

POLITECNICO DI TORINO

SCUOLA DI DOTTORATO

Dottorato in Elettronica e Comunicazioni - XXVI ciclo

Tesi di Dottorato

Telecommunication Subsystem Design for Small Satellite



Haider Ali

Tutore
Prof. Claudio Sansoé

Coordinatore del corso di dottorato
Prof. Ivo Montrosset

Marzo 2014

Acknowledgements

“In the name of Allah, the most Gracious and the Merciful”; who granted me the ability and courage to complete my doctorate thesis.

First and foremost I extend my gratitude to my Tutor Professor Claudio Sansoé and AraMiS Project Coordinator Professor Leonardo M. Reyneri, for giving me an opportunity to work in the AraMiS group. Your guidance and supervision have helped me a great deal during these three years. I appreciate your encouragement in research and for your patience during my stay here.

I would also like to thank the whole AraMiS team at Politecnico di Torino for their help during these three years, in particular Prof. Dante Del Corso. I also like to thank Danilo Roascio and Juan Carlos de Los Rios for their help in learning different hardware and software tool.

I want to appreciate their continuous support and help of my colleagues Anwar Ali and Muhammad Rizwan Mughal. These three years would never have been so pleasant and enjoyable without them.

I am also thankful to “Higher Education Commission”, Pakistan (HEC) for providing me this opportunity to undertake Masters and PhD studies here at Politecnico di Torino, Italy.

Last but not least, I am very grateful to my family for their prayers and support during this period. In the end I want to dedicate my thesis work to my late father who believed in me and has inspired me all my life.

Summary

Satellite design has always been considered as an extremely expensive and high risk business which not only requires vast knowledge and expertise but also extensive budget. Primarily, this concept was based on initial development and launching cost. Secondly, it was impossible to repair and substitute parts in space (this was true up to 1993: the first Hubble Space Telescope servicing mission), which made the design more tough because it required advanced fault tolerance solutions and extreme reliability. With the passage of time academic entities and small companies have also entered this market. Low cost design techniques have played an important role in the aerospace market growth over the recent years, but they can still play an important role in future developments. At present, several private companies are also providing an affordable launch services which lowers the accumulative cost. Many universities and Small and Medium Enterprises (SMEs) worldwide are trying to further reduce the satellite costs. One recent example of these efforts is the CubeSat concept: a really small satellite, built using commercial components.

This thesis work also proceeds in the same direction: the novel AraMiS (Italian acronym for Modular Architecture for Satellites) architecture and its different tiles (main subsystem) will be described. The goal of AraMiS is to progress beyond the concept of CubeSats and create a true modular architecture. The main idea of the AraMiS is modularity at mechanical, electronic and testing levels. These modules can then be assembled together to get desired requirements for the targeted mission, which allows for an effective cost sharing between multiple missions.

This thesis deals with the design and development of telecommunication subsystems for AraMiS Project and in particular for AraMiS-C1 satellite on a single CubeSat standard module called tile. The implementation of S-band transceiver over the half of telecommunication tile is not a trivial task. Several techniques were employed to make possible such a reduction in terms of size, weight and power consumption while still achieving desirable performance for communication link. COTS components have been used for 1B9_CubeTCT sub modules implementation. COTS particularly for RF Front End design were selected on the basis of performance in harsh LEO environment, power losses, dimension and space occupied onboard 1B9_CubeTCT. In order to cope with such anomalies on the 1B9_CubeTCT Subsystem; different housekeeping sensors have been employed at various point of the tile.

This thesis work also elaborates on the S-band antennas design for both versions of AraMiS satellites with uses innovative technique to enhance performance while keep the size, weight and cost within acceptable margins.

The first couple of chapter presents an introduction to AraMiS project and AraMiS-C1 satellite. In chapter 3 there is a discussion of different satellite design flow configurations.

The chapter 4 comprehensively discusses the 1B9_CubeTCT, which is the CubeSat standard telecommunication tile developed for AraMiS-C1 and other CubeSat standard nano-satellites. It consists of S-band and UHF OBRF modules which provide radio communication link between satellite and the earth. It also gives an indepth explanation on the design and development of each sub-module onboard 1B9_CubeTCT that includes CubeTCT S-band transceiver, RF front end, Housekeeping Sensors, Tile Regulators, anti-latchup protection circuit and RF matching network.

Chapter 5 deals with the S-band antenna design, fabrication and testing for conventional AraMiS architecture and AraMiS C-1 satellite. It explains in detail, the design, implementation and testing of single patch antenna and AraMiS patch array that are used for AraMiS C-1 and Conventional AraMiS satellites respectively.

The chapter 6 provides link budget estimation for different scenarios, ranging for worst to the best possible case. A brief description on the Polito ground station and its key attributes is also provided. The Link budget estimates for other ground stations (GENSO members) are also performed and feasibility of developed 1B9CubeTCT hardware is verified.

Chapter 7 introduces the AraMiS protocol, developed in compliances with GENSO project which aims for providing an extended communication link for a satellite (remotely via internet), by using a GENSO member ground station around the World. A new frame format is defined which makes AraMiS satellite compatible with GENSO. Later in the chapter are presented different possible scenarios of AraMiS Protocol mechanism during normal operation case and also in case of packet loss (in both uplink and downlink communication.)

Contents

Acknowledgements	i
Summary	iii
Contents	v
Acronyms & Abbreviations	ix
Chapter 1 Introduction	1
1.1 Problem Statement.....	2
1.2 Proposed Solutions	2
1.3 Thesis Organization.....	2
Chapter 2 Introduction to Small Satellites	5
2.1 PiCPoT	6
2.2 AraMiS Project.....	7
2.2.1 AraMiS Satellite Subsystems	7
2.2.2 Mechanical subsystem.....	8
2.2.3 Power management subsystem.....	9
2.2.4 AraMiS Telecommunications Subsystem	9
2.2.5 Tile Computer subsystem	10
2.2.6 Attitude determination and control subsystem	10
2.2.7 Payload	11
2.3 AraMiS-C1	11
2.3.1 ISIS - 1-Unit CubeSat Structure	12
2.3.2 1B8_CubePMT Module	13
2.3.3 1B9_CubeTCT	14
2.3.4 1B31A Tile Radio Frequency Module 437MHz	15
2.3.5 ISIS - Deployable Antenna System for CubeSats	15
2.3.6 1B31B Tile S-band Radio Frequency Module	16
2.3.7 1B3211_Single Patch Rodgers	16
2.3.8 Payload	16
2.3.9 Battery	17
2.3.10 Tile Data Handling (OBDH) and Software	17
2.3.11 On-board Computer.....	18
Chapter 3 Satellite Design Flow	21
3.1 Introduction	21
3.2 Design flow Configuration.....	21

3.2.1	Physical Module Based Configuration	21
3.2.2	Satellite on Demand Configuration	22
3.2.3	Reusable Design Configuration	22
Chapter 4	Telecommunication Tile: 1B9_CubeTCT.....	23
4.1	Overview.....	23
4.2	1B9_CubeTCT Design Overview.....	26
4.3	CubeTCT Transceiver (1B31B1_OBRF_CC2510).....	27
4.3.1	CubeTCT Regulators & Load Switch.....	30
4.3.2	Housekeeping Sensors	32
4.3.3	1B133B_Temperature_Sensor.....	34
4.3.4	1B12_Anti-Latchup	35
4.3.5	CubeTCT RF Front End	36
Chapter 5	AraMiS Antenna: Design, Fabrication & Testing	43
5.1	Introduction.....	43
5.2	Single Patch Antenna.....	44
5.2.1	Design Specifications/Material Selection	44
5.2.2	Design Calculations	46
5.2.3	Simulation Analyses & Measurements	46
5.3	Patch Antenna Array.....	49
5.3.1	Theoretical Overview	50
5.3.2	Array Structure Design	50
5.3.3	Array Structure Concept	52
5.3.4	Simulation Analyses & Measurement	57
5.4	Conclusion	61
Chapter 6	S-band Link Budget Estimation for AraMiS Satellites	63
6.1	Polito Ground Station	63
6.1.1	Introduction.....	63
6.1.2	Link Budget Calculation.....	64
Chapter 7	AraMiS Telecommunication Protocol.....	71
7.1	Project GENSO.....	71
7.1.1	GENSO Architecture & Operation.....	72
7.2	AraMiS Protocol Architecture	75
7.2.1	On-board Radio Frequency.....	76
7.2.2	Telemetry/Command Processor:.....	76

7.2.3	Housekeeping Processor:.....	76
7.3	Protocol Structure.....	76
7.3.1	LONG execution time commands (LONG):	76
7.3.2	SHORT execution time commands (SHORT):	77
7.4	Packet Format Definition	77
7.4.1	Packages in uplink.....	78
7.4.2	Packages in downlink.....	80
7.5	Defined Functions	83
7.6	Protocol Operation.....	84
7.6.1	SHORT command	84
7.6.2	SHORT command with loss in the uplink.....	85
7.6.3	SHORT command with loss in the downlink.....	86
7.6.4	LONG Subsequent request with command	87
7.6.5	LONG command with ACK loss in the downlink.....	88
7.6.6	Download with Fragmentation	89
7.6.7	Combined LONG/SHORT Command Operations	90
7.7	Performance Analysis.....	91
7.7.1	RDT commands SHORT.....	91
	Chapter 8 Conclusions.....	97
	Bibliography	99

Acronyms & Abbreviations

SMEs	Small Medium Enterprises
EPS	Electric Power Supply
ADCS	Attitude Determination & Control Subsystems
COTS	Commercial-Off-The-Shelf
CubePMT	CubeSat Power Management Panel
CubeTCT	CubeSat Telecommunication Tile
DET	Department of Electronics & Telecommunication
LEO	Low Earth Orbit
POD	Picosatellite Orbital Deployer
OBC	On-Board Computer
PMTs	Power Management Tiles
PDB	Power Distribution Bus
OBDH	On-board Data Handling
I2C	Inter Integrated Circuit
ISIS	Innovative Solution In Space, Inc
ESR	Equivalent Series Resistance
JTAG	Joint Test Action Group
ESL	Equivalent Series Inductance
ADS	Attitude Determination System
ACS	Attitude Control System
TxRx	Transceiver
Tx	Transmitter
Rx	Receiver
SPI	Serial Peripheral Interface
PCB	Printed Circuit Board
PA	Power Amplifier
LNA	Low Noise Amplifier
UHF	Ultra High Frequency
SEU	Single Event Upset
SEL	Single Event Latchup
EMI	Electromagnetic Interference
TI	Texas Instruments, Inc.
EIRP	Effective Isotropic Radiated Power
CAD	Computer Aided Design
BFN	Beam Forming Network

CP	Circularly Polarized
RHCP	Right Handed Circular Polarization
LP	Linear Polarization
HPBW	Half Power Beam Width
VNA	Vector Network Analyzer
RL	Return Loss
IIP3	Third Order Input Intercept Point
RDT	Requested Delivery Time
UML	Unified Modeling Language
MCU	Main Control Unit
POLITO	Politecnico di Torino
OBRF	Onboard Radio Frequency
TCP	Telemetry Command Processor
MCC	Mission Control Client
GSS	Ground Station Server
AUS	Authentication Server
ESA	European Space Agency
GFSK	Gaussian Frequency Shift Keying
BER	Bit Error Rate

Chapter 1

Introduction

The small satellite domain has been expanding considerably over the last decade through its low cost space mission characteristics. The major reason behind this fact is the availability of already developed ready-to-use subsystems and COTS components in the market. This has made it quite feasible from Universities and SMEs from a financial and development perspective to build their own satellites. CubeSat was the first such nano-satellite which was developed by California Polytechnic State University in collaboration with Stanford University. This CubeSat (1U) having dimension of $100 \times 100 \times 100 \text{mm}^3$ has evolved since, as CubeSat Standard [1]. Many other universities and SMEs have adopted this standard and are developing subsystems for it. The Department of Electronic & Telecommunication (DET) here at Politecnico di Torino, also developed their first nano-satellite called PiCPoT [2], which was intended to be launched together with other university satellites by a DNEPR LV rocket in July 2006, which unfortunately couldn't deploy due to a launcher failure. Since then, DET began work on another project called AraMiS [3]. The key feature of AraMiS design approach is its modular and scalable nature. These modules are designed to be reused for multiple missions that can help in significant reduction of the overall budget, development and testing time. It simply requires a reassembling of the already developed modules to achieve the mission specific target. AraMiS project module implementations have been done using COTS components which are low cost and easily available from market. AraMiS satellites have been specifically designed to withstand the harshness of Low Earth Orbit (LEO) while providing an expected operational life of around 5 years. AraMiS provides financially feasible and higher performance space missions with dimensions equal to or larger than CubeSats.

AraMiS modularity concept is based on tiles. These tiles or panel bodies can be aluminum panels, Honey comb lateral structures or printed circuit board with solar panel on exterior side and other electronic subsystems on interior side. The 1B9_CubeTCT is the (UHF and S-band) telecommunication tile and 1B8_CubePMT is the power management, attitude determination and control tile developed for AraMiS-C1 and other nano-satellites of CubeSat standard.

AraMiS-C1 is a CubeSat standard nano-satellite developed on the AraMiS project design approach. Four sides of the AraMiS-C1 are equipped with 1B8_CubePMT modules. The remaining two sides are devoted to the telecommunication tiles called 1B9_CubeTCT modules. 1B9_CubeTCT has RF frontend and carry a commercial deployable UHF antenna (one side) and a patch type S-band antenna (the other side). Inside the satellite there is room for batteries and payload boards.

1.1 Problem Statement

The major objective of this thesis is the design and implementation of telecommunication subsystem on a single CubeSat dimension tile called 1B9_CubeTCT. Such a design approach aims at reducing the weight, cost, power consumption and space occupation of the subsystems in such a way that can maintain the system performance and operation within safe link budget margins while providing a communication link between the satellite and ground station.

This work provides an innovative patch array antenna design for the conventional AraMiS satellite architecture (160x160x160mm³). It also provides a single patch antenna design for the smaller CubeSat Standard AraMiS C-1. Both these designs are done for S-band (2.43GHz) and must comply with the link budget criteria.

The other goals include proposing a GENSO compliant protocol for AraMiS C-1 satellite that can extend the communication link between satellite and ground station from a few minutes of an hour to virtually 24 hours.

1.2 Proposed Solutions

1B9_CubeTCT is based on the AraMiS project design approach. The main concept of the AraMiS design is the scalability and modularity of all the subsystems. These two characteristics set's apart this innovative architecture from main stream satellite design. The most significant approach for cost reduction of nano-satellite missions is to lower the design cost, non-recurrent fabrication costs and space qualification/testing cost which usually accounts for more than half of the overall budget. It is made possible by sharing the same design among multiple missions. Design reuse is the main feature of the AraMiS project, that is, to have a modular architecture based on a small number of modules which can be reused in multiple missions.

The use of COTS components also provides a further cost reduction in subsystems design. As these components are not design for space environment, therefore, it is vital to follow a proper selection process. COTS components greatly help in reducing mission cost but they required proper safety margins to be considered during design to allow safe operations in the harsh space environment.

For miniaturization and weight reduction purposes a single patch antenna and patch antenna array has been designed and developed for AraMiS C-1 and Conventional AraMiS satellite design.

For enabling the AraMiS C-1 satellite with GENSO Project there has been developed a compliant protocol that provides a 24 hours communication link to the satellite in LEO by utilizing the GENSO member ground stations remotely via internet. The protocol has are also been tested with different scenarios of possible errors both in uplink and downlink.

1.3 Thesis Organization

This thesis deals with the design and development of telecommunication subsystems for AraMiS Project and in particular for AraMiS-C1 satellite on a single CubeSat standard module. The implementation of S-band transceiver over a half on telecommunication tile is not a trivial task.

Several techniques were employed for reduction in terms of size, weight and power consumption while still achieving desirable performance for communication link. COTS components were used for 1B9_CubeTCT systems implementation. COTS for RF Front End design were selected on the basis of performance in harsh LEO environment, power losses, dimension and space occupied onboard 1B9_CubeTCT. In order to cope with the anomalies of the 1B9_CubeTCT subsystems different housekeeping sensors have been employed at various points of the tile.

This thesis also elaborates on the S-band antenna design for both versions of AraMiS satellites by employing innovative techniques to enhance performance while keeping the size, weight and cost within acceptable margins.

The chapter 2 presents an introduction to the AraMiS project and AraMiS-C1 satellite. Chapter 3 discusses different satellite design flow configurations.

Chapter 4 provides comprehensive details about the 1B9_CubeTCT, the CubeSat standard telecommunication tile developed for AraMiS-C1 and other nano-satellites based on similar standard. It also presents the designed S-band Telecommunication subsystem providing a radio communication link between satellite and the ground station. The design of each submodules onboard the 1B9_CubeTCT has been described in this work. They include the S-band transceiver, RF front end, Housekeeping Sensors, Tile Regulators, anti-latchup protection circuit and RF matching network.

Chapter 5 deals with the antenna design, fabrication and testing for both conventional AraMiS architecture and CubeSat Standard AraMiS C-1 satellite. The design, implementation and testing of single patch antenna and AraMiS patch array that are developed for AraMiS C-1 and Conventional AraMiS satellites respectively have also been discussed in great depth.

In chapter 6 the radio communication link budget calculations are performed for different scenarios ranging from worst to the best possible case. It also introduces and describes the Polito ground station and its various features. There is also a comparison with Link budget estimates for other ground stations (GENSO members).

Chapter 7 describes the AraMiS protocol that has been developed in compliance with GENSO project to provide an extended communication link which makes possible to track a satellite (remotely via internet) from any GENSO member ground station around the World. A new frame format is defined which makes AraMiS satellite compatible with GENSO. Also, different possible scenarios are presented in case of packet loss during communication in uplink or downlink.

Chapter 2

Introduction to Small Satellites

Small satellites market has considerably grown over the last decade. This has been made 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 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 [4-5]. CubeSat identifies the standard for small satellite with dimensions $10 \times 10 \times 10 \text{ cm}^3$ and a maximum mass of 1 kg, having a structure adapted to the launcher POD (Pico-satellite Orbital Deplorer). Another feature of CubeSat is the use of low-cost commercial components (COTS: Commercial Off-The-Shelf). These features greatly help in the reduction of cost and development time. In addition, the weight reduction allows the use of less expensive launchers. Satellites based on CubeSat standard have made it possible for potential clients to buy and assemble their own satellites. This (CubeSat) standard laid the foundation for several projects of nano-satellites by many universities and SMEs throughout the world. Different universities across Europe such as University of Wurzburg in Germany, the Norwegian University of Science and Technology and Italian universities including; the University of Rome La Sapienza, the University of Trieste and Politecnico di Torino.

Practically, any artificial satellite of low mass and low volume can be considered as a small satellite. However, the definition can be extended 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 revolutionized the landscape of space exploration. They facilitate quicker, cost effective and reliable access to the space. This provides a potential opportunity to have smaller project groups and encourages new actor 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.

2.1 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 LEO with the aim of:

- 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 onboard sensors
- Taking Earth's surface images
- Encouraging and creating interdisciplinary activities to enhance the interaction and coordination of various departments, faculty, PhD students and other students of the Politecnico.

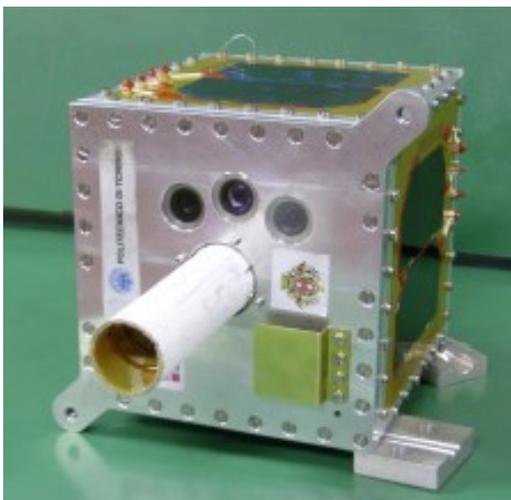


Figure 2.1: Photograph of PiCPoT satellite

PiCPoT is a satellite of cubic structure with dimensions $13 \times 13 \times 13 \text{ cm}^3$ and a mass of 2.5 kg. The six external faces are made of aluminum alloy type 5000 AlMn. Photograph of PiCPoT is shown in Figure 2.1. Solar panels are mounted on the five external faces which provide power to the PiCPoT. The sixth face has test connector and three cameras. There are two antennas (for 437 MHz and 2.4 GHz) which are used to communicate with the ground station. PiCPoT also contains two Kill-Switches which can guarantee the absence of power until the satellite is not completely detached from the launcher. The interior of the satellite houses two types of batteries (NiCd and NiMh) to power the satellite in the absence of the solar power.

The satellite has the following electronic circuit boards:

- Power Supply: recharge the batteries by converting incoming voltage from the solar panels and also maintain the temperature of the satellite.
- Power Switch: converts the battery voltage to different voltage levels required by all the other subsystems onboard PiCPoT.

- RxTx: receives and transmits signals between the ground station and satellite via 437 MHz and 2.4 GHz channels that operate in half duplex mode.
- ProcA and ProcB: are the two on-board computers implemented on separate PCBs. They duplicate the same functions. These processors and boards are built with different technologies and different components that provide redundancy to avoid possible failure. They both also acquire telemetry data and communicate with Payload and RxTx. These processors are not operated simultaneously for efficient utilization of the onboard available power.
- Payload: is responsible for acquiring earth images via onboard photo/video cameras. It also compresses these images to JPEG and sends them to the ProcA or ProcB, which manages their transmission to the ground.

2.2 AraMiS Project

Modular architecture for Satellites (AraMiS) is a project that wants to take further this concept and create a true modular architecture. The design approach of AraMiS architecture is to provide low-cost and high performance space missions with dimensions larger than CubeSats. The feature of AraMiS design approach is modularity. These modules can be reused for multiple missions which helps in significant reduction of 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 is to have a modular architecture based on a small number of flexible and powerful modules which can be reused as much as possible in various missions.

This architecture is intended for different satellite missions, from small systems weighing from 1kg to larger missions. The Figure 2.2 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 redesign 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 times to fit mission requirements [6].

2.2.1 AraMiS Satellite Subsystems

The AraMiS satellites can achieve the desired flexibility level by combination of several subsystems together. The main subsystems of this architecture are:

- Mechanical
- Power management
- Attitude determination and control
- Telecommunication

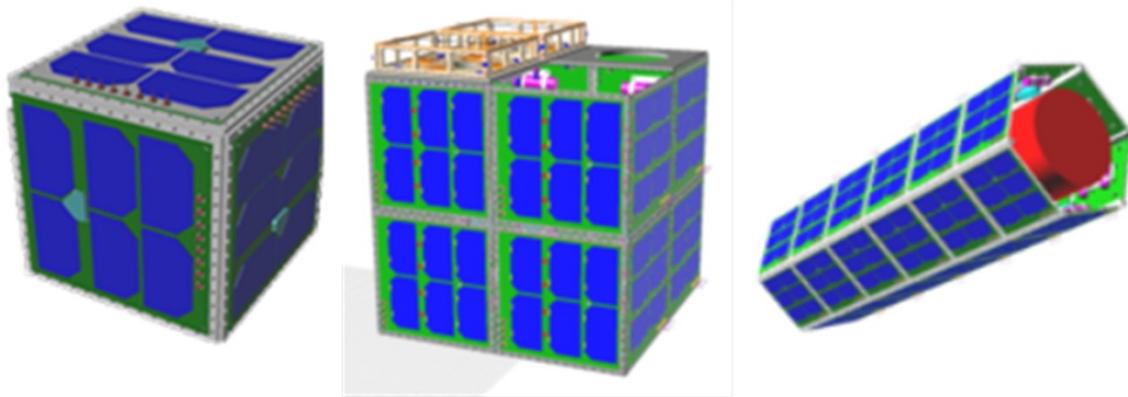


Figure 2.2: Different AraMiS satellites architecture

2.2.2 Mechanical subsystem

The mechanical subsystem is the backbone of a satellite. It provides physical structure to the satellite and holds in place all the subsystem together and also protect them from environment conditions. The main material used for building the AraMiS structure is Aluminum, used in particular for its light weight. The structure skeletal is made by metallic square rods while the power management and telecommunication subsystems are mounted on thin panels that are screwed to the rods. The power management tiles cover the satellite with solar panels mounted on the external face. The number of these tiles mainly depends on satellite size and power requirement. This provides a degree freedom to mission designers since size and generated power can be increased by simply adding more modules.

All the tiles are connected on the external faces of the satellite and the payload is mounted inside which can be altered by the mission requirement. Figure 2.3 shows the Aluminum frames of AraMiS structure.

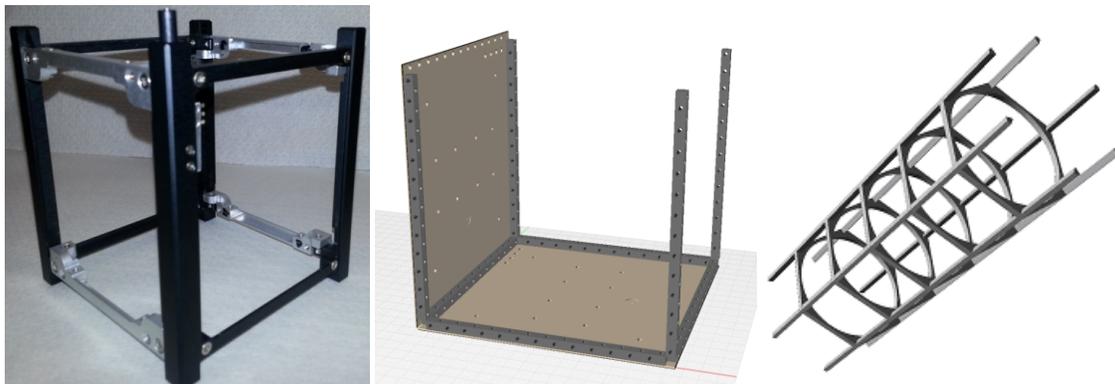


Figure 2.3: AraMiS mechanical structures

2.2.3 Power management subsystem

The power management subsystem is responsible for generating, storing and delivering power to all the other satellite subsystems. It is one of the most critical subsystem as a failure here can lead to shutting down everything. Fault tolerance is an important parameter and most of the design solutions were selected for this reason.

Conventionally, power management is mission dependent which requires ad-hoc development for the specific needs. This tends to increase overall system cost and testing time. For this reason the AraMiS project uses modular power management system that can be adapted for various missions. Figure 2.4 shows different solar panels of AraMiS satellites.

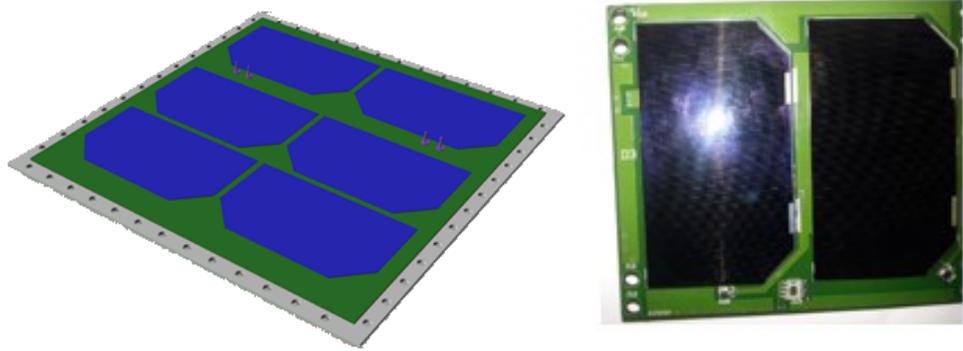


Figure 2.4: AraMiS solar panels

2.2.4 AraMiS Telecommunications Subsystem

The AraMiS telecommunications subsystem follows the modularity concept. There is a basic telecommunication tile that is provided in a standard AraMiS satellite. In case of special applications, dedicated tiles (as shown in Figure 2.5), can be added to meet mission criteria. This module is used to receive command and control packets from the ground station and to send back telemetry and status information. The bandwidth needed to exchange this kind of information is usually low, so the RF link was designed for low speed and low power. The module has been designed using COTS components which were selected to achieve good fault tolerance level. There are two different frequency bands used for satellite and ground communication: the UHF 437MHz and the S-band 2.4 GHz. To reduce occupied bandwidth, both channels are implemented using half-duplex protocol, sharing the same frequency for downlink and uplink.

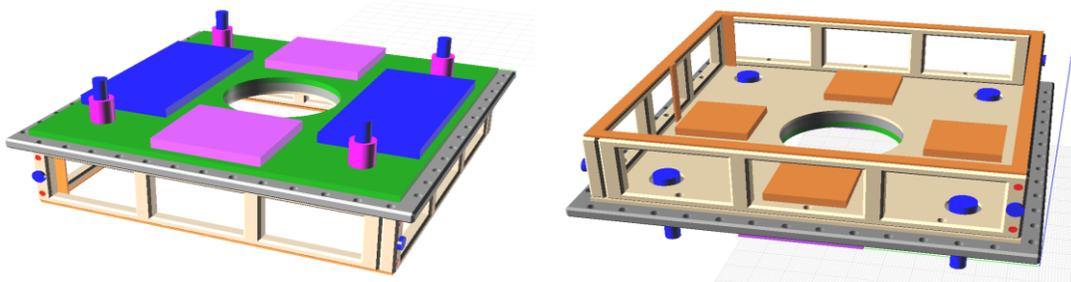


Figure 2.5: AraMiS telecommunication module

2.2.5 Tile Computer subsystem

Tile Computer (OBC) used in AraMiS satellites consist of redundant MSP430 microcontrollers and FPGAs, that are mainly responsible for overall satellite system management. Some of key responsibilities perform by OBC includes:

- Creating and transmitting (by Transceiver board) Beacon packets
- Decoding and executing commands
- Executing attitude control algorithm
- Storing housekeeping data
- Controlling Payload sub-systems

2.2.6 Attitude determination and control subsystem

This subsystem is mainly responsible for sensing and modifying satellite orientation for keeping the tile subsystems pointing at their targets. Attitude control can be performed in passive or active way: passive attitude control is usually achieved by mounting a permanent magnet in the satellite which acts as a compass in the Earth magnetic field. This system is extremely simple and consumes no power. The main drawback is lack of spin control due to the variable Earth magnetic field. Active control is performed using controlled actuators that modify satellite attitude on OBC commands. In AraMiS, attitude control is automatically performed by the satellite using magnetorquer and reaction wheels (as shown in Figure 2.6).

For attitude determination, three types of attitude sensors are used: magnetic, spin and Sun sensors. These sensors consist of COTS components which were selected on the basis of small dimension, light weight and low power consumption while achieving better performances.

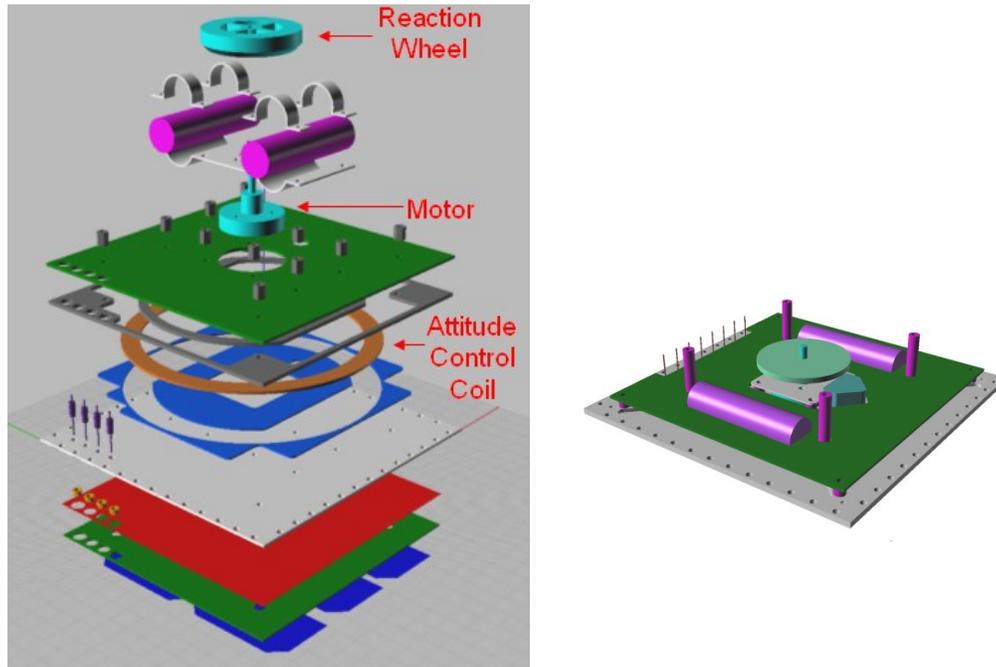


Figure 2.6: AraMiS ADCS modules

2.2.7 Payload

The payload is heavily mission dependent and the architecture was developed to allow high flexibility on it: the main requirements that the AraMiS architecture poses on the payload is its compatibility with the tile power distribution and data bus (as can be seen in Figure 2.7). Different payloads can be fitted in the various configurations but mechanical fixtures should be developed to connect them to the mechanical structure.

