SOFTWARE USERS MANUAL
FOR THE
PERSI-ALICE INSTRUMENT

DECEMBER 2004

SwRI® Project 05310

Document No. 05310.02-PAS_SUM-01

Contract NASW-02008

Prepared by

SOUTHWEST RESEARCH INSTITUTE®
Automation and Data Systems Division
6220 Culebra Road, San Antonio, Texas 78228-0510
(210) 684-5111 • FAX (210) 522-5499

SOFTWARE USERS MANUAL FOR THE PERSI-ALICE INSTRUMENT

SwRI Project 05310

Document No. 05310.02-PAS_SUM-01

Contract NASW-02008

Prepared by:	Maarten Versteeg SwRI Software Project Manager	16 December 2004
Approved by:	Ronnie Killough SwRI Section Manager	30 December 2004
Approved by:	/s/ John Scherrer SwRI Project Manager	6 January 2005
Approved by:	/s/ P. Brian Gupta SwRI Quality Assurance	7 January 2005
Approved by:	/s/ L. D. McCullough SwRI Project CM	7 January 2005

Signatures on file Document Control.

Automation and Data Systems Division
Southwest Research Institute
P. O. Drawer 28510
6220 Culebra Road
San Antonio, Texas 78228-0510
(210) 684-5111

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

REVISION NOTICE

Draft 1: 9 November 2004
Initial Issue: December 2004

1. SCOPE

1.1 Project Identification

Project Title:	PERSI-Alice: Flight and Ground Software
Project Number:	15-05310.02
Software Project Manager:	Maarten Versteeg
Start Date Phase C/D:	2/10/2003
End Phase C/D:	12/15/2004

1.2 Project Overview

SwRI®'s Instrumentation and Space Research Division (Division 15) designed and build a UV (Ultraviolet) spectrometer instrument for flight on the New-Horizons (NH) spacecraft to Pluto/Charon and to the Kuiper Belt. The launch of the New Horizons spacecraft is planned for January 2006. The UV spectrometer is called Alice and is based on a previous UV spectrometer, also called Alice that was built for the Rosetta mission. Where needed, the two instruments will be distinguished by specifying the mission: R-Alice (for Rosetta-Alice;) and P-Alice (for PERSI- or Pluto-Alice).

The general instrument configuration of P-Alice is similar to R-Alice; some changes, however, were made to adapt P-Alice to the specific needs of the NH mission. The main changes include unique spacecraft interfaces, some added redundancy, and the addition of a Solar Occultation Channel (SOC) that involves the addition of a new input aperture designed to view the Sun during the solar occultation phases of the Pluto/Charon and Kuiper Belt flybys.

1.3 Document Overview

This document contains the software users manual for the PERSI-Alice instrument. Although it is in principle a software users manual, it describes many of the instrument operational systems aspects as the software determines many of these. The document focuses on the software referred to as the flight software (PERSI-Alice Flight Software - PAFS) version 1.01 but were needed it will also include some aspects of the PERSI-Alice GSE system and its software (referred to as the ground software: PERSI-Alice Ground Software - PAGS). In some cases, operational issues are mentioned that may originate from the spacecraft or spacecraft GSE level.

Chapter 2 of this document starts with a general description of the function and operations of the PERSI-Alice instrument; this provides a framework for the instrument operations. This chapter also describes some operational aspects and the relations between the different parts of the PERSI-Alice system.

The next chapter (Chapter 3) contains an overview of the instrument operations. This chapter describes the general operations and is intended to provide sufficient context to enable understanding of the more detailed description focusing on details that is provided in the next chapter (Chapter 4). This should make it possible to understand the descriptions given in that chapter for which no order could be found that would allow for sequential reading.

In a number of appendixes, the detailed description of the telemetry is given including the compete description of the low speed telemetry and the telecommands including parameters and operational effects. Also listed in Appendix F are the contents of the internal stored parameter list, the meaning of the stored entries and their initial and hard-coded default values. The last appendix lists the error codes that are used by the software to report problems in a field in the housekeeping telemetry.

1.4 Related Documents

Document ID:	05310-PAS_SDP-01
Originator:	Southwest Research Institute, San Antonio TX
Issue:	Rev 0, Chg. 3, August 19 2002
Title:	Software Development Plan for Pluto-Alice Flight and Software
Applicability:	Describes the processes and procedures used for software development in the development of the PERSI-Alice Flight and Ground software.

Document ID:	05310.02-PAFS_SDD-01
Originator:	Southwest Research Institute, San Antonio TX
Issue:	Rev. 1, Chg. 0, December 10 2004
Title:	Software Description Document PAFS software
Applicability:	Describes as build PERSI-Alice Flight software.

Document ID:	05310.02-PAS_VDD-02
Originator:	Southwest Research Institute, San Antonio TX
Issue:	Rev. 0, Chg. 0, July 2 2004
Title:	Version Description Document for PAFS software version 1.01
Applicability:	Describes as build PERSI-Alice Flight software.

Document ID:	05310-PAS_SRS-01
Originator:	Southwest Research Institute, San Antonio TX
Issue:	Rev. 1, Chg. 0, March 26 2003
Title:	Software Requirements Specification for PAFS and PAGS software
Applicability:	Describes Software Requirements Specification for PERSI-Alice Flight and Ground software.

Document ID:	PAIP-02-15-05310
Originator:	SwRI
Issue:	Revision 1, October 2003
Title:	Performance Assurance Implementation Plan for New Horizon Pluto-Kuiper Belt Mission
Applicability:	Overall PAIP for New Horizon

Document ID:	7399-9046
Originator:	Applied Physics Laboratory
Issue:	Revision -: 6/12/2003
Title:	New Horizons Spacecraft to PERSI-ALICE Interface Control Document
Applicability:	Contains interface description between Spacecraft and Alice Instrument, referred to as [ICD section].

Document ID:	05310.02-ISPEC-01
Originator:	Southwest Research Institute, San Antonio TX
Issue:	November 3rd, 2003)
Title:	Instrument Specification for the Alice Instrument New Horizons Mission
Applicability:	Contains Alice Instrument description

Document ID: 05310.02-CDHSPEC-01
Originator: Southwest Research Institute, San Antonio TX
Issue: Rev. 1 Chg. 0 September 2003
Title: Specification for the PERSI-Alice C&DH Electronics
Applicability: Contains specification of the hardware of the PERSI Alice C&DH

1.5 Acronyms and Abbreviations

Acronym	Explanation
A	Ampere
ACQMEM	Acquisition memory
Alice	Just a name, not an abbreviation
AnodeV	Anode voltage
APID	Application ID
APL	Applied Physics Laboratory
AU	Astronomical Unit ($1.496 \cdot 10^9$ km)
bps	bits per second
C&DH	Command and Data Handling
CCSDS	Consultative Committee for Space Data Systems
CM	Command Message
CRC	Cyclic Redundancy Check
DDL	Double Delay Line
EEPROM	Electrically Erasable PROM
EGSE	Electrical Ground Support Equipment
EM	Electrical Model
FIFO	First In First Out (buffer)
FM	Flight Model
GSE	Ground Support Equipment
GSEOS	Ground Support Equipment – Operating System
HK	Housekeeping (data packet)
HS	High Speed (Telemetry)
HV	High Voltage
HVPS	High Voltage Power Supply
HW	Hardware
Hz	Hertz – 1/second
I&T	Integration and Test
I/F	Interface
ICD	Interface Control Document
ISR	Interrupt Service Procedure
ITF	Instrument Transfer Frame

Acronym	Explanation
kbps	kilo bits per second (= 1000 bps)
kHz	kilo Hertz = 1000 Hertz
kV	kilo Volt = 1000 Volt
LAT	Limited Angle Torque (motor)
LS	Low Speed (Telemetry)
LVDS	Low Voltage Differential Signal (high speed interface specification)
LVPS	Low Voltage Power Supply
Mbps	Mega bits per second
MCP	Micro Channel Plate (detector)
McpV	MCP voltage
MET	Mission Elapsed Time
ms	milli second
NH	New Horizons
OAP	Off-Axis Parabolic mirror
PAFS	PERSI Alice Flight Software
PAGS	PERSI Alice Ground Software
P-Alice	PERSI-Alice to distinguish it from Rosetta-Alice (R-Alice)
PAS	PERSI Alice Software / System
PC	Personal Computer
PERSI	Pluto Exploration Remote Sensing Investigation
PHD	Pulse Height Distribution
PPS	Pulse Per Second
PROM	Programmable Read Only Memory
RAM	Random Access Memory
s, sec	seconds
S/C	Spacecraft
SMA	Shape Metal Alloy (actuator)
SMM	State Machine Mode (backup acquistion mode)
SOC	Solar Occultation Channel
SRS	Software Requirements Specification
SSR	Solid State Recorder (spacecraft data storage)
STB	Software Test Bed
StripI	strip current
SW	Software
SwRI	Southwest Research Institute
TC	TeleCommand
TiNi	Brand name of used shape metal alloy (SMA) actuators

Acronym	Explanation
TM	Telemetry
TSOC	Tombaugh Science Operations Center
μA	Micro Ampere = 10^{-6} Ampere
UART	Universal Asynchroneous Receiver / Transmitter
UV	Ultraviolet
WPA	Wax Pellet Actuator
XOR	Exclusive OR (logical operation)

1.6 Notation Conventions

Wherever a specific reference is made to a Telecommand, a field in the Housekeeping packet (see Appendix C), or a field in the Parameter file (see Appendix F), the parameter name will be preceded by respectively 'Tc_', 'Hk_' or 'P_'. Error codes are indicated by their name and are preceded by lowercase letters 'ec', all error codes are listed in Appendix G.

2. INSTRUMENT SYSTEM OVERVIEW

This chapter gives and overview of the PERSI-Alice instrument including operations and describes the different software modules within the P-Alice instrument system and their relations. The P-Alice instrument is part of the PERSI investigation, but has an independent electrical and mechanical interface with the New Horizons spacecraft.

To provide a context for the description of the software and the interfaces, this chapter starts with an overview of the Alice Instrument. The P-Alice spectrometer consists of an optical system and the controlling/power electronic hardware. To provide an overview of the spectrometer functions, the operational modes and the produced science data are briefly described. Next, an overview of the instrument electrical hardware is provided, as this is needed to understand the functions that the software has to perform. Finally, the interfaces of the flight software to the different hardware components are described.

2.1 PERSI-Alice Instrument overview

PERSI-Alice is an UV spectrometer that is sensitive to Ultraviolet (UV) light in the range of 520-1870 Å. The instrument consists of a telescope section with an off-axis parabolic mirror, and a spectrograph section that includes a diffraction grating and a microchannel plate (MCP) detector. The MCP detector is an electro-optical device sensitive to extreme and far ultraviolet light. It consists of a photo cathode-coated MCP surface that converts UV photons to electrons, an MCP Z-stack configuration of three MCPs to amplify the signal, and a two-dimensional double delay-line (DDL) anode readout array. The first (x) dimension provides the spectral location of the detected photon and the second (y) dimension provides one-dimensional spatial information. The DDL detector system outputs to the command-and-data-handling (C&DH) electronics, pixel location for each detected photon event consisting of a spectral and spatial coordinate. The events are processed by the C&DH electronics. The C&DH is also the controller of Alice; it receives commands from the spacecraft, acquires data from the MCP detector system, and returns telemetry to the spacecraft. Science data telemetry generation is performed by the detector hardware but the C&DH also controls this function. Alice has two acquisition modes (see below) in which the spectral/spatial data from the detector is processed by the C&DH subsystem.

All following descriptions assume a nominal operating instrument. The instrument hardware also provides a basic, hardware controlled, default pixel list acquisition mode, which is activated when the instrument control hardware detects (multiple) successive (software) failures. This mode is called the ‘hardware limp-along’ mode; for a description of this mode, the reader is referred to the instrument C&DH hardware description. Once this mode is activated any software control of the acquisition operation is disabled, until the instrument is reset (power cycle or S/C reset).

2.1.1 Science Operations

2.1.1.1 Detector and Detector Electronics

Alice is not an imager in the traditional sense. It is a Rowland-circle spectrograph that uses state-of-the-art MCP detector and UV optics technology. The telescope section focuses light entering one of the two entrance apertures (the airglow or SOC) onto the entrance slit of the spectrograph section. Light that passes through the slit strikes a concave holographic diffraction grating, which focuses the Alice, passband wavelengths onto the MCP detector. Photoelectrons are created by the photo cathode material on the front surface of the MCP Z-Stack and are multiplied by the Z-Stack to create an amplified charge cloud of ~10^7 electrons/event that is accelerated across a narrow gap to the DDL 2-D anode array. When this charge cloud exceeds a specified amplitude level set by the detector threshold, it will result in a valid detected event. In response to the event, the anode outputs signals are converted by the detector electronics into a pixel location on the array. Thus, the use of the word 'pixel' refers to the resolved location where the charge cloud hits the anode. The detector electronics encodes 1024 spectral pixel columns and 32 spatial pixels. For each event, the detector also reports the amplitude of the charge pulse as a six-bit number that can be histogrammed into a pulse-height distribution by the C&DH electronics. The field-of-view (FOV) of the instrument is such that six of the spatial rows (3 top/3 bottom) are not in the active field of view of the instrument and are used to store the pulse-height data along with two detector electronic stimulation pixels, both of which are used for electronic diagnostics and detector health monitoring.

The Alice instrument can process the detected pixel events in two different primary modes of operation – histogram and pixel list which are described below. In both acquisition modes, the hardware provides a double buffer; one buffer is used for the current data acquisition, and the data in the other buffer is sent to the spacecraft. This makes it possible to perform continuous acquisitions. In both cases, science data transfer from the instrument to the spacecraft will occur at a burst rate of 1 Mbps. The block size of the transfers is 32 kword, resulting in transfer duration of about ½ second.

2.1.1.2 Histogram Acquisition Mode

In histogram acquisition mode, the detector counts the number of events that are occurring at each of the 1024*32 = 32,768 (32 k) "locations" provided by the detector electronics over a period of time (called an exposure). Each of the histogram counters consists of a 16-bit number; hence for each location a maximum number of 65,535 events can be counted. When this maximum number is reached during an exposure, the counter is no longer incremented and the corresponding pixel becomes saturated. By operationally selecting suitable exposure durations, (excessive) saturation may be prevented. In addition to the spectral/spatial histogram, a pulse height histogram based on the pulse height information that is provided with each detected event is acquired. A small area in the bottom rows (outside the active field of view) is reserved and 64 locations are used to store the counters for this histogram. This pulse height histogram is referred to as the pulse height distribution, and is useful for instantaneous health monitoring of the detector MCP gain characteristics. These counters are also 16 bits deep and, thus, also provide for a maximum count of 65,535 events per bin. When an exposure is completed, a single read out results, in which each of the 32 k histogram locations is represented as a 16-bit number.

2.1.1.3 Pixel List Acquisition Mode

In pixel list acquisition mode, the instrument stores event data as it arrives from the detector system. Each detected event is stored in a consecutive word (16 bits) in the acquisition memory. In this mode, the detector memory is used as a 32,768 locations deep list that can store 32,768 separate events. To indicate event information, the top bit of the word is cleared and the remaining 15 bits contain the event information (spectral and spatial location). In order to provide timing information, "time hacks" are also stored in this acquisition memory. These time-hacks are inserted at a programmable interval between 4 and 512 ms, and also take up one word in the detector memory. For the time hacks the top bit is set and the lower 15 bits provide a counter (fixed 4 ms rate) value. During a pixel list acquisition, the pulse height data is discarded. When the acquisition memory is full or the acquisition is stopped, the buffer contents are read out and a block of 32 k words is sent to the spacecraft. The filling rate of the acquisition memory depends on the actual event rate and the selected time hack interval.

2.1.1.4 Science data format and masking operation

The PAFS does not have direct access to the detector event data; this data is handled by the acquisition hardware and stored in the acquisition memory according to the selected acquisition mode. Each detector event consists of an event location specified as a spectral (10-bits resolution) and a spatial (5-bits resolution) coordinate in addition to this the 'pulse height' of an event is determined (6-bits resolution). The acquisition mode determines how this data is stored in the acquisition memory:

- In pixel list mode, the acquisition memory contains addresses of pixel events in sequence as they occur. In Pixel List mode there is also a time mark (a special value) inserted into the sequence on a periodic basis in order to provide event timing information.
- In histogram mode, there is a mapping between locations in the memory array and pixel locations on the detector. Each location is associated with one pixel, and contains the total number of event counts for that pixel that has accumulated within the integration period. A small area of the memory (64 entries) is set aside to count the number of pulse heights for each intensity, resulting in a pulse height distribution.

Before any event is processed, a 'hot-pixel' filtering operation is performed. This filtering operation allows for the exclusion of events from certain areas of the detector array. Eight user definable filter masks are provided, and one fixed filter mask is defined that masks the lower left corner of the detector (this area contains the header word and in histogram mode the pulse height distribution data.) The hardware provided masks consists of rectangular areas with a size of 4 spatial (y) rows and 32 spectral (x) columns. Events that are registered within this area are excluded from any further processing, but these events are included in the measured (raw) event countrate. The location of each of the 8 user definable masks is specified using a single byte. The top 3 bits of each mask specification define the spatial location of the mask and the bottom 5 bits specify the spectral location. This means that the rectangular filters can only be located at multiples of their respective spatial (y) and spectral (x) sizes. The filter area starts in the specified row and column and run up to (but not including) the next possible filter start row and column.

Dimension	Data range	Hot Segment Filter/Mask			
		specification byte	start location	possible start location	end location
Spatial (y)	**0-31**	**top 3 bits**	**4*y**	**0,4,8, … 28**	**4*y+3**
Spectral (x)	**0-1023**	**lower 5 bits**	**32*x**	**0,32,64, … 992**	**32*x+31**

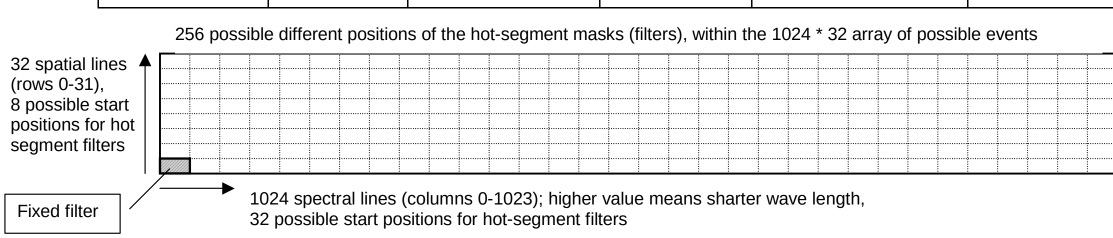

Figure 1: Hot pixel mask (filter) locations.

Upon system (hardware) reset, all user definable filter specifications are reset to zero, so the location of the user definable filter areas correspond to the fixed filter in the lower left corner of the event array.

2.2 Software Test and Communication Configurations

The complete software suite for the Pluto Alice project consists of two parts. One of these is included in the P-Alice instrument, and the other one is located in the P-Alice GSE engineering workstation.

During the P-Alice development and flight, different communication configurations have to be supported. Final configuration will be the flight configuration in which the instrument is mounted on the spacecraft, and the spacecraft provides a TC/TM communication with the ground system. During the Pluto flyby, the spacecraft will be at a distance of about 32 AU from the Earth resulting in a one-way communication time of more than 4 hours. This means that during the flyby, the onboard system has to operate largely in an autonomous mode.

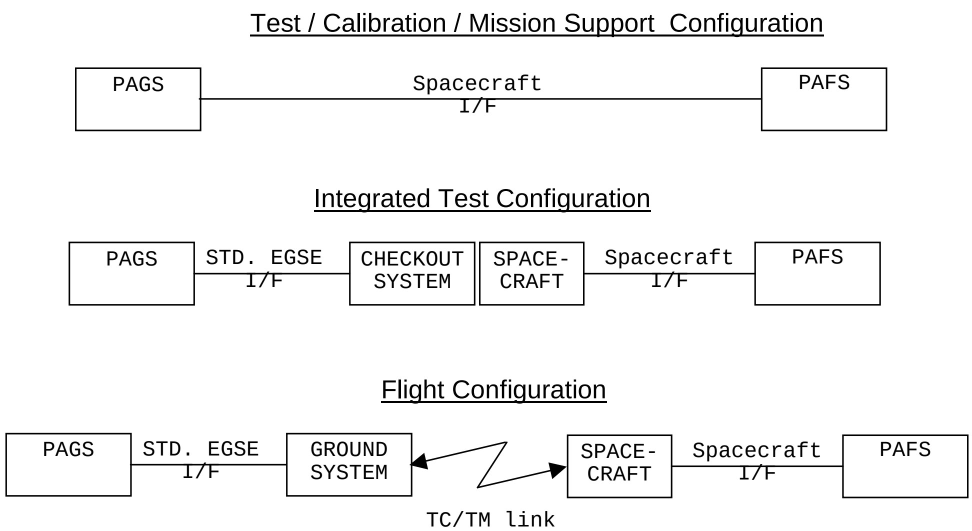

Figure 2: Test and communication configurations for the P-Alice system

During development, test and integration at the various levels, the communication between the Alice ground system and P-Alice will use various hardware configurations as shown in Figure 2. These different configurations support the different test and integration activities.

In addition to the available hardware platforms, a software simulator of the processor (Keil51 simulator) is available to support the mission. This simulator includes capabilities to functionally simulate the hardware that directly interfaces with the processor; this provides a basic environment for the execution of the flight software. This simulator environment will be maintained as it provides a platform to support tests and it provides a very detailed visibility of the internal states of the processor. This simulator together with a software breadboard system (Electrical Model (EM)) will form the basis of the mission support system.

2.3 Hardware Operational Environment

2.3.1 Processing Hardware

Following is a summary of the hardware environment in which the Alice Flight Software executes. Refer to Appendix A for an overview and the Alice C&DH specification for details about Alice hardware.

Here are the primary features of the PAFS execution environment:

- 8051 compatible micro controller, 4.000 MHz clock,
- 32 kilobytes non-volatile memory: primary program memory (PROM),
- 4*32 kilobytes non-volatile memory: patchable program and tables memory (EEPROM),
- 32 kilobytes volatile memory: general data memory (RAM),
- Circuitry for receiving telecommands (redundant TC interface), with external interrupt,
- Circuitry for sending telemetry (redundant low rate TM interface), with external interrupt,
- Circuitry for real-time timekeeping; including periodic and timesync interrupt,
- Science data acquisition memory (2* 64 kbyte(used): ACQMEM) and control circuitry, also generating the high rate science TM packets,
- Detector interface (including event counter, STIM etc.),
- Watchdog timer,
- Circuitry for acquiring analog housekeeping (status) measurements,
- Circuitry for controlling the system actuators (including heaters).

2.3.2 Instrument Internal interfaces

An overview of the interfaces of the P-Alice flight software is presented in the software context diagram shown in Figure 3.

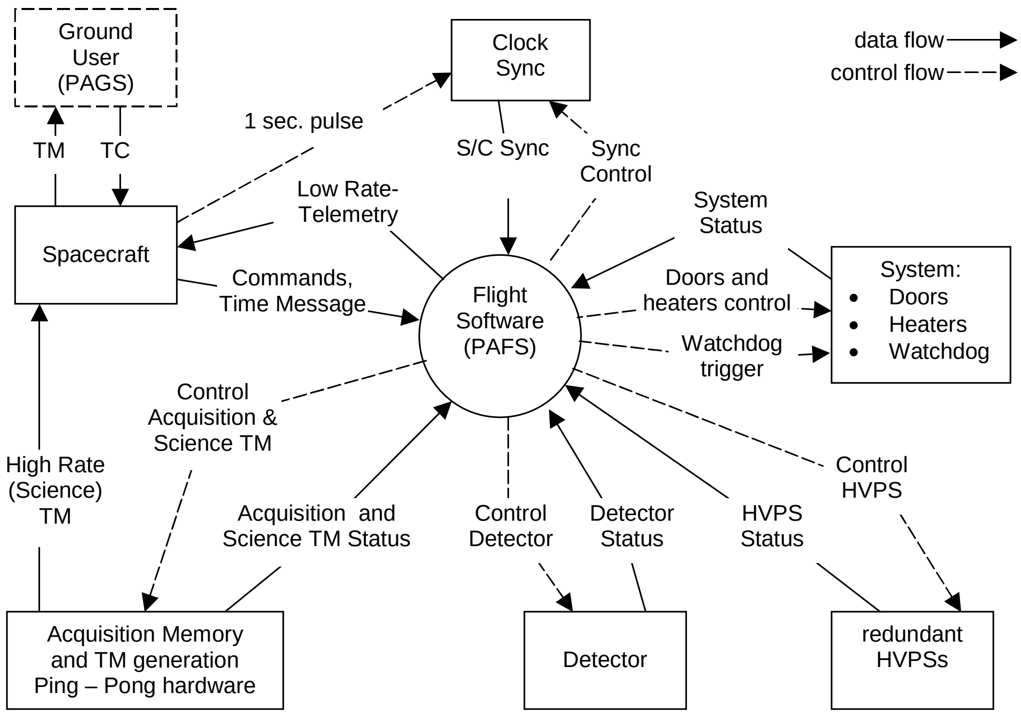

Figure 3: P-Alice Flight Software Context diagram

2.3.3 Safety Configurations

For use during ground testing there are two redundant hardware safing mechanisms included in the Alice instrument. These safing mechanisms define an operational mode, which is configured by means of an external instrument red-tag safety plug. This Actuator Safety plug will allow for separate enabling of the primary and redundant sides:

- High Voltage safing plug (HV/20) (separate for primary and redundant HVPSs)
- Actuator safing plug (separate for primary and redundant sides)

The four safing status signals are read by the PAFS to be reported in the HK data stream to allow verification during ground testing. This status is only reported; it does not influence any (on-board) software control function.

2.3.3.1 HV/20 safety configuration

The P-Alice instrument has a hardware selected safe operational mode in which the HV activation is limited to a HV value that allows safe operation of the MCP detector at ambient pressure. This prevents any damage to the instrument during integration and test caused by unintentional activation of the HVPS (HV activation to voltage levels exceeding 1 kVolt at room pressure can cause arcing of the detector). This operational mode is configured by means of an external instrument red-tag safety plug, which provides a separate control of either of the redundant HVPSs.

This HV/20 mode reduces the full-scale command signal for the HVPS. This H/20 mode does not affect the HV read back values, so these will read back at the corresponding limited range values (i.e., read back values correspond to the actual voltage, not the value/20). This means that the HV safety checking cannot be activated in the HV/20 mode, as the read backs will show the limited values:

- Mcp Voltage — read back of the scaled down value, so the check verifying commanded versus read-back HV will fail,
- Strip Current — linear with Mcp Voltage results in read back of scaled down value,
- Anode Voltage — has a nominal operational value of about 600 Volt, at the reduced control value of the HVPS the Anode Voltage regulation will not yet be active, so a reduced value will be read**.**

A consequence of these differences is that when the HV/20 safety plug is in place nominal systems operations are not possible. The 'Mcp Voltage' and the 'Anode Voltage' safety mechanisms will detect non-nominal conditions and trigger the safety mechanism. To operate P-Alice (for instance to verify commanding sequences), the “Mcp Voltage” and the 'Anode Voltage' safety checks must be disabled.

2.3.3.2 Door actuator safety configuration

The P-Alice instrument has a hardware selected (configuration plug) safe configuration in which one time door and valve actuators physically cannot be activated. A separate safing for primary and redundant actuators is available. This prevents these one-time mechanisms from being activated unintentionally during integration and test. Note that this safety configuration does not affect the aperture door or heaters from being operated (the aperture door will need to be unlocked).

2.4 Instrument External (spacecraft) interfaces

Although the P-Alice instrument is a part of the PERSI instrument package, it has seven separate independent redundant interfaces directly to the spacecraft.

- Instrument power interface
- Actuator power interface
- Time synchronization interface
- Low speed serial telecommand interface (TC)
- Low speed serial telemetry interface (TM)
- High speed science telemetry (HS)
- Instrument reset interface

2.5 GSE system overview

The P-Alice Ground System consists of two PCs: one executes the PAGS and the second one acquires the high-speed science data. The two PCs are interfaced through a TCP/IP network interface. The GSEOS PC is the main GSE PC that provides a user/operator interface and controls the GSE operations. This system will fetch the acquired science data from the second PC using the network interface and present the complete results. The GSEOS PC has both the needed hardware interfaces to interface with the P-Alice instrument directly or via the general spacecraft ground support system. The PAGS will be developed using the GSEOS framework.

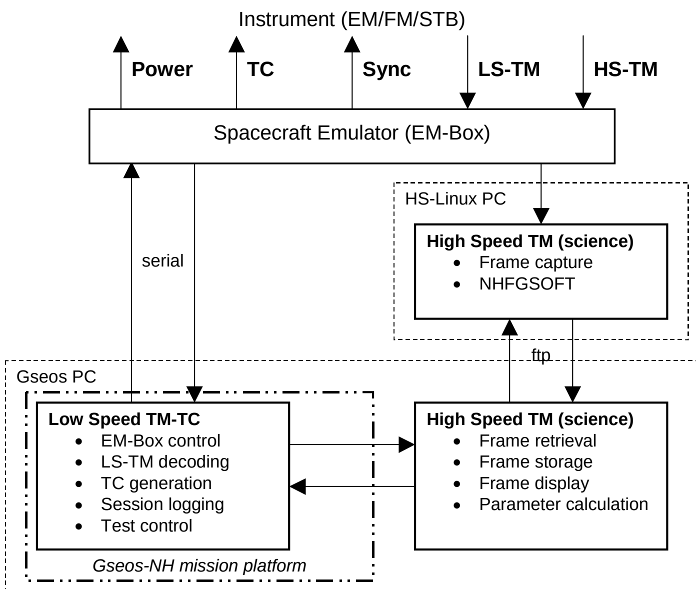

Figure 4: Overview Alice GSE system (Instrument Testing phase)

After the instrument is mounted on the spacecraft, the data streams will be routed through the spacecraft data handling system to the spacecraft ground system. The Spacecraft ground system makes the instrument data available to the GSE via a standard EGSE interface using a Tcp/Ip based protocol. The instrument GSE can tap into this data stream and access and present the PERSI-Alice instrument telemetry. In that configuration, the GSE no longer needs the HS interface PC nor the dedicated hardware interface.

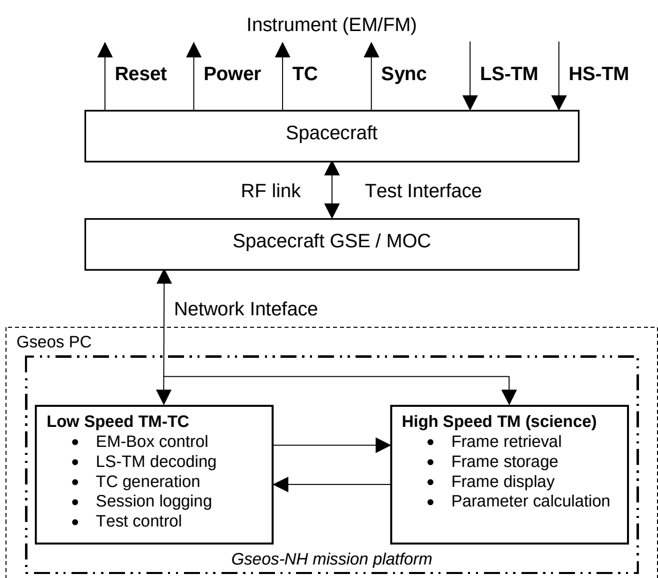

Figure 5: Overview Alice GSE system (Spacecraft Testing/Operations phase)

3. INSTRUMENT OPERATIONS OVERVIEW

3.1 Instrument Startup

Upon power on or software reset, the software starts and performs a code check. Instrument power may be applied either to instrument power bus A, bus B, or both. The code check determines which code will be started, either PROM or one of the four EEPROM pages. If no problems are detected in the code check of the PROM, execution initially starts in PROM. After completion of this code check and clearing of the memory, the various software modules are initialized; this basic software initialization includes:

- Enter safe state (deactivate any actuator operation)
- Pre-set parameter table with hard coded (nominal operational) values
- Parameter table initialization (with majority vote based values from EEPROM)
- Set discriminator to default value (based on parameter table)
- Start the aperture closing operation (if enabled and actuator power available)

After this, the nominal operations of the system start and the software starts generation of the telemetry data stream at a rate of one packet per second. The first telemetry packet will appear between 2 and 3 seconds after startup depending on whether an external sync pulse (1PPS) is detected during the system startup.

