

ГУАП

КАФЕДРА № 41

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

доц., канд. техн. наук		О.А. Кононов
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

по курсу: ОСНОВЫ МИКРОПРОЦЕССОРНОЙ ТЕХНИКИ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	4711		Хасанов Б.Р.
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2020

Цель работы

Изучить управление светодиодами по циклу на примере процессора STM32F407VG и написать программу, которая будет заставлять мигать светодиод.

1 Теоретические сведения

Схема подключения четырёх светодиодов на плате STM32F4Discovery представлена на рисунке 1.1

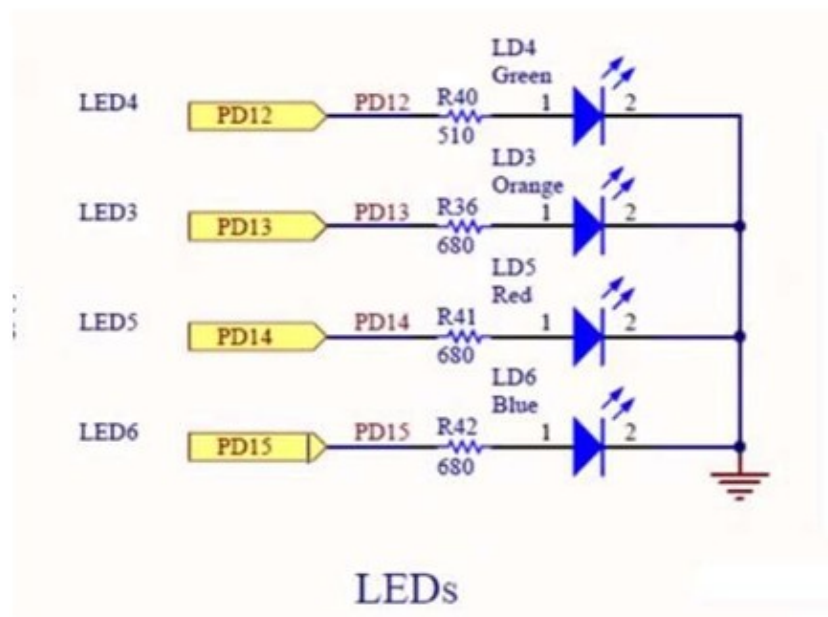


Рисунок 1.1 - Схема подключения светодиодов на плате STM32F4 Discovery

Как видно, диоды находятся на 12, 13, 14, 15 выводах порта D. GPIO (General Purpose Input/Output) - самый простой и примитивный способ организации работы с внешними устройствами. Порты могут работать в двух режимах: вход (прием сигнала) и выход (передача сигнала). Работают они только с логическими уровнями 0 (Low) или 1 (High). Например, если подключить к порту в режиме выхода светодиод (как указано на рисунке 1), то при подаче сигнала высокого уровня светодиод будет светиться, а при подаче низкого – потухнет. Если включить вывод в режим входа и

подключить к нему кнопку, соединённую с землёй, то с помощью микроконтроллера можно отслеживать ее состояние: нажатое или отжатое.

2 Программа циклического включения светодиодов

В рамках работы требовалось написать программы с использованием двух стандартных библиотек stm32: CMSIS и StdPeriph. Программы должны поочерёдного, циклически включать светодиоды RGBY – по кругу.

У обеих реализаций программы структура проекта одинаковая:

- В main.c содержится основной код программы
- В init.c содержится код функции инициализации, в данном случае это инициализация 12, 13, 14 и 15 портов ввода вывода
- Заголовочные файлы main.h и init.h содержат объявление функции и директивы предпроцессора, это нужно чтобы не захламлять файлы содержащие исходный текст программы, в данном случае это main.c и init.c

2.1 Программа написанная с использованием библиотеки CMSIS

Файл “main.c”

```
#include "main.h"
```

```
void Delay(volatile uint32_t nCount)
```

```
{  
    while(nCount--) {}  
}
```

```
int main(void){
```

```
    LEDs_ini(); //Инициализируем диоды
```

```
    while(1){ // Запускаем бесконечный цикл
```

```
        GREEN_ON(); // Включаем зелёный светодиод
```

```
        Delay(800000); // Ждём
```

```
        LEDS_OFF(); // Выключаем зелёный светодиод
```

```

        Delay(800000); // Ждём
        ORANGE_ON(); // Включаем оранжевый светодиод
        Delay(800000); // Ждём
        LEDS_OFF(); // Выключаем оранжевый светодиод
        Delay(800000); // Ждём
        RED_ON(); // Включаем красный светодиод
        Delay(800000); // Ждём
        LEDS_OFF(); // Выключаем красный светодиод
        Delay(800000); // Ждём
        BLUE_ON(); // Включаем синий светодиод
        Delay(800000); // Ждём
        LEDS_OFF(); // Выключаем синий светодиод
        Delay(800000); // Ждём
    }
}

```

Файл “main.h”

```
#include "init.h"
```

```
#define GREEN_ON() GPIOD → ODR=0x1000 // Генерируем высокий
уровень сигнала на выходе порта 12 портов ввода вывода линии D
```

```
#define ORANGE_ON() GPIOD → ODR=0x2000 // Генерируем высокий
уровень сигнала на выходе порта 13 портов ввода вывода линии D
```

```
#define RED_ON() GPIOD → ODR=0x4000 // Генерируем высокий
уровень сигнала на выходе порта 14 портов ввода вывода линии D
```

```
#define BLUE_ON() GPIOD → ODR=0x8000 // Генерируем высокий
уровень сигнала на выходе порта 15 портов ввода вывода линии D
```

```
#define LEDS_OFF() GPIOD → ODR=0 // Генерируем низкий уровень на
выходе всех портов ввода вывода линии D
```

```

#ifndef MAIN_H
#define MAIN_H
//
#endif

