Flexible Test Manager for Lift Truck Electronics Modules

Author(s):
Bradley J. Westfall, Manager, Software Engineering, Data Science Automation, Inc.
Jeffrey J. Scott, Director, Operations, Data Science Automation, Inc.

NI Product(s) Used:
LabVIEW 2014
NI USB-9861
NI GPIB-USB-HS

Category:
Advanced Manufacturing and Control
Transportation
RF and Communications

The Challenge

Create a Test Manager software framework with LabVIEW that provides the ability for an application to dynamically execute specific tests written by a client's engineers.

The Solution

Combining the benefits of traditional LabVIEW and LabVIEW Object-Oriented Programming (LVOOP) to create clean programming interfaces between the software framework and the client's customized tests created by their engineers.

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.

Purpose

To achieve product reliability goals for lift trucks, electronic modules on these trucks are subjected to reliability and qualification tests during the development process. A fundamental part of this testing is a functional test. Environmental stresses are applied to these modules. Functional tests are performed before, during, and after the stress application to verify the module is still performing as intended. Application of the environmental stresses may result in a defect or reveal a weakness in the design that can progress from a state that does not immediately affect performance to one that degrades or even stops the product from performing acceptably. This progression, called precipitation of defects, does not provide any useful information to the design team if it is not detected. A functional tester provides one means of defect detection capability.

The purpose of the functional tester is to exercise all of the circuitry within any given device under test (DUT). The test manager software provides a framework for execution of application specific tests written by test engineers. These tests control the DUT (through communications interfaces) as well as all of the programmable instruments such as digital multi-meters, switch units, signal generators, power supplies and some custom-built hardware such as power distribution circuitry and electronic loads. Each of the inputs on the DUT is stimulated by an output of a programmable instrument. The DUT actual response is monitored by the instruments and the communications interfaces to the DUT and compared to the expected response. The functional test manager runs on a Windows PC that interacts with the programmable instruments via NI-USB GPIB and LXI. It interacts with the DUT via NI-USB CAN interfaces and the RS422 interface (via RS232 or USB adapters). Up to six DUTs may be connected at one time; however, they are tested sequentially.

Approach
Using National Instrument's LabVIEW and DAQ hardware, Data Science Automation was able to create an intuitive operator interface while providing a flexible development environment for the client's test engineers. (Figures 1 & 2) This was accomplished by implementing LabVIEW's ability to produce applications using an Object-Oriented Programming (LVOOP) paradigm. Object-oriented programming has demonstrated its superiority over procedural programming as an architecture choice in several programming languages including G.

Test Manager
Data Science Automation created a stable and flexible test manager that first loaded all test code modules (created by the client's test engineers) into memory. By preloading the test code modules objects into memory, the application could seamlessly transition from one test to another with no noticeable effect to the CPU's performance. This was only available due to LabVIEW's ability to dynamically invoke the desired code module object when requested. (Figure 3) Since the test manager only had to invoke LabVIEW's ability to launch the requested test, DSA was able to focus our programming energy creating a framework to manage the test module sequencing.

