REMCO SCHOEMAKER

ROBUST AND FLEXIBLE COMMAND & DATA HANDLING ON BOARD THE DELFFI FORMATION FLYING MISSION

MASTER OF SCIENCE

AT

CHAIR OF SPACE SYSTEMS ENGINEERING - AEROSPACE ENGINEERING
DELFT UNIVERSITY OF TECHNOLOGY

ROBUST AND FLEXIBLE COMMAND & DATA HANDLING ON BOARD THE DELFFI FORMATION FLYING MISSION

REMCO SCHOEMAKER

As Part of an International Network of 50 CubeSats

Supervisor: ir. Jasper Bouwmeester

7th of July 2014 – version 1.0 - final version

A Singularitarian is someone who understands the Singularity

and has reflected on its meaning for his or her own life.
— Ray Kurzweil

Following the light of the sun,

we left the Old World.
— Christoper Columbus

ABSTRACT

After a decade of worldwide CubeSat development, the Von Karman Institute took the lead, to prove the concept of a large swarm of 50 CubeSats which will be launched in early 2016. In addition, the QB50 mission has the main objective to measure the characteristics of the lower thermosphere. In scope of the QB50 project the TU Delft will develop two nanosatellites called Delta and Phi (DelFFi) which will perform formation flying experiments as well. In order to enable these experiments a propulsion system, an inter satellite link and an improved attitude determination & control system (ADCS) will be integrated into the DelFFi nanosatellites.

The lessons learned of operating Delfi-n3Xt - the predecessor of DelFFi - and recommendations for future Delfi missions are outlined in this thesis. During the operations of Delfi-n3Xt the Command & Data Handling System (CDHS) performed well. Mainly some new requirements with respect to telecommanding are proposed for DelFFi. However, because the design of the DelFFi nanosatellites is significantly different than Delfi-n3Xt, including completely new subsystems and capabilities, the CDHS needs to be re-evaluated as well. Hence, an analysis of the requirements for the CDHS of DelFFi is provided. In the thesis a total of 24 requirements are proposed or requested for modifications.

In addition, a design is proposed to include flexible telemetry (FlexTLM) and a telemetry history playback functionality for DelFFi. This robust but flexible design will enable the DelFFi mission to transmit more relevant telemetry, while at the same time semi-autonomous FlexTLM configurations will reduce the time resources required of the mission operators. Calculations proved the science data downlink requirement of 300 kbits per day can be satisfied during the entire mission, while real-time telemetry is still available as well. More calculations have shown the FlexTLM solution can potentially enable the DelFFi downlink to transmit at least 64 % more relevant telemetry than a fixed telemetry solution to the Delfi ground station.

In the second part of this thesis, the feasibility of wireless intra-spacecraft communication for nanosatellites was investigated. An analysis of the performance and considerations of such an implementation is given, based on experiments conducted with Bluetooth Low Energy (BLE) modules in a representative nanosatellite environment. The on-ground experiments with BLE have proven the performance of the wireless link in a nanosatellite is excellent and the power consump-

tion is moderate. The observed Packet Error Rates (PER) were lower than 0.3 %.

A proposal was made to implement a completely wireless BLE temperature sensor experiment on DelFFi. This experiment will be powered by a small solar cell and supercapacitors to allow operation in eclipse. The wireless experiment on-board DelFFi will provide validation of the BLE technology in a space environment. This experiment will pave the way for more extensive wireless intra-spacecraft capabilities for future nanosatellites.

No one who achieves success does so
without acknowledging the help of others.
The wise and confident
acknowledge this help with gratitude.

— [Alfred North Whitehead 2]

ACKNOWLEDGMENTS

This thesis would not have been completed without the guidance and support of a number of people. Therefore, I would like to thank and acknowledge the support of the following persons.

First of all, I would like to thank my supervisor ir. J. Bouwmeester for his advice and critical reviews of my conducted research and engineering design. Also, the knowledge he shared about former nanosatellites such as Delfi-C3 and Delfi-n3Xt was a valuable input for my engineering work on DelFFi. Furthermore, I appreciate the electronics engineering support of Nuno Baltazar dos Santos for the conducted Bluetooth Low Energy experiments for wireless intra-spacecraft communication. In addition, I would like to acknowledge the software engineering development effort of Maher Sallem to accelerate the Bluetooth Low Energy experiments. Besides all the former mentioned persons, I would like to thank Dr. Guo - the project manager of DelFFi - for his daily lead of the DelFFi project and Gang Liu for our cooperation on the Command & Data Handling System.

Other people I would like to thank, who are too many to name here, are my DelFFi team members for all the discussions and experiences we shared. Finally, I would like to thank my friends, family, parents and sister for their support throughout my studies in Delft.

CONTENTS

1 INTRODUCTION 1
- 1.1 The QB50 Mission 1
- 1.2 The DelFFi Mission 1
- 1.3 Thesis Work 2
- 1.4 Thesis Outline 3

2 DELFI-N3XT OPERATIONS EXPERIENCES 5
- 2.1 Mission Operations Organization & Infrastructure 7
- 2.2 OBC Reboot Behavior 7
- 2.3 Switch Count Errors 9
- 2.4 Update Software Parameters Capability 9
- 2.5 Telecommands Preparation & Log 10
- 2.6 Radio Amateur Participation 11
- 2.7 Telemetry Flexibility & Efficient Usage of Bandwidth 13
- 2.8 Software Interface Control 13
- 2.9 Transponder Test & Satellite Malfunction 14
- 2.10 Conclusions & Proposed Requirements for DelFFi 16

I IMPROVED CDHS ARCHITECTURE FOR DELFFI 17

3 DELFFI CDHS REQUIREMENTS ANALYSIS 19
- 3.1 DelFFi Data Protocol Selection 19
- 3.2 GAMALINK device 20
- 3.3 Telemetry & Telecommanding 22
- 3.4 Data Acquisition 25
- 3.5 Subsystems Interfaces 29
- 3.6 Monitoring 31
- 3.7 Revisions of CDHS Requirements During the Project 32
- 3.8 Summary & Statistics 34

4 FLEXIBLE CDHS & TELEMETRY 35
- 4.1 Telemetry Budget Analysis for DelFFi 35
- 4.2 Reconfigurable Telemetry 38
- 4.3 Telemetry History Playback Definition 47
- 4.4 DelFFi Software ICD FlexTLM Implementation 52
- 4.5 FlexTLM Ground Segment Implications 57
- 4.6 Telemetry History Feasibility Analysis 57
- 4.7 FlexTLM Assembly Software 59
- 4.8 Conclusion & Requirements 61

II WIRELESS INTRA-SPACECRAFT COMMUNICATION FOR NANOSATELLITES 63

5 NANOSATELLITE BLUETOOTH LOW ENERGY EXPERIMENTS 65
- 5.1 Past Wireless Intra-Spacecraft Communication Experiments 66

5.2 Hardware & Wireless Protocol Selection 68
5.3 Experimental Setup 69
6 BLUETOOTH LOW ENERGY EXPERIMENTS RESULTS 81
6.1 Results & Analysis of Experiments 81
6.2 Wired versus Wireless Nodes 87
6.3 Conclusion 88
7 PROPOSAL FOR A WIRELESS BLE TEMPERATURE SENSOR EXPERIMENT ON DELFFI 91
7.1 Communication Architecture 91
7.2 Power Supply & Storage 93
7.3 Placement & Integration 100
7.4 Estimated Development Effort & Resources 101
7.5 Conclusion & Requirements 103
8 CONCLUSIONS 105
8.1 Delfi-n3Xt Operations Lessons Learned 105
8.2 DelFFi CDHS Requirements 105
8.3 Flexible Telemetry 106
8.4 Wireless Intra-spacecraft Communication 107

