

## Практическое занятие №

### Тема: «Световая волна»

**Цель работы:** приобрести практические навыки по подключению и программированию Семисегментного индикатора и сдвигового регистра 74НС595N на платформе Arduino.

#### 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.
- Ответить на контрольные вопросы.

#### 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;
- ответы на контрольные вопросы.

## ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Светодиод (LED) — полупроводниковый прибор, излучающий свет при прохождении тока. Подключается через резистор (220 Ом) для ограничения тока.

Цифровые пины Arduino (2-13) могут работать как выходы (OUTPUT), подавая напряжение +5V (HIGH) или 0V (LOW).

#### *Функции Arduino:*

`pinMode(pin, mode)` — настройка пина как входа (INPUT) или выхода (OUTPUT).

`digitalWrite(pin, value)` — установка HIGH/LOW на пин.

`delay(ms)` — пауза в миллисекундах.

`random(min, max)` — генерация случайного числа в диапазоне [min, max).

#### *Логика программ:*

Программа 1: последовательное включение/выключение светодиодов (волна).

Программа 2: случайное включение/выключение.

Программа 3: чередование двух режимов с использованием функций.

**НЕОБХОДИМО ВОСПРОИЗВЕСТИ СХЕМУ В ПРОГРАММЕ FRITZING В РЕЖИМЕ МОНТАЖА И ПРИНЦИПИАЛЬНУЮ СХЕМУ**

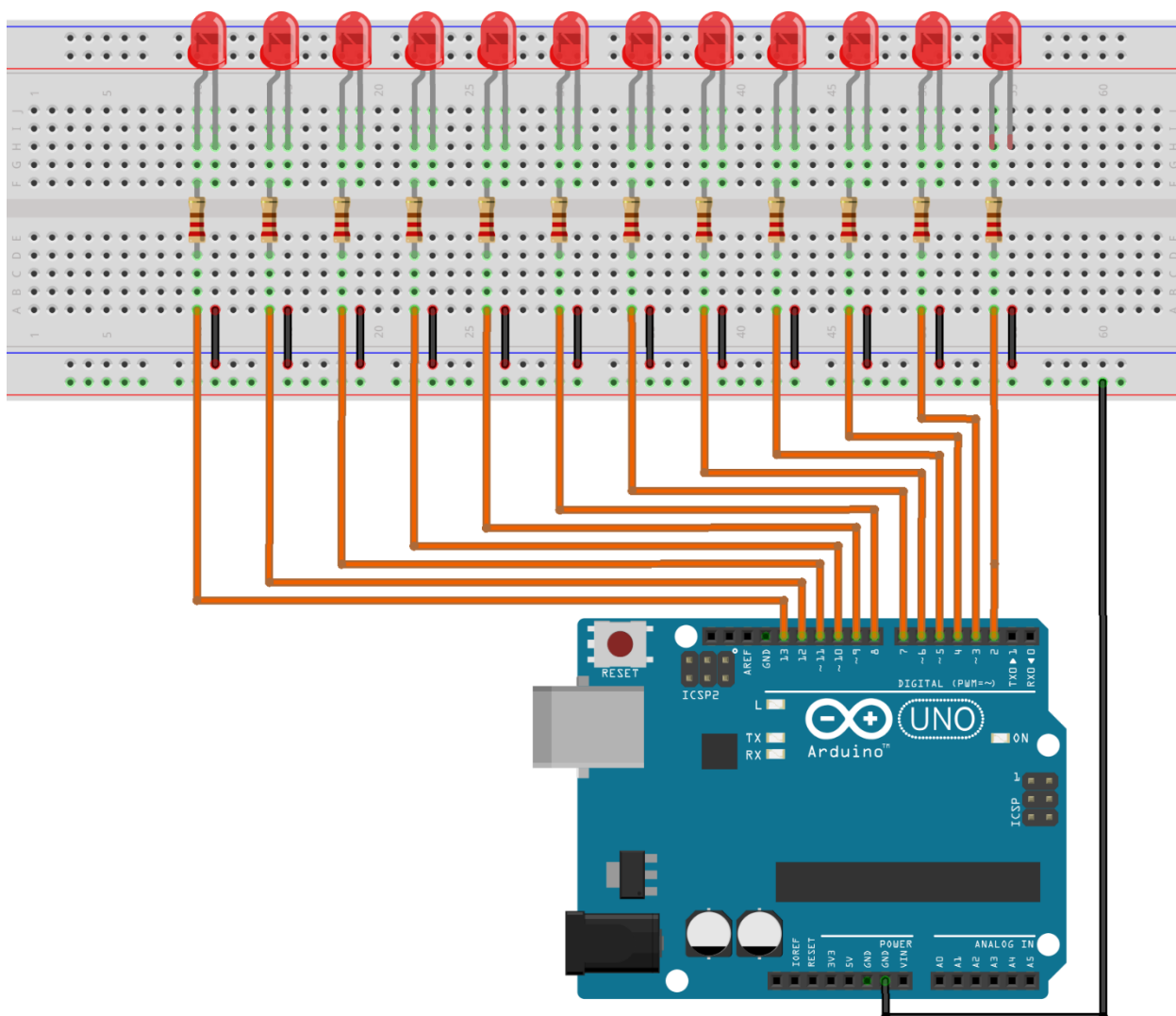


Рисунок 1 – Схема подключения

***Подключение компонентов***

Каждый светодиод подключен анодом (+) к пину 2-13 через резистор 220 Ом, катодом (-) к GND. Резистор необходим в схеме для предотвращения перегорания светодиода.

## ЗАДАНИЯ

### Программа 1: Последовательная световая волна

```
// Пины для светодиодов
const int firstPin = 2;
const int lastPin = 13;

void setup() {
  // Настройка пинов 2-13 как выходы
  for (int pin = firstPin; pin <= lastPin; pin++) {
    pinMode(pin, OUTPUT);
  }
}

void loop() {
  // Включение светодиодов от пина 2 к 13
  for (int pin = firstPin; pin <= lastPin; pin++) {
