ДОДАТОК А
Код програми

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Reflection.Emit;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Net.Mime.MediaTypeNames;

namespace Telemetry
{
    public partial class DashCompass : UserControl
    {
        private float radius = 80;
        private float arrowAngle = 45;

        public DashCompass()
        {
            InitializeComponent();
        }

        private void myPanel1_Paint(object sender, PaintEventArgs e)
        {
            float arcWidth = 10;
            float arrowLength = radius - 5;
            float arrowWidth = 10;
```

```
        e.Graphics.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.AntiAlias;

        //Рисуємо шкалу
        Graphics g = e.Graphics;
        Pen pen_shkala = new Pen(Color.LightGray, arcWidth);
        Pen pen_arrow = new Pen(Color.Red, 1);
        Pen pen_line = new Pen(Color.Black, 2);
        Brush brush = new SolidBrush(Color.Red);

        // Центр форми
        Point centerPoint = new Point(this.ClientSize.Width / 2, this.ClientSize.Height / 2);

        for (float angle = 0; angle <= 360; angle += 6)
        {
           // Обчислення прямокутника для малювання дуги
           float rectWidth = 2 * radius;
           float rectHeight = 2 * radius;
           float rectX = centerPoint.X - radius;
           float rectY = centerPoint.Y - radius;
           g.DrawArc(pen_shkala, rectX, rectY, rectWidth, rectHeight, angle, 2);
        }

        //draw line
        g.DrawLine(pen_line, centerPoint.X, centerPoint.Y, centerPoint.X, centerPoint.Y - radius +
35);
        g.DrawLine(pen_line, centerPoint.X, centerPoint.Y, centerPoint.X, centerPoint.Y + radius -
35);
        g.DrawLine(pen_line, centerPoint.X, centerPoint.Y, centerPoint.X - radius + 35,
centerPoint.Y);
        g.DrawLine(pen_line, centerPoint.X, centerPoint.Y, centerPoint.X + radius - 35,
centerPoint.Y);

        //draw text
```

```
// Колір тексту
Brush brush_Text1 = new SolidBrush(Color.Red);
Brush brush_Text2 = new SolidBrush(Color.Black);

// Вивід тексту
// Встановлення шрифту
System.Drawing.Font font;
try
{
   font = new System.Drawing.Font("Square721 BT", 14);
}
catch (ArgumentException)
{
   // У випадку, якщо шрифт не знайдено, використовуємо стандартний шрифт
   font = new System.Drawing.Font(FontFamily.GenericSansSerif, 12);
}
g.DrawString("N", font, brush_Text1, centerPoint.X - 10, centerPoint.Y - radius + 10);
g.DrawString("S", font, brush_Text2, centerPoint.X - 10, centerPoint.Y + radius - 30);
g.DrawString("W", font, brush_Text2, centerPoint.X - radius + 8 , centerPoint.Y - 10);
g.DrawString("E", font, brush_Text2, centerPoint.X + radius - 30, centerPoint.Y - 10);

//Рисуємо вказівник ==========
// Обчислення вершин трикутника (стрілки компаса)
float radians = arrowAngle * (float)Math.PI / 180;

PointF tipPoint = new PointF(
   centerPoint.X + arrowLength * (float)Math.Cos(radians),
   centerPoint.Y - arrowLength * (float)Math.Sin(radians)
);

PointF leftBasePoint = new PointF(
```

```
            centerPoint.X + arrowWidth / 2 * (float)Math.Sin(radians),
            centerPoint.Y + arrowWidth / 2 * (float)Math.Cos(radians)
        );

        PointF rightBasePoint = new PointF(
            centerPoint.X - arrowWidth / 2 * (float)Math.Sin(radians),
            centerPoint.Y - arrowWidth / 2 * (float)Math.Cos(radians)
        );

        PointF[] arrowPoints = { leftBasePoint, rightBasePoint, tipPoint };

        // Малюємо трикутник
        g.FillPolygon(brush, arrowPoints);

        //----
        // draw pint of center
        int dotDiameter = 20;
        int dotX = centerPoint.X - dotDiameter / 2;
        int dotY = centerPoint.Y - dotDiameter / 2;
        Brush brushPoint = new SolidBrush(Color.Black);
        g.FillEllipse(brushPoint, dotX, dotY, dotDiameter, dotDiameter);
    }

