

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра САУ**

**Лабораторная работа №4**  
**по дисциплине «МПУ»**  
**Вариант 7**

Студенты гр. 6

\_\_\_\_\_

Преподаватель

\_\_\_\_\_

Санкт-Петербург

2018

**Цель работы.** Программирование процедур формирования периодических звуковых и световых сигналов.

**Задание.** Необходимо написать программу вывода на звуковой излучатель фамилии одного из ее авторов в кодах азбуки Морзе, сопровождая звуковые сигналы выводом на RGB-светодиод световых сигналов заданного цвета. Данную последовательность необходимо выводить в бесконечном цикле с паузой в 5 с.

Частота	Цвет тире	Цвет точки
700	Красный	Фиолетовый

Таблица 1. Данные задания

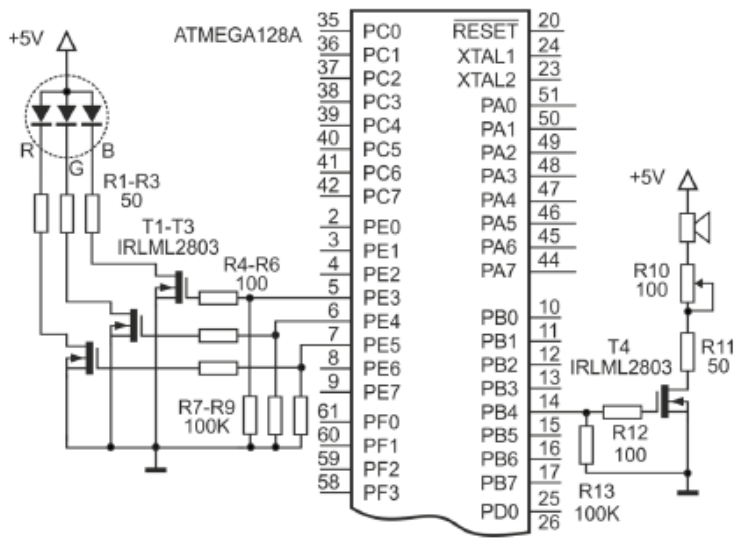


Рис.1 Схема подключения RGB-светодиодов и звукового излучателя

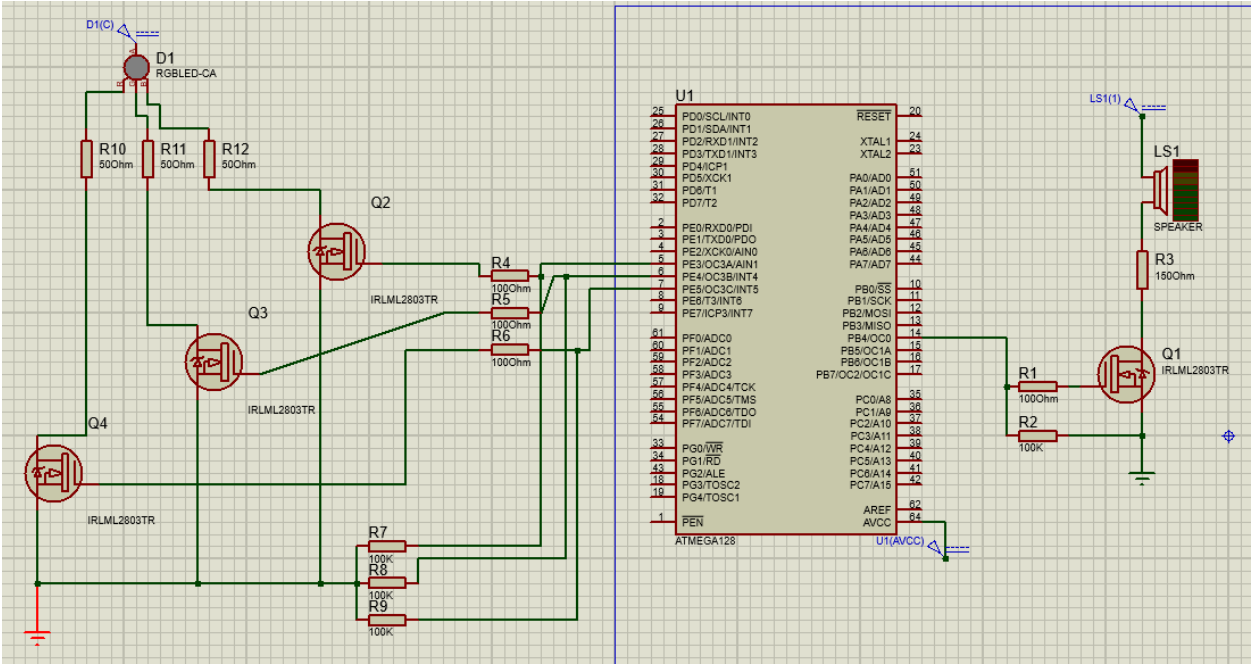
Буква	Код	Буква	Код	Буква	Код	Буква	Код
А	• —	Б	— • • •	В	• • —	Г	— — •
Д	— • •	Е	•	Ж	• • • —	З	— — • •
И	• •	Й	• — — —	К	— • —	Л	• — • •
М	— —	Н	— •	О	— — —	П	• — — •
Р	• • •	С	• • •	Т	—	У	• • —
Ф	• • — •	Х	• • • •	Ц	— • — •	Ч	— — — •
Ш	— — — —	Щ	— — • —	Ы	— • — —	Ь	— • • —
Э	• • — • •	Ю	• • — —	Я	• • • —		

