

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

ОТЧЕТ
по лабораторной работе №5
по дисциплине «Организация ЭВМ и систем»
Тема: использование аппаратных прерываний

| | | |
|--------------------|-------|-----------------|
| Студентка гр. 1361 | _____ | Горбунова Д. А. |
| Студент гр.1361 | _____ | Голубев Д. В. |
| Преподаватель | _____ | Гречухин М. Н. |

Санкт-Петербург
2022

Цель работы

Знакомство с различного вида аппаратными прерываниями и создание собственных подпрограмм обработки прерываний.

Формулировка задания

По заданию преподавателя разработать алгоритм и реализовать программу подключения собственной подпрограммы обработки прерывания и использовать её в цепочке со стандартной подпрограммой обработки прерывания от одного из следующих устройств компьютера:

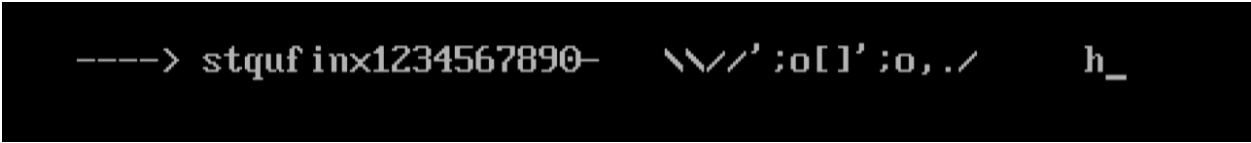
1. системный таймер;
2. клавиатура;
3. контроллер накопителя на гибких магнитных дисках;
4. таймер реального времени;
5. контроллер накопителя на жёстком магнитном диске.

Техническое задание

Разработать алгоритм и реализовать программу подключения собственной подпрограммы обработки прерывания и использовать ее в цепочке со стандартной подпрограммой обработки прерывания клавиатуры. А именно: на экран выводится значение нажимаемой кнопки равное порядковому значению в алфавите с конца. Т.е. если нажата клавиша А выводиться Z, если В – то Y, и так далее.

Результат работы программы

Данная программа выводит на экран реверснутое значение клавиш латинского алфавита, остальная клавиатура работает без изменений.



```
----> stquf inx1234567890-  \\'//';o[]';o,./  h_
```

