

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



ISO 9001:2015

ĐIỀU KHIỂN NHIỆT ĐỘ LÒ QUA GIAO DIỆN LABVIEW

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
NGÀNH ĐIỆN TỰ ĐỘNG CÔNG NGHIỆP

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



ISO 9001:2015

ĐIỀU KHIỂN NHIỆT ĐỘ LÒ QUA GIAO DIỆN LABVIEW

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

NGÀNH ĐIỆN TỰ ĐỘNG CÔNG NGHIỆP

Sinh viên: Vũ Trọng Tá

Người hướng dẫn: Th.S Ngô Quang Vĩ

Cộng Hoà Xã Hội Chủ Nghĩa Việt Nam

Độc lập – Tự Do – Hạnh Phúc

-----o0o-----

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên : Vũ Trọng Tá – MSV : 1512102032

Lớp : ĐC1901- Ngành Điện Tự Động Công Nghiệp

Tên đề tài : Điều khiển nhiệt độ lò qua giao diện LabVIEW.

CÁC CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ nhất:

Họ và tên : Ngô Quang Vĩ
Học hàm, học vị : Thạc sĩ
Cơ quan công tác : Trường Đại học dân lập Hải Phòng
Nội dung hướng dẫn : Toàn bộ đề tài

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 2019.

Yêu cầu phải hoàn thành xong trước ngày.....tháng.....năm 2020

Đã nhận nhiệm vụ Đ.T.T.N

Sinh viên

Đã giao nhiệm vụ Đ.T.T.N

Cán bộ hướng dẫn Đ.T.T.N

Vũ Trọng Tá

Th.S Ngô Quang Vĩ

Hải Phòng, ngày.....tháng.....năm 2020

HIỆU TRƯỞNG

GS.TS.NGŨT TRẦN HỮU NGHỊ

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN TỐT NGHIỆP

Họ và tên giảng viên:

Đơn vị công tác:

Họ và tên sinh viên: Chuyên ngành:

Nội dung hướng dẫn:

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

.....
.....
.....
.....
.....
.....

2. Đánh giá chất lượng của đồ án/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. Ý kiến của giảng viên hướng dẫn tốt nghiệp

Được bảo vệ Không được bảo vệ Điểm hướng dẫn

Hải Phòng, ngày ... tháng ... năm

Giảng viên hướng dẫn

(Ký và ghi rõ họ tên)

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN CHĂM PHẢN BIỆN

Họ và tên giảng viên:

Đơn vị công tác:

Họ và tên sinh viên: Chuyên ngành:

Đề tài tốt nghiệp:

.....

.....

1. Phần nhận xét của giáo viên chăm phản biện

.....

.....

.....

.....

2. Những mặt còn hạn chế

.....

.....

.....

.....

3. Ý kiến của giảng viên chăm phản biện

Được bảo vệ Không được bảo vệ Điểm hướng dẫn

Hải Phòng, ngày ... tháng ... năm

Giảng viên chăm phản biện

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Thấm thoát đã 4 năm trôi qua từ ngày đầu tiên em bước trên vào mái trường đại học thân yêu. Hoàn thành xong đồ án tốt nghiệp lần này cũng là lúc em kết thúc quãng thời gian ở nơi đây. Được học tập, nghiên cứu và rèn luyện bản thân tại trường Đại học Dân lập Hải Phòng là niềm tự hào và vinh dự của em cũng như gia đình em. Nhờ có mái trường thân yêu em được học hỏi biết bao nhiều tri thức kinh nghiệm thực tế quý báu của các thầy các cô cũng như được chia sẻ trao đổi với các bạn sinh viên tại đây. Đó chính là những thứ vô giá đồng hành cùng em xuyên suốt sự nghiệp sau này.

Em xin gửi lời cảm ơn trân thành và sâu sắc đến các thầy cô trong khoa Điện – Điện, Ban giám hiệu, Phòng đào tạo Trường Đại học Dân lập Hải Phòng đã tận tình giúp đỡ chỉ bảo em trong thời gian qua. Đồ án tốt nghiệp lần này là cột mốc quan trọng trong việc học tập miệt mài của em tại trường. Em cũng xin gửi lời cảm ơn đến các bạn sinh viên lớp DC1901 cũng như các bạn sinh viên trong trường đã nhiệt tình trao đổi và chia sẻ kiến thức luôn động viên và đoàn kết trong lúc gặp khó khăn.

Sau cùng, em muốn bày tỏ lòng biết ơn nhất đến thầy giáo Ths. Ngô Quang Vĩ đã nhiệt tình chỉ bảo, hướng dẫn tạo điều kiện thuận lợi để em được hoàn thiện đồ án một cách chính chu hoàn thiện nhất. Em xin kính chúc quý thầy cô sức khỏe, hạnh phúc và thành công trong sự nghiệp trồng người.

Sinh viên thực hiện

Vũ Trọng Tá

MỤC LỤC

Lời mở đầu	1
CHƯƠNG 1: TỔNG QUAN VỀ NGÔN NGỮ LẬP TRÌNH LabVIEW	3
1.1: Giới thiệu về LabVIEW	3
1.2: Giao diện của LabVIEW.....	5
1.2.1: Front panel	5
1.2.2: Block Diagram	6
1.3: Các thanh công cụ	7
1.3.1: Thanh công cụ của Front panel.....	7
1.3.2: Thanh công cụ của Block Diagram.....	8
1.3.3: Bảng điều khiển Palettes	8
1.4: Các bảng điều khiển và các bảng chức năng:	10
1.4.1: Bảng điều khiển (Controls Palette):	10
1.4.2: Bảng chức năng (Functions Palette):	13
1.5: Cấu trúc, hoạt động của vòng lặp:.....	21
1.5.1: While Loop:	21
1.5.2: For Loop:.....	22
CHƯƠNG 2: ĐIỀU KHIỂN Lò NHIỆT ĐIỆN TRỞ	24
2.1: Giới thiệu:	24
2.2: Ưu nhược điểm của lò điện so với các lò sử dụng nhiên liệu.....	24
2.3: Nguyên lý làm việc của lò điện trở	25
2.4: Các phương pháp điều khiển lò điện trở	25
2.4.1: Điều khiển dùng Role	27
2.4.2: Điều khiển dùng Thyristor	28
2.4.3: Kết luận	30
2.5: Các nguyên tắc điều khiển Thyristor (Triac)	31
2.5.1: Nguyên tắc thẳng đứng tuyến tính	31
2.5.2: Nguyên tắc điều khiển thẳng đứng “arccos”.....	32
2.5.3: Sơ đồ khối mạch điều khiển.....	33
CHƯƠNG 3: ĐIỀU KHIỂN NHIỆT ĐỘ Lò QUA GIAO DIỆN LABVIEW ..	35
3.1: Phương án thiết kế.....	35
3.1.1: Yêu cầu thiết kế.....	35
3.1.2: Phương án điều khiển.....	35
3.2: Giới thiệu Card USB-9090.....	39

3.3: Sensor LM35	46
3.4: Chương trình điều khiển bằng ngôn ngữ LabVIEW	47
3.4.1: Lập trình giao tiếp và điều khiển thiết bị với card USB 9090	47
3.4.2: Chương trình hoàn chỉnh	55
Chương 4: KẾT LUẬN	57
Tài liệu tham khảo.....	58

Lời mở đầu

Trong nền công nghiệp hiện đại, thuật ngữ đo lường và điều khiển đã trở nên quá quen thuộc. Hiện tại các hệ thống thiết bị điều khiển tự động đều mang một tỷ trọng rất lớn các phần tử ở mọi lớp cấu trúc có yếu tố kỹ thuật ghép nối với máy tính. Kỹ thuật ghép nối máy tính đã góp phần làm thay đổi bộ mặt các trung tâm điều khiển, nơi theo dõi – xử lý các tín hiệu, dữ liệu thu thập được để tối ưu hoá vận hành sản xuất để đạt được những yêu cầu nhiệm vụ được đề ra. Khi xây dựng các hệ thống đo lường và điều khiển kết nối với máy tính cần có sự hỗ trợ của các phần mềm chuyên dụng đảm bảo yêu cầu về tốc độ giao tiếp cũng như độ tương thích cao.

Yêu cầu về đo lường và điều khiển ngày càng phức tạp, để đáp ứng điều đó công ty National Instruments đã không ngừng nghiên cứu đổi mới phần mềm LabVIEW. Ngôn ngữ lập trình này ngày càng trở nên quen thuộc trong các hệ thống đo lường và điều khiển và là giá trị quan trọng cho các kỹ sư. Chính vì thế em đã chọn LabVIEW là đối tượng nghiên cứu và hỗ trợ cho đề tài lần này.

Đối tượng và phạm vi nghiên cứu:

Đối tượng nghiên cứu chung của đề án “Điều khiển nhiệt độ lò qua giao diện LabVIEW” này là:

- Tìm hiểu cách thức hoạt động và sử dụng phần mềm LabVIEW như sau: tên gọi và nguồn gốc LabVIEW, chức năng của LabVIEW trong thực tế, giao diện phần mềm, chức năng của các công cụ, cấu trúc hoạt động của các vòng lặp.

- Nghiên cứu Card USB HDL 9090: cách kết nối với máy tính, lập trình giao tiếp và điều khiển với thiết bị với card.

- Điều khiển và ổn định nhiệt độ lò nhiệt điện trở sử dụng mạch điều khiển Triac.

Ý nghĩa thực tiễn của đề án:

LabVIEW là một phần mềm phổ biến và được ứng dụng rộng rãi trong các ngành công nghiệp tự động. Môi trường LabVIEW mở tương thích hầu hết với tất cả các phần cứng hiện nay mà điển hình là Card USB HBL 9090 từ đó tạo mã nguồn và khả năng kết nối tới hàng nghìn thiết bị giúp tập hợp dữ liệu dễ dàng và đơn giản. Từ đó chúng ta có thể kết hợp những ứng dụng mà LabVIEW mang lại vào các hệ thống hiện nay. Chính vì thế mà LabVIEW trở thành lựa chọn giá trị cho các kỹ sư Việt Nam và thế giới trong thời đại hiện nay.

Nội dung thực hiện đề tài của 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 nhiệt độ lò nhiệt điện trở giao tiếp với máy tính bằng LabVIEW.

Chương 4: Kết luận.

Hải Phòng, tháng 10 năm 2019

Sinh viên thực hiện

Vũ Trọng Tá

CHƯƠNG 1: TỔNG QUAN VỀ NGÔN NGỮ LẬP TRÌNH LabVIEW

1.1: Giới thiệu về LabVIEW

LabVIEW (viết tắt của nhóm từ Laboratory Virtual Instrumentation Engineering Workbench) là một phần mềm máy tính được phát triển bởi công ty National Instruments, Hoa kỳ. LabVIEW còn 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 trình C, Pascal. Bằng cách diễn đạt cú pháp thông qua các hình ảnh trực quan trong môi trường soạn thảo, LabVIEW đã được gọi với tên khác là lập trình G (viết tắt của Graphical, nghĩa là đồ họa).

LabVIEW (Virtual Instrument Engineering Workbench) là một môi trường phát triển dựa trên ngôn ngữ lập trình đồ họa, thường được sử dụng cho mục đích đo lường, kiểm tra, đánh giá, xử lý, điều khiển các tham số của thiết bị. LABVIEW là một ngôn ngữ lập trình đa năng, giống như các ngôn ngữ lập trình hiện đại khác. LABVIEW gồm có các thư viện thu nhận dữ liệu, một loạt các thiết bị điều khiển, phân tích dữ liệu, biểu diễn và lưu trữ dữ liệu. Nó còn có các công cụ phát triển được thiết kế riêng cho việc nối ghép và điều khiển thiết bị.

LABVIEW khác với các ngôn ngữ lập trình thông thường ở điểm cơ bản là: các ngôn ngữ lập trình khác thường dùng trên cơ chế dòng lệnh, trong khi đó LABVIEW dùng ngôn ngữ lập trình Graphical để tạo ra các chương trình ở dạng sơ đồ khối.

Trong LABVIEW ta xây dựng giao diện người sử dụng bằng việc thiết lập các công cụ và các đối tượng. Giao diện người sử dụng được hiểu như là một Front Panel rồi sau đó ta đưa code vào trong sơ đồ khối để điều khiển các đối tượng ở trên front panel. Sơ đồ khối cũng có thể hiểu giống như một lưu đồ thuật toán.

LABVIEW được tích hợp đầy đủ các chức năng giao tiếp với các phần cứng GPIB, VXI, PXI, RS-232, RS-485, các thiết bị thu nhận dữ liệu. LABVIEW cũng xây dựng các đặc trưng cho việc kết nối các ứng dụng của ta với Web sử dụng LABVIEW Web Server và, chuẩn mạng TCP/IP và Active X.

LABVIEW được dùng nhiều trong các phòng thí nghiệm, lĩnh vực khoa học kỹ thuật như tự động hóa, điều khiển, điện tử, cơ điện tử, hàng không, hóa sinh, điện tử y sinh,... Hiện tại ngoài phiên bản LABVIEW cho các hệ điều hành

Windows, Linux, Hãng NI đã phát triển các mô-đun LABVIEW cho máy hỗ trợ cá nhân (PDA).

Các chức năng chính của LabVIEW có thể tóm tắt như sau:

- 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 thông qua các cổng giao tiếp: RS232, RS485, USB, PCI, 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 so với các ngôn ngữ khác như Visual Basic, Matlab, ...
- Cho phép thực hiện các thuật toán điều khiển như PID, Logic mờ (*Fuzzy Logic*), một cách nhanh chóng thông qua các chức năng tích hợp sẵn trong LABVIEW.
- Cho phép kết hợp với nhiều ngôn ngữ lập trình truyền thống như C, C++, ...

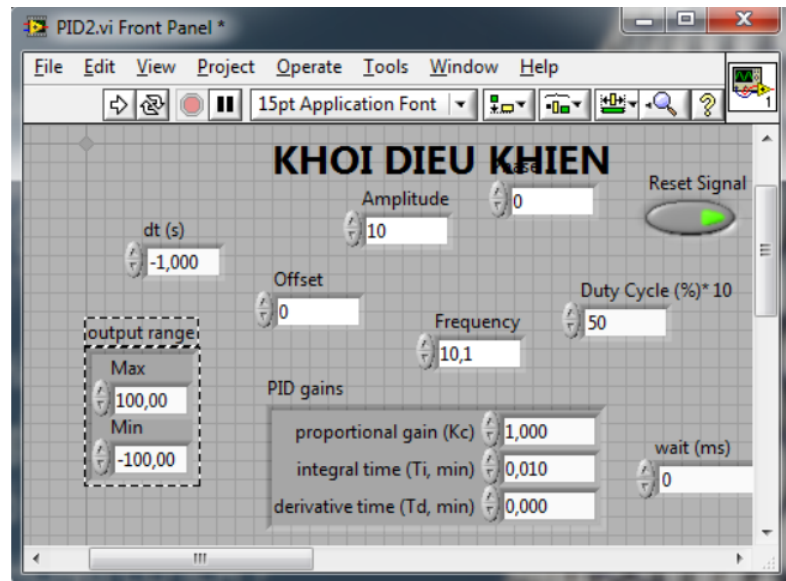
Các phần mềm nhúng vào LabVIEW

- Wolfram Research Mathematica
- Microsoft Excel
- The MathWorks MATLAB and Simulink
- MathSoft Math CAD
- Electronic Workbench MultiSim
- Texas Instruments Code Composer Studio
- Ansoft RF circuit design software
- Microsoft Access
- Microsoft SQL Server
- Oracle

1.2: Giao diện của LabVIEW

1.2.1: Front panel

Là giao diện của người sử dụng. Ví dụ sau đây minh họa front panel.



Hình 1.1: Front panel

Xây dựng front panel với các bộ điều khiển (controls) và các hiển thị (Indicators), chúng được sử dụng với các chức năng vào ra dữ liệu. Các điều khiển bao gồm các núm (knobs), nút ấn (push buttons), mặt đồng hồ và các thiết bị vào dữ liệu khác. Control là các đối tượng được đặt trên Front Panel để cung cấp dữ liệu cho chương trình. Nó tương tự như đầu vào cung cấp dữ liệu.

Để mở giao diện người dùng trước tiên ta phải khởi động LabVIEW. Đối với Windows 7 trở xuống ta chọn Start > All Program > National Instruments LabVIEW. Khi đó phần mềm sẽ bắt đầu khởi động và sau 1 khoảng thời gian sẽ xuất hiện cửa sổ Getting Started. Tại cửa sổ này chúng ta có thể chọn mới (NEW) BlankVI để mở VI trống, Empty Project hoặc VI from Template (VI mẫu) để mở giao diện người dùng. Hoặc mở những giao diện đã được viết sẵn như trên hình minh họa ở mục (Open). Chúng ta cũng có thể tùy chọn tạo mới theo yêu cầu ở mục New> More hay tùy chọn mở ở các nơi lưu trữ ngoài thư viện mặc định ở Browse

Cửa sổ Getting Started được biểu thị bởi hình dưới đây:

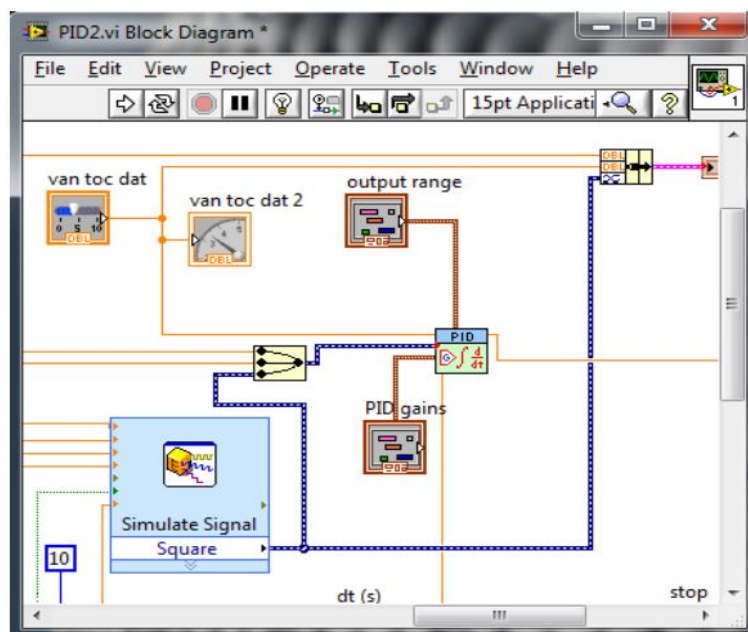


Hình 1.2: Cửa sổ Getting Started

1.2.2: Block Diagram

Là sơ đồ khối: Block Diagram của 1 VI là một sơ đồ được xây dựng trên môi trường LABVIEW, nó có thể gồm nhiều đối tượng và các hàm khác nhau để tạo các câu lệnh để chương trình thực hiện. Block Diagram là một mã nguồn đồ họa của 1 VI. Các đối tượng trên Front Panel được thể hiện bằng các thiết bị đầu cuối trên Block Diagram, không thể loại bỏ các thiết bị đầu cuối trên Block Diagram. Các thiết bị đầu cuối chỉ mất đi sau khi loại bỏ đối tượng tương ứng trên Front panel.

Ví dụ sau đây minh họa Block Diagram:



Hình 1.3: Block Diagram

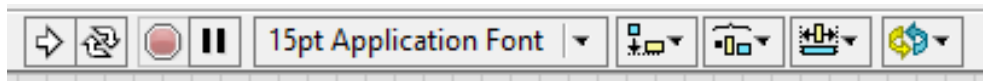
Cấu trúc của một Block Diagram gồm các thiết bị đầu cuối (Terminal). Nút (Node) và các dây nối (wire).

- Terminal: là các cổng mà dữ liệu truyền qua giữa Block Diagram và Front panel, và giữa các Node trong Block Diagram. Các Terminal nằm ở dưới dạng các Icon của các Function.
- Nodes: là các phần tử thực hiện chương trình, chúng tương tự như các mệnh đề, toán tử, hàm và các chương trình con trong các ngôn ngữ lập trình thông thường.
- Wires: là các dây nối dữ liệu giữa các node.

