

Университет ИТМО

Факультет программной инженерии и компьютерной техники

Кафедра вычислительной техники

Лабораторная работа № 4 по дисциплине
"Информационно-управляющие системы"

Вариант 4

Выполнили:

Айтуганов Д. А.

Чебыкин И. Б.

Группа: Р3401

Проверяющий: Пинкевич В. Ю.

Санкт-Петербург, 2017

Задание

Скорость последовательного канала – 4800 бит/с. Вычитатель десятичных чисел. Диапазон значений уменьшаемого и вычитаемого – от 0 10 до 99 10 включительно. Разность может быть как положительной, так и отрицательной. При помощи клавиатуры SDK-1.1 вводятся уменьшаемое и вычитаемое (десятичные числа), причем разделителем введенных значений является символ «-» (кнопка «В» на клавиатуре), символом начала вычисления – «=» (кнопка «#»). Вводимые с клавиатуры числа и результат должны выводиться в последовательный канал и отображаться в терминале персонального компьютера. Каждое новое выражение начинается с новой строки. Должен быть предусмотрен контроль ввода корректных значений.

Исходный код

```

#include "main.h"
#include "kb.h"
#include "serial.h"

#define NORMAL 0xFF
#define DEBUG 0xFE

#define BUFFSZ 8

#define NEXT_CH 0

#define NUM_SZ 2

#define POS_FIRSTNUM_S 0

#define ST_FIRST 0
#define ST_SECOND 1

unsigned char READ_FIFO[BUFFSZ] = {0};
unsigned char ir = 0;

unsigned char first_size = 0;
unsigned char second_size = 0;
char first_num = -1;
char second_num = -1;

unsigned char state = 0;

void print_error(){
    EA = 0;
    type(EOL);
    type("Invalid arguments.");
    type(EOL);
    EA = 1;
}

void print_num(char num) {
    if(num < 0) {
        uart_s_write('-');
        num *= -1;
    }
    if(num > 9) uart_s_write(num / 10 + '0');
    uart_s_write(num % 10 + '0');
}

void print_result() {
    char i = first_num - second_num;
    print_num(i);

    ir = 0;
    type(EOL);
}

static int to_num(char *num, unsigned char size, int fifo_pos) {
    if(size > 1) *num = (READ_FIFO[fifo_pos] - '0') * 10 + (READ_FIFO[fifo_pos+1] - '0');
    else *num = READ_FIFO[fifo_pos] - '0';
    if(*num > 99) {
        return -1;
    }
    return 0;
}

static int add_char(unsigned char button) {
    READ_FIFO[ir++] = button;
    if ( (ir == BUFFSZ && READ_FIFO[ir - 1] != '#') || (ir == 1 && READ_FIFO[ir - 1] == '#') ) {
        return -1;
    }
}

```

```

if(READ_FIFO[ir - 1] >= '0' && READ_FIFO[ir - 1] <= '9') {
    if(state == ST_FIRST) {
        if(++first_size > NUM_SZ) return -1;
    }
    if(state == ST_SECOND) {
        if(++second_size > NUM_SZ) return -1;
    }
    return 0;
}
if(READ_FIFO[ir - 1] == 'B') {
    if(first_num >= 0) return -1;
    if(first_size == 0) return -1;
    state = ST_SECOND;
    return 1;
}
if(READ_FIFO[ir - 1] == '#') {
    if(first_size == 0) return -1;
    if(second_size == 0) return -1;
    state = ST_FIRST;
    return 2;
}
return -1;
}

static void reset() {
    state = ST_FIRST;
    first_size = 0;
    second_size = 0;
    ir = 0;
    ET0 = 1;
    first_num = -1;
    second_num = -1;
}

static void fail() {
    reset();
    type(EOL);
    print_error();
}

void main() {
    unsigned char dip = 0, button = 0, j = 0;
    int rc = 0;

    uart_s_init(S4800);
    init_kb_timer();

    EA = 1;

    while (1) {
        dip = readdip();
        if (dip == NORMAL) {
            if (!is_queue_empty()) {
                ET0 = 0;

                button = get_input();
                if(button == 'B') uart_s_write('-');
                else if(button == '#') uart_s_write('=');
                else uart_s_write(button);

                rc = add_char(button);
                if(rc < 0) {
                    fail();
                    continue;
                }
                if(rc == 1) {
                    rc = to_num(&first_num, first_size, POS_FIRSTNUM_S);
                    if(rc < 0) {
                        fail();
                    }
                }
            }
        }
    }
}

```

```

    }
    if(rc == 2) {
        rc = to_num(&second_num, second_size, first_size + 1);
        if(rc < 0) {
            fail();
            continue;
        }
        print_result();
        reset();
    }

    ET0 = 1;
}
} else if (dip == DEBUG) {
    if (!is_queue_empty()) {
        ET0 = 0;
        uart_s_write(get_input());
        type(EOL);
        ET0 = 1;
    }
}
else {
    leds(0xAA);
}
}
}

```

Вывод

В результате выполнения лабораторной работы был получен опыт работы с клавиатурой в SDK-1.1.