III APPENDIX 109
A PROJECT MANAGEMENT 111
B THESIS PLANNING 113
C DELFI-N3XT MISSION SUCCESS RATE 115

BIBLIOGRAPHY 119

LIST OF FIGURES

Figure 1 Artist impression of Delfi-n3Xt in space. [3] 5
Figure 2 Delfi-n3Xt operations timeline. 6
Figure 3 Mission control room during Delfi-n3Xt LEOPS. 7
Figure 4 Image of the South Atlantic Anomaly (SAA) at approximately 560 km altitude. [4] 8
Figure 5 Locations of OBC reboot events of Delfi-n3Xt as of 10 December 2013. (MapLink, Google Maps) 8
Figure 6 Overview of received Delfi-n3Xt frames per day. 11
Figure 7 Delfi-n3Xt radio amateur coverage part 1. (WX-Track) 12
Figure 8 Delfi-n3Xt radio amateur coverage part 2. (WX-Track) 12
Figure 9 Illustration of the GAMANET concept. [5] 21
Figure 10 Estimated average received telemetry per day for the DelFFi mission calculated for different orbital altitudes. 37
Figure 11 Design Option Tree (DOT) of the flexible telemetry architecture. Items highlighted in orange are chosen for the FlexTLM approach. 46
Figure 12 Concept of telemetry history definition orbit. [Adapted, 6] 48
Figure 13 Visualization of telemetry history playback definition. 49
Figure 14 Designed activity flow of the FlexTLM assembly software. 60
Figure 15 Delfi-C3 Autonomous Wireless Sun Sensor (AWSS). [7] 66
Figure 16 Picture of the all-optical satellite OPTOS. [8] 67
Figure 17 Arduino Due board. [9] 70
Figure 18 BLE113 module breakout board. 70
Figure 19 Nanosatellite test model with unpopulated PCBs. 74
Figure 20 Schematics of the BLE113 module. [10] 75
Figure 21 Top view of the BLE113 radiation pattern. [11] 78
Figure 22 Front view of the BLE113 radiation pattern. [11] 78
Figure 23 Side view of the BLE113 radiation pattern. [11] 79
Figure 24 Distance versus measured Packet Error Rate of the client in the test model. 83
Figure 25 Distance versus measured Packet Sent Error Rate of the server in the test model. 83

Figure 26 Amount of PCBs versus Packet (Sent) Error Rate of the client and server in the test model with a distance of 25 cm. (without battery) 84
Figure 27 TX power setting versus Packet (Sent) Error Rate of the client and server in the test model for 8 PCBs + battery pack with a distance of 25 cm. 84
Figure 28 Picture of Delfi-C3 spare model Bluetooth Low Energy test setup. 85
Figure 29 Wireless BLE temperature sensor experiment communication architecture. 92
Figure 30 Conceptual circuit diagram of the wireless BLE experiment with a super capacitor and solar cell. 93
Figure 31 Picture of the EDLC252520-351-2F-21 super capacitor. [12] 96
Figure 32 Picture of the DMT334R2S474M3DTA0 super capacitor. [13] 96
Figure 33 Physical layout and keep out areas of the current DelFFi body panel configurations. The left panel represents the configuration without a hinge and the right panel one with a hinge. 98
Figure 34 KySat-1 during CubeSat acceptance testing. [14] 100

LIST OF TABLES

Table 1 Summary of DelFFi CDHS requirement analysis statistics. 34
Table 2 Estimated daily science and tagging data volume of the FIPEX instrument. 38
Table 3 Information frame structure of the AX.25 protocol. [15] 39
Table 4 Summary of telemetry flexibility architecture options. 42
Table 5 Subsystems ICD data example micropropulsion. 42
Table 6 AX.25 frame transmit sequence table example for a 1 Hz transmission frequency. 43
Table 7 Example of a fictitious telecommand for AX.25 frame content control. 45
Table 8 Example of a fictitious telecommand to control the telemetry history playback. 50

Table 9 DelFFi software ICD subsystem summary for different telemetry profiles. 53
Table 10 Possible ADCS modes and other active subsystems during each DelFFi mission phase. [16] [17] 54
Table 11 Overview of daily data volume savings for different telemetry packets as a result of the FlexTLM implementation for different orbital altitudes. 56
Table 12 Overview of AX.25 frame inclusion frequency for the WOD and FIPEX science data based on different orbital altitudes based on a transmitter data rate of 9.6 kbps. 58
Table 13 Average power consumption for different TX settings and power profiles. 82
Table 14 Influence of power profiles on PER and PSER. 84
Table 15 Delfi-C3 EMI influence on PER and PSER. 86
Table 16 Selection of super capacitors potentially suitable for the wireless temperature experiment. 96
Table 17 Space available for an additional independent solar cell on the DelFFi body panels. 97
Table 18 Estimated hardware costs for the wireless BLE temperature sensor experiment. 102
Table 19 Estimated development time & resources required for the wireless BLE temperature sensor experiment. The time required is based on the work of one full time engineer. 102

LISTINGS

Listing 1 XML code in the project.bgproj file. 77
Listing 2 XML code in the hardware.xml file. 77
Listing 3 XML code in the config.xml file. 77

NOMENCLATURE

ADCS	Attitude Determination & Control System
AFF	Autonomous Formation Flying
AFH	Adaptive Frequency Hopping
AIT	Assembly, Integration & Testing
Analytical Graphics	AGI
API	Application Programming Interface
AWSS	Autonomous Wireless Sun Sensor
BER	Bit Error Rate
BLE	Bluetooth Low Energy
CAN	Controller Area Network
CDHS	Command & Data Handling System
CL	Confidence Level
COTS	Commercial Off-The-Shelf
CRC	Cyclic Redundancy Check
DAB	Deployment Antenna Board
DAQ	Data Acquisition
DSSB	Delfi Standard System Bus
DUDe	Delft Universal Data Extractor
Elapsed Time Counter	Elapsed Time Counter
EMC	Electromagnetic Interference Compatibility
EMI	Electromagnetic Interference
EPFL	École polytechnique fédérale de Lausanne
EPS	Electrical Power System
ESR	Equivalent Series Resistance

ETC	Elapsed Time Counter
FCS	Frame Check Sequence
FEC	Forward Error Correction
FIPEX	Flux-ϕ-Probe-Experiment
FlexTLM	Flexible Telemetry
FPGA	Field-programmable Gate Array
GATT	Generic Attribute Profile
GENSO	Global Educational Network for Satellite Operations
GPS	Global Positioning System
GS	Ground Station
IrDA	Infrared Data Association
ISL	Inter Satellite Link
LEOPS	Launch & Early Operations Phase
MAC	Media Access Control
MPPT	Maximum Power Point Tracker
MPPT	Maximum Power Point Tracking
NUTS	Norwegian University of Technology Test Satellite
OBC	Onboard Computer
OPTOS	All-Optical Satellite
OTA	Over-the-air
OWSL	Optical Wireless Links to Intra-Spacecraft
PCB	Printed Circuit Boards
PER	Packet Error Rate
PID	Protocol Identifier
PLL	Phase Lock Loop
PSER	Packet Sent Error Rate

RF	Radio Frequency
RSSI	Received Signal Strength Indication
RX	Receiving Mode
SAA	South Atlantic Anomaly
SCS	Satellite Control Software
SD	Secure Digital
SDK	Smart Development Kit
SEE	Single Event Effects
SET	Single Event Transients
SoC	System on a Chip
SPI	Serial Peripheral Interface
STM	Surface Thermal Monitor
STX	S-band transmitter
System Tool Kit	STK
TASC	Triangular Advanced Solar Cells
TCS	Thermal Control System
TLE	Two-line Element
TNC	Terminal Node Controller
TRL	Technological Readiness Level
TVC	Thrust Vector Control
TX	Transmitting Mode
UART	Universal Asynchronous Receiver/Transmitter
VKI	Von Karman Institute
WOD	Whole Orbit Data
XML	Extensible Markup Language

1 INTRODUCTION

On November 21st of 2013 the second Delfi nanosatellite has been launched into orbit. The Delfi-C3 successor Delfi-n3Xt was launched by a Dnepr rocket from a base in Yasny, Russia. Several new subsystems are integrated in the 3U CubeSat Delfi-n3Xt. Amongst other systems it includes an Attitude Determination & Control System (ADCS), experimental micropropulsion thrusters and an S-band transmitter. Extensive operations of Delfi-n3Xt has provided new insights and lessons learned which can be applied on DelFFi. Especially the Command and Data Handling System (CDHS) was analyzed and improvements were suggested. In addition, a design for a flexible telemetry system for DelFFi ensuring the downlink of more relevant telemetry is outlined. Also, research into wireless intra-spacecraft communication has been conducted by hardware experiments and analysis of the results obtained. Based on the experiments conducted a proposal for a wireless intra-spacecraft communication experiment on DelFFi was made.

