**VIT**

U N I V E R S I T Y

(Estd. u/s 3 of UGC Act 1956)

Vellore - 632 014, Tamil Nadu, India

**School of Information Technology and Engineering**

B.Tech (IT)

**ANALOG CUM DIGITAL CLOCK IN C**

**MUDIT MAHESHWARI**

**(13BIT0016)**

**SWATI KUMARI**

**(12BIT0223)**

**SUBMITTED TO**

**PROF. LAXMIPRIYA G G**

**DATE: 02-05-2016**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Contents** | **Page No** |
| 1 | Abstract | 3 |
| 2 | Introduction | 4 |
| 2.1 | Overview | 4 |
| 2.2 | About Turbo C++ | 4 |
| 2.3 | Analog Clock | 4 |
| 3 | Requirement analysis and design | 5 |
| 4 | Implementation | 6 |
| 5 | Sample Screenshots | 10 |
| 6 | Testing | 11 |

**ABSTRACT**

The Project idea is to display the clock with computer graphics. This project used the local time, fetching from the computer and display it on screen that is the program displays a clock which is synchronized with the system’s clock.

This project implements the wall clock with the round circular board and three conic geometrics which forms the different (sec, min & hour) hands.The clock is designed using C language. The compiler used for designing the clock is Turbo C++.

**INTRODUCTION**

**OVERVIEW:**

The aim of this project is to implement an application package of computer graphics using Turbo C++ 3.0. Here we represent the concepts displaying Analog Clock using Turbo C++.

**ABOUT TURBO C++:**

-Turbo C++ was first introduced by Borland in 1990.

-It is a C++ compiler and integrated development environment (IDE) and computer language.

-Turbo C++ provides an environment called IDE (Integrated Development Environment).

- The editor is used to create the source file, compile it, link it and then execute it.

**ANALOG CLOCK:**

Analog Clock is the clock that that represents time by the position of hands on a dial.

It usually indicate time using angles. The most common clock face uses a fixed numbered dial or dials and moving hand or hands. It usually has a circular scale of 12 hours, which can also serve as a scale of 60 minutes and 60 seconds. The only other widely used clock face today is the 24 hour analog dial, because of the use of 24 hour dial in military organizations and timetables.

**REQUIREMENT ANALYSIS AND DESIGN**

**Software Requirements**

* Turbo C++ 3.0
* graphics.h library

**User Interface Requirements**

* Background color of the clock is Black.
* It contains 3 needles one for hour, one for minute and one for seconds which automatically updated every few seconds.
* A proper delay is set between needle moments and timings.
* Second Needle moves from 1 to 12 in clockwise direction.
* Digital time is also displayed.
* The second needle has the ticking sound.

The following are the functions used in the program.

**User defined Functions:**

* **clockLayout()**  
  We have used this function to print the clock layout i.e. clock dial and the markings on the clock. If we observe clearly, the clock has hours marking each separated by 30 degrees and each hour is divided into 5 markings each making an angle of 6 degrees. So, iterating the markings for every 30 degrees gives hours and iterating markings with 6 degrees give minutes markings on the clock. The clock would look like this after executing this function.
* **secHand()**  
  It is clear from the name that this function deals with something related to the seconds hand. This function is going to get the present second from the system clock and incline the line according to a particular angle. Eg: if the present seconds is 5 then the angle of the seconds hand with respect to the vertical must be 30 degrees, i.e. 5\*6=30.
* **minHand()**  
  This function fulfills the task of moving the minutes hand based on the system clock. The minutes hand must be inclined 6 degrees for every minute passing. Eg: if the elapsed minutes are 30 then the minutes hand angle must be making 180 degrees with the vertical.
* **hrHand()**This function is going to print an inclined hours line. The function shall be designed to get the present hour and also the no. of elapsed minutes from the system clock and incline the line according to a particular angle.
* **main()**  
  The first lines in main are graphic initialization. Coming to the while loop, the while loop iterates for every 100 milliseconds reprinting all the functions. The main function calls the above functions. This program is designed in such a way such that it gives a view of getting the static picture of clock every second and combining all the pictures to make a moving analog clock.

