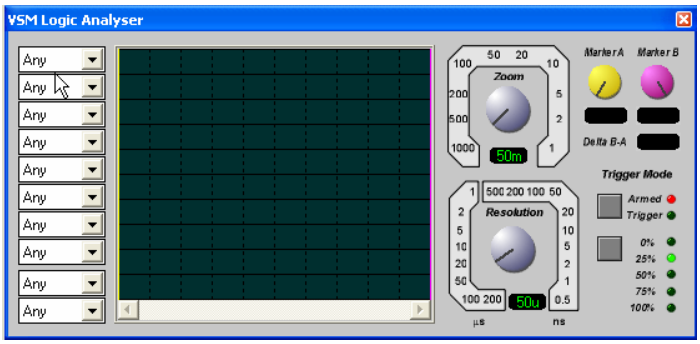


рис 34

Триггер сработает, и на экране анализатора появится следующие:

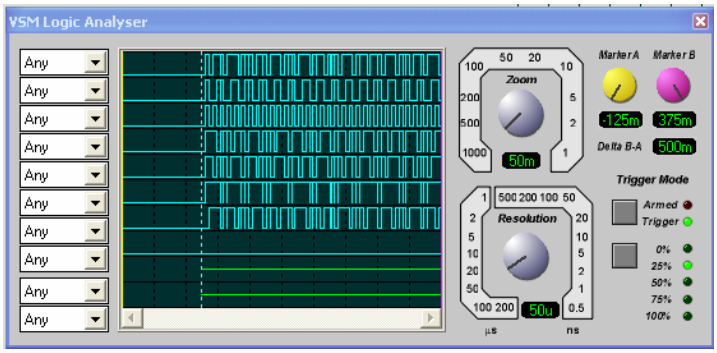


рис 35

Для чего нужен анализатор, теперь думаю объяснять не надо.

К анализатору можно подключать восемь одиночных каналов и две шины.

Слева экрана находятся окна выбора условий для срабатывания триггера. Восемь для каналов и два для шин. Для каналов условия определяются логическими уровнями, а для шин шестнадцатеричным числом от 0 до FF. Триггер сработает если все условия совпадут. Регуляторы справа экрана: нижний это выбор разрешения, верхний -увеличение экрана просмотра .Кнопка с надписью Armed Trigger взвод триггера . Под ней кнопка выбора заполнения буфера анализатор до и после точки срабатывания триггера , которая обозначена на экране пунктирной вертикальной линией .Два регулятора сверху перемещают временные маркеры , оба равнозначны, но имеют разные цвета . В окошках под регуляторами отображается время от начала буфера до маркера. В окошке с надписью Delta B-A показана разность значений верхних окон, то есть «расстояние» от одного маркера до другого.

С использованием оставшихся инструментов попробуйте разобраться сами.
Экспериментируйте, не бойтесь испортить чего-либо.

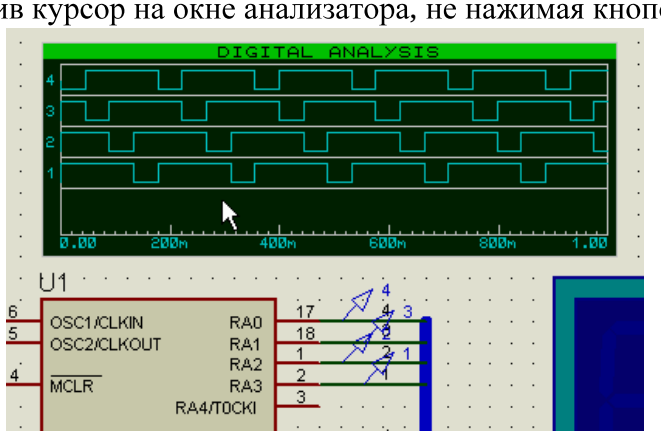
Есть более мощные средства анализа, чем описанные выше осциллограф и логический анализатор. Собранны они в группу SIMULATION GRAPH. Описание всех их и их возможностей займет очень много времени. Поэтому приведу лишь небольшой пример, который поможет вам разобраться в принципе использования анализаторов.

Выберите режим SIMULATION GRAPH. В списке щелкните по DIGITAL (цифровой).

Теперь щелкнув в пустом месте схемы, не отпуская левой кнопки, растяните прямоугольник. Это окно анализатора. Сейчас нам нужно добавить пробники. Включите режим VOLTAGE PROBE . Разместите пробники на проводах идущих к катодам

индикатора, просто щелкая по проводам левой кнопкой. Подсоединяясь, пробники, автоматически принимают метку провода. Затем перетащите по одному пробники в окно анализатора, сначала выделив пробник правой кнопкой, а затем, схватив левой бросить в окне.

Если не получается, попробуйте увеличить схему.
Теперь поместив курсор на окне анализатора, не нажимая кнопок мыши, нажмите **пробел**.



(рис prot_036)

Чтобы развернуть или наоборот свернуть окно анализатора щелкните левой кнопкой по зеленой полосе вверху окна.

На этом пока все.
Максимов Алексей aka Dosikus aka Maksimus

симулятор – отладчик

PROTEUS VSM

ISIS

На основе ядра SPICE3F5

Labcenter Electronics Co.