The Alice instrument can be in one of four states (including powered off) at any time. Three of these states are actually 'operational' states. Initially, after power up or a reset, the instrument will always enter the Safe state. In the safe state the instrument will always be idle, the HVPS, heaters and actuators will all be deactivated.

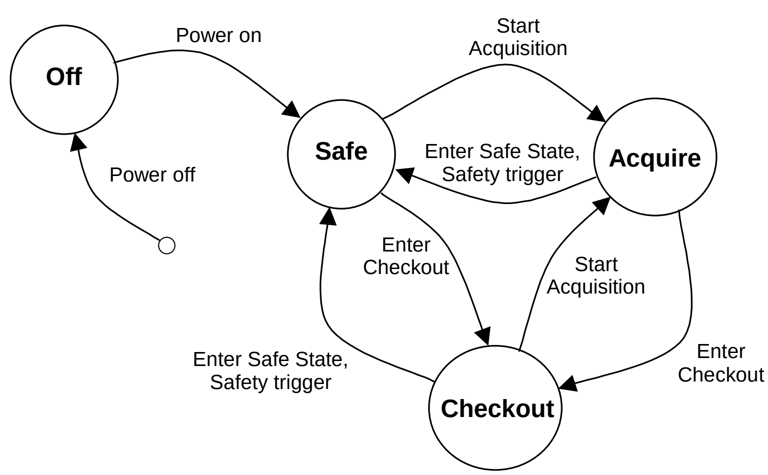

Figure 6: Operational states, state diagram

Changes between the states can either be the result of received telecommands or autonomous by the instrument due to an unmasked safety trigger. The current state of the instrument is always reported in the housekeeping telemetry.

3.2 Telemetry Generation

Every second after startup in nominal situation, the instrument will generate a single low speed telemetry packet. The gathering of the data for the telemetry packet starts immediately after the instrument receives an external sync pulse (1PPS). If no 1PPS pulse is received by the instrument, the instrument will continue to generate a single telemetry packet every second, now based on the internal instrument clock. The gathering of the information either retrieves information from the hardware directly (where possible) or in some cases retrieves most recent acquired data (for measurements taking more time or periodic measurements). This means that the reported housekeeping data may be up to 1 second old.

In each telemetry packet, the instrument reports the instrument concept of Mission Elapsed Time (MET). This time is represented as a 32-bit integer number specifying the number of seconds that have passed since some epoch. A 32-bit integer number allows the specification of a period of $2^{32} = 4294967296$ seconds (about 136 years). Normally this instrument time will be synchronized to the spacecraft time and allow for correlation of the instrument reported status with spacecraft events.

The generated telemetry packet is transferred to the spacecraft in an ITF (Instrument Transfer Frame) format. The generated ITF always contains non-packetized instrument state information followed by a complete CCSDS packet containing the instrument housekeeping (APID 482). The spacecraft can use the non-packetized data for control purposes (autonomy) and also will pack this data (together with other sub-system data) into a separate spacecraft telemetry packet (APID 008). In addition to this, the P-Alice generated ITF may contain a memory dump packet. This memory dump packet is again a CCSDS formatted TM packet (APID 481). The instrument housekeeping packet contains the complete instrument state description including the operation mode. A single packet was defined that includes both the operational and the diagnostic information needed during normal operations of the instrument. Some additional diagnostic data may be obtained by directly dumping selected instrument memory areas using the available memory dump command.

The non-packetized data consists of a sub-set of the housekeeping telemetry. So, when the complete instrument housekeeping packet is available, there is no need for the non-packetized instrument telemetry. The spacecraft though may not (immediately) relay all instrument generated telemetry data to the ground, or, in some cases, the spacecraft may sub-sample specific telemetry streams so that the different packets are transmitted to ground at different rates. In these cases, the non-packetized (critical) instrument housekeeping data may be the only data available.

In addition to this low speed telemetry, the instrument uses a high-speed interface to transfer science data from the instrument to the spacecraft. This transfer consists of raw science frames which are always 32k word (16-bit words = 65536 bytes) in the P-Alice case. These packets will be stored in the spacecraft Solid State Recorder (SSR) memory, and, before transfer to the ground, these packets will be reformatted into CCSDS transfer frames (APID 4B1 and 4B3 for pixellist respective histogram science data). High-speed science data frames are only produced during or at the end of commanded acquisition operations. Each generated science frame includes a sequence number counting the number of frames generated since the last instrument startup; this number is also included in the housekeeping telemetry stream allowing for correlation between the generated science data and the housekeeping data describing the instrument conditions during the acquisition.

3.3 Telecommanding

The instrument operations are commanded using a set of 25 separate telecommands. The instrument can process up to one command per second. The acceptance and completion status of the command execution is reported in the housekeeping data. The instrument verifies incoming telecommands before they can be executed; this basic verification includes a format and a checksum check of the telecommand. The instrument has two redundant telecommand interfaces, but, within any given one second timeframe, the instrument will only accept telecommands from a single interface. Based on data received from both interfaces and the status of the also redundant 1PPS synchronization signal, the instrument decides which telemetry interface to use.

In addition to this verification mechanism, the instrument implements two additional mechanisms to protect the instrument from anomalous telecommands. Some commands are only allowed when the instrument is in CHECKOUT state. In addition to this, a number of commands have been declared 'critical'. For P-Alice this means that within a nominal 30-second timeout period a specific confirmation command has to be received before the actual (critical) command execution starts. During most of the in flight operations, this timeout is short and means that the confirmation already has to be issued before confirmation of the acceptance of the command has been received on the ground.

The set of telecommands can be divided into three categories:

General operations – These allow for the complete basic operational commanding of the instrument. This includes setting and storing of parameters and starting and stopping of the science acquisitions. This set of seven commands allows for the full science operations of the instrument.

Manual operations – Additional capabilities needed during commissioning and instrument verification are provided by 15 additional telecommands that allow for extended command options. Some of these commands may be used for science operations depending on the selected operational philosophy.

Memory functions – Software code management and maintenance and additional debugging functions are provided by three general purpose memory functions that allow for verification, load and dump of memory blocks.

Table 1: Command execution checks

Action	Checks	Effect
Command acceptance	• Command format and checksum verification, • Allowed command state, • Critical command confirmation, • Command parameter verification	Increment either command accepted or command rejected counter, Report error code if command was rejected
Command execution	• Check commanded results, • Completion of command execution	Increment command executed counter when execution is completed successfully, Report error code if command execution failed

Whenever the instrument detects errors while accepting or executing commands, an error will be reported in the generated telemetry packet. This includes an identifier for the telecommand (if available) and a general error code. The error code remains being reported in the telemetry data until another error is detected or the instrument is reset. This simple form of error reporting is limited to reporting a single error per second. An additional mechanism implementing a small error log is available for more extensive problem investigation. The command code for any successful command is also reported in the telemetry data so the telemetry registration can be used to reconstruct the received telecommands. Note that the parameters of a telecommand are not included in this reporting.

3.3.1 Acquisition commanding

The basic science data acquisition function of the instrument is handled by two 'start Acquisition' telecommands. One starts the histogram acquisition mode and the other the pixellist acquisition mode. These two commands each have two parameters; one defines whether the aperture door of the instrument is to be opened for the acquisition, and the other parameter allows for the selection of some test modes. All other acquisition parameters are controlled by the parameter list. This is a list retrieved from non-volatile instrument memory at startup. Once these general instrument parameters have been set, these parameters are expected to need little change, and this mechanism results in a very simple acquisition commanding.

Once the acquisition start has been commanded, the instrument goes through an acquisition startup sequence as listed in Table 2. A number of the steps in this sequence are optional as they depend on the current state of the instrument. This means that the duration of the acquisition startup (although deterministic) depends on the current state of the instrument. In any case, the actual acquisition startup will start synchronized to the spacecraft synchronization signal (next 1PPS synchronization pulse), so the exact time of acquisition start is known and can be planned for. For some cases, it still may be useful to configure the instrument before issuing the start acquisition command.

Table 2: Acquisition startup configuration sequence

Step	Parameter	Duration
Set the discriminator level	P_DISCRIMINATOR	~ 0 seconds
Command the pixel STIM	P_Stim_Enable	~ 0 seconds
Set the pixel list hack interval counter (only used for pixel list acquisition)	P_PIXELLIST_HACK,	~ 0 seconds
Set all eight hot segment masks	P_HOTSEG_1-8	~ 0 seconds
Command the aperture door to the state specified in the command parameter, wait for the control period if the indicated state of the door is not equal to the requested state	telecommand parameter	Door control period or ~ 0 seconds (if door indicated already in requested state)
Command the HVPS to the specified level () following the ramp-up algorithm specified in the HVPS module, the setup will only continue when the specified HV level is actually commanded (reached). If the HV was already commanded on before the start acquisition command this step will not add any time to the acquisition start processing	P_HV_LEVEL	Ramp-up time or ~ 0 seconds (if HV supply already at configured level)
Clear acquisition memory	N/A	~ 0.1 seconds

Step	Parameter	Duration
Initialize the acquisition timeout timer	P_ACQ_TIMEOUT	~ 0 seconds
For histogram acquisition initialize the exposure timeout timer	P_HISTO_EXP_DUR	~ 0 seconds

Whether the aperture door is opened for a specific acquisition depends on the operational procedures. The aperture door could (remain) closed for SOC acquisitions to prevent any spurious light from entering the optical system through the airglow aperture. For airglow acquisitions, the aperture door has to be open; this could either mean that the door is left open all the time or only commanded open for the airglow acquisitions. In either case in order to move the aperture door one of the actuator busses should be active; otherwise, any door commanding will have no effect.

For pixellist acquisitions a science data frame will be generated whenever the acquisition memory if full of data (photon and timehack events). The time it takes for this memory to fill depends on the selected timehack rate and on the brightness of the observed object. When an acquisition memory buffer is full, the acquisition will immediately continue in the next memory buffer while the filled buffer is transferred to the spacecraft and stored in the SSR. As this buffer full event is related to the number of detected event, it is not synchronized to the instrument time sync. The transfer of the first buffer to the spacecraft will be completed in about 0.5 seconds so in nominal operations the buffer will be available for acquisition before the second buffer is filled. For histogram acquisitions there is no such acquisition buffer full status, and the acquisition will continue until a pre-defined exposure period has expired, then the same action takes place as for the pixellist acquisition and the acquisition continues in the second buffer while the first frame is transferred.

Nominally, the acquisition operation will continue until the instrument is commanded to 'stop acquisition' explicitly. In addition, a timeout mechanism is implemented that will also terminate the acquisition mode when a predefined timeout period has expired; this mechanism may also be used to command acquisitions of a pre-determined duration. When the acquisition is terminated, the data already in the acquisition buffer is send to the spacecraft. For a pixellist acquisition still, a complete 32 k-word science frame is used even though it may only be partially filled. The used part in the pixellist frame can be recognized either in the pixellist frame itself as the data consists of all 'zero' events which will normally not occur in the data and will also be reported in the housekeeping packet. To command the 'stop acquisition' either the 'enter safe state' or 'enter checkout state' telecommand can be used. Both commands have the same effect in that the acquisition is terminated. The 'enter safe state' will, in addition to terminating the acquisition, also deactivate the HVPS and close the aperture door (if enabled and actuator power available).

3.3.2 State Machine Mode (SMM)

The State Machine Mode (SMM) is a backup pixellist acquisition mode that is activated autonomously if the software repeatedly fails. It is created to recover from a catastrophic software or system error and is inherently more reliable as it doesn't need any software to function and only a limited part of the hardware is used. The SMM mode uses hardware defined acquistion parameters that were fixed at instrument integration time. Therefore these parameters probably won't be the optimal parameters for the acquistion but it gives a chance to recover at least part of the science.

The instrument software is protected from end-less loop by a hardware watchdog mechanism. When this watchdog expires multiple times without the instrument being power cycled or hardware reset, the instrument will enter the hardware controlled State Machine Mode (SMM) acquisition. In this mode the instrument will perform a contineous pixellist acquisition with hardware defined parameters and generate the corresponding pixellist science data frames on the high speed link. The SMM will only terminate when the instrument is switched off. Note that any expiration of the watchdog counter already indicates a non-nominal event; normal instrument operations should never result in expiration of the watchdog timer.

Table 3: State Machine Mode triggers.

Watchdog resets	Minimum duration	Condition	Effect
>=4	16 sec	Always	Request instrument power cycle via HK flag
>=8	32 sec	Double instrument power supply	Activate hardware controlled SMM acquisition
>=16384	18.2 hr	Single instrument power supply	Activate hardware controlled SMM acquisition

After entering the hardware controlled SMM pixellist acquisition mode, the software is locked out from any acquisition control. Software could still be executing though but is will be limited to monitoring hardware operations. In addition telemetry could still be generated which may instrument status (via the low speed telemetry link), but this is no longer required for the acquistion to continue. The hardware controlled acquistion performs a pixellist acquistion with fixed parameters. These parameters were defined at instrument integration time and are listed in Table 4. At the start of the acquistion the hardware will also command the SOC door (primary and redundant) to open (execution of thise commands requires an active actuator power bus) using a fixed 160 ms pulse. Activation of the HVPSs will not follow a controlled rampup, the HVPSs will immediately be commanded to the full operating level.

Table 4: State Machine Mode (SMM) operational parameters

Parameter	Fixed value	Remarks
SOC aperture door	'open'	Activated (commanded open) for 160 ms at start of SMM, only has effect when one of the actuator busses is activated
Commanded High Voltage Level	157 (~ -4.5 kVolt)	HW configured value
HVPS select	both	Actual HVPS activation depends on the status of the instrument power busses
STIM	on	
Discriminator Level	43 (~ 0.5 Volt)	HW configured value
Time Hack rate	4 ms	

This watchdog mechanism protects the instrument from a fatal software failure by still providing a limited acquisition mode. Operation software may trigger this mechanism intentionally by causing the watchdog timer to expire multiple times. This provides for a way to test the SMM mode acquisition. This mechanism will fail to protect from a failing command channel; whenever the commanding channel is lost the software may still operate perfectly but just not receive any commands. But this requires a failure of both redundant Telecommanding channels.

4. INSTRUMENT OPERATIONS DETAILS

4.1 Instrument Startup

After power up or reset, the software initially starts executing special code from PROM that based on checksum verification determines which code is going to be executed. The system has redundant code storage in the form of one PROM memory page and four EEPROM memory pages that can all contain a complete executable code image. No copying of code is involved as the code can be directly executed from any of these five pages using page-switching hardware. No further selftest is performed at startup, as no options are available to take corrective actions. A selftest function is available in the normal operational system that allows an operator to select various test functions. Only after completing this code, the operational code is activated.

4.1.1 Startup Code Check

The system has redundant code storage in the form of one PROM memory page and four EEPROM memory pages that can all contain a complete executable code image. The startup implements the startup sequence as listed in Figure 7; this code attempts to find the first code page that contains correct code (starting with PROM and proceeding through the four EEPROM pages if needed). If all five-code storage pages are flagged incorrect, execution starts in the PROM page as this has the least probability of being corrupted.

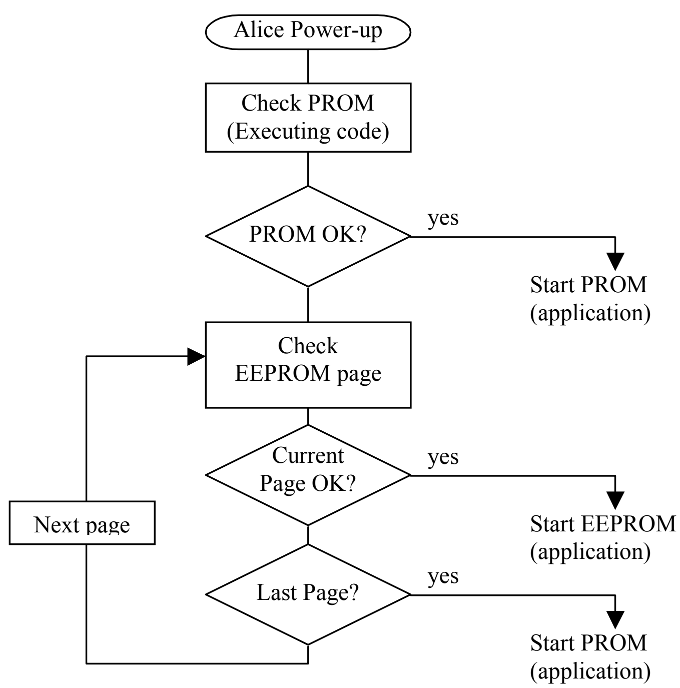

Figure 7: Alice Startup Code Check

The initial startup software first calculates the checksum over the complete PROM code that is going to be executed. This check uses a byte-wise16-bit xor-rotate checksum (same algorithm as show in section

4.7) resulting in a 16-bit checksum. Each of the four EEPROM pages stores it's own checksum (which could be different when different code versions are loaded). The checksum is calculated over the complete 32-kbyte code memory excluding the last 130 bytes. The first two bytes of these last 130 bytes are used to store the checksum itself (the remaining locations are used to store the 'non-volatile' parameter lists in the EEPROM memory (pages 1-3) copies and as scratch pad during EEPROM memory testing (page 4)). After selecting the code image to be executed, the startup sequence continues with the initialization of the different memory spaces and the start of the actual (operational) instrument software

This code checking is not foolproof. This verification relies on some code executing correctly form the PROM memory (which must be correct) but as this is a smaller amount of code the changes for corruption are less. To enable the check to run, some PROM code should be able to execute, but the required part is very small (~50 bytes) compared to the complete application software. Also the check might fail because of a bit error in a non-essential code location or in the checksum itself, but, after the system is started in EEPROM, it is always possible to jump back to PROM code. Even though this startup code check uses the less reliable memory as a backup strategy for the PROM, it improves the overall chances that the Alice software will start. The current executing code (PROM/EEPROM and EEPROM page) is reported in two fields in the housekeeping TM packet.

When modifying EEPROM code (patch command), it is safest to start with the last page and keep the lower numbered pages as backup of the original PROM contents (note: Alice will be launched with five identical copies of the flight software stored in PROM and EEPROM). These are created using the ‘duplicatePROM’ function, which copies the PROM contents to all four EEPROM pages). The programming of EEPROM can only be performed when the PAFS is executing from PROM (see 4.7).

4.1.2 System startup activity timing

The next two tables show the results from simulations of the system during the startup phase. The results show the time (in seconds) at which various activities occur during the startup sequence in a case where no errors are detected in the PROM memory. If errors would be detected, the startup sequence would take longer as additional operations are required to verify the EEPROM code pages.

After simulation start, the instrument software first goes through the nominal startup sequence consisting of a PROM check and initialization of the memory areas. When this is completed, one of the first actions are a number of early hardware initializations including the termination of any ongoing actuator activation (including HVPS). This is visible in both cases: “1.680 – Aperture control stopped”. This means that the complete internal startup is completed within 1.68 seconds. Next the parameter file is read from 3 pages of the EEPROM memory; this is shown when the software switches the EEPROM ‘memctl’ register. After that, the aperture door is commanded to close as defined in the used configuration file; door control is activated for the (configuration file determined) period of about 1.8 seconds. The two presented cases show the effect of the absence of the 1PPS clock on the system startup. If there is no 1PPS clock, the instrument will, after waiting for some time and when no 1PPS is received, start processing on the internal clock. Note that for the simulated 1PPS active case, the simulation case shows a specific case where the 1PPS happens at the same time when the instrument power is applied. If power up occurs in different phase timing may differ slightly.

The hexadecimal dumps in the simulations show the raw contents of the generated housekeeping packets. These consist of the complete ITF packet with the first synchronization byte changed to 0x 64 for simulation control purposes.

1PPS and clock message active

```
  0.000 - simulation start (start 1pps & time message)
  1.000 - 1pps + time msg 10000
  1.680 - Aperture control stopped (- s)
  1.688 - memctl, switch EEPROM page (p0.0) to (p1.3)
  1.697 - memctl, switch EEPROM page (p1.3) to (p2.3)
  1.706 - memctl, switch EEPROM page (p2.3) to (p3.3)
  1.899 - Start Aperture control close
  2.000 - 1pps, previous not acknowledged
  2.087 - TM transfer started
  2.120 - itf transmitted (116 bytes)
0x00: 64 fa 30 04 6b 00 6d 40 00 00 20 21 00 00 00 fe
0x10: 00 00 00 00 0c 82 c0 00 00 59 00 0f 42 40 20 c3
0x20: 21 00 00 00 00 00 00 ff ff fe 00 00 00 10 00 00
0x30: 00 01 86 ca 01 f7 00 00 00 00 7f ff ff ff 00 00
0x40: 00 00 02 00 00 02 00 00 00 0c 00 00 66 66 68 68
0x50: 78 7a 82 66 00 00 00 10 47 11 e0 48 8a 00 00 00
0x60: 02 00 3f 43 45 f1 f3 f5 f7 f9 fb fd aa ff 20 f8
0x70: 01 33 56 17
  3.000 - 1pps + time msg 10002
  3.087 - TM transfer started
  3.120 - itf transmitted (116 bytes)
0x00: 64 fa 30 04 05 00 6d c0 00 00 20 21 00 00 00 fe
0x10: 00 00 00 00 0c 82 c0 01 00 59 00 00 27 12 20 c3
0x20: 6c 00 00 00 00 00 00 ff ff fe 00 00 00 10 00 00
0x30: 00 01 86 ca 02 f1 00 00 00 00 7f ff ff ff 00 00
0x40: 00 00 02 00 00 02 00 00 00 0c 00 00 66 66 68 68
0x50: 78 7a 82 66 00 00 00 10 47 11 20 48 8a 08 45 00
0x60: 1f 00 3f 43 45 f1 f3 f5 f7 f9 fb fd a8 ff 20 f2
0x70: 02 1e 34 5c
  3.676 - Aperture control stopped (1.777 s)
  4.000 - 1pps + time msg 10003
  4.087 - TM transfer started
```

No 1PPS or clock message

```
  0.000 – start simulation
  1.680 - Aperture control stopped (- s)
  1.688 - memctl, switch EEPROM page (p0.0) to (p1.3)
  1.697 - memctl, switch EEPROM page (p1.3) to (p2.3)
  1.706 - memctl, switch EEPROM page (p2.3) to (p3.3)
  1.896 - Start Aperture control close
  3.159 - TM transfer started
  3.197 - itf transmitted (116 bytes)
0x00: 64 fa 30 04 16 00 6d 40 00 00 20 21 00 00 00 fe
0x10: 00 00 00 00 0c 82 c0 00 00 59 00 0f 42 40 20 c3
0x20: 01 00 00 00 00 00 00 ff ff fe 00 00 00 10 00 00
0x30: 00 01 86 ca 03 04 00 00 00 00 7f ff ff ff 00 00
0x40: 00 00 02 00 00 02 00 00 00 0c 00 00 66 66 68 68
0x50: 78 7a 82 66 00 00 00 10 47 11 00 48 8a 09 cd 00
0x60: 14 00 3f 41 43 f1 f3 f5 f7 f9 fb fd a8 ff 20 06
0x70: 01 33 13 34
  3.687 - Aperture control stopped (1.791 s)
  4.159 - TM transfer started
  4.197 - itf transmitted (116 bytes)
0x00: 64 fa 30 04 7b 00 6d c0 00 00 20 21 00 00 00 fe
0x10: 00 00 00 00 0c 82 c0 01 00 59 00 0f 42 41 20 c3
0x20: 01 00 00 00 00 00 00 ff ff fe 00 00 00 10 00 00
0x30: 00 01 86 ca 03 fe 00 00 00 00 7f ff ff ff 00 00
0x40: 00 00 02 00 00 02 00 00 00 0c 00 00 66 66 68 68
0x50: 78 7a 82 66 00 00 00 10 47 11 00 48 8a 0a fb 00
0x60: 19 00 3f 41 43 f1 f3 f5 f7 f9 fb fd a8 ff 20 00
0x70: 02 1e b2 92
  5.160140 - TM transfer started
  5.198099 - itf transmitted (116 bytes)
0x00: 64 fa 30 04 15 00 6d 40 00 00 20 21 00 00 00 fe
0x10: 00 00 00 00 0c 82 c0 02 00 59 00 0f 42 42 20 c3
0x20: 01 00 00 00 00 00 00 ff ff fe 00 00 00 10 00 00
0x30: 00 01 86 ca 04 f8 00 00 00 00 7f ff ff ff 00 00
```

4.2 Telemetry Generation

The instrument generates both low speed telemetry and high-speed telemetry. Both are transmitted via a redundant interface to the spacecraft. Data on primary and redundant interfaces is identical. The housekeeping telemetry generation starts a few seconds after power up and in nominal operations does not cease until the instrument power is switched off. The high-speed science telemetry is only generated when the instrument performs acquisition operations.

The data handling system of the spacecraft will further influence the telemetry timing. For the low speed data either all telemetry or a sub-sampled set may be forwarded immediately to the ground. In addition to this all telemetry will be recorded on the Solid State Recorder (SSR) from which all or sections may be later on selected for download.

4.2.1 Low Speed Telemetry Generation Timing

After receiving a 1PPS pulse, the housekeeping task starts gathering the required telemetry data. When the complete TM packet is build, the TM transfer will start and sending of small blocks of bytes to the TM FIFO will continue until the complete packet has been send. If the transfer could be sustained at a continuous transfer rate of 38k4 baud, the transfer of the TM packets would be completed after:

Nominal TM packet (non-packetized + HK packet)	116 bytes	30 ms
Data dump TM packet (above + MD packet)	262 bytes	68 ms

In the system most tasks have a higher priority than the TM transfer task and, thus, the effective transfer time may be larger as the actual transfer will occur in interrupted chunks. Note that in case a data dump is being generated the software in parallel to the sending out of a larger TM packet also builds the next data dump packet so this operation also increases the software execution load.

On the actual instrument some measurements were performed that confirmed the TM timing for the nominal (idle) situation. In the software-simulated environment a simulation was run with various loads on the system the following values were found for the TM transfer link usage:

Table 5: Low Speed TM transfer timing

Activity	TM start delay	TM transfer duration	TM complete time
	FM hardware measurement: times in seconds		
idle	0.082	0.040	0.122
	Simulation results: average times in seconds		
idle	0.087	0.034	0.121
dump (memory)	0.147	0.180	0.327
idle	0.087	0.033	0.120
checksum	0.137	0.569	0.706
idle	0.088	0.033	0.121
testpattern	0.130	0.623	0.754
acquistion	0.088	0.033	0.121

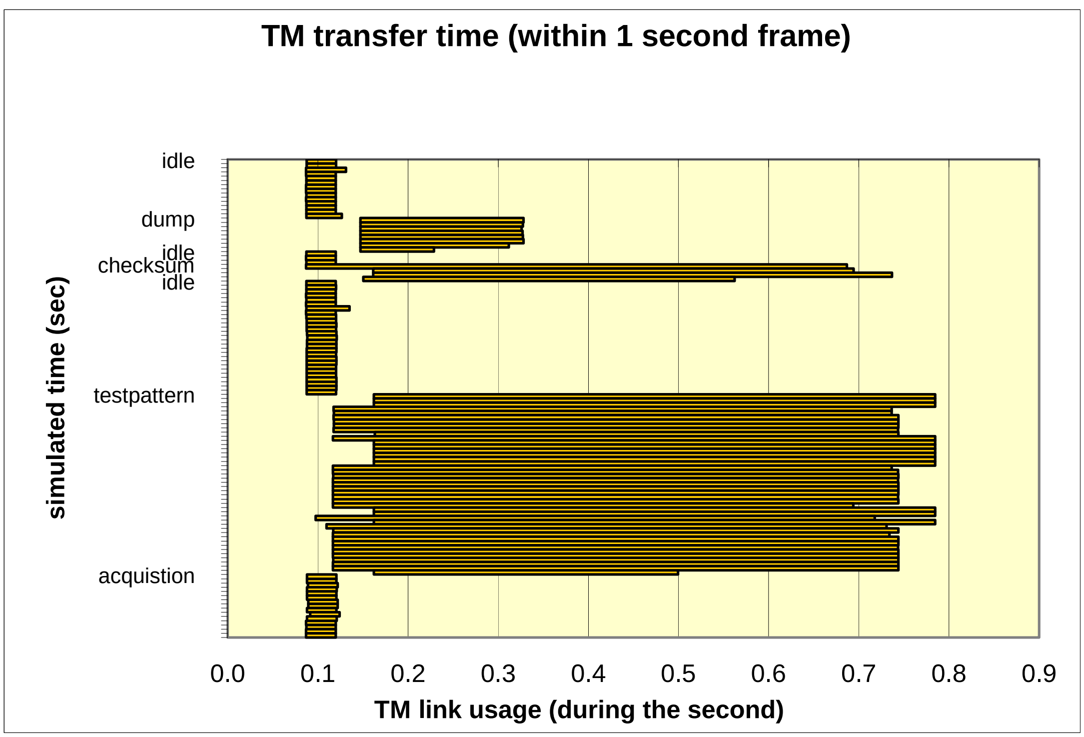

Figure 8: Simulation results: Low Speed TM transfer timing

In the dump phase a double effect is shown, both the needed TM transfer time increases and the load on the system increases as the gathering of data for the next dump packet proceeds. At the end of the test, a test pattern acquisition was started; initially this puts a very large load on the system during the generation of the test pattern. Once the pattern generation is complete though, the software has hardly anything left to do, and the situation is similar to an idle situation. This situation corresponds to a normal acquisition where the software after startup of the hardware acquisition has hardly anything left to do.

4.2.1.1 High Rate (Science) Telemetry

The high rate science TM transfers from the instrument to the spacecraft is almost completely controlled by instrument hardware; the software is only involved in controlling this interface indirectly during acquisition control. The software controls the start and stop of acquisition operations and in case of a histogram acquisition the exposure time. In pixellist acquisition mode, the rate at which the acquisition buffer is filled is determined by the observation and the selected time hack rate. Science data transfers will occur whenever and acquisition buffer is filled. The data transferred via this link from the instrument to the spacecraft always consist of complete frames of 32768 16-bit words. After storing this data in the SSR, the spacecraft will take case of sending this data to the ground for this transfer the science frames will be packaged into a large number of CCSDS transfer packets.

Transfer of a single 32 kword (16-bit) science frame via the serial LVDS interface to spacecraft data handling system takes 17 clock pulses for each 16-bit word to transfer. At the 1MHz clock rate this results in a complete frame transfer time of: (16+1) * 32768 / 1MHz = ~0.5 sec.

The CCSDS packets used by the spacecraft to transfer the science data to the ground can contain up to 480 bytes. This means that to transfer a complete (un-compressed) Alice science frame 137 CCSDS packets are used.

4.3 Telecommanding

Telecommands are sent to the Alice instrument in the data field of command message ITFs as described in Appendix B. Both telecommands and time synchronization messages are send via this interface in the ITF format to the instrument. In nominal operation, the spacecraft will every second send a time message to the instrument and up to one telecommand may be send every second. The section continues with some general description of the two formats and the processing of the two message types followed by a detailed description of the processing applied to the incoming data stream. This description is complex, as it both needs to deal with input from the redundant interfaces and needs to be capable of handling numerous error conditions.

4.3.1 S/C Time Message

Each second the spacecraft data handling system will send a Time Message to Alice via the serial RS-422 interface. This Time Message will be transferred as an ITF formatted packet. This message will contain the S/C time that is valid at the next time synchronization pulse. The transfer of the time message will be completed not later than 50 milliseconds before the next time (sync) pulse. The message data for the S/C Time Message is shown in Table 15. The size of the S/C time message is fixed and the message length field in the ITF is always 5. Under off-nominal conditions, the S/C C&DH may stop sending the S/C Time messages.

The time message together with the periodic synchronization pulse is used to synchronize the instrument clock. In addition, the time message also contains the 'memory dump' allowed flag that determines whether the instrument is allowed to include memory dump data in the next generated telemetry transfer.