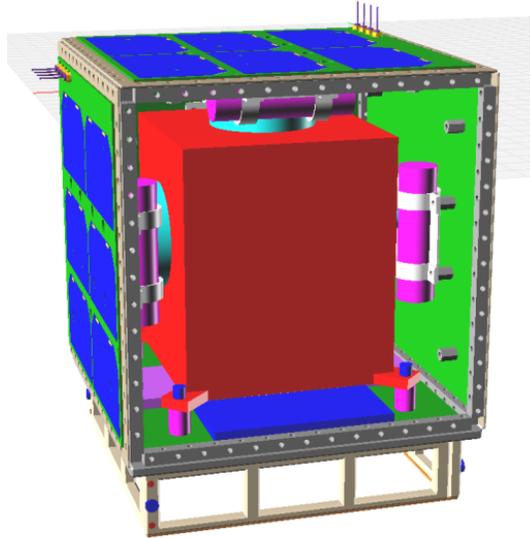


Figure 2.7: Payload inside AraMiS

2.3 AraMiS-C1

AraMiS-C1 is a 1U CubeSat standard nano-satellite built as a demonstrator of the AraMiS modular architecture developed since 2007 at Politecnico di Torino. 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 S-band 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.

AraMiS-C1 is based on the modular architecture AraMiS, which is modular at electrical, mechanical and software level. In this architecture, major bus functionalities are split over a number of identical modules that are placed in a proper manner in each tile (integrated inside the same PCB), making the design, maintenance, manufacturing, testing and integration very simple. The modules interconnect and dynamically exchange data and power in a distributed and self-configuring architecture, which is flexible because standardized interfaces between components are specified. Product variations can then take place substituting modular components without affecting the rest of the system. AraMiS-C1 is made by assembling a number of tiles developed at Politecnico di Torino, as detailed further, plus a few commercial off-the-shelf subsystems from

ISIS's CubeSat shop. Photograph of 1U AraMiS-C1 with four 1B8_CubePMT and two communication tiles is shown in Figure 2.8.



Figure 2.8: Photograph of AraMiS-C1

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 has the following subsystems:

- ISIS-1-Unit CubeSat Structure
- 1B8_CubePMT modules
- 1B9_CubeTCT modules
- Payload
- Battery pack
- UHF antenna
- Harness
- Onboard data handling
- Tile Processor
- Tile Computer (OBC)

2.3.1 ISIS - 1-Unit 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 built 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 Figure 2.9.



Figure 2.9: ISIS 1-Unit CubeSat structure

Parameter	Value	Unit
Part production tolerances	0.1	mm
Maximum supported mass (total)	2000	gram
Primary Structure Mass	100	gram
Primary + Secondary Structure Mass	200	gram
Outside Envelope (l×w×h)	100 x 100 x 113.5	mm ³
Inside Envelope (l×w×h)	98.4 x 98.4 x 98.4	mm ³
Thermal Range (min - max)	-40 to +80	°C

Table 2.1: Characteristic parameters of the ISIS 1-Unit CubeSat structure

2.3.2 1B8_CubePMT Module

EPS and attitude ADCS 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. It is built on an 82.5x96 x1.6 mm³ 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. COTS components have been used for 1B8_CubePMT implementation which is low cost and easily available from the market. It has been developed on the design approach of AraMiS architecture [7].

1B8_CubePMT subsystems include:

- 1B111B Solar Panel
- 1B1121D Boost converter
- 1B1251 Switching and linear regulators
- Housekeeping Sensors
- 1B221 Magnetometer
- 1B211B Gyroscope
- 1B235 Sun Sensor
- 1B222 Magnetic torque actuator)
- 1B223 Magnetorquer coil
- 1B4222 Tile Processor

2.3.3 1B9_CubeTCT

1B9_CubeTCT is a four layered PCB with dimensions 98.0mm×98.0mm×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 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 1B9_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 (TI) CC2510 Transmitter (Tx) and Receiver (Rx). The data processing, monitoring and control operation for 1B9_CubeTCT is also performed by these commercially available transceivers which have an internal 8051 core microcontroller. A photograph of 1B9_CubeTCT module is shown in Figure 2.10.



Figure 2.10: Photograph of 1B9_CubeTCT module

The main focus for 1B9_CubeTCT design is to realize a high bandwidth communication link using low cost COTS components which are available in market. This has made it feasible for the universities and SMEs to enter the field of satellite design. Moreover, a modular architecture approach has been employed for complete nano-satellite design. This modular approach makes the redesign convenient by the reuse of the identical modules.

2.3.4 1B31A Tile Radio Frequency Module 437MHz

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 2.11, 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.

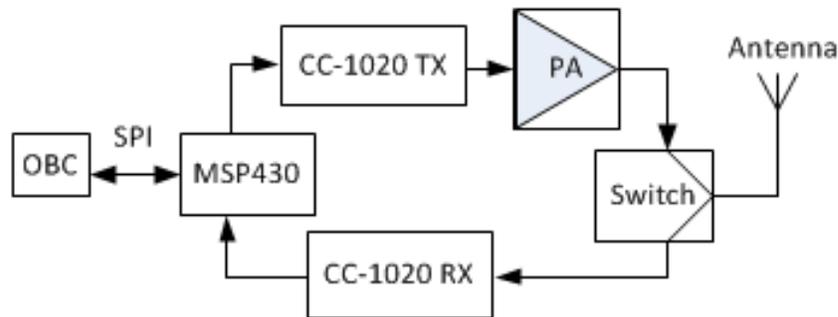


Figure 2.11: UHF band Communication Block Scheme.

2.3.5 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.

2.3.6 1B31B Tile S-band Radio Frequency Module

Consist of CC2510 transceivers that are configured separately as transmitter and receiver, respectively. The tile functions are performed by the 8051 cores of each CC2510 transceivers. The RF front end consists of PA, LNA that are connected to transmitting CC2510 and receiving CC2510 which is then fed to a four element monopole antenna via Solid State Switch as shown in Figure 2.12.

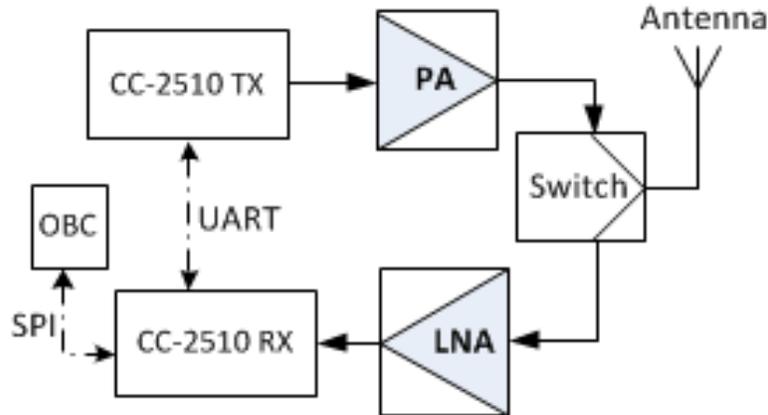


Figure 2.12: S-band Communication link Block Scheme.

2.3.7 1B3211_Single Patch Rodgers

For S-band communication Link, single patch antenna has been selected that is operated at 2.45 GHz. The compact design allows it to be mounted on external face of 1B9_CubeTCT. It is fed via coax feed from the internal S-band Module. Among the available materials, Rodgers' Duroid Substrate 6002 was selected which has a relative permittivity ' ϵ_r '= 2.94 and thickness ' t '=30mil/0.762mm.

2.3.8 Payload

The payload is a scientific/technological payload composed of three independent boards, 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 (E)

Characteristics common to Payload - Board 1, Payload - Board 2 and Payload - Board 3 are:

- mass: 35g
- size 80x80x8 mm²
- power consumption: 100mW average
- data rate requirements: about 50B every minute
- Harness

The harness of AraMiS-C1 is very simple. Each AraMiS - C1 has:

- MOLEX - 4 pins PicoBlade header connector for Power Distribution Bus; 12V nominal; max 200mA, except for the 1B9_CubeTCT tile, which can sink up to 800mA. This also shares ground node and kill switch connection.
- MOLEX - 5 pins PicoBlade header for tile data bus (I2C)
- MOLEX - 8 pins PicoBlade header for JTAG programming of microcontroller, to be used only on ground for microcontroller configuration

The power harness is therefore a bus connection for Power Distribution Bus; one plug for each tile and the payload (7 plugs). The data harness is therefore a bus connection for I2C; one plug for each tile and the payload (7 plugs). The RF harness is a coax cable from the 1B9_CubeTCT tile to the opposite ISIS - Deployable Antenna System for CubeSats.

All harnesses are tied to the Internal Structure of the AraMiS-C1. Connectors are fixed by means of appropriate silicone resin.

2.3.9 Battery

It consists of 10 rechargeable secondary batteries from SANYO. They are assembled in two packs of five batteries each.

- Nominal Voltage = 12V
- Capacity = 900mAh
- Mass = 230g
- Min Temperature = -20 °C
- Max Temperature = +60 °C

2.3.10 Tile Data Handling (OBDH) and Software

On Board Data Handling Subsystem is one of the critical subsystems of any satellite mission. Lot of design effort needs to be done for reliable data handling among different modules (sensors, actuators, etc.). The following sections show the detailed documentation On Board Data Handling and Software sections of AraMiS-C1 [8].

Tile Data Handling Bus

All the data handling of AraMiS-C1 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 (MSP430F5438 from Texas Instruments) [9]. The Slave may be a Tile or any other subsystem. Master communicates with the Slave via 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.

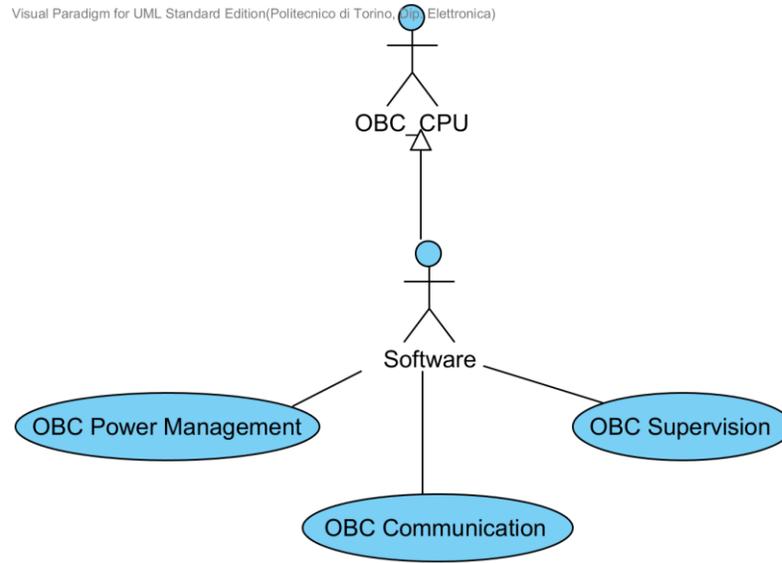
Most data transfers contain an appropriate START element; an 8-bit Master address; 8-bit Slave address ; a 16-bits command; an 8-bit data length field (only for Write Data and Broadcast Write Data); data (1B to 256B; except read data); a 16-bit CRC check and an appropriate STOP element. For AraMiS-C1, we select I2C physical layer for all connections.

2.3.11 On-board Computer

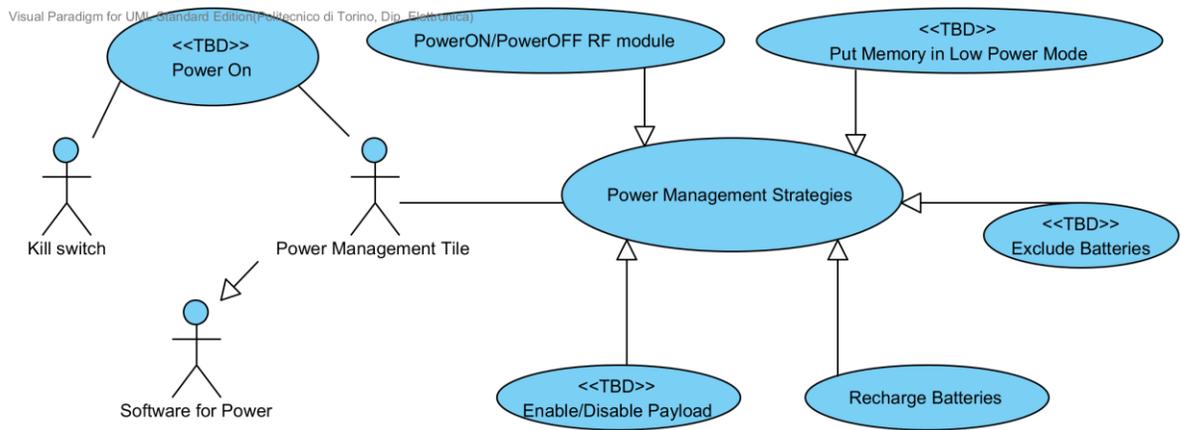
On-Board Computer (OBC) is the heart of any satellite mission. Use case diagram of Figure 2.13 shows the main functions of OBC. Some salient features of the OBC to provide the satellite flight segment with the following features:

- Processing resources for the flight mission software
- Telemetry and telecomm-and services and interfaces with the RF communication chain
- General communication services with the Avionics and payload equipment through an on-board communication bus.
- Time synchronization and distribution
- Failure tolerance architecture based on the use of redundancy implementation principle
- Managing payload data on the satellite and then transmitting it to ground station in proper format and efficiently.

One of the 1B8_CubePMT tile processors acts as the On Board Computer (OBC) for AraMiS-C1.



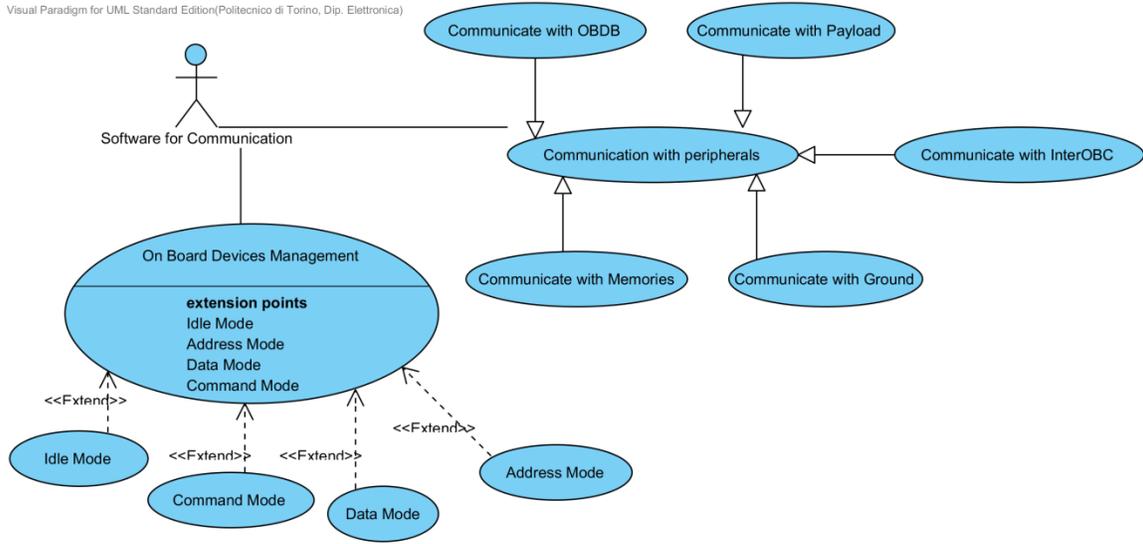
(a)



(b)

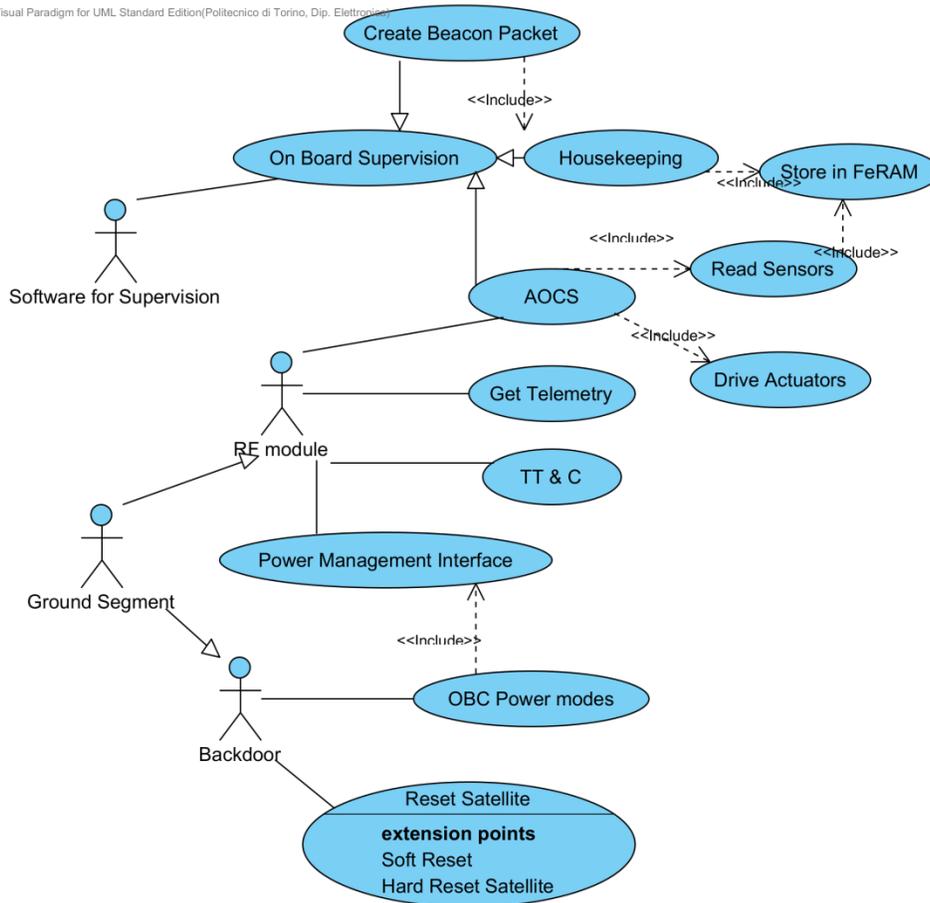
2.3 - AraMiS-C1

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



(c)

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



(d)

Figure 2.13: Major functions of OBC (a) OBC use cases (b) OBC power management (c) OBC communication (d) OBC supervision

Chapter 3

Satellite Design Flow

3.1 Introduction

In this chapter we are going to discuss the design techniques of AraMiS tiles and modules. Basically three types of design configurations are used. These configurations are discussed here and at the end a trade-off analysis of using modules in different spacecraft configurations has been given.

3.2 Design flow Configuration

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.

3.2.1 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.

3.2.2 Satellite on Demand Configuration

This configuration embeds the already developed and tested modules inside the PCB of the tile. In such a configuration; the design of physical modules is done permanently. Testing at tile and mission level is required and not at the module level. The CubeSat standard tile is built using this approach

3.2.3 Reusable Design Configuration

The reusable design configuration is an optimized 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 Figure 3.1. In [10], we have presented design and test of bus interface circuit on single module emphasizing the concept of plug and play approach.

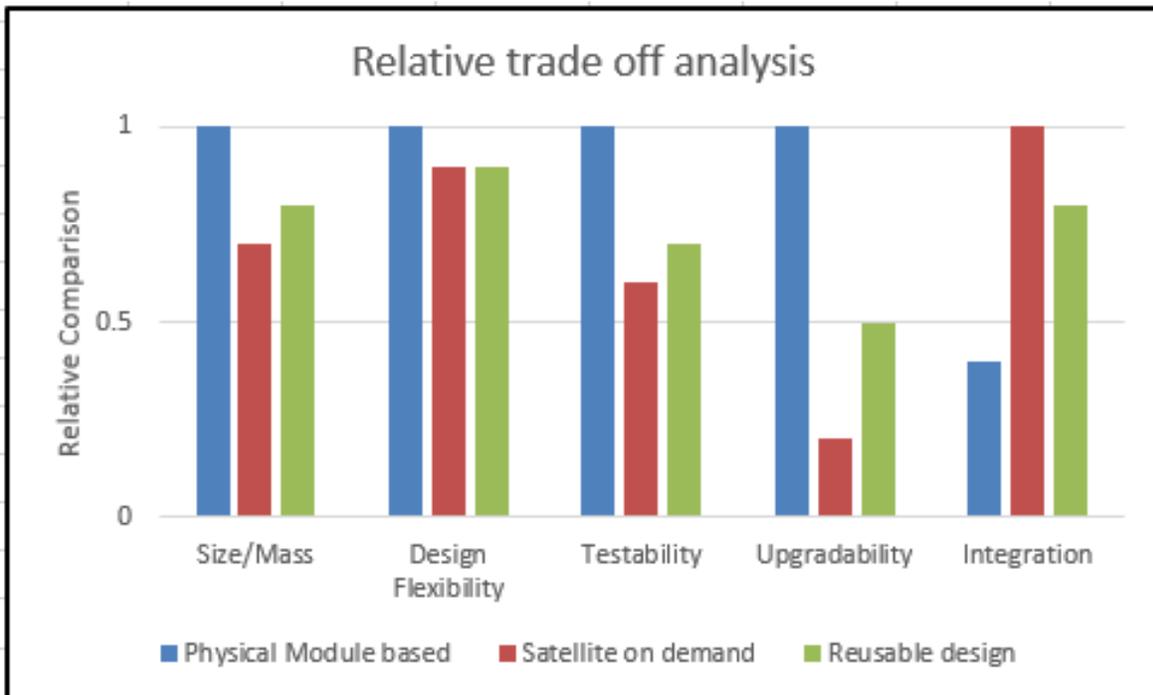


Figure 3.1: Trade off analysis of using modules in different spacecraft configurations

Chapter 4

Telecommunication Tile: 1B9_CubeTCT

Communication subsystem is a vital functional element for any satellite and it cannot be overlooked. The ever increasing bandwidth requirements have driven the Designers to move for conventional VHF/UHF bands to more high frequency bands. This shift is more fluent for medium and large satellites which have abundant resources interms of power budget and space available. On the contrary, for micro/nano-satellites this technological shift is not so convenient mainly due to scarcity of available space and power budget. For CubeSats standard this shift is even more difficult because of hard constraints on space, power budget and cost. The aim of this work is to implement a feasible design solution that could cater for bandwidth requirements while satisfying these constraints.

4.1 Overview

The AraMis-C1 is the most recent implementation of AraMiS architecture that is based on CubeSat Standard. The design process of AraMiS is based on tiles. Tiles are printed circuit boards that also form the outer structure of the Satellite. They have dual functionality; (i) provide mechanical strength and structure to the satellite; (ii) provide operational functionality (such as power management, telemetry, telecommunications, etc). Inside the satellite there is room for batteries and payload boards. AraMiS-C1 consists of 4 power management tiles (1B8_CubePMT) telecommunication tiles (1B9_CubeTCT) with S-band patch antenna (One Side) and a commercial deployable UHF antenna (the other side) mounted on external faces. A photograph of 1U AraMiS-C1 is shown in Figure 4.1.

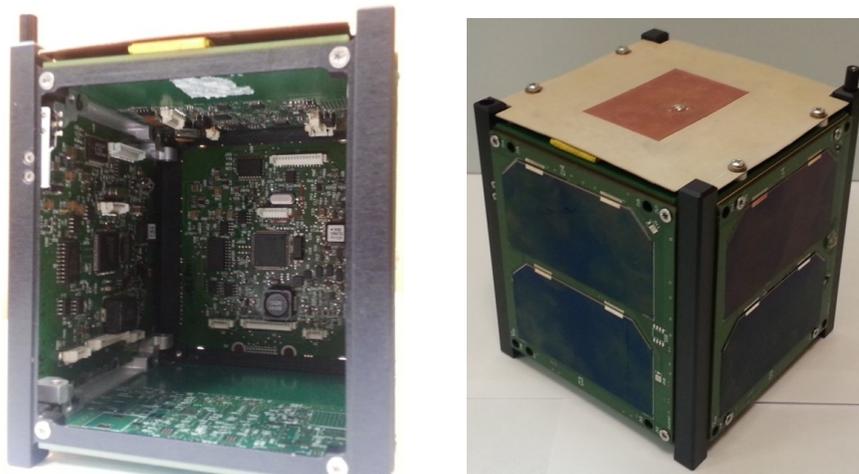


Figure 4.1: Photograph of AraMiS-C1 with four 1B8-CubePMT and two 1B9_CubeTCT tiles

The main approach for designing 1B9_CubeTCT is to realize a high bandwidth communication link using low cost COTS components which are available in market. This modular design approach of AraMiS-C1 makes the redesign convenient by the reuse of similar modules [3],[11].

Each CubePMT (as shown in Figure 4.2) contains electric power supply (EPS) and attitude determination & control subsystem (ADCS). The external surface of CubePMT consists of solar panels and sun sensor whereas; internal surface contains components and connectors. The current version of 1B9_CubeTCT only consists of (1B31B) S-band subsystem. In future 1B9_CubeTCT design will also incorporate (1B31A) UHF subsystem. The regulator/ sensors module and connectors required by 1B9_CubeTCT have been implemented on the same communication Tile as shown in Figure 4.3. From here onwards the 1B9_CubeTCT will refer to the current version which only contains (1B31B) S-band subsystem.

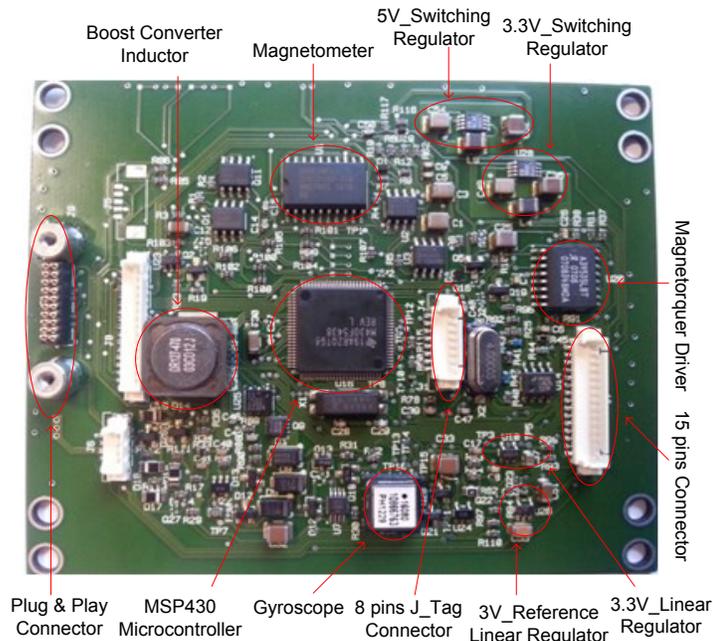


Figure 4.2: Photographs of 1B8-CubePMT module's solar panel and components sides [7]

1B9_CubeTCT is a four layered PCB with dimensions $98.0\text{mm} \times 98.0\text{mm} \times 1.55\text{mm}$. 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 bottom surface. All the digital traces are routed on the third layer. 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 1B9_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 [12]. The data processing, monitoring and various control operation for 1B9_CubeTCT is also perform by these commercially available transceivers which has an internal 8051 core microcontroller.

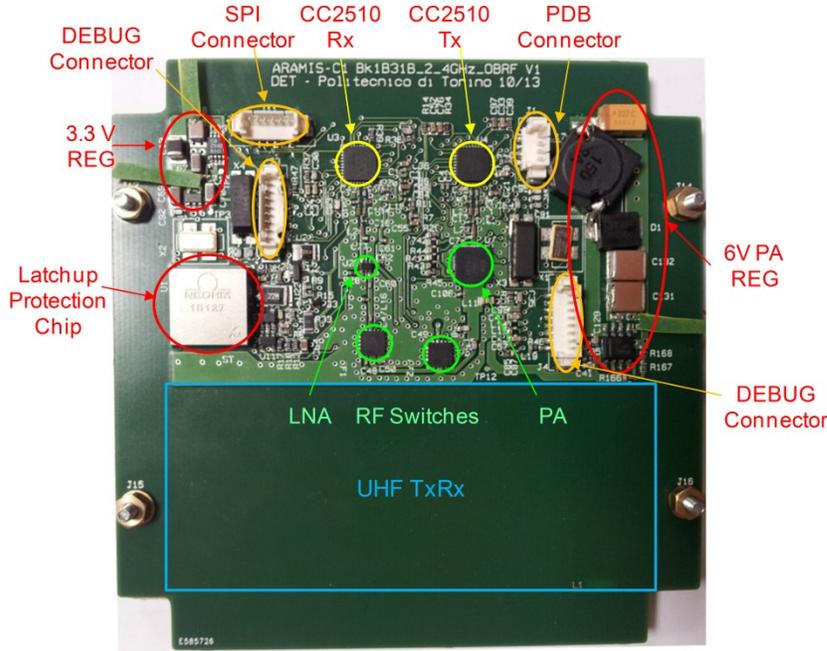


Figure 4.3: 1B9_CubeTCT Photograph

The main focus for 1B9_CubeTCT design is to realize a high bandwidth communication link using low cost COTS components which are available in market. This has made it feasible for the universities and SMEs to enter the field of satellite design. Moreover, a modular architecture approach has been employed for complete nano-satellite design. This modular approach makes the redesign convenient by the reuse of the identical modules.

The 1B9_CubeTCT module subsystems names are according to the nomenclature used for these subsystems inside UML diagrams [13].

UML (Unified Modeling Language), a widely used modeling language in high level object oriented designs is becoming a standard in vast application areas. UML provides a lot of independent diagrams, including the usecase diagram, class diagram, sequence diagram, requirement diagram, state diagram etc. Different diagrams exhibit system properties in different views. The choice of the diagram usually depends upon the characteristics of the system [14].

UML has been used for the design and documentation of the AraMiS project. All the subsystems blocks and their functionalities have been described using the UML. Hardware/software architecture of AraMiS, detailed documentation of each subsystem has been implemented using UML. The design methodology employs UML class diagram for every module. With the proposed approach, every subsystem is composed of a hardware (HW) part and a corresponding SW support.

1B9_CubePMT UML class diagram is given in Figure 4.4. All the subsystems names are according to the nomenclature used for their classes in UML. In this thesis, we will use the corresponding UML names of all the subsystems.

4.2 - 1B9_CubeTCT Design Overview

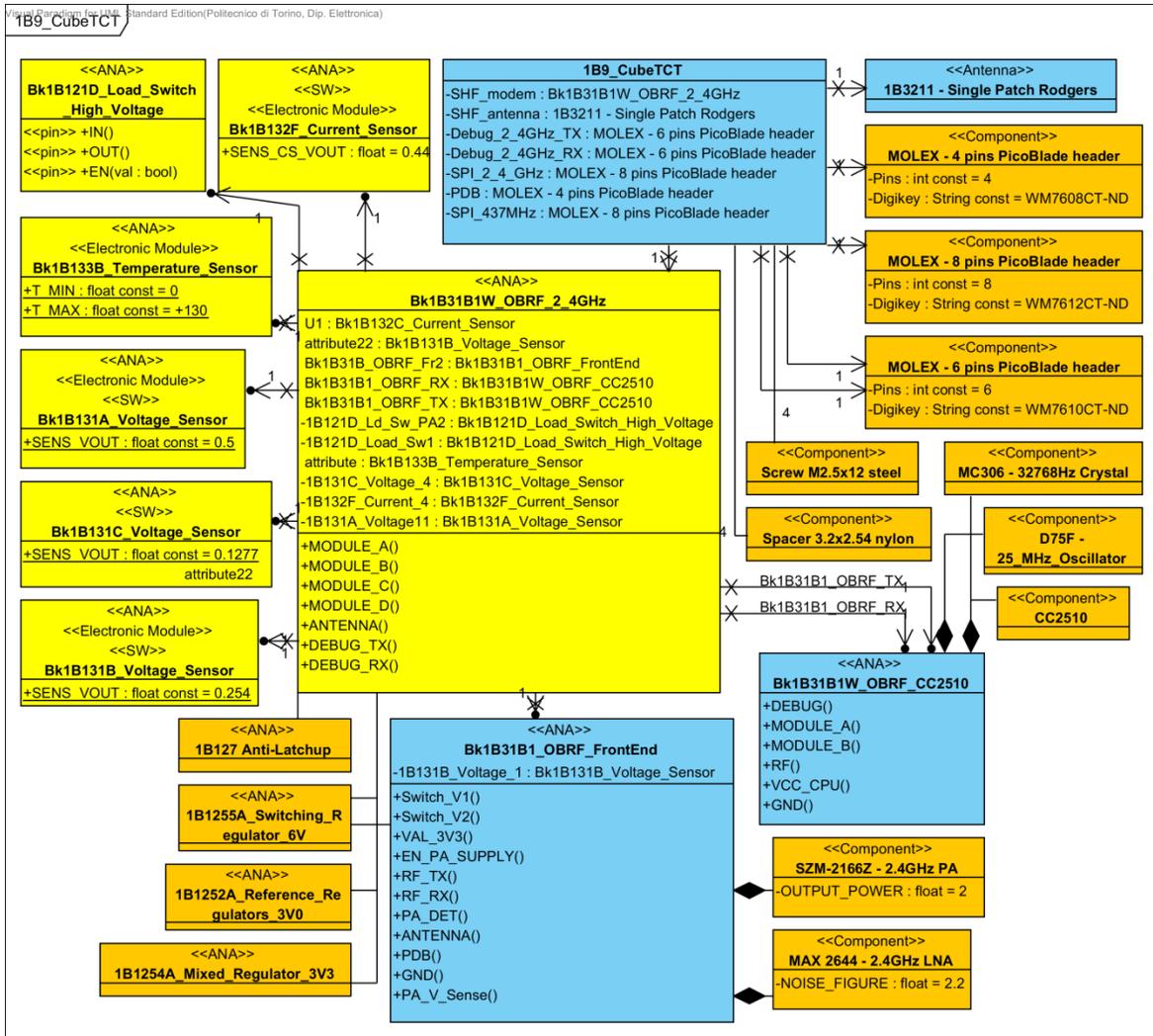


Figure 4.4: UML Class diagram for 1B9_CubeTCT showing association with all other subclasses and modules

4.2 1B9_CubeTCT Design Overview

To ensure a consistent communication radio link, the 1B9_CubeTCT has been designed as a separate subsystem from satellite stand point. It can communicate with other subsystem and “On Board Computer” (OBC) via SPI interface. 1B9_CubeTCT draws power from onboard Power Management System (EMS) via Power Distribution Bus (PDB). Power Regulation is performed by 1B9_CubeTCT Power Regulation module. There are three power regulators that provide voltage levels of 6V, 3.3V and 3V Ref on 1B9_CubeTCT. Anti-latchup protection system is there to protect CMOS circuitry against radiation hazards. Realtime monitoring of voltage, current and temperature is also performed by voltage, current and temperature sensors on 1B9_CubeTCT to ensure proper functionality and housekeeping.

