



Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

ОТЧЕТ

по лабораторной работе № 6

Название: Программирование микроконтроллеров STM32

Дисциплина: Микропроцессорные системы

Студент

ИУ6-63Б

(Группа)

(Подпись, дата)

В.К. Залыгин

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Е.Ю. Гаврилова

(И.О. Фамилия)

Москва, 2025

Цель работы:

- знакомство с микроконтроллером семейства STM32 и приемами его программирования;
- получение навыков отладки программ в среде STM32CubeIDE;
- получение опыта прототипирования устройств с использованием макетной платы.

Вариант 16


№	Тип светофора	Описание программы
10		Даны две последовательности работы светофора. Последовательность А: красный – 1 с, желтый – 1 с, зеленый – 1 с. Последовательность Б: красный – 3 с, зеленый – 3 с. По нажатию кнопки переключать светофор с одной последовательности на другую: если активна А, то переключить на Б и наоборот.

Рисунок 1 – Индивидуальное задание по варианту

Граф состояний перехода светофора по индивидуальному варианту

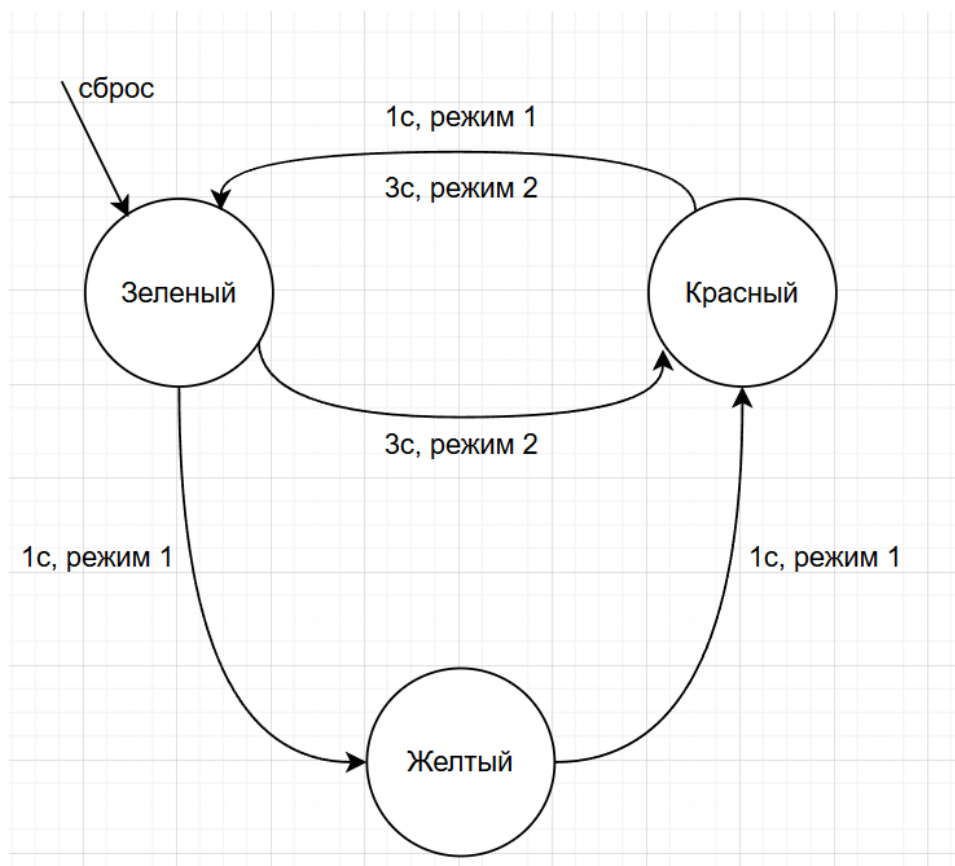


Рисунок 2 – Граф состояний перехода светофора

Добавим новую структуру данных TrafficLightStep, которая будет в себе содержать состояние светофора и некоторые переменные.

Листинг 1 - Новая структура

```
typedef enum
{
    TL_GREEN,
    TL_YELLOW,
    TL_RED
} TrafficLightState; //list of possible traffic light states
#define TrafficLight_GPIO_Port GPIOA
#define LED_ON 0U
#define LED_OFF 1U
#define TrafficLight_PlanSize1 3U // количество состояний
#define TrafficLight_PlanSize2 2U // количество состояний
#define TrafficLight_State_Duration1 6U // длительность состояния
#define TrafficLight_State_Duration2 6U*3 // длительность состояния
```

Подкорректируем работу со светодиодами в нашем варианте, оставив только нужные состояния.

Листинг 2 — Измененная функция TrafficLight_SetState

```
void TrafficLight_SetState(TrafficLightState state)
{
    switch (state)
    {
        case TL_GREEN:
            HAL_GPIO_WritePin(TrafficLight_GPIO_Port,
TL_RED_Pin|TL_YELLOW_Pin, LED_OFF);
            HAL_GPIO_WritePin(TrafficLight_GPIO_Port, TL_GREEN_Pin, LED_ON);
            break;
        case TL_YELLOW:
            HAL_GPIO_WritePin(TrafficLight_GPIO_Port,
TL_RED_Pin|TL_GREEN_Pin, LED_OFF);
            HAL_GPIO_WritePin(TrafficLight_GPIO_Port, TL_YELLOW_Pin,
LED_ON);
            break;
        case TL_RED:
            HAL_GPIO_WritePin(TrafficLight_GPIO_Port,
TL_GREEN_Pin|TL_YELLOW_Pin, LED_OFF);
            HAL_GPIO_WritePin(TrafficLight_GPIO_Port, TL_RED_Pin, LED_ON);
            break;
        default:
            break;
    }
}
```

Изменим функционал обработчика прерываний таймера под нашу структуру,

сравнивая теперь количество случившихся прерываний с установленной длительностью текущего состояния.

