

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ  
УНИВЕРСИТЕТ»

Кафедра ИиСП

Отчет  
по лабораторной работе № 5  
по дисциплине «Машинно-зависимые языки программирования»  
Вариант 1

Выполнил: ст. гр. ПС-11

Маркин И. А.

Проверил: доцент, доцент  
кафедры ИиСП Баев А.А.

г. Йошкар-Ола  
2025

Цель работы:

- 1)Разобраться с заданием, которое дано в методичке
- 2)Установить Proteus и разобраться в нем
- 3)Научиться писать код для него
- 4)Собрать схемы для ATmega328P

Задания на лабораторную работу:

- 1)Написать коды и сделать схемы для следующих заданий:  
Подключить кнопку, Подключить 7-сегментный индикатор,  
Подключить кнопку, Сделать секундомер)
- 2)Оптимизация кода
- 3)Задание по вариантам(Секундомер до 99)
- 4)Задание от преподавателя
- 5)Сделать макет схем

Теоретические сведения:

ПРИМЕНЕНИЕ МИКРОКОНТРОЛЛЕРОВ В РАДИОТЕХНИЧЕСКИХ И  
БИОМЕДИЦИНСКИХ СИСТЕМАХ

Практическая часть:

1) Написать код и сделать схемы для следующих заданий:

Подключить светодиод, Подключить 7-сегментный индикатор и кнопку,

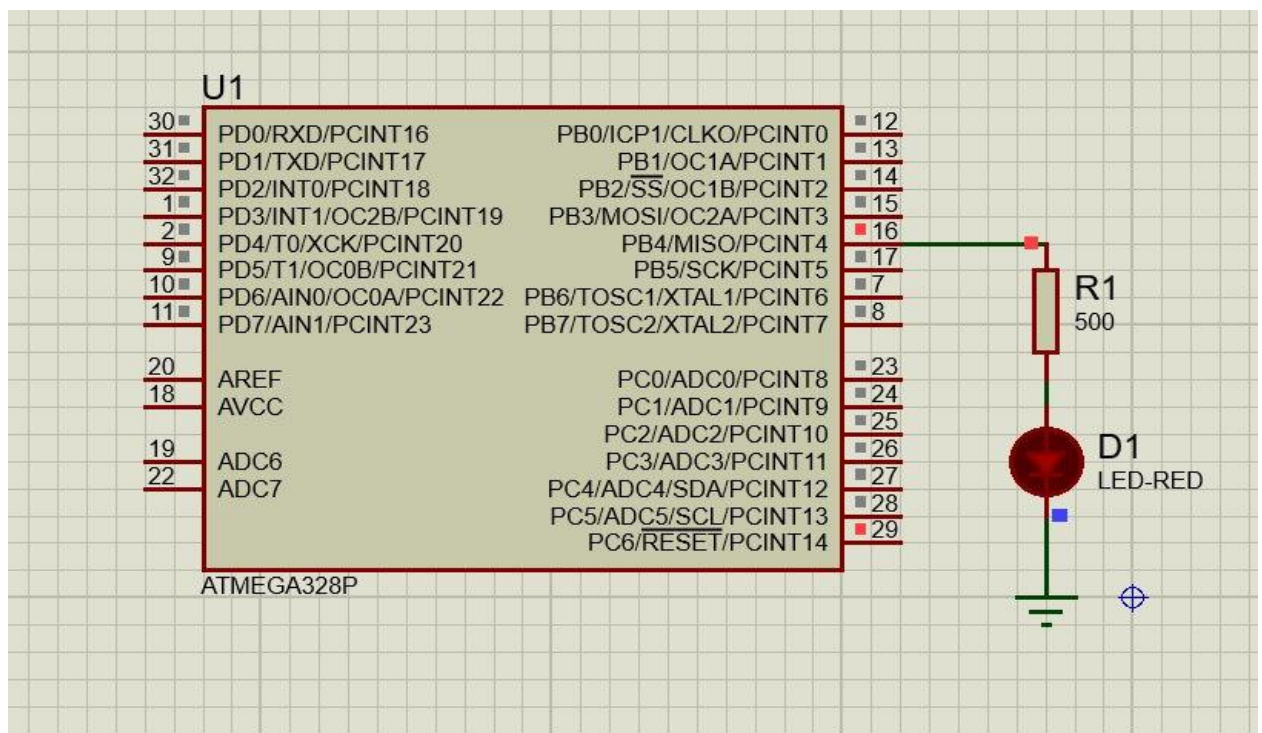
Сделать секундомер, Внешнее прерывание

1. Подключить светодиод

Код на C:

```
#include <avr/io.h>
int main(void)
{
    DDRB = 0b00010000;
    PORTB = 0b00010000;
    while(1){}
}
```

Схема:



2. Подключить 7-сегментный индикатор

Код на C:

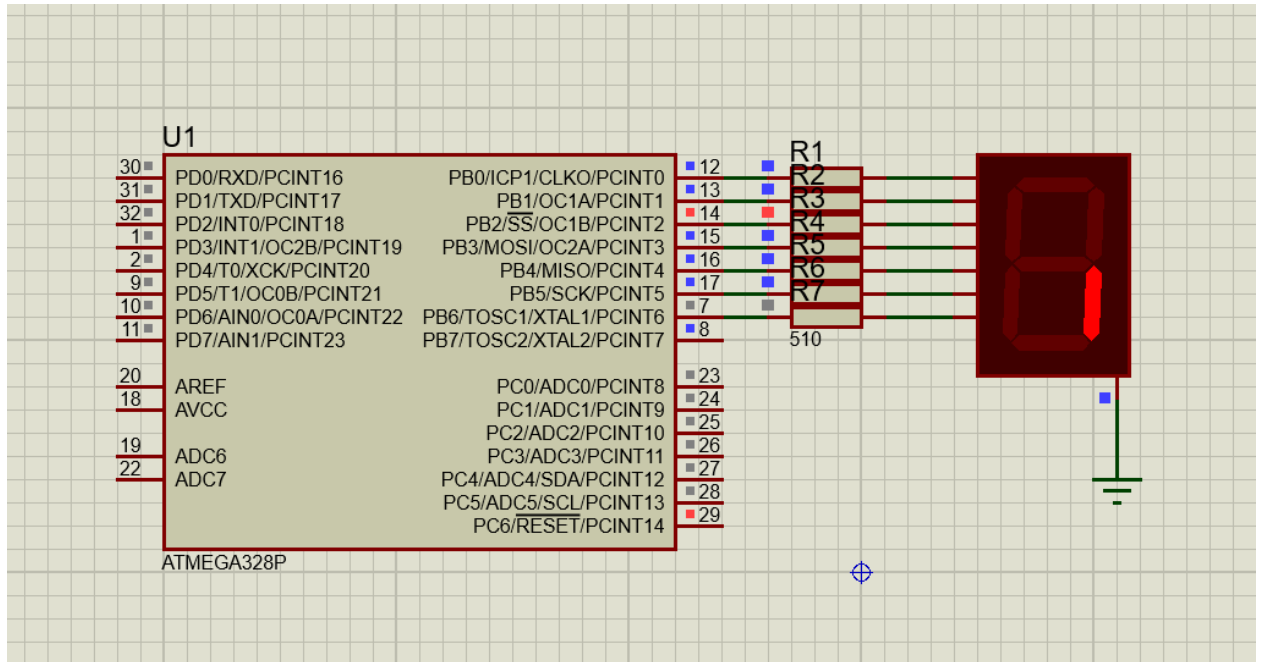
```
#define F_CPU 1000000UL
#include <avr/io.h>
#include <util/delay.h>
int main(void)
{
    DDRB = 0xFF;
    while(1)
```

```

{
    for(int i = 0; i < 6; i++)
    {
        PORTB = (1 << i);
        _delay_ms(200);
    }
}

```

**Схема:**



### 3. Подключение кнопки к 7-сегментному индикатору

Код на C:

```

#define F_CPU 1000000UL
#include <avr/io.h>
#include <util/delay.h>
int main(void)
{
    DDRB = 0xFF & ~(1 << PINB7);
    PORTB |= (1 << PINB7);
    int button = 0;
    while(1)
    {
        for(int i = 0; i < 6; i++)
        {
            button = PINB & (1 << PINB7);
            if(button != 0)
            {
                PORTB = (1 << i);
                _delay_ms(200);
            }
            else
            {

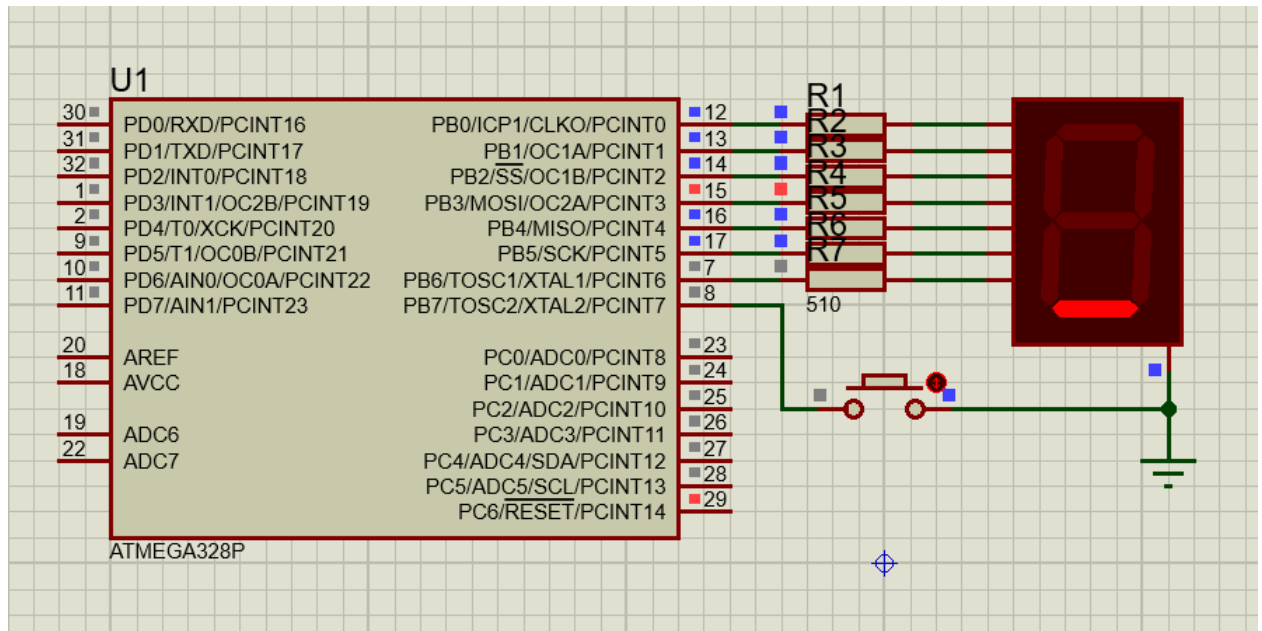
```

```

        PORTB = (0x20 >> i);
        _delay_ms(200);
    }
}
}