Рисунок 1 — Результат работы программы

Приложение 1. Блок-схема реализуемого кода

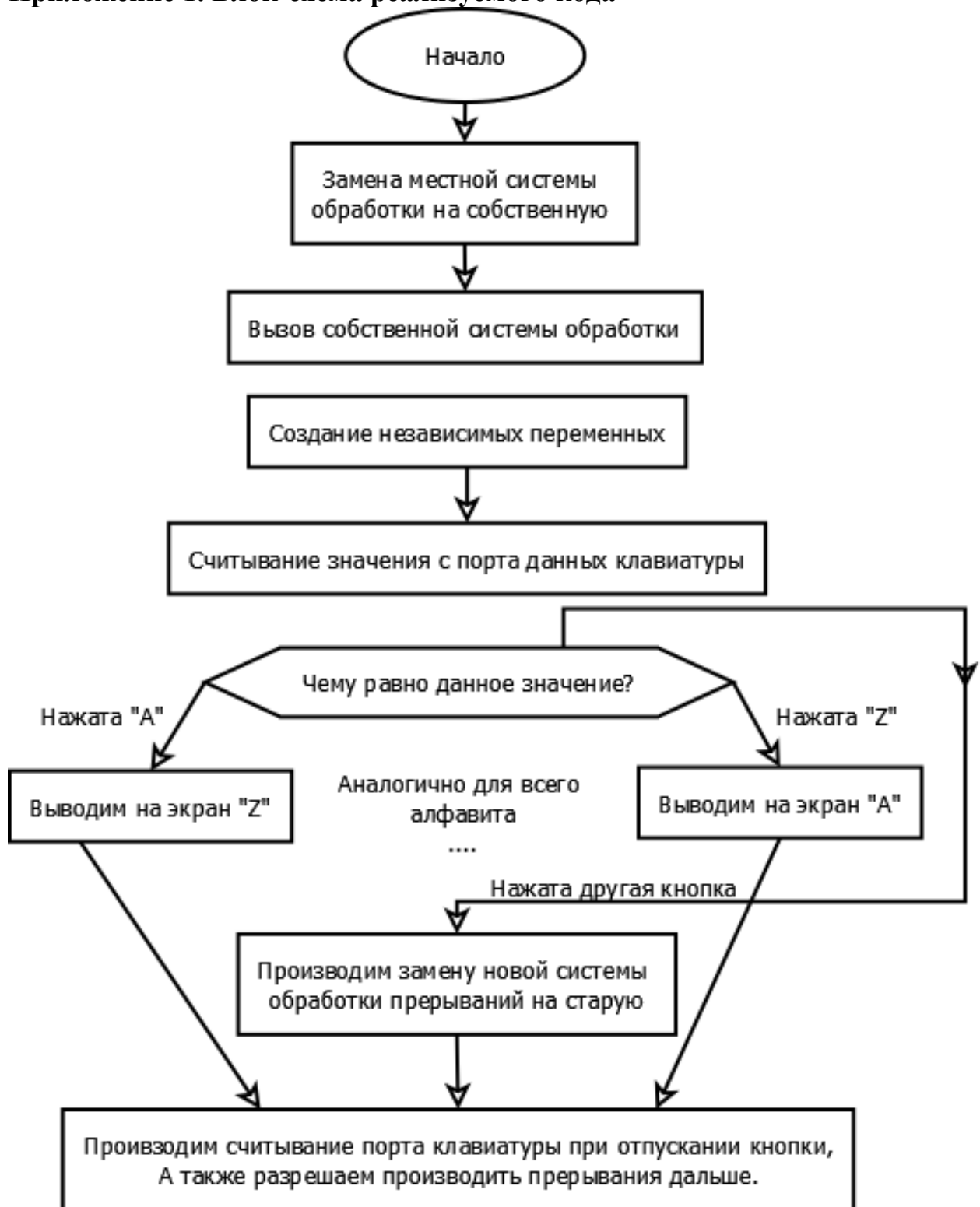


Рисунок 2 – Блок-схема исходного кода

Приложение 2. Структурная схема аппаратных средств

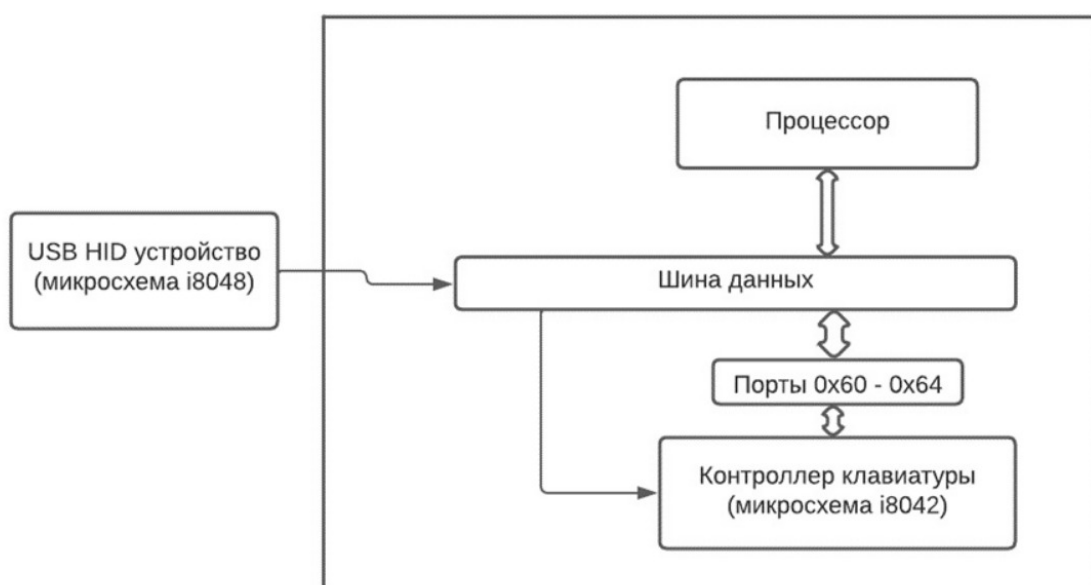


Рисунок 3 – Схема аппаратных средств.

Приложение 3. Исходный код программы

```
#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <iostream.h>
#ifdef __cplusplus
#define __CPPARGS ...
#else
#define __CPPARGS
#endif
void interrupt (*old9)(__CPPARGS);
void interrupt new9(__CPPARGS);

void main()
{
char string[10000];
clrscr();
old9=getvect(9);
setvect(9,new9);
printf("\n----> ");
scanf("%s",string);
setvect(9,old9);
}

void interrupt new9(__CPPARGS)
{
    unsigned char a = inportb(0x60);
    unsigned char oldvectport;
    unsigned char c;
    switch (a){
        case(1):
            (*old9)(__CPPARGS);
            break;
        case 30: printf("z");break;
        case 48: printf("y");break;
        case 46: printf("x");break;
        case 32: printf("w");break;
        case 18: printf("v");break;
        case 33: printf("u");break;
        case 34: printf("t");break;
        case 35: printf("s");break;
        case 23: printf("r");break;
        case 36: printf("q");break;
        case 37: printf("p");break;
```

```

        case 38: printf("o");break;
        case 50: printf("n");break;
        case 49: printf("m");break;
        case 24: printf("l");break;
        case 25: printf("k");break;
        case 16: printf("j");break;
        case 19: printf("i");break;
        case 31: printf("h");break;
        case 20: printf("g");break;
        case 22: printf("f");break;
        case 47: printf("e");break;
        case 17: printf("d");break;
        case 45: printf("c");break;
        case 21: printf("b");break;
        case 44: printf("a");break;
        default: (*old9)(__CPPARGS); break;
    }
    c=inportb(0x61);
    outportb(0x61,c|0x80);
    outportb(0x61,c);
    outportb(0x20,0x20);
}

```