Southern Africa Large Telescope
Prime Focus Imaging Spectrograph

SAAO Detector Subsystem:

SALT-3199AS0001: Software Document

SAAO PFIS Detector Subsystem Team:
Dave Carter
Luis Balona
Etienne Bauermeister
Geoff Evans
Willie Koorts
James O’Connor
Darragh O’Donoghue
Faranah Osman
Stan van der Merwe

Issue 1.1
11 February 2003
Issue History

<table>
<thead>
<tr>
<th>Number And File Name</th>
<th>Person</th>
<th>Issue</th>
<th>Date</th>
<th>Change History</th>
</tr>
</thead>
<tbody>
<tr>
<td>SALT-3199AS0001 Software Issue 1.1.doc</td>
<td></td>
<td>1.1</td>
<td>11 Feb 2003</td>
<td>PFIS CDR version</td>
</tr>
</tbody>
</table>

TABLE OF CONTENTS

1 Scope..................................................................................................................................................5
2 Referenced Documents ..........................................................................................................................5
3 Description of Software To Be Developed ..........................................................................................5
  3.1 Requirements Analysis ....................................................................................................................6
  3.2 Software Specification Review (PDR) ...............................................................................................7
  3.3 Software Design ...............................................................................................................................7
  3.4 Software Critical Design Review (CDR) ...........................................................................................7
  3.5 Software Coding and Debug ............................................................................................................7
  3.6 Software Code Reviews ...................................................................................................................8
  3.7 Module Testing ................................................................................................................................8
  3.8 Software Testing ..............................................................................................................................8
  3.9 Subsystem Commissioning and Integration ......................................................................................8
  3.10 Software handover ..........................................................................................................................8
4 Software Safety ......................................................................................................................................9
  4.1 Safety certificate .............................................................................................................................9
  4.2 Communication Integrity ..................................................................................................................9
  4.3 Initialisation ......................................................................................................................................9
  4.4 Start-up and Shut Down Procedure ...................................................................................................9
5 Generic Software Requirements .............................................................................................................10
  5.1 Naming and Tagging Conventions ....................................................................................................10
  5.2 Remote Initialisation .........................................................................................................................10
  5.3 Data ..................................................................................................................................................10
  5.4 Software cyclic execution .................................................................................................................10
  5.5 Data time stamping ............................................................................................................................10
  5.6 Modular Design .................................................................................................................................10
  5.7 Measuring Units ................................................................................................................................11
  5.8 Synchronisation ................................................................................................................................11
  5.9 Unused Code ......................................................................................................................................11
  5.10 Software Comments .........................................................................................................................11
  5.11 Self-changing code ..........................................................................................................................12
  5.12 Communication methods ................................................................................................................12
6 Specific PDET Software Requirements ...............................................................................................12
  6.1 Operating Systems ............................................................................................................................12
  6.2 Development Software .......................................................................................................................12
  6.3 Application Software ..........................................................................................................................12
  6.4 Man-Machine Interfaces ....................................................................................................................12
7 Deliverables ...........................................................................................................................................13
8 Configuration Control

9 Software Specification

9.1 PDET CON

9.1.1 Functional Requirements: PDET CON

9.1.2 Program/Exposure Initiation/Termination

9.1.3 Image Display and Interaction

9.1.4 Data Storage

9.1.5 Peak-Up

9.1.6 Communication With Precision Time Source

9.1.7 Communication With PDET KER

9.1.8 Communication With PDET PCI

9.1.9 Communication With PDET SDSU

9.2 PDET KER

9.2.1 Functional Requirements: PDET KER

9.2.2 Communication With PDET MMI

9.2.3 Communication With PCONDI

9.2.4 Communication With The TCS

9.2.5 Communication With The Science Database

9.2.6 Communication With PDET CON

9.3 PDET MMI

9.3.1 Functional Requirements: PDET MMI

9.4 PDET SDSU

9.4.1 Functional Requirements: PDET SDSU

9.5 Sub-Systems Controller

10 Technical Requirements

10.1 Software Architecture

10.2 Software Interfaces

10.3 Modes, States and Events

10.4 Software Capabilities

10.4.1 Communication

10.4.2 Initialisation

10.4.3 Command Interpretation and Generation

10.4.4 Status Reporting

10.5 Operating System

10.6 Resource Allocation

11 Generic Software Requirements

12 Software Testing
ACRONYMS AND ABBREVIATIONS

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATP</td>
<td>Acceptance Test Procedure</td>
</tr>
<tr>
<td>ATR</td>
<td>Acceptance Test Report</td>
</tr>
<tr>
<td>BITE</td>
<td>Built-in Test Equipment</td>
</tr>
<tr>
<td>CDR</td>
<td>Critical Design Review</td>
</tr>
<tr>
<td>COTS</td>
<td>Commercial off the shelf</td>
</tr>
<tr>
<td>ELS</td>
<td>Event Logger Software</td>
</tr>
<tr>
<td>HET</td>
<td>Hobby-Eberly Telescope</td>
</tr>
<tr>
<td>I/O</td>
<td>Input/Output (Device)</td>
</tr>
<tr>
<td>ICD</td>
<td>Interface Control Dossier</td>
</tr>
<tr>
<td>MMI</td>
<td>Man-Machine Interface</td>
</tr>
<tr>
<td>MTBF</td>
<td>Mean Time Between Failures</td>
</tr>
<tr>
<td>MTTR</td>
<td>Mean Time to Repair</td>
</tr>
<tr>
<td>OEM</td>
<td>Original Equipment Manufacturer</td>
</tr>
<tr>
<td>OPT</td>
<td>Operational Planning Tool</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer</td>
</tr>
<tr>
<td>PDR</td>
<td>Preliminary Design Review</td>
</tr>
<tr>
<td>PFIS</td>
<td>Prime Focus Imaging Spectrograph</td>
</tr>
<tr>
<td>PI</td>
<td>Principal Investigator (Astronomer)</td>
</tr>
<tr>
<td>PIPT</td>
<td>PI Planning Tool</td>
</tr>
<tr>
<td>PLC</td>
<td>Programmable-Logic Controller</td>
</tr>
<tr>
<td>SA</td>
<td>SALT Astronomer</td>
</tr>
<tr>
<td>SALT</td>
<td>Southern African Large Telescope</td>
</tr>
<tr>
<td>SAMMI</td>
<td>SA Machine Interface</td>
</tr>
<tr>
<td>SC</td>
<td>Software Component (e.g. part of the TCSS)</td>
</tr>
<tr>
<td>SDB</td>
<td>Science Database</td>
</tr>
<tr>
<td>SD</td>
<td>Software Design</td>
</tr>
<tr>
<td>SDP</td>
<td>Software Development Plan</td>
</tr>
<tr>
<td>SI</td>
<td>Software Item (the TCSS is a Software Item)</td>
</tr>
<tr>
<td>SO</td>
<td>SALT Operator</td>
</tr>
<tr>
<td>SOMMI</td>
<td>SO Machine Interface</td>
</tr>
<tr>
<td>SRS</td>
<td>Software Requirement Specification</td>
</tr>
<tr>
<td>STARCAT</td>
<td>Object Catalogue</td>
</tr>
<tr>
<td>SW</td>
<td>Software</td>
</tr>
<tr>
<td>TBC</td>
<td>To Be Confirmed</td>
</tr>
<tr>
<td>TBD</td>
<td>To Be Determined</td>
</tr>
<tr>
<td>TCS</td>
<td>Telescope Control System</td>
</tr>
<tr>
<td>TCSS</td>
<td>TCS Server</td>
</tr>
<tr>
<td>VI</td>
<td>Virtual Instrument (Labview function)</td>
</tr>
</tbody>
</table>
1 Scope

