

POLITECNICO DI TORINO

SCUOLA DI DOTTORATO

Dottorato in Electronics and Telecommunication Engineering – XXV  
ciclo

Tesi di Dottorato

# Onboard Communication Systems for Low Cost Small Satellites



Muhammad Rizwan Mughal

**Tutore**

Prof. Leonardo M. Reyneri

**Coordinatore del corso di dottorato**

Prof. Ivo Montrosset

February 2014



*Dedicated to my late  
father who was the real  
role model for me.*



# Acknowledgements

I will start *"In the name of Allah, the Most Gracious and the Most Merciful"*.

This work would not have realized without the utmost support and cooperation of several people.

I would like to thank my father who was a role model for me and without his continuous encouragement, I would not even think of where I am at the moment.

I would like to thank Prof. Leonardo M. Reyneri, for being my advisor during this work. I must say that I was quite lucky to work with Prof. Reyneri whose dedication to work encouraged me all the times. It is obvious that this thesis would not have been possible without his continuous support. I should thank him for the freedom I had during this work and for the patience he had in helping me.

I feel honored to be part of AraMiS team and would like to thank all the members for their help during these three years, in particular Prof. Claudio Sansoe and Prof. Dante Del Corso. A special thanks go to Danilo Roascio for the continuous technical, administrative and linguistic support during my stay. I would also thank my colleagues Mr. Anwar Ali and Mr. Haider Ali for their continuous support, encouragement and team work.

I also thank all those friends whose company made my time quite pleasant. During these times, I had suffered some hard times specially by the death of my beloved father and uncle. That was in-fact the hardest time of my life. I must thank my brother, Imran Mughal, who was present with the family in all those tough periods and for his continuous support in those difficult times.

Last but not the least, I should thank my mother who stood firm in all the hard times which gave me strength to achieve this milestone.

Torino, Feb 10<sup>th</sup>, 2014

Muhammad Rizwan Mughal



# Summary

The development of a large number of Nano and Pico-satellite missions, with spacecrafts of mass lower than 10 kg and 1 kg respectively, started in the beginning of this century due to the availability of low-cost piggyback launch opportunities. Such small satellites are usually built using commercially available electronic components that are not qualified for the space environment. This approach reduces the total cost of the satellite missions but at the expense of design effort which is needed not to compromise the reliability of the designed spacecrafts. One of the foremost design efforts in this regard is the design re-use method which extends the cost reduction to the system level, and helps in simplifying the development cycle for a space mission.

The on-board communication subsystem consist of critical set of elements common to every mission, and therefore is not exempt from such a design philosophy. The on-board networks, on-board transceivers, and the protocols are all critical elements for a spacecraft mission and, at the same time, some of the most specialized and complex ones. Innovative data communication systems are therefore desirable for the future space missions.

The size of the satellites keeps reducing as the time progresses, therefore the harness mass and complexity inside the satellite becomes a prime challenge. An innovative approach to smart harness is therefore necessary which reduces the wiring harness for intra-satellite communication.

This thesis copes with several problems related to spacecraft subsystem development, integration and testing and proposes some solutions that can help in both keeping system development and production cost low while still achieving good performances.

Chapter 1 starts with the design goals of the work and introduction to the Modular Architecture of Small Satellites ([AraMiS](#)) project. The biggest design goals of space systems of current era are the cost, time and complexity issues. Modularity and cost-sharing between multiple missions will appear as optimal solutions for reducing development costs, while the use of commercial components (COTS) will be presented as a way to simplify procurement and further lower system cost

In Chapter 2, the smart harness approach is proposed which reduces the traditional harness complexities inside the small spacecrafts. The chapter focuses on the

design of small spacecrafts which are completely modular and flexible. Modularity at mechanical, electrical and testing level will be discussed in this chapter.

Chapter 3 addresses the complete life cycle of a subsystem module i.e. from conception to the final design and testing. The module life cycle uses a variety of Unified Modelling Language (UML) diagrams to fulfill different design stages.

Chapter 4 proposes different types of spacecraft configurations based on smart harness approaches including physical module based, satellite on demand and reusable design configurations. A design trade-off is also performed for these configurations.

Chapter 6 proposes the design technique of physical module based spacecraft configuration which is based on physical plug and play connectors and logical slots for the subsystem modules. A honeycomb based tile is discussed in this chapter which is used for larger and more demanding spacecraft structures.

In Chapter 7, the requirement of data communication across different subsystems of the spacecrafts are described. The use cases have been discussed and the implementation rules have been defined in this Chapter.

Chapters 8,9 and 10 focus on module design for intra-satellite communication purposes. The modules have been designed for wired as well as wireless data communication. The wired solution is based on on-board data bus module for inter-tile data communication. Wireless solutions included both optical and radio frequency based solutions. The optical module has been designed for optical free space as well as glass fiber based communication purposes. The comparison between theoretical and practical results has been made. The radio frequency based module is based on commercial module and simpliciTI protocol stack.

In Chapter 11, the functional testing of modules, tiles and whole satellites is discussed. The testing scheme of functional test board is also highlighted in this chapter.



# Contents

<b>Acknowledgements</b>	V
<b>Summary</b>	VII
<b>1 Introduction</b>	3
1.1 Small Satellites . . . . .	3
1.2 Problem Statement . . . . .	4
1.3 Literature Review . . . . .	5
1.3.1 Picpot . . . . .	8
1.3.2 AraMiS . . . . .	8
1.4 Subsystem Organization . . . . .	9
<b>2 Smart Harness</b>	13
2.1 Introduction . . . . .	13
2.2 Modularity at Mechanical Level . . . . .	15
2.2.1 Tile Dimensions . . . . .	15
2.2.2 Subsystem Modules . . . . .	17
2.2.3 Extended Modules for Larger Subsystems . . . . .	20
2.2.4 Logical Modules . . . . .	21
2.2.5 Modules Electrical Interface . . . . .	22
2.3 Modularity at Protocol Level . . . . .	22
2.3.1 Supported Communication Protocols . . . . .	25
2.4 Modularity at Hardware/Software Level . . . . .	26
2.4.1 Modular Software Drivers . . . . .	27
2.4.2 Radiation Hardening . . . . .	29
2.5 Conclusion . . . . .	30
<b>3 Module Life Cycle</b>	33
3.1 Introduction . . . . .	33
3.2 UML Diagrams . . . . .	35
3.2.1 Specifications: Use Case Diagrams . . . . .	35

3.2.2	Architecture and Design: Class Diagrams . . . . .	37
3.2.3	Verification:Sequence Diagrams . . . . .	37
3.3	Design Flow . . . . .	38
3.4	Conclusion . . . . .	39
<b>4</b>	<b>Satellite Design</b>	<b>41</b>
4.1	Introduction . . . . .	41
4.2	Design Flow . . . . .	41
4.2.1	Physical Module Based Configuration . . . . .	42
4.2.2	Satellite on Demand Configuration . . . . .	42
4.2.3	Reusable Design Configuration . . . . .	42
<b>5</b>	<b>Satellite Design: Satellite on Demand</b>	<b>45</b>
5.1	Introduction . . . . .	45
5.2	AraMiS-C1 . . . . .	45
5.2.1	ISIS-1U CubeSat Structure . . . . .	46
5.2.2	1B8 CubePMT . . . . .	47
5.2.3	1B9 CubeTCT . . . . .	48
5.2.4	Payload . . . . .	50
5.2.5	System Level Integration . . . . .	50
<b>6</b>	<b>Satellite Design: Physical Module Based Configuration</b>	<b>53</b>
6.1	Introduction . . . . .	53
6.2	Honeycomb Tile . . . . .	54
6.2.1	Mechanical Structure . . . . .	54
6.3	Design Modularity . . . . .	55
6.4	Power Supply Converters . . . . .	57
6.5	Distributed Power Management . . . . .	60
6.6	Housekeeping Management of Honeycomb Tile . . . . .	67
6.7	Conclusion . . . . .	69
<b>7</b>	<b>Intra-Satellite Communication</b>	<b>71</b>
7.1	Introduction . . . . .	71
7.2	Requirements . . . . .	71
7.3	1B45 Usecases . . . . .	72
7.3.1	Configuration and status management . . . . .	73
7.3.2	Housekeeping Management . . . . .	74
7.3.3	User defined messages and commands . . . . .	75
7.3.4	Supervision and Emergency Recovery . . . . .	75
7.4	Tile Processor . . . . .	75
7.5	Implementation . . . . .	78

<b>8</b>	<b>Module Design: 1B411 On-board Data Bus Module</b>	<b>79</b>
8.1	Introduction . . . . .	79
8.2	Cross Bus . . . . .	80
8.3	Design of Interface Module . . . . .	81
8.3.1	Transmitter . . . . .	82
8.3.2	Receiver . . . . .	83
8.3.3	Coupling . . . . .	84
8.3.4	Snubber to Supress Ringing . . . . .	84
8.4	Fault Analysis . . . . .	85
<b>9</b>	<b>Module Design: Optical Module</b>	<b>91</b>
9.1	Introduction . . . . .	91
9.2	Functional Description . . . . .	93
9.2.1	Transmitter Design . . . . .	94
9.2.2	Receiver Design . . . . .	95
9.3	Radiation testing of selected optical components . . . . .	100
9.4	Free Space Communication Architecture . . . . .	102
9.5	Glass Fiber Based Communication Architecture . . . . .	104
<b>10</b>	<b>Module Design: On-Board Radio Frequency Module</b>	<b>111</b>
10.1	Introduction . . . . .	111
10.2	Overview . . . . .	111
10.3	Hardware Platform . . . . .	112
10.3.1	CC 1110/2510 . . . . .	112
10.4	Software Platform: simpliciTI . . . . .	112
10.4.1	API . . . . .	113
10.4.2	Packet Sniffer . . . . .	114
10.5	Implementation . . . . .	114
<b>11</b>	<b>1C25 Functional Testing</b>	<b>117</b>
11.1	Introduction . . . . .	117
11.2	Functional Test Board . . . . .	118
11.3	1C25 Use Case Diagrams . . . . .	119
11.3.1	1C25 Use Cases . . . . .	119
11.3.2	Configuration Use Cases . . . . .	120
11.3.3	Configuration and Status Test Use Cases . . . . .	124
11.4	Testing of Functional Test Board . . . . .	127
11.4.1	Regulators . . . . .	127
11.4.2	Processor . . . . .	127
11.4.3	Crystal Oscillators . . . . .	127
11.4.4	House Keeping Sensors . . . . .	129

11.4.5 Module Testing . . . . .	130
<b>12 Conclusion</b>	<b>133</b>
<b>Bibliography</b>	<b>135</b>
<b>Appendix A 1B81 Power Management Tile Honeycomb Schematics</b>	<b>143</b>
<b>Appendix B 1411 On-Board Data Bus Module Schematics</b>	<b>155</b>
<b>Appendix C List of Acronymes</b>	<b>159</b>

# List of Tables

- 2.1 Electrical Interface between AraMiS Modules and Tile Processor . . . 23
- 2.2 Interface of AraMiS module signals. For each connector (A-H), corresponding number of channels (in digits) are shown . . . . . 24
- 2.3 Interface of AraMiS module signals with the MSP430F5438A . . . . . 24
- 2.4 Interface of AraMiS module signals with the MSP430F5438A . . . . . 25
- 2.5 Hardening of data using redundancy . . . . . 30
- 8.1 Fault Analysis . . . . . 88
- 9.1 Theoretical and practical received currents for certain input radiated power for guided light communication. . . . . 107

# List of Figures

1.1	An undesirable spacecraft violating the design philosophy . . . . .	4
1.2	A relatively desirable spacecraft structure which is simple, faster and better . . . . .	5
1.3	A top level UML diagram for AraMiS architecture . . . . .	10
2.1	Drawing of single tile different spacecraft configurations by arranging tiles in certain configurations . . . . .	14
2.2	2 Mechanical drawing of single tile (a) External view (b) Internal view	16
2.3	Photograph of 1U CubeSat compatible tile (a) External view (b) Internal view (c) arranged in 1U configuration . . . . .	16
2.4	Photograph of Honeycomb Tile . . . . .	17
2.5	Generic tile architecture showing physical and logical connectors . . . .	18
2.6	Dimensions and details of single, double and quadruple modules (seen from component side, towards satellite interior) . . . . .	19
2.7	Fabricated modules (a) single module: component side view (b) single module: connector side view (c) double module: component side view (d) double module:connector side view . . . . .	20
2.8	Layout of Extended modules . . . . .	21
2.9	Signal scheme of module slots . . . . .	22
2.10	UML class diagram view of AraMiS modular drivers . . . . .	27
2.11	Three UML classes which are used to implement Triple Modular Redundancy (TMR) into Software (SW) programs in a transparent way. The classes offer more operations than shown here, where they have been truncated to increase visibility . . . . .	31
3.1	Use case diagram of magnetic attitude subsystem module . . . . .	36
3.2	Design flow of magnetic attitude subsystem module . . . . .	39
4.1	Trade off analysis of using modules in different spacecraft configurations	43
5.1	ISIS Frame for AraMiS-C1 . . . . .	47
5.2	Cube Telecommunication Tile . . . . .	48
5.3	System level integration of different tiles through inter tile communication and power distribution bus . . . . .	51

5.4	AraMiS Satellite on Demand configuration fabricated usign 1B8 CubePMT and 1B9 CubeTCT . . . . .	52
6.1	Honeycomb Tile (a) A honeycomb core C is a sandwich structure between two face sheets B (b) photograph view of designed tile . . . .	54
6.2	Module interfaced with honeycomb tile. Embedded power and data harness is available on slots and connectors and ultimately to all subsystems . . . . .	55
6.3	Power management class diagram for honeycomb tile showing internal subsystems. Classes have been shown in different color based on their type. . . . .	57
6.4	Block diagram of power supply converters . . . . .	58
6.5	Block diagram of Buck converter with MPPT. . . . .	59
6.6	Simulation results of MPPT Buck convertor. . . . .	60
6.7	Block diagram of Buck converter with MPPT. . . . .	61
6.8	Honeycomb tiles assembled in hexagonal shaped spacecraft with optical telescope payload (mock-up) . . . . .	62
6.9	(a) Power management actors controlled by central power management (b) Static I-V output characteristics of power management subsystem . . . . .	63
6.10	Illustration of different tiles connected with power distribution bus. Solar panel, storage and load for each connected tile is shown. . . . .	65
6.11	Simulation analysis of proposed power bus. . . . .	66
6.12	Housekeeping use case diagram of honeycomb tile showing different use cases and actors. . . . .	67
7.1	Possible Types of Faults . . . . .	74
7.2	Illustration of Basic Protocol . . . . .	76
8.1	Interconnecting two tiles in series (a) Typical (b) Proposed approach	80
8.2	Cross bus module (a) Single (b) Network of six . . . . .	81
8.3	Design of Bus Interface circuit . . . . .	82
8.4	Receiver inputs . . . . .	83
8.5	Implemented Bus Interface Module (a) Test circuit (b) Final Module	85
8.6	Possible Types of Faults . . . . .	86
8.7	Block Diagram of connecting six modules . . . . .	87
8.8	Block Diagram of connecting six modules . . . . .	88
8.9	Transmit and Receive of Random Bit Sequence . . . . .	89
9.1	Communication system overview . . . . .	93
9.2	Transmitter block diagram . . . . .	95
9.3	Receiver block diagram . . . . .	96
9.4	Frequency Response of Receiver Stages . . . . .	98
9.5	Photodiode current Vs Received Voltage . . . . .	99
9.6	LED radiant intensity Vs angular displacement . . . . .	101

9.7	Photodiode sensitivity at various TID levels . . . . .	101
9.8	Relative Intensity vs Angular Displacement, courtesy Vishay . . . . .	103
9.9	Received photocurrent for certain distances . . . . .	104
9.10	Reception at different positions due to transmitters . . . . .	105
9.11	Optical light guidance in the Fiber . . . . .	105
9.12	Architecture of glass fiber based communication showing tiles . . . . .	109
10.1	simpliciTI Class Diagram . . . . .	113
10.2	Packet Sniffer showing a string transmitted between two modules . . . . .	115
10.3	average RSSI for certain transmit power levels . . . . .	116
11.1	Software setup of functional testing . . . . .	118
11.2	Test side of functional testing . . . . .	119
11.3	Functional test board with modules under tests . . . . .	120
11.4	Functional testing use case diagram . . . . .	121
11.5	Configuration use case diagram . . . . .	122
11.6	Module Description File . . . . .	123
11.7	A tile description file . . . . .	124
11.8	A satellite description file . . . . .	125
11.9	A satellite description file . . . . .	126
A.1	Functional test class diagram . . . . .	144
A.2	Power Management Tile Honeycomb Schematics . . . . .	145
A.3	Tile Processor Schematics . . . . .	146
A.4	Power Converters Schematics . . . . .	147
A.5	Mixed Regulator . . . . .	148
A.6	3.3V Linear Regulator . . . . .	149
A.7	Switching Regulator . . . . .	150
A.8	Maximum power point tracker block . . . . .	151
A.9	Comparator Hysteresis . . . . .	152
A.10	Buck Converter . . . . .	153
B.1	OBDB Module . . . . .	156
B.2	OBDB Module Schematics . . . . .	157







# Chapter 1

## Introduction

### 1.1 Small Satellites

The market for small satellites has considerably grown since the beginning of this century. It has become possible due to the availability of low cost launch vectors. This cost reduction has made it feasible for universities and small industries to enter the satellite market. In this regard the first real NanoSatellite termed CubeSat was developed by California Polytechnic State University in collaboration with Stanford University. Satellites based on CubeSat standard have made it possible for potential clients to buy and assemble their own satellites from a kit.

Practically, any artificial satellite of low mass and low volume can be considered as a small satellite. However, the definition can be enlarged to any system designed with the small satellite philosophy. This may include features such as commercial off-the-shelf components, modular systems, less redundancy, open sourcing, incremental missions, etc.

Over the past few years, small satellites have changed the landscape of space exploration. They facilitate quicker, cost effective and reliable access to space. This provides a potential opportunity to have smaller PROJECT GROUPS and encourages new actors to develop their capabilities in the space domain. Small satellites are encouraging spacecrafts to test and try novel methods and technologies (E.g. open source hardware and software, formation flying), which might not be under the purview of large scale satellites. This remains the reasons as to why small satellites have been considered a disruptive technology by numerous space mission experts.

Small satellite programs are particularly attractive since they are "affordable". There shall be no surprises in the near future, if more and more developing countries, groups from the academic world or even small teams of space enthusiasts develop their own space mission based on small satellites. The small satellite platform is catering to new actors such as developing countries, students, and amateurs.

## 1.2 Problem Statement

One of the biggest challenges to nowadays space industry is to design space systems which are cheaper, faster and better. Significant reduction in the design and mission costs are required in order for a space system to be cheaper. The design and implementation time needs to be quite lower for a space system to be faster. A space system is considered to be better if it achieves most of the mission targets. The very first question that comes into mind is that how to achieve this Cheaper-Faster-Better philosophy? The possible options to achieve this philosophy are to

- Scale down the satellites
- Simplify their design
- Reduce Power and data harness
- Simplify their testing

In order to meet the above mentioned requirements, a lot of design effort is needed. Fig.1.1 shows a satellite structure which is designed by not taking into account all these parameters. The visual inspection of the figure shows a lot of wiring harness spread all around the structure which makes this design quite undesirable. On

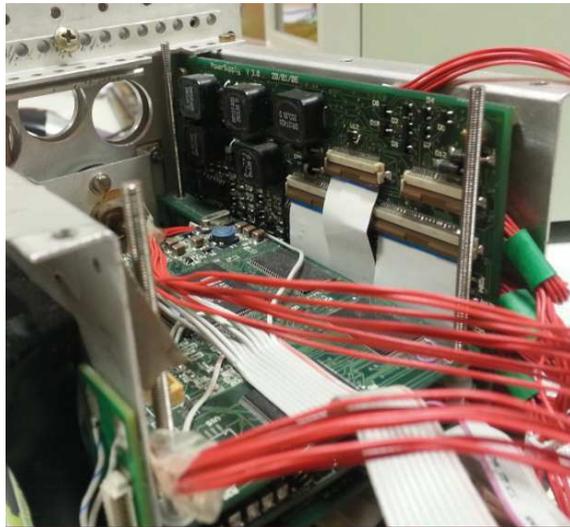


Figure 1.1. An undesirable spacecraft violating the design philosophy

the the other hand, Fig.1.2 shows a small spacecraft with minimum wiring harness and other complexities , simplifying the design and is a highly desirable system. The subsequent chapters will emphasize on the design approach which significantly achieved the Cheaper-Faster-Better philosophy.



Figure 1.2. A relatively desirable spacecraft structure which is simple, faster and better

## 1.3 Literature Review

The main scope of this thesis is to simplify the design of small satellites in general and intra-satellite communication in particular. The typical wired communication creates a lot of harness mass and interfacing complexities. Therefore, some solutions in literature were reviewed for possible use or improvement. In order to adopt the Cheaper-Better-Faster design philosophy, the plug and Play design has attracted the attention of industry and academia.

The conventional methods in the satellite design have been unable to adapt the rapid pace of now a days space industry needs. In particular, the rapid response satellite concept challenges the traditional satellite design methods. Rapid response satellites need to be designed rapidly in response to the major disasters such as flood monitoring, earthquake disaster monitoring, and many others. Therefore new design concepts and methods in small satellite design are necessary.

The Plug and play(PnP)is a class of technology facilitating the discovery of a hardware component in a system without the need for physical device configuration or user intervention in resolving resource conflicts. PnP of satellite system is an important way to achieve rapid space response, and PnP has a great important significance in rapid system reconfiguration, maintenance in orbit and function extension of the rapid response satellite. The Naval Research Laboratory and MIT have embarked on a series of rapidly responsive Tactical Satellite (TacSat), which employs the technology of modular satellite platform and standardized interface for payload and so on. It is shown that they can produce and launch a satellite in a few weeks or months, at the same time the satellite can adapt to a lot of different payloads. Research in the design of plug and play satellites is carried on a number of research institutes.

The Plug-and-Play Satellite (PnPSat) program [1–3] by Air Force Research Laboratory (AFRL) is designed for rapid and modular designs and the satellites can be produced automatically by using self-describing mechanisms.

AeroAstro,s efforts to enable responsive space are gathered under the name SMARTBus. SMARTBus incorporates a set of mechanical, electrical, and logical (interoperability / software) standards for the interaction of modular spacecraft subsystems [4].

A modular satellite concept was proposed by Lyke [5] in 2012. It can form a complex satellite platform quickly and reliably by arranging a number of open common blocks. Each component can be similarly thought of a black box. These components, all with standardized interface, can be easily combined to form a Monarch satellite system. The host of the Monarch satellite operates a variety of open-source apps to control the components. These apps are similar to apps on the iOS or Android platform.

A number of wired architectures for spacecraft data communication have also been proposed in the literature. At present, many fault-tolerant bus systems have been developed, realized and used for space applications [6–8]. Point-to-point communication structures are very reliable: a single fault or a node which starts generating data on the communication medium does not cause problems for all the system [9], but it results in high costs, volume and weight due to the number of wires. Data transmission using bus systems is cheaper, but also more sensitive to faults.

Several different standards for short range communications have been defined. Wireless communications employ standards such as Infra-red Data Association (IrDA), Blue-Tooth, Zig-Bee etc., and wired communications employ standards such as Universal Asynchronous Receive Transmit (UART), UART IrDA line mode, Serial Peripheral Interface (SPI), Inter Integrated Circuit (I2C) etc. [10, 11]. Each of them has specific advantages and drawbacks. In space environment, the bus needs to be fault tolerant and operationally reliable.

Data communication using wireless approach on board the small satellites is still in the early stages as it is prone to much external interferences. In [12–15], the authors describe the analysis of using Optical wireless communication bus and discuss the use of this standard to exchange data between the sub-systems of a satellite.

Wired communication uses physical wires to transmit data across different sub-systems. Using physical wires requires the electronic signals to be transmitted over a metal conductor. Currently, this is the most reliable way of transmitting/receiving data onboard the satellite missions. Depending on the protocol, either differential signaling or single ended signaling scheme can be used with wired communication. A differential signal [16] represents difference between two physical quantities. In differential signaling, the signal value is the difference between individual voltages

whereas in single-ended signaling, ‘ground’ is used as voltage measurement reference. Differential signaling brings many advantages over single ended signaling such as tolerance to ground offsets, common mode rejection, resilience to outside electromagnetic interference and cross-talk, and canceling of the magnetic field of each conductor. Since differential signaling only provides physical mechanism to transmit bits across the medium, any user defined encoding scheme can be sent/received across the Low Voltage Differential Signalling (LVDS) link.

In [17], the Optical Wireless Communications have been proposed as a solution for partial or total replacement of conventional wired data transmission links inside the spacecraft. The technology is known generically as Optical Wireless Links for intra-Spacecraft communications (OWLS). A demonstration of several optical wireless networks for TM/TC distributed units has been developed inside a 1:1 scale Venus Express (VEX) mock-up under the framework of an ESA activity leaded by INTA(National Institute of Aerospace Technology of Spain).

In [13, 14], the authors describe analysis of using wireless communication bus and discuss the use of this standard to exchange data between the sub-systems of a satellite. The non-space grade commercial optical transceivers utilize infrared LEDs, which may suffer from displacement damage depending on the device’s construction [18, 19]. In addition, the IrDA transceivers are packaged in plastic packaging with the emitter/receiver pair. This plastic may also become less transmissive with increasing radiation dose. Keeping in view the above mentioned problems in commercial transceivers that have not been designed for space radiation environment, novel discrete transceiver for Infrared data communication (IrDA) inside small satellites has been designed using discrete Light Emitting Diode (LED)s and photo-diodes. Using discrete solution gives more freedom in wavelength selection and best components keeping in view space radiation problem.

Research towards the small spacecrafts started during the last decade. In 2001, Professor Robert Twiggs at Stanford University, USA, in collaboration with Professor Jordi Puig-Suari at California Polytechnic State University, have defined and implemented the standard for small satellite called CubeSat [20, 21]. The objective of the project was to provide a standard platform for the design of small satellites. A common deployer is used, significantly reducing cost and development time and enabling frequent launches. This allows multiple high schools, colleges, and universities from around the world to develop and launch their own satellites without having to interface directly with launch providers. The CubeSat identifies the standard for small satellite with dimensions 10 x10 x10 cm and a maximum mass of 1 kg, having a structure adapted to the launcher POD ( Pico-satellite Orbital Deployer ).

### 1.3.1 Picpot

The Department of Electronics & Telecommunication Engineering, in collaboration with the Department of Aerospace Engineering at Politecnico di Torino started work on the PICPOT (Small Cube of Politecnico di Torino) project in January 2004. The major objective of the project was to design, implement and launch experimental satellite into Low Earth Orbit (LEO) with the aims to

- Check the operation and reliability of COTS components in space
- Acquire and transmit to Earth Station several images and measurements carried into orbit by the on-board sensors
- Take Earth's surface images
- Encourage and create interdisciplinary activities to enhance the interaction and coordination of various departments, faculty, PhD students and other students of the Politecnico.

The PICPOT was a satellite of cubic structure with dimensions 13 x 13 x 13 cm and a mass of 2.5 kg. It was intended to be launched together with other university satellites by a DNEPR LV rocket in July 2006. Unfortunately a problem in the first stage of the carrier led to destruction of all satellites and our aim of going to space could not be accomplished.

Then afterwards, we started working on a completely modular and flexible architecture called [AraMiS](#).

### 1.3.2 AraMiS

[AraMiS](#) takes further the concept of small satellites by employing a true modular approach in the design of small spacecrafts. This design approach provides low-cost and high performance space missions with any desired spacecraft dimensions and configurations. The core of the design is the subsystem development on a number of small, flexible and powerful modules. These modules can be reused for multiple missions achieving significant reduction in the overall budget, development and testing time. One has just to reassemble the required subsystems to achieve the targeted specific mission. Design reuse is the rationale behind the AraMiS project. This architecture is intended for different satellite missions, from small systems weighing from 1kg to larger missions. Figure 2.1 depicts a number of configurations that show the potential capabilities of the proposed architecture. Modularity has been implemented in different ways. From the mechanical perspective, larger satellite structures can be conveniently realized by combining several small modular structures. The modularity concept has also been intended from electronic standpoint.

Most of the internal subsystems are developed in such a manner they can be composed together to enhance performance. One such example is the power management subsystem. In conventional missions: to get maximum solar power, solar cells are mounted on all the available surfaces but their number can be different in various missions, thus requiring to re-design each time this system. This new modular approach makes use of a standard module, as can be seen from in figure 2.1 which can be replicated many time to fit mission requirements.

Several schemes were applied in order to simplify the [AraMiS](#) architecture and

## 1.4 Subsystem Organization

The most part of the thesis uses a number of [UML](#) diagrams to emphasize different stages of the design. The top level [UML](#) diagram for the [AraMiS](#) architecture is shown in Fig.1.3

The *1A System Level Elements* contains all system level elements of the [AraMiS](#) project, including, among others:

- *Bk1A1\_Common\_Codes*: tile ID codes, telemetry and telecommand codes for 1B8 Power Management Tile and 1B9 Telecommunication Tile , error codes, common component material, etc.

The *1B Subsystem Elements* contain all the core subsystems of the spacecraft. They include

- *1B1 Power Management Subsystem* is in-charge of the power generation, storage, distribution and centralized power management for the spacecraft.
- *1B2 Attitude and Orbit Subsystem* consists of inertial attitude and magnetic attitude subsystems for attitude control. Different types of algorithms are used for the orbit control.
- *1B3 TT&C Telecommunication Subsystem* responsible for the telemetry, telecommand and telecommunication links for the spacecraft.
- *1B4 On Board Data Handling Subsystem* emphasizes at physical and protocol levels of data communication and data handling subsystem
- *1B5 Payload Subsystem* manages all the mission specific payloads
- *1B6 Mechanical Subsystem* emphasises the mechanical architecture for the satellite. This includes assembling the parts, thermal analysis, panel mechanics and harness

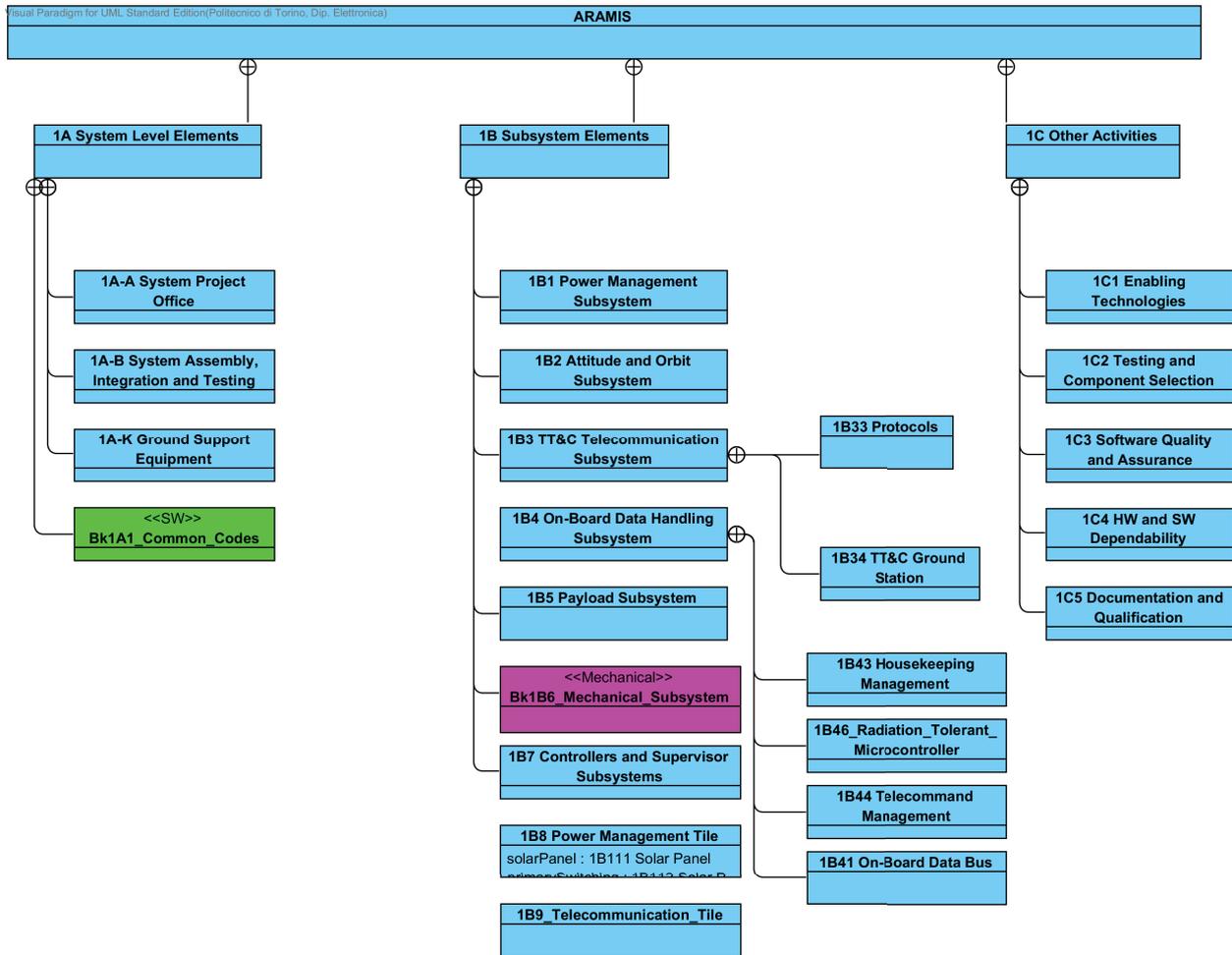


Figure 1.3. A top level UML diagram for AraMiS architecture

- *1B7 Controllers and Supervisor Subsystems* consists of all the subsystems and components aimed at monitoring the operation of the satellite, detecting faults and taking actions accordingly.
- *1B8 Power Management Tile* is a complete panel body, which houses a 1B1 Power Management Subsystem, a 1B2 Attitude and Orbit Subsystems, a 1B6 Mechanical Subsystem and some other common objects
- *1B9 Telecommunication Tile* is a complete panel body, which houses a 1B1 Power Management Subsystem, 1B3 TT&C Telecommunication Subsystem, a 1B6 Mechanical Subsystem and some other common objects

The *1C Other Activities* consists of component selection, functional testing, software and quality assurance, documentation and qualification etc.

The major part of this thesis consists of 1B4, 1B8 and 1C subsystems.



# Chapter 2

## Smart Harness

### 2.1 Introduction

Embedded harness hosts power and data harness subsystems which are very vital elements and distribute electrical power and signals all around the spacecraft subsystems and payloads [22]. Smart harness technique embeds power, data and radio frequency harness with additional signal processing capabilities [23]. The chapter proposes an innovative approach to smart harness in the design of modular small satellites. Modular or module based design approach is based on dividing a system into smaller parts or modules that can be created independently and assembled together to achieve the desired performance of the complete system. This design approach is in general low cost because the design, qualification and test cost is shared among multiple modules. The modules are developed in parallel rather than typical system developed using serial approach and results in the reduction of design time. Modular design combines the advantages of standardization with those of customization [5, 24-27]. Upon fabrication, the modules are rapidly combined into a custom printed circuit board layout and the job of prototype testing can begin. In addition, the modular design significantly reduces the cost of the overall system because the cost is shared among a number of modules that can be reused in the system many times. The AraMiS architecture is based on standardized panel bodies or blocks called tiles which are used to build small satellites according to the specific requirements [28, 29]. Each tile offers a power and data standardized interface with mechanical support for small subsystems. The outer tiles are of two types namely power management tile and telecommunication tile. The power management tile uses high level of integration and compact stack of different materials and resins. It is composed mostly of solar panels, rechargeable batteries, a battery charger and an housekeeping module to keep track of telemetry data inside the tile and an active magnetic and inertial control system. An appropriate number of such tiles is placed

around a cubic or any other desired shape and represents a pre-designed and pre-assembled modular architecture. Fig.2.1 shows the drawing of a single tile and some possible satellite configurations by arranging a number of tiles. Telecommunication

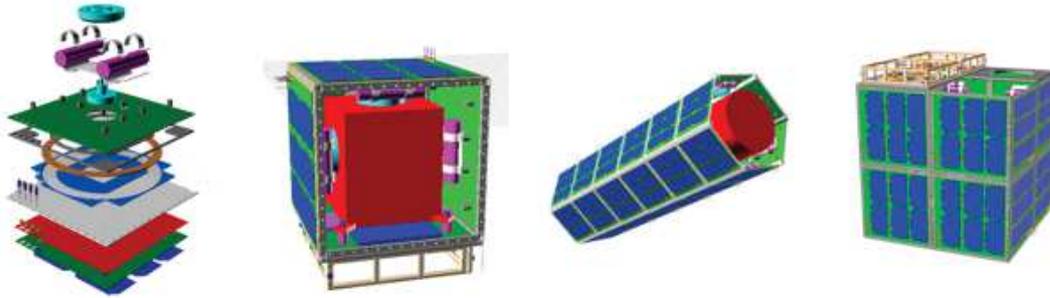


Figure 2.1. Drawing of single tile different spacecraft configurations by arranging tiles in certain configurations

tiles are composed of microcontroller-based programmable transceiver, 437MHz and 2.4GHz modem, power amplifiers (for transmission) and low-noise amplifiers (for reception) and an antenna system. This kind of tile is placed on one of the faces of the satellite, preferably pointing to ground, and manages the exchange of data and commands to/from ground stations. The tiles are modular and scalable at mechanical, protocol and hardware/software levels. By modularity at mechanical level we mean that the tiles and subsystem modules can be fabricated in any desired dimension and combined together in any desired satellite configuration. Scalability at electrical level is obtained by using the symmetrical electrical signal scheme for every connected module. Modularity at protocol level is achieved by using a basic protocol consisting of multiple communication protocols, thus giving flexibility for onboard communications. Every subsystem is either embedded onto the tile or housed in daughter boards which are attached to the tile via commercial spring loaded connectors. The tiles having pluggable connectors can either be used as a test bed for testing of individual subsystem modules or used as a motherboard in flat sat configuration. The subsystems support multiple communication protocols using smart mapping mechanism. A one wire memory for storing configuration and calibration data is associated to each sensor, including the analog ones. Each tile is therefore a system made of pluggable subsystems which can be tailored to mission-specific needs without re-design. The tile software is also modular and allows a quick adaptation to the specific subsystem. The basic software for the Central Processing Unit (CPU) is properly hardened to guarantee a high level of radiation tolerance. The chapter is organized according to the following sequence. Section2.2 describes the modularity at thermo-mechanical level for tile and corresponding pluggable modules that are

placed on each tile. Section 2.3 discusses the modularity at protocol level. Section 2.4 describes the UML based hardware and software co-design of acAraMiS architecture using a UML approach.

## 2.2 Modularity at Mechanical Level

### 2.2.1 Tile Dimensions

Modularity at mechanical level is achieved by fabricating the tiles with different mechanical size and dimensions and assembling them in different mission dependent configurations. The tiles have been fabricated using different technologies including Aluminium based, Printed Circuit Board (PCB) only and honeycomb structures. The structure and technology of each tile has been discussed which makes it easier to use in specific missions.

#### Single Size Al Structure

A 16.5x16.5 cm<sup>2</sup> tile with 1.6mm thick monolithic Aluminum structure, for cheaper and smaller structures. Fig. 2.2 shows internal and external view of a single tile containing standardized interconnection points for pluggable modules. The mechanical subsystem is made of chromate conversion coated alodined Al frame which bears solar cells, batteries, reaction wheel, motor and all the electronic elements. This Aluminium frame, with its high strength to weight ratio, corrosion resistance and radiation protection characteristics is normally used in most of avionic systems including satellite structures [30]. The single size Al tiles can be used in any satellite configuration for LEO orbit, ensuring proper temperature and radiation levels for using normal Components off the shelf (COTS) devices.

#### CubeSat Standard Tile

An 8.25 x 9.8cm<sup>2</sup> tile, with all electronic components integrated and compatible with 1U, 2U, 3U...6U CubeSat dimensions [31]. This structure is designed in accordance with standard 1U cubic structure of 10 x 10 x 10cm<sup>3</sup> dimensions. Each power management tile has Electric Power Supply (EPS) and Attitude Determination and Control Subsystem (ADCS). Solar panels are mounted on the external side and all necessary subsystems on the internal side of the printed circuit board. The technology allows developing CubeSat of any size in a much faster and more reliable way than traditional CubeSats leaving much more empty space for payload and batteries. Fig. 2.3 shows the external and internal view of CubeSat standard tile and arrangement of tiles in a cube configuration.

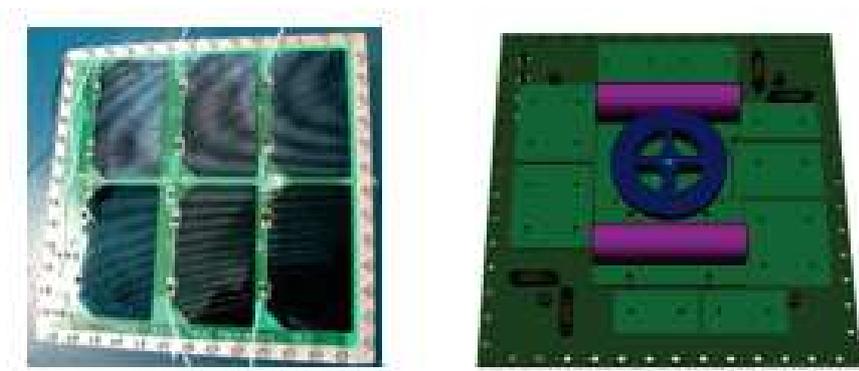


Figure 2.2. 2 Mechanical drawing of single tile (a) External view (b) Internal view



Figure 2.3. Photograph of 1U CubeSat compatible tile (a) External view (b) Internal view (c) arranged in 1U configuration

Out of the total cube area ( $10 \times 10 \text{ cm}^2$ ), each of the four tile PCBs occupy  $0.3 \text{ cm}^2$  including passive components leaving  $9.4 \times 9.4 \text{ cm}^2$  empty area for standard boards and batteries. The standard PC/104 boards are integrated by using ad hoc washers to marginally extend one face of the CubeSat. The only large components are power inductor and gyroscope having maximum height  $0.7 \text{ cm}$  and  $0.5 \text{ cm}$  respectively. They have been placed on the tiles in such a way that upon mounting, they fit between two adjacent PC/104 boards. Most of the tile connectors shown in the photograph are for testing and debugging purposes and are not mounted in the final version. Fig.2.3(c) indicates the tiles integrated in 1U structure with standard CubeSat boards mounted.

## Honeycomb Tile

A 16.5x33.0 cm<sup>2</sup> tile, with 10mm thick honeycomb structure for more rigid and larger structures [32]. Fig. 2.4 shows a photograph of honeycomb tile. Each tile hosts 14

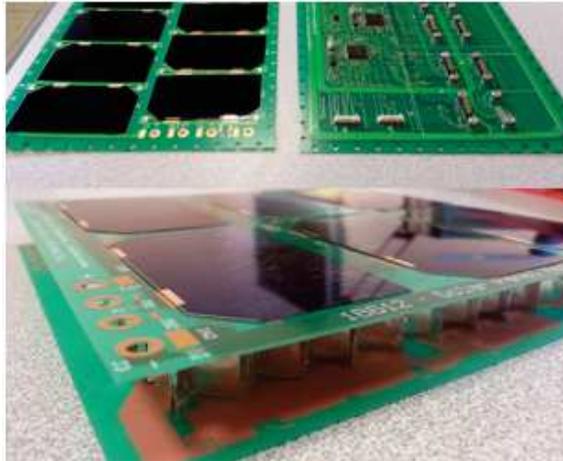


Figure 2.4. Photograph of Honeycomb Tile

solar cells and two lithium ion batteries. The tiles are fabricated autonomously and combined together using Al rods in any desired structure. These mechanical rods have screw holes at 11.75mm pitch, which ensures high mechanical strength together with an excellent electromagnetic sealing at 2.4 GHz. The structure is such as to guarantee power, data and RF harness. Fig.2.1 also highlights different rods to connect the AraMiS tiles in certain configurations. The simplest shape that can be built with the manufactured tiles is a cube having five power management tiles and one telecommunication tile held together with 12 rods.