**Inbuilt Functions:**

* **Initgraph()**

Initgraph is used to initialize the graphics system. This function must be called to start the graphics system. It initializes the system by loading a graphics driver from disk then putting the system into graphics mode.

* **sound(int freq)**

Sound function produces the sound of a specified frequency

* **sleep()**

This function suspends the execution of the current thread until the time-out interval elapses.The time interval for which execution is to be suspended, in milliseconds.

* **getmaxx()**

getmaxx function is used to get the maximum (screen-relative) x value for the current graphics driver and mode.

* **getmaxy()**

getmaxy function is used to get the maximum y value of graphics screen.

* **setcolor()**

 setcolor function is used to set the foreground color.

* **pieslice()**

pieslice function draws and fills a pie slice centered at (x,y) with given radius. The slice travels from stangle to endangle. The slice is outlined in the current drawing color and then filled using the current fill pattern and fill color. The angles for pieslice are given in degrees. They are measured counterclockwise, with 0 degrees at 3 o'clock, 90 degrees at 12 o'clock, and so on.

* **line()**

line function is used to draw a line in the graphics screen. It uses the color and style set by the setcolor and setlinestyle functions.

* **gettime()**

gettime function is used to find current system time. We pass address of a structure variable of type(struct time). This function is defined in the header file named dos.h.

* **restorecrtmode()**

restorecrtmode function restores the original video mode detected by initgraph.

**IMPLEMENTATION**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<dos.h>

#include<math.h>

#define arg\_sec M\_PI/30

#define arg\_hour M\_PI/6

#define arg\_min M\_PI/360

void main()

{

int gd=DETECT,gm,sec=0,hour,min,x1=0,y1=0,x2=0,y2=0,x3=0,y3=0;

char \*k[13]={"1","2","3","4","5","6","7","8","9","10","11","12"};

struct time t;

initgraph(&gd,&gm,"C:/TURBOC3/bgi");

setcolor(YELLOW);

circle(300,200,200);

circle(300,200,180);

setfillstyle(1,RED);

floodfill(300,390,YELLOW);

settextstyle(DEFAULT\_FONT,0,2);

int a,b;

for(int i=1;i<=13;i++)

{

a=160\*cos(arg\_hour\*i-M\_PI\_2);

b=160\*sin(arg\_hour\*i-M\_PI\_2);

outtextxy(a+300,b+200,k[i-1]);

}

int dig\_sec;

char Time\_Dig[14];

while(!kbhit())

{

settextstyle(7,0,4);

setcolor(GREEN);

outtextxy(255,110,"SWATI");

setcolor(BLUE);

outtextxy(255,80,"MUDIT");

settextstyle(7,0,1);

setcolor(BLACK);

line(300,200,x1+300,y1+200);

line(300,200,x2+300,y2+200);

line(300,200,x3+300,y3+200);

gettime(&t);

if(sec!=t.ti\_sec)

{

sound(5000);

delay(1);

nosound();

}

hour=t.ti\_hour;

sec=t.ti\_sec;

min=t.ti\_min;

Time\_Dig[0]=hour/10+48;

Time\_Dig[1]=hour%10+48;

Time\_Dig[2]=':';

Time\_Dig[3]=min/10+48;

Time\_Dig[4]=min%10+48;

Time\_Dig[5]=':';

Time\_Dig[6]=sec/10+48;

Time\_Dig[7]=sec%10+48;

