



**Министерство науки и высшего образования  
Российской Федерации  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технологический университет «СТАНКИН»  
(ФГБОУ ВО «МГТУ «СТАНКИН»)**

---

Институт цифровых интеллектуальных систем

Дисциплина: «Программирование встроенных систем управления»

**Лабораторная работа № 2**

**GPIO, управление портами ввода-вывода в Arduino**

Выполнил:

студент группы АДМ-21-05

\_\_\_\_\_  
(подпись)

Абдулзагиров М.М.  
(ФИО)

Принял

преподаватель:

\_\_\_\_\_  
(подпись)

Панфилов П. В.  
(ФИО)

Дата: \_\_\_\_\_

Москва 2022

## Работа с кнопками D2 D3.

### Задание

Написать программу которая выполет следующие действия:

#### Не продвинутый вариант

1. Короткое нажатие (менее 500 мс) на кнопку D2 зажигает красный светодиод (D12), длительное нажатие (более 500 мс) на эту кнопку гасит красный светодиод(D12)
2. Нажатие на кнопку D3 по очереди зажигает отдельные каналы многоцветного светодиода (D9-D11). Порядок зажигания синий, красный, зеленый, выключено. Для опроса этой кнопки использовать функцию `yield()`.
3. Смена состояния светодиодов и нажатия на кнопки выводится в консоль.

### Описание программы

В качестве IDE использовалась VS Code с расширением PlatformIO. В основном цикле идет обработка нажатий кнопки D2 для управления красным светодиодом. Дребезг обрабатывается программно с помощью задержки. Обработка нажатия на кнопку D3 идёт в цикле `yield()`, вызываемом во время задержки в основном цикле. Для обработки дребезга в данном случае нельзя использовать задержку, поэтом в данном случае с помощью `millis()` и набора флагов фиксируется время нажатия, и после 30 миллисекунд (можно и меньше) проверяется статус нажатия и происходит смена цвета.

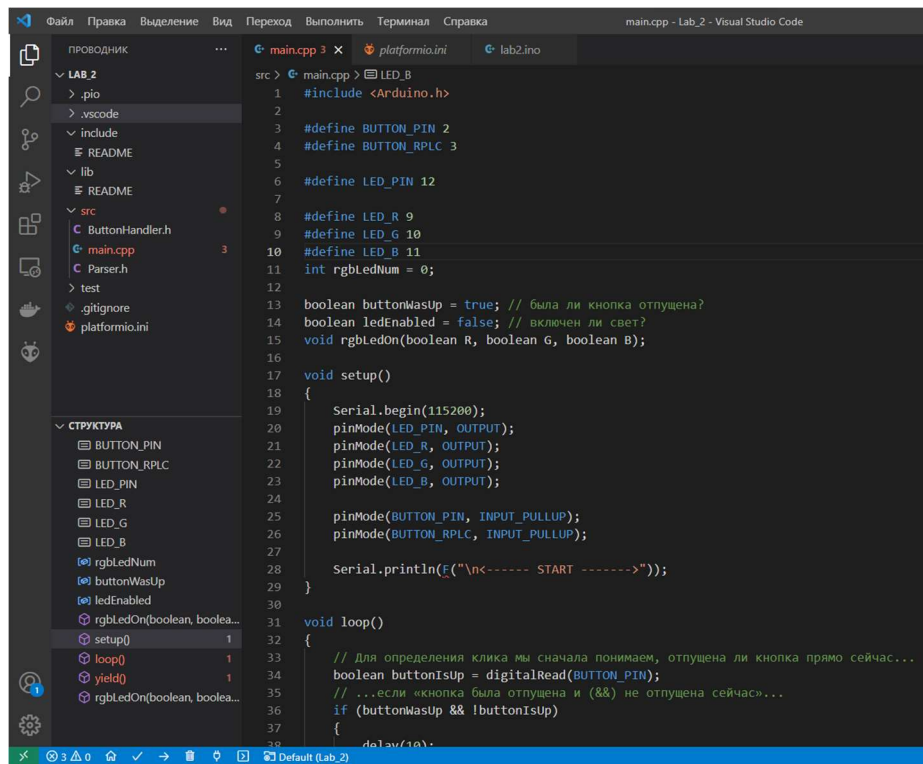


Рис 1. Окно VS Code.