### 2.2.2 Subsystem Modules

Every sub-system is housed in small daughter board modules and is supplied, by the main tile, with power and data distribution functions, power and data harness, mechanical support, and is attached and interconnected with space-grade spring-loaded connectors. There is the possibility to connect up to eight connectors on a single tile depending on the configuration. Connections are electrically and mechanically modular i.e. they can handle simple systems with a single analog channel up to larger systems with up to 14 analog channels, up to 64 digital Input/Output (I/O)s and central processing units with standard serial communication such as [UART](#), [IrDA](#), [SPI](#) and [I2C](#). Each module hosts a one wire memory for storing its own calibration and configuration data which is read automatically by the tile processor to provide

calibrated housekeeping. Each module holds an enable signal through which it can be activated or deactivated via software. The standardized connection points for the tiles have different dimensions depending upon the functions of every pluggable module. The designed modules contain electronic circuitry on top and spring loaded connectors on bottom side. Fig.2.5 shows the tile central processing unit interfaced with pluggable modules. Major subsystems have been designed on small modules

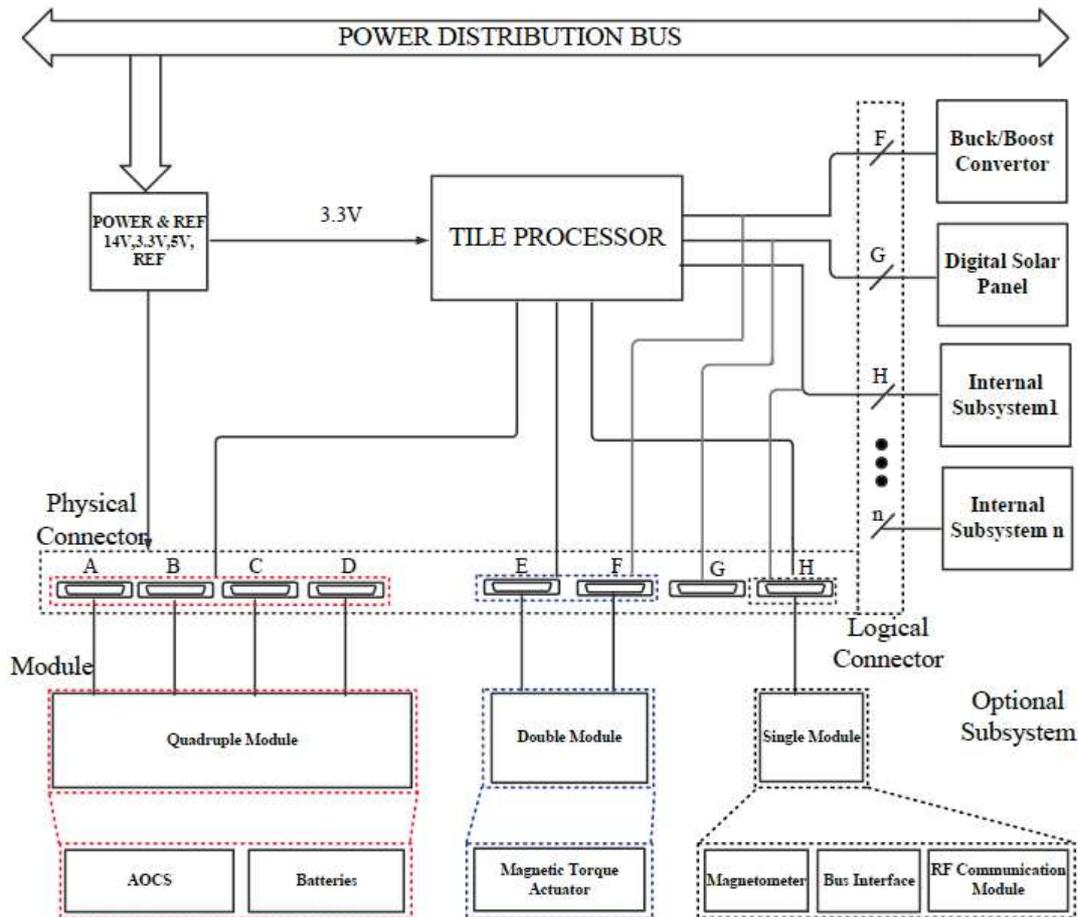


Figure 2.5. Generic tile architecture showing physical and logical connectors

and the size of each module depends on the complexity of each subsystem. Each module is optional and it can be connected on a certain tile only if the specific mission needs it. Logical connectors have been shown for the subsystems that are embedded on the tile PCB. Several spacecraft configurations are possible depending on the type of modules. Physical module based configuration comprises physical

daughter boards connected to the tile via pluggable connectors whereas logical module based configuration consists of modules that are connected by means of logical connectors. A hybrid combination of the modules is used for satellite on demand configuration. Three types of daughter board modules are currently being used for physical module based design i.e. single, double and quadruple. Receptacle and plug connectors are used on the tiles and modules respectively. The subsystem modules having header connectors can be connected onto the receptacle connectors of tile. Every single, double or quadruple module contains either one, two or four connectors respectively. The printed circuit board layout has been designed using commercial Mentor Graphics Expedition PCB. Commercially available connectors have been evaluated and used in each module. The modules exchange data and power in a distributed and self-configuring architecture which is flexible because standardized interfaces between the components are specified. These modules are currently being fabricated for all major subsystems of AraMiS. Fig. 2.6 shows dimensions of single, double and quadruple modules and placement of receptacle connector in each module as seen from component side. All the electronic components are placed on the

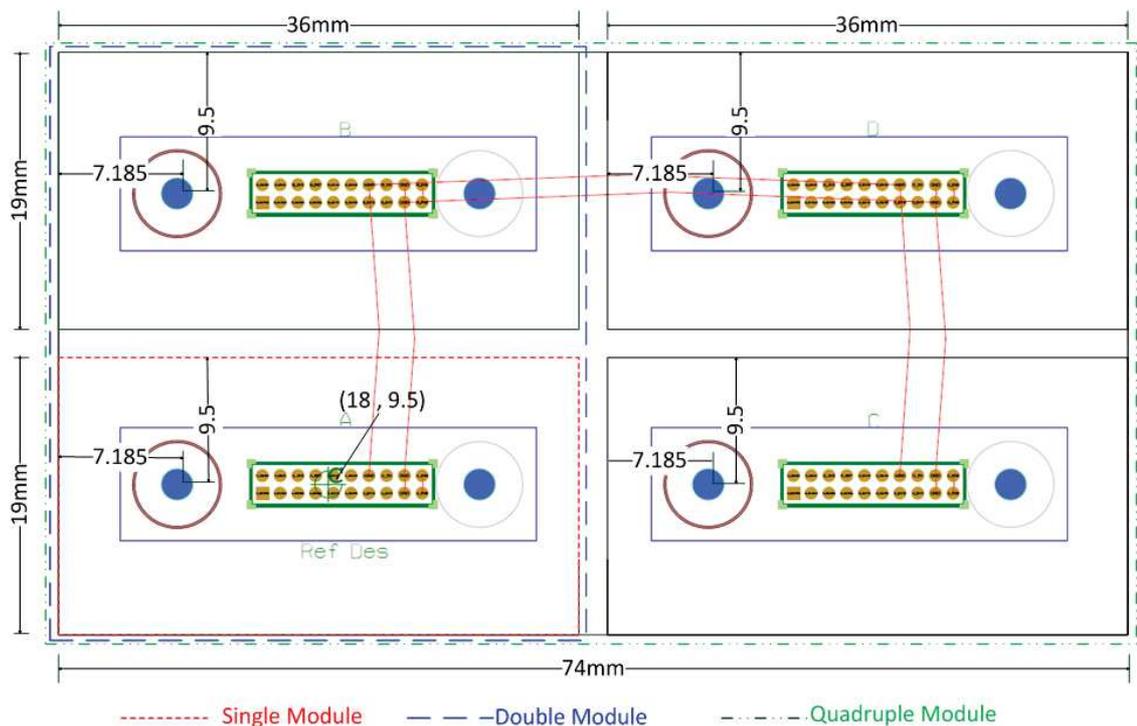


Figure 2.6. Dimensions and details of single, double and quadruple modules (seen from component side, towards satellite interior)

top side of the modules and spring loaded connector on the bottom side to be connected on the tile. Figure 2.7 shows some photographs of the fabricated modules. Each module placed on an AraMiS tile is compliant with the software section. Since

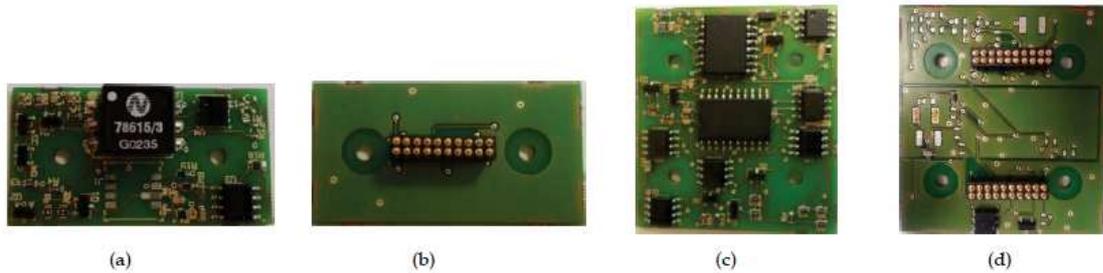


Figure 2.7. Fabricated modules (a) single module: component side view (b) single module: connector side view (c) double module: component side view (d) double module:connector side view

the number of communication channels of any processor is limited, a mapping mechanism has been implemented to use all the communication channels as effectively as possible. As an example, any two single modules that require RS232 communication cannot be placed on connectors A and B simultaneously because RS232 communication channel is not available in two consecutive connectors. Further details about the supported applications can be found in table 2.2. Every double module consists of two connectors that are placed on each AraMiS tile. Magnetic torque actuator has been designed to fit on a double module. Quadruple Module consists of four connectors (four single or two double). Complete attitude and orbit control system (AOCS) and battery subsystem are fabricated on quadruple modules. This modular design technique gives a lot of flexibility because the printed circuit board for any other mission can be generated very easily and with small effort.

### 2.2.3 Extended Modules for Larger Subsystems

There are a number of subsystems that require a larger area. Extended modules have been designed which allow more area for the components.

The Fig. 2.8 shows the dimensions and details of extended single, double and quadruple modules.

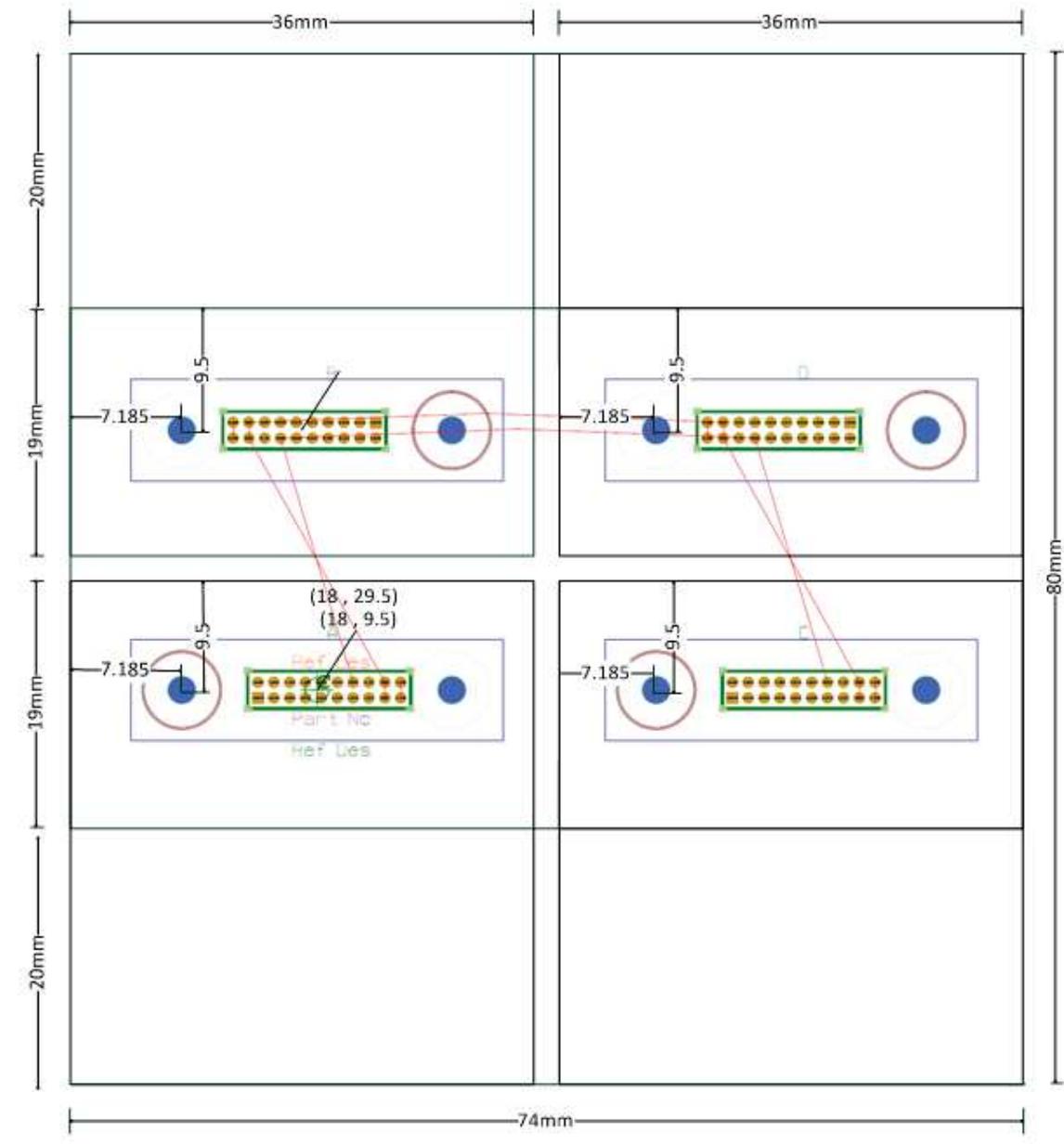


Figure 2.8. Layout of Extended modules

### 2.2.4 Logical Modules

All the subsystem module schematics and printed circuit board (PCB) layout have been designed in compliance with this modular approach. There is also the possibility to take the desired subsystem schematics and PCB layout and insert them

within the main tile PCB with direct connections using logical modules instead of using physical connector based modules. The processor signals have been mapped in a symmetrical way with the logical module signals. This kind of approach is used in logical based spacecraft configuration where the modules placed permanently on the tile. 1U, 2U, 3U...6U CubeSat standard tiles have been fabricated mainly using this scheme.

### 2.2.5 Modules Electrical Interface

Modularity at electrical level is achieved by using the desired standardized modules only plugged when required. The software is capable of detecting the desired module and performs the intended functionality. The physical module and module slot contains a number of digital, analog, power and communication channels as shown in Fig.2.9

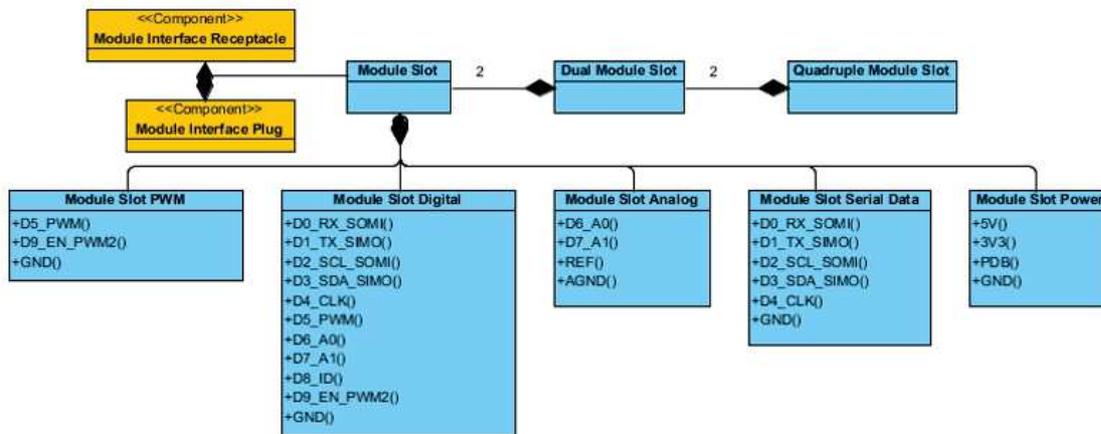


Figure 2.9. Signal scheme of module slots

The modules pluggable connectors have symmetrical signal interface with the tile processor which is depicted in table 2.1. The double and quadruple modules contain two and four connectors respectively and therefore, additional digital, analog and communication channels. The logical modules have also the same interface with the tile processor but via logical slots instead of physical connectors.

## 2.3 Modularity at Protocol Level

The tile software is also modular and allows a quick adaptation to specific subsystems. The basic software for the CPU is properly hardened to guarantee high level

Signal	Symbol	I/O	Description
Power Distribution Bus	PDB	O	Power Distribution bus to supply 12-18V unregulated power supply. This power bus is shared to all AraMiS avionics
+5V	5V	O	Positive 5V $\pm 5\%$ power supply
+3.3V	3V3	O	Positive 3.3V $\pm 5\%$ power supply
Reference Voltage	REF	O	3V $\pm 1\%$ reference voltage
Digital I/O	D0-D9	I/O	Digital Input/Output. Any of these pins can be used as general purpose digital I/O if not configured for other purposes.
Analog I	A0-A1	I	Analog Input
UART/IrDA Receive	RX	I	A standard UART serial line receive, used in RS232 mode and IrDA mode. This is connected to the RX pin of MSP430 UART.
UART/IrDA Transmit	TX	O	A standard UART serial line receive, used in RS232 and IrDA mode. This is connected to the RX pin of MSP430 UART.
SPI	SOMI	I	Slave Out Master In. Used in SPI
	SIMO	O	Slave In Master Out. Used in SPI
	CLK	O	SPI Clock
I <sup>2</sup> C	SCL	O	Serial Clock
	SDA	I/O	Serial Data
Identification	ID	I/O	To be connected with 1 wire memory for module identification.
External Signals	EXT1-EXT2	I/O	External general purpose connectors for routing purposes
Pulse Width Modulation	PWM,PWM2	O	Pulse Width Modulation
Ground	GND	-	Common Ground Pin
Analog Ground	AGND	-	Analog Ground
Enable	EN	-	Each connected module is enabled by activating this signal.

Table 2.1. Electrical Interface between AraMiS Modules and Tile Processor

of radiation tolerance at very low cost. The software has been completely defined, specified, designed and documented using UML tool [33–36]. This section defines the pin mapping of each pluggable module for several functions and an overview of supported communication protocols. Connections to the connectors are also electrically and mechanically modular. Table 2.2 shows the pin mapping of connectors (A-H) embedded on each tile for module placement. The modules support multiple functions and communication protocols when plugged on AraMiS tile. Every signal of table 2.2 can be used for digital I/O functions if it is not dedicated for special functions. Analog channels A0 and A1 read the inputs of voltage, current or temperature sensors to be sensed by the microcontroller. The CPU timers are capable of generating pulse width modulated signals (PWM, PWM2) essential for generation of duty cycle for switching regulators and set/reset pulses for magnetometer module

	A	B	C	D	E	F	G	H
<b>PDB</b>	2	2	2	2	2	2	2	2
<b>3V3</b>	1	1	1	1	1	1	1	1
<b>5V</b>	1	1	1	1	1	1	1	1
<b>RS232</b>	1	-	1	-	1	-	1	-
<b>IrDA</b>	1	-	1	-	1	-	1	-
<b>SPI</b>	*conditional	1	conditional	1	conditional	1	conditional	1
<b>I<sup>2</sup>C</b>	1	1	1	1	1	1	1	1
<b>Analog</b>	2	2	2	1	2	2	2	1
<b>Digital</b>	8	8	8	8	8	8	8	8
<b>PWM</b>	2	1	2	1	2	1	2	1
<b>Interrupt</b>	1	1	1	1	1	-	1	1
<b>Shared Digital</b>	2		2		2		2	

Table 2.2. Interface of AraMiS module signals. For each connector (A-H), corresponding number of channels (in digits) are shown

etc. Each tile connector has capability of multiple analog, digital and communication channels. The detailed pin mapping of module signals with the latest used tile processor, MSP430F5438A is shown in table 2.3 and table 2.4.

Conn	Pin	A	B	C	D
D0/RX/SOMI	11	P3.5/UC <del>A0</del> SOMI/UCA0RXD	P7.3/TA1.2	P5.7/UCA1SOMI/UCA1RXD	P8.1/TA0.1
D1/TX/SIMO	9	P3.4/UC <del>A0</del> SIMO/UCA0TXD	P8.0/TA0.0	P5.6/UCA1SIMO/UCA1TXD	P8.2/TA0.2
D2/SCL/SOMI	7	P3.2/UCB0SCL	P3.2/UCB0SCL	P5.4/UCB1SCL	P5.4/UCB1SCL
D3/SDA/SIMO	5	P3.1/UCB0SDA	P3.1/UCB0SDA	P3.7/UCB1SDA	P3.7/UCB1SDA
D4/CLK	3	P3.0/UC <del>A0</del> CLK	P3.3/UC <del>A0</del> STE	P3.6/UCA1CLK	P5.5/UCA1STE
D5/PWM	1	P4.0/TB0.0	P4.1/TB0.1	P4.2/TB0.2	P4.3/TB0.3
D6/A0	12	P6.0/A0	P6.2/A2	P6.4/A4	P6.6/A6
D7/A1	10	P6.1/A1	P6.3/A3	P6.5/A5	P4.7/TB0CLK/ SMCLK
D8>ID/INT	4	P1.0/TA0CLK/ ACLK	P1.2/TA0.1	P1.4/TA0.3	P1.6/SMCLK
D9/EN/PWM2/INT	2	P1.1/TA0.0	P2.5	P1.5/TA0.4	P1.7

Table 2.3. Interface of AraMiS module signals with the MSP430F5438A

	Pin	E	F	G	H
D0/RX/SOMI	11	P9.5/UCA2SOMI/UCA2RXD	P8.7	P10.5/UCA3SOMI/UCA3RXD	P8.5
D1/TX/SIMO	9	P9.4/UCA2SIMO/UCA2TXD	P8.4/TA0.4	P10.4/UCA3SIMO/UCA3TXD	P8.6/TA1.1
D2/SCL/SOMI	7	P9.2/UCB2SCL	P9.2/UCB2SCL	P10.2/UCB3SCL	P10.2/UCB3SCL
D3/SDA/SIMO	5	P9.1/UCB2SDA	P9.1/UCB2SDA	P10.1/UCB3SDA	P10.1/UCB3SDA
D4/CLK	3	P9.0/UCA2CLK	P9.3/UCA2STE	P10.0/UCA3CLK	P10.3/UCA3STE
D5/PWM	1	P4.4/TB0.4	P4.5/TB0.5	P4.6/TB0.6	P8.3/TA0.3
D6/A0	12	P6.7/A7	P5.1/A9	P7.5/A13	P7.7/A15
D7/A1	10	P5.0/A8	P7.4/A12	P7.6/A14	P9.7
D8/ID/INT	4	P2.0/TA1CLK/ MCLK	P2.2/TA1.1	P2.4/RTCCLK	P2.6/ACLK
D9/EN/PWM2/INT	2	P2.3/TA1.2	P2.1/TA1.0	P1.3/TA0.2	P2.7/ADC12CLK/DMAE0

Table 2.4. Interface of AraMiS module signals with the MSP430F5438A

### 2.3.1 Supported Communication Protocols

The AraMiS architecture employs a basic protocol which initiates communication between the tile processor and the pluggable modules. The basic protocol supports multiple communication protocols including RS232 UART, IrDA, SPI, and I2C communication protocols [10, 37] for modules that are connected using a wired approach.

#### RS232

A standard UART communication in RS232 asynchronous mode requires two external pins; transmit and receive, namely UCAXRXD and UCAXTXD for tile processor [38]. Connector pin RX is internally connected to the UCAXRXD pin of the microcontroller and TX is connected to the UCAXTXD of the microcontroller. At most four UARTs are available on the MSP430, so RS232 signals are only available on modules A, C, E, G. Table 2.2 shows the communication channels for RS232 mode.

#### IrDA

The IrDA encoder sends a pulse for every zero bit in the transmit bit stream coming from RS232 UART. It also uses UCAXRXD and UCAXTXD only to be configured

for IrDA in software. Therefore as in case of RS232, IrDA is only available on modules A, C, E, G.

## I2C

I2C is a two wire protocol that requires only two lines; serial data (SDA) and serial clock (SCL). In a single master scheme, any required number of slave devices can be connected to SDA and SCL lines. In order to use I2C, all SCL and SDA signals are shared in pairs i.e. connectors A, B share same I2C channel. Similarly pair C, D; pair E, F and pair G, H share the same I2C channels. SCL and SDA signals are connected to UCBxSCL and UCBxSDA signals respectively. The shared channels are indicated in table 2.2.

## SPI

SPI is either three or four wire protocol that requires slave out master in (SOMI), slave in master out (SIMO) and clock (CLK) signals [10]. Optional active low slave transmit enable (STE or EN) signal may also be needed in some cases. Modules B, D, F, H have capability of SPI protocol whereas modules A, C, E, G can be configured for SPI if not configured for RS232 mode.

## 2.4 Modularity at Hardware/Software Level

A trade-off was performed at the early stage of design between low and high level programming languages. Low level languages are machine dependent and are close to hardware. They are more complicated and cannot be optimized like those of their high level counterparts. They are faster but difficult to interpret by human programmer. The high level techniques are machine independent languages that are easy to write and modify. The programmer can write the programs without much knowledge of the processor. These programs are then translated into machine language by a compiler. Based on the above analysis, the hardware/software control of every pluggable module was implemented in high level language using Unified Modelling Language (UML). The design methodology employs UML class diagrams for every module. With the proposed approach, every subsystem is composed of a hardware (HW) part i.e. each of the small module boards described in section 2.2.2, a corresponding software support, properly hardened against radiation and an appropriate I/O interface. Fig. 2.10 shows a UML class diagram of a pluggable module interface showing the association of IOmodule\_MSP\_x class with the tile processor through t\_UART\_MODES, ADC\_CHANNEL, and PWM classes therefore; the modular software is able to perform any desired function when a specific module is accessed. The electromechanical parameters of the module are described by appropriate class

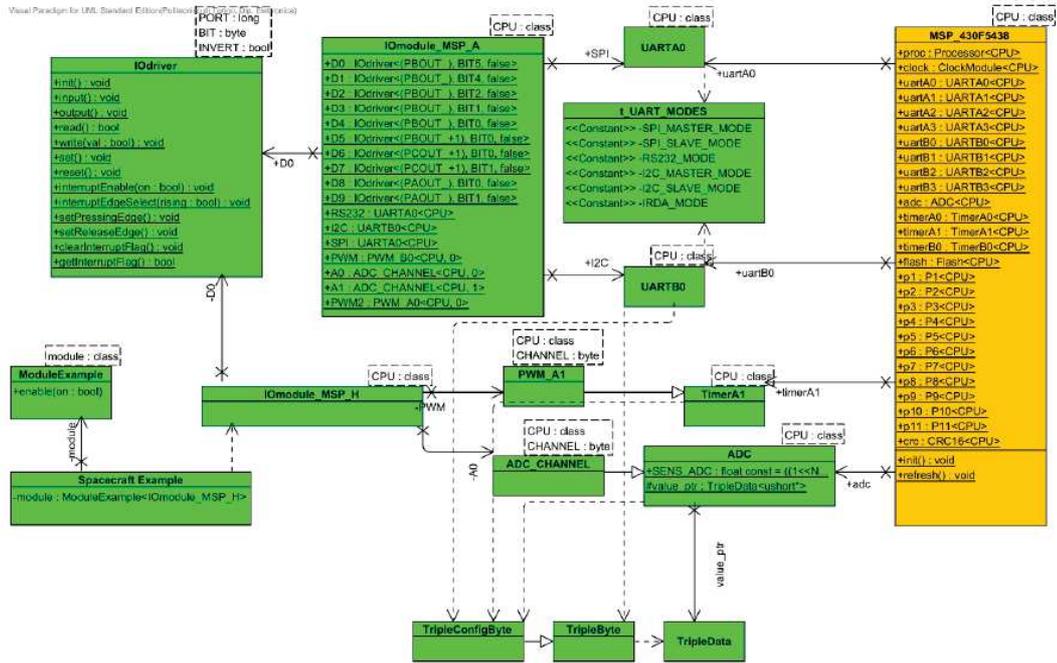


Figure 2.10. UML class diagram view of AraMiS modular drivers

attributes, properly documented. UML class for any module is brought inside the UML model of the spacecraft under construction at integration time. When the module is enabled, the correct software is automatically generated.

### 2.4.1 Modular Software Drivers

The class `IOModule_MSP_x` implements the software drivers for the signals of table1 indicated as class attributes. Digital I/O D0-D9 are associated with `IOdriver` class which is a generic input output software driver used with any processor for input/output functionalities. Each pin is instantiated by setting the corresponding template parameter for the desired function. The processor pins for either normal digital input/output or special purpose functionalities are configured by the class `IOdriver` according to following template parameters;

- **PORT**: address of register corresponding to the output port associated with the desired pin
- **BIT**: mask corresponding to the specific bit of output port associated with the desired pin

- INVERT: sets the logic level of the signal connected with input pin

Supported communication protocols are selected by the `t_UART_MODES` class. This class supports multiple serial communication modes. Each UART module is named with a different letter; UARTA is different from UARTB, etc. If more than one identical modules are implemented on one device, these modules are named with incrementing numbers. Any device having two UARTA modes is named UARTA0 and UARTA1. The tile processor can support up to four UARTA and UARTB modes. UAR $T_x$  modules support RS232 protocol, pulse shaping for IrDA communications and SPI protocol while UAR $TB_x$  modules support I2C and SPI protocols. Fig. 2.10 shows the association of IOmodule with different UART modes. RS232 is associated to UAR $T_x$  while I2C is associated to UAR $TB_x$  while SPI can use either of the two. The inputs of voltage, current or temperature sensors are controlled by the `ADC_CHANNEL` class which provides a conversion of analog input data to 12 bit digital signals. The ADC channel of the tile processor supports fast 12-bit analog-to-digital conversions. Timer and PWM classes implement the hardware and driver routines for timers. The software driver can initiate the timer operation by selecting either auxiliary clock (ACLK), main clock (MCLK) or external clock sources [19]. The software control can also reset, enable, disable or refresh the timers. This is a form of UML-based object oriented design approach which allows building either a spacecraft or, in this case, one of its subsystems according to specific mission requirements with very limited risks, effort and design time. A simple example class has been created to demonstrate the concept of using the modular software drivers. Spacecraft Example is a test spacecraft which contains a specific module ModuleExample. This module is a software class and it can be activated when a physical or logical module is plugged. In this example, it is a magnetic coil actuator and it has a function to enable the coil. The spacecraft uses `IOmodule_MSP_H` class to implement the function `enable()` of ModuleExample. The syntax of the generated code is as follows

```
#include "ModuleExample.h"

template <class module> ModuleExample<module>::enable(bool on) {
    module::D9.write(on);
}
```

This will send a logic 1 on the enable signal (D9) of `IOmodule_MSP_H`, actuating the coil associated to magnetometer. Therefore, every pluggable module is activated, deactivated and programmed using the proposed design approach. All the classes have been designed using radiation hardening techniques. Fig. 2.10 shows the association of certain classes with the radiation hardening classes. Although the presented approach is particular for one family of microprocessor, it can easily be

extended to any other family or devices. All the class diagrams of ADC, timers, UART, IOModule and radiation hardening are generic and support different types of processors including TI, Chipcon family. The microprocessor family can easily be changed by changing an implementation flag during compilation of the class diagram of processor and corresponding software drivers in UML. Currently, work is underway to extend support for other commercial processors including PIC and Arduino. Next subsection describes in detail the software radiation hardening technique based on triple modular redundancy.

## 2.4.2 Radiation Hardening

The software programs running on commercial off the shelf processor are more prone to radiation. They may suffer from sensitivity to ionizing radiations, therefore to radiation induced single event effects (SEE). Single event upsets (SEUs) are transient faults caused by the ionization of a single charged particle and typically cause bit-flips which is an undesired change of state in the content of a storage element. Most software-based approaches are aimed at detecting faults, some of them apply redundancy to high-level source code by means of automatic transformation rules, whereas some others use instruction redundancy at low level (assembly code) in order to reduce the code overhead and performance degradation, and improve the detection rates [39–44]. The AraMiS software was developed using a high level software radiation-hardening technique completely developed in house. The radiation hardening technique aim at protecting the on-board computer and on-board data handling functions from SEEs (mostly SEUs), in particular:

- Bit-flips and data corruption in data storage.
- Corruption of time and spacecraft status and configuration registers (e.g. subsystem enable/disable, configuration word, calibration data, etc.), which might potentially be harmful to the whole system.
- Corruption of peripheral configuration registers in the micro-controller (e.g. interrupt enables and flags, configuration words of Analog to Digital Converter (ADC)s, UARTs and Timers, etc.).

The radiation-hardening technique is based on the use of appropriate C++ classes from the hardened data (Hdata) package developed in house, which can be used in a standard C++ program instead of standard data type. For instance, a short can be substituted by the so-called TripleShort, which automatically and transparently stores three copies of the same value and votes or recovers data whenever required. A normal C++ program can still be compiled by modifying only the data type definitions. This allows reusing software algorithms and procedures which have already

been validated and tested without any specific effort apart from redefining data types. For instance, the piece of code on the left column of table 2.5 can be changed to the code in the right column by changing only the first line. In the table 2.5,

Normal program	Hardened program
<pre>short a=3, b=5; short c; c = a+b;</pre>	<pre>TripleShort a=3, b=5; short c; c = a+b;</pre>

Table 2.5. Hardening of data using redundancy

instruction `a=3, b=5;` automatically stores the values "3" and "5" into three replicas "a" and "b", respectively. The operation `a+b` automatically sums up each replica of "a" with the corresponding replica of "b". The assignment `c =` to a variable which has been chosen to be non-replicated (a standard "short") automatically votes and stores the result into "c". This simple approach to TMR is made possible by using a set of software classes developed internally, one of which is partially visible in the figure below (TripleData). A similar approach can also be applied to replicate and protect microcontroller and peripheral configuration register, by means of the specialized class TripleConfigByte shown in Fig. 2.11, which periodically and autonomously refreshes configuration registers starting from three replicas of register content which are hidden into the class, yet preserving the volatility of certain configuration bits. All the software of AraMiS, including all the drivers of the subsystems, has been developed using the Hdata package. The radiation hardening technique was tested by injecting random background faults (i.e. emulating single event upsets) in configuration registers and data memory cells. The results are quite promising and yet to be published. The presented technique is intrinsically more redundant because most of the functions are intrinsically triplicated.

## 2.5 Conclusion

In this chapter we have shown a completely modular and flexible approach in the design of small satellites using the proposed smart harness technique. Since most of the tiles are identical in mechanical design and dimensions, it gives more flexibility in designing the satellite in any desired shape like cubic, hexagonal etc. Multiple choices of onboard data communication solutions reduce the chance of erratic transmission. UML based design for hardware/software control of each tile's telemetry and housekeeping is also quite flexible and modular. Finally, the pluggable single,

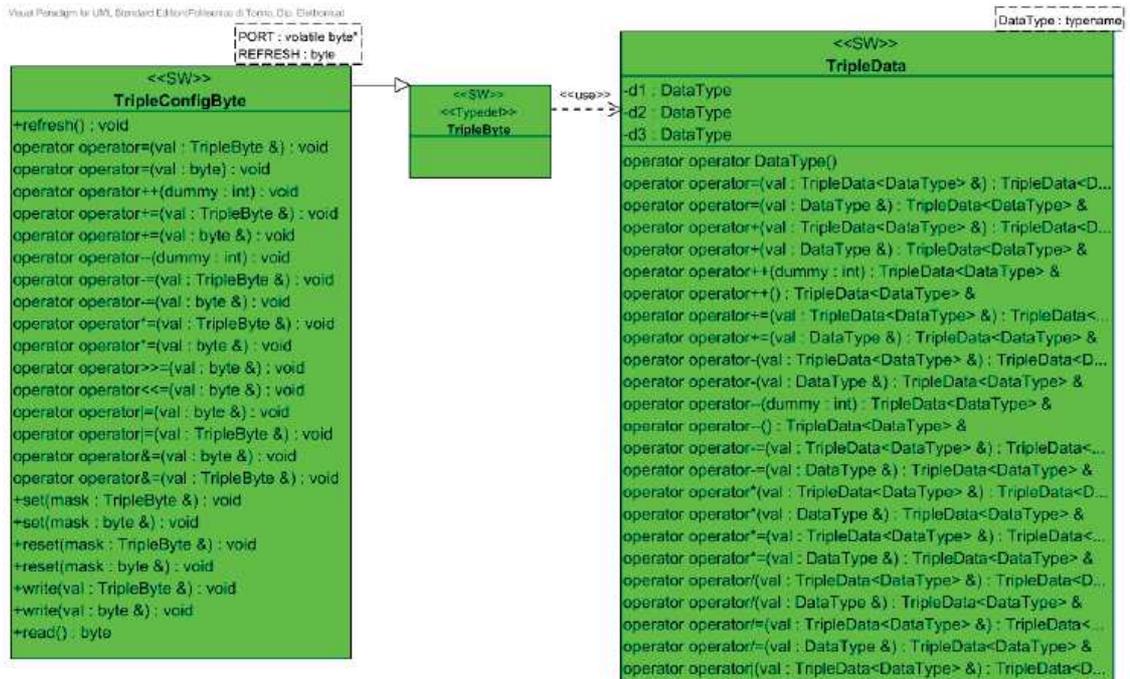


Figure 2.11. Three UML classes which are used to implement TMR into SW programs in a transparent way. The classes offer more operations than shown here, where they have been truncated to increase visibility

double and quadruple modules are capable of handling all the core functions of the satellites and can be designed according to specific payload for every mission.



# Chapter 3

## Module Life Cycle

### 3.1 Introduction

Designing a space system is a very complex task. It must be split into simpler functions and subsystems. Each function and subsystem has to be specified, designed, reviewed, validated, implemented, tested, deployed, and most importantly documented. Development requires a wide number of tools, techniques and approaches. Designing a low cost or university satellite has to cope with a number of additional tough constraints: Firstly, the individual designers and teams mostly consist of students who are available for a short period of time. They often appear and disappear quickly. Secondly, the designers and teams are often inexperienced and do not have much exposure to the vast variety of practical tools. Therefore, someone has to lay down specifications for heterogeneous teams. Innovative methodologies are necessary that allow the integration of more and more complex systems with innovative design, testing and documentation. This chapter proposes the design technique of initialization, requirements, documentation, and modeling of subsystems of small satellites as a test case, all using [UML](#) [33–36]. In particular, [AraMiS](#) is an innovative modular architecture alternative to CubeSats, for larger and more demanding applications, with a modularity at mechanical, electronic and testing levels, to be used mostly in [LEO](#) Satellites (600 km to 800 km). Modularity helps to share costs among multiple missions, to reduce manufacturing and testing costs and to easily increase redundancy. Actual satellite technology leads to high costs for space missions. The use of [COTS](#) reduces development time, cost and size. Reduced reliability of [COTS](#) can be compensated by proper design and redundancy. A key element of the [AraMiS](#) architecture is the design and documentation methodology used, which is based on an extension of [UML](#), as described here. The aim of this chapter is to introduce an approach to this problem using [UML](#) as a high level specification, description and documentation language. The purpose of this approach is

to obtain a complete development flow for mixed-systems able to produce, on one side, documentation always close to real project implementation and, on the other, a fast and reliable method for reducing time- to-market in developing these objects. The design of all the major subsystems of [AraMiS](#) has been carried out by using [UML](#). Initially developed in 1995 for designing software, the [UML](#) was optimally adapted to the description of systems made of both hardware and software. It is based on the representation of entities involved in the system functioning and all interactions among them. There are many advantages of using this language with most important being

- Make easier the project understanding, even by people external to the project, thanks to a graphical/conceptual representation of the elements that make the system (components, subsystems, signals, functions...) starting from a high level description to a specific one.
- Simplify and improve the description of system functionalities and the specification definition providing a common basis in the approach of designing the units forming the whole system.
- Make exportable the system building blocks (which are independent form each other) such that they can be re-used in other projects so implementing the modularity concept.

The [UML](#) design approach for [AraMiS](#) comes from the evolution of the University Satellites at Politecnico di Torino:

- [PICPOT](#) [45], launched on July 2006, but launcher blew up during flight
- [ARAMIS](#), heavily modular architecture for more demanding applications.

An innovative approach to teamwork design of complex system was needed:

- From mission- to circuit- and component-level
- Compatible with very heterogeneous and inexperienced teams and with a variety of CAD tools (mechanical, electrical, software, etc.)

Among all possible utilities that [UML](#) offers, there are certain types of diagrams used in UML including, but not limited to use case diagram, class diagram and sequence diagram.

## 3.2 UML Diagrams

### 3.2.1 Specifications: Use Case Diagrams

Use case diagrams show main function of the system (use cases) and the entities that are outside the system (actors). Use case diagrams show how the class and objects of the class relate and hierarchical associations and object interaction between classes and objects. These diagrams allow us to specify the requirements of the system and show interaction between system and external actors. These diagrams are the starting point in the system modelling and consist of actors and use cases.

#### Actors

Actors are generic entities, human users, other systems or the external environment, which interact with the system under design and implements one or more use cases. They are usually shown as sketched people with a short name which identifies the role in the system. They are associated with a detailed documentation. The list of actors is fundamental to understand all entities which might interact with the system. The actors are very fundamental entities and missing an actor will miss all the interfaces and functions associated with it. Therefore they are critical in the design and the designers need to put more time in identifying the possible actors during early stages of design. Let us imagine a situation where the designer forgot to add an actor "tester" in the early stage of the design. The possible consequences may be

- There might not be possibility of test connector to test the satellite
- There will be no internal access node for debugging
- No software will be added for detailed testing
- There might not be special satellite mode to allow testing before launch, etc.

#### Use Cases

The major work of the actors is to interact with different subsystems of the satellite. The main concerning points are

- What does and actor expect from a system?
- What does a system expect from an actor?

All this is detailed by Use Cases, which are usually described by an oval with a name which shortly describes it. Building up the list of use cases means starting

to specify the functions of the satellite or its subsystem and therefore thinking to the mission. There exist several kinds of relations between use cases and actors including generalization, inclusion, extensions, associations etc. This section details the use cases of the magnetic attitude subsystem of the AraMiS architecture. The use case diagram defines the functional specifications of the project and is shown in Fig.3.1 for magnetic torque actuator subsystem.

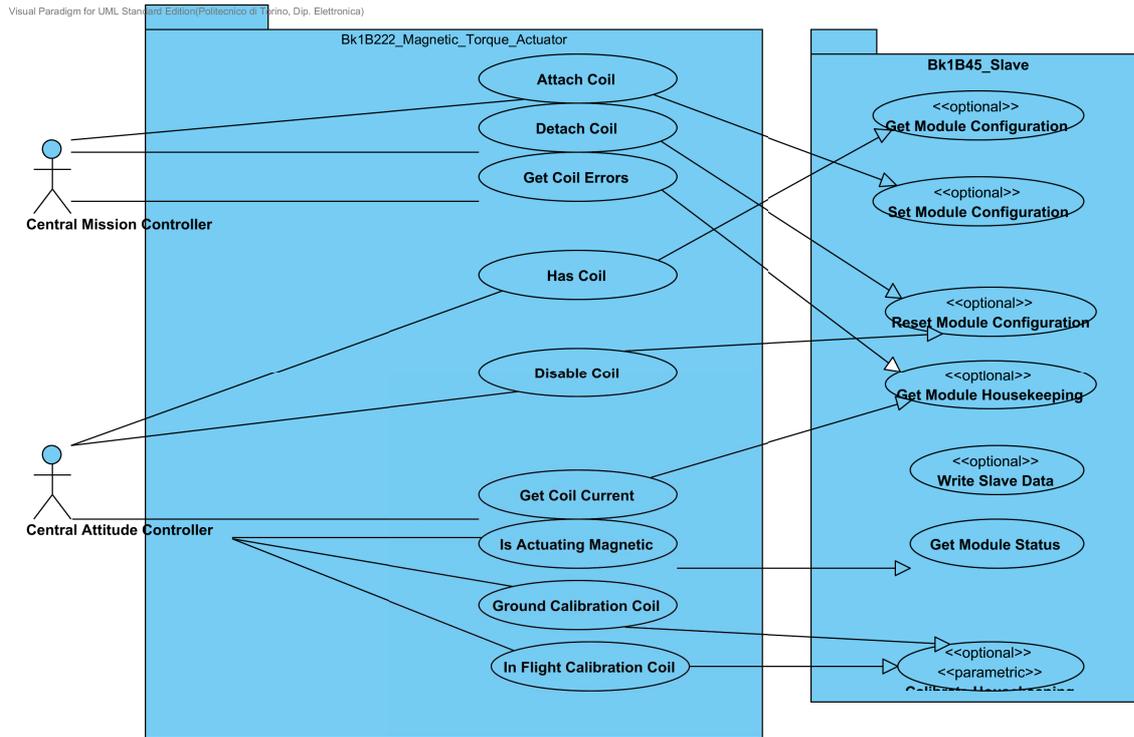


Figure 3.1. Use case diagram of magnetic attitude subsystem module

The central mission controller and central attitude controller actors interact with the system and perform many tasks. The central mission controller is an entity (likely a software routine running on the onboard computer) in charge of satellite supervision, fault detection and management and emergency management whereas the central attitude controller is an entity (likely a software routine running on the On-board Computer (OBC) in charge of managing the attitude control subsystem in nominal operation. Fault and emergency handling are left to the Central Mission Controller. All other use cases implement most of the magnetic actuation and control functions such as attach/ detach coil, get voltage and current levels of coil and housekeeping sensors etc. All the housekeeping functions are performed by using data handling technique described in [1].

### 3.2.2 Architecture and Design: Class Diagrams

The objects have tendency to know things i.e. they have attributes and they do things i.e. they have methods. All objects of the same type are represented by a class. In UML notion, classes are depicted as boxes with three sections, the top one indicates the name and stereotype of the class, the middle one lists the attributes of the class, and the bottom one lists the methods. Each object in UML classes can either be associated with hardware (HW), software (SW), an analog (ANA) implementation etc. depending on the stereotype of each class. Each stereotype is labelled with a specific colour. Each subclass has objects which contain different attributes and operations. UML classes and objects are used to specify any electronic, mechanical, software element of a system. The attributes of the class store data for the class. The attributes can either be constant, therefore representing characteristics common to all objects of that type variable, therefore storing time-variable data which are part of the class. Classes in turn can be made of one or more other classes (hierarchical structure). The generic UML class for AraMiS is shown in Fig.2 showing unique class for every major subsystem. The class diagram of attitude and orbit control subsystem has been elaborated as a test case. The magnetic attitude control system together with the inertial attitude control system is used to accomplish the desired rotation to the satellite, by sending commands directly from the ground. The system consists of certain sensing and control classes as shown in the Fig.3. The class diagram of magnetic attitude subsystem consist of

- Bk1B222\_Magnetic\_Torque\_Actuator block consisting of solenoid coil.
- Bk1B221\_Magnetometer\_Sensor block consisting of a magnetic field sensor and conditioning electronics.

The current flowing through the solenoid provides the tile angular momentum while the magnetometer measures the values of earth magnetic field in orbit. The block 1B22 uses a microprocessor located in the Tile Power Management to handle the different sub-commands to perform housekeeping calculations and calibrate the telemetry data.

### 3.2.3 Verification:Sequence Diagrams

Sequence diagrams describe how structural elements communicate with one another and time sequence of events. Time increases vertically from top to bottom. Fig.4 shows the sequence diagram of simple data handling example. Fig. shows sequence diagram to give the set/ reset pulses to our magnetometer. The controller sends either set or reset pulses to the magnetometer bock which ultimately passes to the magnetometer sensor.

The detailed analysis of using all the UML diagrams in the design, implementation and testing of any electronic system is described in the next section.

