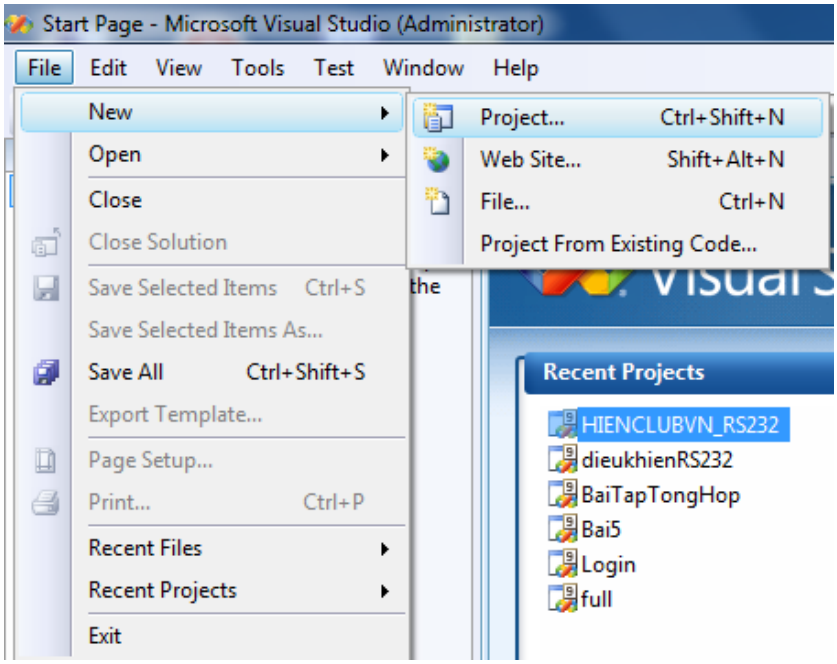
HƯỚNG DẪN VIẾT PHẦN MỀM TEST RS232 CHO VI ĐIỀU KHIỂN VỚI VISUAL STUDIO C# 2008

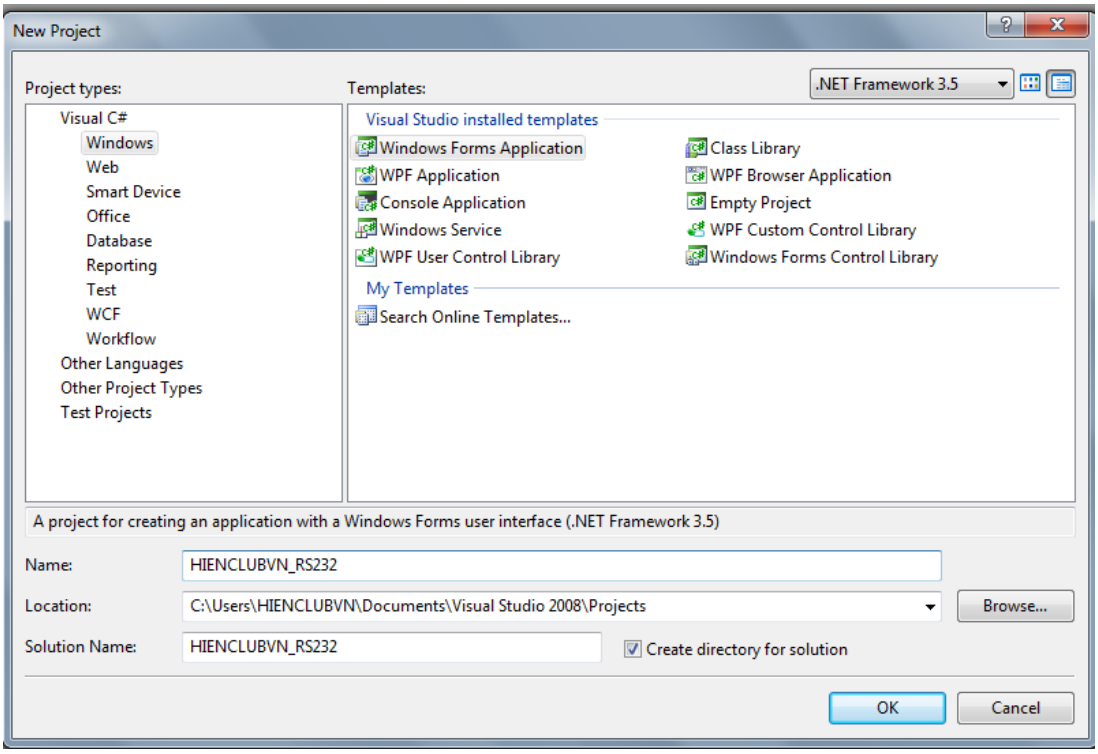
- Chào các bạn, hôm nay ngày *05/08/2011*, đang nghỉ phép về nhà buồn quá không có việc gì để làm. Nên làm bài TUT hướng dẫn anh em newbie sài C# 2008 để giao tiếp với vi điều khiển.
- Như chúng ta đã biết đấy chơi vi điều khiển mà ko giao tiếp với PC thì chả có gì phải nói cả. Trên các diễn đàn cũng đã bàn luận về vấn đề này khá nhiều nhất là với VB6, VC++ … Nhưng với C# thì có lẽ là hơi ít. Ít ra thì tôi cũng đau đầu khi mới làm vì ko tìm thấy tài liệu ưng ý.
- Có lẻ điểm mạnh của C# chúng ta ko cần phải nói đến nữa. Bài TUT sẽ đi sâu vào phần giao tiếp còn phần C# các bạn nên đọc qua về nó. Nói chung là nó khả dễ để tiếp thu so với thằng VB cũ kĩ và VC++ phức tạp. Bài TUT viết cho newbie nên tôi đã rất cố gắng làm cho nó chi tiết để các bạn dễ hiểu hơn.
- Trước khi vào chúng ta sẽ điểm qua 1 số nội dung trong TUT này:
  - SerialPort với C#
  - 1 số hiệu ứng (phản ứng) của C#

1. Đầu tiên là mở VC# lên, và tạo 1 project mới.

