Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Московский государственный технический университет имени Н.Э. Баумана»

(МГТУ им. Н.Э. Баумана)



## ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_\_\_\_Информатика и системы управления\_\_\_\_\_\_\_\_\_\_

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

**ОТЧЕТ**

**Лабораторная работа №3**

**По курсу**

**“Архитектура ЭВМ”**

Тема:

“Синхронизация микроконтроллера и управление таймерами”

Вариант 3

Студент Гасанзаде М.А.

*фамилия, имя, отчество*

Группа ИУ7-56

Москва, 2019

**Оглавление**

[Постановка задачи 3](#_Toc5101)

[Листинг 4](#_Toc5102)

[Вывод 7](#_Toc5103)

[Список литературы 7](#_Toc5103)

# Постановка задачи.

# Цель работы: изучение системы синхронизации микроконтроллера NXP LPC2368 и принципов функционирования таймеров общего назначения. В ходе работы студенту необходимо ознакомиться с теоретическим материалом, касающимся системы синхронизации и таймеров, разработать и отладить программу функционирования микроконтроллера NXP LPC2368 с использованием отладочных плат SK-LPC2368 и TM1638LED&KEY.

# Использованное оборудование: разработка и тестирование проводились в программе “Keil uVision”, которая предоставляет пользователю набор средств для написания и отладки кода программ для микроконтроллеров семейств ARM7, ARM9, Cortex M3 и других.

**Постановка задачи:** Устройство состоит из трех исполнительных механизмов и кнопки, подключенных к устройству управления на основе микроконтроллера NXP LPC2368. Разработать программу функционирования микроконтроллера, управляющего работой устройства и обеспечивающую заданную логику его работы:

Устройство управления кофеваркой, состоящее из мельницы, нагревателя и клапана кипятка. Программа функционирования:

a) нагрев воды;

b) при нажатии на кнопку: помол в течении 3,5 секунд;

c) заварка (2,5 секунды); d) отключение.

Частота внешнего генератора: 12 МГц.

Частота процессорного ядра: 32 МГц.

Частота синхронизации таймера: 8 МГц.

# Листинг

#include <LPC23xx.H>

#define STB 26 //Port1.26

#define CLK 27 //Port1.27

#define DIO 28 //Port1.28

void delay**(**unsigned int t**)** **{**

//Сбросить таймер

T0TC **=** 0x00000000**;**

//Установить задержку в мс в регистре совпадения MCR

T0MR0 **=** t**;**

//Запустить таймер

T0TCR **=** 0x00000001**;**

//Ожидаем окончания счета

**while** **(**T0TCR**&**0x1**)** **{};**

**}**

void tm1638\_sendbyte**(**unsigned int x**)** **{**

unsigned int i**;**

IODIR1 **|=** **(**1**<<**DIO**);**//Устанавливаем пин DIO на вывод

**for(**i **=** 0**;** i **<** 8**;** i**++)**

**{**

IOCLR1**=(**1**<<**CLK**);**//Сигнал CLK устанавливаем в 0

delay**(**1**);**//Задержка

**if** **(**x**&**1**)** **{**IOSET1**=(**1**<<**DIO**);}** //Устанавливаем значение на выходе DIO

**else** **{**IOCLR1**=(**1**<<**DIO**);}**

delay**(**1**);**//Задержка

x **>>=** 1**;**

IOSET1**=(**1**<<**CLK**);**//Сигнал CLK устанавливаем в 1

delay**(**2**);**

**}**

**}**

unsigned int tm1638\_receivebyte**()** **{**

unsigned int i**;**

unsigned int x**=**0**;**

IODIR1 **&=** **~(**1**<<**DIO**);**//Устанавливаем пин DIO на ввод

**for(**i **=** 0**;** i **<** 32**;** i**++)**

**{**

IOCLR1**=(**1**<<**CLK**);**//Сигнал CLK устанавливаем в 0

delay**(**1**);**//Задержка

**if** **(**IOPIN1**&(**1**<<**DIO**))** **{**

x **|=** **(**1**<<**i**);**

**}**

delay**(**1**);**//Задержка

IOSET1**=(**1**<<**CLK**);**//Сигнал CLK устанавливаем в 1

delay**(**2**);**

**}**

**return** x**;**

**}**

void tm1638\_sendcmd**(**unsigned int x**)**

**{**

//Устанавливаем пассивный высокий уровень сигнала STB

IOSET1**=(**1**<<**STB**);**

//Устанавливаем пины CLK,DIO,STB на вывод

IODIR1 **=** **(**1**<<**CLK**)|(**1**<<**DIO**)|(**1**<<**STB**);**

//Устанавливаем активный низкий уровень сигнала STB

IOCLR1**=(**1**<<**STB**);**

tm1638\_sendbyte**(**x**);**

**}**

void tm1638\_setadr**(**unsigned int adr**)** **{**

//Установить адрес регистра LED инидикации

tm1638\_sendcmd**(**0xC0**|**adr**);**

**}**

void tm1638\_init**()** **{**

unsigned int i**;**

//Разрешить работу индикации

tm1638\_sendcmd**(**0x88**);**

//Установить режим адресации: автоинкремент

tm1638\_sendcmd**(**0x40**);**

//Установить адрес регистра LED инидикации

tm1638\_setadr**(**0**);**

//Сбросить все

**for** **(**i**=**0**;**i**<=**0xf**;**i**++)**

tm1638\_sendbyte**(**0**);**

//Установить режим адресации: фиксированный

tm1638\_sendcmd**(**0x44**);**

**}**

void Timer0\_Init**(**void**){**

//Предделитель таймера = 12000

T0PR **=** 20000**;**

//Сбросить счетчик и делитель

T0TCR **=** 0x00000002**;**

//При совпадении останавливаем, сбрасываем таймер

T0MCR **=** 0x00000006**;**

//Регистр совпадения = 1000 (1 Гц)

T0MR0 **=** 1000**;**

**}**

int main **(**void**)** **{**

unsigned int flag**,** i**;**

Timer0\_Init**();** /\* Настроить таймер \*/

tm1638\_init**();**/\* Конфигурируем TM1638 \*/

**while** **(**1**)** **{** /\* Бесконечный цикл \*/

flag **=** 1**;**

// Put all lights off

tm1638\_setadr**(**1**);**

tm1638\_sendbyte**(**0**);**

tm1638\_setadr**(**3**);**

tm1638\_sendbyte**(**0**);**

tm1638\_setadr**(**5**);**

tm1638\_sendbyte**(**0**);**

tm1638\_sendcmd**(**0x46**);**

i **=** tm1638\_receivebyte**();**

**while** **(**i **!=** 0**)**

**{**

tm1638\_setadr**(**1**);**

tm1638\_sendbyte**(**1**);**

delay**(**0xfff**);**

tm1638\_setadr**(**1**);**

tm1638\_sendbyte**(**0**);**

tm1638\_setadr**(**3**);**

tm1638\_sendbyte**(**1**);**

delay**(**0xfff**);**

tm1638\_setadr**(**3**);**

tm1638\_sendbyte**(**0**);**

tm1638\_setadr**(**5**);**

tm1638\_sendbyte**(**1**);**

delay**(**0xfff**);**

tm1638\_setadr**(**5**);**

tm1638\_sendbyte**(**0**);**

tm1638\_sendcmd**(**0x46**);**

**}**

**}**

**}**

**Выводы**

Устройство функционирует в соответствии с поставленной задачей.

**Список используемой литературы**

1. EVM\_Lab4.pdf

2. Индивидуальны\_ задания 4.pdf