MATLAB PROJECT :

PROJECT DETAILS:

TITLE: ANALOG CLOCK USING MATLAB

INTRODUCTION:

NASA might get a fancy digital-count display and retire the analog clock from the Apollo era, but the passion for the analog clock does not end. An analog clock has moving hands, where the smaller one is the hour's hand and travels 30° in one hour, and the longer one is the minute's hand and travels 360° in an hour. The second's hand rotates with a step of 6°. Here we look to design an analog clock using MATLAB.

If you have an analog watch, it tells the time with hands that sweep around a dial: the position of the hands is a measurement of the time. How much the hands move is directly related to what time it is. So if the hour hand sweeps across two segments of the dial, it's showing that twice as much time has elapsed compared to if it had moved only one segment. The point is that the hand's movements over the dial are a way of representing passing time. It's not the same thing as time itself: it's a representation or an analogy of time.

Just because digital technology has advantages, that doesn't mean it's always better than analog. An analog watch might be far more accurate than a digital one if it uses a high-precision movement (gears and springs) to measure time passing. If it has a sweeping second hand, it represents the time more precisely than a digital watch whose display shows only hours and minutes. Surprisingly, analog watches can also keep time better than quartz ones.

Generally, the most expensive watches in the world are analog ones (of course, that's partly because people prefer the way they look), though the world's most accurate atomic clocks show time with digital displays.



CODE EXPLANATION : (PART BY PART)

Here we start are code,

% analog clock using matlab

clc;                           % clears all the text from the Command Window, resulting in a clear screen

clear;                         % To clear all variables from the current workspace

clf;                           % deletes all children of the current figure that have visible handles.

format shortg;

% for an analog watch we will create a circle of radius 10

We start by clearing the command window using the clc and clear keywords available in matlab and we also format the text using the format shortg function.

x=0;                           % x and y are the co-ordinates

y=0;

r=10;                          % radius of the circle

hold on;                       % it is used to overlap multiple outputs in a single frame

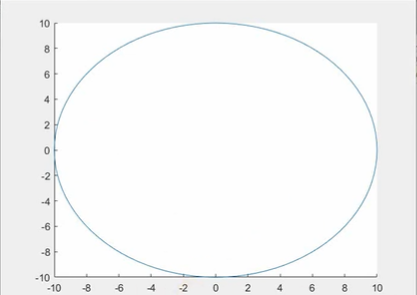
th = 0:pi/50:2\*pi;             % this is the time period 0 to 2pi (0-360 degrees) in the span of 50(180/50)

xunit=r\*cos(th)+x;             % x and y units in the x and y direction

yunit=r\*sin(th)+y;             % maximum value of cos and sin is 1 and minimum is -1

h=plot(xunit,yunit);           % now we plot the x and y and hence obtain a circle of radius 10

This piece of code helps us to create a circle with units.



r=8;                           % radius of inner circle

s=[1];                         % string will start from 1 to 12

for th=pi/6: pi/6: 2\*pi        % time period pi/6 which means 180/6 means 30 so , in the span of 30 till 360

    ytemp=r\*cos(th);           % xtemp and ytemp are just co-ordinates

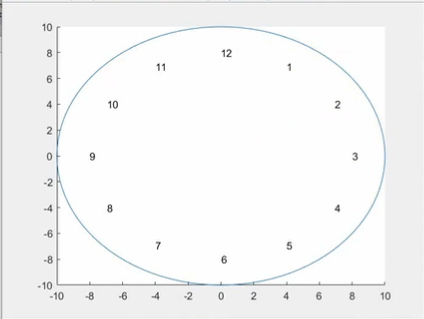
    xtemp=r\*sin(th);           % and cos and sin are used to locate the position

    text(xtemp,ytemp,num2str(s));   % we fetch the co-ordinates and the number converted to string

    s=s+1;                     % initially it was one then we will increment it till end (360 ) and then it will end

end                            % end the for loop

This piece of code will help us form the inner circle of the clock where the hours will be mentioned from 1 to 12.



Now we will start with the main logic of the clock,

% while 1 means always true means the clock is always running (infinite loop)

while(1)

c=clock;                       % it will take the system time

c=c(1,4:6);                    % all the data will be stored in c

min=c(1,2);                    % minutes

sec=c(1,3);                    % seconds

% it is used to denote the am and pm thing

if (c(1,1)>12)

    hr=c(1,1)-12;

else

    hr=c(1,1);

end

min1=ceil(min/12);              % for minute we have to do round off and ceil is just round off to 12

theta=(hr\*pi)/6 + (min1\*pi)/30;

f=figure(1);

hold on;

% xtemp and ytemp are the co-ordinates

% we will be showing how the lines are there

The shorter hand is for the hours,

The longer hand is for the seconds,

The medium hand is for the minutes.

ytemp=3\*cos(theta);

y=[0 ytemp];

xtemp=3\*sin(theta);

x=[0 xtemp];

p=plot(X,Y);

hold on;           % used to overlap the output

theta1=(min\*pi)/30;  % means that it is the centre and all the lines should start from the centre and be aligned

