

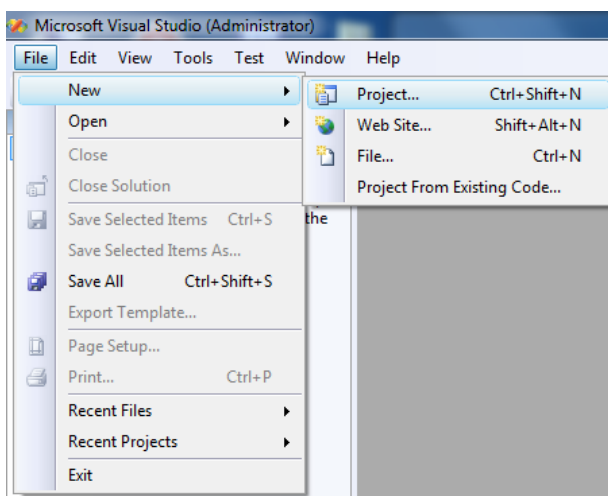
HƯỚNG DẪN VẼ ĐỒ THỊ VỚI VISUAL STUDIO C# 2008

Hôm nay, ngày 05/08/2011 đúng là 1 ngày buồn như chó chết (con chó nhà bên cạnh bị nấu dạ cây ^_^), lại ngồi viết TUT cho anh em engineer mới vào nghề.

