

### Sách kèm theo CD gồm:

Các ví dụ mẫu LabVIEW

Phần mềm cần thiết học LabVIEW

Video Clip Demo

Chương trình đào tạo LabVIEW tại Việt Nam

2 Hỗ trợ nhanh: <http://labview.hocdelam.org> - [labview.help@gmail.com](mailto:labview.help@gmail.com) | 098 8868 524

## LỜI NÓI ĐẦU

Bộ sách học lập trình LabVIEW từ cơ bản đến nâng cao gồm 7 tập:

- **Tập 1: Lập trình LabVIEW** hướng dẫn lập trình LabVIEW trình độ cơ bản.
- **Tập 2: Cơ điện tử ứng dụng với LabVIEW** hướng dẫn xây dựng các ứng dụng với LabVIEW trong các ngành kỹ thuật.
- **Tập 3: Sổ tay các hàm trong LabVIEW** giải thích tường tận cách sử dụng các hàm trong các module của LabVIEW.

Song song ba tập sách trên là 4 tập sách chuyên đề kỹ thuật chuyên sâu:

- **Tập 4: Kỹ thuật mô phỏng trong với LabVIEW**
- **Tập 5: Đo lường và kiểm tra tự động**
- **Tập 6: Những bài báo khoa học chọn lọc sử dụng LabVIEW** được công bố tại các tạp chí, hội thảo quốc tế phục vụ học viên sau đại học, các sinh viên khá giỏi và kỹ sư muốn nghiên cứu chuyên sâu.
- **Tập 7: 101 Ứng dụng hay với LabVIEW** là tập sách hướng dẫn thực hiện các ứng dụng dùng cho mọi lứa tuổi, chuyên ngành. Từ học sinh lớp 5 đến các giảng viên Đại học các ngành Vật lý, Toán học, Kỹ thuật.

Tất cả các tập sách này được viết cho đối tượng là người **mới bắt đầu tìm hiểu LabVIEW**, phù hợp với sinh viên, giảng viên và kỹ sư có nhu cầu tự học và xây dựng các ứng dụng thuộc các ngành kỹ thuật đo lường, điều khiển tự động, kỹ thuật ôtô, robotics, vật lý, tự động hóa. Tài liệu này có ba ưu điểm quan trọng không có trong các tài liệu khác sẽ giúp bạn:

- **Học nhanh LabVIEW** nhờ cách tiếp cận vấn đề “học để làm” đặc trưng mà Hocdelam Group đã phát triển và áp dụng thành công trong nhiều năm.
- **Học trọng tâm** nhờ vào các kết luận, quy tắc vàng, mẹo vặt (tip) do bản thân tác giả rút ra từ kinh nghiệm lập trình LabVIEW nhiều năm.
- **Học và xây dựng các ứng dụng** dễ dàng nhờ vào gói sản phẩm: *1- Phần mềm LabVIEW, 2-Bộ thí nghiệm đa năng HDL-9000 giá thành thấp tương thích LabVIEW, 3-Giáo trình hướng dẫn và ví dụ mẫu*.

Sách hướng dẫn xây dựng các ứng dụng dựa trên những phần cứng chính hãng như *CompactRIO, USB 6008/6009, PCI7250* và những phần

cứng giá thành thấp như *Hocdelam USB-9001*, *Hocdelam USB 9090*. (Xem sản phẩm tại <http://sanpham.hocdelam.org> hoặc <http://ni.com>).

Để sử dụng phần đầu của tập 1 hướng dẫn lập trình LabVIEW, bạn chỉ cần đặt phần mềm LabVIEW. Phần đầu của cuốn sách tập trung vào các khái niệm về LabVIEW, cấu trúc chung của chương trình LabVIEW, các loại control và indicator, vòng lặp, cấu trúc, mảng, bó, phân tích dữ liệu, mà còn cung cấp kiến thức về giao tiếp máy tính qua cổng COM theo chuẩn RS232 và cổng USB, giao tiếp mạng TCP/IP, mô phỏng và điều khiển mobile robot, trong môi trường 3D, thu thập và xử lý ảnh. Ngoài ra, quy tắc vàng khi lập trình, kỹ thuật xây dựng giao diện người dùng (GUI) cũng sẽ được đề cập chi tiết.

Phần 2 của tài liệu này hướng dẫn bạn phát triển các **ứng dụng thực tế** như điều khiển PID động cơ DC, lập trình Robocon, thời gian thực với LabVIEW dựa trên các card sản xuất bởi National Instruments, Hoa Kỳ hoặc bộ thí nghiệm đa năng HDL-9000, card Hocdelam USB-9001 giá thành thấp.

Để học LabVIEW hiệu quả, người học cần thực hiện theo hướng dẫn của mỗi bài. Thực hành nhiều lần các ví dụ và hoàn thành các bài tập trong sách.

Tác giả chân thành cảm ơn PGS. TS. Đỗ Văn Dũng đã truyền cảm hứng trong suốt quá trình nghiên cứu những năm qua, quý thầy cô tại Đại học Sư phạm Kỹ thuật Tp.HCM. Đặc biệt là sự giúp đỡ quan trọng của National Instruments về tinh thần và vật chất cho việc nghiên cứu cùng những khích lệ từ ông Chandran Nair–Giám đốc NI ASEAN, Ông Suntisuk–Giám đốc quản lý Thái Lan-Việt Nam, Ông Phạm Quốc Hùng và Ông Nguyễn Hồ Nam tại NI Việt Nam. Chân thành cảm ơn PGS. TS. Từ Diệp Công Thành, ĐH Bách Khoa Tp.HCM, TS. Nguyễn Đức Hùng từ ĐH Tamaia, Úc, TS. Chung Tấn Lâm đã đọc bản thảo và góp ý cho tập sách.

Hiểu biết của tác giả còn hạn chế nên không thể tránh khỏi các sai sót, thư liên hệ và góp ý vui lòng gửi về: *Nguyễn Bá Hải, Phòng thí nghiệm Cơ điện tử Ô tô – DH, Sư phạm Kỹ thuật Tp.HCM, Số 01-Võ Văn Ngân, Quận Thủ Đức, Tp.HCM; Email: bahai@hcmute.edu.vn. Hotline: 0937-159-700.*

### Tác giả

# BÀI 1

## GIỚI THIỆU LabVIEW

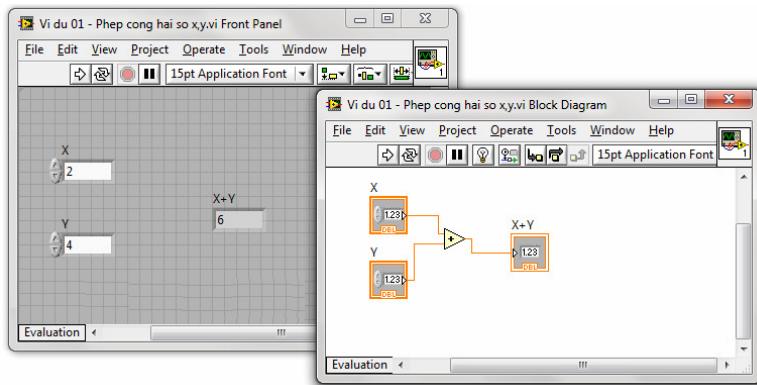
Học xong bài này, bạn sẽ nắm được:

- Khái niệm về LabVIEW và một số thuật ngữ
- Phân biệt được LabVIEW với các ngôn ngữ khác

### 1.1. LabVIEW là gì?

LabVIEW (viết tắt của Laboratory Virtual Instrumentation Engineering Workbench) là môi trường ngôn ngữ đồ họa hiệu quả trong việc giao tiếp đa kênh giữa con người, thuật toán và các thiết bị.

Gọi LabVIEW là ngôn ngữ đồ họa hiệu quả vì về cách thức lập trình, LabVIEW khác với các ngôn ngữ C (hay Python, Basic, vv.) ở điểm thay vì sử dụng các **tùy chỉnh** (tùy chỉnh) cố định thì LabVIEW sử dụng các **khối hình ảnh** sinh động và các **dây nối** để tạo ra các **lệnh** và các **hàm** như trong hình 1.1. Cũng chính vì sự khác biệt này mà LabVIEW đã giúp cho việc lập trình trở nên **đơn giản** hơn bao giờ hết, đặc biệt, LabVIEW rất phù hợp đối với kỹ sư, nhà khoa học, hay giảng viên. Chính sự đơn giản, dễ học, dễ nhớ đã giúp cho LabVIEW trở thành một trong những công cụ phổ biến trong các ứng dụng **thu thập dữ liệu** từ các cảm biến, **phát triển các thuật toán**, và **điều khiển** thiết bị tại các phòng thí nghiệm trên thế giới.



**Hình 1.1: Mã nguồn viết bằng LabVIEW**

Về ý nghĩa kỹ thuật, LabVIEW cũng được dùng để lập trình ra các chương trình (source code: **mã nguồn**) trên máy tính tương tự các **ngôn ngữ lập trình dựa trên chữ** (text-based language) như C, Python, Java, Basic, vv.

Đồng thời, LabVIEW hỗ trợ các kỹ sư, nhà khoa học và sinh viên, vv. xây dựng (thực thi) các thuật toán một cách nhanh, gọn, sáng tạo, và dễ hiểu nhờ các khối hình ảnh có tính gợi nhớ và cách thức hoạt động theo kiểu dòng dữ liệu (data flow) lần lượt từ trái qua phải. Các thuật toán này sau đó được áp dụng lên các mạch điện và cơ cấu chấp hành thực nhờ vào việc kết nối hệ thống thật với LabVIEW thông qua nhiều chuẩn giao tiếp như chuẩn giao tiếp RS232 (giao tiếp qua cổng COM), chuẩn USB, chuẩn giao tiếp mạng TCP/IP, UDP, chuẩn GPIB, vv. Vì vậy LabVIEW là một ngôn ngữ giao tiếp đa kênh.

LabVIEW hỗ trợ hầu hết các hệ điều hành (Windows (2000, XP, Vista, Windows7), Linux, MacOS, Window Mobile, Window Embedded. Hiện tại, LabVIEW 2010 là phiên bản mới nhất. Một số phiên bản cũ của LabVIEW bao gồm 2009, 8.6, 8.5, 7.1, 6i. Nhìn chung hai phiên bản kề nhau 2010 và 2009 không có sự khác nhau nhiều. Tuy nhiên có sự khác biệt đáng kể giữa các bản LabVIEW 7.1, LabVIEW 8.5 và LabVIEW 2009. Bạn có thể chọn một trong ba phiên bản sau cùng để dùng cho việc học lập trình, trong cuốn sách này, chúng tôi sử dụng LabVIEW 2009.

### **Ban có biết?**

LabVIEW được tiến sĩ James Truchard, Đại học Texas, Hoa Kỳ sáng tạo ra vào năm 1986, và đã trở thành một công cụ không thể thiếu trong các ngành kỹ thuật. Sản phẩm đầu tiên bắt đầu trong nhà để xe của Chủ tịch, CEO và đồng sáng lập NI – James Truchard ở Austin, Texas, Hoa Kỳ.

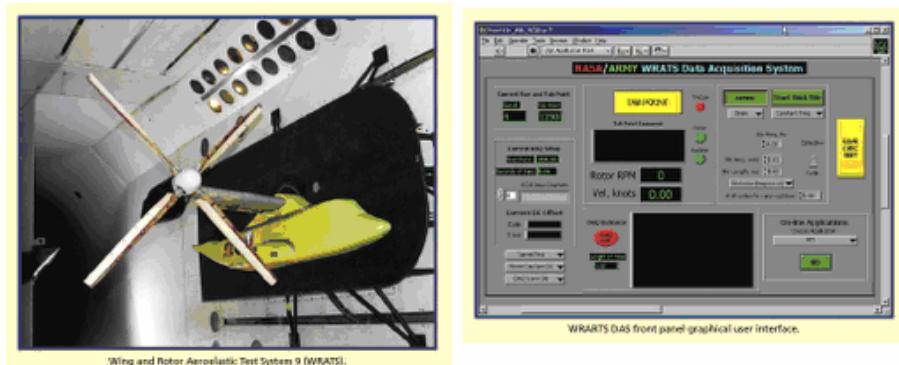
## **1.2. Các ứng dụng của LabVIEW**

LabVIEW được sử dụng trong các lĩnh vực đo lường, tự động hóa, cơ điện tử, robotics, vật lý, toán học, sinh học, vật liệu, ôtô, vv. Nhìn chung:

- LabVIEW giúp kỹ sư kết nối bất kỳ cảm biến, và bất kỳ cơ cấu chấp hành nào với máy tính.
- LabVIEW có thể được sử dụng để xử lý các kiểu dữ liệu như tín hiệu tương tự (analog), tín hiệu số (digital) hình ảnh (vision), âm thanh (audio), vv.
- LabVIEW hỗ trợ các giao thức giao tiếp khác nhau như RS232, RS485, TCP / IP, PCI, PXI, và như vậy.
- Bạn cũng có thể tạo ra các thực thi độc lập và các thư viện chia sẻ (ví dụ thư viện liên kết động DLL), bởi vì LabVIEW là một trình biên dịch 32-bit.

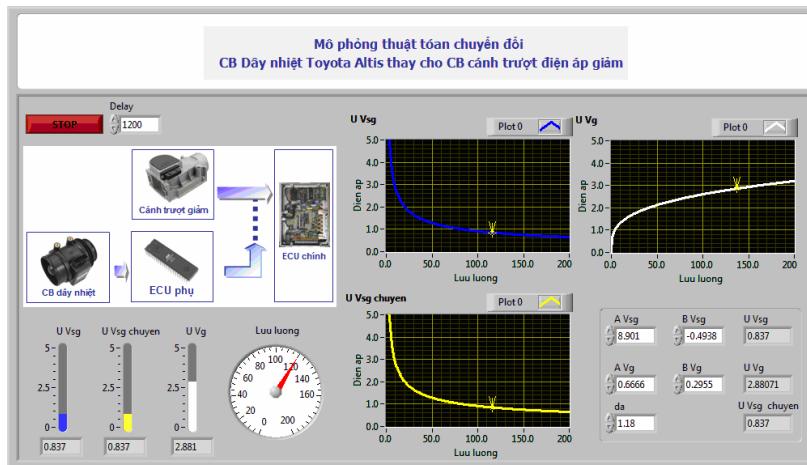
LabVIEW đã trở nên phổ biến ở các phòng thí nghiệm ở Nhật, Hàn, Mỹ, Anh, Đức, vv. Phần này trình bày một số ứng dụng của LabVIEW tiêu biểu và các dự án tại các phòng thí nghiệm trong và ngoài nước mà tác giả cuốn sách này đã tham gia thực hiện dự án.

**Ứng dụng đo lường**, trong hình 1.2 là giao diện thu thập dữ liệu các thông tin cần thiết của tàu vũ trụ cỡ nhỏ tại cơ quan hàng không và vũ trụ NASA, Hoa Kỳ.



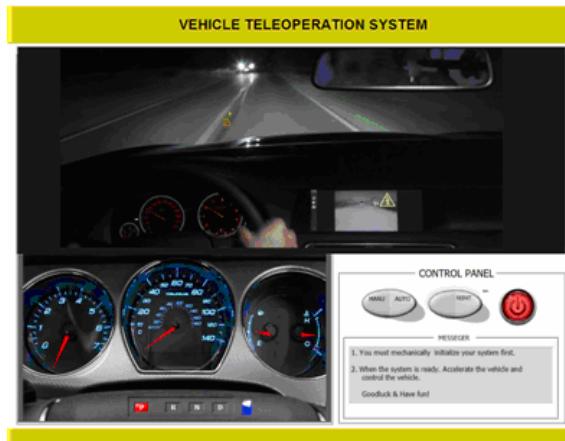
**Hình 1.2: Thu thập dữ liệu tại Cơ quan hàng không và vũ trụ - NASA**

Ứng dụng hình 1.3 này giới thiệu áp dụng của việc sử dụng LabVIEW và card Hocdelam USB 9001 hoặc NI USB 6008 để thực hiện đo tín hiệu, vẽ biểu đồ đặc tuyến các cảm biến trong ôtô và thực nghiệm thuật toán chuyên đổi cảm biến nhằm hạ giá thành sửa chữa xe ôtô. Ứng dụng này được thực hiện tại đại học Sư phạm Kỹ thuật Tp.HCM năm 2008.



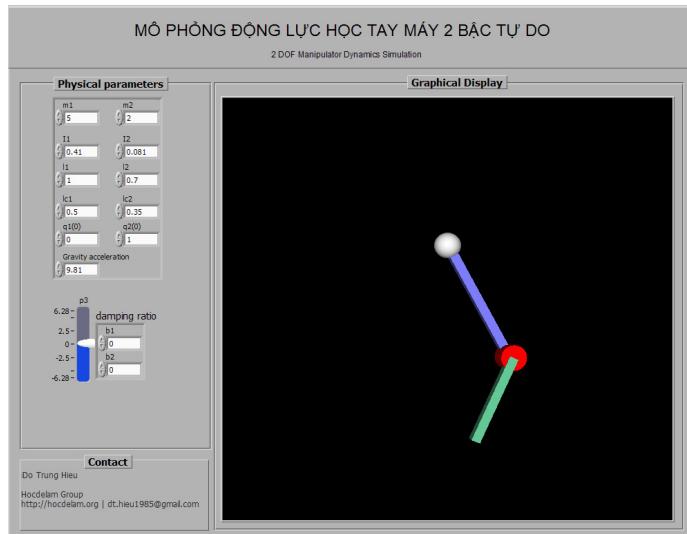
**Hình 1.3: Thu thập dữ liệu từ cảm biến đo gió trong ôtô và thí nghiệm thuật toán chuyên đổi cảm biến**

**Điều khiển xe ô tô từ xa**, hình 1.4 là giao diện điều khiển ôtô bảy chỗ (xe Captival) từ xa được thực hiện bởi thành viên Hocdelam Group tại phòng thí nghiệm Biorobotics, Hàn Quốc. Giao diện này hoàn toàn được xây dựng trong môi trường lập trình LabVIEW có khả năng hiển thị các thông số và tín hiệu thực như: vận tốc xe, mức xăng, vị trí tay số của xe, video truyền từ xe qua mạng không dây, âm thanh từ động cơ tỷ lệ thuận với vị trí bướm ga cũng được giả lập làm cho việc điều khiển xe từ xa giống với việc lái xe trực tiếp nhằm nâng cao chất lượng điều khiển xe.



**Hình 1.4: Giao diện lái ôtô từ xa**

**Mô phỏng 3D**, hình 1.5 mô tả ứng dụng mô phỏng một cánh tay robot đơn giản do Thạc sĩ Đỗ Trung Hiếu thực hiện.



**Hình 1.5: Điều khiển tay Robot**

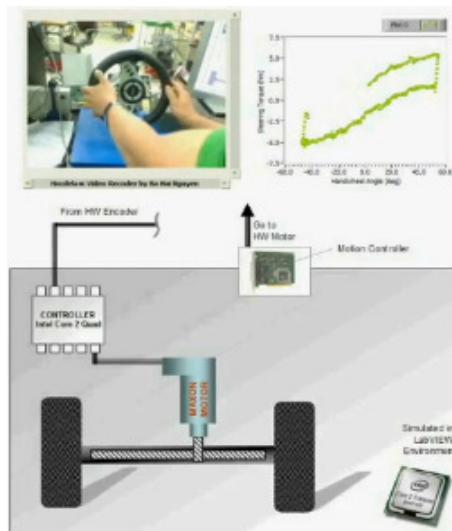
**8 Hỗ trợ nhanh:** <http://labview.hocdelam.org> - [labview.help@gmail.com](mailto:labview.help@gmail.com) | 098 8868 524

**Điều khiển phương tiện không người lái**, hình 1.6 mô tả ứng dụng LabVIEW điều khiển robot không người lái nhằm dò tìm và khám phá dưới nước của tập đoàn Nexans.



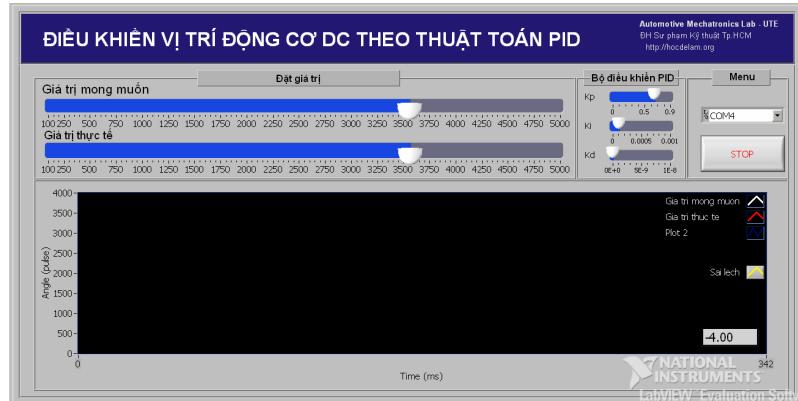
**Hình 1.6: Robot dưới nước (Spider) được phát triển dựa trên LabVIEW của công ty Nexans**

**Thu thập hình ảnh và mô phỏng động lực học**, hình 1.7 trình bày ứng dụng mô phỏng hệ thống lái không trực lái trong ô tô. Đồng thời, hình ảnh từ webcam (webcam thông thường gắn qua cổng USB) được thu thập và đưa lên giao diện người dùng (Graphical User Interface – GUI). Xem chi tiết đề tài này bằng cách search google cụm từ sau: “Steer-By-Wire Systems with Realistic Steering Feel”. Xem video clip *Steer-by-wire.avi* và file *Steer-By-Wire Systems with Realistic Steering Feel.pdf* trong CD kèm theo sách này.



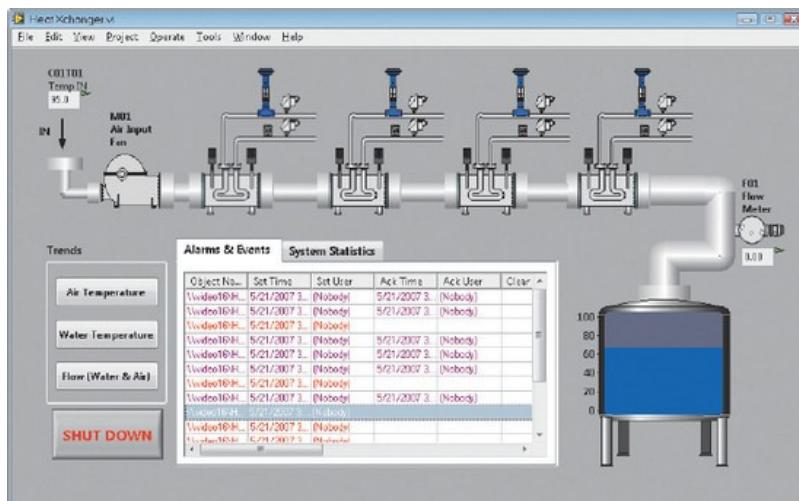
**Hình 1.7 Hệ thống lái không trực lái tại phòng thí nghiệm Biorobotics, KUT, Hàn Quốc**

**Thuật toán điều khiển tự động vị trí động cơ DC theo thuật toán PID và giao diện trực quan** trong hình 1.8 được Hocdelam Group viết dựa trên cơ sở phần mềm LabVIEW. Với giao diện này, người điều khiển sẽ dễ dàng quan sát giá trị vị trí mong muốn, và vị trí thực tế của động cơ, đồng thời, so sánh và đánh giá được tốc độ đáp ứng, độ ổn định của động cơ DC trong quá trình vận hành.



**Hình 1.8 Điều khiển động cơ DC theo thuật toán PID**

**Đo lường, giám sát và điều khiển hệ thống công nghiệp (SCADA)**, hình 1.9 trình bày ứng dụng của LabVIEW trong hệ thống điều khiển và giám sát dùng trong công nghiệp.



**Hình 1.9: Hệ thống đo lường, giám sát và điều khiển trong công nghiệp**

Tới đây, bạn đã hình dung phần nào những ích lợi và sức mạnh của ngôn ngữ lập trình đồ họa LabVIEW. Vậy, chúng ta hãy cùng bắt tay vào học và sử dụng LabVIEW nhé.

**10 Hỗ trợ nhanh:** <http://labview.hocdelam.org> - [labview.help@gmail.com](mailto:labview.help@gmail.com) | 098 8868 524

### 1.3 Download và cài đặt LabVIEW

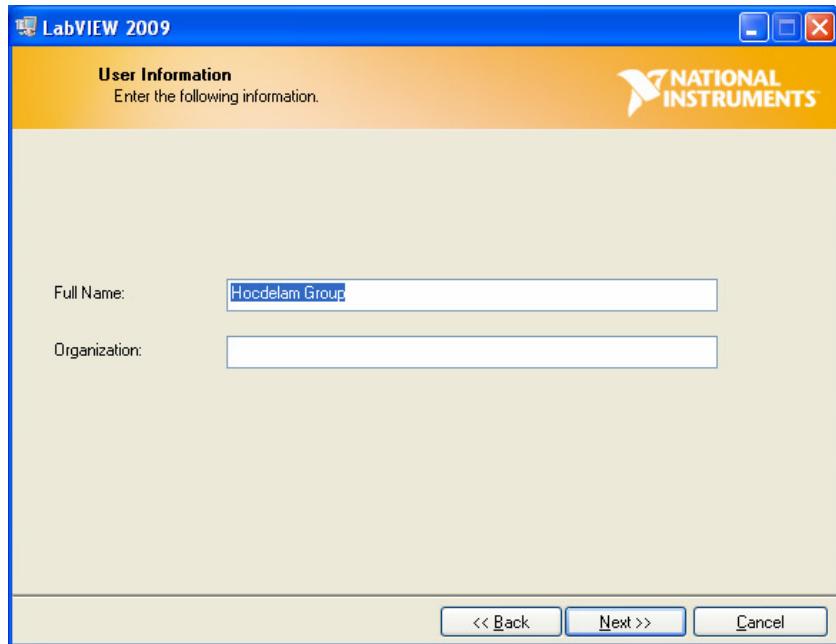
Để học và làm việc với LabVIEW, ta chỉ cần cài bản LabVIEW dùng thử (evaluation version), so với phiên bản có bản quyền (như bản professional version) thì bản dùng thử có giới hạn 30 ngày (bạn có thể cài lại sau khi hết hạn để dùng tiếp lâu dài) và số thư viện hàm (function library) sẵn có bị giới hạn. Tuy nhiên, thật may mắn là ta hoàn toàn có thể tự xây dựng các thư viện này từ những hàm cơ bản có sẵn trong bản dùng thử. Hãy yên tâm học và sử dụng LabVIEW nếu bạn chưa có đủ kinh phí để trang bị cho mình LabVIEW có bản quyền nhé.

Bạn tải phần mềm LabVIEW 2009 tại <http://labview.hocdelam.org> hoặc mua bộ đĩa CD phần mềm LabVIEW và giáo trình lập trình LabVIEW qua Email: supports@hocdelam.org. Nhìn chung, các phiên bản LabVIEW 8.5, 8.6 và 2010 cũng được cài đặt tương tự. Về cách thức cài đặt, ta cài LabVIEW như các phần mềm thông thường như sau:

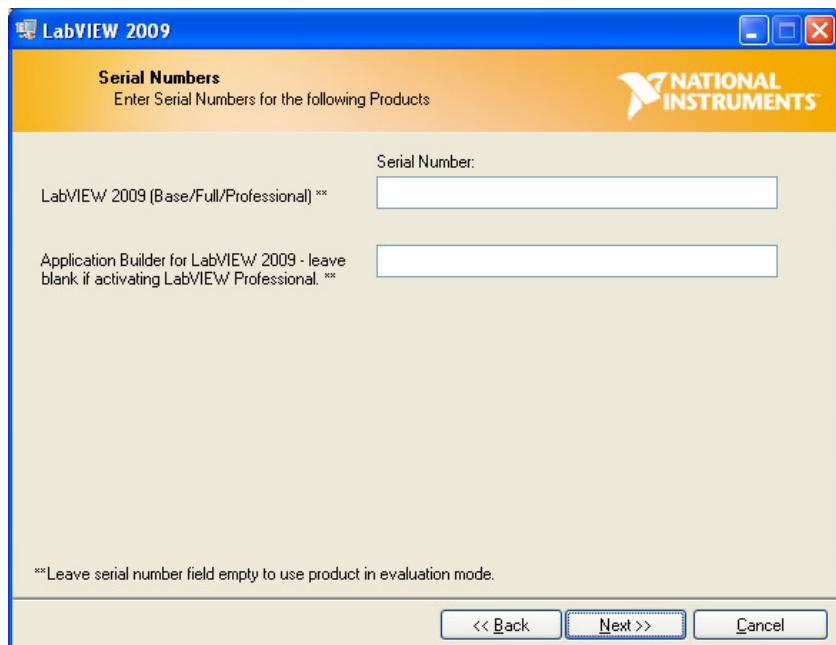
Giải nén file **LabVIEW 2009.rar**, chạy file Setup.exe trong thư mục **LabVIEW 2009**. Sau đó bạn cài đặt phần mềm theo các bước lần lượt ở các hình sau (các hình này không được đánh số hình).



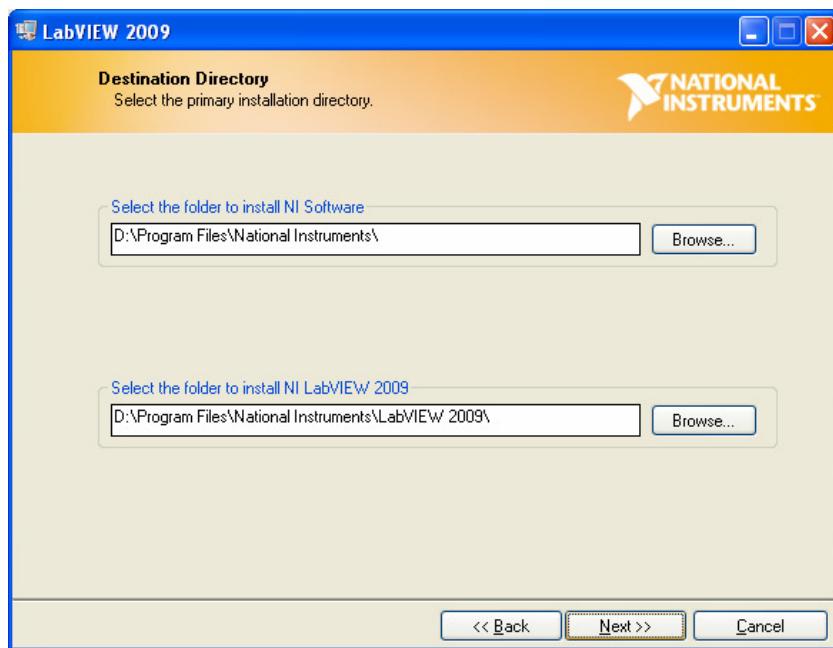
Chọn Next



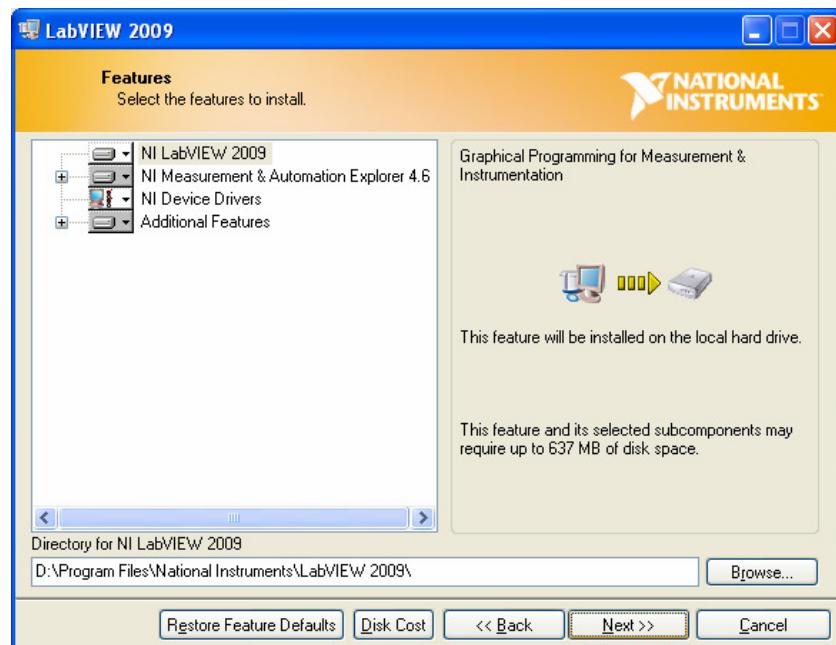
Chọn Next



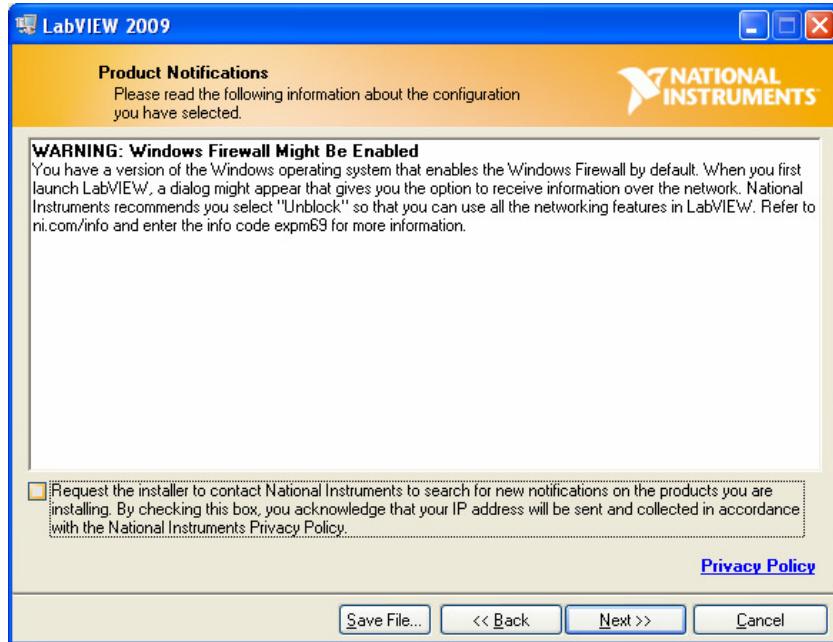
Chọn Next



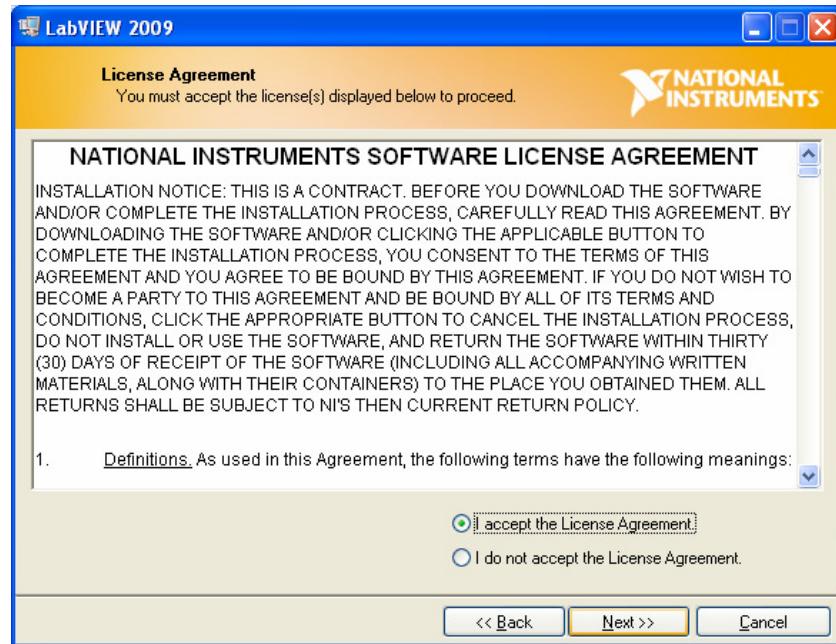
Chọn Next



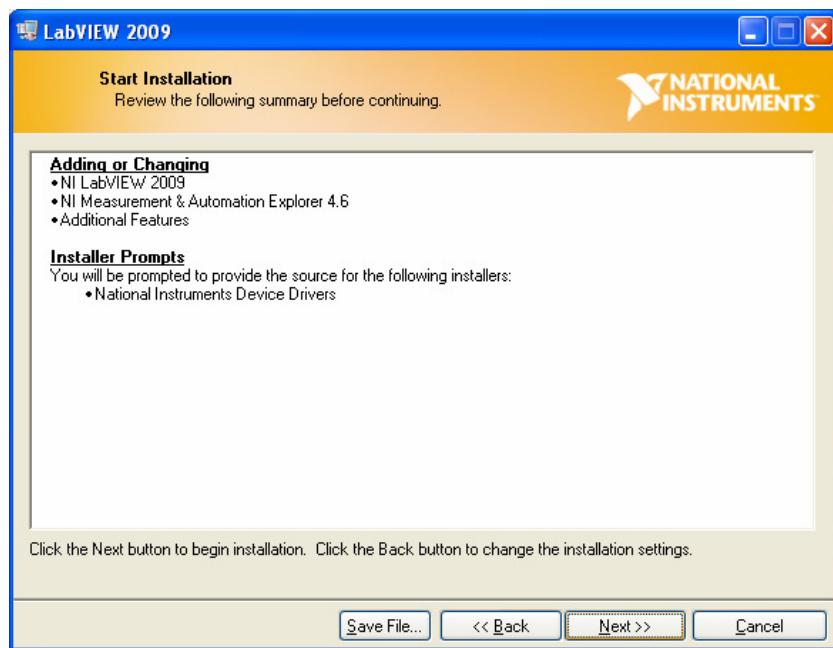
Chọn Next



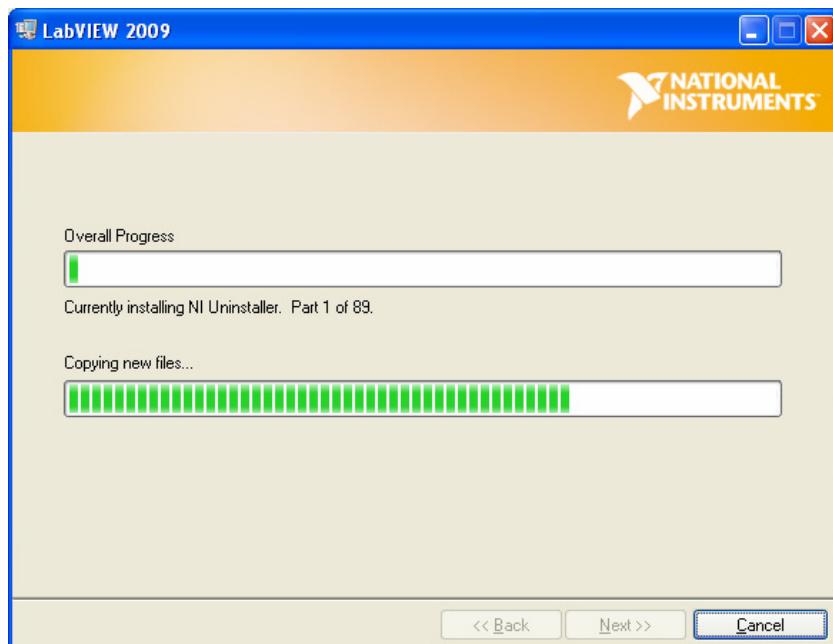
Chọn Next



Chọn I accept và chọn Next

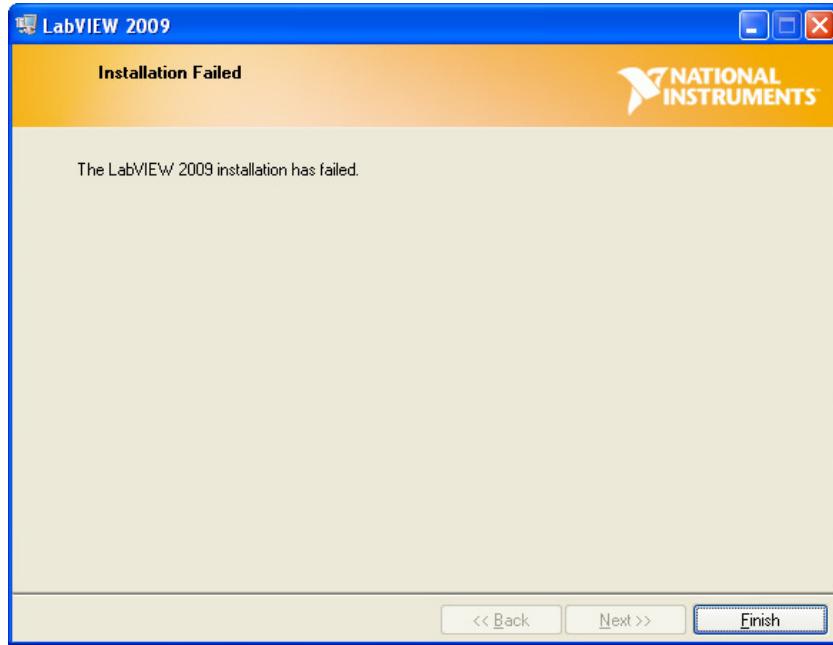


Chọn Next



Chọn **Later** nếu được hỏi cài Driver và chọn Finish để hoàn thành việc cài đặt.