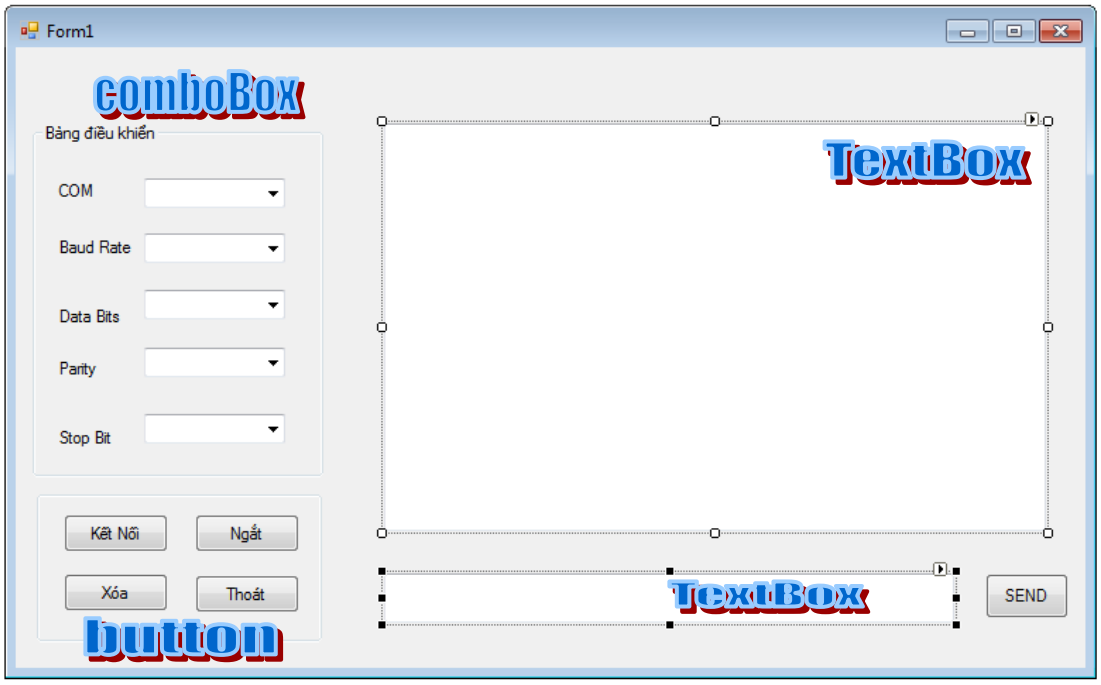
- *File / New / Project…*



- Chọn Windows Forms Application và đặt tên cho chúng. Đưa đến cho chúng ta 1 giao diện của lập trình Form. Giống như các bạn làm bằng VB vậy

