DOCUMENT NUMBER:	**PY-SSP-OU-UM-001**
ISSUE DATE:	**22 October 97**
Revision Date:	**12/7/06**

HUYGENS

SURFACE SCIENCE PACKAGE

ON-BOARD SOFTWARE USERS MANUAL

ISSUE 2.2

		Rev	Date:
PREPARED BY:	D J Parker	1.0	22/10/1997
REVISED BY:	M R. Patel and L. R. Hobbs	2.0	13/08/2004
	M. R. Leese	2.1	12/12/2005
	M. R. Leese and B. Hathi	2.2	12/07/2006

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Administrative

Scope

This document provides the flight software design for the Huygens Surface Science Package. This document satisfies the requirement detailed in the Huygens Document Requirements Description.

Purpose

This document describes the design and internal functioning of the flight software and relates this information into the applicable requirements specifications(s). The description is presented in a top-down fashion beginning with the software subsystem as a whole and successively describing lower level components. The data and procedure structures as well as the operational data flows supported by these structures are explained. Software external interfaces and the module to module interfaces are described. Provisions for error isolation and verification characteristics are given.

Applicable Documents

	Huygens Experiment Interface Document Part A
PY-SSP-RAL-PL-0001	Huygens Surface Science Package Product Assurance Plan
PY-SSP-UKC-SP-005	Huygens Surface Science Package Software Requirements Document
MIL-STD-1750A	Sixteen-Bit Computer Instruction Set Architecture
	BAe Low Power Processor User Manual

Document Conventions and Abbreviations

Flag	1 bit
Byte	8 bits
Word	16 Bits
Long Word	32 Bits

LSB	Least Significant Bit	Note: Bit 0 is the LSB and bit 15 is the MSB of a
MSB	Most Significant Bit	word, this is opposite to the 1750a specification.

BCP	Broadcast Pulse
CDMS	Command Data Management System
EID	Experiment Interface Document
SSP	Surface Science Package
PDS	Planetary Database System

Sensors

Acc-E	External Accelerometer
Acc-I	Internal Accelerometer
Api-S	Acoustic Properties Instrument-Sounder
Api-V	Acoustic Properties Instrument - Velocity
Den	Density Subsystem
Per	Permittivity Subsystem
Ref	Refractometer Subsystem
Thp	Thermal Properties Subsystem
TIL	TILT Subsystem
HK	Housekeeping

System

ADC	Analogue to Digital Converter
DAC	Digital to analogue Converter
DRB	Data Routing Bus
DDB	Descent Data Broadcast Message
EEPROM	Electrically Erasable Programmable Read Only Memory

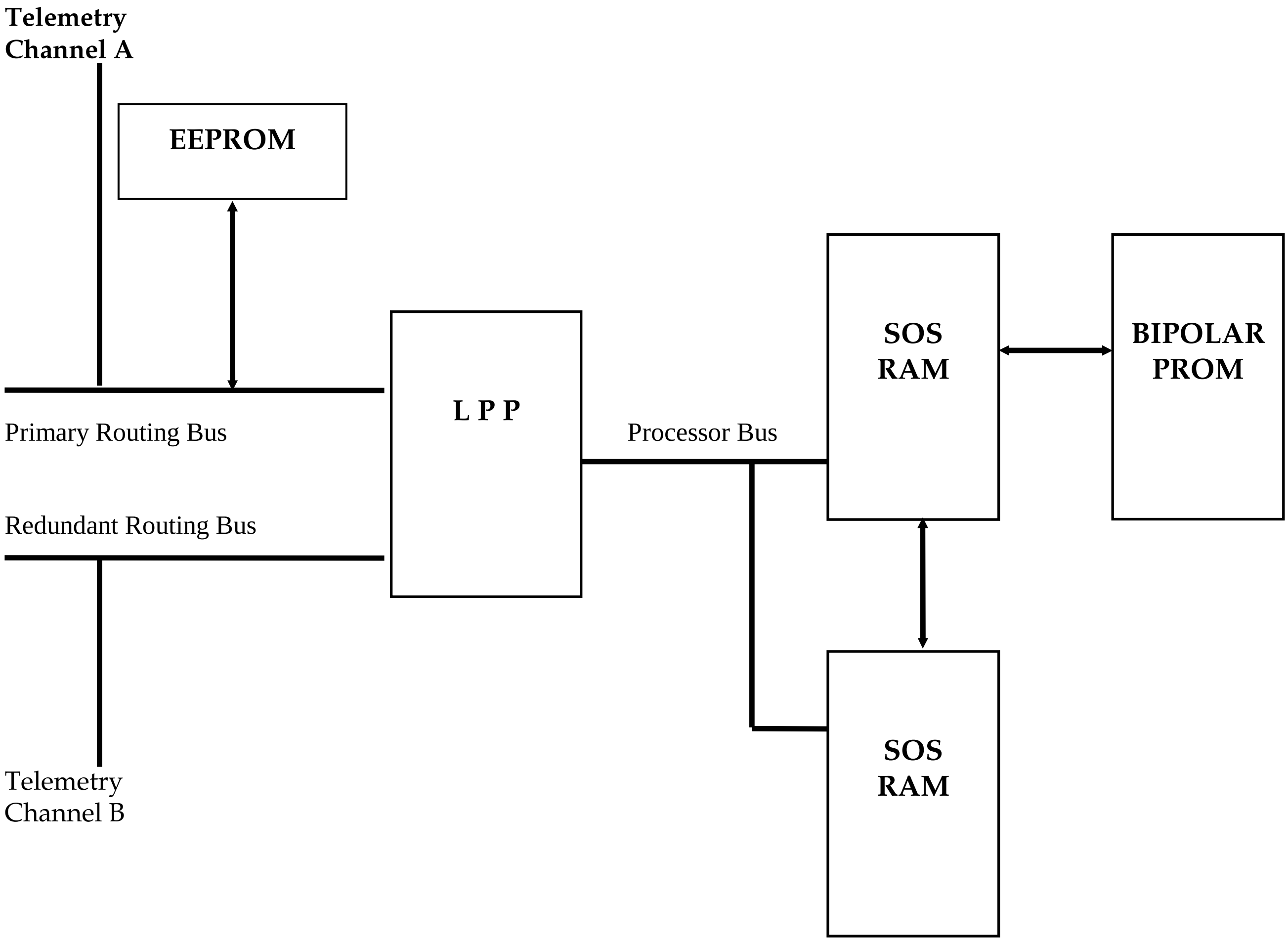
EPROM	Erasable Programmable Read Only Memory
FIFO	First in First Out
LPP	Low Power Processor
ML	Memory Load
PROM	Programmable Read Only Memory
RAM	Random Access Memory
SOS	Silicon On Sapphire
TM	Telemetry

Documentation

TBD	To be Determined
TBC	To be Confirmed

1.2 SSP MEMORY CONFIGURATION

The diagram below shows the physical connection of SSP memory. Note that only PROM and RAM are directly connected to the processor bus, which enhances the reliability. The EEPROM, which can be updated by telecommand, is accessed via an I/O port.



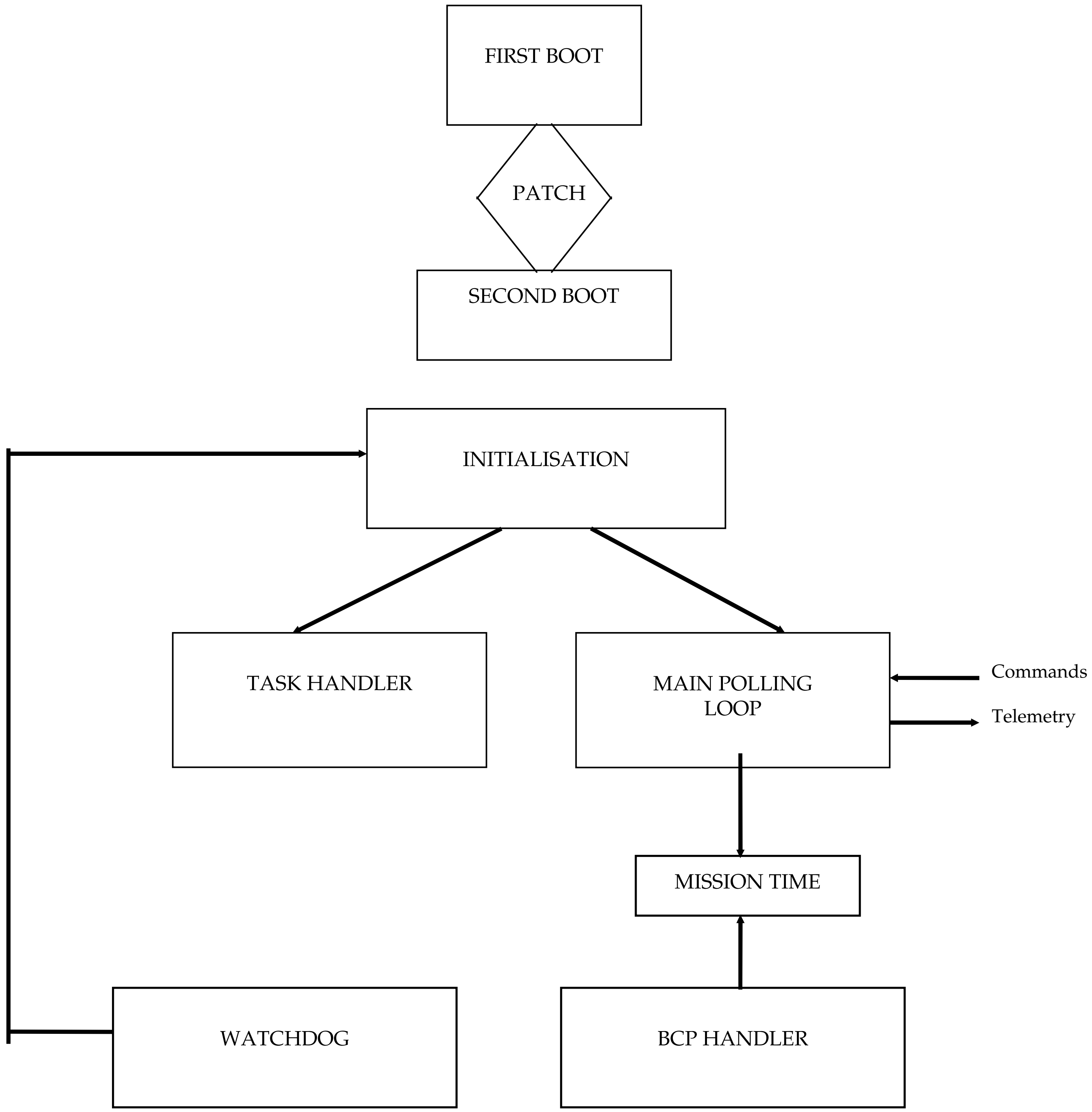
On power up program is copied from PROM to RAM. The EEPROM is then checked to see if it contains any patches. If it does then the program in RAM is updated from EEPROM.

System reliability is also increased by the use of dual Data Routing Busses. Each bus is connected to half of the sensor conditioning circuits, and one spacecraft interface. The EEPROM is on one bus only.

2. SOFTWARE ARCHITECTURE

2.1 Overall Software structure

The diagram below shows the main modules of the SSP code. There are two phases, Boot and Normal Operation. During normal operation synchronous tasks are scheduled by the Task Handler while others are scheduled in sequence by the Main Polling Loop.



2.2 Boot Up

2.2.1 Boot Up Sequence

On power-up SSP enters a boot sequence as shown below. During most of the steps in this sequence, a value is written to the Mode field of the SSP Status Word, which can be a useful diagnostic aid.

Step	Mode	Action
1	N/A	Save stack pointer for diagnostics
2	N/A	Set up stack pointer
3	N/A	Save registers RO-R14 for diagnostics
4	N/A	Clear all of high ram area
5	N/A	Disable Interrupts
6	N/A	Initialise routing bus
7	0	Initialise parallel I/O ports
8	0	Run initialisation patch (if present)
9	1	Check Ram
11	2	Copy PROM Code to RAM
12	3	Turn 12V Converter A on
13	4	Start Task Scheduler
14	5	Start watchdog and BCP handlers
15	6	Reset command handler and synchronise DDB to BCP
16	0	Wait for 10 seconds to allow EEPROM override commands
17	1	Read patches from EEPROM
18	2	Initialise telemetry handler
19	2	Still going flag toggled
20	1	Set current SSP mode in status word (nominally 1)
21	1	Jump to main polling loop.

2.2.2 Use of EEPROM for Patching

The EEPROM is intended to hold updates to the baseline SSP flight software. The updates are held as individual patches. These patches are read into memory during initialisation.

There are two types of patches:

a) The initialisation patch which is loaded into memory as soon as the processor is able to access the EEPROM and then the patch is executed. The function of this patch is to be able to rectify problems that occur during the initialisation phase of the SSP s/w. The patch can either make small modifications to the flight s/w or can replace entire sections of the flight s/w.

b) The second type of patch is loaded once the flight s/w has executed its initialisation sequence and has been copied from PROM to RAM. There is a ten second delay between initialising the command handler and command FIFO, and loading the patches to allow the patch loading to be overridden by a telecommand.

There can be several patches each of which contains modification to the flight s/w or additions to it. In each case the patch is read from the EEPROM into spare memory where the patch CRC is checked. If the patch CRC is OK then the patch is copied to the appropriate memory location.

The EEPROM Memory Map is as follows:

Address	Label	Comments
0	EE-Patch List	Start address of patch list

The patch List Structure is as follows:-

Offset (Words from EE-PatchList)	Field	Comments
0	PatchNumber	Number of patches in list
1	Address Patch 1	EEPROM address of 1st patch
2	Address patch 2	EEPROM address of 2nd patch
:	:	: : :
n	Address patch n	EEPROM address of last patch

The structure of a patch is as follows:-

Offset	Field	Comments
0	Length	Length of patch data including CRC
1	Memory Address	Address to write patch to when loaded from EEPROM
2	Patch Data	1st Word of Patch Data
3	Patch Data	2nd Word of Patch Data
:	: :	: : :
Length+1	Patch Data	Last Word of Patch Data
Length+2	CRC	CRC

These commands associated with patches:-

Write Patch Writes patch data to the EEPROM, the patch CRC is verified before the patch is written.

Read PatchReads a patch and transmits the patch data via the engineering datastream.

Delete Patch Deletes a patch by clearing the patch address in the patch list.

Clear EEPROM Can be used to remove all patches.

2.3 Software Execution

After boot up, the SSP software performs some initialisation then enters its normal execution. Tasks are divided into two categories: Time critical and not so time critical. The critical ones are executed under interrupt control via a 'Task Scheduler'. The others run in a sequential fashion via the 'Main Polling Loop' during the gaps allowed by the task scheduler.

2.3.1 Task Scheduler

Module	TK.	
Directory	KERNEL	
Files:	Code	Task.ASM
	Variables	Task.VAR
	Constants	Task.COM
	Macros	Task.MAC
	Make	Task.MAK

The task scheduler enables a variety of data acquisition tasks that have different sampling strategies to be serviced whilst still maintaining accurate timing. A task has certain requirements:

i) The task has to be run at pre-defined intervals e.g. a complete sample every 10 seconds. This has been defined as the cycle time.

ii) The task consists of a number of functions that have to be carried out in a certain order e.g. initialise sensor, take n readings, process data, transmit data.

iii) Each function may have to be run at a specific time e.g. 10 readings taken at 1ms intervals.

The task scheduler uses Processor Timer A to generate an interrupt every 1ms. When the scheduler is activated by the interrupt it scans the current task table.

Each entry in a task table consists of 4 fields.

TK-Task Address	The address to start executing the task
TK-Counter	The number of timeslices used from the current cycle
TK-Time Slices Used	The number of timeslices left before the task is rescheduled
TK-Cycle	The cycle time of the task in timeslices

For each entry in the task table the scheduler will decrement the counter TK-Counter. If the counter becomes 0 then the code at the address in TK-Task Address is executed. The task runs until it hands back control to the Task Scheduler. The task is able to use registers RO-R6 as they are saved by the Timer A interrupt handler. Register R6 holds the address of the current task entry in the task table. All remaining registers should be preserved.

