

Final Presentation

AAML STUDY

AVIONICS ARCHITECTURE MODELLING LANGUAGE

2014, May 22th

© GMV; Property of GMV All rights reserved





CONTENTS

- 1. Introduction and objectives.
- 2. Study activities.
 - Activity 1. Specification of the avionics-relevant analyses.
 - Activity 2. Specification of the modelling language features.
 - Activity 3. Demonstration and prototyping.
- 3. Prototype demonstration Use case.
- 4. Conclusions and future work.



© GMV

AAML - Final Presentation

INTRODUCTION AND OBJECTIVES



INTRODUCTION

- The ESA AAML (Avionics Architecture Modelling Language) study aims at advancing the avionics engineering practices towards a model-based approach.
- □ Consortium led by GMV:



- GMV's tasks: specification of the analyses (algorithms), identification of specialized modelling entities, implementation of the modelling language, graphical editor, modelling environment, analysis tools and use case.
- Thales's tasks: selection of use cases, identification of relevant analyses (input, outputs and tasks in each development phase), identification of modelling entities and design views.
- Project Kick-Off Meeting on February 2013.
- Project Acceptance Review on April 2014.





SCOPE AND BACKGROUND (1/2)

- Defining an avionic architecture for a given project means making several key architecture choices and sizing several performance parameters.
- □ The selection is usually based on:
 - The architect's expertise and background.
 - Avionics-specific analyses (to perform trade-offs).

Traditional process:

- Each type of analysis is based on a dedicated model (sometimes ad-hoc to solve a particular problem).
- Some training is required to be used effectively.

AAML model-based approach:

- Usage of a single architectural model.
- The same input (i.e., single source model) is used to perform different avionics analyses.
- The analyses cover most of the phases of the life-cycle.

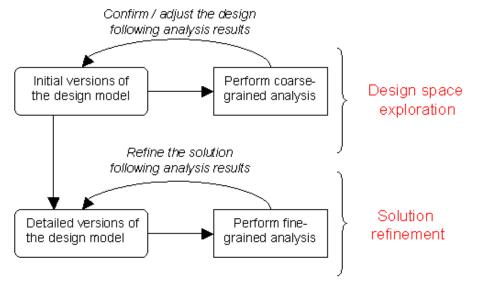


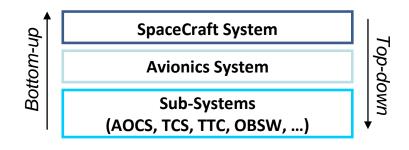


SCOPE AND BACKGROUND (2/2)

Advantages:

- Improves the consistency.
- It allows architects to trace their decisions during the model refinement iterations.
- Facilitates the reuse (e.g., instantiation off-the-shelf components).
- Contribution to the objectives of the SAVOIR initiative:
 - Opportunity to bridge (or make closer) the S/C, avionics and SW paradigms.
 - It provides further results to be considered in the definition of the On-Board Software Reference Architecture (OSRA).







© GMV



STUDY OBJECTIVES

□ The objectives of this study are threefold:

1. Specification of avionics-relevant analyses.

The identification and detailed specification of the analyses relevant to support the avionics architectural design.

It includes the prioritization of the various analyses with respect to their added value to the architects.

2. Specification of modelling language features.

The specification of the modelling language features necessary to support the identified avionics-relevant analyses.

3. Demonstration and prototyping.

The prototyping of software tooling to demonstrate the automation of the selected analyses based on a modelling language that complies to the specification defined within this study.



AAML - Final Presentation

STUDY ACTIVITIES





STUDY ACTIVITIES

- ☐ Three main activities:
 - Activity 1. Specification of the avionics-relevant analyses.
 - Activity 2. Specification of the modelling language features.
 - Activity 3. Demonstration and prototyping.





ACTIVITY 1 – EVALUATION OF AVIONICS-RELEVANT ANALYSES

- □ Several **Use Cases** were selected from real missions:
 - Goal: Identify the relevant avionics analyses and their priorities.
 - Missions: Sentinel 3, GöKTürk, Herschel/Planck, MTG and ExoMars.
- Result: A catalogue and a complete description of all the analyses of interest for the avionics design in the different phases of development, including their inputs and expected results.

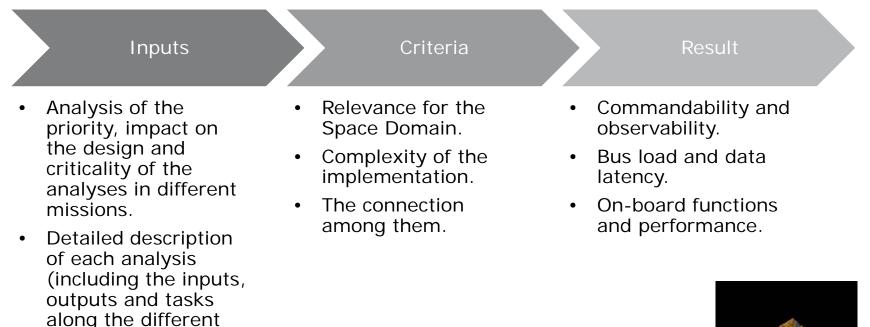
Analysis	Priority	Complexity	Weight on Design
Satellite mode definition, RAMS, FDIR and autonomy concept	HIGH	HIGH	HIGH
Design consistency and correctness checks	HIGH	LOW	HIGH
Commandability and observability	MEDIUM → HIGH	MEDIUM	LOW
Bus load and data latency	LOW → HIGH	LOW → HIGH	MEDIUM
Space/ground communication	LOW → HIGH	HIGH	LOW → MEDIUM
Avionics resources	MEDIUM	LOW	MEDIUM
On-board functions and performance	MEDIUM	MEDIUM	MEDIUM
Power and mass	MEDIUM	LOW	MEDIUM





ACTIVITY 1 – SELECTION OF AVIONICS-RELEVANT ANALYSES

□ Selection of 3 analyses:





ThalesAlenia

□ Selection of the use case: Sentinel 3 mission.

phases of the development).



ACTIVITY 1 – COMMANDABILITY AND OBSERVABILITY ANALYSIS

- **Goal**: Size the RF communication system.
- **Metrics**: Data throughput [bps], link occupation [%], link occupation margin.
 - Coarse-grained analysis (Phases: 0 to B). Few information on the equipment needs in terms of TM/TC is available. Estimations are made based on the information defined and on the knowledge of similar missions.
 - Fine-grained analysis (Phases: C to F). It is based on accurate real data at equipment level.





ACTIVITY 1 – BUS LOAD AND DATA LATENCY ANALYSIS (1/2)

Bus load and data latency: RS-422/RS-232/RS-485.

- Goal: Size the RS-422/RS-232/RS-485 serial links.
- Metrics: Data latency [ms], message transmission time [ms], bus load [%], bus margin, bus utilization.
 - Coarse-grained analysis (Phases: 0 to B). The message size is based on estimations.
 - Fine-grained analysis (Phases: C to D). Accurate real inputs are used.





ACTIVITY 1 – BUS LOAD AND DATA LATENCY ANALYSIS (2/2)

Bus load and data latency: MIL-STD-1553.

Goal: Size the MIL-STD-1553 bus.

Note. Only BC-RT and RT-BC patterns are considered.

- MIL-STD-1553 schedulability analysis.
- Coarse-grained analysis (Phases: 0 to B). Message size is based on estimations and the bus occupation profile is not defined.
 - Metrics: Data latency [ms], message transmission time [ms], bus load Ο per minor frame [%], bus load per major frame [%].
- Fine-grained analysis (Phases: C to D). All the inputs required at equipment level are available.
 - Metrics: Data latency [ms], message transmission time [ms], bus load [%], bus margin [%], bus utilization [%].







ACTIVITY 1 – ON-BOARD FUNCTIONS AND PERFORMANCE ANALYSIS

- **Goal:** Analyse the CPU load and memory sizing.
 - **CPU Metrics**: CPU usage [%], CPU throughput [MIPS], CPU usage margin [%].
 - Coarse-grained analysis (Phases: 0 to B). WCET cannot be measured. It is computed based on estimations on the size of the code.
 - Fine-grained analysis (Phases: C to F) based on the WCET already available.
 - **Memory Metrics**: Non-volatile/volatile memory size [MB], non-volatile/volatile memory margins[%].
 - Coarse-grained analysis (Phases: 0 to B). A default stack size is 0 assigned.
 - Fine-grained analysis (Phases: C to F). The stack size of each function is computed.





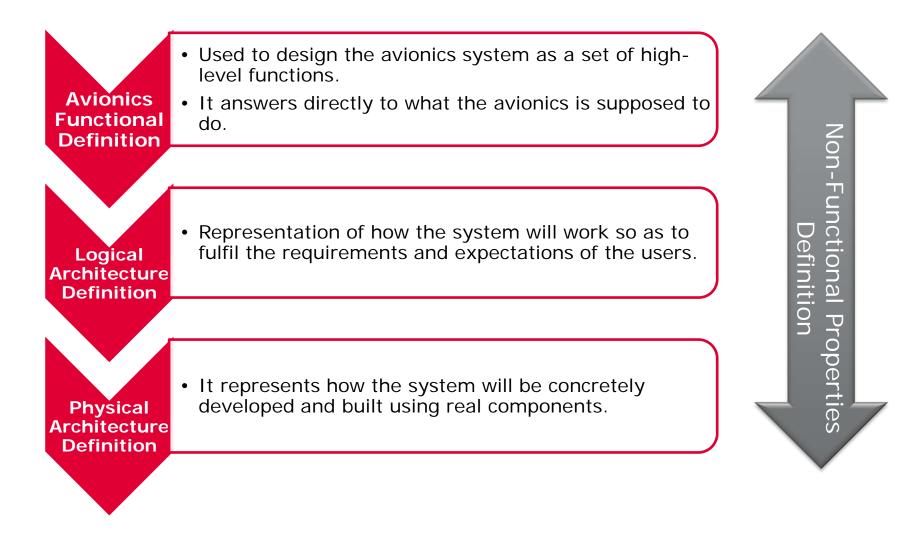
STUDY ACTIVITIES

- □ Three main activities:
 - Activity 1. Specification of the avionics-relevant analyses.
 - Activity 2. Specification of the modelling language features.
 - Activity 3. Demonstration and prototyping.





