

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG



ISO 9001 : 2008

ĐỒ ÁN TỐT NGHIỆP

NGÀNH: ĐIỆN TỬ VIỄN THÔNG

Người hướng dẫn: CN. Nguyễn Huy Dũng

Sinh viên: Phạm Trung Hiếu

HẢI PHÒNG 2013

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

**ỨNG DỤNG LABVIEW ĐIỀU KHIỂN Lò NHIỆT
GHÉP NỐI VỚI MÁY TÍNH**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC CHÍNH QUY
NGÀNH: ĐIỆN TỬ VIỄN THÔNG**

Người hướng dẫn: CN. Nguyễn Huy Dũng

Sinh viên: Phạm Trung Hiếu

HẢI PHÒNG 2013

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên : Phạm Trung Hiếu.

Mã SV: 1351030011

Lớp : ĐT1301.

Ngành: Điện tử viễn thông

Tên đề tài: Ứng dụng LabVIEW điều khiển lò nhiệt ghép nối
với máy tính

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp (về lý luận, thực tiễn, các số liệu cần tính toán và các bản vẽ).

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Các số liệu cần thiết để thiết kế, tính toán.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. Địa điểm thực tập tốt nghiệp.

.....

.....

.....

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ nhất:

Họ và tên: Nguyễn Huy Dũng.

Học hàm, học vị: Cử nhân.

Cơ quan công tác: Trường Đại học Dân lập Hải Phòng.

Nội dung hướng dẫn:.....

.....

.....

.....

Người hướng dẫn thứ hai:

Họ và tên:

Học hàm, học vị:

Cơ quan công tác:.....

Nội dung hướng dẫn:.....

.....

.....

.....

Đề tài tốt nghiệp được giao ngày.....tháng.....năm 2013

Yêu cầu phải hoàn thành xong trước ngày.....tháng.....năm 2013

Đã nhận nhiệm vụ ĐTTN

Sinh viên

Đã giao nhiệm vụ ĐTTN

Người hướng dẫn

Hải Phòng, ngày tháng.....năm 2013

Hiệu trưởng

GS.TS.NGŨT Trần Hữu Nghị

PHẦN NHẬN XÉT CỦA CÁN BỘ HƯỚNG DẪN

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Đánh giá chất lượng của khóa luận (so với nội dung yêu cầu đã đề ra trong nhiệm vụ Đ.T. T.N trên các mặt lý luận, thực tiễn, tính toán số liệu...):

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. Cho điểm của cán bộ hướng dẫn (ghi bằng cả số và chữ):

.....

.....

.....

Hải Phòng, ngày.....tháng.....năm 2013

Cán bộ hướng dẫn

PHẦN NHẬN XÉT TÓM TẮT CỦA NGƯỜI CHĂM PHẢN BIỆN

1. Đánh giá chất lượng đề tài tốt nghiệp về các mặt thu thập và phân tích số liệu ban đầu, cơ sở lý luận chọn phương án tối ưu, cách tính toán chất lượng thuyết minh và bản vẽ, giá trị lý luận và thực tiễn đề tài.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Cho điểm của cán bộ phản biện (Điểm ghi cả số và chữ).

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Hải Phòng, ngày.....tháng.....năm 2013
Người chấm phản biện

MỤC LỤC

LỜI NÓI ĐẦU	1
CHƯƠNG 1: TỔNG QUAN VỀ NGÔN NGỮ LẬP TRÌNH LABVIEW	4
1.1: Tổng quan về labVIEW	4
1.1.1. LabVIEW là gì?	4
1.1.2. Vai trò của LabVIEW	4
1.1.3. Các chức năng chính của LabVIEW	5
1.1.4. Phần mềm nhúng vào LabVIEW	5
1.1.5. Các giao thức kết nối.....	6
1.1.6. Các Module và bộ công cụ LabVIEW	6
1.1.6.1. Các module LabVIEW	6
1.1.6.2. Các bộ công cụ LabVIEW	7
1.1.7. LabVIEW làm việc như thế nào?.....	7
1.2. Các thành phần của LabVIEW	8
1.2.1. Bảng giao diện (The Front panel)	8
1.2.2. Sơ đồ khối (The Block Diagram)	11
1.3. Những công cụ lập trình LabVIEW	13
1.3.1. Tools Palette	13
1.3.2. Bảng điều khiển (Controls Palette)	14
1.3.3. Bảng các hàm chức năng (Function palette)	16
1.4. Các loại Control và Indicator	24
1.4.1. Các Control thường dùng	24
1.4.2. Các dạng Indicator thường dùng	26
1.4.3. Kiểu dữ liệu trong LabVIEW và chuyển đổi dữ liệu	29
1.5. Vòng lặp While (While Loop), vòng lặp For (For Loop)	30
1.5.1. Vòng lặp While (While Loop)	30
1.5.2. Vòng lặp For (For Loop).....	32
1.6. Mảng.....	33
1.6.1. Khái niệm về mảng và cách tạo mảng	33
1.6.2. Trích dữ liệu từ một mảng.....	35
1.7. Bó	37
1.8. Cách tạo thiết bị ảo và thiết bị ảo con.	40
CHƯƠNG 2 ĐIỀU KHIỂN Lò NHIỆT ĐIỆN TRỞ	47
2.1. Giới thiệu.....	47

2.2.Ưu nhược điểm của lò điện so với các lò sử dụng nhiên liệu.....	47
2.3.Nguyên lý làm việc của lò điện trở	48
2.4.Các phương pháp điều khiển lò điện trở	49
2.4.1.Điều khiển dùng Role	49
2.4.2.Điều khiển Thyristor	50
2.4.3.Kết luận	52
2.5.Các nguyên tắc điều khiển Thyristor (Triac)	53
2.5.1. Nguyên tắc điều khiển thẳng đứng tuyến tính	53
2.5.2. Nguyên tắc điều khiển thẳng đứng “arccos”	54
2.5.3. Sơ đồ khối mạch điều khiển.....	55
CHƯƠNG 3 ĐIỀU KHIỂN LÒ NHIỆT ĐIỆN TRỞ GIAO TIẾP VỚI MÁY	
TÍNH BẰNG LABVIEW.....	57
3.1.Phương án thiết kế.....	57
3.1.1.Yêu cầu thiết kế.....	57
3.1.2Phương pháp điều khiển.....	57
3.2: Giới thiệu Card USB-9001.....	58
3.2.1: Thông số kỹ thuật.....	58
3.2.2Cách sử dụng.....	60
3.3 Mô hình điều khiển sử dụng card USB-9001	61
3.3.1. Sensor LM35	61
3.3.2.Xây dựng mô hình điều khiển:.....	62
3.4: Chương trình điều khiển bằng ngôn ngữ LabVIEW.....	63
CHƯƠNG 4: KẾT LUẬN – HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI	65
4.1.Các kết quả đã thực hiện được trong đề tài.....	65
4.2.Những hạn chế.....	65
4.3.Hướng phát triển của đề tài	65
TÀI LIỆU THAM KHẢO	67

DANH MỤC HÌNH

Hình 1.1 Phạm vi ứng dụng của LabVIEW.....	5
Hình 1.2 Các giao thức kết nối của LabVIEW.....	6
Hình 1.3 Bảng giao diện mới	9
Hình 1.4 Mô tả tên của VI hiện thời đang tải	9
Hình 1.5 Thanh công cụ giao diện	10
Hình 1.6 Sơ đồ khối của LabVIEW	12
Hình 1.7 Bảng Tool Palette.....	13
Hình 1.8 Bảng mẫu Controls	15
Hình 1.9 Bảng điều khiển và chỉ thị số	15
Hình 1.10 Bảng điều khiển và chỉ thị logic.....	16
Hình 1.11 Bảng Graph.....	16
Hình 1.12 Bảng Functions	17
Hình 1.13 Hàm cấu trúc- Structures Function	17
Hình 1.14 Hàm mảng – Function Array	18
Hình 1.15 Hàm cụm & biến thể – Cluter & Variant.....	18
Hình 1.16 Hàm số học – Numeric Function	19
Hình 1.17 Hàm Boolean.....	19
Hình 1.18 Hàm chuỗi – String Function.....	20
Hình 1.19 Hàm so sánh – Comparison Functions	20
Hình 1.20 Hàm Thời gian – Time function	21
Hình 1.21 Hàm Dialog & User Interface	21
Hình 1.22 Hàm File I/O- File I/O Function.....	22
Hình 1.23 Hàm dạng sóng – Waveform.....	22
Hình 1.24 Hàm điều khiển ứng dụng- Application Control.....	23
Hình 1.25 Hàm đồng bộ hoá - Synchronization Function	23
Hình 1.26 Hàm đồ họa và âm thanh – Graphic & Sound Function	23
Hình 1.27 Hàm phát sinh báo cáo – Report Generation Function.....	24
Hình 1.28: Cách lấy Control.....	25
Hình 1.29: Copy nhanh bằng việc kéo thả.....	25
Hình 1.30: Cách lấy Indicator	27
Hình 1.31: Cách lấy Indicator	28
Hình 1.32: Cách lấy Indicator	28
Hình 1.33: Các kiểu dữ liệu trong LabVIEW.....	29

Hình 1.34: Lấy While Loop tại BD	30
Hình 1.35: Tính tổng với While Loop	31
Hình 1.36: Sử dụng shift register	32
Hình 1.37: For Loop	32
Hình 1.38: Mảng 1 chiều dạng số, 2 chiều dạng boolean, và 3 chiều dạng chuỗi	33
Hình 1.39: Lấy mảng từ FP	33
Hình 1.40: Tạo Numeric array.....	34
Hình 1.41: Kéo dài mảng	34
Hình 1.42: Copy nhanh một mảng	34
Hình 1.43: Chương trình hoàn thiện.....	35
Hình 1.44: Kết quả phép cộng 2 mảng	35
Hình 1.45: Lấy hàm Index array.....	36
Hình 1.46: Chương trình truy xuất giá trị thành phần trong array	36
Hình 1.47: Tổng 2 array	37
Hình 1.48: Tạo array 2 chiều.....	37
Hình 1.49: Kéo thả Cluster ra Front panel	38
Hình 1.50: Cluster đã tạo ra	38
Hình 1.51 Tạo Cluster indicator	39
Hình 1.52: Kết quả tạo ra là Cluster indicator sẽ hiển thị đúng Cluster control	39
Hình 1.53: Phương pháp tạo Cluster bằng hàm Bundle	39
Hình 1.54 Bóc tách các phần tử trong một bó	40
Hình 1.55 Ví dụ minh họa	41
Hình 1.56 Cửa sổ Icon Editor	42
Hình 1.57: Ví dụ minh họa vẽ Icon	42
Hình 1.58a: Các bước vẽ Icon	43
Hình 1.58b: Các bước tạo Icon.....	43
Hình 1.58c: Các bước tạo Icon.....	44
Hình 2.1 . Sơ đồ điều khiển bằng Role.....	49
Hình 2.2 . Đồ thị quan hệ giữa nhiệt độ và công suất cấp	50
Hình 2.3 . Sơ đồ điều khiển bằng Thyristor	50
Hình 2.4 . Dạng điện áp ra điều khiển bằng Thyristor.....	51
Hình 2.5. Nguyên tắc điều khiển thẳng đứng tuyến tính.....	54
Hình 2.6. Nguyên tắc điều khiển thẳng đứng.....	55

<i>Hình 2.7. Sơ đồ khối mạch điều khiển</i>	55
<i>Hình 3.1. Mạch điều khiển lò điện trở</i>	57
<i>Hình 3.2. Sơ đồ chức năng Card USB-9001</i>	61
<i>Hình 3.3. Điều khiển lò nhiệt điện trở ghép nối với máy tính</i>	62
<i>Hình 3.4. Sơ đồ ghép nối phân cứng điều khiển lò nhiệt điện trở</i>	63

LỜI NÓI ĐẦU

Điều khiển là một lĩnh vực quan trọng của đời sống xã hội, của nền kinh tế quốc dân, của khoa học kỹ thuật và của nền đại công nghiệp. Bất cứ ở vị trí nào, bất cứ làm một công việc gì mỗi người trong chúng ta đều tiếp cận với điều khiển. Nó là khâu quan trọng quyết định sự thành bại trong mọi hoạt động của chúng ta.

Nền đại công nghiệp hiện nay càng ngày càng được nâng cao, mức độ tự động hóa với mục đích nâng cao năng suất lao động, giảm chi phí sản xuất, giải phóng con người ra khỏi những vị trí làm việc nguy hiểm và độc hại ... Để tiếp cận với nền đại công nghiệp có trình độ tự động hóa cao National Instruments là tiên phong đứng đầu về các dụng cụ ảo, phương pháp đo đạc và tự động hóa của các kỹ sư, nhà khoa học trong công nghiệp. LabVIEW nhằm nâng cao khả năng đo lường công nghiệp nhờ những đặc điểm mới được thiết kế cho những giao diện phân tích và điều khiển tiên tiến, quản lý hệ thống phân tán nâng cao và đích (target) mới cho giao diện người máy (HMI). Môi trường LabVIEW mở tương thích với mọi thiết bị đo với các trợ giúp tương tác, tạo mã nguồn và khả năng kết nối tới hàng nghìn thiết bị giúp tập hợp dữ liệu dễ dàng. Vì LabVIEW cung cấp tính kết nối tới hầu hết mọi thiết bị đo, nên bạn có thể dễ dàng kết hợp những ứng dụng LabVIEW mới vào các hệ thống hiện đại. Chính vì vậy mà LabVIEW ngày càng được ứng dụng rộng rãi và trở thành lựa chọn hàng đầu của các kỹ sư và các nhà khoa học trên toàn thế giới.

Mục đích nghiên cứu của đề án:

- Nghiên cứu về cách thức sử dụng phần mềm LabVIEW, tìm hiểu về các ứng dụng của LabVIEW như: cách tạo giao diện, lập trình sơ đồ khối, làm thế nào để tạo thiết bị ảo (VI) và thiết bị ảo con (Sub VI) ...
- Nghiên cứu về card USB- 9001, cách truyền và nhận tín hiệu từ phần mềm LabVIEW tới card USB- 9001.
- Điều khiển và ổn định nhiệt độ trong lò nhiệt bằng phương án điều khiển kiểu Role

Đối tượng và phạm vi nghiên cứu:

Với đề tài: “**Ứng dụng LabVIEW điều khiển lò nhiệt ghép nối với máy tính**”, đối tượng nghiên cứu của em bao gồm: nghiên cứu về phần mềm LabVIEW, card USB- 9001, và lò nhiệt điện trở. Thu thập tín hiệu từ cảm biến (cảm biến nhiệt độ), giao tiếp với PC thông qua card USB- 9001, phần mềm LabVIEW xử lý tín hiệu đưa vào, sau đó sẽ xuất tín hiệu ra để điều khiển nhằm ổn định nhiệt độ trong lò.

Ý nghĩa thực tiễn của đề án:

Nghiên cứu phần mềm LabVIEW với những ứng dụng rất rộng rãi, bởi vì bằng phần mềm chúng ta có thể thiết kế, điều khiển và kiểm tra như các phần cứng điều khiển và đo đạc. LabVIEW có khả năng kết nối tới rất nhiều thiết bị giúp tập hợp dữ liệu dễ dàng, đồng thời cung cấp tính kết nối tới hầu hết mọi thiết bị đo, vì vậy có thể dễ dàng kết hợp những ứng dụng LabVIEW mới vào các hệ thống hiện đại.

Ngày nay, cùng với sự phát triển của xã hội, ổn định nhiệt độ được ứng dụng rộng rãi trong công nghiệp và nông nghiệp. Trong công nghiệp, ổn định nhiệt độ được ứng dụng trong các lò nung thép, lò nung gạch, trong các đường ống dẫn dầu, lò hơi trong các nhà máy bia rượu... Trong nông nghiệp ổn định nhiệt độ được ứng dụng trong các lò ấp trứng, lò sấy và bảo quản các sản phẩm nông sản. Như vậy, việc ổn định nhiệt độ có vai trò quan trọng góp phần thúc đẩy sự phát triển của nền kinh tế.

Nội dung thực hiện đề tài của chúng em gồm 4 chương:

Chương 1: Tổng quan về ngôn ngữ lập trình LabVIEW.

Chương 2: Điều khiển lò nhiệt điện trở.

Chương 3: Điều khiển lò nhiệt giao tiếp với máy tính bằng phần mềm LabVIEW

Chương 4: Kết luận – Hướng phát triển của đề tài.

Do thời gian có hạn và kiến thức chuyên môn cũng như thực tế còn hạn chế nên bản đồ án này không tránh khỏi những thiếu sót. Kính mong nhận được sự chỉ bảo, góp ý của các thầy cô và các bạn để đồ án này được hoàn thiện hơn.

Em xin chân thành cảm ơn sự giúp đỡ chỉ bảo tận tình của các thầy, cô, đặc biệt là thầy Nguyễn Huy Dũng đã tạo điều kiện tốt nhất để em hoàn thành đồ án này. Em xin kính chúc các thầy cô luôn luôn mạnh khỏe để có thể tiếp tục dìu dắt nhiều thế hệ sinh viên.

Hải Phòng, tháng 07 năm 2013

Sinh viên thực hiện

Phạm Trung Hiếu

CHƯƠNG 1:

TỔNG QUAN VỀ NGÔN NGỮ LẬP TRÌNH LABVIEW

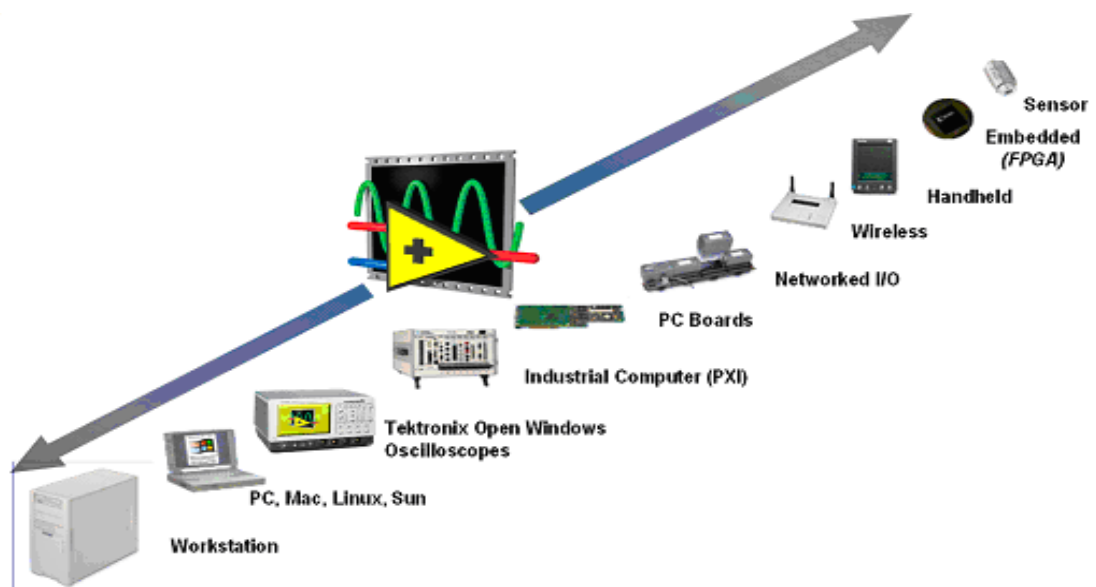
1.1: Tổng quan về labVIEW

1.1.1. LabVIEW là gì?

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) là ngôn ngữ lập trình đồ họa sử dụng các biểu tượng (Icon) thay cho những dòng lệnh để tạo ứng dụng.

1.1.2. Vai trò của LabVIEW

- Kiểm tra, đo kiểm và phân tích tín hiệu trong kỹ thuật (đo nhiệt độ, phân tích nhiệt độ trong ngày)
- Thu thập dữ liệu (Data Acquisition), (thu thập các giá trị áp suất, cường độ, dòng điện,...)
- Điều khiển các thiết bị (điều khiển động cơ DC, điều khiển nhiệt độ trong lò ...)
- Phân loại sản phẩm (dùng chương trình xử lý ảnh để phân biệt sản phẩm bị lỗi, phế phẩm)
- Báo cáo trong công nghiệp (thu thập, phân tích dữ liệu và báo cáo cho người quản lý ở rất xa thông qua giao thức truyền TCP/IP trong môi trường mạng Ethernet)
- Giao tiếp máy tính và truyền dẫn dữ liệu qua các cổng giao tiếp (hỗ trợ hầu hết các chuẩn giao tiếp như USB, PCI, COM, RS-232, RS-485)



Hình 1.1 Phạm vi ứng dụng của LabVIEW

1.1.3. Các chức năng chính của LabVIEW

- Thu thập tín hiệu từ các thiết bị bên ngoài như cảm biến nhiệt độ, hình ảnh từ webcam, vận tốc của động cơ...
- Giao tiếp với các thiết bị ngoại vi thông qua nhiều chuẩn giao tiếp như: RS232, RS485, USB, PCI, TCP/IP, Ethernet.
- Mô phỏng và xử lý các tín hiệu thu nhận được để phục vụ các mục đích nghiên cứu hay mục đích của hệ thống mà người lập trình mong muốn.
- Xây dựng các giao diện người dùng một cách nhanh chóng và thẩm mỹ hơn nhiều lần so với các ngôn ngữ như VB, Matlab, Visual C...
- Cho phép kết hợp với nhiều ngôn ngữ truyền thống như C, C++...
- Cho phép thực hiện các thuật toán điều khiển như PID, Logic mờ (Fuzzy).

