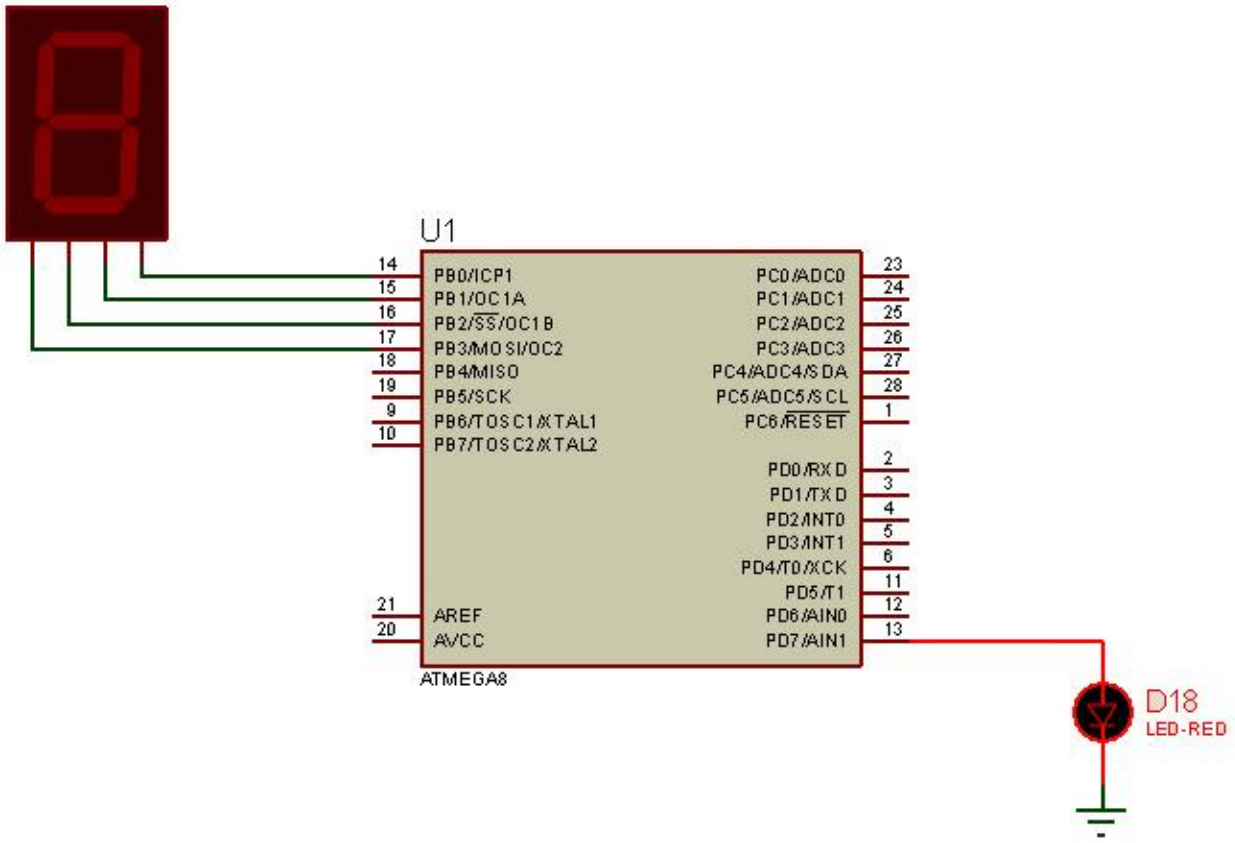
Вариант 1

Индикатор от 0 до F, LED мигает.



Задача состоит из двух подзадач.

Для первой – такой код:

```
#include <io.h>
#include <delay.h>
void main(void)
{
DDRB =255;
PORTB = 0;

while (1)
  {
    PORTB++;
    if (PORTB == 15) PORTB=0;
    delay_ms(500);
   }
}
```

Для второй – такой код:

```
#include <io.h>
#include <delay.h>
void main(void)
{
DDRD = 0b10000000;
PORTD = 0b10000000; // 128
while (1)
    {
```

```
   if (PORTD == 128) PORTD = 0;
      else PORTD = 128;
   delay_ms(500);
   }
}
```

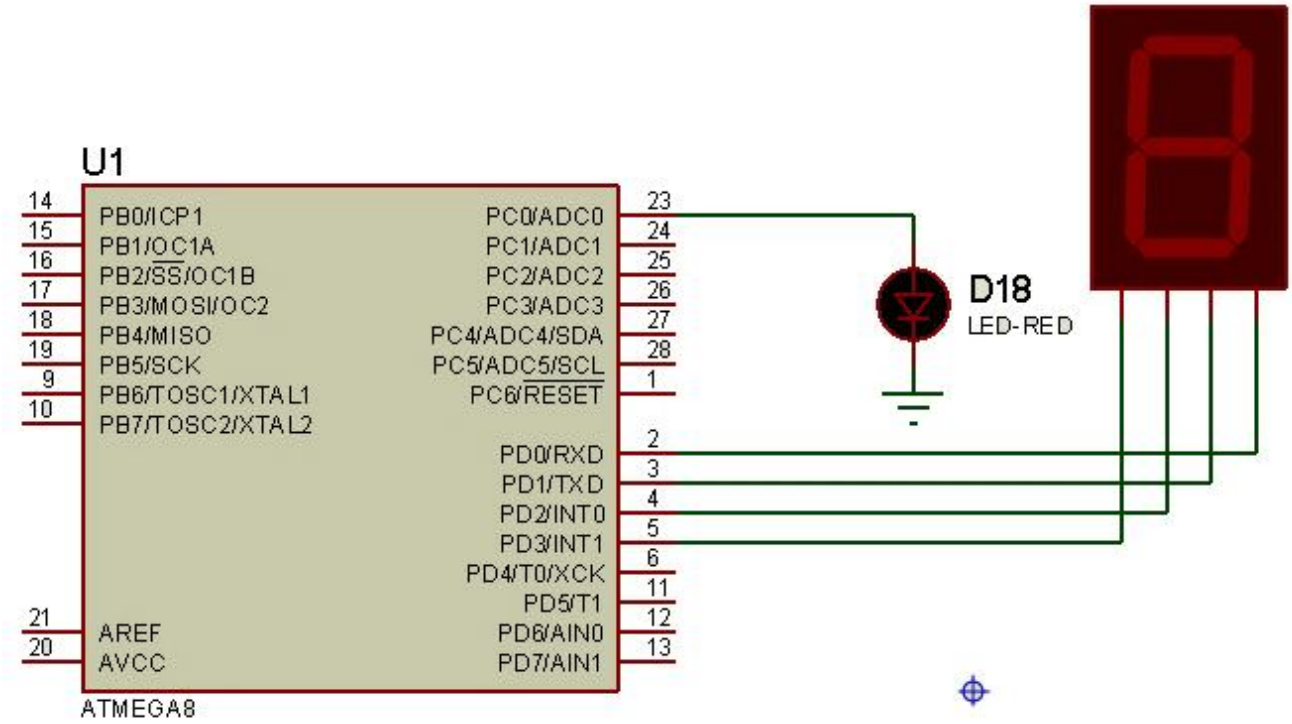
Для одновременного мигания и индицирования цифр – такой код:

```
#include <io.h>
#include <delay.h>
void main(void)
{
DDRB =255;
PORTB = 0;
DDRD = 0b10000000;
PORTD = 0b10000000; // = 128
while (1)
  {
   if (PORTD == 128) PORTD = 0;
      else PORTD = 128;
   PORTB++;
   if (PORTB == 15) PORTB=0;
     delay_ms(500);
  }
}
```

Вариант 2

Индикатор от 0 до F, LED мигает.

