ALCATEL ALENIA SPACE
An Alcatel/Finmeccanica company

AURORA AVIONICS ARCHITECTURE

FINAL REPORT

	Responsability Compagny	Date	Signature
Written by			
Alcatel team			
Verified by			
Bruno MASSON	Data handling section header	5/10/05	B Masson
Approved by			
Bernard ALISON	Project manager	5/10/05	

PT Code :
Emitting entity : OS/I/MC
(original holding)

ALCATEL
SPACE

TABLE OF CONTENTS

LIST OF FIGURES

1. PURPOSE

This document presents a synthetic report of the main results of the Aurora avionics architecture definition study (ESA contract 17451/03/NL/LvH.
It is issued to close the study, in complement with the final presentation held in ESTEC on 6th of September 2005.

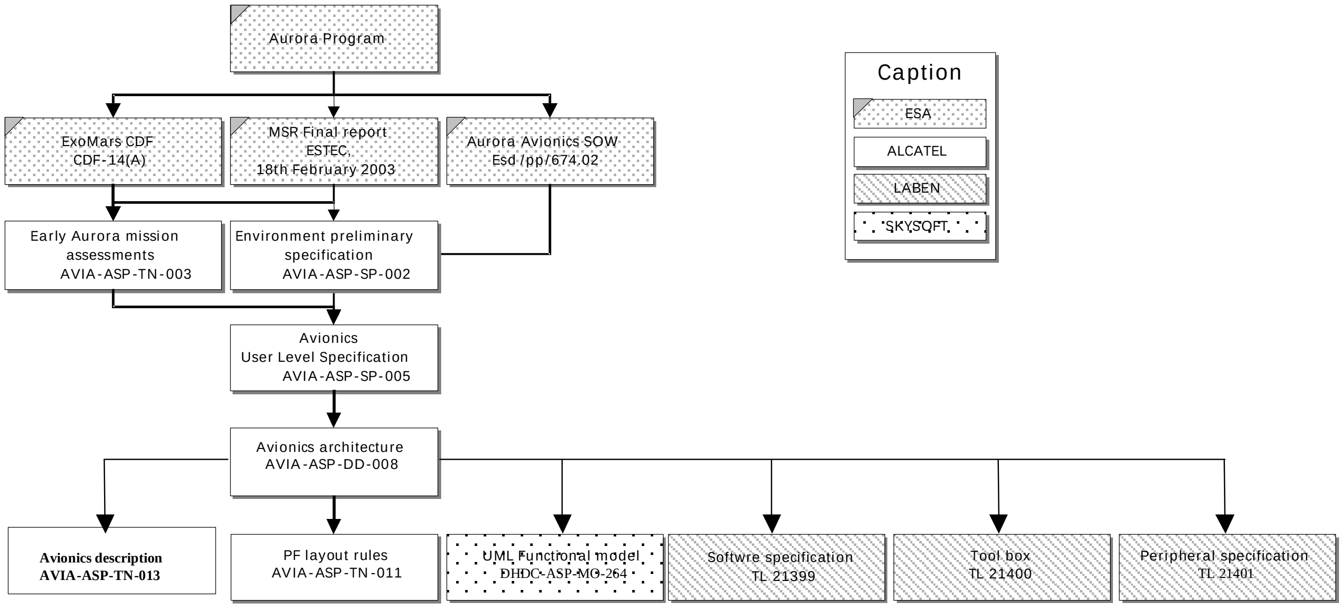
2. DOCUMENTATION

2.1 Applicable document

Aurora avionics Statement of Work	Esd/pp/674.02 iss. 1

2.2 Reference documents

Early Aurora mission assessments	AVIA-ASP-TN-003 iss. 1
Environment preliminary specification	AVIA-ASP-SP-002 iss. 1
Avionics User Level Specification	AVIA-ASP-SP-005 iss. 2
Peripheral modules specification	TL 21401 iss. 1
Comware software specification	TL 21399 iss.2
Avionics Architecture Document	AVIA-ASP-DD-008 iss. 2
Avionics description	AVIA-ASP-TN-013 iss.1
Aurora avionics platform layout rules	AVIA-ASP-TN-011 iss. 1
Aurora avionics UML functional model	AVIA-ASP-MO-264
Aurora avionics road map	AVIA-ASP-PA-007 iss; 1
Tool box data package	TL 21400 iss. 1



2.3 Terminology and abbreviations

ASW	Application SoftWare
ATB	Avionics Test Bench
CFDP	CCSDS File Distribution Protocol
IMU	Inertial Measurement Unit
OBCP	On Board Control Procedure
OBSW	On Board SoftWare
OCM	Orbit Control Mode
PCDU	Power Conditioning and Distribution Unit
RCT	Reaction Control Thruster
PUS	Packet Utilisation Standard
SAM	Sun Acquisition Mode
SGM	SafeGuard Memory
SMU	Satellite Management Unit
SOIS	Spacecraft On board Interface Services
SSDHI	Satellite Software Data Handling Interfaces
STB	Software Test Bench
STR	Star Tracker
SVF	Software Validation Facility
THR	Thruster

3. STUDY CONTEXT

3.1 Study objectives

The main objectives of the present study are:

- identify the needs for Aurora missons and specify a modular avionics able to cover the Data Handling area,
- define this avionics and specify the main modules,
- perform a UML functional model of the avionics.

The specification of the SpaceCraft Core Computer and implementation trade-offs, initially included in the frame of the study, have been contractually excluded to be treated though a independent contract called "HICDS evolution for Aurora".

At the beginning of the study, a clarification of the perimeter of the avionics has been performed and a avionics core has been defined which includes:

- Processing function, with all its hardware resources (processing, memory, power supply) including reconfiguration module, TMTC conditioning, mass memory.
- Hardware functions required to interface with all spacecraft subsystems, through standard busses, standard input/outputs and specific input/outputs if needed.
- Middleware providing all services required by applications to operate the hardware
- CPU and memory allocations for application software needed to manage the mission, the spacecraft, the passenger (if any) and the payload (if any).

Consequently, the avionics core does not include:

- The AOCS sensors/actuators and their software (if embedded into the units)
- The electrical & power sub-system including conditioning and distribution functions
- The Telecommand / Telemetry Sub-system (RF receiver/emitter and associated -
- The pyrotechnic devices and associated mechanisms
- The pointing mechanisms for solar arrays, antenna….
- The thermal sensors and actuators

3.2 Study logic

The main steps of the study are:

- mission reference analysis
- avionics user requirements
- avionics architecture definition
- avionics UML model development
- platform layout rules definition
- building blocks and tool box approach
- peripherical core hardware specification
- middleware specification
- avionics PDR

The following figure describes the logic of the study and the interaction between the different tasks:

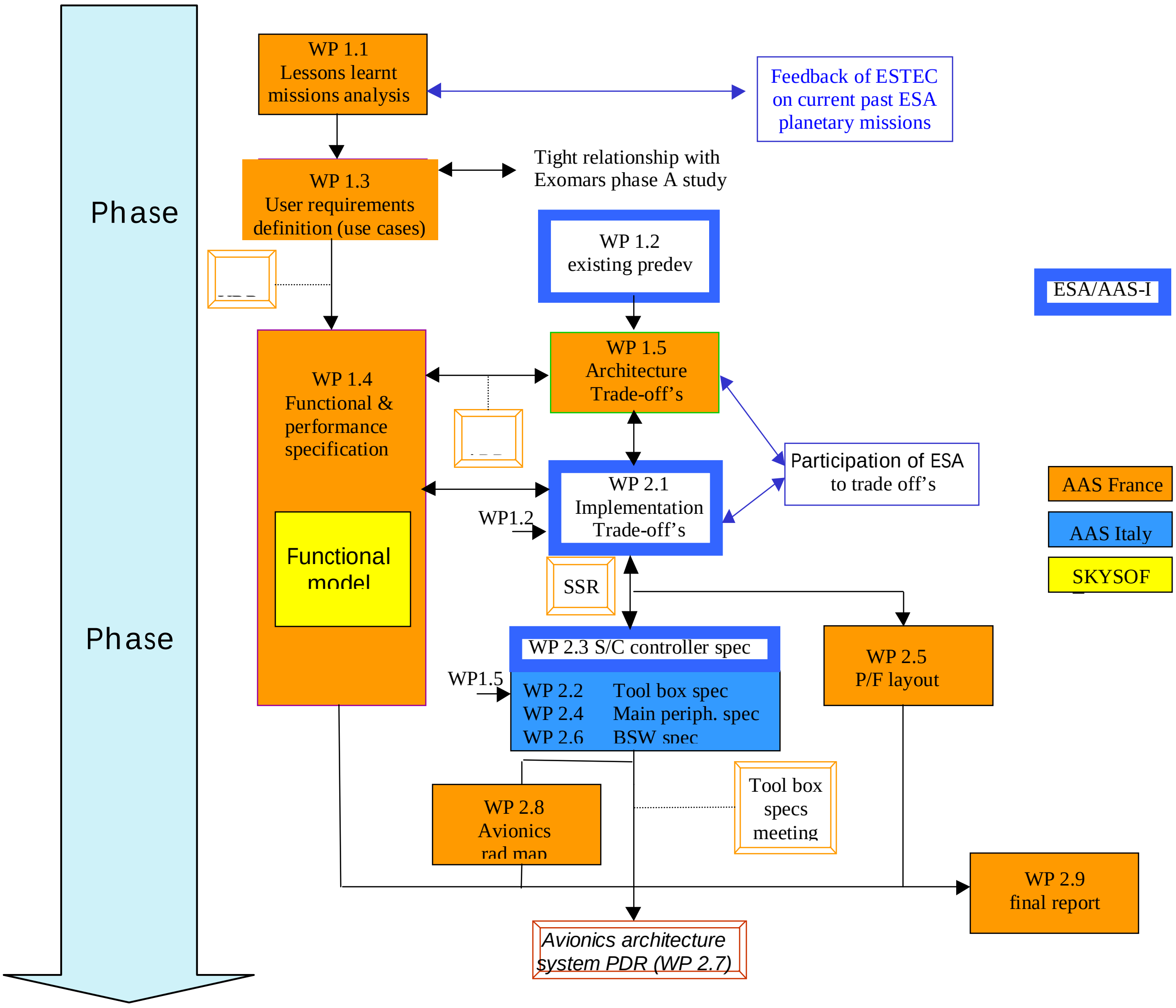


Figure 3-1 Study logic

3.3 Main drivers of the study

The key elements which drive the Aurora avionics study are:

Advance solution
Autonomy
Modularity
Adaptability
Standardization
Cost reduction

4. REFERENCE MISSION ANALYSIS

The EXOMARS, MSR and ERC missions have been retained as representative mission inputs for the AURORA avionics study.

The EXOMARS mission is composed of an Orbiter carrying a Descent Module to land the Rover on the surface of Mars. When in its final Mars Orbit, the Orbiter will operate as a data relay satellite between Earth and the Rover. Each module (Orbiter, DM and Rover) is requiring to embed an avionics.

The MSR mission is composed of:

- a Carrier which carries the Descent Module to land yje Mars Ascent Vehicule (MAV) and surface module on the surface of Mars.
- an Orbiter which captures the Sample orbiting around Mars Rendez Vous operation, returns back to Earth and release near the Earth, the Earth Re-entry Capsule (ERC), where the sample is sealed in.

A dedicated avionics is embedded into the Orbiter, the Carrier, the Descent Module and the Mars Ascent Vehicule.

The objective of the ERC is to land the sample container on Earth without breaking the sealing system. The ERC is a passive element with no “intelligence”. Consequently, there is no interest to implement an avionics as defined in this study.

4.1 Mission analysis

The reaction time and the consequences at system level of a failure are depending on the phase criticality.

The phases are split into Fail-safe and fail-op phases :

Fail-safe : on failure event, the S/C is allowed to interrupt the current operation. The S/C is then autonomously set in safe mode, using back-up hardware configuration. For some cases, the safe mode triggering is preceded by a manoeuvre. Typically during Terminal Rendez Vous, a manoeuvre must be executed first to avoid collision with the sample.

Fail-operational : on failure event, the S/C is not allowed to trigger the safe mode, otherwise it will induce a loss of the mission. The S/C shall recover the failure by switching over to the back-up configuration and shall resume the current operation, within a limited time duration., Fail-op operations are usually time critical.

For EXOMARS, the phases which are considered to be fail-op, are :

- LEOP : any interruption in the sequence of S/C initialisation will be harmful to get the correct S/C attitude (i.e. SA deployed and Sun pointed and LGA Earth pointed).
- MOI : the S/C will miss the Mars orbiting if any interruption during the MOI manoeuvre.

In the following table, is presented the maximum assessed time delay of interruption, function by function. It means to avoid the loss of mission, any dis-functioning in the listed functions shall be detected, recovered on board and the allowed mode (i.e. safe mode or current mode resumed) triggered.

		Cruise						MOI	Mars Orbit				
Phases	LEOP	Commis-sioning	Inner	Outer	Rover/DM health checks	TCM	Mars approach		DM deploy.	Science	Search	Conjonction	Terminal RDV
Autonomy	FO	FS	FS	FS	FS	FS	FS	FO	FS after re-orbit man.	FS	FS	FS	FS with avoidance man.
Functions													
AOCS/Prop	< 100 sec (TBC)	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 1 to 10 sec (1)
Power (TBC)	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec
Thermal Regulation	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec
Communication	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h

Tableau 1 : Exomars phase time criticality

(1) the maximum time delay in terminal RDV depends on the relative speed and the distance min between Orbiter and sample. A 5 cm/sec speed and 50 cm distance (min distance for a LIDAR) will lead to ~5 seconds.

For MSR Orbiter, the phases which are considered to be fail-op, are :

- LEOP : any interruption in the sequence of S/C initialisation will be harmful to get the correct S/C attitude (i.e. SA deployed and Sun pointed and LGA Earth pointed).
- MOI : the S/C will miss the Mars orbiting if any interruption during the MOI manoeuvre.
- OS capture : when the Orbiter is very closed to the sample, it is no longer possible to trigger a safe mode without risk of collision and hazardous consequences.
- TEI : idem MOI
- ERV targeting – Earth avoidance : the Earth avoidance is a strong requirement. As soon as the Orbiter is on Earth collision trajectory (ERV targeting), it is assumed that ground has not enough time to react in case of anomaly, and the Earth avoidance manoeuvre shall be performed.

		Cruise						MOI	Mars Orbit						Cruise back			
Phases	LEOP	Commis-sioning	Inner	Outer	ERC health checks	TCM	Mars approach		Mars Orbit	Conjonction	OS search	terminal RDV	OS capture	TEI	Cruise back to Earth	TCM	ERV targeting & release	Earth avoidance
Autonomy	FO	FS	FS	FS	FS	FS	FS	FO	FS	FS	FS	FS with avoidance man.	FO	FO	FS	FS	FO	FO
Functions																		
AOCS/Prop	< 100 sec (TBC)	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	<1 to 10 sec (TBC)	< 1 sec (TBC)	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec (1)
Power (TBC)	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec
Thermal Regulation	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec
Communication	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h

Tableau 2 : MSR Orbiter mission phase time criticality

(1) a probability of crash on the Earth shall be also considered. A typical value is 10^{-6} (CNES MSRO requirement).

For MSR Carrier and MAV, the phases which are considered to be fail-op, are :

- LEOP : any interruption in the sequence of S/C initialisation will be harmful to get the correct S/C attitude (i.e. SA Sun pointed and LGA Earth pointed).
- Mars avoidance : It is assumed that ground has not enough time to react in case of anomaly, and the Earth avoidance manoeuvre shall be performed.
- EDL : no safe mode is allowed during in the EDL sequence.
- Ascent : any interruption during the MAV ascension will cause the MAV to crash on Mars surface sooner or later.
- OS capture : the MAV shall avoid to trigger the MAV safe mode during the OS capture operation.

		Cruise											Mars Orbit
Phases	LEOP	Commis-sioning	Inner	Outer	DM/MAV health checks	TCM	DM targeting & separation	Mars avoidance	Coast	EDL	Surface	Ascent	Mars Orbit
Autonomy	FO	FS	FS	FS	FS	FS	FS	FO	FS	FO	FS	FO	FS/FO*
Functions													
AOCS/Prop	< 100 sec (TBC)	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 0.4 to 1 sec (1)	< 100 sec	< 1 sec (TBC)	< 1 to 10 sec (2)
Power (TBC)	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec
Thermal Regulation	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	< 100 sec	NA	< 100 sec	< 100 sec	< 100 sec
Communication	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	< 24 h	NA	< 24 h	< 24 h	< 24 h

Tableau 3 : MSR Carrier/MAV mission phase time criticality

(1) the time criticality during EDL is driven by the parachute opening date (TTG : “time to go”) computation. The TTG value is derived from the time-stamping of the “5g” event. Assuming the accelerometer measures are acquired at 0,2 s rate, any interruption in the acquisition shall not be longer than 0,4 to 1 second (best case). Then the parachute opening can be triggered at TTG +/- 1 second.