Time\_Dig[8]='\0';

outtextxy(270,250," ");

x1=150\*cos(arg\_sec\*sec-M\_PI\_2)\*0.98;

y1=150\*sin(arg\_sec\*sec-M\_PI\_2)\*0.98;

x2=150\*cos(arg\_sec\*min-M\_PI\_2)\*0.9;

y2=150\*sin(arg\_sec\*min-M\_PI\_2)\*0.9;

if(hour>12) hour-=12;

x3=150\*cos(arg\_hour\*hour-M\_PI\_2+arg\_min\*min)\*0.6;

y3=150\*sin(arg\_hour\*hour-M\_PI\_2+arg\_min\*min)\*0.6;

setcolor(YELLOW);

line(300,200,x1+300,y1+200);

setcolor(CYAN);

line(300,200,x2+300,y2+200);

setcolor(WHITE);

line(300,200,x3+300,y3+200);

setcolor(YELLOW);

outtextxy(270,250,Time\_Dig);

delay(50);

}

getch();

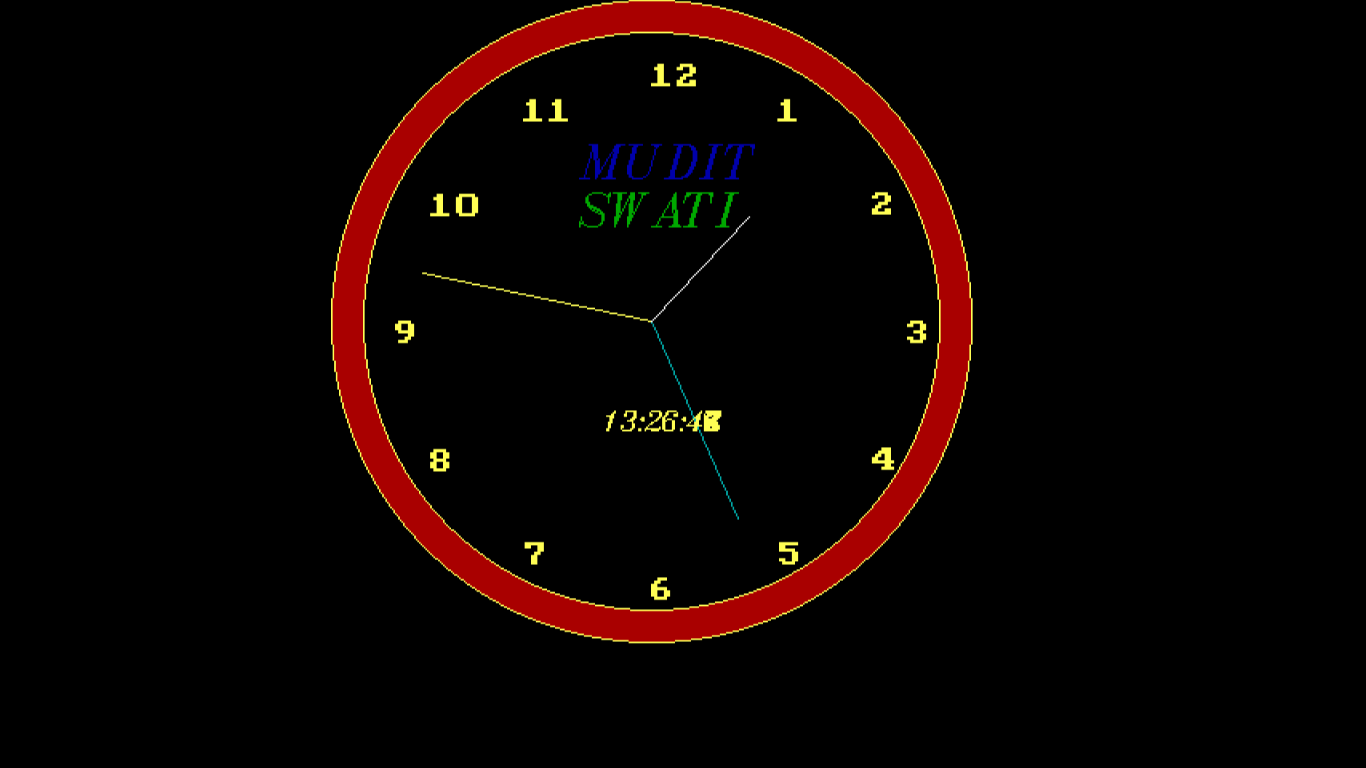
closegraph();

restorecrtmode();