```

Схема:



#### 4. Секундомер

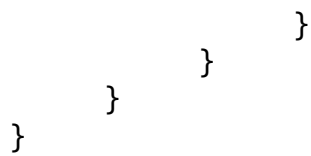
Код на C:

```

#define F_CPU 1000000UL
#include <avr/io.h>
#include <util/delay.h>
int main(void)
{
    DDRB = 0xFF & ~(1 << PINB7);
    PORTB |= (1 << PINB7);
    int button = 0;
    while(1)
    {
        for(int i = 0; i < 6; i++)
        {
            button = PINB & (1 << PINB7);
            if(button != 0)
            {
                PORTB = (1 << i);
                _delay_ms(200);
            }
            else
            {
                PORTB = (0x20 >> i);
                _delay_ms(200);
            }
        }
    }
}

```

}



## 5. Внешнее прерывание

Код на С:

```
#define F_CPU 1000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
uint8_t segments[] =
{
    0b00111111,
    0b00000110,
    0b01011011,
    0b01001111,
    0b01100110,
    0b01101101,
    0b01111101,
    0b00000111,
    0b01111111,
    0b01101111,
};

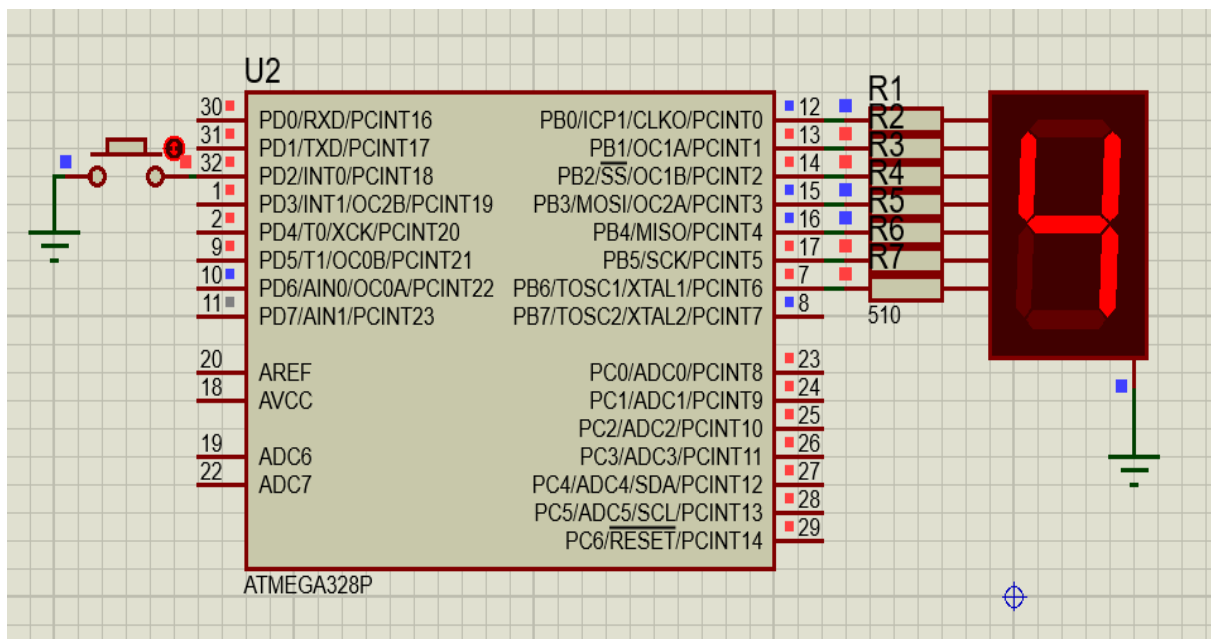
volatile int switch_state = 0;
volatile int counter = 0;
ISR(INT0_vect)
{
    if(switch_state == 0)
    {
        switch_state = 1;
    }
}
```

```

else
{
    switch_state = 0;
    counter = 0;
}
}
int main(void)
{
    DDRB = 0xFF;
    PORTD |= (1 << PIND2);
    EIMSK |= (1 << INT0);
    EICRA |= (1 << ISC01);
    sei();
    while(1)
    {
        if(switch_state == 0)
        {
            if(counter < 10)
            {
                PORTB = segments[counter];
                counter += 1;
                _delay_ms(500);
            }
            else
            {
                counter = 0;
                PORTB = segments[counter];
                counter += 1;
                _delay_ms(500);
            }
        }
    }
}

```

Cхема:



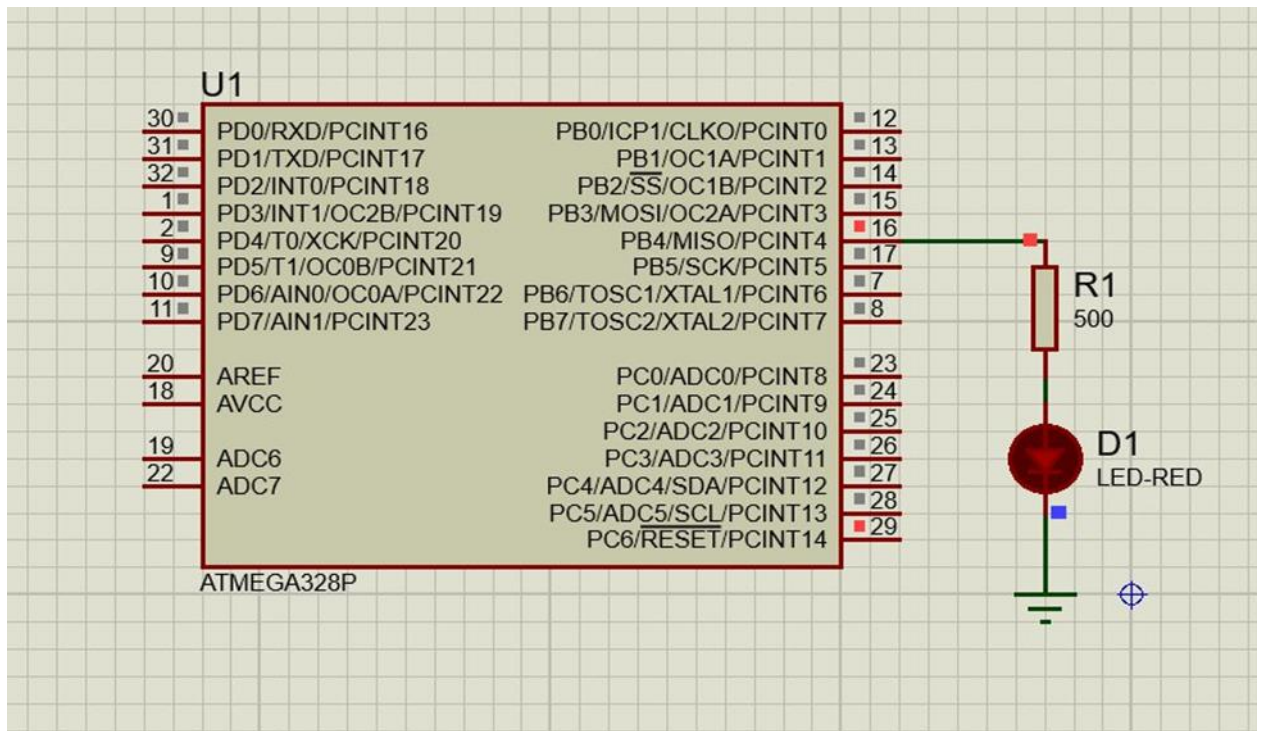
## 2) Оптимизация кода

### 1. Подключить светодиод

Код на C:

```
#include <avr/io.h>
int main(void)
{
    DDRB = 0b00010000;
    PORTB = 0b00010000;
    while(1){}
}
```

Схема:



### 2. Подключить 7-сегментный индикатор

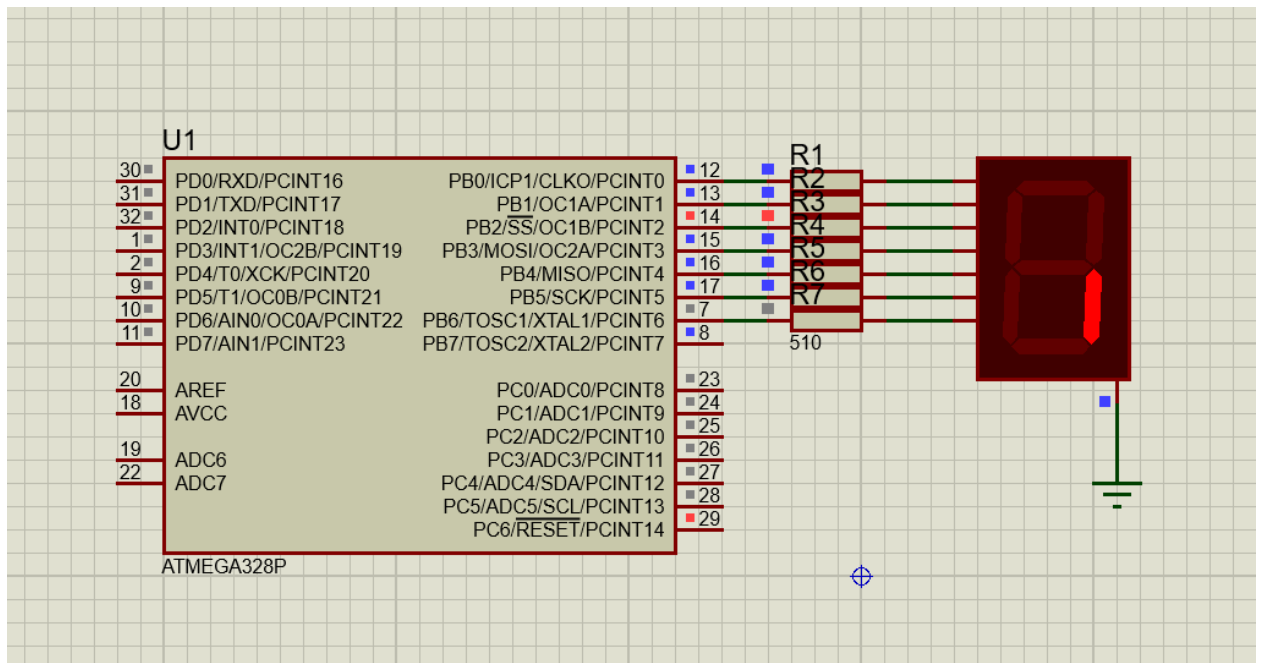
Код на C:

```
#define F_CPU 1000000UL
#include <avr/io.h>
#include <util/delay.h>
int main(void)
{
    DDRB = 0xFF;
    while(1)
    {
        for(int i = 0; i < 6; i++)
        {
            PORTB = (1 << i);
            _delay_ms(200);
        }
    }
}
```



}

Схема:

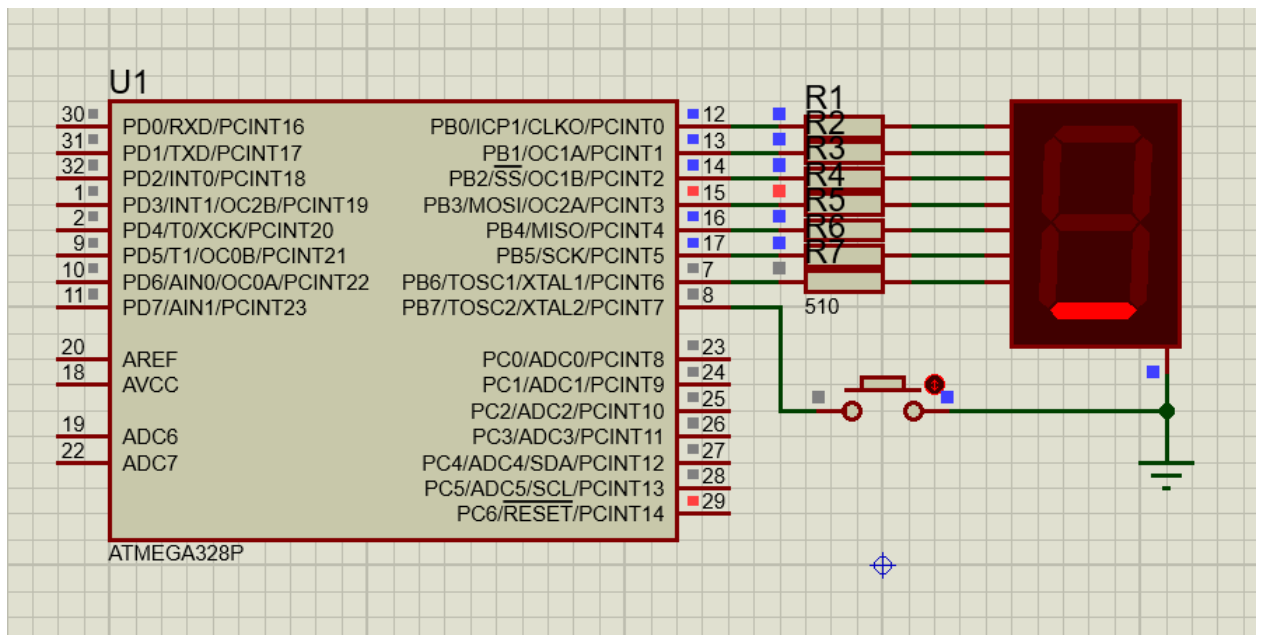


### 3. Подключение кнопки к 7-сегментному индикатору

Код на C:

```
#define F_CPU 1000000UL
#include <avr/io.h>
#include <util/delay.h>
int main(void)
{
    DDRB = 0xFF & ~(1 << PINB7);
    PORTB |= (1 << PINB7);
    while(1)
    {
        for (int i = 0; i < 6; i++)
        {
            if (PINB & (1 << PINB7) != 0)
            {
                PORTB = (1 << i);
            }
            else
            {
                PORTB = (0x20 >> i);
            }
            _delay_ms(200);
        }
    }
}
```

Схема:



#### 4. Секундомер

Код на C:

```
#define F_CPU 1000000UL
#include <avr/io.h>
#include <util/delay.h>
uint8_t segments[] =
{
    0b00111111,
    0b00000110,
    0b01011011,
    0b01001111,
    0b01100110,
    0b01101101,
    0b01111101,
    0b00000111,
    0b01111111,
    0b01101111,
};

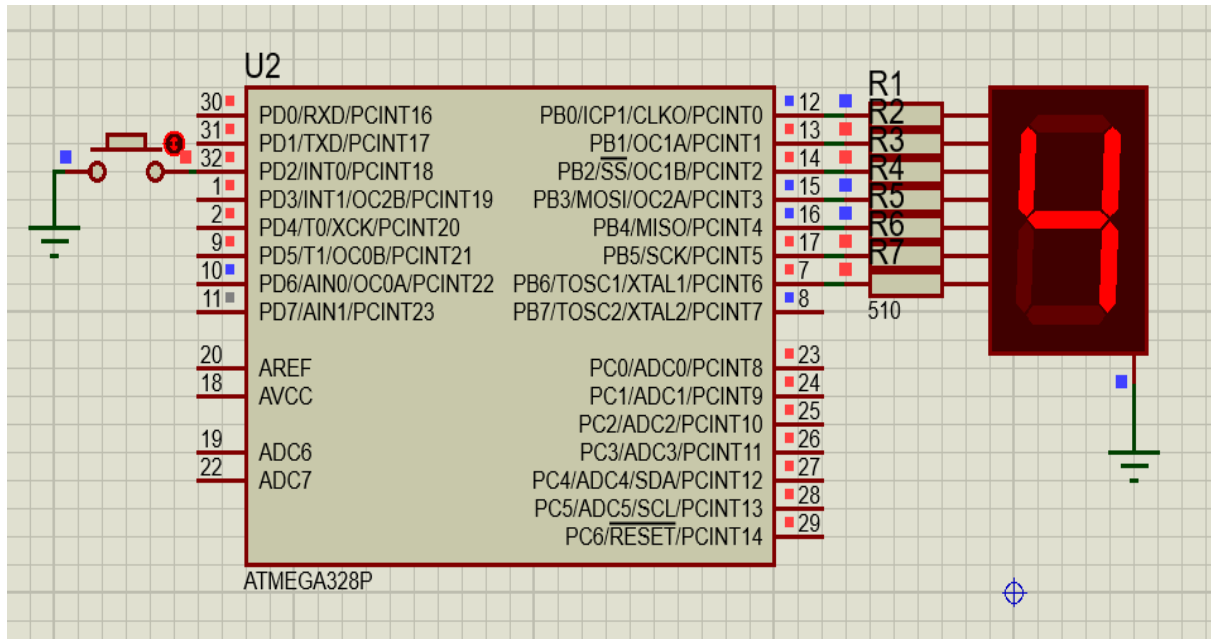
int main(void)
{
    DDRB = 0xFF;
    DDRD &= ~(1 << PIND2);
    PORTD |= (1 << PIND2);
    int counter = 0;
    int switch_state = 0;
    while(1)
    {
        if (!(PIND & (1 << PIND2)))
        {
            while (!(PIND & (1 << PIND2))) {}
            switch_state ^= 1;
            if (!switch_state) counter = 0;
        }
    }
}
```

```

    if (!switch_state)
    {
        PORTB = segments[counter];
        _delay_ms(1000);
        counter = (counter + 1) % 10;
    }
}

```

Схема:



## 5. Внешнее прерывание

Код на C:

```

#define F_CPU 1000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
uint8_t segments[] =
{
    0b00111111,
    0b00000110,
    0b01011011,
    0b01001111,
    0b01100110,
    0b01101101,
    0b01111101,
    0b00000111,
    0b01111111,
    0b01101111,
};
volatile int switch_state = 0;
volatile int counter = 0;
ISR(INT0_vect)

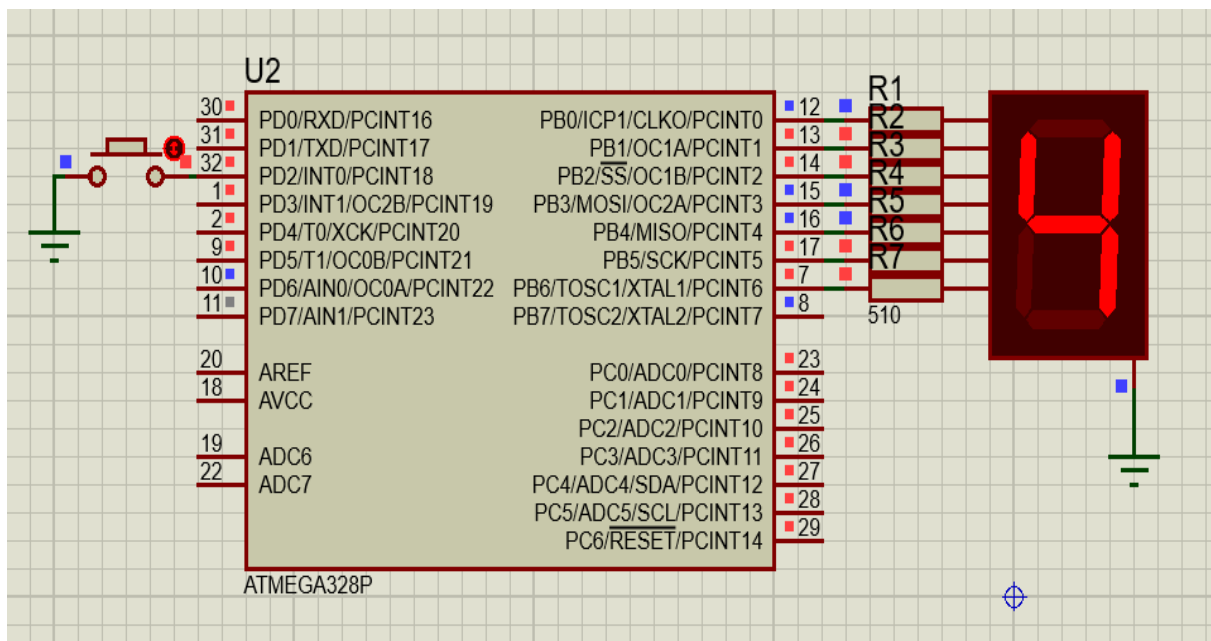
```

```

{
    switch_state ^= 1;
    if (!switch_state) counter = 0;
}
int main(void)
{
    DDRB = 0xFF;
    PORTD |= (1 << PIND2);
    EIMSK |= (1 << INT0);
    EICRA |= (1 << ISC01);
    sei();
    while(1)
    {
        if (!switch_state)
        {
            PORTB = segments[counter];
            counter = (counter + 1) % 10;
            _delay_ms(500);
        }
    }
}

```

Схема:



### 3)Задание по вариантам

Секундомер до 99(Дополнительное задание из методички)

Код на C:

```

#define F_CPU 1000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
uint8_t segments[] =
{

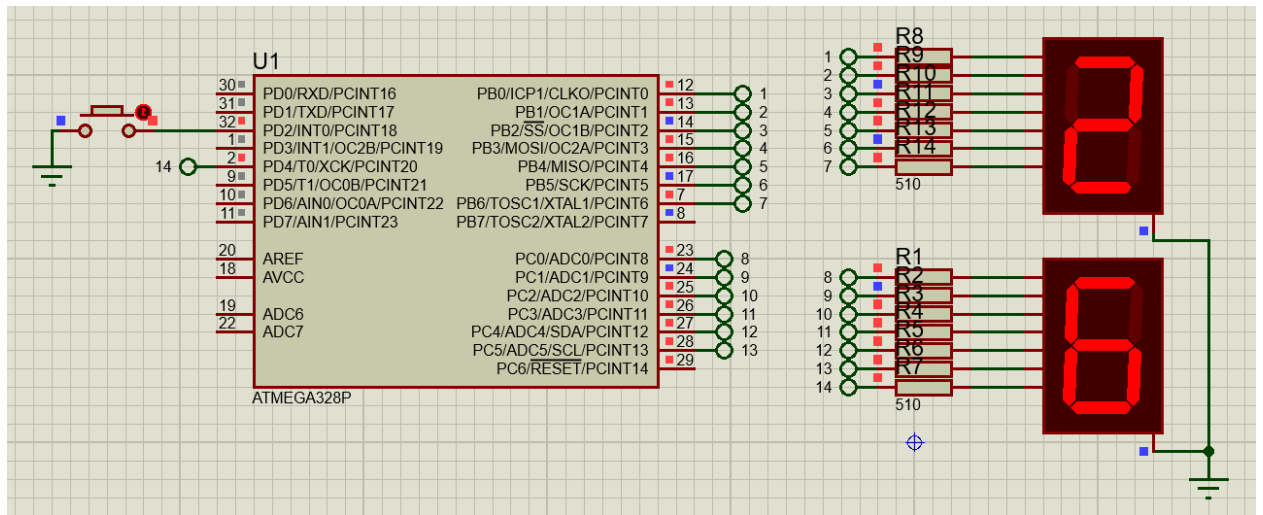
```

```

        0b00111111,
        0b00000110,
        0b01011011,
        0b01001111,
        0b01100110,
        0b01101101,
        0b01111101,
        0b00000111,
        0b01111111,
        0b01101111
    };
    volatile uint8_t counter = 0;
    ISR(INT0_vect)
    {
        counter++;
        if (counter > 99)
        {
            counter = 0;
        }
    }
    int main(void)
    {
        DDRB = 0xFF;
        DDRC = 0xFF;
        DDRD |= (1 << PIND4);
        PORTD |= (1 << PIND2);
        EIMSK |= (1 << INT0);
        EICRA |= (1 << ISC01);
        sei();
        while (1)
        {
            PORTB = segments[counter / 10];
            PORTC = segments[counter % 10];
            if (segments[counter] & (1 << 6))
            {
                PORTD |= (1 << PIND4);
            }
            else
            {
                PORTD &= ~(1 << PIND4);
            }
            counter = (counter + 1) % 100;
            _delay_ms(1000);
        }
    }

```

Cхема:



#### 4)Задание от преподавателя

Вариант 20, Форма 3 – ёлка, Эфффекты – 0, 7, 6.

Код на C:

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
volatile uint32_t leds_status = 0;
uint8_t step_counter = 0;
uint8_t current_effect = 0;
uint8_t move_direction = 0;
const uint8_t leds_count[] = {1, 2, 6, 10, 15, 19};
ISR(INT0_vect)
{
    current_effect = (current_effect + 1) % 3;
    leds_status = 0;
    step_counter = 0;
    move_direction = 1;
}
void run_effect_0(void)
{
    if (move_direction == 1)
    {
        leds_status |= (1 << step_counter);
        step_counter++;
        if (step_counter >= 19)
        {
            move_direction = 0;
        }
    }
    else
    {
        leds_status &= ~(1 << (step_counter - 1));
        step_counter--;
        if (step_counter == 0)
        {
            move_direction = 1;
        }
    }
}
void run_effect_7(void)
{
    for (uint8_t i = 0; i < leds_count[step_counter]; i++)
    {
        leds_status |= (1 << i);
    }
    step_counter++;
    if (step_counter >= 6)
    {
        step_counter = 0;
        leds_status = 0;
    }
}
void run_effect_6(void)
{
    leds_status ^= (1 << 9);
    if (step_counter < 9)
    {
        leds_status |= (1 << (9 - step_counter));
        leds_status |= (1 << (9 + step_counter));
        step_counter++;
    }
}
```