4.3.2 Instrument (Tele)command Message

The general structure of the Alice telecommands conforms to the New Horizon TC format as shown in Table 16, but Alice does not verify the 32-bit CM checksum field. The number of command words and the meaning of the values depend on the specific command as specified by the Op-code field. Two commands are special in that the spacecraft defines the op-code and structure; they are the memory load and memory dump telecommand. For P-Alice, these commands are described / executed just like any of the other telecommands.

P-Alice uses a compact command set with 25 different telecommands to control the operation of the complete instrument as listed in Table 23. Each second, Alice is capable of handling (starting execution) of a single telecommand in addition to the time message. For commands for which the timing is critical, like the acquisition command, the actual operation will start at the next sync pulse (or assumed pulse if no real sync signal is available).

Except for the memory function related telecommands, all commands are small and require either no parameters or only a single word. So, the overall P-Alice telecommand size for the nominal telecommands is either 8 or 12 bytes. The memory function telecommands though are not meant to be used during nominal (science) operations.

Telecommands that can be executed immediately and that take hardly any time will be executed directly by the task that receives the telecommand. Some telecommands take inherently more time to execute, for instance because the command generates multiple memory dump packets or includes extended hardware activation. In these cases, the telecommand receiving process verifies the parameters and activates the processing of the telecommand by a separate task. From this point on, the telecommand processing task is ready to receive (and process) a new telecommand.

4.3.3 Command Processing

The overall command processing is a fairly complex function as it both handles the redundant telecommand channels and it also deals with the error detection and recovery. Also included in this processing is the general command processing that is performed before the actual command is executed after being dispatched to the appropriate module.

The telecommand dispatch and processing is not dependant on the occurrence of the clock synchronization pulse (1PPS). Most commands don't even require any form of synchronization and are executed immediately; only the acquisition start and stop are synchronized. This independence of the synchronization signal means also that the instrument in principle can handle multiple commands per second as long as the internal 256-byte software buffer doesn't overflow and the command processing doesn't take more time than available. This means that the instrument can easily handle multiple simple commands for which the execution is nearly instantaneous or even that a 'start acquisition' command together will the required 'confirm critical' command could be send to the instrument within a one second timeframe.

4.3.3.1 Redundant channel handling

The software is initially listening for the command input from either serial channel. As soon as a single byte is received from a channel, this channel becomes the active channel and the data from the other channel is discarded (see Figure 9). This initial condition may be overridden by a selection in the parameter file that disables a specific channel is case of known problems. Once a telecommand is received and handled, the input processing remains in the same state (listening to the selected active channel) whenever the S/C communication remains nominal, meaning that the synchronization pulse is received on the same side as the telecommand.

The detection of a command error may occur at various levels within the telecommand decoding function. Whenever the synchronization signal disappears on the active side or too many errors are detected in the received commands, the input processing restarts itself to the condition of listening to both sides. This selection process results in a smooth switchover between the primary and redundant telecommand channels without the need for an explicit reset of the instrument both in the presence (nominal case) and absence of the spacecraft synchronization pulse; an overview of the handling of the different nominal and non-nominal TC cases is shown in Table 6.

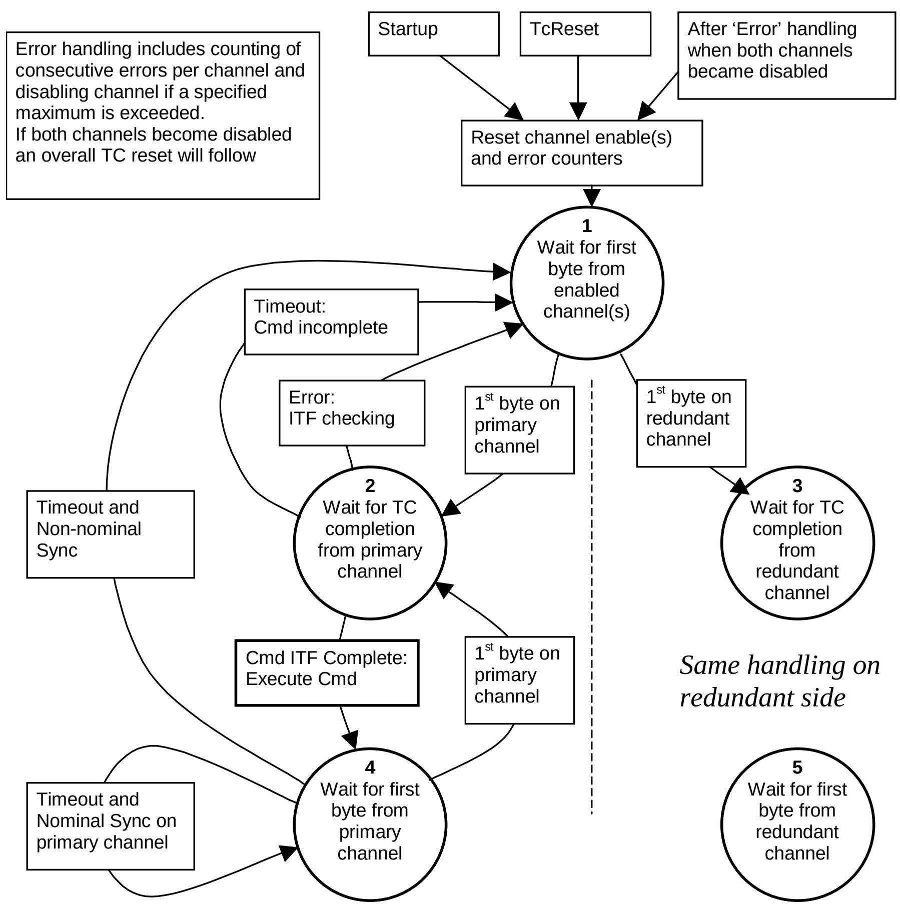

Figure 9: Primary - Secondary command processing

Table 6: TC processing behavior

Situation	Behavior	Switchover time
Nominal situation, TC and Sync active on a single channel	Telecommand acceptance follows active channel	Direct (next second)
No sync, TC on either channel	Telecommand acceptance follows active channel	Channel retention timeout (few seconds)
Commands on both channels	First detected byte determines active channel, remainder of data from inactive channel is discarded	Channel retention timeout (few seconds)
Sync only on other channel	Telecommand acceptance follows active channel	Channel retention timeout (few seconds)
Partial command	After command timeout the system returns to the waiting for initial by state	Not applicable

4.3.3.2 Character (Byte) input processing

Telecommand processing starts with data bytes being received by either one of the hardware receive UARTs (see Figure 10). These UARTs includes a 20-byte internal hardware FIFO buffer. This buffer is smaller than the maximum size telecommands (memory load telecommands can be up to 144 bytes + 7 bytes ITF header in size). The software (ISR TMTC) provides an additional buffer function that decouples the command processing from the input stream. This software task (indicated by a circle in the diagram) that handles the input from the UART only uses data from a single source at any given moment, determined by the current active side. This software is able to transfer the full rate input (transfers at 38k4 baud resulting in a maximum of 3840 bytes per second) into the internal software buffer.

The worst-case arrival time of the telecommand bytes at 38400 baud is once every 0.26 milliseconds; this means that the hardware FIFO can buffer little over 5 milliseconds. Within that period the software has to be able to transfer the received data to the software buffer. This second level software buffer is capable of storing 256 bytes corresponding to worst-case 66 milliseconds of data. During nominal operations this maximum should never be reached though, as only one telecommand and one synchronization message are to be handled every second and their maximum sizes are smaller than the available software buffer size.

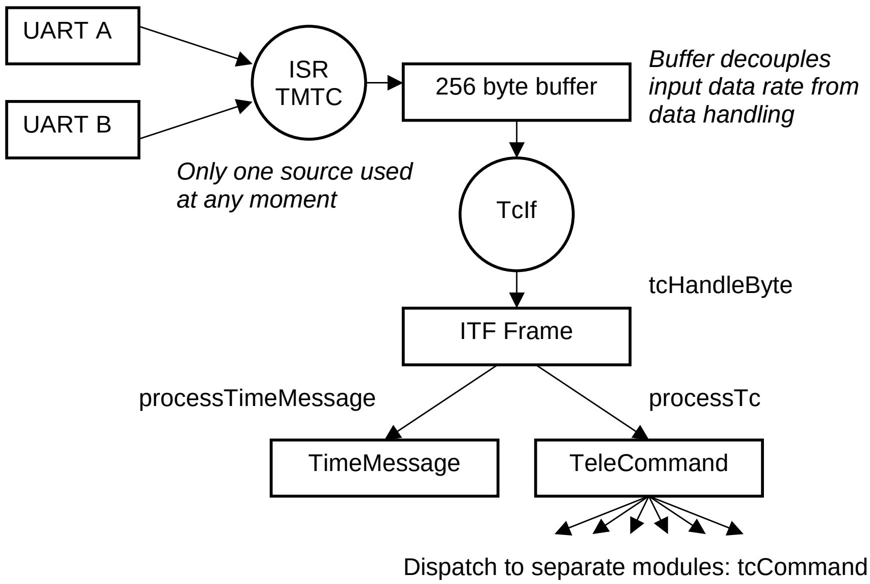

Figure 10: Command input handling

4.3.3.3 ITF reassembly and checking

The software (TcIf task as shown in Figure 10) reassembles the stream of received bytes into the ITF transfer frames using a state machine. This state machine decodes the synchronization pattern and the following ITF header bytes and then reads as many bytes as indicated. If the telecommand is not completed within a specified period, the decoding times out and the decoder returns to its initial state. Received ITF are checked before being accepted. The checking includes the following verification:

- ITF message type
- Telecommand (ITF) checksum

ITF message size

Once a complete ITF frame has been received the processing determines whether a time message or a command message was received. Received time messages are used for the instrument Clock function (see 4.4.3) where the received data (time and dump allowed status) is used for the time synchronization. Received telecommands are further checked and forwarded to the appropriate module for processing. Additional checking of the telecommand package includes:

ITF length versus indicated command length

Valid command opcode

- Indicated length versus expected command length

The command checksum is not included in this check for a correct command package. In case a problem is detected, the command decoding is aborted, an appropriate error message (see section 4.4.2 and Appendix G) is generated and included in the telemetry data and the system starts waiting for the next command.

4.3.3.4 Critical command processing

The Alice instrument provides two levels of command verification to prevent erroneous commanding of conditions that might cause damage to the instrument. The first level of protection is provided by the operational state selection; not all commands are allowed in all states. In addition, a 'critical' command mechanism is used that requires confirmation for certain types of commands. These critical commands are different from the critical commands mentioned in the ground system; there the critical command system prevents sending command to the spacecraft until that operation is confirmed.

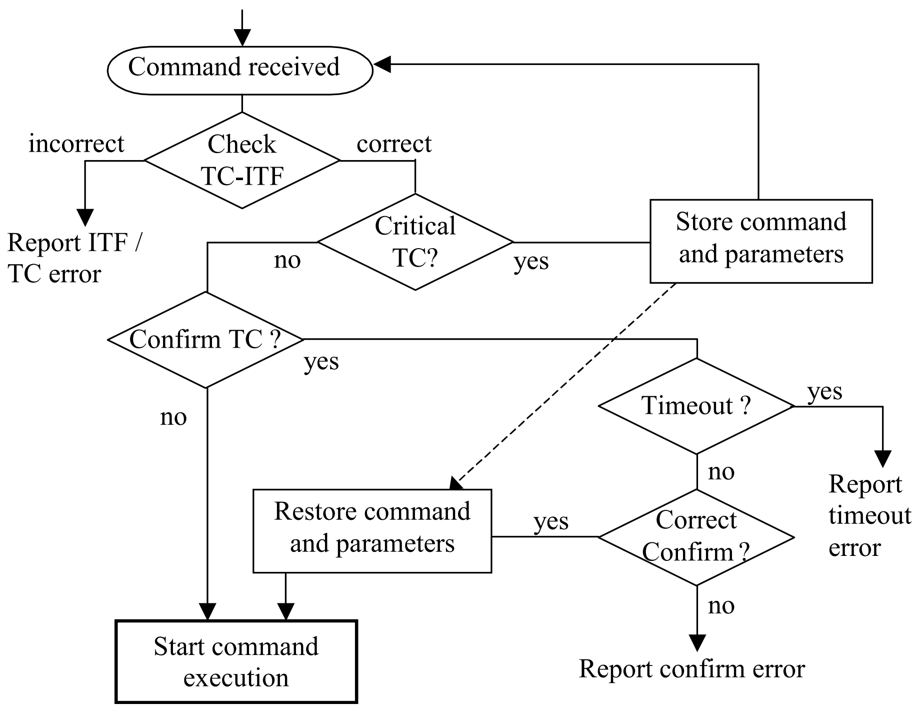

Figure 11: Critical command handling flowchart

Commands that are marked as critical such as 'door opening' and 'high voltage activation' are related to the safety of the instrument. These commands require a two-step commanding process before they are executed. After receiving such a critical command, execution doesn't start immediately but the command is initially stored in an internal buffer. Once the confirmation command (a separate telecommand) is received within the timeout period specified in the parameter file (P_CMD_TIMEOUT), the command execution will start. If no confirmation is received within this timeout, a confirmation is received for the wrong command or another command is received, the initial command will be discarded. This failure will be indicated in the housekeeping packets. Figure 11 shows a flowchart to illustrate the basic functionality of the critical command handling.

4.4 General Instrument operations

4.4.1 Instrument state

In the previous chapter the instrument states were already described and a state transition diagram was presented (Figure 6*)*. Here the states (Table 7) and the state transitions (see Table 8).are described in more detail.

Table 7: Operational states

State	TM	Power	Description
Off	N/A	Zero	Instrument not active
Safe	Only HK TM	Minimal	HV and heaters off
Acquire	HK TM and Science TM	Nominal (HV active), may include initial door opening/closing	HV On (but level may be set to zero) automated safety checking active, acquisition hardware active (heaters may remain at commanded setpoint)
Checkout	HK and Memory Dump TM (memory dump packets are only included when requested and when allowed by the s/c)	Varies, HV, door operations, actuator and/or heaters may be activated	Used during test, verification and commissioning, manual commanding, safety checking active (when HV activated), heaters may be commanded on

Table 8: State transition actions

Transition	Allowed	Actions
Safe to Checkout Safe to Acquire	Only when safety timeout is not active (or override is active)	Clear reported 'last active safety'
Checkout to Acquire	always	None (acquisition start is part of the command processing)
Checkout to Safe Acquire to Safe	always	- HVPS off, - stop actuator commanding, - stop heater control, - close aperture door (if enabled and actuator bus powered)
Acquire to Checkout	always	None, posting the synchronized acquisition stop is part of the command processing

4.4.2 Error reporting

The housekeeping packet includes three counters that indicate the acceptance and execution state of received telecommands. One of these two counters will be received whenever a telecommand is received.

In addition, an executed command counter is reported that will be incremented whenever a command successfully completes execution. Whenever the rejected command counter is incremented, or when the executed command counter is not incremented after a command was accepted, an error would be reported. There are also some other cases where errors could be reported that are not directly related to execution of telecommands.

Two error-reporting mechanisms are implemented in the instrument. In each generated TM packet (and in the non-packetized telemetry) the most recent detected error and, if applicable, the related telecommand are reported. In addition to this mechanism a 16-entry circular buffer error log is maintained.

4.4.2.1 Most recent error reporting

The housekeeping packet contains the following fields:

- TcLastFailCode- last reported error code,
- TcLastFailed - last failed telecommand.

These two fields will be updated whenever a new error is detected. The last failed command code is a one byte field that contains the low order byte of the command code that caused the problem. In some cases, for instance, when errors are not directly related to telecommands being executed, or when no telecommand was yet recognized, only the error code filed will be updated. The error codes used by the instrument are listed in Appendix G.

The same error codes remain reported in the housekeeping packet until either another error is detected or when the instrument is reset. When multiple errors are detected in the same one-second period, only the last error code will be reported. After power on (or hardware reset) the last failed command will be set to 'no command' (special code ccNO_CMD = 0xff) and the error code will be set to indicate the power up reset state (special code ecNO_ERROR_INITIAL = 0xfe). Both error reporting fields are reset whenever the instrument receives the Tc_RESET_TC_STATUS command. In that case, the error code will indicate the reset state (special code ecNO_ERROR_RESET = 0xfd). This error reporting mechanism in addition to the three 'command' counters should be sufficient for 'nominal' error status reporting.

4.4.2.2 Circular error log

To support additional error investigation, the instrument implements a circular error log. The contents of the buffer may be dumped using the available memory dump command. For each detected error, this log contains time of occurrence, the error code and some status information. The error log uses the same basic error codes as reported in the housekeeping data. The error log provides a way to investigate in case multiple errors per second are reported or when not all housekeeping packets are available for investigation.

To keep the mechanism simple, the size of the circular buffer is limited to the size of one memory dump packet (128 bytes = 8 entries). The error log is located at the fixed address 0x8200 in normal memory data space (DATA).

At startup the circular error log is initialized with time values 0x46726565 ('Free') and no error code (ecNO_ERROR_INITIAL = 0xfe). The error pointer is initialized to the first entry. The error pointer is never reset during nominal operations. When a Tc_RESET_TC_STATUS is executed an additional special entry is added to the log (special code: ecNO_ERROR_RESET = 0xfd). In any case the next entry to be used in the error log is indicated by a special entry (ecNEXT_ERROR_LOG_SLOT = 0xf8).

4.4.3 Instrument Time

The instrument maintains a MET (Mission Elapsed Time) with a resolution of 1 second. The instrument does not maintain the time with a higher accuracy than one second, although a number of operations are synchronized to the received 1PPS pulse. In normal use, this time is synchronized to the spacecraft MET by means of the time message and the 1PPS synchronization pulse. The Alice instrument was designed such that performance will be gracefully degraded when either or both of these mechanisms are unavailable. If the 1PPS pulse becomes unavailable, the instrument will start generating an internal (assumed) clock event that is derived from the X-tal oscillator and use this for all internal timing. The time message also transfers the memory dump allowed flag from spacecraft to the instrument. If no time synchronization message is received, the MET value will simply be incremented at the next 1PPS pulse (real or assumed).

The instrument concept of time consists of a 32-bit counter that counts the seconds since some given moment. A 32-bit integer number allows the specification of a period of 2^{32} = 4294967296 seconds (about 136 years); this should be enough for even an extended mission. In the nominal operation, a time update is received every second the time is received from the spacecraft in a time message; the value reported in this message becomes valid at the first following synchronization pulse. The module does not check the received time message for any specific value or range; any received time value is accepted and will become active at the next synchronization pulse. When no time message was received in the second before the synchronization pulse, the current time value will simply be incremented. This means that in case the time message generation ceases the instrument will still report incrementing (unique) time values. If no time message is received after an instrument reset, counting will restart from a fixed value (1000000).

4.4.3.1 Clock synchronization

The clock synchronization is a very simple software task that spends most of its time waiting for the synchronization signal (1PPS). If this signal is received, the instrument time will be updated; this means that either the time value received in the last second (time message) becomes active or that the time counter is simply incremented. In addition to this time update, the received 1PPS pulse is also used to synchronize the start and stop of acquisitions.

Whenever no synchronization signal is received after a waiting period (consisting of the 1-second nominal time and a 100 millisecond margin) the system ‘assumes’ a missing synchronization signal and the time update (and acquisition synchronization) is triggered in the same way. After this, the software starts using an internal generated synchronization signal that is generated every second to trigger the Clock module. This internal generation is terminated whenever a new external synchronization signal is detected. To prevent synchronization events occurring to close together (when transitioning from the internal to the external synchronization) the first received external synchronization pulse is discarded before the nominal synchronization processing is resumed. Note that there is no provision to recover from too many synchronization signals being received. This anomaly is considered too unlikely to validate the increased complexity needed to handle this.

4.4.4 Parameter storage

The instrument software includes a parameter file that stores a number of parameters that control instrument operations. For parameters that only occasionally change, this mechanism allows for small telecommands while still providing operational flexibility. To facilitate operations, this parameter file was implemented as a general mechanism that stores all of these constants in one location. The parameters in this table are listed in Appendix F. The contents of the parameter file are loaded into data memory from EEPROM memory whenever the system is started or restarted (see Figure 12) and these values in RAM are used during operations. At any time after that, individual entries in the parameter file can be altered by command (Tc_SET_PARAMETER). These changes to the table values only affect the values stored in this copied (active) table. A change to the table can be made permanent (persistent) by first changing the value in this table (Tc_SET_PARAMETER) and then commanding a parameter table store operation (Tc_STORE_PARAMETERS) that copies the current RAM table values into the three redundant EEPROM tables. At any time a reload of the data memory version of the table from the EEPROM memory can be commanded (Tc_LOAD_PARAMETERS), restoring the table to its power up default values.

The physical storage location of the redundant parameter memory is at the end of the last three of the four EEPROM code pages. At the end of each of these pages a small area of memory is reserved for this parameter storage. The three copies are thus spread throughout the instrument memory and are not all in one location. Whenever the parameters are loaded from the EEPROM memory into the data memory system start-up or explicitly by telecommand, see Figure 12), the values from all three tables are read and a three-way voting mechanism is used to determine the correct value. The following Table 9 shows how the system handles the different voting situations. Reporting of possible problems will use the command failure code as defined in the housekeeping packet (Hk_LAST_FAIL_CODE, see also Appendix G).

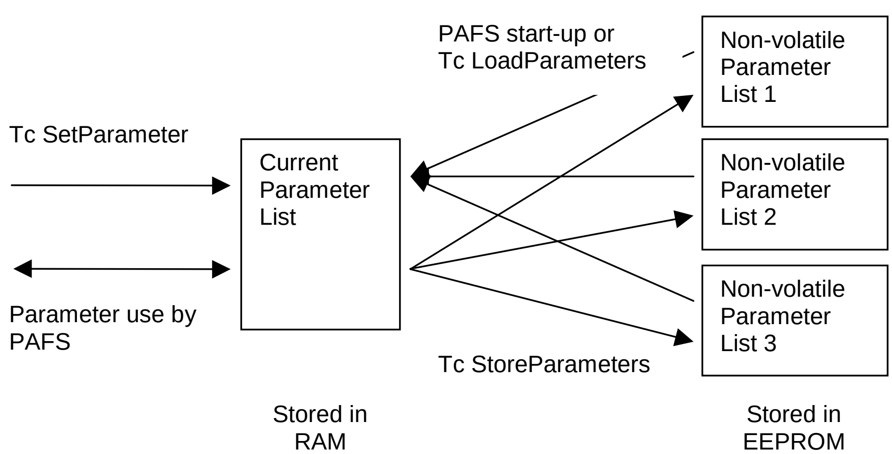

Figure 12: Parameter List storage and transfer

The 'permanent' version of the tables is stored in EEPROM memory. This memory can only be modified a limited number of times (hardware components are rated for 10000 page mode write cycles (see section 4.7.2), so it should be more than sufficient for the nominal mission operations). To keep track of the number of modifications made (write cycles), an additional count value (P_NumberOfModifications) is maintained in the table as the last entry. As part of the commanded table store operation (Tc_STORE_PARAMETERS), the current count value will be incremented so this entry always stores the number of performed write cycles. This telecommand can only be executed when the system is currently executing code from PROM. This is a hardware limitation, as otherwise the code execution will conflict with the EEPROM write operation (see section 4.7).

Table 9: Parameter File majority "Voting"

Voting Case	PAFS Action
All 3 agree	Use the value for the parameter
Two agree	Use the value of the two tables that agree for the parameter value, report the problem
All 3 different	Don't change the parameter value, go to SAFE state, report the error

The system implements a mechanism to verify the current contents of the parameter table. In order to limit the required telemetry bandwidth, a single (byte size) parameter is reported in each housekeeping telemetry packet (Hk_PARAM_VALUE). In order to identify the current reported parameter value; the index of this value is also included in the housekeeping telemetry packet (Hk_PARAM_INDEX). Reporting of a specific parameter can be requested by specifying the requested parameter by telecommand. This parameter is stored in an entry in the parameter list so the parameter to be requested is specified by the 'SetParameter' telecommand. This results in reporting the requested parameter starting with the next telemetry packet. This reporting continues until a different parameter is selected. To facilitate verification of all the parameter values (e.g. during testing) a special value is defined (cyclic reporting = 0xff) which will command the reporting of all the parameter values in successive housekeeping packages. In this mode, the identification of the current reported parameter value (Hk_PARAM_INDEX) cycles through all the parameters indices. Again, this reporting will continue cyclical until a specific parameter is requested. In addition to this the available memory dump command can always be used to get a snapshot of the complete contents. Either the current active table that is stored in DATA memory at address 0x8300 or one of the non-volatile tables stored at address 0x7f80 in the first three EEPROM pages can be requested.

The system also supports a directed read in which a parameter of the 'LoadParameters' telecommand specifies from which of the three tables the data should be loaded. In this case only a single table is read so no voting is applied. This operation is only intended for non-nominal operations where the system operator needs to circumvent known problems or needs to verify separate memory banks. In addition to this the table can also be initialized with 'hard-coded' values, this loads default values from PROM memory into the table; these values are the same values that are used to initialize the table before the initial 'start-up' load is performed.

4.4.5 Aperture Door Operations

After release of the launch latch, the airglow aperture door can be operated multiple times during the mission. A simple Limited Angle Torque (LAT) motor moves the aperture door. Two read-back sensors are used to determine positively if the door is fully open or fully closed. Control of the door uses a very simple open loop mechanism. To operate the door the control is always activated for the parameter file specified control period (P_DOOR_CONTROL, nominal 1.8 sec). No feedback from the reported door state to the door control function will be implemented (open loop control).

The door sensors use optical switches; in order to limit power consumption, the optical switches are only activated when a door measurement is needed. This same sensing system is also used for the detector door position switch. The system determines the position of both doors periodically once per second. The two aperture door position sensors allow for positive verification of both the opened and the closed state. For the detector door (one time operation) only a single switch is available that can verify the opened position.

4.4.5.1 Aperture door performance measurement

In order to monitor wear of the door during the lifetime, a measurement function is implemented that measures the door operation times. This consists of a special function, which performs a detailed timing measurement of the door operation. The special function can be activated using the TcStartProgram telecommand specifying the start address of the function (0x25b2). The special function will temporarily stop normal processing but since the performance measurement is completed so quick no telemetry data will be lost. After measurement completion the nominal instrument operation will continue.

A single activation of the measurement function will perform four time measurements with a 0.1 ms resolution. The first two measurements report the results of the open movement of the door by reporting start and completion of the movement. The start of the movement is defined as the moment the door leaves the first opto switch and the completion is defined as the time when the end opto switch is reached. The second two measurements perform the same on the closing operation. All four values are reported as 16 bit unsigned integers. The results of the measurements are stored in a 128-byte circular memory buffer at address 0x8500 in DATA memory. The memory dump function can be used to retrieve these measurement results.

During radiometric calibration the following values were measured under vacuum for the door operation times.

Table 10: Measured Aperture door movement times in vacuum

	Movement start	End reached
Aperture door open	14.8 ms	63.3 ms
Aperture door close	14.2 ms	65.6 ms

4.4.6 HVPS control

The control of the High Voltage Power Supply (HVPS) is a main task of the instrument software. The instrument contains two redundant HVPSs that power the (single) detector. Each HVPS is controlled through a voltage level setting, an enable (on/off) control and an arm control. Each HVPS is powered from the corresponding Low Voltage Power Supply (LVPS) and a HVPS can only be activated when the corresponding LVPS and thus spacecraft instrument power bus is on. Nominally, both HVPSs are controlled in parallel, but two flags in the parameter file can be used to disable either one in case of problems.

The HVPSs are controlled automatically by the software as needed to support science data acquisition. In the CHECKOUT state the HVPSs may also be commanded directly by telecommand (Tc_ACTIVATE_HVPS, Tc_DEACTIVATE_HVPS). In both cases the same rules govern the operation of the HVPSs. Both supplies have separate monitors for the MCP voltage (McpV), the strip current

(StripI), and the anode voltage (AnodeV); these parameters are measured and monitored by the Safety task.

Operation of the HVPSs is related to the safety monitoring. The HVPS control controls the HVPS activation, and, during this activation, the safety monitoring will periodically verify the measured parameters. This safety monitoring is a separate activity that may in case of problems override the HVPS control. HVPS control will command the HVPS hardware according to the rules described in this section. The relations of the HVPS control with the safety checking are the following:

- First, in parallel the safety checking will continuously monitor the HVPS performance and take corrective actions that override the HVPS control Function
- Second, if a safety condition is in effect, then the HVPS operating procedures refuse to activate the HVPS as long as such condition is in effect (and not masked).

Regardless of whether the HVPS is activated automatically or by direct telecommand, the HVPS activation and de-activation complies with the following procedures:

4.4.6.1 HV rampup

The HVPS control operates the redundant HVPS power supplies that provide both the adjustable Mcp Voltage and the fixed Anode Voltage to the detector. The turn-on/set procedure is used to set the HVPS to a programmable value. The HVPSs can only be turned on when the Instrument is NOT in the SAFE state. The HVPS set point value can either originate from the parameter file (P_HV_LEVEL) or from the HV on command (Tc_ACTIVATE_HVPS) parameter.

1. HVPS can only be commanded to any level that is lower or equal to the maximum HV level as specified in the parameter file (P_HV_MAX_HVSET), otherwise the turn-on procedure is rejected.
2. The turn-on procedure automatically enforces a (slow) ramp-up of the HVPS voltage level (not a sudden step):
   - If the commanded voltage level is above the current set-point (voltage will be increased) then the voltage setting is incremented by a step size determined by the step fraction parameter, with a step time (P_HV_STEP_TIME) between each increment. The step parameter may either specify a fractional or a linear stepping algorithm.
   - If the commanded voltage level is below the current set-point (voltage will be decreased), then the voltage setting shall be commanded immediately.
3. Turn-off procedure takes precedence over the turn-on procedure, either when manually started by means of telecommand (Tc_DEACTIVATE_HVPS) or because of triggering of the safety check.

A configuration parameter allows for the selection of either one or both HVPSs to be activated. The control provides a gradual ramp-up of the high voltage to allow for read-back monitoring and checking (by the Safety module) during the ramp-up to prevent damage to the HVPSs. Although hardware provides full redundant HV level control for both HVPSs, the system always commands both HVPSs to the same level. The ramp-up is controlled by two configuration parameters adjusting the step time (1-255 seconds) and the step fraction that determines the rate at which the step approaches the requested set point (see example in **Figure** 13). This mechanism results in a gradual decreasing steps when the commanded set point is approached. The mechanism is very flexible and can support slow ramp up in hours in the initial phases but also a quick operational ramp up in half a minute when sufficient confidence is acquired.

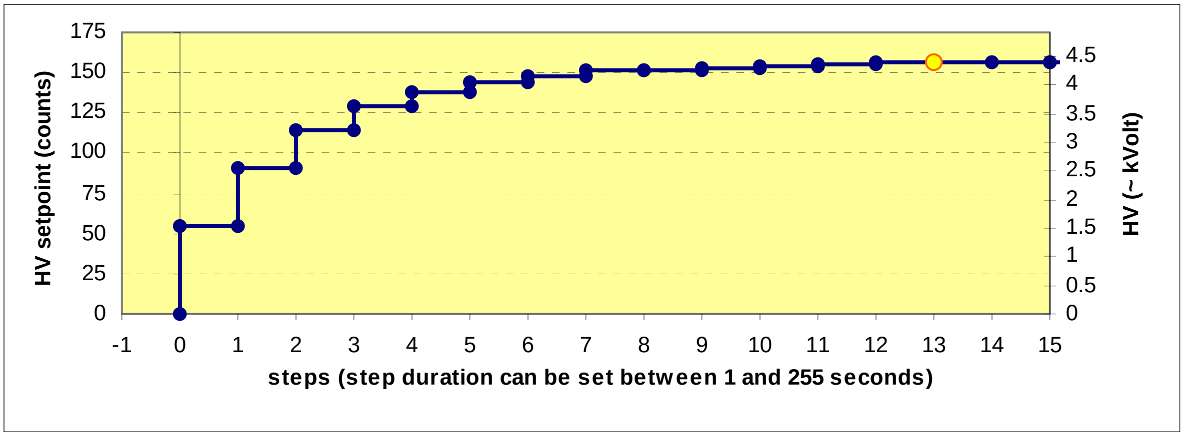

Figure 13: Example HV ramp-up from 0 to set point 157 (fraction = 45)

The algorithm below is used to determines the step size. The factor 16 in the step calculation is a constant pre-scaling of the HvFrac parameter used to obtain a useful adjustable range. In addition to this fractional control, the software also supports a linear ramp-up function; this mode is selected by specifying a value for HvFrac less than 16.

```
if (HvReq>HvSet) {          /* stepped increase   */
    if (HvFrac<16) {        /* -linear rampup     */
        step= HvFrac;
    } else {                /* -fractional rampup */
        step= [(HvReq-HvSet)*(16/HvFrac)];
        if (step == 0) step= 1;
    }
    HvSet= HvSet + step;
} else {                    /* immediate decrease */
    HvSet= HvReq;
}
```

In the linear ramp-up mode the HV rampup will just make equal size steps to the set point (the version 1.01 of the software has a problem though, when a step size larger than one is selected in the linear mode the endpoint may be overstepped (SPR-002)). This mode is considered a backup mode and the nominal operational mode is expected to be the fractional control.

