NASA/CR—2013-216514

GRC GSFC TDRSS Waveform Metrics Report

Dale J. Mortensen
Vantage Partners, LLC, Brook Park, Ohio

June 2013

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at *http://www.sti.nasa.gov*

- E-mail your question to *help@sti.nasa.gov*

- Fax your question to the NASA STI Information Desk at 443–757–5803

- Phone the NASA STI Information Desk at 443–757–5802

- Write to:
  STI Information Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076–1320

NASA/CR—2013-216514

GRC GSFC TDRSS Waveform Metrics Report

Dale J. Mortensen
Vantage Partners, LLC, Brook Park, Ohio

Prepared under Contract NNC12BA01B

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

June 2013

Level of Review: This material has been technically reviewed by NASA technical management OR expert reviewer(s).

Available from

NASA Center for Aerospace Information
7115 Standard Drive
Hanover, MD 21076–1320

National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Available electronically at http://www.sti.nasa.gov

Preface

National Aeronautics and Space Administration (NASA) is developing an on-orbit, adaptable, Software Defined Radio (SDR)/Space Telecommunications Radio System (STRS)-based testbed facility to conduct a suite of experiments to advance technologies, reduce risk, and enable future mission capabilities on the International Space Station (ISS). The Space Communications and Navigation (SCaN) Testbed Project will provide NASA, industry, other Government agencies, and academic partners the opportunity to develop and field communications, navigation, and networking technologies in the laboratory and space environment based on reconfigurable, software defined radio platforms and the STRS Architecture. The project was previously known as the Communications, Navigation, and Networking reConfigurable Testbed (CoNNeCT), Also included are the required support efforts for Mission Integration and Operations, consisting of a ground system and the NASA Glenn Research Center (GRC) Telescience Support Center (TSC). This document has been prepared in accordance with GRCs Configuration Management Procedural Requirements GLPR 8040.1 and applies to the CoNNeCT configuration management activities performed at GRC. This document is consistent with the requirements of SSP 41170, Configuration Management Requirements, International Space Station, and GLPR 7120.5.30 Space Assurance Requirements (SAR).

This document presents the metrics collected during the GRC NASA Goddard Space Flight Center (GSFC) Tracking and Data Relay Satellite System (TDRSS) Waveform development, especially with a STRS perspective.

Contents

List of Tables

List of Figures

GRC GSFC TDRSS Waveform Metrics Report

Dale J. Mortensen
Vantage Partners, LLC
Brook Park, Ohio 44142

1.0 Introduction

1.1 Purpose

This report documents the metrics kept during the development of the GRC GSFC TDRSS (GGT) Waveform. The metrics are focused on porting and the STRS architecture relevance.

1.2 Scope

The report presents software metrics and porting metrics for the GGT Waveform. The porting was from a ground-based COTS SDR, the SDR-3000 to the CoNNeCT the NASA Jet Propulsion Laboratory (JPL) SDR. The report does not address any of the Operating Environment (OE) software development, nor the original TDRSS waveform development at the GSFC for the COTS SDR. With regard to STRS, the report presents compliance data and lessons learned.

2.0 Applicable Documents

This section lists the NASA/Government and non-NASA/Government specifications, standards, guidelines, handbooks, or other special publications applicable to this document.

2.1 Reference Documents

Reference documents are those documents that, though not a part of this document, serve to clarify the intent and contents of this document.

Document number	Reference document
GRC-CONN-PLAN-0076	GRC GSFC TDRSS Waveform Development Plan
GRC-CONN-BDD-0079	GRC GSFC TDRSS Waveform Design Description
GRC-CONN-REQ-0077	GRC GSFC TDRSS Waveform Requirements
GRC-CONN-VDD-0527	GRC GSFC TDRSS Waveform Software Version Description
GRC-CONN-OPI-0530	GRC GSFC TDRSS Waveform User's Guide
GRC-CONN-DBK-0924	GRC GSFC TDRSS Waveform Performance Data Book
TM-2010-216809	STRS Architecture Release 1.02.1
GLPR 7150.1	GRC Software Engineering Requirements
NPR 7150.2	NASA Software Engineering Requirements

3.0 Waveform Development and Porting Overview

The GGT Waveform was developed for the JPL SDR on the SCaN Testbed. This software development was not straight-forward nor traditional, for it was the first port of a STRS waveform. As Figure 3.1 shows the development was a collaboration with GSFC and JPL. The focus of this report is on the porting effort performed by GRC, more specifically taking the COTS SDR version of the software and moving it to the space-based JPL SDR. In this manner the resources already expended at GSFC would be reused for the flight experiment, and the benefits of STRS might be demonstrated. The process is highlighted in Figure 3.1.

Porting the waveform from a COTS platform to a space-based target presented several challenges, as Figure 3.2 illustrates. First, fewer processing resources are available on the space-based SDR. Field Programmable Gate Array (FPGA) resources are half in number and size (i.e., component count and gate count). Therefore, some functionality and features had to be removed from the design. This is described more in Section 5.0.

Secondly, the I/Q mixing and conversion to IF is different, with the COTS SDR design the I/Q mixing is done in the digital domain and then the DAC output is at IF. However, the JPL SDR has an analog I/Q modulator that mixes baseband signals from two DACs. The DAC and ADC hardware capabilities are also reduced in terms of sampling rate and bit resolution when compared to the COTS SDR.

Other interface differences affected the porting effort. Foremost there is the SpaceWire data interface that was not present on the COTS platform. So the original waveform used a simple TTL clock and data interface which had to be converted to the SpaceWire standard.