(то же самое, но изменены выводы)

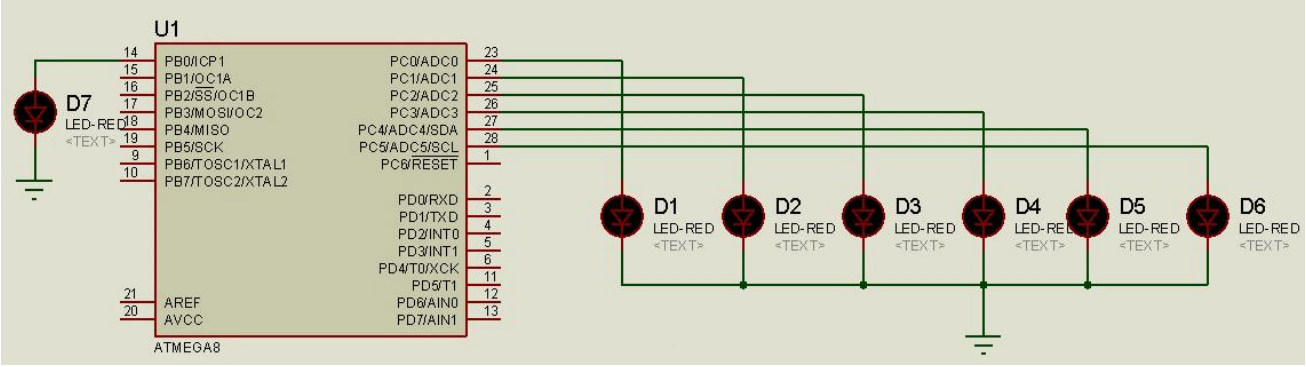


Программный код:

```
#include <io.h>
#include <delay.h>
void main(void)
{
DDRD =255;
PORTD = 0;
DDRC = 0b00000001;
PORTC = 0b00000001; // 1
while (1)
   {
   if (PORTC == 1) PORTC = 0;
      else PORTC = 1;
   PORTD++;
   if (PORTD == 15) PORTD=0;
     delay_ms(500);
   }
}
```

Вариант 3

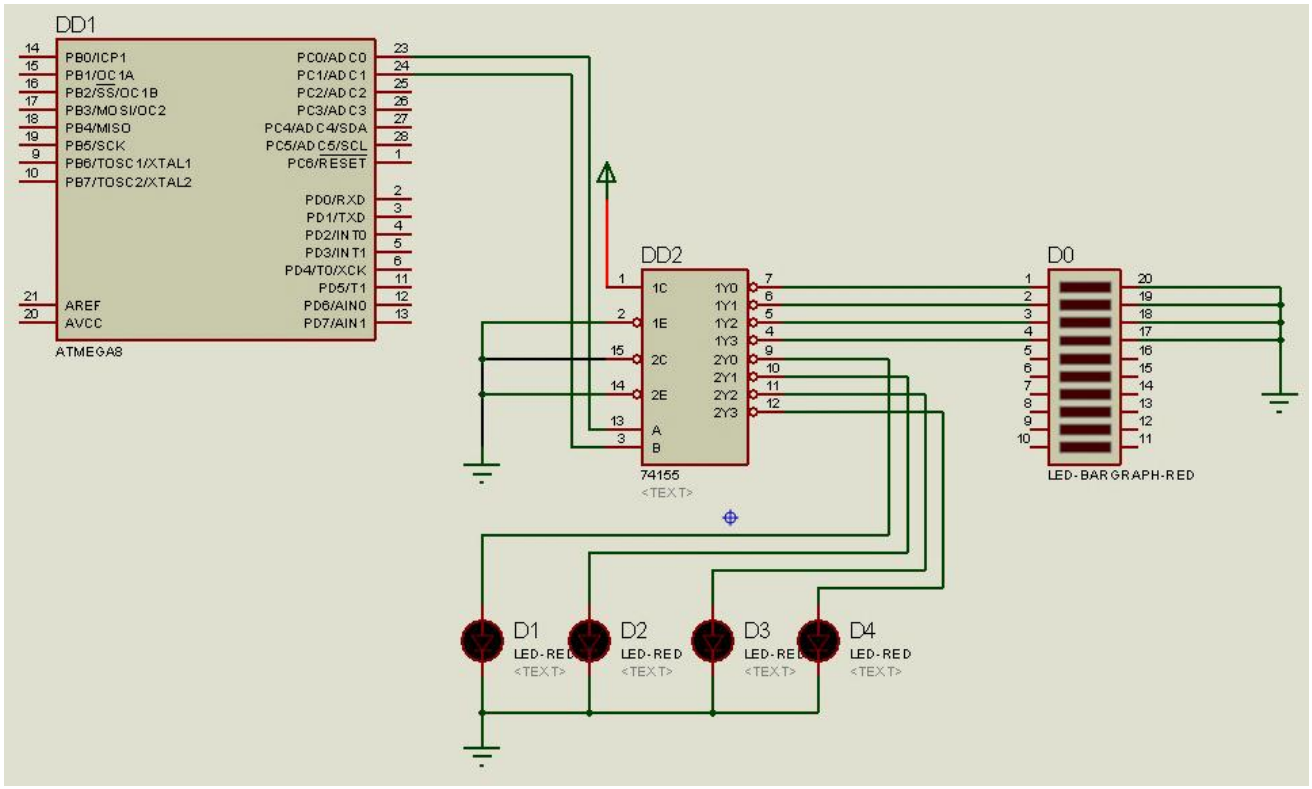
Огонь бежит слева направо (от D1 к D6), D7 мигает.




```
#include <io.h>
#include <delay.h>
void main(void)
{
DDRC =0b111111;
PORTC = 1;
DDRB =1;
PORTB = 1;
while (1)
   {
    PORTB.0 = ~PORTB.0;  // инверсия состояния только одного вывода порта
   if (PORTC == 0b1000000) PORTC=1;
    delay_ms(500);
   PORTC = PORTC << 1;  // сдвиг влево
   }
}
```

Вариант4

Сдвоенный дешифратор даёт бегущий огонь на светодиоды и на барграф (синхронно).
(схема чуть сложнее, но код – проще)



Слева направо / сверху вниз:

```
#include <io.h>
#include <delay.h>
void main(void)
{
DDRC =0b0000011;
PORTC = 0;
while (1)
    {
    if (PORTC == 4) PORTC=0;
     delay_ms(500);
    PORTC ++;
    }
}
```

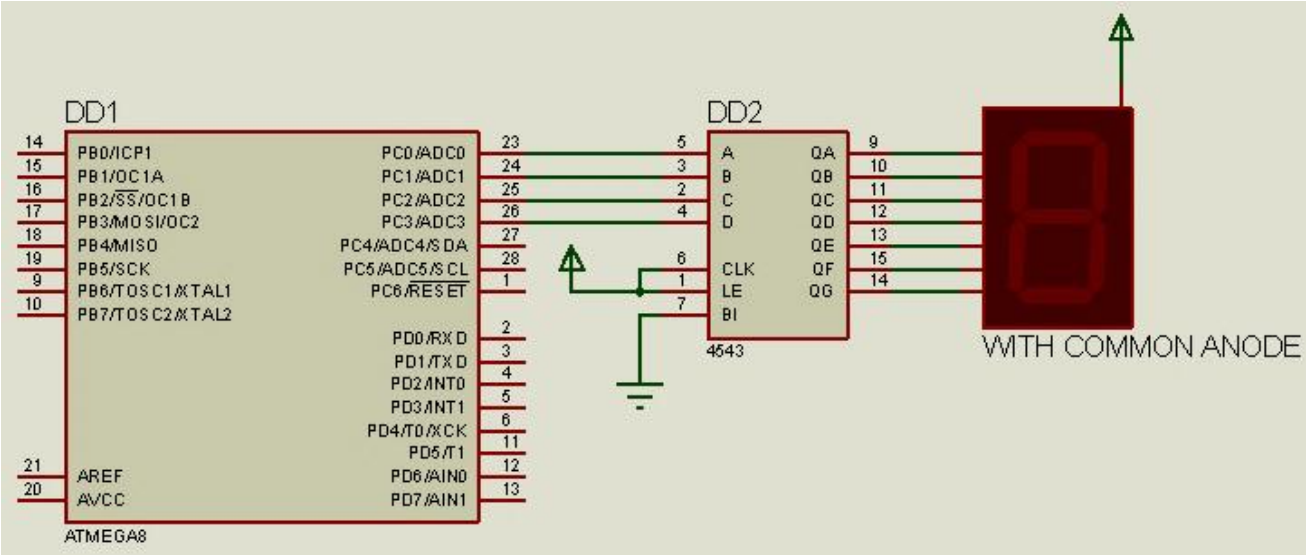
Справа налево / снизу вверх:

```
#include <io.h>
#include <delay.h>
void main(void)
{
DDRC = 0b0000011;
PORTC = 0b0000011; // = 3
while (1)
   {
```

```
    PORTC--; // декремент, т.е. уменьшение значения на 1
    delay_ms(500);
    if (PORTC == 0) PORTC = 4;
    }
}
```

