Министерство образования Республики Беларусь

Учреждение образования Белорусский государственный университет информатики и радиоэлектроники

Кафедра ЭВМ

Отчёт по лабораторной работе №3

«Программирование таймера»

Проверил: Выполнил:

к.т.н., доцент студент гр., 250502

Одинец Дмитрий Николаевич Грибовская А. А.

Минск 2024

**Задача**

1. Запрограммировать второй канал таймера таким образом, чтобы динамик компьютера издавал звуки.
2. Для всех каналов таймера считать слово состояния и вывести его на экран в двоичной форме.
3. Для всех каналов таймера рассчитать коэффициент деления (значение счетчика CE) и вывести его на экран в шестнадцатеричной форме.

**Листинг программы**

#include <stdio.h>

#include <conio.h>

#include <dos.h>

#include <stdlib.h>

#include <iostream.h>

int tones[] = {

262, 262, 294, 262, 349, 330, 262,

262, 262, 294, 262, 392, 349, 262,

262, 262, 523, 440, 349, 330, 294, 494,

466, 466, 440, 349, 392, 349

};

int durations[] = { 200, 200, 400, 400, 400, 800, 400,

200, 200, 400, 400, 400, 800, 400,

200, 200, 400, 400, 400, 400, 800, 400,

200, 200, 400, 400, 400, 800

};

void PlayTone(int frequency, int duration) {// звук одной ноты

int counter;

outp(0x43, 0xb6); // установка регистра управления таймера для канала 2, режим 3, двоичный формат

counter = (int)(1193180L / frequency);

outp(0x42, counter & 0x00ff); // установка младшего байта счетчика

outp(0x42, (counter & 0xff00) >> 8); // установка старшего байта счетчика

outp(0x61, inp(0x61) | 3); // включение динамика

delay(duration); // задержка на указанное время

outp(0x61, inp(0x61) & 0xfc); // выключение динамика все 1 кроме двух посл нулей

}

void CheckChannels() {

unsigned int temp, delayVal, delayLow, delayHigh;

char\* binaryString = new char[28];

unsigned i;

outp(0x43, 0xe2); // Установка команды для чтения состояния канала 0

temp = inp(0x40); // Чтение слова состояния канала 0

itoa(temp, binaryString, 2);

printf("Слово состояния канала 0: %s\n", binaryString);

outp(0x43, 0x06); // Фиксация канала 0

outp(0x43, 0x36); // Установка команды для чтения задержки канала 0

delayLow = inp(0x40); // Чтение младшего байта задержки

delayHigh = inp(0x40); // Чтение старшего байта задержки

delayVal = delayHigh \* 256 + delayLow;

itoa(delayVal, binaryString, 16);

printf("Задержка канала 0: %s\n", binaryString);

outp(0x43, 0xe4); // Установка команды для чтения состояния канала 1

temp = inp(0x41); // Чтение слова состояния канала 1

itoa(temp, binaryString, 2);

printf("Слово состояния канала 1: %s\n", binaryString);

outp(0x43, 0x46); // Фиксация канала 1

outp(0x43, 0x76); // Установка команды для чтения задержки между прерываниями канала 1

delayLow = inp(0x41); // Чтение младшего байта задержки

delayHigh = inp(0x41); // Чтение старшего байта задержки

delayVal = delayHigh \* 256 + delayLow;

itoa(delayVal, binaryString, 16);

printf("Задержка канала 1: %s\n", binaryString);

outp(0x43, 0xe8); // Установка команды для чтения состояния канала 2

temp = inp(0x42); // Чтение слова состояния канала 2

itoa(temp, binaryString, 2);

printf("Слово состояния канала 2: %s\n", binaryString);

outp(0x43, 0x86); // Фиксация канала 2

outp(0x43, 0xb6); // Установка команды для чтения задержки канала 2

delayLow = inp(0x42); // Чтение младшего байта задержки

delayHigh = inp(0x42); // Чтение старшего байта задержки

delayVal = delayHigh \* 256 + delayLow;

itoa(delayVal, binaryString, 16);

printf("Задержка канала 2: %s\n", binaryString);

delete[] binaryString;

}

int main() {

for (int i = 0; i < 28; i++) {

PlayTone(tones[i], durations[i]);

}

CheckChannels();

return 0;

}

**Тест**

