

Санкт-Петербургский национальный исследовательский университет  
информационных технологий, механики и оптики

Кафедра вычислительной техники

**Дисциплина «Организация ЭВМ и систем»**

**Лабораторная работа №8**

**Вариант 1**

**Выполнили:**

Съестов Дмитрий Вячеславович

Хлопков Дмитрий Сергеевич

Группа Р3317

**Преподаватель:**

Скорубский Владимир Иванович

Санкт-Петербург  
2018

## Цель работы

Организовать в C51 последовательный ввод по прерыванию. Интерпретировать прерывание чтением кода из массива кодов сканирования. Числа при вводе в формате с естественной запятой преобразуются в формат с плавающей.

1 2 3 4  
5 6 7 8  
9 0 (,) .  
(-) . . e

Поскольку Keil не поддерживает работу с клавиатурой, то вводим виртуальную клавиатуру, сканкоды которой представляют собой инверсный унитарный код, где старшая тетрада соответствует строке, а младшая – столбцу.

## Описание алгоритма

- 1) Обработчик прерывания считывает очередную клавишу из массива кодов `scancodes[]`.
- 2) По коду определяется символ и помещается в массив `result[]`.
  - Если это запятая, устанавливаем масштаб, который будет увеличиваться десятикратно с каждой последующей цифрой.
  - Если это минус, делаем пометку, что число отрицательное.
  - Если это цифра, увеличиваем масштаб и приписываем её к `number`.
  - Если это символ конца ввода, завершаем считывание.
- 3) Делим `number` на полученное значение масштаба (если он установлен)
- 4) Если был введён минус, умножаем `number` на -1.

## Исходный код

```
#define KEY(row, col) ~((0x80 >> row) | (0x08 >> col))
#define SC_1      KEY(0, 0)
#define SC_2      KEY(0, 1)
#define SC_3      KEY(0, 2)
#define SC_4      KEY(0, 3)
#define SC_5      KEY(1, 0)
#define SC_6      KEY(1, 1)
#define SC_7      KEY(1, 2)
#define SC_8      KEY(1, 3)
#define SC_9      KEY(2, 0)
#define SC_0      KEY(2, 1)
#define SC_DOT    KEY(2, 2)
#define SC_MINUS  KEY(3, 0)
#define SC_END    KEY(3, 3)

char key, i, digit, result[7], isNegative;
float number;
int scale;
//-1,234e
char code scancodes[]={SC_MINUS, SC_1, SC_DOT, SC_2, SC_3, SC_4, SC_END};
```

```

void wait(int t)
{
    while(t--);
}

char what(void);

void scan() interrupt 2 //INT1
{
    wait(100);
    key = scancodes[i];
    what();
    wait(100);
}

int main ()
{
    number = 0;
    i = 0;
    scale = 0;

    IT1 = 1;
    EX1 = 1;
    EA = 1;      //    interrupt type

    while (digit!='e'); //end of input
    while(1);
}

char what(void)
{
    switch (key)
    {
        case SC_0:      digit = '0'; break;
        case SC_1:      digit = '1'; break;
        case SC_2:      digit = '2'; break;
        case SC_3:      digit = '3'; break;
        case SC_4:      digit = '4'; break;
        case SC_5:      digit = '5'; break;
        case SC_6:      digit = '6'; break;
        case SC_7:      digit = '7'; break;
        case SC_8:      digit = '8'; break;
        case SC_9:      digit = '9'; break;
        case SC_MINUS:  digit = '-'; break;
        case SC_DOT:    digit = ','; break;
        case SC_END:    digit = 'e'; break;
        default:        digit = 0xff; break;
    }
    switch(digit)
    {
        case ',':
            scale = 1;
            result[i++] = digit;
            break;

        case '-':
            isNegative = 1;
            result[i++] = digit;
            break;
    }
}

```

```
    case 'e':
        if(scale != 0) number /= scale;
        if(isNegative) number = -number;
        scale = 0;
        i = 0;
        break;

    default:
        scale *= 10;
        number = number * 10 + (digit & 0x0f);
        result[i++] = digit;
        break;
}
return digit;
```

```
}
```