

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 №6

РАБОТА С ТАЙМЕРАМИ. АППАРАТНЫЙ ШИМ

по курсу: ОСНОВЫ МИКРОПРОЦЕССОРНОЙ ТЕХНИКИ

РАБОТУ ВЫПОЛНИЛ

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

Санкт-Петербург 2020

Цель работы

Изучить работу с таймерами на примере аппаратного управления яркостью светодиодов на примере процессора STM32F407VG

1 Теоретические сведения

Модуляция – нелинейный электрический процесс, при котором параметры одного сигнала (несущего) изменяются при помощи другого сигнала (модулирующего, информационного). В связной технике широко применяется частотная, амплитудная, фазовая модуляция. В силовой электронике и микропроцессорной технике распространение получила широтно-импульсная модуляция.

При широтно-импульсной модуляции исходного сигнала неизменными остаются амплитуда, частота и фаза исходного сигнала. Изменению под действием информационного сигнала подвергается длительность (ширина) прямоугольного импульса. В англоязычной технической литературе обозначается аббревиатурой PWM – pulse-width modulation.

Сигнал, промодулированный по ширине импульса, формируется двумя способами:

- аналоговым;
- цифровым.

При аналоговом способе создания ШИМ-сигнала несущая в виде пилообразного или треугольного сигнала подается на инвертирующий вход компаратора, а информационный – на неинвертирующий. Если мгновенный уровень несущей выше модулирующего сигнала, то на выходе компаратора ноль, если ниже – единица. На выходе получается дискретный сигнал с частотой, соответствующей частоте несущего треугольника или пилы, и длиной импульса, пропорциональной уровню модулирующего напряжения.

В качестве примера на рисунке 1 приведена модуляция по ширине импульса треугольного сигнала линейно-возрастающим. Длительность

выходных импульсов пропорциональна уровню выходного сигнала.

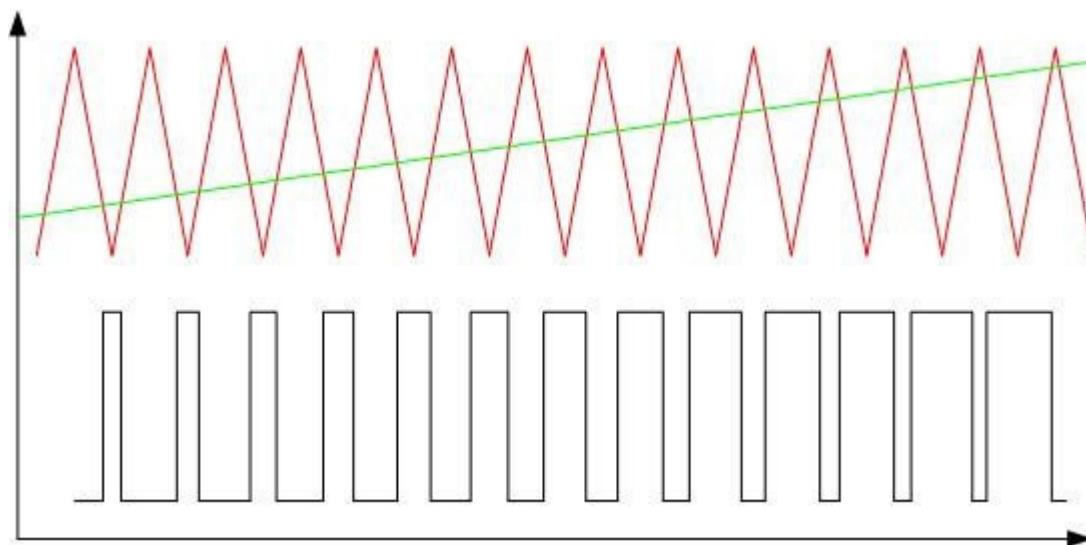


Рисунок 1 – ШИМ

Аналоговые ШИМ-контроллеры выпускаются и в виде готовых микросхем, внутри которых установлен компаратор и схема генерации несущей. Имеются входы для подключения внешних частото задающих элементов и подачи информационного сигнала. С выхода снимается сигнал, управляющий мощными внешними ключами. Также имеются входы для обратной связи – они нужны для поддержания установленных параметров регулирования. Такова, например, микросхема TL494. Для случаев, когда мощность потребителя относительно невелика, выпускаются ШИМ-контроллеры со встроенными ключами. На ток до 3 ампер рассчитан внутренний ключ микросхемы LM2596.

Цифровой способ осуществляется применением специализированных микросхем или микропроцессоров. Длина импульса регулируется внутренней программой. Во многих микроконтроллерах, включая популярные PIC и AVR, «на борту» имеется встроенный модуль для аппаратной реализации ШИМ, для получения PWM-сигнала надо активировать модуль и задать параметры его работы. Если такой модуль отсутствует, то ШИМ можно организовать чисто программным методом, это несложно. Этот способ дает более широкие возможности и предоставляет больше свободы за счёт гибкого использования выходов, но задействует большее количество

ресурсов контроллера.

