

Практическое задание №3

Волков Егор Алексеевич

14 декабря 2024 г.

1 Схема подключения

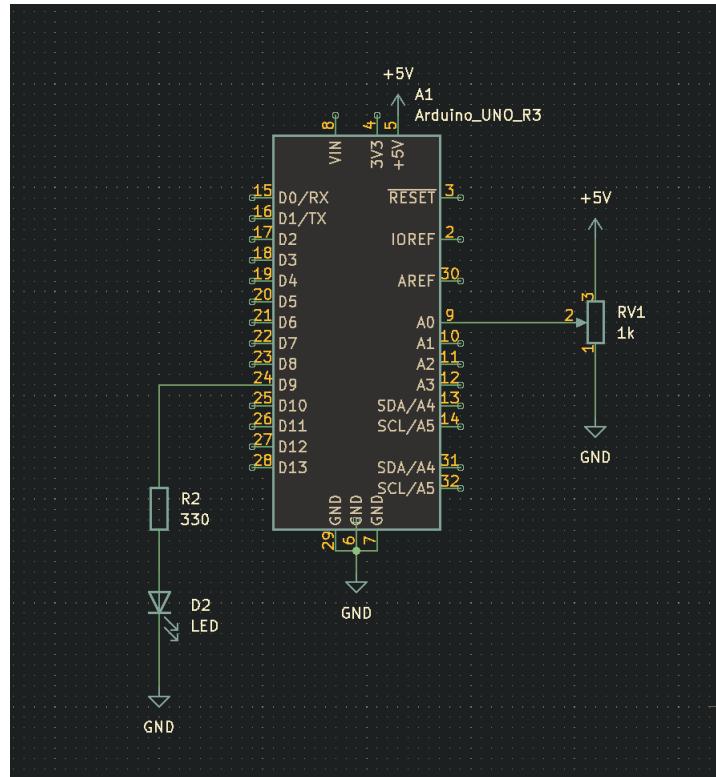


Рисунок 1. Схема подключения

1.1 Код

```
#include <avr/io.h>
#include <util/delay.h>

#define LED_PORT PORTB
#define LED_DDR DDRB
#define ADC_CHANNEL 0
#define LED_PIN PB1

void adc_init() {
    ADMUX = (1 << REFS0);
    ADCSRA = (1 << ADEN) | (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0);
}
```

```

uint16_t adc_read( uint8_t channel ) {
    ADMUX = (ADMUX & 0xF8) | (channel & 0x07);
    ADCSRA |= (1 << ADSC);
    while (ADCSRA & (1 << ADSC));
    return ADC;
}

void pwm_init() {
    DDRB |= (1 << LED_PIN);
    TCCR1A = (1 << WGM10) | (1 << WGM11) | (1 << COM1A1);
    TCCR1B = (1 << WGM12) | (1 << CS11);
}
void pwm_set_duty( uint8_t duty ) {
    OCR1A = duty;
}

int main(void) {
    adc_init();
    pwm_init();
    while (1) {
        uint16_t adc_value = adc_read(ADC_CHANNEL);
        uint8_t pwm_value = adc_value >> 2;
        pwm_set_duty(pwm_value);
        _delay_ms(10);
    }
}