1.1 THE QB50 MISSION

After the successful nanosatellite development programs of Delfi-C3 and Delfi-n3Xt, the TU Delft will develop two new nanosatellites. In scope of the European QB50 project the TU Delft will build two triple unit CubeSats called Delta and Phi (DelFFi) which will be part of a swarm of about fifty nanosatellites. This swarm of nanosatellites will be launched with the mission objective to take atmospheric measurements in early 2016. CubeSats are especially suitable for the QB50 mission because the low Earth orbit increased drag causes a relatively fast orbital decay and an expected mission lifetime of about eight months. Larger conventional satellites are generally too costly to perform these type of missions and therefore CubeSats are an excellent alternative.

1.2 THE DELFFI MISSION

The DelFFi nanosatellites will include an advanced ADCS, micropropulsion and an inter satellite link called GAMALINK to enable formation flying. The following mission statement has been defined for DelFFi:

> "The DelFFi mission shall demonstrate autonomous formation flying and provide enhanced scientific return within QB50 from 2015 onwards, by utilizing two identical triple-unit CubeSats of TU Delft which further advance the Delfi-n3Xt platform."

In order to allow the DelFFi satellites to perform their mission the CDHS of the Delfi-n3Xt based platform needs to be redesigned. More specifically, changes need to be made to improve the flexibility of the CDHS. The CDHS of a satellite is sometimes expressed as the brain of a satellite. It makes sure all data is distributed and commands are properly addressed [18].

1.3 THESIS WORK

The lessons learned from the operations of Delfi-n3Xt are discussed and improvements to the CDHS for DelFFi are proposed. The main research question concerning the Delfi-n3Xt operations is:

> "What aspects of the CDHS can be improved for DelFFi?"

In addition, the requirements with respect to the CDHS of DelFFi are analyzed and modifications to the requirements are listed. Also a preliminary analysis about the GAMALINK device is provided to support future CDHS engineers in the development of the GAMALINK activity flows for the OBC software.

New important features of DelFFi include the implementation of reconfigurable telemetry and telemetry history playback in the CDHS. This will provide the mission analysts of DelFFi more relevant telemetry in order to enhance the knowledge about the DelFFi satellites. Therefore, one of the main tasks of this thesis will be the design of the flexible telemetry and telemetry history playback functionality for DelFFi. This functionality should be robust, but at the same time allow flexible operations. The main research question for this part of the thesis is:

> "Can a reliable flexible telemetry system increase the scientific return of the DelFFi mission and to what extent?"

More specifically, will the flexible telemetry system ensure the data requirements of DelFFi can be met? What is the most robust and flexible telemetry (FlexTLM) design for DelFFi? Next to the design of the FlexTLM system a software Interface Control Document (ICD) will be created as well. In this software ICD the FlexTLM solution will be implemented to show the actual application of the FlexTLM system and enable the verification of the data requirements of DelFFi.

Another important part of this thesis will look into the feasibility of wireless intra-spacecraft communication for nanosatellites. Experiments were performed to determine the suitability to implement a wireless intra-spacecraft communication experiment onboard DelFFi. The central research question which will be answered is:

> "Does a wireless data link in a nanosatellite have benefits over traditional wired data links?"

If this question can be answered positively, the recommended implementation of a wireless intra-spacecraft communication link for nanosatellites will be proposed for the DelFFi mission. The following subquestion needs to be answered as well: ¨Which wireless protocol and hardware is most suitable for internal data communication in a nanosatellite?¨ Wireless experiments with Bluetooth Low Energy have been conducted in a representative nanosatellite environment.

1.4 THESIS OUTLINE

In order to improve the CDHS for DelFFi the experiences gained by operating Delfi-n3Xt will be analyzed. Encountered problems during the operations of Delfi-n3Xt and potential improvements for the CDHS of DelFFi will be discussed in chapter 2. Subsequently, in chapter 3 the requirements related to the CDHS of DelFFi which need to be taken into account will be outlined. In chapter 4 a concept will be proposed for a more flexible telemetry system for the DelFFi mission.

Hereafter, in the last part of this thesis an outline of wireless intra-spacecraft communication on-ground experiments will be provided in chapter 5. The results and findings of this experiment will be given in chapter 6. Finally, a proposal for a wireless temperature sensor node experiment onboard DelFFi will be made in 7. The thesis will be closed in chapter 8 with a summary of all the conclusions made and recommendations for future work. In appendix A a summary about how this thesis was managed can be found.

2 DELFI-N3XT OPERATIONS EXPERIENCES

In this chapter the experiences gained and lessons learned by operating the Delfi-n3Xt nanosatellite with respect to the CDHS and general Delfi Ground Station (GS) operations will be analyzed. Especially areas which can be improved for future Delfi nanosatellites are discussed and recommendations to change certain parts of the CDHS architecture for the DelFFi nanosatellites are made. The following mission statement was defined for Delfi-n3Xt:

> "Delfi-n3Xt shall be a reliable triple-unit CubeSat of TU Delft which implements substantial advances in 1 subsystem with respect to Delfi-C³ and allows technology demonstration of 2 payloads from external partners from 2012 onwards."

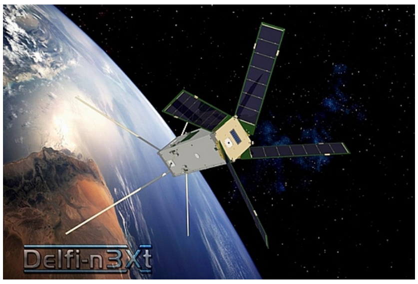

Figure 1: Artist impression of Delfi-n3Xt in space. [3]

In the first section of this chapter the mission operations organization and infrastructure is explained. Hereafter, in section 2.2 the reboot behavior of the Delfi-n3Xt Onboard Computer (OBC) will be discussed. In section 2.3 anomalies which occurred with the switch counters in the software are discussed. Thereafter, in section 2.4 possible improvements and recommendations for the onboard software will be given. In section 2.5 the lessons learned for the preparations of sending telecommands will be given. Afterwards, in section 2.6 the limited radio amateur participation of Delfi-n3Xt will be discussed and possible solutions to mitigate the dependency on radio amateurs in future Delfi nanosatellite missions will be given. Subsequently, in section 2.7 an implementation of the flexible definition of telemetry and efficient usage of data bandwidth will be proposed. In section 2.8 a method for a more structured way of synchronizing telemetry parameter definitions between different software will be introduced. Finally, in section 2.9 information about the conducted transponder tests and the events leading to a malfunctioning satellite is given. In figure 2 an overview can be found of the Delfi-n3Xt operations in order to gain a general insight about the sequence of events.

