

# Проектирование устройств

Домашнее задание №7

Евгений Зеленин

8 декабря 2024 г.

## 1. Работа с прерываниями

**Условия задачи.** *Задание 1.* Воспользуйтесь аппаратным прерыванием и “поймайте” сигнал с кнопки. Другими словами, ваша программа будет что то делать, например мигать светодиодом, но когда вы нажмете на кнопку, это значение будет выведено в *Serial*. Также это может быть звуковой сигнал, или любая индикация, которая вам по душе.

*Задание 2\*.* Необходимо сделать прерывание по таймеру, и при этом, чтобы каждый таймер “ловил” свое событие. Программа должна взаимодействовать как минимум с двумя таймерами. Можно использовать любые доступные библиотеки.

### Описание эксперимента

Объединим условия двух заданий в одно. Сделаем устройство, для управления режимами работы светодиодов, работающее по следующему принципу: нажатия двух кнопок обрабатываются с помощью аппаратных прерываний по нисходящему фронту импульса. В обработчике прерываний устанавливается или сбрасывается флаг, отвечающий за работу светодиода. Далее, первый светодиод управляется с помощью программного таймера через счетчик, а второй с помощью аппаратного через прерывание.

Схема подключения показана на рисунке 1. При нажатии на кнопку, статус соответствующего флага выводится в терминал из обработчика прерывания.

