dap TECHNOLOGY

SOFTWARE

AS5657-ATE

Setup	DUT	Log	Report	ATE GUI

AS5657 TestSuite
Customizable Test Scripts

EXE | > | C++

ATE Framework

Host Computer

FireDiagnostic Suite API

AS5643
AS5643
Protocol

ATE
Extension

FireStack API

AS5643
AS5643
Personality Profile

ATE
Personality Profile

FireSpy HW / FW

1394
Analysis Eng.

FireTrac HW / FW

1394
Bus Engine

MIL1394
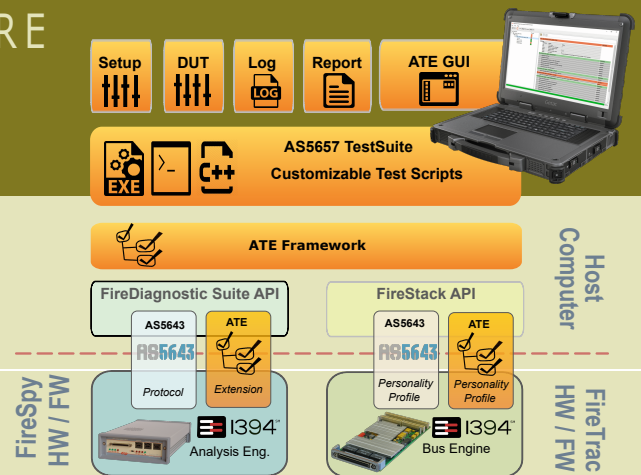
PRODUCT OVERVIEW:

The AS5657 Automated Test Environment (AS5657-ATE) is designed to automate device compliance testing according to the SAE-AS5657 "Test Plan/Procedure for AS5643 IEEE-1394b Interface Requirements for Military and Aerospace Vehicle Applications". The purpose of this tool is to automate and simplify the task of AS5643 compliance verification and testing, which of course is essential for device compatibility within avionics and aerospace programs. The tool utilizes and benefits from proven test and interface components already developed by DapTechnology and adds test executive functionality together with predefined test scripts and a customization and expansion environment.
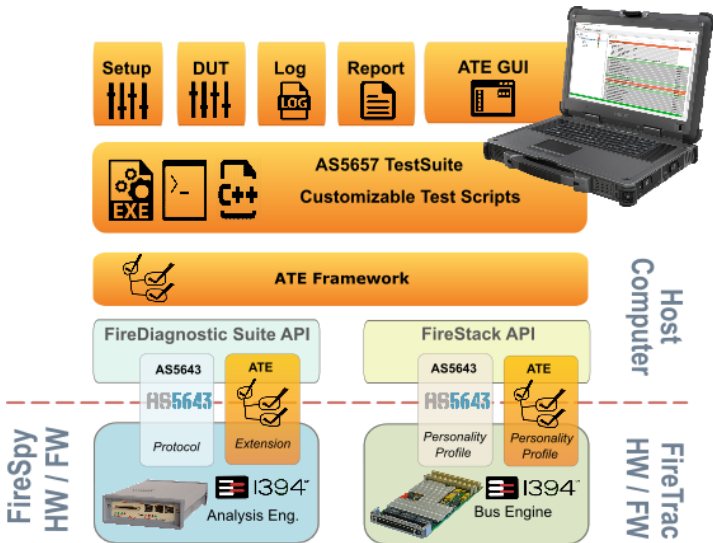
Usage Scenarios

The AS5657-ATE is useful throughout the entire Life Cycle of a product starting at an early stage of engineering, extending through verification and production and even expands into sustainment of a typical avionics program.

Thanks to its expansion capability – the ATE Framework and the test framework can be modified and extended - customization and adaptation to reflect specific project requirements can be implemented easily and with straight forward methods. Usage areas include but are not limited to:

- Requirements Definition
- Software Systems Design
- Software Quality Assurance
- Verification and Validation of hardware modules
- Product Acceptance
- Systems/Network Integration
- Systems Simulation (includes load testing, HIL, PIL, etc.)
- Module/LRU Qualification Testing (e.g. VMC, Flight Control Actuator, etc.)
- ….

Estimating the cost savings of using the AS5657-ATE over doing the verification testing manually is a relatively complex undertaking since the AS5657 testing specification refers not only to AS5643 but also to IEEE-1394 and its related test specification. This makes the overall task involved and also requires a good understanding of the involved technologies. Taking shortcuts and/or voiding these tests as part of an accompanying process during system engineering can require costly rework and cause delays.

