

Основы технического волшебства

(Проектная электроника / Электроника для начинающих)

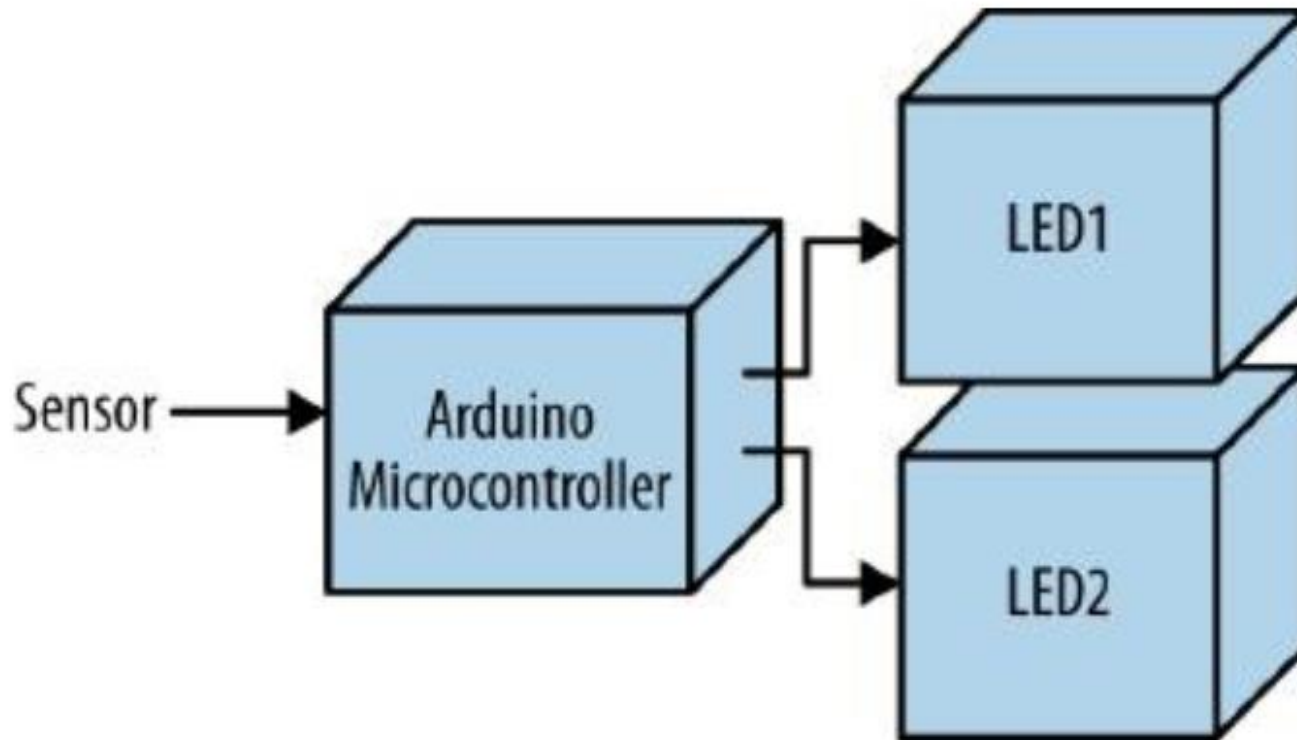
Занятие 27

Светодиоды



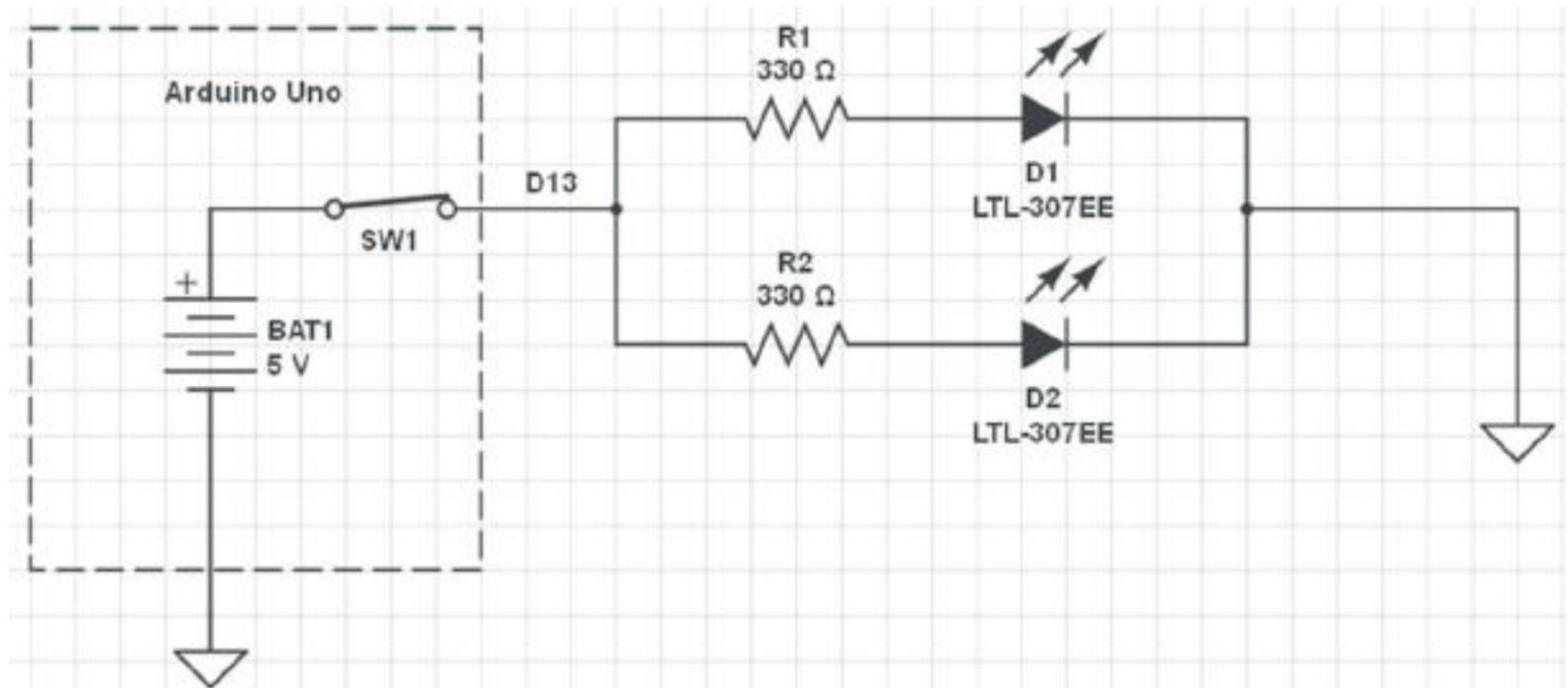
ПРОЕКТ 1: Маячок с двумя светодиодами.

