

Keysight Body/Safety ECU Functional Test, Reference Solution

Notice: This document contains references to Agilent. Please note that Agilent's Test and Measurement business has become Keysight Technologies. For more information, go to www.keysight.com.

Notices

Copyright Notice

© Keysight Technologies 2016

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Keysight Technologies as governed by United States and international copyright laws.

Manual Part Number

U8989-90005

Edition

Edition 1, Feb 1, 2016

Printed in:

Printed in Malaysia

Published by:

Keysight Technologies
Bayan Lepas Free Industrial Zone,
11900 Penang, Malaysia

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Declaration of Conformity

Declarations of Conformity for this product and for other Keysight products may be downloaded from the Web. Go to <http://www.keysight.com/go/conformity>. You can then search by product number to find the latest Declaration of Conformity.

U.S. Government Rights

The Software is “commercial computer software,” as defined by Federal Acquisition Regulation (“FAR”) 2.101. Pursuant to FAR 12.212 and 27.405-3 and Department of Defense FAR Supplement (“DFARS”) 227.7202, the U.S. government acquires commercial computer software under the same terms by which the software is customarily provided to the public. Accordingly, Keysight provides the Software to U.S. government customers under its standard commercial license, which is embodied in its End User License Agreement (EULA), a copy of which can be found at <http://www.keysight.com/find/sweula>. The license set forth in the EULA represents the exclusive authority by which the U.S. government may use, modify, distribute, or disclose the Software. The EULA and the license set forth therein, does not require or permit, among other things, that Keysight: (1) Furnish technical information related to commercial computer software or commercial computer software documentation that is not customarily provided to the public; or (2) Relinquish to, or otherwise provide, the government rights in excess of these rights customarily provided to the public to use, modify, reproduce, release, perform, display, or disclose commercial computer software or commercial computer software documentation. No additional government requirements beyond those set forth in the EULA shall apply, except to the extent that those terms, rights, or licenses are explicitly required from all providers of commercial computer software pursuant to the FAR and the DFARS and are set forth specifically in writing elsewhere in the EULA. Keysight shall be under no obligation to update, revise or otherwise modify the Software. With respect to any technical data as defined by FAR 2.101, pursuant to FAR 12.211 and 27.404.2 and DFARS 227.7102, the U.S. government acquires no greater than Limited Rights as defined in FAR 27.401 or DFAR 227.7103-5 (c), as applicable in any technical data.

Warranty

THE MATERIAL CONTAINED IN THIS DOCUMENT IS PROVIDED “AS IS,” AND IS SUBJECT TO BEING CHANGED, WITHOUT NOTICE, IN FUTURE EDITIONS. FURTHER, TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, KEYSIGHT DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH REGARD TO THIS MANUAL AND ANY INFORMATION CONTAINED HEREIN, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. KEYSIGHT SHALL NOT BE LIABLE FOR ERRORS OR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, USE, OR PERFORMANCE OF THIS DOCUMENT OR FOR ANY INFORMATION CONTAINED HEREIN. SHOULD KEYSIGHT AND THE USER HAVE A SEPARATE WRITTEN AGREEMENT WITH WARRANTY TERMS COVERING THE MATERIAL IN THIS DOCUMENT THAT CONFLICT WITH THESE TERMS, THE WARRANTY TERMS IN THE SEPARATE AGREEMENT SHALL CONTROL.

Safety Information

CAUTION

A CAUTION notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a CAUTION notice until the indicated conditions are fully understood and met.

WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

Waste Electrical and Electronic Equipment (WEEE) Directive 2002/96/EC

This instrument complies with the WEEE Directive (2002/96/EC) marking requirement. This affixed product label indicates that you must not discard this electrical or electronic product in domestic household waste.

Product category

With reference to the equipment types in the WEEE directive Annex 1, this instrument is classified as a “Monitoring and Control Instrument” product.

The affixed product label is as shown below.



Do not dispose in domestic household waste.

To return this unwanted instrument, contact your nearest Keysight Service Center, or visit <http://about.keysight.com/en/companyinfo/environment/takeback.shtml> for more information.

Sales and Technical Support

To contact Keysight for sales and technical support, refer to the support links on the following Keysight websites:

- www.keysight.com/find/ts8989ref
(product-specific information and support, software and documentation updates)
- www.keysight.com/find/assist
(worldwide contact information for repair and service)

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

Table of Contents

Waste Electrical and Electronic Equipment (WEEE) Directive 2002/96/ EC	3
Product category	3
Sales and Technical Support	3
Introduction	7
Software Overview	8
TestExec SL	8
TS-5000 Instrument Library	9
Keysight Body/Safety ECU Functional Test Demo Program Overview	10
BodySafetyDemoGUI overview	12
BodySafetyDemoProgram overview	14
Instrument library overview	14
Datalogging overview	16
BodySafetyDemoGUI Details	18
Program execution	18
Running test with UI	19
Closing the program	20
BodySafetyDemoProgram Details	21
Instrument initialization	21
Running test in TestExec SL	22
Switching topologies	22
Body/Safety ECU test	26
Closing the program	47

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

Introduction

This document describes the required procedure to build and modify the Keysight Body/Safety Electronic Control Unit (ECU) Functional Test Demo Program. Executable versions of this program are included with TS-8989 Body/Safety Reference Solution Demo Systems. Demo program and supporting files are available from your Keysight representative.

The following are the initial features in Version 1.0:

- Support for E6198B family Pin Matrixes and Load Cards
- Support for N6700B family modular Power Supply
- Support for M9183A Digital Multi-Meter
- Support for M9188A Dynamic D/A Converter
- Support for M9216A High Voltage Data Acquisition
- DUT standby current test via E6178B
- High Side Driver measurement using M9183A, including:
 - Voltage output signal via E8792A switch matrix
 - Current drawn from driver to high power load via E6176A load card
- Frequency Input stimulus using M9188A with DUT assisted test result
- Low Side Solenoid Driver measurement, including:
 - Data acquisition of solenoid driver by capturing the "on" voltage level and likewise, when the driver turns off, it must accurately capture the resting voltage level using M9216A
 - Analysis on driver functionality from the data captured, saturation voltage during "on" state, and fly-back voltage to verify quality of DUT clamping circuit

Software Overview

TestExec SL

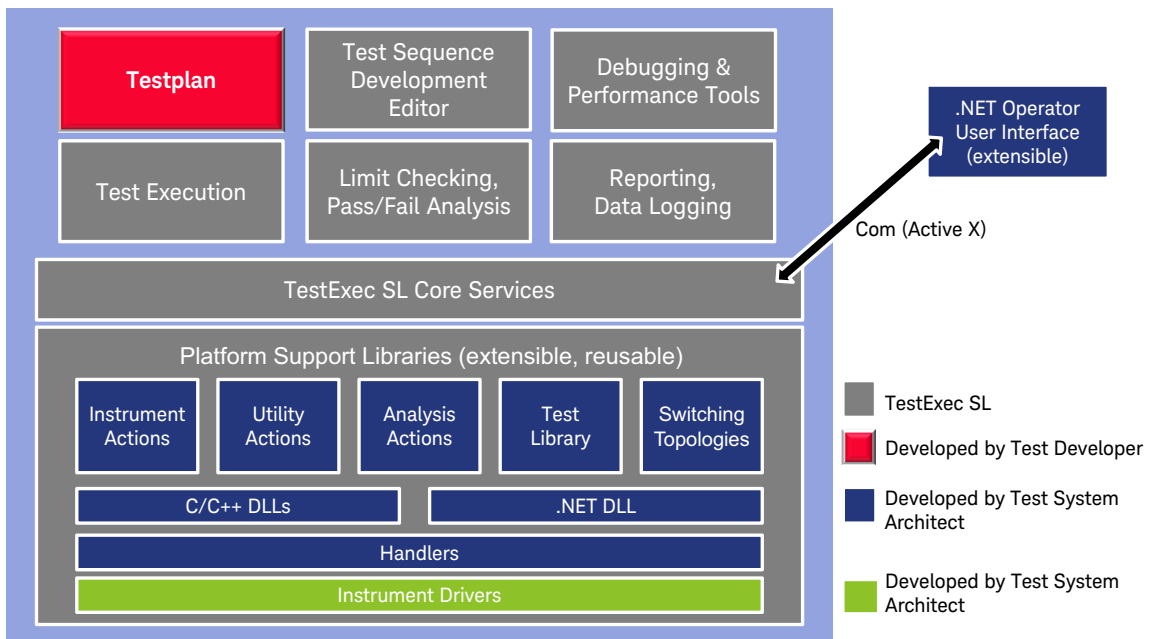
The Keysight TestExec SL software comes preinstalled on the PXI embedded PC or Industrial PC (IPC) of the TS-8989 system, and is ready for software development. TestExec SL software helps to reduce development time by providing a complete test development and test execution environment which also allows the user to organize and order tests, reconfigure the tests, profile the execution speed, and debug tests. The software provides an efficient and effective structure for developing the test plans and sequencing for testing featuring the following:

- 1** Reusable hardware actions/library created by software team, letting user reuse the hardware actions in various programs. In addition, topology editor/switch manager with streamlined switching topology configurations helps to define logical paths between instrument, fixture, and DUT.
- 2** Simple hierarchical structure and advanced executive tools, including advanced sequencing and test flow controls, and debugging tools.
- 3** Test profiler helps to monitor and track tests, display key parameters in graphical format, and provide insights to optimize test throughput.
- 4** Open architecture using Microsoft COM standard for easy sharing; data logging and reporting with customizable format for database systems.

For more information about Keysight TestExec SL, visit [TestExec SL](#) webpage to explore its video introduction and free software trial with online help.

TS-5000 Instrument Library

Built-in actions and library are essentially easy-to-use commands that help to facilitate communication between instruments with TestExec SL. The TestExec SL includes TS-5000 family hardware library for instrument and switching box that have been qualified to work seamlessly on the TS-8989 without compromising compatibility and performance. The diagram below illustrates the concept of TestExec SL software platform as well as its link with instrument actions/libraries. Testplan and switching topologies files (in red) are the only effort for system development, since Keysight standard TS-5000 action library/handlers (in blue) has been already developed by Keysight system architect. The developer only needs to create the hardware handler when he needs to integrate instruments that are not supported by the TS-5000 library.



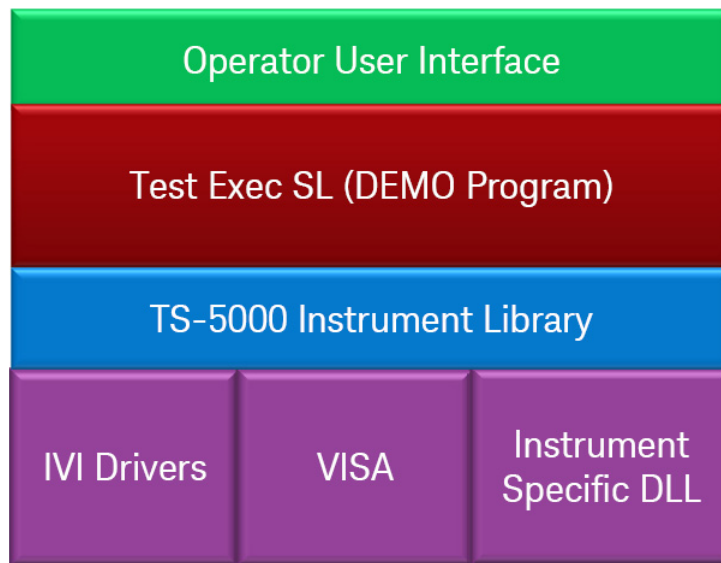
Please refer to the *TS-5000 System Software User Manual* to guide you through how the TestExec SL platform works with the TS-5000 family hardware.