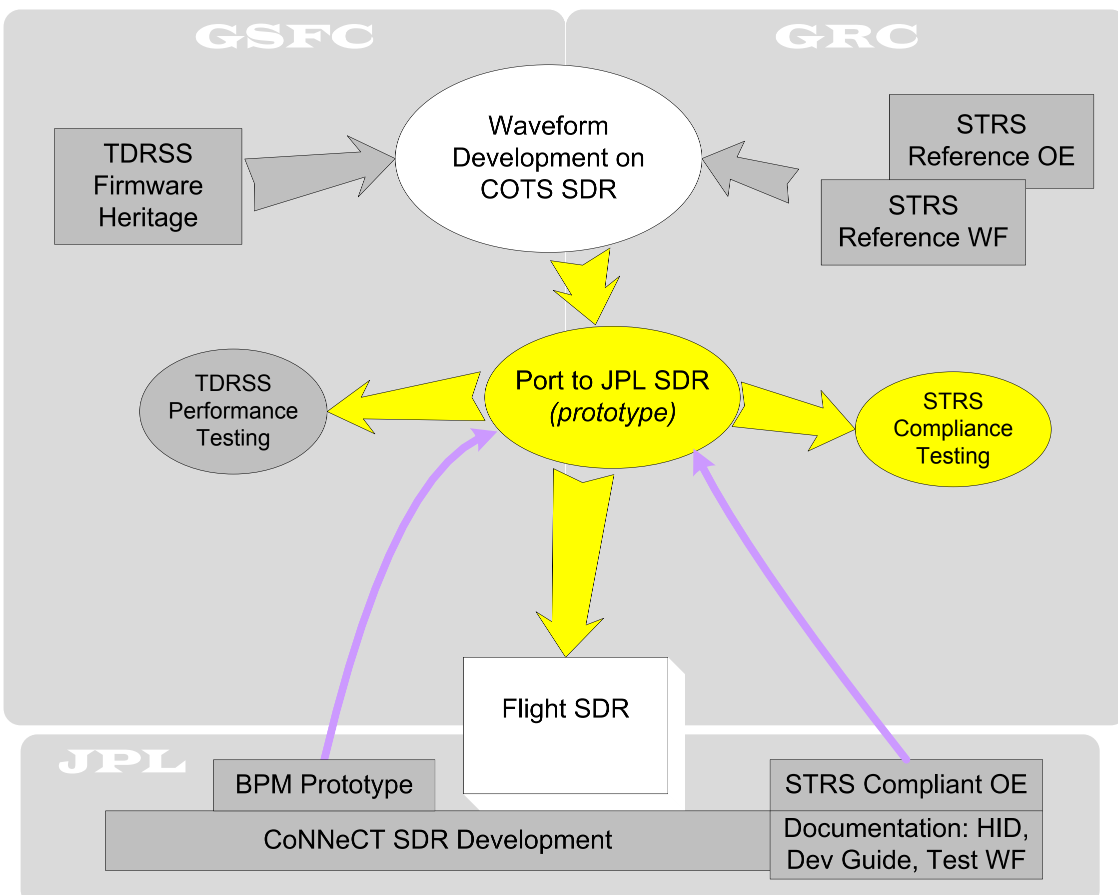

Figure 3.1.—Development and Porting Flow

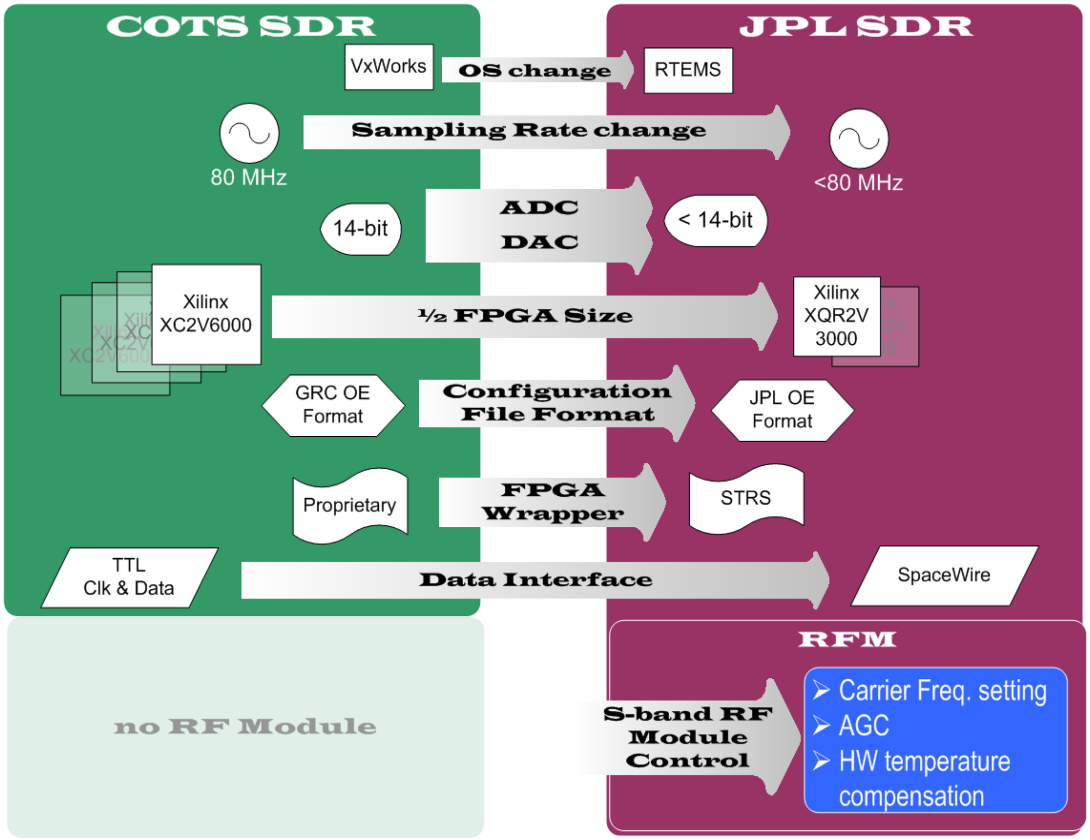

Figure 3.2.—Platform Differences

STRS helped with the software but the FPGA abstraction is different between platforms. The COTS SDR has a proprietary abstraction of the hardware, whereas the JPL SDR has a module architecture within the FPGAs. More details on how STRS helped with porting are discussed in Section 8.0.

Finally, there were hardware environmental effects for which to compensate. This is commonly more significant with a space SDR because of the wider temperature ranges. Also significant is the COTS SDR's absence of an RF front end. Thus, up and down conversion, frequency control, and gain control were additional functions for the ported waveform on the JPL SDR.

4.0 Level of Effort

During the waveform porting the level-of-effort was tracked in terms of days for various tasks. There were three primary engineers who worked the tasks, totaling 374 days effort over the course of 2 years. Thirty-six tasks were tracked during that time, which have been subsequently grouped into ten categories as shown in Figure 4.1.

Because this GGT Waveform is the baseline for the CoNNeCT project launch the porting effort can blur with system integration and testing. However, the tracked days reported here were segregated from system testing as much as possible. Therefore the data reported in Figure 4.1 does **not** include CoNNeCT System integration, performance, and environmental testing (i.e., vibration, thermal vacuum, EMI). It should be noted, however, there were waveform bugs and shortcomings found during system testing that necessitated software revisions. These corrections and enhancements performed were not tracked for this report, and may have distorted the metrics of Figure 4.1 slightly. For example, the "Testing" portion of the effort is relatively small (5 percent) but may have been larger if the waveform-specific test time could be extracted from the system integration testing.

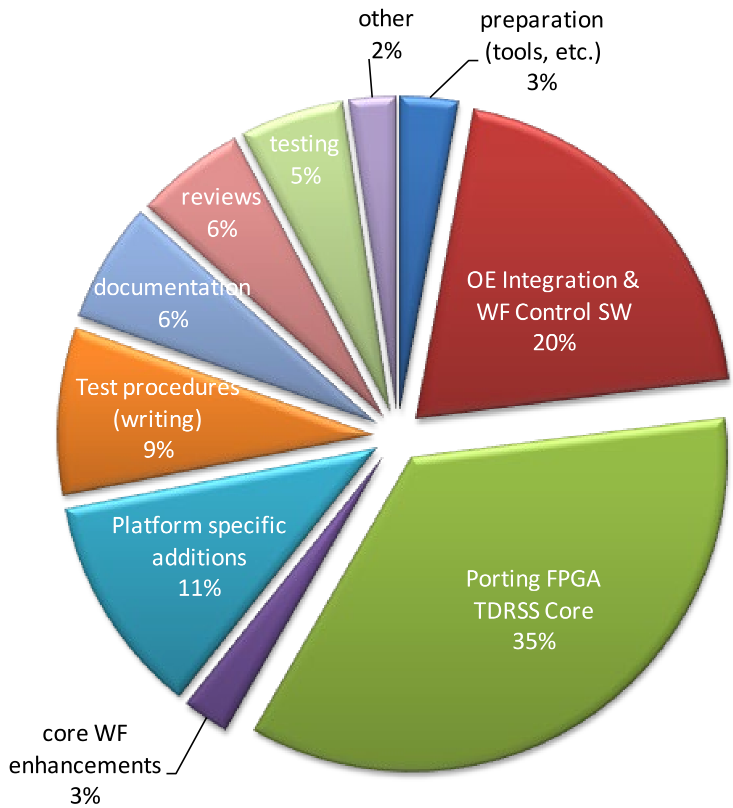

Figure 4.1.—Level of Effort Breakdown

FPGA Portion: Most of the Waveform processing takes place in the FPGA component, and thus the "Porting FPGA TDRSS Core" category was the largest portion of the effort at 35 percent. Grouped in this category were the following tasks:

- Convert model (Simulink) to match JPL SDR specs (sample rate changes, etc.)
- Convert/rewrite FPGA constraint files
- Modify, convert, and trim GSFC HDL code (remove Q channel, other extra stuff)
- Add improved BitSync from GSFC
- Add pulse shape filtering (GPP control needed based on data rate). Now also for Deep Space Network non-interference, and with PA 1 dB compression compensation.
- Interfacing to JPL Wrapper (ADC, DAC, SpaceWire, GPP, RFM)
- Timing debugging
- Simulation Test benches (ModelSim)
- JPL ADC clocking

GPP Portion: The second most significant effort was in the OE Integration and WF Control SW area. This grouping is comprised of:

- Modify, and trim GSFC C++ code on COTS SDR in preparation for port (bug fixes, STRS compliance, flight code upgrade)
- Modify/fix GSFC "Monitor Task" thread. Required changes in FPGA and GPP code.
- Port GPP code to JPL BPM
- Update configuration files for JPL SDR OE

- Write FPGA Device Drivers
- Telemetry Design/Development for JPL OE

Platform Specific: A flight SDR to flight SDR port would not typically contain the flight platform specific functions added during the port. These accounted for 11 percent of the effort. In part the extra efforts were due to the COTS platform not having an RF front end, and the inaugural use of the JPL SDR and JPL OE.

Non-Reuse Efforts: Almost half of the porting effort was **not** directly related to reusing the COTS SDR waveform, and would have been done in a new development regardless. These included:

- Test procedure development
- Documentation
- Reviews
- Testing
- Adding platform-specific functions

5.0 Field Programmable Gate Array (FPGA) Utilization

The COTS SDR platform for which the waveform was first developed had more than adequate FPGA resources. However, the space-based target SDR is relatively limited, especially when trying to keep the second of its two FPGAs free so that a GPS waveform could run in parallel on the second FPGA. Originally the GSFC waveform was designed to meet most of the functionality described in the Space Network User's Guide. As such it consisted of two BPSK signal chains combined in quadrature to form QPSK modulation. It was determined that the QPSK mode was not required for the baseline waveform, and subsequently much of the quadrature (Q) side of the waveform could be trimmed, leaving the required BPSK functionality.

Table 5.1 compares FPGA utilization before and after trimming. In its original form the waveform code could not be implemented in a single FPGA of the target platform.