Блок-диаграмма



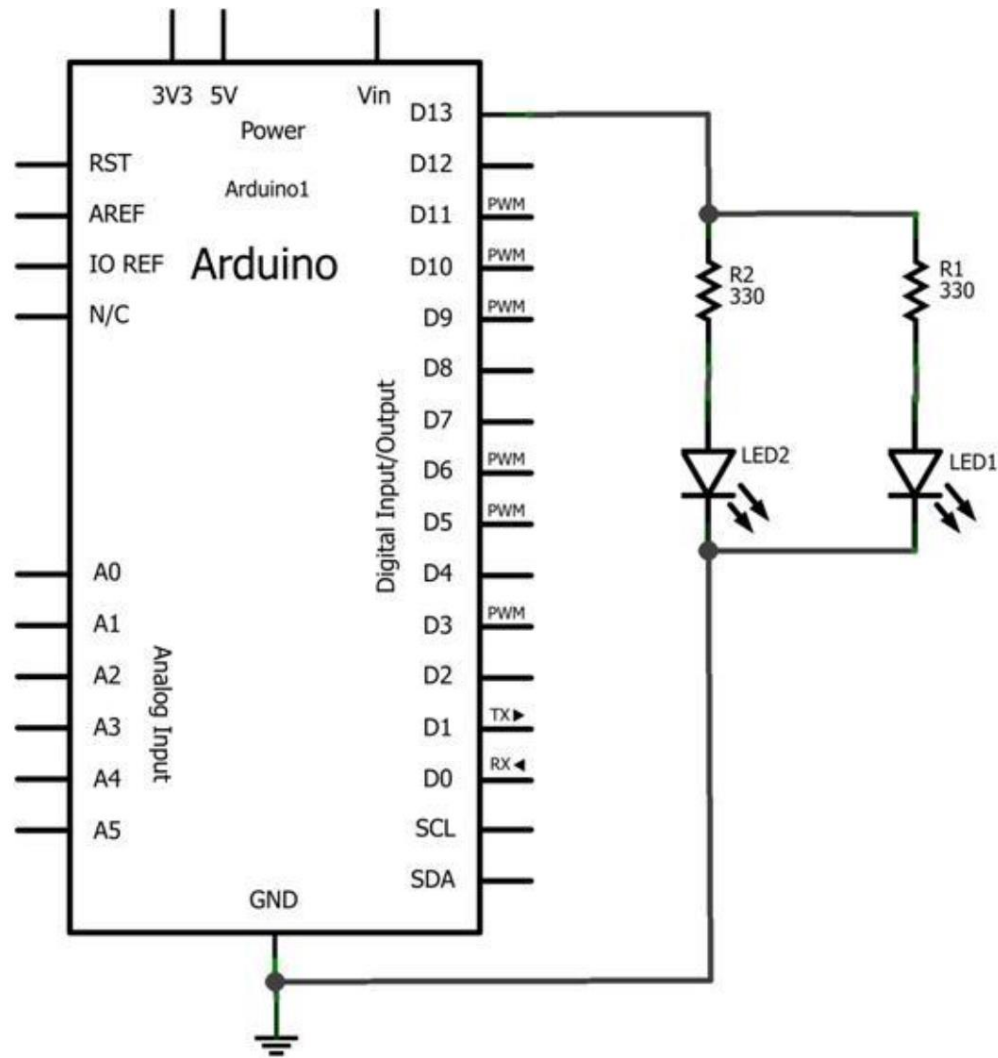
ПРОЕКТ 1: Маячок с двумя светодиодами.

Электрическая цепь (теория)



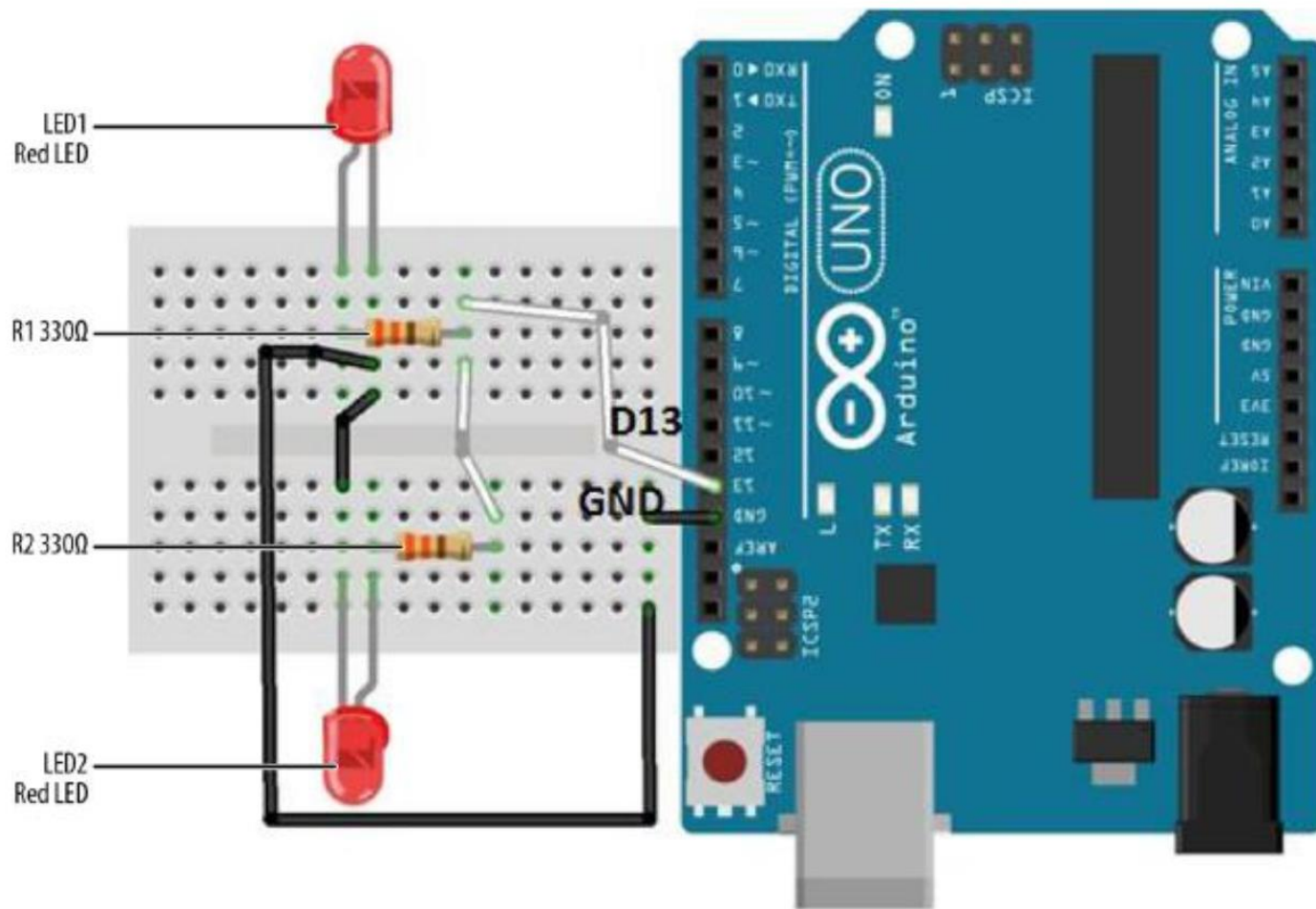
ПРОЕКТ 1: Маячок с двумя светодиодами.

Принципиальная электрическая схема



ПРОЕКТ 1: Маячок с двумя светодиодами.

Монтажная электрическая схема



ПРОЕКТ 1: Маячок с двумя светодиодами.

Программный код (скетч)

```
int led = 13;
```

```
// the setup routine runs once when you press reset:
```

```
void setup() {
```

```
// initialize the digital pin as an output:
```

```
pinMode(led, OUTPUT);
```

```
}
```

```
// the loop routine runs over and over again forever:
```

```
void loop() {
```

```
digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
```

```
delay(1000); // wait for a second
```