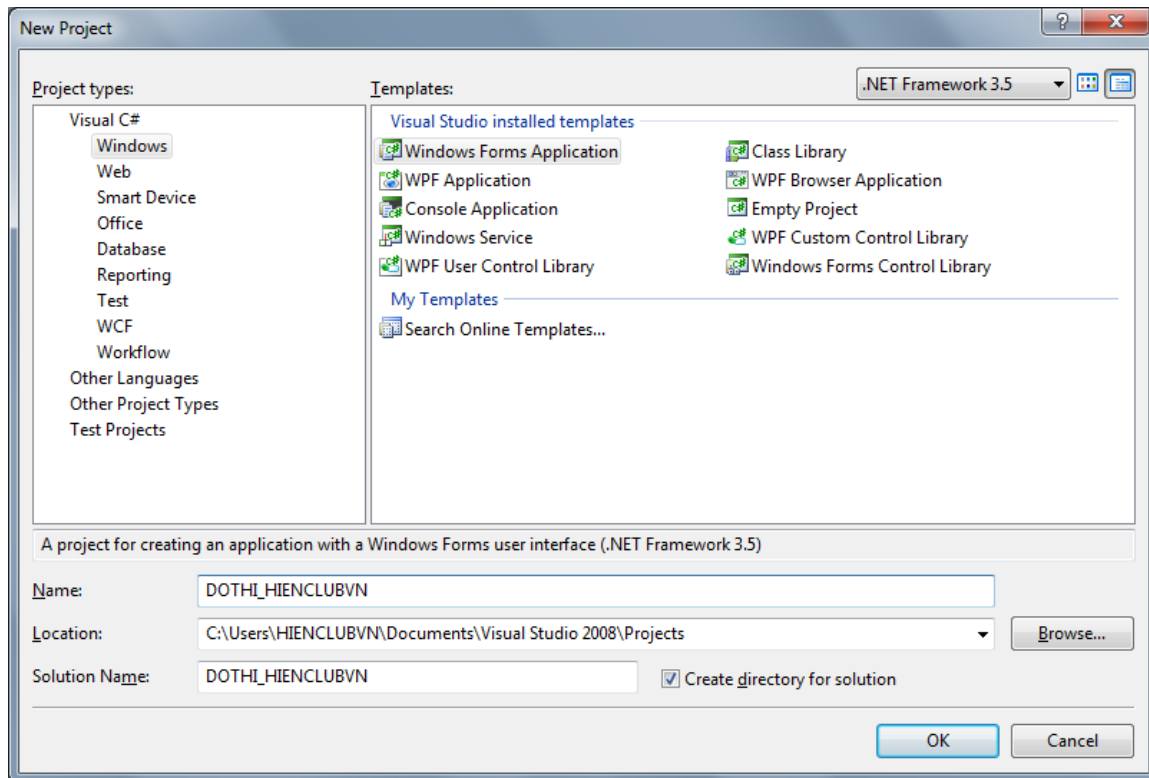
1. Giới thiệu

- Trong TUT đầu tôi đã hướng dẫn các bạn giao tiếp COM với vi điều khiển. Giao tiếp như vậy coi như là xong rồi. Nhưng mà chúng ta thấy đây nhận được dữ liệu đã tốt rồi nhưng phải vẽ được Graph thì mới giải quyết được bài toán.
- Trong 1 lần tình cờ thấy 1 ông anh vẽ đồ thị với VB6 nhìn thấy mà thèm cứ nghĩ ông này thật là *Pờ-rô-féc-săn-nô*. Nhưng sau khi tìm hiểu thì hóa ra ông ta sài phần mềm vẽ teacher gì đó (hic không nhớ!). Tôi mới mon men lên mạng tìm cho C# thì có cũng khá nhiều công cụ mạnh mẽ cho vẽ đồ thị, nhìn rất chuyên nghiệp cứ như là dùng Excel để vẽ ấy. Kể đến là có *ZedGraph*, *Component one*
- Trong khuôn khổ tài liệu này tôi sẽ cố gắng hướng dẫn chi tiết vẽ đồ thị với *ZedGraph*, tuy không đẹp và pro bằng *Component One* nhưng là công cụ mạnh về code nên rất dễ làm chủ code của mình.
- Hi vọng tài liệu sẽ giúp ích được *anh em mới vào nghề* chơi vi điều khiển và tập code với C#