TABLE 5.1.—FPGA RESOURCES PRE-PORT COMPARISON

FPGA resource	Initial utilization, %	Ported utilization, %
Total slice registers	94.5	59.8
Occupied slices	176.7	99.9
Total 4 input LUTs (look up tables)	98.2	72.4
MULT (multipliers) 18x18s	109.4	85.4

6.0 Processor Code Source Lines of Code (SLOC)

Porting the processor code of the GGT waveform was relatively straight-forward because the high-level 'C/C++' language source was simply compiled for a different processor. Performance differences between the two processors are not critical since the GPP part of the GGT Waveform provides control and configuration, not signal processing. However, the two SDR platforms also have different operating systems, VxWorks for the COTS SDR versus RTEMS for the space SDR, so a different compiler build environment had to be established. Setting up and learning the new build environment increased development time considerably. But even more extensive was the additional code development needed for the space SDR specific hardware.

Being the first operational waveform for the JPL SDR required the development and integration of several platform-specific functions, including:

- Temperature compensation: modulator, power amplifier, and frequency
- Tx Automatic Level Control (ALC)
- Run tests
- Ground tests for calibration
- Verbose logging to serial (because of limited 1553 telemetry)

Figure 6.1 shows the progression of the GPP source code through development, starting with the original ground-based waveform on the COTS platform. The initial port to the JPL Prototype did not include any platform-specific additions. The Prototype does not include any RF circuits/modules, so it is most like the COTS SDR-3000 platform. The additional software for the flight version can mostly be considered “Radio Services” from the STRS perspective. This is code that is waveform independent, but platform specific. Much of it should have been incorporated with the OE if schedule had permitted. It will still be available in the STRS Application Repository for future waveform developers to reuse.

Regarding processor memory resources, the additional platform-specific functions code did not push any limits for the JPL SDR. There is 128 MB of SDRAM on-board, and Figure 6.2 shows the progression of memory resources through the development. It is important to note the memory utilization shown includes the OE, which includes the RTEMS OS and associated file system. The builds with the GPS Capture and s-band capture/playback test waveforms included are an additional 70 kbytes (not shown in figure).

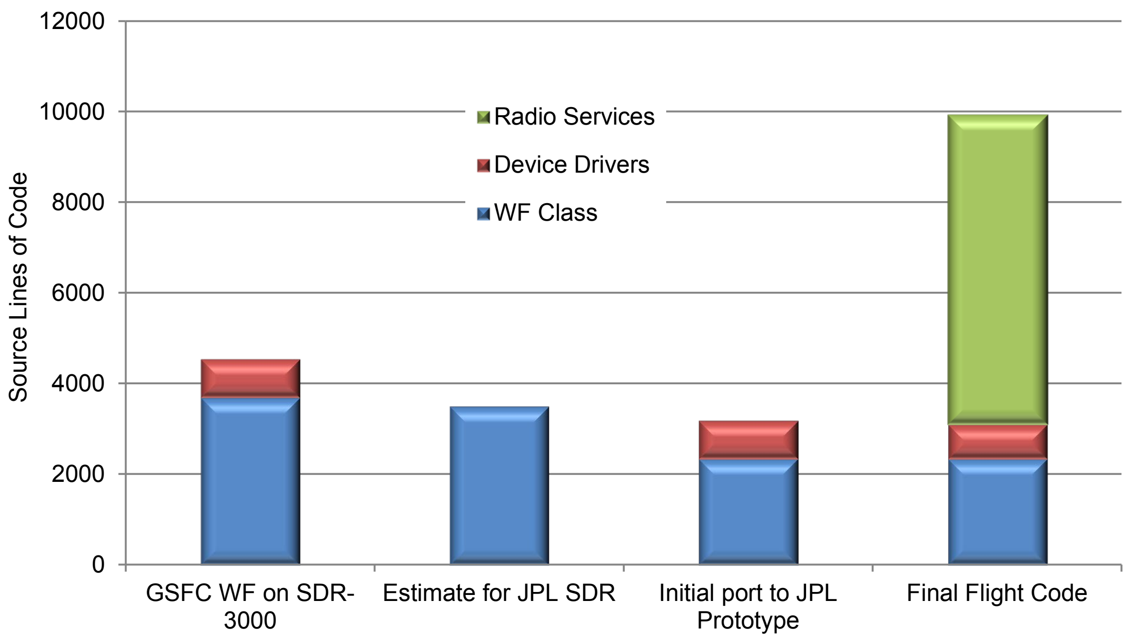

Figure 6.1.—SLOC Porting Progression

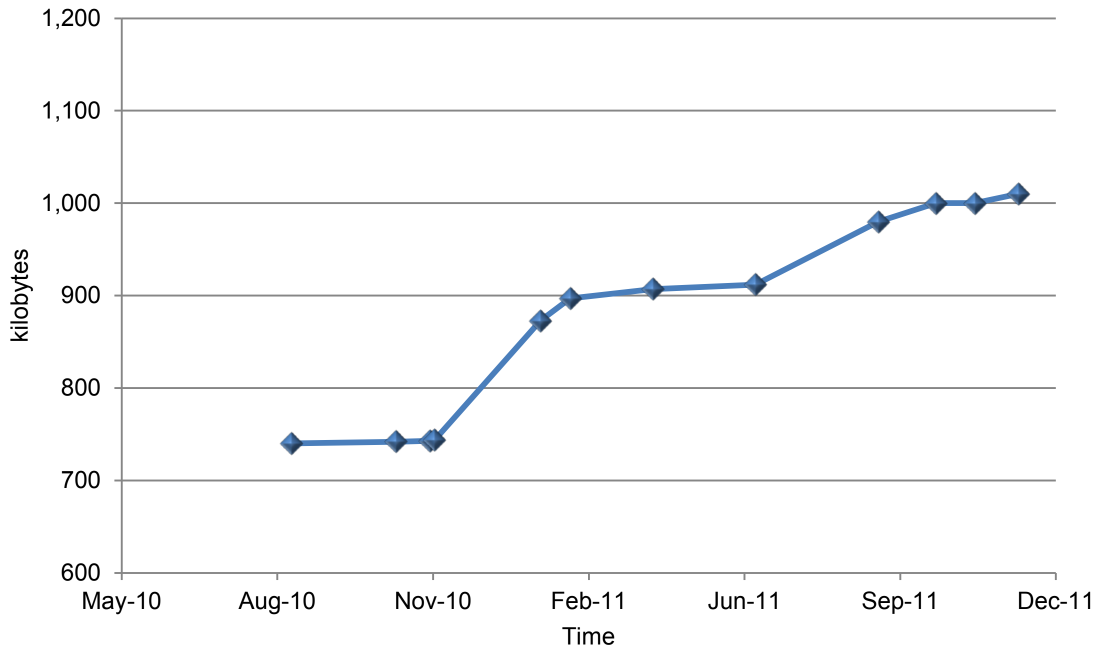

Figure 6.2.—Binary Executable Size (including OE)

```
CPU Usage by thread
    ID         NAME        SECONDS   PERCENT
0x09010001    IDLE       6018.650534  97.029
0x0b010002              0.161051  0.002
0x0b010003              2.772315  0.044
0x0b010004              1.824151  0.029
0x0b010005              0.000092  0.000
0x0b010006              35.151370  0.566
0x0b010007              32.431500  0.522
0x0b010008              0.003171  0.000
0x0b010009              0.001350  0.000
0x0b01000a              0.007956  0.000
0x0b01000b              0.063833  0.001
0x0b01000c              0.000241  0.000
0x0b01000d              10.003583  0.161
0x0b01000e              1.578389  0.025
0x0b01000f              0.007322  0.000
0x0b010010              0.000741  0.000
0x0b010013             100.174635  1.614
Time since last CPU Usage reset 6202.893745 seconds
```

Figure 6.3.—CPU Utilization Snapshot

Processor utilization for the waveform application is small, as expected since it only handles the low rate control and telemetry at 1 Hz. Using the “cpu” command available with the OE, it was observed that the GGT Waveform consumes far less than 10 percent of the processor. On average the GGT Waveform uses less than 3 percent of the processor, as shown in the serial telemetry listing of Figure 6.3. This is a listing of the processor tasks, their time utilization and a corresponding percentage of the total which is produced with the “cpu” command. The Figure 6.3 sample was over 6000 seconds of operation. All of the threads listed are for the OE and Waveform, except the first one listed, “IDLE”.

7.0 STRS and 7150 Compliance

The NASA 7150.2A requirements and the STRS Architecture requirements direct the software development process and the software's implementation, respectively. All waveform software for the SCaN Testbed is to meet a modified Class E standard referred to as Class E+. NPR 7150.2 compliance was achieved with some exceptions for the GGT Waveform. The CoNNeCT project did not have resources to complete level 4 requirements verification within the tight schedule, which this development fell under. Early in the project there was also debate as to which software class would be required. The complete Class E+ 7150.2A compliance matrix is shown in Appendix D. Document mapping from GLPR 7150.1 requirements for the development is in Table 7.1, which also shows the correlation to STRS documentation requirements in the far right column.

