

Проектирование устройств

Домашнее задание №4

Евгений Зеленин

22 ноября 2024 г.

1. Динамическая индикация

Условия задачи. Задание 1. На занятии мы делали бегущий светодиод, для этого мы создавали последовательность, в которой указывали битность светодиодов. Сделайте так, чтобы индикация была разной, например: горят четыре любых светодиода, горят пять любых светодиодов. Можно создать последовательность сложнее, и добиться многоуровневой индикации. Можно использовать семисегментный индикатор. Задание 2.* Подключите семисегментный индикатор через сдвиговый регистр, или создайте каскад из сдвиговых регистров.

Схема устройства

Объединим два задания в одном и сделаем его немного сложнее: подключим 8 RGB светодиодов через 3 сдвиговых регистра и будем управлять как цветом, так и порядком включения светодиодов.

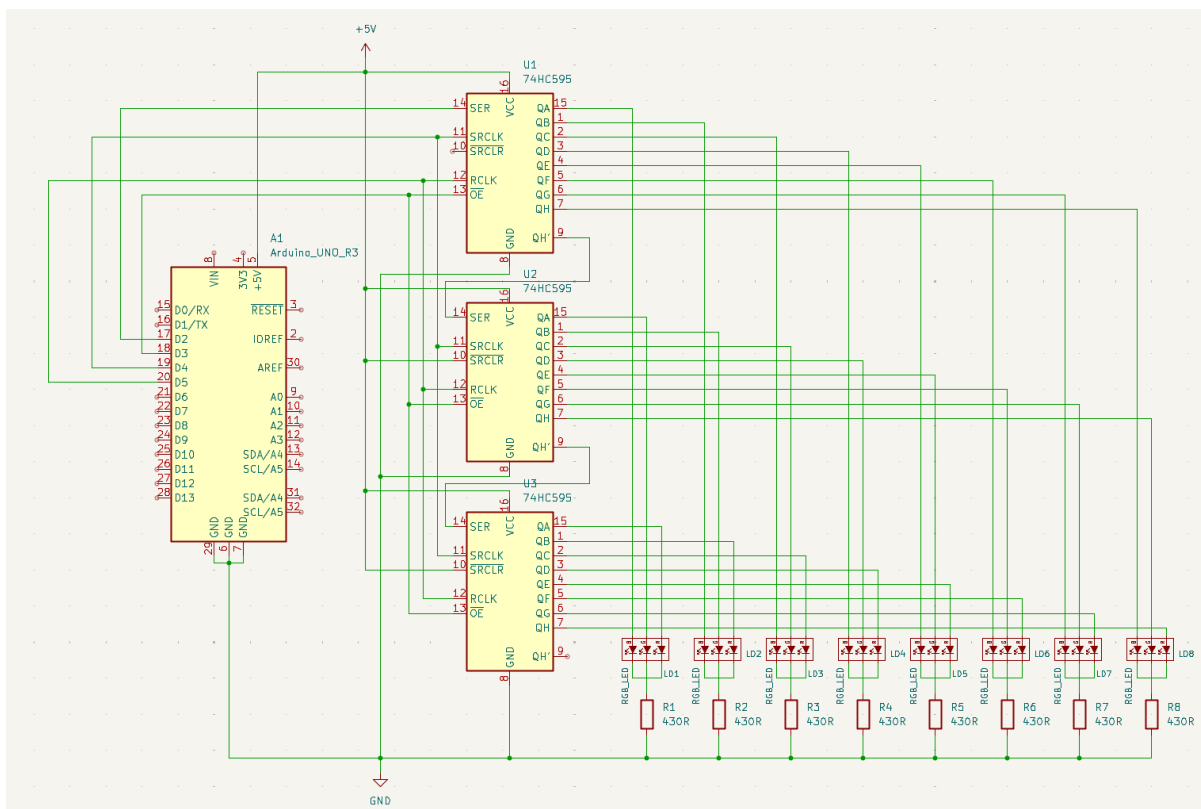


Рис. 1: Схема устройства

Соберем устройство на макетной плате (рисунок 2).