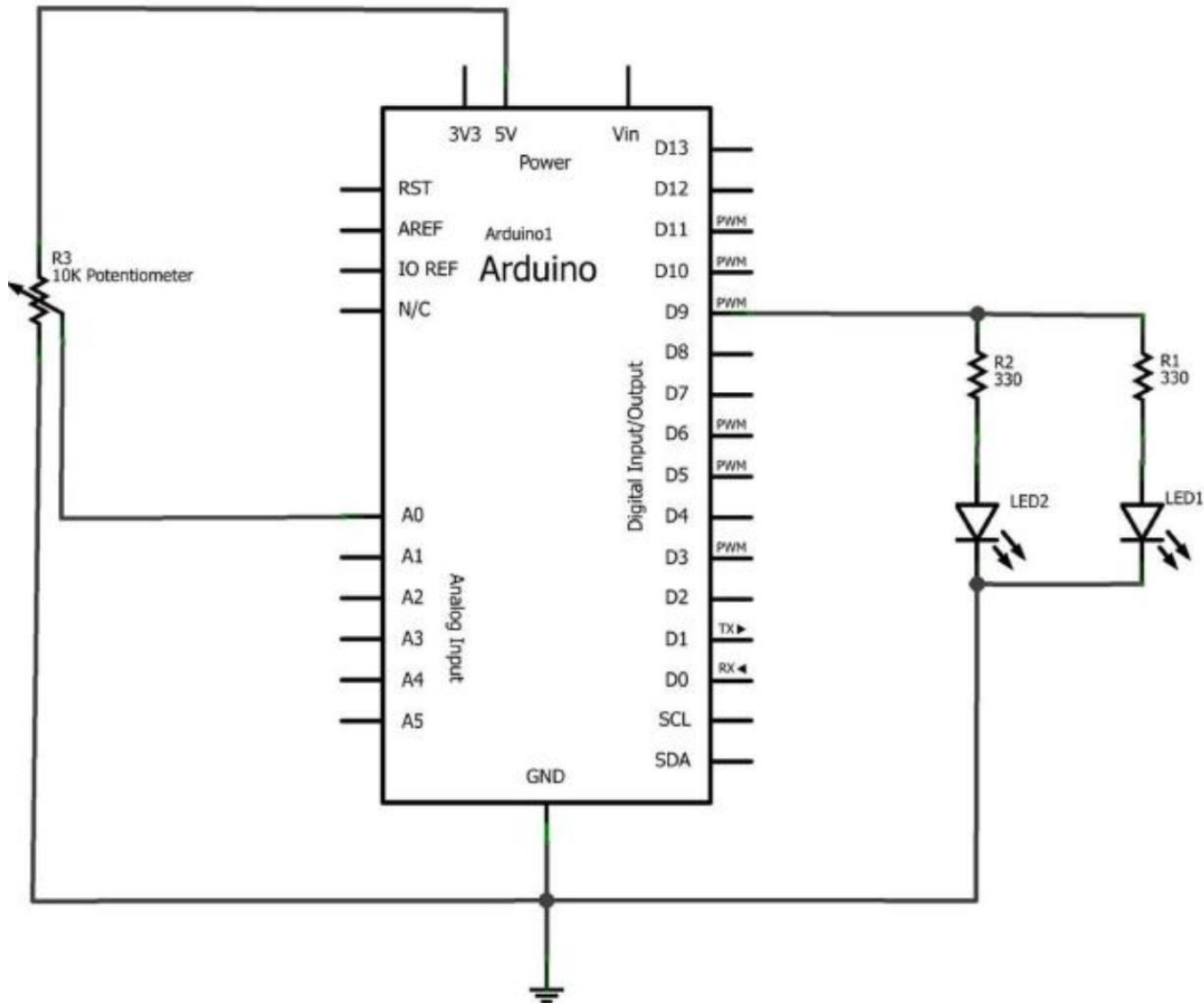
```
digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
```

```
delay(1000); // wait for a second
```

```
}
```

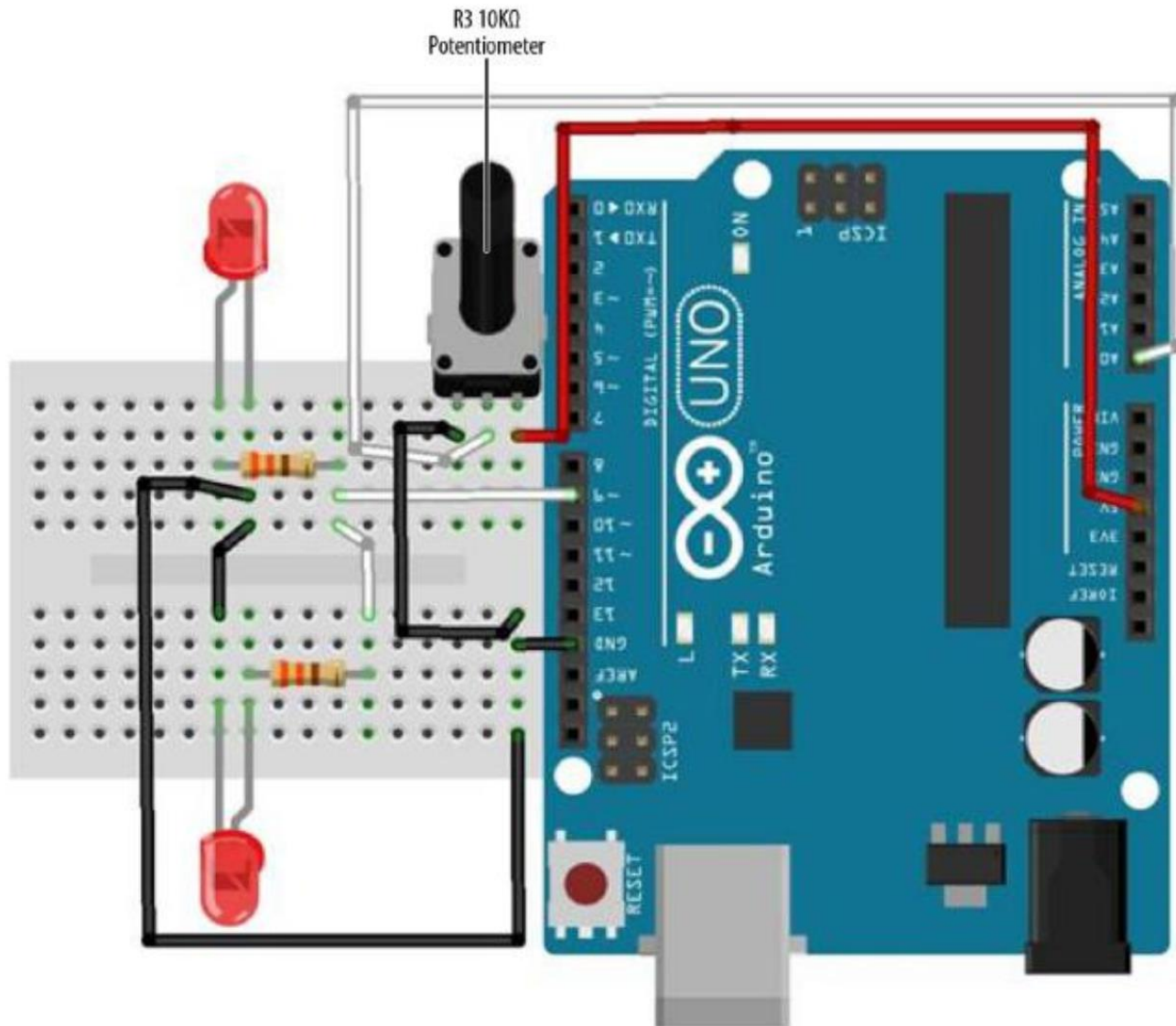
ПРОЕКТ 2: Регулируемый маячок с 2 светодиодами.

Принципиальная электрическая схема



ПРОЕКТ 2: Регулируемый маячок с 2 светодиодами.

Монтажная электрическая схема



ПРОЕКТ 2: Регулируемый маячок с 2 светодиодами.

Программный код (скетч) 1/2

```
int led = 9; // pin D9 assigned to led variable.
```

```
// A 10K potentiometer center pin wired to pin A0.
```

```
// One pin is wired to +5V with the other connected to GND.
```

```
int PotIn = A0; // pin A0 assigned to PotIn variable.
```

```
int Flash; // Flash variable to be used with "delay" instruction.
```

```
// the setup routine runs once when you press reset:
```