(2) As for the Orbiter, when the OS/MAV is close to Orbiter, transition to safe mode is no longer allowed. The MAV will operate in FO mode.

4.2 Specific requirements

Analysis of the missions has allowed to identified some specific requirements which are not usual in previous projects.

Separable stages which are jettisoned during the mission life. As an example the MSR carrier is composed of 4 modules (carrier, descent module, MAV first stage and MAV second stage) which are connected together through jettisonable links.

At least two modules require the implementation of a sleep/wake-up function in order to limit the power consumption: the descent module which is powered by battery is switched off during the coast phase (phase between separation from Orbiter and start of entry into the Mars atmosphere) and the rover/ MAV are switched off during the Mars night.

Orbiter shall provide the capability of relaying data between Ground and other modules and will implement spacecraft to spacecraft communication. The other modules (descent module, rover and MAV) will implement capability to communicate with the Orbiter in addition with more classic communication with Earth.

4.3 Environment requirements

This section summarizes the most dimensioning requirements applicable for the environment of Aurora modules.

4.3.1 Natural environment

Natural environment depends of the phase in the lifetime of the module.

Ground life covers the activity at system level for integration and test onto the module

The launch from Earth is characterized by a pressure loss from 1050 mb to 10^{-5} mb with a pressure rate between 20 mb/s and 50 mb/s.

During the orbital life, the modules are submitted to a radiation environment (< 10 krads) and unit temperature is directly defined by the module thermal control: the temperature of internal electronic units is controlled between 0 °C and 40 °C except for the rover for which the baseplate temperature is maintained lower than – 40 °C.

The descent phase is applicable only to the rover and the MAV. During descent the module is plunged into Martian atmosphere, mainly formed of CO_2 and units may be subject to particulate contamination from the thermal decomposition of the heatshield.

On the Mars surface, the module is subject to the dust deposition and the temperature variation due to the day/night period and a low level of radiation (< 2 krads for a 6-month mission).

4.3.2 Mechanical environment

Mechanical environment is defined by the launch phase and depends of the selected launcher. The specific study of the MAV will define if the specific loads generated by the launch from Mars are lower or upper than these generated during the launch from Earth. In the second case, specific requirements shall be added to units embedded in the MAV.

The electronic units shall withstand to the shocks generated by pyro mechanisms as deployment, stage separation.
For units embedded in the Descent Module (rover and MAV data handling units), additional requirement shall be added to cover Mars landing shock. Preliminary figures are TBC:

Shock level	Shock duration
120 g	50 ms

Tableau 4 : Mars landing shock qualification levels

4.3.3 Radiation

The total radiation dose seen by an unit is depending of the lifetime in orbit and the thickness of protection. The specification can be sized for the worst case : orbiter with a 5-year mission. In most of applications, a protection of at least 0.8 mm thickness can be performed by the mechanical structure for units located inside the module. So a 30 krads total dose shall be considered for data handling units.

The rover electronics can be specified differently. During the cruise from Earth to Mars, the rover is stowed into its carrier and relatively well protected against particle flux. The duration of the mission on Mars surface does not exceed 3 months and Mars provides a very efficient protection (16 g/cm² thanks to Mars atmosphere for 2π strd and Mars planet for the other 2π strd). So a 2 krads total dose shall be considered for rover data handling units.

Electronic parts of Aurora avionics shall be latchup free.

The occurrence of critical SEU (i.e. SEU able to affect the mission – transition to safe mode or reconfiguration during critical phase) shall be lower than the occurrence divided by ten (TBC) of hardware failure.

4.3.4 EMC constraints

The Aurora modules include a UHF receiver, to communicate with other modules when they are closed from Mars. The MSR orbiter could include also a radio detection finder (RDF), for the Rendezvous phase with active orbiting samples. Both instruments are susceptible to perturbation by emissions radiated from the units.

The units shall not radiate noise greater than – 10 dBµV/m at one meter distance, in the band 390 MHz to 450 MHz.

5. USER REQUIREMENTS

5.1 Main drivers

Main drivers induced by AURORA missions are:

Strong need for autonomy: Indeed, communication from spacecraft with earth may be difficult due to distance, and long non visibility periods. This constraint is even more stringent if an orbiter is used to relay communication between earth and a rover.

High availability during short periods: During some critical phases, as orbit insertion, any abort, interruption, or degradation of performances may induce the loss of the mission.

Reliability: It is a key driver, independently of availability constraints.

Mass/ Power consumption / volume/ cost: These drivers, ordinary for spacecraft, are however made more stringent in Martian missions, especially mass and power parameters. Snowball effect is specially important on landers or module ascent vehicles.

Parallel to these operational constraints, some design drivers have been identified:

Genericity: Avionics shall be used for a large spectrum of Martian missions. And even for low earth orbit satellites.

Modularity: Avionics shall allow the development of standard SW applications (GNC, EPS, TH, …) independently of HW. I.e. HW evolution shall transparent for SW applications. Modularity is then bringing the following properties:

Interfaces specifications driven by Standards when possible,
Functional segregation to allow validation at module level,
Capacity to implement new module on demand limiting impacts on avionics,
High reuse feature.

Evolution: Avionics shall be compatible, as far as possible, to adaptation to specific needs, and to evolutions. This driver is linked to genericity and modularity.

Requirement	Need/justification	Driver
Modular architecture	Ability to cope with ESA Martian missions Ability to cope with other missions (Observation/ …) With various design constraints (P/L processing, centralised, decentralised…) To cope with distribution of the processing on the S/C (multi-modules approach) Ability to embed or not functions related to autonomy	Minimisation of non recurring costs from one mission to the other Compatibility with standards (PUS, SOIF, SSDHI, buses)
Incremental validation capability	To permit coupling of items issued from various companies	Geographical return Non proprietary blocs or functions availability Available advanced technology
High level of performance of the digital system	To avoid heavy engineering activities related to performances optimisation: "To forget the digital system in the avionics behaviour"	CPU throughput and bus consistent margins
Robustness & reliability	To satisfy fail safe but also failop constraints over long durations	
Reliable development process	To limit costs and risks	Automated process - Functional Modelling - Automatic code generation - Automatic production of : - data base inputs - testing scenarios
Perennial development tools and technology	To secure capacity to produce avionics over a 10 to 15 years periods (TBC)	Target missions in the interval 2008-2015.

5.2 Definition of user requirements

The user requirements have been defined from the reference missions as inputs and with the help of Use Case Diagrams (following the UML standard)

As defined in the following UCD, the users involved in Aurora Avionics are two kinds:

The "Ground users". Complexity of such missions requires several operators as to their authority and mission involvement.

"Mission Centre" dedicated to scientific analysis such as payload instrument operations, image acquisitions; … Several mission centres may be active at the same time.

"Control Centre" dedicated to spacecraft control. A single control centre is considered.

"Aurora Spacecraft" which embeds Aurora Avionics. Two spacecraft types are identified:

- The “**Aurora Carrier**” , which is in charge of interplanetary cruise to release its embedded spacecraft passenger, and acts as a relay between Earth and another spacecraft (mainly its passenger):
  - MSR Carrier,
  - ExoMars Orbiter and MSR Orbiter which support additional capacities such as Mars orbit insertion, rendez-vous, and payload operations (docking system or scientific).
- The **“Aurora Passenger”** (space probe), which is in charge of Mars landing and surface operations (payload embedded), gathers:
  - ExoMars Rover which support locomotion capacity,
  - Main Ascent Vehicle (MAV) which supports smart landing and take off to Mars orbit.

Avionics missions are specified considering three main levels of requirements

- Operational needs: Collection of requirements related to Earth ground mission & control centres needs. These needs are mainly driven by Mars eclipses and Earth rotation which limit communication session duration, Sun conjunctions and long distances which limits communication data rate.
  - **Provide spacecraft commandability** to permit operation of the spacecraft by earth ground users.
  - **Provide spacecraft observability** to provide ground with all data relevant to control of correct operations and defaults. Two main categories of data are distinguished: periodic data for normal operations (housekeeping, passenger telemetry) and aperiodic data for specific operations and events such as dumps and command acknowledges.

- On board functional needs: All needs required by spacecraft functions and subsystems
  - **Preserve spacecraft mission**: ensuring mission time line execution, handling any single failure in order to maintain the mission, support vital functions.
  - **Provide support to all subsystems and functions**: command and acquisition, on board time, FDIR, mode management
  - **Relay telecommunications**: to ensure communication between rover and ground through orbiter servicing.

Controller core needs: All services required to controller core

Provide support to all subsystems and functions: processing resources, data storage, communications, command and acquisition interfaces.

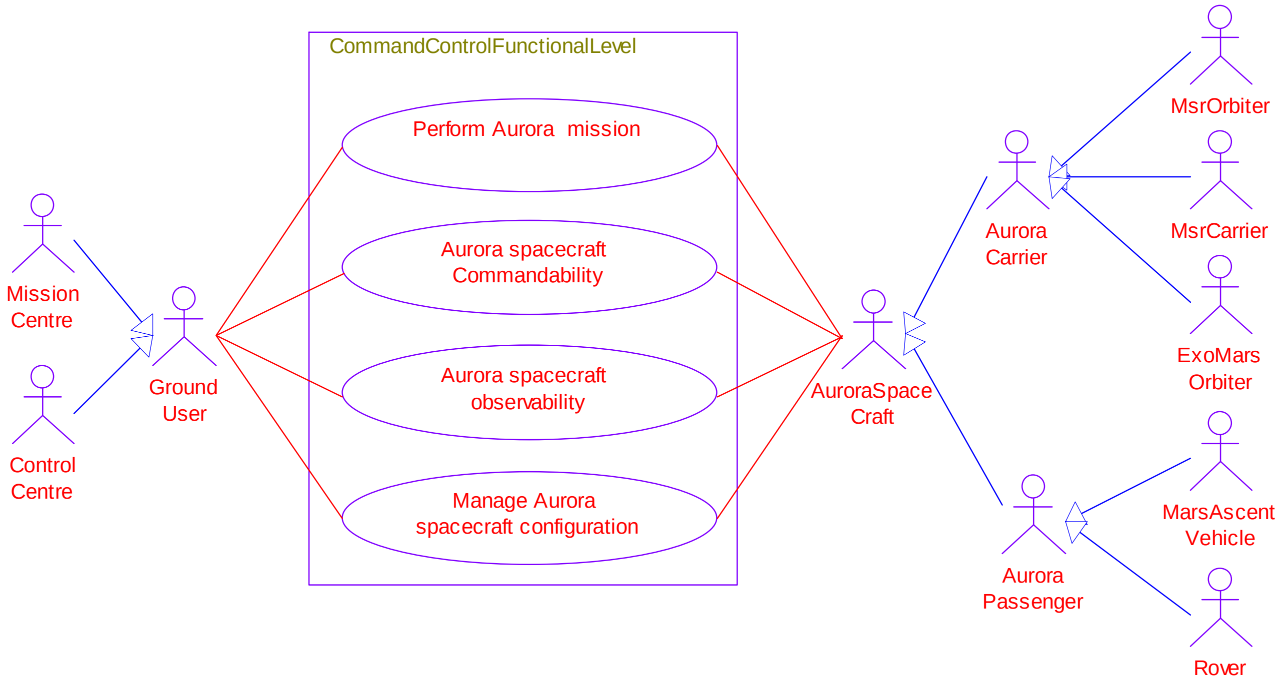


Figure 5-1 UCD Command control functional level

5.3 Functional architecture

The different needs have been written in a such way leading to define a generic and modular avionics. As Aurora Avionics study is modularity driven, Avionics shall integrate as far as possible the commonalties, in particular concerning the interfaces. Mainly, generic Aurora Avionics external interfaces are shown in Figure 5-2.

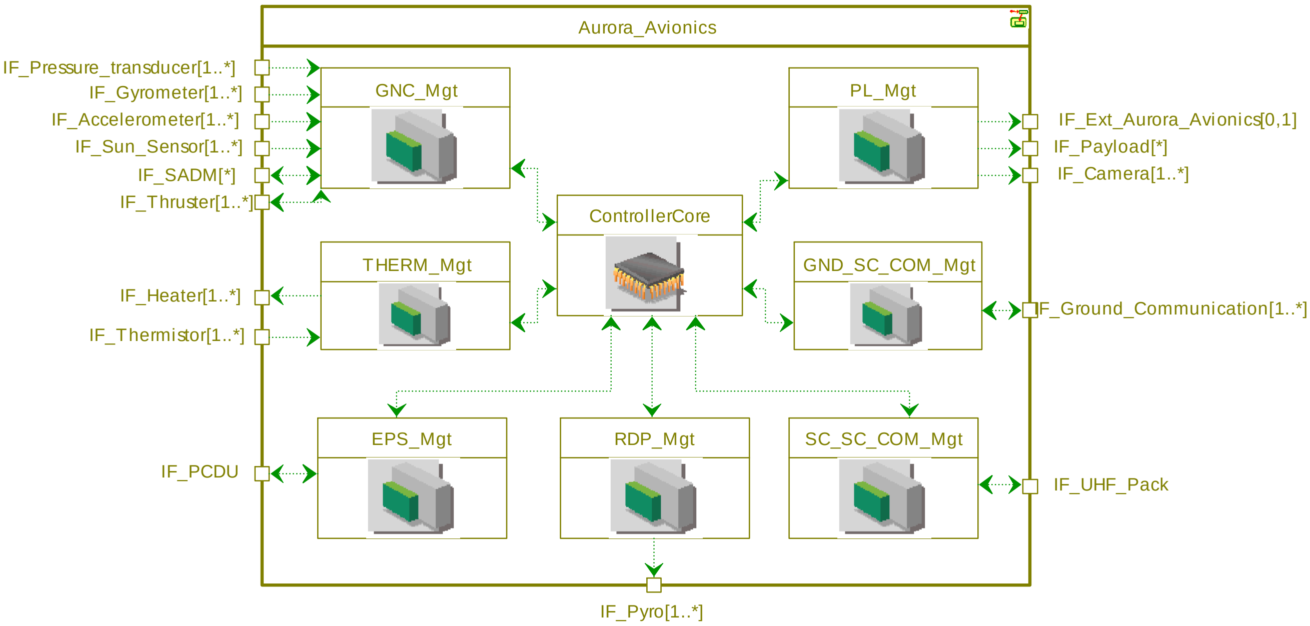


Figure 5-2 External interfaces of generic Aurora avionics

Functional needs have been split following a functional architecture.

