

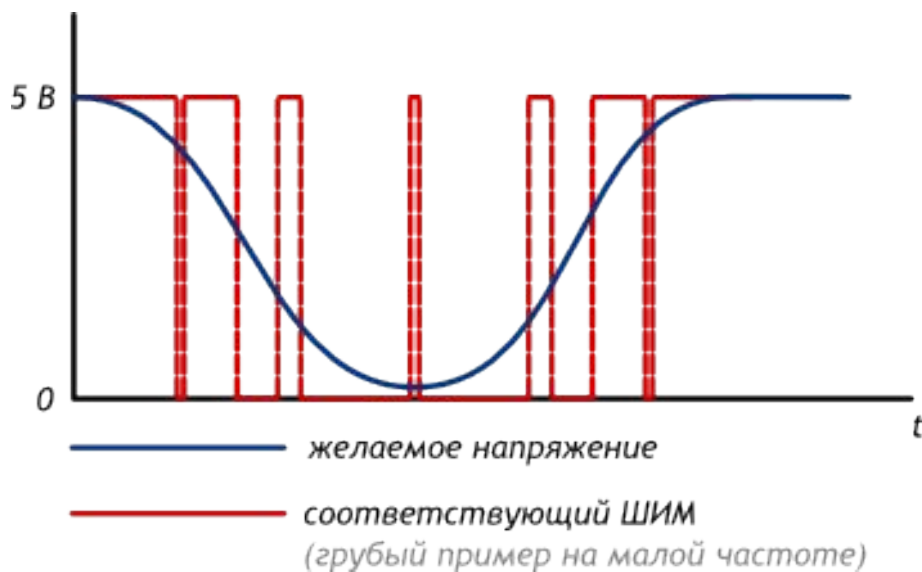
Тема 2

Широтно-импульсная модуляция

Микроконтроллеры обычно не могут выдавать **произвольное напряжение**. Они могут выдать либо напряжение питания (5 В), либо землю (0 В).

Широтно-импульсная модуляция (ШИМ, PWM) – это управление средним значением напряжения на нагрузке путём изменения **скважности** импульсов, управляющих ключом.

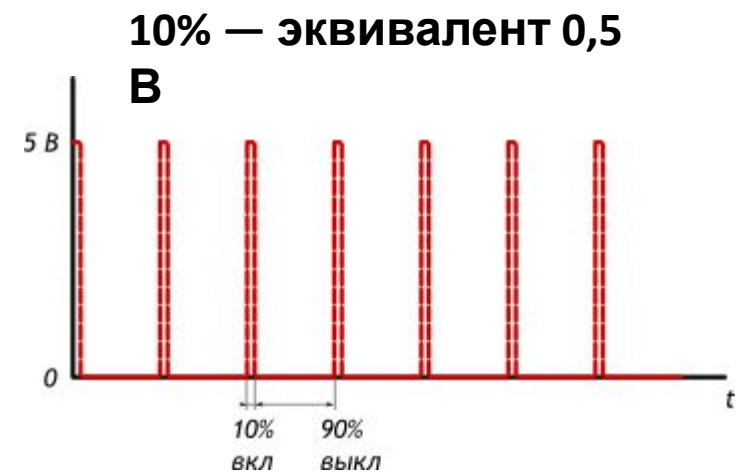
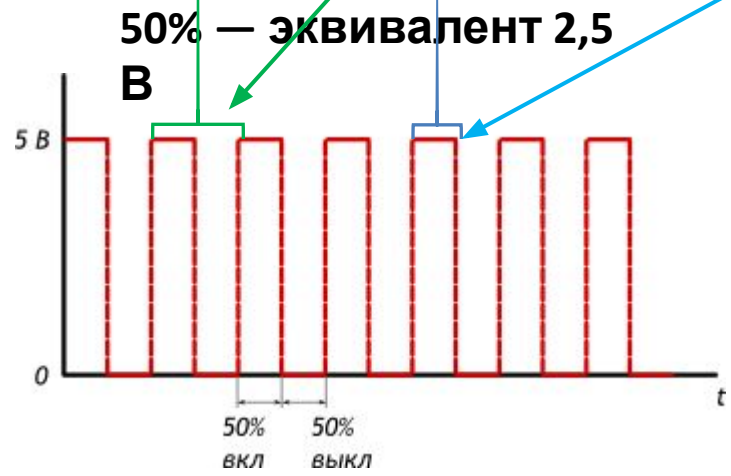
Широтно-импульсная модуляция – это управление тем, какой промежуток времени был **включен** выход микроконтроллера и какой – **выключен**. При этом переключение происходит **часто**.