```

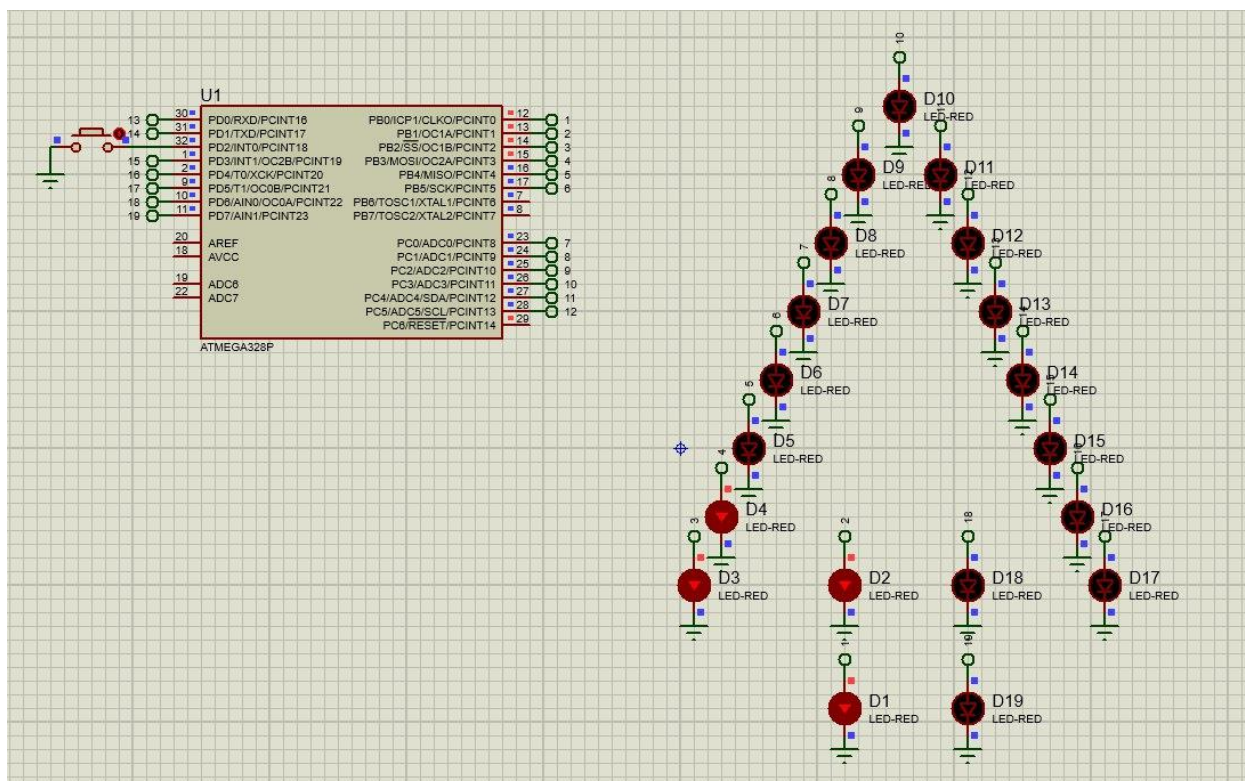
        else
        {
            step_counter = 0;
            leds_status = 0;
        }
    }
}
int main(void)
{
    DDRB = 0xFF;
    DDRC = 0xFF;
    DDRD = 0xFF;
    DDRD &= ~(1 << PIND2);
    PORTD |= (1 << PIND2);
    EIMSK |= (1 << INT0);
    EICRA |= (1 << ISC01);
    sei();
    while(1)
    {
        switch (current_effect)
        {
            case 0:
                run_effect_0();
                break;
            case 1:
                run_effect_7();
                break;
            case 2:
                run_effect_6();
                break;
        }
        PORTB = leds_status & 0x3F;
        PORTC = (leds_status >> 6) & 0x3F;
        PORTD = ((leds_status >> 12) & 0x03) | ((leds_status >> 11) & 0xF8);
        _delay_ms(50);
    }
}

```

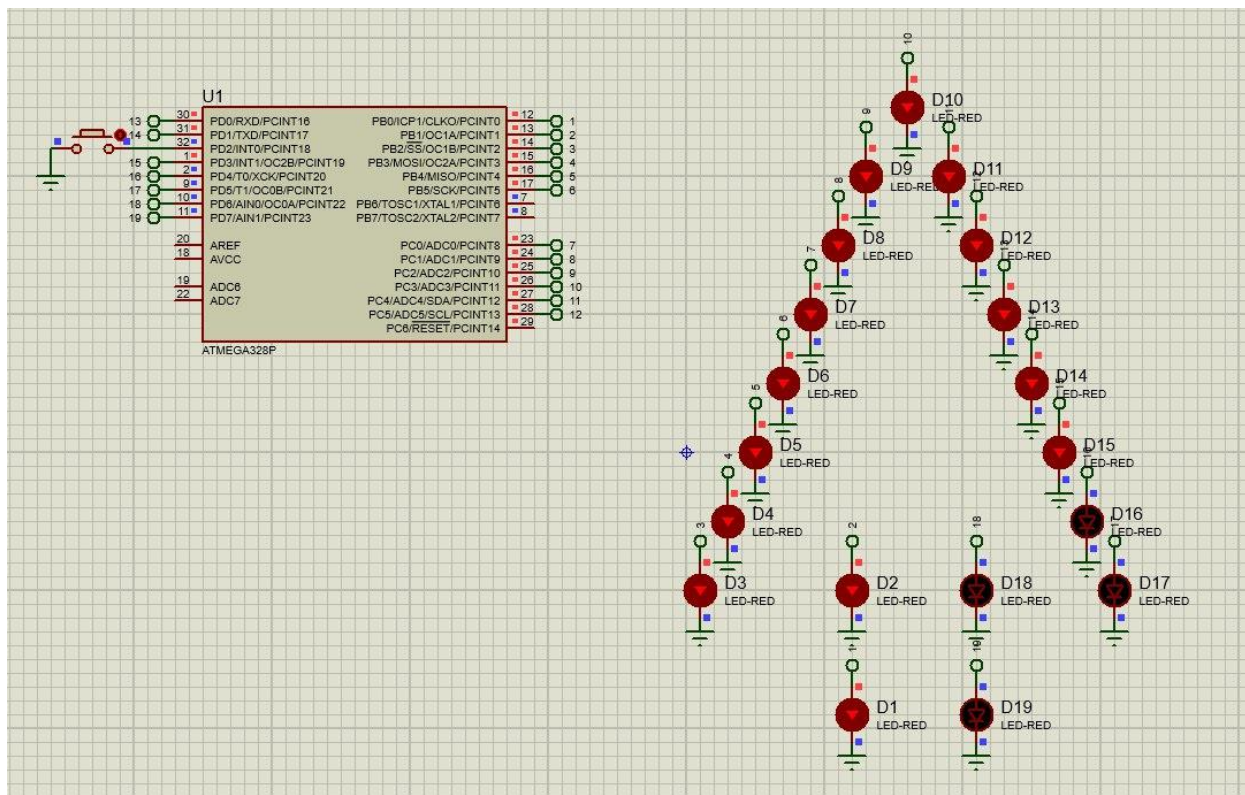