1B9_CubeTCT communicate with OBC and other modules using SPI interface of the Rx. The Rx is always periodically sniffing for radio packet from the ground station. In the case, when a

packet is received by Rx, it performs a CRC check and FEC. The received packet is decapsulated and sent to OBC. For transmission OBC forwards data to Tx which is encapsulated and transmitted to the ground station. Besides performing radio communication the 1B9_CubeTCT transceiver also performs various status monitoring and control tasks. Different Sensors (temperature, voltage, current sensors), PA and anti-latchup circuits are monitored by these transceivers via analog and digital I/O ports. The PA load switch, Regulators (3V Ref, 3.3V and 6V), PA and RF switches are also controlled by Tx and Rx as shown in Figure 4.5.

Most efforts for use of COTS components in space environment are aimed to protect the CMOS circuits against fatal events such as latchup due to radiation hazards. To protect against such events, a commercial anti-latchup circuit component (1B127) is being used on 1B9_CubeTCT which isolates power to the load for a finite time to ensure latchup is extinguished [15].

The main subsystems of 1B9_CubeTCT include:

- CubeTCT Transceiver (1B31B1_OBRF_CC2510)
- CubeTCT Power Regulator & Load Switch
- Housekeeping Sensors
- 1B9_CubeTCT Anti-Latchup protection (1B127A)
- CubeTCT RF Front End (1B31B1W_OBRF_FrontEnd)

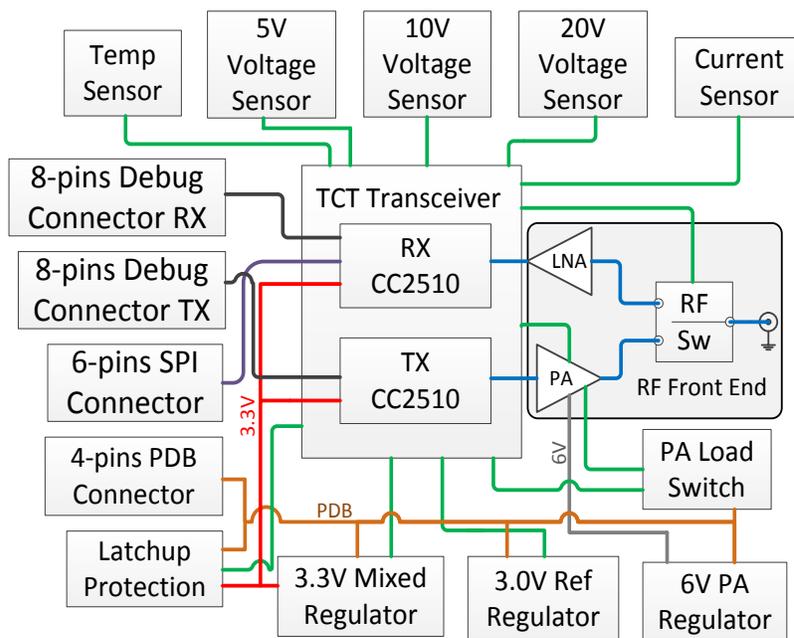


Figure 4.5: Block View of 1B31B1W S-band 1B9_CubeTCT system

4.3 CubeTCT Transceiver (1B31B1_OBRF_CC2510)

1B9_CubeTCT transceiver consists of two TI's CC2510 transceivers. Each of CC2510 transceiver chip is separately configured as a transmitter (Tx) and a receiver (Rx). It is a low-cost,

wireless communication application at low power and low voltage. Its main features are described as [12]:

1. Synthesis of high performance RF transceiver with the microcontroller, based on the architecture of the 8051 Enhanced
2. Ability to program the flash memory blocks 8, 16, 32 KB
3. Frequency range 2400- 2483.5 MHz
4. 1, 2, 4 KB of RAM
5. Supply voltage range 2.0 to 3.6 V
6. High sensitivity between -100dBm at 10 kbps
7. Possibility to program the data rate over 500 kbps
8. Low power consumption (22mA received, 23mA in transmission) with a microcontroller running at 26 MHz
9. Programmable output power up to 1dBm for all power
10. 2 USART Interfaces
11. 4 Timers
12. 14-bit ADC upto 8 inputs

The 1B9_CubeTCT transceiver have 8051 core which for Rx is setup as a master, while for Tx is configured as slave on 1B9_CubeTCT. Both Rx and Tx communicate via UART interface which is configured as a null modem. There is also debug interfaces available for the flash programming and debugging each of the CC2510 chips. The analog and digital I/O pins available on each of the chip are utilized for the housekeeping of 1B9_CubeTCT. The monitoring of temperature sensor, voltage sensor, current sensors, power amplifier output level onboard 1B9_CubeTCT is also performed by interfacing these signals onto the analog ADC input pins of CC2510. Also, the RF switches, PA load switches, PA enable signals and Regulator enable signals are interfaced via the Digital I/O pins of CC2510 as shown in Figure 4.6 below:

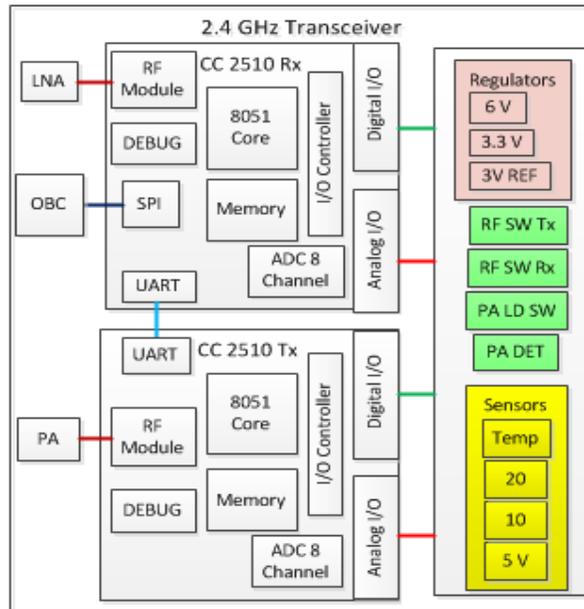


Figure 4.6: Block view of 2.4 GHz Transceiver (1B31B1_OBRF_2.4GHz) along with interfaces.

In order to interface and connect the CC2510 chip with the other devices onboard the 1B9_CubeTCT they were mapped into two modules (named Module A and Module B) where set of available signals on each pin were mapped onto the existing AraMiS modular architecture as shown in the following Table 4.1 [16].

Connector	Pin	A	B
D0/RX/SOMI	11	P0.2	P1.7
D1/TX/SIMO	9	P0.3	P1.6
D2/SCL/SOMI	7	P2.0	P2.0
D3/SDA/SIMO	5	-	-
D4/CLK	3	P0.5	P1.5
D5/PWM	1	P1.2	P1.1
D6/A0	12	P0.0	P0.6
D7/A1	10	P0.1	P0.7
D8/ID/INT	4	P0.4	P1.0
D9/EN/PWM2/INT	2	P1.3	P1.4

Table 4.1: Mapping of CC2510 pins onto AraMiS Module

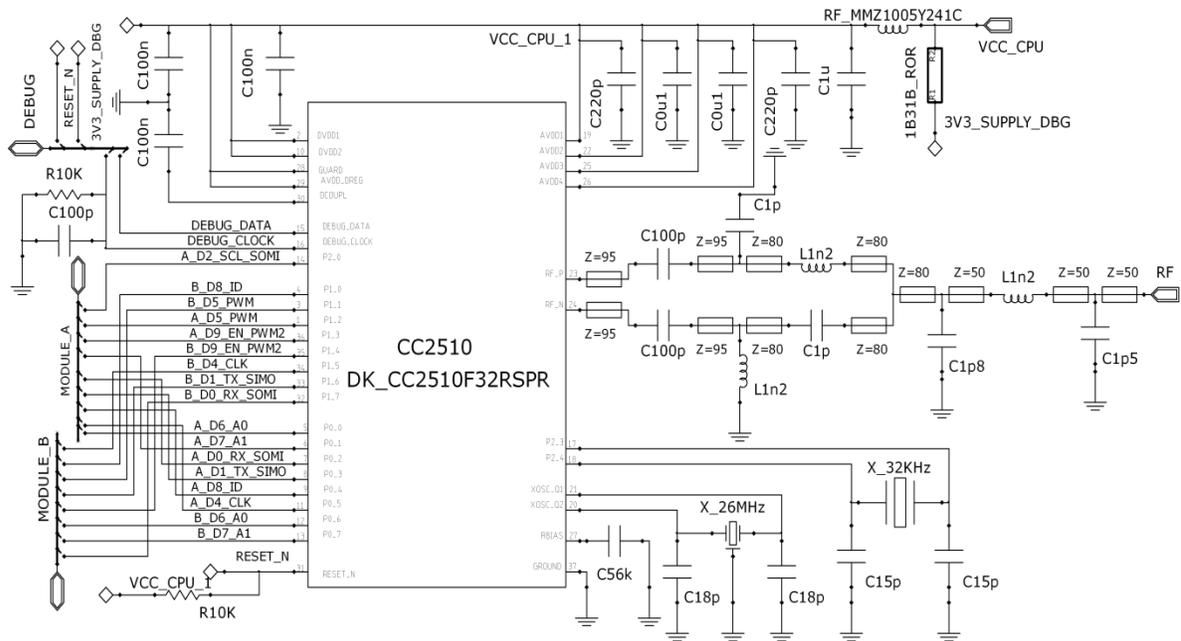


Figure 4.7: Mentor Schematic of 1B31B CC2510 Transceiver.

4.3.1 CubeTCT Regulators & Load Switch

The PDB voltage level (12-18V) needs to be downconverted to low voltage levels (i.e. 3V Ref, 3.3V and 6V) in order to be used by different subsystem components. For this reason 1B9_CubeTCT uses a linear (3V Ref), switched regulator (6V) and a mixed (3.3V) regulators. The linear regulators in general have smaller dimensions and require fewer auxiliary components. The main disadvantage associated with using linear regulator is of low power efficiency which increases the power losses. These power losses can be minimized by reducing the down conversion step between the regulator input and output stage or if low current is drawn from the regulator. Switching regulators have higher efficiencies but take larger space on the PCB for implementation and require more auxiliary components [17]. The choice of whether using linear and switching regulators or mixed regulator depends on the desired application.

1B1252A_Reference_Regulator_3V0

A linear regulator (COTS component) is used for the 3V reference voltage supply. The output of this regulator provides a stable input reference to the 1B9_CubeTCT current sensor. It down converts 5V input to 3V reference voltage. Output current and input/output voltage difference is low which results in negligible power loss, therefore a linear regulator is used. It can be enabled from the transceiver core via EN_REF. The schematic of the 3V reference voltage supply is shown in the Figure 4.8.

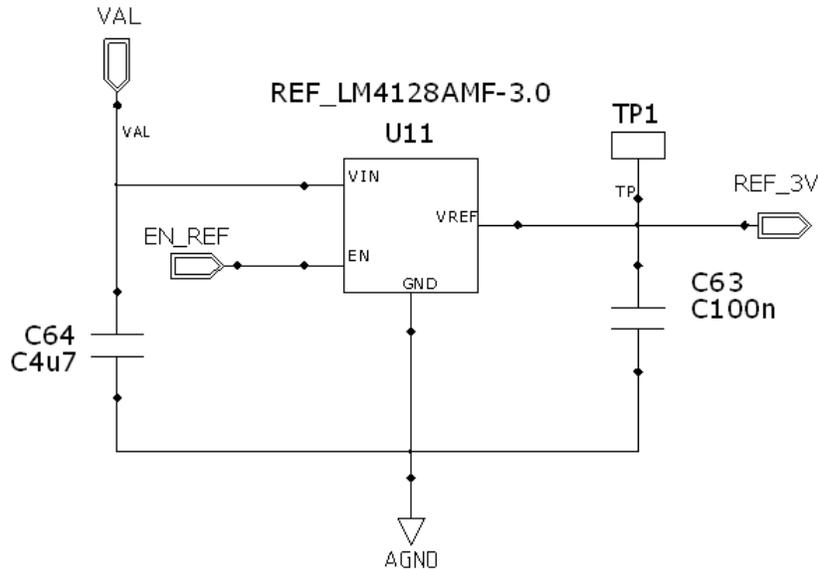


Figure 4.8: Schematic of the 3V reference liner voltage regulator

1B1255A_Switching_Regulator_6V

As, in AraMiS-C1 the power amplifier is the device that draws the most power (when satellite is transmitting in S-band), therefore, a very efficient and stable power supply is required that can provide 6V and current of upto 2.5A. Buck switching regulator (TI's TPS 5450) is used for downconversion of PDB to 6V and feeds the power amplifier (PA). Can be enabled/ disabled using EN_6V control signal from transceiver core [18].

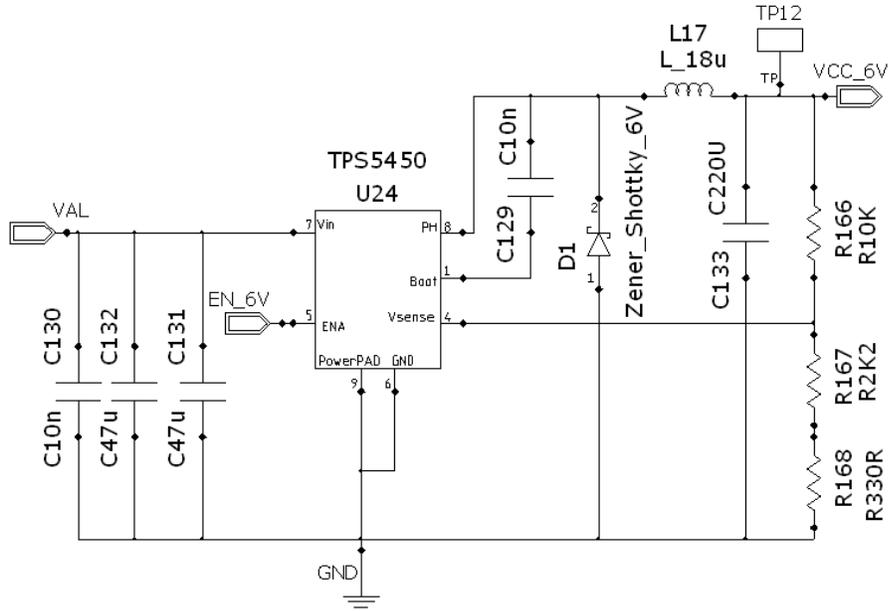


Figure 4.9: Schematic of 6V switched voltage regulator for PA

1B1254A_Mixed_Regulator_3V3

The switched regulators have high efficiency but at the same time have poor EMI characteristics which make them unsuitable for driving microcontrollers and CC2510 in particular. Also, a linear voltage regulator has an efficiency issue which makes them undesirable for our application. Therefore, a combination of switch and linear regulator called mixed regulator has been designed to provide an acceptable solution. This regulator has two stages. The first is switching stage that steps down voltage from PDB to 5V. It is a high efficiency stage that minimizes the power losses. The second stage is linear and it further takes down voltage to 3.3V and mitigates any possible EMI issues. Both these stages can be controlled by an EN_3V3 and EN_5V signal from the transceiver core as shown in Figure 4.10.

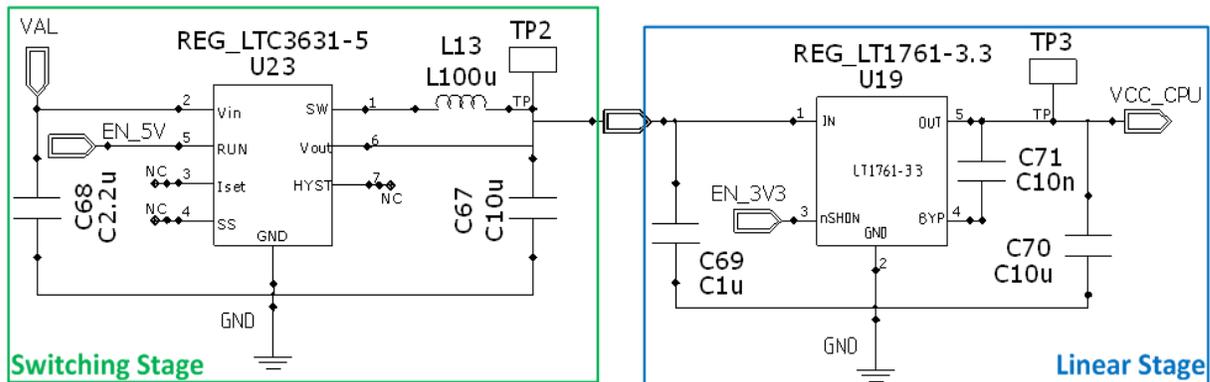


Figure 4.10: Schematic of 3.3V mixed voltage regulator with switching and linear stages

1B121D_Load_Switch_High_Voltage

The 1B9_CubeTCT draws considerable amount of power when the power amplifier is enabled for transmission (4W). In order to control this power consumption a load switch is mounted at the PDB input of 6V switching regulator to control PA more effectively. This load switch can control the supply and cutoff power to the PA through enable signal (EN) from the transceiver core as shown below in Figure 4.11.

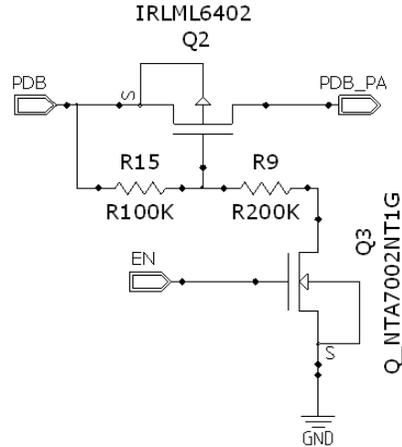


Figure 4.11: Schematic of the load switch

4.3.2 Housekeeping Sensors

The 1B9_CubeTCT contains voltage, current and temperature sensors that can provide housekeeping and monitoring. These sensors are mounted at different points on the tile. These sensors are connected to the analog (ADC) ports of the transceiver core which takes necessary counter measures in case of any abnormalities or anomalies to protect system from any possible damage.

1B131X_Voltage_Sensors

1B9_CubeTCT consists of four voltage sensors having voltage limits of 10V, 20V and (two with) 5V (1B131B_Voltage_Sensor, 1B131C_Voltage_Sensor, 1B131A_Voltage_Sensors, respectively), which constantly monitor the output voltage levels of Switch Regulator, PDB, 3V REF and 3.3V. Schematic of the 20V voltage sensor mounted on the PDB input is shown in Figure 4.12.

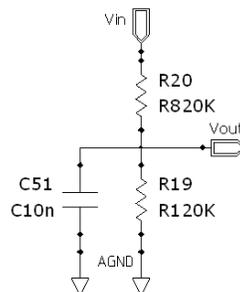


Figure 4.12: Schematic of (1B13C) 20V voltage sensor

The analog output voltage (V_{out}) connected to ADC of the Transceiver, is given by (4-1);

$$V_{out} = V_{in} \frac{R_{19}}{R_{19} + R_{20}} \quad (4-1)$$

By observing schematic from the output side, it behaves as a low pass filter. The sampling frequency ($f_{sampling}$) is determined by the value of C_{51} and R_{eq} ($R_{19} // R_{20}$). In order to avoid aliasing, the criteria given in (4-2) should be satisfied;

$$f_{sampling} \geq 2 \times \frac{1}{2\pi C_{51} R_{eq}} \quad f_{sampling} \geq 2 \times \frac{1}{2 \times \pi \times C_1 \times R_{equ}} \quad (4-2)$$

From (4-2) it is evident that $f_{sampling} \geq 304$ Hz in order to avoid aliasing. Figure 4.13 shows the internal ADC circuit of the CC2510 microcontroller has a resistor (R_i) and a capacitor (C_i).

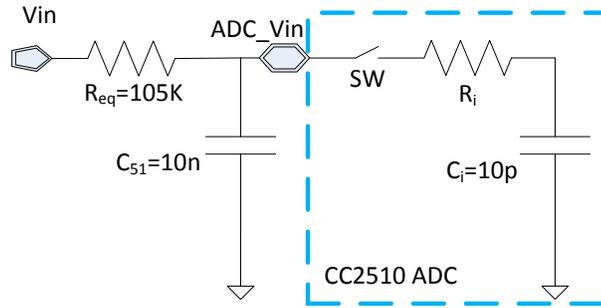


Figure 4.13: Equivalent circuit of 20V voltage sensor attached on CC2510 ADC pin

R_i has a small value around 10Ω and can be neglected, while C_i is in parallel with C_{51} and can add error. It can add aliasing by changing the sampling frequency point. Choosing a suitable value of C_{51} prevent this aliasing. The charges tend to distribute between C_{51} and C_i : The charge at C_i (Q_i) is zero when switch is open and C_{51} (Q_{51}) accommodates the total charge (Q_t) as given in (4-3):

$$Q = C_i V_i = 0 Q_i = C_i \cdot V_i = C_i \cdot 0 = 0C$$

$$Q = C_t V_t = Q_t \quad (4-3)$$

When switch (SW) is closed, C_i starts to charge from Q_t . During transitional state the two capacitors are at the same voltage as given by (4-4):

$$Q_i = C_i V_a = \Delta Q$$

$$Q_{51} = Q_t - \Delta Q = C_{51} V_a \quad (4-4)$$

Where V_a is the C_{51} voltage after SW closing and is given by (4-5);

$$V_a = V_{51} \frac{C_{51}}{C_{51} + C_i} \quad (4-5)$$

To find out the relative error introduced by C_i , we compare the difference ($V_t - V_a$) and original signal V_t which is given by (4-6):

$$e_{V_{51}} = \frac{\Delta V}{V_{51}} = \frac{|V_{51} - V_a|}{V_{51}} = \frac{C_i}{C_i + C_{51}} \quad (4-6)$$

For reducing this error, C_{51} should have a very high value as compared to C_i . The CC2510 with 12bit ADC and supply voltage of 3.3V has a quantization error 0.0805%. The voltage error ($e_{V_{51}}$) should be smaller than the quantization error. It reflects that C_{51} value should be very high as compare to C_i , in order to keep the error smaller. The selected C_{51} value is 500 times greater than C_i .

1B132F_Current_Sensor

1B9_CubeTCT contains a current sensor that monitors the total tile current consumption at the PDB input. It has a maximum current rating of 5.682A. Figure 4.14 shows current sensor schematic where output voltage (CS_VOUT) is connected to the ADC of the transceiver and is given by (4-7) [19];

$$V_o = \frac{I_{Sense} \times R_{Sense} \times R_{out}}{5k\Omega} \quad (4-7)$$

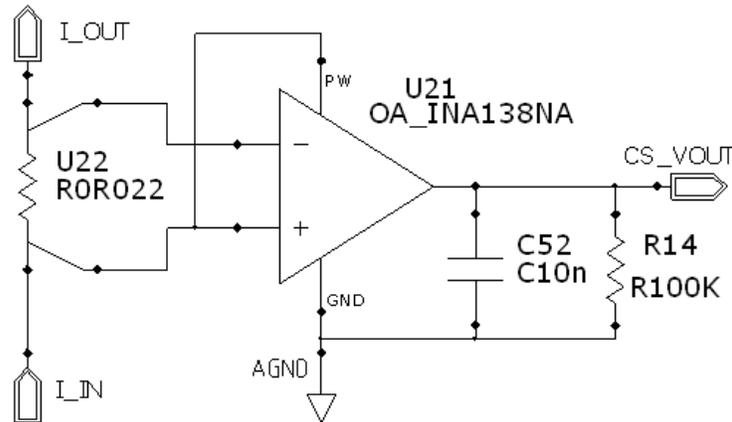


Figure 4.14: Current sensor at PDB input of 1B9_CubeTCT (Mentor Graphics schematic).

By observing above schematic from the output side, it behaves as filter. The sampling frequency ($f_{sampling}$) is set by the value of C_{52} and R_{14} . In order to avoid aliasing, criteria in (4-8) should be satisfied:

$$f_{sampling} \geq 2 \times \frac{1}{2\pi C_{out} R_{out}} \quad (4-8)$$

4.3.3 1B133B_Temperature_Sensor

1B9_CubeTCT module has one temperature sensor which is a glass protected NTC thermistor and has a temperature range from 0°C to 130°C. Output voltage of the sensor is linearly

proportional to temperature and it is sensed by the ADC of Transceiver. Its resistance range varies between 2.2k Ω ~100k Ω , highly accurate with 1% of tolerance. Output voltage of the sensor is linearly proportional to the temperature. Mentor Graphics schematic of the implemented temperature sensor is shown in Figure 4.15.

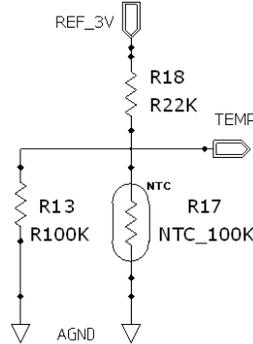


Figure 4.15: Schematic of the temperature sensor

4.3.4 1B12_Anti-Latchup

The *NeOhm Latchup and overcurrent protection system* 1B127 [15] has been used onboard 1B9_CubeTCT to protect components or systems against latch-up and to prevent system damages that can be caused by striking of high energy particle. These kinds of event can be discovered by monitoring the device supply current and by observing a sharp increase in it. Such a scenario can result in Single Event Latchup (SEL) [20]. The only solution in such case will be to turn power off as quick as possible. It prevents latch-up damages on ICs by turning them off before they can become critical. The particle strike is detected by a sudden increase in current consumption which exceeds a programmed threshold. Power supply is then cutoff in a short time and it is turned on again after an off time has elapsed to ensure that the device will be correctly operating afterwards as.

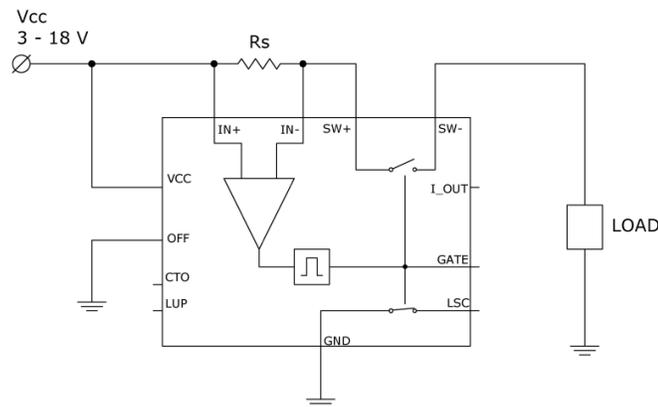


Figure 4.16: Basic 1B127 anti-latchup protection circuit [15]

The circuit used onboard 1B9_CubeTCT latch-up protection system implementation is depicted in Figure 4.16 having only one external component sense resistor (R_s). The latch-up threshold

current(I) can be programmed by means of R_S according to the equation (4-9); where threshold voltage (V_{th}) has a typical value of 30mV [15]. R_S resistor could also withstand the Joule effect:

$$R_S = \frac{V_{th}}{I} \quad (4-9)$$

4.3.5 CubeTCT RF Front End

The RF front End is the most important subsystem of the 1B9_CubeTCT tile. In transmit chain it is comprised of the transmitting CC2510 RF matching network and Power amplifier that connects to the (external) patch antenna using two RF switches. The receive chain starts from antenna that is connected through two RF switches to the LNA that further connects to the receiving CC2510 via RF matching network as depicted in Figure 4.17. This is a half-duplex system using same antenna for transmit and receive. The receiver and transmitter chains are isolated together using two RF switches. The concept behind using two RF switches is to provide a better isolation to the LNA. The matching network of CC2510 has been designed in a symmetric fashion to avoid any delays and lags. The RF front end consists of COTS components which have been selected according to certain design requirements. The LNA, PA and RF switches tend were tested individually for compliance. The passive components used in the overall design of matching network are chosen on the bases of least parasitic behaviors and to ensure optimum matching (according to the manufacturer specifications).

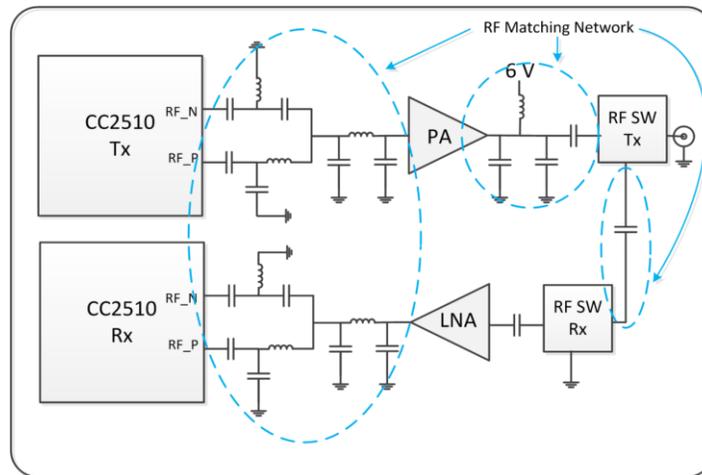


Figure 4.17: RF front end Block diagram

The RF front end consists of following major modules:

- 4.3.5.1. Low Noise Amplifier (LNA)
- 4.3.5.2. Power Amplifier (PA)
- 4.3.5.3. RF Switches
- 4.3.5.4. RF Matching network

4.3.5.1 1B316B_Low_Noise_Amplifier

From the characteristics shown on the datasheet for the CC2510 transceiver found denotes the value that cannot be transmitted over a certain amount of data rate, but the sensitivity is better to have a value of -81dBm allowing a data rate equal to 500kbps. So it was necessary to use a low noise amplifier, LNA, to mitigate the noise power than signal. The purpose of using LNA is to provide a high gain and the least possible noise figure. For LNA design of 1B9_CubeTCT different COTS components were analyzed on the basis of gain and noise figure. Among them the Maxim MAX2644 was chosen as the best suitable option because it provides a maximum gain of 17 dB and a noise figure of 2.2 dB (at 2.43GHz) [21]. It can be used in two different configurations, one as a high gain option but is not very linear, so we have a value of IIP3 of -3dBm. The second allows for a gain of 16 dB and an IIP3 of 1dBm which has been chosen for our design, as depicted in Figure 4.18. A length of 10.16mm has been added between bypass capacitors C₆₂ and C₆₁ to mitigate 3.3V supply noises to the LNA.

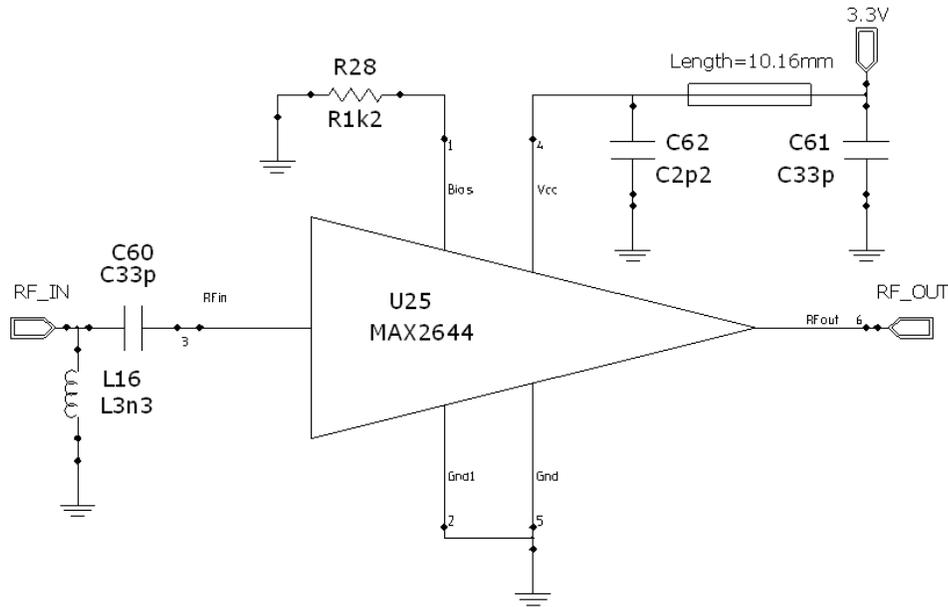


Figure 4.18: LNA Schematic of 1B9_CubeTCT RF front end

4.3.5.2 1B315B_Power_Amplifier

The power amplifier (RFMD SZM-2166Z) selected is COTS component [22]. Its a high linearity Class AB power amplifier designed technology HBT (Heterojunction Bipolar Transistor) with both InGaP GaAs semiconductor devices used in high frequency (suitable for space environment). The power amplifier provides an output power of 33dBm and a gain of 36 dB. This power amplifier incorporates an active bias circuit which gives designers the flexibility to optimize performance in specific applications. There is a power detector and a power enable signal available for its control and monitoring.. In our case a power of 2W is generated at the and the voltage source of 6V at a frequency of 2.4GHz as shown in Figure 4.19.