```

2 Работа и сборка схемы

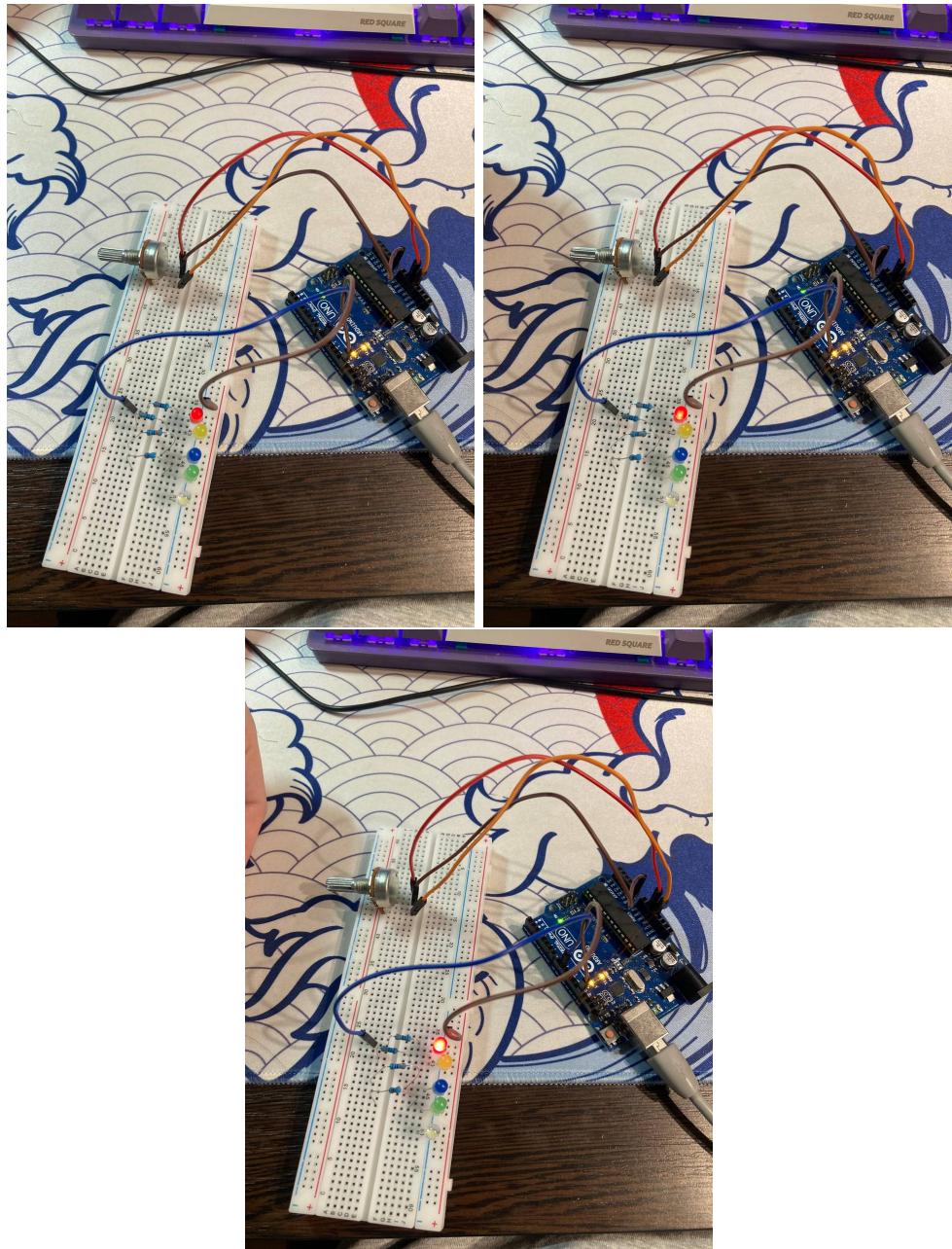


Рисунок 2. Работа схемы

3 Подключение датчика

Решил взять со дна ящика BMP180 и померить температуру с давлением. Датчик чувствует себя плохо, что уж сказать. Решил выводить

только температуру. С целью экономии времени решил не изобретать колеса и взять: <https://github.com/radionoise/avr-bmp180/> - библиотеку для работы с ним. Там к сожалению уже и I2C свой, но подменять имена я не стал и решил использовать его.

4 Код подключения датчика

```
#include <avr/io.h>
#include <util/delay.h>
#include "bmp180.h"
#include "i2c.h"
#include "uart.h"

#define SEGMENT_PORT_TENS PORTD
#define SEGMENT_DDR_TENS DDRD
#define SEGMENT_PORT_ONEC PORTB
#define SEGMENT_DDR_ONEC DDRB

const uint8_t digit[] = {
    0b0011111, // 0
    0b00000110, // 1
    0b01011011, // 2
    0b01001111, // 3
    0b01100110, // 4
    0b01101101, // 5
    0b01111101, // 6
    0b00000111, // 7
    0b01111111, // 8
    0b01101111 // 9
};

void display_tens(uint8_t value) {
    SEGMENT_PORT_TENS = digit[value];
}

void display_ones(uint8_t value) {
    SEGMENT_PORT_ONEC = (digit[value] >> 1);
    if (digit[value] & 1) {
        SEGMENT_PORT_TENS |= (1 << PB7);
```

```

        }

}

int main(void) {
    SEGMENT_DDR_TENS = 0xFF;
    SEGMENT_DDR_ONEC = 0xFF;

    int16_t temperature = 0;
    Bmp180CalibrationData calibrationData;
    uint8_t result = bmp180ReadCalibrationData(&calibrationData);
    if (result == BMP180_OK) {
        Bmp180Data bmp180Data;
        uint8_t counter = 0;
        while (1) {
            if (counter == 50) {
                result = bmp180ReadData(BMP180_OSS_STANDARD, &bmp180Data);
                if (result == BMP180_OK) {
                    temperature = bmp180Data.temperatureC;
                }
                counter = 0;
            }
            uint8_t tens = temperature / 10;
            uint8_t ones = temperature % 10;
            display_tens(tens);
            display_ones(ones);
            counter++;
            _delay_ms(10);
        }
    }
    return 0;
}

