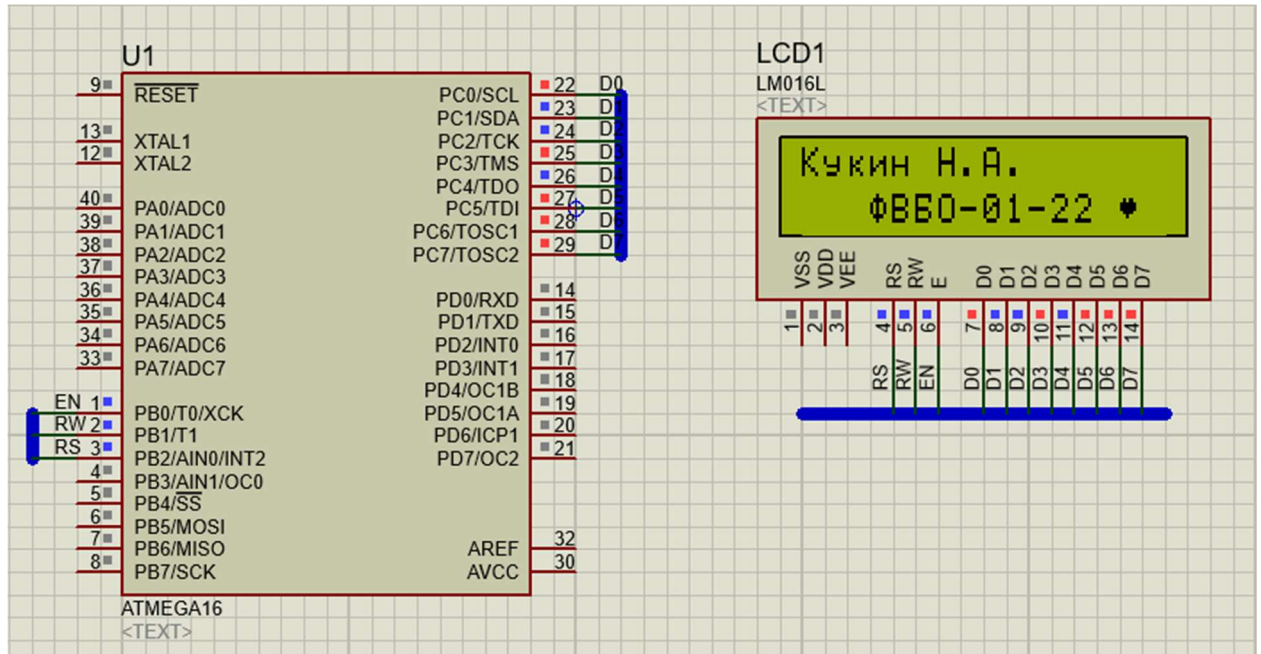


Разработать программное обеспечение, выводящее фамилию и имя студента на верхней строке, название группы- по центру нижней строки ЖК дисплея. Кириллические символы при выводе на экран разработать самостоятельно.

