

EET 438B
Sequential Control and Data Acquisition
Laboratory 1
Introduction to the LabVIEW Programming Environment

Laboratory Learning Objectives

1. Develop skills to navigate in the LabVIEW development environment, create simple programs, run programs and save them for later use.
2. Use charts and graphs to visualize collected data and display program results.
3. Build a graphical user interface using LabVIEW Components.
4. Utilize FOR, WHILE, IF-THEN, CASE programming structures to create program logic.
5. Create and utilize shift registers to store variables between iterations in FOR and WHILE loops.
6. Identify and create data types and structures in LabVIEW.
7. Explain the basic principles of data flow programming.
8. Use basic mathematical and Boolean operators and programming structures to produce programs given a specified functionality.

Theoretical Background

Personal computers (PCs) and microcontroller technologies provide the means to monitor and control processes. These processes can be complex industrial systems, or as simple as a coffee pot. The advent of low-cost electronics and the pervasive adaptation of PCs make these universal tools ideal for collecting data, processing it and taking control actions based on the results. High level programming languages using the Data Flow programming concept and graphical interface development reduce the time development time required of these systems.

Automated data acquisition (DAQ) and control using PCs requires a combination of dedicated hardware and software tools to bring measurements into the PC and convert it to meaning output. The data can be simply displayed, processed and displayed, or used to control other devices. Figure 1 shows the key components and subsystems of a typical DAQ system. The diagram

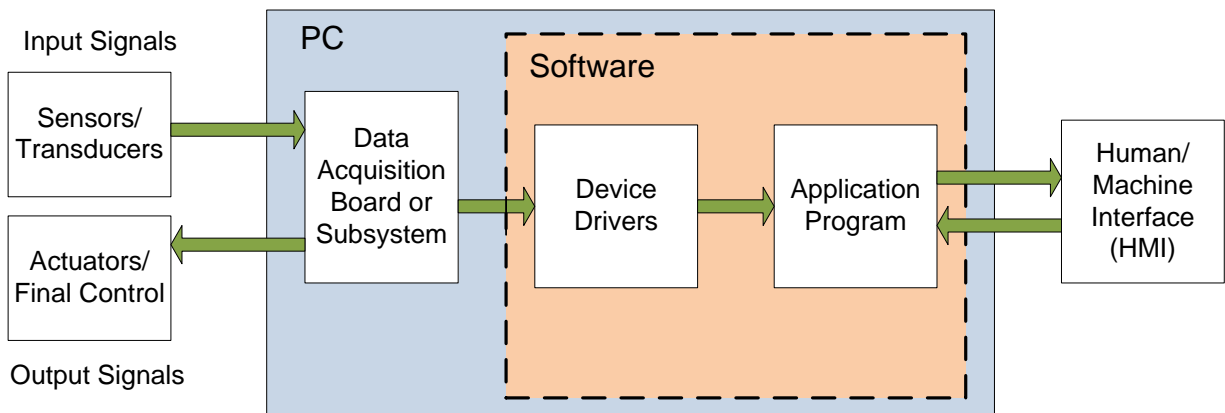


Figure 1. Block Diagram of a PC-based Data Acquisition and Control System.

shows the interconnection of the hardware and software of the systems. A DAQ system collects inputs from the physical environment using sensors and transducers. These devices can be analog, digital or a combination of both types. The digital sensors may be simple switches or switches designed to change state based on temperature, pressure, or position. The transducers take measurements of process or environmental variables and convert them into voltages or currents. Transducers continually monitor a variable and produce electric signals that relate to the measured quantity.

The DAQ system may also output both analog and digital values in response to data collected from the inputs. The digital signals implement on/off control of valves, solenoids, relays, and other electromechanical devices either directly or through an interface circuit. Analog outputs provide a continuous control signal drives a final control element responsible for controlling continuously monitored processes. These processes include, speed, temperature, flow and level control.

Input and output signal connect to a PC through appropriate wiring and a terminal box. The terminal box offers easy access to all the input/output signals of a data acquisition board (DAQ) of stand-alone subsystem. The DAQ board may be installed in an expansion slot of a PC chassis or be part of a separate system that still connects to a PC through a USB or similar interface. The DAQ board can have a number of analog and digital I/O points available for use. It may also have digital counter I/O that can measure frequency or output a pulse train.