The PFIS detector package is being supplied to the University of Wisconsin by the SAAO. This document specifies all aspects of the software for the PFIS Detector Package.

2 Referenced Documents

The following documents are referenced in this specification and are applicable to the extent specified herein.

1000AA0030 SALT Safety Analysis
1000AB0044 SALT Labview Coding Standard
1000AD0005 SALT Computer Architecture
1000AS0040 SALT Operational Requirements
1700AS0001 TCS Specification
1773AS0001 TCS Interlock Panel Specification
1000AS0049 SALT Data Interface Control Dossier

3 Description of Software To Be Developed

The PFIS detector software comprises the following computers and units/applications. Only the software in bold is new application software that is covered by the development plan. The software in bold italics is assumed to be the responsibility of the University of Wisconsin or the SALT Project.

a. PFIS Control PC (hereafter called PCON)
   - *PCON Control Program Software.*
   - *PCON Detector Interface (designated PCONDI) Software. This is the main interface to the PDET.*
   - Labview Data Socket (part of the standard Labview Application)

b. PFIS Detector PC (hereafter called PDET)
   - *PDET Kernel Software (designated PDET KER). This will interact with the PCONDI residing in the PCON machine. PDET KER will also interact with PDET MMI and PDET CON as described below.*
   - *PDET MMI Software (designated PDET MMI). This is the interface to PDET for development and maintenance via the PDET PC keyboard. It will be similar to PCONDI, although the latter will exert control not via an MMI.*
   - *PDET Control Software (designated PDET CON). This is software that will receive instructions from PCONDI, and control all the detector hardware.*
   - *PDET PCI Card Software (designated PDET PCI). This is software that is supplied by Astronomical Research Cameras with their SDSU II CCD controllers. If Real Time Linux is used, its functionality within the Real Time Linux operating system environment will be emulated by SAAO developed software.*
   - *PDET SDSU III Control Software (designated PDET SDSU), including the software in the subsytem controller. This is software that is initially supplied by Astronomical*
Research Cameras with their SDSU II CCD controllers. The supplied software will be used as a prototype for an SAAO developed equivalent, tailored for the PDET application.

- This machine will contain no other applications.

c. Data Reduction PC

- This will be very similar to the PI computer, but will be located at SALT.
- **PDET Data reduction pipeline for SI mode (designated PDET DRED).**

In addition, the PDET software may interact with these machines forming part of the SALT TCS. Their main applications software units are also indicated:

d. TCS Server

- SALT TCS Server application
- Labview Data Socket (part of the standard Labview Application)

e. Data Processor

- Science database. This is the organised storage and retrieval of all instrument configuration, calibration, science and telescope data pertaining to science observations made.

f. Event Logger

- Event Logger application. This software is used to record, retrieve and display user-defined events, based on data flowing between the TCS components and the telescope subsystems. A second function is to display telescope status and failure information that is vital to both the Astronomer and Operator.

Table 1 shows the planned schedule for the work.

<table>
<thead>
<tr>
<th>Milestone</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development Plan &amp; Specification</td>
<td>March 2003</td>
</tr>
<tr>
<td>Design &amp; Prototype</td>
<td>15 Jun 2003</td>
</tr>
<tr>
<td>Coding &amp; Test of Modules Complete</td>
<td>15 Sep 2003</td>
</tr>
<tr>
<td>Integration Complete</td>
<td>15 Oct 2003</td>
</tr>
<tr>
<td>ATP</td>
<td>Mid-Oct 2003</td>
</tr>
<tr>
<td>Science data reduction pipeline</td>
<td>1 Mar 2004</td>
</tr>
</tbody>
</table>

3.1 Requirements Analysis

A distinct software specification (SRS) shall be written for each of the following software items, using the document number 3199AS0001:

<table>
<thead>
<tr>
<th>Title</th>
<th>Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDET KER Software Specification</td>
<td>PDET KER</td>
</tr>
<tr>
<td>PDET MMI Software Specification</td>
<td>PDET MMI</td>
</tr>
<tr>
<td>PDET Control Software Specification</td>
<td>PDET CON</td>
</tr>
<tr>
<td>PDET PCI Card Software Specification</td>
<td>PDET PCI</td>
</tr>
<tr>
<td>PDET SDSU Control Software Specification</td>
<td>PDET SDSU</td>
</tr>
</tbody>
</table>
We assume there already exist requirements analysis for PCOND1.

3.2 Software Specification Review (PDR)

Prior to starting the software design, the Software Requirement Specification shall be reviewed by the development team and the SALT Team. The purpose of the review is to verify that the software requirements have been correctly identified.

3.3 Software Design

Prior to coding the software, it is essential to structure and design the software to meet not only the functional requirements of the SRS, but also the maintainability, reliability and testability requirements. The output of the Software Design process will be in various forms, but the major design aspects shall be documented in the Software Design Section (SD). At least the following shall be addressed:

(a) A high-level **design description**, describing the overall integration and interaction of the modules how this relates to the software states and modes.

(b) An updated copy of the **software architecture diagram**.

(c) A detailed **functional-flow and data-flow diagram**, showing all the software modules and the precise data flowing between them. The implementation of specific timing, synchronisation and interrupts requirements shall be illustrated. For Labview software, the mechanism of data flow (i.e. wires or VI server calls) shall be identified.

