Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

Лабораторная работа №2

по дисциплине

«**Встроенные системы**»

Выполнили: студенты группы P3212

Бригада №4

Сараев Владислав Витальевич

Земнухов Владимир Сергеевич

Федоров Сергей Денисович

Кульбако Артемий Юрьевич

Преподаватель:

Ключев Владислав Аркадьевич

Санкт-Петербург

2020 г.

**Задание:**

Разработать и реализовать драйверы GPIO (управление светодиодными индикаторами и обработка нажатий кнопки контроллера SDK-1.1М). Написать программу с использованием разработанных драйверов по алгоритму, соответствующему варианту задания.

Вариант 4:

Реализовать двоичный двухразрядный счётчик на светодиодах, с возможностью вычитания. Быстрое нажатие кнопки должно прибавлять 1 к отображаемому на диодах двоичному числу, по переполнению счётчика должна отображаться простая анимация: мигание обоими светодиодами, затем количество миганий зелёным светодиодом, равное количеству переполнений. Долгое нажатие кнопки должно вычитать 1 от отображаемого на диодах двоичного числа, если происходит вычитание из нуля – уменьшается на 1 количество переполнений, и отображается анимация, аналогичная анимации при переполнении.

**Инструментарий:**

**STM32CubeIDE –** это продвинутая платформа разработки C / C ++ с функциями настройки периферии, генерации кода, компиляции кода и отладки для микроконтроллеров и микропроцессоров STM32. Он основан на инфраструктуре ECLIPSE ™ / CDT и наборе инструментов GCC для разработки и GDB для отладки. Это позволяет интегрировать сотни существующих плагинов, которые дополняют функции ECLIPSE ™ IDE.

**Zadig –** программа для установки USB-драйверов.

**Исходный код** - <https://github.com/lmtspbru/SDK-1.1M/tree/master/SDK_LED>

**Отрывок, производящий манипуляции с внешними устройствами –**

**int COUNTER = 0;**

**int OVERFLOW = 0;**

**void change\_overflow(int delta){**

**if (OVERFLOW + delta < 0){**

**OVERFLOW = 0;**

**} else {**

**OVERFLOW += delta;**

**}**

**void display\_overflow(){**

**HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13, 1);**

**HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_14, 1);**

**HAL\_Delay(1000);**

**HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13, 0);**

**HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_14, 0);**

**HAL\_Delay(500);**

**for (int i = 0; i < OVERFLOW; i++){**

**HAL\_GPIO\_TogglePin(GPIOD, GPIO\_PIN\_13);**

**HAL\_Delay(500);**

**HAL\_GPIO\_TogglePin(GPIOD, GPIO\_PIN\_13);**

**HAL\_Delay(500);**

**}**

**}**

**void display\_counter(){**

**switch(COUNTER){**

**case 0: {**

**HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13, 0);**

**HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_14, 0);**

**break;**

**}**

**case 1: {**

**HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13, 1);**

**HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_14, 0);**

**break;**

**}**

**case 2: {**

**HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13, 0);**

**HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_14, 1);**

**break;**

**}**

**case 3: {**

**HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13, 1);**

**HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_14, 1);**

**break;**

**}**

**}**

**void change\_counter(int delta){**

**COUNTER += delta;**

**if (COUNTER > 3){**

**change\_overflow(1);**

**display\_overflow();**

**COUNTER = 0;**

**} else if (COUNTER < 0){**

**change\_overflow(-1);**

**display\_overflow();**

**COUNTER = 3;**

**}**

**HAL\_Delay(300);**

**}**

**int is\_pressed(){**

**return (int) HAL\_GPIO\_ReadPin(GPIOC, GPIO\_PIN\_15) ? 0 : 1;**

**/\* 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();**

**/\* USER CODE BEGIN 2 \***

**/\* USER CODE END 2 \*/**

**/\* Infinite loop \*/**

**/\* USER CODE BEGIN WHILE \*/**

**while (1)**

**{**

**/\* USER CODE END WHILE \*/**

**/\* USER CODE BEGIN 3 \*/**

**if (is\_pressed()) {**

**HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_15, 1);**

**HAL\_Delay(250);**

**HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_15, 0);**

**if (is\_pressed()){**

**change\_counter(-1);**

**} else {**

**change\_counter(1 );**

**}**

**display\_counter();**

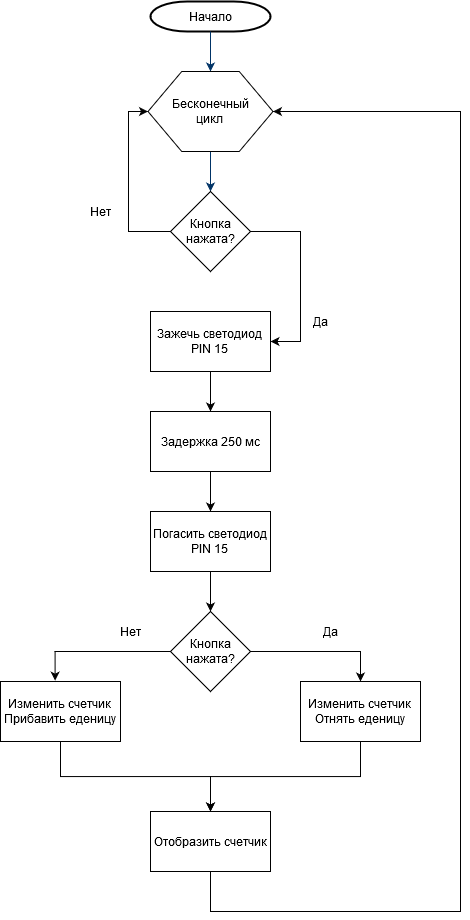
**}**

**}**

**/\* USER CODE END 3 \*/**

**}**

**Блок схема:**

****

**Выводы:** в ходе выполнения данной лабораторной работы нами был запущен и модифицирован (исходя из варианта) учебный проект в STM32CubeIDE. Данная работа показала нам наглядность взаимодействия с системой, через внешний устройства, а также реализацию логики взаимодействия.