SUBJECT:	**Technical Note — SPIRE — Canadian Contribution — Software**		
PREPARED BY:	**Trevor Fulton**		
DOCUMENT No:	**SPIRE-UOL-NOT-001554**		
ISSUE:	**Version 1.0**	**Date:**	**6 March 2003**

APPROVED BY:		**Date:**
CSA	**Victor Zilinskas**	
SPIRE Co-Investigator:	**David Naylor**	
SPIRE, Local Project Manager, Canada:	**Peter Davis-Imhof**	? Signature Not Verified Peter Davis Digitally signed by Peter Davis DN: cn=Peter Davis, o=University of Lethbridge, ou=Department of Physics, c=CA Date: 2003.03.11 10:09:37 -07'00'
SPIRE TFTS Software Engineer:	**Trevor Fulton**	? Signature Not Verified Trevor Fulton Digitally signed by Trevor Fulton DN: cn=Trevor Fulton, o=University of Lethbridge, ou=SPIRE -- TFTS -- Software, c=CA Date: 2003.03.11 10:38:26 -07'00'
RAL AIV Manager	**Dave Smith**	
RAL AIV Data Engineer	**Ken King**	

Distribution

Victor Zilinskas		CSA
Steve Torchinsky		CSA
David Naylor	SPIRE	ULETH
Peter Davis-Imhof	SPIRE	ULETH
Trevor Fulton	SPIRE	ULETH
Kris Dyke	SPIRE	ULETH
Ken King	SPIRE	RAL
Sunil Sidher	SPIRE	RAL
Tanya Lim	SPIRE	RAL
Steve Guest	SPIRE	RAL
Jeff Payne	SPIRE	RAL

Table of Contents

List of Figures

List of Tables

Applicable Documents

AD01	SPIRE Test Facility Data ICD (SPIRE-UoL-PRJ-001452), Version 1.1
AD02	SPIRE QLA User Guide (SPIRE-RAL-DOC-001428), Issue 0.1
AD03	HCSS use cases for ILT technical note FSCDT/TN-023, Issue 1.3
AD04	SPIRE CQM Performance Test Specification (SPIRE-RAL-DOC-001123), Draft 0.4

Reference Documents

RD01	HCSS Software Verification and Validation Plan (HERSCHEL/HSC/DOC/0284), Issue 1, Revision 1

Glossary

AD	Applicable Document
CQM	Cryogenic Qualification Model
DPU	Data processing unit
EGSE	Electrical Ground Support Equipment
FT	Fourier Transform
GUI	Graphical user interface
HCSS	Herschel Common Science System
HK	Housekeeping
IA	Interactive Analysis
ICD	Interface Control Document
IDL	Interactive Data Language
IE	Instrument Engineer
ILT	Instrument Level Test
IT	Instrument Tester
LIA	Lethbridge Interactive Analysis
NRT-QLA	Near Real Time Quick Look Analysis
RAL	Rutherford Appleton Laboratory
RD	Reference Document
SCOS	Spacecraft Operating System
SPIRE	Spectral and Photometric Imaging Receiver
TBD	To Be Determined
TBW	To Be Written
TC	Telecommand Packet
TFTS	Test Fourier Transform Spectrometer
TM	Telemetry Packet
U500	Unidex 500
UAT	User Acceptance Testing
UofL	University of Lethbridge
USR	User
UU	Unidex units (10 nm units)
VT	Verification tool
ZPD	Zero path difference

1 Introduction

This document describes the software components that are relevant in the context of the Canadian contribution to SPIRE. It defines all the components to be provided by the University of Lethbridge (UofL) and Rutherford Appleton Laboratory (RAL). It also defines functional requirements, delivery timelines and testing requirements.

1.1 Changes

This version of the technical note contains numerous changes based on both the written response to the original draft as well as the subsequent teleconference. In particular, this version features:

- A reorganization of the control interface requirements.
- A more detailed description of the interactive analysis requirements.
- An explanation of the testing procedure to be employed.
- An expanded list of the timelines for the deliverables.
- Open issues that require input from RAL are highlighted.

1.2 Structure of the Document

The following section summarizes all software components that are relevant for the tests to be performed with the TFTS (see Figure 1) in the CQM ILT phase.

The User Requirements, System Specifications, and Test Scenarios are specified for all three applications to be provided by the University of Lethbridge:

TFTS Verification Tool	Section 2
TFTS Communication Server	Section 3
Lethbridge Interactive Analysis	Section 4

User Requirements, System Specifications and Test Scenarios are specified for the functionality to be provided by RAL:

TFTS Control Interface	Section 5
Parameter Display	Section 5

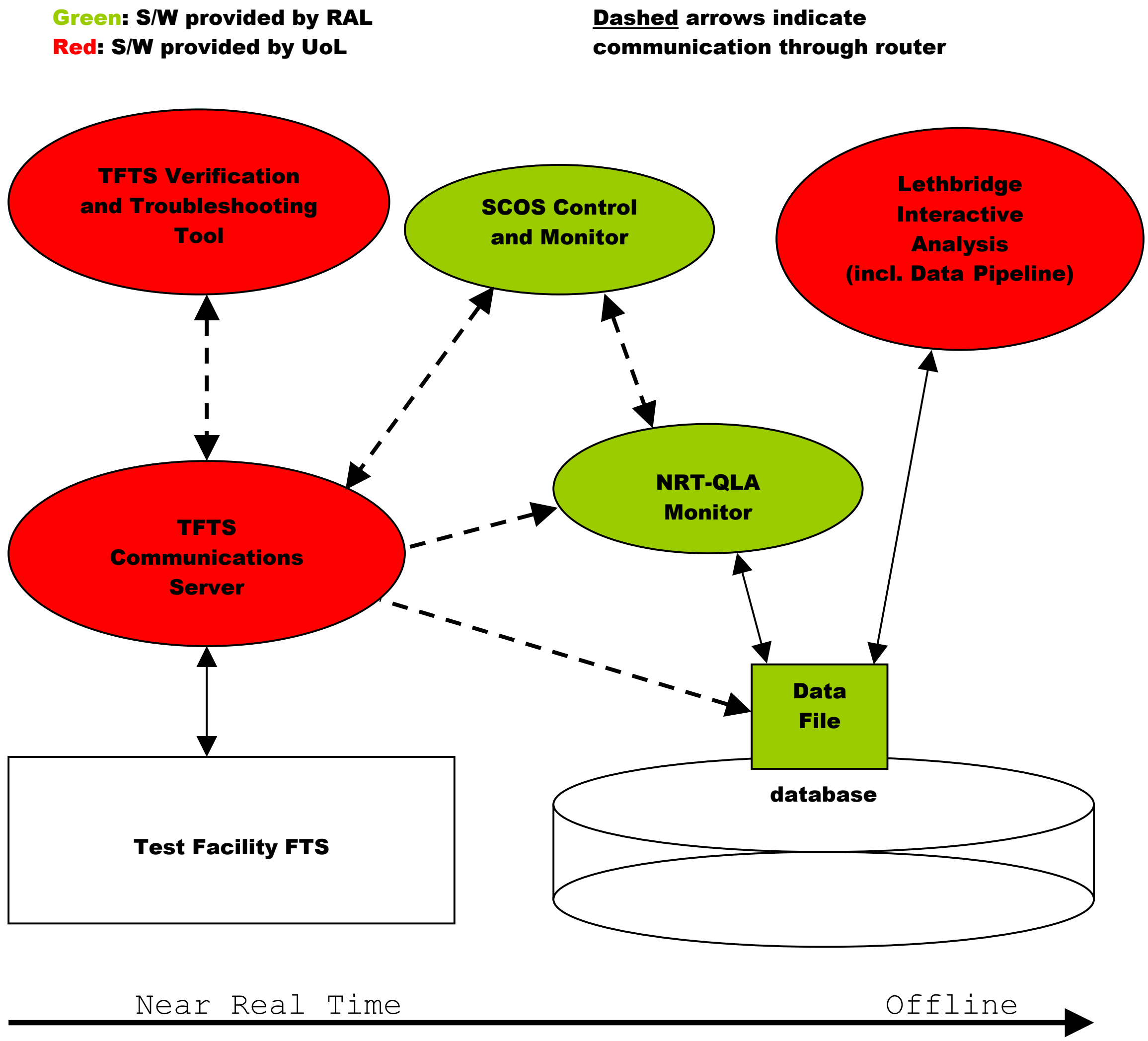

Green: S/W provided by RAL
Red: S/W provided by UoL
Dashed arrows indicate communication through router
TFTS Verification and Troubleshooting Tool
SCOS Control and Monitor
Lethbridge Interactive Analysis (incl. Data Pipeline)
NRT-QLA Monitor
TFTS Communications Server
Data File
database
Test Facility FTS
Near Real Time
Offline

Figure 1: Overview of all applications relevant for the CQM ILT phase with the TFTS

1.3 Overview of relevant software components

The University of Lethbridge will deliver the following software to operate the TFTS in the CQM ILT phase:

1.3.1 TFTS Verification Tool

Purpose

The verification tool is a self-sufficient front-end graphical user interface that the UofL team will use to validate the TFTS instrument. Upon delivery to RAL, the UofL team will use the verification tool to verify that the TFTS instrument is functioning according to specifications and to identify and resolve any problems before the scheduled CQM ILT tests begin.

Users

- Instrument Engineer during the organization of ILT

Interfaces

- Sending TC to the TFTS Communication Server through the Router.
- Receiving TM from the TFTS Communication Server through the Router.
- Science and housekeeping data produced from verification procedures are displayed in the NRT-QLA.
- Accepts commands from the User through a GUI.
- Flags exception reports to the User through a GUI.

Documentation

- Design Specifications (§2)
- User Manual (to be written)

1.3.2 TFTS Communication Server

Purpose

The communication server accepts TC for the TFTS and sends TM packets from the TFTS back to the router.

Users

- N/A

Interfaces

- Accepts TC from SCOS through Router.
- Accepts TC from TFTS Verification Tool through Router.
- Sends TM to Router.
- Sends all accepted TC to the Unidex 500 card that drives the TFTS.

Documentation

- ICD (AD01)
- Unidex 500 Manual (SPIRE-UoL-DOC-001519)

1.3.3 Lethbridge Interactive Analysis

Purpose

This offline tool allows for in-depth analysis of the data collected during ILT. It aggregates and processes relevant data to analyze the spectral response of the bolometers. The data analysis includes a processing pipeline that will serve as a template for the processing pipeline of the Interactive Analysis. Data storage and retrieval is not part of this application.

Users

- Instrument Tester during the ILT phase.

Interfaces

- Reads data files (signal data from the bolometer and the position data from the TFTS) that have been retrieved from the database.
- Writes data files (spectrum for a specific bolometer and possibly other data) that will be ingested into the database.

Documentation

- Design Specifications (§4)
- User Manual.

1.4 Overview of functionality provided by RAL

RAL will provide the following functionality for the operation of the TFTS in the CQM ILT phase:

1.4.1 TFTS Control Interface

Purpose

The TFTS Control Interface enables the test operator to efficiently run the test procedures. The operator will use the interface to issue sequences of TC and individual TC to the TFTS. Any exceptions during the execution of the tests will be flagged to the operator. This functionality will probably be provided as part of SCOS.

Users

- Instrument Tester during the execution of ILT

Interfaces

- Sending predefined sequences of TC to the TFTS Communication Server through the Router.
- Sending individual TC to the TFTS Communication Server through the Router.
- Receiving TM (housekeeping, exception, and acknowledgement reports) from the TFTS Communication Server through the Router.
- The User Interface provides full control over issuing TC and full visibility of TM.

Documentation

- Requirements for test scripts (§5.1.1)
- Requirements for individual commands (§5.1.1)

- Requirements for data display (§5.1.2)

1.4.2 Parameter Monitor

Purpose

The Parameter Monitor will enable the test operator to view system parameters in (near) real time to make sure that the execution of the tests is not corrupted. Housekeeping data from the TFTS and bolometer signals need to be provided to that they can be viewed, printed, and stored by the user. This functionality will probably be provided partly through SCOS and the NRT-QLA.

Users

- Instrument Engineer during the preparation of ILT
- Instrument Tester during the execution of ILT

Interfaces

- Receiving TM from the TFTS Communication Server through the Router.
- Receiving TM from the bolometers through the Router.

Documentation

- Requirements for data display (§5.1.2)
- Requirements for data hardcopies (§5.1.2)
- Requirements for data storage (to be determined)
- SPIRE QLA User Guide, version 0.1 (AD02)

1.5 Quality Assurance

Software development needs to be complemented with tests to assure the quality of the results. For the Herschel mission a software testing plan has been defined (RD01). In order to meet the specific demands for the software development with respect to the Canadian contribution to SPIRE, the general testing strategy needs to be adapted. The specific situation of the software development for the TFTS is characterised by:

a. a small software development team at the University of Lethbridge.
b. split responsibilities for the software development between the UofL and RAL.
c. the need to test software and hardware integration.

The Quality Assurance process for the software development done at the University of Lethbridge will include the following steps:

- Unit Testing
- Integration Testing
- User Acceptance Testing (UAT)
- Operational Readiness Testing

Figure 2 shows the order in which the tests will be performed.

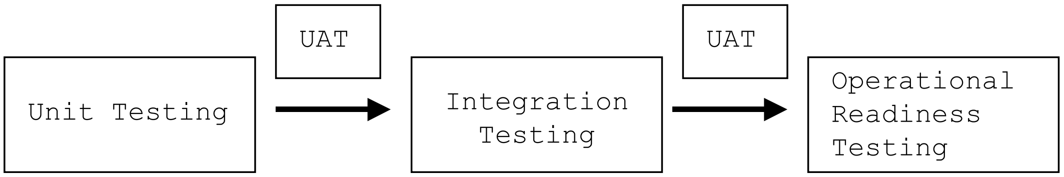


Figure 2: Order of tests

For each software component there will be one test document detailing tests and results. A test environment (a mirror of relevant parts of the EGSE and HCSS) will be created and maintained at the University of Lethbridge for the execution of those tests. In order to identify areas of concern quickly it is necessary to test as early and as frequently as possible. Coordination of development between different applications will be essential to allow early test runs.