A task can hand back control to the task scheduler via one of several macros:

TK-Pause xx	Suspends execution of the task for xx timeslices. The value xx is loaded into TK-Counter and added to TK-Time Slices Used. Control is then handed back to the task scheduler.
TK-Wait Rx	Same as TK-Pause except the number of timeslices to suspend execution for is held in register Rx. Only registers R0-R5 can be used.

TK-Cycle	Suspends execution of the task until a known time from the time the current task cycle was started. The number of time slices used since the start of the task, TK-TimeSlicesUsed is subtracted from the overall cycle time TK-Cycle. The result is then located into the task counter TK-Counter. The number of time slices used. TK-TimeSlicesUsed is set to 0.
TK-LongWait Rx Ry-	Suspends execution of a task for longer than 65536 timeslices. This is similar to TK-Wait except Rx and Ry hold the 1sw and msw of the delay time respectively. TK-Counter is used as before being loaded with 1sw of the delay time. TK-Time Slices Used is aliased as TK-CounterHi and is loaded with the msw +1 of the delay time. Every time the task scheduler schedules the task the macro code decrements TK-CounterHi until it also becomes zero and the task is resumed. Note that TK-Cycle will not take account of the time TK-LongWait.

Task Start-up

At each SSP mode change a different sampling strategy is required. To do this the task table has to be changed to reflect the change in the sampling strategy.

At a mode change the Task Handler loads TK-Task Table with a new task table held in program memory. Originally these will have been in PROM and will have been loaded into RAM at power on. They can be modified either by telecommand or with replacements held in EEPROM. These start-up task tables are identical in form to the running task table and are simply copied to TK-Task Table at the appropriate time.

The format of a startup task table entry is as follows:

TK-TaskAddress	Start address of the task
TK-TimeSlicesUsed	0
TK-Count	The number of timeslices after the table is loaded before the task actually runs.
TK-Cycle	The cycle time of the task. This is set here and is not altered (usually) by the task.

The start-up task table has to be constructed so that there is minimum interference between tasks. Effectively this means that no two tasks should be scheduled to run in the same time slice. This is not always possible but in the majority of cases where exact timing is necessary then this should be achievable. Most of the SSP tasks run at fairly well defined intervals such as every second or some whole number of seconds. Here the initial value of TK-Count can be used to stagger these tasks so that they do not conflict. There are complications caused by tasks that run for a large successive number of timescales e.g. a REF readout and those that have non uniform sampling timing e.g. THP and those that have to be sampled very fast e.g. ACC-I.

One problem with reloading the task table is that it is not known what state the existing tasks were in. Ordinarily, this does not matter but it will impact the handling of the data streams. If task is terminated then the data stream that it is loading to may not have a complete sample in it. Any incomplete data stream packets are padded out to their normal length to simplify data processing.

2.3.2 Main Polling Loop

The following tasks are executed in sequence:

START:	Command Packet Handler
	Telemetry Packet Handler
	ACC-E Impact Processing (if impact detected)
	Main Processing (see below)
	Telemetry Packet Handler
	Increment Poll Count (reset on BCP)
	Jump to Start.

2.3.3 Main Processing

These background tasks are executed in priority order when the main polling loop reaches "Main Processing". The tasks are:

1.	ACC-I	Impact Processing
2.	ACC-I	Data Processing (all modes)
3.	API-S	Data Processing
4.	REF	Data processing.

2.4 SSP Modes

The SSP experiment has eight modes.

MODE	DESCRIPTION	STARTS AT
0 (or 8)	EMC Test	Command Only
1	Upper Atmosphere	t_o
2	Mid Atmosphere	t_o + 10 Minutes
3	Lower Atmosphere	t_o + 85 Minutes
4	Proximity	Attitude = 7km or t_o + 120 min.
5	Surface	Impact or Proximity + 31 Min.
6	Extended Surface	Impact + 3 Minutes
7	Diagnostic	Command Only

The modes each define a different data sampling scheme. A summary may be found in Appendix D - Excel sheet FMDATA.XLS

Modes 1-6 only are used during the nominal mission. The modes are entered automatically in response to the Descent Data Broadcast (except for Proximity and Impact explained later).

The instrument may be commanded into any mode. In this case the command contains a parameter which can be used to freeze SSP into that mode; otherwise it may change autonomously based on certain criteria.

2.4.1 Mode 0 or 8 (EMC)

In Mode 0 only housekeeping packets are generated. All available packets are filled, so the housekeeping is sampled much faster than normal. This mode is used during ground test and is especially useful during EMC testing.

Mode 0 is not used during a nominal descent.

2.4.2 Mode 1 (Upper Atmosphere)

When SSP is switched on it starts in Mode 1. After it has locked to the DDB, the mode may then change if appropriate.

In Mode 1 DEN, PER, TIL and HK are regularly sampled. There are also stimulation (self test) sequences for ACC-E, ACC-I and THP sensors. There are 5 THP samples:

Wire 0 : A/D Converter Cross-Calibration
Wire 1 : Low Current Only
Wire 2 : Low Current Only
Wire 3 : Low Current Only
Wire 4 : Low Current Only

Mode 1 corresponds to the period when the CDMU polling rate is 3 packets/16 sec. SSP leaves Mode 1 at T_o + 10 minutes when CDMU polling changes to 8 packets/16 sec.

2.4.3 Mode 2 (Mid Atmosphere)

On entering Mode 2, ACC-E is stimulated (and packets generated) for checkout purposes. Thereafter the normal sampling scheme is followed for all sensors; see Appendix D.

In Mode 2 SSP can adapt to an increase in polling rate if this occurs, by increasing the number of API-S samples. This will generate an extra 1.5 SSP packets/16 secs at most.

SSP leaves Mode 2 at T_o + 85 minutes when CDMU polling changes to Step 3 (11 SSP packets/16 sec).

2.4.4 Mode 3 (Lower Atmosphere)

On entering Mode 3, ACC-E is stimulated for checkout purposes. Other sensors continue regular sampling. API-S is already sampling at its maximum rate of 0.7 Hz so no extra packets can be generated. This means that SSP can only fill about 90% of its nominal allocation (970 out of 1138).

2.4.5 Mode 4 (Proximity)

SSP enters proximity mode at an altitude of 7 km or at a time out of (T_o + 120 minutes).

ACC-I is sampled continuously at 500 samples/sec and REF sampling drops to only one sample per 15 mins.

API-S reverts to adaptive sampling and transmits extra data in proximity mode. At an altitude of 1km (reported in the probe DDB) the API-S pulse length changes to 2 mS for enhanced accuracy (N.B. This change is only made if in the DDB Word 5 bit 15 is set to 0, i.e. "measured altitude". This prevents the 2 ms operation taking place if the ≤ 1km altitude is an estimated value rather than the real altitude derived from the RAU A and RAU B. The practical outcome of this is that for the Titan mission data the API-S pulse width would be reduced to 2ms at 1km indicated altitude, but would switch back to 10 ms pulse width at a lower altitude once the RAU were unable to function and the DDB altitude bit was reset to 1, i.e. "predicted altitude".).

SSP can increase its packet usage to approx 22 packets/16 secs in Mode 4.

2.4.6 Mode 5 (Surface)

When impact is detected SSP enters Mode 5 (the impact detection algorithm is discussed later). At this point some special processing takes place to build an Impact Packet before normal sensor sampling recommences.

The Impact Packet contains processed ACC-E and ACC-I impact data plus new samples from the other sensors. Impact times as detected by ACC-E and ACC-I are included. Impact Packet building takes about 1.5 secs. As a high priority the spacecraft interface FIFOs are loaded with eight copies of the impact data packet.

Normal packet building and sensor sampling then recommences. There will be some data left over from pre-impact (say 6 packets) still to be transmitted, and the complete ACC-E and ACC-I impact buffers and the surface REF readout also join the data output queue. At (T impact + 15 secs) a THP sample is taken.

Transmission of packets immediately after impact:

TYPE	NUMBER OF PACKETS	APPROX. DURATION SECS
Last Pre-impact	1 or 2	2
Impact	4	5
Remaining pre-impact	6 max	8
ACC-E	5	22
ACC-I	7	22
REF	12	22
THP	1	22

Impact Processing

After Impact has been detected some special processing takes place to build the Impact Packet before normal sensor sampling resumes. The tasks are listed below:-

Send some existing data streams		
PER sample		
Trigger APIS and APIV		
2 DEN Samples - write to Impact Packet		
TIL - write to Impact Packet		
Send Tele		
REF X3	→	REF datastream
REF Calc	→	Impact Packet
Send tele		
Process ACC-I		Impact Packet
ACC-I Buff	→	datastream
Process ACC-E	→	Impact Packet
Send ACC-E buff if non-zero		
Read APIV	→	Impact Packet
Send Tele		
Process AP1S	→	Impact Packet
Send APIS buff		
Send Tele		
Read PER and CON	→	Impact Packet
Send IMP x 4		

2.4.7 Mode 6 (Extended Surface)

Three minutes after impact SSP starts Mode 6. The principal change is that API-S switches to a mode optimised for ocean depth sounding. Ten pulses are emitted followed by a break of 20 seconds.

2.4.8 Mode 7 (Diagnostic)

In Mode 7 no telemetry packets are generated except on command. This mode may be useful for debugging hardware faults which may develop during cruise.

2.5 Proximity Mode and Impact Detection Algorithm

Changing into Proximity Mode

The primary concern is to ensure that Proximity Mode is entered before the Probe impacts the surface for the minimum descent scenario and maximum Titan topography, for whatever failures occur on the Radar Altimeters or Descent Data Broadcast (DDB). However, entry into Proximity

Mode at too high an altitude is to be avoided since the REF sensor is not read in Proximity Mode and any potential atmospheric data from the ACC-I sensor is reduced.

No notice is taken of whether the Altimeter data in the DDB is real or predicted. The expected errors in the predicted altitude at 7km are small provided that at least one radar altimeter has functioned by that altitude. For example, if the altimeter worked but failed at 10km then the error at 7km from plotting the nominal descent gradient would be -200m for the Minimum Descent and +100m for the maximum descent.

By increasing the nominal altitude for the SSP mode switch into proximity mode to 7km and selecting a timeout of 120 minutes we can achieve this. For any failure Proximity Mode is entered at a minimum altitude of 2.5km above the nominal 0km.

Allowing for the 10% accuracy on the radar altimeters, the maximum altitude at which SSP can enter Proximity Mode is 7.7km.

Changing into Surface Mode

The primary concern here is not to enter Surface Mode too early, or else the impact data could be missed. However we definitely need to get to Surface mode before the end of the mission, so a timeout is required in case of failure of the impact detection mechanisms. If Surface Mode is never entered then any data frozen in the ACC-E buffer would be lost, and other sensors would not sample in their surface modes.

After Proximity Mode is entered there is no requirement to know by which method the mode was selected. The change into Surface Mode should be initiated by an SSP algorithm which uses the signal from the internal accelerometer (ACC-I); this will detect a solid or liquid impact. If this should fail the timeout comes into operation; this timeout is set to 31 minutes after Proximity mode, the maximum descent time expected.

In the event that switching to Surface Mode is initiated by the timeout, the time that the Probe may have been on a liquid surface at the nominal 0km altitude would be between 6.7 and 24 minutes. This depends on the actual descent profile and whether Proximity Mode was entered by altitude or timeout. For proximity mode entry by altitude data the maximum time on a liquid surface would be 12.9 minutes (which would be a realistic time for Probe/Experiment survival). Further details are given in Figure 2.4.1.

Mode selection flow chart:

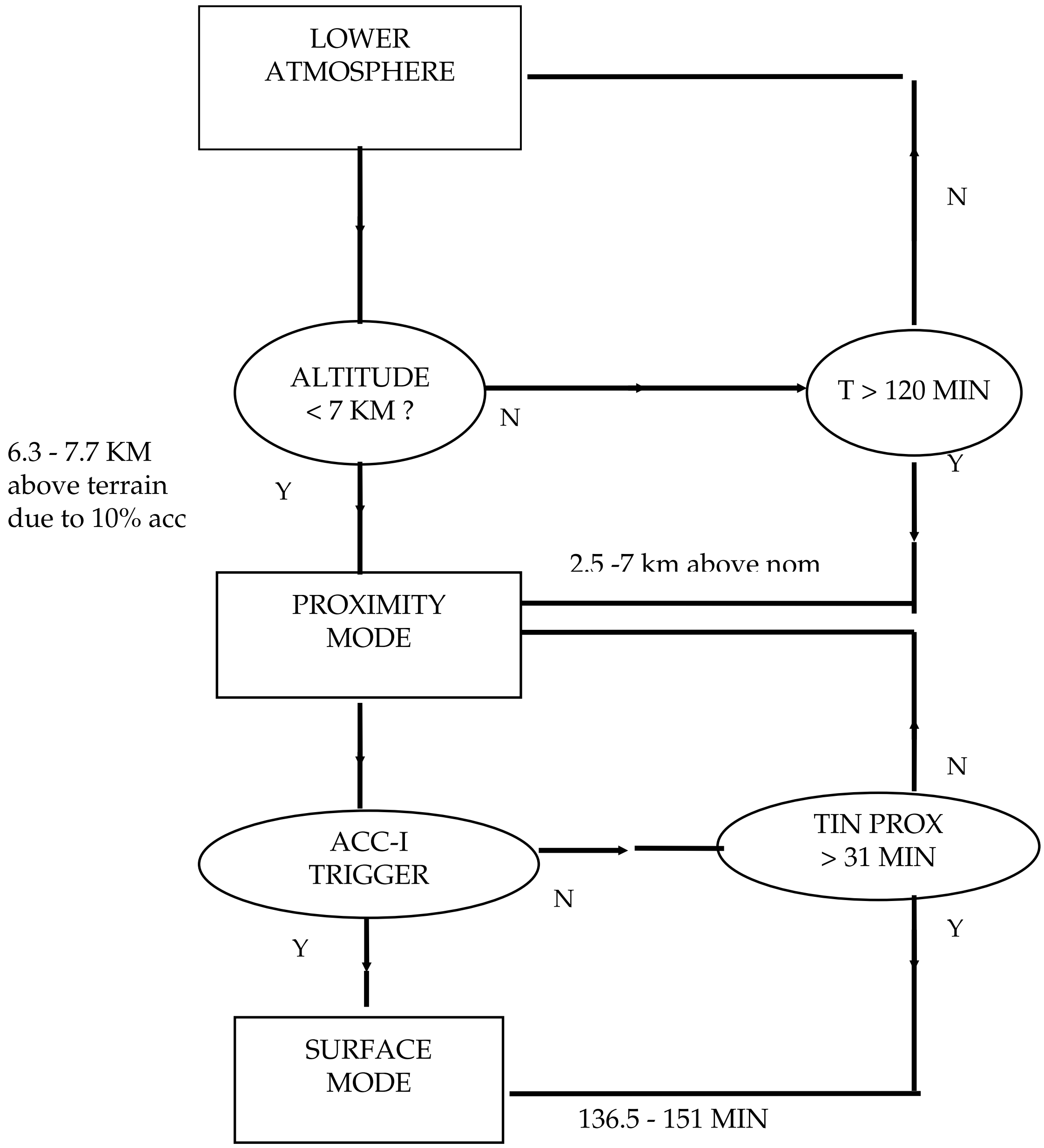


Figure 2.4.1 SSP Mode Change Fault Tree (7 km Proximity Altitude)

3. DATA HANDLING

3.1 Mission Status - The Descent Data Broadcast

Mission status is obtained from the descent data broadcast (DDB) message that is sent to SSP from the CDMS every two seconds. There are four items of information contained in this message; the mission time, the altitude of the spacecraft, the mission phase and the spin rate of the spacecraft. These items are copied into the SSP housekeeping packets.

3.1.1 Mission Time

The mission time within the DDB contains 2 bits which define before or after T0. As SSP is always switched on after T0 these two LSBs are ignored. The remaining 14 bits define the mission time to a resolution of 2 seconds, with full scale of 16384.

15 8	7 0
t t t t t t t t t t	t t t t t t x x

t - 14 bit mission time
x - don't care

3.1.2 SSP Time

The mission time is combined with the 10 bit time from the onboard millisecond counter to form a 24 bit time SSP Time value that has a resolution of 2ms. Full scale of SSP time would be 16384 secs. or 4 hours, 33 minutes, 4 seconds.

