

**Национальный исследовательский университет ИТМО**

**Факультет программной инженерии и компьютерной техники**

Лабораторная работа №2  
по дисциплине «Встроенные системы»

Выполнил: Ларочкин Г.  
Шуст И.  
Группа: Р3400  
Преподаватель: Ключев А.О.

Санкт-Петербург  
2020 г.

## Описание виртуальной лаборатории (cLab)

Для работы предоставлен базовый шаблон, в который можно писать пользовательский код. Исходный код шаблона находится по ссылке [https://github.com/lmtspbru/SDK\\_cLAB](https://github.com/lmtspbru/SDK_cLAB). Клиентская часть выполнена в виде веб-приложения, где пользователь может загрузить свою программу на исполнение.

В интерфейсе пользователя есть поле для загрузки bin-файла проекта и 2 окна: терминал, в котором отображается вывод программы по SDK\_TRACE\_Print и график прошедших событий.

Данная виртуальная лаборатория имеет 3 лампочки, возможность добавить 10 пользовательских событий и функции для вывода текста в консоль. Кроме того, в системе можно работать с прерываниями и таймером. Можно использовать трассировочный буфер для вывода каких-либо данных в терминал и для отображения событий возможно между командами *MX\_TRACE\_Init()* и *SDK\_TRACE\_Stop()*. При этом на выполнение пользовательского кода отведено 8 секунд.

## Описание задачи

Реализовать с помощью LED передачу сообщения «Привет» с помощью азбуки Морзе («Привет» = «.--. .-. .. --. -»). Длительность «-» в 2 раза больше длительности «.» Длительность паузы равна длительности «.» Зеленым передавать «-», желтым «.», красным – паузу.

## Выполнение

Для работы необходимо написать исходный код на языке C, конфигурация портов ввода-вывода, а также различной периферии производится в STM32CubeIDE (рекомендуется) или STM32CubeMX. Необходимо собрать проект и отправить бинарный файл на сервер.

Для передачи сообщения “Привет” с помощью LED было реализовано три функции *led\_dash()*, *led\_dot()*, *led\_delay()* которые зажигают светодиод необходимым цветом и делают паузу T миллисекунд.

## Описание схемы светодиодов

Для того, чтобы зажечь светодиод необходимым цветом необходимо подать корректное значение на соответствующие пины. Схему светодиодов можно рассмотреть на рисунке 1.

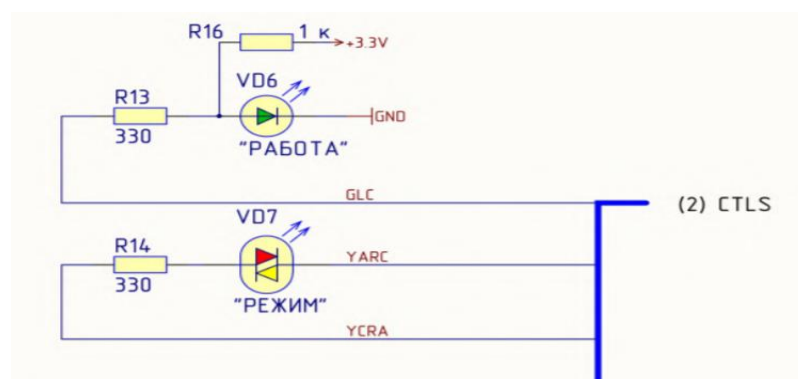


Рис. 1 – схема светодиодов на STM32

Для того, чтобы зажечь зеленый цвет, необходимо выставить сигнал *GLC*, а сигналы *YARC*, *YCRA* убрать. Чтобы зажечь желтый цвет необходимо подать сигнал на *YARC* и убрать сигнал на *GLC* и *YCRA*. Соответственно, чтобы зажечь красный цвет необходимо подать сигнал *YCRA*, а сигналы *YARC*, *GLC* убрать.

## Описание пинов

Для определения пинов соответствующих сигналов необходимо заглянуть в электрическую принципиальную схему SDK-1.1M. Материал можно найти по ссылке [https://sdk.ifmo.ru:9443/resource/sdk/sdk11m\\_schematic.pdf](https://sdk.ifmo.ru:9443/resource/sdk/sdk11m_schematic.pdf). На рисунке 2 можно увидеть пины для данных сигналов, они находятся на порте GPIOD на пинах PD13, PD14, PD15.