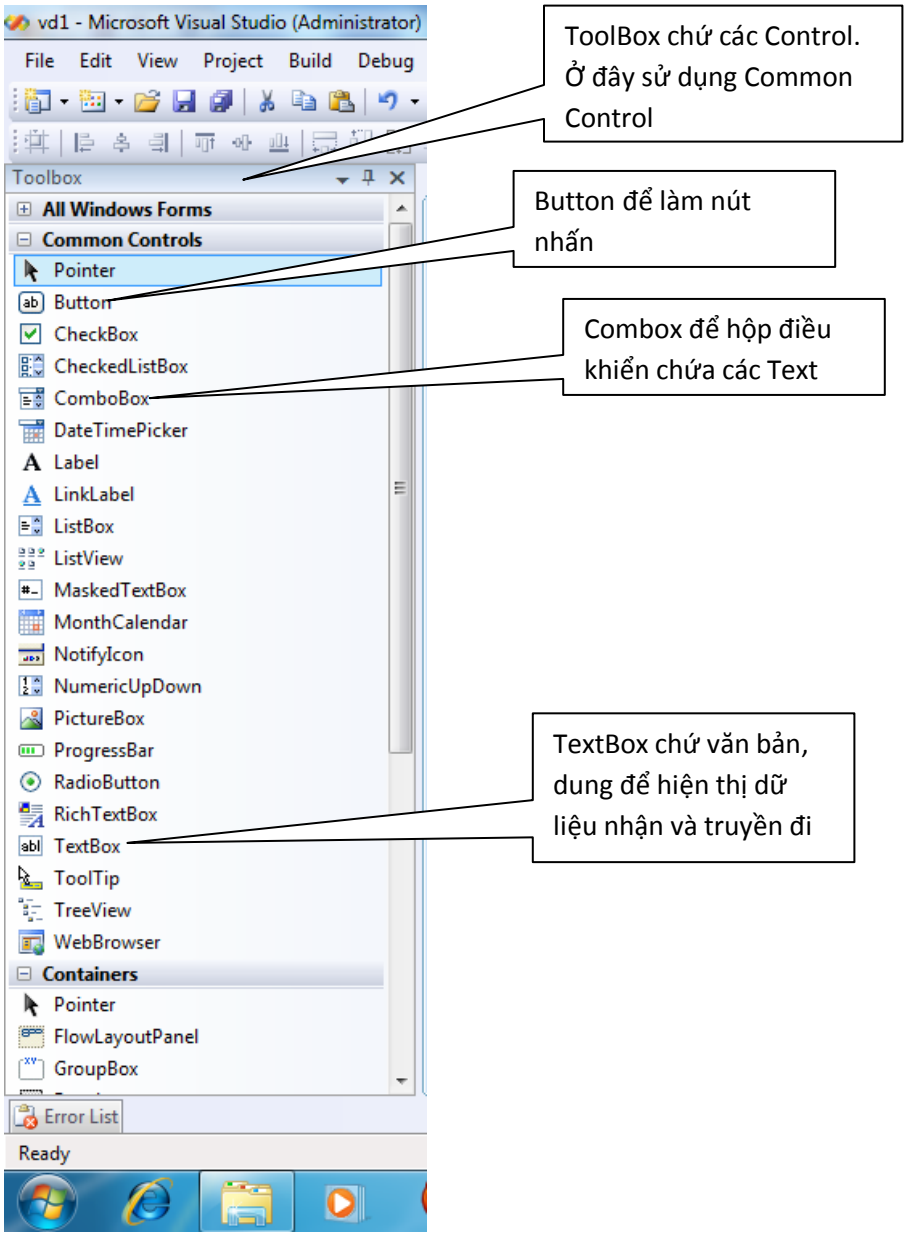


2. Tiếp đến là chúng ta sẽ xây dựng giao diện như hình bên dưới

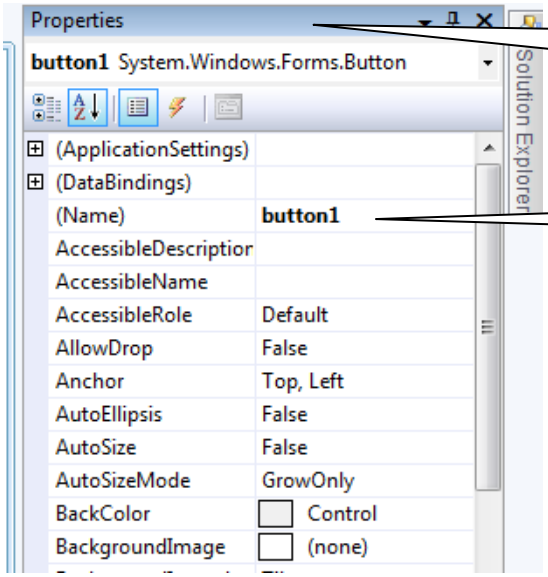


- Như hình trên cấu trúc rất đơn giản chỉ chứ ComboBox, Button và TextBox
- COM, BaudRate, Data Bits, Parity, Stop Bit là các ComboBox

- Kết nối, Ngắt, Xóa, Thoát và SEND là các Button
- Còn lại 2 ống trống có viền xung quanh chính là các TextBox
- Chúng ta sẽ tiến hành tạo và đặt tên cho chúng.

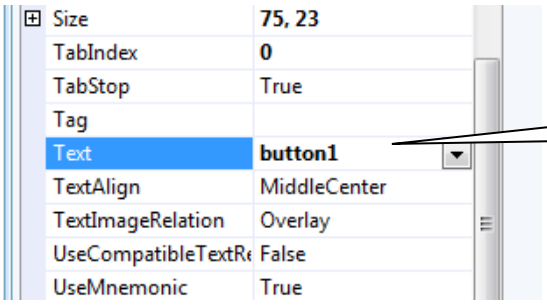


- Và nhớ phải đặt tên cho chúng sau mỗi lần kéo ra



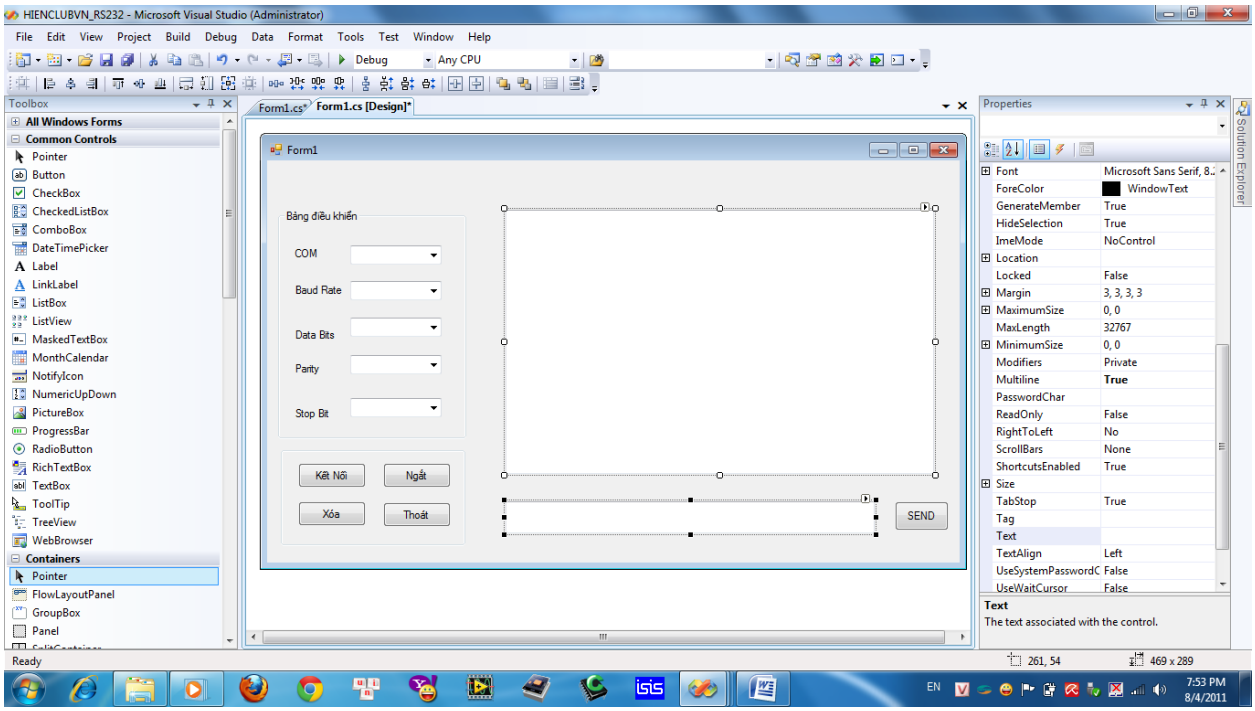
Properties : Nơi để thay đổi các thuộc tính của đối tượng.

Tên sẽ được đặt ở đây. Đây chính là tên của đối tượng.