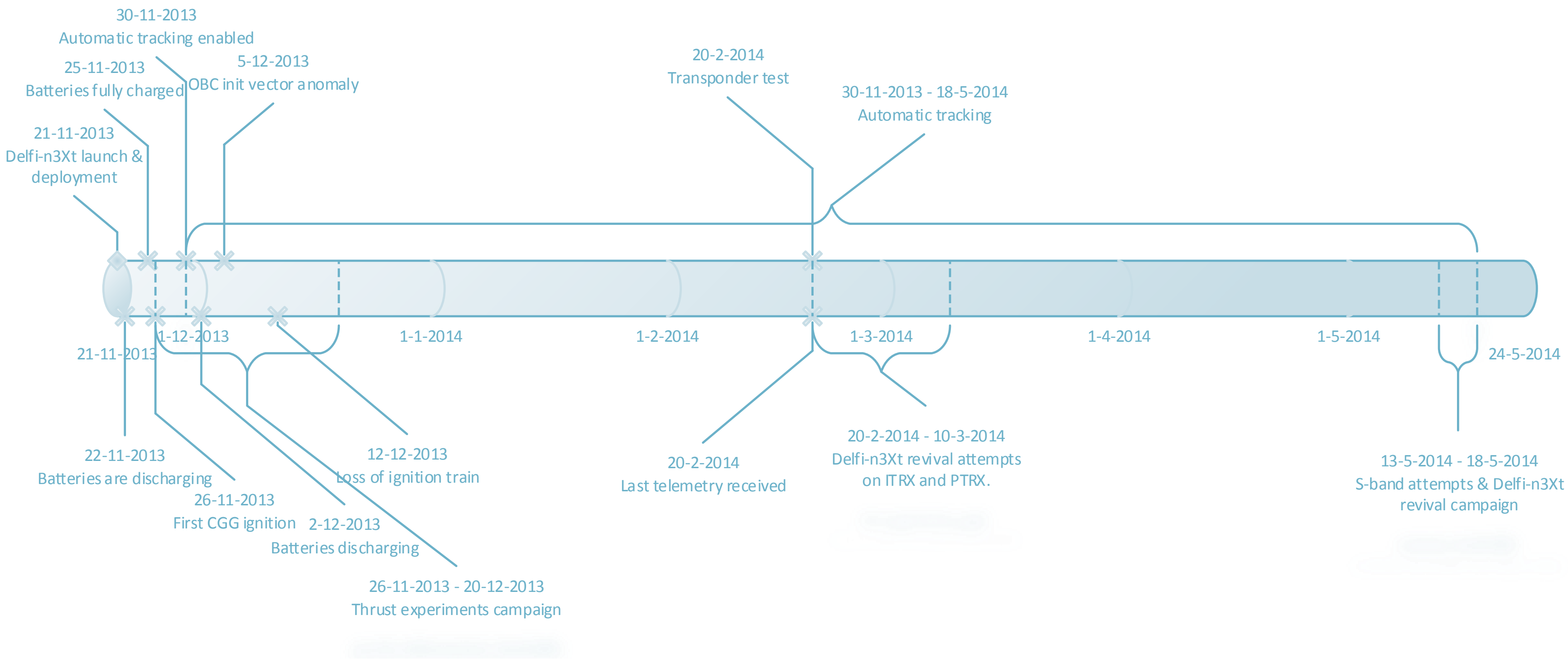

Figure 2: Delfi-n3Xt operations timeline.

2.1 MISSION OPERATIONS ORGANIZATION & INFRASTRUCTURE

A summary about the organization and infrastructure of the Delfi mission operations is given in this section, before details about the performance of Delfi-n3Xt will be provided in the remainder of this chapter. During the launch day of Delfi-n3Xt there was a dedicated mission control room which housed all the mission subsystem analysts. These mission analysts were monitoring the data of the first passes through a web interface to check if all subsystems performed nominally. All mission analysts prepared the Launch & Early Operations Phase (LEOPS) by reading the relevant documentation for their subsystems. The Delfi ground station manned by licensed radio operators made sure telemetry was received during each Delfi-n3Xt pass.

Figure 3: Mission control room during Delfi-n3Xt LEOPS.

After the launch day a schedule was made for the staff of seven radio operators to attend Delfi-n3Xt passes with the Delfi GS in shifts. Most of these passes were attended during sun mode, but some in eclipse as well to verify the transceiver was off or to send telecommands. Every pass new telemetry was recorded in the Delfi database and the mission analysts were able to make conclusions based on the investigation of this data. If actions had to be taken the appropriate telecommands and procedures were consolidated by coordination between the project manager, mission analysts and radio operators.

2.2 OBC REBOOT BEHAVIOR

During the operations of Delfi-n3Xt several reboots of the OBC were observed every day. It was not immediately clear what the reason for these reboots was. The received telemetry from Delfi-n3Xt did not indicate any specific reason for the reboot of the OBC. Therefore the time and the exact locations in the orbit around Earth of the event reboots were determined. A hypothesis was that maybe a certain anomaly was the underlying reason of the reboots. For example, the South Atlantic Anomaly (SAA) could be a cause of the reboots. The SAA is a weak part in the Earth´s magnetic field where cosmic rays and charged particles can reach lower places in the atmosphere than at other locations [4]. Satellites passing this area during orbits are exposed to additional radiation. In figure 4 the location of the SAA as measured by the ROSAT (Röntgensatellit) mission can be seen.

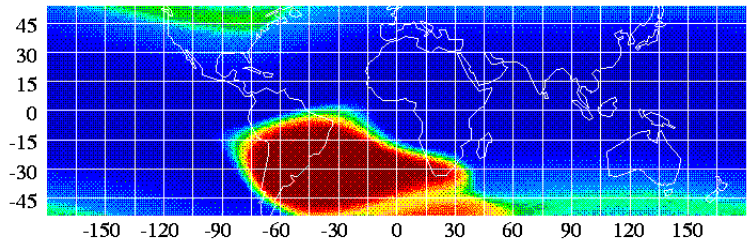

Figure 4: Image of the South Atlantic Anomaly (SAA) at approximately 560 km altitude. [4]

In order to test the hypothesis that the SAA was the cause of the OBC reboots of Delfi-n3Xt, a map with the locations of the reboot events was created as one can see in figure 5. This map was generated based on the frame counter which is reset to zero after every OBC reboot and is correlated to the Elapsed Time Counter (ETC). Although reboots have taken place in the SAA region, there is no statistical proof at all. Actually in figure 5 it can be seen that the locations of the reboots are very random and spread all over the Earth. In total 198 OBC reboots have occurred as of February 20th, 2014.

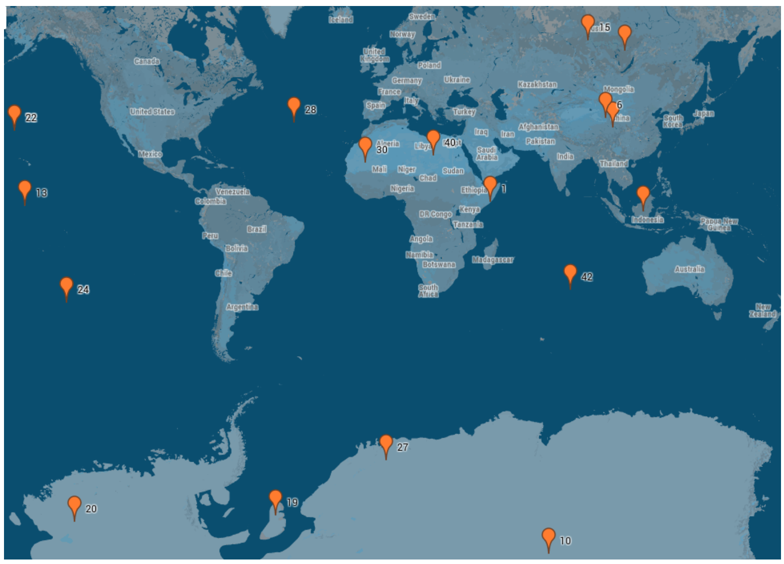

Figure 5: Locations of OBC reboot events of Delfi-n3Xt as of 10 December 2013. (MapLink, Google Maps)

Since the vast majority of the reboots cannot be correlated with the SAA, another reason needs to be found. However, it is quite a challenge to determine what is then the actual reason for the OBC reboots. Mainly because there is not much information about the I^2C recovery included in the telemetry. The Delfi-n3Xt OBC is unable to determine which subsystem - I^2C slave device - pulled low one of the I^2C lines. In order to determine which microcontroller held a bus line low exact timing is necessary and thus an external crystal is needed [19]. Nevertheless, it is most likely the OBC reboots occurred because of I^2C lock-ups since the OBC is one of the last subsystems to perform a power reset [20].

2.3 switch count errors

In the Delfi-n3Xt software switch counters are implemented to prevent infinite loops. However, it is uncertain how well the switch count logic actually performs. Since during Delfi-n3Xt operations there were some problems bringing the S-band transmitter (STX) back online. Only after setting the STX switch counter manually to zero the STX turned back on again. This worked even though the telemetry already indicated the switch counter was at zero. According to the activity flows the switch counter should have been set back to zero autonomously after a certain time. This indicates there is something wrong in the software or a bit flip occurred leading to this anomaly. The bit flip could have caused a value change outside the telemetry range. Nevertheless, although some anomalies occurred the switch count logic performed well for its primary purpose by preventing infinite loops.

2.4 update software parameters capability

It would have been beneficial for Delf-n3Xt if certain parts of the software could have been updated. For example, there was a problem with the pressure range conversion of the propulsion system and therefore not the whole pressure range could be measured. A simple update in the software could have fixed this error. Also the Maximum Power Point Trackers (MPPT) caused problems because of erroneous software/algorithms. Subsystems needed to be switched off because not enough power was generated at some point. Hence, faulty parameters have been usually not a problem of the OBC itself, but rather errors in individual subsystems.

It is therefore recommended that the CDHS of DelFFi will have the capability to update more individual parameters of subsystem software in orbit (see requirement 1). Of course this software parameter update capability can induce reliability issues, but would nevertheless be very beneficial because there always is a probability of unknown errors in the code. If some of these errors can be fixed during operations it might increase mission success of the payloads to a significant extent.

