**ЗАДАЧА 1**

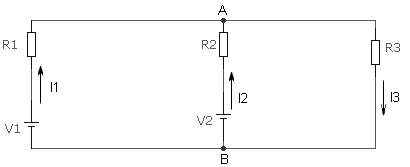
***Часть 1***

***«Решение систем уравнений в Excel на примере расчета токов в ветвях электрической цепи с использованием законов Кирхгофа»***

**Пример 1**

**Задание:** Определите токи в каждой ветви цепи постоянного тока, (см. рис. ниже), если указаны следующие величины:

ε1 ε2



**Решение.**

Направления токов выбраны условно, если после решения значения каких-то токов получатся отрицательными, значит реальное направление для данного тока противоположно выбранному.

Т.к. ветвей вцепи три, составим три уравнения. Первое – по первому закону Киргофа – для узла А, и два – по второму закону Киргофа - для контура, по которому протекают токи I1 и I3, и для контура с токами I2 и I3. В обоих случаях направление обхода контура выберем по часовой стрелке.

Тогда получим следующую систему уравнений:



Подставляя известные числовые параметры (сопротивления и ЭДС), получим такую систему трёх уравнения с тремя неизвестными токами:



Решим эту систему методом Крамера. Этот метод решения систем линейных алгебраических уравнений подходит для систем с ненулевым определителем основной матрицы. Эта матрица строится по коэффициентам левых частей выровненных (т.е. с учётом, что некоторые коэффициенты в некоторых уравнениях могут быть равными нулю) уравнений, и имеет вид:



Составляем матрицы переменных:

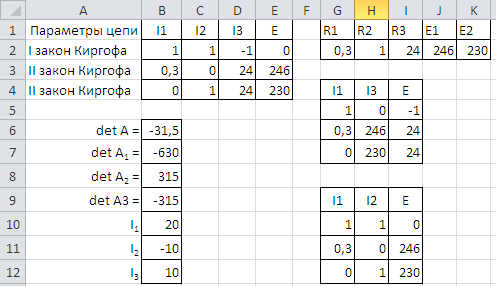
Далее вычисляем определители этих матриц, используя навыки применения функции МОПРЕД, полученные при выполнении предыдущей лабораторной работы.

Затем находим отношения определителей матриц переменных к основной матрице, которые и будут искомыми значениями токов в ветвях:

I1 = detА1/detА; I2 = detА2/detА; I3 = detА3/detА.

**Задание 1**: Рассчитать I1 I2 I3

На листе Excel это должно выглядеть примерно так:



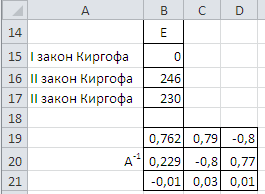
Здесь в ячейках B2:E4 – ссылки на соответствующие значения параметров цепи, записанные в ячейках G2:K2. В ячейках G5:I7 и G10:I12 – матрицы A2 и A3. В ячейках B6:B9 – формулы для вычисления соответствующих определителей, например   
det A1 =МОПРЕД(C2:E4). И в ячейках B10:B12 – формулы для искомых токов.

Примечание: Как следует из решения, реально ток I2 направлен в противоположную условно выбранной сторону, что говорит о том, что второй источник ЕДС работает в режиме потребителя.

***Докажем верность вычислений.***

Нужно подставить в исходную систему уравнений найденные значения токов, и убедиться, что в правых частях 2-го и 3-го уравнений системы будут действительно суммы ЭДС, входящих в контур контуров, а 1-го – 0 (сумма источников токов, (не)входящих в узел).

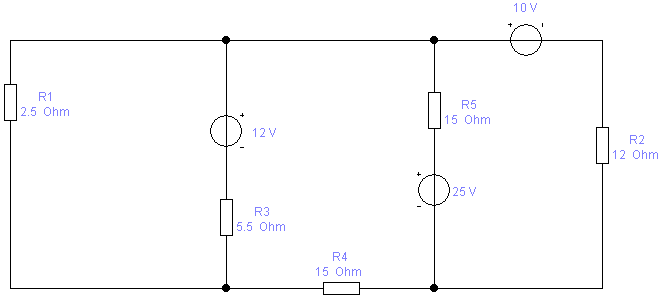
На листе Excel это должно выглядеть примерно так:



Например, в ячейке B16 – формула для левой части второго уравнения, имеющая вид: =B10\*G2+B12\*I2.

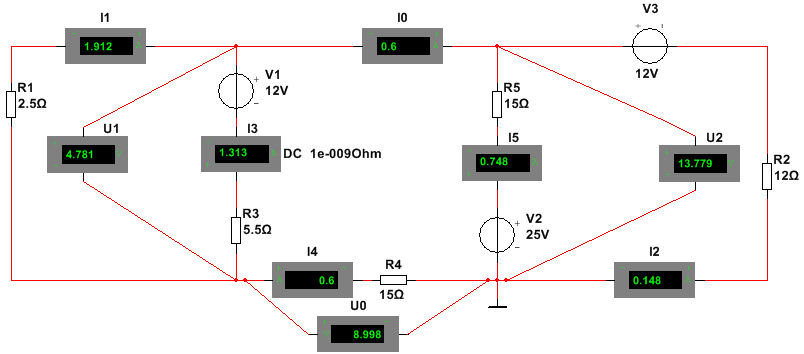
**Пример 2**

Схема

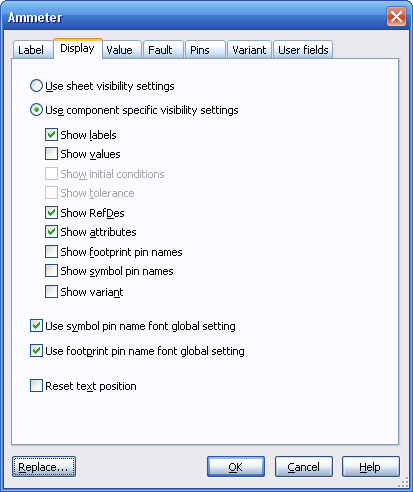


**Решение – схема в программе Multisim**

«Собранная» цепь с измерительными приборами имеет вид:

****

Примечание: для избавления от ненужных надписей (внутр. сопротивление приборов) амперметры и вольтметры настроены следующим образом:



**Решение в Excel.**

Направления токов выбраны условно, если после решения значения каких-то токов получатся отрицательными, значит реальное направление для данного тока противоположно выбранному.

В цепи 3 узла (A=A') Т.к. ветвей в цепи 5, составим 5 уравнений Первое – по первому закону Киргофа – для узла А, и два – по второму закону Киргофа - для контура, по которому протекают токи I1 и I3, и для контура с токами I2 и I3. В обоих случаях направление обхода контура выберем по часовой стрелке.