23 16	15 8	7 0
t t t t t t t t	t t t t t t m m	m m m m m m m m

t - 14 bit mission time
m - 10 bit millisecond counter time

3.1.3 Mission Phase

This is an 8 bit value that is defined by ESA as follows:

00	-	Entry/Descent
03	-	Ground Checkout
0c	-	Ground Checkout Suspended
OF	-	Ground Checkout De-activated
F3	-	Cruise Checkout
FC	-	Cruise Checkout Suspended
FF	-	Cruise Checkout De-activated

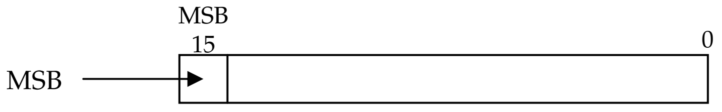
See EID Part A Section 3.6.3.8.

The only use that SSP makes of this information is check for "Cruise Checkout" phase. In this case TIL is not enabled and THP high current is not switched on. "Suspend" is not sensed; SSP detects changes in mission time only and makes appropriate mode changes.

In ground checkout phase a nominal descent mission is followed with timeout to Surface Mode (no impact), unless ACC-I is stimulated by command, electronically or physically as part of a test.

3.1.4 Altitude

The altitude is stored as a 16 bit word as defined in EID Part A Section 3.6.3.8. The altitude will be used to enter proximity mode at a height of 7km.

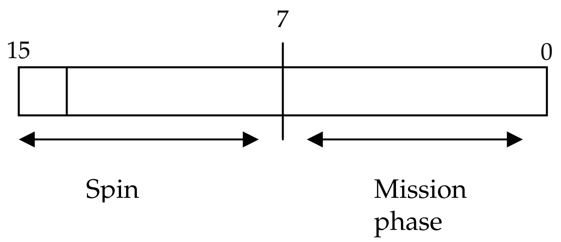


1 = Predicted Altitude
0 = Measured Altitude

LSB = 10 m

3.1.5 Spin

The spin will be stored in an 8 bit byte, at present it is not used.



LSB = 0.1 rpm

3.2 Packet Telemetry

3.2.1 Concept

Huygens uses a packet telemetry system as defined by ESA standards. Huygens packets as sent by SSP to the CDMU are 126 bytes long, of which 118 bytes is available for actual sensor data. The sensors are required to be sampled at rates varying from 1 second to 15 minutes and each data sample can have a size varying from 2 to 1k bytes.

To provide a uniform method of data handling we invented the 'Data Stream' concept, in which each sensor fills its own buffer until a predetermined number of bytes have accumulated. This number can be different for each sensor and mode of SSP. The data are combined with a header and trailer to form a 'Datastream Packet'. Datastream Packets fit within the Huygens packets (in the SSP data field) and are arranged to be an exact fit wherever possible; however for sensors generating large volumes of data per sample (THP, API-S, REF, ACC-E, ACC-I) they spread over several Huygens packets.

The data analysis system for SSP must recognise first Huygens packets, then identify the Datastream type and synchronise on the Datastream packet structure.
For a pictorial overview of data sample formats and complete packets see Appendix A.

3.2.2 Packet Headers

3.2.2.1 Huygens Packets

The Huygens packet consists of 126 bytes :-

Packet ID (2 bytes)	Sequence Control (2 bytes)	Packet Length (2 bytes)	SSP Data Source (2 bytes)	SSP Data (118 bytes)

Packet ID — Value = 0x0F94 for CDMU-A
Value = 0x0FB4 for CDMU-B

Sequence Control — Value = 11xxxxxx xxxxxxxx
where xxxxxx xxxxxxxx is a 14 bit Huygens packet counter.

Packet Length — Value = 0x0077

SSP Data Source — Value = cccccccc cccciiii
where cccccccc cccc is a 12 bit datastream packet counter.
where iiii is the Datasteam ID, see below:

SSP Data — is the field used by datastream packets, see the next section.

Datastream ID	Data stream
0	Engineering
1	Impact
2	ACC-I
3	API-S
4	API-V
5	DEN
6	PER
7	REF
8	THP
9	TIL
A	Housekeeping
B	ACC-E

3.2.2.2 Datastream Packets

The datastream packet format is the same for each stream, however the length is variable. Headers and trailers are fixed (8 bytes total). Details of each datastream packet type may be found under the section for each sensor. To fit in a Huygens packet, the data length would be 110 bytes.

Start Sync (2 bytes)	Time (3 bytes)	Mode (1 byte)	Data Samples (variable)	End Sync (2 bytes)

Start Sync Value = 0x8888

Time is the SSP Time when the first data sample was made.

Mode Value = ffff mmmm
where ffff is mostly unused (uses listed under sensors).
where mmmm is the SSP mode when the samples were taken.

Data Samples detailed under sensor sections following.

End Sync Value = 0x9999

3.2.3 ACC-E

3.2.3.1 Sensor Sampling

An ACC-E sample consists of 512 x 8 bit values. Sampling stops at impact so that the FIFO buffer contains the impact signature.

3.2.3.2 ACC-E Datastream Packet, ID = 1 (Modes 0, 5, 7)

Datastream Packet length = 520 bytes
Measurements/packet = 1 impact signature

This packet contains the full impact signature and is transmitted during the descent sequence only following the impact.

The packet can be identified by the Mode field in the Datastream Header i.e.
Mode Value = ffff mmmm
where ffff is the datastream packet ID = 0001
where mmmm is the SSP mode when the samples were taken.

Datastream Header (6 bytes)	Sensor Samples (512 bytes)	End Sync (2 bytes)

The Sensor Samples data field consists of the raw 512 x 8 bit sensor samples.

3.2.3.3 ACC-E Datastream packet, ID = 2 (Modes 1,2,3)

Datastream Packet length = 110 bytes
Measurements/packet = 1 compressed impact signature

This packet contains a compressed impact signature consisting of 4 calculated values plus a selection of raw data around the peak response.

Calculated values are :-

a) The peak value of the impact signature.
b) The offset of the peak value from the start of the impact signature.
c) The offset from the start of the impact signature of the 1/2 maximum point before the peak.
d) The offset from the start of the impact signature of the 1/2 maximum point after the peak.

These are stored in 4 x 16 bit words. These values are followed by 102 bytes of raw data from the impact buffer: the first four, 94 around the peak, and the last four; giving a total of 110 bytes which fits in a Huygens packet.

Field	Offset	Size (Bytes)
Max Signal	0	2
Max Position	2	2
Pre-Max FWHM position	4	2
Post-Max FWHM position	6	2
First 4 samples in buffer	8	4
94 peak samples	12	94
last 4 samples in buffer	106	4

Figure 3.2.1.1 ACC-E Compressed Impact Packet data field

3.2.4 ACC-I

3.2.4.1 Sensor Sampling

A single data sample from ACC-I consists of a 12-bit value that will be stored in a 16-bit word. The sample rate is 500 samples/sec.

3.2.4.2 ACC-I Datastream packet, ID = 0 (Modes 1,2,3,6)

Datastream Packet length = 110 bytes
Measurements/packet = 18

In these modes, one measurement consists of 200 samples, processed to give three 16 bit words which hold the maximum sample value, the mean and the variance. Used in Modes 1, 2, 3, 5, 6.

Field	Offset (Bytes)	Size (Bytes)
Maximum Value	0	2
Mean	2	2
Variance	4	2

The packet consists of 18 processed samples as above followed by 2 bytes padding with a value of 0x9999, giving a total datastream packet length of 118 bytes.

3.2.4.3 ACC-I Datastream packet, ID = 1 (Modes 0,5,7)

Datastream Packet length = 768 bytes
Measurements/packet = 1 impact signature

The ACC-I impact signature consists of 512 single data samples. ACC-I is sampled at a rate of 500Hz during the proximity phase. The samples are stored in a 512 x 16-bit word FIFO. The sampling is frozen on impact so that the 512 samples obtained show the acceleration profile when impact occurred. Once the FIFO has been frozen the impact signature is packed into a 512 x 12 bit word array for transmission.

3.2.4.4 ACC-I Datastream packet, ID = 2 (Mode 1)

Datastream Packet length = 110 bytes
Measurements/packet = 1 compressed impact signature.

An ACC-I compressed impact signature has the same format as the ACC-I compressed signature. Four 16 bit values are calculated, these are:

a) The peak value of the impact signature.
b) the offset of the peak value from the start of the impact signature.
c) The offset from the start of the impact signature of the 1/2 maximum point before the peak.
d) The offset from the start of the impact signature of the 1/2 maximum point after the peak.

Field	Offset	Size (Bytes)
Max Signal	0	2
Max Position	2	2
FWHM position pre-peak	4	2

FWHM position post-peak	6	2
First 4 samples in buffer	8	6
60 peak samples	14	90
last 4 samples in buffer	104	6

Figure 3.2.1.2c ACC-I Compressed Impact Signature

3.2.5 API-S

3.2.5.1 Sensor sampling

An API-S return signal consists of 1024 samples, where each sample is an 8 bit value (i.e. raw data held in a 1 kilobyte array). Time step between each sample is 1 milli-second. To accommodate limited transmission bandwidth, three data reduction strategies are used depending on the mission phase as described below:-

3.2.5.2 Checkout (raw) Sample (mode = 0, 8)

Only used during in-flight/cruise or ground testing.
NOT USED DURING THE TITAN MISSION.
Datastream Packet length = 1010 bytes
Measurements/packet = 1

Field	Offset (Bytes)	Size (Bytes)	Description
Sync	0	2	Sync Word = 8888 hex
Time	2	3	
Mode	5	1	SSP Mode
Peak	6	2	Peak Position
Data	8	1000	Raw Data
Sync	1008	2	9999 hex

3.2.5.3 Atmosphere Sample (modes 2, 3)

Datastream Packet length = 36 bytes
Measurements/packet = 1 binned return

Samples 50 to 570 are extracted from the API-S return signal and binned into 26 bins of 20 (i.e. each bin gives the average of 20 samples).

For modes 2 and 3 ONLY the bins packing order in the telemetry packets is swapped pair-wise as shown by the figure below (i.e. "Little Endian" storage format).
(N.B. Titan PDS data are corrected but any older, non PDS complaint data sets may need correcting).

Actual order of bins (where each bin is average of 20 samples)

1	2	3	4	5	6

Order of bins in telemetry packets

2	1	4	3	6	5

3.2.5.4 Proximity Sample (modes 4, 5)

Datastream Packet length = 146 bytes

Measurements/packet = 1 compressed return

The Apis return signal is reduced using the following strategy:-

- Discard first 20 samples. (After this step, the 1st sample = sample 21st from raw data set).
- Determine the peak position (i.e. position of the API-S sample with the highest value) from remaining samples.
- Store peak position in the telemetry packet marked by "Peak". **Please note to recover correct peak position, an offset of 20 needs to be added to the peak position** (due to 20 discarded samples).
- If peak < 100, then set the peak to 100, for data reduction purposes. Else keep the actual peak position.
- Transmit the first +/- 30 samples around the peak without compression (i.e. raw data).
- Transmit the remaining samples between peak +/- 100 as bins of 4 (i.e. average 4 samples to give 1 value).
- Transmit all remaining samples as bins of 20.

In algorithm terms, the compression around the peak can be expressed as:-

```
Starting at first sample
while this sample number is less than the total no of samples
if this sample is within 30 samples of the peak sample
        compressed data equals raw data
look at next sample if this sample is within 100 samples of the peak sample
        compressed data equals the mean of the next 4 samples
look at fourth sample from this sample
if this sample is not within 100 samples of the peak sample
        compressed data equals the mean of the next 20 samples
look at 20th sample from this sample.
```

The net effect of this reduction strategy is to retain features around the peak from the API-S return signature (as long as the peak signature is after 120 milli-seconds from the start of the return).

3.2.5.5 Extended Surface Sample (mode 6)

Datastream Packet length = 646 bytes
Measurements/packet = 1, consisting of 10 pulses.

Every Depth Sounding:
10 API-S pulses are sent approximately 1 second apart. The data from all 10 measurements are compressed by binning each into 50 bins of 20 samples. Since the first 20 samples are discarded, the 1st bin represents the average of API-S return from 21ms to 40 ms .
The return which has the maximum signal is also processed as for proximity mode.

Field	Offset (Bytes)	Size (Bytes)	Description
Sync	0	2	Sync Word = 8888 hex
Time	2	3	Time at which sample taken
Mode	5	1	Mode = 6
Peak	6	2	Peak Position
API-S Data	8	136	Peak Sample
API-S Data	144	500	10 Compressed Samples
Sync	644	2	Sync Word = 9999

3.2.6 API-V

3.2.6.1 Sensor sampling

An API-V measurement consists of two 12 bit values. These are a measure of the local velocity of sound in forward and reverse directions.

12 Bits	12 Bits
VEL 1	VEL 2

Figure 3.2.2.1a API-V Sample

Once two measurements have been collected these will be compressed into 3 x 16 bit words:-

word	0	1	2
Each digit represents 4 bits -->	1 1 1 2	2 2 1 1	1 2 2 2

3.2.6.2 API-V Datastream packet (all modes)

Datastream Packet length = 118 bytes
Measurements/packet = 36

3.2.7 DEN

3.2.7.1 Sensor Sampling

A DEN measurement consists of a single 12 bit value.
Four samples are compressed into 3 x 16 bit words.
A Den offset measurement will also be made and placed in the housekeeping packet.

3.2.7.2 DEN Datastream Packet

Datastream Packet length = 118 bytes
Measurements/packet = 72

2 bytes of padding, value = 0x9999 are appended to the end of the packet.

3.2.8 PER and CON

3.2.8.1 Sensor Sampling

PER and CON measurements are each single 12 bit values.
Four 12 bit values are packed into 3 x 16 bit words:

1	2	3
P E R C	O N P E	R C O N

(P E R C | O N P E | R C O N) = 1 measurement

PER and CON offset measurements will be placed in the housekeeping packet.

3.2.8.2 PER and CON Datastream packet

Datastream Packet length = 118 bytes
Measurements/packet = 36

2 bytes of padding, value = 0x9999 are appended to the end of the packet.

3.2.9 REF

3.2.9.1 REF Sensor Sampling

A REF sample consists of 512 x 8 bit values. A complete REF readout sequence consists of 3 samples, one with the internal illumination LED on, one with no illumination and a third with the external illumination LED on.

3.2.9.2 REF Datastream packet

Datastream Packet length = 520 bytes
Measurements/packet = 1

Each packet contains a single REF readout. The illumination mode is indicated in byte 6 of the datastream header:

Mode Value = iiii mmmm
where iiii is the illumination mode. 0 = dark
1 = external
2 = internal
where mmmm is the SSP mode.

3.2.10 THP

3.2.10.1 THP Sensor Sample

A THP sample consists of a series of 16 bit measurements: the preamplifier offset, initial temperature value, and 60 x temperature values, each value being the average of 16 ADC samples.

3.2.10.2 THP Datastream packet, wires 1-4 (ID = 0/1)

Datastream Packet length = 132 bytes
Measurements/packet = 1

For THP the wire number is indicated in byte 6 of the datastream header:

Mode	Value = wwww mmmm			
	where	wwww	is the wire number	1 = 25 microns diameter
				2 = 25 microns diameter
				3 = 10 microns diameter
				4 = 10 microns diameter
	where	mmmm	is the SSP mode.	

Data field :

Offset (bytes)	Length (bytes)	Name
0	2	Preamp offset
2	2	Initial Temperature
4	120	60 Averaged Temperature Values

3.2.10.3 THP Datastream packet, wires 0 (ID = 2)

Datastream Packet length = 132 bytes
Measurements/packet = 31 samples, 12bit + 16 bit

In mode 1 a single checkout packet is generated to inter-calibrate the 12 bit and 16 bit A/D converters in SSP. The THP D/A converter is used as a signal source to generate a staircase waveform of 31 levels. Readings from each A/D converter are assembled into a packet which has the wire number set to zero. ('wwww' field above).

3.2.11 TIL

3.2.11.1 TIL Sensor sample (modes 1-4 @ 1Hz)
(modes 5 + 6 @ 2Hz)

A TIL data sample consists of 4 x 12 bit values which correspond to a voltage range of -5V to +5V. These values are stored in 16 bit words:

6 Bytes

TLXH	TLXL	TLYH	TLYL

In addition, the excitation voltages and offsets are measured and placed in the Housekeeping packet.