Keysight Body/Safety ECU Functional Test Demo Program Overview

The Body/Safety demo program can be operated under Operator User Interface (UI) mode to enable testing and data collection. In some cases there may be a need to modify the demo program or include some additional test items from the demo program. You may launch the test program in TestExec SL software for editing as well as testing. The demo program is also useful as a guide for best practices for various switching and measurements. The *TS-5000 System Software User Manual* is available to assist you.

This section of the document provides instructions on how to build the solution and an overview of the solution structure.

The following diagram shows the high level structure of the software package:



The Package contains only two main layers of software: the Operator User Interface and the Demo Program.

Instrument interfaces (TS-5000 instrument action/library) were developed in .dll or .lib by the Keysight system architect.

Operator user interface:

BodySafetyDemoGUI: The graphical user interface developed with VB.NET. This UI is customized with waveform display of solenoid driver time response.

Alternatively, if the waveform display is unnecessary, user can launch the demo program through TestExec SL typical operator UI from:

All Programs -> Agilent TestExec SL 7.1 -> TypicalOpUI

Demo program:

BodySafetyDemoProgram:

“TS8989_ECU_Demo.tpa” is the TestExec SL testplan for typical functional testing of body/safety DUT in manufacturing environment.

“TS8989System.ust” and “peITS8989 DUT uut.ust” are the switching topologies files of the system and DUT respectively.

Instrument library:

Instruments used in this demo are listed below, with its library residing in C:\Program Files (x86)\Agilent\TS-5000 System Software\Bin:

hwhdmm.dll: PXI DMM M918xA

hwhdacInst.dll: PXI Analog Output M9188A

hwhDAQ.dll: PXI Data Acquisition M9216A

hwhSwitchUnit.dll: E6198B family Pin Matrixes and Load Cards

E8792A – 32-Pin Matrix Card

E6178B – 8 Channels 30 A Load Card

E6176A – 16 Channels 7.5 A Load Card

hwhps.dll: N6700B family modular power supply

Since the PXI CAN card is not part of TS-5000 family instrumentation, a custom library is developed and stored in

\$UserFilesDir\$\UUTs\Bin\PxlTestExecHandler.dll

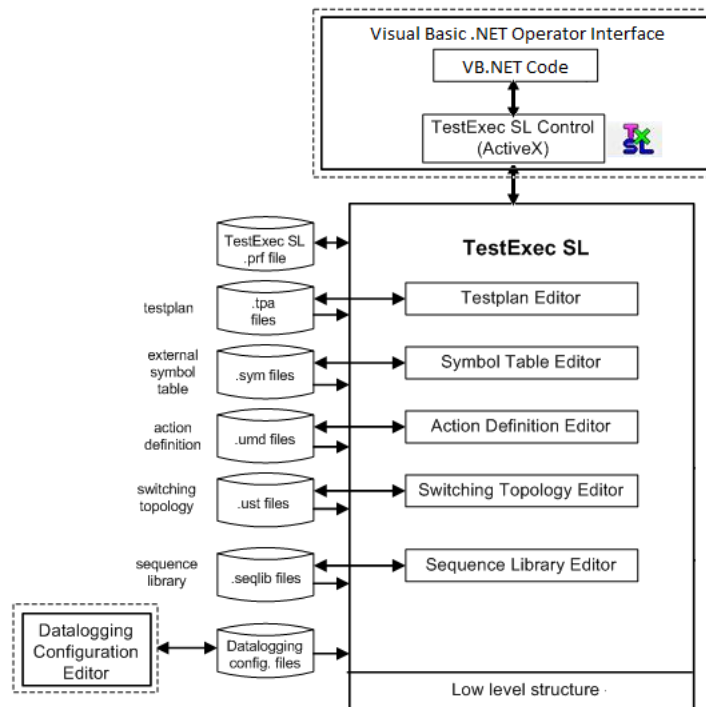
Data logging:

TestExec SL default data logging allows data export in spreadsheet, XML (eXtensible Markup Language), or Keysight 3070 formats.

BodySafetyDemoGUI overview

The *BodySafetyDemoGUI* is a Visual Basic .NET project that provides operator UI for the Body/Safety Demo Program. The UI is mainly designed for product design verification or mass production environment. Buttons are created allowing user to choose which TestExec SL testplan to be loaded, run, stop, or abort the testing.

In the following figure, notice how with codes written in Visual Basic .NET, the UI actually resides outside the TestExec SL process space, and communicates with TestExec SL via ActiveX controls. The TestExec SL control enables the codes in the UI to start the TestExec SL program, run testplans, and return status information from the testplans.



Keysight Body/Safety ECU Functional Test Demo Program Overview

The project includes the following files:

File	Description
formMain.vb	A Windows form that is the main operator user interface for the Body/Safety demo program.
frmErrorDialog.vb	A Windows form that displays exceptions information when application error exceptions are encountered.
frmLoadTestplan.vb	A Windows form that is used to support the loading of testplans. It is designed around reading the registered testplan entries under the [Testplan Reg] section in the Preferences file.
frmLogin.vb	A Windows form that is used to support user logging in of the Body Safety Demo Program.
frmPreDefinedTxSLUserMessages.vb	A Windows form that is used to handle predefined user messages that may be sent from TestExec SL testplans.
frmSelectVariant.vb	A Windows form that is used to handle the task of selecting a testplan variant for the testplan execution.
frmSplash.vb	A Windows form that is used to indicate the program application information.
modAppSpecific.vb	A file that is used to hold information specific to the program application.
modConfiguration.vb	A file that is used to set configuration variables for the program application.
AssemblyInfo.vb	A file that is used to modify general information of the program application file.

BodySafetyDemoProgram overview

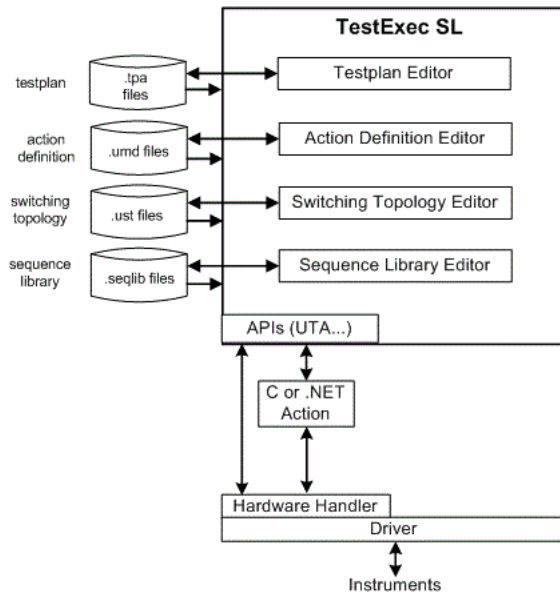
Inside TestExec SL there are various editors (refer to the figure on [page 12](#)), each of which lets you create and edit files of a specific type. For example, the Testplan Editor stores information about testplans in files with “.tpa” extensions; and Topology Editor stores information about hardware and its signal paths in files with “.ust” extensions.

“*TS8989_ECU_Demo.tpa*” is the testplan which defines the test sequence flow of the body/safety ECU functional testing. The testplan is where the predefined sequences, tests and actions are used together. Sequence is a named series of test groups, tests, and/or flow control statements executed in a predefined order. Sequence Main contains desired test items for the ECU testing, sequence AfterMain performs cleanups after Main sequence execution, and sequence Exception performs cleanups whenever an exception is triggered.

The Topology Editor is used to make the Test Executive aware of hardware modules that are available as resources during testing. “*TS8989System.ust*” is the system topology file which defines all instrumentation and its switchable connections within the system. The “*pe1TS8989 DUT uut.ust*” is the DUT topology file which defines nodes of ECU, as well as wiring connections between the ECUs and TS-8989 system.

Instrument library overview

The figure on the following page shows an extensive set of application programming interfaces (APIs) whose function names begin with "UTA". These provide a programmatic interface to actions that do useful tasks, such as making measurements. These APIs also let TestExec SL control instruments via hardware handlers that are in a software layer residing between TestExec SL and a conventional instrument driver. As the paths in the diagram imply, actions can control instruments directly, via a hardware handler. Typically PXI, LXI, and GPIB hardware instrumentation are controlled with pre-developed TS-5000 actions, or customized actions. Whereas switching modules, such as relay matrixes, multiplexer, and load card relay are controlled by TestExec SL without the layer of instrument-specific actions.



Action Definition Editor stores information about actions in files whose extensions are “.umd”. Default TestExec SL actions (.umd files) are stored in C:\Program Files (x86)\Agilent\TestExec SL 7.1\Actions.

TS-5000 family instrumentation actions are stored in C:\Program Files (x86)\Agilent\TS-5000 System Software\actions, with the instrument hardware handler stored in C:\Program Files (x86)\Agilent\TS-5000 System Software\Bin.

In this demo, third party CAN communication card actions are developed in .NET style, and stored in %UserFilesDir%\UUTs\Actions, with the custom hardware handler stored as %UserFilesDir%\UUTs\Bin\PxITestExecHandler.dll.

Datalogging overview

The Datalogging Configuration Editor, which is a separate application provided within TestExec SL, lets you manipulate a set of configuration files (.rdef) that determine how datalogging operates.

Datalogging automatically collects data about tests that have pass/fail limits specified for them when a testplan runs. Subsequent study of this data can help you improve the testing and manufacturing process and track the testing done on a particular DUT. The system writes a new file of datalogging information each time a testplan or a loop in a testplan runs, automatically names each file with a unique hexadecimal name derived from the system date and time plus an extension of “.xml” or “.log” depending upon the format chosen for it.

File format can be designated by editing the \$UserFilesDir\Test Systems\Preferences.upf, in this example spreadsheet file format was selected under **[Data Log]** session.

```

File Edit Format View Help
[Data Log]
; Format File specifies the data record format. Definition File specifies the
; fields that can be logged. Select one pair of files from the following list.
; selection is done by commenting out the current choices (; in first column)
; and removing the ";" on the new choices.

; For an XML formatted file using TxSL style names, use:
;Format File=$UserFilesDir\Test Systems\TxSLXMLRecords.rdef
;Definition File=$UserFilesDir\Test Systems\TxSLXMLFields.fdef

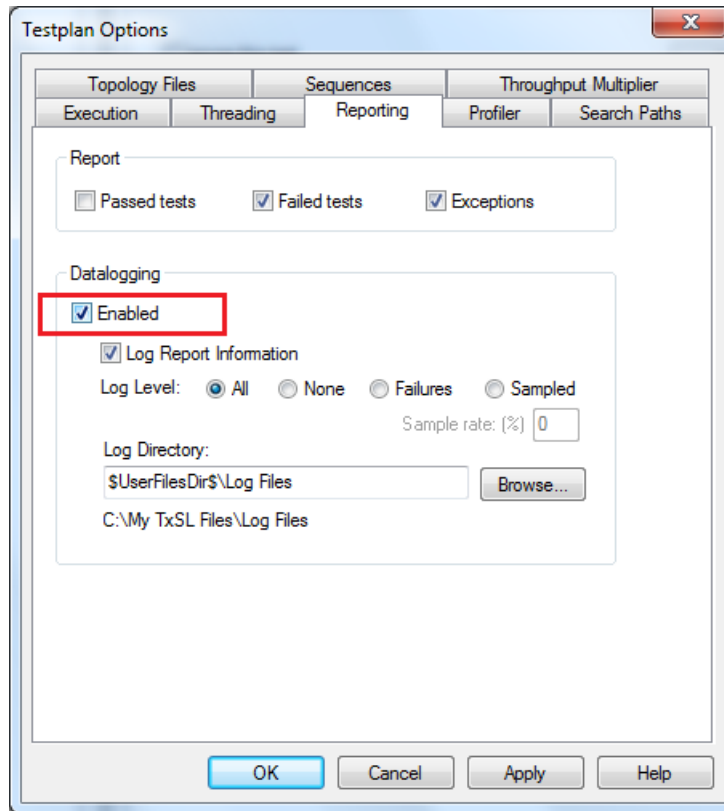
; For a spreadsheet compatible file using TxSL style names, use;
;Format File=$UserFilesDir\Test Systems\TxLSSRecords.rdef
;Definition File=$UserFilesDir\Test Systems\TxLSSFields.fdef

; For spreadsheet format using 3070 names, use:
;Format File=$UserFilesDir\Test Systems\ssfmdtdef.ini
;Definition File=$UserFilesDir\Test Systems\ssdsdef.ini

; For the Agilent 3070 format using 3070 names, use:
;Format File=$UserFilesDir\Test Systems\hpfmdtdef.ini
;Definition File=$UserFilesDir\Test Systems\dsdef.ini

; The default data log files are shown below.
;Format File=$UserFilesDir\Test Systems\hpfmdtdef.ini
;Definition File=$UserFilesDir\Test Systems\dsdef.ini
    
```

