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

Кафедра ИиСП

# Отчет
по лабораторной работе № 6

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)Прочитать 2 раздел по МК

2)Научиться писать код для вывода 2-х и более индикаторов

3)Изучить как работают сдвиговые регистры

**Задания на лабораторную работу:**

1) Написание кода из методички (Реализация динамической индикации, Подключение индикаторов с помощью регистров)

2) Оптимизация основного кода

3) Задание из методички(1 вариант)

4) Задание от преподавателя(Гирлянда)

## 1. Теоретические сведения

ПРИМЕНЕНИЕ МИКРОКОНТРОЛЛЕРОВ В РАДИОТЕХНИЧЕСКИХ И БИОМЕДИЦИНСКИХ СИСТЕМАХ

## 2. Практическая часть

### 1. Написание кода из методички (Реализация динамической индикации, Подключение индикаторов с помощью регистров)

*1.Реализация динамической индикации*

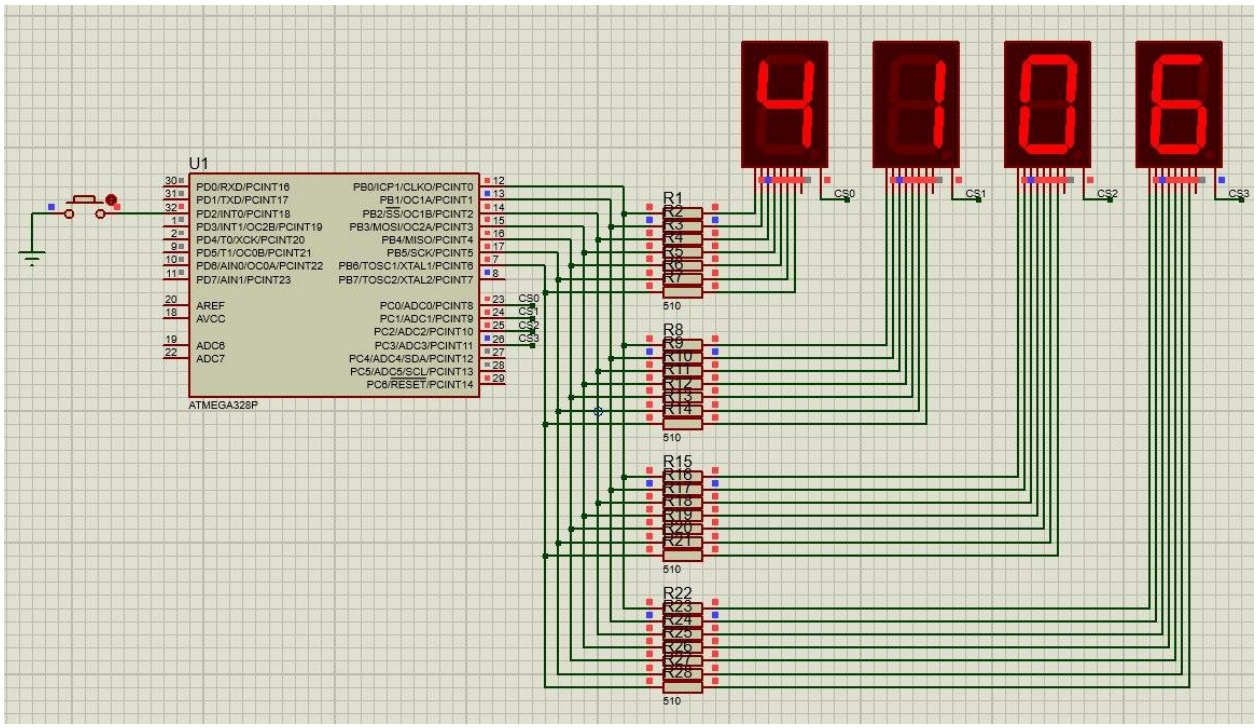
**Код на С:**

```
#define F_CPU 1000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
uint8_t segments[]={
	0b00111111, // 0 - A, B, C, D, E, F
	0b00000110, // 1 - B, C
	0b01011011, // 2 - A, B, D, E, G
	0b01001111, // 3 - A, B, C, D, G
	0b01100110, // 4 - B, C, F, G
	0b01101101, // 5 - A, C, D, F, G
	0b01111101, // 6 - A, C, D, E, F, G
	0b00000111, // 7 - A, B, C
	0b01111111, // 8 - A, B, C, D, E, F, G
	0b01101111, // 9 - A, B, C, D, F, G
};
void InitPorts(void);
void send_data(uint8_t data, uint8_t ind);
void InitTimer0(void);
void Bin2Dec(uint16_t data);
volatile uint16_t cnt = 0;
volatile uint8_t switch_state = 0;
volatile uint8_t bcd_buffer[] = {0,0,0,0};
int main(void)
{
	InitPorts();
	InitTimer0();
	EIMSK|=(1 << INT0); //ВключитьINT0
	EICRA|=(1 << ISC01); //Настройка INT0 на прерывание по спаду
	sei(); //Глобальное разрешение прерываний
	while(1)
	{
		if(switch_state == 0)
		{
		Bin2Dec(cnt);
		if(cnt < 9999)
		{
			cnt++;
		}
		else
		{
			cnt=0;
		}
```

```
            _delay_ms(100);
            }
      }
}
//-----------------------------------------
ISR(TIMER0_COMPA_vect){
      send_data(bcd_buffer[3],0);
      send_data(bcd_buffer[2],1);
      send_data(bcd_buffer[1],2);
      send_data(bcd_buffer[0],3);
}
ISR(INT0_vect)
{
      if(switch_state == 0)
      {
            switch_state = 1;
      }
      else
      {
            switch_state = 0;
            cnt = 0;
      }
}
void InitPorts(void)
{
      DDRB = 0xFF;
      DDRC = (1 << PINC0 | 1 << PINC1 | 1 << PINC2 | 1 << PINC3);
      PORTC = 0x0F;
      DDRD = (0 << PIND2);
      PORTD |= (1 << PIND2);
}
void send_data(uint8_t data, uint8_t ind)
{
      PORTC = 0x0F &~ (1<<ind);
      PORTB = segments[data];
      _delay_ms(5);
      PORTB = 0;
      PORTC = 0x0F;
}
void InitTimer0(void)
{
      TCCR0A = (1 << WGM01); //режим CTC - Clear Timer on
      //Compare
      TCCR0B = (1 << CS02 | 1 << CS00);//prescaler = sys_clk/1024
      TCNT0 = 0x00; //начальное значение счетчика
      OCR0A = 16; //порог срабатывания
      TIMSK0 |= (1 << OCIE0A);
      //включение прерывания при достижении порога A
}
void Bin2Dec(uint16_t data)
{
```

```
	bcd_buffer[3]=(uint8_t)(data/1000);
	data = data - bcd_buffer[3]*1000;
	bcd_buffer[2] = (uint8_t)(data/100);
	data = data - bcd_buffer[2]*100;
	bcd_buffer[1] = (uint8_t)(data/10);
	data = data - bcd_buffer[1]*10;
	bcd_buffer[0] = (uint8_t)(data);
}
```