A POWERFUL ATE SOLUTON:

The AS5647-ATE is a highly productive and high capacity software package and is organized as a powerful test framework, ATE Framework, together with an application control interface, ATE GUI, and the relevant test definitions for AS5657, AS5657 Test Suite.

AS5657-ATE Test Setup

The following figure shows test configuration used to perform the AS5657-ATE. The Device Under Test (DUT) is connected to a FireSpy bus analyzer and then connected to a FireTrac interface card (see diagram). A FireSpy can be either located in the same PC (using PCI connectivity) or externally (connected via USB).
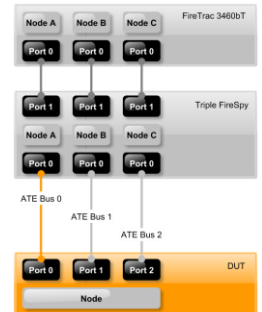

By using a combination of a Triple FireSpy and a FireTrac it is possible to perform an almost fully automated test with very little user input required.

For all specified tests, the DUT will be viewed as a "black box" with the 1394 connector being the only access portal to the DUT. "Black Box" also means that the test person has little or no knowledge about the functions, features, and internal operations of the DUT.

AS5657 Test Suite

All tests are written in C++ and compiled into a Test Suite executable. The AS5657 Test Suite executable is a console application that can be controlled from the included Graphical User Interface, ATE GUI, or it may be integrated in any other test execution framework.
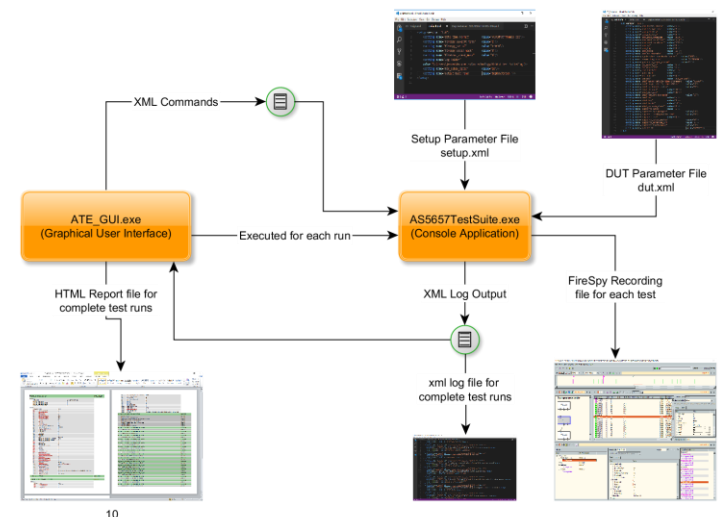
All input and output to and from a Test Suite executable is provided in the form of XML. Upon launching the Test Suite it will load the XML configuration files it was pointed to from the command line and it will start outputting test results in real-time on the console, also in XML format. While the Test Suite is running it will listen for certain special commands in XML format. This could for example be used to abort a running test. In addition to outputting the test results on the console, a log file with the same information is written to file and a FireSpy Recording is saved for each test. When the ATE GUI is used for test execution a test report in HTML format is generated for each test run.

Test Coverage & Program Adaptation

The out-of-the-box AS5657_TestSuite provides full coverage of the tests specified in AS5657 except the Power Class Register test. Where AS5657 adopted tests specified in the 1394Base Test Specification then the AS5657 Test Suite scripts will execute these.

However, it is essential to understand the test case details. Since AS5643 (and consequential AS5657) is only a framework and does not specify all details[1] regarding a specific implementation DapTechnology in general does not have access to or knowledge of such program-specific definitions. Typically the top-level integrator or high-level program partners are in the best position to make suitable adaptations to the AS5657 Test Suite and therefore DapTechnology provides this part of the overall test solution available as source code and editable parameter files to approved licensors. This "customization" can be established by the top-level program integrator and shared with partner companies (thus creating consistent test definitions program wide) or defined by each licensors of the AS5757-ATE solution independently.

Custom Tests and Customization

The AS5657-ATE provides an almost complete implementation of an automated AS5657 test procedure. However, since some parts of the AS5643 protocol are program-specific, it is not possible to include those details in the test source code. Many of those program-specific aspects of the protocol are included in the DUT configuration file format. Some aspects, however, are too complex to be configured using a configuration file. For those tests, the source code itself needs to be tailored to the specific device that needs to be tested. This chapter explains how existing tests can be customized and how new tests can be added to the test suite.