4.4.7 Actuator firing

The PERSI-Alice instrument software controls the activation of all one-time actuators that are incorporated in the instrument. The controlled actuators consist of one redundant Wax Pellet Actuator (WPA) and three redundant Shape Metal Alloy (SMA) actuators. One of the redundant SMA actuator channels though is used to control the detector release valve. This detector release valve control uses the same control mechanism as the SMA although the used activation timing will probably be different.

Nominal control allows for activating either the primary or the redundant side of each actuator, although for contingency operations both sides of a single actuator may be activated.

The activation (firing) of the actuator and the selection of the activated side are directly controlled by a telecommand, but the actuators can only be activated when at the same time the appropriate actuator (heater) power circuit is activated by the spacecraft. Independent from the software an actuator safe plug is used to disable the actuator activation during ground testing. Activation of any of the actuators may be terminated by a separate telecommand option. The SMA and WPA have very different timing and control requirements; control functions for WPA and SMA will be described separately in the next two paragraphs.

The Alice instrument has five different one-time actuator mechanisms; partly these mechanisms are realized in a redundant way. The five actuator mechanisms operate the various protection mechanisms in the instrument:

- Detector door: one (redundant) WPA to open the door (once),
- Solar Occultation Channel door: one (redundant) SMA to open the door (once),
- Failsafe door: one SMA to open the failsafe door (once),
- Aperture door latch: one (redundant) SMA to unlatch the door (a separate actuator operates the door after unlatching see 4.4.5),
- Vent valve one actuator (solenoid) to open the detector vent valve (once).

4.4.7.1 SMA

The SOC door, failsafe door and aperture door latch are controlled by SMA actuators. For the SOC and aperture latch redundant mechanisms are available, the failsafe door has only a single actuator mechanism. The control of the SMA is directly time controlled; the control period is determined by a single shared configuration parameter (P_TINI_CONTROL), but during operations different values may be configured. This configuration parameter allows for a range of 5 to 1275 milliseconds of control (with a resolutionof 5 ms). Once activated the control remains activated for the defined control period. The required SMA activation time depends on the temperature, at room temperature a value of about 80 ms is specified.

4.4.7.2 Vent valve actuator

The detector housing vent valve actuator consists of a bi-stable solenoid. During the flight, only the open function can be commanded. This solenoid requires a fixed activation period just like the SMA actuators. The vent valve is actually wired in the system in place of the missing redundant fail-safe door actuator and uses the same control hardware and software. Therefore in some places this channel might still be referred to as the redundant fail-safe door. For a description of the control function see the section above. For the detector release valve a shorter activation time of about 30 ms seems to be sufficient.

4.4.7.3 WPA

The detector door actuator is a High Output Paraffin (HOP) actuator, also called Wax Pellet Actuator (WPA). In flight, the detector door can only be opened, not closed (during ground testing prior to launch, it can be manually re-closed). A wax pellet actuator with redundant bridge wires actuates the detector door opening mechanism. The detector door release mechanism has a sensor (switch) to indicate when the wax pellet actuator plunger has extended to a certain position and a separate sensor to indicate when the detector door is open. The software controls both bridge wires in the actuator and reads the status of both sensors. The WPA activation duration depends again on the environment temperature; in vacuum at room temperatures activation times of about 2 minutes have been observed.

The control of the WPA is also time limited but the activation period will also be terminated when the WPA feedback switch indicates that the active position is reached. This redundant scheme is used to prevent over-activating of the wax pellet actuator (and possible contamination). Note that the detector door position indicator is NOT used in the WPA control algorithm. The configuration parameter (P_WPA_TIMEOUT) in this case defines a timeout period after which the actuator is in any case switched off. The configuration parameter allows for a timeout period in the range of 10 to 2550 seconds (with a resolution of 10 seconds). A separate configuration parameter (P_WpaSensorEnable) allows for the disabling of the feedback switch; in that case, the control defaults to a purely time based control.

4.4.8 Heater control and temperature measurements

The instrument contains two sets of redundant decontamination heater on the OAP mirror and grating (four heaters in total). These heaters are used to raise the temperature of these two optical surfaces to clean them from contaminations. A control algorithm based on the temperatures measured on these optical surfaces is used to prevent overheating. In parallel to the OAP mirror heaters, two additional heaters are mounted on SOC the pickup mirror. No separate temperature control is implemented on these heaters although the temperature of the SOC is reported in the housekeeping data.

In nominal operations, this decontamination will be performed separate from any acquisition operations, although the software doesn't prevent heater activation concurrent with acquisition operations. But heater commanding (TC acceptance) can only occur when the instrument is in the checkout state.

The control of the heaters uses a very simple on-off control to prevent overheating; both OAP mirror and grating have their own control. Whenever the last measured temperature is below the set point, the selected heater(s) of the corresponding redundant heater pair will be activated. For this control, each of the heated surfaces is equipped with redundant temperature sensors. A configuration parameter determines whether the primary or the redundant sensor is used for the control algorithm (P_HtrSenseGrating and P_HtrSenseMirror).

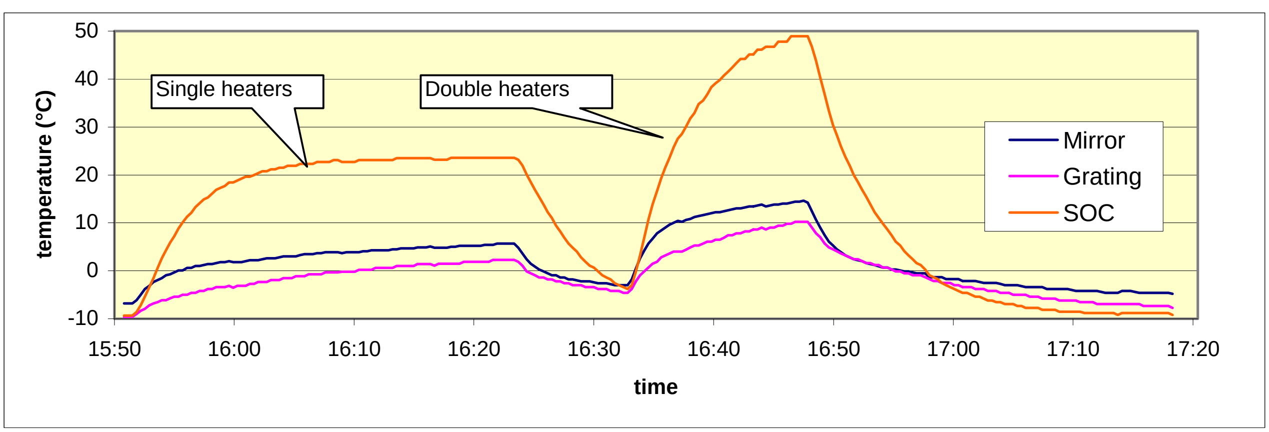

Figure 14: PERSI-Alice heater activation in vacuum (20040810)

To prevent excessive switching, the control algorithm is only activated every 8 measurements, so, at a maximum, one switching action (on or off) per heater can occur per 8 seconds. The control of both redundant heater pairs is staggered in time so the switching actions of both heater pairs can never occur at the same time. The selection (and activation) of either primary or redundant heaters (or both) for each surface is again based on configuration parameters (P_MirrorHtr1Enable, P_MirrorHtr2Enable, P_GratingHtr1Enable and P_GratingHtr2Enable), but also the activation of primary and/or redundant actuator but may be used for selection.

In a vacuum environment the temperature responses of the optical surfaces were measured and even though no thermal equilibrium was reached the general trends are clear. The first activation only activated the primary heaters on all optical surfaces; in the second test both the primary and redundant heaters on the three optical surfaces were activated. In Figure **14** it is clear that the SOC mirror has a smaller thermal mass and is better isolated from the environment as it changes temperature much faster and reaches a much larger temperature difference.

4.5 Acquisition processing

After commanding an acquisition start, the instrument will first go through an acquisition startup sequence that brings the instrument in the proper acquisition state. Most of the acquisition parameters are controlled by settings retrieved from the parameter file but two are defined by parameters of the acquisition start command. These startup actions were already presented in Table **2**. If the selected acquisition state was already commanded, this setup will hardly take any time but when door operations of high voltage rampup is involved the startup may take a considerable (deterministic) amount of time. During this startup the safety checking ensures that the high voltage activation does not endanger the instrument through monitoring of the safety parameters.

Next the actual acquisition is started; this can either be a pixellist or a histogram acquisition. A histogram acquisition may consist of multiple exposures of a duration defined in the parameter file that each result in the production of a separate science data frame. A pixellist acquisition stores all detected photon events together with time hack data. The time it takes for a pixellist acquisition to fill an acquisition buffer depends on the selected time hack rate and the number of photon events detected. The number of photon events detected depends on the brightness of the object and the detector area that is not masked with a hot pixel mask. Each time when an acquisition buffer is filled, a science data frame will be produced. Nominally the acquisition is terminated by a separate ‘stop acquisition’ telecommand. To prevent endless acquisition, the instrument implements an acquisition timeout function that terminates an acquisition automatically when the defined timeout period is expired. This timeout mechanism may also be used as the sole purpose of acquisition duration control. During acquisition the safety checking continuously ensures that the high voltage activation and does not endanger the instrument through monitoring of the safety parameters.

When an acquisition is terminated either by commanding the instrument directly to the SAFE state or by a safety event during the acquisition processing, the instrument will be brought in a safe condition. The HVPS will be disabled, heaters deactivated (if they were on) and the aperture door will be closed (if enabled and if one of the actuator busses was activated). When an acquisition is terminated by commanding the instrument to the CHECKOUT state or when the acquisition timeout was reached, the instrument remains in an active state and a new acquisition can be started immediately.

In addition to normal actuations the instrument also supports ‘test pattern’ acquisitions. These modes are available for interface verification. In these modes basically the normal acquisition sequence occurs but instead of producing actual acquired science data frames, pre-defined test patterns are generated and send to the spacecraft.

4.5.1 Acquisition Startup

The acquisition sequence starts after receiving the confirmation to an acquisition command. The startup sequence and the duration of the various steps are listed in Table 2. The acquisition start sequence can only start when the instrument is able to change to 'ACQUIRE' state, meaning that no safety condition is currently in effect (safety timeout equals zero). The start of an acquisition can either be commanded from the SAFE or from the CHECKOUT state. The difference between the two is that in the CHECKOUT state, the HVPS and/or the heaters could already be commanded on. This would for instance allow for a pre-verification of the stability of the MCP and anode voltages or an acquisition with heaters activated (used during decontamination operations). As a first step the acquisition hardware is configured for the acquisition by loading the acquisition (hardware) control registers with parameters from the parameter file (P_DISCRIMINATOR, P_StimEnable, P_PIXELLIST_HACK, P_HOTSEG_1-8).

The 'dooropen' parameter of the start acquisition command defines the commanded state of the aperture door. If the aperture door is already in the requested state (according to the door sensors), the commanding of the door will still take place but the startup sequence continues immediately. If the door was not in the requested state the door commanding will also take place but the startup sequence will pause until the door commanding is completed. Note that the door commanding period is a fixed period that is defined by a parameter in the parameter file (currently 1.8 sec). In this case the position reported by the door sensors is not taken into account anymore, the acquisition startup will continue after the defined commanding time irrespective of the (sensor) reported door position. These actions ensure that the door movements have completed before any acquisition activity is started. Note that the aperture door can only be moved when one of the actuator busses is activated. For the nominal histogram airglow observations, the Aperture door will be commanded open. For the pixel list occultation observations, the Aperture door may remain closed to prevent contamination of the occultation data by light entering the airglow aperture.

The next step is the high voltage activation. The HVPS activation always follows the rampup sequence as described in section 4.4.6. The acquisition startup only continues when the specified HVPS setting (as defined in the parameter file) is reached. This means that this step can be immediate when the HVPS was already commanded to the desired level, but the rampup may take a considerable period as defined in the HVPS rampup parameters in the parameter file. If during the HVPS rampup, a safety condition occurs the complete acquisition is terminated and the instrument returns to the safe state.

4.5.2 Acquisition processing

After these steps, the instrument is ready to start the acquisition. In order to start the acquisition at a deterministic point in time, the actual start of the acquisition processing is synchronized to the synchronization pulse (1PPS). The actual control of the acquisition hardware occurs about 2 ms after the instrument receives. At this point either the pixellist or the histogram acquisition mode is selected (hardware) and detector events are stored in the acquisition memory. For the pixellist acquisition, the acquisition memory actually fills; the detector events and time hacks are stored in a sequential list that fills the acquisition memory. The progress of this can be observed in the housekeeping data in the pixellist counter. For the histogram acquisition, the detector events are counted based on the spectral/spatial and amplitude 'position' of the event (histogramming). Then after a defined exposure time, the acquired image is transferred to the spacecraft.

In both cases, the instrument can perform continuous uninterrupted acquisitions. Meaning that there is no dead time in which the detector events are lost in between filling and switchover of an acquisition buffer. When the first buffer is filled, or an exposure is completed, acquisition continues in the second buffer. In parallel the first buffer is transferred to the spacecraft. The only case in which data could be lost is when the acquisition buffers fills faster than the data can be sent to the spacecraft.

In parallel to the acquisition operation by the instrument, the normal generation of housekeeping data continues providing ancillary data describing the conditions of the acquisition. One of the parameters reported in the housekeeping data is the measured raw countrate. This countrate value provides a temporal registration of the overall detected eventrate. This can be especially important for the histogram acquisition mode, as this mode normally does not include any temporal information.

4.5.2.1 Pixel List Mode

The pixel list mode is completely hardware controlled; after setup no further software actions are required, until the acquisition terminated by the software. The software determines the overall acquisition duration. Two acquisition buffers are used so acquisition in one buffer can continue while the contents of other buffer is transferred to the spacecraft. Switchover between the two acquisition buffers when a buffer is filled with pixel list data will be handled automatically by hardware; software will be informed when a buffer switchover occurs. At the end of the acquisitions, software terminates the acquisition mode, which will force transfer of the remaining buffer contents to the s/c (flush). Whenever the instrument generates a complete science frame, this will be reported in the housekeeping data; both the frame number of the last generated frame and the time (MET) at which the frame was generated are included in the housekeeping data. Note though that the generation of the science frame is data driven and that the acquisition buffer swap will not be synchronized to the spacecraft time. The reported frame time will be the current time, meaning the time (MET) that became current during the last time synchronization (1PPS).

During the acquisition, 16-bit words are written into the acquisition memory. The first word of the acquisition memory stores the frame identifier; so 32767 words are left to store the event and time hack data. Normal events in the pixel list memory are indicated by bit 15 of the word set to zero, the other 15 bits contain the spectral and spatial information of the event. Time hacks have bit 15 of the word set and the remaining 15 bits will be used to include a counter so timing correlation remains possible even when some data is missing (15 bits hack rate allows for 128 seconds of time information before the time hack counter wraps around). The most recent value of the time hack counter together with the science packet block counter will be included in the housekeeping data to allow time correlation.

The time it takes for a buffer to be filled with pixel list data depends on the actual count rate and the selected time hack rate. At the predicted highest event rate of 30 kHz and the highest time hack period of 4 ms, the 32-kword (32767 entries) pixel list memory could fill in 1.1 seconds. A more realistic maximum of the nominal maximum 10 kHz count rate at the highest time hack rate would result in filling the pixel list memory in 3.2 seconds. The transfer of a complete high-speed frame to the spacecraft will always takes 0.55 sec (see section 4.2.1.1). If the current transfer is not completed before a second acquisition buffer is filled the instrument temporarily can’t store (acquire) and some data will be lost. This case though will only happen during abnormally high event rates (above 58571 photon events per second), well outside normal operational range of the instrument.

The instrument generates frames on the high-speed data link when the acquisition memory is full and, hence, the transfer is inherent a burst transfer. Still an average datarate can be calculated. Based on the two above-mentioned cases, this results in average datarates of respectively 480 kbps and 164 kbps.

As mentioned before, it is possible that due to a very high event rate (>58571 events/second) the science transfers to the spacecraft can't keep up with acquisition and data will be lost. This can only occur in Pixel List mode, as here the frame generation rate is dependant on the observed event rate. In these cases, an interruption in the acquisition may occur. This interruption is not a reason to terminate the acquisition but it will be reported in the HK data stream (Hk_HSTM_OVFLW_ST)**.**

4.5.2.2 Histogram Mode

The instrument software controls the histogram mode exposure timing. The hardware performs the histogram operation in which events are counted. The raw pixel events will be histogrammed (counted) into the acquisition memory based on the spectral and spatial 'location'. The histogramming uses a read-increment-write operation to the addressed histogram memory (bin) location that corresponds to the spectral and spatial 'location'. So for each spectral-spatial 'location'; the number of events that were detected at that 'location' are counted. The increment operation will saturate at maximum count number, so when more than 65535 events are detected for a given location, the maximum number of counts will be reported. In addition, a second histogram operation will be performed on the event pulse height (amplitude) data. This results in what is called the pulse height data, a distribution of the amplitude of the detected events. This data will be stored in an otherwise unused area of the acquisition memory (beginning of second and third row of the array). This uses the same read-increment-write operation; this also has to use 16 bit 'event' counts (just like the main event histogramming); P-Alice uses 6 bits to represent the pulse height resulting in 64 pulse-height bins.

The histogram acquisition does not 'fill' the acquisition memory like the pixellist acquisition does. The counting of events could continue and even when some locations (pixels) are saturated there is no hard reason to terminate the acquisition operation. To provide a temporal resolution of the acquisition a single histogram acquisition may consist of multiple exposures. When an exposure is completed the instrument switches acquisition buffers and the acquisition continues in a fresh (cleared) acquisition buffer. The duration of histogram exposures is defined by a parameter file entry (P_HISTO_EXP_DUR), and can be selected anywhere between 1 second to 65535 seconds (~18 hours).

The selected exposure duration defines when science data frames are sent to the spacecraft; in all cases the frames are again 32768 words in size. The size of the data frames is independent of the number of detected events, as a frame always represents the full histogram covering the detector range of 1024 spectral lines * 32 spatial columns. Data transfers again occur in bursts, the 32 kword frame is transferred in about 0.55 sec (see section 4.2.1.1). An average high-speed science datarate can be calculated that depends on the selected exposure rate:

$$\textit{average data rate} = \frac{524288}{\textit{exposure duration (in seconds)}} \; \textit{(bps)}$$

So for reasonable exposure duration of 100 seconds the average data rate for the high-speed science data would only be 524 bps.

Again each time an acquisition buffer swap occurs, the value of the Last Acquisition Completion time in the HK telemetry will be updated; this provides information to perform post acquisition timing correlation. Note that in case of a histogram acquisition the exposure timing is synchronized to the 1PPS pulse.

This will also happen when the acquisition is terminated as described below. This provides a means to track the acquisition timing relative to the spacecraft elapsed time.

4.5.3 Acquisition termination

There are three reasons why an ongoing acquisition operation may be terminated:

- Commanded 'stop acquisition',
- Acquisition timeout,
- Safety event.

Whenever the acquisition termination occurs synchronized the actual hardware acquisition process will be terminated about 2 ms after the instrument receives the synchronization pulse (1PPS).

Nominally an acquisition will be terminated by an explicit command that commands the system out of the acquisition mode and thereby terminates the acquisition operation. The acquisition termination will, just like the acquisition start, be synchronized to the spacecraft synchronization signal (next 1 second pulse (real or assumed, see section 4.4.3.1)), so the exact time of acquisition completion is known. To stop an acquisition the instrument may be commanded either to the SAFE or CHECKOUT state. When switching to the SAFE state terminates an acquisition, this will also result in the deactivation of the HV and when enabled (P_DoorEnable), the closing of the Aperture door (when one of the actuator power busses was active). When commanding the instrument to the CHECKOUT state terminates the acquisition, no further actions are taken.

In addition to this (direct commanded), the instrument implements a timeout mechanism that will terminate an acquisition after a defined period of time (P_ACQ_TIMEOUT). This timeout prevents the instrument from generating an unbounded amount of science data in case the 'stop acquisition' command is missed. When the acquisition is terminated due to an acquisition timeout the system behaves as if a manual change to CHECKOUT mode was commanded. This means that the acquisition termination is synchronized (1PPS) and that no additional HV or state changes occur. The acquisition timeout period may be specified anywhere from 1 second to 65535 seconds (~18 hours). This timeout mechanism may also be used to command an acquisition of a predefined duration. Operationally this may facilitate commanding for acquisitions of pre-determined duration as no further commanding action is needed.

If during an acquisition a safety event occurs, the instrument will return to the 'SAFE' state and the ongoing acquisition will be terminated as quickly as possible. In this specific case the acquisition termination is not synchronized to the spacecraft time. This transition to the SAFE state terminates the acquisition, it will also result in the deactivation of the HV and when enabled (P_DoorEnable), the closing of the Aperture door (when one of the actuator power busses was active).

When the acquisition is terminated, the acquisition hardware will be commanded to stop the acquisition, and the contents of the current used acquisition buffer will be transferred to the spacecraft. In histogram and pixel list mode this means that the complete buffer contents are transferred, although in pixel list mode only a part of the memory will be filled with pixel list events (as indicated by a pointer in the housekeeping data (Hk_PIXEL_LIST_CNT)). In any case, this will also cause the Last Acquisition

Completion in the HK telemetry to be updated, indicating the time at which the last acquisition was completed.

4.5.4 Test mode acquisition

In addition to the normal acquisition modes, the instrument supports test acquisition modes. Test mode and pattern are selected by a parameter of the 'start acquisition' telecommand. In these modes no actual detector data is acquired, but the acquisition memory is filled with known (deterministic) test patterns. Except from filling the memory with these patterns the complete nominal acquisition sequence is performed, including set up of the environment, most important:

HV will be activated if the system is configured to do so,

Aperture door will be commanded if enabled and will move if one of the actuator power busses is active.

This means that the nominal acquisition timing is very similar to the actual acquisition operation. But instead of sending the acquired science data via the high-speed link the instrument will send the known test patterns. For the histogram acquisition the complete nominal exposure/acquisition timing is followed. For the pixel list acquisition a small adaptation had to be made; instead of waiting for the acquisition memory to be filled, the exposure duration parameter (P_HISTO_EXP_DUR) is now also used to determine when the pixel list science data transfer should be started (like for the histogram acquisitions).

The generation of the different test patterns takes some time depending on the selected test pattern (see Appendix E). This test pattern generation starts when the normal acquisition would start (after setting up the conditions in the nominal sequence). As long as the test pattern generation is not completed, the science transfer is simply skipped. When the pattern generation is completed before the acquisition command is terminated, the first science transfer will start on the following first scheduled exposure duration. This test pattern generation function may be used for the various test of the high-speed science data transfer. Different known test patterns can be generated that support both low level interface tests and higher level end-to-end test of the science data transfer mechanism. The test pattern generation function fills both acquisition memory pages with a known pattern. Test patterns include:

- Incrementing 16-bit number pattern starting at zero
- Decrementing 16-bit number pattern starting at zero
- Simple 16-bit constant bit-alternating pattern 0xa5c3
- Leave acquisition memory contents as is
- Histogram like data, based on some tables and constants, including pulse height distribution and STIM pulses if enabled
- Pixel list like data, based on some tables and constants, including simulated time hacks at a fixed rate.

Note that the first word from the acquisition memory is never read by the transmission hardware; instead of the first memory word the identification word is transmitted (see Appendix D).

4.5.4.1 Histogram Test Pattern

Fill the indicated detector memory with a simulated histogram test pattern based on a hard coded single line spectrum with a spectral range from location 98 to 916. The rows of the test pattern are generated by scaling the values from the single spectral line with a one byte-scaling factor, resulting in histogram pixel values in the range of 0 to 65025. The first two and the last two rows of the histogram have scaling factors of zero so these entries will be empty and simulate the dark count areas in the normal histogram. The test pattern generation also adds two simulated STIM entries to the array when the STIM function was active during the commanded test acquisition. The simulated STIM entries consist of normal distributed events around the nominal STIM locations with a mean of 0.2 and a deviation of 1.2. The function will also add a pseudo pulse height distribution (PHD). The PDH will be added to the start of the second and third histogram line and consists of a 8000*251 times a simple normal distribution with a mean of 23 and a deviation of 12 limited to a value of 64005.

4.5.4.2 Pixellist Test Pattern

The simulated pixel list pattern contains time hacks every 31 entries and the hack counter uses the current value of the time hack setting to control the increment of hacks. The generated pixellist consists of eight fixed event-generating locations:

(2,112), (24,112), (30,358), (14,420), (9,585), (5,716), (5,718) and (14,912)

The actual generated simulated events vary around these eight points though using a simple triangle shaped distribution in the spectral dimension. In the spatial dimension the simulated events occur on a single line. The generated events vary around the 8 locations in a fixed offset sequence:

0,1,-2,-1,0,1,2,-1,0,-2,1,-1,0,2,1,0,-1,-3,-2,1,2,-1,0,1,3,-1

The purpose of this test pattern is the verification of the end-to-end transfer and processing chain and this test pattern can be processed by GSE as a normal pixel list data set. The pixellist test pattern does not include any STIM related entries.

4.6 Instrument Safety and Monitoring

The safety-checking function is always active and checks a number of parameters for out of limit conditions and reports the results of these checks in the housekeeping data. When an unmasked safety event is detected safing actions will be performed and will cause an autonomous commanding of the instrument to the SAFE state and the associate safing actions. To prevent this autonomous safing, when mission critical observations are being performed (e.g. during the Pluto and Charon solar occultation observations), the safety checking can be disabled (masked) to prevent possible termination due to accidental detection of a safety condition.

4.6.1 Safety checking

The Alice instrument performs a number of different internal safety checks to protect the instrument from possible damage. These checks and actions are completely handled within the instrument to guarantee a quick response in case problems are detected. This monitoring uses various internally measured parameters to determine proper operation. The rate of the monitoring depends on the particular parameter checked, as different parameters become available at different rates.

There are three classes of monitored parameters for the internal safety checking:

HVPS parameters Mcp Voltage, Anode Voltage and Strip Current measured by the safety task every 100 ms

Detector count rate determined by the safety task every second

Temperatures measured and checked by the heater task every second

If a safety condition in any of these three classes is detected, the safety processing determines if any action to protect the instrument should be taken.

4.6.1.1 HVPS parameter monitoring

Whenever the instrument is powered on, the instrument safety checking function will monitor the performance of the HVPSs. The main task of this monitoring is to prevent any damage to the instrument due to a problem with HVPS or detector. The immediate action taken, when an out of limit condition is detected by the safety checking function, is to deactivate the HVPSs. During critical science operations this safing action may be disabled to prevent accidental deactivation of the HVPS (and thus termination of the acquisition operation).

The safety checking function measures the HVPS parameters of both of the HVPSs every 100 ms and checks the measured values against predefined limits (see Table 11). If any of the verified parameters are outside these limits for a specified (configurable) number of consecutive measurements, the safety mechanism is triggered. All the used limits are stored in the parameter file. This provides for standard defined limits after instrument power-up but still offers the possibility of in-flight adaptations, for instance to compensate for aging or changed operational plans. Note that as the HK telemetry packets are only generated once every second the actual measurements triggering the safety condition may not be included in the telemetry data, but the housekeeping data provides identification of the current safety conditions (Hk_LAST_SAFETY).

For all three HV parameter checks a separate 'occurrence counter' checks that an out-of-limits condition only triggers the safety processing when the condition exists for more than a specified number of consecutive measurement cycles. This prevents measurement noise from triggering the safety handling. For instance when a limit of 5 is defined, it will be triggered when for five consecutive measurements (= 5*100 ms = 0.5 sec), an out of limit condition, is detected. The limit values for the 3 separate occurrence counters are specified in the parameter table (P_HV_FAIL_MCP, P_HV_FAIL_STRIP and P_HV_FAIL_ANODE).

Table 11: High Voltage safety checking

<table>
<tr><th>Parameter</th><th>Check</th><th>Description</th></tr>
<tr><td>MCP Voltage
(largest read-back)</td><td>• HvSet ≤ Maximum Limit
• When commanded HV is above minimum:
| McpV - Expected HV | ≤ delta</td><td>Commanded High voltage not too large and matches commanded voltage</td></tr>
<tr><td>Strip Current
(sum of read-backs)</td><td>• StripI ≤ Maximum Limit</td><td>Strip current not too large</td></tr>
<tr><td>Anode Voltage
(largest read-back)</td><td>• AnodeV ≤ Maximum Limit
• When commanded HV is above minimum:
AnodeV ≥ Minimum Limit</td><td>Anode voltage within expected limits</td></tr>
<tr><td colspan="3">The parameter values for these limit checks are obtained from the parameter table. The following parameters are used:
• P_HV_LOW_SAFETY (Hv Setpoint above which HVPSs output reach nominal levels)
• P_HV_MAX_HVSET (Maximum allowed Hv setpoint)
• P_HV_MCP_TOL (Maximum Mcp voltage tolerance)
• P_HV_MAX_STRIPI
• P_HV_MIN_ANODEV and P_HV_MAX_ANODEV
• P_DAC_ADC_FACTOR (converts from Hv setpoint to expected Hv readback value)
Expected HV (readback) = HvSet * P_DAC_ADC_FACTOR</td></tr>
</table>

4.6.1.2 Count rate monitoring

A bright object is determined by a measured high detector count rate exceeding a defined threshold. This is monitored as a safety condition as too much UV-light on the detector can damage the instrument (detector). The count rate value is calculated every second from the 24 bit raw-event counter value. The difference of two of the periodic readings from this counter is calculated and limited to a maximum of 65535 (16 bit value) and reported in the housekeeping data (Hk_COUNT_RATE). The count rate threshold is determined by a parameter (P_MAX_COUNT_RATE) and can be adjusted as needed to accommodate background counts (such as due to a "hot spot"). Note that the count rate measurement comes from the detector hardware (raw event counter) before the "hot pixel region" filters are applied.

4.6.1.3 Temperature monitoring

Alice contains eight internal temperature sensors that are monitored at a one second rate. A sensor limit condition occurs if any monitored temperature sensor goes above a defined maximum safe limit. For each of the sensors, a separate maximum limit temperature is defined and stored in the parameter file (P_MAX_????_TEMP). If a sensor fails, the checking for that specific temperature channel can be disabled by masking the specific temperature check (P_TEMP_MASK), or by defining a maximum value for that specific limit check.

4.6.2 Safety processing

If any of the above mentioned safety conditions occurs, and the corresponding safety check is enabled, safety processing is triggered and the safety timeout timer is started (timeout period set to P_SAFETY_TIME). Also, if the safety override is not active, the instrument will immediately switch to the SAFE state. This transition to the SAFE state switches off the HVPSs, disables the heaters, terminates any actuator firing and, (when enabled (P_DoorEnable) when one of the actuator power busses is activated), closes the aperture door. During the safety timeout period the instrument will remain in the SAFE state (see Table 8). Thus, the software prevents any further operation or any actions that could damage the instrument (open door, high voltage on) until all of the safety conditions have disappeared for at least the timeout period.

