Next Generation of Test Automation and Systems

Jad Faraj
Critical Commununication Solutions Group
Keysight Technologies, Inc
Santa Rosa, USA
jad_faraj@keysight.com

Jacob Sanderson
Critical Commununication Solutions Group
Keysight Technologies, Inc
Santa Rosa, USA
jake.sanderson@keysight.com

Kenneth Carolus
Critical Commununication Solutions Group
Keysight Technologies, Inc
Santa Rosa, USA
ken_carolus@keysight.com

***Abstract*— In this paper, we explore a new software architecture that embodies the key attributes of effective test automation development, as well as demonstrates the essential elements for fast product development. We will present real cases for radio test deployments and optimization to troubleshoot and characterize all current and future radios.**

***Keywords*— Automatic Test Equipment, Test and Measurement, Land Mobile Radio, Military Communication, Radio communication.**

I. Introduction

Working to provide test solutions for the world's leading public safety and Military Communication (MilCom) radio manufacturers and maintainers, we see one driving constant: a combination of evolution and revolution in the underlying technologies as well as expansion of radio standards (i.e. Analog and digital formats such as AM, FM, APCO P25, DMR, TETRA, WLAN, Bluetooth) and radio capabilities. For example, one of the hottest topics in the Land Mobile Radio (LMR) industry today is the use of private Long-Term Evolution (LTE). On February 22, 2012, the United States Congress created the First Responder Network Authority (FirstNet), which allocated 108 MHz of spectrum in 700 MHz band to support the creation of a nationwide interoperable broadband network [1]. LTE technology enables very high-speed data communications that is not possible with current LMR technologies (APCO, DMR, TETRA…etc). LTE was designed to deliver high bandwidth mobile data which allows mobile devices to stream video or to transfer large amounts of data quickly.

While these advanced technologies are great news for public safety and MilCom personnel, they create many challenges for test engineers and radio manufacturers. Product technologies are more sophisticated and product development cycles are getting shorter. In the test and measurement industry, the number and complexity of product tests have increased significantly. Product testing is an essential part of the product qualification and manufacturing cycle. Design validation test and manufacturing functional test combined with six sigma analysis and results monitoring using Statistical Quality Control (SQC) are methods to monitor product and process health.

The test automation software (SW) environment influences many aspects of the test processes and product development workflow, from test development and deployment times to test repeatability and throughput. It's no surprise that selecting the right test SW environment and hardware are critical to test system effectiveness.

II. Hardware and Software Related Roles and Activities

When considering the test system and software environment, it helps to consider how they are going to be used. There are various roles and responsibilities during test development, and effective Hardware (HW) & SW environments need to address the needs of all.

The test engineer may have the most balanced knowledge of the product itself and the test system operation, including test plan development. The test engineer maintains a close relationship with the Research and Development (R&D) team, since advanced test technologies and methods may need to be shared. As opposed to a test engineer, a software developer creates core measurement IP, some of which may also be leveraged from R&D especially if they're cutting edge technologies. In addition, the SW developer maintains the operator Graphical User Interface (GUI) and test results reporting systems. Finally, the New Product Integration (NPI) engineer may makes use of the Structured Query Language (SQL) data and verifies product and process readiness.

In some organizations the test engineer, SW developer and NPI engineer may all be the same person. Regardless, it helps to understand the different roles and responsibilities since they all interact with the HW and test software environment. The important takeaway is that collaboration of various groups and sharing of product architecture and test knowledge, measurement science, GUI and results archiving as well as test plan creation and optimization is essential to a smooth and effective test operation. The test system and software environment should be selected with team collaboration in mind.

III. Key Considerations of Effective Test Automation Software Environments

For test automation applications there are various software environments to consider. Both text-based and visual programming environments are readily available for general instrumentation control and measurements.

Visual programming works well for some basic measurements. They provide insight to simple program flow, and graphical controls and widgets are available for physical items such as thermometers, pressure vessels and flow rates. However, visual programming environments become challenging to use for large scale test applications. Editing becomes challenging; the graphical nature means the application may extend over several pages and the interconnect hard to follow. Also, visual programming tools have higher software overhead which may slow execution times. Program debugging also becomes more challenging.