Тогда получим следующую систему уравнений:



Подставляя известные числовые параметры (сопротивления и ЭДС), получим такую систему трёх уравнения с тремя неизвестными токами:



Решим эту систему методом Крамера. Этот метод решения систем линейных алгебраических уравнений подходит для систем с ненулевым определителем основной матрицы. Эта матрица строится по коэффициентам левых частей выровненных (т.е. с учётом, что некоторые коэффициенты в некоторых уравнениях могут быть равными нулю) уравнений, и имеет вид:



Составляем матрицы переменных:







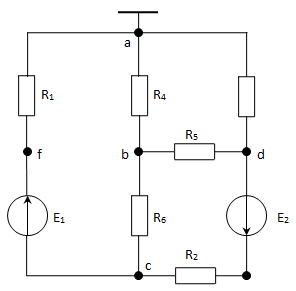
Далее вычисляем определители этих матриц, применяя функцию МОПРЕД.

Затем находим отношения определителей матриц переменных к основной матрице, которые и будут искомыми значениями токов в ветвях:

I1 = А1/А; I2 = А2/А; I3 = А3/А.

**Пример 3**

Схема



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Параметры цепи | | | | | | | |
| R1 | R2 | R3 | R4 | R5 | R6 | E1 | E2 |
| Ом | | | | | | В | |
| 10 | 18 | 5 | 10 | 8 | 10 | 20 | 35 |

Первый закон Кирхгофа: 

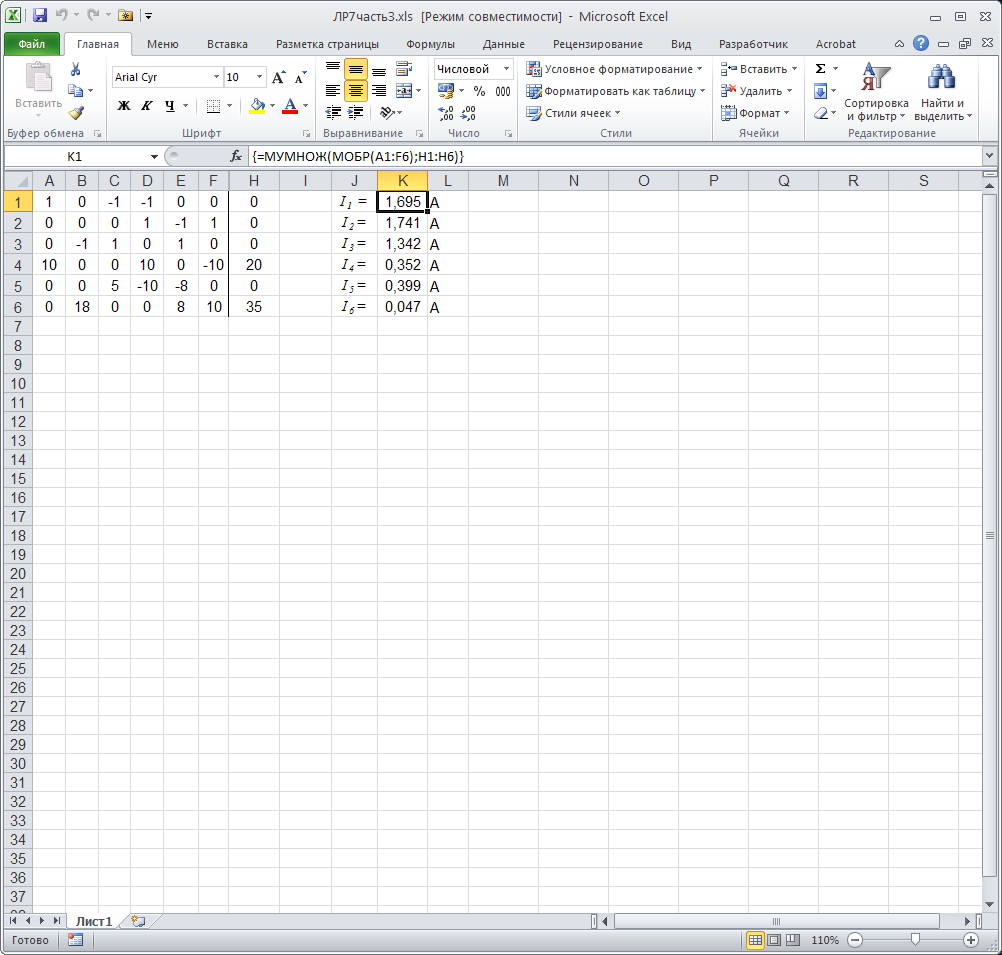
Второй закон Кирхгофа: 

Для вычисления искомого вектора неизвестных токов умножим матрицу, обратную главной матрице системы А, на столбец свободных членов Е:

I = А-1\*Е, используя функции массива МОБР (MINVERSE) и МУМНОЖ (MMULT) (вложив МОБР в МУМНОЖ). Для этого выделите ячейки K1:K6, введите формулу =МУМНОЖ(МОБР(A1:F6);H1:H6), и удерживая зажатой клавиши Ctrl и Shift, нажмите клавишу Enter.

Результат:



Проведём проверку. Выделим ячейки H8:H13 и введём формулу =МУМНОЖ(A1:F6;K1:K6).

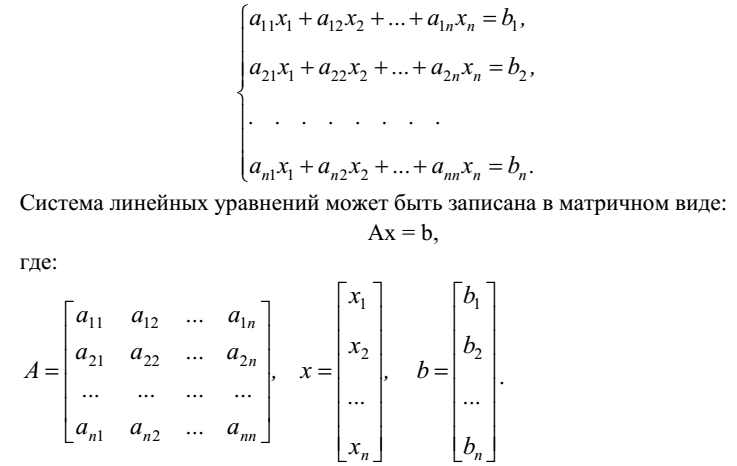
Удерживая зажатой клавиши Ctrl и Shift, нажмите клавишу Enter. В результате в ячейках H8:H13 должен получится столбец значений, совпадающих со значениями ячеек H1:H6.

***Примечание***: для округления значений выделите ячейки, в контекстном меню (открывающемся при нажатии правой кнопкой мыши) выберите «Формат ячеек» числовой и укажите число знаков после запятой.

**Решение систем уравнений в MathCad**

*Решение систем уравнений матричным методом*

Рассмотрим систему n линейных алгебраических уравнений относительно n неизвестных х1, х2, …, хn:



Если det A0 то система или эквивалентное ей матричное уравнение имеет единственное решение.

***Пример 1.*** *Решение систем уравнений с помощью функции Lsolve*

Системы линейных уравнений удобно решать с помощью функции lsolve. Функция lsolve(А, b) - возвращает вектор решения x такой, что Ах = b.

