



Certified Experts in Automation Engineering to Design, Control, Test & Adapt

## Air Handler Characterization Using AMCA Standard 210

### Author(s):

Gregory C. Cala, Ph.D., Partner & Chief Technology Officer, Data Science Automation, Inc.  
Benjamin A. Rayner, Senior Automation Systems Architect, Data Science Automation, Inc.

### NI Product(s) Used:

NI-LabVIEW 2013  
NI cDAQ – 9178  
NI-9220  
NI-9213  
NI-9263  
NI-9425  
NI-9476  
NI-9923

### Category:

Machine Control  
Oil, Gas, and Energy  
Structural and Physical Test and Monitoring

### The Challenge

Characterize air handlers using Air Movement and Control Association (AMCA) Standard 210 in a form that allows in-house staff to maintain the application.

### The Solution

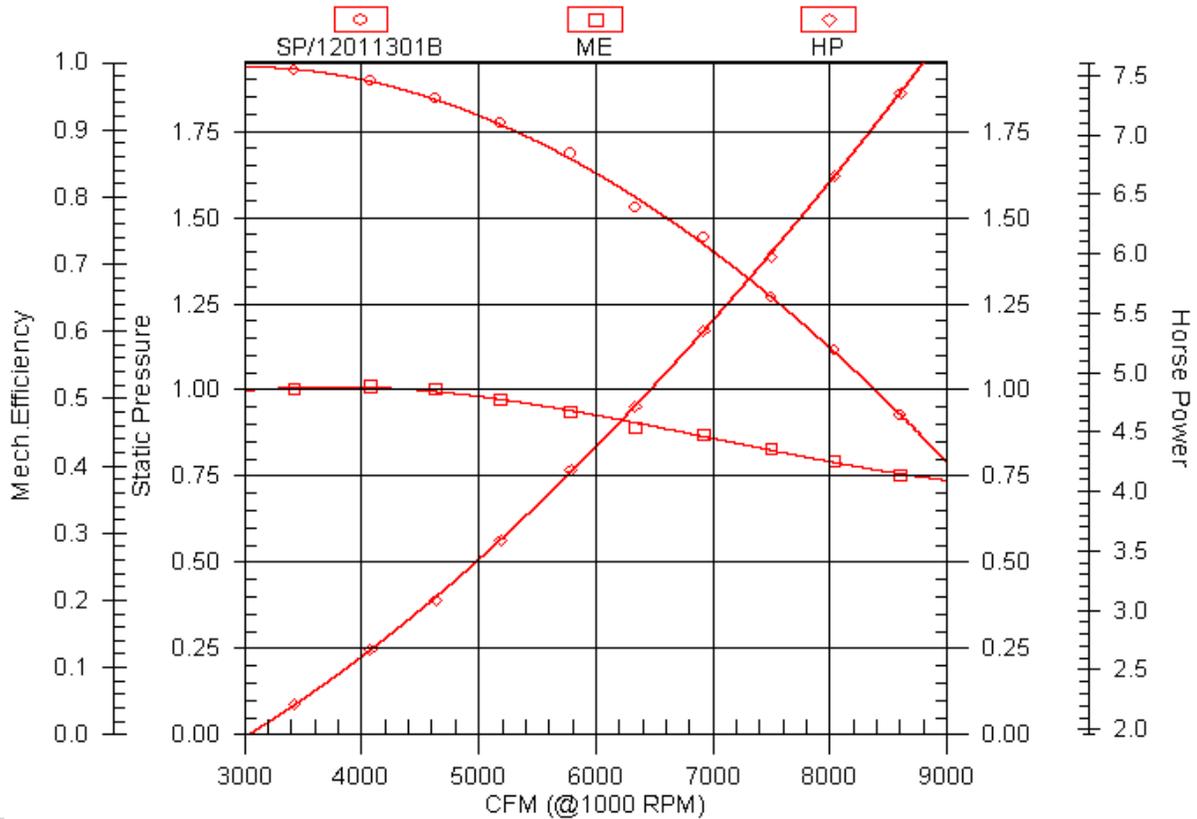
The Library feature of LabVIEW was utilized to design and implement an application that closely mimicked the physical system. In-house staff took over code changes within days of delivering the application.