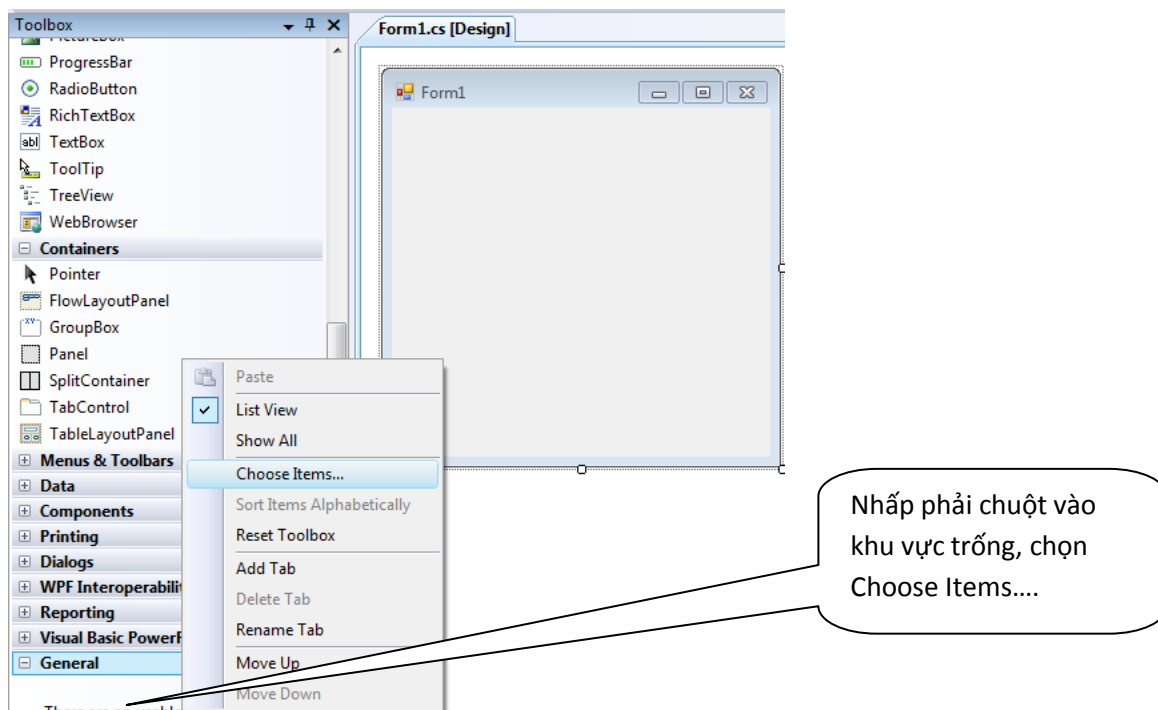
2. Tạo Project



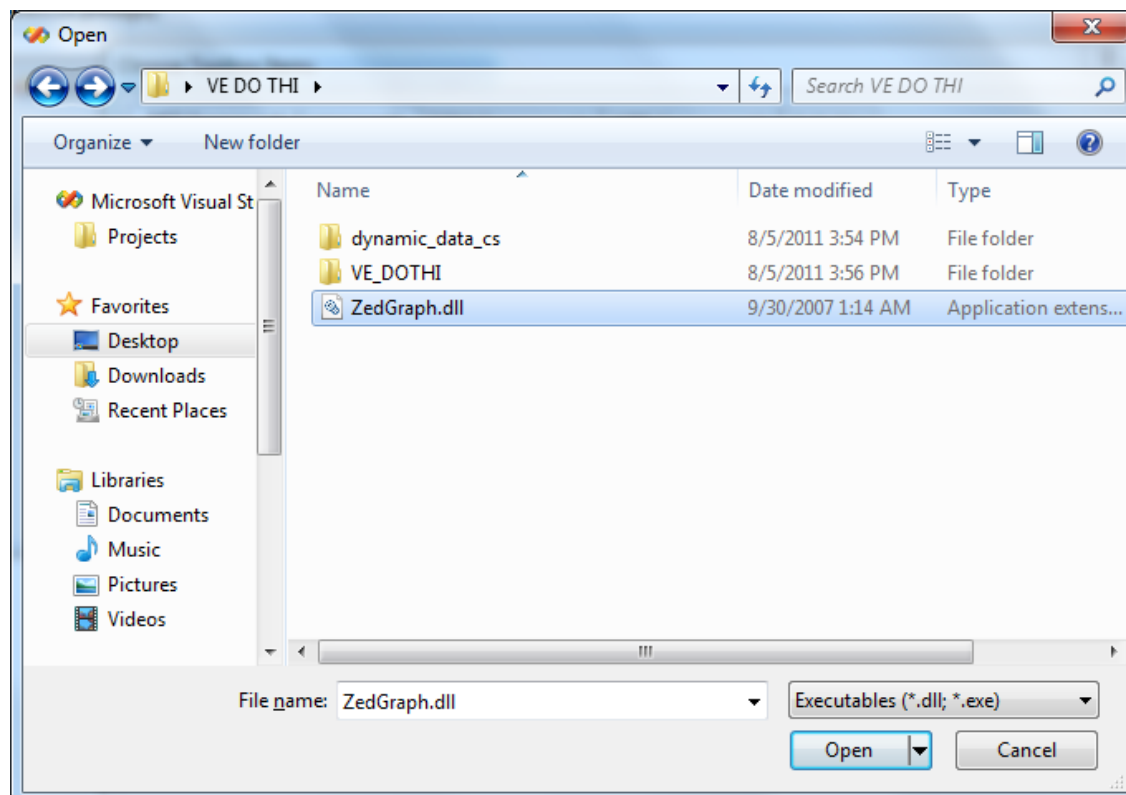
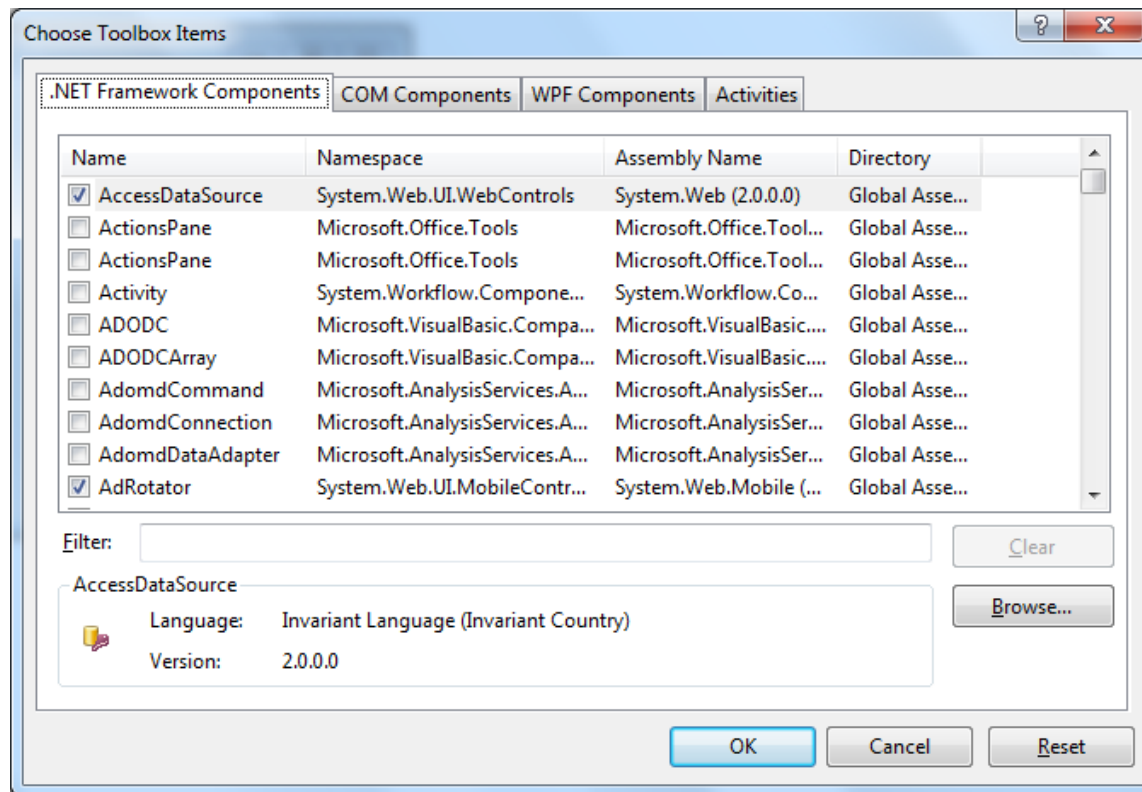
- *File / New / Project...*
- Và đặt tên cho nó