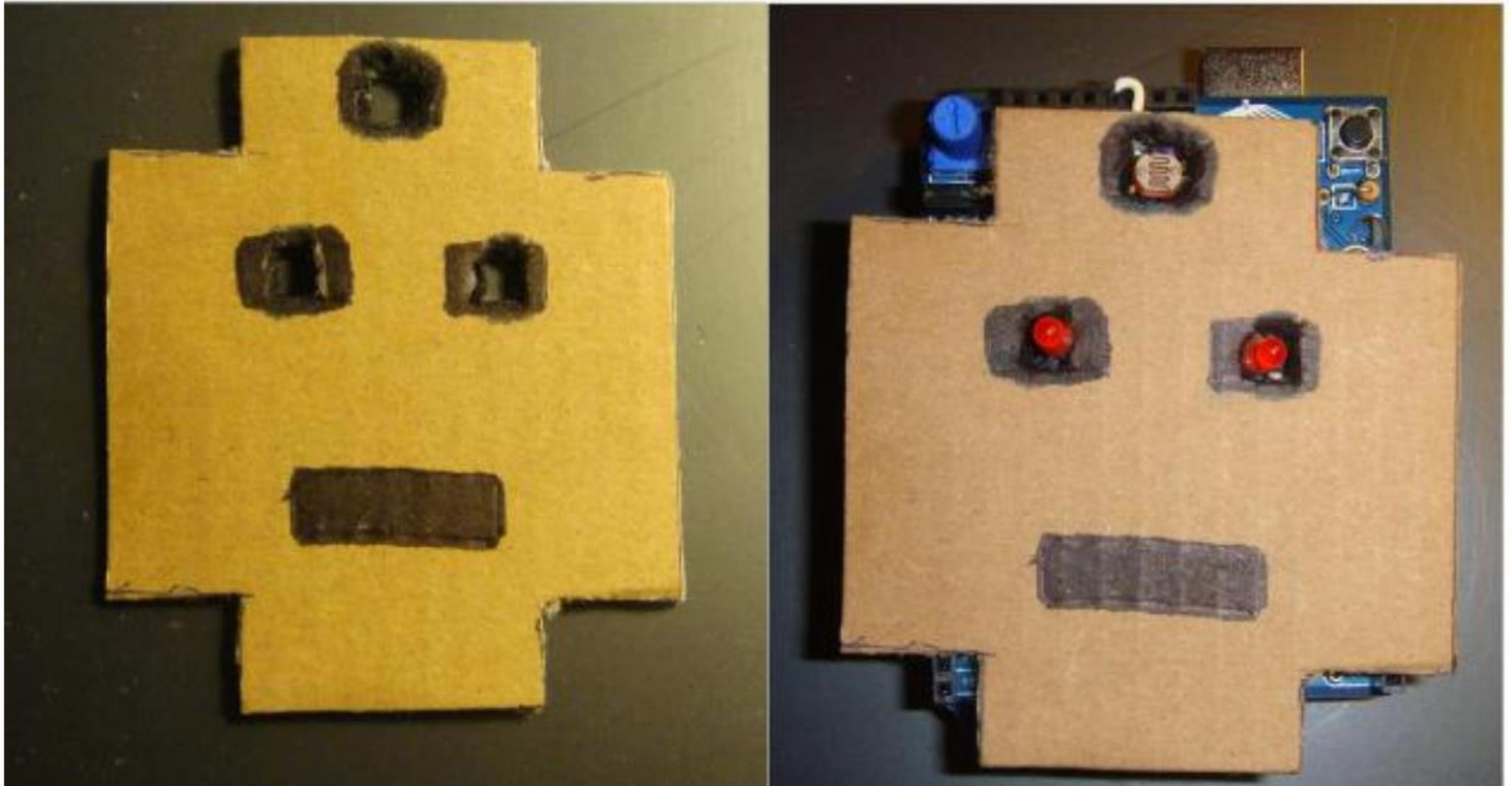
```
void setup() {
```

ПРОЕКТ 2: Регулируемый маячок с 2 светодиодами.

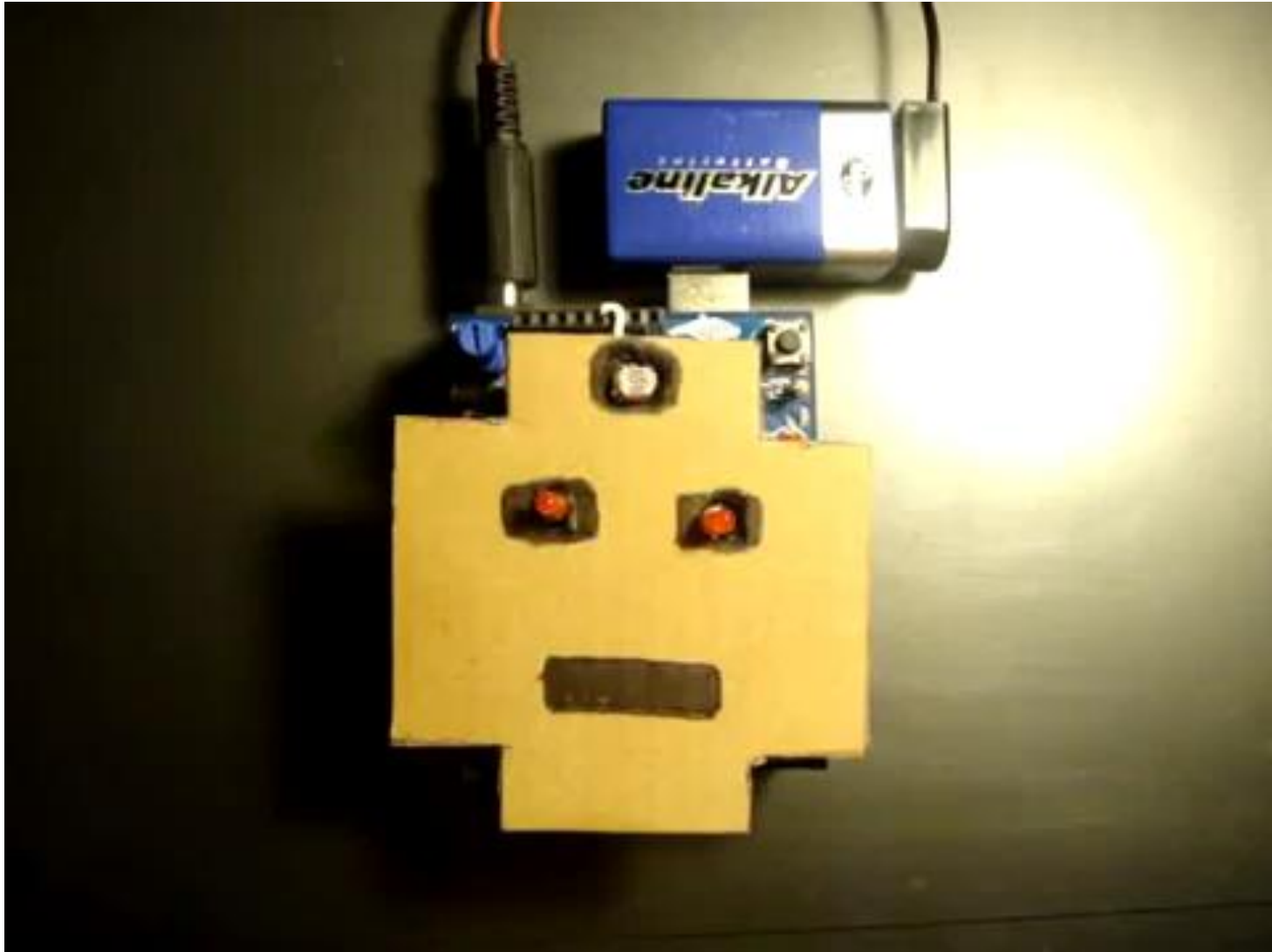
Программный код (скетч) 2/2

```
// initialize the digital pin as an output:  
pinMode(led, OUTPUT);  
// initialize the analog pin as an input:  
pinMode(PotIn, INPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  Flash =analogRead(PotIn); // read 10K pot, store value in Flash variable  
  digitalWrite(led, HIGH); // turn the LED on (HIGH voltage level = on)  
  delay(Flash); // wait for a Flash time delay in seconds  
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW  
  delay(Flash); // wait for a Flash time delay in seconds  
}
```

