

C# (C Sharp) - Nhận dữ liệu từ Arduino

[Đỗ Hữu Toàn](#) gửi vào Chủ nhật, 26 Tháng 6, 2016 - 22:40

• [76856 LƯỢT XEM](#)

1. Giới thiệu

Lấy nguồn cảm hứng từ bài viết [C# \(C Sharp\) – Một cách điều khiển Arduino bằng máy tính](#) của bạn Đình Hồng Thái . Mình sẽ xây dựng phần tiếp theo của loạt bài giao tiếp với Arduino bằng máy tính thông qua C#, dựa trên công nghệ Window Form Application....Nói nghe nguy hiểm vậy, nhưng thực ra bài viết khá đơn giản...keke. Nội dung chính của bài này đó là: “Nhận dữ liệu từ Arduino đến máy tính, thông qua phần mềm tự tạo bằng C#” !!!!!

Thực ra nó cũng tương tự như Serial Monitor của Arduino IDE...nhưng nếu trong các cuộc thi, bạn sẽ không thể tự vỗ ngực rồi nói Serial Monitor là do mình tự tạo ra đc..... Vì vậy, bạn hãy tự tạo phần mềm theo phong cách riêng của chính mình.....!!!

2. Cách làm

Để làm được phần mềm này....các bạn phải có công cụ hỗ trợ đó là: Microsoft Visual Studio hoặc Visual C# - Một công cụ của các lập trình viên chuyên nghiệp (các bạn có thể xem lại bài [C# \(C Sharp\) – Một cách điều khiển Arduino bằng máy tính](#) của bạn Đình Hồng Thái để biết thêm chi tiết nhé!). Sau khi cài đặt xong, triển khai!

a. Phần giao diện

b. Phần code

```
1. using System;
2.
3. using System.Collections.Generic;
4.
5. using System.ComponentModel;
6.
7. using System.Data;
8.
9. using System.Drawing;
10.
11. using System.Linq;
12.
13. using System.Text;
14.
15. using System.Windows.Forms;
16.
17. // Thêm 3 em này vào là OK, để sài SerialPort
18.
19. using System.IO;
20.
21. using System.IO.Ports;
```

```

22.
23. using System.Xml;
24.
25. // Bắt đầu code
26.
27.
28.
29. namespace nhandulieuCOM
30.
31. {
32.
33.     public partial class Form1 : Form
34.
35.     {
36.
37.         string InputData = String.Empty; // Khai báo string buff
           dùng cho hiển thị dữ liệu sau này.
38.
39.         delegate void SetTextCallback(string text); // Khai báo
           delegate SetTextCallBack voi tham so string
40.
41.         public Form1()
42.
43.         {
44.
45.             InitializeComponent();
46.
47.             // Khai báo hàm delegate bằng phương thức DataReceived
           của Object SerialPort;
48.
49.             // Cái này khi có sự kiện nhận dữ liệu sẽ nhảy đến
           phương thức DataReceive
50.
51.             // Nếu ko hiểu đoạn này bạn có thể tìm hiểu về
           Delegate, còn ko cứ COPY . Ko cần quan tâm
52.
53.             serialPort1.DataReceived += new
           SerialDataReceivedEventHandler(DataReceive);
54.
55.             string[] BaudRate = { "1200", "2400", "4800", "9600",
           "19200", "38400", "57600", "115200" };
56.
57.             comboBox2.Items.AddRange(BaudRate);
58.
59.         }
60.
61.
62.
63.         private void Form1_Load(object sender, EventArgs e)
64.
65.         {
66.
67.             comboBox1.DataSource = SerialPort.GetPortNames(); //
           Quét các cổng COM đang hoạt động lên comboBox1

```

```

68.
69.         comboBox2.SelectedIndex = 3;
70.
71.         // Set tỉ chế độ mặc định cho nó nguy hiểm tí!!!!
72.
73.     }
74.
75.
76.
77. // Cái này là chế độ làm tươi... Hiển thị trạng thái, các bạn xem lại
    bài của Đinh Hồng Thái để biết thêm chi tiết
78.
79.     private void timer1_Tick(object sender, EventArgs e)
80.     {
81.
82.
83.         if (!serialPort1.IsOpen)
84.         {
85.
86.
87.
88.
89.             label5.Text = ("Chưa kết nối");
90.
91.             label5.ForeColor = Color.Red;
92.
93.         }
94.
95.         else if (serialPort1.IsOpen)
96.         {
97.
98.
99.
100.
101.             label5.Text = ("Đã kết nối");
102.
103.             label5.ForeColor = Color.Green;
104.
105.
106.         }
107.     }
108.
109. }
110.
111.     private void DataReceive(object obj,
        SerialDataReceivedEventArgs e)
112.     {
113.
114.
115.         InputData = serialPort1.ReadExisting();
116.
117.         if (InputData != String.Empty)
118.
119.         {