Исходный код программы:

Листинг 1.

```

/*
 * МГТУ СТАНКИН
 * Программирование встроенных систем управления
 * Лабораторная работа №2
 * АДМ-21-05
 * Абдулзагиров Мурад Магомедович
 * Murad.Abdulzagirow@gmail.com
 */

//выводы кнопок
#define BUTTON_PIN 2
#define BUTTON_RPLC 3

// выводцы светодиодов
#define LED_PIN 12
#define LED_R 9
#define LED_G 10
#define LED_B 11
int rgbLedNum = 0;

boolean buttonWasUp = true; // была ли кнопка отпущена?
boolean ledEnabled = false; // включен ли свет?
void rgbLedOn(boolean R, boolean G, boolean B); // установка нужного цвета

```

```

void setup()
{
    Serial.begin(9600); // инициализация COM порта
    // инициализация выводов светодиодов
    pinMode(LED_PIN, OUTPUT);
    pinMode(LED_R, OUTPUT);
    pinMode(LED_G, OUTPUT);
    pinMode(LED_B, OUTPUT);
    // инициализация входов с внутренним стягивающим резистором
    pinMode(BUTTON_PIN, INPUT_PULLUP);
    pinMode(BUTTON_RPLC, INPUT_PULLUP);

    Serial.println("\n<----- START ----->");
}

void loop()
{
    // Для определения клика мы сначала понимаем, отпущена ли кнопка прямо сейчас...
    boolean buttonIsUp = !digitalRead(BUTTON_PIN);
    // ...если «кнопка была отпущена и (&&) не отпущена сейчас»...
    if (buttonWasUp && buttonIsUp)
    {
        delay(10);
        // ...и считываем сигнал снова
        buttonIsUp = !digitalRead(BUTTON_PIN);
        if (buttonIsUp)
        { // если она всё ещё нажата...
            // это короткий клик!
            // флаг состояния светодиода (зажечь)
            bool state = true;

            delay(500);
            // ...и считываем сигнал снова
            buttonIsUp = !digitalRead(BUTTON_PIN);
            if (buttonIsUp) // это длительный клик!
                state = false; //отключим всеодиод

            // установка полученного состояния
            digitalWrite(LED_PIN, state);
            Serial.print("Red led state: ");
            Serial.println(state);
        }
    }

    // запоминаем последнее состояние кнопки для новой итерации
    buttonWasUp = buttonIsUp;
    delay(10); //для работы функции yield() без нажатий
}

void yield()
{
    static long time;
    static long pastTime;
    static boolean pastIsUp = false;    // флаг нажатия в прошлом цикле

```

```

static boolean ledIsChange = false; //флаг

time = millis(); // текущее время
boolean buttonIsUp = !digitalRead(BUTTON_RPLC);
if (buttonIsUp && pastIsUp == false) //если кнопка нажать и она была отпущена
ранее
{
    pastIsUp = true;
    pastTime = time;    // сохраняем текущее время
    ledIsChange = true; //разрешаем переключить свет
}

// если прошло 30 миллисекунд и кнопка была нажата и всё ещё нажата и разрешено
переключить свет
if (time - pastTime >= 30 && pastIsUp && buttonIsUp && ledIsChange)
{
    // устанавливаем вледующий цвет (или выключение)
    switch (rgbLedNum)
    {
        case 0:
            rgbLedOn(1, 0, 0);
            Serial.println("RGB led state: red");
            rgbLedNum++;
            break;
        case 1:
            rgbLedOn(0, 1, 0);
            Serial.println("RGB led state: green");
            rgbLedNum++;
            break;
        case 2:
            rgbLedOn(0, 0, 1);
            Serial.println("RGB led state: blue");
            rgbLedNum++;
            break;
        case 3:
            rgbLedOn(0, 0, 0);
            Serial.println("RGB led state: off");
            rgbLedNum = 0;
            break;
        default:
            break;
    }
    ledIsChange = false; // запрещаем переключение цвета
}
if (!buttonIsUp && pastIsUp == true) //если кнопка сейчас разжата, но она была
нажата
    pastIsUp = false;                //устанавливаем статус прошлого нажатия в
false
}

/// установка нужного цвета
void rgbLedOn(boolean R, boolean G, boolean B)
{
    digitalWrite(LED_R, R);

```

```
digitalWrite(LED_G, G);  
digitalWrite(LED_B, B);  
}
```

## Результаты выполнения программы

Протестируем программу на отладочной плате arduino nano с микроконтроллером atMega 328P