Таблица 1. Коды азбуки Морзе, соответствующие буквам

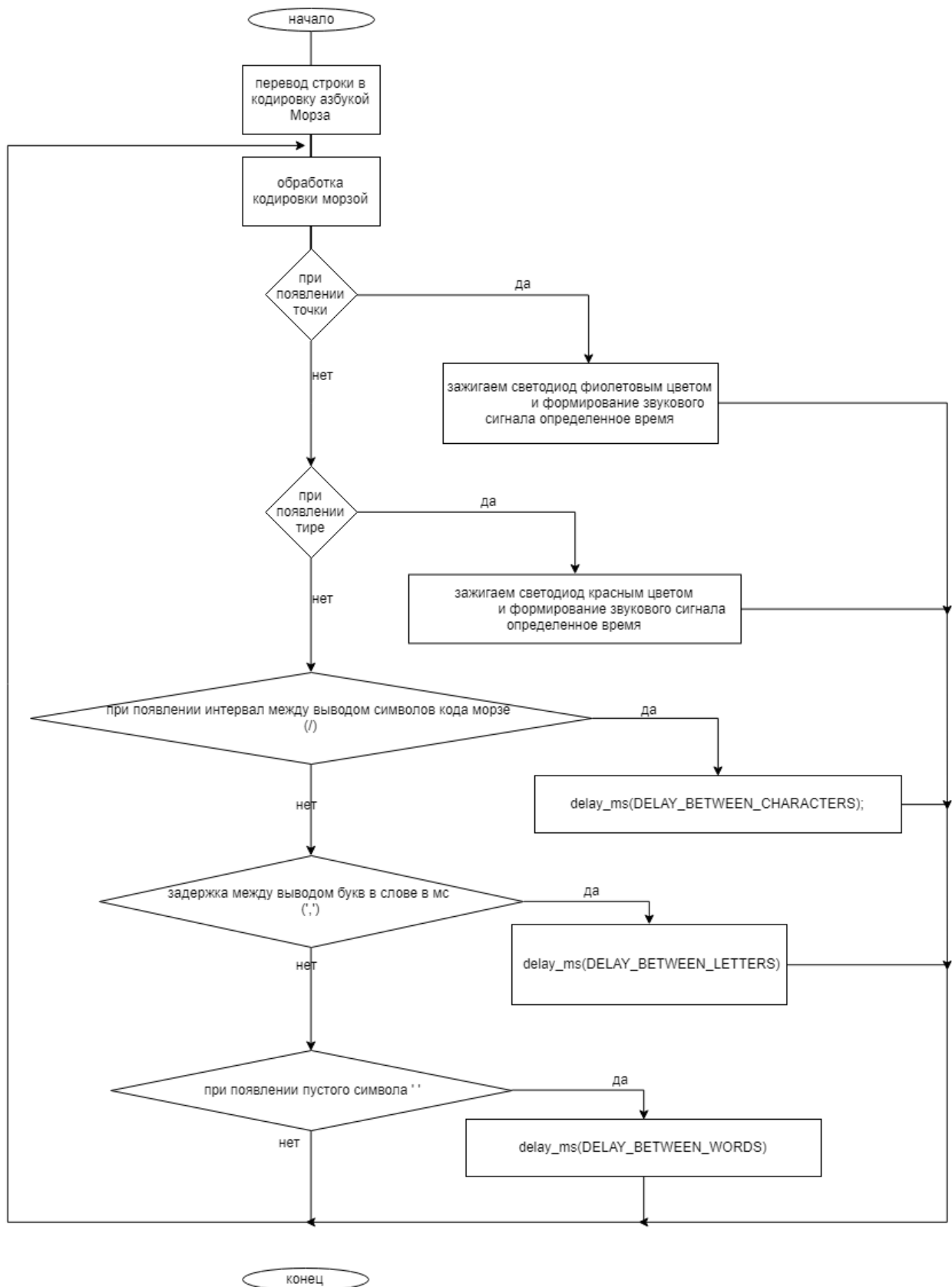
Цвет свечения	Выводы диода	Разряды PORTE		
		PE3	PE4	PE5
Красный	R	0	0	1
Зеленый	G	0	1	0
Синий	B	1	0	0
Белый	R + G + B	1	1	1
Желтый	R + G	0	1	1
Фиолетовый	R + B	1	0	1
Голубой	G + B	1	1	0

