**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕНЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.Шухова»**

**(БГТУ им. В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №1

Дисциплина: ЭВМ и периферийные устройства

Тема: «Изучение принципов программного управления внешними устройствами на примере вывода информации на цифровой индикатор»

Выполнил студент группы ВТ - 31

Проверил: Шамраев А.А.

Белгород 2019

**Цель работы**: изучить принципы функционирования и возможности программного управления цифровым индикатором, разработать алгоритм и программу для вывода информации на цифровой индикатор.

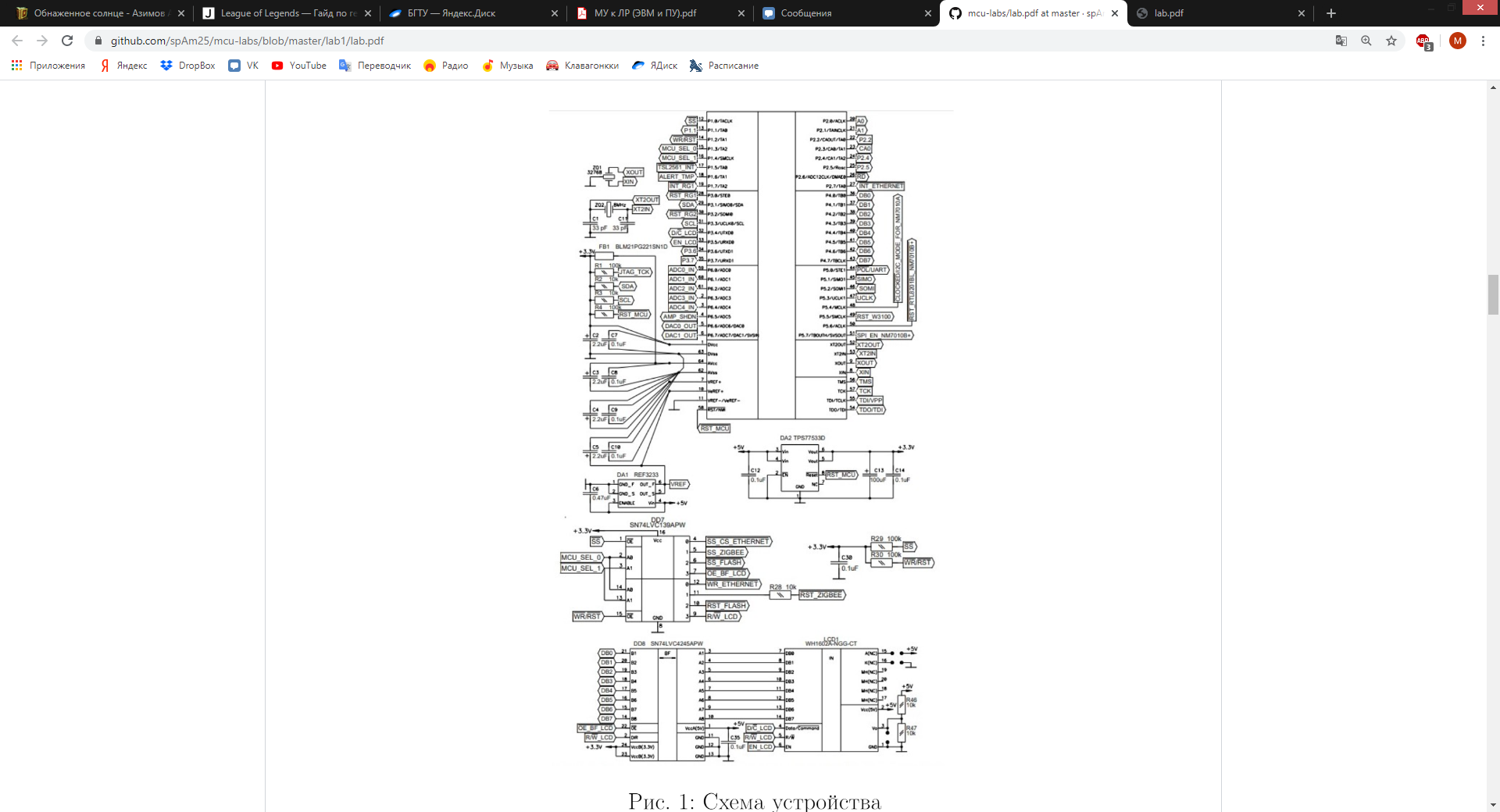
**Задание**: Разработать в среде программирования Code Composer программу для микроконтроллера MSP430, которая выполняет вывод информации (фамилию, имя и отчество студента, и группу) на экран цифрового индикатора.

**Порядок выполнения работы**:

* запустить Code Composer IDE.
* создать пустой проект.
* создать файл ресурса для кода программы и подключить его к проекту.
* выполнить компиляцию исходного модуля программы и устранить ошибки, полученные на данном этапе.
* проверить работоспособность программы и показать результаты работы преподавателю.