ACTIVITY 2 – AAML MODELLING PROCESS







ACTIVITY 2 – AAML MODELLING LANGUAGE

Definition of the **AAML Modelling Language**.

- Entities: general entities, data model, avionic entities (e.g., avionic function, logical component, physical components), communication entities (e.g., operation), interaction entities (e.g., interface port), state modelling.
- Non-functional properties to describe the entities in the various non-functional dimensions of interest

Elaboration of Language Extension Mechanisms.

- AAML is a domain specific language (DSL).
- Evaluation:
 - Approaches: (i) Usage of profiles or (ii) the implementation of a new meta-model based on some metamodel language.
 - Modelling languages: AADL, UML, SysML, MARTE, CHESS-ML (CHESS project) and Space Component Model (adopted and extended in COrDeT activities for OSRA).
- Selection:

AAML Final Presentation

- SCM was selected as the inspiration for the definition of the AAML implementation.
- Implementation:
 - AAML modelling language is split into two parts: *AAML_Core* and *AAML_Extension*.





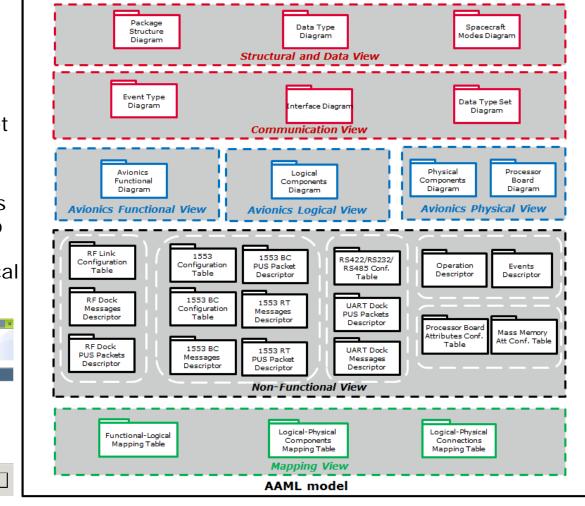


ACTIVITY 2 – AAML DESIGN VIEWS

Definition of the AAML Design Views.

- Representation of a the whole system from the perspective of a related set of concerns.
- In AAML, there are specialized representations of the system according to the avionics design phase (Avionics Functional, Logical and Physical Views).

Change viewpoints selection status (see tooltip for details about each





Viewpoints Selection

Selected viewpoints

🗹 🎂 🛛 Avionics Functional View

Mapping View

Avionics Logical View

Avionics Physical View

Communication View

Non-Functional View

Structural and Data View

viewpoint)

 \checkmark

M 🚯

?

OK

Cancel

© GMV



STUDY ACTIVITIES

- □ Three main activities:
 - Activity 1. Specification of the avionics-relevant analyses.
 - Activity 2. Specification of the modelling language features.

Activity 3. Demonstration and prototyping.





ACTIVITY 3 – DEMONSTRATION AND PROTOTYPING (1/2)

Goal:

- Implementation of a prototype demonstrator for a graphical editor (design views) and analyses tools.
- □ Technology:
 - Developed as a set of <u>Eclipse</u> plug-ins that configure a design and analysis environment integrated into the Eclipse platform.
 - <u>Obeo Designer</u> is used for the graphical editor.
- Capabilities:
 - Creation/modification of an AAML model through the graphical editor.
 - o Usage of different Design Views.
 - Edition by means of different kinds of diagrams/tables.
 - Configuration of the avionics analyses from a GUI based on Eclipse wizards.
 - Execution of the avionics analyses.
 - Identification of model inconsistences.





ACTIVITY 3 – DEMONSTRATION AND PROTOTYPING (2/2)

□ Analysis outputs:

- A file containing the <u>analysis results</u>.
- A file containing <u>debug information</u>, where the user may examine which model elements/properties were used for computing the analysis results.
- A set of error/ Ele Edit Diagram Navigate Search Project AAML Analyses Bun Window Help Acceleo warning I Packa D Model 11 ** D C *new Physical Com... C *new LIART Dock Me... mew UART Dock PU. messages when type fiter text © Palette SHECO.C. Local Session: AAML et <<Logical Component>> Components issues are Representations per FDF CLogical Component > > Representations per Error Eve Data Flow Ports AAML example.aaml Bus load and Data late detected during *
 Avionic Model Model selection <<Lopical Component>> aaml * + Avionics Function TCS_Acq_Manage Select a model and the output folder for analysis results the analysis. # + Avionic Functio Model: /AAML_models/Sentinel3.aaml Browse. + Avionic Functio **J** Avionics Functi Output folder Demonstration: + + Avionics Logical / emperature_InDataFlowPort Use default location + + Logical Compo Location: //home/assert/workspace ≱ ♦ Logical Compo **Bus selection** aaml + + Logical Compo Select the bus(es) to analyze Sentinel-3 + Logical Compo I + Logical Compo Result @Error Log 生 Problems 🖾 Conso **Bus type** Analysis kind Bus Use Case. + + Logical Compo + Logical Component TCSobject P/F M15538 MIL_STD_15538 Coarse 🔻 + + Logical Compo < Back Next > SMU_PDHU R5485 Coarse Property Semantic OLCI_PDHU R\$485 Fine Logical Component TCS STVIE Avionics Physical
 Avionics Physical
 Avionics
 Avion SLSTR PDHU **RS485** Coarse + Avionic Function CalculateTCSObjectiv Appearance Avionic Function + Data Types Mapped Onto + Processor Board ProcessorBoard + Spacecraft Mode Select All Deselect All All Coarse All Fine TCSobjectiveMGT Name + + Communication * + System Mapping < Back Next > Cancel **Einish**



AAML - Final Presentation

PROTOTYPE DEMONSTRATION-USE CASE

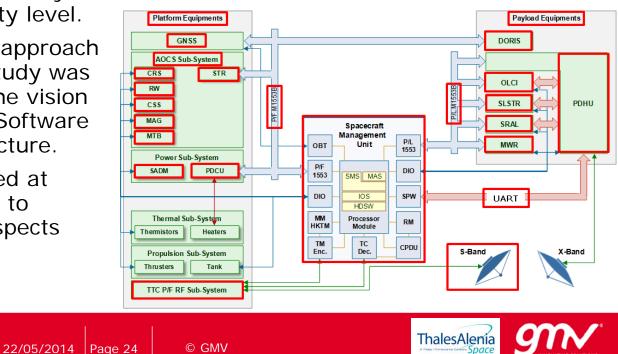




PROTOTYPE DEMONSTRATION – USE CASE

Use Case:

- It was intended to demonstrate at full extend all the capabilities of the AMML modelling methodology, meta-model and toolset as well as its corresponding avionics analyses.
- It was based on the Sentinel-3 mission.
- It allowed:
 - Exercising various analyses at avionics level based on a real case with a classical complexity level.
 Platform Equipments
 - Ensuring that the approach proposed in the study was compatible with the vision on the On-Board Software Reference Architecture.
 - The use case aimed at being rich enough to evaluate all the aspects of the approach.









Functional Architecture Definition.



- Logical Architecture Definition.
- Physical Architecture Definition.
- Execution of the analysis.





□ Steps:

- Functional Architecture Definition.
- Logical Architecture Definition.



- Physical Architecture Definition.
- Execution of the analysis.





□ Steps:

- Functional Architecture Definition.
- Logical Architecture Definition.

Physical Architecture Definition.

Execution of the analysis.







□ Steps:

- Functional Architecture Definition.
- Logical Architecture Definition.
- Physical Architecture Definition.

Execution of the analysis.







<<Sensor>>

<<1553

RT1553B

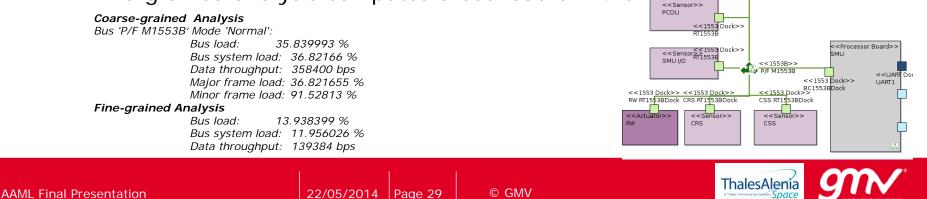
PROTOTYPE DEMONSTRATION – C&O AND 1553B RESULTS

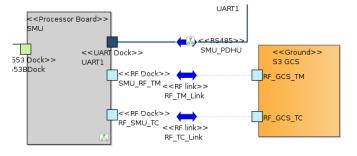
Commandability and observability.

- TM link at very low level of occupation (1.3-2.6%).
- TC links:
 - Assumption of visibility window of 10 min.
 - o TC upload of SRAL binary: 7974 bps, 55.1% of occupation.
 - o TC upload of CSW binary: 20796 bps, 143.7% of occupation.

Bus load: 1553B.

- Scheduling: Major frame of 1000 ms and a minor frame of 125 ms.
- Fine-grained analysis computes a bus load of 14%.





<<Sensor>> GNSS

<<Sensor>> STR

<<1553 Dock>> RT1553E

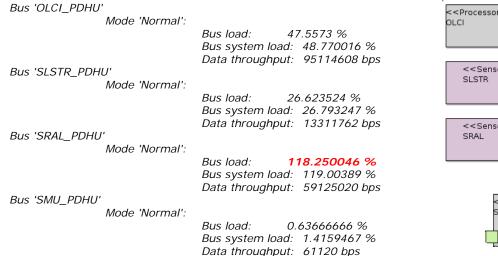
<<1553 Dock>> RT1553B

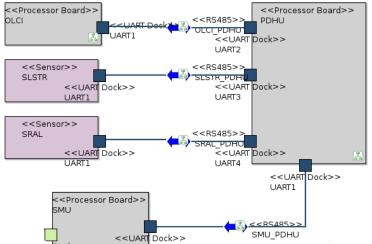


PROTOTYPE DEMONSTRATION – UART RESULTS

Bus load: UART.

Fine-grained analysis detects that SRAL-PDHU data exchange exceeds bus capability (due to calibration messages).





After introduction of calibration mode the bus load is reduced to 55.4% (normal) and 62.8% (calibration).