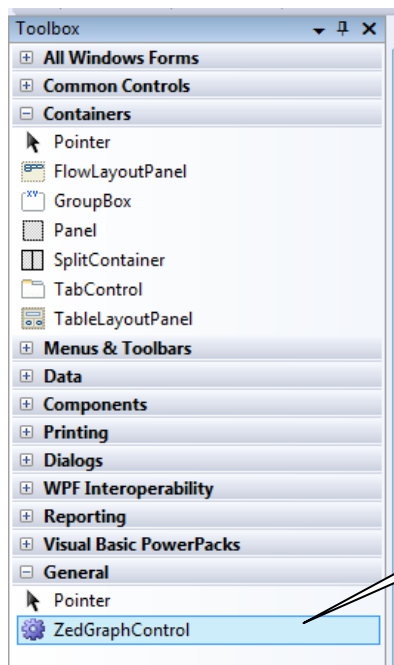
- Để sử dụng được control ZedGraph chúng ta phải add nó vào Toolbox bằng cách



- Hiện ra 1 cửa sổ mới và bạn phải nhấn **Browse...** chỉ ra đường dẫn đến **ZedGraph.dll**

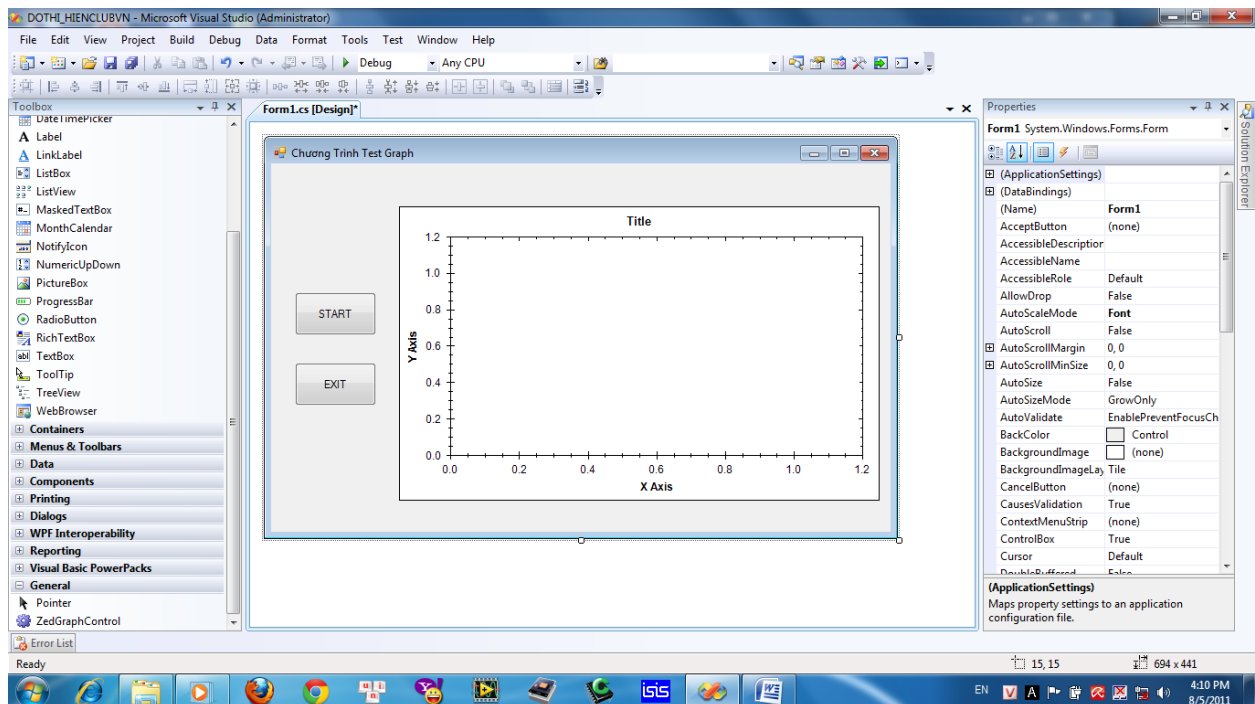


- Khi ToolBox có Control như bên dưới là OK rồi.

