|  |  |
| --- | --- |
|  | **Министерство образования и науки 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)** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

Отчет по лабораторной работе №3 по курсу: «Архитектура ЭВМ»

По теме: «Синхронизация микроконтроллера и управление таймерами»

Студент: Аминов Т.С

Группа: ИУ7-55Б

Преподаватель:

Попов А.Ю.

*Москва, 2019 г.*

**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.

**Задание.**

Вариант 1. Устройство прогрева двигателя внутреннего сгорания, включающее клапан подачи горючей смеси, устройство зажигания, стартер.

Программа функционирования:

1. Пуск стартера, кратковременное открытие клапана горючей смеси на 0.05 секунды и зажигание при закрытом клапане оставшееся время такта (частота: 10 Гц);
2. при нажатии на кнопку: отключение стартера;
3. через 5 секунд после нажатия – отключение зажигания и закрытие клапана;
4. отключение.

Частота внешнего генератора: 12 МГц.

Частота процессорного ядра: 24 МГц.

Частота синхронизации таймера: 12 МГц.

**Расчет параметров таймера**

Для корректной работы используемого в программе таймера были проведены необходимые расчеты параметров. По условию задания необходимо произвести задержки длительностью 5 с и 0.05 с. Частота процессорного ядра должна соответствовать 24 МГц.

M = 40 = 0x0028 hex

N = 2 = 0x0002 hex

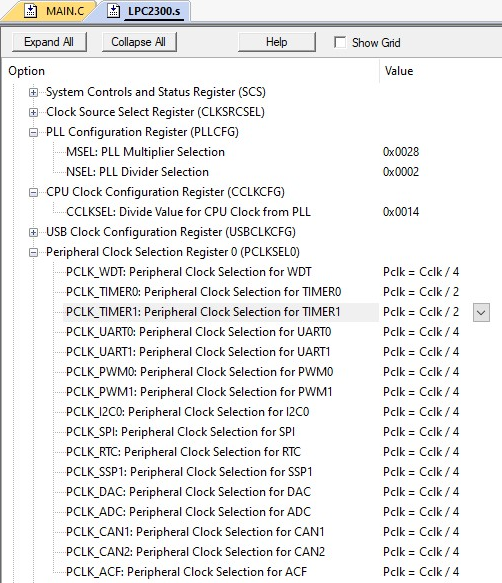
CCLKSEL = 20 = 0x0014 hex

𝐹𝑖𝑛 = 12 МГц по условию задания, тогда 𝐹𝑐𝑐𝑜 = = 480 МГц

𝐹𝑐𝑝𝑢 = = 24 МГц

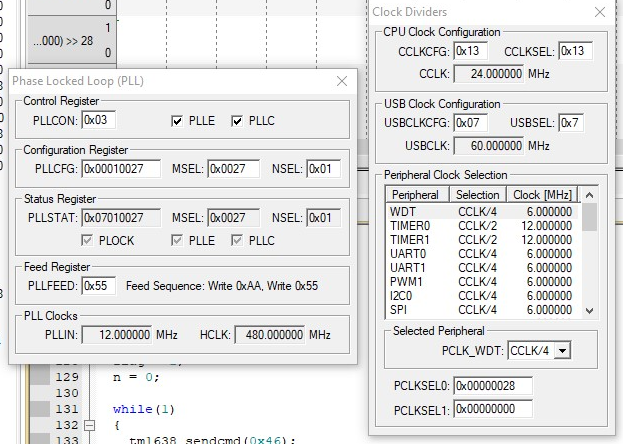
Частота синхронизации таймера = 12 МГц =

**Настройки в программе**



CCLKSEL = 20 = 0x0014 hex, т.к. необходимо вводить значение уже на единицу больше. Получено опытным путем.

**Проверка вычислительных расчетов**

****

PLLIN = 12 МГц, HCLK = 480 МГц, CCLK = 24 МГц,

Timer0 = Timer1 = 12 МГц

**Листинг программы:**

#include <LPC23xx.H>

#define BIT\_BTTN (1<<29)

#define STB 26 //Port1.26

#define CLK 27 //Port1.27

#define DIO 28 //Port1.28

#define STATE\_OFF 0

#define STATE\_ON 1

#define DIOD\_VALVE 1

#define DIOD\_IGNITE 2

#define DIOD\_STARTER 3

void TimerDelay(void)

{

T0TC = 0x00000000;

T0TCR = 0x00000001;

while (T0TCR&0x1);

}

void delay(unsigned int count)

{

unsigned int i;

for (i=0;i<count;i++){}

}

void Timer0\_Init(void)

{

//Предделитель таймера = 600

T0PR = 600;

//Сбросить счетчик и делитель

T0TCR = 0x00000002;

//При совпадении останавливаем, сбрасываем таймер

T0MCR = 0x00000006;

//Регистр совпадения = 1000 (1 Гц)

T0MR0 = 100;

}

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);

if (IOPIN1 & (1 << DIO))

{

x |= (1 << i);

}

delay(1);

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 SetDiod(unsigned diodNo, int state)

{

diodNo \*= 2;

diodNo--;

tm1638\_setadr(diodNo); //устанавливаем адрес

tm1638\_sendbyte(state); //шлем данные

}

int main (void)

{

unsigned i;

unsigned flag = 0;

unsigned j = 0;

unsigned temp = 0;

tm1638\_init();

Timer0\_Init();

SetDiod( DIOD\_VALVE, STATE\_OFF );

SetDiod( DIOD\_IGNITE, STATE\_OFF );

SetDiod( DIOD\_STARTER, STATE\_OFF );

while (1)

{

switch(flag)

{

case 0:

// Включение стартера и попеременное включение и выключение диода зажигания и клапана

SetDiod(DIOD\_STARTER, STATE\_ON );

SetDiod(DIOD\_VALVE, STATE\_ON );

TimerDelay();

SetDiod(DIOD\_VALVE, STATE\_OFF );

SetDiod(DIOD\_IGNITE, STATE\_ON );

TimerDelay();

SetDiod(DIOD\_IGNITE, STATE\_OFF );

// Обработка нажатия кнопки

tm1638\_sendcmd(0x46);

i = tm1638\_receivebyte();

if (i == 1)

{

SetDiod(DIOD\_STARTER, STATE\_OFF ); // Отключение стартера

temp = 1;

}

if (temp == 1) // Cчетчик после нажатия кнопки

j++;

if (j == 50)

{

SetDiod(DIOD\_IGNITE, STATE\_OFF ); // Отключение зажигания с закрытием клапана

SetDiod(DIOD\_VALVE, STATE\_OFF );

flag = 1;

}

case 1:

// Полное отключение

SetDiod(DIOD\_IGNITE, STATE\_OFF );

SetDiod(DIOD\_VALVE, STATE\_OFF );

SetDiod(DIOD\_STARTER, STATE\_OFF );

}

}

}

**Вывод.**

Программа была успешно протестирована, функционирует на плате в соответствии указанному заданию. Работа светодиодов правильно имитирует устройство работы стиральной машины. Как следствие выполнения практического задания, изучены системы синхронизации микроконтроллера NXP LPC2368 и принципов функционирования таймеров общего назначения.