

# **Отчёт по лабораторной работе**

**Дисциплина: Архитектура ЭВМ**

## **Работа №4. Синхронизация микроконтроллера и управление таймерами**

**Вариант 8**

Студент	Керимов А. Ш.
Группа	ИУ7-54Б

Преподаватель	А.Ю. Попов
---------------	------------

Москва 2019

## Введение

**Цель работы** — изучение архитектуры микроконтроллеров ARM7 TDMI и средств проектирования и отладки цифровых устройств на их основе.

**Индивидуальное задание.** 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. 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.

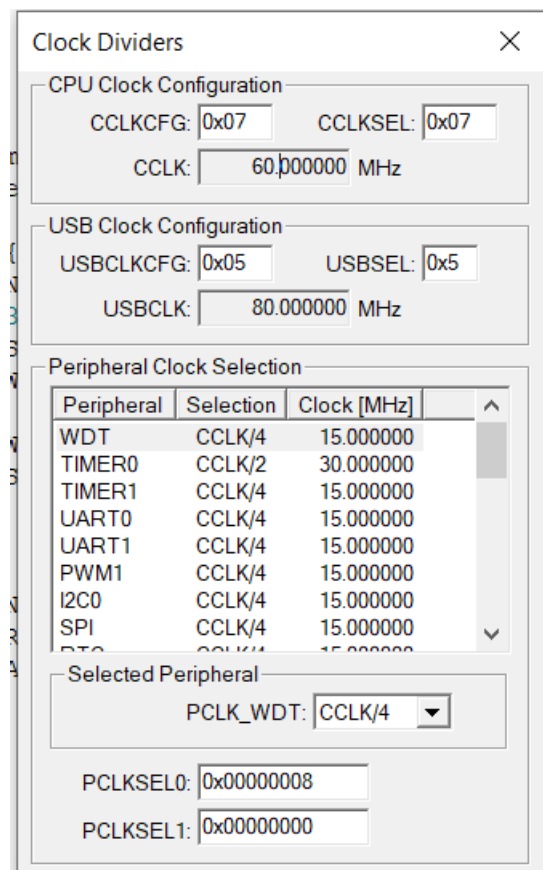
Частота синхронизации таймера: 30 МГц.

## Настройка таймера

Для корректной работы используемого в программе таймера были проведены необходимые расчеты параметров. В соответствии с заданными начальными условиями (PLL = 12, CCLK = 60, Timer0 = 30) были получены следующие результаты:

M = 40, N = 2; CCLKSEL = 8; USB Divider = 6;

Настройки таймера можно видеть на рисунках ниже:



[-] PLL Configuration Register (PLLCFG)	
MSEL: PLL Multiplier Selection	0x0028
NSEL: PLL Divider Selection	0x0002
[-] CPU Clock Configuration Register (CCLKCFG)	
CCLKSEL: Divide Value for CPU Clock from PLL	0x0008
[-] USB Clock Configuration Register (USBCLKCFG)	
USBSEL: Divide Value for USB Clock from PLL	0x06

## Листинг программы

```
#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_setaddr(unsigned int adr) {
```

```

        //Установить адрес регистра LED индикации
        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 индикации
    tm1638_setaddr(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;
}

void setbyte(unsigned int addr, unsigned int byte) {
    tm1638_setaddr(addr);
    tm1638_sendbyte(byte);
}

#define DYNAMIC 1
#define STRONG 3
#define WEAK 5

int main (void) {
    unsigned n, i;

    Timer0_Init();
    tm1638_init();

    n = 1;
    for (;;) {
        tm1638_sendcmd(0x46);
        i = tm1638_receivebyte();

        if (i == 0) {
            setbyte(DYNAMIC, 1);
            if (n++ % 3) {
                setbyte(STRONG, 0);
                setbyte(WEAK, 1);
            } else {
                setbyte(WEAK, 0);
                setbyte(STRONG, 1);
            }
        }
        else {
            setbyte(DYNAMIC, 0);
            setbyte(STRONG, 0);
            setbyte(WEAK, 0);
        }
    }
}

```

```
        }  
        delay(200);  
    }  
}
```

## **Тестирование**

Программа была протестирована на микроконтроллере на базе микросхемы TM1638. В качестве индикаторов использовались первые три светодиода на плате. Программа функционировала в полном соответствии с индивидуальным заданием.

## **Заключение**

В результате выполнения лабораторной работы были изучены системы синхронизации микроконтроллера NXP LPC2368 и принципы функционирования таймеров общего назначения; в соответствии с индивидуальным заданием была разработана программа, моделирующая запуск двигателя. Программа была протестирована на микроконтроллере на базе микросхемы TM1638