Automation with LabVIEW

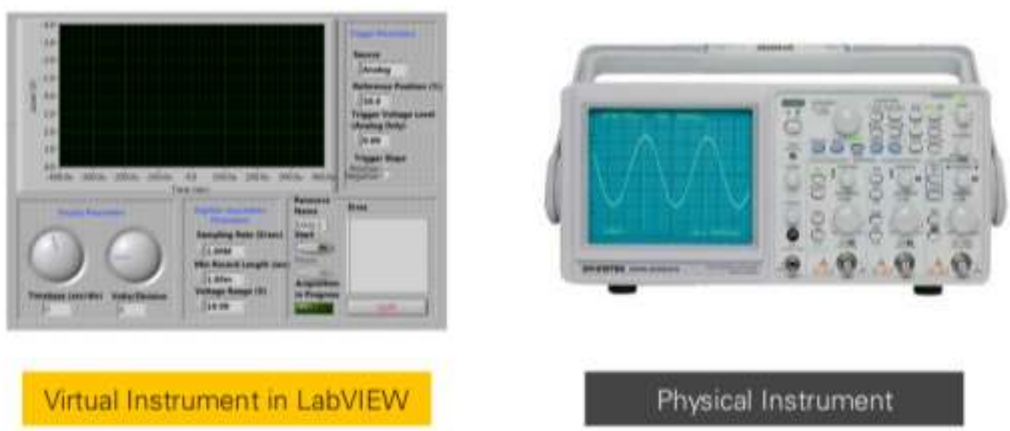
CBEMS 140B
Winter 2017

UC Irvine
Department of Chemical Engineering and Materials Science

Introduction

Most of our experiment involves data measurement followed by an action item corresponding to the data recorded. For example, in the heat exchanger experiment at 140A, when the heater is turned on, the sensor tells the heating element that the water is at room temperature and the element starts heating. Once the water temperature in the reservoir reached the set temperature, the heater stops providing heat. On the other hand, when the operator open the water tap to provide cooling stream to the heat exchanger, once the outlet shows sign of a moderate water stream, the operator stops turning the tap furthermore. In our terminology, this serial of event is part of process control. Sometimes, it is automated, like the coordination of the temperature sensor and the heating element. In old fashion operation, it is the human who makes the decision, like opening the tap and stop at a desired position.

LabVIEW is an alternative tool to help automate a process control event. LabVIEW stands for Laboratory Virtual Instrument Engineering Workbench. It is developed by National Instruments and was first released in 1986. It is capable of developing sophisticated measurement, task and control applications. LabVIEW integrate hardware and software and acts as a virtual instrument because their appearance and operations imitate physical instruments*:



In this lab, general basic of LabVIEW will be introduced and a computer with a LabVIEW licensed software will be provided as a self-study learning tool. The ultimate objective is to introduce the application of LabVIEW and generate interest for further learning of this software. Please keep in mind that LabVIEW is essentially a graphical computer language (G programming). In order to write a functional code, a programmer needs to understand LabVIEW syntax and its memory management, which are out of the scope of this laboratory exercise.

Experiment A

Objective

- To lean the basic components of LabVIEW.

Background

A LabVIEW project consists of the “Front panel” and the “Block diagram”. The front panel with control and indicator defines the user interface. The block diagram defines the code.

Procedure

Perform Exercise 1-2 Concept: Locating Controls, Functions and Vis in LabVEIW Core I Exercise Guide

Experiment B

Objective

- To meet LabVIEW as a tool to design user friendly control panel.

Background

LabVIEW integrate hardware and software and acts as a virtual instrument. The program’s appearance and operations imitate physical instruments and it gives the programmer flexibility to design the control panel graphical presentation.

Procedure

Perform Exercise 2-1 Selecting a Tool in LabVEIW Core I Exercise Guide

Experiment C

Objective

- To meet LabVIEW as a computer language

Background

The programming language used in LabVIEW, named G, is a dataflow programming language (compared to control flow model in C++ and most other text-based languages). Execution is determined by the structure of a graphical block diagram (the LabVIEW-source code) on which the programmer connects different function-nodes by drawing wires. These wires propagate variables and any node can execute as soon as all its input data become available.