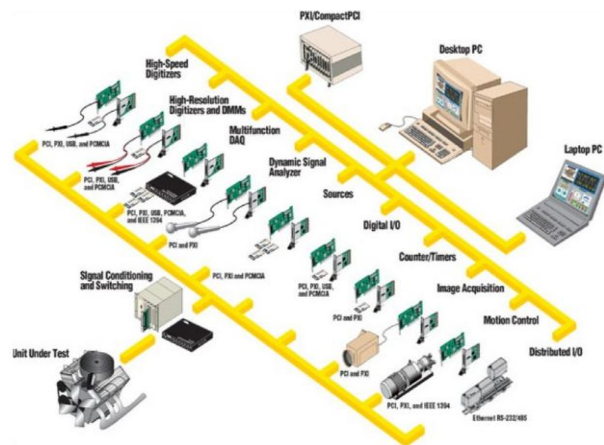
1.1.4. Phần mềm nhúng vào LabVIEW

- Wolfram Research Mathematica.
- Microsoft Excel.
- The MathWorks MATLAB and Simulink.
- MathSoft MathCAD.
- Electronic Workbench MultiSim.
- Texas Instruments Code Composer Studio.
- Ansoft RF circuit design software.

- Microsoft Access.
- Microsoft SQL Server.
- Oracle.

1.1.5. Các giao thức kết nối

- Ethernet
- CAN
- DeviceNet
- USB
- IEEE 1394
- RS-232
- GPIB
- RS-485



Hình 1.2 Các giao thức kết nối của LabVIEW

1.1.6. Các Module và bộ công cụ LabVIEW

1.1.6.1. Các module LabVIEW

Để tăng cường sức mạnh và mở rộng khả năng của bộ phần mềm phát triển LabVIEW, NI cung cấp thêm các module hỗ trợ đến nhiều loại phần cứng nhúng khác nhau:

- Module thời gian thực (LabVIEW Real-Time Module).
- Module FPGA.
- Module điều khiển giám sát và ghi dữ liệu (LabVIEW Datalogging and Supervisory Control Module).
- Module biểu đồ trạng thái (LabVIEW Statechart Module).
- Module mô phỏng và thiết kế bộ điều khiển (LabVIEW Control Design and Simulation Module).
- Module phát triển thị giác (NI Vision Development Module).
- Module cho màn hình cảm ứng và PDA (LabVIEW PDA and LabVIEW Touch Panel Module).
- LabVIEW DSP Module (xử lý tín hiệu số).

1.1.6.2. Các bộ công cụ LabVIEW

NI cũng thêm vào LabVIEW các bộ công cụ để đem lại các tiện ích khác nhau như: tạo báo cáo, phân tích nâng cao, thông tin liên lạc cơ sở dữ liệu, phân tích âm thanh và rung động.

- Bộ công cụ kết nối cơ sở dữ liệu (LabVIEW Database Connectivity Toolkit).
- Bộ công cụ xử lý tín hiệu nâng cao (LabVIEW Advanced Signal Processing Toolkit).
- Bộ đo lường âm thanh và rung động (LabVIEW Sound and Vibration Measurement Suite).
- Bộ công cụ nhận dạng hệ thống (LabVIEW System Identification Toolkit).
- Bộ công cụ tương tác mô phỏng (LabVIEW Simulation Interface Toolkit).
- Bộ công cụ theo dõi thực thi thời gian thực (LabVIEW Real-Time Execution Trace Toolkit).
- Bộ công cụ kết nối Internet (LabVIEW Internet Toolkit).
- Bộ công cụ điều biến (LabVIEW Modulation Toolkit).
- Bộ công cụ điều khiển PID (LabVIEW PID Control Toolkit).
- Bộ công cụ thiết kế bộ lọc số (LabVIEW Digital Filter Design Toolkit).

1.1.7. LabVIEW làm việc như thế nào?

LabVIEW được biết đến như là một ngôn ngữ lập trình với khái niệm hoàn toàn khác so với các ngôn ngữ lập trình truyền thống như ngôn ngữ C, Pascal... Bằng cách diễn đạt cú pháp thông qua các hình ảnh trực quan trong môi trường soạn thảo có sẵn hàng ngàn thư viện, hàm và cấu trúc lập trình, LabVIEW đã được gọi với tên khác là lập trình G (viết tắt của Graphical).

Những chương trình LabVIEW được gọi là những thiết bị ảo (Virtual Instruments – VIs), bởi vì hình dạng và cách hoạt động giống với những thiết bị vật lý, chẳng hạn như máy nghiệm dao động, máy hiện sóng...

Trong LabVIEW, bạn xây dựng giao diện người dùng bằng cách sử dụng một bộ các công cụ và đối tượng, và cửa sổ Front panel được xem như là giao

diện người dùng. Còn cửa sổ Block diagram chứa các hàm thao tác là các biểu tượng đồ họa, nơi mà dòng dữ liệu thực thi.

1.2. Các thành phần của LabVIEW

LabVIEW bao gồm các thư viện của các hàm chức năng và các công cụ phát triển được thiết kế đặc biệt dành cho thiết bị điều khiển. Các chương trình LabVIEW được gọi là những dụng cụ ảo bởi vì sự xuất hiện và hoạt động của chúng mô phỏng các dụng cụ thực tế. Các VI có cả 2 tương tác đó là: một tương tác giao diện người dùng và một mã nguồn tương đương, và truy nhập các tham số từ các VI tầng cao.

LabVIEW gồm có 3 thành phần chính đó là: bảng giao diện (The Front Panel), sơ đồ khối (The Block Diagram) và biểu tượng & đầu nối (The Icon - Connect).

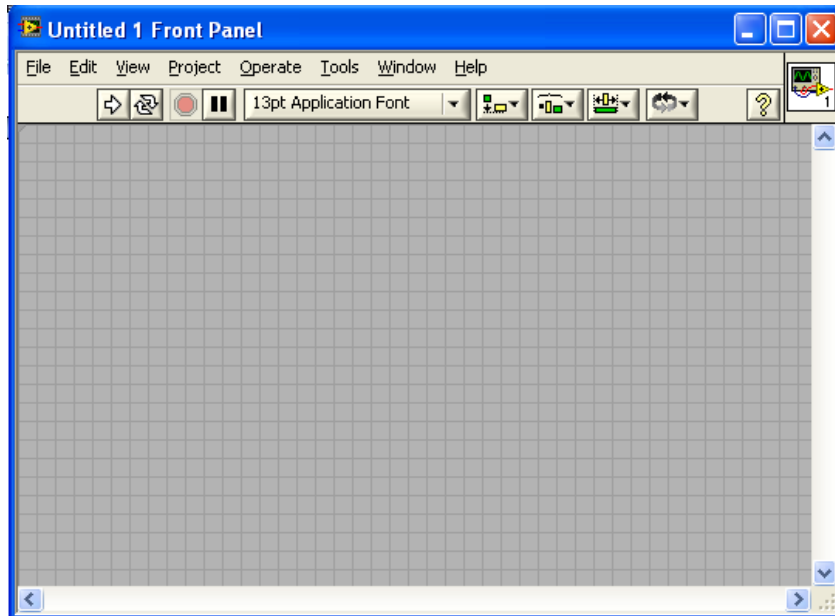
1.2.1. Bảng giao diện (The Front panel)

Front Panel là giao diện mà người sử dụng hệ thống nhìn thấy. Các VI bao gồm một giao diện người dùng có tính tương tác mà được gọi là bảng giao diện, vì nó mô phỏng mặt trước của một dụng cụ vật lý. Bảng giao diện có thể bao gồm các nút, các nút đẩy, các đồ thị và các dụng cụ chỉ thị và điều khiển khác. Bạn nhập vào dữ liệu sử dụng bàn phím và chuột rồi sau đó quan sát các kết quả trên màn hình máy tính.

Vào **Start>>All Programs>> National Instruments LabVIEW** một cửa sổ LabVIEW xuất hiện. Bạn tiếp tục chọn **Evaluate** và cửa sổ **Getting Started** sẽ xuất hiện ngay sau đó. Bạn chọn **Blank VI** để hiển thị bảng giao diện hoặc bạn có thể chọn **New** và sau đó hộp thoại **New** xuất hiện và trong hộp thoại đó mặc định con trỏ ở danh mục **Blank VI**. Để hiển thị bảng giao diện bạn chỉ cần kích vào nút **OK** ở phía góc phải dưới. Cả hai cách trên đều để mở bảng giao diện mới để bạn có thể xây dựng một VI mới hoàn toàn.

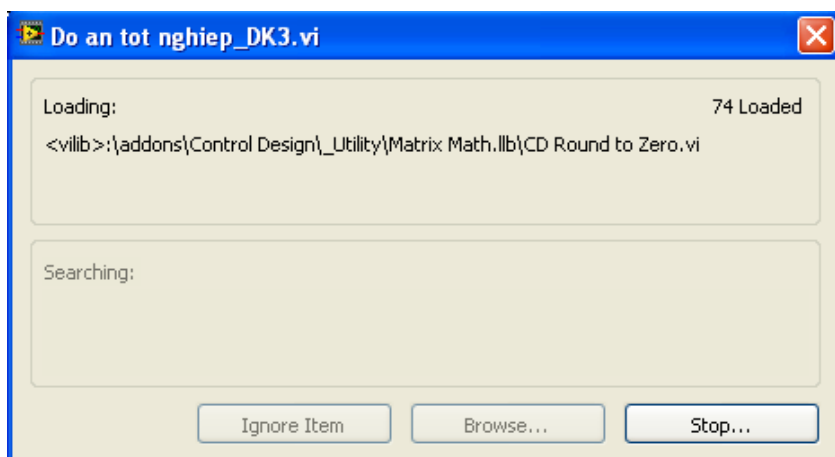
Ngoài ra bạn có thể mở một bảng giao diện có sẵn trong LabVIEW bằng cách trong hộp thoại **New**, từ mục **Create New**, lựa chọn **VI>>From template>>Tutorial (Getting Started)>>Generate and Display**. Và sau đó

kích nút **OK** để hiển thị bảng giao diện. Bảng giao diện sẽ xuất hiện như hình 1.3 sau đây:



Hình 1.3 Bảng giao diện mới

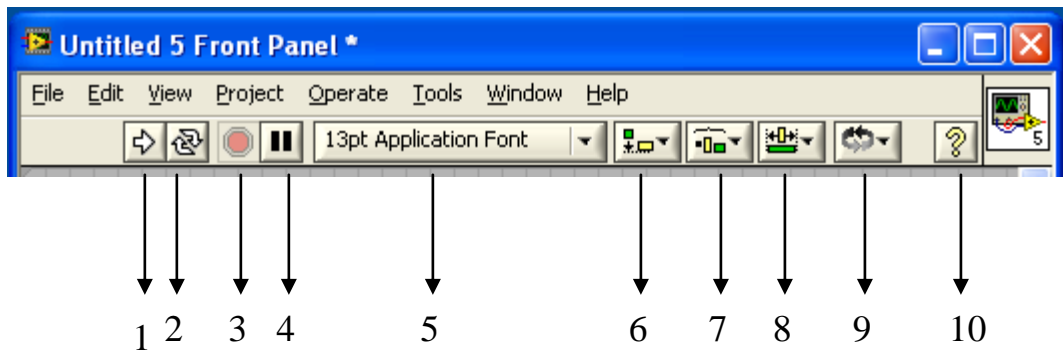
Ta cũng có thể mở bảng giao diện của một VI có sẵn trong thư viện LabVIEW bằng cách trong hộp thoại bảng giao diện vào **File>>Open** sau đó kích đúp vào các ví dụ có sẵn. Trong khi VI đang tải, một hộp thoại xuất hiện, cái mà mô tả tên của VI hiện thời đang tải, tên của điều khiển cứng mà VI được định vị trên đó, các thư mục và các đường dẫn đang được tìm kiếm, và số lượng VI trong quá trình tải. Hộp thoại xuất hiện như hình 1.4 bên dưới:



Hình 1.4 Mô tả tên của VI hiện thời đang tải

Trong bảng giao diện bao gồm một thanh công cụ của các nút lệnh và các dụng cụ chỉ báo trạng thái mà bạn sử dụng cho quá trình chạy và xử lý các VI.

Nó cũng bao gồm những tùy chọn phông và các tùy chọn phân phối và sắp thành hàng cho việc soạn thảo các VI.



Hình 1.5 Thanh công cụ giao diện

Trong đó:

1. Nút chạy chương trình (thanh không sáng – bị vỡ: lỗi, phải sửa lại chương trình)
2. Nút chạy lặp
3. Nút dừng cưỡng ép chương trình
4. Nút tạm dừng
5. Text setting (màu sắc, định dạng, kích thước- phông)
6. Gióng đều đối tượng theo hàng dọc và ngang
7. Phân bố các đối tượng
8. Thay đổi kích thước các đối tượng
9. Lệnh bổ sung
10. Cửa sổ trợ giúp

Các lưu ý khi hoạt động VI

1. Trong bảng giao diện, chạy VI bằng cách kích vào nút chạy trên thanh công cụ



Nút chạy thay đổi để chỉ báo rằng VI đang chạy



2. Sử dụng công cụ Operating để thay đổi các giá trị giới hạn cao và thấp. Đầu tiên chiếu sáng giá trị cũ, sau đó bằng việc tiếp tục nhấn đúp giá trị bạn muốn thay đổi, hoặc kích và kéo ngang qua giá trị với công cụ Labeling. Khi nào giá trị ban đầu được chiếu sáng, nhập một giá trị mới và nhấn <Enter>. Bạn cũng có

thể kích trên nút nhập vào trong thanh công cụ, hoặc kích chuột trong một vùng mở của cửa sổ để nhập vào giá trị mới.

3. Thay đổi điều khiển trượt Update Period, bằng cách đặt công cụ Operating trên thanh trượt và kéo của nó tới một vị trí mới.

4. Thực hành điều chỉnh những điều khiển khác.

5. Dừng VI bằng cách kích vào công tắc chuyển đổi thu nhận. VI không thể dừng ngay lập tức bởi vì VI còn phải đợi cho phương trình hay sự phân tích cuối cùng đặt tới hoàn thành thao tác.

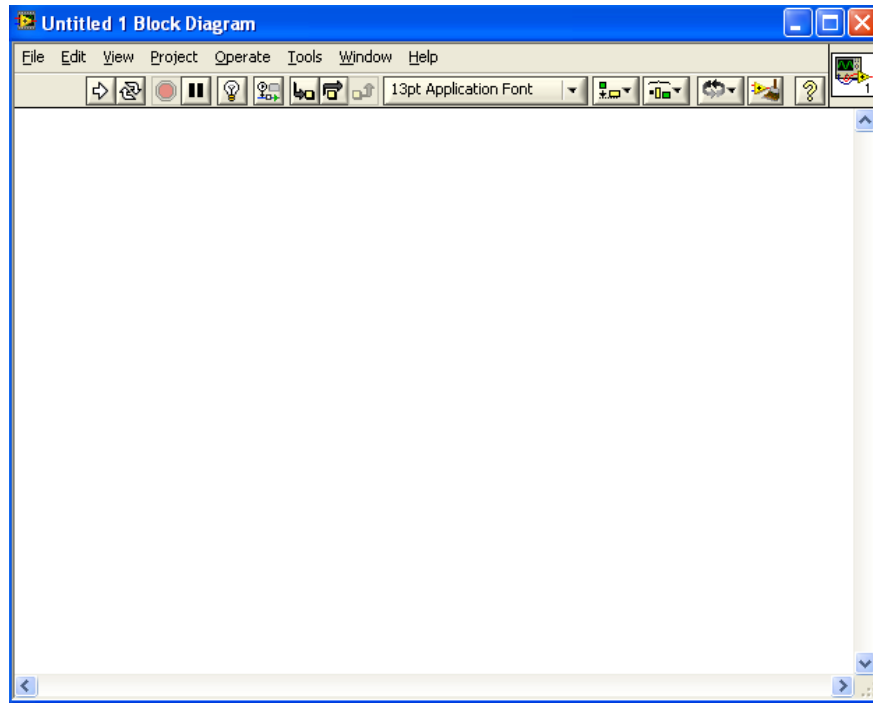
Lưu ý: Ta nên đợi cho một VI thực thi hoàn toàn hoặc nên thiết kế một cách thức để dừng nó, chẳng hạn như đặt một công tắc trên giao diện.

Mặc dù VI dừng nếu ta kích vào nút dừng trên thanh công cụ, đây không phải là cách tốt nhất để dừng các VI lại bởi vì nút dừng dừng chương trình ngay lập tức. Điều này có thể làm gián đoạn các hàm chức năng I/O, và vì thế nó có thể dẫn đến tình trạng không mong muốn.

1.2.2. Sơ đồ khối (The Block Diagram)

Sơ đồ khối chứa đựng mã nguồn đồ thị, thường biết như là mã G hoặc mã sơ đồ khối, cho đến VI chạy như thế nào. Mã sơ đồ khối sử dụng đồ thị biểu diễn các chức năng để điều khiển các đối tượng trên giao diện. Các đối tượng trên giao diện xuất hiện như biểu tượng các thiết bị trên sơ đồ khối. Kết nối điều khiển và các đầu của dụng cụ chỉ thị tới Express VIs, VIs, và các chức năng. Dữ liệu chuyển thông qua dây dẫn từ các điều khiển đến các VI và các hàm chức năng, từ các VI và các hàm chức năng đến các VI và các hàm chức năng khác, và từ các VI và các hàm chức năng đến các dụng cụ chỉ thị. Sự di chuyển của dữ liệu thông qua các nút trên sơ đồ khối xác định mệnh lệnh thực hiện của các VI và các hàm chức năng. Sự di chuyển dữ liệu này được biết như lưu đồ lập trình.

1. Mở sơ đồ khối của một hệ thống nào đó bằng cách chọn **Window>>Show Block Diagram**. Hoặc cũng có thể gọi tới sơ đồ khối bằng cách trên bảng giao diện nhấn <Ctrl E>. Sơ đồ khối có nền màu trắng như hình 1.6 dưới đây:



Hình 1.6 Sơ đồ khối của LabVIEW

2. Sự phân cấp

Sức mạnh của LabVIEW định vị trong bản chất sự phân cấp của các VI. Sau khi bạn tạo ra một VI, bạn sử dụng nó như một VI con trong sơ đồ khối của một VI tầng cao hơn. Bạn có thể có một số vô tận bản chất của các tầng trong sự phân cấp.

3. Các dạng dây nối trên sơ đồ khối

	Vô hướng	Mảng 1 chiều	Mảng 2 chiều
Kiểu Numeric			
Kiểu Boolean			
Kiểu String			
Kiểu Dynamic			

1.3. Những công cụ lập trình LabVIEW

Các công cụ lập trình trên LabVIEW bao gồm các công cụ để tạo ra các thiết bị ảo. Nó bao gồm các công cụ trong bảng giao diện (The Front Panel) và các công cụ trong sơ đồ khối (Block Diagram).

1.3.1. Tools Palette

LabVIEW sử dụng một bảng Tools nổi, bảng mà bạn có thể sử dụng để soạn thảo và gỡ lỗi các VI. Bạn sử dụng phím <Tab> tới bảng thông qua các công cụ sử dụng thông thường trên bảng mẫu. Nếu bạn có đóng Tools palette, chọn **View>>Show Tools Palette** để hiển thị bảng mẫu. Tools palette được minh họa như hình 1.7 dưới đây:



Hình 1.7 Bảng Tool Palette



Automatic Selection Tool: công cụ lựa chọn tự động



Operating tool: đặt những mục bảng mẫu Controls và Function trên bảng giao diện và sơ đồ khối.



Positioning tool: những lựa chọn vị trí, thay đổi kích thước và lựa chọn các đối tượng



Labeling tool: soạn thảo văn bản và tạo ra các nhãn tự do.



Wiring tool: nối dây các đối tượng với nhau trong sơ đồ khối.



Object pop-up menu tool: mang lên trên một thực đơn pop-up cho một đối tượng.



Scroll tool: cuộn xuyên qua cửa sổ không sử dụng thanh công cụ cuộn.



Breakpoint tool: thiết đặt các điểm dừng trên các VI, các hàm chức năng, các vòng lặp, các chuỗi và các trường hợp.



Probe tool: tạo ra các đầu dò trên các dây



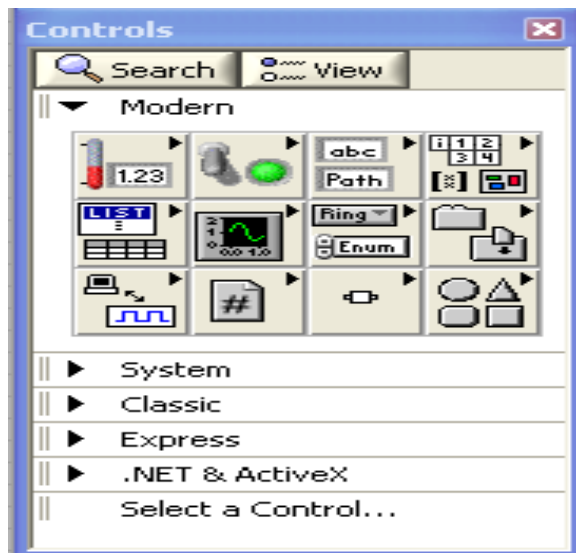
Color copy tool: sao chép các màu để dán tới Color tool



Color tool: thiết đặt các màu nền và màu nổi.