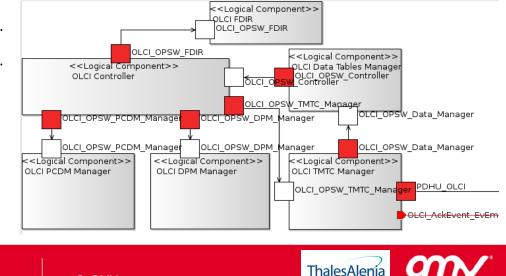




PROTOTYPE DEMONSTRATION – ON-BOARD FUNCTIONS AND PERFORMANCE RESULTS

On-board functions and performance.

- Firstly, only one EEPROM is used.
- The fined-grained analysis detects:
 - Computation of CPU load of OLCI OPSW: **51.6%**.
 - SRAM occupation of **35.7%** (below 50%).
 - High EEPROM1 occupation: 124.4%.
- Reallocation of DPM, PCDM, FDIR, PDHU PL and SMU logical components over EEPROM2:
 - EEPROM1 occupation: **67.3%**.
 - EEPROM2 occupation: **56.9%**.



AAML - Final Presentation

CONCLUSIONS AND FUTURE WORK





CONCLUSIONS

- □ AAML study has provided:
 - Identification and evaluation of the avionics analyses.

AAML modelling language:

- The set of entities and non-functional properties included in the language are precise and practical enough for capturing the avionics architecture and to be used as input for specialized avionics analysis.
- The language suitably supports the possibility of both **coarse- and fine-grained specification** by means of the non-functional properties defined.
- The three levels of definition (functional, logical and physical) supported by the language provide all the necessary means to manage the **different phases** of conception and implementation of the avionics system.

AAML toolset:

- **Design and analysis** of the avionics system.
 - The execution of the avionics analysis process is fast and easy to follow.
 - The analysis outputs are useful to confirm or modify the system design and to detect model inconsistencies or missing information.





FUTURE WORK

- Some future work activities have been identified during this study.
- They represent improvements in the modelling language and the toolset.
- The future work activities have been prioritized: High, Medium, Low.
- □ Some **examples**:

Future Work	Priority
Extend the meta-model and toolset to support additional avionic analyses	MEDIUM
Improve the analysis reports output format	MEDIUM
Develop and independent model consistency validator	HIGH
Include hierarchy levels	MEDIUM
Modelling errors	LOW





Thank you

Elena Alaña Salazar ealana@gmv.com

Space Systems Business Unit Avionics & On-Board SW Division



AAML - Final Presentation USE CASE IMAGES





-Space

SENTINEL 3: PACKAGE STRUCTURE

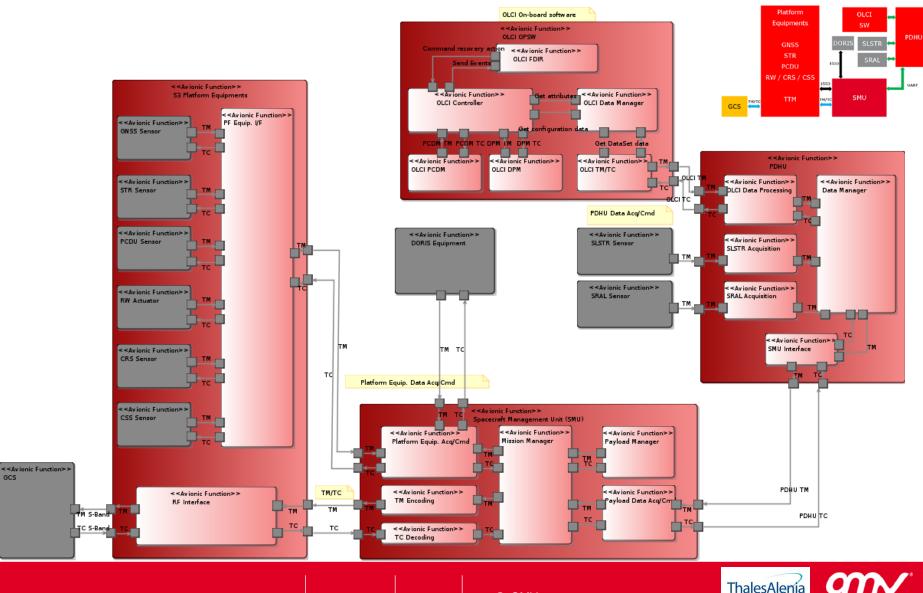
< <data package="">> Data Types</data>	< <communication package="">> Communication</communication>	< <s c="" modes="" package="">> Spacecraft Modes</s>
< <data package="">> Basic Types</data>		Spacecrait Modes
< <data package="">> Structure Types</data>	< <communication package="">> Interfaces</communication>	
< <data package="">> PDHU P/L Types</data>	< <communication package="">> Events</communication>	
< <data package="">> TM/TC PF Types</data>		
< <avionics functional="" package="">> Avionics Functional Architecture</avionics>	< <avionics logical="" package="">> Avionics Logical Architecture</avionics>	< <avionics package="" physical="">> Avionics Physical Architecture A</avionics>
	< <system mapping="" package="">> System Mapping</system>	
		₽ 48
		ThalesAlenia

22/05/2014 Page 37





SENTINEL 3: AVIONICS FUNCTIONAL DIAGRAM



AAML Final Presentation

GCS

22/05/2014 Page 38



SENTINEL 3: DATA TYPES (I)

<< Array Ty

OLCI_Calibr

<< Array Ty

OLCI_Norma

<<Array Ty SMU_Scien

<<Array Ty SMU_HKTM

<< Array Ty SMU_DiagD

<<Array Ty SLSTR_Day

<< Array Ty

SLSTR_Nigh

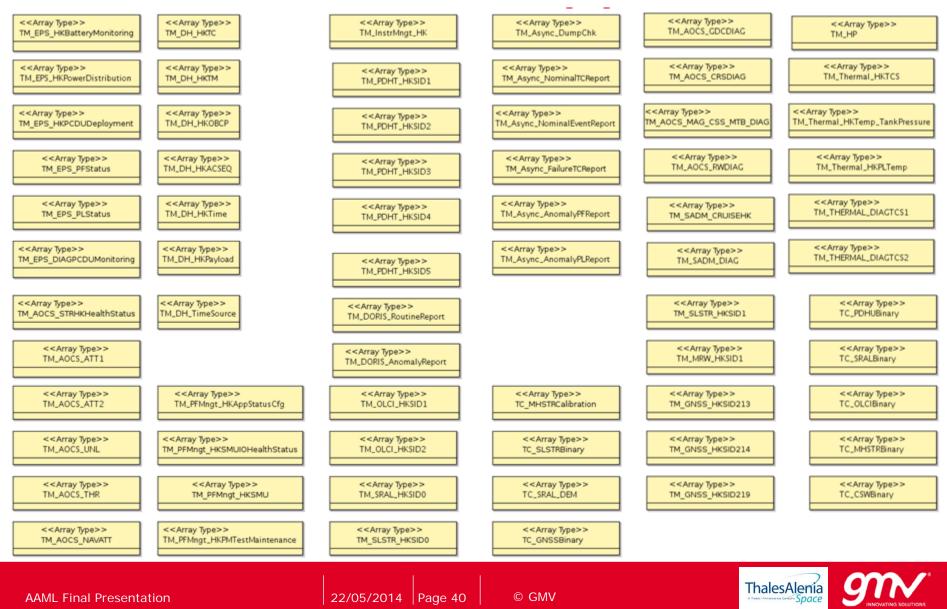
< <fixed lenght="" string="">> NameStr</fixed>	<<>StructuredType>> ScientificStructure
	StructureName :NameSt
	Id :Uint16 Offset :Uint32
< <integer>></integer>	Soluci :float32
Uint16	□ Value2 :double64
< <integer>></integer>	< <array type="">> ScientificArray100</array>
Uint32	ociencinci anay 200
	< <structuredtype>></structuredtype>
< <float>> float32</float>	ScientificData
fioat52	📼 data :ScientificArray100
< <float>></float>	
double64	

ype>> pration	< <array type="">> SRAL_TM_CAL1_LRM_I2Q2</array>	< <array type="">> Array_Uint16_1</array>	< <array type="">> Array_Uint16_18</array>	< <array type="">> ArraySingleChar</array>	< <array type="">> Array_Uint16_96</array>
ype>> nal	< <array type="">> SRAL_TM_CAL1_LRM_IC</array>	< <array type="">> Array_Uint16_2</array>	< <array type="">> Array_Uint16_32</array>	< <array type="">> ArrayKiloChar</array>	< <array type="">> Array_Uint16_118</array>
ype>> nce	< <array type="">> SRAL_TM_CAL1_SAR</array>	< <array type="">> Array_Uint16_3 <<array type="">></array></array>		< <array type="">> Array_Uint16_30</array>	< <array type="">> Array_Uint16_124</array>
ype>>	< <array type="">> SRAL_TM_CAL2_SAR</array>	Array_Uint16_4 < <array type="">> Array_Uint16_5</array>	< <array type="">> Array_Uint16_20</array>	< <array type="">> Array_Uint16_32</array>	< <array type="">> Array_Uint16_126</array>
Dump	SRAL_TM_SELF_TEST	< <array type="">> Array_Uint16_8</array>	< <array type="">> Array_Uint16_22</array>	< <array type="">> Array_Uint16_40</array>	< <array type="">> Array_Uint16_128</array>
ype>> y	< <array type="">> SRAL_TM_ECHO_LRM</array>	< <array type="">> Array_Uint16_9</array>	< <array type="">> Array_Uint16_24</array>	< <array type="">> Array_Uint16_48</array>	< <array type="">> Array_Uint16_152</array>
ype>> ht	< <array type="">> SRAL_TM_ECHO_SAR</array>	< <array type="">> Array_Uint16_11</array>	< <array type="">> Array_Uint16_26</array>	< <array type="">> Array_Uint16_62</array>	< <array type="">> Array_Uint16_192</array>
		< <array type="">> Array_Uint16_12</array>	< <array type="">> Array_Uint16_28</array>	< <array type="">> Array_Uint16_64</array>	< <array type="">> Array_Uint16_416</array>
		< <array type="">> Array_Uint16_16</array>	< <array type="">> Array_Uint16_29</array>	< <array type="">> Array_Uint16_80</array>	< <array type="">> Array_Uint16_512</array>