1.5.1 Unit Testing

- Unit tests verify the functionality of a single software unit. They are planned and executed by the developer of a software unit. A unit can be any logically coherent part of an application (class, function, the application itself).
- It is the responsibility of the tester to ensure that the user requirements (use cases, other material) are available and consolidated.
- For each unit the main success, alternate, and exception scenarios need to be spelled out, including detailed descriptions, parameter ranges, and expected responses.
- Test results report about the success or failure of tests with supplementary documentation about the nature of potential issues.
- A traceability matrix maps test scenarios onto user requirements to ensure that all functionality required by the user has been tested.
- Test tools (JyPad, Router Monitor (byte level), Router Monitor (command level), TM Generator, Engineering Simulator, HK Generator) will be available for unit tests. However, tests will not be automated.

1.5.2 Integration Testing

- Integration tests verify the functionality of at least two related applications. They are planned and executed by the developers of those applications.
- It is the responsibility of the testers to identify all relevant interfaces and to ensure that the user requirements (use cases, other material) are available and consolidated.
- For each interface, the main success, alternate, and exception scenarios need to be spelled out, including detailed descriptions, parameter ranges, and expected responses.
- Test results report about the success or failure of tests with supplementary documentation about the nature of potential issues.

- A traceability matrix maps test scenarios onto user requirements to ensure that all functionality required by the user has been tested.
- Only applications that will be part of the final system will be used during integration testing. This should help avoid errors that derive from test tools.

1.5.3 User Acceptance Testing

- User Acceptance Tests give users a chance to verify the provided functionality of software. These tests are planned and executed by the users of software based on their requirements.
- Test results report about the success or failure of tests with supplementary documentation about the nature of potential issues.

1.5.4 Operational Readiness Testing

- Operational Readiness Tests focus on the integration of software and hardware. They are planned and executed by the users of software based on their requirements.
- Test results report about the success or failure of tests with supplementary documentation about the nature of potential issues.
- The successful completion of the operational readiness tests is the final step of the software development and the software and hardware will be ready for its intended use.

2 TFTS Verification Tool

The UofL team, upon arrival for the SPIRE CQM ILT, will use the TFTS verification tool to verify that the TFTS instrument is functioning properly. In addition, the UofL team will use the verification tool initialise the TFTS instrument. The TFTS verification tool will communicate directly with the router and serve as a front-end for the TFTS Communication Server.

2.1 User requirements

The user requirements for the TFTS verification tool are presented here as a use case.

2.1.1 Verification tool use case

[User]: Validate TFTS

Brief description:
This use case describes the user requirements for the verification of the TFTS instrument. The verification of the TFTS instrument will be performed upon delivery of the TFTS for the SPIRE CQM ILT.

Actors:
IT: Instrument Tester (primary actor)
VT: TFTS verification tool (secondary actor)

Preconditions:
TFTS: The TFTS instrument is registered with the EGSE router.
HCSS: The EGSE router is functioning properly.

Minimal postconditions:

- Verify that the TFTS executes each of the telecommands in the appropriate manner.
- Find the position of ZPD for the TFTS instrument.

Main success scenario:
1 IT: Start verification tool
2 IT: Perform connection test with TFTS instrument
3 IT: Test TFTS response to telecommands
3.1 VT: Issue each telecommand in series to the TFTS (via the router)
3.2 IT: Observe instrument response
4 IT: Cycle stage between Top and Bottom limits
4.1 IT: Observe instrument response
5 IT: Find ZPD of the TFTS instrument
5.1 IT: Record position of ZPD

Extensions:
4.1a IT: Abort test gracefully
4.2a VT: Issue truncate TC to TFTS

4.1b IT: Abort test immediately
4.2b VT: Issue halt TC to TFTS

Open issues:

- Will the verification tool be available for troubleshooting purposes during the CQM ILT?

Comments:

- None

2.2 System specifications

The UofL development team will develop the verification tool as a front-end graphical user interface (GUI). The functionality of the verification tool GUI must be sufficient to meet all the user requirements presented in §2.1.

2.2.1 Verification tool GUI

The functions that the TFTS Verification Tool will provide may be divided into two main groups. One group consists of functions that pertain to normal operation of the TFTS while the other group consists of those functions that handle exceptions and provide diagnostic information. The design of the TFTS Verification Tool GUI reflects this distinction as it consists of two tab elements. One of the tabs will contain elements relevant to testing the scanning feature of the TFTS (Operation Tab, Figure 3). The other tab will contain elements useful for troubleshooting the TFTS and for handling exceptions (Troubleshooting Tab, Figure 4).

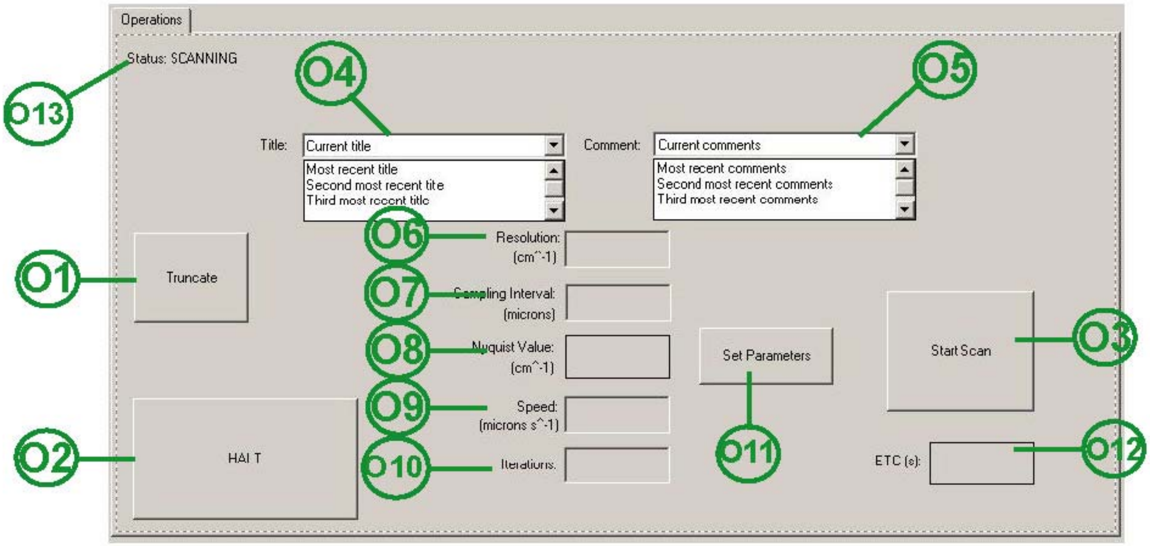


Figure 3: TFTS Verification Tool GUI, Operation Tab

2.2.1.1 Operation Tab

The Operation tab (Figure 3) of the verification tool will contain the following elements:

Exceptions

The functionality of the elements in this portion of the verification tool allows the user to respond to exceptions during a scan test.

O1. Truncate Scan button

Telecommand	Parameters	Reference
Truncate Scan	N/A	AD01§3.2.8.4.13

O2. Halt button

Telecommand	Parameters	Reference
Abort Scan	N/A	AD01§3.2.8.4.12

Normal Operation

The elements in this section of the GUI allow the user to verify that the TFTS performs the scan function properly.