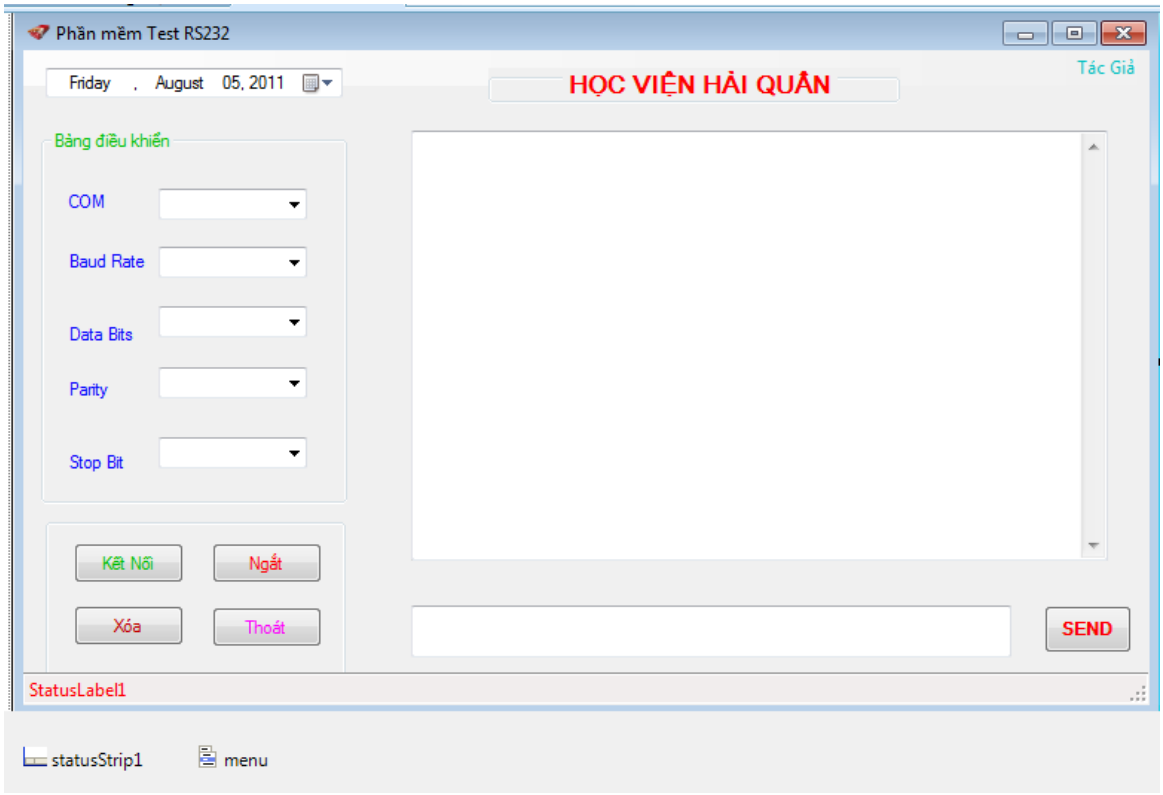


Kéo xuống bên dưới, đặt tên ở Text. Tên này sẽ hiện thị lên Button

Tương tự như vậy với các phần còn lại, chúng ta sẽ xây dựng được giao diện

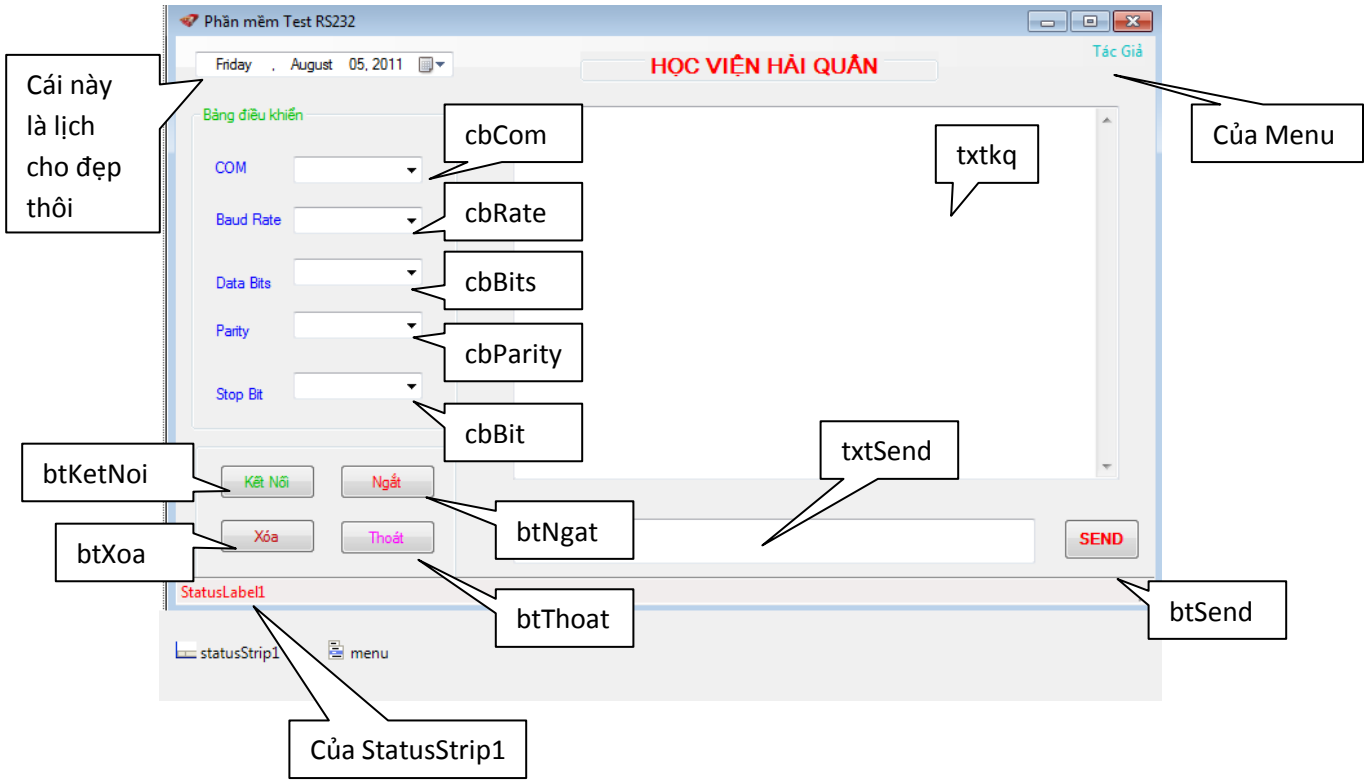


Thêm 1 chút mắm, muối chúng ta sẽ có giao diện như hình bên dưới



- Giao diện như vậy là đã xong, phần quan trọng chính là code

- Trước khi đến code ta sẽ giải thích nội dung của giao diện cũng như tên của các control cho tiện trong việc tham khảo code.



