Developer's Guide

Keysight RF PA/FEM Characterization & 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.

KEYSIGHT TECHNOLOGIES

Notices

Copyright Notice

Manual Part Number

Y1299-90007

Published By

Keysight Technologies
Ground Floor and Second Floor, CP-11
Sector-8, IMT Manesar – 122051
Gurgaon, Haryana, India

Edition

Edition 2, July, 2015

Regulatory Compliance

This product has been designed and tested in accordance with accepted industry standards, and has been supplied in a safe condition. To review the Declaration of Conformity, go to http://www.keysight.com/go/conformity.

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 OF 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.

KEYSIGHT TECHNOLOGIES DOES NOT WARRANT THIRD-PARTY SYSTEM-LEVEL (COMBINATION OF CHASSIS, CONTROLLERS, MODULES, ETC.) PERFORMANCE, SAFETY, OR REGULATORY COMPLIANCE, UNLESS SPECIFICALLY STATED.

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.

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.

Safety Notices

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.

The following safety precautions should be observed before using this product and any associated instrumentation.

This product is intended for use by qualified personnel who recognize shock hazards and are familiar with the safety precautions required to avoid possible injury. Read and follow all installation, operation, and maintenance information carefully before using the product.

WARNING

If this product is not used as specified, the protection provided by the equipment could be impaired. This product must be used in a normal condition (in which all means for protection are intact) only.

The types of product users are:

- Responsible body is the individual or group responsible for the use and maintenance of equipment, for ensuring that the equipment is operated within its specifications and operating limits, and for ensuring operators are adequately trained.
- Operators use the product for its intended function. They must be trained in electrical safety procedures and proper use of the instrument. They must be protected from electric shock and contact with hazardous live circuits.
- Maintenance personnel perform routine procedures on the product to keep it operating properly (for example, setting the line voltage or replacing consumable materials). Maintenance procedures are described in the user documentation. The procedures explicitly state if the operator may perform them. Otherwise, they should be performed only by service personnel.
- Service personnel are trained to work on live circuits, perform safe installations, and repair products. Only properly trained service personnel may perform installation and service procedures.

WARNING

Operator is responsible to maintain safe operating conditions. To ensure safe operating conditions, modules should not be operated beyond the full temperature range specified in the Environmental and physical specification. Exceeding safe operating conditions can result in shorter lifespans, improper module performance and user safety issues. When the modules are in use and operation within the specified full temperature range is not maintained, module surface temperatures may exceed safe handling conditions which can cause discomfort or burns if touched. In the event of a module exceeding the full temperature range, always allow the module to cool before touching or removing modules from chassis.

Keysight products are designed for use with electrical signals that are rated Measurement Category I and Measurement Category II, as described in the International Electrotechnical Commission (IEC) Standard IEC 60664. Most measurement, control, and data I/O signals are Measurement Category I and must not be directly connected to mains voltage or to voltage sources with high transient over-voltages. Measurement Category II connections require protection for high transient over-voltages often associated with local AC mains connections. Assume all measurement, control, and data I/O connections are for connection to Category I sources unless otherwise marked or described in the user documentation.

Exercise extreme caution when a shock hazard is present. Lethal voltage may be present on cable connector jacks or test fixtures. The American National Standards Institute (ANSI) states that a shock hazard exists when voltage levels greater than 30V RMS, 42.4V peak, or 60VDC are present. A good safety practice is to expect that hazardous voltage is present in any unknown circuit before measuring.

Operators of this product must be protected from electric shock at all times. The responsible body must ensure that operators are prevented access and/or insulated from every connection point. In some cases, connections must be exposed to potential human contact. Product operators in these circumstances must be trained to protect themselves from the risk of electric shock. If the circuit is capable of operating at or above 1000V, no conductive part of the circuit may be exposed.

Do not connect switching cards directly to unlimited power circuits. They are intended to be used with impedance-limited sources. NEVER connect switching cards directly to AC mains. When connecting sources to switching cards, install protective devices to limit fault current and voltage to the card.

Before operating an instrument, ensure that the line cord is connected to a properly-grounded power receptacle. Inspect the connecting cables, test leads, and jumpers for possible wear, cracks, or breaks before each use.

When installing equipment where access to the main power cord is restricted, such as rack mounting, a separate main input power disconnect device must be provided in close proximity to the equipment and within easy reach of the operator.

For maximum safety, do not touch the product, test cables, or any other instruments while power is applied to the circuit under test. ALWAYS remove power from the entire test system and discharge any capacitors before: connecting or disconnecting cables or jumpers, installing or removing switching cards, or making internal changes, such as installing or removing jumpers.

Do not touch any object that could provide a current path to the common side of the circuit under test or power line (earth) ground. Always make measurements with dry hands while standing on a dry, insulated surface capable of withstanding the voltage being measured.

The instrument and accessories must be used in accordance with its specifications and operating instructions, or the safety of the equipment may be impaired.

Do not exceed the maximum signal levels of the instruments and accessories, as defined in the specifications and operating information, and as shown on the instrument or test fixture panels, or switching card.

When fuses are used in a product, replace with the same type and rating for continued protection against fire hazard.

Chassis connections must only be used as shield connections for measuring circuits, NOT as safety earth ground connections.

If you are using a test fixture, keep the lid closed while power is applied to the device under test. Safe operation requires the use of a lid interlock.

Instrumentation and accessories shall not be connected to humans.

Before performing any maintenance, disconnect the line cord and all test cables.

To maintain protection from electric shock and fire, replacement components in mains circuits – including the power transformer, test leads, and input jacks – must be purchased from Keysight. Standard fuses with applicable national safety approvals may be used if the rating and type are the same. Other components that are not safety-related may be purchased from other suppliers as long as they are equivalent to the original component (note that selected parts should be purchased only through Keysight to maintain accuracy and functionality of the product). If you are unsure about the applicability of a replacement component, call an Keysight office for information.

WARNING

No operator serviceable parts inside. Refer servicing to qualified personnel. To prevent electrical shock do not remove covers. For continued protection against fire hazard, replace fuse with same type and rating.

PRODUCT MARKINGS:

The CE mark is a registered trademark of the European Community.

Australian Communication and Media Authority mark to indicate regulatory compliance as a registered supplier.

ICES/NMB-001
ISM GRP.1 CLASS A

This symbol indicates product compliance with the Canadian Interference-Causing Equipment Standard (ICES-001). It also identifies the product is an Industrial Scientific and Medical Group 1 Class A product (CISPR 11, Clause 4).

KCC-REM-ATi-BLMODSF01

South Korean Class A EMC Declaration. This equipment is Class A suitable for professional use and is for use in electromagnetic environments outside of the home. A 급 기기 (업 무 용 방 송 통 신 기 자 재) 이 기 기 는 업 무 용 (A 급) 전 자 파 적 합 기 기 로 서 판 매 자 또 는 사 용 자 는 이 점 을 주 의 하 시 기 바 라 며 , 가 정 외 의 지 역 에 서 사 용 하 는 것 을 목 적 으 로 합 니 다 .