**Ход работы:**

Схема стенда



Демонстрация работы программы:



Содержимое файла main.c

#include <msp430.h>

#include "stdio.h"

#include "system\_define.h"

#include "system\_variable.h"

#include "function\_prototype.h"

#include "main.h"

void main(void) {

WDTCTL = WDTPW + WDTHOLD;

Init\_System();

Init\_System\_Clock();

LCD\_init();

LCD\_set\_pos(0,2);

LCD\_message("Горохов и");

LCD\_set\_pos(1,0);

LCD\_message("Сидорова ВТ-31");

LCD\_set\_cursor(3);

LCD\_WriteCommand(64);

LCD\_WriteByte(0b000000000,1);

LCD\_WriteByte(0b000000000,1);

LCD\_WriteByte(0b000010010,1);

LCD\_WriteByte(0b000000000,1);

LCD\_WriteByte(0b000010010,1);

LCD\_WriteByte(0b000001100,1);

LCD\_WriteByte(0b000000000,1);

LCD\_WriteByte(0b000000000,1);

LCD\_set\_pos(0,0);

LCD\_WriteByte(0,1);

while(1);

}

Содержимое файла lcd.c

// LCD-display functions

#include "function\_prototype.h"

#include "sysfunc.h"

#include "lcd.h"

//Таблица киррилицы

char LCD\_table[64]={

0x41,0xA0,0x42,0xA1, //0xC0...0xC3 <=> А Б В Г

0xE0,0x45,0xA3,0x33, //0xC4...0xC7 <=> Д Е Ж З

0xA5,0xA6,0x4B,0xA7, //0xC8...0xCB <=> И Й К Л

0x4D,0x48,0x4F,0xA8, //0xCC...0xCF <=> М Н О П

0x50,0x43,0x54,0xA9, //0xD0...0xD4 <=> Р С Т У

0xAA,0x58,0xE1,0xAB, //0xD5...0xD7 <=> Ф Х Ц Ч

0xAC,0xE2,0xAC,0xAE, //0xD8...0xDB <=> Ш Щ Ъ Ы

0x62,0xAF,0xB0,0xB1, //0xDC...0xDF <=> Ь Э Ю Я

0x61,0xB2,0xB3,0xB4, //0xE0...0xE4 <=> а б в г

0xE3,0x65,0xB6,0xB7, //0xE5...0xE7 <=> д е ж з

0xB8,0xA6,0xBA,0xBB, //0xE8...0xEB <=> и й к л

0xBC,0xBD,0x6F,0xBE, //0xEC...0xEF <=> м н о п

0x70,0x63,0xBF,0x79, //0xF0...0xE4 <=> р с т у

0xE4,0xD5,0xE5,0xC0, //0xF5...0xE7 <=> ф х ц ч

0xC1,0xE6,0xC2,0xC3, //0xF8...0xEB <=> ш щ ъ ы

0XC4,0xC5,0xC6,0xC7 //0xFC...0xEF <=> ь э ю я

};

byte LCD\_row, LCD\_col, n;

void LCD\_init()

{

wait\_1ms(20);

P3DIR |= (D\_nC\_LCD + EN\_LCD);

Reset\_EN\_LCD();

LCD\_WriteCommand(0x3C);

wait\_1ms(1);

LCD\_WriteCommand(0x3C);

wait\_1ms(1);

LCD\_WriteCommand(0x0C);

LCD\_clear();

LCD\_WriteCommand(0x06);

}

void LCD\_message(const char \* buf)

{

n = 0;

while (buf[n])

{

if ( (LCD\_row < LCD\_MAXROWS-1) && (LCD\_col >= LCD\_MAXCOLS) )

LCD\_set\_pos(++LCD\_row, 0);

if (LCD\_col >= LCD\_MAXCOLS )

LCD\_set\_pos(0,0);

LCD\_WriteData( LCD\_recode(buf[n]) );

LCD\_col++;

n++;

}

}

void LCD\_clear()

{

LCD\_WriteCommand(0x01);

LCD\_row=0;

LCD\_col=0;

}

void LCD\_set\_pos(byte row, byte col)

{

if (row > LCD\_MAXROWS-1)

row = LCD\_MAXROWS-1;

if (col > LCD\_MAXCOLS-1)

col = LCD\_MAXCOLS-1;

LCD\_row = row;

LCD\_col = col;

LCD\_WriteCommand( BIT7 | ((0x40 \* LCD\_row) + LCD\_col) );

}

byte LCD\_get\_row()

{

return LCD\_row;

}

byte LCD\_get\_col()

{

return LCD\_col;

}

void LCD\_set\_cursor(byte cursor)

{

if (cursor > 3)

cursor = 2;

LCD\_WriteCommand(cursor | BIT2 | BIT3);

}

void LCD\_WriteCommand(char byte)

{

LCD\_WriteByte(byte, 0);

}

void LCD\_WriteData(char byte)

{

LCD\_WriteByte(byte, 1);

}

void LCD\_WriteByte(char byte, char D\_nC)

{

DB\_DIR = 0x00;

Set\_MCU\_SEL\_0();

Set\_MCU\_SEL\_1();

Reset\_D\_nC\_LCD();

Set\_nWR\_nRST();

Reset\_nSS();

Set\_EN\_LCD();

Set\_EN\_LCD();

Set\_EN\_LCD();

while (DB\_IN & BIT7);

Reset\_EN\_LCD();

Set\_nSS();

if (D\_nC) Set\_D\_nC\_LCD();

else Reset\_D\_nC\_LCD();

Reset\_nWR\_nRST();

Reset\_nSS();

DB\_DIR = 0xFF;

DB\_OUT = byte;

Set\_EN\_LCD();

Set\_EN\_LCD();

Set\_EN\_LCD();

Reset\_EN\_LCD();

Set\_nSS();

DB\_DIR = 0x00;

Set\_nWR\_nRST();

}

char LCD\_recode(char b)

{

if (b<192) return b;

else return LCD\_table[b-192];

}