### 3.3 Design Flow

The UML diagrams are very helpful in building subsystems from conception to the final design and testing. Fig.3.2 shows a design sequence for magnetometer subsystem of AraMiS architecture. The use case diagrams indicate all the functions of the magnetometer managed by central mission controller and central attitude controller actors. These use cases are then realized by a set of UML classes in class diagram. The class diagram shows the subclasses for magnetometer and either hardware, software and analog components needed in order to complete the design. Each class is represented by a specific color based on its type. The classes have been divided into different types based on the attributes. The software classes contain the software codes in order to implement the specific use cases. This makes the design quite easier by placing pieces of codes in the corresponding classes and then generating the entire code at any time in the design. The detailed documentation of every use case, class and sequence diagram is written in the respective location. The built-in project documentation design offers a high level of flexibility, user control and customization. After selecting output format using the desired template, the project documentation can easily be customized according to the user needs. There is also possibility of selecting the level of detail for each element, such as including hierarchy diagrams to assist the communication of class relationships. There is also possibility to insert hyperlinks to aid the navigation in the project documentation. After the use case and class diagrams, the final design is realized as shown in Fig.3.2. Every hardware and component stereotyped class corresponds to physical component. The physical schematics can then be realized by use of any CAD tool. The final design is ready to be implemented at this stage. After the design is physically available, the next step is to test it. Testing sequence is realized by sequence diagrams. The test is performed according to the sequence diagram and verified with the requirement diagrams. If the test sequence is in accordance with the requirements, the final module is ready to be integrated onto the tiles. At the end, a complete documentation report of all the steps of the design can be easily generated. The generated code for testing and realization is quite clear, readable and well documented and properly commented. Moreover, the generated code clearly depicts the system design in terms of relationship between classes, their associations, etc.

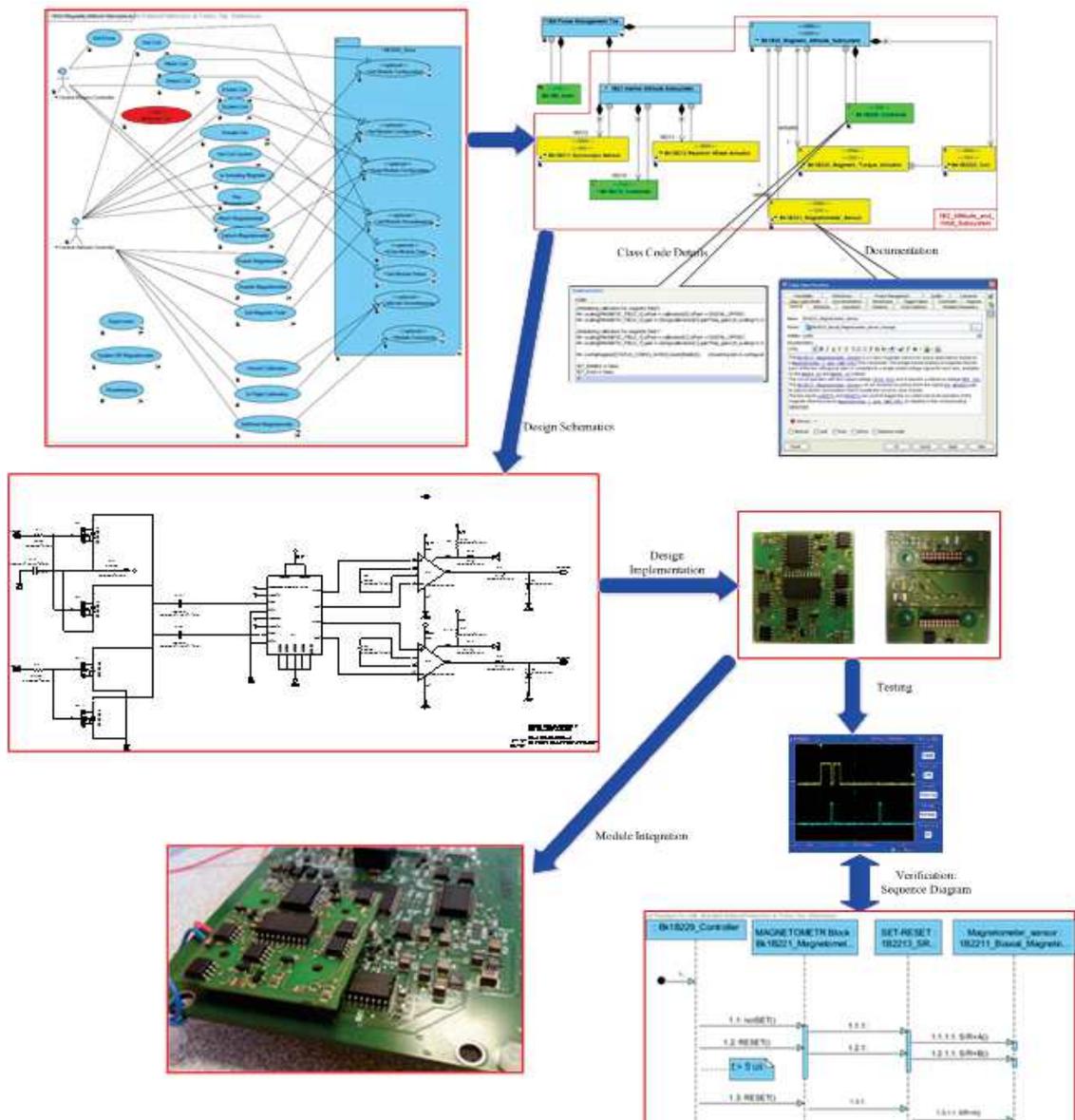


Figure 3.2. Design flow of magnetic attitude subsystem module

### 3.4 Conclusion

This chapter has introduced the complete life cycle of a module subsystem by use of powerful features of UML. There are many advantages of UML based design over typical design approach. The UML design approach is helpful to better understand the functionality of the system and overcome design limitations on the early stages of

the system. Design complexity and development time is reduced by using proposed approach. Extensive documentation of each diagram helps in understanding the system much better, reuse of UML singleton classes helps minimize probability of error. In addition, interaction between different blocks is much easier. UML tool can generate code for several programming languages. The generated code is quite clear, readable and well documented and properly commented. Moreover, the generated code clearly depicts the system design in terms of relationship between classes, their associations, etc. The whole life cycle is applied to all the subsystems from idea until the development and testing.

# Chapter 4

## Satellite Design

### 4.1 Introduction

### 4.2 Design Flow

The design technique of the AraMiS tiles and modules is quite flexible and a number of spacecraft configurations are possible including physical, logical and satellite on demand. The key design flow sequence is described as follows.

- Design the new subsystems either on single, double or quadruple module configuration.
- Test the subsystems on ground using development board. The tiles having processors and pluggable connectors can be transformed very easily into the development boards for the functional testing of modules.
- Integrate each module on the tile connectors in a physical module based satellite configuration. The specific module can be used only if the desired mission needs it.
- Embed the logical modules in the main tile for a logical module based satellite configuration. This configuration is permanent and cannot be altered after the design phase.
- Integrate physical and logical modules on a single tile for a custom satellite on demand configuration. This approach takes advantages of both physical and logical configurations

### 4.2.1 Physical Module Based Configuration

This configuration uses standard tiles hosting multiple connectors and the individual subsystems are developed on physical daughter boards, connected to the tile via pluggable connectors. The subsystem modules are used only if the specific mission needs it. This configuration achieves high level of design flexibility, testability and upgradability. The testing of modules, tiles and the whole satellite is needed in this type of configuration. This configuration is mostly employed for teaching/research purposes because the integration of the physical modules causes highly integration complexities.

### 4.2.2 Satellite on Demand Configuration

This configuration embeds the already developed and tested modules inside the PCB of the tile. Therefore in this configuration, the design of physical modules in a permanent configuration. Testing at tile and mission level is required and not at the module level. The CubeSat standard tile is built using this approach

### 4.2.3 Reusable Design Configuration

The reusable design configuration is an optimised spacecraft configuration based on the customer requirements. This configuration reuses the satellite on demand configuration with minor addition or removal of specific subsystems on customer demands. This configuration actually follows the Cheaper-Faster-Better philosophy. The testing is performed only at mission level as the modules and tiles of the previously tested configurations are used in this spacecraft.

A design trade-off was performed for using different types of modules in the above mentioned satellite configurations. Physical module based configuration has high level of design flexibility, testability and can easily be upgraded but the shortcomings are increased mass/weight because of using separate modules, less space for payload and difficult integration of physical modules on the tile connectors. Logical module based configuration cannot be upgraded once designed. Moreover it has lower testability than the other schemes. The detailed comparison of the above configurations is depicted in Fig.4.1. In [46], we have presented design and test of bus interface circuit on single module emphasizing the concept of plug and play approach.

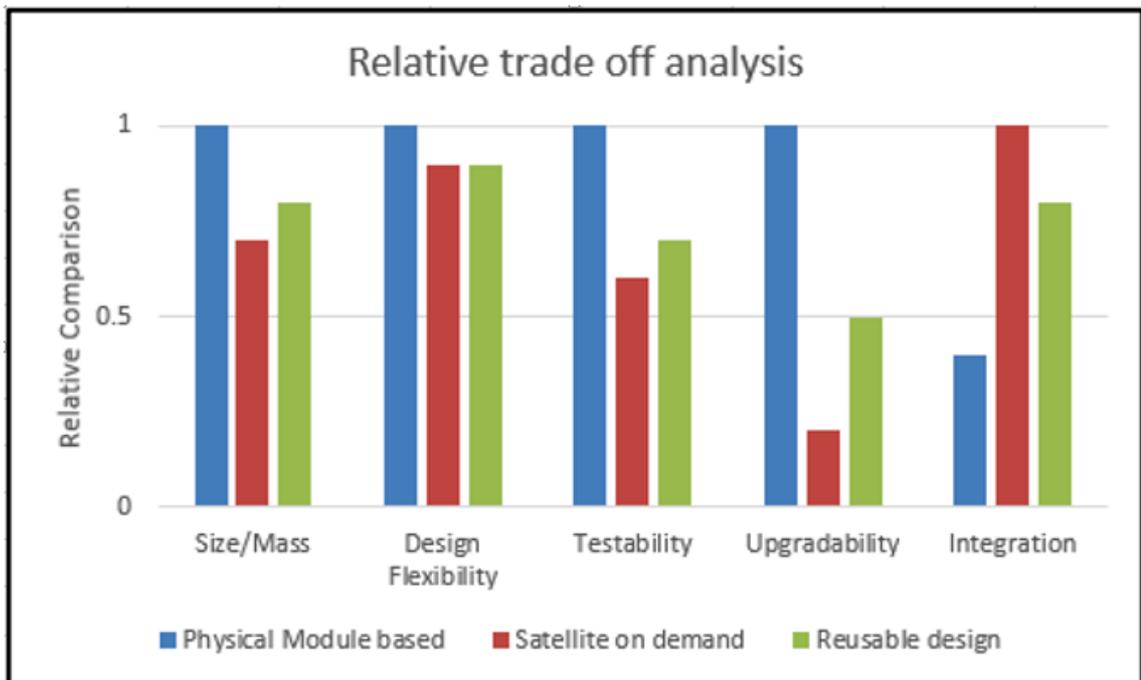


Figure 4.1. Trade off analysis of using modules in different spacecraft configurations



# Chapter 5

## Satellite Design: Satellite on Demand

### 5.1 Introduction

This chapter focuses on the satellite on demand configuration. The main idea of satellite on demand configuration has already been presented in Chapter 4. All the necessary subsystem modules have been integrated in a single PCB making the design, integration and testing quite simple. This configuration is permanent and the next variant of this configuration is *reusable design configuration*. The next sections describe in detail the satellite structure built using this configuration.

### 5.2 AraMiS-C1

AraMiS - C1 is a 1U CubeSat standard nanosatellite built as a demonstrator of the AraMiS modular architecture completely using the satellite on demand technique. Four sides of the satellite are equipped with identical tiles that mount solar panels on the exterior and a combined power management, attitude control and computing subsystem on the interior. The other two sides are devoted to the telecommunication links and carry a commercial deployable UHF antenna (one side) and a patch type SHF antenna (the other side) on the exterior. Inside the satellite there is room for batteries and payload boards. Once deployed, the ISIS - Deployable Antenna System for CubeSats will release four antenna baffles. No drag augmenting device and no other deployable device (except antennas) is installed. The AraMiS - C1 is designed to be functional over a period of two to three years on an orbit in the 500 km range, but even lower orbits with higher atmospheric drag that will guarantee a few months in space are acceptable for our purposes. Obviously longer orbital life (at least one year) will be more appropriate for the scientific objectives of the

mission. AraMiS-C1 have the following subsystems.

- ISIS - 1-Unit CubeSat Structure
- 1B8-CubePMT modules
- 1B9-CubeTCT modules
- Payload
- Battery pack
- UHF antenna
- Harness
- On-board data handling
- Tile Processor
- On-Board Computer (OBC)

Electric power supply and attitude determination & control subsystems are the most essential elements of any aerospace mission. Efficient EPS and precise ADCS are the core of any spacecraft mission. So keeping in mind their importance, they have been integrated and developed on a single tile called 1B8-CubePMT module.

### 5.2.1 ISIS-1U CubeSat Structure

The ISIS 1-Unit CubeSat structure is developed as a generic, modular satellite structure based upon the CubeSat standard. The design created by ISIS allows for multiple mounting configurations, giving CubeSat developers maximum flexibility in their design process. The stack of PCBs and other flight modules can be build up first in the secondary structure and integrated with the load carrying frames at the end of the process, ensuring accessibility of the flight avionics. In addition, the use of a load carrying frame and detachable shear panels allows for access to all parts of the spacecraft avionics, even after final integration by removing one or more of the shear panels. The modular structure allows up to two 1-Unit stacks of PCBs, or other modules, to be mounted inside the structure. ISIS-1-Unit CubeSat structure is shown in Fig.5.1

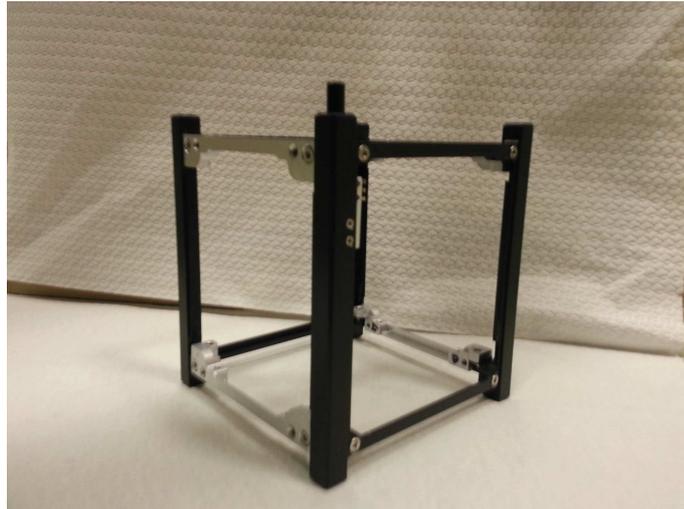


Figure 5.1. ISIS Frame for AraMiS-C1

### 5.2.2 1B8 CubePMT

CubePMT is built on an 82.5 x 96 x 1.6 mm<sup>3</sup> FR4 PCB which also acts as mechanical structure. It has solar panels and sun sensor on the external side while internal side contains components of power management and attitude subsystems. Modular power management tiles (PMTs) are already available in the market but they are less efficient, heavier in weight, consume more power and contain less number of subsystems. Commercial of the shelf (COTS) components have been used for CubePMT implementation which are low cost and easily available from the market. CubePMT is developed on the design approach of AraMiS architecture.

CubePMT subsystems include:

- Solar panel
- Boost converter
- Switching and linear regulators
- Load Switches
- Magnetometer
- Gyroscope
- Sun sensor
- Housekeeping Sensors (Current, voltage and temperature sensors)

- Magnetorquer coil
- Magnetic torque actuator
- Tile processor (MSP430)

All the subsystems have been discussed in detail in [47]. The next subsection describes the telecommunication tile of AraMiS-C1 cube.

### 5.2.3 1B9 CubeTCT

CubeTCT is a four layered PCB with dimensions 98.0mm x 98.0mm x 1.55mm. The components are mounted on the bottom layer (layer 4) while the top layer (layer 1) contains a feed point for mounting detachable S-band patch antenna (2.4 GHz). The routing for RF, Power and other analog traces are performed on the



Figure 5.2. Cube Telecommunication Tile

bottom surface. All the digital traces are routed on the third layer (3). The second and fourth layers are ground planes which help in shielding RF, Power and analog traces from digital signals. Therefore, ensures a considerably better signal integrity. The CubeTCT communicates with onboard computer (OBC) through a 6 pin SPI connector. Power Distribution Bus (PDB) is made available using a separate 4 pin connector. Two 8 pin connectors are available for separate programming and debugging of the Texas Instruments CC2510 Transmitter TX and Receiver RX. The data processing, monitoring and various control operation for CubeTCT is also perform by these commercially available transceivers which has an internal 8051 core microcontroller. A photograph of CubeTCT module is shown in figure 5.2 and subsequent design implementations are presented in [48]. The system level

integration of AraMiS-C1 uses both power management and telecommunication tiles and are discussed in next subsection.

### **1B31A Tile Radio Frequency Module 437MHz**

The UHF Module is used both as a transmitter and receiver, in half duplex configuration mode. It consists of MSP430 microcontroller performing protocol functions and two CC1020 transceivers that are configured separately as a transmitter and receiver respectively. As shown in the figure 2.10, the RF front end consists of Power Amplifier, LNA that are connected to transmitting cc1020 and receiving cc1020 which is then fed to a four element monopole antenna via Solid State Switch.

### **ISIS - Deployable Antenna System for CubeSats**

The ISIS deployable antenna system contains up to four tape spring antennas of up to 55 cm length. The system can accommodate up to four monopole antennas, which deploy from the system after orbit insertion. The wires are melted using two redundant heating elements per wire. RF phasing / BalUn circuitry ties the antennas together in for instance a turnstile configuration, two dipoles, or just one dipole. ISIS can configure the antenna system to be compatible with all UHF and VHF radios which are typically used for CubeSats. Depending on the configuration, one or two radios in the CubeSat can connect to the antenna system by means of miniature RF connectors. Furthermore, the top face of the antenna system can accommodate a two solar cells solar panel and provisions have been made such that it can be customized for customers who require sensors or other systems to protrude to the exterior, e.g. camera apertures. The antenna system has been designed for maximum compatibility with existing COTS CubeSat components. It is compatible with any UHF and/or VHF radio system. It can be mounted on top and bottom faces of all ISIS CubeSat structures and Pumpkin rev C and rev D CubeSat structures. For custom made structures, which adhere to the CubeSat standard mechanical envelope, mounting should also be possible.

### **1B31B Tile Radio Frequency Module 2.4GHz**

CC2510 transceivers that are configured separately as a transmitter and receiver, respectively. The Protocol functions are performed by the Embedded 8051 core tile each cc2510 transceivers. The RF front end consists of Power Amplifier, LNA that are connected to transmitting CC2510 and receiving CC2510 which is then fed to a four element monopole antenna via Solid State Switch.

### 5.2.4 Payload

The payload is a scientific/technological payload composed of three independent board, developed by different groups. All of them have the common aim of comparing the sensitivity to radiations of commercial microcontrollers in order to make results available to the scientific community. The same (or a similar) program will run on all the boards, aimed at counting the SEUs which accumulate during the lifetime of a microcontroller in LEO orbit, under similar conditions. The three boards will host

- Payload - Board 1: a Texas Instrument’s MSP430F5438 microcontroller
- Payload - Board 2: a Microchip’s PIC microcontroller
- Payload - Board 3: a few flavors of Xilinx’s PicoBlaze soft-core on an Actel ProAsic FPGA, with different levels of HW radiation hardening. This payload has been developed at University of Alicante

### 5.2.5 System Level Integration

System level configuration of the cube architecture is shown in Fig.5.3. It shows the inter tile communication and kill switch interface using I2C communication. Telecommunication tile, additionally, uses SPI for communication of S band CC2510 transceiver [49] with the on board computer. The central power distribution bus distributes the power and reference voltage signals to each tile and ultimately corresponding modules using a centralized scheme [50]. The photograph of system level integrated AraMiS satellite on demand is shown in Fig.5.4

The on-board computer is put in deep low power mode during integration and launch. The CPU and all clocks are disabled in this mode and power consumption of tile processor is ultra-low i.e. 3.6uW (LPM4 of [38]). The on-board batteries provide 15Wh storage capacity which ensures several tens of years of standby time. Battery self-discharge would be more significant than power consumption. There is no problem of electromagnetic interference since the clocks are disabled during the launch. As soon as the kill switch is released, it sends the OBC a wake up interrupt which ultimately turns PDB on and additionally sends the enable signal to all the tiles using I2C interface. Once launched, the system will draw more current (depending on the requests from OBC) but in this case, batteries are normally charged by means of solar panels. In case of battery being excessively discharged (e.g. long usage of magnetorquer or RF links), the tile processor turns automatically into low-power mode either disabling or limiting the use of power-hungry elements (e.g. magnetorquer). A software safety mechanism has also been added to properly wake-up the board in case of radiation-induced resets or latches.

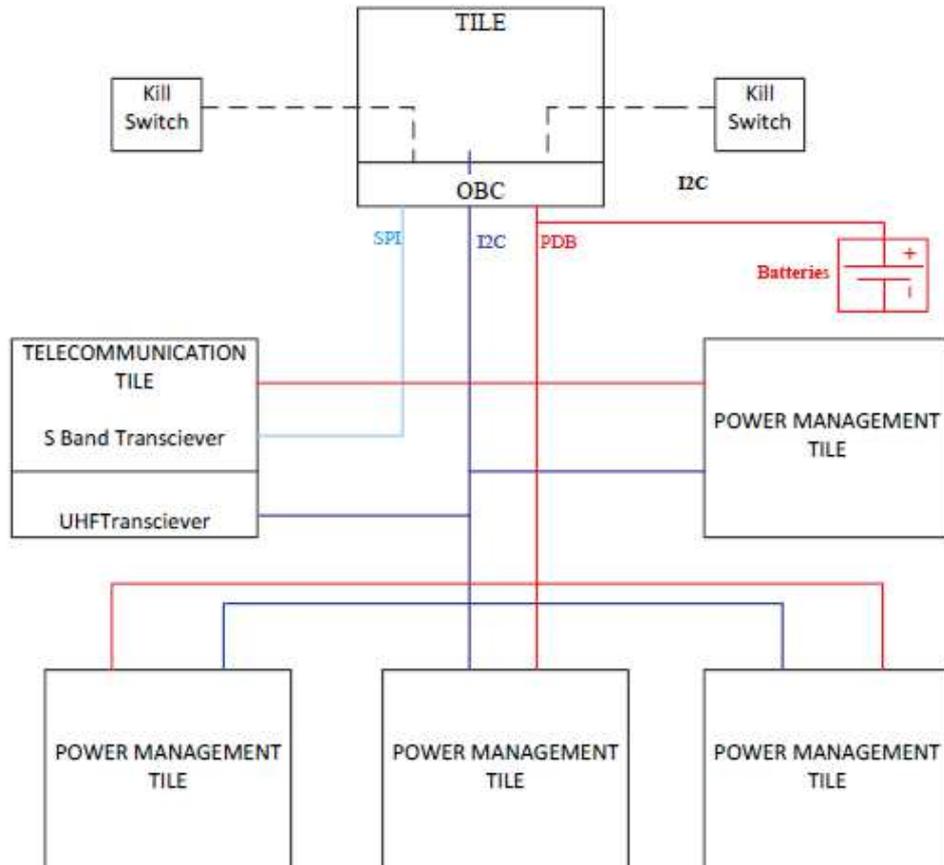


Figure 5.3. System level integration of different tiles through inter tile communication and power distribution bus



Figure 5.4. AraMiS Satellite on Demand configuration fabricated using 1B8 CubePMT and 1B9 CubeTCT

# Chapter 6

## Satellite Design: Physical Module Based Configuration

### 6.1 Introduction

Traditional satellite technologies integrate solar panels, thermo mechanical subsystems, power management, data processing and harness subsystems together in a late stage of design. More recently this approach has become inefficient and its limitation can easily be overcome with modern manufacturing technologies. This chapter proposes an innovative approach to embed power, signal processing and harness together with thermo mechanical subsystem(s) and when required, solar panels. The approach has been developed for the AraMiS architecture for low-cost modular satellites, but it can easily be adapted to other architectures, missions and spacecraft sizes. The proposed approach uses very thin commercial PCBs (0.2 or 0.3mm thick) as the lateral skins for honeycomb structure. The interior side also contains commercial tile processors and plug & play connectors for any desired subsystem placement. The processors implement common functionalities for signal processing, data communication and control operation. The interior side can also host power conversion, for an improved fault-tolerant interface of solar panels with the power management subsystem. A high-performance power distribution bus has also been tested, for a distributed approach to satellite power management. The proposed design uses exclusively the UML diagrams for illustration purpose and software handling of housekeeping data.

## 6.2 Honeycomb Tile

### 6.2.1 Mechanical Structure

Power management and data communication subsystems are the key elements in the design and operation of any spacecraft mission. Spacecraft structure is responsible for protecting the payload and all avionics. In [51], the authors describe efficient and cost effective structures based on honeycomb cores for future spacecraft missions. Determining which structures are the most reliable as well as low weight is very important. In AraMiS honeycomb tile, sandwich structure is used for lighter and more rigid configurations which can meet strength and stiffness requirements of any small sized spacecraft. Honeycomb panels are advanced sandwich elements which consist of lightweight honeycomb core sandwiched between two FR4 or fiberglass face skins [52]. On one face skin of the honeycomb, the PCB is printed with copper implementing interconnections among solar cells and protection diodes. On the other face, the PCB hosts traces for components, modules and one or two commercial tile processors. For spacecraft applications, lightweight Aluminum is used for the honeycomb core as shown in Fig.6.1

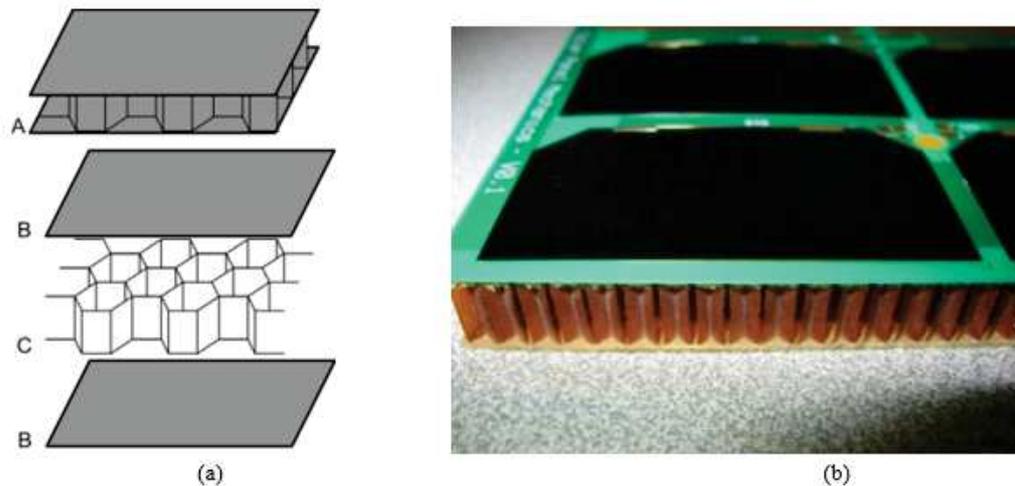


Figure 6.1. Honeycomb Tile (a) A honeycomb core C is a sandwich structure between two face sheets B (b) photograph view of designed tile

The tiles are manufactured in different technology and sizes and can be arranged in any satellite configuration. Each configuration uses a number of tiles connected in either series, parallel or hybrid configurations for different mission requirements. They make the design, integration, manufacturing, testing and maintenance very convenient [53].

## 6.3 Design Modularity

Design reuse and modularity are the keywords of AraMiS architecture. Fig.6.2 shows general architecture of honeycomb tile emphasizing different blocks of power supply voltages and module interface. Dual tile processor and multiple connectors, either

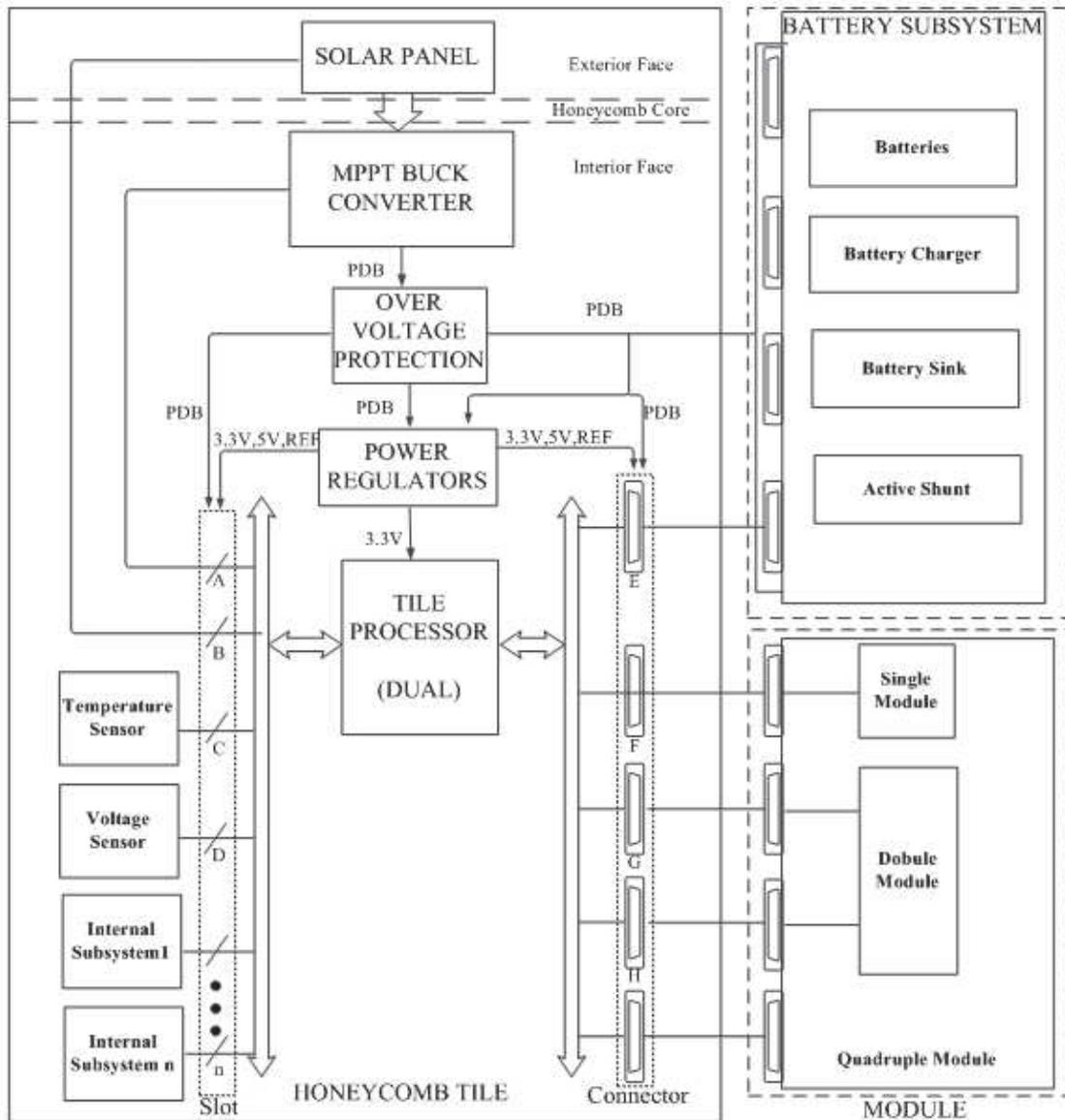


Figure 6.2. Module interfaced with honeycomb tile. Embedded power and data harness is available on slots and connectors and ultimately to all subsystems

physical or logical, are connected on each tile. Every connector has a number of input, output, digital, analog and power channels available through integrated wiring harness. All connectors have these signals available through similar scheme. Any module connected on the tile draws the desired current depending on its requirement but not exceeding the maximum current ratings of the connector. Each connector has capability to be interfaced with communication protocols such as UART, IrDA, SPI and I2C [10]. Internal subsystems of the tile are interfaced with the tile processor via slots and external modules. These modules use spring loaded plug and play connectors. Module subsystems are currently being designed in single, double and quadruple configuration requiring one, two and four connectors respectively [53]. Slots and connectors have same signal scheme from the tile processor. A redundant power bus, capable of power conversion (from solar panels) and power storage (to batteries), is distributed over several tiles. The primary source of power used on each tile is solar panels. Buck converter with maximum power point tracker (MPPT) operates solar panels and converts solar panel power to power distribution bus (PDB) voltage. This PDB is available directly on each connector and also to charge the batteries. A number of switching and linear regulators convert the PDB voltage to low voltage levels (i.e. 3V, 3.3V and 5V) used by all subsystem components. The battery subsystem is a pluggable module further divided into four blocks i.e. batteries, battery charger, battery source and shunt over-voltage protection. Two batteries connected in series (each having 3.6V nominal voltage) are used in any tile to store 7.2V nominal voltage. Certain batteries have been tested to be used in each AraMiS tile. When solar panel is not generating enough power, batteries start sourcing the required power to PDB through battery discharger. An over-voltage protection is provided on each tile to prevent PDB from over-voltage. Each subsystem is managed using UML. A separate UML class with unique attributes and operations has been assigned to each subsystem, either internal or module. The UML class diagram for internal subsystems of honeycomb tile is aimed at generating and optimizing power and handling housekeeping data. A generic naming sequence has been used for each subclass which can be seen in Fig.6.3. The honeycomb power management tile is composed of

- 1B111C\_Solar\_Panel\_HoneyComb is a solar panel array that consists of 14 triple junction, GaAs solar cells (2.2V each) and protection diodes. It converts the solar power to electrical power used by the satellite subsystems. They are connected in series to achieve an output voltage of approximately 30.8V with 0.4A current and 12.32W output power.
- 1B1121C\_Primary\_Switching\_Buck contains the electronic switching components required to convert solar panel voltage (30.8V) to PDB voltage.

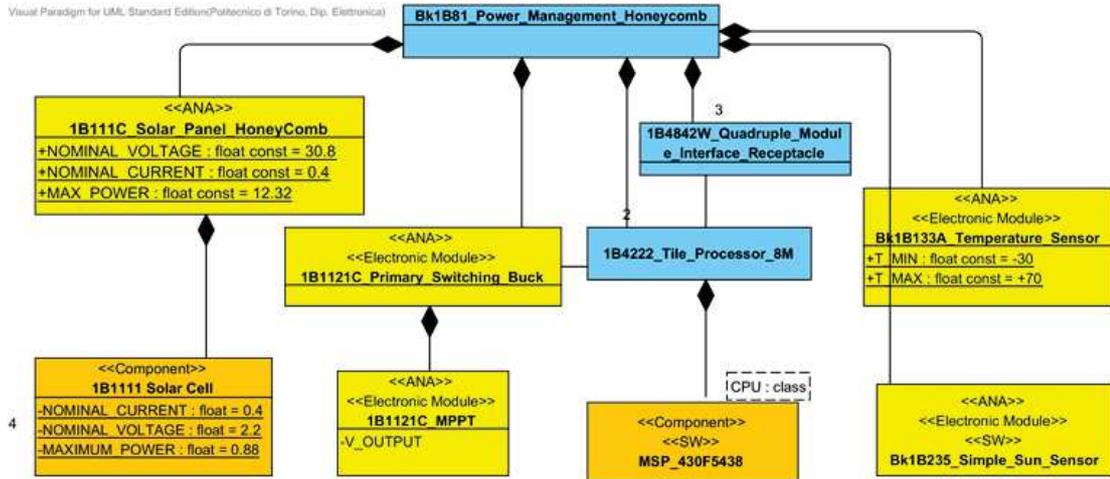


Figure 6.3. Power management class diagram for honeycomb tile showing internal subsystems. Classes have been shown in different color based on their type.

- 1B4222\_Tile\_Processor\_8M hosts one of the tile processors which is a commercial MSP430F5438 [38] controller. It allows communication between slots, connectors and processors onboard the same or other tiles. This processor has one active and six software selectable low-power operation modes. Any desired clock frequency can be generated by use of two external crystal oscillators.
- 1B4842W\_Quadruple\_Module\_Interface\_Receptacle represents the pluggable module connectors which are available on each tile. In general, each tile hosts a number of connectors arranged in single, double or quadruple configurations as shown in Fig6.7.
- Various sensors depicted in Bk1B133A\_Temperature\_Sensor and Bk1B235\_Simple\_Sun\_S

The next sections describe power converters and distributed power management for the designed tiles

## 6.4 Power Supply Converters

The board has different levels of distributed power bus. This section details about the power regulators that have been used for different types of switching and linear power supplies. Fig.6.4 shows the block diagram of different types of power supplies onboard the tile. Switching and linear regulators have been utilized to output the fixed voltage and distribute all across the subsystems and pluggable modules. The

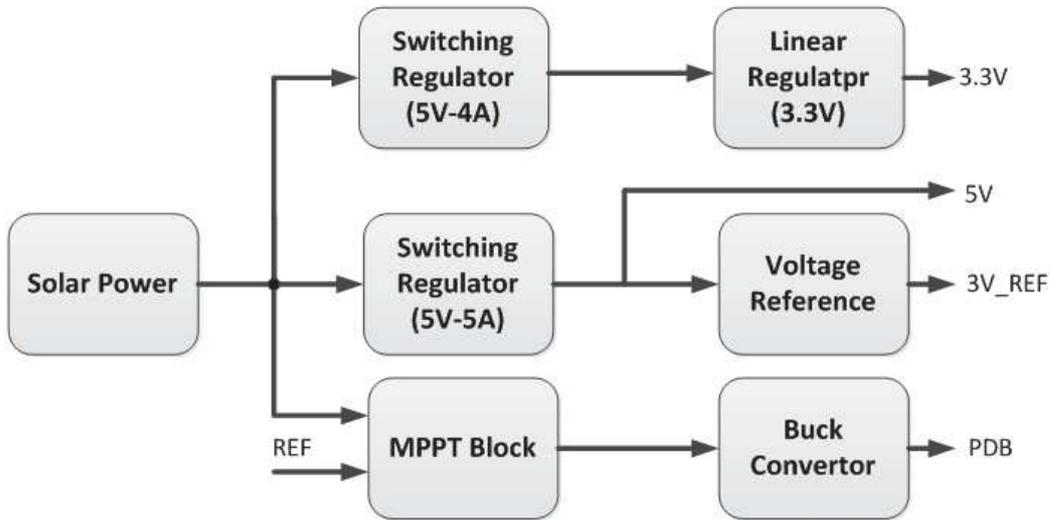


Figure 6.4. Block diagram of power supply converters

switching regulators used in design are TPS5450 from Texas Instruments (TI). Any desired output voltage can be obtained from the Fig.6.4 by using the appropriate circuitry. The 3.3V supply uses a two stage mixed regulator. The first stage is high efficiency switching regulator that converts solar power to 5V and the second stage is a linear regulator with stable output to be supplied to the processors and components needing stable supply. The schematics of different regulators are attached in annex section. The schematic of Buck converter is shown in the Fig.A.10. The Buck block converts the high solar panel voltage to Power Distribution Bus level, which is spread to the rest of the system. The switching of Buck converter for maximum power point tracking is controlled using an MPPT block which consists of a hysteric comparator and a MOS driver block. The comparator output is used to control the MOS driver pulses depending on the duty cycle. The simulation results of the designed converter are shown in Fig.6.6. The input voltage from the solar cells is shown in the blue trace which may be as high as up to 30V for the honeycomb tile. The outputs of hysteric comparator and the MOS driver block are depicted in light and dark green traces respectively. The red trace represents the output voltage generated by the proposed system. The output voltage rises until the power distribution bus voltage level. The dimensions of each honeycomb tile are 16.5cmx33cm, having 14 solar cells in series and two Li-Ion rechargeable batteries on a separate quadruple

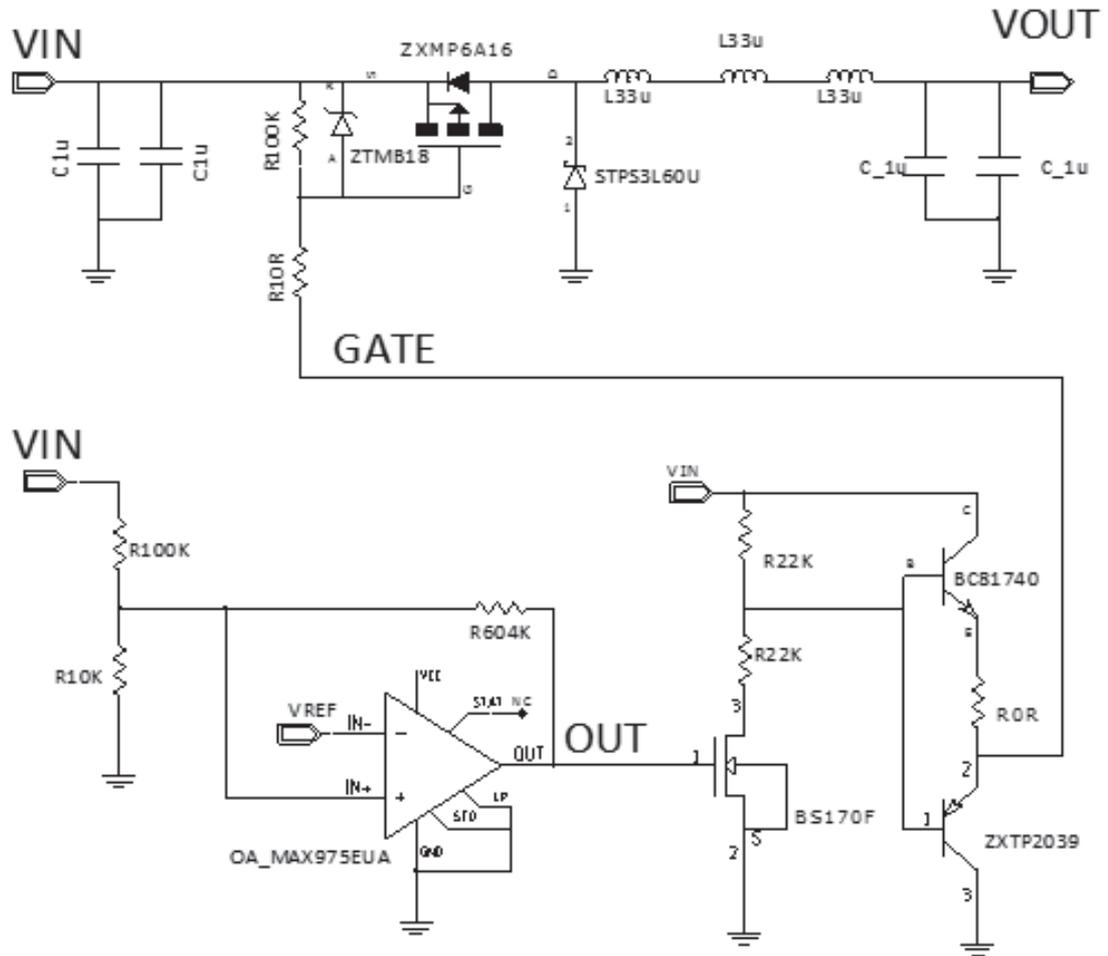


Figure 6.5. Block diagram of Buck converter with MPPT.

module. It offers 30W continuous power and about 15Wh storage capability (considering two batteries of 2200mAh and 3.6V nominal voltage). The printed circuit board (PCB) layout (both top and bottom skins) of the fabricated honeycomb tile, emphasizing the power regulators, pluggable connectors and tile processors is shown in Fig.6.7. It shows the dual tile processors interfaced with internal and external subsystems through slots and connectors respectively. The tile has been designed in modular manner with different modular connectors hosting necessary subsystem together with power and data harness in a distributed way. The designed approach gives more flexibility in the pluggable module subsystem selection and a lot more empty space for the payload. The schematics and printed circuit board layout of

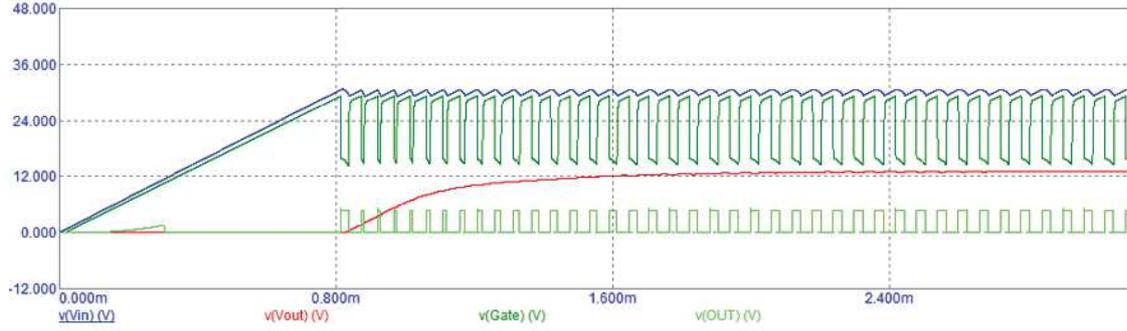


Figure 6.6. Simulation results of MPPT Buck convertor.

most of the subsystems has been embedded in the main tile reducing harness complexities. The space grade spring loaded connectors host additional mission specific subsystems designed on small module boards and the main tile provides power and data distribution functions, power and data harness and mechanical support to these subsystems. The harness is reduced significantly at system level using the proposed design approach because the design uses smart harness approach to embed power and data processing complexities at tile level [53, 54]. Fig.6.8 shows the mock-up of a spacecraft architecture in which the tiles have been assembled together in a hexagonal shaped spacecraft structure with optical telescope payload. There is enough space available for batteries, pluggable subsystems and the payload.

## 6.5 Distributed Power Management

Various subsystems interact with the PDB in such a way that power generated by all illuminated solar cells is autonomously gathered and supplied either to payload or stored into any empty energy storage unit. The power management and distribution functions have been assigned several UML actors. An actor is an entity which interacts with the system and performs many tasks. Fig.6.9a shows Central Power Management actor which is a software routine and controls several power management functions such as enable or disable of primary source, battery source, battery charger, active shunt and load actors when the satellite is on the fly. The static I-V characteristics of each actor are shown in Fig.6.9b, with specific operating voltages.