This product complies with the WEEE Directive marketing requirement. The affixed product label (above) indicates that you must not discard this electrical/electronic product in domestic household waste. **Product Category**: With reference to the equipment types in the WEEE directive Annex 1, this product is classified as “Monitoring and Control instrumentation” product. Do not dispose in domestic household waste. To return unwanted products, contact your local Keysight office, or for more information see http://about.keysight.com/en/companyinfo/environment/takeback.shtml.

This symbol indicates the instrument is sensitive to electrostatic discharge (ESD). ESD can damage the highly sensitive components in your instrument. ESD damage is most likely to occur as the module is being installed or when cables are connected or disconnected. Protect the circuits from ESD damage by wearing a grounding strap that provides a high resistance path to ground. Alternatively, ground yourself to discharge any built-up static charge by touching the outer shell of any grounded instrument chassis before touching the port connectors.

This symbol on an instrument means caution, risk of danger. You should refer to the operating instructions located in the user documentation in all cases where the symbol is marked on the instrument.

This symbol indicates the time period during which no hazardous or toxic substance elements are expected to leak or deteriorate during normal use. Forty years is the expected useful life of the product.

CLEANING PRECAUTIONS:

WARNING

To prevent electrical shock, disconnect the Keysight Technologies instrument from mains before cleaning. Use a dry cloth or one slightly dampened with water to clean the external case parts. Do not attempt to clean internally. To clean the connectors, use alcohol in a well-ventilated area. Allow all residual alcohol moisture to evaporate, and the fumes to dissipate prior to energizing the instrument.

Contents

Introduction

This document describes the required procedure to build and modify the Keysight RF PA/FEM Characterization and Test Demo Program (PA Demo Program). Executable versions of this program are included with all of the PA Reference Solution Demo Systems. Source code for the PA Demo Program is available from your Keysight representative.

Version 3.0 of the PA Demo Program incorporates support for additional modules and updates to the test procedures.

The following are the initial features in Version 1.0:

- Support for M9391A PXIe Signal Analyzer
- Support for M9381A PXIe Signal Generator
- RF Measurements including:
  - Power acquisition and FFT acquisition measurements for servo loop, power measurements and ACPR
  - EVM using X-Series Measurement Applications
- Envelope Tracking feature

The following are the new features in Version 2.0:

- Support for M9393A PXIe Signal Analyzer
- Support for M937XA PXIe Vector Network Analyzer
- New Digital Pre-distortion (DPD) and improved Envelope Tracking (ET) Features:
  - Integration of N7614B Power Amplifier Test Suite for DPD and ET characterization
  - Integration of Signadyne model SD-PXE-AOU-H3353-1G PXI Arbitrary Waveform Generator
  - Automated VSG/AWG alignment procedure using ACPR measurement
  - Design Verification Test Demonstration User Interface
- Refactoring of PowerAmpTestLib dll:
  - Split into multiple libraries supporting different instrument types and data handling functions
  - Addition of Interface classes to support multiple versions of instrument types
    - PXIe VSG/VSA
    - Envelope Tracking AWG
- Improvements to RF Measurement Techniques:

    - Combine power acquisition and FFT acquisition measurements for servo loop, power measurements and ACPR into single methods
    - Move calculation of channel power from FFT data into a single method
    - Improve power calculations for FFT acquisition mode to better match measurements from power acquisition mode
    - Split EVM measurements into “initialize” and “fetch” methods to allow additional driver based measurements to be performed while EVM is calculated in X-Series apps
    - Support for sub 1 frame waveforms generated from the latest versions of Signal Studio applications for WCDMA, LTE-FDD and LTE-TDD

The following are the new features in Version 3.0:

- Support for Digital Source Measure PXI module, M9195A, for RFFE interface to the DUT
- Support for Measurement Acceleration Module, M9451A, for FPGA based DPD and ET analysis and signal generation
- Refactoring of Test Code:
    - Created interface class for DPD/ET calculations. Instances for this interface include software only calculations using the N7614B software and accelerated calculations using the M9451A Measurement Acceleration Module. A “null” instance was also created to allow running test sequences that include DPD and ET when simulating the VSG and VSA.
    - Created Interface class for digital I/O to allow use of the new M9195A Digital Source Measure PXI module or the SignalCraft Scout USB module. Null instance was also created to be used when no DIO module is available
    - Simplification of code in DPD/ET sequence by using methods in DPD/ET interface class. Also DPD/ET sequence now supports multiple standards in addition to LTE, including WCDMA and WLAN. To support this, DPD/ET test sequence no longer includes EVM measurements from X-Apps, only Delta EVM from the DPD/ET interface
- Other Program improvements
    - Automatic detection of VISA resource strings for all instruments other than VSG and VSA. The VSG and VSA use IVI configuration store saved configurations, which are automatically loaded into the PA Demo GUI. For other resources, the VISA Resource String fields have been removed from the GUI. Non-PXI instruments which support multiple interfaces, such as the 33522B AWG and N6700 DC Power mainframe, will be supported for UBS and GPIB. LAN is not supported in the PA Demo Program.

- User Waveforms in the DPD/ET tests. The user can now select a waveform to use with the DPD test sequence. Supported waveform types include Signal Studio, MATLAB, Binary Integer and CSV Text files.

Driver Installation

Several software and driver packages are required to build and run the PA demo program. The program uses the released versions of the IVI drivers. Refer to the *PA Demo Program User's Guide* for information on the required drivers and installation process

The PA demo program includes a “ReadMe File Versions” text file that documents the version of the drivers referenced in the program. If later versions of the drivers are used, it may be required to update the references in the program to the latest version.

The software packages should be loaded in the order shown.

The M90XA and N7614B software packages and the M9451A Measurement Acceleration Module require licenses to run. Contact your Keysight representative for these licenses.

M90XA X-Series Measurement Applications Overview

The M90XA X-Series Measurement Applications software (X-Series apps) consists of two primary elements. The first is a version of the X-Series apps that can be used with the M9391A PXIe VSA and M9393A PXIe Performance VSA. This runs as an application on the PC that also includes the PXIe VSA modules connected via PCI Express. This can be a PXIe embedded controller or a PC connected to the PXIe mainframe using the M9021A interface card. The second element is the M9000 Resource Manager that enables the modular VSA to be used by the X-Series app software and directly by the IVI driver without having to close and reopen a session to the hardware. This capability allows the use of the driver for high speed operation of items including ACPR measurement and power servo routines and to use the X-Series apps software for measurements such as EVM.

In the shared usage mode, IVI driver commands are sent to the modular VSA IVI driver, as they would be, in other programs that only use the IVI driver without the M90XA software, and SCPI commands are sent to the X-Series apps software as they would to the X-Series apps running in an MXA or PXA.

There are new commands to communicate with the Resource Manager to switch between modes. At a high level, the Resource Manager “locks” the modular hardware to either the IVI driver or to the X-Series app software. To switch the hardware from one mode to the other, the hardware must be “unlocked” from the first program before the second program can use it. The X-Series app software automatically checks out the hardware before each measurement and then checks the hardware back after the measurement is complete. For the IVI driver, you must send a command to the M9000 software to check out and check back in. This process is shown in the example code.