TIL Excitation Voltages:

6 Bytes

TOPXH	TOPXL	TOPYH	TOPYL

TILOffset Voltages:

6 Bytes

TLXO	TLYO	TOPXO	TOPYO

3.2.11.2 TIL Datastream packet

Datastream Packet length = 118 bytes
Measurements/packet = 18

2 bytes of padding, value = 0x9999 are appended to the end of the packet.

3.2.12 Housekeeping

3.2.12.1 Housekeeping Samples

There are 11 temperature sensors sampled by SSP; eight diodes, one IC and two AD590s. They are all digitised to 12 bit accuracy.

The majority of voltage monitors are digitised to 12 bit accuracy. The exceptions are three values sampled by the 16 bit converter.

Various processor status words and bytes are also included in the housekeeping packet.

3.2.12.2 Housekeeping Datastream packet

Datastream Packet length = 110 bytes
Measurements/packet = 1

The housekeeping packet is built at the time it is required using variables currently in memory; the current values are updated by the main processing task. The 12 bit samples are packed (4 samples in 3 x 16 bit words). A list of housekeeping parameters and their position in the packet is given below. Where the parameter does not occupy a whole byte or word, the mask column shows which are bits contain the parameter.

Offset (Bytes)	Description	Size (Bytes)	Mask	Mnemonic
0	Huygens Packet Header - ID	2		HUYPKTID
2	Huygens Packet Header - Counter	2		HUYPKTCNT
4	Huygens Packet Header - Length	2		HUYPKTLEN
6	SSP Data Stream Count	2	0xFFF0	STRMCNT
6	SSP Data Stream ID (=0x0A)	2	0x000F	STRMID
8	Start Sync = 0x8888	2		STARTSYNC
10	SSP Time	3		SSPTIME
13	SSP Mode	1		MODE
14	Altitude	2		ALTITUDE
16	Spin Rate	1		SPIN
17	Mission Phase	1		PHASE
18	Time Last Mode Change	3		TIMELM
21	Last Mode	1		LASTMODE

Offset (Bytes)	Description	Size (Bytes)	Mask	Mnemonic
22	Altitude Last Mode Change	2		ALTLM
24	SSP Command Count	2		SSPCMDCNT
26	Broadcast Count A	2		BCASTCNTA
28	Broadcast Count B	2		BCASTCNTB
30	Command Packet Count A	2		CMDPKTCNTA
32	Command Packet Count B	2		CMDPKTCNTB
34	Telemetry Packet Count A	2		TMPKCNTA
36	Telemetry Packet Count B	2		TMPKCNTB
38	Command Error Count (Total)	1		CMDERRCNT
39	Command Error Code (Last Error)	1		CMDERRCD
40	Last Command Sequence Number	1		LCMDSQNO
41	Last Command Code	1		LCMDCODE
42	Engineering Packet Count	2	0xFFF0	ENGPKTCNT
44	Impact Packet Count	2	0xFFF0	IMPPKTCNT
46	ACC-I Packet Count	2	0xFFF0	ACCIPKTCNT
48	API-S Packet Count	2	0xFFF0	APISPKTCNT
50	API-V Packet Count	2	0xFFF0	APIVPKTCNT
52	DEN Packet Count	2	0xFFF0	DENPKTCNT
54	PER Packet Count	2	0xFFF0	PERPKTCNT
56	REF Packet Count	2	0xFFF0	REFPKTCNT
58	THP Packet Count	2	0xFFF0	THPPKTCNT
60	TIL Packet Count	2	0xFFF0	TILPKTCNT
62	HK Packet Count	2	0xFFF0	HKPKTCNT
64	ACC-E Packet Count	2	0xFFF0	ACCEPKTCNT
66	Temp 1 (THP Sensor Body)	3	0xFFF000	THPT
66	Temp 2 (REF Sensor Board)	3	0x000FFF	REFSENT
69	Temp 3 (REF Prism Tip)	3	0xFFF000	REFPRTIPT
69	Temp 4 (REF Prism Base)	3	0x000FFF	REFPRBASET
72	Temp 5 (PER Sensor Body)	3	0xFFF000	PERT
72	Temp 6 (Top Hat Foam)	3	0x000FFF	TOPHATT
75	Temp 7 (SSPE 1 - 16 bit A/D)	3	0xFFF000	THPADCT
75	Temp 8 (SSPE 7 - 5v Converter A)	3	0x000FFF	CONVT
78	Temp 9 (TIL Sensor)	3	0xFFF000	TILTT
78	Temp 10 (SSPE Reference)	3	0x000FFF	SSPEBOXT
81	Temp 11 (2.5v Ref)	3	0xFFF000	2V5REFT
81	2.5v Ref Mon	3	0x000FFF	2V5
84	4.5v REF Mon	3	0xFFF000	4V5
84	-9v Ref Mon	3	0x000FFF	M9V
87	+12v Mon	3	0xFFF000	P12V
87	-12v Mon	3	0x000FFF	M12V
90	+5v Mon	3	0xFFF000	P5V
90	Test Input Via 12 Bit A/D	3	0x000FFF	TEST
93	THP Voltage Ref - Gas (Via 12 Bit)	3	0xFFF000	VREFG
93	THP Voltage Ref - Liquid (Via 12 Bit)	3	0x000FFF	VREFL
96	ACC-E Pre-amp Output	3	0xFFF000	ACCEPRE
96	DEN Offset	3	0x000FFF	DENOFF
99	CON Offset	3	0xFFF000	CONOFF
99	PER Offset	3	0x000FFF	PEROFF
102	TIL Excitation X, +ve	3	0xFFF000	TOPXH
102	TIL Excitation X, -ve	3	0x000FFF	TOPXL
105	TIL Excitation Y, +ve	3	0xFFF000	TOPYH
105	TIL Excitation Y, -ve	3	0x000FFF	TOPYL

Offset (Bytes)	Description	Size (Bytes)	Mask	Mnemonic
108	TIL Offset X, Top	3	0xFFF000	TOPXO
108	TIL Offset Y, Top	3	0x000FFF	TOPYO
111	TIL Offset X	3	0xFFF000	TLXO
111	TIL Offset Y	3	0x000FFF	TLYO
114	ACC-I Offset	2		ACCIOFF
116	Errors	1		ERRORS
117	Status Byte - (Port C + A?)	1		STATBYTE
118	THP Voltage Ref - Gas (Via 16 Bit)	2		VREFG16
120	Test Input via 16 Bit A/D	2		TEST16
122	+5v Mon Via 16 Bit A/D	2		P5V16
124	End Sync = 0x9999	2		ENDSYNC

Bit Allocation for ERRORS - byte 116

Bit	Function
7	Timer Over-run (Task has exceeded 1mS)
6	EEPROM write timeout (Verification after 10mS)
5	BCP Failure
4	2 Sec Failure
3	DDB Failure
2	No DDB Sync
1	Memory Error
0 (LSB)	THP Wire Broken

3.2.13 Impact Packet

3.2.13.1 Impact Packet sample

The Impact packet contains a summary data set from all sensors, especially the accelerometer impact signatures. Section 2.4.6 describes how the packet is built at the beginning of mode 5. The packet is transmitted 8 times to maximise the chances of collecting this most important data from SSP.

3.2.13.2 Impact Datastream packet

Datastream Packet length = 118 bytes
Measurements/packet = 1

Parameter	Data Stream Packet Offset	Size (Bytes)	Description
Start Sync	0	2	= 0 x 8888
Impact Time	2	3	Mission Time at ACC-I Impact
MODE	5	1	= XY, where X=Packet_ID, Y=Mode
ACC-I Peak Position	6	2	
ACC-I Pre-Position	8	2	Half Height Point
ACC-I Post-Position	10	2	Half Height Point
ACC-I Samples	12	12	8 x 12 bit samples around peak
Spare	14	2	
ACC-E Sample Time	26	3	Mission Time at ACC-E Impact
Mode	29	1	= XY, where X=not used, Y=Mode
ACC-E Peak Position	30	2	
ACC-E Pre-Position	32	2	Half Height Point

ACC-E Post-Position	34	2	Half Height Point
ACC-E Samples	36	10	10 Samples Around Peak
REF Internal RI Point	46	2	Max Differential
REF Internal Raw Data	48	10	10 Samples Around RI Point
REF External RI Point	58	2	Max Differential
REF External Raw Data	60	10	10 Samples Around RI Point
API-S Sample Time	70	3	Mission Time
Mode	73	1	= 5
API-S Peak Signal	74	2	
API-S Peak Position	76	2	
API-S Pre-Position	78	2	Half Height Point
API-S Post-Position	80	2	Half Height Point
API-V Sample	82	4	3 Bytes APIV, + 1 spare
DEN Sample Time	86	3	Mission Time
Mode	89	1	= 5
DEN Sample	90	2	
PER Sample	92	4	PER + CON, 3 Bytes + 1 spare
TIL Sample Time	96	3	
Mode	99	1	= 5
TIL Sample	100	6	4 x 12 Bits
Temps 2 - 8	106	10	7 x 12 Bits
End Sync	116	2	= 0x9999

3.2.14 Engineering

3.2.14.1 Engineering Samples

Engineering samples are taken in response to commands only, generally used for memory-dump and fault diagnostic purposes.

3.2.14.2 Engineering Datastream Packets

Engineering packets have the following general format. Contents of the data field depend on the command which requested the engineering packet. Details are given in the following sections, with the command which requests the information.

Field	Size (Bytes)	Description
Sync	2	Sync Word = 8888 hex
Time	3	Time Command Executed
Result	1	Command Result, 0 = OK
Command ID	2	ID of Command Executed
Command Sequence Number	2	Number of Command Executed
Data	Variable	Data Generated by Command
Sync	2	Sync Word = 9999 hex

Figure 3.2.14.2 SSP Command Reply Format

3.3 SSP Status Words

CDMU B	Bit No.	CDMU A
Software Impact	15 (MSB)	Software Impact
Sounder Mode	14	Processor Watchdog OK
All OK	13	All OK
Command Count LSB	12	Command Count LSB
DDB Loss	11	SSP Mode MSB
BCP Loss	10	SSP Mode NMSB
Redundant 12v Enabled	9	SSP Mode LSB
Nominal 12V Enabled	8	12V OK
! Processor Valid	7	! Processor Valid
Impact	6	Impact
ROM +5V	5	ROM +5V
Bus B + 5V	4	Bus B + 5V
Bus A + 5V	3	Bus A + 5V
! ML FIFO Empty	2	! ML FIFO Empty
! TM FIFO B Empty	1	! TM FIFO B Empty
! TM FIFO A Empty	0 (LSB)	! TM FIFO A Empty

Meaning of Status Flag Bits

'!' indicates an active low status.

Bits 0 - 7 (identical in both words) are derived from hardware only.
Bits 8 - 15 (CDMU A) are software status, routed via J4-J5.
Bits 8 - 15 (CDMU B) are control bits and software status routed via J4-J5.

CDMU-A

B15	SW Impact	1 = LPP Algorithm based on ACC-I data has detected impact.
B14	-	-
B13	All OK	1 = SSP Self Test Complete.
B12	Cmd Count LSB	The LSB of the command counter, toggles on receipt of a valid command
B11-9	SSP Mode	0 = Checkout 1 = Upper Atmosphere 2 = Mid Atmosphere 3 = Lower Atmosphere 4 = Proximity 5 = Surface 6 = Extended Surface 7 = Diagnostic
B8	12V OK	
B7	! Processor Valid	0 = CDMU-A valid, 1 = CDMU-B
B6	Impact	1 = ACC-E or ACC-I has detected impact
B5	ROM+5V	1 = Processor is booting from PROM, 0 = Running from RAM.
B4	Bus B +5V	1 = Supply OK, 0 = latchup protection has tripped.
B3	Bus A +5V	1 = Supply OK, 0 = latchup protection has tripped.
B2	!ML FIFO Empty	0 = Command Buffer Empty
B1	!TM FIFO B Empty	0 = Telemetry Buffer B Empty
B0	!TM FIFO A Empty	0 = Telemetry Buffer Empty

CDMU-B

B15	S W Impact	1 = LPP Algorithm based on ACC-I data has detected impact.
B14	Sounder Mode	0 = Atmosphere; 1 = Proximity.
B13	All OK	1 = Self Test Complete
B12	Cmd Count LSB	The LSB of the command counter; toggles on receipt of a valid command
B11	DDB Loss	1 = loss, 0 = OK
B10	BCP Loss	1 = loss, 0 = OK
B9	Redundant 12V Enabled	1 = Enabled, 0 = Disabled
B8	Nominal 12V Enabled	1 = Enabled, 0 = Disabled
B7	! Processor Valid	0 = CDMUA, 1 = CDMU B
B6	Impact	1 = ACC-E or ACC-I has detected impact
B5	ROM +5V	1 = Processor is booting from PROM, 0 = Running from RAM.
B4	Bus B +5V	1 = Supply OK, 0 = Latchup Protection has Tripped
B3	Bus A +5V	1 = Supply OK, 0 = Latchup Protection has Tripped
B2	! ML FIFO Empty	0 = Command Buffer Empty
B1	! TM FIFO B Empty	0 = Telemetry Buffer Empty
B0	! TM FIFO A Empty	0 = Telemetry Buffer Empty

3.4 Commands

Commands are not used during the descent to Titan as there is no command link from earth. However they may be used during cruise, most likely to send software patches to SSP or to check those already in memory at launch.

3.4.1 Command Packet Format

The telecommand protocol is defined in EID Part A Section 3.6.3.6. The packet has a standard 6 byte Huygens Packet Header followed by SSP data as required. The first word of SSP Command Data indicates the number of SSP commands within this Huygens Command Packet. More than one SSP command may sometimes be used, for example when patching, as long as the total length is less than 256 bytes.

Huygens Packet Header			SSP Command Data		Error checking
Packet Identification (2 bytes)	Packet Sequence Control (2 bytes)	Packet Length (2 bytes)	Data Field Header (2 bytes)	Command Data (Variable)	CRC (2 bytes)

Packet Identification — Value = 0x1F94 for CDMU-A
Value = 0x1FB4 for CDMU-B

Packet Sequence Control — Value = 11cccccc cccccccc
where 11 are bits fixed at '1'
where cccccc cccccccc is a sequence counter, normally not used.

Packet Length — Value = length of data field in bytes + 1

Data Field Header — Value = 0xnnmm
where nn is the number of commands in this packet.
where mm is the number of command words.

Command Data — See next section

CRC — Cyclic Redundancy code, calculated from whole packet contents, is checked within SSP on reception.

3.4.2 SSP Command Format

Within the Command Data field (above) can be a single, or multiple commands of the format given below:

Field	Size (Bytes)	Description
Command ID	2	see list of Command IDs in table below.
Sequence Number	2	may be used to identify a number of command of the same ID.
Parameters	Variable	Command Parameters

A summary of all SSP commands so far implemented is given below. Details of each command may be found in the following sections.

Command ID (Code)	Function	Parameters	Eng. Packet?
0	Read RAM	Address, Length	Y
1	Write Ram	Address, Length, Data	
2	Read EEPROM	Address, Length	Y
3	Write EEPROM	Address, Length, Data	
4	Read PROM	Address, Length	Y
5	Verify RAM	Block (Low or High)	Y
6	Verify EEPROM	-	Y
7	Verify PROM	-	Y
8	Clear EEPROM	-	
10	Read DRB	DRB Address	Y
11	Write DRB	DRB Address, Data Word	
12	Execute DRB Sequence	Length, Commands	Y
12	Read I/O	I/O Address	Y
14	Write I/O	I/O Address, Data Word	
15	Execute I/O Sequence	Length, Commands	Y
16	Read Parallel Port	Port Number	Y
17	Write Parallel Port	Port, Data, Mask	
18	Write Telemetry	Channel, Length, Data Words	
19	Reset Data Stream	Channel Mask	
20	Read Processor Status	-	Y
21	Execute Code	Length, Code Words	
22	Send Housekeeping Pkt	-	
23	Read 12 bit A/D	Multiplexer Address, # Samples	Y
24	Read 16 bit A/D	Multiplexer Address, # Samples	Y
30	Stimulate ACC-E	Pulse Length	
31	Test ACC_E	Pulse Length	
32	Readout ACC-E	-	
33	Test ACC_I	Pulse Length	
34	Stimulate ACC-I	Pulse Length	
35	Readout ACC-I	-	
36	Readout API-S	Mode (for tele packet format)	
37	Sample API-V	# Samples	
38	Sample DEN	# Samples	
39	Sample PER	# Samples	
40	Readout REF	Illumination	
41	Readout THP	Wire Number / Mode	
42	Sample TIL		
50	Set SSP Mode	Mode Number, Lock Option	
51	Write Patch	Patch Number, Data	
52	Read Patch	Patch Number	Y
53	Delete Patch	Patch Number	

3.4.3 SSP Commands

3.4.3.1 Read RAM - 0

This command will dump any area of memory into an Engineering packet.

