

CHÀO MỪNG NGÀY THÀNH LẬP TRƯỜNG 01/4/2018

KẾT NỐI GIỮA PHẦN MỀM C# VÀ THIẾT BỊ NI

CONNECT BETWEEN C# SOFTWARE AND NI EQUIPMENTS

ĐỒNG XUÂN THÌN

Khoa Điện, Trường Cao đẳng VMU

Tóm tắt

NI là thiết bị phần cứng của hãng National Instrument. Đây là một hãng rất nổi tiếng với nhiều thiết bị phần cứng được sử dụng trong lĩnh vực đo lường. Thiết bị NI rất dễ kết nối với phần mềm Matlab và LabView, hai phần mềm này có thể tạo giao diện giám sát và điều khiển. Ngoài ra, phần mềm C# cũng có thể được sử dụng để tạo giao diện giám sát và điều khiển, nhưng việc kết nối giữa phần mềm C# và thiết bị NI rất phức tạp.

Từ khóa: C#, NI, kết nối C# và NI, NI9215.

Abstract

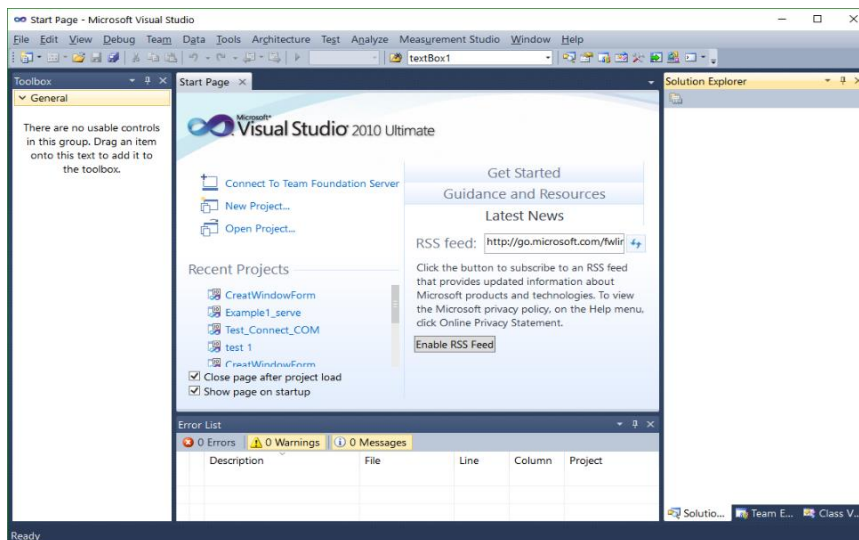
NI equipments are the products of National Instrument Company that is very famous in measurement. It can easily connect with Matlab and LabView, these software can be used to create control interfaces. In additional, C# software also can create interfaces but it really complicate in programming and handshake with NI equipments.

Keywords: C#, NI, connect C#&NI, NI9215.

1. Giới thiệu

1.1. Phần mềm C#

Phần mềm C# là một phần rất quen thuộc nằm trong bộ Visual.NET. Hầu hết những người xây dựng giao diện để giám sát và điều khiển hệ thống thực trên máy tính đều sử dụng phần mềm này. Việc tải về và cài đặt phần mềm này rất đơn giản, chúng ta có thể tham khảo các nguồn tài liệu trên internet. Giao diện chính của phần mềm Microsoft Visual Studio khi cài đặt thành công như Hình 1.



Hình 1. Phần mềm Microsoft Visual Studio 2010



Hình 2. NI 9215

1.2. Thiết bị NI

Như đã biết, National Instrument là một hãng thiết bị rất nổi tiếng trong lĩnh vực đo lường và điều khiển. Một loại thiết bị thông dụng hay được sử dụng để kết hợp với phần mềm Matlab hoặc LabView đó là card PCI.