ПРОЕКТ 3: ОЖИВИ ЕГО. ФранкенБОТ

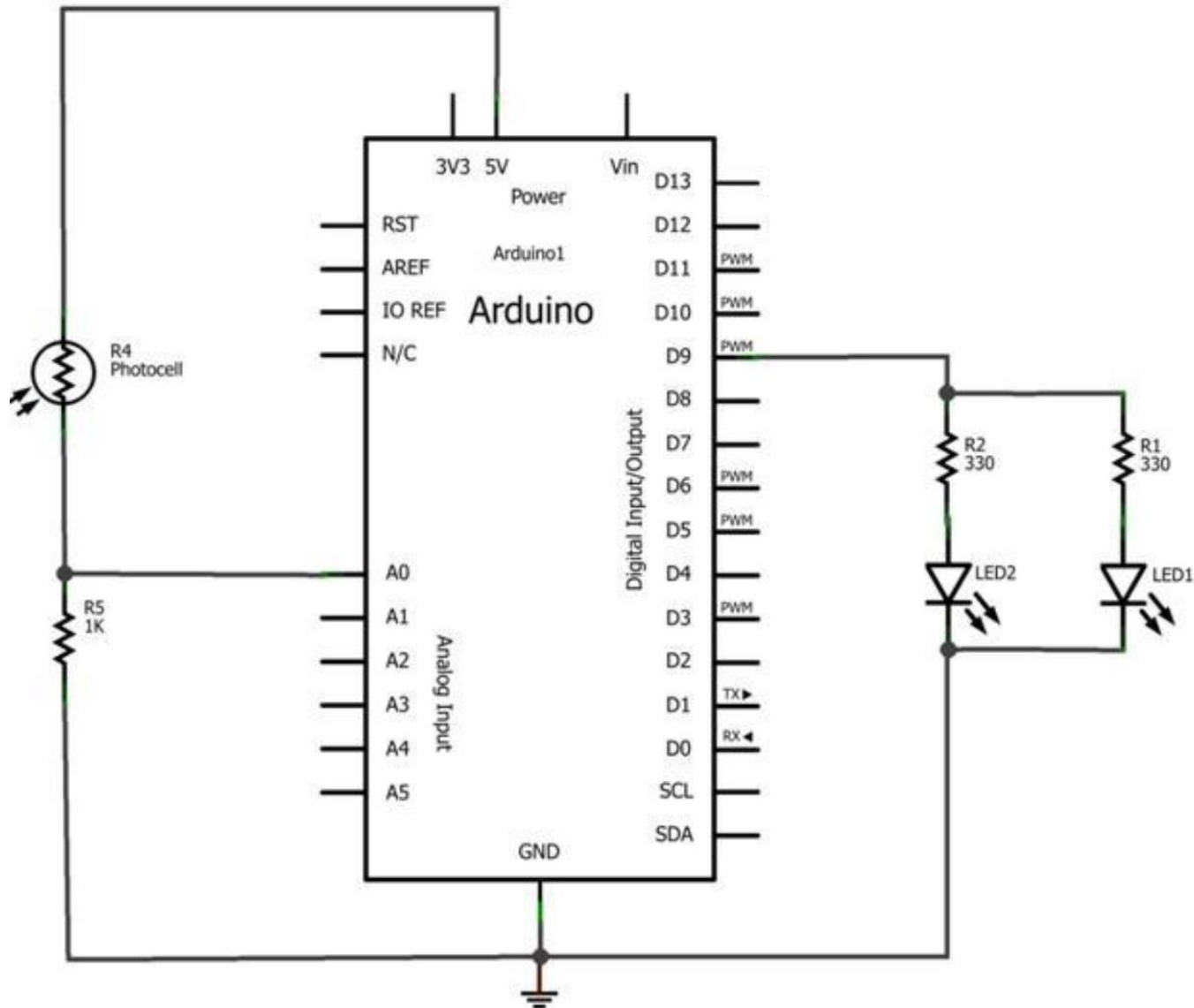


ПРОЕКТ 3: ОЖИВИ ЕГО. ФранкенБОТ



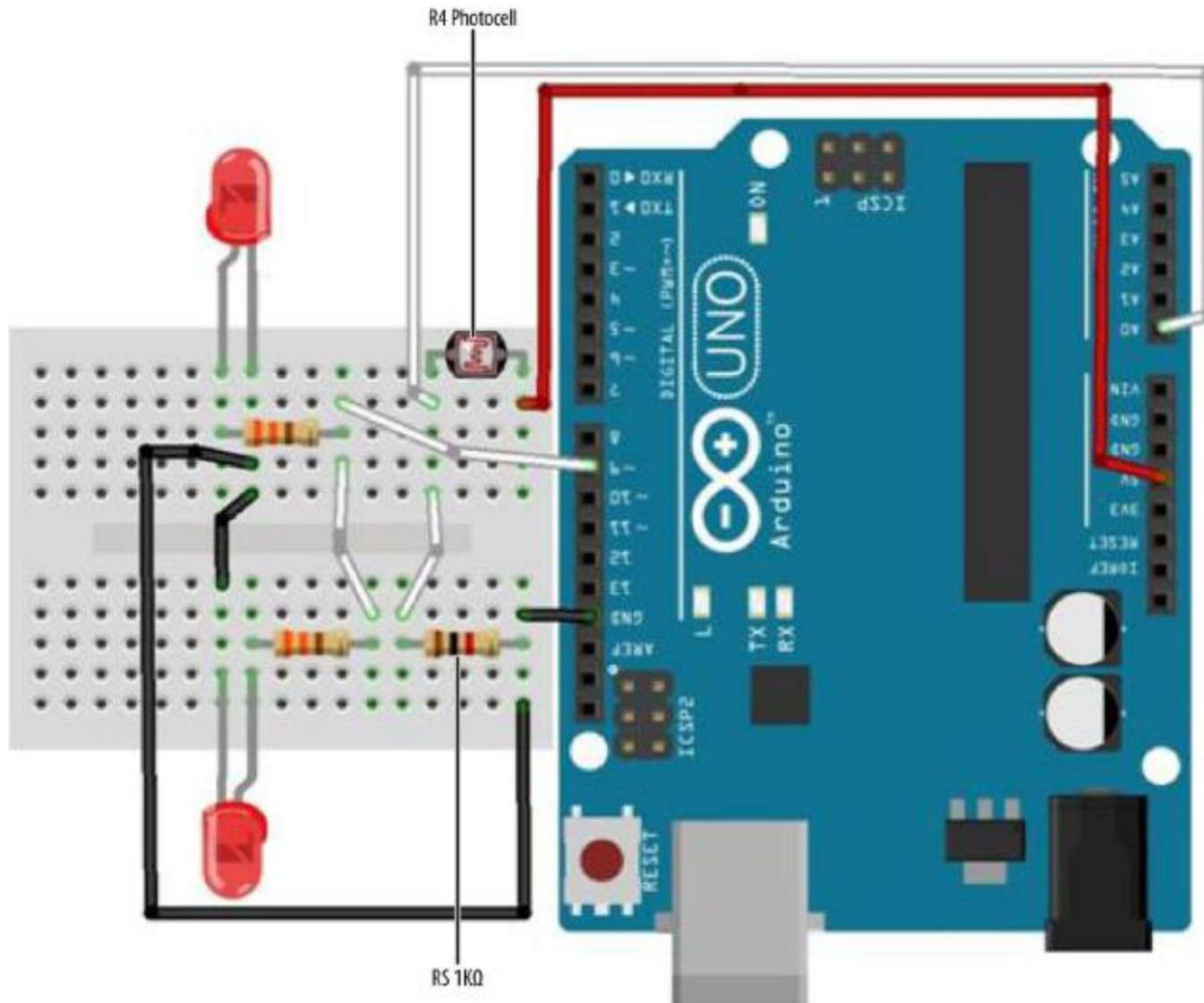
ПРОЕКТ 3: ОЖИВИ ЕГО. ФранкенБОТ

Принципиальная электрическая схема



ПРОЕКТ 3: ОЖИВИ ЕГО. ФранкенБОТ

Монтажная электрическая схема



ПРОЕКТ 4: Улыбка ФранкенБОТа.

**Использую сдвиговый регистр сделай ФранкенБОТу
злую улыбку из светодиодов.**

Схему и код программы разрабатываешь ты сам!