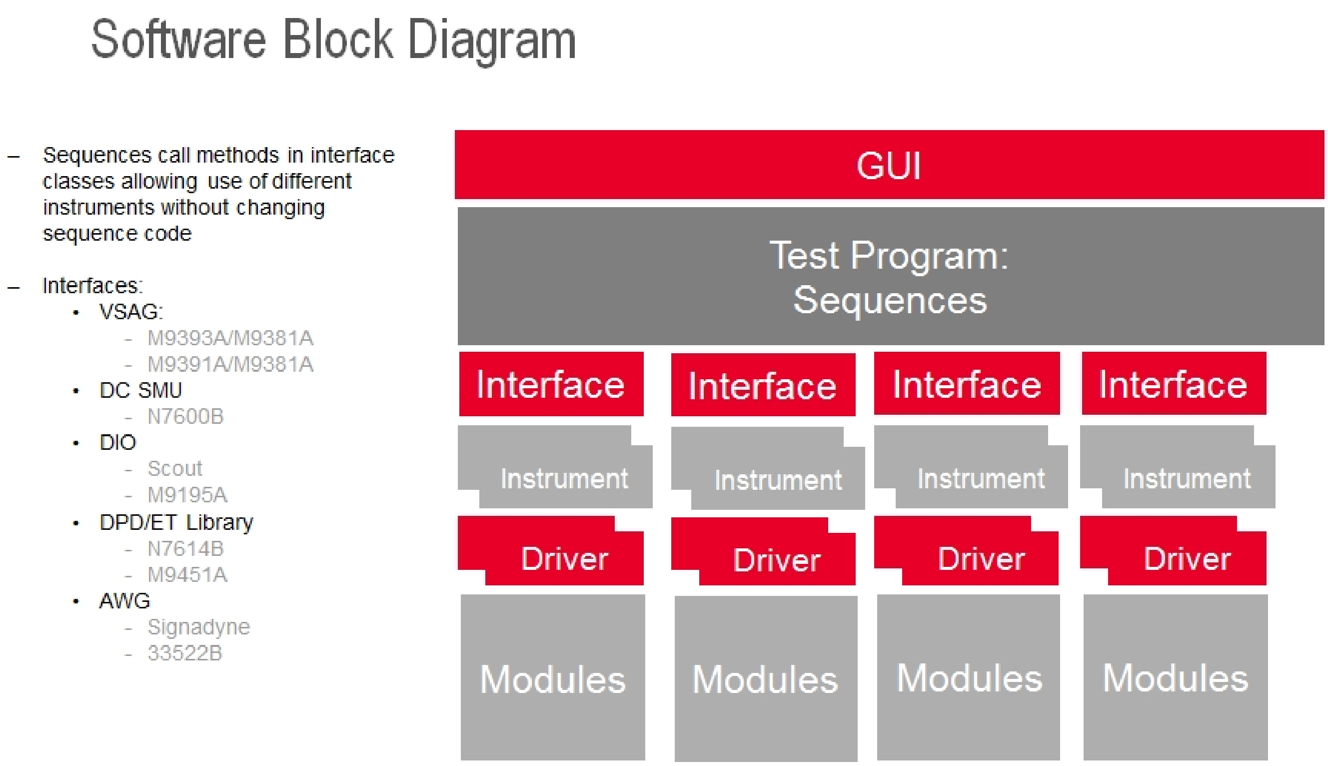
Keysight RF PA/FEM Characterization and Test Demo Program Overview

The PA Demo Program can be configured from the front panel to allow use for testing and collecting data under a large number of conditions. You may use this for evaluations by using the executable version of the program. A User's Guide is available to assist you.

In some cases there may be a need to modify the PA Demo Program or include some of the code from the demo program for use with your own test software. The demo program is also useful as a guide for best practices for various measurements.

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

The Demo program is a Visual Studio 2013 C# solution containing several projects. The following diagram shows the high level structure of the software:



The Program contains three main layers of software: the user interface, the test program and the instrument interfaces:

- User Interface Projects:
  - PowerAmpDemoGUI: The main user interface for the demo program
  - Vsag_Monitors: Interactive GUIs for the IVI drivers for PXIe VSG and PXIe VSA

    - DVT_GUI: Interactive user interface to demonstrate design validation tests
- PowerAmpDemoProgram: The test sequence for the specific device
- Instrument Control Projects:
    - IVsag: Interface class for the PXIe VSG and PXIe VSA
        - Vsag_M9391A: PXIe VSA and PXIe VSG control using the M9391A and M9381A
        - Vsag_M9393A: PXIe VSA and PXIe VSG control using the M9393A and M9381A
    - IetArb: Interface class for arbitrary waveform generator used for envelope tracking signal generation
        - Arb_33522B: ARB control for the Keysight 33522B waveform generator
        - Arb_Signadyne: ARB control for the Signadyne xxx waveform generator
        - Arb_Null: Null class used when no AWG is present in the system
    - IDcSmu: Interface class for DC power supply
        - DcSmu_N6700: Control for Keysight N6700 DC Power Supply
        - DcSmu_Null: Null class used when no DC Power supply is present in the system
    - IDio: Interface class for DIO modules used for sending RFFE commands to DUT
        - Dio_M9195A: RFFE Control using the M9195A Digital Source Measure PXI module
        - Dio_Scout: RFFE Control using the SignalCraft Scout USB DIO Module
        - DIO_Null: Null class used when no DIO module is present in the system
    - IDpdEt: Interface class for DPD and ET Signal Generation and Analysis
        - DpdEt_N7614B: DPD/ET using the N7614B Signal Studio for PA Test software package via the programming API to that software
        - DpdEt_M9451A: DPD/ET using the M9451A Measurement Acceleration Module. N7614B is still used for some functions in this class
        - DpdEt_Null: Null class to be used when the VSG and VSA are simulated. The simulated data from the VSG will cause errors with the N7614B class, thus this class is used in simulation
    - DataLog: Data logging methods used to save all test data to files and to display results to GUI

– Vna: Control for Keysight M937X PXI Vector Network Analyzer

Instrument Control Overview

In the first release of the PA Demo Program, all of the instrument control was in a single project, PowerAmpTestLib. The driver sessions and most of the properties in the TestLib were exposed so that they can be modified in either the TestProgram or GUI. In the second release, support was added for the M9393A PXIe VSA. Since the M9391A and M9393A use separate IVI drivers, the previous approach needed to be modified as the driver functions could not be called directly from the TestProgram or GUI without having case statements at every call. To resolve this issue, all of the methods that call the PXIe VSG and PXI VSA IVI drivers were moved into an interface class. The TestProgram and GUI now call methods and set properties that are defined in the interface class and no longer have direct access to the IVI drivers. This approach allows a single copy of a test sequence to work with both the M9391A and M9393A and any future instruments that are added to the PA Demo Program. The second release also applied the interface approach to the arbitrary waveform generator used to create envelope tracking stimulus.

In the third release of the program, the interface approach was extended to cover the DIO module, DPD/ET Analysis library and the DC SMU.

In version 1.0, the TestProgram and PowerAmpTestLib objects were created when the GUI program was started. In version 2.0 and 3.0, the TestProgram is still created when the GUI program starts but the instrument objects are not created until the "Init" button is pressed as the GUI is used to select which instruments will be used.

For all the instrument types except of the PXIe VSG/VSA, all of the code to control the instrument is included in a single source code file. For the PXIe VSA/VSG, there are several files, due to the large amount of code for these instruments.

