Министерство образования Республики Беларусь

Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра электронно-вычислительных машин

Лабораторная работа №7

«Работа видеоподсистемы»

Выполнил:

студент группы 724402

Чернявский Я.А.

Проверил:

к.т.н., доцент Селезнёв И.Л.

Минск  
 2019

1. **Цель работы**

Изучить работу видеоподсистемы персонального компьютера используя функции BIOS для управления режимами работы.

1. **Описание алгоритма**
2. Вызвать прерывание INT 10h для работы с видеоадаптером.
3. Установить режим отображения на экране и цвет фона.
4. Перейти к рисованию фигуры.
5. **Теоретические сведения**

**Типы видеоадаптеров:**

Видеосистема имеет две основные части: видеоадаптер и монитор (дисплей). В DOS видеоадаптер называется консолью (CON), другой составной частью которой является клавиатура. Видеоадаптер может быть реализован в виде отдельной платы или встроен в видеосистему. Развитие адаптеров (видеоадаптеров) происходит путем совершенствования предыдущего адаптера (расширения функциональных возможностей) и сохранения программной совместимости. Существуют следующие типы видеоадаптеров (стандартов): монохромный адаптер дисплея и принтера (MDA), цветной графический адаптер (CGA), усовершенствованный графический адаптер (EGA), видеографическая матрица (VGA) и т. д. Некоторые показатели адаптеров приведены в Таблице 3.1.

Таблица 3.1 – Показатели адаптеров



**MDA** (Monochrome Display Adapter – монохромный адаптер дисплея) – простейший видеоадаптер, применявшийся в первых IBM PC. Работает в текстовом режиме с разрешением 80х25 (столбцов-строк), поддерживает 5

атрибутов текста: обычный, яркий, инверсный, подчеркнутый и мерцающий.

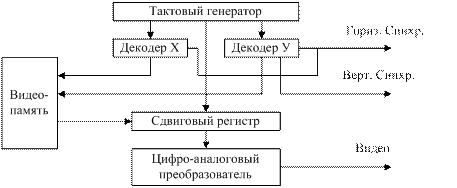


Рисунок 3.1 – Структурная схема простейшего видеоконтроллера

**HGC** (Hercules Graphics Card - графическая карта Hercules) - расширение MDA с графическим черно-белым режимом 720x348, разработанное фирмой Hercules.

**CGA** (Color Graphics Adapter - цветной графический адаптер) - первый адаптер с графическими возможностями. Текстовый режим 40x25 (столбцов и строк) и 80x25 (матрица символа - 8x8), графический режим с разрешениями 320x200 или 640x200. Наибольшая цветовая глубина — 4 [бита](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82) (16 цветов) (Рисунок 3.2).



Рисунок 3.2 - 16-цветная палитра CGA

**Стандартные текстовые режимы:**

1. 40×25 символов, 16 цветов.

Каждый символ имеет размер 8×8 точек. Эффективное разрешение экрана — 320×200 пикселов (пропорции пиксела — 1:1,2), при этом невозможно обращение к каждому пикселу отдельно. Всего доступно 256 различных символов, начертания которых хранятся в [ПЗУ](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%BD%D0%BE%D0%B5_%D0%B7%D0%B0%D0%BF%D0%BE%D0%BC%D0%B8%D0%BD%D0%B0%D1%8E%D1%89%D0%B5%D0%B5_%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE) видеокарты . Для каждого выводимого символа возможно задать цвет самого символа и цвет фона, оба цвета выбираются из палитры. Видеокарта обладает достаточным объёмом [ОЗУ](https://ru.wikipedia.org/wiki/%D0%9E%D0%97%D0%A3) для хранения восьми видеостраниц.

1. 80×25 символов, 16 цветов.

Используется тот же набор символов, что и для режима 40×25. Эффективное разрешение экрана — 640×200 пикселов (пропорции пиксела — 1:2,4), также невозможно обращение к отдельным пикселам. Так как на экран возможно вывести вдвое больше символов, ОЗУ видеокарты достаточно для хранения четырёх видеостраниц.

**Стандартные графические режимы:**

1. Режим низкого разрешения

320×200 пикселей, так же, как и у текстового режима 40×25.

Несмотря на узкую палитру, CGA отличался от других видеосистем того времени тем, что возможно обращение к любому отдельно взятому пикселю, без каких-либо [конфликтных зон](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%84%D0%BB%D0%B8%D0%BA%D1%82_%D0%B0%D1%82%D1%80%D0%B8%D0%B1%D1%83%D1%82%D0%BE%D0%B2). Одновременно можно использовать только четыре цвета, которые нельзя выбрать самостоятельно — для данного режима определены две палитры (Рисунок 3.3):

Палитра № 1: пурпурный, сине-зелёный, белый и цвет фона (по умолчанию — чёрный).