SENTINEL 3: DATA TYPES (II)





SENTINEL 3: SPACECRAFT MODES

<<Mode>> Standby

<<Mode>> Normal

<<Mode>> Safe





SENTINEL 3: COMMUNICATION ENTITIES (I)

	SMU Mil-STD-1553B Acq/Cmd		PDHU UART Acq/Cmd	OLCI OPSW Interfaces
< <interface>></interface>	< <interface>></interface>	< <interface>></interface>	< <interface>> PDHU_OLCI_Ac</interface>	< <interface>></interface>
SMU_STR_Acq © GetSTRStatus2 GetSTRStatus2 © GetSTRATM3 © GetSTRATM3 © GetSTRATM2 © GetSTRATM2 © GetSTRATM4	SMU_GNSS_Comman © SetTime SetAtionsAfPatch © SetCommandDurp © GredAmeriaAray_Unit5 © CmdAttindeVector:Array_Unit5]	SMU_PCDU_Acc © GetDistributionModul: © GetTMTC © BistributionModule : Array_Uint16_ © PistributionNodule : Array_Uint16_ © TMTC : Array_Uint16_2: © TMTC : Array_Uint16_2: © TMTC : Array_Uint16_2:	 Get0LCINormalData Get0LCICalibrationData OLCINormalData : OLC_Norm OLCICalibrationData : OLCI_Calibrat 	OLCI_OPSW_Controlle Time_Manage Startupf FCLKx4_Schedulei
© GeSTRATMA GeSTRATMS GeSTRATMS GeSTRATMS GeSTRATMS GeSTRATMS GeSTRATMD GeSTRATM10 GeSTRATM11 GeSTRATM12		 SetHS() HS : Array_Uintl6_1	PDHU_SLSTR_Ac GetSLSTRDayData(OUT SLSTR_C GetSLSTRNightData SLSTRDayData : SLSTR_Da SLSTRDightData : SLSTR_Nig	< <interface>> OLCI_OPSW_TMTC_Manag∈</interface>
© GetSTRATM13 © GetSTRATM14 © GetSTRATM15 © GetSTRATM15 © GetSTRATM15 © Staus1:Aray_Uint16_11 © Staus2:Aray_Uint16_31 © Staus3:Aray_Uint16_3 © ATM2:Aray_Uint16_3 © ATM2:Aray_Uint16_3	© Pollingt ♥ Navigation5olution : Array_Uint16 ♥ TimeCorrelation : Array_Uint16 ♥ HKParameteReport : Array_Uint15; SMU_GNS5_5-TW O Pollingt O Pollingt O GetTrackingState G GetStatlineNtwe Statu	<pre></pre> <interface>> SMU_DORIS_Ac Get0oppie GetTRHavjatio GetCeodesicalNavjatio Get2oation Get2oation Get2oation Get2oation Get2oation </interface>	<pre>> POHU_SRAL_Acq_CA @ GetSRALCall_LRM_I2Q; @ GetSRALCall_LRM_I2Q; @ GetSRALCall_SAR @ GetSRALCall_SAR @ GetSRALCal2_SAR % SRALCal2_LRM_I2Q2 : SRAL_TM_CAL1_LRM_I % SRALCal2_LRM_I0 : SRAL_TM_CAL1_SAF </pre>	© SMU_If_TM_Bus_Manage © SMU_If_TC_Bus_Manage © SMU_If_Non_Pus_Manage © Commandability_TC_Serv
 ▲ TMB : Array_Uint15_3 	GerConstellation Status GerConstellation Status GerCarrierPhase GerCarrierPhase GerCodePhase GerCode	© GetAnomalyt © Polling © Doppler: Array_Unit16_3 © ITRFHavigation : Array_Unit15_ © GeodesicalNavigation : Array_Unit15_ © Jacottaingation : Array_Unit15_3 © Reutine : Array_Unit15_3 © Anomaly: Array_Unit15_3 <td>SRALCal2_SAR : SRAL_TM_CAL2_S4 POHU_SRAL_Acq_NORMA @ GetSRAL_Echo_LRM @ GetSRAL_Echo_SAR @ GetSRAL_SelfTest \$ SRALEchoLRN : SRAL_TM_ECHO_LR \$ SRALEchoLSAR : SRAL_TM_ECHO_SA \$ SRALEchtelFest : SRAL_TM_SELFTE:</td> <td><<interface>> OLCI_OPSW_PCDM_Manag If_SIN2_Manage If_Com_Int_Manage</interface></td>	SRALCal2_SAR : SRAL_TM_CAL2_S4 POHU_SRAL_Acq_NORMA @ GetSRAL_Echo_LRM @ GetSRAL_Echo_SAR @ GetSRAL_SelfTest \$ SRALEchoLRN : SRAL_TM_ECHO_LR \$ SRALEchoLSAR : SRAL_TM_ECHO_SA \$ SRALEchtelFest : SRAL_TM_SELFTE:	< <interface>> OLCI_OPSW_PCDM_Manag If_SIN2_Manage If_Com_Int_Manage</interface>
♦ ATM15 : Array_Uint16_3 ♦ ATM17 : Array_Uint16_3 < <interface>> SMU_RW_Acq © GetAcqData(© GetHealthMonitorin, ♥ RWAcqData: Array_Uint16_;</interface>	Contellations and Array Unit15_1 Gastaline Status Array Unit15_1 ConstellationStatus : Array Unit15_1 ContellationStatus : Array Unit15_11 CarrierPhase Array Unit15_12 CarrierPhase : Array Unit15_13 CodePhase : Array Unit15_13 CodePhase : Array Unit15_15 NoiseHistoram : Array Unit15_15 AccStatus : Array Unit15_15 CodeStatus : Array Unit15_13 CodeStatus	© SeuDORISTIME © CmdAthinde :Amy_Uint16_1 © CmdTime : Array_Uint16_ < CmdTime : Array_Uint16_	< <interface>> PDHU_SMU_Cmi</interface>	< <interface>> OLCI_OPSW_DPM_Manage If_SIN1_Manage If_Com_Int_Manage</interface>
RWHealthMonitoring : Array_Uint15_ < <interface>> SMU_RW_Crnd Image: SetRWFineSchedule : Array_Uint15_11</interface>	GPSIIavUFCIencephereni: Array_Uint15_1 GPSIIavUFCIencephere: Array_Uint15_ Minimumliavigation5olution: Array_Uint16_ MITGPSTCorrelation: Array_Uint16_ AuxiliavyData: Array_Uint16_1	© GetCRSUAPTE/StateRegiste © GetCRSUAPTE/Data © GetCSMUOHealthMonitorin © AOCSGroupRetrieval: Array_Uint15_ © HKAcgGroupRetrieval: Array_Uint15_4 © CRSUARTE:StateRegiste: Array_Uint16_ © CRSUARTE:Data:Array_Uint16_ © CRSUARTE:DAta:Array_UINT	SHUBScienceData : SHU_DiagDur SHUDiagDumpData : SHU_DiagDur SHU PM IF's < <interface>> SHU PM PL AcqCm</interface>	< <interface>> OLCI_OPSW_Data_Manage ⓒ Memory_Manage:</interface>
< <interface>> SMU_CRS_Acq © GetAcqData(© GetHealthMonitorin CRSAcpData, Array_Unt15_ ♥ CRSAcpData, Array_Unt15_</interface>	Politing Control Co	<pre></pre> SMU J0_Cmd Sex0C5RAMpantrol=lecto SetHXGrupRAMPaintrol=lecto SetUcFIFORere SetUcFICRAMPaintrol=lecto SetUcFICRAMPaintrol=lecto SetUcFICRAMPaintrol=lecto SetUcFICRAMPaintrol=lecto	<\nterface>> SMU_PM_TC	< <interface>> OLCI_OPSW_FDI</interface>
< <interface>> SMU_CRS_Cmc SetCRSCmdData CRSCmdData : Array_Uint16_1 <<interface>></interface></interface>	< SMU_GNSS_M-TM Polling(Ger0umpReport DumpReport.Array_Unit16_5:	SetCyclicProgrammin SetAyncProgrammin SetAyncProgrammin SetAyncProgramming SetASNDSteppetMotorContc ACCSRANPointerSelection McGroupRANPointerSelection WARTFIFORest: Array_Uint16 CyclicPANPointerSelection ACREADPlogintsSelection	< <interface>> SMU_PM_TM <<interface>> SMU_PM_PF_Acc</interface></interface>	FDIR_EQSOL_Manage FDIR_Single_EDAC_Manage FDIR_Watchdog_Manage FDIR_Scrubbing
SMU_CSS_Acq GetAcqData GetHealthMonitorinn CSSAcqData:Array_UintL6_ CSSHealthMoniotring:Array_UintL6_	<interface>> SMU_CS5_Cmc © SetCSSCmdData & CSSCmdData : Array_Uint16_</interface>	CyclicCmdProgramming : Array_Uint15 AsyncRAMPointerSelection : Array_Uint15 AsyncProgramming : Array_Uint15 SADMStepperMotorControl : Array_Uint15	< <interface>> SMU_PM_PF_Cm</interface>	



AAML Final Presentation

22/05/2014 Page 42



Space

SENTINEL 3: COMMUNICATION ENTITIES (II)

