

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра информационных систем

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Организация ЭВМ и систем»
Тема: Исследование видеосистемы (графический режим)

Студенты гр. 3372

Преподаватель

Беляев К.В.
Лазарев Ф.Н.

Кочетков А.В

Санкт-Петербург

2024

Цель работы.

Изучение работы с видеосистемой в графическом режиме, вывод графика заданной функции с масштабированием и разметкой осей.

Задание на лабораторную работу.

1. Разработать программу для вывода на экран графика заданной функции (рисунок 1).

1	$\text{Sin}^2(x/2) + \text{Sqrt}(x)$	$3\pi/2$	15π
---	--------------------------------------	----------	---------

Рисунок 1 – Заданная функция (вариантное задание)

2. Произвести разметку осей и проставить истинные значения точек.
3. Найти максимальное значение функции на заданном интервале и вывести в отдельное окно на экране.

Структурная схема аппаратных средств

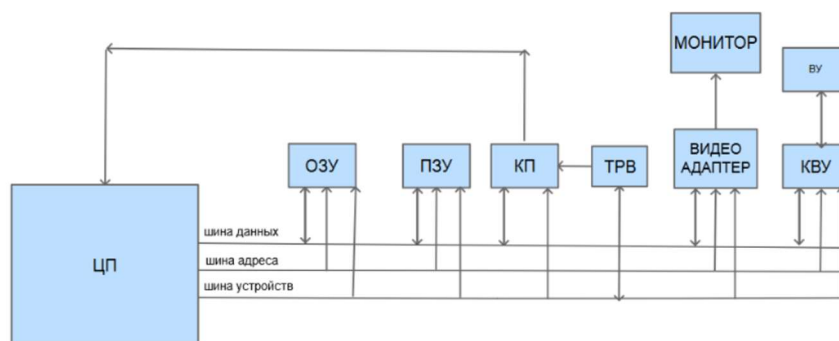


Рисунок 2 – Структурная схема аппаратных средств

Пример запуска программы

1. При запуске программы отображается график заданной функции согласно варианту (рисунок 3):