*Решим систему уравнений*





***Решение систем уравнений с помощью функций Find или Minerr***

Для решения системы уравнений с помощью функции Find необходимо выполнить следующее:

1. Задать начальное приближение для всех неизвестных, входящих в систему уравнений. MathCAD решает систему с помощью итерационных методов;
2. Напечатать ключевое слово Given. Оно указывает MathCAD, что далее следует система уравнений;
3. Введите уравнения и неравенства в любом порядке. Используйте [Ctrl]= для печати символа =. Между левыми и правыми частями неравенств может стоять любой из символов <, >,≤  и ≥**;**
4. Введите любое выражение, которое включает функцию Find, например: х:= Find(х, у).
5. Ключевое слово Given, уравнения и неравенства, которые следуют за ним, и какое - либо выражение, содержащее функцию Find, называют блоком решения уравнений.

Функция *Minerr* очень похожа на функцию *Find* (использует тот же алгоритм). Если в результате поиска не может быть получено дальнейшее уточнение текущего приближения к решению, *Minerr* возвращает это приближение. Функция *Find* в этом случае возвращает сообщение об ошибке. Правила использования функции *Minerr* такие же, как и функции *Find*.

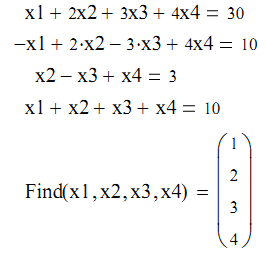
Функция **Minerr(x1, x2, . . .) -** возвращает приближенное решение системы уравнений. Число аргументов должно быть равно числу неизвестных.

Ключевое слово *Given*, уравнения и неравенства, которые следуют за ним, и какое - либо выражение, содержащее функцию *Find*, называют **блоком решения уравнений**.

***Пример 3.***  *Решение системы уравнений с помощью функции Find*

x1 := 0 x2 := 0 x3 := 0 x4 := 0 Начальные приближения

Given



**ЗАДАЧА2**

**Определение вида зависимости между полученными данными с использованием встроенных функций Microsoft Excel.**

Рассмотрим построение линии регрессии с помощью MS Excel на примере следующей задачи. Известна табличная зависимость G(L). Построить линию регрессии и вычислить ожидаемое значение в точках 0, 0.75, 1.75, 2.8,4.5.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **L** | 0 | 0,5 | 1 | 1,5 | 2 | 2,5 | 3 | 3,5 | 4 |
| **G** | 1 | 2,39 | 2,81 | 3,25 | 3,75 | 4,11 | 4,45 | 4,85 | 5,25 |

Введем таблицу значений на лист MS Excel и построим точечный график (Вставка → Диаграммы → Точечная). Рабочий лист примет вид, изображенный на рис. 2.

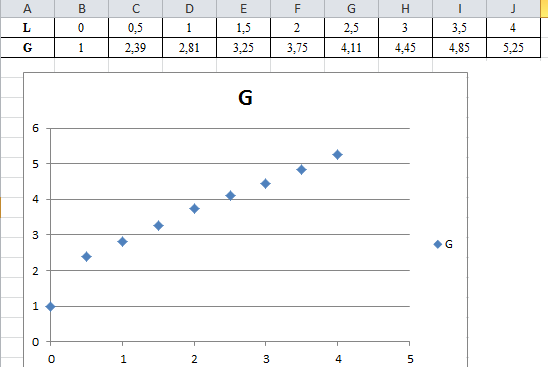


Рисунок 2

Для того, чтобы рассчитать значения коэффициентов регрессии A и B выделим ячейки К2:L2, обратимся к мастеру функций и в категории Статистические выберем функцию ЛИНЕЙН. Заполним аргументы функции так, как показано на рис.3.

Для введения функции массива одновременно в несколько ячеек необходимо, после ввода функции, удерживая Ctrl+Shift нажать Enter. Результат приведён на рис.4.

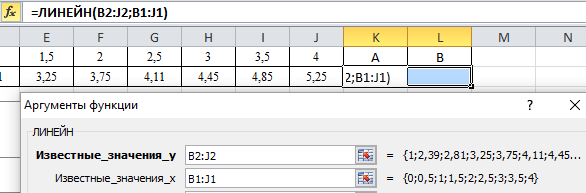


Рисунок 3

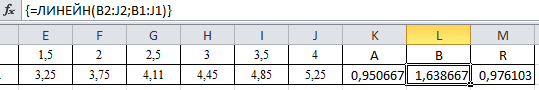


Рисунок 4

Для расчета значения коэффициента корреляции R в ячейку M2 была введена следующая формула: =КОРРЕЛ(B1:J1;B2:J2).

Для вычисления ожидаемого значения в точках 0; 0,75; 1,75; 2,8; 4,5 занесем их в ячейки L9:L13. Затем выделим диапазон ячеек M10:M13 и введем формулу:

= ТЕНДЕНЦИЯ(B2:J2;B1:J1;L9:L13). (см. рис.6)

Изобразим ожидаемые значения на диаграмме. Для этого выделим экспериментальные точки на графике, щелкнем правой кнопкой мыши и выберем команду «Выбрать данные». В появившемся диалоговом окне для добавления линии регрессии щелкнем по кнопке Добавить (см. рис.5).

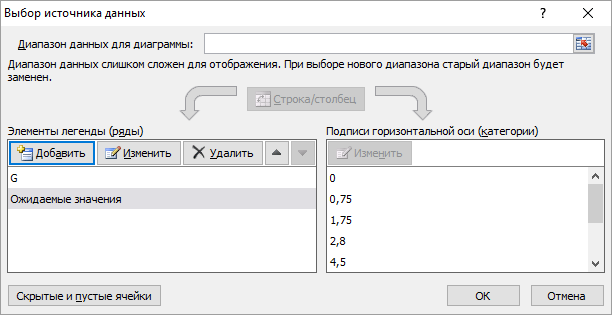


Рисунок 5

В качестве имени введем «Ожидаемые значения», в качестве Значения Х: L9:L13, в качестве Значения Y: M9:M13 (см. рис.6). Далее выделяем линию регрессии, для изменения ее типа щелкаем правой кнопкой мыши и выбираем команду Тип диаграммы.

