

## Практическое занятие № Тема: «ШИМ-контроллер»

**Цель работы:** приобрести практические навыки по подключению и программированию светодиодов с помощью ШИМ-контроллера.

### Последовательность выполнения работы:

- Собрать схемы на макетной плате, иначе при отсутствии набора Arduino в web-приложениях (<https://wokwi.com/projects/new/arduino-uno> или <https://www.tinkercad.com/>) для приведенных примеров.
- Запрограммировать микроконтроллер согласно заданию в примере.

### Содержание отчета:

- название практического занятия, его цель;
- фото или скриншоты собранной схемы;
- написанный программный код вставить текстом, Courier New, 12 кегль, одинарный отступ без абзацев;
- вывод о проделанной работе;
- файл Fritzing с принципиальной и монтажной схемой.

## ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

### ПОДКЛЮЧЕНИЕ СВЕТОДИОДОВ

Светодиоды — это полупроводниковые элементы, которые служат для индикации и освещения. Они имеют полярность (+ и —) и чувствуют направление движения постоянного тока. Если подключить светодиод неправильно, то постоянный ток не пройдет и прибор не засветится. Кроме того, светодиод может выйти из строя при неправильном подключении. Анод (длинная ножка светодиода) подключается к плюсу.

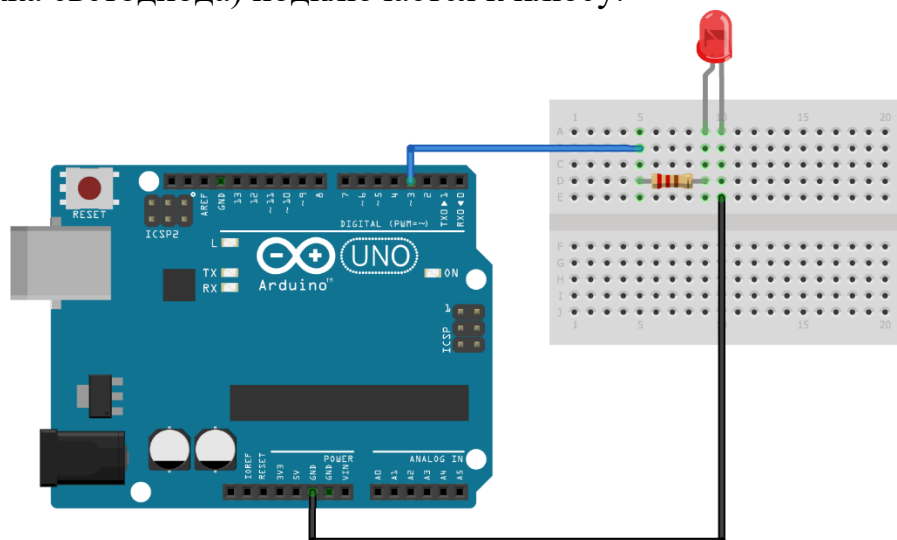


Рисунок 1 – Светодиод

## 1.1 Управление светодиодом с помощью ШИМ Ардуино

На некоторых МК есть ЦАП - цифро-аналоговый преобразователь (DAC, Digital-to-Analog Converter), он преобразует численное значение из программы в соответствующее напряжение в Вольтах и выводит его на пин. Простейший ЦАП можно собрать из десятка резисторов - R2R ЦАП, рассмотрим его в отдельном уроке.

Для большинства задач ЦАП на самом деле и не нужен - его заменяет ШИМ сигнал. В Arduino UNO ШИМ можно реализовать на цифровых выводах, обозначенных «PWM» или «~». На плате Arduino Uno это 3, 5, 6, 10, 11. Скважность импульса задается в пределах от 0 до 255.

Если светодиод половину времени будет включён, а половину выключен, то визуально будет казаться, что он светится в половину своей яркости. Это называется широтно-импульсная модуляция (ШИМ или PWM по-английски). С помощью ШИМ можно управлять «аналоговым» компонентом с помощью цифрового кода.