Частота прямоугольной последовательности импульсов у Arduino приблизительно **16 МГц**, т.е. уровень сигнала меняется от высокого (5 В) к низкому (0 В) приблизительно **16 000 000 раз** каждую секунду.

Скважность – это отношение времени включения и выключения элемента (например, светодиода).

$$S = \text{период импульсов} / \text{длительность импульсов}$$

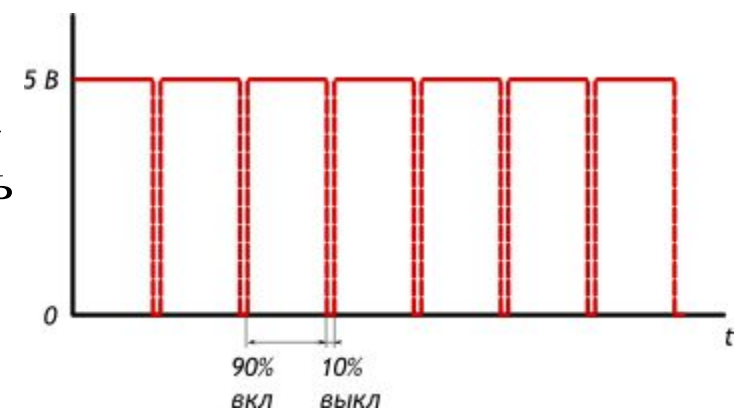


Коэффициент заполнения — это обратная величина скважности

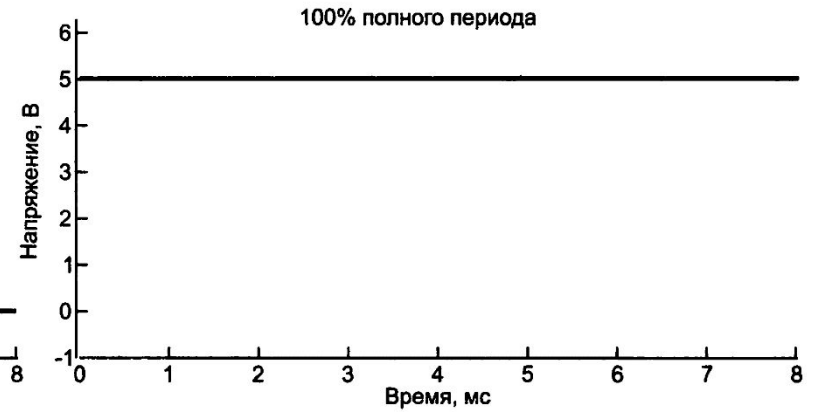
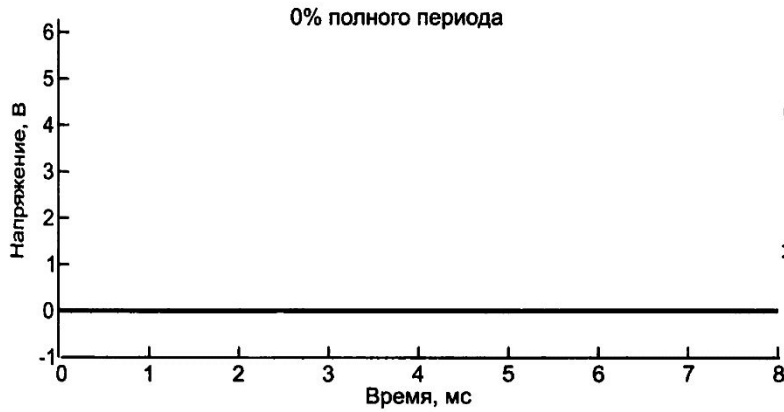
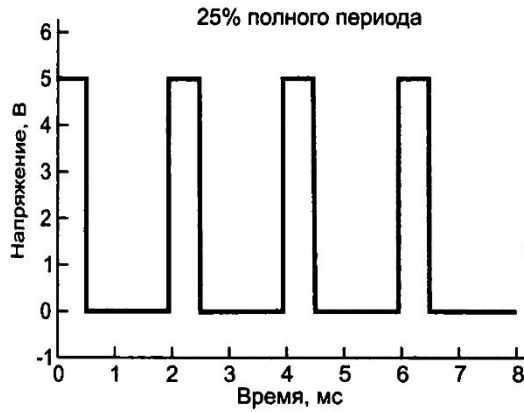
$$D = 1 / S.$$

Это **процент времени**, когда прямоугольный импульс имеет уровень **HIGH**, ко всему периоду повторения.