Using these measured values, the Gain (G_{dB}) and Efficiency (η) was calculated using the following equations:

$$G_{dB} = P_{out} \text{ dBm} - P_{in} \text{ dBm} \quad (4-10)$$

and
$$\eta = \frac{P_{out}}{P_{al}} \quad (4-11)$$

where:
$$P_{al} = VI \quad (4-12)$$

Figure 4.20 depicts the measured gain of the PA which is acceptable for 1B9_CubeTCT. The efficiency is also reported in Figure 4.21 which seems to agree with the manufacturer values in reported in the datasheet [22].

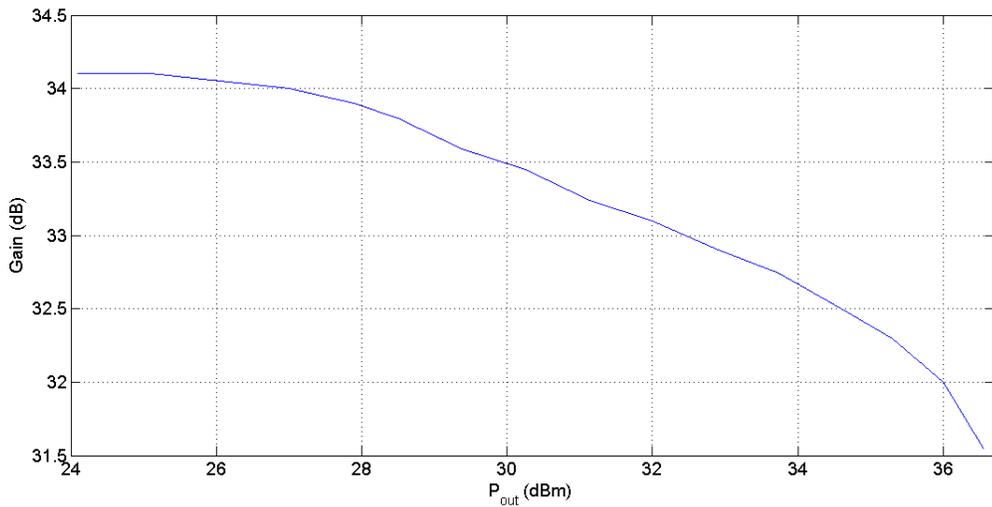


Figure 4.20: Measured Gain of power amplifier SZM-2166Z.

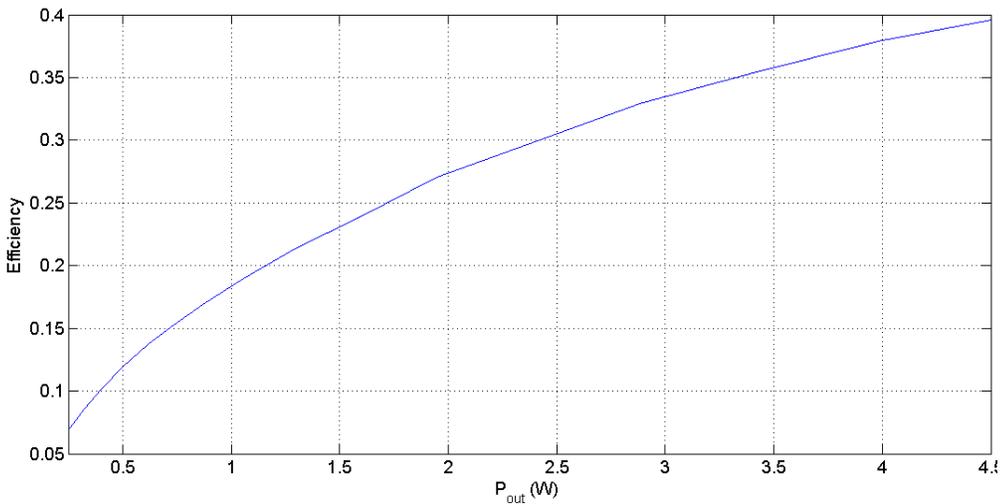


Figure 4.21: Efficiency of SZM-2166Z

4.3.5.3 1B317B_RF_Switches

The RF switches designed for 1B9_CubeTCT front end uses two COTS components TriQuint (TQP4M0010) [23]. These are high absorption GaAs SPDT switches that provide the most suitable results among the other contenders. Each switch can sustain a maximum input power of 36dBm and provides an isolation of 45dB between each RF port with an insertion loss of 0.9 dB at 2.43GHz. The RF switch operates on 3.3V and both them are controlled by the enable control signals V_c from the transceiver core as shown in Figure 4.22. When V_c is low RF_C and RF_1 are connected, whereas RF_C and RF_2 are isolated or in other words the receive chain is enabled. On the contrary, if V_c is high RF_C and RF_2 are connected while RF_C and RF_1 gets isolated therefore, the transmit chain is enabled as shown in figure. Here, a precautionary measure is mandatory to enable power amplifier only once the transmit chain is enabled to avoid high power dissipation within the RF switch which can cause physical damage to the 1B9_CubeTCT.

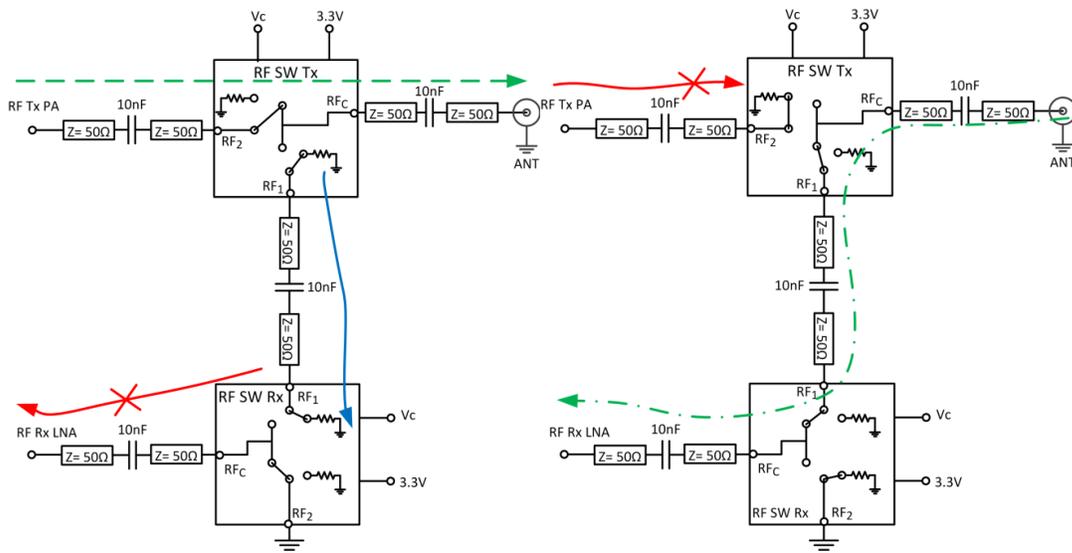


Figure 4.22:(a) RF switches: transmit mode ($V_c = \text{Low}$); (b) RF Switches receiving mode ($V_c = \text{High}$)

4.3.5.4 RF Matching Network

The 1B9_CubeTCT has been realized on a 4 layer PCB, as already illustrated in the first section of this chapter. The RF traces have been placed on the top most layer of PCB. RF front end network has been designed on an enhanced FR-4 substrate with dielectric constant (ϵ_r) of 4.7 and height (h) of 0.36 mm with an operating frequency of 2.43 GHz as shown in Figure 4.23. This increases the need for proper RF matching network to yield acceptable results. Thus, all the selected RF COTS components were closely analyzed and for each of them a matching network was designed in close compliance with the manufacturer specifications and 1B9_CubeTCT design requirements. The matching impedances for CC2510 chips RF front end, PA, LNA and switches were transformed into equivalent trace width along with electrical lengths into physical lengths using AWR microwave office tx-line tool [24]. The matching impedances used in 1B9_CubeTCT design are tabulated with their equivalent trace width (W) and in following Table 4.3.

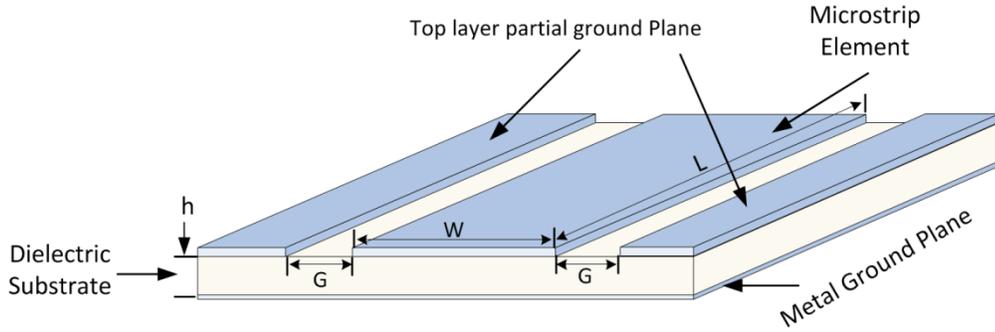


Figure 4.23: 1B9_CubeTCT PCB layout top two layers

Impedance Z_0 (Ω)	Trace Width W (mm)
15.8	3.17
24.8	1.71
50	0.6
80	0.12
95	0.10

Table 4.3: Computed trace widths with AWR Txline tool for RF front end matching

To further improve the shielding of RF traces from other neighboring components and noise sources a top layer partial ground plane is placed in vacant spaces. Also, for better isolation between the transmission and receive chain have been places at a separation of 3mm, with top layer ground plane. Also for proper impedance matching the Gap (G) of 0.3mm has been placed between the RF traces and the top layer partial ground plane. The matching network passive components selection has also been done while considering the COTS with least amount of parasitic such as ESL and ESR values that can give rise to mismatching and shifts the matching frequency.

Chapter 5

AraMiS Antenna: Design, Fabrication & Testing

5.1 Introduction

Innovative antenna design has recently become as one of most vital subsystem of Satellite Communication. In small satellites however, this subsystem is often overlooked, either due to consideration of weight or available space. Using traditional design approach is not feasible and does not offer any significant benefits. With increasing availability of COTS components in S-band and higher bands it is now possible to design Antenna and RF subsystem which comply with given requirements.

Space is a vast unexplored territory and it is important not to lose track/control of a satellite. As payloads become more demanding in terms of bandwidth requirements it is desirable to have a high data rate communication link. This has led to the development of antenna system that has acceptable gain, Wider Beamwidth and sufficient bandwidth. Therefore, such an antenna subsystem is required that can address solution for this problem. Such provision is made in the antenna design of AraMiS Project.

These antennas have been developed as part of the AraMiS project [2-3]: the basic idea behind the project is to develop a set of small separate modules that can be then combined to create bigger structures. This antenna is part of the external face of the telecommunication module (or tile, as it is assembled on the aluminum panel used also as part of the mechanical structure). Another module developed in same framework is a power management tile, designed to be put in parallel with other modules.

This chapter introduces two such antenna designs at center frequency $f_c = 2.43$ GHz for S-band communication and tracking which fulfills the link budget criteria and provides reusability for future missions:

- 5.1. Single Patch Antenna: Designed for newer CubeSat compatible AraMiS C-1 Satellites ($100 \times 100 \times 100$ mm³). It is a linearly Polarized (LP) rectangular patch antenna that can be accommodated over a single 100×100 mm² tile. It provides a wider beamwidth, adequate gain and acceptable bandwidth.
- 5.2. Patch Antenna Array: Designed for Conventional AraMiS Satellites ($160 \times 160 \times 160$ mm³). It is a right handed circularly polarized (RHCP) patch antenna array that provides acceptable gain, beamwidth and bandwidth.

Now let's have a more detailed look of these designs.

5.2 Single Patch Antenna

The AraMiS C-1 Satellite has dimension of $100 \times 100 \times 100 \text{mm}^3$ where each face has a dimension of $100 \times 100 \text{mm}^2$. This limited availability of space onboard AraMiS C-1 imposes a critical constraint on antenna design in terms of size, mass and dimension. To resolve these issues different antenna technologies and structures were considered among which patch antenna was considered suitable for our s-band antenna design.

Some of the key advantages of patch antenna include compact structure, small dimensions and low cost. Another important advantage is simple and convenient manufacturing process (compared with other antenna manufacturing technologies). It is a type of antenna that can be realized with same procedure as a PCB.

Patches, however have a few disadvantages such as low efficiency, polarization impurity, low power, spurious radiation due to power supply and difficulty to obtain high bandwidth. In the given antenna design as we are operating at a RF output power level of 2 Watts, linear polarization and an RF matching network that suppress the supply spurious radiations. The bandwidth limitation can be reduced using a thinner dielectric with lower dielectric constant. Moreover, the tradeoffs associated with our design between advantages and shortcomings are acceptable.

5.2.1 Design Specifications/Material Selection

Substrate materials considered for our design is Rogers (6002 Duroid) substrate was chosen for the design. Here the design process is explained in detail. For the design of Patch antenna following specifications were considered:

- Center operating frequency (f_c) = 2.43 GHz
- Return Loss (RL) < -10dB (for 20 MHz band)
- Gain (G_{Max}) \approx 6dB

Rogers Duroid Substrate 6002 (Tm) was selected suitable for the design that provides a relative permittivity (ϵ_r) = 2.94 and has a height (h) = 30mil/ 0.762mm.

As patch antenna characteristics are governed by its dimensions. Choosing these dimensions accurately and precisely can determine the performance of the antenna itself. As shown below in Figure 5.1, these dimensions/parameters consist of:

- Length of Patch (L)
- Width of Patch (W)
- Guided wavelength (λ_g)
- Feed Position on Patch.

Having decided on the shape of the patch we can now understand the working principle of a microstrip antenna through a comparison with the transmission lines model. In reality the microstrip itself is a guiding structure [25]. Therefore, equivalence can be made between the patch and a transmission line radiating from both ends, as shown in the following Figure 5.2.

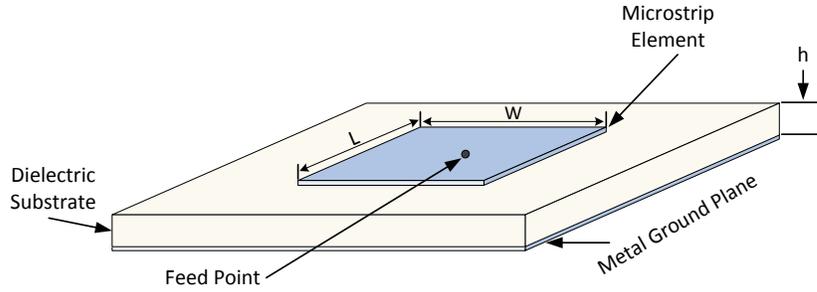


Figure 5.1: Rectangular Patch Antenna

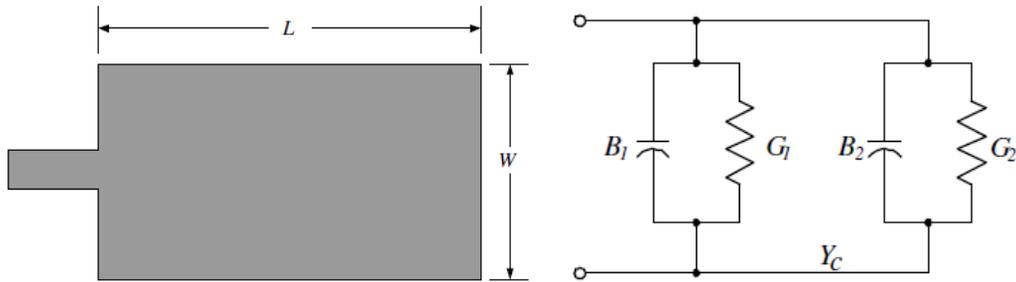


Figure 5.2: Patch and its equivalent transmission line model

In above Figure 5.2, the L is the length of the transmission line with very low characteristic impedance (Z_c). One of the main problems with the patch antennas is the fringing effect, as shown in Figure 5.3. Due to which the field lines do not travel inside the dielectric but part of it is subtended outside (i.e in air), thereby increasing the size of the Patch [26].

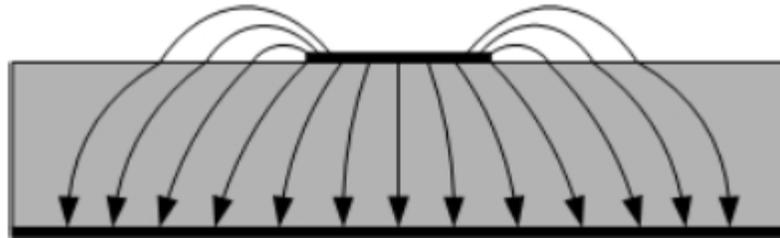


Figure 5.3: Microstrip Patch field lines under the influence of the fringing effect.

This phenomenon is related to the size of the patch and substrate, in particular to the ratio of L/h and ϵ_r . The effect of fringing can be reduced when $L/h \gg 1$. It must be kept in mind that this phenomenon can affect the resonant frequency of the patch itself. Since, the field lines are no longer confined in the dielectric substrate below the patch which varies the value of ϵ_r to effective dielectric constant (ϵ_{reff}) as defined by following equation:

$$\epsilon_{reff} = \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2} \left[1 + 12 \frac{h}{W} \right]^{-1/2} \quad (5-1)$$

The above expression (5-1) of ϵ_{reff} is valid for our design as there is only single type of dielectric layer under the patch. Thus, the ϵ_{reff} is formed by Duroid 6002 and air.

5.2.2 Design Calculations

At first the guided wavelength (λ_g) is calculated with following formula:

$$\lambda_g = \frac{\lambda_o}{\sqrt{\epsilon_r}} \quad (5-2)$$

$$\lambda_g = \frac{c}{f_c \sqrt{\epsilon_r}} = 7.14 \text{ cm} \quad (5-3)$$

Now, 'L' can be approximately calculated using following relation:

$$L \approx 0.5\lambda_g = 0.5 \frac{\lambda_o}{\sqrt{\epsilon_r}} = 3.56 \text{ cm}$$

The dimension L is the comparatively the most critical and is slightly less than $\lambda_g/2$ by an amount ΔL . The width must be kept less than a wavelength in the dielectric substrate material so that higher modes will not be excited. In our case we choose $W = 5.18 \text{ cm}$ that which is relatively less than λ_g and provides a radiation resistance of 192.72Ω . Once width is fixed we can find W , we can compute ϵ_{reff} using equation (5-1). The new guided wavelength (λ_g') is also recomputed with $\lambda_g' = 7.221 \text{ cm}$. let us now compute the change in length (ΔL) which must be removed from the patch design L to account and compensate for the fringing effect as shown in Figure 5.4 [27].

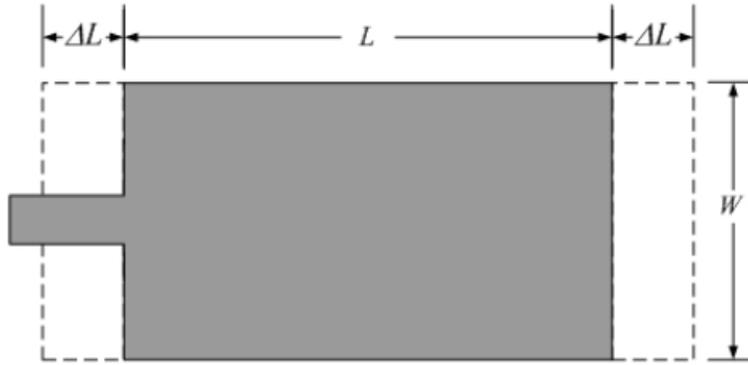


Figure 5.4: Change in length ΔL due to fringing effect.

The length ΔL can be obtained through the following relationship:

$$\frac{\Delta L}{h} = 0.412 \frac{(\epsilon_{reff} + 0.3) \left(\frac{W}{h} + 0.264\right)}{(\epsilon_{reff} - 0.258) \left(\frac{W}{h} + 0.8\right)} = 0.49593 \quad (5-4)$$

Once, ΔL is known we can compute the effective length L_{eff} as:

$$L_{eff} = L - 2\Delta L = 3.495 \text{ cm} \quad (5-5)$$

Similarly, the feed point (x) = 1.1518 cm was calculated by following relation:

$$R_{rad}(x) = \frac{R_{rad}(0)}{2} \left[1 + \cos \frac{2\pi x}{L} \right] \quad (5-6)$$

5.2.3 Simulation Analyses & Measurements

Now having all the necessary patch antenna parameters we proceed with its modeling, design and simulation using HFSS CAD tool [28]. Patch antenna 3D layout model was created as shown in

Figure 5.5. With a slight modifications of calculated patch parameters such as x and L the antenna was simulated and desirable results were obtained interms of Return Loss RL, Gain, HPBW.

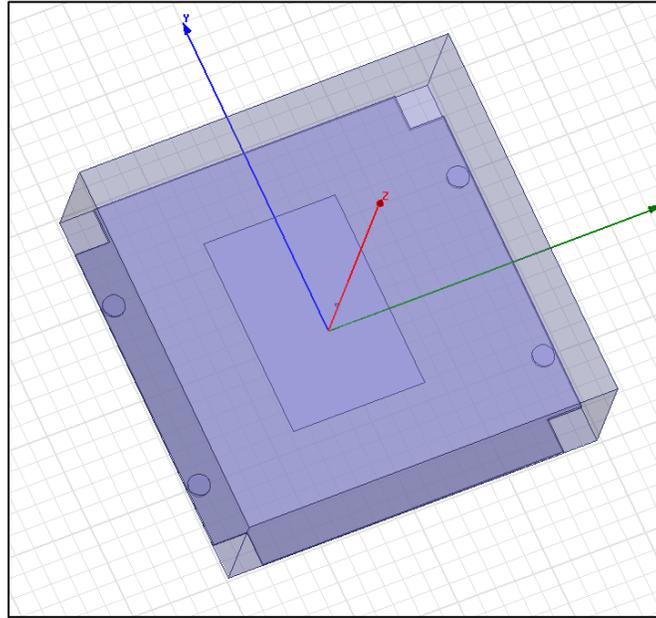


Figure 5.5: Simulated patch antenna structure in HFSS.

The simulated for patch antenna gain pattern for $\varphi = 0^\circ, 90^\circ$ and 180° were performed as reported in Figure 5.6. The 3D gain pattern, as depicted in Figure 5.7, was also computed as for which provides a more detailed visual perspective of how the gain is oriented in space around the antenna.

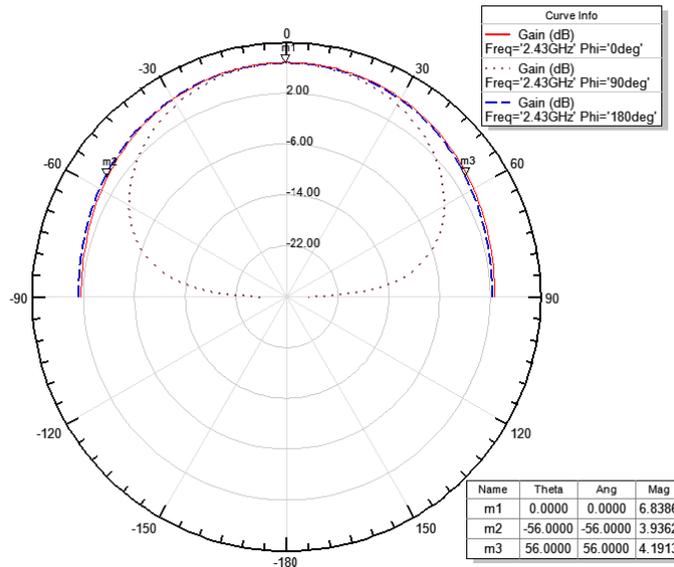


Figure 5.6: Single patch gain pattern for azimuth angles of $\varphi = 0^\circ, 90^\circ$ and 180° at f_c .

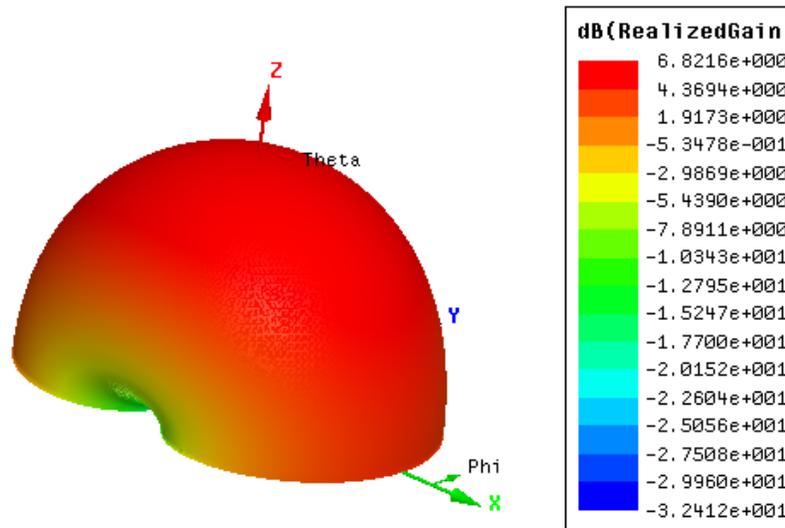


Figure 5.7: 3D gain pattern for single patch element at f_c .

The gain pattern for the antenna is broader with a HPBW of 110° in the broadside and the simulated antenna -15dB RL bandwidth of 20 MHz. The gain pattern is depicted in the above Figure 5.7. Maximum simulated gain for patch antenna is 6.8dBi at $f_c = 2.43$ GHz.

Once the simulation yielded adequate results the patch antenna was then realized on Rogers 6002 substrate as shown in Figure 5.8. The return loss (RL) was measured using vector network analyzer (VNA) and measurement results were found in close agreement with the simulated data as shown in Figure 5.9. The patch antenna *Gain* pattern was also measured inside anechoic chamber and was comparable with the simulated data as shown in Figure 5.10.



Figure 5.8: Realized patch antenna on Rogers 6002 substrate (tm)

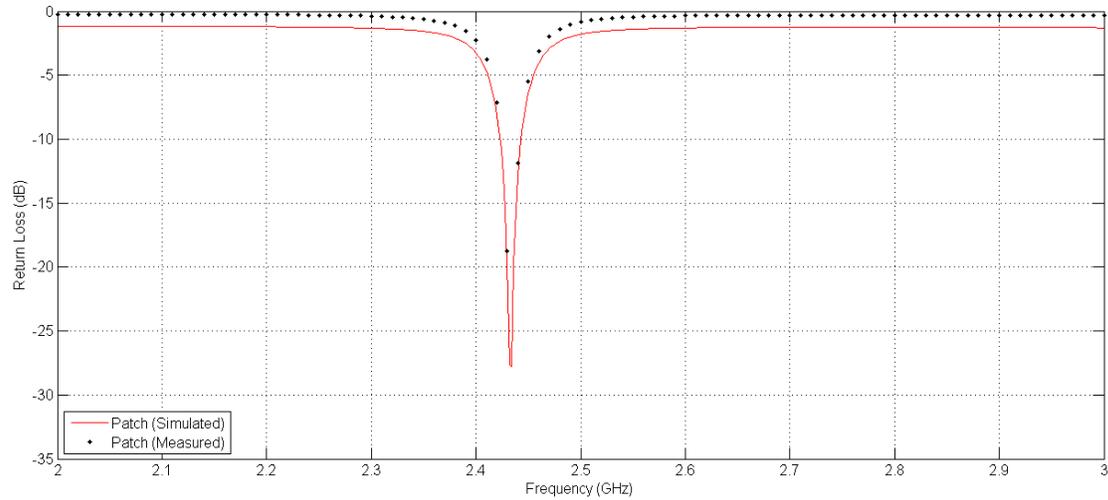


Figure 5.9: Patch Antenna RL comparison: Simulated v. Measured

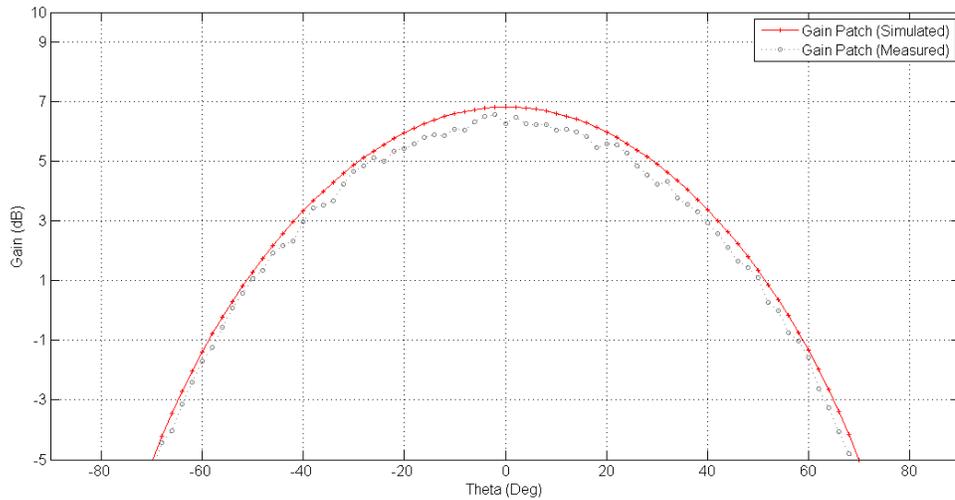


Figure 5.10: Patch Antenna Gain comparison: Simulated v. Measured

This large HPBW of patch antenna can help the satellite to keep tracking for longer instants as it has a broader horizon to sight a particular ground station even if having an acute orientation at that instant.

5.3 Patch Antenna Array

The antenna array consists of a four (2x2) patch array antennas each fed with a 90° phase shift with respect to adjacent patch. All these patch array elements together provide a circularly polarized (CP) electromagnetic wave. As, satellite maintains its rotational and translational motion in the orbit, circular polarization of electromagnetic wave is employed to ensure a more stable communication link. Using such configuration of patch arrays makes it possible to create a CP electromagnetic wave with efficient utilization of available area and also making it low cost. It has comparatively higher gain and provides a narrower beamwidth.

5.3.1 Theoretical Overview

In arrays circular polarization is conventionally realized by using CP radiating elements where each element generally requires two feed ports with a hybrid coupler to obtain the required two orthogonal polarizations and a 90° phase differential [29]. For wider bandwidth applications, often four feed ports with 0° , 90° , 180° and 270° phase differentials are employed [30].

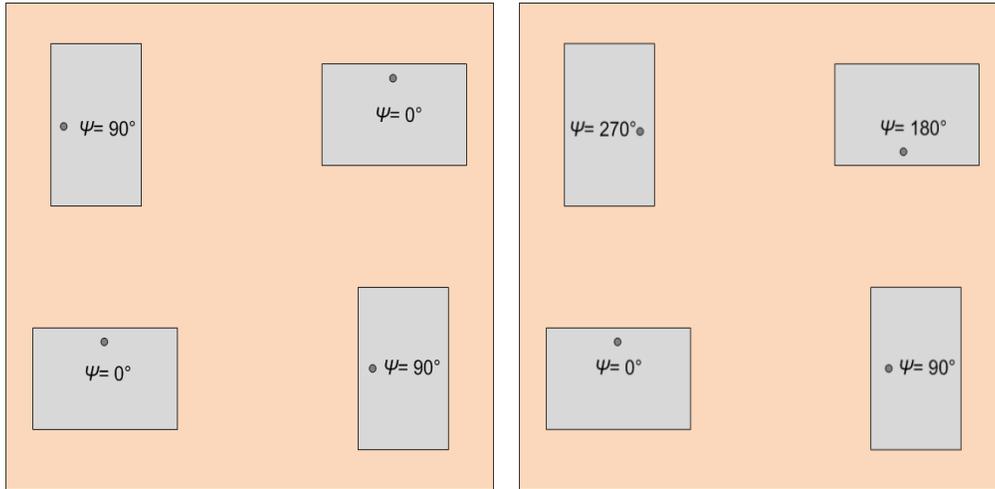


Figure 5.11: 2x2 grid configuration of four LP Patches (a) $0^\circ, 90^\circ, 0^\circ$ & 90° ; (b) $0^\circ, 90^\circ, 180^\circ$ & 270°

The CP patch array can also be realized by placing the radiating elements each linearly polarized (LP) in a particular manner in terms of orientation and phase shift. For instance CP is generated in a simple case of four LP patch elements by arranging them on a 2x2 square grid configuration with all 4 element having adjacent phase shift of 0° , 90° , 0° and 90° respectively; introduced by feeding network as shown in Figure 5.11(a). Final phase shift of 0° , 90° , 180° and 270° for the four radiating elements is obtained by proper orientation of each element as shown in Figure 5.11(b). It is accomplished by placing diagonal elements on array grid inverted to each other. This ensures excitation of adjacent patch elements with 90° phase delay which creates two orthogonal LP that collectively produce CP. Such angular orientations are also useful in suppressing inter-element coupling and cross polarization [31].