### Introduction

Data Science Automation (DSA) is a premier National Instruments (NI) Alliance Partner that specializes in automating and educating the world leading companies. Clients choose DSA because of DSA's deep knowledge of National Instruments products, disciplined process of developing adaptive project solutions, staff of skilled Certified LabVIEW Architects and Certified Professional Instructors, and unique focus on empowerment through education and co-development.

### Background

Most people don't realize our world is full of blowers, fans and other air movers. And few of us have taken the time to understand what makes one air mover different than another. Even fewer have quantified those differences. This paper discusses a new control program written in LabVIEW and investigates how the nature of the test process guided the design and implementation by leveraging the use of LabVIEW Libraries.

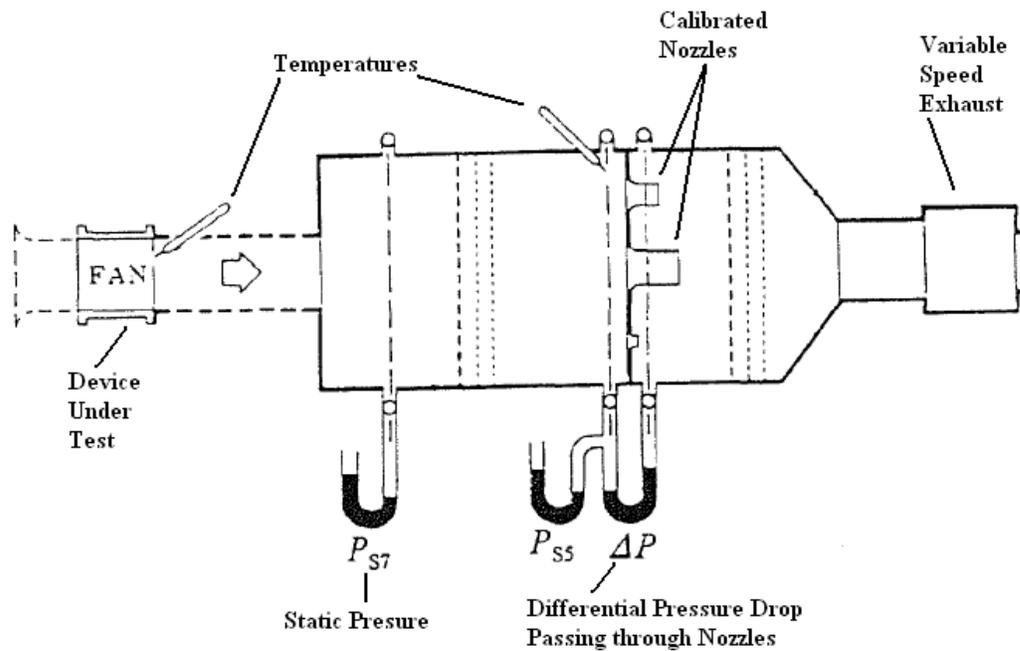


**Figure 1. Typical Air Handler Performance Curves**

Figure 1 shows a typical graph showing the performance of an air handling device. This standard plot of performance allows product designers to evaluate if an air handler will meet the needs of their application and perform efficiently. The Mechanical Efficiency and Horse Power are derived from measurements of the device under test. The airflow CFM (Cubic Feet per Minute) is calculated using the differential pressure measured across calibrated nozzles (Figure 2) in the wind tunnel itself (Figure 3). The Static Pressure is measured at the outlet of the air handler.