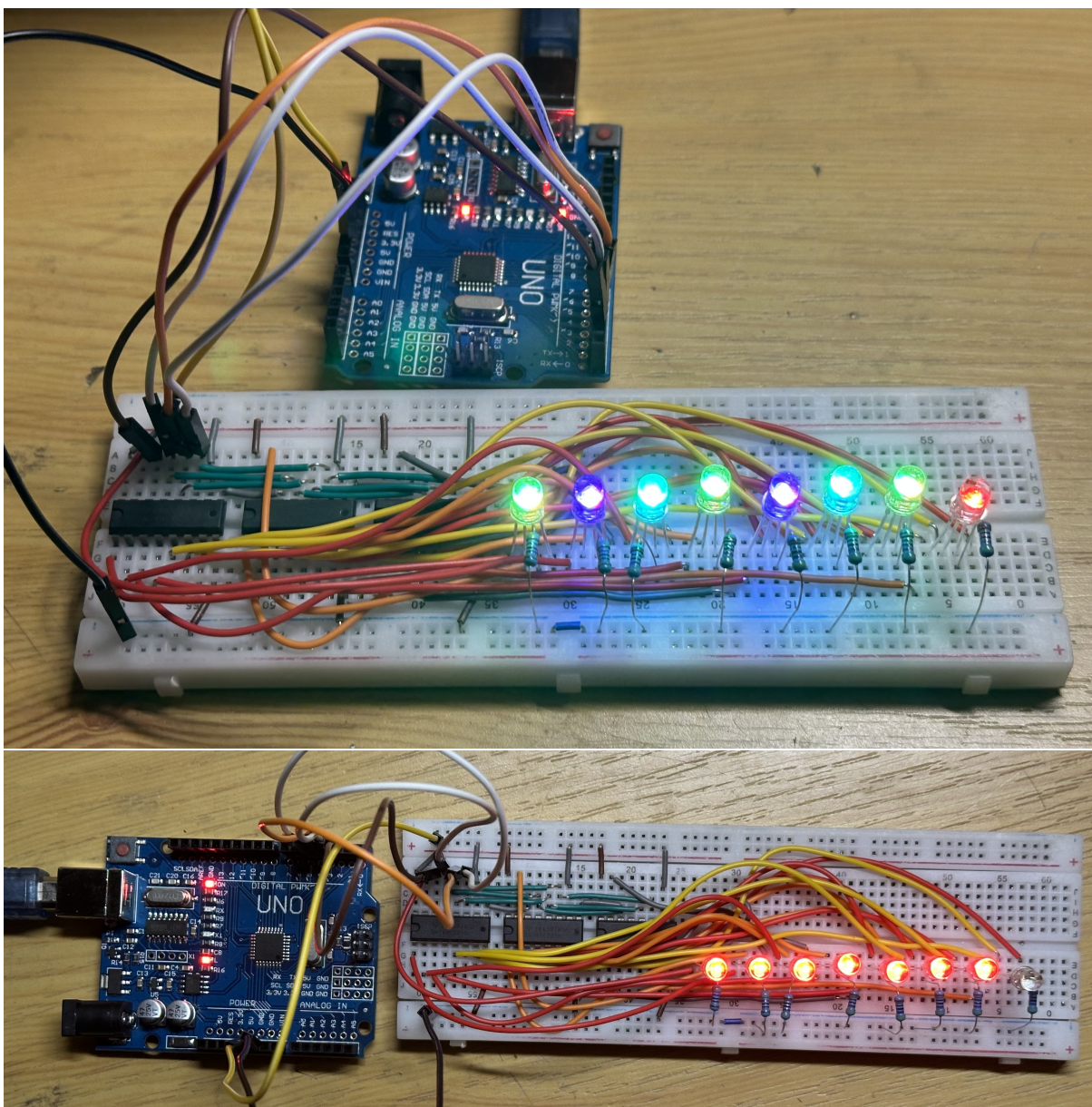


Рис. 2: Сборка макета

Чтобы выводить данные на светодиоды напишем следующий код:

```
uint8_t dataPin = 2;
uint8_t oe_pin = 3;
uint8_t latchPin = 4;
uint8_t clockPin = 5;
enum colors{
    RED = 1,
    GREEN,
    BLUE
};

void setup() {
    pinMode(latchPin, OUTPUT);
    pinMode(clockPin, OUTPUT);
    pinMode(dataPin, OUTPUT);
    pinMode(oe_pin, OUTPUT);
    Serial.begin(9600);
}

void left(uint8_t color, uint8_t pattern); // движение влево left (цвет, шаблон 2 байт)
void right(uint8_t color, uint8_t pattern); // движение вправо
void mirror(uint8_t color, uint8_t pattern); // зеркальное движение
void color(); //игра цветных огней

void loop() {
    analogWrite(oe_pin, 127); // регулировка яркости с помощью ШИМ

    color();

    mirror(RED,1);
    mirror(GREEN,3);
    mirror(BLUE,7);

    left(RED, 255);
    right(RED,255);
    left(RED, 255);
    right(RED,255);

    left(BLUE, 1);
    right(BLUE, 1);

    left(BLUE, 1);
    right(BLUE, 1);

    left(GREEN, 7);
    right(GREEN, 7);
```

```

    left(GREEN, 7);
    right(GREEN, 7);
}

void left(uint8_t color, uint8_t pattern){
    bool red = 0, green = 0, blue = 0;
    switch(color){
        case RED:
            red = 1;
            break;
        case GREEN:
            green = 1;
            break;
        case BLUE:
            blue = 1;
            break;
    }
    for(int i = 0; i < 8; i++){
        digitalWrite(latchPin, LOW);
        shiftOut(dataPin, clockPin, LSBFIRST, pattern * red << i); // red
        shiftOut(dataPin, clockPin, LSBFIRST, pattern * green << i); // green
        shiftOut(dataPin, clockPin, LSBFIRST, pattern * blue << i); // blue
        digitalWrite(latchPin, HIGH);
        delay(70);
    }
}

void right(uint8_t color, uint8_t pattern){
    bool red = 0, green = 0, blue = 0;
    switch(color){
        case RED:
            red = 1;
            break;
        case GREEN:
            green = 1;
            break;
        case BLUE:
            blue = 1;
            break;
    }

    for(int i = 7; i >= 0; i--){
        digitalWrite(latchPin, LOW);
        //используем сдвиг для получения нужного слова
        shiftOut(dataPin, clockPin, LSBFIRST, pattern * red << i); // red
    }
}