Палитра № 2: красный, зелёный, коричневый/жёлтый и цвет фона (по умолчанию — чёрный).

При установке бита интенсивности доступны яркие варианты палитр.

1. Режим высокого разрешения

640×200 пикселей, как и у текстового режима 80×25.

Этот режим монохромный, доступны только белый и чёрный цвет (цвета можно изменить).

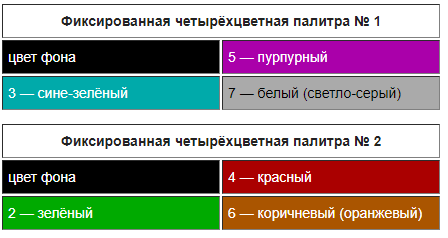


Рисунок 3.3 – Фиксированные палитры CGA

**EGA** (Enhanced Graphics Adapter - улучшенный графический адаптер) - дальнейшее развитие CGA. В текстовых режимах дает формат 80x25 (матрица символа 8x14). Графический режим с разрешением 640x350. Количество одновременно отображаемых цветов - 16, палитра расширена до 64 цветов (Рисунок 3.4).

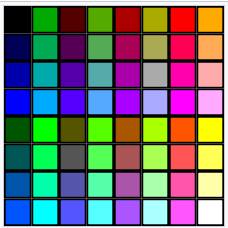


Рисунок 3.4 – Палитра цветов EGA

В текстовых режимах используется два типа шрифтов. Стандартный шрифт EGA формируется матрицей 7×9 в ячейке 8×14 пикселов. Для совместимости с CGA используется шрифт с матрицей 7×7 в ячейке 8×8 пикселов.

EGA — первый видеоадаптер IBM, позволяющий программно менять шрифты текстовых режимов.

Адаптером поддерживались три текстовых режима. Первые два были стандартными:

* с разрешением 80x25 символов и 640x350 пикселей;
* с разрешение 40x25 символов и 320x200 пикселей.

А вот разрешение третьего режима составляло 80x43 символов и 640x350 пикселей. Для его использования требовалась предварительная установка режима 80×25 и загрузка шрифта 8×8 с помощью команды BIOS.

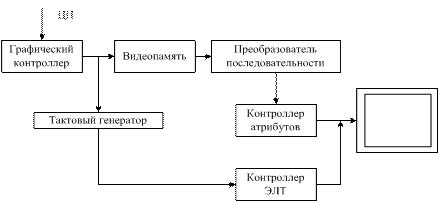


Рисунок 3.5 – Структурная схема видеоадаптера EGA-VGA

**MCGA** (Multicolor Graphics Adapter - многоцветный графический адаптер) - введен фирмой IBM в ранних моделях PS/2.

**VGA** (Video Graphics Array – массив визуальной графики) расширение MCGA совместимое с EGA. Добавлен текстовый режим 720x400 для эмуляции MDA и графический режим 640x480.

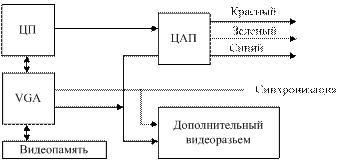


Рисунок 3.6 – Структурная схема видеоадаптера VGA

**SVGA** (Super VGA - "сверх" VGA) - расширение VGA с добавлением более высоких разрешений. Видеорежимы добавляются из ряда 800x600, 1024x768, 1152x864, 1280x1024, 1600x1200 - все с соотношением 4:3. Цветовое пространство расширено до 65536 (High Color) или 16 млн (True Color).

**Типы дисплеев:**

*1. Композитный дисплей*

Имеет один аналоговый вход. Видеосигнал поступает в дисплей в стандарте NTSC (National Television System Commitete). Стандарт NTSC используется в бытовом телевидении. Композитный дисплей обычно применяется совместно с видеоадаптером CGA.

*2. Цифровой дисплей*

Имеет от одной до шести входных линий. На цифровом дисплее может отображаться до 2n различных цветов, где n равно количеству входных линий. Данный тип дисплеев используется вместе с видеоадаптерами CGA и EGA.

