*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана***  ***(национальный исследовательский университет)»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

**Отчет**

**по лабораторной работе №2**

**Вариант 6**

**Дисциплина:**

Организация ЭВМ и систем

**Название лабораторной работы:**

Изучение средств ввода и вывода алфавитно-цифровой информации и индикации с использованием микроконтроллеров ARM7

Студент гр. ИУ6-72б  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.С. Бурлаков**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2019

# ВВедение

## Цель работы

Изучение средств управления LED индикаторами и клавиатурными матрицами на базе микросхемы TM1638, а также изучение средств внутрисхемной отладки программ микроконтроллеров ARM7 TDMI. В ходе работы студенту необходимо ознакомиться с особенностями функционирования средств индикации и кнопочных клавиатур на основе микросхемы TM1638, ознакомиться со средствами внутрисхемной отладки программ, разработать и отладить программу индикации и сканирования клавиатуры с использованием отладочной платы SK-LPC2368 и платы индикации TM1638LED&KEY

# основная часть

**Индивидуальное задание**

6. Устройство управления летательным аппаратом, включающее двигатель, приводы подкрылков и хвостового оперения. Программа функционирования:

a) разгон и поднятие подкрылков (взлет);

b) при нажатии на кнопку: поворот и отключение всех приводов.

**Листинг программы функционирования микроконтроллера**

/\* Пример 1.

Управление портами ввода/вывода.

Для правильного связывания модулей отметить

Options -> Linker -> Use Memory Layout from Target Dialog

\*/

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

#define STB 26 //Port1.26

#define CLK 27 //Port1.27

#define DIO 28 //Port1.28

int delay\_value = 0xff;

void delay(unsigned int count) {

unsigned int i;

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

}

void tm1638\_sendbyte(unsigned int x) {

// ДИАГРАММА ПЕРЕДАЧИ БАЙТА

//

// STB ----| |-----

// |-------------------...-------|

//

//

// CLK ----| |-----| |-...-| |-----

// |-----| |-----| |-----|

//

//

// |-----|-----|----...----|-----|

// DIO -------| D0 | D1 | D2 D6 | D7 |--

// |-----|-----|----...----|-----|

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(delay\_value);//Задержка

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

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

delay(delay\_value);//Задержка

x >>= 1;

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

delay(delay\_value \* 2);

}

}

unsigned int tm1638\_receivebyte() {

// ДИАГРАММА ПРИЕМА БАЙТА

//

// STB ----| |-----

// |-------------------...-------|

//

//

// CLK ----| |-----| |-...-| |-----

// |-----| |-----| |-----|

//

//

// |-----|-----|----...----|-----|

// DIO -------| D0 | D1 | D2 D6 | D7 |--

// |-----|-----|----...----|-----|

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(delay\_value);//Задержка

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

x |= (1<<i);

}

delay(delay\_value);//Задержка

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

delay(delay\_value \* 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(0x8F);

//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);

}

int main (void) {

unsigned int i, vzlet = 1;

tm1638\_init();

tm1638\_setadr(1); // Поворота нет

tm1638\_sendbyte(0);

tm1638\_setadr(3); // Двигатель запущен

tm1638\_sendbyte(1);

tm1638\_setadr(5); // Подкрылки подняты

tm1638\_sendbyte(1);

while (1) { /\* Бесконечный цикл \*/

tm1638\_sendcmd(0x46);

i = tm1638\_receivebyte();

if (i & 0x01){ // Нажатая кнопка

tm1638\_setadr(1); // Поворот есть

tm1638\_sendbyte(1);

tm1638\_setadr(3); // Двигатель есть

tm1638\_sendbyte(1);

tm1638\_setadr(5); // Подкрылки опущены

tm1638\_sendbyte(0);

vzlet = 0;

}

else if (vzlet != 1){

tm1638\_setadr(1); // Поворота нет

tm1638\_sendbyte(0);

tm1638\_setadr(3); // Двигатель есть

tm1638\_sendbyte(1);

tm1638\_setadr(5); // Подкрылки опущены

tm1638\_sendbyte(0);