3. Phần CODE

```
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;
// Thêm 3 em này vào là OK, để sài SerialPort
using System.IO;
using System.IO.Ports;
using System.Xml;
```

- Chuyển sang phần code bằng cách nhấp phải chọn viewCode, chúng ta sẽ thêm 2 thư viện IO và xml vào.

```
namespace HIENCLUBVN_RS232
{
    public partial class Form1 : Form
    {
        SerialPort P = new SerialPort(); // Khai báo 1 Object SerialPort mới.
        string InputData = String.Empty; // Khai báo string buff dùng cho
hiển thị dữ liệu sau này.
```

```
        delegate void SetTextCallback(string text); // Khai bao delegate
SetTextCallBack voi tham so string
```

- Trước mắt bạn chỉ quan tâm đến câu lệnh `SerialPort P = new SerialPort(); // Khai báo 1 Object SerialPort mới.`
- Bước tiếp là chúng ta tạo dữ liệu cho các comboBox, bước này bạn nên copy và dán vào project của mình nhớ thay tên các combo cho đúng với tên mà bạn đặt.

```
public Form1()

        {
            InitializeComponent();
            // Cài đặt các thông số cho COM
            // Mảng string port để chứ tất cả các cổng COM đang có trên máy
tính
            string[] ports = SerialPort.GetPortNames();

            // Thêm toàn bộ các COM đã tìm được vào combox cbCom
            cbCom.Items.AddRange(ports); // Sử dụng AddRange thay vì dùng
foreach
            P.ReadTimeout = 1000;
            // Khai báo hàm delegate bằng phương thức DataReceived của Object
SerialPort;
            // Cái này khi có sự kiện nhận dữ liệu sẽ nhảy đến phương thức
DataReceive
            // Nếu ko hiểu đoạn này bạn có thể tìm hiểu về Delegate, còn ko
cứ COPY . Ko cần quan tâm
           P.DataReceived += new SerialDataReceivedEventHandler(DataReceive);

            // Cài đặt cho BaudRate
            string[] BaudRate = { "1200", "2400", "4800", "9600", "19200",
"38400", "57600", "115200" };
            cbRate.Items.AddRange(BaudRate);

            // Cài đặt cho DataBits
            string[] Databits = { "6", "7", "8" };
            cbBits.Items.AddRange(Databits);

            //Cho Parity
            string[] Parity = { "None", "Odd", "Even" };
            cbParity.Items.AddRange(Parity);

            //Cho Stop bit
            string[] stopbit = { "1", "1.5", "2" };
            cbBit.Items.AddRange(stopbit);
            // Mấy cái này khá đơn giản, bạn đừng hỏi vì sao.  cứ COPY paste
cho nhanh. :D
        }
```

Bạn nên lưu ý là :

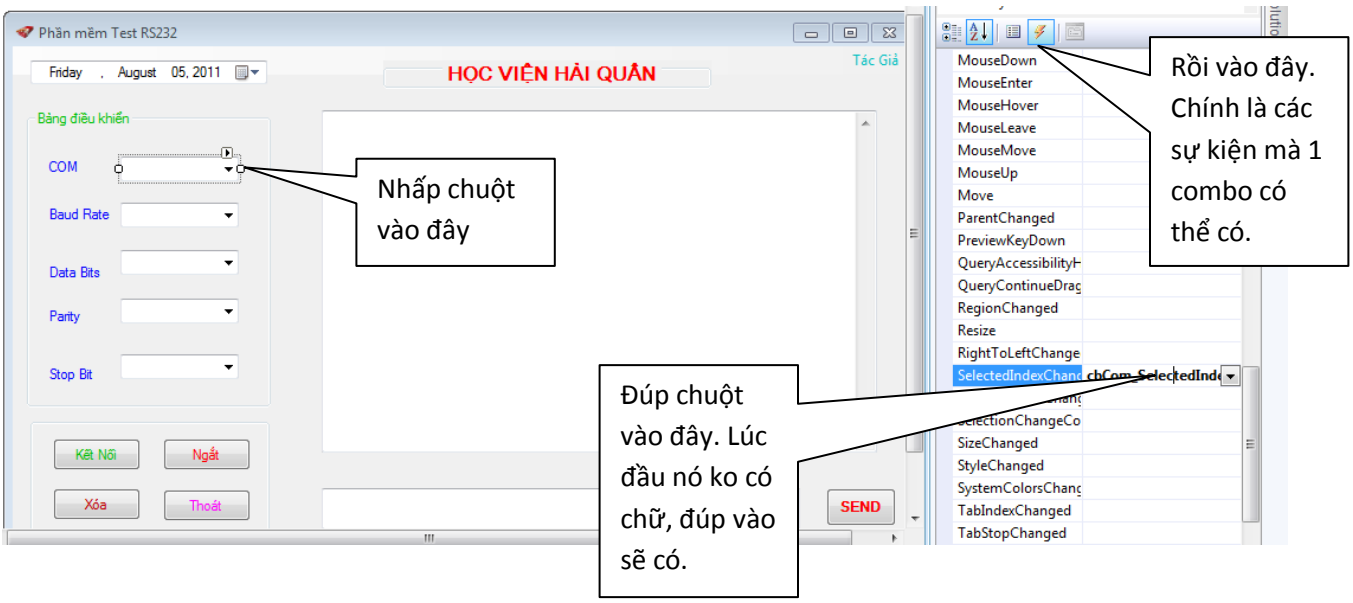
```
public Form1()

        {
```

```
            InitializeComponent();
```

Đã được tạo ra trước đó bởi C#, chúng ta Copy, Paste thì phải sau hàm khởi tạo `InitializeComponent();`

Nói chung, cơ bản đến đây là xong phần khởi tạo các combox cho SerialPort rồi.