O3. Start Scan button

Telecommand	Parameters		Reference
Read Unidex 500 Parameter	current position		AD01§3.2.8.4.12
Move Table	direction	(see note 1)	AD01§3.2.8.4.6
	distance	(current position - lower limit position)	
Acquire Single-sided Interferogram	Title	(O4)	AD01§3.2.8.4.10
	Comment	(O5)	
	Distance	(upper Limit – lower Limit)	
	Sampling interval	(O7)	
	Velocity	(O9)	
	Number of Iterations	(O10)	
	Acceleration	(V8)	

note 1. IF (current position > lower limit position)
direction = UP
IF (current position < lower limit position)
direction = DOWN

In addition to these buttons, the GUI tab should contain the following data fields and menus:

O4. Title Field: A field that accepts as input a string no greater than 80 characters long. An associated drop-down menu displays the five most recent titles. The operator will have the option to select a title from the drop-down menu but should still be able to edit that selection.

O5. Comment Field: A field that accepts as input a string no greater than 240 characters long. An associated drop-down menu displays the five most recent comments. The operator will have the option to select a comment from the drop-down menu but should still be able to edit that selection.

O6. Resolution field: A field that accepts a floating-point number representing the desired resolution in units of wavenumbers (cm^{-1}).

O7. Sampling interval: A field that accepts a floating-point number representing the sampling interval in units of microns (μm) (This value will have to be converted to unidex units prior to being used as a TC parameter).

O8. Nyquist Value: This is a read only field whose value is determined by the chosen Sampling Interval (O7). The Nyquist Value is equal to 1/(2×Sampling Interval) in units of cm^{-1}.

O9. Speed field: A field that accepts a floating-point number representing the speed of the stage in units of μm/s (This value will have to be converted to (unidex units)/s prior to being used as a TC parameter)

O10. Number of iterations: A field that accepts an integer that must be greater than zero.

O11. Set Parameters button
The operator will press this button to confirm his or her choices of the parameters in fields O6 to O10 for the upcoming scan. Selecting this button will trigger a series of internal calculations to determine if each parameter is within its acceptable range. The value in the Nyquist Value field (O8) will also be calculated and displayed when the Set Parameters button is pressed.

NB Conversion: 1 unidex unit = 10nm, therefore 1μm = 100 unidex units

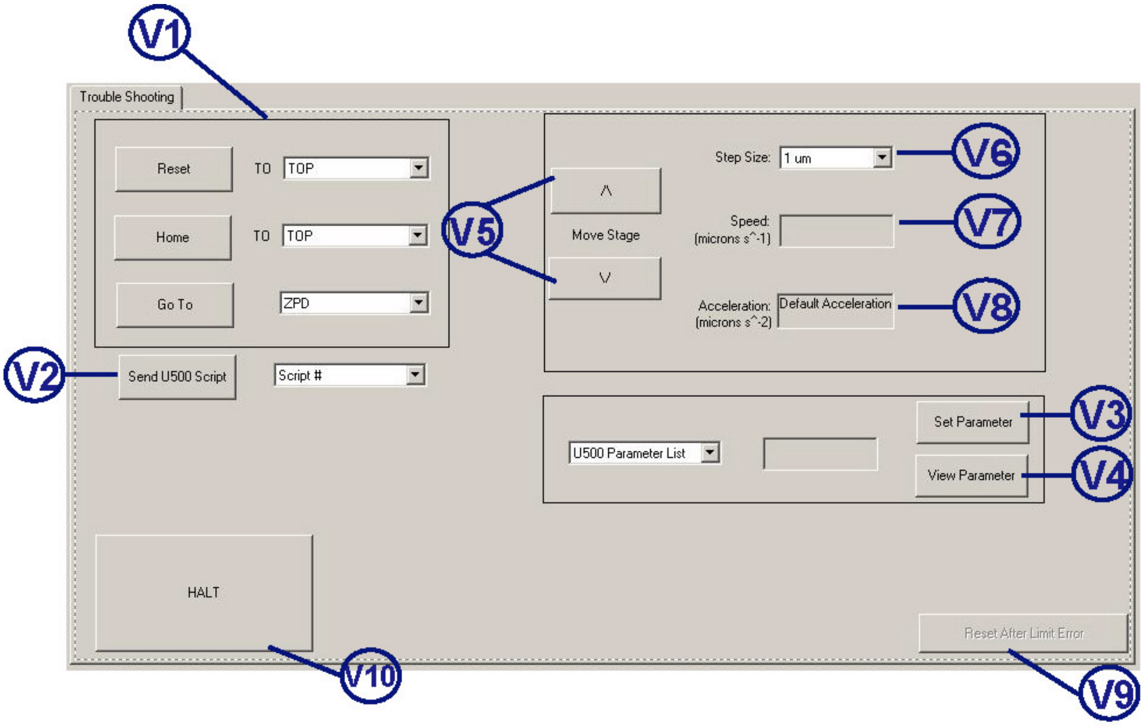
TFTS Status Information

O12. Estimated Time to Complete: The initial value of this read-only field should be calculated based on the values chosen for distance, velocity and number of iterations.

O13. Status field: There are three possible status states for the TFTS as listed in Table 1. The verification tool GUI should show the TFTS in one of these states upon receipt of the associated Telecommand Acceptance Report Telemetery packet (AD01§2.1.5). The verification tool GUI should not show a change in the status of the TFTS until it receives another of the TM packets listed in Table 1.

TFTS Status State	Most Recent TM
Idle	1,7
Scanning	1,1
Exception	5,2

Table 1: TFTS Status States



Trouble Shooting

Reset TO TOP
Home TO TOP
Go To ZPD
Send U500 Script — Script #
Step Size: 1 um
Speed: (microns s^-1)
Acceleration: (microns s^-2) Default Acceleration
U500 Parameter List

Figure 4: TFTS Verification Tool GUI, Tab 2

2.2.1.2 Troubleshooting Tab

The Troubleshooting tab (Figure 4) of the verification tool will contain the following elements:

V1. <u>Reset the TFTS</u>

The purpose of the elements in this group is to reset the TFTS. There are three elements in this group corresponding to the three degrees of reset for the TFTS. The operator chooses the degree of the TFTS reset by selecting one of the buttons in this group and chooses the final location of the stage with the associated dropdown menu.

The functionality of the buttons in this group is as follows:

GUI Button	Action
Go To	3
Home	2, 3
Reset	1, 2, 3

Actions:

1. Reset Unidex 500 card
2. Move stage to calibration position
3. Move stage to a specific position

The telecommand sequences for the reset functions are shown in Table 2.

			Telecommand	Parameters	Reference
Reset			Reset TFTS	N/A	AD01§3.2.8.4.3
	Home		Home TFTS	N/A	AD01§3.2.8.4.4
		Go to	Move Table	direction (see note 1) distance $\vert$calibration pos. - desired pos.[2]$\vert$	AD01§3.2.8.4.6

Table 2: TFTS reset telecommand sequences

note 1. IF (current position > desired position)
Direction =UP
IF (current position < desired position)
Direction = DOWN

note 2. The desired position is that specified by the operator with the associated drop-down selection.

Direct Communication with the Unidex 500

The following GUI functions allow the operator to communicate directly with the Unidex 500.

V2. Send U500 Script button

Telecommand	Parameters	Reference
Run U500 Script	script ID	AD01§3.2.8.4.11

V3. Set Parameter button

Telecommand	Parameters	Reference
Write Unidex 500 Parameter	parameter number (see note 1) parameter value (see note 2)	AD01§3.2.8.4.8

V4. View Parameter button

Telecommand	Parameters	Reference
Read Unidex 500 Parameter	parameter number (see note 1)	AD01§3.2.8.4.7

When the value of this parameter is returned, it is displayed in the associated data field.

note 1. The parameter number is based on the parameter chosen in the associated drop-down menu.

note 2. The parameter value is given by the value in the associated data entry field.