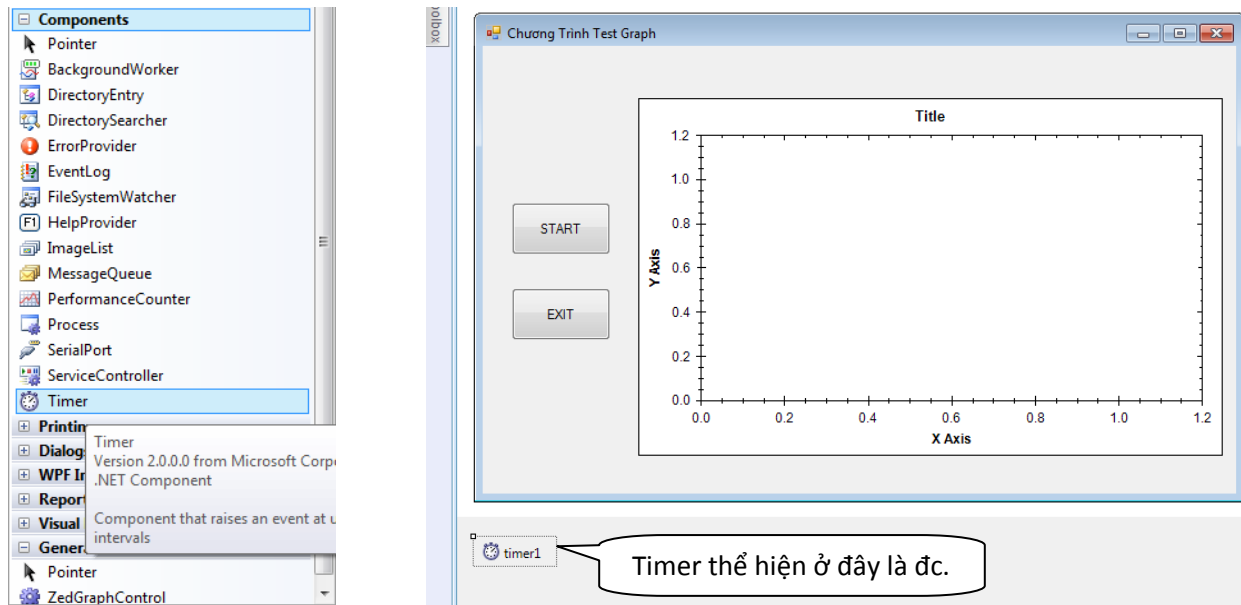


Em nó ở đây, chỉ việc kéo thả vào Form là OK

- Sau khi kéo thả và thêm các button được như hình bên dưới.



- Vậy là phần giao diện là ổn rồi, bây giờ kéo thêm timer ra, mục đích để tạo thời gian chạy trong phần đồ thị.



3. VIẾT CODE

- Trước hết thêm thư viện vào

```
// chú ý : Khai báo thư viện này
using ZedGraph;
```

- Khai báo trước Form1

```
// Khai báo 1 biến dùng cho timer, chạy cột thời gian tính bằng ms
```

```
int tickStart = 0;
```

```
public Form1()
{
    InitializeComponent();
}
```

- Đúp vào tiêu đề để lấy sự kiện Load và xem code Form1_Load bên dưới.

```
private void Form1_Load(object sender, EventArgs e)
{
    // khi khởi động sẽ được chạy
    GraphPane myPane = zedGraphControl1.GraphPane; // Khai báo sửa
    dụng Graph loại GraphPane;
    // Các thông tin cho đồ thị của mình
    myPane.Title.Text = "Đây là title trên cùng, bạn thay đổi cho phù
    hợp với chương trình";
    myPane.XAxis.Title.Text = "Thời gian cột X";
    myPane.YAxis.Title.Text = "Tiêu đề cột Y";
    // Định nghĩa list để vẽ đồ thị. Để các bạn hiểu rõ cơ chế làm
    việc ở đây khai báo 2 list điểm <=> 2 đường đồ thị
    RollingPointPairList list1 = new RollingPointPairList(1200);
```

```

        // Ở đây sử dụng list với 1200 điểm (có thể thêm nhiều liệu tại
        đây)
        RollingPointPairList list2 = new RollingPointPairList( 1200 );
        // dòng dưới là định nghĩa curve để vẽ.
        LineItem curve1 = myPane.AddCurve("đường 1", list1, Color.Red,
        SymbolType.None); // Color màu đỏ, đặc trưng cho đường 1
        // SymbolType là kiểu biểu thị đồ thị : điểm, đường tròn, tam
        giác ....
        LineItem curve2 = myPane.AddCurve("đường 2", list2, Color.Blue,
        SymbolType.None); // Color màu Xanh, đặc trưng cho đường 2

        // ví dụ khoảng cách là 50ms 1 lần
        timer1.Interval = 50;
        //timer1.Enabled = true; // Kích hoạt cho timer1
        //timer1.Start(); // Chạy Timer1

        // Định hiện thị cho trục thời gian (Trục X)
        myPane.XAxis.Scale.Min = 0; // Min = 0;
        myPane.XAxis.Scale.Max = 30; // Mã = 30;
        myPane.XAxis.Scale.MinorStep = 1; // Đơn vị chia nhỏ nhất 1
        myPane.XAxis.Scale.MajorStep = 5; // Đơn vị chia lớn 5

        // Gọi hàm xác định cỡ trục
        zedGraphControl1.AxisChange();

        // Khởi động timer về vị trí ban đầu
        tickStart = Environment.TickCount;
    }