Đến đây bạn có thể cài đặt các thông số cho rs232 bằng tay, nếu làm cái này thì rs232 có thể gọi là Full, và ta có thể hoàn toàn làm chủ nó về những cái nhỏ nhất.



Lúc đúp vào ta đã tạo ra 1 sự kiện là khi thay đổi các giá trị trên combo thì chúng sẽ được gọi đến hàm phục vụ, và công việc của chúng ta sẽ là ở đây.

```
        private void cbCom_SelectedIndexChanged(object sender, EventArgs e)
          {
              if (P.IsOpen)
              {
                  P.Close(); // Nếu đang mở Port thì phải đóng lại
              }
              P.PortName = cbCom.SelectedItem.ToString(); // Gán PortName bằng
COM đã chọn
          }
```

- Như ví dụ trên là sẽ gọi đến hàm gán giá trị cổng đang chọn cho PortName
- Tương tự như vậy chúng ta sẽ làm cho toàn bộ comboBox, các bạn có thể tham khảo code dưới để hiểu rõ hơn.

```
private void cbCom_SelectedIndexChanged(object sender, EventArgs e)
          {
              if (P.IsOpen)
              {
```

```
                P.Close(); // Nếu đang mở Port thì phải đóng lại
            }
            P.PortName = cbCom.SelectedItem.ToString(); // Gán PortName bằng
COM đã chọn
        }

        private void cbRate_SelectedIndexChanged(object sender, EventArgs e)
        {
            if (P.IsOpen)
            {
                P.Close();
            }
            P.BaudRate = Convert.ToInt32(cbRate.Text);
        }

        private void cbBits_SelectedIndexChanged(object sender, EventArgs e)
        {
            if (P.IsOpen)
            {
                P.Close();
            }
            P.DataBits = Convert.ToInt32(cbBits.Text);
        }

       private void cbParity_SelectedIndexChanged(object sender, EventArgs e)
        {
            if (P.IsOpen)
            {
                P.Close();
            }
            // Với thằng Parity hơn lằng nhằng. Nhưng cũng OK thôi. ^_^
            switch (cbParity.SelectedItem.ToString())
            {
                case "Odd":
                    P.Parity = Parity.Odd;
                    break;
                case "None":
                    P.Parity = Parity.None;
                    break;
                case "Even":
                    P.Parity = Parity.Even;
                    break;
            }
        }

        private void cbBit_SelectedIndexChanged(object sender, EventArgs e)
        {
            if (P.IsOpen)
            {
                P.Close();
            }
            switch (cbBit.SelectedItem.ToString())
            {
                case "1":
                    P.StopBits = StopBits.One;
                    break;
```

```
                case "1.5":
                    P.StopBits = StopBits.OnePointFive;
                    break;
                case "2":
                    P.StopBits = StopBits.Two;
                    break;
            }
        }
```

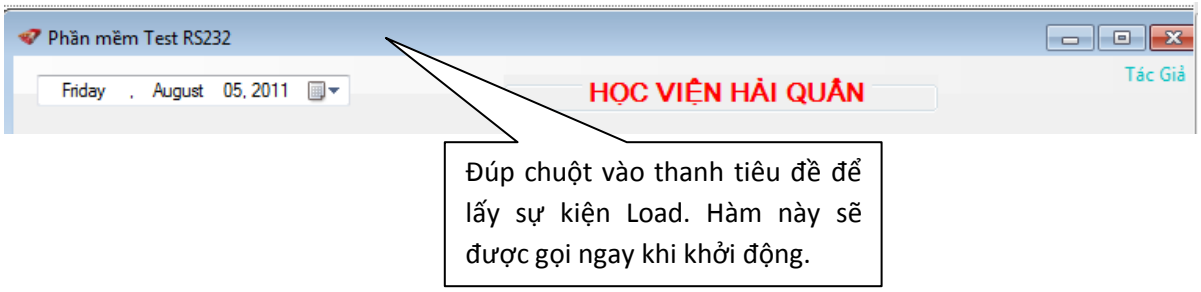
Vậy là đã xong. Công việc khởi tạo đã hoàn tất, còn bây giờ là công việc quan trọng nhất. Đó chính là xây dựng các hàm, thủ tục cho việc nhận và truyền dữ liệu qua COM

```
// Hàm này được sự kiện nhận dử liệu gọi đến. Mục đích để hiển thị thôi
        private void DataReceive(object obj, SerialDataReceivedEventArgs e)
        {
            InputData = P.ReadExisting();
            if (InputData != String.Empty)
            {
                // txtIn.Text = InputData; // Ko dùng đc như thế này vì khác
threads .
                SetText(InputData); // Chính vì vậy phải sử dụng ủy quyền tại
đây. Gọi delegate đã khai báo trước đó.
            }
        }
        // Hàm của em nó là ở đây. Đừng hỏi vì sao lại thế.
        private void SetText(string text)
        {
            if (this.txtkq.InvokeRequired)
            {
                SetTextCallback d = new SetTextCallback(SetText); // khởi tạo
1 delegate mới gọi đến SetText
                this.Invoke(d, new object[] { text });
            }
            else this.txtkq.Text += text;
        }
        // Toàn bộ cái này bạn nên COPY, nó cũng làm tôi đau đầu. :D
```