Листинг 3 — Измененная функция Timer_OverflowHandler

```
void Timer_OverflowHandler(void)
{
    HAL_GPIO_TogglePin(BOARD_LED_GPIO_Port, BOARD_LED_Pin); //toggle
    LED
    TimerOverflowCount++;

    if (state == 0) {
        if (TimerOverflowCount == TrafficLight_State_Duration1) {
            TimerOverflowCount = 0;
            TrafficLight_PlanIndex++;

            if (TrafficLight_PlanIndex == TrafficLight_PlanSize1) {
                TrafficLight_PlanIndex = 0;
            }

            TrafficLight_SetState(TrafficLight_Plan[TrafficLight_PlanIndex
]);
        }
    } else {
        if (TimerOverflowCount == TrafficLight_State_Duration2) {
            TimerOverflowCount = 0;
            TrafficLight_PlanIndex++;

            if (TrafficLight_PlanIndex == TrafficLight_PlanSize2) {
                TrafficLight_PlanIndex = 0;
            }

            TrafficLight_SetState(TrafficLight_Plan[TrafficLight_PlanIndex
]);
        }
    }
}
```

Также по заданию мы должны переключаться в новое состояние, не дожидаясь таймера после нажатия на кнопку. Реализуем переход в следующее состояние в обработчике нажатия этой кнопки.

Листинг 4 — Измененная функция HAL_GPIO_EXTI_Callback

```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if (GPIO_Pin == EXT_BUTTON_Pin) //check if external button
    triggered
    {
        if (state == 0) {
            state = 1;
        } else {
```

```
    state = 0;  
  }  
}  
}
```

На рисунке 3 представлен макет светофора с кнопкой.

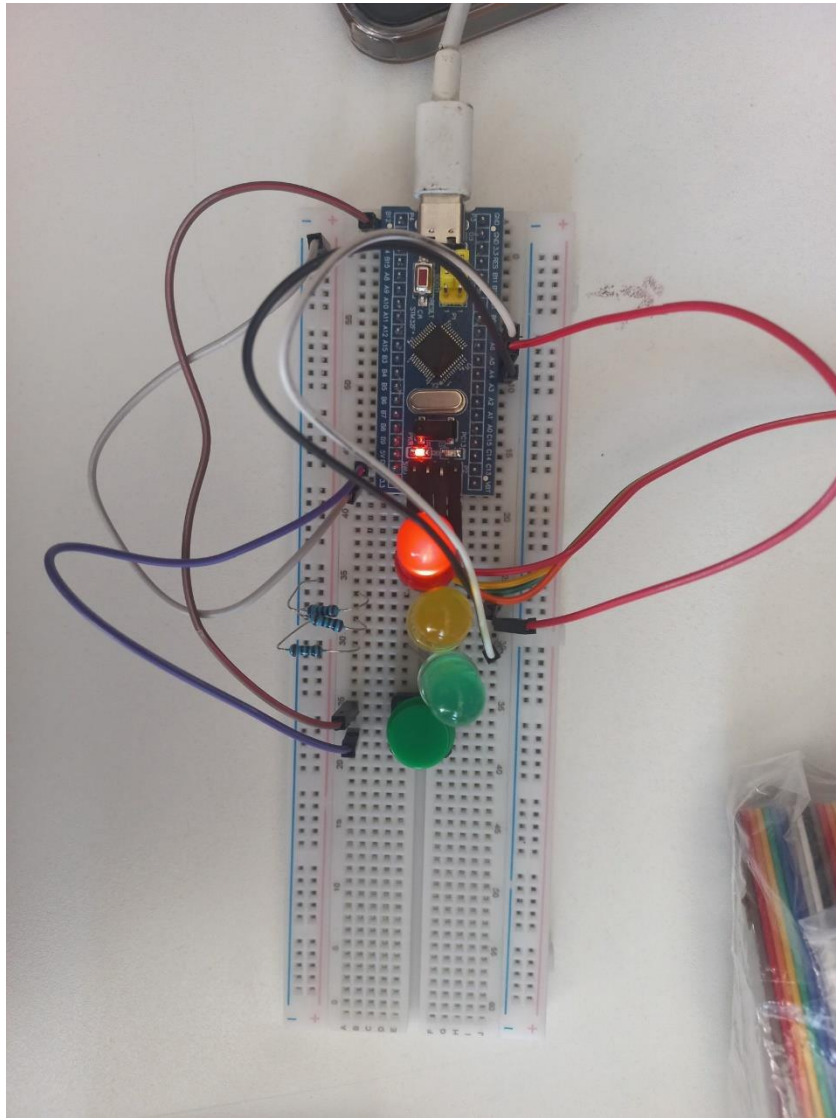


Рисунок 3 — Макет

Вывод

В ходе работы было проведено знакомство с микроконтроллером семейства STM32, изучены основные приёмы его программирования в среде STM32CubeIDE, а также получены навыки отладки и прототипирования устройств на макетной плате.