Санкт-Петербургский национальный исследовательский университет

информационных технологий, механики и оптики
(Университет ИТМО)

Факультет программной инженерии и компьютерной техники

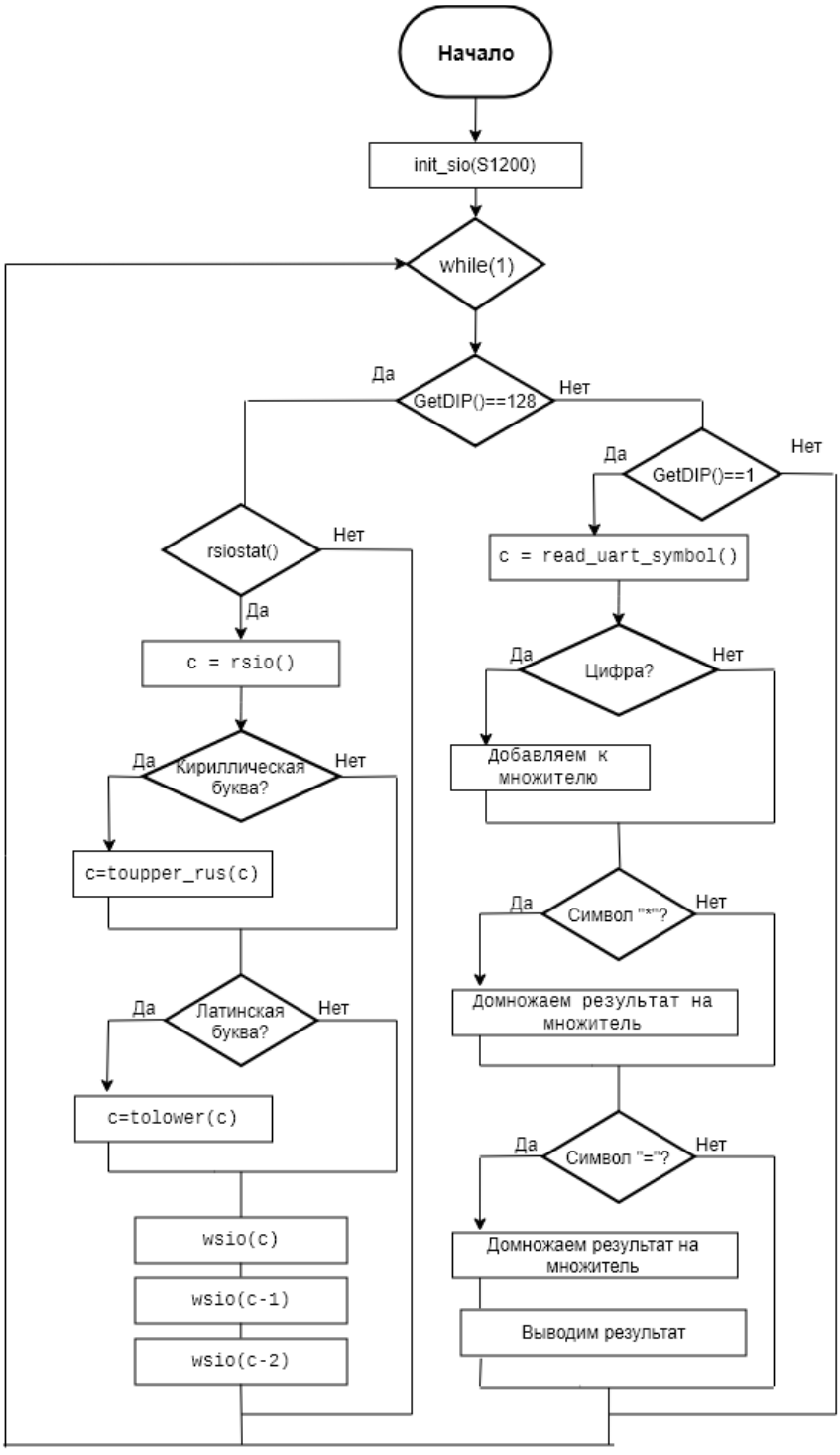
Кафедра вычислительной техники

Лабораторная работа №4
по дисциплине «Информационно-управляющие системы»
«Последовательный интерфейс RS-232. UART»
Вариант 6

Выполнили
студенты 4 курса,
группы P3400
Абыков Айдар Альбертович
Бурангулов Аскар Азаматович
Сапожников Борис Константинович
Руководитель:
Ключев Аркадий Олегович

Санкт-Петербург — 2018

1. Задание



Блок-схема

Разработать и написать драйверы последовательного канала для учебно-лабораторного стенда SDK-1.1 без использования прерываний. Написать тестовую программу для разработанных драйверов, которая выполняет определенную вариантом задачу.

Скорость последовательного канала – 1200 бит/с.