Đến đây là phần nhận đã xong, tiếp đến phần gửi dữ liệu đi. Bạn nhấp đúp vào button SEND, để lấy sự kiện rồi viết hàm theo bên dưới

```
// Đến hàm gửi data xuống COM
        private void btSend_Click(object sender, EventArgs e)
        {
            if (P.IsOpen)
            {
                if (txtSend.Text == "") MessageBox.Show("Chưa có dữ
liệu!","Thông Báo");
                else P.Write(txtSend.Text);
            }
            else MessageBox.Show("COM chưa mở.", "Thông báo",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            txtSend.Clear();
        }
        // Đến đây coi như mọi việc đã ngon lành cành đào rồi.
```

```
        // Để chọn tiện trong việc Test thì chúng ta sẽ làm thêm bước nữa.
cho mấy cái thông số hay dùng được chọn. Ko cần thiết nếu bạn cảm thấy ko
cần.
```




```
  private void Form1_Load(object sender, EventArgs e) // sẽ được gọi khi mở
chương trình.
        {
            cbCom.SelectedIndex = 0; // chọn COM được tìm thấy đầu tiên
            cbRate.SelectedIndex = 3; // 9600
            cbBits.SelectedIndex = 2; // 8
            cbParity.SelectedIndex = 0; // None
            cbBit.SelectedIndex = 0; // None
            // Hiện thị Status cho Pro tí
            status.Text = "Hãy chọn 1 cổng COM để kết nối.";
        }
```

Bước cuối cùng là giải quyết 4 button còn lại để kết nối, ngắt kết nối, thoát khỏi chương trình và xóa dữ liệu củ. Bước này khá đơn giản bạn chỉ việc nhấp đúp vào từng button để lấy sự kiện, có thể tham khảo code dưới.

```
private void btKetNoi_Click(object sender, EventArgs e)
        {
            try
            {
                P.Open();
                btNgat.Enabled = true;
                btKetNoi.Enabled = false;
                // Hiện thị Status
                status.Text = "Đang kết nối với cổng " +
cbCom.SelectedItem.ToString();
            }
            catch (Exception ex)
            {
                MessageBox.Show("Không kết nối được.","Thử
lại",MessageBoxButtons.OK,MessageBoxIcon.Error);
            }
        }

        private void btNgat_Click(object sender, EventArgs e)
        {
            P.Close();
            btKetNoi.Enabled = true;
            btNgat.Enabled = false;
            // Hiện thị Status
```

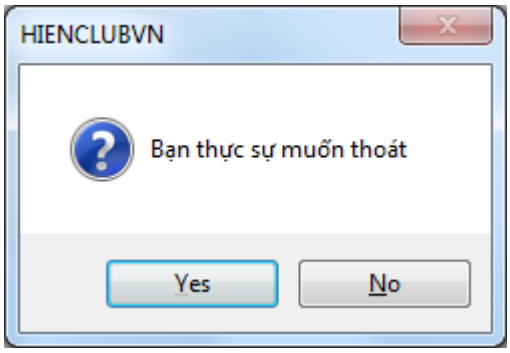
```
            status.Text = "Đã Ngắt Kết Nối";
        }

        private void btThoat_Click(object sender, EventArgs e)
        {
            DialogResult kq = MessageBox.Show("Bạn thực sự muốn thoát",
"HIENCLUBVN", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (kq == DialogResult.Yes)
            {
                MessageBox.Show("Cảm ơn bạn đã sử dụng chương
trình","HIENCLUBVN");
                this.Close();
            }
        }

        private void btXoa_Click(object sender, EventArgs e)
        {
            txtkq.Text = "";
            txtSend.Text = "";
        }
```

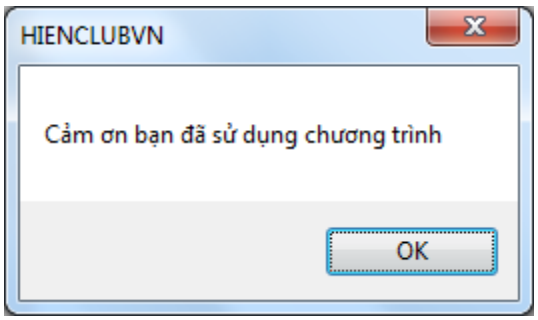
Đến đây là mọi việc đã ngon lành cành đào rồi. Để cá nhân hóa chương trình của mình bạn có thể thêm các label như ví dụ là : HỌC VIỆN HẢI QUÂN hay là menu như Tác Giả và 1 số hiệu ứng khác mang tính chuyên nghiệp để *Profectional* hơn :D

- Bằng cách là thêm 1 project mới: *Project / Add Windows Form …*

Và tạo nó theo mẫu là mình ưu ý. Cuối cùng là thêm hàm sự kiện khi nhấn vào tác giả như code dưới.