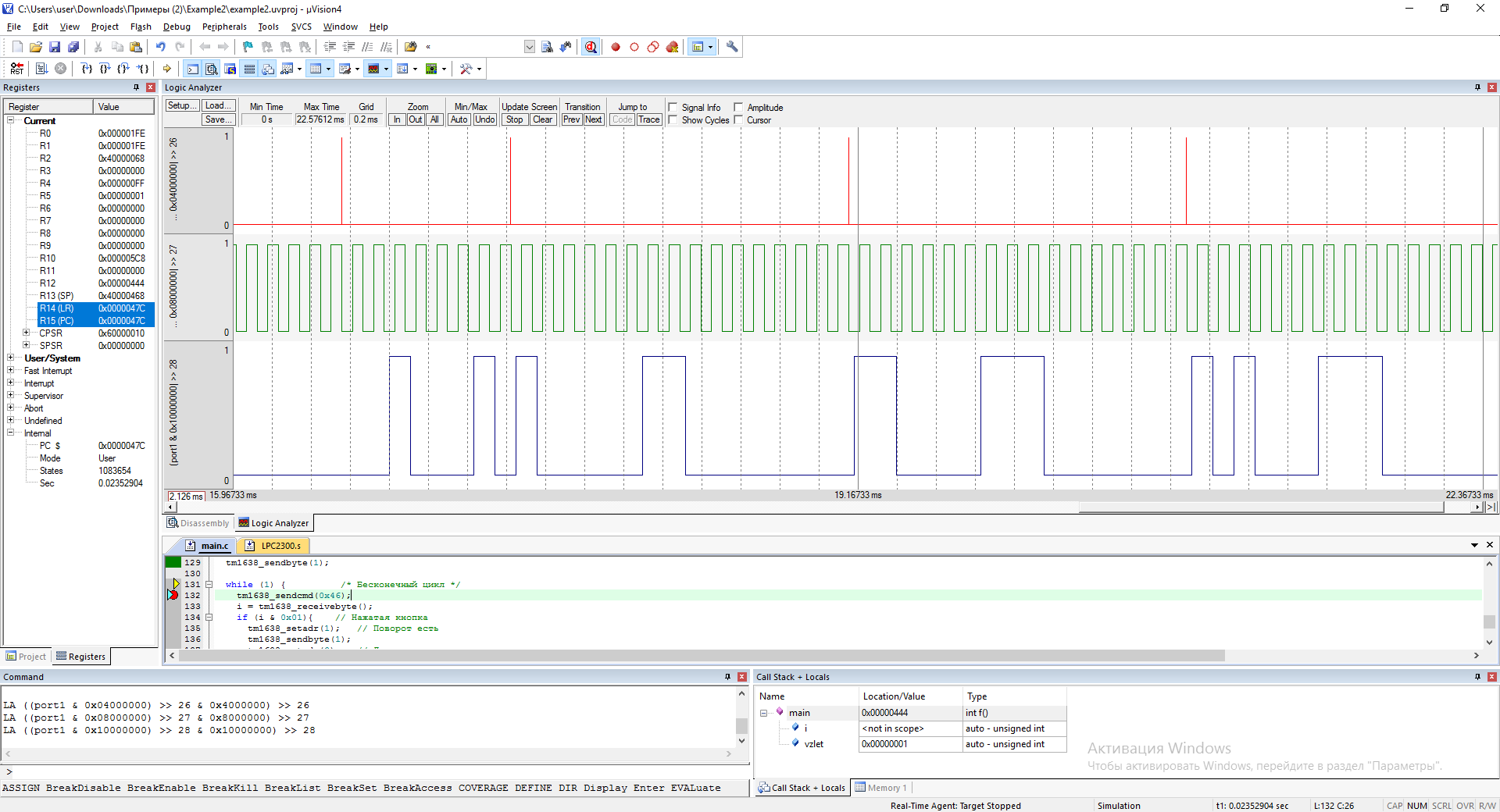
}

delay(delay\_value);

}

}

**Осциллограмма работы программы**



**Результаты тестирования программы**

Согласно заданию программа должна иметь 3 состояния:

1. Взлет – подкрылки, разгон
2. Поворот – разгон, поворот
3. Полет - разгон

Изначально программа находится в первом состоянии, после нажатия кнопки переходить во второе, после отжатия в третье. Далее при нажатии кнопки программа находится во втором состоянии, при отжатии в третьем.

Во время тестирования на плате:

1. В первом состоянии горят второй и третий светодиоды;
2. Во втором состоянии горят первый и второй светодиоды;
3. В третьем состоянии горит второй светодиод;
4. Переход во второе состояние происходит по нажатию кнопки;
5. Переход в третье состояние происходит по отжатию кнопки.

# Заключение

В ходе данной лабораторной работы была разработана программа функционирования устройства летательного аппарата. Согласно осциллограмме и результатам тестирования программы на плате она является полностью работоспособной.