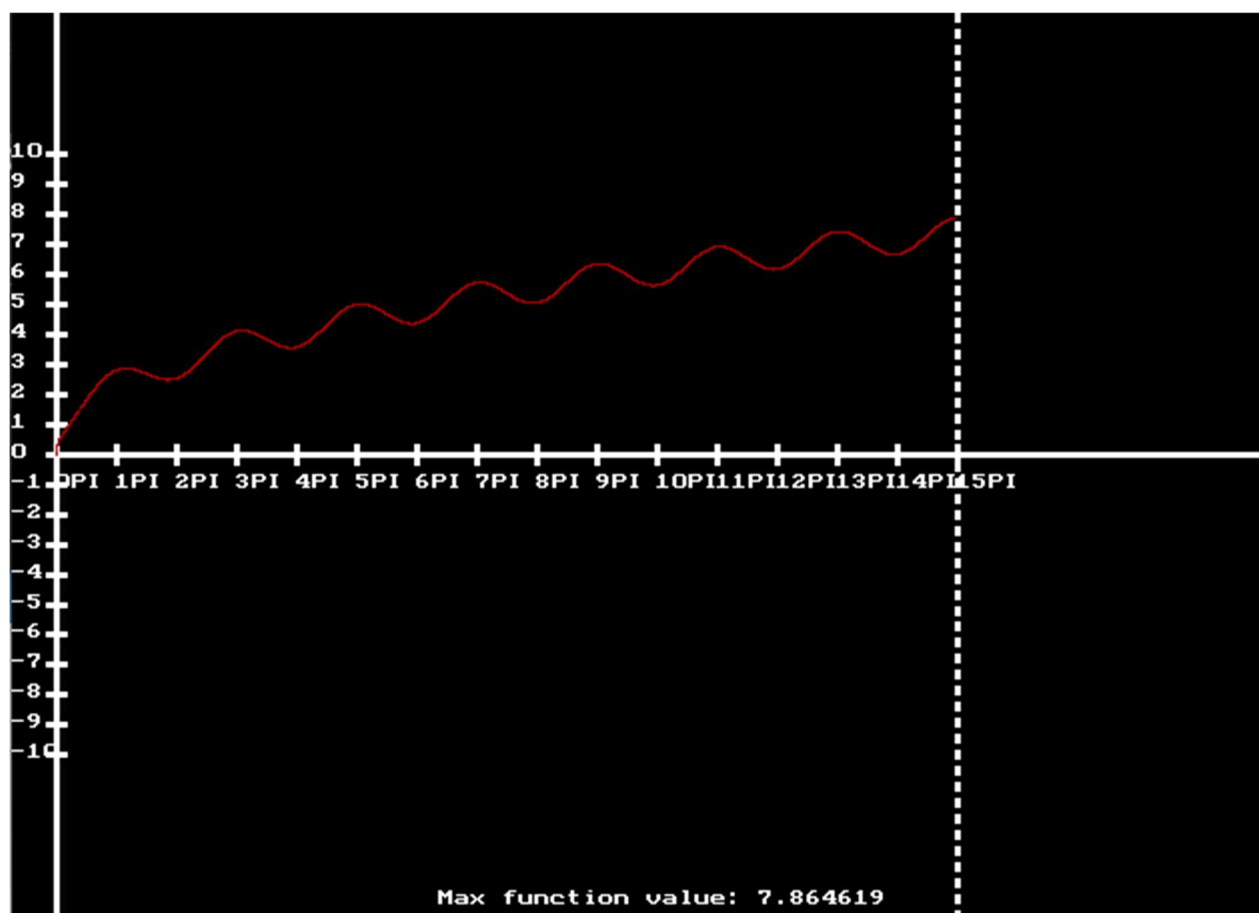


Рисунок 3 – Отображение графика функции

Вывод

В результате выполнения работы были изучены принципы работы видеосистемы в графическом режиме и получены навыки по работе с ней. Написание и отладка программы происходили в Turbo C++.

ПРИЛОЖЕНИЕ А

РАБОЧИЙ КОД

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <dos.h>
#include <graphics.h>
#include <math.h>
int main() {
    const double pi = 3.14159265358979323846;
    const double x_approach = 30 / pi;
    const double y_approach = 15;
    const int ten_num = 10;
    const float start_out = 0;
    const int end_out = 15;
    const int hatch_out = 30;
    const int number_out = 10;
    const int lines_out = 22;
    double y_result;
    double x_result;
    double y_out;
    double x_out;
    double max_result = -100;
    int i;
    int graph_driver, graph_mode, graph_error_code;
    int max_x;
    int max_y;
    char symbols_out[10];
    char maximum_out[50];
    clrscr();
    graph_driver = DETECT;
    detectgraph(&graph_driver, &graph_mode);
    initgraph(&graph_driver, &graph_mode, "C:\\TURBOC3\\BGI");
    // Проверка на ошибку инициализации графики
    graph_error_code = graphresult();
    if (graph_error_code != grOk) {
        printf("Ошибка: %s\n", grapherrormsg(graph_error_code));
        getch();
        return 255;
    }
    // Получаем максимальные значения координат
    max_x = getmaxx();
    max_y = getmaxy();
    setviewport(0, 0, max_x, max_y, 0);
    setlinestyle(0, 0, 3);
    line(lines_out, max_y, lines_out, 0); // вертикальная ось
    line(lines_out, max_y / 2, max_x, max_y / 2); // горизонтальная ось
    // Разметка оси X (от 0 до 15?)
    for (i = 0; i <= end_out; i++) {
        sprintf(symbols_out, "%dPI", i);
```

```

    outtextxy(lines_out + hatch_out * i, max_y / 2 + ten_num, symbols_out);
    line(lines_out + hatch_out * i, max_y / 2 + ten_num / 2, lines_out + hatch_out * i, max_y / 2 -
ten_num / 2);
}
// Разметка оси Y (от -10 до 10)
for (i = -number_out; i <= number_out; i++) {
    sprintf(symbols_out, "%d", i);
    outtextxy(0, max_y / 2 - hatch_out * i / 2 - ten_num / 2, symbols_out);
    line(lines_out + ten_num / 2, max_y / 2 - hatch_out * i / 2, lines_out - ten_num / 2, max_y / 2 -
hatch_out * i / 2);
}
// Рисуем асимптоты
setlinestyle(3, 0, 3);
line(lines_out + hatch_out * start_out, max_y, lines_out + hatch_out * start_out, 0); // начальная
setlinestyle(0, 0, 3);
setviewport(0, 0, max_x, max_y, 0);
// Рисуем график функции
for (x_result = start_out * pi; x_result <= end_out * pi; x_result += 0.001) {
    y_result = pow(sin(x_result / 2), 2) + sqrt(x_result); // функция
    x_out = x_result * x_approach;
    y_out = y_result * y_approach;
    if (y_result > max_result) {
        max_result = y_result;
    }
    putpixel(lines_out + x_out, max_y / 2 - y_out, 4); // рисуем пиксель графика
}
// Рисуем конечную асимптоту
setlinestyle(3, 0, 3);
line(lines_out + hatch_out * end_out, max_y, lines_out + hatch_out * end_out, 0);
// Вывод максимума функции
sprintf(maximum_out, "Max function value: %f", max_result);
outtextxy(max_x / 3, max_y - lines_out, maximum_out);
getch();
closegraph();
return 0;
}

```