Входы и выходы регистра сдвига

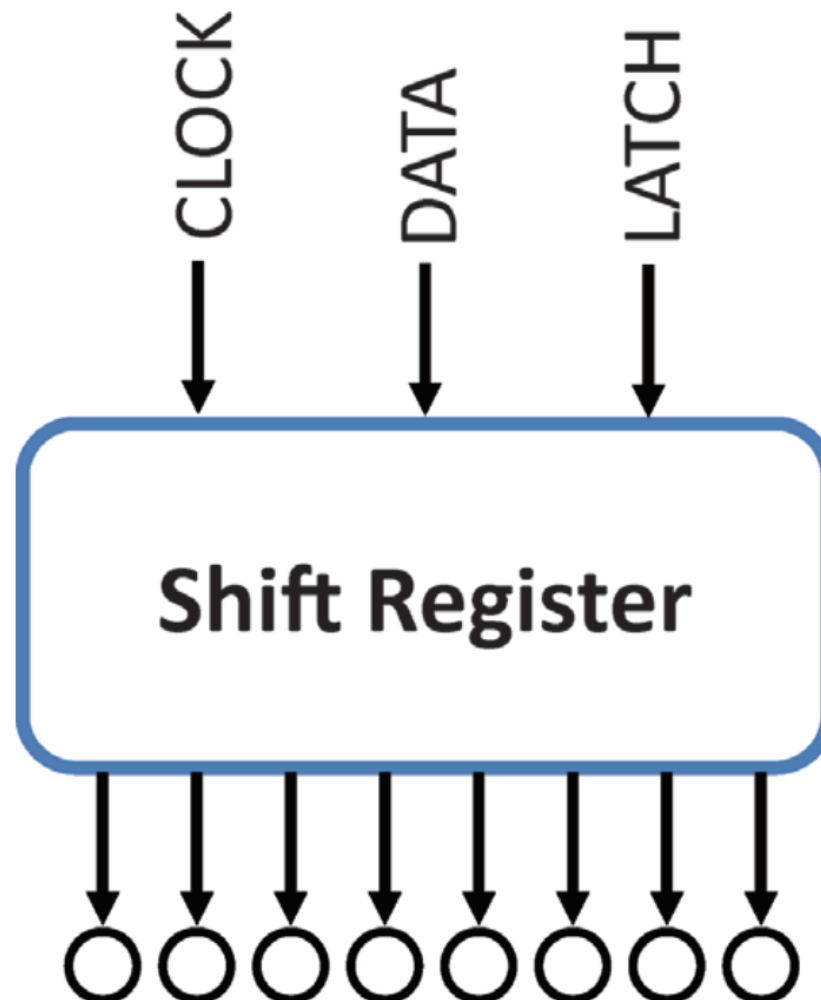


Figure 7-1: Shift register input/output diagram

Выходной сдвиговый регистр 74HC595

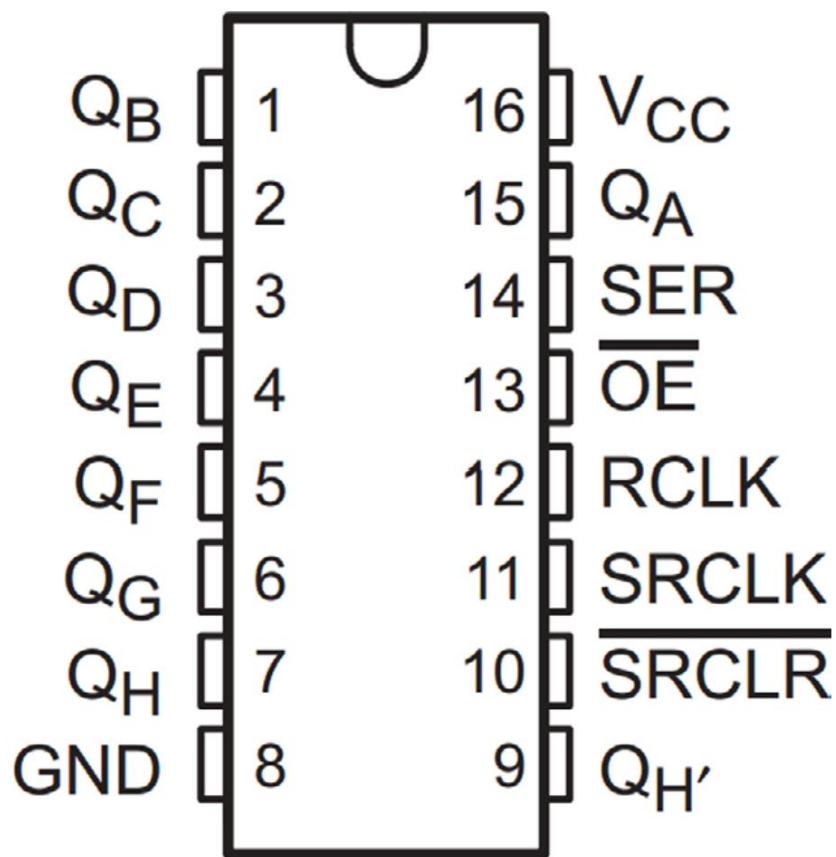


Figure 7-2: Shift register pin-out diagram

На этом уроке мы разберем работу сдвигового регистра 74HC595, который может хранить 8 бит данных.

Схема выводов микросхемы представлена на картинке.

Кстати, у этого регистра есть и отечественный аналог КР1564ИР52

Создание световых эффектов с помощью сдвигового регистра

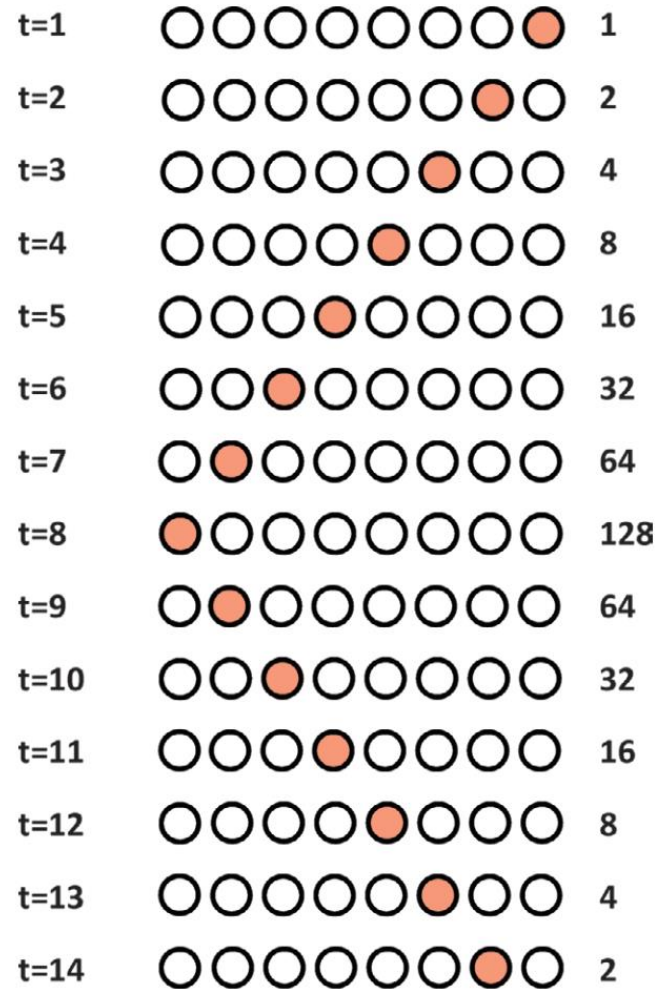


Figure 7-6: Light rider animation steps

«Бегущий всадник»