Field	Size (Bytes)	Description
Command ID	2	= 0x0000
Sequence Number	2	
Address	2	Address of RAM to be read
Length	2	Length of Data to be read in Words

Engineering Packet Reply

Field	Size (Bytes)	Description
Sync	2	= 0x8888
Time	3	Time Command Executed
Result	1	Command Result, 0 = OK
Command ID executed	2	= 0
Command Number	2	Number of Command Executed
Length	2	Length of Data Returned in Words
Data	Length * 2	Data Read from Ram
Sync	2	Sync Word = 9999 hex

3.4.3.2 Write RAM - 1

This command will write directly to RAM. Patching would normally be used instead.

Field	Size (Bytes)	Description
Command ID	2	= 0x0001
Sequence Number	2	
Address	2	Address of Ram to be Written
Length	2	Length of Data to be Written in Words
Data	Length * 2	Data to be Written to Ram

3.4.3.3 Read EEPROM - 2

Field	Size (Bytes)	Description
Command ID	2	= 0x0002
Sequence Number	2	
Address	2	Address of EEPROM
Length	2	Length of Data to be Read in Words

Engineering Packet Reply

Field	Size (Bytes)	Description

Sync	2	= 8888 hex
Time	3	Time Command Executed
Result	1	Command Result, 0 = OK
Command Code	2	Code of Command Executed
Command Number	2	Number of Command Executed
Length	2	Length of Data Returned in Worlds
Data	Length * 2	Data Read from EEPROM
Sync	2	Sync World = 9999 hex

3.4.3.4 Write EEPROM - 3

This command will write directly to EEPROM. Patching would normally be used instead.

Field	Size (Bytes)	Description
Command ID	2	= 0x0003
Sequence Number	2	
Address	2	Address of EEPROM to write
Length	2	Length of Data to be Read in Words
Data	Length * 2	Data to be Written to EEPROM

3.4.3.5 Read PROM - 4

Field	Size (Bytes)	Description
Command ID	2	= 0x0004
Sequence number	2	
Address	2	Address of PROM to be Read
Length	2	Length of Data to be Read in Words

Reply

Field	Size (Bytes)	Description
Sync	2	Sync World = 8888 hex
Time	3	Time Command Executed
Result	1	Command Result, 0 = OK
Command Code	2	Code of Command Executed
Command Number	2	Number of Command Executed
Length	2	Length of Data Returned in Worlds
Data	Length * 2	Data Read from PROM
Sync	2	Sync World = 9999 hex

3.4.3.6 Verify RAM Block - 5

Field	Size (Bytes)	Description
Command ID	2	= 0x0005
Sequence Number	2	
Block number	2	0 = Low Ram Block, 1 = High Ram Block

Reply

Field	Size (Bytes)	Description
Sync	2	= 8888 hex
Time	3	Time Command Executed
Result	1	Command Result, 0 = OK
Command Code	2	= 0005
Command Number	2	Number of Command Executed
Number of Errors	2	Number of Errors Found
1st Error Address	2	Address at which 1st Error Occurred
1st Error	2	Type of Error
2nd Error Address	2	Address at which 2nd Error Occurred
2nd Error	2	Type of Error
:	:	:
:	:	:
nt. Error Address	2	Address at which nt. Error Occurred
nt. Error	2	Type of Error
Sync	2	Sync World = 9999 hex

3.4.3.7 Verify EEPROM

This command calculates a CRC (checksum) of its contents and dumps via an engineering packet.

Field	Size (Bytes)	Description
Command ID	2	= 0x0006
Command Number	2	Sequence Number of Command

Reply

Field	Size (Bytes)	Description
Sync	2	= 8888 hex
Time	3	Time Command Executed
Result	1	Command Result 0 = OK
Command Code	2	= 0x0006
Command Number	2	Number of Command Executed
CRC	2	CRC Value of EEPROM
Sync	2	= 9999 hex

3.4.3.8 Verify PROM Contents

N.B. The 12V supply has to be turned off during PROM Verification
(Ground Test or Cruise Phase only)

Field	Size (Bytes)	Description
Command ID	2	= 0x0007
Command Number	2	Sequence Number of Command

Reply

Field	Size (Bytes)	Description
Sync	2	= 8888 hex
Time	3	Time Command Executed
Result	1	Command Result, 0 = OK
Command ID executed	2	= 0x0007
Command Number	2	Number of Command Executed
CRC	2	CRC Value of PROM
Sync	2	= 9999 hex

3.4.3.9 Clear EEPROM - 8

N.B. All Telemetry is Stopped for 2 Minutes

Field	Size (Bytes)	Description
Command ID	2	=0x0008
Command Number	2	

3.4.3.10 Read Data Routing Bus - 10

This command is a low level read for debugging and was never used.

Field	Size (Bytes)	Description
Command ID	2	= 0x000A
Sequence Number	2	
Address	2	DRB Address to be Read

Reply

Field	Size (Bytes)	Description
Sync	2	= 8888 hex
Time	3	Time Command Executed
Result	1	Command Result, 0 = OK
Command Code	2	= 0x000A
Command Number	2	Number of Command Executed
Data	2	Data Read from DRB
Sync	2	= 9999 hex

3.4.3.11 Write DRB - 11

Low level - not used.

Field	Size (Bytes)	Description
Command ID	2	=0x000B
Sequence Number	2	

Address	2	DRB Address to be Written to
Data	2	Data to be Written to DRB

3.4.3.12 Execute DRB Sequence

Low level - not used

Field	Size (Bytes)	Description
Command ID	2	0x000C
Sequence Number	2	
Length	2	Number of DRB Commands to Execute
DRB Command	2 or 4	1st DRB Command
DRB Command	2 or 4	2nd DRB Command
:	:	:
:	:	:
DRB Command	2 or 4	nth DRB Command

Reply

Field	Size (Bytes)	Description
Sync	2	= 8888 hex
Time	3	Time Command Executed
Result	1	Command Result, 0 = OK
Command ID	2	= 0x000C
Command Number	2	Number of Commands Executed
Length	2	Length of Data Returned (Words)
Data	Length * 2	Data Generated by Sequence
Sync	2	= 9999 hex

3.4.3.13 Read I/O

Low level - not used.

Field	Size (Bytes)	Description
Command ID	2	0x000D
Sequence Number	2	
Address	2	I/O Address to be Read

Reply

Field	Size (Bytes)	Description
Sync	2	Sync Word = 8888 hex
Time	3	Time Command Executed
Result	1	Command Result, 0 = OK
Command ID	2	ID of Command Executed
Command Number	2	Number of Commands Executed
Data	2	Data Read from 1/0 Port
Sync	2	Sync Word = 9999 hex

3.4.3.14 Write I/O

Low Level - not used.

Field	Size (Bytes)	Description
Command ID	2	0x000E
Sequence Number	2	
Address	2	I/O Address to be Written to
Data	2	Data to be written to I/O

3.4.3.15 Execute I/O Sequence

Low level - not used

Field	Size (Bytes)	Description
Command ID	2	0x000F
Sequence Number	2	
Length	2	Number of I/O Commands to Execute
I/O Command	2 or 4	1st I/O Command
I/O Command	2 or 4	2nd I/O Command
:	:	:
		:
I/O Command	2 or 4	nth I/O Command

Reply

Field	Size (Bytes)	Description
Sync	2	= 8888 hex
Time	3	Time Command Executed
Result	1	Command Result, 0 = OK
Command ID	2	= 0x000F
Command Number	2	Number of Commands Executed
Length	2	Length of Data Returned (Words)
Data	Length * 2	Data Generated by Sequence
Sync	2	= 9999 hex

3.4.3.16 Read Parallel Port

Low level - not used.

Field	Size (Bytes)	Description
Command ID	2	= 0x0010
Sequence Number	2	
Port	2	Port to Read

Reply

Field	Size (Bytes)	Description

Sync	2	= 8888 hex
Time	3	Time Command Executed
Result	1	Command Result, 0= OK
Command ID	2	0x0010
Command Number	2	Number of Commands Executed
Data	2	Data Read from Parallel Port
Sync	2	= 9999 hex

3.4.3.17 Write Parallel Port

Low level - not used

Field	Size (Bytes)	Description
Command ID	2	0x0011
Sequence Number	2	
Port	2	Port to Read
Data	1	Data to be Written to Port/Register
Mask	1	Bit Mask to use to Write to Port

3.4.3.18 Write Telemetry (to Data Stream)

low level - not used

Field	Size (Bytes)	Description
Command ID	2	0x0012
Sequence Number	2	Sequence Number of Command
Channel	2	Channel to Write to (Data Stream ID)
Length	2	Number of Words of Data to Write
Data	Length * 2	Data to Write to Telemetry Channel

3.4.3.19 Clear Telemetry Stream(s)

Low level - not used.

Field	Size (Bytes)	Description
Command ID	2	0x0013
Sequence Number	2	Sequence Number of Command
Channel Mast	2	Mask Selects Data Streams to be Cleared

3.4.3.20 Read Processor Status

low level - not used.

Field	Size (Bytes)	Description
Command ID	2	0x0014
Sequence Number	2	

Reply

Field	Size (Bytes)	Description
Sync	2	= 8888 hex
Time	3	Time Command Executed
Result	1	Command Result, 0 = OK
Command ID	2	ID of Command Executed
Command Number	2	Number of Commands Executed
Registers	32	Contents of Registers RØ-R15
Status Word	2	Contents of Status Word
Fault Register	2	Contents of Fault Register
Interrupt Mask	2	Contents of Interrupt Mask Register
Pending Interrupt	2	Contents of Pending Interrupt Register
Timer A	2	Timer A Value
Timer B	2	Timer B Value
m Counter	2	Millisecond Counter Value
Sync	2	= 9999 hex

3.4.3.21 Execute Code

low level - not used.

Field	Size (Bytes)	Description
Command ID	2	= 0x0015
Sequence Number	2	
Length	2	Size of Code (Words)
Data	Length * 2	Code to be executed

3.4.3.22 Send Housekeeping Packet

Field	Size (Bytes)	Description
Command ID	2	= 0x0016
Sequence Number	2	

3.4.3.23 Read 12 Bit ADC

Field	Size (Bytes)	Description
Command ID	2	= 0x0017
Sequence Number	2	
Mux Address	2	12 Bit Multiplexer Address
Number of Samples	2	Number of Samples

Reply

Field	Size (Bytes)	Description
Sync	2	= 8888 hex
Time	3	Time Command Executed
Result	1	Command Result, 0 = OK
Command ID	2	= 0x0017
Command Number	2	Number of Command Executed
Number of Samples	2	
Data	2 * No of Samples	Data Sampled from 12 Bit ADC
Sync	2	= 9999

3.4.3.24 Read 16 Bit ADC

low level - not used.

Field	Size (Bytes)	Description
Command ID	2	0x0018
Sequence Number	2	
Mux Address	2	16 Bit Multiplexer Address
Samples	2	Number of Samples

Reply

Field	Size (Bytes)	Description
Sync	2	= 8888 hex
Time	3	Time Command Executed
Result	1	Command Result, 0 = OK
Command ID executed	2	=0x0018
Command Number	2	Number of Commands Executed
Number of Samples	2	
Samples	2 * No of Samples	Data Sampled from 16 Bit ADC
Sync	2	Sync Word = 9999

3.4.3.25 Stimulate ACC-E

This command generates a voltage pulse at the ACC-E sensor stimulation electrode, which is enough to initiate an impact packet if SSP is in the correct mode.

Field	Size (Bytes)	Description
Command ID	2	0x001E
Sequence Number	2	
Duration	2	Pulse Length N x 4 µS

3.4.3.26 Test ACC-E

Not used with the FM software

Field	Size (Bytes)	Description
Command ID	2	0x001F
Sequence Number	2	
Count	2	Pulse Length = N x µS approx.

3.4.3.27 Readout ACC-E

Not used with FM software.

Field	Size (Bytes)	Description
Command ID	2	0x0020
Sequence Number	2	

3.4.3.28 Test ACC-I

Field	Size (Bytes)	Description
Command ID	2	0x0021
Sequence Number	2	
Count	2	Pulse Length in mS

3.4.3.29 Stimulate ACC-I

Field	Size (Bytes)	Description
Command ID	2	0x0022
Sequence Number	2	
Length	2	Pulse Length in mS

3.4.3.30 Readout ACC-I

Field	Size (Bytes)	Description
Command ID	2	0x0023
Sequence Number	2	Sequence Number of Command

3.4.3.31 Readout API-S

Field	Size (Bytes)	Description
Command ID	2	0x0024
Sequence Number	2	
Mode	2	Readout Data Format

3.4.3.32 Sample API-V

Field	Size (Bytes)	Description
Command ID	2	0x0025
Sequence Number	2	
Number of Samples	2	Number of Samples to be Taken

3.4.3.33 Sample DEN

Field	Size (Bytes)	Description
Command ID	2	0x0026
Sequence Number	2	
Number of Samples	2	Number of Samples Taken

3.4.3.34 Sample PER

Field	Size (Bytes)	Description
Command ID	2	0x0027
Sequence Number	2	
Number of Samples	2	Number of Samples to be Taken

3.4.3.35 Readout REF

Field	Size (Bytes)	Description
Command ID	2	0x0028
Sequence Number	2	
Illumination	2	0 = Dark, 2 = Internal, 1 = External

3.4.3.36 Readout THP

Field	Size (Bytes)	Description
Command ID	2	0x0029
Sequence Number	2	
Wire Number	1	1-4, or 0 = A/D Calibration
Mode	1	Determines Sample Table

3.4.3.37 Sample TIL

Field	Size (Bytes)	Description
Command ID	2	0x002A
Sequence Number	2	
Number of Samples	2	Number of Samples to be Taken

3.4.3.38 Set SSP Mode

Field	Size (Bytes)	Description
Command ID	2	0x0032
Sequence Number	2	
Mode	2	SSP Mode + Lock Flag

Normal unlocked modes (1-6) are used in descent.
Checkout modes are 7 (no housekeeping) and 0 (housekeeping).
The locked modes are 8-15.

3.4.3.39 Write Patch

Field	Size (Bytes)	Description
Command ID	2	0x0033
Sequence Number	2	
Patch Number	2	Number of Patch to write
EEPROM Address	2	Address in EEPROM to write patch to
Length	2	Length of patch data including CRC (excluding address)
Memory Address	2	Address to patch in memory
Data	2n	Patch data
CRC	2	CRC Check

3.4.3.40 Read Patch

Field	Size (Bytes)	Description
Command ID	2	0x0034
Sequence Number	2	
Patch Number	2	Patch number to read

Reply

Field	Size (Bytes)	Description
Sync	2	= 0x8888
Time	3	Time Command Executed
Result	1	Command Result, 0 = OK
Command ID	2	= 0x0034
Sequence Number	2	of Command Executed
Length	2	Length of patch
	2	
	2	
Data	Length 2	

3.4.3.41 Delete Patch

Field	Size (Bytes)	Description
Command ID	2	0x0035
Sequence Number	2	
Patch Number	2	Patch number to delete

4. SPACECRAFT INTERFACE HANDLING

4.1 Telecommands and DDBs

Telecommands and DDBs transmitted from either CDMU A or CDMU B are stored in a h/w FIFO which is part of the SSP spacecraft interface electronics. The LPP can determine if there is any data in the FIFO by reading the FIFO empty flag. The processor can also determine if the CDMU is in the process of transmitting data into the FIFO. When data is present the LPP can read data from the FIFO and process it. The LPP can also reset the FIFO, this is done on power up to clear any "garbage" that may be present in the FIFO.