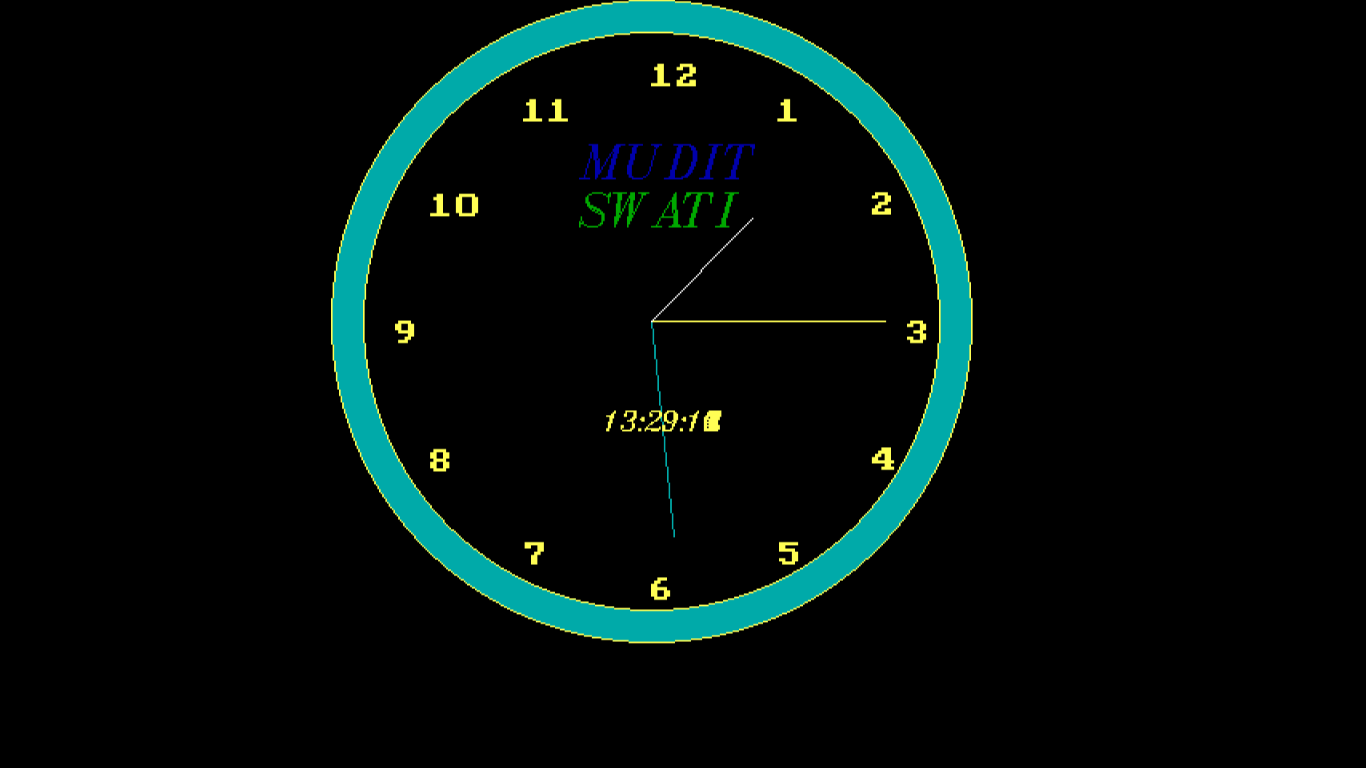
}

**SAMPLE SCREENSHOTS**

Original Screenshot:

****

After changing the color:

****

**TESTING**

Testing process of this code is very simple. When we execute the program it displays the current system time properly during both day and night.

It also displays proper digital time and has the ticking sound.