(d) The **software design** of each module must be provided. This shall indicate the specific data inputs, outputs, processing and timing requirements for that module and shall give specific formula’s and algorithms that are to be executed. Details of global variables, interrupt operation, timing implementation and shall be defined. The design shall be documented in pseudo-code, flow diagrams or English narrative.

3.4 Software Critical Design Review (CDR)

Prior to full-scale software coding, the software design shall be reviewed by the supplier development team and the client. The purpose of the review is to verify that the requirements of the SRS and other implicit requirements have been adequately and efficiently addressed in the design. It is an opportunity for the development team to co-ordinate the hardware, software and equipment designs and to ensure that non-functional requirements such as maintainability, testability and reliability are adequate.

The CDR shall address the overall software design (architecture, data flow, timing) and detailed design of each module.

3.5 Software Coding and Debug

During this process the software code for each module is generated according to the design defined in the SD. Specific coding standards, metrics and conventions are applied (as defined elsewhere in this document) and software comments inserted.

In parallel with the software coding process, a **Software Acceptance Test Procedure (ATP)** is defined
and documented by the developer. Tests shall be defined to verify that the software complies with each requirement of the SRS. This document shall subject to approval by the University of Wisconsin.

3.6 Software Code Reviews

The source code of each completed module is reviewed by the development team to check the appropriateness of software style, efficiency and to co-ordinate interfacing modules. The appropriate method of testing each module shall be agreed. The client may at his discretion attend such reviews. A record shall be kept of each review and the comments recorded. The implementation of such comments shall be verified during module testing.

3.7 Module Testing

Software modules shall be individually tested prior to integration with the other modules, to an appropriate level. Testing a module may use either a simple stub simulating interfaces to other modules or another module (or group of modules) that has already been tested. The results of each module test shall be recorded, albeit informally.

3.8 Software Testing

Tested modules are incrementally integrated together and progressively checked. When all the software has been integrated, the tests defined in the Software ATP shall be executed where possible. The precise hardware and software configuration tested shall be defined. From this point forward, all software changes shall be logged. A TCS Server simulator shall be used to verify the communications interface to each computer prior to delivery of that computer and its software by the developer. A communication test using the simulator shall be part of each computer item’s ATP.

At this point the software shall be fully under Configuration Control (see section 8) and all software changes managed.

3.9 Subsystem Commissioning and Integration

The next step of the process is to integrate the software with the PCONDI software. Commissioning is complete when the PDET Software ATP, which verifies the performance against its specification, has been passed.

The final step of the process, during which the final aspects of the software items performance is verified, is the System Integration, when all the subsystems are integrated to form an operating instrument. Only when the PDET ATP has been successfully completed, can each SW item be said to be complete.

3.10 Software handover

During step 3.9, the responsibility for maintenance of the software is transferred from the original developer to the U. of Wisconsin. This delivered software package shall contain a full definition of the latest software configuration, including the following:

a. A Version Definition – a table indicating the current revision numbers of each of the software modules of each software item

b. The Software Configuration Definition – an electronic copy of all configuration data for
operating systems, firmware, set-up data, calibration constants, user-defined parameters etc.
c. The *Software Source Code* of the present software version
d. *Original legal copies* of the operating systems, compilers, tools, utilities that are required to maintain the software
e. Final copies of *Operating, Maintenance and Calibration procedures* where applicable
f. A final version of the *Safety Certificate*

4 Software Safety

4.1 Safety certificate

A Safety Certificate shall be issued for PDET Software. The certificate shall identify all the software items that form part of the PDET software suite.

4.2 Communication Integrity

Communication integrity between subsystems and all equipment items shall be monitored by all items receiving data. Failure to receive correct data or failure to receive any data from a particular device shall be reported the operator via the Event Logger.

Detection of communication failure shall be facilitated by using the “Validity Word” in the communicated Labview data, or a similar method for non-Labview Software.

Each software item shall fail in a safe fashion if it does not receive the required data. Gradual degradation of system performance should be allowed where possible.

4.3 Initialisation

PDET software shall be in a safe state when un-initialised or switched off. Similarly, un-initialised inputs (e.g. from other subsystems) shall not cause incorrect responses from the software.

The following initialisation sequence shall be followed by all software:

a. Switch all outputs to a safe state (e.g. motors, OFF)
b. Indicate “Initialisation State” to the operator
c. Check the integrity of the processing hardware and memory using simple arithmetic checks
d. Check communication with and correctness of peripheral devices (if applicable)
e. Verify the correctness of configuration data and then initialise variables accordingly
f. Check communication with interfacing computers
g. If all operations are successful, report “System Okay” to the operator and enter into a “ready” state, where after the state will be determined by switches, commands, data etc. If operations a. to e. are not successful, report “System Start Failure” and indicate the type of failure encountered. If communications with another computer cannot be established, this should be reported.

4.4 Start-up and Shut Down Procedure
During Start-up and Shut Down, preventative measures shall be taken to handle process conditions as well as Inputs and Outputs in a safe manner.

5 Generic Software Requirements

5.1 Naming and Tagging Conventions

Each SW component shall be uniquely identified with a sensible name. File extensions native to the programming language used, shall be adhered to (i.e. Labview files *.vi, *.glb, *.ctl and *.rtm).

All variables, memory and block naming shall be clear, logical and understandable. A uniform convention shall be used throughout an item, preferably using whole English words. Where compilers/interpreters do not support long variable names, a consistent abbreviation may be used, with a clear definition in the appropriate documentation.

Naming conventions will be agreed during the Software PDR.

5.2 Remote Initialisation

It shall be possible to trigger the initialisation sequence described in 4.3 remotely via the normal communication to an item. (e.g. The operator must be able to send a “reset” command across the Ethernet to any computer to trigger initialisation). This is not applicable to MMI applications.

5.3 Data

A set of Critical Data, over and above data required for functional operation, shall be agreed with the client for each computer item. This data set shall be updated at a rate of at least 1Hz and be sent to the Event Logger:

- Item Mode
- Item Health Status
- Fault list

This Data Set will be finalised during the Critical Design Review.

Where internal variables may assist diagnostics, their values should also be transmitted to the Event Logger.

5.4 Software cyclic execution

After the completion of initialisation, the code of a SW item shall execute in a cyclic fashion, at a constant rate, commensurate with the control bandwidth/frequency/latency required.