Each of the safety checks can be individually masked by command, in which case that condition is ignored and is not used to determine the need for a shutdown. A general safety override is implemented which disables all safety actions and allows immediate operation of the instrument regardless of the safety condition(s) in effect. All safety condition information is included in the instrument housekeeping data (Hk_SAFETY_ACTIVE, Hk_LAST_SAFETY, Hk_SAFETY_TIMEOUT, Hk_SAFETY_ST, Hk_SAFETY_OVRD and Hk_SAFEMASK).

When a safety condition occurs during an acquisition, already acquired science data will not be lost. This case is handled as if a normal acquisition termination had occurred. Although the acquisition duration will be cut short (the actual acquisition termination time is reported in the housekeeping data (Hk_LAST_ACQ_DONE_TIME)), the already acquired data is not lost and will still be sent to the spacecraft.

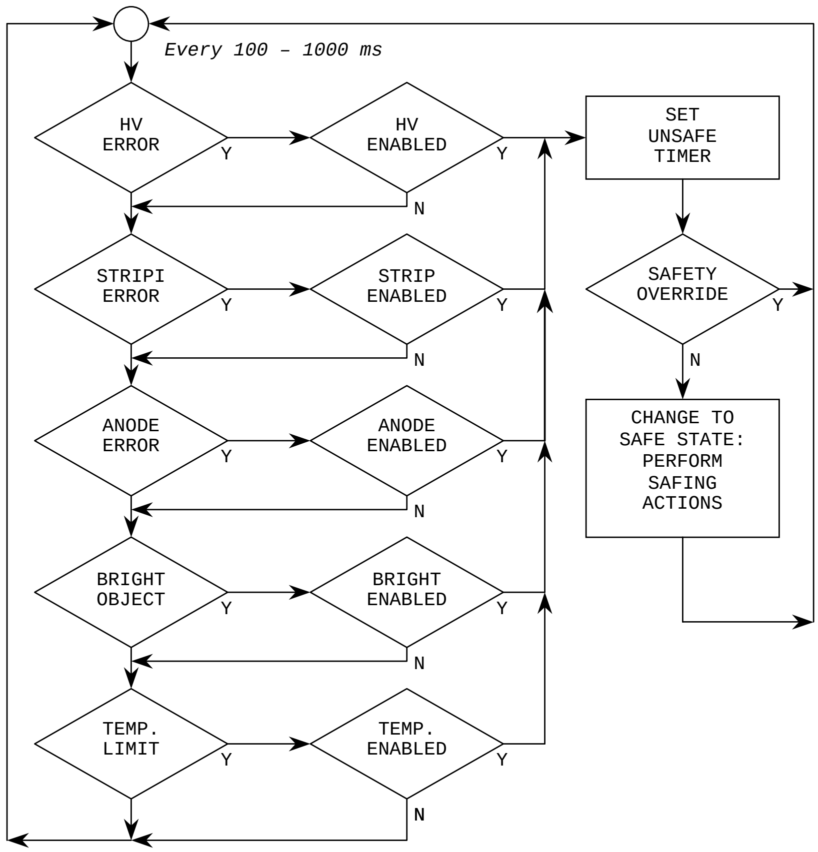

Figure 15: Safety Monitoring Algorithm Flowchart

Figure 15 shows a flowchart to demonstrate and clarify the operation of the safety algorithm. If any one of the safety conditions is true and it is enabled (not masked), then the "unsafe" timer will be reset. And, unless the safety override is in effect, actions will be taken to safe the instrument. The safing actions will command the instrument to the SAFE state and, therefore, include the following actions:

1. Deactivating the HVPSs,
2. Deactivate Actuator control,
3. Deactivating both decontamination heaters,
4. If enabled (P_DoorEnable) and if at least one of the actuator power busses is activated: close the aperture door.

4.6.3 WatchDog Timer

The onboard hardware includes a watchdog mechanism. The watchdog system is meant to catch catastrophic failures in the PERSI-Alice Flight Software and, therefore, the software cannot disable this function. When the experiment on-board software does not trigger the hardware watchdog mechanism within a 4 second period, a hardware reset is issued and the on-board software restarts. Writing a specific byte pattern to the watchdog reset register performs this triggering. This mechanism prevents hanging software from permanently missing any control deadlines. The hardware reset ensures that the processor including all surrounding hardware circuitry is completely reset and returns to a (known) power-up state. This full reset should be enough to clear any specific software hang-up. The hardware reset will cause the HVPS to be turned off immediately, but no (hardware) command is issued to the aperture door control so the door may remain open.

The hardware provides a mechanism that counts the number of watchdog resets since the last power cycle. After multiple resets, this counter will eventually initiate the hardware only ‘state machine’ acquisition mode (see section 3.3.2). To support system test, the value of this watchdog expiration counter is reported in the housekeeping packet. Also a single special action is taken in order to use the spacecraft autonomy rules as a backup mechanism to prevent unintended ‘limp along’ mode activation. Whenever the number of watchdog resets exceeds 4 (fixed number, not a parameter, to keep the code and checking as simple as possible), the software will request a power cycle by the spacecraft by setting the ‘power-off’ request flag in the non-packetized telemetry. Explicit disabling of the corresponding spacecraft autonomy rule still allows for intentional overriding of this mechanism.

4.7 Memory functions

The instrument supports the three memory function telecommands that can interact with the various memory types in the C&DH

- Memory Check – calculates a checksum over a specified block of memory
- Memory Load – load a specified block of data into specified memory
- Memory Dump – dump a specified block of memory

Some memory functions are limited in that they don’t apply to all memory types.

The memory check function calculates a byte wise 16-bit xor rotate checksum over a specified block of memory. The result of the checksum calculation is reported in the housekeeping packet (Hk_MEM_CHECKSUM). At the start of the calculation, the result is set to zero and the reported value is not updated until the calculation is complete. At instrument startup automatically a checksum calculation is performed over the complete executing code and the result is reported in the housekeeping packet. The checksum function is used to verify the contents of an area of memory, for instance to verify the result of a load operation or as part of the periodic instrument checkout to check that the memory contents was not corrupted. The various checks performed by the instrument on the contents of the code memory (both at instrument startup and during execution of the code selftest function) use the same checksum algorithm. The result of the checksum calculation over the full code memory area (0-0x7f7d) is for each code page stored in the 16-bit word at code memory address 0x7f7e. The memory check function may be performed on all instrument memory types.

The memory load function copies the received data bytes directly into the specified memory. For the RAM based memory types this consists of a simple write to the indicated location with an appropriate setting of the page selection for the acquisition memory (only the inactive side of the acquisition memory can be written to). For the EEPROM memory, some additional preparations are needed to enable the write operation. This function uses the EEPROM block write mode, so only a single wait is needed to wait for the device internal EEPROM programming to complete (see 4.7.2). After completing any of the memory load functions, the just modified memory is read back and the two blocks are compared. Only when the two blocks are equal is a successful completion status returned, otherwise an error status will result. If the load function is used to load data to a memory area where no simple memory exists (e.g.

special function registers) write and read may not result in the same data, which would result in an error message, although the action was actually successful. Memory load can be performed in all memory types except for the PROM memory that has a fixed content and can't be modified during flight.

The memory dump function reads the contents of the memory area and sends the contents of this memory to the spacecraft in the form of memory dump packets. A single memory dump command may generate multiple memory dump packets as each packet can only contain up to 128 bytes of data and a dump request could request much more data. Memory dump packets are only generated when the 'dump allowed' flag is set (included in the time synchronization message, see 4.3.1). When no memory dump packets are accepted, the memory dump function waits until the spacecraft again accepts packets. The memory dump function generates packets on the fly, but the software can easily generate packets with 128 bytes of data within one second. This means that a large memory dump operation can generate data sufficiently fast to add a memory dump packet to each generated ITF frame. The last generated packet of a memory dump sequence may contain less than the full 128 data bytes, but the packet is still a full size memory dump packet.

4.7.1 EEPROM write

The EEPROM memory provides a non-volatile storage function that can be modified during flight. It is used in the instrument both to store the default values of parameter list and to enable (semi-permanent) updates of the instrument flight code. The hardware provides a paging mechanism that allows one of the four defined EEPROM pages to be moved into the code space ands replace the PROM memory that normally resides in that area. The four EEPROM pages all are a full 32 kbyte in size and can contain the complete flight software. At launch, the plan is to have all four EEPROM pages contain the same version flight software that is contained in the PROM memory as a backup.

The write function on the EEPROM memory always uses the memory in the block write mode. This limits the number of write cycles needed to change the contents of the memory. The write EEPROM function includes an appropriate wait period (10s of milliseconds) to wait until the internal device programming cycle is completed. During this period data read from any location of the EEPROM is not stable so this write function can only be called when code is being executed either from PROM (or RAM). The software checks for this and will not allow EEPROM writes when this constraint is not met.

4.7.2 EEPROM Limited life issues

The used EEPROM devices are only guaranteed for a limited number of write cycles. The devices are used in the block write mode which allows for writing up to 128 bytes in a single operation. When the devices are used in this block write mode the devices are guaranteed for up to 10000 write cycles per block including the appropriate derating factors. For write operations of the parameter list to the non-volatile parameter storage in EEPROM, the number of write cycles used is automatically maintained and stored within the EEPROM (specified location in the parameter file, see Appendix F). For code patches and updates, there is no automatic mechanism that counts the number of used EERPOM write cycles and a manual administration will be needed.

Another issue is the guaranteed data retention time of the EEPROM devices that is limited to 10 years (again after appropriate derating). The EEPROM memory in the system is used both for the storage of patched code (up to 4 complete pages) and for the parameter file (3 redundant copies). Failure of either of these functions is not directly fatal to the instrument operations. It would either reduce the provided redundancy (no 'permanent' code patches, although execution from RAM is still possible), or require some additional commanding (no parameter storage).

The allowed number of write cycles is not expected to be an issue for the EEPROM code storage. Even for a mission duration of 17 years, this number would allow for more than one update every day. Still operational procedures will have to include a tracking of the number of commanded programming cycles (both pre- and post-launch). The data retention limitation can be handled by re-programming the devices at a certain point(s) in the mission. This reprogramming can be the result of a code patch operation (see section 4.7.4) or when the original PROM code is still in use; the PROM code may be used to 'refresh' the EEPROM contents. The system includes a special procedure to copy the PROM contents to all four EEPROM pages (see section 4.7.3).

The non-volatile version of the parameter file is expected to be updated more frequently, but still the number of allowed write cycles is expected to be much larger than the number of modifications expected during the mission. Temporary changes of parameter values only affect the RAM based (current working) version of the table and, thus, don't count to the total number of write cycles. The parameter file will only be written to EEPROM when the system receives and explicit commanded to do so. To facilitate tracking of the number of write cycles a counter is maintained within the parameter file (see section 4.4.4) registering the total number of Parameter File write cycles. Nominal operations are expected to include a number of parameter file adjustments to include the results of the in-flight measurements in the acquisition parameters; also nominal operations are expected to include the setting of acquisition duration parameters and writing these into non-volatile storage. This expected nominal usage should satisfy the 'refresh' requirements to guarantee proper data retention. In addition, the triple redundant storage (see section 4.4.4) of the parameter file would provide an indication when problems start to develop.

4.7.3 Duplicate PROM

The instrument software includes a special function that can copy the contents of the PROM memory to the four EEPROM pages. This procedure was used to prepare the system for the launch configuration in which all four EERPOM pages will contain an identical copy of the (PROM) flight software. This function is not part of the nominal processing system started via the start program function at address 0x252b. Copying will take about 250 sec; during this period, the normal instrument processing is temporary suspended. This means that for this period no telemetry will be generated and no telecommands (including time messages) will be accepted. If telecommands (including time messages) are send during that period they will probably overflow the input hardware FIFO and will probably not be recognized.

4.7.4 Patch upload

Software patches may be uploaded using the load memory telecommand. Four EEPROM pages are available that can each store a complete version of the instrument software. A full software version may be uploaded in any of these pages.

The updated software can be delivered to the ground operation system in the form of an S-Record. Special conversion software has been created that will then converts this single S-record into a sequence of 255 memory load commands and corresponding confirm criticals. Each of these memory load commands will load 128 bytes into the EEPROM memory. The generated sequence will include small delays in between the commands to ensure that the system accepts all the telecommands and the check that ensures that not more than one telecommand per second is sent. As the spacecraft data handling software only implements a single buffer for this telemetry going to the PERSI-Alice instrument, the commands all have to be spaced at least 2 seconds apart. This is not a limitation caused by the Alice instrument; the processing of the memory load commands can easily be completed at a 1 Hz rate. Even with these 2 seconds delays though the complete upload of a full updated software version only will take 255 * 2 * 2 seconds (~17 minutes).

4.8 Internal Selftest

Instead of having a build-in test that is always executed on system startup, the instrument software includes a self-test that can be activated by a separate telecommand. The test executed at system startup is limited to a quick code memory check to verify that the core image is correct (see section 4.1.1). This check makes sense as a backup for the PROM code storage is available that may be activated to correct the problem. This results in a quick startup and provides for an extended test when required. In any case, failures detected by the self-test can't automatically result in corrective actions. The single string nature of the instrument (C&DH) makes automatic recovery impracticable and in case of problems ground intervention is required (as some workaround may still exist). The internal self-test consists of a number of tests that can be performed within the environment of the executing real-time kernel. If more extensive test functions are required to diagnose a problem, the appropriate test functions have to be separately loaded either in RAM or EEPROM memory and can be executed through the 'start program' telecommand. Such extended test procedures need not be limited to running in the environment of the real-time kernel.

The execution of the different tests within the self-test function is controlled by the parameter of the telecommand which functions as a bit-field selecting the various tests. The results of the tests are reported in the housekeeping packet (Hk_TEST_STATUS). When the self-test command execution is started, all bits in this result byte will be set; when any of the steps completes without a failure, the corresponding bit in the result word will be cleared. All these tests are deterministic tests so beforehand the expected execution duration will be known for each of the steps.

Table 12: Self-test functions

Testcode	Test	Duration
0x01	Acquisition memory page 0	32 sec
0x02	Acquisition memory page 1	32 sec
0x04	Acquisition memory page window switching	7 sec
0x08	Free (unused) data RAM space	48 sec
0x10	Currently executing Code memory (checksum)	29 sec
0x20	Redundant parameters files (EEPROM)	<1 sec
0x40	EEPROM code memory (even pages, checksum)	48 sec
0x80	EEPROM code memory (odd pages, checksum)	48 sec

The acquisition memory test functions verify the memory within a given page by filling the complete page with a changing 16-bit pattern and then verifying the contents of all memory locations. The acquisition memory window switch test function verifies the paging mechanism by filling the first 1000 words on each page with a changing pattern that depends on the selected page and then verifying the contents of those 1000 locations on each page.

The RAM memory test can within the executing real time kernel only test the unused RAM memory. Given the small memory footprint of the application software, this means that still 86% of the available general purpose RAM can be verified, so this should provide a fairly good indication about the status of the RAM memory. The test fills the RAM memory with a 7 byte long pattern, first of a shifting 'one' and then of a shifting 'zero', in between and at the end the contents of the memory is again verified. An odd pattern length was selected so the chances of missing a stuck data or address line are minimized. If a more extensive memory test or a test of the RAM used is required, a special purpose test function could be loaded in RAM before executing the test while the operational system is temporary suspended. In order to be able to restart the real-time kernel after completion of the test, the contents of the used RAM memory should be temporary saved in this case.

The code memory and the 4 EEPROM pages are checked by calculating the checksum over the code area (0-0x7f7d) and comparing these to the pre-calculated checksums that are stored in the code memory pages at address 0x7f7e. Normally this test will be successful as any update of the code in EEPROM should also include an update of the corresponding checksum (a full patch will include the update of the checksum). The parameter file test verifies the contents of the 3 redundant versions of the parameter file that is stored there and verifying that all three copies are identical.

Appendix A - Overview C&DH Electronics Hardware

As a reference an overview is provided of the C&DH Electronics hardware.

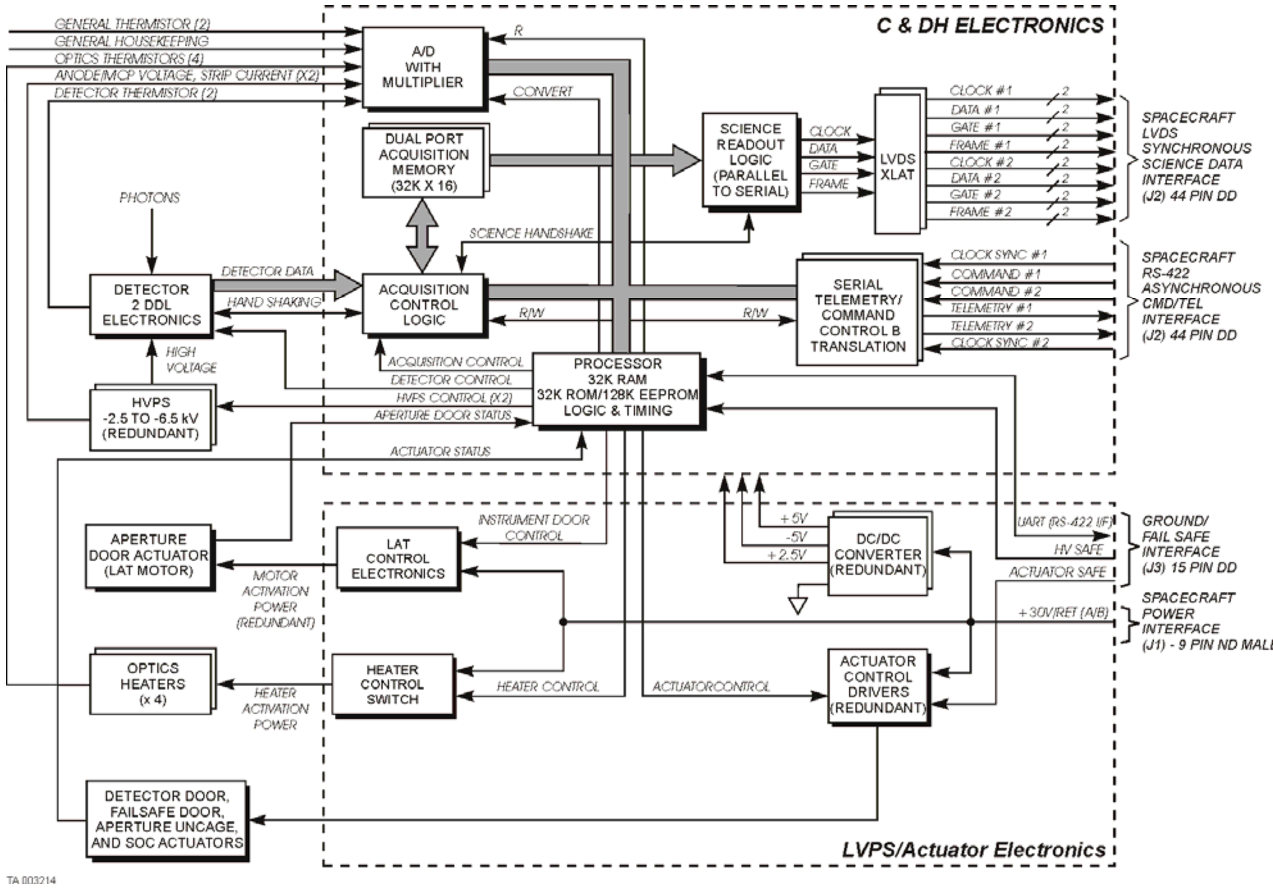

Figure 16: C&DH Electronics block diagram

Electronic block diagram notes:

- Acquisition memory consists of 2 pages each 32k x 16, one is accesses by the acquisition hardware, the other may after the Science readout be accessed by the processor,
- One of the general termistors is mounted on the SOC mirror,
- J2 consists of a single connector.

The power block diagram (Figure 17) shows the instrument configuration with the redundant instrument and actuator power busses. The redundant instrument power busses each power a separate Low Voltage Power Supply (LVPS), which is directly coupled to a corresponding High Voltage Power Supply (HVPS). The instrument detector and electronics consist of a single string. On the actuator side, the redundant actuator/heater power busses power the corresponding actuators, except for the Aperture door motor, this is a single string element that may be powered by either actuator power bus.

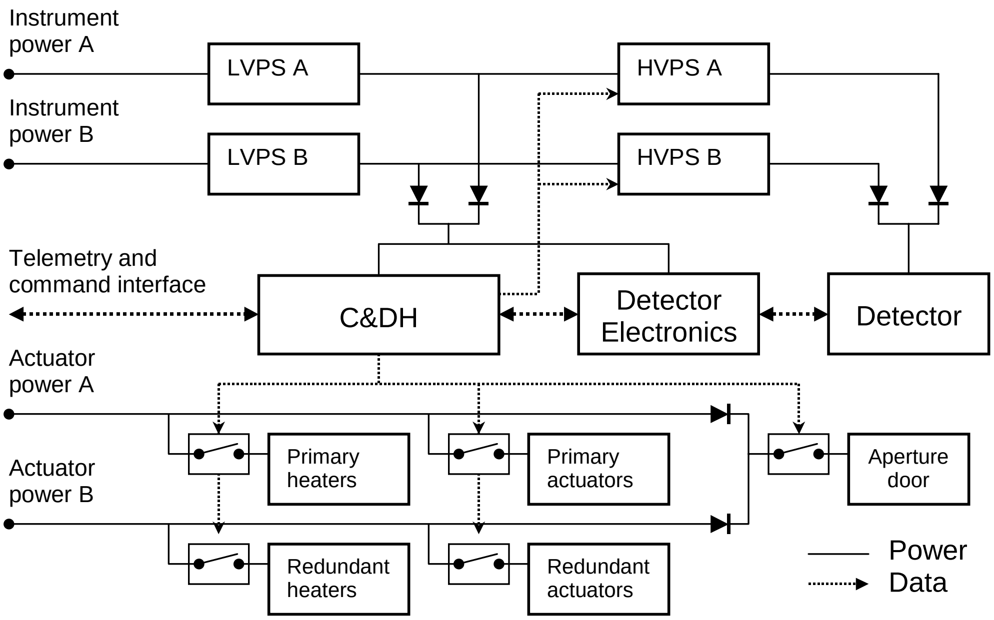

Figure 17: PERSI-Alice instrument power block diagram

Instrument power consumption

During integration testing power measurements on the FM model instrument were performed. The measurements are all based on a nominal bus voltage of 30 Volt, performed at room temperature.

Table 13: Instrument power consumption

	A-side only current	B-side only current	A+B side (total current)
Instrument power busses			
Instrument idle	116 mA	115 mA	132 mA
HVPS on	121 mA	120 mA	142 mA
Heaters on	124 mA	123 mA	148 mA
Actuator/heater power bus			
Mirror heater	87 mA	86 mA	174 mA
Grating heater	43 mA	42 mA	86 mA
Aperture door activated (< 2 sec)	60 mA	60 mA	60 mA
WPA activated	189 mA	190 mA	379 mA [1)]
SMA peak power (<100 ms)	630 mA	630 mA	1260 mA.[1)]
Valve peak power (<50 ms)	N/A[2)]	444 mA	N/A[2)]
1) possible but only during non-nominal operations 2) valve only connected to redundant power bus			

Processor memory map

The processor of the C&DH is compatible with the 8051 processor except that the internal memory has been extended from 128 bytes to 256 bytes. This processor has a separate data and code spaces. The following memory map shows how the processor accesses these external memory spaces and the different peripherals (the internal memory is not included in this memory map).

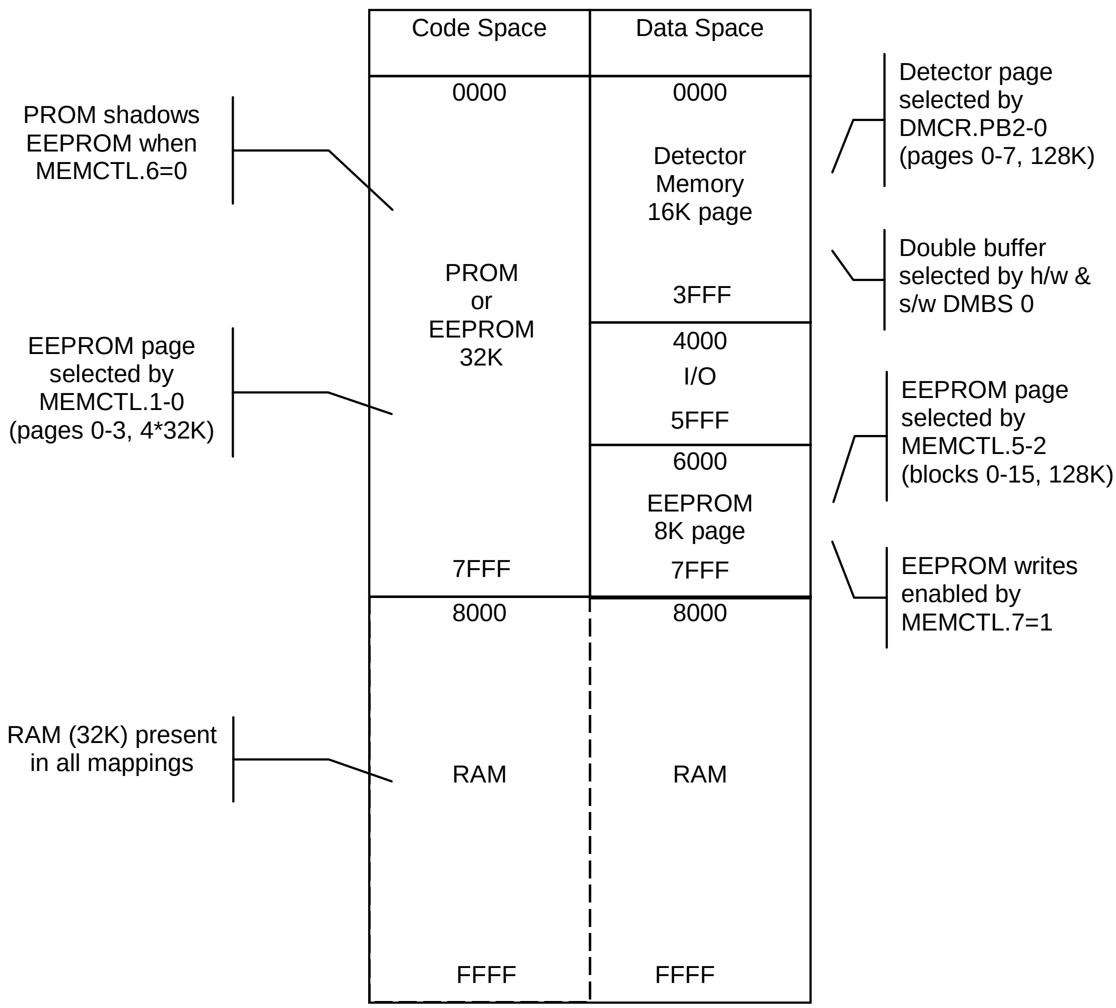

Figure 18: P-Alice C&DH processor memory map

Appendix B - Low Speed TM and TC Transfer Format

All low speed communication with the instrument uses ITF formatted (see Table 14) messages.

Table 14: Instrument Transfer Frame format

Byte #	Description	Value
0, 1 and 2	Frame Synchronization	0xfe, 0xfa, 0x30
3	Type of contained data:	0x01 - S/C time command 0x02 - telecommand 0x04 - telemetry
4	Checksum	Byte-wise XOR of all bytes following checksum field
5 – 6	Message Length	M - length of Message Data field in bytes
7 - 7+M	Message Data	Data (M bytes)

Table 15: Spacecraft time message data

Name	Size in bytes	Description
S/C Time	4	Time in seconds valid at the next 1-second (time sync) pulse
Memory dump Allowed	1	0x00 – Memory dump allowed 0x01 – Memory dump not allowed

Table 16: Telecommand Message format

Name	Size in bytes	Description
Op-code	2	Instrument defined field to specify the TC function, for the general memory load and dump functions s/c defined values are used
Macro field (1 bit)	2	Unused for Alice, set to zero
Command Word Count (15 bits)		Number of (32-bit) words in the command message including op-code, macro field, word count and checksum (= W)
Command Words	(W-2)*4	instrument defined command (parameter) words (32 bits)
Command Message (CM) checksum	4	32 bit exclusive-OR of all the words of the command message, not used / checked by P-Alice

The TC receiving task will perform a number of low level tests based on the ITF format, Alice will indicate any problems using the following error messages as reported in the Housekeeping Telemetry (LAST_FAIL_CODE). Note that these checks apply both to the Telecommand and Time Message ITF frames.

ITF format check error codes		
Errors:	0x01	ecINCORRECT_ITF_CHECKSUM_1
	0x02	ecINCORRECT_ITF_CHECKSUM_2
	0x03	ecUNSUPPORTED_ITF_MESSAGE_TYPE_1
	0x04	ecUNSUPPORTED_ITF_MESSAGE_TYPE_2
	0x05	ecTOO_LARGE_ITF_CMD_PACKAGE_1
	0x06	ecTOO_LARGE_ITF_CMD_PACKAGE_2
	0x07	ecTOO_SHORT_ITF_CMD_PACKAGE_1
	0x08	ecTOO_SHORT_ITF_CMD_PACKAGE_2
	0x09	ecINCORRECT_1ST_ITF_SYNC_BYTE_1
	0x0a	ecINCORRECT_1ST_ITF_SYNC_BYTE_2
	0x0b	ecINCORRECT_2ND_ITF_SYNC_BYTE_1
	0x0c	ecINCORRECT_2ND_ITF_SYNC_BYTE_2
	0x0d	ecINCORRECT_3RD_ITF_SYNC_BYTE_1
	0x0e	ecINCORRECT_3RD_ITF_SYNC_BYTE_2
	0x10	ecITF_ERROR_WAITING_1_2
	0x11	ecTC_BUFFER_OVERFLOW_1
	0x12	ecTC_BUFFER_OVERFLOW_2
	0x13	ecRX_FLUSH_FAILED_1
	0x14	ecRX_FLUSH_FAILED_2
	0x2c	ecTIME_MESSAGE_TOO_SHORT
	0x2d	ecTIME_MESSAGE_TOO_LONG

The TC receiving task only checks the match between the Telecommand ITF packet size and the number of command words as indicated in the Telecommand message format. Note that the Telecommand message format 32-bit checksum is not checked. Further checks apply to the specific telecommands and are listed in Appendix E.

Telecommand message format check errors		
Error:	0x22	ecWRONG_NUMBER_OF_COMMAND_WORDS

Appendix C - Low Speed Telemetry

The Alice non-packetized telemetry contents (capitalized Field Names listed in Table 17), are copied by the spacecraft in to the S/C HSK packet (APID 004). These fields can be accessed in that packet using the specified names, prefixed with 'CDH_ALICE_' and post fixed with '_004'. Note for bit-fields the number in brackets specifies the number of bits associated with the specific Telemetry field.