```

// Để tiện cho việc sử dụng chúng ta sẽ xây dựng 1 hàm draw phục vụ cho việc vẽ đồ thị

```

    public void draw(double setpoint1, double setpoint2) // Ở ví dụ này
    chúng ta có 2 đường
    {
        if (zedGraphControl1.GraphPane.CurveList.Count <= 0)
            return;
        // Kiểm tra việc khởi tạo các đường curve
        // Đưa về điểm xuất phát
        LineItem curve1 = zedGraphControl1.GraphPane.CurveList[0] as
        LineItem;
        LineItem curve2 = zedGraphControl1.GraphPane.CurveList[1] as
        LineItem;
        if (curve1 == null)
            return;
        if (curve2 == null)
            return;

        // list chứa các điểm.
        // Get the PointPairList
        IPointListEdit list1 = curve1.Points as IPointListEdit;
        IPointListEdit list2 = curve2.Points as IPointListEdit;

```

```

        if (list1 == null)
            return;
        if (list2 == null)
            return;

        // Time được tính bằng ms
        double time = (Environment.TickCount - tickStart) / 1000.0;

        // Tính toán giá trị hiện thị

        // Muốn hiện thị cái gì thì chỉ việc thay vào setpointx
        list1.Add(time, setpoint1); // Đây chính là hàm hiển thị dữ liệu
của mình lên đồ thị
        // list2.Add(time, setpoint2); // Đây chính là hàm hiển thị dữ
liệu của mình lên đồ thị
        // Ko vẽ setpoint2 mà thử vẽ đồ thị hình sin với 3 seconds per
cycle
        list2.Add(time, Math.Sin(2.0 * Math.PI * time / 3.0));
        // đoạn chương trình thực hiện vẽ đồ thị
        Scale xScale = zedGraphControl1.GraphPane.XAxis.Scale;
        if (time > xScale.Max - xScale.MajorStep)
        {
            //xScale.Max = time + xScale.MajorStep;
            // xScale.Min = xScale.Max - 30.0; // Timer chạy qua 30 sẽ tự
động dịch chuyển tịnh tiến sang trái
            // Nếu ko muốn dịch chuyển mà chạy bắt đầu từ 0 thì :
            xScale.Min = 0;
            if (button3.Text == "COMPACT")
            {
                xScale.Max = time + xScale.MajorStep;
                xScale.Min = xScale.Max - 30.0;
            }
            else
            {
                xScale.Max = time + xScale.MajorStep;
                xScale.Min = 0;
            }
        }

        // Vẽ đồ thị
        zedGraphControl1.AxisChange();
        // Force a redraw
        zedGraphControl1.Invalidate();
    }

```

Để gọi hàm vẽ chúng ta sẽ kích hoạt cho Timer1 chạy, đúp vào Timer1 để lấy sự kiện, sau đó viết code như code tham khảo bên dưới.

```

private void timer1_Tick(object sender, EventArgs e)
{
    draw(5, 20);
}

```

- Để các bạn hiểu hơn về cơ chế làm việc, tôi đã thêm các *Button* điều khiển mục đích của nó là : Cho mở *Timer* để bắt đầu vẽ đồ thị bằng **START**, cho hiện thị 2 dạng chạy thời gian bằng button **COMPACT** và thoát chương trình bằng Exit

Các bạn có thể tham khảo code mẫu.

```
int check = 0; // để cho khi nhấn Start là sẽ bắt đầu từ điểm 0
private void button1_Click(object sender, EventArgs e)
{
    if (button1.Text == "START")
    {
        timer1.Enabled = true;
        button1.Text = "STOP";
        // Khởi động timer về vị trí ban đầu
        if (check == 0)
        {
            tickStart = Environment.TickCount;
            check = 1;
        }
    }
    else
    {
        timer1.Enabled = false;
        button1.Text = "START";
    }
}

private void button2_Click(object sender, EventArgs e)
{
    this.Close();
}

private void button3_Click(object sender, EventArgs e)
{
    if (button3.Text == "COMPACT") button3.Text = "SROLL";
    else button3.Text = "COMPACT";
}

private void tácGiảToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form2 frm = new Form2();
    frm.ShowDialog();
}
```