После форматирования графика он примет вид, похожий на изображенный на рис.6:

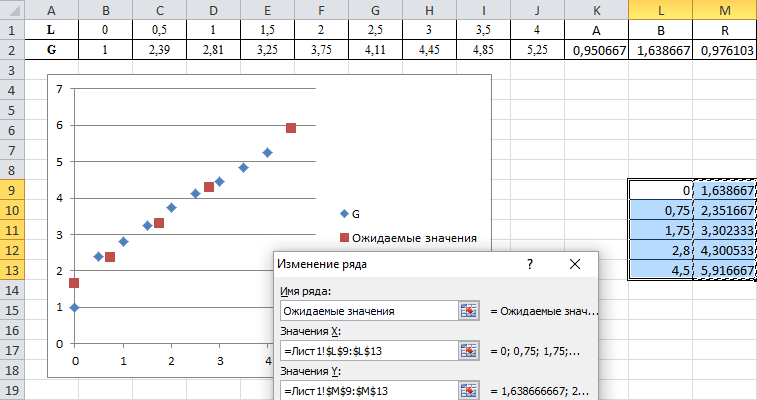


Рисунок 6

Теперь добавим на диаграмму линию регрессии (тренда). Для этого выделим экспериментальные точки на графике, щелкнем правой кнопкой мыши и выберем команду «Добавить линию тренда» В открывшемся окне «Формат линии тренда» выберите следующие параметры (см. рис.7):

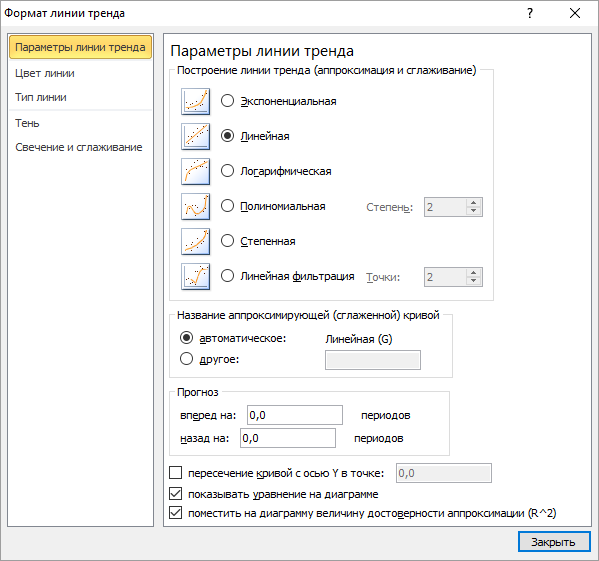


Рисунок 7

***Внимание:*** *для вариантов с полиномом степени 2 и выше коэффициенты выбирается тип линии «Полиномиальная» с указанной в задании степенью!*

Результат представлен на рис.8

Рисунок.8

Как видно, полученные коэффициенты совпадают с рассчитанными с помощью функции ЛИНЕЙН.

Теперь найдём степень расхождения данных, полученных с помощью полученных коэффициентов с экспериментальными данными.

В третьей строке листа введите формулы для прямой, построенной по известным коэффициентам a и b. Для этого в B3 введите формулу вида =$K$2\*B$1+$L$2 (с абсолютной адресацией для ссылок на ячейки, содержащие a и b. В четвёртой строке посчитайте корреляцию как разность между рассчитанными и экспериментальными значениями. Скопируйте полученные формулы в остальные ячейки 3-й и 4-й строк таблицы. Результат должен получиться таким:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| L | 0 | 0,5 | 1 | 1,5 | 2 | 2,5 | 3 | 3,5 | 4 |
| G | 1 | 2,39 | 2,81 | 3,25 | 3,75 | 4,11 | 4,45 | 4,85 | 5,25 |
| Проверка | 1,64 | 2,11 | 2,59 | 3,06 | 3,54 | 4,02 | 4,49 | 4,97 | 5,44 |
| Корреляция | 0,64 | -0,28 | -0,22 | -0,19 | -0,21 | -0,09 | 0,04 | 0,12 | 0,19 |

Далее (в строке 5) найдите квадраты разностей (по формулам вида =B4^2) и просуммируйте их. Результат должен получиться таким (см. рис.9):

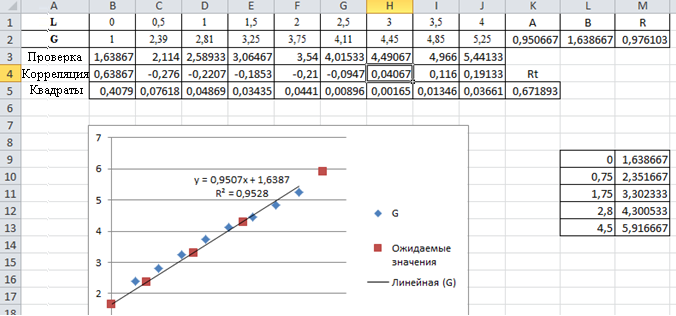


Рисунок 9

**ЗАДАЧА 3**

**Нахождение коэффициентов зависимости с помощью блока «Поиск решения»**

***Внимание:*** *исходные данные по вариантам для данной части задания те же, что и для предыдущих!*

Пусть в результате эксперимента получена следующая зависимость Z(T)



Необходимо подобрать коэффициенты зависимости Z(t)=At4+Bt3+Ct2+Dt+K методом наименьших квадратов.

Эта задача эквивалентна задаче нахождения минимума функции пяти переменных



Введем табличную зависимость в рабочий лист MS Excel и построим график функции (см. рис. 10).

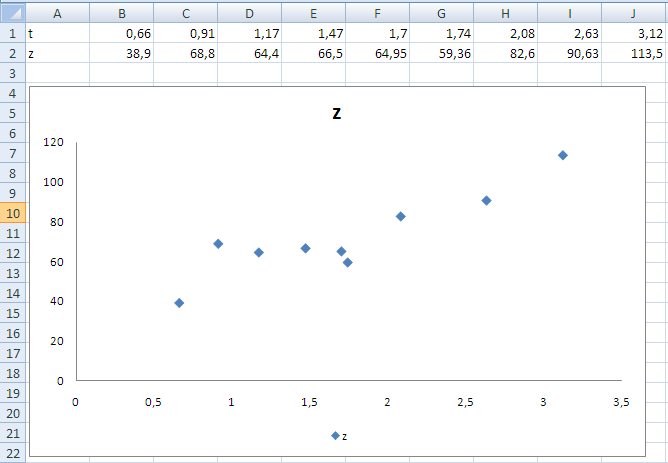


Рисунок 10

Рассмотрим процесс решения задачи оптимизации. Пусть значения А, В, С, D и К хранятся в ячейках K1:K5. Для этого в ячейки K1:K5 введем произвольные числа. Теперь в 23-ю строку введем значения функции . В ячейку **B23** введем значение функции в первой точке (ячейка B1) **=$K$1\*B1^4+$K$2\*B1^3+$K$3\*B1^2+$K$4\*B1+$K$5** (см. рис. 11). Получим ожидаемое значение в точке **B1**. Затем растянем эту формулу на весь диапазон **B23:J23**. В 24-ю строку введем квадраты разности между экспериментальными и расчетными точками. В ячейку **B24** введем формулу **=(B23-B2)^2** и растянем эту формулу на весь диапазон **B24:J24**.В ячейке **В25** будем хранить суммарную квадратичную ошибку. Для этого введем формулу **=СУММ(B24:J24)**.

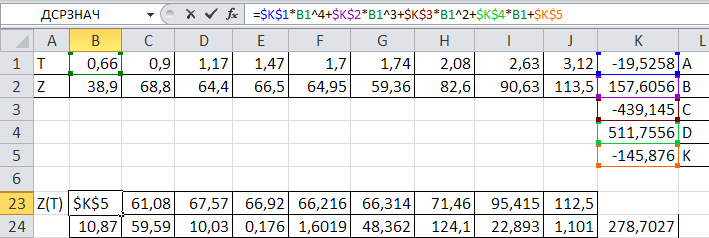


Рисунок 11

Теперь осталось с помощью блока **«Поиск решения»** решить задачу оптимизации без ограничений. Этот блок установит минимум в ячейке **B25** (формула ) изменяя содержимое ячеек **K1:K5** (переменные А, В, С, D и К) (см. рис. 12).