4.1.1 Command Handler

Module	CM	
Directory	COMMAND	
Files:	Code	Command.ASM
	Variables	Cm.VAR
	Constants	Cm.CON
	Make	Command.MAK

The command handler is responsible for reading telecommands and DDBs from the telecommand FIFO and processing them.

The command handler is run from the main background polling loop. It is therefore interruptible by the task scheduler.

On entry the command handler checks to see if there is any data in the telecommand FIFO. If the FIFO is empty then the handler exits.

If there is data in the FIFO then the handler checks to see if data is still being loaded into the FIFO by the CDMU. If it is then the handler exits. This is done to ensure that the handler does not read incomplete packets.

If there is a whole packet in the FIFO, i.e. the FIFO has data in it and is not being loaded then the packet is read from the FIFO. If the packet length field and the amount of data in the FIFO are inconsistent then an error is signalled and the handler exits discarding the corrupted packet. If a complete packet has been received the process packet ID is examined to see if it is a telecommand or a DDB. The packet count is checked against the process ID counter to check that the packet has not already been received. If the packet count is less than or equal to the process ID counter then an error is signalled and the packet is discarded. However, if the packet count is 0 then it is accepted and the process ID counter is reset.

If the packet count is valid then the process ID counter is loaded with the packet count.

In all there are 4 process ID counters for received data, these are:-

HK_Broadcast_CounterA	DDB Channel A Process ID Counter
HK_Broadcast_CounterB	DDB Channel A Process ID Counter
HK_Command_CounterA	Telecommand Channel A Process ID Counter
HK_Command_CounterB	Telecommand Channel B Process ID Counter

If the packet is a DDB then the mission time, altitude and phase are updated. The new mission time and altitude are used to see if SSP should change its mode. If a mode change is detected then the task handler is activated to change the task table.

If the packet is a command packet a reply header the command count is checked to ensure that the same command is not executed more than once. If the command count is OK then the command is executed via a vector table. The command ID is used as an offset into the command table CM_Command Table, the value read at this offset is the address of the code that implements the command. This code is jumped to.

Any output from the command is transmitted as telemetry via a sensor datastream in the case of sensor commands or the engineering data stream in the case of engineering commands.

If an error is detected then the command number and an error code is written to the housekeeping data area for transmission with the next housekeeping packet, if the incoming packet has been recognised as a command, as opposed to a DDB or not recognised at all, then an error packet is returned via the engineering datastream.

Error Codes are as follows:-

0	All OK
1	Packet already received
2	Packet length error
3	Packet checksum invalid
4	Unrecognised Command
5	Command already executed
6	Packet header invalid ID
7	Not able to execute this command in current mode
8	Command executed but failed.

4.1.2 Descent Data Broadcasts

4.1.2.1 Mission Time

The mission time is updated each time a DDB is processed, this should occur every 2 seconds. A 10 bit timer is synchronised with the DDB so that a 24 bit mission time with a resolution of 2ms is achieved. This time can be used to time stamp sensor sample times and packet transmission times.

The mission time is also used to switch the SSP mode as certain events occur. The default switching points are as follows, note that all times are assumed to be after time T0 and some mode changes will be instigated by external events:

0	Power On, Switch from cruise checkout to Upper Atmosphere Mode
10 minutes	Switch from Upper Atmosphere to Mid Atmosphere Mode
85 minutes	Switch from Mid Atmosphere to Lower Atmosphere Mode
120 minutes	Switch from Lower Atmosphere to Proximity Mode
151 minutes	Switch from Proximity Mode to Surface Mode
154 minutes	Switch from Impact to Extended Surface Mode

See Section 4.3 for a description of the DDB handling.

Timer Handler

Module	TI
Directory	TIMER

Files:	Macro	Timer.MAC
	Variables	Ti.VAR

The timer handler provides a selection of macros which enable the use and updating of the mission time. The mission time as received in the DDB is stored in the variable TI_MissionTime. The updating of and reading of this variable is available to all modules via the timer handler macros.

TI_Set Mission Time Rx	Updates TI_Mission Time with the contents of Rx which hold the mission time as received from the DDB.
TI_GetMission Time Rx	Loads the contents of TI_Misson Time into register Rx.
TI_Get2ms Timer Rx	Loads Rx with the value of the 2ms timer that is synchronised to the DDB.
TI_Reset2ms Timer	Resets the 2ms timer to 0, this enables the timer to be synchronised with the DDB.
TI_ReadMission Time Rx Rx+1	Reads the full 24 bit mission time into register Rx and the top 8 bits of register Rx+1. This is a combination of the TI_Mission Time and the 2ms timer. See Section 3.1.1. Used to time stamp sensor samples.

4.1.2.2 Altitude

The altitude of the spacecraft from the surface of Titan is transmitted to SSP via the DDB, SSP copies the altitude into the variable HK_Altitude. The altitude measured is used to switch SSP from Lower Atmosphere mode to Proximity mode 3km above the surface.

4.1.2.3 Mission Phase

The mission phase is one of the following:

Hex	Mission Phase
00	Entry/Descent
03	Ground Checkout
0C	Ground Checkout Suspended
0F	Ground Checkout Deactive
F3	Cruise Checkout
FC	Cruise Checkout Suspended
FF	Cruise Checkout Deactivated

SSP shall use the mission phase to:
Ensure that any experiment that may be damaged by operation in vacuum or 0g is operated in any phase in a manner such that damage will not be sustained. The mission phase is stored by SSP in the bottom 8 bits of the variable HK_Phase.

4.1.2.4 Spin

This is transmitted to SSP via the DDB and indicates the rate of spin of the spacecraft. This is not used by SSP but is stored in the top 8 bits of the variable HK_Phase.

4.1.2.5 CRC

SSP checks the CRC on all received DDBs and telecommands. An invalid CRC results in the DDB/telecommand being rejected. The same CRC algorithm is also used to check the integrity of EEPROM patches and the contents of the PROM and EEPROM.

4.1.3 SSP Command Types

Command ID (Code)	Function	Parameters	Eng. Packet?
0	Read RAM	Address, Length	Y
1	Write Ram	Address, Length, Data	
2	Read EEPROM	Address, Length	Y
3	Write EEPROM	Address, Length, Data	
4	Read PROM	Address, Length	Y
5	Verify RAM	Block (Low or High)	Y
6	Verify EEPROM	-	Y
7	Verify PROM	-	Y
8	Clear EEPROM	-	
10	Read DRB	DRB Address	Y
11	Write DRB	DRB Address, Data Word	
12	Execute DRB Sequence	Length, Commands	Y
12	Read I/O	I/O Address	Y
14	Write I/O	I/O Address, Data Word	
15	Execute I/O Sequence	Length, Commands	Y
16	Read Parallel Port	Port Number	Y
17	Write Parallel Port	Port, Data, Mask	
18	Write Telemetry	Channel, Length, Data Words	
19	Reset Data Stream	Channel Mask	
20	Read Processor Status	-	Y
21	Execute Code	Length, Code Words	
22	Send Housekeeping Pkt	-	
23	Read 12 bit A/D	Multiplexer Address, # Samples	Y
24	Read 16 bit A/D	Multiplexer Address, # Samples	Y
30	Stimulate ACC-E	Pulse Length	
31	Test ACC_E	Pulse Length	
32	Readout ACC-E	-	
33	Test ACC_I	Pulse Length	
34	Stimulate ACC-I	Pulse Length	
35	Readout ACC-I	-	
36	Readout API-S	Mode (for tele packet format)	
37	Sample API-V	# Samples	
38	Sample DEN	# Samples	
39	Sample PER	# Samples	
40	Readout REF	Illumination	
41	Readout THP	Wire Number / Mode	
42	Sample TIL		
50	Set SSP Mode	Mode Number, Lock Option	
51	Write Patch	Patch Number, Data	
52	Read Patch	Patch Number	Y
53	Delete Patch	Patch Number	

4.2 Use of Broadcast Pulse

A broadcast pulse is sent from the CDMU to SSP every 1/8th of a second, this pulse is made available to the processor as an interrupt. The SSP spacecraft interface counts 16 broadcast pulses and generated a 2 sec interrupt. The s/w can synchronise this pulse so that the processor is interrupted when a broadcast message is received. In practice these two interrupts are not enabled but the interrupt pending flags are used to indicate that a broadcast message or pulse has occurred. This enables the mission time to be kept up to date.

The 2 sec interrupt is also used to reset an onboard 10 bit counter. This gives the SSP time a resolution of 2ms.

During initialisation a watchdog interrupt, the BCP 1/8th sec interrupt and BCP 2 sec interrupt are started.

Then the BCP is synchronised to the DDB such that the 2 sec interrupt occurs on the BCP pulse after the DB has been received. If no DDBs occur in 8 consecutive DDB time periods then an error flag is set in the housekeeping indicating that there is no DDB/BCP synchronisation.

Once SSP is running, the three interrupts service the BCP as follows:

The 2 sec BCP interrupt checks to see that a DDB has been received, if it has then the interrupt updates the mission time from the DDB. If it has not then the mission time is incremented by 2 seconds. After DDB failures then the DDB is ignored, a DDB failure is flagged in the housekeeping and the mission time updated automatically by 2 seconds on each 2 second interrupt.

The 1/8th sec BCP interrupt triggers a watchdog timer interrupt. The watchdog timer interrupt checks to see if a 2 sec interrupt should have occurred. If it has not then the Watchdog timer simulates a 2 sec BCP interrupt.

The timer is set up to timeout after 350ms (350ms = 2 BCP times + 100ms). If the timer times out then the 1/8th sec BCP interrupt has probably failed. the timer flags a 1/8th sec BCP failure in the housekeeping and reprograms itself to timeout every 125ms (1/8th sec) so taking over from the 1/8th sec interrupt.

4.3 Telemetry

Telemetry packets are transmitted via a telemetry FIFO to the CDMS. There is a separate FIFO for each of the two telemetry channels. The processor can determine if the last telemetry packet has been transmitted by reading the FIFO empty flag. When the processor recognises that the last packet has been transmitted it can load the FIFO with a new packet. The processor can reset the FIFOS clearing them of any data. The FIFOS will be reset on power up.

4.3.1 Telemetry Handler

Module	TM	
Directory	TELEM	
Files:	Code	Telem.ASM
	Variables	Tm.VAR
	Constants	Tm.CON
	Make	Telem.MAK

The telemetry handler is responsible for taking data from the data streams inserting this data into a packet and then loading the packet into the two telemetry FIFOS.

The telemetry handler is executed by the main background program loop, so it can be interrupted by the task scheduler if a sensor needs servicing.

When the handler starts executing it first checks to see if either of the telemetry FIFOS are empty. If they are both full then the handler exits.

If one or both FIFOS are empty then the handler checks the impact data stream, if there is enough data in the datastream to fill a packet then an impact packet is constructed and loaded into both telemetry FIFOS. A FIFO is not loaded if there is not enough data in the impact data stream to fill a packet then the other data streams are polled in turn. If any data stream has enough data to fill a packet then the data is inserted into a packet and the packet is loaded into one or both FIFOS. If the engineering data stream is polled then if it has any data in it then this data is loaded into a packet. The empty packet space is padded out with (AAAAhex). The packet is then loaded to one or both telemetry FIFOS.

The handler is designed so that if one telemetry channel fails then it shall transmit all the telemetry on the other channel. In normal operation the telemetry shall be transmitted on both channels.

5. SENSOR INTERFACE HANDLING

5.1 ACC-I

The ACC-I sensor is an accelerometer that measures the acceleration of the spacecraft. This sensor is used as the main impact detector.

5.1.1 ACC-I Module

Module	ACC-I	
Directory	ACC-I	
Files:	Code	ACC-I.ASM
	Variables	ACC-VAR
	Constants	ACC-I.CON
	Make	ACC-I.MAK

5.1.2 Sampling

Functions: ACC_I_Sample
ACC_I_ 500 Hz Sample

The sample function selects the ACC-I sensor input and reads the output of the sensor. The reading is returned in register RO.

The 500 Hz sampling function takes a reading of the ACC-I sensor using ACC_I_ Sample, the reading has the ACC-I offset subtracted from it and the resulting value is stored in a circular buffer. This function is used by the ACC-I command functions and by the ACC-I proximity task. The ACC-I offset is subtracted from the ACC-I reading so that the impact detection mechanism does not trigger on the offset signal.

5.1.3 Checkout

Functions: ACC_I_ Checkout Task

ACC_I_ Checkout Process
ACC_I_ Process Impact Data

The ACC-I checkout task is run when SSP enters Mode 1, this task applies a stimulation signal to the sensor electronics and reads the sensor at 500 Hz for 512 samples. when the task terminates the ACC-I ready flag is set to indicate that there is data to be processed. The background ACC-I process detects the ACC-I ready flag and processes the ACC-I data buffer. The peak signal position and amplitude is found as well as the FWHM positions, this data is saved in the checkout packet. The rest of the checkout packet is filled with data from the buffer, 4 samples are taken from the beginning and end of the buffer and the region around the peak is also copied to the checkout packet. As the data is 12 bit data it is compressed such that 4 samples are compressed into three 16 bit words. The checkout packet fits exactly into a Huygens packet so that all the data is transmitted in a single packet.

5.1.4 Atmospheric Modes

TBD

Function: ACC_I_ Sample
ACC_I_ Task
ACC_I_ Atmosphere Process

The ACC-I sensor is sampled at 500 Hz for 0.4 seconds. Once finished the ACC-I ready flag is set. The background process detects the ready flag and processes the data. The maximum, mean and variance of the 200 samples are found. These values are loaded to the ACC-I data stream. The mean value is added to a running total of mean values and an ACC-I offset is calculated from this which is saved in the housekeeping data.
This measurement is repeated every second (i.e. the data contains the max, mean and variance of the first 0.4 seconds of every second).

5.1.4.1 Task Timeline

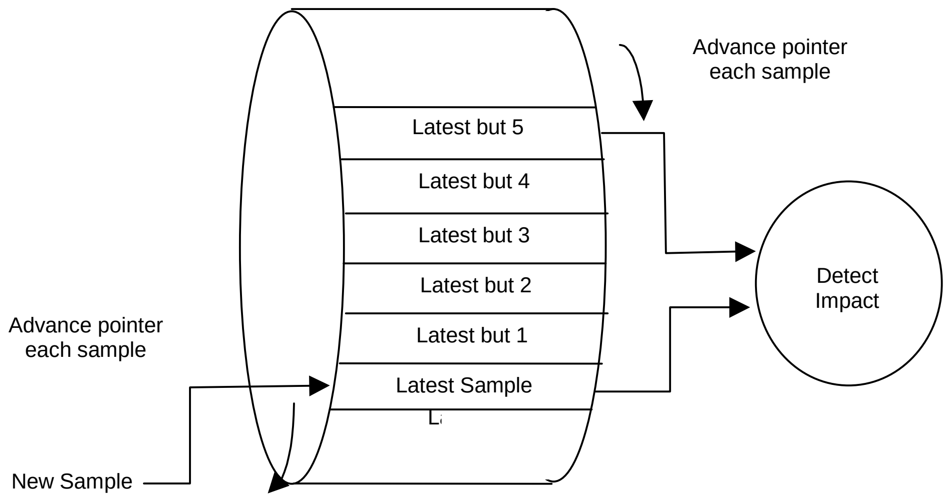
Function: ACC_I_ Task

Time (ms)	Action
0	Determine Sample Time
2	Read 1st Sample
4	Read 2nd Sample
6	Read 3rd Sample
:	: :
308	Read 199th Sample
400	Read 200th Sample and Set ACC-I Ready Flag

5.1.5 Proximity Mode

Function: ACC_I_ 500 Hz Sample
ACC_I_ Proximity Task
ACC_I_ Look for Impact
ACC_I_ Process Impact
ACC_I_ Process Impact Data

A sample of the ACC-I sensor is taken every 2ms (500 Hz) using the 12 bit ADC and is stored in a circular buffer (512 word FIFO). After each acquisition, the latest sample is compared to the detection threshold and, if greater, the Acc_I_TotalSignal count is incremented. The latest value but 5 is also compared to the threshold and, if greater, the Acc_I_TotalSignal count is decremented. Thus Acc_I_Total signal contains a count of the number of samples exceeding the threshold in the last 5.