PORT D	
PD0	81 USART2/CAN1_RX
PD1	82 USART2/CAN1_TX
PD2	83
PD3	84 USART2_CTS
PD4	85 USART2_RTS
PD5	86 USART2/CAN1_TX
PD6	87 USART2/CAN1_RX
PD7	88 I2C1_NIRST
PD8	55 U3_TX/SPI3_MOSI
PD9	56 U3_RX/SPI3_MISO
PD10	57
PD11	58 U3_CTS/SPI3_NCS
PD12	59 U3_RTS/SPI3_SCK
PD13	60 GLC
PD14	61 YARC
PD15	62 YCRA

Рис. 2 – схема GPIOD SDK-1.1M

## Описание программы

Для подачи сигнала SET/RESET на пин использовалась функция *HAL\_GPIO\_WritePin()*, а для задержки использовалась функция *HAL\_DELAY()*. Далее представлен код, который реализует передачу сообщения с помощью подсветки LED.

```

/* Private user code -----*/
/* USER CODE BEGIN 0 */
#define T 150
void led_dash()
{
    SDK_TRACE_Timestamp(LED1, 1);
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_RESET);
    HAL_Delay(2*T);
    SDK_TRACE_Timestamp(LED1, 0);
}

void led_dot()
{

```

```

    SDK_TRACE_Timestamp(LED2, 1);
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_RESET);
    HAL_Delay(T);
    SDK_TRACE_Timestamp(LED2, 0);
}

void led_delay()
{
    SDK_TRACE_Timestamp(LED3, 1);
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_SET);
    HAL_Delay(T);
    SDK_TRACE_Timestamp(LED3, 0);
}

/* USER CODE END 0 */
/**
 * @brief The application entry point.
 * @retval int
 */
int main(void)
{
    /* USER CODE BEGIN 1 */
    /* USER CODE END 1 */
    /* MCU Configuration-----*/
    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();
    /* USER CODE BEGIN Init */
    /* USER CODE END Init */
    /* Configure the system clock */
    SystemClock_Config();
    /* USER CODE BEGIN SysInit */
    /* USER CODE END SysInit */
    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_IWDG_Init();
    /* USER CODE BEGIN 2 */
    /* Do not remove this code below */
    MX_TRACE_Init();
    SDK_TRACE_Start();
    /* Do not remove this code from above */
    /* Place your code from here */

    // Привет using Morse: .-- .-- .. -- . -
    led_dot();
    led_dash();
    led_dash();

```

```

led_dot();
led_delay();

led_dot();
led_dash();
led_dot();
led_delay();

led_dot();
led_dot();
led_delay();

led_dot();
led_dash();
led_dash();
led_delay();

led_dot();
led_delay();

led_dash();
led_delay();

SDK_TRACE_Timestamp(PRINT, 1);
SDK_TRACE_Print("%s", "Привет");
SDK_TRACE_Timestamp(PRINT, 0);

/* Place your code before here */
/* Do not remove this code below */
SDK_TRACE_Stop();
/* Do not remove this code from above */
}

```

## Результат программы

На рисунке 3 можно наблюдать результат выполнения программы в виртуальной лаборатории. На диаграмме событий видно передаваемое сообщение *Привет*. Также явно заметно, что передача символа “-” занимает в два раза больше времени, чем “.” В самом выводе программы видно сообщение “Привет” и его timestamp, которое и должно было вывестись в конце всей программы.

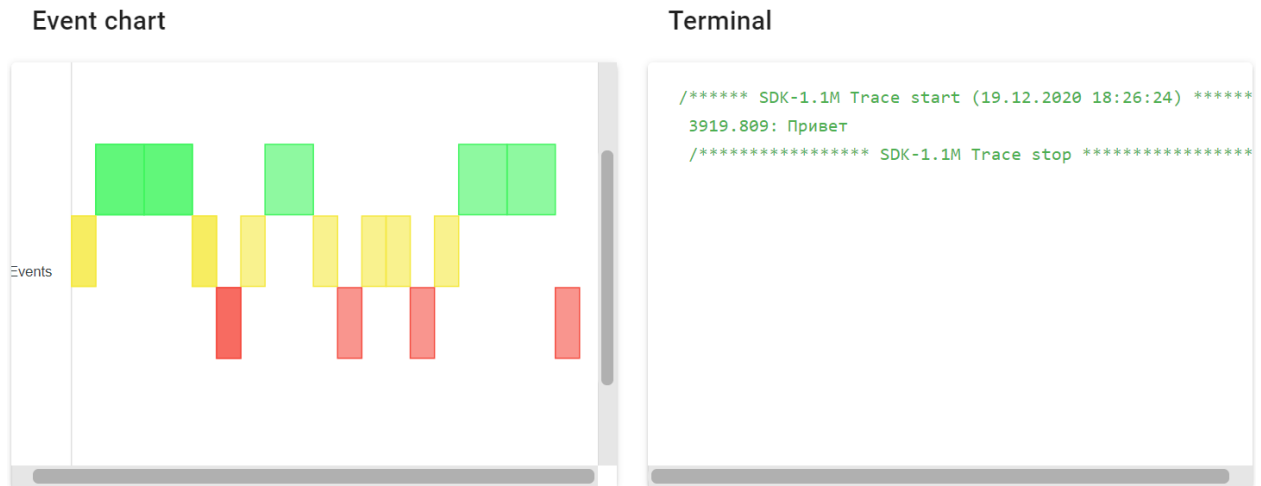


Рис. 3 – результат выполнения программы

## Вывод

В ходе данной лабораторной работы были изучены основные возможности виртуального стенда *cLab*. Также была изучена электрическая принципиальная схема SDK-1.1M, с помощью которой были определены необходимые пины. Была реализована программа передачи сообщения “Привет” с помощью светодиодов стенда SDK-1.1M.