БГУИР

Кафедра ЭВМ

Отчет по лабораторной работе №4

Тема: «Системный таймер»

Выполнил:

студент группы 750501 Новицкий А.Д.

Проверил:

к.т.н., доцент Одинец Д.Н.

Минск

2019

**СОДЕРЖАНИЕ**

1. Постановка задач………………………………………………………………..3

2. Алгоритм решения задачи…………………………………………………...4-5

3. Листинг программы…………………………………………………………..6-8

4. Тестовые примеры……………………………………………………………...9

5. Заключение…………………………………………………………………….10

**ПОСТАНОВКА ЗАДАЧ**

Под MS DOS написать программу, которая:

1) c помощью системного таймера генерирует звук заданной частоты (по вариантам);

2) выводит слово состояния для каждого канала в двоичном виде;

3) определяет коэффициент деления для каждого канала в 16-ричном виде.

\*Реализовать генератор случайных чисел от нуля до заданного с клавиатуры числа.

**АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ**

Генерация звука:

1. Записать в управляющий регистр таймера слово 10 11 011 0 через порт 0x43, что означает - выбор третьего канала для последующей записи, в который будет записана сначала младшая половина и затем старшая половина, установка режима генерации прямоугольных импульсов, 16-битный режим
2. Получить делитель частоты поделив частоту таймера на частоту, которую необходимо получить
3. Записать в третий канал системного таймера младшую часть делителя частоты через порт 0x42, а затем – старшую
4. Включить динамик, записав в два младших бита порта 0x61 значение “1”
5. Выполнить необходимую задержку
6. Включить динамик, записав в два младших бита порта 0x61 значение “0”

Получение слова состояния:

1. Записать в управляющий регистр таймера слово 11 1 0 001 0 через порт 0x43, что означает – режим чтения из первого канала без фиксации текущего значения в 16 битном режиме.
2. Получить значение первого канала системного таймера через порт 0x40
3. Записать в управляющий регистр таймера слово 11 1 0 010 0 через порт 0x43, что означает – режим чтения из второго канала без фиксации текущего значения в 16 битном режиме.
4. Получить значение второго канала системного таймера через порт 0x41
5. Записать в управляющий регистр таймера слово 11 1 0 100 0 через порт 0x43, что означает – режим чтения из третьего канала без фиксации текущего значения в 16 битном режиме.
6. Получить значение третьего канала системного таймера через порт 0x42

Определение коэффициента деления:

1. Получить делитель частоты, поделив частоту системного таймера на максимально допустимое значение
2. Записать в управляющий регистр таймера слово 10 11 010 0 через порт 0x43, что означает – режим записи во третий канал сначала младшей половины, а затем старшей половины делителя частоты, установка режима генерации синусоидальных импульсов
3. Включить системный таймер, установив младший бит в порте 0x61 в значение “1”
4. Записать в управляющий регистр таймера слово 10 00 0000 через порт 0x43, что означает – выбор третьего канала, его блокировка и последующее чтение младшей и старшей части
5. Чтение младшей половины из канала 3 через порт 0x42
6. Чтение старшей половины из канала 3 через порт 0x42
7. Получение полного значения и его вывод
8. Отключение системного таймера установив младший бит в порте 0x61 в значение “0”

**ЛИСТИНГ ПРОГРАММЫ**

#include <dos.h>

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#define TimerFrequensy 1193180

int frequencyArray[9] = { 293,392,293,392,523,659,880,784,698 };

int delayArray[9] = { 200,400,200,400,200,200,400,200,200 };

void generateSound(int frequency, int d);

void print(int state);

void strcan();

void getRandomValue(long val);

void kdcan();

int main()

{

int maximum;

for (int i = 0; i < 5; i++) {

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

generateSound(frequencyArray[j], delayArray[j]);

}

delay(100);

}

strcan();

printf("\n");

kdcan();

printf("Get max val: ");

scanf("%d", &maximum);

getRandomValue(maximum);

return 0;

}

void generateSound(int frequency, int d)