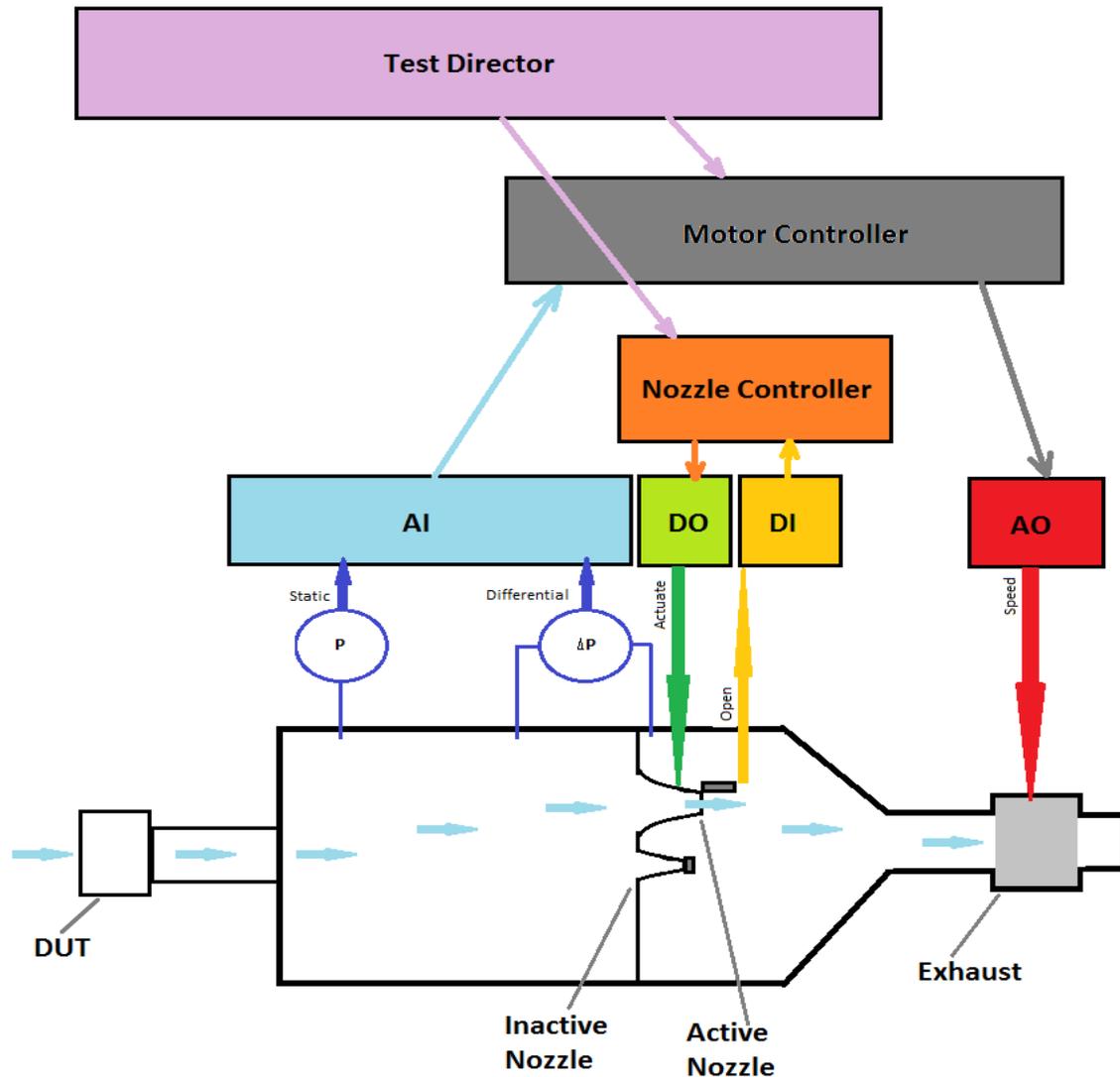


**Figure 2. Typical Calibrated Nozzles used to Determine Air Flow Rate (CFM)**



**Figure 3. Typical Wind Tunnel Schematic (page 41 of AMCA Standard 210)**

The basic process of characterizing the air handler involves adjusting the exhaust motor speed such that the DUT (Device Under Test) can be characterized under varying CFM rates. If while acquiring the lower CFM measurements either the Static Pressure or the Differential Pressure move outside acceptable limits, the nozzles are changed and adjustments made to adapt to the new nozzle combinations.



**Figure 4. Overlay of Software Design Components with the Physical Hardware**

#### A Cohesive Design

The application was implemented using a highly cohesive design where the software components for the most part are inspired by and associated with each of the Wind Tunnel sub-systems.