In one hour the hours hand should rotate by 30 degrees,

And the minutes hand should rotate by 360 degrees.

ytemp1=5\*cos(theta1);

Y1=[0 ytemp1];

xtemp1=5\*sin(theta1);

X1=[0 xtemp1];

p1=plot(X1,Y1);

theta2=(sec\*pi)/30;

ytemp2=7\*cos(theta2);

Y2=[0 ytemp2];

xtemp2=7\*sin(theta2);

X2=[0 xtemp2];

p2=plot(X2,Y2);

% pause 1 will cause a delay of 1 second in the rotation of second hand

pause(1);           % we are using this function for 1 second

% and then according to second we will adjust the hours and minutes

% after 360 rotations of the minute hand 30 degree rotation of the hour hand

% we have to delete whatever plot was there each and every time in a loop so

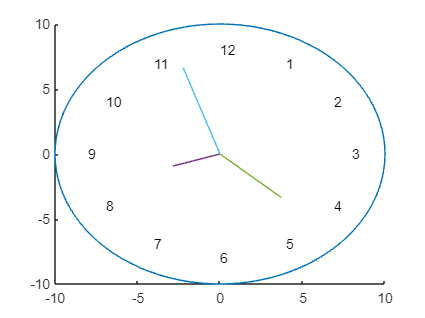
% that the new location will be added

delete(p2);

delete(p1);

delete(p);

end



This image represents how the clock will be figured after the successful execution of our code.

FINAL SOURCE CODE:

% analog clock using matlab

clc;                           % clears all the text from the Command Window, resulting in a clear screen

clear;                         % To clear all variables from the current workspace

clf;                           % deletes all children of the current figure that have visible handles.

format shortg;

% for an analog watch we will create a circle of radius 10

% this peice of code helps us to create a circle with units

x=0;                           % x and y are the co-ordinates

y=0;

r=10;                          % radius of the circle

hold on;                       % it is used to overlap multiple outputs in a single frame

th = 0:pi/50:2\*pi;             % this is the time period 0 to 2pi (0-360 degrees) in the span of 50(180/50)

xunit=r\*cos(th)+x;             % x and y units in the x and y direction

yunit=r\*sin(th)+y;             % maximum value of cos and sin is 1 and minimum is -1

h=plot(xunit,yunit);           % now we plot the x and y and hence obtain a circle of radius 10

% this will be the radius for the inner circle where the hours will be mentioned (1-12)

r=8;                           % radius of inner circle

s=[1];                         % string will start from 1 to 12

for th=pi/6: pi/6: 2\*pi        % time period pi/6 which means 180/6 means 30 so , in the span of 30 till 360

    ytemp=r\*cos(th);           % xtemp and ytemp are just co-ordinates

    xtemp=r\*sin(th);           % and cos and sin are used to locate the position

    text(xtemp,ytemp,num2str(s));   % we fetch the co-ordinates and the number converted to string

    s=s+1;                     % initially it was one then we will increment it till end (360 ) and then it will end

end                            % end the for loop

% now we will start with the logic of the clock

% while 1 means always true means the clock is always running (infinite loop)

while(1)

c=clock;                       % it will take the system time

c=c(1,4:6);                    % all the data will be stored in c

min=c(1,2);                    % minutes

sec=c(1,3);                    % seconds

% it is used to denote the am and pm thing

if (c(1,1)>12)

    hr=c(1,1)-12;

else

    hr=c(1,1);

end

min1=ceil(min/12);              % for minute we have to do round off and ceil is just round off to 12

theta=(hr\*pi)/6 + (min1\*pi)/30;

f=figure(1);

hold on;

% xtemp and ytemp are the co-ordinates

% we will be showing how the lines are there

% the shorter hand for hours

% the longer hand for seconds

% the medium hand is for the minute

ytemp=3\*cos(theta);

y=[0 ytemp];

xtemp=3\*sin(theta);

x=[0 xtemp];

p=plot(X,Y);

hold on;           % used to overlap the output

theta1=(min\*pi)/30;  % means that it is the centre and all the lines should start from the centre and be aligned

% in one hour the hours hand should rotate by 30 degrees

% in one hour the minutes hand should rotate by 360 degrees

ytemp1=5\*cos(theta1);

Y1=[0 ytemp1];

xtemp1=5\*sin(theta1);

X1=[0 xtemp1];

p1=plot(X1,Y1);

theta2=(sec\*pi)/30;

ytemp2=7\*cos(theta2);

Y2=[0 ytemp2];

xtemp2=7\*sin(theta2);

X2=[0 xtemp2];

p2=plot(X2,Y2);

% pause 1 will cause a delay of 1 second in the rotation of second hand

pause(1);           % we are using this function for 1 second

% and then according to second we will adjust the hours and minutes

% after 360 rotations of the minute hand 30 degree rotation of the hour hand

% we have to delete whatever plot was there each and every time in a loop so

% that the new location will be added

delete(p2);

delete(p1);

delete(p);

end