67th International Astronautical Congress (IAC), Guadalajara, Mexico, 26-30 September 2016.

IAC-16-D1.6.6

# RAPID, AUTOMATED TEST, VERIFICATION AND VALIDATION FOR CUBESATS

**Yaseen Zaidi[a]*, Norman G. Fitz-Coy[b] and Robert van Zyl[a]**

[a] *French South African Institute of Technology (F'SATI), Cape Peninsula University of Technology, Symphony Way, Bellville, Western Cape, South Africa 7530*, yaseen.zaidi@ieee.org, vanzylr@cput.ac.za
[b] *Dept. of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611 USA*, nfc@ufl.edu
* Corresponding Author

## Abstract

From the experiences learned in the ZA-CUBE-1 mission, bringing up of a small-scale mission assurance facility is reported. The first of a series of actions envisaged to accomplish the facility is achieving functional test and verification, and the same under modest temperature cycling. The former is initiated by setting up a test and measurement system comprising of the legacy equipment from the previous space project, while a thermal chamber is procured as a first-pass of environmental validation. The test system is driven autonomously using a highly pliant software controller, which is the executable tool for the conceived methodology of systems engineering life cycle and mission assurance. Besides automated electrical and temperature measurements, the software has been crafted to accommodate for Phase B/C deliverables by way of simulations, virtual prototypes, emulation of operation scenarios with hardware tools and the mission software—all unified in a single platform. The system is exploited in validating a S-band transmitter while economizing time, and in obtaining valuable insight in transmission performance over thermal loads, which may result in revising the mission requirements and impacting satellite system parameters. The goal of the work is to shorten the iterative mission engineering in top-down and bottom-up cycles, through automation and, ultimately ensure substantial consistency and traceability in the design flow.

**Keywords:** Automation, Automated Test Equipment, CubeSat, Mission Assurance, Verification, Validation

### Acronyms/Abbreviations

Automated Test Equipment (ATE), Verification and Validation (V&V)

## 1. Introduction

CubeSat Development is characterized by decisions in mission objectives, system requirements, payload/bus configurations and launcher selection that are often revised late in the engineering cycle. In order to accelerate the development and to readily adjust to the project dynamics, the French South African Institute of Technology (F'SATI), through a systems engineering approach, is proceeding to develop a broader scope mission assurance facility.

The vision behind the facility is about cutting the engineering cycle by overlapping the iterative phases of design, development, test, Verification and Validation (V&V), as soon as the prototypes are available. Functional testing of assembled and integrated hardware and software units can be performed rapidly before subjecting to environmental qualification. The facility is conceived to achieve high coverage of design verification. From the success and the lessons learned through the ZA-CUBE-1 mission, the expertise and legacy equipment are being reused to piece together a test automation setup in a phased way. This approach is adopted to avoid a hefty and immediate one-time investment. In the meantime, technical needs analyses are under way for the procurement of the remaining elements that will upgrade the facility to full compliance toward suitable qualification level of space systems built with the Commercial Off The Shelf (COTS) grade components. The priority of future qualification is on mechanical acceleration loads and vacuum cycled thermal loads.

The main elements of rapid V&V systematic scheme are the "Missurance" measurement system suite and an instrumentation bus hosting a variety of physical layer interfaces and communication protocols. The usability, expandability and the re-configurability of the networked test apparatus in the automation setup are of priority. Missurance is built around an integrated development platform with libraries for measurement, analysis and User Interface (UI) design. Missurance allows reliability and determinism in the electrical measurements. Missurance is a modular architecture that has product, test suite and Automated Test Equipment (ATE) centric views. Depending on the user requirements, test applications can be configured to a specific product or test, or simply a specific ATE. Missurance may also be re-built for limited applicability, e.g., last minute sanity check of the battery capacity. Due to the variety of interfaces in the pool of ATEs, the bus supports high accessibility of the measurements including IEEE-488B, USB ($I^2C$), LAN/TCP, UART, RS-232, custom wired and wireless links, which may be needed at certain test access points.

The literature on nanosatellites lies in three categories. First category [1,2,3,4] is of the emergence of nanosatellite trend and survey papers, which investigate recently launched or planned systems, their capabilities, on-board technologies within the CubeSat Design Specification constraints and the possibilities in the space data services from these missions. The second category is of papers on the techniques for the modeling and analysis of sensors, actuators or a subsystem. This seems to be a pervasive theme. Numerous study cases are reported on variety of hardware e.g., solar panels, attitude control schemes and hardware in the loop controls. These studies are narrow in scope and lack a generalized methodology for the broader process of mission design and development. The third category is related to mission failures, whether review of missions or probabilistic analysis of system reliability [5,6,7]. According to [7], in a survey of 2500 spacecraft failures, 25% of the problems were attributed to design (functionality), 20% due to the environment and 24% were parts and quality related. Given that the objective of mission assurance is failure minimization, the works do not shed light on the principles and procedures that the CubeSat developers community may adopt. The effort outlined by Cho et al. [8] is a first account of comprehensive tests contemplated for the nanosatellite mission success. This work is largely environmental qualification related and does not address functional testing; the dominant and urgent focus of the developers working with budgetary constraints, and omits assurance methodology for missions on short timetables. The portrayal in [9] is also about a quick and light facility. It, however, targets linked observational assets built for space and aerospace domains such as network of CubeSats and UAVs. The body of work is a concurrent design center, at a leaner scale than existing ones at JPL (Team X) and GSFC (MDL). Thus, confines to Phase-B level of concept development and iterates using the established methods, processes for compatibility reasons and the experience of NASA. With the present introductory work, we hope to instigate interest in the development groups to share ideas, experiences, replicate methodologies and for the new entrants to start at a higher foundation.