```

Файл “init.c”

```
#include "init.h"
```

```
void LEDs_ini(void)
```

```
{
```

```
    RCC->AHB1ENR |= RCC_AHB1ENR_GPIODEN; // Включаем
тактирование портов ввода-вывода линии D
```

```
    GPIOD → MODER=0x55000000; // Конфигурируем 12, 13, 14, 15
порты ввода-вывода линии D как порты выхода
```

```
}
```

Файл “init.h”

```
#include "stm32f4xx.h"
```

```
void LEDs_ini(void);
```

2.1 Программа написанная с использованием библиотеки StdPeriph

Файл “main.c”

```
#include "main.h"
```

```
void Delay(volatile uint32_t nCount)
```

```
{
```

```
    while(nCount--) {}
```

```
}
```

```

int main(void){
    LEDs_ini(); // Инициализируем диоды
    while(1){
        CHANGE_GREEN(); // Включаем зелёный светодиод
        Delay(800000); // Ждём
        CHANGE_GREEN(); // Выключаем зелёный светодиод
        Delay(800000); // Ждём
        CHANGE_ORANGE(); // Включаем оранжевый светодиод
        Delay(800000); // Ждём
        CHANGE_ORANGE(); // Выключаем оранжевый светодиод
        Delay(800000); // Ждём
        CHANGE_RED(); // Включаем красный светодиод
        Delay(800000); // Ждём
        CHANGE_RED(); // Выключаем красный светодиод
        Delay(800000); // Ждём
        CHANGE_BLUE(); // Включаем синий светодиод
        Delay(800000); // Ждём
        CHANGE_BLUE(); // Выключаем синий светодиод
        Delay(800000); // Ждём
    }
}

```

Файл “main.h”

```
#include "init.h"
```

```

#define CHANGE_GREEN() GPIO_ToggleBits(GPIOD, GPIO_Pin_12) //
Меняем состояние порта 12 портов ввода вывода линии D
#define CHANGE_ORANGE() GPIO_ToggleBits(GPIOD,
GPIO_Pin_13) // Меняем состояние порта 13 портов ввода вывода линии D

```

```
#define CHANGE_RED() GPIO_ToggleBits(GPIOD, GPIO_Pin_14) //
Меняем состояние порта 14 портов ввода вывода линии D
#define CHANGE_BLUE() GPIO_ToggleBits(GPIOD, GPIO_Pin_15) //
Меняем состояние порта 15 портов ввода вывода линии D
```

```
#ifndef MAIN_H
#define MAIN_H
//
#endif
```

Файл “init.c”

```
#include "init.h"
```

```
GPIO_InitTypeDef Init_LEDs; //Объявление структуры
```

```
void LEDs_ini()
{
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD,
ENABLE); // Включаем тактирование портов ввода-вывода линии D
    Init_LEDs.GPIO_Pin = GPIO_Pin_12 | GPIO_Pin_13 | GPIO_Pin_14
| GPIO_Pin_15; // Конфигурируем 12, 13, 14, 15 порты ввода-вывода линнии D
как порты выхода
    Init_LEDs.GPIO_Mode = GPIO_Mode_OUT; //Настройка выводов
на выход
    Init_LEDs.GPIO_Speed = GPIO_Speed_2MHz; //Скорость работы
порта
    Init_LEDs.GPIO_OType = GPIO_OType_PP; //Настройка PUSH-
PULL на выводе
    Init_LEDs.GPIO_PuPd = GPIO_PuPd_NOPULL;
    GPIO_Init(GPIOD, &Init_LEDs); //Инициализация структуры
```

```
}
```

Файл “init.h”

```
#include "stm32f4xx.h"
```

```
void LEDs_ini(void);
```

3 Результаты работы программы.

Вывод компилятора показан на рисунке 3.1

Работу программы можно увидеть по ссылке

<https://imgur.com/a/XgIy84F>

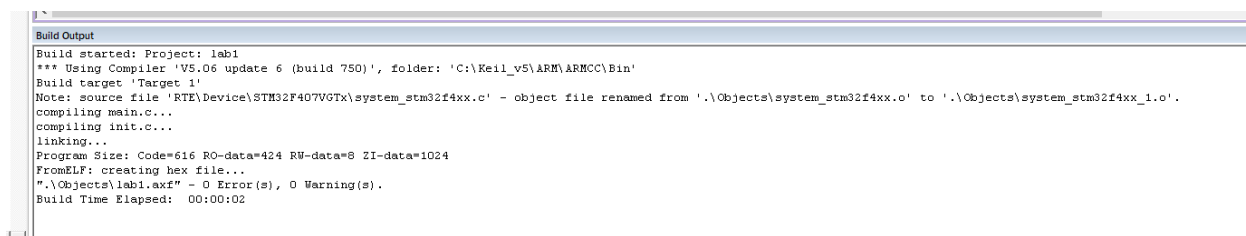


Рисунок 3.1 – Компиляция программы (build output)

Вывод

В рамках данной работы ознакомился с методом написания программ с помощью библиотек StdPeriph и CMSIS. Были написаны программы в которых организован код для поочерёдного, циклического включения светодиодов RGBY – по кругу. Осуществлена проверка работоспособности программы путём загрузки её на плату и сравнения результатов с заданием. Результат заданию соответствует