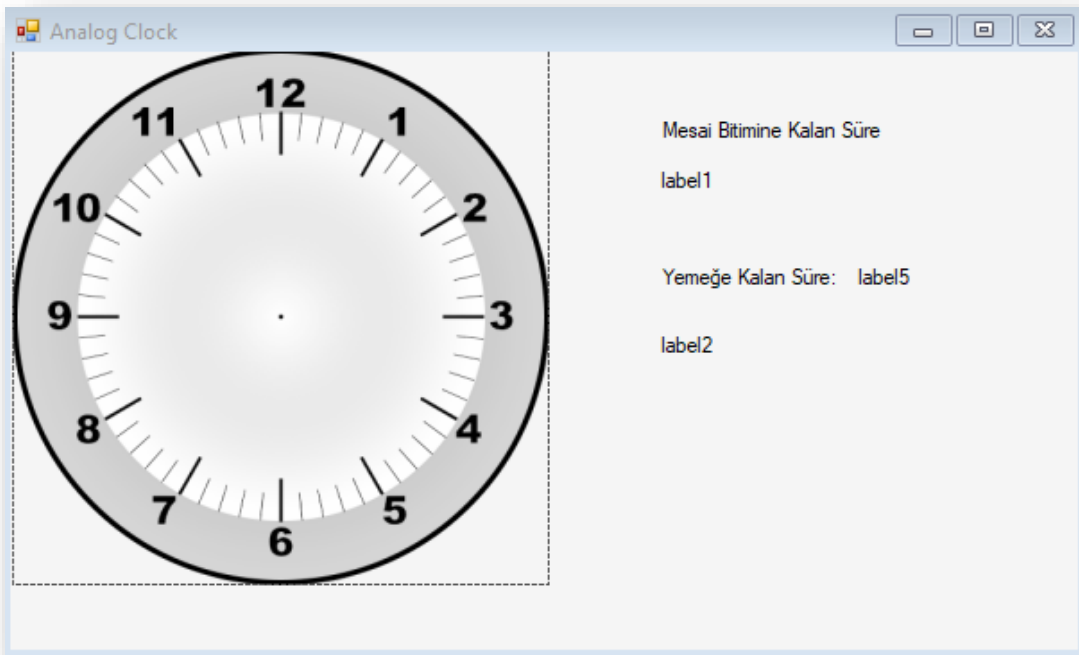


# Analog Clock Project

In this project, my consultant asked me to design an Analog Clock by taking the system time and when the application is run, the Analog Clock would advance according to the current system time. In addition to these, in addition to the current time information, it was requested that it display the time left until the end of the working day and the time left until the mealtime, and if the working hour is over or the meal hour has passed, this should be written on the screen.

Let's start with the design of **Form 1**, which appears on the screen when the application is run. In Form 1, two **PictureBoxes** are used to visualize the Analog Clock, seconds bar, minute bar and hour bar. In addition, five **Textboxes** are used to show the user the end of work hours and the time left for dinner. The appearance of **Form 1** is given in the image below.



**Figure 1** The picture of Analog Clock Project Form 1

Behind the Analog Clock PictureBox seen in **Figure 1**, there is another **PictureBox** to which hour, second and minute bars are added. In addition, Labels have been placed on the right side of **the PictureBoxes** for remaining time information. Locations of **PictureBoxes** are set to 0;0 and sizes are set to 300;300.

In the coding section of **the Form 1** interface, first the positions and sizes of the data in **the digital clock** are defined to display them in Analog Clock format. These dimensions are defined as 300;300 in accordance with **the PictureBox** where the Analog Clock is located. In addition, the dimensions of the hour, minute and second bars have been made. Additionally, **the Timer** variable is defined to calculate the remaining working hours and meal hours requested from us. The codes shown in the image below are given.

```

public partial class Form1 : Form
{
    Timer t; = new Timer();
    // WIDTH VE HEIGHT PICTUREBOX'IN SIZE'I SECOND , MINHAND VE HRHAND'IN UZUNLUĞU
    //
    int WIDTH = 300, HEIGHT = 300, SECOND = 140, MINHAND = 110, HRHAND = 80;
    // X VE Y KORDİNATLARININ TANITILMASI
    int cx, cy;
    Bitmap bmp; //bmp değişkeni pictureBox1'in kullanmka için tanımladık
    Graphics g;

    1 başvuru
    public Form1()
    {
        1 başvuru
        private void Form1_Load(object sender, EventArgs e)
        {
            bmp = new Bitmap(WIDTH , HEIGHT ); // pictureBox1 size'nı bmp değişkenine kopyaladık

            cx = WIDTH / 2;
            cy = HEIGHT / 2; // pictureBox1 in orta noktasını bu değişkenini konumlandırdık

            t.Interval = 1000; // timer'ı ilerlerleme aralığı 1 sn ayarlamak için yazdık.
            t.Tick += new EventHandler(this.t_Tick); //timerı başlatmak için
            t.Start();
        }
    }
}