The AS5657 Test Suite makes use of several external libraries. The graphic below provides an overview of external dependencies for the AS5657 Test Suite executable.
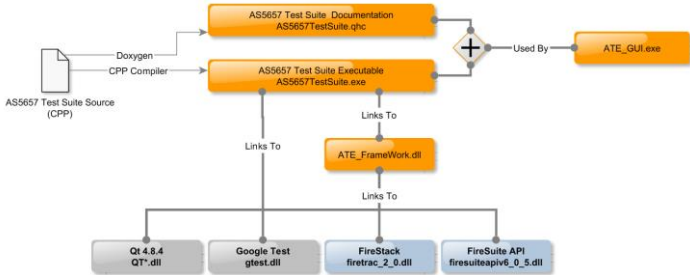
The Test Suite is built on and linked against the ATE Framework. This Framework is based on a software unit testing framework and contains extensions to make it a full device compliance testing framework.

## ATE Graphical User Interface (ATE GUI)

The AS5657-ATE offers a graphical user interface that can be used to select and execute automated tests. During and after test execution the test status can be viewed in specific windows.
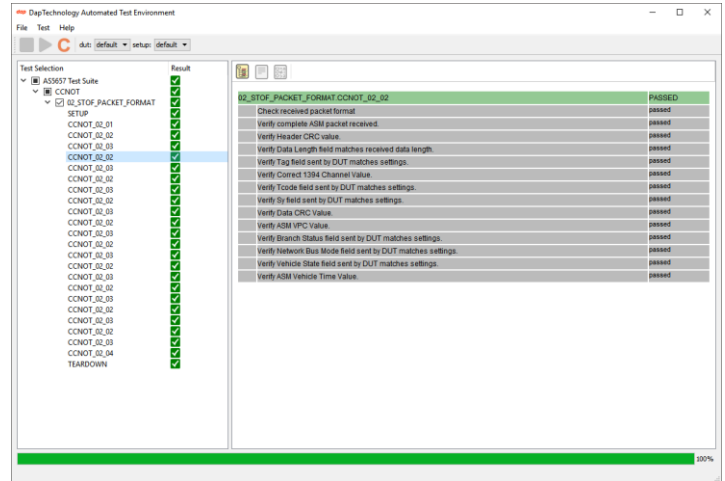
After completing the selected tests, the results can be reviewed in the *test results* tab as shown in the following picture.
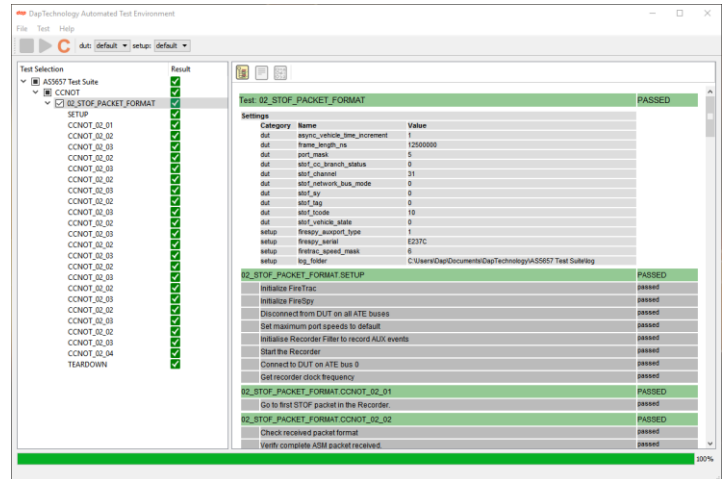
Failed tests are clearly highlighted and explanations are given for each failing test.

---
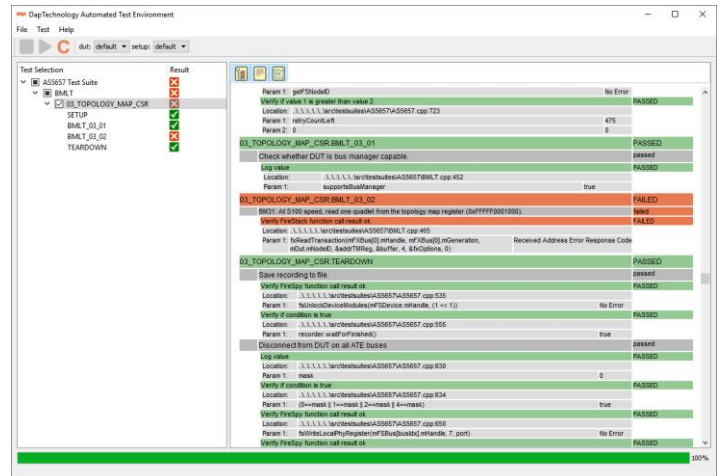
[1] Examples for such program specific definitions typically are the system transmission speed, STOF frame rate and offsets, STOF and ASM packet payload definitions, ….