Table 17: P-Alice Telemetry data

Field Name	Size in bytes	Description
Standardized instrument status		
HEARTBEAT (1)	1	Alternating bit, changing state for each consecutive generated telemetry message [ICD 5.4], matches the least significant bit of the APID 482 sequence count
BOOT_APPL (1)		Fixed value 1: Alice is always executing application code when generating telemetry messages
TURNOFF_REQUEST (1)		Instrument requested turnoff: 1- instrument requests shutdown by s/c
STAT_SPARE (5)		Unassigned
Critical housekeeping		
CMD_ACC_8BIT	1	Modulo 2^8 count of commands accepted (started at zero at instrument startup)
CMD_REJ_8BIT	1	Modulo 2^8 count of commands rejected (started at zero at instrument startup)
Instrument defined status		Other non-packetized instrument status needed by the in the evaluation of autonomy rules
SAFETY_ACTIVE (1)	1	Safety or safety timeout active: 1- Safety (timeout) active
ACQUIRE_MODE (1)		Acquisition mode: 0-pixel list, 1-histogram, only meaningful when OPERATING_STATE=acquire
OPERATING_STATE (2)		Instrument operating state: 0-illegal (off), 1-checkout, 2-safe 3-acquire
RESTART_REQUEST (1)		Instrument requested restart: 1- request power cycle of instrument by s/c
WPA_DRIVEN (1)		Detector door actuator: 1- WPA activated (control signal)
MIRROR_HEATER_ST (1)		Mirror decontamination heater: 1- Mirror heater commanded on
GRATING_HEATER_ST (1)		Grating decontamination heater: 1- Grating heater commanded on
CURR_EXEC_CODE (3)	1	Current executing code: 1- PROM, 4- EEPROM_1, 5- EEPROM_2, 6- EEPROM_3, 7- EEPROM_4
LAST_SAFETY (3)		Last detected safety: 0- No safety, 1- Countrate, 2-McpV, 3- StripI, 4- AnodeV, 5- Temperature; cleared on transition out of safe state
APDOOR_ST (2)		Aperture door state: 0- Error, 1- Closed, 2- Open, 3- In between
COUNT_RATE	2	Instrument detected event countrate (Hz)
CMD_EXEC_CNT_8BIT	1	Modulo 2^8 count of commands executed (started at zero at instrument startup)
LAST_FAIL_CODE	1	Last reported failure code: 0x00, 0xff- Unused, 0xfe- No error, see Appendix G

Field Name	Size in bytes	Description
MAX_MCP_VOLT	1	Maximum High Voltage value measured in last second (for values see Table 18)
MAX_STRIP_CURR	1	Maximum measured total HV Strip current in last second (for values see Table 18)
CCSDS packetized information		
Offset to First Packet Header	2	Fixed value = 0. Designates that P-Alice always packs at least one complete embedded CCSDS packet in the ITF data field [ICD 5.2.1.7.2]
Housekeeping packet	96	Embedded CCSDS formatted Alice Housekeeping packet
Optional: **Memory Dump packet**	0 or 146	Embedded CCSDS formatted Memory dump packet only generated / included when explicitly requested by telecommand and allowed by S/C C&DH, one memory dump request may result in multiple packets being generated. The packet size is fixed to 128 data bytes
Total telemetry data size	109 or 255	

Numbers in brackets '()' indicate the size in bits of fields smaller than one byte. The TM fields in the Alice housekeeping packet (APID 482) excluding the fields in the header starting with PH and SH, can be accessed in the packet by prefixing the listed names with 'ALICE_HK.' (note the dot). For the header fields the full name is specified.

Table 18: P-Alice CCSDS Housekeeping TM packet

Parameter (size in bits where applicable)	Size in bytes	Description
Primary Header (CCSDS) [ICD 5.2.1.7.3]		
PH_VER_NUM_482 (3)	2	Version Number, fixed value = 0; designates a source packet
PH_PKT_TYP_482 (1)		Type Indicator, fixed value = 0; designates a telemetry packet
PH_SH_FLG_482 (1)		Secondary Header Flag, fixed value = 1; designates presence of secondary header
PH_APP_ID_482 (11)		Application Process identifier, fixed value = 0x482; designates Alice housekeeping packet [ICD 5.2.1.7.3.1]
PH_SEQ_FLG_482 (2)	2	Grouping Flags, fixed value = 3; single packet dump: each P-Alice housekeeping packet is completely self contained
PH_SEQ_CNT_482 (14)		Source Count, continuous sequence count of all generated CCSDS housekeeping packets since startup (modulo 16384), starting at zero
PH_PKT_LEN_482	2	Number of secondary header + data bytes (90) - 1 = 89 (fixed size Hk packet)

Parameter (size in bits where applicable)	Size in bytes	Description
Secondary Header (CCSDS)		
SH_TIME_482	4	Alice MET time will nominally be synchronized to the S/C time as send to Alice using the time synchronization message [ICD 5.2.1.7.3.3]
Data (CCSDS)		
General Instrument State		
SAFETY_ACTIVE (1)	1	Safety or safety timeout active: 1- Safety (timeout) active
ACQUIRE_MODE (1)		Acquisition mode: 0-pixel list, 1-histogram, only meaningful when OPERATING_STATE=acquire
OPERATING_STATE (2)		Instrument operating state: 0-illegal (off), 1-checkout, 2-safe 3-acquire
RESTART_REQUEST (1)		Instrument requested restart: 1- request power cycle of instrument by s/c, copied request status from actual request in non-packetized TM, see Table 17
WPA_DRIVEN (1)		Detector door actuator: 1- WPA activated (control signal)
MIRROR_HEATER_ST (1)		Mirror decontamination heater: 1- Mirror heater commanded on
GRATING_HEATER_ST (1)		Grating decontamination heater: 1- Grating heater commanded on
POWER_A_ST (1)	1	Instrument power status: 1 - Primary instrument power active
POWER_B_ST (1)		Instrument power status: 1 - Secondary instrument power active
TURNOFF_REQUEST (1)		Instrument requested turnoff: 1- instrument requests shutdown by s/c, copied request status from actual request in non-packetized TM, see Table 17
Spare (1)		unassigned
HVPS1_SAFE_ST (1)		HVPS 1 safing status: 1 - safing plug 1 installed
HVPS2_SAFE_ST (1)		HVPS 2 safing status: 1 - safing plug 2 installed
ACTR1_SAFE_ST (1)		Actuator 1 safing status: 1 - safety plug 1 installed
ACTR2_SAFE_ST (1)		Actuator 2 safing status: 1 - safety plug 2 installed
TC/TM Interface Status		
CMD_RECEIVED_ST (1)	1	Telecommand receive status: 1 - Command received during last second
SYNC_MSG_RECEIVED_ST (1)		Timesync message receive status: 1 - Valid Timesync message (ITF) received during last second
SYNC_PLS_RECEIVED_ST (1)		Timesync pulse receive status: 1 - Timesync pulse received during last second
CRIT_CMD_PENDING_ST (1)		Critical command pending: 0 - none, 1 - TC pending
MEM_DUMP_ALLOWED_ST (1)		Memory dump allowed status: 1 - memory dump allowed, reflects status received in most recent S/C time message

Parameter (size in bits where applicable)	Size in bytes	Description
TC_IF_STATUS (3)		Primary/redundant TC channel status 1 - waiting for first byte from either channel 2 - waiting for completion of primary TC 3 - waiting for completion of redundant TC 4 - waiting for first byte from primary channel 5 - waiting for first byte from redundant channel
CMDS_ACCEPTED	2	Modulo 2^{16} count of commands accepted [1)]
CMDS_REJECTED	2	Modulo 2^{16} count of commands rejected [1)]
CMDS_EXECUTED	2	Modulo 2^{16} count of commands executed [1)]
LAST_CMD_ACCEPTED	1	Id (low byte)of last accepted command
LAST_CMD_FAILED	1	Id (low byte) of last failed command
LAST_FAIL_CODE	1	Last failure code command/execution, see
CRIT_CMD_TIMEOUT	1	Remaining timeout period for a critical command
HSTM_HEADER	2	Header (word) of last generated science packet, bottom 12 bits are the block counter (see Table 20). Note that this word is not updated atomically, but counter and status bits change separately
Acquisition and Detector Status		
Spare (1)	1	-
DETDOOR_ST (1)		Detector door status (one time): 0 - open, 1 - not open
APDOOR_ST (2)		Aperture door status: 0 - error, 1 - closed, 2 - open, 3 - in between
Spare (1)		
WPA_SWITCH_ST (1)		Detector door actuator (WPA) stroke switch status: 1 - WPA engaged (monitor signal)
HVPS1_CMD_ST (1)		Commanded state of HVPS-1: 1 - on
HVPS2_CMD_ST (1)		Commanded state of HVPS-2: 1 - on
HACKRATE_ST (3)	1	Current selected time hack rate, per hardware register: 0= 4 ms; 1= 8 ms; …; 7= 512 ms
HSTM_OVFLW_ST (1)		High Speed Science TM: 1 - overflow occurred in high speed science transfer
HVPS1_REP_ST (1)		Reported status of HVPS-1: 1 - on
HVPS2_REP_ST (1)		Reported status of HVPS-2: 1 - on
CURRENT_ACQ_MEM (1)		H/W selected acquisition memory side: 0 - ping, 1 - pong
PIXEL_STIM_ST (1)		Pixel STIM command status: 0 - off, 1 - on
COUNT_RATE	2	Count rate per second (Hz), as measured during the last second
EVENT_CNT	3	Current value of the hardware detector event counter
TIME_HACK_CNT	2	Most recent value of the time hack counter always indicating units of 4 ms, value might vary slightly (jitter) as the hardware updates this counter multiple times per second
PIXEL_LIST_CNT	2	Most recent value of the h/w pixel list pointer (storage location of next pixellist event, indicates progress of pixel list acquisition)
EXPOSURE_TIMEOUT	2	Remaining current histogram exposure time in seconds (duration of histogram exposure based on value specified in parameter file)

Parameter (size in bits where applicable)	Size in bytes	Description
LAST_ACQ_DONE_TIME	4	Time of the last acquisition completion (ping pong buffer swap or end of acquisition)
ACQ_TIMEOUT	2	Remaining time in seconds of the acquisition timeout counter
HVPS_SET_VOLT	1	DAC counts (0 kV – 6 kV: 5 ~ 200 V, 157 ~ 4.5 kV)
MCP1_VOLT	1	ADC counts (almost equal to Hv DAC counts) [2)]
ANODE1_VOLT	1	ADC counts (190 ~ 600 Volt) [2)]
STRIP1_CURR	1	ADC counts (56 ~ 8 µA, 96 ~ 12 µA) [2)]
MCP2_VOLT	1	ADC counts (almost equal to Hv DAC counts) [2)]
ANODE2_VOLT	1	ADC counts (190 ~ 600 Volt) [2)]
STRIP2_CURR	1	ADC counts (56 ~ 8 µA, 96 ~ 12 µA) [2)]
MAX_MCP_VOLT	1	ADC counts (almost equal to Hv DAC counts), maximum Mcp voltage measured during the last second
MAX_STRIP_CURR	1	ADC counts (55 ~ 8 µA, 136 ~ 16 µA), maximum value of the combined strip current measured in the last second.
DISCRIMINATOR_VOLT	1	Discriminator read back, ADC counts (V = n*3 / 255)
Temperatures and heater set points		
MIRROR_SETPOINT_TEMP	1	Mirror heater control setpoint temperature, ADC counts (215 counts ~ 45 ºC)
GRATING_SETPOINT_TEMP	1	Grating heater control setpoint temperature, ADC counts (215 counts ~ 45 ºC)
MIRROR_A_TEMP	1	Primary mirror temperature, ADC counts (168 counts ~20 ºC, 181 counts ~25 ºC) [3)]
MIRROR_B_TEMP	1	Redundant mirror temperature, ADC counts (168 counts ~20 ºC, 181 counts ~25 ºC) [3)]
GRATING_A_TEMP	1	Primary grating temperature, ADC counts (168 counts ~20 ºC, 181 counts ~25 ºC) [3)]
GRATING_B_TEMP	1	Redundant grating temperature, ADC counts (168 counts ~20 ºC, 181 counts ~25 ºC) [3)]
DET_ELEC_TEMP	1	Detector electronics temperature, ADC counts (168 counts ~20 ºC, 181 counts ~25 ºC) [3)]
DET_HOUSE_TEMP	1	Detector housing temperature, ADC counts (168 counts ~20 ºC, 181 counts ~25 ºC) [3)]
CDH_TEMP	1	C&DH electronics temperature, ADC counts (168 counts ~20 ºC, 181 counts ~25 ºC) [3)]
SOC_TEMP	1	SOC mirror temperature, ADC counts (168 counts ~20 ºC, 181 counts ~25 ºC) [3)]
Safety status reporting		
SAFETY_TIMEOUT	2	Unsafe timer in Seconds, 0 means no safety active/pending
LAST_SAFETY (3)		Last activated safety: 0 - No safety, 1 - Countrate, 2 - McpV, 3 - StripI, 4 - AnodeV, 5 - Temperature; cleared on transition out of SAFE state
TEMP_SAFETY_ST (1)	1	Temperature safety: 1 - safety in effect
ANODE_SAFETY_ST (1)		Anode Voltage safety 1 - safety in effect
STRIP_SAFETY_ST (1)		Strip Current safety 1 - safety in effect
HV_SAFETY_ST (1)		HV safety 1 - safety in effect
BRIGHT_SAFETY_ST (1)		Bright object (countrate) safety: 1 - safety in effect

Parameter (size in bits where applicable)	Size in bytes	Description
SAFETY_OVRD (1)	1	Safety override flag: 1 - all safety handling is overridden (deactivated)
Spare (2)		-
TEMP_SAFEMASK (1)		Temperature safety mask: 1 - masked
ANODE_SAFEMASK (1)		Anode safety mask: 1 - masked
STRIP_SAFEMASK (1)		Strip safety mask: 1 - masked
HV_SAFEMASK (1)		HV safety mask: 1 - masked
BRIGHT_SAFEMASK (1)		Bright safety mask: 1 - masked
Diagnostic data		
CODE_ST(2)	1	Current executing code: 0,3 - Illegal, 1 - PROM, 2 - EEPROM
EEPROM_ST (2)		Current selected EEPROM code page, only meaningful when CODE_ST=2 (EEPROM): 0 - EEPROM page 1 1 - EEPROM page 2 2 - EEPROM page 3 3 - EEPROM page 4
HW_VERSION (4)		Board version id (0 - illegal, 1 - EM, 2 - SM, 3 - FM, 5 - SIM,…, 7 – PROM, …), read from parameter file, used to distinguish source of recorded TM data
SW_MAJOR (4)	1	Software major version number
SW_MINOR (4)		Software minor version / build number
RX_INT_A_OFF_ST (1)	1	Primary command channel interrupt disable: 1- disabled
RX_INT_B_OFF_ST (1)		Secondary command channel interrupt disable: 1- disabled
SYNC_A_ST (1)		Primary sync pulse received in last second (hw): 1 - 1PPS pulse received
SYNC_B_ST (1)		Secondary sync pulse received in last second (hw): 1 - 1PPS pulse received
FRAME_ERR_A (1)		Primary TC receiver h/w frame error status: 1- frame error detected
FRAME_ERR_B (1)		Secondary TC receiver h/w frame error status: 1- frame error detected
TC_OVRUN_A (1)		Primary TC receiver h/w overrun status: 1- overrun error detected
TC_OVRUN_B (1)		Secondary TC receiver h/w overrun status: 1- overrun error detected
MEM_CHECKSUM	2	Checksum calculated in response to the last issued check memory command, initially this reports the power on (executing code) checksum
PROC_IDLE	2	Count of passes though the (Rtx) idle loop, measure of processor idle time (last second)
PROC_SCHED	2	Count of calls to (Rtx) scheduler (last second)
TEST_STATUS	1	Self test status, each '0' bit indicates that the corresponding test completed successfully, a 1 indicates not selected or failed
TASK_#_STACK	10* 1	Value of stack pointer for all 10 active PAFS tasks: 0 - main task, 1 - safety monitor, 2 - TM gather task, 3 - HVPS control task, 4 - actuator fire task, 5 - aperture door task, 6 - TC interface task, 7 - TM interface task, 8 - heater task, 9 - slow task
MIN_STACK	1	Minimum amount of free stack space detected (for any task) during the last second

Parameter (size in bits where applicable)	Size in bytes	Description
FIRST_DELETED	1	Task number of the first deleted task (0-9); set to 10 if no deleted task exists
SLOW_TASK_STATE (3)	1	Slow task status: 0 - start, 1 - idle 2 - memory check 3 - memory dump 4 - memory load 5 - histrogram acquisition 6 - pixellist acquisition 7 - internal test
EX_MAXED_ST (1)		Watchdog expiration count above 15
EXPIRE_CNT (4)		Number of watchdog expirations (modulo 16) occurred since the last (power-on or spacecraft) instrument reset, used by hardware to determine when to start limp-along mode
FINE_RTC	1	Current value of the hardware fine clock register, counts time in 4 ms units (modulo 256)
Parameter value reporting		
PARAM_INDEX	1	Last Requested (parameter reported in PARAM_VALUE) parameter index
PARAM_VALUE	1	Current value of last requested parameter
Packet conclusion		
HK_CHECKSUM	2	HK Packet checksum, calculated checksum before instrument sends HK data to S/C
Total HK packet size	96	Fixed size CCSDS packet

1) Counters start counting from zero at instrument startup and will wrap around when the maximum value is reached
2) These analog parameters are measured at a higher rate (10 measurements per second) to support the instrument safety checking, the most recent measured value is reported
3) These analog parameters are measured at a lower rate (1 measurement per 10 seconds), one temperature is measured per second and reported as soon as the new value becomes available

The memory dump packets are meant for special fault diagnostics, and they will only be generated upon specific memory dump request in the form of the Memory Dump Telecommand. These packets are used to transfer the contents of the instrument on-board memory to the ground station. These transfers are included for contingency operations to verify the actual contents on the experiment on-board memory.

The generation of these packets is requested by the memory dump telecommand; it may be used to request the contents of all experiment on-board memory (PROM, acquisition memory, RAM and EEPROM pages 1 to 4). The size of the memory dump telemetry packets will always be 128 data bytes; if a smaller dump is requested still a full size dump packet will be generated. Multiple packets will be generated when a memory dump is requested that does not fit in a single dump packet. The format of the memory dump packet is described in Table 19; it follows the rules for a proper formatted CCSDS packet, and the standard described in the PERSI-ALICE ICD [ICD 5.2.1.7.4.1]. If present, each ITF telemetry frame will contain a second complete CCSDS packet, but a complete memory dump may consist of a series of CCSDS packets. As each of the packets includes a complete description of the area of memory that is included in the packet, no additional sequence identification is needed.

Table 19: P-Alice CCSDS memory dump packet

Parameter	Size in bytes	Description
Primary Header		
PH_VER_NUM_481 (3)	2	Version Number, fixed value = 0; designates a source packet
PH_PKT_TYP_481 (1)		Type Indicator, fixed value = 0; designates a telemetry packet
PH_SH_FLG_481 (1)		Secondary Header Flag, fixed value = 1; designates presence of secondary header
PH_APP_ID_481 (11)		Application Process identifier, fixed value = 0x481; designates Alice memory dump packet [ICD 5.2.1.7.3.1]
PH_SEQ_FLG_481 (2)	2	Grouping Flags, fixed value = 3; single packet dump: each P-Alice memory dump packet is completely self contained
PH_SEQ_CNT_481 (14)		Source Count, continuous sequence count of all generated CCSDS memory dump packets since startup (modulo 16384), starting at zero
PH_PKT_LEN_481	2	Number of bytes (secondary header + data bytes – 1): 139 (fixed size MD packet)
Secondary Header		
SH_TIME_481	4	Current Mission Elapsed Time (MET), Alice time will nominally be synchronized to the S/C time as send to Alice using the time synchronization message
Data		
START_ADDRESS	4	Start address of the dumped memory
BYTE_COUNT	2	Number of actual data bytes included (N = 1-128)
MEMORY_TYPE	1	Memory type contained in the dump packet: 0x50 = RAM 0x51 = EEPROM page 1 0x52 = EEPROM page 2 0x53 = EEPROM page 3 0x54 = EEPROM page 4 0x55 = Acquisition Memory (currently inactive page) 0x56 = PROM
Spare	1	Unused, set to 0
DATA_BLOCK_#	N	Data bytes (N = 1-128)
Filler	128-N	Filler bytes, set to zero, completing the packet to a fixed size
Total Memory packet size	146	Fixed size 146 bytes

Appendix D - High Speed Science Telemetry

Science telemetry consists of 'raw' science data packets (frames) of 32768 16-bit words in size. Science telemetry packets are generated by the acquisition hardware and sent to the spacecraft via the dedicated LVDS interface [ICD 5.3].

To identify the Science packets, a single 16–bit word header starts the packets. The header is generated by the acquisition hardware and includes the information listed in Table 20. The complete header word of the most recent generated science TM packet shall be included in the housekeeping TM packet to allow for correlation between the two data streams (low and high speed) after receiving the data by the ground segment.

Table 20: P-Alice Science packet telemetry header

Field	Size in bits	Description
Packet contents (msb)	1	0 - pixel list 1 - histogram
Memory	1	0 - ping (side A), 1 - pong (side B)
Last block	1	0 - intermediate block 1 - last block (acquisition cycle terminated)
HW acquisition	1	HW controlled acquisition (hardware limp along), overrides software control after 8 consecutive watchdog timeouts (see C&DH hardware documentation)
Block number	12	Least significant 12 bits of the block number

The contents of the remainder of the data packets depends on the acquisition mode:

- Pixel List — Each word (32767) in the remainder of the packet either describes a photon event or a time hack. This is indicated by bit 15. An event word encodes the location of the detected event consisting of a 10 bit encoded spectral location and a 5 bit encoded spatial location. The time hack is generated at a periodic basis (4 – 512 ms) and is used to provide temporal information about the photon events. In the remaining 15 bits, the time hack word contains an incrementing counter value to allow for data recovery in case of lost packets.
- Histogram — Each word (32767) in the remainder of the packet contains the number of photon events detected for that specific spectra/spatial location. This count value (histogrammed) saturates at a maximum value of 65535 to indicate a completely filled counting bin (meaning the counters do not wrap around). In the first four lines (1024 words) the first 32 words are excluded from this basic histogramming. The first row contains the header word and the second and third row contain the 64 words of pulse height information. This pulse height information is a similar list of counted (histogrammed) occurrences but now of the pulse height (event strength) of the separate events. Pulse height is available in 6-bit resolution resulting in 64 bins. These counters saturate in a similar way to a value of 65535.

For test purposes the instrument can fill the memory with a known (deterministic) pattern so the interfaces to the spacecraft and ground can be verified.

The spacecraft will package the P-Alice High Speed Telemetry data into CCSDS packages before sending the data to the ground. Different APIDs are used to distinguish the histogram and pixellist data packets see Table 21. Each CCSDS science packet can transfer a segment of up to 480 data bytes, in order to transfer a full P-Alice frame of 32768 words (16-bits), 137 science packets are needed, the first 136 packets will all be full size segments of 480 bytes, the last packet will transfer the remaining 256 bytes. The grouping flags of the packets indicate the start and end segment within a complete frame transfer.

Table 21: P-Alice science APIDs

APID	Identifier	Science data	Description
0x4B1	Alice_1	Pixellist frame	
0x4B3	Alice_2	Histogram frame	

Table 22: P-Alice CCSDS science packet

Parameter	Size in bytes	Description
Primary Header		
PH_VER_NUM_4B? (3)	2	Version Number, fixed value = 0; designates a source packet
PH_PKT_TYP_4B? (1)		Type Indicator, fixed value = 0; designates a telemetry packet
PH_SH_FLG_4B? (1)		Secondary Header Flag, fixed value = 1; designates presence of secondary header
PH_APP_ID_4B? (11)		Application Process identifier, see Table 21
PH_SEQ_FLG_4B? (2)	2	Grouping Flags: 1 – first segment 0 – intermediate segment 2 – last segment
PH_SEQ_CNT_4B? (14)		Source Count, continuous sequence count of all generated packets (per APID) (modulo 16384)
PH_PKT_LEN_4B?	2	Number of bytes (secondary header + data bytes – 1): 511 or 287
Secondary Header		
SH_PACKET_TIME_4B?	4	Spacecraft MET at time the Telemetry packet is constructed
SH_COLLECT_TIME_4B?	4	Spacecraft MET at time the high-speed science data was collected
Data		
ERROR_STATUS	8	Information from the SSR forward error correcting code, not important for simple decoding
SSR_HEADER	16	Information from the SSR storage administration, not important for simple decoding
DATA_BLOCK	480	Data bytes, all packets except the last packet 480 bytes, the last one is 256 bytes

Appendix E - Telecommands

PERSI-Alice operations are controlled by a set of 25 telecommands as listed in Table 23. In the remainder of this appendix, the functionality of these telecommands are described in detail, including the criteria that allow correct execution and the reported status. The instrument only accepts a number of these commands whenever the instrument is in the CHECKOUT state. To prevent inadvertent activation of a number of critical functions a critical command mechanism is implemented. For these so called 'critical commands' a separate confirmation command is needed before the execution of the actual command start. Note that this mechanism is completely separate from the critical command handling as implemented by the New Horizons ground system.

Table 23: P-Alice Telecommands

Op-code (hex)	Mnemonic	Telecommand	Critical
	General commands		
0x4101	ALICE_NOP	No Operation	N
0x4102	ALICE_ENTER_SAFE_STATE	Enter Safe State	N
0x4103	ALICE_ENTER_CHECKOUT_STATE	Enter Checkout State	N
0x4104	ALICE_CONFIRM_CRITICAL	Confirm Critical Command	N
0x4105	ALICE_START_HISTOGRAM	Start Histogram Acquisition	Y
0x4106	ALICE_START_PIXELLIST	Start Pixel List Acquisition	Y
0x4107	ALICE_SET_PARAMETER	Set Parameter	Y
0x4108	ALICE_STORE_PARAMETERS	Store Parameters	Y
0x4109	ALICE_LOAD_PARAMETERS	Load Parameters	N
0x410a	ALICE_ACTIVATE_PIXEL_STIM	Set Pixel Stimulator On	N
0x410b	ALICE_DEACTIVATE_PIXEL_STIM	Set Pixel Stimulator Off	N
0x410c	ALICE_SET_DISCRIMINATOR	Set Discriminator Level	N
0x410d	ALICE_CLOSE_APERTURE_DOOR	Close Aperture Door	N
0x410e	ALICE_DEACTIVATE_HVPS	Set High Voltage Off	N
	Only in Checkout state		
0x410f	ALICE_OPEN_APERTURE_DOOR	Open Aperture Door	Y
0x4110	ALICE_ACTIVATE_HVPS	Set High Voltage On	Y
0x4111	ALICE_CONTROL_HEATER	Control Heater	N
0x4112	ALICE_ACTIVATE_WPASMA	Activate WPA/SMA	Y
0x4113	ALICE_PERFORM_SELF_TEST	Perform Self Test	N
0x4116	ALICE_START_PROGRAM	Start Program	Y
0x4117	ALICE_REQUEST_RESTART	Request restart or power cycle	Y
0x4118	ALICE_RESET_TC_STATUS	Reset TC status	N
	Memory functions (only in checkout state)		
0x4119	ALICE_CHECK_MEMORY	Check Memory	N
0x0014	ALICE_LOAD_MEMORY	Load Memory	Y
0x0015	ALICE_DUMP_MEMORY	Dump Memory	N

After receiving a complete telecommand and passing the ITF frame checks (see Appendix B), a number of general checks are performed to verify consistency of the received command bytes and to verify that the instrument is in a proper state to receive the command. As a result of these checks a command may be rejected as indicated by incrementing the rejected command counter, which is reported in the Housekeeping data. In addition to this reporting an error code and if available a code indicating the offending command will also be reported in the housekeeping data (LAST_CMD_FAILED and LAST_FAIL_CODE). These general error codes are listed in the following table and are not repeated in the error codes related to the specific telecommands (the specific description will list 'none' if no command specific error codes are applicable).

General telecommand check error codes		
Errors:	0x20	ecIMPROPER_LENGTH_FOR_COMMAND
	0x21	ecUNKNOWN_UNRECOGNIZED_TELECOMMAND
	0x23	ecCOMMAND_REQUIRES_CHECKOUT_STATE
	0x24	ecALREADY_CRITICAL_COMMAND_PENDING
	0x25	ecINCORRECT_CONFIRM_CRITICAL
	0x26	ecMISSING_CONFIRM_CRITICAL
	0x27	ecNO_CRITCAL_COMMAND_PENDING
	0x28	ecCRITICAL_COMMAND_TIMEOUT

The remainder of this appendix provides a detailed description of all the PERSI-Alice telecommands; this includes a description of the command parameters and their effects and the criteria governing the acceptance or rejection and execution reporting related to these telecommands. The acceptance, rejection and execution of a telecommand are indicated by the corresponding 3 counters that are included in the PERSI-Alice housekeeping telemetry. A detected error is also reported in the housekeeping data in the filed 'last error' and the command that caused the error is included in the last failed command.

Mnemonic:	ALICE_NOP		
Command code:	0x4101	**Critical Command:**	No
Command size:	8 bytes	**Only in Checkout state:**	No
Description:	Command without any significant action used to verify TC path, command will always be accepted and executed successfully [ICD 5.2.1.8]		
	ident	**size**	**description**
Parameters:	none	-	-
Acceptance:	always		
Execution:	always		
Errors:	none		

Mnemonic:	ALICE_ENTER_SAFE_STATE		
Command code:	0x4102	**Critical Command:**	No
Command size:	8 bytes	**Only in Checkout state:**	No
Description:	Change to safe state. This includes deactivation of HV, deactivation of one time actuator control, deactivation of decontamination heaters and closing aperture door (if enabled). If an acquisition was active at the moment of this command this acquisition will be stopped at the next timesync pulse.		
	ident	**size**	**description**
Parameters:	none	-	-
Acceptance:	always		
Execution:	always		
Errors:	none		

Mnemonic:	ALICE_ENTER_CHECKOUT_STATE		
Command code:	0x4103	**Critical Command:**	No
Command size:	8 bytes	**Only in Checkout state:**	No
Description:	Change to checkout state. This is only possible when currently no safety condition is pending (or the safety override is active). If an acquisition was active at the moment of this command this acquisition will be stopped at the next timesync pulse.		
Parameters:	**ident** / none	**size** / -	**description** / -
Acceptance:	Telecommand only is accepted if the change to CHECKOUT state is completed or if the delayed change to CHECKOUT state is scheduled (when acquisition was in progress).		
Execution:	Change to the CHECKOUT state may fail if a safety condition is in effect (safety timeout > 0) and safety override is not active, command will not be accepted in that case either.		
Errors:	0x30 ecFAILED_TO_CHANGE_TO_CHECKOUT		

Mnemonic:	ALICE_CONFIRM_CRITICAL		
Command code:	0x4104	**Critical Command:**	No
Command size:	12 bytes	**Only in Checkout state:**	No
Description:	Critical commands need to be confirmed by the next command. This command must be a confirm critical command and it must be received within the timeout period. The fact that a critical command is pending will be indicated by housekeeping parameter 'CRIT_CMD_PENDING_ST'. The timeout period is determined by configuration parameter 'P_CMD_TIMEOUT' and the remaining timeout is reported in housekeeping parameter 'CRIT_CMD_TIMEOUT'. When a pending critical command times out an error code is generated, but the failing command is not reported.		
Parameters:	**ident** CONFIRMED_COMMAND unused	**size** 2 bytes 2 bytes	**description** Command code of command to be confirmed -
Acceptance:	A confirm critical command is accepted when a critical command is pending and when the parameter of the confirm command matches the command code of the pending critical command. After that the specific rules for acceptance of the confirmed (critical) command have to be satisfied (see separate (critical) command descriptions).		
Execution:	The confirm critical command itself is never executed but the command that is being confirmed will (re-)start execution when the confirm critical is received and this may lead to successful command completion. This execution is described in the separate (critical) command descriptions.		
Errors:	0x25 ecINCORRECT_CONFIRM_CRITICAL 0x26 ecMISSING_CONFIRM_CRITICAL 0x27 ecNO_CRITCAL_COMMAND_PENDING 0x28 ecCRITICAL_COMMAND_TIMEOUT		

Mnemonic:	ALICE_START_HISTOGRAM		
Command code:	0x4105	**Critical Command:**	Yes
Command size:	12 bytes	**Only in Checkout state:**	No

Description: Start acquisition in histogram mode, acquisition parameters are taken from internal parameter file, acquisition continues until commanded off (by commanding safe or checkout state) or until the parameter file defined timeout is reached. Test mode selection results in a deterministic test pattern instead of acquired data in the science TM packets.
In preparation for the acquisition the following actions are completed:

- Set discriminator, STIM and hot-segment registers to parameter file defined values.
- Command aperture door to position requested in the command parameter
- Command HVPS to the level indicated in the parameter file
- Clear the acquisition memory (unless test mode is selected)

Start of the actual acquisition is synchronized to the next sync pulse. During acquisition the time remaining until acquisition timeout and the remaining exposure timeout are reported in the housekeeping data.
When operating in test mode (MODE_SELECT>0) the normal actions and timing of the acquisition are maintained, but the science TM generation is skipped until the test pattern generation is completed.