	< <interface></interface>	< <interface>> SMU_TM_PF_DF</interface>	< <interface>> SMU_TM_PL_InstrumentMng</interface>	< <interface>> SMU_TM_Async</interface>
Encode/13/Encod	EncodeHKPowerDistribution(OUT TM_EPS_HKPowerDistrib EncodeHKPCDUDeployment(OUT TM_EPS_HKPCDUDeploy	© EncodeMgtHKTC(OUT TM_DH_HKTC MgtHK' © EncodeMgtHKTM(OUT TM_DH_HKTM MgtHK' © EncodeMgtHKOBCP(OUT TM_DH_HKOBCP MgtHKOBC	EncodeHKInstrumentMngt(OUT	EncodeDumpChk(OUT TM_Async_DumpChk DumpCl EncodeNominalTCReport(OUT TM_Async_Nomin EncodeNominalEventReport(OUT TM_Async_Nomin
<pre></pre>	EncodePLStatus(OUT TM_EPS_PLStatus PLStatu CarcodePLAGPCDUMonitoring(OUT TM_EPS_DIAGPCDUMoni HKBatteryMonitoring : TM_EPS_HKRStateryMonito HKPCUDDeployment : TM_EPS_HKPCDUDeploym PFStatus : TM_EPS_PLStatus PLStatus : TM_EPS_PLStatus	EncodeMgtHKTime(OUT TM_OH_HKTime MgtHKT EncodeMgtHKFayload(OUT TM_DH_HKPay EncodeTimeSource(OUT TM_DH_TimeSource TimeSc MgtHKTC:TM_DH_HKT MgtHKTC:TM_DH_HKT MgtHKOBCP:TM_OH_HKKOBC MgtHKACSEQ:TM_OH_HKACSE(MgtHKACSEQ:TM_DH_HKACSE(MgtHKFayload:TM_OH_HKRaylo	SMU_TM_PL_PDH' EncodePDHT_HKSID1(OUT TM_PDHT_HF EncodePDHT_HKSID2(OUT TM_PDHT_HF EncodePDHT_HKSID3(OUT TM_PDHT_HF EncodePDHT_HKSID4(OUT TM_PDHT_HF EncodePDHT_HKSID5(OUT TM_PDHT_HF HK_PDHT_SID1: TM_PDHT_HKSI HK_PDHT_SID2: TM_PDHT_HKSI	EncodeAnomalyPFReport OUT TM_Async_Anoma EncodeAnomalyPLReport OUT TM_Async_Anoma DumpCh: TM_Async_DumpCh NominalTCReport: TM_Async_NominalTCRe NominalEventReport: TM_Async_NominalEventRe FailureTCReport: TM_Async_AnomalyPFRep
	< <interface>></interface>	< <interface>></interface>	<pre></pre>	
EncedetRig OUT TM_ACCS_UNU UN EncedetRig OUT TM_ACCS_CREADER OUT TC_MARKSMU UN EncedetRig OUT TM_ACCS_CREADER OCH EncedetRig OUT TM_ACCS_CREADER OUT TC_MARKSMU SMUUMER AND EncedetRig OUT TM_ACCS_CREADER OUT TC_GREAT SMUUMER AND EncedetRig OUT TM_ACCS_CREADER SMUUMER AND SRALER AND SR		EncodeHKAppStatusCfg(OUT TM_PEMngt_HKAppStatu		
Encode#HXPUUT TM_AOCS_THE TH C=Incode#HXPUTs_MAXTT NAXA Encode#LXPUTS_CUST_NAXTT NAXA C=Incode#IXPUTS_CUST_NAXTT NAXA EncodeHXPUTS_CUST_NAXTT NAXA C=IncodeHXPUTS_CUST_NAXTT NAXA EncodeHXPUTS_CUST_NAXTT NAXA SMUUCH=AtmStatus_TM_PFMng_HXSMS_SMUUCH=AtmStatus_SMUUCH=AtmStatus_SMUUCH=AtmStatus_TM_PFMng_HXSM EncodeHXPUTS_CUST_NAXTT NAXACS_CRSDLAG CGSS_MTB_D SMUUCH=AtmStatus_TM_PFMng_HXSM STRHX=HatmStatus_TM_AOCS_NAATT_NAXT SOULCE EncodeHXPTM_CUST_NAXTT NAXACS_STRHXHealthSta <	EncodeATT2(OUT TM_AOCS_ATT2 AT	EncodeHKSMUIOHealthStatus(OUT		DecodeMHSTRCalibration(OUT TC_MHSTRC; DecodeSLSTRBinard OUT TC_SLSTRBinard SLSTRB;
incadeCDDAG OUT TM_AOCS_CRSDLAG GDCD/A incadeCDDAG OUT TM_AOCS_CRSDLAG GDCD/A incadeCSDAG OUT TM_AOCS_CRSDLAG GDCD/A incadeRDAG_CSS_MTB_DIAG(OUT TM_AOCS_MAG_CSS_MTB incadeRSTRDIAG_ATM incadeST	incodeTHR(OUT TM_AOCS_THR TH	EncodeHKPMTestMaintenance(OUT		DecodeSRALDEM(OUT TC_SRAL_DEM SRALDE
EncodeMAG_CSS_MTE_DIAG(OUT TM_AOCS_MAG_CSS_MTE incodeRVDIAG(OUT TM_AOCS_WDIAG RVDIAG) incodeRVDIAG(OUT TM_AOCS_WDIAG RVDIAG) incodeSTRDIAG_ATM incodeSTRDIAG_CS_MTB_D incodeSTRDIAG_CS_MTB_D incodeSTRDIAG_CS_MTB_DO incodeSTRDIAG_CS_MTB_C	ncodeGDCDIAG(OUT TM_AOCS_CRSDIAG GDCDIA	SMUIOHealthStatus : TM_PFMngt_HKSMUIOHealthSta		DecodePDHUBinary(OUT TC_PDHUBinary PDHUBinary)
incodeRWDIAG QUT TM_AOCS_RWDIAG RWDIAG incodeSTRDIAG_ATM incodeSTRD				
<pre>incodeSTRDIAG_ATM TRHKHeathStatus:ITM_AOCS_ATC TRHKHeathStatus:ITM_AOCS_ATC TRHKHeathStatus:ITM_AOCS_ATC TRHKHeathStatus:ITM_AOCS_ATC TRHKHeathStatus:ITM_AOCS_ATC TT2:ITM_</pre>	ncodeRWDIAG(OUT TM_AOCS_RWDIAG RWDIA(DecodeMHSTRBinary(OUT TC_MHSTRBinary MHS
ATT1: TM_AOCS_AT JML: TM_AOCS_AT JML: TM_AOCS_AT JML: TM_AOCS_AT JML: TM_AOCS_STH SEALDEMAY S	EncodeSTRDIAG_ATM:	< <interface>></interface>		MHSTRCalibration : TC_MHSTRCalibra
UHL : TM_AOCS_UHI THR : TM_AOCS_TH NAVATI : TM_AOCS_NAVAT GEncodeIDAGTCSI (UIT TM_THERMAL_DIAGTCS DIAG GEOCHAG : TM_AOCS_GOEDIA CRSDIAG : TM_SADM_CRUISEH CRCDAGEDIAG CSS_MTE_D CRCDAGEDIAG : TM_SADM_CRUISEH CRCDAGEDIAG : TM_SADM_CRUISEH CRUISE : TM_CRUISE : TM_UTM_PL_DORIS CRUISE : TM_CRUISE : TM_CRUISE : TM_SISTE_HKSI CRUISE : TM_CRUISE : TM_CRUISE : TM_SISTE_HKSI CRUISE : TM_CRUISE : T	ATT1:TM_AOCS_AT	EncodeHKTCS(OUT TM_Thermal_HKTCS T		SRALDEM : TC_SRAL_DE!
THR: TIM_AOCS_TH SRALBinay: TC_SRALBina MAXATT: TM_AOCS_NAVAT GEncodeDIAGTCSI QUIT TM_THERMAL_DIAGTCSI DIAG SRALBinay: TC_SRALBina GOCDIAG: TM_AOCS_GOCDIA GEncodeDIAGTCSI QUIT TM_THERMAL_DIAGTCSI DIAG SRALBinay: TC_SRALBina GOCDIAG: TM_AOCS_GOCDIA GEncodeDIAGTCSI QUIT TM_THERMAL_DIAGTCSI DIAG SRALBinay: TC_SRALBina MAG_CS_MTB_DIAG: TM_AOCS_MAG_CSS_MTB_D Thermal_HKT(FFTemp_TaniPresure: TM_Thermal_HKTemp_TankPres SMU_TM_PL_SRA SMU_TM_PF_SADN OLAG_TCSI: TM_THERMAL_DIAGTC OLAG_TCSI: TM_THERMAL_DIAGTC SMU_TM_PL_SLSTF OLAG_TCSI: TM_THERMAL_DIAGTC OLAG_TCSI: TM_THERMAL_DIAGTC SMU_TM_PL_SLSTF SMU_TM_PL_SLSTF CRUISE: TM_SADM_CRUISEHK CRUIS SMU_TM_PF_HPTN OLAG_TCSI: TM_THERMAL_DIAGTC SMU_TM_PL_SLSTF SMU_TM_PL_SLSTF OLAG: TM_SADM_CRUISEH OLAG_TCSI: TM_THERMAL_DIAGTC SMU_TM_PL_SLSTF SMU_TM_PL_SLSTF SMU_TM_PL_SLSTF OLAG: TM_SADM_CRUISEH OLAG_TCSI: TM_THERMAL_DIAGTC SMU_TM_PL_SLSTF SMU_TM_PL_MWI SMU_TM_PL_MWI SMU_TM_PL_MWI				
GDCDLAG: TM_AOCS_GDCDIA CRSDLAG: TM_AOCS_GCSDIA! CRSDLAG: TM_AOCS_CRSDIA! MAG_CSS_MTB_DLAG: TM_AOCS_MAG_CSS_MTB_D PHTEMP_TAIN_HEREWILE: TM_Thermal_HKPIre PHTEMP_TAIN_HEREWILE: TM_Thermal_HKPIre PHTEMP_TAIN_HEREWILE: TM_Thermal_HKPIre PHTEMP_TAIN_HEREWILE: TM_Thermal_HKPIre PHTEMP_TAIN_HEREWILE: TM_THERMAL_DIAGTC DIAG_TCS1: TM_THERMAL_DIAGTC DIAG_TCS2: TM_THERMAL_DIAGTC SMU_TM_PF_SADA EncodeCRUISE(OUT TM_SADM_CRUISEHK CRUIS CRUISE: TM_SADM_CRUISEH CRUISE: TM_SADM_CRUISEH <		EncodeDIAGTCS1 OUT TM_THERMAL_DIAGTCS1 DIAG		💊 SRALBinary : TC_SRALBina
MAG_CSS_MTE_DIAG: TM_AOCS_MAG_CSS_MTB_D PLTemp: TM_Thermal_HKPLTe VDAG_TCS1: TM_THERMAL_DIAGTC VDAG_TCS1: TM_THERMAL_DIAGTC VDAG_TCS2: TM_THERMAL_DIAGTC VALUE CRUISE(OUT TM_SODM_CRUISEH DIAG: TM_SODM_DIAG CRUISE + TM_SADM_DIAG CRUISE + TM_SADM_DIAG VALUE VA	GDCDIAG : TM_AOCS_GDCDIA	▼ TCS : TM_Thermal_HKT(💊 MHSTRBinary : TC_MHSTRBina
< <interface>> SMU_TM_PF_SADN EncodeCRUISE(OUT TM_SADM_CRUISEHK CRUIS EncodeDIAG(OUT TM_SADM_DIAG DIA CRUISE : TM_SADM_CRUISEH OIAG : TM_SADM_CRUISEH OIAG : TM_SADM_DIA(<</interface>	MAG_CSS_MTB_DIAG : TM_AOCS_MAG_CSS_MTB_D		<pre> HK_SRAL_SIDU : TM_SRAL_HKSIL </pre>	SWBinary : TC_CSWBinarj
<	RWDIAG · TM AOCS RWDIAC	DIAG_TCS1: TM_THERMAL_DIAGTC DIAG_TCS2: TM_THERMAL_DIAGTC		
EncodeCRUISE(OUT TM_SADM_CRUISEHK CRUIS EncodeDIAG(OUT TM_SADM_DIAG DIA CRUISE: TM_SADM_CRUISEH DIAG : TM_SADM_CRUISEH OLAG : TM_SADM_DIAG SMU_TM_PF_HPTN Image: TM			< <interface>></interface>	
EncodeDIAG(OUT TM_SADM_DIAG DIA CRUISE : TM_SADM_CRUISEH DIAG : TM_SADM_DIA(CRUISE : TM_SADM_CRUISEH DIAG : TM_SADM_DIA(CRUISEH CRUISE		< <interface>></interface>		
DIAG : TM_SÄDM_DÏA(DIAG : TM_SÄDM_DÏA(C EncodeHPTM(OUT TM_HP HPT) HPTM : TM_HF C EncodeHPTM(OUT TM_HP HPT) HPTM : TM_HF C EncodeConstruction Construction Constructi	EncodeDIAG(OUT TM_SADM_DIAG DIA			
< <interface>> SMU_TM_PL_DORI: SMU_TM_PL_DORI: SMU_TM_PL_DORIS_RoutineF EncodeGNSS_SID213(OUT TM_GNSS_HKSID213 GNSS_SID: Encode_DorisAnomalyReport OUT TM_DORIS_AnomalyF EncodeGNSS_SID214(OUT TM_GNSS_HKSID214 GNSS_SID: Encode_DorisAnomalyReport OUT TM_DORIS_AnomalyF K GNSS_SID219(OUT TM_GNSS_HKSID219 GNSS_SID: NotwineReport: TM_DORIS_AnomalyRep K GNSS_SID219(OUT TM_GNSS_HKSID219 GNSS_SID: AnomalyReport: TM_DORIS_AnomalyRep</interface>			HK_SLSTR_SID0 : TM_SLSTR_HKSI	
< <interface>> SMU_TM_PL_DORI: EncodeGNSS_SID213(OUT TM_GNSS_HKSID213 GNSS_SID: Encode_DorisRoutineReport OUT TM_DORIS_RoutineReport <<interface>> EncodeGNSS_SID214(OUT TM_GNSS_HKSID214 GNSS_SID: Encode_DorisRoutineReport Smothereport Smothereport NotifieReport: TM_DORIS_RoutineReport OUT TM_DORIS_AnomalyReport SmU_TM_PL_MWI EncodeGNSS_SID219(OUT TM_GNSS_HKSID219 GNSS_SID: NotifieReport: NotifieReport: Smothereport HK_GNSS_SID213: TM_GNSS_HKSID219 GNSS_SID: AnomalyReport: TM_DORIS_AnomalyRep HK_MWR_SID1</interface></interface>		• · · · · · · · · · · · · · · · · · · ·	V HK_SLSTK_SIDT: TM_SLSTK_HKST	
EncodeGNSS_SID213(OUT TM_GNSS_HKSID213 GNSS_SID: Image: Construction of the con		< <interface>></interface>		
EncodeGNSS_SID213(OUT TM_GNSS_HKSID213 GNSS_SID: EncodeGNSS_SID214(OUT TM_GNSS_HKSID214 GNSS_SID: EncodeGNSS_SID219(OUT TM_GNSS_HKSID219 GNSS_SID: HK_GNSS_SID213: TM_GNSS_HKSID219 GNSS_SID: HK_GNSS_SID213: TM_GNSS_HKSID2	< <interface>></interface>		< <interface>></interface>	
EncodeGNSS_SID219(OUT TM_GNSS_HKSID219 GNSS_SID: AnomalyReport: TM_DORIS_AnomalyReport: TM_DORIS_AnomalyReport: TM_DORIS_AnomalyReport: TM_DORIS_AnomalyReport: TM_SID1: TM_MRW_HKSID		Encode_DorisAnomalyReport(OUT TM_DORIS_AnomalyF		
HK_GNSS_SID213 : TM_GNSS_HKSID2	EncodeGNSS_SID219(OUT TM_GNSS_HKSID219 GNSS_SID:			
HK 0N35 SID214 IM 0N35 HKSID2	HK_GNSS_SID213 : TM_GNSS_HKSID2 HK_GNSS_SID214 : TM_GNSS_HKSID2			
				ThalesAlenia