2 Практическая часть

В рамках работы была написана программа с помощью библиотеки CMSIS

Файл “main.c”

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

```
int duty_cycle_counter;  
int duty_cycles_array [NUMBER_OF_DUTY_CYCLES] = {0x2, 0x4, 0x8, 0x10,  
0x20, 0x40, 0x80, 0xFF};
```

```
void Delay(volatile uint32_t nCount)  
{  
    while(nCount--) {}  
}
```

```
int main(void){  
    LEDs_ini();  
    TIM4_ini();  
  
    while(1)  
    {  
        for (duty_cycle_counter = 0; duty_cycle_counter <  
NUMBER_OF_DUTY_CYCLES; duty_cycle_counter++){ // Итерация по  
коэффициентам заполнения  
            // Установка скважности шим для всех 4-х диодов  
            WRITE_REG(TIM4->CCR1,  
duty_cycles_array[duty_cycle_counter]);  
            WRITE_REG(TIM4->CCR2,  
duty_cycles_array[duty_cycle_counter]);  
            WRITE_REG(TIM4->CCR3,  
duty_cycles_array[duty_cycle_counter]);  
            WRITE_REG(TIM4->CCR4,  
duty_cycles_array[duty_cycle_counter]);  
            Delay(0xF0000); // Ожидание, чтобы можно было увидеть  
текущую яркость  
        }  
    }  
}
```

Файл “main.h”

```
#include "init.h"

#define NUMBER_OF_DUTY_CYCLES 8

#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_1 |
GPIO_MODER_MODER13_1 | GPIO_MODER_MODER14_1 |
GPIO_MODER_MODER15_1);
    WRITE_REG(GPIOD->AFR[1], 0x22220000); // Подключение вывода 4-
х каналов таймера 4 к 4-ом диодам
}

void TIM4_ini(void){
    SET_BIT(RCC->APB1ENR, RCC_APB1ENR_TIM4EN);
    SET_BIT(TIM4->CCER, TIM_CCER_CC1E | TIM_CCER_CC2E |
TIM_CCER_CC3E | TIM_CCER_CC4E); // Разрешение использовать порты
ввода-вывода к которым подключены каналы таймера для ШИМа
    WRITE_REG(TIM4->ARR, 0xFF); // Установка предела счёта таймера
// Выставляем на всех каналах таймера 4 режим работы – ШИМ
    SET_BIT(TIM4->CCMR1, TIM_CCMR1_OC1M_1 |
TIM_CCMR1_OC1M_2);
    SET_BIT(TIM4->CCMR1, TIM_CCMR1_OC2M_1 |
TIM_CCMR1_OC2M_2);
    SET_BIT(TIM4->CCMR2, TIM_CCMR2_OC3M_1 |
TIM_CCMR2_OC3M_2);
    SET_BIT(TIM4->CCMR2, TIM_CCMR2_OC4M_1 |
TIM_CCMR2_OC4M_2);
    // Включаем таймер и заставляем его запоминать значение предела
счёта таймера
```

```
SET_BIT(TIM4->CR1, TIM_CR1_CEN | TIM_CR1_ARPE);
```

Файл “init.h”

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

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

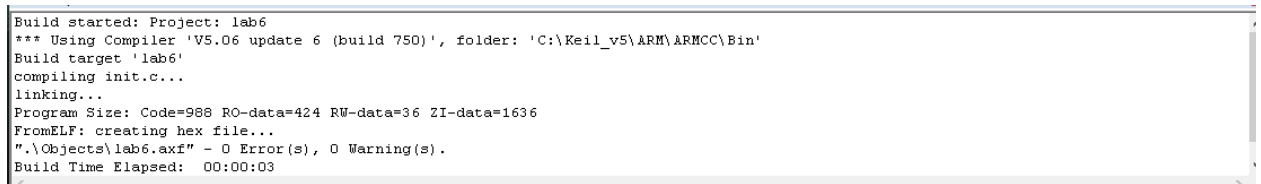
```
void TIM4_ini(void);
```

3 Результаты работы программы.

Вывод компилятора показан на рисунке 2

Работу программы можно увидеть по ссылке

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



```
Build started: Project: lab6
*** Using Compiler 'VS.06 update 6 (build 750)', folder: 'C:\Keil_v5\ARM\ARMCC\Bin'
Build target 'lab6'
compiling init.c...
linking...
Program Size: Code=988 RO-data=424 RW-data=36 ZI-data=1636
FromELF: creating hex file...
".\Objects\lab6.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:03
```

Рисунок 2 – Компиляция программы (build output)

Вывод

В рамках данной лабораторной работы мной был написан код на языке C, с использованием библиотеки CMSIS, для stm32f4, который меняет яркость светодиодов. При присваивании регистру сравнения 4-ого таймера нового значения меняется уровень скважности ШИМ, что приводит к изменению яркости светодиода. Смена яркости реализована с помощью широтно-импульсной модуляции