90% — эквивалент 4,5 В



ШИМ-сигналы с различной скважностью



ШИМ-сигнал можно подать с помощью команды **analogWrite()**

analogWrite() – это встроенная процедура:

подаёт напряжение на заданный пин,
которое может меняться от **0** до **255**,

т.е. **коэффициент заполнения**
(ШИМ-сигнал)

analogWrite(9, 128);

analogWrite(pin, bs);

analogWrite(9, bs);

analogWrite(pin, 128);

Если коэффициент заполнения **D=128**, то светодиод будет гореть,
как будто на него подали **2,5 В**.

Если **D=0**, то это постоянные **0 В**.

Если **D=255**, то это постоянные **5 В**.

пример – p06_led_fade

Плавное управления яркостью светодиода *fade* (увядать)

brightness (яркость)

```
int brightness = 0; // изначальная яркость светодиода
int fadeAmount = 5; // скорость затухания/нарастания
// яркости

void setup()
{
  // инициализируем пин 9, как работающий на выход
  pinMode(9, OUTPUT);
}
```

Не все пины Arduino **поддерживают** работу с ШИМ-сигналами.

Те, которые могут это делать, помечены на плате символом

~ (тильда).

Для Arduino Uno – это пины **3, 5, 6, 9, 10, 11.**

Плавное управления яркостью светодиода

встроенная процедура:
подаёт напряжение на
заданный пин,
которое может меняться
от **0** до **255**,
т.е. **ШИМ-сигнал**

операция «**или**»
при достижении граничного
значения меняем знак
переменной
fadeAmount
на противоположный

```
void loop()
{
  // изменяем яркость светодиода
  analogWrite(9, brightness);

  // на каждом шаге увеличиваем яркость
  // на скорость затухания
  brightness = brightness + fadeAmount;

  // в конце затухания меняем его на
  // нарастание яркости и наоборот
  if (brightness == 0 || brightness == 255)
    fadeAmount = -fadeAmount;

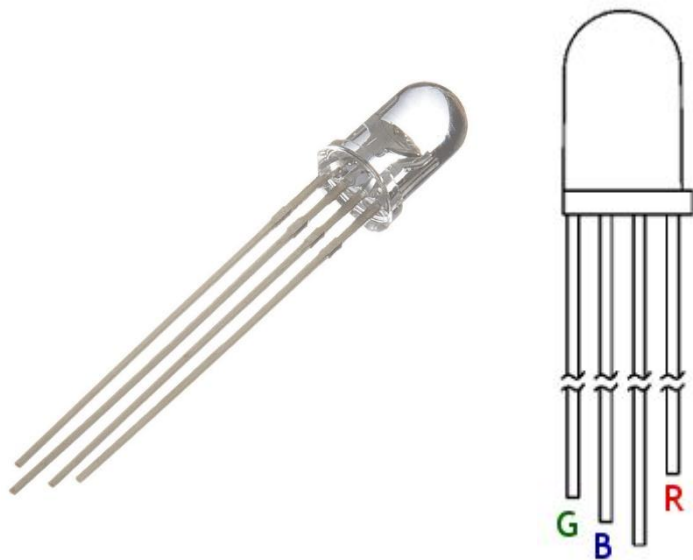
  // ждём 30 миллисекунд
  delay(30);
}
```

коэффициент
заполнения

Трёхцветный светодиод – это три светодиода в одном корпусе: **красный**, **зелёный** и **синий** с одним на всех катодом.

Трёхцветный светодиод ещё называют **RGB-светодиодом**.

Red Green Blue



Внешний вид RGB-светодиода



Обозначение RGB-светодиода на схемах

Светодиод имеет **4** ноги. 3 ноги — **аноды**, соответствующие отдельным цветам и одна — **общий катод**.