Parameters:

ident	size	description
APDOOR_POSITION	1 byte	0 – Door Closed 1 – Door Open
MODE_SELECT	1 byte	0 – actual acquisition 1 – Incrementing number test pattern (~30 sec) 2 – Decrementing number test patter (~30 sec) 3 – Constant value (0xa5c3) test pattern (~23 sec) 4 – Leave acquisition memory as is (0 sec) 5 – pseudo histogram test pattern (~46 sec) 6 – pseudo pixel list test pattern (~40 sec)
-	2 bytes	unused

Acceptance: Always; command is stored when waiting for 'Confirm Critical'. 'Confirm Critical' is accepted when the acquisition start is accepted for execution by the 'slow' task when:

- Parameter values are within limits (door & mode)
- Slow task is available for execution (no acquisition, test or memory function active).
- Instrument can change to acquire state (no safety pending).

Execution: After correct confirmation and after reaching specified HV level when the actual acquisition is started or when a test mode acquisition is commanded and the generation of the specified test pattern is complete.

Errors:

0x51	ecNO_ACQ_START_WRONG_DOOR_PARAM
0x52	ecNO_ACQ_START_WRONG_TEST_PARAM
0x53	ecNO_ACQ_START_SLOW_TASK_BUSY
0x54	ecNO_ACQ_START_SAFETY_PENDING
0xa4	ecSTART_HISTO_ACQUISITION_FAILED
0xa2	ecCLEAR_HISTO_ACQUISITION_FAILED
0xa0	ecSYNCED_HISTO_ACQUISITION_FAILED
0xa6	ecACQUISITION_TIMEOUT_OCCURRED

Mnemonic:	ALICE_START_PIXELLIST		
Command code:	0x4106	**Critical Command:**	Yes
Command size:	12 bytes	**Only in Checkout state:**	No

Description: Start acquisition in pixel list mode, acquisition parameters are taken from internal parameter file, acquisition continues until commanded off (by commanding safe or checkout state) or until the parameter file defined timeout is reached. Test mode selection results in a deterministic test pattern instead of acquired data in the science TM packets. In preparation for the acquisition the following actions are completed:

- Set discriminator, STIM, hack timer and hot-segment registers to parameter file defined values.
- Command aperture door to position requested in the command parameter
- Command HVPS to the level indicated in the parameter file
- Clear the acquisition memory (unless test mode is selected)

Start of the actual acquisition is synchronized to the next sync pulse. During acquisition the time remaining until acquisition timeout is reported in the housekeeping data.
When operating in test mode (MODE_SELECT>0) the normal actions and timing of the acquisition are maintained, except that exposure duration parameter is used to initiate the science TM generation, also the science TM generation is skipped until the test pattern generation is completed.

Parameters:

ident	size	description
APDOOR_POSITION	1 byte	0 – Door Closed 1 – Door Open
MODE_SELECT	1 byte	0 – actual acquisition 1 – Incrementing number test pattern (~30 sec) 2 – Decrementing number test patter (~30 sec) 3 – Constant value (0xa5c3) test pattern (~23 sec) 4 – Leave acquisition memory as is (0 sec) 5 – pseudo histogram test pattern (~46 sec) 6 – pseudo pixel list test pattern (~40 sec)
-	2 bytes	unused

Acceptance: Always; command is stored when waiting for ‘Confirm Critical’. ‘Confirm Critical’ is accepted when the acquisition start is accepted for execution by the ‘slow’ task when:

- Parameter values are within limits (door & mode)
- Slow task is available for execution (no acquisition, test or memory function active).
- Instrument can change to acquire state (no safety pending).

Execution: After correct confirmation and after reaching specified HV level when the actual acquisition is started or when a test mode acquisition is commanded and the generation of the specified test pattern is complete.

Errors:

0x51	ecNO_ACQ_START_WRONG_DOOR_PARAM
0x52	ecNO_ACQ_START_WRONG_TEST_PARAM
0x53	ecNO_ACQ_START_SLOW_TASK_BUSY
0x54	ecNO_ACQ_START_SAFETY_PENDING
0xa5	ecSTART_PIXEL_ACQUISITION_FAILED
0xa3	ecCLEAR_PIXEL_ACQUISITION_FAILED
0xa1	ecSYNCED_PIXEL_ACQUISITION_FAILED
0xa6	ecACQUISITION_TIMEOUT_OCCURRED

Mnemonic:	ALICE_SET_PARAMETER		
Command code:	0x4107	**Critical Command:**	Yes
Command size:	12 bytes	**Only in Checkout state:**	No
Description:	Set specified internally stored parameter to specified byte value. As one of the parameters is the safety mask, this TC is also used to change the active safety mask. Also the parameter reported in the HK packet is specified itself as a parameter, so this is also changed using this command. The value of the parameter value cannot be checked, as the different parameter file entries have different valid ranges.		
Parameters:	**ident** / **size** / **description** PARAMETER_INDEX — 1 byte — Range: 0 – 58, see Appendix F PARAMETER_VALUE — 1 byte — Range 0x00 – 0xff, meaning dependant on the specific parameter value unused — 2 bytes — -		
Acceptance:	Direct; command is stored when waiting for 'Confirm Critical'. 'Confirm Critical' will be accepted when the parameter id is within the valid range.		
Execution:	After correct confirmation, when parameter id is within range and the specified value has been stored successfully.		
Errors:	0xb0 ecPARAMETER_INDEX_OUT_OF_RANGE 0xb1 ecPARAMETER_READBACK_FAILED		

Mnemonic:	ALICE_STORE_PARAMETERS		
Command code:	0x4108	**Critical Command:**	Yes
Command size:	8 bytes	**Only in Checkout state:**	No
Description:	Store parameter table into non-volatile EEPROM memory (all three copies), before performing the store operation the counter indicating the number of table write operations will be incremented. This command can only be executed when the instrument is executing code from PROM, hardware does not allow for an EEPROM program operation when executing from EEPROM (EEPROM reads are used to indicate programming progress during programming operations). When multiple storage failures (in different EEPROM pages) are detected only the last detected failure will be reported. For a storage error only the page is indicated, memory dump commands could be used to investigate this problem further.		
Parameters:	**ident** / **size** / **description** none — - — -		
Acceptance:	Direct; command is stored when waiting for 'Confirm Critical'. 'Confirm Critical' is accepted when the currently code from PROM is being executed.		
Execution:	After correct confirmation, when currently code from PROM is being executed, all the three store operations have been completed and the correct storage has been verified.		
Errors:	0xb5 ecSTORE_PARAMS_ONLY_WHEN_EXEC_PROM 0xb2 ecSTORE_PARAMS_READBACK_FAILED_1 0xb3 ecSTORE_PARAMS_READBACK_FAILED_2 0xb4 ecSTORE_PARAMS_READBACK_FAILED_3		

Mnemonic:	ALICE_LOAD_PARAMETERS		
Command code:	0x4109	**Critical Command:**	No
Command size:	12 bytes	**Only in Checkout state:**	No
Description:	Load parameter table from non-volatile memory into operational table, restores power-on condition, parameter allows for restore from all redundant stored tables using a majority vote, restore from a selected single table or initialize to a hardcoded default set. When the parameter file is loaded from a selected single table the instrument state will always change to SAFE. When the parameter file is loaded using majority voting from the 3 EEPROM stored tables the following actions are taken, for each of the byte parameters in the parameter file: – All 3 agree: Use the value for the parameter – Two agree: Use the value of the two tables that agree for the parameter value, report the problem – All 3 different: Command fails, don't change the parameter value, go to SAFE state, report the error		
	ident	**size**	**description**
Parameters:	TABLE_SOURCE	1 byte	0 – load from all three tables (majority vote) 1 – load only from table 1 2 – load only from table 2 3 – load only from table 3 17 – load from hardcoded (PROM) table, see values listed in Appendix F.
	unused	3 bytes	
Acceptance:	When a correct table source is specified		
Execution:	When a single table source is specified always, when a majority vote is requested only when no errors are detected.		
Errors:	0xb6 ecINCORRECT_TABLE_FOR_PARAM_LOAD 0xb7 ecLOAD_PARAM_TABLE_1_MISMATCH 0xb8 ecLOAD_PARAM_TABLE_2_MISMATCH 0xb9 ecLOAD_PARAM_TABLE_3_MISMATCH 0xba ecLOAD_PARAM_ALL_DIFFERENT		

Mnemonic:	ALICE_ACTIVATE_PIXEL_STIM		
Command code:	0x410A	**Critical Command:**	No
Command size:	8 bytes	**Only in Checkout state:**	No
Description:	Command stimulator on, the current commanded state of the pixel STIM is reported in housekeeping parameter 'PIXEL_STIM_ST'. The pixel STIM may either be activated by direct telecommand or as a side effect of the commanded start of an acquisition at which memory the parameter file entry 'P_StimEnable' determines whether STIM is commanded on or off. Executing this command will also set the entry 'P_StimEnable' in the active copy of the Parameter File, so a commanded acquisition start after any manual commanding will use the last commanded STIM state, but after an instrument reset the EEPROM value will be used.		
	ident	**Size**	**description**
Parameters:	none	-	-
Acceptance:	always		
Execution:	when hardware control is successfully activated		
Errors:	0x89 ecSTIM_NOT_ACTIVATED		

Mnemonic:	ALICE_ DEACTIVATE_PIXEL_STIM		
Command code:	0x410B	**Critical Command:**	No
Command size:	8 bytes	**Only in Checkout state:**	No
Description:	Command stimulator off, the current commanded state of the pixel STIM is reported in housekeeping parameter 'PIXEL_STIM_ST'. The pixel STIM may either be activated by direct telecommand or as a side effect of the commanded start of an acquisition at which memory the parameter file entry 'P_StimEnable' determines whether STIM is commanded on or off. Executing this command will also clear the entry 'P_StimEnable' in the active copy of the Parameter File, so a commanded acquisition start after any manual commanding will use the last commanded STIM state, but after an instrument reset the EEPROM value will be used..		
	ident	**size**	**description**
Parameters:	none	-	-
Acceptance:	always		
Execution:	when hardware control is successfully deactivated		
Errors:	0x88 ecSTIM_NOT_DEACTIVATED		

Mnemonic:	ALICE_SET_DISCRIMINATOR		
Command code:	0x410C	**Critical Command:**	No
Command size:	12 bytes	**Only in Checkout state:**	No
Description:	Set discriminator to the specified value. Executing this command will also set the entry 'P_DISCRIMINATOR' in the active copy of the Parameter File to the specified value, so a commanded acquisition start after any manual commanding will use the last commanded Discriminator level, but after an instrument reset the EEPROM value will be used.		
	ident	**size**	**description**
Parameters:	DISC_LEVEL	1 bytes	Level (1 byte)
	-	3 bytes	unused
Acceptance:	always		
Execution:	always		
Errors:	none		

Mnemonic:	ALICE_CLOSE_APERTURE_DOOR		
Command code:	0x410D	**Critical Command:**	No
Command size:	8 bytes	**Only in Checkout state:**	No
Description:	Command the aperture door to the closed position. Aperture door is always commanded to the requested close position for the full period specified in the parameter value (P_DOOR_CONTROL). At the end of the control period either execution completion or when the door sensors do not report closed state an error is reported.		
	ident	**size**	**description**
Parameters:	none	-	-
Acceptance:	Always, immediate after command processing started.		
Execution:	When door status reports closed position at end of control period.		
Errors:	0x48 ecCOMMANDED_DOOR_MOVE_FAILED		

Mnemonic:	ALICE_DEACTIVATE_HVPS		
Command code:	0x410E	**Critical Command:**	No
Command size:	8 bytes	**Only in Checkout state:**	No
Description:	Command HVPS off, first change the HV setpoint to zero and then disable the HVPS control. The command itself never fails but an error message may result when this command terminates a previous HVPS activation (rampup) command. Executing this command will also set the entry P_HV_LEVEL in the active copy of the Parameter File to zero, so a commanded acquisition start after any manual commanding will use the last commanded HV level, but after a reset the EEPROM value will again be used.		
	ident	**size**	**description**
Parameters:	none	-	-
Acceptance:	always		
Execution:	always		
Errors:	0x83 ecHVPS_OFF_BEFORE_RAMPUP_COMPLETE		

Mnemonic:	ALICE_OPEN_APERTURE_DOOR		
Command code:	0x410F	**Critical Command:**	Yes
Command size:	8 bytes	**Only in Checkout state:**	Yes
Description:	Command the aperture door to the open position. Aperture door is commanded to the requested open position for the full period specified in the parameter value (P_DOOR_CONTROL). At the end of the control period either execution completion or when the door sensors do not report open state an error is reported.		
	ident	**size**	**description**
Parameters:	none	-	-
Acceptance:	Direct; command is stored when waiting for 'Confirm Critical'. 'Confirm Critical' is accepted when the instrument is in CHECKOUT state.		
Execution:	After correct confirmation, when door status reports closed position at end of control period.		
Errors:	0x48 ecCOMMANDED_DOOR_MOVE_FAILED		

Mnemonic:	ALICE_ACTIVATE_HVPS		
Command code:	0x4110	**Critical Command:**	Yes
Command size:	12 bytes	**Only in Checkout state:**	Yes
Description:	Command the HVPS to the specified level. The HV is commanded to follow a slow rampup to prevent damage to the detector. The rampup timing is controlled by parameter file values (P_HV_STEP_SIZE and P_HV_STEP_TIME). Decreases to a lower HV level are executed immediately. Steps to a higher HV level are executed using a fraction of the remaining difference between current value and commanded value. This means that initially larger steps will be taken but at the end of the ramp-up small steps are used. After each step the ramp up waits the specified time (P_HV_STEP_TIME) to let the HV settle. During the ramp-up the safety checking is active (as always) and the HV parameters will be checked according to the safety checks every 100 ms. Executing this command will also set the entry P_HV_LEVEL in the active copy of the Parameter File to the specified level, so a commanded acquisition start after any manual commanding will use the last commanded HV level, but after a reset the EEPROM value will be used again.		
Parameters:	**ident** / **size** / **description** HV_LEVEL / 1 byte / HV Level (range 0 – 214, 0 – 6.0 kVolt, 157 ~ 4.5 kVolt) unused / 3 bytes / -		
Acceptance:	Direct; command is stored when waiting for 'Confirm Critical'. 'Confirm Critical' is accepted when the instrument is in CHECKOUT state and commanded value is below the maximum allowed command value as defined in the parameter file (P_HV_MAX_HVSET).		
Execution:	After correct confirmation and checks, when requested HV level is reached.		
Errors:	0x80 ecHV_COMMANDED_ABOVE_MAXIMUM_LIMIT 0x83 ecHV_RAMPUP_TERMINATED 0x82 ecHV_LEVEL_SET_ABOVE_MAXIMUM_LIMIT 0x81 ecHVPS_OFF_BEFORE_RAMPUP_COMPLETE		

Mnemonic:	ALICE_CONTROL_HEATER		
Command code:	0x4111	**Critical Command:**	No
Command size:	12 bytes	**Only in Checkout state:**	Yes
Description:	Specify a temperature control set point for the control algorithm of the selected heater. Commanding the heater to a set point of zero will turn the heater off, commanding a heater to the maximum level (255) will turn the heater on continuously. In all other cases the heater(s) will be activated whenever the measured temperature is below the set point. Heaters will immediately be turned off whenever the instrument transitions to the SAFE state. The heater control algorithm evaluates the set point and measured temperature every 10 seconds, so it may take up to 10 seconds before a specified heater is activated after the command is send and the measured temperature is below the set point. The parameter file entries 'P_HtrSenseMirror' and 'P_HtrSenseGrating' control whether either the primary or the redundant temperature sensor is used by the temperature control algorithm. The parameter file entries 'P_MirrorHtr?Enable' and 'P_GratingHtr?Enable' control which of the two redundant heaters may be activated, these parameters may also select both or neither heaters of a pair. Heaters use the actuator power bus so heaters can only be activated when the corresponding actuator bus is activated.		
Parameters:	**ident** / **size** / **description** HEATER / 1 byte / Heater id 1 – Mirror decontamination heater 2 – Grating decontamination heater SETPOINT / 1 byte / Heater set point (0-255, -30 - +60 ˚C) unused / 2 bytes / -		

Mnemonic:	ALICE_CONTROL_HEATER
Acceptance:	When the instrument is in CHECKOUT state and a correct heater is specified.
Execution:	Direct when the requested set point is loaded in the temperature controller for the specified heater.
Errors:	0x90 ecINCORRECT_HEATER_IDENTIFICATION

Mnemonic:	ALICE_ACTIVATE_WPASMA		
Command code:	0x4112	**Critical Command:**	Yes
Command size:	12 bytes	**Only in Checkout state:**	Yes
Description:	Activate the primary, the redundant or both actuators of a given actuator pair. The activation of any given actuator control will be terminated if: – the actuation period times out, – in case of the WPA the WPA switch is activated and the switch control is enabled (P_WpaSensorEnable) – the system changes to the SAFE state, – a deactivation WPA/SMA command is sent, Duration of the actuator activation for actuators labeled as SMA is defined by the parameter file entry P_TINI_CONTROL (in a range from 5 ms – 1.275 sec, with a typical value of 70 ms for the real SMAs, but dependant on temperature and typical 30 ms for the valve solenoid). Maximum duration of the actuator activation for the actuators labeled as WPA (detector door) is defined by the parameter file entry P_WPA_TIMEOUT (in a range 10 sec – 2550 sec, a typical activation takes 150 seconds but is dependant on temperature).		
	ident	**size**	**description**
Parameters:	ACTUATOR	1 byte	Actuator id 0 – none, deactivate all 1 – primary actuator door lock actuator (SMA) 2 – redundant actuator door lock actuator (SMA) 3 – both actuator door lock actuators (SMA) 4 – failsafe door actuator (SMA) 5 – backup vent valve (‘SMA’ solenoid) 6 – both failsafe and valve actuators (SMA) 7 – primary soc cover actuator (SMA) 8 – secondary soc cover actuator (SMA) 9 – both soc cover actuators (SMA) 10 – primary detector door actuator (WPA) 11 – secondary detector door actuator (WPA) 12 – both detector door actuators (WPA)
	unused	3 bytes	-
Acceptance:	Direct; command is stored when waiting for ‘Confirm Critical’. ‘Confirm Critical’ is accepted when the instrument is in CHECKOUT state and: – The controlling task is not active and a valid actuator is specified or – The actuator is commanded off		
Execution:	After correct confirmation, when execution completes without being terminated		

Mnemonic:	ALICE_ACTIVATE_WPASMA	
Errors:	0x40	ecINCORRECT_ACTUATOR_IDENT_NUMBER
	0x41	ecACTUATOR_CONTROLLER_TASK_BUSY
	0x42	ecSMA_CONTROL_TERMINATED
	0x43	ecWPA_CONTROL_TERMINATED
	0x44	ecSMA_ZERO_DURATION_ACTIVATION
	0x45	ecWPA_ZERO_DURATION_ACTIVATION
	0x46	ecWPA_SWITCH_ENABLED_AND_ACTIVE

Mnemonic:	ALICE_PERFORM_SELF_TEST		
Command code:	0x4113	**Critical Command:**	No
Command size:	12 bytes	**Only in Checkout state:**	Yes
Description:	Perform the self test sequences specified in the test mask. The duration of the command execution depends on the selected tests. The tests are executed starting by the test selected by the least significant bit to the test selected by the most significant bit. At the start of the test execution the reported test status byte (Hk_TEST_STATUS) will be set to 0xff and this status will be reported in at least one HK packet.. Whenever one of the specified tests completes successfully the corresponding bit will be cleared. Bits corresponding to not selected test will always be cleared (in order of test selection).		
	ident	**size**	**description**
Parameters:	TEST_MASK	1 byte	Test mask (1 byte) 0x01 – Acquisition memory A (32 sec) 0x02 – Acquisition memory B (32 sec) 0x04 – Acquisition page switching (7 sec) 0x08 – unused Data RAM (48 sec) 0x10 – Code Memory (EXEC) (29 sec) 0x20 – EEPROM Parameter files (<1 sec) 0x40 – EEPROM pages 1, 3 (48 sec) 0x80 – EEPROM pages 2, 4 (48 sec)
	unused	3 bytes	
Acceptance:	When the instrument is in CHECKOUT state and the 'slow' Task is ready to accept a command.		
Execution:	When execution of the selected tests by the 'slow' Task completes		
Errors:	0xd0	ecNO_TEST_START_SLOW_TASK_BUSY	
	0xd1	ecEEPROM1_CHECKSUM_MISMATCH	
	0xd3	ecEEPROM2_CHECKSUM_MISMATCH	
	0xd3	ecEEPROM3_CHECKSUM_MISMATCH	
	0xd4	ecEEPROM4_CHECKSUM_MISMATCH	

Mnemonic:	ALICE_START_PROGRAM		
Command code:	0x4116	**Critical Command:**	Yes
Command size:	12 bytes	**Only in Checkout state:**	Yes

Description: (Re)start execution of the onboard software at specified address. The three planned reasons for execution of this command are:

1- Start execution of the code in the PROM memory, this corresponds to a software reset and causes a complete reinitialization of the software environment, corresponding to startup after completion of the system hardware reset,

2- Start execution of the code in one of the four EEPROM pages upon TC command, after an EEPROM page has been loaded with updated code, or with the initial stored copy of the code in PROM,

3- Start execution of some special code entry to perform a function that is not supported by the standard commands, like executing the duplicate PROM code or executing a special test function that has previously been uploaded into RAM or EEPROM.

Note that in the processor mapping the PROM and EEPROM memory appears in the bottom 32 kbyte of the memory space, the top 32 kbyte always maps to the shared (code and data space) RAM memory. This RAM memory may still be used to contain executable code as long as it doesn't interfere with any general data variables used by the software.

	ident	**size**	**description**
Parameters:	CODE_MEMORY	1 byte	0x51 – EEPROM page 1 0x52 – EEPROM page 2 0x53 – EEPROM page 3 0x54 – EEPROM page 4 0x56 – CODE memory
	unused	1 byte	-
	START_ADDRESS	2 bytes	Start address (address range 0..65535)

Acceptance: Direct; command is stored when waiting for 'Confirm Critical'. 'Confirm Critical' is accepted when the instrument is in CHECKOUT state and a correct memory type is specified for the commanded restart. This last acceptance may not be reported though as the restart is executed immediately and only if the called function returns the acceptance reporting can be performed.

Execution: After correct confirmation and parameter verification, always. The execution (status) may not be reported though, as the restart is executed immediately and only if the called function returns the execution reporting can be performed.

Errors: 0x99 ecINCORRECT_MEMORY_TYPE_FOR_RESTART

Mnemonic:	ALICE_REQUEST_RESTART			
Command code:	0x4117		**Critical Command:**	Yes
Command size:	12 bytes		**Only in Checkout state:**	Yes
Description:	Set request restart flags in the P-Alice HK packets. In response the spacecraft will power down or issue restart (depending on state and autonomy rules) to P-Alice. The request flags remain set until another set of request flags is received by the system (or the instrument is reset). This command is only meant to support testing of the instrument request system, other (spacecraft) command options would be available to directly command a power down or instrument reset.			
	ident	**size**	**description**	
Parameters:	REQUEST	1 byte	0 – no request 1 – request power off 2 – request power cycle 3 – request power off and power cycle	
	unused	3 bytes	-	
Acceptance:	Direct; command is stored when waiting for 'Confirm Critical'. 'Confirm Critical' is accepted when the instrument is in CHECKOUT state and a valid request is specified.			
Execution:	After correct confirmation, always.			
Errors:	0x98 ecINCORRECT_REQUEST_STARTMASK			

Mnemonic:	ALICE_RESET_TC_STATUS			
Command code:	0x4118		**Critical Command:**	No
Command size:	8 bytes		**Only in Checkout state:**	Yes
Description:	Reset TC failure status detection (start receiving from both TC channels again). Execution of this command will also reset the 'last failed command' and 'last failure code' fields that are reported in the housekeeping packet.			
	ident	**size**	**description**	
Parameters:	none			
Acceptance:	When the instrument is in CHECKOUT state.			
Execution:	always			
Errors:	none			

Mnemonic:	ALICE_CHECK_MEMORY			
Command code:	0x4119		**Critical Command:**	No
Command size:	16 bytes		**Only in Checkout state:**	Yes
Description:	Calculate a 16-bit byte wise xor rotate checksum over a specified block of memory, the calculated value is reported in the standard HK packet. At the start of the execution of this command the memory checksum value reported in the HK data packet will be set to zero. The memory checksum value reported in the HK data packet will contain the valid result when the completion of command execution is indicated. Calculation of the checksum is dependant on the block size, over a 32 kbyte of CODE (PROM) or EEPROM space will take about 24 seconds.			

Mnemonic:	ALICE_CHECK_MEMORY		
	ident	**size**	**description**
Parameters:	START_ADDRESS	4 bytes	Start address
	LENGTH	2 bytes	Number of bytes
	MEMORY_TYPE	1 byte	Memory type from where the data should be checked, for P-Alice the following types are defined for dump operations: 0x50 – DATA memory (addr range 0..65535) 0x51 – EEPROM page 1 (addr range 0..32767) 0x52 – EEPROM page 2 (addr range 0..32767) 0x53 – EEPROM page 3 (addr range 0..32767) 0x54 – EEPROM page 4 (addr range 0..32767) 0x55 – Acquisition memory (addr range 0.. 65535, by definition inactive page) 0x56 – CODE memory (address range 0..65535)
	unused	1 byte	
Acceptance:	When the instrument is in CHECKOUT state, a correct memory specification is received and the 'slow' Task is available for command execution.		
Execution:	After completion of the checksum calculation always		
Errors:	0x60 ecDATAMEM_START_TOO_LARGE 0x61 ecDATAMEM_LENGTH_TOO_LARGE 0x62 ecDATAMEM_INCORRECT_BLOCK_SIZE 0x63 ecACQMEM_START_TOO_LARGE 0x64 ecACQMEM_LENGTH_TOO_LARGE 0x65 ecACQMEM_INCORRECT_BLOCK_SIZE 0x66 ecCODEMEM_START_TOO_LARGE 0x67 ecCODEMEM_LENGTH_TOO_LARGE 0x68 ecCODEMEM_INCORRECT_BLOCK_SIZE 0x69 ecEEPROM_MEM_START_TOO_LARGE 0x6a ecEEPROM_MEM_LENGTH_TOO_LARGE 0x6b ecEEPROM_MEM_INCORRECT_BLOCK_SIZE 0x6c ecMEMORY_INCORRECT_MEMORY_TYPE 0x6d ecZERO_LENGTH_MEM_BLOCK 0x70 ecNO_MEMCHECK_START_SLOW_TASK_BUSY		

Mnemonic:	ALICE_LOAD_MEMORY		
Command code:	0x0014	**Critical Command:**	Yes
Command size:	20-144 bytes	**Only in Checkout state:**	Yes
Description:	Load the specified memory with new data/code. For P-Alice the maximum length of the data to be loaded in a single load command is 128 bytes. When loading into EEPROM the block may not straddle a 128-byte boundary and code must be executed from PROM. Verification of the memory load includes a comparison between the data read back from the updated memory and the data that was loaded into the memory. This may fail if the memory written to is not actual memory but control registers that are not read/write accessible as are present in the DATA memory space (see C&DH memory map in **Error! Reference source not found.**)		

Mnemonic:	ALICE_LOAD_MEMORY		
	ident	**size**	**description**
Parameters:	START_ADDRESS	4 bytes	Start address for the data to be written, specified within the instrument address space.
	LENGTH	2 bytes	Number of bytes to be loaded, for P-Alice maximum of 128.
	MEMORY_TYPE	1 byte	Memory type where the data should be loaded, for P-Alice the following types are defined for load operations: 0x50 – DATA memory (address range 0..65535) 0x51 – EEPROM page 1 (addr range 0..32767) 0x52 – EEPROM page 2 (addr range 0..32767) 0x53 – EEPROM page 3 (addr range 0..32767) 0x54 – EEPROM page 4 (addr range 0..32767) 0x55 – Acquisition memory (addr range 0..65535, by definition inactive page)
(continued)	**ident**	**size**	**description**
	unused	1 byte	-
	DATA	n bytes	Data to be loaded into memory (1-128 bytes).
	pad	f bytes	Pad up to size that is a multiple of 4 (0-3 bytes).
Acceptance:	Direct; command is stored when waiting for ‘Confirm Critical’. ‘Confirm Critical’ is accepted when the instrument is in CHECKOUT state, a correct memory specification is received, when loading EEPROM currently executing from PROM and the ‘slow’ Task is available for command execution.		
Execution:	After correct confirmation and parameter verification when the successful load of the data is verified		
Errors:	0x60 ecDATAMEM_START_TOO_LARGE 0x61 ecDATAMEM_LENGTH_TOO_LARGE 0x62 ecDATAMEM_INCORRECT_BLOCK_SIZE 0x63 ecACQMEM_START_TOO_LARGE 0x64 ecACQMEM_LENGTH_TOO_LARGE 0x65 ecACQMEM_INCORRECT_BLOCK_SIZE 0x69 ecEEPROM_MEM_START_TOO_LARGE 0x6a ecEEPROM_MEM_LENGTH_TOO_LARGE 0x6b ecEEPROM_MEM_INCORRECT_BLOCK_SIZE 0x6c ecMEMORY_INCORRECT_MEMORY_TYPE 0x6d ecZERO_LENGTH_MEM_BLOCK 0x73 ecNO_MEMLOAD_LENGTH_ABOVE_128 0x74 ecNO_MEMLOAD_INTO_CODE_PROM 0x74 ecNO_MEMLOAD_BLOCK_STRADDLES_128 0x75 ecNO_EEPROM_LOAD_EXECUTING_EEPROM 0x78 ecMEMORY_LOAD_READBACK_MISMATCH		

Mnemonic:	ALICE_DUMP_MEMORY		
Command code:	0x0015	**Critical Command:**	No
Command size:	20 bytes	**Only in Checkout state:**	Yes

Description: Generate memory dump packages on the low rate telemetry interface, multiple dump packages may be generated when the request it too large to be transferred in a single CCSDS memory dump packet (maximum P-Alice dump package transfers 128 data bytes).
The duration of the command execution is influenced by the memory dump flow control that is executed by the spacecraft (flow control is included in the time synchronization message). Generation of memory dump packets may be prevented for an arbitrary period of time.
An ongoing memory dump command will be terminated when the instrument leaves the CHECKOUT state.

Parameters:

ident	**size**	**description**
START_ADDRESS	4 bytes	Start address for the data to be read, specified within the instrument address space.
LENGTH	4 bytes	Number of bytes to be dumped
MEMORY_TYPE	1 byte	Memory type from where the data should be dumped, for P-Alice the following types are defined for dump operations: 0x50 – DATA memory (addr range 0..65535) 0x51 – EEPROM page 1 (addr range 0..32767) 0x52 – EEPROM page 2 (addr range 0..32767) 0x53 – EEPROM page 3 (addr range 0..32767) 0x54 – EEPROM page 4 (addr range 0..32767) 0x55 – Acquisition memory (addr range 0.. 65535, by definition inactive page) 0x56 – CODE memory (address range 0..65535)
unused	3 bytes	-

Acceptance: When the instrument is in CHECKOUT state, a correct memory specification is received and the ‘slow’ Task is available for command execution.

Execution: After completion of the dump operation (duration also depending on spacecraft flow control) always.

Errors:

0x60	ecDATAMEM_START_TOO_LARGE
0x61	ecDATAMEM_LENGTH_TOO_LARGE
0x62	ecDATAMEM_INCORRECT_BLOCK_SIZE
0x63	ecACQMEM_START_TOO_LARGE
0x64	ecACQMEM_LENGTH_TOO_LARGE
0x65	ecACQMEM_INCORRECT_BLOCK_SIZE
0x66	ecCODEMEM_START_TOO_LARGE
0x67	ecCODEMEM_LENGTH_TOO_LARGE
0x68	ecCODEMEM_INCORRECT_BLOCK_SIZE
0x69	ecEEPROM_MEM_START_TOO_LARGE
0x6a	ecEEPROM_MEM_LENGTH_TOO_LARGE
0x6b	ecEEPROM_MEM_INCORRECT_BLOCK_SIZE
0x6c	ecMEMORY_INCORRECT_MEMORY_TYPE
0x6d	ecZERO_LENGTH_MEM_BLOCK
0x71	ecNO_MEMDUMP_START_SLOW_TASK_BUSY
0x72	ecMEMDUMP_HALTED_NO_CHECKOUT_STATE
0x79	ecMEMORY_DUMP_TERMINATED

Appendix F - Internally Stored Parameter List

A number of instrument operations are controlled by configuration parameters stored in a table called the Parameter File. The current contents of this table in data memory (RAM) can be changed using the 'ALICE_SET_PARAMETER' command and the current contents can be inspected using the parameter reporting function in the HK packet. Upon system startup the Parameter File is initialized with values read from a triple redundant store in non-volatile (EEPROM) memory. If this initial restore fails the value of the specific parameter defaults to a hardcoded backup value. The list with default values specify the parameter file of the FM status on 2004/09/02. The names of the parameter fields (all names with the prefix 'P_') match the parameter names (without the 'P_') for the 'ALICE_SET_PARAMETER' command used to modify the corresponding field.