**Схема:**




*2.Подключение индикаторов с помощью регистров*

**Код на С:**

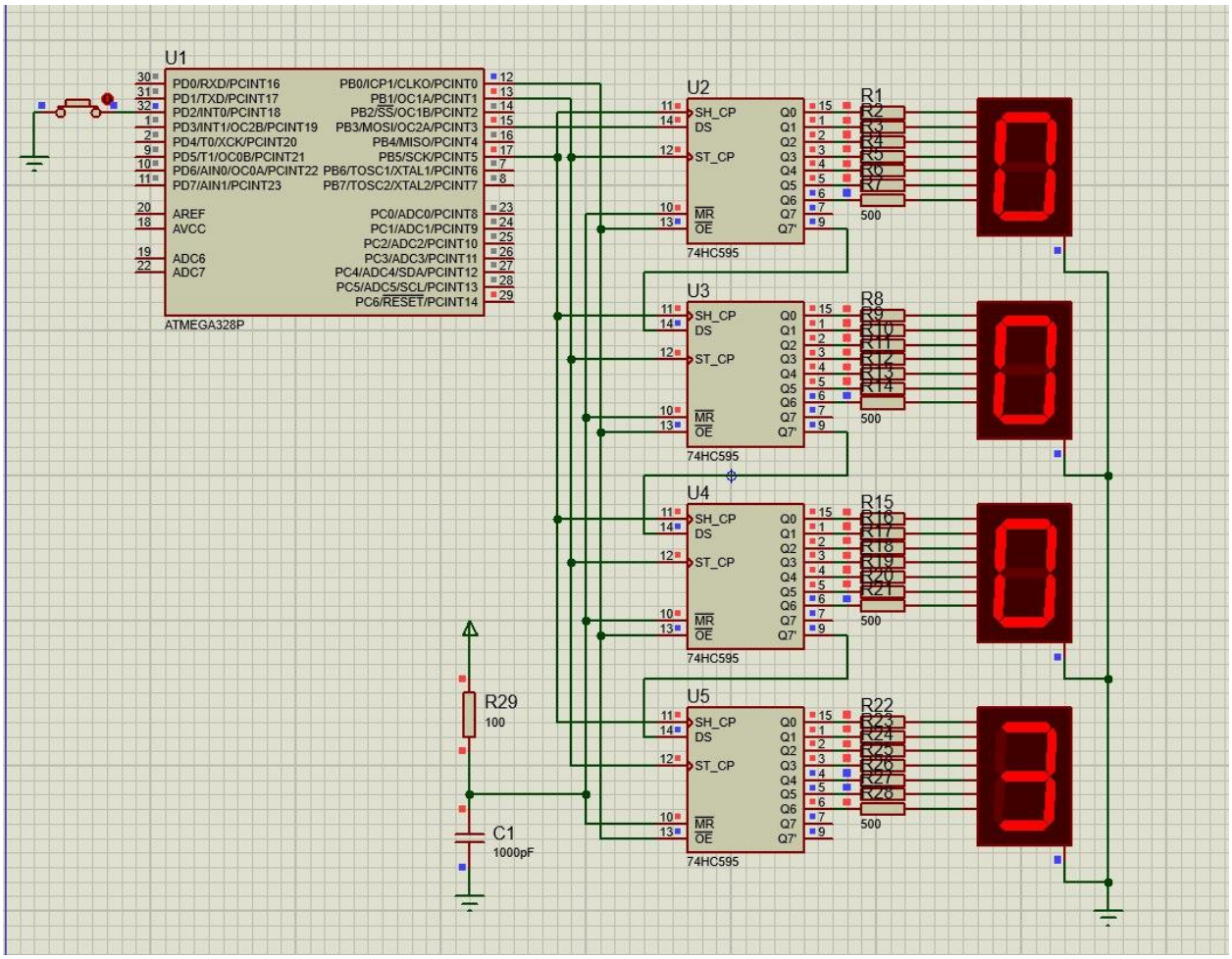
```
#define F_CPU 1000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
uint8_t segments[]={
	0b00111111, // 0 - A, B, C, D, E, F
	0b00000110, // 1 - B, C
	0b01011011, // 2 - A, B, D, E, G
	0b01001111, // 3 - A, B, C, D, G
	0b01100110, // 4 - B, C, F, G
	0b01101101, // 5 - A, C, D, F, G
	0b01111101, // 6 - A, C, D, E, F, G
	0b00000111, // 7 - A, B, C
	0b01111111, // 8 - A, B, C, D, E, F, G
	0b01101111, // 9 - A, B, C, D, F, G
};
void InitPorts(void);
```

```
void send_data(uint8_t data, uint8_t ind);
void InitTimer0(void);
void Bin2Dec(uint16_t data);
void InitTimer1(void);
void StartTimer1(void);
void StopTimer1(void);
void SendData(uint8_t data);
void DisplayData(uint16_t data);
volatile uint16_t cnt = 0;
volatile uint8_t switch_state = 0;
volatile uint8_t bcd_buffer[] = {0,0,0,0};
int main(void)
{
	InitPorts();
	InitTimer1();
	EIMSK |= (1 << INT0); //разрешить прерывание INT0
	EICRA |= (1 << ISC01);//Запуск по заднему фронту INT0
	sei();	//Разрешение прерываний
	PORTB &= ~(1 << PINB0); //OE = low (active)
	DisplayData(0);
	while(1)
	{ }
}
//-------------------------------------------
ISR(TIMER1_COMPA_vect)
{
	DisplayData(cnt);
	if(cnt < 9999)
	{
		cnt++;
	}
	else
	{
		cnt = 0;
	}
}
ISR(INT0_vect)
{
	if(switch_state == 0)
	{
		switch_state = 1;