In section 2, we discuss the model development in a CubeSat project that essentially sets the scope of testing and verification, the required levels and respective checkpoints. In section 3, we concisely narrate our plan of action for CubeSat test and verification. Section 4 recounts the details of the framework developed toward execution of the test methodology. The section 4 is about the elaboration of the framework and provisions made therein. We hint at the rationale of the test philosophy for speedy and accurate test suites that record reproducible measurements and especially by enlarging the test scope by incorporating prototype or virtual hardware/software and development tools as well as the extensive network of the instruments to the framework. An important dimension noted in section 4 is the software life cycle and the accommodation of change, which is supported in the framework through modularity and adaptablity. The framework is applied on a S-band comms unit in case study section 5. In section 6 the results are studied, section 7 presents a prospect on next steps toward expansion of the framework and lastly in section 8 we remark major observations from this authorship.

## 2. CubeSat Model Philosophy

The test philosophy directly stems from the model philosophy. The general model philosophy in low cost missions follows below.

### *2.1 FlatSat Model*

The FlatSat is a crude model comprised of various subsystems, typically modules of CubeSat and PCBs or breadboards of certain functionality, distributed over a workbench. A flatbed base with power source, electrical interfaces and debug support may be used instead of spreading out the units and circuit boards on a table top. The FlatSat is a primitive attempt of assembly, integration and test to electrically integrate various assemblies and the first ever hardware software interaction to functionally check out basic device fusion. For example, during livening up the on-board computer, the system boot-up sequence, peripheral drivers, telemetry collection and correct functioning of the commanding may be verified. The FlatSat configuration provides high accessibility of various electrical interfaces in order to debug software and troubleshoot hardware problems, which are essential to initial respiring of the embedded design. Much of the testing performed in a FlatSat model is manual. The PCBs used on a FlatSat model may or may not be in flight/actual form factor. Several testsets are used to probe expected signal levels, interrupts and timing delays. The outcome from the FlatSat exercise is to size the CubeSat modules, with basic functionality, in mass, power and fitting in CubeSat dimensions. The performance of the design may be mere satisfactory which may be boosted when converted to the actual form factor.

### *2.2 Engineering Model and Engineering Qualification Model*

The Engineering Model (EM) is the integrated model in the form factor specified in the CubeSat Standard. The EM includes the updates since FlatSat testing, whether hardware, software, interface or performance related. If the EM is qualified with environmental tests, a copy of EM built after improvements from the environmental test results is the Engineering Qualification Model (EQM). The EQM is tested, verified and validated for acceptance-level qualification, but lightly. The EM undergoes, depending on budget constraints and previous design experience, thermal vacuum, sine and random vibration testing to the acceptance criteria, i.e., at the specified margins in the criteria, well below the qualification levels. In the nanosatellite arena, the EM is seldom qualified for the qualification limits, unless there are absolute mission requirements and budgets available.

### *2.3 Flight Model*

In larger satellites, the Flight Model (FM) is built with components that are screened for the flight standard. However, in nanosatellite, the copy of EQM with minor updates is the FM. The FM may or may not be tested for acceptance levels.

### *2.4 Test Philosophy*

For standard payloads and smaller U's the models discussed in sections 2.1, 2.2 and 2.3 may be adequate. However, for larger CubeSats and CubeSats carrying advanced and sensitive instruments, additional models may be required and subsequently optimum number of tests too. This decision rests with the systems engineering organization.

The CubeSats are generally low cost satellites built with COTS Electrical, Electronic and Electromechanical parts. Thus, the CubeSat designs for ordinary payloads such as standard camera or automatic identification system in the low earth orbit, at the very minimum, should be qualified for the models discussed above. For advanced missions e.g., intended for the deep space where environment will be harsher, sensitive payloads e.g., mechanized optics and sophisticated subsystems e.g., propulsion or mechanisms for deployable appendages—higher levels of qualification are expected. During the environmental testing of the Engineering Qualification Model (EQM), expensive hardware e.g., solar panels, GPS or any other modules that have been qualified by the manufacturer prior to delivery for acceptance-level testing, may be removed.

No particular test sequence is generally preferred or recommended. Since the purpose is to find defects as early as possible, any test may be performed as soon as module/unit level hardware becomes available. This decision often rests with the availability of test setup. For example, in testing for radiation, the beam use time is negotiated well in advance. Typically, thermal testing can be performed with relative ease as functional testing is being carried on. This is especially useful to benchmark communication ICs and frequency dependent or jitter sensitive circuits that may erratically behave in certain temperature band.

The EQM-FM approach is preferred for CubeSats [10] for utilizing spare parts of EQM in FM if it misbehaves toward the launch campaign, provided the project budget allows building EQM after an EM. This is especially useful if the EQM is able to tolerate non-destructive evaluation and since behaviorally and environmentally the EQM is nearly equivalent to the FM.

The level of mission assurance needed in a low cost space mission is a difficult question. The constraints presented by a low cost mission, weather, commercial grade components or unavailability of adequate qualification facilities have not stopped CubeSat missions from flying. No matter what level of validation or qualification is applied, functional testing and verification are always comparatively less expensive, safe and ensure correct operation of the mission at ground conditions. In many applications, a multitude of failures is traced to ab initio functional design. For example, in the Application Specific Integrated Circuit (ASIC)/FPGA industry, a survey revealed that as many as 60% of very expensive respun of the ASIC chips is due to the functional design errors. In over a decade, this trend improved to about 48% [11]. Functional flaws in space systems are no different as affirmed by [7] for the larger satellites. We infer nanosatellite may not be too far either. We plan to steer our resources to full-scale functional test and verification.

