**Отчёт по лабораторной работе**

**Дисциплина: Архитектура ЭВМ**

**Работа №3. Синхронизация микроконтроллера и управление таймерами**

Вариант 8

Студент Жумабаев С.К.

Группа ИУ7-53

Преподаватель А.Ю. Попов

Москва 2018

**Практическая часть**

Цель работы – изучение системы синхронизации микроконтроллера NXP

LPC2368 и принципов функционирования таймеров общего назначения.

*Вариант 8.* Устройство управления метрономом, состоящее из двух блоков генерации звука и динамика. Генераторы выдают звуки, соответствующие слабым и сильным долям такта. В каждый момент работает только один генератор. При паузе между сигналами динамик отключен. Программа функционирования: трехдольный ритм 100 ударов в минуту. Длительность звукового сигнала (0.2 секунды). При нажатии на кнопку: отключение. Частота внешнего генератора: 12 МГц. Частота процессорного ядра: 60 МГц. Частота синхронизации таймера: 30 МГц.

**Листинг программы**

#include <LPC23xx.H> /\* Описание LPC23xx \*/

#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 = 12000;

//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 tick = 0;

unsigned int flag = 0;

unsigned int i;

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

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

while (1) {

i = 1;

tm1638\_sendcmd(0x46);

i = tm1638\_receivebyte();

if ( i != 0 ) {

for (i=1; i<6; i+=2){

tm1638\_setadr(i); //устанавливаем адрес

tm1638\_sendbyte(0); //шлем данные

}

tick = 0;

flag = 0;

} else {

tick %= 3;

if (flag) {

if (tick++ == 0) {

tm1638\_setadr(1);

tm1638\_sendbyte(0);

tm1638\_setadr(5);

tm1638\_sendbyte(0);

} else {

tm1638\_setadr(3);

tm1638\_sendbyte(0);

tm1638\_setadr(5);

tm1638\_sendbyte(0);

}

flag = 0;

} else {

if (tick == 0) {

tm1638\_setadr(1);

tm1638\_sendbyte(1);

tm1638\_setadr(5);

tm1638\_sendbyte(1);

} else {

tm1638\_setadr(3);

tm1638\_sendbyte(1);

tm1638\_setadr(5);

tm1638\_sendbyte(1);

}

flag = 1;

}

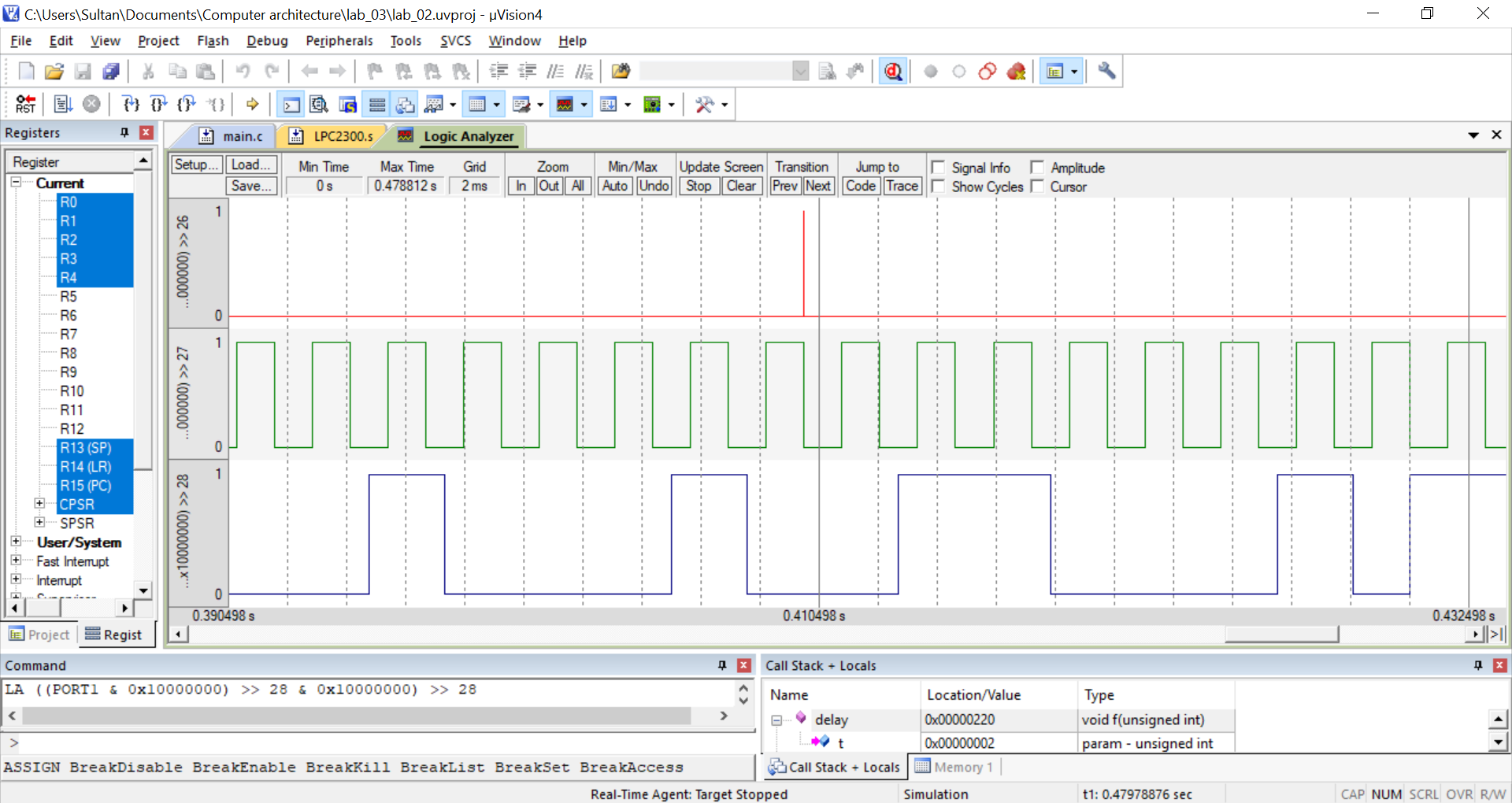
}

delay(20);

}

}

**Осциллограмма**

****

26 – STB

27 – CLK

28 – DIO

**Вычислительная часть**

Fcco = 2 \* M \* Fin / N, где M = 7, N = 1, Fin = 30

Fcpu = Fcco / (CCLKSEL(7:0) + 1), где CCLKSEL(7:0) = 7

Fcco = 420

Fcpu = 60

**Итог**

Программа корректно выполняет требуемые функции. Кнопка включает и выключает метроном, который настроен на такт ¾. Таким образом, отображается сильная или слабая доля такта, а так же включается и выключается динамик метронома.