

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 №3

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 с помощью прерываний от таймера на примере процессора STM32F407VG и написать программу, которая будет поочередно, друг за другом включать и выключать светодиоды по прерыванию от таймера.

1 Теоретические сведения

Таймер - очень важное периферийное устройство в наборе внутренней аппаратуры микроконтроллера: он используется для самых разных задач от управления двигателями до генерации периодических событий в приложениях. Спецификации таймера доступны во всех руководствах STM32, и информации по таймерам очень много из-за их широких функциональных возможностей.

Микроконтроллеры STM32 имеют несколько таймеров, которые используются для генерации отсчёта времени, подсчёта импульсов, измерения длительности импульсов и периодов сигналов, генерации сигналов ШИМ, переключения внешних устройств. Однако в отличие от большинства 8-битных микроконтроллеров, в которых есть 2/3 таймера с ограниченным функционалом, таймеры STM32 весьма изощрённые и сложные. Это объясняет тот факт, что описание модулей таймеров занимает примерно 25% даташита любого STM32.

Таймеры микроконтроллеров STM32 можно поделить на следующие категории:

- Продвинутые (Advanced Timers)
- Общего назначения (General Purpose Timers, сокращенно GP)
- Базовые (Basic Timers)

General purpose (GP) таймеры имеют все функции стандартного модуля таймера-счетчика, они почти такие же, как в большинстве 8-битных микроконтроллеров. GP-таймеры могут использоваться для любых связанных с отсчетом времени и подсчетом событий целей, в том числе и

генерация ШИМ и захват импульсов. Обычно в микроконтроллере STM32 больше всего именно таймеров GP, чем таймеров других классов. Изучение GP-таймеров позволяет лучше понять базовые концепции.

У базовых таймеров нет каналов ввода/вывода для захвата входных импульсов (событий) и генерации ШИМ, поэтому такие таймеры используются только для целей отсчета времени. Базовые таймеры доступны только в старших моделях STM32, и это самый простой класс таймеров.

Продвинутые таймеры в основном похожи на GP-таймеры, но имеют дополнительные возможности по генерации комплементарных сигналов ШИМ, а также формировать сигнал торможения и фазы ШИМ с "мертвым" интервалом, предотвращающим сквозные токи силового моста (dead-time). Эти функции полезны для приложений, связанных с управлением двигателями, с силовыми инверторами, системами SMPS и другими задачами, связанными с источниками питания электроники и управлением мощностью. В большинстве микроконтроллеров STM32 есть как минимум один такой таймер. В старших моделях STM32, продвинутых таймеров может быть два.

Каждый таймер в микроконтроллере STM32 не зависит от других, и поэтому не использует никакие общие ресурсы. Единственное, что общее между таймерами, это типы регистров, принципы именования регистров и общий принцип работы. С очень малыми исключениями модули таймеров более или менее совместимы по всем семействам микроконтроллеров STM32. Например, Вы не найдете значительных различий в аппаратуре таймеров при миграции проекта с серии STM32F1xx на серию STM32F4xx. Это может быть совсем не так для других аппаратных устройств - портов GPIO, ADC, и т. п.

2 Практическая часть

В рамках работы была написана программа с помощью библиотеки CMSIS

Файл “main.c”

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

```
int led_switch_count = 0;
```

```
void TIM7_IRQHandler(void){ // Прерывание от 7-ого таймера
    CLEAR_BIT(TIM7->SR,  TIM_SR_UIF); // Очищаем флаг
прерывания 7-ого таймера
    led_switch_count++;
    switch(led_switch_count){ //На каждое нажатие зажигается свой
диод
        case 1:
            CLEAR_REG(GPIOD->ODR); //очищаем регистр
выходных значений диодов
            SET_BIT(GPIOD->ODR,      GPIO_ODR_ODR_12);
//Зажигаем оранжевый диод
            break;
        case 2:
            CLEAR_REG(GPIOD->ODR);
            SET_BIT(GPIOD->ODR,  GPIO_ODR_ODR_13); //
Зажигаем красный диод
            break;
        case 3:
            CLEAR_REG(GPIOD->ODR);
            SET_BIT(GPIOD->ODR,  GPIO_ODR_ODR_14); //
Зажигаем синий диод
```

```

        break;
    case 4:
        CLEAR_REG(GPIOD->ODR);
        SET_BIT(GPIOD->ODR,    GPIO_ODR_ODR_15);    //
Зажигаем зелёный диод
        led_switch_count = 0;
        break;;
    }
}

```

```

int main(void){
    LEDs_ini(); // Инициализация диодов
    TIM7_ini(); // Инициализация 7-ого таймера
    while(1);
}

```

Файл “main.h”

```

#include "init.h"

```

```

#ifndef MAIN_H
#define MAIN_H
//
#endif

```

Файл “init.c”

```

#include "init.h"

```

```

void LEDs_ini(void)

```

```

{
    SET_BIT(RCC->AHB1ENR, RCC_AHB1ENR_GPIODEN);
    SET_BIT(GPIOD->MODER,    GPIO_MODER_MODER12_0    |
GPIO_MODER_MODER13_0      |    GPIO_MODER_MODER14_0      |
GPIO_MODER_MODER15_0);
}

```

```

void TIM7_ini(void){
    NVIC_EnableIRQ(TIM7_IRQn);
    SET_BIT(RCC->APB1ENR, RCC_APB1ENR_TIM7EN);
    WRITE_REG(TIM7->PSC, 10000);
    WRITE_REG(TIM7->ARR, 1000);
    WRITE_REG(TIM7->DIER, TIM_DIER_UIE);
    SET_BIT(TIM7->CR1, TIM_CR1_CEN);
}

```

Файл “init.h”

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

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

```
void TIM7_ini(void);
```

3 Результаты работы программы.

Вывод компилятора показан на рисунке 3.1

Работу программы можно увидеть по ссылке

<https://imgur.com/a/9B6tORI>

```
Build started: Project: lab3
*** Using Compiler 'V5.06 update 6 (build 750)', folder: 'C:\Keil_v5\ARM\ARMCC\Bin'
Build target 'lab3'
compiling main.c...
linking...
Program Size: Code=976 RO-data=424 RW-data=4 ZI-data=1636
FromELF: creating hex file...
".\Objects\lab3.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:01
```

Рисунок 3.1 – Компиляция программы (build output)

Вывод

В рамках данной работы ознакомился с работой базового таймера на stm32f4, написал программу, которая по кругу, друг за другом, переключает диоды по прерыванию от таймера