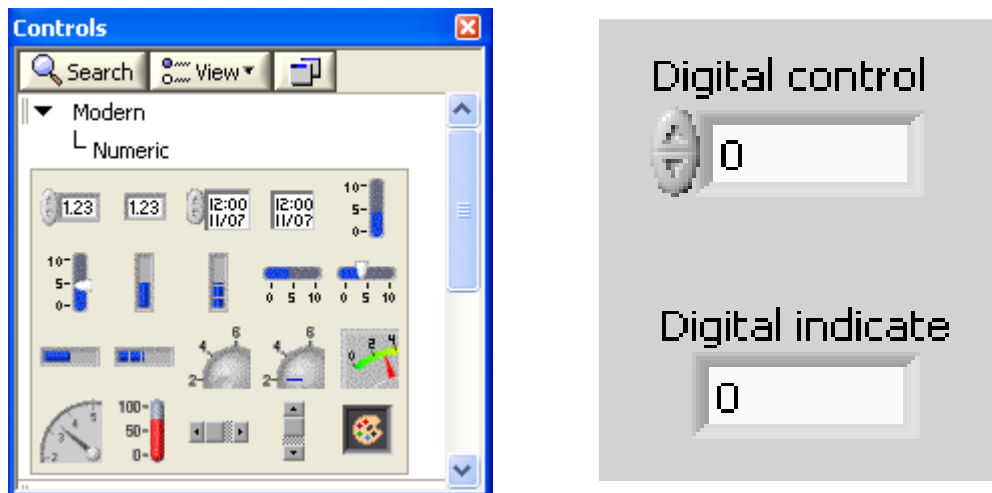
1.3.2. Bảng điều khiển (Controls Palette)

Bảng Controls bao gồm một đồ thị, bảng nổi mà tự động mở ra khi bạn khởi động LabVIEW. Bạn sử dụng bảng này để đặt các điều khiển và các dụng cụ chỉ thị trên bảng giao diện của một VI. Mỗi biểu tượng lớp trên chứa đựng các bảng mẫu con. Nếu bảng Controls không xuất hiện, bạn có thể mở bảng bằng cách lựa chọn **View>>Show Controls Palette** từ menu của bảng giao diện. Bạn cũng có thể bật lên trên một vùng mở trong bảng giao diện để truy nhập một sự sao chép tạm thời của bảng Controls. Sự minh họa sau đây hiển thị lớp đầu tiên của bảng Controls.



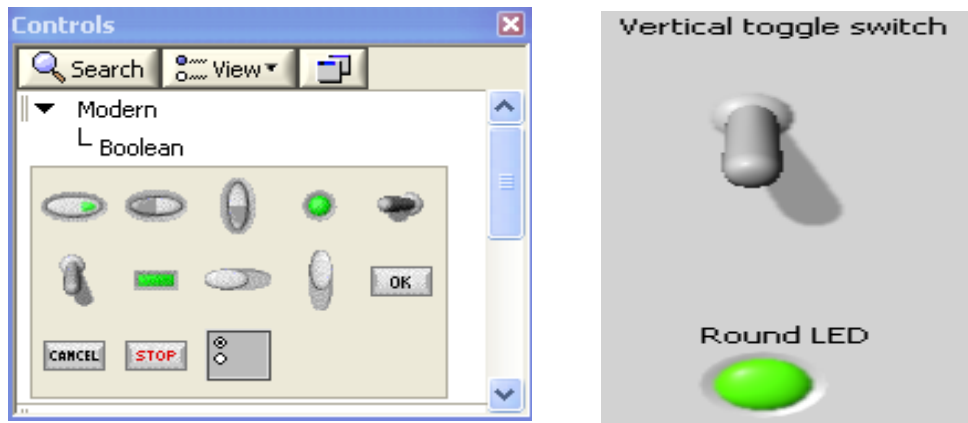
Hình 1.8 Bảng mẫu Controls

1. **Numeric:** Các điều khiển và dụng cụ chỉ thị số (Numeric Controls and Indicator). Ta dùng điều khiển số để nhập các đại lượng số, trong khi đó những dụng cụ chỉ thị số thì hiển thị các đại lượng số. Hai đối tượng số được sử dụng thông dụng nhất đó là *digital control* - điều khiển số và *digital indicator* - chỉ thị số.



Hình 1.9 Bảng điều khiển và chỉ thị số

2. **Boolean:** Các điều khiển và dụng cụ chỉ thị kiểu logic (Boolean Controls and Indicator). Ta sử dụng điều khiển và dụng cụ chỉ thị kiểu logic cho việc nhập và hiển thị các giá trị kiểu Bool (đúng/sai- True/False). Các đối tượng đại số Bool mô phỏng các chuyển mạch - công tắc, các nút bấm, đèn LED. Các đối tượng đại số Bool được sử dụng thông dụng nhất là *vertical toggle switch* - công tắc đảo chiều thẳng đứng và *round LED* - đèn LED xung quanh.



Hình 1.10 Bảng điều khiển và chỉ thị logic

3. Graph: Bao gồm Graph 2D, Graph 3D

Trong đó Graph 2D được chia thành 2 loại :

- Waveform graph : Dùng để biểu diễn những hàm đơn trị có dạng $y = f(x)$, với những khoảng chia ngang nhau trên các trục.
- XY graph : dùng để biểu diễn các hàm đa trị như đường tròn hay dạng sóng thay đổi theo thời gian.



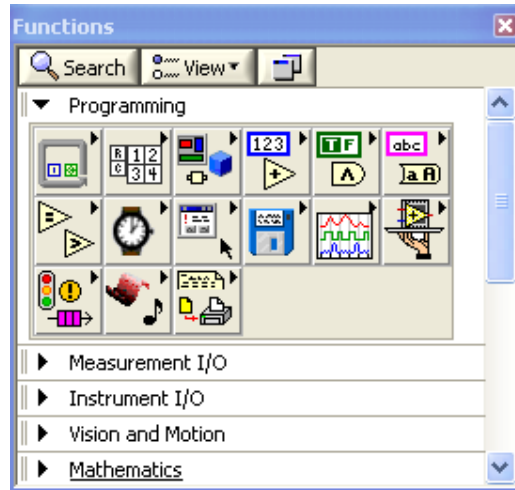
Hình 1.11 Bảng Graph

Ngoài ra, LabVIEW còn rất nhiều thư viện trong bảng mẫu Control như: System, Classic, Express, Control Design & Simulation... Trong đó có hỗ trợ rất nhiều hàm với chức năng khác nhau. Việc sử dụng các hàm trong từng thư viện hết sức linh hoạt và tùy thuộc vào mục đích sử dụng và yêu cầu từng bài toán.

1.3.3. Bảng các hàm chức năng (Function palette)

Bảng Function bao gồm một bảng đồ thị, bảng nổi mà tự động mở ra khi bạn chuyển tới sơ đồ khối. Bạn sử dụng bảng này để đặt các nút (hằng số, dụng cụ chỉ thị, các VI và ...) trên sơ đồ khối một VI. Mỗi biểu tượng lớp trên chứa

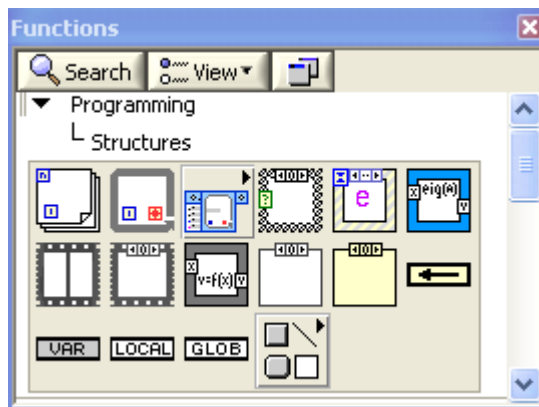
đựng các bảng mẫu con. Nếu bảng Function không xuất hiện rõ ràng, bạn có thể chọn **View>>Show Function Palette** từ menu của sơ đồ khối để hiển thị nó. Bạn cũng có thể mở ra trên một vùng mở trong sơ đồ khối để truy nhập một sự sao chép tạm thời của bảng Functions. Lớp trên của bảng Functions được minh hoạ như hình 1.12 sau đây:



Hình 1.12 Bảng Functions

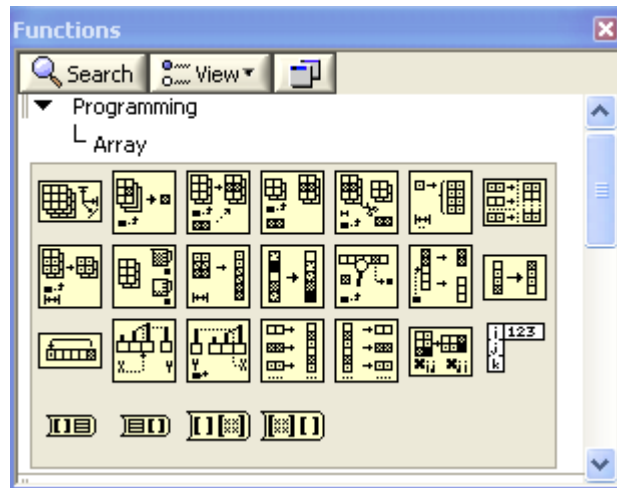
Việc khai thác thể mạnh của LabVIEW trên mỗi lĩnh vực phụ thuộc rất nhiều vào khả năng khai thác thư viện hàm của LabVIEW. Thư viện hàm của LabVIEW được hình tượng hoá trên bảng Function. Người sử dụng dễ dàng truy cập hàm cần dùng bằng cách kích chuột vào biểu tượng trên bảng.

1. Hàm cấu trúc- Structures Function: Bao gồm vòng lặp For, While, cấu trúc Case, Sequence, các biến toàn cục và cục bộ. Đường dẫn truy cập **Function >>Structures**. Biểu tượng của hàm Structures:



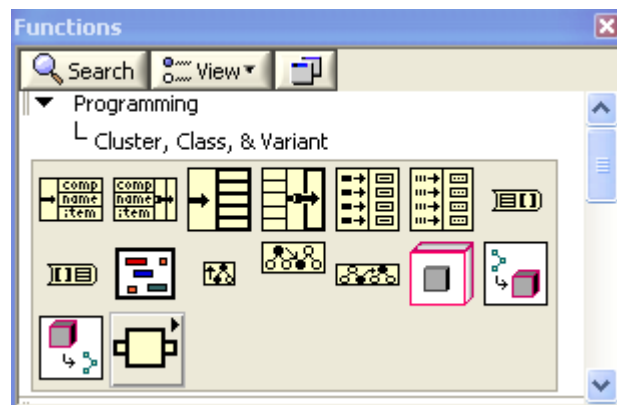
Hình 1.13 Hàm cấu trúc- Structures Function

2. Hàm mảng – Function Array: Sử dụng để tạo ra và điều khiển các mảng. Đường dẫn truy cập: **Function>>Array**. Biểu tượng của hàm:



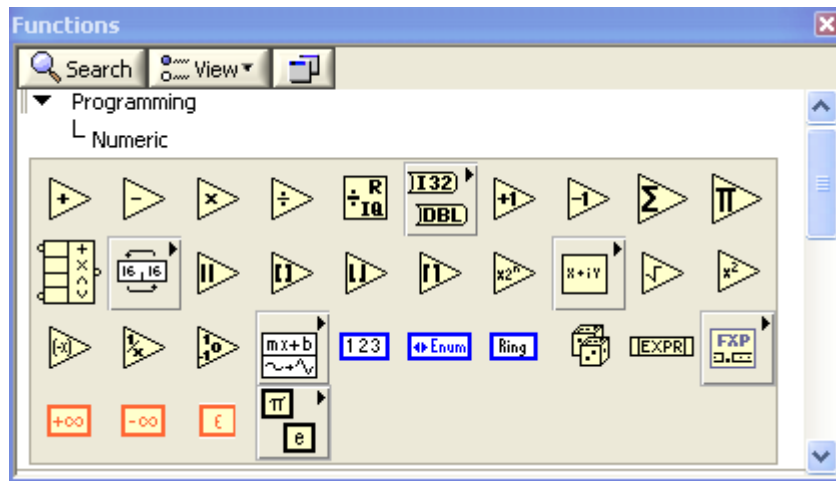
Hình 1.14 Hàm mảng – Function Array

3. Hàm cụm & biến thể – Cluter & Variant: Sử dụng hàm này để tạo ra và điều khiển các cụm, chuyển đổi dữ liệu LabVIEW từ một khuôn dạng bạn có thể thao tác độc lập kiểu dữ liệu, thêm những thuộc tính tới dữ liệu, và chuyển đổi dữ liệu biến thể tới dữ liệu LabVIEW. Đường dẫn truy cập: **Function>>Cluter & Variant**. Biểu tượng của hàm:



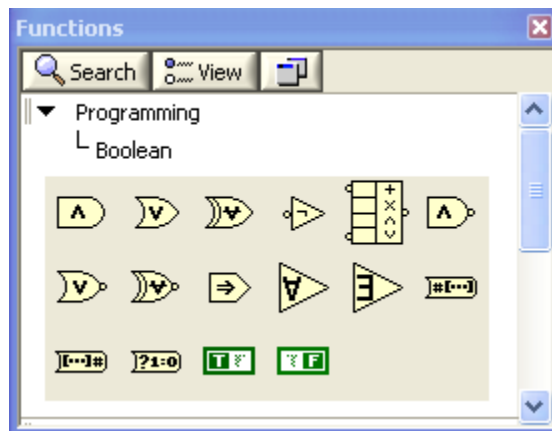
Hình 1.15 Hàm cụm & biến thể – Cluter & Variant

4. Hàm số học – Numeric Function: Sử dụng hàm này để tạo và thực hiện những thao tác số học, lượng giác, Lôgarit, số phức toán học trong các số và chuyển đổi những số từ một kiểu dữ liệu này sang một kiểu dữ liệu khác. Đường dẫn truy cập: **Function>>Numeric**. Biểu tượng của hàm:



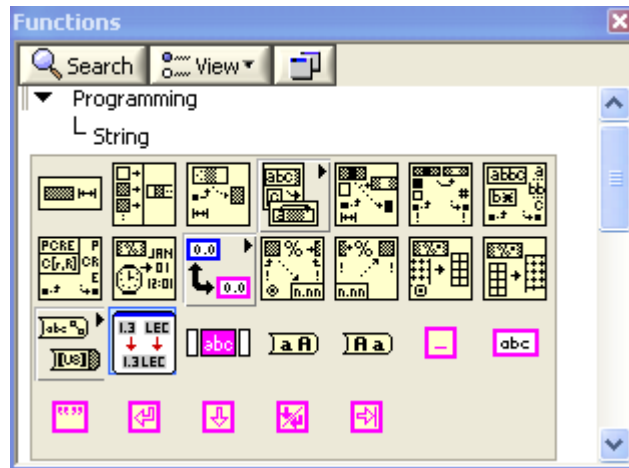
Hình 1.16 Hàm số học – Numeric Function

5. Hàm Boolean- Boolean Function: chứa các hàm logic như: and, or, xor, nor và các hàm logic phức tạp khác. Đường dẫn truy cập: **Function>>Boolean**. Biểu tượng của hàm Boolean:



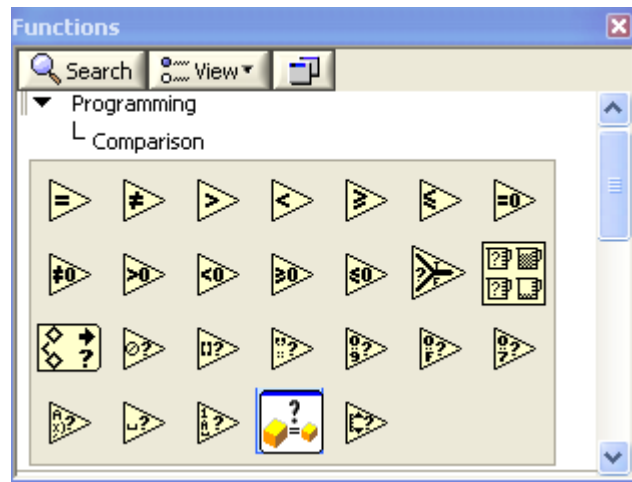
Hình 1.17 Hàm Boolean

6. Hàm chuỗi – String Function: Sử dụng hàm này để liên kết hai hay nhiều chuỗi, tách một tập con của các chuỗi từ một chuỗi, chuyển dữ liệu vào bên trong chuỗi, và định dạng một chuỗi sử dụng trong một công đoạn xử lý từ hoặc ứng dụng bảng biểu. Đường dẫn truy cập: **Function>>String**. Biểu tượng của hàm:



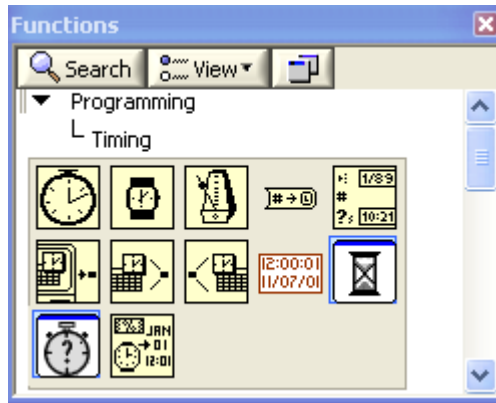
Hình 1.18 Hàm chuỗi – *String Function*

7. Hàm so sánh – Comparison Functions: Sử dụng hàm này để so sánh các giá trị đại số Bool, các chuỗi, các giá trị số, các mảng và các cụm. Hàm so sánh xử lý các giá trị Boolean, string, numeric, array và cluster khác nhau. Bạn có thể thay đổi phương pháp so sánh của vài hàm Comparison. Đường dẫn truy cập: **Function>> Comparison**. Biểu tượng của hàm:



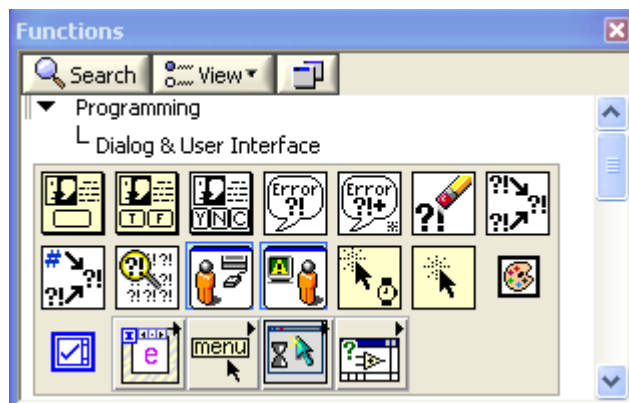
Hình 1.19 Hàm so sánh – *Comparison Functions*

8. Hàm Thời gian – Time function: xác định dòng thời gian, đo khoảng thời gian trôi hoặc trì hoãn một tiến trình trong một khoảng thời gian xác định. Đường dẫn truy cập: **Function>> Timing**. Biểu tượng của hàm Time:



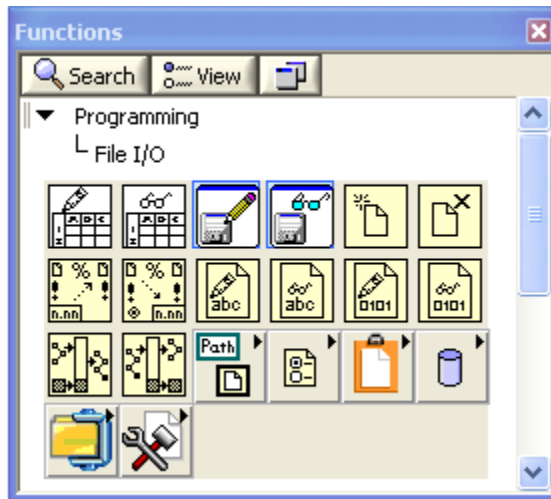
Hình 1.20 Hàm Thời gian – Time function

9. Hàm Dialog & User Interface: Sử dụng hàm này để tạo ra các hộp thoại tới nhắc nhở người sử dụng với các chỉ dẫn. Đường dẫn truy cập: **Function>> Dialog & User Interface**. Biểu tượng của hàm:



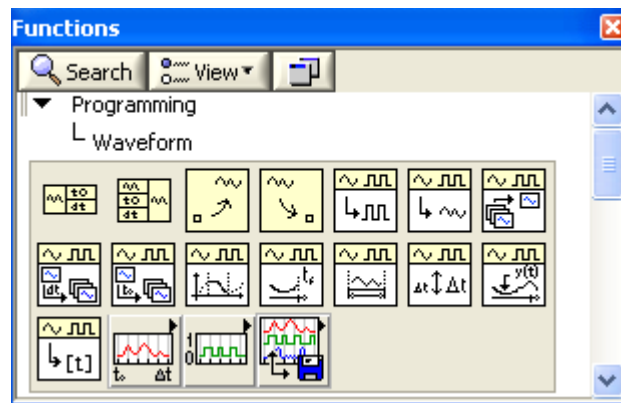
Hình 1.21 Hàm Dialog & User Interface

10. Hàm File I/O- File I/O Function: thực hiện các chức năng cho một tập tin như lưu, mở tập tin theo dạng nhị phân, spreadsheet, đóng một tập tin... Ngoài ra hàm này còn chứa các chức năng mở rộng khác về lưu trữ dữ liệu. Đường dẫn truy cập: **Function >> File I/O**. Biểu tượng của hàm File I/O:



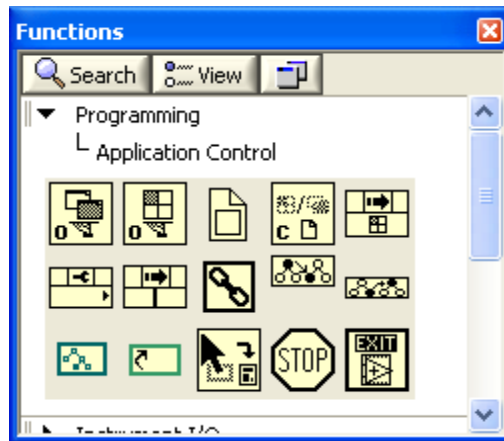
Hình 1.22 Hàm File I/O- File I/O Function

11. Hàm dạng sóng – Waveform: Sử dụng hàm này để xây dựng dạng sóng mà bao gồm các giá trị dạng sóng, thay đổi thông tin, để thiết lập và khôi phục các thành phần và thuộc tính của dạng sóng. Đường dẫn truy cập: **Function>>Waveform**. Biểu tượng của hàm là:



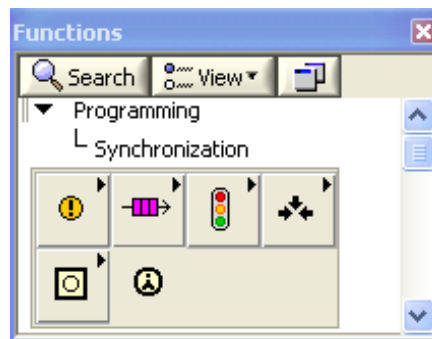
Hình 1.23 Hàm dạng sóng – Waveform

12. Hàm điều khiển ứng dụng- Application Control: Sử dụng hàm này để lập trình các VI điều khiển và các ứng dụng LabVIEW trên máy tính địa phương hoặc qua một mạng. Ta có thể sử dụng các VI và các hàm chức năng này để định dạng nhiều VI tại cùng một thời điểm. Đường dẫn truy cập: **Function>>Application Control**. Biểu tượng của hàm:



Hình 1.24 Hàm điều khiển ứng dụng- Application Control

13. Hàm đồng bộ hoá - Synchronization Function: Sử dụng hàm này để đồng bộ các nhiệm vụ thi hành song song và để chuyển dữ liệu giữa các nhiệm vụ song song. Đường dẫn truy cập: **Function>> Synchronization**. Biểu tượng của hàm là:



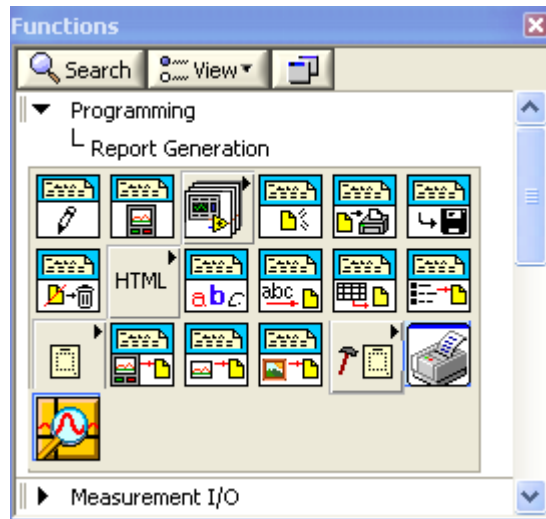
Hình 1.25 Hàm đồng bộ hoá - Synchronization Function

14. Hàm đồ họa và âm thanh – Graphic & Sound Function: Sử dụng hàm này để tạo ra yêu cầu hiển thị, dữ liệu cổng vào và cổng ra từ các phai đồ họa và cho chạy những âm thanh. Đường dẫn truy cập: **Function>>Graphic & Sound**. Biểu tượng của hàm là:



Hình 1.26 Hàm đồ họa và âm thanh – Graphic & Sound Function

15. Hàm phát sinh báo cáo – Report Generation Function: Sử dụng hàm này để tạo và điều khiển các báo cáo của các ứng dụng LabVIEW. Đường dẫn truy cập: **Function>>Report Generation**. Biểu tượng của hàm là:



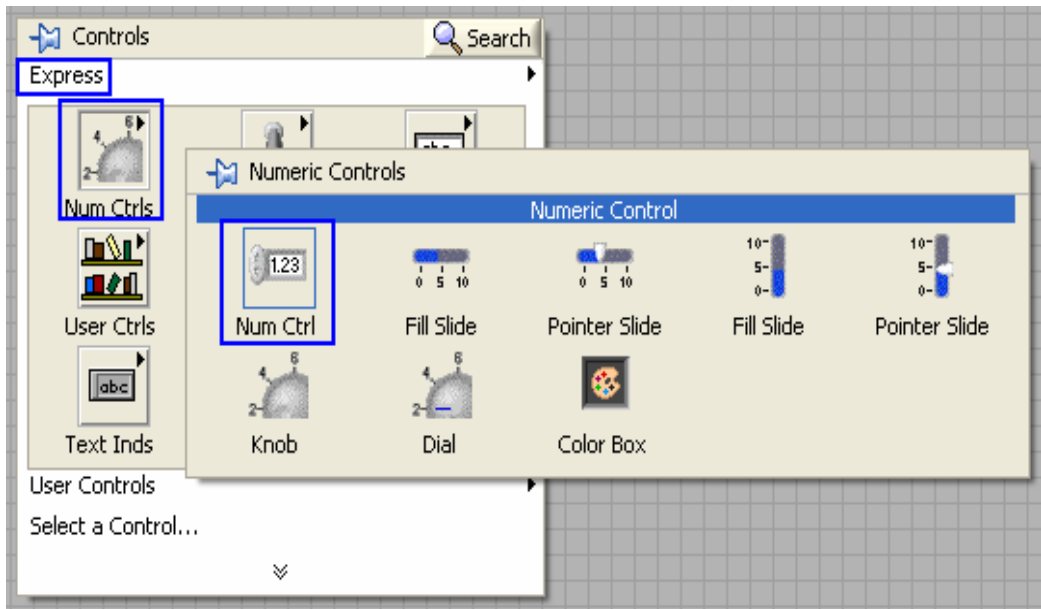
Hình 1.27 Hàm phát sinh báo cáo – Report Generation Function

1.4. Các loại Control và Indicator

1.4.1. Các Control thường dùng

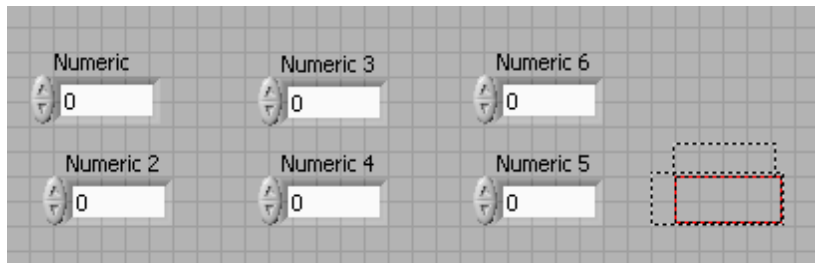
Các control thường dùng gồm các dạng: numeric, fill slide, pointer slide, knob, dial, constant, string, simulated signal, vv.

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 1.28: Cách lấy Control

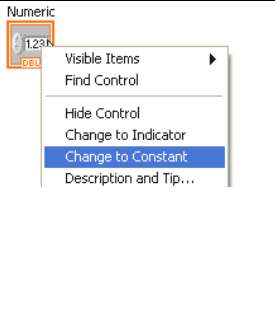
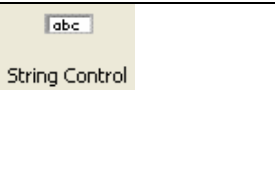
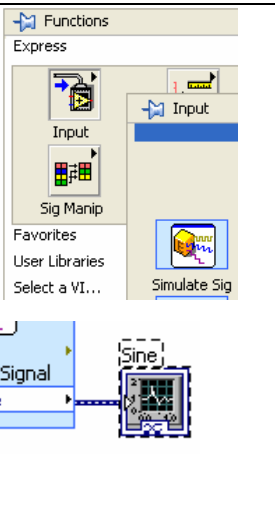
Ta có thể copy nhanh các Control bằng cách nhấn Ctrl (Trên bàn phím) và kéo thả các Control bằng chuột trái.



Hình 129: Copy nhanh bằng việc kéo thả

Các Control thường dùng được tóm tắt tại bảng sau:

 Num Ctrl	Numeric control: là control dạng số
 Fill Slide	Fill slide: Control có dạng một thanh trượt
 Pointer Slide	Pointer slide: Control dạng thanh trượt có nút kéo
 Knob	Knob: Nút vặn
 Dial	Dial: đĩa xoay

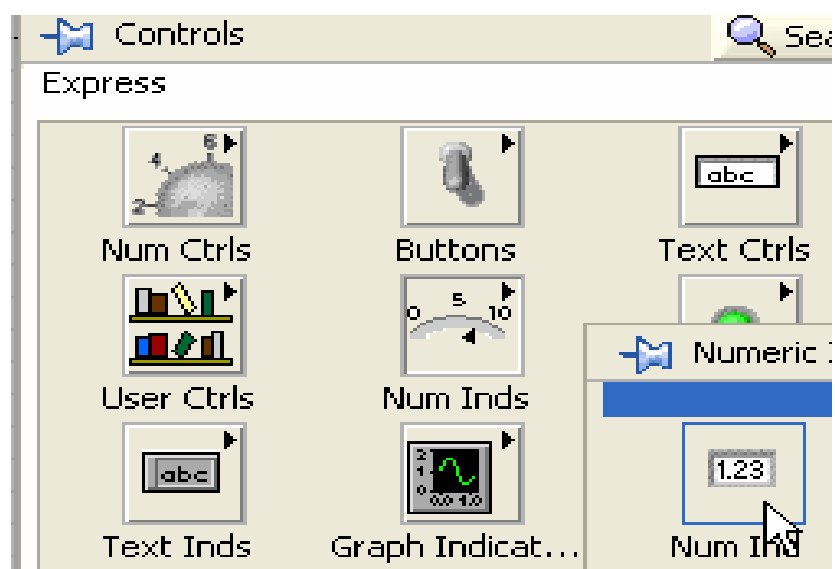
	<p>Constant: hằng số, hay có thể xem là một dạng numeric control nhưng giá trị không thay đổi trong suốt quá trình chạy chương trình. Để tạo Constant ta lấy một Numeric control, chuột phải lên Numeric control chọn Change to constant.</p>
	<p>String control: là 1 dòng chữ hay còn gọi là text control. Sử dụng text Control này để nhập các chữ hoặc chuỗi ký tự, hoặc một câu văn.</p>
	<p>Simulated signal: 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.</p> <p>Nối đầu ra của khối vừa lấy với một Graph bằng cách chọn Right Click lên đầu ra, Create> Indicator.</p>

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

1.4.2. 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.

Để lấy các Indicator, ta vào thư viện theo đường dẫn: FP> Express > Numeric Indicator> như hình 1.30 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.

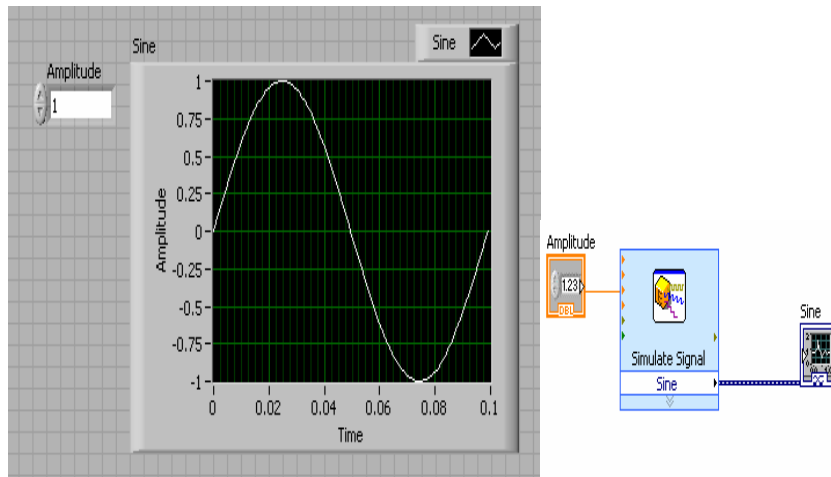


Hình 1.30: Cách lấy Indicator

Bảng tóm tắt các Indicator như sau:

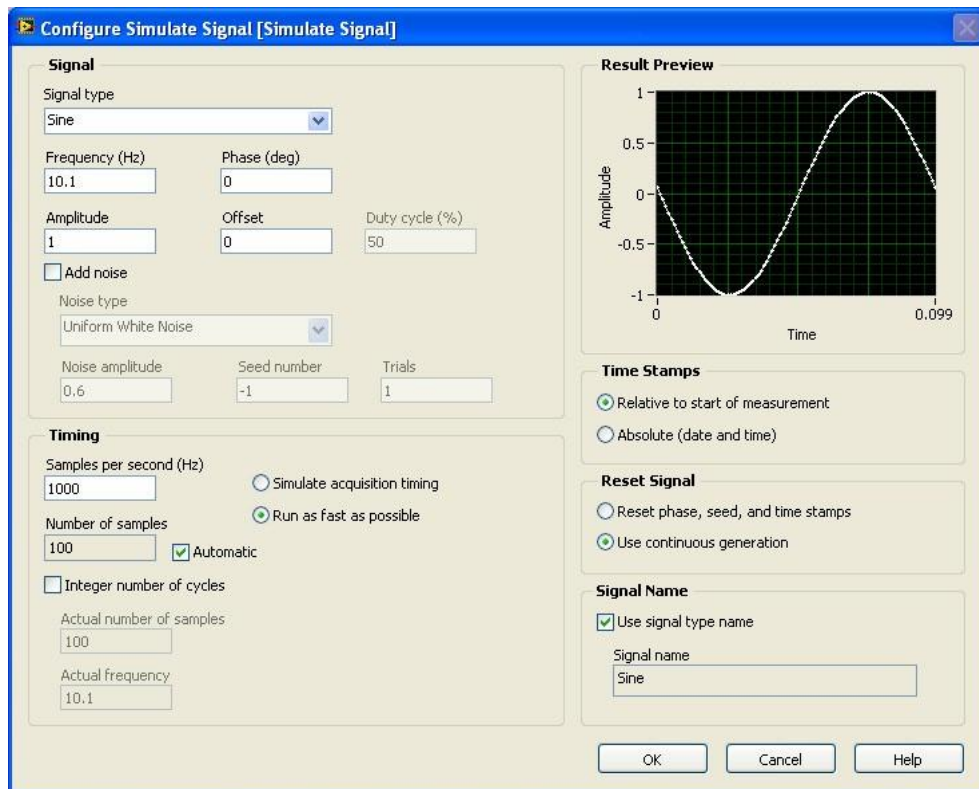
 Num Ind	Numeric indicator: là Indicator dạng số
 Meter	Meter: Indicator có dạng đồng hồ vuông
 Gauge	Gauge: Indicator dạng đồng hồ vuông
 Thermometer	Thermometer: Cột nhiệt độ
 Grad Bar	Graduated Bar: Thanh hiển thị quá trình
 String Indicator	String: là 1 dòng chữ hay còn gọi là text Control, dùng để xuất các chữ hoặc chuỗi ký tự, hoặc một câu văn.
 Chart	Chart: là biểu đồ hiển thị các giá trị theo trục thời gian
 Graph	Graph: là đồ thị thường được dùng để hiển thị các tín hiệu dạng sóng (waveform).
 XY Graph	XY Graph: đồ thị hiển thị quan hệ giữa hai tín hiệu X và Y hoặc dùng trong bài vẽ đồ thị

Ví dụ về sử dụng Chart trình bày như hình 1.31



Hình 1.31: 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 1.32.

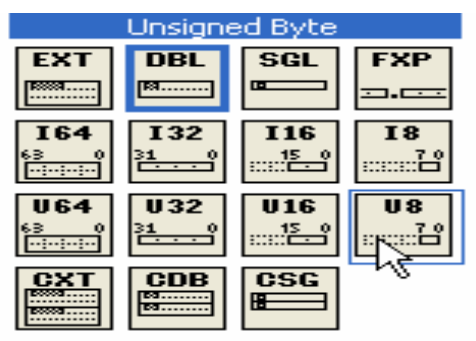


Hình 1.32: Cách lấy Indicator

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.

1.4.3. 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 nằm 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 1.13 liệt kê các kiểu dữ liệu trong LabVIEW.



Hình 1.33: 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
	Byte signed	8	-128 tới 127
	Word signed	16	-32,768 tới
	Long signed integer	32	-2,147,483,648 tới 2,147,483,647
	Quad signed	64	-1e19 tới 1e19
	Byte unsigned	8	0 tới 255
	Word unsigned integer	16	0 tới 65,535
	Long unsigned integer	32	0 tới 4,294,967,295

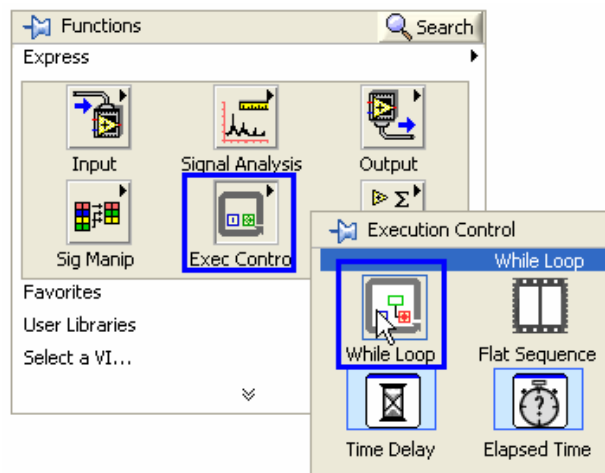
Cách chuyển đổi kiểu dữ liệu: chuột phải lên đối tượng muốn chuyển đổi chọn Representation rồi chọn kiểu dữ liệu mong muốn.

1.5. Vòng lặp While (While Loop), vòng lặp For (For Loop)

1.5.1. Vòng lặp While (While Loop)

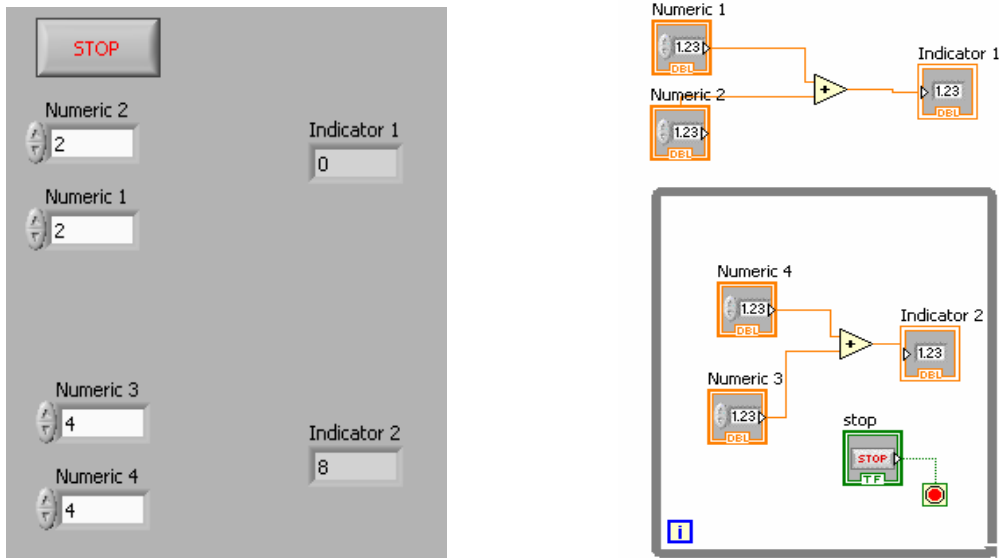
Vòng lặp while là vòng lặp có điều kiện như trong hình 1.14. Ý nghĩa của vòng lặp While là cho phép chạy chương trình mãi tới khi nào nút Stop được nhấn thì mới dừng lại. Để lấy While Loop ta vào BD> Express> Execution> While loop như trình bày trong hình 1.34.

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).



Hình 1.34: Lấy While Loop tại BD

Xét ví dụ trong hình 1.35 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. 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 1.35.

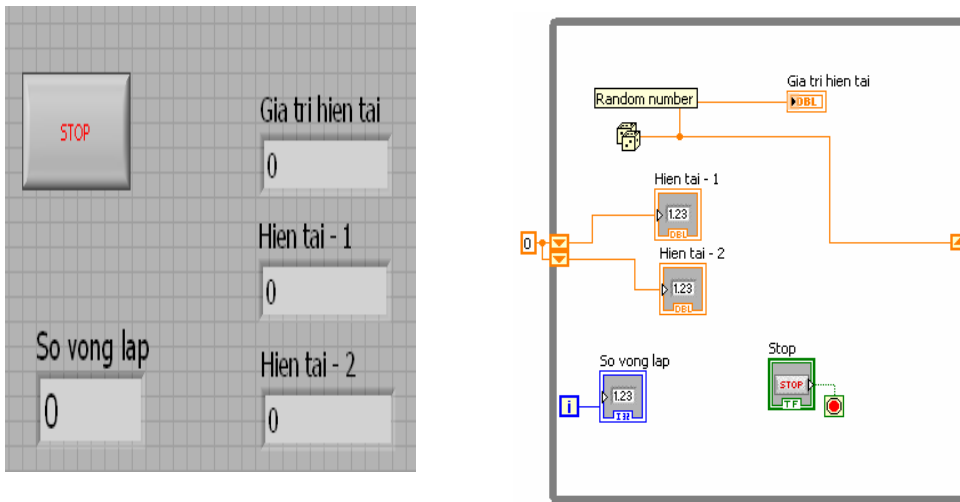


Hình 1.35: Tính tổng với While Loop

Ngoài ra, trong While Loop còn có chân iteration (i) có ký hiệu (i) 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. 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.

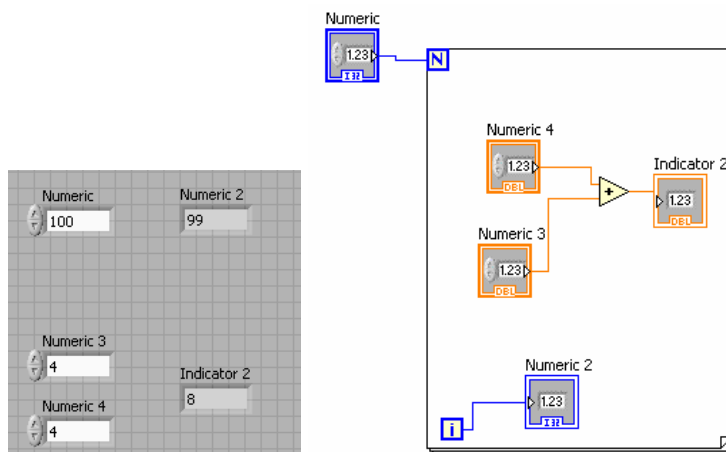


Hình 1.36: Sử dụng shift register

1.5.2. Vòng lặp For (For Loop)

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.