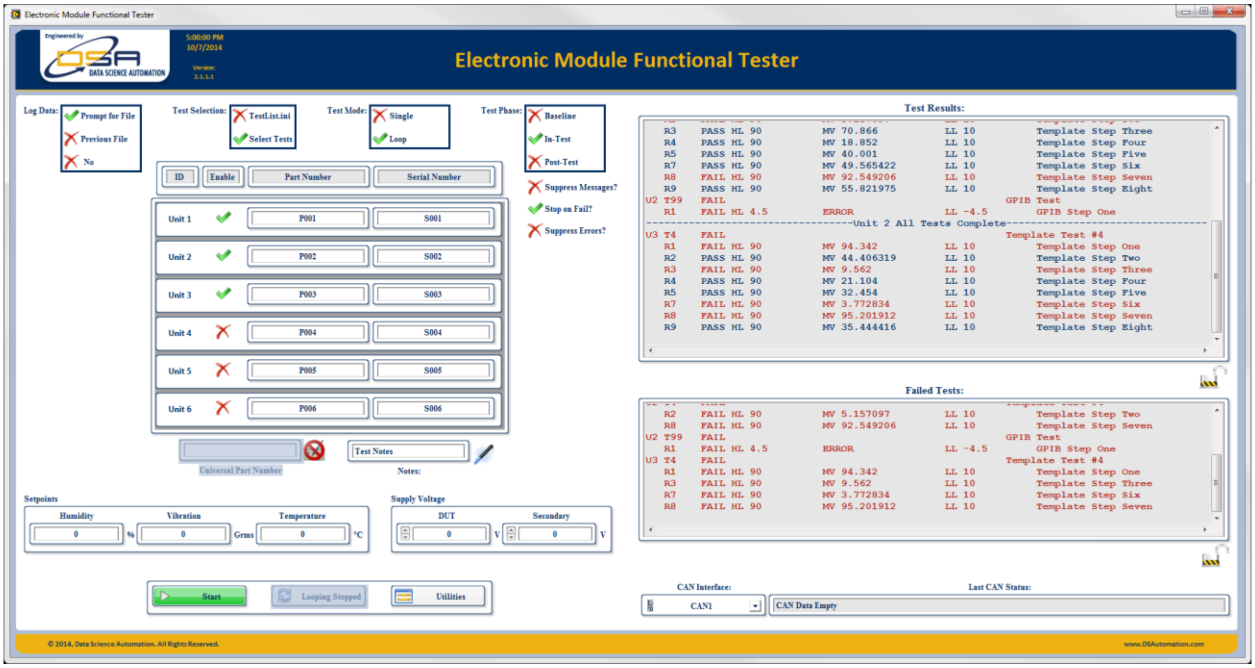
Customizable Code Module
Data Science Automation utilized LVOOP's ability to allow objects to inherit from their parent's classes to create a base class that had all the mandatory functionality (such as file writing, GUI updates, error handling and progress notification). From this base class, the client's engineers would copy template code module objects simply by right clicking in the Project Explorer on the object type needed then selecting: Save>Save As…(Figure 2). From that point forward the client's engineers could develop the source code specific to their requirements without concerning themselves with the tedious mandatory functionality. This approach eliminated the risk of possibly making code errors related to the base class that would lead to the need for debugging and troubleshooting. They could focus solely on the custom aspects of their tests. Also since the client was using NI communications interfaces to interact with the DUT, the client's engineers have well-documented, reliable, commercial off-the-shelf (COTS) tools to leverage in the creation of their test modules.

National Instruments Support
Without National Instruments continued innovation and adaption of the LabVIEW programming environment, Data Science Automation and our client would have had a more difficult path in creating a fully customizable solution. This solution, once fully adapted, will allow our client to produce, test and verify new circuit designs in a fraction of the time while allowing them to upgrade aging test equipment with more modern testing solutions. The success of this system relied on National Instruments' ability to provide a stable and robust development environment while providing the software tools to allow for rapid development and debugging. The ability to quickly integrate COTS communication interface hardware for a broad diversity of industry standard protocols as well as custom protocols available through the LabVIEW Tools Network allowed DSA to cost-effectively meet the client's operational requirements. Once completed with client-developed test modules, the system will replace a legacy, error prone process that took days and sometimes weeks to implement with an automated system that will give the operators, managers and stakeholders real-time feedback on their DUT performance. This saved the company hundreds of man-hours redoing faulty, error filled tests while providing a modern system with real time analyses that can be expanded to meet both near term and long term goals of providing low-cost, high-quality products.

Benefits of Partnership
The Alliance partnership between Data Science Automation and National Instruments has allowed DSA to tackle many extremely complex engineering challenges while maximizing our clients' return on their investment. This Test Manager project was no exception. By selecting LabVIEW and NI communications interfaces Data Science Automation was able to create a system that not only reduced costs but greatly increased the client's productivity and profitability by enabling them to reduce unnecessary re-testing of failed units while provide real-time DUT performance. Also by using National Instruments hardware and software, the Test Manager application leveraged the reliability

and availability of commercial off-the-shelf (COTS) equipment. These savings allowed functional enhancements to the Test Manager.

Contact Information
Bradley J. Westfall, info@DSAutomation.com
Data Science Automation, 375 Valley Brook Road, Suite 106, McMurray, PA 15317-3370
www.DSAutomation.com


Electronic Module Functional Tester
Engineered by
DATA SCIENCE AUTOMATION
Electronic Module Functional Tester
Start
Looping Stopped
Utilities
Universal Part Number
© 2014, Data Science Automation. All Rights Reserved.
www.DSAutomation.com

5:00:00 PM
10/7/2014
Version:
2.1.1.1

Log Data: ✓ Prompt for File ✗ Previous File ✗ No

Test Selection: ✗ TestList.ini ✓ Select Tests

Test Mode: ✗ Single ✓ Loop

Test Phase: ✗ Baseline ✓ In-Test ✗ Post-Test

✗ Suppress Messages?
✓ Stop on Fail?
✗ Suppress Errors?