```

```

        shiftOut(dataPin, clockPin, LSBFIRST, pattern * green << i); // green
        shiftOut(dataPin, clockPin, LSBFIRST, pattern * blue << i); // blue
        digitalWrite(latchPin, HIGH);
        delay(70);
    }
}

void mirror(uint8_t color, uint8_t pattern)
{
    bool red = 0, green = 0, blue = 0;
    switch(color){
        case RED:
            red = 1;
            break;
        case GREEN:
            green = 1;
            break;
        case BLUE:
            blue = 1;
            break;
    }

    for(int i = 0; i < 4; i++){
        uint8_t data = 0;
        data = pattern << i | pattern << 7 - i;
        digitalWrite(latchPin, LOW);
        shiftOut(dataPin, clockPin, LSBFIRST, data * red); // red
        shiftOut(dataPin, clockPin, LSBFIRST, data * green); // green
        shiftOut(dataPin, clockPin, LSBFIRST, data * blue); // blue

        digitalWrite(latchPin, HIGH);
        delay(80);
    }
    for(int i = 3; i >= 0 ; i--){
        uint8_t data = 0;
        data = pattern << i | pattern << 7 - i;
        digitalWrite(latchPin, LOW);
        shiftOut(dataPin, clockPin, LSBFIRST, data * red); // red
        shiftOut(dataPin, clockPin, LSBFIRST, data * green); // green
        shiftOut(dataPin, clockPin, LSBFIRST, data * blue); // blue

        digitalWrite(latchPin, HIGH);
        delay(80);
    }
}

void color(){

```

```

// с помощью сдвига и побитовых операций формируем нужную последовательность
// для записи в регистры
for(int i = 0; i < 8 ; i++){
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, LSBFIRST, (1 | 1 << 3 | 1 << 6) << i); // red
    shiftOut(dataPin, clockPin, LSBFIRST, (1 << 1 | 1 << 4 | 1 << 7) << i); // green
    shiftOut(dataPin, clockPin, LSBFIRST, (1 << 2 | 1 << 5) << i); // blue

    digitalWrite(latchPin, HIGH);
    delay(80);
}

for(int i = 7; i >= 0 ; i--){
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, LSBFIRST, (1 | 1 << 3 | 1 << 6) << i); // red
    shiftOut(dataPin, clockPin, LSBFIRST, (1 << 1 | 1 << 4 | 1 << 7) << i); // green
    shiftOut(dataPin, clockPin, LSBFIRST, (1 << 2 | 1 << 5) << i); // blue
    digitalWrite(latchPin, HIGH);
    delay(80);
}

for(int i = 0; i < 8 ; i++){
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, LSBFIRST, (1 | 1 << 3 | 1 << 6) >> i); // red
    shiftOut(dataPin, clockPin, LSBFIRST, (1 << 1 | 1 << 4 | 1 << 7) >> i); // green
    shiftOut(dataPin, clockPin, LSBFIRST, (1 << 2 | 1 << 5) >> i); // blue
    digitalWrite(latchPin, HIGH);
    delay(80);
}

for(int i = 7; i >= 0 ; i--){
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, LSBFIRST, (1 | 1 << 3 | 1 << 6) >> i); // red
    shiftOut(dataPin, clockPin, LSBFIRST, (1 << 1 | 1 << 4 | 1 << 7) >> i); // green
    shiftOut(dataPin, clockPin, LSBFIRST, (1 << 2 | 1 << 5) >> i); // blue
    digitalWrite(latchPin, HIGH);
    delay(80);
}
}

```

Видео с демонстрацией работы приложено в материалах к занятию.

2. Материалы к занятию

Схемы, демонстрация работы и прочие материалы к занятию расположены в папке на google диск по следующей ссылке: Материалы к ДЗ №4