Вариант 5

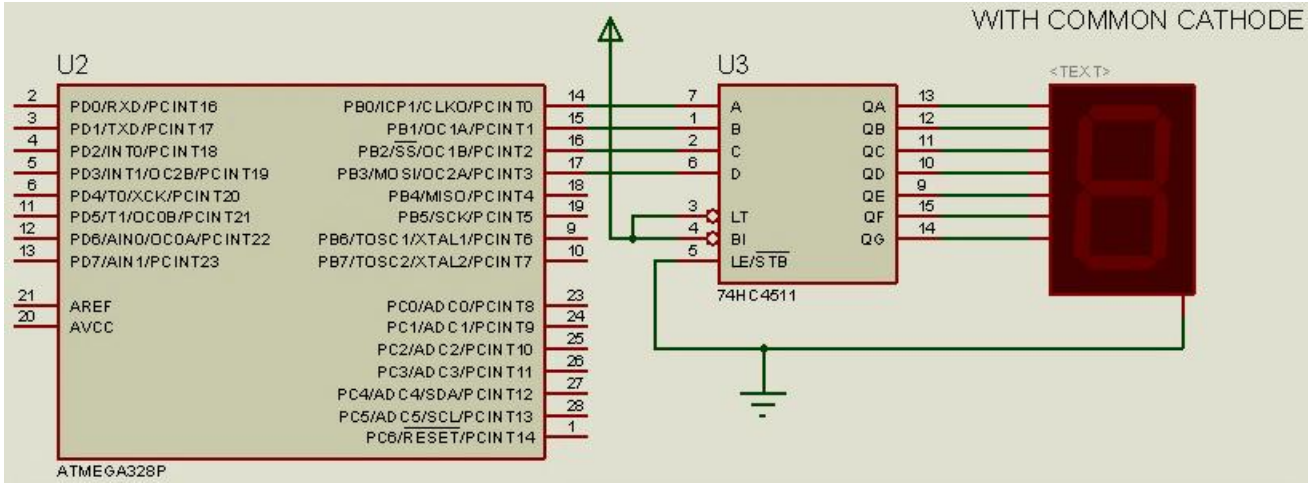
Вывод на индикатор цифр от 0 до 9 с «внешним» семисегментным дешифратором.



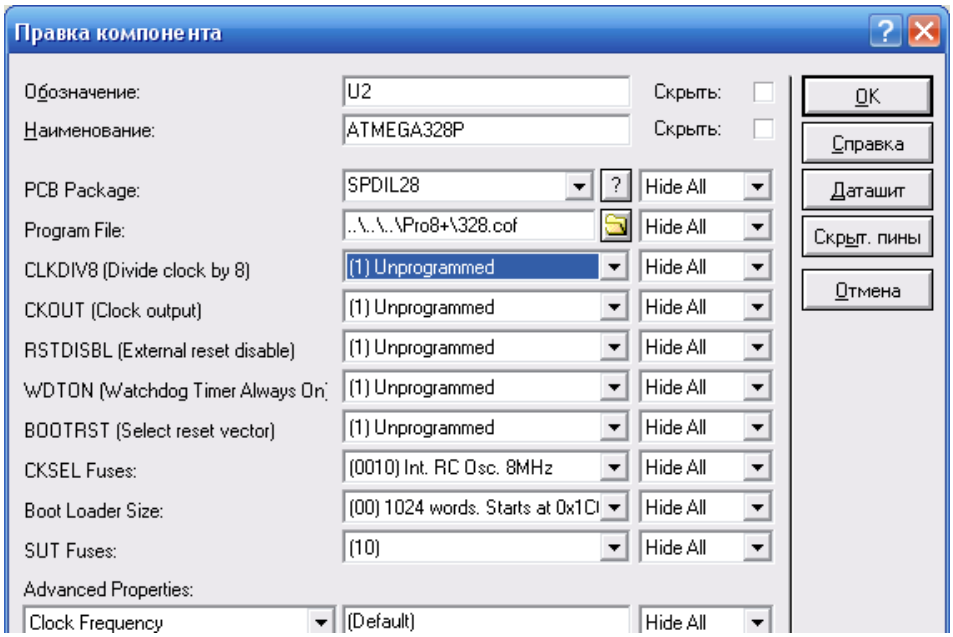

```
#include <io.h>
#include <delay.h>
void main(void)
{
DDRC = 0b0001111;
PORTC = 0;
while (1)
    {
    if (PORTC == 10) PORTC = 0;
    delay_ms(500);
    PORTC++; // инкремент, т.е. увеличение значения на 1
    }
}
```

Вариант 6

Вывод на индикатор цифр от 0 до 9 с «внешним» семисегментным дешифратором.
МК ATMega328P



При создании проекта не забудьте правильно выбрать микроконтроллер.
В настройках МК нужно изменить бит CLKDIV8 на 1:



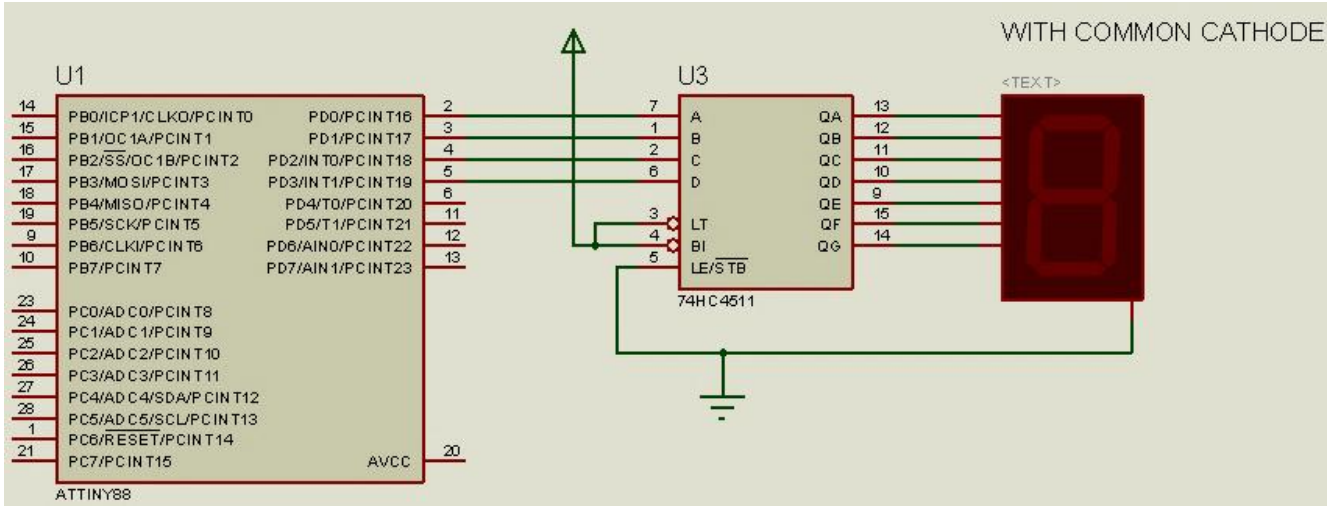
Программный код:

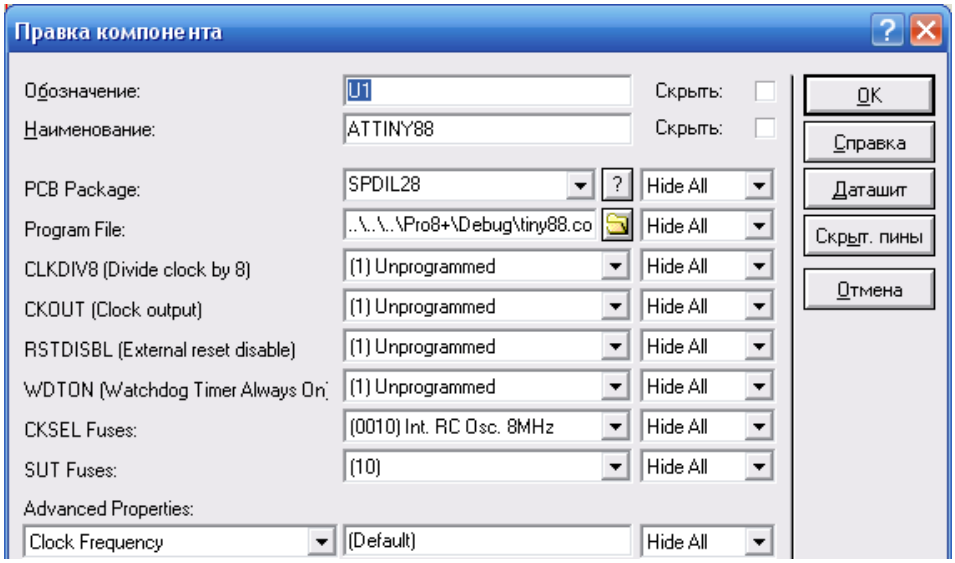
```
#include <io.h>
#include <delay.h>
void main(void)
{
DDRB = 0b0001111;
PORTB = 0;
while (1)
   {
   if (PORTB == 10) PORTB = 0;
   delay_ms(500);
   PORTB++; // инкремент, т.е. увеличение значения на 1
   }
}
```

Вариант 7

Вывод на индикатор цифр от 0 до 9 с «внешним» семисегментным дешифратором.