Таблица 2. Таблица цветов и разрядов

Proteus



Блок-схема



## Код программы.

/\*

Variant 7

- красный . фиолетовый

Частота 700 гц, отсюда период 1429 мксек

-----  
пример исполняемый код

-. ./ ., (каждый символ представляет собой задержку)

пример выполнения кода с задержками

TIME\_DASH DELAY\_BETWEEN\_CHARACTERS TIME\_DOT DELAY\_BETWEEN\_LETTERS TIME\_DOT  
DELAY\_BETWEEN\_CHARACTERS TIME\_DOT DELAY\_BETWEEN\_WORDS TIME\_DOT  
DELAY\_BETWEEN\_CHARACTERS

\*/

#include <mega128a.h>

#include <delay.h>

#include <string.h>

#include <stdio.h>

#include <ctype.h>

#define OUTPUT\_STRING "MY NAME IS VICTOR" //Выводимая строка

#define HALF\_PERIOD 714 // T/2 в мксек

#define TIME\_DOT 500 // длительность точки в мс (·)

#define TIME\_DASH 1500 //длительность тире в мс (-)

#define DELAY\_BETWEEN\_CHARACTERS 500 // интервал между выводом символов кода морзе в мс  
(/)слэш

#define DELAY\_BETWEEN\_LETTERS 1000// задержка между выводом букв в слове в мс ( ) (.)

#define DELAY\_BETWEEN\_WORDS 2000 //задержка между выводом слов в строке в мс ( )пробел

#define MAX\_SIZE\_CODE 100 // максимальный размер кодировкой азбукой морза в байтах

#define PAUSE 5000 //пауза между повторением азбуки морза 5 сек

void main(void)

{

int i;

char j;

int time\_dot = (TIME\_DOT\*1000)/(HALF\_PERIOD\*2) ; //количество итераций цикла , соответствующее  
необходимому времени

int time\_dash = (TIME\_DASH\*1000)/(HALF\_PERIOD\*2) ; //количество итераций цикла , соответствующее  
необходимому времени

//в переменной массива последний символ "\0" !!!

char output\_string[] = OUTPUT\_STRING ; //входная строка

char coded\_string[MAX\_SIZE\_CODE] = "" ; // кодировка азбукой морза

char mrz[]=""; //хранит кодировку символа морза

//-----перевод строки в кодировку азбукой Морза-----

for (i = 0; i < strlen(output\_string); i++)

{

output\_string[i] = tolower(output\_string[i]);

switch (output\_string[i])

{

case 'a': sprintf(mrz, ".-"); break;

case 'b': sprintf(mrz, "-./."); break;

case 'w': sprintf(mrz, ".-/-"); break;

case 'g': sprintf(mrz, "-./."); break;

case 'd': sprintf(mrz, "-./."); break;

case 'e': sprintf(mrz, "."); break;

case 'v': sprintf(mrz, ".-./.-"); break;

