

Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”  
Факультет ПИиКТ



**УНИВЕРСИТЕТ ИТМО**

ОТЧЁТ

По лабораторной работе №1

По предмету: Проектирование вычислительных систем

Вариант 1

Студенты:

Андрейченко Леонид Вадимович

Степанов Михаил Алексеевич

Группа Р34301

Преподаватель:

Пинкевич Василий Юрьевич

## Цель работы

1. Получить базовые знания о принципах устройства стенда SDK-1.1M и программировании микроконтроллеров.
2. Изучить устройство интерфейсов ввода-вывода общего назначения (GPIO) в микроконтроллерах и приемы использования данных интерфейсов.

## Задание

Разработать и реализовать драйверы управления светодиодными индикаторами и чтения состояния кнопки стенда SDK-1.1M (расположены на боковой панели стенда). Контакты подключения кнопки и светодиодов должны быть настроены в режиме GPIO. Функции и другие компоненты драйверов должны быть универсальными, т.е. пригодными для использования в любом из вариантов задания и не должны содержать прикладной логики программы. Функции драйверов должны быть неблокирующими, то есть не должны содержать ожиданий события (например, нажатия кнопки). Также, в драйверах не должно быть пауз с активным ожиданием функция `HAL_Delay()` и собственные варианты аналогичной реализации. Обработка нажатия кнопки в программе должна включать программную защиту от дребезга.

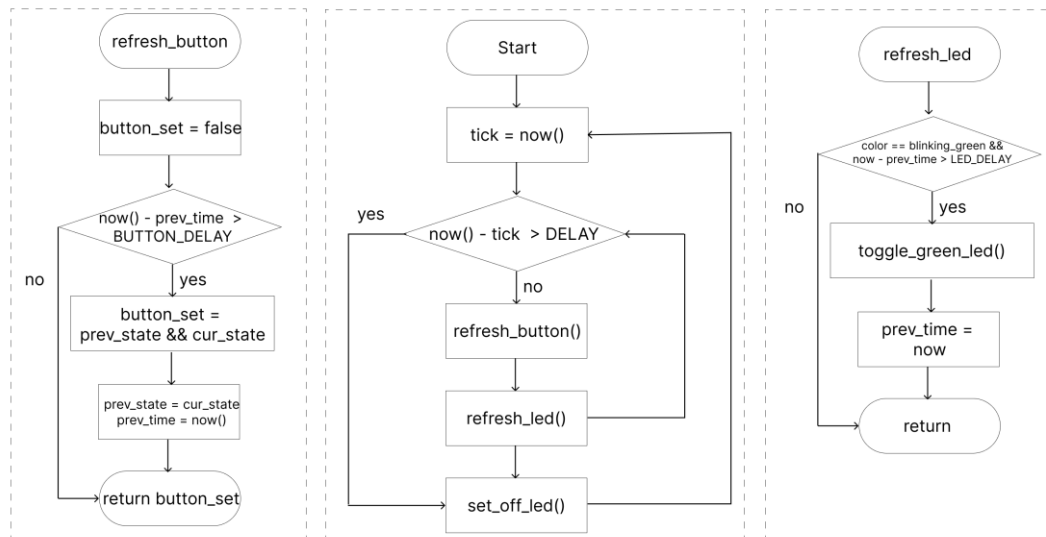
## Вариант

Сымитировать работу светофора пешеходного перехода. Светофор циклически переключает цвета в следующем порядке (порядок условный, соответствие реальному светофору не соблюдается): красный, зелёный, зелёный мигающий, жёлтый, снова красный и т.д. По умолчанию период горения красного в четыре раза больше периода горения зеленого. Если во время горения зеленого мигающего, желтого или красного нажимается кнопка, светофор запоминает необходимость скорейшего переключения на зелёный. После нажатия кнопки общий цикл работы светофора не нарушается, но период горения красного должен быть сокращен до  $1/4$  своего обычного периода. Если кнопка нажата во время горения красного, когда он уже горит более  $1/4$  периода, то сразу происходит переключение на зеленый.

## Организация программы

Основная программа представляет из себя цикл, в котором происходит опрос кнопки и обновление светодиодов. Функции обновления светодиодов и кнопки проверяют момент последнего обновления состояния и принимают решение об его изменении. Также в программе реализована защита от дребезга кнопки для считывания корректного состояния.

## Блок схема алгоритма



## Исходный код

### Main.c

```

43  /* USER CODE BEGIN PV */
44  enum State state = Green;
45  enum State next_state[4] = {Blinking_Green, Yellow, Red
46  uint8_t reduced_red = FALSE;
47  uint32_t tick_loop;
48  uint32_t wait = GREEN_DELAY;
49  /* USER CODE END PV */
50
51  /* Private function prototypes -----
52  void SystemClock_Config(void);
53  static void MX_GPIO_Init(void);
54  /* USER CODE BEGIN PFP */
55
56  /* USER CODE END PFP */
57
58  /* Private user code -----
59  /* USER CODE BEGIN 0 */
60
61  void go_next_state() {
62      reset_led();
63      state = next_state[state];
64  }
65
66  uint8_t need_to_reduce_red(){
67      return !check_button() && !reduced_red && state !=
68  }
69
70  void refresh_delay() {
71      wait = GREEN_DELAY;
72      if (state == Red && !reduced_red)
73          wait *= 4;
74      if (state == Green)
75          reduced_red = FALSE;
76  }
77
78  void reduce_red() {
79      reduced_red = TRUE;
80      if (state == Red)
81          wait = wait / 4;
82  }
83
84  /* USER CODE BEGIN WHILE */
121 while (1)
122 {
123     tick_loop = HAL_GetTick();
124     refresh_delay();
125     while((HAL_GetTick() - tick_loop) < wait)
126     {
127         set_led(state);
128         if (need_to_reduce_red()) {
129             reduce_red();
130         }
131     }
132     go_next_state();

```

## Button\_tools.c

```
1 #include "main.h"
2
3 GPIO_PinState prev = GPIO_PIN_SET;
4 uint32_t tick_button;
5
6 void init_button() {
7     tick_button = HAL_GetTick();
8 }
9
10 GPIO_PinState check_button() {
11     GPIO_PinState now = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15);
12     GPIO_PinState result = (now == prev) ? now : GPIO_PIN_RESET;
13     if ((HAL_GetTick() - tick_button) > BUTTON_DELAY){
14         tick_button = HAL_GetTick();
15         prev = now;
16     }
17     return result;
18 }
```

## Led\_tools.c

```
1 #include "main.h"
2
3 uint32_t tick_led;
4
5 void init_led() {
6     tick_led = HAL_GetTick();
7 }
8
9 void set_led(enum State state) {
10     switch (state) {
11         case Green:
12             HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_SET);
13             break;
14         case Blinking_Green:
15             if ((HAL_GetTick() - tick_led) > BLINK_DELAY){
16                 HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_13);
17                 tick_led = HAL_GetTick();
18             }
19             break;
20         case Yellow:
21             HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_SET);
22             break;
23         case Red:
24             HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_SET);
25             break;
26     }
27 }
28
29 void reset_led() {
30     HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13|GPIO_PIN_14|GPIO_PIN_15, GPIO_PIN_RESET);
31 }
```

## Выводы

В ходе выполнения данной лабораторной работы мы получили базовые знания о принципах устройства стенда SDK-1.1M и программировании микроконтроллеров.

Изучили устройство интерфейсов ввода-вывода общего назначения (GPIO) в микроконтроллерах и приемы использования данных интерфейсов.