Full software update capabilities would require the use of boot loaders. Unfortunately, that is a very risky process and therefore not recommended for the DelFFi mission. Especially if the time and resource constraints of the project are taken into account and thus as many flexible parameters in the onboard software where appropriate should be implemented.

2.5 TELECOMMANDS PREPARATION & LOG

During the first phase of the operations of Delfi-n3Xt a few erroneous commands were prepared and transmitted to the satellite. For example, a thrust command was sent while ignition should have been performed. Also, a small error in the HEX code caused a wrong initialization vector at some point and switched the wrong subsystems on and off. Of course the operations of a satellite always involves some manual labor which is prone to errors. However, with the right procedures in place these mistakes can be prevented to a minimum. It was soon decided telecommands should be checked by at least one other satellite operator to prevent erroneous commands. In addition, commands should be ready at least 15 minutes before a pass, otherwise preparations are rushed and mistakes are easily overlooked. It would also be beneficial that several commands can be temporarily stored in the transmit program before a nanosatellite pass takes place. This would save time and prevent errors if several commands have to be assembled during the pass.

Nevertheless, this faulty commanding has shown the Delfi-n3Xt CDHS is inherently stable. Because the satellite decided autonomously to overwrite the received corrupted initialization vector. This procedure took place because it was mission critical to shut down two receivers at the same time, which should never be possible. Therefore, it can be concluded the system works well and should be implemented in DelFFi as well.

There is one recommended feature which should definitely be implemented in a future version of the telecommanding software (DIGIT). The program shall have the ability to automatically log transmitted telecommands. Since during Delfi-n3Xt operations there was sometimes uncertainty about what was actually sent. In some cases it was hard to determine if there were small errors in the commands which were transmitted and also information about the exact amount of transmit attempts for each command was often lost. It is therefore recommended that in future Delfi ground stations each time a command is transmitted the date, time, type, destination address and parameter content should be saved (see requirement 2). This telecommand information can either be stored in a log file or written to a database.

2.6 RADIO AMATEUR PARTICIPATION

One limitation which was experienced during the operations of Delfi-n3Xt was the lack of received data outside the pass visibility of the Delfi ground station. It was expected more radio amateurs spread all over the world would support the collection of Delfi-n3Xt telemetry. However, due to several reasons this participation was less than anticipated. In figure 6 an overview of the amount of frames received per day for Delfi-n3Xt can be seen. In total 249253 frames were received of which 46701 are identified as unique frames.

One of the main reasons was that the Delft Universal Data Extractor (DUDe) client initially had some performance issues and therefore some radio amateurs gave up or did not succeed in receiving Delfi-n3Xt telemetry. In addition, the bandwidth capabilities of most radio amateur receivers was too limited. Probably more data would have been received if the transmitters had operated on 1200 bit/s. In future missions more flexibility is desired to change the bandwidth of the transmitters during the operations or it needs to be reduced if the satellite is not in view of the Delfi ground station.

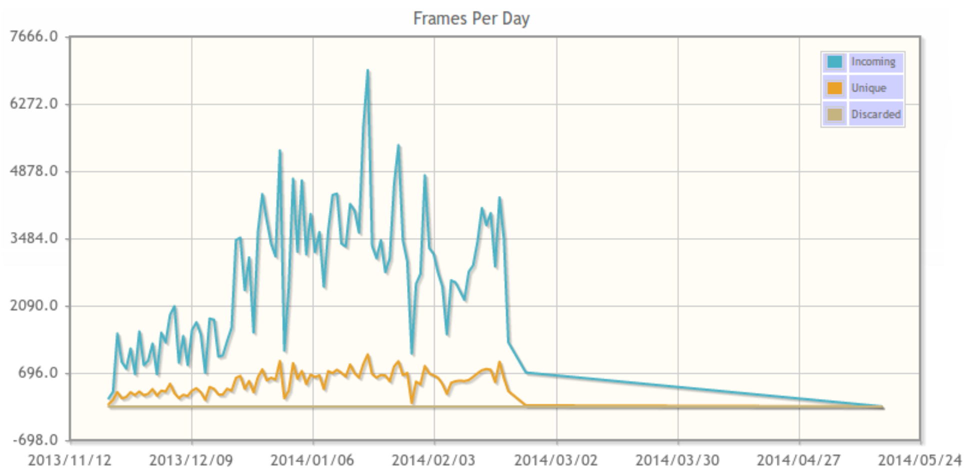

Figure 6: Overview of received Delfi-n3Xt frames per day.

Another important suspected cause of the limited radio amateur participation was the launch of several nanosatellite missions in a short amount of time. In the time span of just two weeks over 30 new nanosatellites were launched into orbit and therefore many radio amateurs probably focused their efforts and time on the nanosatellites they liked most, or with nanosatellites they have a personal connection with. For example, many Dutch radio amateurs participated in the Delfi-n3Xt telemetry collection from the beginning. Most likely other radio amateurs did the same for nanosatellites affiliated with their own country. In figure 7 and 8 the radio amateur coverage for Delfi-n3xt on a world map are displayed. These are all radio amateurs which subscribed before the 19th of November 2013 to participate in Delfi-n3Xt operations. Note that although they all registered for the Delfi database some eventually received much more data than others.

Note that an increase of radio amateur participation was noticed once the most important DUDe client issues were fixed and a few weeks had passed since the many nanosatellite launches. However, because nanosatellites are common nowadays and they are typically launched with many at the same time, it is recommended to make future Delfi nanosatellites such as the DelFFi nanosatellites not depended on radio amateur participation. This can primarily be achieved by not downlinking real time telemetry data only during a pass, but also historical telemetry of the hours before the pass with a certain time interval.

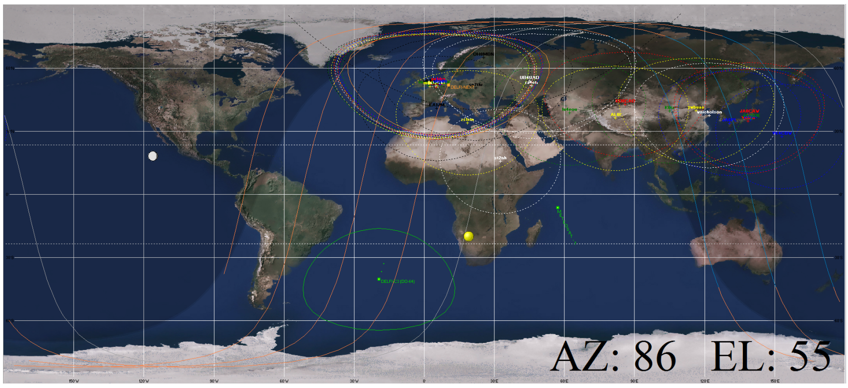

Figure 7: Delfi-n3Xt radio amateur coverage part 1. (WXTrack)

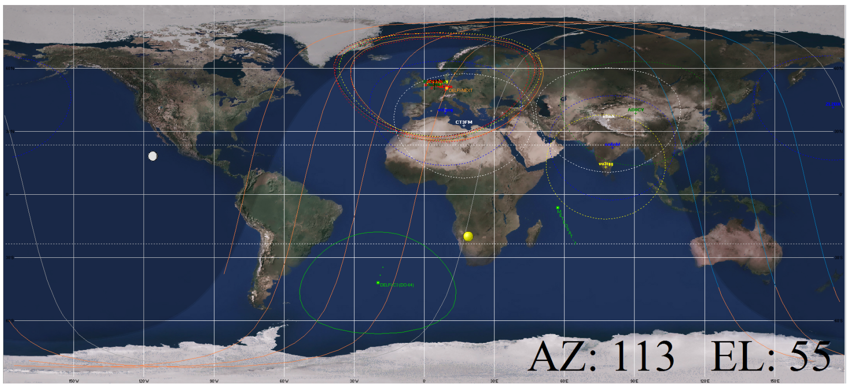

Figure 8: Delfi-n3Xt radio amateur coverage part 2. (WXTrack)

Another additional possibility is to start a strategic partnership with a few other nanosatellite developing universities in the world. In exchange for the services of the Delfi ground station, they support Delfi nanosatellite missions in return. Also, participation in ground station networks under development, such as the École Polytechnique Fédérale de Lausanne (EPFL) Satellite Control Software (SCS), Global Educational Network for Satellite Operations (GENSO) light and GAMANET should be investigated. One of the critical requirements will be that the software architecture should be compatible with the Delfi ground station. Or at least not too costly investments for new hardware need to be made to make the ground station compatible with the new ground station network. Note that the available ground station networks should be carefully evaluated, because it is uncertain how well these systems perform. If it is decided to implement a certain network for Delfi satellite operations, it needs to be ready for stable operations.