As opposed to visual programming, text-based programming methods using standard and modern languages have several advantages. Execution speeds are generally very high, especially for compiled code, and skilled programming expertise is readily available. Commonly available software Integrated Development Environments (IDE) are relatively low cost and easy to use. In addition, modern text-based

978-1-7281-2832-0/19/$31.00 ©2019 IEEE

programming environments support a fast, scalable, and open environment. For example, a text-based programming environment like C#/.NET is well suited for test software for several reasons. Using this main-stream language allows leveraging the extensive amount of engineering hours invested on standard languages and IDE. With these high levels of investment, the IDE and languages have been well proven and optimized for low overhead. The extensive world-wide adoption also means a large well-trained software community is readily available. In addition, The C#/.NET programming model lends itself to a modular SW architecture. For instance, software development kits (SDK) are supported and provide quick library assembly development and standard methods to organize and distribute using tools like package managers.

IV. Key Attibutes of Effective Test Automation Software Environments

For an effective test system software environment there are 5 key attributes that will have a strong influence on the overall success of the test automation framework.

*A. **Simple:***

Product development and deployment is a fast-paced process that is constantly evolving. Test-engineering resources may change as operations ramp up or become distributed. To succeed in this dynamic environment, it’s best to minimize technology barriers while providing required capabilities.

*B. **Scalable:***

Over time, product test may grow and evolve in number and type. Existing products may benefit from increased throughput; new products may require test hardware that hasn’t been used before. In either case, test hardware and software must easily accommodate changes.

*C. **Fast:***

Clearly, overhead times in the test environment must be low enough to minimize the impact on overall test throughput. This attribute is typically consistent with the “simple” attribute, but not always. Consider: parallel operations may improve test speed, but may also increase complexity in the test plan. The software environment should support tools and methods for the analysis of throughput; it should also support parallel processing.

*D. **Efficient:***

Here, this refers to engineering time and deployment. Simple architectures work in your favor because lower complexity often reduces development time. Also, Commercial-Off-The-Shelf (COTS) SW components enable you to leverage the manufacturer’s investment in engineering effort.

*E. **Open:***

This has multiple advantages. First. an open platform offers the flexibility to customize the final test solution as needed and open up options for third-party collaboration (if needed). Second, openness makes it easier to mix and match external hardware and software components for your next generation test and design.

Figure 1 shows a high-level software organization that embody all the attributes above. This structure has been proven to work well for fast, scalable and open test architectures.

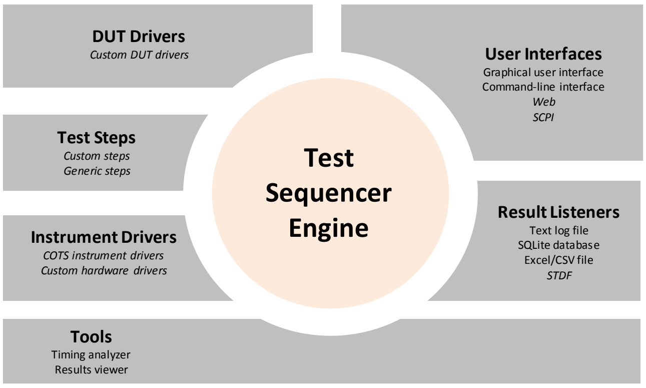


Fig. 1. Versatile architecture for test Automation and sequencing

The test sequencer is central to the overall test process and works in conjunction with the other software blocks. It should be lightweight and nimble with minimal software overhead. The test environment should easily handle development of auxiliary plugins or modular libraries. Also, methods to handle asynchronous events and unexpected exceptions are needed (e.g., delayed responses and timeouts). For optimal throughput, parallel test configurations should be available.

Surrounding the sequencer are essential functional blocks, often architected as separate software modules, commonly referred to as software plugins:

- **Device Under Test (DUT) Drivers**: A SW driver plugin that typically has an Input/output (I/O) layer to enable communication and DUT control. The I/O layer could be standard or proprietary, serial or parallel; these DUT (i.e. Radio) often have unique command sets and test-mode timing. Whether DUT I/O is COTS or home-grown, quick integration with the test environment is important.
- **Test Steps**: The test step may be created by combining code with other method calls to instruments, DUTs or utilities. Key functions include results determination, reduction and interpretation of measurement results such as Error Vector Magnitude (EVM) or symbol demodulation. The software Intellectual Property (IP) of this plugin may come from in-house experts, test-engineering teams or even third parties. In all cases, the environment must easily integrate the IP into the test step and test plan.
- **Instrument Drivers**: In most test systems, core instrumentation is COTS or user-built; Instrumentation I/O is commonly built on standards such as LXI, PXIe or AXIe; the instrument maker usually provides low-level instrument drivers. Therefore, the block must also create the instrument object and support open and closed methods for the instrument session.
- **Tools** like timing and results analyzer are helpful during early development for testing and

debugging. Also, these tools are mandatory for test-throughput optimization and results viewing.