Moving the stage

To find the point of zero optical path difference (ZPD), it may be necessary to move the TFTS stage in a stepwise fashion. This group of the verification tool contains GUI elements designed to allow the operator to perform such a task.

V5. Jog buttons

Two buttons with arrows pointed in opposite directions (UP and DOWN). These buttons will be used to move the stage in the desired direction at a certain step-size, speed and acceleration (based on the values in V6, V7 and V8).

Telecommand	Parameters		Reference
Move Table	direction step size speed acceleration	(see note 1) (V6) (V7) (V8)	AD01§3.2.8.4.7

note 1. IF operator selects UP button
direction =UP
IF operator selects DOWN button
Direction = DOWN

note 2. The operator specifies the values for step size, speed and acceleration for the TC by entering values to the following GUI elements:

V6. Step Size field: A dropdown list that contains a list of possible step sizes (these steps may range from 1μm to 10×10^3μm (1cm)).

V7. Speed field: A field that accepts a floating-point number representing the speed of the stage in units of μm/s (This value will have to be converted to (unidex units)/s prior to being used as a TC parameter).

V8. Acceleration field: This field will be filled with a default value that has not been determined but will be supplied by the UofL team. The Acceleration field will contain a floating-point number representing the acceleration of the stage in units of μm/s^2 (This value will have to be converted to (unidex units)/s^2 prior to being used as a TC parameter)

In addition, the verification GUI will need to prevent the operator from moving the stage beyond the limits of the table. This may require an additional TC to determine the current position of the table.

Exception Handling

The GUI elements contained in this group allow the operator to send the appropriate TC in the event of an error during verification of the TFTS instrument.

V9. Reset After Limit Error button

Telecommand	Parameters	Reference
Reset After Limit Fault	N/A	AD01§3.2.8.4.5

The operator will not be allowed to access this button until the Control GUI receives an Out of Limit Error TM Exception report (§2.2.1.3).

V10. HALT button

Telecommand	Parameters	Reference
Abort Scan	N/A	AD01§3.2.8.4.12

2.2.1.3 Exceptions

The TFTS verification tool will be a stand-alone system; as such, it will be responsible for handling and reporting any exceptions generated by the TFTS instrument during the verification phase. It is essential that the verification tool provides detailed and meaningful error reports.

The TFTS will notify the verification tool of an error by issuing an Exception Report (AD01§4.1.5.2) telemetry packet. According to the TFTS Data ICD (AD01§4.1.21.2.1), there are two types of TM Exception Reports. These are:

E1. DPU Counter Error
E2. Unidex 500 Error

Two fields convey other details with regard to these errors. These are the U500 Status words (32 bits each). One status word contains bits relating to U500 software errors the other status word contains bits relating to U500 hardware errors. Two tables relating the bits in these status words with descriptions of the errors associated with each bit are presented in the TFTS Data ICD (AD01-Table 8 and Table 9).

The receipt of a TM Exception report should trigger the generation of a pop-up notification box within the verification tool. The notification box should inform the operator of the nature of the error or errors by displaying the contents of the software and hardware status words. The operator will then be responsible for correcting the associated error. In the case of an Out of Limit error, regardless of whether this is the only error, the notification box should clearly indicate its occurrence and should further indicate which limit was breached (TOP or BOTTOM). In addition, the Reset After Limit Error button on the verification tool (V9) should become active.

2.3 Test scenarios

The following list of test scenarios is designed to ensure that the verification tool as described in §2.2 meets the user requirements listed in §2.1.

2.3.1 Unit tests

For the TFTS verification tool, the units are defined as the functions that the various elements of the GUI (§2.2.1.1 and §2.2.1.2) are to perform. The scope of these tests is the verification tool itself. The only other systems involved at this stage will be monitoring or simulation tools.

The following is a preliminary list of unit test scenarios for the TFTS verification tool.

TC verification
Use a packet monitoring tool to ensure that parameter values have been converted to the appropriate units.

Use a packet monitoring tool to ensure that the telecommands sent from the verification tool arrive, first at the router then at the test instrument, in the proper format.

TM verification
Use a packet monitoring tool to ensure that the verification tool interprets incoming TM packets correctly.

2.3.2 Integration tests

The integration tests for the TFTS verification tool will complement the unit tests in that the functionality of the GUI units will also be tested at this stage. Where the integration tests differ is that the remote client in this case will be the TFTS communications server rather than a simulator.

The following is a preliminary list of integration test scenarios for the TFTS verification tool.

TC sequence tests
Connect to instrument simulator (through the router) to test whether the verification tool is able to send the appropriate TC in response to particular TM packets.

Latency tests
Test to ensure that the verification tool can process information fast enough to avoid a timeout condition at the TFTS communications server.

2.3.3 User acceptance tests

Verification tool inspection
User test to ensure that the elements contained in the verification tool and the functionality of these elements are satisfactory. A possible result of this test may be to add functionality to the GUI. These added features would need to be unit tested before the integration tests could proceed.

Function test
Once the functions of the verification tool are aggregated into a single GUI, the user will evaluate the functionality of the tool.

2.3.4 Operational readiness tests

The test scenarios for the operational readiness tests of the TFTS verification tool are similar to those at the integration test stage. The added feature of operational readiness tests is that these tests will include the actual TFTS hardware.

The following is a preliminary list of operational readiness test scenarios for the TFTS verification tool.

<u>TC sequence tests</u>
Connect to the actual test instrument (through the router). Test to ensure that the test instrument responds appropriately to the TC and sets of TC sent via the verification tool.

<u>Latency tests</u>
Test to ensure verification tool can process information fast enough to avoid a timeout condition.

2.3.5 Test Traceability Matrix

Table 3 shows the traceability matrix for the test scenarios of the TFTS verification tool.

		User Requirements				
		Unit conversion	**TC format**	**TM interpretation**	**TFTS communication**	**GUI functionality**
UT	**TC verification**	√				
	TM verification			√		
IT	**TC sequence 1**	√	√	√	√	√
	Latency				√	
UAT	**Inspection**					√
	Function	√	√	√	√	√
ORT	**TC sequence 2**	√	√	√	√	√
	Latency				√	

Table 3: Verification tool traceability matrix

Legend:

UT	Unit tests
IT	Integration tests
UAT	User acceptance tests
ORT	Operational readiness tests

2.4 Owner

The UofL development team is responsible for the design, development and delivery of the TFTS verification tool.

2.5 Timeline

The design and development stage of the verification tool application should be complete by April 30, 2003. The testing phase of the verification tool should be completed in time for the verification phase of the TFTS prior to the SPIRE CQM ILT (May 31, 2003).

3 TFTS Communications Server

The details and specifications of the TFTS communications server can be found in the SPIRE Test Facility Data ICD (SPIRE-UoL-PRJ-001452).

3.1 Timeline

The testing phase of the TFTS communications server should be completed in time for the verification phase of the TFTS prior to the SPIRE CQM ILT (May 31, 2003).