According to KCL, sum of current sourced by primary source ( $I_{source}$ ) and current sunk by load ( $I_{sink}$ ) is zero and given by (6.1)

$$I_{source} + I_{sink} = 0 \quad (6.1)$$

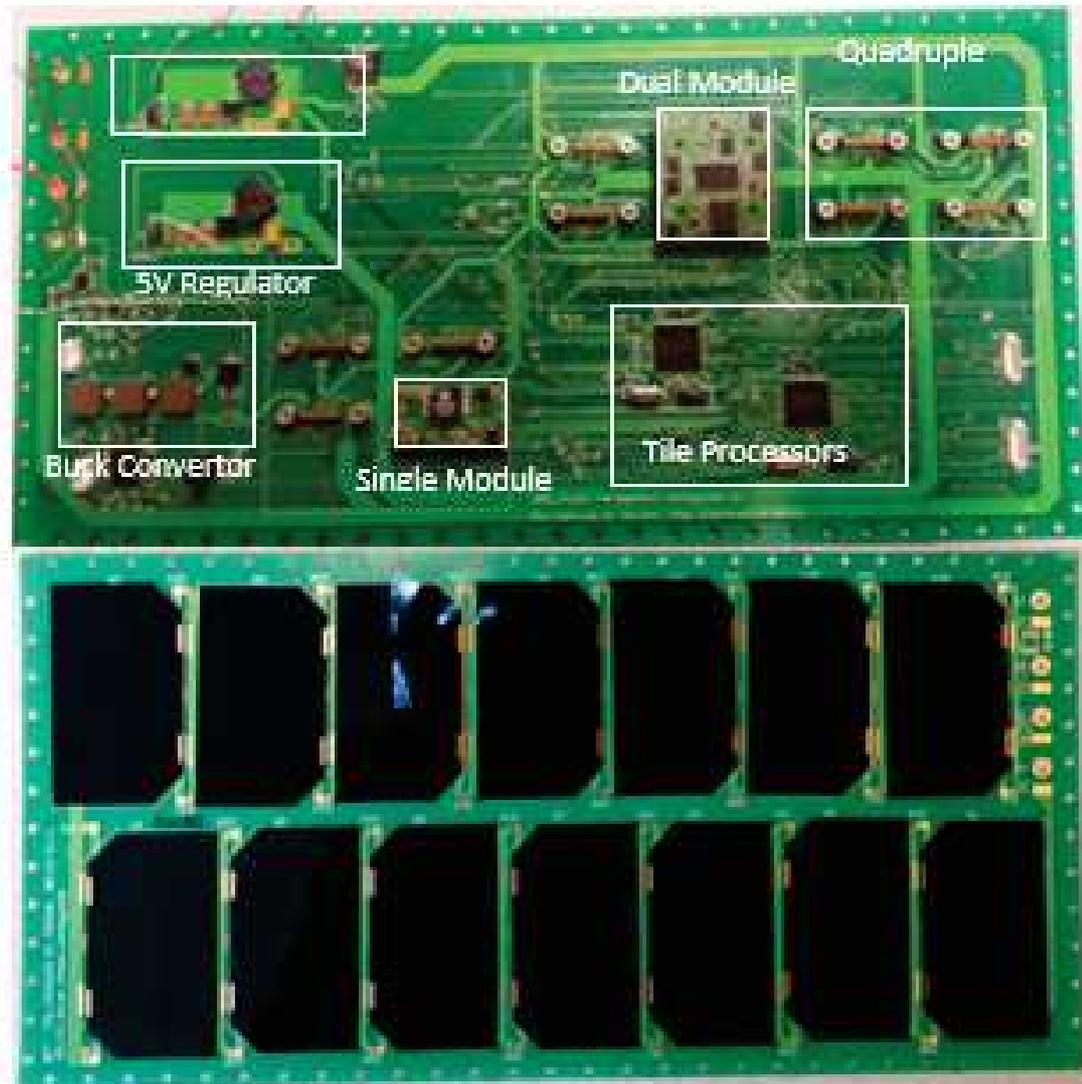


Figure 6.7. Block diagram of Buck converter with MPPT.

In case the current generated by primary source is more than the current sunk by load, the additional current is used to charge the batteries according to a defined priority.

The *Primary Source* actor, mainly solar panel, is the major source of energy. It converts light energy into electrical energy and sources as much power as possible on PDB at a voltage in the range of 12V to 16V. It has two states.

- *Idle*, when not enabled to source power.



Figure 6.8. Honeycomb tiles assembled in hexagonal shaped spacecraft with optical telescope payload (mock-up)

- *Active*, when it is enabled to source power.

The static I-V output characteristic of this actor is hyperbolic and provides nearly constant power as shown in Fig.6.9b. The generated power depends upon the light energy.

The *Battery Source* actor sources the required amount of electrical power on PDB at a voltage up to 14V when primary source is not able to generate enough power i.e. when the primary source current decreases below  $I_{max}$  limit. The battery source has three states.

- *Idle*, when it is not enabled to source power.
- *Active*, when it is enabled to source power.
- *Empty*, when internal energy storage is empty.

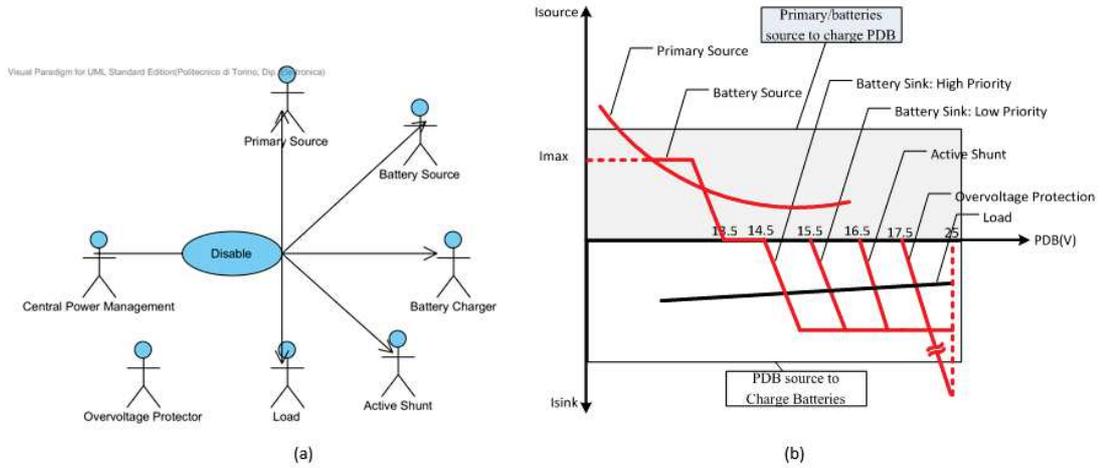


Figure 6.9. (a) Power management actors controlled by central power management  
(b) Static I-V output characteristics of power management subsystem

The static I-V output characteristic of this actor in the *Active* state is linear and given by (6.2)

$$I = \frac{V_r - V}{R_r} \quad (6.2)$$

with the constraint  $I_{sink} = I = I_{max}$ ;  $I_{max}$  depends on the characteristics of the energy storage.  $I_{sink}$  is the current sunk by batteries from PDB, therefore negative ( $-1mA = I_{sink} = 0$ ). Available voltage on PDB is represented by  $V$  and required voltage by  $V_r$ . When primary source is not generating enough power, batteries start to source power to PDB. As soon as the required voltage limit  $V_r$  is achieved, *battery source* is disabled by *central power management*. This situation is shown in the Fig.6.9(b). When *Battery Source* actor is either in *Idle* or *Empty* states, it sinks less than 100uA average and 1mA peak current.

*Battery Charger* actor stores the excess energy from PDB (at a voltage in the range of 14V to 16V) into the batteries. The sink current depends on the capacity of the battery. *Battery Charger* actor has four states.

- *Idle*, when it is not enabled to sink power, therefore not charging batteries.
- *High Priority Charging*, when it is sinking all the surplus energy. All high priority charger actors have priority over low priority *Battery Charger* actors.
- *Low priority Charging*, when it is sinking all the available energy (if any) which is not used by other high priority *Battery Charger* actors.

- *Full*, when the battery is fully charged.

The static I-V output characteristic of this actor in either low priority charging or high priority charging is linear given by (6.3).

$$I = \frac{V - V_r}{R_r} \quad (6.3)$$

with the constraint  $0 = I = I_{max}$ . The battery charging threshold,  $V_r$ , for high priority charging is lower than that of the lower priority charging. In general, for our design,

- $V_r = 14.5\text{V}$  for high priority charging
- $V_r = 15.5\text{V}$  for low priority charging

In *Idle* or *Full* states, the *Battery Charger* actor sinks less than 100uA average and 1mA peak current.

*Active shunt* dissipates the available energy which is not used, into the internal shunt resistor for heating purpose. The static I-V output characteristic of this actor is linear and given by (6.3) with the constraint  $I = 0$ . Fig.6.9 b shows the voltage  $V_r$  for active shunt which in this case is 16.5V for our application.

*Over voltage protector* starts dissipating power when the PDB voltage reaches 17.5V and keeps it under 25V (maximum PDB limit). This is an autonomous protection system that can't be enabled or disabled by *central power management*. The *load* actor represents internal load and the payload that sinks required electrical power from power distribution bus. The static I-V output characteristic of this actor is nearly constant current as shown in Fig.6.9(b). The working point of power distribution bus depends on the active load. If load draws less current, the working point shifts to the primary source. In case load draws more current, then the working point shifts to the battery source in addition with the primary source to source.

#### subsectionOperation of Power Distribution Bus

The proposed power distribution bus for AraMiS is used to interconnect a large number of tiles together in any of the foreseen configurations in a complete modular way. Fig.6.10 shows the distributed power management in which several tiles have been connected through PDB. 6.3. Solar panel, batteries and load are also represented in the figure.

*Primary source* is illuminated by the sun in a number of ways. The generated power by solar panel depends on the incident angle ( $\theta_i$ ) of sun light onto the face of the satellite and is given by (6.4).

$$V = V_{max}\sin\theta_i \quad (6.4)$$

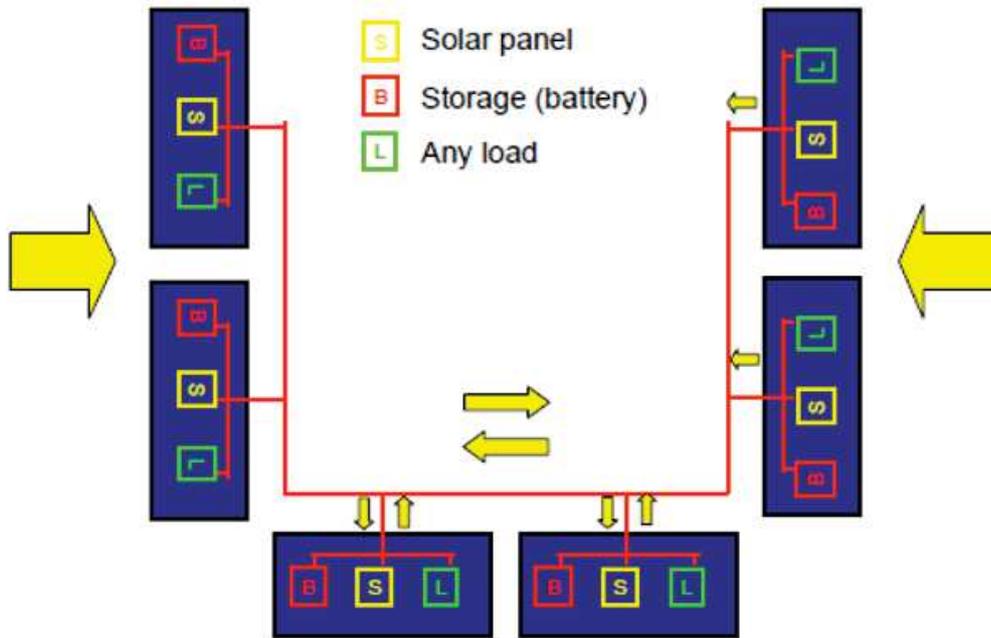


Figure 6.10. Illustration of different tiles connected with power distribution bus. Solar panel, storage and load for each connected tile is shown.

- In case if single face of the satellite is illuminated ( $\theta_i = 90^\circ$ ); the output power will be  $V_{max}$ .
- In case when two faces are illuminated ( $\theta_i = 45^\circ$  for each face); the output power generated will be  $\sqrt{2} \cdot V_{max}$ .

In both of above cases, the corresponding batteries of the tile(s), being high priority, are charged. Power may also traverse to the active subsystems of the corresponding tile (e.g. attitude actuator or payload). In case generated power is surplus, either load of other tiles starts to sink power through power distribution bus or corresponding low priority batteries start charging. If all batteries are charged, while solar panel still generating the power, and no other subsystem requiring any power then the excess power is dissipated to the shunt resistors located on the dark surface of the satellite to generate heat. In all other cases, over voltage protection circuit shunts the additional power to ground.

In case the satellite is in night (40% of a satellite day on average in LEO), no solar panel is generating power, then one or more batteries start sourcing the required

power to the active load in the pre-mentioned scheme. The *central power controller* may decide at any moment which batteries are allowed to source power and which ones are kept isolated in order to save energy for later use.

#### subsectionSimulation Results

The generic configuration of proposed power bus was simulated and the results are shown in Fig.6.11. It takes into account the *primary source*, *battery charge*, *bat-*

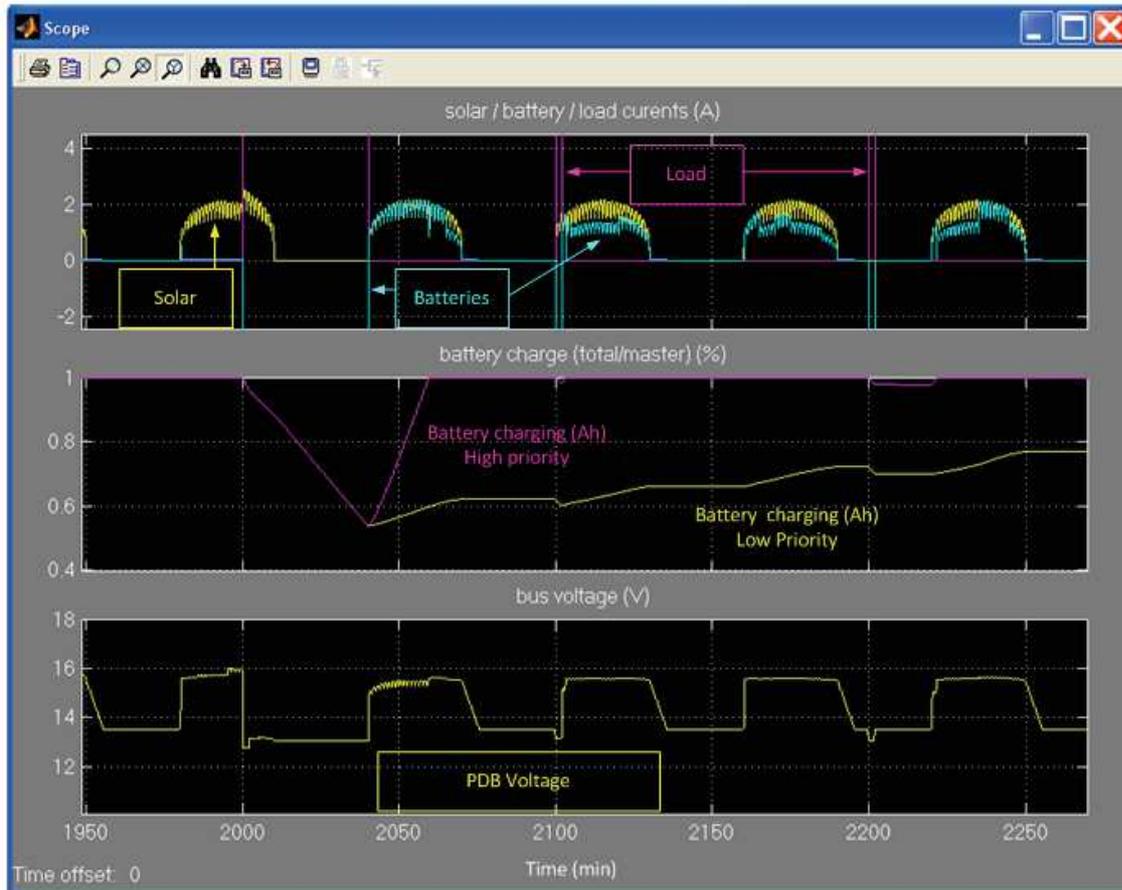


Figure 6.11. Simulation analysis of proposed power bus.

*tery source* and the *load*. Current generated or drawn by simulated *primary source*, *battery source*, *battery sink* and *load* is shown in the upper portion of Fig.6.11. *High* and *low priority battery charging* is shown in the middle portion and unregulated power distribution bus voltage is shown in the lower portion. Mostly, batteries are fully charged, but their voltage level decreases when the load draws current. Comparison of high and low priority battery charging is also evident from the figure.

Power distribution bus voltage is unregulated, therefore it fluctuates between 12 and 18V depending upon charge/discharge periods.

## 6.6 Housekeeping Management of Honeycomb Tile

UML design approach has been used for communication and housekeeping functions at tile level as well as satellite level [33–36]. Each tile houses a smart data handling subsystem for communication between one or more subsystems or tiles, which is based on multiple protocols including RS232, I<sup>2</sup>C, IrDA and SPI. Every tile uses a complete modular approach for the recognition and software control of a particular module when it is plugged.

The use case diagram in Fig.6.12 shows the detailed power management functions and their handling through a number of use cases. The actors interact with the

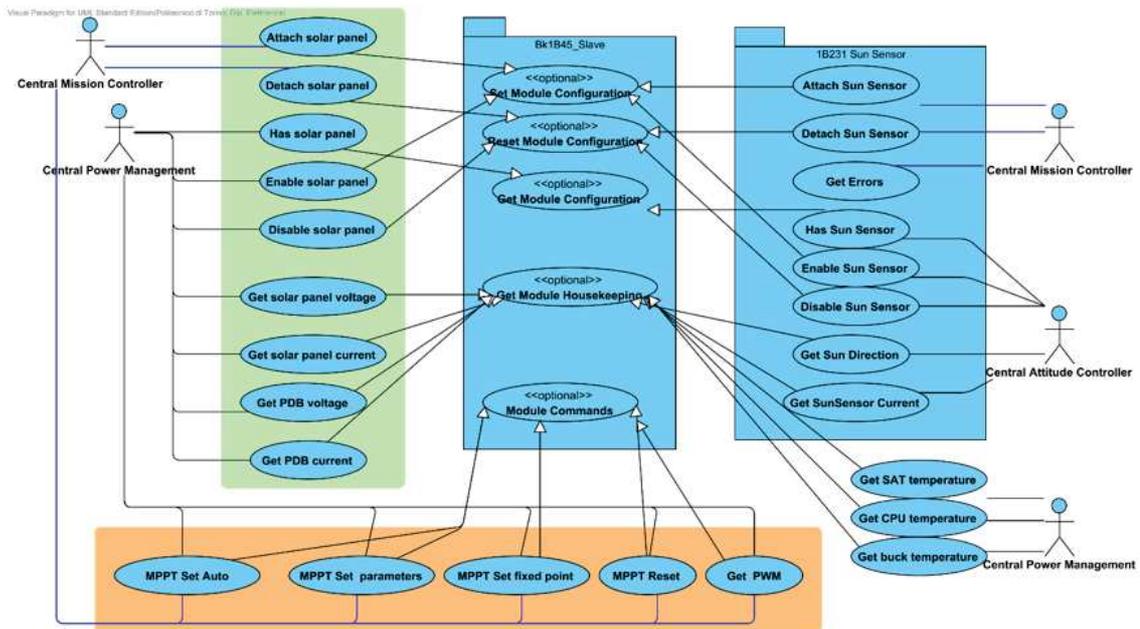


Figure 6.12. Housekeeping use case diagram of honeycomb tile showing different use cases and actors.

system and perform many tasks. Actors *central mission controller*, *central power management* and *central attitude controllers* are entities (software routines running on the central processor) in charge of satellite supervision, fault detection emergency

management and power management in nominal operation. All other use cases implement most of the power management housekeeping functions such as *enable/disable* solar panel, get voltage and current levels of solar panel, PDB and housekeeping sensors, set maximum power point parameters for MPPT Buck converter etc. Each use case is inherited by corresponding class partially from Fig.6.3. As an example *MPPT set parameters*, *MPPT set auto* and *MPPT reset* are inherited by class *1B1121C\_MPPT*.

The main job of class *Housekeeping* is the autonomous storage of mission-dependent housekeeping data, inside vector *housekeeping*. Each software module driver periodically stores the values of their housekeeping in this vector. This vector is contained inside class *Housekeeping*. Attributes and operations of *Housekeeping* class correspond to status, configuration, calibration, data acquisition, history and error information of different modules. Housekeeping data is processed based on different commands from the processor. The peripherals should react according to these commands. Some user defined commands are *CMD\_READ\_DATA*, *CMD\_WRITE\_DATA*, *CMD\_COMMAND*, *CMD\_GET\_HOUSEKEEPING* etc. Each command has specific value and is allocated specific memory location. Some commands are predefined, while others can be added by the designer. Detailed description of the *Housekeeping* class and acquisition of data based on different commands using UML approach is described in [34].

The software control of every module and storage of mission specific data becomes very easy using this approach as soon as every module is configured properly. A sample UML generated code is shown where voltage, current, power and working point of MPPT of the Buck switching module has been acquired, calculated and stored inside vector *housekeeping*. This requires the solar panels *attached* and *enabled* by the *central mission controller* and *central power controller* actors respectively.

```
#include "Bk1B1121CS_Primary_Switching_Buck.h"
template <class HK, class moduleA, class moduleB> \
  Bk1B1121CS_Primary_Switching_Buck<module>::housekeeping(ushort index)\
  if (HK::ConfigRegister[CONFIG_STATUS_WORD] & (ATTACHED || ENABLED))
{
    switch (index) {
        case V_SOLAR:
            moduleB::A0.acquire(HK::housekeeping[V_SOLAR]); \
            // connected to analog channel A0 of tile processor through slot.
            break;
        case I_SOLAR:
            vector at I_SOLAR address
            moduleA::A0.acquire(HK::housekeeping[I_SOLAR]);
```

```
    break;
    case V_PDB:
        moduleB::A1.acquire(HK::housekeeping[V_PDB]);    break;
    case I_PDB:
        moduleA::A1.acquire(HK::housekeeping[I_PDB]);
        break;
    case MPPT_PWM:
        HK::housekeeping[MPPT_PWM]= evaluate_PWM();
        if (MPPT_tracking_enable == true) {
            moduleA::PWM.SetDutyCycle(HK::housekeeping[MPPT_PWM]);
        } else {
            moduleA::PWM.SetDutyCycle(fixed_PWM);
        }
        break;
    }
}
```

By use of approach, all the housekeeping data is managed with a lot of ease and in a completely flexible way.

## 6.7 Conclusion

This Chapter has analyzed an innovative tile based structure for small satellites with integrated data, power subsystems and signal processing capabilities which contains different modules that are pluggable only when required. Extensive battery test has been done and the best of the available batteries is selected in the design. An innovative approach to distributed power management in small satellites has been discussed. All the housekeeping management has been done using UML which makes the control of every module quite easy. In the end, the tile is a complete modular and flexible in design and be used in any desired configuration.



# Chapter 7

## Intra-Satellite Communication

### 7.1 Introduction

The on-board communication is one of the main issues in electronics system level development while working on university satellite mission.

This chapter highlights all the basic protocol for AraMiS which is mostly based on 1B48 module interface at physical level and 1B45 on board data handling at protocol level. The 1B45 contains a set of resources, functions and protocols which are intended to effectively support inter- and intra-tile communication functions between several entities inside the AraMiS architecture.

The basic protocol supports communications between the following pairs of elements.

- for inter-tile communication: the **OBC** (Communication master) and one or more tiles (Communication slave)
- for intra-tile communication: the tile processor and one or more Active AraMiS Modules.

Communication can take place using either the OBDB protocol, the IrDA protocol, the I2C protocol, the RS232 protocol, the SPI protocol, or the Wireless protocol or the IntraBoard protocol.

### 7.2 Requirements

- Send/Receive designer-defined messages to/from the Slave (namely, Read Data/Write Data operation);
- Send Designer-defined commands to the Slave (namely, Command Only operation);

- Acquire Designer-defined housekeeping information from the Slave (e.g. internal voltages, currents, temperatures; see use case Housekeeping Management);
- Set the Slave to sleep mode or wake it up
- Configure the Slave (e.g. enable/disable subsystems)
- Broadcast Write Data - when a Master wants to transfer up to 256B of data to all Slaves;
- Broadcast Command Only - when a Master wants to deliver a data-less command to all Slaves;
- WriteRead Data - when a Master wants to transfer up to 256B of data bidirectionally to/from a Slave.

### 7.3 1B45 Usecases

On-board Data Handling (OBDH) and Software On Board Data Handling Subsystem is one of the critical subsystems of any satellite mission. A lot of design effort needs to be done for reliable way of data handling among different modules (sensors, actuators, etc.). The following sections show the detailed documentation On Board Data Handling and Software sections of AraMiS. On-board Data handling Bus All the data handling of AraMiS takes place in accordance with the Basic Protocol for communication between one or more subsystems developed at Politecnico Di Torino, which supports in a transparent way, several physical layers. The Master is any processor willing to communicate (exchange data and commands) with a Slave. Master may either be the On Board Computer or a Tile Processor (MSP430F5438A from Texas Instruments). The Slave may be a Tile or any other subsystem. Master communicates with the Slave via either SPI, OBDB, or I2C protocols. The Master, via the interface, can at least:

- Send/Receive designer-defined messages to/from the Slave (namely, Read Data/Write Data operation);
- Send Designer-defined commands to the Slave (namely, Command Only operation);
- Acquire Designer-defined housekeeping information from the Slave (e.g. internal voltages, currents, temperatures; see use case Housekeeping Management);
- Set the Slave to sleep mode or wake it up
- Configure the Slave (e.g. enable/disable subsystems)

The Basic Protocol supports the following actions

- Write Data - when a Master wants to transfer up to 256B of data to a Slave;
- Read Data - when a Master wants to read up to 256B of data from a Slave;
- Command Only - when a Master wants to deliver a data-less command to a Slave.
- Broadcast Write Data - when a Master wants to transfer up to 256B of data to all Slaves;
- Broadcast Command Only -Command Only - when a Master wants to deliver a data-less command to all Slaves.

Most data transfers contain an appropriate START element; an 8-bit Master address; 8-bit Slave address ; a 16-bits command; an 8-bit data length field (only for Write Data and Broadcast Write Data); data (1B to 256B; except read data); a 16-bit CRC check and an appropriate STOP element. For AraMiS-C1, we select I2C physical layer for all connections. On-board Data Handling Functions The tile software is also modular and allows a quick adaptation to specific subsystems. The basic software for the CPU is properly hardened to guarantee high level of radiation tolerance at very low cost. The software has been completely defined, specified, designed and documented using UML tool. A variety of UML diagrams (use case, class, sequence, requirement, etc.) have been utilized for the development of each subsystem.

Fig.7.1 shows the top level Data Handling case diagram of an AraMiS tile. This package states information about the satellite modules, acquisition of data from different sensors, actuators etc. to the main processor, and status and configuration of health and safety of different modules onboard.

The above package contains all specifications of the Slave (Module) side of the Subsystem Serial Data Bus. The Master actor is any processor willing to communicate (exchange data and commands) with a Slave i.e. with any AraMiS subsystem. The Master communicates with the Slave via either of the foreseen protocols (I2C protocol, OBDB protocol, SPI protocol) and performs either of configuration and status management, Housekeeping management, user defined messages and commands, supervision and emergency recovery use cases.

### 7.3.1 Configuration and status management

This is a group of use cases aimed at transferring either any mission dependent status information from any Tile to the Master, or configuration information from the Master to any Tile.

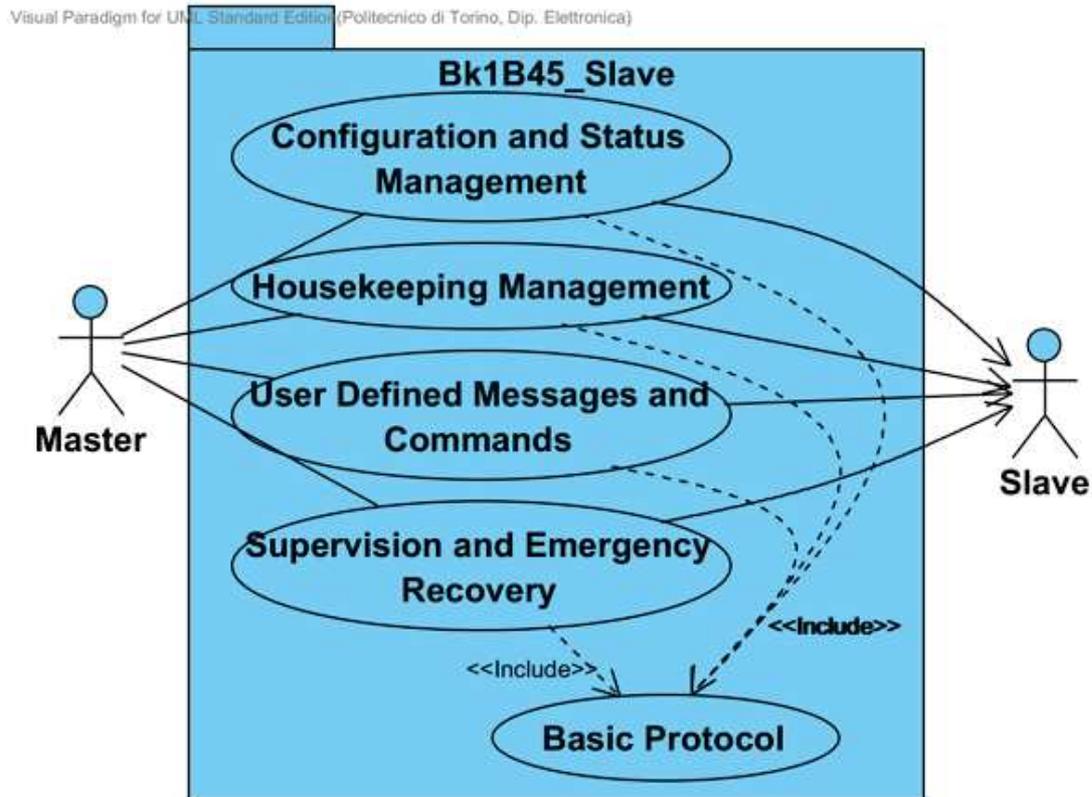


Figure 7.1. Possible Types of Faults

### 7.3.2 Housekeeping Management

This is a group of use cases aimed at returning the history of either all or a subset of the housekeeping parameters, last measured housekeeping data, Housekeeping sensor calibration etc. for all subsystems. Class Housekeeping is aimed at autonomously storing mission-dependent housekeeping data, inside a vector housekeeping. Each AraMiS - C1 module driver periodically stores the values of their housekeeping in this vector. The content of the housekeeping vector is then managed by the Housekeeping Management use case. Attributes and operations of Housekeeping class correspond to status, configuration, calibration, data acquisition, history and error information of different modules

### 7.3.3 User defined messages and commands

In this group of use cases, the Master, via the interface, can send Designer-defined messages to the slave namely; write module data operation, receive Designer-defined messages from the slave namely; read module data and send Designer-defined commands to slave namely; module commands operation.

### 7.3.4 Supervision and Emergency Recovery

This a group of use cases which continuously monitors the correct operation of the system and, in case of the program gets either stuck or corrupted, takes appropriate actions (e.g. rebooting CPU or reloading program). In order to save power, it puts the system into a standby mode where, for instance:

- internal processor is in sleep mode but is able to listen to the bus
- housekeeping sensors are either disabled or unpowered
- other Designer-defined hardware is either disabled or unpowered

Fig.7.2 shows the generalization of Basic Protocol use case. The Basic Protocol is based on different use cases for implementation of 1B45\_On-board data communications. These use cases include Command Only, Read Data, Write Data, Broadcast Command Only and Broadcast Write Data

The Basic Protocol is then implemented in a number of physical implementations: SPI protocol, RS232 protocol, IrDA protocol, OBDB protocol, I2C protocol, Wireless protocol and intra-Board protocol, which differ for the details of the physical support and data rate.

## 7.4 Tile Processor

There is possibility of using one of the several processors in the design the most recent configurations use a 100 pin MSP430F5438A as tile processor. It contains a complete asynchronous communication peripheral called USCI (Universal Serial Communication Interface) which allows executing the tasks of asynchronous transmissions and receptions.

The USCI modules support multiple serial communication modes including UART, SPI and I2C communication. In UART mode, the USCI\_Ax modules connect the device to an external system via two external pins, UCAxRXD and UCAxTXD. The USCI transmits and receives characters at a bit rate asynchronous to another device. Timing for each character is based on the selected baud rate of the USCI. The transmit and receive functions use the same baud-rate frequency. UART mode features include

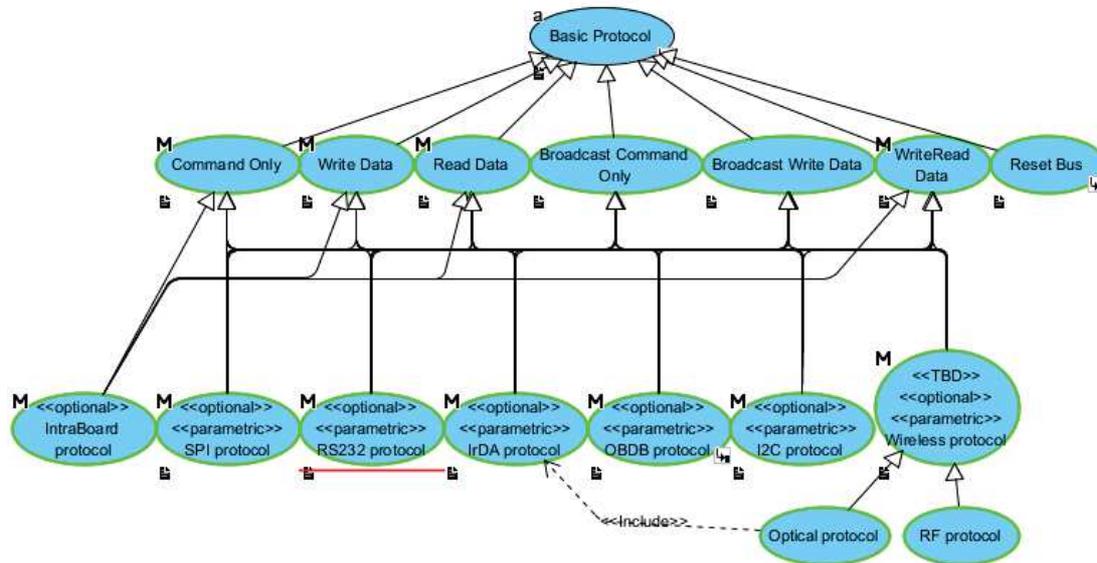


Figure 7.2. Illustration of Basic Protocol

- 7- or 8-bit data with odd, even, or non-parity
- Independent transmit and receive shift registers
- Separate transmit and receive buffer registers
- LSB-first or MSB-first data transmit and receive
- Built-in idle-line and address-bit communication protocols for multiprocessor systems
- Receiver start edge detection for auto wake up from LPMx modes
- Programmable baud rate with modulation for fractional baud-rate support
- Status flags for error detection and suppression
- Status flags for address detection
- Independent interrupt capability for receive and transmit

Each Tile of AraMiS-C1 embeds central processing units with A/D converters embedded with the harness, for handling from sensors, storage, signal processing,

and housekeeping functions. Commercial MSP430F5438A ultra low power 16-Bit RISC Architecture microcontroller is used as Tile Processor which can support up to 25 MHz system clock. The MSP430F5438A has one active mode and six software selectable low-power modes of operation. An interrupt event can wake up the device from any of the low-power modes, service the request, and restore back to the low-power mode on return from the interrupt program. The main functions performed by the tile processor to implement the module software are following

### Digital I/O

There are up to ten 8-bit I/O ports implemented: For 100-pin options, P1 through P10 are complete. P11 contains three individual I/O ports.

- All individual I/O bits are independently programmable.
- Any combination of input, output, and interrupt conditions is possible

### Oscillator and System Clock

The clock system in the MSP430x5xx family of devices is supported by the Unified Clock System (UCS) module that includes support for a 32-kHz watch crystal oscillator (XT1 LF mode), an internal very-low-power low-frequency oscillator (VLO), an internal trimmed low-frequency oscillator (REFO), an integrated internal digitally controlled oscillator (DCO), and a high-frequency crystal oscillator (XT1 HF mode or XT2). The UCS module features digital frequency locked loop (FLL) hardware that, in conjunction with a digital modulator, stabilizes the DCO frequency to a programmable multiple of the selected FLL reference frequency. The internal DCO provides a fast turn-on clock source and stabilizes in less than 5  $\mu$ s. The UCS module provides the following clock signals:

- Auxiliary clock (ACLK), sourced from a 32-kHz watch crystal, a high-frequency crystal, the internal low-frequency oscillator (VLO), the trimmed low-frequency oscillator (REFO), or the internal digitally controlled oscillator DCO.
- Main clock (MCLK), the system clock used by the CPU. MCLK can be sourced by same sources made available to ACLK.
- Sub-Main clock (SMCLK), the subsystem clock used by the peripheral modules. SMCLK can be sourced by same sources made available to ACLK.
- ACLK/n, the buffered output of ACLK, ACLK/2, ACLK/4, ACLK/8, ACLK/16, ACLK/32

### **Timer**

This device contains 16-bit timer/counters TA0, TA1, TB0 with five, three and seven capture/compare registers respectively. It can support multiple capture/compares, PWM outputs, and interval timing. It also has extensive interrupt capabilities. Interrupts may be generated from the counter on overflow conditions and from each of the capture/compare registers.

### **Analog to Digital Converter**

The ADC12 module supports fast 12-bit analog-to-digital conversions. The module implements a 12-bit SAR core, sample select control, reference generator, and a 16-word conversion-and-control buffer. The conversion-and-control buffer allows up to 16 independent ADC samples to be converted and stored without any CPU intervention.

### **Universal Serial Communication Interface**

The USCI modules are used for serial data communication. The USCI module supports synchronous communication protocols such as SPI (3 or 4 pin) and I2C, and asynchronous communication protocols such as UART, enhanced UART with automatic baudrate detection, and IrDA. Each USCI module contains two portions, A and B. The USCI\_An module provides support for SPI (3 pin or 4 pin), UART, enhanced UART, or IrDA. The USCI\_Bn module provides support for SPI (3 pin or 4 pin) or I2C.

## **7.5 Implementation**

The intra-satellite data communication has been realized using 1B48 module interface drivers at physical level and the top level basic protocol of 1B45 on board data handling subsystem.

# Chapter 8

## Module Design: 1B411 On-board Data Bus Module

### 8.1 Introduction

This chapter illustrates general data communication architecture of [AraMiS](#) project and inter-tile data communication bus interface structure of the small modular satellites. The dual redundant data bus will provide exchange of information in almost all operative conditions and connect all the electronic peripherals, supporting the data exchange speed required by the different sub-systems composing the satellite. The proposed architecture uses differential bus with galvanic isolation among all the connected modules in order to have fault tolerance, good signal integrity and to connect as many tiles in series or in parallel. The proposed structure has been tested in almost all possible fault conditions and the simulation and actual results show reliable transmission and reception of signal from one module to the other even under different types of fault conditions. System stability is guaranteed by the proposed characteristic of each subsystem in such a way that any number of modules can be connected to the communication bus without impairing overall performance and stability. This work presents the bus interface circuit located between the processor and the communication channel and also the fault tolerance analysis with different possible types of faults.

AraMiS employs [IrDA](#) Return to Zero Inverted ([RZI](#)) encoding scheme with the proposed circuit. It is a coding format in which the encoder sends a pulse for every zero bit in the transmit bit stream. This coding scheme has been used because it is compatible with pulse transformers and it is easy to implement since it is the coding scheme used by IrDA line and available in many microprocessors. Next sections describe the bus interface structure [\[46\]](#) that will ensure data to be transmitted on differential bus correctly.

## 8.2 Cross Bus

Each tile incorporates solar panels on the external side and basic power routing, data routing and signal processing capabilities on the internal side. Multiple tiles interface through a proprietary self-configuring, dual-redundant, distributed power and data bus. By using modular tiles each having power and data routing capabilities, shape of the satellite can be configured according to payload needs. If payload requires, say e.g., 28V instead of typical tile voltage 14V, tiles can be added in series to have this voltage. If more current is needed, parallel configuration can be used. Usually power and data outputs of each tile use the same ground and the reference voltage level may be different between two tiles. Fig.8.1(a) shows typical way of interconnecting two tiles in series in order to obtain twice the tile voltage at the output. Let's assume

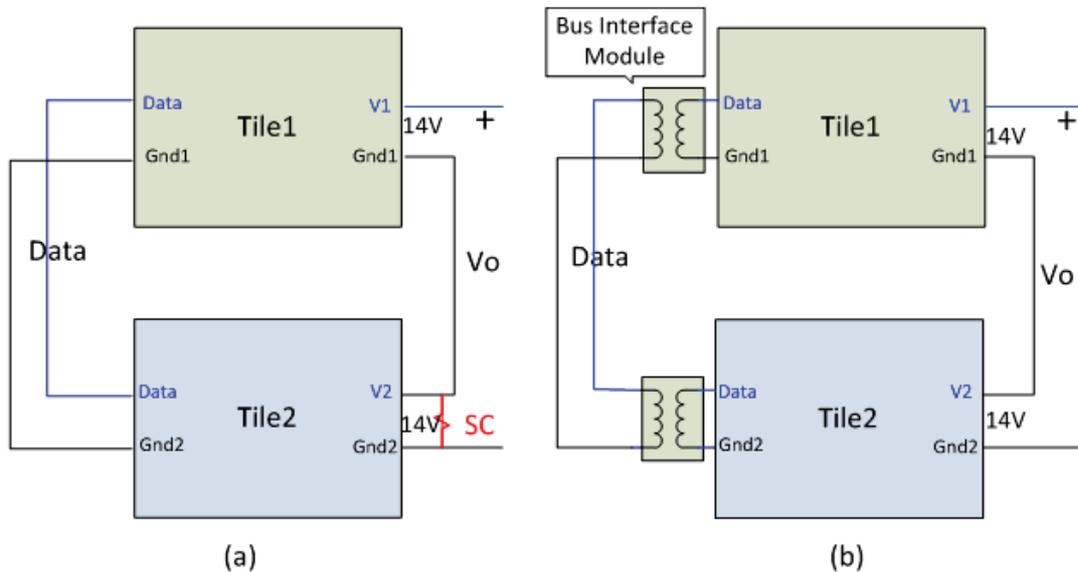


Figure 8.1. Interconnecting two tiles in series (a) Typical (b) Proposed approach

that any data bus that uses common grounds between modules is used. Connecting  $Gnd1$  and  $Gnd2$  for data bus causes a short circuit (SC) between voltage  $V2$  and  $Gnd2$ , ultimately decreasing output voltage. In order to overcome this problem, we propose interface module that provides galvanic isolation between the tiles for inter tile data communication as shown in Fig 8.1(b). In order to achieve high reliability and fault tolerance, an ad hoc interconnection system called cross bus has been proposed by authors of [28]. Fig.8.2(a) shows basic module for cross bus

architecture and Fig.8.2(b) shows a network of basic modules connected with the bus; six tiles connected to form the cubic shaped small satellite. This is a fully redundant system since every node is always reachable unless it suffers four faults on different points which is quite rare even in space applications. In Fig8.2(a), the

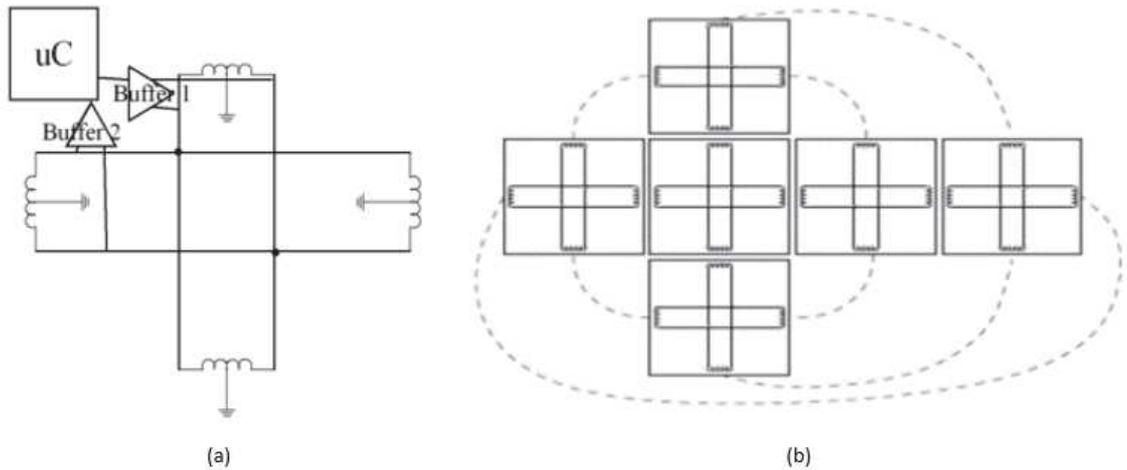


Figure 8.2. Cross bus module (a) Single (b) Network of six

controller sends data to be transmitted on the bus through differential buffer1 and received data from differential buffer2 is sent to the receiver of the controller for further processing. Interconnecting modules of Fig 8.2(a) together, results in a network structure of six tiles labeled Fig.2(b). In the next sections, we present the detailed structure of the basic module i.e. transmitter, receiver and coupling.

### 8.3 Design of Interface Module

The bus interface circuit is connected between communication channel and micro-processor on each tile. Fig.8.3 illustrates the design of bus interface structure. The circuit employs an open drain Metal Oxide Semiconductor Field Effect Transistor which is used as a switching element. The transformer is used as coupling element and the differential receiver is used as a receiver to receive the pulsed signal.

