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 №4

по курсу: ОСНОВЫ МИКРОПРОЦЕССОРНОЙ ТЕХНИКИ

РАБОТУ ВЫПОЛНИЛ

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

Санкт-Петербург 2020

Цель работы

Изучить управление 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.

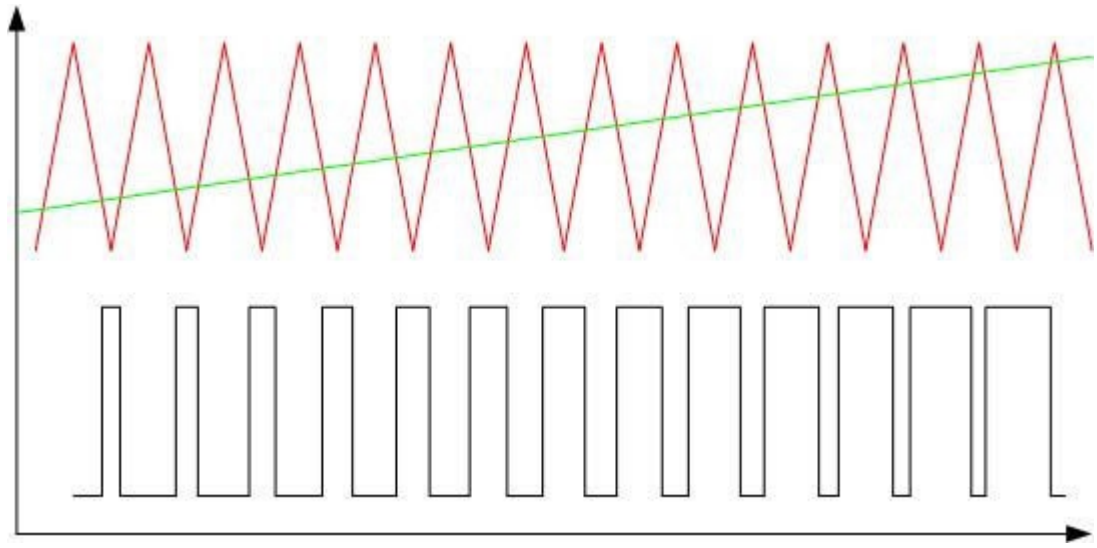

Рисунок 1 – Модуляция по ширине импульса

Аналоговые ШИМ-контроллеры выпускаются и в виде готовых микросхем, внутри которых установлен компаратор и схема генерации несущей. Имеются входы для подключения внешних частотозадающих 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_cycle_time;
int current_duty_cycle;
int impulse_time;
int pause_time;
int duty_cycles_array [NUMBER_OF_DUTY_CYCLES] = {0x2,0x4,0x8,0x10,0x20,0x40,0x80,0xFF};

int main(void){
	LEDs_ini();
	while(1){
		for(duty_cycle_counter = 0; duty_cycle_counter < NUMBER_OF_DUTY_CYCLES; duty_cycle_counter++){ // Итерация по коэффициентам заполнения
			current_duty_cycle = duty_cycles_array[duty_cycle_counter]; // Текущий коэффициент заполнения
			for(duty_cycle_time = 0; duty_cycle_time < PWM_PERIOD; duty_cycle_time++){ // Количество периодов
				for (impulse_time = 0; impulse_time <= current_duty_cycle; impulse_time++){
					// Включение всех доступных диодов на плате
					SET_BIT(GPIOD->ODR, GPIO_ODR_ODR_12);
					SET_BIT(GPIOD->ODR, GPIO_ODR_ODR_13);
					SET_BIT(GPIOD->ODR, GPIO_ODR_ODR_14);
					SET_BIT(GPIOD->ODR, GPIO_ODR_ODR_15);
				}
				for (pause_time = 0; pause_time <= duty_cycles_array[NUMBER_OF_DUTY_CYCLES - 1] - current_duty_cycle; pause_time++) CLEAR_REG(GPIOD->ODR);
			}
		}
	}
}
```

Файл “main.h”

```
#include "init.h"

#define NUMBER_OF_DUTY_CYCLES 8
#define PWM_PERIOD 700

#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);
}
```

Файл “init.h”

```
#include "stm32f4xx.h"

void LEDs_ini(void);
```

3 Результаты работы программы.

Вывод компилятора показан на рисунке 2

Работу программы можно увидеть по ссылке https://imgur.com/a/I2PXTNL

```
FromELF: creating hex file...
".\Objects\sample_project.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:01
Load "D:\\keil\\microproc\\lab4\\Objects\\sample_project.axf"
Erase Done.
Programming Done.
Verify OK.
Application running ...
Flash Load finished at 09:22:57
```

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 заставляет диоды менять яркость. Смена яркости реализована с помощью широтное-импульсной модуляции