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

// Lưu ý : Tác Giả là 1 control của ToolStripMenu bạn có thể lôi nó ra từ
Toolbox
```





Lưu ý : Nhấn F6 để biên dịch lỗi trong quá trình code và F5 để chạy chương trình.

4. Toàn bộ code để tiện 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;
// Thêm 3 em này vào là OK, để sài SerialPort
using System.IO;
using System.IO.Ports;
using System.Xml;
// Bắt đầu code
namespace HIENCLUBVN_RS232
{
    public partial class Form1 : Form
    {
        SerialPort P = new SerialPort(); // Khai báo 1 Object SerialPort mới.
        string InputData = String.Empty; // Khai báo string buff dùng cho
hiển thị dữ liệu sau này.
        delegate void SetTextCallback(string text); // Khai bao delegate
SetTextCallBack voi tham so string
        public Form1()
        {
            InitializeComponent();
            // Cài đặt các thông số cho COM
            // Mảng string port để chứ tất cả các cổng COM đang có trên máy
tính
            string[] ports = SerialPort.GetPortNames();

            // Thêm toàn bộ các COM đã tìm được vào combox cbCom
            cbCom.Items.AddRange(ports); // Sử dụng AddRange thay vì dùng
foreach
            P.ReadTimeout = 1000;
            // Khai báo hàm delegate bằng phương thức DataReceived của Object
SerialPort;
            // Cái này khi có sự kiện nhận dữ liệu sẽ nhảy đến phương thức
DataReceive
            // Nếu ko hiểu đoạn này bạn có thể tìm hiểu về Delegate, còn ko
cứ COPY . Ko cần quan tâm
```

```
            P.DataReceived += new
SerialDataReceivedEventHandler(DataReceive);

            // Cài đặt cho BaudRate
            string[] BaudRate = { "1200", "2400", "4800", "9600", "19200",
"38400", "57600", "115200" };
            cbRate.Items.AddRange(BaudRate);

            // Cài đặt cho DataBits
            string[] Databits = { "6", "7", "8" };
            cbBits.Items.AddRange(Databits);

            //Cho Parity
            string[] Parity = { "None", "Odd", "Even" };
            cbParity.Items.AddRange(Parity);

            //Cho Stop bit
            string[] stopbit = { "1", "1.5", "2" };
            cbBit.Items.AddRange(stopbit);
            // Mấy cái này khá đơn giản, bạn đừng hỏi vì sao.  cứ COPY paste
cho nhanh. :D
        }
        private void cbCom_SelectedIndexChanged(object sender, EventArgs e)
        {
            if (P.IsOpen)
            {
                P.Close(); // Nếu đang mở Port thì phải đóng lại
            }
            P.PortName = cbCom.SelectedItem.ToString(); // Gán PortName bằng
COM đã chọn
        }

        private void cbRate_SelectedIndexChanged(object sender, EventArgs e)
        {
            if (P.IsOpen)
            {
                P.Close();
            }
            P.BaudRate = Convert.ToInt32(cbRate.Text);
        }

        private void cbBits_SelectedIndexChanged(object sender, EventArgs e)
        {
            if (P.IsOpen)
            {
                P.Close();
            }
            P.DataBits = Convert.ToInt32(cbBits.Text);
        }

        private void cbParity_SelectedIndexChanged(object sender, EventArgs
e)
        {
            if (P.IsOpen)
            {
                P.Close();
```

```
            }
            // Với thằng Parity hơn lằng nhằng. Nhưng cũng OK thôi. ^_^
            switch (cbParity.SelectedItem.ToString())
            {
                case "Odd":
                    P.Parity = Parity.Odd;
                    break;
                case "None":
                    P.Parity = Parity.None;
                    break;
                case "Even":
                    P.Parity = Parity.Even;
                    break;
            }
        }

        private void cbBit_SelectedIndexChanged(object sender, EventArgs e)
        {
            if (P.IsOpen)
            {
                P.Close();
            }
            switch (cbBit.SelectedItem.ToString())
            {
                case "1":
                    P.StopBits = StopBits.One;
                    break;
                case "1.5":
                    P.StopBits = StopBits.OnePointFive;
                    break;
                case "2":
                    P.StopBits = StopBits.Two;
                    break;
            }
        }
        // Hàm này được sự kiện nhận dử liệu gọi đến. Mục đích để hiển thị
thôi
        private void DataReceive(object obj, SerialDataReceivedEventArgs e)
        {
            InputData = P.ReadExisting();
            if (InputData != String.Empty)
            {
                // txtIn.Text = InputData; // Ko dùng đc như thế này vì khác
threads .
                SetText(InputData); // Chính vì vậy phải sử dụng ủy quyền tại
đây. Gọi delegate đã khai báo trước đó.
            }
        }
        // Hàm của em nó là ở đây. Đừng hỏi vì sao lại thế.
        private void SetText(string text)
        {
            if (this.txtkq.InvokeRequired)
            {
                SetTextCallback d = new SetTextCallback(SetText); // khởi tạo
1 delegate mới gọi đến SetText
                this.Invoke(d, new object[] { text });
```

```
            }
            else this.txtkq.Text += text;
        }
        // Toàn bộ cái này bạn nên COPY, nó cũng làm tôi đau đầu. :D
        // Đến hàm gửi data xuống COM
        private void btSend_Click(object sender, EventArgs e)
        {
            if (P.IsOpen)
            {
                if (txtSend.Text == "") MessageBox.Show("Chưa có dữ
liệu!","Thông Báo");
                else P.Write(txtSend.Text);
            }
            else MessageBox.Show("COM chưa mở.", "Thông báo",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            txtSend.Clear();
        }
        // Đến đây coi như mọi việc đã ngon lành cành đào rồi.
        // Để chọn tiện trong việc Test thì chúng ta sẽ làm thêm bước nữa.
cho mấy cái thông số
        // hay dùng được chọn. Ko cần thiết nếu bạn cảm thấy ko cần.
        private void Form1_Load(object sender, EventArgs e) // sẽ được gọi
khi mở chương trình.
        {
            cbCom.SelectedIndex = 0;
            cbRate.SelectedIndex = 3; // 9600
            cbBits.SelectedIndex = 2; // 8
            cbParity.SelectedIndex = 0; // None
            cbBit.SelectedIndex = 0; // None
            // Hiện thị Status cho Pro tí
            status.Text = "Hãy chọn 1 cổng COM để kết nối.";
        }

        private void btKetNoi_Click(object sender, EventArgs e)
        {
            try
            {
                P.Open();
                btNgat.Enabled = true;
                btKetNoi.Enabled = false;
                // Hiện thị Status
                status.Text = "Đang kết nối với cổng " +
cbCom.SelectedItem.ToString();
            }
            catch (Exception ex)
            {
                MessageBox.Show("Không kết nối được.","Thử
lại",MessageBoxButtons.OK,MessageBoxIcon.Error);
            }
        }

        private void btNgat_Click(object sender, EventArgs e)
        {
            P.Close();
            btKetNoi.Enabled = true;
            btNgat.Enabled = false;
```

```
            // Hiện thị Status
            status.Text = "Đã Ngắt Kết Nối";
        }

        private void btThoat_Click(object sender, EventArgs e)
        {
            DialogResult kq = MessageBox.Show("Bạn thực sự muốn thoát",
"HIENCLUBVN", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (kq == DialogResult.Yes)
            {
                MessageBox.Show("Cảm on bạn đã sử dụng chương
trình","HIENCLUBVN");
                this.Close();
            }
        }

        private void btXoa_Click(object sender, EventArgs e)
        {
            txtkq.Text = "";
            txtSend.Text = "";
        }

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


    }
}
```