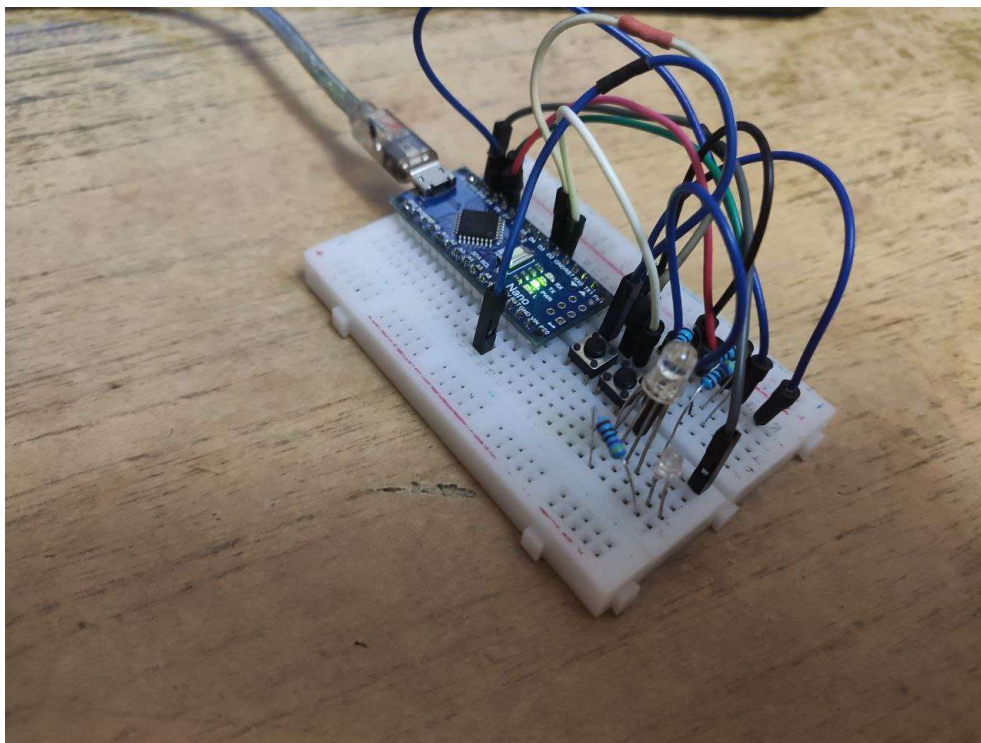


Рис.2. Отладочная плата.

При коротком нажатии на кнопку D2 включается красный светодиод.



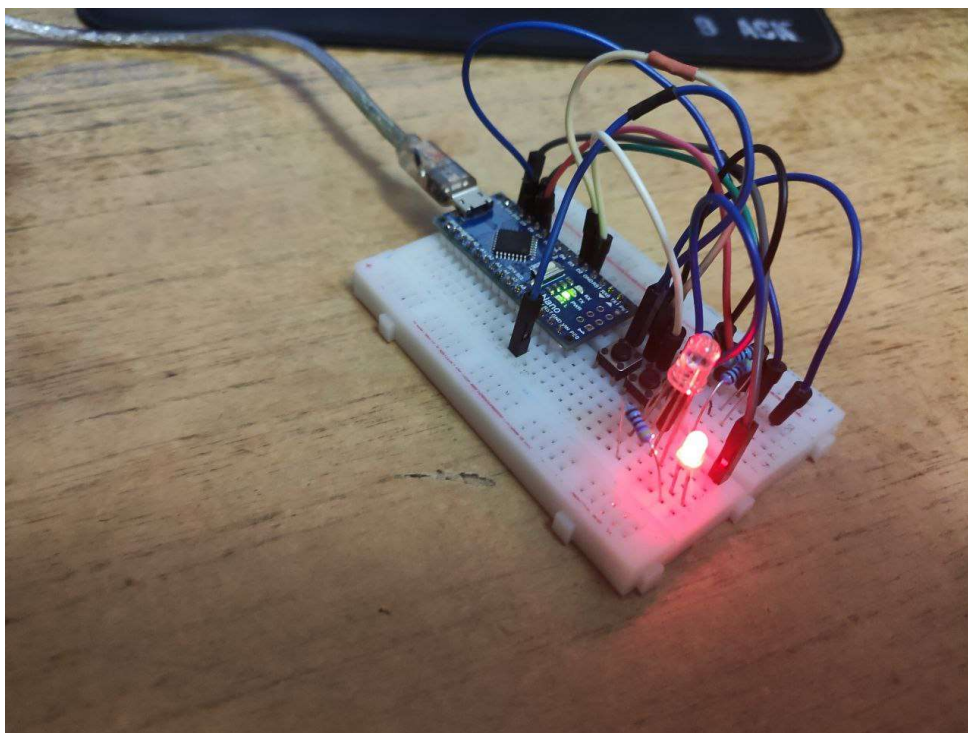


Рис.3. Отладочная плата.