5.5 Data time stamping

Time-critical data will be agreed with each supplier and identified as such in the ICD. All such data shall be time-stamped in an agreed fashion to facilitate synchronisation of subsystems.

5.6 Modular Design
Software shall be designed in a scalable and modular fashion. All software modules (i.e. Labview VI’s, procedure and functions – see SALT Labview Coding Standard) shall be designed to minimise their data interfaces and to group functions that belong together, keeping in mind future growth and hardware upgrades. Compliance to these requirements shall be demonstrated at the PDR, CDR and code reviews. In particular, the following types of functions shall be in independent modules:

- Input/Output hardware communication drivers
- Input/Output scaling from hardware units (e.g. 1024bits) to/from engineering units.
- Initialisation sequences
- User configuration sequences
- Equipment mode/state control
- Mathematical/control algorithms
- Data storage and retrieval
- Data communication
- Fault monitoring and reporting

Identical software functions shall not be repeated in different areas but rather grouped together as a shared function or procedure.

5.7 Measuring Units

The SI metric system shall be used for all processing except for angles, which shall be in Radians. The units of information displayed on MMI displays shall be in “human-friendly” units and will be agreed during the CDR.

5.8 Synchronisation

Two methods of synchronisation are allowed, the selection of which shall be commensurate with the time accuracy requirements and shall be subject to approval during the PDR.

   a. *Network synchronisation*: An NTP server will provide accurate GPS time to all subsystems requesting this via Ethernet. The accuracy of this time should be better than 150ms and would be suitable for most applications.

   b. *Hardware synchronisation*: A precision hardware time signal (e.g. 1 pulse-per-second and 10MHz) will be made available to all items requiring very accurate time (e.g., Tracker Computer, Payload Computer and Instrument Computers). A computer input reads this signal and synchronises SW functions accordingly.

5.9 Unused Code

All unused code and variables shall be removed from the software.

5.10 Software Comments

Over and above the software development documentation, the following documentation shall form and integral part of the software code in the form of comments or function help:

- Each software module shall have a header or associated “help” definition describing the following:
  - The name and purpose of the module
The inputs and outputs of the module and their types
A detailed description of the functions performed by the module. (This may be English narrative or pseudo-code).
- A definition/description of local and global variables used in the module
- English description of the actions performed by SW code. As a guideline give one line of comments per two lines of code for text-based software. For Labview software each VI shall have help information defined, as indicated in the first bullet.

5.11 Self-changing code

Self-changing code shall not be allowed.

5.12 Communication methods

Communication between computers on the Ethernet network, shall be TCP/IP-based Data socket communication or http://, as agreed.

A central database (ICD) shall identify the source, destination, type and update frequency of each parameter. Communications software for Labview items will be provided by SALT, according to the required data types.

6 Specific PDET Software Requirements

6.1 Operating Systems

The Real Time Linux Operating System is proposed for the PDET PC. (It is possible that standard Linux may suffice).

6.2 Development Software

The following development environments are proposed:
- PDET KER: Labview 6
- PDET MMI: Labview 6
- PDET CON: C
- PDET PCI: C
- PDET SDSU: DSP Assembler
- PDET DRED: SQL, C

6.3 Application Software

The SALT Labview Coding Standard shall be applied to all Labview software. The Software Engineering Primer in “Labview Power Programming” and “Internet Applications in Labview” are recommended reading.

6.4 Man-Machine Interfaces
All MMI’s shall be subject to approval by the University of Wisconsin.

7 Deliverables

Unless agreed otherwise, at least the following items shall be delivered for each software item.

a. Original documentation and electronic media of all the bought-out software installed on the computer (including operating systems and device drivers).

b. Original documentation and electronic media of the software development environment in which the software was developed (e.g. compilers, version control tools etc.), unless agreed otherwise.

c. Development environment and operating system configuration data (e.g. memory map set-ups, compiler directives, copy of Linux configuration files, Windows INI files, registry files) on CD-ROM.

d. Documentation as described in section 4 (if applicable according to Table 1), including the following:
   - Software Requirement Specification
   - Software Design Document
   - Software Code Review report/minutes
   - Software Acceptance Test Procedure
   - Software Acceptance Test Report

e. All application software source code, compiled software, installation software and configuration information on CD-ROM.

f. Calibration, Maintenance and operating procedures if applicable

g. A Version Definition (See section 8)

The following will be available for the PDET Software as a whole:

h. A final version of the Safety Certificate
i. A Software Development Plan (this document)
j. Acceptance Test Procedure and Report

8 Configuration Control

Each SW Item shall be uniquely identified by the PFIS configuration name defined earlier.

During the software coding, testing and integration process, maintain the Labview version control tool shall be used, whereby each software module has a revision number reflecting changes made. Changes after initial SW integration shall be controlled and documented. Every update to that module shall result in a change in the revision number of that module. Modules of previous revisions shall not be overwritten or destroyed but kept for recovery purposes.

The integrated software comprising many modules (i.e. the SW Item), shall also have a unique version number at critical stages in the process (e.g. at Software Testing, delivery etc.). A document (the Version Definition) shall be maintained which records the included modules and their revision numbers making up each version of a SW Item.
The following numbering scheme is preferred for revisions and versions:

Example: Tracker Payload SW Version 2.32 comprises the following modules
- Module A: *Initialisation* Revision 5
- Module B: *Hardware set-up* Revision 31
- Module C: *Mode control* Revision 29
- ……………

Where the number before the dot is incremented with major changes to the software or to mark a significant event (e.g. software delivery) and the number after the digit changes with minor modifications.
Figure 1. PDET Modules From The Software Perspective
9 Software Specification

PDET will have its own fully functioning MMI, which may be used not only to enable the human operator to control the detector, but also for development and maintenance purposes. In normal operation, control shall be via PCONDI. It is possible that the PDET MMI will be used by PCONDI for controlling the detector. PCONDI is the responsibility of the University of Wisconsin.

The relationships of these software items to each other are shown in Fig. 1. The main purpose of the software suite is to:

- Interact with PCONDI/PDET MMI to enable them to control the PDET hardware to obtain images/spectra arriving at the PFIS focal plane.
- Display the data on the PDET PC monitor near the SA. The SAMMI/PDET MMI should be able to control this display, as well as interact with it (placement of markers etc.).
- Store the images on the PDET PC disc in FITS format.
- Enable the SAMMI/PDET MMI to use simple algorithms to make quantitative measurements on the displayed image.
- Interact with the TCSS to obtain telescope status information.
- Monitor the safety of the detector and preventing unsafe or inappropriate operation.

