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 P3317

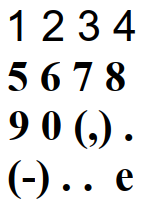
**Преподаватель:**

Скорубский Владимир Иванович

Санкт-Петербург

2018

**Цель работы**  
Организовать в С51 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.



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.

1. Делим number на полученное значение масштаба (если он установлен)
2. 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;

}