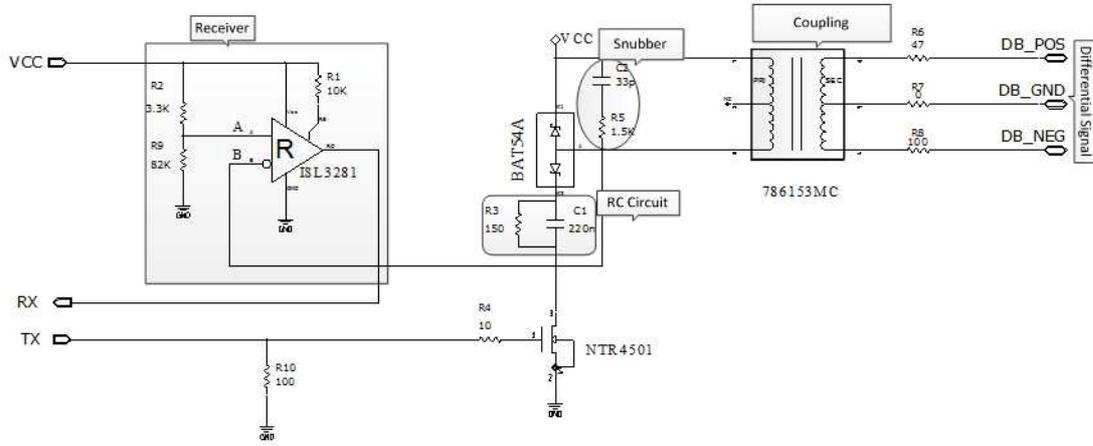


Figure 8.3. Design of Bus Interface circuit

### 8.3.1 Transmitter

The AraMiS employ IrDARZI coding scheme in communication bus therefore it is necessary to transmit pulsed signals onto the bus. The easiest way to generate pulsed waveform is to use a Metal Oxide Semiconductor Field Effect Transistor (MOSFET) and the Gate terminal of the transistor to control current flow. In MOSFET, a voltage on oxide insulated gate induces a conducting channel between drain and source terminals. Switching frequency depends upon the frequency of input pulses. Input pulse of the controller is applied to the Gate of transistor that creates a current path from supply voltage to ground for the time the pulse is high. During the pulse, there is a change of voltage across primary side of the transformer that induces a voltage to its secondary side by the principle of mutual induction. The input signal of each module is pulse waveform (from tile's microcontroller), with maximum 1Mbps baud rate. An RC parallel circuit is placed in the current path from supply to ground as shown in Fig.8.3. Two main advantages are achieved by placing the RC circuit.

- At high frequency i.e. at switching frequency, the capacitor behaves like a short circuit according to (8.1)

$$X_c = \frac{1}{2\pi fC} \quad (8.1)$$

Where  $X_c$  =reactance in ohms

$f$  = frequency in Hz

$C$  =Capacitance in farads

i.e. at high frequency, the reactance is negligible and it can be viewed as a short circuit across capacitor terminals. Effect of resistor is eliminated causing high amplitude current flow through primary side of transformer to ground path of Fig.8.3.

- In the event of faults i.e. when there is either continuous '1' across MOSFET gate terminal, there is zero frequency or steady DC. In this case capacitor becomes an open circuit and current flow through resistor only. We get a limited current with less amplitude and the devices get protected.

### 8.3.2 Receiver

The design of the receiver is based on a RS-485 receiver. Fig.8.3 shows receiver section with non-inverting input labeled as line  $A$  and the inverting input labeled as line  $B$ . Two wires carry the signal voltage and reference voltage. The receiver detects the difference between the two. Since most of the coupled noise is common to both wires, it cancels out. The receiver must see certain voltage difference between  $A$  and  $B$ . If  $A$  is greater than  $B$ , the receiver's output is logic high. If  $B$  is greater than  $A$ , the output is logic low. Fig.8.4 show plot of differential voltage at  $B$  compared with a fix voltage at  $A$  in the receiver section. As soon as voltage at  $A$  becomes

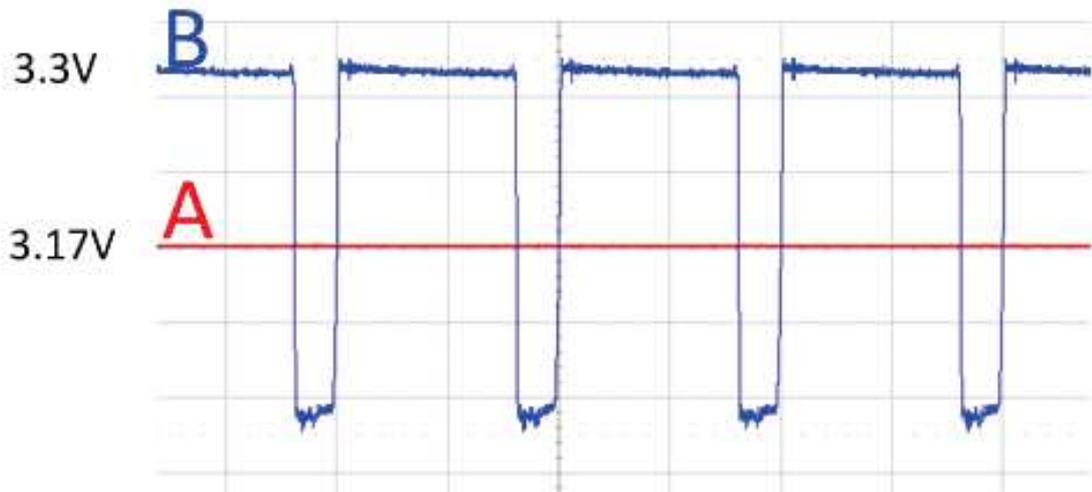


Figure 8.4. Receiver inputs

higher than that on  $B$ , the receiver outputs a high pulse. Commercial low power receiver ISL3281E [55] is being used in proposed design. Receiver inputs feature a “Full Fail-Safe” design, which ensures logic high receive output if receiver inputs are floating, shorted, or terminated but undriven. Another main reason for choosing an RS485 receiver here is that it can withstand a common mode voltage far beyond the power rails, and this happens regularly with the circuit’s configuration.

### 8.3.3 Coupling

For coupling purpose, pulse transformer is used because it is easy to connect with the signal controller, with RZI coding. The only limit of its use is the product of peak pulse voltage and the duration of the pulse ( $V.t$  product). Pulse transformers by definition have a duty cycle of less than 50%. The energy stored in the coil during the pulse must be damped out before the next coming pulse. In order to damp out the energy of the pulse, a commercial Schottky barrier diode, BAT54A [56] has been used as a Flyback diode across primary terminals of the transformer as shown in the Fig 8.3. Schottky barrier diode is a semiconductor diode with a low forward voltage drop (0.15V-0.45V) and a very fast switching action. The second schottky diode in the Metal Oxide Semiconductor (MOS) branch is used to avoid ringing of the transformer/bus with drain to source capacitance,  $C_{ds}$ ; of the MOSFET. The lower voltage drop of the diode can also provide higher switching speed and better system efficiency. The proposed design uses commercial 786153MC pulse transformer [57].

### 8.3.4 Snubber to Suppress Ringing

During the tests, the first non-ideal phenomenon is the ringing at the receiver part. Actually, this is due to the fact that Schottky diode during reverse bias behaves just like a capacitor and energy stored in the inductor i.e., transformer winding starts ringing with the capacitance of reverse biased schottky diode introducing oscillation therefore overshoot is encountered in the voltage waveform. A typical solution is to add an RC parallel snubber circuit to dump the ringing. The basic function of snubber circuit is to absorb energy from the reactances in the circuit [58]. An RC snubber is proposed in our design to overcome the ringing problem as shown in Fig.8.3. The RC snubber damps the ringing and if snubber resistance is equal to the characteristics impedance of the resonant circuit  $\sqrt{\frac{L}{C}}$  then resonant circuit will be critically damped and have no overshoot. The values of these R and C can be optimized experimentally.

The actual fabricated module is shown in Fig.8.5 . This module is placed in a plug and play approach in each AraMis tile.

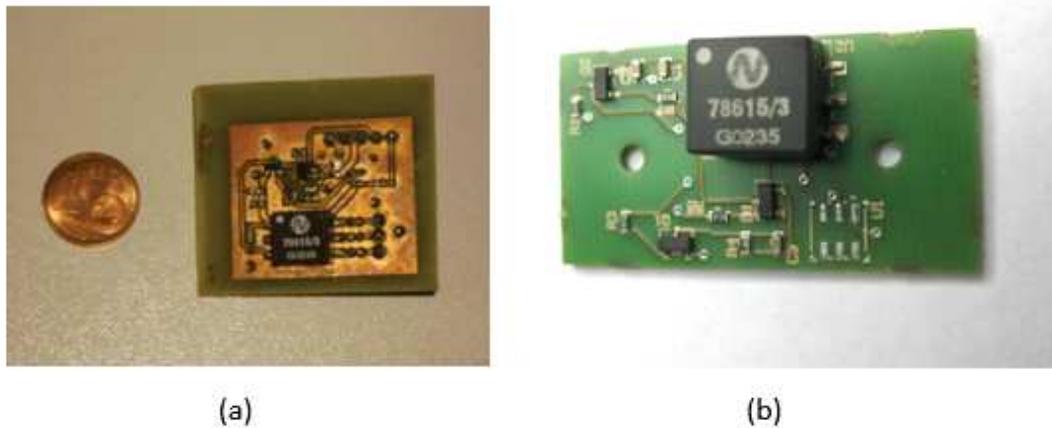


Figure 8.5. Implemented Bus Interface Module (a) Test circuit (b) Final Module

## 8.4 Fault Analysis

As space environment is prone to faults, several types of faults can be encountered during the time when the satellite is in orbit [59]. Detailed fault analysis with all the possible types of faults is discussed in this section. The bus can tolerate the following types of faults

- Short circuit between a conductor of the bus and the ground (see Fig.8.6(a)), the line that has been shorted with ground will have no voltage but the other one will continue to operate causing sufficient differential voltage but with reduced noise margin for the receivers to receive the signal correctly.
- Short circuit between a conductor and a bus power supply (see Fig.8.6(b)) has the same effect as in the above case, but constant supply voltage on the line may damage the winding of the transformer because of the fact that the specific transformer winding may not be able to withstand supply voltage, the solution is to add a series resistance that prevents saturation of the core of the transformer and the bus continues to operate normally with the overhead of noise margins reduction.
- Open circuit of one of the conductors (see Fig.8.6(c)), the system continues to operate because if one of the lines is down, the other one is operating normally. We reduce differential voltage and noise margin but the bus operates continuously.
- Short-circuit of the secondary of a transformer, (see Fig.8.6(d)), the bus continues to operate normally and only the faulty unit will be isolated from the

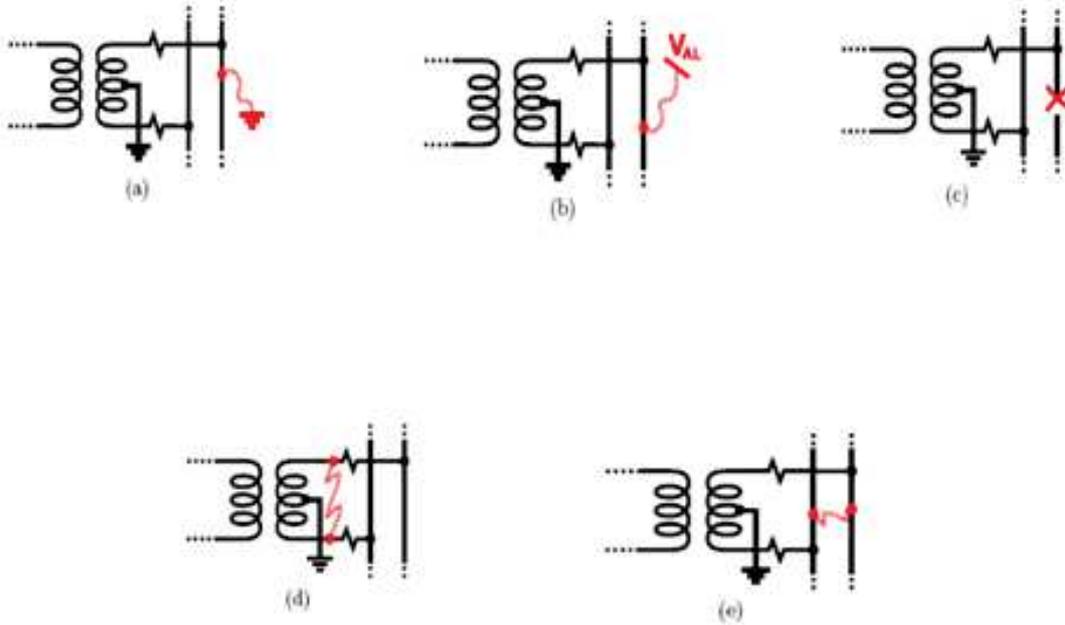


Figure 8.6. Possible Types of Faults

bus i.e. it will not receive any signal.

- Short circuit of both conductors (see Fig.8.6(e)); in this case there is no voltage difference between the conductors of the bus but we still obtain differential voltage at the receivers because of the different values of the termination resistances. The expense is significant reduction of the noise margin.

According to this analysis, even in case of possible faults, the modules transmit and receive the data correctly. Therefore the proposed structure can provide fault tolerance at the expense of losing noise margin. In order to demonstrate the fault analysis, we connected six modules across the bus and then measured mean differential voltages on each receiver in standard situation and then with possible fault condition. Fig.8.7 shows the block diagram of interconnected modules. In the block diagram, module 2 is used as transmitter and all others are receivers. Failure condition always happens at module 4.

Table 8.1 summarizes the mean voltage on the inverting terminal of each receiver i.e. at line  $B$  of each receiver in normal operation and with all possible faults. Under circumstances when there is no activity on the bus, the mean value on line

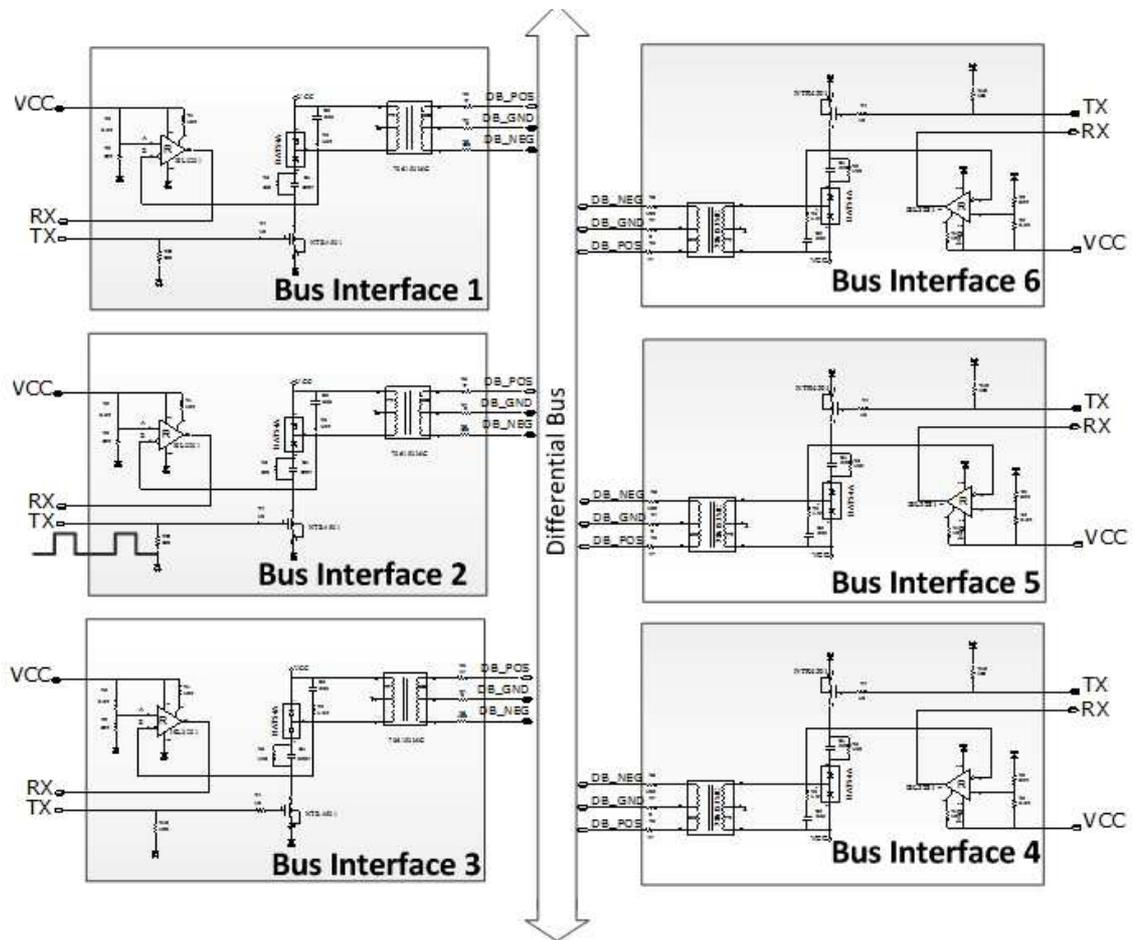


Figure 8.7. Block Diagram of connecting six modules

$B$  remains at constant 3.3V i.e. at supply voltage. Under normal activity, the higher differential voltage causes more drop of supply voltage, ultimately reducing the mean voltage.

The data of Table 8.1 is obtained by using the proposed bus interface circuit with all types of faults listed in Fig. 8.6. The resulted waveforms of interconnecting modules of interface circuit are shown in Fig. 8.8 (a) through (e) in the same order as they had been discussed in Fig. 8.6. This differential voltage is compared with a fixed voltage through a voltage divider (on inverting terminal  $A$ ) as can be shown in basic interface circuit in Fig. 8.3. Depending upon the difference between  $A$  and  $B$ , the receiver generates the output voltage pulses accordingly. Input waveform is shown in yellow trace. There is a high differential voltage on the bus in case of faults (a) and (b). Shortening of both the conductors together causes worst decrease in

Failure Type	No fault	Fault (a)	Fault (b)	Fault (c)	Fault (d)	Fault (e)
Module No.						
1	2.7V	2.85V	2.80V	2.85V	3.1V	3.25V
2	2.5V	2.5V	2.5V	2.5V	2.5V	2.5V
3	2.7V	2.85V	2.85V	2.8V	2.7V	3.15
4	2.7V	2.9V	2.85V	2.8V	0V	3.2V
5	2.7V	2.9V	2.85V	2.8V	2.7V	3.0V
6	2.7V	2.9V	2.85V	2.8V	2.7V	3.0V

Table 8.1. Fault Analysis

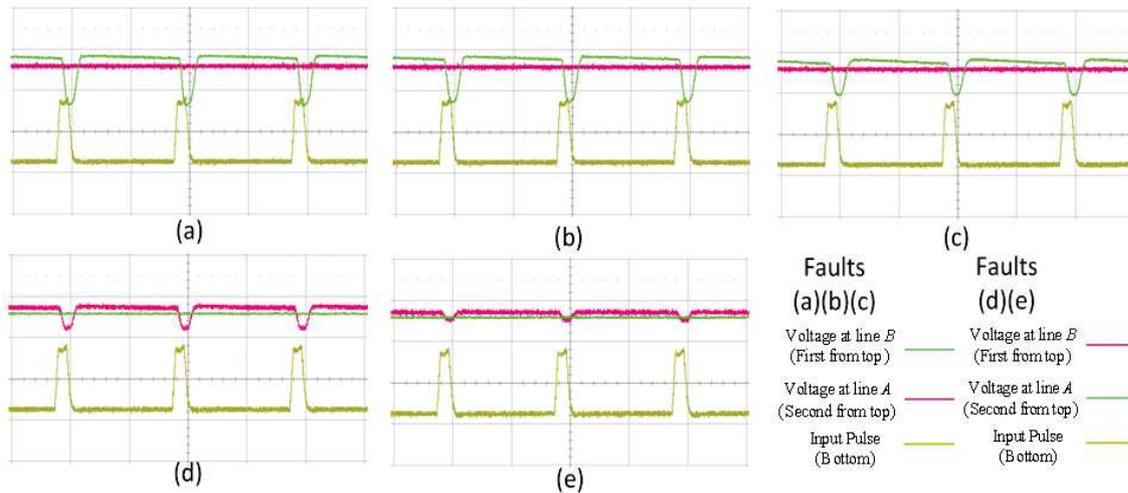


Figure 8.8. Block Diagram of connecting six modules

the differential voltage but the mean differential voltage will increase in this case. It is apparent from column (e) of Table 8.1. Since Table 8.1 lists the mean voltage at the inverting terminal of each receiver, it is apparent from Table results and Fig. 8.8 that this input stays at supply voltage i.e. 3.3V and as soon as there is differential signal (referred to negative terminal of bus DB\_NEG), the level drops below 3.3V. In case of no fault, the high differential signal causes more drop and ultimately we get relatively low mean voltage on the inverting terminal as it can be seen from Table 8.1. Mean value across inverting terminal of the differential receiver is about 2.7V in case of no faults on the bus. Every fault situation causes reduction in mean

voltage according to how severe it is. In case of fault (e), which is the most severe fault, there is a very minor differential signal and a very minor drop in the mean voltage as is apparent from column (e). Since in this demonstration, fault condition happen at module 4 only, it is quite clear from the data in table that we don't get any voltage at the receiver of module 4 when fault (d) happens on it. Since module 2 is used as transmitter in our demonstration, its receiver receives the best differential voltage in any case as it has not to do anything with the faults.

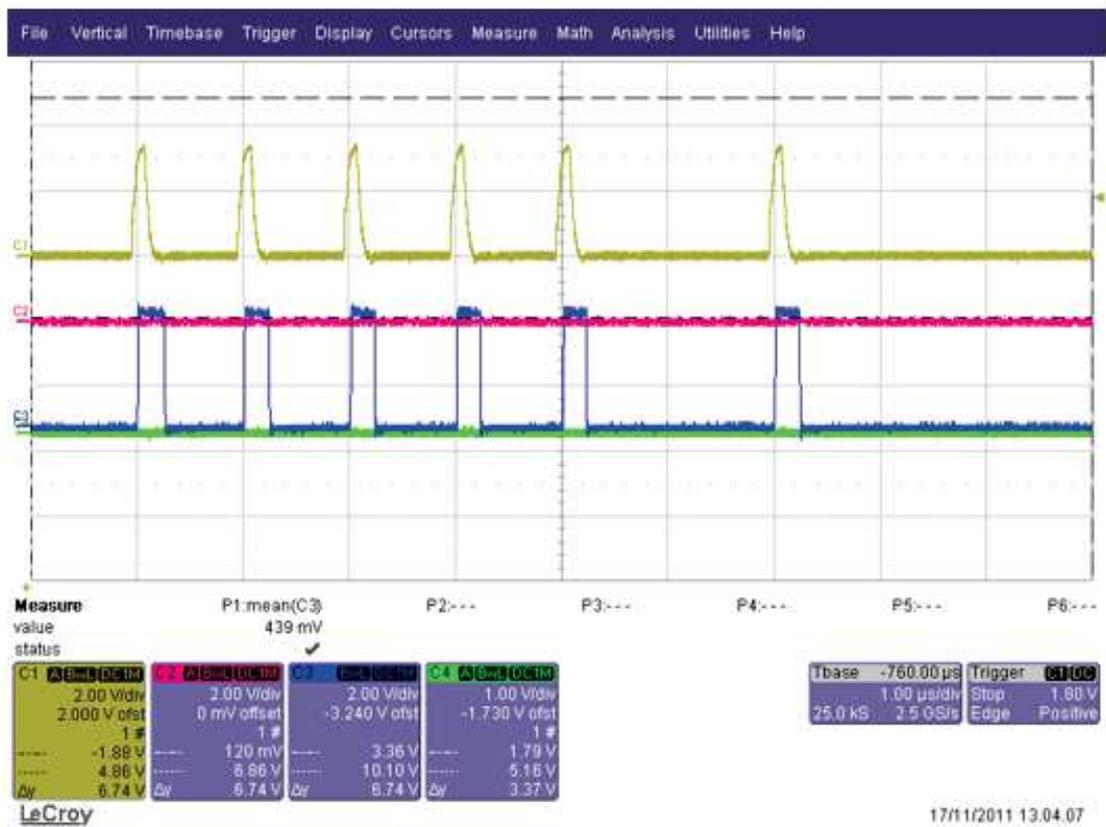


Figure 8.9. Transmit and Receive of Random Bit Sequence

Fig.8.9 shows the performance analysis interface circuit. A random bit sequence generated by MSP430 UART is applied to the transmit pin TX of module 2. The yellow trace show the input transmit stream and blue trace show the voltage level received on the receiver pin RX of the module 4. The transmit and receive behaviour has been tested with random sequence of bits transmitted on the bus with and without fault conditions. The results show no delay in the reception of bit sequence and also acceptable bit error rate.

## Conclusion

In this work we have shown the use of intelligent modular tiles and COTS components to design a fault tolerant system which is the basic aim of AraMiS project i.e. to achieve fault tolerance from COTS components and also the modular design. The data communication interface module has been tested with MSP430 microprocessor at different transmission baud rates and it has shown maximum transmission and reception error within the protocol limits. In this work, we only tested the module using IrDA RZI encoding scheme. In future, we will test this circuit with all possible wired transmission encoding schemes (like 8B10B etc.) and also by modifying the proposed circuit to have only radiation hardened LED and use wireless solution instead of employing differential signaling and pulse transformers.

# Chapter 9

## Module Design: Optical Module

### 9.1 Introduction

A typical spacecraft involves many different components that vary in bandwidth demand. Sensors requiring a very low data rate may reside on simple two or three wire interface such as [I2C](#), [SPI](#) etc. Complex sensors requiring high data rate and bandwidth demands may reside on optical interface. Design technique for optical wireless transceiver is discussed for modular Nano-satellites using low cost [COTS](#) components. The proposed transceiver structure uses infrared light emitting diode to transmit pulses in the near infrared spectrum. The receiver circuit consists of photodiode, pre-amplifier, post-amplifier and a comparator stage. Since noise is a major problem for receiver, a very low noise pre-amplifier and post amplifier stages have been selected. Since the communication system is inside the satellite, there is possibility of sunlight to enter from small holes and blind the receiver. This problem is solved by attenuation circuit. Displacement damage effect on selected [LEDs](#) and photodiodes has been calculated and the resulting drop in efficiency has been verified using proposed transceiver. This optical wireless module is fully scalable depending on the needs of the system and can coexist with more traditional wired communication channels as required. An innovative design of placing glass fiber for reliable communication across the tiles has been discussed. The optical light has been guided in certain directions using double surface mirrors. Theoretical and measurement results using a single fiber in light of sight have been calculated and the results are in close comparison with each other. Free space link has also been modeled and the received current at certain positions in the satellite due to a fixed transmitter has been mapped [[12](#),[60](#)]. The only shortcoming of the free space link is that when the payload is present, it becomes very difficult to maintain line of sight between the nodes and complex optics to reflect the light for the transmitters and receivers becomes a major problem. Every tile hosts two tiles processors (MSP430

controllers) which are responsible for communication across different nodes. At the end, this paper has shown some possible schemes for data communication architecture for small satellite using optical infrared light in parallel with the typical wire based solution.

This chapter is presented in [12, 61] and discusses design technique for data communication transceiver for modular tile based architecture and its use in the implementation of optical communication inside AraMiS. The proposed transceiver modules will be placed in a plug and play approach in each tile.

In [13, 14], the authors describe analysis of using wireless communication bus and discuss the use of this standard to exchange data between the sub-systems of a satellite. The non-space grade commercial optical transceivers utilize infrared LEDs, which may suffer from displacement damage depending on the device's construction [18, 19]. In addition, the IrDA transceivers are packaged in plastic packaging with the emitter/receiver pair. This plastic may also become less transmissive with increasing radiation dose. Keeping in view the above mentioned problems in commercial transceivers that have not been designed for space radiation environment, novel discrete transceiver for Infrared data communication (IrDA) inside small satellites has been designed using discrete LEDs and photo-diodes. Using discrete solution gives more freedom in wavelength selection and best components keeping in view space radiation problem.

The benefit of IrDA transceiver over a wired communication transceiver is both in the intrinsic galvanic isolation and no physical wires. No physical wiring also means reduction in connectors and cables inside small satellites therefore reducing weight and integration complexity.

The benefit of the proposed transceiver over other wireless transceivers (such as Radio Frequency (RF), Bluetooth etc.) is that optical wireless channel is fully immune to radio interference i.e. no electric noise can be radiated. Also the infrared spectrum represents an immense, unregulated bandwidth. Moreover Infrared technique is quite low power as compared to RF. Fig.9.1 shows the block diagram of the implemented communication system. The communication system consists of both electronic and optical components. The incoming digital stream from the tile processor is encoded into Return to Zero Inverted (RZI) pattern, where a zero in the transmitted stream is represented by a pulse and a one is represented by no pulse. This RZI pattern is transmitted to the transmission medium in the form of optical light pulses using infrared light emitting diodes (IRLEDs) of different wavelengths with the use of LED driver circuitry. Both free space optical and glass optical transmission media have been implemented for inter and intra tile data communication. The free space medium requires Line of Sight (LoS) between the transmitters and receivers and hence becomes difficult to implement at times. The receiver block consists of photodiode, transimpedance amplifier and a comparator stage before the data stream is decoded and read by the tile processor for further protocol

management purposes. Fig.9.1 shows the blocks containing optical and electronic components in the system. All the encoding, decoding and filtering stages consist of electronic components while the optical components consist of infrared light emitting diodes, photodiode and guided channel if any. The chapter is developed according to following sequence. Section 2 describes functional description and implementation of the transceiver, section 3 describes radiation effects on LEDs and photodiodes and section 4 discusses the optical component selection and section 5 describes the free space and guide light communication using the proposed receivers.

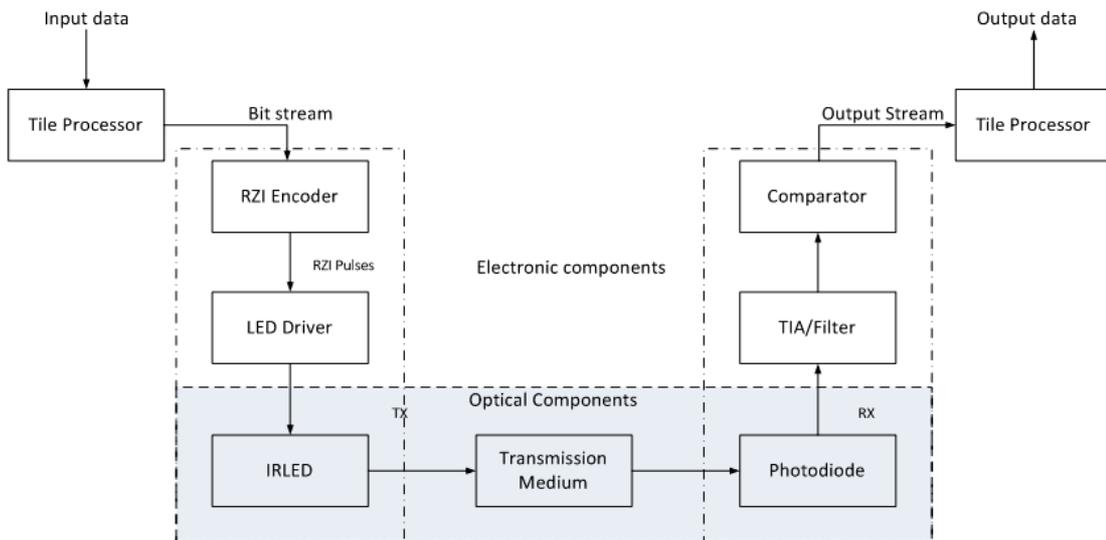


Figure 9.1. Communication system overview

## 9.2 Functional Description

Low speed communication link (0.5~1Mbps) is needed for inter- as well as intra-tile data communication for AraMiS architecture. Typically, the free space optical systems operate at either around 850nm or 1550nm wavelength range. Light emitting diodes (LED) or laser diodes are commonly used for transmitter. Laser diode has much narrow emission beam but it is much more power hungry than LEDs; therefore LED is preferred inside small satellite architecture where available power consumption is the main constraint.

A Receiver (RX) consists of an optical front-end with a pre-amplifier. The optical front-end includes a PIN photo-diode to collect incoming optical radiation, as well as ambient light rejection circuit to reject unwanted radiation. The photo-diode

converts optical power into photo-current, which is then followed by amplification stages, filtering and comparator stages etc. Intensity modulation is the only practicable transmission scheme in free space optical communications [62]. Since free space optical is an unguided propagation, it faces relatively high free-space loss, as part of the radiated optical power may be lost or not captured by the receiver. Furthermore, due to the reflections from walls and other objects, signals travel via different paths and arrive at the receiver at different time instances and with different angles of incidence. Ambient light (sun light that may inter through holes in the satellite) is the dominant noise source in free space optical aerospace systems, which primarily enhances the shot-noise component at the receiver. Another dominant noise source is the amplifier noise which may be reduced by selecting a very low noise amplifier. The chosen encoding scheme is RZI but any other user defined encoding scheme such as Non Return to Zero (NRZ), Return to Zero (RZ), IrDA 4PPM etc. for data transmission can be also be used with the proposed circuit using MSP430 tile processors in Universal Asynchronous Receive Transmit (UART) IrDA mode. We have chosen MSP4305438A micro-controller for controlling the communication because this device has up to four universal serial communication Interfaces with IrDA capability. This device is going to be evaluated under radiation environment but we are quite confident of its functionality under radiation as many CubeSat missions have already used MSP430 processors on-board.

### 9.2.1 Transmitter Design

The design of transmitter is fairly simple. An elementary design consist of a Metal Oxide Semiconductor Field Effect Transistor (MOSFET) whose drain is connected to the cathode terminal of LED while gate is connected to the transmit signal. The pulsed signal applied to the gate terminal is an RZI encoded signal by the tile processor. The anode terminal is connected either directly or through a current limiting resistor to the power supply voltage. Fig.9.2 shows block diagram of transmitter circuit.

The radiated optical power,  $P_o$ , is directly proportional to the amount of current flowing through LED ( $I_{LED}$ ) given by (9.1).

$$P_o = \eta \cdot V I_{LED} \quad (9.1)$$

Where  $\eta$  is the optical efficiency of the LED measured in  $\frac{mW}{Amp}Sr$ . The amount of current flowing through LED is controlled by appropriate resistance  $R_D$ . The appropriate resistance is set by (9.2)

$$R_D = \frac{V_{CC} - V_F}{I_F} \quad (9.2)$$

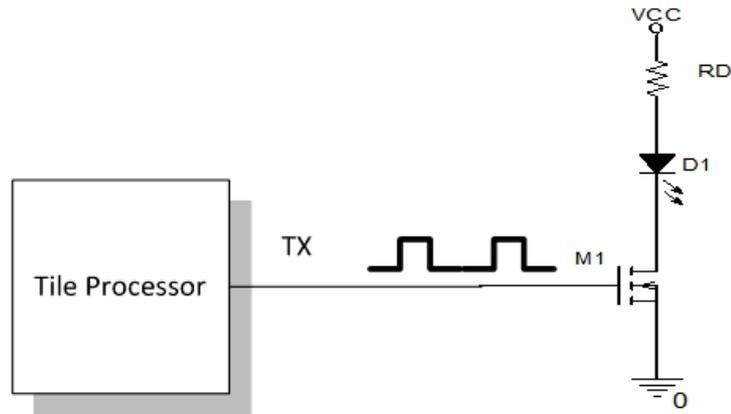


Figure 9.2. Transmitter block diagram

Where  $V_F$  is the forward voltage of LED and  $I_F$  is the resulting current flowing through LED. Typical values of  $V_F$  are in the range of 1.3-2.0V. Equation (9.2) shows that the optical source generates radiation intensity proportional to the variations of its bias current and, at the detector; the optical signal is directly translated into a photocurrent.

Many commercial LEDs and photodiodes have been evaluated for our design requirement [63–65]. The transmitter has been designed for minimum radiant intensity of 25mW/Sr up to 1000mW/Sr which translates to the optical power of 2.5uW/cm<sup>2</sup> to 100uW/cm<sup>2</sup> for 1m distance between transmitter and receiver.

### 9.2.2 Receiver Design

The structure for the receiver consists of PIN photodiode, a first stage low noise Trans-Impedance Amplifier (TIA), a second stage voltage amplifier and a comparator stage. Ambient light rejection circuit is used to filter out ambient dc light and only useful signal pass through. The comparator is used to compare useful signal with the threshold voltage. This threshold level is generated by the level detect block, which measures the strength of the received signal and adjusts the threshold in order to maximize the Bit Error Rate (BER). Fig.9.3 shows the functional block diagram of the implemented receiver.

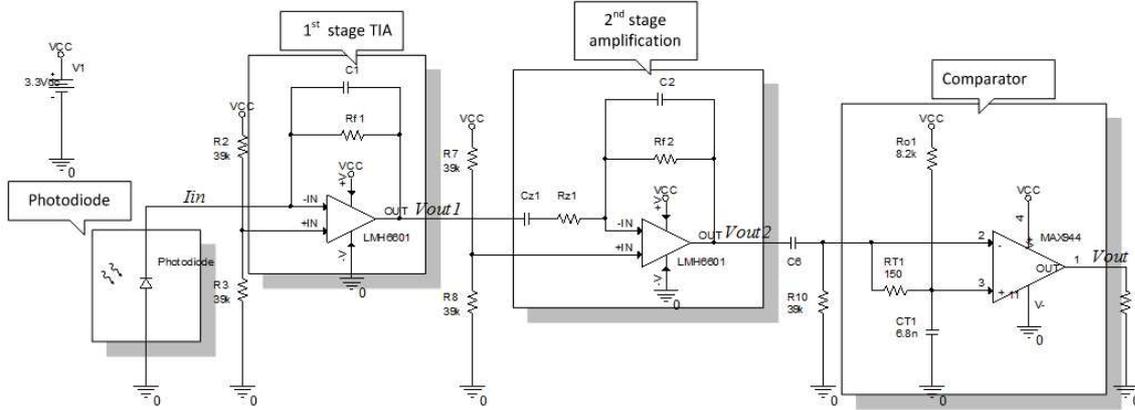


Figure 9.3. Receiver block diagram

### 1<sup>st</sup> Stage Transimpedance Preamplifier

Photodiode current is monitored using either direct voltage monitoring or current to voltage conversion. Direct voltage monitoring employs simple circuitry i.e. diode current is supplied directly to the resistive load but results in severe bandwidth limitations, non-linearity and high dc offset. The current to voltage converter solves these problems. The transimpedance amplifier or current to voltage amplifier is used to convert photocurrents generated by photodiode into useful voltage signals. Fig.9.3 shows the 1<sup>st</sup> stage TIA block. The transfer function for the 1<sup>st</sup> stage is given by

$$H_1(s) = -\frac{V_{out1}}{I_{in}} = -\frac{R_{f1}}{1 + R_{f1}C_1s} \quad (9.3)$$

Equation (9.3) suggests that the frequency response of 1<sup>st</sup> stage is due to the feedback network.

The transmitter radiant intensity requirements inside a simple cube structure require the receiver to handle the signal currents that vary from as small as 300nA up to a fraction of mV; a dynamic range of almost 50dB. This has been achieved by setting the 1<sup>st</sup> stage gain relatively low enough to handle the high dynamic range. The gain is set by using a relatively low value feedback resistor.

Several operational amplifiers were considered to be used as potential candidate for the receiver stages. Texas Instruments LMH6601 was chosen for design because of its low noise, ultra-low input bias current (a few femto amps), high speed, low voltage and large bandwidth features. Moreover this is operated in a single supply operation which was also one of the main parameters in selecting an operational

amplifier.

### ***2<sup>nd</sup> Stage Inverting Post Amplifier***

The 2<sup>nd</sup> stage amplifier has double function i.e. ambient light rejection and further amplification of signal coming from the 1<sup>st</sup> stage. The ambient light can be seen as a dc offset from the photodiode output. In space radiation environment, the optical communication link requires to operate under a variety of demanding ambient light conditions ranging from sun light coming from small holes of mechanical screws and payload. There are a number of techniques available in literature to filter out the ambient noise [66]. High pass filtering scheme has been used to filter out the photocurrents produced by the ambient light. The effect of ambient light on the receiver is very small as the photodiode has operating band far from peak power emission of sun. The amplification ratio is set by setting  $\frac{R_{f2}}{R_{z1}}$  ratio. The transfer function of amplification and filtering stage is given by (9.4)

$$H_2(s) = -\frac{V_{out2}}{V_{out1}} = -\left(\frac{1}{1 + R_{f2}C_2s}\right)\left(\frac{R_{f2}C_{z1}s}{1 + R_{z1}C_{z1}s}\right) \quad (9.4)$$

The rejection frequency is set by setting properly the time constants of bandpass filter.

Fig.9.4 shows the frequency response of 1<sup>st</sup> and 2<sup>nd</sup> stages. The 2<sup>nd</sup> stage has band pass response with the first pole at 300 Hz. The 2<sup>nd</sup> stage voltage gain in the Fig.9.4 has been designed to be 26dB more than the 1<sup>st</sup> stage gain. The theoretical expression for calculating the -3dB bandwidth of both stages is given by (9.5).

$$f_{-3dB} = \sqrt{\frac{GBWP}{2\pi R_{fn}C_{IN}}} \quad (9.5)$$

Where  $GBWP$  is the gain bandwidth product and  $C_{IN}$  is the input capacitance at the inverting input of the amplifier.  $C_{IN}$  depends on diode capacitance and amplifier input capacitance.

Fig.9.4 shows the -3dB bandwidth of about 10MHz. The signal of interest is 1MHz with 20% duty cycle.

### ***Threshold Comparator***

The high dynamic range of receiver due to unpredictable power of received pulses sets a trade-off as there is no fixed threshold voltage to compare the incoming pulses. A dynamic threshold comparator has been designed to generate the digital output stream. The dynamic determination of the switching threshold is necessary to allow the correct functioning of the system independently of the power of the received signal which can vary greatly depending on the distance and the reflections.

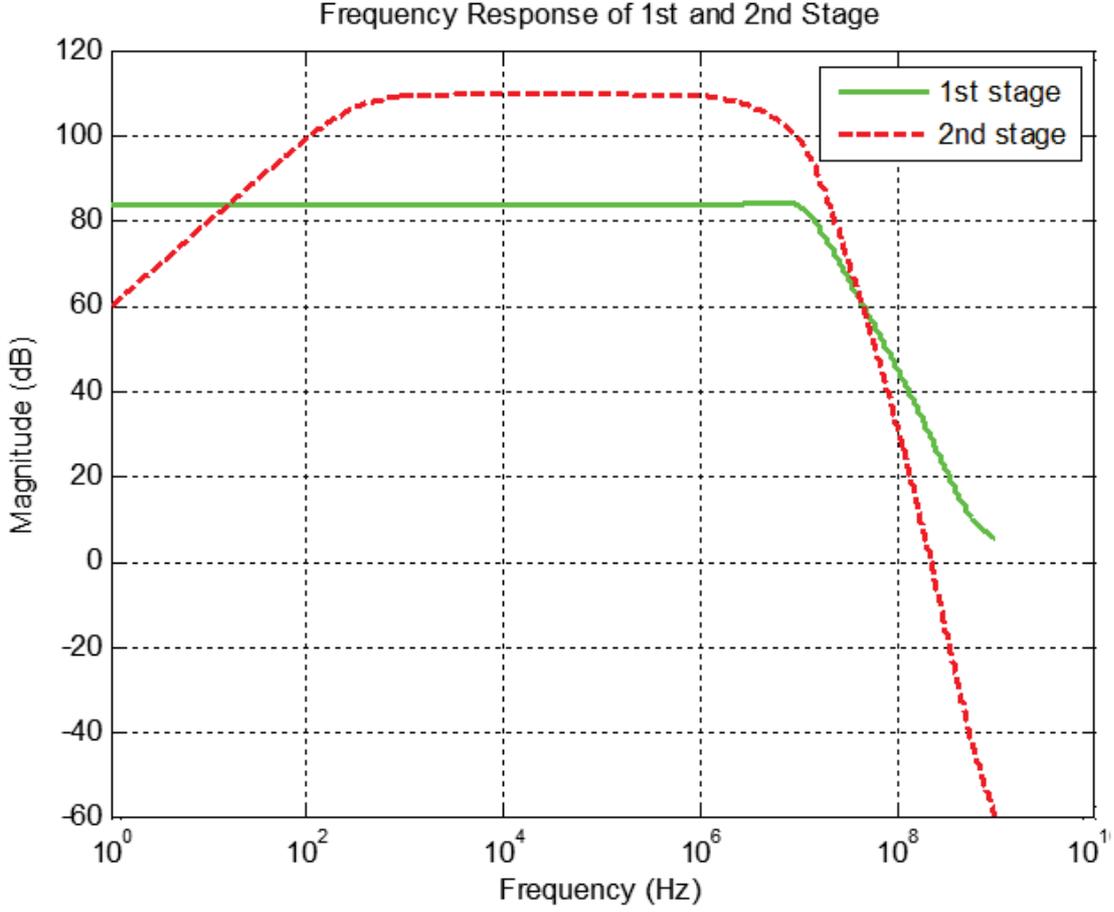


Figure 9.4. Frequency Response of Receiver Stages

The comparator section of the Fig.9.3 was designed that performs the comparison between the signal  $V_{out2}$ , coming from the post-amplifier, and the average value of the signal itself. In order to detect a pulse after a long period of inactivity or after the transmission of multiple zeros, the time constant  $\tau_{T1}$  has to be fast enough equal to bit time  $t_{bit}$ . Time constant is given by (9.6)

$$\tau_{T1} = (R_{T1} \parallel R_{o1}) \cdot C_{T1} \quad (9.6)$$

A good compromise for our requirements has been determined to be time constant equal to period of bit .

$$\tau_{T1} = t_{bit} = 1 \mu s$$

The offset voltage  $V_{off}$  should be at least 50mV higher than zero level so that comparator doesn't switch the output to logic 1 when there is long string of 0s and

bursts of noise may affect the comparator switching. The offset voltage  $V_{off}$ , is set by (9.7) and (9.8)

$$V_{off} = V_{CC} \frac{R_{T1}}{R_{01} + R_{T1}} \quad (9.7)$$

$$\frac{R_{01}}{R_{T1}} \leq \frac{V_{CC}}{V_{off}} - 1 \quad (9.8)$$

Any desired offset voltage is set by setting the respective resistor ratio and respecting the time constant of the circuit. The proposed optical transceiver for free space wireless link is fully scalable depending on the needs of the system and can coexist with more traditional wired communication links as required.