5.3.2 Array Structure Design

The array structure comprises of four patches placed in 2x2 grid with inter-element spacing of $d = 0.428\lambda$ (52.5 mm) from the adjacent element as shown in Figure 5.12. As the available space on the external face of AraMiS satellite is quite limited (less than 150 mm x 150 mm) which makes it difficult (in terms of size and input impedance matching) to use conventional rectangular patches as radiating elements for the array structure design. Thus, specially optimized butterfly patch structures are selected with dimension $L = 25.4$ mm and $W = 10$ mm of each arm as shown in the Figure 5.13. This particular choice of patch structure helps to minimize input impedance ' Z_{in} ' from 400Ω in case of conventional rectangular patch to 260Ω by dividing the Patch into two high impedance strip lines. It also helps in confining and accommodating the array structure over the limited available space on board AraMiS satellite [32].

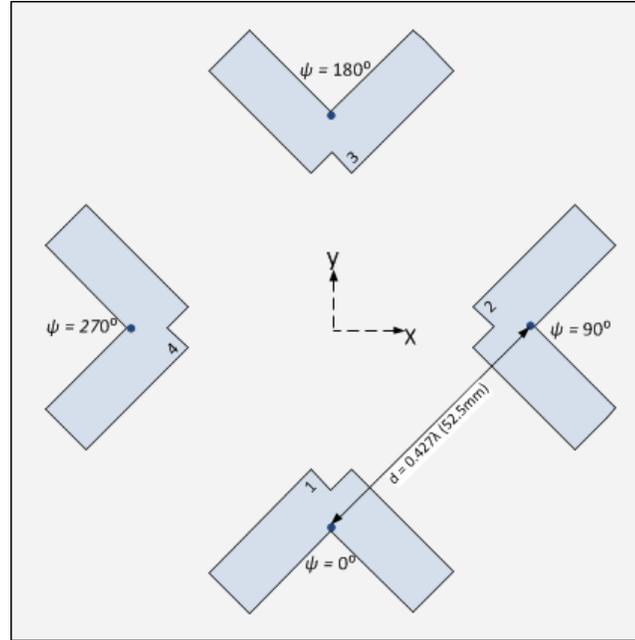


Figure 5.12: 2x2 Patch array with 0° , 90° , 180° & 270° element spaced at 0.428λ (52.5 mm)

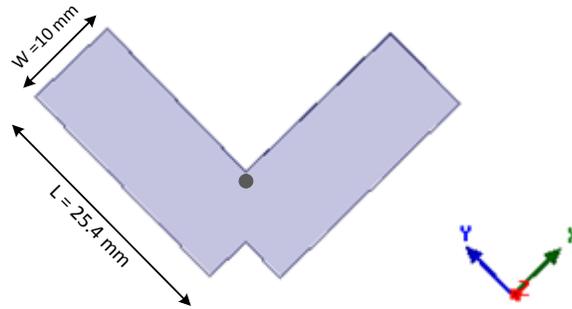


Figure 5.13: Single Patch Element

RT/Duroid 6006 substrate ($\epsilon_r = 6.15$ and thickness $t = 3.8$ mm) has been selected for fabrication of both array elements and feeding network to provide a bandwidth of over 30 MHz at f_c . This selection of patch element dimension together with employed substrate makes array structure highly suitable for our application.

Each patch element is fed 90° out of phase by a linear phase shifted BFN structure that is realized over the same microstrip substrate together with the antenna elements as shown in the Figure 5.14. The BFN consists of three 3dB Wilkinson Bridge couplers and a series of quarter wave length transformers tuned at f_c . A coaxial SMA connector feeds the BFN. As in the first stage 3dB Wilkinson Bridge coupler splits the power equally into two branches of microstrips which are further divided equally in second stage of BFN by two more 3dB Wilkinson Bridge couplers [33].

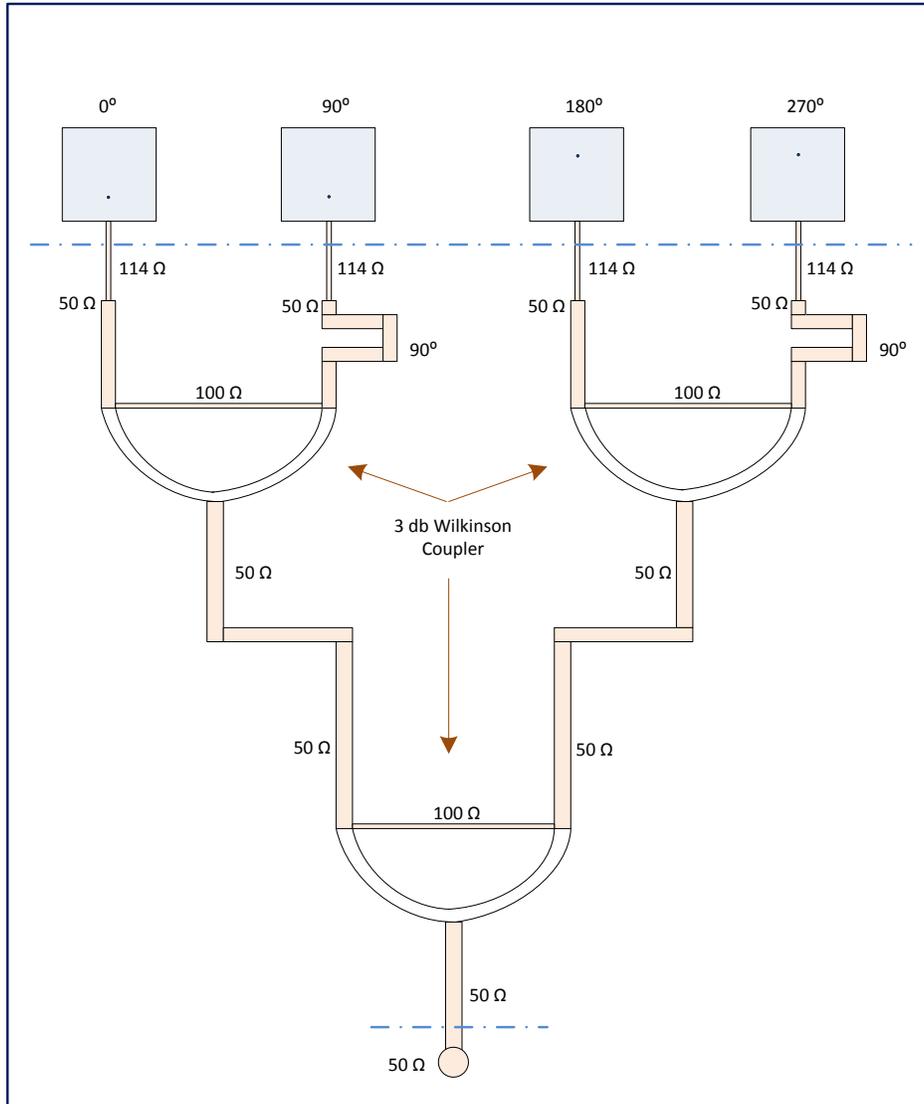


Figure 5.14: Block View of Patch Array BFN

A 90° phase shift is introduced by extension of microstrip physical length as shown in Figure 5.14. The BFN feeds each patch with a phase shift of 0°, 90°, 0° and 90°. In order to obtain a further phase shift of elements as 0°, 90°, 180° and 270°, patch elements are oriented such as each patch is inverted with respect to the opposite diagonal patch element over the array grid. This particular orientation of the array structure provides an additional benefit of cancelling inter-element coupling between the patch elements [31].

5.3.3 Array Structure Concept

The patches are oriented in a particular angular arrangement, in order to create two orthogonal polarized fields. Different feed phases are employed to further create the required phase delays for CP generation. CP field can be attained in broadside direction of an array by using linearly polarized elements with arranging them in the following manner with angle and phase arranged in a 0°, 90° (as shown in Figure 5.15) [34].

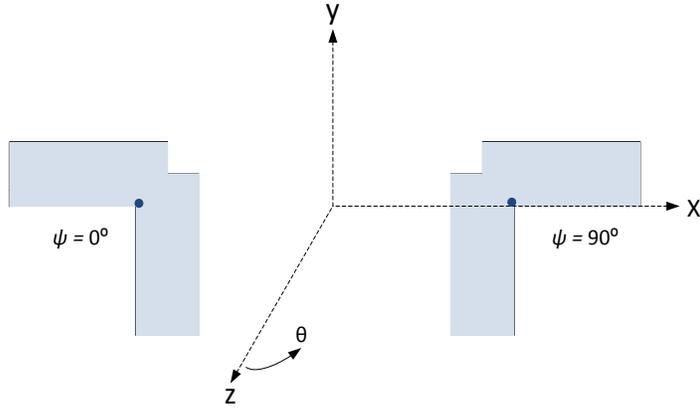


Figure 5.15: Two element microstrip array with 0° and 90° arrangement.

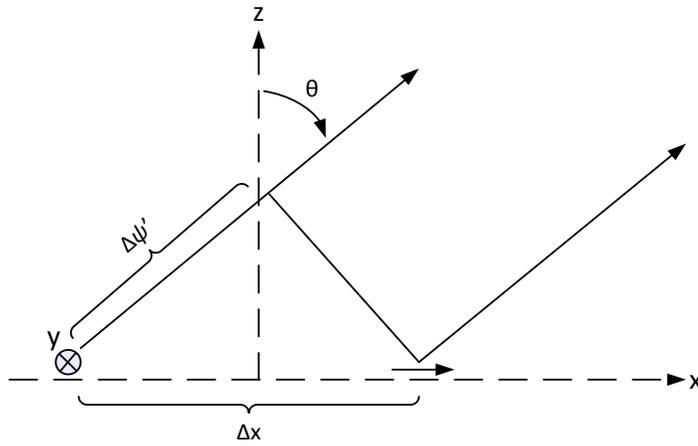


Figure 5.16: Two-element microstrip array: Spatial phase delay $\Delta\psi' = k_0\Delta x \sin \theta$.

The CP produced in this manner deteriorates at angles larger than 10° from the broadside direction in the x - z plane as depicted in Figure 5.17. The spatial phase delay ($\Delta\psi' = k_0\Delta x \sin \theta$, see Figure 5.16), is mainly caused by that is created between the two orthogonally polarized array. This spatial phase delay affects the required 90° phase differential which results in poor CP quality. This spatial phase delay can be resolved by using a 2×2 array as shown in Figure 5.11(a). It is due to the fact that within the two principal planes, the spatial phase delay in one array dimension is opposite to that of the other and hence they cancel each other. Figure 5.11(b) depicts an arrangement that provides considerable polarization improvement in comparison to figure. By symmetry, the two principal plane patterns ($\varphi = 0^\circ$ and 90°) are identical to each other. Therefore, only one is shown in following Figure 5.17.

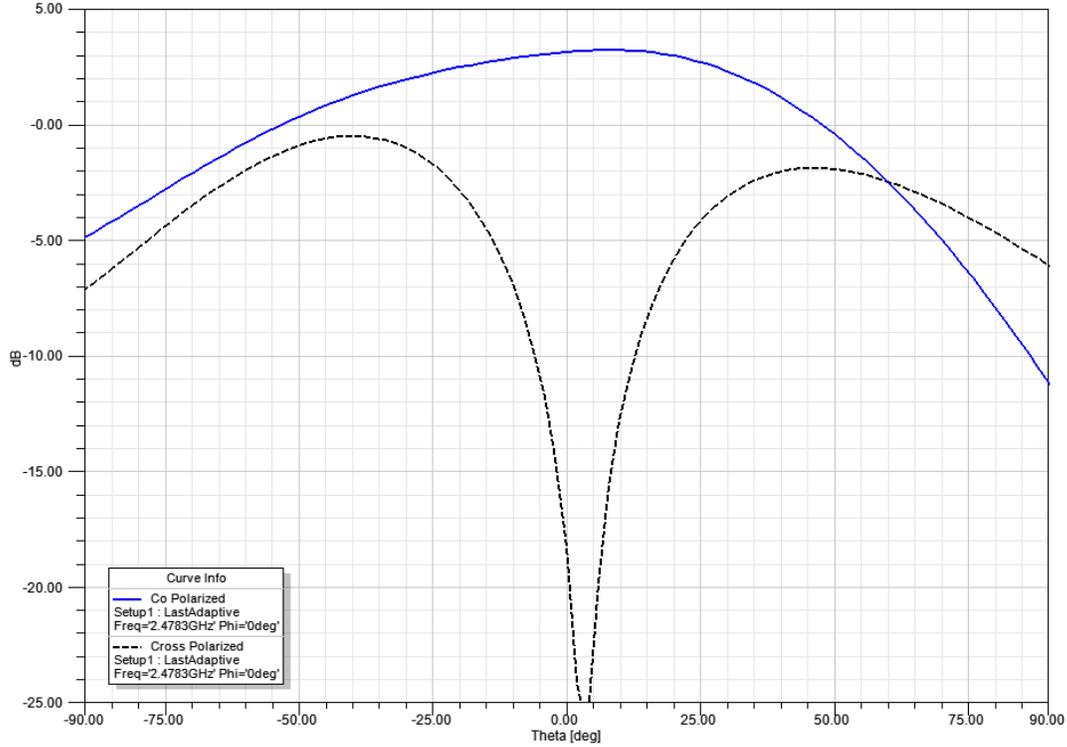


Figure 5.17: CP patterns of the two-element array shown in Figure 2 with spacing 0.428λ

This behavior of 2×2 array can be theoretically explained by considering Figure 5.19. The far field patterns in the two principal planes (the x - z plane or the y - z plane) can be calculated by directly adding the individual fields from all the four patch elements. If the total far field in the x - z plane can be denoted by \mathbf{E} with the horizontal field component for patch 1 is \mathbf{E}_{H1} and the vertical field component from patch 2 by \mathbf{E}_{V2} . Similarly, for patch 3 and 4 have contributing field components denoted as \mathbf{E}_{H3} and \mathbf{E}_{V4} respectively. Then, the total far field can then be evaluated as follows:

$$E_{xz} = E_{H1}e^{-jk_0d \sin \theta} e^{j0^\circ} + E_{V2}e^{jk_0d \sin \theta} e^{j90^\circ} + E_{H3}e^{jk_0d \sin \theta} e^{j0^\circ} + E_{V4}e^{-jk_0d \sin \theta} e^{j90^\circ} \quad (5-7)$$

$$E_{xz} = (E_{H1}e^{j0^\circ} + E_{V4}e^{j90^\circ})e^{-jk_0d \sin \theta} + (E_{H3}e^{j0^\circ} + E_{V2}e^{j90^\circ})e^{jk_0d \sin \theta}$$

Since, all array elements are excited uniformly ($E_H = E_{H1} = E_{H3}$ and $E_V = E_{V2} = E_{V4}$);

Therefore:

$$E_{xz} = (E_H e^{j0^\circ} + E_V e^{j90^\circ}) (e^{-jk_0d \sin \theta} + e^{jk_0d \sin \theta}) \quad (5-8)$$

$$E_{xz} = (E_H e^{j0^\circ} + E_V e^{j90^\circ}) 2 \cos(k_0d \sin \theta)$$

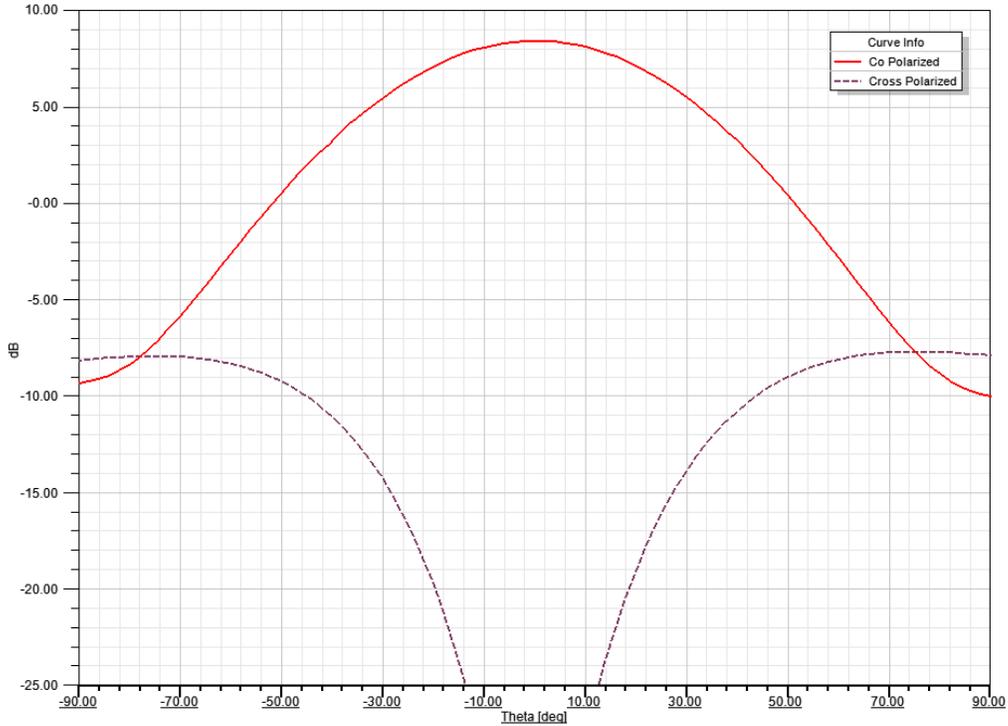
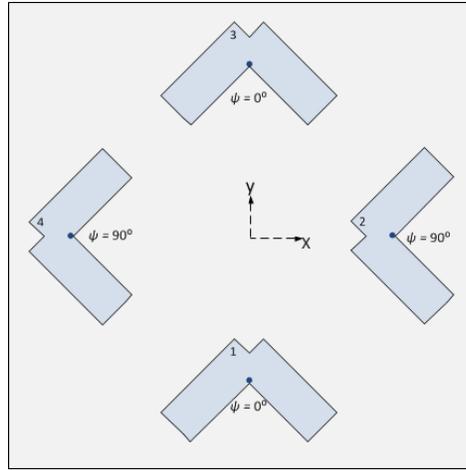


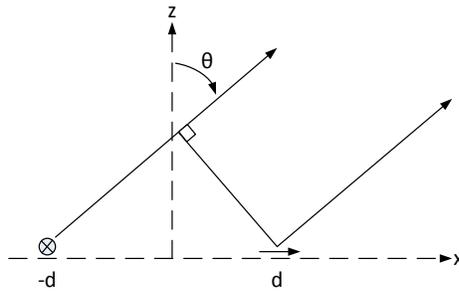
Figure 5.18: Calculated principal plane ($\varphi = 0^\circ$ or 90°) CP patterns of the 2×2 array shown in Figure 5.12 with adjacent element spacing of 0.428λ .

Above equation (5-10) represent a circularly polarized wave with the cosine term is a two-element array factor. This total field \mathbf{E} is equivalent to that generated by two CP elements. The radiated far field is basically equivalent to that generated from two circularly polarized elements created by pairing of the top row elements with the bottom row elements. Using a relatively thicker substrate for 2×2 microstrip array arranged in the $0^\circ, 90^\circ, 180^\circ, 270^\circ$ fashion for both its element orientations and feed phases as shown in Figure 5.20(b) the axial ratio bandwidth of the array can be increased substantially. Its due to the fact that most of the undesired radiation (due to higher order modes of the thick substrate) from the 0° element cancels that from the 180° element, and likewise for the 90° and 270° elements. This behavior is graphically depicted in Figure 5.20. The solid arrows represent the field at the edges due to fundamental TM_{10} mode which creates the co-pol radiation.

The dotted arrows denotes field at the edges due to TM_{02} mode which is depicted here as an illustration of the many modes that contribute to cross-pol. Figure 5.20 (a) shows the $0^\circ, 90^\circ, 0^\circ, 90^\circ$ arrangement in which both co-pol and cross-pol fields from the two diagonal elements support each other. On the contrary to this arrangement, Figure 5.20 (b) the co-pols are sums up but the cross-pols gets cancelled. As a result of this cross pols cancellation we are allowed having wider axial ratio frequency behavior [31].



(a)



(b)

Figure 5.19: Geometrical design layout for array derivation

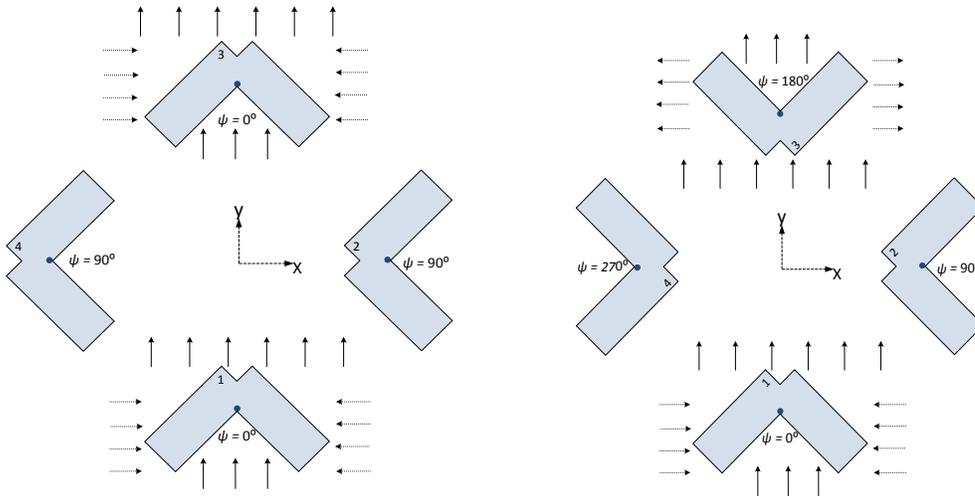


Figure 5.20: (a) $0^\circ, 90^\circ, 0^\circ, 90^\circ$ arrangement depicting cross-pols support each other. (b) $0^\circ, 90^\circ, 180^\circ, 270^\circ$ arrangement describing cross-pols cancel each other effect. Solid arrows represent co-pols, dashed arrows represent cross-pols.

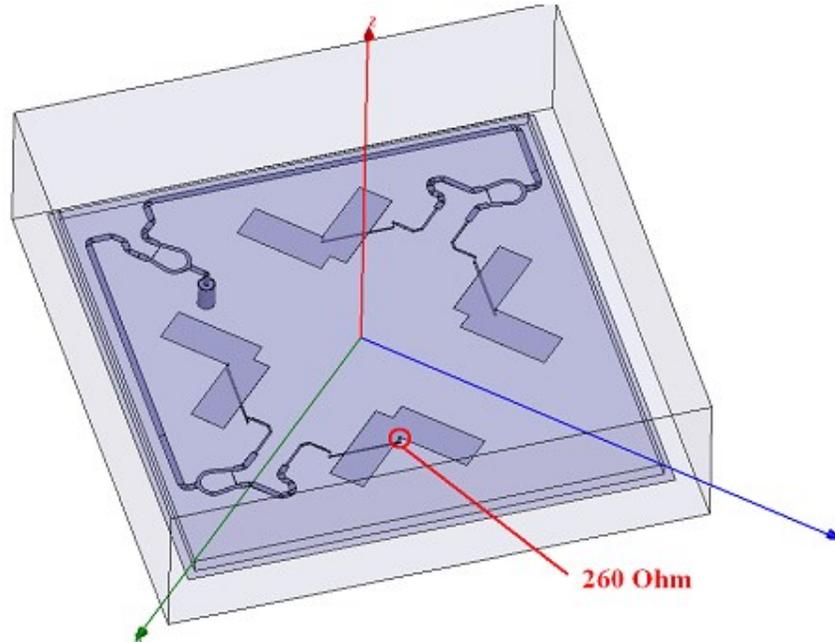


Figure 5.21: Simulated patch array structure in HFSS.

5.3.4 Simulation Analyses & Measurement

Once the parameters for array design are known, a series of simulations were performed using Ansoft HFSS electromagnetic CAD simulation tool [8]. As a primary step a 3D CAD model (according to scale) for array structure was developed in HFSS (as shown in Figure 5.21) and simulated within the frequency bandwidth of 2 GHz to 3 GHz. Different antenna parameters such as return loss (RL) in dB, Gain pattern (in dBi) for certain azimuth planes of $\varphi = 0^\circ, 90^\circ$ and 180° at f_c were computed for patch array as depicted in Figure 5.23. The simulated patch array provides a maximum gain of 8.14 dBi at f_c which tends to reduce by -3dB gradually from the broadside till 24° on either side. Thus, the array computed half power beamwidth (HPBW) of 48° as depicted in Figure 5.23.

To observe the far field pattern of the array in more detail a 3D Gain pattern was also computed and shown below in Figure 5.24. As the gain is concentrated in the main lobe and there are no side or back lobes so most of the power is transmitted and/or received within the desired beam with very few overspill in other directions.

Once the simulated array yielded adequate results the patch antenna was then realized on Rogers 6006 (tm) substrate ($\epsilon_r = 6.15$ and thickness $t = 3.8$ mm) as shown in Figure 5.22. The return loss (RL) was measured using vector network analyzer (VNA) and measurement results were found in close agreement with the simulated data as depicted in Figure 5.25. The patch antenna Gain pattern was also measured inside anechoic chamber and was comparable with the simulated data as depicted in Figure 5.26.

5.3 - Patch Antenna Array



Figure 5.22: Realized patch array antenna on Rogers 6006 substrate (tm)

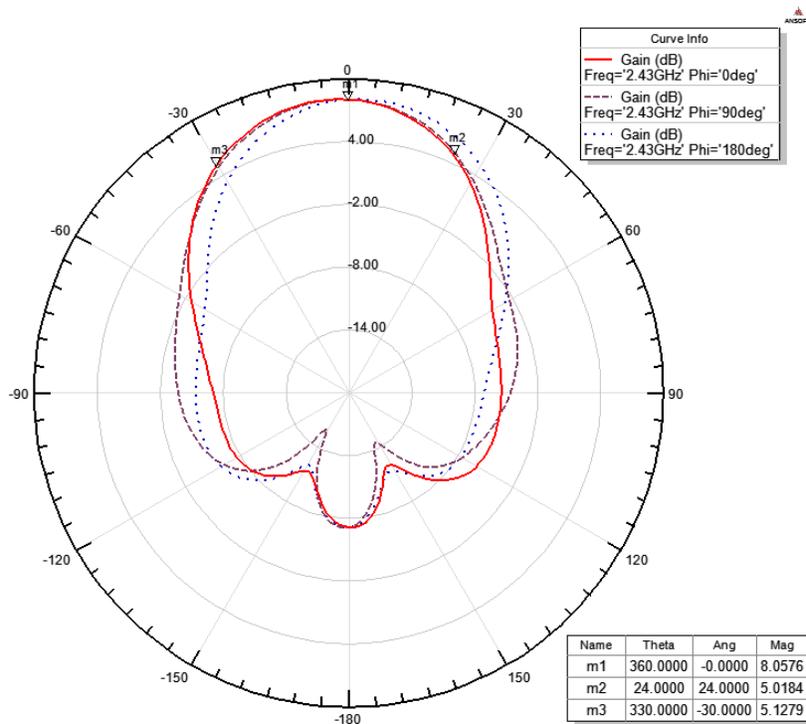


Figure 5.23: Patch array gain pattern for azimuth angles of $\varphi = 0^\circ, 90^\circ$ and 180° at f_c

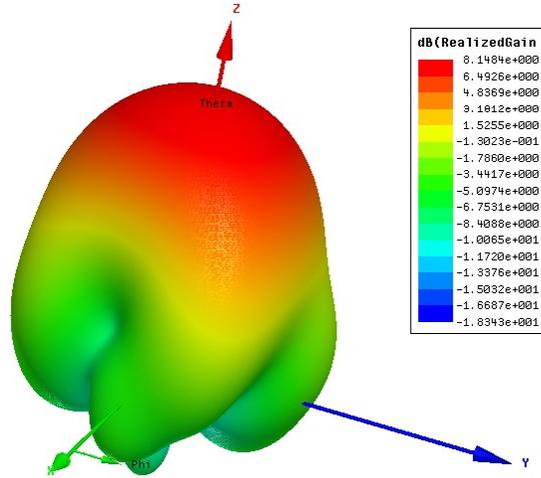


Figure 5.24: 3D gain pattern for patch array at f_c

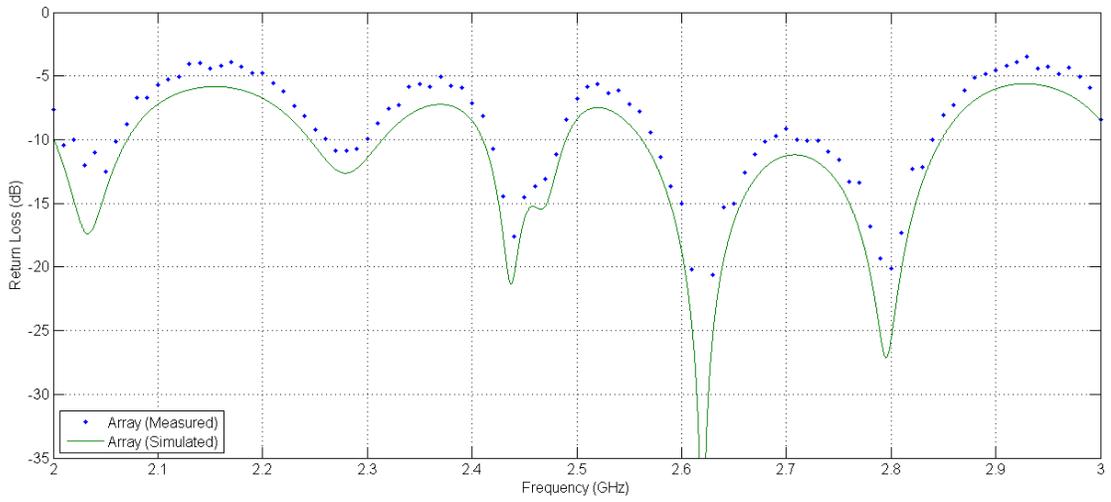


Figure 5.25: Patch Array RL comparison: Simulated v. Measured

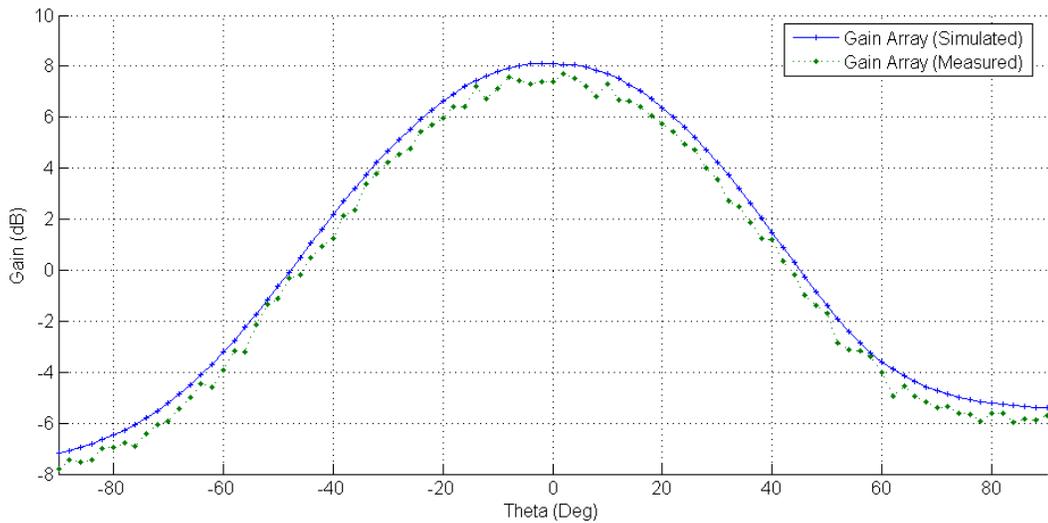


Figure 5.26: Patch Array Gain comparison: Simulated v. Measured

The BFN designed for patch array has been considerably well matched. It is evident from the fact that the RL is less than -15dB for a bandwidth of 45MHz around the center frequency with minimum in-band RL = -20.5dB. Similarly, the RL for single patch is lower than -15dB for a bandwidth of 20MHz around the f_c with minimum in-band RL=-28.5dB. Figure 5.27 provides a comparison of simulated and measured RL for both patch array and single patch. Similarly, the Figure 5.28 provides a comparison of simulated and measured RL for both patch array and single patch.

The bandwidth measured for single patch is relatively lower than for the patch array but at the same time HPBW for patch antenna is relatively high or almost twice. This means interms for performance that patch antenna can maintain radio communication link with even poor antenna orientation towards the Ground station at a relatively lower gain then the patch array.