{

int frequencyDivider;

outp(0x43, 0xB6);

frequencyDivider = TimerFrequensy / frequency;

outp(0x42, frequencyDivider % 256);

frequencyDivider /= 256;

outp(0x42, frequencyDivider);

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

delay(d);

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

}

void print(int state)

{

int i;

char temp[8];

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

{

temp[i] = state % 2 + '0';

state /= 2;

}

for (i = 0; i < 8; i++)

{

printf("%c", temp[i]);

}

printf("\n");

}

void strcan()

{

int state;

outp(0x43, 0xE2);

state = inp(0x40);

printf("Channel 1: ");

print(state);

outp(0x43, 0xE4);

state = inp(0x41);

printf("Channel 2: ");

print(state);

outp(0x43, 0xE8);

state = inp(0x42);

printf("Channel 3: ");

print(state);

}

void getRandomValue(long val)

{

long Low, High, Full, maxVal;

outp(0x43, 0xB4);

maxVal = TimerFrequensy / val;

outp(0x42, maxVal % 256);

maxVal /= 256;

outp(0x42, maxVal);

outp(0x61, inp(0x61) | 0x01);

while (i < 10)

{

i++;

delay(i);

outp(0x43, 0x86);

Low = inp(0x42);

High = inp(0x42);

Full = High \* 256 + Low;

printf("%ld\n", Full);

}

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

}

void kdcan()

{

long kd, kdHigh, kdLow, kdMax;

kdMax = 0;

for (long i = 0; i < 65535; i++)

{

outp(0x43, 0x00);

kdLow = inp(0x40);

kdHigh = inp(0x40);

kd = kdHigh \* 256 + kdLow;

if (kd > kdMax) kdMax = kd;

}

printf("%hx\n", kdMax);

kdMax = 0;

for (i = 0; i < 65535; i++)

{

outp(0x43, 0x40);

kdLow = inp(0x41);

kdHigh = inp(0x41);

kd = kdHigh \* 256 + kdLow;

if (kd > kdMax) kdMax = kd;

}

printf("%hx\n", kdMax);

kdMax = 0;

for (i = 0; i < 65535; i++)

{

outp(0x43, 0x80);

kdLow = inp(0x42);

kdHigh = inp(0x42);

kd = kdHigh \* 256 + kdLow;

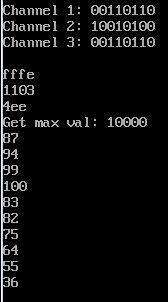
if (kd > kdMax) kdMax = kd;

}

printf("%hx\n", kdMax);

}

**ТЕСТОВЫЕ ПРИМЕРЫ**

****

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения лабораторной работы мной был получен опыт работы с системным таймером и динамиком.

Существует шесть режимов работы системного таймера:

* Генерация прерывания IQ0 при установке счетчика в значение “0”
* Установка в режим ждущего мультивибратора
* Установка в режим генератора импульсов
* Установка в режим генератора прямоугольных импульсов
* Установка в режим программно-зависимого одновибратора
* Установка в режим аппаратно-зависимого одновибратора

Системный таймер имеет три канала которые служат для отслеживания

текущего времени от момента включения компьютера, для работы с контроллером прямого доступа к памяти, для управления системным динамиком.

Основные назначения таймера:

* Организация часов реального времени
* Программирование генератора прямоугольных и синусоидальных импульсов
* Счетчик событий таймера
* Управление двигателями флоппи-дисководов

Для программирования системного таймера необходимо записать управляющее слово в порт 0x43, а затем в один из портов 0x40, 0x41, 0x42.

Для генерации звука системный таймер вырабатывает необходимую частоту, которая затем через второй канал передается в динамик.

Из байта состояния можно получить: формат представления данных (16-битное или двоично-десятичное), режим работы таймера, тип операции, готовность счетчика для считывания, состояние выходного сигнала в момент блокировки счетчика импульсов.

Максимальная частота, которую может сгенерировать с помощью системного таймера равна 1 193 180 Гц, а минимальная – 18,2 Гц.