		StartTimer1();
	}
	else
	{
		StopTimer1();
		DisplayData(cnt);
		switch_state = 0;
		cnt = 0;
	}
	}
```

```
//-------------------------------------------
void InitPorts(void)
{
	DDRB = (1 << PINB0 | 1 << PINB1 | 1 << PINB3 | 1 << PINB5);
	DDRD &= ~(1 << PIND2);
	PORTD |= (1 << PIND2);
}
void InitTimer1(void)
{
	TCCR1A = 0;
	TCCR1B = (1 << CS11 | 1 << CS10 | 1 << WGM12);
	TCNT1 = 0;
	OCR1A = 15624;
}
void StartTimer1(void)
{
	TCNT1 = 0;
	TIMSK1 |= (1 << OCIE1A);
}
void StopTimer1(void)
{
	TIMSK1 &= ~(1 << OCIE1A);
}
void Bin2Dec(uint16_t data)
{
	bcd_buffer[3] = (uint8_t)(data/1000);
	data = data % 1000;
	bcd_buffer[2] = (uint8_t)(data/100);
	data = data % 100;
	bcd_buffer[1] = (uint8_t)(data/10);
	data = data % 10;
	bcd_buffer[0] = (uint8_t)(data);
}
void SendData(uint8_t data)
{
	for(uint8_t i = 0; i < 8; i++)
	{
		PORTB &= ~(1 << PINB5);		//CLK low
		if(0x80 & (data << i))
		{
			PORTB |= 1 << PINB3; //DAT high
		}
		else
		{
			PORTB &= ~(1 << PINB3); //DAT low
		}
		PORTB |= (1 << PINB5); //CLK high
	}
}
void DisplayData(uint16_t data)
{
```

```
        Bin2Dec(data);
        PORTB &= ~(1 << PINB1);
        //clk_out = 0
        SendData(segments[bcd_buffer[0]]);
        SendData(segments[bcd_buffer[1]]);
        SendData(segments[bcd_buffer[2]]);
        SendData(segments[bcd_buffer[3]]);
        PORTB |= (1 << PINB1);
        //clk_out = 1
}
```