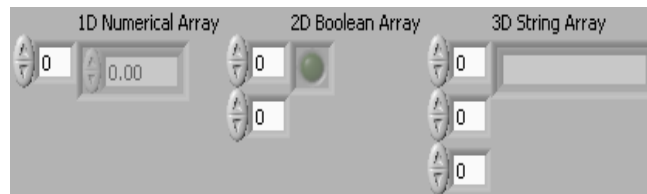


Hình 1.37: For Loop

1.6.Mảng

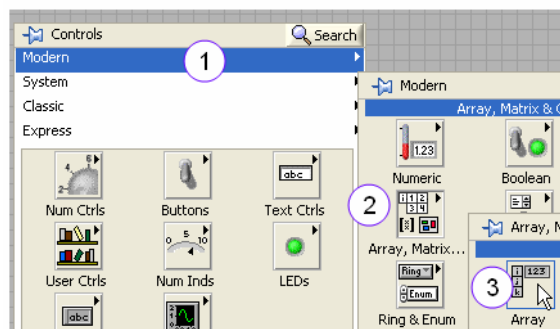
1.6.1.Khái niệm về mảng và cách tạo mảng

Ngoài các loại dữ liệu như đã bàn ở các bài trước, LabVIEW cũng hỗ trợ các cấu trúc dữ liệu kiểu mảng. Như trong ngôn ngữ dựa trên văn bản, Mảng là một khối có nhiều chiều (n-chiều) của một loại dữ liệu nào đó các phần tử trong mảng được đánh chỉ số. Và ta có thể truy suất các phần tử này thông qua các chỉ số đó. Không giống như trong các ngôn ngữ dựa trên văn bản, mảng LabVIEW được tự động thay đổi kích cỡ để phù hợp với dữ liệu của nó.



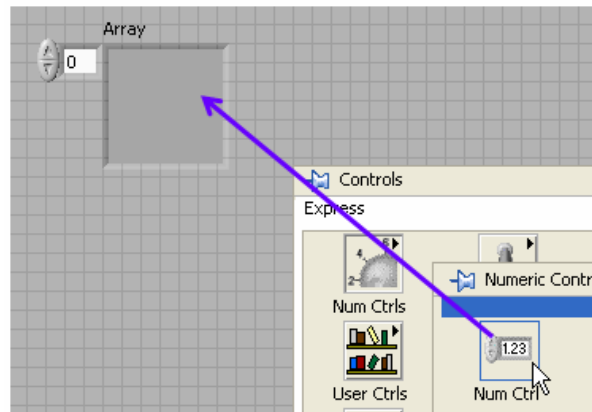
Hình 1.38: Mảng 1 chiều dạng số, 2 chiều dạng boolean, và 3 chiều dạng chuỗi

Cách tạo một mảng như sau: Vào FP> Modern >Array, Matrix> Lấy array ra đặt tại FP




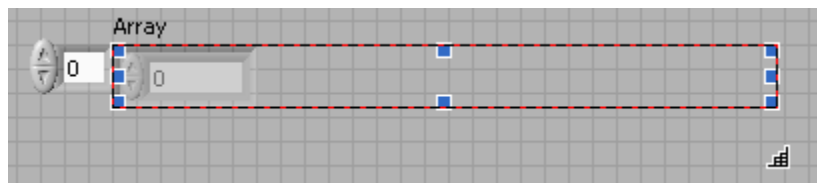
Hình 1.39: Lấy mảng từ FP

Tạo array dạng số bằng cách lấy Numeric control đưa vào array.



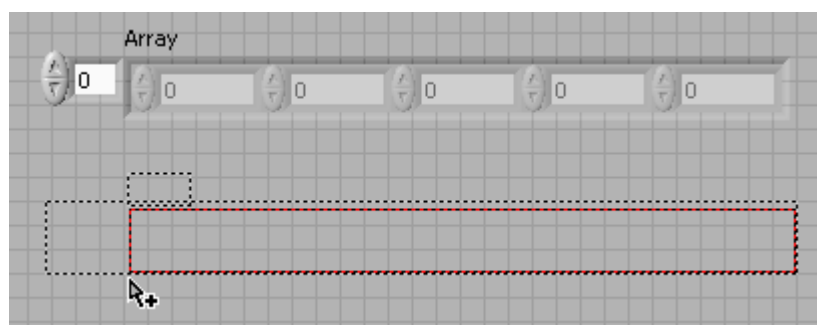
Hình 1.40: Tạo Numeric array

Rê chuột trên mảng cho tới khi có biểu tượng sau: , kéo dài mảng để có một mảng 1 chiều có 5 phần tử.



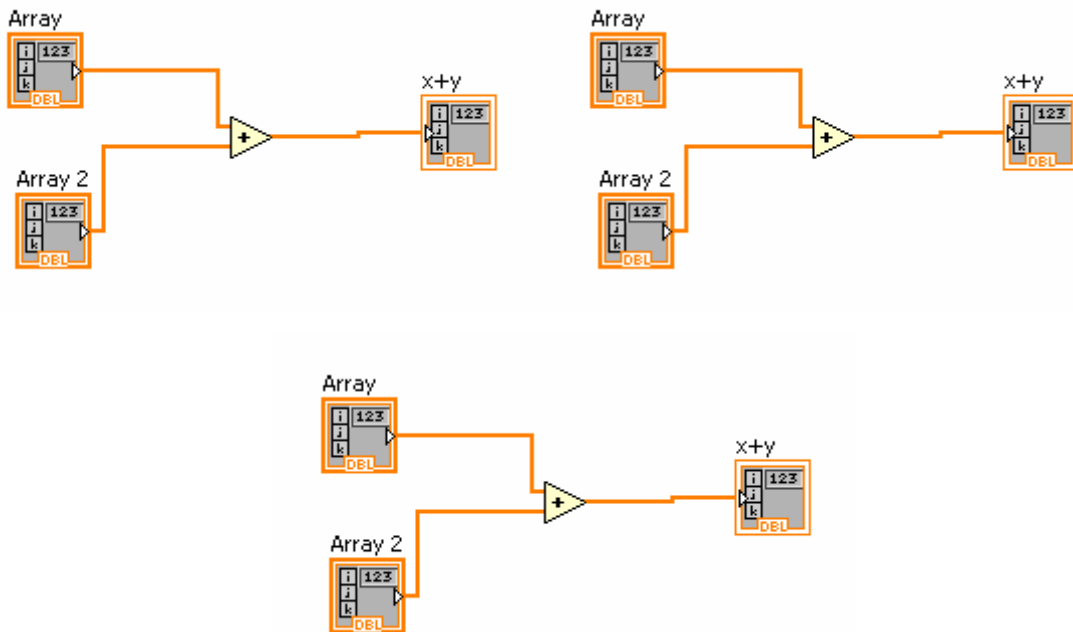
Hình 1.41: Kéo dài mảng

Copy ra một mảng nữa bằng cách nhấn giữ Ctrl+ và nhấn giữa chuột trái lên mảng trên và kéo xuống.



Hình 1.42: Copy nhanh một mảng

Dùng phép cộng nối 2 mảng này lại. Tạo Indicator phía bên phải của hàm cộng, ta được một chương trình tính tổng 2 mảng.

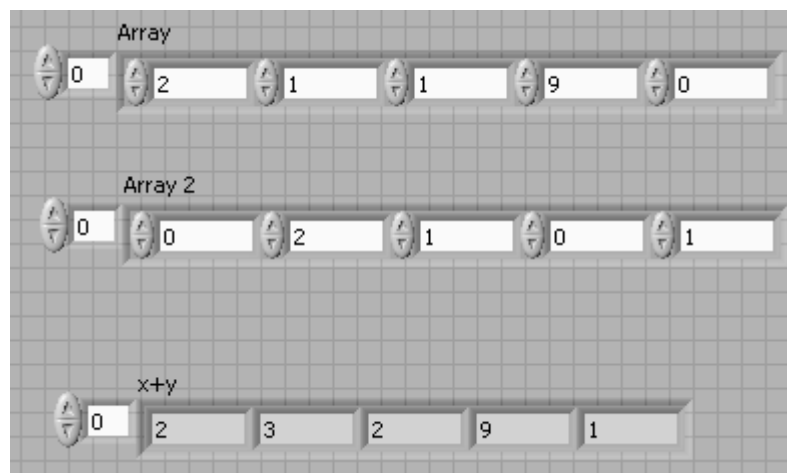


Hình 1.43: Chương trình hoàn thiện

Nhập giá trị vào hai mảng Control. Nhấn chạy chương trình. Ta thấy tổng của hai mảng chính là tổng của các thành phần tương ứng trong mảng.

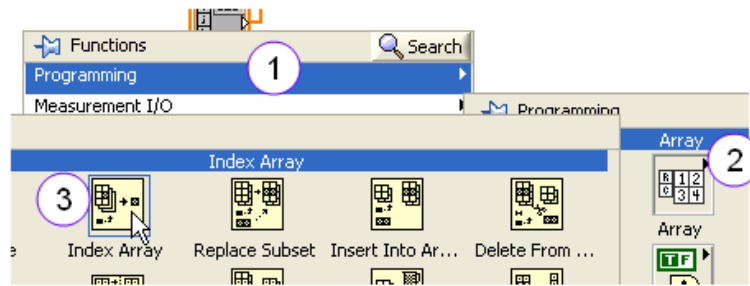
1.6.2.Trích dữ liệu từ một mảng

Ta tiếp tục trích một phần tử từ mảng kết quả để nắm thêm về khái niệm phần tử và chỉ số trong mảng. Để thực hiện VI này ta hãy:



Hình 1.44: Kết quả phép cộng 2 mảng

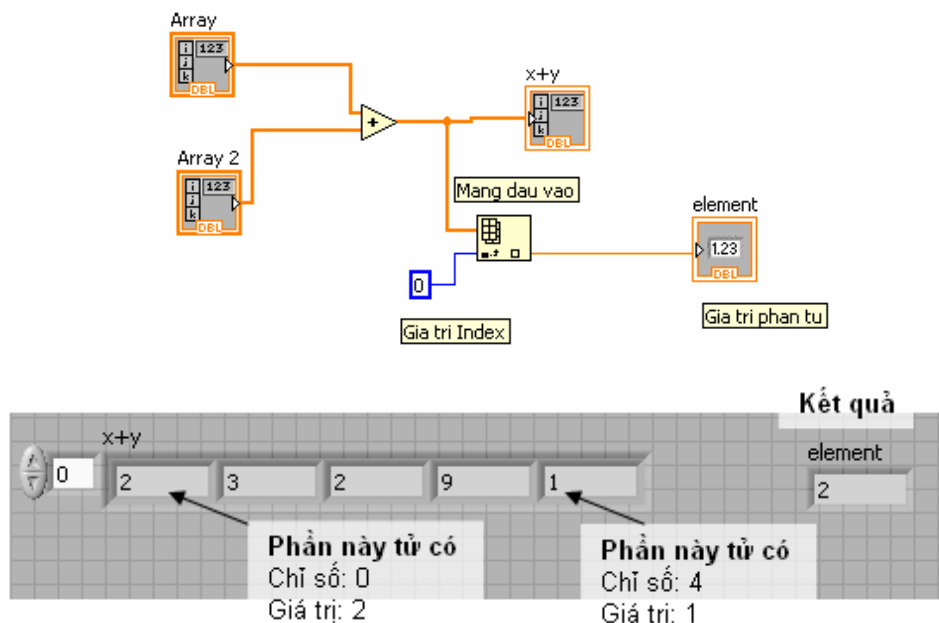
- Vào BD> Programming> Array> Index Array



Hình 1.45: Lấy hàm Index array

- Nối Index array vào array kết quả từ chương trình trên.
- Nhập giá trị 0 vào ô Index và tạo Indicator tại chân element của hàm Index array.

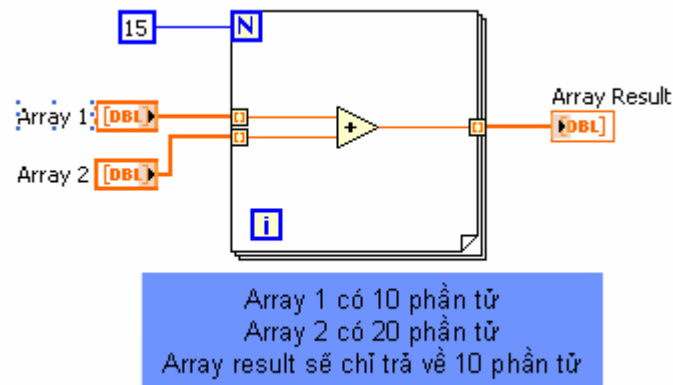
Ta được một chương trình cho phép truy xuất các giá trị của mảng. Ví dụ: Muốn truy xuất giá trị của phần tử có chỉ số 0 thì ta nhập 0 vào ô Giá trị index. Cụ thể ở đây ta có kết quả là 2. Nếu Giá trị index là 1 thì ta có kết quả là 3, vv.



Hình 1.46: Chương trình truy xuất giá trị thành phần trong array

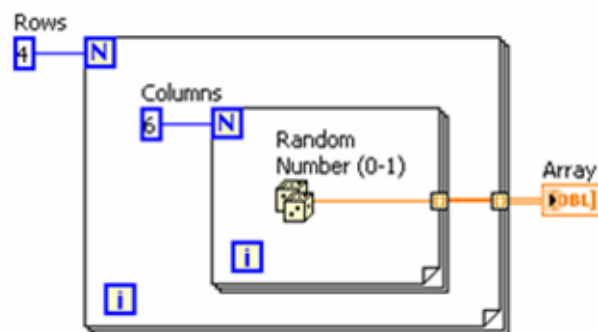
Khi sử dụng array trong For loop. Bạn cần lưu ý một số vấn đề. Cùng nhau xem một ví dụ trên ni.com trong hình 1.46 để giải thích rõ hơn vấn đề này. Chương trình này là một for loop, for loop này lặp lại 15 lần phép cộng các thành phần trong 2 array (array 1 có 10 phần tử, array 2 có 20 phần tử). Kết quả trả về chỉ là một array có 10 phần tử. Lý do: Khi for loop thực hiện tới vòng

lập thứ 11 thì array 1 sẽ không có phần tử thứ 11 nên phép cộng không thể tiếp tục thực hiện.



Hình 1.47: Tổng 2 array

Các Array lớn hơn 1 chiều bạn có thể dễ dàng tạo ra bằng cách ghép các For loop lại với nhau theo hướng [5]. Chú ý rằng bạn cần chọn chế độ Auto-indexing trên thành các For loop trong các trường hợp này.



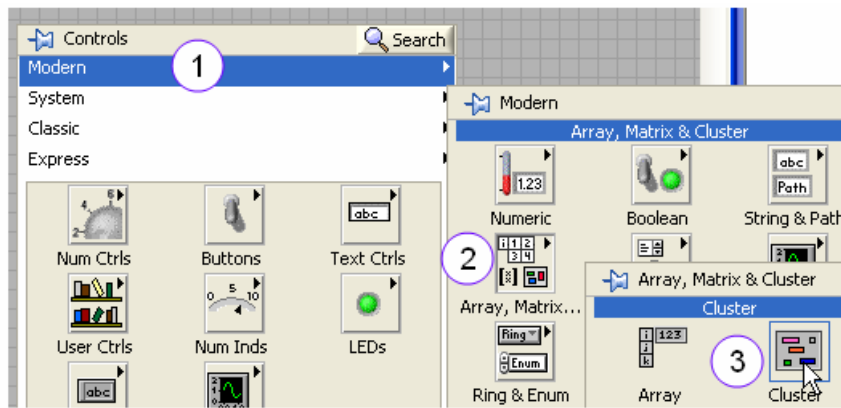
Hình 1.48: Tạo array 2 chiều

1.7.Bó

Bó là kiểu dữ liệu hoạt động như hộp đen có khả năng lưu trữ các loại dữ liệu khác nhau. Có thể hiểu rằng: Bó nược coi là một nhóm các giá trị dữ liệu được đóng gói với nhau để tạo thành một kiểu dữ liệu phức tạp hơn, và thường có ý nghĩa hơn.

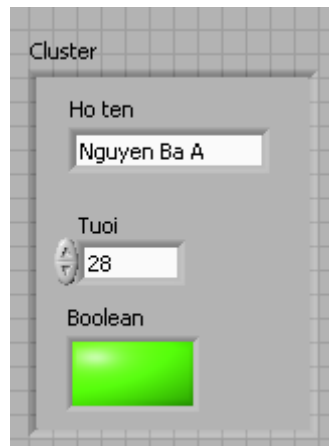
Ví dụ: Hãy tạo một Output cluster (là một Indicator) chứa các thông tin sau: Họ tên, tuổi, giới tính. Thực hiện như sau:

- FP> Mordern> Array, Matrix, > Cluster
- Kéo thả cluster ra ngoài FP



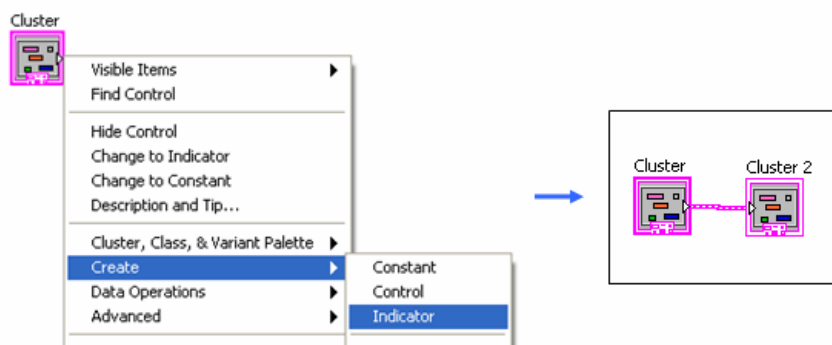
Hình 1.49: Kéo thả Cluster ra Front panel

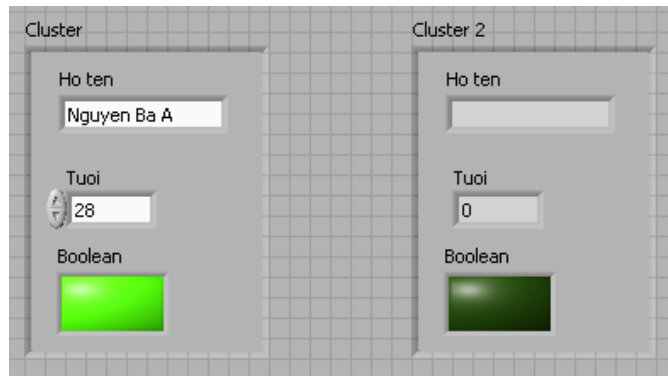
- Đưa vào Cluster này 1 Ctring control để nhập họ tên
- Đưa vào thêm 1 Numeric control để nhập tuổi
- Một LED để chỉ định giới tính (Nam thì LED sáng, Nữ thì LED tắt)
- Nhập các dòng: Họ tên: Nguyễn Bá A, Tuổi: 28, Boolean: Sáng.
- Ta được Cluster control như hình 1.50.



Hình 1.50: Cluster đã tạo ra

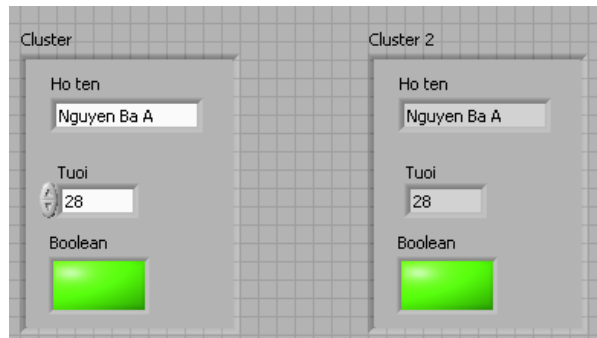
Bây giờ, ta hãy tạo một Cluster indicator như hình 1.51.





Hình 1.51 Tạo Cluster indicator

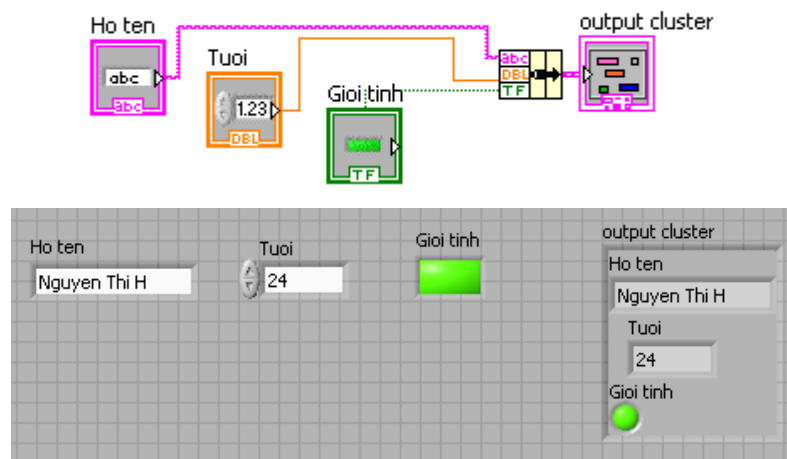
Cho chương trình chạy, ta được kết quả như hình 1.52.



Hình 1.52: Kết quả tạo ra là Cluster indicator sẽ hiển thị đúng Cluster control

Như vậy ta vừa biết phương pháp tạo một cluster. Bây giờ ta sẽ tạo một Cluster như trên bằng cách sử dụng hàm Bundle (tổng hợp) như sau.