## 3 Test, Verification and Validation (V&V) Strategy

The test and V&V approach at F'SATI is the lowest cost by developing minimal number of models, i.e., FlatSat-FM where the FM is in fact the EM that is functionally tested and environmentally qualified for acceptance criteria. The indigenous units of EM are qualified for lower than commercial temperature range and characterized for different products features or modes. Therefore, at the unit level, we emphasize testing, V&V and acceptance-level thermal qualification. The unit level testing is automated for evaluating performance in various operation modes and for thermo-functional testing from -20°C to +50°C in a single cycle at 10-minute dwell time in 10°C steps. This fair specification approximately corresponds to one orbital period in low earth realm, and the extent in time is consistent with the eclipse (1/3) and sunlit (2/3) fractions of the orbital period. Moreover, the dynamic range fits well with the ambient temperature variation encountered in revolutions at the preferred low earth altitudes of nano missions. At system level, we emphasize functional testing of the integrated satellite and only acceptance level thermal bake-out. This is minimal mission assurance in a low cost project. Among the in-orbit and in-flight environmental loads, the heat loads are consistently regular and deterministically act throughout the operation which derates COTS components. This is in no way to belittle the influence of other loads. The damage due to radiation bombardment is probabilistic. In respect to the launch vibrations, the loads are severe and vary from instantaneous pulsed, low to high frequency periodic and random, and ascending in magnitude as much as to 10g, but their cumulative application time is less than 10 minutes. Our manufacturing process of the circuit boards and workmanship of the component mounting is from a source whose quality has been proven in the previous space flight. Therefore, in the current mission we plan to test the EM lightly on a single axis shaker. The EM will undergo restricted thermal vacuum bake test; possibly a single cycle thermal vacuum test as well. The urgency of thermal vacuum testing is for the temperature sensitive infra-red imager. The optics focus requires calibration and the lens mount requires adjustment and alignment for the use in space under vacuum temperature cycling conditions. The radiation test will be a total ionization dose test with a proton source.

## 4. Test and V&V Framework

### *4.1 Missurance*

Toward rapid functional testing, requirements verification and validation of the units under severe temperature conditions of the physical environment, software driven automation is applied. The software suite is called Missurance. Fig. 1 shows the main view of Missurance and the bussed communication topology.

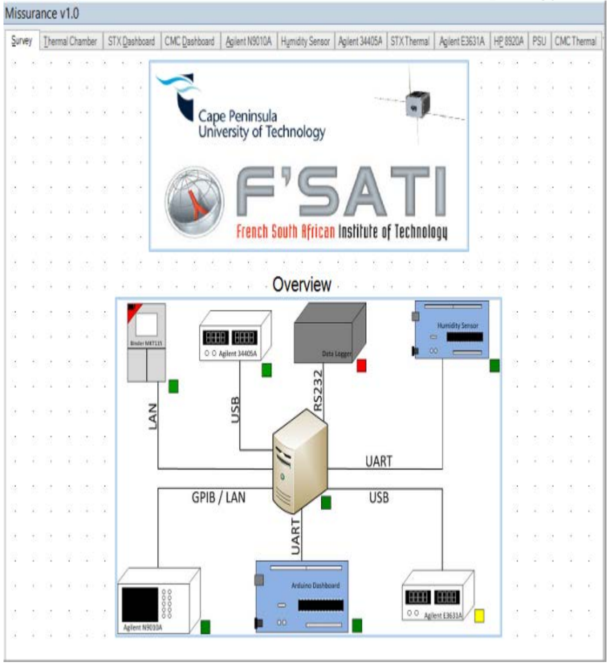


Fig. 1. Missurance main GUI, tabular views and overview of the data interfaces

### *4.1.1 Missurance Architecture and Libraries*

Functional testing is performed at unit level assembly, without mating the remaining units belonging to the subsystem. At this stage, few design decisions and even some system requirements might remain to be frozen. Since significant development is settled, while integration of software/hardware may give rise to certain tweaking, the functional testbed must be highly adaptable and usable. Rapidly and iteratively changing system configuration across the design chain is driven through modeling, simulation and in embedded hardware and software, which gives rise to proportional change in the test setup. The elements of functional testbed are system modeling tools, simulators, emulators, electronic design automation tools for hardware design, Integrated Design Environment (IDE) tools for software development, debuggers, scripting tools, test fixtures, a miscellany of test equipment, the instrumentation bus, test computer and the satellite modules being the Device Under Test (DUT). Clearly, bringing together assorted and diverse tools into a software-controlled setup is a sizable undertaking.

### *4.1.2 C based Development and Testing*

Missurance is an evolving and futuristic test and V&V software. It is projected to integrate various facets of engineering development to a common platform. It is an event-driven, human computer interface environment. The current version executes three loops: configures the DUT, performs functional testing of the DUT using several ATEs and optionally runs the functional test suite under a programmed thermal cycle. At its core, the Missurance runs deterministic and precision execution of test suites. Missurance is developed as C/C++ application. The advantages of C based Application Programming Interface (API) are tremendous. The mission related algorithms developed in MATLAB/SIMULINK may be cosimulated and imported [12,13] e.g., the DUT may be subjected to in-orbit wattage profile under eclipse and sunlit periods, mission simulators may be connected via TCP/IP sockets and therefore simulation data from mission scenarios may be fed to the DUT if required. If a physical FPGA device is still not fully developed, then relevant HDL blocks may be accessed at their ports as C data types and either read in or written to during the testing through the HDL simulator's procedural interface, VPI/VHPI [14,15]. The procedural interfaces allow register transfer level simulation coupling through simulator's C interface to the external test and verification system. For example, the HDL code of the software defined radio FPGA may be coupled as virtual prototype in the Missurance framework. Through test and development, the Missurance application code may be reused to develop a CubeSat simulator. For precision timing needs, the C code reuse, libraries and the IDE's Real-Time Module allow deployment of run-time applications from Windows/PC host to the dedicated hardware. For example, ARM, x86, PowerPC platforms, operating systems such as RTOS, VxWorks and even development tool chains such as Eclipse, Linux/GNU—all have native C interfacing to Missurance. Such support in a CubeSat mission to the embedded engineering is expedient for microprocessor or DSP based modules e.g., the OBC or Attitude Control Computer. A desired aspect across a complex tool chain is support for debugging, which the IDE provides through viewing the execution profile in monitoring memory use, tracking resources, task allocation, spawned processes, thread tree, variables, call stack and setting a watch. Initial release of the software was sequential programming that recently has been changed to multi-threaded style for context-switching and better resource allocation while executing simultaneous and multiple tests on a variety of ATEs and avoiding bus time outs as well as the wait states of ATEs to finish their measurement tasks. The eventual aim of Missurance is to perform automatic measurements; software development under C provides a dandy interface to the instrument's drivers since most ATE manufacturers supply open source C drivers for fine controlling their testsets.