**Схема:**




## 2. Оптимизация основного кода

*1.Реализация динамической индикации*

Код на С:

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

```
	0b00111111,
	0b00000110,
	0b01011011,
	0b01001111,
	0b01100110,
	0b01101101,
	0b01111101,
	0b00000111,
	0b01111111,
	0b01101111
};
volatile uint16_t cnt = 0;
volatile uint8_t switch_state = 0;
volatile uint8_t bcd_buffer[4] = {0};
volatile uint8_t current_digit = 0;
void InitPorts(void)
{
	DDRB = 0xFF;
	DDRC = 0x0F;
	PORTC = 0x0F;
	DDRD &= ~(1 << PIND2);
	PORTD |= (1 << PIND2);
}
void send_data(uint8_t data, uint8_t ind)
{
	PORTC = 0x0F &~ (1 << ind);
	PORTB = segments[data];
}
void InitTimer0(void)
{
	TCCR0A = (1 << WGM01);
	TCCR0B = (1 << CS02) | (1 << CS00);
	OCR0A = 256;
	TIMSK0 |= (1 << OCIE0A);
}
void Bin2Dec(uint16_t data)
{
	bcd_buffer[3] = data / 1000;
	bcd_buffer[2] = (data / 100) % 10;
	bcd_buffer[1] = (data / 10) % 10;
	bcd_buffer[0] = data % 10;
}
int main(void)
{
	InitPorts();
	InitTimer0();
	EIMSK |= (1 << INT0);
	EICRA |= (1 << ISC01);
	sei();
	while (1)
	{
```

```
            if (!switch_state)
            {
                  Bin2Dec(cnt);
                  if (cnt < 9999) cnt++;
                  else cnt = 0;
                  _delay_ms(100);
            }
      }
}
ISR(TIMER0_COMPA_vect)
{
      send_data(bcd_buffer[3 - current_digit], current_digit);
      current_digit = (current_digit + 1) % 4;
}
ISR(INT0_vect)
{
      switch_state ^= 1;
      if (switch_state) cnt = 0;
}
```

*2.Подключение индикаторов с помощью регистров*

Код на С:

```
#define F_CPU 16000000UL
#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 uint16_t cnt = 0;
volatile uint8_t switch_state = 0;
volatile uint8_t bcd_buffer[4] = {0};
void InitPorts(void);
void SendData(uint8_t data);
void DisplayData(uint16_t data);
void InitTimer1(void);
void StartTimer1(void);
void StopTimer1(void);
void Bin2Dec(uint16_t data);
void InitPorts(void)
{
      DDRB = (1 << PINB0) | (1 << PINB1) | (1 << PINB3) | (1 << PINB5);
```

```
	DDRD &= ~(1 << PIND2);
	PORTD |= (1 << PIND2);
}
void InitTimer1(void)
{
	TCCR1A = 0;
	TCCR1B = (1 << CS11) | (1 << CS10) | (1 << WGM12);
	OCR1A = 15624;
}
void StartTimer1(void)
{
	TCNT1 = 0;
	TIMSK1 |= (1 << OCIE1A);
}
void StopTimer1(void)
{
	TIMSK1 &= ~(1 << OCIE1A);
}
void Bin2Dec(uint16_t data)
{
	bcd_buffer[3] = data / 1000;
	bcd_buffer[2] = (data % 1000) / 100;
	bcd_buffer[1] = (data % 100) / 10;
	bcd_buffer[0] = data % 10;
}
void SendData(uint8_t data)
{
	for (uint8_t i = 0; i < 8; i++)
	{
		PORTB &= ~(1 << PINB5);
		if (data & (0x80 >> i)) PORTB |= (1 << PINB3);
		else PORTB &= ~(1 << PINB3);
		PORTB |= (1 << PINB5);
	}
}
void DisplayData(uint16_t data)
{
	Bin2Dec(data);
	PORTB &= ~(1 << PINB1);
	SendData(segments[bcd_buffer[0]]);
	SendData(segments[bcd_buffer[1]]);
	SendData(segments[bcd_buffer[2]]);
	SendData(segments[bcd_buffer[3]]);
	PORTB |= (1 << PINB1);
}
int main(void)
{
	InitPorts();
	InitTimer1();
	EIMSK |= (1 << INT0);
	EICRA |= (1 << ISC01);
```

```
	sei();
	PORTB &= ~(1 << PINB0);
	DisplayData(0);
}
ISR(TIMER1_COMPA_vect)
 {
	DisplayData(cnt);
	if (cnt < 9999) cnt++;
	else cnt = 0;
}
ISR(INT0_vect)
{
	switch_state ^= 1;
	if (switch_state) StartTimer1();
	else
	{
		StopTimer1();
		DisplayData(cnt);
		cnt = 0;
	}
}
```