МК ATTINY88



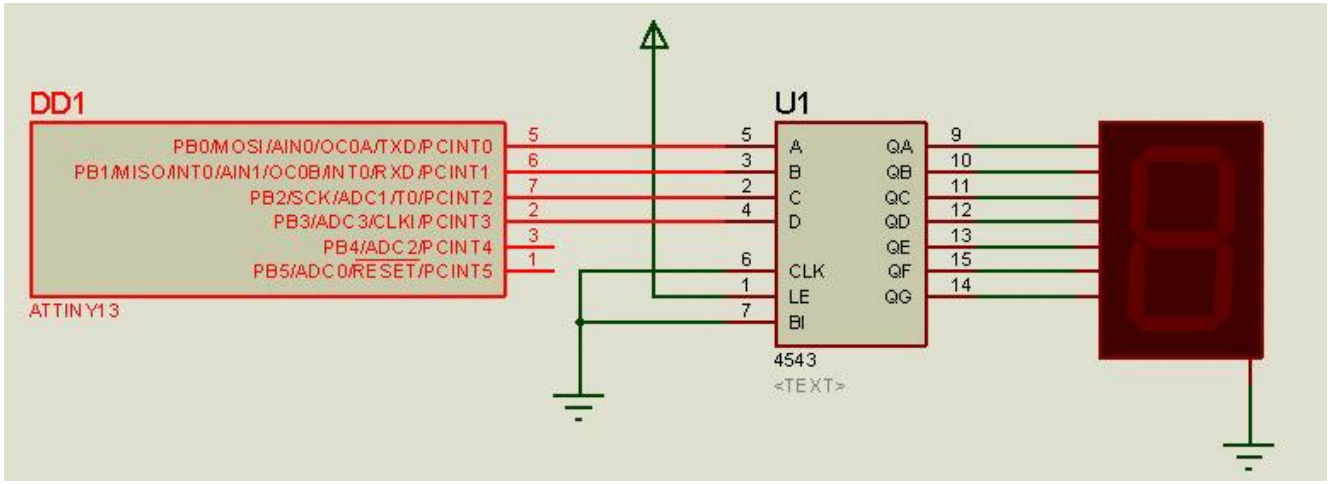


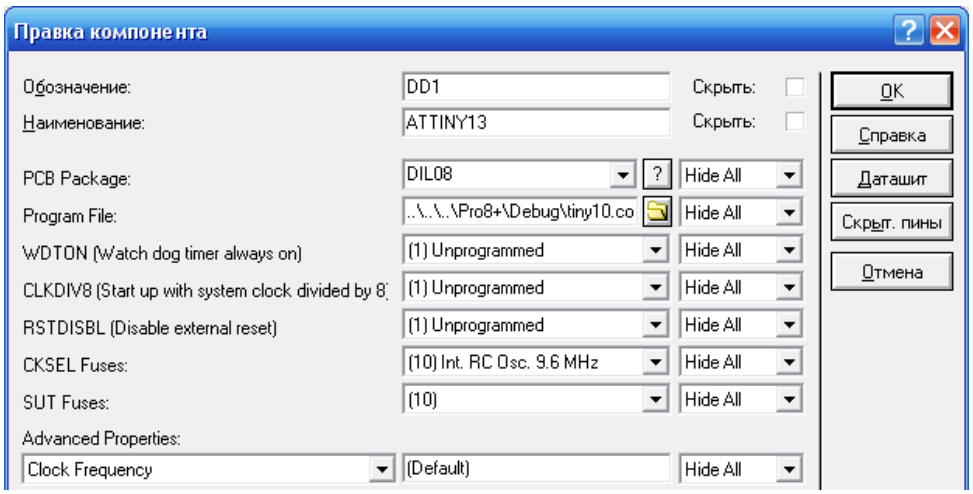
Программный код:

```
#include <io.h>
#include <delay.h>
void main(void)
{
DDRD = 0b0001111;
PORTD = 0;
while (1)
    {
    if (PORTD == 10) PORTD = 0;
    delay_ms(500);
    PORTD++; // инкремент, т.е. увеличение значения на 1
    }
}
```

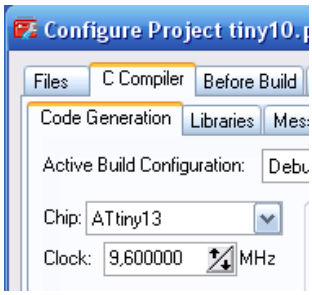
Вариант 8

Вывод на индикатор цифр от 0 до 9 с «внешним» семисегментным дешифратором.





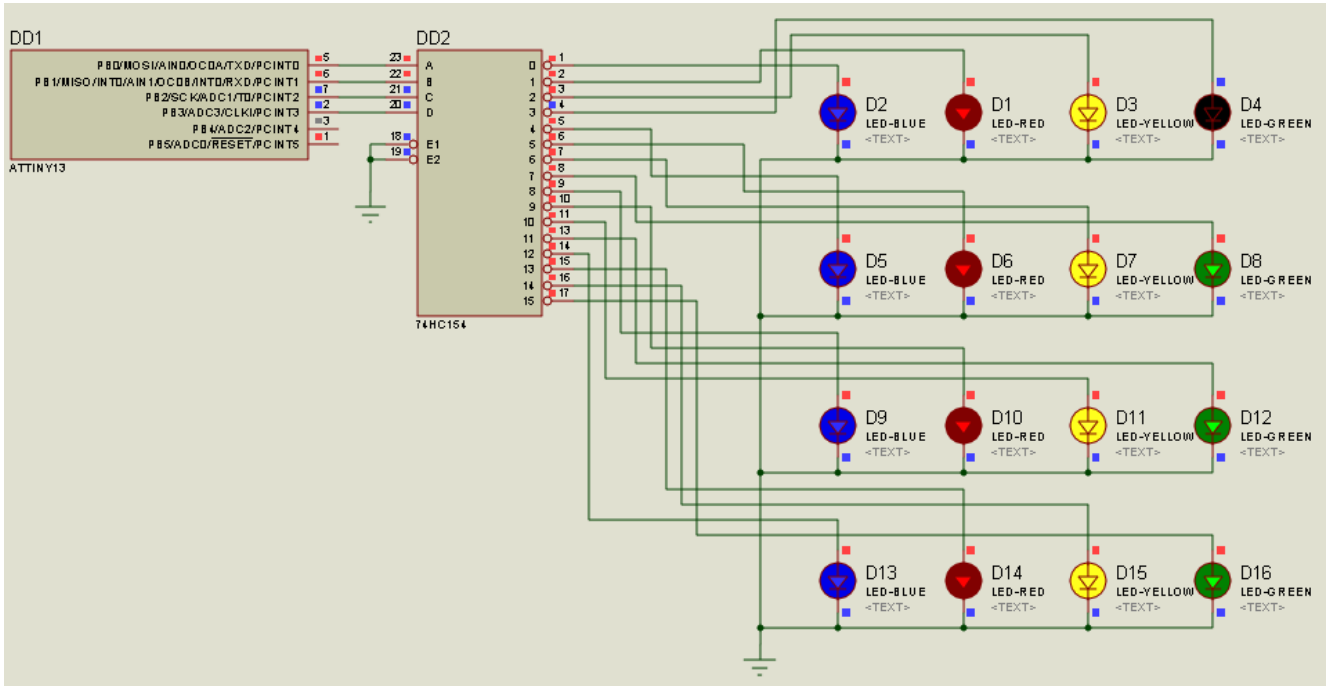
В конфигурации чипа в CodeVisionAVR поставьте Clock = 9,6 MHz:




```
#include <io.h>
#include <delay.h>
void main(void)
{
DDRB = 0b0001111;
PORTB = 0;
while (1)
    {
    if (PORTB == 10) PORTB = 0;
    delay_ms(500);
    PORTB++; // инкремент, т.е. увеличение значения на 1
    }
}
```