Procedure

1. Perform Tutorial 1:LabVIEW Basics printed at:

http://k12lab-support-pages.s3.amazonaws.com/lvbasichome1.html

2. Perform Exercise 4-1 Pass Data Through Tunnels in LabVEIW Core I Exercise Guide

Experiment D

Objective

- To meet LabVIEW as a tool to acquires, analyzes and displays data using a National Instrument hardware

Procedure

Perform Exercise 2-2 Simple AAV VI in LabVEIW Core I Exercise Guide

Experiment E

Objective

- To meet LabVIEW supplemental software to access hardware

Background

In order to lower the requirement of programming, National Instrument developed a program called NI-Max (Measurement & Automation Explorer) to help user access to the hardware made by National Instrument and also those from other manufacturers. With NI MAX, you can:

- Configure your National Instruments hardware and software
- Back up or replicate configuration data
- Create and edit channels, tasks, interfaces, scales, and virtual instruments
- Execute system diagnostics and run test panels
- View devices and instruments connected to your system
- Update your National Instruments software.

Procedure

1. Open the NI-MAX (its icon is pinned to the Start menu).
2. Expand the “Devices and Interfaces” tab. It shows a list of device currently recognized by NI Max.
3. Connect the Extech manometer to the computer using the RS-232 USB cable. The USB driver is pre-installed by the instructor. Turn on the manometer.
4. Click on “View” on the top panel and select “Refresh” (The last option).
5. The new addition on the list should be the Extech manometer. Double click on the manometer tab to view the settings, which include the port binding, status, resource name and port settings.
6. Click on the “Open VISA Test Panel” and navigate other configuration options (actual configurations required for the Extech manometer is out of scope of this lab)
7. Open a blank VI.
8. Right click on the Block Diagram white area. Select “Instrument I/O → VISA → VISA Read” and add to your block diagram. Hover over to the top left of the VISA Read icon until you see “VISA Resource name”. Right click and select “Create→Control”. Now a “VISA I/O” icon is connected to the “VISA Read” icon.
9. Switch to the Front Panel.
10. A “VISA resource name” selective box should be present.
11. Click on the down arrow, the port corresponding to the Extech manometer should be on the list.
12. End of exercise.

Experiment F

Objective

- To practice acquire and outputting signal using NI myDAQ

Background

National Instrument has made a lot of hardware to help acquiring and outputting signal using LabVIEW as the interface. NI myDAQ is one of the simplest models which is capable of capturing analog and digital data and also generating low voltage signal.

Procedure

1. Connect myDAQ to the computer using its USB cable.
2. Open a blank VI
3. Add a DAQ assistant in the block diagram. (Right click on the white area, select "Measurement I/O→DAQmx-Data Acquisition(top left)→DAQ Assistant (bottom left))" and add to your block diagram.
4. Double click on the DAQ assistant icon and set up a VI to acquire analog temperature signal from a thermocouple using myDAQ continuously.
5. Attach a thermocouple to myDAQ based on the channel you selected in step 4.
6. Add a "graph form indicator" to the data output of the DAQ assistant
7. Run the VI and observe the signal presented in your "graph form indicator".
8. Stop the VI.
9. Open a new blank VI
10. Add a DAQ assistant in the block diagram
11. Double click on the DAQ assistant icon and set up a VI to generate analog signal in form of voltage continuously.
12. Attach a LED diode of your choice (either green or yellow) to the channel you selected in step 11.
13. Change the maximum and minimum voltage based on the specification of the diode (The LED diode needs AC voltage.)
14. Run the VI and observe when the diode is lit.
15. Stop the VI.
16. Design a VI to show warning in the form of flashing LED diode when the temperature of the thermocouple exceed a set-point.