NI cũng là thiết bị phần cứng được phát triển bởi hãng này, đây là thiết bị chuyên dùng trong lĩnh vực đo lường và điều khiển. NI khá đa dạng về chủng loại, có thể kết hợp được với các phần mềm Matlab, LabView và cả C#. Việc lựa chọn thiết bị NI phụ thuộc vào yêu cầu kỹ thuật của bài toán điều khiển, một số thông số kỹ thuật cần thiết cho việc lựa chọn NI như sau:

- Loại tín hiệu vào/ra;
- Giới hạn mức tín hiệu vào/ra;

CHÀO MỪNG NGÀY THÀNH LẬP TRƯỜNG 01/4/2018

- Số kênh tín hiệu vào/ra;
- Độ phân giải động của tín hiệu;
- Tần số lấy mẫu tối đa cho phép.

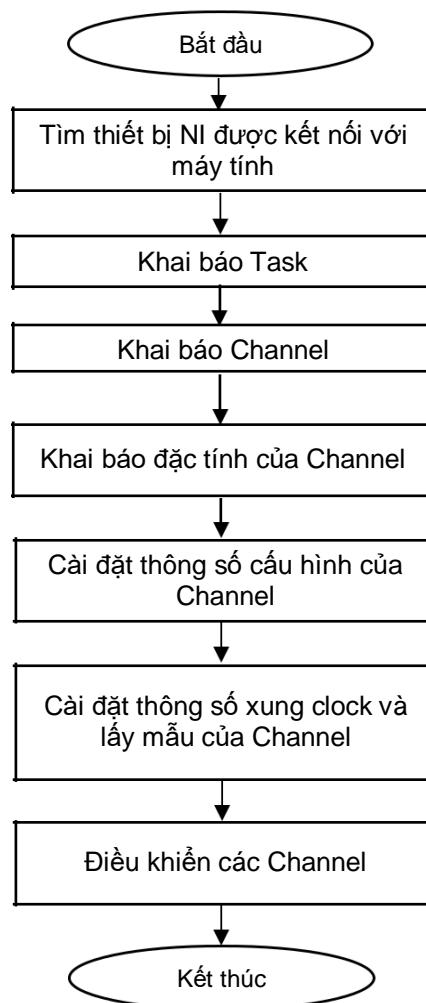
Để thực nghiệm việc kết nối giữa phần mềm C# tác giả lựa chọn thiết bị NI có tên NI9215. Hình ảnh thực tế của NI9215 như Hình 2. Thông số cơ bản của NI9215 như sau:

- Số lượng kênh vào: 04 kênh vào tương tự;
- Độ phân giải ADC: 16 bit;
- Dải tín hiệu vào: 10, 0 VDC;
- Tần số lấy mẫu: 100kHz;
- Nhiệt độ làm việc: -40°C tới 70°C;
- Điện áp bảo vệ: 30, 0 VDC.

2. Kết nối giữa phần mềm C# và thiết bị NI

2.1. Thuật toán giao tiếp giữa C# và NI

Để có thể bắt tay được giữa phần mềm C# và thiết bị NI thì ta cần thực hiện một số việc theo một trình tự nhất định. Trình tự này sẽ giúp cho phần mềm C# nhận diện được thiết bị NI và cài đặt các thông số phù hợp cần thiết cho thiết bị. Thứ tự công việc đó được thể hiện qua Hình 3. Thuật toán giao tiếp giữa C# và NI.



Hình 3. Thuật toán giao tiếp giữa C# và NI

Đầu tiên ta cần phải xác định các thiết bị NI được gắn với máy tính, sau đó chọn thiết bị NI mà chúng ta muốn dùng. Bước tiếp theo ta cần khai báo một “đối tượng” Task để quản lý toàn bộ việc cài đặt thông số cấu hình cho NI, cũng như nhận dữ liệu từ NI gửi về. Sau đó ta cần khai báo một đối tượng “được điều khiển” bởi Task, đó chính là Channel. Channel này sẽ liên kết trực tiếp

CHÀO MỪNG NGÀY THÀNH LẬP TRƯỜNG 01/4/2018

với channel vật lý của thiết bị NI, dựa vào các đặc tính kỹ thuật thực tế của thiết bị NI mà ta sử dụng và mục đích sử dụng để có thể khai báo một cách chính xác thông số này.