The PXIe VSA/VSG project includes the following files:

– Vsag_M939xA.cs: Definition of the member variables used in the project and the constructor for the class, as well as the method to launch and close the PXIe VSA and VSG monitor windows.
– Initialize_M939xA.cs: Includes the startup, closing and state control for the PXIe VSG, VSA and X-Series app software. Also contains code for configuring the triggers for the PXIe VSG, VSA and X-Series apps and routing the triggers on the PXI backplane.
– RF_Measurements_M939xA.cs: Includes methods for setup and measurement of PA tests using the AgM938x and AgM939x IVI drivers, and the X-Series app software. Some of these routines are common across all standards, such as the Input Power Servo routine. Some of the routines are specific to a specific standard, such as the ACPR routine for LTE.
– RF_Waveforms_M939xA.cs: Methods for loading and generating RF waveforms for the VSG.

- Loss_Calibration_M939xA.cs: Methods for applying a correction factor to the input signal level and measured output levels of the DUT and a procedure for calibrating system losses with a power sensor.
- Utilites_ M939xA.cs: Utility methods, such as delay and units conversion.
- agVisa32.cs: Link to Keysight I/O library VISA routines used to send SCPI commands to the X-Series apps.

These projects can be included in a solution to provide instrument control of the PXIe VSG, VSA and the X-Series apps. It can also be used as a reference for using the instruments to make PA measurements.

In addition to the instrument control projects, there is also a shared Data Log project. The Data Log object is created by the TestProgram. Each instrument control Object that performs measurements, has a reference to the data log, to allow a common log for all measurements.

PowerAmpDemoGUI Overview

The PowerAmpDemoGUI project provides the user interfaces for the PA Demo Program. The use of the program is covered in the Keysight RF PA/FEM Characterization and Test Demo Program User's Guide document and will not be covered in this document. In general, the GUI provides buttons that execute methods in the TestProgram project to open and close the instruments and run the tests. Additional controls allow the use set properties in the TestProgram and Instrument Control projects to configure how the tests will be run as well as show the results after the measurements have been run. All of the controls in the GUI include tool tip help text that is shown when the user's mouse hovers over the control.

The project includes the following files.

- PowerAmpDemoGUI.cs: A Windows form that is the main user interface for the PA Demo Program.
- HelpViewer.cs: A Windows form that is used to view the User's Guide from inside the program.
- FormSerialiser.cs: A C# code file that is used to save the state of the main GUI form controls to an XML file and also to load the controls from the contents of an XML file. When the PA Demo Program exits, the state of the GUI is stored into a file named lastState.xml. When the program is launched, the state of the GUI is read from the same file.
- Help Files: MS Word versions of this document and the User's Guide. Also, a PDF version of the User's Guide. The PDF version is copied to the program folder when the project is built.
- Support Files: .CSV template files for user defined frequency and calibration and the .XML files described above. These files are all copied to the program folder when the project is built.

In version 1.0, the GUI to allow interactive control of the PXIe VSG and VSA through the IVI drivers were part of the PowerAmpDemoGUI project. In version 2.0, these have been moved to a separate project. There are now three monitors GUIs, one each for the M9381A, M9391A, and M9393A.

PowerAmpDemoProgram Overview

The PowerAmpDemoProgram contains several methods that are called by the GUI project to configure and run tests. It also includes a number of methods that consist of the test sequence for each of the standards supported in the program.

The PowerAmpTestProgram.cs file includes the constructor for the class and the methods that interface with the GUI. The constructor creates instances of some of the instrument control objects when it is run. In the cases where there is an interface class for an instrument, these objects are created in a separate method that is called when the Init button is pressed in the GUI and before the InitiailizeInstruments method is call. The methods called from the GUI include:

- ConfigTestLists: The GUI includes three listBox controls for displaying the list of X-Series apps, the supported standards and the available measurements. This method populates those controls based on the measurements that are included in the test program.
- CreateVsag: Creates a Vsag object depending on the type of PXI VSA that is selected in the GUI. Currently, M9391A and M9393A are supported. Both PXIe VSAs use the M9381A VSG.
- CreateEtArb: Creates an envelope tracking AWG based on the selections in the GUI. If no ET AWG is selected, a “NULL” object is created that simply returns when each method is called. This allows the Test Sequences to not need to check for the existence of an ET AWG.
- CreateDcSmu: Creates a DC SMU object based on the selections in the GUI. If no DC SMU is selected, a “NULL” object is created that simply returns when each method is called. This allows the Test Sequences to not need to check for the existence of the SMU. Current measurements in the null SMU return without performing any measurements or logging any data.
- CreateRffeDio: Creates a DIO object based on the selections in the GUI. If no DIO is selected, a “NULL” object is created that simply returns when each method is called. This allows the Test Sequences to not need to check for the existence of a DIO.
- CreateDpdEt: Creates an DPD/ET object based on the selections in the GUI. If DPD acceleration is selected, the object for the M9451A is created. If DPD acceleration is not selected, the object that uses the N7614B software is created. If simulate VSG/VSA is selected, a “NULL” object is created that simply returns when each method is called. This is required as the simulation data from the VSA will cause failures in the DPD softwareRffe.

- InitializeInstruments: Performs the initialization routines for the PXIe instruments and X-Series app software and loads all of the required waveforms into the M9381A VSG.
- runSelectedTests: Runs a subset of the standards and measurements based on the current values in the standards and measurements listBox controls in the GUI.
- CloseInstruments: Performs the close instrument method in the TestLib.
- runPowerCalibration: Sets up the conditions and runs the power calibration method in the test library.