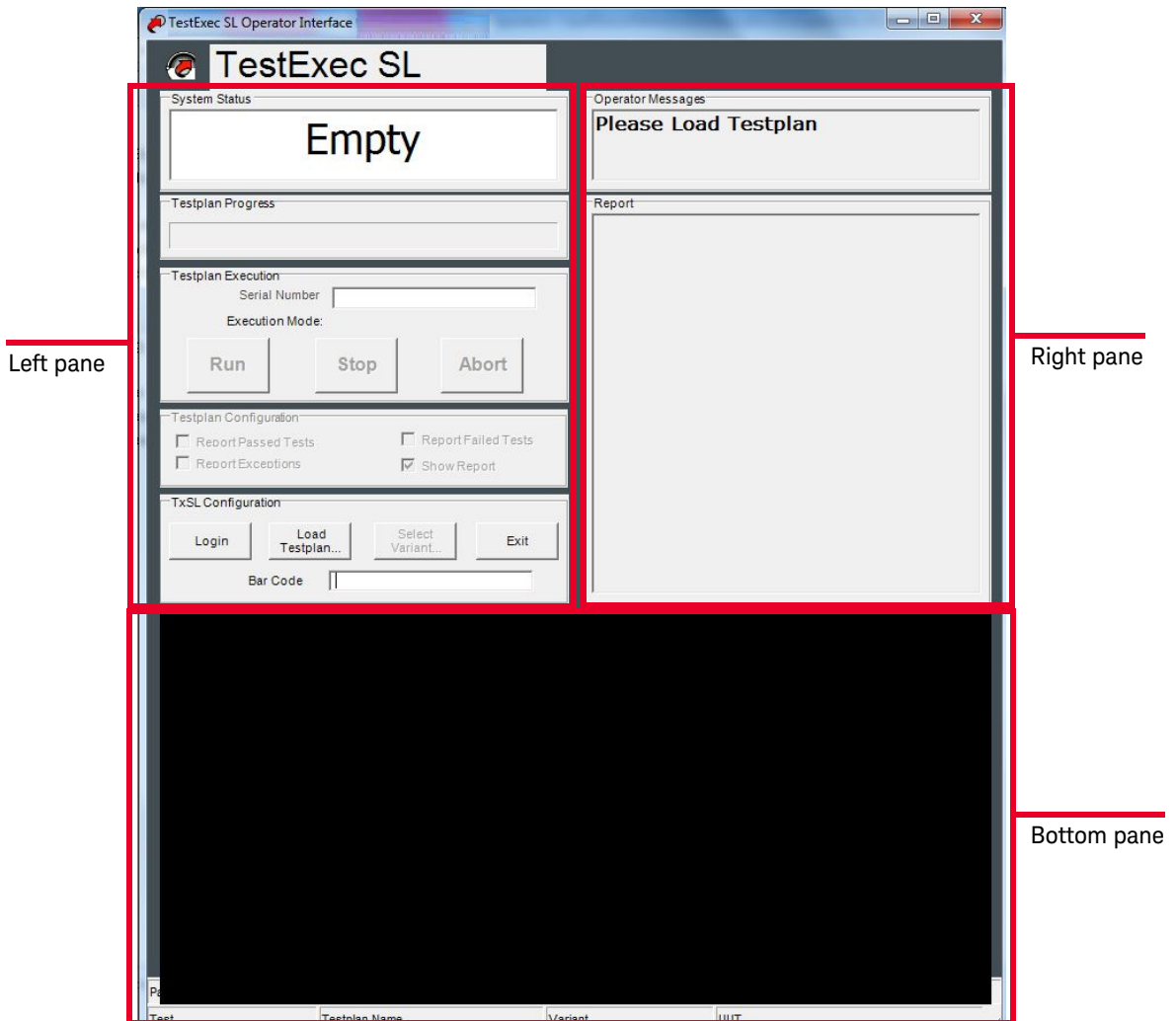
On the other hand, test results will be exported into a datalog directory by enabling Datalogging via TestExec SL reporting option as shown on the following page.



BodySafetyDemoGUI Details

Program execution

The BodySafetyDemoGUI VB.NET project generates TypicalOpUI.exe (an executable program) in .bin after the project is successfully built. By launching the TypicalOpUI.exe as Administrator with the default login name “wild” without password, the UI appears as shown.



In general, the UI left pane consists of buttons to control TestExec SL, for example trigger a start/stop the demo program, meanwhile the right pane returns the TestExec SL test result while the program is running. The bottom pane is not as meaningful as the left/right pane; it is a graphical chart that displays the solenoid driver transient response, which is only useful for those performing design verification.

Running test with UI

“TestExecSL1” is the ActiveX control used in this demo UI program. It provides interaction between OPUI with TestExec SL programmatically. Methods/events calling for TestExec SL software control will be described in the corresponding column of each action taken by user.

Upon login [Method: *TestExecSL1.Security.Login (username, password)*], choose a testplan by clicking the **“Load Testplan”** button [Method: *TestExecSL1.LoadTestplan (pathname)*]. List of testplans for selection was pre-defined by developer in the preferences file (.upf) [Method: *TestExecSL1.RegisteredTestplans.Items (index)*], which allows for different test programs to be used with this UI. Once the demo program is selected [Method: *TestExecSL1.LoadTestplan (path)*], the **“Run”**, **“Stop”**, **“Abort”**, and **“Select Variant”** buttons will be enabled for the control of the TestExec SL program.

The **“Select Variant”** button offers access to the testplan's variants from currently loaded demo program. However, this body/safety demo program only comes with one “Normal” variant, therefore user is not required to select any other variant from “Select Variant”.

Ensure the DUT is properly engaged with the TS-8989 system before clicking **“Run”** to start the demo program [Method: *TestExecSL1.Testplan.Run()*]. The first cycle of test will take a little longer due to the TestExec SL trigger for hardware initialization at the beginning of test – refer to the [Instrument initialization](#) section.

While testing is taking place, each measurement result captured will be populated on the report column located in the right pane of the UI [Method: *Private Sub TestExecSL1_ReportMessage (TextBlock as String)*]. At the end of the test, the UI will then read the waveform data captured by PXI M9216A during the test [Event: *TestExecSL1.SymbolTables (tablename).Symbols (symbolname).Value*], and display both solenoid drivers transient responses in the graphical chart.

While the demo program is running, clicking the “**Stop**” button will cease testing [*Method: TestExecSL1.Testplan.Stop ()*], execution halts after the current test operation has finished, ensuring switching goes back to cleanup state. Meanwhile “**Abort**” would stop testing immediately without any cleanup operations [*Method: TestExecSL1.Testplan.Abort ()*], which leaves the test system's hardware in an unknown state, hence this is only to be used during emergency.

Closing the program

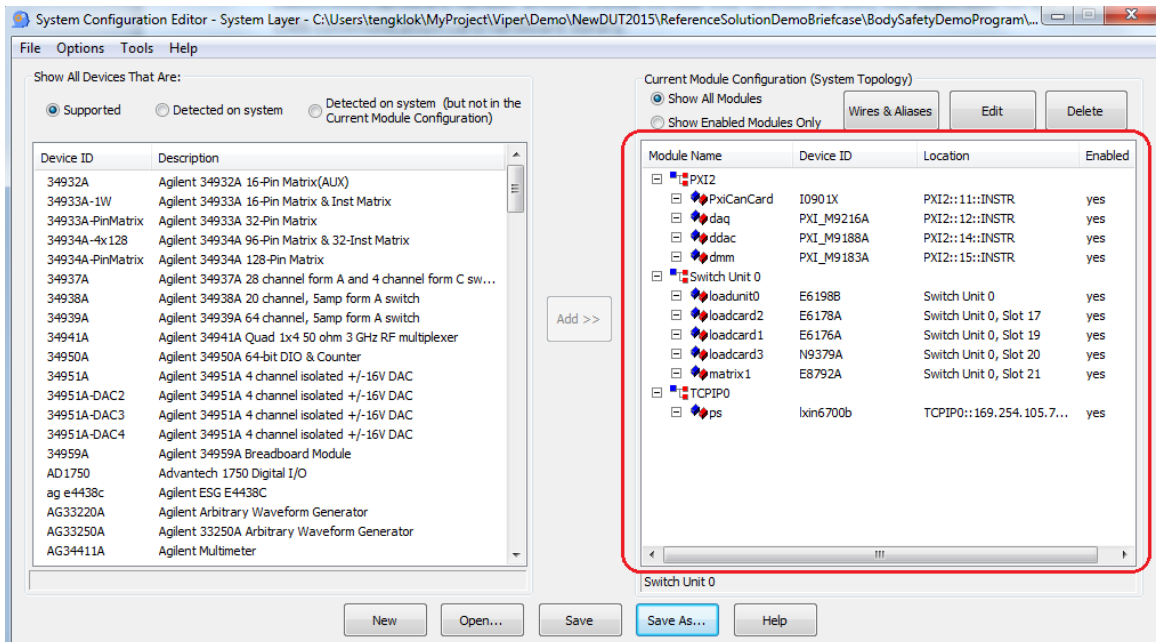
After test completion, clicking the “**Exit**” button will call the form closing event handler. It terminates the TestExec SL software and closes the hardware module that was opened via initialization.

BodySafetyDemoProgram Details

Instrument initialization

While running a testplan for the first time, TestExec SL calls the “Init” function to initialize each instance of a hardware module that uses the hardware handle; in other words the function “opens” a hardware module. That is why the first cycle of test always takes a longer time than the consecutive cycles of testing. Every TS-5000 hardware library implements the “Init” function to initialize the hardware module. While writing customized hardware handlers, this function should be included as well. This is applicable to the PXI CAN communication card hardware library.


In this body/safety demo program, “*TS8989System.ust*” is the system topology file which when loaded to the testplan, TestExec SL software will initialize “enabled” instrument during the first run. Launching the “*TS8989System.ust*” with System Configurator Editor as shown in the following page provides user a glance of which instruments are enabled for testing.



Running test in TestExec SL

User can run the demo program independent of the UI – running the testplan will execute the tests in it. Running the test in TestExec SL provides options for user to execute an entire testplan or only selected tests in the testplan. Running selected tests is a powerful feature for debugging a testplan. However, user needs to take note of the degree of safety and effectiveness, which is dependent on the setup and cleanup tasks associated with the tests.

Besides debugging, running tests in TestExec SL allows user to modify a testplan. Developer has full control adding/deleting test items, configuring instrument setup, and setting specification limits which are used during tester development. A developer who is working on test coverage optimization, or DUT design verification will help raise the value of TestExec SL software.