Figure 9.5 shows the simulation result of implemented receiver. The test has

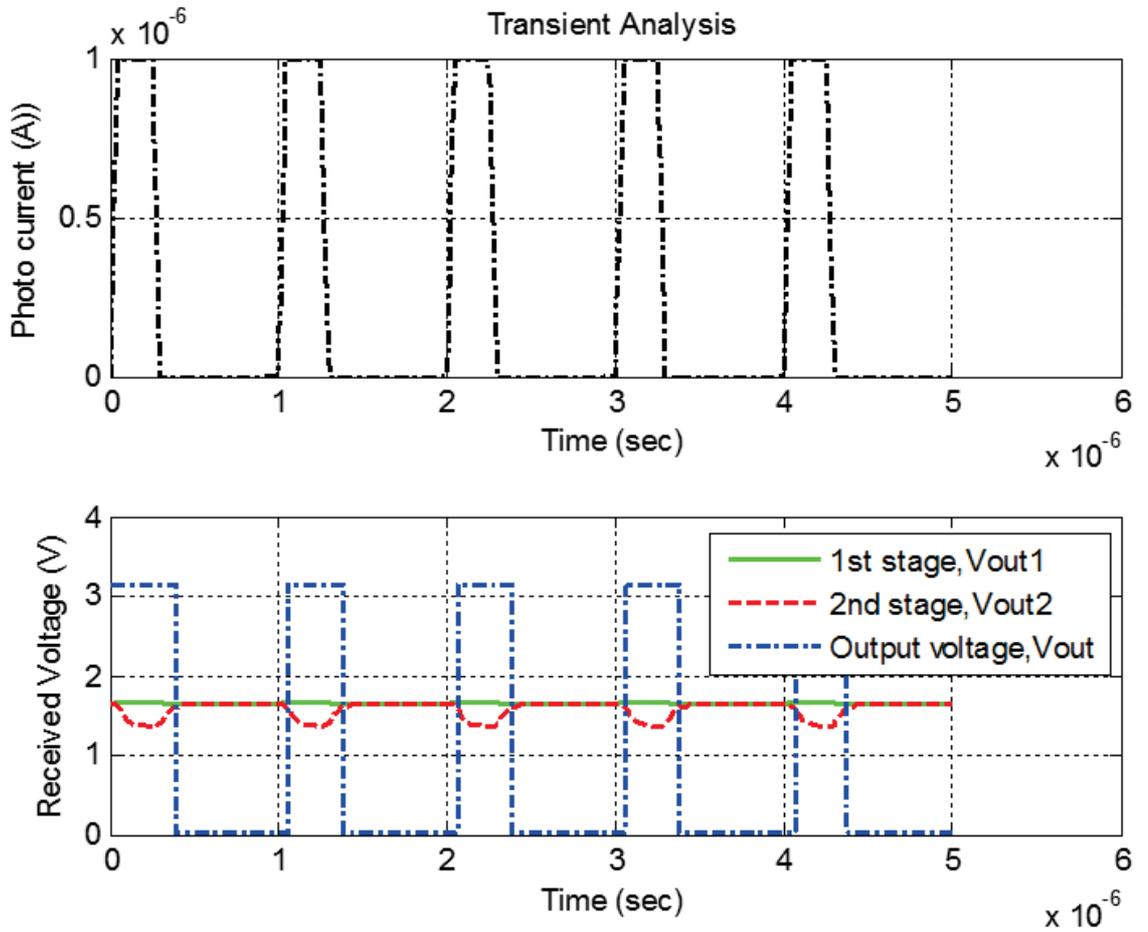


Figure 9.5. Photodiode current Vs Received Voltage

been computed at 1uA received current. The corresponding 1<sup>st</sup> and 2<sup>nd</sup> stage amplification and comparator output results are listed in the Fig.9.5. It is clear that there is a very small delay caused by the receiver circuit.

Fig.9.5 shows for a photocurrent of 1uA, voltage level  $V_{out1}$  after 1<sup>st</sup> amplification stage,  $V_{out2}$  after 2<sup>nd</sup> inverting amplification and filtering stage and  $V_{out}$ , the output of threshold comparator stage available for signal decoding and protocol management purposes. The circuit worked perfectly well for the whole 50dB dynamic range of the incoming signal in a point to point link achieving a BER of less than  $10^{-12}$  in laboratory conditions.

### 9.3 Radiation testing of selected optical components

The mission has been designed to remain in low earth orbit (LEO) at an altitude of 800km for five years lifetime. The Total Ionization Dose (TID) of 10 krad has been estimated taking into consideration the thickness of mechanical structure and different other parameters for each tile [59].

Displacement damage is the most prominent radiation effect on optical devices and the only way to test device tolerance to it is to evaluate device behaviour under a radiation source. Displacement damages in space are generated by protons, in particular by solar protons. Displacement damage testing is performed by simply characterizing the electrical performances of the device under test, exposing it to an irradiation source, without bias, to fixed particle fluence and characterizing it after irradiation to determine the parameter degradation. The radiation source used is generally a mono-energetic proton beam, and the part is irradiated to fluence greater than the mission equivalent fluence. The electrostatic particle accelerator was used for the measurements. Given the accelerator particle fluence and the desired no. of particles hitting the device, exposure time could be calculated. After an exposition session, the device was removed from the component tray and measured using a test equipment and then it was inserted again in the accelerator for further irradiation. The test equipment is measuring LED output power and photo-diode sensitivity; the results are plotted in Figures 9.6 and 9.7.

Fig.9.6 and 9.7 show that the rated output is only available at an angle of 0° and the useful range is available from -20° to 20°. Delta shows decrease in radiant intensity due to radiation dose. From the Fig 9.6 and 9.7 it can be noticed that the LEDs undergo relative attenuation of less than 15%, with TID of 100 krad. The photodiodes suffer similar attenuation patterns with variations, ranging from 2.5% for the PDB-C142 to 6.9% for PDI-C172SMF, for TID of 100 krad. In the case of the photodiodes PDB-C142F and PDI-C172SMF tests were carried out by irradiating

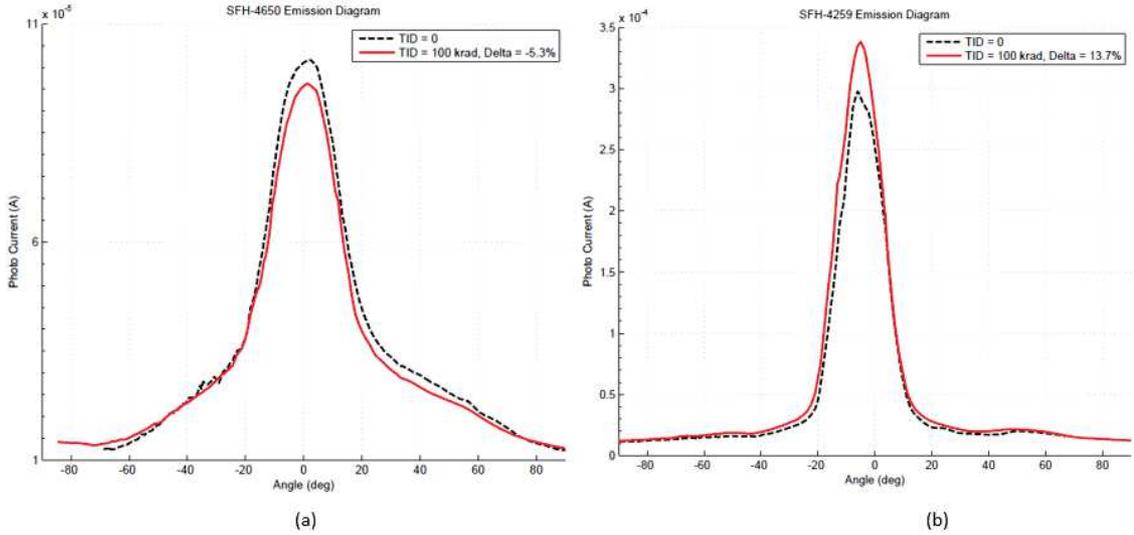


Figure 9.6. LED radiant intensity Vs angular displacement

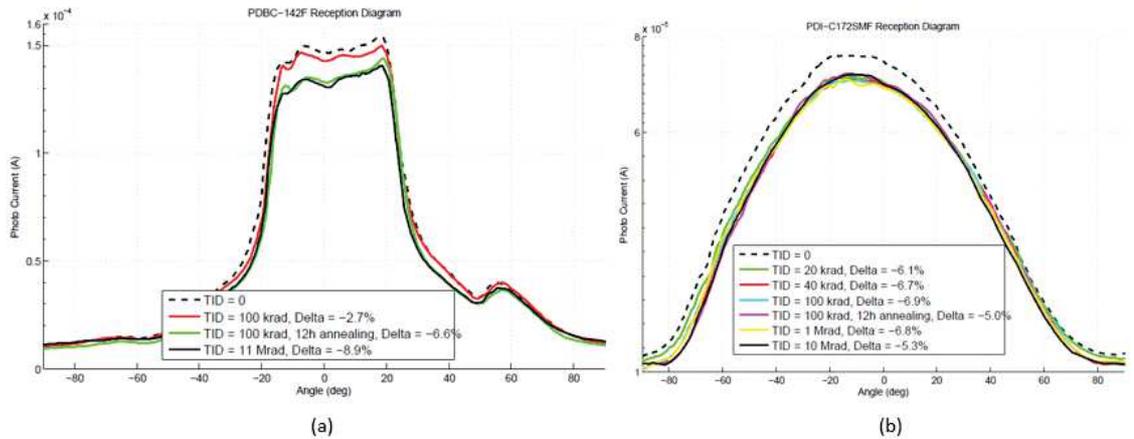


Figure 9.7. Photodiode sensitivity at various TID levels

with TID above 10 Mrad and the maximum variation remains within 10%. It's interesting to note in it as in the case PDI-C172SMF photodiode, an annealing of 12 hours brings back the attenuation characteristic from 6.9% to 5% with a recovery of lattice damage suffered.

In summarizing, the exposure of optical components to Total Ionizing Dose of 10 times the specific mission does not cause a significant attenuation on emission

and reception diagrams of the LED and photodiodes respectively. This, however, caused the displacement damage to the package with limited depth of penetration. For an evaluation of the effects of the displacement damage in the silicon, further tests in future will be needed with more high-energy particles or with a removal of the package of the devices making it possible to expose more the semiconductor radiation.

### IRLED and Photodiode

The main parameter in the selection of LED is the radiated emitter power in terms of light energy and the amount of current required to generate the desired light energy. The radiated optical power is directly proportional to the amount of current flowing through LED,  $I_{LED}$ , given by (9.1). The theoretical expressions for calculation of transmit and receive optical power in free space and guided light for intra tile communication has been discussed in this section.

The transmitted optical light from the LED is not evenly distributed but distributed in the angular range. Fig.9.8 shows a typical graph of relative radiant intensity i.e. emitted power versus angular displacement for commercial LED, TSHG8400 [63]. It shows how directional the emitted light is. The narrower the radiation pattern, the more optical energy is concentrated in particular direction.

The radiation pattern of typical LEDs depends upon the incident power  $P_o$  and radiant intensity  $I_e$ . The beam angle for [63] is given by (9.9).

$$Beam\ angle = \frac{P_o}{I_e} Sr \quad (9.9)$$

The radiation pattern of [11] for 50mW transmit power is 0.8Sr which corresponds to 30° radiant power intensity.

## 9.4 Free Space Communication Architecture

The discrete optical infra-red transmitters and receivers were tested for different free space link distances. The radiation intensity i.e. radiated power per unit solid angle, at a distance  $r$  from the source of optical power, is given by (9.10)

$$I_{received} = \frac{P_o}{r^2} * A_{Photo} * Responsivity \quad (9.10)$$

Fig.9.9 shows the received current at different link distances with variable input current. This plot is very helpful for selecting minimum radiated power which makes sure that the receivers receive sufficient optical power for level detection. Since the dimensions of single tile are 16.5x16.5 cm<sup>2</sup>, it is apparent from the graph that even

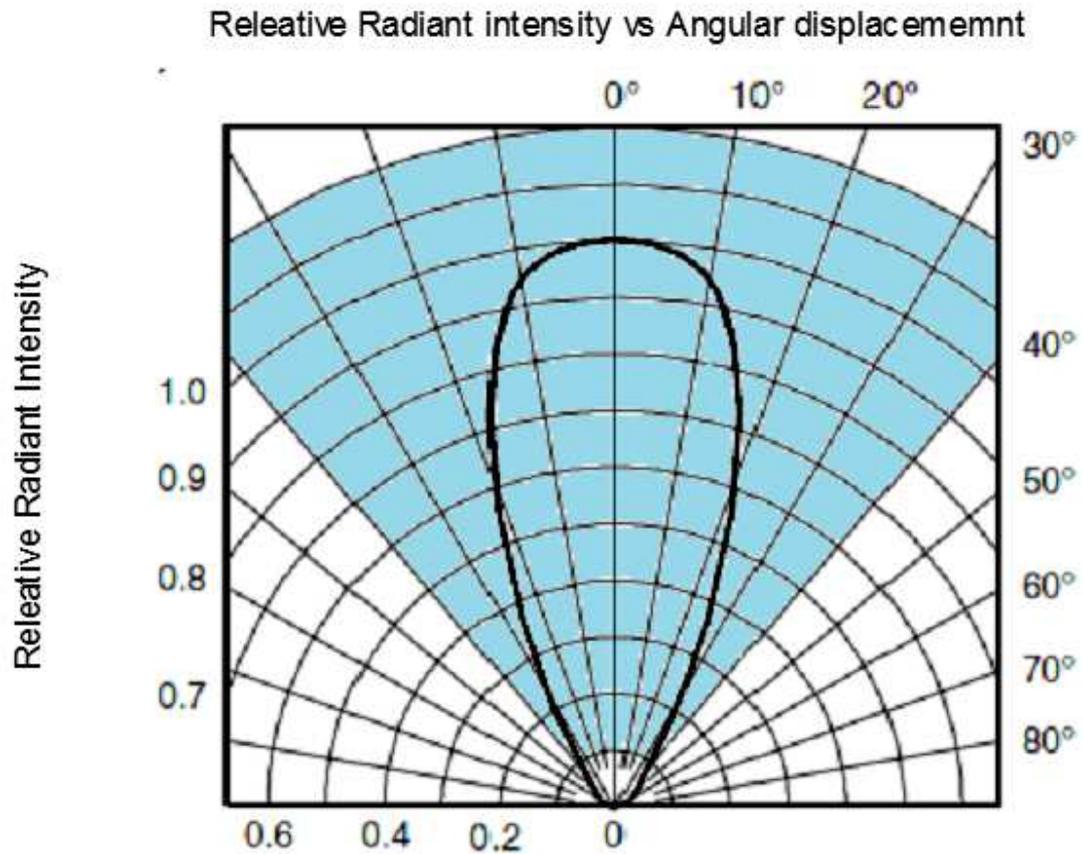


Figure 9.8. Relative Intensity vs Angular Displacement, courtesy Vishay

a very small forward current is enough for good reception of signal. In case of glass optical fiber based communication, power level for transmitter can be increased if the signal has to travel through multiple reflections.

Fig.9.10 shows a possible scheme of implementation of free space communication across the tiles.

Two transmitters,  $TX1$  and  $TX2$  are used for test purposes as shown in the Fig.9.10(a). The receivers are placed at different angular positions and the resulting received current is plotted in the Fig.9.10(b). Transmitter  $TX2$  is placed on the top position and the  $TX1$  is placed on the bottom position. The figures show certain possibilities of reception due to both transmitters either at top, centre and bottom. The results of these practical measurements are in close comparison with the theoretical results and shown in Fig 9.10(b) and also illustrate that the proposed transceiver is able to handle these received currents very efficiently.

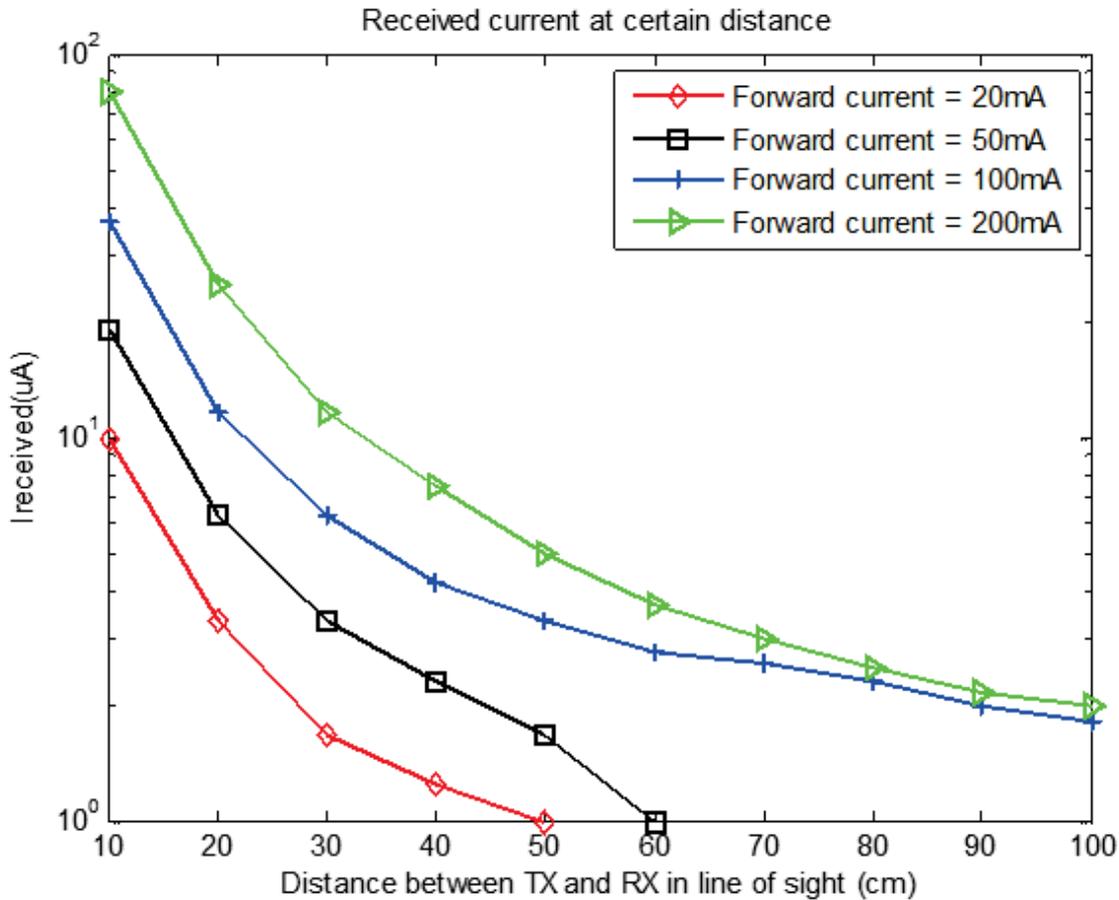


Figure 9.9. Received photocurrent for certain distances

## 9.5 Glass Fiber Based Communication Architecture

A number of configurations for reliable communication for inter- and intra-tile communications using the proposed transceiver were studied and the best proposed implementation structure is shown in the Fig 9.12.

For experimental purposes, a number of glass materials were evaluated for short range communication purposes inside the satellite. *Plexiglass* or *polymethyl methacrylate*, which is a strong, transparent polymer plastic glass showed the best results with respect to the transmission of infrared light in the laboratory conditions. This material has still not been tested in radiation environment. The radiant intensity is guided to the glass optical fibre by use of Snell,s law. In order to interface the

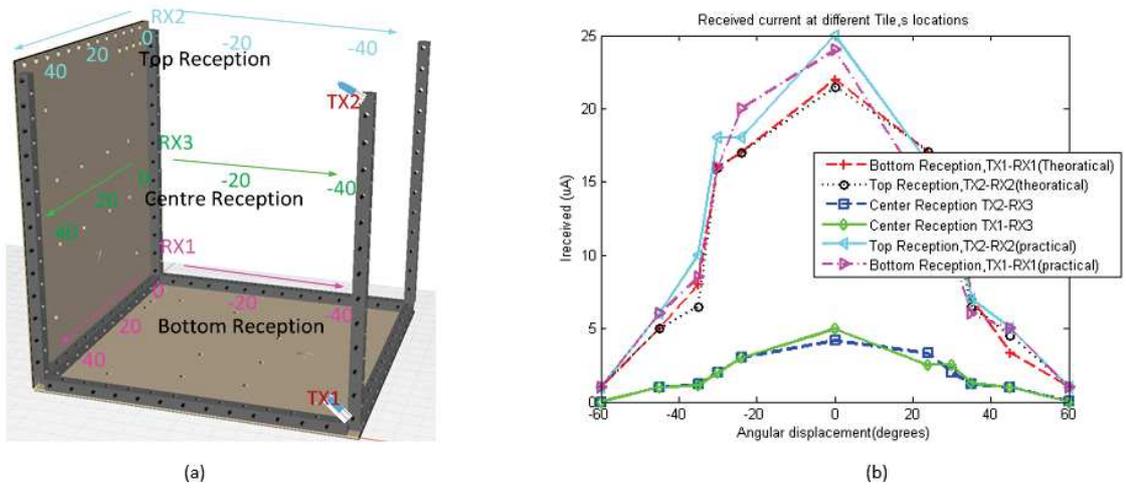


Figure 9.10. Reception at different positions due to transmitters

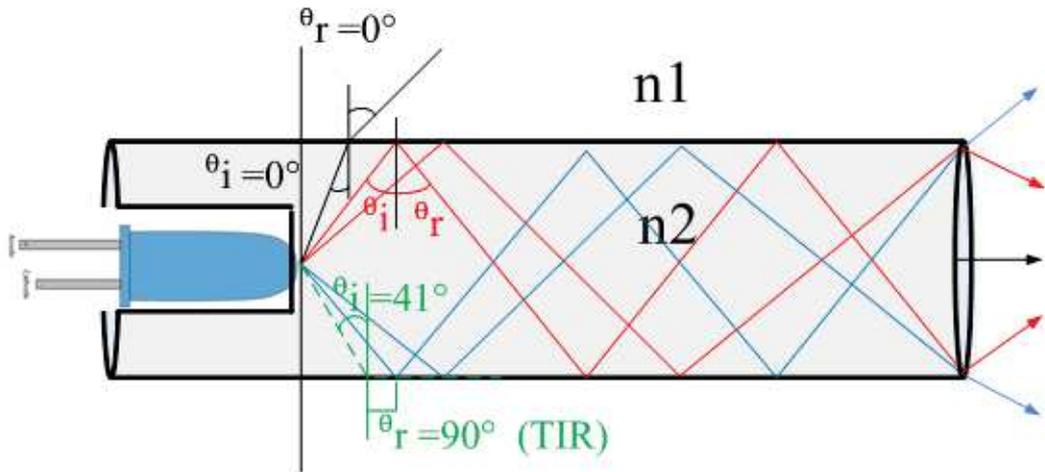


Figure 9.11. Optical light guidance in the Fiber

transmit maximum optical power to the glass fibre, law of refraction is used which relates the indices of refraction  $n$  of the two different media to the directions of

propagation in terms of the angles to the normal given by (9.11).

$$\frac{\sin \theta_i}{\sin \theta_r} = \frac{n_1}{n_2} \quad (9.11)$$

Where  $\theta_i$  is the angle of incidence and  $\theta_r$  is the angle of reflectance. The angles are measured from the normal to the surface, at the point of contact, as shown in Fig.9.11.

The constants  $n$  are the indices of refraction for the corresponding media one being air ( $n_1=1$ ) and the other one glass ( $n_2=1.5$ ).

The angle of incidence of light that make sure that all the optical light is guided inside the glass fibre is given by (9.12)

$$\theta_i = \sin^{-1} \frac{1}{n_2} \quad (9.12)$$

This expression gives critical angle for which the incident ray does not leave the glass fibre, namely when the angle of reflectance is  $90^\circ$ . Any incident angle greater than the critical angle is consequently reflected from the boundary instead of being refracted. Therefore using equation (9.12), we guarantee that any light ray of incident angle from  $41^\circ$  stays inside the glass and hence is received by the receiver. This angle is termed critical angle where Total Internal Reflection (TIR) takes place. The shaded area in Fig.9.8 shows the light radiation guided inside the glass fibre as per above calculations. Fig.9.8 suggests that maximum light emitted by the LED is in between the  $-40^\circ$  to  $40^\circ$  therefore most of the light is successfully reflected in the fiber with very small losses.

Fig.9.12 shows the assembly of six tiles to make a cube structure for AraMiS and possible configuration of glass fiber for inter- as well intra-tile communication. This configuration is used for communication from one tile to any other tile. Each tile uses four double reflector mirrors placed on different positions at certain angles in front of the junction of every four glass fibres. The proposed configuration uses communication in both ways using two separate channels as shown in the Fig.9.12. The reflecting mirrors are placed in such a way that half of the incoming light signal passes straight through and half of it is reflected at right angle to the incident signal. The only shortcoming of this scheme is that some receiving nodes that are close to the transmitters receive high light intensity while the nodes that are placed much far from the transmitter receive very small amount of infrared light. This needs a receiver with high dynamic range to receive from very small currents up to orders of  $nA$  to large currents up to orders of some  $mA$ . The designed discrete receiver has high dynamic range and is able to process the photocurrents of the specified range. The mirrors are double reflectors so the transmission can be carried out either way by putting the transmitters at different positions.

The received current depends on the input optical power and the responsivity of the photodiode at a certain wavelength. The theoretical expression for received current,  $I_{received}$ , is given by (9.13).

$$I_{received} = \frac{Po}{\frac{\pi}{4} * \emptyset^2} * A_{Photo} * Responsivity \tag{9.13}$$

Where  $\emptyset$  is the diameter of the glass fibre,  $A_{photo}$  is the active area of photodiode and responsivity of photodiode is given in  $(A/W)$ . This received current has a high dynamic range depending upon the position of the receiver. Fig 13 shows the photograph of the glass with the guided optical light. Reflecting mirrors are not shown in this photograph.

Table 9.1 shows theoretical and measured values of received current for different values of input radiated optical power. The theoretical results are calculated for *plexiglass* of  $\emptyset = 7.5mm$  and using [5] with photodiode of  $4mm^2$  active area.

No. of Guides	S.No	Radiated Power  Po(mW)	Responsivity  (A/W)	Received current	Received current	Error  (%)
				Theoretical  (mA)	Measured  (mA)	
1	1	10	0.47	0.42	0.39	7.1
	2	20	0.47	0.85	0.78	8.2
2	3	10	0.47	0.21	0.16	23
	4	20	0.47	0.43	0.28	34

Table 9.1. Theoretical and practical received currents for certain input radiated power for guided light communication.

The preliminary results have been obtained taking into account the communication in a single light guide and also using a stage of beam splitter using two light guides. The results using a single light guide are quite acceptable because we were able to guide most of the light into the fiber but the results of using a stage of beam splitter using two guides show very high value of percentage error between the theoretical and measurement results.

## Conclusion

In this chapter, we showed the design technique of building discrete infrared transceiver using COTS components for optimum speed requirements. Effect of TID on photodiodes and LEDs for the entire mission lifetime was also computed. The receiver is effectively able to receive very small amount of current (in the order of nA). The data communication structures using free space and glass fiber give flexibility in the selection of communication channel. The proposed optical communication systems will be used along with the other wired communication approaches for more flexibility.

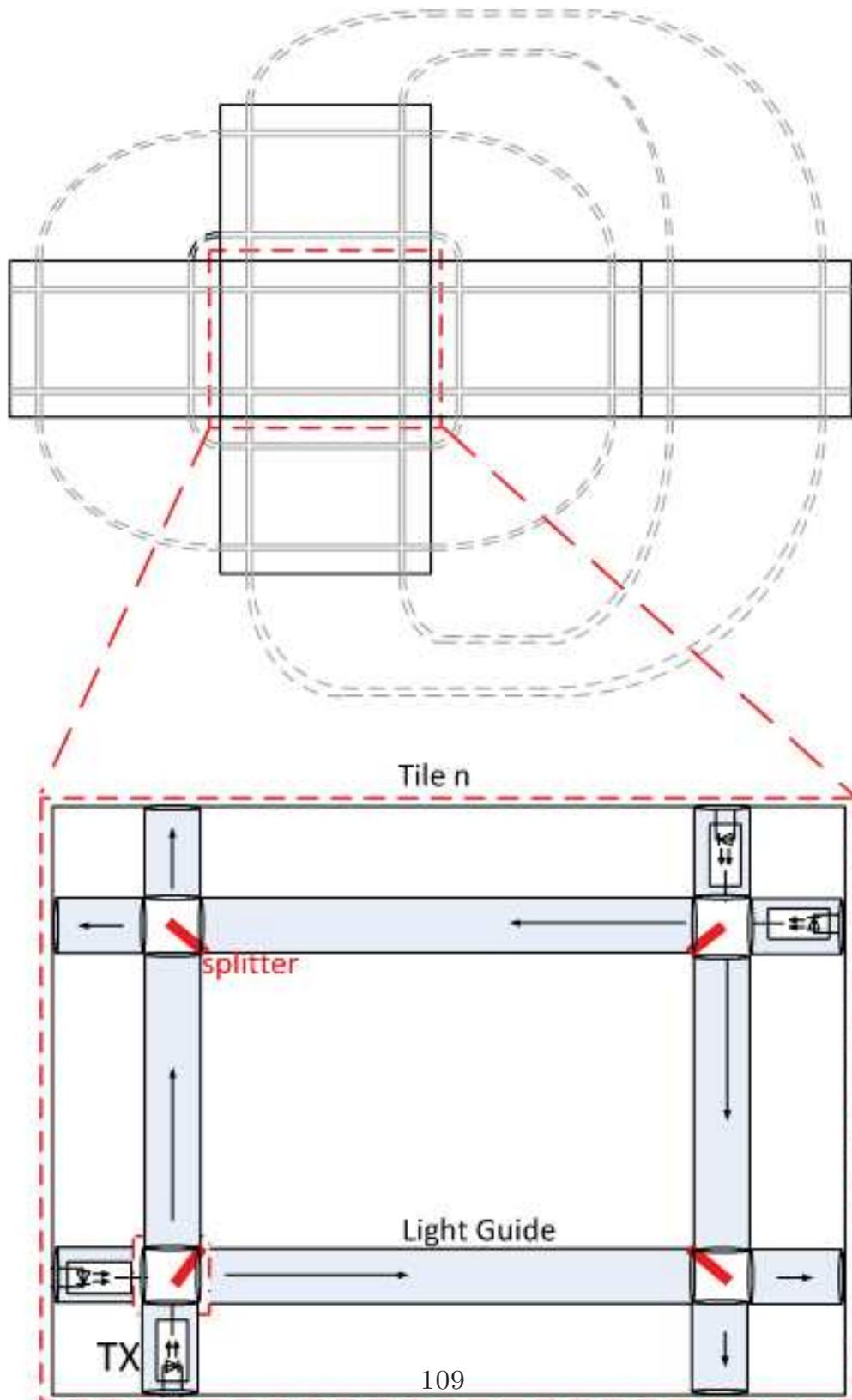


Figure 9.12. Architecture of glass fiber based communication showing tiles



# Chapter 10

## Module Design: On-Board Radio Frequency Module

### 10.1 Introduction

This Chapter focuses on the use of radio frequency modules for on-board data communications. Radio communication between avionics components integrated or installed on-board the spacecraft. There can be a number of advantages achieved using this type of modules including but not limited to

- Reduction of complexity of electrical wiring and harness fabrication with the associated weight saving and higher overall fuel efficiency
- Significant gain in reconfigurability through improved installation flexibility
- Reliable monitoring of parameters belonging to moving or rotating parts.

Therefore keeping in mind the advantages of wireless radio frequency based communications, an AraMiS [RF](#) module having as much compatibility with the current subsystem modules is being designed. The initial experiments using the commercial transceiver modules give a good indication for the potential use of these modules in the intra-satellite on-board data communication

### 10.2 Overview

After much literature review, radio frequency communication system has been developed using TI CC2510 and CC1110 transceiver modules. The protocol stack has been kept compliant with propriety simpliciTI protocol. The receive signal strength for different variable power levels have been computed and plotted. The packet

format for the receive packets has been detected using a commercial packet sniffer software and the received packets from different radio transceivers have been received with a very small packet error rate. We are able to transmit packets with output transmit power as low as up to -30dBm with acceptable average Receive Signal Strength Indicator (RSSI). A complete mapping strategy for wireless communication protocols has been formed considering the limited communication channels available on board the small satellites. All the possible communication systems have been made complaint with the basic protocol of the AraMiS so that the end user is able to use them without knowing much detail about the physical layer of each protocol.

### 10.3 Hardware Platform

The choice of hardware selection for implementing the proposed radio frequency based communication protocol for intra-satellite communication is based on hardware and software selection. The hardware is selected with

- a low-power CPU and low-power radio.
- a programmable radio and CPU in terms of their power mode.
- an open interface for easily expanding the functionality.

Based on the above mentioned considerations, the TI CC1110/CC2510 System on Chip (SoC) was selected [67]. The transceiver has to be made compatible with the electrical and mechanical interface described in chapter 2, section 2.2 therefore an independent module design was needed. The module is under design process and therefore, the commercially available modules were used for testing purposes.

#### 10.3.1 CC 1110/2510

The testing was performed using commercial TI transceivers. The CC2510 and CC1110 provide a wireless solution for intra-satellite communication system. The CC2510 is designed to offer a wireless communications at 2.4GHz, with a bit rate up to 500k Baud. In order to minimize the cost, CC2510 has a highly integrated 8051-compatible micro-controller and up to 32kB of in-system programmable flash memory.

### 10.4 Software Platform: simpliciTI

The software platform for intra tile radio frequency based communication was chosen based on SimpliciTI protocol [68–71]. SimpliciTI is a TI proprietary low-power

RF network protocol. SimpliciTI has very low cost of memory; it only needs less than 8k flash memory and 1k Random Access Memory (RAM) space depend on configuration. It supports 2 basic topologies: strictly peer-to-peer and a star topology. The protocol support is realized in a small number of Application programming interface (API) calls. These APIs support customer application peer-to-peer messaging. The association between two applications, called linking, is done at run time. SimpliciTI support sleep mode for the devices in order to extend the working time. The UML class diagram of implemented communication system using simpliciTI is shown in Fig.10.1.

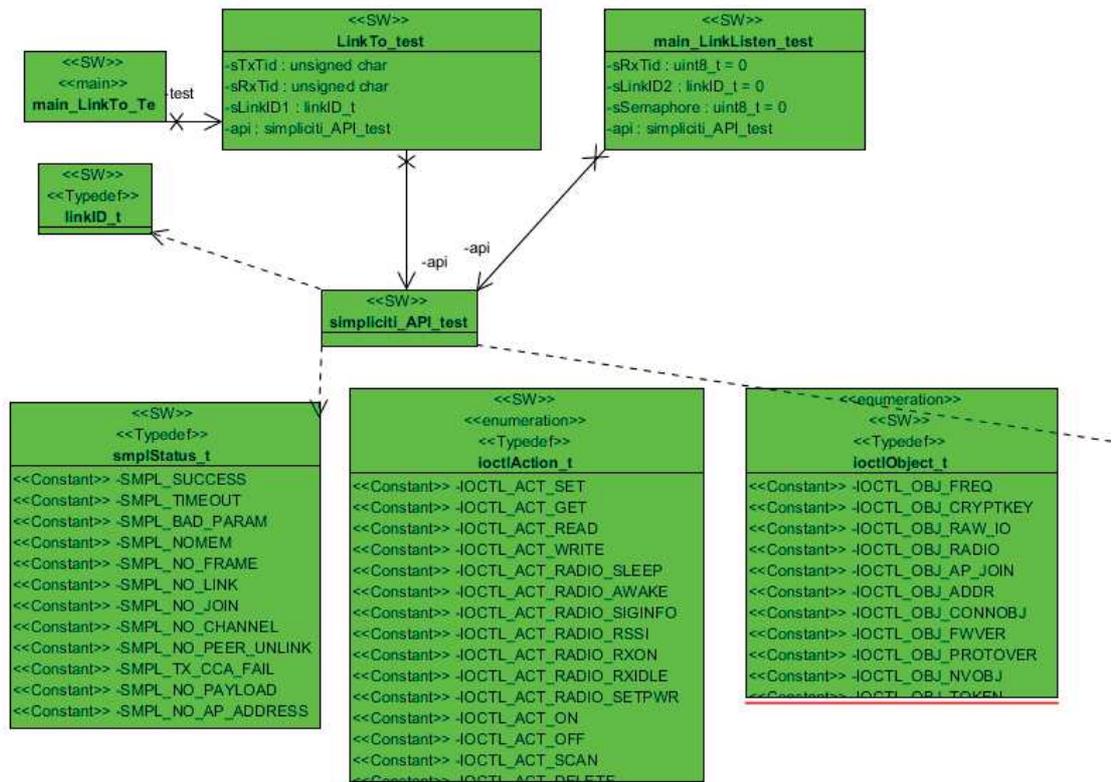


Figure 10.1. simpliciTI Class Diagram

### 10.4.1 API

This section describes the application programming interface for SimpliciTI protocol [72]. The API provides an interface to the services of the SimpliciTI protocol stack. SimpliciTI initialization involves three stages of initialization: board, radio, and stack. Board initialization (BSP) is deliberately separated from the radio

and stack initialization. The radio and stack initialization occur as a result of the SimpliciTI initialization call. The board initialization is a separate invocation not considered part of the SimpliciTI API but it is noted here for completeness. The BSP initialization is partitioned out because customers may already have a BSP for their target devices. Making the BSP initialization explicit in the SimpliciTI distribution makes it easier to port to another target.

### **Board Initialization**

SimpliciTI supports a minimal board-specific BSP. The BSP scope includes GPIO pin configuration for LEDs, switches, and a counter/timer used for protocol chores. It also includes SPI initialization for the dual-chip RF solutions.

### **Radio Initialization**

Radio registers are populated and the radio is placed in the powered, idle state. Most of the radio registers are based on exported code from SmartRF Studio. The default channel is set with the first entry in the channel table.

### **Stack Initialization**

All data structures and network applications are initialized. In addition the stack issues a Join request on behalf of the device. The Join request will fail in topologies in which there is no Access Point. This is expected in this topology and is not an error condition.

## **10.4.2 Packet Sniffer**

The SmartRF Packet Sniffer is a PC software application used to display and store RF packets captured with a listening RF hardware node. The Packet Sniffer filters and decodes packets and displays them in a convenient way, with options for filtering and storage to a binary file format. Figure 10.2 shows the capture of packets by SmartRF packet sniffer.

## **10.5 Implementation**

The simpliciTI protocol stack was implemented for point to point communication testing inside the satellite. Fig. 10.2 shows the transmitter sending a broadcast request for joining and consequently, the receiver device with the given destination address joining the network. The intra satellite communication was tested by placing the receivers at different locations w.r.t. receivers in certain CubeSat configurations

P.nbr.	Time (ms)	Length	Dest. Address	Source Address	Application payload	User Port	RSSI (dBm)	LQI	FCS
1	+0	21	87 56 34 12	54 76 03 05	AraMi S!!!!	0x20	-22	24	OK
2	+3	21	87 56 34 12	54 76 03 05	AraMi S!!!!	0x20	-21	12	OK
3	+6	21	87 56 34 12	54 76 03 05	AraMi S!!!!	0x20	-22	25	OK
4	+9	21	87 56 34 12	54 76 03 05	AraMi S!!!!	0x20	-21	13	OK
5	+12	21	87 56 34 12	54 76 03 05	AraMi S!!!!	0x20	-22	17	OK
6	+15	21	87 56 34 12	54 76 03 05	AraMi S!!!!	0x20	-21	24	OK
7	+18	21	87 56 34 12	54 76 03 05	AraMi S!!!!	0x20	-21	13	OK

Figure 10.2. Packet Sniffer showing a string transmitted between two modules

and the receive signal strength was measured. Fig.10.3 shows the measured RSSI for different power levels of transmitters. The initial result indicate that we can communicate across the tiles with as low as -30dbm transmit power which is very low as compared to the optical solution. Moreover, no line of sight is required to accomplish the communication across the tiles.

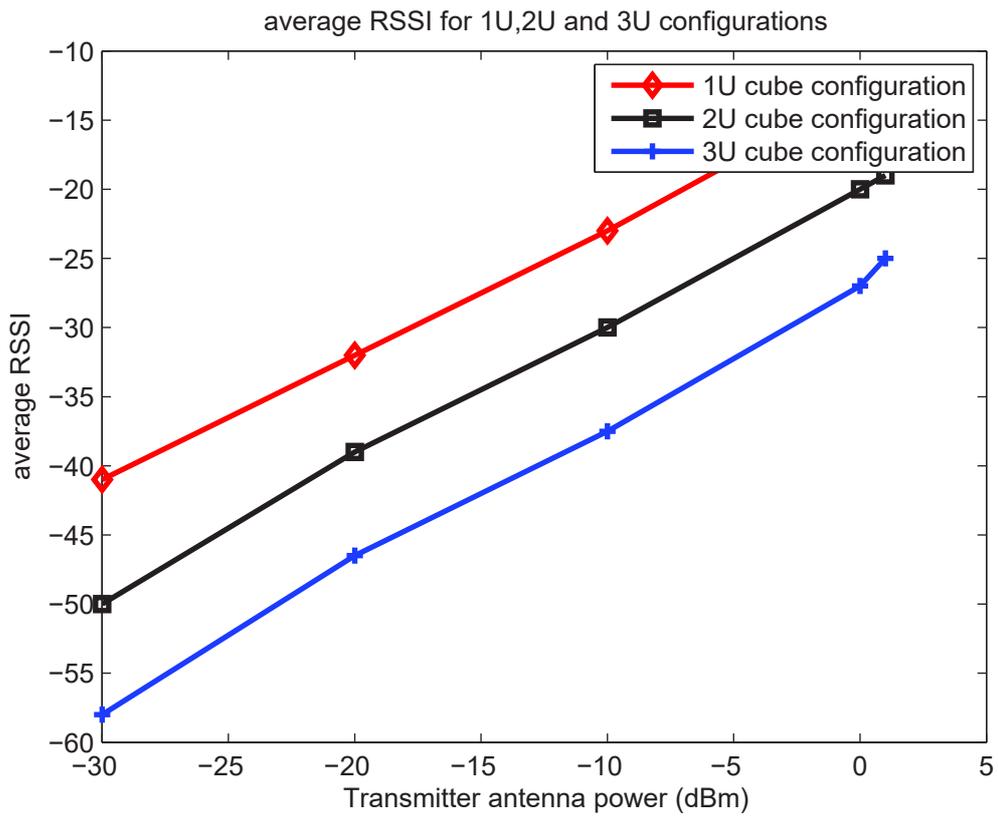


Figure 10.3. average RSSI for certain transmit power levels

# Chapter 11

## 1C25 Functional Testing

### 11.1 Introduction

This chapter shows an innovative testing board developed for the functional testing of small satellites. The testing of subsystem modules is very easy since there is no much complexity in testing the subsystem modules. When tested, these modules can be permanently embedded into the satellite on demand and reusable design configurations. Therefore a testing board with all the necessary interface points is required for functional testing of modules, tiles and whole satellites. The hardware of the test board consists of all the necessary interface points (either physical connectors or logical slots) for pluggable modules for testing purposes. The signals of these interface points have been spread to the two test processors which are also available on the test board. The logical slots interface the internal subsystems present on the board. One of the processors hosts interfacing points to physical quadruple module connectors for pluggable modules. Whereas the other processor hosts logical slot interfaces to the internal subsystems of the testing board including calibration memory, primary power switching, voltage and temperature sensors in addition to the physical interfacing points.

A large number of test points and LEDs have been used in order to simplify the testing process. On the physical test board a number of voltage and current sensors have been used in order to measure the voltage and current levels. The test board interfaces with the testing software running on a computer via a level shifter. The level shifter translates the logic level of UART signals to interface with those of computer.

Each module, tile and satellite has a separate configuration files stored in the software setup therefore any desired module, tile or satellite can be selected by the software for functional testing.

## 11.2 Functional Test Board

The honeycomb tile has been designed in such a way that a small modification is needed to convert it to a test board for functional testing of modules, tiles and whole satellites. The class diagram of functional test board is shown in [A.1](#).

The necessary modification in 1B81 Power Management Honeycomb tile to transform it to the functional test board are following

- Various sensor described by class Bk1B133A\_Temperature\_Sensor .
- RS232 Level Shifter together with the 4-pin connector change logic levels from RS232 standard to TTL.
- Inter Communication Module connects two on-board microprocessors MSP430F5438.
- The JTAG connectors for programming the MSP430.
- Necessary LEDs, test point and jumpers in order to make testing process more user friendly.

The shadowed area is the main part of testing board, on which there are two microprocessors. Each processor has two quadruple module slots. The testing board is connected to PC by a RS232 shifter, where several configuration file rules are defined in the class TileIndex.txt, class Tile Description File and class Module Description File. This is shown in the [Fig.11.1](#). On right side of the testing board, it can connect

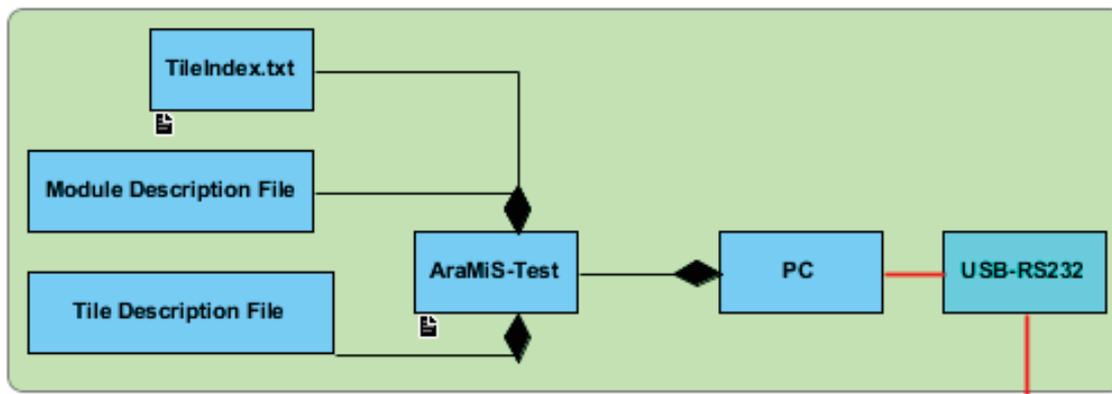


Figure 11.1. Software setup of functional testing

to all the possible modules for testing, they can be a general single module, a tile or a complete satellite as shown in the [Fig.11.2](#). The figure [11.3](#) shows a functional test board showing some modules under test mounted on it

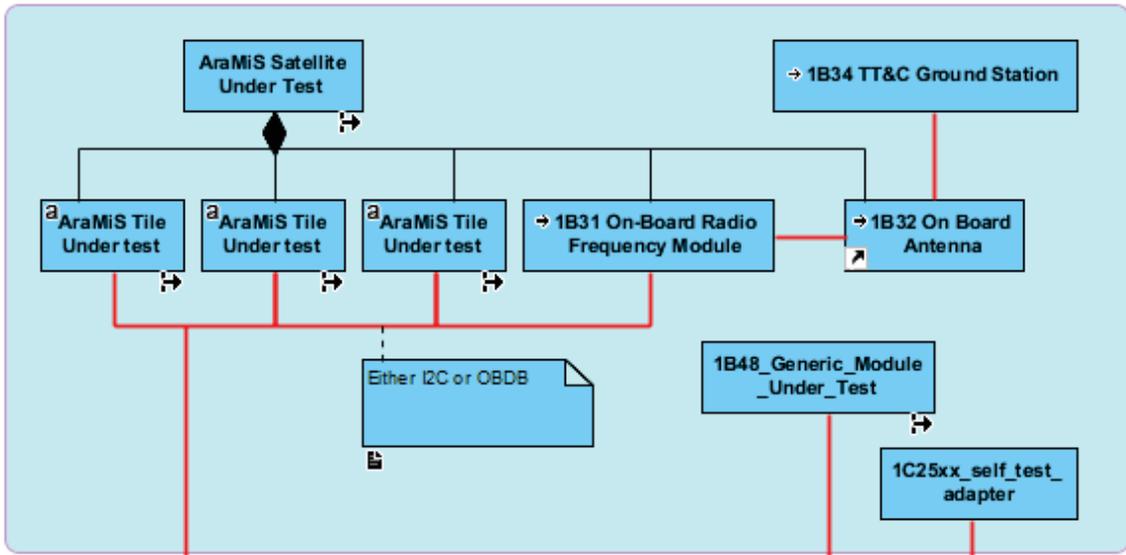


Figure 11.2. Test side of functional testing

## 11.3 1C25 Use Case Diagrams

### 11.3.1 1C25 Use Cases

The generic use case diagram of the test setup is shown in the Fig.11.4 emphasizing certain actors interacting with the system. The actor test is the in-charge of the test process. It either reads, writes data or sends command to either modules, tiles or the whole satellites as part of the testing process.