Compliance of the waveform application with the STRS Architecture Standard is measured in several ways. Manual inspection and testing are also employed, especially for FPGA code. There is a GRC developed compliance test tool to look at the GPP software source code. The compliance test tool found no STRS violations in the GPP code. The complete GGT Waveform matrix of the STRS requirements is found in Appendix E, which also indicates which requirements were verified by the GRC compliance test tool. The compliance tool output is listed in Appendix F.

There were a few "blue flags" or cautions from the test tool. The test tool is not sophisticated enough to distinguish definition from use and tries to indicate that these are non-standard definitions when they should be correct usage. These flags occur so that a manual step will verify that these are correct usage. For the GGT Waveform those blue flags related to Ground Tests indicate that another method is called, which reduces the number of commands needed during a time-critical TVAC test. The non-standard APP_Stop exists to implement a safety feature that stops the Waveform application if the transmit power exceeds a set threshold to protect the hardware.

TABLE 7.1.—DOCUMENTATION COMPLIANCE

Document/category name	GLPR 7150.1 Software (SW) documentation requirement	CoNNeCT document no.	Completion Date	STRS document req.
Development Plan	SW Management Plan (SMP), SW Configuration Management Plan, SW Test Plan	GRC-CONN-PLAN-0076	Aug 2010	12.8, 12.9
Requirements Specification	SW Requirements Specification (SRS)	GRC-CONN-SPEC-0077	Aug 2010	
Design Description	SW Design Description, SW Interface Design Description	GRC-CONN-BDD-0079	Feb 2012	12.2, 99
Test Procedures	SW Test Procedures	(Comm Performance)	-	
Change Request/ Problem Report	SW Change Request/ Problem Report	Use forms specified in GRC-CONN-PLAN-0002	as needed	
Version Description	SW Version Description	GRC-CONN-VDD-0527	Jan 2012	12.6
Metrics Report (including porting metrics)	SW Metrics Report	GRC-CONN-RPT-0528	May 2012	
Test Report (including TDRSS acceptance testing)	SW Test Report	GRC-CONN-RPT-0529	2011	
User's Guide	SW User's Manual	GRC-CONN-OPI-0530	Apr 2012	12.3
Inputs to CoNNeCT Project Acceptance Data Package	Not applicable		Pre-ship review	

8.0 STRS Effects

Since the GGT Waveform is the first waveform application ported with the STRS architecture, it is important to consider porting benefits from STRS. Benefits discussed here are not absolutes because there was no control subject to which a true comparison could be made, i.e., a non-STRS waveform port on the same hardware. As mentioned in Section 3.0 the porting was atypical because the SDRs were from two different application domains, and most specifically because of the COTS SDR not having an RF front end.

The most obvious benefit to the waveform port with the STRS architecture was due to the standard APIs. There were 25 defined standard methods in the “C/C++” code, comprising about 2000 SLOC. Very little of the base code needed to be changed. So the standard APIs allowed for a reduction of rework with the processor software, meaning the OE integration and WF Control portion of the development would have been significantly larger, as highlighted in Figure 8.1. This is also a metric of portability of the GPP code from the JPL SDR to another STRS-compliant platform.

A secondary benefit from STRS came via the OE commanding. Commanding and configuring from OE was the same for both the COTS SDR and the JPL SDR, again because of standard APIs. It should be noted that this benefit was diminished once the JPL SDR was integrated into the SCaN Testbed system, as the Avionics added a small layer of abstraction. This abstraction was primarily semantics and not a new command set.

Parallel OE and waveform development, although not ideal, was enabled by the STRS Architecture. Since JPL was writing the OE code to the Standard, initial porting work on basic functionality (independent of hardware specifics) could begin on the COTS platform using its OE. Parallel development beyond this phase was difficult, however, and is the reason for the partial exception to the STRS-10 requirement, (see Appendix E).

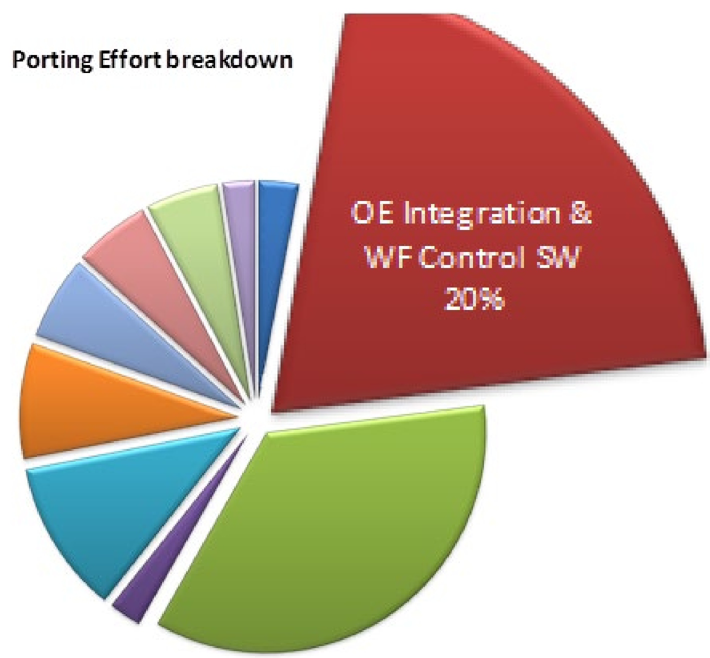

Figure 8.1.—Development Area Most Affected by STRS (compare to Figure 4.1)

9.0 Lessons Learned

Several lessons were learned with the GGT Waveform development that should be considered for future SCaN Testbed experiments as well as the STRS Architecture evolution.

1. STRS Architecture ***was*** helpful for this development (see Section 8.0):
    - The standard APIs reduced porting effort (despite the COTS to space-based platform disparity).
    - Allowed for some parallel development, (forced by schedule constraints) but not ideal.
2. An STRS-compliant SDR platform should compensate for all waveform-independent temperature effects within the OE or dedicated HW. Waveform-dependent compensations may have a standard interface definition (e.g., API) that becomes part of future STRS Standard release.
3. The system integration role should not be assumed to be the platform provider. Clear expectations and requirements at the radio system level would be helpful in project management and planning.
4. STRS addresses an application integrator but does not address system integration directly. The GGT Waveform could have been integrated with the SDR platform correctly as an application, but not perform well in the target LEO ISS environment if system parameters, such as temperature and RF signal level, were not considered. Thoughts in this direction could help to further mature the architecture.
5. Although the STRS architecture standard made possible the parallel SDR platform and GGT Waveform development, there can still be hidden risks with such development. Schedules must be closely synchronized and dependencies maintained. A GGT Waveform example occurred with the SpaceWire interface. The OE delivery of a wrapper for this function occurred after waveform integration. These scenarios may create issues that compromise the end product in terms of optimal performance on a given platform and STRS compatibility. In the case of CoNNeCT this may be especially detrimental to future applications on the SDR which will rely on the GGT Waveform as the baseline waveform.
6. Minimizing the cost of an SDR delivery (CoNNeCT subsystem) by reducing characterization testing can result in more difficult and suboptimal tests at the system level.
7. Be wary of success oriented schedules that do not allow for debug and retesting of unforeseen issues.
8. Porting from more capable platform can be difficult:
    - Reduction in features/performance.
    - Waveform design may need to change (e.g., analog I/Q mod instead of digital)
9. More useful metrics could be found in a comparison of COTS to JPL Prototype only port, or a port of the GGT waveform on the JPL Flight SDR to another STRS flight SDR (e.g., GD or Harris). The port from a COTS SDR to a space SDR is unbalanced and seems less common for NASA, especially as the planned STRS Application Repository grows and is utilized by more NASA missions.
10. Schedule constraints and subsequent waveform requirement verification waiver can lead to undetected SW bugs. For example, the soft-decision Viterbi decoder was not working as designed due to a register being skipped in configuration. Had planned verification testing been completed this bug may have been resolved before system integration testing.

Appendix A.—Acronyms and Abbreviations

A.1 Scope

This appendix lists the acronyms and abbreviations used in this document.