Hỗ trợ nhanh: <http://labview.hocdelam.org> - [labview.help@gmail.com](mailto:labview.help@gmail.com) 0988868524



Nếu gặp khó khăn trong việc tải phần mềm và cài đặt LabVIEW, gửi email về [supports@hocdelam.org](mailto:supports@hocdelam.org), hỗ trợ nhanh qua đường dây nóng: 0937159700.

#### 1.4. Phương pháp để học LabVIEW hiệu quả

Để học LabVIEW nhanh và hiệu quả, bạn nên:

- **Tự mình** lập trình lại các ví dụ và bài tập trong sách sau khi đã cài đặt phần mềm LabVIEW.
- Trong quá trình học luôn thực hiện "**3 quy tắc vàng**" khi lập trình. Ba quy tắc này được trình bày tại bài 2.
- Học nhanh LabVIEW cơ bản qua các video clip (Download miễn phí tại: <http://labview.hocdelam.org>)

#### **Bạn có biết?**

Đối với việc học lập trình thì con đường **duy nhất, nhanh nhất** để giỏi là bạn **phải thực hành**.

Tài liệu tiếng Việt khác có thể tìm tại:

- labview.hocdelam.org
- vagam.hocdelam.org
- ino.com.vn
- google.com.vn (bạn cần search các từ khóa bằng tiếng Việt như “tài liệu labview”, “giáo trình labview”,...).

Tài liệu tiếng Anh có thể tìm tại:

- ni.com
- lava.org
- google.co.uk (bạn cần search các từ khóa bằng tiếng Anh như “LabVIEW tutorial”, “labview book”, “labview demonstration”, “labview application”,vv).
- Một số cuốn sách tham khảo gồm: LabVIEW for everyone, LabVIEW advanced programing technique, vv.

**Đăng ký các khóa học** thông qua các trình đào tạo LabVIEW tại Việt Nam được cung cấp bởi các trung tâm, trường học sau:

- Hocdelam Group cung cấp các chương trình đào tạo theo nhu cầu của doanh nghiệp, nhà máy, trường học và cá nhân trong cả nước (Đăng ký nhanh qua điện thoại: 0937 159 700).
- Đối với người bận rộn, bạn có thể đăng ký học LabVIEW trực tuyến (online) tại <http://labview.hocdelam.org>.
- Trường Đại học Sư Phạm Kỹ thuật thành phố Hồ Chí Minh tổ chức các chương trình đào tạo các khóa học liên quan LabVIEW cho học viên sau đại học, sinh viên hệ đại học chính quy ngành Cơ khí Động lực.
- Các khóa lập trình LabVIEW cơ bản tại Trường Kinh Tế Kỹ thuật Phú Lâm, Q6, Tp.HCM.
- Phòng đào tạo công ty VTK, Hà Nội.
- Học tại Công ty TNHH thương mại dịch vụ kỹ thuật LINH PHÚ, Q. Gò Vấp, Tp.HCM.
- Nếu có vốn tiếng Anh tốt, bạn có thể đăng ký học trực tuyến tại <http://ni.com>.

## Xây dựng ứng dụng thực tế

Để học nhanh LabVIEW và xây dựng được các ứng dụng thực tế bạn có thể trang bị phần cứng như card giao tiếp máy tính dùng để thực tập thu thập tín hiệu cảm biến và điều khiển tự động động cơ, lò nhiệt,... với LabVIEW. Địa chỉ cung cấp phần cứng trực tuyến hoặc tại Tp.HCM.

- <http://ni.com> (tiếng Anh)
- <http://sanpham.hocdelam.org> (tiếng Việt)
- <http://ino.com.vn>
- Công ty Tân Minh Giang, P.10 - Q. Gò Vấp - Tp.HCM.
- Công ty TNHH thương mại dịch vụ kỹ thuật LINH PHÚ, Q. Gò Vấp, Tp.HCM.

### 1.5. Bài tập

**Câu 1:** Hãy nêu khái niệm về LabVIEW

**Câu 2:** Source code là gì?

**Câu 3:** Ưu điểm của ngôn ngữ lập trình đồ họa LabVIEW so với các ngôn ngữ khác lập trình khác là gì?

**Câu 4:** Nêu một số ứng dụng của LabVIEW trong giáo dục và công nghiệp

### Tham khảo

[1] <http://labview.hocdelam.org>

[2] <http://ni.com>

[3] Nguyễn Bá Hải “*Thật đơn giản: Điều khiển động cơ DC theo thuật toán PID*”, 2010.

[4] [wikipedia.org](http://wikipedia.org)

[5] Google.com [Nguồn một số hình ảnh trong bài viết]

## BÀI 2

# CÁC PHÉP TOÁN TRONG LabVIEW

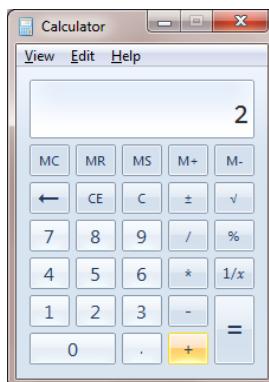
Học xong bài này, bạn sẽ nắm được:

- *Cấu trúc chung của một chương trình LabVIEW*
- *Ba quy tắc vàng khi lập trình LabVIEW*
- *Các phép toán trong LabVIEW*
- *Lưu file, tìm ví dụ, các công cụ trợ giúp*

### 2.1. Chương trình đầu tiên: Phép cộng hai số x, y

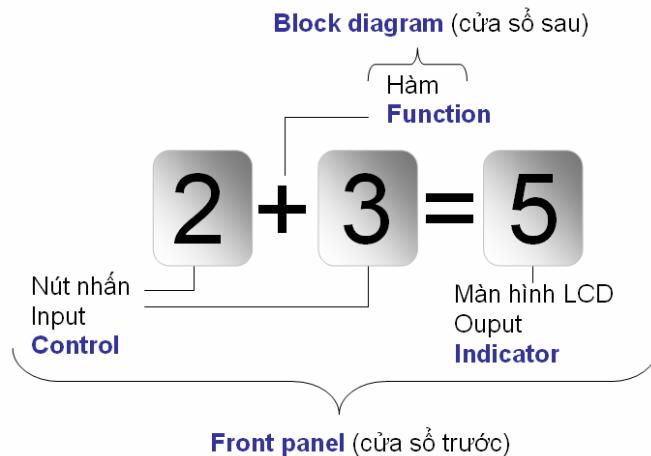
Để đơn giản, hãy xem LabVIEW như phần mềm soạn thảo văn bản Microsoft Office Word. Và việc soạn thảo văn bản trong Word cũng tương tự việc tạo ra Source Code trong LabVIEW. Hãy cùng làm thực hiện chương đầu tiên – **Phép cộng hai số x,y** như sau:

Bình thường, để cộng hai số x,y bằng **máy tính bỏ túi** (pocket calculator) như Hình 2.1. Ví dụ  $x=2$ ,  $y=3$ , thì ta phải nhấn số 2, nhấn dấu +, và nhấn số 3, sau đó nhấn dấu =, và kết quả sẽ hiển thị ra một màn hình LCD. Như vậy ta sẽ có hai **nút nhấn** (2 và 3) để **nhập giá trị**, một **LCD** để **hiển thị giá trị** ra của phép cộng, và **hàm cộng** (+).



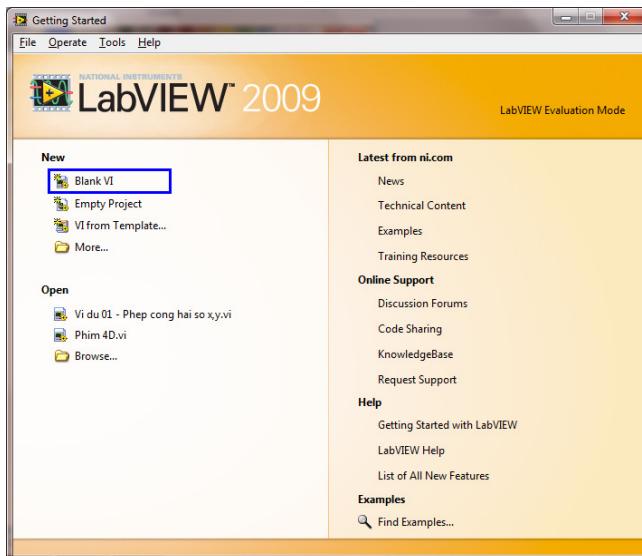
**Hình 2.1: Máy tính bỏ túi**

Trong LabVIEW, các nút nhấn được gọi là các **Control**, các LCD gọi là các **Indicator**. Phép cộng gọi là **hàm (Function)**. Các control và indicator sẽ nằm ở cửa sổ trước (Front panel), và hàm nằm ở cửa sổ sau (Block diagram), còn gọi là cửa sổ chứa các sơ đồ khối. Xem thêm Hình 2.2.

**Hình 2.2: Phân tích ví dụ tổng của x và y**

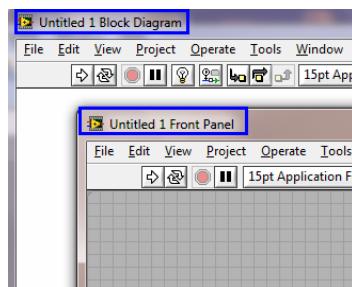
Bây giờ ta thực hiện trong LabVIEW. Khởi động LabVIEW bằng cách vào: All programs > National instruments > LabVIEW 2009 > LabVIEW. Hộp thoại xuất hiện và bạn chọn vào

Cửa sổ giao diện LabVIEW xuất hiện như Hình 2.3.

**Hình 2.3: Cửa sổ giao diện LabVIEW khi mới khởi động**

**Tạo file mới:** Tương tự trình tạo văn bản Microsoft Office Word, để tạo file mới ta có 3 cách: chọn **Blank VI** (VI trắng) (Hoặc chọn: **File> New** hoặc phím tắt **Ctrl+N**), ta thấy xuất hiện **hai cửa sổ** - màu xám (**Front**

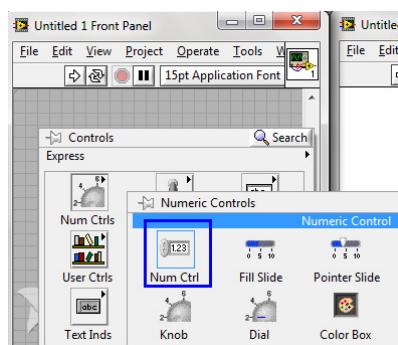
panel) và màu trắng (Block diagram) như trong Hình 2.4. Nhấn Ctrl+T để canh đều hai cửa sổ Front panel (FP) và Block diagram (BD).



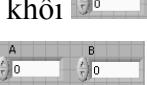
**Hình 2.4: Front panel và Block diagram**

➤ **Tạo control** để nhập số 2 và 3 như sau:

- Click chuột phải (Right click) lên front panel (FP) lấy khối **Numeric Control** bằng cách click chọn khối như [hình 2.5].

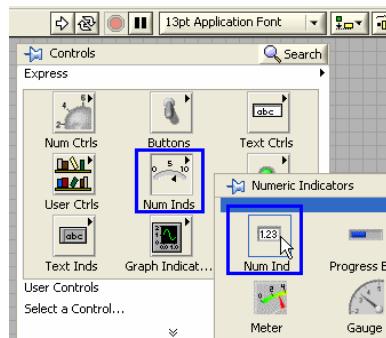


**Hình 2.5: Lấy numeric control trên FP**

- Rê chuột ra vùng trống của FP , chọn vị trí cần đặt **Control** này.
- Nhấn chuột trái (Left Click) chuột phải để  **thả Control** xuống FP 
- Nhập chữ A vào mục tô đen để thay đổi tên khối 
- Tương tự ta tạo khối B để có hai khối A,B 

➤ **Tạo indicator** để hiển thị kết quả của tổng như sau:

- Right click lên FP, vào **Num Inds** (Numeric Indicator) lấy khối **Numeric Indicator** bằng cách click chọn khối như Hình 2.6.

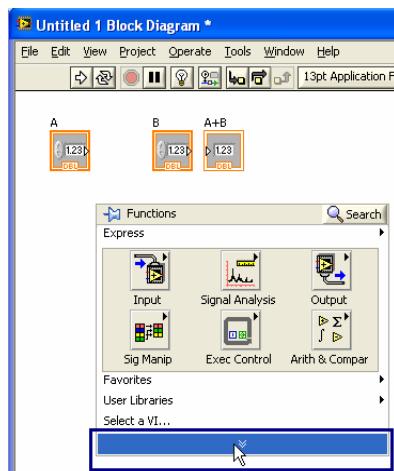


**Hình 2.6: Lấy numeric indicator trên FP**

- Thả xuống vùng trống của FP.
- Sửa tên khối này thành A+B. Ta được

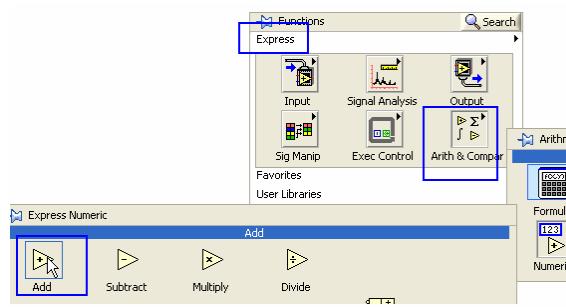
➤ **Lấy hàm cộng (+)** để thực hiện phép cộng hai số 2 và 3 bằng cách:

- Right Click lên Block Diagram (BD). Chọn vào **dấu mũi hai mũi tên** để xô của số xuống.



**Hình 2.7: Lấy hàm cộng tại BD**

- Chọn vào thư viện , chọn vào , chọn hàm cộng .
- Hoặc chọn Express> , chọn hàm cộng như Hình 2.8.

**Hình 2.8: Lấy hàm cộng tại BD**

- Đặt ra vùng trống trong BD (không thể đặt hàm tại FP). Ta được
- 
- Di chuột ra các vị trí đầu (hình của các khối Control và Indicator), sẽ thấy hình cuộn chỉ hiện ra,
- 
- **Left Click** khi cuộn chỉ hiện ra, sau đó  **rê chuột** tới đầu nối của hàm cộng rồi nhá Left Click để kết nối.
- 
- **Tương tự**, ta **nối** B vào hàm cộng. Và nối đầu nhọn bên phải của tam giác hàm cộng vào Indicator A+B
- 
- **Ta được** (Xem ví dụ này trong CD kèm theo tại đường dẫn: **CD/Bài 2/Ví dụ2-1 - Phep cong hai so A,B.vi**)

### ➤ Chạy chương trình

- **Nhập** giá trị 2 vào ô Control A, 3 vào ô Control B
- Nhấn nút **Run** có hình tam giác bên FP hoặc bên BD đều được để chạy chương trình.
- Indicator A+B xuất ra giá trị (trả về giá trị) 5:

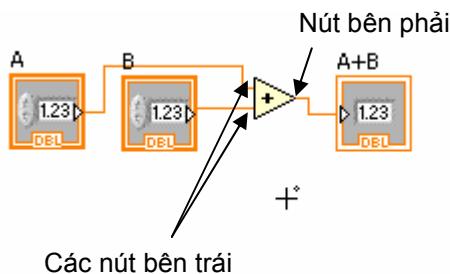
Như vậy, bạn đã vừa **hoàn thành chương trình đầu tiên trong LabVIEW**. Xin chúc mừng bạn!

**Lưu ý:** Thư viện hàm Express là tập con của thư viện Programming. Nói cách khác là những hàm đặt trong thư viện Express đều có trong thư viện Programming. Thư viện Express chỉ chứa những hàm thường dùng.

## 2.2. Phân tích cấu trúc một bài trình LabVIEW

Qua ví dụ mục 2.1 ta thấy, một chương trình LabVIEW (dù đơn giản nhất hay phức tạp nhất cũng được tạo nên một cách rất đơn giản từ 4 thành phần cơ bản sau: các khối **Control** đóng vai trò input (hay giá trị nhập vào), các **Indicator** đóng vai trò Ouput (giá trị hiển thị ra, kết quả), các **hàm** và các **đường dây nối** (wire) các khối và hàm lại. Nếu bạn đã thực hiện được chương trình phép cộng trên thì mọi chương trình phức tạp sau này bạn cũng sẽ **hoàn toàn thực hiện được** khi theo dõi kỹ các tập sách này.

Các Control thì luôn nối vào các nút bên trái của hàm. Các Indicator thì luôn nối vào các nút bên trái của hàm như Hình 2.9. Đặc điểm này là một lưu ý vô cùng quan trọng khi sử dụng qui tắc vàng **Change to** sau này.



**Hình 2.9: Các nút bên trái và bên phải của hàm cộng**

Ta cũng nhận thấy, các Control thì có các nút mũi tên lên-xuống để tăng giảm giá trị (hoặc nhập trực tiếp vào ô Control). Các Indicator thì không thể nhập giá trị vào. Control thì có màu trắng, còn Indicator thì có màu xám.

Một Control có thể đổi thành Indicator (hoặc Constant) bằng cách nhấp Right Click lên Control cần đổi, chọn **Change to Indicator**. Hoặc ngược lại, một Indicator có thể đổi thành Control bằng phép **Change to Control** được đề cập chi tiết ở mục 2.3. Constant chính là một Control có giá trị không đổi suốt thời gian chạy (Run) chương trình.

Các Control và Indicator đều có ảnh của chính mình bên BD, ảnh này tự động tạo ra khi ta lấy các Control và Indicator bên FP.

### Bạn có biết?

Hướng dẫn thực hiện phép cộng này còn được trình bày dưới dạng Video Clip có thể download được miễn phí từ <http://labview.hocdelam.org>.

### 2.3. Ba quy tắc vàng

Ba quy tắc vàng là những **hoạt động phải thực hiện liên tục** trong suốt quá trình lập trình LabVIEW. Ba quy tắc này là:

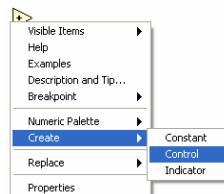
**Quy tắc 1: Right Click> Create/Replace/Change to** để lấy các khôi, tạo khôi, thay thế khôi và đổi chức năng khôi.

**Quy tắc 2: Ctrl + H** để xem sơ đồ chân của hàm và đọc hướng dẫn tóm tắt của một hàm bất kỳ (context help).

**Quy tắc 3: Search** để tìm kiếm các hàm, đối tượng

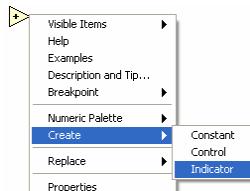
Nếu biết kết hợp các quy tắc này, việc lập trình LabVIEW của bạn sẽ đạt hiệu quả cao như: rút ngắn thời gian, giảm bớt các công đoạn vào thư viện lấy Control, Indicator, vv. Hãy xem cách áp dụng quy tắc thứ nhất (**Quy tắc 1**) trong ví dụ tính tổng hai số A+B trên như sau:

- Lấy hàm cộng từ BD như Hình 2.8
- Rê chuột vào 2 nút phía trái hàm (Hình 2.9) cho tới khi hình cuộn chỉ hiện ra . **Right Click**, chọn **Create**, chọn **Control**.



**Hình 2.10: Cách tạo Control theo quy tắc vàng 1**

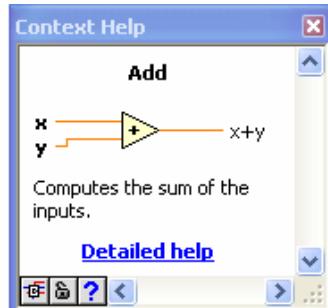
- Tương tự, ta tạo Control thứ 2.
- Rê chuột lên nút bên trái còn lại của hàm. **Right Click**, Chọn **Create**, Chọn **Indicator** như Hình 2.11.



**Hình 2.11: Cách tạo Indicator theo quy tắc vàng 1**

Như vậy, khi áp dụng quy tắc vàng 1 giúp bỏ qua việc lấy các Control và Indicator từ FP và giúp việc lập trình trở nên nhanh hơn, hạn chế số lần Click chuột để tìm các khôi cần thiết trong lập trình.

Về quy tắc Ctrl+H, hãy lấy hàm cộng ra tại Block Diagram, và nhấn Ctrl+H, sau đó đưa chuột lên hàm cộng. Ta sẽ thấy sơ đồ chân và lời giải thích về hàm cộng, nếu nhấn tiếp Ctrl+H lần nữa thì sơ đồ chân sẽ ẩn đi.



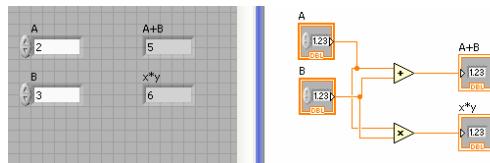
**Hình 2.12: Context help hiển thị sơ đồ chân của Control ( $x,y$ ) và Indicator ( $x+y$ )**

Về quy tắc Search, bạn chọn Right Click vào FP hoặc BD, sau đó chọn chữ Search ở góc phải phía trên của hộp thoại thư viện hàm, sau đó gõ **tên tiếng anh** của các **control** hoặc **indicator** (đối với FP) **hoặc tên hàm** (đối với BD) thì sẽ dễ dàng tìm ra các hàm. Cách này giúp người lập trình tiết kiệm thời gian trong việc tìm kiếm các hàm. Hãy yên tâm nếu bạn không giỏi tiếng Anh vì bạn có thể dễ dàng tra cứu tên của control, indicator hoặc tên hàm bằng các từ điển Việt-Anh hoặc công cụ **Google Translate** trực tuyến.

Ngoài ra, bạn cũng cần lưu ý tới các ví dụ có sẵn của LabVIEW tìm thấy trong menu **Help> Find examples** để tìm các ví dụ (Example) có sẵn trong LabVIEW. Các ví dụ này giúp bạn tiết kiệm đáng kể thời gian để tìm hiểu và làm chủ LabVIEW.

#### 2.4. Các phép toán thông thường

Từ bài toán trên, ta có thể viết chương trình tính tích và tổng hai số A,B như Hình 2.13. Để thực hiện, bạn chỉ cần lấy thêm hàm nhân (cùng vị trí với hàm cộng trong BD), sau đó **đưa chuột** lên dây nối từ A đến hàm cộng , **Click chuột trái** lên dây này, và nối vào đầu vào của hàm nhân.

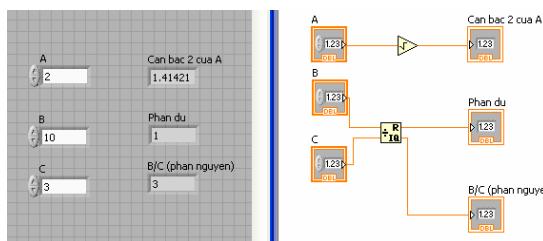


**Hình 2.13: Tính tích và tổng hai số A,B**

Phần này hướng dẫn cách làm các phép toán thông thường trong LabVIEW như phép chia lấy dư, phép tính căn bậc 2, hàm số mũ, trị tuyệt đối, nhân với -1, vv.

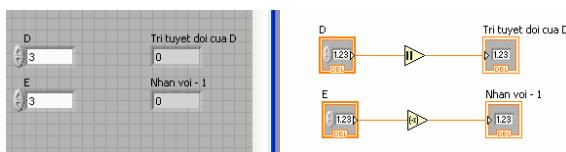
**Chuột phải trên BD, Chọn Programming, Chọn Numeric, ta lấy**

hàm căn bậc hai và chia lấy dư . Tạo các Control và Indicator như hình. Ta sẽ tính được căn của A là 1.41421. Và với phép chia B cho C (10:3) ta được phần nguyên là 3, phần dư là 1.



**Hình 2.14: Phép tính căn và chia lấy dư**

Tương tự, ta có phép trị tuyệt đối và nhân với -1 như trong Hình 2.15. Hãy mở file ví dụ 3: các phép tính để tham khảo chương trình mẫu

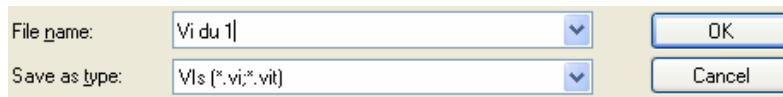


**Hình 2.15: Phép trị tuyệt đối và nhân với -1**

Bạn có thể xem ví dụ về các phép toán trong CD kèm theo sách tại thư mục **CD/Bài 2/Ví dụ2-2 Các phép toán.vi**

## 2.5. Lưu file, mở file, tìm ví dụ, công cụ trợ giúp

Trong LabVIEW, để lưu file ta chọn **File, Save**, Chỉ đường dẫn vị trí để lưu file và **đặt tên** cho file cần lưu là **Ví dụ 1**, nhấn **OK**. Đôi khi file chương trình LabVIEW thì đuôi của nó là **.VI** (tương tự như văn bản trong Microsoft Word có đuôi là **.DOC** hoặc **.DOCX**). Chữ VI viết tắt của chữ Virtual Instrumentation – thiết bị ảo.

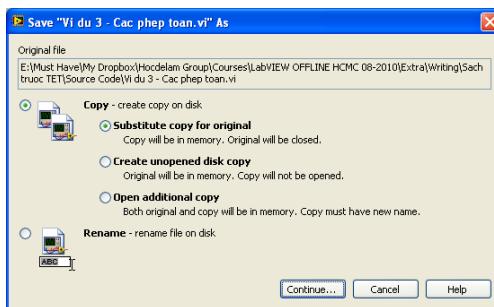


**Hình 2.16: Lưu file**

Để tạo lưu file hiện tại đang mở thành một file mới ta chọn Save As, hộp thoại như Hình 2.17 sau sẽ hiện ra. Ý nghĩa của ba lựa chọn trong mục **Copy** là:

- **Substitute...** có nghĩa File mới tạo sẽ được mở, file cũ sẽ đóng lại.
- **Create...** có nghĩa file cũ sẽ được mở, file mới tạo được đóng lại.
- **Open...** có nghĩa cả hai file cũ và mới tạo sẽ được mở.

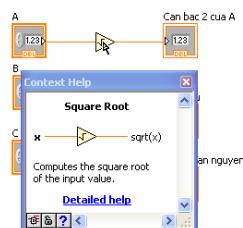
Nếu bạn chọn mục **Rename** thì không có file nào được tạo ra nhưng bạn có thể thay đổi tên của file hiện tại.



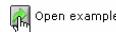
**Hình 2.17: Các lựa chọn khi lưu file bằng Save As**

Để mở file, ta vào File, Open và chọn File cần mở. Lưu ý, các file LabVIEW được lập trình bởi LabVIEW phiên bản quá cũ sẽ không mở được bằng phần mềm LabVIEW phiên bản cao. Và bạn cũng không thể mở các file được lưu bằng LabVIEW có phiên bản cao hơn bằng LabVIEW có phiên bản thấp hơn. Để khắc phục điều này, bạn hãy sử dụng chức năng **Save for previous version** (trong File) khi muốn file của bạn có thể mở bằng các phiên bản LabVIEW thấp hơn.

**Context Help** nhằm giúp giải thích các hàm, các đối tượng trong LabVIEW là một công cụ vô cùng thiết thực. Để hiện Context Help của bất kỳ hàm nào bạn chưa nắm rõ cách dùng, Chọn Ctrl + H. Sau đó đưa chuột lên hàm bất kỳ (ví dụ, hàm tính căn bậc 2), ta sẽ thấy Context Help hiện ra. Để làm mất Context Help, ta nhấn Ctrl + H một lần nữa.

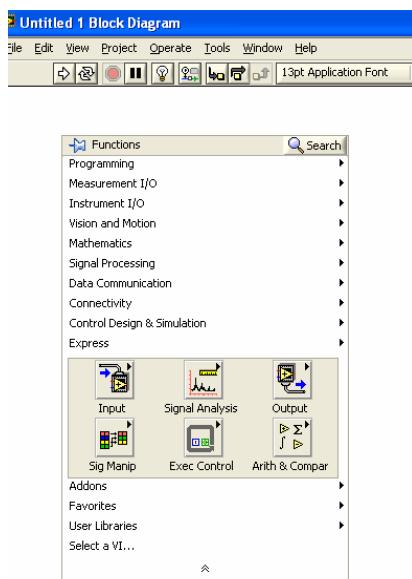


**Hình 2.18: Nhấn Ctrl + H và đưa chuột vào vị trí hàm để thấy Context Help**

Để đọc chi tiết Context Help, bạn chọn Detailed Help màu xanh. Từ Detailed Help, bạn có thể cuộn chuột xuống dưới cùng để mở các ví dụ  có sẵn trong LabVIEW để tham khảo thêm về hàm đang tìm hiểu. Học qua ví dụ là một phương pháp tốt cho những người mới bắt đầu.

## 2.6. Phần đọc thêm

Phần này giới thiệu tóm tắt các thư viện của LabVIEW nằm trong BD. Khi mở BD của LabVIEW bạn thấy có rất nhiều thư viện như trong **hình 2.19**.



**Hình 2.19: Các thư viện hàm trong LabVIEW**

Các thư viện đó có ý nghĩa như sau:

**Programming** là nơi chứa hỗ trợ công cụ, hàm lập trình nói chung (giống các ngôn ngữ khác như C, Matlab, vv...)

**SignalExpress** hỗ trợ thu thập dữ liệu, hiển thị tín hiệu trên máy tính.

**Advanced Signal Processing Toolkit** hỗ trợ xử lý tín hiệu nâng cao.

**Control Design and Simulation** hỗ trợ xây dựng các mô hình động lực học của các hệ thống và thiết kế bộ điều khiển (giống Matlab Simulink).

**Digital Filter Design Toolkit** hỗ trợ thiết kế bộ lọc số.

**PID and Fuzzy Logic Toolkit** hỗ trợ thiết kế bộ điều khiển PID hoặc Fuzzy Logic

**FPGA** hỗ trợ lập trình FPGA

**Real-Time** hỗ trợ lập trình ứng dụng thời gian thực

**Internet Toolkit** hỗ trợ giao tiếp qua mạng Internet.

**Database Connectivity Toolkit** hỗ trợ kết nối cơ sở dữ liệu.

**Vision Development Module** hỗ trợ công cụ phát triển hệ thống thu thập và xử lý ảnh.

**Simulation Interface Toolkit** cho phép kết nối LabVIEW và Matlab Simulink.

**Vision and Motion** hỗ trợ lập trình chuyển động nhiều bậc.

**LabVIEW Embedded Modudle** là modun lập trình nhúng, dùng để lập trình vi điều khiển.

**Express:** Là nơi chứa các hàm thường dùng. Các hàm thường dùng là tập con của thư viện LabVIEW **Programming**.

Trong các thư viện thì thư viện **Programming** là quan trọng nhất. Đó chính là nơi chứa toàn bộ hàm cơ sở của môi trường lập trình đồ họa LabVIEW. Và ở tập sách này, chúng ta sử dụng các hàm trong Programming, sau đó áp dụng các hàm này để bắt tay vào thực hành thu thập dữ liệu, điều khiển PID động cơ DC,...với phần cứng thực.

Từng module trên sẽ trình bày chi tiết trong các tập sách xuất bản tới đây. Các yêu cầu và đóng góp cho các tập sách tiếp theo, vui lòng liên hệ tác giả qua E-mail: bahai.ce@gmail.com.

## 2.7. Bài tập

**Câu 1:** Nêu ý nghĩa của Front Panel và Block Diagram.

**Câu 2:** Nêu cấu trúc chung của một chương trình LabVIEW

**Câu 3:** Viết chương trình thực hiện phép cộng, trừ, nhân, chia hai số A,B. A,B được nhập vào từ bàn phím.

**Câu 4:** Viết chương trình tính hai phép tính sau biết A,B,C,n được nhập vào từ bàn phím.

a)  $\sqrt{A}(17B - C)$

b)  $\sqrt{1102A} \frac{|2B - 3C|}{2011}$

c)  $\frac{\sqrt[n]{2011(A - B)}}{C}$  (gợi ý câu 4c: Hãy chọn công cụ search ở BD để tìm hàm “power of x”).

## BÀI 3

# CÁC LOẠI CONTROL VÀ INDICATOR

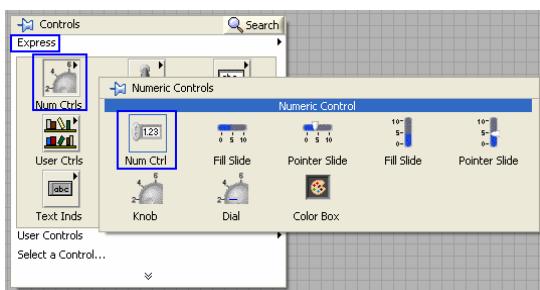
Học xong bài này, bạn sẽ nắm được:

- Các loại Control, Indicator thường dùng
- Các kiểu dữ liệu của các Control và Indicator
- Chuyển đổi qua lại giữa các kiểu dữ liệu
- Bảng, biểu đồ, đồ thị

### 3.1. Các Control thường dùng

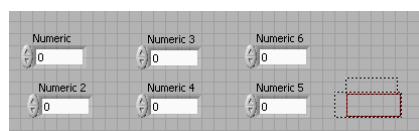
Các control thường dùng gồm các dạng: numeric, fill slide, pointer slide, knob, dial, constant, string, simulated signal, vv.

Để lấy các Control, ta vào thư viện theo đường dẫn: **FP> Express > Numeric Control** như Hình 3.1 và chọn Control cần lấy. Ví dụ lấy Numeric control: Right click trên cửa sổ FP, chọn **Express**, chọn **Numeric Control**, **Chọn Numeric Control** và đặt ra màn hình. Tương tự đối với Slide control và String control ta làm như trên.



*Hình 3.1: Cách lấy Control*

Ta có thể copy nhanh các Control bằng cách nhấn Ctrl (Trên bàn phím) và kéo thả các Control bằng chuột trái.



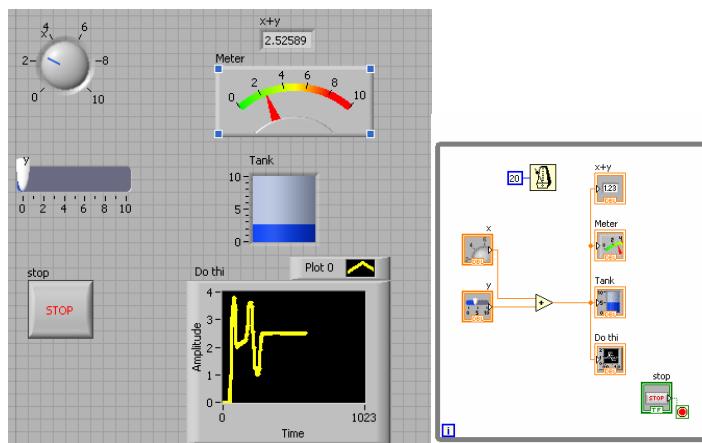
*Hình 3.2: Copy nhanh bằng việc kéo thả*

Các Control thường dùng được tóm tắt tại bảng sau:

	<b>Numeric control:</b> là control dạng số
	<b>Fill slide:</b> Control có dạng một thanh trượt
	<b>Pointer slide:</b> Control dạng thanh trượt có nút kéo
	<b>Knob:</b> Nút vặn
	<b>Dial:</b> Đĩa xoay
	<b>Constant:</b> hằng số, hay có thể xem là một dạng numeric control nhưng giá trị không thay đổi trong suốt quá trình chạy chương trình. Để tạo Constant ta lấy một Numeric control, chuột phải lên Numeric control chọn <b>Change to constant (Quy tắc vàng 1)</b> .
	<b>String control:</b> là 1 dòng chữ hay còn gọi là text control. Sử dụng text Control này để nhập các chữ hoặc chuỗi ký tự, hoặc một câu văn, xem thêm bài 4 – case structure.
	<b>Simulated signal:</b> một tín hiệu được mô phỏng sẵn trong LabVIEW có thể dùng như một Indicator trong một số trường hợp. Lấy Simulated signal bằng cách vào: BD> Express> Input> Simulated Signal. Xác lập các thông số khi bảng thông số hiện ra.  Nối đầu ra của khối vừa lấy với một Graph bằng cách chọn Right Click lên đầu ra, Create> Indicator.