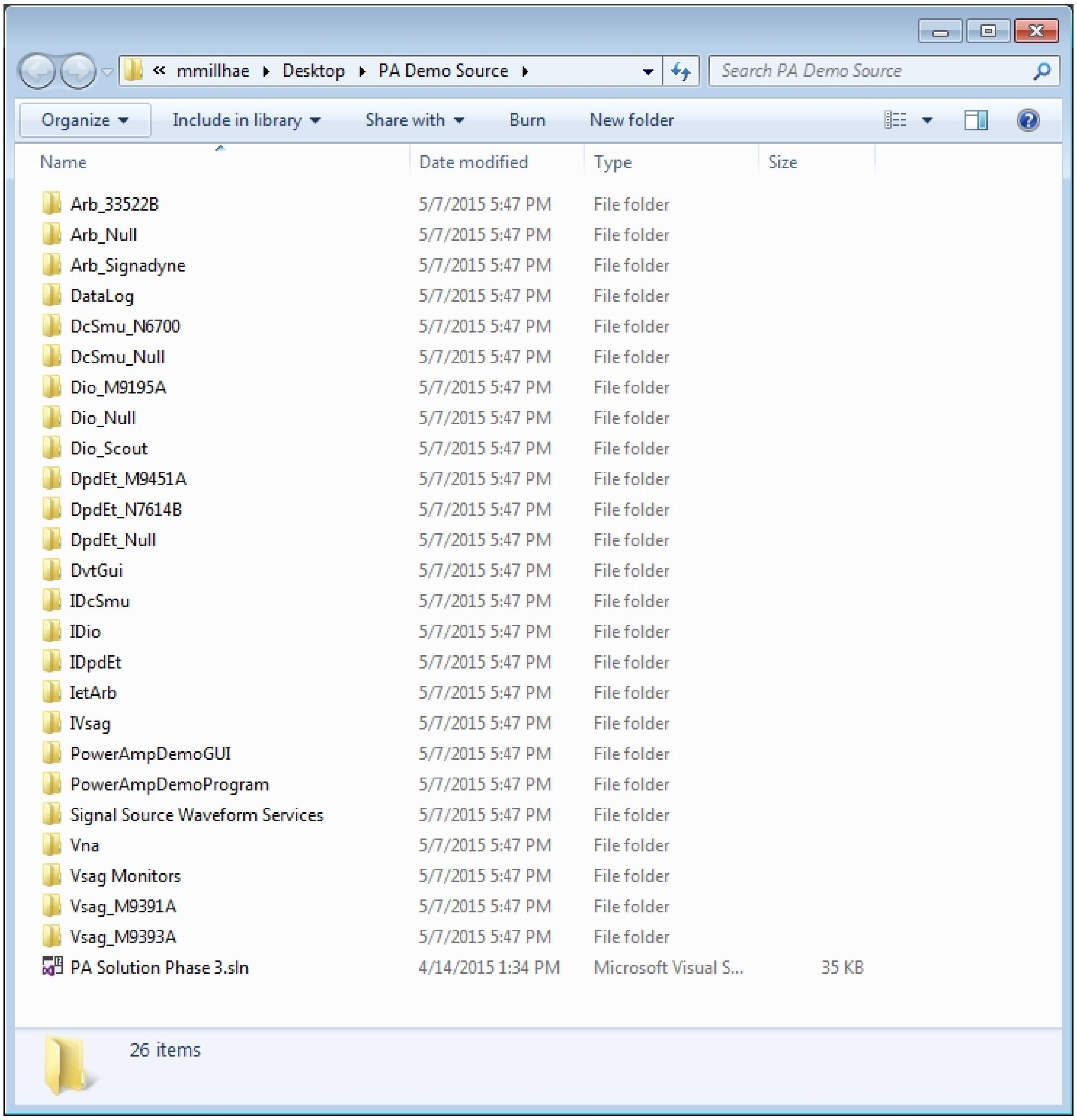
The additional files in this project include:

- Test Sequences: The test sequence method for each standard supported in the Test Program. The method for each standard is included in a separate code file with the name of the standard. Some of the test sequence methods, such as LTE and WLAN have a parameter to specify the bandwidth of the standard to be tested.
- Waveforms: In most cases, the example waveforms installed with the M938x driver are used in the PA Demo Program. Some standards required waveforms that were not included with the examples. These waveform files are included in the project. The waveforms are automatically copied to the program folder when the project is built.

Building the PA Demo Program

The PA Demo Program should be built using Visual Studio 2013. The M90XA software requires .NET version 4.0. The program can be built using the standard version of Visual Studio or any of the advanced versions. It can also be built using the Express Version, with minor modifications to the post build events.

The source code for the project will be provided in a zip file. After unzipping, the following folder structure will be available:



To load the program into Visual Studio, double click the solution file or select:

File | Open Project or Solution from the Visual Studio Menu.

If the correct versions of the IVI drivers are loaded, the program should compile and build successfully. There should be no errors but may be some warnings. If there are errors or warnings about the driver references, insure that the required versions of the drivers are installed or remove references to the drivers and add new references to the installed versions.

PA Demo Program Details

PowerAmpDemoGUI Details

Program Initialization

The PA Demo GUI creates a PowerAmpDemoProgram object and also includes objects for the local panels for the PXIe VSG and VSA and a form to view the help file. When the PowerAmpDemoProgram Object is created, it generates an instance of the data log object and the VNA object. All instrument objects depend on rules selected from the GUI, they are not created until the “Init” button is clicked.

The GUI calls the following methods in the PowerAmpDemoProgram Object during program initialization:

- ConfigTestLists: Called from the GUI constructor, sets the list of values to be shown in the X-Series app, Standards and Measurements lists. The TestProgram should set these to the values supported in that program.

The enabled state of the open, close, run test buttons and the local check boxes are controlled in GUI to match the correct state based on the condition of the program. For example, the open button is enabled on startup and is disabled when the instruments are initialized. All of the methods that control the state of the instruments will set these values to the correct state. The loadGuiState method reads the values from the xml file. If there are file names in any of the data file fields, it will load those file and set the test program properties to the correct values.

In version 2.0 of the program, the PXIe VSA Resource string field in the GUI was changed from a text field to a combo box. During program initialization, the IVI configuration store is read to find saved configurations for the M9391A and M9393A VSAs. The combo box is populated with the names of these configurations. At least one valid configuration must exist or the program will not complete initialization.

In version 3.0, the VSG resource string is also converted to a combo box. All other resource strings are removed from the GUI as they are now automatically detected in the initialization methods.

Instrument Initialization

After the program initialization is complete, the user chooses which instrument should be used in the instrument Setup tab of the GUI. When this is complete, the user clicks the Init button in the GUI. The event handler for this button is the “initButton_Click” method. This method validates that a correct PXIe VSA/VSG configurations have been selected, and sets the TestProgram and instrument object settings related to the instrument addresses and initialization.

Several Methods in the TestProgram are called in this method:

- Create_Vsag: Creates an instance of the Ivsag interface class depending on the text string parameter passed in.
- Create_EtArb: Creates an instance of the IEtArb interface class depending on the text string parameter passed in.
- Create_RffeDio: Creates an instance of the IDio interface class depending on the text string parameter passed in.
- Create_DcSmu: Creates an instance of the IDcSmu interface class depending on the text string parameter passed in.
- Create_DpdEt: Creates an instance of the IEtArb interface class depending on the text string parameter passed in.
- InitializeInstruments: Opens all the Instruments used in the test program, accepts a parameter with the list of X-Series apps to be loaded, if the useXApp property is set to true.

After the InitializeInstruments method returns without errors, the GUI controls are updated to the “Run” condition, with the run related controls enabled and the Init button disabled.

Configuring the Test Condition

When the instrument initialization is complete, the GUI switches to the Test Conditions Tab. There are a large number of settings in this tab, arranged in several sub tabs. The user interaction is primarily to enter values in numeric controls, check boxes and combo boxes. These controls all use one of the two event handlers, setTestProgramValues and setTestProgramValues_Dpd. Within these event handlers, the associated parameters in the Test Program and Instrument objects are set based on the GUI values.

The other primary user interaction is to load a set of values from CSV text files. For each of the CSV settings files, there is a Load File button and unique event handler. Each of these event handers has a common structure:

- Show a file dialog to select the CSV file
- Call a method in the Test Program or one of the Instrument Objects to parse the file and set the values in the associated properties in that object
- Call a method to display these values in the GUI. This method queries the variable just set in the other objects.