    digitalWrite(pin, HIGH); // Включить светодиод
    delay(100);              // Задержка 100 мс
  }

  delay(1000); // Пауза 1 секунда

  // Выключение светодиодов от пина 2 к 13
  for (int pin = firstPin; pin <= lastPin; pin++) {
    digitalWrite(pin, LOW); // Выключить светодиод
    delay(100);             // Задержка 100 мс
  }
}
```

### Программа 2: Случайное включение/выключение

```
const int firstPin = 2;
const int lastPin = 13;
const int numPins = lastPin - firstPin + 1; //
Количество светодиодов (12)
bool ledState[12]; // Массив состояний светодиодов

void setup() {
  for (int pin = firstPin; pin <= lastPin; pin++) {
    pinMode(pin, OUTPUT);
    ledState[pin - firstPin] = false; // Изначально
все выключены
  }
}
```

```

    randomSeed(analogRead(0));    //      Инициализация
генератора случайных чисел
}

void loop() {
    // Включение всех светодиодов случайным порядком
    while (true) {
        int randomPin = random(firstPin, lastPin + 1);
// Случайный пин
        if (!ledState[randomPin - firstPin]) {    //
Если светодиод выключен
            digitalWrite(randomPin, HIGH);
            ledState[randomPin - firstPin] = true;
            delay(1000);
        }
        // Проверка, все ли светодиоды включены
        bool allOn = true;
        for (int i = 0; i < numPins; i++) {
            if (!ledState[i]) allOn = false;
        }
        if (allOn) break;
    }

    delay(1000);

    // Выключение всех светодиодов случайным порядком
    while (true) {
        int randomPin = random(firstPin, lastPin + 1);
        if (ledState[randomPin - firstPin]) { // Если
светодиод включен
            digitalWrite(randomPin, LOW);
            ledState[randomPin - firstPin] = false;
            delay(1000);
        }
        // Проверка, все ли выключены
        bool allOff = true;
        for (int i = 0; i < numPins; i++) {
            if (ledState[i]) allOff = false;
        }
        if (allOff) break;
    }
}
}