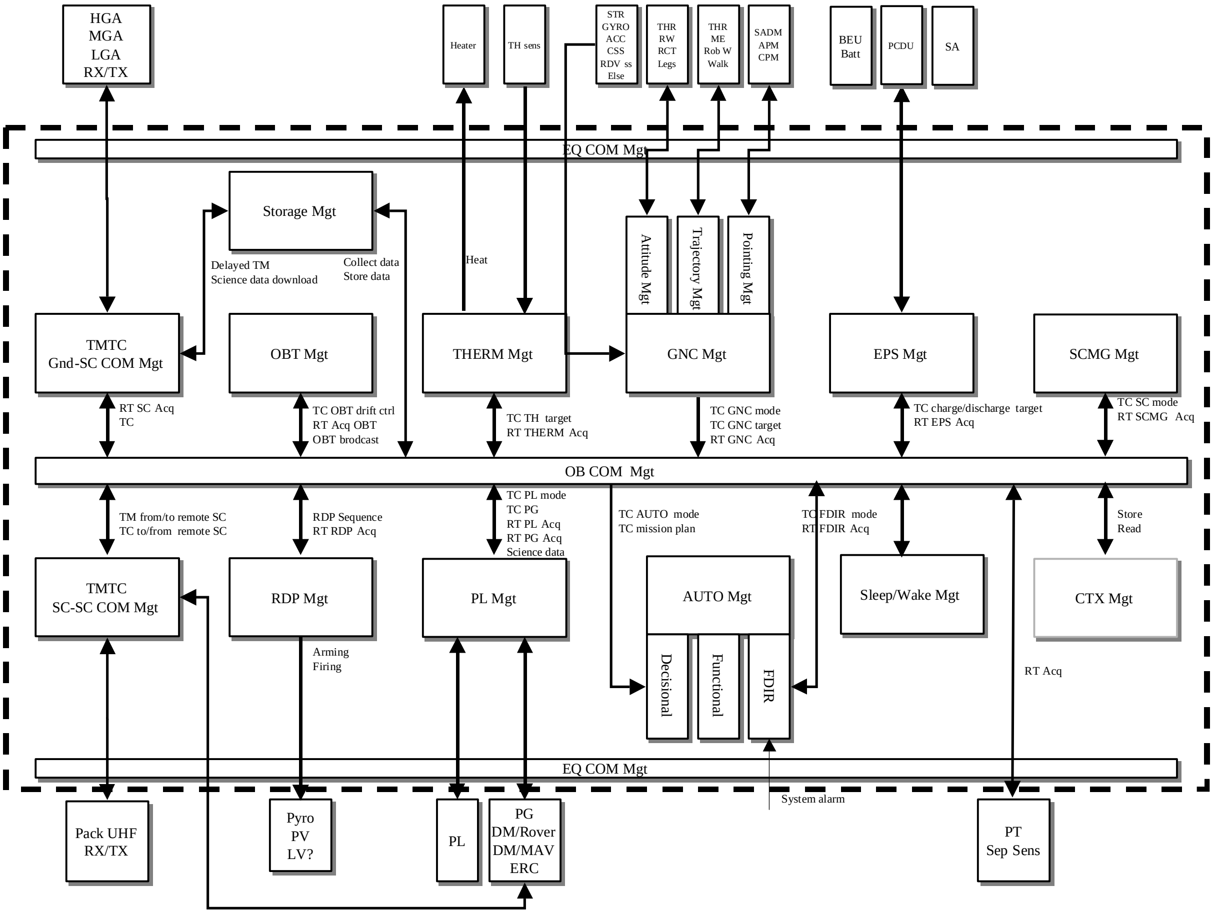


Figure 5-3 Avionics functional architecture

The proposed functional architecture has been defined in respect of following rules:

- Independency between modules in order to limit the exchanged data between modules
- Granularity of modules shall allow to suppress non-used modules depending of the mission.

5.4 Main trade-offs

Following trade-offs have been conducted.:

- PM redundancy
- Function repartition
- Interface with external medium rate busses
- TM management

Conclusions of these trade-offs are shortly presented hereafter.

5.4.1 PM redundancy

Choice of redundancy is driven by following requirements and needs:

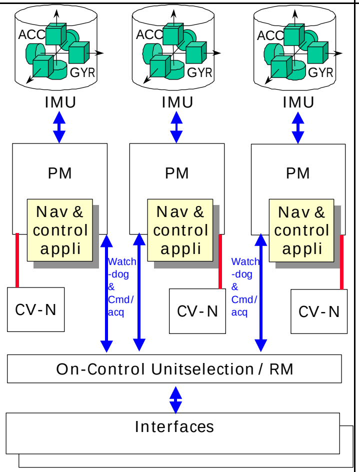
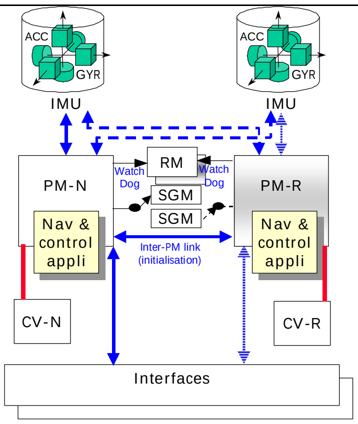
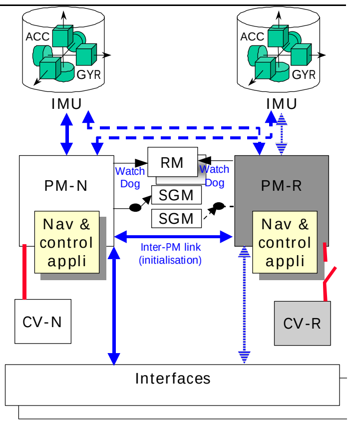
- During identified short critical mission phases, no failure shall induce loss of mission longer than 500ms. This behaviour is called FAIL-OP.
- During all other mission phases, criticity is lower and loss of mission of around 20 seconds is allowed. In these phases, both PM will be used in cold redundancy.
- Genericity and evolution: the design shall be usable for a large spectrum of missions
- Reliability: FAIL-OP requirements shall not lower the Avionics reliability.

Following solutions have been examined for PM redundancy :

- Solution 1 : **Cold redundancy of 2 PM**: One PM is active, while the redundant one is OFF. In case of active PM failure, the second one is switched ON and using SGM, the mission is restored.
- Solution 2 : Warm redundancy of 2 PM: One PM is active and used for control, the redundant one is ON, in shadow mode or in standby mode. In case of active PM failure, the second PM is then used for control to restore the mission:
  - Warm redundancy in shadow mode, Redundant PM and redundant equipment's are active. Redundant PM is fed with all data necessary to run applications (via acquisitions on redundant sensors, or via inter PM link), but can not send any command. This mode allows faster reconfigurations, but increases complexity.
  - Warm redundancy in stand by mode, Redundant PM and redundant equipment's are active. Only permanent functions of Redundant PM are active. In case of active PM failure, necessary data to restore mission shall be read in SGM by redundant PM, and OBSW shall be partially started. So, reconfiguration will last longer than in shadow mode, but complexity will be lower.
- Solution 3 : Majority voting based approach. 3 PM are used ON, and a majority voting logic is used, i.e., PM which behaviour is the most singular (behaviour estimation based on criteria to be defined) is not taken into account. For example, an actuation is done as soon as at least 2 PM's request the same coherent actions.

Solutions 1 and 2 are based on a common Safe Guard Memory (with write/read principles which shall secure the data corruption, such as use of 2 buffers), or on an inter-processing communication capability.

The following table presents the principle of these 3 solutions :

	Majority Voting	**Warm Redundancy**	**Cold Redundancy**
Principle of work overview			
Reconf. Duration	No mission interruption	< 500 ms (TBC)	A couple of seconds Not compliant
Reliability	Reduced if not compatible with cold redundant op.	Mid	Mid
Complexity	High	High	Mid
Mass	Higher for majority voting	Mid	Mid
Power	High	Mid	Low
Cost	High	Mid	Mid
Validation	?	High/Mid	Low
Generality	Low (Hot redundant & majority voting architecture have to be strongly optimised	High for Standby Mode Mid for Shadow Mode	High

The **majority voting** concept is certainly the most interesting in terms of mission availability (it is close from manned flight architecture). However this solution drawbacks are cost induced by system complexity (and potentially, associated validation), its mass and power impact if operated in majority voting configuration; which is the case in phases where the power is possibly critical because the spacecraft is running on batteries (entry and insertion).
It shall be noted that hot redundancy with majority voting is not easily compatible with cold redundant operation.

Cold redundancy is not adapted to rapid reconfiguration (due at least to the processing unit initialisation duration).

Consequently Warm redundancy appears as the most efficient solution for critical phases. Warm redundancy in standby mode solution, shall be chosen, due to its higher genericity. The critical point is in the data-transfer management between the nominal and the redundant PM, which requires a high rate data-link and mechanism to avoid data-corruption (between failure occurrence and detection). This can be ensured through a secured SGM or inter-PM communication links.

5.4.2 Overall function repartition

As a first trade-off, 2 solutions have been examined:

- **Highly centralised avionics**: power distribution and controller in a single box
- **Avionics split in 2 cores**: one controller core and one power conditioning core

Architecture	Highly Centralised	Split in 2 cores	Comment
Mass, Power, Volume	+	-	Centralised architecture is the best solution to minimise mass, power and volume
EMC	-	+	More risks on EMC, if power and low level signals are in the same box
Modularity Adaptation to various missions	-	++	Highly Centralised Avionics can not be optimal for all AURORA missions
Cost	=	=	Lower recurring cost for highly centralised avionics, as long as no modification is needed. In case of design modification, which should the case in AURORA missions, highly centralised avionics appears much more expensive, due to non recurring costs
Development plan	-	+	Possible parallel development if 2 cores

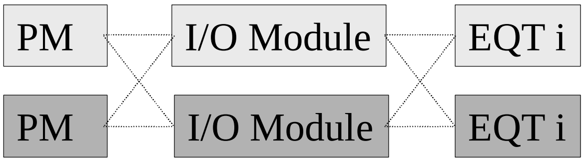
In conclusion of this trade-off, due to modularity needs and to various kinds of missions (a rover is different from an orbiter), AURORA avionics will be oriented between 2 cores :

- **The controller** being the main communication node (intern to the S/C and toward exterior), and in charge of the spacecraft bus related processing and data-management
- **The power conditioning**, which is the core of the power supply network (not part of this study).

The second trade-off is related to the I/O cross-strapping.

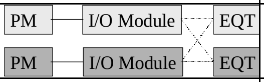
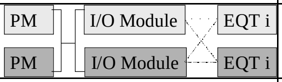
The problem of cross coupling is the following:

Should each I/O module access to each equipment?
Should each processor access to each I/O module?



The I/O interface functions repartition, grouping, redundancy and cross-strap approach is strongly driven by the reliability analyses.

Cross strap between Processor and I/O module has first been examined:

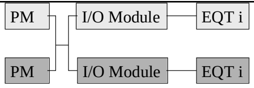
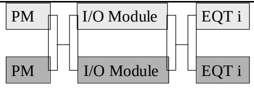
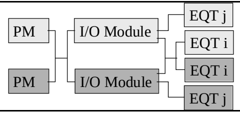
Architecture	No Cross strap between Processor and I/O modules	Cross strap between Processor and I/O modules	Comment
			
Mass, volume, cost	+	-	Cross strap induces moderate increase of mass, cost, volume, and complexity
Reliability	-	++	If no cross strap, loss of a Processor in case of I/O module failure!
Fail Op Capability after one failure	--	+	i.e: loss of FAIL OP capability after one I/O module failure

In conclusion of this trade-off, for reliability and Fail Op constraints, **full cross strap between Processor and I/O modules shall be achieved**.

Then, **cross strap between I/O modules and equipment's** has been examined. Three options have been taken into account:

- No cross strap: It means that each equipment is redounded one by one: For example, at least 3 nominal Reaction Wheels and 3 redundant Reaction Wheels, instead of 4 cross strapped Reactions wheels.
- Full cross strap: From each I/O module, it is possible to command and monitor each equipment or parameter.

Partial cross strap: some equipment's or parameters can be commanded and monitored from both I/O modules (shared), as other ones can only be commanded and monitored from one I/O module (half satellite approach).

Architecture	No Cross strap between I/O modules and equipments	Full Cross strap between I/O modules and equipments	Partial Cross strap between I/O modules and equipments	Comment
				
Mass, volume, cost	--	-	+	Cross strap induces moderate increase of mass, cost, volume, and complexity No cross strap induces strong increase of mass, cost, and volume
Reliability	-	++	+	If not full cross strap, loss of all not cross strapped equipment's linked to this I/O module in case of I/O module failure
Fail Op Capability after one failure	=	++	+	i.e: No loss of FAIL OP capability after one I/O module failure

In conclusion, experience has shown that cross-strap strategy between I/O modules and equipments has to be performed in a case by case approach, at system level, taking into account mission constraints and characteristics of each equipment (reliability, interfaces,…). Chosen strategy is the usually cross strap between I/O modules and external units, optimising cost, mass, power and reliability.

5.4.3 Interfaces with external medium rate busses

Interfaces with external low rate bus, as 1553 bus, are an important item, in terms of CPU performances. Indeed, when processor frequency increases, if I/O access times remain constant, duration of S/W routines will not change significantly.
Interface with low rate external buses can be implemented as independent process, or as service.

Independent process or hardware components have been selected to interface with low rate buses. Indeed, Processor performances shall not be degraded by interfaces access times. HW implementation shall be preferred, but shall remain highly configurable.

An example of the implementation of independent process is provided Figure 5-4 to illustrate the selected solution.

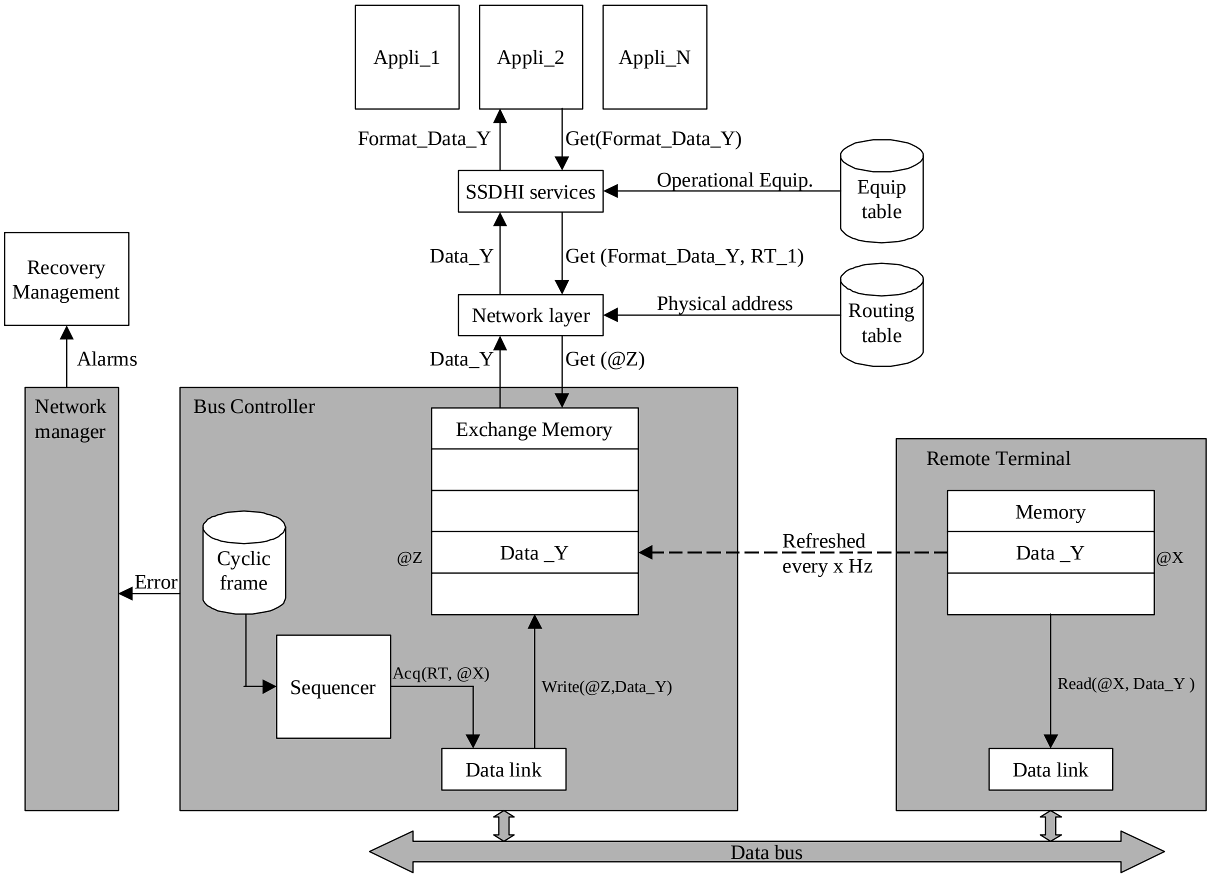


Figure 5-4 Synchronous acquisition data concept

5.4.4 TM management

A first trade-off is linked to TM management strategy versus cyclic TM.
Following options have been foreseen for cyclic TM:

- Each application or service sends, at each of its own cycle, to TM service all data which could be needed for TM. Independently of applications, TM service will use or not these data.
- Each application or service updates, at each of its own cycle, a table of all data which could be needed for TM. Independently of applications, TM service will acquire or not these data.

Synthesis of the trade-off is presented hereafter:

CYCLIC TM	TM data acquired by TM service	TM data sent to TM service	Comment
Modularity	=	=	
Data exchange	+	- -	Depending of TM frequencies, many useless exchanges if data are sent to TM service at each application cycle

In conclusion, for data exchange volume reasons, **cyclic TM data shall be requested by TM service to applications or services**, following Telemetry plan configured by Ground.

A Second trade-off is linked to TM management strategy versus asynchronous TM. Following options have been foreseen for asynchronous TM:

- When an asynchronous TM is to be sent by an application or service, its sends asynchronously to TM service all data needed for TM.
- At each cycle, TM service checks if any application or service requests to send an asynchronous TM, and, if there is a request, acquires it.

Synthesis of the trade-off is presented hereafter:

Asynchronous TM	TM data acquired by TM service	TM data sent to TM service	Comment
Modularity	=	=	
Data exchange	-	+	A little bit more exchanges due to “polling” phase if TM service manages asynchronous TM
Observability in case of failure	-	+	It seems safer if application or service is in charge of sending asynchronous TM

In conclusion, **asynchronous TM data shall be sent to TM service by relevant application or service**.