9.1 PDET CON

PDET CON will be the “central control room” of the detector. For this reason, we consider its functional requirements first.

9.1.1 Functional Requirements: PDET CON

PDET CON will control the master copy of all detector parameters. These parameters are listed here:

1. The “program” defining the sequence of operations and hardware parameters for the next observation.
2. If not specified in the “program”, what exposure time is required for the next image.
3. If not specified in the “program”, what pre-binning is required for the next image.
4. If not specified in the “program”, what windowing, if any, is required for the next image.
5. If not specified in the “program”, what gain is required for the next image.
6. If not specified in the “program”, what readout speed is required for the next image.
7. If not specified in the “program”, whether or not the next image is to be stored on disk.
8. If not specified in the “program”, values of other parameters (TBD). A particularly important parameter here is whether vertical operations are needed during the exposure (for nod and shuffle or similar kinds of manipulation of the detector).
9. An important additional parameter is whether or not repeat mode is enabled.

The concept of a “program” is an important one for PDET detector control. All images are obtained in with a “program”. The values of all camera parameters may be predefined in a specific “program”. For example, a sequence of spectra with different exposure times etc. can be defined. Other “programs” could be defined for obtaining bias images, dark images, repeated short exposures etc.
On the other hand, users can set all individual camera parameters (2-8 in the above list). In addition, single exposures or programs can be placed in repeat mode (item 9 in the above list).

The “snapshot” program is one in which no camera parameters are predefined; instead, they are read from the current values in the MMI front panel.

In this manner, flexibility of operation as well as ease of routine use are assured. This catering for “opposite ends of the operational spectrum” is sufficiently important that it assumes the position of a requirement of the software, as opposed to a design feature.

9.1.2 Program/Exposure Initiation/Termination

PDET CON can initiate the start of the next program/exposure. PCON DET can also abort a currently executing program/exposure.

9.1.3 Image Display and Interaction

PDET CON will display images obtained by the detector with adequate resolution on a large monitor in the SALT Control Room. This monitor will be the PDET PC Monitor (as indicated in Fig. 1). Control of brightness and contrast shall be part of the display algorithms.

PDET CON will also manage interaction with the displayed image to place marks/boxes/windows on the display, extract spectra, overlay templates on the display etc.

9.1.4 Data Storage

PDET CON will also control and manage data storage (in FITS format) on the PDET PC’s disk. (It is necessary that PDET CON carry out this task as opposed to PDET KER/PDET MMI/PCONDI as in very high speed mode, minimum latency is required).

9.1.5 Peak-Up

PDET CON will also perform any calculations required for peaking up a target in the entrance aperture of PFIS. These calculations will return the total counts within a window.

9.1.6 Communication With Precision Time Source

PDET CON/KER shall obtain data from the Precision Time Source in order to ensure that the PDET computer keeps Universal Time to 1 millisec accuracy or better at all times by providing suitably coded GPS-referenced time to the PDET computer.

9.1.7 Communication With PDET KER

PCONDI shall exchange data with PDET KER in order to:

- Receive commands from PDET KER and provide update confirmation of changes to PDET hardware parameters (items 1-8 at the start of this Section). Definitions of any valid programs are included here.
- Receive commands from PDET KER to initiate/terminate the next program/exposure or
terminate the current program/exposure. Any vertical operations are requested similarly. Reporting of the outcome of the command is also required.

- Receive commands from PDET KER to enable interaction with the PDET PC Monitor: place marks/boxes/windows on the display, select objects on the display, overlay templates on the display etc. (Status return is via the appearance of the display).
- Receive commands from PDET KER about the storage of data in FITS format. Reporting the outcome of the command is also required.
- Receive commands from PDET KER in order to carry out simple image processing algorithms:
  i. Extraction of spectra.
  ii. Fitting of Gaussian functions to point sources in imaging mode.
  iii. 2-d line plots of image brightness in any direction across the image.
  iv. Simple aperture photometry to estimate magnitudes of point source objects.
  v. Simple image processing function such as background subtraction, smoothing or median filtering.

Results of these operations will be reported either directly to PDET KER or via the PDET PC display.

- Receive commands from PDET KER to perform calculations for peaking up a target into the entrance aperture of PFIS.

PDET CON will send any additional status required back to PDET KER.

PDET CON will also transmit, on request, compressed images (no larger than 0.5 Mbyte) to PDET KER for onward transmission to PCONDIP (Is this required?).

### 9.1.8 Communication With PDET PCI

PDET CON shall communicate with PDET PCI in order to verify that the interface is functioning properly.

### 9.1.9 Communication With PDET SDSU

PDET CON shall communicate with PDET SDSU in order to effect all hardware control via bi-directional communication. Control & status, and CCD Read out data - data rate (on internal optical fibre) up to 250 Mbits/sec (equates to 3.9 Mpixels/second, theoretical maximum speed). Image size up to 25.2 Mpixels, but usually 12.6 Mpixels.

PDET CON shall issue commands to PDET SDSU in order to:

- Request controller status
- Download CCD-specific control/executable code
- Set pre-bin factor
- Set window/s if any
- Set gain factor & readout speed
- Start/stop exposure
- Open/close shutter
- Vertical operations including shifting the image area to the store area
- Read out CCD data
- Various test functions such as charge pumping, noise testing etc.
9.2 PDET KER

9.2.1 Functional Requirements: PDET KER

PDET KER will primarily be used to relay information between the software items running in the PDET PC, and those running in other computers, as mentioned earlier and illustrated in Fig. 1. This information comprises commands and associated parameters to mediate PDET MMI/PCONDI control, update the detector parameters (items 1-8 at the start of this Section), support peak-ups if necessary, initiate the acquisition, readout and storage of images/spectra etc.

9.2.2 Communication With PDET MMI

PDET KER will exchange data (commands and associated parameters) with the PDET MMI. Data exchange shall be via the Data Socket Server in the PCON machine (or via an alternative, mutually agreed communication protocol).

If PDET MMI is not in MMI control, it may receive a command from PDET KER (when in maintenance mode) to assume MMI control.