Thường các control được chia làm 3 dạng: Boolean, Numeric và String.

Ví dụ về sử dụng các loại control và Indicator khác nhau được trình bày trong hình 3.3. Xem ví dụ này trong CD kèm theo sách này tại thư mục **CD/Bài 3/Ví dụ3-1 các dạng Control và Indicator.vi**

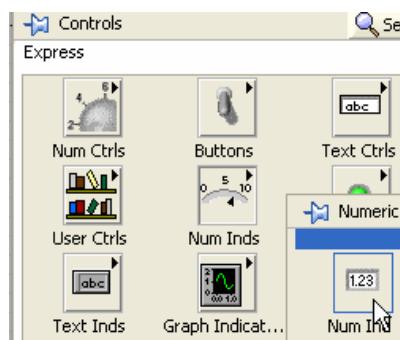


*Hình 3.3 Ví dụ về các control và Indicator*

### 3.2. Các dạng Indicator thường dùng

Tương tự các Control, ta cũng có các Indicator như: numeric, string, slide, vv... Giữa Control và Indicator có thể chuyển đổi qua lại nhờ vào thao tác **Right Click> Change to Control** hoặc **Change to Indicator**. Như vậy ta lại thấy quy tắc vàng **Right Click** đã phát huy hiệu quả (việc chuyển đổi qua lại) trong trường hợp này.

Để lấy các Indicator, ta vào thư viện theo đường dẫn: **FP> Express > Numeric Indicator** như Hình 3.1 và chọn Control cần lấy. Ví dụ lấy Numeric control: Right click trên cửa sổ FP, chọn **Express**, chọn **Numeric Indicator**, Chọn **Numeric Indicator** và đặt ra màn hình. Tương tự đối với Slide control và String control ta làm như trên.

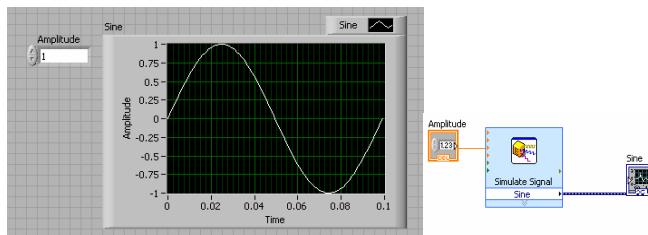


*Hình 3.3: Cách lấy Indicator*

Bảng tóm tắt các Indicator như sau:

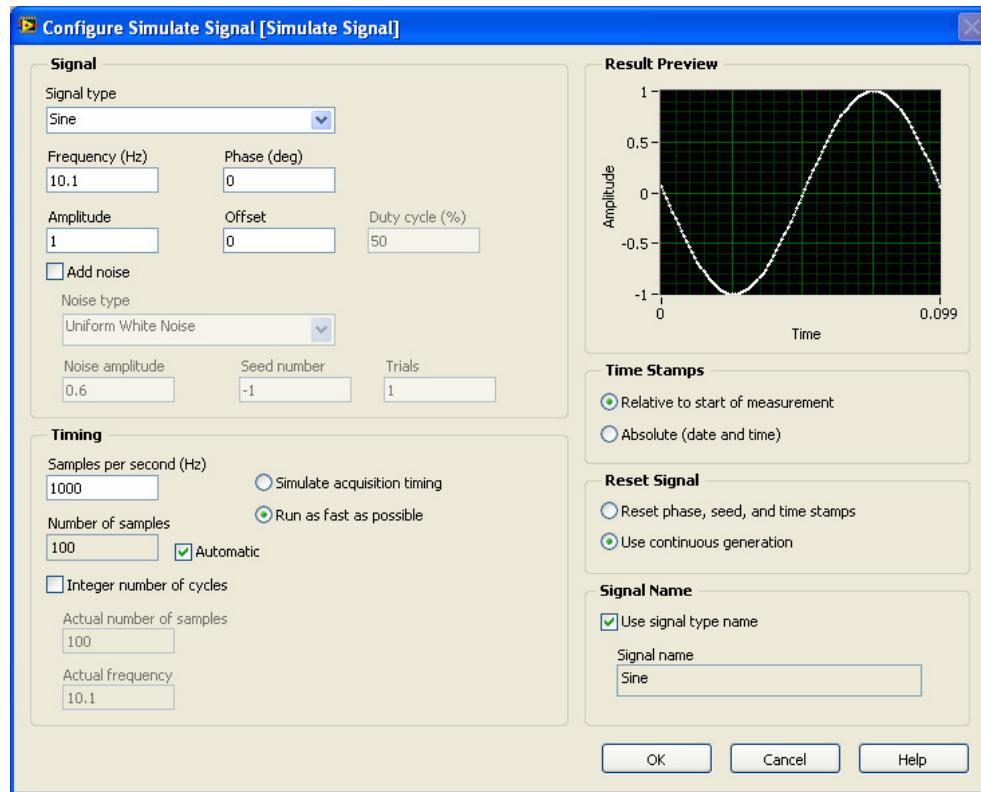
	<b>Numeric indicator:</b> là Indicator dạng số
	<b>Meter:</b> Indicator có dạng đồng hồ vuông
	<b>Gauge:</b> Indicator dạng đồng hồ vuông
	<b>Thermometer:</b> Cột nhiệt độ
	<b>Graduated Bar:</b> Thanh hiển thị quá trình
	<b>String:</b> là 1 dòng chữ hay còn gọi là text Control, dùng để xuất các chữ hoặc chuỗi ký tự, hoặc một câu văn, xem thêm bài 4 – Case structure. Lấy string indicator tại FP> Modern> String & Path> String Indicator.
	<b>Chart:</b> là biểu đồ hiển thị các giá trị theo trục thời gian
	<b>Graph:</b> là đồ thị thường được dùng để hiển thị các tín hiệu dạng sóng (waveform).
	<b>XY Graph:</b> đồ thị hiển thị quan hệ giữa hai tín hiệu X và Y hoặc dùng trong bài vẽ đồ thị hàm số $y=f(x)$ – được trình bày trong bài 4.

Ví dụ về sử dụng Chart trình bày như hình 3.4.



**Hình 3.4: Cách lấy Indicator**

Trong hình trên, để tạo sóng Sine, ta lấy Simulated Input theo đường dẫn: BD> Express> Input> Simulated Input và đặt thông số như mặc định trong hình 3.5.

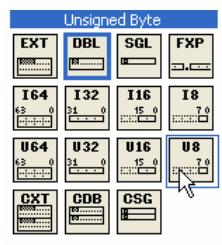


**Hình 3.5: Cách lấy Indicator**

Ngoài ra, một Numeric Indicator đã được sử dụng để chỉnh Amplitude (biên độ) của sóng Sine, và một Graph Indicator để hiển thị sóng Sine. Xem file **Vi du 3-2 Waveform display.vi** trong CD kèm theo sách.

### 3.3. Kiểu dữ liệu trong LabVIEW và chuyển đổi dữ liệu

Kiểu dữ liệu là công cụ dùng để gán (ép) một numeric hay indicator vào một dãy giá trị nào đó. Ví dụ, ta muốn một Numeric Control có giá trị (và chỉ nắm trong giá trị đó ngay cả khi người điều khiển nhập giá trị khác) là 0-255 thì ta sẽ gán Numeric Control đó vào kiểu dữ liệu Unsigned 8 bit (Viết tắt là **U8**). Muốn đổi kiểu dữ liệu của một Control, ta chọn Right Click lên Control đó> Chọn Representation> Chọn Kiểu cần gán. Hình 3.6 liệt kê các kiểu dữ liệu trong LabVIEW.



**Hình 3.6: Các kiểu dữ liệu trong LabVIEW**

Trong LabVIEW có nhiều kiểu dữ liệu khác nhau, ví dụ:

Ký hiệu	Kiểu dữ liệu	Số bit	Khoảng giá trị
I8	Byte signed integer	8	-128 tới 127
I16	Word signed integer	16	-32,768 tới 32,767
I32	Long signed integer	32	-2,147,483,648 tới 2,147,483,647
I64	Quad signed integer	64	-1e19 tới 1e19
U8	Byte unsigned integer	8	0 tới 255
U16	Word unsigned integer	16	0 tới 65,535
U32	Long unsigned integer	32	0 tới 4,294,967,295

Cách chuyển đổi kiểu dữ liệu: chuột phải lên đối tượng muốn chuyển đổi chọn Representation rồi chọn kiểu dữ liệu mong muốn.

### 3.4. Bài tập

**Bài 1:** Viết chương trình tính  $y = 4x^2 + 8$ , với  $x$  nhập từ bàn phím. Trong đó,  $x$  có kiểu Numeric Control,  $y$  là Indicator kiểu đồng hồ.

**Bài 2:** Viết chương trình vẽ đồ thị:  $y = n^2x^2 + m$ . Tập xác định (TXĐ):  $x = \{x \in Z \mid -100 \leq x \leq 100\}$ . Trong đó,  $n$  được nhập vào từ bàn phím và có dạng Numeric Control.

**Bài 3:** Viết chương trình tính  $Y = |b-c|$

$Y$  là Numeric có kiểu Unsigned 8 bit.

$b, c$  là các Numeric kiểu Double precision.

## BÀI 4

# VÒNG LẶP (WHILE LOOP)

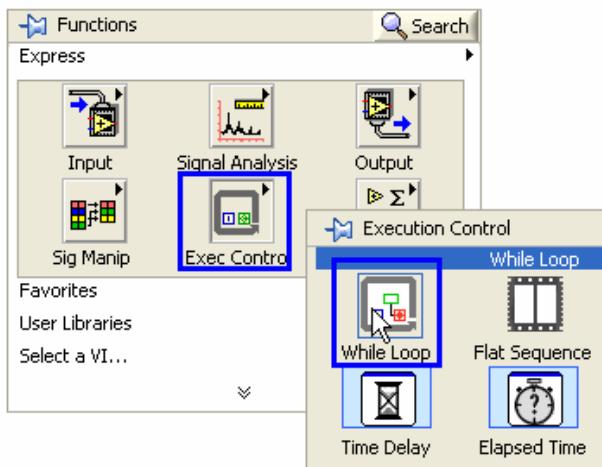
# VÒNG LẶP FOR (FOR LOOP)

Học xong bài này, bạn sẽ nắm được:

- Sử dụng được vòng lặp, cấu trúc thường gặp trong LabVIEW
- Các ứng dụng của while loop, for loop

### 4.1. Vòng lặp while

Vòng lặp while là vòng lặp có điều kiện như trong hình 4.1. Ý nghĩa của vòng lặp while là cho phép chạy chương trình mãi tới khi nào nút stop được nhấn thì mới dừng lại. Để lấy while loop ta vào **BD> Express> Execution> While loop** như trình bày trong hình 4.1

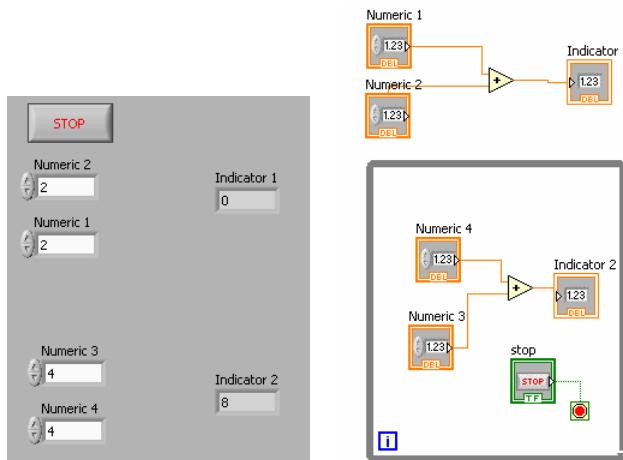


Hình 4.1: Lấy while loop tại BD

While loop sẽ lặp lại chương trình được **đặt trong vòng lặp** này, tới khi nút Stop (conditional terminal) nút Stop tại FP được nhấn (Lưu ý rằng nút Stop có dạng dữ liệu là Boolean-true hoặc false).

Xét ví dụ trong hình 4.2 như sau: Có hai phép tính cộng. Phép cộng A là cộng **Numeric 1 và Numeric 2** hiển thị kết quả ra **Indicator 1**, phép cộng B là phép cộng **Numeric 2 và Numeric 3** hiển thị kết quả ra **Indicator 2**. Khác biệt duy nhất là phép cộng A đặt ngoài vòng lặp còn phép cộng B

đặt trong vòng lặp. Bạn hãy chạy chương trình (nhấn phím tắt Ctrl+R), sau đó thay đổi giá trị của Numeric 1, 2, 3, 4. Ta sẽ quan sát thấy phép cộng A sẽ không có giá trị thay đổi tại Indicator 1. Còn Indicator 2 sẽ thay đổi giá trị khi bạn thay đổi giá trị nhập vào Numeric 3 và 4. FP và BD của ví dụ này được trình bày trong hình 4.2. Xem ví dụ này trong **CD> Bài 4> while loop.vi**

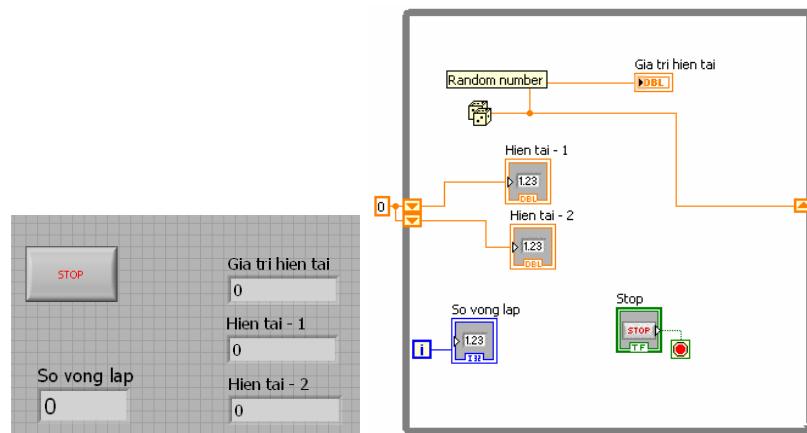


**Hình 4.2: Tính tổng với while loop**

Ngoài ra, trong while loop còn có chân iteration (**i**) có ký hiệu (**□**) là bộ đếm số lần lặp hiện tại (current loop iteration count) (lần thực hiện đầu tiên  $i=0$ ). Giá trị lớn nhất của (**i**) là  $2,147,483,647$  ( $2^{31}$ ), và giữ mãi mãi sau đó. Nếu cần giá trị lớn hơn  $2,147,483,647$ , dùng shift registers có miền giá trị integer lớn hơn, hoặc sử dụng một vòng lặp for loop trong trường hợp này.

Ngoài ra, ta còn có thể sử dụng shift-register trong while loop. Shift-register là một thanh ghi hoạt động như một ô nhớ. Ta sử dụng thanh ghi này để truy cập lịch sử giá trị của một tín hiệu nào đó, ví dụ chương trình của chúng ta đang chạy ở thời gian là 10:30am, và ta muốn xem lại giá trị của tín hiệu đo được từ cảm biến nhiệt độ ở thời điểm 10:29am. Ngoài ra, Shift register cũng có thể dùng để thực hiện các phép toán cộng dồn (Xem trong bài điều khiển PID trong tập 2).

Nói cách khác là có thể dùng shift registers để “chuyển” giá trị của đối tượng/tín hiệu nào đó ở vòng lặp này sang vòng lặp kế tiếp. Bạn đọc có thể xem thêm ví dụ cách sử dụng Shift register trong **CD/Bài 4/shift register.vi** hoặc ví dụ dùng shift register để lập trình bộ điều khiển PID điều khiển động cơ DC sử dụng card Hocdelam USB-9001 ở bài 10 và tải ví dụ này tại website <http://labview.hocdelam.org> (Mục ví dụ mẫu).

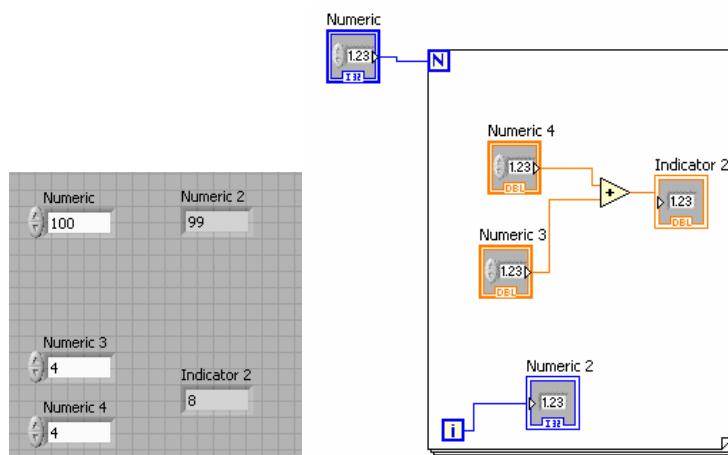


Hình 4.3: Sử dụng shift register

## 4.2. Vòng lặp for

Cũng như while loop, for loop là vòng lặp mà số lần lặp lại có thể định trước bởi người lập trình. Lấy for loop trong thư viện hàm như sau: **BD> Express> Execution> For loop.**

Ví dụ ta muốn chạy chương trình tính tổng A+B trong 100 lần thì sau 100 lần lặp lại phép tính tổng thì chương trình sẽ tự thoát (tự dừng). Như trong hình ta đã đặt số vòng lặp cần lặp lại là 100 (đặt thông qua khối Numeric). Sau khi chạy chương trình thì Numeric 2 (đóng vai trò là Indicator chỉ ra số vòng đã lặp được) sẽ báo số 99. Lý do là LabVIEW đếm vòng lặp đầu tiên có giá trị ( $i=0$ ).



Hình 4.4: For loop

### 4.3. Ứng dụng vòng lặp để vẽ đồ thị

Áp dụng **XY Graph** trong bài 3 và **While loop** trong bài này để vẽ đồ thị hàm số  $y=f(x)$  theo yêu cầu sau: Viết chương trình vẽ đồ thị:  $y = 2x$ . Tập xác định (TXĐ):  $x = \{x \in Z \mid -100 \leq x \leq 100\}$ .

**Theo cách vẽ đồ thị thông thường** được hướng dẫn bởi các thầy, cô dạy toán học phổ thông thì việc vẽ đồ thị được thực hiện như sau:

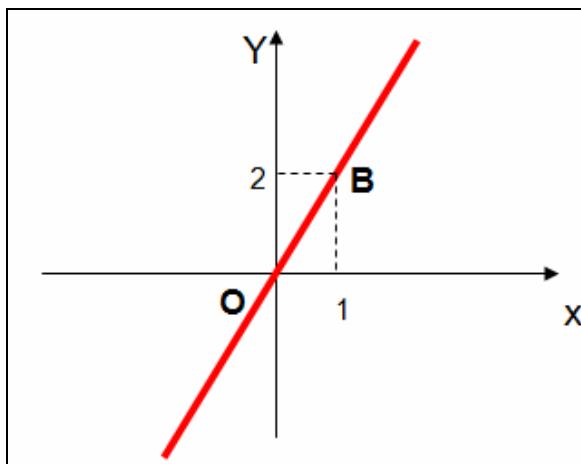
- Lập bảng biến thiên: Do  $y = 2x$  có dạng đường thẳng nên ta chỉ cần lấy tọa độ x,y của hai điểm mà đường thẳng đó đi qua, sau đó nối hai điểm đó lại sẽ cho ta đồ thị của đường thẳng. Bước sau cùng là xóa đồ thị nằm phía ngoài TXĐ và chỉ để lại phần đồ thị nằm trong khoảng  $-100 \leq x \leq 100$ .
- Bảng biến thiên:

Giống như kiến thức ở phổ thông đã học, ta dễ dàng lập bảng biến thiên bằng cách cho giá trị của  $x=0$  và  $x=1$ . Tính  $y$  theo công thức  $y = 2x$ .

**Bảng 4.1 Bảng biến thiên của hàm số  $y = 2x$**

X	$-\infty$	0	1	$+\infty$
Y		0	2	

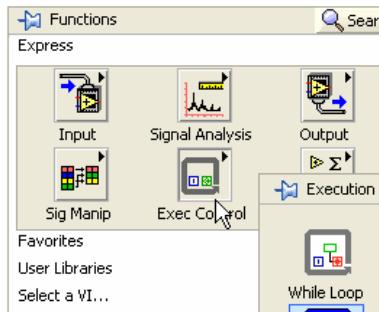
Từ bảng biến thiên trên ta có kết luận là đồ thị  $y = 2x$  là một đường thẳng đi qua hai điểm A(0;0) trùng với gốc tọa độ O và B(1;2). Đồ thị được vẽ dễ dàng bằng thước kẻ nối hai điểm A,B như sau:



**Hình 4.5: Đồ thị hàm số  $y = 2x$**

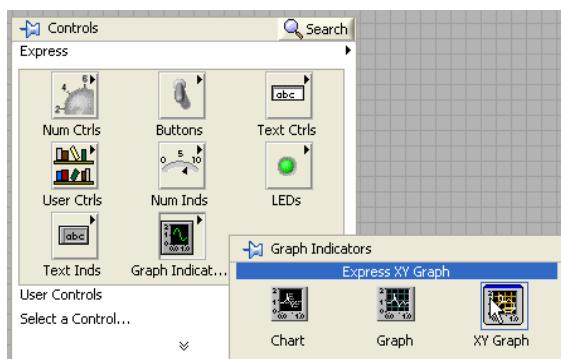
Hãy cùng vẽ đồ thị này bằng LabVIEW sử dụng XY Graph như sau:

- Lấy While loop tại BD>Express>Exec Control>While loop như hình 2.7.



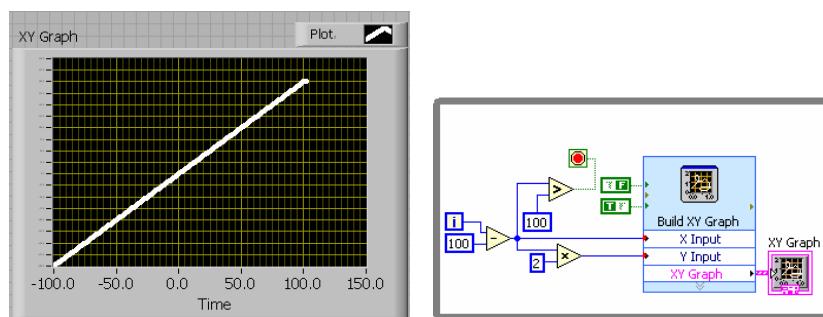
*Hình 4.6: Lấy hàm while loop*

- Lấy đồ thị XY Graph để vẽ đồ thị tại FP>Express> Graph Indicator> XY Graph và tạo các Control, Indicator như hình 2.8.



*Hình 4.7: Vị trí của XY Graph*

Xem file **Vi du 4-1 Dothi y=2x.vi** trong CD kèm theo sách.



*Hình 4.8: Đồ thị hàm số  $y=2x$  vẽ bằng LabVIEW sử dụng hàm while loop*

Trong block diagram ở hình 2.9, ta cần tạo hai Boolean control tại hai chân: Enable – cho phép vẽ đồ thị là True và Reset – xóa đồ thị là False. Việc tạo hai Boolean Control này có thể sử dụng quy tắc Right click> Create constant hoặc ta tạo Control và chọn Right click> Change to constant.

### Bạn có biết?

Nếu đã hoàn thành phần này, bạn đã có đủ khả năng để thực hiện các ứng dụng thú vị với LabVIEW thông qua một trong các Card DAQ (card thu thập dữ liệu) như **NI USB-6008**, **NI USB-6009**, hoặc phần cứng giá rẻ dành cho sinh viên, giảng viên **Hocdelam USB-9001**, **Hocdelam USB-9009**, các ứng dụng bạn có thể nhanh chóng thực hiện là:

- Thu thập dữ liệu từ các cảm biến đo nhiệt độ LM35, đo khoảng cách Sharp, đo áp suất, đo độ chân không, độ pH., đo vận tốc động cơ DC, độ nghiêng của một mặt phẳng, vv.
- Điều khiển đèn LED, động cơ DC, AC, vv...
- Để thực hiện ngay được các ứng dụng này, bạn vui lòng bỏ qua các bài tiếp theo và xem tiếp các Bài 10, 11.
- Phần cứng có thể đăng ký mượn (nếu bạn là sinh viên hoặc giảng viên) hoặc đặt mua tại: <http://sanpham.hocdelam.org>.

## 4.4. Bài tập

**Câu 1:** Hãy nêu khái niệm về while loop

**Câu 2:** Vẽ đồ thị hàm số sau:  $y = Ax + B$ , A và B được nhập vào từ bàn phím.

**Câu 3:** Thực hành đo lường các cảm biến vào máy tính. (Tham khảo thêm Bài 10, 11 để thực hiện câu này).

## BÀI 10

# CÁC ỨNG DỤNG VỚI CARD HOCDELAM USB 9001 VÀ HDL-9000

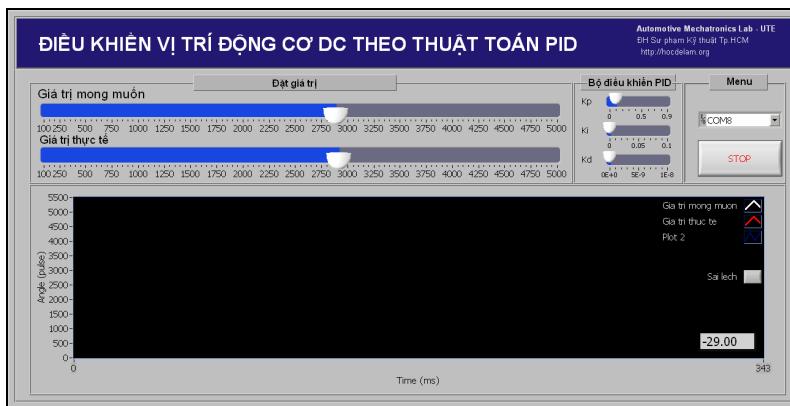
Học xong bài này, bạn sẽ nắm được:

- Thông số kỹ thuật của card Hocdelam USB-9001 và HDL-9000
- Cách thu thập dữ liệu nhanh chóng với Card Hodelam – USB 9001
- Điều khiển vận tốc và vị trí động cơ DC theo thuật toán PID sử dụng bộ thí nghiệm đa năng HDL-9000.

### 10.1 Giới thiệu card Hocdelam USB-9001

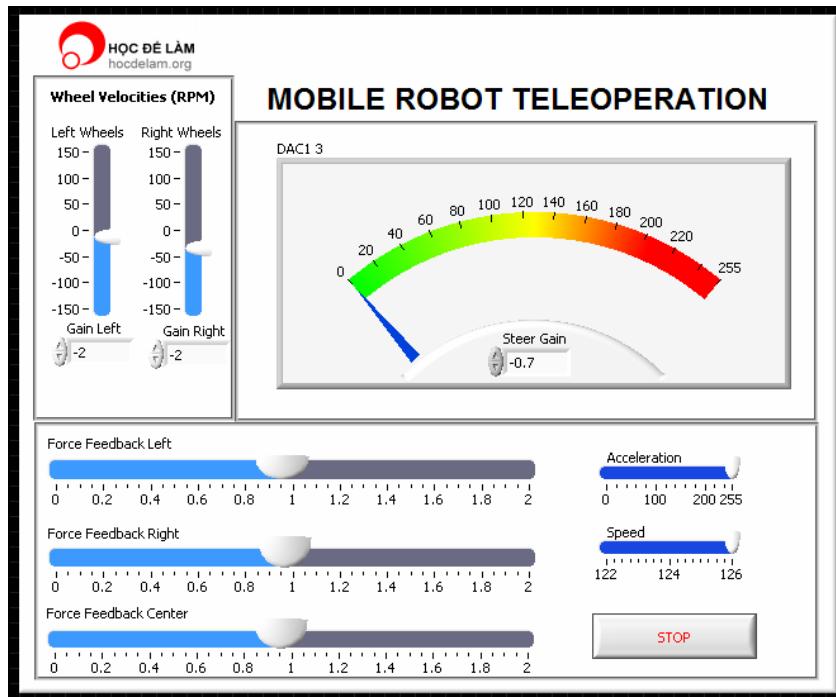
Card giao tiếp máy tính đa năng giá rẻ **Hocdelam USB 9001** có chức năng tương đương các card thu thập dữ liệu USB do các hãng nước ngoài sản xuất có trên thị trường trong và ngoài nước như NI USB 6008/6009-Hoa Kỳ, Advantech – Đài Loan, vv. Card Hocdelam USB-9001 có thể dùng để giao tiếp với máy tính qua cổng USB dựa theo chuẩn RS232. Đặc biệt, đây là sản phẩm phần cứng do Hocdelam Group sản xuất và lập trình thư viện đầy đủ và dễ sử dụng hơn những sản phẩm trên thị trường. Với card này, người sử dụng được hỗ trợ tài liệu tiếng Việt đầy đủ và hỗ trợ trực tuyến thông qua kho ví dụ mẫu tại website <http://labview.hocdelam.org> để nhanh chóng xây dựng các ứng dụng của mình.

- Xem ví dụ LabVIEW mẫu: **CD> Bai 10> Vi du Hocdelam-9000 PID.vi** để biết cách lập trình PID điều khiển động cơ DC. Và Video clip demo: **CD/Demo/DC motor control.**



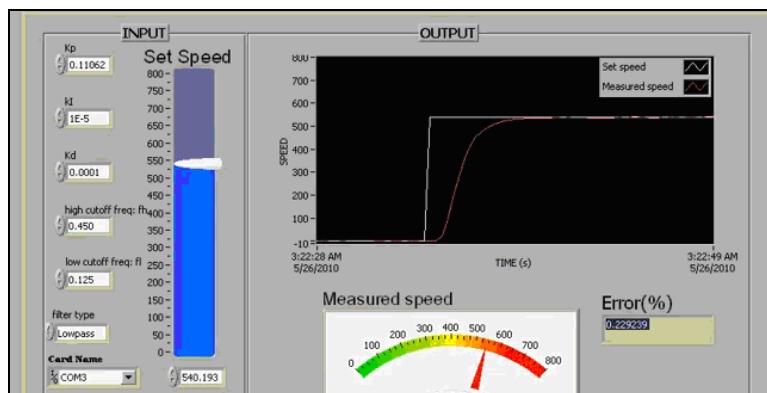
**Hình 10.1 Điều khiển vị trí động cơ DC theo thuật toán PID**

Một ví dụ khác là sử dụng card Hocdelam USB 9001 để điều khiển một mobile robot như hình 10.2. Chi tiết về lập trình Robot 3D và điều khiển PID vận tốc robot sẽ được cập nhật trong lần tái bản tiếp theo của cuốn sách này.



**Hình 10.2 Thu thập dữ liệu từ các cảm biến vào máy tính phục vụ điều khiển mobile robot**

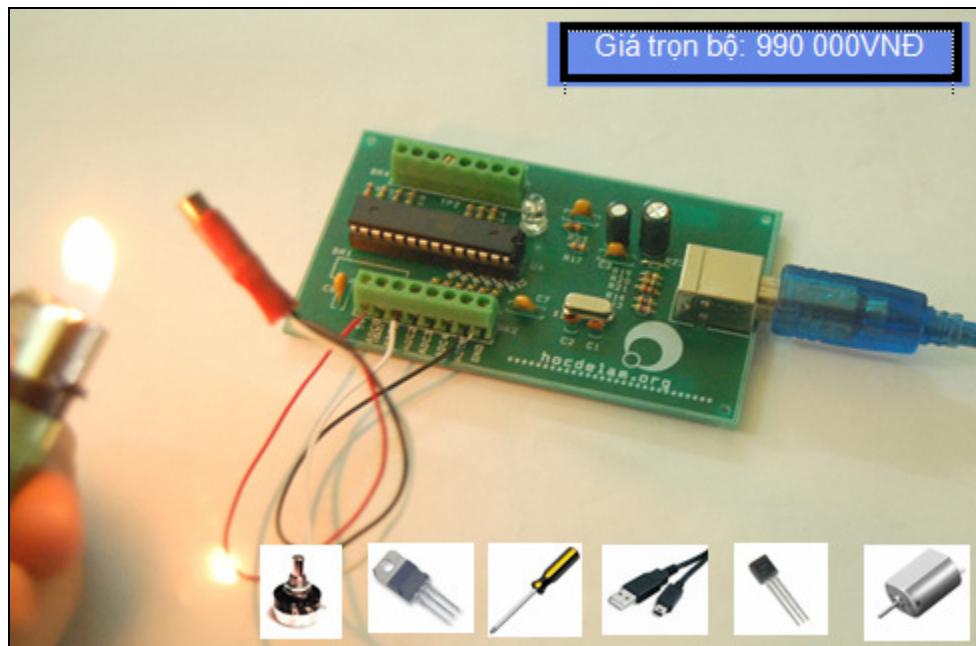
Bạn cũng có thể sử dụng card Hocdelam USB-9001 để thực hiện bài toán điều khiển vận tốc động cơ DC thường gặp trong công nghiệp.



**Hình 10.3 Điều khiển vận tốc động cơ DC theo thuật toán PID**

Tới bây này, bạn đã có thể nghĩ đến việc tự trang bị cho mình một bộ dụng cụ học tập, nghiên cứu LabVIEW giá thành thấp (hỗ trợ giáo dục) như sau:

STT	Tên	Số lượng	Giá tham khảo
1	Card Hocdelam USB-9001(*)	1 cái	890.000
2	Cảm biến nhiệt độ LM35, biến trở, LED, TIP 122	1 bộ	30.000
3	Dây điện cỡ nhỏ, lõi cứng	1 m	5000
4	Dây USB	1 cái	20.000
5	Động cơ DC loại bé	1 cái	30.000
6	Tua vít nhỏ, kiêm, vv..	1 cái	15000
<b>Tổng cộng (VNĐ)</b>			<b>990.000</b>



Giảm giá sinh viên và giảng viên | Liên hệ: <http://sanpham.hocdelam.org>

**Hình 10.4 Bộ dụng cụ học tập và nghiên cứu LabVIEW giá thấp**

## 10.2 Các ứng dụng với Card Hocdelam USB-9001

Phần này lần lượt hướng dẫn các bạn thực hiện xây dựng các ứng dụng thực tế với card Hocdelam USB-9001 gồm:

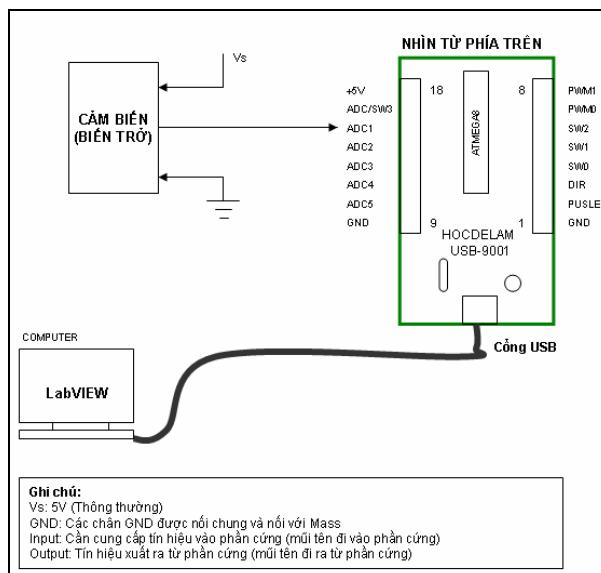
- Thu thập dữ liệu từ các cảm biến nhiệt độ, biến trở, ánh sáng, khoảng cách, vv.
- Điều khiển động cơ DC theo thuật toán P trong bộ PID.
- Điều khiển hệ thống làm mát

Để thực hiện một ứng dụng với LabVIEW, ta phải: **1-Kết nối phần cứng cần thiết, 2-cài đặt phần mềm cần thiết, và 3-tiến hành lập trình.**

### 10.2.1 Phản ứng

#### Trường hợp 1: Bạn sử dụng card Hocdelam USB-9001

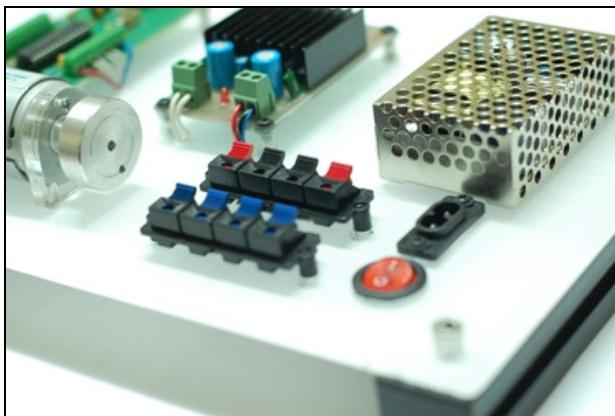
- Kết nối phần cứng theo sơ đồ **hình 10.15**: Nối chân Vout (chân giữa của biến trở, hoặc chân Vout của cảm biến LM35) vào chân ADC 1 của card Hocdelam USB-9001)



**Hình 10.5: Cách kết nối biến trở vào mạch điện**

#### Trường hợp 1: Nếu bạn sử dụng bộ thí nghiệm đa năng HDL-9000

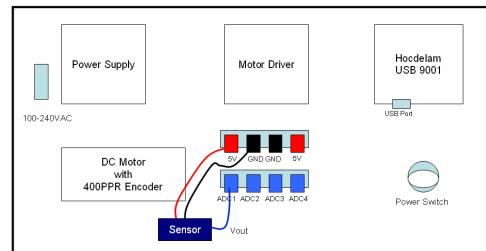
Bộ thí nghiệm đa năng HDL-9000 là một gói giải pháp toàn diện giá thành thấp (*gồm phần cứng, phần mềm tiếng Việt, bài giảng soạn sẵn, và kho thư viện ví dụ mẫu*) trong kỹ thuật dùng đào tạo đo lường cảm biến, điều khiển thiết bị theo thuật toán PID, Fuzzy logic, điều khiển và giám sát qua mạng Internet, vv. trong các ngành cơ điện tử, ô tô, tự động hóa, vật lý, vv. Hiện tại đã có hơn 400 lượt người và doanh nghiệp trong và ngoài nước sử dụng bộ thí nghiệm này. Xem hình 10.16.



**Hình 10.6 Bộ thí nghiệm đa năng HDL-9000**

Để đo cảm biến nhiệt độ hoặc biến trở với HDL-9000, bạn nối cảm biến vào mạch điện như hình dưới 10.17.

$V_{out}$  : Chân tín hiệu của cảm biến.  
Ví dụ: Nếu là biến trở thì chân tín hiệu chính là chân ở giữa. Hai chân nguồn 5V và 0V là hai chân bìa của biến trở.