4 Lethbridge Interactive Analysis

The Lethbridge Interactive Analysis (LIA) will perform extended processing of the CQM ILT data. The LIA will be written in the Interactive Data Language (IDL). The UofL team will provide the LIA as an analysis tool for the CQM ILT testing phase. The data analysis pipeline described here will serve as the precursor to the data analysis pipeline that will be used in the Interactive Analysis (IA) to be provided in Java in the summer 2004 (§5).

4.1 LIA Requirements

The user requirements for the Lethbridge Interactive Analysis software package are presented here as a use case.

4.1.1 Use Case

[User]: Perform LIA

Brief description:
This use case describes how the LIA will run and its interactions with the HCSS archive. It does not identify the specific details of LIA processing. Some of these steps are based on the Perform QLA use case scenario (see AD03, UCF-747) as the normal mode of the LIA is expected to function in a similar manner to the playback mode of the HCSS QLA.

Actors:
IT: Instrument Tester (primary actor)
LIA: Lethbridge Interactive Analysis software (secondary actor)

Preconditions:
ILT: CQM ILT has been performed (UCF-701).
ILT: The required bolometer and TFTS data is in the archive.

Minimal postconditions:
The Fourier transform of a deglitched interferogram that has been aggregated from collected test data.

Stakeholders and interests:
ICC: Want to use LIA as a means to analyse the spectral response of the photometer and spectrometer bolometers (AD04, section 6.2.17).

Main success scenario:
1 IT: Start LIA
2 IT: Select data file for retrieval (see Open issues)
3 LIA: Aggregate bolometer and TFTS data into an interferogram

4 IT: Interact with LIA to analyse data
5 LIA: Perform requested analysis steps (see comments)
Repeat steps 2 to 5 until analysis complete

Extensions:
Automated analysis:
4a USR: Request automatic analysis
5a LIA: Perform automatic analysis on the data (see comments)

Storing analysis results:
6 LIA: Automatically generate LIA report and store in archive (see Open issues).
6a USR: Create LIA report and request storage in the archive (see Open issues).

Open issues:

- The manner in which the LIA will communicate with the data vault to acquire and store both the bolometer/TFTS data input and the LIA data output, is unclear at this time. There has been discussion that indicates that the LIA will not access the database directly, rather the LIA will access a data file. It is unclear whether this data file will contain the complete interferogram data as well as the TFTS science report; the assumption at this time is that it will. The UofL team will specify the format of this data file once it receives additional information from RAL.
- According to AD03 (UCF-747), the playback mode of the HCSS QLA will be able to request the retrieval of data based on:
    - Observation identifier
    - Time period

  As the LIA will be expected to play a similar data processing role as the HCSS QLA (playback mode), it is reasonable to assume that the LIA will be expected to perform a similar task.
- It has been suggested that it may be necessary to co-add interferograms in order to achieve an acceptable signal to noise ratio. It is unclear at this time where in the analysis pipeline this step should occur. The details of such an operation may best be left for an expanded data pipeline use case.
- It is not clear whether the products at each stage of the pipeline should be made available for output (and subsequently for input). This would only affect step 5 and not step 5a.
- Details of any LIA reporting and its subsequent storage (in the archive database or other) are still to be determined.

Comments:

- (Re: 5, 5a): These steps comprise the data analysis pipeline (§4.2.1). It may be useful to expand these steps as a sub-function use case.

4.2 System specifications

The LIA is an offline tool that allows for in-depth analysis of the data collected during CQM ILT. It is understood that the LIA will not interface directly with a database; rather the input to the LIA and the output from the LIA will be in the form of data files (Figure 5). While the precise format and contents of these data files has as yet not been specified (see Open issues, §4.2.2), it is understood that the LIA input data will consist of the science data from the bolometers and the TFTS and the LIA output data will be the Fourier transformed spectral response of the bolometers. The LIA itself will be an interactive data processing pipeline designed to perform the transformation of the input data.

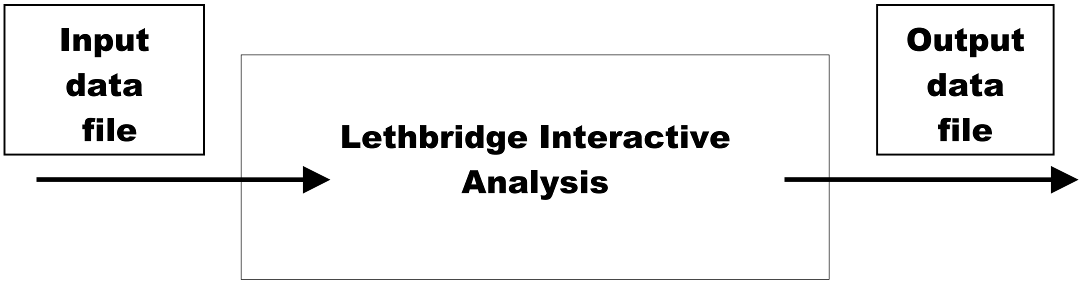


Figure 5: Lethbridge interactive analysis system interface

4.2.1 LIA data processing pipeline

The LIA data processing pipeline contains the functions and routines necessary to transform the raw bolometer (signal vs. time) and TFTS (position vs. time) input data from the CQM ILT into output spectra. Figure 6 shows the LIA data processing modules and the order in which these will be applied to the input data. A description of each LIA processing module is presented below.

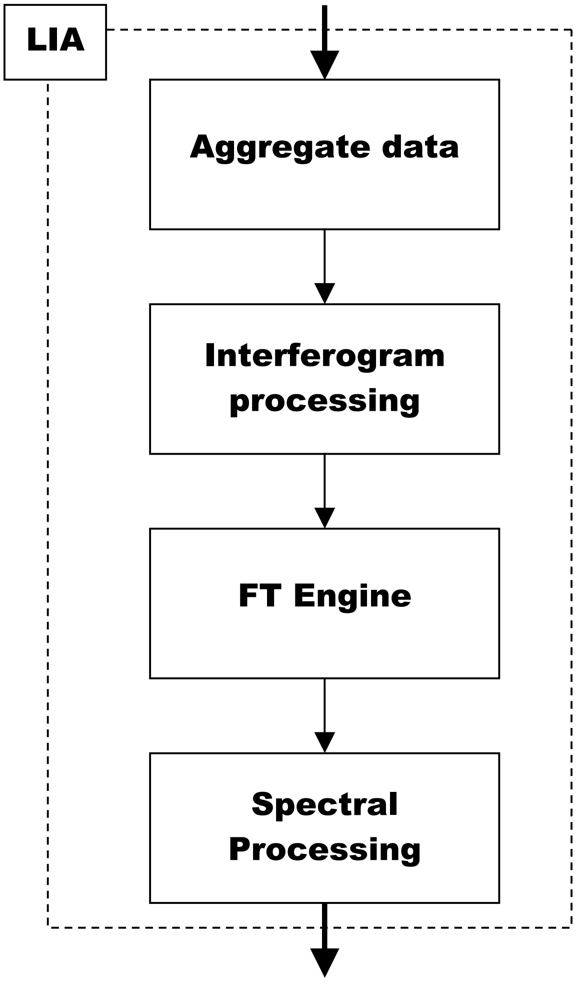


Figure 6: Lethbridge interactive analysis data processing pipeline

1. Aggregate data
   The TFTS nominal science report (AD01§4.1.21.1) and the detector science report are combined at this stage of the LIA data processing pipeline. The result is an interferogram of intensity vs. position.
2. Interferogram processing
   This stage of the LIA data processing pipeline consists of functions that allow the user to inspect and edit the raw interferograms. These processing steps include:
   a. Interactive inspection of the interferograms. This includes functions that will allow the user to pan, zoom, overlay multiple interferograms.
   b. Removing cosmic rays from the interferogram (deglitching). The user will be able to manually remove cosmic rays from the interferogram with the LIA GUI. The LIA may also offer an automated deglitching routine.
3. Fourier Transform (FT) engine