```

```

120.
121.             // textbox1 = InputData; // Ko dùng đc như thế này
    vì khác threads .
122.
123.             SetText(InputData); // Chính vì vậy phải sử dụng ủy
    quyền tại đây. Gọi delegate đã khai báo trước đó.
124.
125.         }
126.
127.
128.
129.     }
130.
131.     // Hàm của em nó là ở đây. Đừng hỏi vì sao lại thế.
132.
133.     private void SetText(string text)
134.     {
135.
136.         if (this.textBox1.InvokeRequired)
137.         {
138.
139.             SetTextCallback d = new SetTextCallback(SetText);
140.             // khởi tạo 1 delegate mới gọi đến SetText
141.
142.             this.Invoke(d, new object[] { text });
143.
144.         }
145.
146.         else this.textBox1.Text += text;
147.
148.     }
149.
150.
151. // Cái dưới tương tự bài của Thái...tạo button mở và đóng Serial
152.
153.     private void button2_Click(object sender, EventArgs e)
154.     {
155.
156.         if (!serialPort1.IsOpen)
157.         {
158.
159.             serialPort1.PortName = comboBox1.Text;
160.
161.             serialPort1.BaudRate =
162.             Convert.ToInt32(comboBox2.Text);
163.
164.             serialPort1.Open();
165.
166.         }
167.
168.     }
169.

```

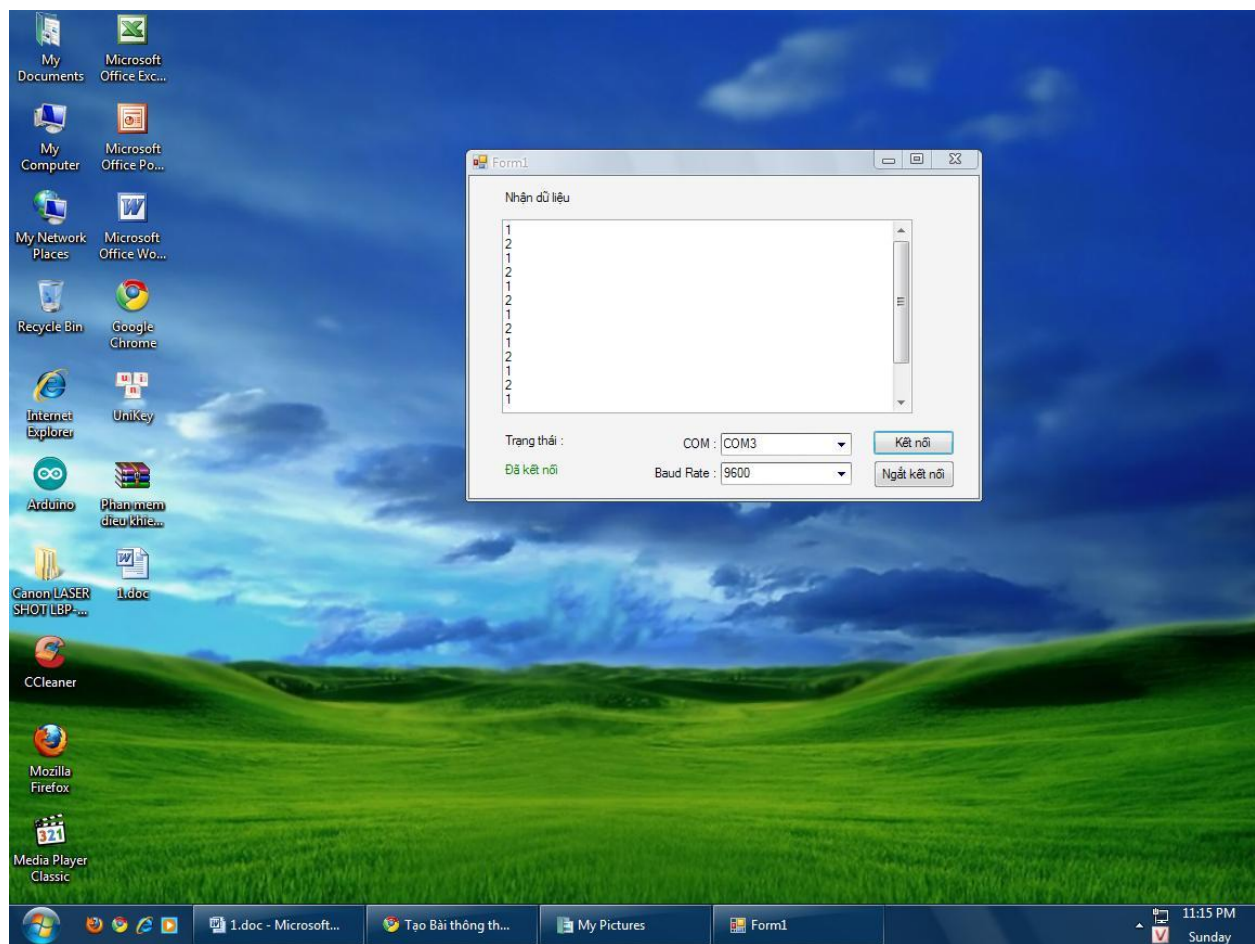
```

170.
171.
172.
173.     private void button1_Click(object sender, EventArgs e)
174.     {
175.
176.         serialPort1.Close();
177.
178.     }
179.
180.
181. }
182.
183. }

```

3. Kết quả

Và đây là thành quả:



Ở bài sắp tới. mình sẽ hướng dẫn chi tiết cách xây dựng lại Serial Monitor bằng C# với đầy đủ chức năng nhận và gửi với giao diện Max Vip...Các bạn hãy chờ đón và ủng hộ mình bằng cách Rate Note bài nhé!!