**Hình 10.7 Kết nối cảm biến vào bộ thí nghiệm đa năng HDL-9000**

Tới đây, bạn đã hoàn tất phần cứng, giờ hãy bắt tay thực hiện việc cài đặt phần mềm.

### 10.2.2 Phần mềm

Khi giao tiếp ta cần cài đặt một số phần mềm như sau: **NI VISA**, **Hocdelam USB-9001 Driver**. Bạn có thể tải các phần mềm này trên website: <http://labview.hocdelam.org> > Mục tài liệu.

**Cài đặt NI-VISA** (là công cụ để LabVIEW hiểu được các cổng giao tiếp, ở đây ta dùng cổng USB). Mở CD kèm theo sách bạn sẽ thấy NI-VISA theo đường dẫn: **CD> Phan mem> NI VISA 4.2.** (Các hình hướng dẫn cài đặt sẽ không được đặt tên)

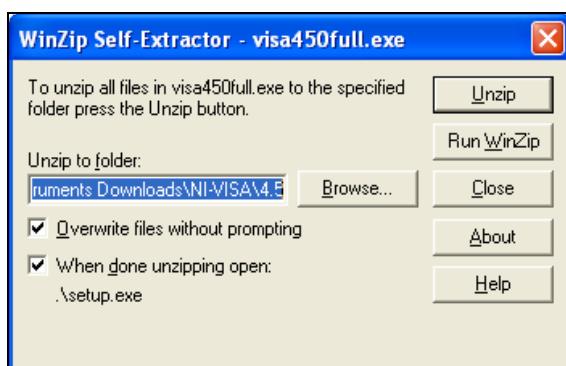
1. Giải nén



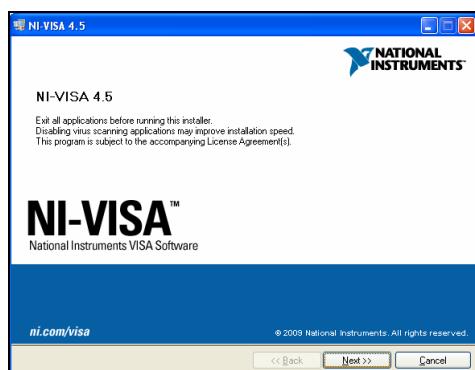
2. Chọn OK



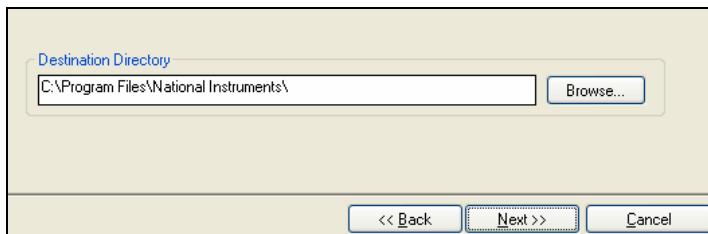
3. Chọn Unzip để giải nén.



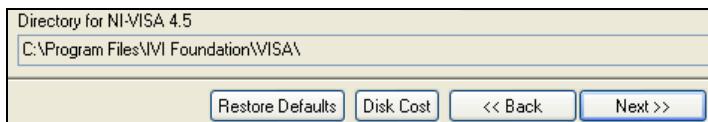
4. Sau khi giải nén chọn Run WinZip rồi chọn Next



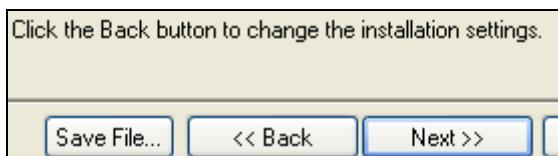
5. Chọn Next



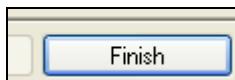
6. Chọn Next



7. Chọn Next



8. Sau khi chờ cài đặt xong, bạn chọn Finish



**Cài đặt Hocdelam USB 9001 Driver** để card Hocdelam USB-9001 có thể giao tiếp được với máy tính (Hệ điều hành) của bạn.

**Nếu máy tính của bạn là Windows XP**

- Bạn vào **CD > Phan mem > 002 Hocdelam USB 9001 Driver > Giải nén Hocdelam USB 9001 Driver Windows XP.rar**
- Chạy file: **DRVINSTSE.EXE**
- Chương trình tự động cài đặt trong vài giây là hoàn thành.

Bạn thấy hộp thoại sau xuất hiện và tự biến mất nhanh trên màn hình thì quá trình cài đặt đã thành công.

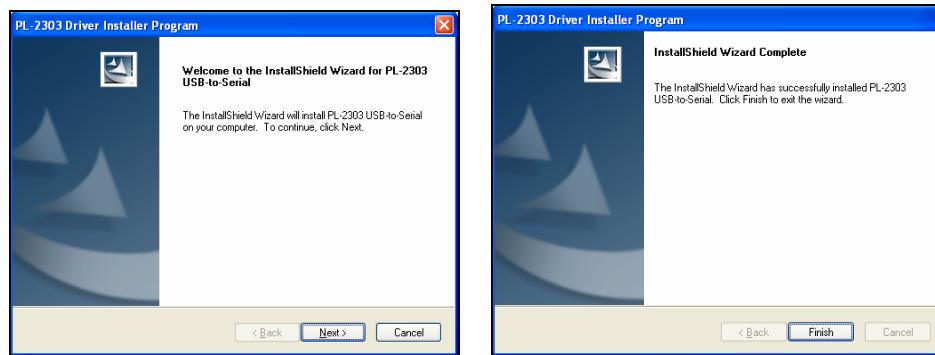


### Nếu máy tính của bạn là Windows Vista hoặc Windows 7

- Bạn vào CD > Phan mem > 002 Hocdelam USB 9001 Driver > Giải nén Hocdelam USB 9001 Windows7 Driver.rar
- Chạy file: Hocdelam USB 9001 Windows7 Driver.exe và làm theo hướng dẫn.

Chọn Next khi hộp thoại sau xuất hiện.

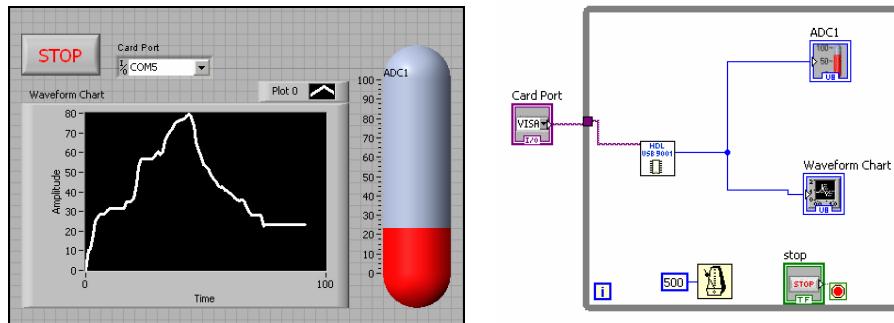
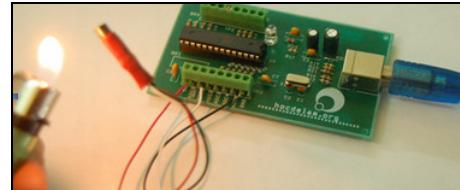
Chọn Finish để kết thúc việc cài đặt.



Đến đây bạn đã hoàn thành công việc chuẩn bị. Giờ ta bắt tay vào lập trình LabVIEW để thu thập dữ liệu từ một biến trớ hoặc một cảm biến nhiệt độ LM35.

#### 10.2.3 Đo nhiệt độ LM35

Mục tiêu ta phải thu thập được tín hiệu từ cảm biến nhiệt độ lên máy tính và vẽ một đồ thị của cảm biến này theo thời gian thực (thời gian đo).



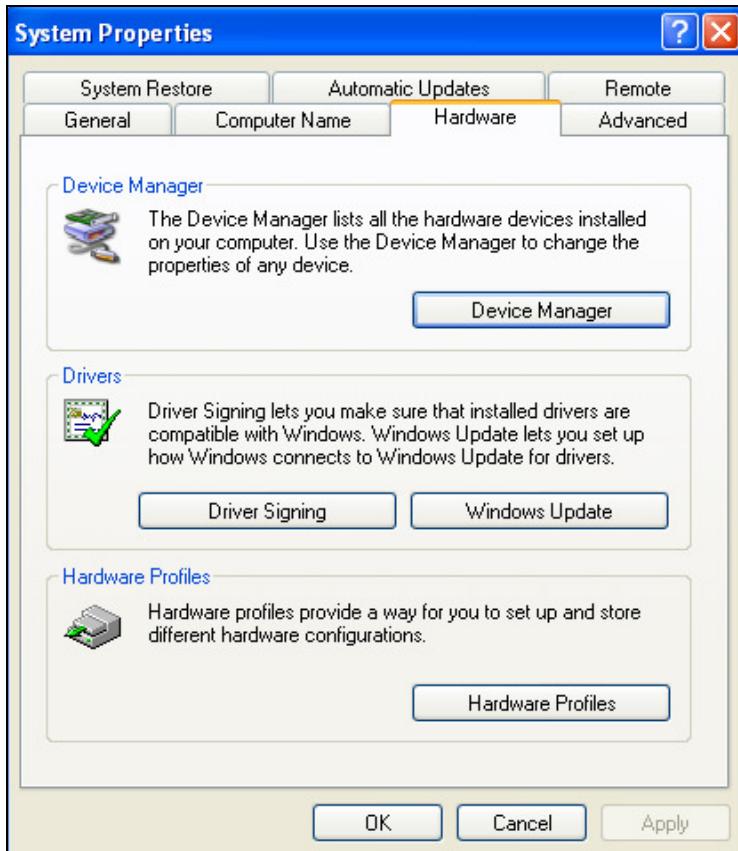
Hình 10.8: Kết quả đạt được

Hãy cùng bắt tay vào thực hiện việc lập trình

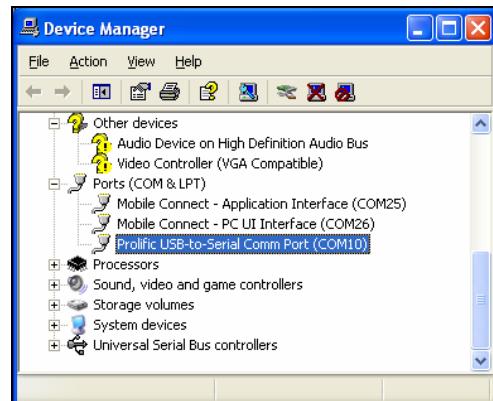
### Bước 1: Kiểm tra cổng kết nối của card giao tiếp với máy tính

Kiểm tra xem máy tính đã nhận card chưa.

- RC lên My computer > Chọn Tab Hardware > Chọn vào Port.



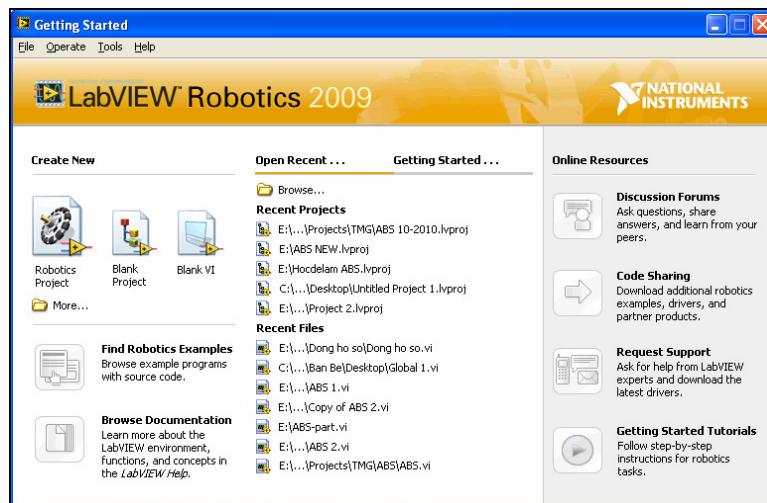
Card đã nhận đúng khi và chỉ khi có dòng chữ tô xanh trong hình.



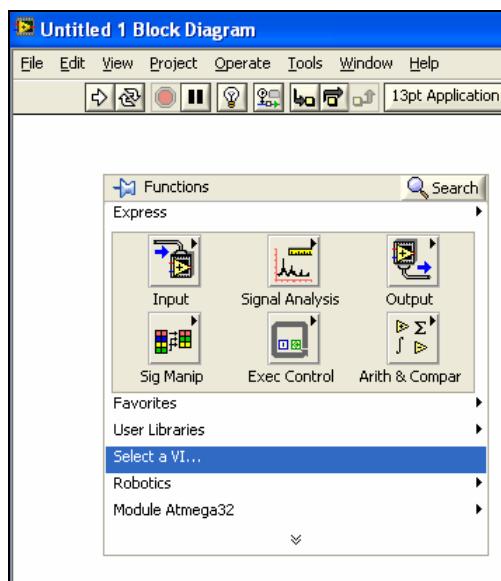
- Kiểm tra cảm biến xem còn tốt không
  - o Cáp nguồn 5V cho cảm biến. Để chân Vout tự do.
  - o Đo Vout bằng đồng hồ đo điện. Khi đo, xoay biến trở thì bạn thấy Vout sẽ thay đổi tương ứng với sự thay đổi của vị trí biến trở.

## Bước 2: Lập trình thu thập dữ liệu

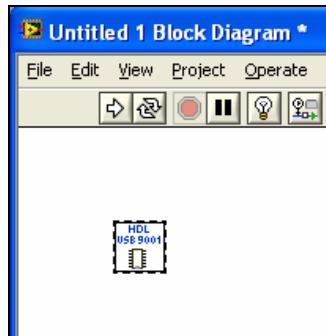
- Để lập trình thu thập dữ liệu ta tại file mới bằng cách click vào File > New



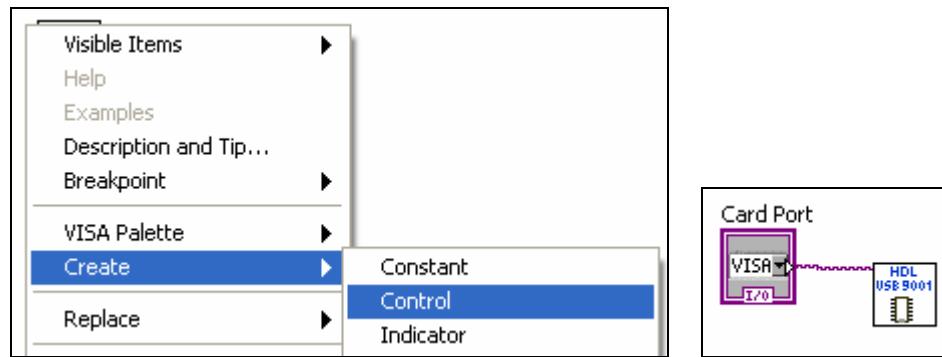
Chọn select a VI



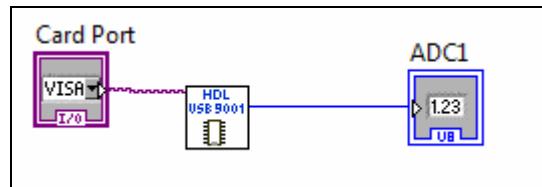
Chọn IO Library Hocdelam USB 9001 – 2009.



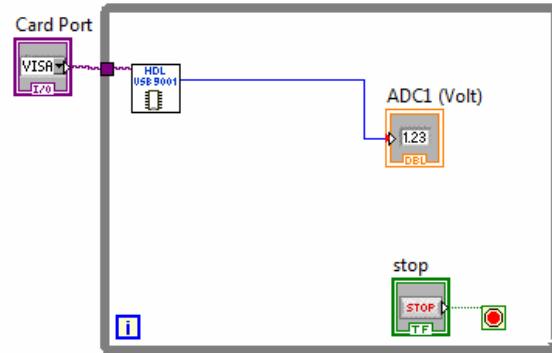
- Right click lên Card port > Create control



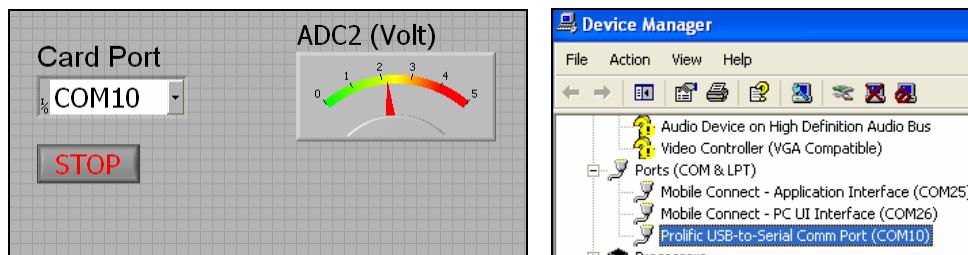
Tạo Indicator ở ADC1 (Tín hiệu đầu ra của biến trở phải được nối với chân ADC 1 của card Hocdelam USB 9001).



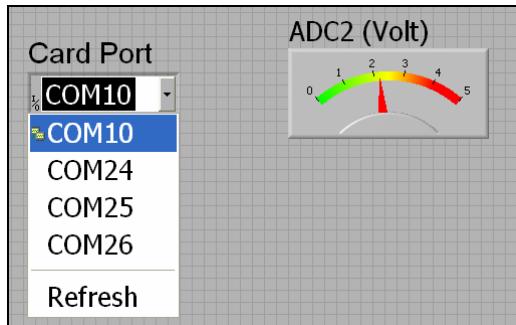
Tạo While loop như hình.



Chỉnh sửa lại front panel cho tiện dùng và quan sát. Kiểm tra số cổng COM của Card Hocdelam USB 9001.



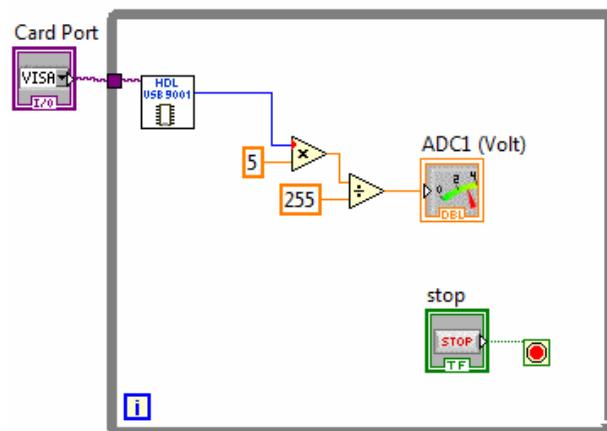
Chọn card phù hợp



*Việc cần làm tiếp là chuyển tín hiệu của vi điều khiển (0-255) thành tín hiệu điện áp (0-5Volt). Ta biết rằng, độ phân giải của kênh ADC của card Hocdelam USB-9001 là 8bit nên giá trị từ 0 đến 5Volt của cảm biến sẽ được chia thành  $2^8=256$  giá trị (tức từ 0 đến 255). Do vậy ta có công thức chuyển giá trị số thành giá trị điện áp thực như sau:*

$$V_{do} = \frac{5S_{card}}{255}$$

Như vậy,  $V_{do}$  này chính là giá trị điện áp Vout của cảm biến mà ta đo được bằng đồng hồ Voltmeter.

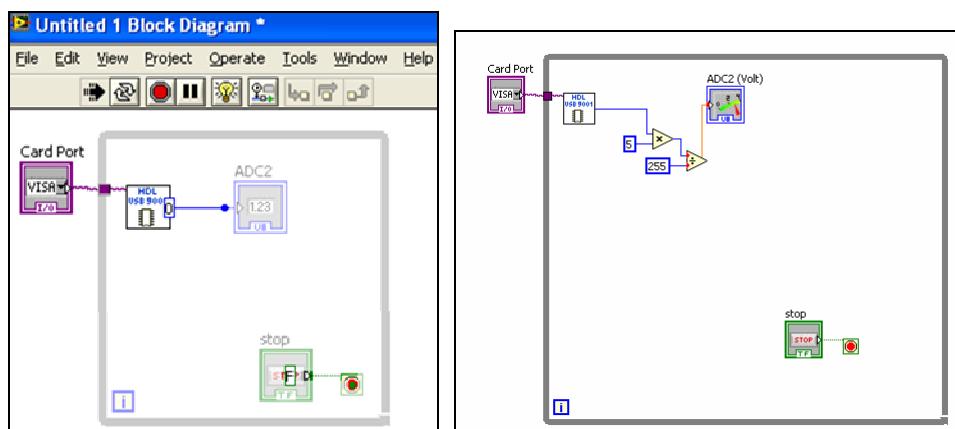


Bạn đã hoàn thành tốt bài toán thu thập dữ liệu từ biến trở hoặc cảm biến nhiệt độ lên máy tính. Cùng chương trình này, bạn có thể thay thế cảm biến nhiệt độ hoặc biến trở, đồng thời thay đổi hòm bằng một cột chỉ thị nhiệt độ như **hình 10.16.** để tiện quan sát.

### Thảo luận về data flow, kiểu dữ liệu và kết nối phần cứng

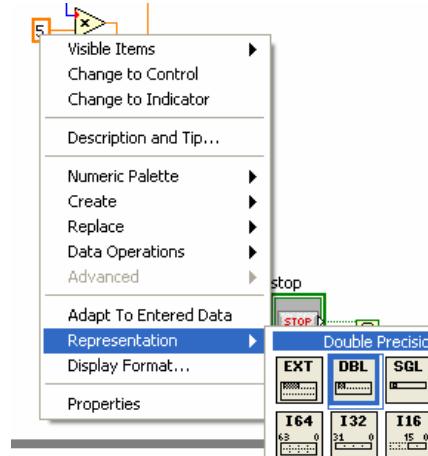
Trước khi kết thúc phần này, chúng ta hãy cùng thảo luận về data flow (dòng dữ liệu), và kiểu dữ liệu trong LabVIEW.

**Data flow** LabVIEW thực thi các lệnh từ trái qua phải. Để quan sát điều này, bạn chọn hình bóng đèn bên cạnh nút Pause, sau đó chạy chương trình. Bạn sẽ thấy các “hạt” tín hiệu chạy từ trái qua phải.



**Kiểu dữ liệu** là công cụ để gán “dải giá trị” cho một biến nào đó. Có nhiều loại kiểu dữ liệu với số bit khác nhau. Số bit càng cao thì biến gán với kiểu dữ liệu đó càng có giá trị lớn. Đôi khi, cần phải thay đổi kiểu dữ liệu để giá trị đo được hiển thị ra một cách chính xác.

Lưu ý: Trong LabVIEW, dây hoặc control, hoặc indicator màu xanh thì có giá trị là Unsigned 8 bit. Tức giá trị tối đa của các khối này là 255).



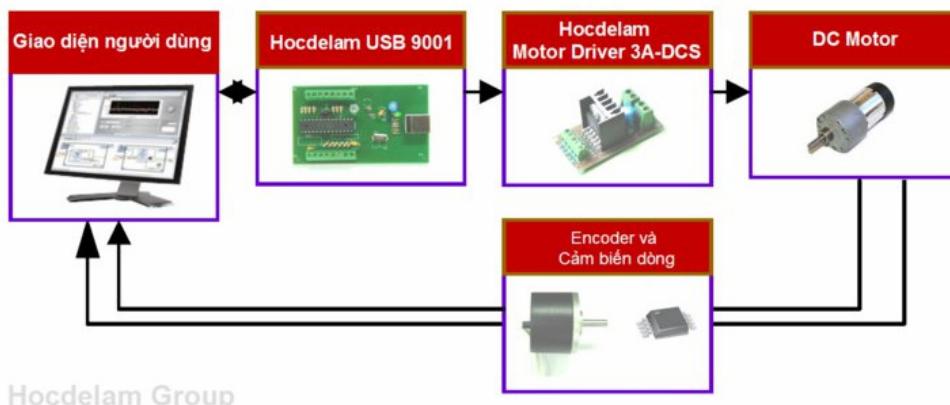
**Lưu ý khi làm việc với phần cứng trong LabVIEW** là trước khi khởi động phần mềm LabVIEW, ta phải cắm card USB vào máy. Nếu LabVIEW đang được mở và card chưa cắm thì ta tắt LabVIEW đi và cắm card vào, sau đó khởi động LabVIEW lên. Nếu đã làm đúng hết các bước mà bạn không giao tiếp được phần cứng để đọc tín hiệu từ cảm biến thì hãy liên lạc [supports@hocdelam.org](mailto:supports@hocdelam.org) hoặc [labview.help@gmail.com](mailto:labview.help@gmail.com) để được hỗ trợ.

### 10.3 Điều khiển PID động cơ DC

#### 10.3.1 Lý thuyết điều khiển PID

Có thể nói bộ điều khiển PID (viết tắt của: Proportional–Integral–Derivative Controller) là một trong những bộ điều khiển phổ biến và quan trọng nhất trong các thiết bị và hệ thống công nghiệp từ ở đĩa CD tới vận tốc xe ô tô đều được thực hiện bởi các thuật toán PID. Tài liệu này đó dùng điều khiển các hệ thống vật lý như động cơ DC (Xem hình 10.9), hệ thống lái tự động trên robot, ô tô, lò nhiệt, vv.

Điều khiển động cơ một chiều theo thuật toán PID

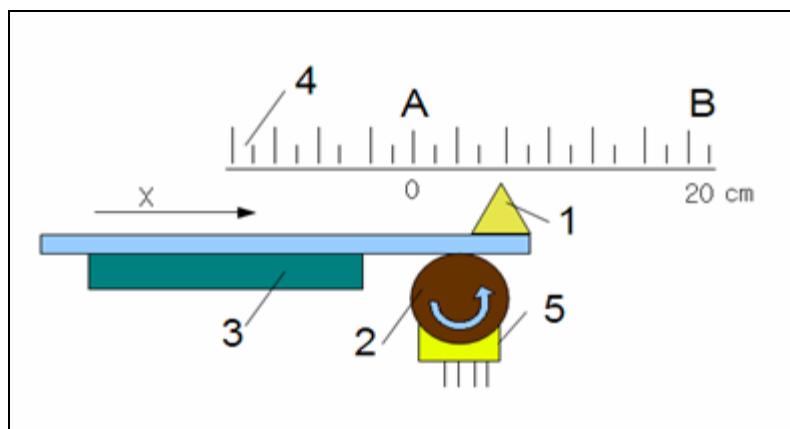


Hình 10.9 Sơ đồ điều khiển động cơ DC theo thuật toán PID

**Điều khiển PID là gì?** Bộ điều khiển PID (Proportional–Integral–Derivative Controller) là một **bộ hiệu chỉnh có phản hồi** nhằm làm **giá trị sai lệch** của một tín hiệu đang được điều khiển **bằng không**. Bộ PID có ba thành phần: proportional - tỷ lệ, integral - tích phân, và derivative - đạo hàm), ba thành phần này đều có vai trò đưa sai lệch về không. Tính chất tác động của mỗi thành phần có đặc điểm riêng được khảo sát chi tiết trong phần sau. Tín hiệu phản hồi (feedback signal) thường là tín hiệu thực được đo bằng cảm biến. **Giá trị sai lệch** là hiệu của tín hiệu đặt (setpoint) trừ cho tín hiệu phản hồi.

- PID là bộ điều khiển thông dụng nhất trong công nghiệp vì tính dễ áp dụng, và mang lại chất lượng điều khiển ổn định cho hệ thống. Cụ thể, bộ điều khiển PID thường sử dụng trong điều khiển động cơ DC, robot, các hệ thống trong ô tô, điều khiển áp suất, băng truyền, vv.

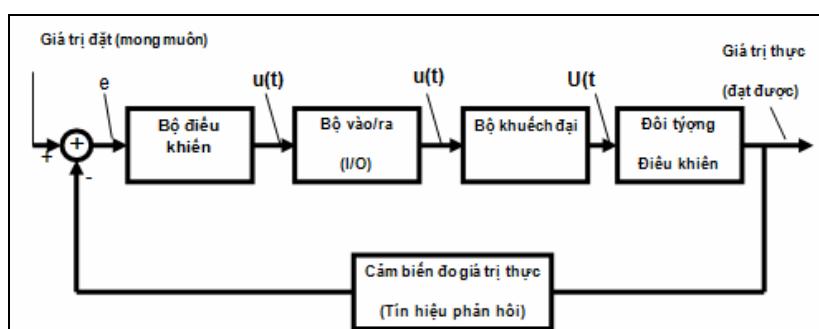
*Ví dụ: Bài điều khiển động cơ trong hình 10.9 được giả sử được dùng để điều khiển vị trí của đầu 1 gắn trên thanh kim loại trượt không ma sát trên bề mặt 3 để thanh di chuyển từ A đến B như **hình 10.10**.*



**Hình 10.10 Cơ cấu cần điều khiển vị trí**

### 10.3.2 Bản chất toán học của thuật toán PID

Sơ đồ của hệ thống trong hình 10.9 được vẽ lại thành hình 10.11.



**Hình 10.11 Bộ PID điều khiển vị trí**

Một bộ điều khiển PID có sơ đồ như hình trên. Trong bộ điều khiển PID, sai lệch  $e$  được tính bằng hiệu giá trị đặt hoặc điểm đặt (Set point  $\theta_s$ ) trừ cho giá trị thực té đo được (measured value của hệ thống  $\theta_m$ ).

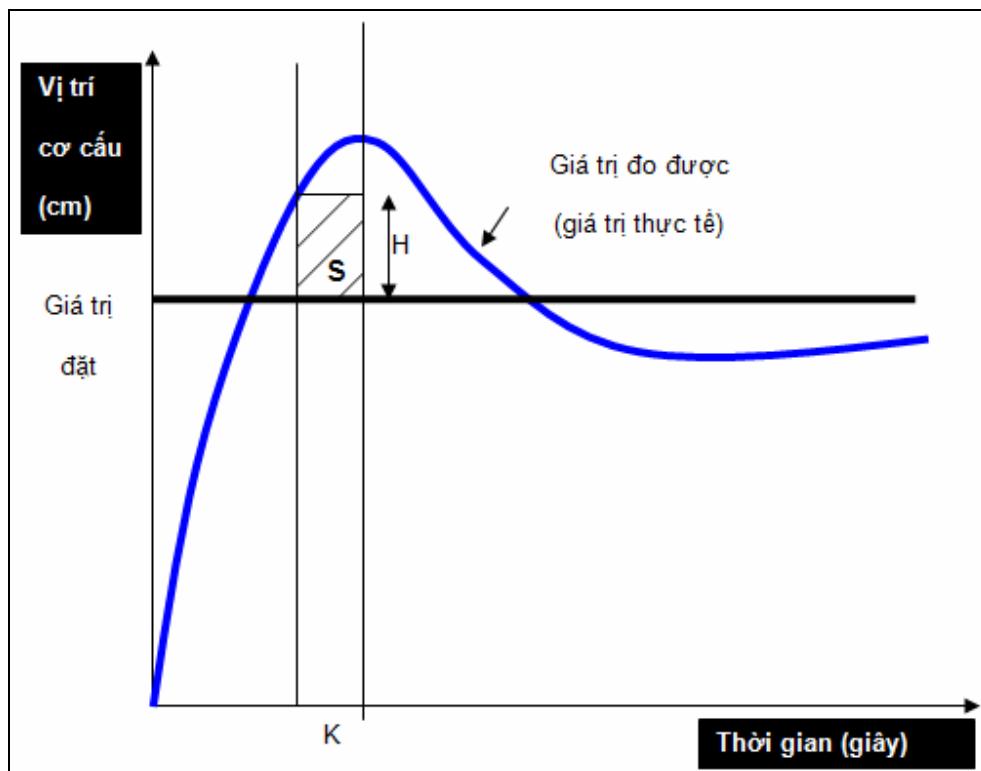
$$e = \theta_s - \theta_m \quad (10.1)$$

**Hoạt động của hệ thống điều khiển vị trí.** Bộ PID này sẽ đọc và hiểu giá trị mà người điều khiển mong muốn (gọi là giá trị đặt, ở đây là vị trí của **B** có tọa độ  $x_B=20\text{cm}$ ), thường người điều khiển đưa giá trị đặt vào bộ điều khiển PID thông qua GUI (Graphical user interface - giao diện người dùng đồ họa). Bộ điều khiển PID sẽ tính sai lệch  $e$ , và qua bộ PID thành tín hiệu điều khiển **u(t)**1 tính theo công thức 10.2, sai lệch sau khi tính toán được truyền ra ngoài hệ thống thực thông qua card vào/ra (I/O) như card Hocdelam USB-9001 hoặc NI 6009 ở đây tín hiệu lúc này là tín hiệu điện áp và được gọi là **u(t)2**. Sau đó, tín hiệu này được khuếch đại nhờ một bộ Driver (ví dụ Motor driver) để tăng tín hiệu đủ công suất điều khiển cơ cấu chấp hành (động cơ DC), gọi là tín hiệu **U(t)**. Tín hiệu điều khiển động cơ sẽ điều khiển cơ cấu 5, khi động cơ quay thì thanh kim loại trượt theo phương X và đầu 1 di chuyển dần từ A tới B. Hoàn thành một vòng điều khiển. Sau đó bộ điều khiển PID sẽ liên tục thực hiện lại việc tính toán sai lệch của vị trí đặt (vị trí B) so với giá trị vị trí thực tế (measured signal) của đầu 1 (nhờ vào bộ đo vị trí gắn với động cơ). Nếu giá trị sai lệch vẫn còn thì bộ điều khiển PID tiếp tục phát ra tín hiệu để quay độ động cơ cho tới khi giá trị thực tế của động cơ trùng khớp với giá trị đặt. Tức khi đó sai lệch sẽ bằng 0. Chừng nào còn sai lệch thì bộ điều khiển PID còn hoạt động để hiệu chỉnh tín hiệu điều khiển. Bản chất toán học của bộ PID sẽ được giải thích trong công thức 10.2. Giá trị tín hiệu đưa vào động cơ được tính là:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad (10.2)$$

- $u(t)$ : tín hiệu điều khiển, là tín hiệu do bộ PID sinh ra, (thường tín hiệu này đi qua một module công suất) và đi vào hệ thống (động cơ một chiều). Có đơn vị phụ thuộc phần cứng, như đối với điều khiển động cơ một chiều 24V thông qua module công suất là một Motor driver 24V thì  $u(t)$  có đơn vị là Volt.
- $K_p e(t)$ : Khâu tỷ lệ.
- $K_p, K_i, K_d$ : các hệ số tỷ lệ, tích phân và vi phân của bộ PID
- $e(t)$  sai lệch tại thời điểm hiện tại.
- $e(\tau)d(\tau)$ : Diện tích S tạo bởi đường cong **giá trị thực tế** (xem hình 1.21), đường thẳng **giá trị đặt**, và các cận là thời điểm trước và sau một vòng điều khiển (cũng chính là thời gian thực hiện hết một vòng lặp toàn bộ chương trình điều khiển **K** trong đồ thị 10.12) hay nói cách khác **K** chính là đơn vị thời gian trích mẫu  $d(\tau)$  (thường thời gian trích  $d(\tau)$  mẫu bé khoảng ms thậm chí nano giây).

Vậy tích phân từ  $0 \rightarrow t$  của sai lệch nhân với thời gian lấy mẫu (trong công thức 10.2) chính là tổng hợp các sai lệch từ khi hệ thống bắt đầu được điều khiển tới thời điểm hệ thống đang hoạt động hiện tại.



**Hình 10.12 Mô tả giá trị đặt, giá trị đo được và diện tích sai lệch**

- Các hệ số luôn không âm: Tức  $K_p, K_i, K_d \geq 0$
- $d[e(t)]/dt$ : Tốc độ thay đổi của sai lệch tại thời điểm hiện tại.

**Ý nghĩa các hệ số gain trong bộ PID khi tăng các hệ số  $K_p, K_i, K_d$  độc lập nhau**

- Với cùng một giá trị  $e(t)$ , nếu tăng  $K_p$ , thì tín hiệu điều khiển  $u(t)$  tăng.  $u(t)$  tăng đồng nghĩa điện áp tác động vào động cơ một chiều tăng lên, điện áp tăng sẽ làm giảm  $e(t)$  nhanh hơn có nghĩa hiệu giá trị đặt và giá trị đo được sẽ giảm nhanh hơn. Khi đó ta gọi khả năng đáp ứng của thông nhanh. Đáp ứng của hệ thống được xem là càng nhanh khi thời gian cần thiết để tín hiệu đầu ra của hệ thống đạt tới giá trị đặt càng nhỏ.

Tuy nhiên, khi  $K_p$  quá lớn, thì hệ thống không ổn định vì có hiện tượng giá trị đo được của hệ thống vượt quá giá trị đặt, gọi là **vọt lố** (overshoot).

- $K_i$  có tác dụng làm tăng tốc độ đáp ứng vì nó cũng làm tăng điện áp ( $U(t)$ ) đặt vào động cơ. Đồng thời khâu này làm **sai lệch tĩnh** (steady-state error) trở về 0 nhờ vào đặc tính cộng dồn sai lệch của phép toán tích phân. Sai lệch tĩnh là sai lệch sau khi tín hiệu đầu ra của hệ thống đã ổn định.
- $K_d$  Khâu D có tác dụng làm ổn định hệ thống. Khi  $e(t)$  đổi dấu tức giá trị phản hồi lớn hơn giá trị setpoint, thì khâu D nó có tác dụng làm giảm tín hiệu điều khiển  $u(t)$ .

Phương pháp định bộ thông số  $K_p, K_i, K_d$  (Các hệ số gain) thủ công thường dựa vào việc thử nghiệm các thông số. Ta xác lập các hệ số  $K_p, K_i, K_d$  bằng không. Sau đó tăng dần  $K_p$  cho tới khi vọt lố đạt gần 1,5 lần giá trị đặt. Hiệu chỉnh  $K_i$  cho sai lệch tĩnh bằng không. Và hiệu chỉnh  $K_d$  cho hệ thống giảm rung lắc.

Ngoài ra bạn còn có thể thực hiện việc chọn các hệ số này bằng phương pháp tìm các hệ số của Ziegler–Nichols như **bảng 10.1**.

**Bảng 10.1: Phương pháp chỉnh PID (Phương pháp Ziegler–Nichols)**

Dạng điều khiển	$K_p$	$K_i$	$K_d$
P	$0.50K_u$		
PI	$0.45K$	$1.2K_p / P_u$	
PID	$0.60K_u$	$2K_p / P_u$	$K_pP_u / 8$

- $K_u$ : Giá trị  $K_p$  làm cho hệ thống mất ổn định 150%.
- $P_u$ : Khoảng thời gian dao động của tín hiệu đo được.