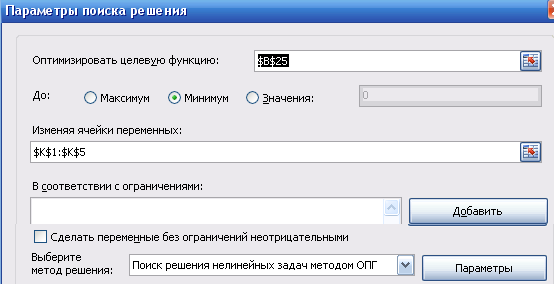


Рисунок 12

После этого в ячейках **K1:K5** получим значения параметров функции А, В, С, D и К функции . В ячейках **B23:J23** получим ожидаемые значение функции в исходных точках. Поместим эти точки в виде отдельной линии на графике. В ячейке **B25** будет храниться суммарная квадратичная ошибка. В результате рабочий лист примет вид (см. рис. 12).

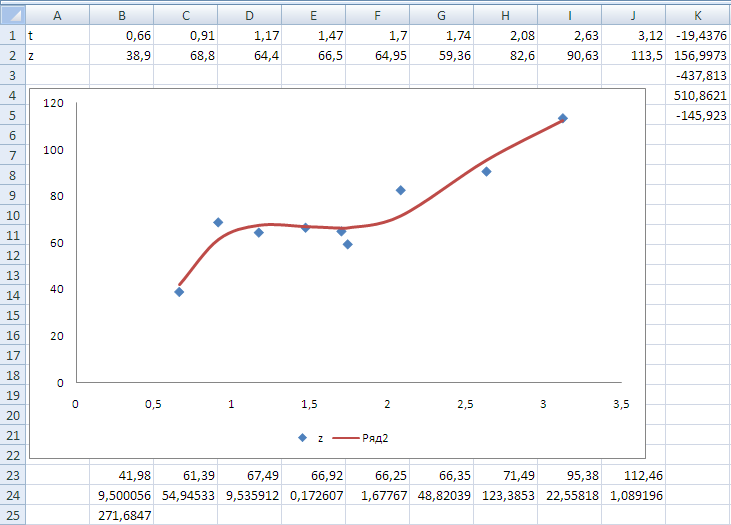


Рисунок 12

* ***На рабочих листах должны быть графики экспериментальной и рассчитанных теоретических зависимостей.***

**Примечание:**

В Excel 2003 и более ранних версиях блок «Поиск решения» вызывается из пункта меню «Сервис». При отсутствии его нужно установить: Сервис → Надстройки в окне Надстройки, установите флажок «Поиск решения».

Для включения надстройки «Поиск решения» в Excel 2007 и более поздних версиях в меню Файл выберите пункт «Параметры», перейдите на вкладку «Надстройки», выделите пункт «Поиск решения» и нажмите на кнопку «Перейти». Кнопка «Поиск решения» появится на ленте «Данные» (см. рисунки 13 и 14).

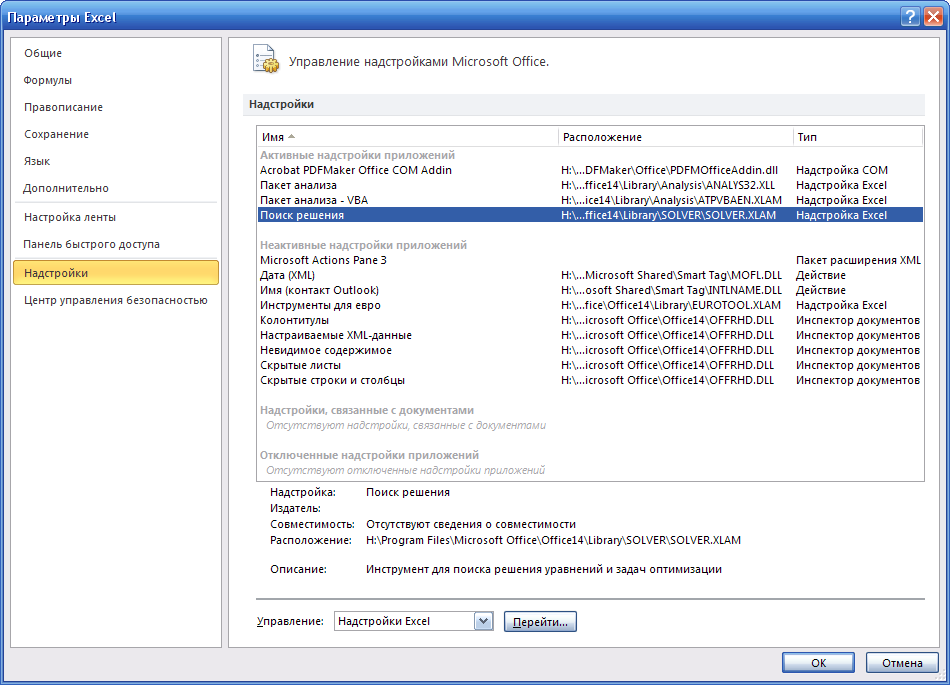


Рисунок 13

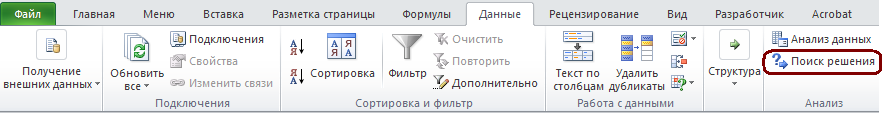


Рисунок 14

**ЗАДАЧА4**

**Моделирование схем с микроконтроллерами в программе Proteus.Isis**

**Программирование микроконтроллеров в программе CodeVisionAVR.**

***Ход работы***

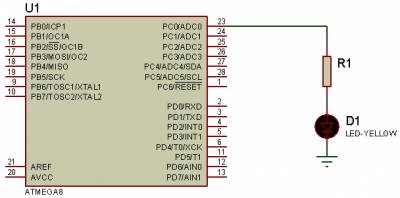
**1. Создание схемы в Proteus.Isis.**

Схема для эксперимента приведена на рис.1.

Наш светодиод рассчитан примерно на силу тока примерно в 20 мА. При этом на светодиоде падает примерно 2 В. Остаётся: 5 В (напряжение VCC) – 2 = 3 В.

По закону Ома I = U/R. Тогда R= 150 Ом.

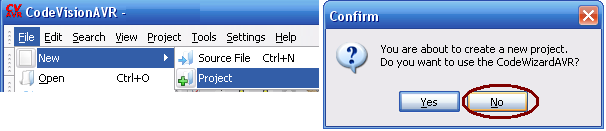
Вычисленное значение R и вводим в качестве параметра Resistance свойств R1.