*3. Аналоговый RGB дисплей*

Имеет три аналоговые входные линии, управляющие красным, зеленым и синим цветами. Уровень напряжения на каждой линии отвечает за интенсивность соответствующего цвета на экране. Количество цветов, которые может отображать аналоговый дисплей, фактически ограничено только возможностями видеоадаптера. Аналоговый дисплей используется совместно с VGA, SVGA, графическими сопроцессорами, акселераторами Windows и видеоадаптерами на локальной шине.

**Поддержка вывода BIOS**

Для более гибкого управления процессом вывода текста на экран предназначены функции BIOS, входящие в группу "video Services" (int 10h) и выполняющие следующие действия:

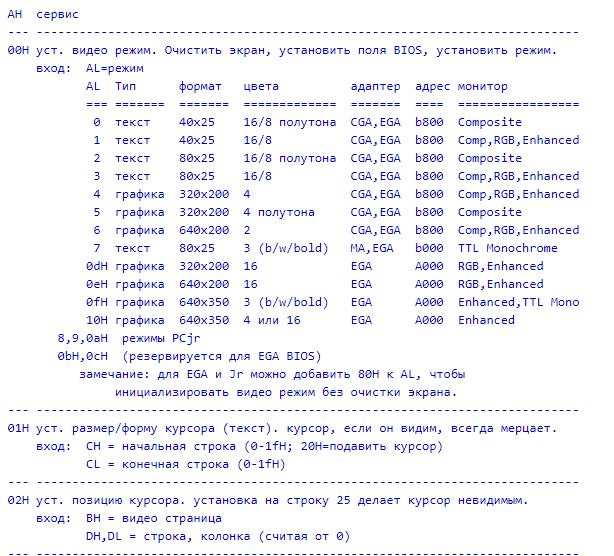
* 09h -- вывод символа и атрибута без перемещения курсора, страница 0;
* 0Аh -- вывод символа без атрибута без перемещения курсора, страница 0;
* 0Еh -- вывод символа без атрибута с перемещением курсора, страница 0;
* 13h -- вывод строки символов с атрибутами на указанную страницу.

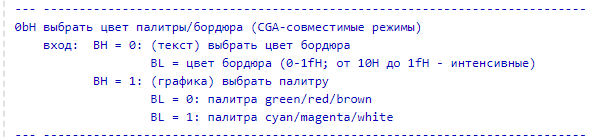
Перед вызовом функции 09h и ОАh в регистры записываются следующие величины: в ah — код функции (09h или OAh); в ai — код выводимого символа (ASCII); bb — не используется; в сх — количество повторов символа; bl — код атрибута, который нужен только для функции 09h.

Для доступа к функциям BIOS видеоадаптера предназначено прерывание INT 10h. Загрузите в регистр AH номер функции BIOS видеоадаптера, которую необходимо вызвать, загрузите остальные регистры процессора в соответствии с вызываемой функцией и выполните прерывание INT 10h.

Большинство функций BIOS видеоадаптера не изменяют содержимое неиспользуемых регистров процессора, однако рекомендуется сохранить наиболее важные регистры.

* Выбор режима работы - функция 00h
* Изменение формы курсора - функция 01h
* Изменение положения курсора - функция 02h
* Определение положения и формы курсора - функция 03h
* Использование светового пера - функция 04h
* Изменение активной страницы видеопамяти - функция 05h
* Свертка текстового окна вверх - функция 06h
* Свертка текстового окна вниз - функция 07h
* Чтение символа и его атрибутов - функция 08h
* Запись символа с атрибутами в текущей позиции курсора - функция 09h
* Запись символа в текущей позиции курсора - функция 0Ah
* Установка цветовой палитры (режимы 4,5,6) - функция 0Bh
* Вывод пиксела - функция 0Ch
* Чтение пиксела - функция 0Dh
* Запись символа в режиме телетайпа - функция 0Eh
* Определение текущего режима работы видеоадаптера - функция 0Fh





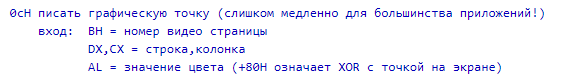


Рисунок 3.7 - функции INT 10H

**Режимы работы видеоадаптеров**

Все видеорежимы делятся на: графические и текстовые. Причём в различных режимах видеоадаптер современных ПК используются разные механизмы формирования видеосигнала, а монитор в обоих режимах работает одинаково.