### *4.1.3 Features and Interfaces*

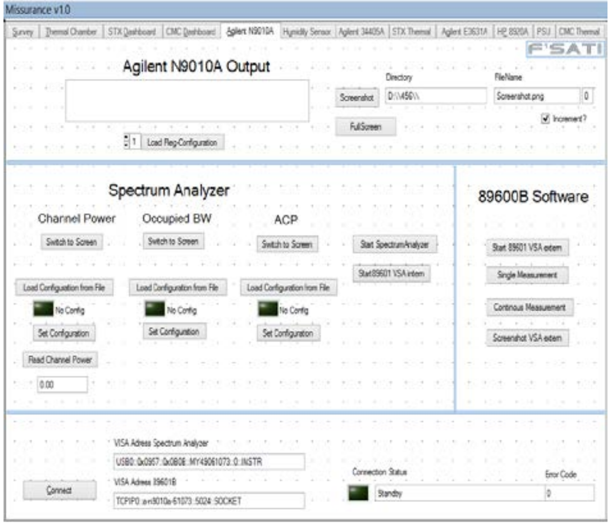
Due to changing requirements and mission dynamics in CubeSat projects, late mission decisions and therefore late design changes as well are typical. Keeping the foregoing in view, the architecture has been conceived to cater agility in test and V&V methodology and for long-term system diversity needs in the CubeSat program. To this end, simplicity of the software architecture has been the main driver. We now discuss distinct facets of the architecture.

### *4.1.4 Modularity*

The test scenarios are triggered with a user driven event(s), followed by several procedural events. The user generates a series of events by clicking check boxes, enabling buttons, selecting from widgets or by loading test inputs from a file. These events, in turn, execute callback functions or procedures that are data driven and are dynamically generated by the actual test code. The order of the events creates a test logic or an execution schedule of the functions and procedure in a particular test flow. In any form of hierarchical nesting, modularity is necessary to seek relationships of the functions and their dependencies on each other and for the overall control of the application. Modularity in code and the level of functional independence also ensure code sharing and reuse.

### *4.1.5 Tabularity*

The UI is a tabular interface. The tabular approach was adopted to maximally utilize the screen real estate as the number of options required to configure a specific test and to setup the related ATEs may take up significant area. The tabular approach allows navigation between different tabs if multiple ATEs are required. The underlying principle for using tabs is abstraction layers. Three layers are possible. In the ATE centric view, the information related to a specific test instrument is encapsulated. Fig. 2 shows the instrument centric view in which all relevant tests of a particular instrument are collated. A user may have a need to run power measurements using a tracking power supply. The instrument centric view would allow setting up the power test. In the test centric view, multiple related tests are grouped together e.g., evaluation of modulation impairments on a particular band of frequencies, Bit Error Test (BER), Occupied Channel Bandwidth (OCBW), Adjacent Channel Power (ACP) and Error Vector Magnitude (EVM) measurements are organized. The highest and most elaborative abstraction is the product centric view in which all information, whether the functional test, DUT configuration, ATE configuration, visual displays or thermal chamber commanding, is bundled together. The product centric view is especially favored when a baseline product has multiple variants, each of which require a corresponding test suite. In this way, the test suite is directly traceable to the product requirements. For example, UHF/VHF units have different options in commercial/amateur frequencies, carrier frequency, RF power levels and bus voltage. All variants of this product are independently tested with own test suite. Any change in the product resulting in a new variant inherently requires replication of the closest parent test tab.



Directory: D:\\456\\

FileName: Screenshot.png 0

☑ Increment?

1

0.00

VISA Adress Spectrum Analyzer: USB0::0x0957::0x0B0B::MY49061073::0::INSTR

VISA Adress 89601B: TCPIP0::a-n9010a-61073::5024::SOCKET

Connection Status: Standby

Error Code: 0

Fig. 2. ATE centric view of instrument specific controls

Besides separation of concerns, one reward of enforcing tabular design is that multiple engineers can separately develop test suites that are integrated into the main design. For example, S-band and C-band unit tests may be developed severally.

### *4.1.6 Usability and Reusability*

High usability is essential for frequent use of Missurance, ease of use has been given proper due. In particular, the UI hides the underlying detail of test and verification. E.g., the setting up the ATE, its address and test specification can be encapsulated in a single button. Many test methods use several same ATEs and test logic. Common code, callback functions and UI controls have been reused to quickly develop new test code. Exception handling and error codes are translated to meaningful messages to precisely locate the problem and ease the troubleshooting.

### *4.1.7 Scalability and Adaptability*

Missurance is scalable. In many situations, a limited version of the testware may be suitable to apply test scenarios to a particular DUT. For example, a lighter version, which only checks OFF/ON status or responses to the telecommands in situ launch vehicle, may be compiled for the launch campaign. Similar needs may arise for a suitcase model for comms testing with the electrical ground support equipment.

### *4.1.8 Maintainability*

Missurance is a long-term living software that is maintained for continuous improvement. Since the methodology assists in different project phases and testware is being developed to qualify both modules/subsystems and the full satellite with varying payloads, software maintainability is a necessity. It is given that the requirements will change and the products will be customized to customer's missions. As RF front-ends and matching filters are tuned to specific transmission requirements, the Engineering Change Orders and Bill of Material are processed for parts going obsolete, new FPGA codes are released or simply circuit designs and board layouts are improved for low power consumption, better noise floor, signal integrity, thermal planes or cost cutting reasons—maintenance of the testware becomes all too crucial and equally challenging. The changes are usually minor; nevertheless, the entire suite needs regression tested. Maintenance is also required as development platforms and instrument driver libraries undergo frequent updates.