### 3. Задание из методички(1 вариант)

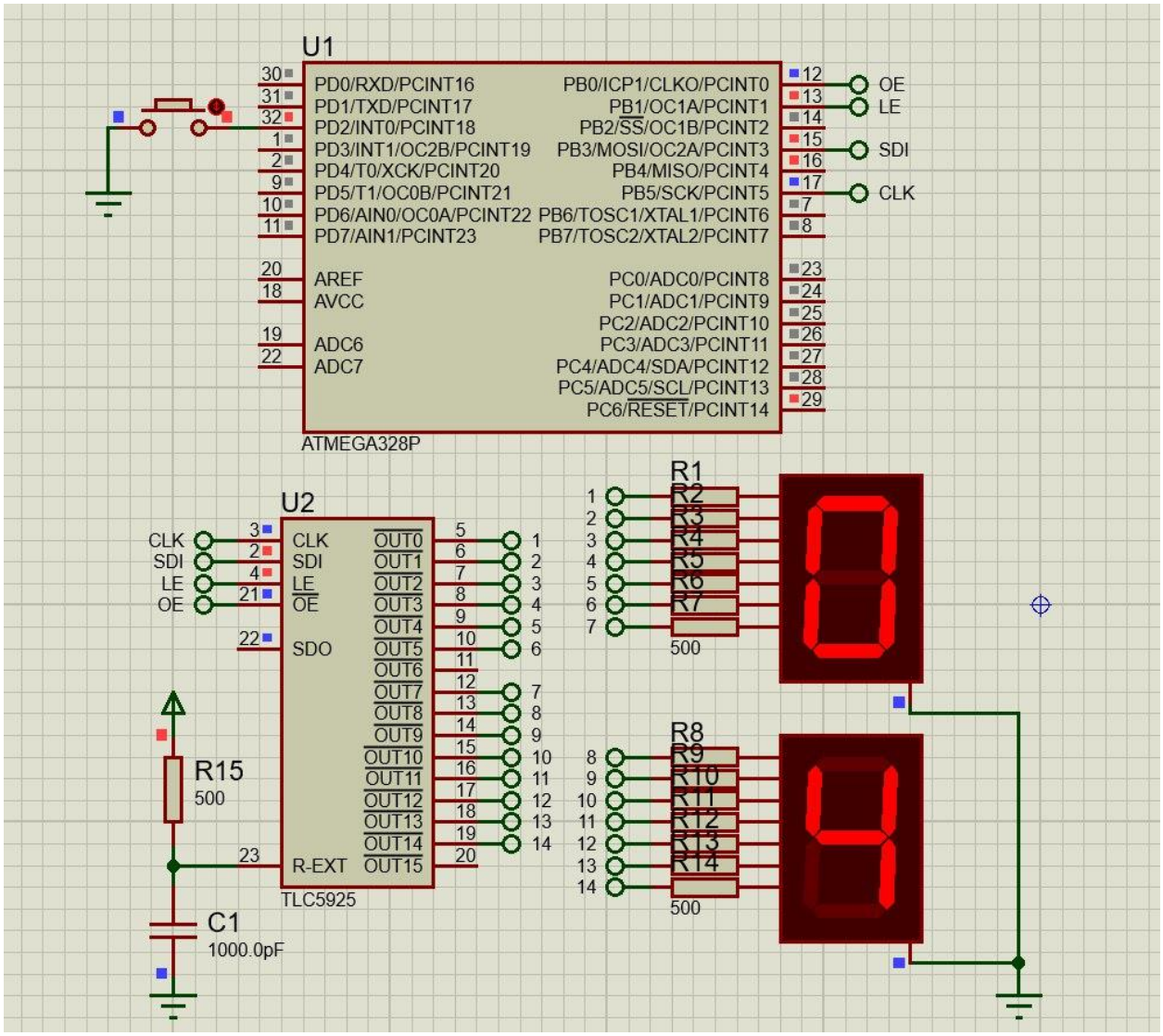
Код на С:

```
#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 uint16_t cnt = 0;
volatile uint8_t switch_state = 0;
volatile uint8_t bcd_buffer[2] = {0};
void InitPorts(void)
{
	DDRB = (1 << PINB0 | 1 << PINB1 | 1 << PINB3 |1 << PINB5);
	DDRD &= ~(1 << PIND2);
	PORTD |= (1 << PIND2);
}
void InitTimer1(void)
{
	TCCR1A = 0;
```

```
	TCCR1B = (1 << CS11 | 1 << CS10 | 1 << WGM12);
	TCNT1 = 0;
	OCR1A = 15624;
}
void StartTimer1()
{
	TCNT1 = 0;
	TIMSK1 |= (1 << OCIE1A);
}
void StopTimer1()
{
	TIMSK1 &= ~(1 << OCIE1A);
}
void Bin2Dec(uint16_t data)
{
	bcd_buffer[1] = (data % 100) / 10;
	bcd_buffer[0] = data % 10;
}
void SPI_send(uint8_t data)
{
	SPDR = data;
	while(!(SPSR & (1 << SPIF)));
}
void DisplayData(uint16_t data)
{
	Bin2Dec(data);
	PORTB &= ~(1 << PINB1);
	SPI_send(segments[bcd_buffer[0]]);
	SPI_send(segments[bcd_buffer[1]]);
	PORTB |= (1 << PINB1);
}
void InitSPI()
{
	DDRB |= (1 << PINB3 | 1 << PINB5);
	SPSR |= (1 << SPI2X);
	SPCR = (1 << SPE | 1 << MSTR);
	PORTB &= ~(1 << PINB3 | 1 << PINB5);
}
int main()
{
	InitPorts();
	InitTimer1();
	InitSPI();
	EIMSK |= (1 << INT0);
	EICRA |= (1 << ISC01);
	sei();
	PORTB &= ~(1 << PINB0);
	DisplayData(0);
	while(1){}
}
ISR(TIMER1_COMPA_vect)
{
	DisplayData(cnt);
	if(cnt < 99) cnt++;
	else cnt=0;
}
```

```
ISR(INT0_vect)
{
	if(switch_state == 0)
	{
		switch_state = 1;
		StartTimer1();
	}
	else
	{
		StopTimer1();
		DisplayData(cnt);
		switch_state = 0;
		cnt = 0;
	}
}
```