    public void update(int value)
    {
        this.arrowAngle = value;
        myPanel1.Refresh();
    }
  }
}
```

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Printing;
using System.Linq;
using System.Reflection.Metadata;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Telemetry
{
    public partial class DashHalfCycle : UserControl
    {
        //private float length = 20;
        //private float distanceFromCenter = 70;
        private float startAngle = 0;
        private float endAngle = 180;
        private float angleStep = 5;
        private float arcWidth = 25;
        private float radius = 80;
        private int value = 10;
        private int minValue = 0;
        private int maxValue = 100;
        private string title = "Speed, m/s";

        public DashHalfCycle()
        {
            InitializeComponent();
```

```
}

private void myPanel1_Paint(object sender, PaintEventArgs e)
{
   e.Graphics.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.AntiAlias;

   //Calc scale
   int lineCount = (int)((endAngle - startAngle) / angleStep);
   float scale = ((maxValue - minValue) / lineCount);
   int porog = (int) (value / scale);

   Graphics g = e.Graphics;
   Pen pen_back = new Pen(Color.LightGray, arcWidth);
   Pen pen_value = new Pen(Color.Green, arcWidth);
   Pen pen_value_min = new Pen(Color.Green, arcWidth);
   Pen pen_value_mid = new Pen(Color.Orange, arcWidth);
   Pen pen_value_max = new Pen(Color.Red, arcWidth);

   // Кольор форми
   Point centerPoint = new Point(this.ClientSize.Width / 2, this.ClientSize.Height / 2);
   centerPoint.Y = centerPoint.Y + (int)(centerPoint.Y / 2.5);

   int porog_1 = 15;
   int porog_2 = 30;
   int curLine= 1;
   for (float angle = startAngle + 180; angle <= endAngle + 180; angle += angleStep)
   {
      // Обчислення прямокутника для малювання дуги
      float rectWidth = 2 * radius;
      float rectHeight = 2 * radius;
      float rectX = centerPoint.X - radius;
```

```
            float rectY = centerPoint.Y - radius;


            // Малюємо дугу
            if (curLine > porog)
            {
               g.DrawArc(pen_back, rectX, rectY, rectWidth, rectHeight, angle, angleStep - 2);
            }
            else
            {
               if (curLine < porog_1) g.DrawArc(pen_value_min, rectX, rectY, rectWidth,
rectHeight, angle, angleStep - 2);
               else if (curLine < porog_2) g.DrawArc(pen_value_mid, rectX, rectY, rectWidth,
rectHeight, angle, angleStep - 2);
               else g.DrawArc(pen_value_max, rectX, rectY, rectWidth, rectHeight, angle, angleStep
- 2);
            }
            curLine++;
         }
      }


      public void update (int value, int minValue, int maxValue)
      {
         this.value = value;
         this.minValue = minValue;
         this.maxValue = maxValue;


         startAngle = 0;
         endAngle = 180;


         label1.Text = String.Format("{0:D3}", value);


         myPanel1.Refresh();
```

```
        }

        private void DashHalfCycle_Load(object sender, EventArgs e)
        {
            label1.Text = String.Format("{0:D3}", value);
            label2.Text = title;
        }
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Printing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Telemetry
{
    public partial class DashHorizontal : UserControl
    {
        private int value = 80;
        private int minValue = 0;
        private int maxValue = 100;
        private string title = "Temperature, C";

        public DashHorizontal()
```

```
{
    InitializeComponent();
}

private void myPanel1_Paint(object sender, PaintEventArgs e)
{
    e.Graphics.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.AntiAlias;

    float lineCount = 30;
    int arcWidth = 4;
    float scale = ((maxValue - minValue) / lineCount);
    int porog = (int)(value / scale);
    int porog_1 = 10;
    int porog_2 = 21;
    int margin = 20;
    int margin_y = 40;

    Graphics g = e.Graphics;
    Pen pen_back = new Pen(Color.LightGray, arcWidth);
    Pen pen_value_min = new Pen(Color.Green, arcWidth);
    Pen pen_value_mid = new Pen(Color.Orange, arcWidth);
    Pen pen_value_max = new Pen(Color.Red, arcWidth);

    int lineSpacing = (this.ClientSize.Width - 2 * margin) / 30;
    int lineHeight = this.ClientSize.Height - 7 * margin;
    float lineWidth = this.ClientSize.Width / 50f;

    // Встановлення пера для малювання ліній
    Pen pen = new Pen(Color.Black, lineWidth);

    for (int i = 0; i < 30; i++)
```

```
        {
            int x = margin + i * lineSpacing;

            if (i >= porog)
            {
                g.DrawLine(pen_back, x, margin_y, x, margin_y + lineHeight);
            }
            else
            {
                if (i < porog_1) g.DrawLine(pen_value_min, x, margin_y, x, margin_y + lineHeight);
                else if (i < porog_2) g.DrawLine(pen_value_mid, x, margin_y, x, margin_y +
lineHeight);
                else g.DrawLine(pen_value_max, x, margin_y, x, margin_y + lineHeight);
            }
        }
    }

    private void DashHorizontal_Load(object sender, EventArgs e)
    {
        label1.Text = String.Format("{0:D3}", value);
        label2.Text = title;
        label3.Text = minValue.ToString();
        label4.Text = maxValue.ToString();
    }

    public void update(int value, int minValue, int maxValue, string title)
    {
        this.value = value;
        this.minValue = minValue;
        this.maxValue = maxValue;
        this.title = title;
```

```
            label1.Text = String.Format("{0:D3}", value);
            label2.Text = title;
            label3.Text = minValue.ToString();
            label4.Text = maxValue.ToString();

            myPanel1.Refresh();
        }
    }
}
```

ДОДАТОК Б

Демонстраційний матеріал

№	Формат	Позначення	Найменування	Кіл.	Прим.
			Документація текстова		
1	*А4*	*ГЮІК.505400.005ПЗ*	*Пояснювальна записка*	*71с*	
			Інші документи		
2	*А4*		*Лістинг програмного коду*	*11с*	
3	*А4*		*Демонстраційний матеріал*	*12с*	*у форматі *.ppt*

Змі	Арк	№ докум.	Підпис	Дата	ГЮІК.505400.005ВД		
Розроб		Калюжний М.В			*Розроблення програмного модуля для відображення телеметрії з датчиків позиціонування мобільного робота*	Літера	Аркуш	Аркушів
Перевірив		Новоселов С.П.						1
Н. Контр.		Бронніков А.І.				ХНУРЕ кафедра КІТАМ		
Затвер.		Невлюдов І. Ш.						