### *4.1.9 Automation*

An exhaustive narrative of experiences in test automation, benefits and pitfalls in a variety of industries and application is in [16]. The main advantage of automation is deterministic and iterative evaluation of functionality and the relative ease of changing the test inputs. Long and tedious thermal testing may be carried overnight or unattended.

The automation generates detailed test reports against the input test conditions, unit configuration and time logs. In addition to the measurement values, screen shots can be captured on advanced ATEs. This aspect is very useful on the ATEs that display measurements on vector data. Efficiency is time, cost control and resources are additional gains. The automation does not target finding hardware/software bugs because, this should happen prior to the automation, although such a concealed and late find will be an added advantage. The thermal testing, however, is exploratory and may bring about circuit peculiarities and sensitivity to certain temperature, after all, that is the objective.

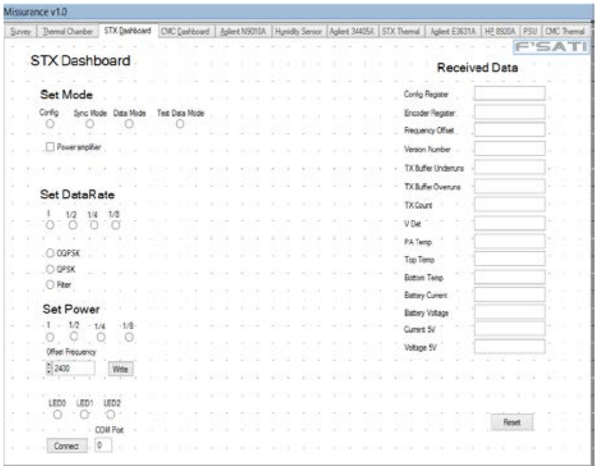
### *4.2 Instrumentation Bus*

A number of physical interfaces and protocols enable acquisition of measurements and commanding of the instruments. These interfaces aid the testware hosting computer's access to the entire testbed comprising of ATEs, environmental chamber, test jigs and the DUT. Apart from the standard bus protocols, certain equipment support propriety frame formats e.g., datalogger of the thermal chamber uses Modbus packeting over TCP/IP. This exclusive messaging has been implemented as part of the instrumentation bus. The de facto interface in the test and measurement industry is the General Purpose Interface Bus (GPIB) [17], which we also use. Almost any other physical interface from GPIB is commercially available in both software and hardware. The degree of equipment connectivity and the richness in driver software around GPIB are the fundamental enablers to expand automation campaign over more comprehensive electrical-functional test and verification coverage.

The expansion of the test network is an on-going process as instrumentation on the bus grows. Ensuring robustness and stability of the bus is an exacting task due to the difficulties arising during test execution in time outs, buffer overflow and devices unresponsiveness, which can hang the testware. These problems are solved by careful implementation of buffering, interpreting acknowledgements and the expected responses from the instruments and as well as repeated live running of the Missurance.

### *4.2.1 DUT Re-configuration*

The V&V coverage is carried out to the complete feature set of the test article. This requires on-the-fly re-configuration of the DUT especially if the performance is to be characterized to the full thermal range. DUT re-configuration is an important step before the execution of the test suite and therefore the necessary interfaces to the electronic fixtures (dashboard) that enable setting up the DUT have been programmed. In the comms payload, re-configuration is by setting the modulation schemes, RF power levels, carrier frequency, data rates, transmission modes and software protocol interfaces (AX.25/$I^2C$). These combinations are autonomously loaded in the FPGA using an Arduino device during execution of the test loops. For example, the command transceiver is an integrated UHF-transmitter and VHF-receiver. The transceiver implements both 9600 bps GMSK and 1200 bps AFSK and operates in full-duplex mode. The transceiver transmit frequency is 400–420 MHz. The frequency of operation is software selectable within the band and is adjustable in 25 kHz steps. The output power is adjustable from 27 to 33 dBm. These selections in the DUT are made using the respect dashboard tab. Fig. 3 illustrates one such tab.



Offset Frequency: 2400

COM Port: 0

Fig. 3. DUT Configuration Interface (dashboard)

### *4.2.2 Abstractions in Instrument Access*

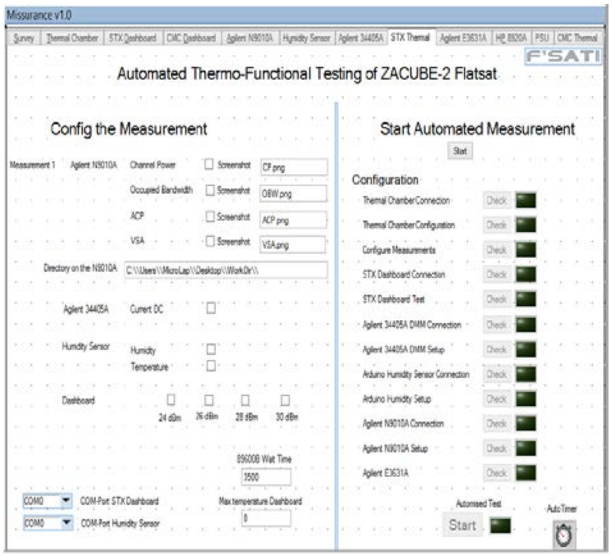
Three layers of access to instruments are possible depending on the interface complexity. First and the highest level is the manufacturers or third party high-level driver, typically a C driver. The second level is the Virtual Instrument Software Architecture (VISA) specification [18], an I/O API of the test and measurement industry for communications, maintained by the IVI Foundation. The most primitive and lowest level and perhaps the most useful one is the legacy Standard Commands for Programmable Instruments (SCPI) [19], also maintained by IVI Foundation. The SCPI language is a pure commanding syntax that is independent of underlying hardware and thus wrapping it in any higher-level language is doable. If communication with the instruments using other modes fails, the SCPI usually works.