As illustrated in Figure 4 the wind tunnel hardware components suggested two of the design components included in the application, namely “Nozzle Controller” and “Motor Controller”. Since these components must act in a coordinated manner with each other, an additional component was implemented which acts to direct the operation of the various device controllers. This component was called the “Test Director” since it directed the actions of the “Controllers” and unified their operation in a manner similar to the role played by a human test director using analog measurement devices.

If we now turn our attention to the demands of the physical control and monitoring of the wind tunnel components we recognize a need to interact with physical control and sensor lines associated with the process variables involved in the characterization process. Examples include:

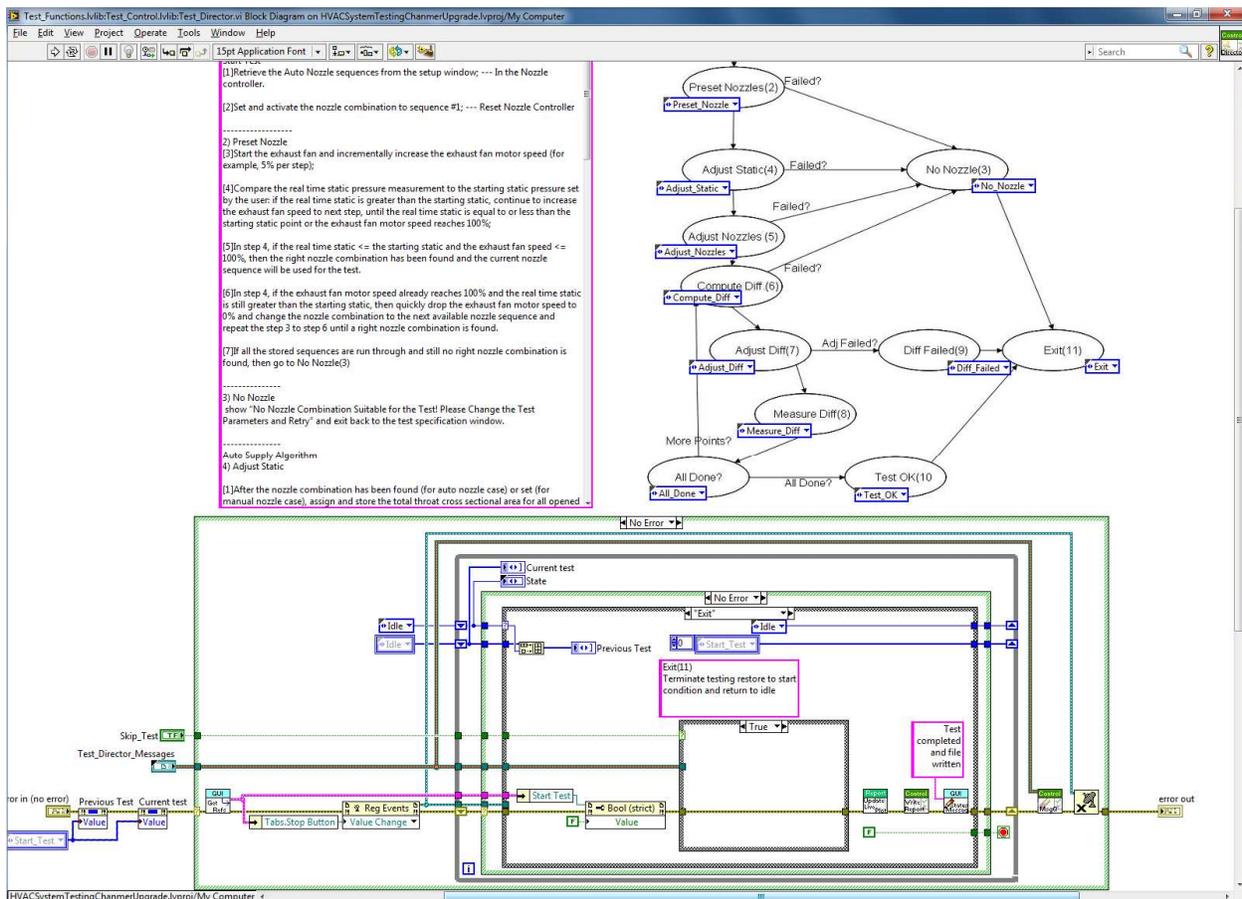
- analog input signals to measure pressure and temperature,
- digital inputs signals to monitor safety interlocks and ensure nozzles are open or closed,
- digital outputs to actuate nozzles
- analog outputs to set the speed of the exhaust motor

Each of these was implemented as 4 additional “Controllers”.