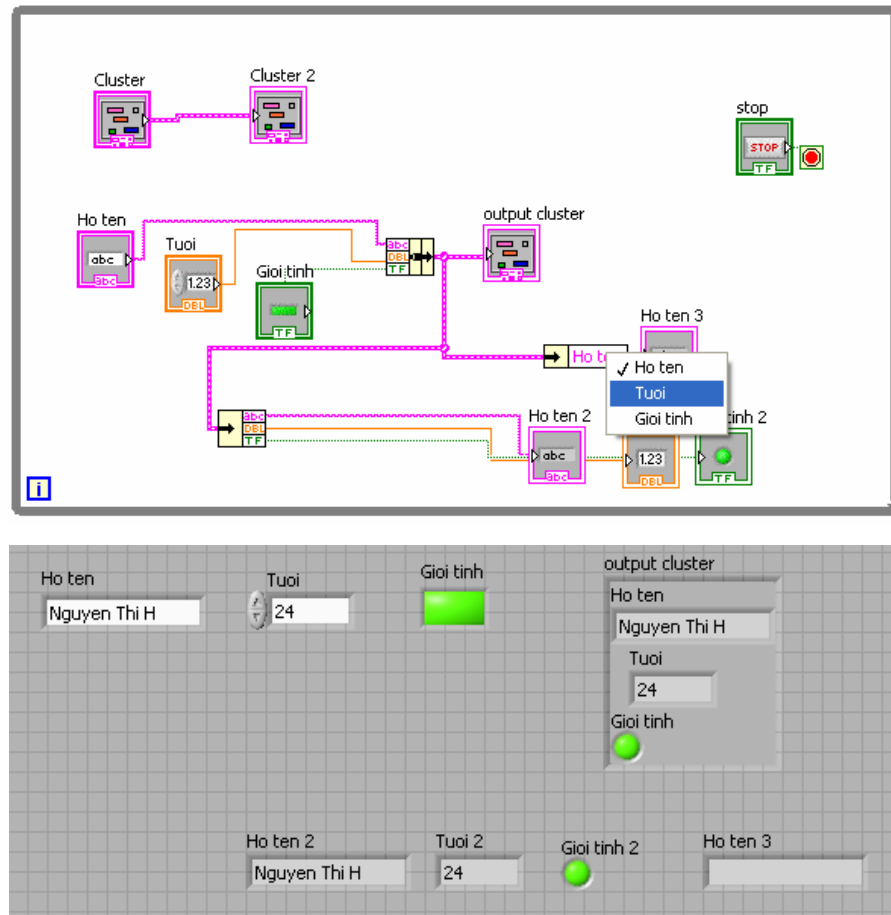
- Lấy các String control, Numeric control và LED đặt tại FP.
- Lấy hàm Bundle tại BD> Programming> Cluster> Bundle. Kết nối lần lượt các control vào Bundle này và tạo Cluster indicator như bài trên.



Hình 1.53: Phương pháp tạo Cluster bằng hàm Bundle

LabVIEW cũng cung cấp hàm Unbundle để ta bóc tách các phần tử trong một bó.

- Lấy một Unbundle nối vào Output cluster. Tạo các indicator từ hàm Unbundle này. Ta sẽ có kết quả là Họ tên, Tuổi và giới tính riêng lẻ. Đây chính là việc bóc tách các phần tử trong một bó.
- Ta có thể dùng hàm Unbundle by name để bóc tách họ tên.



Hình 1.54 Bóc tách các phần tử trong một bó

1.8.Cách tạo thiết bị ảo và thiết bị ảo con.

Một VI có thể phục vụ như một giao diện người dùng hoặc một hoạt động bạn sử dụng thường xuyên. Sau khi bạn học làm thế nào để xây dựng một giao diện và một sơ đồ khối, bạn có thể tạo ra các VI và các VI con và các tùy biến VI mà thuộc quyền sở hữu của bạn.

1. Tìm kiếm từ các ví dụ.

Trước khi bạn xây dựng một VI mới, bạn nên xem xét việc tìm kiếm một ví dụ VI mà đáp ứng các yêu cầu của bạn bằng việc lựa chọn **Help>>Find**

Examples để mở ví dụ tìm ra NI Example Finder. Nếu bạn không tìm thấy một ví dụ VI thích hợp, mở một VI mẫu từ hộp thoại **New** và cư trú bảng mẫu với các VI và các hàm chức năng gắn sẵn tư bảng mẫu **Functions**.

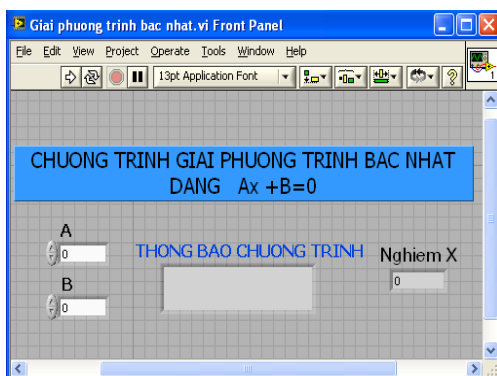
2. Việc sử dụng các hàm chức năng và các VI gắn sẵn.

LabVIEW bao gồm các VI và các hàm chức năng gắn sẵn để trợ giúp bạn xây dựng các ứng dụng đặc trưng, chẳng hạn như các VI và các hàm chức năng thu nhận dữ liệu, các VI mà truy nhập các VI khác, các VI mà giao tiếp với các ứng dụng khác... Bạn có thể sử dụng các VI như các VI con trong một ứng dụng để rút gọn thời gian phát triển. Trước khi bạn xây dựng một VI mới, xem xét việc tìm kiếm bảng **Functions** cho các VI và các hàm thời gian tương tự và việc sử dụng một VI tồn tại như điểm bắt đầu cho một VI mới.

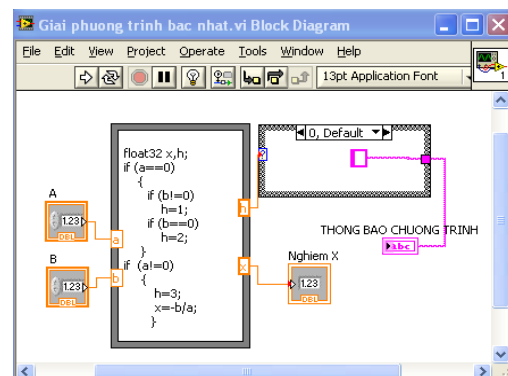
3. Việc tạo ra các VI con

Sau khi bạn xây dựng một VI, bạn có thể sử dụng nó trong một VI khác. Một VI được gọi từ sơ đồ khối của VI khác được gọi là VI con. Bạn có thể sử dụng lại một VI con trong các VI khác. Để tạo ra một VI con, bạn cần xây dựng một bảng nối và tạo ra một biểu tượng. Mỗi VI đều có một icon, nó nằm ở góc trên bên phải của cả 2 cửa sổ Front panel và Block diagram.

Để hiểu rõ cách tạo subVI và icon, connector panel của nó thì ta làm ví dụ sau. Tạo giao diện Front panel của VI và trong Block Diagram ta liên kết giống hình 1.55a và 1.55b bên dưới:



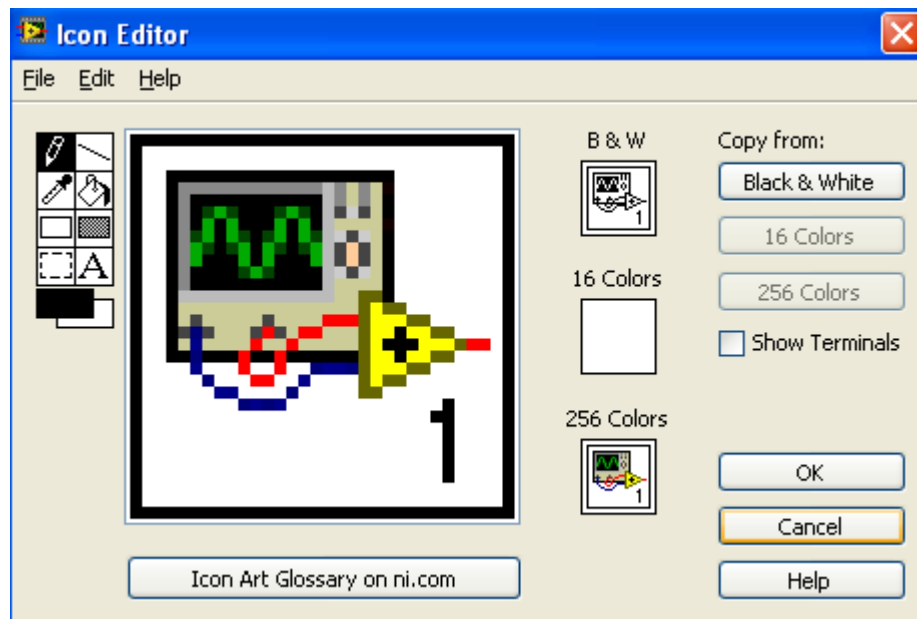
a) *Giao diện*



b) *Sơ đồ khối*

Hình 1.55 Ví dụ minh họa

Để tạo icon thì ta click phải vào biểu tượng ở góc phải bên trên của Front panel hoặc Block diagram. Sau đó bảng Icon Editor xuất hiện như hình dưới, trong đó có các công cụ dùng để vẽ tương tự trong Paint.



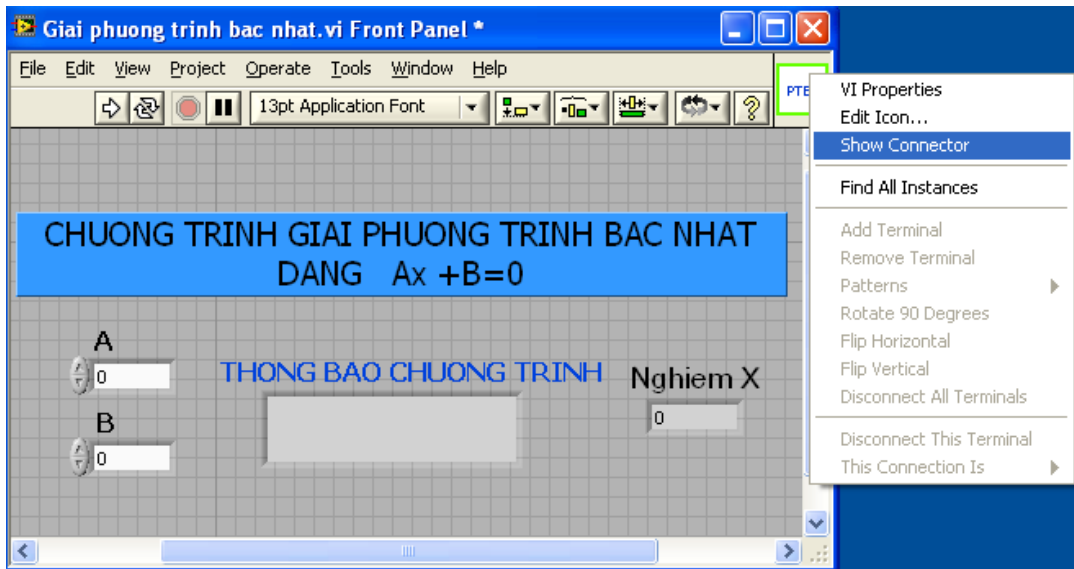
Hình 1.56 Cửa sổ Icon Editor

Giả sử ta vẽ biểu tượng như sau:



Hình 1.57: Ví dụ minh họa vẽ Icon

Bây giờ ta tạo các connector panel. Click phải lên biểu tượng Icon, chọn Show Connector. Như hình 1.58a sau đây:



Hình 1.58a: Các bước vẽ Icon

Sau đó tạo Connector



Hình 1.58b: Các bước tạo Icon

Làm tương tự cho đến Connector cuối cùng như hình 1.58c bên dưới:



Hình 1.58c: Các bước tạo Icon

Sau đó save lại với tên PTB1.vi và tạo một VI mới. Có thể click phải vào block diagram của VI sau đó vào Function palette → Use a VI... Rồi trở tới tập tin PTB1.vi hoặc ta có thể kéo biểu tượng của PTB1.vi đang mở sang block diagram của VI mới.

4. Việc tạo các VI con từ các thành phần của một VI.

Chuyển một thành phần của một VI vào trong một VI con bằng cách sử dụng công cụ Positioning để lựa chọn thành phần của sơ đồ khối bạn muốn sử dụng lại và lựa chọn **Edit>>Create SubVI**. Một biểu tượng cho VI con mới thay thế phần được lựa chọn của sơ đồ khối. LabVIEW tạo ra các điều khiển và các dụng cụ chỉ thị cho VI con mới, tự động định dạng ô vuông đầu nối dựa trên số lượng của các thiết bị điều khiển và dụng cụ chỉ thị bạn đã lựa chọn, và nối dây VI con tới những dây hiện hữu.

Việc tạo một VI con từ một thành phần là tiện lợi nhưng còn đòi hỏi việc quy hoạch cẩn thận để tạo ra một trật tự logic của các VI. Xem xét những đối tượng nào chứa trong thành phần và tránh sự thay đổi chức năng của VI tổng.

5. Việc thiết kế các giao diện VI con.

Đặt các dụng cụ chỉ thị và các điều khiển trên giao diện chúng xuất hiện trong ô vuông đầu nối. Đặt các điều khiển vào bên trái của giao diện và dụng cụ

chỉ thị ở bên phải. Đặt các cụm **error in** vào góc trái dưới của giao diện và các cụm **error out** vào góc phải.

6. Lưu các VI

Chọn **File>>Save** để lưu một VI. Khi bạn lưu một VI, bạn cần phải sử dụng một cái tên để mô tả để bạn có thể nhận ra một cách dễ dàng sau đó. Bạn cũng có thể lưu các VI cho một phiên bản trước đây của LabVIEW làm cho LabVIEW trở nên tiện lợi và để trợ giúp bạn gìn giữ các VI trong 2 phiên bản của LabVIEW khi cần thiết.

7. Việc đặt tên các VI

Khi bạn lưu các VI, sử dụng những cái tên mô tả. Những tên miêu tả, chẳng hạn như **Temperature Monitor.vi** và **Serial Write & Read.vi**, để nhận ra một VI dễ dàng và biết được bạn sử dụng nó như thế nào. Nếu bạn sử dụng những cái tên không rõ ràng, chẳng hạn như VI#1.vi bạn phải tìm nó, khó khăn để nhận ra các VI, đặc biệt nếu bạn đã lưu vài VI với nhau.

Xem xét xem liệu những người dùng sẽ chạy các VI của bạn trên nền khác. Tránh việc sử dụng các ký tự mà một số hệ điều hành dành riêng cho những mục đích đặc biệt, như \ : / ? * < > và #.

Lưu ý: Nếu bạn có vài VI có tên giống nhau đã lưu vào trong máy tính của bạn, tổ chức cẩn thận các VI trong những thư mục khác nhau hoặc các LLB (LabVIEW file that contains a collection of related VIs for a specific use – file LabVIEW mà chứa đựng một tập hợp các VI liên quan cho một sử dụng đặc biệt) để tránh sự tham chiếu LabVIEW lệch hướng VI con khi đang chạy VI lớp trên.

8. Việc lưu giữ một phiên bản trước.

Bạn có thể lưu các VI cho một phiên bản trước của LabVIEW để tiện lợi cho việc nâng cấp LabVIEW và để hỗ trợ bạn gìn giữ các VI trong 2 phiên bản của LabVIEW khi cần thiết. Chọn **File>>Save For Previous Version** để lưu phiên bản trước của LabVIEW.

Khi bạn lưu một VI vào một phiên bản trước, LabVIEW không chuyển đúng VI đó như mọi VI trong trật tự của nó, loại trừ các file trong thư mục labview\vi.lib.

Thường thì một VI sử dụng chức năng không sẵn có trong phiên bản trước của LabVIEW. Trong những trường hợp như vậy, LabVIEW lưu lại như nhiều VI như nó có thể và đưa ra một thông báo là không thể chuyển đổi nó được. Thông báo xuất hiện ngay lập tức trong hộp thoại **Warning**. Kích nút **OK** để ghi nhận các cảnh báo đó và đóng hộp thoại. Kích nút **Save to File** để lưu các cảnh báo tới một file văn bản để quan sát lại sau đó.

9. Tùy biến các VI.

Bạn có thể định dạng các VI và các VI con để làm phù hợp với ứng dụng của bạn cần. Ví dụ, nếu bạn dự định sử dụng một VI như một VI con, yêu cầu người sử dụng đầu vào, định dạng VI vì vậy mà giao diện của nó xuất hiện mỗi lần bạn gọi nó.

Chọn **File>>VI Properties** để định dạng sự xuất hiện và hoạt động của một VI. Sử dụng menu kéo xuống Category tại đỉnh của hộp thoại **VI Properties** để lựa chọn từ vài phạm trù tùy chọn khác nhau.

Hộp thoại **VI Properties** bao gồm các phạm trù tùy chọn sau đây:

- **General:** sử dụng trang này để xác định đường dẫn hiện thời mà một VI đã lưu, số duyệt lại của nó, lịch sử duyệt lại, và bất kỳ sự thay đổi nào làm ra từ khi VI đã được lưu trước đó.
- **Documentation:** sử dụng trang này để thêm một sự mô tả của VI và liên kết tới một chủ đề file trợ giúp.
- **Security:** sử dụng trang này để khoá hoặc đặt mật khẩu một VI.
- **Window Appearance:** sử dụng trang này để tùy biến sự xuất hiện cửa sổ của các VI, chẳng hạn như cửa sổ tiêu đề và kiểu.
- **Window Size:** sử dụng trang này để thiết đặt kích thước của cửa sổ.
- **Execution:** sử dụng trang này để định dạng một VI chạy như thế nào. Ví dụ, bạn có thể định dạng một VI để chạy ngay lập tức khi nó được mở hoặc tạm dừng khi nó được gọi như một VI con.
- **Editor Options:** sử dụng trang này để đặt kích thước của lưới sắp thành hàng cho VI hiện thời và để thay đổi kiểu của điều khiển và dụng cụ chỉ thị LabVIEW tạo ra khi bạn kích phải một thiết bị đầu cuối và chọn **Create>>Control** hoặc **Create>>Indicator** từ menu tắt.

CHƯƠNG 2:

ĐIỀU KHIỂN Lò NHIỆT ĐIỆN TRỞ

2.1. Giới thiệu

Lò điện là một thiết bị điện biến điện năng thành nhiệt năng dùng trong các quá trình công nghệ khác nhau như nung hoặc nấu luyện các vật liệu, các kim loại và các hợp kim khác nhau v.v...

- Lò điện được sử dụng rộng rãi trong các lĩnh vực kỹ thuật :
 - ✓ Sản xuất thép chất lượng cao
 - ✓ Sản xuất các hợp kim phe-rô
 - ✓ Nhiệt luyện và hoá nhiệt luyện
 - ✓ Nung các vật phẩm trước khi cán, rèn dập, kéo sợi
 - ✓ Sản xuất đúc và kim loại bột
- Trong các lĩnh vực công nghiệp khác :
 - ✓ Trong công nghiệp nhẹ và thực phẩm, lò điện được dùng để sấy, mạ vật phẩm và chuẩn bị thực phẩm
 - ✓ Trong các lĩnh vực khác, lò điện được dùng để sản xuất các vật phẩm thủy tinh, gốm sứ, các loại vật liệu chịu lửa v.v...
- Lò điện không những có mặt trong các ngành công nghiệp mà ngày càng được dùng phổ biến trong đời sống sinh hoạt hàng ngày của con người một cách phong phú và đa dạng : Bếp điện, nồi nấu cơm điện, bình đun nước điện, thiết bị nung rắn, sấy điện v.v...

2.2. Ưu nhược điểm của lò điện so với các lò sử dụng nhiên liệu

Lò điện so với các lò sử dụng nhiên liệu có những ưu điểm sau :

- Có khả năng tạo được nhiệt độ cao
- Đảm bảo tốc độ nung lớn và năng suất cao
- Đảm bảo nung đều và chính xác do dễ điều chỉnh chế độ điện và nhiệt độ
- Kín

- Có khả năng cơ khí hoá và tự động hoá quá trình chất dỡ nguyên liệu và vận chuyển vật phẩm
- Đảm bảo điều kiện lao động hợp vệ sinh, điều kiện thao tác tốt, thiết bị gọn nhẹ

Nhược điểm của lò điện:

- Năng lượng điện đắt
- Yêu cầu có trình độ cao khi sử dụng

2.3. Nguyên lý làm việc của lò điện trở

Lò điện trở làm việc dựa trên cơ sở khi có một dòng điện chạy qua một dây dẫn hoặc vật dẫn thì ở đó sẽ toả ra một lượng nhiệt theo định luật Jun-Lenxơ :

$$Q=I^2 RT$$

Trong đó:

- Q** - Lượng nhiệt tính bằng Jun (J)
- I** - Dòng điện tính bằng Ampe (A)
- R** - Điện trở tính bằng Ôm
- T** - Thời gian tính bằng giây (s)

Từ công thức trên ta thấy điện trở **R** có thể đóng vai trò :

- Vật nung : Trường hợp này gọi là nung trực tiếp
- Dây nung : Khi dây nung được nung nóng nó sẽ truyền nhiệt cho vật nung bằng bức xạ, đối lưu, dẫn nhiệt hoặc phức hợp. Trường hợp này gọi là nung gián tiếp.

Trường hợp thứ nhất ít gặp vì nó chỉ dùng để nung những vật có hình dạng đơn giản (tiết diện chữ nhật, vuông và tròn)

Trường hợp thứ hai thường gặp nhiều trong thực tế công nghiệp. Cho nên nói đến lò điện trở không thể không đề cập đến vật liệu để làm dây nung, bộ phận phát nhiệt của lò.

Lò điện trở sử dụng nguồn điện xoay chiều 220V/50Hz, công suất cực đại của lò là 1200W.

Xét về mặt điều khiển, lò điện trở là một khâu quán tính bậc nhất có hàm truyền:

$$W(p) = \frac{K}{T_p + 1}$$

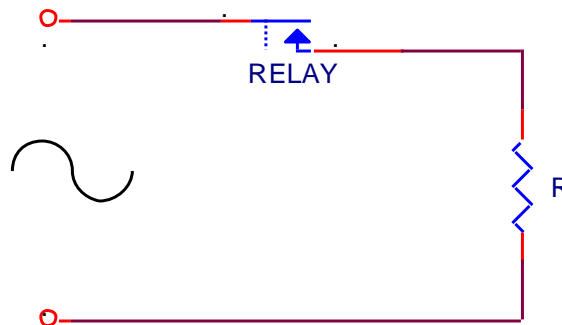
Với các tham số **K**, **T** được xác định một cách gần đúng theo yêu cầu công nghệ.