```

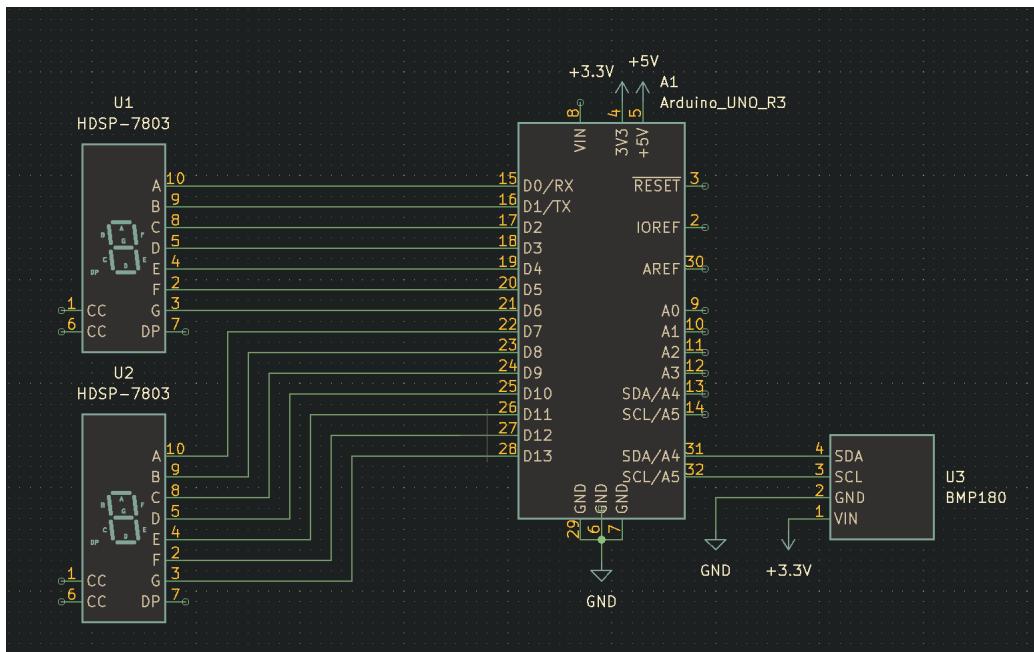
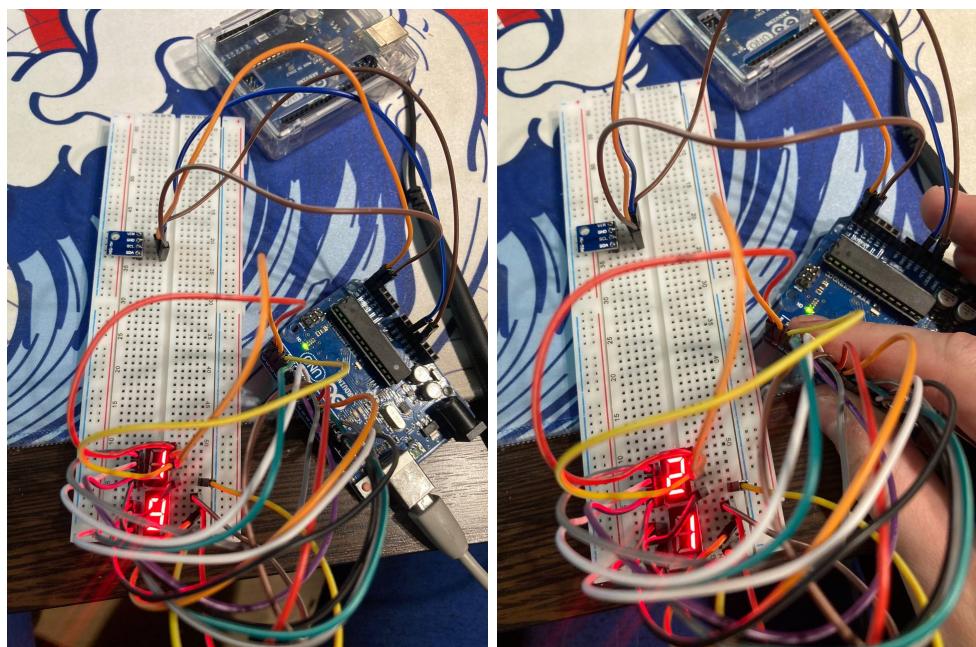


Рисунок 3. Схема подключения



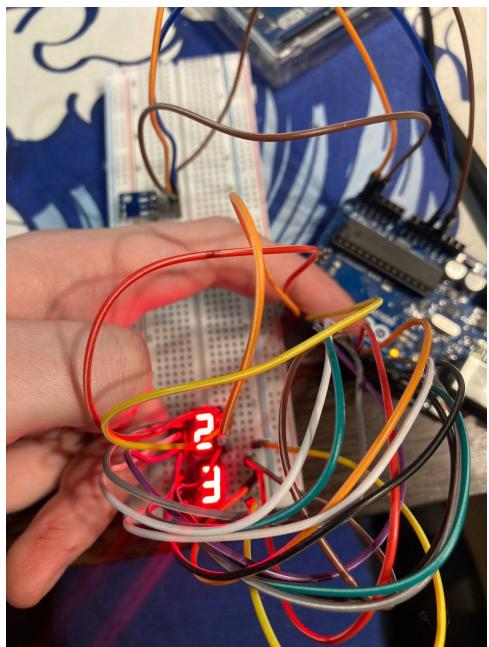


Рисунок 4. Работа схемы