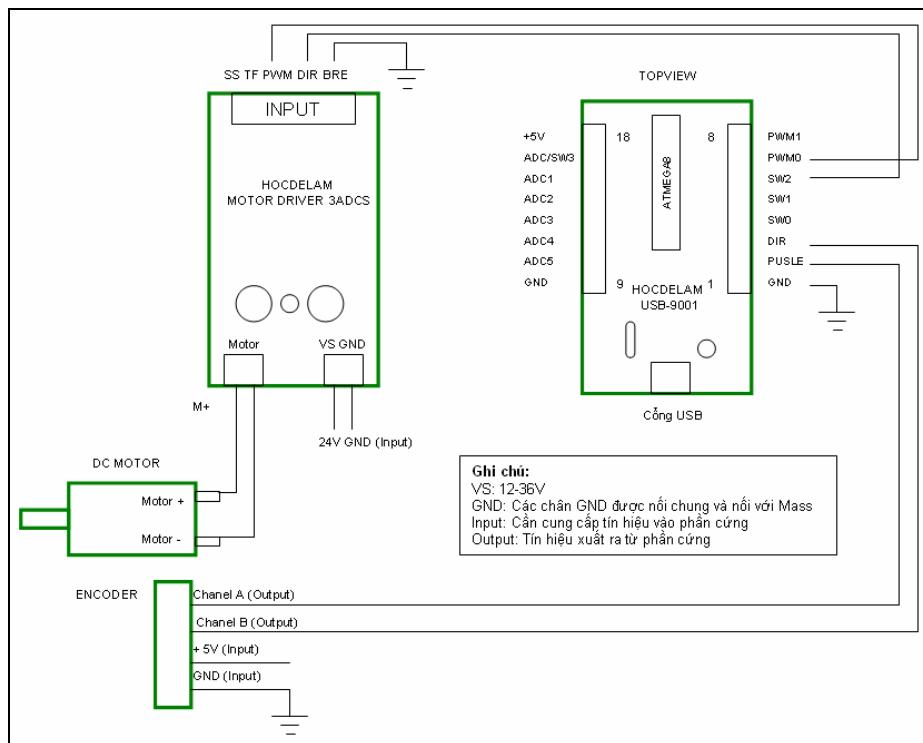
Cao hơn phương pháp xác định các hệ số bằng tay thì còn phương pháp xác định bộ gain  $K_p, K_i$ , và  $K_d$  tự động sẽ được cập nhật ở các tập sách tiếp theo cùng tác giả.

### 10.3.4 Thực hành điều khiển PID cho động cơ DC

Để thực hành bộ điều khiển PID, chúng tôi chọn card Hocdelam USB 9001, phần cứng hoàn thiện như **hình 10.13** và sơ đồ mạch điện **hình 10.14**.

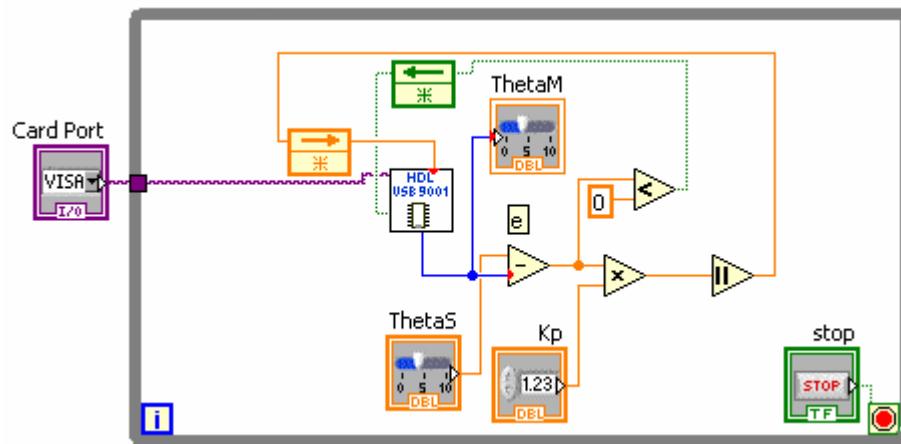


**Hình 10.13 Sơ đồ kết nối phần cứng điều khiển PID động cơ DC**

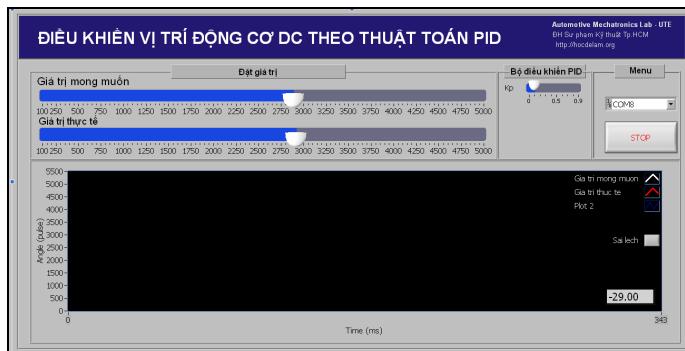


**Hình 10.14 Sơ đồ mạch điện kết nối phần cứng điều khiển PID động cơ DC**

Thuật toán điều khiển động cơ DC trong LabVIEW được lưu dưới dạng File CD>Bai 10> DC motor P control.vi

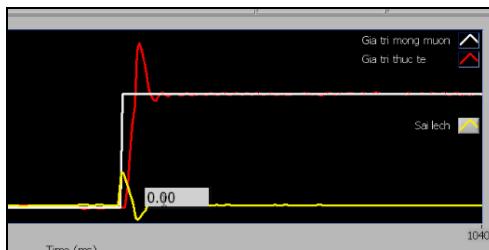


Hình 10.23 Kết quả lập trình P control cho động cơ DC



Hình 10.16 Giao diện người dùng điều khiển P cho động cơ DC

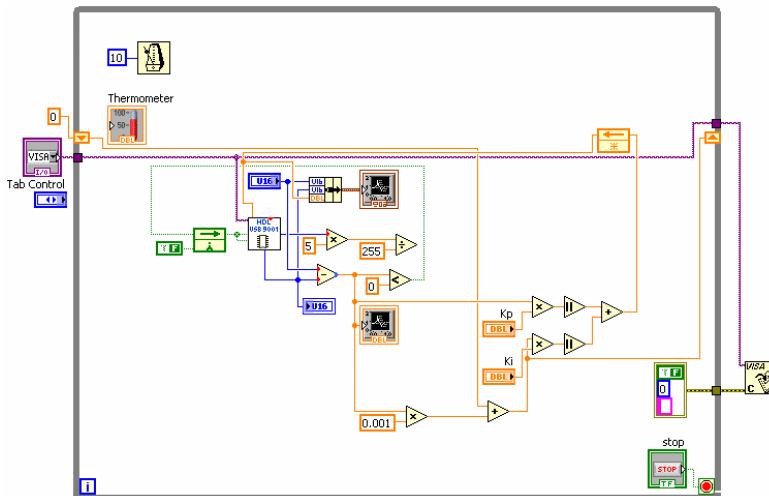
Đáp ứng của vị trí động cơ DC theo thuật toán P khá tốt. Thời gian đáp ứng là 0.2 giây với  $K_p = 0.562$ . Sai lệch tiến tối không sau 0.5 giây. Xem **hình 10.17**. Đường màu trắng là giá trị đặt, đường màu đỏ là giá trị đáp ứng (giá trị đo được từ Encoder), màu vàng là sai lệch.



Hình 10.17 Đáp ứng của vị trí động cơ DC

Do giới hạn cuốn sách này chỉ đề cập các vấn đề cơ bản nhất trong LabVIEW nên để mở rộng và tự lập trình được bài toán PID, bạn hãy đọc bài “*Điều khiển động cơ DC theo thuật toán PID cho mọi người, TS. Nguyễn Bá Hải*” tại website LabVIEW Hocdelam theo đường dẫn : <http://labview.hocdelam.org>> Nghiên cứu> Báo khoa học

Qua CD kèm theo sách, tác giả cũng tặng ví dụ mẫu điều khiển PI cho các bạn tự tìm hiểu và khám phá. BD của bài toán điều khiển PI được cho ở hình 10.26. Xem ví dụ mẫu tại **CD> Bai 10> Vi du Hocdelam-9000 PI.vi**



**Hình 10.26 Điều khiển khâu PI vị trí động cơ DC**

### Ban có biết?

Xem video demo về điều khiển PID trong CD> Demo> DC Motor Control. Bạn có thể liên lạc supports@hocdelam.org hoặc hotline: 0937 159 700 để đăng ký thực miễn phí hành bài toán điều khiển động cơ DC theo thuật toán PID tại phòng thí nghiệm của Hocdelam Group (tại Thủ Đức, Quận 6, và Hà Nội). Do lịch đăng ký thực hành khá đông, nên bạn sẽ phải chờ theo thứ tự đăng ký. Thông thường, bạn sẽ được xếp lịch sau 4 ngày tính từ khi đăng ký.

### Bài tập:

**Câu 1:** Thực hành đo lường cảm biến nhiệt độ LM35 vào máy tính và vẽ đồ thị nhiệt độ dựa vào card Hocdelam USB-9001.

**Câu 2:** Thu thập tín hiệu từ một Encoder vào máy tính qua card Hocdelam USB-9001. (Gợi ý: Xem ví dụ mẫu **CD> Bai 10> Doc encoder.vi**)

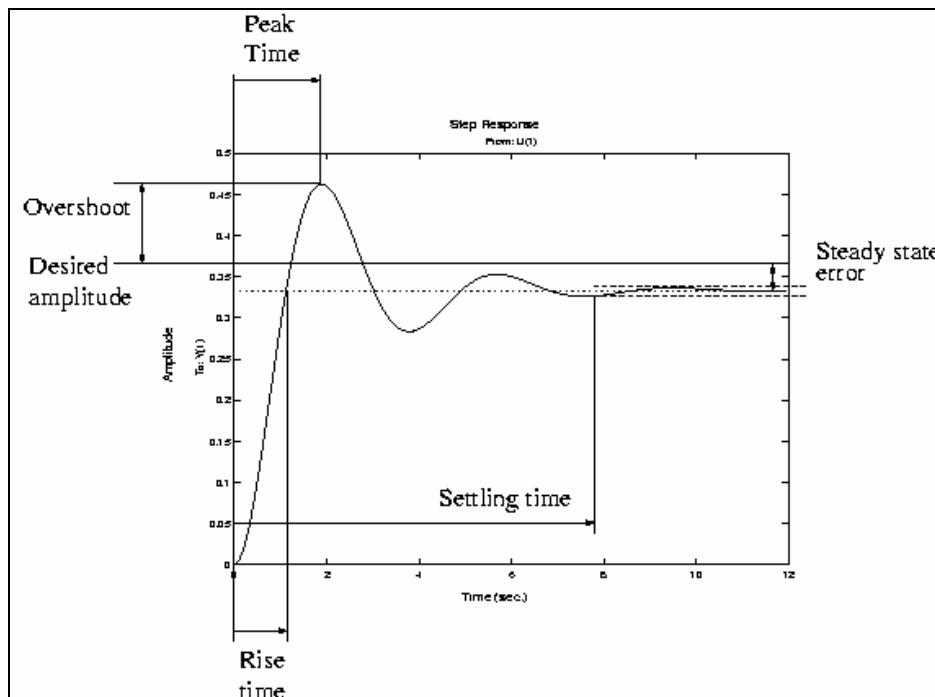
**Câu 3:** Thực hành điều khiển khâu P cho động cơ DC (vị trí và vận tốc).

**Câu 4:** Hãy mô tả hoạt động của bộ điều khiển PID và cách chọn các hệ số trong bộ điều khiển.

**Câu 5:** Thực hành lập trình điều khiển khâu PI vị trí động CD

**Câu 6:** Hãy giải thích tại sao khâu I trong bộ điều khiển PID có thể khử được sai lệch tĩnh của hệ thống?

**Câu 7:** Bạn cần nắm một số thuật ngữ tiếng Anh trong điều khiển tự động trong hình sau (hãy sử dụng từ điển để tra từ).



(Nguồn hình: Wikipedia)

#### Tham khảo bài 10

[1] Wikipedia.org

[2] Controls.ame.nd.edu/ame437/S2003/hw1

[3] TS. Nguyễn Bá Hải, “Điều khiển PID vị trí động cơ DC với LabVIEW”, Hocdelam Group, 2009.

[4] TS. Nguyễn Bá Hải, Thạc Sĩ Ngô Hải Bắc, PGS. TS. Jee-Hwan Ryu “Điều khiển bền vững động cơ DC” Hội thảo khoa học quốc tế URAI, Seoul, Hàn Quốc.

## BÀI 12

# ĐIỀU KHIỂN VỊ TRÍ VÀ VẬN TỐC CHẤT LƯỢNG CAO VỚI CARD PCI 7356

Học xong bài này, bạn sẽ nắm được:

- *Thông số kỹ thuật card PCI 7356*
- *Lập trình với card PCI 7356*
- *Xử lý các lỗi thông dụng*

Trong tài liệu này, tác giả cung cấp thông tin cài đặt phần mềm và phần cứng cho điều khiển chuyển động dùng PCI motion controller NI PCI-73xx hoặc PXI-7356, phương pháp kiểm tra tình trạng làm việc, và hướng dẫn cách lập trình card trước khi tiến hành lập trình các ứng dụng cho card NI PCI-73xx. Tài liệu này cũng có thể dùng tham khảo cho các card PCI DAQ như NI PCI-6010, NI PCI-6220, NI PCI-6503, NI PCI-6601, NI PCI-6610, NI PCI-4472B, NI PCI-6221, , NI PCI-62xx , PCI signal generator (NI-FGEN Signal Generator) như NI PCI-5412, NI PCI-5421, NI PCI-54xx và một số card PCI của advantech. vv. Các thông tin, tài liệu mới nhất về LabVIEW và ứng dụng các card này được cập nhật tại: <http://labview.hocdelam.org>.

### 12.1 Chuẩn bị thiết bị

Để sử dụng card này bạn cần có phần cứng và phần mềm đi kèm gồm:

#### Phần cứng

- Card PCI 73xx (ví dụ PCI 7356), là bộ điều khiển.
- Connector (hoặc giao diện đa năng UMI, 68 chân), dùng cắm dây ra động cơ và encoder.
- Cable nối card ra connector, chuẩn 68 chân.
- Động cơ một chiều DC hoặc động cơ bước, encoder có xung ra ở mức điện áp 0volt (mức thấp) và 5volt (mức cao).
- Máy tính để gắn card.

#### Phần mềm

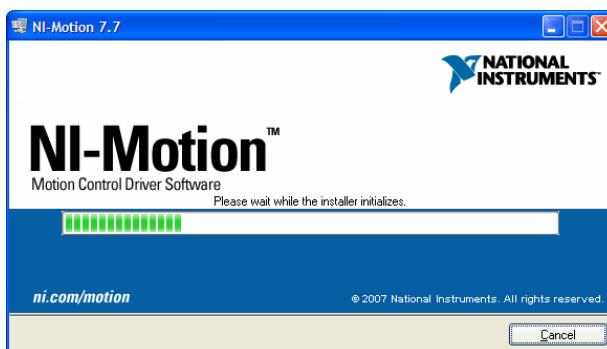
- NI motion driver (để máy tính nhận ra card).
- LabVIEW từ bản 7.1 tới 2009 (để viết chương trình phát triển ứng dụng cho card NI PCI73xx).
- Có thể thay LabVIEW bằng LabWindows/CVI hoặc Visual C++, hoặc Microsoft Visual Basic.

## 12.2 Cài đặt phần mềm

Hệ thống lập trình chuyển động sử dụng card NI PCI và động cơ DC gồm các bước sau:

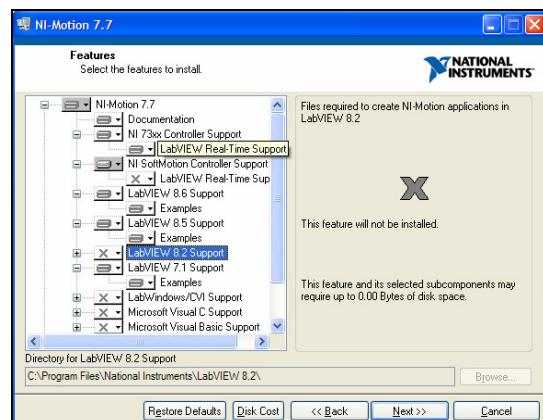
### 12.2.1 Cài đặt NI motion driver (Tải tại <http://labview.hocdelam.org>).

Mục tài liệu > **Driver**, toolkits. (Lưu ý, trong lần cài đặt lần đầu tiên, việc cài đặt driver luôn phải thực hiện trước khi gắn card vào máy tính).



*Hình 12.1 Cài đặt NI Motion*

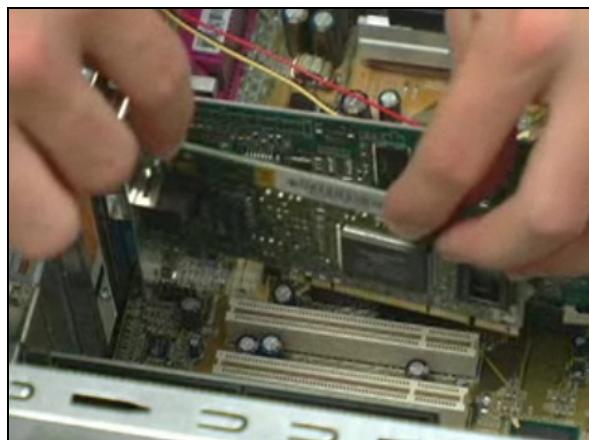
Khi chọn các đặc tính cài đặt, nên chọn các phiên bản LabVIEW khác nhau ví dụ **LabVIEW 7.1 support**, **LabVIEW 8.5 support**, **LabVIEW 8.6 support**. Việc này cho phép sử dụng card PCI 73xx này với bất cả ba phiên bản LabVIEW này. Chọn **LabVIEW Real-time Support**.



*Hình 12.2 Chọn LabVIEW Real-time Support*

### 12.2.2 Phần card NI PCI motion controller (ví dụ NI PCI 7356)

Gắn card NI PCI vào khe PCI của máy tính như hình dưới.



*Hình 12.3 Cài đặt phần cứng card vào máy tính*

#### 12.2.3 Gắn cable nối card NI motion controller vào khối connector (ví dụ UMI 7764-).

Chú ý: Card PCI motion controller của NI có thẻ điều khiển từ 4 tới 8 trục (tương đương 8 động cơ) do vậy bạn cần lưu ý khi gắn cable. Thường nếu dùng một tới bốn động cơ thì **gắn port motion 1-4**. Sau đó nối cable này tới card UNI (Universal Motion Interface).



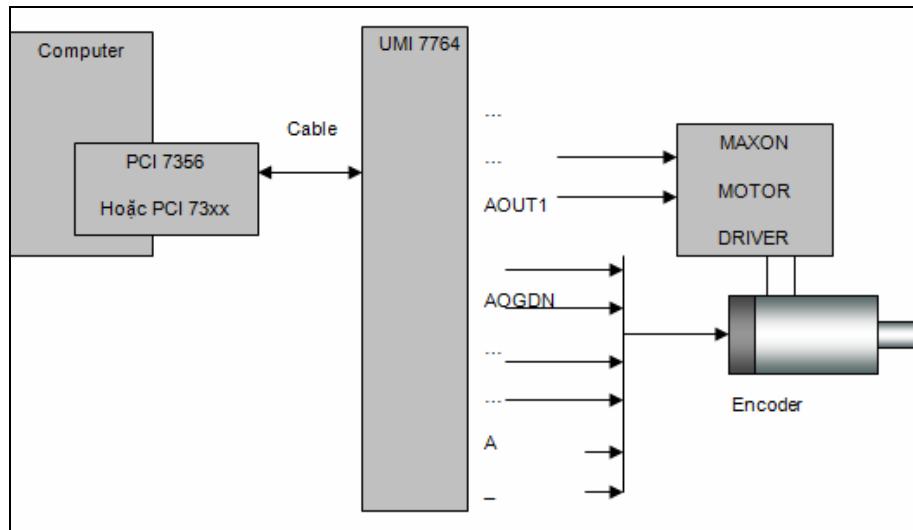
*Hình 12.4 Sơ đồ kết nối vào UMI*

#### 12.2.4 Gắn động cơ và encoder

Dựa vào tên các chân của UMI để lần lượt gắn dây điều khiển động cơ từ card tới motor driver. Có motor driver dùng phương pháp analog control, tức xung PWM sẽ được sinh ra từ driver khi driver đó được cung cấp điện áp từ 0-5 volts. Do vậy, trong trường hợp này ta sẽ nối chân DAC OUT và DAC GDN để gắn với hai dây nguồn động cơ.

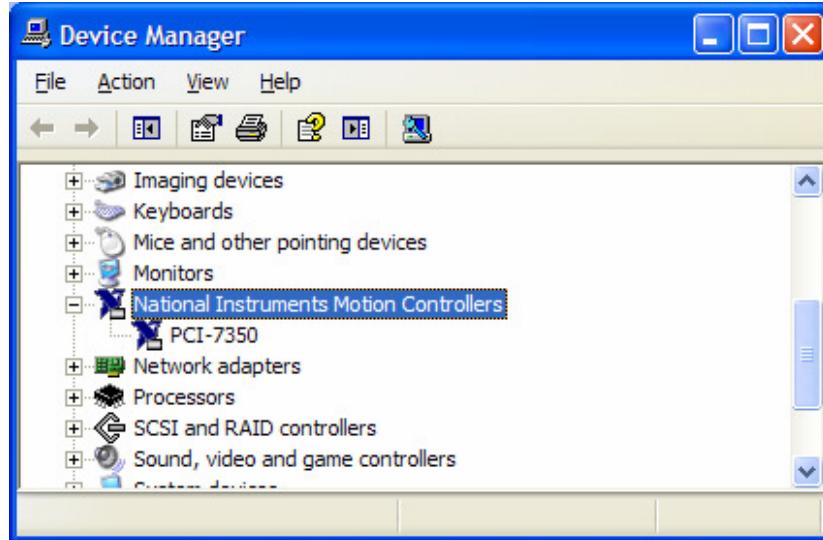
Cần nối các dây của Encoder vào UMI theo ký hiệu. (Có thể dùng hai dây A,B hoặc A,B,Z hoặc cả 6 dây từ encoder).

### 12.2.5 Kết quả kết nối phần cứng



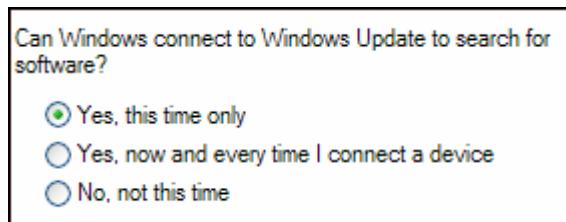
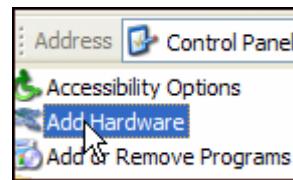
*Hình 12.5 Sơ đồ kết nối động cơ và encoder*

Kiểm tra việc nhận card của máy tính bằng cách chọn chuột phải lên **My computer > Properties > Hardware > Device Manager**, sẽ thấy card như hình dưới.

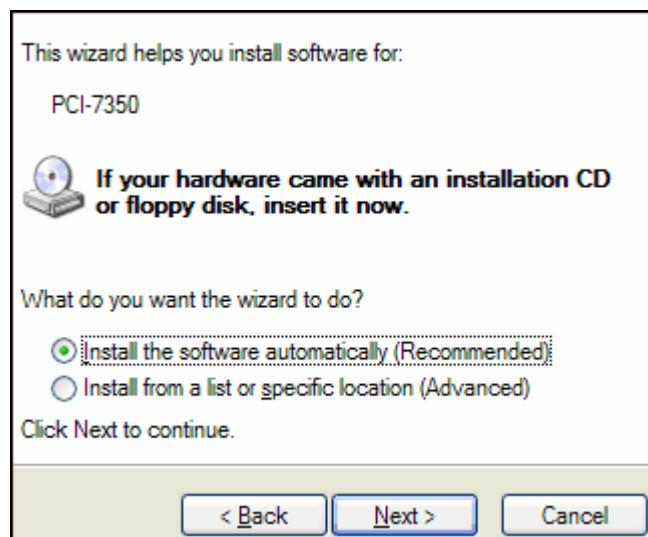


*Hình 12.6 Card đã được nhận vào máy tính*

Sau khi cài đặt, nếu máy tính không nhận ra card thì chọn vào **Control panel > Add and hardware**, chọn **Next** và làm theo hướng dẫn.



*Hình 12.7 Cập nhật card vào hệ điều hành*



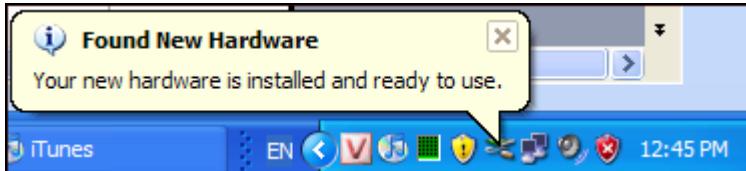
*Hình 12.8 Chọn Recommended*

Lặp lại bước trên thì máy tính sẽ được card theo hình với thông báo **Completing the Found New Hardware Wizard** (tức việc cài đặt hoàn thành).



*Hình 12.9 Hoàn thành cài đặt*

Đồng thời góc phải bên dưới màn hình của bạn hiện lên thông báo **Your new hardware is installed and ready to use** (card đã được cài đặt và sẵn sàng cho bạn sử dụng).

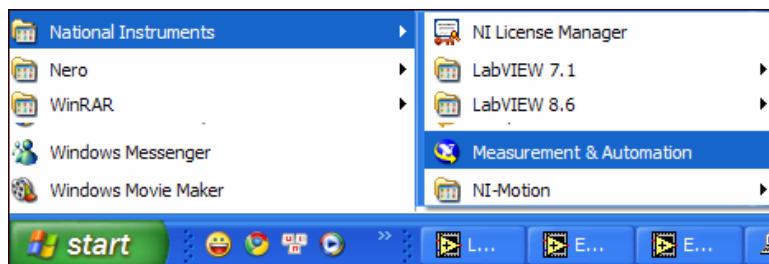


*Hình 12.10 Windows báo đã hoàn tất việc cài đặt*

### 12.3 Thủ phần cứng và phần mềm

Trước khi tiến hành lập trình, cần phải thử lại xem việc kết nối phần cứng và phần mềm đã đúng chưa? Bằng cách:

- Khởi động MAX (vào Start program > National Instrument > Measurement & Automation)



*Hình 12.11 Khởi động MAX*

- Sẽ thấy tên card hiện ra theo đường dẫn sau: My system > Devices & Interface > PCI-7350(1).

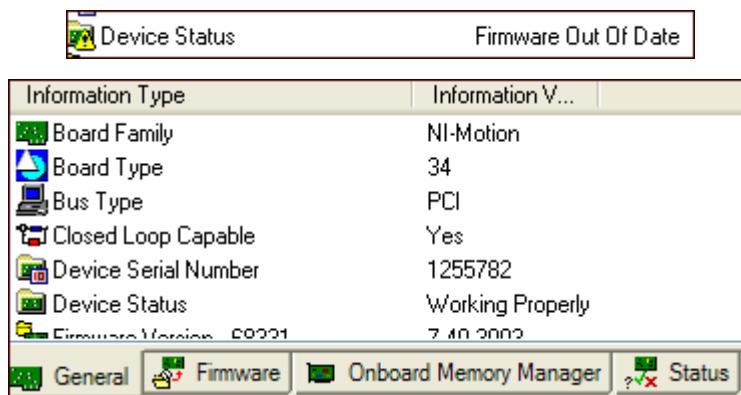
Có hai trường hợp có thể xảy ra:  
Card còn tốt nhưng chưa được cập nhật firmware thì sẽ thấy dấu chấm than này , nếu card bị hỏng thì sẽ thấy dấu chéo nền tròn đỏ này . Có thể xem thêm tại Tab General để biết rõ tình trạng của card còn tốt hay đã hỏng.

Information Type	Information Value
Board Family	NI-Motion
Board Type	34
Bus Type	PCI
Closed Loop Capable	Yes
Device Serial Number	12CB623
Device Status	Device is not responding or has been removed from the system.
Number of ADCs	8
Number of Axes	6
Number of DACs	6
Number of DIO	64
Number of Encoders	8
Number of Steppers	6

*Hình 12.12 Giao diện trong MAX*

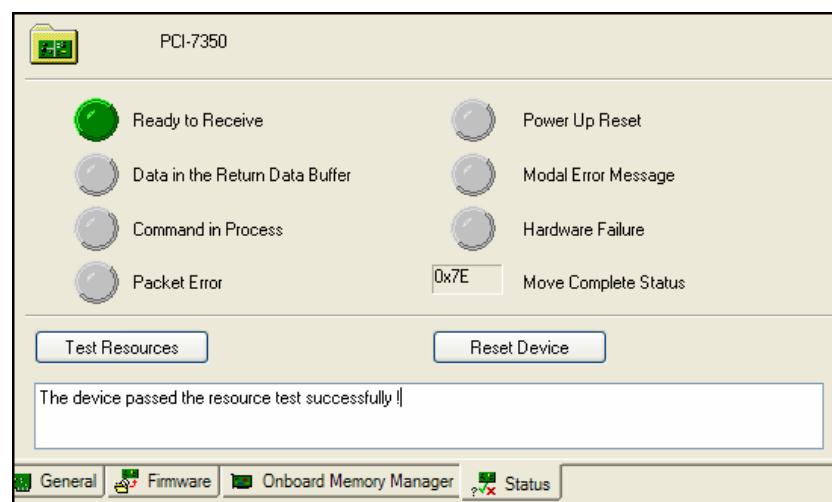
Bạn sẽ thấy dòng chữ Device is not responding or has been removed from the system khi máy tính chưa nhận được card hoặc card bị hỏng. Nếu Device status là Firmware out of date thì cần cập nhật (update) firmware.

Nếu card đã được cập nhật firmware và hoạt động bình thường thì sẽ thấy dòng chữ **The controller is in a reset state** hoặc **working property**.



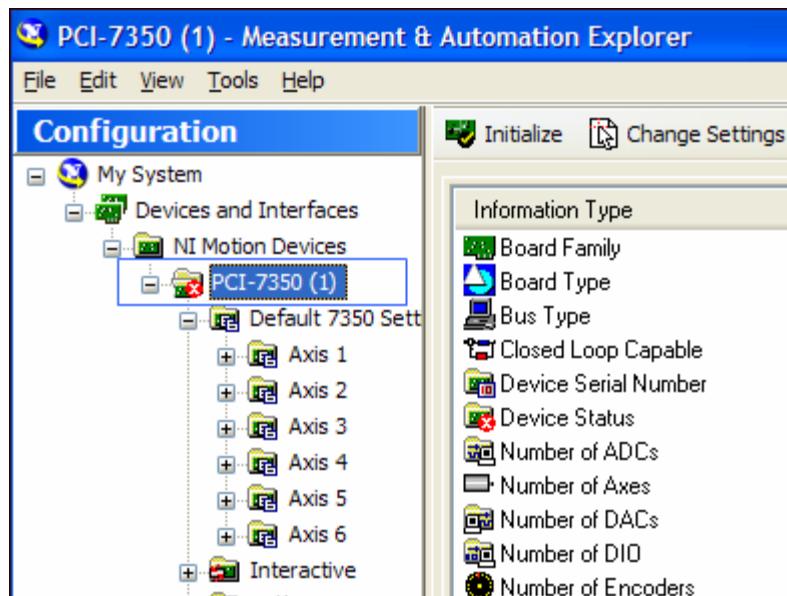
*Hình 12.13 Cập nhật thông tin*

Và tại Tab status sẽ là Ready to receive.

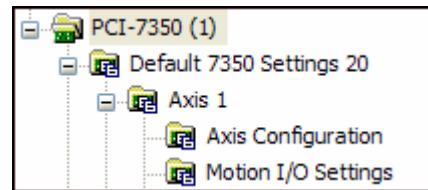


*Hình 12.14 Báo hiệu đã kiểm duyệt thành công*

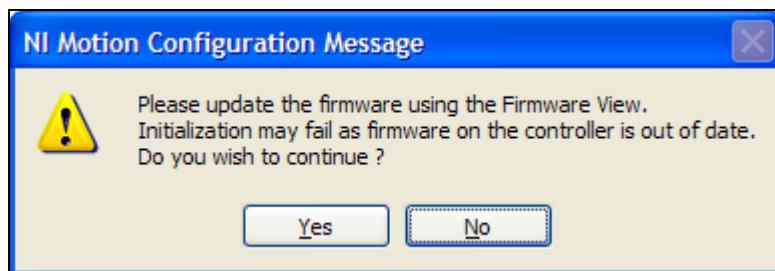
- Nếu thấy dấu nhân trăng trong vòng tròn nền đỏ  (hoặc dấu chấm than ) thì kiểm tra lại xem máy tính đã chắc chắn nhận được card trong **My computer > Properties**.

**Hình 12.15 Dấu chấm than**

- Nếu vẫn còn dấu này chứng tỏ card cần phải initialize (khởi tạo) trước khi dùng. Để khởi tạo, chọn Initialize. Sau khi khởi tạo hoàn thành, dấu nhân báo lỗi tại PCI-7350 sẽ mất như hình bên dưới.

**Hình 12.16 Hết dấu chấm than**

- Khi initialize, nếu có thông báo cần phải **cập nhật Firmware**, thì chọn Yes. Cập nhật xong, chọn Initilaize lại và thực hiện tiếp bước 12.4.

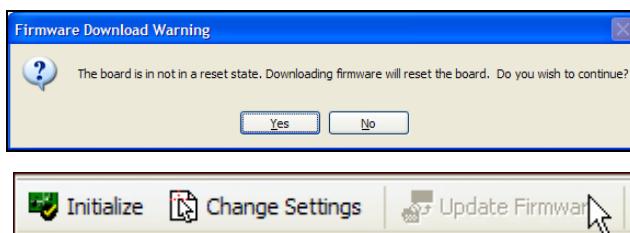
**Hình 12.17 Cập nhật firmware**

Sau khi cập nhật firmware bạn sẽ nhận được thông báo **Firmware was downloaded successfully**.



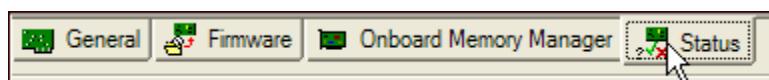
**Hình 12.18 Download firmware thành công**

Việc cập nhật firmware cũng có thể thực hiện bằng cách vào Tab firmware và chọn Update firmware (nếu chữ update firmware mờ đi thì tức là không cần phải update firmware nữa).



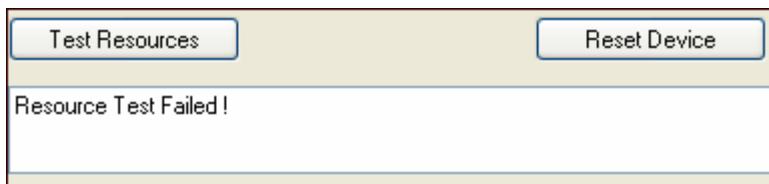
**Hình 12.19 Đổi ID**

- Trường hợp card báo lỗi không cho cập nhật thì vào tab Status.



**Hình 12.20 Kiểm tra status**

Chọn Test resource, Nếu thông báo Resource test failed thì chọn Reset device và làm lại. Sau đó làm lại các bước vừa nêu trong bước 12.3 này.

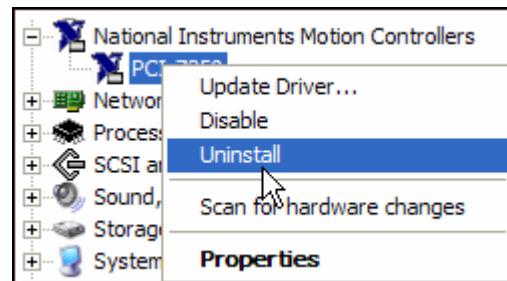


**Hình 12.21 Test báo lỗi**

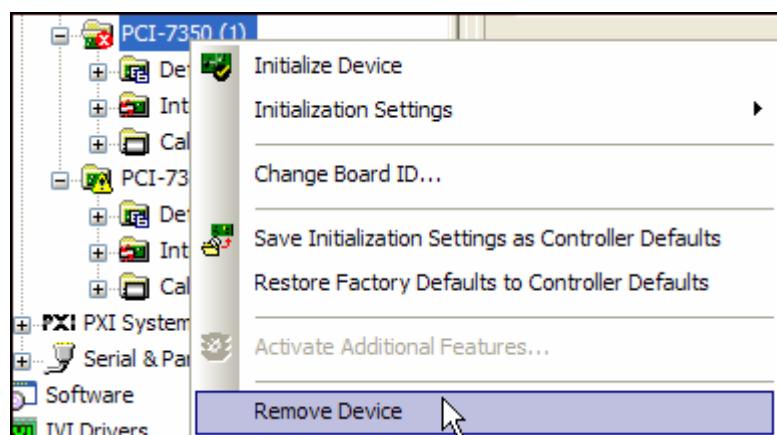
Trước khi tới bước 12.4, bạn cần kiểm tra lại chắc chắn rằng **kết quả Test resources** phải là: **The device passed the resource test successfully** mới đạt yêu cầu.

**Hình 12.22 Test thành công**

- Nếu vẫn có lỗi bạn cần remove card từ **My Computer > Properties**.

**Hình 12.23 Gỡ phần mềm**

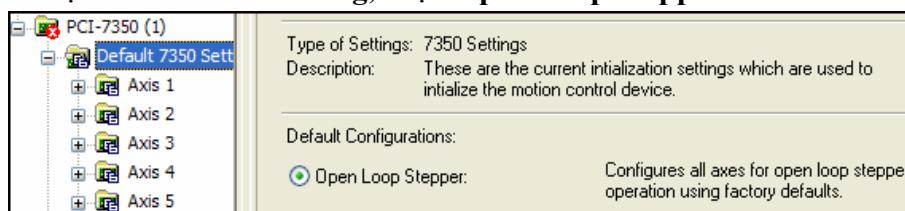
Và remove trong MAX bằng cách nhấn chuột phải lên **PCI-7350**, chọn **Remove Device**.

**Hình 12.24 Gỡ thiết bị**

Sau đó vào **Control panel** thực hiện lại việc **add new hardware**.