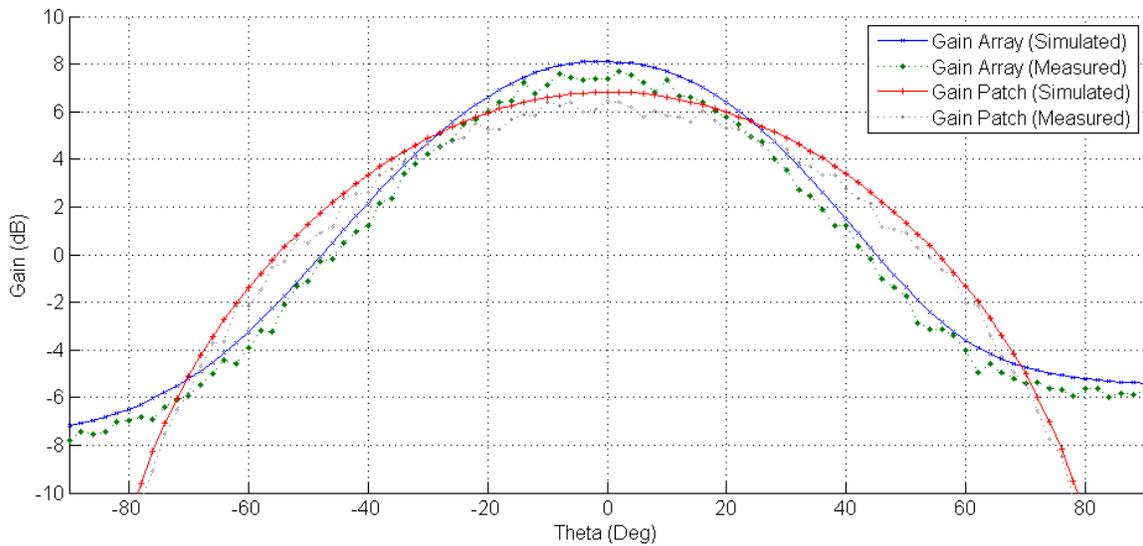


Figure 5.27: Gain comparisons for single patch vs. patch array at f_c

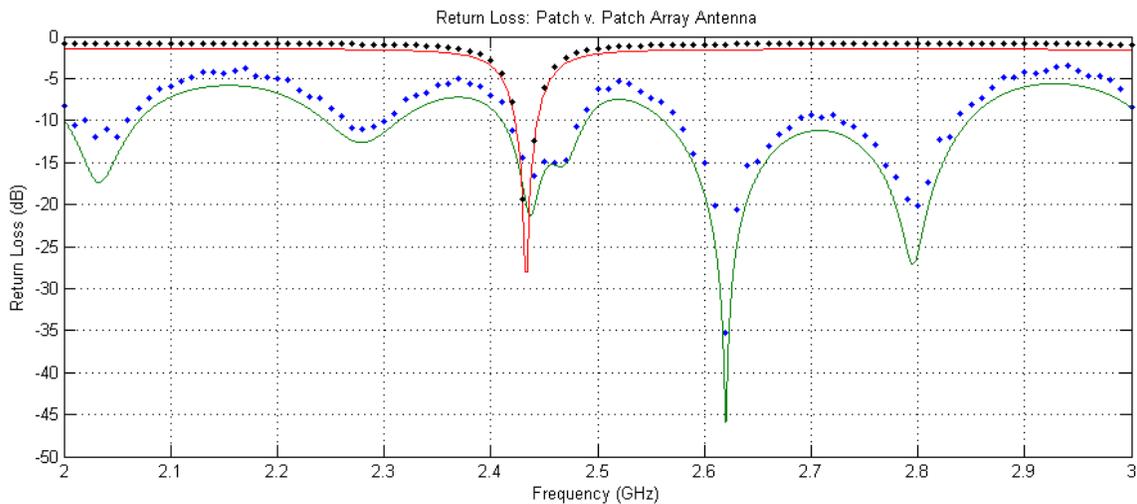


Figure 5.28: RL comparison for single patch and patch array

Antenna Type		Gain Max. (dBi)	HPBW (Deg)	RL (dB)	-15dBBW (MHz)
Single Patch CubeSat (LP)	Simulated	6.8	76	-28.5	20
	Measured	6.2	74	-26	18
Patch Array AraMiS (RHCP)	Simulated	8.11	42	-20.5	50
	Measured	8.1	40	-18	45

Table 5.1: Measure v. simulated parameters for both Patch and Array antennas

5.4 Conclusion

The above table summarizes the significant characteristics of patch antenna and AraMiS array. It also reveals that they are in compliance with the expected link budget requirements for them. They yield considerable link budget margins and can support the S-band communication data rate of upto 500kbps. Though the single patch array is a LP antenna design still its gain is acceptable and has a larger HPBW which means it has a flexible pointing profile with any given ground station and thus maintaining a more line of sight link in each orbit revolution around earth.

The array antenna which is RHCP has been designed for a high bandwidth communication link. Although having a narrower HPBW it provides a wider bandwidth and relatively higher gain which can be useful for missions that requires high bandwidth communication.

Chapter 6

S-band Link Budget Estimation for AraMiS

Satellites

The linkbudget is the most important step in the feasibility of a telecommunication system design. It is even more essential in case of Satellite communication which takes into account many kinds of variable that can affect the radio communication link. The Link budget criteria for the AraMiS satellites have been carefully evaluated. Before moving further on the link budget criteria lets first discuss the available ground station capability of the (AraMiS) Polito ground station.

6.1 Polito Ground Station

6.1.1 Introduction

The ground station at Politecnico di Torino is a GENSO specifications compatible radio communication system that can communicate with orbiting satellites in the given radio amateur satellite service bands [35]. It includes two separate antenna systems in the VHF/UHF and the S-band as depicted in Figure 6.1. It also consists of the electronic and radio communication equipment and a PC for ground station control and communication. It also runs the application that connects the ground station to the GENSO.



Figure 6.1: Antennas of the Politecnico Ground Station

Figure 6.2 presents a block diagram of Polito ground station. It consists of many sub blocks and operates in VHF, UHF and S-band. Here is a briefly overview on the S-band communication system of the Polito ground station and its associated attributes which are important interms of link budget estimation. This includes following features:

- Parabolic Antenna (F/D=0.3, G= 35dBi. RHCP polarization)
- PA: Pout=25W
- LNA: 16dB gain, 0.4dB NF
- RF feeder: (50Ω, attenuation ≈ 1.3 dB)
- EIRP = 45 dBW

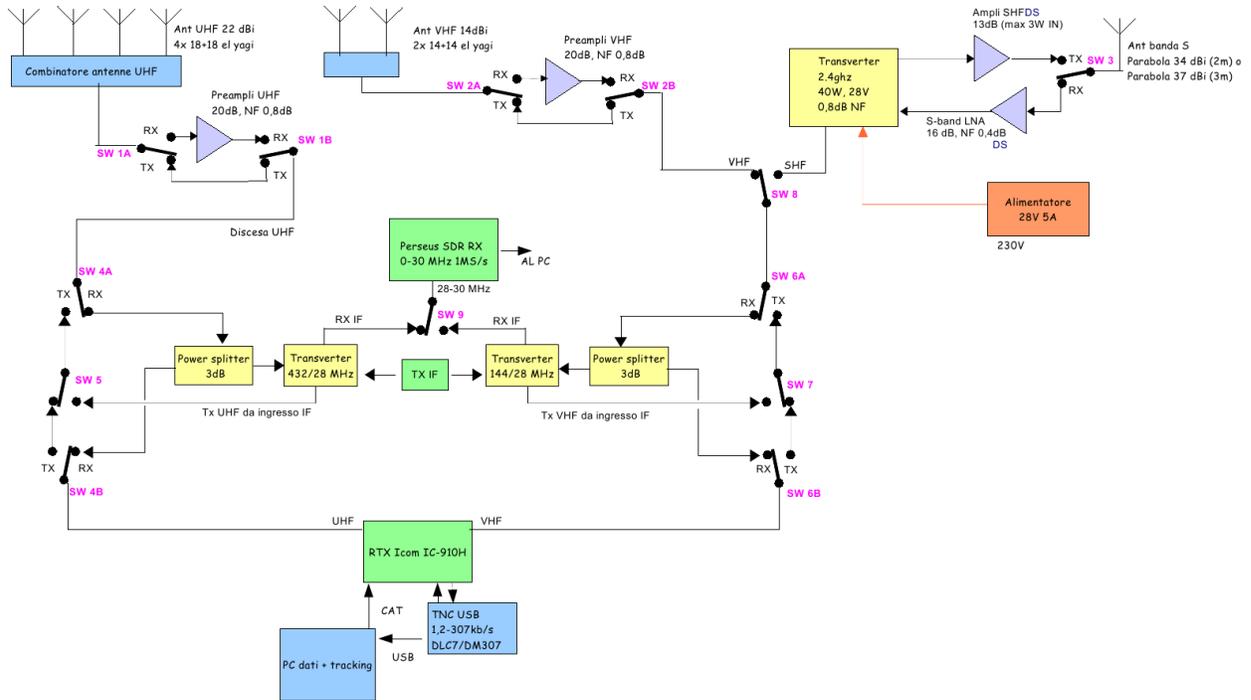


Figure 6.2: Block scheme of the Polito ground station

6.1.2 Link Budget Calculation

In this section we calculate the link budget in the uplink and in downlink. In order to characterize interms of BER of the transmission quality. The calculation of the link budget is based on the equation (6-1) of the transmission [36-38]:

$$P_{Rin} = P_T G_T \frac{1}{\alpha_o \alpha_p \alpha_m} \quad (6-1)$$

Whereas:

- P_{Rin} : input power level at the input of receiver
- P_{T} : transmitted power
- G_{R} and G_{T} gains of the antennas in reception and in transmission, respectively
- α_{o} : attenuation of propagation in free space
- α_{p} : polarization loss
- α_{m} : medium loss (accounts for atmospheric and ionospheric losses)

The expression (6-2) in logarithmic form becomes:

$$P_{\text{Rin}}|_{\text{dBm}} = P_{\text{T}}|_{\text{db}} + G_{\text{T}}|_{\text{dB}} - \alpha_{\text{o}}|_{\text{dB}} - \alpha_{\text{p}}|_{\text{dB}} - \alpha_{\text{m}}|_{\text{dB}} \quad (6-2)$$

Consider a satellite equipped with single patch antenna with gain 6dBi communicating with Polito ground station with parameters already defined in previous section. For the antenna reception we further assume attenuation for polarization loss of 3dB. Whereas the Figure 6.3, by Carnot's theorem we can calculate the distance R of the satellite as a function of elevation and of the altitude by the following expression (6-3):

$$R = -R_{\text{E}} \sin \theta_{\text{EL}} + \sqrt{R_{\text{E}}^2 (\sin^2 \theta_{\text{EL}} - 1) + R_{\text{SV}}^2} \quad (6-3)$$

With;

- R_{E} : radius of the earth
- θ_{EL} : horizon elevation
- R_{SV} : remote satellite and the center of the earth
- h : mean height of satellite from earth's surface

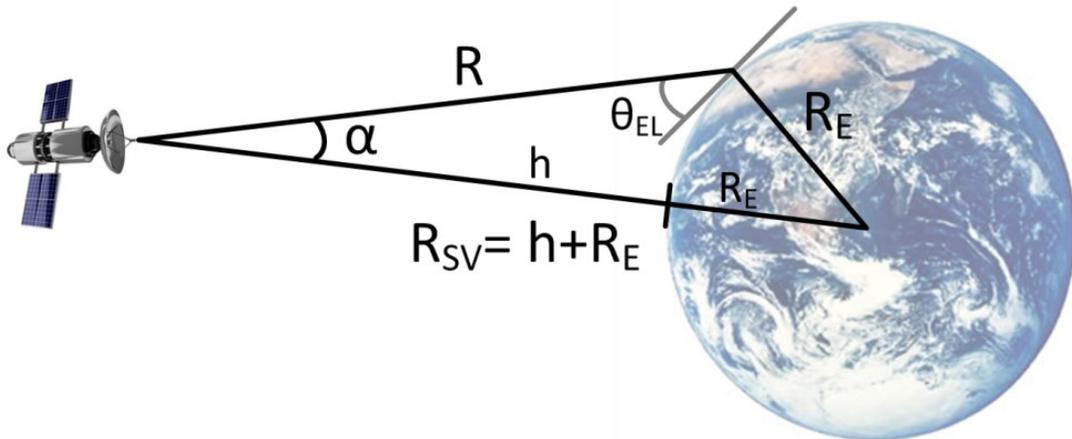


Figure 6.3: Distance and elevation of the satellite as a function of altitude

Recall that the LEO orbits are between 160 and 2000 km altitude. For our calculations we consider an altitude of 800km and elevation for the worst case, that is, $\theta_{\text{EL}}=10^\circ$. In these conditions R can be evaluated by (6-3) and the path loss is calculated using (6-4):

$$\alpha_o = \left(\frac{4\pi R}{\lambda} \right)^2 = 167.7 \text{ dB} \quad (6-4)$$

Finally, we can now estimate the received power P_R by substituting (6-3), (6-4) in (6-2). This received power is then used to compute the Figure of Merit for the Receiver using following (6-5)

$$\left. \frac{G}{T} \right|_{\text{dB/K}} = G_{R|\text{dB}} + 10 \log(T_a + T_{\text{eq}}) \quad (6-5)$$

where:

- T_a : noise temperature of antenna
- T_{eq} : equivalent noise temperature of receiver

The Noise temperature calculated for ground station antenna is evaluated as in (6-6):

$$T_a = \frac{(L-1)290 + T_{\text{sky}}}{L} \quad (6-6)$$

Where:

- T_{sky} : effective sky temperature
- L : Transmission line coefficient (not in dB)

To measure the carrier to noise ratio E_b/N_0 we can use the following relation (6-7):

$$\left. \frac{E_b}{N_o} \right|_{\text{dB}} = P_{R|\text{dBm}} + \left. \frac{G}{T} \right|_{\text{dB/K}} + K_{|\text{dBW/HzK}} - 10 \log B_{|\text{dBHz}} \quad (6-7)$$

Where:

- E_b/N_o : signal to noise ratio
- K : represents Boltzmann's Constant (-228.6 dBW/HzK)
- B : bandwidth in hertz

Thus the computed E_b/N_o is 14.8 dB. When using GFSK with 1dB implementation loss and a Bit error rate (BER) of $1.00e-5$, the estimated link margin is 1.4 dB for downlink communication in S-band for the worst case scenario of path loss ($\theta_{\text{EL}}=10^\circ$). Following Table 6.1 summarizes the calculated link budget estimates.

S-band Downlink BUDGET AraMiS-C1 (Polito GS)		
Satellite	Transmitter Power Output	33 dBm
	Transmission Line Losses	-2.0 dB
	Antenna Gain	6.2 dBi
	EIRP	8.7 dBW
	Antenna Pointing Loss	-1.0 dB
	Antenna Polarization Loss	-3.0 dB
Radio Link Path	Path Loss ($\theta_{EL} = 10^\circ$)	-167.7 dB
	Atmospheric Loss	-0.3 dB
	Ionospheric Loss	-0.2 dB
	Ground Station Isotropic Signal Level	-163.5 dBW
Ground Station	Antenna Pointing Loss	-2.0 dB
	Antenna Gain	35.0 dBi
	Transmission Line Losses	-1 dB
	LNA Noise Temperature	125 K
	Transmission Line Temp	290 K
	Sky Temperature	200 K
	Transmission Line Coefficient	0.7943
	Effective Noise Temperature	344 K
	Figure of Merit (G/T)	8.6 dB/K
	S/N ₀ Power Density	71.8 dBHz
	System Desired Data Rate	500 Kbps
		57.0 dBHz
Telemetry System	E _b /N ₀ (~Spectral efficiency 1bps/Hz)	14.8 dB
	Required BER(Assume G3RUH FSK; N ₀ Coding; 1dB Implementation Loss)	1.00E-05
	Required E _b /N ₀	13.35 dB
System Link Margin		1.4 dB

Table 6.1: S-band Downlink budget for AraMiS-C1 computed for Polito ground station

For the uplink communication the link budget can be estimated in similar manner for the worst case scenario which can be summarized in the given Table 6.2. The link margins estimated in this case is 7.1 dB with an elevation angle ($\theta_{EL} = 10^\circ$).

S-band Uplink BUDGET AraMiS - C1 (Polito GS)		
Ground Station	Transmitter Power Output	44 dBm
	Transmission Line/ Sw Losses	-2.0 dB
	Antenna Gain	35 dBi
	EIRP	45.0 dBW
	Antenna Pointing Loss	-2.0 dB
Radio Link Path	Path Loss (Elevation angle =10°)	-167.7 dB
	Ionospheric Loss	-0.2 dB
	Atmospheric Loss	-0.3 dB
	Isotropic Signal Level	-129.2 dBW
Satellite	Antenna Pointing Loss	-1.0 dB
	Antenna Polarization Loss	-3.0 dB
	Antenna Gain	6.0 dBi
	Transmission Line Losses	-2.0 dB
	LNA Noise Temperature	500 K
	Transmission Line Temp	270 K
	Sky Temperature	290 K
	Transmission Line Coefficient	0.7943
	Effective Noise Temperature	788 K
	Figure of Merit (G/T)	-24.0 dB/K
	S/No Power Density	77.4 dBHz
	System Desired Data Rate	500 Kbps
		57.0 dBHz
	Telemetry Sys.	E_b/N_o (~Spectral efficiency 1bps/Hz)
Required BER(Assume G3RUH FSK; No Coding; 1dB Implementation Loss)		1.00E-05
Required E_b/N_o		13.4 dB
System Link Margin		7.1 dB

Table 6.2: S-band Downlink budget for AraMiS-C1 computed for Polito ground station.

The measured link margin is an indicator of the communication link and the amount of losses that can be endured during communication. The system link margins tend to improve as the satellite start to move from the horizon towards the ground station and at the zenith it is the maximum value. The following Table 6.3 depicts the estimated link margins for different values of θ_{EL} together with the corresponding estimated α_0 .

Link Margins (dB)		Path Loss (α_0)	El. Angle (θ_{EL})
Uplink	Downlink		
6.3	3.9	-165.2 dB	20 °
8.4	6.0	-163.1 dB	30 °
10.7	8.3	-160.8 dB	45 °
12.1	9.7	-159.4 dB	60 °
13.2	10.8	-158.3 dB	90 °

Table 6.3: Link Margins estimated for different θ_{EL} and there corresponding α_0 .

As the AraMiS-C1 S-band communication subsystem (in downlink) must also be compliant with GENSO ground stations. It is important to estimate linkbudget for a most generic GENSO ground stations. An online survey was conducted inorder to have a general estimate of their S-band communication link parameters such as antenna gain (around 20-25 dB) and ground station losses (1-2 dB). It was observed that at a lower data rate (100-120 kbps) and elevation angles larger than 50°, lead to acceptable link margins as reported in the Table 6.4.

Ground Station Ant. Gain (G_R)	Transceiver Data Rate	Link Margins (dB)	
		$\theta_{EL}= 50^\circ$	$\theta_{EL}= 90^\circ$
20 dBi	128 kbps	0.2	1.8
	100 kbps	0.8	2.8
25dBi	128 kbps	4.8	6.8
	100 kbps	5.8	7.8

Table 6.4: Downlink Margins estimated for different θ_{EL} and there corresponding α_0 .

From the above calculation it is evident that the AraMiS S-band communication link is not only feasible and practical for AraMiS ground station communication but also with the GENSO member ground stations which provide S-band downlink facility.

In next chapter we will discuss in more detail about the design and architecture of GENSO compliant AraMiS Protocol.

Chapter 7

AraMiS Telecommunication Protocol

This chapter describes the AraMiS telecommunication protocol particularly developed in compliance with GENSO (Global Educational Network for Satellite Operations) Project. The major constrain for LEO orbit is limited visible telecommunication period from a specific ground station (maximum 20 minutes). AraMiS protocol overcomes this limitation and is designed to ensure compatibility with GENSO (a European Space Agency Project) that supports worldwide network of amateur ground stations. Therefore, extending the communication upto 24 hours a day by tunneling traffic over internet. The AraMiS communication subsystem uses two different narrowband channels, completely independent and redundant: UHF at 435 MHz and S-band at 2.4 GHz. GFSK modulation scheme is used for both bands and in particular AX.25 protocol with 9600 bps data rate is used specifically on the 435 MHz channel for compliance with ham-radio operators where as a high bandwidth 500 kbps data rate for 2.4 GHz. The main attributes for this protocol is to provide communication between satellite and ground station with main focus on (i) The Basic Housekeeping Telemetry, that contains the value of each on-board sensor acquired just before transmission, (ii) Extended Housekeeping Telemetry, which contains some information about basic telemetry (iii) Payload Telemetry, which has mission specific contents. Remote Command and Telemetry for non GENSO communication are handled by proprietary protocol classified as ad-hoc ARQ protocol, where acknowledgements and timeouts are used to achieve reliable data transmission over an unreliable service. Such ad-hoc protocol allows us to design packets without needless overhead information and to optimize parameters to maximize efficiency and reliability. For LEO satellites there is a limitation of visibility from a specific ground station. To overcome this deficiency we decided to develop a new AraMiS protocol design compatible with GENSO project [35]. This protocol is quite similar to the AX.25 connectionless protocol [39] for communication between satellite and any GENSO ground station.

This chapter also provides a detailed level description of AraMiS protocol which is also compliant with both UHF and S-band communication subsystems that are used by AraMiS project [40]. At first we have a brief overview of GENSO Project. Then we describe the Protocol Architecture and frame formats. Later we explain protocol mechanism and a performance analyses.

7.1 Project GENSO

satellites in LEO orbit has the Characteristic to be visible from the same ground station for a very limited period, such that generally a client is able to communicate with its own satellite in LEO orbit for a time not more than 15 minutes in each pass. In the case of CubeSat, this limitation is added to that relating to the low bit rate that typically characterizes the channels of the uplink and downlink of this type of satellites.

The GENSO project has been created based on following considerations:

- i. Each university ground station, when not used with its own satellite (approximately 97% of the time), could be made available as a remote station to other organizations to allow access to their satellite when you do not have a view.
- ii. There is the possibility of involving the amateur radio community, widely spread all over the Earth.

This section describes the basic operation of the project GENSO, taking into account that this project is still in development. In 2006 the project GENSO was created to provide such a ground communication system that maximizes the benefits by joint collaboration of project partners (as discussed above). It is coordinated by the Education Office of ESA (European Space Agency). GENSO aims to create a joint network, via internet, for amateur radio stations and universities that are located across the World (as represented in the Figure 7.1). Its main purpose is to provide universities an extended access to their scientific satellite, even when it is not them directly visible. The involvement of remote stations primarily concerns the reception of telemetry data, but the university mission managers have the possibility of using some of trusted stations for remote control in the uplink, according to the system described in the following section.

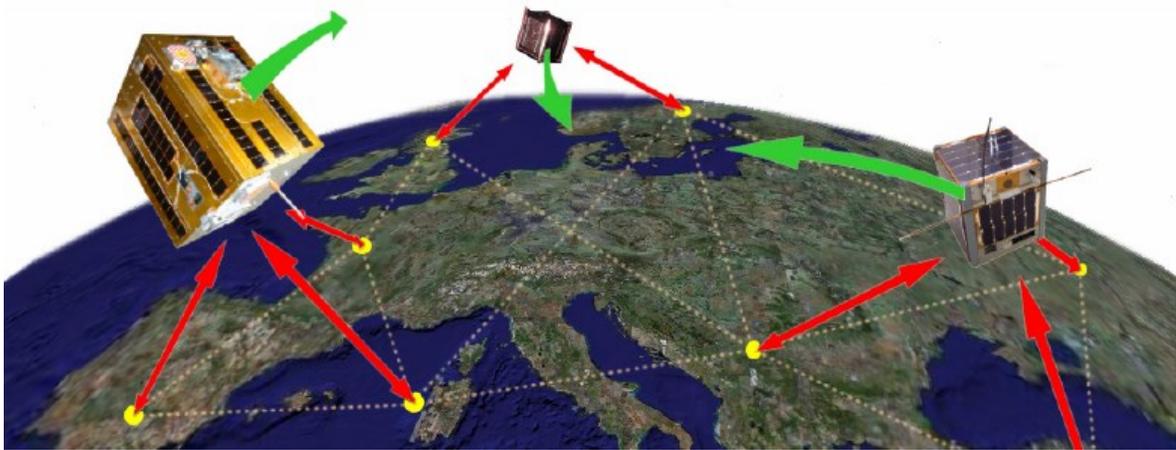


Figure 7.1: GENSO system representation

7.1.1 GENSO Architecture & Operation

GENSO The system consists of three distinct components (as shown in following Figure 7.2) [35]:

- 7.1.1.1. Authentication Server (AUS)
- 7.1.1.2. Ground Station Server (GSS)
- 7.1.1.3. Mission Control Client (MCC)

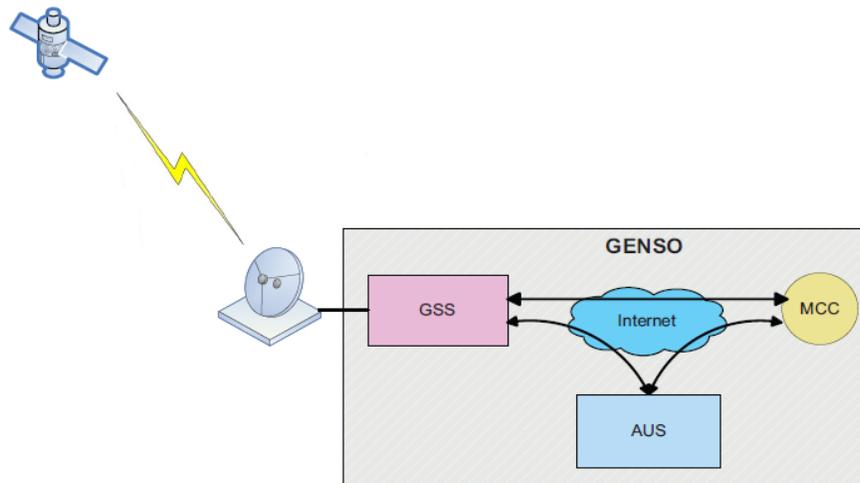


Figure 7.2: System components GENSO

Authentication Server (AUS)

Managed by the directors of GENSO, AUS is the central core of the network. The role of AUS is coordination and mediation between the generic MCC who want to avail of the services offered by the system GENSO and a certain set of GSS. During the development phase of the GENSO network, different architectures have been taken into account, from highly centralized to peer-to-peer networks. Both architectures have their advantages and disadvantages. In first case a Central Server was considered that can perform various tasks such as validation of the identity of the GSS and MCC instances when they log on, to maintain updated lists of the statuses and other attributes of all GENSO ground stations and spacecraft and their distribution to instances of the GSS and MCC as per request and requirement. This topology appeared simple to build, but had several drawbacks that made it unsuitable for GENSO. One major drawback was the lack of scalability. A central server, in fact, can only accept a limited number of connections from other computers. This fails to comply with main objectives of GENSO that is possibility of allowing the participation to an arbitrarily large number of organizations. In future, this could mean hundreds of thousands of satellites and ground stations. Clearly a single central server cannot handle the amount of data generated by thousands of stations, and in each case represent a critical point in case of problems or maintenance.

In the second case, the central authority has a role only during the registration and access of the different actors in the system, while in the service of they work as a peer-to-peer system. The benefits are limited reliance on a central server. The disadvantage is that central authority having poor control over managing GENSO internal system.

Therefore, an intermediate solution has been adopted between the previous cases, that have some servers whose function is to authenticate the nodes of the network, perform encryption, maintain an updated list of satellites and ground stations authorized and distribute this information when required by respectively GSS and MCC, that is, functions that do not require an intensive use of the available bandwidth.

Ground Station Server (GSS)

It is the application running on each amateur station that intends to offer its service to GENSO system. It allows the computer to take control of the rotor antenna, transceiver and the TNC modem to enable the Mission Control Station to communicate with their satellite. [41] To interface with the hardware, the GSS uses the appropriate driver (Ham Library). Just activated by means of the controller that connects with AUS to provide the characteristics of the station (frequencies and operable modulations, geographic location, etc.) that are added to the list of available ground station (GSSL - Ground Station Server List) maintained by the AUS and both receive a detailed list of satellites GENSO (PSL - Participating Spacecraft List), as can be seen in 1 and 1.2 in Figure 7.3. From that moment on, the MCC can book GSS to connect with satellite (Scheduler).

Mission Control Client (MCC)

It is the application running at the control center of the satellite (Mission controllers). Using a graphical interface, operator responsible for the mission is allowed to use the GENSO services. When enabled, the authentication phase provides it the characteristics of its satellite (frequency, modulation, etc.) and receives the updated list of all the available ground station (arrows 2 and 2.2 in the diagram) from AUS. At this point, the MCC can predict when its satellite will pass above which ground station. It can now contact the relevant GSS for usage reservation (arrows 3 to 3.1.1.1 in the diagram). There are different ways to communicate; we consider that during that period booked, the communication between MCC and satellite takes place in a transparent, two-way, real-time (as depicted by 4.1 and 4 in Figure 7.3).

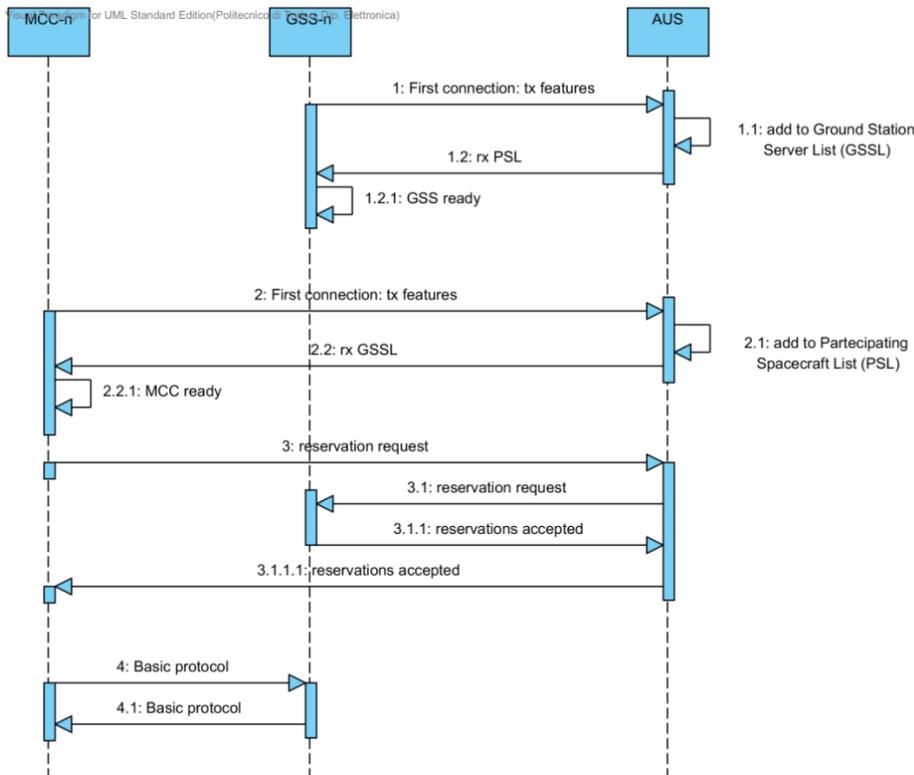


Figure 7.3: Sequence diagram of the operation of the system GENSO

7.2 AraMiS Protocol Architecture

This proposed higher layered protocol is implemented on OBC and manages functions related to control of duplication/ repetition in different ways depending on the type of command received as can be viewed in the transport layer and applying the second (cross-layer) approach. [42] We observe that the peer entity to the other end of OBRF is a generic ground station (GSS equipped with transceiver and TNC in kiss mode), while the peer entity of OBC is the Mission Control Client (POLITO), as shown in Figure 7.3.

The protocol is compliant to the architecture shown in Figure 7.4. The ground segment is constituted by a satellite Mission Controller (MCC), e.g. Politecnico di Torino (POLITO) and a certain number of radio amateur ground stations (GSS). We consider the hypothesis that GSS is able to receive via Internet commands from Mission Controller, to encapsulate it transparently into AX.25 frame and can also receive AX.25 frame i.e. transmitted from satellite, decapsulate data or message and further transmit it to Mission Controller MCC [42]. In Figure 7.6 is shown the encapsulation and decapsulation process. The AraMiS Architecture (as shown in Figure 7.5), apart from the memory, also includes other modules which are; (i) On-Board Radio Frequency (OBRF), a hardware module responsible to receive and transmit frame AX.25 and to execute backdoor commands; (ii) Telemetry Command Processor (TCP), a software in execution On board computer (OBC), responsible to read, interpret and execute ordinary commands, put in the memory data generated by their execution and to command the transmission start; (iii) Housekeeping Processor, another software in execution on OBC, responsible to collect measures from satellite sensors and to put those in the memory.

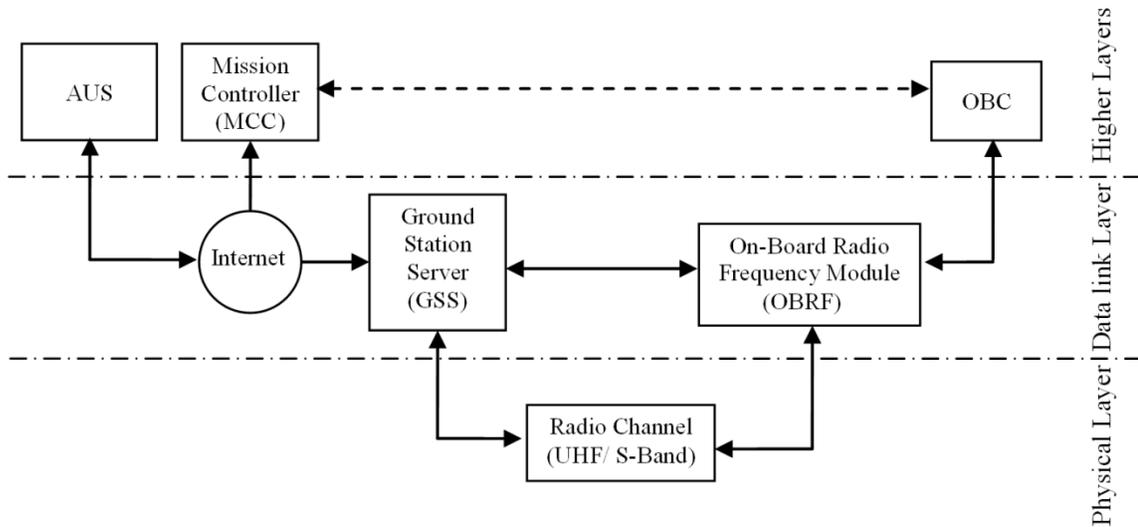


Figure 7.4: Layers and entities of the architecture-AraMiS GENSO in reference to the OSI [43]

To describe the proposed protocol we refer to which shows different blocks of the satellite involved in the development of the protocol as depicted in Figure 7.5. Consider the ground segment of the Mission Controller, which in our case is the Politecnico di Torino, and a GENSO generic ground station is a remote participant. These two actors are connected via internet.