Схема:



**4. Задание от преподавателя(Гирлянда)**

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

Код на С:

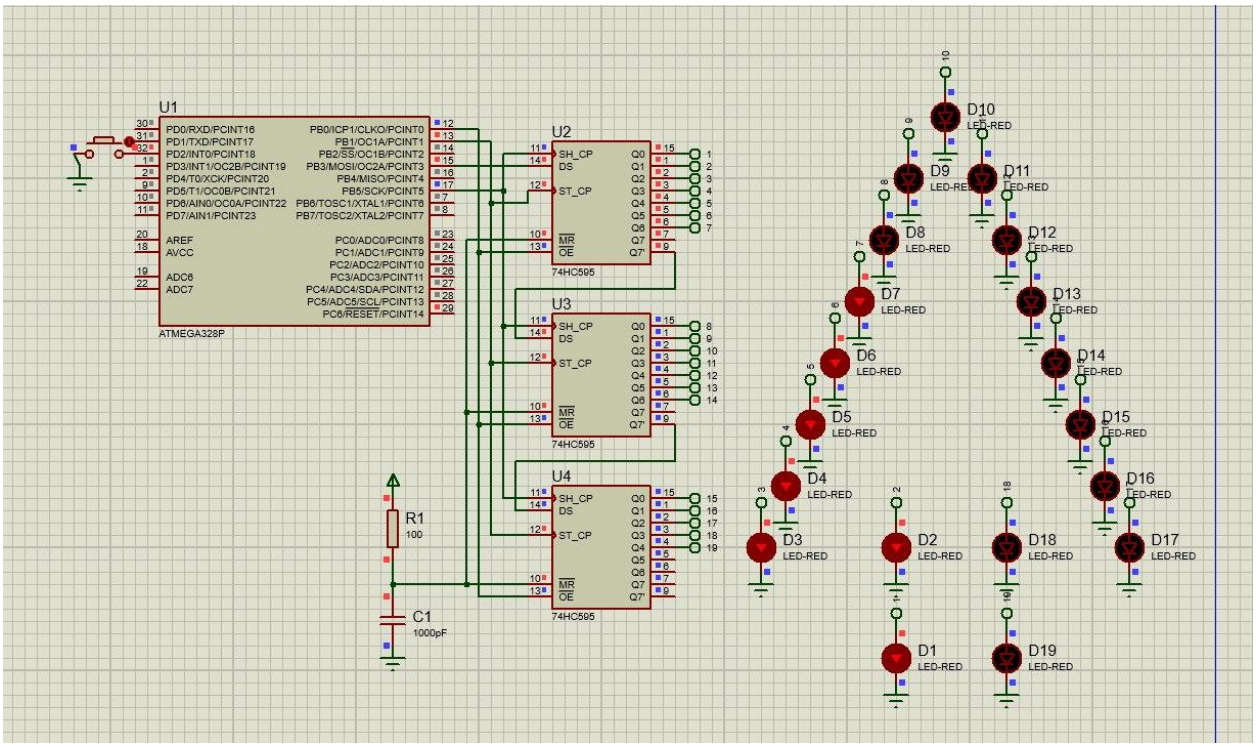
```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
volatile uint32_t leds_status = 0;
volatile uint8_t step_counter = 0;
volatile uint8_t current_effect = 0;
volatile uint8_t move_direction = 0;
volatile uint8_t timer_flag = 0;
const uint8_t leds_count[] = {1, 2, 6, 10, 15, 19};
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;
	}
```

```
}
void SendData(uint8_t data)
 {
	for (uint8_t i = 0; i < 8; i++)
	{
		PORTB |= (1 << PINB5);
		if (data & (1 << i))
		{
			PORTB |= (1 << PINB3);
		}
		else
		{
			PORTB &= ~(1 << PINB3);
		}
		PORTB &= ~(1 << PINB5);
	}
}
void DisplayData()
{
	PORTB &= ~(1 << PINB1);
	SendData((leds_status >> 16) & 0xFF);
	SendData((leds_status >> 8) & 0xFF);
	SendData(leds_status & 0xFF);
	PORTB |= (1 << PINB1);
}
void InitPorts(void)
{
	DDRB |= (1 << PINB0) | (1 << PINB1) | (1 << PINB3) | (1 << PINB5);
	PORTB |= (1 << PINB0);
	DDRD &= ~(1 << PIND2);
	PORTD |= (1 << PIND2);
}
void InitTimer1(void)
{
	TCCR1A = 0;
	TCCR1B = (1 << WGM12) | (1 << CS12) | (1 << CS10);
	OCR1A = 15624;
	TIMSK1 |= (1 << OCIE1A);
}
void StartTimer1(void)
{
	TCNT1 = 0;
	TIMSK1 |= (1 << OCIE1A);
}
void StopTimer1(void)
{
	TIMSK1 &= ~(1 << OCIE1A);
}
int main(void)
{
	InitPorts();
	InitTimer1();
	EIMSK |= (1 << INT0);
	EICRA |= (1 << ISC01);
	sei();
	PORTB &= ~(1 << PINB0);
	while (1)
	{
		if (timer_flag)
		{
			timer_flag = 0;
			switch (current_effect)
```