#### Read Data

Tester reads data from AraMiS Satellite Under Test, AraMiS Tile Under test and 1B48\_Generic\_Module\_Under\_Test. The data mostly are digital signals. and it may read the working mode, read error etc.

#### Write Data

Tester writes data to 1B48\_Generic\_Module\_Under\_Test, AraMiS Satellite Under Test and AraMiS Tile Under test. The data can be written into several registers in the modules and tiles. The data can contain the information of setting the working parameters, etc.

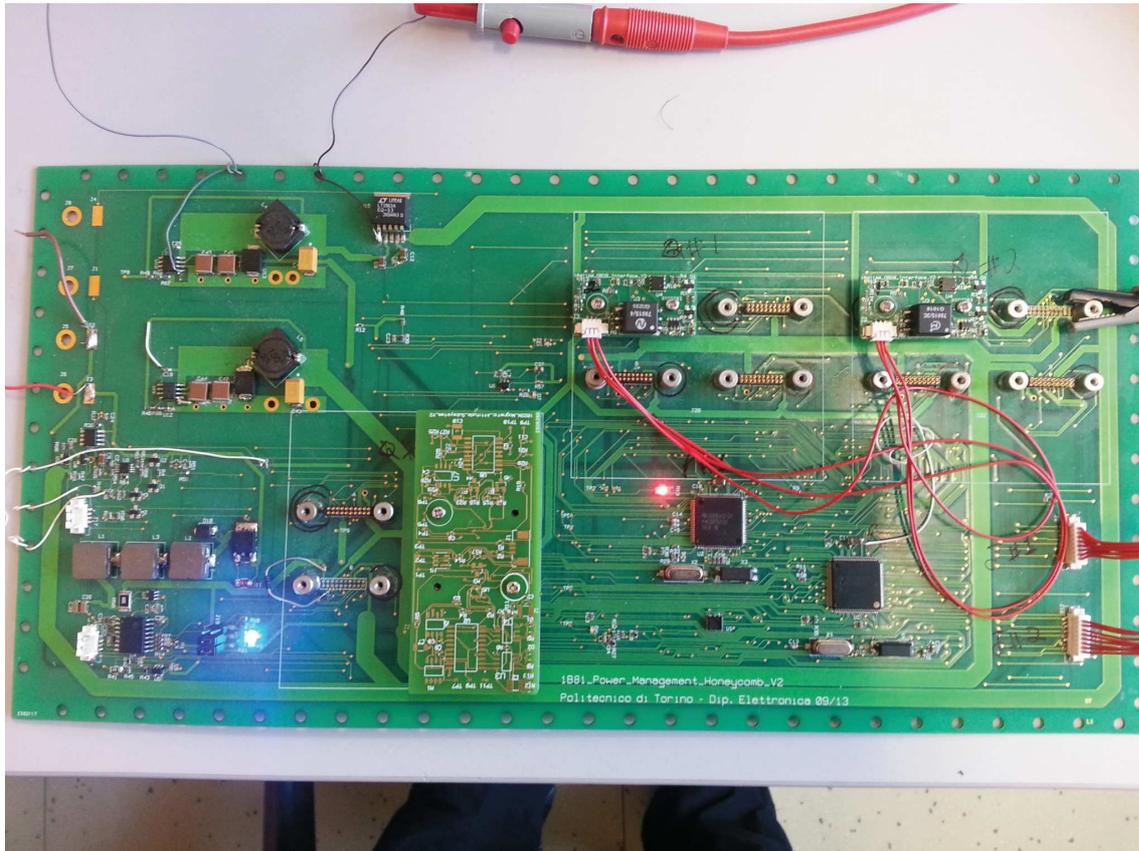


Figure 11.3. Functional test board with modules under tests

### Send command only

The Tester shall type on the PC a command. Tester sends the command to AraMiS Satellite Under Test and does not require any data, but only change the module's working modes. These commands maybe set or reset attach or detach the modules, etc.

### 11.3.2 Configuration Use Cases

The section describes all use cases related to the configuration, including use case diagram, documentation, how to define the configuration file and where to put the configuration files. Fig.11.5 shows an actor *Configurator* who is person in charge of configuring HW/SW parameters according to spacecraft architecture and mission requirements. The following use cases have been defined

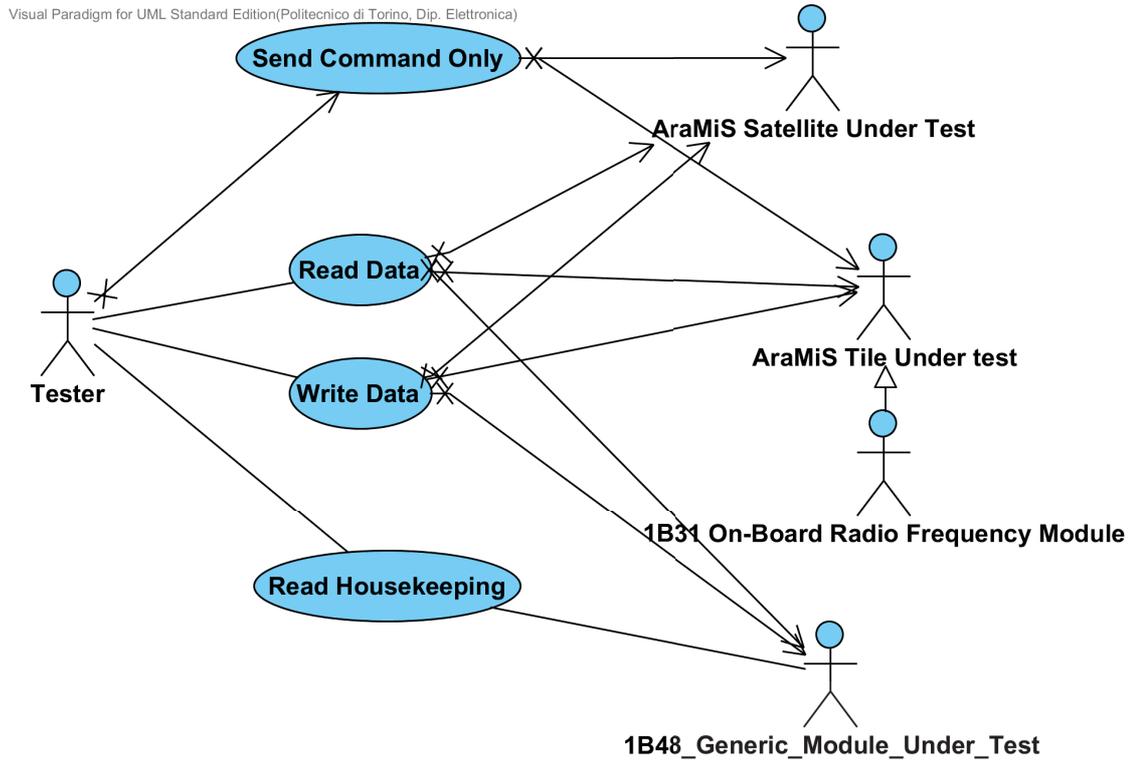


Figure 11.4. Functional testing use case diagram

## Configure Module

The Configurator describes the internal architecture of each type of module by means of an ASCII Module Description File. The more details about the configuration file can be found in *1B45 Subsystem Serial Data Bus*.

The name of the files should be the names of the module types with extension .txt.

The Module Description File has a first line contains the number (N) of configuration bits (usually 16) followed by one line per each function in the module.

Each line contains, separated by commas

- Each bits of a configuration word in the module ;
- Name of function to configure;

An AraMiS module description file is shown in Fig.11.6 The file is read by the AraMiS-Test program when it is started. To change tile configuration, the Configurator has to exit the AraMiS-Test program, change file content and start AraMiS-Test program again.

Visual Paradigm for UML Standard Edition(Politecnico di Torino, 2010, ISBN 978-88-904-0000-0)

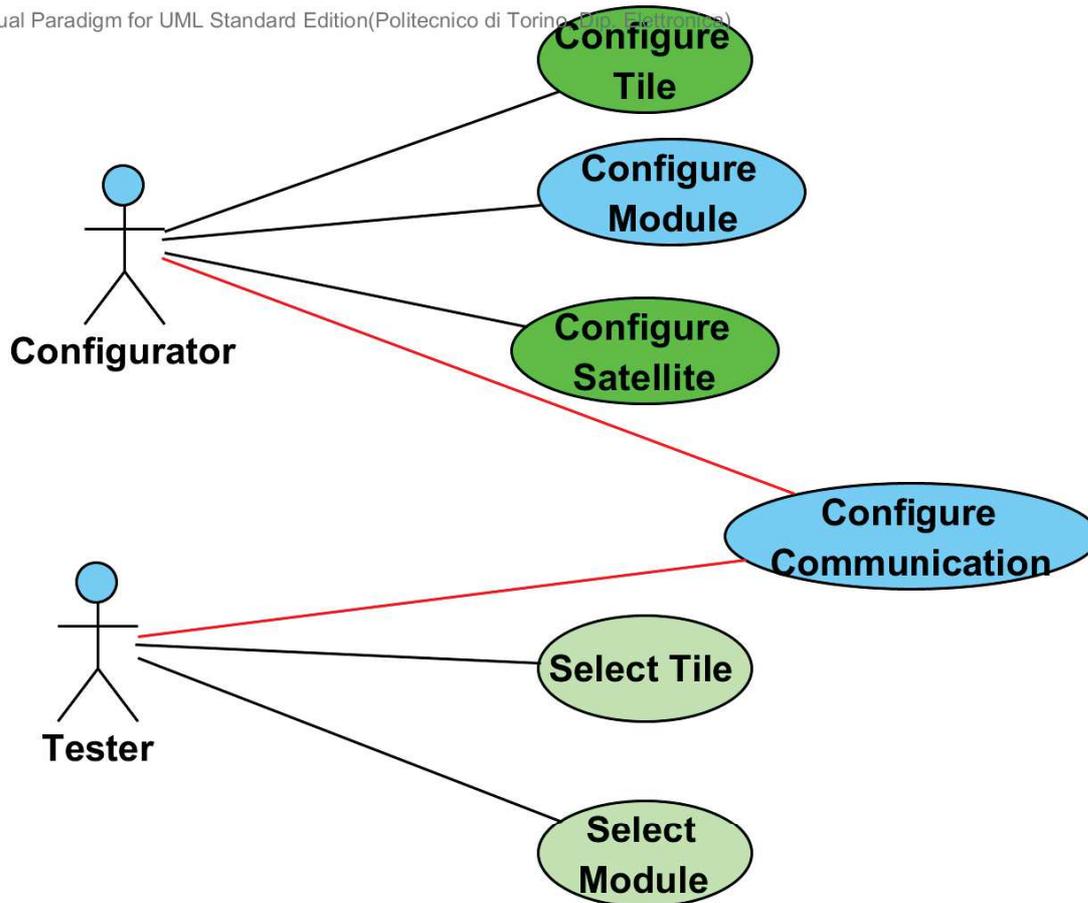


Figure 11.5. Configuration use case diagram

### Configure Tile

The tile description file has a first line which contains the number (N) of subsystems hosted in the tile followed by one line per subsystem formatted as: CW[n], subsystemName : String[N], subsystemType : String[N], where  $0 \leq n < N-1$ . The file is read by the AraMiS-Test program when it is started. To change tile configuration, the Configurator has to exit the AraMiS-Test program, change file content and start AraMiS-Test program again.

The Configurator describes the tiles and names by means of an ASCII file TileIndex.txt. The file has a first line which contains the number N : int of tiles followed by one line per each tile in the satellite. Each line contains, separated by commas

- The name (identifier) of the tile inside the satellite;

```

16
BIT[0],Activate Coil,
BIT[1],Activate sensor,
BIT[2],dettach sensor,
BIT[3],activate battery,
BIT[4],xxxxxxxxxx,
BIT[5],xxxxxxxxxx,
BIT[6],xxxxxxxxxx,
BIT[7],xxxxxxxxxx,
BIT[8],xxxxxxxxxx,
BIT[9],xxxxxxxxxx,
BIT[10],xxxxxxxxxx,
BIT[11],xxxxxxxxxx,
BIT[12],xxxxxxxxxx,
BIT[13],xxxxxxxxxx,
BIT[14],dettach,
BIT[15],Attach,

```

Figure 11.6. Module Description File

- Type of the tile; each tile type shall be associated with a file named Tile-Type.txt
- The physical address of the tile in the satellite

An AraMiS tile description file is shown in Fig.11.7

### Configure satellite

The Configurator describes the tiles, modules and whole satellites by means of an ASCII description files which have certain level of details. Normally, the name of the file should be the name of the module or tile with extension.txt

As an example, Satellite configuration file for an AraMiS-C1 satellite under test is as follows.

```

6
Tile1,1B8_CubePMT,0x35,
Tile2,1B8_CubePMT,0x36,
Tile3,1B8_CubePMT,0x38,
Tile4,1B8_CubePMT,0xe9,
Tile5,1B9_CubeTCT,0x1A,

```

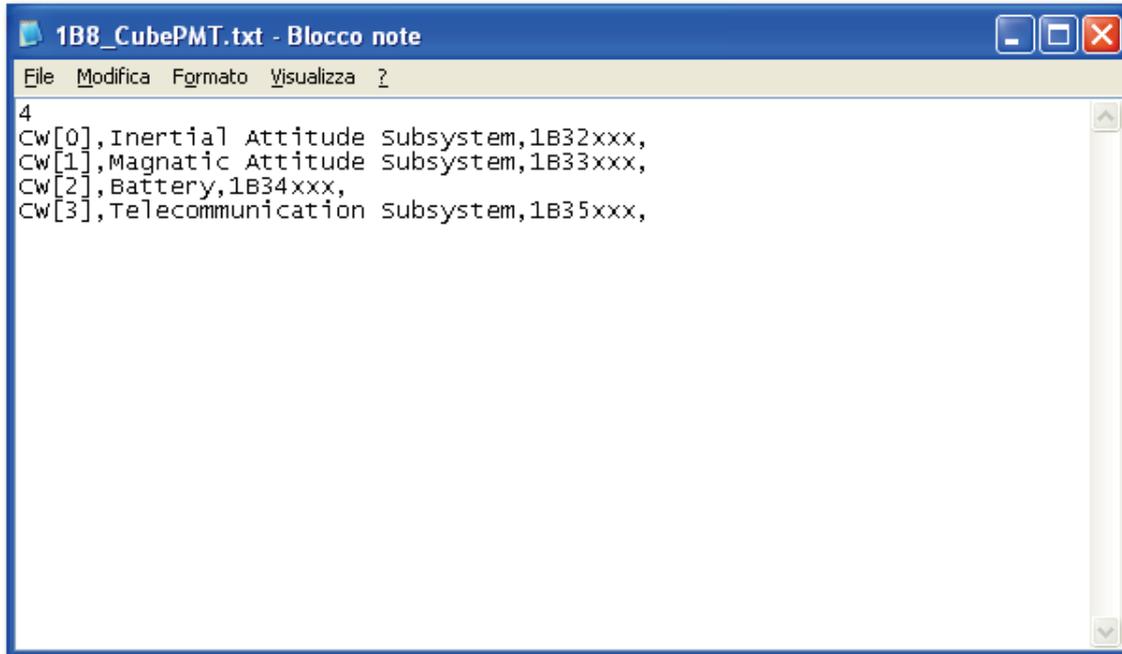


Figure 11.7. A tile description file

Tile6, 1B9\_CubeTCT, 0x1B,

An AraMiS satellite description file is shown in Fig.11.8

The file TileIndex.txt is read by the PC when the program AraMiS-Test is started. To change satellite configuration, one has to exit the AraMiS-Test, change content of file TileIndex.txt and start AraMiS-Test again.

Similarly, by using select tile and select module use cases, the Tester shall select which tile he wants to interact with by means of a pop-up menu showing the list of tiles, their types and physical addresses.

### 11.3.3 Configuration and Status Test Use Cases

The set of use cases define the configuration and status test for any module, tile or satellite under test. Fig.11.9 shows the top level use case diagram.

#### Get Module Status

Firstly, the tester selects tile and module (by means of Select Module and Select Tile) in order to start transfer by means of Get Module Status from *1B45 Subsystem Serial Data Bus*. This in turn will use Basic Protocol in its RS232 protocol variant

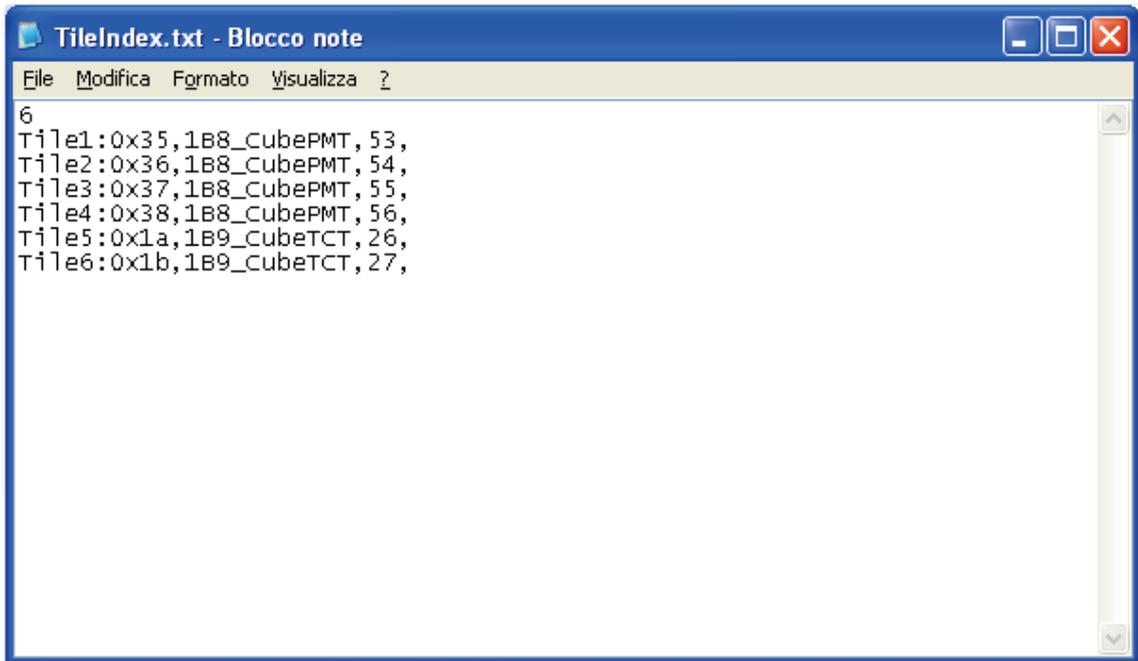


Figure 11.8. A satellite description file

to communicate data from the PC to either an AraMiS Satellite Under Test or to an AraMiS Tile Under test; from there to the *1B48 Generic Module Under Test*.

Then returns the statusRegister: CS\_REDUNDANCY[LENGTH\_STATUS] to the AraMiS-Test and shown in box

### Get Module Configuration

Firstly, the tester selects tile and module (by means of Select Module and Select Tile) in order to start transfer by means of Get Module Status from *1B45 Subsystem Serial Data Bus*. This in turn will use Basic Protocol in its RS232 protocol variant to communicate data from the PC to either an AraMiS Satellite Under Test or to an AraMiS Tile Under test; from there to the *1B48 Generic Module Under Test*.

Then returns the configRegister: CS\_REDUNDANCY[LENGTH\_STATUS] to the AraMiS-Test and shown in box

### Set Module Configuration

First Tester selects tile and module (by means of Select Module and Select Tile). The Tester then selects either individual bit or multiple bits (up to 16) to be set in the selected module by means of the test setup menu. Each bit will also indicate

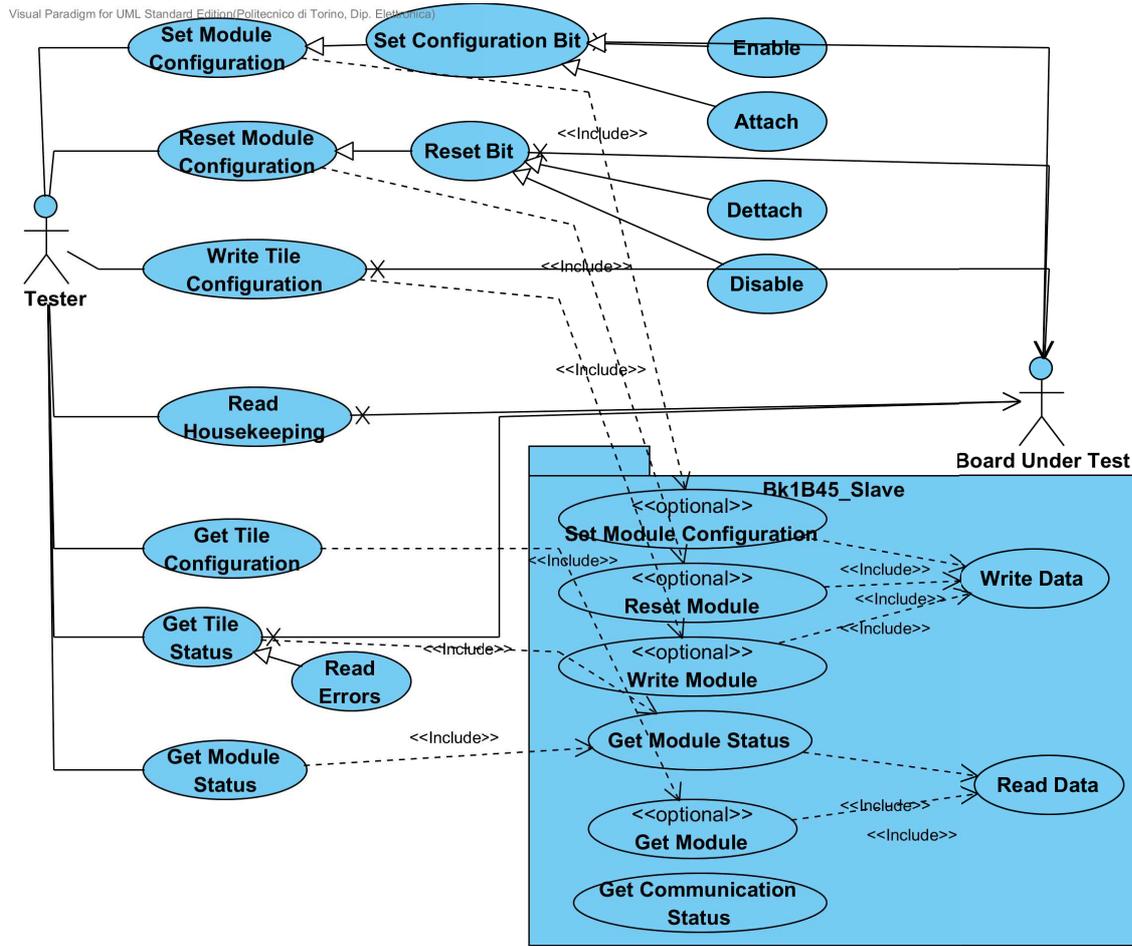


Figure 11.9. A satellite description file

meaning as defined in Configure Module. The Tester will then start transfer by means of Set Module Configuration from *1B45 Subsystem Serial Data Bus*.

This in turn will use Basic Protocol in its RS232 protocol variant to communicate data from the PC to either an AraMiS Satellite Under Testor to an AraMiS Tile Under test; from there to the *1B48 Generic Module Under Test*.

### Reset Module Configuration

First the Tester selects tile and module (by means of Select Module and Select Tile).

The Tester then selects either individual bit or multiple bits (up to 16) to be reset in the selected module by means of the following menu

The Tester will then start transfer by means of Reset Module Configuration from

*1B45 Subsystem Serial Data Bus.*

This in turn will use Basic Protocol in its RS232 protocol variant to communicate data from the PC to either an AraMiS Satellite Under Tester to an AraMiS Tile Under test; from there to the *1B48 Generic Module Under Test*.

## 11.4 Testing of Functional Test Board

Since the test board is a complete hardware consisting of processors, plug and play connectors, temperature, current and voltage sensors, power and regulator stages, therefore the testing of the test setup becomes important.

### 11.4.1 Regulators

The voltage generated by the solar cells was applied to the buck converter in order to output the PDB output and was spread to all the connectors. In addition the solar voltage was applied independent to the 5V switching regulator and 3.3V mixed regulator stages. We were able to get the stable 5V, 3.3V and 3V reference voltages spread all around the subsystems, connectors and processors.

### 11.4.2 Processor

Both the MSP430 tile processors were debug using the IAR embedded workbench. The LEDs attached on the board were programmed using the processors and functioning according to the design.

### 11.4.3 Crystal Oscillators

The crystal oscillators present on the board generate different types of clocks for the processor including Auxiliary Clock (ACLK), Master Clock (MCLK) and Sub Main Clock (SMCLK). The low and high frequency oscillators are used as a source of Unified Clock System. The UCS module for the functional test board includes

- XT1CLK: Low-frequency/high-frequency oscillator that can be used either with low frequency 32768 Hz watch crystals, standard crystals, resonators, or external clock sources.
- XT2CLK: A 4MHz high-frequency oscillator

The crystal oscillators are functional if and only if the fault flags for crystal 1, crystal 2, DCO are cleared. If there exists some fault condition the corresponding flag is set. Any desired frequency i.e. ACLK, MCLK or SMCLK can be selected from either

XT1 or XT2 after the oscillators have been verified to be operational. A routine to clear all fault flags and select any desired clock source is shown below.

```

void config_ucs_xt1xt2(){

    //Setting ACLK= 8MHz;MCLK=8MHz external crystal

                                // output ACLK
P5SEL |= 0x0C;                    // Port select XT2
P7SEL |= 0x03;                    // Port select XT1

UCSCTL6 &= ~(XT1OFF + XT2OFF);    // Set XT1 & XT2 On
// UCSCTL6 |= XCAP_3;            // Internal load cap

UCSCTL3 |= SELREF_7;             // FLL Ref = No selection

UCSCTL6 &= ~(XT1DRIVE0 | XT1DRIVE1 | XT2DRIVE0 | XT2DRIVE1);
    // Set XT1 On...XT1DRIVE0 is defined as 0x0040 i.e 0000 0000 0100 0000
    // XT1DRIVE1 is defined as 0x0080 i.e 0000 0000 1000 0000
//UCSCTL6 &= ~(XT1OFF |(3<<6));
// see ucsctl6..(3<<6) means shift 11 by six positions left i.e xt1drive bits
//UCSCTL6 |= XT1OFF;            // Drive strength, adjusted
                                // according to crystal frequency of 8 MHz
                                // LFXT1 HF mode
UCSCTL1 |= DISMOD;              // turn off DCO FLL
__bis_SR_register(SCG0 | SCG1);
// Turn off DCO FLL , SR syntax is __bis_SR_register(bits), see data sheet for details
// Loop until XT1,XT2 & DCO stabilizes
do
{
    UCSCTL7 &= ~(XT2OFFG + XT1LFOFFG + XT1HFOFFG + DCOFFG);
    // Clear XT2,XT1,DCO fault flags

    SFRIFG1 &= ~OFIFG;          // Clear fault flags
    __delay_cycles(250000);
} while (SFRIFG1&OFIFG);        // Test oscillator fault flag

UCSCTL4 = SELA_0 + SELM_5;      // Select ACLK = XT1
                                // MCLK = XT1

```

}

#### 11.4.4 House Keeping Sensors

The housekeeping sensors including voltage, current, temperature and sun sensor are connected to the analog channel of the processor and they can be tested by reading the corresponding analog channels. The ADC core converts an analog input to its 12-bit digital representation and stores the result in conversion memory. The conversion formula for the ADC result  $N_{ADC}$  is given in eq. 11.1

$$N_{ADC} = 4095 \times \frac{V_{IN} - REF^-}{V_{REF+} - REF^-} \quad (11.1)$$

The following piece of code shows a simple routine to configure control registers and read the voltage of the voltage sensor for 12 bit ADC.

```
#include "io430.h"
int main( void )
{
volatile unsigned int i;
// Stop watchdog timer to prevent time out reset
WDTCTL = WDTPW + WDTHOLD;
P6SEL |= BIT7;
P6DIR &= ~BIT7;
UCSCTL4 |= SELM_5;
ADC12CTL1 = ADC12SSEL_2;
ADC12CTL0 = ADC12ON + ADC12SHT02 + ADC12REFON + ADC12REF2_5V;
// Turn on ADC12, Sampling time-
// On Reference Generator and set to
// 2.5V
ADC12CTL0 &= ~ADC12ENC;
ADC12CTL2 |= 0x0020;
ADC12MCTL0 |= 0x17; // Analog Channel to read
ADC12CTL1 = ADC12SHP; // Use sampling timer
/*ADC12MCTL0 = ADC12SREF_1; */ // Vr+=Vref+ and Vr-=AVss
for ( i=0; i<0x0F30; i++); // Delay for reference start-up
ADC12CTL0 |= ADC12ENC; // Enable conversions
while (1)
{
ADC12CTL0 |= ADC12SC; // Start conversion
while (!(ADC12IFG & BIT0));
// __no_operation(); // SET BREAKPOINT HERE
```

```
for ( i=0; i<0x0F30; i++); // Delay for reference start-up
}
}
```

The conversion results are stored in the memory register which can be used in eq. 11.1 to calculate the stored value and hence the value of the corresponding sensor.

### 11.4.5 Module Testing

After the processors, clocks and on-board sensors have been tested, the next task is to test the modules. The 1B411 on board data bus module discussed in chapter 8 was tested for intra-satellite communication. Some ASCII characters were transmitted from one tile to the other one using the OBDB modules. The routine at the transmitter side is shown below.

```
#include "msp430x54xA.h"
#include "string.h"
void config_ucs_xt1xt2();
volatile unsigned char Uart1TxBuf[256];
void sendstring(char *str);
volatile int count=0;
volatile int i = 'A';
char done = 0;
void main(void)
{
    WDTCTL = WDTPW + WDTHOLD; // Stop WDT
    P11DIR |= BIT1 + BIT0; // P11.1,0 to output direction
    P11SEL |= BIT1 + BIT0; //P11.1 to output MCLK and P11.0 to output ACLK
    P3SEL = 0x30; // P3.4,5 = USCI_A0 TXD/RXD
    config_ucs_xt1xt2();

    UCAOCTL1 |= UCSWRST; // **Put state machine in reset**
    // setting UCSSEL to ACLK
    UCAOCTL1 |= 0x40;
    UCAOCTL1 &= ~0x80; //(just go to the definition ..Here we set 01xxxxxx)

    UCAOBRO = 0x50; // 8MHz 1 MHz (see User's Guide)
    UCAOBR1 = 0;

    CAOMCTL |=UCBRS_0+UCBRF_0;//+UCOS16;
```

```
UCAOIRTCTL =1*UCIRTXPLO+1*UCIRTXPL3+1*UCIRTXPL4+1*UCIRTXPL5+UCIREN;  
    // Irda transmit control with 3/16 of pulse  
  
UCAOCTL1 &= ~UCSWRST;    // **Initialize USCI state machine**  
  
UCAOIE |=UCTXIE;        //enabled tx interrupt  
__bis_SR_register(LPM3_bits+GIE); //General interrupt enable  
__no_operation();       //As per IAR debug instructions  
  
    while(done == 0);  
    sendstring("Hello world from politecnico; Department of Electronics");  
while(1){sendstring("ARAMIS");}  
}
```

We were able to transmit ASCII characters from one tile to the other one using the OBDB interface even after injecting different possible errors discussed in [chapter 8](#).



# Chapter 12

## Conclusion

This thesis has mainly focused on the faster, better and cheaper philosophy in the design of small spacecrafts emphasizing the modularity at different levels. Moreover, design of small modules for intra-satellite data communication helped in the simplification of data communication interfaces.

The main contribution is the smart harness technique which is a completely modular and flexible approach in the design of small satellites. It gives more flexibility in designing the satellite in any desired shape like cubic, hexagonal etc. Multiple choices of on-board data communication solutions reduce the chance of erratic transmission. The UML based design for hardware/software control of each tile's telemetry and housekeeping is also quite flexible and modular. Finally, the pluggable single, double and quadruple modules are capable of handling all the core functions of the satellites and can be designed according to specific payload for every mission.

The complete life cycle of a module subsystem was proposed by use of powerful features of UML . There are many advantages of UML based design over typical design approach. The UML design approach is helpful to better understand the functionality of the system and overcome design limitations on the early stages of the system. Design complexity and development time is reduced by using proposed approach. Extensive documentation of each diagram helps in understanding the system much better, reuse of UML singleton classes helps minimize probability of error. In addition, interaction between different blocks is much easier. UML tool can generate code for several programming languages. The generated code is quite clear, readable and well documented and properly commented. Moreover, the generated code clearly depicts the system design in terms of relationship between classes, their associations, etc. The whole life cycle is applied to all the subsystems from idea until the development and testing. After the module design, the design of satellites either using physical module based configuration, satellite on demand configuration and reusable design configuration was performed. By using the reusable design configuration, we can assemble any customer required system in a very short span

of time.

A number of subsystem modules were realized for intra-satellite communication purposes. The wiring harness of the wired communication across different tiles of the satellite is significantly reduced using the on-board data bus module. The testing results emphasized that the proposed module was quite functional even in case of severe fault conditions.

The next contribution was the design of innovative free space communication solutions for intra-satellite communication.

For optical module design, showed the design technique of building discrete infrared transceiver using COTS components for optimum speed requirements. Effect of TID on photodiodes and LEDs for the entire mission lifetime was also computed. The receiver is effectively able to receive very small amount of current (in the order of nA). The data communication structures using free space and glass fiber give flexibility in the selection of communication channel. The proposed optical communication systems will be used along with the other wired communication approaches for more flexibility.

# Bibliography

- [1] P. Graven, Y. Plan, L.J. Hansen, and S. Harvey. Implementing plug-and-play adcs to support operationally responsive space. In *Aerospace Conference, 2008 IEEE*, pages 1–14, March 2008.
- [2] D. Fronterhouse D. Lanza J. Lyke, S. Cannon. A plug-and-play system for spacecraft components based on the usb standard. In *Proceedings of the 19th Annual AIAA/USU Conference on Small Satellites*, pages 18–24, 2005.
- [3] S. Cannon et al J. Lyke, D. Fronterhouse. A plug-and-play system for spacecraft components based on the usb standard. In *AIAA 3rd Responsive Space Conference, Paper No. RS3-2005-5001, Los Angeles, CA,*, pages 25–28, 2005.
- [4] L. G. Jordan. Aerator S. A. McDermott. Aeroastros smartbus: A low cost modular approach enabling responsive space missions. In *Proceedings of the AIAA 3rd Responsive Space Conference*, pages 1–14, April 2005.
- [5] J.C. Lyke. Plug-and-play satellites. *Spectrum, IEEE*, 49(8):36–42, 2012.
- [6] A.T. Tai, S.N. Chau, and L. Alkalai. Cots-based fault tolerance in deep space: Qualitative and quantitative analyses of a bus network architecture. In *High-Assurance Systems Engineering, 1999. Proceedings. 4th IEEE International Symposium on*, pages 97–104, 1999.
- [7] G.D. Chandler, D.T. McClure, S.F. Hishmeh, J.E. Lumppp, J.B. Carter, B.K. Malphrus, D.M. Erb, W.C. Hutchison, G.R. Strickler, J.W. Cutler, and R.J. Twiggs. Development of an off-the-shelf bus for small satellites. In *Aerospace Conference, 2007 IEEE*, pages 1–16, 2007.
- [8] A. Addaim, A. Kherras, and B. Zantou. Design of store and forward data collection low-cost nanosatellite. In *Aerospace Conference, 2007 IEEE*, pages 1–10, 2007.
- [9] C. Temple. Avoiding the babbling-idiot failure in a time-triggered communication system. In *Fault-Tolerant Computing, 1998. Digest of Papers. Twenty-Eighth Annual International Symposium on*, pages 218–227, 1998.
- [10] F. Leens. An introduction to i2c and spi protocols. *Instrumentation Measurement Magazine, IEEE*, 12(1):8–13, 2009.
- [11] D. Johnson. Implementing serial bus interfaces using general purpose digital instrumentation. *Instrumentation Measurement Magazine, IEEE*, 13(4):8–13,

- 2010.
- [12] Dante Del Corso Muhammad Rizwan Mughal, Leonardo M. Reyneri. Implementation of inter and intra tile optical data communication for nanosatellites. pages 1–7. In: Proc. International Conference on Space Optical Systems and Applications (ICSOS) 2012, 11-3, Ajaccio, Corsica, France, 2012.
  - [13] Stefano Speretta, Danilo Roascio, Leonardo M. Reyneri, Claudio Sansoe, Maurizio Tranchero, Claudio Passerone, and Dante Del Corso. Optical-based {COTS} data communication bus for satellites. *Acta Astronautica*, 66:674 – 681, 2010.
  - [14] R.C. Meitzler, W. Schneider, and R. Conde. Ircomm: spacecraft free-space optical bus development. In *Aerospace Conference, 2005 IEEE*, pages 1583–1588, 2005.
  - [15] Leonardo M. Reyneri Dante Del Corso Muhammad Rizwan Mughal, Anwar Ali. Free space and glass fibre based infrared communication system on-board aramis satellite. pages 1–7. In: Proc. Advanced Infrared Technology and Applications (AITA), Torino, Italy, 2013, 2013.
  - [16] J.M. Benedetto and A. Oliver. High-speed data transmission for spaceborne applications. In *Radiation Effects Data Workshop, 2001 IEEE*, pages 67–71, 2001.
  - [17] et al. H. Guerrero. Optical wireless links for intra-satellite communications (owls). In *3rd Round Table on Micro/Nano-Technologies for Space. ESTEC-ESA*, pages 101–128, 2000.
  - [18] C. Barnes, O. Swift, A. Johnston, B. Rax, and K. LaBel. Radiation effects considerations for the application of photonics in space systems. In *Aerospace Conference, 1998 IEEE*, volume 2, pages 219–239 vol.2, 1998.
  - [19] A.H. Johnston. Radiation effects in light-emitting and laser diodes. *Nuclear Science, IEEE Transactions on*, 50(3):689–703, 2003.
  - [20] Jake A. Schaffner. The electronic system design, analysis, integration, and construction of the cal poly state university cp1 cubesat. In *16th AIAA/USU Conference on Small Satellites*, pages 1–12.
  - [21] Creedon M. Puig-Suari J. Nason, I. Cubesat design specifications document. In <http://ssdl.stanford.edu/cubesat>, pages 1–6, 2001.
  - [22] M. Vasile Komninou and E. Minisci. Optimal power harness routing for small-scale satellite. *in: IAC-11,C3,3,5,x10550*, pages 1–7, October 2011.
  - [23] M. Rizwan Mughal, Anwar Ali, and Leonardo M. Reyneri. Plug-and-play design approach to smart harness for modular small satellites. *Acta Astronautica*, 94(2):754 – 764, 2014.
  - [24] Leonardo Reyneri, Anwar Ali, Muhammad Rizwan Mughal, and Juan Carlos De Los Rios Arbelaez. Scalable plug and play tiles for nanosatellites. pages 1–7. in: 63rd International Astronautical Congress, 2012.

- 
- [25] Jaime Esper and Adaptive Modular. Modular, adaptive, reconfigurable systems:technology for sustainable, reliable, effective, and affordable space exploration. *Space Exploration, AIP Conf. Proc.* 746, 746(1033), 2005.
- [26] C. Y. Clark, K. B.; Baldwin. *The Power of Modularity*, volume 1:. Cambridge, Massachusetts: MIT Press 2000., 2000.
- [27] Peter M. Wegner; Phil Blower; Jason Wilkenfeld. *Standard buses modular buses, and plug-and-play buses; what is the difference and why does it matter?* , Report No A553345, August 2008.
- [28] Stefano Speretta; Leonardo M. Reyneri; Claudio Sansoe; Maurizio Tranchero; Claudio Passerone and Dante Del Corso. Modular architecture for satellites. *in: IAC-07-B4. 7. 09, 58th International Astronautical Congress*, pages 1–7, September 2007.
- [29] Leonardo M. Reyneri; Danilo Roascio; Claudio Passerone ; Stefano Iannone; Juan Carlos de los Rios; Giorgio Capovilla; Antonio Martinez-Alvarez and Jairo Alberto Hurtado. *Modularity and Reliability in Low Cost AOCSs and Advances in Spacecraft Systems and Orbit Determination*. 2012.
- [30] Chang-Da Wen and Issam Mudawar. Emissivity characteristics of polished aluminum alloy surfaces and assessment of multispectral radiation thermometry (mrt) emissivity models. *International Journal of Heat and Mass Transfer*, 48(7):1316 – 1329, 2005.
- [31] Anwar Ali; Leonardo M. Reyneri; Juan Carlos de los Rios; Haider Ali. Innovative power management tile for nanosatellites. *in: IAC-12,C3,4,3,x13998*, pages 1–5, October 2012.
- [32] L. M. Muhammad Rizwan Mughal; Anwar Ali; Haider Ali; Reyneri. A smart honeycomb tile for small satellites. pages 1–7. IEEE Aerospace conference, 2014.
- [33] L. M. ; Reyneri. *UML-based design methodology for designing university satellites*. 1st IAA Conference on University Satellite Missions and CubeSat Workshop, January 2011.
- [34] M.R. Mughal, L.M. Reyneri, and A. Ali. Uml based design methodology for serial data handling system of nanosatellites. In *Satellite Telecommunications (ESTEL), 2012 IEEE First AESS European Conference on*, pages 1–6, 2012.
- [35] M. Tranchero. Hw-sw design flow of a nano-satellite using uml and codesimulink co-design environment. In *Electrotechnical Conference, 2006. MELECON 2006. IEEE Mediterranean*, pages 85–88, 2006.
- [36] L.M. Reyneri. An object oriented codesign flow for low-cost hw/sw/mixed-signal systems based on uml. In *Electrotechnical Conference, 2006. MELECON 2006. IEEE Mediterranean*, pages 80–84, 2006.
- [37] NXP Semiconductors. *I2C-bus specification and user manual*. 2007.
- [38] Texas Instruments. *MSP430 User Guide*. Texas Instruments, Tech, dec 2010.