- **Results Listeners**: Many tests require more than a pass/fail determination. The role of these SW plugins modules is to monitor, manage and store measurement data. The test environment must conform to archive-tool requirements, whether on the local controller or over an enterprise storage management network.
- **User interfaces**: In R&D, the GUI exposes key information for test construction, debug and optimization. Production has different needs compared to R&D: for example, an operator may handle the DUT and control the test plan; more common is purely automated test sequencing and material handling. A flexible sequencer will support multiple GUIs and accommodate manual and automated handling.

V. Radio Test Automation solution

A. *HW & SW architecture*

The radio test automation solution is designed based on a flexible, scalable and open HW and SW architecture. For the test automation SW, the core is using the PathWave Test Automation [2], formally called Test Automation Platform (TAP). Most recently, TAP became on open source test sequencer engine [3].

Figure 2. Shows the SW architecture and organization. The structure of TAP and the radio test automation plugins embody all the features and attributes described in the above section.

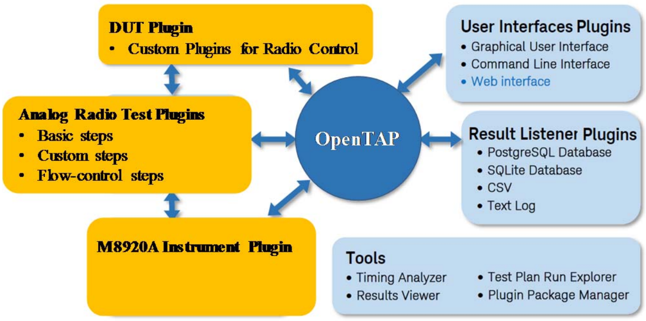


Fig. 2. Radio Test Automation on PathWave Test Automation test architecture.

Figure 3 shows the architecture of the entire radio test system, including the hardware and software. The core solution for the HW is the Keysight PXI Radio Test Set M8920A. This is a modular hardware solution that is designed to reduce the time and effort needed to address the design, production, and maintenance stages of the radio life cycle [4]. Additionally, the M8920A has a wealth of upgradeable features—bandwidth, measurement apps, memory—and more importantly it is future-ready to support wideband applications for Wireless LAN, Bluetooth and LTE to address evolving test needs. All radio test measurement science is implemented in the form of X-Series application SW.

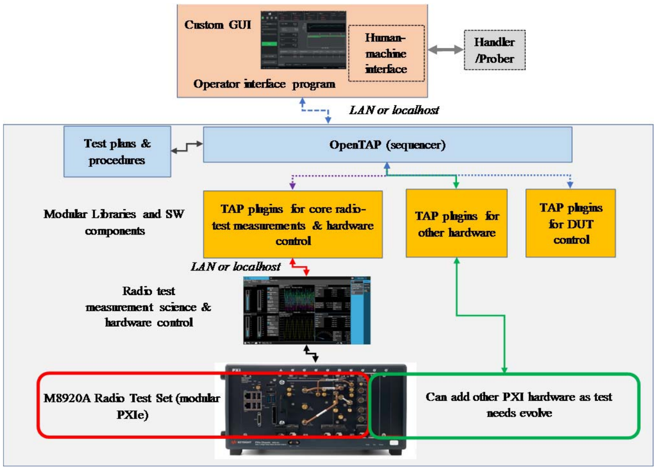

Custom GUI
Human-machine interface
Handler /Prober
Operator interface program
LAN or localhost
Test plans & procedures
OpenTAP (sequencer)
Modular Libraries and SW components
TAP plugins for core radio-test measurements & hardware control
TAP plugins for other hardware
TAP plugins for DUT control
LAN or localhost
Radio test measurement science & hardware control
M8920A Radio Test Set (modular PXIe)
Can add other PXI hardware as test needs evolve

Fig. 3. Best practices for a high-end radio test-automation HW and SW environment.

The third layer includes the Radio Test plugins (KS83120A, KS83122A and KS83123A). These plugins are distributed as modular libraries or software components plugins. Their roles are to control the M8920A, collect radio-test measurement data, and implement test procedures as required by the LMR standards. Moreover, additional user generated plugins libraries are easily developed and distributed using the Visual Studio templates, IDE and KS8400A plugins package manager to address evolving test needs. These plugins are designed to be plugged into OpenTAP. Organizing the test system in a such way is an effective, flexible and scalable architecture that promotes plugins re-use and team collaboration.