1.3: Các thanh công cụ


1.3.1: Thanh công cụ của Front panel


Sử dụng các nút ấn của thanh công cụ dùng để chạy và tạo ra một VI. Thanh công cụ xuất hiện trên front panel có dạng như sau:





Hình 1.4: Thanh công cụ của Front panel

Trong đó:


 Kích nút **Run** dùng để chạy một VI, trong đó VI chạy thì trạng thái nút ấn thay đổi theo nếu VI không không có lỗi gì thì trạng thái có dạng như sau:

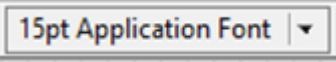
 Khi nút ấn **Run** ở trạng thái này thì có nghĩa VI của ta đang bị lỗi nào đó mà ta cần phải xử lý. Để tìm lỗi ta kích đúp vào nút này để hiển thị danh sách toàn bộ các lỗi trong VI của ta.


 Kích vào nút **Run Continuously** để chạy VI liên tục cho đến khi muốn huỷ hay dừng lại. Ta cũng có thể ấn tiếp nút lệnh này để không cho phép chạy liên tục.


 Trong lúc VI chạy, nút hủy bỏ hoạt động xuất hiện và nếu ta ấn vào biểu tượng này thì chương trình đang chạy dừng ngay lập tức. Với một chú ý nên tránh dùng nút lệnh này để dừng một VI, bởi vì ta sẽ không biết trạng thái


của VI. Ta nên thiết kế chương trình dừng VI ví dụ ta có thể sử dụng chuyển mạch ở front panel.

 Kích vào nút lệnh **Pause** để tạm dừng chương trình VI đang chạy. Khi ta kích vào nút lệnh **Pause** thì LABVIEW sẽ làm sáng vị trí ta dừng hoạt động trong sơ đồ khối. Khi ta muốn chạy tiếp chương trình thì ta ấn lại nút lệnh này.

 Dùng để thiết lập font cho VI bao gồm kích thước, kiểu loại, màu sắc.

 Dùng để sắp xếp các đối tượng thẳng hàng nhau bao gồm các đường thẳng đứng, mép trên, trái ...


 Dùng để phân bố các đối tượng...


 Lựa chọn **Reorder** khi ta có các đối tượng gối lên nhau và ta muốn định nghĩa đối tượng là đứng trước hay đứng sau. Việc lựa chọn một trong các đối tượng với việc định vị vị trí của nó rồi sau đó di chuyển lên phía trước hay di chuyển về phía sau...


1.3.2: Thanh công cụ của Block Diagram




Hình 1.5: Thanh công cụ Block Diagram

 Kích vào nút lệnh **Highlight Execution** ta sẽ thấy luồng dữ liệu chạy trong sơ đồ khối. Khi ta kích lại nút lệnh này quá trình sẽ bị dừng lại.

 Kích vào nút lệnh **Step into** dùng để lặp từng bước một trong vòng lặp, subVI.

 Kích vào nút lệnh **Step over** dùng để bỏ qua một vòng lặp hoặc một subVI.

 Kích vào nút lệnh **Step out** dùng để nhảy ra ngoài vòng lặp hoặc subVI.

1.3.3: Bảng điều khiển Palettes

Việc lập trình trên LABVIEW cần sử dụng các bảng: Tools Palette, Controls Palette, Functions Palette, các bảng đó cung cấp các chức năng để người sử dụng có thể tạo và thay đổi trên giao diện Front Panel và Block Diagram.

Tools Palettes


Tools Palettes xuất hiện trên cả Front Panel và Block Diagram. Bảng này cho phép người sử dụng có thể xác lập các chế độ làm việc đặc biệt của con trỏ chuột. Khi lựa chọn một công cụ, biểu tượng của con trỏ sẽ được thay đổi theo biểu tượng của công cụ đó


Nếu thiết lập chế độ tự động lựa chọn công cụ và người sử dụng di chuyển con trỏ qua các đối tượng trên Front Panel hoặc Block Diagram, LABVIEW sẽ tự động lựa chọn công cụ phù hợp trên bảng Tools Palette


Để truy cập vào **Tools palette** ta chọn Menu: **View→Tools palette**. Các công cụ trong **Tools palette** gồm có:





Hình 1.6: Tool panel


 Operating tool: Dùng để thay đổi giá trị điều khiển hoặc lựa chọn văn bản trong điều khiển.


 Positioning tool: Dùng để lựa chọn, di chuyển, thay đổi các kích thước đối tượng.


 Labeling tool: Dùng để soạn thảo văn bản dạng text và tạo ra các nhãn.


 Wiring tool: Dùng để nối các đối tượng lại với nhau trong sơ đồ khối


 Object shortcut menu: Dùng để truy cập vào một đối tượng bằng cách kích chuột trái.

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

 Breakpoint tool: Dùng để thiết lập điểm dừng trên các VI, functions, nút, dây nối, các cấu trúc lệnh để dừng hoạt động ở tại vị trí này

 Probe tool: Dùng để tạo ra những đầu dò trên các dây nối trong sơ đồ khối. Việc sử dụng Probe tool dùng để kiểm tra các giá trị trung gian trong VI.

 Color copy tool: Dùng để copy các màu cho việc paste bằng việc sử dụng Coloring tool.

 Coloring tool: Dùng để tô màu cho một đối tượng. Nó cũng có thể hiển thị ngay việc thiết lập màu sắc mặt trước và màu nền.

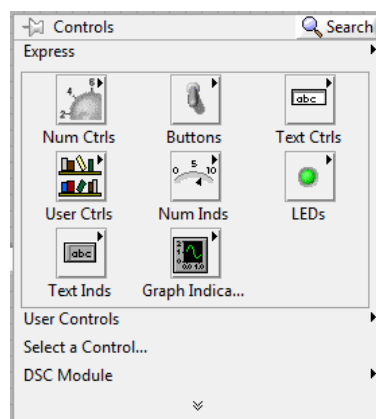
1.4: Các bảng điều khiển và các bảng chức năng:

Các bảng điều khiển và các bảng chức năng là các bảng của đối tượng được sử dụng để tạo ra các VI. Để sử dụng đối tượng trên bảng ta kích vào đối tượng vào đặt nó vào trong front panel hoặc là trong sơ đồ khối.

Sử dụng các nút chỉ dẫn trên bảng **Controls** và bảng **Functions** để xác định và tìm kiếm các điều khiển, các VI và các hàm. Ta cũng có thể làm bằng cách kích chuột phải vào biểu tượng VI ở trên bảng và chọn **Open VI** từ menu phím tắt để mở VI.

1.4.1: Bảng điều khiển (Controls Palette):

Bảng điều khiển chỉ duy nhất xuất hiện trên Front panel. Bảng điều khiển chứa các bộ điều khiển (control) và các bộ hiển thị (Indicator). Bảng điều khiển được minh họa như hình dưới đây.



Hình 1.7: Bảng điều khiển

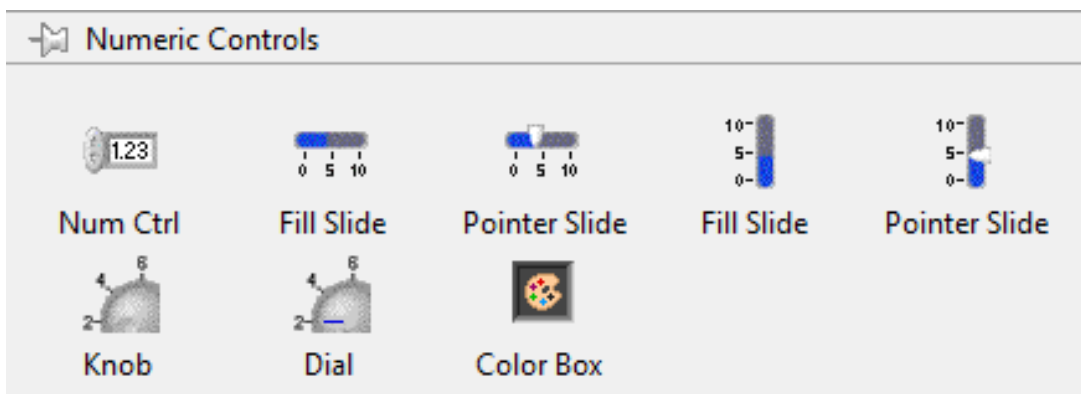
Bảng điều khiển được sử dụng để người sử dụng thiết kế cấu trúc mặt hiển thị gồm các thiết bị ví dụ: các công tắc, các loại đèn, các loại màn hình hiển thị... Với bảng điều khiển này, người sử dụng có thể chọn các bộ thiết bị chuẩn do hãng sản xuất cung cấp.

Bảng điều khiển dùng để cung cấp dữ liệu đầu vào và hiển thị các kết quả đầu ra.

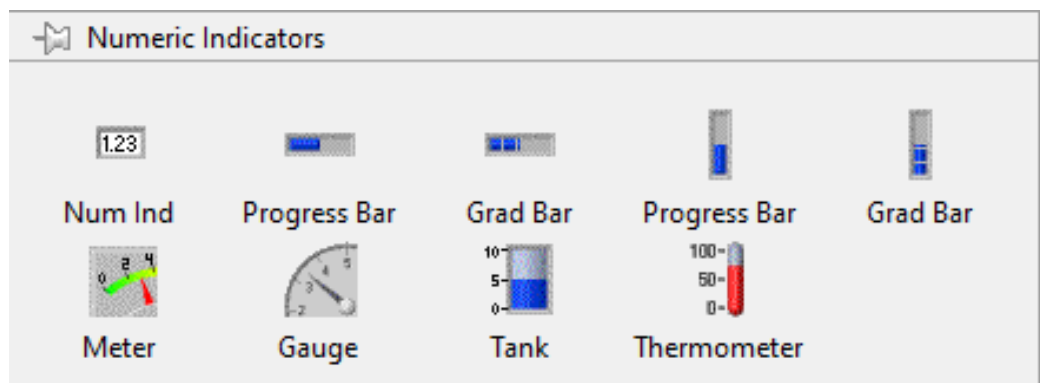
Một số bộ điều khiển và hiển thị trên controls palette:

Numeric Controls/Indicators:

Bộ công cụ này được sử dụng để hiển thị và điều khiển dữ liệu dạng số trong chương trình:



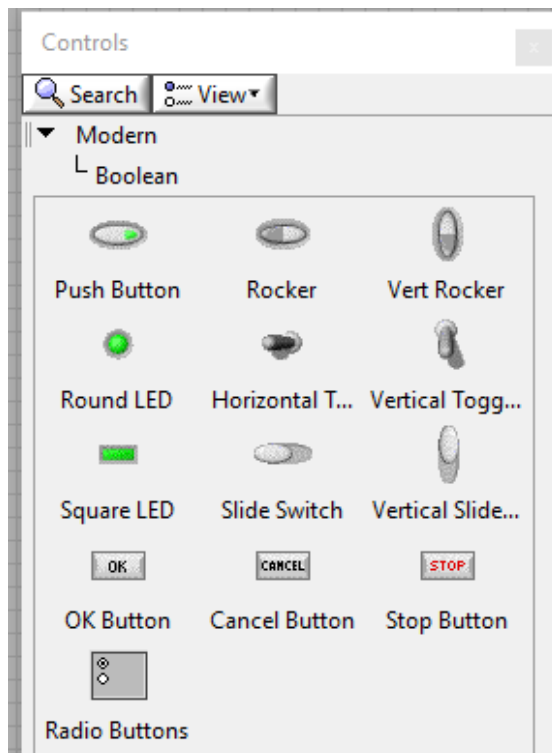
Hình 1.8: Numeric control



Hình 1.9: Numeric Indicators

Boolean Controls/Indicators:

Bộ công cụ này cung cấp 2 giá trị là True và False. Khi thực hiện chương trình người sử dụng sử dụng chuột để điều khiển giá trị của thiết bị. Việc thay đổi giá trị của các thiết bị chỉ có tác dụng khi các thiết bị đó được xác lập ở chế độ là các Control. Còn nếu ở chế độ là các Indicator thì giá trị không thay đổi vì chúng chỉ là các thiết bị hiển thị.

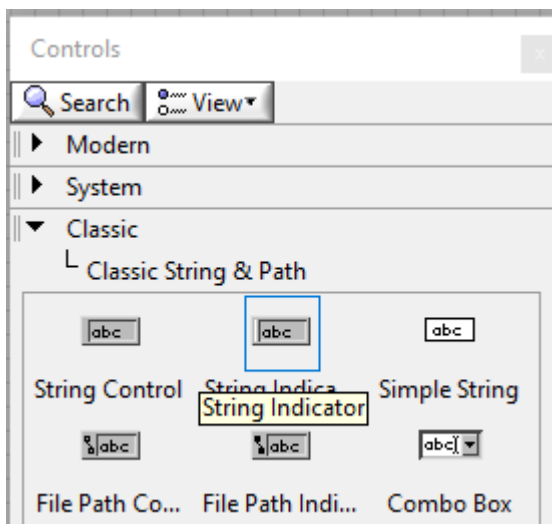


Hình 1.10: Boolean

Tại cửa sổ bảng điều khiển Control → chọn Search → nhập Boolean → Click chọn Boolean<<Modern>> cửa sổ Boolean như hình trên sẽ xuất hiện.

String Controls/Indicators:

Các điều khiển này dùng để nhập và hiển thị các ký tự, nó cũng có thể được xác lập ở chế độ đầu vào hay đầu ra.



Hình 1.11: String Controls/Indicators

Tại bảng điều khiển chọn Search → nhập String & Path → Click chọn String & Path<<Modern>> cửa sổ String & Path sẽ xuất hiện.

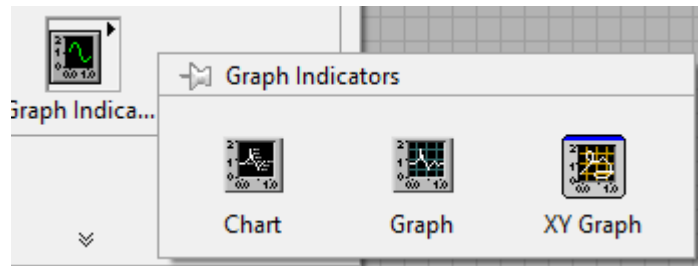
Graph Indicator:

Bao gồm Graph 2D và Graph 3D

Graph 2D Gồm:

Waveform graph: Biểu diễn những hàm đơn trị có dạng $y = f(x)$, với những khoảng chia ngang nhau trên các trục.

XY graph: biểu diễn các hàm đa trị như đường tròn hay dạng sóng thay đổi theo thời gian.

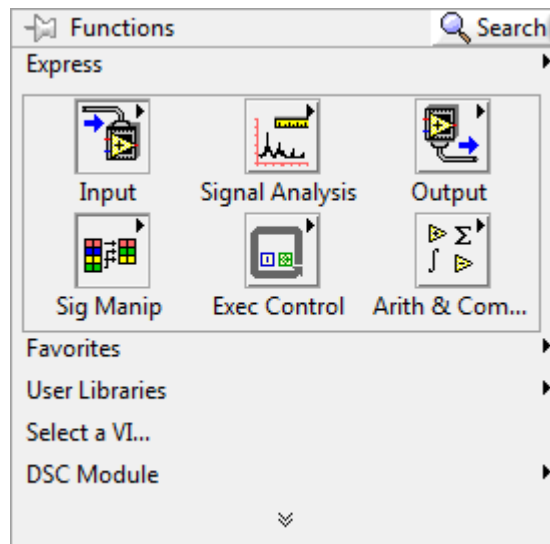


Hình1.12: Graph Indicator

Tại cửa sổ bảng điều khiển Controls chọn Graph Indicator.

1.4.2: Bảng chức năng (Functions Palette):

Bảng Functions palette chỉ xuất hiện trên Block Diagram. Bảng này chứa các VIs và các hàm mà người sử dụng dùng để xây dựng nên các khối lưu đồ. Bảng chức năng có dạng như sau:



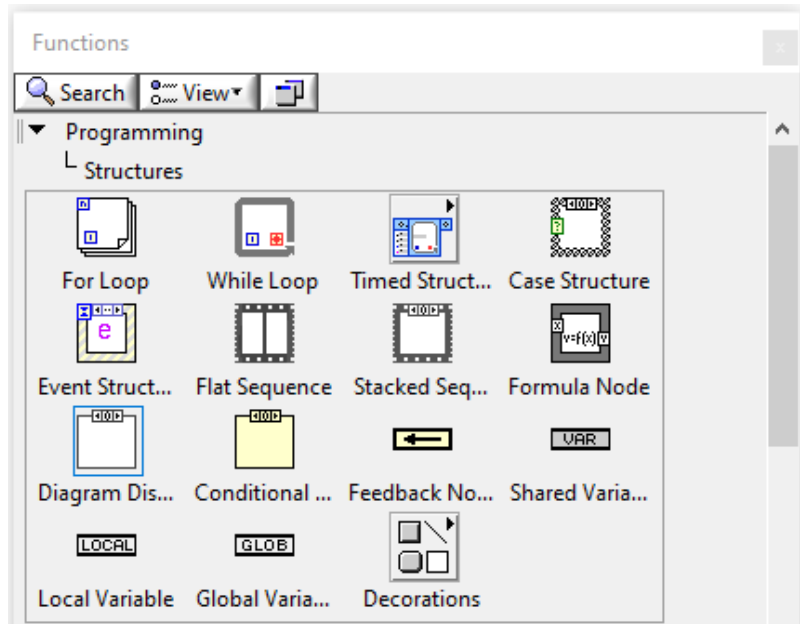
Hình 1.13: Funtions

Với bảng Function Palette, người lập trình thực hiện các cú pháp ví dụ: phép lặp, phép lựa chọn thông qua các nhóm hàm, chức năng đã được cung cấp bên cạnh đó từ bảng này người sử dụng có thể tạo ra và sử dụng lại các hàm, chức năng mà người sử dụng tự xây dựng. Các hàm toán học được minh họa

thông qua các biểu tượng. Khi muốn lựa chọn thực hiện một hàm nào đó thì người sử dụng chọn biểu tượng thể hiện cho hàm đó và có thể kéo thả ở bất kỳ vị trí nào trên Block Diagram sau đó xác định những đầu vào và đầu ra cần thiết.

Structures Function: Hàm cấu trúc

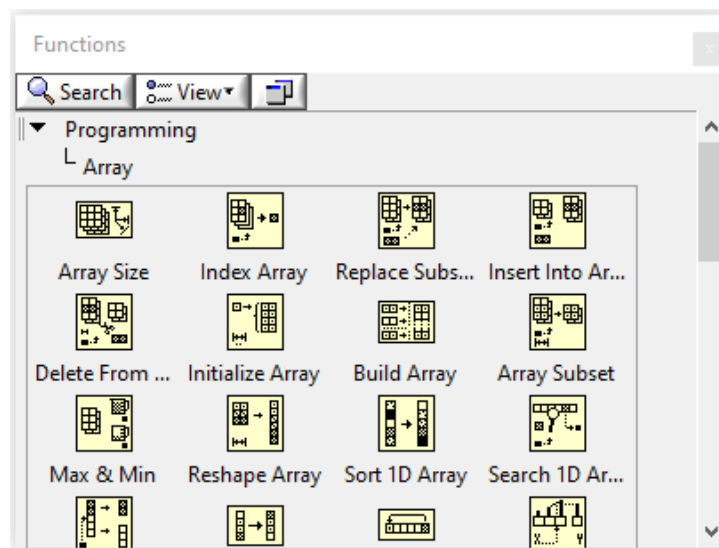
Bao gồm vòng lặp For, While, cấu trúc Case, Sequence, Các biến toàn cục và cục bộ. Tại bảng điều khiển Functions → Search → nhập Structures và chọn cửa sổ Structures sẽ xuất hiện như hình dưới đây:



Hình 1.14: Structures

Function Array: Hàm mảng

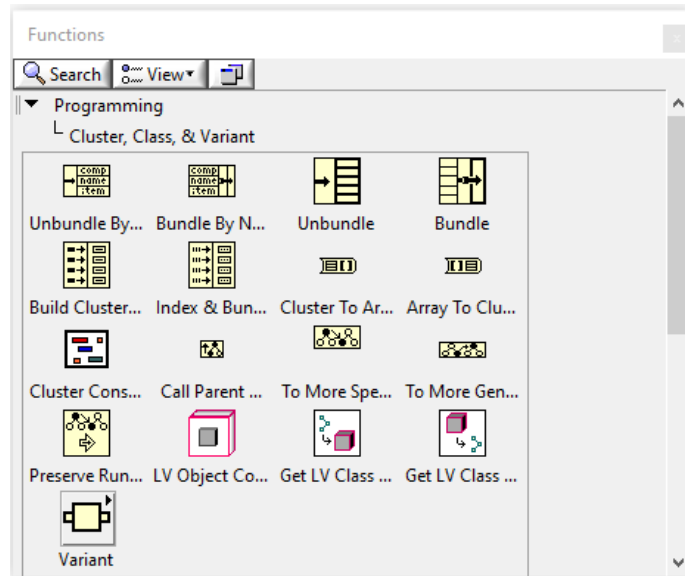
Được dùng để tạo ra và điều khiển các mảng. Từ bảng điều khiển Functions → Search → nhập và chọn Array, cửa sổ Array sẽ xuất hiện như hình dưới đây.