ADC	analog-to-digital converter
ALC	auto level control
BER	bit-error-rate
BPM	Baseband Processing Module
BPSK	binary phase shift keying
CM	Configuration Management
CoNNeCT	Communications, Navigations, and Networking reConfigurable Testbed
COTS	commercial off the shelf
CSCI	computer software configuration item
DAC	digital-to-analog converter
DC	direct current
DDS	direct digital synthesis
Eb/No	energy per bit to noise density ratio
EMI	electromagnetic interference
FEC	forward error correction
FIFO	first in first out
FPGA	field programmable gate array
GD	General Dynamics
GGT	GRC GSFC TDRSS
GIU	Ground Integration Unit
GLPR	Glenn Procedural Requirement
GPP	General Purpose Processor
GPS	Global Positioning System
GRC	NASA Glenn Research Center
GSE	Ground Support Equipment
GSFC	NASA Goddard Space Flight Center
ICD	Interface Control Document
ISS	International Space Station
JPL	NASA Jet Propulsion Laboratory
n/a, N/A	not available, not apply
NASA	National Aeronautics and Space Administration
NEN	near Earth network
NPG	NASA Procedures and Guidelines
NPR	NASA Procedural Requirement
OE	Operating Environment
PA	power amplifier
PN	Pseudo-Noise
PRBS	pseudorandom bit sequence
RF	radio frequency
RFM	Radio Frequency Module
RX, Rx	receive
SCaN, SCAN	Space Communications and Navigation
SDRAM	synchronous dynamic random access memory
SLOC	source lines of code
STRS	Space Telecommunication Radio System

SW	software
TBD	to be determined
TBR	to be resolved
TDRSS	Tracking and Data Relay Satellite System
TRL	Technology Readiness Level
TSIM	TRDSS Simulator
TX (Tx)	transmit
WF	Waveform

Appendix B.—Definitions

B.1 Scope

This appendix lists the definitions used in this document.

Architecture and Design: A description of the mission elements, their interfaces, their logical and physical layout, and the analysis of the design to determine expected performance and margins. Includes System Design Synthesis, System Design Analysis, and System Design Validation products.

Baseline: An agreed-to set of requirements, designs, or documents that will have changes controlled through a formal approval and monitoring process.

Configuration Management: A systematic process for establishing and maintaining control and evaluation of all changes to baseline documentation, products (Configuration Items), and subsequent changes to that documentation which defines the original scope of effort. The systematic control, identification, status accounting, and verification of all Configuration Items throughout their life cycle.

Flight Systems and Ground Support: FS&GS is one of four interrelated NASA product lines. FS&GS projects result in the most complex and visible of NASA investments. To manage these systems, the Formulation and Implementation phases for FS&GS projects follow the NASA project life-cycle model consisting of phases A (Concept Development) through F (Closeout). Primary drivers for FS&GS projects are safety and mission success.

Interface Control Document (ICD): A specification of the mechanical, thermal, electrical, power, command, data, and other interfaces that system elements must meet.

Requirement: The agreed upon need, desire, want, capability, capacity, or demand for personnel, equipment, facilities, or other resources or services by specified quantities for specific periods of time or at a specified time expressed as a “shall” statement. Acceptable form for a requirement statement is individually clear, correct, feasible to obtain, unambiguous in meaning, and can be validated at the level of the system structure at which stated.

Software: As defined in NPD 2820.1, NASA Software Policy.

Specification: A document that prescribes, in a complete, precise, verifiable manner, the requirements, design, behavior, or characteristics of a system or system component.

Success Criteria: Specific accomplishments that must be satisfactorily demonstrated to meet the objectives of a technical review so that a technical effort can progress further in the life cycle. Success criteria are documented in the corresponding technical review plan.

System: (a) The combination of elements that function together to produce the capability to meet a need. The elements include all hardware, software, equipment, facilities, personnel, processes, and procedures needed for this purpose. (Refer to NPR 7120.5.) (b) The end product (which performs operational functions) and enabling products (which provide life-cycle support services to the operational end products) that make up a system. (Refer to WBS definition.)

Technical Performance Measures: The set of critical or key performance parameters that are monitored by comparing the current actual achievement of the parameters with that anticipated at the current time and on future dates. Used to confirm progress and identify deficiencies that might jeopardize meeting a system requirement. Assessed parameter values that fall outside an expected range around the anticipated values indicate a need for evaluation and corrective action. Technical performance measures are typically selected from the defined set of Measures of Performance (MOPs).

Technical Risk: Risk associated with the achievement of a technical goal, criterion, or objective. It applies to undesired consequences related to technical performance, human safety, mission assets, or environment.

Validation (of a product): Proof that the product accomplishes the intended purpose. Validation may be determined by a combination of test, analysis, and demonstration.

Validated Requirements: A set of requirements that are well-formed (clear and un-ambiguous), complete (agrees with customer and stakeholder needs and expectations), consistent (conflict free), and individually verifiable and traceable to a higher-level requirement or goal.

Verification (of a product): Proof of compliance with specifications. Verification may be determined by test, analysis, demonstration, and inspection.

Waiver: A documented agreement intentionally releasing a program or project from meeting a requirement. (Some Centers use deviations prior to Implementation and waivers during Implementation).

Appendix C.—TBDs and TBRs

C.1 Scope

This appendix lists all items in this document that need to be determined (TBD) and that need to be resolved (TBR).

C.2 List of TBDs

TABLE C.1.—TBDS

TBD Number	Description	Section Number	Closure Date/Event

C.3 List of TBRs

TABLE C.2.—TBRS

TBR Number	Description	Section Number	Closure Date/Event

Appendix D.—7150 Compliance Matrix

D.1 Scope

This appendix covers the NPR 7150.2A requirements matrix for the GGT Waveform.

D.2 Class E+ Compliance Matrix

The fifth column in the matrix shows the GGT Waveform's assessment or compliance reference. Some cells are shaded with an orange color to indicate less certain content. Shading in all other columns is a part of the Class E+ matrix as given.

TABLE D.1.—NPR 7150 CLASS E+ COMPLIANCE MATRIX

NPR 7150.2A Requirement Number	Requirement Wording	CoNNeCT Phase II Experiments Expectations (non-avionics)	Comments	GGT WF
SWE-001	This NPR shall be applied to software development, maintenance, retirement, operations, management, acquisition, and assurance activities started after its initial date of issuance [SWE-001].	X - Meet requirement as written		WF Plan (GRC-CONN-PLAN-0076) states this approach.
SWE-013	The project shall develop software plan(s). [SWE-013]	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum: SMP, STP. SMP may be included in the Experiment Plan.	GRC-CONN-PLAN-0076
SWE-014	The project shall implement, maintain, and execute the software plan(s). [SWE-014]	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum: Implement and maintain the schedule.	Reviews held, documents baselined, compat testing… Records of these activities can be found on the CoNNeCT eRoom.
SWE-015	The project shall establish, document, and maintain at least one software cost estimate and associated cost parameter(s) that satisfies the following conditions: [SWE-015] a. Covers the entire software life cycle. b. Is based on selected project attributes (e.g., assessment of the size, functionality, complexity, criticality, and risk of the software processes and products). c. Is based on the cost implications of the technology to be used and the required maturation of that technology.	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum: One cost estimate, cover the entire life cycle, based on size, complexity	GRC-CONN-PLAN-0076 has estimates;
SWE-016	The project shall document and maintain a software schedule that satisfies the following conditions: [SWE-016] a. Coordinates with the overall project schedule. b. Documents the interactions of milestones and deliverables between software, hardware, operations, and the rest of the system. c. Reflects the critical path for the software development activities.	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum: Coordinate with the CoNNeCT schedule	Worked directly with Project scheduler to link deliverables and dependencies. Visio schedules (on eRoom).
SWE-018	The project shall regularly hold reviews of software activities, status, and results with the project stakeholders and track issues to resolution. [SWE-018]	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum: a requirements review to show understanding of CoNNeCT and the SDR along with a test readiness review	Design Review Sep 2009; Status Review Apr 2010
SWE-019	The project shall select and document a software development life cycle or model that includes phase transition criteria for	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum set: internal V&V, integration with our GIU, V&V on GIU	GGT WF has been a part of most reviews. V&V informally via Comm Sys