Для выключения требуется держать нажатой кнопку более 500 мс.

При коротком нажатии на кнопку D3 загораются по очереди красный, зелёный и синий цвет, и при четвёртом нажатии светодиод погасает.

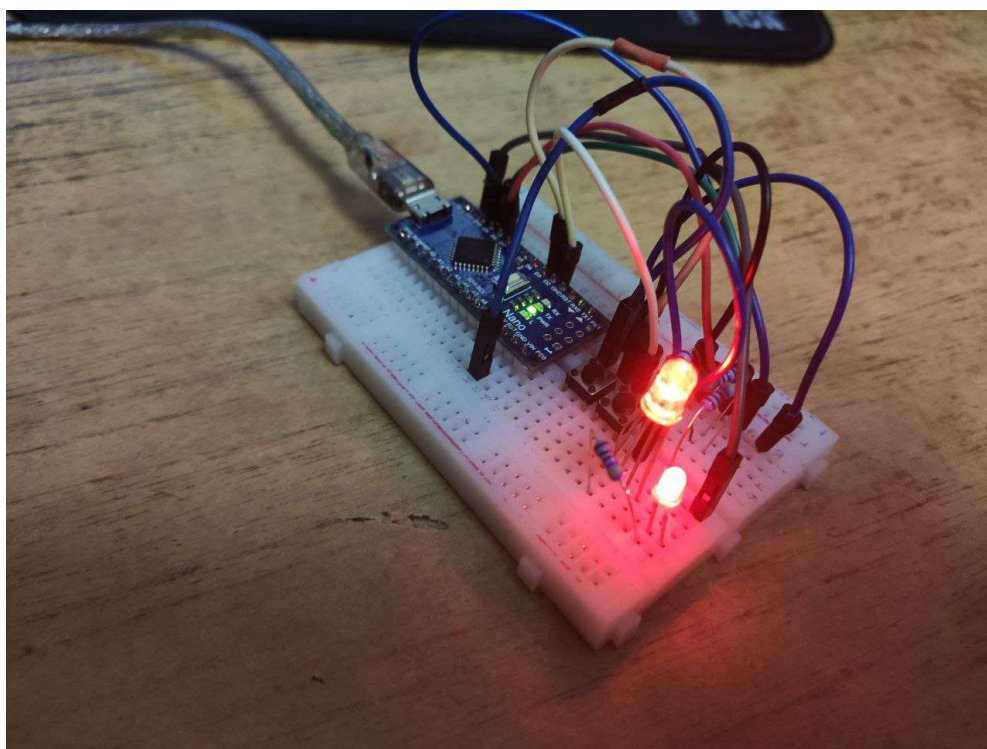


Рис.4. Отладочная плата.