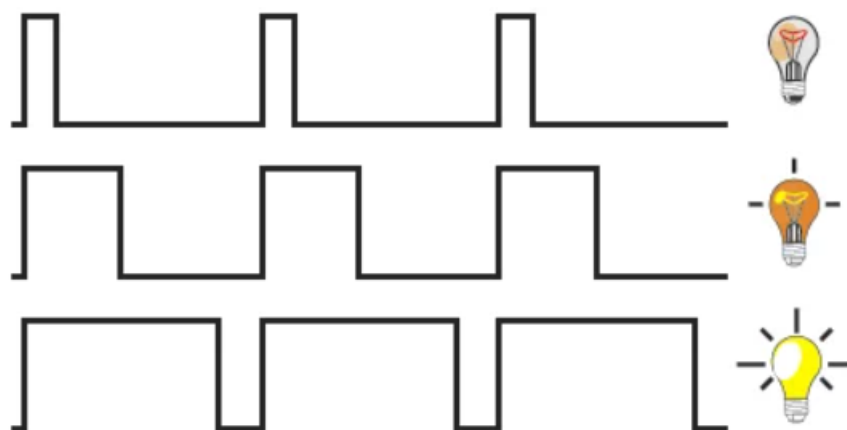


Рисунок 2 – Демонстрация ШИМ

Для реализации ШИМ применяется команда `analogWrite` имеющий следующий синтаксис:

```
analogWrite(вывод, значение);
```