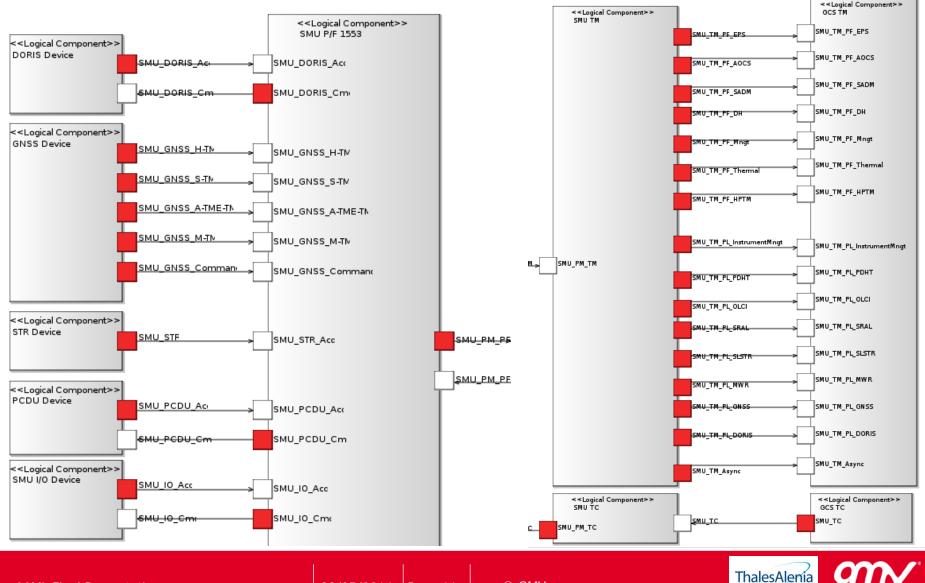
AAML Final Presentation

22/05/2014 Page 43



Cesa

SENTINEL 3: AVIONICS LOGICAL DIAGRAM (I)

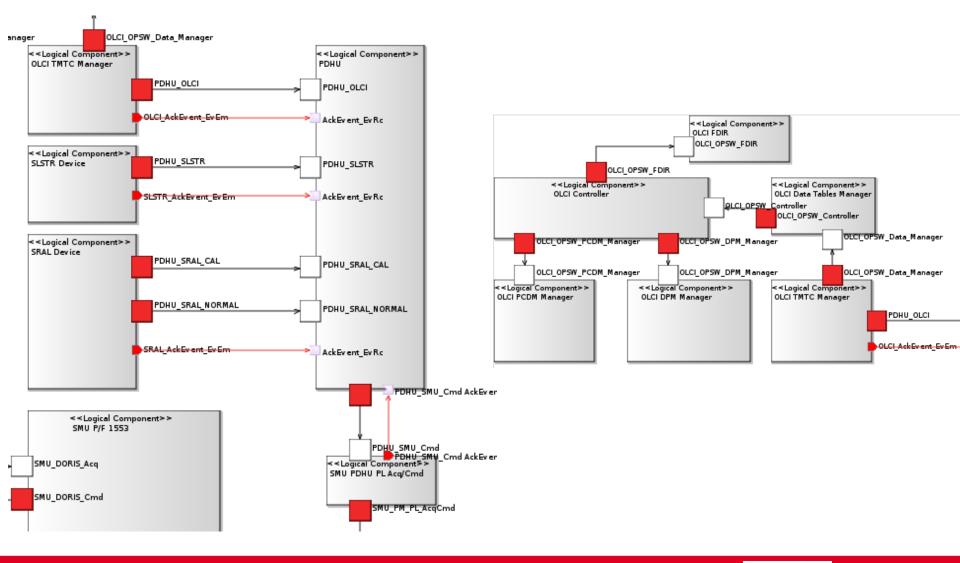


AAML Final Presentation

22/05/2014 Page 44



SENTINEL 3: AVIONICS LOGICAL DIAGRAM (II)







SENTINEL 3: MAPPING FROM AVIONICS FUNCTIONS TO LOGICAL COMPONENTS (I)

	Logical Component OLCI Controller
Avionic Function OLCI FDIR	Logical Component OLCI FDIR
Avionic Function OLCI Controller	Logical Component OLCI Controller
🖘 Avionic Function OLCI Data Tables	Logical Component OLCI Data Tables Manager
Avionic Function OLCI TM/TC	Logical Component OLCI TMTC Manager
Avionic Function OLCI PCDM	Logical Component OLCI PCDM Manager
Avionic Function OLCI DPM	Logical Component OLCI DPM Manager
マ = Avionic Function PDHU	Logical Component PDHU
Avionic Function PDHU OLCI Acq/Cmd	Logical Component PDHU
🖘 Avionic Function PDHU Data Manager	Logical Component PDHU
Avionic Function PDHU SLSTR Acq	Logical Component PDHU
Avionic Function PDHU SMU IF	Logical Component PDHU, Logical Component SMU I/O Device
🖘 Avionic Function PDHU SRAL Acq	Logical Component PDHU
🗢 🖘 Avionic Function Spacecraft Management Ur	Logical Component SMU Processor Module
Avionic Function SMU PDHU PL Acq/Cmd	Logical Component SMU PDHU PL Acq/Cmd, Logical Component PDHU
🖘 Avionic Function SMU Processor Module	Logical Component SMU Processor Module
🖘 Avionic Function SMU TM Encoder	Logical Component SMU TM
Avionic Function SMU P/F 1553	Logical Component SMU P/F 1553
Avionic Function SMU P/L 1553	
🖘 Avionic Function SMU TC Decoder	Logical Component SMU TC
Avionic Function SLSTR Sensor	Logical Component SLSTR Device
Avionic Function SRAL Sensor	Logical Component SRAL Device