Создание световых эффектов с помощью сдвигового регистра

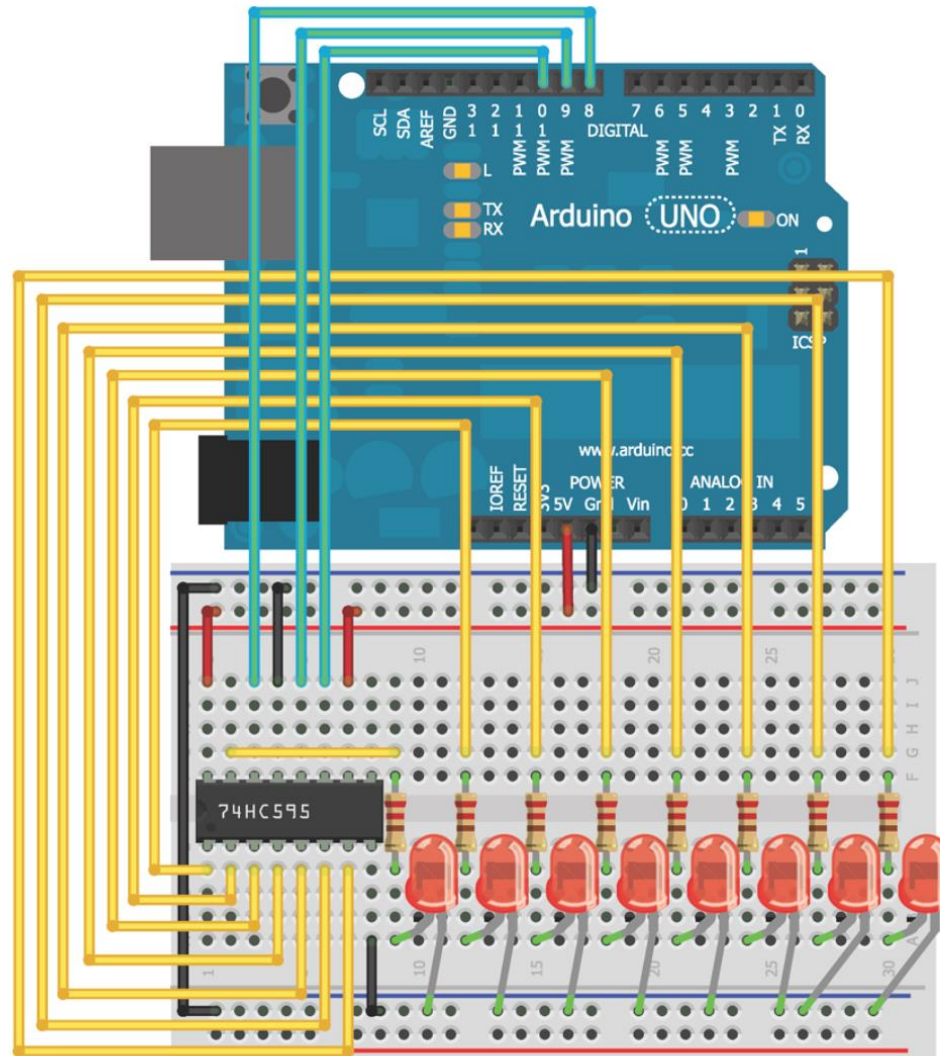


Figure 7-4: Eight LED shift register circuit diagram

Image created with Fritzing.

Передача данных из Arduino в сдвиговый регистр

Теперь можно написать программу для передачи данных из Arduino в сдвиговые регистры. Воспользуемся встроенной в Arduino IDE функцией `shiftOut()` для поразрядной выдачи данных на контакт платы Arduino. Эта функция принимает четыре аргумента:

- ◆ номер контакта DATA;
- ◆ номер контакта CLOCK;
- ◆ порядок выдачи битов;
- ◆ значение, выдаваемое на выход.

```
shiftOut(DATA, CLOCK, MSBFIRST, B10101010);
```

Программный код

```
// Создание световой анимации "бегущий всадник"
const int SER=8;      // Контакт для подключения вывода DATA
const int LATCH =9;   // Контакт для подключения вывода LATCH
const int CLK  =10;    // Контакт для подключения вывода CLOCK

// Последовательность включения светодиодов
int seq[14] = {1,2,4,8,16,32,64,128,64,32,16,8,4,2};

void setup()
{
    // Установить контакты на вывод (OUTPUT)
    pinMode(SER, OUTPUT);
    pinMode(LATCH, OUTPUT);
    pinMode(CLK, OUTPUT);
}

void loop()
{
    for (int i = 0; i < 14; i++)
    {
        digitalWrite(LATCH, LOW);           // LATCH - низкий
        shiftOut(SER, CLK, MSBFIRST, seq[i]); // Старший бит - первый
        digitalWrite(LATCH, HIGH);          // LATCH - высокий
        delay(100);                          // Скорость анимации
    }
}
```

Преобразование между двоичным и десятичным форматами

1	0	1	0	1	0	1	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

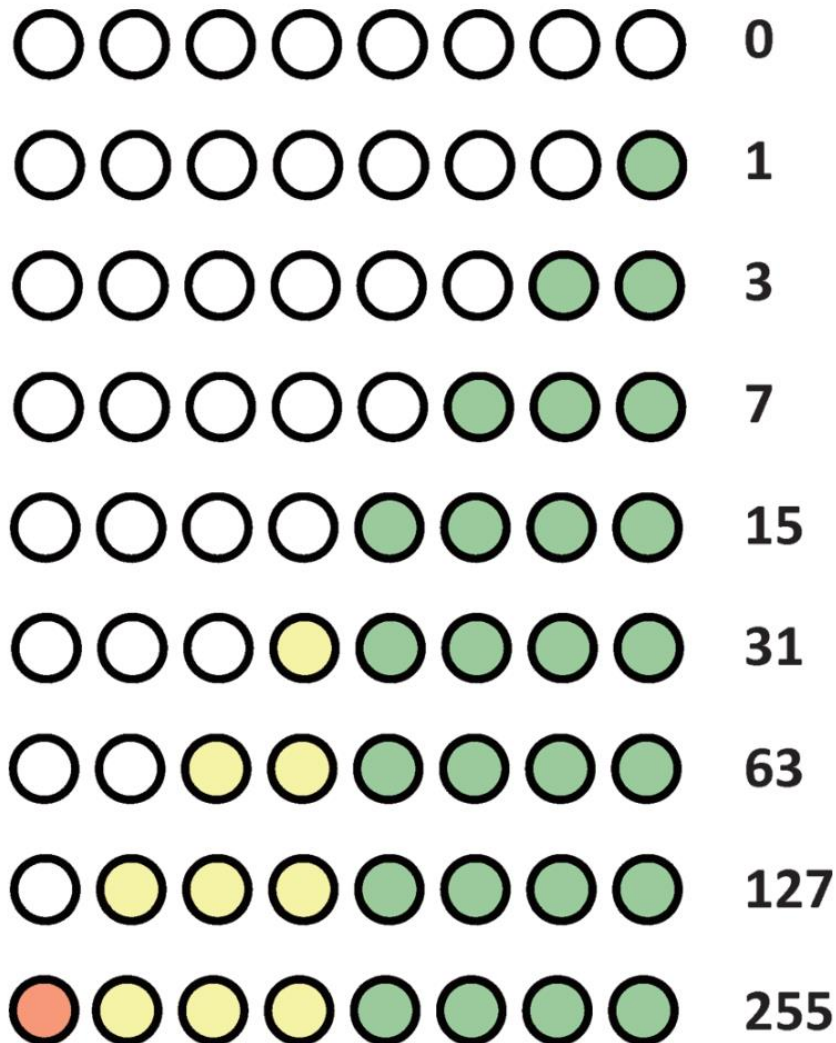
$$1 \times 128 + 0 \times 64 + 1 \times 32 + 0 \times 16 + 1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1 = 170$$

Figure 7-5: Binary to decimal conversion

Двоичное значение каждого бита представляет собой увеличивающуюся степень числа 2.

В нашем случае биты 7, 5, 3, 1 установлены в единицу. Таким образом, чтобы найти десятичный эквивалент, вы складываете 2^7 , 2^5 , 2^3 и 2^1 . Полученное десятичное значение равно 170.

Создание эффекта шкалы индикации расстояния с помощью сдвигового регистра



«Схема индикация
расстояния»