Table 24: P-Alice internally stored parameter list

#	Parameter		Size in bytes	Description	Default value	Hardcoded (backup) value
	General control parameters					
0	P_GENERAL_1	P_TaskCalls (1)	1	Debug only: 1 - report task calls instead of TaskStacks in HK packet (mask = 0x80)	0	0
		Spare (2)		-	-	-
		P_DoorEnable (1)		1 - Enable door close on safety (mask = 0x10)	1	1
		Spare (1)		-	-	-
		P_WpaSensorEnable (1)		1 - Enable wax pellet actuator sensor feedback (mask = 0x04)	1	1
		P_TcDisable (2)		0 - both enabled; 1,3 – disable channel 1; 2 disable channel 2 (mask = 0x03)	0	0
1	P_GENERAL_2	Spare (1)	1	-	-	-
		P_HtrSenseGrating (1)		1 - Secondary grating heater sensor select (mask = 0x40: 0=primary, 1=secondary)	0	0
		P_GratingHtr1Enable (1)		1 - Grating heater 1 control enabled (mask = 0x20)	1	1
		P_GratingHtr2Enable (1)		1 - Grating heater 2 control enabled (mask = 0x10)	1	1
		Spare (1)		-	-	-
		P_HtrSenseMirror (1)		1 - Secondary mirror heater sensor select (mask = 0x04: 0=primary, 1=secondary)	0	0
		P_MirrorHtr1Enable (1)		1 - Mirror heater 1 control enabled (mask = 0x02)	1	1
		P_MirrorHtr2Enable (1)		1 - MIrror heater 2 control enabled (mask = 0x01)	1	1
2	P_CMD_TIMEOUT		1	Critical command timeout period (0-255 sec)	30	30
3	P_TC_MAX_ERROR		1	Number of errors allowed on any TC channels before disabling channel (0 – not active)	5	5

#	Parameter		Size in bytes	Description	Default value	Hardcoded (backup) value
4	P_WPA_TIMEOUT		1	Wax pellet actuator timeout in seconds*10 (0-2550 sec), after the specified time the actuator will be disabled	20 (200 sec)	20 (200 sec)
5	P_TINI_CONTROL		1	Shape metal actuators control time in seconds/200 (0-1.275 sec, steps of 5 ms), shape metal actuator signals will be active for the specified amount of time without feedback	30 (150 ms)	30 (150 ms)
6	P_DOOR_CONTROL		1	Aperture door motion control time in seconds/10 (0-25.5 sec), door control will be active for the specified amount of time without feedback	18 (1.8 sec)	18 (1.8 sec)
7	P_REPORT_PARAM		1	Current parameter value reported in housekeeping (255 results in cyclic reporting).	255 (cyclic)	255 (cyclic)
8	Spare (4)		1	-	-	-
	P_HW_VERSION_ID (4)			H/W Board Version Id included in HK TM packet to identify which model generated the received/ recorded HK TM packet (0 - illegal, 1 - EM, 2 - SM, 3 - FM, 5 - SIM,…, 7 – PROM, …)	**3 (FM)**	7 (PROM default)
	Acquisition control parameters					
9	P_ACQ_GENERA	Spare (3)	1	-	-	-
		P_StimEnable		1 - Enable Pixel STIM at start of acquisition (mask = 0x10)	1	1
		Spare (2)		-	-	-
		P_Hvps1Enable (1)		1 - Enable Primary HVPS when commanded (mask = 0x02)	1	1
		P_Hvps2Enable (1)		1 - Enable Secondary HVPS when commanded (mask = 0x01)	1	1
10	P_DISCRIMINATOR		1	Discriminator set level (units, DAC counts)	43 (~0.5 Volt)	43 (~0.5 Volt)
11	P_HV_LEVEL		1	High voltage operating level (units, DAC counts)	157 (~4.5 kV)	157 (~4.5 kV)
12	P_HV_STEP_SIZE		1	High voltage step fraction (of difference between requested and commanded value)	**45 (16/45 = 0.35)**	37 (16/37 = 0.43)
13	P_HV_STEP_TIME		1	High voltage step duration in seconds (0-255)	10	10
14	P_PIXELLIST_HACK		1	Time hack used for pixel list acquisitions (0- 4 ms, 1- 8 ms, 2- 16 ms, 3- 32 ms, 4- 64 ms, 5- 128 ms, 6- 256 ms, 7- 512 ms)	0 (4 ms)	0 (4 ms)
15, 16	P_HISTO_EXP_DUR		2	Duration of one histogram acquisition exposure in seconds (1-65535)	20	20
17, 18	P_ACQ_TIMEOUT		2	Acquisition timeout, defines backup acquisition termination, specifies maximum acquisition duration in seconds (1-65535)	100	100

#	Parameter	Size in bytes	Description	Default value	Hardcoded (backup) value
19	P_HOTSEG_1	1	Hot segment 1; Each hot segment specification masks out detector events in a rectangular area (32 spectral * 4 spatial pixels), top three bits specify the spatial position of this window, remaining 5 bits specify the spectral position of the window.	0	0
20	P_HOTSEG_2	1	Hot segment 2	0	0
21	P_HOTSEG_3	1	Hot segment 3	0	0
22	P_HOTSEG_4	1	Hot segment 4	0	0
23	P_HOTSEG_5	1	Hot segment 5	0	0
24	P_HOTSEG_6	1	Hot segment 6	0	0
25	P_HOTSEG_7	1	Hot segment 7	0	0
26	P_HOTSEG_8	1	Hot segment 8	0	0
	Safety checking parameters				
27, 28	P_MAX_COUNT_RATE	2	Maximum allowed count rate (Hz)	15000	15000
29	P_HV_LOW_SAFETY	1	HV lowest voltage setting above which the safety checking can be activated, this prevents problems with lower region of control not being linear etc.	60 (~2 kV)	60 (~2 kV)
30	P_DAC_ADC_FACTOR	1	Conversion from to DAC setting to ADC read back used in HVPS checking (factor/240)	208 (*0.87)	208 (*0.87)
31	P_HV_MAX_HVSET	1	Maximum allowed HV setpoint voltage (DAC)	161 (~4.6 kV)	161 (~4.6 kV)
32	P_HV_MCP_TOL	1	MCP voltage tolerance (units, ADC counts)	4 (~100 V)	4 (~100 V)
33	P_HV_FAIL_MCP	1	MCP voltage max fail count	5	5
34	P_HV_MAX_STRIPI	1	Maximum allowed strip current (ADC counts)	127 (~15 uA)	127 (~15 uA)
35	P_HV_FAIL_STRIP	1	Strip current max fail count	5	5
36	P_HV_MIN_ANODEV	1	Minimum allowed anode voltage (ADC counts)	180 (~ 570 V)	180 (~ 570 V)
37	P_HV_MAX_ANODEV	1	Maximum allowed anode voltage (ADC counts)	199 (~ 630 V)	199 (~ 630 V)
38	P_HV_FAIL_ANODE	1	Anode voltage max fail count	5	5
39	P_MAX_MIRR1_TEMP	1	Maximum allowed temperature (ADC counts)	220 (~50 ˚C)	220 (~50 ˚C)
40	P_MAX_MIRR2_TEMP	1	Maximum allowed temperature (ADC counts)	220 (~50 ˚C)	220 (~50 ˚C)
41	P_MAX_GRAT1_TEMP	1	Maximum allowed temperature (ADC counts)	215 (~45 ˚C)	215 (~45 ˚C)
42	P_MAX_GRAT2_TEMP	1	Maximum allowed temperature (ADC counts)	215 (~45 ˚C)	215 (~45 ˚C)
43	P_MAX_DET_ELEC_TEMP	1	Maximum allowed temperature (ADC counts)	224 (~55 ˚C)	224 (~55 ˚C)
44	P_MAX_DET_HOUSE_TEMP	1	Maximum allowed temperature (ADC counts)	215 (~45 ˚C)	215 (~45 ˚C)
45	P_MAX_CDH_ELEC_TEMP	1	Maximum allowed temperature (ADC counts)	224 (~55 ˚C)	224 (~55 ˚C)
46	P_MAX_SOC_MIRR_TEMP	1	Maximum allowed temperature (ADC counts)	220 (~50 ˚C)	220 (~50 ˚C)

#	Parameter	Size in bytes	Description	Default value	Hardcoded (backup) value
47	P_TEMP_MASK	1	Mask to disable certain temperature sensors from the temperature safety check (1=masked, 0- enabled). – 0x80 mirror 1 – 0x40 mirror 2 – 0x20 grating 1 – 0x10 grating 2 – 0x08 detector electronics – 0x04 detector housing – 0x02 c&dh electronics – 0x01 soc mirror	0x00	0x00
48	P_SAFETY_MASK	1	Initial startup value for the safety mask and override (1=masked, 0-enabled) – 0x80 Override (all masked) – 0x10 Temperature safety – 0x08 Anode safety – 0x04 Strip safety – 0x02 HV safety – 0x01 Bright safety	0x00	0x00
49, 50	P_SAFETY_TIME	2	Safety timeout (seconds)	60	60
51-56	P_Spare	6	Unassigned	0	0
57, 58	P_Number of modifications	2	Accumulated count of changes made to the parameter memory in EEPROM memory to provide a tracking mechanism for the maximum number of writes cycles.	**N/A**	0
59-127	Filler (54%)	69	Pad size of table to a full EEPROM block (page)	-	0
	Total table size in bytes	128	Table stored redundantly in internal non-volatile instrument memory		

Appendix G - Error Codes

Table 25: P-Alice Error Codes

```
RCSfile: ErrorDef.h  - $Revision: 1.15  - Date: 2004/05/07 18:24:01
ecUNUSED_00                          = 0x00,
/* ITF decoding errors --------------------------------------------------*/
/* _1 and _2 indicate channel on which byte was received                 */
ecINCORRECT_ITF_CHECKSUM_1           = 0x01, // Tc::handleTcByte()
ecINCORRECT_ITF_CHECKSUM_2           = 0x02, // Tc::handleTcByte()
ecUNSUPPORTED_ITF_MESSAGE_TYPE_1     = 0x03, // Tc::handleTcByte()
ecUNSUPPORTED_ITF_MESSAGE_TYPE_2     = 0x04, // Tc::handleTcByte()
ecTOO_LARGE_ITF_CMD_PACKAGE_1        = 0x05, // Tc::handleTcByte()
ecTOO_LARGE_ITF_CMD_PACKAGE_2        = 0x06, // Tc::handleTcByte()
ecTOO_SHORT_ITF_CMD_PACKAGE_1        = 0x07, // TmTc::timeoutTcSerial(()
ecTOO_SHORT_ITF_CMD_PACKAGE_2        = 0x08, // TmTc::timeoutTcSerial(()
ecINCORRECT_1ST_ITF_SYNC_BYTE_1      = 0x09, // Tc::handleTcByte()
ecINCORRECT_1ST_ITF_SYNC_BYTE_2      = 0x0a, // Tc::handleTcByte()
ecINCORRECT_2ND_ITF_SYNC_BYTE_1      = 0x0b, // Tc::handleTcByte()
ecINCORRECT_2ND_ITF_SYNC_BYTE_2      = 0x0c, // Tc::handleTcByte()
ecINCORRECT_3RD_ITF_SYNC_BYTE_1      = 0x0d, // Tc::handleTcByte()
ecINCORRECT_3RD_ITF_SYNC_BYTE_2      = 0x0e, // Tc::handleTcByte()
ecITF_ERROR_WAITING_1_2              = 0x10, // TmTc::reportItfError()
ecTC_BUFFER_OVERFLOW_1               = 0x11, // TmTc::isrTmTc()
ecTC_BUFFER_OVERFLOW_2               = 0x12, // TmTc::isrTmTc()
ecRX_FLUSH_FAILED_1                  = 0x13, // TmTc::isrTmTc()
ecRX_FLUSH_FAILED_2                  = 0x14, // TmTc::isrTmTc()
/* TM gneration errors --------------------------------------------------*/
ecTM_TRANSFER_MISSED                 = 0x18, // Tm::taskTM()
ecTM_TRANSFER_ERROR                  = 0x19, // TmTc::transmitFrame()
/* General command decoding ---------------------------------------------*/
ecIMPROPER_LENGTH_FOR_COMMAND        = 0x20, // Tc::processTc()
ecUNKNOWN_UNRECOGNIZED_TELECOMMAND   = 0x21, // Tc::tcUnknown()
ecWRONG_NUMBER_OF_COMMAND_WORDS      = 0x22, // Tc::processTc()
ecCOMMAND_REQUIRES_CHECKOUT_STATE    = 0x23, // Tc::processTc()
ecALREADY_CRITICAL_COMMAND_PENDING   = 0x24, // Tc::processTc()
ecINCORRECT_CONFIRM_CRITICAL         = 0x25, // Tc::processTc()
ecMISSING_CONFIRM_CRITICAL           = 0x26, // Tc::processTc()
ecNO_CRITCAL_COMMAND_PENDING         = 0x27, // Tc::tcConfirm()
ecCRITICAL_COMMAND_TIMEOUT           = 0x28, // Tc::timeoutCriticalCommand()
ecTIME_MESSAGE_TOO_SHORT             = 0x2c, // Clock::processTimeMessage()
ecTIME_MESSAGE_TOO_LONG              = 0x2d, // Clock::processTimeMessage()
/* Telecommand Execution ------------------------------------------------*/
ecFAILED_TO_CHANGE_TO_CHECKOUT       = 0x30, // State::tcCheckout()
ecINCORRECT_ACTUATOR_IDENT_NUMBER    = 0x40, // WpaSma::tcActivate()
ecACTUATOR_CONTROLLER_TASK_BUSY      = 0x41, // WpaSma::tcActivate()
ecSMA_CONTROL_TERMINATED             = 0x42, // WpaSma::activateSma()
ecWPA_CONTROL_TERMINATED             = 0x43, // WpaSma::activateDetDoorWpa()
ecSMA_ZERO_DURATION_ACTIVATION       = 0x44, // WpaSma::activateSma()
ecWPA_ZERO_DURATION_ACTIVATION       = 0x45, // WpaSma::activateDetDoorWpa()
ecWPA_SWITCH_ENABLED_AND_ACTIVE      = 0x46, // WpaSma::activateDetDoorWpa()
ecCOMMANDED_DOOR_MOVE_FAILED         = 0x48, // Door::taskDoor()
ecNO_ACQ_START_SAFETY_PENDING        = 0x51, // Slow::commandAcquisition()
ecNO_ACQ_START_SLOW_TASK_BUSY        = 0x52, // Slow::commandAcquisition()
ecNO_ACQ_START_WRONG_TEST_PARAM      = 0x53, // Slow::commandAcquisition()
ecNO_ACQ_START_WRONG_DOOR_PARAM      = 0x54, // Slow::commandAcquisition()
ecDATAMEM_START_TOO_LARGE            = 0x60, // Slow::checkGeneralMemParams(
```

RCSfile: ErrorDef.h - $Revision: 1.15 - Date: 2004/05/07 18:24:01

```
    ecDATAMEM_LENGTH_TOO_LARGE          = 0x61, // Slow::checkGeneralMemParams(
    ecDATAMEM_INCORRECT_BLOCK_SIZE      = 0x62, // Slow::checkGeneralMemParams(
    ecACQMEM_START_TOO_LARGE            = 0x63, // Slow::checkGeneralMemParams(
    ecACQMEM_LENGTH_TOO_LARGE           = 0x64, // Slow::checkGeneralMemParams(
    ecACQMEM_INCORRECT_BLOCK_SIZE       = 0x65, // Slow::checkGeneralMemParams(
    ecCODEMEM_START_TOO_LARGE           = 0x66, // Slow::checkGeneralMemParams(
    ecCODEMEM_LENGTH_TOO_LARGE          = 0x67, // Slow::checkGeneralMemParams(
    ecCODEMEM_INCORRECT_BLOCK_SIZE      = 0x68, // Slow::checkGeneralMemParams(
    ecEEPROM_MEM_START_TOO_LARGE        = 0x69, // Slow::checkGeneralMemParams(
    ecEEPROM_MEM_LENGTH_TOO_LARGE       = 0x6a, // Slow::checkGeneralMemParams(
    ecEEPROM_MEM_INCORRECT_BLOCK_SIZE   = 0x6b, // Slow::checkGeneralMemParams(
    ecMEMORY_INCORRECT_MEMORY_TYPE      = 0x6c, // Slow::checkGeneralMemParams(
    ecZERO_LENGTH_MEM_BLOCK             = 0x6d, // Slow::checkGeneralMemParams(
    ecNO_MEMCHECK_START_SLOW_TASK_BUSY  = 0x70, // Slow::tcMemCheck()
    ecNO_MEMDUMP_START_SLOW_TASK_BUSY   = 0x71, // Slow::tcMemDump()
    ecMEMDUMP_HALTED_NO_CHECKOUT_STATE  = 0x72, // Tm::taskTM()
    ecNO_MEMLOAD_LENGTH_ABOVE_128       = 0x73, // Slow::tcMemLoad()
    ecNO_MEMLOAD_INTO_CODE_PROM         = 0x74, // Slow::tcMemLoad()
    ecNO_MEMLOAD_BLOCK_STRADDLES_128    = 0x74, // Slow::tcMemLoad()
    ecNO_EEPROM_LOAD_EXECUTING_EEPROM   = 0x75, // Slow::tcMemLoad()
    ecSLOW_TASK_BUSY                    = 0x76, // Slow::activateSlowTask()
    ecMEMORY_LOAD_READBACK_MISMATCH     = 0x78, // Memory::slowLoadMemory()
    ecMEMORY_DUMP_TERMINATED            = 0x79, // Memory::slowDumpMemory()
    ecHV_COMMANDED_ABOVE_MAXIMUM_LIMIT  = 0x80, // Hvps::tcHvOn()
    ecHVPS_OFF_BEFORE_RAMPUP_COMPLETE   = 0x81, // Hvps::commandHvOff()
    ecHV_LEVEL_SET_ABOVE_MAXIMUM_LIMIT  = 0x82, // Hvps::commandHvOn()
    ecHV_RAMPUP_TERMINATED              = 0x83, // Hvps::taskHvps()
    ecSTIM_NOT_DEACTIVATED              = 0x88, // Detector::tcStimOff()
    ecSTIM_NOT_ACTIVATED                = 0x89, // Detector::tcStimOn()
    ecINCORRECT_HEATER_IDENTIFICATION   = 0x90, // Heater::tcHeater()
    ecINCORRECT_REQUEST_STARTMASK       = 0x98, // System::tcRequest()
    ecINCORRECT_MEMORY_TYPE_FOR_RESTART = 0x99, // System::tcStartProg()

    /* Acquisition Control -------------------------------------------------*/
    ecSYNCED_HISTO_ACQUISITION_FAILED   = 0xa0, // Acquire::slowAcquireHistogra
    ecSYNCED_PIXEL_ACQUISITION_FAILED   = 0xa1, // Acquire::slowAcquirePixelLis
    ecCLEAR_HISTO_ACQUISITION_FAILED    = 0xa2, // Acquire::slowAcquireHistogra
    ecCLEAR_PIXEL_ACQUISITION_FAILED    = 0xa3, // Acquire::slowAcquirePixelLis
    ecSTART_HISTO_ACQUISITION_FAILED    = 0xa4, // Acquire::slowAcquireHistogra
    ecSTART_PIXEL_ACQUISITION_FAILED    = 0xa5, // Acquire::slowAcquirePixelLis
    ecACQUISITION_TIMEOUT_OCCURED       = 0xa6, // Acquire::timeoutAcquisition(

    /* Parameter File ------------------------------------------------------*/
    ecPARAMETER_INDEX_OUT_OF_RANGE      = 0xb0, // Param::tcSetParm()
    ecPARAMETER_READBACK_FAILED         = 0xb1, // Param::tcSetParm()
    ecSTORE_PARAMS_READBACK_FAILED_1    = 0xb2, // Param::tcStoreParms()
    ecSTORE_PARAMS_READBACK_FAILED_2    = 0xb3, // Param::tcStoreParms()
    ecSTORE_PARAMS_READBACK_FAILED_3    = 0xb4, // Param::tcStoreParms()
    ecSTORE_PARAMS_ONLY_WHEN_EXEC_PROM  = 0xb5, // Param::tcStoreParms()
    ecINCORRECT_TABLE_FOR_PARAM_LOAD    = 0xb6, // Param::tcLoadParms()
    ecLOAD_PARAM_TABLE_1_MISMATCH       = 0xb7, // Param::loadMajorityEepromPar
    ecLOAD_PARAM_TABLE_2_MISMATCH       = 0xb8, // Param::loadMajorityEepromPar
    ecLOAD_PARAM_TABLE_3_MISMATCH       = 0xb9, // Param::loadMajorityEepromPar
    ecLOAD_PARAM_ALL_DIFFERENT          = 0xba, // Param::loadMajorityEepromPar
```

RCSfile: ErrorDef.h - $Revision: 1.15 - Date: 2004/05/07 18:24:01

```
/* Internal Errors ------------------------------------------------------*/
ecTOO_MANY_BYTES_FOR_EEPROM_WRITE   = 0xc0, // Memory::writeEeprom()
ecTOO_MANY_BYTES_FOR_EEPROM_READ    = 0xc1, // Memory::readEeprom()

/* Self Test ------------------------------------------------------------*/
ecNO_TEST_START_SLOW_TASK_BUSY      = 0xd0, // Slow::tcSelfTest()
ecEEPROM1_CHECKSUM_MISMATCH         = 0xd1, // Memory::testAllCodeMemory()
ecEEPROM2_CHECKSUM_MISMATCH         = 0xd3, // Memory::testAllCodeMemory()
ecEEPROM3_CHECKSUM_MISMATCH         = 0xd3, // Memory::testAllCodeMemory()
ecEEPROM4_CHECKSUM_MISMATCH         = 0xd4, // Memory::testAllCodeMemory()

/* No errors ------------------------------------------------------------*/
ecNEXT_ERROR_LOG_SLOT               = 0xf8, // Tm::logError()
ecNO_ERROR_RESET                    = 0xfd, // Tm::resetLastFailed()
ecNO_ERROR_INITIAL                  = 0xfe, // Tc::Init()
ecUNUSED_FF                         = 0xff
```

Appendix H - Parameter Conversions

All temperature sensor values reported by the instrument and also the two temperature setpoints use raw (ADC) count values. These can be converted into engineering values (°C) using the following polynomial (where C is the raw count value):

$$Temp = -78.03 + 2.385 * c - 4.087 \cdot 10^{-2} * c^2 + 3.752 \cdot 10^{-4} * c^3 - 1.601 \cdot 10^{-6} * c^4 + 2.594 \cdot 10^{-9} * c^5$$

All other measured and commanded parameters use piecewise linear conversions to determine th eengineering values.

Table 26: Piecewise linear conversion parameters

Parameter	Raw counts	Engineering unit
High Voltage Setpoint (control voltage)	0	0.000 kVolt
	31	-1.343 kVolt
	255	-6.943 kVolt
Mcp Voltage (both) and Max Mcp Voltage (readback voltages)	0	0.000 kVolt
	25	-1.332 kVolt
	255	-7.881 kVolt
Anode Voltage (both)	0	0.000 Volt
	255	-808.000 Volt
Maximum Strip Current (combined)	0	0.000 µA
	1	2.640 µA
	255	27.560 µA
Strip current HVPS 1	0	0.000 µA
	1	2.360 µA
	255	46.760 µA
Strip Current HVPS 2	0	0.000 µA
	1	2.400 µA
	255	45.420 µA
Discriminator setpoint	0	0.000 Volt
	255	3.000 Volt

When only a single HVPS is powered, the Strip current measured on that HVPS should be used as the overall detector strip current. When both HVPSs are powered (but not necessary activated), the Maximum Strip current should be used as the overall detector strip current.

Table 27: Parameter Conversions Values

Counts	Temp	HVset	McpV	AnodeV	MaxStripI	StripI1	StripI2	Discrim
	deg C	kVolt	kVolt	Volt	μA	μA	μA	V
0	*-78.0*	0.00	0.00	0	0.0	0.0	0.0	0.00
16	*-48.9*	-0.69	-0.85	-51	4.1	5.0	4.9	0.19
23	*-40.7*	**-1.00**	-1.23	-73	4.8	6.2	6.1	0.27
32	*-32.9*	-1.37	-1.53	-101	5.7	7.8	7.7	0.38
37	-29.6	**-1.49**	-1.67	-117	6.2	8.7	8.5	0.44
43	-26.3	-1.64	-1.84	-136	6.8	9.7	9.5	**0.51**
48	-24.1	-1.77	-1.99	-152	**7.3**	10.6	10.4	0.56
57	-20.7	**-1.99**	-2.24	-181	8.1	12.1	11.9	0.67
64	-18.5	-2.17	-2.44	-203	8.8	13.4	13.1	0.75
77	-14.7	**-2.49**	-2.81	-244	10.1	15.6	15.3	0.91
80	-13.8	-2.57	-2.90	-253	10.4	16.2	15.8	0.94
96	-8.6	-2.97	-3.35	-304	12.0	19.0	18.5	1.13
97	-8.3	**-2.99**	-3.38	-307	12.1	19.1	18.7	1.14
112	-2.7	-3.37	-3.81	-355	13.5	21.8	21.2	1.32
117	-0.7	**-3.49**	-3.95	-371	14.0	22.6	22.0	1.38
128	3.9	-3.77	-4.26	-406	15.1	24.6	23.9	1.51
137	7.6	**-3.99**	**-4.52**	-434	16.0	26.1	25.4	1.61
141	9.3	**-4.09**	*-4.63*	-447	16.4	26.8	26.1	1.66
144	10.5	-4.17	*-4.72*	-456	16.7	27.4	26.6	1.69
145	10.9	**-4.19**	*-4.75*	-459	16.8	27.5	26.8	1.71
149	12.5	**-4.29**	*-4.86*	-472	17.2	28.2	27.5	1.75
153	14.1	**-4.39**	*-4.98*	-485	17.6	28.9	28.1	1.80
157	15.7	**-4.49**	*-5.09*	-497	17.9	29.6	28.8	1.85
160	16.9	-4.57	*-5.18*	-507	18.2	30.2	29.3	1.88
176	**23.1**	*-4.97*	*-5.63*	-558	19.8	33.0	32.0	2.07
192	30.0	*-5.37*	*-6.09*	**-608**	21.4	35.7	34.7	2.26
208	39.5	*-5.77*	*-6.54*	-659	22.9	38.5	37.5	2.45
224	*54.7*	*-6.17*	*-7.00*	-710	24.5	41.3	40.2	2.64
240	*80.8*	*-6.57*	*-7.45*	-760	26.1	44.1	42.9	2.82
255	*121.3*	*-6.94*	*-7.88*	-808	27.6	46.8	45.4	3.00

Red – red limits
Yellow – yellow limits
Bold – nominal manual HV rampup steps
Green – typical operating levels (at 20 deg C)

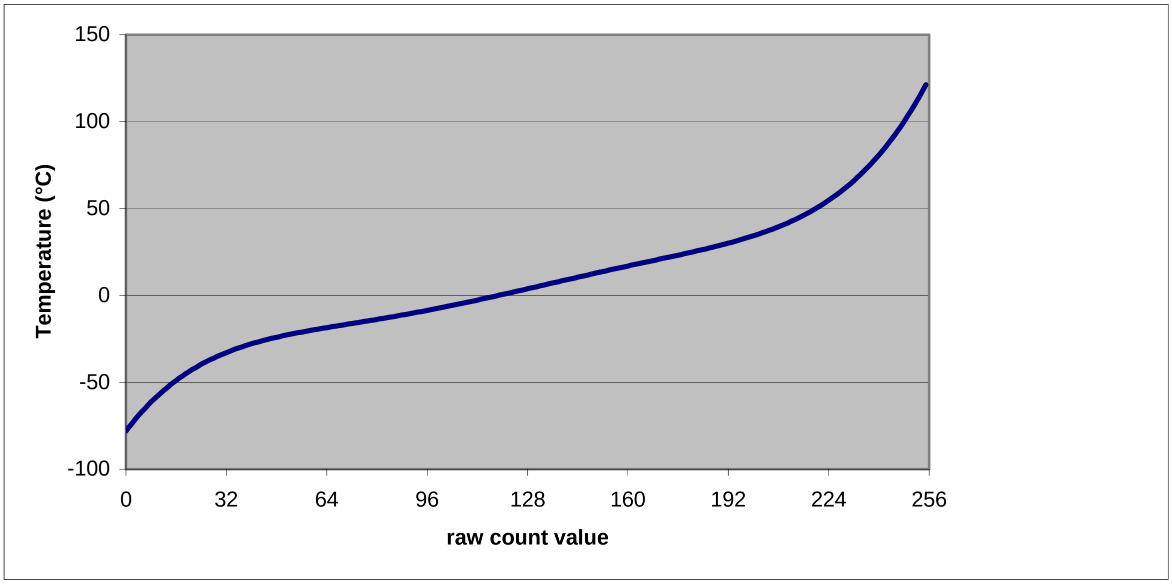

Figure 19: Temperature Conversion Curve

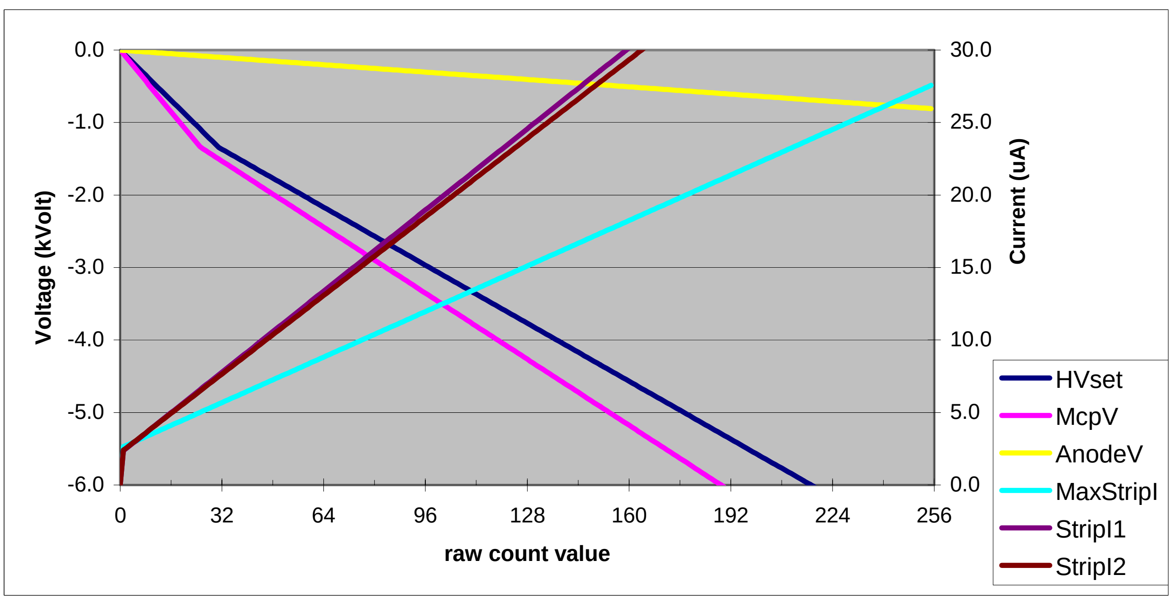

Figure 20: HVPS Parameters Conversion Curve