Bước tiếp theo là cài đặt các thông số cấu hình và xung clock cho channel, bước này cần lưu ý tra theo các thông số kỹ thuật của NI để nhận/gửi dữ liệu một cách chính xác. Cuối cùng là đọc/ghi dữ liệu từ/tới channel vật lý, nếu chỉ đọc/ghi dữ liệu một lần thì rất dễ nhưng không có tính ứng dụng nhiều trong điều khiển. Hầu hết các quá trình điều khiển là “Real Time” (RT) nên ta cần đọc/ghi dữ liệu từ/tới channel vật lý một cách liên tục, tức là ta phải dùng một vòng lặp vô hạn để phục vụ việc giao tiếp với NI. Như vậy thì chương trình điều khiển sẽ không thể làm gì khác ngoài việc giao tiếp với NI, để khắc phục tình trạng này thì ta phải dùng chương trình đa nhiệm.

2.2. Các thông số cấu hình cho NI

+) Khai báo Channel:

Việc khai báo channel có vai trò rất quan trọng trong việc xác định tín hiệu (vào hay ra) và dạng của tín hiệu (số hay tương tự). C# hỗ trợ 4 kiểu khai báo channels [1-3]:

- AIChannel;
- AOChannel;
- DIChannel;
- DOChannel.

+) Khai báo đặc tính Channel:

Khai báo đặc tính channel sẽ quyết định dạng của tín hiệu mà chúng ta sử dụng. C# hỗ trợ khai báo các dạng đặc tính như sau:

- AnalogSingleChannelReader;
- AnalogSingleChannelWriter;
- AnalogMultiChannelReader;
- AnalogMultiChannelWriter;
- DigitalSingleChannelReader;
- DigitalSingleChannelWriter;
- DigitalMultiChannelReader;
- DigitalMultiChannelWriter.

+) Cài đặt thông số cấu hình cho Channel:

Việc cấu hình chi tiết cho channel không phức tạp nhưng có rất nhiều dạng cảm biến khác nhau được hỗ trợ. Dưới đây là một số cấu hình đầu vào cho tín hiệu analog được hỗ trợ bởi phần mềm C# hay được sử dụng:

- CreateAccelerometerChannel: Dùng cho cảm biến gia tốc;
- CreateFrequencyVoltageChannel: Dùng để tính giá trị tần số của điện áp lấy mẫu;
- CreateLvdtChannel: Dùng cho cảm biến LVDT;
- CreateMicrophoneChannel: Dùng cho Microphone;
- CreateResistanceChannel: Dùng cho đầu vào là điện trở;
- CreateStrainGageChannel: Dùng cho cảm biến đo xoắn trục;
- CreateVoltageChannel: Dùng cho đầu vào là điện áp (thường dùng nhất).

Đối với mỗi hàm cấu hình cho tín hiệu thì sẽ có vài tham số cần cài đặt (có hướng dẫn cụ thể trong từng hàm cài đặt). Ví dụ với hàm CreateVoltageChannel() thì có các tham số cần cài đặt như sau:

- Tên của channel vật lý;
- Tên đặt cho channel vật lý;
- Dạng của điện áp đầu vào;
- Giới hạn trên và giới hạn dưới của tín hiệu vào;
- Đơn vị được quy đổi sau khi đọc giá trị đầu vào.

+) Cài đặt thông số xung clock và lấy mẫu của channel:

Thông số xung clock thay đổi phụ thuộc vào dạng của tín hiệu là digital hay analog và là tín hiệu vào hay tín hiệu ra. Với NI9215 thì có 04 thông số cơ bản cần cài đặt cho xung clock gồm:

- Nguồn xung: Có thể sử dụng nguồn xung clock ngoài hoặc xung clock của NI;