Эффекты:



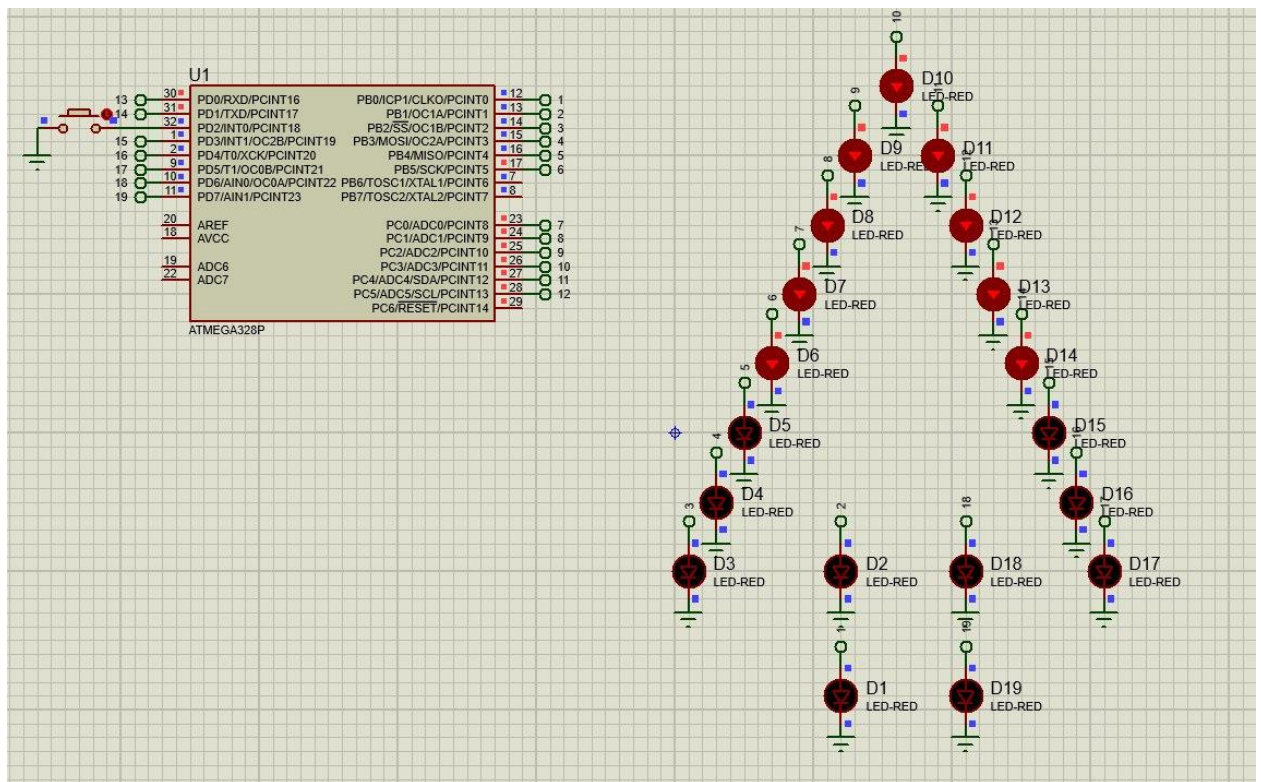
## 1)Эффект 0



## 2)Эффект 7



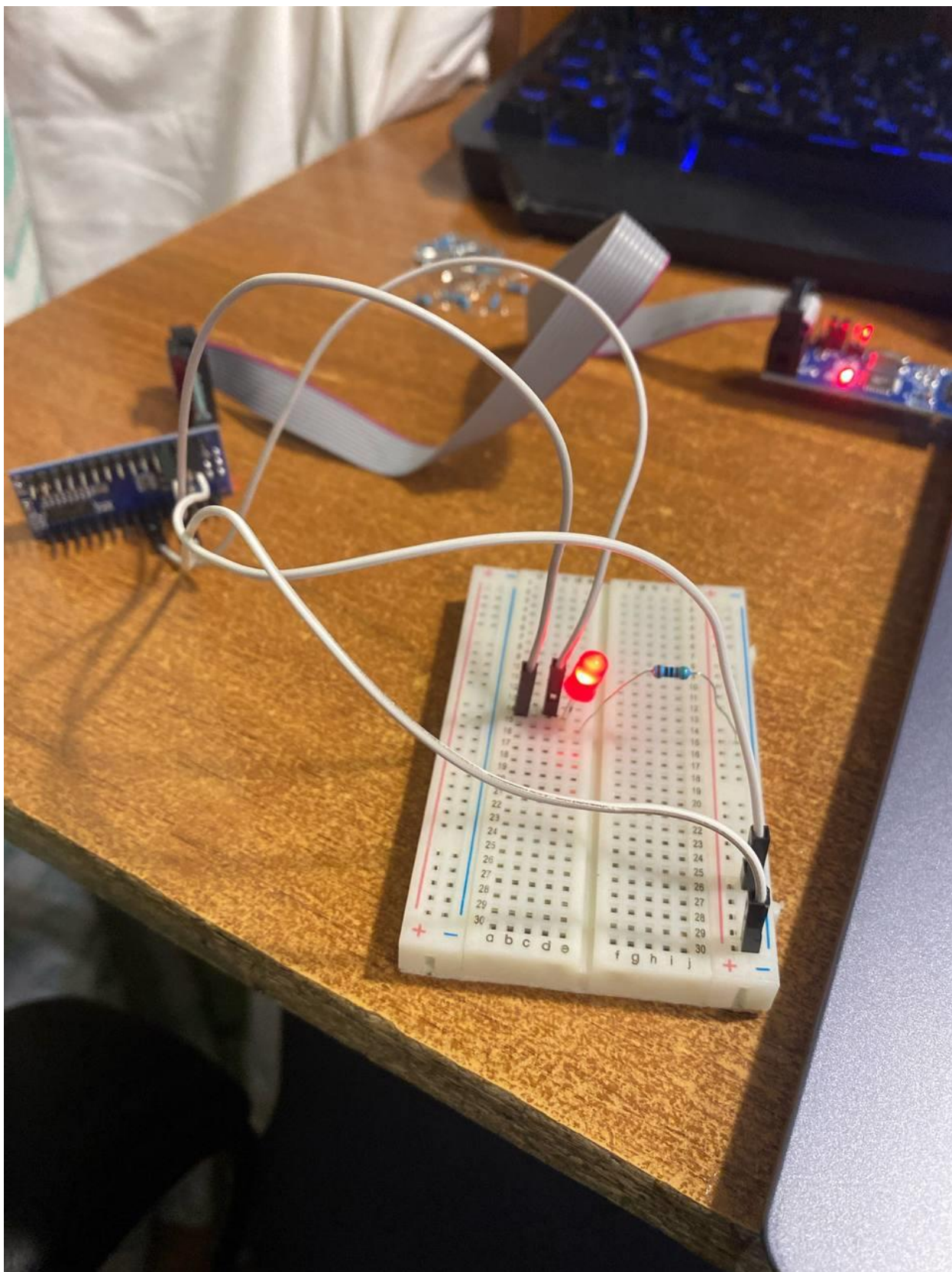
## 3)Эффект 6



## 5.Макет схем

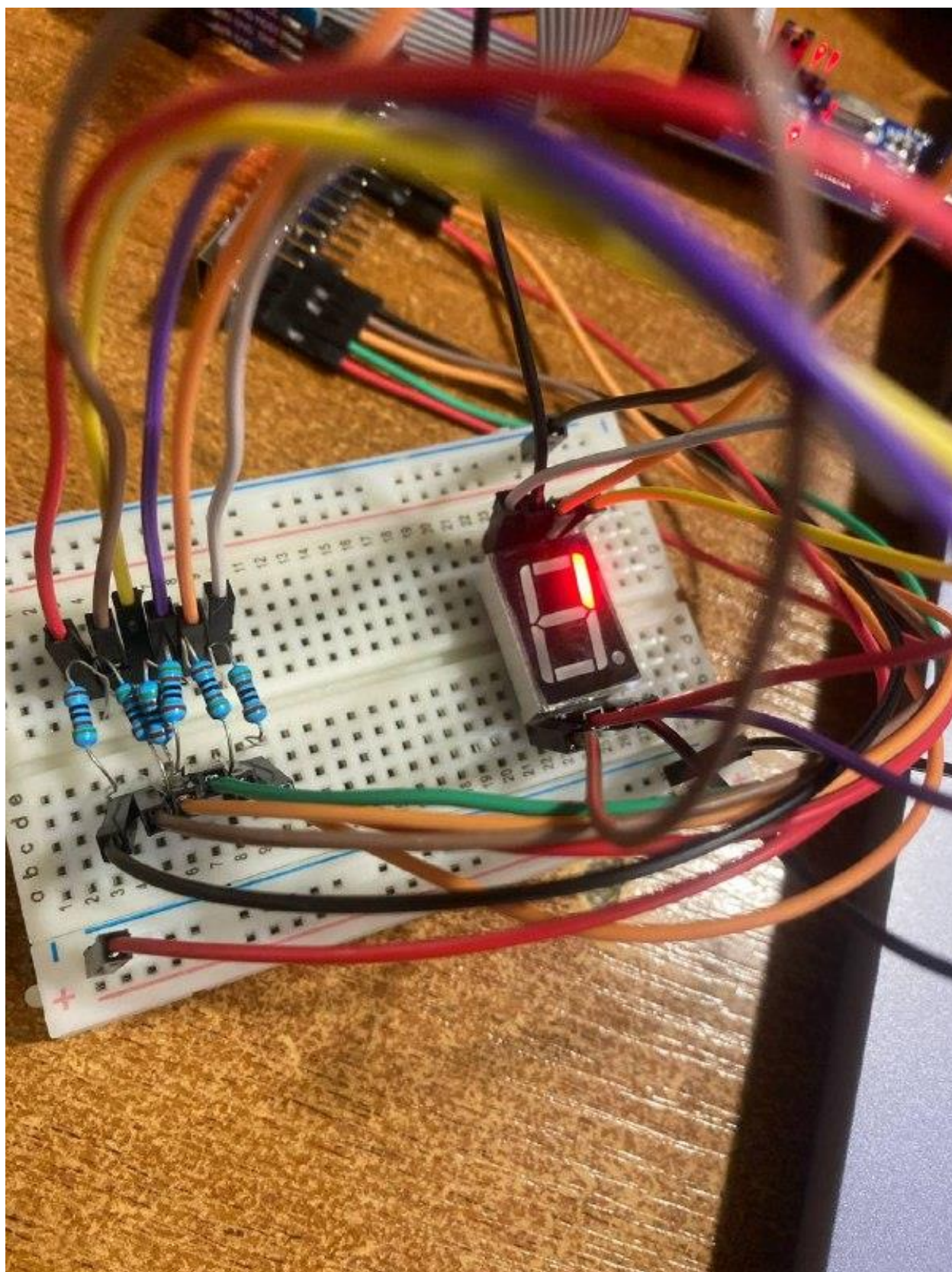
### 1.Подключение светодиода



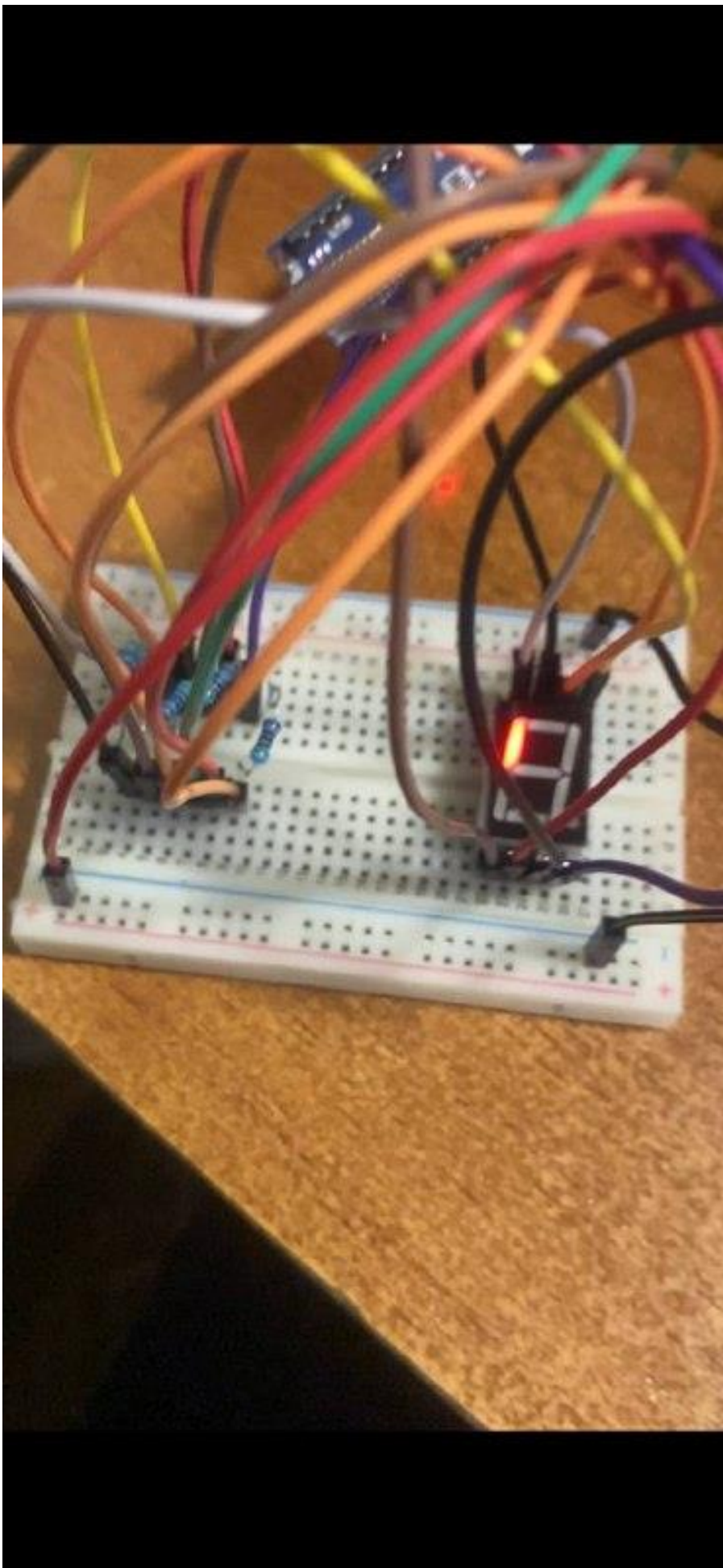


*2.Подключение индикатора*



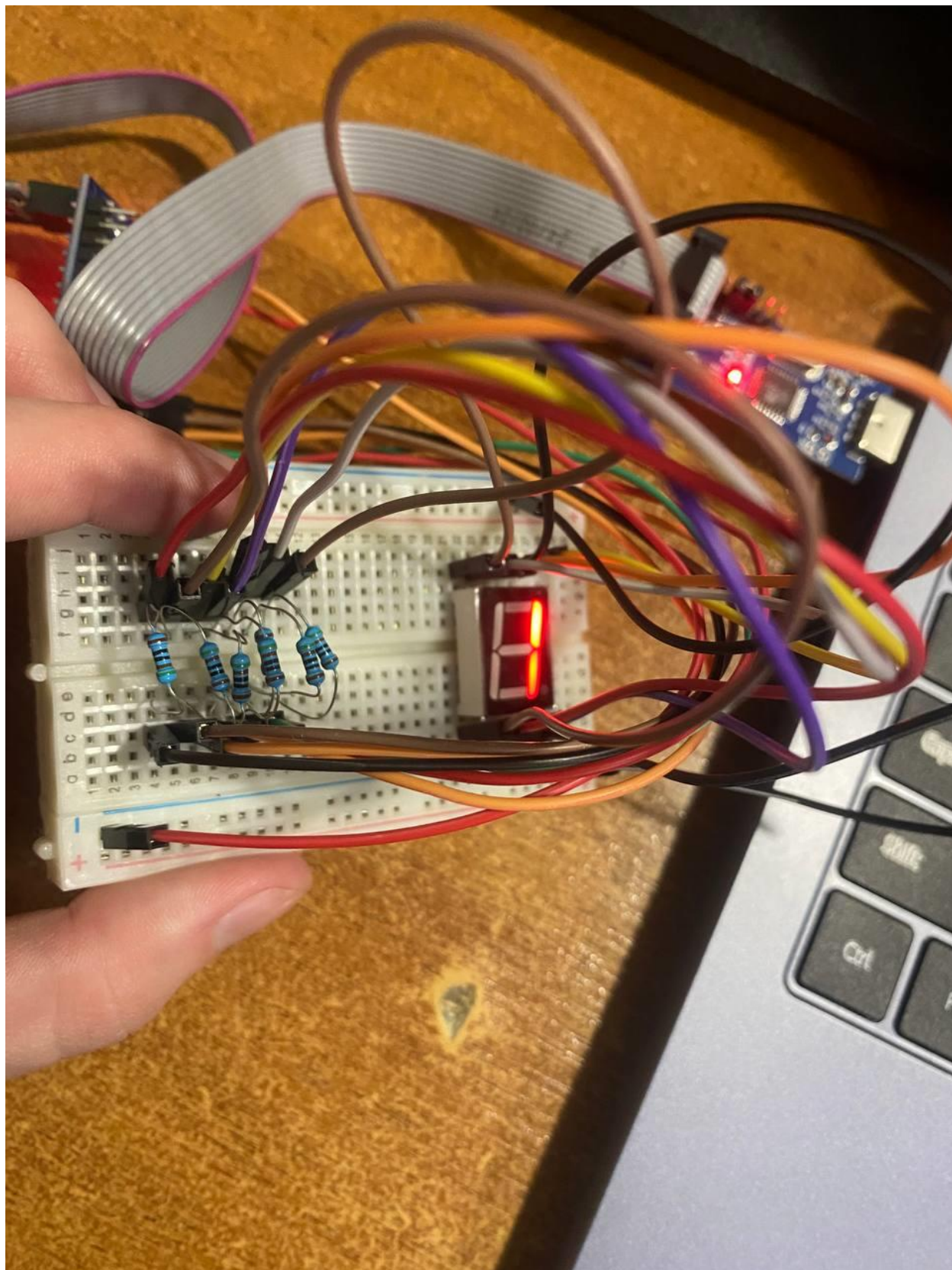


*3.Подлкючение кнопки*



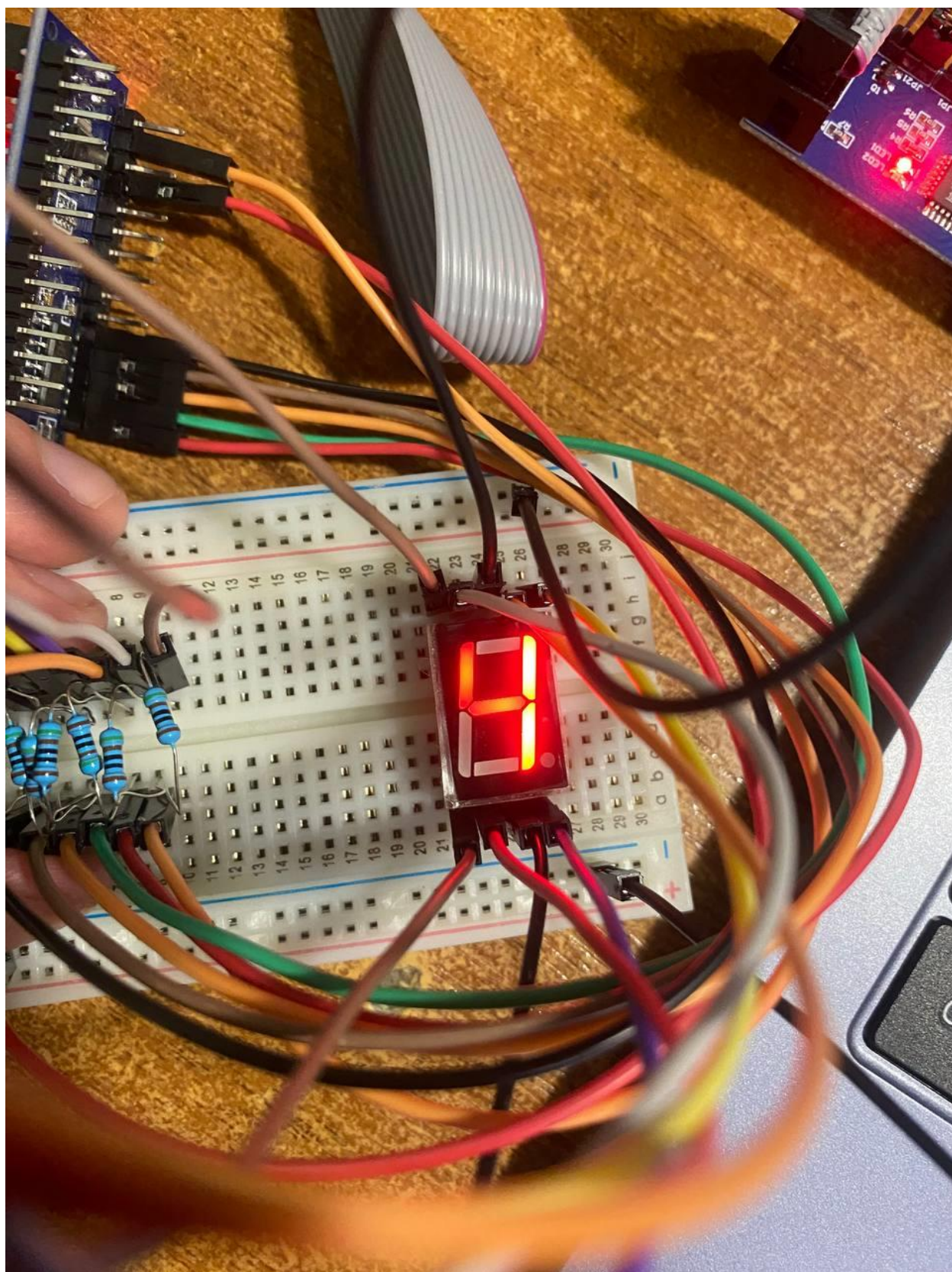