2.7 TELEMETRY FLEXIBILITY & EFFICIENT USAGE OF BANDWIDTH

After the LEOPS of Delfi-n3Xt, many parameters were less useful or not necessary to monitor anymore and thus quite some telemetry bandwidth was wasted in the remainder of the operations of Delfi-n3Xt. For example, once the deployment of the solar panels had been confirmed no continuous information about the deployment status was required anymore.

The foregoing and the arguments about the orbit data coverage in section 2.6 require a more flexible definition of the telemetry. Therefore, it is proposed to implement the functionality to redefine the telemetry, which is downlinked during operations of future Delfi nanosatellites. A design and implementation of this flexible telemetry functionality will be outlined in chapter 4. In order to achieve a more spread data orbit coverage a telemetry history playback functionality needs to be integrated as well. This will provide a higher data yield for parts of the orbit which are not covered by (participating) ground stations.

2.8 SOFTWARE INTERFACE CONTROL

The operator web interface used for Delfi-n3Xt was really helpful for the operators. However, it did have its weaknesses such as uncertainty about the validity of values of certain parameters and the same was true for the DUDe client. Usually these errors are caused by wrong conversion of the values in the backend of the telemetry processing. Currently, the definition of these parameters and keep them synchronized between the flight and ground station software systems during development seems to be a manually performed job. Errors in this conversion or mismatch of variable names might be prevented if the parameters are specified in for example special purpose Extensible Markup Language (XML) documents and continuously updated. If an update of parameter definitions takes place in either the CDHS or the ground station server, these XML sheets should be easy to convert to the actual software of the different systems involved. In this way the conversion of the parameters is better synchronized and the amount of invalid variables is reduced.

Note that a similar setup can be used to define the telecommands. Each telecommand field should specify properties such as an address number and a variable value range if applicable. These XML sheets can then also be loaded in an improved transmitter program and should make transmitting commands for the operators easier. At the same time errors can be prevented because a more user friendly interface can be designed, by automatically converting the XML specifications to drop down selection menus and backend conversion to HEX codes in the transmitter software.

2.9 TRANSPONDER TEST & SATELLITE MALFUNCTION

One of the last experiments to be performed onboard Delfi-n3Xt was the transponder test. On the 20th of February 2014 the first attempts to use the transponder were made. First of all, it is important to note that the transponder is located on the ITRX and hence the initialization vector was defined in such a way that the PTRX was put in receiving mode only. In this way potential interference of the PTRX was prevented and power budget problems avoided. Since the ITRX was set in transponder mode no telemetry was received during the transponder tests. Unfortunately, the transponder tests itself did not work, although the beacon could be heared. After ten minutes of transponder mode operations the ITRX was switched to transceiver mode and hence some telemetry was received in order to check all systems were still functioning nominally. Nonetheless, after a few telemetry packets were received the ITRX was intentionally switched to transponder mode again. All initialization vectors sent to the satellite were volatile commands. Therefore, at the next eclipse everything should have been set automatically to nominal operations again.

Another attempt to test the transponder was planned the next day. However, the ground station operators noticed the satellite was not transmitting anymore and no signal was received at all. In addition, it was discovered no telemetry was uploaded to the database since the transponder test. Under normal circumstances several radio amateurs should have received telemetry in a one day time span and thus it was quickly concluded Delfi-n3Xt was not transmitting at all or at least not nominally. After this conclusion was made, it was decided to abort all transponder tests and try to get the transceivers back online into transmitting mode. In the following days, initialization vectors were transmitted and also direct I^2C pass through commands to trigger a temporary signal burst were uplinked. These attempts to revive Delfi-n3Xt were not successful and hence another method to identify the satellite status needed to be found.

An alternative Delfi-n3Xt revival option, or at least to get a sign of life, was to try to receive the signal of the S-band transmitter. Hence, a Delfi-n3Xt S-band campaign was planned during which several commands were transmitted to let the S-band transmitter dump its data. The Delfi ground station and a few radio amateurs were tracking Delfi-n3Xt and listening to the S-band signal. Unfortunately, no trace of an S-band signal of Delfi-n3Xt was observed and thus no conclusions about the status of Delfi-n3Xt could be made. It is assumed the underlying problem is caused by the transmitter, onboard software or internal data communication. The Delfi ground station will remain automatically tracking Delfi-n3Xt and listening for signals.

Based on the experience with the malfunctioning of Delfi-n3Xt the following feature requests for the TRXVU are considered to be made to ISIS [21]:

- If the TRXVU has not received an AX.25 packet from the OBC within a few minutes, the transmitter needs to downlink an AX.25 frame every minute containing some key housekeeping parameters. For example, the main bus voltage level, the time since startup, time elapsed since last I^2C communication and the I^2C status. This feature would ensure that potential problems with the OBC and/or I^2C data bus can be identified. This is a software defined functionality and thus should not entail too much work to implement while improving failure analysis if necessary.

Some other features which might be useful to have available on the TRXVU are:

- The ability of the receiver microcontroller to process a direct to I^2C bus telecommand. This microcontroller should then act as an I^2C master for a single time. An encryption key shall be set in the non-volatile memory to resolve subsystem problems.

- A telecommand should be able to set the TRXVU in an AX.25 loop back mode in which uplinked packets are forwarded to the downlink directly. This functionality would provide radio amateurs the ability to use the satellite as a transponder if the satellite is malfunctioning. It shall be possible to turn off this loop back mode by means of a dedicated telecommand.

2.10 CONCLUSIONS & PROPOSED REQUIREMENTS FOR DELFFI

Although Delfi-n3Xt did not achieve the two year lifetime requirement the mission success rate is still very high. According to the predefined mission success criteria the total mission success rate is 84.3 % (see appendix C). The main reason being that almost all experiments were already successfully conducted before the satellite was malfunctioning.

Based on the lessons learned of Delfi-n3Xt operations, which were discussed in this chapter, the following requirements for the DelFFi mission have been defined:

1. Parameters in the OBC software shall be configurable by ground command. (SAT.2.6.2.1.3-I.01 [22]) (see subsection 2.4)
2. The Delfi ground station shall automatically log any transmitted telecommands. (GS.2.1.1-F.06 [22]) (see subsection 2.5)
3. It shall be possible to preassemble and temporary store telecommands before transmission.
4. The logged transmitted telecommands shall be accessible through the telemetry web interface.
5. Enable the downlink of telemetry history to the ground station.

Part I

IMPROVED CDHS ARCHITECTURE FOR DELFFI

3 DELFFI CDHS REQUIREMENTS ANALYSIS

In this chapter the requirements imposed on the CDHS of DelFFi will be discussed. The requirements listed in this chapter are derived from different sources. First of all the general QB50 mission requirements [23], Flux-ϕ-Probe-Experiment (FIPEX) [24] payload requirements and GAMALINK device [5] requirements are used as a reference. In addition, some reusable requirements of the Delfi-n3Xt mission are stated. Finally, DelFFi specific requirements are discussed and listed.

Many requirements are derived from third party subsystems and therefore should be critically analyzed. Especially because some requirements may conflict with DelFFi system requirements. Also, the requirements for the CDHS have evolved during the project and therefore a reflection of the requirements changes throughout the project, and how this affected the CDHS design is made as well. Note that not all DelFFi CDHS requirements are listed and analyzed in this chapter. Only the CDHS requirements relevant to the design of the flexible telemetry system, the wireless temperature sensor experiment and miscellaneous requirements which need critical analysis are discussed.

In the first section, a short analysis about the DelFFi data protocol selection is given. In section 3.2 considerations with respect to the GAMALINK device will be provided. The GAMALINK subsystem deserves a separate section in this chapter to provide some more clarity about the functionalities of this system which are planned to be used on DelFFi. An overview of these functionalities will enable the determination of which parameters need to be included in the telemetry. Hereafter, in section 3.3 requirements with respect to telecommanding are listed. Subsequently, in section 3.4 requirements relevant for the data acquisition are discussed. In section 3.5 requirements which are specifically related to subsystem interfacing are treated. Finally, in section 3.6 requirements which affect the monitoring function of the CDHS are specified.