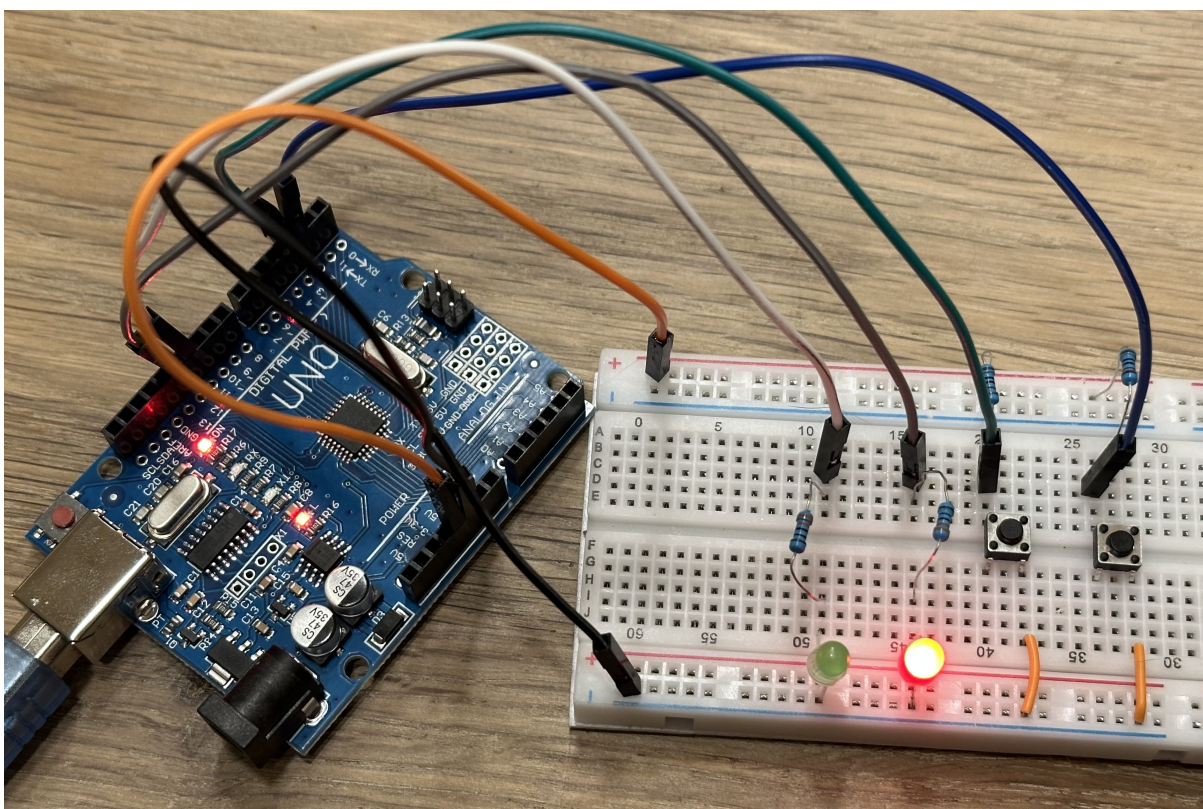


Рис. 1: Схема подключения

Для работы с таймерами использовалась библиотека *GyverTimers.h*. Ниже представлен код реализации проекта:

```

#include <GyverTimers.h>
#define BT1 2
#define BT2 4
#define LED1 11
#define LED2 12
//флаги разрешения на запуск диодов
bool led1_on = 0;
bool led2_on = 0;

//счетчик для второго светодиода
uint32_t cnt_2 = 0;

void setup() {
    // подключаем последовательный порт
    Serial.begin(9600);

    //подключение кнопок
    pinMode(BT1, INPUT);
    pinMode(BT2, INPUT);

    //подключение диодов
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);

    //запускаем таймер 1
    Timer1.setFrequency(4); // таймер для первого прерывания, частота 4 Гц
    Timer1.enableISR();

    //добавляем обработчики прерываний на кнопки
    attachInterrupt(0, bt1_interrupt, FALLING);
    attachInterrupt(1, bt2_interrupt, FALLING);
}

void loop() {
    //мигание диодом по программному таймеру
    if (led2_on && millis() - cnt_2 > 600) {
        cnt_2 = millis();
        digitalWrite(LED2, !digitalRead(LED2));
    }
}

//обработка прерывания по нажатию кнопки 1 с выводом статуса в терминал
void bt1_interrupt() {
    led1_on = !led1_on;
    Serial.println((String) "led1_on = " + led1_on);
};

```

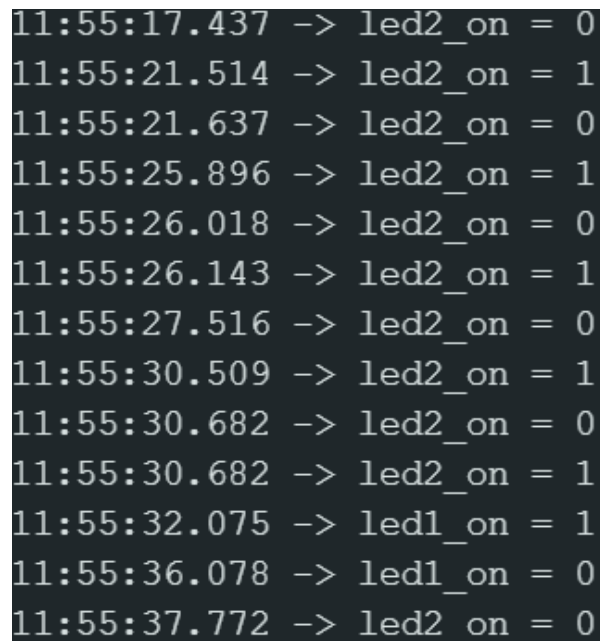
```

//обработка прерывания по нажатию кнопки 2 с выводом статуса в терминал
void bt2_interrupt() {
    led2_on = !led2_on;
    Serial.println((String) "led2_on = " + led2_on);
};

//обработчик прерывания таймера
ISR(TIMER1_A) {
    //моргаем светодиодом, если флаг взведен
    if (led1_on) {
        digitalWrite(LED1, !digitalRead(LED1));
    }
}

```

В программе реализован функционал для вывода статуса нажатий в терминал прямо из обработчика прерывания (рисунок 2).



```

11:55:17.437 -> led2_on = 0
11:55:21.514 -> led2_on = 1
11:55:21.637 -> led2_on = 0
11:55:25.896 -> led2_on = 1
11:55:26.018 -> led2_on = 0
11:55:26.143 -> led2_on = 1
11:55:27.516 -> led2_on = 0
11:55:30.509 -> led2_on = 1
11:55:30.682 -> led2_on = 0
11:55:30.682 -> led2_on = 1
11:55:32.075 -> led1_on = 1
11:55:36.078 -> led1_on = 0
11:55:37.772 -> led2_on = 0

```

Рис. 2: Вывод статуса из обработчика прерываний в терминал

## 2. Материалы к занятию

Демонстрация работы расположена в папке на google диск по следующей ссылке: Материалы к ДЗ №7