In synthesis, **centralised TM service shall be implemented**:

- Acquiring cyclic TM at commanded frequencies.
- Receiving asynchronous TM from relevant services or applications

6. SOFTWARE ARCHITECTURE

The Aurora Onboard Software consists of the following parts:

- SMU System Software
  - Boot software (Firmware)
  - I/O Drivers (low level, HW related)

- Operating System and Language related libraries
  - Real-time Operating System (like OSTRALES or other systems)
  - OS Library
  - Runtime Library (C or Ada)

- I/O-System (high level I/O services) – typically SOIS (SW) layers and SSDHI

- Application Libraries
  - PUS Library and eventual additional DH services libraries
  - AOC Library
  - Mathematical Library

- System Management Application Software offers the central services required for the application operation:
  - Inter-application SW communication network management
  - Core Data Handling and TM/TC management application, including TM Memory Management
  - Command/Control (on-board implementation of satellite Modes) Modes management application, including units management
  - FDIR, HK-monitoring and event management application, including Safe Guard Memory Management. The FDIR receives from the subsystem the parameters to be monitored and according to its configuration (in function of the satellite C/C mode), it can take the necessary autonomous reconfiguration actions.

- Subsystem Management Application Software
  - EPS management application
  - Thermal control application
  - TT&C management application
  - Payload Management Application
- AOC Application
- Test interface (SIF Library, etc.)

The OBSW application layers are to a large extend based on generic mechanisms, which can be parametrised either by the System Data-Based (SDB) or to some extend in flight by ground command. This feature is major to ensure a robust and convenient maintenance of the ASW.

The S/W ensures also a large modularity, which favors the re-use, development schedule and the S/W maintenance.

A S/W Bus provides standard "horizontal" interfaces to all the applications enabling telecommand and telemetry exchanges. In particularly, this enables the subsystem application to relies on the system management application to select and switch ON or OFF, the equipment to be set on-control according to the current satellite mode, and unit status.

The application are in charge to get or send from the system I/O services, respectively acquisition or commands.

The Figure 6-1shows the composition of the different software layers.

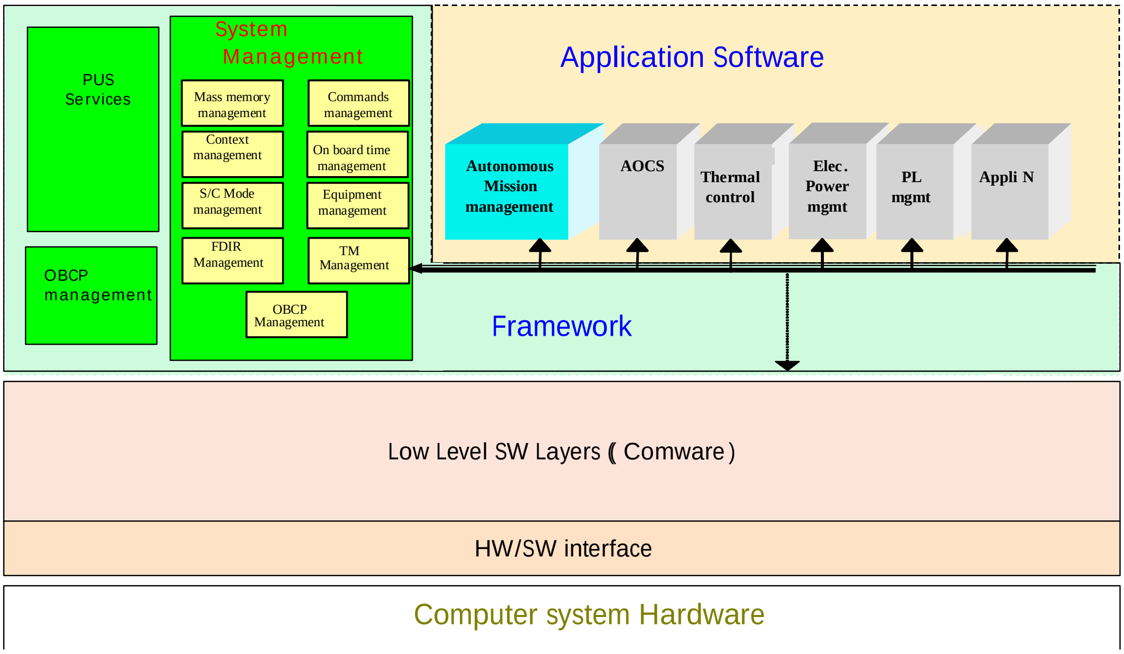


Figure 6-1 : OBSW architecture overview

The System Data Base parts associated to the avionics shall provide the modularity and mechanisms associated to the OBSW modularity

The Figure 6-2 illustrates the links between the Satellite Data Base and the software architecture.

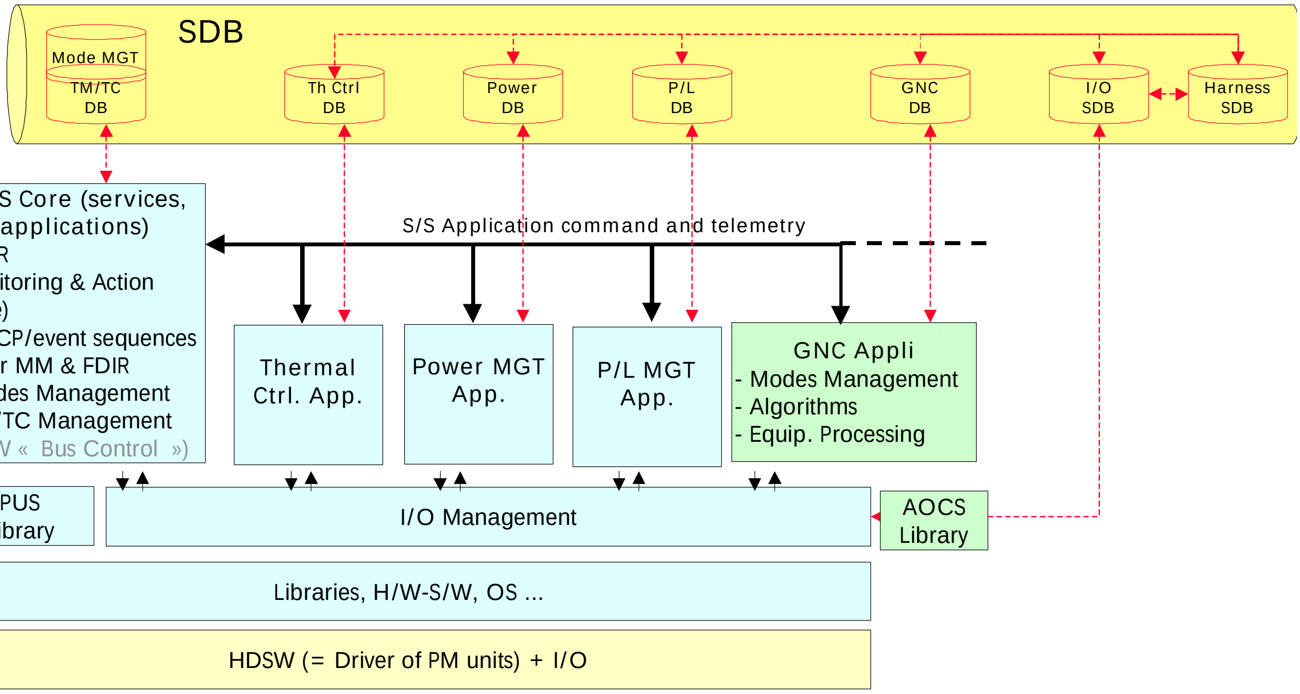


Figure 6-2 OBSW modularity and link with the Satellite data Base

7. PLATFORM LAYOUT RULES

7.1 Power protection and commutation

A trade-off has been performed between SSPC based and fuse based power distribution architectures.

The two main functions, such as protection and ON/OFF function, will be either located in the PCDU or at equipment level. The main possible implementations are proposed here after.

7.1.1 ON/OFF switched power line

The functions needed to be implemented are :

protection
nominal ON/OFF switching
contingent ON/OFF switching.

Taking into account equipment architecture, the functions shall be implemented as follows :

Protection & Nominal ON/OFF commutation implemented within the PCDU – Contingent function implemented in power bus user equipment

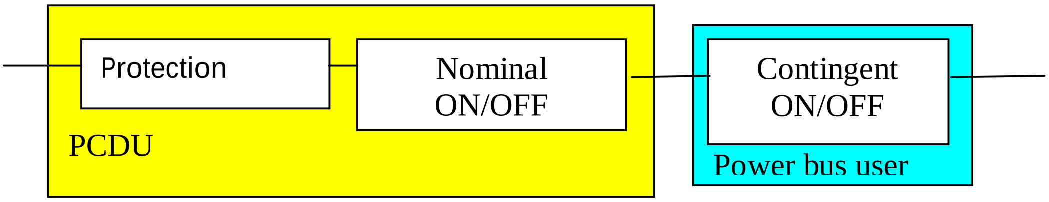


Figure 7-1 ON/OFF switched power line function implementation 1

Protection & Contingent ON/OFF commutation implemented within the PCDU – Nominal ON/OFF commutation implemented in power bus user equipment

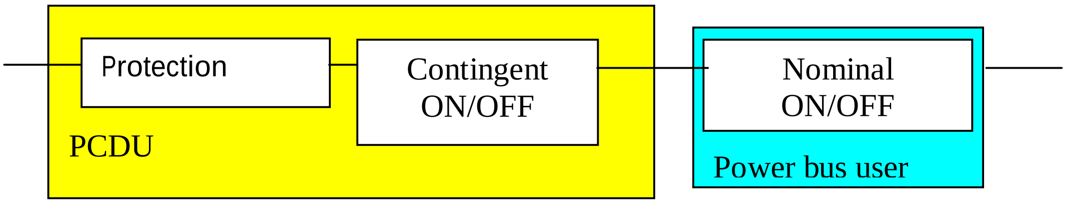


Figure 7-2 ON/OFF switched power line function implementation 2

Protection commutation implemented within the PCDU – Nominal ON/OFF & Contingent functions implemented in power bus user equipment

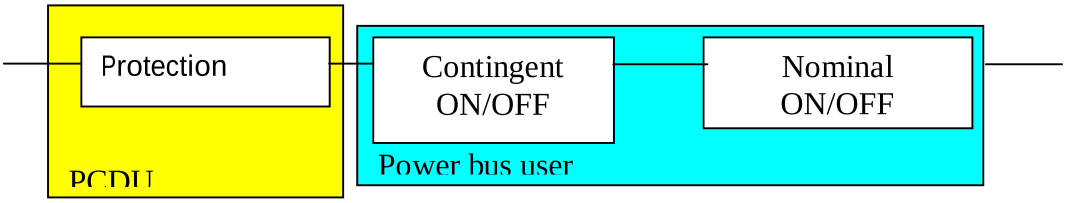


Figure 7-3 ON/OFF switched power line function implementation 3

7.1.1.1 On permanent power lines

As described in Figure 7-4, only protection function shall be implemented.

Figure 7-4: On permanent power line function

The protection function implementation will be done within PCDU.

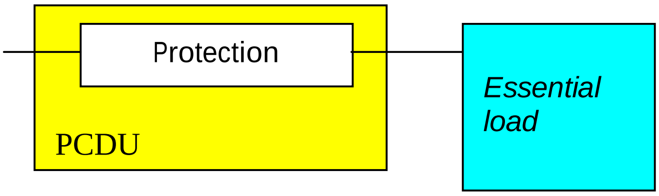


Figure 7-5 On permanent line function implementation 4

7.1.2 Trade-off conclusion

Possible architectures have been described for each implementation. Trade-off has been performed between fuse and LCL on following aspects:

- Power impact due to voltage drop onto the distribution line
- mass of the protection
- EMC (voltage transient , current conducted emissions, voltage conducted susceptibility)
- FDIR (failure propagation, outage recovery)
- Reliability of the protection
- Mechanical environment
- AIT constraints
- Heritage
- Cost

The conclusion of the trade-off is:
as far as cost aspects are concerned a solution with fuses is slightly less expensive than and LCL based power distribution.
as far as technical aspects are concerned, the power distribution using LCL is advantageous. In particular in critical mission phases where it is not acceptable to have a voltage drop of the power bus.

7.2 Harness mass reduction

One way to reduce the harness mass at satellite level is to promote the use of serial bus to distribute and acquire data on the platform. The further step is to replace the copper based link by a wireless link.

Concerning the low rate bus which is the most often used for acquisition/command exchanges, the trade-off has already been performed between CAN, 1553, ODH485 and TTP/C (see RD 02). 1553 and CAN busses have been selected : the first one is the current standard bus and allows to reuse existing units (AOCS units in particular) and the second one is the future standard and will allow to connect new units embedding CAN bus coupler.

In order to facilitate the transfer from 1553 to CAN, it is essential to standardize the software layer in charge to manage the user interface protocol through the CAN and 1553 busses. This layer called convergence layer will allow to migrate from 1553 to CAN bus with a minimum impact on the application software definition. This point is addressed in the following section.

An estimation of the spacecraft harness mass has been established for a typical Aurora mission (the chosen target is the Exomars orbiter). Following table provides the mass breakdown of the harness by signal type.

This table shows that the distribution of the harness mass versus the type of signal doesn’t reveal a unique way allowing to dramatically cut down the overall mass of the harness. Parallel effort should be done to decrease the linear mass of heavier cable, limiting the length of the wires by implementing remote terminals closer from the units to acquire or command and to replace cable by wireless links. The replacement of impedance adapted wire (120 Ohms) for short link would be analysed and can lead to a significant reduction of the overall harness mass.

Signal type	RPW	PWR	SDPW	ELC	HLC	LLC	RS	ANA
Number of lines	6	60	34	10	202	6	86	89
Length in meter	250	180	110	30	700	20	300	300
Wire type	18 Al	20 Al	26	26	28	28 sh	28	28 sh
Wire mass g/m	8,74	3,85	3,73	3,73	2,35	7,62	2,35	7,62
Mass per signal	2185	693	410,3	111,9	1645	152,4	705	2286

Signal type	1553	422	SBDL	EED	Safety	TH	HT
Number of lines	20	44	44	88	6	290	108
Length in meter	60	150	150	400	30	1000	400
Wire type	77 Ohms	120 Ohms	120 Ohms	26 sh	26 sh	28	26
Wire mass g/m	21	18	18	9,24	9,24	2,35	3,73
Mass per signal	1260	2700	2700	3696	277,2	2350	1492

TOTAL MASS	**8188,6**

RPW	Solar array and Power Bus (Raw PoWer)
PWR	PoWeR distribution
SDPW	SecunDary PoWer Distribution
HLC	High Level Command)
LLC	Low Level Command
RS	Relay Status
ANA	ANAlogue (or sensitive) signal
1553	1553 Long Stub interconnection
422	RS422 line
SBDL	Single Balanced Data Link
Safety	safety
Th	Thermistor
HT	HeaTer for thermal control

Additional reduction of harness mass could be obtained thanks to the use of wireless links, as optical (data transmission by infrared light modulation) or RF (Bluetooth or wifi standards). The proposed approach is to implement this new technology through an incremental process: the first step is a validation in flight condition of a such link by implementing a non critical link, the second step would be a implementation of several links in different configuration allowing the last step which could be a generalisation of wireless links. For the first step, the most efficient way which limits the risk on the mission is to implement a redundant connection with a wireless link for the nominal one and a classical link for the redundant.

Candidate for this experimental link could be the bus between the OBC and the payload. A more attractive implementation could be the jettisonnable link between two modules. The in flight demonstration of the capability of wireless link to remove the need of separable connector with associated complex mechanism would provide a real technological progress. A such implementation for a RF link can be done by a transceiver / antenna implemented on the orbiter on one side and the symmetrical unit on the separable module on the other side. The antenna shall be placed on the external surface of the structure or internally in front of a specific hole. As long as the modules are attached the proximity of both antennas allows a correct data transmission and after separation the transmitter is not yet used but doesn't present a risk of short circuit on the data transmission line as separable connector.

7.3 Conclusion

This study has allowed to define a set of rules applicable to the layout of a platform to be developed in the frame of Aurora projects.

These rules have been split in following areas:
Power distribution and localisation of power protection and commutation devices
Power grounding concept applicable to a jettisionable module
Harness definition and layout
Use of remote terminals
EMC design constraints

They would be implemented in the General Design and Interface Requirement (GDIR) applicable to future missions.