ID	Enable	Part Number	Serial Number
Unit 1	✓	P001	S001
Unit 2	✓	P002	S002
Unit 3	✓	P003	S003
Unit 4	✗	P004	S004
Unit 5	✗	P005	S005
Unit 6	✗	P006	S006

Notes: Test Notes

Setpoints

Humidity	Vibration	Temperature
0 %	0 Grms	0 °C

Supply Voltage

DUT	Secondary
0 V	0 V

Test Results:

```
   R3     PASS HL 90        MV 70.866         LL 10        Template Step Three
   R4     PASS HL 90        MV 18.852         LL 10        Template Step Four
   R5     PASS HL 90        MV 40.001         LL 10        Template Step Five
   R7     PASS HL 90        MV 49.565422      LL 10        Template Step Six
   R8     FAIL HL 90        MV 92.549206      LL 10        Template Step Seven
   R9     PASS HL 90        MV 55.821975      LL 10        Template Step Eight
U2 T99    FAIL                                         GPIB Test
   R1     FAIL HL 4.5       ERROR             LL -4.5      GPIB Step One
----------------------------------Unit 2 All Tests Complete----------------------------------
U3 T4     FAIL                                         Template Test #4
   R1     FAIL HL 90        MV 94.342         LL 10        Template Step One
   R2     PASS HL 90        MV 44.406319      LL 10        Template Step Two
   R3     FAIL HL 90        MV 9.562          LL 10        Template Step Three
   R4     PASS HL 90        MV 21.104         LL 10        Template Step Four
   R5     PASS HL 90        MV 32.454         LL 10        Template Step Five
   R7     FAIL HL 90        MV 3.772834       LL 10        Template Step Six
   R8     FAIL HL 90        MV 95.201912      LL 10        Template Step Seven
   R9     PASS HL 90        MV 35.444416      LL 10        Template Step Eight
```

Failed Tests:

```
   R2     FAIL HL 90        MV 5.157097       LL 10        Template Step Two
   R8     FAIL HL 90        MV 92.549206      LL 10        Template Step Seven
U2 T99    FAIL                                         GPIB Test
   R1     FAIL HL 4.5       ERROR             LL -4.5      GPIB Step One
U3 T4     FAIL                                         Template Test #4
   R1     FAIL HL 90        MV 94.342         LL 10        Template Step One
   R3     FAIL HL 90        MV 9.562          LL 10        Template Step Three
   R7     FAIL HL 90        MV 3.772834       LL 10        Template Step Six
   R8     FAIL HL 90        MV 95.201912      LL 10        Template Step Seven
```