2.4. Các phương pháp điều khiển lò điện trở

Có rất nhiều phương pháp điều khiển lò điện trở, song phổ biến hiện nay có hai phương pháp là điều khiển dùng Role và điều khiển dùng Thyristor hoặc Triac.

2.4.1. Điều khiển dùng Role

a, Sơ đồ nguyên lý



Hình 2.1 . Sơ đồ điều khiển bằng Role

b, Nguyên lý điều khiển

Nguyên tắc của điều khiển dùng Role là điều khiển hai vị trí có trễ. Đối tượng là lò điện trở đặt trong môi trường nên luôn tỏa nhiệt ra môi trường xung quanh. Xét:

$$\Delta Q = Q_c - Q_t$$

Tương ứng với

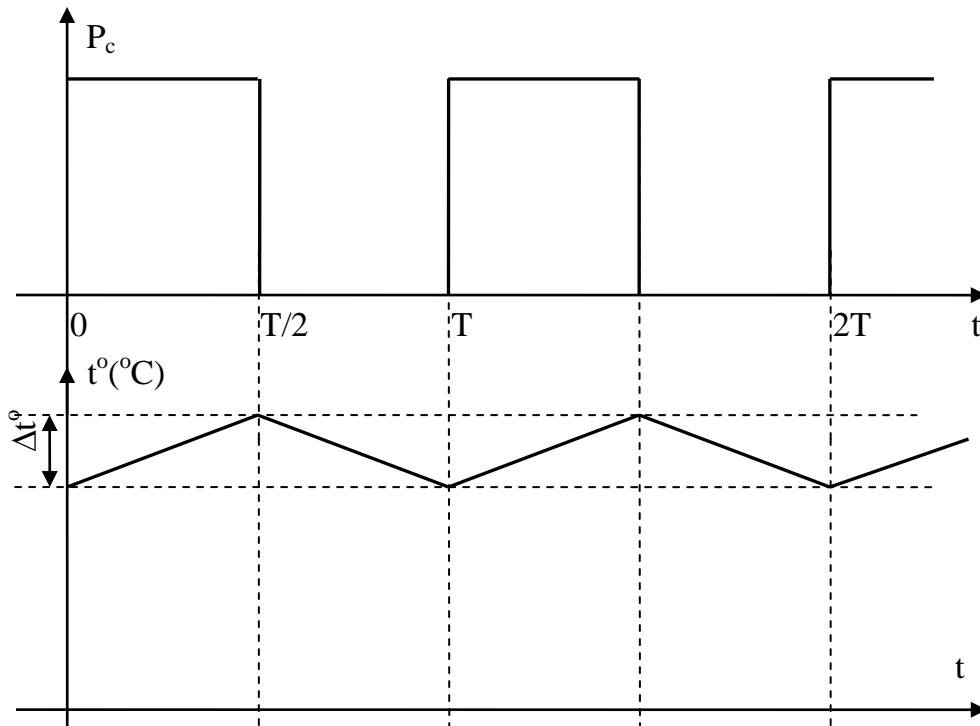
$$\Delta P = P_c - P_t$$

Khi muốn tăng nhiệt độ ta tăng Q_c , tức là tăng công suất cấp P_c cho lò lớn hơn P_t sao cho

$$\Delta Q > 0. (\Delta P > 0)$$

Vậy điều khiển Role chính là điều khiển công suất trung bình của dòng điện cấp cho nguồn.

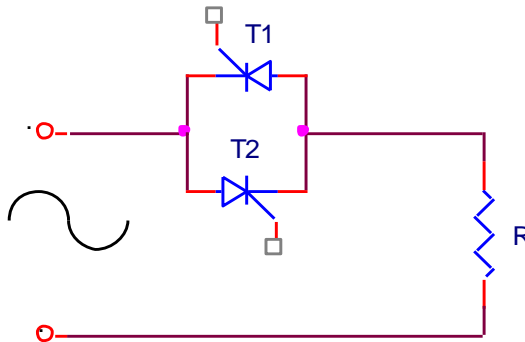
Một cách trực quan ta có đồ thị quan hệ giữa công suất và nhiệt độ theo thời gian sau:



Hình 2.2 . Đồ thị quan hệ giữa nhiệt độ và công suất cấp

2.4.2. Điều khiển Thyristor

a, Sơ đồ nguyên lý



Hình 2.3 . Sơ đồ điều khiển bằng Thyristor

b, Nguyên lý điều khiển

Khác với điều khiển dùng Role, điều khiển dùng Thyristor là điều khiển công suất cấp vào. Cũng từ công thức:

$$\Delta Q = Q_c - Q_t$$

Tương ứng với

$$\Delta P = P_c - P_t$$

Nếu như trong phương pháp điều khiển dùng Role, ΔP hoặc âm hoặc dương tùy theo việc ta ngắt hay đóng Role, thì trong phương pháp này ta điều khiển công suất cấp P_c sao cho $\Delta P = 0$ khi hệ thống ổn định.

Để điều khiển P_c ta điều khiển điện áp cấp vào lò. Muốn vậy ta cần tính toán được góc mở α của Thyristor.

Dựa theo công thức tính giá trị trung bình của điện áp cấp cho tải R khi ta điều chỉnh bằng Thyristor như sau:

$$U_{tb} = \sqrt{\frac{1}{2\pi} \int_0^{2\pi} u(t)^2 dt} = \sqrt{\frac{1}{2\pi} \left(\int_{\alpha}^{\pi} (U_0 \sin(\omega t))^2 dt + \int_{\pi+\alpha}^{2\pi} (U_0 \sin(\omega t))^2 dt \right)}$$

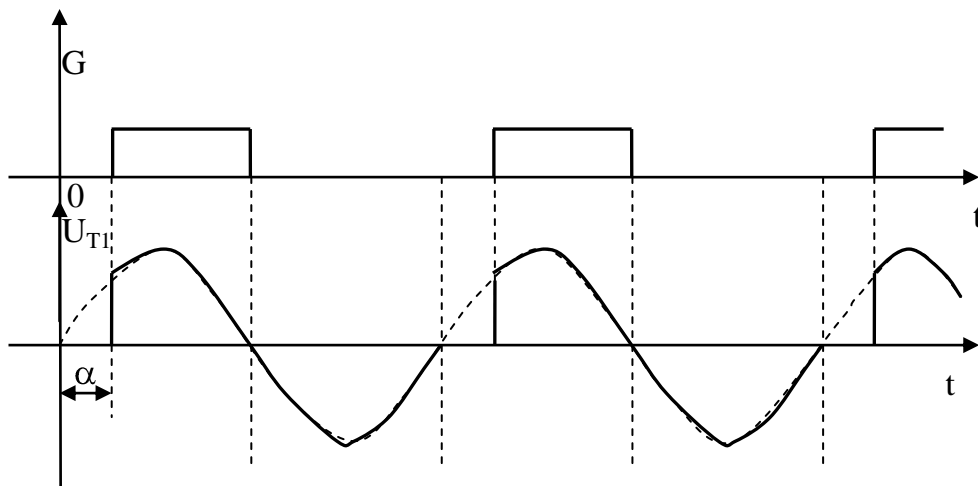
$$U_{tb} = U_0 \sqrt{\frac{2\pi - 2\alpha + \sin 2\alpha}{2}}$$

Với $U_0 = \sqrt{\frac{U^2}{\pi}}$

Vậy có thể nói điều khiển dùng Thyristor chính là điều khiển góc mở α của nó.

$$P_c \sim U_{tb} \sim \alpha.$$

Từ đó ta xác định góc mở van thích hợp để đạt được giá trị điện áp trung bình cấp cho tải. Đồ thị của phương pháp này như sau:



Hình 2.4 . Dạng điện áp ra điều khiển bằng Thyristor

2.4.3. Kết luận

Trong hai phương pháp trên, phương pháp nào cũng có những ưu nhược điểm của nó. Tùy theo yêu cầu của bài toán mà ta lựa chọn phương pháp thích hợp.

Đầu tiên, ta xét phương pháp điều khiển dùng Role

- Ưu điểm:
 - ✓ Cấu tạo phần cứng đơn giản.
 - ✓ Phần mềm tính toán không có gì khó, có thể dùng các mạch điều khiển analog mà không cần lập trình.
 - ✓ Giá cả phải chăng.
- Nhược điểm:
 - ✓ Độ trễ của Role ảnh hưởng lớn đến chất lượng điều khiển. Nếu ta đóng ngắt quá nhanh sẽ dẫn đến tình trạng Role không hoạt động, nếu đóng ngắt quá chậm thì sai lệch nhiệt độ sẽ lớn.
 - ✓ Nhiệt độ luôn luôn không ổn định mà lúc nào cũng dao động trong một giới hạn Δt nhất định.
 - ✓ Đường đặc tính điều khiển không trơn. Do vậy khó cho việc điều khiển các đối tượng có nhiệt độ theo một qui luật nhất định.
 - ✓ Chịu ảnh hưởng của nhiễu như: nhiệt độ buồng đốt, nhiệt độ vật nung, nhiệt độ môi trường... sẽ ảnh hưởng đến nhiệt lượng toả Q_t .

Phương pháp điều khiển dùng Thyristor

- Ưu điểm: Chất lượng điều khiển tốt hơn phương pháp điều khiển dùng Role rất nhiều:
 - ✓ Nhiệt độ lò ổn định.
 - ✓ Có thể điều khiển nhiệt độ của đối tượng theo một đường cong bất kì với chất lượng tương đối tốt.
 - ✓ Có sự cách li về điện.
- Nhược điểm:
 - ✓ Phần mềm tính toán phức tạp hơn phương pháp kia. Do vậy đòi hỏi cấu hình phần cứng tương đối cao.

✓ Giá thành đắt hơn phương pháp điều khiển dùng Role.

Từ các ưu-nhược điểm trên của hai phương pháp ta thấy nếu công việc yêu cầu độ chính xác cao, chất lượng tốt thì ta áp dụng phương pháp hai. Còn nếu chất lượng vừa phải thì có thể dùng phương pháp một để tiết kiệm chi phí.

2.5.Các nguyên tắc điều khiển Thyristor (Triac)

Trong thực tế người ta thường dùng hai nguyên tắc điều khiển: thẳng đứng tuyến tính và thẳng đứng “arcos” để thực hiện điều chỉnh vị trí xung trong nửa chu kỳ dương của điện áp đặt trên Thyristor cũng như Triac.

2.5.1. Nguyên tắc điều khiển thẳng đứng tuyến tính

Theo nguyên tắc này người ta dùng hai điện áp.

- Điện áp đồng bộ, ký hiệu U_s , đồng bộ với điện áp đặt trên hai đầu cực của Thyristor, Triac thường đặt vào đầu đảo của khâu so sánh.

- Điện áp điều khiển, ký hiệu U_{cm} (điện áp 1 chiều có thể điều chỉnh được biên độ) thường đặt vào đầu không đảo của khâu so sánh.

Hiệu điện thế đầu vào của khâu so sánh là:

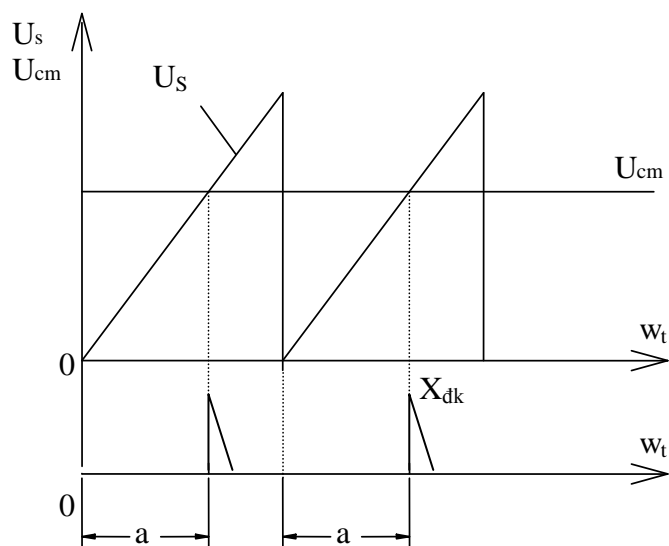
$$U_d = U_{cm} - U_s.$$

Mỗi khi $U_s = U_{cm}$ thì khâu so sánh lật trạng thái, ta nhận được “sườn xuống của điện áp đầu ra của khâu so sánh, sườn xuống này thông qua đa hài một trạng thái ổn định, tạo ra 1 xung điều khiển.

Như vậy, bằng cách làm biến đổi U_{cm} , người ta có thể điều chỉnh được thời điểm xuất hiện xung ra, tức là điều chỉnh được góc α .

Giữa α và U_{cm} có quan hệ như sau:

$$\alpha = \pi \frac{U_{cm}}{S_{sm}} \text{ người ta lấy } U_{cmmax} = U_{sm}.$$



Hình 2.5. Nguyên tắc điều khiển thẳng đứng tuyến tính

2.5.2. Nguyên tắc điều khiển thẳng đứng “arccos”

Theo nguyên tắc này người ta dùng hai điện áp.

- Điện áp đồng bộ U_s , vượt trước $U_{AK} = U_m \sin \omega t$ của Thyristor một góc $\pi/2$ bằng $U_s = U_m \cdot \cos \omega t$.
- Điện áp điều khiển U_{cm} là điện áp một chiều, có thể điều chỉnh được biên độ theo hai chiều (dương và âm).

Nếu đặt U_s vào cổng đảo và U_{CM} vào cổng không đảo của khâu so sánh thì khi $U_s = U_{cm}$, ta sẽ nhận được xung rất mảnh ở đầu ra của so sánh khi khâu này lật trạng thái:

$$U_m \cos \varphi = U_{cm}$$

Do đó: $\alpha = \arccos \left(\frac{U_{cm}}{U_m} \right)$

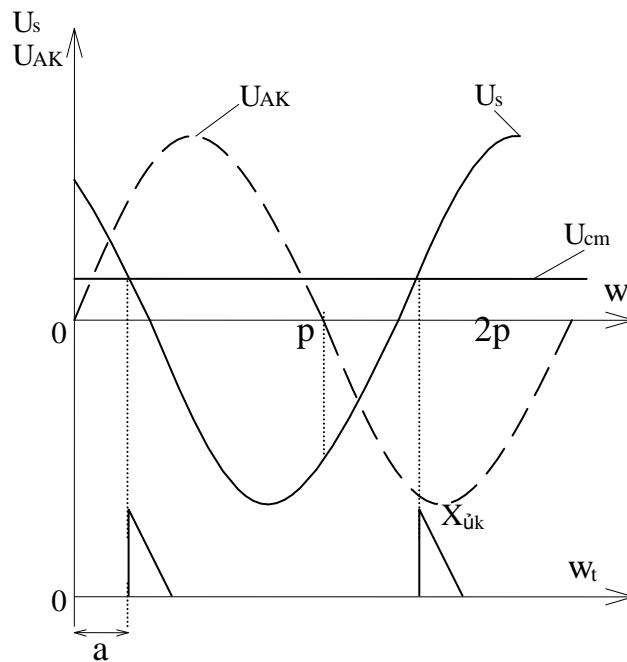
Khi $U_{cm} = U_m$ thì $\alpha = 0$.

Khi $U_{cm} = 0$ thì $\alpha = \frac{\pi}{2}$

Khi $U_{cm} = -U_m$ thì $\alpha = \pi$

Như vậy, khi điều chỉnh U_{cm} từ trị $U_{cm} = +U_m$, đến trị $U_{cm} = -U_m$, ta có thể điều chỉnh được góc α từ 0 đến π .

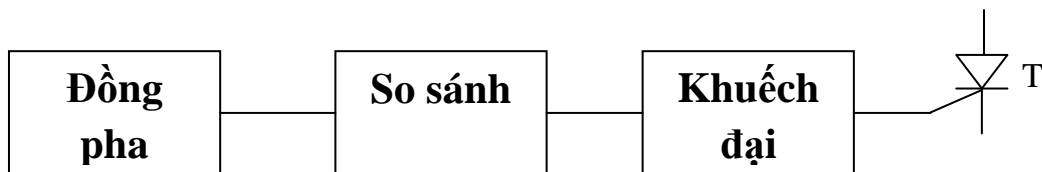
Nguyên tắc điều khiển thẳng đứng “arccos” được sử dụng trong các thiết bị chỉnh lưu đòi hỏi chất lượng cao.



Hình 2.6. Nguyên tắc điều khiển thẳng đứng

2.5.3. Sơ đồ khối mạch điều khiển.

Để thực hiện tốt được việc điều khiển Thyristor, triac thì mạch điều khiển bao gồm các khâu cơ bản sau:



Hình 2.7. Sơ đồ khối mạch điều khiển

Với sơ đồ này nhiệm vụ của các khâu như sau:

- Khâu đồng pha có nhiệm vụ tạo ra điện áp tựa U_{rc} (thường gặp là điện áp dạng răng cưa tuyến tính) trùng pha với điện áp anod của Thyristor.
- Khâu so sánh có nhiệm vụ so sánh giữa điện áp tựa với điện áp điều khiển U_{dk} , tìm thời điểm hai điện áp này bằng nhau ($U_{dk} = U_{rc}$). Tại thời điểm hai điện áp này bằng nhau thì phát xung ở đầu ra để gửi sang tầng khuếch đại.

- Khâu tạo xung có nhiệm vụ tạo ra xung phù hợp để mở Thyristor. Xung để mở Tiristor có yêu cầu: sườn trước dốc thẳng đứng, để bảo đảm yêu cầu Thyristor mở tức thời khi có xung điều khiển (thường gặp loại xung này là xung kim hoặc xung chữ nhật); đủ độ rộng với độ rộng xung lớn hơn thời gian mở của Thyristor, đủ công suất, cách ly giữa mạch mạch điều khiển với mạch động lực (nếu điện áp động lực quá lớn).

CHƯƠNG 3

ĐIỀU KHIỂN Lò NHIỆT ĐIỆN TRỞ GIAO TIẾP VỚI MÁY TÍNH BẰNG LABVIEW

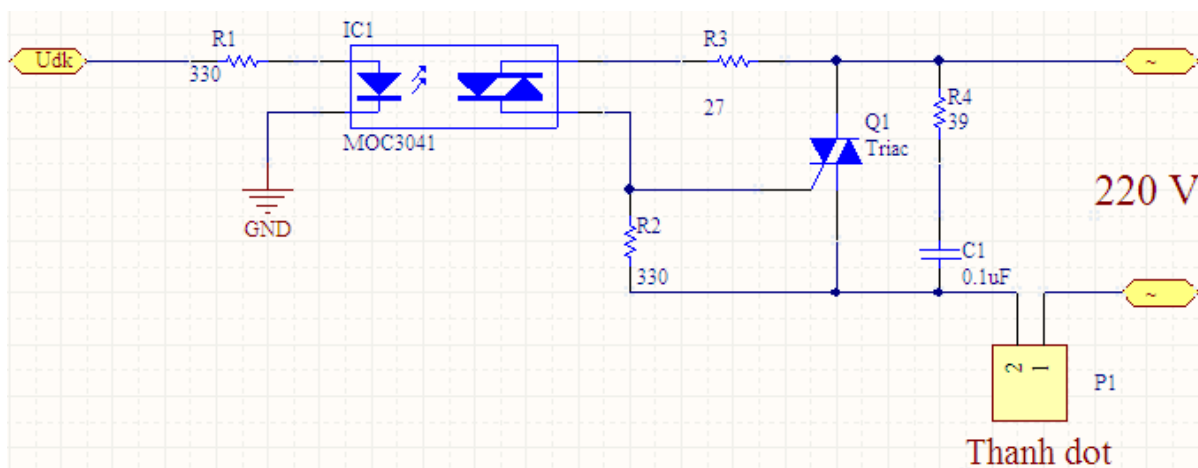
3.1. Phương án thiết kế

3.1.1. Yêu cầu thiết kế

- Mô hình lò nhiệt điện trở sử dụng bóng đèn sợi đốt 200W/ 220V.
- Sử dụng sensor cảm biến nhiệt bán dẫn LM35 để đo nhiệt độ.
- Sử dụng card USB- 9001 để ghép nối giữa máy tính và khối công suất điều khiển lò.
- Áp dụng phần mềm LabVIEW viết chương trình điều khiển và tạo giao diện điều khiển trên máy tính.

3.1.2 Phương pháp điều khiển

- Chọn phương án điều khiển kiểu Role, nhưng thay Role bằng Triac để đóng mở mềm.
- Mạch điều khiển Triac được thiết kế theo sơ đồ sau:



Hình 3.1. Mạch điều khiển lò điện trở

Mạch hoạt động theo nguyên lý sau: (điều khiển theo phương pháp xung)

- Khi nhiệt độ buồng đốt còn nhỏ hơn nhiệt độ đặt (dựa vào sensor LM35), khi đó $U_{dk} = 5V$, MOC dẫn làm Triac mở có điện áp qua Thanh đốt (điện áp lúc này Full 220V).

- Nếu nhiệt độ buồng đốt vượt quá nhiệt độ đặt, khi đó $U_{dk} = 0V$, MOC ngừng dẫn làm Triac khóa không có điện áp qua Thanh đốt, nhiệt độ tạm thời giữ ở đây (vì nhiệt có quán tính chậm).