8. UML FUNCTIONAL MODEL

This activity has produced a functional model of Aurora avionics using the Unified Modelling Language (UML).

The goal of this activity is to develop a model compliant with the User Requirement Document using a UML software language to specify an Object Oriented software environment for AURORA project.

The used tool is I-Logix UML Rhapsody.

The main characteristics of this model is:

- to provide a high level of abstraction and a language implementation independence
- to ease of additional development and to strengthen communicability of model between all engineers involved in the development and validation process thanks to the strong use of functional and architecture diagrams
- modularity of architectural subsystems and genericity of common components allowing their reusability through the definition of a module library
- animation capability for a better comprehension of the behaviour of the avionics and to allow performing test scenario for conformity verification of the initial requirements

8.1 UML Aurora model presentation

The overall avionics has been split in 14 subsystems interconnected through functional ports (see Figure 8-1):

- Ground - Spacecraft communication
- Spacecraft to spacecraft communication (used for communication between the Orbiter and other modules)
- Spacecraft mode management
- Failure Detection Isolation and Recovery
- Context management
- Data storage
- On Board Time
- Equipment management
- Guidance & Navigation Control
- Thermal control
- Electrical Power Subsystem
- Release, Deployment and Pyrotechnic (RDP) activation
- Sleep and Wake-up function
- Payload management

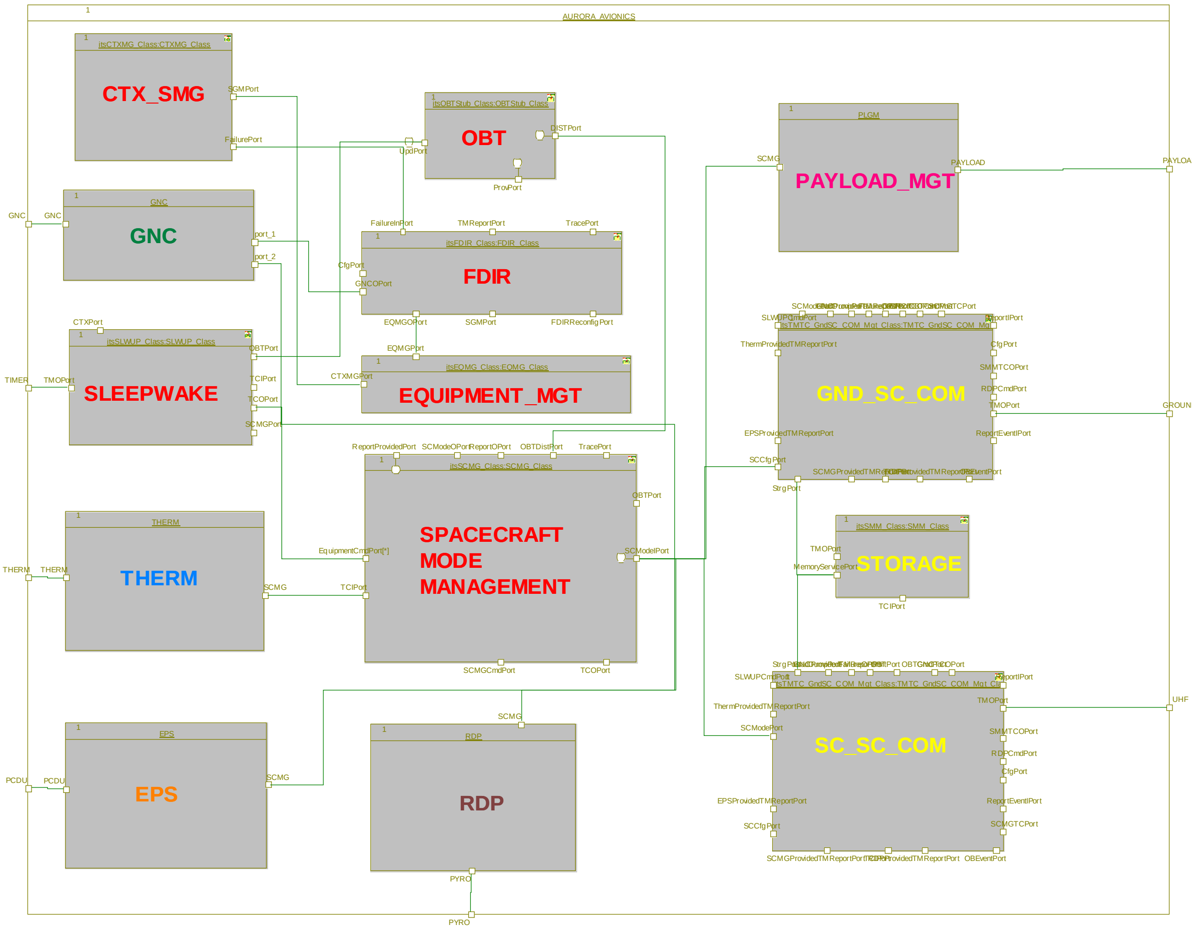


Figure 8-1 UML functional model

Each subsystem is broken down into a modular sub-model as presented in Figure 8-2. The functionality of each sub-model is modelled with the aid of diagrams. Object model diagram are used for defining relations amongst objects.

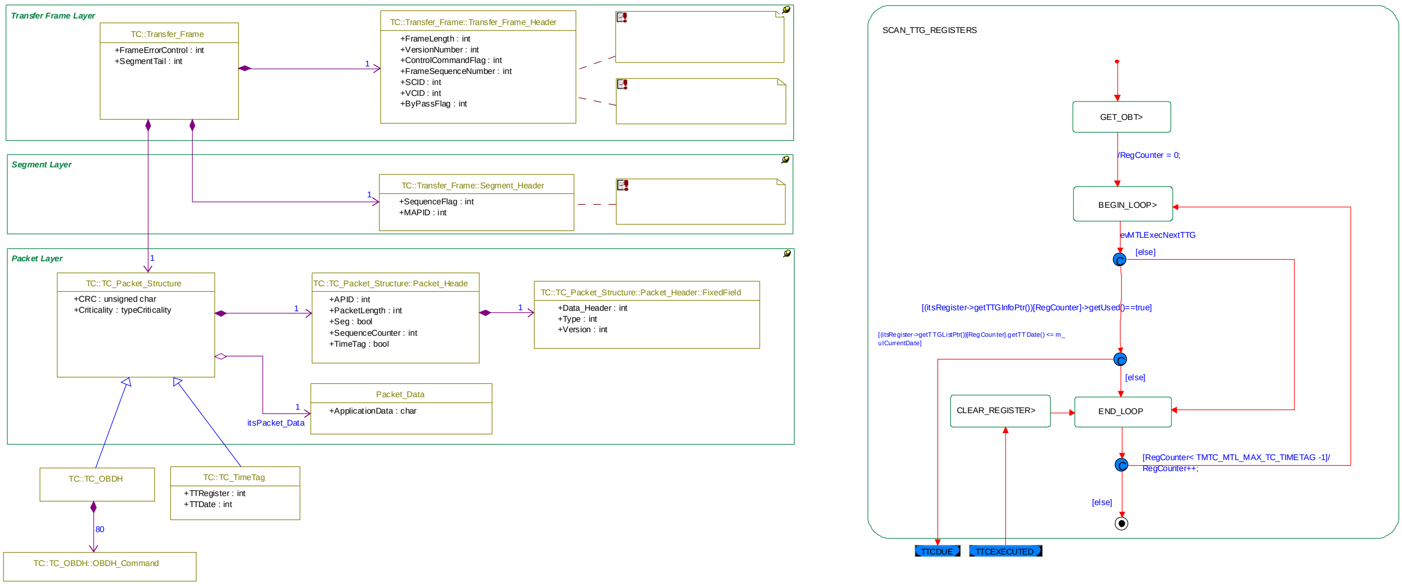


Figure 8-2 Example of a sub-model

8.2 Model performances

As the model is based on the User Requirement Document (URD), the traceability is maintained by linking design components with the URD requirements. As the requirements are numbered in the URD, this implementation allows to build a coverage matrix of the model versus the requirements.

The Rhapsody tool permits model execution with its observability traced into UML sequence diagrams.

Engineers who have provided requirements define a test scenario (input script) which is executed by the model are the generated outputs can be checked through visual diagrams by the engineer in charge of the validation. The execution of these test scenario allows an early validation of the test sequence which will be executed during validation phase on the real hardware.

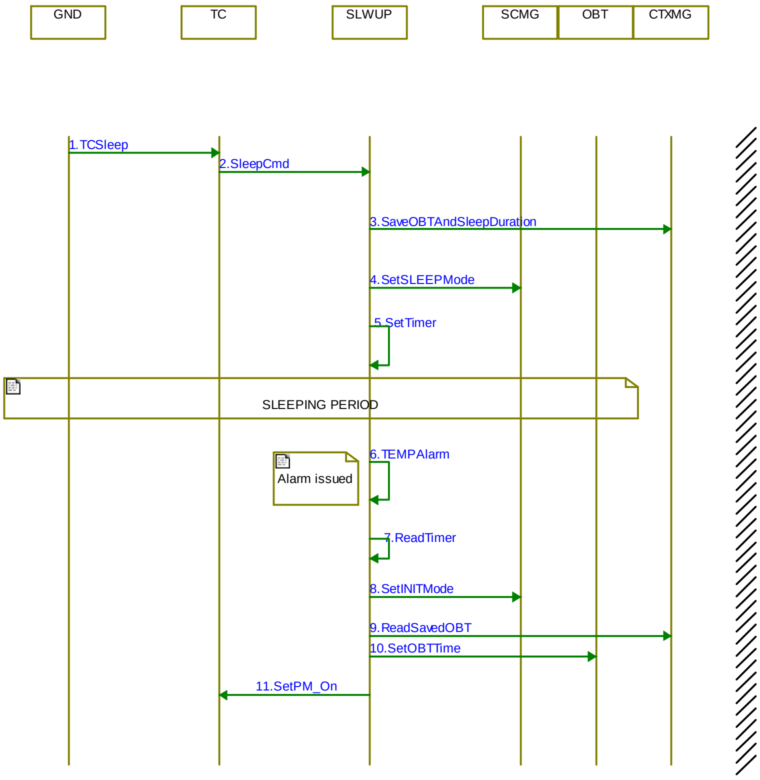


Figure 8-3 Example of a sequence diagram

8.3 Model delivery

The deliverable model is composed of different modules. Each module delivery is composed of:

- A model library in TLB file format (readable by Rhapsody)
- A model report which provides the description of the module (input/ output and functional description)
- A test report which provides the behavior of the module for a test scenario.

The two last reports are textual documents.

9. BUILDING BLOCKS AND TOOL-BOX

9.1 Avionics Tool-Box Definition

The activity has been started by a discussion about the tool-box concept, to get a common understanding and to set up the basis for a workable and effective implementation.
The expectations of the various involved industrial entities, as well as scenario in which the Tool-Box shall be exploited, both in terms of technology trend and geo-political constraints.

Basically two industrial entities have been identified:

Spacecraft Integrators:
They would like, at one extreme, to see entire equipments as atomic elements, although this is often not possible due to the peculiar characteristics of different missions (especially true in the scientific and exploration missions case).
On the opposite side, the System engineer wants just to relay on well consolidated Interface Standards, to ensure compatibility of different equipment and to ease integration.

Equipment Manufacturers:
They share with Spacecraft Integrators the interest in having consolidated interfaces, in terms of standards, tools and physical elements to be used in the realisation of its products.
Items like IP Cores and Components are favoured, rather than Boards.
In particular, these elements shall deal with implementation of standard services associated to the selected Interface Standards, which represents the Toll-Box back-bone.
Of course, the Equipment Manufacturer is focusing on the maturity and reliability (in terms of design and performances and compliance to the relevant standard) of the available tool box elements.

The above outlined wishes of the major involved industrial entities need to be melted with some programmatic constraints, which can be reasonably expected to occur in the near future.

On one side in fact, the evolution of electronics technologies allows to allocate more and more functions in a continuously decreasing volume, resulting in the possibility to include functions originally allocated to different boxes within a single equipment.
This trend is a reality since many years, resulting in a effective reduction of system masses and costs.

On the other hand, reducing the number of distinct piece of equipment needed to build up the system makes more difficult to satisfy geographical returns.

To find a compromise, we can envisage a way to accommodate in a single housing modules provided by different companies.

This means that macro blocks shall be defined, comprehensive of well identified functions, with a limited amount of well defined interfaces among them. Such building blocks, which

can be also composed by more than one physical modules, shall be such to allow separate development and validation to a large extent.
The key issue is again the definition of interfaces among them.
The above outlined approach is eased by the Avionics Architecture defined in the frame of the study.
We can in fact identify three Macro Blocks, which are:

- The Spacecraft Controller Core
- The Processing and Storage Block
- The Low Rate Peripherals

Based on the above considerations, we can define two level of Building-Blocks / Tool-Box elements:

The first one is founded on **"Atomic"** elements, such as IP Cores and Components, made available to the whole community, with a rather deep involvement of ESA in their promotion and certification

The second aims at specifying **Macro blocks**, which can be also build up by exploiting atomic elements, but are more regarded as industrial products.

9.2 Macro-Blocks

As identified above, the overall Avionics can be regarded as a composition of some elements.

In this context, the objective is to ensure that blocks developed by different companies can be easily assembled together (by an equipment leader) to cope with a specific project programmatic constraints.

Interfaces are the pivot of such strategy; however, according to the increased blocks complexity, also the associated communication protocols shall be defined, not just limiting the standardisation at the physical layer.

Another aspect, to be addressed when pursuing the objective of housing boards from different manufacturers in the same mechanical enclosure, is that of mechanical standardisation.
A review of the experience in the standardisation area matured in the last 20 years by AASI (Laben) has been carried-on.
The recommendation is to take a common reference, to the Eurocard form factor, which is already popular in the European Space Industry, however, not pursuing standardisation to the extreme (as happening in the commercial embedded world), to overcome the typical contrast between standardisation and optimisation.

Attention has been then focused on the specification of Interfaces between the three major Avionics Macro Blocks.

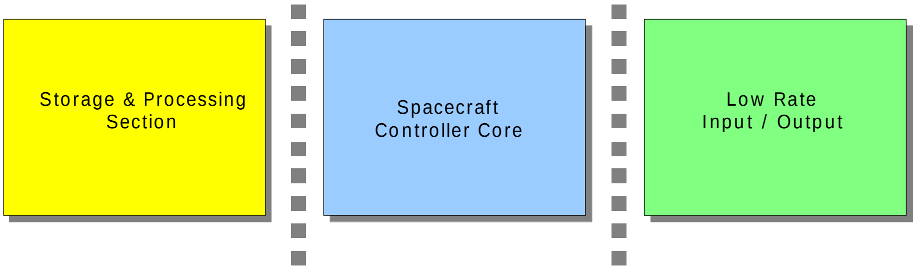


Figure 9-1 High Level Building Blocks and Interfaces

9.2.1 Interfaces between S/CCC and Storage / Processing Modules

The main purpose of these interfaces is to transfer data from Payloads to Mass Memory and from Mass Memory to Telemetry. To achieve this, however, other messages need to be exchanged on the network.

The typical application scenario is depicted in Figure 9-2

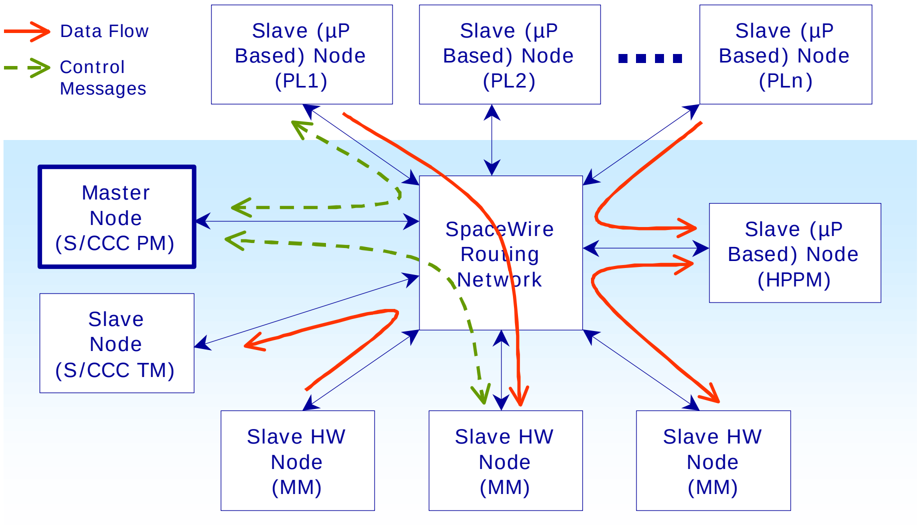


Figure 9-2 Application Scenario

The proposed approach is:
Use of SpaceWire (ECSS-E-50-12A) at Physical Layer
Adoption of a Networking Protocol (SnP)

As SnP, adoption of the **MULTIPLE ACCESS COMMUNICATION PROTOCOL (MACP)** is proposed.
It has been presented at SpaceWire WG Meeting n°3, and it is especially suited for systems with resources (like Mass Memory) accessible by different users.
It is specifically supporting Data Packets transfers directly between slave nodes.