The following example shows the detailed test result of test LLT.14, step 11. This step of the test checks whether a register value is set to the expected value and consists of one *part*: Verify if two values are equal. The *show parameters* button shows these two values: the variable *cmstr*, which is 1, and the expected value 0. The location tag shows that this part can be found in the *LLT.cpp* file, at line 1065.

14_CYCLE_START_PACKETS.LLT_14_11		FAILED
CM45: Check whether the STATE_CLEAR.cmstr bit is set to 0.		failed
Verify if two values are equal		FAILED
Location:	..\..\..\..\src\testsuites\AS5657\LLT.cpp:1065	
Param 1:	cmstr	1
Param 2:	0	0

Logging and Reporting

After running a test all essential details for the test execution are logged and the test results are rendered in a comprehensive test report.

Command Line Interface

The ATE supports a number of command line options, which makes it possible to control the tests from an external application. Using the command line interface, tests can be selected, started and stopped. All input and output will be in XML format. The XML files with the test output can be found in the log directory as specified in the configuration files. Certain test parts may need user input, e.g. connecting nodes or generating power resets on devices.

Settings

The parameters of the AS5657-ATE can be defined through configuration files. For user convenience, these files will be generated automatically with default values on first use of the ATE GUI. Some of these values may be correct for the used test setup. However, most of the settings allow to be tailored to the setup and device under test, thus providing the most flexible customization and program adaptation ability.

ATE Framework

The ATE Framework is scaffolding that is laid to provide an execution environment for the automation test scripts. The framework provides the user with various benefits that helps them to develop, execute and report the automation test scripts efficiently. Advantage of the test framework are evident in the reusability of code, amount of test coverage, recovery scenarios, low maintenance cost, minimal manual intervention and easy reporting.

The ATE Framework is open to the implementation of other test definitions and provides the ability to generate additional automated testing, which is unrelated to the AS5657 requirements. The provided solution can also be used for production testing, verification and validation of other functions, interfaces, … The implementation is beyond the scope of the package provided by DapTechnology and is fully under the customers control, responsibility and liability.

SPECIFICATIONS:

ATE for	AS5657 - Test Plan/Procedure for AS5643 IEEE-1394b Interface Requirements for Military and Aerospace Vehicle Applications
	20020005 - Base 1394 Test Suite Definition with Extensions for 1394b and IEEE-1394-2008 (in parts as required by AS5657)
Part number:	AS5657-ATE
	Note : FireSpy and FireTrac HW have to either be purchased separately or provided by the user.

Operating System: Windows™ 7 and Windows™ 10

REQUIREMENTS:

- **For running tests:**
  - S400 Triple FireSpy™ with an up-to-date ATE and AS5643 Protocol license
  - S400 Triple FireTrac™ with an up-to-date ATE and Full Personality Profile license
  - PC running Microsoft Windows™ 7 or newer
    - FireTrac™ for Windows™ (latest version) http://www.daptechnology.com/firetracupdates
  - FireDiagnostics Suite (latest version)
    - http://www.daptechnology.com/firespyupdates

- **Additional requirements for test development (customization)**
  - CMake™ 3.7.2 or higher*
    - https://cmake.org
  - Qt™ 4.8.4 Open Source Edition *
    - http://www.qt.io
  - Doxygen 1.8.13*
    - http://www.doxygen.org
  - Graphviz™ 2.28 or later*
    - http://www.graphviz.org
  - Microsoft HTML Help Workshop*
    - msdn.microsoft.com
  - Microsoft Visual Studio™ (2008)*

  *) Specific versions may change with future AS5657-ATE releases.

REFERENCES:

Windows and *Visual Studio* are trademarks of Microsoft Corporation.
Graphviz is a trademark of AT&T INTELLECTUAL PROPERTY, LLC.
QT is a trademark of the The QT Company Ltd.
CMake is a trademark of Kitware Incoporated.
FireSpy and *FireTrac* are trademarks of DapHolding B.V.

CONTACT INFORMATION:

sales@daptechnology.com   www.daptechnology.com

dap TECHNOLOGY •

DapTechnology B.V.
Beatrixstraat 4
7573AA Oldenzaal
The Netherlands
Ph: +31 541 532941

dap USA •

DapUSA, Inc.
780 W San Angelo Street
Gilbert, AZ 85233
United States of America
Ph: +1 480 422 1551