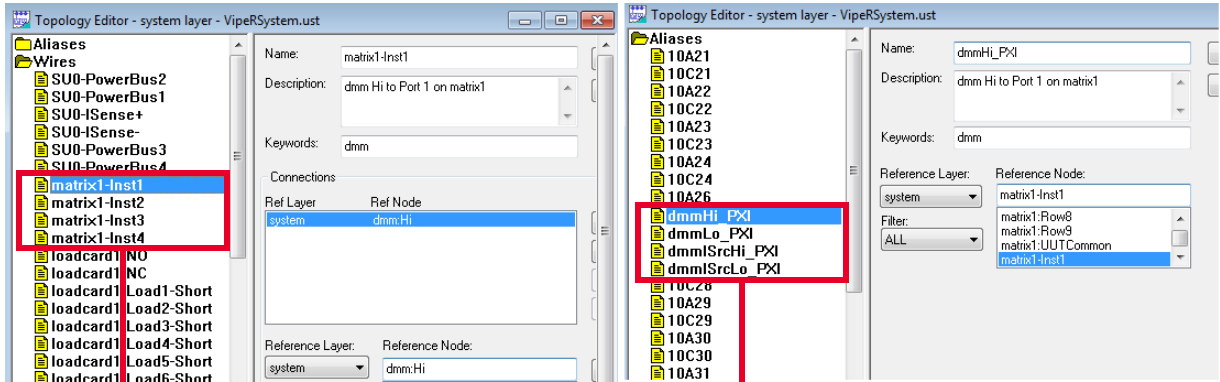
Refer to “*TS8989_ECU_Demo.tpa*” as the demo program testplan of body/safety ECU functional testing. Starting the test is just a simple step: Launching the testplan by clicking  in the toolbar, or selecting **Debug -> Run** in the menu bar to start running the entire testplan.

Switching topologies

“*TS8989System.ust*” is loaded as system topology file while “*pelTS8989 DUT uut.ust*” is loaded as DUT topology file for “*TS8989_ECU_Demo.tpa*”. Topology Files options specify these .ust files that are used for the system and UUT layers.

The “*TS8989System.ust*” system topology file not only contains instrumentation information, but also defines switchable nodes/connections. Module designation is used to determine the instruments installed in the system, and the Aliases and Wires are used to generate switching paths.

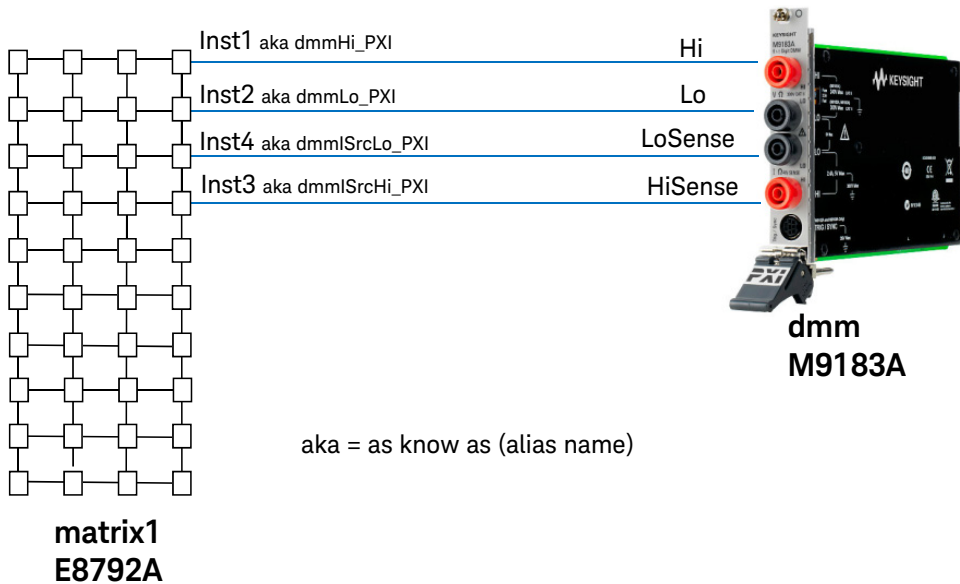
Take M9183A as an example – dmm and matrix1 are module names for M9183A digital multimeter and E8792A matrix card, named by the developer. Voltage/resistance +/- terminals of M9183A (dmm:Hi and dmm:Lo) are wired to E8792A instrument1 and instrument2 terminals (matrix1-Inst1 and matrix1-Inst2), meanwhile Sense/Current +/- terminals of M9183A (dmm:HiSense and dmm:LoSense) are wired to E8792A instrument3 and instrument4 terminals (matrix1-Inst3 and matrix1-Inst4) via dual banana connectors. Alternate/Alias names are populated to matrix1 Inst1 till Inst4 as dmmHi_PXI, dmmLo_PXI, dmmISrcHi_PXI, and dmmISrcLo_PXI in sequence.



matrix1 wires connection

matrix1 alias name

All these wires and aliases are generated automatically by the System Configurator Editor, representing TS-8989 hardware wiring from M9183A to E8792A. The diagram below gives you an idea on how information of dmm and matrix1 is translated into "TS8989System.ust" topology files.

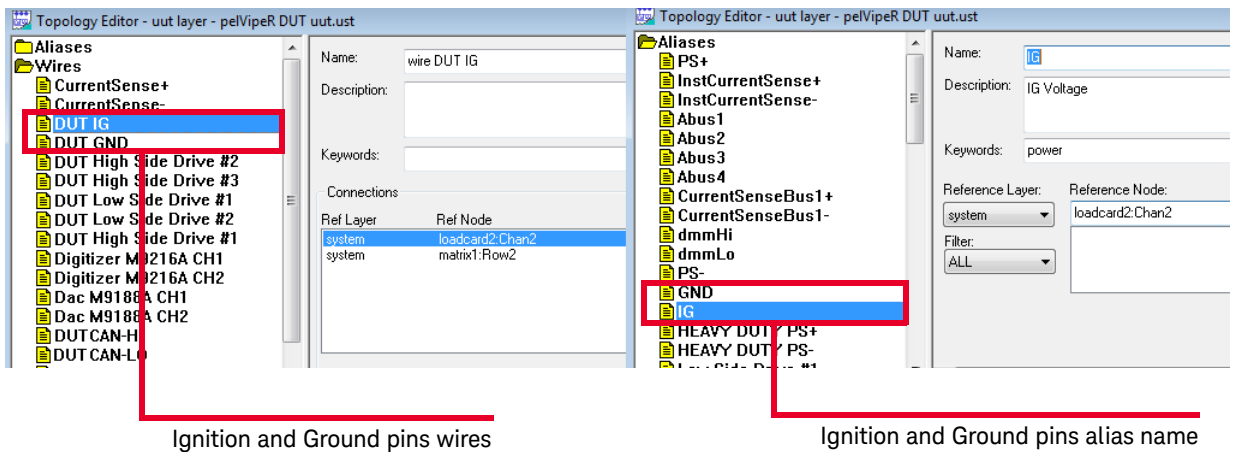


BodySafetyDemoProgram Details

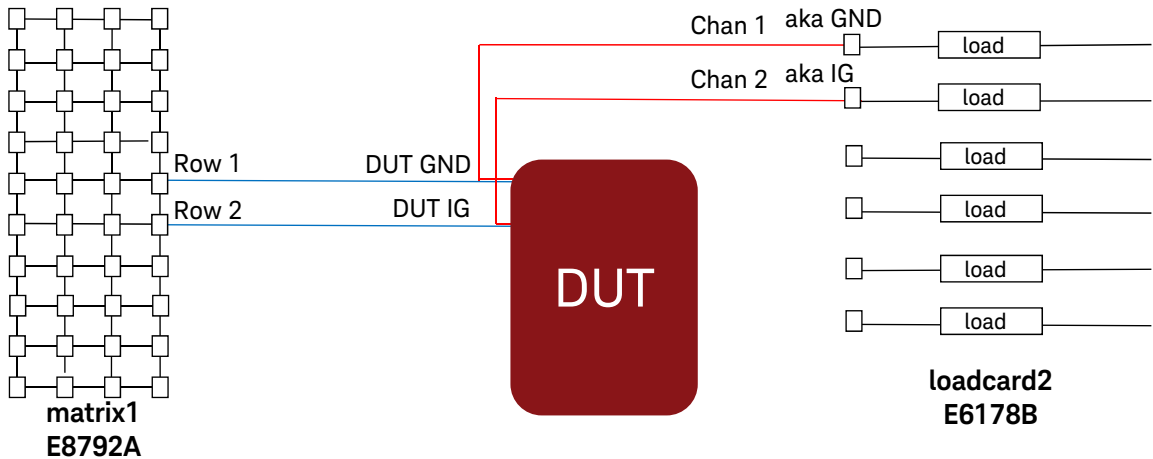
It is necessary to add or modify the “*TS8989System.ust*” wires/aliases if there are changes made on default system architecture, such as Load Card power buses configuration, and I-Sense wiring.

The “*pelTS8989 DUT uut.ust*” topology file specifies connections of all DUT nodes to the TS-8989. Module designation is usually not used in the UUT layer since the topology only consists of DUT information. Aliases and Wires are created by the developer to generate names and paths between DUT and TS-8989.

For example, with the Ignition and Ground pin of the DUT, the Ignition pin is named as “*DUT IG*” with the cabling connection to *matrix1:Row2* and *loadcard2:Chan2* (E6178B load card channel 2), while the Ground pin is named as “*DUT GND*” with the cabling connection to *matrix1:Row1* and *loadcard2:Chan1*. Both Ignition and Ground pins having alternate names as IG and GND respectively that are defined in Aliases designation.



Users have to plan their hardware resources accordingly and design their fixture/cabling connectivity to DUT. Wires and aliases in the UUT layer will then be created by the developer to indicate DUT nodes and their cabling connectivity. The diagram below shows the Ignition and Ground physical cabling fabricated for this demo, which translates into “*pelTS8989 DUT uut.ust*” topology files.