### Components Defined

With this brief review of the simplified system we find inspiration for the following major components:

Test Director – performs the decision making and coordination that would be performed by a human if the test was conducted manually. (See Figure 5 for an idea of how the test execution flows.) While the Test Director contains logic capable of handling all of the various modes of operation it does little work aside from determining which of the other components in the system should be utilized for each condition. So like and good supervisor it coordinates the efforts of its minions.



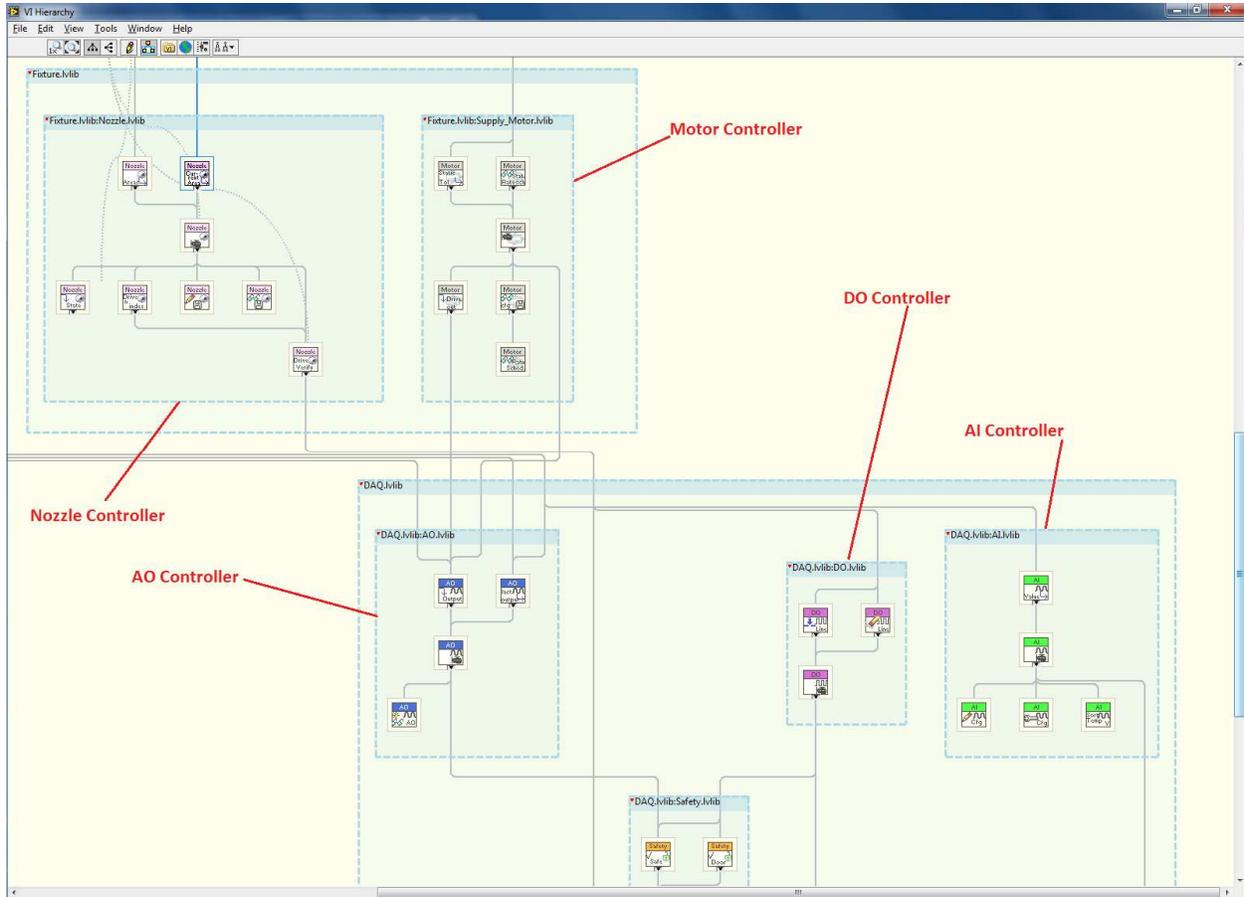
**Figure 5. Test Director Source Code (top right shows the state transition diagram)**