NPR 7150.2A Requirement Number	Requirement Wording	CoNNeCT Phase II Experiments Expectations (non-avionics)	Comments	GGT WF
	each life cycle phase (e.g., formal review milestones, informal reviews, software requirements review (SRR), preliminary design review (PDR), critical design review (CDR), test readiness reviews, customer acceptance or approval reviews). [SWE-019]			testing and TDRSS Compat testing.
SWE-020	The project shall classify each of system and subsystem containing software in accordance with the software classification definitions for Class A, B, C, D, E, F, G and H in Appendix E. [SWE-020]	X - Meet requirement as written		Was originally classified as 'C', but now is 'E+'
SWE-021	If a system or subsystem evolves to a higher software classification as defined in Appendix E, then the project shall update its plan to fulfill the added requirements per the Requirements Mapping Matrix in Appendix D. [SWE-021]	X - Meet requirement as written		N/A
SWE-022	The project shall implement software assurance per NASA-STD-8739.8, NASA Software Assurance Standard. [SWE-022]	P(Center) - Meet per Center defined process which meets non-empty requirement subset		GRC-CONN-PLAN-0076 describes the ways in which software assurance was implemented. Had witnessed builds, SVN checkouts, and uploads to Flight system.
SWE-023	When a project is determined to have safety critical software, the project shall ensure that the safety requirements of NASA-STD-8719.13, Software Safety Standard, are implemented by the project. [SWE-023]	X - Meet requirement as written		No safety critical functions with the GGT WF.
SWE-024	The project shall ensure that actual results and performance of software activities are tracked against the software plans. [SWE-024]		Minimum: Show schedule status	Not sure if this was done formally/intentionally against GRC-CONN-PLAN-0076.
SWE-028	The project shall plan software verification activities, methods, environments, and criteria for the project. [SWE-028]	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum: cover in the STP	Comm System testing, Compat testing
SWE-029	The project shall plan the software validation activities, methods, environments, and criteria for the project. [SWE-029]	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum: cover in the STP	Done down to Level 3 of project; WF is level 4.
SWE-030	The project shall record, address, and track to closure the results of software verification activities. [SWE-030]	X - Meet requirement as written		Done down to Level 3 of project; WF is level 4.
SWE-031	The project shall record, address, and track to closure the results of software validation activities. [SWE-031]	X - Meet requirement as written		Done down to Level 3 of project; WF is level 4.
SWE-033	The project shall assess options for software acquisition versus development. [SWE-033]			N/A for SDRs
SWE-034	The project shall define and document or record the acceptance criteria and conditions for the software. [SWE-034]			N/A or GRC-CONN-REQ-0077
SWE-036	The project shall determine which software processes, activities, and tasks are required for the project. [SWE-036]			GRC-CONN-PLAN-0076
SWE-037	The project shall define the milestones at which the software supplier(s) progress will be reviewed and audited as a part of			Relative to GSFC and JPL, GRC-CONN-PLAN-0076.

NPR 7150.2A Requirement Number	Requirement Wording	CoNNeCT Phase II Experiments Expectations (non-avionics)	Comments	GGT WF
	the acquisition activities. [SWE-037]			
SWE-038	The project shall document software acquisition planning decisions. [SWE-038]			N/A or GRC-CONN-PLAN-0076
SWE-039	The project shall require the software supplier(s) to provide insight into software development and test activities; at a minimum the following activities are required: monitoring integration, review of the verification adequacy, review of trade study data and results, auditing the software development process, participation in software reviews and systems and software technical interchange meetings. [SWE-039]			N/A or GRC-CONN-PLAN-0076
SWE-044	The project shall require the software supplier(s) to provide software metric data as defined in the project's Software Metrics Report. [SWE-044]			N/A or GRC-CONN-PLAN-0076
SWE-046	The project shall require the software supplier(s) to provide a software schedule for the project's review and schedule updates as requested. [SWE-046]			N/A or GRC-CONN-PLAN-0076
SWE-048	The project shall document in the solicitation the software processes, activities, and tasks to be performed by the supplier. [SWE-048]			N/A or GRC-CONN-PLAN-0076
SWE-049	The project shall document the software requirements. [SWE-049]	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum: Document requirements in electronic format readable by NASA.	GRC-CONN-REQ-0077
SWE-050	The project shall identify, develop, document, approve, and maintain software requirements based on analysis of customer and other stakeholder requirements and the operational concepts. [SWE-050]			Several changes were made during baseline process of GRC-CONN-REQ-0077.
SWE-053	The project shall collect and manage changes to the software requirements. [SWE-053]	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum: Provide summary of changes to NASA.	GRC-CONN-REQ-0077 is in the CM tracking system.
SWE-057	The project shall transform the allocated and derived requirements into a documented software architectural design. [SWE-057]			GRC-CONN-BDD-0079
SWE-060	The project shall implement the software design into software code. [SWE-060]			Released software follows Design Description (GRC-CONN-BDD-0079).
SWE-062	The project shall ensure that the software code is unit tested per the plans for software testing. [SWE-062]	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum: Provide results of unit testing to NASA.	Comm System testing, Compat testing
SWE-063	The project shall provide a Software Version Description document for each software release. [SWE-063]	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum: Documented version number of the executable software, data integrity checks for the executable code, open problem reports with workarounds	GRC-CONN-VDD-0527
SWE-065	The project shall establish and maintain: [SWE-065] a. Software Test Plan(s). b. Software Test Procedures. c. Software Test Reports.	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum set: test plan, test reports	Comm Verification & Tech Experiment Test Plan (GRC-CONN-PLAN-0283); many test procedures from this; reports in the form of as-runs and two analysis reports

NPR 7150.2A Requirement Number	Requirement Wording	CoNNeCT Phase II Experiments Expectations (non-avionics)	Comments	GGT WF
				(GRC-CONN-ANA-0855 & 0854).
SWE-066	The project shall perform software testing as defined in the Software Test Plan. [SWE-066]	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum: Provide test results.	Numerous as-runs in CMTS.
SWE-068	The project shall evaluate test results and document the evaluation. [SWE-068]	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum: Provide test results.	GGT Waveform Performance Data Book (GRC-CONN-DBK-0924)
SWE-069	The project shall document defects identified during testing and track to closure. [SWE-069]	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum: list of problem reports with workarounds	3 or 4 CPARs were generated, all resolved; TVL Mantis system has ~50 entries, most fixed or deferred.
SWE-070	The project shall verify, validate, and accredit software models, simulations, and analysis tools required to perform qualification of flight software or flight equipment. [SWE-070]			RF test equipment all certified, as well as custom test SW.
SWE-072	The project shall provide and maintain bidirectional traceability from the Software Test Procedures to the software requirements. [SWE-072]			Considered adding this to the Req cross-reference in Design doc, but have not yet.
SWE-077	The project shall complete and deliver the software product to the customer with appropriate documentation to support the operations and maintenance phase of the software's life cycle. [SWE-077]	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum: provide a user guide or operations manual, licenses for any licensed software, version description	GRC-CONN-OPI-0530
SWE-079	The project shall develop a Software Configuration Management Plan that describes the functions, responsibilities, and authority for the implementation of software configuration management for the project. [SWE-079]	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum: develop a SW CM Plan that ensures code is under CM control; may be just a section of the SW Management Plan	GRC-CONN-PLAN-0001
SWE-080	The project shall track and evaluate changes to software products. [SWE-080]	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum: Provide summary of changes to NASA and notify NASA of any changes that threaten the schedule.	Weekly meetings and status/schedule reports; TVL Subversion system.
SWE-081	The project shall identify the software configuration items (e.g., software documents, code, data, tools, models, scripts) and their versions to be controlled for the project. [SWE-081]	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum: identify the software configuration items	GRC-CONN-PLAN-0001, GRC-CONN-PLAN-0076
SWE-085	The project shall establish and implement procedures for the storage, handling, delivery, release, and maintenance of deliverable software products. [SWE-085]	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum: org. to develop a release process for delivering software products to NASA or NASA may develop one in which case this requirement goes away	GRC-CONN-PLAN-0001, GRC-CONN-PLAN-0076
SWE-102	The Software Development or Management Plan shall contain: [SWE-102] a. Project organizational structure showing authority and responsibility of each organizational unit, including external organizations (e.g., Safety and Mission Assurance, Independent Verification and Validation (IV&V), Technical Authority, NASA Engineering and Safety Center, NASA Safety Center). b. The safety criticality and classification of each of the systems and subsystems containing software.	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum: Include schedule, life cycle, cost estimate (non-University only), software and hardware environment, software and hardware tools, V&V plan (may be separate document), training plans, CM plan	GRC-CONN-PLAN-0076 includes minimum elements.

NPR 7150.2A Requirement Number	Requirement Wording	CoNNeCT Phase II Experiments Expectations (non-avionics)	Comments	GGT WF
	c. Tailoring compliance matrix for approval by the designated Engineering Technical Authority, if the project has any waivers or deviations to this NPR. d. Engineering environment (for development, operation, or maintenance, as applicable), including test environment, library, equipment, facilities, standards, procedures, and tools. e. Work breakdown structure of the life-cycle processes and activities, including the software products, software services, non-deliverable items to be performed, budgets, staffing, acquisition approach, physical resources, software size, and schedules associated with the tasks. f. Management of the quality characteristics of the software products or services. g. Management of safety, security, privacy, and other critical requirements of the software products or services. h. Subcontractor management, including subcontractor selection and involvement between the subcontractor and the acquirer, if any. i. Verification and validation. j. Acquirer involvement. k. User involvement. l. Risk management. m. Security policy. n. Approval required by such means as regulations, required certifications, proprietary, usage, ownership, warranty, and licensing rights. o. Process for scheduling, tracking, and reporting. p. Training of personnel, including project unique software training needs. q. Software life-cycle model, including description of software integration and hardware/software integration processes, software delivery, and maintenance. r. Configuration management. s. Software documentation tree. t. Software peer review/inspection process of software work products. u. Process for early identification of testing requirements that drive software design decisions (e.g., special system level timing requirements/checkpoint restart). v. Software metrics. w. Content of software documentation to be developed on the project. x. Management, development, and testing approach for handling any commercial-off-the-shelf (COTS), government-off-the-shelf (GOTS), modified-off-the-shelf (MOTS), reused, or open source software component(s) that are included within a NASA system or subsystem.			
SWE-103	The Software Configuration Management Plan shall contain: [SWE-103] a. The project organization(s). b. Responsibilities of the software			GRC-CONN-PLAN-0001 contains these.