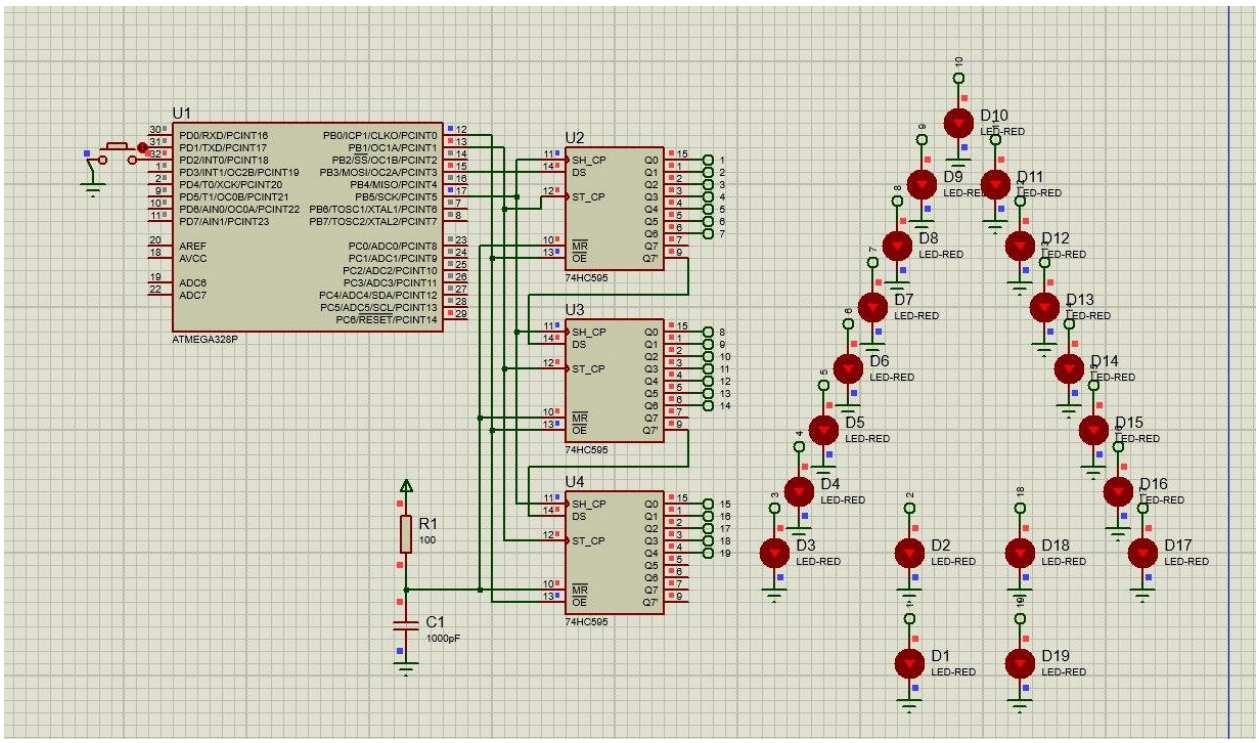
```
                        {
                                case 0: run_effect_0(); break;
                                case 1: run_effect_7(); break;
                                case 2: run_effect_6(); break;
                        }
                        DisplayData();
                }
        }
}
ISR(INT0_vect)
{
        current_effect = (current_effect + 1) % 3;
        leds_status = 0;
        step_counter = 0;
        move_direction = 1;
        if(current_effect == 0) StartTimer1();
        else StopTimer1();
}
ISR(TIMER1_COMPA_vect)
{
        timer_flag = 1;
}
```