Вариант 9

МК ATTiny13 с помощью дешифратора 74HC154 последовательно гасит один из 16 светодиодов



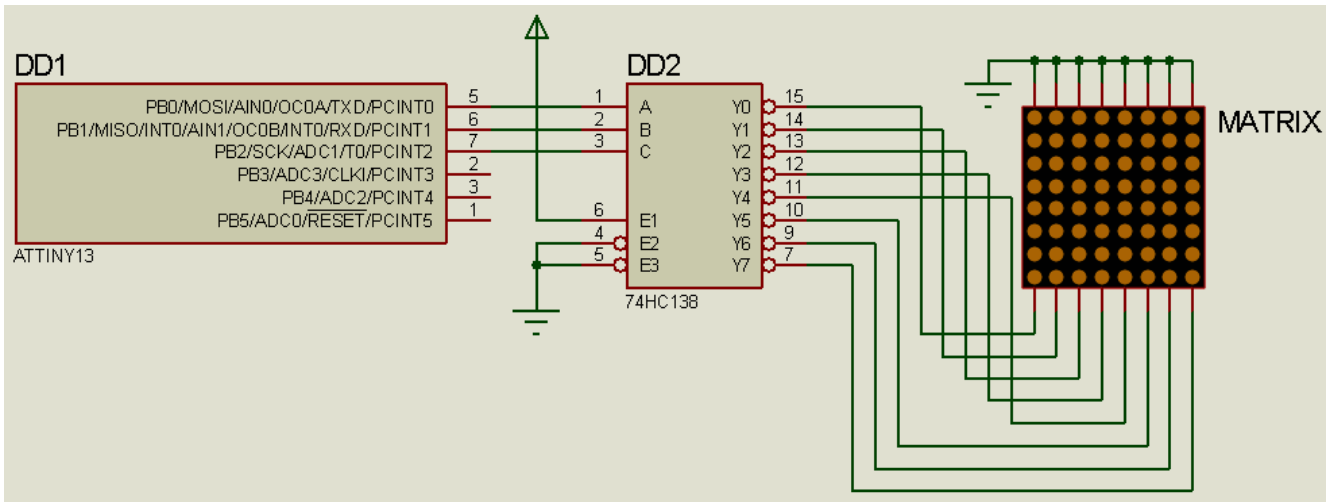
Clock = 9.6 MHz (см. Вариант 8)

(Расположить светодиоды можете по-другому, не обязательно в матрицу и с такими цветами)

```
#include <io.h> // подключение библиотек с функциями
#include <delay.h>
void main(void)
{
DDRB = 0b0001111; // направление
PORTB = 0; // начальное значение
while (1)
   {
   if (PORTB == 16) PORTB = 0; // возврат к началmyjve состоянию
   delay_ms(100);  // задержка
   PORTB++; // инкремент, т.е. увеличение значения на 1
   }
}
```

Вариант 10

ATTiny13 гасит матрицу по столбцам (слева направо) (с помощью дешифратора 74HC138)

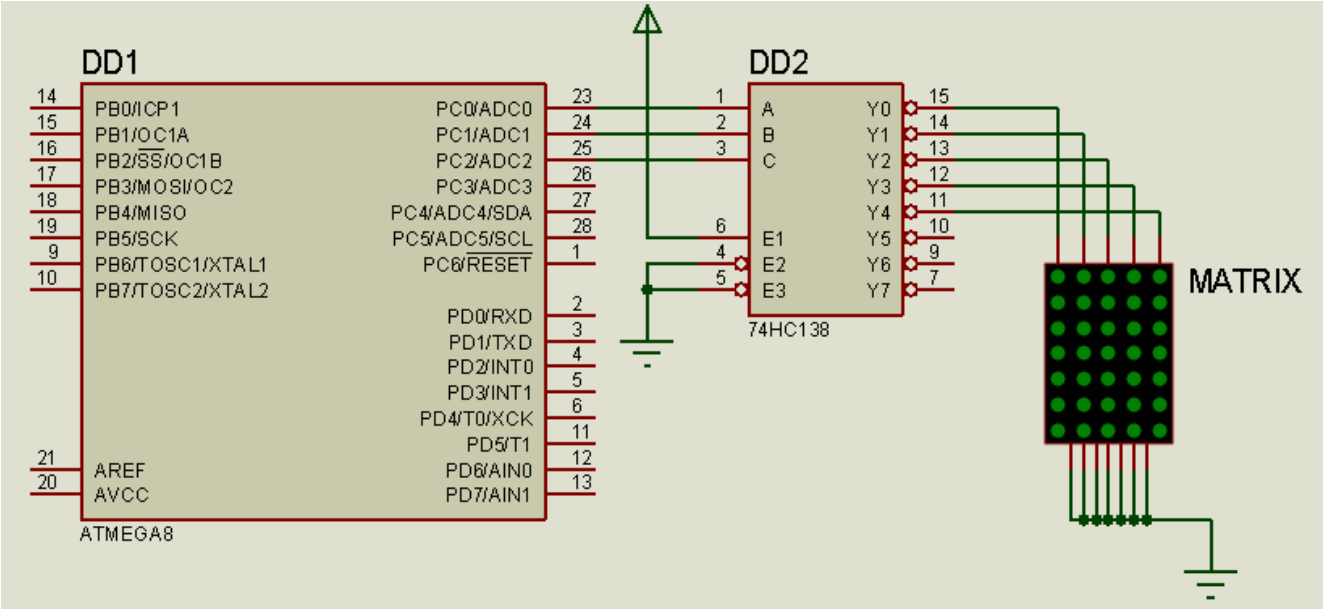


Clock = 9.6 MHz (см. Вариант 8)

```
#include <io.h> // подключение библиотек с функциями
#include <delay.h>
void main(void)
{
DDRB = 0b0001111; // направление
PORTB = 0; // начальное значение
while (1)
   {
   if (PORTB == 8) PORTB = 0; // обнуление по достижению максимума
   delay_ms(200);  // задержка
   PORTB++; // инкремент, т.е. увеличение значения на 1
   }
}
```

Вариант 11

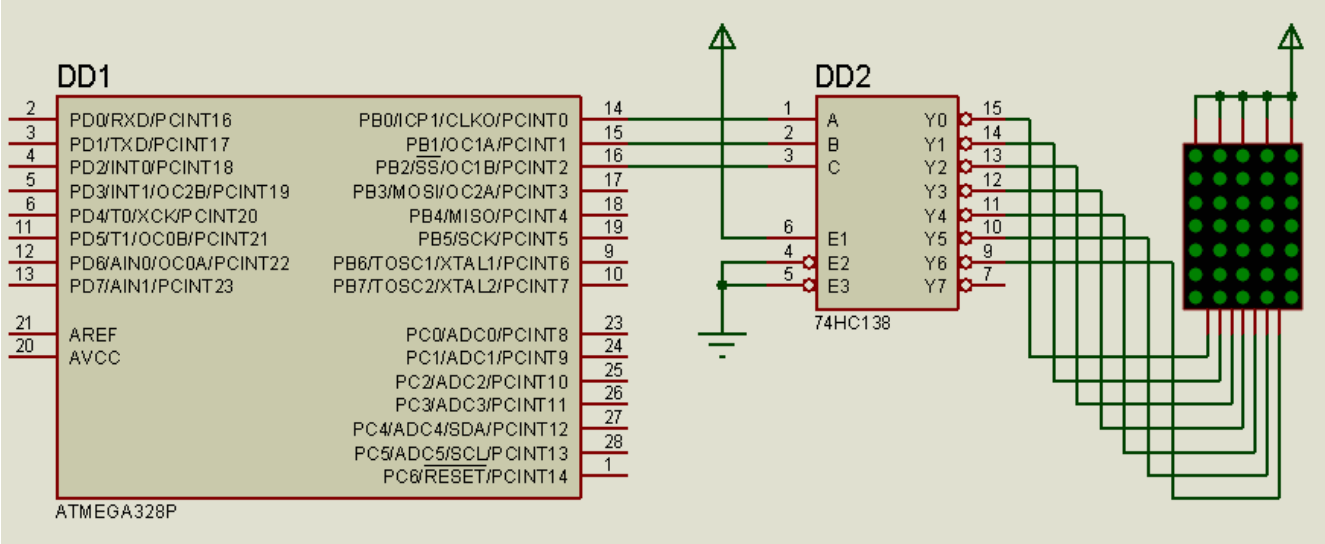
ATMega8 гасит матрицу по столбцам (слева направо) (с помощью дешифратора 74HC138)




```
#include <io.h>
#include <delay.h>
void main(void)
{
DDRC = 0b0001111;
PORTC = 0;
while (1)
   {
   if (PORTC == 5) PORTC = 0;
   delay_ms(500);
   PORTC++; // инкремент, т.е. увеличение значения на 1
   }
}
```