There are setup CSV files for the following items:

- DC SMU Setup
- RFFE Commands
- S-Parameter Trace and Segment setup
- PXI VSA/G Test frequencies
- PXI VSA/G Calibration Data

Running Tests

There are two options available for running tests. The Primary method for running tests is the “Run Selected Tests” button. The event handler for this button, runSelectedTests_Click, is the primary sequencer for running tests. This sequencer is designed to emulate the flow of a production test being run on a number of DUTs. The sequencer flow is as follows:

- Disable local control of the instruments.
- Open and initialize the Data log, if data logging is selected.
- Perform the following setup routines, if the setup for these instruments has changed since the tests were last run:
  - Load the PXIe VSG/VSA
  - Calibration Data
  - Setup the DC SMUs
  - Setup the VNA segments and traces
- Send the RFFE Commands to the DUT and update the GUI with the read back values. This is done if RFFE is enabled.
- Close the output relays on the DC SMUs if they are not currently closed. Normally, the DC is left on at the end of running tests to allow local control with the DUT on. There is an option to turn the DC off at the end of the tests.
- Create lists of the sequences to run and measurements to run, based on the items selected in the GUI menus.
- Loop over the following for the number of times in the “Repeat Count” field
  - Run the “runSelectedTests” method in the Test Program.
  - Update the Results Display in the GUI if “Show Details” is selected.
  - Increment the data log Test Index variable, allowing the data from each repetition of the loop to be saved.
- Close the data log.
- Open the DC SMU relays, if that option is selected.

There are a few options that impact the operation of this method:

- If the target power increment is set to a non-zero value, the target power for the DUT output changes by that amount each time the tests are repeated.
- If the Abort button is clicked while the tests are running, the loop terminates after the current test is complete.
- If the “Lock GUI while running” option is selected, all of the GUI interaction, including updating the results and polling the Abort button are suspended. This is useful for timing analysis where there is a concern that the GUI overhead is impacting the overall time.

DVT Interface

The second option for running tests is the DVT interface. The DVT interface was added in version 2.0 of the solution to include design validation testing, in addition to production testing. The DVT Interface is a separate form that is launched by checking the DVT interface check box in the upper left corner of the main GUI. The DVT interface is a second sequence designed to emulate the design validation process where tests are run over a large number of test conditions, typically in nested loops.

The DVT interface is a second sequencer. The primary differences are that only one “Selected Standard” is run and it is run at the center of 3 nested loops. The nested loops can be configured by the user via a CSV text file. Each loop can vary one of three parameters, frequency, DUT output power level, and DC SMU voltage. The loops can be arranged in any order. The data for each iteration is available at the end of the test and plots summarizing key measurements across some or all of the test conditions can be shown.

The event handler for the DVT Interface check box is debugTestBox_CheckedChanged. When the check box is selected, lists of the standards to run and measurements to run, based on the items selected in the GUI menus are created and passed to the DVT interface object as it is created. The Run Selected Tests check box is disabled. When the check box is cleared, the DVT interface object is destroyed and the Run Selected Tests button is enabled.

The DVT Interface object has three main sections:

- Building a tree from the test conditions
- Sequencer to run the selected test over all of the test conditions
- Display the Test Results

The left side of the DVT Interface GUI contains a treeview object. The tree is populated with all the test conditions that are measured in the nested loops. These values are contained in arrays for frequency, power and DC Voltage levels and strings which contain which parameter is used for each of the three loops. When the DVT Interface is opened, a default set of values is loaded into these properties based on the cellular frequencies and target Pout in the main GUI. The buildTree method creates the nodes in the treeview based on these values. The tree can be updated by loading a DVT setup file. The setup file contains which parameters are used for each of the loops, how many iteration of each loop will be run and the values of the parameters for that loop. Each measurement node in the tree is tagged with a number that is the order of the test to be run. When the tests are run the data log Test Index matches this number, allowing the data to be reviewed after the run.

The sequencer section is in the runDvtButton_Click method that is called when the Run Test button is clicked. This method contains the code to run one test at the center of three nested loops. At the beginning of each loop, SetLoopParam method is called with the string containing the parameter for that loop and the index of the loop. The SetLoopParam method changes the hardware as required for that loop condition. The DVT sequencer uses the same data log as the normal run test sequencer does. In this case, the test index is used to set which test condition was run. After the test is run in the inner loop, the current data is displayed in the results section of the GUI.

The third section of the DVT interface is the results display. There are two tabs in the DVT GUI for results. The first tab displays the results from one test condition. A table is built with many of the measured values. If the LTE DPD/ET measurement is run the plots of AM/AM and AM/PM conversion, before or after the DPD is applied are also shown. Since the DVT GUI was primarily developed for the DPD/ET design validation tests, there is more functionality in the GUI for this set of tests. However, any of the measurements can be run in the DVT GUI. There is a check box in this tab to select displaying the current measurement results. When this is checked, the data for each data set is shown when that measurement completes. If the user selects a node from the treeview, the results display switches to that test. This can be done while the tests are running or after the tests are complete. The second tab in the results section is for summary data plots. There is a single plot that displays results of a selectable parameter for part or all of the measured data. To see one of these plots, the user selects one of the higher level nodes in the results tree. If the top node, Measurement Results, is selected, the data for all of the measurements is plotted. Above the graph there is a combo box for selecting which parameter is plotted. The plot includes one trace for each execution of the inner most loop. The X-Axis of the plot includes the values of the inner most loop. The legend for the plot includes the conditions of the two outer loops to identify the data.

Local Mode

Local mode operation is supported for the PXI VSA, PXI VNA and X-Series applications. There is a check box in the GUI for each of these local mode options. When the PXI VSA and X-Series Applications are put in local mode, the VSG will also be put in local mode. When any of the local modes are enabled, the Run Selected Tests button is disabled.

The local mode event handlers are:

- LocalxApp_CheckedChanged: Calls the setXappMode method in the Vsag. This toggles the X-Series Applications between remote (Single Sweep) and local (Continuous Sweep) modes.
- localM9391_CheckedChanged: Calls the showVsgMonitor and showVsaMontitor methods in Vsag when checked and stopVsgMonitor and stopVsaMonitor methods in Vsag when unchecked.

- localVnaBox_CheckedChanged Calls the localVna method in Vna with a Boolean equal to the checked state.

The local mode event handlers create an object for the PXIe VSG and VSA monitor panels, passing in the driver reference from the TestLib. These objects are disposed when local mode exits. The Run Tests Button is disabled while local mode is active to prevent local modes changes to the drivers while tests are running.

Closing the Program

CloseInstruments: Called from the Close button event handler and the form closing event handler, closes all the driver sessions and terminates the X-Series apps software. It is required to call this from the form closing event handler. If the program exits without closing the drivers, the PC will need to be rebooted before the instruments can be used again.

PowerAmpDemoProgram Details

Initialization

The PowerAmpDemoProgram Object creates an instance of the PowerAmpTestLib in the constructor. The constructor for the TestLib takes parameter to size the data log variables. This allows all of the data acquired in one complete run of the test program to be available in the data log at the end of the run.

Required Methods

The structure of the PS demo program allows for different test programs to be used with the same GUI, or to be used independently of the GUI. The following methods are called from PowerAmpDemoGUI and must be included in the Test Program:

- ConfigTestLists: Populates the lists of X-App names, Standards Names and Measurement Names. These should match the tests performed in the test program.
- InitializeInstruments: Opens all of the drivers and load all of the required waveforms into the VSG.
- CloseInstruments: Closes all of the drivers.
- runSelectedTests: Runs the test methods based on the list of standards and measurements passed from the GUI.
- readFreqData: Load the test frequencies from a CSV file.
- initTestFreqs: Sets the test frequencies to the default values.
- CreateVsag: Creates the Vsag object based on the string parameter passed from the GUI.