Hình 1.15: Array

Cluster, Class & Variant: Hàm cụm & biến thể

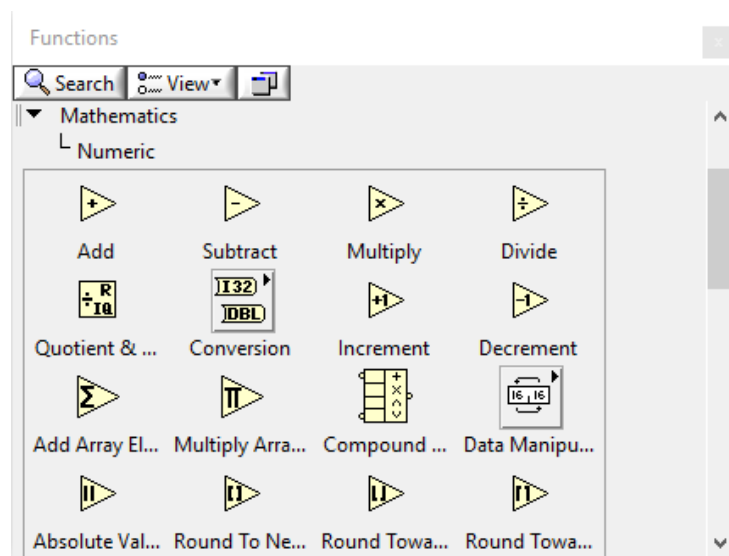
Hàm này được sử dụng để tạo ra và điều khiển các cụm, chuyển đổi dữ liệu LabVIEW từ một khuôn dạng, có thể thao tác độc lập kiểu dữ liệu, thêm những thuộc tính tới dữ liệu, và chuyển đổi dữ liệu biến thể tới dữ liệu LabVIEW. Tại bảng điều khiển Functions → Search → Nhập và chọn Cluster, Class & Variant cửa sổ sẽ xuất hiện như hình dưới đây:



Hình 1.16: Cluster, Class & Variant

Numeric: Hàm số học

Hàm này được sử dụng để tạo ra và thực hiện những thao tác số học, lượng giác, Logarit, số phức toán học trong các số và chuyển đổi những số từ một kiểu dữ liệu này sang một kiểu dữ liệu khác. Từ bảng điều khiển Functions → Search → nhập và chọn Numeric. Cửa sổ sẽ xuất hiện như hình dưới đây:

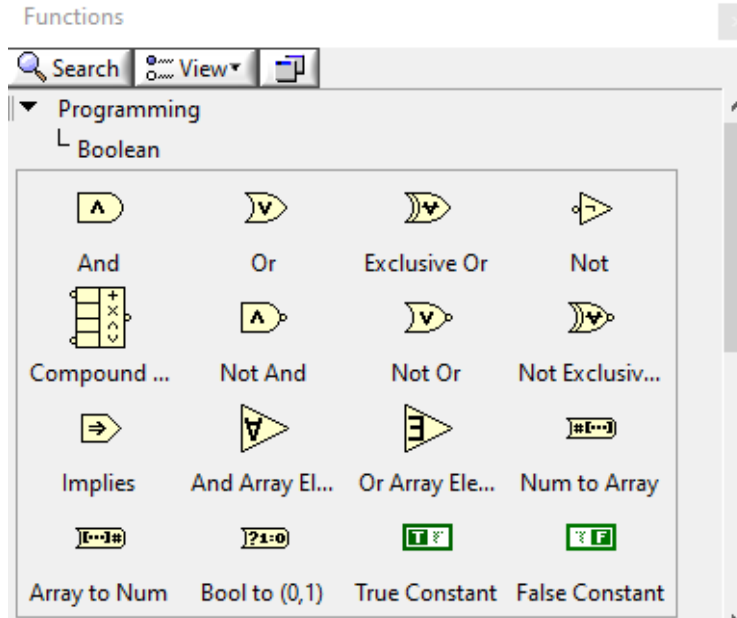


Hình 1.17: Numeric

Boolean Function: Hàm Boolean

Chứa các hàm logic như and, or, xor, nor và các hàm logic phức tạp khác.

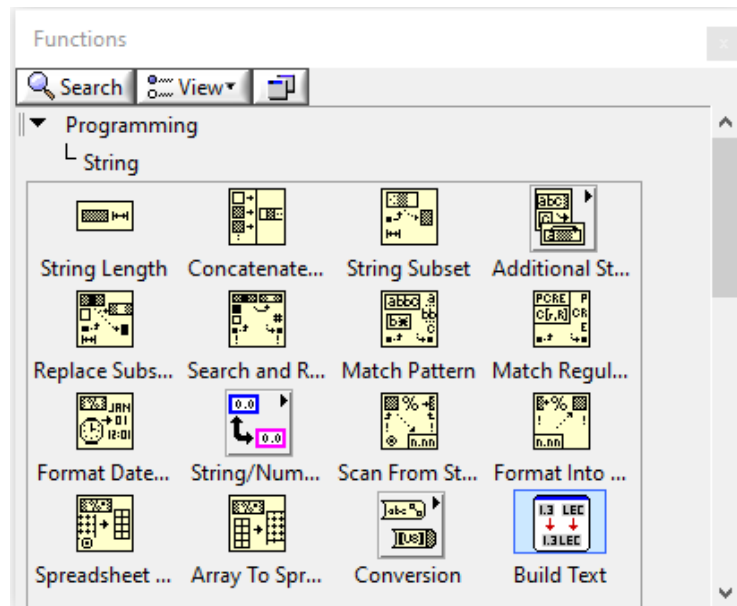
Tại bảng điều khiển Function → Search → Nhập và chọn Boolean. Cửa sổ sẽ hiện ra như hình dưới đây:



Hình 1.18: Boolean Function

String Function: Hàm chuỗi

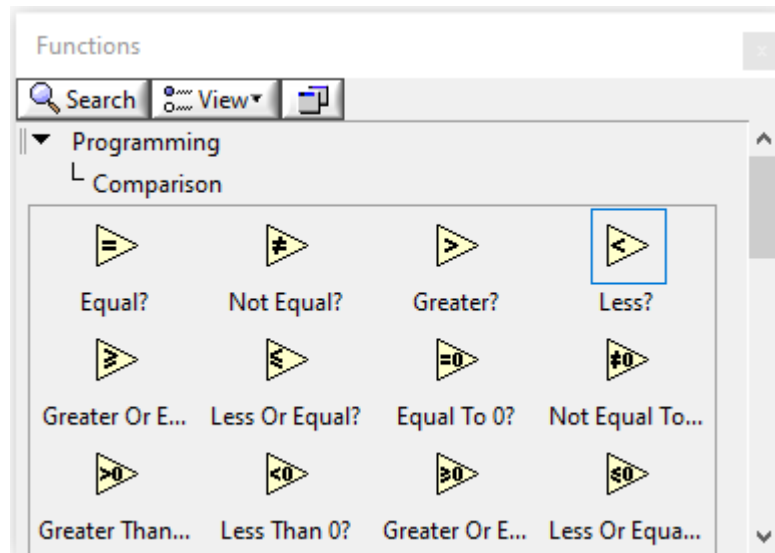
Hàm dùng để liên kết hai hay nhiều chuỗi, tách một tập con của các chuỗi từ một chuỗi, chuyển dữ liệu vào bên trong chuỗi, định dạng một chuỗi sử dụng trong một công đoạn xử lý từ hoặc ứng dụng bảng biểu. Tại bảng điều khiển Function → Search → nhập và chọn String. Cửa sổ sẽ hiện ra như hình dưới đây:



Hình 1.19: String Function

Comparison: Hàm So Sánh

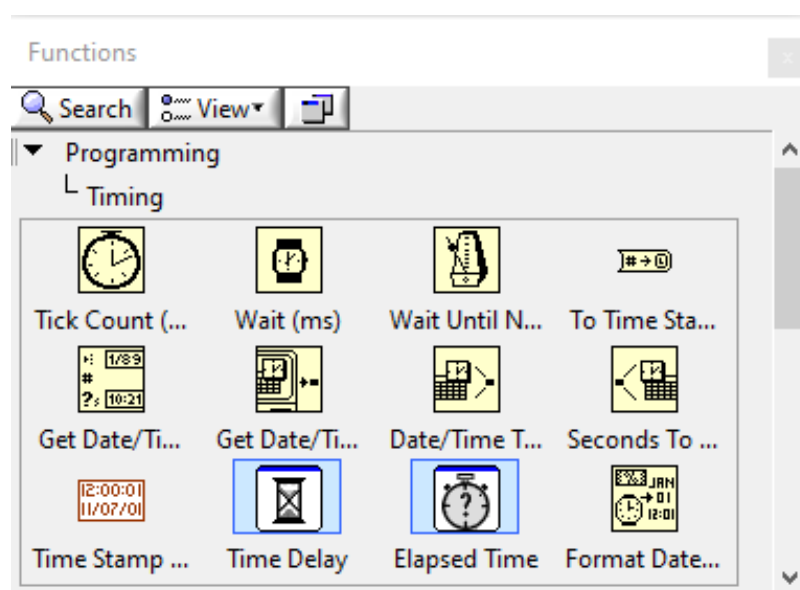
Hàm này sử dụng để so sánh các giá trị đại số Bool, các chuỗi, các giá trị số, các mảng và các cụm. Hàm so sánh xử lý các giá trị Boolean, String, Numeric, Array và Cluster khác nhau. Ta có thể thay đổi phương pháp so sánh của vài hàm Comparison. Từ bảng điều khiển Function → Search → Nhập và chọn Comparison. Cửa sổ sẽ xuất hiện như hình dưới:



Hình1.20 : Comparison

Timing: Hàm thời gian

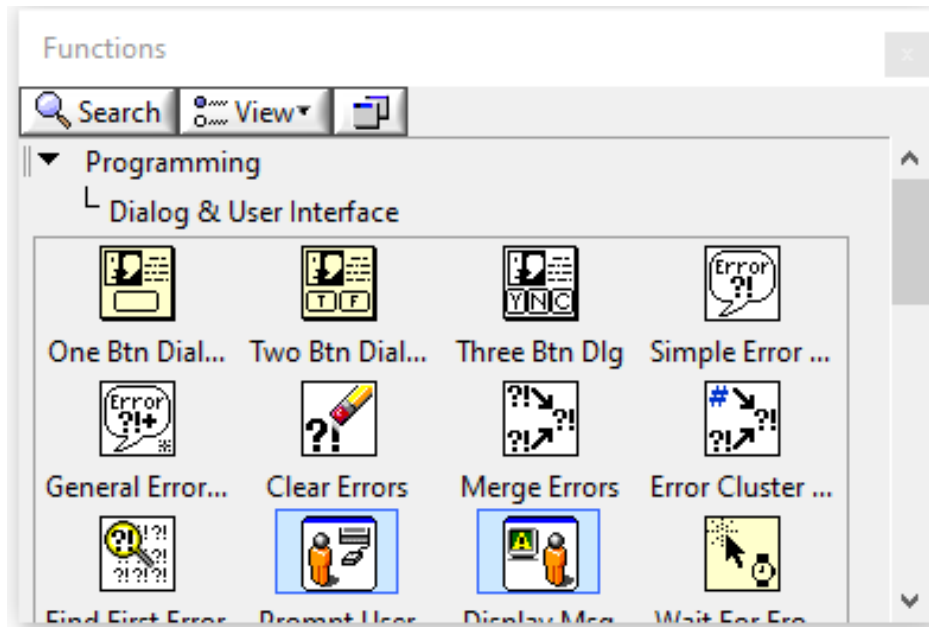
Hàm này sử dụng để xác định dòng thời gian, đo khoảng thời gian trôi hoặc trì hoãn một tiến trình trong một khoảng thời gian xác định. Tại bảng điều khiển Function→Search→Nhập và chọn Timing. Cửa sổ Timing sẽ xuất hiện như hình dưới:



Hình 1.21: Timing

Dialog & User Interface

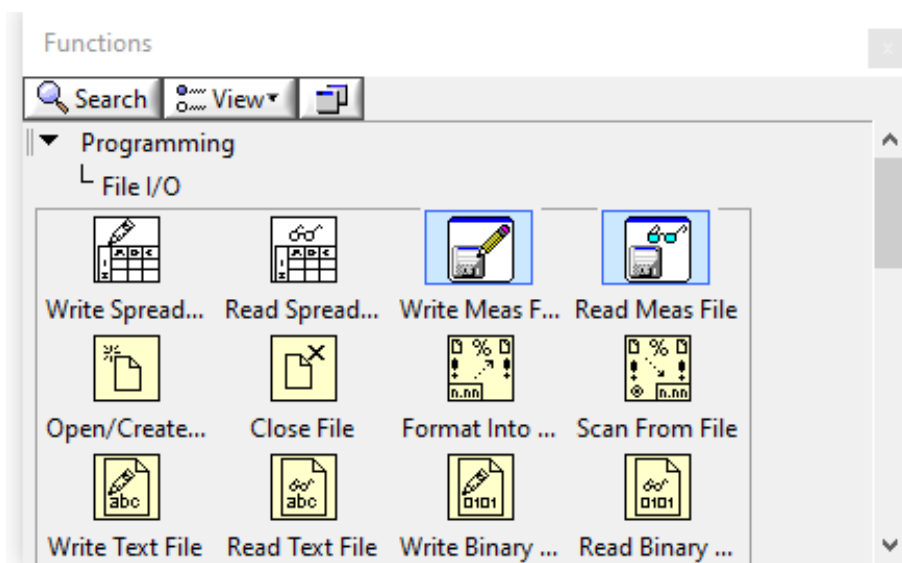
Hàm này được sử dụng để tạo ra các hộp thoại tới nhắc nhở người sử dụng với các chỉ dẫn. Tại bảng điều khiển Function→Search→Nhập và chọn Dialog & User Interface. Cửa sổ Dialog & User Interface sẽ xuất hiện như hình dưới đây:



Hình 1.22: Dialog & User Interface

File I/O

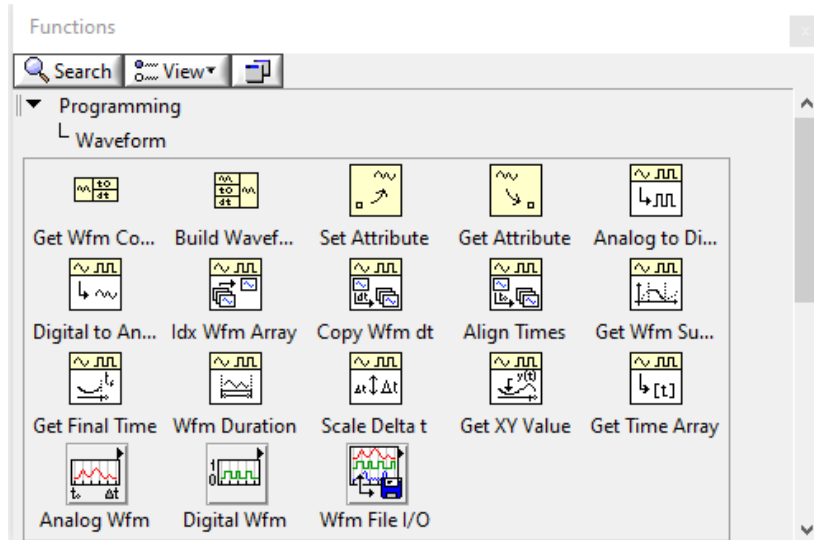
Hàm này thực hiện các chức năng cho một tập tin như lưu, mở tập tin theo dạng nhị phân, spreadsheet, đóng một tập tin...Ngoài ra hàm này còn chứa các chức năng mở rộng khác về lưu trữ dữ liệu. Từ Bảng điều khiển Function→Search→Nhập và chọn File I/O. Cửa sổ File I/O sẽ xuất hiện như hình dưới đây:



Hình 1.23: File I/O

Waveform: Hàm dạng sóng

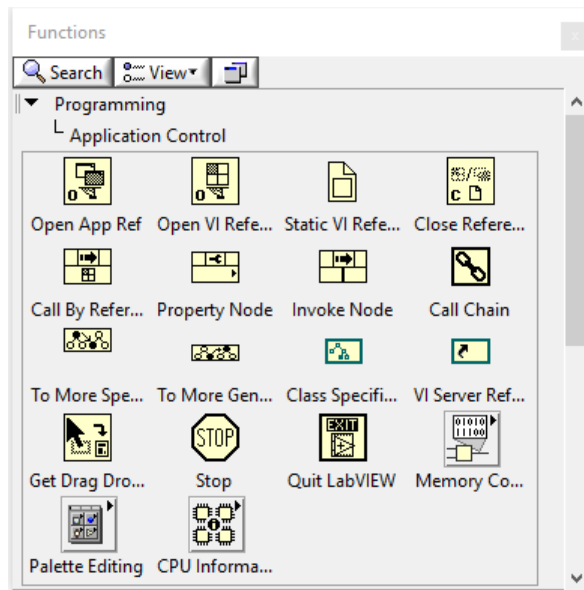
Sử dụng hàm này để xây dựng dạng sóng mà bao gồm các giá trị dạng sóng, thay đổi thông tin, để thiết lập và khôi phục các thành phần và thuộc tính của dạng sóng. Từ bảng điều khiển Function → Search → Nhập và chọn Waveform. Cửa sổ Waveform sẽ xuất hiện như hình dưới đây:



Hình 1.24: Waveform

Application control: Hàm điều khiển ứng dụng

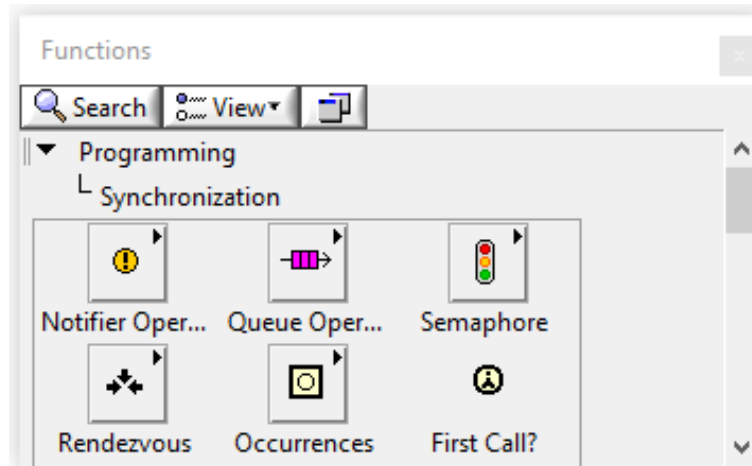
Hàm này được sử dụng để lập trình các VI điều khiển và các ứng dụng LabVIEW trên máy tính hiện tại hoặc qua một mạng. Ta có thể sử dụng các VI và các hàm chức năng này để định dạng nhiều VI tại cùng một thời điểm. Tại bảng điều khiển Function → Search → Nhập và chọn Application control và cửa sổ sẽ xuất hiện như hình dưới đây:



Hình 1.25: Application control

Synchronization: Hàm đồng bộ

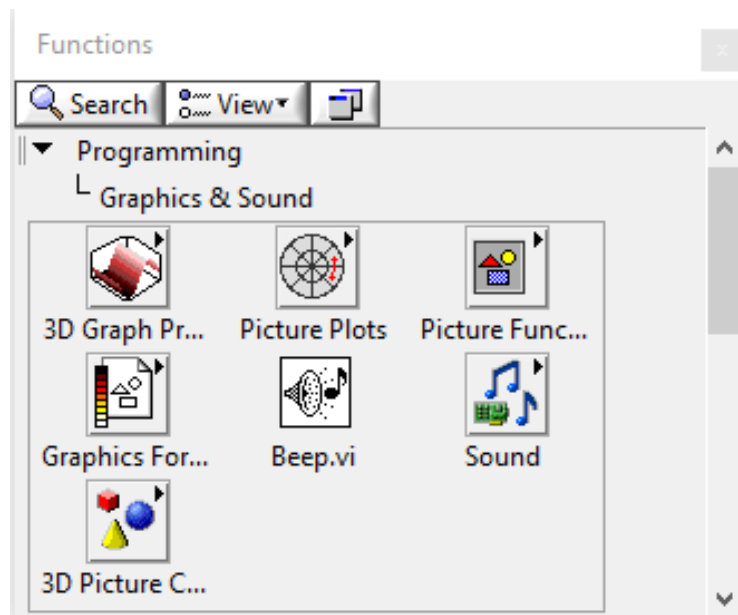
Sử dụng hàm này để đồng bộ các nhiệm vụ thi hành song song và để chuyển dữ liệu giữa các nhiệm vụ song song. Tại bảng điều khiển Function → Search → Nhập và chọn Synchronization cửa sổ sẽ xuất hiện như hình dưới:



Hình 1.26: Synchronization

Graphic & Sound: Hàm đồ họa và âm thanh

Với yêu cầu hiển thị, dữ liệu công vào ra phải là đồ họa và âm thanh ta sẽ sử dụng hàm này. Tại bảng điều khiển Function → Search → Nhập và chọn Graphic & Sound. Cửa sổ sẽ xuất hiện như hình dưới đây:

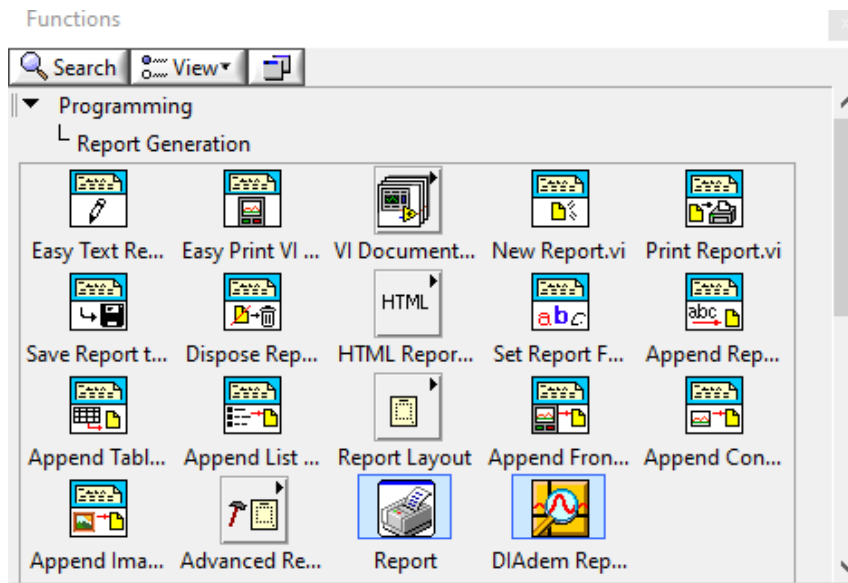


Hình 1.27 Graphic & Sound

Report Generation: Hàm phát sinh báo cáo

Hàm này được sử dụng để tạo ra và điều khiển các báo cáo của các ứng dụng LabVIEW. Tại bảng điều khiển Function → Search → Nhập và chọn Report

Generation. Cửa sổ Report Generation sẽ xuất hiện:



Hình 1.28: Report Generation

1.5: Cấu trúc, hoạt động của vòng lặp:

1.5.1: While Loop:

Giống như vòng lặp Do Loop hoặc Repeat – Until Loop trong ngôn ngữ văn bản. Vòng lặp While Loop là một cấu trúc lặp thực hiện sơ đồ bên trong nó cho đến khi giá trị boolean đưa tới conditional terminal (một terminal đầu vào) là trùng với điều kiện được thiết lập để thực hiện vòng lặp. để truy cập while loop ta chọn menu: **functions** → **structures** → **while loop**. Sau đó sử dụng con trỏ kích và kéo tạo ra vùng mong muốn mà ta muốn lặp.

Biểu tượng của while loop được minh họa ở hình dưới đây.



Hình 1.29: Vòng lặp While loop

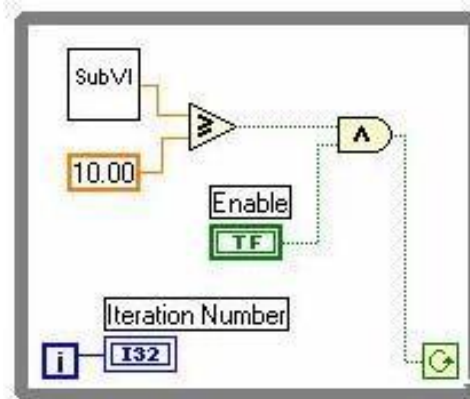
VI kiểm tra Conditional Terminal tại cuối mỗi vòng lặp, do đó While Loop luôn thực hiện ít nhất một lần. Iteration Terminal là một Terminal đầu ra mà đưa ra số lần vòng lặp thực hiện được. Việc tính số lần lặp luôn được bắt đầu từ 0. Vì vậy, nếu vòng lặp chạy một lần thì Iteration Terminal đưa ra kết quả 0.

Việc thực hiện vòng lặp có thể được xác định thông qua Conditional Terminal. Tại Conditional Terminal, ta có thể chọn các điều kiện:

- Stop if true.
- Continue if true.

Việc xác định điều kiện để thực hiện vòng lặp tại Conditional Terminal rất quan trọng vì nếu không xác định đúng thì vòng lặp có thể rơi vào vòng lặp vô hạn.

Trong sơ đồ khối sau hoạt động của vòng lặp While Loop hoạt động cho đến khi giá trị đầu ra từ subVI là bé hơn 10 hoặc điều khiển **Enable** là FALSE. Đầu ra của hàm AND là TRUE chỉ khi hai đầu vào là TRUE, ngoài ra khi một trong các đầu vào là FALSE đầu ra là FALSE.



Hình 1.30: While loop với điều kiện lặp

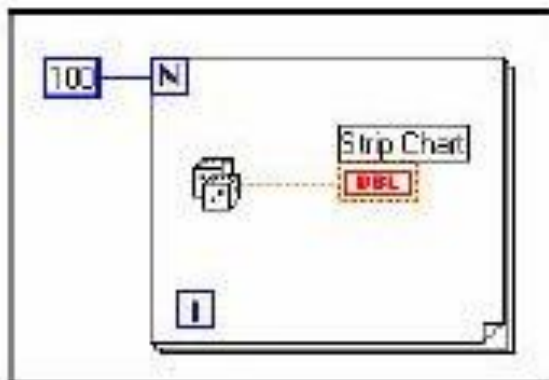
1.5.2: For Loop:

- Cấu trúc vòng lặp For Loop là quá trình thực hiện lặp trong sơ đồ khối với số vòng lặp xác định. Ta chọn vòng lặp For Loop từ vị trí Functions → Structures → For loop và chứa đoạn mã mà ta muốn lặp nằm bên trong vòng for loop. Một vòng lặp For Loop là một hộp có kích thước nào đó bao gồm 2 terminal: count terminal (là đầu vào của terminal) và iteration terminal (đầu ra của terminal). Count terminal là số lần lặp. Iteration terminal chứa số lần lặp đã được thực hiện.
- Cấu trúc vòng lặp For Loop khác với cấu trúc vòng lặp While Loop ở chỗ vòng lặp For Loop hoạt động với số lần lặp xác định. Trong khi đó vòng lặp While Loop chỉ dừng quá trình lặp khi giá trị điều kiện được kiểm tra là đúng. Hoạt động của vòng lặp For Loop tương đương với đoạn mã sau:

For i = 0 to N-1

- Bên trong sơ đồ khối thực hiện và lặp lại đến khi đến giá trị N-1.

- Ví dụ sau đây minh họa hoạt động của vòng lặp phát ra 100 số ngẫu nhiên và hiển thị lên đồ thị:



Hình 13: For loop

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 tỏa ra một lượng nhiệt theo định luật Jun-Lenxơ:

$$Q = I^2RT$$

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:

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

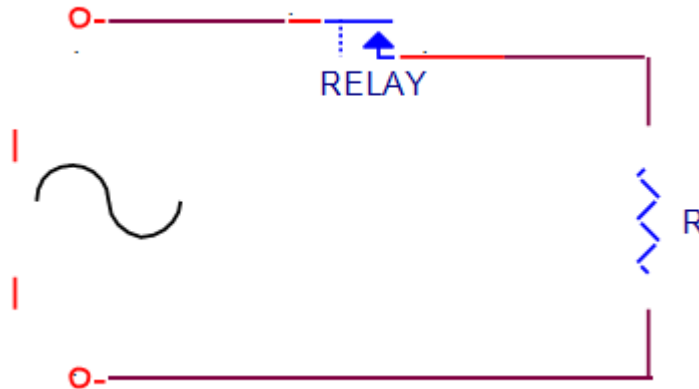
Với các tham số K, T được xác định một cách gần đúng theo yêu cầu công nghệ.

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

Sơ đồ nguyên lý



Hình 2.1: Sơ đồ điều khiển bằng Role

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

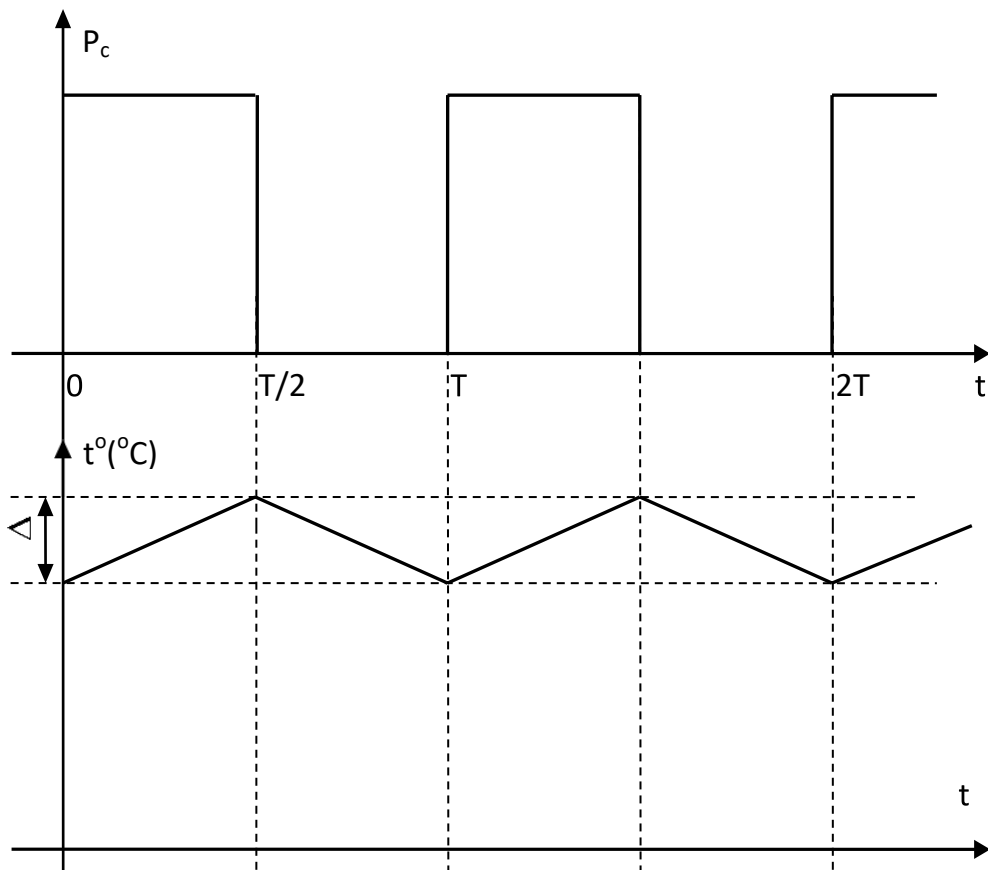
$$\Delta P = P_c - P_t$$

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.

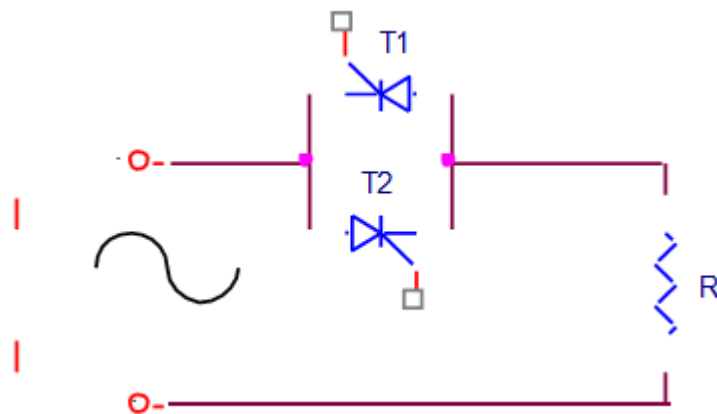
Một cách trực quan ta có đồ thị quan hệ giữa công suất và nhiệt độ theo thời gian sau:



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

2.4.2: Điều khiển dùng Thyristor

Sơ đồ nguyên lý



Hình 2.3: Sơ đồ điều khiển bằng Thyristor

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.

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:

$$U_{tb} = \sqrt{\frac{1^2}{2\pi} \int_0^\pi u(t)^2 dt}$$

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

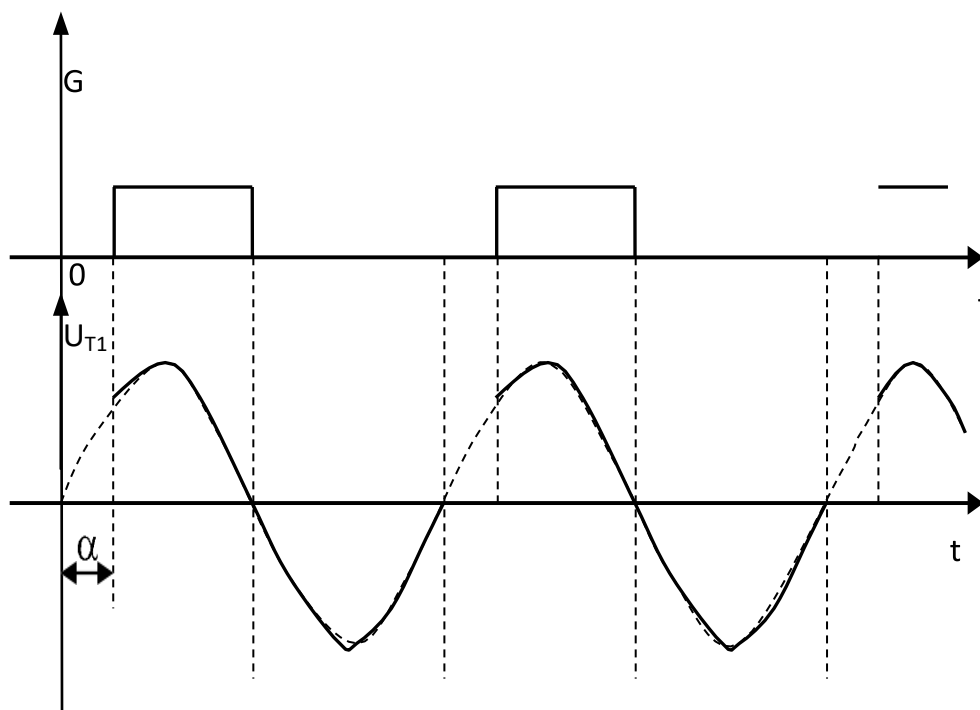
$$U_{tb} = U_0 \sqrt{\frac{2\pi - 2\alpha + \sin \alpha}{2}}$$

$$\text{Với } U_0 = \sqrt{\frac{U^2}{\pi}}$$

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.

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 nhiều như: nhiệt độ buồng đốt, nhiệt độ vật nung, nhiệt độ môi trường... sẽ ảnh hưởng đến nhiệt lượng tỏa 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 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 “arcs” 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 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:

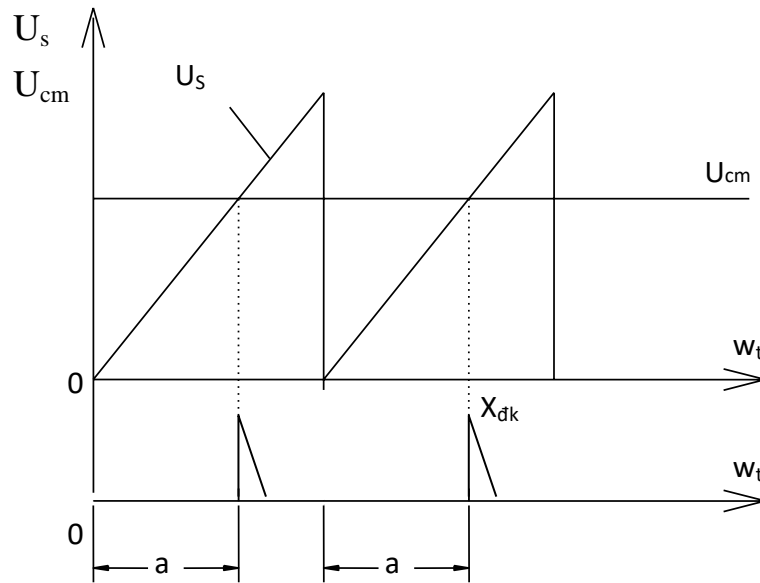
$$U_d = U_{cm} - U_s$$

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:

$$\alpha = \pi \frac{U_{cm}}{U_{sm}} \text{ người ta lấy } U_{cm\max} = U_{sm}$$



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 $U_s = U_m \cos \omega t$.

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:

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

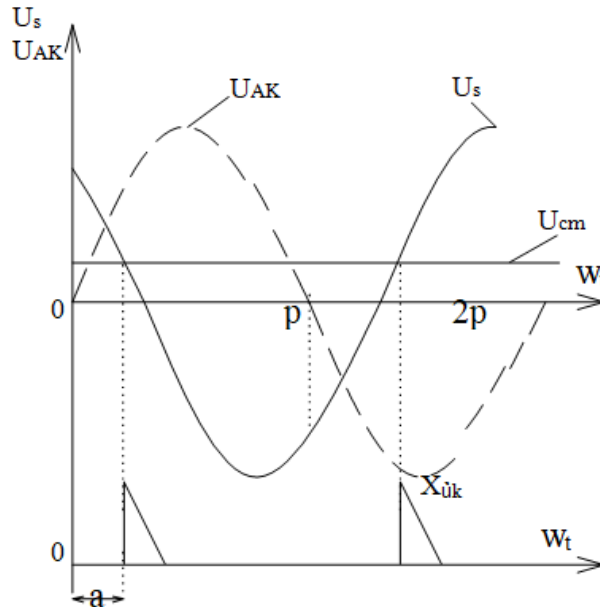
Khi $U_{cm} = U_m$ thì $\alpha = 0$.