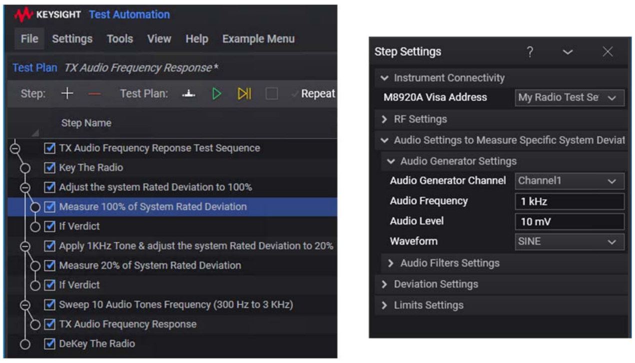
B. *Test deployement across the product workflow*

This section will demonstrate the flexibility and scalability of such hardware and software architectures by showing cases using several measurements for an analog FM radio.

1) ***Transmit Audio Frequency response measurement***:

The audio frequency response is defined in the TIA 603 standard document [5] as the degree of closeness to which the frequency deviation of the transmitter follows a prescribed characteristic, typically, 6 dB/octave pre-emphasis.

Figure 4 show the test plan with all the test steps that are needed to perform this measurement in manufacturing. Each test step in this sequence has different settings and several parameters. This is very important for test engineers, especially if they want to perform an operation that goes beyond what is specified in the standards. This can be useful when performing advanced troubleshooting or characterization



Test Plan *TX Audio Frequency Response**

Step Name

☑ TX Audio Frequency Reponse Test Sequence
☑ Key The Radio
☑ Adjust the system Rated Deviation to 100%
☑ Measure 100% of System Rated Deviation
☑ If Verdict
☑ Apply 1KHz Tone & adjust the system Rated Deviation to 20%
☑ Measure 20% of System Rated Deviation
☑ If Verdict
☑ Sweep 10 Audio Tones Frequency (300 Hz to 3 KHz)
☑ TX Audio Frequency Response
☑ DeKey The Radio

Instrument Connectivity
M8920A Visa Address: My Radio Test Se
RF Settings
Audio Settings to Measure Specific System Deviat
Audio Generator Settings
Audio Generator Channel: Channel1
Audio Frequency: 1 kHz
Audio Level: 10 mV
Waveform: SINE
Audio Filters Settings
Deviation Settings
Limits Settings

Fig. 4. Transmit Audio Frequnecy Response measurement using TAP and the radio test automation plugins KS83120A.

Test engineers can now easily configure their test plans by simply dragging and dropping each test step. During this process, no programming or SCPI commands are needed—and that enables test engineers to focus on what matters most to them: product development, test strategy and test optimization.

Figure 5 shows the measurement results of the transmitter audio frequency response test plan. As you can see, the measurements are in close agreement with the values specified in the TIA standard—and this radio is operating within its specified limits. You can clearly see the 6 dB/octave characteristics in the range of 300 Hz to 5 kHz.

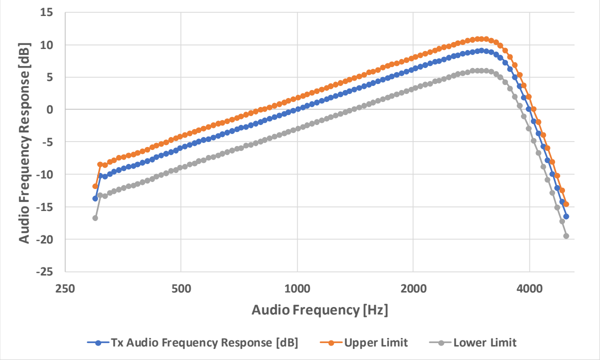
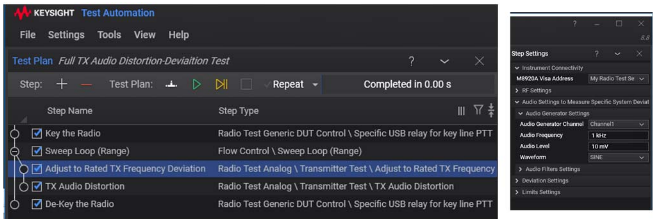


Fig. 5. Transmit Audio Frequnecy Response measurement results.

*2) **Transmit Frequency Deviation and Audio Distortion**:.*

In this use case, we're using the Radio Test Automation plugins with what we call flow control. In other words, we use the test steps in a high-level sweep loop to capture several parameters in the step settings.

Figure 6. Shows a use case to demonstrate how you can re-use the radio test automation plugins to perform specific operations that go beyond the standards. In this example, we're using the test steps in a high-level sweep loop to capture several parameters in the step settings. Here, we're sweeping the audio voltage level and at each one we are measuring transmit frequency deviation and audio distortion for an FM transmitter. This measurement is often made early in the development of an FM transceiver to characterize the overall quality of its transmitter.