- CreateEtArb: Creates the EtArb object based on the string parameter passed from the GUI.
- CreateRffeDio: Creates the Dio object based on the string parameter passed from the GUI.
- CreateDcSmu: Creates the DC SMU object based on the string parameter passed from the GUI.
- CreateDpdEt: Creates the DPD/ET object based on the string parameter passed from the GUI.

Power Amplifier Tests

The tests for each standard are contained in a separate code file. The tests call methods in the Vsag Interface class, allowing them to run with any of the selected instruments. All of the tests that use the Vsag share a common structure, with the basic flow as shown below:

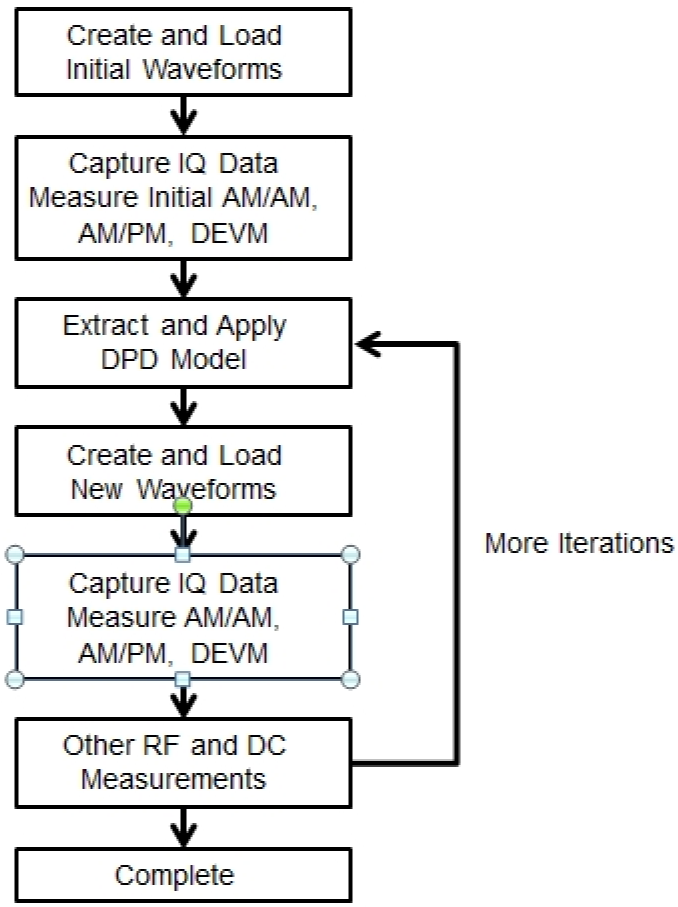
- Parameter Setup: Set the name of the ARB waveform to be used and configure a number of basic parameters in the PXIe VSA, such as bandwidths, filter shapes, and sample rates. This is done by passing the name of the standard to the setupParameters method in the TestLib. LTE and WLAN have an parameter for the bandwidth, which is used in this section for the required setup.
- Start the Test Timer.
- Setup the PXIe VSG and VSA to the initial conditions for the test.
- If EVM is being measured, setup the X-App to the correct mode and configure the EVM measurement for that mode.
- Loop over the list if test frequencies
  - Set the PXI VSG and VSA to the correct frequency
  - Servo to the correct output power
  - For each measurement, if enabled, make the measurement. If the measurement uses the X-Series apps, follows these steps:
    - Unlock the PXI VSA from the driver.
    - Set the X-App Frequency.
    - Make the measurement.
    - Lock the PXIe VSA to the driver.
    - Execute the Reset to Default Properties method of the PXI VSA Driver.
    - Reset the PXIe VSA frequency, power level and trigger mode.
- Stop the Timer.
- Record the total test time in the data log.

Tests that use other instruments, instead of the PXIe VSA/VSG, will still follow the same general structure with the timing and data logging, but will not use the specific

Vsag settings. For example, the S-Parameter measurement makes a single measurement using the setup defined in the S-Parameter setup file.

DPD and ET

The DpdEtTest is somewhat different from the other tests that use the Vsag. This test uses the API for the N7614B to generate RF waveforms and uses either the N7614B or M9451A module to generate envelope waveforms and perform the DPD model extracted from measured IQ data. The following diagram shows the basic test flow of the DPD/ET test:



The DPD/ET test also is only performed on one frequency, the first frequency defined in the cellular frequency array. The DPD/ET test is also the only way to generate envelope waveforms for the ET AWG. The required Signal Studio license must be installed on the host PC to use N7614B and M9451A to generate RF and envelope signals.

The N7614B software does not control the PXIe VSG and VSA in this use model. The Vsag and EtArb objects control the instruments. The N7614B software is only used to generate waveforms and extract the DPD models. At this time, N7614B does not support direct control of the PXIe VSG and VSA using the N7614B GUI. The use model shown in this example must be followed.

The DVT Interface was primarily designed to show the results of the DPD/ET tests, supporting showing the results of ACPR, EVM, AM/AM and AM/Pm plots.

In version 3.0, there were several changes to the DPD/ET test:

- The code to control the N7614B software was refactored into the IDpdEt interface class. One instance of this class uses just the N7614B software and one instance uses a combination of the N7614B software and the M9451A Measurement Acceleration Module.
- The DPD/ET test sequence now supports additional standards beyond LTE.
- The user can also select a different waveform for DPD/ET tests from the GUI. These waveforms can be any of the formats supported by N7614B, including MATLAB, integer binary, text and Signal Studio.

With the M9451A, data is transferred between the VSG, the VSA and the M9451A measurement acceleration module using peer to peer data transfers, improving throughput and allowing transfer of encrypted waveform data. The following process is used to generate DPD waveforms for the VSG using the M9451A:

- Up-sample the RF waveform for the DPD Reference waveform: This data needs to be oversampled by a factor of 2-4 from the original symbol rate to allow proper operation of the DPD calculations. In the demo program, the N7614B software is used to up-sample the initial waveform and convert the data to Signal Studio .wfm file formats to allow easy loading into the M9381A.
- Load the Reference waveform into the VSG.
- Load a second copy of the reference waveform into the VSG as a placeholder for the DPD waveform.
- Copy the reference waveform data from VSG memory to M9451A memory, using peer to peer data transfer.
- Play the Reference waveform.
- Capture IQ data at the output of the VSA at the same sample rate as the reference waveform. Copy this data from the VSA capture memory to the M9451A memory using peer to peer data transfer. It is also possible to copy the IQ data from the VSA to the M9451A by first uploading the data to the application and then loading into the M9451A. This option is available to support VSAs that do not include peer to peer data transfer capability. Peer to peer should be used when available as it improves throughput.
- Perform the model extraction in the M9451A. This process generates a look up table that can be used to pre-distort the reference waveform. The LUT, AM/AM, AM/PM data and a delta EVM metric can be saved from this step.