CHÀO MỪNG NGÀY THÀNH LẬP TRƯỜNG 01/4/2018

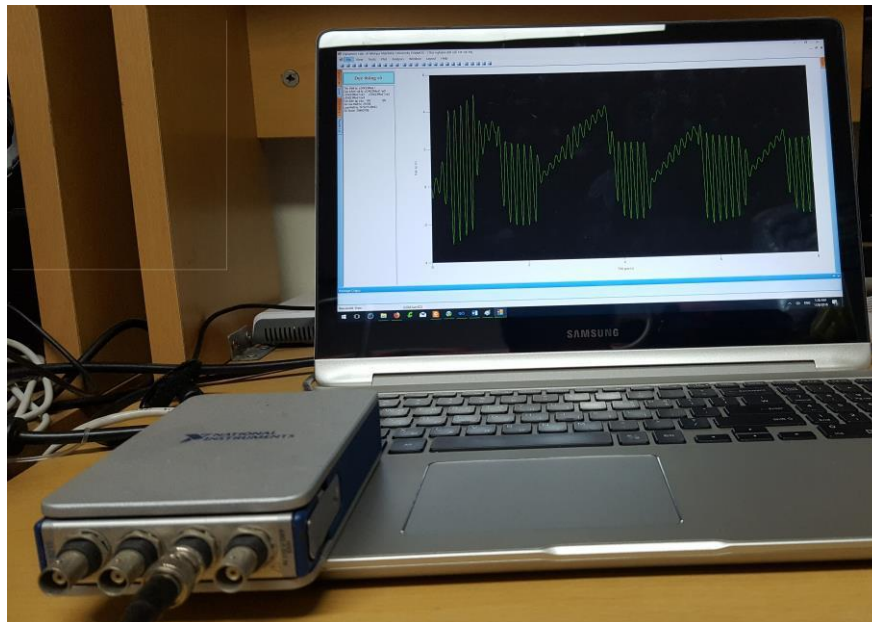
- Tốc độ lấy mẫu: Là số mẫu được lấy trong 1 giây, thông số này kiểu double;
- Cạnh tác dụng của xung clock: Có thể chọn cạnh lên hoặc cạnh xuống của xung clock;
- Chế độ lấy mẫu: Có 3 chế độ lấy mẫu nhưng ta thường dùng 2 chế độ đó là chế độ lấy mẫu xác định và chế độ lấy mẫu liên tục.

3. Kết quả và đánh giá

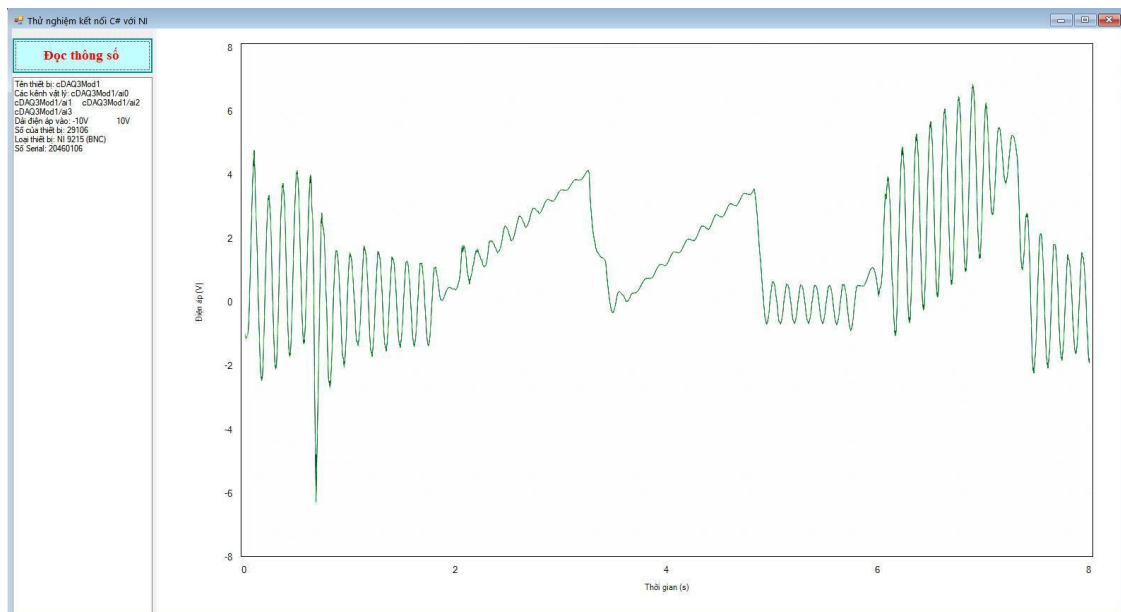
3.1. Kết quả

Tác giả đã thực hiện thử nghiệm kết nối giữa phần mềm C# với thiết bị NI9215. Hình ảnh thí nghiệm như Hình 4.