#### 12.4 Thiết lập các thông số trong MAX

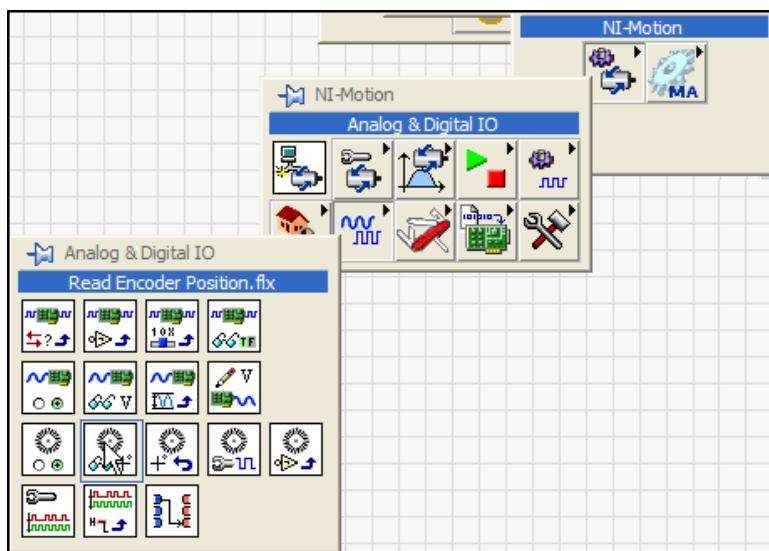
Tại Default 7350 setting, chọn **Open Loop Stepper**

**Hình 12.25 Thủ Đefault**

**Axis1 > Axis configuration:** Chọn stepper motor, và enabled Encoder, type of feedback là Encoder. Sau đó save to all axes (nếu có nhiều động cơ). Sau đó chọn Initialize lần nữa để chuyển qua bước 12.5.

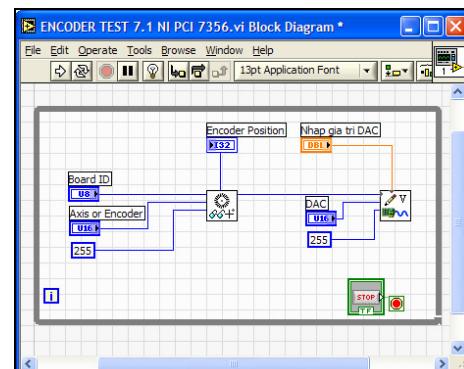
### 12.5. Thử động cơ và chương trình điều khiển động cơ một chiều DC bằng Analog output

Dùng chương trình xuất DAC trong thư viện của Motion. Thư viện này nằm trong **Block diagram**.



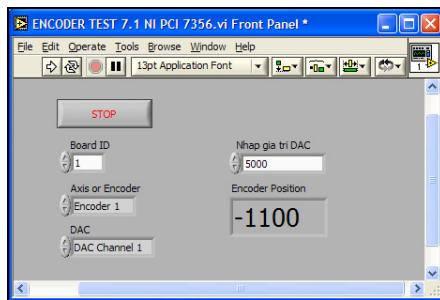
Hình 12.26 Thư viện lập trình chuyển động

Viết chương trình đơn giản như sau. (Tải chương trình này tại).



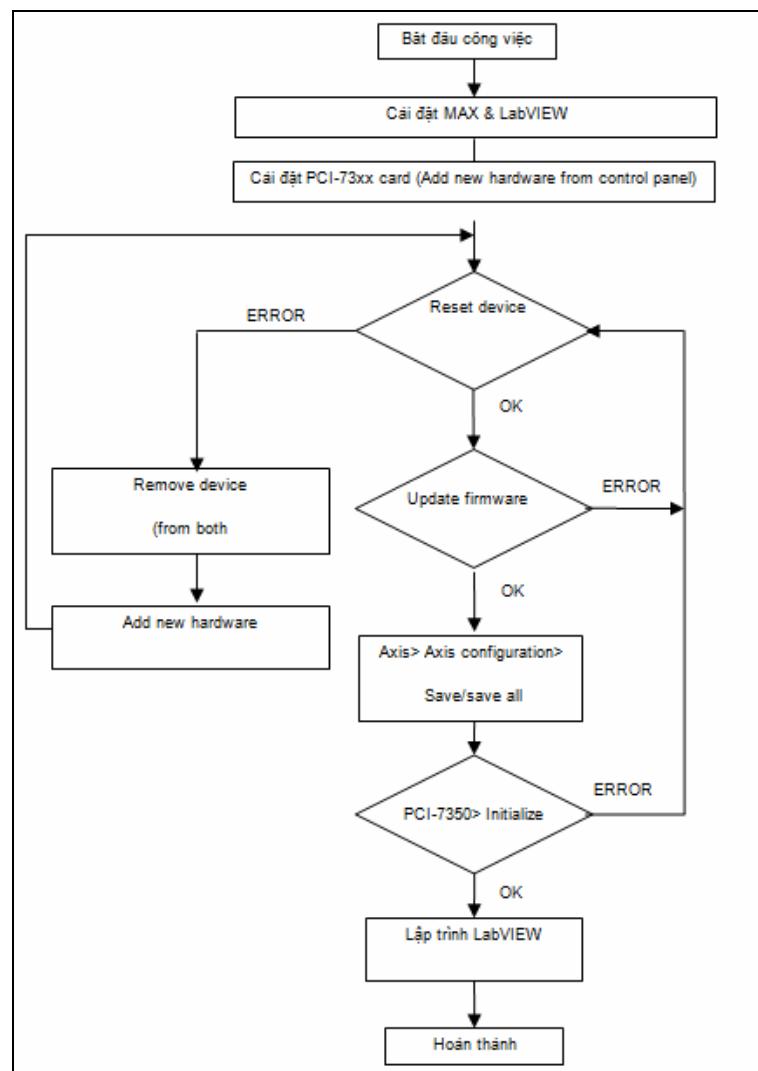
Hình 12.27 Chương trình hoàn thiện xuất DAC

Bạn sẽ thấy động cơ quay khi nhập giá trị vào “Nhập giá trị DAC”, (ví dụ nhập 5000). Đồng thời bạn sẽ đọc được tín hiệu encoder. (xem hình front panel bên dưới, dấu trừ nói lên chiều quay của động cơ).



Hình 12.28 FP của chương trình

### Tóm tắt các bước cài đặt và xử lý lỗi cho card PCI-73xx



Hình 12.29 Lưu đồ xử lý lỗi

## 12.6 Điều khiển chuyển động

### 12.6.1 Điều khiển chuyển động là gì?

Điều khiển chuyển động là việc sử dụng phần mềm (ví dụ ngôn ngữ lập trình G trong LabVIEW) để điều khiển phần cứng (ví dụ: động cơ) theo một chiến lược (strategy) nào đó. Chiến lược đó thường gồm các yêu cầu về tọa độ (vị trí) của trực động cơ (hoặc vật gắn lên trực động cơ), thời gian tăng tốc, giảm tốc và vận tốc chuyển động, vv. Ứng dụng của điều khiển chuyển động thường gặp là máy cắt điều khiển bằng máy tính CNC, hoặc tay robot hàn di chuyển theo đường hàn giữa hai tấm kim loại, robot di động đi theo một sơ đồ đường đi được định trước. vv.

### 12.6.2 Công cụ trong điều khiển chuyển động

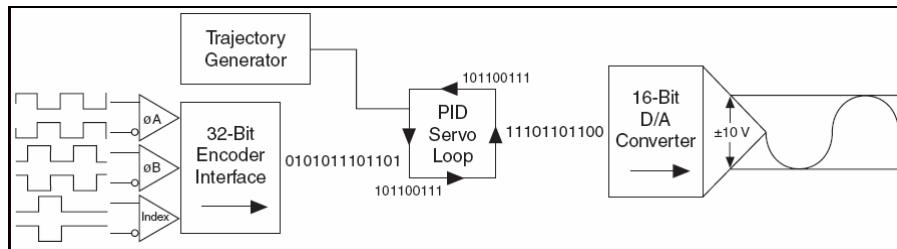
Có thể sử dụng các phần mềm sau:

- Motion assistant, cho phép tạo ứng dụng điều khiển chuyển động rất nhanh. Người thiết kế chỉ cần vẽ các quỹ đạo (đường di chuyển) mà hệ thống cần đi theo. Sau đó công cụ này tự động xuất ra mã chương trình (code) dưới dạng ngôn ngữ C/C++, Visual Basic hoặc LabVIEW.
- Visual C/C++, cho phép viết code dùng ngôn ngữ C để điều khiển card PCI 73xx.
- Visual Basic, cho phép viết code trong Visual Basic C để điều khiển card PCI 73xx.
- LabVIEW, là ngôn ngữ đồ họa, dễ sử dụng, và là ngôn ngữ tốt nhất khi dùng card NI PCI 73xx nói riêng cũng như các phần cứng do công ty NI sản xuất và cung cấp. Trong tài liệu này LabVIEW được sử dụng để tạo các ứng dụng.

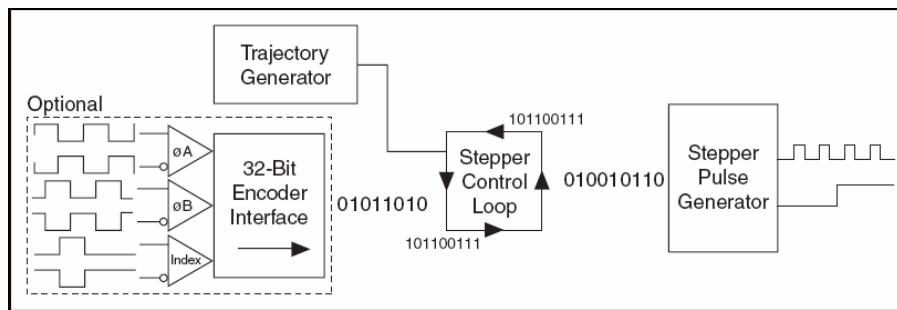
### 12.6.3 Sử dụng LabVIEW trong điều khiển chuyển động

#### Tổng quan về điều khiển chuyển động

Điều khiển động cơ thường theo hai sơ đồ dưới đây. Tức thiết lập yêu cầu quỹ đạo và chiến lược chuyển động (Trajectory), sau đó đọc tín hiệu phản hồi (Encoder), qua bộ điều khiển (PID) tới bộ chuyển đổi DAC để xuất ra điện áp hoặc bộ PWM generator để phát xung rồi cấp xung hoặc điện áp đó tới bộ khuỷu chêch đại (motor driver) và tới động cơ. Sơ đồ điều khiển động cơ Servo hoặc động cơ DC cho trong hình sau:

**Hình 12.30 Điều khiển servo motor**

Với động cơ bước thì việc điều khiển cũng tương tự. Khác là thay vì xung PWM hoặc analog output chuyển thành xung bước. (Stepper pulse). Sơ đồ điều khiển động cơ bước cho trong hình dưới:

**Hình 12.31 Điều khiển Stepper motor**

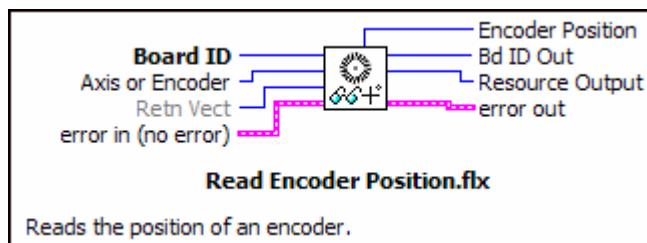
### Đọc Encoder

#### Tại sao cần encoder?

**Encoder** dùng để đo/xác định vị trí, hoặc từ vị trí đo được tính ra vận tốc hoặc gia tốc của chuyển động quay như trực quay động cơ. Vận tốc xe, vận tốc xoay của khớp trong robot, vv.

#### Cách thức đọc tín hiệu encoder

Để lọc đọc encoder, ta sử dụng hàm Read Encoder Position (Card PCI Motion control) hoặc counter (card DAQ). Ở đây giải thích hàm **Read Encoder**.

**Hình 12.32 Hàm đọc Encoder**

**Board ID:** là một số duy nhất được gắn vào cho card thông qua phần mềm MAX.

**Axis or Encoder:** là trục hoặc encoder cần đọc.

**Retn Vect:** đích trả về của tín hiệu bởi hàm đọc encoder. Ví dụ về host (0xFF), tới biến (0x01 tới 0x78), tới biến gián tiếp (0x81 tới 0xF8), không trả về thì điền (0).

**Error in (no error):** để thông báo các lỗi nếu có.

**Encoder Position:** giá trị đọc được (đơn vị là quadrature counts).

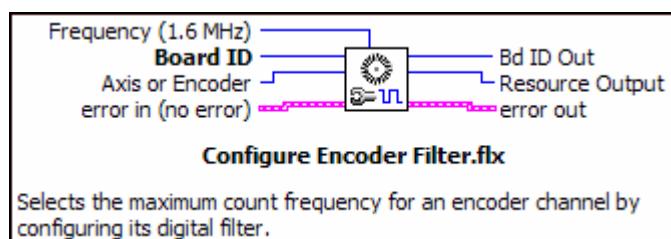
**Bd ID Out:** dùng cho việc tái sử dụng ID.

**Resource Output:** dùng cho việc tái sử dụng Axis, Encoder, vv.

**Error out:** dùng tái sử dụng từ các error out trước đó.

### Lọc nhiễu tín hiệu encoder

Để lọc nhiễu, ta sử dụng hàm Configure Encoder Filter.



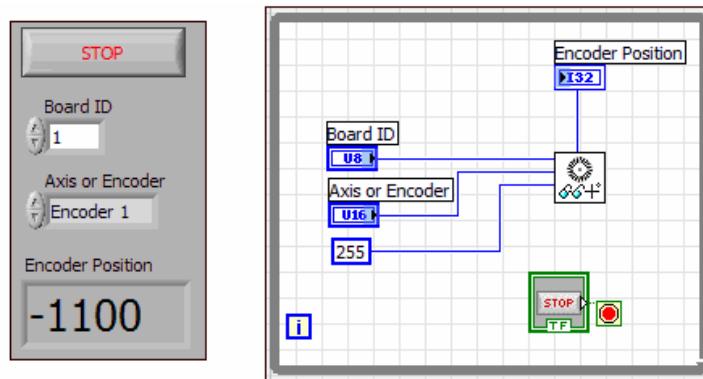
*Hình 12.33 Xác lập Encoder*

Xem bảng giá trị tần số và tần số đếm cao nhất bên dưới.

Frequency Value	Maximum Count Frequency
0	25.6 MHz
1	12.8 MHz
2	6.4 MHz
3	3.2 MHz
4	1.6 MHz (default)
5	800 kHz
6	400 kHz
7	200 kHz
8	100 kHz
9	50 kHz
10	25 kHz
11	Disabled

*Hình 12.34 Tần số và tần số đếm cao nhất*

### Ví dụ mẫu đọc Encoder (block diagram, front panel)



**Hình 12.35 Chương trình đọc Encoder**

#### Lưu ý

Cần chú ý khi quy đổi và scale các giá trị của encoder, đặc biệt khi sử dụng động cơ có hộp số (gear).

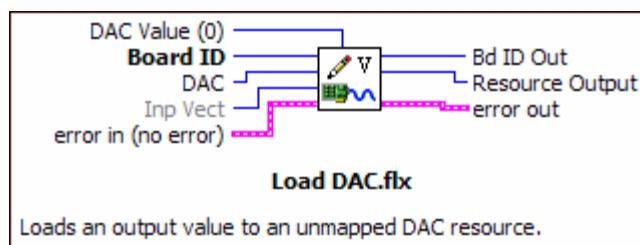
#### 12.6.4 Xuất tín hiệu điện áp (Analog output)

##### Tại sao cần tín hiệu điện áp?

Vì một số bộ khuyếch đại động cơ có thể điều khiển bởi tín hiệu Analog, hoặc khi cần một nguồn Analog trong điều khiển tự động, robotics, thí nghiệm. Ta sẽ phải xuất ra tín hiệu này.

##### Cách thức xuất tín hiệu điện áp

Để xuất điện áp, ta sử dụng hàm Load DAC.



**Hình 12.36 Hàm xuất tín hiệu DAC**

Để phát được tín hiệu điện áp tại DAC OUTPUT cần:

- Trong MAX, tab Axis xuất điện áp phải chọn (set) là stepper hoặc disable.

- Chọn stepper open loop trong Max.

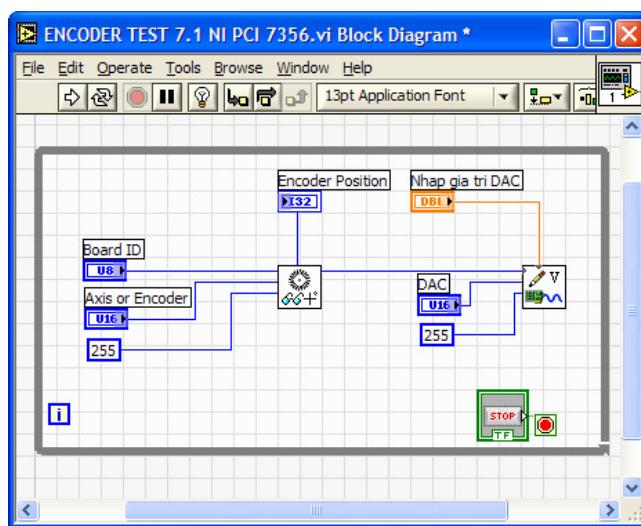
**DAC Value** có giá trị từ -32,768 to +32,767 tương đương xuất ra  $\pm 10$  V (Đối với PCI 7356, các card khác cần xem datasheet của card).

Chú ý: Không sử dụng DAC torque limits và offsets khi xuất trực tiếp điện áp (Load DAC)

**Board ID** xem phần encoder.

**DAC**: Là DAC cần sử dụng để xuất điện áp.

**Ví dụ mẫu (block diagram, front panel) cho trong hình sau:**



**Hình 12.37 Hoàn thành**

Ví dụ trên gồm hai phần, đọc encoder và xuất analog output.

### Chú ý

Chỉ dùng hàm này khi gửi tín hiệu điện áp tới các kênh unmapped DAC. Tức chỉ khi các DAC này không được sử dụng cho Servo motor control thì nó mới được sử dụng như một kênh Analog output đa mục đích.

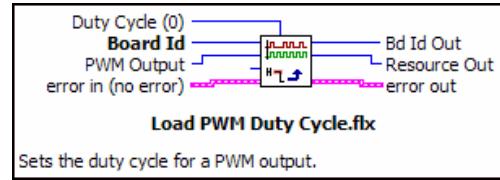
### Điều chế (xuất) xung PWM

#### Tại sao cần PWM?

Điều chế xung thường gặp khi cần điều khiển động cơ DC, động cơ không chổi than hoặc các thí nghiệm liên quan tới nguồn xung ở đầu vào.

### Cách thức điều chế xung

Để xuất PWM, ta sử dụng hàm Load PWM Cycle.



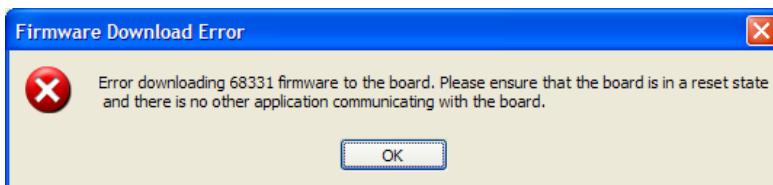
Hình 12.38 Điều chế xung

**Duty Cycle** có giá trị từ 0 tới 255.

**PWM Output** là chọn nguồn xuất PWM (1 hoặc 2, vv.).

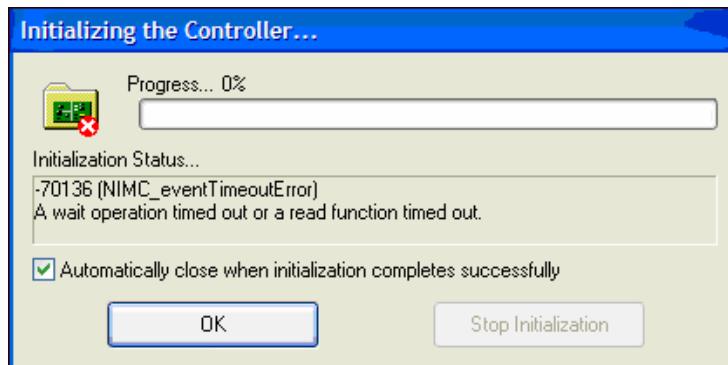
### 12.7 Một số lỗi thường gặp với card PCI 7356

**Firmware download error** xuất hiện khi cập nhật firmware. Bạn cần reset lại card từ tab **Status** sau đó update firmware lại.



Hình 12.39 Lỗi firmware

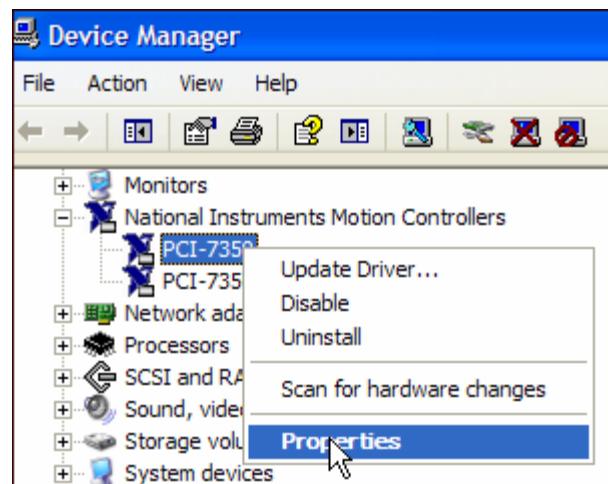
### Thông báo lỗi khi Initialize card.



Hình 12.40 Lỗi khi Initialize

**Card báo lỗi không cập nhật firmware** và **fail** tại test resource. Trong trường hợp này cần tháo card ra một vài tiếng. Sau đó thực hiện lại từ đầu sơ đồ tóm tắt các bước thực hiện lập trình card PCI 7356 trong tài liệu đã trình bày.

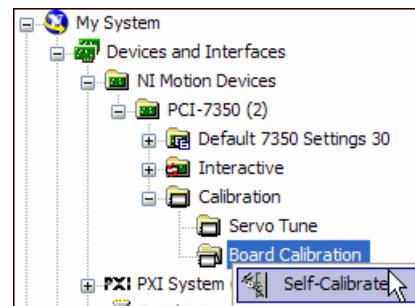
**Trường hợp lỗi xảy ra khi có hai card PCI-73xx gắn vào,** cần lưu ý ta có thể tìm được PCI Slot 1 hoặc 2 bằng cách chọn properties vào Device manager.



Hình 12.41 Xem đặc tính

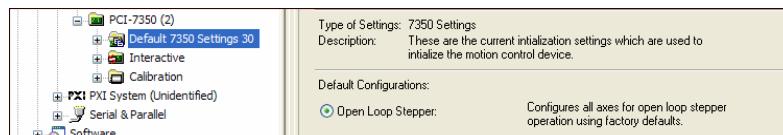
Không xuất được điện áp bằng **Load DAC** dù việc **Initialize** đã hoàn thành và đã đọc được encoder.

Khi đó bạn cần **Calibrate** lại card bằng cách chọn chuột phải lên **Board calibration**, chọn **Self-Calibrate**.



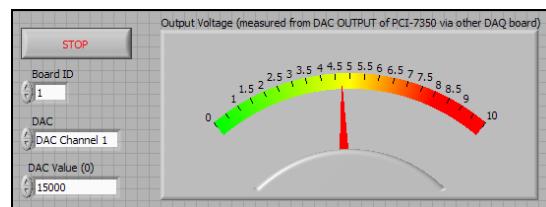
Hình 12.42 Calibrate

Và chắc chắn **Current setting** phải là **Open Loop Stepper** như sau.



Hình 12.43 Chọn Open Loop Stepper

Đồng thời lưu ý, muốn xuất điện áp khoảng **5Volt** thì phải nhập vào số **15000** (tùy vào số bit của bộ DAC, xem thêm tại help của Load DAC) chứ không phải nhập số **5**.

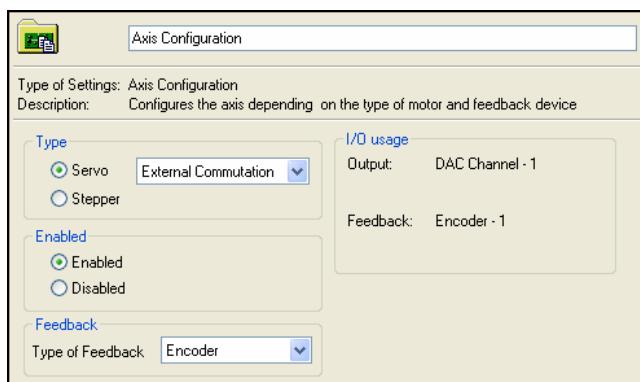


Hình 12.44 FP của chương trình

### Test resources và Update firmware

Đối với card còn tốt, Khi chưa update **firmware** (có dấu chấm than  ) thì test resource vẫn phải thành công (The device passed the resource test successfully!). Đối với card có hư hỏng thì việc test resource sẽ fail.

Đọc **Velocity** sử dụng hàm **Read Velocity** của **LabVIEW** thì cần phải chọn lại setting là **Servo: External Communication** và **Enable feedback**. Cần chọn Open stepper motor trước để xuất Analog output, sau chạy tốt chương trình xuất DAC thì mới vào Servo, chọn External Communication để đọc encoder.



**Hình 12.45 Chọn Servo mode**

Tới đây, bạn đã có thể triển khai lập trình bộ điều khiển chuyển động với card PCI. Nếu có khó khăn hay lỗi không khắc phục được, bạn có thể gửi một email tới tác giả qua [bahai@hcmute.edu.vn](mailto:bahai@hcmute.edu.vn) để được trợ giúp. Chúc bạn thành công.

#### Bạn có biết?

Card PCI 7356 có thể hỗ trợ điều khiển cùng lúc 6 động cơ. Độ phân giải của Encoder counter là 32bit. Card này có thể mua được từ công ty TNHH Linh Phú, Công ty Tân Minh Giang, Công ty DBM, Công ty INO hoặc Công ty National Instrument Vietnam.

## 12.7 Bài tập

**Câu 1:** Thực hành đo Encoder với card PCI 7356

**Câu 2:** Thực hành điều khiển PID với card PCI 7356

## Tài liệu tham khảo

Các tài liệu này nói về ứng dụng của LabVIEW sử dụng Card NI PCI-7356.

- [1] <http://labview.hocdelam.org>.
- [2] Nguyễn Bá Hải, Giáo trình Lập trình LabVIEW, 2008, Hocdelam Group.
- [3] <http://ni.com>.
- [4] Ba Hai Nguyen, Jee-Hwan Ryu, "Semi-Experimental Results on a Measured Current Based Method for Reproducing Realistic Steering Feel of Steer-By-Wire Systems," 15th Asia Pacific Automotive Engineering Conference in Hanoi-Vietnam, 26-28 October 2009 (APAC15), Hanoi, Vietnam, 2009. [Download PDF](#)
- [5] Ba Hai Nguyen, Jee-Hwan Ryu, Van-Dung Do, "Velocity Estimation For Haptic Applications," The 2009 International Forum on Strategic Technologies (IFOST2009), Hochiminh, Vietnam, 2009. [Download PDF](#)
- [6] Ba Hai Nguyen, Hai-Bac Ngo, Jee-Hwan Ryu, "Novel Robust Control Algorithm of DC Motors," The 6th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2009), Seoul, Korea, 2009. [Download PDF](#)
- [7] Van-Dung Do, Thanh-Thuong Tran, Ba-Hai Nguyen, "Simulation and Experimental Results on Study of Automotive Airflow Meter Swappability," 15th Asia Pacific Automotive Engineering Conference in Hanoi-Vietnam, 26-28 October 2009 (APAC15), Hanoi, Vietnam, 2009. [Download PDF](#)
- [8] Ba Hai Nguyen, Jee-Hwan Ryu, "Direct Current Measurement Based Steer-By-Wire Systems for Realistic Driving Feeling," Proceedings of the IEEE International Symposium on Industrial Electronics 2009, Seoul, Korea. [Download PDF](#)
- [9] NI-Motion Help, cách chọn động cơ, thiết kế các chuyển động cơ bản như một động cơ hoặc chuyển động kết hợp nhiều động cơ (trục), thu thập hình ảnh phụ vụ ứng dụng chuyển động, khái niệm về contouring and breakpoints.

- [10] NI-Motion Readme, các yêu cầu về máy tính để sử dụng card NI PCI 73xx, các tính năng mới, các lỗi mới cập nhật, các lưu ý đáng chú ý về card.
- [11] NI-Motion Function Help, Dùng để tham khảo (reference) cho ngôn ngữ C and Visual Basic. Gồm tên hàm, mô tả hàm, các thông số dùng trong hàm, và lỗi của hàm.
- [12] NI-Motion VI Help Tham khảo về các hàm LabVIEW VI dùng trong điều khiển chuyển động.
- [13] Measurement & Automation Explorer Help for Motion, cách thức thiết lập (configure) card NI 73xx, một số khái niệm về phân tích biểu đồ Bode, thông số vòng lặp của hệ thống.
- [14] NI 73xx user manual, tài liệu mô tả về phần cứng và ứng dụng của card NI 73xx.

## BÀI 15

# CÁC CHUYÊN ĐỀ MỞ RỘNG

Học xong bài này, bạn sẽ nắm được:

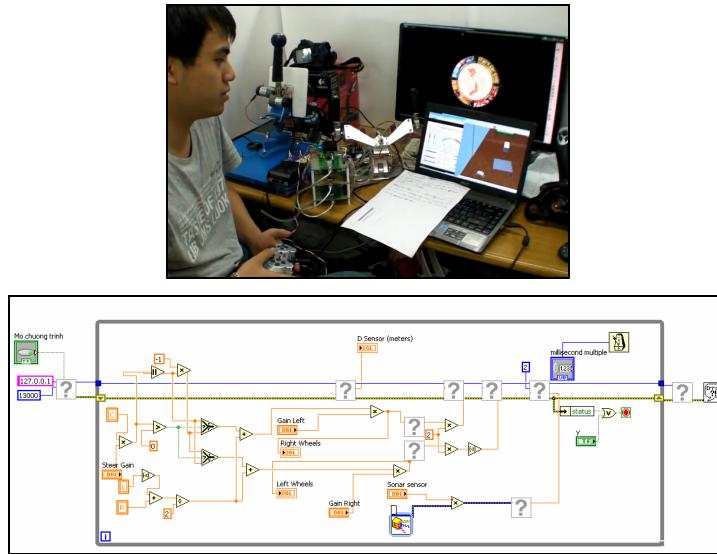
- Khái niệm và sử dụng 3D trong LabVIEW
- Thực hành lập trình vi xử lý ARM đơn giản
- Xử lý ảnh cơ bản với LabVIEW

Do thời gian có hạn, tập sách này được xuất bản với mục đích giúp bạn bắt đầu làm quen và thực hiện một số ứng dụng với LabVIEW. Ở lần tái bản sau, chúng tôi sẽ lần lượt hướng dẫn người dùng toàn bộ các thư viện, hàm và công cụ có trong LabVIEW. Tuy nhiên, để bạn có cái nhìn khái quát về các ứng dụng này, cuốn sách này sẽ tóm tắt những nội dung đó thành các chuyên đề mở rộng dưới đây.

### 15.1 Điều khiển hình ảnh (2D và 3D) trong LabVIEW

Bên cạnh các ứng dụng đã trình bày trong các bài trước, LabVIEW còn hỗ trợ người dùng mô phỏng 3D với nhiều lựa chọn như mô phỏng với 3D Picture Control Module, mô phỏng với Solidworks, vv... Như trong hình, ta thấy việc mô phỏng 3D trong LabVIEW cho phép người sử dụng xây dựng một mô hình thực tế dựa trên mô phỏng vật lý (có mô hình động lực học) với cảnh quan tuyệt đẹp hình 3D tùy chỉnh. Ưu điểm của sử dụng mô phỏng 3D là:

- Để tạo các mô hình sản phẩm mẫu (prototype) nhanh chóng từ những ý tưởng trước khi sản xuất hàng loạt.
- Để dàng chọn lựa, thay đổi kết cấu thiết kế robot hay hệ thống trước khi quyết định sản xuất.
- Để nhanh chóng kiểm tra hành vi của robot và các ứng dụng mới mà không cần tới bất kỳ phần cứng nào ngoài phần mềm LabVIEW.
- Để dễ dàng tạo ra môi trường mới cho robot để di chuyển.
  - Để đơn giản hóa hành vi của robot và các ứng dụng sửa lỗi
- Để phát triển và chứng minh tính khả thi của robot đang thiết kế cho nhà đầu tư với chi phí gần như bằng không (0).



**Hình 15.1 Ví dụ mẫu về điều khiển robot ảo 3D với LabVIEW qua tay game**

Để thực hiện phần này bạn cần cài đặt LabVIEW, Robotic module, hoặc bạn cũng có thể tự thiết kế mô hình robot 3D bằng các phần mềm và công cụ 3D khác như Solidworks, Maya, OpenGL, vv... Tập sách tái bản lần tới sẽ đề cập chi tiết phần này.

Một công cụ lập trình 3D khác ngay trong bản dùng thử của LabVIEW là 3D Picture control cũng giúp bạn dễ dàng thực hiện các ứng dụng 3D cơ bản được trình bày như sau:

### 15.1.1 Tổng quan về kỹ thuật mô phỏng

Mô phỏng là gì? – Mô phỏng là tái hiện giống thật động học, động lực học của một vật, một hệ thống, một hay nhiều robot...

Tại sao cần mô phỏng? Mô phỏng (2D, 3D) là tái hiện giống thật động học, động lực học của một vật, một robot, một hệ thống. Kỹ thuật mô phỏng từ lâu đã đóng vai trò hết sức quan trọng trong giáo dục đào tạo và công nghiệp. Trong giáo dục việc mô phỏng những mô hình sẽ giúp cho người học tiếp cận vấn đề nhanh hơn. Việc mô phỏng những mô hình thực tế sẽ giúp giảm chi phí đầu tư trang thiết bị mà vẫn đảm bảo chất lượng đào tạo.

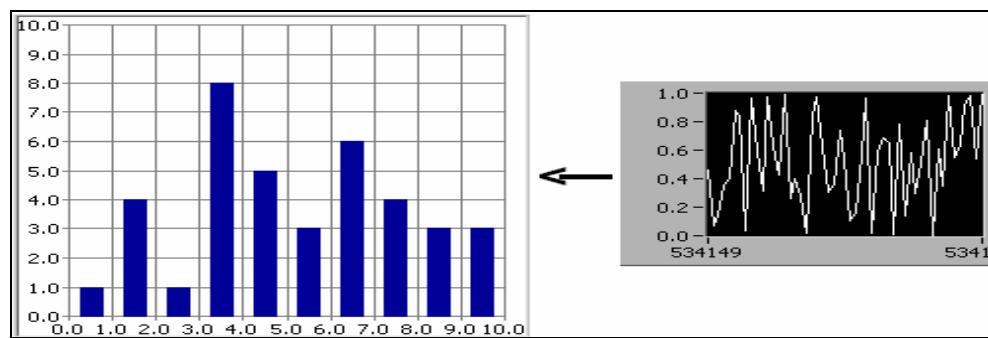
Trong công nghiệp việc mô phỏng sẽ giúp ta phát hiện sai sót, nhanh chóng điều chỉnh trước khi đi vào sản xuất. Nhờ vào việc đó mà chúng ta có thể tiết kiệm thời gian và chi phí gia công, sản xuất. Ngoài ra, việc mô phỏng cũng giúp chúng ta giới thiệu ý tưởng tới các đối tác một cách nhanh chóng, tiện lợi mà không cần có một sản phẩm thực.

Mô phỏng 3D sử dụng phần mềm LabVIEW là một giải pháp hiệu quả giúp cho các giáo viên, giảng viên, các kỹ sư và các nhà nghiên cứu khoa học nói chung có thể thực hiện được ý tưởng của mình một cách nhanh chóng. Các phần mềm mô phỏng 2D, 3D:

- Inventor;
- Solidwork;
- Google SketchUp;
- Studio của ABB;
- EasyRob (Mô phỏng robot)
- LabVIEW (Ở bài này, chúng tôi chọn LabVIEW làm công cụ)
- Vv...

### 15.1.2 Giới thiệu về 3D Picture Control trong LabVIEW

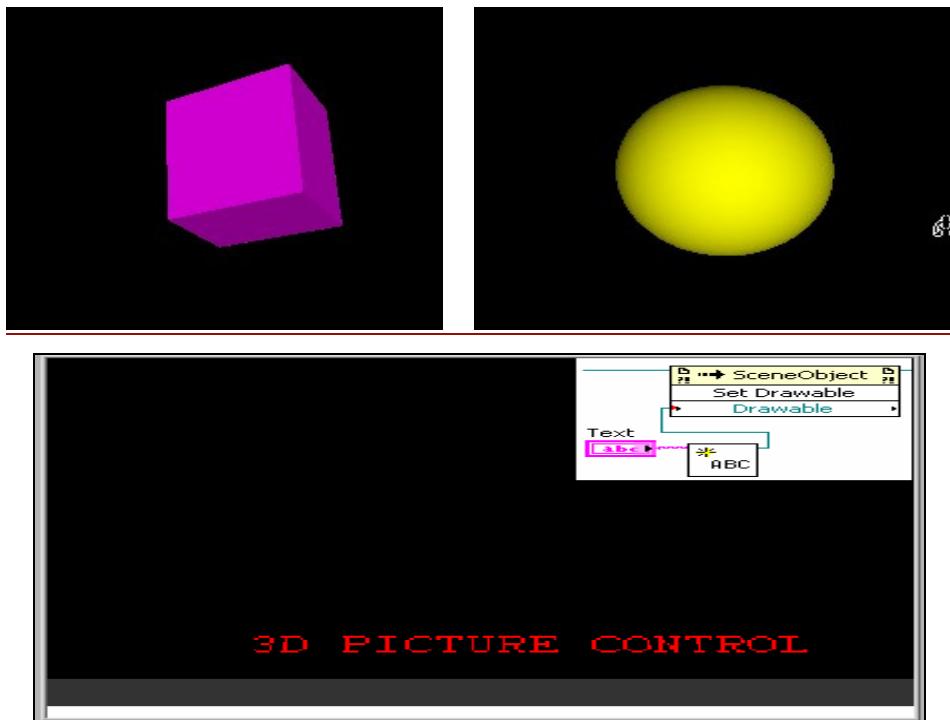
- Công cụ cần thiết gồm những gì?
  - o LabVIEW 2009 Evaluation
  - o Kiến thức cơ bản về vật lý
  - o Tính toán động học, động lực học các hệ cơ học.
- Tài liệu
  - o LabVIEW 3D Picture Control Help;
  - o Internet.
- 3D Picture Control trong LabVIEW làm được gì?
  - o Vẽ 2D, biểu diễn các khảo sát dưới dạng biểu đồ:



**Hình 15.2 Ứng dụng của Picture control**

- Vẽ các hình khối 3D:

Khối cầu, khối hình lập phương hay các hình khối có hình dạng phức tạp...