If PDET MMI is in MMI control, PDET KER may receive commands from PDET MMI in order to:

- Return MMI control to PCONDI
- Update PDET hardware parameters (items 1-8 at the start of this section). This request will be forwarded to PDET CON. The outcome of the request will be returned to PDET MMI. “Program” definition is included here.
- Initiate/Abort the next/current program or exposure. This request will be forwarded to PDET CON. Any vertical operations are requested similarly. The outcome of the request will be returned to PDET MMI.
- Enable interaction with the PDET PC Monitor: place marks/boxes/windows on the display, select objects on the display, overlay templates on the display etc. The outcome of the request will be via the PDET PC display.
- Control the storage of data in FITS format. Reporting the outcome of the command is also required.
- Carry out simple image processing algorithms:
  i. Extracting spectra
  ii. A read out of pixel position of a mouse-type pointer.
  iii. Fitting of Gaussian functions to point sources in imaging mode.
  iv. 2-d line plots of image brightness in any direction across the image.
  v. Simple aperture photometry to estimate magnitudes of point source objects.
  vi. Simple image processing function such as background subtraction, smoothing or median filtering.
  
  Results of these operations will be reported either directly to PDET MMI or via the PDET PC display.
- Receive commands from PDET KER to perform calculations for peaking up a target into the entrance aperture of PFIS.

PDET KER will transmit any additional status required back to PDET MMI.
PDET KER will also transmit, on request, compressed images (no larger than 0.5 Mbytes) to PDET MMI.

9.2.3 Communication With PCONDI

PDET KER will exchange data (commands and associated parameters) with the PCONDI. Data exchange shall be via the Data Socket Server in the PCON machine (or via an alternative, mutually agreed communication protocol).

If PCONDI is not in control, it may receive a command from PDET KER (when in maintenance mode) to assume MMI control.

If PCONDI is in control, PDET KER may receive commands from PCONDI in order to:

- Return control to PDET MMI
- Update PDET hardware parameters (items 1-8 at the start of this section). This request will be forwarded to PDET CON. The outcome of the request will be returned to PDET MMI. “Program” definition is included here.
- Initiate/Abort the next/current program or exposure. This request will be forwarded to PDET CON. Any vertical operations are requested similarly. The outcome of the request will be returned to PCONDI.
- Enable interaction with the PDET PC Monitor: place marks/boxes/windows on the display, select objects on the display, overlay templates on the display etc. The outcome of the request will be via the PDET PC display.
- Control the storage of data in FITS format. Reporting the outcome of the command is also required.
- Carry out simple image processing algorithms:
  - Extracting spectra
  - A read out of pixel position of a mouse-type pointer.
  - Fitting of Gaussian functions to point sources in imaging mode.
  - 2-d line plots of image brightness in any direction across the image.
  - Simple aperture photometry to estimate magnitudes of point source objects.
  - Simple image processing function such as background subtraction, smoothing or median filtering.
  Results of these operations will be reported either directly to PDET MMI or via the PDET PC display.
- Receive commands from PDET KER to perform calculations for peaking up a target into the entrance aperture of PFIS.

PDET KER will transmit any additional status required back to PCONDI.

PDET KER will also transmit, on request, compressed images (no larger than 0.5 Mbytes) to PCONDI.

9.2.4 Communication With The TCS

PDET KER may exchange data with the TCS residing in the TCSS machine. Data exchange shall be via the Data Socket Server in the TCSS machine (or via an alternative, mutually agreed communication protocol).

PDET KER may request information from the TCS:
Telescope pointing data including telescope azimuth, tracker position, hour angle, zenith distance, RA, Dec, Epoch etc.
Target pointing data such as target name, RA, Dec, Epoch etc.
Additional data …. (U of Wisconsin to specify)

PDET KER will supply the TCS with such information as it requires (U of Wisconsin to specify):

- Yadda yadda yadda (SALT to specify)
- Additional yadda yadda (SALT to specify)

### 9.2.5 Communication With The Science Database

At low priority, so as to avoid interference with PDET’s normal operations, PDET KER will transmit one or more images to the Data Processor computer. PDET images may be as large as 25.2 M pixels (of 2 bytes/pixel) but will usually be no larger than 12.6 M pixels. Data rates to be determined by the data reduction requirements.

### 9.2.6 Communication With PDET CON

PDET KER shall exchange data with PDET CON in order to:

- Issue commands to PDET CON and provide update confirmation of changes to PDET hardware parameters (items 1-8 above). “Program” definition is included here.
- Issue commands to PDET CON to initiate/terminate the next program/exposure or terminate the current program/exposure. Any vertical operations are requested similarly. Reporting of the outcome of each command is also required.
- Issue commands to PDET CON to enable interaction with the PDET PC Monitor: place marks/boxes/windows on the display, select objects on the display, overlay templates on the display etc. (Status return is via the appearance of the display).
- Issue commands to PDET CON about the storage of data in FITS format. Reporting the outcome of the command is also required.
- Issue commands to PDET CON in order to carry out simple image processing algorithms:
  i. A read out of pixel position, RA and Dec of a mouse-type pointer.
  ii. Fitting of Gaussian functions to point sources.
  iii. 2-d line plots of image brightness in any direction across the image.
  iv. Simple aperture photometry to estimate magnitudes of objects.
  v. Simple image processing function such as background subtraction, smoothing or median filtering.

Results of these operations will be reported either directly to PDET KER or via the PDET PC display.
- Issue commands to PDET CON to perform calculations for peaking up a target in the entrance aperture of PFIS.

PDET CON will send any additional status required back to PDET KER.

PDET CON will also transmit, on request, compressed images (no larger than 0.5 Mbytes) to PDET KER.
9.3 PDET MMI

9.3.1 Functional Requirements: PDET MMI

The PDET MMI will be used to provide a user interface to control the detector. It is expected that very similar functionality shall reside with PCONDI but not via an MMI in that case.

In normal operational mode, and upon initiation of all software units, it is assumed that MMI control will reside with PCONDI.

If PDET MMI is/is not in MMI control, PDET MMI may issue/receive a command to/from PDET KER to transfer/receive MMI control/control to/from PCONDI.

If PDET MMI is in MMI control, then it will communicate with PDET KER in order to:

- Update PDET hardware parameters (items 1-8 above). This request will be forwarded to PDET CON. The outcome of the request will be returned to PDET MMI. “Program” definition is included here.
- Initiate/Abort the next/current program or exposure. This request will be forwarded to PDET CON. Any vertical operations are requested similarly. The outcome of the request will be returned to PDET MMI.
- Enable interaction with the PDET PC Monitor: place marks/boxes/windows on the display, select objects on the display, overlay templates on the display etc. The outcome of the request will be via the PDET PC display.
- Control the storage of data in FITS format. Reporting the outcome of the command is also required.
- Carry out simple image processing algorithms:
  i. Extracting spectra.
  ii. A read out of pixel position, RA and Dec of a mouse-type pointer.
  iii. Fitting of Gaussian functions to point sources.
  iv. 2-d line plots of image brightness in any direction across the image.
  v. Simple aperture photometry to estimate magnitudes of objects.
  vi. Simple image processing function such as background subtraction, smoothing or median filtering.