MACP foresees 4 packet types:
Command packet sent by a master node to a slave node

Reply packet answering to a command packet (from slave to master node)
Data packet sent by a source node to a destination node (i.e. any slave or master node)(*)
Interrupt message packet sent by a slave node to a master node (*)
(*) MACP specific, not supported by RMAP

9.2.2 Interfaces between S/CCC and Low Rate Peripheral Modules

Typically, the following tasks shall be supported on this link:

Acquisition of discrete House-keeping monitoring channels of various type (Analog, Thermistors, Pressure transducers, Bi-Level, Relay Status, Serial Digital, …)

Acquisition of Thermistors for thermal Control task

Commanding of Thermal Control Heaters

Distribution of Asynchronous pulsed or serial digital commands for user’s on7off switching or operational mode setting.

Acquisition of Attitude and Orbit Control Sensors data, such as Sun Sensors and Reaction Wheels

Distribution of Commands to drive AOC actuators (mainly RCS Thrusters)

Distribution of other Propulsion system commands, such as Boost Motor and Latch Valves commands

Distribution of commands to drive specific system actuators and acquisition of relevant monitoring channels (e.g. Solar Array and Antenna Position Driving)

As a matter of fact, execution of above tasks is subjected to different constraints, in term of accuracy and phasing to system timing.

Both Synchronous and Asynchronous operations shall be consequently supported.

Use of CAN Bus as Physical Layer is proposed, which offers several advantages, such as:
Multimaster operations (transfer od data from IO as soon as ready)
Bus access prioritised (low individual messages latency)
Fully HW management
Low Cost tools
Intrinsic Bus Monitoring capability

Adoption of a subset of CANOpen is recommended as High level protocol, which is well suitable to cope with above constraints and also already subject of attention within ESA in the last years.

9.3 Atomic Elements Library

As far as the atomic elements library is concerned, these have been identified with the primary objective to support all identified system interfaces aspects.

They can be grouped as follows:

- EXTERNAL SPACECRAFT INTERFACES
    - Communications with Ground
    - Communications among Orbiters and Landers
- ON BOARD SYSTEMS INTERFACES
    - SpaceWire related Atomic Building Blocks
    - MIL-STD-1553B related Atomic Building Blocks
    - CAN Bus related Atomic Building Blocks
- INTERNAL IO MODULES INTERFACES
    - *covered by CAN Bus*
- INTERNAL STORAGE AND PROCESSING INTERFACES
    - *covered by SpaceWire*
- PROCESSING CORES
- COMWARE LAYER

The identified elements are belonging to the following types:

- HARDWARE
    - Components
    - IP Cores
- SOFTWARE
    - BSW SW Drivers
    - ComWare SW Services

For each identified item, a short specification is given, and the current status as well as possible development approach has been identified.

The identified Atomic tool-box elements are here presented in a summary table, ordered by type and status.

HARDWARE ITEMS:

Name	Type	Status	Notes
Single Chip TMTC ASIC (SCTMTC)	Component	Developed in the frame of HICDS project. To be commercialised via Atmel.	Compliance / Applicability of ECSS Standards currently in preparation shall be verified
SPROUT (SpaceWire Router)	Component	Under development. To be commercialised via Atmel.	
AT797E CPU	Component	Under development. To be made available by Atmel.	
MIL-STD-1553B Bus Tranceiver	Component	To be developed and made available from an European supplier.	
CAN Bus Tranceiver	Component	To be developed or made available from an European supplier (e.g. TJA1050T device from Philips)	
Proximity-1 Manager	Component	To be developed.	
Leon2-FT	VHDL IP Core	Available from ESA	FPU not included
PTME (Packet Telemetry Encoder)	VHDL IP Core.	Available - Can be Licensed from ESA.	Compliance / Applicability of ECSS Standards currently in preparation shall be verified
PDEC3AMBA (Packet Telecommand Decoder with AMBA AHB Interface)	VHDL IP Core.	Available - Can be Licensed from ESA (limited to developments that are funded by the European Space Agency)	Compliance / Applicability of ECSS Standards currently in preparation shall be verified
HurriCANe (CAN Controller)	VHDL IP Core	Available - Can be Licensed from ESA	
SPWb (SpaceWire CODEC)	VHDL IP Core	Available - Can be Licensed from ESA.	
SPW-AMBA	VHDL IP Core	Available - Can be Licensed from ESA	Compliance to ECSS-E-50-12A to be verified.
Leon2-FT	VHDL IP Core	Under development. Expected to be available in future from TBD	
SPW-MACP	VHDL IP Core	To be developed	
MIL-STD-1553B BC/RT/BM	VHDL IP Core	To be developed or agreement with Commercial supplier to be settled.	
CANOpen Remote Node	VHDL IP Core	To be developed.	Can be started from running initiatives
CANOpen Network Manager	VHDL IP Core	To be developed.	
Atmel Leon2-FT Hard Macro	Hard Core	Expected to be made available in future from Atmel on ACT18RHA.	Detailed info not available.

SOFTWARE ITEMS:

Name	Type	Status	Notes
SCTMTC Drivers	BSW SW Driver	Existing in HICDS BSW	
Proximity-1 ASIC Drivers	BSW SW Driver	To be developed.	
SpW-AMBA Drivers	BSW SW Driver	To be developed.	Can be derived from HICDS drivers for Laben Core
Router Drivers	BSW SW Driver	To be developed.	
SpW_MACP Drivers	BSW SW Driver	To be developed.	
MIL-1553 BC/RT/BM Drivers	BSW SW Driver	To be developed.	In conjunction with IP Core
CANOpen Remote Node Drivers	BSW SW Driver	To be developed.	In conjunction with IP Core
CANOpen Network Manager Drivers	BSW SW Driver	To be developed.	In conjunction with IP Core
TMTC Adapters	ComWare SW Service	Existing in HICDS SLSW.	
Proximity-1 Adapters	ComWare SW Service	To be developed.	
SpW-MACP Adapters	ComWare SW Service	To be developed.	
MIL-1553 Adapters	ComWare SW Service	To be developed.	Can be derived from HICDS one
CANOpen Adapters	ComWare SW Service	To be developed.	Can be derived from HICDS one
ComWare	ComWare SW Services basic set	HICDS SLSW to be extended	

The Atomic building blocks can be exploited to build-up Avionics elements. Some examples are depicted in the following figures.

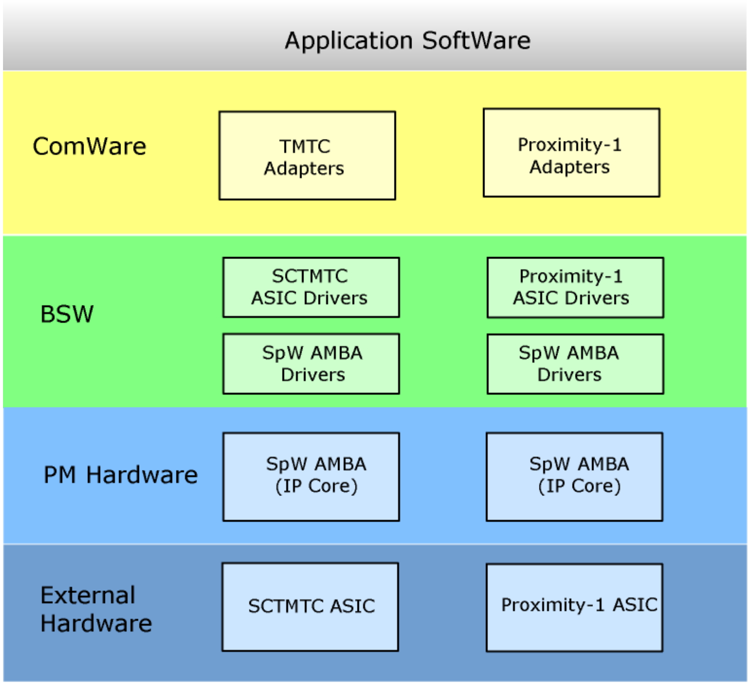


Figure 9-3 Building Blocks and Communication with External entities

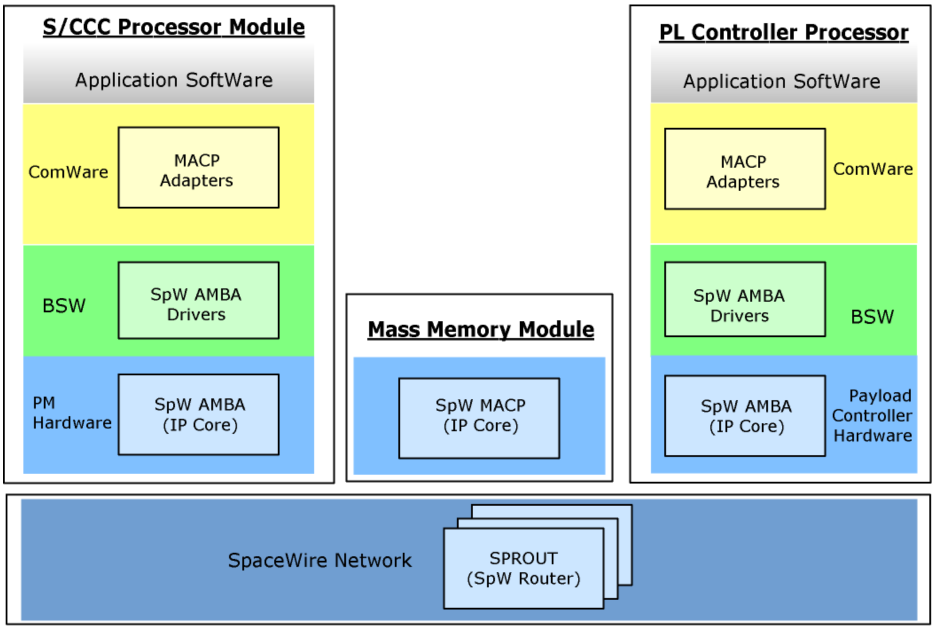


Figure 9-4 Building Blocks and Communication through SpaceWire Network

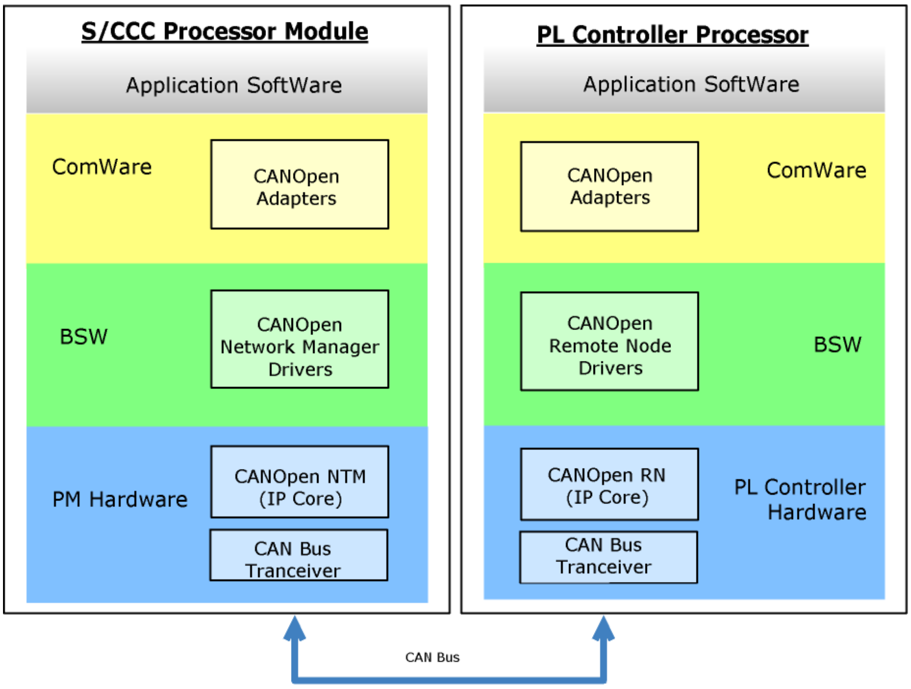


Figure 9-5 Building Blocks and Communication through CAN bus

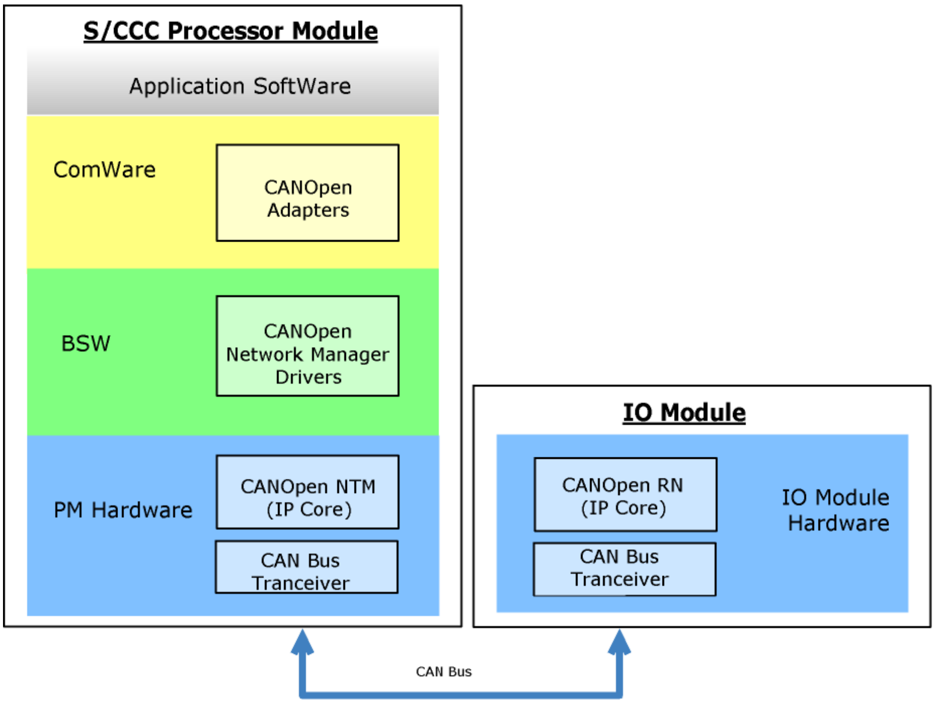


Figure 9-6 Building Blocks and Com through CAN bus with I/O Modules

10. PERIPHERAL MODULES SPECIFICATIONS

Specifications have been defined for the main peripheral modules identified in the frame of the Avionics Architecture definition.
These Modules are:

Mass Memory Module

Proximity-1 Module

Low Rate I/O Modules, covering
    Standard House-Keeping (Analog and Digital)
    Propulsion
    Standard Attitude Control Sensors/Actuators

All the modules are of course fitting in the defined Avionics Architecture in particular providing interfaces to the other Avionics elements as specified in the frame of the Tool-Box activity. As such, they can be developed by exploiting Tool-Box Atomic elements.

10.1 Mass Memory Module

The main characteristics of the Mass Memory Module are summarized in the following Table, while its general block diagram is shown in Fig. XXX.
It is worth to recall that the core of the Module will be an ASIC or FPGA implementing communications with the rest of the Avionics according to SpaceWire and MACP Protocol.

1	Modularity	64Gbit (=2^36 bits)
2	Base memory	Currently 512Mbit (32Mbit x 16 or 64Mbit x 8) SDRAM
3	DRAM packaging technology	3D MCM ("cubes") with 8 512Mbit SDRAM layers
4	Memory Module Configuration	10 independent columns of memory cubes + back up column (cold redundant)
5	Redundancy philosophy	memory stack plus back up column
6	Error correcting code	RS(10,8,8)
7	Error correction capability	1 symbol (1 byte)
8	latch-up protection	At column level
9	Access to Memory stack	*SpaceWire*, under Supervisor remote control
10	I/O interface handling towards other MMs	*SpaceWire* "Dynamic Router"
11	Scrubbing and Refresh	HW, at programmable frequency
12	Management of Internal Redundancy	SW, based on anomalies signalling by HW
13	Power supply voltage	3.3V
14	Power Consumption	2.3W @ 10 MHz SDRAM frequency SpaceWire channels @ 25 Mpbs rate, & MM complete scrubbing within 500 seconds 1.3W @ 10 MHz SDRAM frequency, no store & downlink performed, & MM complete scrubbing within 500 seconds 0.7W @ 10 MHz SDRAM frequency, no store & downlink performed, & MM complete scrubbing within 1000 seconds 3.3W @ 15 MHz SDRAM frequency SpaceWire channels @ 50 Mbps rate, & MM complete scrubbing within 500 second

15	Connectors	SpaceWire: 2 Microminiature 9-pin D-type Connectors (TBC) BackPlane: 1 HE801 connector
16	PCB size	Extended Double Eurocard (233 x 200 mm)
17	Weight	750 gr.

