

Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Лабораторная работа №4

«Программирование системного таймера»

Вариант 14

Выполнил:

Студент группы 050504

Матусевич С.К.

Проверил:

Преподаватель

Одинец Д.Н.

Минск, 2022

## 1. 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.

## 2. 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 делении 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.

## 3. 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();

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;
            }
        }
    }
}

//функция считывающая слова состояния каналов
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); //заносим управляющее слово,
                //соответствующее команде RBC (Чтение состояния канала) и
                //номеру канала 0
                state = inp(0x40); //чтение слова состояния канала 0
                CharToBin(state, bin_state);
                printf("Channel 0x40 word: %s\n", bin_state);
                break;
            }
            case 1:
            {

```

```

        bin_state[0] = '\0';
        outp(0x43, 0xE4); //заносим управляющее слово,
        //соответствующее команде RBC (Чтение состояния канала) и
номеру канала 1
        state = inp(0x41); //чтение слова состояния канала 1
        CharToBin(state, bin_state);
        printf("Channel 0x41 word: %s\n", bin_state);
        break;
    }
    case 2:
    {
        bin_state[0] = '\0';
        outp(0x43, 0xE8); //заносим управляющее слово,
        //соответствующее команде RBC (Чтение состояния канала) и
номеру канала 2
        state = inp(0x42); //чтение слова состояния канала 2
        CharToBin(state, bin_state);
        printf("Channel 0x42 word: %s\n", bin_state);
        break;
    }
}
free(bin_state);
return;
}

```

//функция перевода в двоичный код

```

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';
}

```

//функция установки значения счетчика

```

void SetCount(int iDivider) {
    long base = 1193180; //максимальная частота
    long kd;
    outp(0x43, 0xB6); //10110110 - канал 2, операция 4, режим 3, формат 0
    kd = base / iDivider;
    outp(0x42, kd % 256); // младший байт делителя
    kd /= 256;
    outp(0x42, kd); //старший байт делителя
    return;
}

```

//функция работы с громкоговорителем

```

void TurnSpeaker(int isActive) {
    if (isActive) {
        outp(0x61, inp(0x61) | 3); //устанавливаем 2 младших бита 11
        return;
    } else {
        outp(0x61, inp(0x61) & 0xFC); //устанавливаем 2 младших бита 00
        return;
    }
}

```

```

    }
}

//функция воспроизведения песни
void PlaySound() {
    for (int i = 0; i < 9; i++) {
        SetCount(notes[i]);
        TurnSpeaker(1); //включаем громкоговоритель
        delay(note_delay); //устанавливаем длительность мс
        TurnSpeaker(0); //выключаем громкоговоритель
    }
}

```

#### 4. Тестирование программы

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

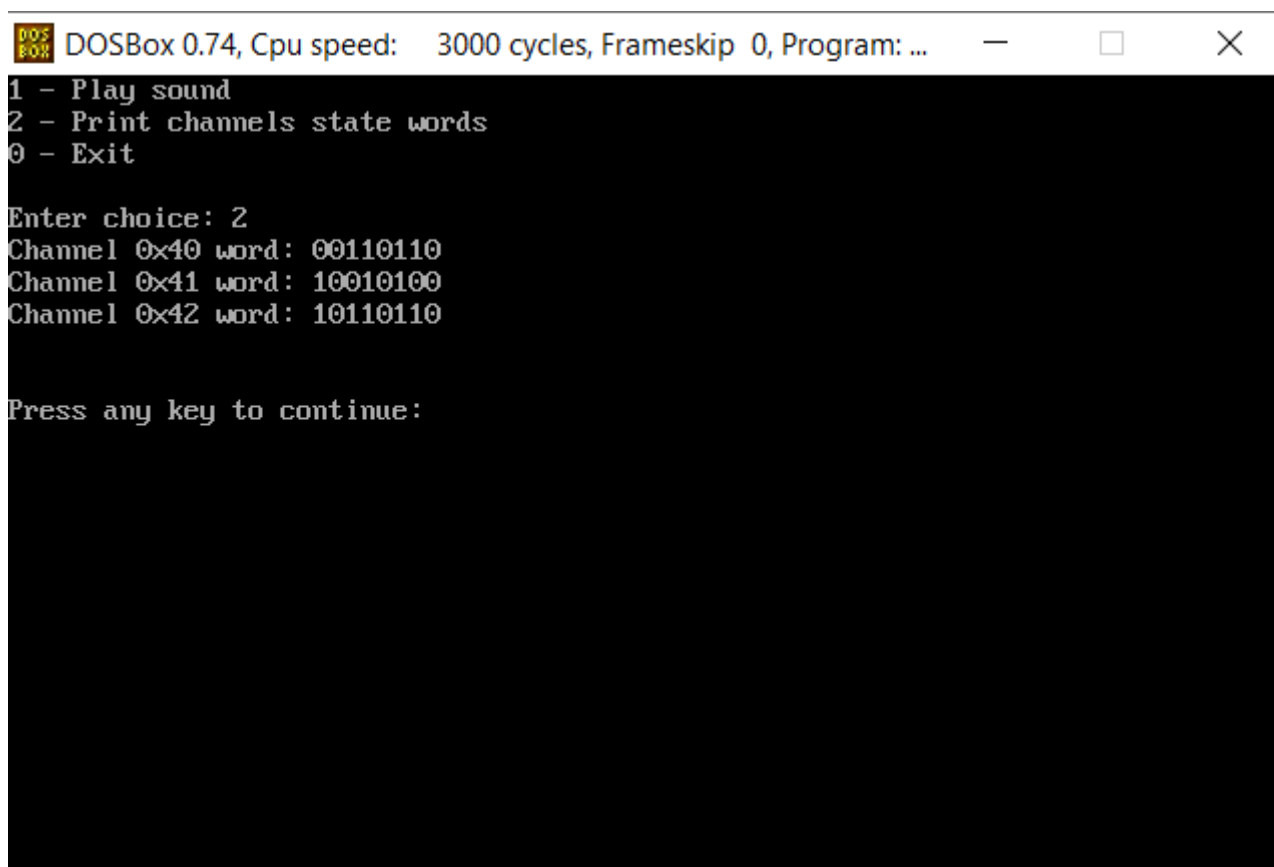


Рисунок 4.1 – Результат работы программы при выводе слов состояния каналов таймера.

## **5. 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.

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