The DAQ card samples the continuous signals and converts them to binary values in a process called analog-to-digital conversion (ADC or A/D). The card also provides an analog voltage output that can produce both an adjustable dc value of a simulated waveform. The digital I/O points are compatible with Transistor-to-Transistor Logic (TTL) standards. This means that the points can drive low voltage and current loads.

The DAQ hardware must communicate with the application program through software called device drivers. Device drives provide the necessary software routines for the PC operating system and the application program to pass data and commands. The National Instruments (NI) applications software is called LabVIEW and the NI hardware drivers are in the NI-Measurement and Automation Explorer (NI-MAX) software. DAQ projects that collect physical data and output control signals require the installation of both software packages. The NI-MAX not only provides drivers, it also includes a user interface for creating data collection channels and tasks. The software includes hardware testing routines with graphic output that make it useful in troubleshooting combined hardware and software projects.

LabVIEW software is a tool used for application software development. This programming method uses a block diagram approach to program, not the structured text of other languages like C or BASIC. This software allows rapid development of complex user interfaces and data collection processes. Developers can focus on construction of the overall system and data analysis, while the software environment handles many lower level operations. LabVIEW programs are called virtual instruments (VIs) because they can take the place of actual test instruments.

The laboratories in this course will initially focus on learning how to program using the LabVIEW environment. The labs will proceed to NI-MAX usage and the integration of both analog and digital I/O. The series of labs using LabVIEW will end with an integrated measurement and control application.

Procedure

- 1.) Acquire a copy of the student or community version of LabVIEW and the latest version of the NI-DAQmx driver package. Follow this link to download the community version of the software: <https://www.ni.com/en-us/shop/labview/select-edition/labview-community-edition.html>. Install all software on your own computer. LabVIEW is also available on computers in D30, and D122 on campus.
- 2.) Follow the following link to NI video training modules and view the content of the first four lessons <https://www.ni.com/getting-started/labview-basics/>. Note that each of the four parts have subsections. View all the subsections of each item. This site is a good reference for answering questions about how LabVIEW operates. It also has links to programming examples that are helpful in developing programs.
- 3.) Complete the programming assignments listed below and print both front panel and block diagrams to document your work.
- 4.) Write short (up to one page double-space) summaries on how each program operates. Explain how each part of the program works and what each symbol of the block diagram represents.

Programming Assignments

Construct the following LabVIEW Programs

- 1.) Create a Labview virtual instrument (VI) that takes the inputs from 4 numeric controls and displays the sum of all the inputs, and the average of all the inputs. Save this program on a disk so it can be demonstrated later.
- 2.) Create a VI that has three inputs. Two inputs will represent the setpoints of a window comparator and one input that represents the input to the comparator. The setpoints will be labeled upper trip point-UTP and lower trip point LTP. The window comparator implements the following logic.

$$\begin{aligned} X_{LTP} \leq X \leq X_{UTP}, Y &= \text{true} \\ X > X_{UTP}, Y &= \text{false} \\ X < X_{LTP}, Y &= \text{false} \end{aligned}$$

Where:

- X_{LTP} = lower trip point value (Numeric)
- X_{UTP} = upper trip point value (Numeric)
- Y = program output value (Boolean)
- X = program input value (Numeric)

Program the VI to simulate the logic of a window comparator that is given above. Use a binary indicator (LED) to show the overall output of the window comparator logic. Include inputs for both the UTP and LTP values. Save the program on a disk so it can be demonstrate later.

- 3.) Create a VI that has 8 toggle switches as inputs, a single digital indicator as an output, and 8 LEDs as binary outputs. The VI should simulate the operation of an 8-bit digital-to-analog converter and output a decimal value from 0 to 255 in the digital indicator for a given pattern of inputs. The digital to analog converter converts binary number values represented by the switch positions into decimal equivalents. The LEDs will be on when the corresponding switch is on. When the switches are in the "on" position, they represent a logic 1 and a logic 0 when in the off position. Save the program on a disk for later demonstration.

Lab 1 Assessment

Submit the following items for grading and perform the listed actions to complete this laboratory assignment.

- 1.) Complete the online quiz over LabVIEW.
- 2.) Printouts of the front and back panels of the LabVIEW programs.
- 3.) Written program descriptions as described in part 4 of the procedure.