Вариант 12

ATMega8 гасит матрицу по строкам (сверху вниз) (с помощью дешифратора 74HC138)

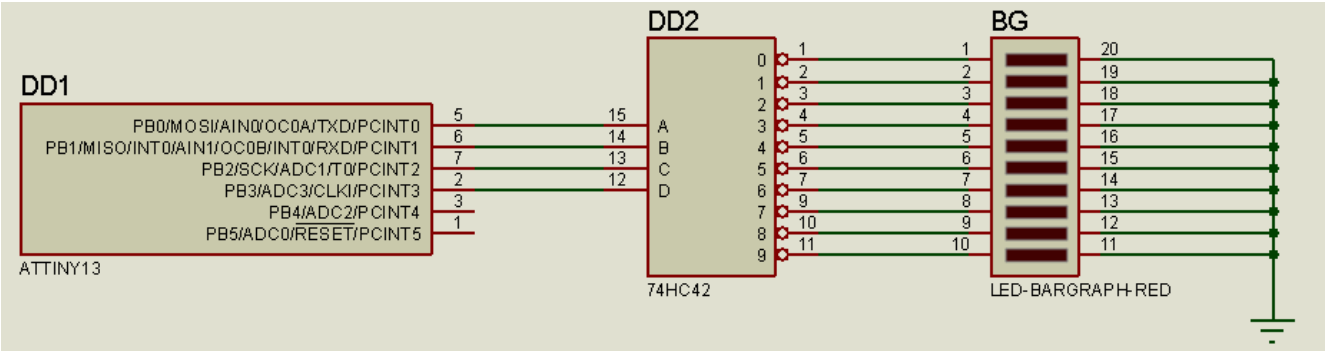


Настройка здесь

```
#include <io.h>
#include <delay.h>
void main(void)
{
DDRB = 0b0001111;
PORTB = 0;
while (1)
    {
    if (PORTB == 7) PORTB = 0;
    delay_ms(500);
    PORTB++; // инкремент, т.е. увеличение значения на 1
    }
}
```

Вариант 13

ATTiny13 с помощью дешифратора 74HC42 гасит сегменты барграфа снизу вверх.



Clock = 9.6 MHz (см. Вариант 8)

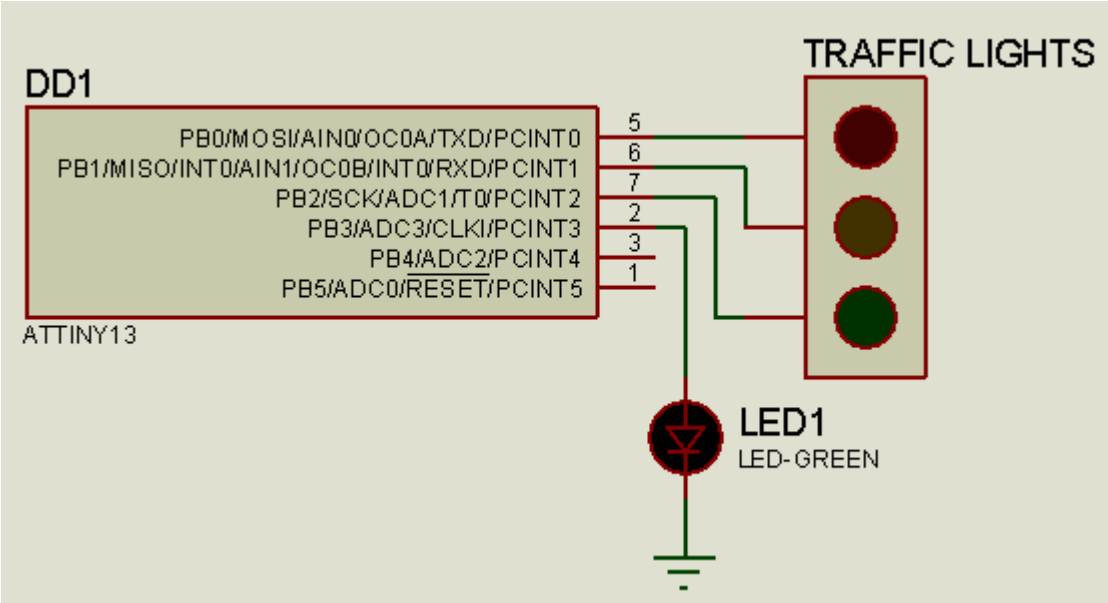
```
#include <io.h> // подключение библиотек с функциями
#include <delay.h>
```

```
void main(void)
{
DDRB = 0b001111; // направление
PORTB = 10; // начальное значение
while (1)
    {
    PORTB--; // декремент, т.е. уменьшение значения на 1
    delay_ms(200);  // задержка
    if (PORTB == 0) PORTB = 10;
    }
}
```

Вариант 14

Светофор.

После загорания жёлтого красный какое-то время ещё горит; жёлтый зажигается чуть раньше затухания зелёного.



Код программы (вариант посложнее)

```
#include <io.h> // подключение библиотек с функциями
#include <delay.h>
void main(void)
{
int x=5; // переменная - коэффициент длительности
int i;
DDRB = 0b001111; // направление
PORTB = 0; // начальное значение
while (1)
   {
   PORTB.0=1; // красный
   delay_ms(500*x);  // задержка
   PORTB.1=1; // жёлтый
   delay_ms(200*x);
   PORTB.0=0;
   delay_ms(300*x);
```

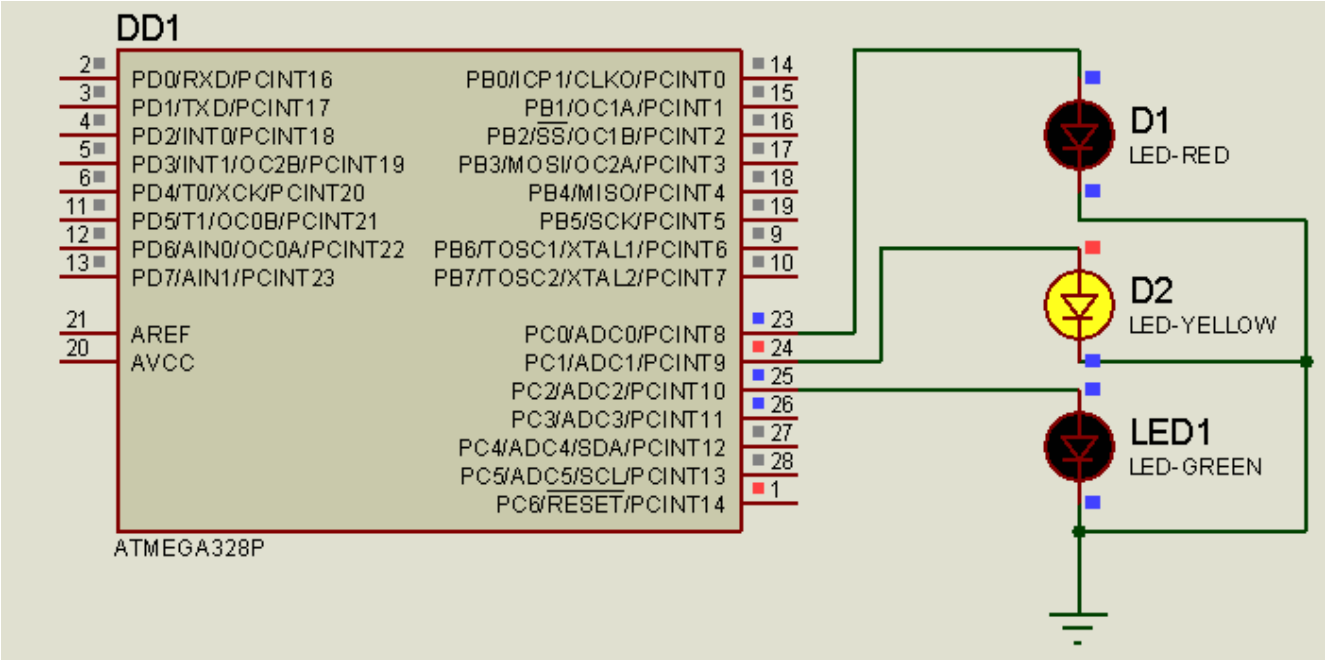
```
   PORTB.1=0;
   PORTB.2=1;  // зелёный
   for (i=0; i<5; i++)
   {       // вместо дополнительного зелёного тут может стучать некий зуммер
      PORTB.3=1;
      delay_ms(50*x);
      PORTB.3=0;
      delay_ms(50*x);
   }
   PORTB.1=1;
   delay_ms(200*x);
   PORTB.2=0;
   delay_ms(200*x);
   PORTB.1=0;
   }
}
```

