Белорусский Государственный Университет Информатики и Радиоэлектроники

Кафедра ЭВМ

Отчет по лабораторной работе № 1

Тема: «Программирование таймера»

Выполнил:

студент гр.XXXXXX

XXXXX X.X.

Проверил:

Третьяков А.Г.

Минск 2016

#### Постановка задачи.

#### Запрограммировать второй канал таймера таким образом, чтобы динамик компьютера издавал звуки. Для всех каналов таймера считать слово состояния и вывести его на экран в двоичной форме.

#### Алгоритм решения задачи.

#### Для того чтобы динамик компьютера издавал звуки, необходимо выполнить следующие действия:

#### Вывести в порт управляющего регистра с адресом 43h управляющее слово 10110110, соответствующее каналу 2, режиму 3.

#### Установить значение счётчика канала 2 таймера: в порт 42h вывести значение, полученное при разделении 1193180 на требуемую частоту в герцах, причём вначале вывести младший, а затем старший байты.

#### Установить в 1 два младших бита порта 61h для включения звука. Для этого вначале считывается байт из порта 61h в рабочую ячейку памяти, устанавливаются нужные биты, затем выводится новое значение байта в порт 61h.

#### Для выключения звука два младших бита порта 61h устанавливаются в 0.

#### Для чтения слова состояния каналов необходимо:

#### Вывести в порт управляющего регистра с адресом 43h управляющее слово, соответствующее команде RBC и номеру канала.

#### Затем из порта нужного канала выводится слово состояния.

#### Листинг программы.

#### #include <stdio.h>

#### #include <conio.h>

#### #include <dos.h>

#### #include <stdlib.h>

#### const int TONALITY = 1600;

#### const int SOUND\_DURATION\_MILISECONDS = 3000;

#### void SetTonality (int tonalityHerz);

#### void ReadStatusWords(void);

#### void EnableSound();

#### void DisableSound();

#### short HerzToShort(int);

#### void main()

#### {

#### char c;

#### clrscr();

#### 

#### outp(0x43, 0xB6); // channel 2, operation 4, mode 3, format 0

#### 

#### SetTonality(TONALITY);

#### 

#### printf("Press:\n'1' - Play sound\n'2' - Write status words\n'Esc' - quit\n\n");

#### 

#### while(c != 27)

#### {

#### c = getch();

#### switch(c)

#### {

#### case '1':

#### EnableSound();

#### delay(SOUND\_DURATION\_MILISECONDS);

#### DisableSound();

#### break;

#### case '2':

#### ReadStatusWords();

#### break;

#### case 27: break;

#### }

#### }

#### 

#### clrscr();

#### return;

#### }

#### void EnableSound()

#### {

#### char port61;

#### 

#### port61 = inp(0x61);

#### port61 = port61 | 3;

#### outp(0x61, port61);

#### }

#### 

#### void DisableSound()

#### {

#### char port61;

#### port61 = port61 & 0xFFFC;

#### outp(0x61, port61);

#### }

#### void SetTonality (int frequency)

#### {

#### short value = HerzToShort(frequency);

#### outp(0x42, (char)value); // low byte

#### outp(0x42, (char)(value >> 8)); // high byte

#### }

#### short HerzToShort(int value)

#### {

#### return 1193180/value;

#### }

#### void ReadStatusWords()

#### {

#### unsigned char temp;

#### char \*str;

#### int i;

#### str=(char\*)calloc(9, sizeof(char));

#### 

#### for(i =0;i<3;i++)

#### {

#### outp(0x43, 0xe2 + (i \* (i+1)));

#### temp = inp(0x40+i);

#### itoa(temp, str, 2);

#### printf("Channel state word %d: %s\n",i, str);

#### }

#### 

#### free(str);

#### 

#### return;

#### }

#### Результаты работы программы.

#### В программе реализовано меню, позволяющее выбрать тестируемый функционал (воспроизведение звука системным динамиком, вывод слов состояния каналов): '1' – издать звук, '2' – вывести слова состояния каналов. Выход из программы производится по нажатию Esc.