**Hình 15.3 Một số kết quả đơn giản với 3D picture control**

- Mô phỏng động học & động lực học

Động học là gì? - Là ngành khoa học nghiên cứu sự thay đổi vị trí của vật này so với vật khác.

Động lực học là gì? - Là ngành khoa học nghiên cứu về sự tương tác lực giữa các chất điểm với nhau.

Các bước mô phỏng:

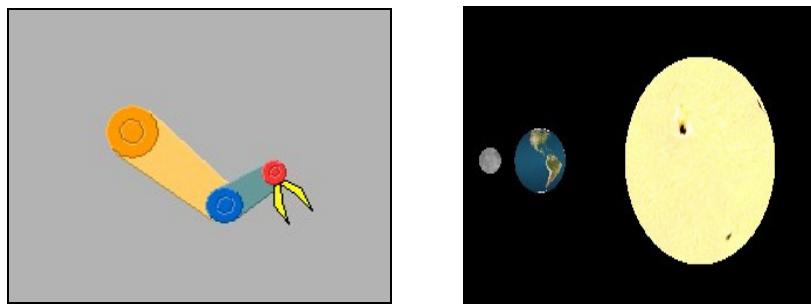
Tạo các hình khối theo yêu cầu;

Thiết lập quan hệ (ràng buộc) cho các hình khối;

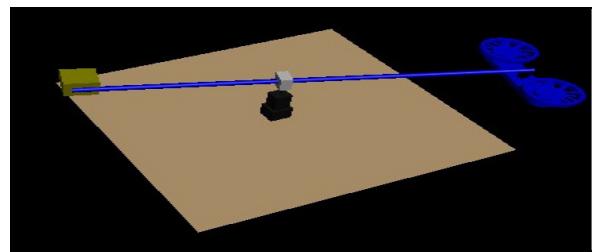
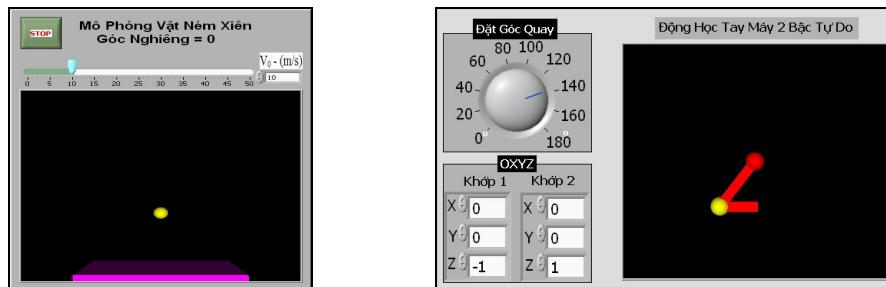
Viết thuật toán điều khiển;

Đưa thuật toán điều khiển vào hệ mô phỏng.

- Các Demo mẫu

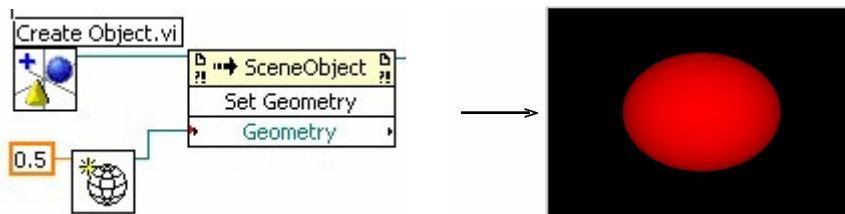
**Hình 15. 4 Mô phỏng động học cánh tay robot**

Mô phỏng ĐH quỹ đạo bay của mặt trời, trái đất, mặt trăng

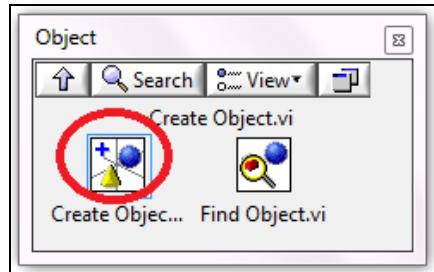
**Hình 15.5 Mô phỏng động lực học Helicopter****Hình 15. 6 Mô phỏng vật ném xiên****Hình 15.7 Mô phỏng động học cánh tay 2 bậc tự do**

### 15.1.3 Từng Bước Thao Tác Mô Phỏng 3D Trong LabVIEW

- Thao tác vẽ các khối căn bản (cầu, hộp, trụ...)

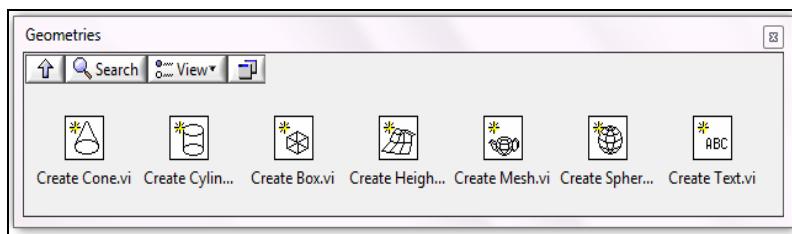
**Hình 15.8 Hàm Creat Object**

Hàm **Create Object** là lệnh tạo một hình khối mới (cầu, trụ, hộp...) để hiển thị lên môi trường 3D của LabVIEW. Để lấy làm **Create Object**, tại cửa Block Diagram ta phải chuột >> Programming >> Graphics & Sound >> 3D Picture Control >> Object >> Create Object.



**Hình 15.9 Vị trí hàm Create Object**

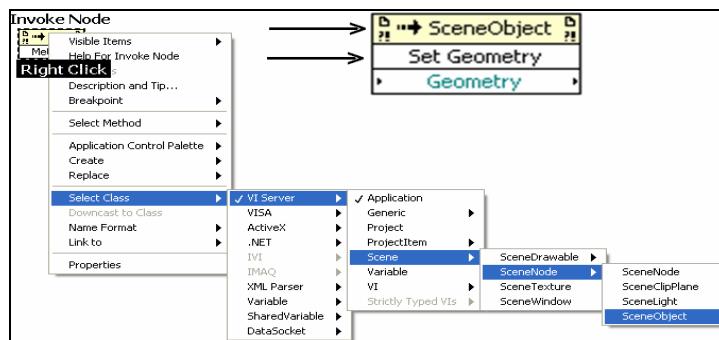
Muốn tạo hình khối nào thì bên cửa sổ BD (Block Diagram) ta phải chuột >> Programming >> Graphics & Sound >> 3D Picture Control >> Geometry, sau đó chọn các hình khối cần tạo.



**Hình 15.10 Thư viện Geometry**

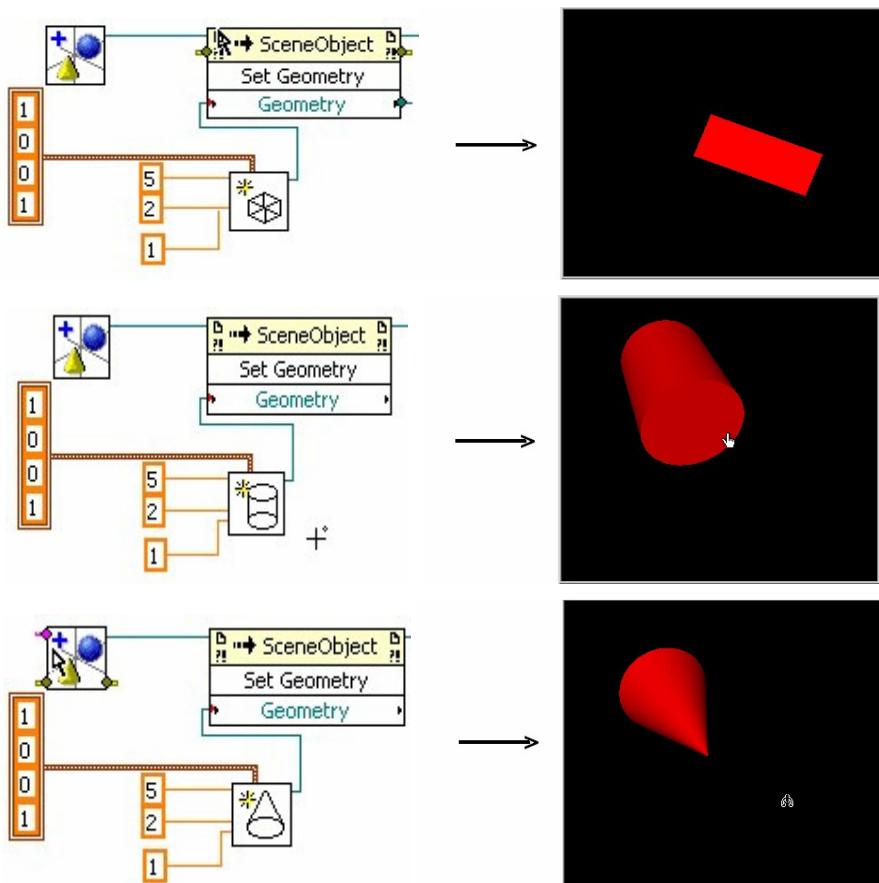
Các hình khối mà LabVIEW hỗ trợ trong thư viện 3D Picture Control.

Khối **SceneObject** được tạo thành từ khối **Invoke Node** nằm trong thư viện **Application Control**, là khối có chức năng tạo không gian hoạt động cho vật thể.



**Hình 15.11 Tạo không gian hoạt động cho vật thể**

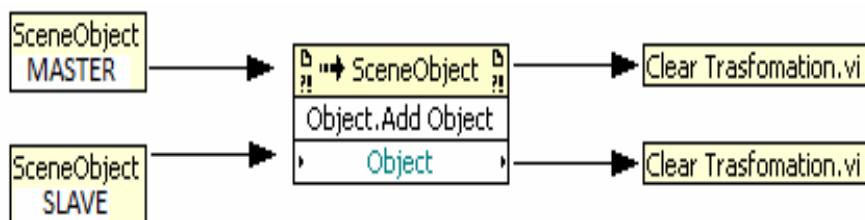
Tương tự ta có thể tạo được các hình khối sau:



**Hình 15.12 Tạo các hình khối**

- Thao tác ràng buộc cho các khâu:

Ta dùng khôi **Invoke Node**, với Class là **SceneObject**, Method là **Object. Add Object**.



**Hình 15.13 Ràng buộc**

- Thao tác gán kiểu chuyển động (tịnh tiến, quay) cho các khâu.

Để tạo các kiểu chuyển động cho các khâu, ta tìm tới cửa sổ Transformation trong thư viện 3D PICTURE CONTROL và chọn **Translate Object** (nếu là muốn gán chuyển động tịnh tiến) hay **Rotate Object** (nếu muốn gán chuyển động quay):

- Viết thuật toán (động học, động lực học) điều khiển chuyển động cho các khâu:

Tùy vào yêu cầu mô phỏng (động học động lực học) của ứng dụng mà lập trình thuật toán điều khiển.

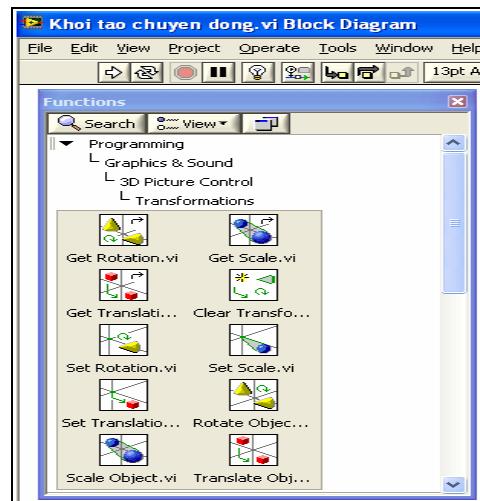
- Thao tác gán thuật toán điều khiển chuyển động tới các khâu:
- Chạy mô phỏng.

#### 15.1.4 Mô Phỏng Vật Lý: Vật Ném Xiên

Bạn cần thực hiện các bước sau:

- Tạo nền (mặt đất), Tạo vật ném xiên
- Ràng buộc cho vật ném xiên và mặt đất
- Viết thuật toán ném xiên ( $Y=V_0t-1/2*gt^2$ )
- Gán thuật toán bay xiên cho vật
- Chọn vận tốc ban đầu  $V_0(m/s)$  và chạy mô phỏng.

Ngoài ra, bạn cũng có thể thực hiện các bài toán mô phỏng phức tạp hơn để có thể tạo ra môi trường 3D như sau (*trích đề tài Cao học Mô phỏng 3D với LabVIEW để chế tạo mô hình tập lái xe ô tô và xe tăng, tác giả: KS. Trần Ngọc Trung, giáo viên hướng dẫn: TS. Nguyễn Bá Hải*):



**Hình 15.14 Thư viện hàm tạo chuyển động**



**Hình 15.15 Giao diện mô hình tập lái xe**

### 15.1.5 Mô phỏng kết hợp LabVIEW với các phần mềm thiết kế 3D khác (Google Sketchup, Solidworks...)

Vì phần mềm LabVIEW chỉ tạo được các hình khối đơn giản đó là vì sao chúng ta cần sử dụng các phần mềm thiết kế 3D khác để có thể mô phỏng các hệ thống phức tạp hơn. Trong khóa học này sẽ sử dụng 2 phần mềm là Google SketchUp và Solidworks.

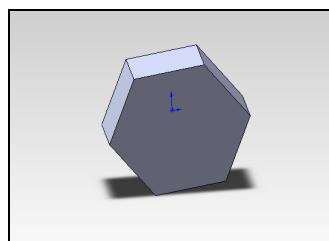
Để gọi một vật thể từ phần mềm thiết kế 3D thứ 2 thì LabVIEW hỗ trợ các đuôi mở rộng sau: ASE, STL, VRML97. Trong giới hạn bài giảng này sẽ hướng Import file mở rộng VRML97 vào môi trường 3D của LabVIEW.

Để có thể Import 1 vật thể từ phần mềm Google SketchUp hay Solidworks vào 3D LabVIEW đòi hỏi trước tiên ta cần biết sử dụng 2 phần mềm trên. Sau đây sẽ là các 2 ví dụ đơn giản giúp bạn có thể gọi 1 file mở rộng VRML97 vào môi trường 3D của LabVIEW:

**Ví dụ 1:** Import 1 khối lục giác từ phần mềm Solidworks vào 3D Picture của LabVIEW:

Thực hiện:

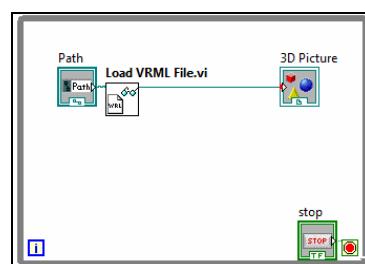
- Bước 1: Thiết kế khối lục giác bằng phần mềm Solidworks sau đó save as sang đuôi mở rộng VRML97 (lucgiac.wrl)



**Hình 15.16 Vẽ bằng Solidworks**

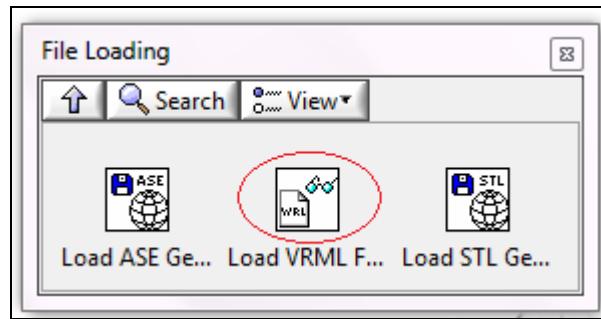
- Bước 2: lập trình LabVIEW

Thuật toán:



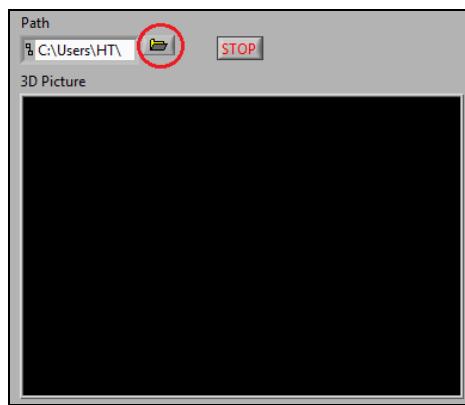
**Hình 15.17 Đưa file vào LabVIEW**

Hàm Load VRML File được lấy trong thư viện 3D Picture Control  
**>>File Loading** của LabVIEW



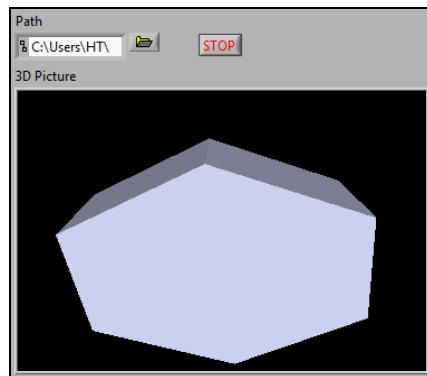
**Hình 15.18 Load VRML file**

- Bước 3: Load file lucgiac.wrl được export từ Solidworks tại cửa sổ **Front Panel**



**Hình 15.19 FP của chương trình**

- Bước 4: Chạy chương trình và thấy kết quả:

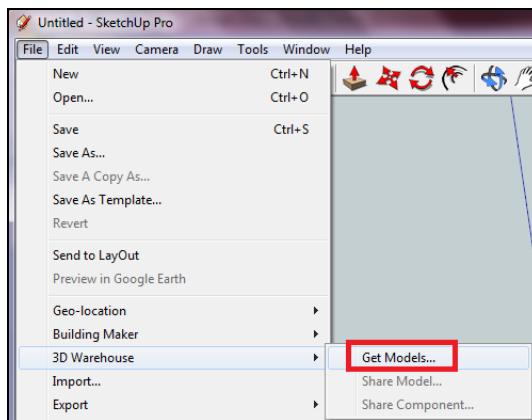


**Hình 15.20 Kết quả**

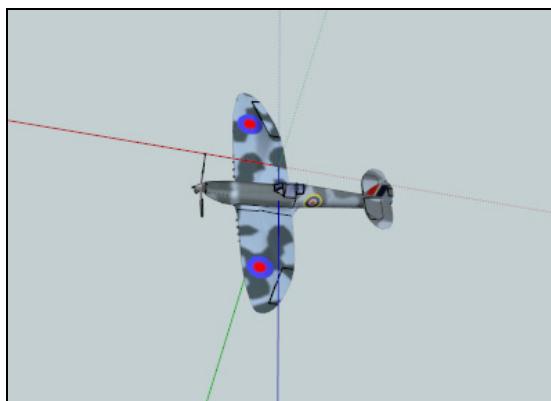
**Ví dụ 2:** Import 1 chiếc máy bay vào 3D Picture của LabVIEW sử dụng phần mềm thiết kế Google SketchUp :

Thực hiện:

- Bước 1: Thiết kế máy bằng phần mềm Google SketchUp hoặc để tiết kiệm thời gian ta có thể lấy chiếc máy bay tại thư viện mở của Google. Để lấy chiếc máy bay từ thư viện **3D Warehouse** ta thực hiện như sau: tại giao diện của phần mềm Google SketchUp ta chọn **File >> 3D Warehouse >> Get Models** (yêu cầu máy tính cần có internet), sau đó tại cửa sổ **3D Warehouse** tìm với từ khóa **plane**, tiếp theo chọn cái plane hợp ý và download:



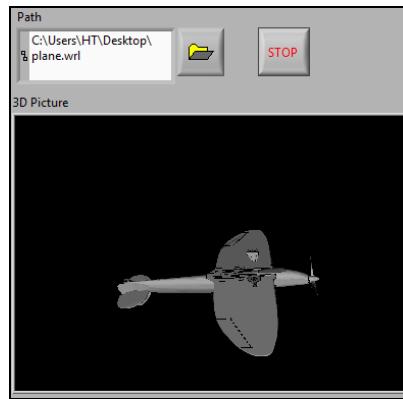
*Hình 15.21 Get model*



*Hình 15.22 Lấy chiếc máy bay*

Chiếc máy bay bên phần mềm Google SketchUp sau khi download từ **Warehouse** về

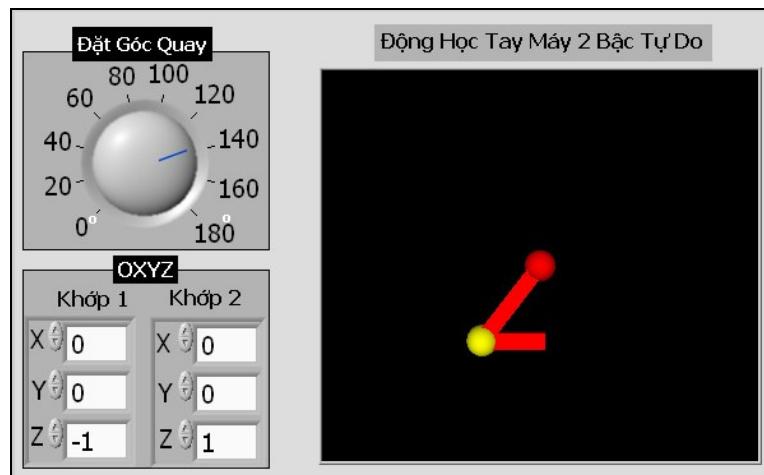
- Bước 2: lập trình trên ví dụ 1 ta được kết quả như sau:



**Hình 15.23** Chiếc máy bay bên phần mềm LabVIEW

#### 15.1.6 Mô Phỏng Robot: Động Học Cánh Tay Robot 2 Bậc Tự Do

- Tạo Khớp thứ nhất
- Tạo Khâu thứ nhất
- Tạo Khớp thứ 2
- Tạo khâu thứ 2
- Ràng buộc Khớp 1 & Khâu 1 & Khớp 2 & Khâu 2
- Viết thuật toán điều khiển động học: nhập góc quay cho các khớp theo các trục X, Y, Z
- Gán thuật toán điều khiển cho cánh tay
- Nhập thông số và mô phỏng.



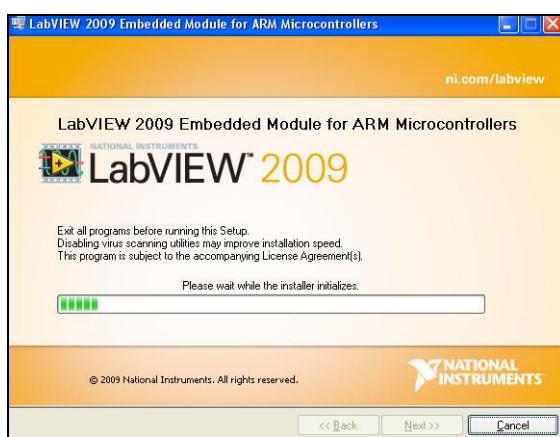
**Hình 15.24**

## 15.2 Lập trình vi điều khiển ARM với LabVIEW

Để lập trình LabVIEW với vi điều khiển ARM tác giả chọn KIT ARM cung cấp bởi National Instruement để làm ví dụ. Các bước thực hiện chi tiết trong phần này chỉ đúng với giả thuyết bạn có KIT này để thực hành.

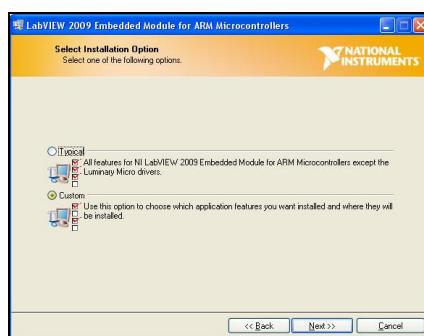
### 15.2.1 Cài đặt DVD Mô đun nhúng cho vi điều khiển ARM

1. Khởi động và đăng nhập tài khoản admin
2. Cho đĩa DVD vào máy. Nếu cài đặt không tự động bắt đầu, thì double click vào *autorun.exe* để cài đặt



*Hình 15.25*

3. Chọn **Install LabVIEW Embedded Module for ARM Microcontrollers**
4. Làm theo hướng dẫn trên màn hình hiển thị cài đặt mô đun nhúng cho vi điều khiển ARM. Bạn phải chọn cài đặt **custom** và chọn vào mục cài đặt **Luminasy Micro Driver for EK-LM3S8962**. click Continue Anyway để tiếp tục cài đặt



*Hình 15.26*

5. Trong quá trình cài đặt, cài đặt sẽ tự động tải RealView Microcontroller Development Kit. Làm theo hướng dẫn trên màn hình hiển thị để cài đặt RealView Microcontroller Development Kit
6. Sau khi cài đặt hoàn thành, khởi động lại máy
7. Tiến hành đăng ký bản quyền phần mềm

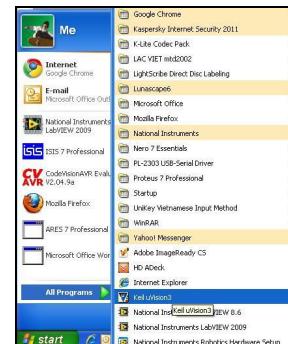
### 15.2.2 Kết nối phần cứng MCB 2300 tới máy tính

1. Các thiết bị cần thiết cho việc cài đặt

- Phần cứng MCB 2300
- Máy vi tính có 2 cổng USB
- Bộ kết nối ULINK2 USB-JTAG
- 2 dây cáp USB

2. Các bước thực hiện

- Khởi động chương trình Keil μ Vision3
- Kết nối ULINK2 USB-JTAG tới cổng USB của máy tính bằng dây cáp
- Kết nối ULINK2 USB-JTAG tới phần cứng MCB2300



**Hình 15.27**



**Hình 15.28**

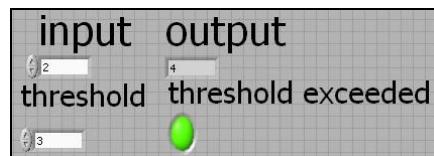
- Kết nối cổng USB từ phần cứng MCB2300 tới cổng USB của máy tính để cấp nguồn cho phần cứng MCB2300

### 15.2.3 Lập trình trên LabVIEW

#### 1. Tạo một chương trình trên LabVIEW

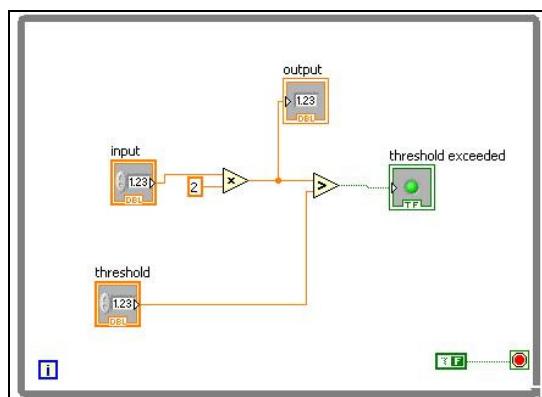
- Tạo 2 numeric controls
- Tạo 1 numeric indicator
- Tại một đèn led

## 2. Đặt tên theo hình sau



Hình 15.29

## 3. Lập trình như hình tại Block diagram



Hình 15.30

Hoàn thành demo đầu tiên với vi điều khiển ARM sử dụng KIT có sẵn cung cấp bởi National Instruments bạn sẽ thấy LED sáng.

## 15.3 Giải pháp về thời gian thực với LabVIEW

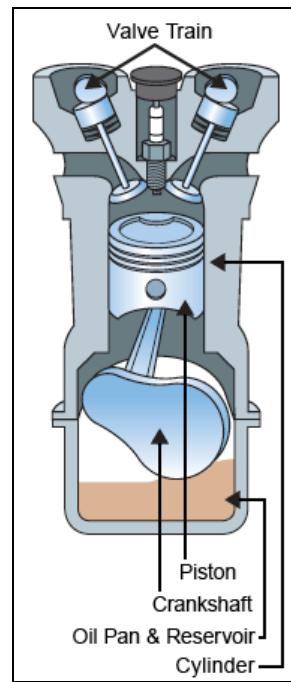
### 15.3.1 Khái niệm về thời gian thực

Thời gian thực là gì? Ta thường sai lầm khi nghĩ hệ thống thời gian thực (Real-Time System, RTS) là hệ thống có tốc độ xử lý nhanh (gần với thời gian thực của tự nhiên). Thực ra một hệ thống thời gian thực có đặc điểm sau:

Thực hiện được một hoặc một số nhiệm vụ (task) trong một thời gian qui định. Đảm bảo được dù có bất kỳ điều gì xảy ra đối với hệ thống thì số nhiệm vụ trên vẫn hoàn thành trong đúng thời gian đó. Cần nhớ hệ RTS gồm: Phần mềm RT (real-time software hoặc real-time module hay RT toolkit) và Phần cứng RT (real-time hardware).

Ví dụ: túi khí (airbag) trên xe hơi là một RTS. Vì khi có cảm biến tai nạn được kích hoạt. Túi khí sẽ được bung ra trong một khoảng thời gian rất ngắn (được định trước bởi kỹ sư lập trình). Giả sử nếu túi khí không là một RTS thì tài xế sẽ chết trước khi túi khí bung. Ví dụ khác về RTS là bộ điều khiển đánh lửa trong động cơ xăng.

Tia lửa cần được điều khiển chính xác ở chu kỳ đốt (tức sau khi chu kỳ hút đã hút hỗn hợp khí và xăng vào rồi, tới chu kỳ nén, nén hòa khí xăng-khí lên áp suất cao, thì tia lửa mới được phát ra bởi bộ điều khiển RTS). Nếu không đúng thời điểm này thì động cơ sẽ không hoạt động được hoặc hoạt động với hiệu suất rất thấp.



**Hình 15.31 Động cơ đốt trong**

**Vậy máy tính cài hệ điều hành Window có phải là RTS?** Trước hết bản thân phần mềm window là không phải RT software vì window là hệ điều hành đa chức năng (chứ không thiết kế cho ứng dụng RT trong kỹ thuật). Thường đơn giản như việc ta dùng phần mềm chat (yahoo messenger) để chat theo thời gian thực (tức bạn gửi tin thì người được gửi sẽ nhận ngay sau khi bạn nhấn enter). Tuy nhiên khi máy có virus hoặc có một phần mềm, hay phần cứng nào tác động (interrupt) thì dù nhiên tin messenger bị đứng (treo máy). Do vậy dù bạn có nhấn enter rồi thì tin vẫn không gửi đi được.

Thứ hai, phần cứng của máy tính để bàn là một hệ thống được thiết kế với mục đích đa năng (của máy tính thông thường như soạn thảo văn bản, chơi game, nghe nhạc, vv). Do vậy, so với máy tính đặc biệt dùng xây dựng lên hệ điều hành RT (RT Computer) thì máy để bàn (Desktop PC) có tính năng kém hơn trong việc tạo ứng dụng RT.

Giải pháp xây dựng hệ thống thời gian thực cho các yêu cầu trên như thế nào? Như đã đề cập ở trên, một hệ thống thời gian thực bao gồm phần mềm thời gian thực và phần cứng thời gian thực. Hai thành phần này được phân bổ ở hai máy tính dùng để phát triển (lập trình) các chương trình (máy host) và máy đích (target RT computer).

Từ đó ta có các lựa chọn sau:

**RT target:** Có thể dùng RT computer, single-board computer hoặc dùng ngay chính máy tính của bạn nếu máy của bạn thỏa mãn điều kiện về RT target. (phương pháp kiểm tra này LabVIEW Hocdelam sẽ đề cập sau). Host computer: Laptop hoặc desktop PC thông thường. Thông thường host PC giao tiếp và làm việc với target PC thông qua chuẩn Ethenet. Vì vậy bạn cần có card mạng rời hoặc card mạng onboard. (Nếu máy của bạn vào internet được có nghĩa máy của bạn đã có card mạng).

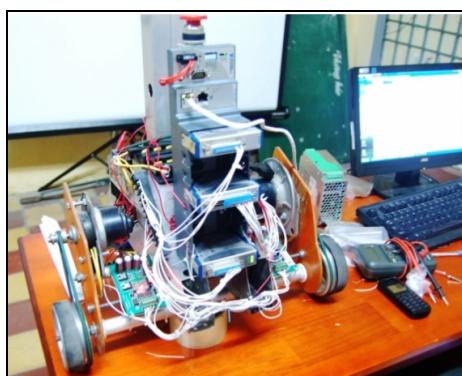
**IO Board:** PCI, PXI, hoặc DAQ chuyên Dùng. RT Software: Dùng USB boot disk sau đó cài LabVIEW RT Module lên máy của bạn), hoặc dùng RTX (real-time extension for Windows XP) cài xong xong với OS của bạn. (LabVIEW Real-Time Module for RTX Targets).

### 15.3.2 Sử dụng các card thời gian thực

Trong các ứng dụng về thời gian thực CompactRIO với vị trí là **Target computer** là một lựa chọn phù hợp nếu bạn không quá lo lắng về vấn đề kinh phí. Để lập trình CompactRIO bạn cần cài đặt các module sau:

- LabVIEW 8.6
- LabVIEW Realtime module tương ứng
- NI-RIO 3.2 hoặc mới hơn
- LabVIEW FPGA

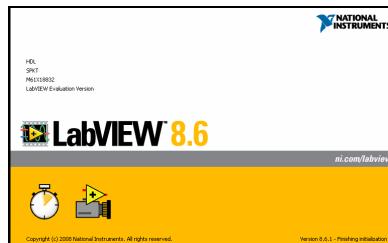
**Host computer** là máy tính Laptop hoặc máy tính Desktop của bạn. Về phần cứng, bạn cần set up (thiết lập) một ứng dụng cụ thể. Trong bài này, tác giả lấy ví dụ một Robot tự động do nhóm sinh viên SPKT-MIN ĐH Sư phạm Kỹ thuật Tp.HCM thực hiện.



**Hình 15.32 Robot tự động của đội MINSPKT sử dụng CompactRIO để điều khiển thời gian thực**

Lập trình cho CompactRIO khá đơn giản sau khi bạn đã kết nối các phần cứng của mình. Các bước lập trình cho CompactRIO cụ thể là:

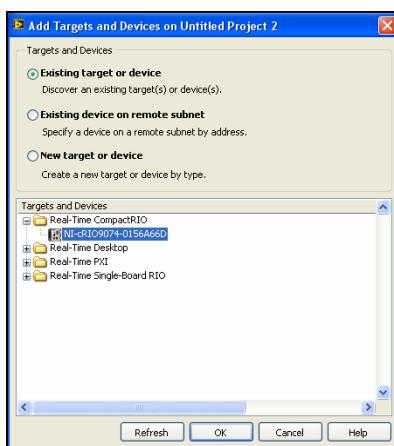
### Bạn mở LabVIEW lên



**Hình 15.33 Khởi động LabVIEW**

Vào New Project > Add> Target > Select CompactRIO

Chọn Module cần lấy tín hiệu

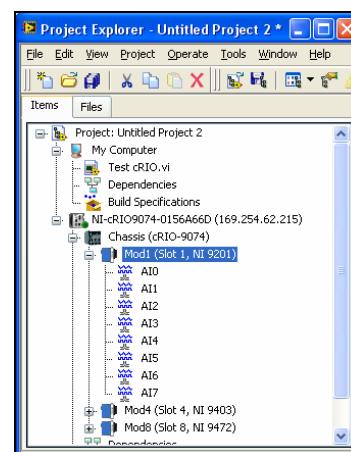


**Hình 15.34 Tạo Project**

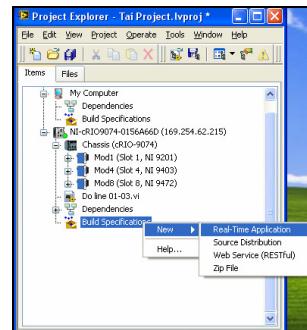
Kéo thả kênh AI0 để đọc tín hiệu Analog.

Build & Deploy & Set as start up

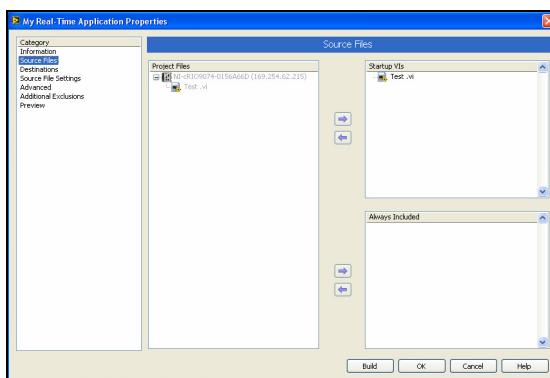
Tạo ra một ứng dụng thời gian thực (Real-time application)



**Hình 15.35 Module Âmlog**

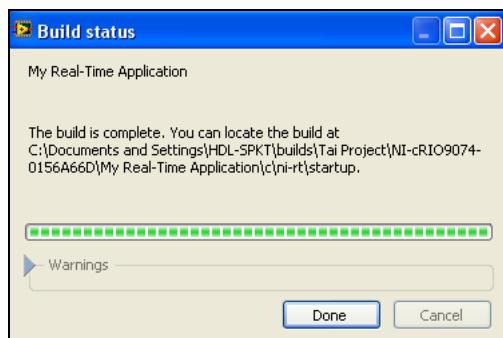


**Hình 15.36 Tạo Real-Time Application**



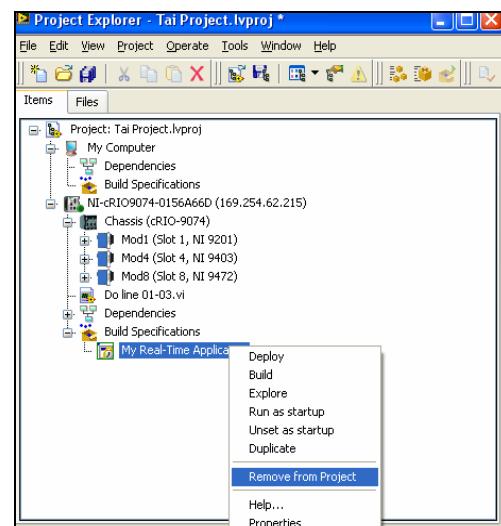
Hình 15.37 Chọn Source file

Chờ cho hệ thống Build thành công.



