МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

Институт компьютерных наук и технологий

Высшая школа искусственного интеллекта

Отчёт по лабораторной работе №2 по дисциплине

ПРОГРАММИРОВАНИЕ МИКРОКОНТРОЛЛЕРОВ

на тему:

Использование библиотек в Keil μVision5

Работу выполнил:

студент группы 5130201/20001

Тугай В.В.

Преподаватель:

Вербова Н. М.

Санкт-Петербург

2024

# Содержание

1. Цель
2. Постановка задачи
3. Алгоритм программы
4. Работа с осциллографом
5. Полученные результаты
6. Вывод
7. Приложение

**Цель**

Ознакомление с основными приемами работы с документацией при составлении программ для микроконтроллеров. Приобретение навыков работы с осциллографом и оценочной платой MCBSTM32F200 в качестве измерительного генератора.

**Постановка задачи**

Используя библиотеки Keil μVision5, разработать программу для микроконтроллера (МК) STM32F200, которая включает и выключает светодиод.

**Алгоритм программы**

В функции Delay выполняется цикл на 2 миллиона итераций, это позволяет установить промежуток времени между действиями над светодиодом. Изначально светодиод загорается и гаснет с одинаковым промежутком времени. В дальнейшем промежуток между затуханием и включением был увеличен в три раза.

**Работа с осциллографом**

Для измерения ширины положительного и отрицательного

уровня (TВКЛ) сигнала с ШИМ/прямоугольного сигнала был получен снимок (См. Рисунок 1). Размах сигнала = 1.6 \* 2 = 3.2В; Период сигнала = 1.6с; Частота сигнала = 1/1.6 = 0.625с.

A screen shot of a computer

Description automatically generated

Рисунок 1

Снимок нескольких тактов сигнала

Для Рисунка 1 ширина положительного уровня — 0.8 деления: 0.8 \*1с = 0.8с; ширина отрицательно уровня — 0.8 деления: 0.8 \*1с = 0.8с;

Широтно-импульсная модуляция (ШИМ) является типичным методом, используемым для управления мощностью аналоговых схем (или электрических приборов) с целью получения специфических периодов включения и выключения за счет изменения ширины импульса. Период ШИМ = 1.6c. Коэффициент заполнения = 0.8/1.6 = 1/2. Коэффициент заполнения равен 50%, тогда время состояния “включено” равно времени состояния “выключено”.

Для измерения времени нарастания формы волны был получен снимок (См. Рисунок 2). Время нарастания: время, затраченное сигналом на изменение от 10% до 90% его конечной величины во время перехода сигнала от низкого уровня к высокому уровню. Время нарастания = 1.9\*50нс = 95 нс.

A screen shot of a graph

Description automatically generated

Рисунок 2

Снимок нарастания формы волны такта сигнала

Для измерения времени спада формы волны был получен снимок (См. Рисунок 3). Время спада: время затраченное сигналом на изменение от 90% до 10% его конечной величины при переходе сигнала от высокого уровня к низкому уровню. Время нарастания = 3.8\*50нс = 190 нс.

A screen shot of a graph

Description automatically generated

Рисунок 3

Снимок спада формы волны такта сигнала

Для изменения содержимого счетчиков в программе таким образом, чтобы добиться скважности сигнала = ¼, в код программы были добавлены два вызова функции Delay после выключения светодиода. Был получен снимок (См. Рисунок 4)

A screen shot of a graph

Description automatically generated

Рисунок 4

Снимок нескольких тактов сигнала с измененной скважностью

**Полученные результаты**

Реализованная на языке C программа (см. Приложение) успешно выполняет поставленную задачу.

**Вывод**

В результате выполнения лабораторной работы был создан проект в среде Keil µVision, написана программа на языке C++ и проведен анализ сигнала с помощью асциллографа. Программа реализует последовательное включение и выключение диода G7 на микроконтроллере STM32F200. Была получена характеристика частоты сигнала, размах, период, время нарастания и спада. Было выяснено, что время нарастания сигнала примерно в два раза меньше времени спада сигнала.

**Приложение**

#include "stm32f2xx.h" // Device header

void delay ()

{

unsigned long i; // Counter for blinky delay

i=0;

for(i=0; i<2000000; i++){} // Delay

}

int main ()

{

RCC->AHB1ENR |= 1ul<<6; // Enable port G clocking

GPIOG->MODER = (GPIOG->MODER & ~1ul<<15) | 1ul<<14;

for (;;)

{

GPIOG->ODR |= 1ul<<7;

delay ();

GPIOG->ODR &= ~1ul<<7;

delay ();

}

}