The satellite, as well as memory and peripherals, we consider the modules:

7.2.1 On-board Radio Frequency

It is responsible for receiving and decoding the frame in uplink and encoding and transmission of the downlink frame.

7.2.2 Telemetry/Command Processor:

Its main tasks are to read the received commands in uplink, interpret and execute (or run them to a specific device), to perform memory management (allocating space, enter or retrieve data), and to initiate the transmission of the downlink *mission data* (Data generated by the payload of the satellite) and *housekeeping data* (Information related to the self-monitoring of the satellite)

7.2.3 Housekeeping Processor:

It is responsible for collecting measurements from various sensors distributed in the satellite in order to generate the *housekeeping data*; insert them into memory. Some examples of such data are, those related to the orbital position, temperature and state of charge of the batteries, the rule of availability of peripherals, etc.

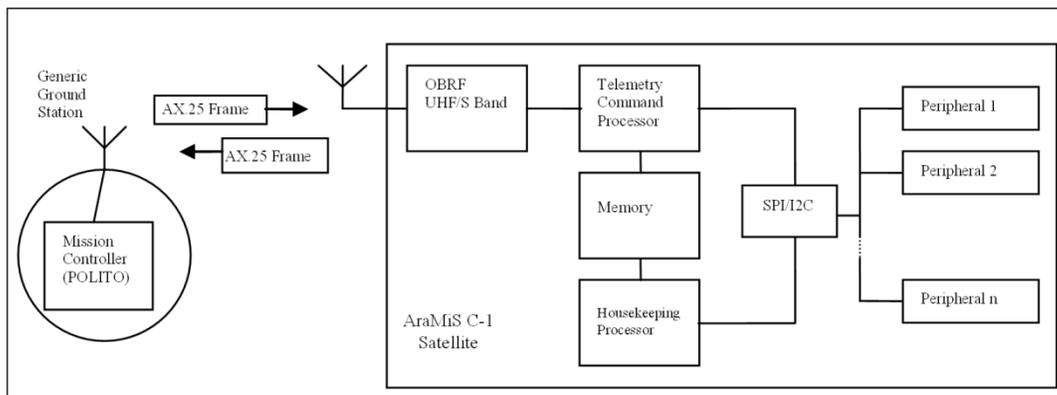


Figure 7.5: Global Architecture of the system

In general, for each command that is received by the satellite, at the end of its execution results in data or messages that needs to be sent back to Earth. From telecommunications perspective, the protocol provides to define the format of the commands, messages and data together with the manner in which these are exchanged between the MCC and OBC.

7.3 Protocol Structure

The commands from ground station are broadly categorized into two main classes based on their execution time:

7.3.1 LONG execution time commands (LONG):

These commands either have execution that can last for a considerably long time (for example in the case of a command of orientation) or the execution is scheduled in a future moment (such as commands that requires acquiring an image on the other side of the earth).

7.3.2 SHORT execution time commands (SHORT):

They simply require the retrieval and transmission of data from satellite. These can consist of data already residing in memory due to periodically generated (housekeeping functions) or the result of a previously sent long command.

One of major requirements this protocol must satisfy is its ability to control when to start the transmission of data or message in downlink in such a way that they can reach a specific member of the system ground station GENSO of which the MCC has booked the use, possibly, but not necessarily, the same one that forwarded the command.

For SHORT commands the requirement are automatically satisfied by allowing the transmission of data to earth as soon as it become available. In fact, by finding the difference in time between command and its response, it can be estimated that for how long the ground station that has transmitted the command will be visible to receive the response, except the last command that is just sent before the end of the satellite visibility period. However, two other approaches were considered to satisfy requirement in reference to the LONG commands:

- (i) Data retained in memory and automatically transmitted in an instant determined by the satellite itself, for example, when it reaches a certain orbital position;
- (ii) Data retained in memory and transmitted only upon receipt by the earth by a further request command (getData). Among these two approaches,

The latter option is chosen, since it is considered simpler and more robust for our system.

7.4 Packet Format Definition

Figure 7.6 shows the process of (de-)encapsulation where N(S) refers to sequence number of sent frames, N(R) is sequence number of received frames” and Command/Data/Message is the information field in the frame AX.25 [39]. The dashed line represents the interface between the OBRF that controls the functions of the physical layer/datalink manages the functions of transport and application layer protocol. This section shows the packet formats, i.e command uplink and downlink messages handled by the TCP.

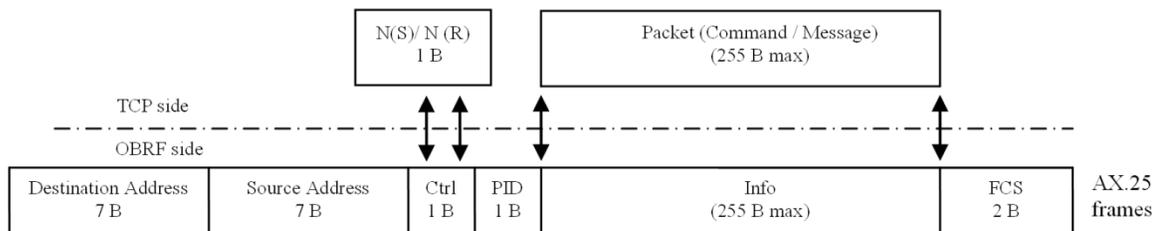


Figure 7.6: (Decapsulation) encapsulation phase

For the protocol operation; the following types of packets have been defined with respect to uplink and downlink communication.

In the uplink:

7.4.1.1 backdoorCommand

7.4.1.2 ordinaryCommand

In the downlink:

7.4.2.1 pureNACK

7.4.2.2 ordinaryDownlink

Each packet has its own format having predefined division into subfields where each occupying bit/byte provides a different meaning. This allows us to increase the efficiency of the protocol by avoiding the transport of unnecessary information, such as labels of quantities and units of measure. The destination entity must be able to distinguish the type of message received to decode correctly according to its format.

7.4.1 Packages in uplink

In uplink there are two types of packet formats.

backdoorCommand

The backdoorCommand perform a number of basic functions directly by MCU of OBRF. Their usefulness comes during recovery or management of some countermeasures in case of malfunctions in the rest of the system. The format of the backdoorCommand is represented in Figure 7.7

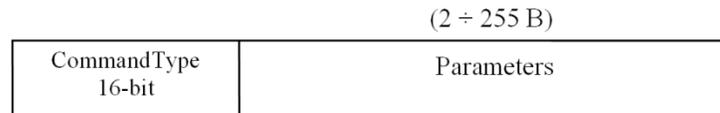


Figure 7.7: Format backdoorCommand

CommandType: Contains the code that uniquely identifies the command in the command set supported by OBRF. Using a table of correspondence, the MCU of OBRF can identify set of tasks that must be followed when receiving such a command. For Backdoor command the first bit of this field is always 0b0.

Parameters: It specifies the values of any parameters that is associated with command.

When OBRF decapsulates the packet from the frame, it reads the command code field to recognize a backdoor command, to interpret and execute it locally. Different commands are understood as ordinary commands and they are put in the output in the way that TCP can perform read by a polling loop. It is important to know that backdoor command doesn't use N(R), therefore control field of AX.25 frame is fixed at the value 0x03.

ordinaryCommand

Ordinary commands generally includes SHORT and LONG commands. They are forwarded transparently from OBRF to OBC to be handled there. Its structure generally consists of two fields (as shown in Figure 7.8 and Figure 7.9)

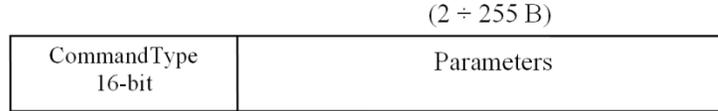


Figure 7.8: Format of an ordinaryCommand

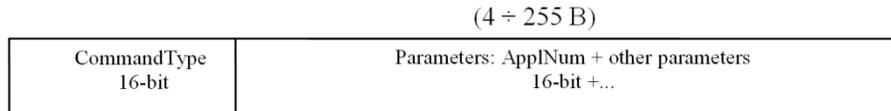


Figure 7.9: Particularization of an ordinaryCommand when it contains a LONG

CommandType: The first bit is equal 0B1 indicates to OBRF it as ordinaryCommand and is then forwarded transparently to the OBC and later acquired by the Telemetry/Command Processor. The remaining bits are coded for unique identification of command in the command set supported by the satellite. Using a table of correspondence the Telemetry/Command Processor will identify task needed to be executed by such command. In following Table 7.1 is an example of encoding for some commands (taking account of the first bit to 1).

ordinaryCommand Types	COMMAND_code
GetData	0x8000
GetFrag	0x8001
BasicTelemetry	0x8002
BatteryLifeStatus	0x8003
Orientation	0x8004

Table 7.1: Possible encodings for the field commandType

Parameters: Specify parameters values accompanying the command. For LONG commands first value that is added is always the Application Number, a number that will be used to assign name to the data generated from the execution of this LONG. It requires a later request of the data. The rest of parameter values that follows are specific for this command. For example, orientation command, will have as parameters the Application Number followed by an argument that indicates a value in degrees. GetData and GetFrag, as shown below are the commands by which a request for data or a fragment is made, the first added value is the Application Number of the previous LONG command for which data call has been generated and in the case of GetFrag follow the numbers of the fragment that needs to be received in the next transmission (Figure 7.10 and Figure 7.11).

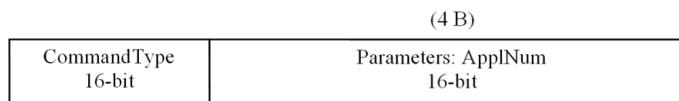


Figure 7.10: ordinaryCommand when it contains GetData

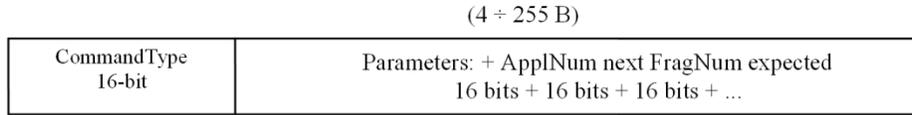


Figure 7.11: ordinaryCommand when it contains a GetFrag

Given the critical nature of commands, an error could affect the outcome of the mission itself which is beyond the CRC check of AX.25 Frame. Therefore, a threefold repetition of ordinaryCommand within the field provides as an additional mechanism for data integrity. Info more complex forms of coding are evaluated.

7.4.2 Packages in downlink

In the downlink, there are two categories of packet formats.

pureNACK

It is the only event of note that is generated in a completely autonomous fashion by OBRF when a command is received that does not exceed the CRC check, but it is properly addressed to the satellite. The format of the pureNACK has single field messageType (Figure 7.12). Its first bit is set to 0b0 to distinguish this NACK generated by OBRF, and then to the datalink layer, of any other type of message level generated From OBC (ordinaryDownlink).

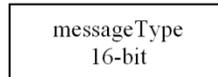


Figure 7.12: Format for pureNACK and all messages that contain no data

The Table 7.2 illustrates the behavior of OBRF based on the validity of the destination address and the CRC. This action has been defined in order for avoid frames from other users which might arrive at the satellite and can excite the OBRF. It also informs the mission controller when the transmission of a command fails because of CRC Check and then passed to anticipate retransmission without waiting for the expiration of the time-out.

Destination Address	CRC	Behavior
NO	NO	Ignore
NO	YES	Ignore
YES	NO	Send to Ground pureNACK
YES	YES	Forwards the Decapsulated Packet to TCP

Table 7.2: OBRF actions according to the destination address validity and CRC.

ordinaryDownlink

This category belongs to the majority of application layer messages generated in response to commands and beacon signal. These messages are passed transparently from OBC to OBRF. Its structure is generally formed by two fields Figure 7.13.

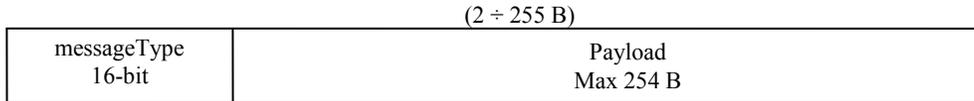


Figure 7.13: Format of a ordinaryDownlink

messageType: The first bit is equal to 0B1 to distinguish this message from a pureNACK generated in the datalink layer, while remaining bits constitute a code that uniquely identifies the message within a predefined manner. Table 7.3 elaborates an example of coding for the major types of message (taking into account that first bit is 1).

Type of ordinaryDownlink	messageType_code
BEACON	0x8000
ACK_DATA	0x8001
ACK_FRAG	0x8002
CMD_RECEIVED	0x8003
CMD_DUPLICATED	0x8004
DATA_NRDY	0x8005
BAD_CMD	0x8006
CMD_NOT_EXE	0x8007
PER_ERR	0x8008
MEMORY_FULL	0x8009
UNKNOWN_COMMAND_NUM	0x8010

Table 7.3: Possible encodings for the field MessageType

Payload: This is the field that contains any data that may accompany the message, as will be explained below, presenting the main types of message types.

BEACON

It is a packet that is transmitted at regular intervals (e.g every 30 seconds) in order to communicate the state of operation of the satellite and facilitate tracking. The Payload field in this case carries a basic form of telemetry (Figure 7.13).

ACK_DATA

This message is sent in response to SHORT commands and in particular to a command GetData. It is an acknowledgment message that carries the same payload in the data produced by the command (Figure 7.14).

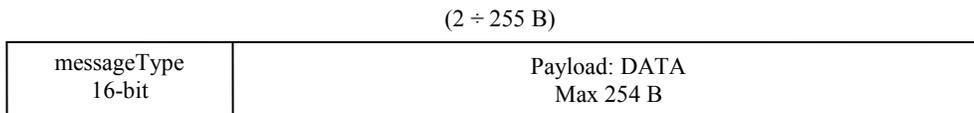


Figure 7.14: Response to ordinaryCommand that contains ACK_DATA

ACK_FRAG

This is the packet message in response to GetFrag command. It is similar, but the first two bytes of the Parameters field indicate the number of the remaining fragments of the total requested data (Figure 7.15).

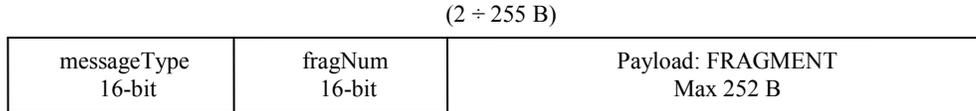


Figure 7.15: Message sent in response to ordinaryCommand that contains ACK_FRAG

All types of messages that follow share the same format (as shown in Figure 7.16 below), since they require only one field of the messageType.

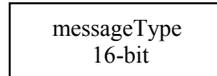


Figure 7.16: Format for all messages that carry data

CMD_RECEIVED

The Message that TCP provides the Mission Controller for the acknowledgment of successful acquisition of a long command.

CMD_DUPLICATED

The Message that TCP provides the Mission Controller for the acknowledgment of successful acquisition of a copy of a long command.

DATA_NRDY

This is the message that the TCP, upon receipt of a GetData or GetFrag, communicates to Mission Controller that the requested data are not yet ready in memory. This situation normally occurs when execution of LONG command is not yet finished.

BAD_CMD

The message through which TCP informs the Mission Controller that it has received an unrecognized command that is not supported by command set.

CMD_NOT_EXE

This is the message that the TCP informs the Mission Controller has received a command that is not executable.

PER_ERR

This is the message that the TCP informs the Mission Controller has received a command that refers to a non-responding device.

MEMORY_FULL

This is the message that the TCP informs the Mission Controller has received a command whose data cannot be stored in memory because it is full.

UNKNOWN_CMD_NUM

This is the message that the TCP informs the Mission Controller has received a GetData or GetFrag Application Number which does not reference any LONG command previously received.

7.5 Defined Functions

Before elaborating further let us summarize (as follows) the list of functions that were defined for the operation of the protocol and system.

sendCom(N(S), comType, params): is the message with which the MCC sends a command via the Internet at GSS to be transmitted to the satellite. Its arguments are:

- N(S): sequence number of sent frames (3 bits)
- ComType: identification code of the command (2 bytes)
- Parameters: covers topics specifiers command.

frame(destAddr, sourceAddr, N(S), info): represents the transmission of radio channel of the frame in the uplink and downlink. Its arguments are the fields of AX.25 frames:

- destAddr and sourceAddr: are the sender and/or destination station address
- N(S): is the number of frames transmitted sequence
- Info: The maximum field size of 255 bytes that contains the package.

sendMessage(N(R), messageType, payload): When referring to the ground segment, represents the message with which the GSS sends the frame content via the Internet received from the satellite to the MCC. Its arguments are:

- N(R): number of the last frame received from the satellite (3 bits)
- MessageType: identification code of the message
- Payload: contains the data in the case of ACK_DATA or a fragment in the case of ACK_FRAG.

readCom(N(S), comType, params): Represents the operation of poll by the TCP output of the OBRF. If no command is present returns false, otherwise it returns true, and arguments:

- Sequence number N(S)
- ComType
- Params: optional arguments of the command specifiers.

The following functions are involved during a command request SHORT telemetry.

get(command, data): the HP is running periodic queries of the generic device in order to collect housekeeping data. Its arguments are:

- Command: the code that requires a specific telemetry
- Data: data collected.

put(command, data): represents that part of the HP memory telemetry data collected.

getShort: TCP is that after receiving a command SHORT request telemetry data, makes a memory read.

The following functions are involved in a LONG command:

exec(command, arguments): is the TCP LONG after receiving a command, the device communicates with the responsibility to educate them. His arguments are

- Command: The instructions for the device
- Arguments: any data that specify the command

put (applNum, date): TCP is that after an exec allocate space in memory to hold data that will be generated after the execution of a command LONG. Returns true if there is space in memory, false if the memory is full. Subsequently represents the TCP which, after a get that makes the data, inserts the data in memory. Its arguments are:

applNum: Identifies the command and the data generated when inserted into the memory

Date: is the data generated by the execution of the command.

get (command, data): It is the TCP periodically queries the devices on which waits a given after the execution of a LONG. Return NULL if the data is not present; otherwise it returns the data. Its arguments are:

- Command: Specifies the device that data type is expected
- Data: contains the data returned

getLong (applLong, date): is the TCP after receiving a command SHORT request data from a previous LONG command, makes a memory read. Return NULL if the data is not present, otherwise it returns the data. Its arguments are:

- ApplNum: identifies the data element is required to recover from memory
- Data: the data is retrieved.

7.6 Protocol Operation

The operation of the protocol can be illustrate by means of sequence diagrams that represent the temporal sequence of events involved in the case of command SHORT management. Then there is an illustration in the case of a LONG command management, both in the case of transmission in the absence of errors. Subsequently, same sequences are reconsidered with assumption of errors. Finally a diagram is shown that represents the situation in which LONG and SHORT commands are combined together, it represents the situation for the protocol common operation [44]. The following use cases are also elaborated according to the exact sequence diagrams that are derived from the original function of *Visual Paradigm* software used by the working group of AraMiS in UML to describe the entire project [13].

7.6.1 SHORT command

The above Figure 7.17 shows the sequence of events related to a SHORT command in the absence of errors. The (1. get) is the Processor Housekeeping function while maintaining a query for a periodic output of the various sensors distributed in the satellite in order to collect the data and the (2. putShort) and places them in memory. In a generic instant assume that the Mission Controller decides to request, for example, the Housekeeping Telemetry data through the message (4. sendCom), Via the Internet to send to that ground station.

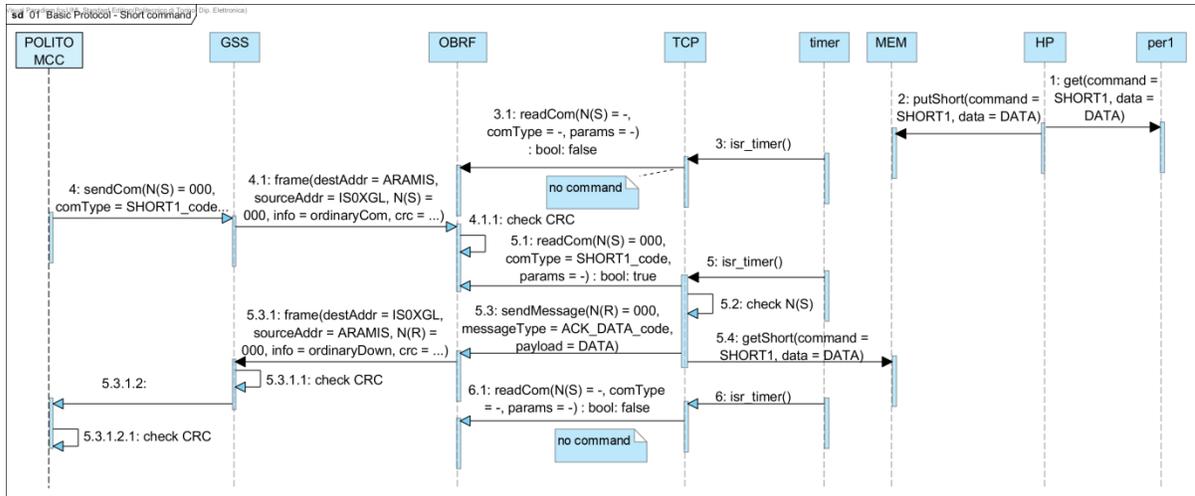


Figure 7.17: SHORT command sequence of events in absence of errors

When the ground station receives it, encapsulates the AX.25 frames by inserting in the various fields according to the arguments and in particular: AraMiS in the destination address field, its CALLSIGN in the return address, the sequence number in field N(S) and the Info field is the command format of the commands will be described later. The ground station then transmits the frame on the radio channel in the satellite direction (4.1 frames). The frame is received by OBRF Module which controls the CRC (4.1.1 CRC verification) and address of the recipient. Assuming that the CRC and the recipient address is correct, the sequence number and the command are decapsulated and made available in output so that they can be read from the Command Processor implemented on OBC by the function 5.1 readCom. Here it should be noted that the connection between the OBRF and TCP is based on a polling loop Where OBRF is the hardware module and TCP is software module The TCP then performs periodic readings (readCom) from OBRF output, initialized by a timer. Once the TCP has acquired the content from OBRF output, it checks the sequence number (5.2 Check N(S)) and then interprets the command according to its own table. Finding that it is a SHORT command and is executed regardless of whether it is received for the first time or not. Its execution consists in the simple recovery of the requested data from the memory (5.3 getShort), and packet generation which has the dual purpose of acknowledgment of the of frames reception coming from ground and transport of the required data. This message is passed to OBRF (5.4 sendData) which then transmits on the downlink with a procedure symmetrical to that used by the ground station for the uplink.

7.6.2 SHORT command with loss in the uplink

The Figure 7.18 illustrates the case in which the frame is lost in the uplink containing the SHORT command (1.1 frames). In this situation the Mission Controller will never receive the ack with the data (ACK_DATA), for which an elapsed time interval equal to the time-out retransmits the same command with the same sequence number (5. sendCom). It assumes that reception and all other activities continue to take place as described in the ideal case.

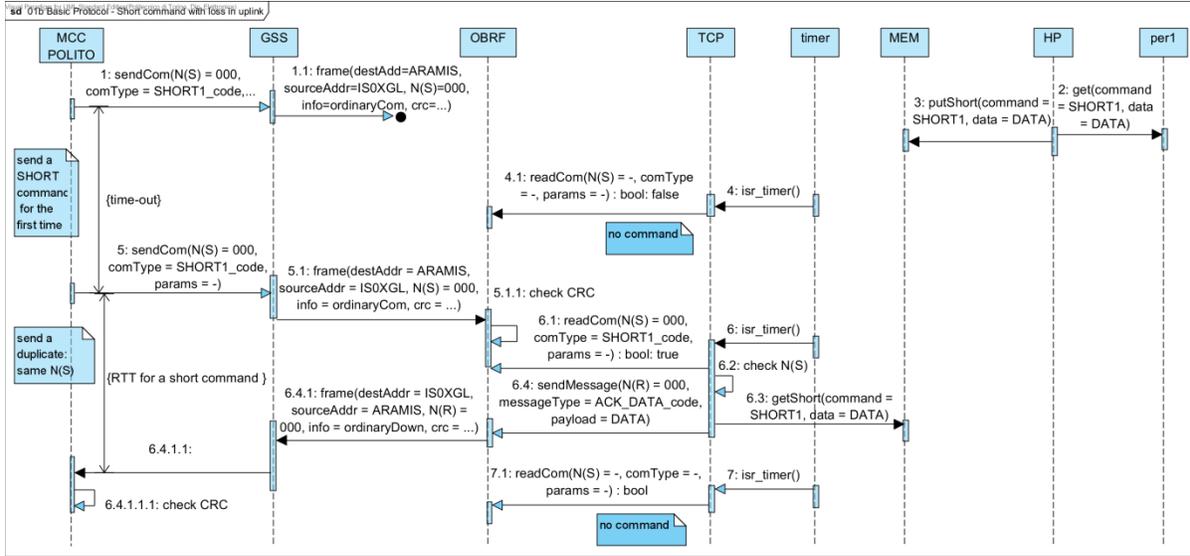


Figure 7.18: SHORT command sequence of events with frames lost in uplink

7.6.3 SHORT command with loss in the downlink

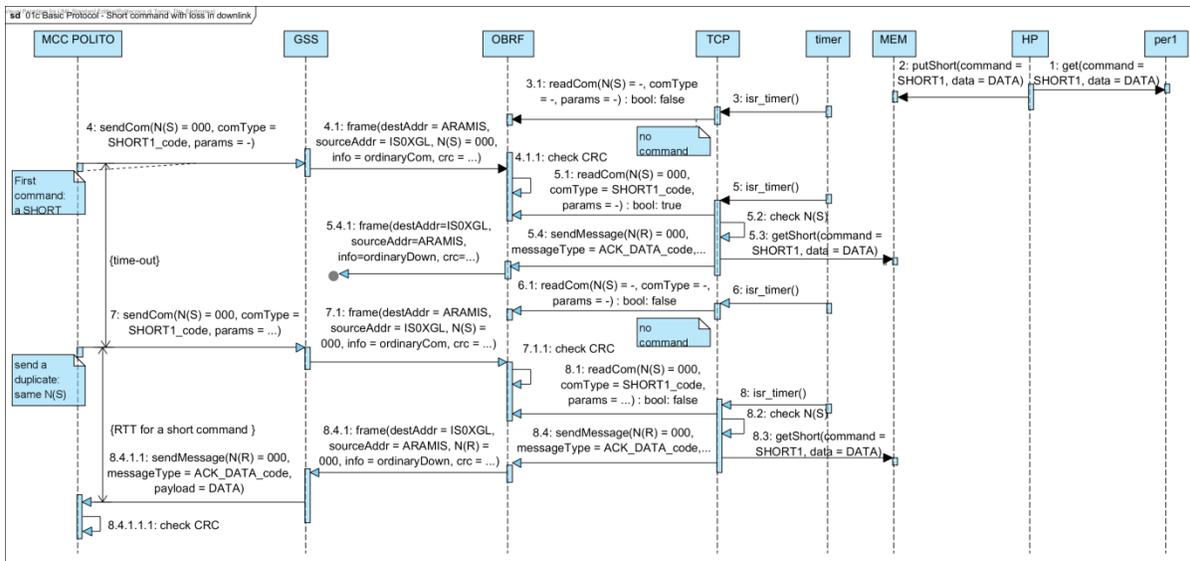


Figure 7.19: SHORT Command sequence diagram with frames lost in the downlink

The above Figure 7.19 illustrates the case in which the ACK_DATA is lost in the downlink (5.4.1 Frames). Compared to the case where it is lost in the uplink, now the satellite has received and executed the command. From Mission Controller stand point the situation is the same as the previous case: after a time-out interval without having received the ACK_DATA, the Mission Controller retransmits the same identical frame containing the same command (7. sendCom). When it reaches the satellite, the TCP from the control of the sequence number N(S) is able to recognize it as a copy of the previous command. Since this is a SHORT command i.e. a command that requires only data together with an ACK, therefore, it is re-executed identically (from 8.3 getShort until the end).

7.6.4 LONG Subsequent request with command

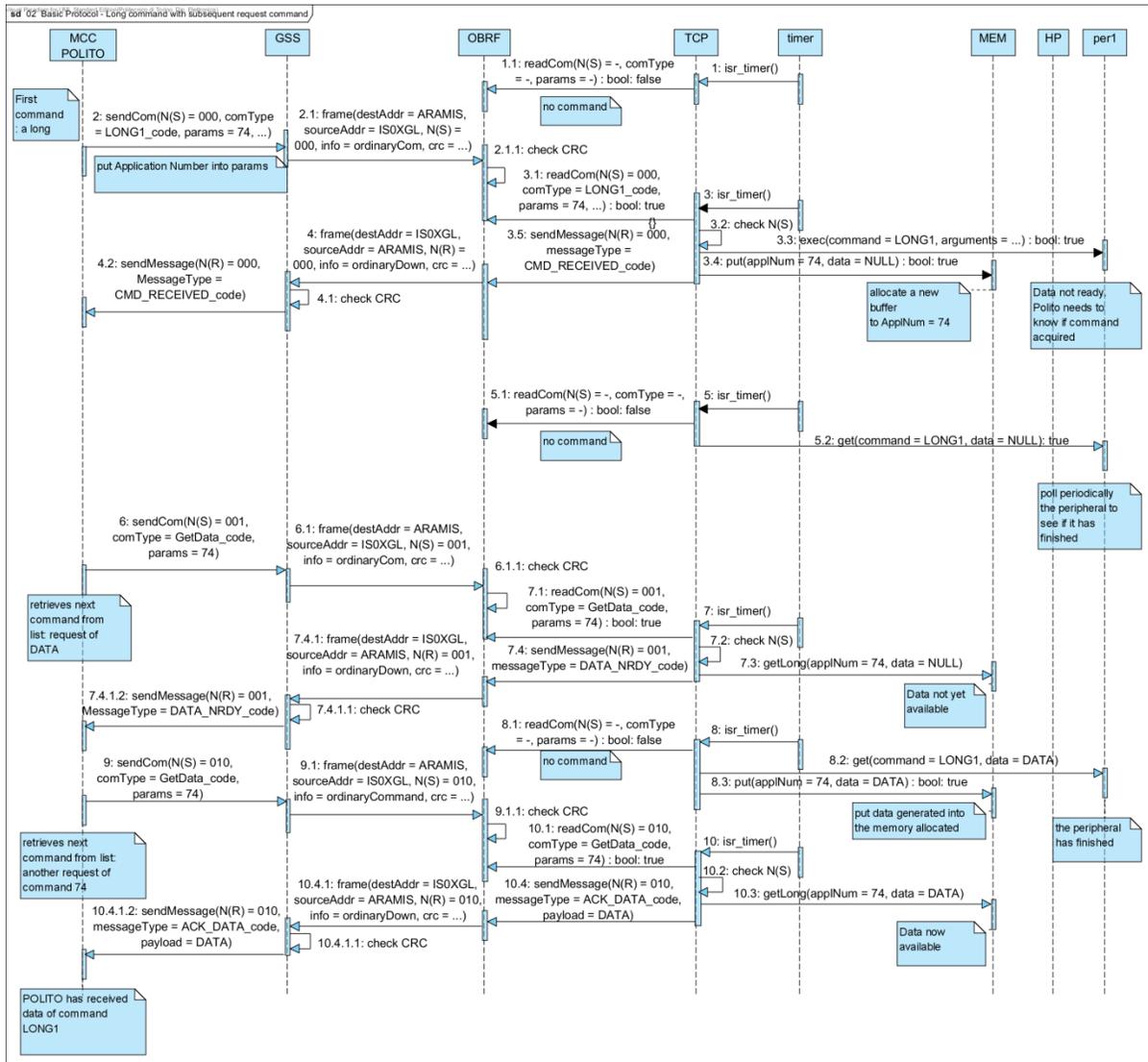


Figure 7.20: LONG command sequence of events with no errors

The sequence diagram (Figure 7.20) shows events for a LONG command without errors. In a generic instant it is assumed that the Mission Controller decides to request, for example, perform the image acquisition at a later instant. Through (2. sendCom), as in the previous case, the ground station sends the command accompanied now by a parameter, Application Number, which is a sequential number that identifies each command transmitted. It also serves as a reference in a subsequent request command for the data generated from its execution. The command is delivered to the TCP in the same fashion as the SHORT command. TCP verify the sequence number 3.2 Check N(S) to see if the command is received for the first time and then interprets the command according to a specific table. Assuming it's the first time it is received, the TCP communicates with the peripheral device responsible for its execution (3.3 exec) and immediately allocates space in memory to hold data that will be generated at the end of its execution (3.4 put). LONG commands for the Mission Controller still need to know right away if it has been received. Therefore, upon

reception of a LONG command, the TCP immediately generates a pure acknowledgement, that contains the message "command received" to be transmitted in the downlink (from 3.5 sendMessage until 4.2 sendMessage). TCP from this moment onwards periodically queries the concerned peripheral device for its end of execution and data acquisition. The function Get 5.2 shows one of these queries for which the execution is not yet finished.

