Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

**МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ**

Кафедра програмування та захисту інформації

**Звіт**

з виконаної лабораторної роботи № 5

дисципліни “ Архітектура комп’ютерів”

на тему

«Розробка та ініціалізація власних шрифтів і нестандартних символів

за допомогою програмування знакогенератора відеоадаптера ПК.»

Виконав :

студент академічної групи КІ-15

Аннаєв А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перевірив :

Викладач

Минайленко Р. М.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кропивницкий- 2017

Лабораторная работа № 5

**Тема**: Программирование последовательного порта передачи данных на плате IBM РС

**Цель**: Приобрести навыки по программированию асинхронного адаптера, организации передачи и приема данных с помощью последовательного интерфейса

Задание:

Разработать структурную схему организации двустороннего обмена информацией между двумя ПК, соединенными с помощью последовательного порта передачи данных (СОМ).

Разработать блок-схему алгоритма и программу, которая с помощью функций BIOS передает через СОМ-порт строку символов - ФИО студента.

Разработать блок-схему алгоритма и программу обеспечения одностороннего обмена информацией между двумя компьютерами через СОМ-порт: с клавиатуры первого ПК пользователем вводится строка символов посимвольно передает через СОМ-порт на второй ПК, который записывает оджержаний строку символов в текстовый файл.

Лістинг

#include<stdio.h>

#include<dos.h>

void main()

{

int i;

char mas[]={'A','n','n','a','e','v',' ','A','r','s','l','a', 'n'};

outportb(0x3fB,0x80);

outportb(0x3b8,0x30);

outportb(0x3f9,0x00);

outportb(0x3fB,0x03);

outportb(0x3Fc,0x00);

outportb(0x3f9,0x00);

i=0;

while(i<sizeof(mas))

{

do

{

outportb(0x3F8,mas[i]);

i++;

}while(inport(0x3FD) & 0x20)==0);

delay(100);

if((inport(0x3FD)& 0x10)==1)

printf("Обрывлинии");

}

}

#include<conio.h>

#include<dos.h>

voidmain()

{

unsigned char kod[8];

static unsigned char mas[20];

int i,z;

clrscr();

for(i=0;i<sizeof(kod);i++)

kod[i]=peekb(0xffff,0x05+i);

z=kod[3]\*10+kod[4];

mas[0]=kod[3];

mas[1]=kod[4];

mas[2]='';

z=(kod[0]-48)\*10+(kod[1]-48);

switch(z)

{

...

}

i=0;

outportb(0x2fb,0x80);

outportb(0x2f8,0x30);

outportb(0x2f9,0x00);

outportb(0x2fb,0x03);

outportb(0x2fc,0x00);

outportb(0x2f9,0x00);

while(i<sizeof(mas)/sizeof(char))

{

do

{

outportb(0x3f9,mas[i]);

i++;

}while((inport(0x3fd)&0x20)==0);

delay(100);

if(inport(0x3fd)&0x1)==1)

printf("Обрыв линии");

}

getchar();

}

}

Выводы: В ходе выполнения лабораторной работы 5 я приобрела навыки по программированию асинхронного адаптера, организации передачи и приема данных с помощью последовательного интерфейса.

Контрольные вопросы:

1. Что такое асинхронный адаптер, каково его назначение?

Это периферийное устройство компьютера, который предназначен для асинхронного приема и передачи данных на физическом уровне. Обычно он представляет собой отдельную плату или же расположен прямо на материнской плате компьютера. Его называют еще асинхронным адаптером RS-232-C, или портом RS-232-C. Каждый асинхронный адаптер обычно содержит несколько портов RS-232-C, через которые к компьютеру можно подключать внешние устройства.

2. Дайте определение последовательной передачи данных. Чем она отличается от параллельной?

Последовательная передача данных означает, что данные передаются с использованием единой линии.

3. Определите назначение стартового, стопового битов и бита четности.

