



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

Архитектура ЭВМ

Лабораторная работа №3

Синхронизация микроконтроллера и управление таймерами

Выполнила:

Овчинникова А. П.

Группа:

ИУ7-55Б

Вариант 16

Москва, 2019 г.

Цель работы – изучение системы синхронизации микроконтроллера NXPLPC2368 и принципов функционирования таймеров общего назначения.

Задание.

Устройство состоит из трех исполнительных механизмов и кнопки, подключенных к устройству управления на основе микроконтроллера NXPLPC2368, подключенного к внешнему генератору синхросигнала. Разработать программу функционирования микроконтроллера, управляющего работой устройства и обеспечивающую заданную логику его работы при заданных параметрах частоты генератора, частоты процессорного ядра, частоты синхронизации периферии.

Устройство управления светофором. Программа функционирования:

- а) При нажатии на кнопку: зеленый сигнал работает в течении 2-х секунд.
- б) Одновременно работают зеленый и желтый сигнал (1 секунда).
- с) Работает красный сигнал (10 секунд).

Частота внешнего генератора: 12 МГц.

Частота процессорного ядра: 24 МГц.

Частота синхронизации таймера: 12 МГц.

Вычислительная часть.

По условию $F_{cpu} = 24$ МГц.

$$F_{cpu} = \frac{F_{cco}}{CCLKSEL(7:0) + 1}$$

Пусть $CCLKSEL(7:0) = 11$. Тогда $F_{cco} = 24 \cdot 12 = 288$

$$F_{cco} = \frac{2 \cdot M \cdot F_{in}}{N}$$

По условию $F_{in} = 12$ МГц. Тогда $\frac{M}{N} = \frac{F_{cco}}{2F_{in}} = \frac{288}{24} = 12$. Возьмем $M = 12$, $N = 1$.

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);} //Сбрасываем значение на
```

```
        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 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_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

```

```

TOMCR = 0x00000006;
//Регистр совпадения = 1000 (1 Гц)
TOMR0 = 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();

        // 1 - green, 3 - yellow, 5 - red

        // Проверка нажатия кнопки
        tm1638_sendcmd(0x46);
        i = tm1638_receivebyte();

        if (i == 1)
        {
            tm1638_setadr(1); // green
            tm1638_sendbyte(1);

```

```
        delay(1000);

        tm1638_setadr(1); // green
        tm1638_sendbyte(0);
    }

    tm1638_setadr(1); // green
    tm1638_sendbyte(1);
    tm1638_setadr(3); // yellow
    tm1638_sendbyte(1);
    delay(2000);

    tm1638_setadr(1);
    tm1638_sendbyte(0);
    tm1638_setadr(3);
    tm1638_sendbyte(0);

    tm1638_setadr(5); // red
    tm1638_sendbyte(1);
    delay(10000);
    tm1638_setadr(5); // red
    tm1638_sendbyte(0);

    // delay(0xffff);
}
}
```

Вывод.

Таким образом, программа корректно выполняет требуемые функции.