На каждый принятый по последовательному каналу символ (от персонального компьютера к SDK-1.1) в ответ передается этот же символ и 2 предшествующих ему символа согласно таблице ASCII (от SDK-1.1 к персональному компьютеру) и отображается в терминальной программе. Причем все символы русского алфавита отображаются в верхнем регистре, все символы английского алфавита – в нижнем регистре. Например, на символ 'л' ('Л') ответом является 'ЛКЙ', '5' – '543', 'i' ('I') – 'ihg' и т.д.

Умножитель десятичных чисел. Диапазон значений множителей - от 010 до 9910 включительно. Контроллеру SDK-1.1 по последовательному каналу со стороны персонального компьютера с использованием терминальной программы передаются множители (десятичные числа), причем разделителем введенных значений является символ умножения («*»), концом ввода является символ равенства («=»), получившееся выражение отображается в терминале персонального компьютера. После чего контроллер возвращает результат операции, который отображается в терминале. Каждое новое выражение начинается с новой строки. Сигнализация в случае ввода некорректных значений
- сообщение об ошибке в последовательный канал и зажигание светодиодов.

2. Исходный текст программы с комментариями.

lab4.c

```
#include "aduc812.h"
#include "async.h"
#include "sio.h"
#include "sync.h"
#include "system.h"
#include "led.h"
#include <stdlib.h>
#include <ctype.h>


#define READ_OK 0
#define READ_OUT_OF_RANGE_ERROR 1
#define READ_INVALID_CHAR_ERROR 2

#define ONE_CODE    49
#define ZERO_CODE   48

// Производит необходимую трансформацию литеральных символов
// Если русский литерал в верхнем регистре - вернуть тот же символ в нижнем регистре
// Если английский символ в нижнем регистре - вернуть тот же символ в верхнем регистре
// Вход: символ
// Выход: трансформированный символ
unsigned char TransformLetter(unsigned char c) {
//int transformLetter(int c){
       //CP866
       if( c>='a' && c<='z' )
                     return c;
       if( c>='A' && c<='Z' )
              return c + 0x20; //c=tolower(c);
       if( c>='A' && c<='п' || c>='р' && c<='ё' ){
              if( c>='a' && c<='п' )
                     return c - 0x20;
              if( c>='р' && c<='я' )
                     return c - 0x50;
              if( c=='ё' )
                     return c - 0x01;
       }
       return c;
}

// Определяет, является ли символ десятичной цифрой
// Вход: проверяемый символ
// Выход: 0 - не является
//              1 - является
bit IsDigit(unsigned char symb) {
       return symb >= '0' && symb <= '9';
}


unsigned char read_uart_symbol() {
       unsigned char c = 0;
       c = ReadUART();
       while (c == 0) {
              if (GetDIP() != 128)
              return 0;
              c = ReadUART();
       }
       return c;
}
```

```
void error(char * str ){
	leds(0xAA);
	type(str);
}


void main(void) {

	unsigned char c;
	unsigned int /*long*/ res = 1;
	unsigned char str[10/*3*/] = {0};
	unsigned char dig_count = 0;
	unsigned int ready_number = 0;
	unsigned char i = 0;

	 init_sio( S1200 );

		while (1) {

		if (GetDIP() == 128) {
			ES = 1;
			EA = 1;
			while (1) {
				c=read_uart_symbol();
				wsio(c);
				leds(0x00); //убираем сигнал ошибки
				if (GetDIP() != 128) {
					ES = 0;
					break;
				}

				if(IsDigit(c)){
					if(dig_count>=2){
						error("\r\nТрех- или более значное число\r\n");
						ready_number=0;
						dig_count=0;
						continue;
					}
					dig_count++;
					ready_number *= 10;
					ready_number += c - '0';
				}
				else if(c=='*'){
					if(dig_count==0){
						error("\r\nВы не ввели число\r\n");
					}
					else{
						res *= ready_number;
						ready_number=0;
						dig_count=0;
					}
				}
				else if(c=='='){
					res *= ready_number;
					type(itoa(res, str, 10));
					type("\r\n");
					res = 1;
					ready_number=0;
					dig_count=0;
				}
				else{
					if(c!=' ' && c!='\t'){
						error("\r\nИнвалидный символ\r\n");
```

```
                                        res = 1;
                                        ready_number=0;
                                        dig_count=0;
                                }
                        }
                        if (GetDIP() != 128) {
                                ES = 0;
                                break;
                        }
                }
        }
        if (GetDIP() == 1) {
                ES = 0;
                EA = 0;
                while (1) {
                        leds(0x00); //убираем сигнал ошибки
                        if (rsiostat()) {
                                c = rsio();

                                if (GetDIP() != 1)
                                        break;

                                c = TransformLetter(c);
                                wsio( c );
                                wsio( c-1 );
                                wsio( c-2 );
                                type("\r\n");
                                if (GetDIP() != 1)
                                        break;
                        }
                }
        }
    }
}
```

4. Основные результаты.

В результате выполнения работы были разработаны и написаны драйверы последовательного канала для учебно-лабораторного стенда SDK-1.1 без использования прерываний а также тестовая программу для разработанного драйвера, которая выполняет определенную вариантом задачу.