Стартовый бит сигнализирует о начале передачи данных. Далее передаются биты данных, сначала младшие, потом старшие. Если используется бит четности P, то передается и он. Бит четности имеет такое значение, чтобы в пакете битов общее количество единиц (или нулей) была парная или нечетное.

В самом конце передаются один или два стоповых бита STOP, которые завершают передачу байта. Затем уровень линии передачи снова устанавливается в 1 до прихода следующего стартового бита.

4. В каких единицах измеряется скорость передачи данных?

Скорость передачи данных обычно измеряется в бодах. Боди - это количество битов, передаваемых в секунду. При этом учитываются и старт / стоп биты, а также бит четности.

5. Какая базовая адрес размещения асинхронного адаптера?

На этапе инициализации системы модуль POST BIOS тестирует имеющиеся асинхронные адаптеры и инициализирует первые два. Их базовые адреса располагаются в области данных BIOS начиная с адреса 0000: 0400h.

Первый адаптер COM1 имеет базовый адрес 3F8h и занимает диапазон адресов от 3F8h к 3FFh. Второй адаптер COM2 имеет базовый адрес 2F8h и занимает адреса 2F8h ... 2FFh.

6. Какие прерывания генерируют асинхронные порты COM1-COM4?

Асинхронные адаптеры могут генерировать прерывания:

- COM1 - IRQ4

- COM2 - IRQ3

- COM3 - IRQ4

- COM4 - IRQ3

7. Назначение порта 3F8h.

Этот порт соответствует регистру передаваемых данных. Для передачи в порт 3F8h необходимо записать передаваемый байт данных. После приема данных от внешнего устройства они могут быть прочитаны с этого порта.

8. Какие функции используются при работе с асинхронным адаптером? Через какое прерывание они доступны?

Эти функции доступны через прерывания INT14h.

Первая функция предназначена для инициализации портов асинхронного адаптера

На входе: AH = 00h;

                DX = номер порта: 0 - COM1, 1 - COM2;

                AL = параметриинициализации.

На выходе: AH = Состояние порта асинхронного адаптера;

                AL = состояние модема.

Для передачи байта используется следующая функция:

На входе: AH = 01h;

                DX = номер порта: 0 - COM1, 1 - COM2;

                AL = передаваемый байт.

На выходе: AL сохраняется;

                AH = состояние порта асинхронного адаптера,

Если бит 7 регистра AH установлен в 1,

Возникла ошибка.

Функция 02h предназначена для приема байта:

На входе: AH = 02h;

                DX = номер порта: 0 - COM1, 1 - COM2.

На выходе: AL = принят байт;

                AH = состояние порта асинхронного адаптера,

Если регистр AH не равно 0,

Возникла ошибка.

Состояние порта асинхронного адаптера можно определить с помощью функции 03h:

На входе: AH = 03h;

                DX = номер порта: 0 - COM1, 1 - COM2.

На выходе: AH = состояние порта асинхронного адаптера

                AL = состояние модема.

11. Сформулируйте порядок инициализации последовательного адаптера.

Первое, что должна сделать программа, работающая с асинхронным адаптером - установить протокол обмена и скорость передачи данных. После загрузки операционной системы для асинхронных адаптеров устанавливается скорость 2400 бод, не выполняется проверка на четность, используются один стоповый бит и восьмибитовых длина передаваемого символа. Вы можете изменить этот режим командой MS-DOS MODE.

Выполнив введение из порта 3FBh, программа может получить текущий режим адаптера. Для установки нового режима измените нужные вам поля и запишите новый байт режима по адресу 3FBh.

Если надо задать новое значение скорости обмена данными, перед записью байта режима установите старший бит этого байта в 1. Затем последовательно двумя командами вывода загрузите делитель частоты тактового генератора. Младший байт запишите в порт 3F8h, старший - в порт 3F9h.

Перед началом работы необходимо также инициализировать регистр управления прерываниями (порт 3F9h), даже если в вашей программе не используются прерывания от асинхронного адаптера. Если прерывания вам не нужны, запишите в этот порт значение 0.

На этом инициализацию можно считать законченной.