```

**Figure 2** The picture of Analog Clock Project Introduction to Variables

In the C# form application, the *"DateTime.Now"* command is used to retrieve the current time information from the system. Using this command, hour, minute and second variables in the system were written to integer variables. In addition, since the Analog Clock image in this section consists of a two-dimensional photograph, the position, colour and size of the hour, minute and second hands were adjusted and drawn using the *"DrawLine"* command. To place the hour, minute and second hands in the middle of the clock image, we defined these bars as an element of the clock image. The specified codes are given in the image below.

```

private void t_Tick(object sender, EventArgs e)
{
    g = Graphics.FromImage(bmp); //bmp bizim pictureBox1 alanımız bunun üstünde çeşitli çizimler yapabilmek için Graphics komutuyla g değişkenine atarız

    //Sistemden saat bilgilerini çektik.
    int ss = DateTime.Now.Second;
    int mm = DateTime.Now.Minute;
    int hh = DateTime.Now.Hour;

    int[] handCoord = new int[2]; // iki elemanlı bir dizi(array) oluşturduk

    g.Clear(Color.Transparent); // akrep yelkovan ve saniyenin bir sonraki konuma gittiğinde önceki oluşan konumu temizler

    handCoord = msCoord(ss, SECOND); //sistemden aldığımız saat bilgilerini mscoord fonk kullanarak handcoord dizisine yazarak gönderdik
    g.DrawLine(new Pen(Color.Red, 1f), new Point(cx, cy), new Point(handCoord[0], handCoord[1])); //Drawline komutunda ise saniye çubuğunu çizdirdik

    handCoord = msCoord(mm, MINHAND); //sistemden aldığımız saat bilgilerini mscoord fonk kullanarak handcoord dizisine yazarak gönderdik
    g.DrawLine(new Pen(Color.Black, 2f), new Point(cx, cy), new Point(handCoord[0], handCoord[1])); //Drawline komutunda ise dakika çubuğunu çizdirdik

    handCoord = hrCoord(hh % 12, mm, HRHAND); //sistemden aldığımız saat bilgilerini mscoord fonk kullanarak handcoord dizisine yazarak gönderdik
    g.DrawLine(new Pen(Color.Blue, 3f), new Point(cx, cy), new Point(handCoord[0], handCoord[1])); //Drawline komutunda ise saat çubuğunu çizdirdik

    pictureBox1.Image = bmp; //oluşan bmp'yi pictureBox1'e aktardık
    pictureBox1.Parent = pictureBox2; // pictureBox1 (pictureBox1'in pictureBox2 üzerinde transparent'in düzgün çalışması için yazdık
}

```

**Figure Hata!** Belgede belirtilen stilde metne rastlanmadı. *The picture of Analog Clock Project Introduction to Datetime Variables*

The current system time included in the information given to me is written in the Header section of **Form 1**. In addition, the necessary detailed procedures for the end time and remaining hours information have been set. The codes mentioned above are given in the image below.

```

this.Text = "Analog Clock - " + hh + ":" + mm + ":" + ss; // Formun ismini sistemden alınan saat bilgilerini dijital bir şekilde yazdırdık

int kalansaay = Math.Abs(16 - hh);
int kalandak = Math.Abs(34 - mm);
int kalansan = Math.Abs(59 - ss);
label1.Text = Convert.ToString(kalansaay + ":" + "Saat" + kalandak + ":" + "Dakika" + kalansan + ":" + "Saniye");

int kalansaatt = (12 - hh);
int kalandakk = (15 - mm);
int kalansann = (59 - ss);
int kalansaatt1 = Math.Abs(12 - hh);
int kalandakk1 = Math.Abs(15 - mm);
int kalansann1 = Math.Abs(59 - ss);
int kalansure = (kalansaatt * 60) + kalandakk1;
if ((kalansaatt) < 0)
{
    label3.Text = "Mola Saati Geçti";
    label2.Hide();
    label5.Text = Convert.ToString(kalansure + "Dakika geçti");
}
else if (kalansaatt < 0 && kalandakk < 0)
{
    label3.Text = "Mola Saati Geçti";
    label2.Hide();
    label5.Text = Convert.ToString(kalansure + "Dakika geçti");
}
else if (kalansaatt == 0 && kalandakk == 0)
{
    label2.Text = "İyi dinlenmeler.";
    label5.Hide();
    label3.Hide();
}
else if (kalansaatt < 0 && kalandakk > 0)
{
    label2.Text = Convert.ToString(kalansaatt1 + " " + "Saat" + kalandakk1 + " " + "Dakika" + kalansann1 + " " + "Saniye");
    label5.Hide();
}
else
{
    label2.Text = Convert.ToString(kalansaatt1 + " " + "Saat" + kalandakk1 + " " + "Dakika" + kalansann1 + " " + "Saniye");
    label5.Hide();
}