3.1 DELFFI DATA PROTOCOL SELECTION

During development of Delfi-C3 it was discovered microcontrollers can pull down I²C lines indefinitely as long as power is retained.

Therefore the Delfi Standard System Bus (DSSB) monitors the I^2C interface of Delfi-n3Xt and can isolate subsystem errors from the main bus. In addition to I^2C errors isolation the DSSB also provides overcurrent protection. During the design phase of DelFFi a decision had to be made by the Delfi team whether to proceed with the former DSSB as was implemented in Delfi-n3Xt, or to use another available protocol standard for the data bus. Therefore a trade-off was performed taking into account several aspects in order to make the best decision with respect to DelFFi and future Delfi missions.

A few protocol standards were considered such as the Controller Area Network (CAN) which is the de facto standard in the automotive industry and is known for its reliability. Other protocol standards which have been evaluated were hybrids such as I^2C and CAN working together. After several team meetings it was decided to not use DSSB again, but migrate to the CubeSat standard. Important considerations to select the CubeSat standard were that more Commercial Off-The-Shelf (COTS) parts are available and hence development costs/required personnel would be reduced. Disadvantages are that there is less heritage of Delfi-n3Xt and its potentially lower reliability than DSSB.

3.2 GAMALINK DEVICE

The GAMALINK device will mainly be used by the DelFFi nanosatellites to support the Autonomous Formation Flying (AFF) package. This GAMALINK device will enable the exchange of position and attitude data between the nanosatellites Delta and Phi to support the AFF. The Inter Satellite Link (ISL) required to perform this exchange of information will be established over the S-band. A visualization of the concept of the GAMANET is illustrated in figure 9. In this chapter a small analysis about the GAMALINK device is performed, to gain a better understanding about the operations and telemetry involved in the usage of this device. This analysis will contribute to a more complete FlexTLM design.

As of 29/01/2014 Tekever indicated [25] the GAMALINK device is at Technological Readiness Level (TRL) 6. Also, the average power consumption is about 10% above the 1.5 W which is specified in the GAMALINK ICD [5] but Tekever is confident this can be reduced significantly by optimizing the Field-programmable Gate Array (FPGA).

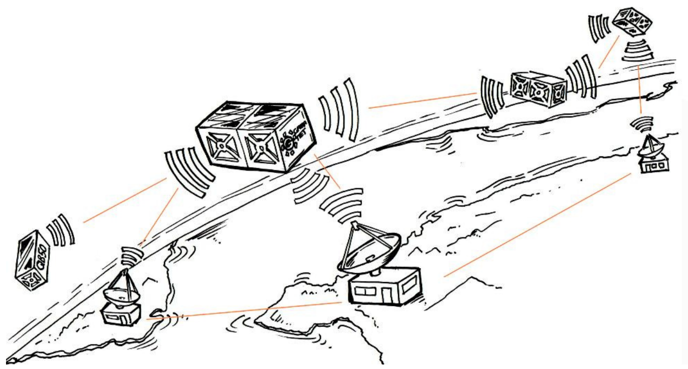

Figure 9: Illustration of the GAMANET concept. [5]

In addition to the ISL capability the GAMALINK will provide several other functions which will be implemented in the Delta and Phi nanosatellites, depending on the need of the DelFFi mission, and the time available to integrate these functionalities in the DelFFi platform. For example, there is also the possibility to relay data through other satellites to the Delfi ground station, which are using the GAMALINK device over the so called GAMANET. This ad hoc network enabled by GAMALINK devices will provide a great potential for a longer duration ground station coverage. According to the latest information the data can be relayed over a maximum of four nodes/satellites [25]. However, it still needs to be proven how much of the relayed data over the GAMANET will eventually reach the Delfi ground station and telemetry database.

The GAMALINK also includes a UHF/VHF transceiver which can be used as a backup for the TRXVU. Nevertheless, more systems engineering has to be performed before the Delfi ground station will be made ready for the GAMALINK device. It is recommended that a future DelFFi communications engineer evaluates the need and compatibility to use the GAMALINK as a backup option.

Another nice aspect of the GAMANET is that there are already many ground stations available in Portugal and especially Brazil which can potentially cover larger parts of the DelFFi orbits. Since there are also about eight other teams [25] which are participating in the GAMANET more ground stations become available as well. For example, teams in China and India also indicated they will be using the GAMALINK device in their QB50 satellites. The ground stations exchange data with each other over a kind of torrent network.

A capability of the GAMALINK which will most likely not be used for the DelFFi satellites is the OBC backup functionality. It would entail too much work and interfacing to make the GAMALINK OBC ready as a proper backup option. Also, science unit supporting software available on the OBC will not be used. Because full control over the science unit with the main DelFFi OBC and Electrical Power System (EPS) is desired. The available SD card storage on the GAMALINK will only be used for temporary storage for relayed data over the GAMANET and ISL, since there is already sufficient SD card storage which communicates with the DelFFi OBC directly.

3.3 TELEMETRY & TELECOMMANDING

In this section requirements with respect to the telemetry and telecommanding of DelFFi are analyzed. Especially the implications for the flexible telemetry design will be discussed. Verifications of the requirement content and proposed actions are outlined for each requirement.

3.3.1 *It shall be possible to request and transmit saved science and housekeeping packets from OBC memory separately. [ACCEPTED]*

Source: SAT.2.6-F-17 [22] / FPX-SW-0330 [24]

DISCUSSION The FIPEX payload requirements dictate this functionality. However, it is not clearly argumented why this functionality is required. The most likely explanation is that due to limited data bandwidth a lot less housekeeping data can be transferred to the ground if the telemetry is mixed with science data. This requirement will be met because the flexible telemetry approach as described in chapter 4 will support the separation of science and housekeeping packets in different AX.25 frames.

ACTION This requirement can be maintained because the flexible telemetry solution (see chapter 4) will inherently support this functionality.

3.3.2 *The ability to receive byte encoded command scripts with an error detection mechanism (at least CRC16) assuring consistency of the command script and feedback to the ground station (ACK/NACK). (QB50-SYS-1.5.7 and QB50-SYS- 1.5.11) [ACCEPTED]*

Source: FPX-SW-0050 [24]

DISCUSSION The FIPEX payload requirements request this functionality. These command scripts are executed on the OBC which controls the science unit. Therefore the command scripts integrity should be ensured by a CRC algorithm. Note that the command script data is already verified by the transceiver, but this Cyclic Redundancy Check (CRC) is also required to make sure there are no bit errors introduced when the command script data is send from the transceiver to the OBC.

action An additional requirement is proposed:

¨A telemetry field shall be reserved to include the acknowledgement of the last received command script.¨

3.3.3 *The satellite shall communicate a volume of at least 0.3 Megabits of science unit data per day to the ground station that is operated by the university providing the CubeSat. [ACCEPTED]*

Source: SAT.2.6-P-1 [22] / FPX-SW-0360 [24]

discussion The science data requirement drives the total bandwidth left for housekeeping data. In chapter 4 calculations are performed with respect to the estimated total data which can be received on average and whether still a reasonable amount of housekeeping data can be downlinked.

action It needs to be verified if the stated 0.3 Mbits of science unit per day is actually aligned with the recommended operations of the FIPEX. After analysis in subsection 4.1.1 the science data volume is indeed approximately 300 kbits per day. However, if one also considers all the time, position and attitude data packets which need to be tagged with each measurement, 500 kbits per day is actually required. It is important to take this fact into account when assembling the telemetry definition.

In addition, it needs to be verified if the requirement can be fulfilled by the DelFFi mission. This was analyzed in section 4.6 and it was concluded this requirement can be achieved. Depending on the mission phase and other needs even more science data can be downlinked if desired by the mission operators.

3.3.4 *VHF shall not be used for downlink. [ACCEPTED]*

Source: SAT.2.3-C-06 [22] / QB50-SYS-1.5.1 [23]

DISCUSSION Implementing a reliable S-band system for the DelFFi satellites induces more complexity, higher pointing accuracy requirements and overall higher risks. Because it is not allowed to use a VHF downlink on the DelFFi satellites according to this requirement the only viable alternative left is a UHF downlink.