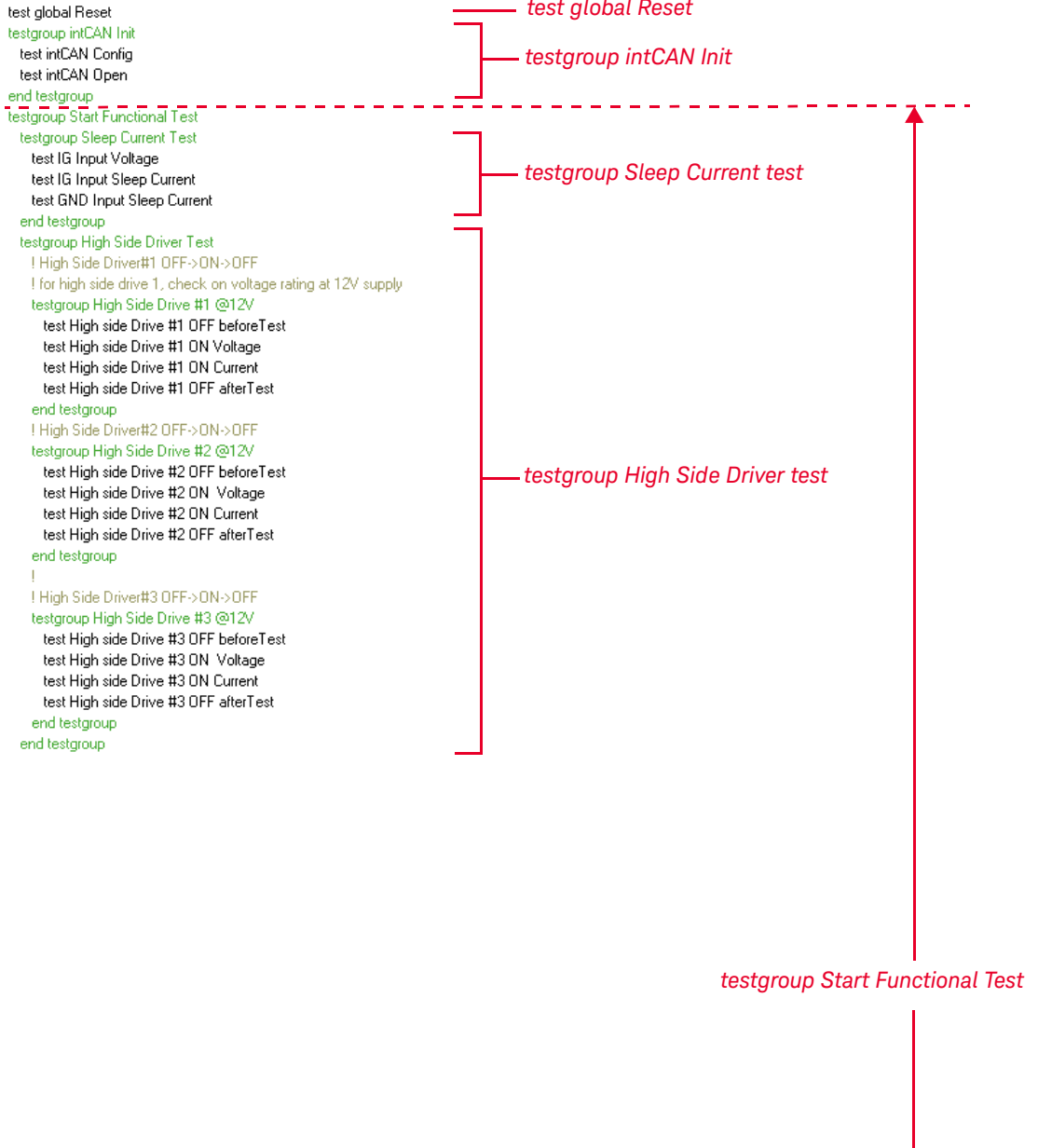


Connections between instruments and the DUT are not permanently “hard wired” but are controlled programmatically via “switching” to effectively utilize hardware resources. Both topology files define switchable connections and the wiring with a logical node name that is easy to use and remember. “*TS8989System.ust*” and “*pelTS8989 DUT uut.ust*” tell TestExec SL switching editor on available adjacent switching nodes, which help to simplify switching programming without the need of referring to complex routing architecture. The application note [Configuring Signal and Load Switching Using Keysight TestExec SL](#) – available on www.keysight.com/find/testexecsl will help you further understand how to use the TestExec SL switching manager feature.

Body/Safety ECU test

The demo program testplan “TS8989_ECU_Demo.tpa” runs a typical functional test for an ECU utilizing different Keysight hardware. The following test/testgroup was included in the Main sequence:

- *test global Reset*: A test that calls for reset of all instruments defined in the system topology.
- *testgroup intCAN Init*: Testgroup that configures CAN setting followed by switching on the communication until end of test.
- *testgroup Start Functional Test*: Testgroup that configures and switches on the power supply output to DUT IG and GND via load card. Functional testing starts after the power up of DUT; the testgroup consists of four sub-testgroups for the following tests:
 - *testgroup Sleep Current test*: Testgroup that measures the amount of current drawn by the DUT while the DUT is inactive.
 - *testgroup High Side Driver test*: Testgroup that validates the driver function by connecting equivalent model of loads with biasing on the driver. Key measurements are voltage/current drawn during on-off operation of the high side driver.
 - *testgroup Frequency Input test*: Testgroup that simulates the effect of sensors to the DUT input. Measurement is done by reading back the ECU response under DUT assisted test mode.
 - *testgroup Low Side Driver test*: Testgroup that validates the driver function by connecting equivalent model of inductive loads with biasing on the solenoid driver. Key measurements are voltage/current drawn during on off operations, as well as the transient time response of the solenoid driver.



BodySafetyDemoProgram Details

```
testgroup Frequency Input Test
test DAC Output Pulse Wave Frequency Inpu1&2
test DUT setup for Frequency Input
testgroup Frequency Input1
loop
test Analog Input1 Receive CAN MSG
exit if SequenceLocals.iCANRxSize > 0
end loop
test Analog1 Volt
test Analog1 Freq
end testgroup
testgroup Frequency Input2
loop
test Analog Input2 Receive CAN MSG
exit if SequenceLocals.iCANRxSize > 0
end loop
test Analog2 Volt
test Analog2 Freq
end testgroup
testgroup Frequency Phase Input1 - 2
loop
test Analog Input1 Receive CAN MSG 0x203
exit if SequenceLocals.iCANRxSize > 0
end loop
test Time in us between Analog1 Rising to Analog2 Rising
end testgroup
test DUT cleanup for Frequency Input
test DAC Reset Pulse Wave Frequency Inpu1&2
end testgroup
testgroup Low Side Driver Test
! Low Side Driver#1 OFF->ON->OFF
testgroup Low Side Drive #1
test Low side Drive #1 OFF beforeTest
test Low side Drive #1 ON Voltage
test Low side Drive #1 ON Current
test Low side Drive #1 OFF afterTest
test Low side Drive #1 Wave Capture Setup
test Low side Drive #1 Wave Capture Initiate
test Low side Drive #1 Wave Capture ON->OFF
waitthread NewWaitthread
test Low side Drive #1 WaveformDataWrite
test Low side Drive #1 Analyze VSaturation
test Low side Drive #1 Analyze V Peak
end testgroup
! Low Side Driver#2 OFF->ON->OFF
testgroup Low Side Drive #2
test Low side Drive #2 OFF beforeTest
test Low side Drive #2 ON Voltage
test Low side Drive #2 ON Current
test Low side Drive #2 OFF afterTest
test Low side Drive #2 Wave Capture Setup
test Low side Drive #2 Wave Capture Initiate
test Low side Drive #2 Wave Capture ON->OFF
waitthread NewWaitthread
test Low side Drive #2 WaveformDataWrite
test Low side Drive #2 Analyze VSaturation
test Low side Drive #2 Analyze V Peak
end testgroup
end testgroup
end testgroup
```

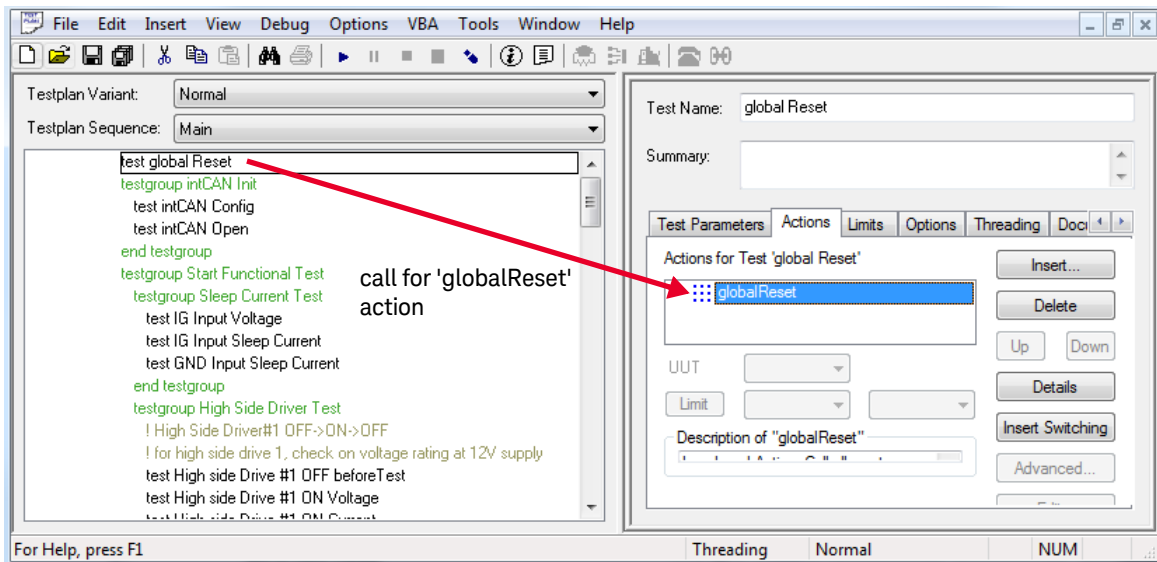
testgroup Frequency Input test

testgroup Low Side Driver test



test global Reset

This test does not perform any functionality checking but calling 'globalReset' action resets routines of instrument listed in the system topology "TS8989System.ust". The action calls for "Reset" of functions that are implemented in the hardware handler. Although TestExec SL automatically calls the "Reset" function at the startup of instrument initialization, the action "globalReset" was added to safeguard every instrument that has been set to its default state for each consecutive test. This action resets instruments before resetting switching hardware handlers to prevent any hot switching which may lead to deterioration of switching life cycles.



testgroup intCAN Init

The testgroup consists of two tests that call for customized actions to configure and open CAN port #1. High Speed CAN (ISO-11898) protocol standard is used to establish communication to the demo DUT with built-in ISO-11898 standard.

testgroup Start Functional Test

Like tests, testgroups can have actions associated with them. This lets each testgroup have an associated list of actions that do tasks before and after the tests inside the group. As mentioned earlier, the *testgroup Start Functional Test* contains four major testgroups, acting as outer group to apply a common set of startup for all testgroups and tests residing within.

'Switching' action is called to connect power supply terminals to DUT Ignition and Ground via E6178B relays with the following switching paths:

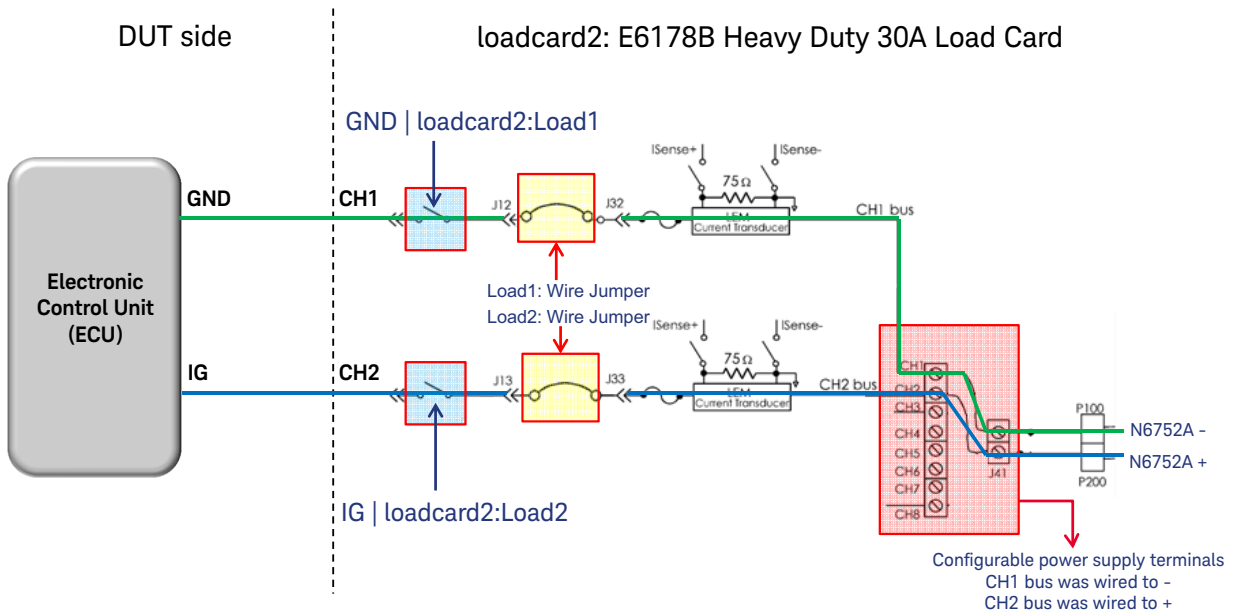
“

GND | loadcard2:Load1

IG | loadcard2:Load2

“

Load1 and Load2 are wire jumpers attached by the user to provide nearly zero resistance for the supply path.