Nozzle Controller – Engages active nozzles, ensures their opening and closing and provides a method for sequencing to larger or smaller nozzles. As shown in Figure 4 the Nozzle Controller acts under the control of the Test Director and performs its required function by acting on the DI Controller and the DO Controller. (Note: For the sake of brevity the Safety module is not illustrated in Figure 4. All DO operations are supervised by the Safety module which monitors the state of the Wind Tunnel to ensure operator safety).

Motor Controller – Used to adjust the flow through the wind tunnel and achieve the desired flow rates. The Motor Controller provides the required mechanism that is invoked by the Test Director when the volume of air (CFM) needs adjusted.

**Realizing the Design in LabVIEW**

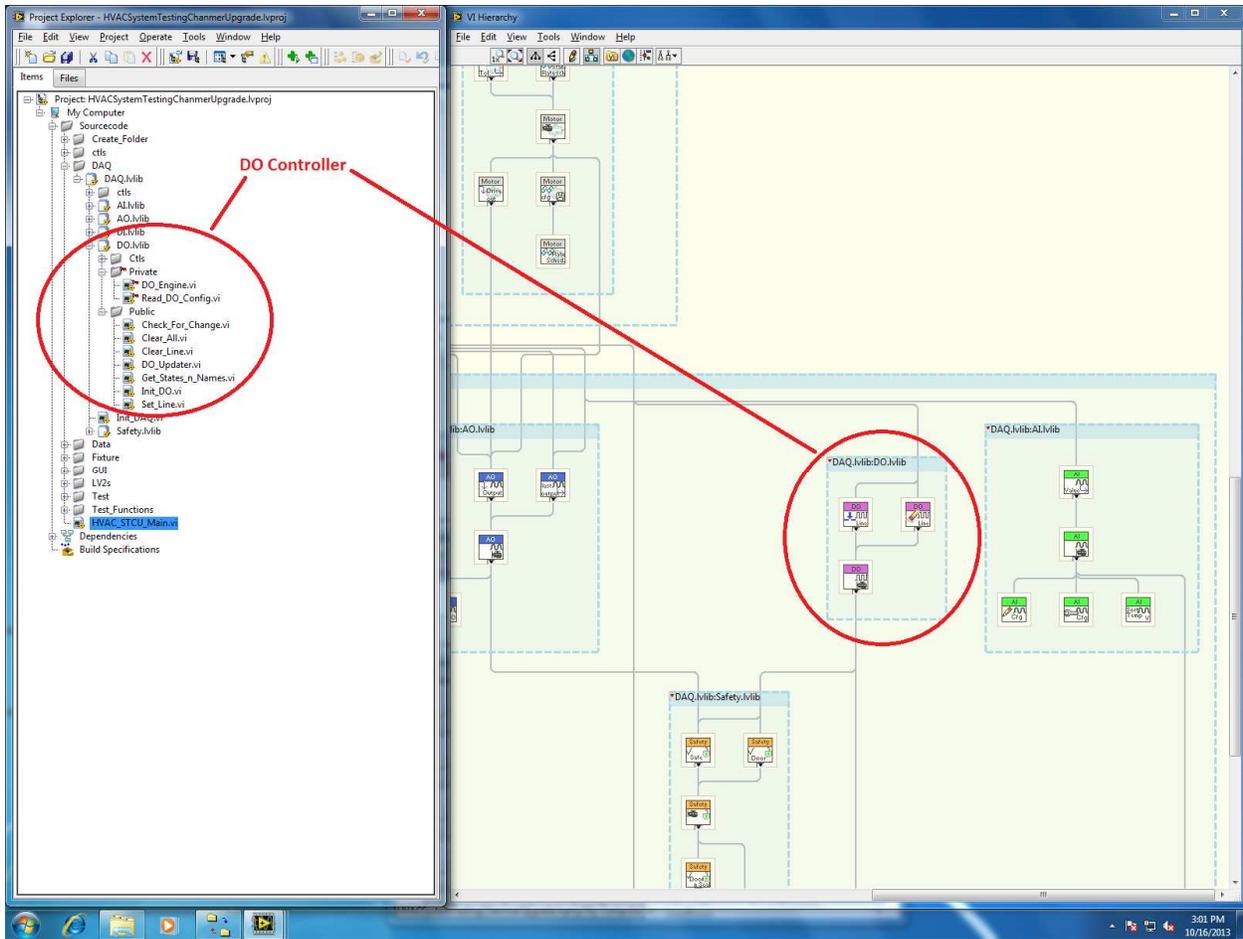
Figure 5 showed the implementation of the Test Director and includes a state called “Adjust Static”. Figure 6 shows the hierarchy of the sub-VIs called from the “Adjust Static.VI”. The hierarchy is shown with the VIs “grouped by library”. The Test Director Library is not shown in this image and off the screen to the top. Comparing Figure 6 with Figure 4 we see that LabVIEW allows us to implement a software design that closely mimics the high level design. This approach makes it easy to support and enhance the application.