The threshold for impact detection is initialised in the code to 80 TM counts which corresponds to approximately 3.5 g (2048 counts = approx 90g)

The number of samples above threshold to indicate impact, Acc_I_ExceedLimit, is initialised in the code to 3. An impact is assumed if Acc_I_Total reaches 3.

After impact is detected the sensor continues to be read at 500 Hz for a further 448 samples.

5.1.5.1 Task Timeline

Task Name: ACC_I_ Proximity Task

Time (ms)	**Action**
0	Read Sample Calculate Total Signal
2	Read Sample Calculate Total Signal
4	Read Sample Calculate Total Signal
:	: : :
Impact-2	Read Sample Calculate Total Signal
Impact	Freeze ACC-I
Impact + 2	Read Sample
Impact + 4	Read Sample
:	: :
Impact + 894	Read Sample
Impact + 896	Set Impact Detected Flag, Terminate Task

5.1.6 Surface Modes

See Atmospheric Modes Section 5.1.4.

5.2 DEN

The Den sensor measures the density of the liquid in which the Den sensor is immersed. The sensor is operational throughout the mission at 1 Hz. The atmospheric measurements allowing offset drift and temperature changes to be monitored (this may also provide secondary information on turbulence and probe accelerations).

5.2.1 Den Module

Module	Den	
Directory	Den	
Files:	Code	Den.ASM
	Variables	Den.VAR
	Constants	Den.CON
	Make	Den.MAK

5.2.2 Sampling

Function: Den_Sample

The Den sensor is read simply by selecting the multiplexer address for Den and reading the output voltage of the sensor using the 12 bit ADC. The sensor is read 8 times and the average of the 8 reading is calculated.

5.2.3 Processing

Function: Den_Process Sample
Den_Task

Firstly a Den sample is taken with Den disabled; this reading is saved in the housekeeping packet data area and is the Den offset. Den is then sampled 4 times with Den being enabled for each reading. These 4 samples are compressed into 3 16 bit words and loaded to the Den datastream for transmission.

5.2.4 Task Timeline

Function: Den_Task

Time (ms)	Action
0	Determine Sample Time, Sample Den Offset and then Enable Den
4	Sample Den, Disable Den
1000	Enable Den
1004	Sample Den, Disable Den
2000	Enable Den
2004	Sample Den, Disable Den
3000	Enable Den
3004	Sample Den, Disable Den
3006	Process Sample
3008	Transmit Sample

5.3 PER and CON

The PER sensor measures the permittivity of the fluid in which the sensor is immersed. The sensor is operational throughout the mission.

The conductivity of the fluid can be measured by the PER sensor when it is not clocking, this is known as CON.

5.3.1 Per Module

Module	PER	
Directory	PER	
Files:	Code	PER.ASM
	Variables	PER.VAR
	Constants	PER.CON
	Make	PER.MAK

5.3.2 Sampling

Functions: PER_Sample
CON_Sample

The PER sensor is read by reading the sensor via the 12 bit ADC, CON has to be disabled and PER has to be clocked for 1 second before taking a reading.

The CON sensor is read by reading the PER input via the 12 bit ADC, CON has to be enabled for 1 second before taking the reading and PER clocking has to be disabled during the ADC conversion.

5.3.3 Processing

Functions: PER_Process
PER_Task

When the PER task starts at each mode change PER and CON are read with CON disabled and PER not clocking, these two readings are the offset measurements for PER and CON and are saved in the housekeeping data area.

CON is then enabled and sampled followed by PER being enabled (CON disabled) and sampled. CON and PER are sampled again and then the 4 samples are compressed into 3 16 bit words before being transmitted. This process is repeated such that PER and CON are read once every 10 seconds.

5.3.4 Task Timeline

Function: PER_Task

Time (ms)	Action
0	Disable CON, Disable PER Clocking
1000	Sample PER and CON offsets, Enable CON
10000	Sample CON, Enable PER
11000	Sample PER, Enable CON
20000	Sample CON, Enable PER
21000	Sample PER, Enable CON, Process Samples, Transmit Data
30000	Sample CON, Enable PER
31000	Sample PER, Enable CON
40000	Sample CON, Enable PER
41000	Sample PER, Enable CON, Process Samples, Transmit Data
:	: : : : :

5.4 REF

The REF sensor is a light sensitive array; this array has 512 pixels and is illuminated by light refracted by a prism. The light source is provided by two LEDs. A light/dark boundary is seen in the refracted light. The position of this is determined by the refractive index of the liquid in which the prism is immersed (N.B. The REF sensor can only determine refractive index, R.I. if immersed in a liquid, or if a thin film of liquid condenses on the REF prism – in which case only the internal mode can detect R.I.).

5.4.1 REF Module

Module REF

Directory REF

Files:	Code	REF.ASM
	Variables	REF.VAR
	Constants	REF.CON
	Make	REF.MAK

5.4.2 Checkout

None.

5.4.3 Upper Atmosphere

The REF sensor does not operate during upper atmosphere mode.

5.4.4 Mid and Lower Atmospheric Modes

5.4.4.1 Sampling

Functions: REF_Reset
REF_Clear
REF_Illuminate
REF_Readout

For each full sample the REF sensor is read out 3 times using different illumination modes.

The three modes are:

i) Dark - No Illumination.

ii) External - Prism Illuminated by the External LED.

iii) Internal- - Prism Illuminated by the Internal LED.

The sensor is read out as follows:

i) Sensor readout register is initialised by clocking the start and pixel clocks.

ii) Sensor array is cleared by clocking each pixel out but not digitising it.

iii) The readout time is taken, the prism is illuminated as required.

iv) Array is exposed to light for a pre-calculated exposure time.

v) Illumination is switched off.

vi) Sensor readout register is initialised by clocking the start and pixel clocks.

vii) Array is readout by clocking the pixel clock and digitising each pixel with the 12 bit ADC.

5.4.4.2 Processing

The background task detects the REF data ready flag which is set when the REF task has completed readout and processes the REF data. Each 12 bit pixel value is compressed to 8 bits by shifting each value left by 3 bits (dividing by 8) and storing the LSB which loses the most significant bit. The 512 pixels are therefore compressed into 512 bytes which are loaded to the REF datastream for transmission.

5.4.4.3 Task Timeline

Function: REF_Task

Time (ms)	Action
0	Reset Sensor
1	Clear 6 Pixels
2	Clear 6 Pixels
:	: : :
85	Clear Last 6 Pixels
86	Reset sensor, internally illuminate prism for pre-calculated internal exposure time, read sample
	time, expose
Exposure	Turn off illumination
Exposure +1	Read 6 Pixels
Exposure + 2	Read 6 Pixels
:	: : :
Exposure + 85	Read Last 8 Pixels
Exposure + 86	Set REF Data Ready Flag

The above process is repeated at 16000 ms with no illumination, and at pre-calculated external exposure time (this equals 50 x internal exposure time) with external illumination.

5.4.4.4 Pre-Calculated Exposure Time

The REF module has a table of 330 points giving the required internal mode exposure time for a measured sensor temperature in 1 K steps from 0 K to 329 K (although, as initialised, the exposure only changes at 10K intervals). The measured temperature used is nominally taken from the REF_SENSE_DIODE. If REF_SENSE_DIODE were to fail then the software would default to the other REF temperature diodes (first REF_PRBASE and then REF_PRTIP).

The following equation is used by the software to derive diode temperature (in K) from the diode sensor output (in bits). (N.B. This may differ very slightly to the preferred conversion formula for the REF SENSE DIODE, but was the best available when software was written. In fact the difference is less than 4 K at around 100 K temperature).

REF_SENSE_DIODE_TEMP (K) = 433.085 - 0.204*REF_SENSE Output (bits)

The table is given below:

```
;	Exposure times in ms for internal illumination mode for temperatures 0K - 329K
;	External mode illumination time  = Internal mode illumination time * 50

	Ref_Internal Mode Exposure Times in ms:			; Temperature in K
DW	0020,0020,0020,0020,0020,0020,0020,0020,0020,0020	; 0K - 9K
DW	0020,0020,0020,0020,0020,0020,0020,0020,0020,0020	; 10K - 19K
DW	0020,0020,0020,0020,0020,0020,0020,0020,0020,0020	; 20K - 29K
DW	0020,0020,0020,0020,0020,0020,0020,0020,0020,0020	; 30K - 39K
DW	0020,0020,0020,0020,0020,0020,0020,0020,0020,0020	; 40K - 49K
DW	0020,0020,0020,0020,0020,0020,0020,0020,0020,0020	; 50K - 59K
DW	0020,0020,0020,0020,0020,0020,0020,0020,0020,0020	; 60K - 69K
DW	0020,0020,0020,0020,0020,0020,0020,0020,0020,0020	; 70K - 79K
DW	0020,0020,0020,0020,0020,0020,0020,0020,0020,0020	; 80K - 89K
DW	0020,0020,0020,0020,0020,0020,0020,0020,0020,0020	; 90K - 99K
DW	0020,0020,0020,0020,0020,0020,0020,0020,0020,0020	; 100K - 109K
DW	0020,0020,0020,0020,0020,0020,0020,0020,0020,0020	; 110K - 119K
DW	0022,0022,0022,0022,0022,0022,0022,0022,0022,0022	; 120K
DW	0022,0022,0022,0022,0022,0022,0022,0022,0022,0022	; 130K
DW	0025,0025,0025,0025,0025,0025,0025,0025,0025,0025	; 140K
```

```
DW      0030,0030,0030,0030,0030,0030,0030,0030,0030,0030      ; 150K
DW      0040,0040,0040,0040,0040,0040,0040,0040,0040,0040      ; 160K
DW      0040,0040,0040,0040,0040,0040,0040,0040,0040,0040      ; 170K
DW      0050,0050,0050,0050,0050,0050,0050,0050,0050,0050      ; 180K
DW      0050,0050,0050,0050,0050,0050,0050,0050,0050,0050      ; 190K
DW      0060,0060,0060,0060,0060,0060,0060,0060,0060,0060      ; 200K
DW      0077,0077,0077,0077,0077,0077,0077,0077,0077,0077      ; 210K
DW      0087,0087,0087,0087,0087,0087,0087,0087,0087,0087      ; 220K
DW      0095,0095,0095,0095,0095,0095,0095,0095,0095,0095      ; 230K
DW      0115,0115,0115,0115,0115,0115,0115,0115,0115,0115      ; 240K
DW      0120,0120,0120,0120,0120,0120,0120,0120,0120,0120      ; 250K
DW      0140,0140,0140,0140,0140,0140,0140,0140,0140,0140      ; 260K
DW      0160,0160,0160,0160,0160,0160,0160,0160,0160,0160      ; 270K
DW      0180,0180,0180,0180,0180,0180,0180,0180,0180,0180      ; 280K
DW      0230,0230,0230,0230,0230,0230,0230,0230,0230,0230      ; 290K - 299K
DW      0250,0250,0250,0250,0250,0250,0250,0250,0250,0250      ; 300K - 309K
DW      0275,0275,0275,0275,0275,0275,0275,0275,0275,0275      ; 310K - 319K
DW      0300,0300,0300,0300,0300,0300,0300,0300,0300,0300      ; 320K - 329K
```

5.4.5 Proximity Mode

As for atmospheric modes except that only 1 pixel is read per timeslice.

5.4.6 Impact

REF is read out in all three modes and the data transmitted via the REF datastream. The data is used to calculate the refractive index of the liquid and this is inserted into the impact packet along with samples from the boundary points.

The internal refractive index is calculated by subtracting the dark readout data from the internal readout data. The resulting values are then smoothed by averaging each value over 8 values. The dark/light boundary can be determined by finding the maximum positive gradient of the smoothed data. The external refractive index is found in a similar way except that the light/dark boundary is determined by finding the maximum negative gradient.

Note that due to the REF linearity range (first 400 pixels only) only the first 400 pixels are processed.

5.4.7 Surface Modes

As for Atmospheric Modes.

5.5 TIL

The TIL sensor measures the tilt of the spacecraft in two orthogonal directions relative to the local gravity vector. The sensor is operational throughout the mission.

5.5.1 TIL Module

Module	TIL
Directory	TIL

Files:	Code	TIL.ASM
	Variables	TIL.VAR
	Constants	TIL.CON
	Make	TIL.MAK

5.5.2 Sampling

Function: TIL_Sample

The TIL hardware is clocked by a 500 Hz square wave. There are 4 measurements that can be taken, these are Tilt X, Tilt Y, Top x and Top Y. The Tilt X and Y value indicate the tilt of the spacecraft in X and Y and the Top measurements indicate the driving voltage to each direction. Each value is measured twice, once during the +ve clock phase and again 1 ms later during the -ve phase. To read the sensor the software has to perform at least one read during each phase of the cycle. To do this the sampling routine reads the output from the sensor twice, if the two value differ by more than 8 then the samples were taken whilst the clock was changing phase so a third sample is taken if both samples differ by less than 8 the first sample is used. If the valid reading is positive then it is saved in the positive location for the sample otherwise the negative location is used.

5.5.3 Checkout

Functions: TIL_Disable
TIL_Sample Offsets

The electronic offset of the sensor sub-system is measured by disabling the TIL input and reading TIL as if it were operational. The readings taken are saved in the housekeeping data. These offsets are read at the start of each SSP mode change.

5.5.4 Processing

Function: TIL_Process

The eight 12 bit reading are compressed into six 16 bit words before transmission.

5.5.5 Operation - All Modes

Function: TIL_Task

The TIL task starts by disabling the TIL sensor and measuring the offsets. Once this has been done TIL is re-enabled and the main task loop begins.

The task loop starts by measuring the TIL X and TIL Y values. This process is then repeated 1 ms later for the next phase of the TIL clock. The Top X and Top Y values are measured similarly and finally the values are compressed and loaded to the TIL datastream for transmission.

The task loop is repeated at intervals of 1 second.

5.5.6 Task Timeline

Function: TIL_Task

Time (ms)	Action
0	Disable TIL
1000	Sample TIL Offsets and Re-Enable TIL
2000	Sample TIL X and Y
2001	Sample TIL X and Y
2002	Sample Top X and Y
2003	Sample Top X and Y
2005	Compress Data and Transmit It
3000	Sample TIL X and Y
:	: : :

5.6 THP

The THP sensor is designed to measure the thermal conductivity of the fluid the sensor is immersed in. The measurement is obtained by heating one of the 4 wires and measuring the wire's resistance change over time.

5.6.1 THP Module

Module	THP	
Directory	THP	
Files:	Code	THP.ASM
	Variables	THP.VAR
	Constants	THP.CON
	Make	THP.MAK

5.6.2 Sampling

Functions: THP_Sample,THP_Startup, THP_SelectWire, THP_Sampling

The THP sensor is sampled as follows:

i) Wire 1, 2 3 or 4 is selected, if the selected wire is broken then an alternative wire is chosen.

ii) A measurement is taken with no current, this is the THP offset measurement.

iii) Low current is applied to the wire.

iv) The initial resistance of the wire is measured.

v) The initial measurement has a gain factor applied and an offset added and is loaded into the offset DAC.

vi) High current is applied to the wire.

vii) 60 readings are taken over a time period, the readings are spaced logarithmically and the time period depends on the SSP mode.

5.6.3 Processing

Functions: THP_Sampling

Each reading is the mean of 16 measurements of the sensor. If the mean exceeds a value of $7FF0^{16}$ then the high current is switched off and the rest of the data sample is filled with the last reading.

5.6.4 Safety Mechanisms

Functions: RB_16_High Current

High Current will NOT be applied to any THP wire if:

i) The DDB indicates that we are in cruise checkout.

ii) The DDB indicates that we are in descent phase AND the mission time is <T0 + 10 minutes.

5.6.5 Checkout

Function: THP_Checkout Task

For the first 'THP' sample in Mode 1 a check of the A/D calibration is performed.

The checkout consists of loading a series of 31 increasing values to the DAC; both the 12 and 16 bit ADCs are used to measure output. The results are stored in a standard THP sample with the first 31 samples being the 16 bit results and the second 31 values the 12 bit results. The results can be used to determine if both ADCs and the DAC are operating correctly.