  Results of these operations will be reported either directly to PDET MMI or via the PDET PC display.

- Support of peak-up of a target in the entrance aperture of PFIS.

PDET MMI will solicit any additional status required from PDET KER.

9.4 PDET SDSU

9.4.1 Functional Requirements: PDET SDSU

The SDSU controller software shall:

- Respond to commands from PDET CON and reply with status and data as appropriate.
- Control the CCD detector clock and bias voltages correctly and safely at all times.
• Maintain the CCD detectors at a constant temperature (within 0.5°C P-V).
• Control the PDET shutter unit.
• Start/stop exposures, controlling the exposure times to <10 msec accuracy in shuttered mode, and 1 msec accuracy in frame transfer or slot mode.
• Perform vertical transfers when requested.
• Read out the CCD detectors and transmit the digital data to CON. Readout to be accomplished at a range of specified speeds (pix/sec), gain factors, prebin/window formats and modes as shown in the table below. Readout speed jitter must be kept to a minimum during image readout.
• Provide functionality for testing such as charge pumping, etc.
• Transmit messages between CON and the sub-systems controller.

<table>
<thead>
<tr>
<th>Readout Mode</th>
<th>Readout Speed</th>
<th>Prebin</th>
<th>Window</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full frame</td>
<td>Fast or slow</td>
<td>X by Y</td>
<td>Yes</td>
<td>Exposure defined by shutter</td>
</tr>
<tr>
<td>Frame transfer</td>
<td>Fast</td>
<td>X by Y</td>
<td>Yes</td>
<td>Exposure defined by frame transfer.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Frame transfer mask in FT position</td>
</tr>
<tr>
<td>Slot mode</td>
<td>Fast</td>
<td>X by Y</td>
<td>Yes</td>
<td>Exposure defined by frame transfer.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Frame transfer mask in slot position</td>
</tr>
<tr>
<td>Driftscan</td>
<td>Fast or slow</td>
<td>X</td>
<td>No</td>
<td>Exposure defined by drift rate &amp;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>length of CCD (61.5 mm). CCD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>must be aligned with edge (4102 pix) in E-W direction.</td>
</tr>
</tbody>
</table>

### 9.5 Sub-Systems Controller

Auxiliary subsystems excluding the shutter will be controlled by an SAAO designed PIC microcontroller-based unit. The shutter assembly will be controlled directly by the SDSU controller via the sub-systems controller to maintain better control of exposure timing.

The Sub-Systems controller shall:

• Respond to commands received from PDET CON and return status
• Perform all detector control functions required such as ion pump on/off control
• Monitor and control temperatures as required
10 Technical Requirements

10.1 Software Architecture

Fig. 1 shows the major software blocks of PDET along with the external and internal interfaces. Fig. 1 also shows the languages to be used in the different software blocks.

10.2 Software Interfaces

Table 1 provides details of each of the interface shown in Fig. 1. This data indicated here is for information only. The most current data will be identified in the SALT Data ICD, referenced in Section 2.

Table 1: PDET Software Interfaces

<table>
<thead>
<tr>
<th>Number</th>
<th>Interfacing applications</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,2</td>
<td>PDET CON ↔ PDET SDSU</td>
<td>See Section 9.1.9.</td>
</tr>
<tr>
<td>3</td>
<td>PDET CON → Display</td>
<td>See Section 9.1.3.</td>
</tr>
<tr>
<td>4,5</td>
<td>PDET CON ↔ PDET KER</td>
<td>See Section 9.1.7.</td>
</tr>
<tr>
<td>6,7</td>
<td>PDET KER ↔ PDET MMI</td>
<td>See Section 9.2.2</td>
</tr>
<tr>
<td>8</td>
<td>PDET KER → D-Base</td>
<td>Transmission of stored image data to the science data base. See Section 9.2.5</td>
</tr>
<tr>
<td>9,10</td>
<td>PDET KER ↔ TCS</td>
<td>PDET communication with the TCS. See Section 9.2.4</td>
</tr>
<tr>
<td>11</td>
<td>PCONDI → PDET KER</td>
<td>Commands and data from PCONDI to PDET KER. See Section 9.2.3</td>
</tr>
<tr>
<td>12</td>
<td>PDET KER → PCONDI</td>
<td>Image data and status from PDET KER to PCONDI. See Section 9.2.3</td>
</tr>
<tr>
<td>13</td>
<td>PDET KER ↔ PDET PCI, SDSU</td>
<td>Commands and data from CON to/from SDSU via PCI. (Section 9.1.8)</td>
</tr>
</tbody>
</table>
10.3 Modes, States and Events

Figure 2. System Modes

Fig. 2 illustrates the different modes of PDET. See Table 2 for a description of these modes. Table 3 describes the mode transition events.
### Table 2: PDET Modes

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
<th>States</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dead</td>
<td>Power to PDET Computer, CCD and Subsystem Controller, and Cryotiger compressor is all switched off.</td>
<td></td>
</tr>
<tr>
<td>Off</td>
<td>Power to PDET Computer is switched off; power to CCD and Subsystem Controller, and Cryotiger compressor is switched on.</td>
<td>CCD Controller, Subsystem Controller and Cryotiger Compressor switched on.</td>
</tr>
<tr>
<td>Standby</td>
<td>Power to PDET Computer, CCD and Subsystem Controller, and Cryotiger Compressor switched on. In this state the PDET Computer is running.</td>
<td>PDET computer switched on. Boot-up scripts have been executed. Ready to initialise.</td>
</tr>
<tr>
<td>Ready</td>
<td>All subsystems powered up and initialised:</td>
<td>Controller code down-loaded. Control entity selected. All software modules now running. System health has been checked.</td>
</tr>
<tr>
<td></td>
<td>• Detector controller active with default setup</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Detectors at set point temperature</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Shutter closed.</td>
<td></td>
</tr>
<tr>
<td>Maintenance Mode</td>
<td>Full detector parameters available for MMI.</td>
<td>Under SA control.</td>
</tr>
<tr>
<td>Program Mode</td>
<td>Executing a pre-defined sequence of exposures.</td>
<td>Can be aborted.</td>
</tr>
<tr>
<td>Image Processing</td>
<td>Post-processing a previously obtained image: e.g. adjusting its display or overlaying markers on it; extracting spectra, displaying templates</td>
<td>Under SO control.</td>
</tr>
<tr>
<td>Mode</td>
<td>Detector parameters being updated by operator interaction with MMI.</td>
<td>See those listed in Section 9: “Functional Requirements for PDET CON.</td>
</tr>
<tr>
<td>Error</td>
<td>Any errors, which affect the capability of the detector to perform as designed, will place PDET in this mode. Sensor readings and status reporting will continue as far as possible. Error reporting must be sufficient to guide the astronomer to the source of the problem.</td>
<td></td>
</tr>
</tbody>
</table>
### Table 3: PDET Mode Transitions