NPR 7150.2A Requirement Number	Requirement Wording	CoNNeCT Phase II Experiments Expectations (non-avionics)	Comments	GGT WF
	configuration management organization. c. References to the software configuration management policies and directives that apply to the project. d. All functions and tasks required to manage the configuration of the software, including configuration identification, configuration control, status accounting, and configuration audits and reviews. e. Schedule information, which establishes the sequence and coordination for the identified activities and for all events affecting the plan's implementation. f. Resource information, which identifies the software tools, techniques, and equipment necessary for the implementation of the activities. g. Plan maintenance information, which identifies the activities and responsibilities necessary to ensure continued planning during the life cycle of the project. h. Release management and delivery.			
SWE-104	The Software Test Plan shall include: [SWE-104] a. Test levels (separate test effort that has its own documentation and resources, e.g., component, integration, and system testing). b. Test types: (1) Unit testing. (2) Software integration testing. (3) Systems integration testing. (4) End-to-end testing. (5) Acceptance testing. (6) Regression testing. c. Test classes (designated grouping of test cases). d. General test conditions. e. Test progression. f. Data recording, reduction, and analysis. g. Test coverage (breadth and depth) or other methods for ensuring sufficiency of testing. h. Planned tests, including items and their identifiers. i. Test schedules. J. Requirements traceability (or verification matrix). k. Qualification testing environment, site, personnel, and participating organizations.		Minimum: lay out plans for integration testing with the GIU and V&V on the GIU (this should be P(Center) since SWE-028 is)	Comm Verification & Tech Experiment Test Plan (GRC-CONN-PLAN-0283) along with the derived test procedures.
SWE-109	The Software Requirements Specification shall contain: [SWE-109] a. System overview. b. CSCI requirements. (1) Functional requirements. (2) Required states and modes. (3) External interface requirements. (4) Internal interface requirements. (5) Internal data requirements. (6) Adaptation requirements. (7) Safety requirements. (8) Performance and timing requirements. (9) Security and privacy requirements. (10) Environment requirements. (11) Computer resource requirements.		Minimum: functional requirements, states and modes, external interface requirements, performance and timing requirements, computer resource requirements	GRC-CONN-REQ-0077 contains the minimum elements.

NPR 7150.2A Requirement Number	Requirement Wording	CoNNeCT Phase II Experiments Expectations (non-avionics)	Comments	GGT WF
	(a) Computer hardware resource utilization requirements. (b) Computer software requirements. (c) Computer communications requirements. (12) Software quality characteristics. (13) Design and implementation constraints. (14) Personnel-related requirements. (15) Training-related requirements. (16) Logistics-related requirements. (17) Packaging requirements. (18) Precedence and criticality of requirements. c. Qualification provisions (e.g., demonstration, test, analysis, inspection). d. Bidirectional requirements traceability. e. Requirements partitioning for phased delivery. f. Testing requirements that drive software design decisions (e.g., special system level timing requirements/checkpoint restart). g. Supporting requirements rationale.			
SWE-111	The Software Design Description shall include: [SWE-111] a. CSCI-wide design decisions/trade decisions. b. CSCI architectural design. c. CSCI decomposition and interrelationship between components: (1) CSCI components: (a) Description of how the software item satisfies the software requirements, including algorithms, data structures, and functional decomposition. (b) Software item I/O description. (c) Static/architectural relationship of the software units. (d) Concept of execution, including data flow, control flow, and timing. (e) Requirements, design and code traceability. (f) CSCI's planned utilization of computer hardware resources. (2) Rationale for software item design decisions/trade decisions including assumptions, limitations, safety and reliability related items/concerns or constraints in design documentation. (3) Interface design.			GRC-CONN-BDD-0079 contains all but may be lacking in (1b); GSFC's document contains some of these items for the core WF.
SWE-116	The Software Version Description shall identify and provide: [SWE-116] a. Full identification of the system and software (i.e., numbers, titles, abbreviations, version numbers, and release numbers). b. Executable software (i.e., batch files, command files, data files, or other software needed to install the software on its target computer). c. Software life-cycle data that defines the software product. d. Archive and release data. e. Instructions for building the executable software including, for example, the	P(Center) - Meet per Center defined process which meets non-empty requirement subset	Minimum: Documented version number of the executable software, data integrity checks for the executable code, open problem reports with workarounds	GRC-CONN-VDD-0527 does not contain (e), this is in the Design doc. Item (f) is contained indirectly via the listed md5 and crc files.

NPR 7150.2A Requirement Number	Requirement Wording	CoNNeCT Phase II Experiments Expectations (non-avionics)	Comments	GGT WF
	instructions and data for compiling and linking and the procedures used for software recovery, software regeneration, testing, or modification. f. Data integrity checks for the executable object code, and source code. g. Software product files (any files needed to install, build, operate, and maintain the software). h. Open change requests and or problem reports, including any workarounds. i. Change requests and/or problem reports implemented in the current software version since the last Software Version Description was published.			
SWE-120	For those cases in which a Center or project desires a general exclusion from requirement(s) in this NPR or desires to generically apply specific alternate requirements that do not meet or exceed the requirements of this NPR, the requester shall submit a waiver for those exclusions or alternate requirements for approval by the NASA Headquarters' Chief Engineer with appropriate justification. [SWE-120]	X - Meet requirement as written		TBD
SWE-121	Where approved, the requesting Center or project shall document the approved alternate requirement in the procedure controlling the development, acquisition, and/or deployment of the affected software. [SWE-121]	X - Meet requirement as written		TBD
SWE-125	Each project with software components shall maintain a compliance matrix against requirements in this NPR, including those delegated to other parties or accomplished by contract vehicles. [SWE-125]	X - Meet requirement as written		This table is it!
SWE-132	The project's software assurance organization shall perform an independent classification assessment. [SWE-132]	X - Meet requirement as written		TBD
SWE-133	The project, in conjunction with the Safety and Mission Assurance organization, shall determine the software safety criticality in accordance with NASA-STD-8739.8. [SWE-133]	X - Meet requirement as written		Done
SWE-136	The project shall validate and accredit software tool(s) required to develop or maintain software. [SWE-136]			All COTS with the exception of TSIM GSE SW, and this test SW was accredited.
SWE-139	Centers and projects shall fully comply with the "shall" statements in this NPR that are marked with an "X" in Appendix D consistent with their software classification. [SWE-139]	X - Meet requirement as written		per this spreadsheet
SWE-140	When the requirement and software class are marked with a "P (Center)," Centers and projects shall meet the requirement with an approved non-null subset of the "shall" statement (or approved alternate) for that specific requirement. [SWE 140]	X - Meet requirement as written		per this spreadsheet

Appendix E.—STRS Compliance Matrix

E.1 Scope

This appendix covers the STRS Architecture Standard (STRS-ATP-00001) requirements matrix for the GGT Waveform.

E.2 STRS v1.01 Compliance Matrix

Not all STRS requirements are listed in the matrix, only those pertaining to the waveform application. Requirements not listed here apply to the OE and/or platform. The "Tested" column indicates if the GRC Compliance test tool (Script) was used to verify the requirement, or another method was used. A check mark (√) in the last column indicates that the tested methods demonstrated compliance with the requirement.

TABLE E.1.—STRS V1.01 COMPLIANCE MATRIX

Requirements	Description	Tested	Compliance and comments
STRS-10	An STRS application shall use the infrastructure STRS API and POSIX API for access to platform resources.	Script + Inspect	Script √; SpW exception - due to JPL OE schedule.
STRS-12	Application development artifacts shall be submitted to the NASA STRS Repository. The use will be subject to the appropriate license agreements. The application development artifacts shall include, as a minimum, the following:	Inspect	
12.1	☐High level system or component software model		Simulink: GSFC's (not functionally accurate)
12.2	☐Documentation of application firmware external interfaces (e.g., signal names, descriptions, polarity, format, data type, and timing constraints)		GSFC's "STRS Build 16 ICD" & some of Design doc (GRC-CONN-BDD-0079)
12.3	☐Documentation of STRS application behavior		User's Guide (GRC-CONN-OPI-0530)
12.4	☐Application function sources (e.g., C, C++, header files, VHDL, Verilog)		TVL SVN CM
12.5	☐Application libraries, if applicable (e.g., EDIF , DLL)		TVL SVN CM
12.6	☐Documentation of application development environment and tool suite		VDD Table 3-2; Build process per Design doc, make files and projects files in TVL SVN CM.
	o Include application name, purpose, developer, version, and configuration specifics		Missing purpose – will be added to VDD Table 3-2.
	o Include the hardware on which the application is executed, its OS, OS developer, OS version, and OS configuration specifics		VDD
12.7	☐Test plan and results documentation		Comm Performance Tests; (GGT specific plan draft started by S. Mainger, never completed); GRC-CONN-DBK-0924.
12.8	☐Identification of Flight Software Development Standards used		Dev Plan
STRS-13	If the STRS application has a component resident in an SPM (e.g., FGPA firmware), then it shall accept configuration and control commands from the STRS Operating Environment.	Observe	Yes. (see UG configs and Design doc)