- [39] M.N. Lovellette, K.S. Wood, D. L. Wood, J.H. Beall, P.P. Shirvani, N. Oh, and E.J. McCluskey. Strategies for fault-tolerant, space-based computing: Lessons learned from the argos testbed. In *Aerospace Conference Proceedings, 2002. IEEE*, volume 5, pages 5–2109–5–2119 vol.5, 2002.
- [40] M. Rebaudengo, M.S. Reorda, and M. Violante. A new software-based technique for low-cost fault-tolerant application. In *Reliability and Maintainability Symposium, 2003. Annual*, pages 25–28, 2003.
- [41] N. Oh, P.P. Shirvani, and E.J. McCluskey. Error detection by duplicated instructions in super-scalar processors. *Reliability, IEEE Transactions on*, 51(1):63–75, 2002.
- [42] N. Oh, P.P. Shirvani, and E.J. McCluskey. Control-flow checking by software signatures. *Reliability, IEEE Transactions on*, 51(1):111–122, 2002.
- [43] J. Chang, G.A. Reis, and D.I. August. Automatic instruction-level software-only recovery. In *Dependable Systems and Networks, 2006. DSN 2006. International Conference on*, pages 83–92, 2006.
- [44] A. Martinez-Alvarez, S. Cuenca-Asensi, F. Restrepo-Calle, F.R.P. Pinto, H. Guzman-Miranda, and M.A. Aguirre. Compiler-directed soft error mitigation for embedded systems. *Dependable and Secure Computing, IEEE Transactions on*, 9(2):159–172, 2012.
- [45] D. Del Corso, C. Passerone, L. Reyneri, C. Sansoe, S. Speretta, and M. Tranchero. Design of a university nano-satellite: the picpot case. *Aerospace and Electronic Systems, IEEE Transactions on*, 47(3):1985–2007, 2011.
- [46] M.R. Mughal, D. Roascio, and L.M. Reyneri. Data communication bus interface for modular architecture of small satellites. In *Satellite Telecommunications (ESTEL), 2012 IEEE First AESS European Conference on*, pages 1–6, 2012.
- [47] Anwar Ali. *Power management, attitude determination and control systems of small satellites*. PhD thesis, Politecnico di Torino, Italy, 2014.
- [48] Haider Ali. *Telecommunication subsystem design for nanosatellites*. PhD thesis, Politecnico di Torino, Italy, 2014.
- [49] T. Instruments. Low-power SoC (system-on-chip) with MCU. feb 2013.
- [50] Leonardo M. Reyneri, Stefano Speretta, and Dante Del Corso. A redundant power bus for distributed power management for a modular satellite. in: *IAC-09.C3. 2.8*, 2009.
- [51] G. Bianchi, G.S. Aglietti, and G. Richardson. Development of efficient and cost-effective spacecraft structures based on honeycomb panel assemblies. In *Aerospace Conference, 2010 IEEE*, pages 1–10, 2010.
- [52] Light sandwich constructions, url:<http://www.ctihuatai.com/products-sandwichen.htm>. Technical report, 2008.
- [53] M. Rizwan Mughal, Anwar Ali, and Leonardo M. Reyneri. Plug-and-play design approach to smart harness for modular small satellites. *Acta Astronautica*,

- 94(2):754 – 764, 2014.
- [54] Anwar Ali, M.Rizwan Mughal, Haider Ali, and Leonardo Reyneri. Innovative power management, attitude determination and control tile for cubesat standard nanosatellites. *Acta Astronautica*, 96(0):116 – 127, 2014.
- [55] Intersil,isl3281e datasheet. Online: <http://www.intersil.com/data/fn/fn6543.pdf>.
- [56] Fairchild, bat54 schottky diodes datasheet. Online: <http://www.fairchildsemi.com/ds/BA/BAT54C.pdf>.
- [57] Pulse transformers,transformer equivalent circuit, 2008. Rhombus Industries Inc.
- [58] Philip C. Todd. *Snubber circuits: Theory, Design and Applications* . TI, 1993.
- [59] Danilo Roascio. Sottosistema di comunicazione tollerante ai guasti per satellite. Master's thesis, Dept. Elect. Eng., Politecnico Di Torino., Italy, 2008.
- [60] M.R. Mughal, L.M. Reyneri, J.C. de los Rios, and H. Ali. Development of optical wireless transceiver for data communication in small satellites. In *Satellite Telecommunications (ESTEL), 2012 IEEE First AESS European Conference on*, pages 1–6, 2012.
- [61] M.R. Mughal, L.M. Reyneri, J.C. de los Rios, and H. Ali. Development of optical wireless transceiver for data communication in small satellites. In *Satellite Telecommunications (ESTEL), 2012 IEEE First AESS European Conference on*, pages 1–6, 2012.
- [62] A. Paraskevopoulos, J. Vucic, S.-H. Voss, and K.-D. Langer. Optical free-space communication systems in the mbps to gbps range, suitable for industrial applications. In *Optomechatronic Technologies, 2009. ISOT 2009. International Symposium on*, pages 377–382, 2009.
- [63] Vishay semiconductors,tshg8400,bpv10nf datasheets. Vishay semiconductors.
- [64] Osram opto semiconductors:, sfh4655,sfh4501 datasheets. OSRAM opto semiconductors.
- [65] Advanced photonics; pdb-c142f, pdb-c172smf datasheets. Advanced photonics.
- [66] A.-T. Avestruz, J.I. Rodriguez, R. Hinman, G. Livshin, E.C. Lupton, and S.B. Leeb. Stability considerations and performance of wide dynamic range, ambient light active rejection circuits in photodiode receivers. In *American Control Conference, 2004. Proceedings of the 2004*, volume 1, pages 367–373 vol.1, 2004.
- [67] Cc2510fx/cc2511fx preliminary data sheet (rev. 1.2) swrs055a. TI.
- [68] Simpliciti sample application user's guide. TI, 2009.
- [69] Fang Shanshan. Protocol Implementation Using TI's SimpliciTI Platform. Master's thesis, VAASAN AMMATTIKORKEAKOULU UNIVERSITY OF APPLIED SCIENCES, 2011.
- [70] Modifying radio register parameters for simpliciti applications. TI.
- [71] Application report, wireless sensor monitor using the ez430-rf2500,slaa378d. TI, 2011.
- [72] Simpliciti application programming interface. TI, 2009.

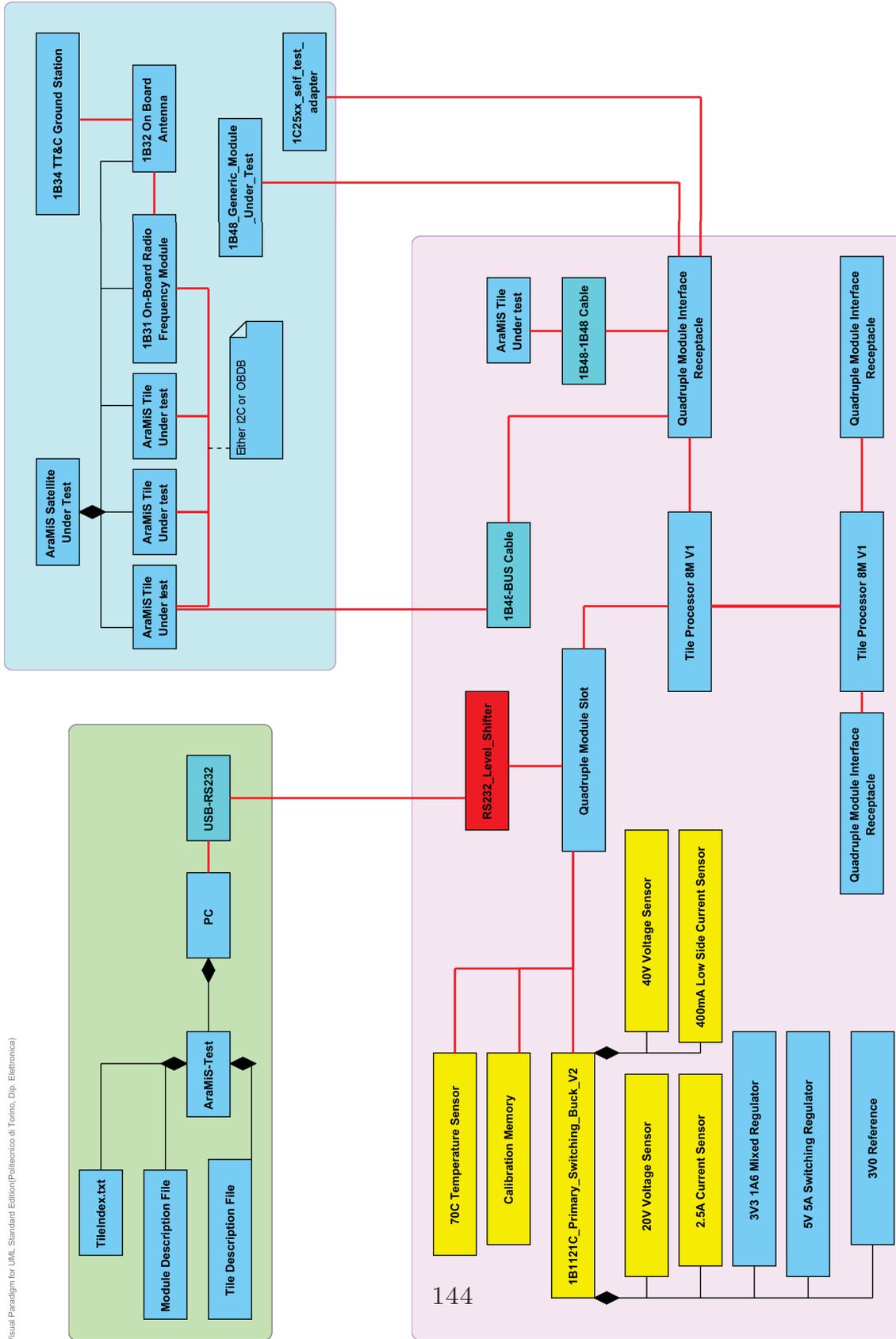


# Appendices



## Appendix A

# 1B81 Power Management Tile Honeycomb Schematics



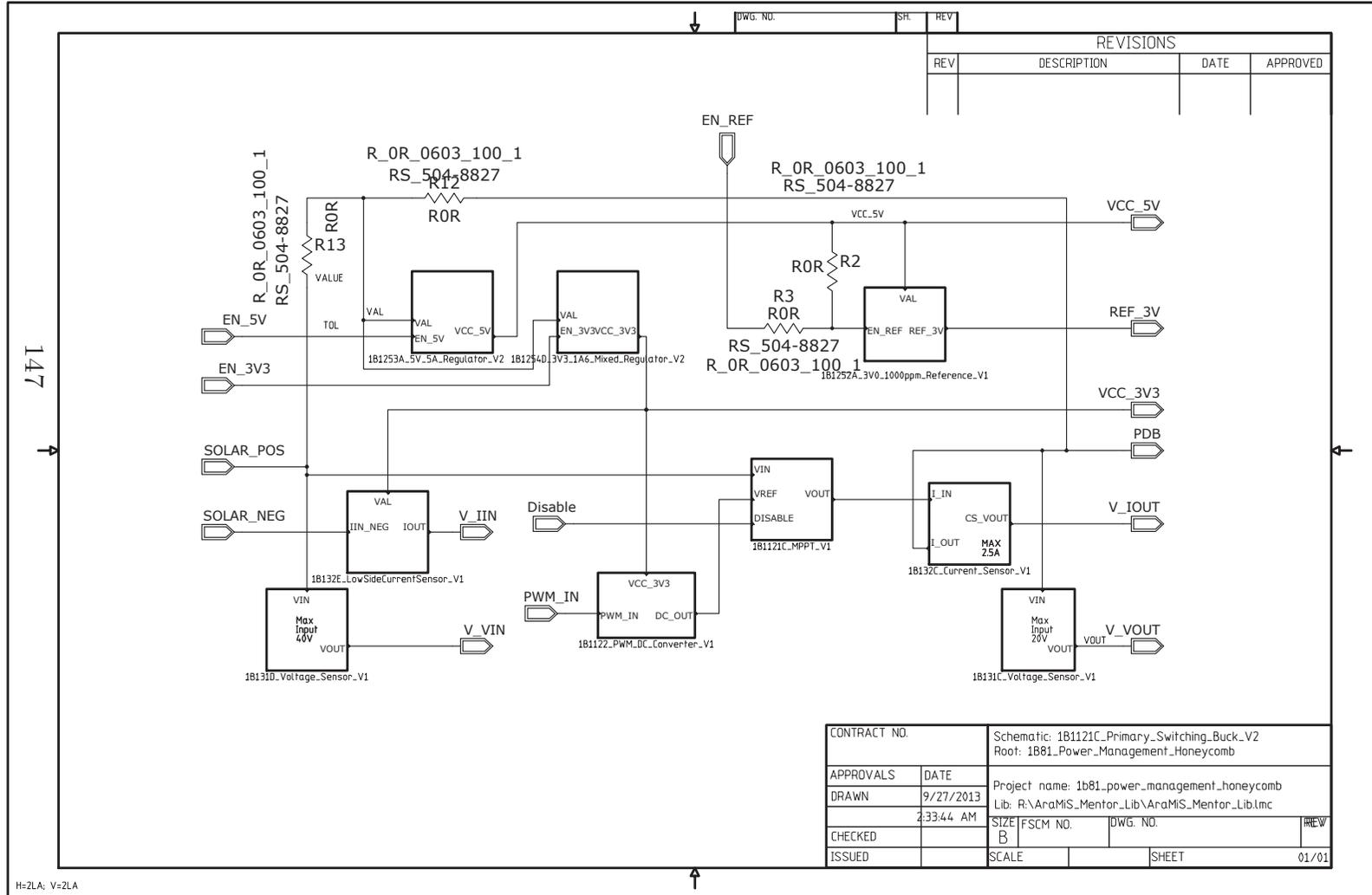
Visual Paradigm for UML, Standard Edition (Politecnico di Torino, Dip. Elettronica)

Figure A.1. Functional test class diagram





Figure A.4. Power Converters Schematics



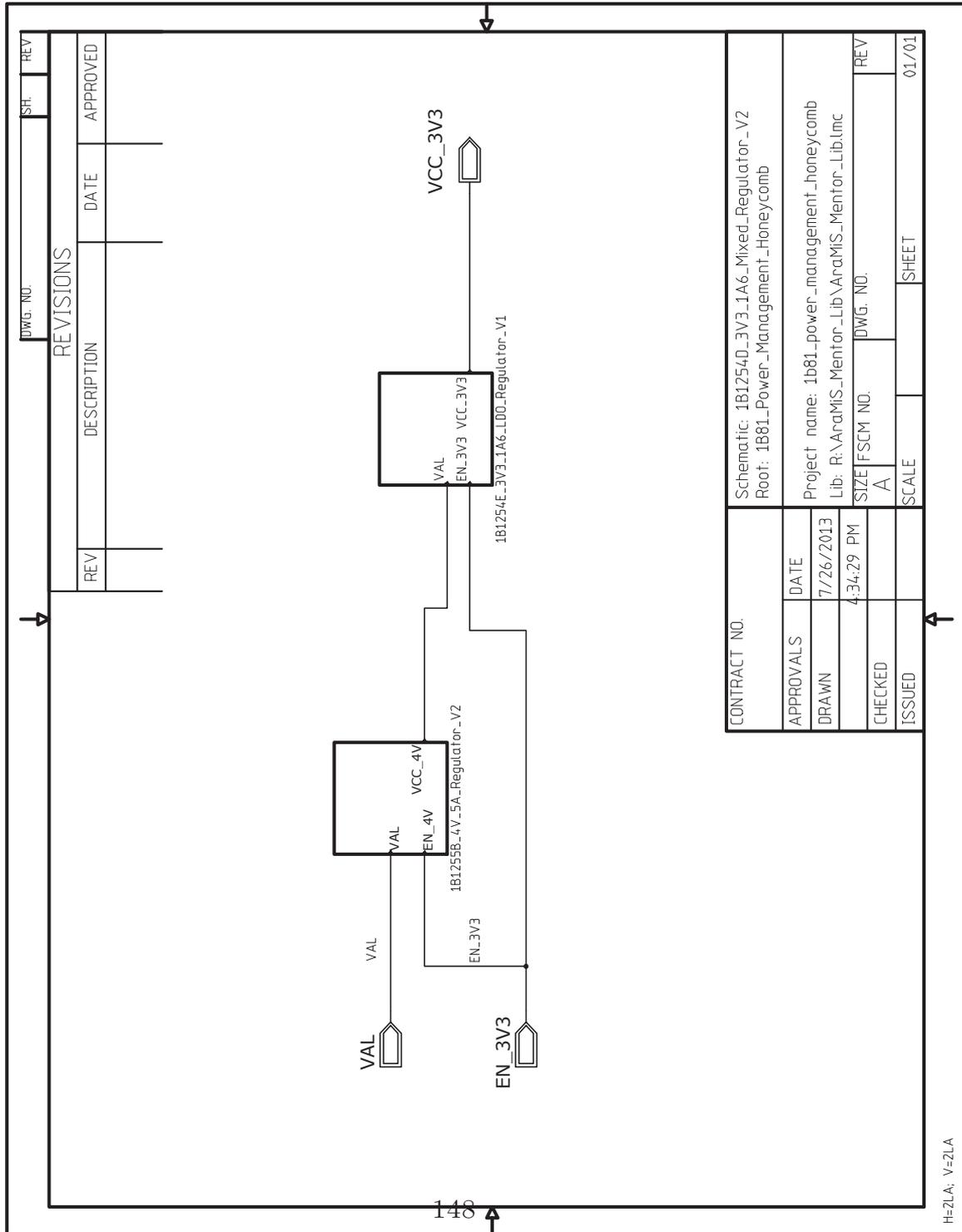


Figure A.5. Mixed Regulator

Figure A.6. 3.3V Linear Regulator

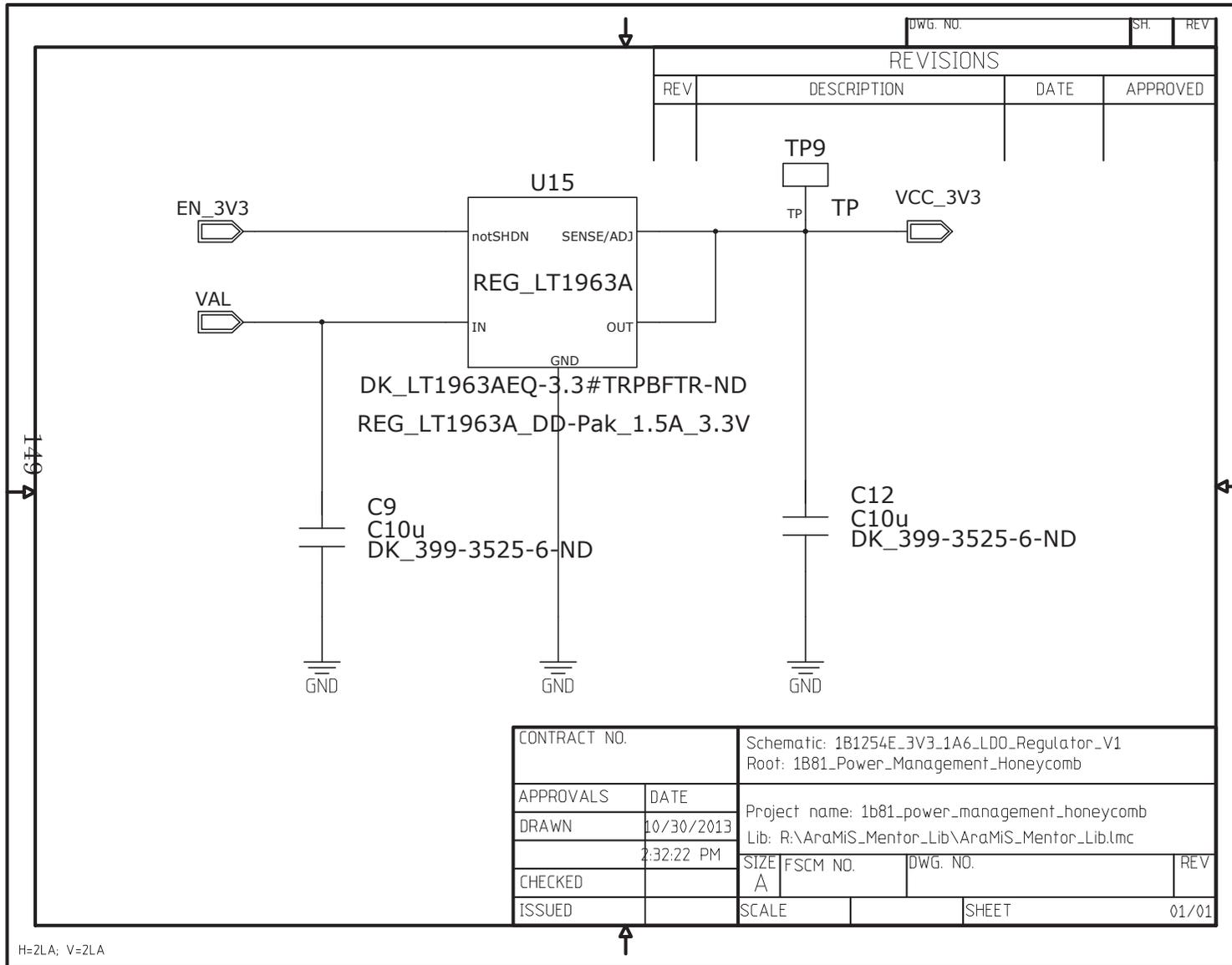
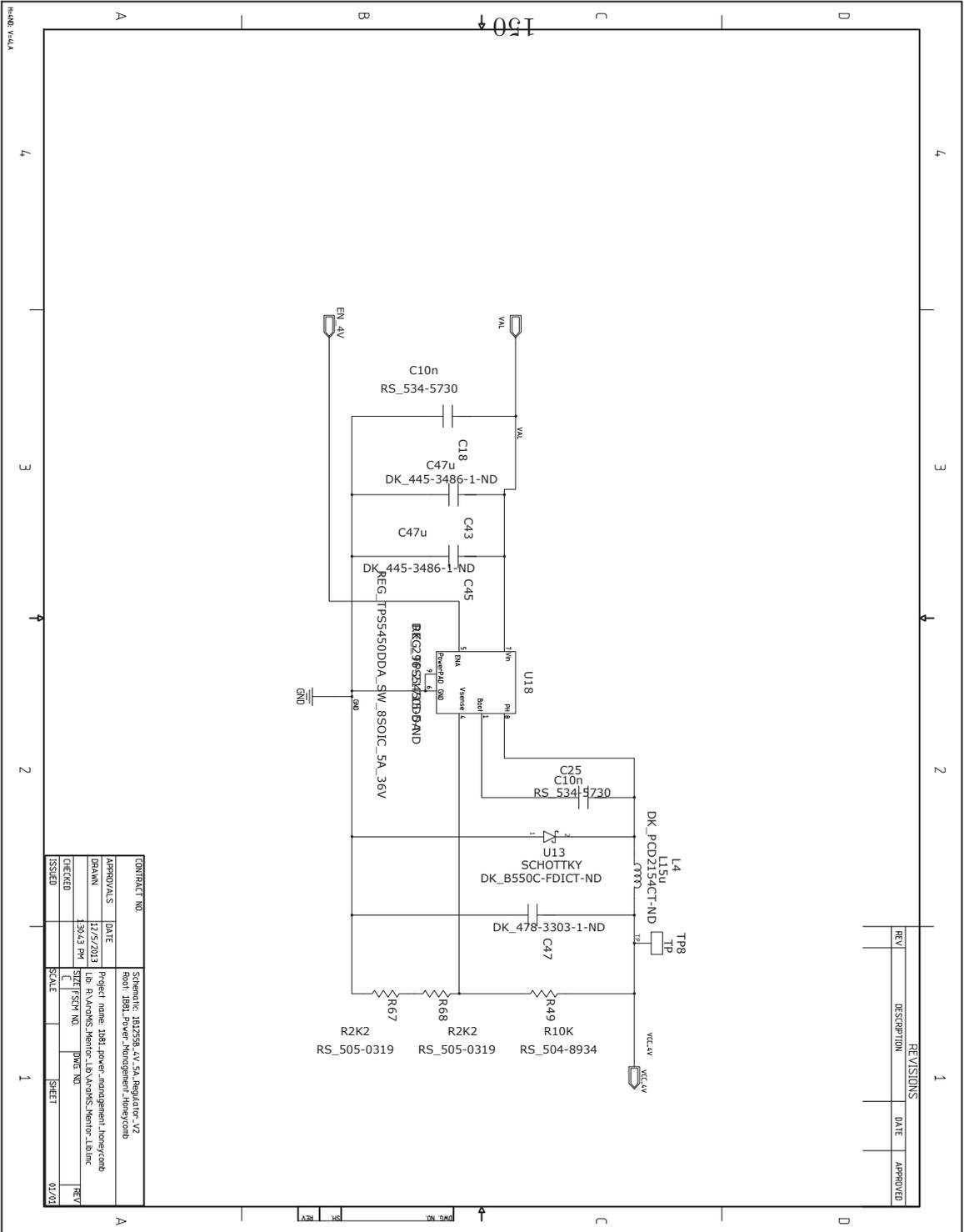


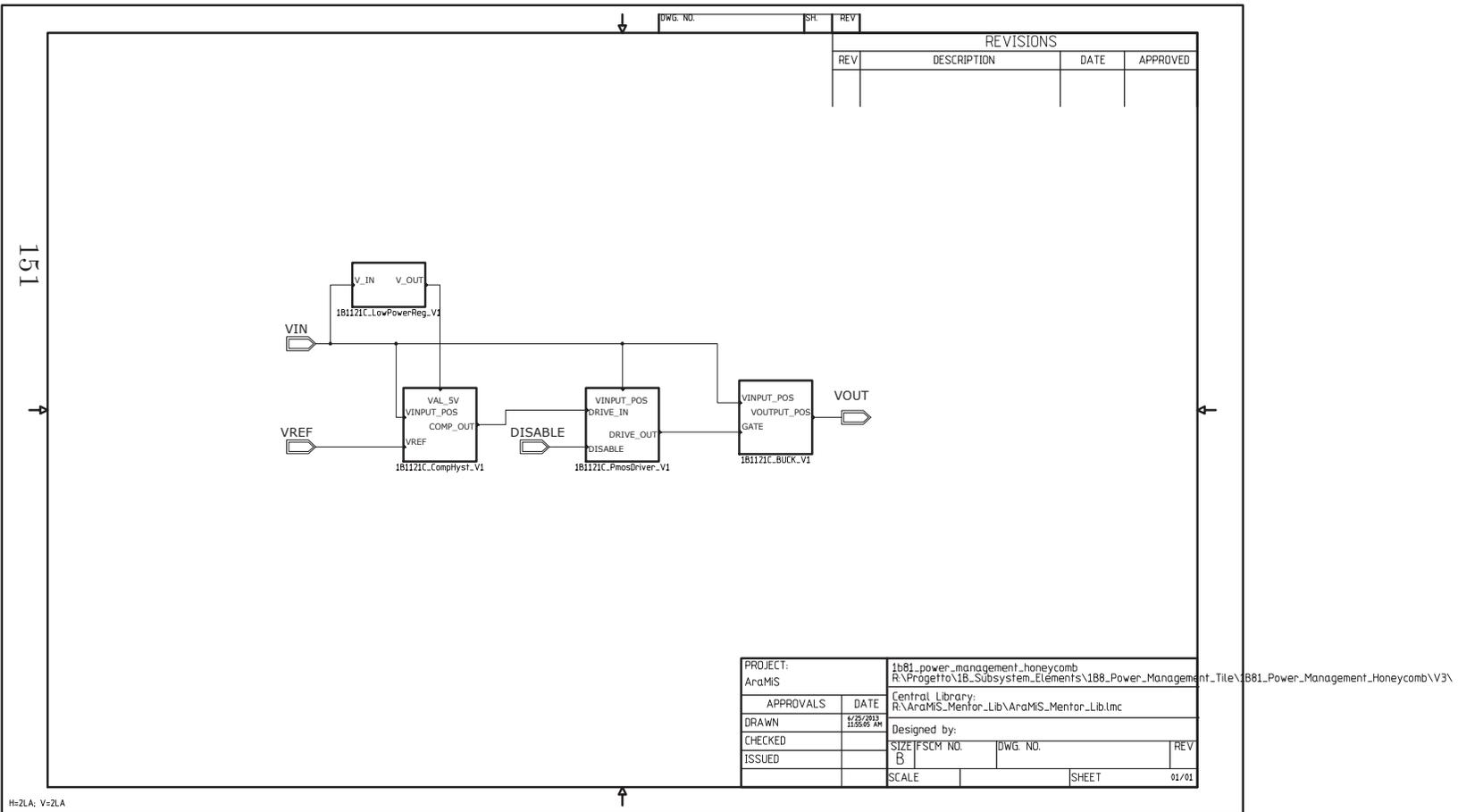
Figure A.7. Switching Regulator

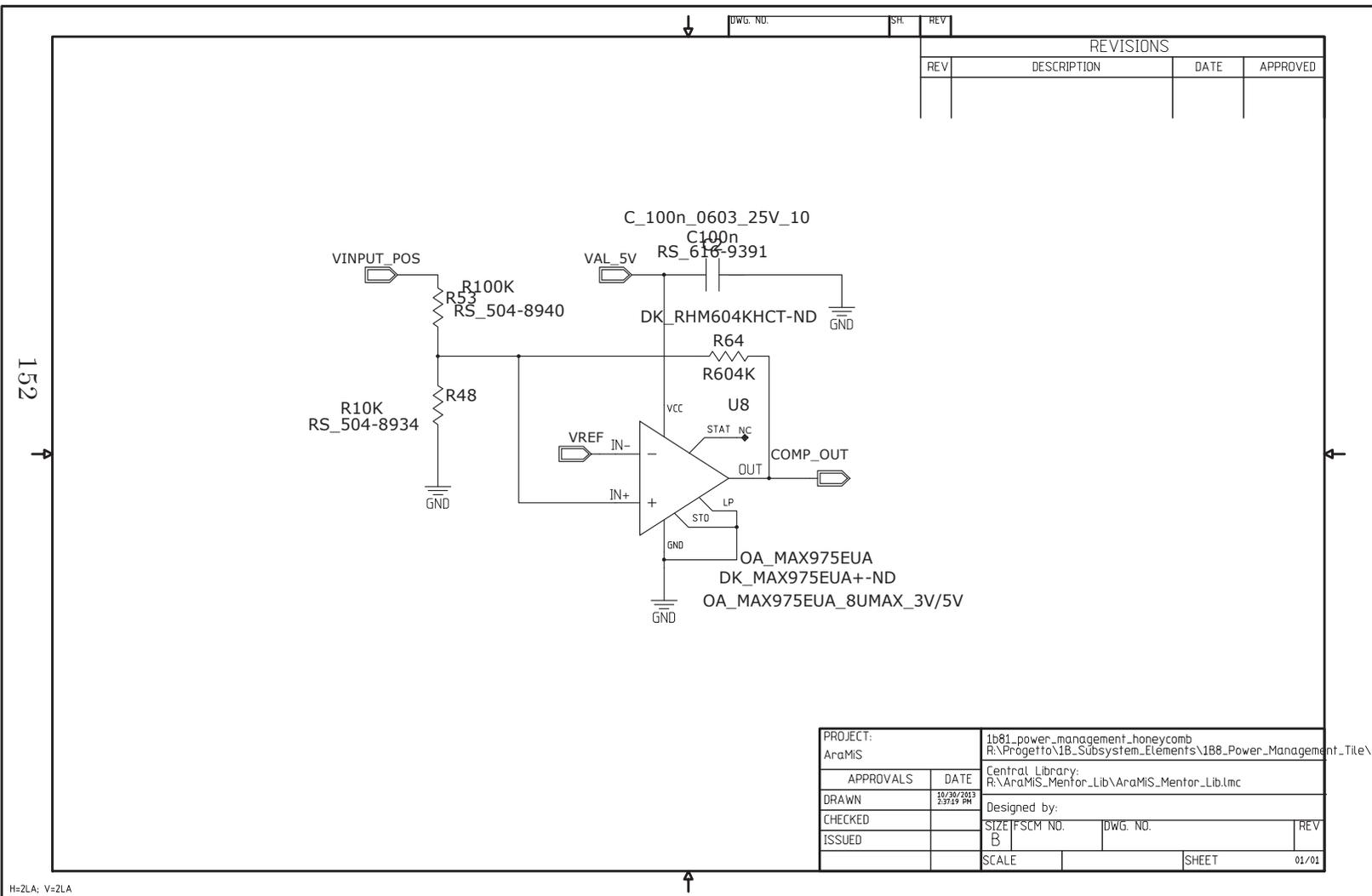


CONTRACT NO	DATE	PROJECT NAME	DWG NO	SHEET
181259-AV-5A_Regulator_V2	12/25/2013	1811_power_management_honeycomb		01/01
APPROVALS		LIB: R:\Archives_Mentor-LIB\Archives_Mentor-LIB.dwg		
DRAWN	30/03/09			
CHECKED				
ISSUED				

REV	DESCRIPTION	DATE	APPROVED

Figure A.8. Maximum power point tracker block





REVISIONS			
REV	DESCRIPTION	DATE	APPROVED

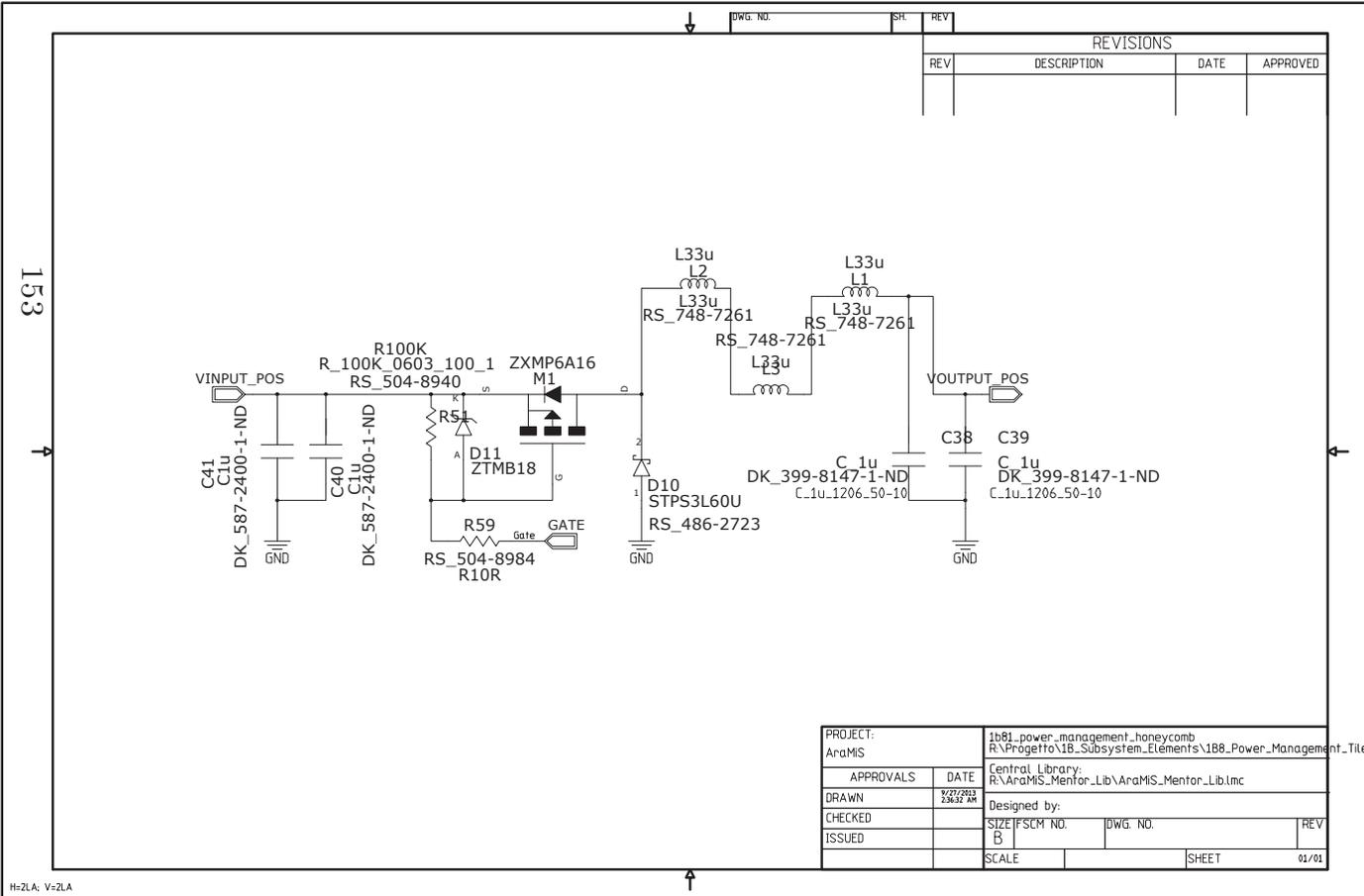
PROJECT:		1b81_power_management_honeycomb	
AraMIS		R:\Projetto\1B_Subsystem_Elements\1B8_Power_Management_Tile\1B81_Pow	
APPROVALS	DATE	Central Library: R:\AraMIS_Mentor_Lib\AraMIS_Mentor_Lib.lmc	
DRAWN	10/26/2013 2:37:19 PM	Designed by:	
CHECKED		SIZE/FSCM NO.	DWG. NO.
ISSUED		B	
		SCALE	SHEET 01/01

Figure A.9. Comparator Hysteresis

H=2LA; V=ZLA

152

Figure A.10. Buck Converter



REVISIONS			
REV	DESCRIPTION	DATE	APPROVED

PROJECT:		1b81_power_management_honeycomb	
AraMIS		R:\Projeto\18_Subsystem_Elements\1B8_Power_Management_Tile\B81_Power_Management_Honeycomb\13\	
APPROVALS	DATE	Central Library: R:\AraMIS_Mentor_Lib\AraMIS_Mentor_Lib.lmc	
DRAWN	9/27/2013 2:36:32 AM	Designed by:	
CHECKED		SIZE/FSCM NO.	DWG. NO.
ISSUED		B	
		SCALE	SHEET 01/01



## Appendix B

### 1411 On-Board Data Bus Module Schematics

B – 1411 On-Board Data Bus Module Schematics

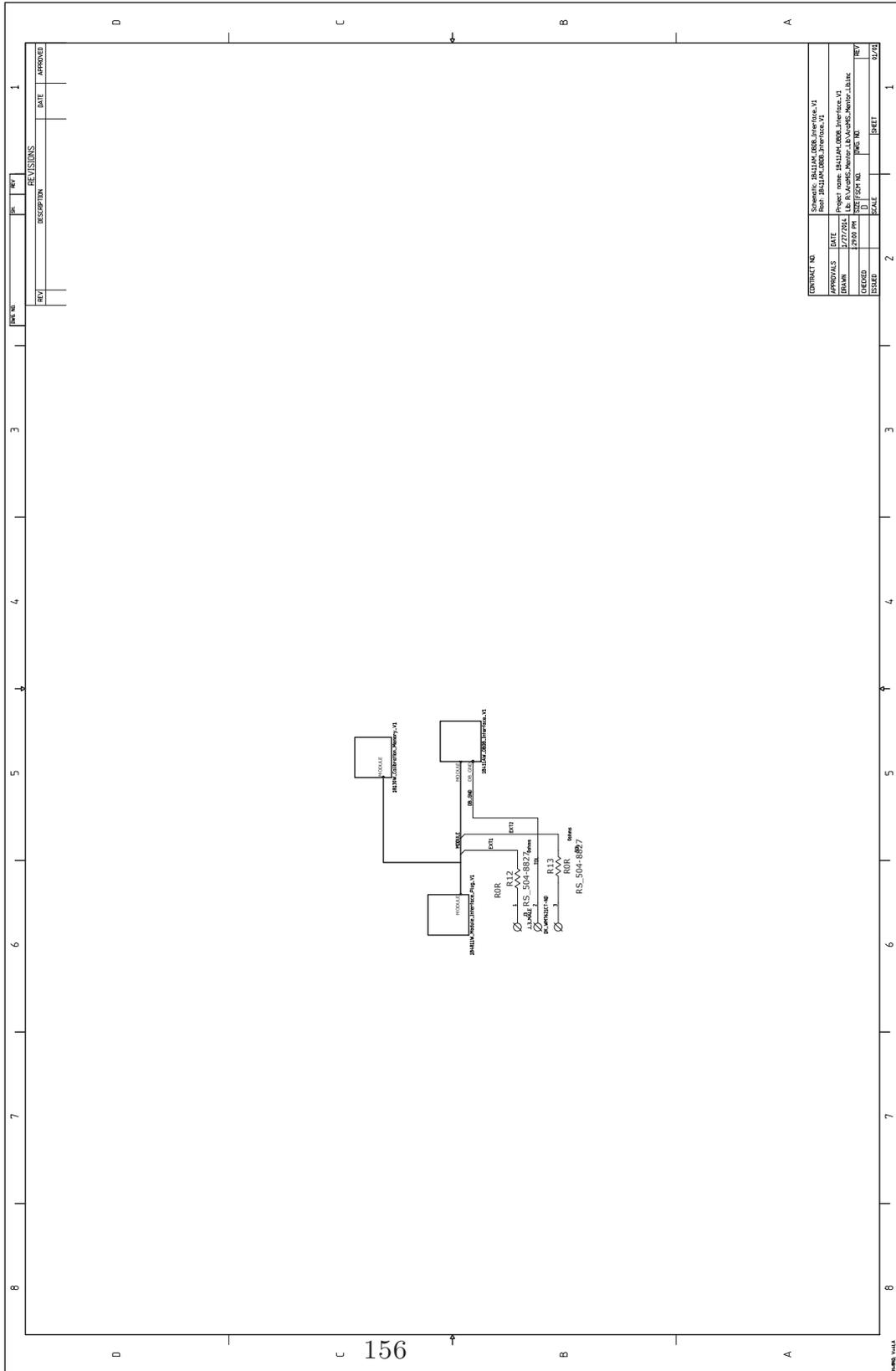
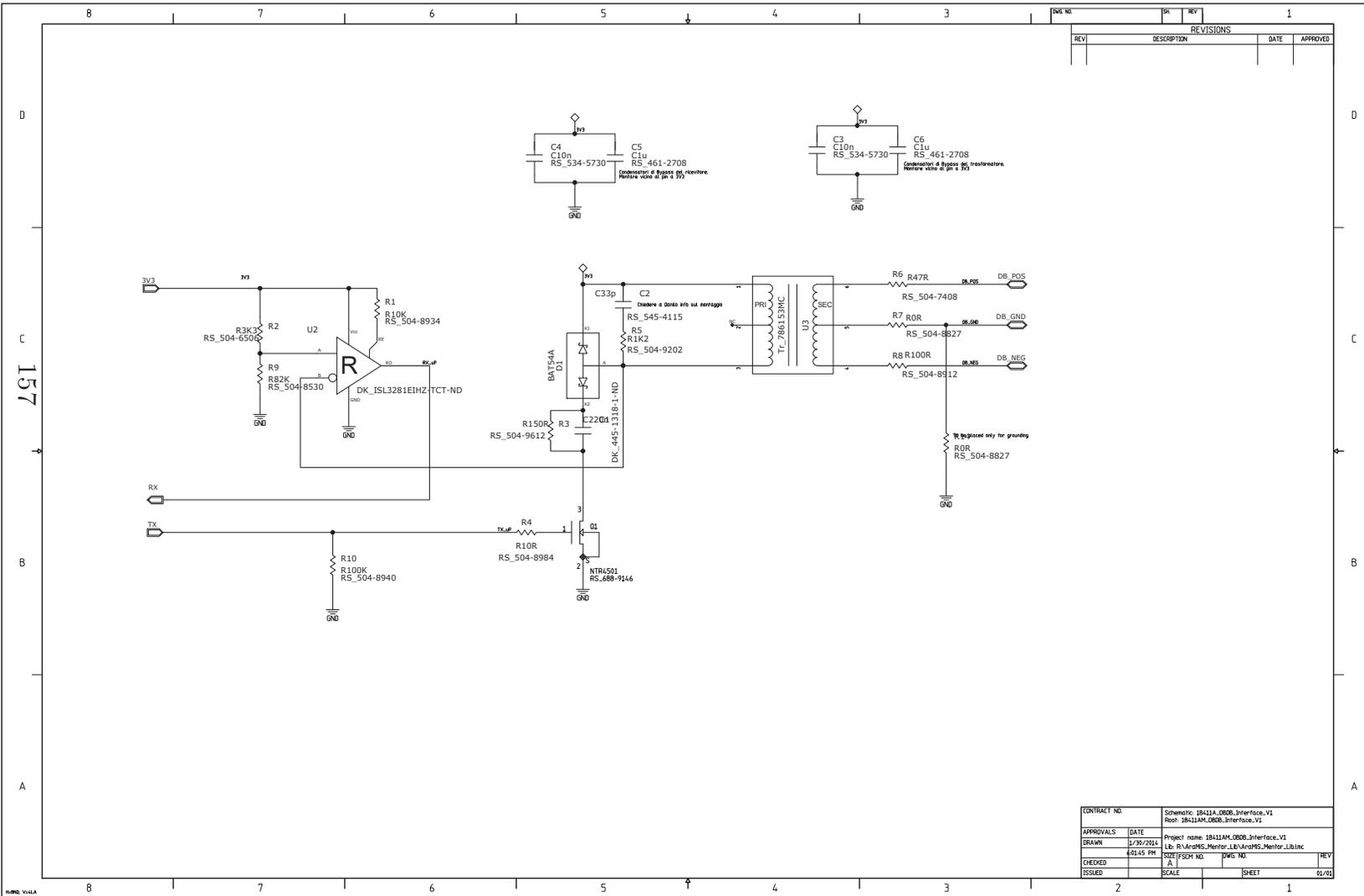


Figure B.1. OBDB Module

Figure B.2. OBDB Module Schematics





# Appendix C

## List of Acronymes

<b>AraMiS</b>	Modular Architecture of Small Satellites
<b>COTS</b>	Components off the shelf
<b>API</b>	Application programming interface
<b>OWLS</b>	Optical Wireless Links for Intra-Spacecraft
<b>CPU</b>	Central Processing Unit
<b>PCB</b>	Printed Circuit Board
<b>LEO</b>	Low Earth Orbit
<b>EPS</b>	Electric Power Supply
<b>ADCS</b>	Attitude Determination and Control Subsystem
<b>UART</b>	Universal Asynchronous Receive Transmit
<b>IrDA</b>	Infra-red Data Association
<b>SPI</b>	Serial Peripheral Interface
<b>I2C</b>	Inter Integrated Circuit
<b>I/O</b>	Input/Output
<b>UML</b>	Unified Modelling Language
<b>PWM</b>	Pulse Width Modulation
<b>SOMI</b>	Slave Out Master In

<b>SIMO</b>	Slave In Master Out
<b>CLK</b>	Clock
<b>SDA</b>	Serial Data
<b>SCL</b>	Serial Clock
<b>LPM</b>	Low Power Mode
<b>OBC</b>	On-board Computer
<b>PDB</b>	Power Distribution Bus
<b>RF</b>	Radio Frequency
<b>HW</b>	Hardware
<b>SW</b>	Software
<b>ADC</b>	Analog to Digital Converter
<b>ACLK</b>	Auxiliary Clock
<b>MCLK</b>	Main Clock
<b>SMCLK</b>	Sub Main Clock
<b>TI</b>	Texas Instruments
<b>SEU</b>	Single Event Upset
<b>SEE</b>	Single Event Effects
<b>Hdata</b>	Hardened Data
<b>TMR</b>	Triple Modular Redundancy
<b>MPPT</b>	Maximum Power Point Tracker
<b>MOS</b>	Metal Oxide Semiconductor
<b>KCL</b>	Kirchoff,s Current Law
<b>OBDB</b>	On-Board Data Bus
<b>LVDS</b>	Low Voltage Differential Signalling
<b>TX</b>	Transmit

---

<b>RX</b>	Receiver
<b>RZI</b>	Return to Zero Inverted
<b>LED</b>	Light Emitting Diode
<b>LoS</b>	Line of Sight
<b>TIA</b>	Trans-Impedance Amplifier
<b>GBWP</b>	Gain Bandwidth Product
<b>TID</b>	Total Ionization Dose
<b>DD</b>	Displacement Damage
<b>TIR</b>	Total Internal Reflection
<b>RSSI</b>	Receive Signal Strength Indicator
<b>BER</b>	Bit Error Rate
<b>SoC</b>	System on Chip
<b>RAM</b>	Random Access Memory
<b>OWLS</b>	Optical Wireless Links for intra-Spacecraft communications