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 .

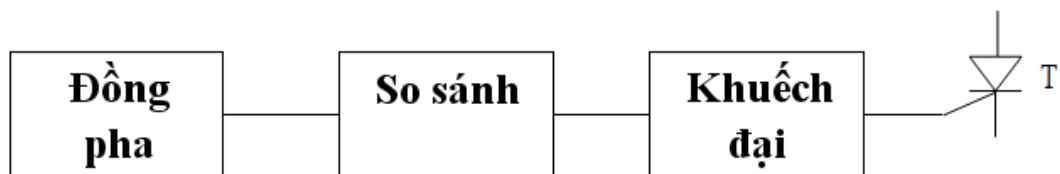
Nguyên tắc điều khiển thẳng đứng “arccos” được sử dụng trong các thiết bị chỉnh lưu đòi hỏi chất lượng cao.



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 $U_{đk}$, tìm thời điểm hai điện áp này bằng nhau ($U_{đk} = U_{rc}$). 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 Thyristor 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 NHIỆT ĐỘ LÒ QUA GIAO DIỆN LABVIEW

3.1: Phương án thiết kế

3.1.1: Yêu cầu thiết kế

Lò nhiệt điện trở 220V 100W

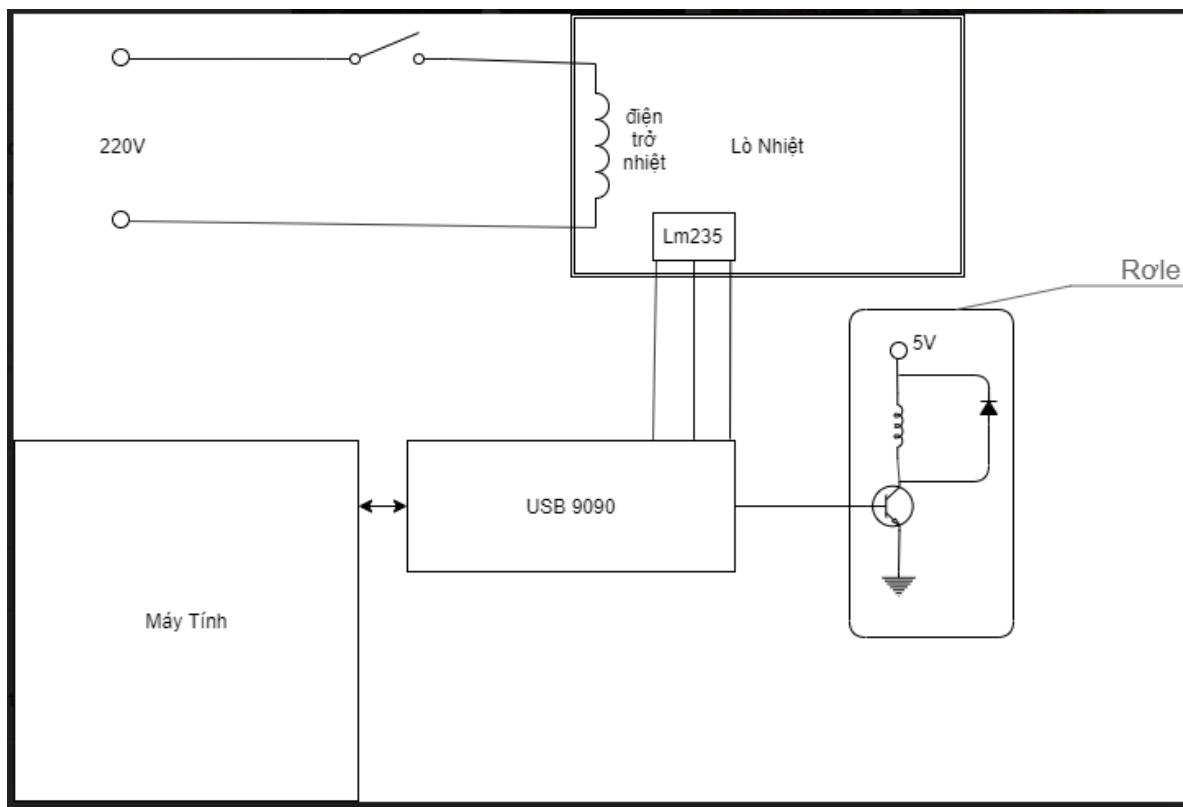
- Sử dụng sensor cảm biến nhiệt LM35 để đo nhiệt độ.
- Sử dụng USB-9090 để ghép nối giữa máy tính và khối công suất điều khiển lò
- Áp dụng phần mềm LabVIEW viết chương trình điều khiển và tạo giao diện điều khiển trên máy tính.

3.1.2: Phương án điều khiển

Chọn phương án điều khiển:

a. Sử dụng rơle kiểu ON/OFF:

Phần cứng được thiết kế theo sơ đồ sau ON/OFF



Hình 3.1: Sơ đồ phân cứng kiểu ON/OFF

Mạch hoạt động theo nguyên lý sau:

Điều khiển theo phương pháp xung

- Khi nhiệt độ trong lò nhỏ hơn nhiệt độ đặt (dựa vào sensor LM35), khi đó $U_{dk} = 5V$ Role đóng tiếp điểm, điện trở nhiệt trong lò được cấp điện 220v tạo nhiệt độ cho lò.

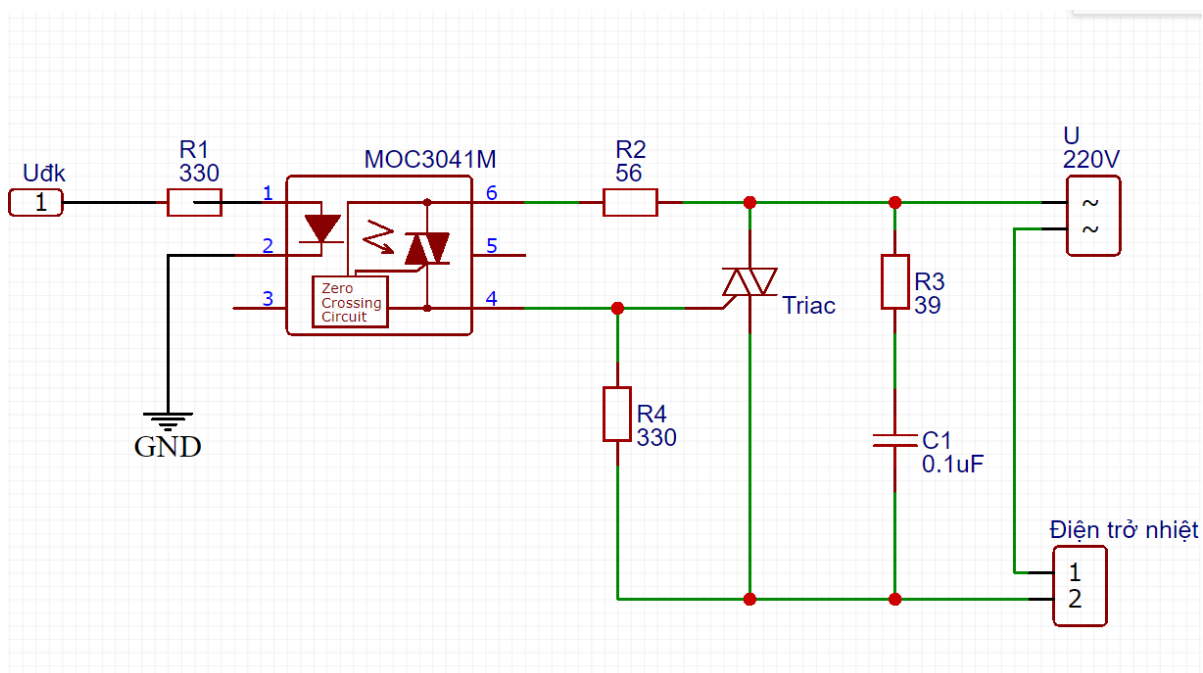
- Khi nhiệt độ trong lò lớn hơn nhiệt độ đặt, khi đó Role mở tiếp điểm làm cho điện trở trong lò ko còn dòng điện đi qua. Nhiệt độ trong lò tạm thời dừng lại ở nhiệt độ đặt cho đến khi giảm dần cho đến khi thấp hơn sẽ lặp lại quá trình bên trên.

Ưu điểm:

- Giá thành rẻ, phần cứng dễ dàng thực hiện
- Nhược điểm:
- Độ trễ lớn
- Nhiệt độ chênh lệch không ổn định
- Chịu ảnh hưởng của nhiễu.

b. Thay thế Role bằng Triac:

Mạch điều khiển Triac được thiết kế theo sơ đồ sau:



Hình 3.2 : Mạch điều khiển Triac

Mạch hoạt động theo nguyên lý sau:

- Khi nhiệt độ trong lò nhỏ hơn nhiệt độ đặt, LM35 gửi tín hiệu điện áp tới chân ADC của USB 9090. Bằng phần mềm LabVIEW máy tính xử lý dữ liệu và gửi lại tín hiệu số $U_{dk} = 5V$. Lúc này MOC3041M dẫn kích mở Triac cho dòng điện chạy qua, điện trở nhiệt lúc này được cấp điện 220v tạo nhiệt độ cho lò.

- Khi nhiệt độ trong lò lớn hơn nhiệt độ đặt, tương tự như trên nhưng khi này $U_{dk} = 0V$. MOC3041M không còn dẫn, Triac không cho dòng điện chạy qua, điện trở lúc này không còn được cấp dòng điện 220V. Nhiệt độ trong lò không tiếp tục tăng lên mà dừng lại và giảm dần cho đến khi lặp lại quá trình ban đầu.

Ưu điểm:

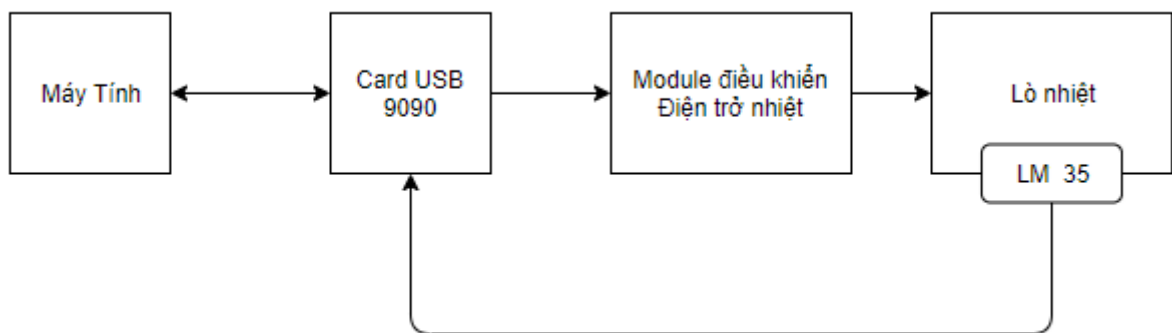
- Nhiệt độ ổn định
- Cách ly về điện
- Ít chịu ảnh hưởng của nhiễu
- Nhược điểm:
- Giá thành cao, thực hiện phức tạp

Do đó

⇒ Chọn phương án điều khiển bằng Module điều khiển Triac

Phần điều khiển được thiết kế theo sơ đồ sau:

Mô hình điều khiển được xây dựng như sau:



Hình 3.2: Sơ đồ điều khiển lò nhiệt điện trở

Trong đó:

- Máy tính: tạo giao diện và điều khiển lò nhiệt bằng phần mềm LabVIEW thông qua kết nối với card USB 9090.

- Card USB 9090: giao tiếp với máy tính qua cổng USB. Nó nhận dữ liệu nhiệt độ từ LM35 qua ADC chuyển thành tín hiệu số đưa về máy tính. Bằng phần mềm LABVIEW máy tính sẽ xử lý, thông qua card USB 9090 tạo ra U_{dk} , điều khiển Triac (Module điều khiển Điện trở nhiệt).

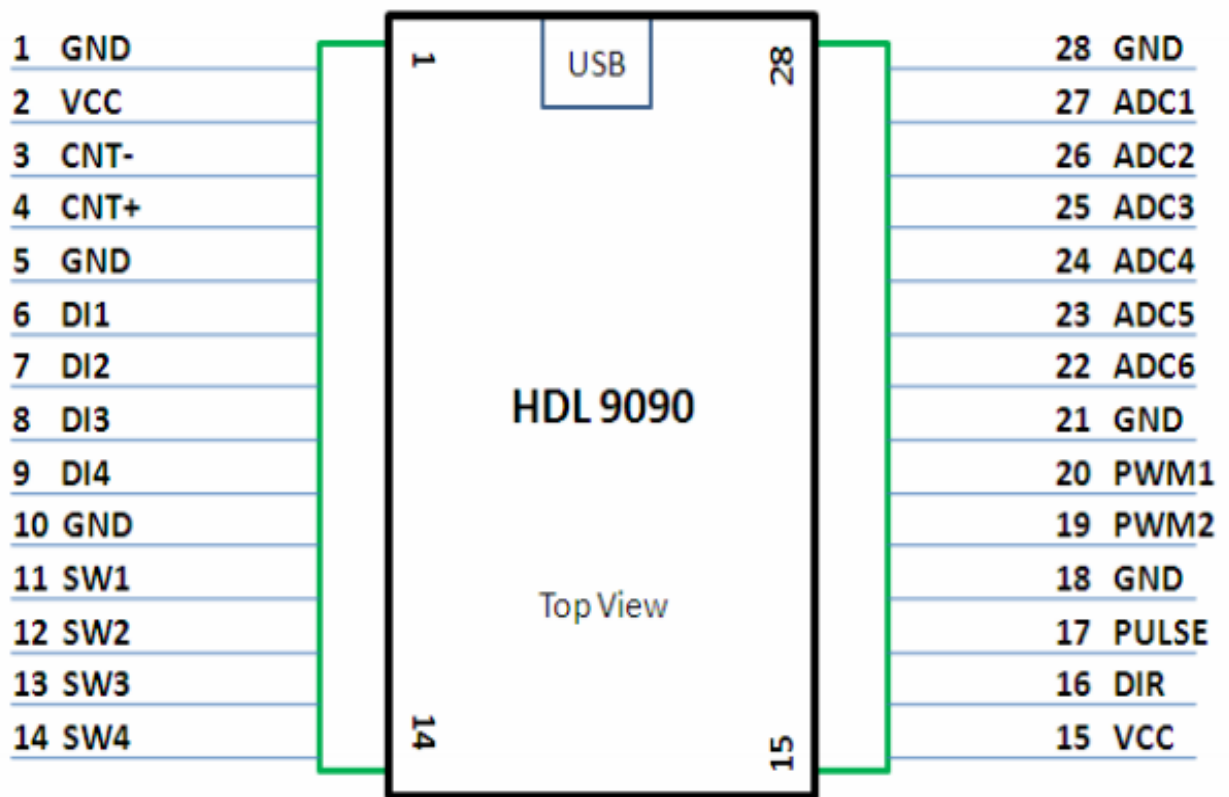
- Module công suất điều khiển trực tiếp nguồn điện cấp cho Điện trở nhiệt cấp nhiệt độ cho lò nhiệt

- LM35: sensor đo nhiệt độ và truyền tín hiệu tới ADC của Card USB 9090.

3.2: Giới thiệu Card USB-9090

Card USB HDL 9090 là card thu thập dữ liệu và xuất tín hiệu điều khiển đa năng thế hệ tiếp theo của HDL 9001. Ngoài chức năng đã có trên HDL 9001 như thu thập dữ liệu từ các cảm biến, điều khiển ON/OFF, điều chế xung PWM v.v. HDL 9090 còn có chức năng Digital Input giúp bạn ứng dụng linh hoạt hơn. Với tính năng vượt trội như tốc độ cao ADC cao hơn 5 lần và chính xác hơn 4 lần so với HDL 9001, card USB 9090 là lựa chọn tối ưu cho ứng dụng thu thập dữ liệu vào điều khiển

Mô tả các chân tín hiệu của card USB 9090:

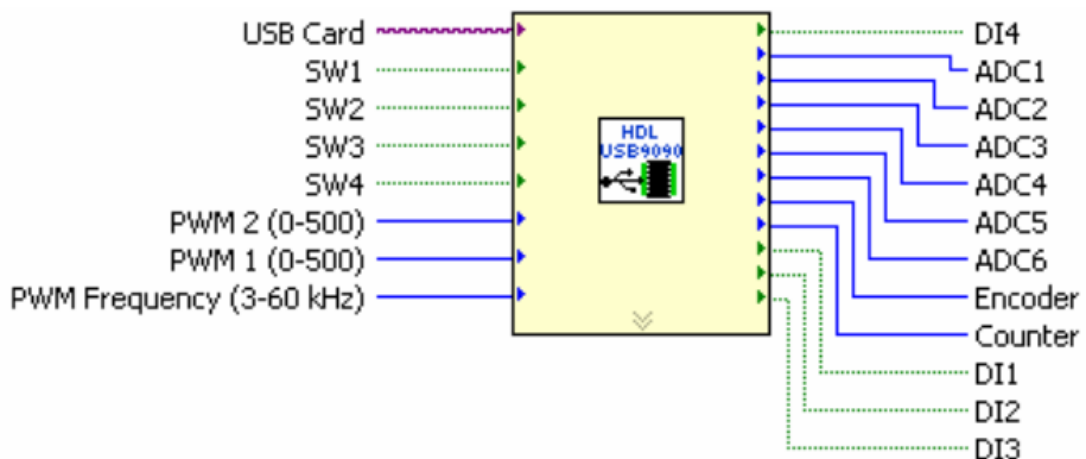


Hình 3.4: Các chân tín hiệu Card USB 9090

Ký hiệu	Mô tả	Loại
GND	Ground- Chân mát	Nguồn
VCC	Nguồn 5V lấy từ USB	Nguồn
CNT+/CNT	Chân đếm Counter	Input
DI1-DI4	Tín hiệu vào dạng số 0-5V	Input
SW1-SW4	Tín hiệu ra dạng số 0-5V	Output
DIR	Set bộ đếm xung đếm lên(5V) hay đếm xuống(0V)	Input
PULSE	Đếm xung tín hiệu 0-5V	Input
PWM1-PWM2	Xuất tín hiệu PWM để điều khiển	Output
ADC1-ADC6	Nhận tín hiệu vào tương tự (Analog) 0-5V	Input

Phần mềm đi kèm:

- NI VISA phiên bản 3.1 hoặc cao hơn: đây là phần mềm công ty NI, cần cài phần mềm trước khi muốn giao tiếp LabVIEW (PC) với bất kì thiết bị ngoại vi nào.
- Thực hiện các bước kết nối thiết bị HDL USB 9090 vào máy
- Hàm HDL 9090: để truyền nhận tín hiệu từ LabVIEW với card HDL USB 9090, sơ đồ chân (I/O) của hàm như sau:



Hình 3.5: Sơ đồ chân I/O

Mô tả các chân của hàm HDL USB 9090 để lập trình như bảng sau:

Chân	Loại	Mô tả
USB Card	Control	Tạo control tại chân này để chọn thiết bị USB 9090
SW1-SW4	Control	Nối giá trị Boolean (TRUE-FALSE) vào các chân này để phân cứng HDL 9090 xuất ra tín hiệu số tương ứng (TRUE 5V, FALSE 0V)
PWM1-PWM2	Control	Nối giá trị số nguyên 0-500 vào để phân cứng xuất xung PWM có duty cycle tương ứng là 0-100%
PWM Frequency (3-60kHz)	Control	Nối giá trị từ 3000-60000 vào để chọn tần số tín hiệu PWM card xuất ra
DI1-DI4	Indicator	Giá trị tín hiệu số đọc được từ chân DI1-DI4 trên card. Giá trị mặc định là True (5V)
ADC1-ADC6	Indicator	Giá trị ADC (0-1023) đọc được từ các chân ADC tương ứng trên card
Encoder	Indicator	Giá trị encoder 0-100000

Kết nối card HDL USB 9090 với máy tính:

Máy tính sử dụng để kết nối cần cài sẵn NI VISA phiên bản 3.1 hay cao hơn. Sau đó thực hiện các bước sau đây:

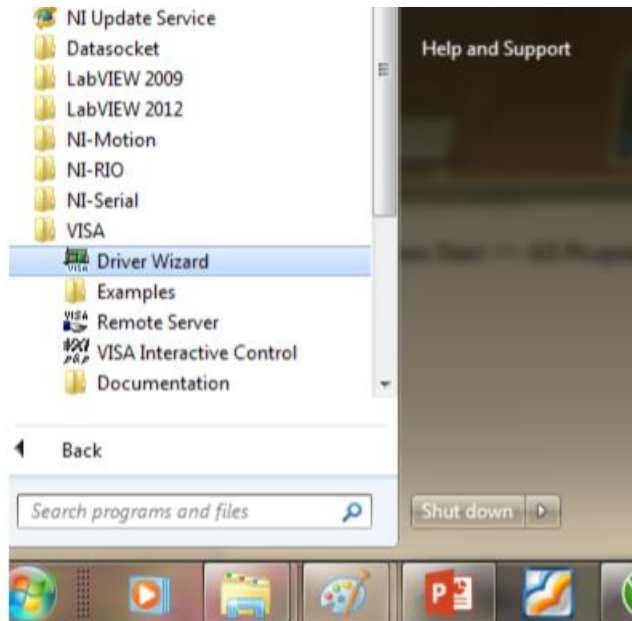
Bước 1:

Kết nối HDL USB 9090 vào máy tính bằng dây USB qua cổng giao tiếp USB



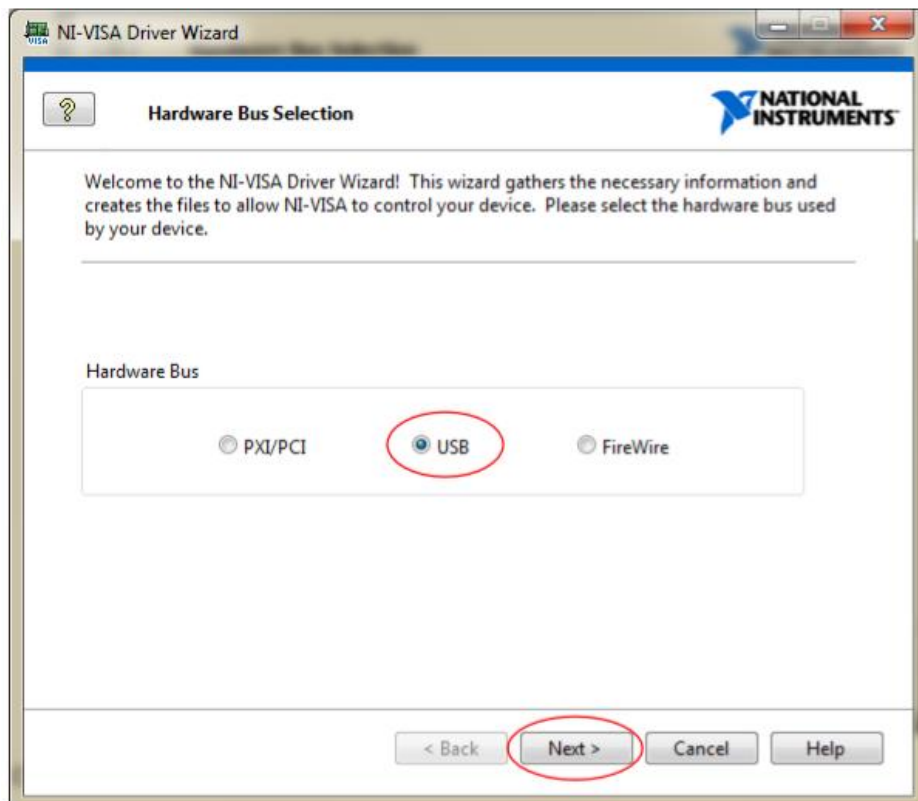
Bước 2:

Chọn Start > All Programs > National Instruments > VISA > Driver Wizard



Bước 3:

Chọn USB -> Next



Bước 4:

Nhập 9090 vào khung USB Manufacturer ID (Vendor ID)

Nhập 0001 vào khung USB Model Code (Product ID)

NI-VISA Driver Wizard

USB - Device Information

Welcome to the NI-VISA Driver Wizard for USB! This wizard gathers the necessary information to allow NI-VISA to control your USB device. Please enter the requested information about your device.

This wizard generates an INF file for use with Windows 2000/XP/Vista. The INF file tells the operating system to allow NI-VISA to control the USB device that you specify here.

USB Manufacturer ID (Vendor ID)
0x 9090

USB Model Code (Product ID)
0x 0001

Manufacturer Name
Device Manufacturer Name

Model Name
Device Model Name

Compound Device?

2 Number of Interfaces

NOTE: Using this wizard may not be necessary! If your device conforms to the USB Test & Measurement Class (USBTMC) protocol, NI-VISA can already detect and control it. Do not use this wizard to create an INF file.

< Back Next > Cancel Help

Sau khi nhập xong chọn Next

Bước 5:

Nhập HDL-9090 vào khung Instrument Prefix (thay cho chữ prefix có sẵn).

NI-VISA Driver Wizard

Output Files Generation

Thank you for using the NI-VISA Driver Wizard! This wizard will now generate the files which allow NI-VISA to control your device. Please specify the INF file name prefix and the directory in which to write the files.

If you are creating a distribution kit for your driver or application to be installed on other computers, read the instructions that are included in the header of the generated INF file.

Instrument Prefix (INF file name)
HDL-9090

Directory in which to save the generated files:
C:\Users\TOAN\Documents\National Instruments\NI-VISA\HDL-9090

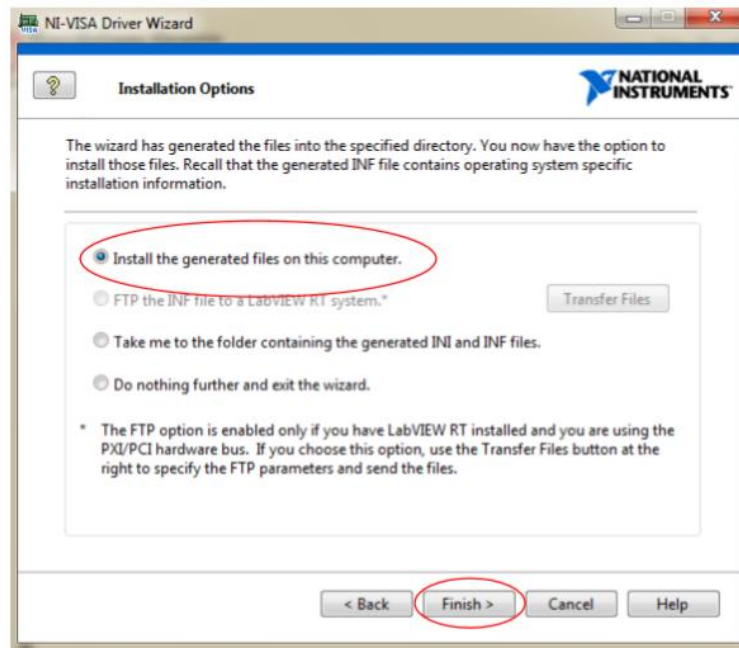
Files to be generated:
C:\Users\TOAN\Documents\National Instruments\NI-VISA\HDL-9090\HDL-9090.inf
C:\Users\TOAN\Documents\National Instruments\NI-VISA\HDL-9090\HDL-9090_vista.inf

< Back Next > Cancel Help

Sau khi nhập xong chọn Next

Bước 6:

Chọn mục Install the generated files on this computer

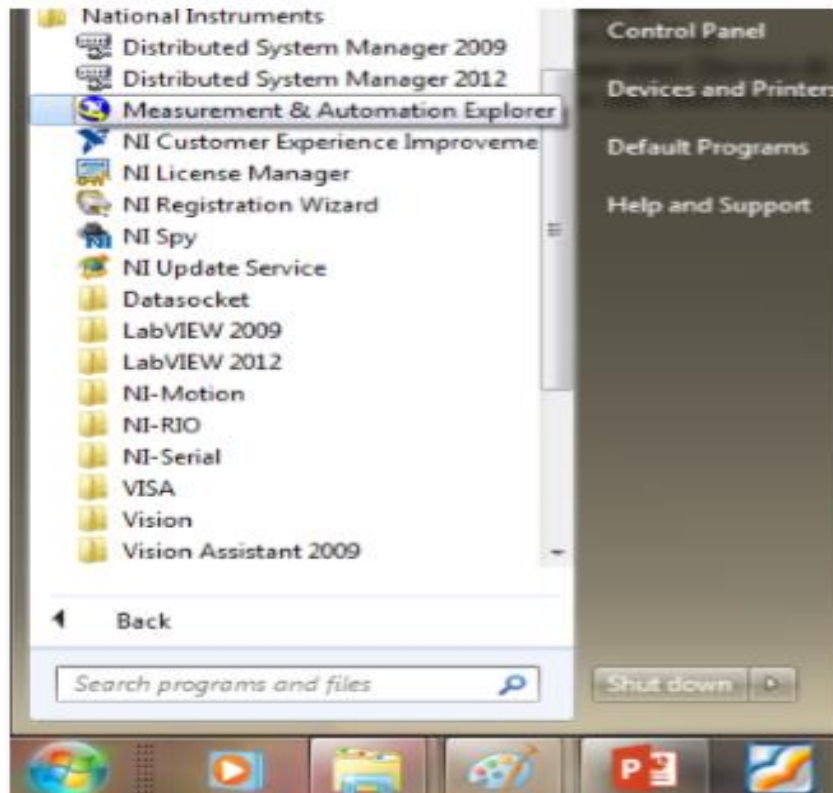


Chọn Finish

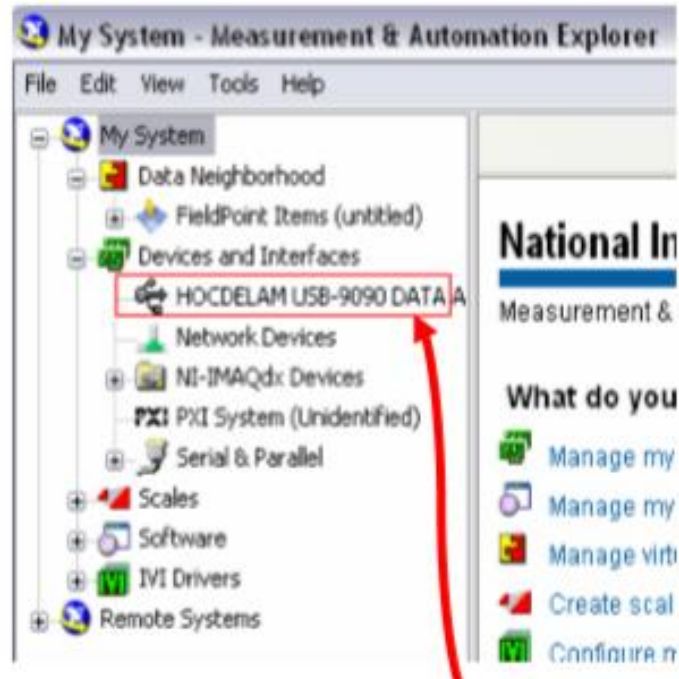
Lúc này card HDL 9090 sẽ được cài và nhận dạng trong NI MAX. Chờ một vài phút để quá trình hoàn tất.

Bước 7:

Mở phần mềm Measurement & Automation Explorer ra:



Chọn Device & Interface:



Lúc này thiết bị HDL USB 9090 đã được nhận biết như hình trên là việc cài đặt đã thành công.

3.3: Sensor LM35

Giới thiệu:

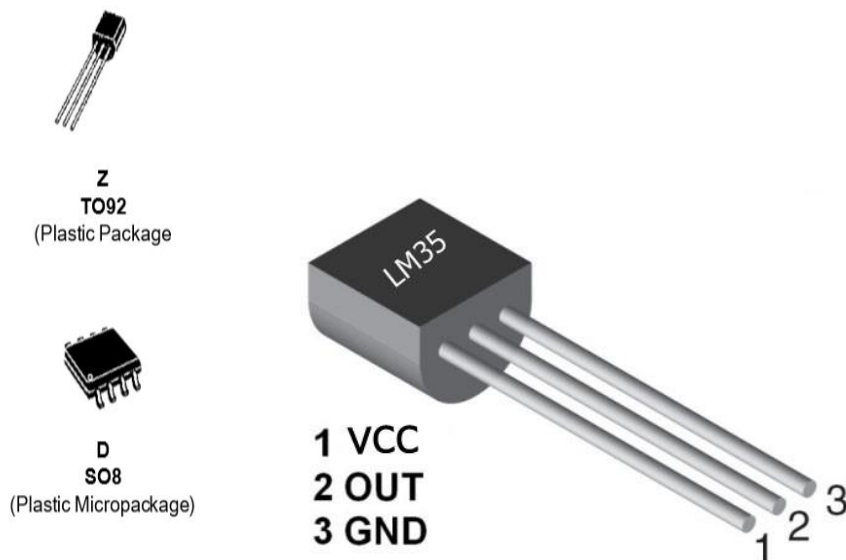
LM35 là cảm biến nhiệt độ chính xác dễ dàng hiệu chỉnh. Hoạt động như một zener 2 cực, LM35 có điện áp đánh thủng tỷ lệ thuận với nhiệt độ tuyệt đối ở 10mV/ °C. Với trở kháng hoạt động nhỏ hơn 1Ω, thiết bị hoạt động trong phạm vi hiện tại từ 400μA đến 5mA mà hầu như không có thay đổi nào về hiệu suất. Khi được hiệu chỉnh ở 25°C, LM35 thường có sai số nhỏ hơn 1°C trong phạm vi nhiệt độ 100°C. Không giống như các cảm biến khác, LM35 có đầu ra tuyến tính.

Các ứng dụng cho LM35 bao gồm hầu hết mọi loại cảm biến nhiệt độ trong phạm vi -40°C đến 125°C. Trở kháng thấp và đầu ra tuyến tính làm cho việc giao tiếp với mạch đọc hoặc mạch điều khiển đặc biệt dễ dàng.

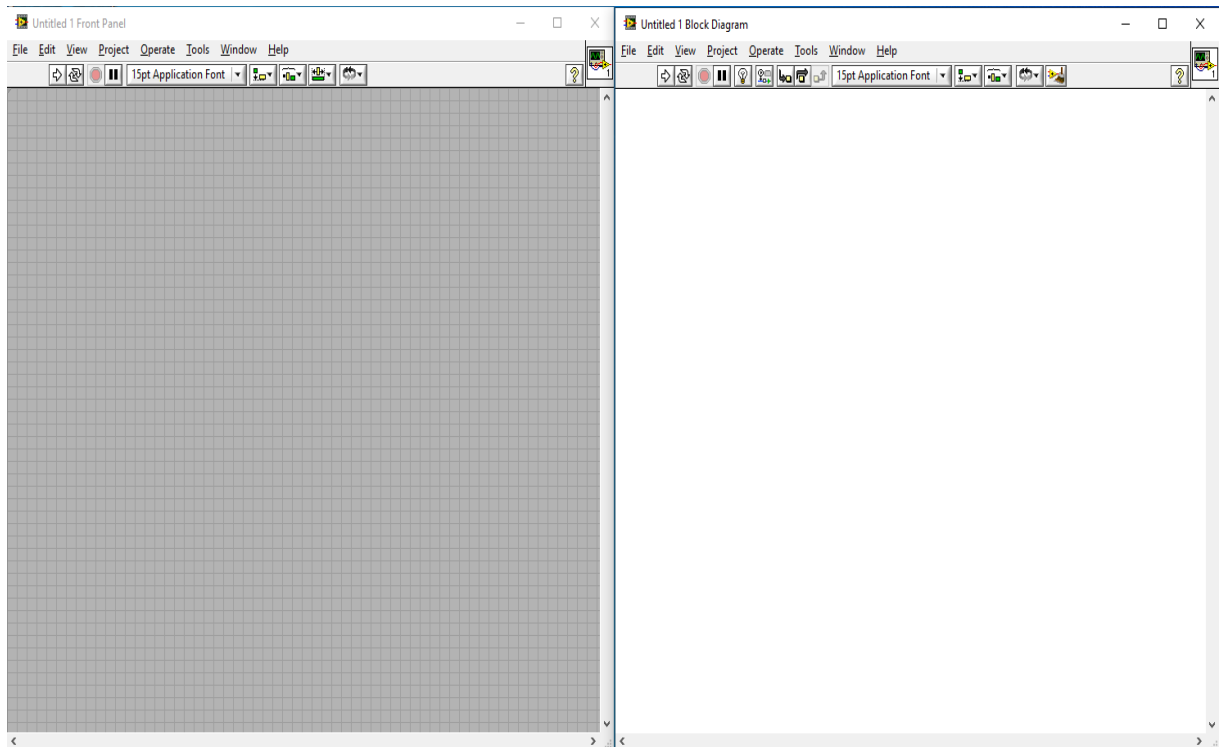
Tính năng:

- Độ chính xác +/- 1°C
- Hoạt động từ 400μA đến 5mA
- Trở kháng nhỏ hơn 1Ω
- Dễ dàng hiệu chuẩn
- Phạm vi nhiệt độ hoạt động rộng
- Giá thành rẻ

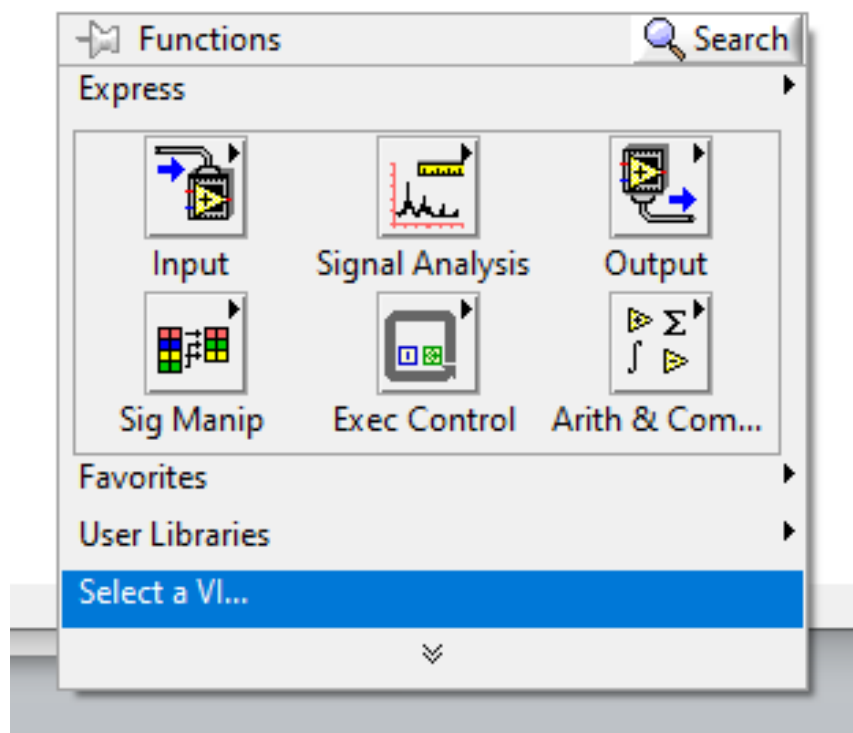
Kiểu chân và các kết nối



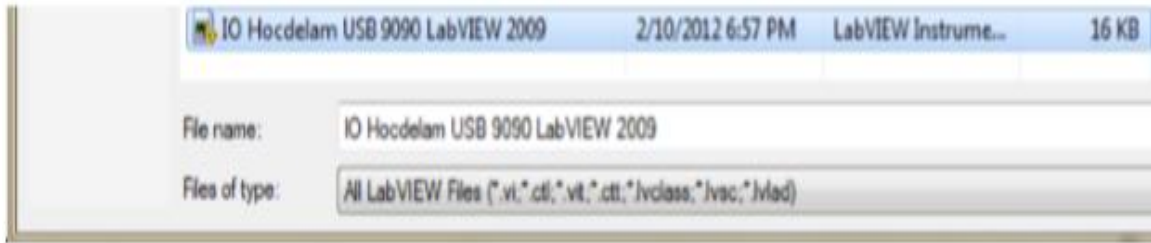
- Mở LabVIEW, mở mới một VI:



- Trên Block Diagram (BD), Right Click (RC) -> chọn Select a VI

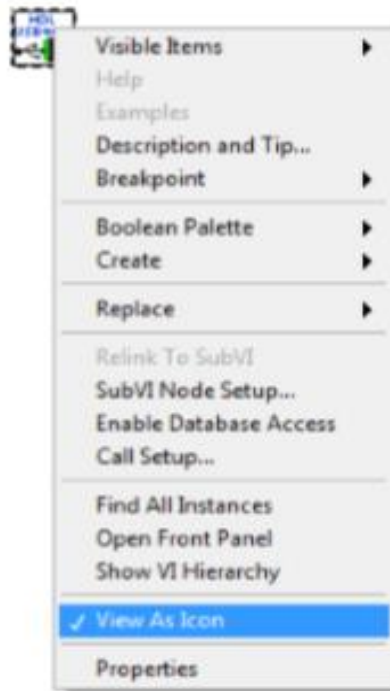


- Chọn IO Library Hocdelam USB 9090.VI



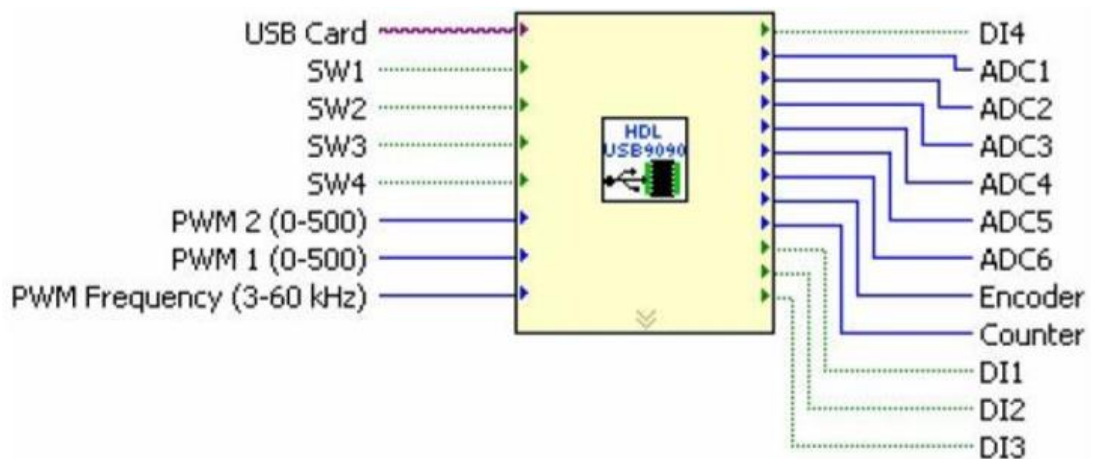
Chọn OK

- Right Click lên hàm HDL 9090 và bỏ chọn mục “View as Icon.”

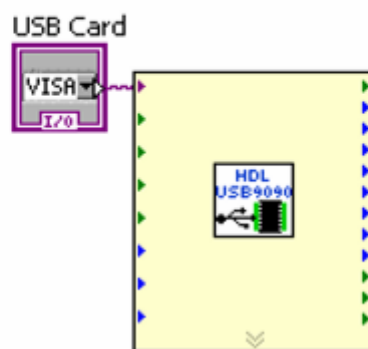
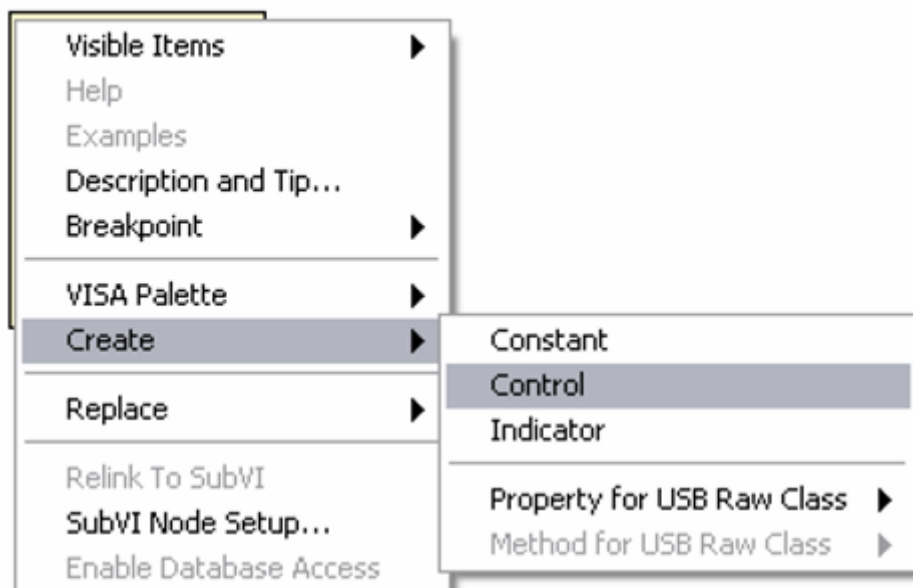


Lúc này hàm sẽ có các chân như sau:

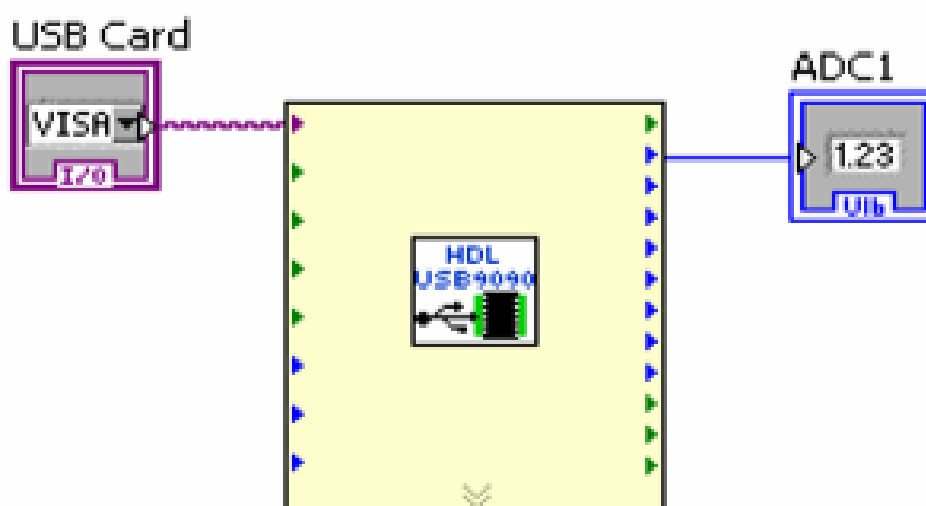
Khi rê chuột đến chân nào thì tên của chân đó sẽ hiện lên để nhận biết



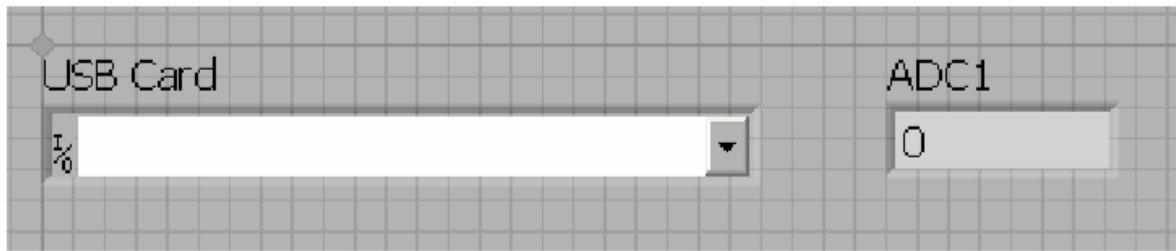
- Right Click lên hàm tại chân USB Card và chọn Create >> Control



- Right Click tại chân ADC1 và chọn Create >> Indicator

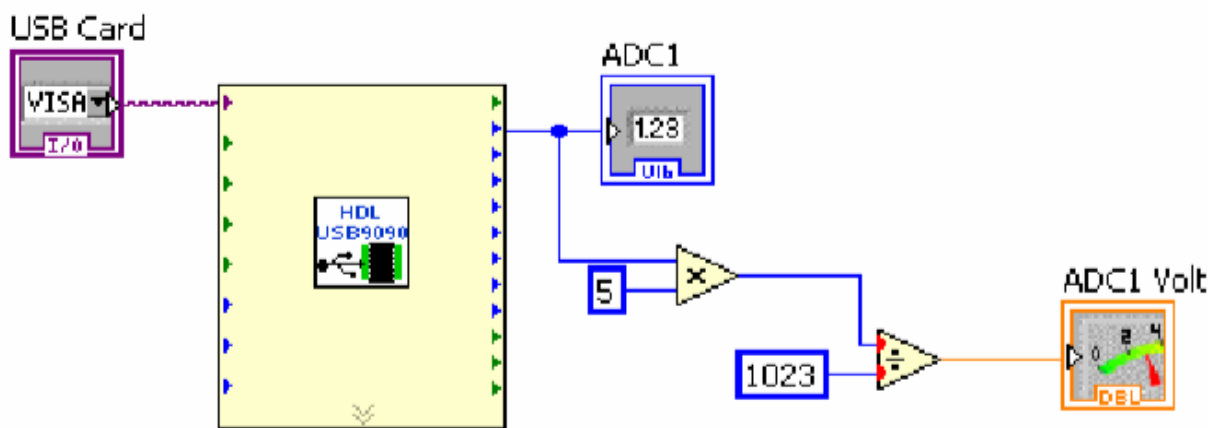


- Bên cửa sổ Front Panel sẽ hiện thị như hình bên dưới:

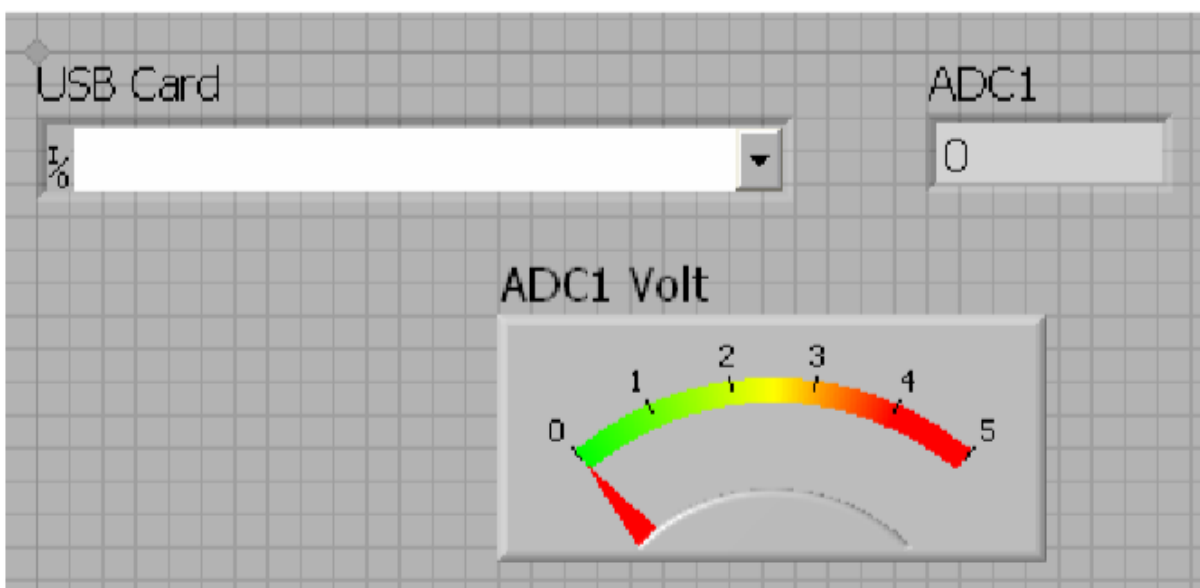


- Bây giờ có thể chọn thiết bị tại ô USB Card trên Front Panel và cho chương trình chạy. Giá trị nhận về sẽ trong khoảng 0-1023 vì ADC có độ phân giải 10 bit.

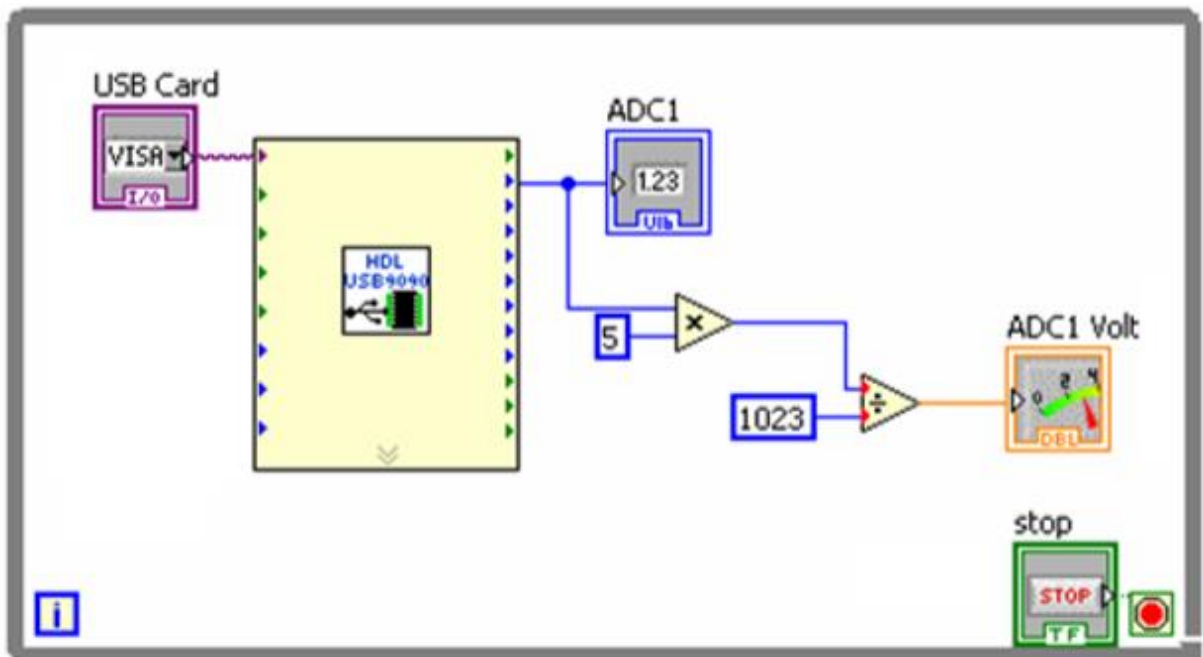
- Để biến giá trị ADC này thành điện thế 0-5V thì phải thực hiện phép toán (giá trị ADC $1 * 5 / 1023$) như sau:



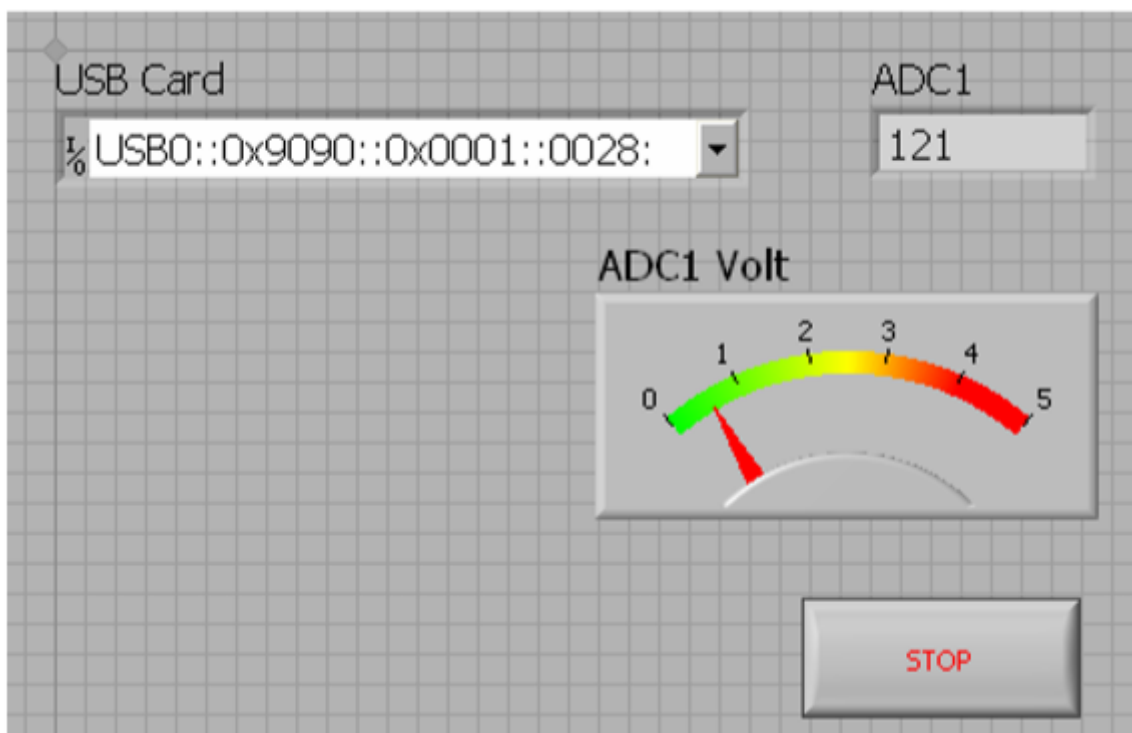
- Sau đó, để hiển thị giá trị điện thế Volt của ADC1, tạo một đồng hồ trên Front Panel, đặt tên là ADC1 Volt và nối như hình trên



- Cuối cùng để chương trình chạy liên tục thì phải dùng 1 vòng lặp trên Block Diagram



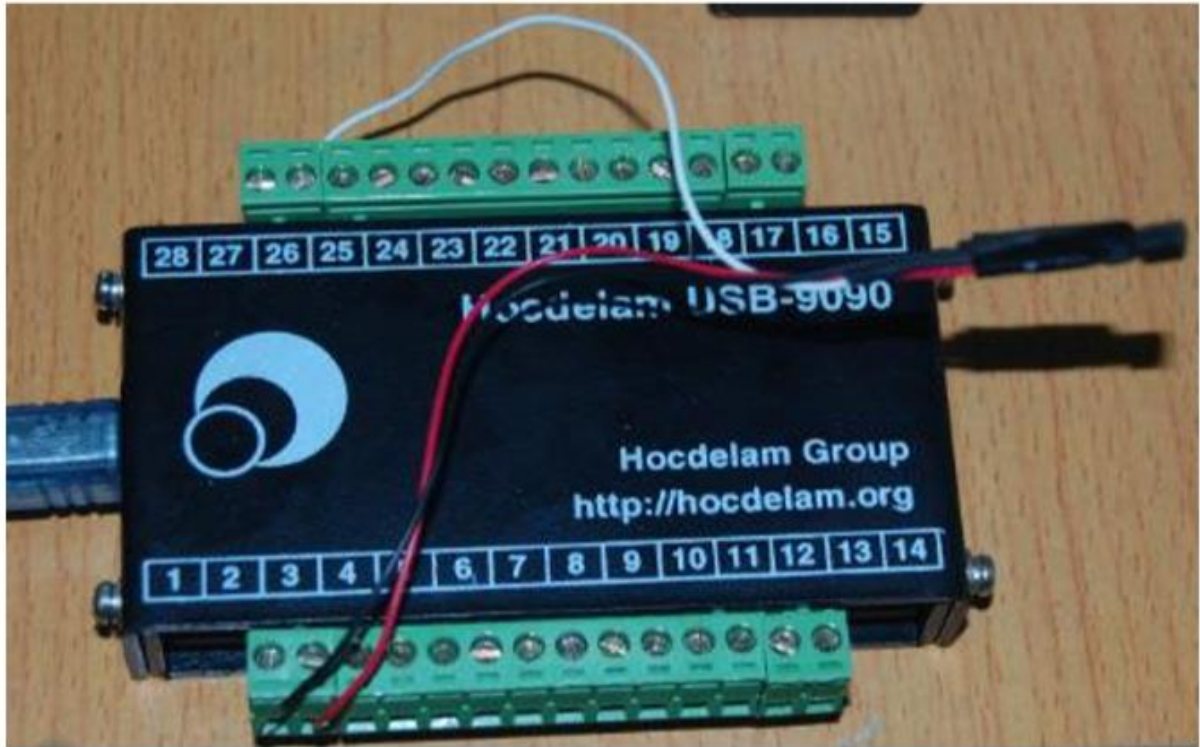
- Có thể chọn thiết bị USB và chạy thử chương trình để xem giá trị ADC1 Volt