* ***Графический режим***– является основным режимом работы видеосистемы современного ПК. В таком режиме на экран монитора можно вывести текст, рисунок, фотографию и т. д. В графическом режиме в каждой ячейке кадрового буфера (матрицы NxM k-разрядных чисел) содержит код цвета соответственного пикселя экрана. Разрешение экрана при этом также равно NxM. Адресуемым элементом экрана является минимальный элемент изображения-пиксел. По этой причине графический режим называется АРА. Иногда число k называется глубиной цвета. При этом количество одновременного отображения цветов равно 2n , а размер кадрового буфера, необходим для хранения цветных изображений с разрешением N x M и глубины цвета k, составляет N x M бит.
* ***Текстовый режим.***В таком (символьном) режиме, как и в графическом, изображение на экране монитора представляет собой множество пикселей и характеризуется разрешением NxM. Однако все пиксели на группы, называются знакоместами или символьными позициями, размером p\*q. В каждом из знакомест может быть изображен один из 256 символов. Таким образом на экране умещается M/q=M, символов строк по N/p символов в каждой.N/p-столбцов и M/q-строк.



Рисунок 3.7 - Стандартные режимы работы видеоадаптеров

1. **Листинг кода программы**

//724402-2 Chernyavsky Y.A.

//L7 Rabota Videopodsistems

#include <dos.h>

#include <stdio.h>

#include <conio.h>

#define VIDEO (char far\*)0xb8000000

void print(char\*str);

void printInfo(void);

void CGA(void);

void EGA(void);

void myPutsXY(char\*, int, int); //вывод строки в конкретное место

void myclear(int, int, int); //очистка области экрана (начальная строка, начальный столбец, длина)

void top(void); //вывод шапки

volatile int Column = 0;

volatile int Line = 0;

int main(void)

{

int flag = 1;

char ch = 0;

while (flag)

{

myclear(0, 0, 2000);

top();

ch = getch();

switch (ch)

{

case '1':

CGA();

break;

case '2':

EGA();

break;

case '0':

flag = 0;

break;

}

}

myclear(0, 0, 2000);

return 0;

}

void EGA(void) { //режим EGA

union REGS inregs, outregs;

unsigned char j, color = 1;

inregs.h.ah = 0x00; //Выбор режима

inregs.h.al = 0x0e; //установка режима EGA

int86(0x10, &inregs, &outregs);

inregs.h.ah = 0x0B;

inregs.h.bh = 0x00; //задание цвета фона

inregs.h.bl = 0x07; //номер устанавливаемого цвета фона

int86(0x10, &inregs, &outregs);

while (1) {//отрисовка линии

if (kbhit() && getch() == 48)

break;

inregs.h.ah = 0x0C;

inregs.h.al = color++;//задание цвета линии

int86(0x10, &inregs, &outregs);

for (j = 0; j < 200; j++) {

inregs.h.ah = 0x0c; // функция отрисовки пикселя

if (color > 15) color = 1;

if (color == 7) color = 8;

inregs.h.al = color; // задание цвета линии

inregs.x.cx = 0x30 + j; // x координата

inregs.x.dx = 0xaa - j / 3.3; // y координата

int86(0x10, &inregs, &outregs);

inregs.x.cx = 0x30 + j; // x координата

inregs.x.dx = 0x30 + j / 3.3; // y координата

int86(0x10, &inregs, &outregs);

if (j < 123)

{

inregs.x.cx = 0x30; // x координата

inregs.x.dx = 0x30 + j; // y координата

int86(0x10, &inregs, &outregs);

}

if (j < 100)

{

inregs.x.cx = 0x30 + 200; // x координата

inregs.x.dx = 0x30 + j + 10; // y координата

int86(0x10, &inregs, &outregs);

}

color++;

}

delay(300);

}

inregs.h.ah = 0x0;

inregs.h.al = 0x3;

int86(0x10, &inregs, &outregs);

}

void CGA(void) { //режим CGA

union REGS inregs, outregs;

unsigned char i, j, color;

color = 1;// номер цвета

inregs.h.ah = 0x00;

inregs.h.al = 0x04;//установка режима CGA

int86(0x10, &inregs, &outregs);

inregs.h.ah = 0x0B; //выбор палитры

inregs.h.bh = 0x00; //задание цвета фона

inregs.h.bl = 0x00; //номер устанавливаемого цвета фона(bl,gr,r,br)

int86(0x10, &inregs, &outregs);

while (1) {//отрисовка прямоугольника

if (kbhit() && getch() == '0')

break;

inregs.h.ah = 0x0c; //вывод пикселя

inregs.h.al = color;//задание цвета линии

int86(0x10, &inregs, &outregs);

for (i = 0; i < 100; i++) {

for (j = 0; j < 100; j++) {

inregs.x.cx = 0x70 + i;//х координата

inregs.x.dx = 0x30 + j;// у координата

int86(0x10, &inregs, &outregs);