Мой вариант: «Кукин Н.А.», «ФВБО-01-22».



```
#define F_CPU 12000000UL
```

```
#include <util/delay.h>
```

```
#include <avr/io.h>
```

```
#define RS 2
```

```
#define RW 1
```

```
#define EN 0
```

```
void lcd_init(void);
```

```
void lcd_com(unsigned char);
```

```
void lcd_dat(unsigned char);
```

```
void lcd_string(char*);
```

```
void create_custom_symbol(unsigned char location, unsigned char *char_map);
```

```
void upload_custom_symbol();
```

```
int main(void) {
```

```
    // Разработать программное обеспечение, выводящее фамилию и имя  
    студента на верхней строке,
```

```
    // название группы - по центру нижней строки ЖК дисплея.
```

```
    Кириллические символы при выводе
```

```
    // на экран разработать самостоятельно
```

```
    // Кукин Н.А. ФВБО-01-22
```

```

lcd_init(); // Инициализирую дисплей
upload_custom_symbol(); // Записываю в CGRAM новые символы

// Посимвольно вывожу на экран нужный текст
lcd_com(0x80); // Курсор на начало первой строки
lcd_dat(75); // К
lcd_dat(121); // у
lcd_dat(1); // к
lcd_dat(2); // и
lcd_dat(3); // н
lcd_dat(16); // Пробел
lcd_dat(72); // Н
lcd_dat(46); // Точка
lcd_dat(65); // А
lcd_dat(46); // Точка

lcd_com(0xC3); // Курсор с отступом на вторую строку
lcd_dat(4); // Ф
lcd_dat(66); // В
lcd_dat(5); // Б
lcd_dat(79); // О
lcd_dat(45); // -
lcd_dat(48); // 0
lcd_dat(49); // 1
lcd_dat(45); // -
lcd_dat(50); // 2
lcd_dat(50); // 2
lcd_dat(16); // Пробел
lcd_dat(0); // Сердечко

lcd_com(0xE9); // Убираем курсор

while (1) { }
}

void lcd_com(unsigned char p){
    PORTB &= ~(1<<RS);
    PORTB |= (1<<EN); // EN=1 начало записи команды
    PORTC = p; // Вывод команды на шину данных экрана
    _delay_us(500);
    PORTB &= ~(1<<EN); // EN=1 конец записи команды
    _delay_us(500);
}

```

```

void lcd_dat(unsigned char p){
    PORTB|=(1<<RS)|(1<<EN); // Начало записи данных
    PORTC=p; // Установка значений
    _delay_us(500);
    PORTB&=~(1<<EN); // Конец записи данных
    _delay_us(500);
}

void lcd_init(void){
    DDRB |= (1<<RS)|(1<<RW)|(1<<EN); // Управление на вывод

    // Обнуляем биты управления и шин данных
    PORTB=0x00;
    DDRC=0xFF;
    PORTC=0x00;

    _delay_us(500);
    lcd_com(0x08); // Полное включение дисплея
    _delay_us(500);
    lcd_com(0x3C); // 8 бит данных 2 строки
    _delay_us(500);
    lcd_com(0x01); // очистка строки
    _delay_us(500);
    lcd_com(0x06); // сдвиг курсора вправо
    _delay_us(900);
    lcd_com(0x0F); // курсор показан и мигает
}

// Выводит на экран строку символов (латиница)
void lcd_string(char *str){
    char data=0;
    while(*str){
        data=*str++;
        lcd_dat(data);
    }
}

void create_custom_symbol(unsigned char location, unsigned char *char_map) {
    unsigned char i;
    lcd_com(0x40 + (location * 8)); // Установка адреса CGRAM для символа
    for (i = 0; i < 8; i++) {
        lcd_dat(char_map[i]); // Запись байтов символа в CGRAM
    }
}

```

```

void upload_custom_symbol() {
    // Массив содержащий 8 символов
    unsigned char custom_symbols[8][8] = {
        {0b00000, 0b01010, 0b11111, 0b11111, 0b01110, 0b00100, 0b00000,
0b00000},
        {0b00000, 0b00000, 0b01001, 0b01010, 0b01100, 0b01010, 0b01001,
0b00000},
        {0b00000, 0b00000, 0b10001, 0b10011, 0b10101, 0b11001, 0b10001,
0b00000},
        {0b00000, 0b00000, 0b10001, 0b10001, 0b11111, 0b10001, 0b10001,
0b00000},
        {0b00100, 0b01110, 0b10101, 0b10101, 0b10101, 0b01110, 0b00100,
0b00000},
        {0b11111, 0b10001, 0b10000, 0b11110, 0b10001, 0b10001, 0b11110,
0b00000},
        {0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b00000,
0b00000},
        {0b00000, 0b01010, 0b11111, 0b11111, 0b01110, 0b00100, 0b00000,
0b00000}
    };

    // Загрузка символов в CGRAM
    for (int i = 0; i < 8; i++) {
        create_custom_symbol(i, custom_symbols[i]);
    }
}

```