Test Plan *Full TX Audio Distortion-Deviaition Test*

Completed in 0.00 s

Step Name	Step Type
☑ Key the Radio	Radio Test Generic DUT Control \ Specific USB relay for key line PTT
☑ Sweep Loop (Range)	Flow Control \ Sweep Loop (Range)
☑ Adjust to Rated TX Frequency Deviation	Radio Test Analog \ Transmitter Test \ Adjust to Rated TX Frequency
☑ TX Audio Distortion	Radio Test Analog \ Transmitter Test \ TX Audio Distortion
☑ De-Key the Radio	Radio Test Generic DUT Control \ Specific USB relay for key line PTT

Step Settings

Instrument Connectivity

M8920A Visa Address: My Radio Test Se

RF Settings

Audio Settings to Measure Specific System Deviat

Audio Generator Settings

Audio Generator Channel: Channel1

Audio Frequency: 1 kHz

Audio Level: 10 mV

Waveform: SINE

Audio Filters Settings

Deviation Settings

Limits Settings

Fig. 6. Radio Test Automation in flow control mode.

As shown in Fig 7. This measurement can clearly reveal the effect of the audio voltage level on FM deviation linearity and overall transmitter distortion. This test plan highlights how easy it is to deploy these plugins to different departments in your organization—and then re-use the plugins to improve productivity and also optimize, in a flexible way, testing for each phase of product-development life cycle.

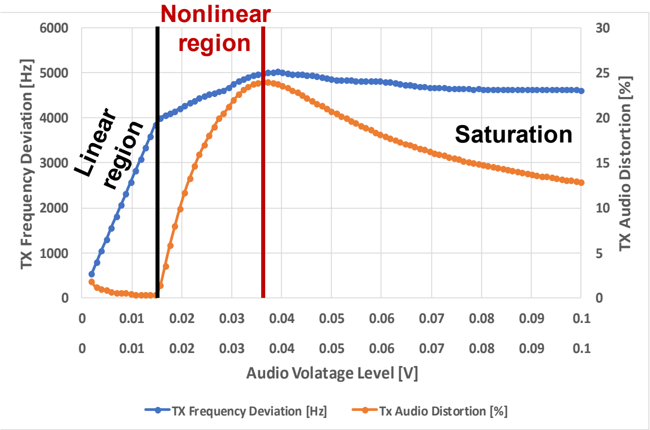


Fig. 7. Full transmit audio distortion and deviation for an FM transmitter surement using TAP and the radio test automation Plugins KS83120A.

Conclusion

We have discussed the benefits of carefully selecting the test-software environment for test automation. We have also illustrated how the right environment is crucial to ensuring successful test automation and operation.

A few use cases helped show how an open, scalable and flexible hardware and software environment promotes reuse of technologies, enhances team collaboration, and minimizes cost of test—all while addressing the challenges of radio-test qualification and manufacturing test.

The combination of the Keysight PXI radio test set and the Test Automation Platform enables a connected workflow and ensures future readiness for radio-test automation and measurements. In turn, this helps test engineers quickly create and deploy radio-test solutions for next-generation designs. It also improves the effectiveness of test automation by enabling greater reuse of test IP across the product development lifecycle.

References

[1] Middle Class TAX Relief And Job Creation Act of 2012, Public Law 112– Publication 96, FEB. 22, 2012. Available: https://www.congress.gov/112/plaws/publ96/PLAW-112publ96.pdf

[2] Keysight Technologies, K8400A Test Automation on PathWave, Literature number: 5992-1909EN, keysight.com, Available: https://literature.cdn.keysight.com/litweb/pdf/5992-1909EN.pdf?id=2796881\

[3] Keysight Technologies and Nokia Launch OpenTAP™, the Open Test Automation Project for Accelerating Software Development, News Releases, May. 21, 2019. Available: https://about.keysight.com/en/newsroom/pr/2019/21may-nr19075.shtml and at https://www.OpenTAP.io

[4] Keysight Technologies, Keysight M8920A PXIe Radio Test Set, Literature number: 5992-2802EN, keysight.com, Available: https://literature.cdn.keysight.com/litweb/pdf/5992-2802EN.pdf?id=2957540

[5] Land Mobile FM or PM - Communications Equipment - Measurement and Performance Standards, TIA 603 E, March 2016. Available: https://global.ihs.com/doc_detail.cfm?document_name=TIA-603&item_s_key=00144319