}

}

delay(800);

color++;

if(color > 3) color = 1;

}

inregs.h.ah = 0x00;

inregs.h.al = 0x03;//переход в текстовый режим

int86(0x10, &inregs, &outregs);

}

void myPutsXY(char \*str, int row, int col)

{

int i;

char far\* v = (char far \*)0xb8000000 + row \* 160 + col \* 2;

for (i = 0; str[i] != '\0'; i++)

{

\*v = str[i];

v += 2;

}

}

void myclear(int row, int col, int len)

{

int i;

char far\* v = (char far\*)0xb8000000 + row \* 160 + col \* 2;

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

{

\*v = ' ';

v += 2;

}

}

void top(void)

{

char\* str0 = "724402-2 Chernyavsky Y.A.";

char\* str1 = "L7 Rabota Videopodsistem";

char\* str2 = "1 - CGA (Color Graphics Adapter) mod 4 |320x200|";

char\* str3 = "2 - EGA (Enhanced Graphics Adapter) mod E |640x200|";

char\* str4 = "0 - Exit";

myPutsXY(str0, 0, 0);

myPutsXY(str1, 1, 0);

myPutsXY(str2, 2, 0);

myPutsXY(str3, 3, 0);

myPutsXY(str4, 4, 0);

}

1. **Скриншоты выполнения программы**

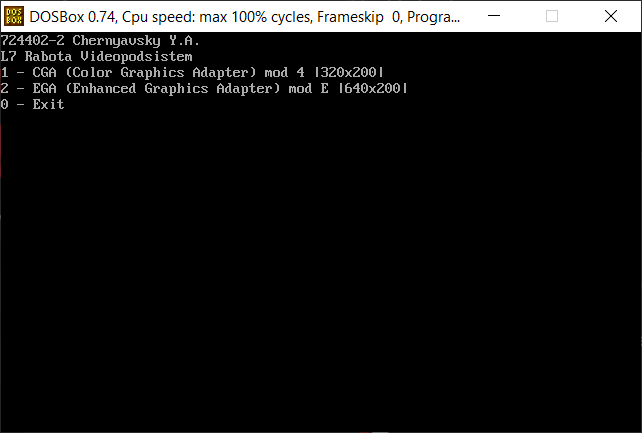


Рисунок 5.1 – Начало программы

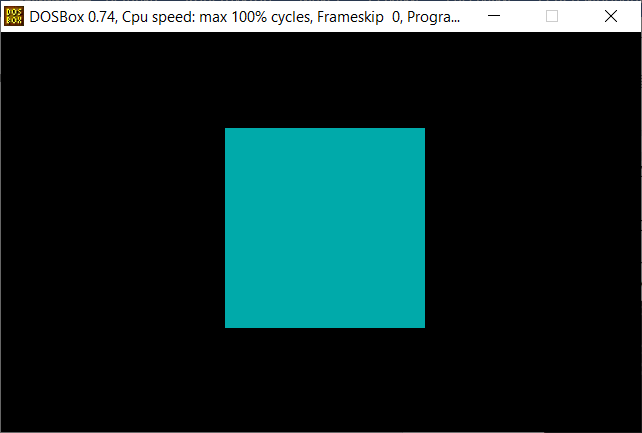


Рисунок 5.2 – Квадрат (Режим CGA)

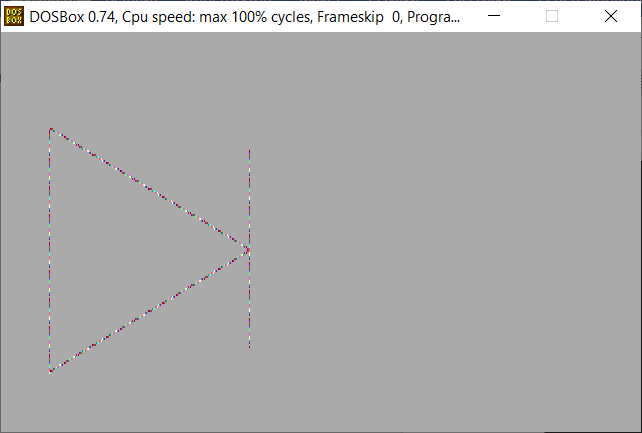


Рисунок 5.3 – Фигура (Режим EGA)

1. **Выводы**

В ходе выполнения лабораторной работы была изучена работа видеоподсистемы персонального компьютера, использованы функции BIOS для управления режимами работы.