4. Full CODE để tham khảo

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
```



```
// chú ý : Khai báo thư viện này
using ZedGraph;

namespace DOTHI_HIENCLUBVN
{
    public partial class Form1 : Form
    {
        // Khai báo 1 biến dùng cho timer, chạy cột thời gian tính bằng ms
        int tickStart = 0;
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            // khi khởi động sẽ được chạy
            GraphPane myPane = zedGraphControl1.GraphPane; // Khai báo sử
            dụng Graph loại GraphPane;
            // Các thông tin cho đồ thị của mình
            myPane.Title.Text = "Đây là title trên cùng, bạn thay đổi cho phù
            hợp với chương trình";
            myPane.XAxis.Title.Text = "Thời gian cột X";
            myPane.YAxis.Title.Text = "Tiêu đề cột Y";
            // Định nghĩa list để vẽ đồ thị. Để các bạn hiểu rõ cơ chế làm
            việc ở đây khai báo 2 list điểm <=> 2 đường đồ thị
            RollingPointPairList list1 = new RollingPointPairList(1200);
            // Ở đây sử dụng list với 1200 điểm (có thể thêm nhiều liệu tại
            đây)
            RollingPointPairList list2 = new RollingPointPairList( 1200 );
            // dòng dưới là định nghĩa curve để vẽ.
            LineItem curve1 = myPane.AddCurve("đường 1", list1, Color.Red,
            SymbolType.None); // Color màu đỏ, đặc trưng cho đường 1
            // SymbolType là kiểu biểu thị đồ thị : điểm, đường tròn, tam
            giác ....
            LineItem curve2 = myPane.AddCurve("đường 2", list2, Color.Blue,
            SymbolType.None); // Color màu Xanh, đặc trưng cho đường 2

            // ví dụ khoảng cách là 50ms 1 lần
            timer1.Interval = 50;
            //timer1.Enabled = true; // Kích hoạt cho timer1
            //timer1.Start(); // Chạy Timer1

            // Định hiện thị cho trục thời gian (Trục X)
            myPane.XAxis.Scale.Min = 0; // Min = 0;
            myPane.XAxis.Scale.Max = 30; // Mă = 30;
            myPane.XAxis.Scale.MinorStep = 1; // Đơn vị chia nhỏ nhất 1
            myPane.XAxis.Scale.MajorStep = 5; // Đơn vị chia lớn 5

            // Gọi hàm xác định cỡ trục
            zedGraphControl1.AxisChange();

            // Khởi động timer về vị trí ban đầu
            tickStart = Environment.TickCount;
        }
    }
}
```

```

// Để tiện cho việc sử dụng chúng ta sẽ xây dựng 1 hàm draw phục vụ
cho việc vẽ đồ thị
public void draw(double setpoint1, double setpoint2) // Ở ví dụ này
chúng ta có 2 đường
{

    if (zedGraphControl1.GraphPane.CurveList.Count <= 0)
        return;
    // Kiểm tra việc khởi tạo các đường curve
    // Đưa về điểm xuất phát
    LineItem curve1 = zedGraphControl1.GraphPane.CurveList[0] as LineItem;
    LineItem curve2 = zedGraphControl1.GraphPane.CurveList[1] as LineItem;
    if (curve1 == null)
        return;
    if (curve2 == null)
        return;

    // list chứa các điểm.
    // Get the PointPairList
    IPointListEdit list1 = curve1.Points as IPointListEdit;
    IPointListEdit list2 = curve2.Points as IPointListEdit;

    if (list1 == null)
        return;
    if (list2 == null)
        return;

    // Time được tính bằng ms
    double time = (Environment.TickCount - tickStart) / 1000.0;

    // Tính toán giá trị hiện thị

    // Muốn hiện thị cái gì thì chỉ việc thay vào setpointx
    list1.Add(time, setpoint1); // Đây chính là hàm hiển thị dữ liệu
của mình lên đồ thị
    // list2.Add(time, setpoint2); // Đây chính là hàm hiển thị dữ
liệu của mình lên đồ thị
    // Ko vẽ setpoint2 mà thử vẽ đồ thị hình sin với 3 seconds per
cycle
    list2.Add(time, Math.Sin(2.0 * Math.PI * time / 3.0));
    // đoạn chương trình thực hiện vẽ đồ thị
    Scale xScale = zedGraphControl1.GraphPane.XAxis.Scale;
    if (time > xScale.Max - xScale.MajorStep)
    {
        //xScale.Max = time + xScale.MajorStep;
        // xScale.Min = xScale.Max - 30.0; // Timer chạy qua 30 sẽ tự
động dịch chuyển tịnh tiến sang trái
        // Nếu ko muốn dịch chuyển mà chạy bắt đầu từ 0 thì :
xScale.Min = 0;
        if (button3.Text == "COMPACT")
        {
            xScale.Max = time + xScale.MajorStep;
            xScale.Min = xScale.Max - 30.0;
        }
        else
        {

```