Fig. 4. S-band Transmitter (STX)

## 5. Comms Payload Testing

The STX in Fig. 4 is a compact S-band transmitter designed for CubeSat nanosatellite missions. It is compatible with the CubeSat nanosatellite standard, with a CubeSat Kit PC/104 form factor. The STX implements QPSK or OQPSK modulation with Intelsat IESS-308 based encoding which ensures compatibility with commercial ground segment receivers. The STX frequency of operation is selectable from 2.2 – 2.3 GHz (commercial band). The frequency of operation is user selectable within the band. The carrier frequency is adjustable in 500 kHz steps. The output power is adjustable in 2 dB steps from 24 dBm to 30 dBm. The STX is configured via an $I^2C$ data bus and high-speed payload data is sent via SPI. Data transmission rates of up to 2 Mbps are supported with ½, ¼ and ⅛ rate modes.

The Missurance framework is employed to evaluate radio transmission performance of the STX in -20$^o$C to +50$^o$C temperature range for a duration resembling the sunlit and eclipse periods in the sun synchronous orbit propagation. The three test loops discussed in section 3.2 are unified in a single tab “STX Thermal.” The source code relevant to the individual tabs has been reused as the procedures and the calls undersurface the UI elements of the superlative STX Thermal tab. In Fig. 5, the illustration of this tab has controls related to the three loops: the DUT configuration loop (dashboard functionality), the STX DUT functional test loop and the outer thermal test loop, inside which both inner loops are executed i.e., the DUT is re-configured for different features and communication channel performance is measured in the specific temperature increments.



CP.png
OBW.png
ACP.png
VSA.png
C:\\Users\\MicroLap\\Desktop\\WorkDir\\
3500
COM0
COM0
0

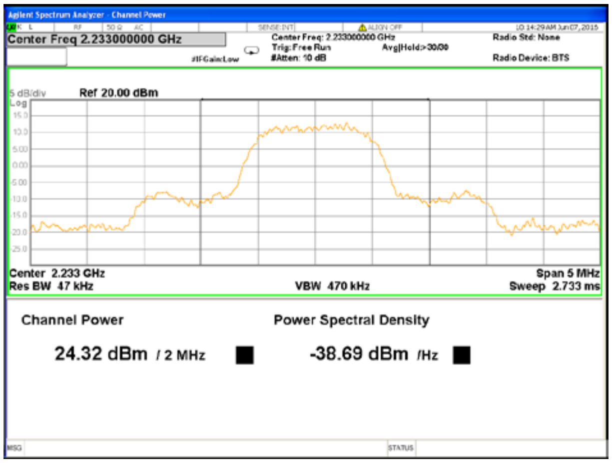
Fig. 5. Automated, combined functional and thermal testing of S-band comms payload

## 6. Results

For the type of digital modulation schemes in the STX, the figures of merit of the RF transmission are based on the Vector Signal Analysis (VSA). Therefore, the choice of testset is a spectrum analyzer with VSA capabilities in measuring modulation impairments in the IQ vectors.

Several performance parameters of the RF transmit power at four possible levels are measured over the temperature cycle.

Fig. 6 is a measure of channel power [20] of 24 dBm at 10.4°C. $I^2R$ loss amounts to about 1 dB power loss in the matching network. Fig. 7 shows ACP which illustrates the leaked power above or below the main channel.



Center Freq 2.233000000 GHz
Center Freq: 2.233000000 GHz
Trig: Free Run
Avg|Hold:>30/30
#Atten: 10 dB
#IFGain:Low
Radio Std: None
Radio Device: BTS

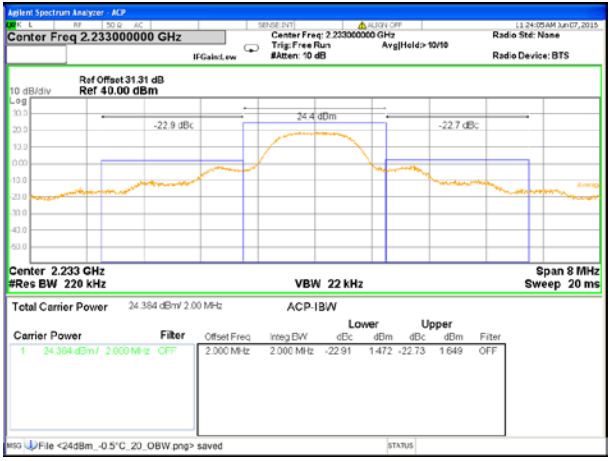
Center 2.233 GHz
Res BW 47 kHz
VBW 470 kHz
Span 5 MHz
Sweep 2.733 ms

Channel Power
24.32 dBm /2 MHz

Power Spectral Density
-38.69 dBm /Hz

Fig. 6. Channel power at central frequency measured at 10.4°C

The measurements reveal the performance of the front-end filter in passing the desired band only and cut offs to the nearby channels, failure or which may cause RFI interference to other circuits and susceptibility to the transmitting channel itself. The ultimate bearing of ACP is on BER, ISI and SNR.



Center Freq 2.233000000 GHz
Center Freq: 2.233000000 GHz
Trig: Free Run
Avg|Hold:>10/10
#Atten: 10 dB
IFGain:Low
Radio Std: None
Radio Device: BTS
Ref Offset 31.31 dB
Ref 40.00 dBm