22/05/2014 Page 46



SENTINEL 3: MAPPING FROM AVIONICS FUNCTIONS TO LOGICAL COMPONENTS (II)

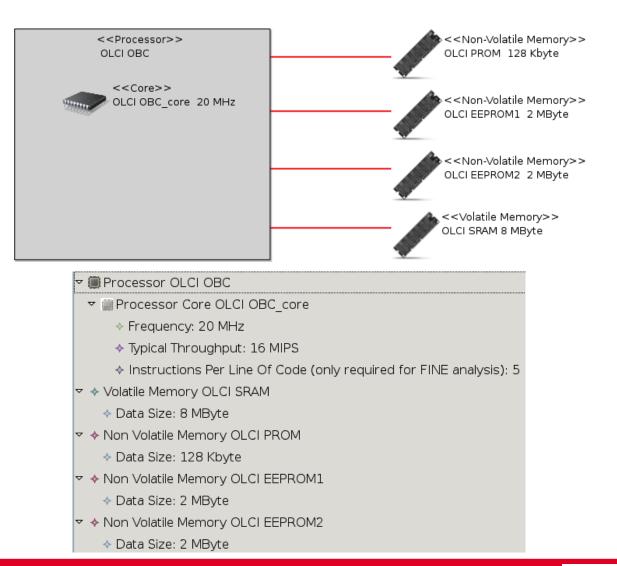
Avionic Function S3 Platform Equipments
 Avionic Function PF TTC RF
 Avionic Function PF 1553 I/F
 Avionic Function GNSS Sensor
 Avionic Function STR Sensor
 Avionic Function PCDU Sensor
 Avionic Function RW Actuator
 Avionic Function CRS Sensor
 Avionic Function GCS
 Avionic Function DORIS Equipment

Logical Component SMU Processor Module
Logical Component SMU P/F 1553
Logical Component GNSS Device
Logical Component STR Device, Logical Component SLSTR Device
Logical Component PCDU Device
Logical Component RW Device
Logical Component CRS Device
Logical Component CSS Device
Logical Component GCS TM, Logical Component GCS TC
Logical Component DORIS Device





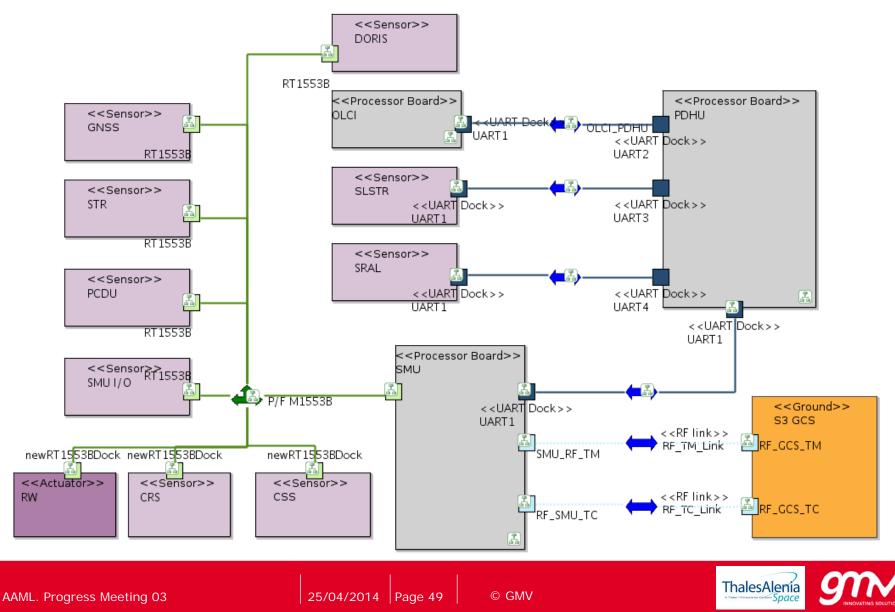
SENTINEL 3: PROCESSOR BOARDS







SENTINEL 3: PHYSICAL ARCHITECTURE





SENTINEL 3: MAPPING FROM LOGICAL TO PHYSICAL COMPONENTS

	Physical Component	Processor Core (Only for	Non Volatile Memory Use	Volatile Memory Used (Only for Proc
▼ 🕄 Logical Component PDHU	Processor Board PDHU			
Processor Board Mapping Descriptor		Processor Core PDHU_Proc_core	Non Volatile Memory PDHU Non\	Volatile Memory PDHU VolatileMemory
Logical Component SRAL Device	Sensor SRAL			
🗢 🕄 Logical Component OLCI TMTC Manager	Processor Board OLCI			
Processor Board Mapping Descriptor		Processor Core OLCI OBC_core	Non Volatile Memory OLCI EEPR	Volatile Memory OLCI SRAM
Processor Board Mapping Descriptor		Processor Core OLCI OBC_core	Non Volatile Memory OLCI EEPR	Volatile Memory OLCI SRAM
Logical Component SLSTR Device	Sensor SLSTR			
▼ 🖘 Logical Component SMU P/F 1553	Processor Board SMU			
Processor Board Mapping Descriptor		Processor Core SMU_Proc_core	Non Volatile Memory SMU NonVo	Volatile Memory SMU VolatileMemory
Logical Component STR Device	Sensor STR			
Logical Component GNSS Device	Sensor GNSS			
🖘 Logical Component PCDU Device	Sensor PCDU			
Logical Component DORIS Device	Sensor DORIS			
🖘 Logical Component GCS TM	Ground S3 GCS			
🗢 🐔 Logical Component SMU TM	Processor Board SMU			
Processor Board Mapping Descriptor		Processor Core SMU_Proc_core	Non Volatile Memory SMU NonVo	Volatile Memory SMU VolatileMemory
🗢 🖘 Logical Component SMU TC	Processor Board SMU			
Processor Board Mapping Descriptor		Processor Core SMU_Proc_core	Non Volatile Memory SMU NonVo	Volatile Memory SMU VolatileMemory
🖘 Logical Component GCS TC	Ground S3 GCS			
Logical Component SMU I/O Device	Sensor SMU I/O			
🖘 Logical Component RW Device	Actuator RW			
Logical Component CSS Device	Sensor CSS			
🗢 🕄 Logical Component OLCI Data Tables Ma	Processor Board OLCI			
 Processor Board Mapping Descriptor 		Processor Core SMU_Proc_core	Non Volatile Memory OLCI EEPR	Volatile Memory OLCI SRAM
マ =□ Logical Component OLCI Controller	Processor Board OLCI			
 Processor Board Mapping Descriptor 		Processor Core OLCI OBC_core	Non Volatile Memory OLCI PRO№	Volatile Memory OLCI SRAM
 Processor Board Mapping Descriptor 		Processor Core SMU_Proc_core	Non Volatile Memory OLCI PRON	Volatile Memory OLCI SRAM
マ ≂ Logical Component OLCI DPM Manager	Processor Board OLCI			
 Processor Board Mapping Descriptor 		Processor Core SMU_Proc_core	Non Volatile Memory OLCI PRO№	Volatile Memory OLCI SRAM
🗢 🖘 Logical Component OLCI PCDM Manage	Processor Board OLCI			
Processor Board Mapping Descriptor		Processor Core SMU_Proc_core	Non Volatile Memory OLCI PRON	Volatile Memory OLCI SRAM
マ € Logical Component OLCI FDIR	Processor Board OLCI			
Processor Board Mapping Descriptor		Processor Core SMU_Proc_core	Non Volatile Memory OLCI PRON	Volatile Memory OLCI SRAM
🗢 🖘 Logical Component SMU PDHU PL Acq/0	Processor Board SMU			
Processor Board Mapping Descriptor		Processor Core SMU_Proc_core	Non Volatile Memory OLCI PRON	Volatile Memory OLCI SRAM
マ =⊐ Logical Component SMU Processor Mod	Processor Board SMU			
		Processor Core SMU_Proc_core	Non Volatile Memory OLCI PRON	Volatile Memory OLCI SRAM