The FT engine module consists of routines that transform the incoming interferogram data into a spectrum. There are two modes of processing in this module: one mode for single-sided interferograms and one mode for double-sided interferograms.

a. Regrid interferogram. Shift the interferogram in order to have one measurement point correspond to the point of zero optical path difference.
b. Apodisation. Remove outlying points in the frequency domain that do not contain scientific information.
c. Phase correction (single-sided interferograms only).
d. Gain correction.
e. Wavelength correction.
f. Fourier transform. Transform the data from intensity vs. position to intensity vs. frequency.

Due to the complex nature of the processing routines contained in the FT engine module, it may be necessary to restrict the amount of degree to which this module is interactive.

4. <u>Spectral processing</u>
   This processing module contains mathematical routines that allow the user to manipulate the spectral products. Some of the routines that this module will feature are listed below.
   a. Averaging
   b. Co-adding
   c. Difference (subtract background)

4.2.2 Open issues

<u>Data Files</u>

There are two main open issues with respect to the data files.

1. **Format**. The precise format of the input and output data files is still unclear.
2. **Content**. Will the input data file contain both the bolometer and the TFTS science data or will these data be in separate files?

<u>Application</u>

1. **Data storage and retrieval**. The LIA will require some sort of data storage and retrieval functionality if only to retrieve the raw interferogram and TFTS data files and store the resulting spectrum. While this is an open issue in and of itself (see above), it is also not clear whether the LIA should allow the user to store data after each module of the processing pipeline.
2. **Processing history**. Where should the file containing the details of the processing performed on a set of data be stored and how should the processing history file be linked to the processed data?
3. **Auto-analysis**. Should the LIA feature an auto-analysis mode that will process the input data with a set of default processing parameters?

4.3 Test scenarios

The following list of test scenarios is designed to use the verification tool as described in §4.2 to meet the user requirements listed in §4.1.

4.3.1 Unit tests

To be determined.

4.3.2 Integration tests

To be determined.

4.3.3 User acceptance tests

To be determined.

4.3.4 Operational readiness tests

To be determined.

4.4 Owner

The UofL development team is responsible for the design, development and delivery of the LIA.

4.5 Timeline

The deadline to complete the development of the LIA is April 30, 2003. A tested version of the LIA that is ready to run at RAL will need to be complete in time for the SPIRE CQM ILT.

5 Specifications for TFTS Operation

A test operator through SCOS, which may provide control and monitoring capabilities, will operate the TFTS. To this end, a number of SCOS control scripts need to be provided. A list of the functional requirements for these scripts is presented below. Error handling will be managed by SCOS and is not presented in this document.

5.1 User requirements

5.1.1 TFTS control interface

The user requirements for the TFTS control interface are presented here as a use case.

[User]: Run TFTS test procedures

Brief description:

This use case describes how the EGSE-ILT control interface will enable the instrument tester to execute the test procedure for the TFTS. The control interface software will be developed by RAL.

Actors:

IT: Instrument Tester (primary actor)

Preconditions:

- Validation of the TFTS instrument has been performed
- The TFTS instrument is registered with the EGSE router.
- The EGSE router is functioning properly.
- Detector array and bolomter have been selected and configured.

Minimal postconditions:

- None.

Success postconditions:

- The TFTS test procedure is successfully executed.

Main success scenario:

1 IT: Initialize TFTS
2 EGSE: Send initialization TC to TFTS instrument
3 IT: Select TFTS test procedure
4 IT: Supply test procedure parameters
5 EGSE: Execute test procedure
5.1 EGSE: Verify test procedure parameters
5.2 EGSE: Send test procedure TC to TFTS instrument
5.3 EGSE: Display TFTS housekeeping parameters

5.4 TFTS: Send end of test procedure TM
5.5 EGSE: Detects end of test procedure
Repeat steps 3 to 5 until ILT is complete

Extensions:
5.2a IT: Abort TFTS test procedure gracefully
5.3a EGSE: Send truncate TC to TFTS instrument
Return to step 1

5.2b IT: Abort TFTS immediately
5.3b EGSE: Send abort TC to TFTS instrument
Return to step 1

Frequency of occurrence:
Often during ILT.

Open issues:
- Will the operator wish to issue individual TC to TFTS instrument?

Comments:
None

5.1.2 Parameter displays

During the normal operation of the TFTS, certain parameters need to be displayed in order to make sure that the TFTS is working properly. The following sections list the parameters in descending order of priority.

1. Essential

At a minimum, three parameters must be displayed during normal operation of the TFTS:

1. The signal of the bolometer currently being tested as a function of time in a graphical representation. In addition to the basic graphical display, additional functions such as zoom and follow would be beneficial.
2. The current speed plus the direction of the TFTS stage as a numerical value. These parameters are contained in the TFTS housekeeping reports (AD01§4.1.3.2).
3. The estimated time to complete a test procedure. The initial value of this parameter may be determined based on the distance, scan speed and number of iterations TC parameters as selected by the operator.

The functionality currently provided by the NRT-QLA is sufficient to understand during tests whether the TFTS is working according to specifications.

2. Useful

In addition to the required parameters, the display of some other parameters would be beneficial for normal operation of the TFTS. From the TFTS housekeeping report (see Table 4), these include:

- The TFTS status
- The current iteration
- The number of requested iterations
- DPU counter (TFTS) time

Parameter Name	Description
OBSID	Observation ID
BBID	Building Block ID
ITERATIONS	Number of scans to perform in a scan sequence
CURR_ITERATION	Current scan iteration being executed
CURR_VELOCITY	Requested velocity of current scan iteration
CURR_ACCELERATION	Requested acceleration of current scan iteration
CURR_SAMP_INTERVAL	Requested sampling interval of current scan iteration
CURR_DISTANCE	Requested scan distance of current scan iteration
DPU_COUNTER_TIME	315 KHz (DPU) timestamp of latest table position
NUM_TC	Number of telecommands received by TFTS
NUM_TM	Number of telemetry packets sent by TFTS
DIRECTION	Stage movement direction UP 0x0000 DOWN 0x0001
TASK_STATUS	TFTS software state IDLE 0x0000 SCANNING 0x0001 ERROR 0x0002
U500_HW_STATUS	State of Unidex 500 hardware

Table 4: TFTS Housekeeping parameters

3. Nice to have

In addition to the displays mentioned previously, it would be nice if a function that allows multiple bolometer signal displays to be overlaid were available. It would also be useful if the bolometer signal display could be converted from signal versus time to signal versus position. The conversion factor (stage speed) for this operation is provided in the TFTS housekeeping report.