Requirements	Description	Tested	Compliance and comments
STRS-16	The STRS *Application-provided Application Control API* shall be implemented using C or C++.	Inspect	Yes. (see source in SVN repository)
STRS-20	Each STRS application shall contain: *#include "STRS_ApplicationControl.h"*	Script	(missing 0 include files)
STRS-22	If the *STRS Application-provided Application Control API* is implemented in C++, the STRS application class shall be derived from the *STRS_ApplicationControl* base class.	Inspect	√; uses the STRS_ApplicationControl class
STRS-23	If the STRS application provides the *APP_Write* method, the STRS application shall contain *#include "STRS_Sink.h"*	Script	method not provided by GGT WF
STRS-25	If the *STRS Application-provided Application Control API* is implemented in C++ AND the STRS application provides the *APP_Write* method, the STRS application class shall be derived from the *STRS_Sink* base class.	Inspect	method not provided by GGT WF
STRS-26	If the STRS application provides the *APP_Read* method, the STRS application shall contain *#include "STRS_Source.h"*	Script	method not provided by GGT WF
STRS-28	If the *STRS Application-provided Application Control API* is implemented in C++ AND the STRS application provides the *APP_Read* method, the STRS application class shall be derived from the *STRS_Source* base class.	Inspect	method not provided by GGT WF
STRS-29	Each STRS application shall contain a callable *APP_Configure* method as described in Table 8-3.	Script	Script √ 9 APP required methods found out of 9
STRS-30	Each STRS application shall contain a callable *APP_GroundTest* method as described in Table 8-4.	Script	Script √; 9 APP required methods found out of 9
STRS-31	Each STRS application shall contain a callable *APP_Initialize* method as described in Table 8-5.	Script	Script √; 9 APP required methods found out of 9
STRS-32	Each STRS application shall contain a callable *APP_Instance* method as described in Table 8-6.	Script	Script √; 9 APP required methods found out of 9
STRS-33	Each STRS application shall contain a callable *APP_Query* method as described in Table 8-7.	Script	Script √; 9 APP required methods found out of 9
STRS-34	If the STRS application provides data to the infrastructure, then the STRS application shall contain a callable *APP_Read* method as described in Table 8-8.	Script	method not provided by GGT WF
STRS-35	Each STRS application shall contain a callable *APP_ReleaseObject* method as described in Table 8-9.	Script	Script √; 9 APP required methods found out of 9
STRS-36	Each STRS application shall contain a callable *APP_RunTest* method as described in Table 8-10.	Script	Script √; 9 APP required methods found out of 9
STRS-37	Each STRS application shall contain a callable *APP_Start* method as described in Table 8-11.	Script	Script √; 9 APP required methods found out of 9
STRS-38	Each STRS application shall contain a callable *APP_Stop* method as described in Table 8-12.	Script	Script √; 9 APP required methods found out of 9
STRS-39	If the STRS application receives data from the infrastructure, then the STRS application shall contain a callable *APP_Write* method as described in Table 8-13.	Script	method not provided by GGT WF

Requirements	Description	Tested	Compliance and comments
STRS-54	When an STRS application has a non-fatal error, the STRS application shall use the *STRS_Log method* (Table 8-27) with a target handle ID of constant STRS_ERROR_QUEUE.	WFCCN & Inspect	√;
STRS-55	When an STRS application has a fatal error, the STRS application shall use the *STRS_Log* method (Table 8-27) with a target handle ID of constant STRS_FATAL_QUEUE.	WFCCN & Inspect	√;
STRS-56	When an STRS application has a warning condition, the STRS application shall use the *STRS_Log* method (Table 8-27) with a target handle ID of constant STRS_WARNING_QUEUE.	WFCCN & Inspect	√;
STRS-57	When an STRS application needs to send telemetry, the STRS application shall use the *STRS_Log* method (Table 8-27) with a target handle ID of constant STRS_TELEMETRY_QUEUE.	WFCCN & Inspect	√;
STRS-77	The STRS applications shall use the *STRS Infrastructure Messaging* methods to send messages between applications and/or the infrastructure with a single target handle ID.	Inspect	Messaging not used.
STRS-82	Any portion of the STRS Applications on the GPP needing time control shall use the *STRS Infrastructure Time Control* methods to access the hardware and software timers.	Inspect	√; The monitoring task thread uses a POSIX call to "sleep" for 1 second. The exact interval is not important.
STRS-91	STRS Applications shall use POSIX methods except for the unsafe functions listed in Table 8-57.	Script	Script √;
STRS-97	An STRS application shall use the *STRS_Log* and *STRS_Write* methods to send STRS telemetry set information to the external system.	Inspect	√;
STRS-99	The STRS application developer shall document the necessary application information to develop a pre-deployed application configuration file in XML.	Inspect document	TVL SVN CM
STRS-101	The pre-deployed STRS application configuration file shall identify, as a minimum, the following application attributes and default values	Inspect	Only way to include all of these with JPL's OE is to put as comments. --> XML file is being updated
	☐Identification		
	☐Unique STRS handle name for the application		
	☐Class name (if applicable)		
	☐State after processing the configuration file		
	☐Required resources		
	☐Memory in bytes		
	☐Number of gates or logic elements		
	☐Configuration parameters containing the STRS handle, names of files, devices, queues, waveforms and services needed by the STRS application		
	☐Values and constraints for all operationally configurable parameters		
	☐Filename(s) of loadable images for resources		

Appendix F.—STRS Compliance Test Tool Output

F.1 Scope

This appendix lists the GRC STRS Compliance Test Tool output for the GGT Waveform, flight release, version 1.1.3. For more information about the tool please see the STRS Compliance Testing document found at: http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20120000912_2012000901.pdf

F.1 Compliance Tool Output Listing

The tool is essentially a script that runs on the source file directory of the waveform application. The following is a listing of the script output. Text in blue color is indicative of a warning not a non-compliance.

```
************************output listing below**********************************
```

STRS Compliance Results Mon May 7 12:04:02 EDT 2012
Directory ../jplggt

- Process directory ../jplggt

Occurrences	Item examined
0	POSIX methods not allowed
0	Deprecated methods
0	QuicComm methods
0	Missing STRS_TEST_STATUS from APP_RunTest
9	APP required methods found out of 9
0	APP required methods missing
6	Non-standard APP method definitions or invocations
0	Extra APP methods
796	STRS methods
4	Distinct STRS methods out of 42
0	Extra STRS methods
0	Missing STRS include files
0	Need to address 0 errors

STRS Application-provided methods:

```
STRS APP_Configure required and found:       2 (Full signature: 2)
STRS APP_GroundTest required and found:       7 (Full signature: 2)
STRS APP_Initialize required and found:       2 (Full signature: 2)
STRS APP_Instance required and found:        2 (Full signature: 2)
STRS APP_Query required and found:         2 (Full signature: 2)
STRS APP_ReleaseObject required and found:      2 (Full signature: 2)
STRS APP_RunTest required and found:        2 (Full signature: 2)
STRS APP_Start required and found:         2 (Full signature: 2)
STRS APP_Stop required and found:          3 (Full signature: 2)
```

5 / 7 non-standard for APP_GroundTest:

1. jplggtWF_Configure.cpp:2376: APP_GroundTest((STRS_TestID) 5, NULL);
2. jplggtWF_Configure.cpp:2411: APP_GroundTest((STRS_TestID) 4, NULL);
3. jplggtWF_Configure.cpp:2439: APP_GroundTest((STRS_TestID) 6, NULL);
4. jplggtWF_Configure.cpp:2467: APP_GroundTest((STRS_TestID) 2, NULL);
5. jplggtWF_Configure.cpp:2495: APP_GroundTest((STRS_TestID) 3, NULL);

1 / 3 non-standard for APP_Stop:

1. jplggt_timedtasks.cpp:395: APP_Stop();

Appendix G.—Mantis Bug Tracking Statistics

G.1 Scope

This appendix lists the Mantis bug tracking statistics for this development.

G.2 Mantis Issues Statistics

The following tables show the number of issues reported by severity, priority, and by category. There were a total of

By severity	Total issues
Feature	9
Trivial	1
Tweak	1
Minor	24
Major	27
Crash	1

By category	Total
DOCS	1
FPGA	16
GPP	46

By priority	Total
None	2
Low	1
Normal	42
High	12
Urgent	4
Immediate	2