3.2: Giới thiệu Card USB-9001

3.2.1: Thông số kỹ thuật

Thông số chung	
Cổng kết nối	USB (chuẩn giao tiếp RS232)
Hỗ trợ hệ điều hành	Windows
Kiểu đo	6 kênh đo điện áp (ADC) 1 bộ đếm xung từ các loại encoder (đếm lên hoặc xuống tùy theo chiều quay encoder)
Điều khiển	· 4 kênh xuất tín hiệu số · 2 kênh xuất tín hiệu điều chế xung (PWM)
Họ DAQ	
Đọc tín hiệu Analog	
Số kênh	6 SE
Tốc độ lấy mẫu	142S/s
Độ phân giải	8 bits
Trích mẫu đồng thời	Không
Ngưỡng điện áp giới hạn lớn nhất	0 tới 5 V
Độ chính xác	10 mV ($V_{ref}=2.56V$)
Tín hiệu analog từ các loại cảm biến	Nhiệt độ, áp suất, lưu lượng vv.
Lĩnh vực ứng dụng đo điện áp	Điều khiển tự động, ô tô, công nghiệp
Xuất tín hiệu PWM	
Số kênh	2
Tốc độ cập nhật	100 S/s
Độ phân giải	8 bits
Ngưỡng điện áp	0..5 V

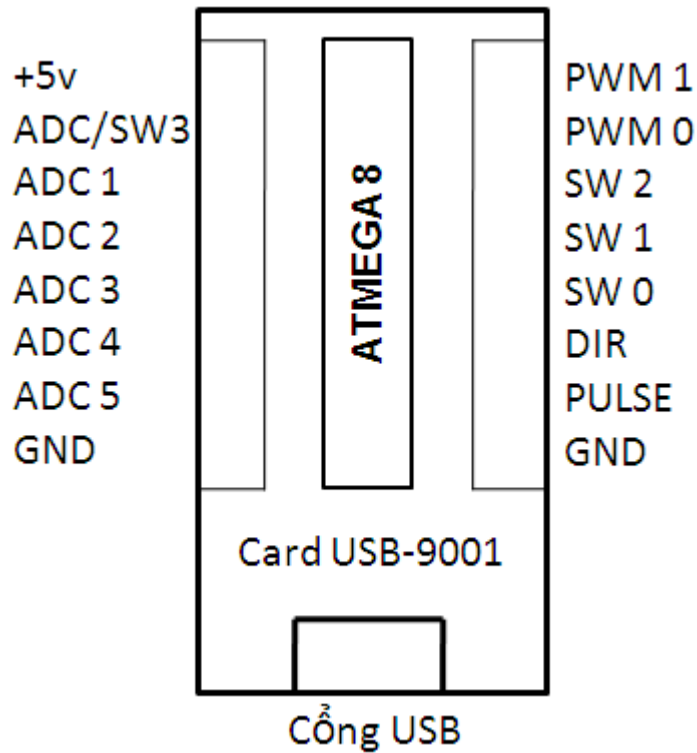
Tín hiệu điều khiển dòng điện	10 mA (dòng ngắn mạch)
Các chân xuất tín hiệu số	
Số kênh	4
Timing	Software
Logic Levels	TTL
Ngưỡng điện áp ra	0..5 V
Output Current Flow	Sinking, Sourcing
Dòng điện (Kênh/Tổng)	10 mA/100 mA
Bộ đếm xung	
Số bộ đếm	1 (đếm lên hoặc đếm xuống)
Độ phân giải	16 bits
Tần số nguồn xung lớn nhất	250 KHz
Độ rộng xung vào nhỏ nhất	2 us
Mức logic	TTL
Ngưỡng cực đại	0..5 V
Ứng dụng	Đo tốc độ động cơ từ Encoder, đo xung, vv.
Cho phép thực hiện nhớ tạm	Yes
Tác động (Triggering)	Digital

3.2.2 Cách sử dụng

Chân	Kí Hiệu	Giá trị	Mô tả	Giá trị Reset
Input	ADC0 -ADC5		Nhận tín hiệu dạng tương tự(analog). Vref sẽ là 5v trên USB hoặc 2.55v do set trên máy tính	NA
Input	PULSE		Đếm xung cạnh lên (0-5v)	0
Input	DIR	0 5V	Set bộ đếm xung PULSE đếm xuống Set bộ đếm xung PULSE đếm lên	5v
output	PWM0 PWM1		Tạo xung với tần số cố định và hệ số xung thay đổi từ 0-255 tùy số đặt trên máy tính (xung 0-5v và 2 tổng trở 470Ohm)	0
Output	SW0-SW2 SW3		Tính hiệu ra dạng số(0 hoặc 5v. Tổng trở 470 Ohm) tùy set trên máy tính Tính hiệu ra dạng số (0 hoặc 5v), sẽ không sẽ không sử dụng ADC0	0
Nguồn	GND		Mass	0
Nguồn	+5v		Lấy từ USB	5v

Sơ đồ chân:

- ADC0-ADC5: trả về giá trị chuyển đổi các chân ADC tương ứng (0-255).
- DAC0-DAC1: đặt giá trị ngõ ra chân PWM cho chân DAC tương ứng (0-255).
- SW0-SW3: đặt giá trị cho 3 ngõ ra số (TRUE-FALSE).
- PULSE: trả về giá trị số xung đã đếm từ chân PULSE (giá trị từ 0-65535).



Hình 3.2. Sơ đồ chức năng Card USB-9001

3.3 Mô hình điều khiển sử dụng card USB-9001

3.3.1. Sensor LM35

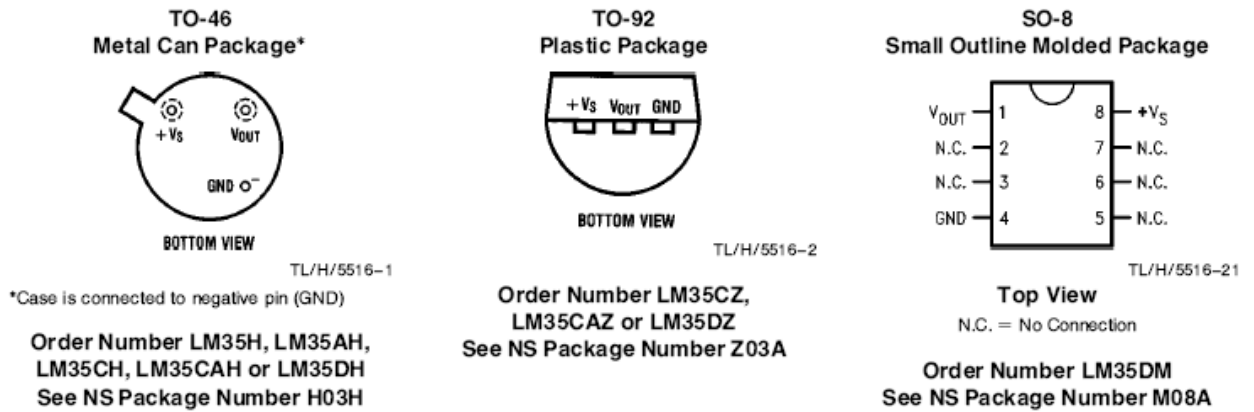
LM35 là họ cảm biến nhiệt, mạch tích hợp, chính xác cao có điện áp đầu ra tỷ lệ tuyến tính với nhiệt độ Celsius. Họ này không yêu cầu căn chỉnh bên ngoài. LM35 cho ra điện áp 10mV ứng với thay đổi nhiệt độ là 1⁰C.

Bảng 3.1 giới thiệu một số thông số kỹ thuật chính của họ LM35:

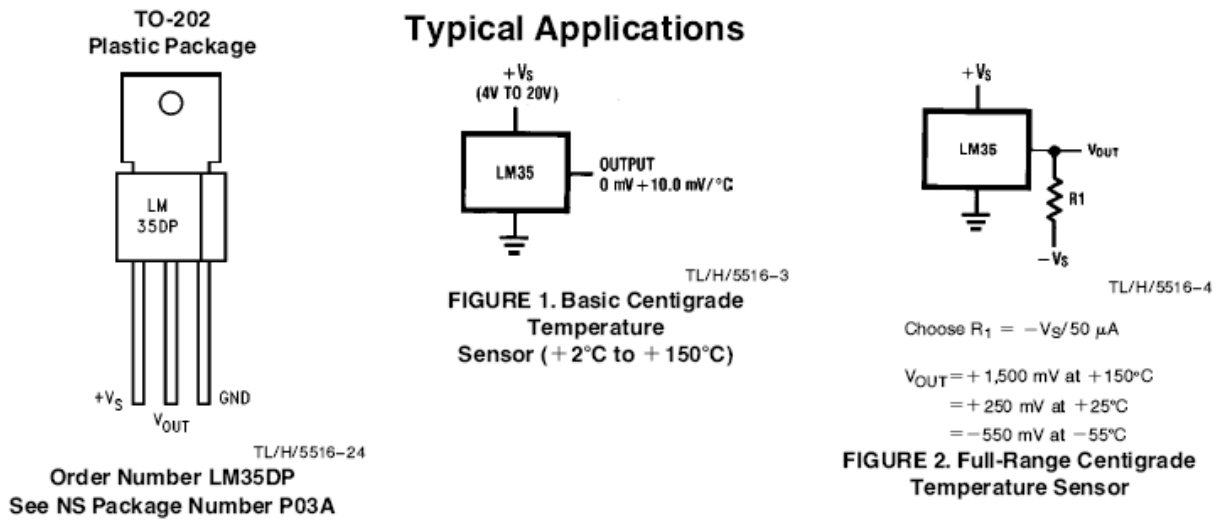
Mã sản phẩm	Dải nhiệt độ	Độ chính xác	Đầu ra
LM35A	-55 C to +150 C	+ 1.0 C	10 mV/F
LM35	-55 C to +150 C	+ 1.5 C	10 mV/F
LM35CA	-40 C to +110 C	+ 1.0 C	10 mV/F
LM35C	-40 C to +110 C	+ 1.5 C	10 mV/F
LM35D	0 C to +100 C	+ 2.0 C	10 mV/F

Kiểu chân và sơ đồ ứng dụng:

Connection Diagrams

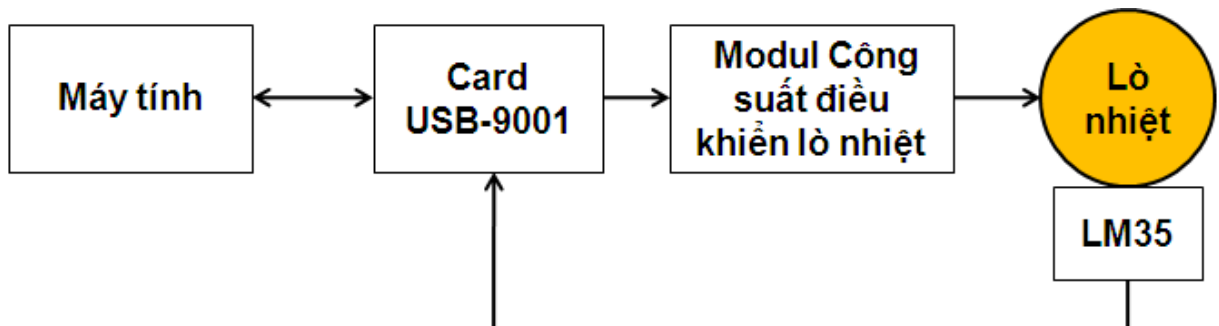


Typical Applications



3.3.2. Xây dựng mô hình điều khiển:

Mô hình điều khiển được xây dựng như hình 3.3.

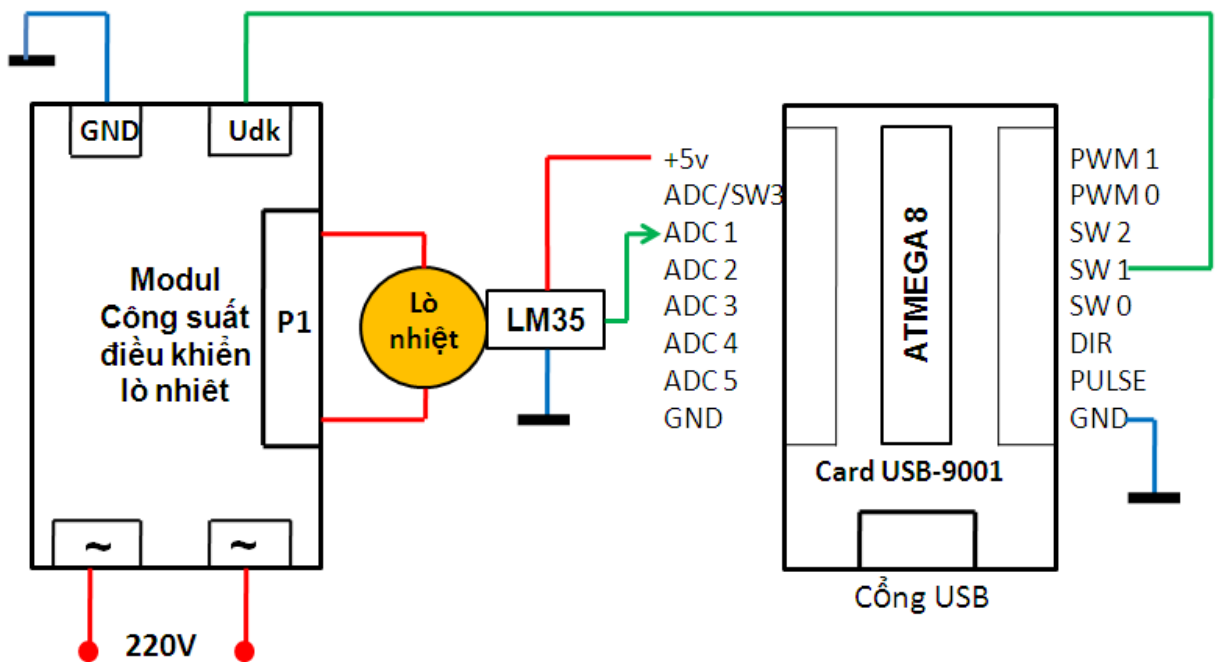


Hình 3.3. Điều khiển lò nhiệt điện trở ghép nối với máy tính

Trong đó:

- Máy tính: tạo giao diện và điều khiển lò nhiệt bằng phần mềm LabVIEW thông qua card USB_9001.

- Card USB-9001: giao tiếp với máy tính qua cổng USB. Nó nhận dữ liệu nhiệt độ từ LM35 qua ADC1 (8 bit) chuyển thành tín hiệu số đưa về máy tính. Bằng chương trình LabVIEW máy tính sẽ xử lý, thông qua card USB-9001 tạo ra Udk, điều khiển Triac (Modul Công suất điều khiển lò nhiệt) đóng mở nguồn cấp cho Lò nhiệt.
- Modul Công suất điều khiển lò nhiệt: trực tiếp đóng mở nguồn cấp điện cho lò nhiệt.
- LM35: sensor đo nhiệt độ lò nhiệt
- Lò nhiệt điện trở: mô hình sử dụng bóng đèn sợi đốt 200W/ 220V.



Hình 3.4. Sơ đồ ghép nối phần cứng điều khiển lò nhiệt điện trở

3.4: Chương trình điều khiển bằng ngôn ngữ LabVIEW

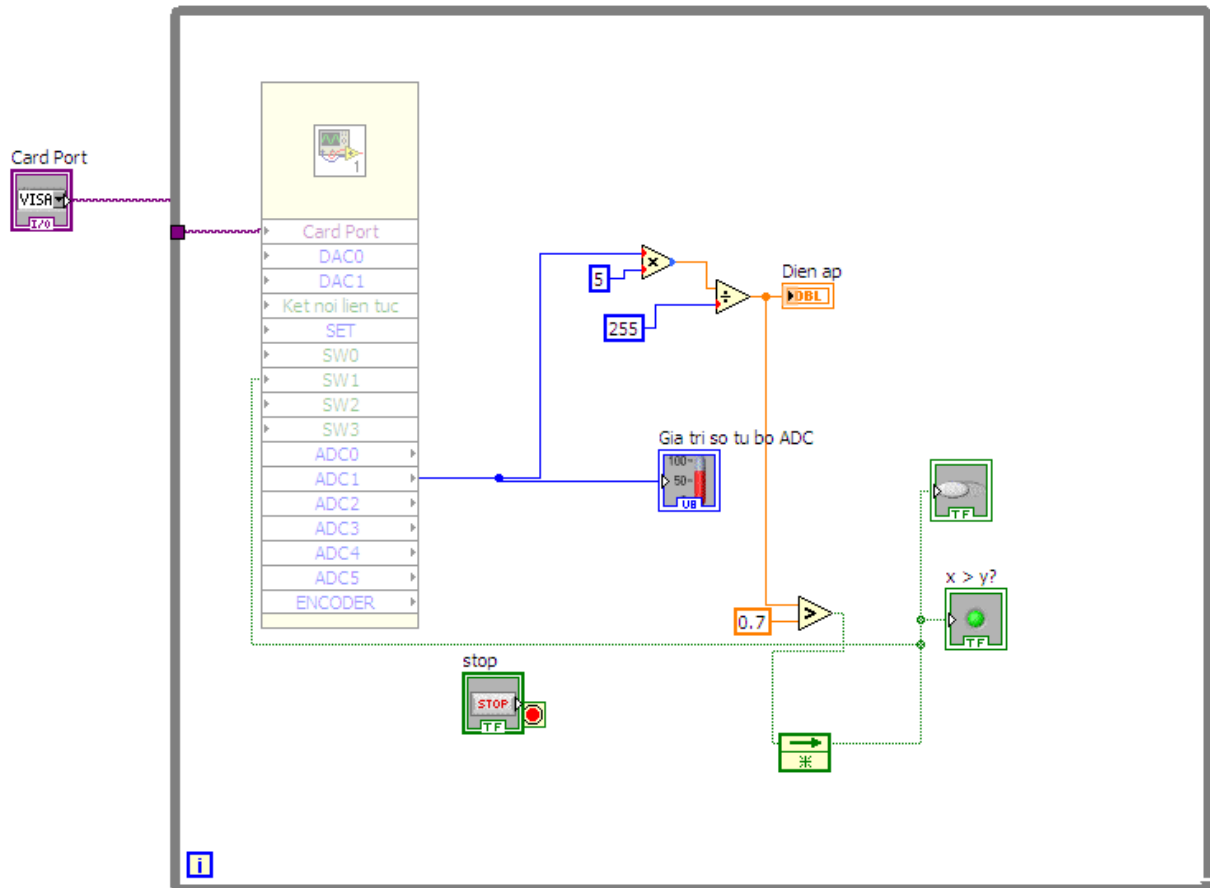
Chương trình LabVIEW trên máy tính tạo ra giao diện người dùng và điều khiển lò nhiệt điện trở.

Độ phân giải kênh ADC của Card USB-9001 là 8 bit, nên giá trị từ 0 đến 5V của sensor LM35 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 V_{out} của sensor mà ta đo được bằng đồng hồ Voltmeter.

Code LabVIEW:



CHƯƠNG 4:

KẾT LUẬN – HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI

4.1. Các kết quả đã thực hiện được trong đề tài

- Tìm hiểu về LabVIEW cơ bản, các giao thức kết nối, phần mềm nhúng vào LabVIEW, và các bộ công cụ đi kèm.
- Thiết kế và thi công hoàn thành giao diện người dùng (The Front Panel) và sơ đồ khối (The Block Diagram) trên LabVIEW để điều khiển và ổn định nhiệt độ lò nhiệt.
- Thiết kế hoàn thành mạch điều khiển nhiệt độ.
- Hoàn thành bản thuyết minh theo đúng thời gian được giao.

4.2. Những hạn chế

Đồ án của em tuy đã đáp ứng được yêu cầu của đề tài đặt ra, tuy nhiên vẫn còn nhiều hạn chế:

- Ứng dụng của LabVIEW rất rộng rãi, bản thân chúng em vẫn chưa tìm hiểu được nhiều. Giao thức kết nối giữa LabVIEW với thiết bị cũng đa dạng ngoài card PCI cũng có thể dùng giao tiếp qua cổng USB, RS232, RS485... Tuy nhiên chúng em mới tìm hiểu với card USB- 9001.
- Đối với mô hình lò nhiệt của chúng em, tín hiệu vào là điện áp, tín hiệu ra là nhiệt độ. Nhưng trong thực tế đối tượng còn chịu ảnh hưởng của rất nhiều yếu tố khác như: nhiệt độ môi trường, độ ẩm, gió, yếu tố làm mát... Tuy nhiên, bài toán ứng dụng của chúng em có hạn chế là có một tín hiệu vào và một tín hiệu ra (hệ SISO).

4.3. Hướng phát triển của đề tài

- LabVIEW là một công cụ mạnh trong đo lường và điều khiển. Ứng dụng LabVIEW điều khiển được nhiều đối tượng trong các lĩnh vực khác nhau. Chúng ta có thể dùng LabVIEW để kiểm tra, giám sát tự động trong quá trình sản xuất hay đánh giá chất lượng sản phẩm. Chúng ta cũng có thể dùng

LabVIEW để đo tín hiệu tương tự tốc độ cao hay ứng dụng để liên kết với PLC hoặc các phần cứng công nghiệp khác.

- Phát triển bài toán ứng dụng theo hướng nhiều đầu vào tác động. Mô hình trở thành hệ MISO (hệ nhiều đầu vào – một đầu ra). Khi đó thuật toán điều khiển sẽ thay đổi.

Em rất mong các em sinh viên khóa sau sẽ tiếp tục nghiên cứu, phát triển đề tài được hoàn thiện hơn và ứng dụng hiệu quả trong thực tế.

TÀI LIỆU THAM KHẢO

Tiếng việt

1. Evtatech – Vagam:Học nhanh LabVIEW tiếng việt
2. Hieutq: Tự học nhanh Labview 8.2 bằng hình ảnh.

Tiếng anh

3. Getting Stared With LabVIEW – Author (Apache Software Foundation).
4. LabVIEW Fundamentals - Author (Apache Software Foundation).
5. LabVIEW Quick Reference Card – National Instruments Corporation.
6. LabVIEW Advanced I, của hãng National Instrument.
7. LabVIEW Basics I + Basics II Course Manual của National Instrument.
8. PCI – 1710/1710 HG Multifunction DAS card for PCI bus user’s Manual.
9. LabVIEW Tutorial Manual - National Instruments Corporation.

Internet

10. <http://www.ni.com>
11. <http://www.advantech.com/support>
12. <http://www.dieukhientudong.htm>
13. http://techteach.no/labview/lv85/pid_control/index.htm
14. <http://www.dientuvienthong.net>