CAN Interface: CAN1

Last CAN Status: CAN Data Empty

Figure 1 - Operator Display after Testing

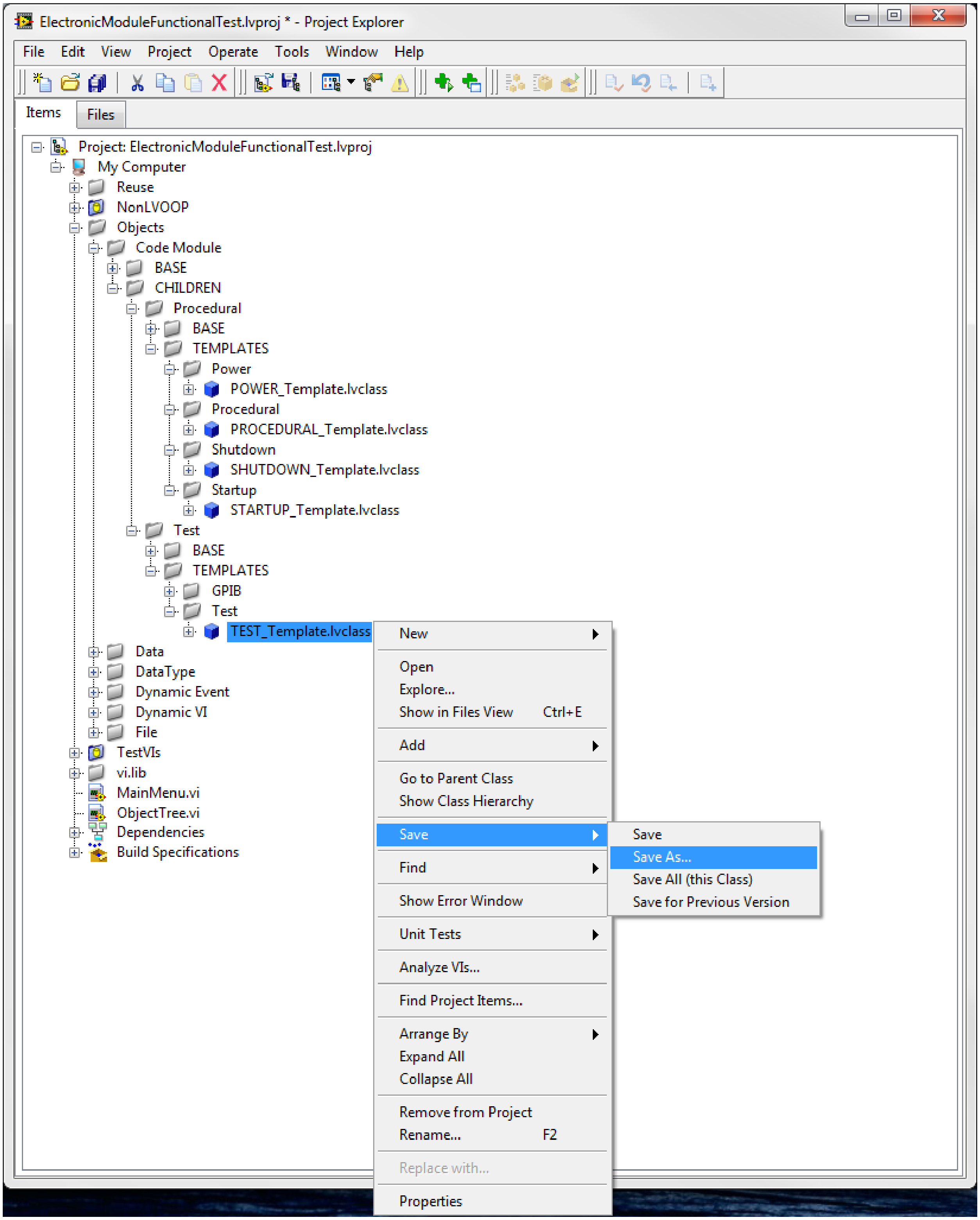
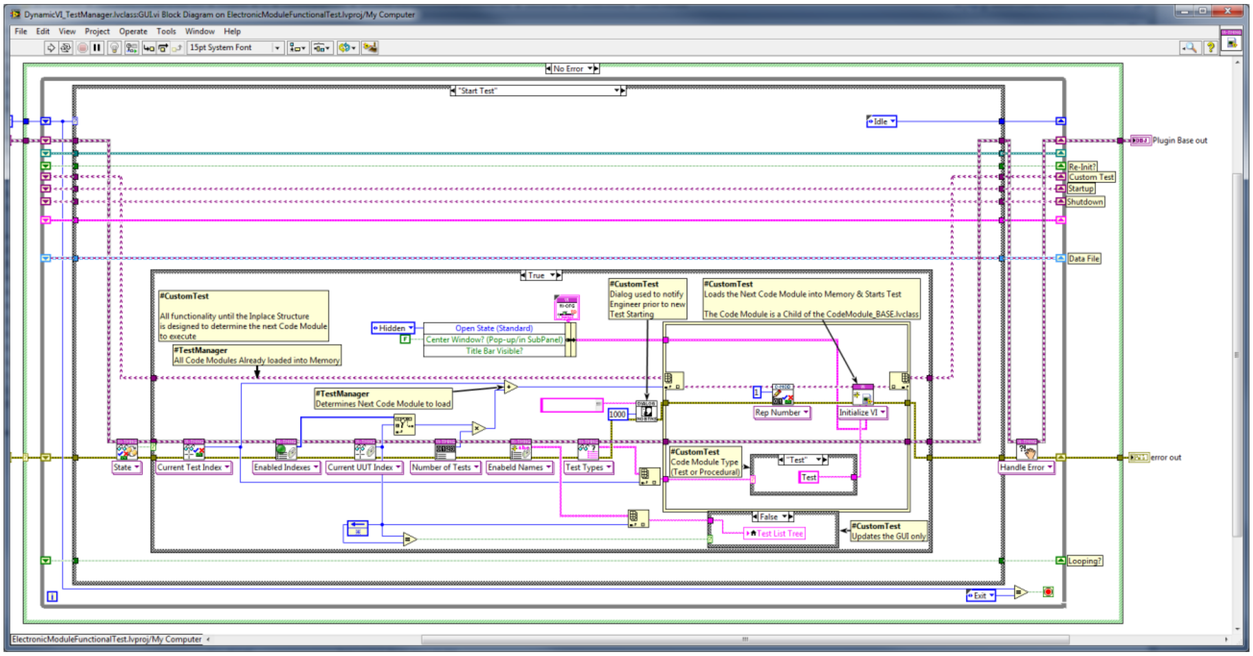


Figure 2 - New Test Code Module Template Creation



DynamicVI_TestManager.lvclass:GUI.vi Block Diagram on ElectronicModuleFunctionalTest.lvproj/My Computer

#CustomTest
All functionality until the Inplace Structure is designed to determine the next Code Module to execute

#TestManager
All Code Modules Already loaded into Memory

#TestManager
Determines Next Code Module to load

#CustomTest
Dialog used to notify Engineer prior to new Test Starting

#CustomTest
Loads the Next Code Module into Memory & Starts Test

The Code Module is a Child of the CodeModule_BASE.lvclass

#CustomTest
Code Module Type (Test or Procedural)

#CustomTest
Updates the GUI only

Figure 3 - Dynamic Execution of the Next Code Module Test