[](http://herozero.do.am/_nw/0/85316891.png)

**2. Создание программного кода в *CodeVisionAVR***

Для того, чтобы микроконтроллер работал, ему нужен программный файл – «прошивка». Создадим программу в среде программирования для микроконтроллеров AVR CodeVisionAVR. Откройте в главном меню CodeVisionAVR в группе HP InfoTech .

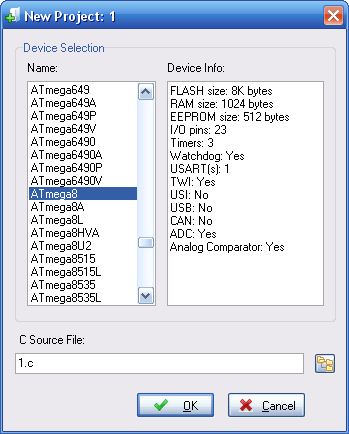
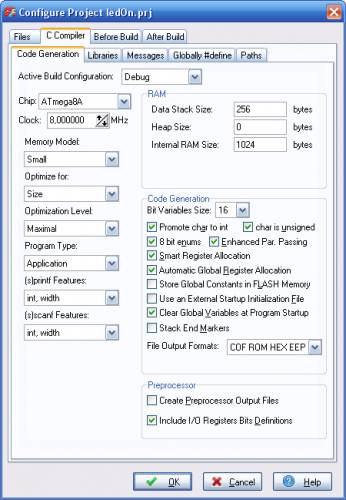
Создаём проект без мастера.



Сохраните проект в своей папке.

Выбираем наш chip ATMega8A.

После создания проекта откроется окно его конфигурации (его всегда можно открыть так: Project → Configure → C Compiler). Убедимся, что тактовая частота (Clock) = 8 MHz, остальное тоже оставляем дефолтное.

 [](http://herozero.do.am/_nw/0/12558848.png)

Комментарии (неисполняемый код) пишутся после двух слешей: // Управление портом С

Подключаем заголовочные файлы директивами

#include <io.h> //(ввод/вывод) и

<delay.h> // (задержка).

Пишем главную функцию

void main(void)

 {

   DDRC=0b11111111; // Data Direction Registr - регистр-переключатель направления данных порта С.

К DDR подключены пины порта, он определяет их направление выдачи сигнала.

1 – пин настроен на вывод сигнала.

0 (по умолчанию) – на вход (считывание состояние).

0b – в двоичной системе.

Все единицы – значит, все пины порта настроены на вывод сигнала.

Выбор направления данных можно задать и 16-ричными цифрами, тогда получим равносильный оператор DDRC = 0xFF (вспомним тетрады: 1111(=F) 1111(=F)).

Итак, пишем инструкции в программе.

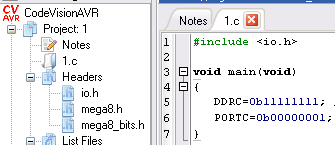
Для начала научимся подавать напряжение:

PORTC=0b00000001; // подаём на нулевой (младший) пин логическую единицу (5 В), а на остальных оставляем логический 0.

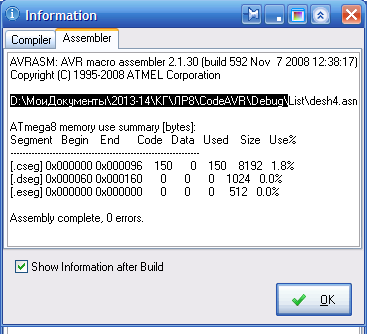
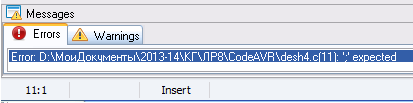
Пока все, закрываем программу

}

Вот что, по минимуму, должно получиться:



Компилируем код (**F9**). Если всё хорошо, увидим диалог с информацией о программе, если нет – то, кроме диалога и список ошибок в нижней части окна программы.

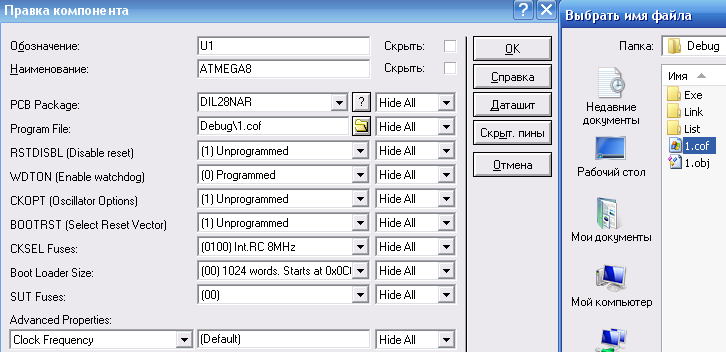
 

Исправим (если они есть) ошибки (дабл-клик по описанию ошибки выделит некорректную строку, либо следующую за ней, если предыдущую забыли закрыть точкой с запятой) и построим (по **Ctrl+F9**) hex-файл прошивки (попутно создаются и отладочные файлы в других форматах, напр. cof – для отладки программ CVAVR).

Бинарный файл .hex (который исполняется МК) будет в папке \Debug\Exe, а .cof – в \Debug.

**3. Прошивка МК в Proteus.Isis.**

В свойствах МК ATMega8 изменяем только CKSEL Fuses – выставляем в 0100, что задаёт тактирование МК от внутреннего RC-генератора на частоте 8MHz, и указываем в качестве ProgramFile созданный в CodeVisionAVR .cof–файл (выбор .cof -файла вместо .hexпозволит вести отладку по исходному коду):

****

Всё, схема должна заработать – должен загореться светодиод.

**4.** После того, как мы научились зажигать светодиод, сможем его и потушить.

Для этого в исходном коде добавьте команду задержки после включения

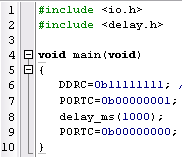
delay\_ms(1000); // пауза в 1 сек.

Но для того, чтобы она заработала, необходимо подключить библиотеку, в которой она прописана. Для этого в начале кода добавьте

#include <delay.h>

После задержки отключите светодиод сбросом в 0 младшего разряда порта С командой

PORTC=0b00000000;



Перепрошейте МК и проверьте в Протеусе работу схемы – светодиод должен включиться на секунду и потухнуть.

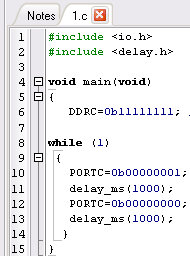
**5.** Мигание светодиода можно организовать в бесконечном цикле

while (1)

  {

  }

Перенесите в него команды установки напряжения и задержки (конфигурирование направления данных оставьте перед циклом), добавьте задержку после гашения светодиода и проверьте результат (мигание) в Протеусе.