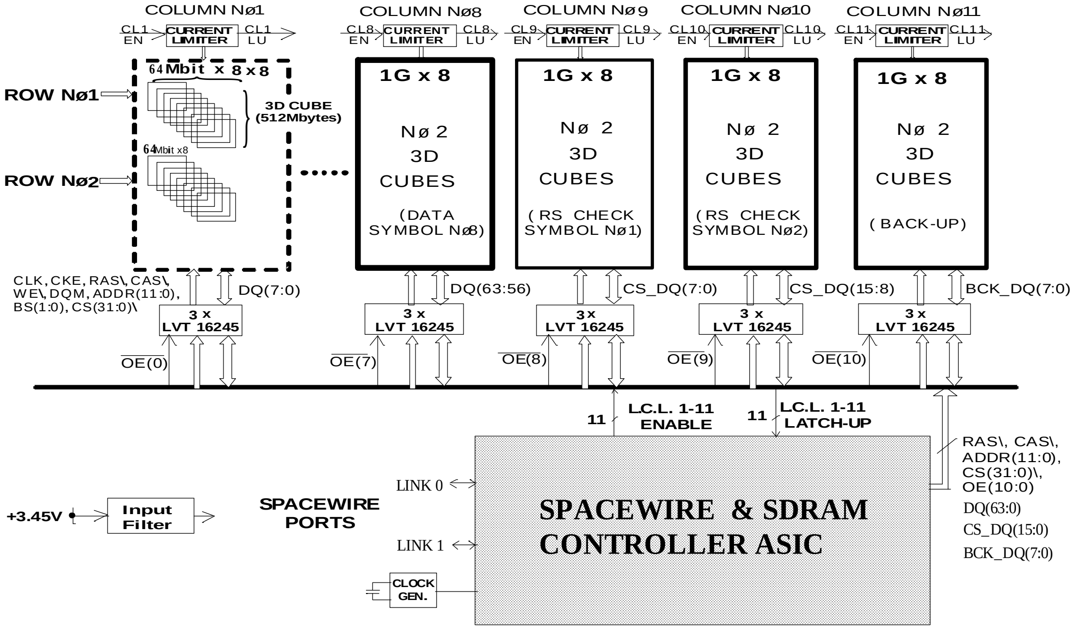


Figure 10-1 Memory Module Architecture

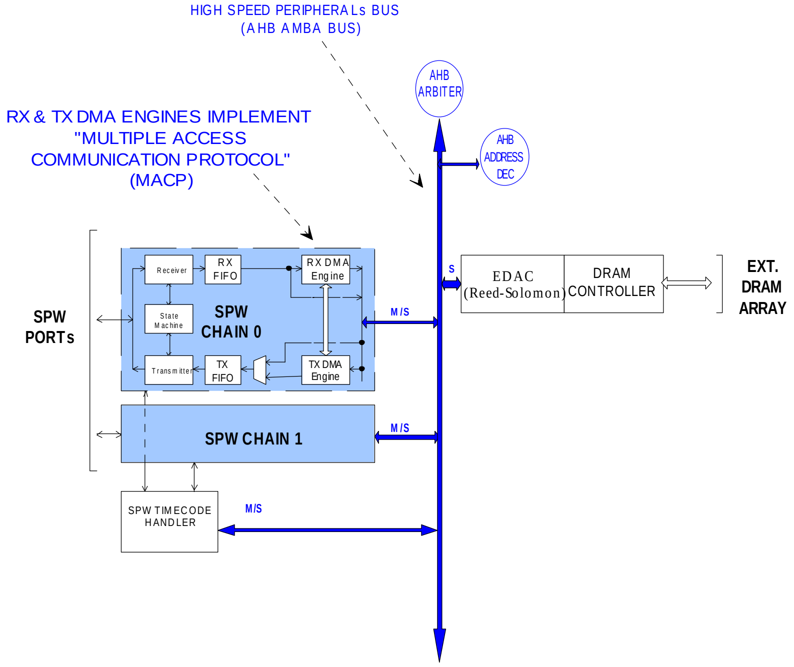


Figure 10-2 SpaceWire and memory controller FPGA

10.2 Proximity-1 Module

The studied module aims at supporting both Orbiter and Lander applications. As such., it includes an Optional CPDU function, which may be exploited, in a lander to generate configuration pulsed commands in case the Processor is not available (a functionality similar to the High priority Commands which are generated on Satellites under ground control).
To module also support different cross-strapping configurations, with RF TxRx section and with Processor Modules, to allow optimised system configuration.
The Proximity-1 Module Block diagram is shown in Figure 10-3

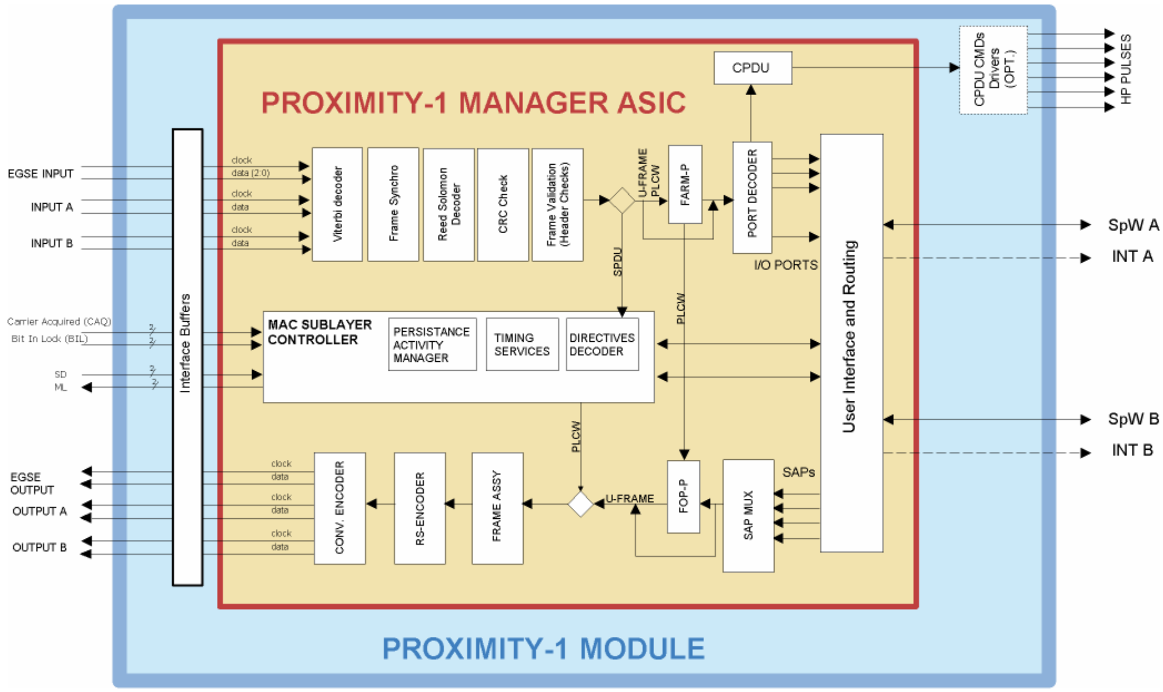


Figure 10-3 Proximity-1 Module

As shown in the picture, the Module is mainly composed by an ASIC. Its functionalities have been analysed in detail, considering the applicable CCSDS Protocol and similarities / differences with respect to the usual packet TLM & Packet TLC protocols in use between Space and Ground have been identified.

Actually, the specified component can be regarded as the corresponding one to the SCTMTC ASIC, devoted to communications among probes, landers, rovers, orbiting constellations, and orbiting relays, as it will be the case for Aurora Missions.

10.3 Low Rate I/O Modules

Four Types of modules have been studied, having the main characteristics and capabilities summarised in the following table.

Module Name	Functions	Redundancy	Size	Q.ty (Typ. System)
Standard Analog HK	120 Thermistors 32 Analog DBE	2 modules, usable either in Hot or Cold redundancy	1 DE Ext	2
Standard Digital HK	4 SD16 Acq. 4 ML16 Cmds 128 Switch Status 32 Bi-Level DBE	2 modules, usable either in Hot or Cold redundancy	1 DE Ext	2
Propulsion Module	16 Thrusters CMDs 1 ABM Commands 5 Latch Valves Commands	2 modules, usable either in Hot or Cold redundancy	1 DE Ext	2
Standard ACE and Motors	1 Coarse Sun Sensor 1 Reaction Wheel 1 SADM 1 APM	4 modules, usable either in Hot or Cold redundancy	½ DE Ext	4

All the Modules communicate with the Spacecraft Controller Core via CAN Bus and implements CANOpen high level protocol in hardware. As such they can be realised by exploiting the relevant Tool-Box Atomic elements.

The Block Diagrams of the four Modules are shown in the following pictures.

