Министерство образования Республики Беларусь

Учреждение образования Белорусский государственный университет информатики и радиоэлектроники

Кафедра ЭВМ

Отчёт по лабораторной работе №3

«Программирование системного таймера»

Вариант 27

Проверил: Выполнил:

к.т.н., доцент студент гр.150504

Одинец Дмитрий Николаевич Шалаш А. С.

Минск 2023

**Задача**

***Первая часть (общее задание).*** Запрограммировать второй канал таймера таким образом, чтобы динамик компьютера издавал звуки.

***Вторая часть:***

* + Для всех каналов таймера считать слово состояния и вывести его на экран в двоичной форме.
* Для всех каналов таймера рассчитать коэффициент деления (значение счетчика CE) и вывести его на экран в шестнадцатеричной форме.

**Алгоритм**

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.
* Установить в 0 два младших бита порта 61h для выключения звука.

Для чтения слова состояния каналов необходимо:

* Вывести в порт управляющего регистра с адресом 43h управляющее слово, соответствующее команде RBC (*Чтение состояния канала)* и номеру канала.
* Вывести из порта нужного канала слово состояния.

**Листинг программы**

Далее приведен листинг программы, реализующей все поставленные задачи.

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <dos.h>

unsigned int notes[] = { 196, 261, 329, 196, 261, 329, 196, 261, 329 };

unsigned int note\_delay = 400;

void PlaySound();

void StateWords();

void CharToBin(unsigned char state, char\* str);

void TurnSpeaker(int isActive);

void SetCount(int iDivider);

void Menu();

void CE\_inp\_convert(unsigned char high, unsigned char low);

int main() {

Menu();

return 0;

}

void Menu() {

int choice = 0;

while (1) {

system("cls");

printf("1 - Play sound");

printf("\n2 - Print channels state words");

printf("\n0 - Exit");

printf("\n\nEnter choice: ");

scanf("%d", &choice);

if (choice >= 0 && choice <= 2) {

switch (choice) {

case 0:

return;

case 1:

PlaySound();

break;

case 2:

StateWords();

printf("\n\nPress any key to continue: ");

scanf("%d", &choice);

break;

}

}

}

}

void StateWords()

{

char\* bin\_state;

int iChannel;

unsigned char state;

bin\_state = (char\*)calloc(9, sizeof(char));

if (bin\_state == NULL)

{

printf("Memory allocation error");

exit(EXIT\_FAILURE);

}

for (iChannel = 0; iChannel < 3; iChannel++)

{

switch (iChannel)

{

case 0:

{

outp(0x43, 0xE2);

state = inp(0x40);

CharToBin(state, bin\_state);

printf("\nChannel 0x40 word: %s\n", bin\_state);

unsigned char low\_bits = inp(0x40);

unsigned char high\_bits = inp(0x40);

CE\_inp\_convert(high\_bits, low\_bits);

break;

}

case 1:

{

bin\_state[0] = '\0';

outp(0x43, 0xE4);

state = inp(0x41);

CharToBin(state, bin\_state);

printf("\nChannel 0x41 word: %s\n", bin\_state);

unsigned char low\_bits = inp(0x41);

unsigned char high\_bits = inp(0x41);

CE\_inp\_convert(high\_bits, low\_bits);

break;

}

case 2:

{

bin\_state[0] = '\0';

outp(0x43, 0xE8);

state = inp(0x42);

CharToBin(state, bin\_state);

printf("\nChannel 0x42 word: %s\n", bin\_state);

unsigned char low\_bits = inp(0x42);

unsigned char high\_bits = inp(0x42);

CE\_inp\_convert(high\_bits, low\_bits);

break;

}

}

}

free(bin\_state);

return;

}

void CE\_inp\_convert(unsigned char high, unsigned char low)

{

char\* CE\_response = (char\*)malloc(5 \* sizeof(char));

int i = 3;

while (i > 1)

{

char tmp1 = high & 15;

if (tmp1 >= 10)

{

CE\_response[i - 2] = 'A' + (tmp1 - 10);

}

else

{

CE\_response[i - 2] = '0' + tmp1;

}

char tmp2 = low & 15;

if (tmp2 >= 10)

{

CE\_response[i] = 'A' + (tmp2 - 10);

}

else

{

CE\_response[i] = '0' + tmp2;

}

i--;

high >>= 4;

low >>= 4;

}

CE\_response[4] = 0;

printf("CE register: 0x%s\n", CE\_response);

}

void CharToBin(unsigned char state, char\* str)

{

int i, j;

char temp;

for (i = 7; i >= 0; i--)

{

temp = state % 2;

state /= 2;

str[i] = temp + '0';

}

str[8] = '\0';

}

void SetCount(int iDivider) {

long base = 1193180;

long kd;

outp(0x43, 0xB6);

kd = base / iDivider;

outp(0x42, kd % 256);

kd /= 256;

outp(0x42, kd);

return;

}

void TurnSpeaker(int isActive) {

if (isActive) {

outp(0x61, inp(0x61) | 3);

return;

}

else {

outp(0x61, inp(0x61) & 0xFC);

return;

}

}

void PlaySound() {

for (int i = 0; i < 9; i++) {

SetCount(notes[i]);

TurnSpeaker(1);

delay(note\_delay);

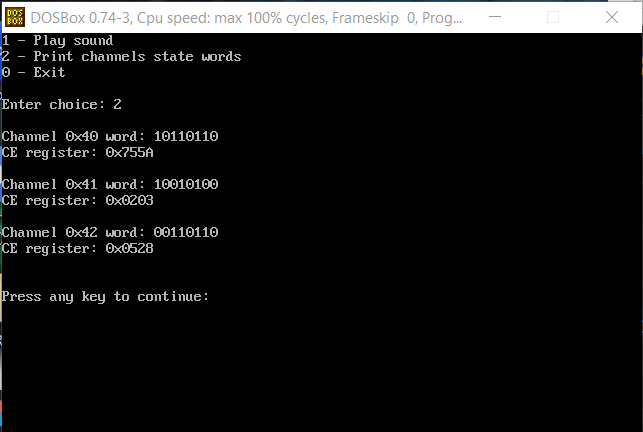
TurnSpeaker(0);

}

}

**Тест**

Во время работы программы происходит звучание системного динамика. Также для всех каналов таймера выводится на экран в двоичной форме слово состояния и значение счетчика CE в шестнадцатеричной форме:

****

**Заключение**

В ходе лабораторной работы удалось запрограммировать второй канал таймера таким образом, чтобы динамик компьютера издавал звук, а также для всех каналов таймера было считано слово состояния и выведено на экран в двоичной форме, было рассчитано значение счетчика CE и выведено на экран в шестнадцатеричной форме.

Программа компилировалась в Turbo C++ и запускалась в DOS, который эмулировался с помощью DosBox.