Код программы (вариант попроще)

```
#include <io.h> // подключение библиотек с функциями
#include <delay.h>
void main(void)
{
DDRB = 0b001111; // направление
PORTB = 0; // начальное значение
while (1)
   {
   PORTB.0=1; // красный
   delay_ms(500);  // задержка
   PORTB.1=1; // жёлтый
   delay_ms(200);
   PORTB.0=0;
   delay_ms(300);
   PORTB.1=0;
   PORTB.2=1;  // зелёный
   PORTB.3=1;  // дополнит.
   delay_ms(300);
   PORTB.1=1;
   delay_ms(200);
   PORTB.2=0;
   PORTB.3=0;
   delay_ms(200);
   PORTB.1=0;
   }
}
```

Вариант 15

Соберите схему «Светофор». После загорания жёлтого красный какое-то время ещё горит; жёлтый зажигается чуть раньше затухания зелёного.

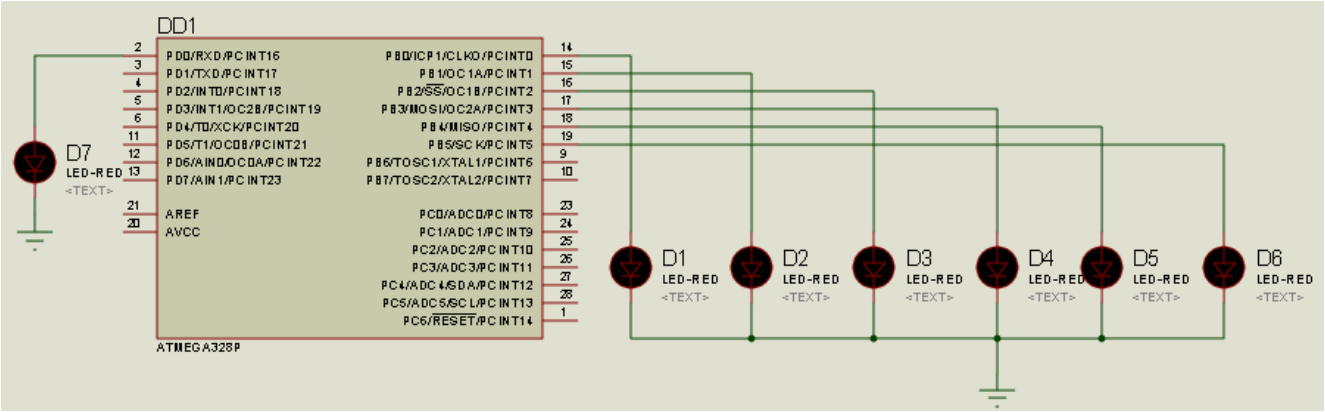


Настройка здесь

```
#include <io.h> // подключение библиотек с функциями
#include <delay.h>
void main(void)
{
DDRC = 0b000111; // направление
PORTC = 0; // начальное значение
while (1)
    {
    PORTC.0=1; // красный
    delay_ms(1000);  // задержка
    PORTC.1=1; // жёлтый
    delay_ms(300);
    PORTC.0=0;
    delay_ms(500);
    PORTC.1=0;
    PORTC.2=1;  // зелёный
    delay_ms(1000);
    PORTC.1=1;
    delay_ms(200);
    PORTC.2=0;
    delay_ms(500);
    PORTC.1=0;
    }
}
```

Вариант 16

Огонь бежит справа налево (от D6 к D1), D7 мигает.

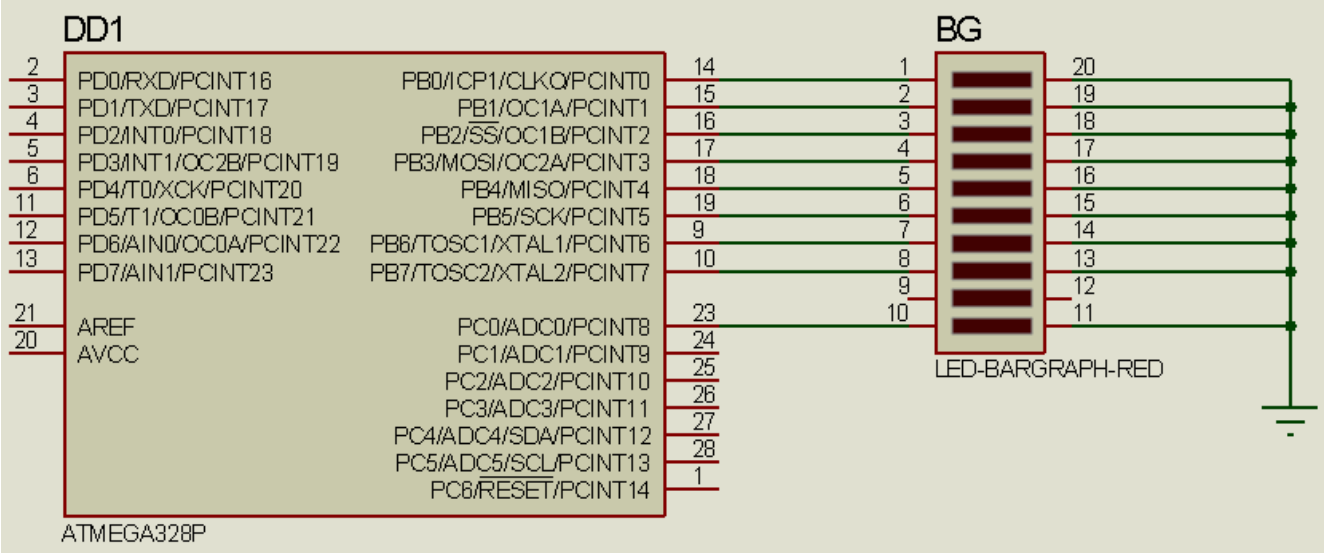


Настройка здесь

```
#include <io.h>
#include <delay.h>
void main(void)
{
DDRB =0b111111;
PORTB = 0b100000;
DDRD =1;
PORTD = 1;
while (1)
   {
    PORTD.0 = ~PORTD.0;  // инверсия состояния только одного вывода порта
   if (PORTB == 0) PORTB = 0b100000; // возврат к D6
    delay_ms(500);
   PORTB = PORTB >> 1; // сдвиг вправо
   }
}
```

Вариант 17

Огонь бежит снизу вверх (от 8-го сегмента барграфа к 1-му), 10-й мигает.