Эффекты:

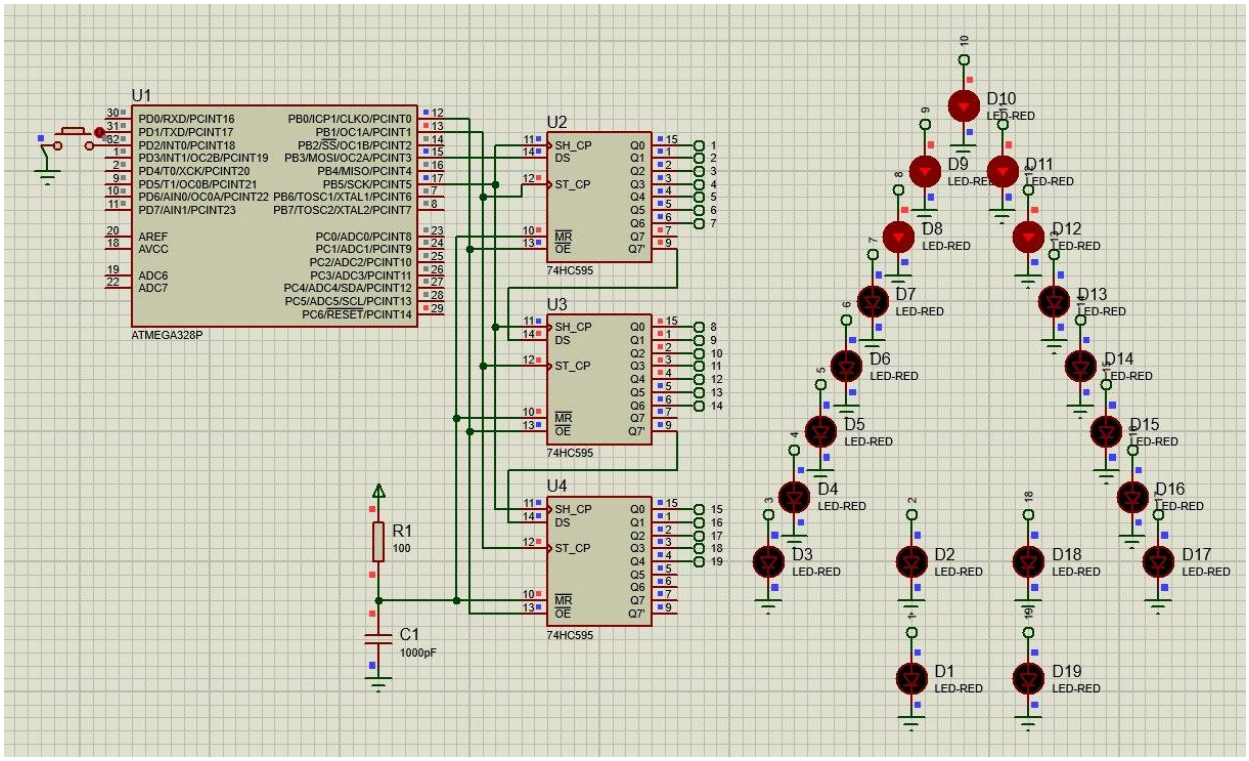
1) Эффект 0




2) Эффект 7




3) Эффект 6




**Выводы:** В данной лабораторной работе я познакомился с таким понятием как “таймер” и научился его применять в реализации на практике, а также подключать сдвиговые регистры к индикаторам. Также применил навыки из 5 лабораторной работы и воссоздал гирлянду, используя сдвиговые регистры.