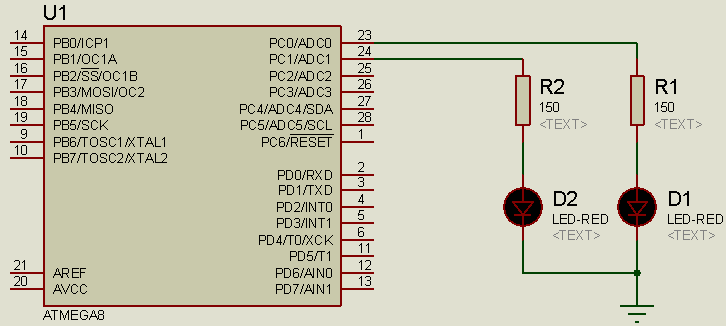


**6.** Добавьте в схему (на пин DC.1, что соответствует разряду 1 порта С) ещё один светодиод и  помигайте ими поочерёдно (один – зажигается, другой – гаснет, через секунду – наоборот).

Для этого нужно всего лишь изменить предпоследнюю команду на

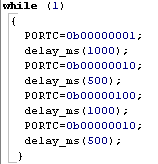
PORTC=0b00000010;

Соответственно отредактируйте схему в Протеусе и проведите в нём эмуляцию.

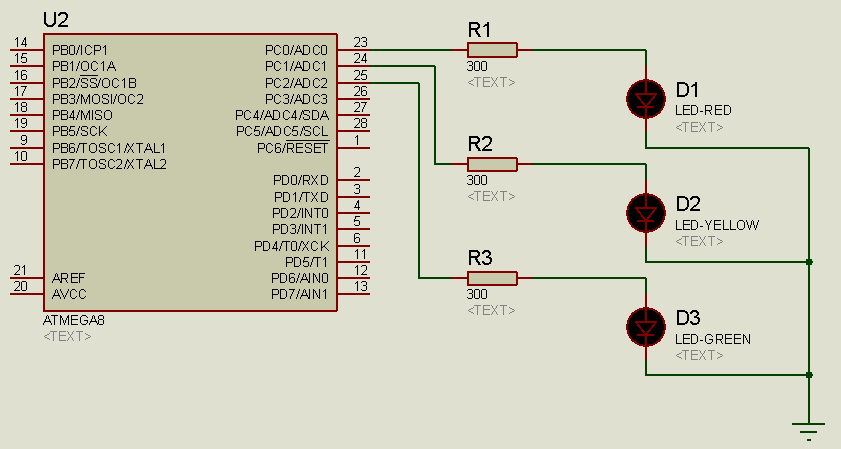


**7.** Добавьте в схему третий светодиод и создайте программу "**Светофор**" – мигания по схеме, близкой к реальному светофору.

В простейшем случае код цикла будет таким:

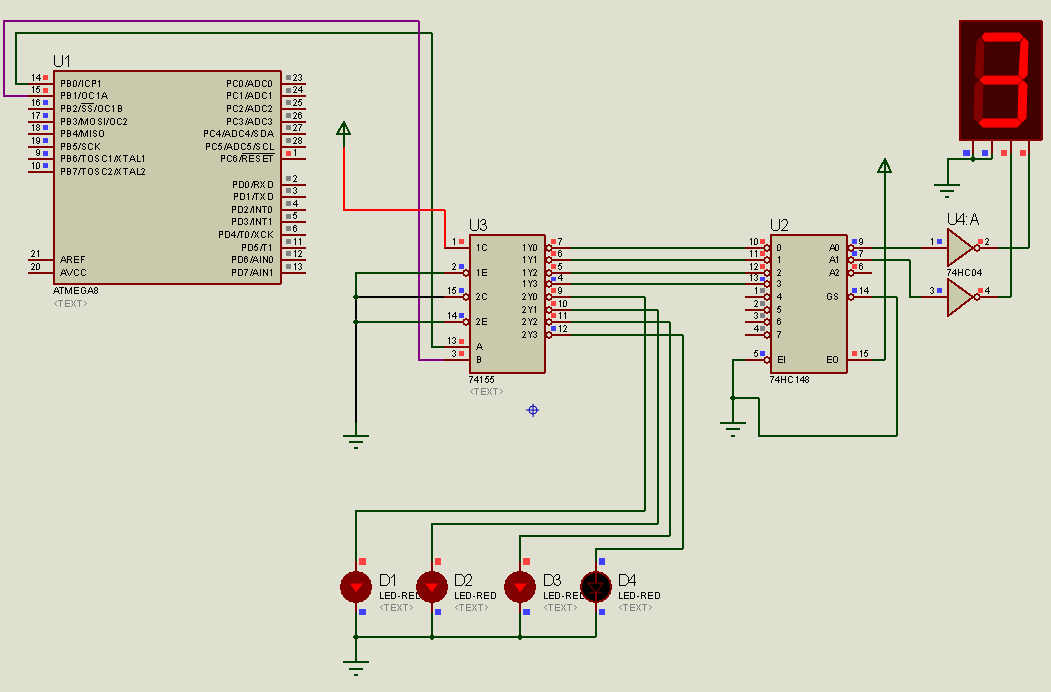


**Схема:**

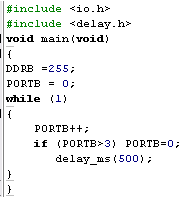


**Часть 2**

**8. Смоделируем дешифратор** 2→4 на ИМС 74155 (представляющей собой сдвоенные дешифраторы 2→4). Выходы одного дешифратора подадим на 4 светодиода, а второго – через шифратор 74148 – на 7-сегментный индикатор с встроенным дешифратором:

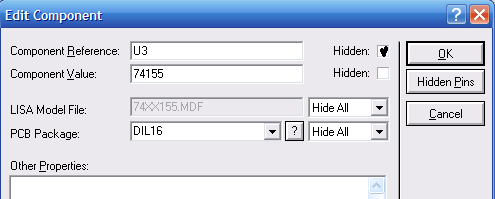


Программный код:



Светодиоды будут поочерёдно гаснуть, а на индикаторе высвечиваться цифры от 0 до 3.

Кстати, чтобы под компонентом не отображался навязчивый <Text>, введите пробел в Other свойствах компонента, а чтобы убрать идентификаторы ИМС, поставьте галочку в свойствах Hidden:

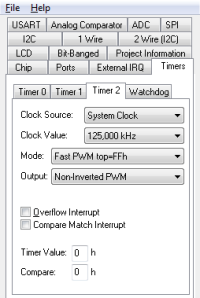
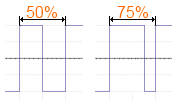


**9. Самостоятельное задание:**

1. Измените направление перемещения бегущего огня на противоположное; на двустороннее
2. Сделайте, чтобы индикатор показывал другие цифры;
3. В схеме задания 7 измените алгоритм мигания светофора на более приближённый к реальному.

**Часть 3**

ШИМ (PWM) — широтно-импульсная модуляция, способ регулирования напряжения. Пульсирующее напряжение, в зависимости от заполнения периода, эквивалентно постоянному напряжению. Т.е. если в течение периода напряжение 50% времени было включено, 50% выключено, то эквивалент постоянного напряжения будет равен 50% от номинального. В цифрах это так: было 5В постоянного напряжения прогнали через ШИМ получили 2,5В. Если заполнение импульса равно 75%, то эквивалентное постоянное напряжение будет 3,75В.