#### 4. Секундомер



#### 5. Внешнее прерывание

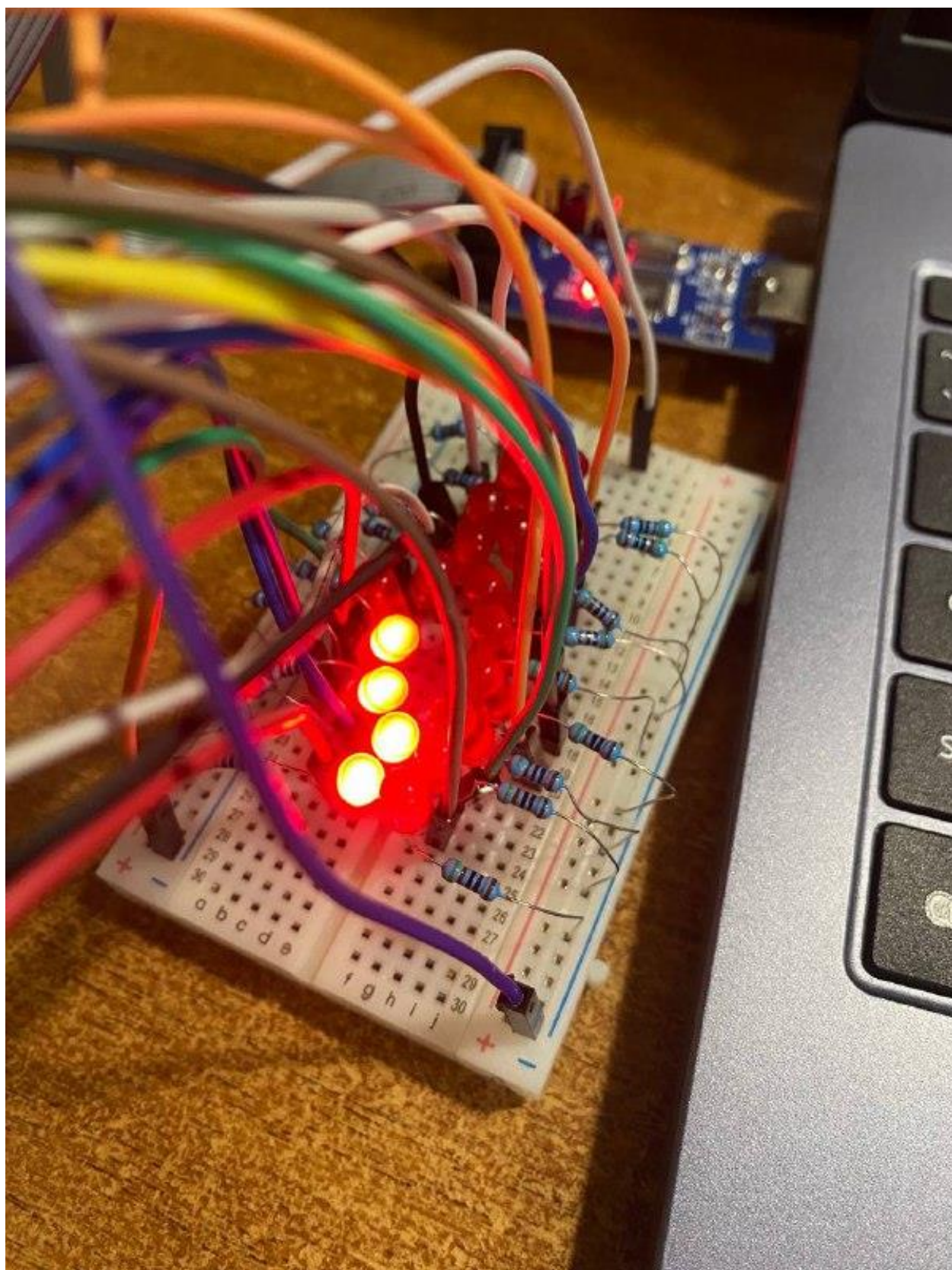




Эффекты:

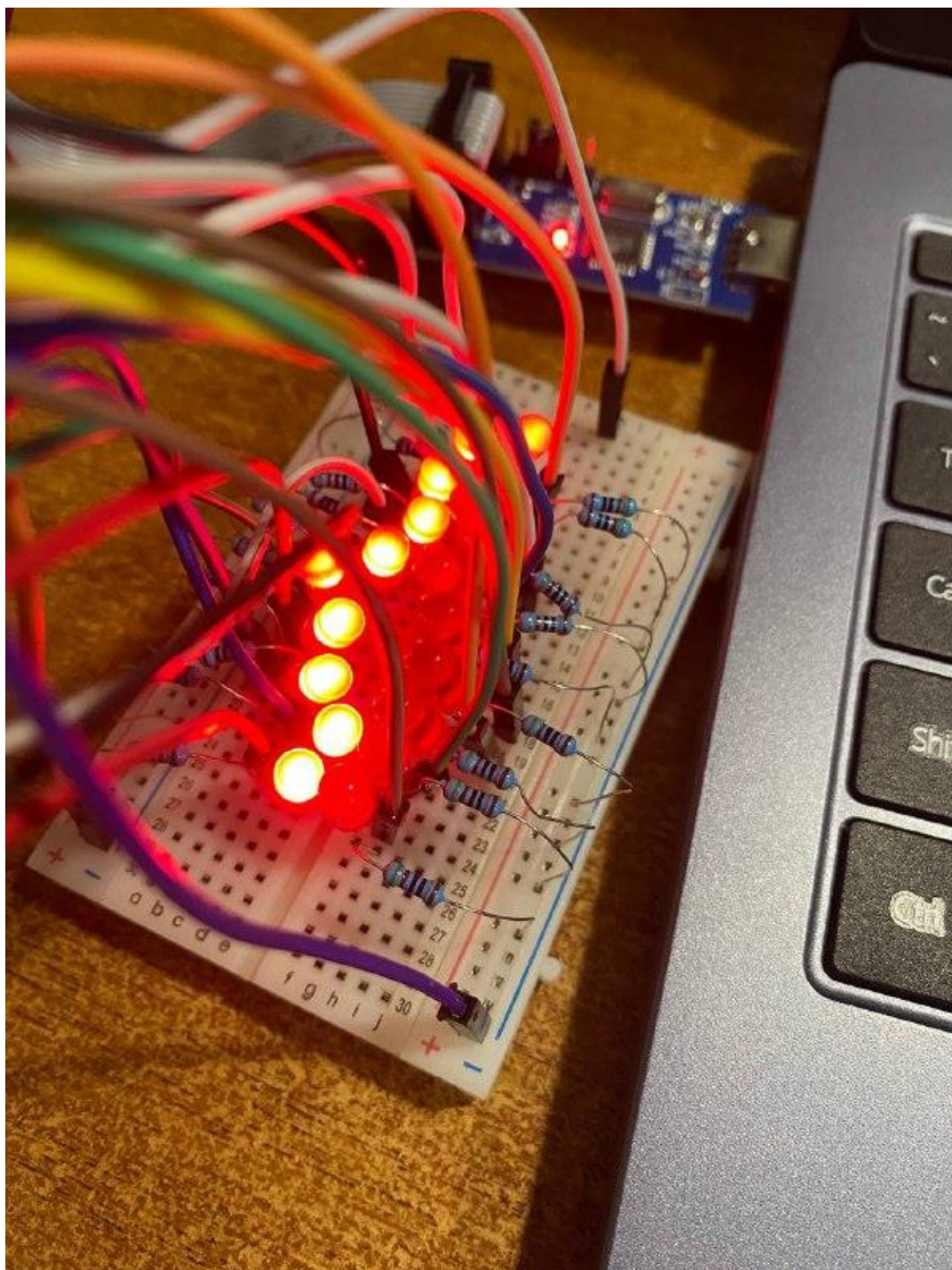
1) Эффект 0





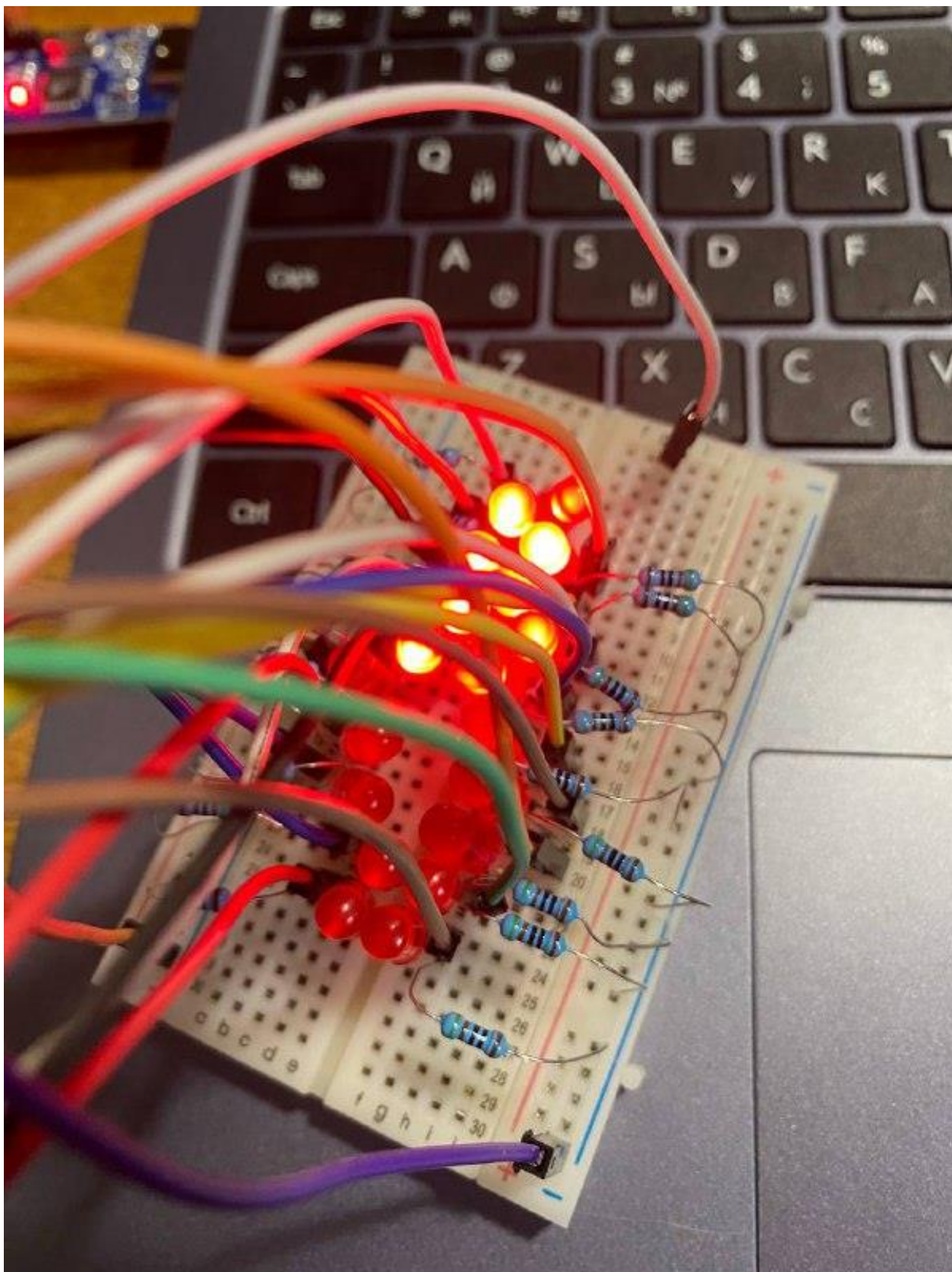
2)Эффект 7





3)Эффект 6





Выводы: В данной лабораторной работе я впервые попробовал сделать схему для кода, который написал и смог реализовать некоторые схемы на практике, а также и в самом железе.