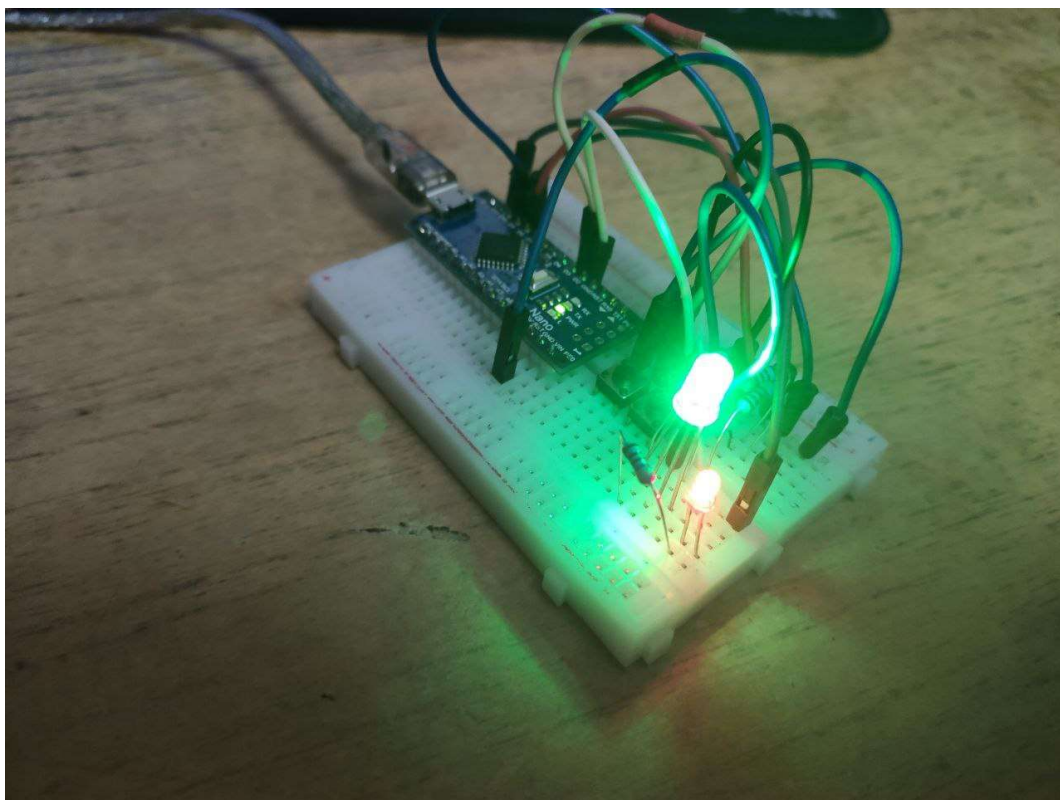


Рис.5. Отладочная плата.

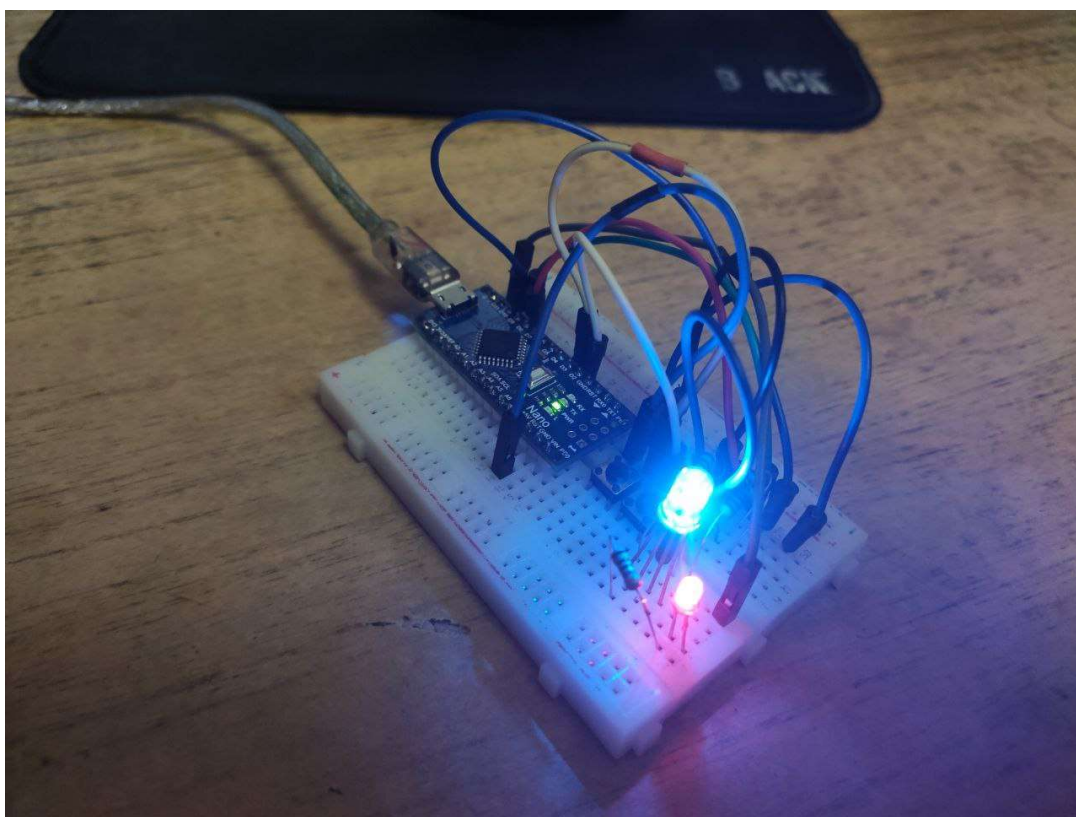


Рис.6. Отладочная плата.