5.2 System specifications

During normal operation, the TFTS instrument will operate in three states. These states are ON, STANDBY and SCANNING. The operator moves the TFTS instrument from one state to another by issuing a script command. Figure 7 shows the three states of the TFTS instrument and the control scripts that move the TFTS instrument from one state to another. Note that upon completion of a successful scan, the TFTS moves from the SCANNING to the STANDBY state.

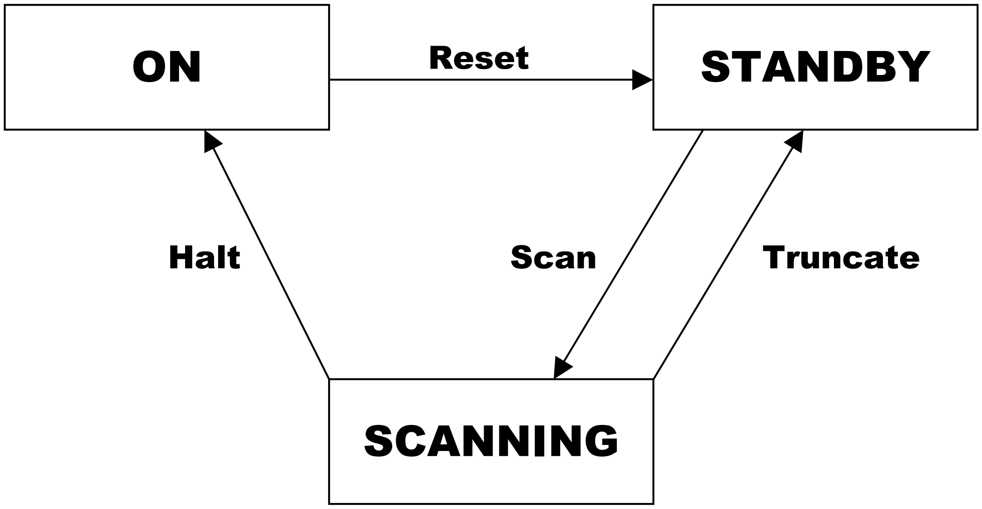


Figure 7: TFTS State Diagram

5.2.1 Control scripts

S1. Truncate

Required script arguments: None

Telecommand	Parameters	Reference
Truncate Scan	N/A	AD01§3.2.8.4.13

S2. Scan

Required script arguments:

- Title
- Comment
- Resolution (in units of cm^{-1})
- Sampling Interval (in units of µm)
- Number of Iterations
- Velocity (in units of µm/s)
- Acceleration (in units of $\mu m/s^2$)

Before any telecommand packets are sent, the script should ensure that the parameters are each within their appropriate ranges.

Argument	Range
Resolution	(see note 1)
Sampling interval	(0.01 — 83,886.07) µm
Number of iterations	1 — $(2^{16}-1)$
Velocity	(0.01 — 55,000) µm/s
Acceleration	(0.01 — 1,000) $\mu m/s^2$

Title	80 characters
Comment	240 characters

Table 5: Ranges for Scan command arguments

note 1. This range has not been determined. The UofL will provide the range for this parameter prior to the CQM ILT.

If the script detects that any of the parameters exceed the ranges as specified in Table 5, execution of the script should stop and the operator should be informed of the error.

Once the script has verified that all the arguments are within their ranges, another set of calculations should be performed. These calculations are necessary to convert the command line arguments into telecommand arguments. The conversion sequence is as follows:

Calculate upper scan limit (A) and scan distance (SD) from the Resolution argument:

```
$SL = 1/(2 \times \text{Resolution})$
IF $SL \geq ZPD$
    $A = TOP$
ELSE
    $A = ZPD-SL$
$SD = SL+(ZPD-A)$
```

The telecomand sequence is as follows:

Telecommand	Parameters		Reference
Read Unidex 500 Parameter	current position		AD01§3.2.8.4.7
Move Table	direction distance speed acceleration	(see note 1) $\lvert$current position-A$\rvert$ (see note 2) (see note 2)	AD01§3.2.8.4.6
Acquire Single-sided Interferogram	title comment distance sampling interval # of iterations speed acceleration	(see note 2) (see note 2) $\lvert$SD$\rvert$ (see note 2) (see note 2) (see note 2) (see note 2)	AD01§3.2.8.4.10

note 1. IF (current position ≥ A (in unidex units))
        direction = UP
    IF (current position < A (in unidex units))

direction = DOWN

note 2. The values for the title, comment, sampling interval, speed and acceleration parameters will be based on the arguments supplied by the operator to the "Scan" script. The script will need to convert the distance, speed and acceleration parameters from the micron units that the operator supplies to unidex units before sending the TC.

NB Conversion: 1 unidex unit = 10nm, therefore 1μm = 100 unidex units

S3. Reset
Required script arguments: None

Telecommand	Parameters		Reference
Reset TFTS	N/A		AD01§3.2.8.4.3
Home TFTS	N/A		AD01§3.2.8.4.4
Read Unidex 500 Parameter	current position		AD01§3.2.8.4.7
Move Table	direction distance speed acceleration	UP current position (see note 1) (see note 1)	AD01§3.2.8.4.6

note 1. The UofL will provide default values for the speed and acceleration TC parameters.

S4. Halt
Required script arguments: None

Telecommand	Parameters	Reference
Abort Scan	N/A	AD01§3.2.8.4.12

5.3 Test scenarios

The following test scenarios are designed to ensure that the elements described in §5.2 satisfy the user requirements listed in §5.1 are satisfied.

5.3.1 Unit tests

The software necessary to implement the CQM ILT test procedure scripts and parameter displays will be developed by the software development team at RAL. As a result, the format of the unit tests is at the discretion of the RAL software development team.

5.3.2 Integration tests

The software necessary to implement the CQM ILT test procedure scripts and parameter displays will be developed by the software development team at RAL. As a result, the format of the integration tests is at the discretion of the RAL software development team.

5.3.3 User acceptance tests

A test environment similar to that which will be used for the CQM ILT will be created and maintained at the University of Lethbridge. As a result, the UofL team will be able to test some of the features of the CQM ILT before arriving at RAL. In particular, the UofL team hopes to test the parameter display functionality (NRT-QLA) with its test environment.

5.3.4 Operational readiness tests

The members of the UofL team will arrive at RAL with the TFTS hardware prior to the CQM ILT. After the TFTS hardware is initialised (§2), the functions and displays required for the CQM ILT should be tested to ensure that the user requirements are met.

Hardware
These tests will ensure that the TFTS hardware responds properly to the test procedure scripts.

Parameter displays
These tests will ensure that the parameter displays that are to be used during the CQM ILT satisfy the user requirements.

5.4 Owner

RAL is responsible for writing the test scripts that will control the TFTS instrument through SCOS. The UofL team is responsible for supplying RAL with the algorithms necessary to write the test scripts.

5.5 Timeline

Implementation of the control and display requirements for the TFTS requirements should be ready for the SPIRE CQM ILT (May 31, 2003).

6 Data Analysis Pipeline

The data processing pipeline will be a by-product of the interactive data processing processing performed in the LIA with additional routines added from the Interactive Analysis (IA) software. Unlike the LIA, which will be written in IDL, the pipeline must be implemented with Java. The scope of the processing pipeline is not clear at this time. As a result, the scope of the pipeline needs to be determined.

Deadline: The data processing pipeline, written in Java, must be supplied to the HCSS development team by July 1, 2004.