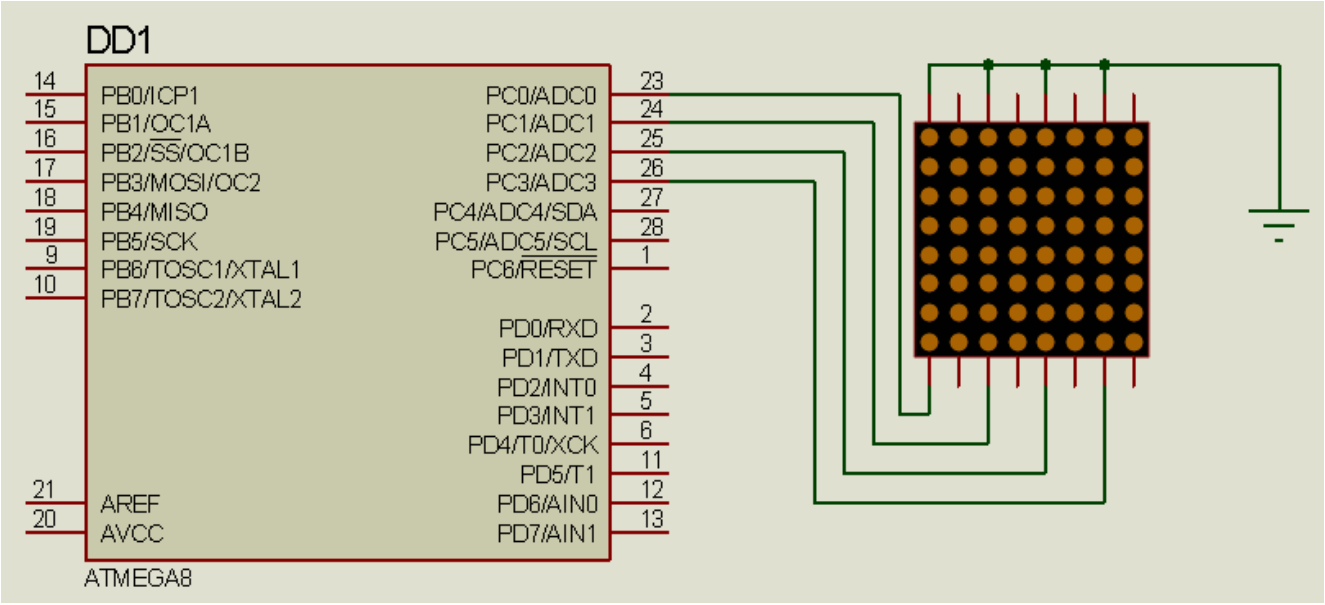


Настройка здесь

```
#include <io.h>
#include <delay.h>
void main(void)
{
DDRB =0b11111111;
PORTB = 0;
DDRC =1;
PORTC = 1;
while (1)
   {
    PORTC.0 = ~PORTC.0;  // инверсия состояния только одного вывода порта
   if (PORTB == 0) PORTB = 0b10000000; // возврат к D6
    delay_ms(500);
   PORTB = PORTB >> 1; // сдвиг вправо
   }
}
```

Вариант 18

Огонь бежит слева направо через столбец.

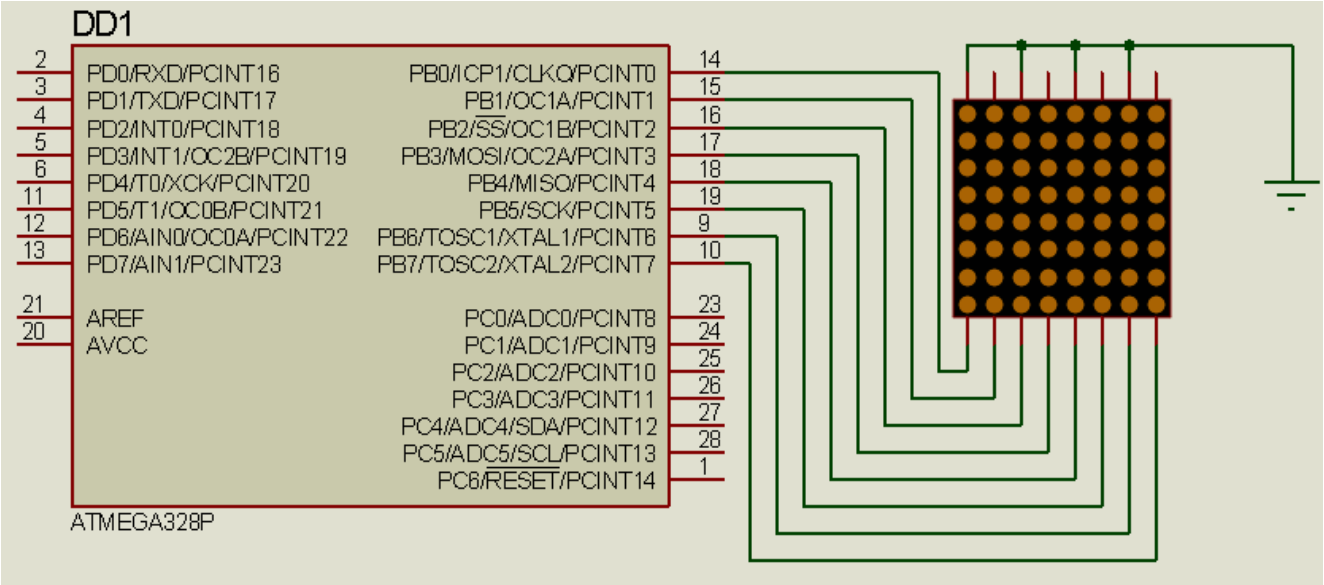



```
#include <io.h>
#include <delay.h>
void main(void)
{
DDRC =255;
PORTC =1;
while (1)
   {
    delay_ms(100);

    PORTC = PORTC << 1;  // сдвиг влево
    if (PORTC == 0b0010000) PORTC = 1;  //
   }
}
```

Вариант 19

Огонь бежит слева направо через столбец.



Настройка здесь

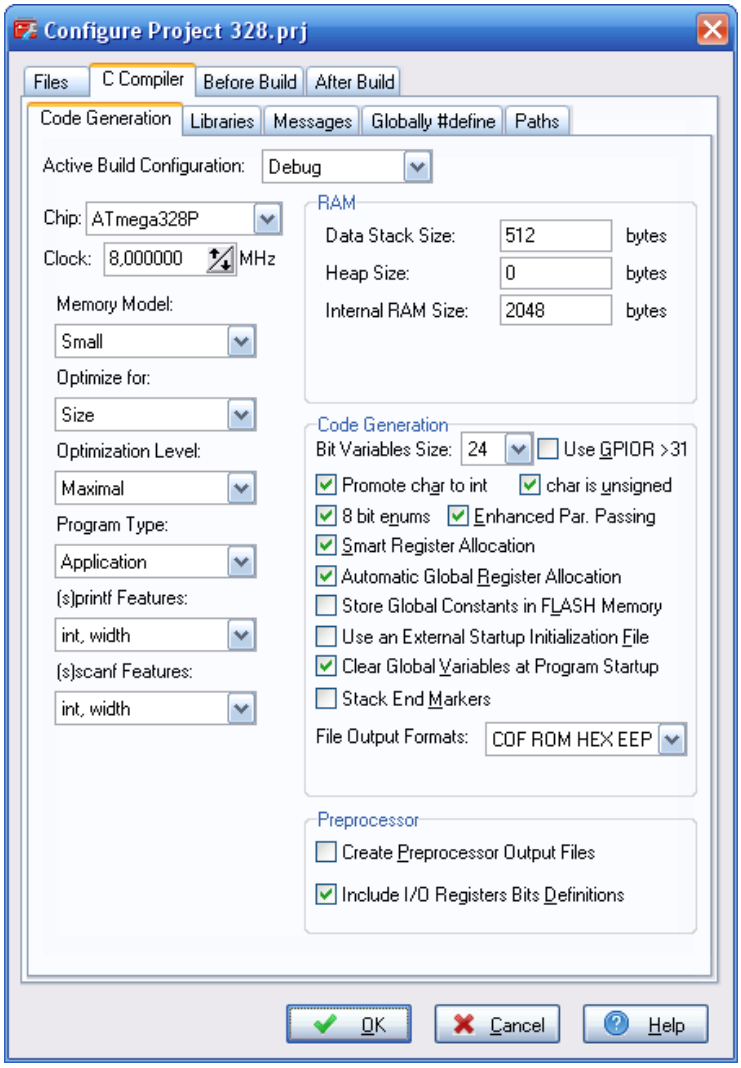
```
#include <io.h>
#include <delay.h>
void main(void)
{
DDRB =255;
PORTB = 0b01010101;
while (1)
    {
     delay_ms(100);
     PORTB = PORTB << 1;  // сдвиг влево
     if (PORTB == 0) PORTB=0b01010101;  //
    }
}
```

Общие примечания

Текст после двух слешей // – это поясняющие код комментарии, программе не мешает.

Не забывайте после каждого изменения программы «перезаряжать» файл прошивки в свойствах микроконтроллера.

После создания проекта можно изменить МК, его частоту и др. так: Project → Configure →



Номера вариантов (возможно) будут перетасованы.

Обращаю внимание, что в разных задачах задействованы разные микроконтроллеры, разные выводы, «противоположные» задачи (типа справа налево / снизу вверх, «туда и обратно»).

На КР задача будет сформулирована чуть подробнее, но программного кода на распечатках не будет. **(Можете взять на КР на флешке эти файлы! На наличие их на компьютерах не надейтесь!)**

Как программировать МК AVR в Протеусе, см. здесь.