Figure 7-8: Bar graph decimal representations

Создание эффекта шкалы индикации расстояния с помощью сдвигового регистра

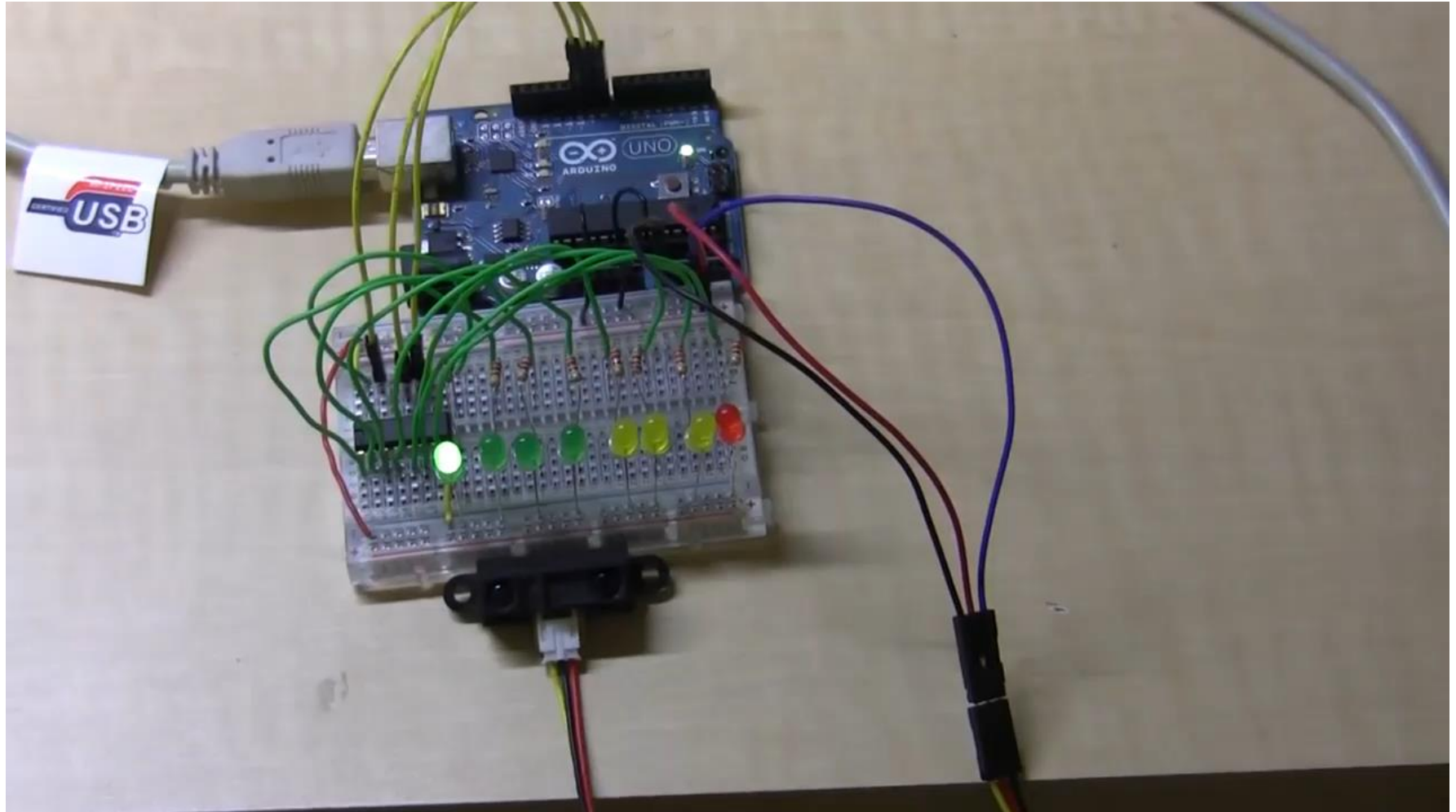


Схема индикации расстояния (парктроник)

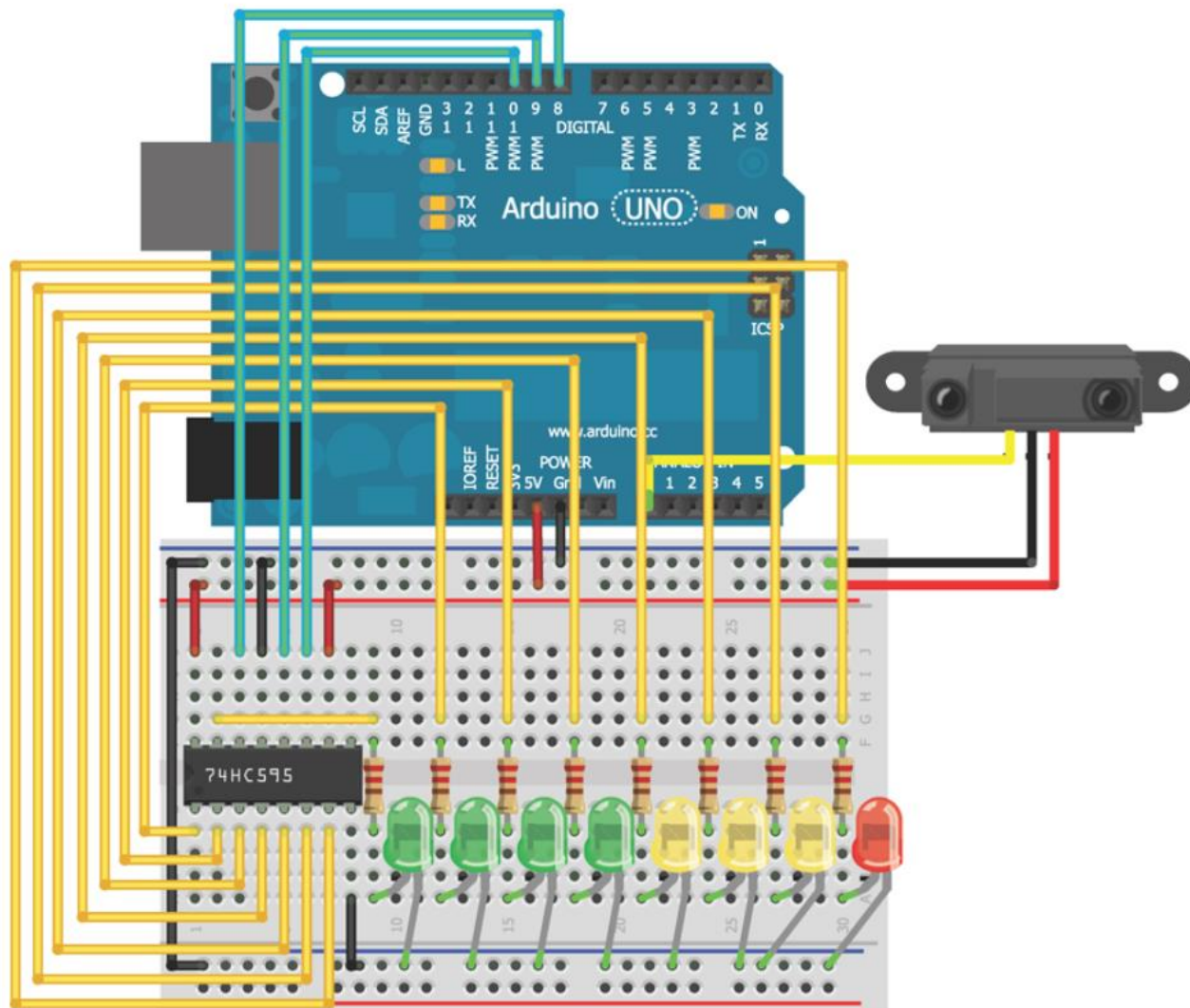


Figure 7-7: Distance-responsive bar graph

Передача данных из Arduino в сдвиговый регистр

Теперь можно написать программу для передачи данных из Arduino в сдвиговые регистры. Воспользуемся встроенной в Arduino IDE функцией `shiftOut()` для поразрядной выдачи данных на контакт платы Arduino. Эта функция принимает четыре аргумента:

- ◆ номер контакта DATA;
- ◆ номер контакта CLOCK;
- ◆ порядок выдачи битов;
- ◆ значение, выдаваемое на выход.

```
shiftOut(DATA, CLOCK, MSBFIRST, B10101010);
```

Программный код

```
//A bar graph that responds to how close you are

const int SER  =8;    //Serial output to shift register
const int LATCH =9;    //Shift register latch pin
const int CLK   =10;   //Shift register clock pin
const int DIST  =0;    //Distance sensor on analog pin 0

//Possible LED settings
int vals[9] = {0,1,3,7,15,31,63,127,255};

//Maximum value provided by sensor
int maxVal = 500;

//Minimum value provided by sensor
int minVal = 0;

void setup()
{
    //Set pins as outputs
    pinMode(SER, OUTPUT);
    pinMode(LATCH, OUTPUT);
    pinMode(CLK, OUTPUT);
}

void loop()
{
    int distance = analogRead(DIST);
    distance = map(distance, minVal, maxVal, 0, 8);
    distance = constrain(distance,0,8);

    digitalWrite(LATCH, LOW);           //Latch low - start sending
    shiftOut(SER, CLK, MSBFIRST, vals[distance]); //Send data, MSB first
    digitalWrite(LATCH, HIGH);          //Latch high - stop sending
    delay(10);                          //Animation speed
}
```