Center 2.233 GHz
#Res BW 220 kHz
VBW 22 kHz
Span 8 MHz
Sweep 20 ms

Total Carrier Power 24.384 dBm/ 2.00 MHz

ACP-IBW

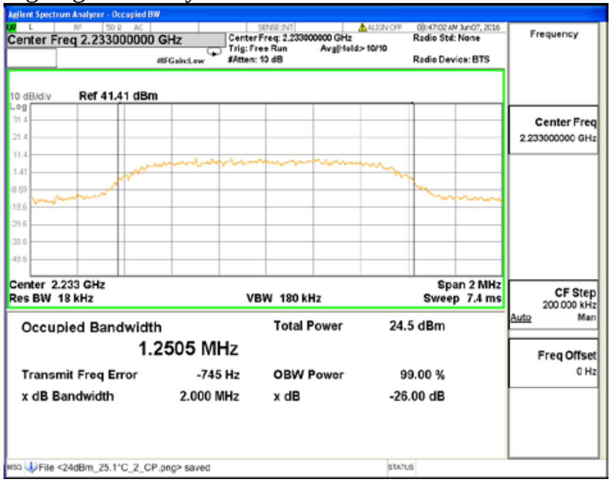
| Carrier Power | | | Filter |
|---|---|---|---|
| 1 | 24.384 dBm/ | 2.000 MHz | OFF |

| | | Lower | | Upper | | |
|---|---|---|---|---|---|---|
| Offset Freq | Integ BW | dBc | dBm | dBc | dBm | Filter |
| 2.000 MHz | 2.000 MHz | -22.91 | 1.472 | -22.73 | 1.649 | OFF |

File <24dBm_-0.5°C_20_OBW.png> saved

Fig. 7. Adjacent channel power at central frequency measured at -0.5°C

The percentage of total power transmitted, usually 99%, over the transmitted frequencies is OCBW. The criteria was developed by Carson for analog frequency modulation in which infinite number of sidebands are produced, however, most energy is concentrated around the carrier frequency. For properly tuned transmitting channels, low in distortion, occupied bandwidth power is characterized by filtering and modulation type. In Fig. 8, 99% of the power is contained within 1.2505 MHz. In various transmission standards, the occupied bandwidth is typically specified at a specific symbol rate. Thus, this measurement should be complemented with a corresponding symbol rate. The bottom right quadrant in Fig. 9 gives the symbol rate.



Center Freq 2.233000000 GHz
Center Freq: 2.233000000 GHz
Trig: Free Run
Avg|Hold:>10/10
#Atten: 10 dB
#IFGain:Low
Radio Std: None
Radio Device: BTS
Ref 41.41 dBm

Center Freq 2.233000000 GHz
CF Step 200.000 kHz
Freq Offset 0 Hz

Center 2.233 GHz
Res BW 18 kHz
VBW 180 kHz
Span 2 MHz
Sweep 7.4 ms

Occupied Bandwidth
1.2505 MHz

| | | | |
|---|---|---|---|
| | | Total Power | 24.5 dBm |
| Transmit Freq Error | -745 Hz | OBW Power | 99.00 % |
| x dB Bandwidth | 2.000 MHz | x dB | -26.00 dB |

File <24dBm_25.1°C_2_CP.png> saved

Fig. 8. Total occupied bandwidth at carrier frequency measured at 25.1°C

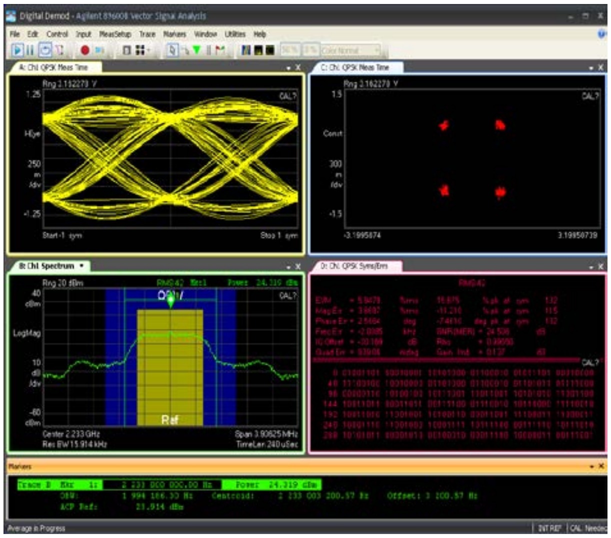


Fig. 9. Transmission impairments at 24 dBm

Fig. 9 also depicts the capture of the signal quality in constellation and eye diagrams. Several transmission impairments are recorded: frequency, phase and gain errors, measured IQ error with respect to the ideal symbol location, RMS % value of EVM and the waveform quality factor ρ. These metrics help characterizing the modulation quality and effect of linearity of the Power Amplifier (PA), which is a critical component of the RF transmission chain. As more DC current is drawn to output the RF power, the self-heating of PA becomes crucial in nonlinear analysis. In summary, the stability of transmission is confirmed in only small deviation of impairments over temperature range. The integrity of the back-end digital modulation (FPGA) and the containment of the spectrum re-growth through the impedance matching filter/network in the front-end are validated. DC-to-RF power conversion (PA) is determined for 4 programmable power levels which show 20-25% of efficiency over temperature at different RF power levels. The PA performs better at 30dBm, which could be improved by losing spectral efficiency. Tuning a PA for wider band is a difficult problem and biasing plays a pivotal role. Adjusting the biasing voltage for the desired frequency band and achieving the right spectral spread, while controlling the impairment parameters and paying the price of heat dissipation is the design space exploration problem.

The long and short of the exercise is that the temperature measurements give out a range of parametric boundaries of the RF transmission. It is up to the designer to perform trade analysis and optimize the parameters according to the criteria, whether customer requirements, telemetry/telecommand communication, high rate payload data download or simply a basic RF subsystem which lends itself easy to the power subsystem and the whole satellite.

## 7. Discussion

From the measurement shown, the mission design and systems engineers can determine the duty cycle of the S-band communication subsystem, i.e., the largest time window in which the PA can operate to transmit data to the ground segment before excessive heating begins deteriorating the channel performance or before the thermal shut down of the PA. Such insight plays a rule in the link budget, SNR $(E_b/N_0)$, Equivalent Isotropic Receive Power (EIRP), receiving antenna gain, receiver sensitivity and $G/T$ figure, to name a few.

Other questions surfacing from the corresponding DC current measurements may require refining the power budget, thermal design margins of the satellite, heat sinks in the PCB layout, so on and so forth.