case 'z': sprintf(mrz, "-./."); break;

```

case 'i': sprintf(mrz, "."); break;
case 'j': sprintf(mrz, "-/-"); break;
case 'k': sprintf(mrz, "-/-"); break;
case 'l': sprintf(mrz, "-/./"); break;
case 'm': sprintf(mrz, "-"); break;
case 'n': sprintf(mrz, "-"); break;
case 'o': sprintf(mrz, "-/-"); break;
case 'p': sprintf(mrz, "-/./"); break;
case 'r': sprintf(mrz, "-/./"); break;
case 's': sprintf(mrz, "-/./"); break;
case 't': sprintf(mrz, "-"); break;
case 'u': sprintf(mrz, "-/./"); break;
case 'f': sprintf(mrz, "-/./"); break;
case 'h': sprintf(mrz, "-/./"); break;
case 'c': sprintf(mrz, "-/./"); break;
case 'q': sprintf(mrz, "-/./"); break;
case 'y': sprintf(mrz, "-/./"); break;
case 'x': sprintf(mrz, "-/./"); break;
case '1': sprintf(mrz, "-/-/-"); break;
case '2': sprintf(mrz, "-/./"); break;
case '3': sprintf(mrz, "-/./"); break;
case '4': sprintf(mrz, "-/./"); break;
case '5': sprintf(mrz, "-/./"); break;
case '6': sprintf(mrz, "-/./"); break;
case '7': sprintf(mrz, "-/./"); break;
case '8': sprintf(mrz, "-/./"); break;
case '9': sprintf(mrz, "-/./"); break;
case '0': sprintf(mrz, "-/./"); break;
case '!': sprintf(mrz, "-/./"); break;
case ',': sprintf(mrz, "-/./"); break;
case ':': sprintf(mrz, "-/./"); break;
case ';': sprintf(mrz, "-/./"); break;
case '(': sprintf(mrz, "-/./"); break;
case ')': sprintf(mrz, "-/./"); break;
case '"': sprintf(mrz, "-/./"); break;
case '-': sprintf(mrz, "-/./"); break;
case '/': sprintf(mrz, "-/./"); break;
case '?': sprintf(mrz, "-/./"); break;
case '!': sprintf(mrz, "-/./"); break;
case ' ': sprintf(mrz, " "); break; //пробел между словами
case '@': sprintf(mrz, "-/./"); break;
default: sprintf(mrz, ""); break;

}
strcat(coded_string, mrz); //добавляем к конечной строке код символа в азбуке морза
if ((output_string[i] != ' ') && (output_string[i+1] != ' ')) //условие для устранения лишних
задержек при пробеле
{
    strcat(coded_string, ","); // указывает на задержка между выводом букв в слове
}

}
//-----

//обработка кодировки морзой
DDRB |= 1<<4;

```

```

DDRE |= (1<<3)|(1<<4)|(1<<5);
while (1)
{
    delay_ms(PAUSE); //пауза между повторением азбуки морза 5 сек
    for( i = 0; i < strlen(coded_string) ; i++) //перебираем кодировку строки, последний символ
отбрасываем, так как он является запятой, после которой не следует символов
    {
        switch(coded_string[i])
        {
            case '.': //при появлении точки
            {
                PORTE |= (1<<3)|(1<<5); //зажигаем светодиод фиолетовым цветом
                //формирование звукового сигнала определенное время
                for(j=0; j <= time_dot;j++)//звуковой сигнал длится нужное время
                {
                    PORTB |= 1<<4; //передний фронт сигнала
                    delay_us(HALF_PERIOD); //время импульса
                    PORTB &= ~(1<<4) ; //задний фронт сигнала
                    delay_us(HALF_PERIOD); //время паузы между импульсами
                }
                PORTE &= ~((1<<3)|(1<<5)) ; //тушим светодиод
                break;
            }
            case '-':
            {
                PORTE |= 1; //зажигаем светодиод красным цветом
                //формирование звукового сигнала определенное время
                for(j=0; j <= time_dash ;j++)//звуковой сигнал длится нужное время
                {
                    PORTB |= 1<<4; //передний фронт сигнала
                    delay_us(HALF_PERIOD); //время импульса
                    PORTB &= ~(1<<4) ; //задний фронт сигнала
                    delay_us(HALF_PERIOD); //время паузы между импульсами
                }
                PORTE &= ~(1) ; //тушим светодиод
                //delay_ms(DELAY_BETWEEN_LETTERS);// интервал между выводом символов кода морзе в
мс
                break;
            }
            case '/': //при появлении интервал между выводом символов кода морзе
            {
                delay_ms(DELAY_BETWEEN_CHARACTERS);
                break;
            }
            case ',': //задержка между выводом букв в слове в мс
            {
                delay_ms(DELAY_BETWEEN_LETTERS);
                break;
            }
            case ' ': //задержка между выводом слов в строке
            {
                delay_ms(DELAY_BETWEEN_WORDS);
                break;
            }
        }
    }
}

```

```
}  
}  
}
```

**Вывод:** Программирование процедур формирования периодических звуковых и световых сигналов.