Thí nghiệm bao gồm: thiết bị phần cứng NI9215, giao diện lấy các thông số cơ bản của NI9215 được xây dựng bởi phần mềm C#, thiết bị NI được kết nối với laptop thông qua dây nối USB. Trên giao diện gồm 2 phần: Textbox để hiển thị các thông số của NI, graph để hiển thị dữ liệu đọc về từ NI.



Hình 4. Hình ảnh thử nghiệm



Hình 5. Kết quả mẫu tín hiệu thu được

Trong thử nghiệm này, tác giả đã dùng tín hiệu đầu vào là điện áp lấy từ cảm biến dao động. Kết quả của mẫu tín hiệu được vẽ thành đồ thị bởi khối Graph như Hình 5 [4], [5].

CHÀO MỪNG NGÀY THÀNH LẬP TRƯỜNG 01/4/2018

Ở đây tác giả chọn Sampling Rates = 8192, tức là lấy 8192 mẫu tín hiệu vào trong một giây. Tốc độ lấy mẫu có thể chọn bất kỳ và nhỏ hơn tốc độ lấy mẫu tối đa cho phép của thiết bị. Tuy nhiên, để thuận tiện cho các thuật toán xử lý số nhị phân thì ta nên chọn tốc độ lấy mẫu là 2n. Cũng trong thử nghiệm này thì tác giả lấy đúng 8192 mẫu, tức là lấy số mẫu tín hiệu trong một giây. Lưu ý, việc lấy bao nhiêu mẫu và tốc độ lấy mẫu không hoàn toàn phụ thuộc vào nhau, điều này còn phụ thuộc vào chế độ lấy mẫu ta cài đặt. Nếu chọn chế độ lấy mẫu là xác định thì số lượng mẫu sẽ nhỏ hơn hoặc bằng tốc độ lấy mẫu, còn với chế độ lấy mẫu là liên tục thì số lượng mẫu không phụ thuộc vào tốc độ lấy mẫu. Tuy nhiên, số lượng mẫu có thể lấy được không phải là vô tận, nó phụ thuộc vào hai yếu tố đó là tốc độ xử lý tín hiệu của máy tính được sử dụng và nằm trong giới hạn của kiểu double.

3.2. Đánh giá

Thực hiện theo trình tự các bước được đề cập tại thuật toán giao tiếp giữa C# và NI đã có thể bắt tay được giữa phần mềm C# và thiết bị NI.

Với thử nghiệm đầu tiên này, tác giả không những lấy được mẫu tín hiệu trên một kênh AI của thiết bị NI9215 mà còn truy vấn được các thông số danh định của thiết bị NI9215 bao gồm: Tên thiết bị, tên các kênh vật lý, dải điện áp đầu vào, mã số thiết bị, loại thiết bị và số serial của thiết bị.

Hạn chế của vấn đề nghiên cứu: Mới chỉ lấy dữ liệu trên một kênh đầu vào của thiết bị NI9215, chưa lấy được tín hiệu trên tất cả các kênh đầu vào cũng như chưa thử nghiệm được trên các thiết bị NI khác.

Hướng phát triển nghiên cứu tiếp theo:

- + Một là xây dựng phần mềm lấy dữ liệu trên tất cả các kênh đầu vào của thiết bị NI9215;
- + Hai là thử nghiệm trên một vài thiết bị NI khác.

TÀI LIỆU THAM KHẢO

- [1] www.ni.com/tutorial/5409/en/.
- [2] <http://zone.ni.com/reference/en-XX/help/370473H-01/mstudiowebhelp/html/5de00fb2/>.
- [3] <https://forums.ni.com/t5/Measurement-Studio-for-NET/How-to-properly-use-AOChannels-CreateFunctionGenerationChannel/td-p/669811>.
- [4] <https://msdn.microsoft.com/en-us/library/dd456769.aspx>.
- [5] [https://msdn.microsoft.com/en-us/library/system.windows.forms.datavisualization.charting.chart\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.windows.forms.datavisualization.charting.chart(v=vs.110).aspx).

Ngày nhận bài: 14/03/2018

Ngày nhận bản sửa: 02/04/2018

Ngày duyệt đăng: 04/04/2018