```

**Figure 4** The picture of Analog Clock Project of Remaining Time Calculations

Then, functions called "*msCoord*" and "*hrCoord*" were written to adjust the angles of the hour, minute and second hands defined above for the Analog Clock. In these functions, calculations are made accordingly since the full rotation of the Analog Clock is 360 degrees. Since the minute hand and second hand move by 60 units, "*msCoord*" function was used for the minute hand and second hand. The codes written for "*msCoord*" function are given below.

```

private int[] msCoord(int val, int hlen) // Value = sistemin saati hlen= çubuğun uzunluğu
{
    int[] coord = new int[2]; // Yeni iki elemanlı dizi tanımlı
    val *= 6; // value'muz o anki değeri derece cinsine çevirme
    if (val >= 0 && val <= 180) // Yeni derecemiz 0-180 arası olursa
    {
        coord[0] = cx + (int)(hlen * Math.Sin(Math.PI * val / 180));
        coord[1] = cy - (int)(hlen * Math.Cos(Math.PI * val / 180));
    }
    else // Yeni derecemiz 181-360 arası olursa
    {
        coord[0] = cx - (int)(hlen * Math.Sin(Math.PI * val / 180));
        coord[1] = cy - (int)(hlen * Math.Cos(Math.PI * val / 180));
    }
    return coord;
}

```

**Figure 5** The picture of Analog Clock Project msCoord Function.

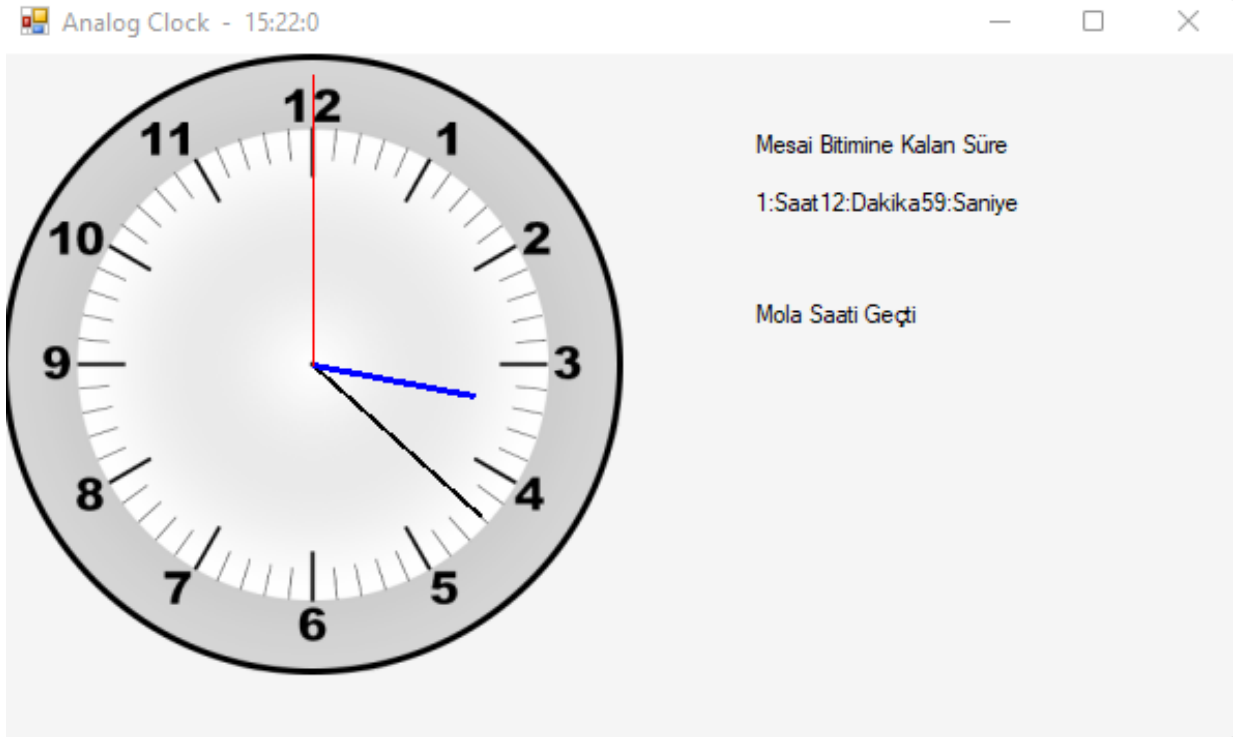
The minute and second hands calculated using the *msCoord* function given in *Figure 5* have been adjusted. Then, the “*hrCoord*” function was written to adjust the hour hand, which advances 12 units per hour and 0.5 units per minute. The codes written in this function are shown in the image below.

```
private int[] hrCoord(int hval, int mval, int hlen)
{
    int[] coord = new int[2];
    // each hour makes 30 degrees , each min makes 0.5 degrees
    int val = (int)((hval * 30) + (mval * 0.5));

    if (val >= 0 && val <= 180)
    {
        coord[0] = cx + (int)(hlen * Math.Sin(Math.PI * val / 180));
        coord[1] = cy - (int)(hlen * Math.Cos(Math.PI * val / 180));
    }
    else
    {
        coord[0] = cx - (int)(hlen * -Math.Sin(Math.PI * val / 180));
        coord[1] = cy - (int)(hlen * Math.Cos(Math.PI * val / 180));
    }
    return coord;
}
```

*Figure 6 The Picture of Analog Clock Project hrCoord Function*

The form interface created using the codes given above is shown in the image below.



*Figure 7 The Picture of Analog Clock Project.exe*