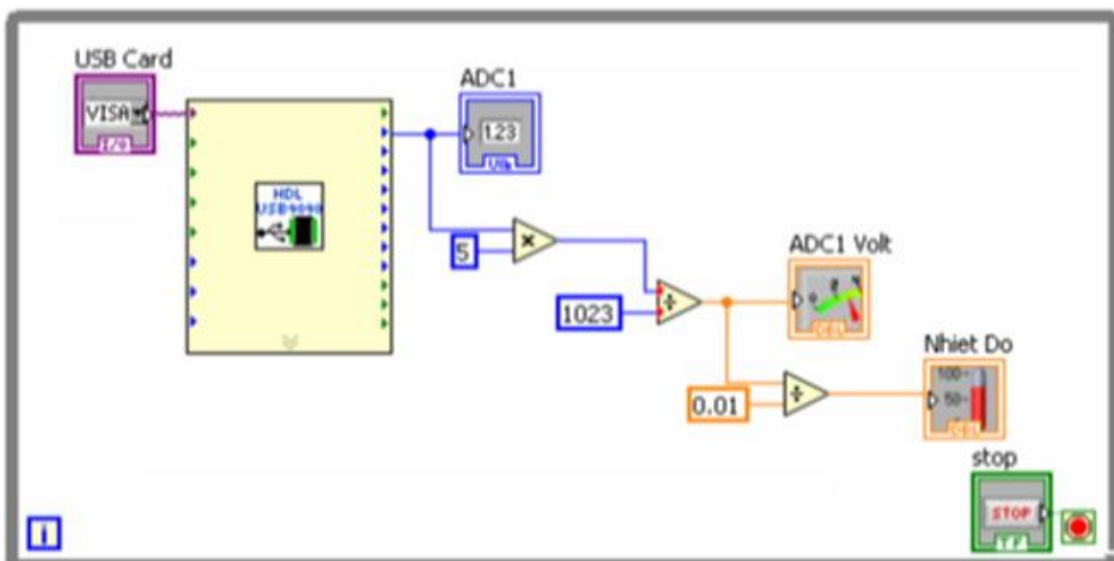
- Như vậy đã hoàn thành thu thập dữ liệu analog từ biến trở và biến đổi thành giá trị điện thế tương ứng.

b. Thu thập dữ liệu từ cảm biến nhiệt độ LM35

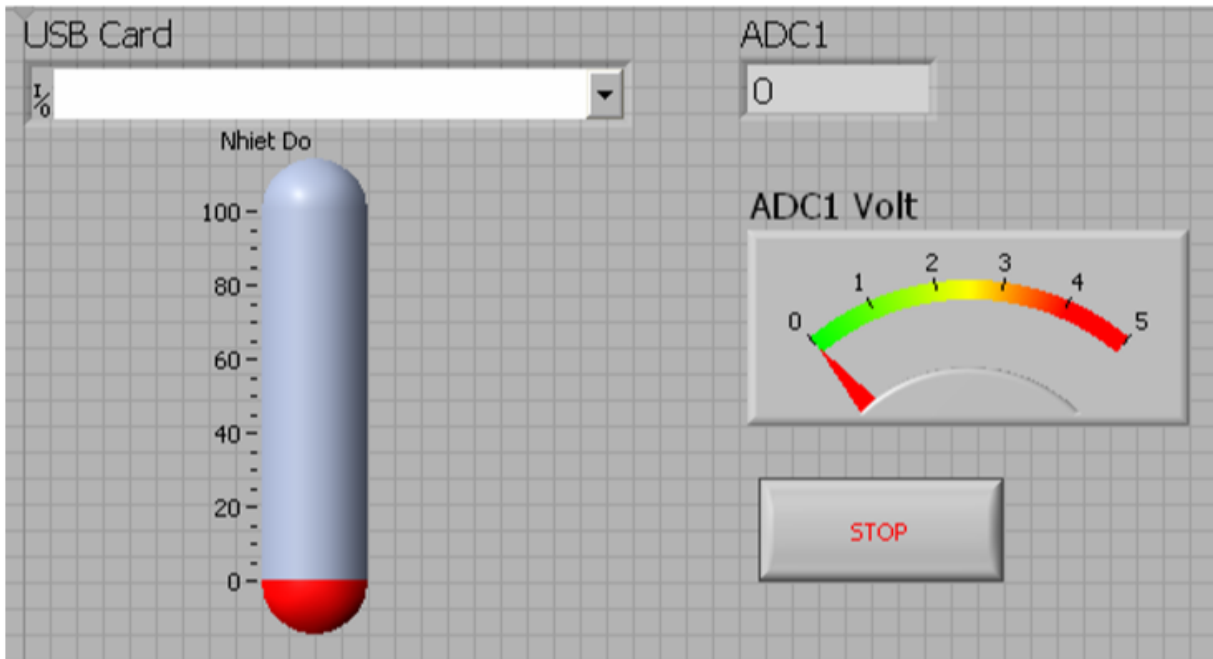
- Cảm biến nhiệt độ LM35 cũng có 3 dây đỏ, đen, trắng như biến trở. Do đó hoàn toàn có thể kết nối hoàn toàn giống với biến trở bên trên. (Đỏ - VCC, đen - GND, trắng - ADC1).



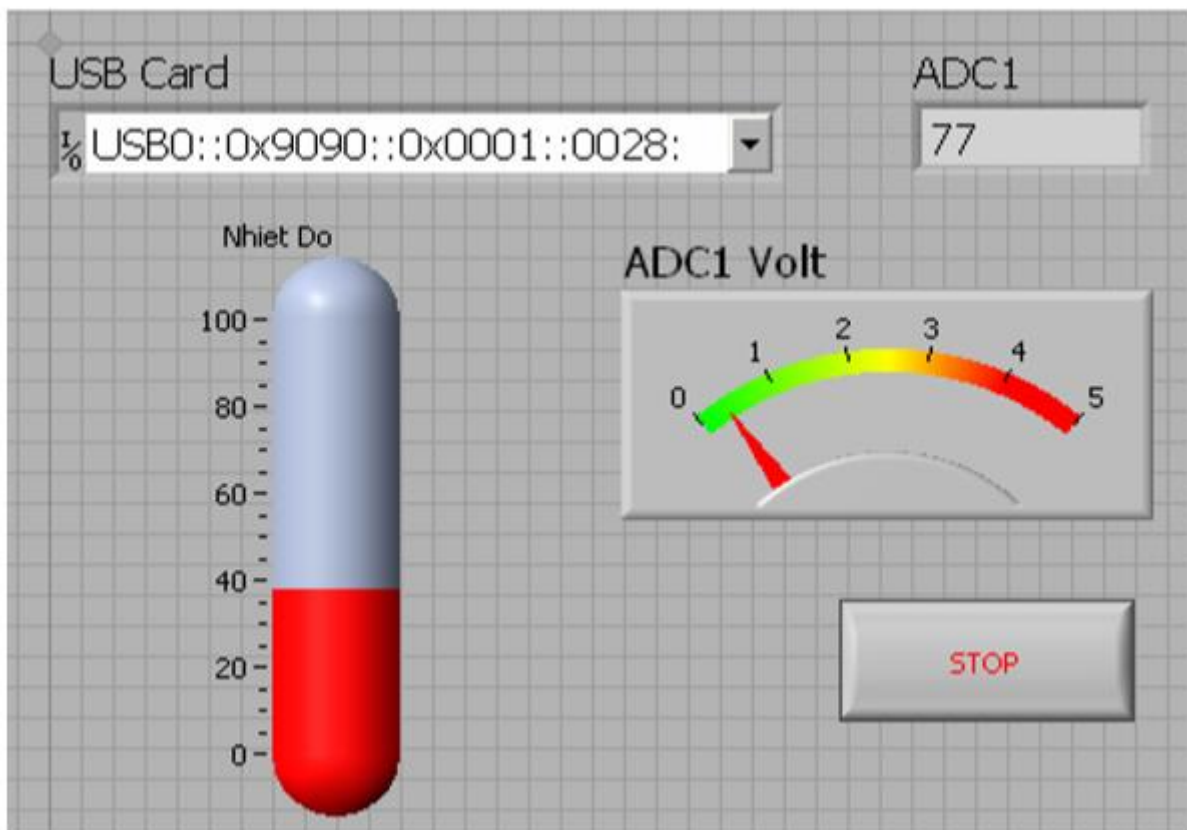
- Sau đó sẽ viết code trên Block Diagram (BD) tương tự phần trên. Vấn đề nhỏ ở đây là làm sao biến đổi được giá trị hiệu điện thế ADC1 Volt thành giá trị nhiệt độ.
- Để thực hiện biến đổi này thì ta cần xem thông tin của nhà sản xuất cảm biến. Cảm biến LM35 có thông số quy đổi là 10mV/ °C. Nên ta sẽ thêm một phép tính nữa là (nhiệt độ = ADC1 Volt/0.01 như hình dưới đây:



- Sau đó quay lại Front Panel và tạo 1 Indicator nhiệt kế để thể hiện nhiệt độ:

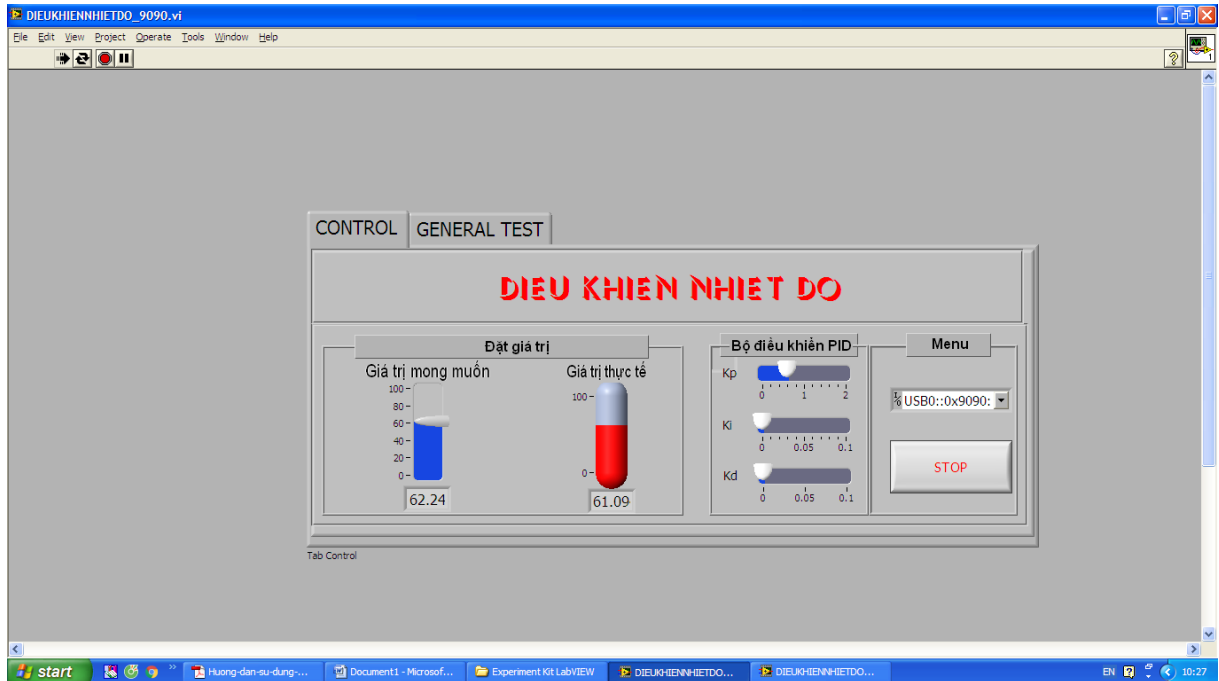


- Chọn thiết bị HDL 9090 trong ô USB Card và chạy chương trình ta sẽ nhận được nhiệt độ:

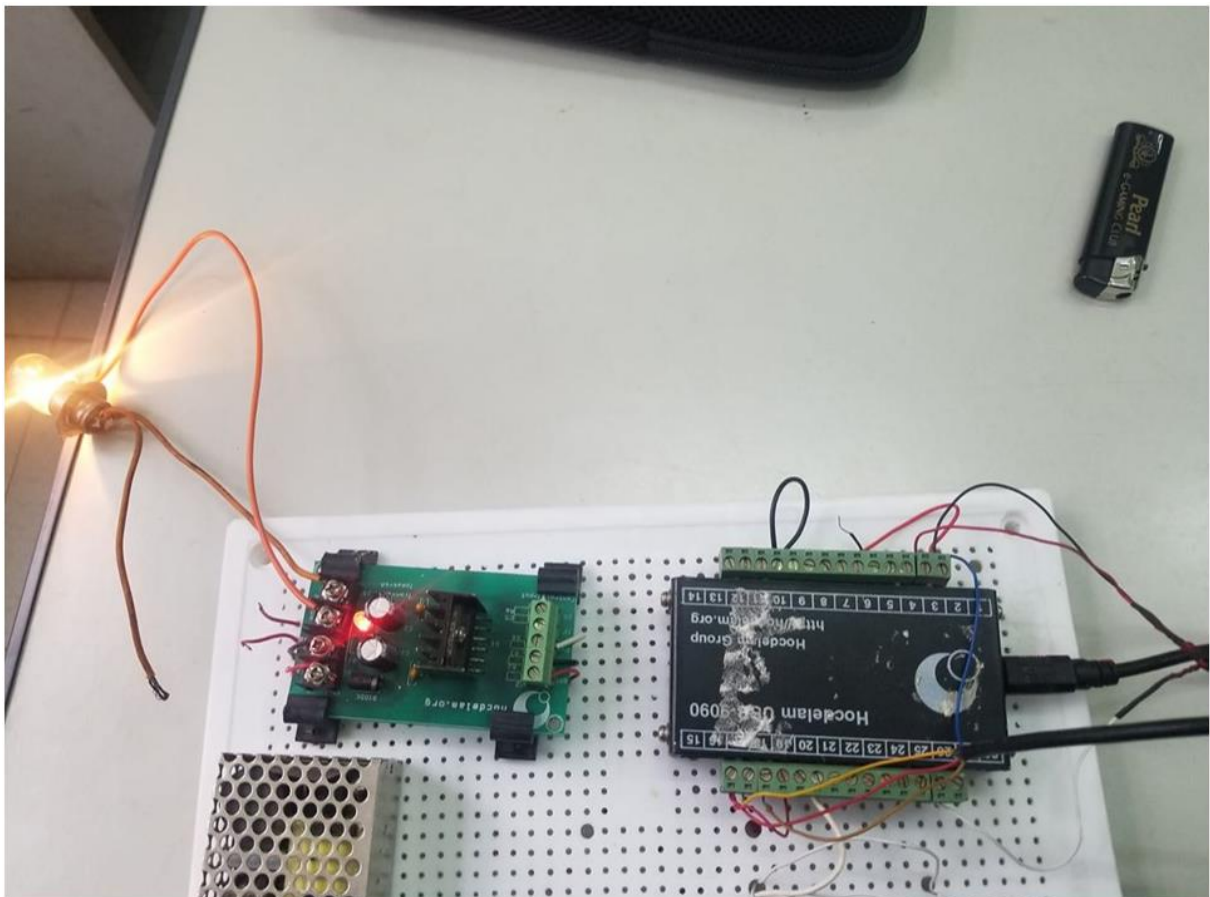


Như vậy đã hoàn thành đọc tín hiệu nhiệt độ cảm biến nhiệt độ LM35 sử dụng card USB HDL 9090

3.4.2: Chương trình hoàn chỉnh



Giao diện phần mềm điều khiển nhiệt độ lò hoàn chỉnh



Mô phỏng lò

Hoạt động của lò như sau:

- Ta mở phần mềm, nhấn chọn Start và thiết lập nhiệt độ đặt (ví dụ ở đây là 60 °C).
- Do nhiệt độ trong phòng đo được bằng cảm biến LM35 nhỏ hơn nhiệt độ đặt nên bóng đèn (giả lập là điện trở nhiệt) được cấp điện.
- Để mô phỏng lò nhiệt được ra nhiệt đến nhiệt độ 60 °C ta làm nóng cảm biến LM35 đến nhiệt độ đặt.
- Khi đó bóng đèn sẽ từ từ tối dần và tắt hẳn khi LM35 báo nhiệt độ ≥ 60 °C.
- Khi lò nguội dần nhiệt độ giảm dần thì bóng đèn sáng dần.

Chương 4: KẾT LUẬN

Các kết quả đã đạt được trong đề tài:

- Tìm hiểu về ngôn ngữ lập trình LABVIEW, giao diện, công cụ, các bảng điều khiển và chức năng từng bảng, hoạt động và cấu trúc của vòng lặp,...
- Tìm hiểu về lò nhiệt điện trở: nguyên lí và những cách điều khiển nhiệt và ổn định nhiệt độ lò nhiệt điện trở
- Thiết kế và hoàn thiện giao diện (The Front Panel) và sơ đồ khối (The Block Diagram) trên LABVIEW để điều khiển và ổn định nhiệt độ lò nhiệt điện trở.
- Thiết kế và thi công mạch điều khiển nhiệt độ.

Những hạn chế:

- Mô hình đã đảm bảo được yêu cầu đề tài nhưng trong thực tế, nhiệt độ của lò nhiệt còn chịu ảnh hưởng của nhiều yếu tố ngoại vi khác như nhiệt độ môi trường, độ ẩm không khí, độ giữ nhiệt của lò, ...
- Mới tận dụng được 1 trong nhiều ứng dụng của phần mềm LabVIEW đối với hệ thống điều khiển tự động.
- Giao thức kết nối của LabVIEW mới chỉ tìm hiểu thông qua card USB

Kết luận:

Do thời gian nghiên cứu có hạn nhưng nhờ sự chỉ bảo tận tình và hỗ trợ của thầy giáo hướng dẫn Ths. Ngô Quang Vĩ em đã hoàn thành được đề tài “Nghiên cứu điều khiển nhiệt độ lò nhiệt điện trở bằng phần mềm LabVIEW” lần này.

Em đã cố gắng hoàn thành đề án hết sức mình nhưng do kinh nghiệm cũng như kiến thức của bản thân còn nhiều hạn chế nên không để tránh khỏi những sai sót trong quá trình làm. Chính vì vậy em mong muốn nhận được sự chỉ bảo của các quý thầy cô và các bạn để giúp em hoàn thiện đề án của mình một cách chính chu nhất.

Với kiến thức chuyên môn về phần mềm, lý thuyết về công nghệ và kinh nghiệm thực tế còn nhiều thiếu sót và hạn chế nên em rất mong muốn có được và sẵn sàng đón nhận mọi sự chỉ bảo, đánh giá và góp ý của thầy cô và các bạn nhằm hoàn thiện tốt đề án lần này.

Em xin trân thành cảm ơn!

Tài liệu tham khảo

1. Lập trình LabVIEW - TS Nguyễn Bá Hải
2. Getting Stared With LabVIEW – National Instruments
3. Hướng dẫn sử dụng card HDL USB 9090 - Hocdelam.org
4. LabVIEW Fundamentals – National Instruments