**вывод** - номер вывода ШИМ,

**значение** - значение ШИМ (0-255), применительно к светодиодам это будет их яркость свечения.

**0** - всегда выключен, **255** - всегда включен.

## ЗАДАНИЯ

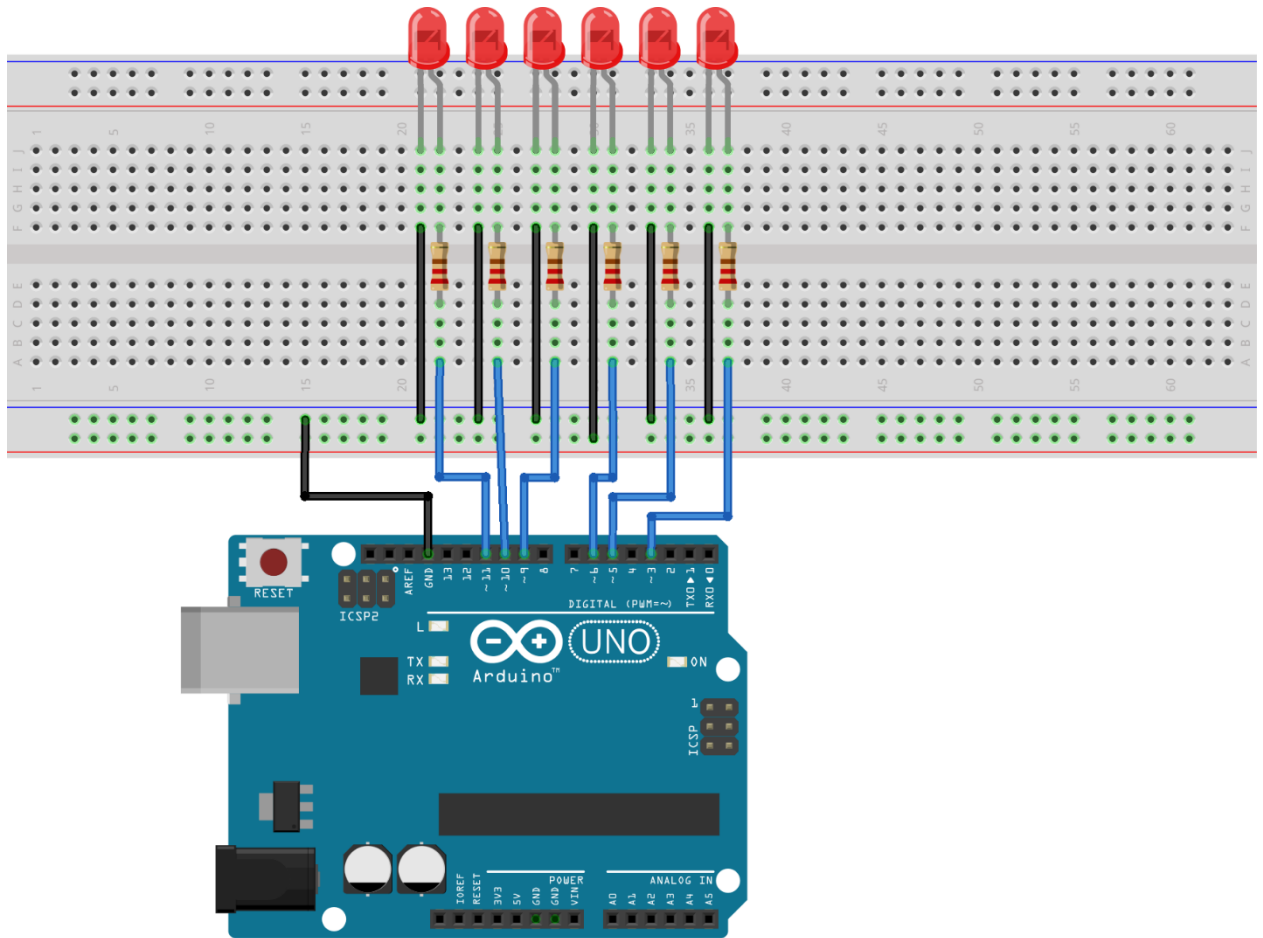


Рисунок 3 – Схема подключения

### Программа 1:

```
int ledPin = 3; //к 3 выводу подсоединен светодиод

void setup() {
  pinMode(ledPin, OUTPUT); // инициализация пина на вывод
}

void loop() {
  analogWrite(ledPin, 255); //полная яркость (255/255 = 1)
  delay(1000); // пауза 1 сек
  digitalWrite(ledPin, LOW); //выключить светодиод
  delay(1000); //пауза 1 сек
  analogWrite(ledPin, 191); //яркость на 3/4 (191/255≈
0,75)
  delay(1000); //пауза 1 сек
  digitalWrite(ledPin, LOW); //выключить светодиод
  delay(1000); //пауза 1 сек
```

```

    analogWrite(ledPin, 127); //половина яркости
(127/255≈0,5)
    delay(1000); // пауза 1 сек
    digitalWrite(ledPin, LOW); //выключить светодиод
    delay(1000); //пауза 1 сек
    analogWrite(ledPin, 63); //четверть яркости
(63/255≈0,25)
    delay(1000); // пауза 1 сек
    digitalWrite(ledPin, LOW); //выключить светодиод
    delay(1000); //пауза 1 сек
}

```

### Программа 2:

```

int ledPin = 9; //светодиод подключен к выводу 9
int intensity; //яркость светодиода

void setup() {
    pinMode(ledPin, OUTPUT); // инициализация пина на выход
}

void loop() {
    //плавно увеличиваем яркость (0 to 255)
    for (intensity=0;intensity<=255;intensity++)
    {
        analogWrite(ledPin,intensity);
        delay(200);
    }
    delay(2000); //ждем 2 сек

    //плавно уменьшаем яркость (255 to 0)
    for (intensity=255;intensity>=0;intensity--)
    {
        analogWrite(ledPin,intensity);
        delay(200);
    }
    delay(2000); //ждем 2 сек
}

```

### Программа 3:

```

const int leds[] = {3, 5, 6, 9, 10, 11};
const int numLeds = 6;

void setup() {
    for (int i = 0; i < numLeds; i++) {

```

```

    pinMode(leds[i], OUTPUT);
  }
}

void loop() {
  // Плавное увеличение яркости
  for (int brightness = 0; brightness <= 255; brightness++) {
    for (int i = 0; i < numLeds; i++) {
      analogWrite(leds[i], brightness);
    }
    delay(10);
  }

  // Плавное уменьшение яркости
  for (int brightness = 255; brightness >= 0; brightness--) {
    for (int i = 0; i < numLeds; i++) {
      analogWrite(leds[i], brightness);
    }
    delay(10);
  }
}

```

## Программа 4:

```

const int leds[] = {3, 5, 6, 9, 10, 11};
const int numLeds = 6;

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

void loop() {
  // Увеличение яркости всех светодиодов
  for (int brightness = 0; brightness <= 255; brightness++) {
    for (int i = 0; i < numLeds; i++) {
      analogWrite(leds[i], brightness);
    }
    delay(15);
  }

  // Последовательное уменьшение яркости
  for (int i = 0; i < numLeds; i++) {
    for (int brightness = 255; brightness >= 0; brightness--) {
      analogWrite(leds[i], brightness);
      delay(5);
    }
  }
}

```