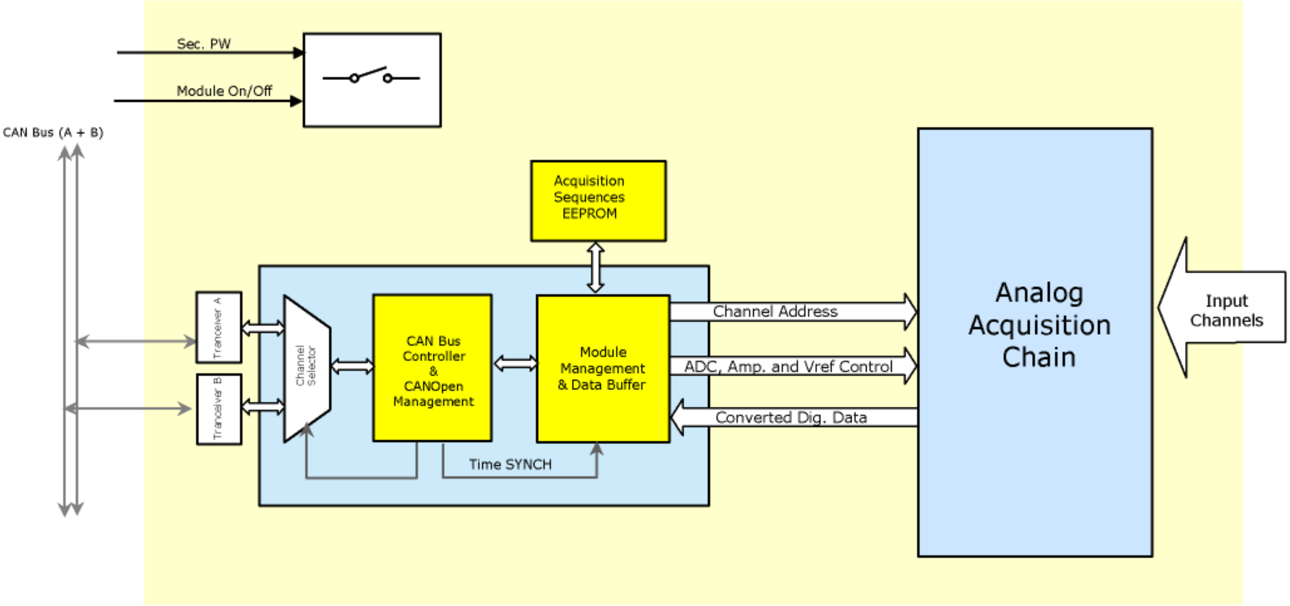


Figure 10-4 Analog HK Module General Block Diagram

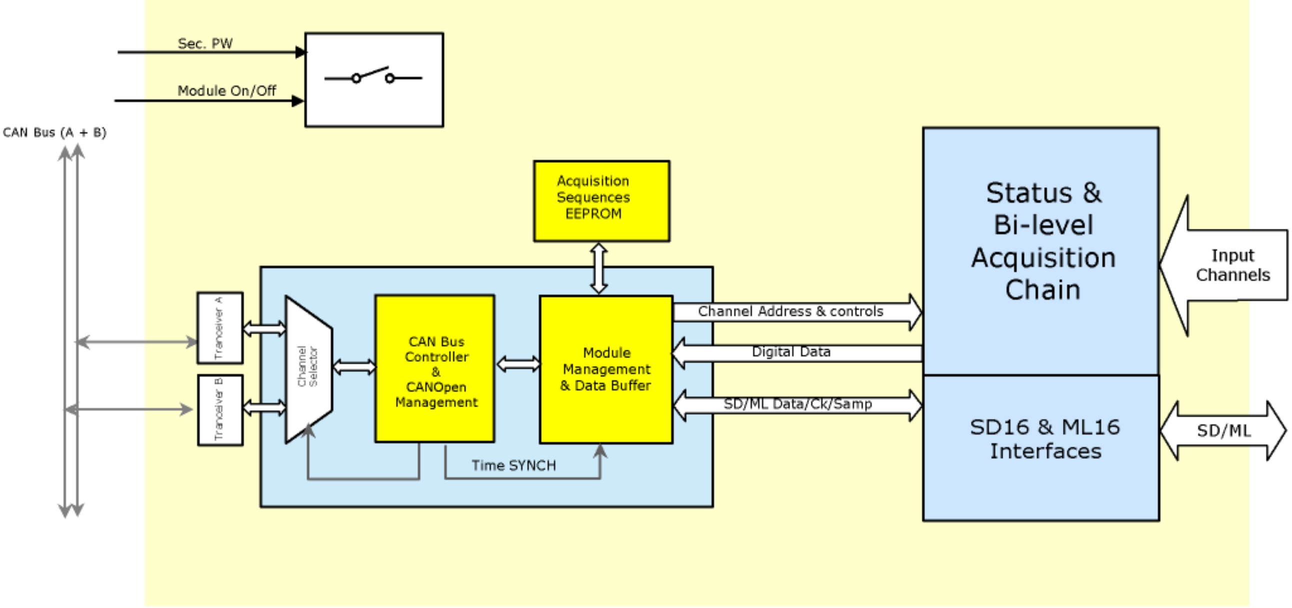


Figure 10-5 Digital HK Module General Block Diagram

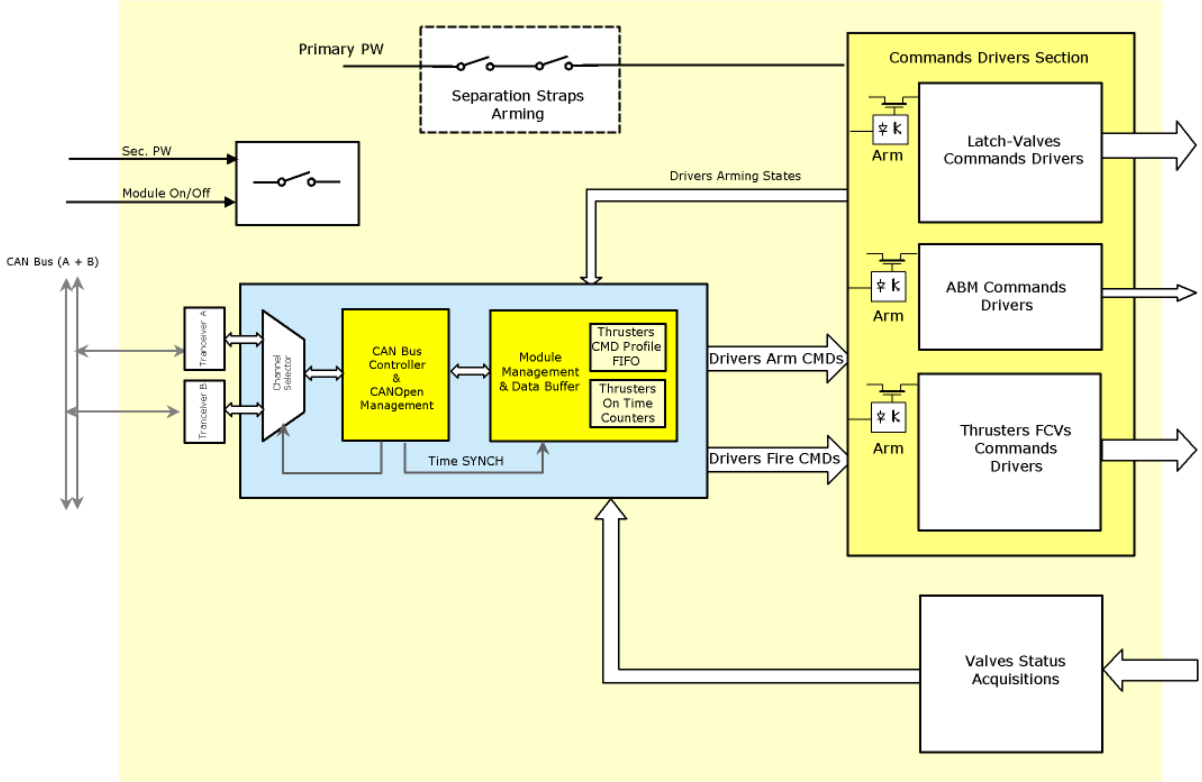


Figure 10-6 Propulsion Module General Block Diagram

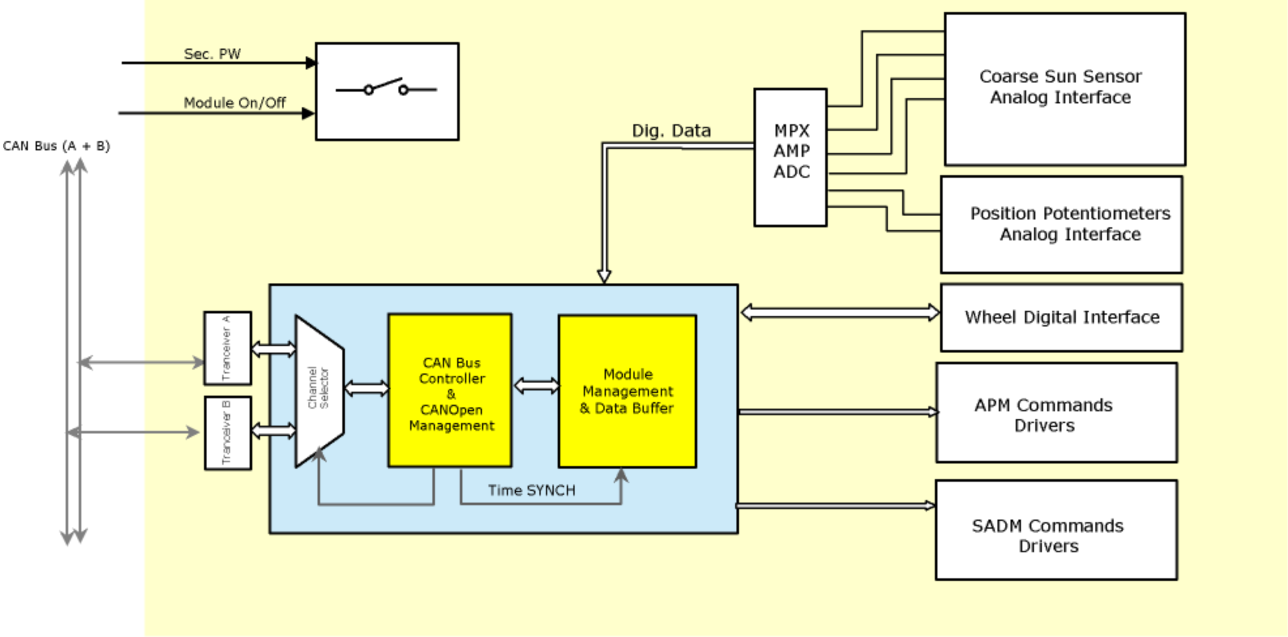


Figure 10-7 Standard ACE Module

11. SOFTWARE SPECIFICATION

Services specifications have been defined starting from HICDS SLSW design activity and using the following input documents (SOIF recommendations):
Spacecraft Onboard Interfaces – Concepts and Rationale Green Book, March 2003;
Intra-network Service Definition Red Book, Sept. 2002;
Intra-network Service Interface Definition, Red Book Apr. 2002;
SOIF C&DA Service – Capability Set 1 – Device Read and Write Red Book, Oct. 2002.
Services organization and mapping is listed in the table below:

SOIF Services			COMWARE Services
C&DA	**CS1**	**Device access**	**Abstract device**
	CS2	**Engineering unit conversion**	**Engineered device**
	CS3	**Data product acquisition**	**Run time device**
	CS4	**Monitoring**	**Monitored device**
	CS5	**Device virtualisation**	***Included in abstract device***
	CS6	**Data pooling**	**Sampled device**
Time Distribution			**Time Distribution**
Configuration Management			**Configuration Management Service**
FDir			**Fault/Status Detection Service**
Communication		**Not Guarandeed, Connectionless Service**	**Unconfirmed Datagram Service**
		Guardandeed, Connectionless Service	**Confirmed Datagram Service**
		Guarandeed, Connection Oriented Service	**Stream Service**
		File Transfer	**File Transfer Service**

Table 11-1 – SOIF/COMWARE services.

Where highlighted items represent the basic capabilities already implemented in the framework of HICDS and used by others as basic services.

The main features of COMWARE have been identified in terms of:

- Service Interface:
  - API is independent from device configuration and peripherals localization;
  - API has a hierarchical structure following the Service hierarchy;
  - API has been defined for ADA95 and C languages;
- Configurability:
  - Network and abstract device databases are used to define network layout and device virtualisation;
  - Configuration management service allows to switch between predefined network configurations;
- Scalability:
  - High level services extend lower level service functionality;
  - Sub-network adapters can be "plugged-in" to interface new or different communication/control buses.

Specific needs for AURORA missions have also been taken into account including in the specification Mass Memory management via SpaceWire, Proximity-1 link for lander-orbiter communication and CANOpen compliant I/O Modules accessed through CAN Bus.

This following model have been used to define service specification.

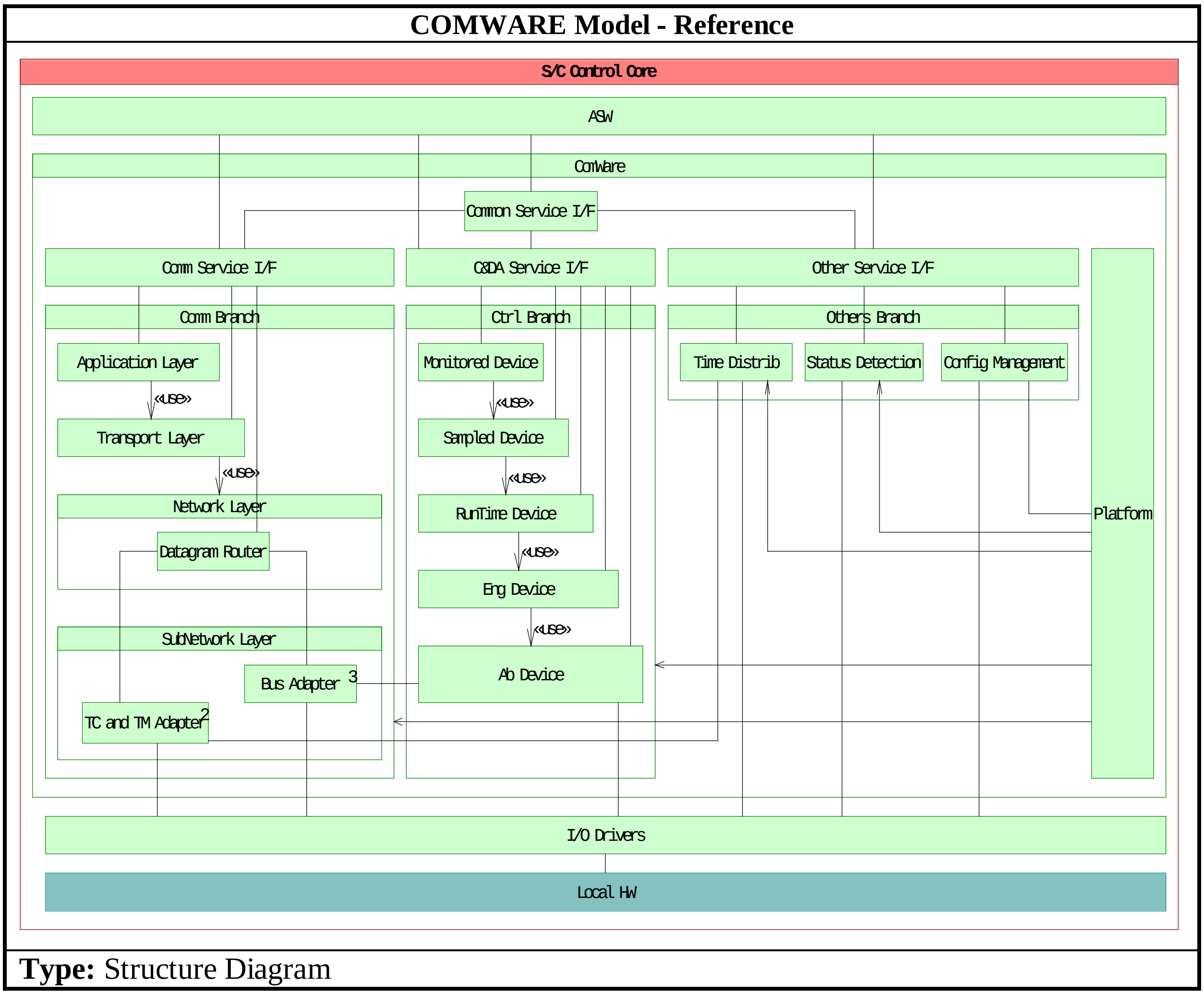
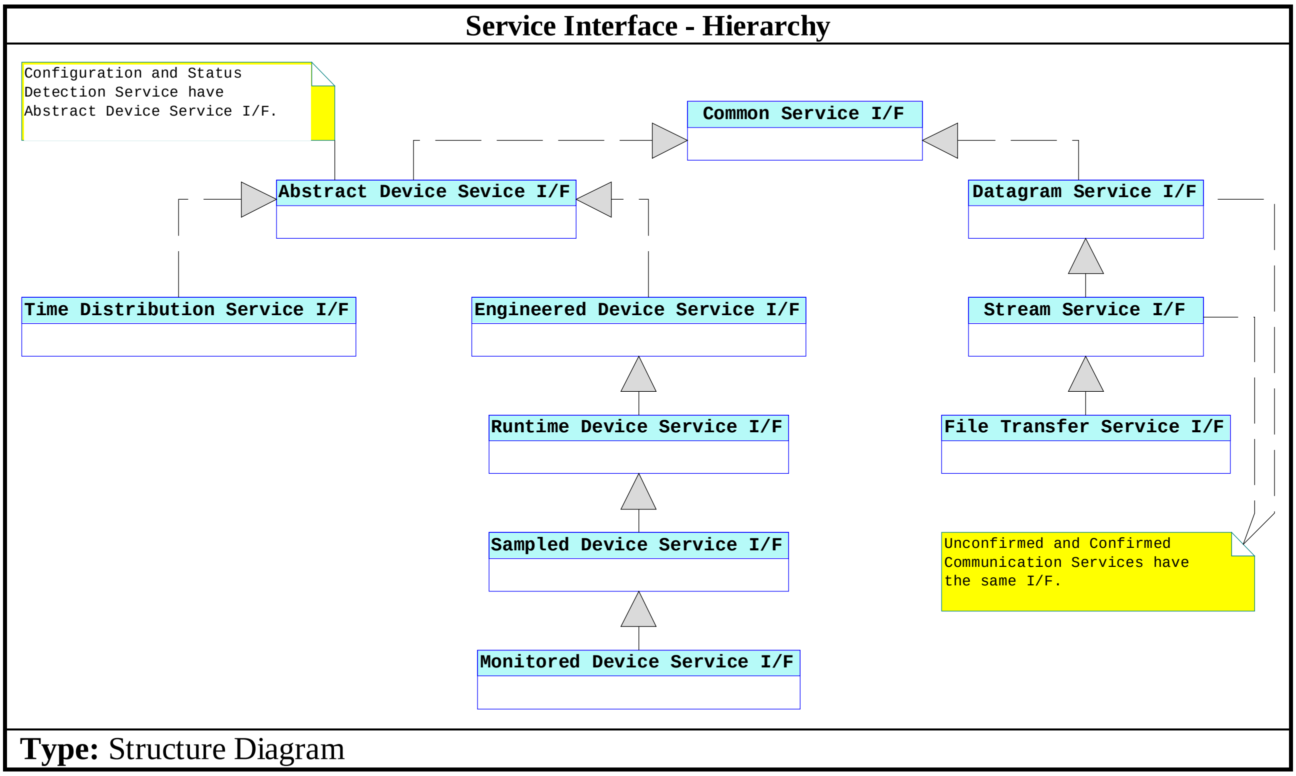


Figure 11-1 COMWARE Model.

Service I/F modules contain the provided interfaces to ASW for Communication Service, Command and Data Acquisition Service and Other Services. Service I/F structure (see the diagram below) reflects the SOIF hierarchal capabilities.



Configuration and Status Detection Service have Abstract Device Service I/F.

Unconfirmed and Confirmed Communication Services have the same I/F.

Figure 11-2 Service Interface - Hierarchy

Control Branch contains the internal implementation of the abstract device model used to provide control and data acquisition services.
Communication Branch contains the internal implementation of the networking stack to provide communication services and of the sub-network layer, also used for control and data acquisition when bus access is required.
Sub-network layer is composed by a set of Adapters in charge to adapter upper layers protocols to media dependent protocols.

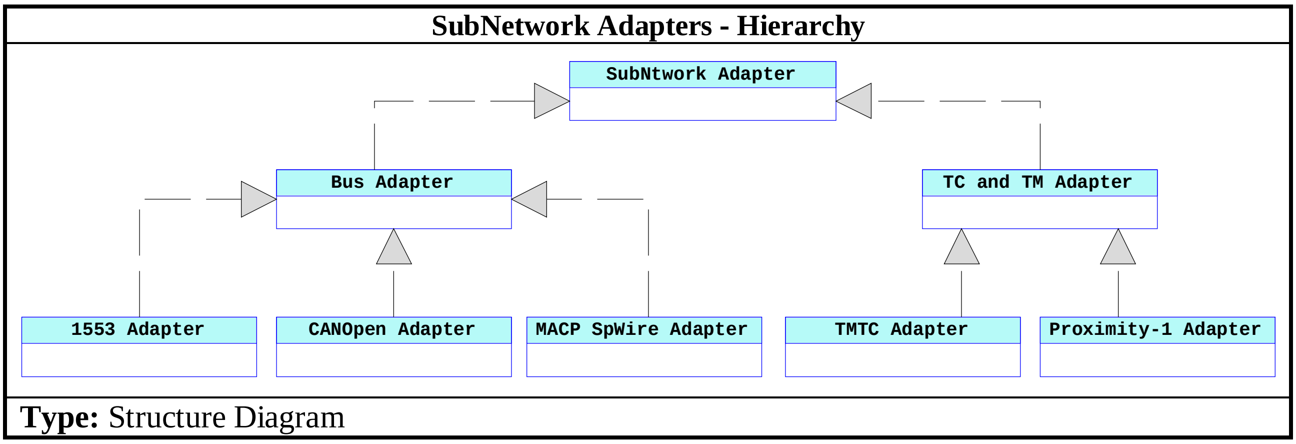


Figure 11-3 SubNetwork Adapters - Hierarchy (Structure Diagram)

Other Service Branch contains the internal implementation of OBT distribution, Configuration Management, Status Detection services.

Platform contains the onboard databases used by COMWARE to configure all the services (network and resources database).

Future releases the COMWARE shall also take into account of the recent new SOIS standard definitions, not available at the time of the AURORA AVIONICS study.

12. CONCLUSION

The Aurora avionics architecture definition study has allowed to provide following improvements reusable for future Martian missions:

A definition of a modular architecture has been established. This architecture has been upgraded in the frame of a Alcatel study (internal funding) including the development of a software moke-up based on the definition of this avionics. This new architecture will become the generic design proposed by Alcatel Alenia Space for future scientific missions: Aurora (Exomars, MSR), Bepi Colombo, Gaia….

A global UML functional model is available and constitutes the first step of a reusable library of functional models.

This study has given the opportunity to **develop a software layer called framework** based on the conclusions of this study. This development will be done by Spacebel in the frame of an independent contract between ESA and Spacebel “Aurora avionics data system standard services architecture”.

Specifications for different key elements are available:
- peripherical modules
- middleware software

These specifications can be used as input for a future development of these modules.

END OF DOCUMENT