```

### Программа 3: Случайное включение/выключение

```
const int firstPin = 2;
const int lastPin = 13;
const int numPins = lastPin - firstPin + 1;
bool ledState[12];
int programStep = 0; // Шаг программы: 0 - режим 1,
1 - режим 2

void setup() {
  for (int pin = firstPin; pin <= lastPin; pin++) {
    pinMode(pin, OUTPUT);
    ledState[pin - firstPin] = false;
  }
  randomSeed(analogRead(0));
}

void loop() {
  if (programStep == 0) {
    waveEffect(); // Режим 1: волна
    programStep = 1;
  } else {
    randomEffect(); // Режим 2: случайный порядок
    programStep = 0;
  }
}

// Функция для последовательной волны
void waveEffect() {
  for (int pin = firstPin; pin <= lastPin; pin++) {
    digitalWrite(pin, HIGH);
    delay(100);
  }
  delay(1000);
  for (int pin = firstPin; pin <= lastPin; pin++) {
    digitalWrite(pin, LOW);
    delay(100);
  }
}

// Функция для случайного включения/выключения
void randomEffect() {
  // Сброс состояний светодиодов
  for (int i = 0; i < numPins; i++) ledState[i] =
false;
}
```

```
// Включение в случайном порядке
while (true) {
    int randomPin = random(firstPin, lastPin + 1);
    if (!ledState[randomPin - firstPin]) {
        digitalWrite(randomPin, HIGH);
        ledState[randomPin - firstPin] = true;
        delay(1000);
    }
    bool allOn = true;
    for (int i = 0; i < numPins; i++) {
        if (!ledState[i]) allOn = false;
    }
    if (allOn) break;
}

delay(1000);

// Выключение в случайном порядке
while (true) {
    int randomPin = random(firstPin, lastPin + 1);
    if (ledState[randomPin - firstPin]) {
        digitalWrite(randomPin, LOW);
        ledState[randomPin - firstPin] = false;
        delay(1000);
    }
    bool allOff = true;
    for (int i = 0; i < numPins; i++) {
        if (ledState[i]) allOff = false;
    }
    if (allOff) break;
}
}
```

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какой резистор используется и зачем?
2. Почему в loop() первой программы два цикла for?
3. Что делает delay(100) в волновом эффекте?
4. Зачем в программе 2 массив ledState?
5. Как работает random(firstPin, lastPin + 1)?
6. Почему в программе 2 используется randomSeed(analogRead(0))?
7. Как проверяется, что все светодиоды включены?
8. Что означает bool allOn = true?
9. Как программа 3 переключается между режимами?
10. Зачем в randomEffect() сбрасывать ledState?
11. Почему пины 0 и 1 не используются?
12. Что произойдет, если убрать delay(1000) между включением и выключением?
13. Как изменить скорость волны в программе 1?
14. Можно ли использовать пины 0 и 1 для светодиодов? Почему?
15. Как избежать повторного включения одного светодиода в случайном режиме?
16. Какая роль переменной programStep?
17. Почему в waveEffect() нет проверки состояний светодиодов?
18. Как добавить третий режим в программу 3?
19. Что делает digitalWrite(pin, LOW) в цикле выключения?
20. Как изменить порядок выключения на обратный в программе 1?

## ОТВЕТЫ НА ВОПРОСЫ

1. Резистор 220 Ом ограничивает ток для защиты светодиода.
2. Первый цикл включает светодиоды, второй — выключает.
3. delay(100) создает плавную анимацию волны.
4. Массив ledState отслеживает состояние каждого светодиода.
5. random() генерирует случайный пин в диапазоне [2, 13].
6. randomSeed() обеспечивает разные последовательности случайных чисел при каждом запуске.
7. Проверка через цикл и флаг allOn.
8. allOn становится false, если хотя бы один светодиод выключен.
9. Переменная programStep чередует вызовы waveEffect() и randomEffect().
10. Сброс ledState нужен для корректной работы случайного режима при повторном вызове.
11. Пины 0 и 1 используются для UART (последовательной связи).
12. Исчезнет пауза между включением и выключением.
13. Изменить значение в delay().
14. Не рекомендуется — это пины последовательного порта.
15. Проверять состояние светодиода через ledState.
16. Хранит текущий режим программы 3.
17. В волновом режиме состояния меняются линейно, проверка не нужна.

18. Добавить новую функцию и изменить логику programStep.
19. Устанавливает на пине LOW, выключая светодиод.
20. Заменить `pin <= lastPin` на `pin >= firstPin` в цикле выключения.