```

        xScale.Max = time + xScale.MajorStep;
        xScale.Min = 0;
    }
}

// Vẽ đồ thị
zedGraphControl1.AxisChange();
// Force a redraw
zedGraphControl1.Invalidate();
}

private void timer1_Tick(object sender, EventArgs e)
{
    draw(5,20);
}
int check = 0; // để cho khi nhấn Start là sẽ bắt đầu từ điểm 0
private void button1_Click(object sender, EventArgs e)
{
    if (button1.Text == "START")
    {
        timer1.Enabled = true;
        button1.Text = "STOP";
        // Khởi động timer về vị trí ban đầu
        if (check == 0) { tickStart = Environment.TickCount; check = 1; }
    }
    else
    {
        timer1.Enabled = false;
        button1.Text = "START";
    }
}

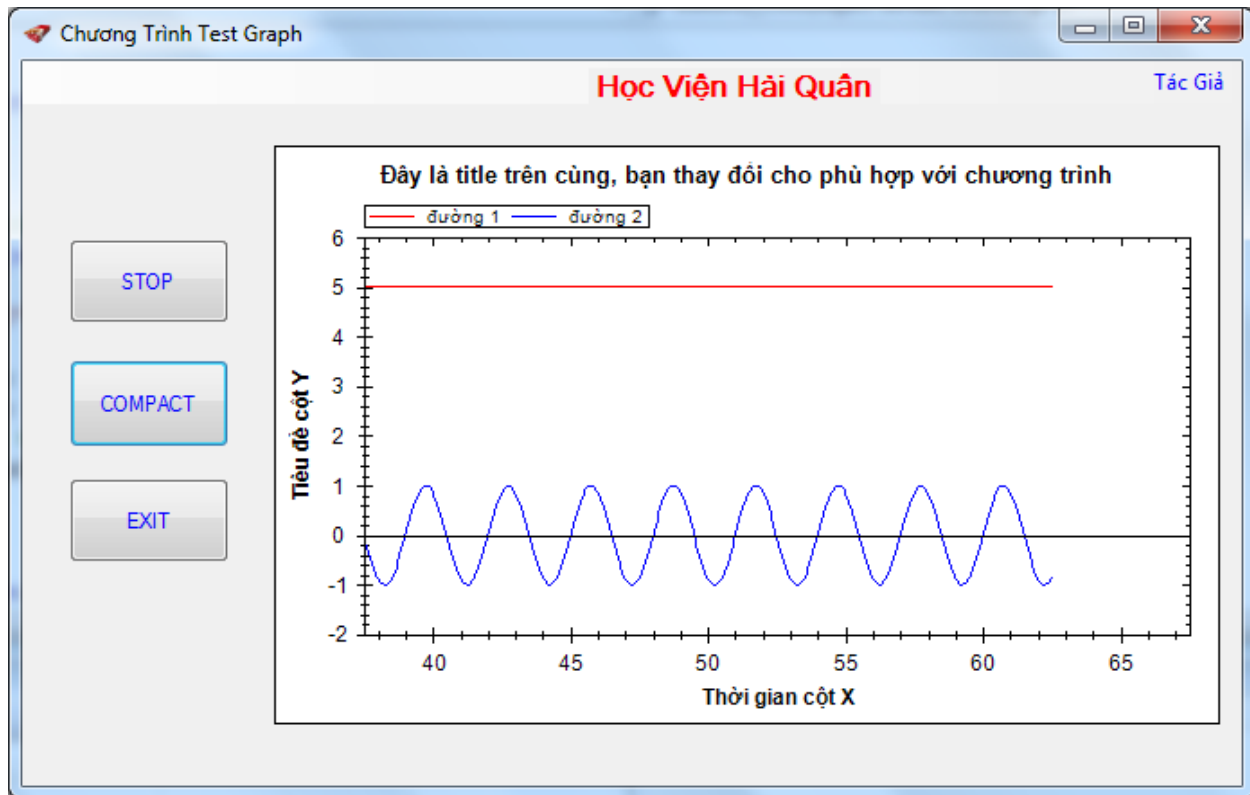
private void button2_Click(object sender, EventArgs e)
{
    this.Close();
}

private void button3_Click(object sender, EventArgs e)
{
    if (button3.Text == "COMPACT") button3.Text = "SCROLL";
    else button3.Text = "COMPACT";
}

private void tácGiảToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form2 frm = new Form2();
    frm.ShowDialog();
}
}
}

```

5. Một số hình ảnh về giao diện



Hình trên là thời gian chạy (đồ thị dịch tịnh tiến), hình bên dưới bắt đầu từ 0 (dồn lại)