- Apply the lookup table to the reference waveform to generate the DPD waveform. Copy this waveform to the VSG, overwriting the second copy of the reference waveform that was loaded in a previous step. If the DPD procedure is run multiple times, the second copy of the waveform will continually be updated with the latest DPD waveform.

For envelope tracking, the envelope waveform is generated from the Reference IQ waveform, the shaping table and two scaling parameters, one for the envelope voltage and one for the magnitude data. The envelope scaling factor factors in the gain in the output stage of the AWG and the gain of the envelope tracking power supply. This allows the M9451A to directly calculate the waveform data that is loaded in the AWG module. The following formula calculates this value with the following equation where dacScale = 32767 and awgGain = 1.5.

```
double envScaleFactor = (etpsGain * 2 * arbScale) / dacScale;
dpdAcc.EnvelopeTracking.EnvelopeScaleFactor = envScaleFactor;
```

The magnitude scale factor is applied to the Reference IQ data before the value is looked up in the shaping table. When the magnitude scale factor is 1, the peak value of the magnitude produces the maximum voltage in the shaping table. Setting the magnitude scale factor to values between 0 and 1 causes a smaller portion of the shaping table to be used. The magnitude scale factor is calculated with the following formula:

```
double magScale = Math.Pow(10, envAbsRfOver / 20);
dpdAcc.EnvelopeTracking.MagnitudeScaleFactor = magScale;
```

Where envAbsRfOver is the value in dB of how much lower the PA output will be compared to an output power level that uses the full range of the envelope shaping table.

The envelope data is first copied to the demo program and then to the AWG.

Instrument Control Details

In version 3.0 of the PA Demo Program, all of the instrument objects have been moved to an interface class with the exception of the VNA. For the other instruments, the model of instrument to be used is determined at run time by selections in the GUI. The tests in the testplan, described above, call the method in the interface classes. .

Instrument Classes created in TestPlan Constructor:

```
public PowerAmpDemoProgram()
{
    dataLog = new DataLog(numTests, maxAcprMeas, numCurrMeas);
    vna = new Vna(dataLog);

    // Set any default parameter values
    targetPout = -5;
    targetGain = 16;
    poutMargin = 0.05;

    initTestFreqs();
}
```

Instrument created via interface classes:

```
public void createVsag(string vsaType)
{
    switch (vsaType)
    {
        case "M9393A":
            vsag = new NSM9393A.Vsag_M9393A(numTests, dataLog);
            break;
        case "M9391A":
            vsag = new NSM9391A.Vsag_M9391A(numTests, dataLog);
            break;
        default:
            throw new Exception("Unsupported VSA Type:  " + vsaType);
    }
    // Load the calibration data
    vsag.initCalData(1, 6);
}

public void createEtArb(string arbType)
{
    switch (arbType)
    {
        case "33522B":
            etArb = new NSETARB.Arb_33522B();
            break;
        case "Signadyne":
            etArb = new NSETARB.Arb_Signadyne();
            break;
        case "NULL":
            etArb = new NSETARB.Arb_NULL();
            break;
        default:
            throw new Exception("Unsupported ARB Type:  " + arbType);
    }
}
```

```
public void createRffeDio(string dioType)
{
    switch (dioType)
    {
        case "Scout":
            dio = dio = new NSDIO.Dio_Scout();
            break;
        case "M9195A":
            dio = new NSDIO.Dio_M9195A();
            break;
        case "NULL":
            dio = new NSDIO.Dio_NULL();
            break;
        default:
            throw new Exception("Unsupported RFFE DIO Type:  " + dioType);
    }
    dio.rffeAddress = new int[0];
    dio.rffeCommand = new string[0];
    dio.rffeData = new int[0];
    dio.rffeReadback = new bool[0];
    dio.rffeReadData = new int[0];
    dio.rffeRegister = new int[0];
}

public void createDcSmu(string dcSmuType)
{
    switch(dcSmuType)
    {
        case "N6700":
            dcSmu = new NS_DCSMU.DcSmu_N6700(dataLog);
            break;
        case "NULL":
            dcSmu = new NS_DCSMU.DcSmu_NULL();
            break;
        default:
            throw new Exception("Unsupported DC SMU Type:  " + dcSmuType);
    }
    dcSmu.dcCurrLim = new double[0];
    dcSmu.dcVoltage = new double[0];
    dcSmu.doCurrMeas = new bool[0];
}
```

```
public void createDpdEt(string dpdType)
{
    switch (dpdType)
    {
        case "M9451A":
            dpdEt = new NSDPD_M9451A.DpdEt_M9451A(vsag, etArb, dataLog);
            break;
        case "N7614B":
            dpdEt = new NSDPD_N7614B.DpdEt_N7614B(vsag, etArb, dataLog);
            break;
        case "NULL":
            dpdEt = new NSDPD_Null.DpdEt_Null(vsag, etArb, dataLog);
            break;
        default:
            throw new Exception("Unsupported DPD ET Type:  " + dpdType);
    }
}
```

Data Log Class

The data logging of measurement results and test times is done at the instrument control level. Each method in the instrument control classes records the time to execute the method and any measurement results from that method. A single data log object is created in the test plan and provided to the instrument classes as a parameter when creating these objects. Each method that uses the data log to record results should use the following format:

```
// Log Results
string[] dataNames = { "Pout", "Pout Time" };
string[] dataUnits = { "dBm", "ms" };
string[] numFormats = { "0.00", "0.000" };
double[] dataValues = { dataLog.chanPower[dataLog.testIndex], (double)sw1.ElapsedTicks / swFreq * 1000.00 };
dataLog.logDataItems(dataNames, dataUnits, numFormats, dataValues);
```

Vsag Interface Class

The Vsag class controls the PXIe VSG and VSA used for most of the tests in the PA Demo Program. The PA Demo program currently supports two Vsag models, one for the M9381A and M9391A and the other for the M9381A and M9393A. The PXIe VSG and VSA were combined into one class to provide more direct examples on how the drivers for these two instruments are used together.

The methods defined in the Vsag class include the following categories:

- Instrument Setup:
    - Load waveforms
    - Configure Trigger setup
- Test Condition setup

    - Set Acquisition Parameters
    - Set Instrument Waveform, Frequency, amplitude and configure acquisition
- Servo Input Power
- Power and Spectrum measurements using the IVI Driver
    - Channel Power
    - ACPR
    - Harmonics
    - Power vs. time
    - Gain Compression
- X-Series application Measurement Configuration
- X-Series application EVM and SEM measurements

The structure of the Channel Power and ACPR measurements was simplified from version 1.0 of the PA Demo Program. In this version there is a single method for each measurement and internal to that method the use of the power or FFT mode is selected. Also, the calculation of channel power from the FFT data is all done in a single method that is called from the power and ACPR measurements.

For EVM measurements, new methods were added that split the capture of the data for EVM from getting the results. This allows using the hardware for other measurements while the X-Series applications are calculating EVM. Typically, this allows enough time to measure all of the harmonics while the EVM result is being calculated. This approach is implemented for LTE-FDD, LTE-TDD, WCDMA and TD-SCDMA.

Edition 2, July, 2015

Y1299-90007
www.keysight.com