The 31 values are (in hex):

0000	000C	0018	0024	0030	003C	0048	0054	0060	006C
0078	0084	0090	009C	00A8	00B4	00C0	00CC	00D8	00E4
00F0	00FC	0108	0114	0120	012C	0138	0144	0160	0168
0174									

5.6.6 Upper Atmosphere Mode (1)

Function: THP_Upper Task

Wires 3 and 4 are sampled for a period of 60 seconds.

Note that in Cruise checkout and Descent low current only will be used. In ground checkout high current will be used. See 5.2.1.4 above.

5.6.7 Mid Atmosphere Mode (2)

Function: THP_Mid Task

A wire is sampled every 60 seconds. Wires 3 and 4 are each sampled for 15 seconds followed by 105 seconds off.

5.6.8 Low Atmosphere Mode (3)

Function: THP_Lower Task

A wire is sampled every 60 seconds. Wires 3 and 4 are each sampled for a period of 15 seconds followed by 105 seconds off. If both wires are broken then wires 1 and 2 are used.

5.6.9 Proximity Mode (4)

Function: THP_Proximity Task

A wire is sampled every 60 seconds. Wires 3 and 4 are each sampled for a period of 15 seconds followed by 105 seconds off. If both wires are broken then wires 1 and 2 are used.

5.6.10 Surface Mode (5)

Function: THP_Surface Task

A wire is sampled every 15 seconds. All 4 wires are each sampled for a period of 5 seconds followed by 55 seconds off. Wires 1 and 2 are sampled alternately with wires 3 and 4 i.e. the sampling sequence is 1, 3, 2, 4...(or 2, 3, 1 ,4 depending on which wire was sampled first at the start of the mode).

5.6.11 Extended Surface Mode (6)

Function: THP_Extended Task

Wires 1 and 2 are sampled for 15 seconds and wires 3 and 4 sampled for 25 seconds. Again wires 1 and 2 are sampled alternately with wires 3 and 4 i.e. the sampling sequence is 1, 3, 2, 4...(or 2, 3, 1, 4 depending on which wire was sampled first at the start of the mode). So wire 1 or 2 will be on for 15 seconds, all off for 30 seconds, 3 or 4 on for 25 seconds, all off for 30 seconds.

5.6.12 Sample Timings

These tables define the time gap in ms between each of the 60 samples.

5 Second Sample

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	3	4	4	6	6	7	9	9
11	13	15	18	20	23	27	31	37	42
48	56	66	75	87	102	116	136	158	182
210	244	284	328	378	440	510	590	682	

15 Second Sample

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	4
4	5	6	7	8	10	11	13	16	19
22	26	31	36	43	51	59	71	83	98
116	136	162	190	226	266	312	370	438	514
608	718	848	1000	1182	1392	1646	1942	2290	

25 Second Sample

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	2	4	4
5	7	7	9	11	12	15	18	21	26
30	36	42	51	61	72	86	102	122	146
172	206	246	290	348	414	494	586	698	832
992	1180	1404	1674	1992	2372	2824	3364	4004	

60 Second Sample

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	2	3	5	5	6
8	9	11	14	16	20	24	29	35	42
52	61	75	91	108	132	160	194	232	282
342	412	498	602	730	880	1064	1286	1556	1880
2274	2746	3322	4016	4854	5868	7096	8574	10368	

5.7 ACC-E

ACC-E is sampled at 10 kHz by hardware. The ACC-E analogue data is converted to an 8 bit value using a hardware pseudo logarithmic conversion technique. The converted data is then loaded into a 512 byte FIFO. When the hardware detects an impact (Set in hardware as a 177 mV threshold at the output of the pseudo logarithmic amplifier. This equates to a signal of 80 mV at the ACC-E output, equivalent to an input force of 3.9 N at room temperature or 7.2 N at liquid nitrogen temperature) the data is frozen in the FIFO so that approximately 64 samples have been taken before the impact and the rest have been taken after impact. The hardware can indicate to the processor that impact has occurred and the processor can then read the contents of the FIFO. The processor can restart the ACC-E hardware once an impact has been detected.

5.8 API-V

The API-V sensor measures the local velocity of sound.

5.8.1 API-V Module

Module	API-V	
Directory	API	
Files:	Code	API-V.ASM
	Variables	API.VAR
	Constants	API.COn
	Make	API.MAK

5.8.2 Checkout

None.

5.8.3 Operation - All Modes

Sampling Rate: 1 Hz
Data Rate: 1.67 bytes/second

5.8.4 Sampling

Functions: API_V_Trigger
API_V_Read Sample

The API-V sensor is triggered by reading the API-V trigger register on the API sensor electronic board. Two velocity readings are taken in opposite directions. These results are available 10 ms after triggering and can be read as two 12 bit numbers from the data registers on the sensor electronics board

5.8.5 Processing

Function: API_V_Compress Sample

Two samples of two 12 bit measurements are compressed into three 16 bit words.

5.8.6 Task Time

Function: API_V_Task

Time (ms)	Action
0	Determine Sample Time and Trigger First API-V Sample
10	Read 1st Sample and Trigger Second API-V Sample
20	Read 2nd Sample
22	Process Samples
24	Transmit Samples

5.9 API-S

The API-S sensor is an acoustic sounder. It emits acoustic signals and receives any reflections of the signal from solid or semi-solid objects.
The API-S sensor is triggered by writing to the API-S trigger register on the API sensor electronics board. The sensor emits a signal and waits for the return signal. The return signal is available to be read 1 second after triggering. The return signal is stored in a 1000 byte buffer on the sensor electronic board and can be read by reading the API-S data register.

5.9.1 API-S Module

Module API-S
Directory API-S

Files:		
	Code	API-S.ASM
	Variables	API.VAR
	Constants	API.CON
	Make	API.MAK

5.9.2 Checkout

None.

5.9.3 Upper Atmosphere Mode

There is no operation of the API-S sensor during upper atmosphere mode.

5.9.4 Mid and Lower Atmospheric Modes

Sampling Rate: 1 Sample every 2 seconds (minimum).

5.9.4.1 Sampling

Functions: API_S_Trigger
API_S_Read Sample

The sensor is triggered every two seconds during mid and lower atmosphere modes. One second after triggering API-S the API-S ready flag is set. This indicates to the background process that the data can be processed. Once the data has been processed the flag is cleared. The background process as well as processing the data can judge if there is any spare data rate available for API-S and speed up or slow down the sampling cycle as required. When the ready flag is cleared the cycle time is updated and a new sample taken at the start of the next sample cycle.

5.9.4.2 Processing

Function: API_S_Atmosphere Compress

Only bytes 50 to 570 are used of the total 1000 byte sample. These 520 bytes are compressed into 26 bytes by taking the mean of each of 26 blocks of 20 bytes.

For Modes 2 and 3 ONLY the bins packing order in telemetry packets is swapped pair-wise as shown by the figure below (i.e. "Little Endian" storage format).
(N.B. Titan PDS data are corrected but any older, non PDS complaint data sets may need correcting).

Actual order of bins (where each bin is average of 20 samples)

1	2	3	4	5	6

Order of bins in telemetry packets

2	1	4	3	6	5

5.9.4.3 Task Timeline

Function: API_S_Task

Time (ms)	**Action**
0	Get Sample Time and Trigger API-S Sample
1000	Set API-S Ready Flag
1100	See if API-S Ready flags Clear - Repeat every 100 ms until Clear Modify API-S Cycle Time.

Note that a time slice slot is reserved for API every 100 ms, this enables the cycle time to be varied by multiples of 100 ms.

5.9.5 Proximity Mode

Sampling Rate: 1 Sample every 2 seconds (Minimum)

5.9.5.1 Sampling

Function: API_S_Pulse Length
API_S_Trigger
API_S_Read Sample

As for atmospheric mode except the pulse width may be set to short (2 ms) at an altitude below 1 km (provided that the "measured altitude" flag is set in the DDB. See section 2.4.5.) by setting the pulse width signal to the API-S sensor electronics board.

5.9.5.2 Processing

Function: API_S_Proximity Compress

The peak signal (see section 3.2.5.4) is found in the return signal then the data is compressed as follows :

Peak Signal ± 30 Values	-	Not compressed (61 bytes)
Peak Signal ±100 Values	-	Compress into bins of 4 samples/bin (35 bytes)
Rest	-	Compress into bins of 20 samples/bin (42 bytes)

5.9.5.3 Task Timeline

As for Mid and Lower Atmosphere.

5.9.6 Surface Mode

As for proximity mode. Data for first sample in this mode is further analysed on the data stored in the impact packet.

5.9.7 Extended Surface Mode

Sampling Rate: 10 Samples at one second intervals every 30 seconds
Data Rate: 26.17 bytes/sec.

5.9.7.1 Sampling

As for atmospheric mode except cycle time is nominally 10 seconds followed by a 20 second gap.

5.9.7.2 Processing

Function: API_S_Extended Process

The sample which contains the return signal is processed as for proximity mode, all 10 samples are compressed into 50 bins of 20 samples.

5.9.7.3 Task Time

Function: API_S_Surface Task

Time (ms)	Action
0	Get sample time and trigger 1st API-S sample
1000	Set API-S ready flag
1100	If ready flag clear trigger 2nd sample (100 ms repeat if flag not clear)
2100	Set API-S ready flag
2200	If ready flag clear trigger 3rd sample (100 ms repeat if flag not clear)
3200	Set API-S ready flag
3300	If ready flag clear trigger 4th sample (100 ms repeat if flag not clear)
4300	Set API-S ready flag
4400	If ready flag clear trigger 5th sample (100 ms repeat if flag not clear)
5400	Set API-S ready flag
5500	If ready flag clear trigger 6th sample (100 ms repeat if flag not clear)
6500	Set API-S ready flag
6600	If ready flag clear trigger 7th sample (100 ms repeat if flag not clear)
7600	Set API-S ready flag
7700	If ready flag clear trigger 8th sample (100 ms repeat if flag not clear)
8700	Set API-S ready flag
8800	If ready flag clear trigger 9th sample (100 ms repeat if flag not clear)
9800	Set API-S ready flag
9900	If ready flag clear trigger 10th sample (100 ms repeat if flag not clear)
10900	Set API-S ready flag
11000	If ready clear wait for 60 seconds before restarting sampling

Count reset to starting a new sample. This means that at a mode change there will be some incomplete samples so the EGSE must be aware of this. In the case of the continuous data streams the data samples may lose synchronisation with the packets.

6. ROUTING BUS

6.1 Overview

The routing bus provides a redundant control/data path between the LPP and the various peripherals. The data routing bus is entirely separate to the main processor bus so that any problems on the data routing bus do not affect the processor. The two buses are nominated the primary bus and the redundant bus. The SST experiment splits the experiment electronics between the two buses as follows:

Primary Routing Bus: Spacecraft Interface A
12 Bit Analogue Board

Redundant Routing Bus: Spacecraft Interface B, including EEPROM
16 Bit Analogue Board
API Board

6.2 Routing Bus Handlers

Module ROUTING

6.2.1 Address Map

Filename: RB_Mem.MAP

The Routing Bus is mapped into the processor's memory space at address $FC00_{16}$ the sub-system addresses are as follows:

Spacecraft I/F	FC70
LPP	FC78
16 Bit Board	FCB0
12 Bit Board	FCD0
API Board	FCE0

Note that the A or B spacecraft interface is selected by using the primary or redundant routing bus.

6.2.2 Low Level Handler

Filenames:	Code	RB_Utils.ASM
	Macro	RB_Utils.MAC
	Make	RB.MAK

The low level handler provides a set of functions and macros to enable other handlers to use the bus.

Function	Description
RB_Initialise	Initialises the data routing bus by mapping it into memory, the 12 bit board is then reset. This function has to be called before any access to the routing bus can be made.

Macro	Parameters	Description
RB_Write	Rx.Address	Writes the data held in register Rx to routing bus address. Note that the routing bus base address is automatically added to Address before writing the data. No check is made to see if the data was received OK.
RB_Write	Rx.Address, Ry	Same as above except the address written to is Address + Contents of Ry.
RB_Read	Rx.Address	Reads a value from address of the routing bus and returns the value read in Rx. Note that no check is made to see if data is really present.
RB_Read	Rx.Address, Ry	Same as above except the address read from is the sum of Address + Contents of Ry.
RB_Set Active Bus	Bus	Activates the Primary or Redundant bus. Bus should be RB_Primary Bus or RB_Redundant Bus.

6.2.3 12 Bit Board Handler

Module RB_12

Filenames:	Code	RB_Adc.ASM

Macro	RB_Adc.MAC
Variable	RB_Adc.VAR
Make	RB.MAK

The 12 bit board handler provides a set of functions and macros to enable the 12 bit sensor handlers to control and sample 12 bit sensors.

Function	Description
RB_12_SetMuxAddr	This sets the multiplexer address to the address held in register R1, this enables the ADC to sample data from more than one device. The mux address must be set up before reading data from the ADC. See RB_Mem.MAP for valid multiplexer addresses.
RB_12_ReadAdc	This function uses the 12 bit ADC to convert the signal at the current multiplexer address. The value of the signal sampled is returned in register RO.
RB_12_FastReadAdc	As the ADC requires up to 30µs to perform a sample this function enables processing to continue whilst the ADC is performing a conversion. The first call will start the ADC conversion process and the second call will return the data sampled by the ADC and also start a second conversion. A third call will return the data from the second conversion and start a third. There must be 30 µs between each call to this function to give the ADC time to convert.
RB_12_Reset	This function resets the 12 bit board and sets the multiplexers and control signals to a known state.

Macro	Parameters	Description
RB_12_Raise	Signal	Raises the signal control line signal. See RB_Mem.MAP for signal definitions. Note that the signal is not actually raised until RB_12_apply Signals is called.
RB_12_Lower	Signal	As above except signal is lowered.
RB_12_Toggle	Signal	As above except signal state is inverted.
RB_12_Set Signals Rx		Sets the state of all the signals to the contents of Rx where each bit in Rx refers to a specific signal. Note RB_12_ApplySignals should be used afterwards to set the physical control signals.
RB_12_GetSignalsRx		Returns the current state of the control signals in Rx. This will reflect the state of the physical control signals only if RB_12_ApplySignals was last used.
RB_12_Apply Signals Rx		Applies any control signal changes using the above macros to the physical control signals. The current state of the control signals is returned in Rx.

6.2.4 Spacecraft Interface Handler

Module	RB_SC

Filenames:	Code	RB_SC.ASM
	Macro	RB_SC.MAC
	Make	RB.MAK

The spacecraft interface handler provides a set of functions and macros to enable the command and telemetry sensor handlers to manage control and data to and from the main spacecraft.

Function	**Description**
RB_SC_Read/ML.MLFifo	Reads the command FIFO, the number of words to read is passed in R0 and the address to write the data in R2. On exit R0 holds the number of data words not read by the routine, i.e. it should be 0 if all data was read.
RB_SC_WriteTMFifo	Writes data to the telemetry FIFO, the number of words to write is passed in R0 and the address of the data in R2. R0 should be 0 if all the data was written to the FIFO.
RB_SC_TMFifosEmpty	This function checks both telemetry FIFO and sets the zero flag if one or both of them are empty.

Macro	**Parameters**	**Description**
RB_SC_Status	Rx	Returns the contents of the spacecraft interface status register. This indicates the current state of the command and telemetry FIFOs.
RB_SC_FifoEmptyFifo	Rx	Sets the zero flag if FIFO is empty. FIFO should be TM or CM. The state of the zero flag is returned in register Rx.
RB_SC_FifoLoading	Rx	Sets the zero flag if the command FIFO is being loaded. The state of the loading flag is returned in register Rx.
RB_SC_TmWrite Word	Rx	Write the contents of Rx to the telemetry FIFO.
RB_SC_MIReadByte	Rx	Reads a byte from the command FIFO into Rx.
RB_SC_PacketReady		Sets the packet ready signal

6.2.5 16 Bit Board Handler

Module RB_16

Filenames:	Code	RB_16.ASM
	Macro	RB_16.MAC
	Variable	RB_16.VAR
	Make	RB.MAK

The 16 bit board handler provides a set of functions and macros to enable the 16 bit sensor handlers to control and sample 16 bit sensors.