[](http://avr-start.ru/wp-content/uploads/2012/12/set1.png)[](http://avr-start.ru/wp-content/uploads/2012/12/pwm_50.png)

Теперь приступим к практической реализации. Будем при помощи микроконтроллера изменять заполнение от 0 до 100%, потом от 100% до нуля.

Запускаем CodeVision. Создаем проект при помощи мастера. В разделе таймеров (Timers), выбираем Timer 2 и выставляем настройки как на рисунке.

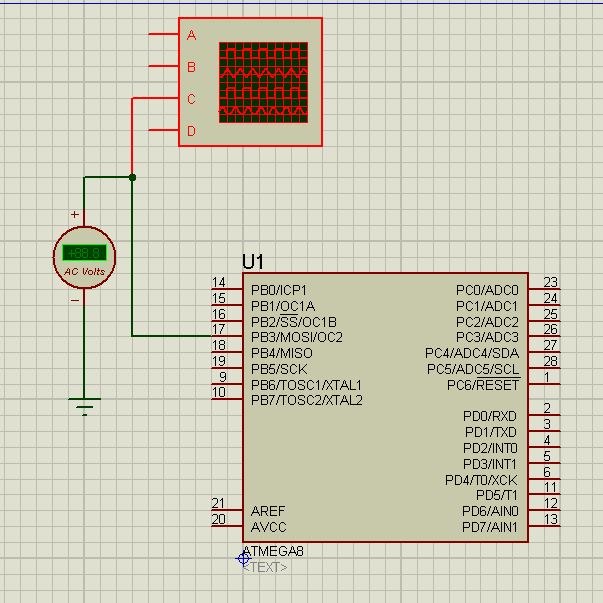
Переменная OCR2 отвечает за величину заполнения импульса. Изменяется данная величина от 0 до 255(0хFF), т.е. 255 соответствует 100% -му заполнению (постоянный ток). Следовательно, если нужно 30% заполнение (255/100)\*30=77. Далее 77 переводим в шестнадцатеричную систему OCR2=0x4D;

Изменяя величину TCCR2, мы можем регулировать частоту ШИМ. Величина частоты работы ШИМ кратна частоте, на которой работает микроконтроллер. В проекте использована частота микроконтроллера 8 МГц, частоту ШИМ использовали 125 кГц, следовательно делитель равен 8/125=64 0x6C в двоичной системе счисления 1101100. Последние цифры 100 регистра TCCR2 отвечают за выбор частоты работы ШИМ

Код прост: сначала в цикле увеличиваем заполнение от 0 до 255(ff), потом уменьшаем от 255 до 0:

Схема в Proteus Isis

#include <mega8.h>

#include <delay.h>

void main(void)

{

PORTB=0x00;

DDRB=0x08;

ASSR=0x00;

TCCR2=0x6C;

TCNT2=0x00;

OCR2=0x00;

TIMSK=0x00;

while (1)

{

while(OCR2<0xff)

{

OCR2=OCR2+0x01;

delay\_ms(5);

}

while(OCR2>0x00)

{

OCR2=OCR2-0x01;

delay\_ms(5);

}

};

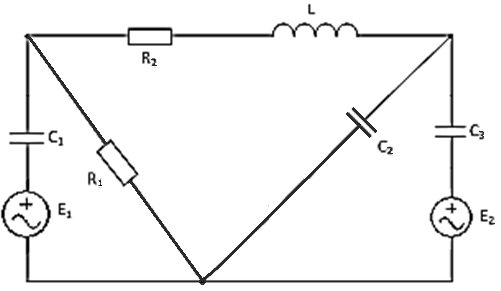
}

**ЗАДАЧА 5**

Составить систему уравнений относительно комплексов тока в схеме по законам Кирхгофа, если

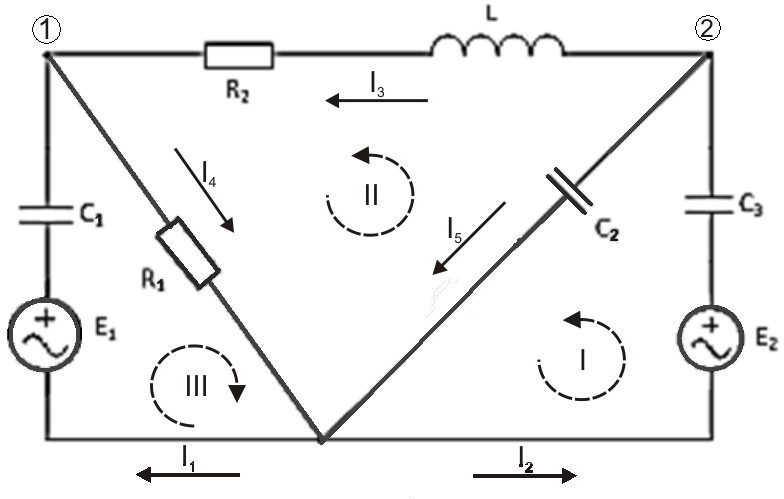
R1 = R2 = 100 Ом; С1 = С2 = С3 = 1 мФ; L = 100 мГн;

параметры идеальных источников ЭДС одинаковы: E1 = E2 = 220 В, ν = 15,9155, ϕ0 = 0.



**РЕШЕНИЕ**

Составим уравнения по законам Кирхгофа, предварительно задавшись положительными направлениями токов и обходов контуров.



Поскольку в цепи имеются 2 независимых узла и 3 контура, запишем два уравнения по первому и три – по второму закону Кирхгофа. Система уравнений примет вид (Xc берётся с противоположным знаком):



Матрица в символическом виде:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | I1 | I2 | I3 | I4 | I5 | E |
| узел 1 | -1 | 0 | -1 | 1 | 0 | 0 |
| узел 2 | 0 | -1 | -1 | 0 | 1 | 0 |
| контур I | 0 | -jXc | 0 | 0 | -jXc | E |
| контур II | 0 | 0 | R+jXL | R | jXc | 0 |
| контур III | -jXc | 0 | 0 | R | 0 | E |

Находим циклическую частоту: ω = 2 = 100 рад/с

Вычисляем реактивные сопротивления: Xc = 1/ω⋅C = 10 Ом, XL = ω⋅L = 10 Ом.

Далее подставляем числовые значения в матрицу.

Хотя Microsoft Excel и располагает функциями для работы с комплексными числами, но для решения системы уравнений с комплексными числами использование Excel без специальных надстроек или пользовательских функций – весьма сложная задача. На последней лабораторной работе мы ознакомились с технологией работы с матрицами в программе Mathcad, использование комплексных чисел в которой не вызывает никаких затруднений. Далее приведён листинг решения:



При вводе мнимой единицы не нужно вводить знак \*. При вводе числа i без коэффициента необходимо вводить 1i.















Оглавление

[**ЗАДАЧА 1** 1](#_Toc477364762)

[**ЗАДАЧА2** 8](#_Toc477364766)

[**ЗАДАЧА 3** 12](#_Toc477364767)

[**ЗАДАЧА4** 15](#_Toc477364768)

[**ЗАДАЧА 5** 22](#_Toc477364769)