Next, 'psProgVI' action is inserted with the following parameter set on modular power supply N6752A residing in N6700B:

'psProgVI'

ps = ps1-N6752A (select first channel of N6700B - N6752A)

voltage = 12.0 (set the voltage level at 12 V)

current = 8.0 (set current limit at 8 A)

connect = 1 (turn on power supply internal relay)

As per the configuration above, the power supply will then turn on to supply 12 V with current limit at 8 A to the demo DUT, after the E6178B relay is switched on.

'Switching' action also includes following paths to connect CAN terminals to DUT via E8792A relays. CAN terminals are routed through E8792A Aux pins rather than direct wiring so that the DUT can be disconnected from CAN for certain tests that require isolation from CAN terminals.

“

DUT CAN-HI | INTERLATIN CAN I0910X CAN-HI

DUT CAN-LO | INTERLATIN CAN I0910X CAN-LO

“

testgroup Sleep Current test

The testgroup consists of test items to verify voltage feed into the IG pin, as well as the amount of current drawn while the DUT is inactive.

Test IG Input Voltage calls for 'Switching' action to form the connection of IG and GND pins to dmm Hi and Lo respectively with the following paths:

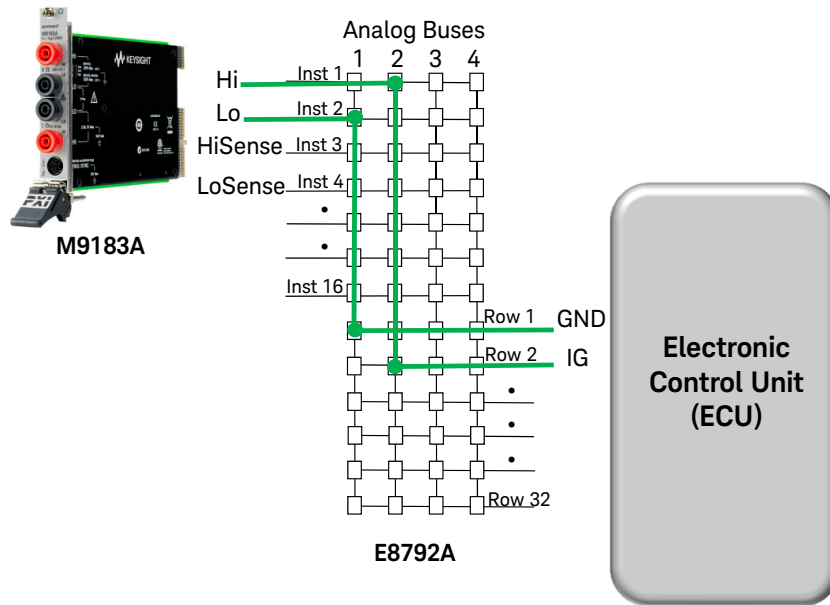
“

GND | Abus1 | dmmLo

IG | Abus2 | dmmHi

“

GND pin is routed through Analog Bus #1 while IG pin is routed through Analog Bus #2 to create the connection to dmm.



Voltage of IG with reference to GND will then be taken to verify the level of voltage supplied at IG using 'dmmMeasureDCV' action. The high level action created by Keysight system architect includes an option for configuration setup and return of measurement results by the M9183A:

'dmmMeasureDCV'

dmm = dmm (select M9183A)

AccuracySpeed = 1.0 (select medium speed with aperture time at 5000 μ s)

ExpectedReading = 20.0 (set expected result for dmm to select appropriate dynamic range)

MeasurementMode = 3 (select mode to configure the dmm and get measurement result)

Result = x.x (the return result in Volt)

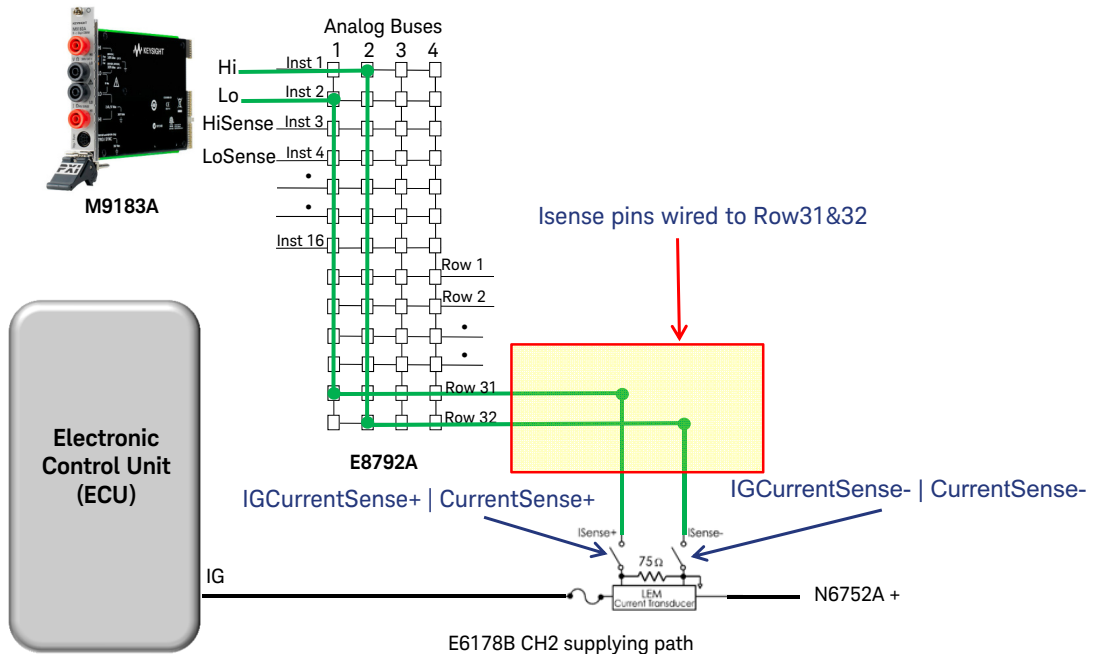
Test IG Input Sleep Current and **test GND Input Sleep Current** are the tests for current measurement at IG and GND paths respectively. E6178B features current transducers on supplying paths, inducing current from primary to secondary at the attenuation ratio of 1000:1.

The secondary current transducer is connected with a high-precision 75-Ohm resistor in series for current monitoring purposes. DUT cabling is designed with current sense from E6198B utility output wired to E8792A row 31 and row 32. Voltage drops across the 75-Ohm resistor can be measured with dmm using a similar method as in **Test IG Input Voltage**, but routed through row 31 and row 32. The diagram below illustrates the switching architecture of **test IG Input Sleep Current**, where the switching paths were added as:

“

```
IGCurrentSense- | CurrentSense- | Abus2 | dmmHi
IGCurrentSense+ | CurrentSense+ | Abus1 | dmmLo
```

“



The action used is 'dmmMeasureCurrent' where voltage measured will be converted into current using the equation: $I_{ch} = V_{drop} / 0.075$.

'dmmMeasureCurrent'

dmm = dmm (select M9183A)

AccuracySpeed = 1.0 (select medium speed with aperture time at 5000 μ s)

ExpectedCurrent = 0.5 (set expected result for dmm to select appropriate dynamic range)

MeasurementMode = 3 (select mode to configure the dmm and get measurement result)

SenseResistance = 0.075 (set sense resistor based on voltage-current conversion formula)

MeasuredCurrent = x.x (the return result in Ampere)

Test GND Input Sleep Current is repeated using the same methodology as **test IG Input Sleep Current**; the only dissimilarity is the current sense, which is switched at Channel 1 of E6178B, with GND connected to the negative terminal of power supply.

testgroup High Side Driver test

The testgroup consists of a few sub testgroups for a total of three high side drivers for functional checking. High side drivers are designed to control the desired output of an ECU. Take for instance, built in drivers of body control ECU controls of vehicle cargo lamps, wipers, door locks etc. as reaction from triggered switches input or sensors. In typical functional testing, resistors are modeled to connect to ECU drivers as emulation of automotive loads effect such as lighting and motors. The following will discuss about **testgroup High Side Drive #1 @12 V** only, elaborating tests that satisfy functional testing of resistive high side driver:

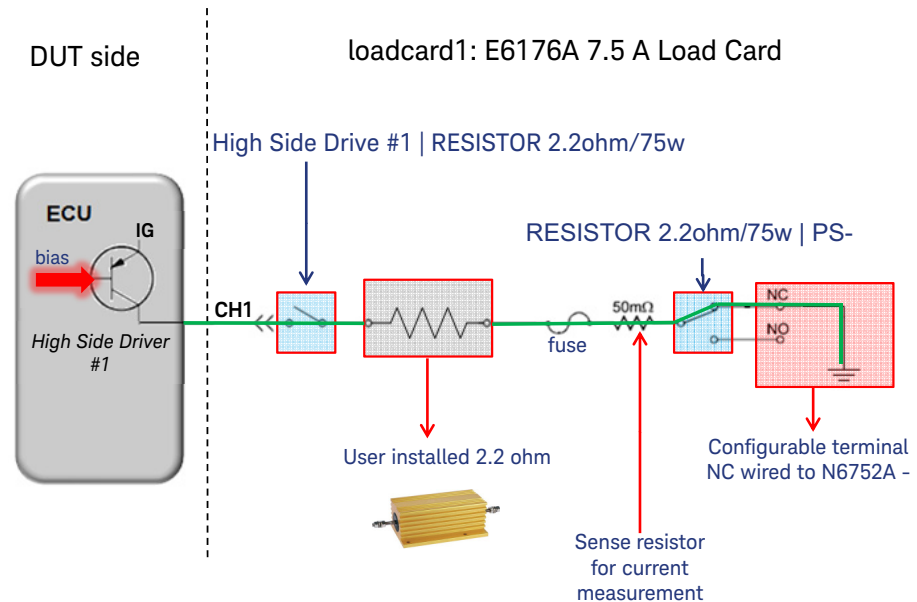
Test High side Drive #1 OFF beforeTest verifies the *off* state of driver #1 before biasing. Voltage level of driver output will be measured with the expected result of approximately 0 V.

Test High side Drive #1 ON Voltage and **test High side Drive #1 ON Current** are tests when driver #1 is turned on. CAN command is sent to bias driver #1 with its output pre-connected to a pulled-down 2.2 Ω resistor within its testgroup level. The following is the switching path on E6176A load card channel 1:

“

High Side Drive #1 | RESISTOR 2.2ohm/75w | PS-

“



The switching closes the Channel 1 single pole single throw (SPST) relay, and the other single pole double throw (SPDT) relay is kept at NC (normally closed) to connect the driver #1 output to 2.2 Ω resistor followed by E6198B power bus 1 (PB1), that is wired to the N6752A negative terminal (ground).