AAML. Progress Meeting 03

25/04/2014 Page 50



SENTINEL 3: MAPPING FROM LOGICAL TO PHYSICAL CONNECTIONS

🖩 S3 Logical Connections - Physical Connections Mapping Table 🛛									
	Connection	HW Communication Media	TC/TM Standard						
♦ Mapping: RF-TM_PF_EPS Connection	Interface Connection communicationSMU_TM_PF_EPSSMU_TM_PF_EPS	RFlink RF_TM_Link	PUS						
♦ Mapping: RF-TC Connection	Interface Connection communicationSMU_TCSMU_TC	RFlink RF_TC_Link	PUS						
Mapping: RF-TM_PF_AOCS Connection	Interface Connection communicationSMU_TM_PF_AOCS	RFlink RF_TM_Link	PUS						
Apping: RF-TM_PF_SADM Connection	Interface Connection communicationSMU_TM_PF_SADMSMU_TM_PF_SADM	RFlink RF_TM_Link	PUS						
♦ Mapping: RF-TM_PF_DH Connection	Interface Connection communicationSMU_TM_PF_DHSMU_TM_PF_DH	RFlink RF_TM_Link	PUS						
♦ Mapping: RF-TM_PF_Mngt Connection	Interface Connection communicationSMU_TM_PF_MngtSMU_TM_PF_Mngt	RFlink RF_TM_Link	PUS						
Mapping: RF-TM_PF_Thermal Connection	Interface Connection communicationSMU_TM_PF_ThermalSMU_TM_PF_Thermal	RFlink RF_TM_Link	PUS						
♦ Mapping: RF-TM_PF_HPTM Connection	Interface Connection communicationSMU_TM_PF_HPTMSMU_TM_PF_HPTM	RFlink RF_TM_Link	PUS						
Aapping: RF-TM_PL_InstrMngt Connect	Interface Connection communicationSMU_TM_PL_InstrumentMngtSMU_TM_PL_Ir	RFlink RF_TM_Link	PUS						
♦ Mapping: RF-TM_PL_PDHT Connection	Interface Connection communicationSMU_TM_PL_PDHTSMU_TM_PL_PDHT	RFlink RF_TM_Link	PUS						
♦ Mapping: RF-TM_PL_OLCI Connection	Interface Connection communicationSMU_TM_PL_OLCISMU_TM_PL_OLCI	RFlink RF_TM_Link	PUS						
Mapping: RF-TM_PL_SRAL Connection	Interface Connection communicationSMU_TM_PL_SRALSMU_TM_PL_SRAL	RFlink RF_TM_Link	PUS						
♦ Mapping: RF-TM_PL_SLSTR Connection	Interface Connection communicationSMU_TM_PL_SLSTRSMU_TM_PL_SLSTR	RFlink RF_TM_Link	PUS						
Apping: RF-TM_PL_MWR Connection	Interface Connection communicationSMU_TM_PL_MWRSMU_TM_PL_MWR	RFlink RF_TM_Link	PUS						
♦ Mapping: RF-TM_PL_GNSS Connection	Interface Connection communicationSMU_TM_PL_GNSSSMU_TM_PL_GNSS	RFlink RF_TM_Link	PUS						
Apping: RF-TM_PL_DORIS Connection	Interface Connection communicationSMU_TM_PL_DORISSMU_TM_PL_DORIS	RFlink RF_TM_Link	PUS						
♦ Mapping: 1553-SMU_GNSS_A-TME-TM	Interface Connection communicationSMU_GNSS_A·TME-TM	MIL STD 1553B P/F M1553B	PUS						
Mapping: RF-TM_Async Connection	Interface Connection communicationSMU_TM_AsyncSMU_TM_Async	RFlink RF_TM_Link	PUS						
♦ Mapping: 1553-SMU_DORIS_Acq Conn	Interface Connection communicationSMU_DORIS_AcqSMU_DORIS_Acq	MIL STD 1553B P/F M1553B	PUS						
♦ Mapping: 1553-SMU_GNSS_H-TM Conn	Interface Connection communicationSMU_GNSS_H-TM	MIL STD 1553B P/F M1553B	PUS						
♦ Mapping: 1553-SMU_STR_Acq Connect	Interface Connection communicationSMU_STR	MIL STD 1553B P/F M1553B	PUS						





SENTINEL 3: OPERATION DESCRIPTORS (TM/TC)

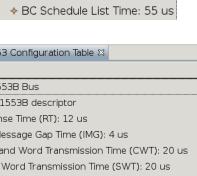
			S/C Mo	ode	Type [S	/A]	Freq.,	MIAT [Hz]	Packet		
r 🗆 RF	F Dock: SMU_RF_TM										
⊽ ♦	Operation: EncodeHKBatteryMonito	pring	Mode N	ormal	SYNCHR	ONOUS	0.0312	5			
		-							PUS		MU_TC RF Link Configuration Table 8
⊽ 🍐	Operation: EncodeHKPowerDistribu	ition	Mode N	ormal	SYNCHR	ONOUS	0.125				
	 Packet Standard 								PUS	▼ □	RF link
									FUS		
~ ~	Operation: EncodeHKPCDUDeployn	nent	Mode No	ormal	SYNCHR	ONOUS	1.0E-9			~	RF link descriptor
	 Packet Standard SMU TM PE Deals PUS Packets Descriptor 	M							PUS		Upload Occupation: 87.5%
⊽ ♦	SMU_TM RF Dock PUS Packets Descriptor			Cult Car		Data Cal	1 [D: +-1	O under and CD			Download Occupation: 100%
	▼ ■ RF Dock: SMU RF TM	Ser∨ice 1	уре	Sub-Ser	vice lype		гругеј	Overnead [B	yte] № Packets (💠 Total Upload Rate: 64,000 bps
- ♦											♦ Total Download Rate: 64,000 bp
Ť	 ✓ Operation: Encoder instatter yrionitorit ✓ + PUS Descriptor 	128		128		45.0		20.0			
	 Number Packets 								1		
~ ~											
	✓ ♦ PUS Descriptor	128		128		130.0		20.0			
⊽ ♦	Number Packets								1		
											III TM DE Link Configuration Table M
- ♦	▼ ♦ PUS Descriptor	128		128		48.0		20.0			IU_TM RF Link Configuration Table 🛚 [
• •	Number Packets								1		
										▼ ■ [RF link
		3		25		234.0		20.0		~ <	► RF link descriptor
	Number Packets								1		♦ Upload Occupation: 100%
											· · ·
		3		25		61.0		20.0			 Download Occupation: 85%
	 Number Packets A Operation: EncodeDIACDCDI Monitori 								1		♦ Total Upload Rate: 1,000,000 bps
	 Operation: EncodeDIAGPCDUMonitori OPUS Descriptor 	128		128		42.0		20.0			💠 Total Download Rate: 1,000,000 b
	 POS Descriptor Number Packets 	120		120		-2.0		23.0	1		
	✓ Administer Frackets ✓ ♦ Operation: EncodeSTRHKHealthStatus								-		
	 ✓ Operation: Encoded in a like lead istated ✓ ♦ PUS Descriptor 	3		25		140.0		20.0			
	 Number Packets 								1		

ThalesAlenia



SENTINEL 3: OPERATION DESCRIPTORS (1553)

	S/C Mode	Type [S/A]	Freq./MIAT [Hz	Packet S	tanda Minor Frame	Slot	RT Sub-Address		
♦ MIL STD 1553 BC Dock: BC1553BD	D							III SMU 1553 Bus Controller Config	
	Mode Normal	SYNCHRONOUS	1.0						ura
Packet Standard				PUS					
▽ 💠 1553 Communication Profile								✓ ♦ MIL_STD_1553B BC Dock	
♦ Minor Frame/Slot					1	1	10	V IL STD 1553B BC Descript	or
♦ Minor Frame/Slot					2	1	10	♦ BC Start Time: 21 us	
✓ ♦ Operation: SetDORISTime	Mode Normal	SYNCHRONOUS	1.0						
A Packet Standard				PUS				♦ BC Stop Time: 19 us	
						_		♦ BC Schedule List Time: 55	us
♦ Minor Frame/Slot STR 1553 Remote Terminal PUS * * STR 1553 Remote Terminal PUS	S Packets Desc	riptor 🛚			1	2	14		
		1	Type Data Fie	ld [Byte]	Overhead [Byte	1 Nº Pa	ckets (Optional)	■ SMU Bus 1553 Configuration Table 🛛	
✓ ♦ MIL STD 1553 RT Dock: RT15.						1 1 1 1 1 4			
✓ ♦ Operation: GetSTRStatus1								✓ ♣ MIL STD 1553B Bus	
✓ ♦ PUS Descriptor	1	1	24.0		20.0			 ✓ MIL STD 1555B bds ✓ ♦ MIL STD 1553B descriptor 	
 V V OS Descriptor Number Packets 		1	24.0		20.0	1		 Response Time (RT): 12 us 	
						1		 Inter-Message Gap Time (IMG): 4 us 	
▼ ♦ Operation: GetSTRStatus2								 Command Word Transmission Time (CWT). 	20.11
✓ ♦ PUS Descriptor	1	1	64.0		20.0			 Status Word Transmission Time (SWT): 20 	
Number Packets						1		 Data Word Transmission Time (DWT): 16 u 	
									5
	1	1	18.0		20.0			Maximum Throughput: 1,000,000 bps	
Number Packets						1		Bus Maximum Margin: 50% Major Exempt (only provided for fine grained)	
▼ ♦ Operation: GetSTRATM1								 Major Frame (only needed for fine-grained) Minor Frame (only needed for fine grained) 	
✓ ♦ PUS Descriptor	1	1	64.0		20.0			Minor Frame (only needed for fine-grained	bus a
Number Packets						1			
	1	1	64.0		20.0				
♦ Number Packets						1			
▼ ♦ Operation: GetSTRATM3									
✓ ♦ PUS Descriptor	1	1	64.0		20.0				
 Number Packets 						1			
✓ ♦ Operation: GetSTRATM4						-			
 ✓ Operation: GetShocking ✓ ♦ PUS Descriptor 	1	1	64.0		20.0				
	-	-	04.0		20.0	1			
Number Packets		1				1			_



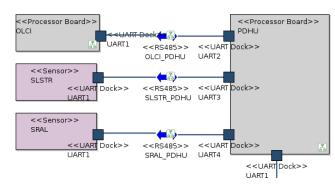
- /ord Transmission Time (DWT): 16 us
- m Throughput: 1,000,000 bps
- ximum Margin: 50%
- ame (only needed for fine-grained bus analysis):
- rame (only needed for fine-grained bus analysis):





SENTINEL 3: OPERATION DESCRIPTORS (UART)

	S/C Mode	Type [S/A]	Freq./MIAT [Hz]	Packet Standard
▼ ♦ UART Dock: UART1				
	Mode Normal	SYNCHRONOUS	22.7272	
♦ Packet Standard				PUS
	Mode Normal	SYNCHRONOUS	22.7272	
Packet Standard				PUS
▼ ♦ Event: AckEvent	Mode Normal	ASYNCHRONOUS	22.727274	
♦ Packet Standard				PUS



	Service Type	Sub-Service Type	Data Field [Byte]	Overhead [Byte]	Nº Packets
▼ ♦ UART Dock: UART1					
✓ ◆ PUS Descriptor	1	1	33582.0	20.0	
Number Packets					5
	1	1	35490.0	20.0	
Number Packets					10
▼ ♦ Event: AckEvent					
▼ ◆ PUS Descriptor	1	1	2.0	20.0	
Number Packets					1

- ▽ 🔶 RS-485
 - - ♦ Data Latency: 1.02 ns
 - Maximum Throughput: 50 Mbps
 - Maximum Load Margin: 50%