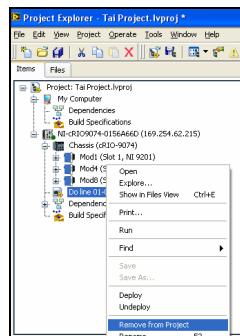
Hình 15.38 Thông báo đã build thành công

- Để tải xuống target computer (deploy)
- Chọn lên My Realtime-application > Chọn vào Deploy.
- Sau khi deploy, muốn nạp lại ta phải Remove Real-time application trong cRIO

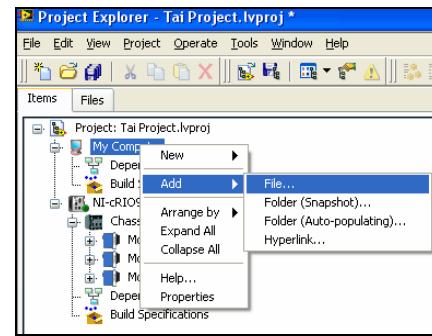


Hình 15.39 Remove dự án từ Application

Để bỏ đi VI này trong dự án (Project). Đưa một file vào dự án mới



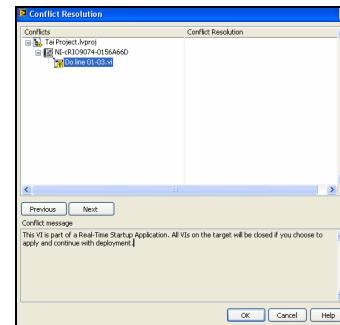
Hình 15.40 Remove File từ dự án



Hình 15.41 Thêm file

Chương trình hỏi có thay thế không? Bạn chọn YES.

Bạn đã hoàn thành tất các bước để lập trình hệ thống thời gian thực với NI CompactRIO.

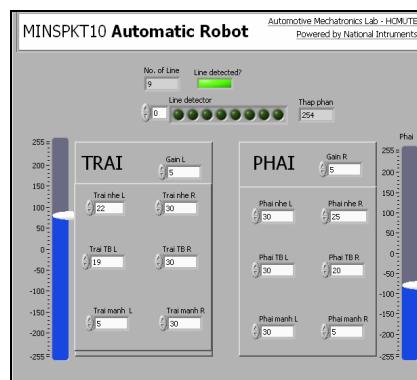


Hình 15.42 Thay thế file

## CHƯƠNG TRÌNH MẪU

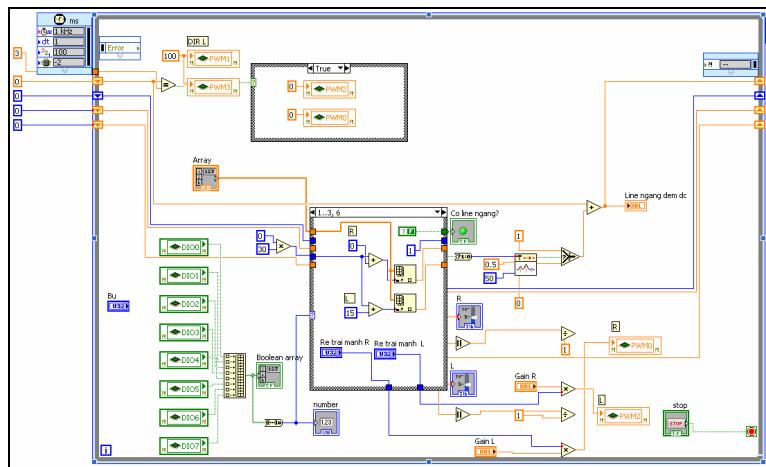
Chương trình mẫu bên dưới mô tả ví dụ điều khiển robot tự động dò line và điều khiển cơ cấu chấp hành với LabVIEW sử dụng CompactRIO.

Bạn có thể tải chương trình mẫu này tại: <http://labview.hocdelam.org>  
> Tài liệu > Tài liệu LabVIEW > Ví dụ mẫu.)



Hình 15.43 Ví dụ Robot dò line của nhóm sinh viên năm nhất MINSPKT

## Front panel chương trình điều khiển robot tự động với CompactRIO



**Hình 15.44 BD của chương trình dò line**

## Blockdiagram chương trình điều khiển robot tự động với CompactRIO

Một điều thú vị và thuận lợi là bạn có thể thay thế CompactRIO với 3 card giao tiếp máy tính giá thành thấp Hocdelam USB-9001 để thực hiện đề tài này với chính LapTOP hoặc máy tính của bạn. Nhưng cần lưu ý là việc thay thế này chỉ nhằm giảm giá thành phần cứng để phục vụ làm demo, học tập và nghiên cứu thuật toán điều khiển Robot trong môi trường giáo dục. Bởi Hocdelam USB-9001 không phải là một phần cứng Real-time như CompactRIO. (Xem video clip việc thay thế card Hocdelam USB 9001 tại: [http://www.youtube.com/watch?v=X\\_eJ7HLDdz4](http://www.youtube.com/watch?v=X_eJ7HLDdz4))

### 15.4 Xử lý ảnh với LabVIEW

Trong kỹ thuật, xử lý ảnh giúp con người xây dựng nhiều hệ thống thông minh và hữu ích từ việc đếm sản phẩm, phát hiện sai lệch (của việc dán nhãn sản phẩm) phân loại sản phẩm cũng như tự động đo lường và ra quyết định điều khiển thiết bị. Một số ứng dụng quen thuộc của xử lý ảnh có thể thực hiện với LabVIEW là:

- Đếm số điện trở trên một mạch điện
- Nhận dạng mặt người
- Tìm lỗi của một sản phẩm (khi sản phẩm bị rạn, nứt, méo, vv)

Xử lý hình ảnh là việc xử lý tín hiệu đầu vào là một hình ảnh, chẳng hạn như một khung ảnh hoặc video, đầu ra của xử lý hình ảnh có thể hoặc là một hình ảnh hoặc một tập hợp các đặc điểm hoặc các thông số liên quan đến hình ảnh.

Dựa trên các thông số đầu ra đó, người ta có thể tiến hành điều khiển thiết bị theo ý muốn ví dụ: Đếm số viên thuốc trong một vỉ đựng, nếu số thuốc (Đầu ra) ít hơn một số thuốc yêu cầu nào đó thì tay robot sẽ tự động bổ sung thêm cho đủ số lượng hoặc gạt vỉ thuốc không đạt yêu cầu ra.

Hầu hết các kỹ thuật xử lý hình ảnh liên quan đến việc xử lý hình ảnh như là một tín hiệu hai chiều và áp dụng tiêu chuẩn kỹ thuật xử lý tín hiệu với nó. Xử lý hình ảnh thường dùng để chỉ xử lý hình ảnh kỹ thuật số, nhưng xử lý hình ảnh quang học và tương tự cũng có thể xảy ra. Bài viết này là về các kỹ thuật sử dụng công cụ Vision assistant để thực hiện các bài xử lý ảnh cơ bản. Bạn có thể gửi yêu cầu hỗ trợ thêm tài liệu xử lý ảnh nâng cao bằng việc gửi email về [supports@hocdelam.org](mailto:supports@hocdelam.org).

#### 15.4.1 Cài đặt và sử dụng Vision Assistant

##### Yêu cầu hệ thống cài đặt:

	Tối thiểu	Yêu cầu
Vi xử lý	Pentium 233MHz hay cao hơn	Pentium4/M hay cao hơn
RAM	256MB	512MB
Hiển thị	Độ phân giải 1024x768 với 16bit màu	Độ phân giải 1024x768 với 24-32bit màu
Chỗ trống ổ cứng	700MB	1GB
Hệ điều hành	Windows 7/vista/xp/2000	
Trình duyệt	Internet Explorer hoặc Firefox	

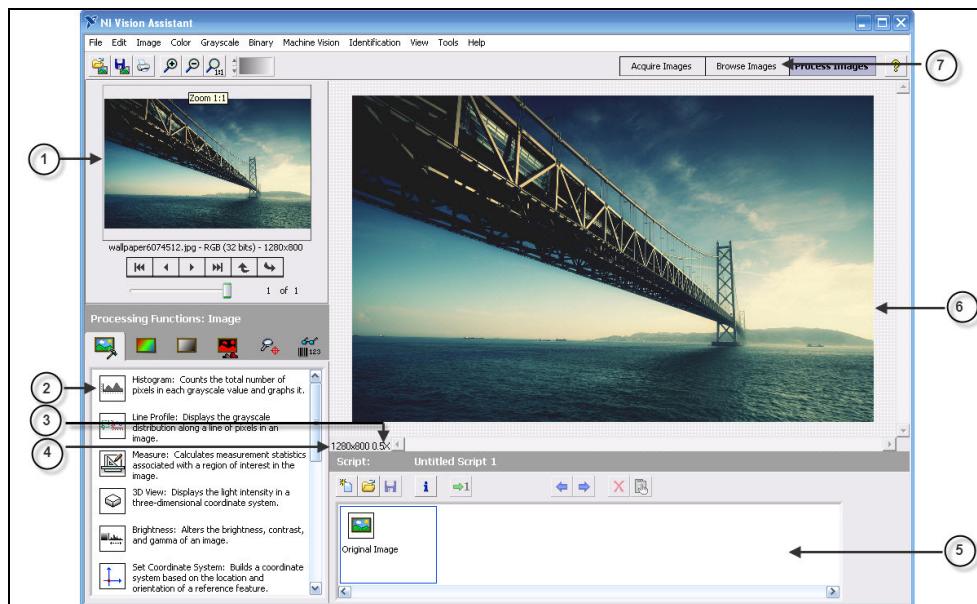
##### Hướng dẫn cài đặt:

- Chạy file setup trong thư mục NI-IMAQ và NI-IMAQdx
- Chạy file setup trong thư mục **Vision Development Module**

(Bạn có thể tải các file này tại <http://labview.hocdelam.org> > Tài liệu > Tài liệu LabVIEW > Xử lý ảnh với LabVIEW)

- **Giới thiệu:** Vision Assistant là công cụ để tạo mẫu và kiểm tra quá trình xử lý ảnh. Sau khi xây dựng các thuật toán xử lý ảnh với các đoạn mã (Script) của Vision Assistant, ta có thể kiểm tra thuật toán đó với các ảnh khác.

- Môi trường làm việc của Vision Assistant:



**Hình 15.45 Giao diện Vision assistant**

1.Cửa sổ làm việc; 2.Bảng các chức năng xử lý; 3.Tỉ lệ thu phóng; 4.Kích thước ảnh; 5.Cửa sổ mã; 6. Cửa sổ chính; 7.Các thẻ thao tác ảnh

**Cửa sổ làm việc (Reference Window)** – hiển thị ảnh đang được xử lý.

**Cửa sổ mã (Script window)** – cửa sổ ghi lại các bước xử lý ảnh.

**Cửa sổ xử lý ảnh (Processing Window)** – xử lý ảnh để được các kết quả mong muốn.

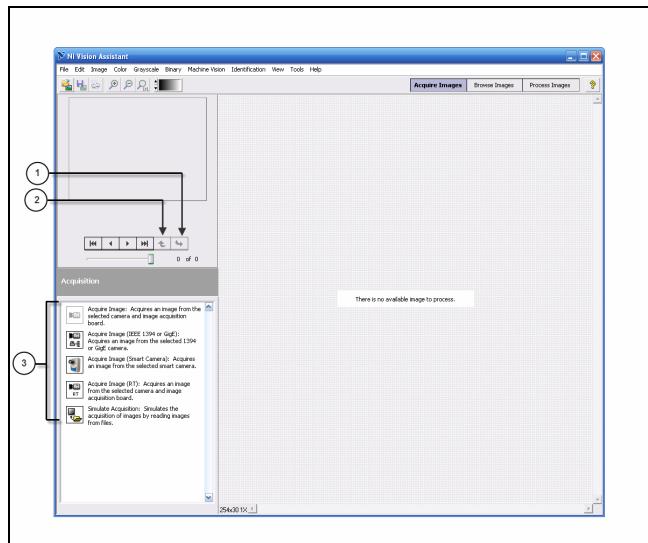
**Duyệt ảnh (Image Browser)** – gồm các ảnh được load vào trong Vision Assistant.

#### 14.4.2 Thu nhận hình ảnh với Vision Assistant:

Vision Assistant cho phép sử dụng 3 loại thu nhận hình ảnh là thu nhận và hiển thị 1 hình(snap), thu nhận và hiển thị hình ảnh một cách liên tục (grab) và thu nhận và hiển thị theo các thiết lập (sequence).

Vision Assistant hỗ trợ nhiều loại camera khác nhau như : các loại camera thông minh của NI,các camera IEEE 1394 và camera Gigabit Ethernet (GigE).

- **Cửa sổ thu nhận hình ảnh:**



**Hình 15.46 Cửa sổ thu nhận ảnh**

1.Đưa ảnh hoạt động; 2.Sắp xếp ảnh thu nhận vào bộ duyệt; 3.Các chức năng thu nhận ảnh

- **Thu nhận và hiển thị hình đơn(Snapping an Image) :**

1. Chọn **File»Acquire Image** và lựa chọn các chức năng thu nhận hình ảnh

2. Chọn nút **Acquire Single Image** để thu nhận hình đơn

3. Chọn nút **Store Acquired Image in Browser** để đưa ảnh vào bộ duyệt ảnh

- **Thu nhận và hiển thị hình ảnh một cách liên tục(Grabbing an Image):**

1. Chọn **File»Acquire Image** và chọn các chức năng thu nhận hình ảnh

2. Chọn nút **Acquire Continuous Images** để thu nhận và hiển thị hình ảnh 1 cách liên tục.Chọn nút này 1 lần nữa để dừng thu nhận hình ảnh.

3. Chọn nút **Store Acquired Image in Browser** để đưa ảnh vào bộ duyệt ảnh.

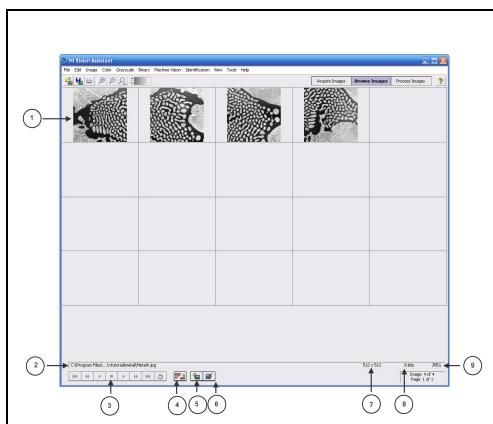
- **Thu nhận và hiển thị theo các thiết lập (Acquiring a Sequence of Images):**

1. Chọn **File»Acquire Image** và chọn các chức năng thu nhận hình ảnh.
2. Chọn nút **Sequence Acquisition**
3. Thiết lập các thông số :
  - **Number of Frames**—số khung hình muốn thu nhận
  - **Skip Count**—số khung hình cần bỏ qua giữa các bước thu nhận.
  - **Line**—đường trigger vật lý.
  - **Action**—hoạt động của trigger.Bao gồm **Disabled**, **Trigger start of acquisition**, và **Trigger each image**
  - **Timeout**—thời gian trigger xảy ra (khoảng mili giây).
  - **Polarity**—được thiết lập nếu tín hiệu thu nhận của trigger là cạnh lên hoặc cạnh xuống.

#### 15.4.3 Xử lý ảnh với Vision Assistant :

Các bước làm việc với **Vision Assistant** :

1. Khởi động **Vision Assistant** từ **Start»All Programs»National Instruments Vision Assistant**.
2. Chọn hình ảnh cần xử lý bằng cách click **Open Image**
3. Chọn thẻ **Browse Images** trong **Vision Assistant** và chọn ảnh cần duyệt



*Hình 15.47 Thẻ Browse images*

1.Duyệt ảnh; 2.Vị trí của ảnh; 3.Nút điều khiển; 4.Chế độ xem chi tiết/toàn kích thước; 5.Mở ảnh; 6.Tắt ảnh đã lựa chọn; 7.Kích thước ảnh;  
8.Loại ảnh; 9.Dạng ảnh

4. Qua thẻ **Image Processing** lựa chọn các **Script** trong cửa sổ **Processing Functions** để xử lý ảnh
5. Kiểm tra chức năng xử lý ảnh của **Scripts** đã lựa chọn bằng cách click chọn nút **Run Once** 
- **Vision Assistant** có thể xuất ra file Labview, code C,...

#### Tạo file VI của LabView :

1. Chọn **Tools»Create LabVIEW VI**. Chọn phiên bản LabVIEW
2. Chọn tên, chọn **Current Script** để tạo VI từ script đã tạo trong Vision Assistant.
3. Chọn **Image File** để tạo VI lấy ảnh từ ổ cứng

#### Tạo chương trình C:

1. Chọn **Tools»Create C Code**. Chọn tên.
2. Chọn **Create Main Function** khi muốn tạo chương trình chính để kiểm tra các chức năng xử lý ảnh
  - **Main File Name:** điền tên file C
  - **Image Source:** chọn file ảnh nguồn được sử dụng trong chương trình chính.
3. Chọn thư mục lưu file được tạo ra

#### 15.4.4 Sử dụng chức năng định vị màu sắc để nhận biết vị trí và số lượng led :

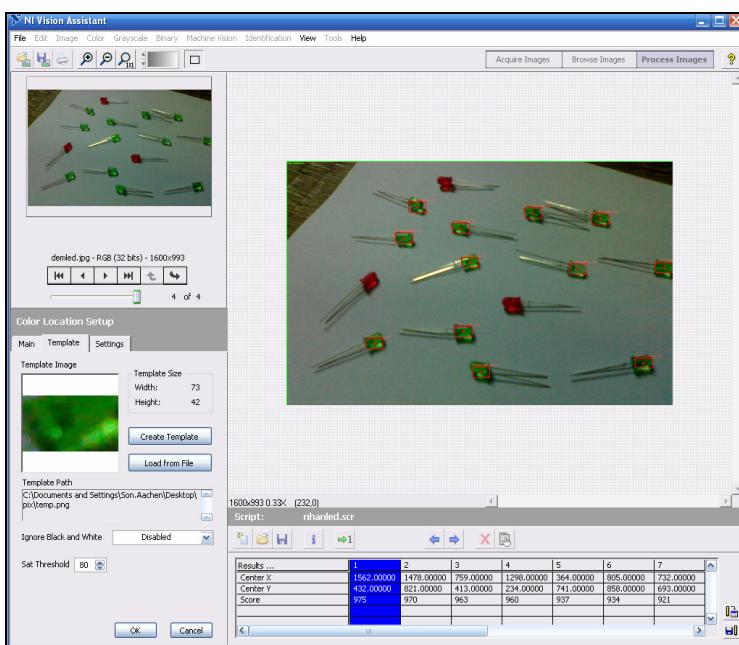
Dựa trên hình ảnh led xanh có sẵn thực hiện xử lý qua **Vision Assistant** để xác định số lượng led có trên hình và vị trí của led trên hình đó. Các hình ảnh có sẵn để xử lý :



*Hình 15.48 Chọn ảnh để xử lý*

Chọn ảnh bắt kì để xử lý, ở đây ta chọn ảnh demled(2).jp

### KẾT QUẢ :



**Hình 15.49 Kết quả sau khi xử lý**

Ta thấy được 12 led xanh và vị trí cụ thể của các led được khoanh thành các ô màu đỏ trên hình.

### **Bạn có biết?**

Thông thường code viết bằng LabVIEW chậm hơn nếu so sánh cùng một chương trình tương đương được soạn bằng mã C. Tuy nhiên tốc độ xử lý chương trình phần lớn phụ thuộc vào việc tối ưu chương trình mà bạn viết hơn là tốc độ xử lý của bản thân chương trình đó

### **15.5 Bài tập**

**Câu 1:** Thực hành tạo một tay robot 2 bậc tự do với LabVIEW

**Câu 2:** Viết một chương trình điều khiển LED với vi điều khiển ARM với LabVIEW

**Câu 3:** Thực hiện chương trình mô phỏng mobile robot với LabVIEW (tìm hiểu thêm thông tin tại <http://labview.hocdelam.org> để tải tài liệu mô phỏng mobile robot và phần mềm cần thiết cho công việc này).

**Câu 4:** Thực hành xử lý ảnh đếm số ngón tay chụp vào từ camera.

**Tham khảo bài 15**

- [1] TS. Nguyễn Bá Hải, KS. Lê Sĩ Toàn “*Lập trình 3D trong LabVIEW*”, Hocdelam Group, 2011.
- [2] LabVIEW programming with ARM, National Instrument
- [3] Đề tài Tiến sĩ, Nguyễn Bá Hải, Jee-Hwan Ryu, “Điều khiển mobile robot từ xa”, 2010, KUT, Hàn Quốc.
- [4] TS. Nguyễn Bá Hải, KS. Vũ Sơn, “Tài liệu xử lý ảnh”, Hocdelam Group, 2011
- [5] TS. Nguyễn Bá Hải, KS. Phan Ngọc Trung “*Thiết kế mô hình tập lái xe ô tô*”, Chuyên đề 2 - Đề tài Thạc Sĩ, ĐH Sư phạm Kỹ thuật Tp.HCM, 2011.
- [5] TS. Nguyễn Bá Hải, KS. Trần Trung “*Thiết kế mô hình tập lái xe xe tăng*”, đề tài nghiên cứu sinh, Viện Cơ học Việt Nam.

## BÀI 16

### HỖ TRỢ NGƯỜI HỌC

Phần này cung cấp cho bạn các tiện ích giúp bạn học tốt LabVIEW. Các tiện ích gồm: các bảng này có thể dùng để tra cứu hoặc photo dán lên những vị trí mà bạn hay quan sát.

#### 16.1 Bảng phím tắt bàn phím (Keyboard Shortcuts)

Trong LabVIEW, bạn có thể sử dụng các phím tắt để nhanh chóng thực hiện các thao tác lập trình.

Keyboard Shortcut	Description
<b>Object/Movement</b>	
Shift-click	Selects multiple objects; adds object to current selection.
Arrow keys	Moves selected objects one pixel at a time.
Shift-Arrow keys	Moves selected objects several pixels at a time.
Shift-click (drag)	Moves selected objects in one axis.
Ctrl-click (drag)	Duplicates selected object.
Ctrl-Shift-click (drag)	Duplicates selected object and moves it in one axis.
Shift-resize	Resizes selected object while maintaining aspect ratio.
Ctrl-resize	Resizes selected object while maintaining center point.
Ctrl-Shift-resize	Resizes selected object while maintaining center point and aspect ratio.
Ctrl-drag a rectangle	Adds more working space to the front panel or block diagram.
Ctrl-A	Selects all front panel or block diagram items.
Ctrl-Shift-A	Performs last <u>alignment</u> operation on objects.
Ctrl-D	Performs last <u>distribution</u> operation on objects.

Double-click open space	Adds a <a href="#">free label</a> to the front panel or block diagram if automatic tool selection is enabled.
Ctrl-mouse wheel	<a href="#">Scrolls through subdiagrams</a> of a Case, Event, or Stacked Sequence structure.
Spacebar (drag)	Disables <a href="#">preset alignment positions</a> when moving object labels or captions.

### Navigating the LabVIEW Environment

Ctrl-F	Finds objects or text.
Ctrl-Shift-F	Displays <a href="#">Search Results</a> window or the <a href="#">Find Project Items</a> dialog box with the previous search results.
Ctrl-G	Searches for next instance of object or text.
Ctrl-Shift-G	Searches for previous instance of VIs, object, or text.
Ctrl-Tab	Cycles through LabVIEW windows in the order the windows appear onscreen. (Linux) The order of the windows depends on the window manager you use.
Ctrl-Shift-Tab	Cycles the opposite direction through LabVIEW windows.
Ctrl-L	Displays the <a href="#">Error List</a> window.
Ctrl-Shift-W	Displays the <a href="#">All Windows</a> dialog box.

### Navigating Front Panels and Block Diagrams

Ctrl-E	Displays block diagram or front panel window.
Ctrl-#	Enables or disables grid alignment. On French keyboards, press the <Ctrl-"/> keys. (Mac OS) Press the <Command-*> keys.
Ctrl-/	Maximizes and restores window.
Ctrl-T	Tiles front panel and block diagram windows.
Ctrl-Shift-N	Displays the <a href="#">Navigation</a> window.
Ctrl-I	Displays the <a href="#">VI Properties</a> dialog box.

Ctrl-Y              Displays the [History](#) window.

#### Navigating the VI Hierarchy Window

Ctrl-D              Redraws the [VI Hierarchy](#) window.

Ctrl-A              Shows all VIs in the VI Hierarchy window.

Ctrl-click VI      Displays the subVIs and other nodes that make up the VI you select in the VI Hierarchy window.

Enter              [Finds the next node](#) that matches the search string after initiating a search by typing in the VI Hierarchy window.

Shift-Enter        [Finds the previous node](#) that matches the search string after initiating a search by typing in the VI Hierarchy window.

#### Debugging

Ctrl-Down arrow [Steps into node](#).

Ctrl-Right arrow [Steps over node](#).

Ctrl-Up arrow [Steps out of node](#).

#### File Operations

Ctrl-N              Opens a new, blank VI.

Ctrl-O              Opens an existing VI.

Ctrl-W              Closes the VI.

Ctrl-S              Saves the VI.

Ctrl-Shift-S        Saves all open files.

Ctrl-P              Prints the window.

Ctrl-Q              Quits LabVIEW.

#### Basic Editing

Ctrl-Z              Undoes last action.

Ctrl-Shift-Z        Redoes last action.

Ctrl-X Cuts an object.

Ctrl-C Copies an object.

Ctrl-V Pastes an object.

### Help

Ctrl-H Displays [Context Help](#) window. (Mac OS) Press the <Command-Shift-H> keys.

Ctrl-Shift-L Locks [Context Help](#) window.

Ctrl-? or <F1> Displays [LabVIEW Help](#).

### Tools and Palettes

Ctrl Switches to next most useful tool.

Shift Switches to Positioning tool.

Ctrl-Shift over open space Switches to Scrolling tool.

Spacebar Toggles between two most common tools if [automatic tool selection is disabled](#).

Shift-Tab Enables automatic tool selection.

Tab Cycles through four most common tools if you [disabled automatic tool selection](#) by clicking the Automatic Tool Selection button. Otherwise, enables automatic tool selection.

Arrow keys Navigates temporary Controls and Functions palettes.

Enter Navigates into a temporary palette.

Esc Navigates out of a temporary palette.

Shift-right-click Displays a temporary version of the [Tools palette](#) at the location of the cursor.

### SubVIs

Double-click subVI [Displays subVI front panel](#).

Ctrl-double-click subVI	<a href="#">Displays subVI block diagram and front panel.</a>
Drag VI icon to block diagram	<a href="#">Places that VI</a> as a subVI on the block diagram.
Shift-drag VI icon to block diagram	<a href="#">Places that VI</a> as a subVI on the block diagram with constants wired for controls that have non-default values.
Ctrl-right-click block diagram and select VI from palette	Opens front panel of that VI.

### Execution

Ctrl-R	<a href="#">Runs the VI.</a>
Ctrl - .	Stops the VI, when used while VI is running.
Ctrl-M	Changes to run or edit mode.
Ctrl-Run button	Recompiles the current VI.
Ctrl-Shift-Run button	Recompiles all VIs in memory.
Ctrl-Down arrow	<a href="#">Moves key focus inside</a> an array or cluster, when used while VI is running.
Ctrl-Up arrow	<a href="#">Moves key focus outside</a> an array or cluster, when used while VI is running.
Tab	Navigates the controls or indicators according to <a href="#">tabbing order</a> , when used while VI is running.
Shift-Tab	Navigates backward through the controls or indicators, when used while VI is running.
Wiring	
Ctrl-B	Deletes all <a href="#">broken wires</a> in a VI.
Esc, right-click, or click terminal	Cancels a wire you started.

Single-click wire Selects one segment.

Double-click wire Selects a branch.

Triple-click wire Selects entire wire.

A While wiring, disables automatic wire routing temporarily.

Double-click (while wiring) Tacks down wire without connecting it.

Spacebar [Toggles automatic wiring](#) while moving objects.

Shift-click Undoes last point where you set a wire.

Ctrl-click input on function with two inputs Switches the two input wires.

Spacebar Switches the direction of a wire between horizontal and vertical.

## Text

Double-click Selects a single word in a string.

Triple-click Selects an entire string.

Ctrl-Right arrow Moves forward in string by one word when using text that has characters that use single bytes, such as Western character sets. Moves forward in string by one character when using text that has characters that use multiple bytes, such as Asian character sets.

Ctrl-Left arrow Moves backward in string by one word when using text that has characters that use single bytes, such as Western character sets. Moves backward in string by one character when using text that has characters that use multiple bytes, such as Asian character sets.

Home Moves to beginning of current line in string.

End Moves to end of current line in string.

Ctrl-Home	Moves to beginning of entire string.
Ctrl-End	Moves to end of entire string.
Shift-Enter	Adds new items when entering items in enumerated type controls and constants, ring controls and constants, or Case structures. In <a href="#">string constants</a> , press the <Shift-Enter> keys on the keyboard to disable autosizing if it is enabled. If autosizing is disabled, press the <Shift-Enter> keys on the keyboard to display a scroll bar in the constant.
Esc	Cancels current edit in a string.
Ctrl-Enter	Ends text entry.
Ctrl-=	Increases the current font size.
Ctrl--	Decreases the current font size.
Ctrl-0	Displays the <a href="#">Font</a> dialog box.
Ctrl-1	Changes to the <a href="#">Application</a> font in Font dialog box.
Ctrl-2	Changes to the <a href="#">System</a> font in the Font dialog box.
Ctrl-3	Changes to the <a href="#">Dialog font</a> in the Font dialog box.
Ctrl-4	Changes to the <a href="#">current</a> font in the Font dialog box.

## 16.2 Quy tắc vàng

Bạn cũng có thể in 3 quy tắc vàng và dán vào nơi thường đọc để nhớ:

**Quy Tắc Vàng Trong LabVIEW**

**RC> Select, Create, Replace, Change to**

**Ctrl + H: Tra cứu sơ đồ chân**

**Search: Tìm kiếm hàm và đối tượng**

## **Ban có biết?**

Tác giả thật sự vui khi nhận được các góp ý của bạn đọc để tiếp tục hoàn thiện các tập sách về LabVIEW. Mỗi góp ý hay sẽ nhận lại được một cuốn sách trong lần tái bản tới đây. Bạn cũng có thể cùng tham gia viết các tập sách tiếp theo về LabVIEW.

Liên lạc với tác giả qua: [bahai@hcmute.edu.vn](mailto:bahai@hcmute.edu.vn). Chúc quý bạn đọc xây dựng được các ứng dụng hay nhất của mình với LabVIEW.

### 16.3 Ghi chú

## **--GHI CHÚ--**

Tủ Sách Hocdelam Group  
**CHÚC BẠN THÀNH CÔNG**



## TRÍCH NGANG VỀ TÁC GIẢ

Tiến sĩ Nguyễn Bá Hải sinh năm 1983, tốt nghiệp Đại học Ngành Cơ khí Động lực-ĐH Sư phạm Kỹ thuật Tp.HCM, Thạc sĩ và Tiến sĩ chuyên ngành Biorobotics (Robot sinh học) tại phòng thí nghiệm Biorobotics, trường ĐH Công nghệ và Giáo dục Hàn Quốc.

Hiện tại ông đang giảng dạy tại ĐH Sư phạm Kỹ thuật Tp.HCM với vai trò trưởng phòng thí nghiệm Cơ điện tử Ô tô.

Chuyên môn của ông gồm: Thu thập dữ liệu và điều khiển thiết bị, điều khiển phương tiện từ xa có phản hồi lực, giao diện giữa người và máy (HMI), mô phỏng và điều khiển hệ thống. Ông đã có 4 phát minh sáng chế được cấp bằng tại Hàn Quốc, nhiều công trình công bố tại hiệp hội kỹ sư điện-điện tử Hoa Kỳ (IEEE), FISITA, Tạp chí ô tô quốc tế IJAT, vv. Hiện tại ông trực tiếp hướng dẫn và tư vấn cho nhiều đề tài Tiến sĩ và Thạc sĩ trong ngành Kỹ thuật cho các trường đại học trong và ngoài nước. Đồng thời, tham gia bình duyệt (review) tại các hiệp hội kỹ thuật quốc tế.

Là người đã sáng lập nhóm Hocdelam Group và cùng Hocdelam đã tổ chức hàng trăm khóa học cho kỹ sư, giảng viên và sinh viên các trường Đại học, Cao đẳng, THCN, và doanh nghiệp nước ngoài cũng như trong nước như Fujikura, Memsonics, Trường Hải (THACO), Quatest3, ĐH BK TP.HCM, ĐH GTVT Hà Nội, CĐ CĐ Bà Rịa Vũng Tàu, Trung cấp nghề Bình Thuận, vv. Từ 2010, Ông thiết kế chương trình khóa học 1 đô la, ông đã cùng nhóm Hocdelam giúp hơn 400 cá nhân tiếp cận được với kiến thức về: thu thập dữ liệu, điều khiển thiết bị, trực quan hóa việc học kỹ thuật, nâng cao năng lực tìm kiếm tài nguyên Internet, tự học hiệu quả, kỹ năng mềm trong học tập và nghiên cứu, xây dựng dự án trong kỹ thuật, vv. Chương trình này cũng đã được giới thiệu qua các Báo và Đài truyền hình trong cả nước.

Sau nhiều năm tìm hiểu và sử dụng LabVIEW, ông đã thực hiện và đồng thực hiện thành công hơn 40 dự án sử dụng LabVIEW trong giáo dục, công nghiệp, y sinh, cơ điện tử, vật lý, tự động hóa và ô tô. Tư vấn cho nhiều trường ĐH xây dựng và khai thác các phòng thí nghiệm hiệu quả. Tập sách LabVIEW căn bản này ra đời nhằm phần nào hỗ trợ người yêu thích LabVIEW bắt đầu hành trình học tập và làm chủ LabVIEW một cách đơn giản và dễ dàng nhất.

**HÌNH ẢNH MỘT SỐ SẢN PHẨM SỬ DỤNG LabVIEW**

**Liên hệ mua các tập sách tiếp theo tại:**

**Hocdelam Group**

Phòng TN Cơ điện tử Ô tô – Khoa Công Nghệ Kỹ thuật Ô tô  
ĐH Sư phạm Kỹ thuật Tp.HCM. Số 01-Võ Văn Ngân-Thủ Đức-Tp.HCM  
Hotline: 098 8868 524 | Email: supports@hocdelam.org

## MỤC LỤC

### Bài 1: Giới thiệu LabVIEW

1.1 LabVIEW là gì?	05
1.2 Các ứng dụng của LabVIEW	06
1.3 Download và cài đặt LabVIEW	11
1.4 Phương pháp để học LabVIEW hiệu quả	16
1.5 Bài tập	18

### Bài 2: Các phép toán trong LabVIEW

2.1 Chương trình đầu tiên: Phép cộng hai số x,y	19
2.2 Cấu trúc một bài trình LabVIEW	24
2.3 Ba quy tắc vàng khi lập trình LabVIEW	25
2.4 Các phép toán thông thường	26
2.5 Lưu file, mở file, tìm ví dụ, công cụ trợ giúp	27
2.6 Phần đọc thêm	29
2.7 Bài tập	30

### Bài 3: Các loại Control và Indicator

3.1 Các control thường dùng	31
3.2 Các indicator thường dùng	33
3.3 Kiểu dữ liệu trong LabVIEW và chuyển đổi dữ liệu	35
3.4 Bài tập	36

### Bài 4: Vòng lặp while, vòng lặp for

4.1 Vòng lặp while	37
4.2 Vòng lặp for	39
4.3 Ứng dụng vòng lặp	40

4.4 Bài tập	42
<b>Bài 5: Cấu trúc case</b>	
5.1 Cấu trúc case	43
5.2 Ứng dụng cấu trúc case	44
5.3 Bài tập	49
<b>Bài 6: Các hàm và công cụ hữu dụng thường gặp khác</b>	
6.1 Mô phỏng tín hiệu	51
6.2 Local variable	54
6.3 Formulas	57
6.4 Bài tập	60
<b>Bài 7: Mảng và bó</b>	
7.1 Mảng	61
7.2 Bó	65
7.3 Bài tập	69
<b>Bài 8: Tạo subVI, định thời gian, và lưu dữ liệu</b>	
8.1 Tạo và sử dụng subVI	71
8.2 Định thời gian trong LabVIEW	77
8.3 Lưu dữ liệu	79
8.4 Bài tập	82
<b>Bài 9: Giao diện người dùng (GUI)</b>	
9.1 Khái niệm về giao diện người dùng	89
9.2 Xây dựng một GUI thẩm mỹ và hiệu quả	85
9.3 Tạo file .exe (application file)	88
9.4 Bài tập	88

## Bài 10: Xây dựng các ứng dụng với card Hocdelam USB-9001 và HDL-9000

10.1 Giới thiệu card Hocdelam USB-9001	89
10.2 Các ứng dụng với Card Hocdelam USB-9001	91
10.3 Điều khiển PID động cơ DC	102
10.4 Bài tập	109

## Bài 11: Thu thập dữ liệu với card NI USB-6008/6009

11.1 Giới thiệu card NI USB 6008/6009	111
11.2 Hướng dẫn lập trình Card NI USB6008/6009	114
11.3 Thực hành lập trình LabVIEW đo analog với card NI USB-6008	117
11.4 Kỹ thuật phát xung tín hiệu với card NI USB-6008/6009	120
11.5 Bài tập	124

## Bài 12: Điều khiển chuyển động với họ card NI PCI 7350

12.1 Chuẩn bị thiết bị phần cứng và phần mềm	125
12.2 Cài đặt phần mềm	126
12.3 Thủ phần mềm và phần cứng	130
12.4 Thiết lập các thông số trong MAX	134
12.5 Thủ động cơ và chương trình điều khiển động cơ một chiều DC bằng Analog output	135
12.6 Điều khiển chuyển động	137
12.7 Một số lỗi thường gặp với card PCI 7350	142
12.8 Bài tập	144

## Bài 13: Các chuyên đề mở rộng

13.1 Giao tiếp RS232	147
13.2 Thực hành giao tiếp RS232	148

13.3 Bài tập	152
<b>Bài 14: Các chuyên đề mở rộng</b>	
14.1 Giao tiếp TCP/IP	153
14.2 Ví dụ Lập trình TCP/IP với LabVIEW	155
14.3 Bài tập	156
<b>Bài 15: Các chuyên đề mở rộng</b>	
15.1 Điều khiển hình ảnh (2D và 3D) trong LabVIEW	157
15.2 Lập trình Vi điều khiển ARM trong LabVIEW	169
15.3 Giải pháp về thời gian thực trong LabVIEW	171
15.4 Xử lý ảnh với LabVIEW	177
15.5 Bài tập	183
<b>Bài 16: Hỗ trợ người học</b>	
16.1 Bảng phím tắt	185
16.2 Quy tắc vàng	191
16.3 Ghi chú	192