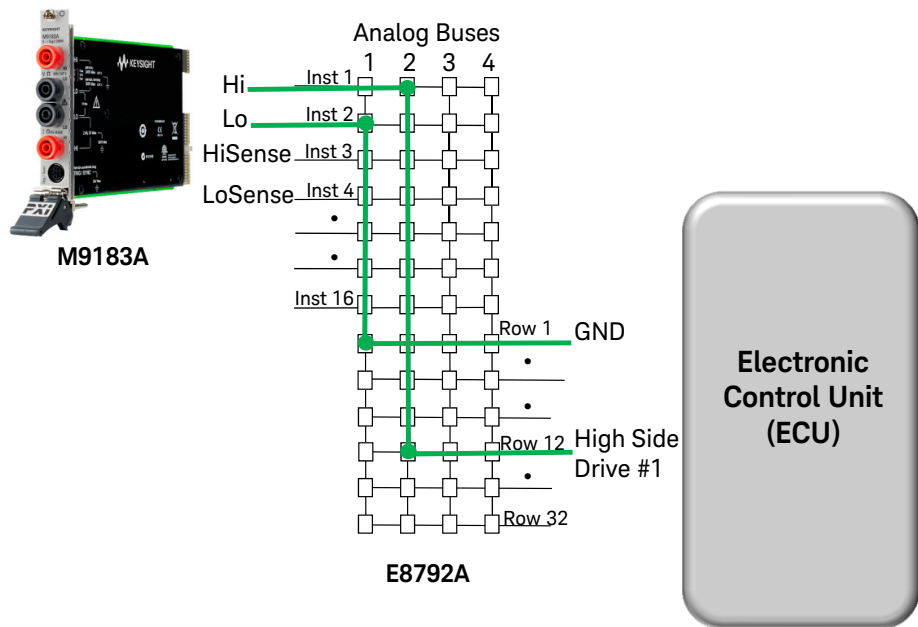
When the driver is turned on with biasing, voltage output is measured with matrix switching from driver output to dmm. The switching paths below are then connected, followed by 'dmmMeasureDCV' action which is called in sequence:

“

GND | Abus1 | dmmLo

High Side Drive #1 | Abus2 | dmmHi

“



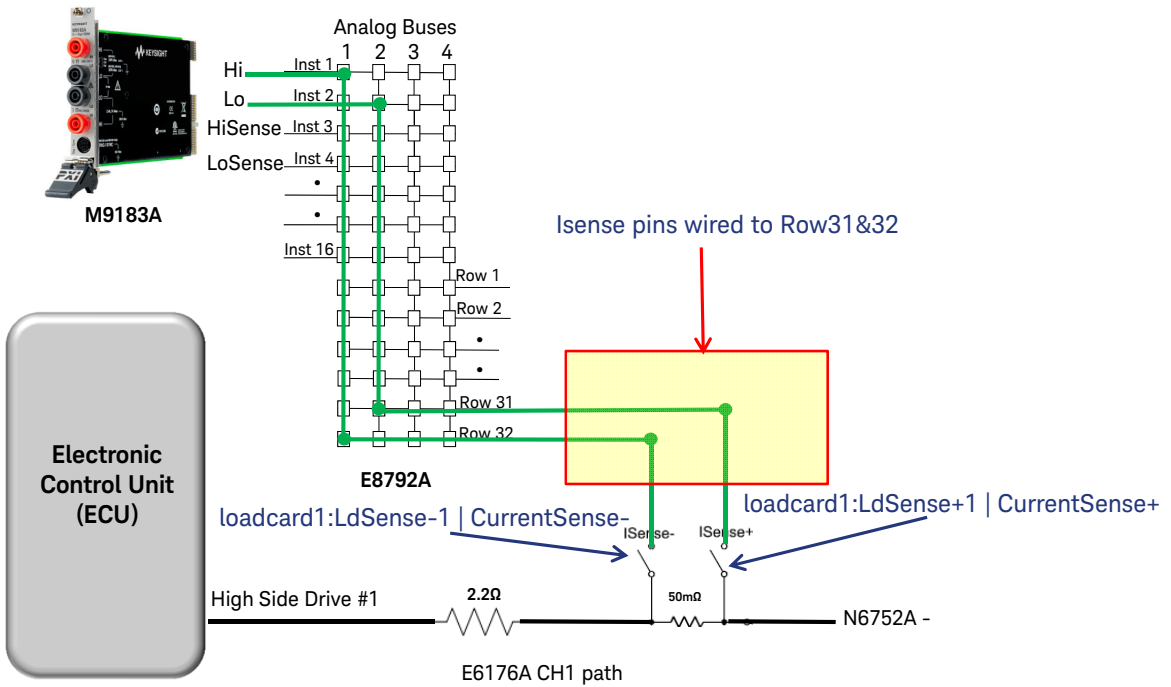
In the meantime, the E6176A load card features 50 mΩ sense resistors for current sensing. Similar to the measurement method of current transducer, voltage drop across the resistor will be obtained after the following switching connection is established:

“

loadcard1:LdSense-1 | CurrentSense- | Abus1 | dmmHi

loadcard1:LdSense+1 | CurrentSense+ | Abus2 | dmmLo

“



Similar to E6178B, voltage measurement with current conversion is achieved with the use of the action 'dmmMeasureCurrent', and a slightly different equation: $I_{ch} = V_{drop} / 0.05$ (Sense Resistance = 0.05). Both voltage and current tests will determine if the driver is working properly when biasing is induced.

Lastly, **test High side Drive #1 OFF afterTest** was created to verify the *off* state of driver #1 when CAN command was sent to halt driver biasing, where the driver output is expected to be nearly 0 V since the driver has been shut down. Tests will continue for the second and third drivers, with defined resistors 10 Ω and 20 Ω respectively, for examination on the drivers' quality.

testgroup Frequency Input test

Body/Safety ECU functionality highly relies on inputs from multiple sensors for monitoring purposes. Speed, position, force and etc. are sensors that are generally equipped in today's electronically-driven body/safety automotive components. Typically, Hall Effect sensors are used, and their voltage or current outputs vary in response to the change of magnetic field.

The testgroup elaborates how to use M9188A Dynamic D/A Converter, which generates multiple waveforms to the demo DUT for frequency input testing. The **testgroup Frequency Input test** applies common switching to turn on routing from M9188A signal channels and ground through E8792A Auxiliary pins:

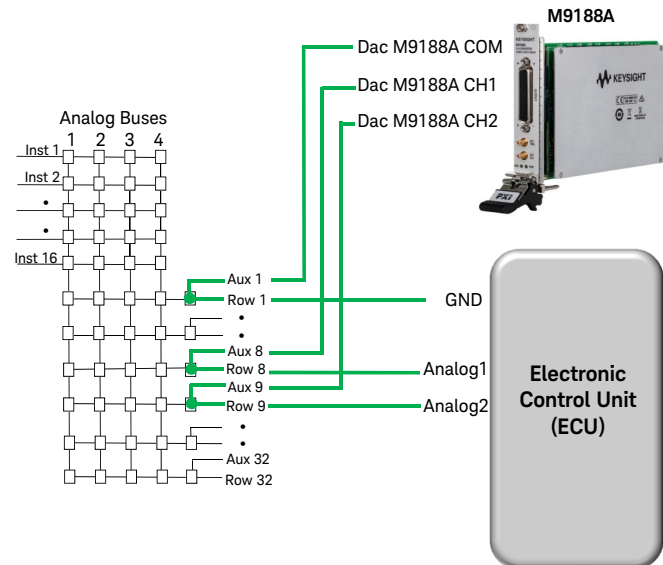
“

Analog1 | Dac M9188A CH1

Analog2 | Dac M9188A CH2

GND | DAC M9188A COM

“



Although M9188A includes built-in relays for isolation purposes, the E8792A Auxiliary design helps to disconnect the M9188A from DUT if needed, such as isolation of signal ground to common ground.

In this demo program, emulation of two Hall Effect sensors that monitor the rotation and speed of power windows are generated for the Window Lift Control ECU. The M9188A is programmed to generate two asymmetric square waves at 500 Hz with 180 degree phase shift. The sampling of TestExec SL actions in the hardware library of M9188A for waveform generation is as in **test DAC Output Pulse Wave Frequency Inpu1&2**.

'dac_ConfigStdWaveformSingleChannel'

DAC = ddac (select M9188A module)

Channel = 1 (configure on channel 1)

Output Mode = 0-Voltage Mode (select voltage mode)

Waveform Type = 2-Square Wave (select standard square waveform)

Amplitude = 10.0 (set voltage peak to peak at 10 V)

Offset = 5.0 (shift waveform up so that $V_{min} = 0$ V, $V_{max} = 10$ V)

Phase Shift = 0.0° (zero phase shift)

Frequency = 500.0 Hz (set frequency at 500 Hz)

Duty Cycle = 50.0% (specify duty cycle at 50%)

'dac_ConfigStdWaveformSingleChannel'

DAC = ddac (select M9188A module)

Channel = 2 (configure on channel 2)

Output Mode = 0-Voltage Mode (select voltage mode)

Waveform Type = 2-Square Wave (select standard square waveform)

Amplitude = 10.0 (set voltage peak to peak at 10 V)

Offset = 5.0 (shift waveform up so that $V_{min} = 0$ V, $V_{max} = 10$ V)

Phase Shift = 180.0° (specify 180° phase shift)

Frequency = 500.0 Hz (set frequency at 500 Hz)

Duty Cycle = 50.0% (specify duty cycle at 50%)

‘dac_ConfigOutputRelaySingleChannel’

DAC = ddac (select M9188A module)

Channel = 1 (configure on channel 1)

Output Relay State = 1 (enable output by turning its internal relay)

‘dac_ConfigOutputRelaySingleChannel’

DAC = ddac (select M9188A module)

Channel = 2 (configure on channel 2)

Output Relay State = 1 (enable output by turning its internal relay)

‘dac_InitTrigger’

‘dac_SendSoftwareTrigger’

Under *TEST*, or *DUT-Assisted* mode operation, signals applied can be queried from DUT via communications established. DUT under *TEST* mode is embedded with algorithm to identify applied signals, and stores its response or reading into dedicated registry/memory. The results are then been inquired through the established CAN communication link.

Testgroup Frequency Input1, **testgroup Frequency Input2**, and **testgroup Frequency Phase Input1 - 2** demonstrate example of tests to read ECU dedicated memory state, followed by mathematical calculation for raw data conversion into readable results. **Testgroup Frequency Input1** and **testgroup Frequency Input2** check voltage and frequency results of applied signals, which are 10 V at 500 Hz. While **testgroup Frequency Phase Input1 - 2** checks the phase differences of both applied signals at 180°.

testgroup Low Side Driver test

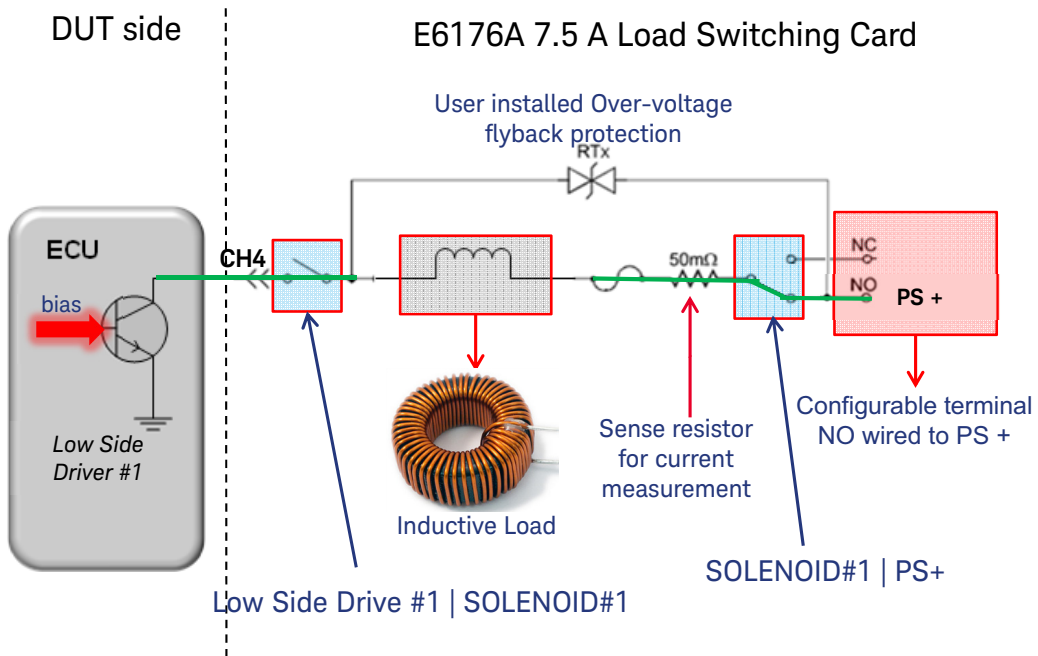
The testgroup consists of two sub-testgroups for the two low side drivers for functional checking. Like high side drivers, low side drivers are designed to control the desired output of an ECU. The test will demonstrate the effect of low side drivers using inductors, which typically represent models for automotive loads control such as solenoids drivers for actuators. The following section will discuss **testgroup Low Side Drive #1** only, which sufficiently elaborates the test methodology for inductive low side driver.