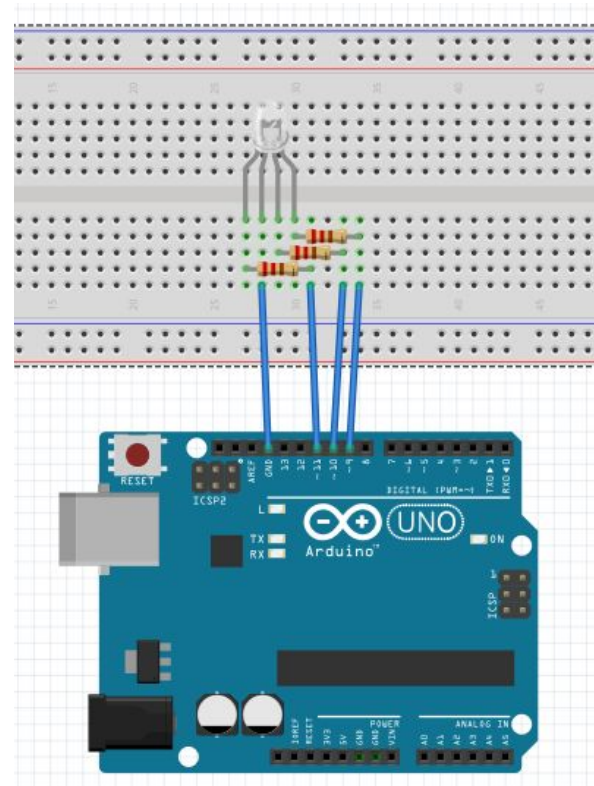
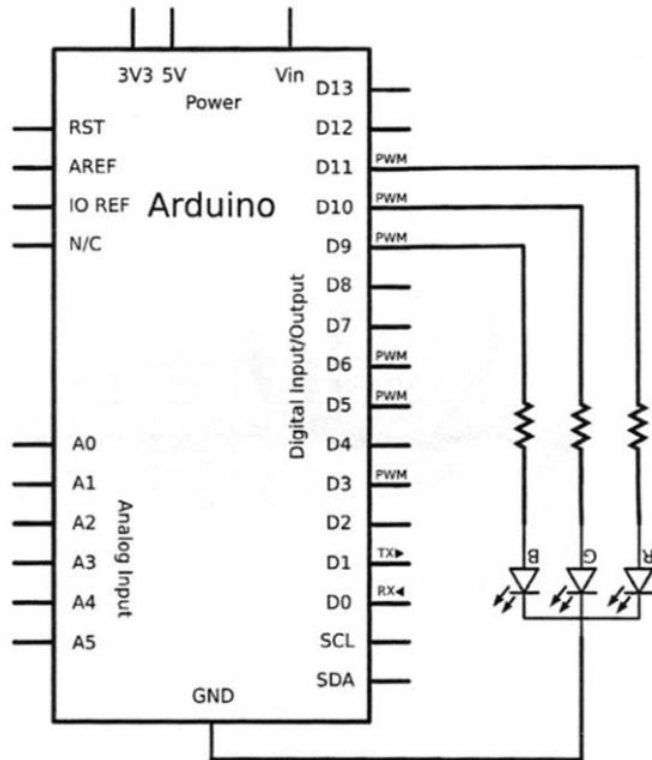
Подавая сигнал на один из анодов, можно добиться свечения одним из цветов.

Используя широтно-импульсную модуляцию для всех анодов одновременно, можно получить свечение **произвольным цветом**. Всего $256^3 = 16\,777\,216$ цветов и оттенков.



**Принцип
смешения цветов**

Схема включения RGB-светодиода



пример – p06_rainbow

rainbow (радуга)

Радуга на Arduino

```
int brightness = 255; // текущее значение яркости
int fadeAmount = 5; // скорость затухания
```

```
// пины, на которых расположен светодиод
int leds[3] = {9, 10, 11};
```

```
// индексы пинов, с которыми мы работаем
// в текущий момент
int curPin[2] = {0, 1};
```

настройка пинов с помощью
оператора **for**

```
void setup()
{
  for(int i = 0; i < 3; i++)
    pinMode(leds[i], OUTPUT);
}
```

Радуга на Arduino

```
int leds[3] = {9, 10, 11};
```

В каждый момент времени управляем **двумя** светодиодами, которые определяются значениями элементов массива **curPin**

Номер светодиода 1	Номер светодиода 2	Тек. знач. яркости
9	10	255
curPin[0]=0	curPin[1]=1	250
		245
		...
		0
<hr/>		
curPin[0] = 1	$0+1\%3=1$	
curPin[1] = 2	$1+1\%3=2$	
10	11	255
		...
		0
<hr/>		
curPin[0] = 2	$1+1\%3=2$	
curPin[1] = 0	$2+1\%3=0$	
11	9	255
		...

```
void loop()
{
    // подаём на два разных пина значение равное
    // brightness и 255-brightness
    analogWrite(leds[curPin[0]], brightness);
    analogWrite(leds[curPin[1]], 255 - brightness);

    // уменьшаем значение brightness
    // на скорость затухания
    brightness = brightness - fadeAmount;

    // когда значение brightness становится равным
    // нулю, меняем номера пинов, с которыми мы
    // работаем
    if (brightness == 0) {
        curPin[0] = (curPin[0] + 1) % 3;
        curPin[1] = (curPin[1] + 1) % 3;
        brightness = 255;
    }
    delay(30);
}
```

Остаток от деления