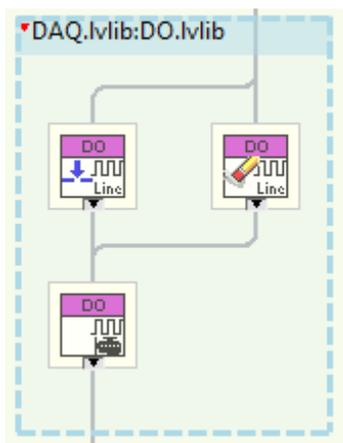
**Figure 6. Hierarchy of the Controller Libraries.**

Anyone familiar with the system design (Figure 4) and the measurement process can identify which of the elements in the high-level design will have to be modified. An example to illustrate this feature follows.

For this simple example we will start with a possible change request to “Add a new Digital output signal”. Examination of Figure 4 indicates the change will affect the DO Controller. Viewing the application hierarchy (Figure 6) shows us the DO Controller is implemented using VIs from the DO library. Opening any of the VIs in the DO library allows us to view the VI in the project (ctrl-shift-e). Expanding the virtual folders in the DO library shows us that there are only nine VIs in the DO Library so even if the icons (Figure 7 and 8) did not offer a clue as to the nature of the constituent sub-VIs there are only nine VIs involved.



**Figure 7. The DO Controller LabVIEW Library (Ivlib) in the Project (left) and in the Hierarchy Screen (right)**



**Figure 8. Expanded View of the DO.Ivlib Hierarchy shows the VI's "Set Line" (top left) "Clear Line" (Top Right) and the DO Controller (bottom)**

### **Summary**

A well-structured and easy to maintain measurement application starts with first understanding the nature of the work that needs to be performed. Once the process is known the physical nature of the fixture can be cast in such a way as to offer the resources needed to complete the work. The previous two points are true for all programming languages. When using LabVIEW we have the option to compose libraries that mimic the high-level physical design. Using the hierarchy screen with a LabVIEW application we can view how the components relate to each other. Following through with descriptive icons we can realize the power of LabVIEW's graphical system design interface and get a good idea of the functionality associated with each of the VIs that make up a library.

### **Closing**

LabVIEW programs are only as good as we write them. Knowledge of and use of the available tools and features of LabVIEW allows us to write better and more easily understood applications.

### **Contact Information:**

Gregory C. Cala, Ph.D., [gcc@DSAutomation.com](mailto:gcc@DSAutomation.com)

Data Science Automation Inc., 375 Valley Brook Road, Suite 106, McMurray PA 15317

(724)942-6330

[www.DSAutomation.com](http://www.DSAutomation.com)