<table>
<thead>
<tr>
<th>EVENT</th>
<th>From Mode</th>
<th>To Mode</th>
<th>SENSOR/INPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>DEAD OFF</td>
<td>OFF DEAD</td>
<td>Button – Cryotiger Compressor Button – CCD Controller Button – Subsystem Controller</td>
</tr>
<tr>
<td>1</td>
<td>OFF STANDBY</td>
<td>STANDBY OFF</td>
<td>Button – PDET Computer Button – PDET Computer</td>
</tr>
<tr>
<td>2</td>
<td>STANDBY</td>
<td>READY STANDBY</td>
<td>On successful power up and auto/operator initialisation sequence Software/operator initiated shutdown sequence</td>
</tr>
<tr>
<td>3</td>
<td>READY STANDBY</td>
<td>ERROR READY</td>
<td>Error conditions. For example: Problem with subsystems: e.g. Shutter. Problem with detector temperature Problem with controller over-temperature Problem with fibre data comms. Operator initiated – on detecting problems</td>
</tr>
<tr>
<td>4</td>
<td>READY MAINTENANCE PROGRAM IMAGE PROC.</td>
<td>ERROR MAINTENANCE PROGRAM IMAGE PROC.</td>
<td>Error Recovery – Operator initiated</td>
</tr>
<tr>
<td>5</td>
<td>READY MAINTENANCE PROGRAM IMAGE PROC.</td>
<td>READY MAINTENANCE PROGRAM IMAGE PROC.</td>
<td>Error Recovery – Operator initiated</td>
</tr>
<tr>
<td>6, 13</td>
<td>READY MAINTENANCE PROGRAM IMAGE PROC.</td>
<td>ERROR MAINTENANCE PROGRAM IMAGE PROC.</td>
<td>Error Recovery – Operator initiated</td>
</tr>
<tr>
<td>7</td>
<td>ERROR</td>
<td>READY</td>
<td>Error Recovery – Operator initiated</td>
</tr>
<tr>
<td>8</td>
<td>READY MAINTENANCE</td>
<td>MAINTENANCE PROGRAM</td>
<td>In Maintenance mode: detailed control enabled. Operator initiated.</td>
</tr>
<tr>
<td>9</td>
<td>READY PROGRAM</td>
<td>IMAGE PROCESSING</td>
<td>Performing a pre-defined sequence of images. Operator initiated.</td>
</tr>
<tr>
<td>10</td>
<td>READY IMAGE PROCESSING</td>
<td>IMAGE PROCESSING</td>
<td>Post-processing of a pre-obtained image from Maintenance, Snapshot or Program modes</td>
</tr>
<tr>
<td>11</td>
<td>MAINTENANCE PROGRAM IMAGE PROC.</td>
<td>READY MAINTENANCE PROGRAM IMAGE PROC.</td>
<td>Operator initiated: mode over. Exposure/Program finished/aborted Operator initiated: mode over.</td>
</tr>
<tr>
<td>12</td>
<td>MAINTENANCE PROGRAM IMAGE PROC.</td>
<td>ERROR MAINTENANCE PROGRAM IMAGE PROC.</td>
<td>Error condition during an exposure or post-processing.</td>
</tr>
<tr>
<td>13</td>
<td>ERROR STANDBY</td>
<td>STANDBY</td>
<td>Error conditions not resolved.</td>
</tr>
<tr>
<td>14</td>
<td>READY INPUT → READY</td>
<td>READY</td>
<td>Operator initiated: parameter update. System returns to READY mode.</td>
</tr>
</tbody>
</table>
10.4 Software Capabilities

10.4.1 Communication
a. PDET SW shall communicate with PCONDI and any other SALT subsystems, using the methods described in Fig. 1 and the Functional Requirements sections, as amended by the SALT Data ICD.
b. The variables communicated shall be defined using the cluster data types identified in the SALT Data ICD.
c. The standard “SALT Communication Module” shall be used to perform Labview Data Socket communication, at a rate commensurate with the requirements of the applicable data.

10.4.2 Initialisation
a. The PDET PC and SW shall initialise to a safe, known state. Communication with the SDSU controller and subsystem controller shall be established and all hardware shall be placed in a safe, “ready” mode.
b. The SW shall monitor the status of each subsystem’s initialisation and shall only complete initialisation when all the subsystems are known to be ready, unless this process is aborted via an operator command.
c. No hardware movement or CCD operations shall be allowed without operator intervention.

10.4.3 Command Interpretation and Generation
a. All commands to/from PCONDI, TCS or the Data Processor shall be text-based. These commands will define the required action and appropriate parameters associated with the action.
b. PDET SW shall interpret the commands and change modes and/or initiate the appropriate action, only if appropriate (see c.)
c. PDET SW shall only initiate actions or change modes, if it is appropriate in terms of safety or allowable detector operation. A set of “operating rules” shall define this operation, incorporating the actual states of the detector.
d. Changing modes shall be implemented using a state machine, following a matrix of present states, required states, conditions to be met and actions to be performed.
e. When PDET is placed in maintenance mode, the above requirements can be overridden.
f. It shall be possible to operate the detector in a limited fashion, with some of the subsystems in Maintenance mode.

10.4.4 Status Reporting
a. PDET SW shall report to PCONDI all status information.

10.5 Operating System
PDET SW shall run the Linux or Real Time Linux operating system.

10.6 Resource Allocation
The TCS software shall be capable of performing its function on a PC with the following specifications:
- 2GHz Pentium
- 256MB memory
- 800 x 600 resolution (local MMI)

11 Generic Software Requirements

PDET Software shall comply with the requirements specified in the PDET Software Development Plan and the Labview Coding Standard, as referenced in section 2.
12 Software Testing

A full description of the software testing will be developed during the design phase.