After an adequate time interval (in relation to the previously sent LONG command), the Mission Controller may decide to inquire about the given data. Then generates the GetData command accompanied by the Application Number as a parameter of the command we want to retrieve the data and sends it to the ground station (6. sendMessage). The command after reaching the TCP is interpreted and it inquires the memory (Get 7.3) according to the memory allocation belonging to Application Number specified. In case if no data is yet available it returns a NULL. The TCP then creates a packet containing the message "data not ready" which is sent in the downlink (from 7.4 sendMessage to 7.4.1.2 sendMessage). The function Get 8.2 shows a further iteration of the query of the peripheral device by the TCP for which, finally, the execution is completed and the data is returned and placed in memory (8.3 put). In a later instant, the Mission Controller again decides to retrieve the data, so it sends again the (9. GetData) command. This time the query to the memory (Get 10.3) returns the data and can generate a TCP packet that passes ACK_DATA to OBRF (10.4 sendData) and is transmitted in the downlink. We conclude by noting that the command getData, in fact, is a SHORT command.

7.6.5 LONG command with ACK loss in the downlink

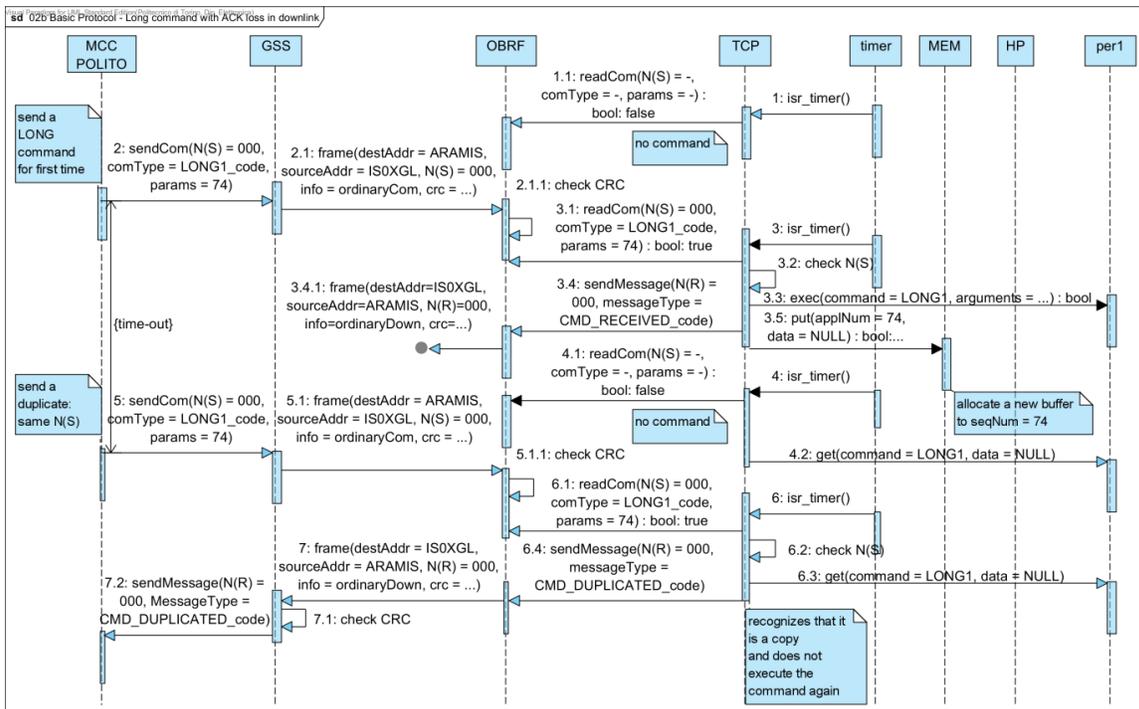


Figure 7.21: Long command sequence of events with frame lost in uplink

Regarding the frame lost in the uplink of a LONG command, the case of a SHORT command can be observed. The above Figure 7.21 instead illustrates the case follows loss of pureNack that in

7.6.7 Combined LONG/SHORT Command Operations

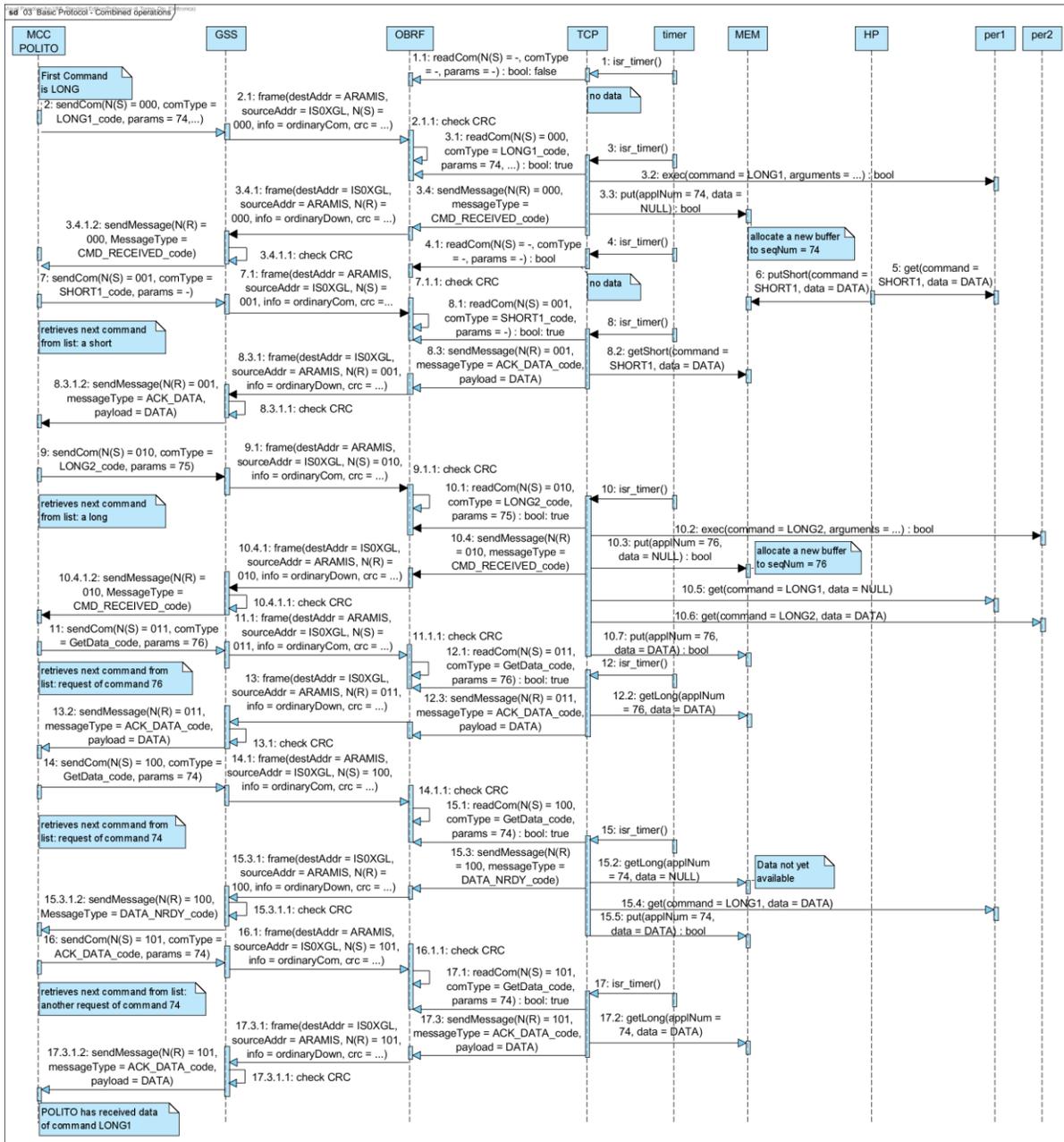


Figure 7.23: LONG and SHORT command handling at same time

Figure 7.23 shows a segment of time in which LONG and SHORT commands are handled together during the actual operation of the protocol. It is significant to note that between a LONG command and corresponding application data, other SHORT and LONG can be transmitted in the meantime. Moreover it may happen that the data generated by a LONG most recent is required before the one generated by a less recently LONG.

7.7 Performance Analysis

In this section, the throughput is calculated, i.e the quantity of useful data transferred per unit of time, in relation to a command SHORT for the recovery of a given maximum size of a single frame.

7.7.1 RDT commands SHORT

We define the *Requested Delivery Time* (RDT) as the time between the instant when the Mission Controller is transmitting the first bit of the command and the instant in which the Mission Controller receives the last bit of the response.

The RDT commands SHORT is the time that elapses between the instant when the first bit transmitted is the command uplink and the instant when the last bit of ACK_DATA is received in downlink. The Figure 7.24 represents the sequence already having regard to sending a command SHORT and receiving your response, ACK_DATA, where, however, were observed durations of the different operations. In particular we have ($T_{Internet_up}$ and $T_{Internet_down}$) respectively the represents the time taken for the command sent by the Mission Controller via the Internet to reach the remote ground station and the time it takes for the data sent from the ground station via the Internet to reach the Mission Controller. Their value is highly variable depending on the state of the network between the Mission Controller and the ground station. We can assume that their sum is of the same order of magnitude of the round trip time obtained by performing a ping test to a remote server, i.e 50 to 200ms.

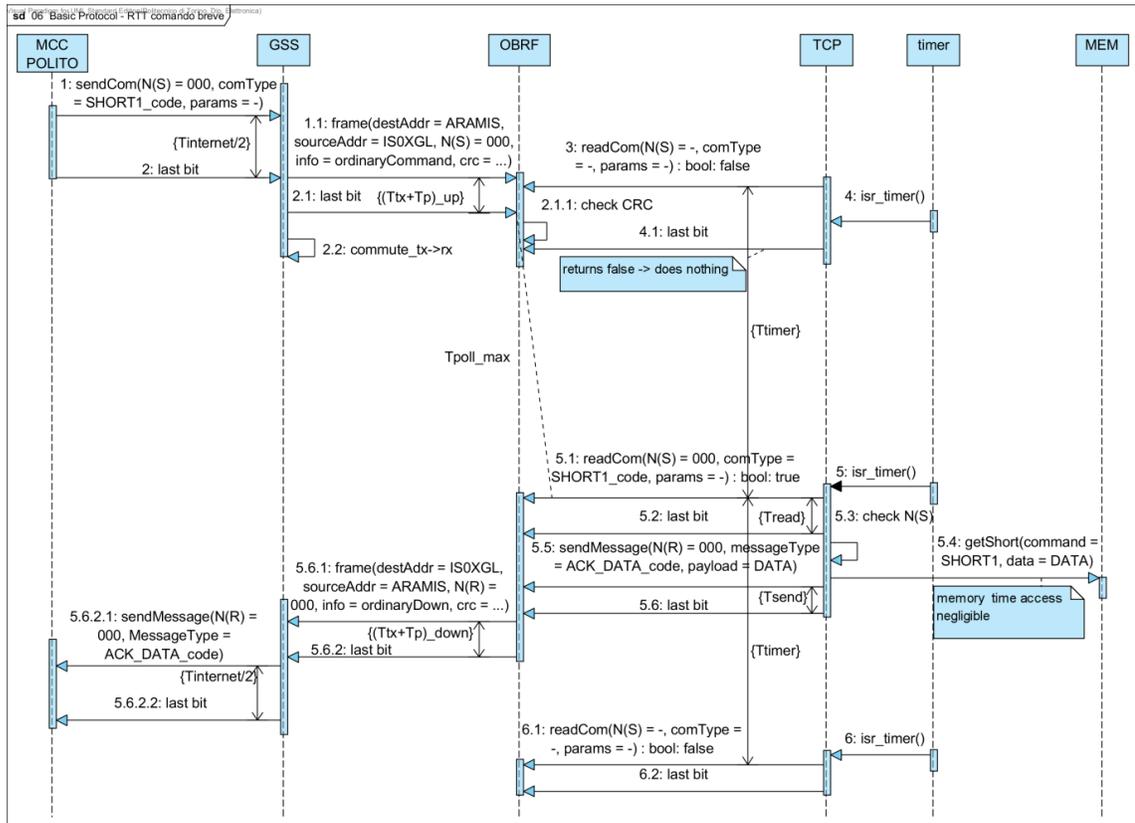


Figure 7.24: Disclosure of the times in the process of sending a command SHORT

$(T_{tx}+T_p)_{up}$ and $(T_{tx}+T_p)_{down}$: are respectively the time taken by the frame transmitted from the ground station by radio to reach OBRF of the satellite and the time taken to reach from OBRF frame transmitted by radio to ground stations.

More particularly, we have:

$$T_{tx} = \frac{L_{sync} + L_{header} + L_{payload}}{R_b} \quad (7-1)$$

and

$$T_p = \frac{d}{c} \quad (7-2)$$

T_{poll} is time that can elapse between the instant in which the command is ready exit from OBRF and the instant at which the poll by the TCP. It should be considered in the worst case, i.e one in which the read cycle starts an infinitesimal time before the command to be available. In this case the actual reading slide to the next period of the timer, for which;

$$T_{poll_MAX} = T_{timer} - \varepsilon$$

T_{read} and T_{send} : are respectively the time taken for the reading by the TCP sequence number $N(S)$ and of the command contained in the Info field and the time taken for writing by the TCP sequence number $N(R)$ and the data be encapsulated in the Info field to be transmitted in response.

$$T_{read} = \frac{D_{com}}{R_{b_int}} \quad (7-3)$$

$$T_{send} = \frac{D_{msg}}{R_{b_int}} \quad (7-4)$$

The bit-rate of the TCP connection between the OBRF and depends on the type of internal interface used in the final realization. If this is, for example, using an SPI serial interface then $R_{b_int} \leq 115.2\text{kbps}$.

They are considered negligible execution times of routine execution software and firmware of OBRF to TCP which substantially coincides with the access time to memory. The bit-rate R_{b_int} and the period of the timer T_{timer} , unlike the other quantities that are imposed by the system, must be sized so as to improve the RDT. We observe that the condition must be satisfied:

$$T_{timer} > (T_{read} + T_{send})_{MAX} \quad (7-5)$$

The timer time, i.e, cannot be less than the sum of the reading time of the command over the writing time of the reply message, if they should be considered of maximum size of 255 B. It is reasonable to consider using an interface between TCP and OBRF having $R_{b_int} = 100\text{Kbps}$. Then, $T_{timer} > 0.041\text{s}$.

The Request Delivery Time is therefore a sum of all these times:

$$RDT = T_{Internet_up} + (T_{tx} + T_p)_{up} + T_{timer} + T_{read} + T_{send} + (T_{tx} + T_p)_{down} + T_{Internet_down} \quad (7-6)$$

The diagram Figure 7.25 shows the RDT for a command SHORT and a message ACK_DATA, both in maximum size, in function of R_{b_int} .

$$R_{b_int} = 70 \div 150 \text{ kbps} \text{ and } T_{timer} = 0.06 \div 0.11 \text{ s}$$

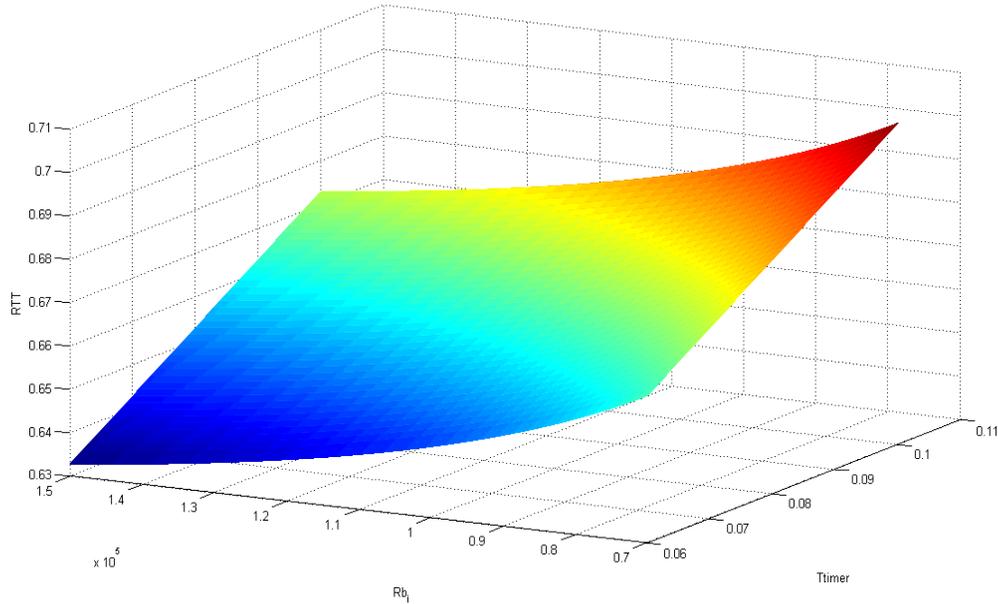


Figure 7.25: RDT as a function of R_{b_int} & T_{timer} and in the case of frames to maximum size

To get an estimate of the RDT of this Protocol, assume $R_{b_int} = 100 \text{ kbps}$ and the $T_{timer} = 0.06$ to keep a small margin having neglected the processing time of the TCP. Furthermore, we fix the typical values:

$$T_{Internet_up} = T_{Internet_down} = 0.05 \text{ s}$$

$$L_{sync} + L_{header} + L_{payload} = 20\text{B} + 18\text{B} + 255\text{B} = 293\text{B}$$

$$d = 800 \text{ Km}$$

$$D_{com} = D_{msg} = 255\text{B}$$

we obtain; $RDT = 0.656 \text{ s}$

Considering the expression of Thruput:

$$R'_B = \frac{L_{payload}}{RDT} \quad (7-7)$$

We obtain the estimate $R'_{B_{up}} = R'_{B_{down}} = 3.1\text{kbps}$ compared with a data rate on the radio channel of 9600 bps. This data rate seems quite meager in terms of data transfer. In order to improve on the thrupt using fragmented communication this RDT can be significantly improved.

No. of fragments	RDT	Through Put (R'_B)
10 Frames	02.8 s	7.07 kbps
50 Frames	12.4 s	7.61 kbps
100 Frames	24.4 s	7.82 kbps

Table 7.4: Fragmented communication with RDT and corresponding thrupt at UHF transceiver bit rate of 9600bps and $R_{b_{int}} = 100\text{kbps}$

This shows that even with fragmentation and using standard UHF transceiver the attained thrupt is still not very high. As many of GENSO member ground stations supports S-band in downlink with a bit rate of around 500kbps. Hence, the thrupt in downlink can be improved further.

For a bit rate of 500kbps and $R_{b_{in}} = 500\text{kbps}$ (faster SPI interface between OBRF and TCP) then $T_{timer} = 0.01\div 0.06\text{s}$. Using the given equations (6-4), (6-5), (6-6) and (6-7) the RDT and R'_B have been recomputed given in following Table 7.5:

No. of fragments	RDT	Through Put (R'_B)
10 Frames	0.432 s	54.27 kbps
50 Frames	0.783 s	149.76 kbps
100 Frames	1.220 s	191.97 kbps

Table 7.5: Fragmented downlink communication with RDT and corresponding thrupt at S-band transceiver bit rate of 500kbps and $R_{b_{int}} = 500\text{kbps}$

The above tables reveal that the thrupt increases with respect to the number of fragments sent in downlink. Although, the GENSO protocol only support 9600 bps in the uplink communication (for UHF band only). Whereas, in downlink communication, it can also support S-band communication. Various scenarios were studied for AraMiS protocol considering different TCP polling interval (associated with bitrates $R_{b_{int}}$ ranging from 100÷500 kbps). On the other hand S-band transceiver data rates (R_{txrx}) were also assumed for a variation from 100÷500kbps. The corresponding results are depicted in Figure 7.26 and Figure 7.27 ; respectively. Analyzing these results reveal that thrupt increases markedly as the subsequent number of fragments in downlink increases. Also for a higher $R_{b_{int}}$ and R_{txrx} there is a step increase of thrupt for a fixed number of fragments. There is a saturation point beyond which no further thrupt can be improved. It is due to the fact that the first uplink command send for fragmented data is also accounted in RDT for the fragmented downlink data.

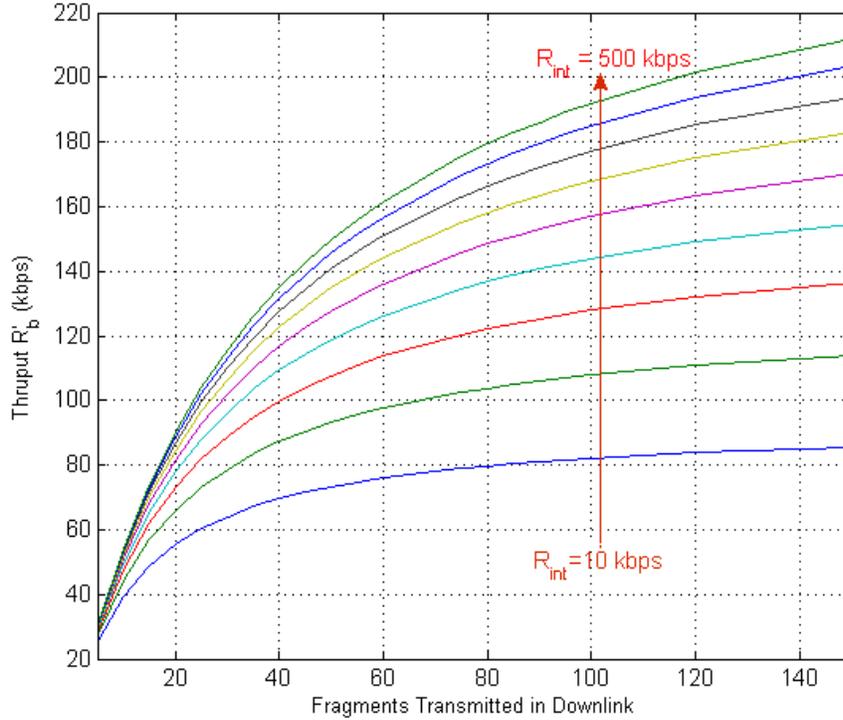


Figure 7.26: Thruput in downlink for different number of fragment with R_{int} (at $R_{txrx}=500\text{kbps}$)

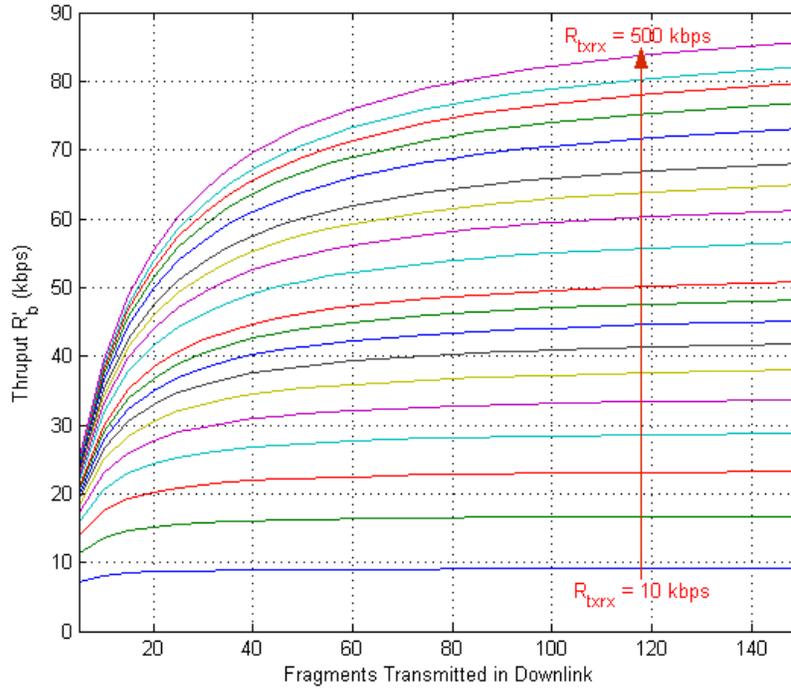


Figure 7.27: Thruput in downlink for different number of fragment with R_{txrx} (at $R_{int}=100\text{kbps}$)

From above results it can be concluded that on order to fully utilize the GENSO potential it is preferable to communicate in uplink with 9600bps UHF and use the S-band communication in the

downlink which improves throughput significantly. This can be even further enhanced if fragmented data communication is performed in downlink. In this manner the overhead time delay of $T_{\text{Internet_up}}$ and $T_{\text{Internet_down}}$ can also be reduced. It can also compensate the slow UHF transmission rate of uplink (9600bps = 0.024 s for packet length of 293 B).

Chapter 8

Conclusions

The work performed in this thesis provides an illustration of designing a complete subsystem using the AraMiS design approach. This design approach is adopted in the design, implementation and testing of all the subsystems. The 1B9_CubeTCT module is a fundamental step in designing and implementation of communication subsystems over a single tile for CubeSat standard nanosatellites. It presents a novel solution in solving constraints such as cost, power consumption and physical dimension. To implement such a subsystem for a satellite on single module is not a trivial job. The subsystems implementation has been performed in this work using COTS which tends to have lower reliability but adopting proper selection and designing techniques can overcome this problem. The selection criteria for COTS have been based on performance, lower dimensions, cost and power consumption which can help to attain acceptable results both in terms of performance and financial feasibility.

The design of S-band RF front end for 1B9_CubeTCT was not a straight forward task. The RF COTS integration required careful design. For performance analyses all RF components were at first tested separately over a reference design. They were later integrated with proper matching on a 4 layered FR-4 PCB. All traces dimensions were optimized and tuned to ensure proper matching over a limited dimension of 98 x 51 mm.

The second part of the work addressed the design of S-band antennas for both AraMiS-C1 and conventional AraMiS Satellites. This task was following from the initial theoretical design to simulation and then its realization. The prototyped structures were later tested for performance on the bases of RL, gain, and bandwidth measurements. The simulated and measured data results were found in close agreement and also satisfied linkbudget requirements.

The later part of the thesis work discusses AraMiS telecommunication protocol that has been developed in compliance with GENSO project. Different scenarios dealing with SHORT and LONG time based commands were explained in detail. Various cases of single and fragmented data requests were also evaluated in normal and frames lost (in uplink or downlink) cases. Performance analyses were carried out to optimally utilize this protocol over the GENSO to attain acceptable throughput.

Bibliography

- [1] Riki Munakata, CubeSat Design Specifications, Rev.12, California State Polytechnic University, 2009.
- [2] Passerone C., Tranchero M., Speretta S., Reyneri L., Sansoe C., Del Corso D., Design Solutions for a University Nano-satellite, Aerospace Conference, 2008 IEEE , vol. no. pp.1,13, 1-8 March 2008.
- [3] Stefano Speretta, Leonardo M. Reyneri, Claudio Sanso'e, Maurizio Tranchero, Claudio Passerone, Dante Del Corso, Modular Architecture for Satellites, 58th IAC, Hyderabad, India, 24 - 28 September 2007.
- [4] Satellite Classification. http://centaur.sstl.co.uk/SSHP/sshp_classify.html
- [5] Jake A. Schaffner. The electronic system design, analysis, integration and construction of the cal poly state university cp1 cubesat. Technical report, Cal Poly State University, 2002.
- [6] Stefano Speretta, Project solutions for low cost space missions, PhD thesis, March 2010
- [7] Anwar Ali, M. Rizwan Mughal, Haider Ali, Leonardo M. Reyneri, Innovative power management, attitude determination and control tile for CubeSat standard Nano-satellites, Acta Astronautica, Volume 96, March–April 2014, Pages 116–127
- [8] Muhammad Rizwan Mughal, Onboard Communication Systems for Low Cost Small Satellites, PhD thesis, Politecnico di Torino, Italy, February 2014.
- [9] T. Instruments, MSP430 User's Guide, Texas Instruments, Tech. Rep., SLAU208H, Dec 2010.
- [10] M. Muhammad Rizwan Mughal; Anwar Ali; Haider Ali; L. Reyneri. A smart honeycomb tile for small satellites. pages 1–7. IEEE Aerospace conference, 2014.
- [11] Anwar Ali, Leonardo M. Reyneri, Haider Ali and M. Rizwan Mughal, Components selection for a simple boost converter on the basis of power loss analysis, 63rd International Astronautical Congress, Naples Italy, 1-5 Oct. 2012.
- [12] CC2510 datasheet. [Online] Available <http://www.ti.com/product/cc2510f32>
- [13] UML tutorial: <http://www.tutorialspoint.com/uml/index.htm>
- [14] Mughal, M.R.; Reyneri, L.M.; Ali, A., "UML based design methodology for serial data handling system of Nano-satellites," Satellite Telecommunications (ESTEL), 2012 IEEE First AESS European Conference on , vol., no., pp.1,6, 2-5 Oct. 2012, doi: 10.1109/ESTEL.2012.6400194.
- [15] 1B127 datasheet [Online] <http://polimage.polito.it/aramis/products/1B127/22.06.20091B127.pdf>

- [16] M. Rizwan Mughal, Anwar Ali, Leonardo M. Reyneri, Plug-and-play design approach to smart harness for modular small satellites, *Acta Astronautica*, Volume 94, Issue 2, February 2014, Pages 754-764, ISSN 0094-5765, <http://dx.doi.org/10.1016/j.actaastro.2013.09.015>.
- [17] Chester Simpson, *Linear and Switching Voltage Regulator Fundamentals*, National Semiconductor.
- [18] TPS5450 datasheet [Online] Available <http://www.ti.com/product/tps5450>
- [19] INA138 datasheet, High-Side Measurement Current Shunt Monitor, Burr Brown Products from Texas Instruments.
- [20] D. Del Corso, C. Passerone, L. M. Reyneri, C. Sansoe, M. Borri, S. Speretta, and M. Tranchero, "Architecture of a Small Low-Cost Satellite," 10th Euromicro Conference on Digital System Design, pp. 428-431, August 2007.
- [21] Maxim 2644 datasheet [Online] <http://datasheets.maximintegrated.com/en/ds/MAX2644.pdf>
- [22] RFMD SZM-2166Z datasheet [Online] <http://www.rfmd.com/CS/Documents/SZM-2166ZDS>
- [23] TriQuint TQP4M0010 datasheet [Online] <http://www.triquint.com/products/p/TQP4M0010>
- [24] Tx-line tool AWR microwave Office [Online] <http://www.awrcorp.com/microwave-office>
- [25] C. A. Balanis, *Antenna Theory*, 2nd ed. New York: Wiley, 1997.
- [26] *Handbook of Microstrip Antennas*, J. R. James and P. S. Hall, Eds., Peter Peregrinus, London, UK, 1990
- [27] R. C. Johnson and H. Jasik, *Antenna Engineering Handbook*, 3rd ed. New York: McGraw-Hill, 1993.
- [28] High Frequency Structure Simulation (HFSS), ANSOFT Co., Pittsburgh, PA, USA.
- [29] J. Kerr, "Microstrip polarization techniques," in *Proc. Antenna Appl. Symp.*, Urbana, IL, 1978.
- [30] T. Chiba, Y. Suzuki, and N. Miyano, "Suppression of higher modes and cross polarized component for microstrip antennas," in *IEEE Antennas Propagat. SOC. Int. Symp. Dig.*, May 1982, pp. 285-288.
- [31] John Huang, "A technique for an Array to Generate Circular Polarization with Linearly Polarized Elements", *IEEE transactions on Antennas & Propagation*, Vol.34, pp 1113-1124, September 1986.
- [32] Ali H., Aru A., Sansoe C., Ali A., Mughal R., *Reconfigurable Antenna Design for Nanosatellites*, Intl. ESTEL Conference, 2012, Rome, Italy, Sep, 2012.
- [33] D.M. Pozar, *Microwave Engineering*, Third Edition, John Wiley & Sons, 2004.
- [34] J. Huang, "C.P. microstrip array with wide axial ratio bandwidth and single feed L.P. elements," in *IEEE Antennas Propagat. Soc. Int. Symp. Dig.*, June 1985, pp. 705-708.

- [35] ESA Education Project [Online] <http://www.GENSO.org/>
- [36] G. Máral & M. Bousquet, Satellite Communications Systems, Fifth Edition, 2009.
- [37] T. Pratt, C. Bostian & J. Allnutt, Satellite Communications, Second Edition, 2003.
- [38] L. Ippolito, “Satellite Communications Systems Engineering”, 2008.
- [39] W. A. Beech, D. E. Nielsen, J. Taylor. “AX.25 Link Access Protocol for Amateur Packet Radio”, Version 2.2, American Radio Relay League and the Tucson Amateur Packet Radio Corporation, 1998.
- [40] S. Speretta, L. Reyneri, C. Sansoe’, M. Tranchero, C. Passerone, and D. Del Corso, “Modular architecture for satellites,” 58th International Astronautical Congress, September 2007.
- [41] M. Davidoff, The satellite radio amateur's handbook, First Edition, Amer Radio Relay League; 1990.
- [42] Global Educational Network for Satellite Operators, Functional Block Diagram, Revision 616. July 2007.
- [43] S. Tanenbaum, D. J. Wetherall, Computer Networks, Fifth Edition, Prentice Hall, 2010.
- [44] Ali H., Aru A., Sansoe C., Reyneri L., Design & Development of the Telecommunication Protocol for AraMiS Nanosatellites, 63rd International Astronautical Congress (IAC-2012), Naples, Italy, Sep, 2012.