Test Low side Drive #1 OFF beforeTest is the test that verifies off state of driver #1 before biasing. Voltage level of driver output will be measured with the expected result of nearly 12 V, the supply level. **Test Low side Drive #1 ON Voltage** and **test Low side Drive #1 ON Current** are the tests that run when driver #1 is turned on. CAN command is sent to bias driver #1 with its output pre-connected to a pulled-up inductor within its testgroup level. The following is the switching path on E6176A load card channel 4:

“

Low Side Drive #1 | SOLENOID#1 | PS+

“



The switching closes Channel 4 single pole single throw (SPST) relay for inductor connectivity. In the meantime, single pole double throw (SPDT) relay is switched to NO (normally open) to establish connectivity of inductor to E6198B power bus 2, where N6752A positive terminal (PS+) is fed for pull-up purposes. When the driver is turned on with biasing, voltage output with its current drawn will be measured with matrix switching to dmm. **Test Low side Drive #1 ON Voltage** and **test Low side Drive #1 ON Current** are the tests that recap the methodology of voltage and current measurement as in High Side Drive #1.

The inductive nature of the solenoids causes considerable voltage flyback when the solenoid is turned off. Hence, other than measuring operation of low side driver, flyback characteristic is needed since it is a solenoid driver. In this dynamic way of testing, M9216A High Voltage Data Acquisition module was used to process the signal generated by the solenoid driver, to capture multiple characteristics of the solenoid driver behavior in a single test, such as saturation and flyback voltage.

Just like M9188A, M9216A is routed through E8792A Auxiliary relay for isolation purposes, the *testgroup Low Side Driver test* has closed its Auxiliary relays for M9216A connectivity with the following switching paths:

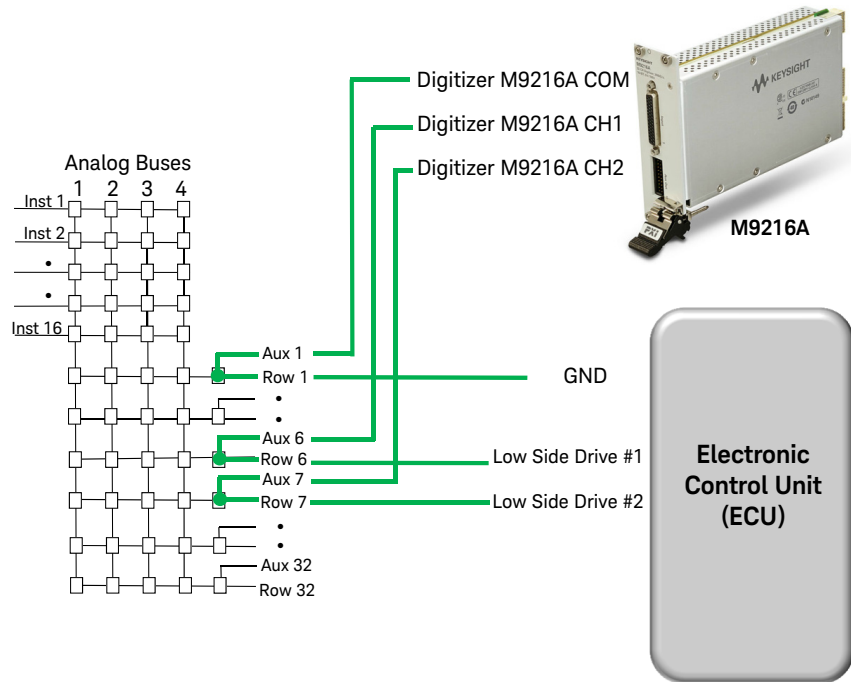
“

Low Side Drive #1 | Digitizer M9216A CH1

Low Side Drive #2 | Digitizer M9216A CH2

GND | Digitizer M9216A COM

“



Test Low side Drive #1 Wave Capture Setup is created to set up the M9216A. Here are the actions used in the hardware library of M9216A with their parameter settings:

‘daqIVIConfMeasurementMode’

Inst = daq (select M9216A)

Mode = 0 (set measurement mode as single measurement)

TrigSource = 0 (set trigger source as channel)

'daqlvSingleConflnputPort'

Inst = daq (select M9216A)

InputPort = 1 – Port 1 (select Channel 1 for Low Side Drive #1 measurement)

TrigLevel = 5.0 V (specifies 5 V threshold as triggering event)

Hyst = 0.0001 V (set hysteresis at 0.0001 V)

TrigEdge = 1-Negative (select negative trigger edge)

TrigCount = 1 (set only one single edges will start for waveform recording)

TrigDelay = 0.0 (set no delay time from the valid trigger event to the first point in the waveform record)

PreSampleCount = 1000 (set the number of samples to be collected before a trigger event recognized)

RecordSize = 32767 (set the total of samples to be acquired as full waveform)

TimeOut = 2.0 (set time out at 2 second in any case no triggering event detected)

Next, **test Low side Drive #1 Wave Capture Initiate** calls for action 'daqlvSingleWave' so that it can initiate the M9216A for any triggering event:

'daqlvSingleWave'

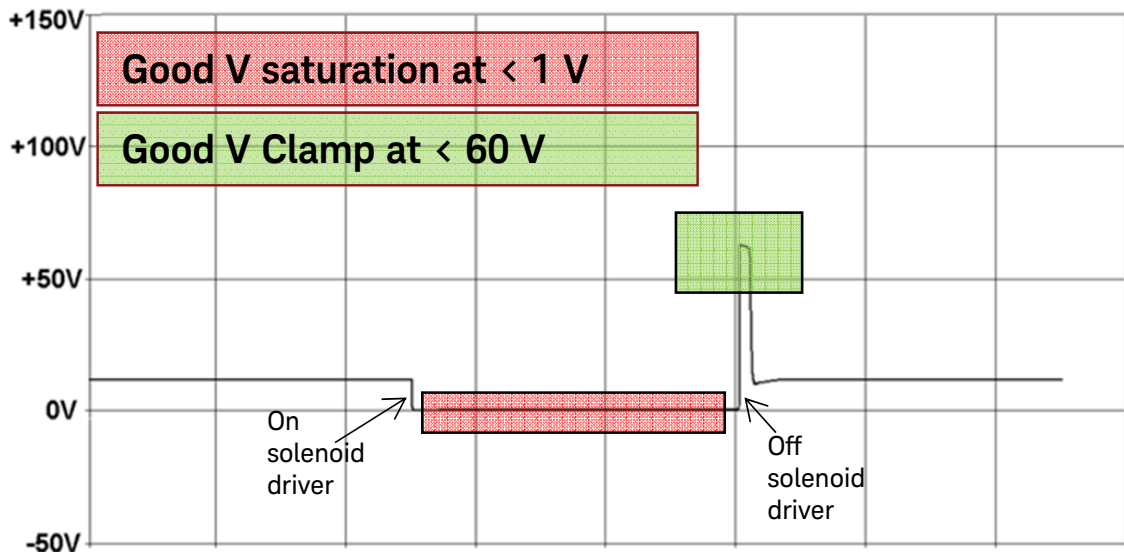
Inst = daq (select M9216A)

InputPort = 1 – Port 1 (select Channel 1 for measurement)

hiWave = SequenceLocals.waveform1 (keep returned data array result in TestExec SL symbol table)

In the meantime **test Low side Drive #1 Wave Capture ON->OFF** transmits the CAN command to switch on the solenoid driver, with a delay of 50 ms with the use of action 'agttcmiscDelayMsec' before another CAN command is sent to turn off the solenoid driver. Note that both **test Low side Drive #1 Wave Capture Initiate** and **test Low side Drive #1 Wave Capture ON->OFF** are threaded so that the M9216A is on standby for triggering. When the solenoid driver is turned off, low side drive #1 output voltage will drop and when it crosses over 5 V, M9216A Channel 1 will detect the triggering event and start its data acquisition in 250 kSa/sec sampling speed, followed by storing the waveform in symbol table SequenceLocal.waveform1.

Waveform array is then retrieved for data analysis. Typically there is voltage saturation during the on state and transient flyback voltage when solenoid drive is switched off. Saturation voltage is important for characterizing the health of the solenoid driver electronics. Typical saturation voltage values range from 0.5 - 1 V. On the other hand, flyback voltage output is measured to verify that the protective clamping circuit on ECU's solenoid driver is functioning correctly, 60 V is the maximum specification limit in this demo.



Test Low side Drive #1 WaveformDataWrite calling action 'WaveformDataWrite' to retrieve SequenceLocal.waveform1 and stored it as a text file (C:\My TxSL Files\waveform1.txt).

Next, both **test Low side Drive #1 Analyze VSaturation** and **test Low side Drive #1 Analyze V Peak** are created to read in waveform1.txt. Both voltage saturation level and flyback voltage are then analyzed with the following actions of AnalyseWaveform.dll library:

'AnalyseTextWaveformMinMax'

File = C:\My TxSL Files\waveform1.txt (read the waveform result from M9216A)

Presample Size = 0 (set zero pre-sample size for analysis from first data point)

Sampling Rate = 250000 (set data sampling rate as M9216A acquisition rate)

Start Time = 0.0 ms (set 0.0 ms as the start time of identification of pulse edge)

End Time = -1.0 ms (set -1 ms to automatically calculate end time at the end point of data)

Level = 5.0 V (set 5 V as pulse edge level start time)

Hysteresis = 0.01 V (set hysteresis level at 0.01 V)

Vmin Delay = 0.0 ms (set 0 ms delay after start time, time calculation will start after delay)

Average Time Vmin = 30.0 ms (specifies 30 ms for voltage averaging calculation)

Slope = 1 - Negative Triggering (set falling slope as the start of triggering)

Vmin = xxx (returned voltage saturation result)

'AnalyseTextPeakPulse'

File = C:\My TxSL Files\waveform1.txt (read the waveform result from M9216A)

Presample Size = 1000 (set 1000 pre-sample size for analysis from 1001 data points)

Sampling Rate = 250000 (set data sampling rate as M9216A acquisition rate)

Start Time = 0.0 ms (set 0.0 ms as the start time of identification of pulse edge)

End Time = -1.0 ms (set -1 ms to automatically calculate end time at the end point of data)

Hysteresis = 0.01 V (set hysteresis level at 0.01 V)

Average Time = 0.0 ms (specifies 0 ms for voltage peak searching without averaging)

Peak Voltage = xxx (returned voltage flyback result)

The test will be carried on for the second solenoid driver, with its defined inductor for examination on the driver quality, marking the end of the functional test.

Finally, demo program is completed after execution of AfterMain sequence, shutting off the power supply and CAN communication session.

Closing the program

TestExec SL calls “close” function to each instance of a hardware module while testplan is closed. This function closes a hardware module opened with the “Init” function. Every TS-5000 hardware handler implements this function with dedicated codes to close the particular instrument, freeing or deleting any memory associated with a structure created with the ‘Init’ function. Codes that the user writes to implement this function should also include this function to close the hardware module.

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

This information is subject to change without notice. Always refer to the English version at the Keysight website for the latest revision.

© Keysight Technologies 2016
Edition 1, Feb 1, 2016

Printed in Malaysia



U8989-90005

www.keysight.com