The next release of Missurance is planned to deliver orbital power load emulation to the CubeSat electrical power supply with the in-orbit load conditions and application timing. This addition will help in finalizing power management algorithm and calibration of photovoltaic cells. Programmable power supplies will power up the satellite power supply, acting as if the power is coming from the solar panel. The algorithm to control the source power will be programed as the solar angle and orbit angle β change during the orbit propagataion. Another feature would be to reuse the power supply coding to induce current in 3-axis Helmholtz coil cage. The idea is to dynamically produce geomagnetic field the CubeSat will experience in-orbit. Beside these, work is in progress to introduce new tabs for variants of current comms products and especially higher data rate modules.

## 8. Conclusions

Due to small engineering cycle of CubeSat program, a framework for mission assurance is presented which is non-sequential. Since Cube mission are largely developed with the embedded systems, integration is at the forefront of design cycle, which translates to co-engineering i.e., co-design, co-verify, co-optimize, co-constrain, co-explore essentially co-anything if not everything. The Missurance application, which currently has test and verification focus, is planned to grow as a multifaceted, versatile and general purpose mission assurance tool that serves throughout the project cycle. The test system allows robust connectivity of the test apparatus, both in software protocols and in hardware interfaces. The gain in saving time in monotonous thermal testing is extra. Automation alleviates manual testing; repeatable and reproducible measurements can be obtained. Due to possibility of feedback from the test campaign to the system redefinition or refinement, the methodology and framework conforms to systems engineering V model.

Test and measurement system will be gradually augmented to host more test equipment and testability features in the Missurance testware. The aim is to evaluate functional and electrical performance through physical layer level testing of all subsystems and eventually of the integrated EM of the CubeSat. After that, the EM will be subjected to the thermal characterization.

In the present paper, we have stressed the functional test and verification approach by way of aggressive, automated coverage, followed by thermal testing. CubeSat mission engineering process is hardly sequential. The presented methodology is test and measurement oriented and supports accelerated V&V. The framework of the methodology aims to integrate simulation, emulation, design and development, thereby achieving co-engineering. The purpose has been to ensure thorough validity of the functionality of the units first and then the integrated satellite system, while operation in the thermal environment is ensured more at the unit level and thinly at the system level.

## References

[1] J. Bouwmeester, J. Guo, Survey of Worldwide Pico- and Nanosatellite Missions, Distributions and Subsystem Technology, Acta Astronautica 67 (2010) 854–862.

[2] R. Sandau, K. Brieß, M. D'Errico., Small Satellites for Global Coverage Potential and Limits, ISPRS Journal of Photogrammetry and Remote Sensing 65 (2010) 492–504.

[3] R. Sandau, Status and Trends of Small Satellite Missions for Earth Observation, Acta Astronautica 66 (2010) 1-12.

[4] D. Selva, D. Krejci, A Survey and Assessment of the Capabilities of Cubesats for Earth Observation, Acta Astronautica 74 (2012) 50–68.

[5] Statistical Reliability Analysis of Satellites by Mass Category: Does spacecraft Size Matter? G.F. Dubos, J-F. Castet, J.H. Saleh, Acta Astronautica 67 (2010) 584–595.

[6] M. Tafazoli, A Study of On-orbit Spacecraft Failures, Acta Astronautica 64 (2009) 195–205.

[7] D.M. Harland, R.D. Lorenz, Space Systems Failures, Springer-Praxis, Berlin, 2006.

[8] M. Cho, H. Masui, T. Hatamura, K. Date, S. Horii, S. Obata, Overview of the Nano-satellite Environmental Test Standardization: Test Campaign and Standard Draft, SSC12-VII-10, 26th Annual AIAA/USU Conference on Small Satellites, 2012.

[9] D. Selva, B. Dingwall, S. Altunc, A Concept for an Agile Mission Development Facility for CubeSat and Suborbital Missions, 2016 IEEE Aerospace Conference, Big Sky, Montana, USA, 5–12 March.

[10] QB50 System Requirements and Recommendations, Issue 7, 13 February 2015, https://www.qb50.eu/index.php/tech-docs/category/25-up-to-date-docs, (accessed 31.08.16).

[11] The 2012 Wilson Research Group Functional Verification Study, https://blogs.mentor.com/verificationhorizons/blog/2013/09/08/part-12-the-2012-wilson-research-group-functional-verification-study/, (accessed 08.09.16).

[12] MATLAB API for Other Languages, http://www.mathworks.com/help/matlab/programming-interfaces-for-c-c-fortran-com.html, (accessed 31.08.16).

[13] Simulink Real-Time C API, http://www.mathworks.com/help/xpc/api/what-is-xpc-target-c-api.html, (accessed 31.08.16).

[14] IEEE Standard for SystemVerilog--Unified Hardware Design, Specification, and Verification Language," in IEEE Std 1800-2012 (Revision of IEEE Std 1800-2009) , vol., no., pp.1-1315, Feb. 21 2013.

[15] IEEE Standard VHDL Language Reference Manual Amendment 1: Procedural Language Application Interface," in IEEE Std 1076c-2007 (Amendment to IEEE Std 1076-2002), vol., no., pp.c1-214, Sept. 5 2007.

[16] D. Graham, M. Fewster Experiences of Test Automation: Case Studies of Software Test Automation, Addison-Wesley Professional, 2012.

[17] IEEE Standard 488.2-1992, IEEE Standard Codes, Formats, Protocols and Command Commands for Use with ANSI/IEEE Standard 488.1-1987. New York, NY, 1998.

[18] Interchangeable Virtual Instruments Foundation (IVI), http://www.ivifoundation.org/, (accessed 30.08.16).

[19] Standard Commands for Programmable Instruments http://www.ivifoundation.org/docs/scpi-99.pdf, (accessed 30.08.16).

[20] J. Miall, RF power measurements, in: R.J. Collier, A.D. Skinner (Eds.), Microwave Measurements, third ed., IET Electrical Measurement Series, Volume 12, London, 2007, pp. 329–348.