ACTION The currently selected transceiver is the TRXVU [26] which uses a UHF downlink. Therefore the current selection of hardware is already compliant with this requirement.

3.3.5 *If UHF is used for downlink, the CubeSat shall use a downlink data rate of at least 9.6 kbps. [FURTHER ANALYSIS]*

Source: SAT.2.3-P-04 [22] / QB50-SYS-1.5.2 [23]

DISCUSSION UHF will be used as was discussed in subsection 3.3.4 and for the downlink a data rate of at least 9.6 kbps needs to be provided. It is, however questionable why QB50 would impose this performance requirement as long as DelFFi complies with the FIPEX science data requirement as stated in subsection 3.3.3.

ACTION The TRXVU allows data rates between 1200 bps and 9600 bps [26]. In section 4.1 the telemetry budget is analyzed and the received data volume estimates for 1200 bps are rather low. At the beginning of the DelFFi mission 800 kbits can be downlinked per day while nearing the end of the mission this is reduced to 200 kbits. Therefore a 1200 bps data rate is not really an option. However, for a data rate of 4800 bps the data volume which can be downlinked is 3 Mbits at the start of the DelFFi mission to 1 Mbits later during the mission. Hence, a data rate between 4800 bps and 9600 bps can be selected to find the right balance between data volume and link reliability. Therefore, a more extensive trade-off has to be conducted by a future DelFFi communications engineer.

3.3.6 *Parameters in the onboard computer (OBC) software shall be configurable by ground command. [ACCEPTED]*

Source: SAT.2.6.2.1.3-I-01 [22]

DISCUSSION It is highly desirable to be able to change parameters in the software of the nanosatellite during operations for tweaking certain subsystems, as was already discussed in section 2.4. During the operations of Delfi-n3Xt this was proven to be absolutely beneficial for the ADCS and OBC.

ACTION This requirement is currently part of the DelFFi requirements overview [22]. The exact parameters which need to be configurable by telecommand shall be identified. If the application layer is designed, a list of parameters which are potentially useful to change during DelFFi mission operations, needs to be created.

3.4 DATA ACQUISITION

In this section the requirements related to the data acquisition of the DelFFi satellites are analyzed. The data acquisition requirements can have consequences for the flexible telemetry and hence need proper discussion.

3.4.1 *The data acquisition control loop shall have a frequency of 0.5 Hz with an accuracy of at least 1 ms. [FURTHER ANALYSIS]*

Source: SAT.2.6.2.1.3-P-01 [22]

DISCUSSION This requirement is currently stated in the DelFFi requirements overview [22] and has been derived from the Delfi-n3Xt requirements. However, it is still under discussion what the actual data acquisition control loop frequency for DelFFi shall be. It is anticipated a data acquisition control loop frequency of 1 Hz should be feasible as well. Nevertheless, only with real hardware the exact control loop frequency can be determined because it all depends on I^2C packet request time delays etc.

ACTION Until the exact control loop frequency is determined, all systems which are designed based on this requirement shall be flexible to adjust this in a later phase of the DelFFi project.

3.4.2 *OBC shall have at least two independent mass memory units with at least 10MByte reserved for science unit science and housekeeping data. [ACCEPTED]*

Source: SAT.2.6-P-02 [22] / FPX-SW-0040 [24]

DISCUSSION The FIPEX payload requirements dictate this redundancy. This will enable dumping of historical data during passes over ground stations. Although two independent memory storage units are requested this is not a goal in itself. Failure tolerance for science data integrity is desired and therefore CRC algorithms should be implemented to ensure the integrity of the science data.

ACTION Note that more requirements will be derived for the interface with the Secure Digital (SD) cards. These requirements will depend on the chosen interface which will most likely be the Serial Peripheral Interface (SPI) [27]. There are other proprietary interfaces available as well which allow higher data throughputs. However, the use of these protocols requires relatively expensive licenses and the DelFFi satellites do not require such high data rates. The data rates supported by SPI are already significantly higher than the 100 kbps supported by the standard I²C bus.

3.4.3 *Every science packet shall be tagged with the position of the satellite at the time that the RDY line goes high (indicating that the packet is ready in the science instrument), accurate to within 1 km. Position error estimates shall be provided for each position tag. [DISCARDED]*

Source: SAT.2.6-F-06 [28] / QB50-SYS-1.5.11 [29]

DISCUSSION This functionality provides the scientists position information for the atmospheric models. However, this would be a challenging requirement to comply with, especially if the Global Positioning System (GPS) module on the GAMALINK does not work well. Maybe this information can also be tagged to the science packets on ground.

ACTION Note that this requirement has been discarded as of QB50 ICD issue 4 [30]. The main reason this requirement has been discarded is that the payload ICD should specify the required position accuracies.

3.4.4 *Every science packet shall be tagged with the real time that the RDY line goes high (indicating that that packet is ready in the science instrument), accurate to within 1 seconds. Time error estimates shall be provided for each time tag. [DISCARDED]*

Source: SAT.2.6-F-07 [28]/ QB50-SYS-1.5.12 [29]

DISCUSSION This functionality provides the scientists time information for the atmospheric models.

ACTION Note that this requirement has been discarded as of QB50 ICD issue 4 [30]. The main reason this requirement has been discarded is that the payload ICD should specify the required time accuracies instead of the general QB50 system requirements overview.

3.4.5 *The CDHS shall have an autonomous mode in which it controls the payloads: On/off; selection of data and data transmission to the ground, including priorities and retransmission, and; selection of different processing modes (e.g. different compression ratio's or techniques). [REDEFINED]*

Source: SAT.2.6-F-09 [22]

DISCUSSION Several functionalities are stated in this single requirement for the specific autonomous mode:

1. **The CDHS should automatically decide if the payload should be switched on or off.** Criteria shall be defined when the payload can be turned on or off.
2. **Automatic selection of which data needs to be transmitted to the ground.** Probably the underlying assumption is based on the fact that more data will be generated by the satellite than can be transmitted to the ground. It shall be further defined on which parameters the data is selected.
3. **The data transmission shall be based on priorities and retransmission has to be performed as well.** Retransmission implies that some type of acknowledgements of the ground are required. However, no active acknowledgement system will be implemented on DelFFi and thus missed data shall be requested for retransmission manually.
4. **Selection of different processing modes shall be possible.** It is questionable if different compression techniques are really

required and whether this does not complicate the software architecture unnecessarily.

ACTION It is recommended this requirement is split up in three new requirements and the last part of the requirement (4) is discarded. The requirements as proposed in the discussion about this requirement can be included in the DelFFi requirements overview.

3.4.6 *According to a priority set by telecommands or a pre-defined schedule, the CDHS shall collect packaged data and transmit it to the COMMS for telemetry downlink. [ACCEPTED]*

Source: SAT.2.6-F-10 [22]

DISCUSSION Setting a priority to collect packaged data and transmit it to the COMMS seems to be unnecessary, because it is desired housekeeping data will always be downlinked in sun mode similar to Delfi-n3Xt, and thus allowing other ground stations to collect telemetry as well. A pre-defined schedule can be used as well but then significant coordination is required by the DelFFi mission operators to determine which ground stations will actively gather DelFFi telemetry, where they are located and when the DelFFi satellites pass over the ground station concerned.

ACTION In subsection 4.3.4 the considerations with respect to scheduling for the DelFFi mission are discussed and it is concluded scheduling is highly desirable. Also, in subsection 4.2.7 about the flexibility of the CDHS it is determined all data can be collected in the data acquisition loop and hence no priorities set by telecommands are necessary.

3.4.7 *According to a request set by telecommands or a pre-defined schedule, the CDHS shall be able to retransmit a subset of packets to the COMMS for telemetry downlink. [ACCEPTED]*

Source: SAT.2.6-F-11 [22]

DISCUSSION This will be part of the functionality to downlink telemetry on request. When a DelFFi satellite is in the visibility of the Delfi ground station a command should be send between which

Elapsed Time Counter (ETC) times certain data is desired by the mission analysts.

ACTION More elaboration of this functionality will be provided in section 4.3.

3.4.8 *The CDHS shall provide in all modes to the COMMS a real-time status report of all critical system parameters. [ACCEPTED]*

Source: SAT.2.6-F-12 [22]

DISCUSSION Critical system parameters were already directly fed to the COMMS systems in Delfi-n3xt. No major obstacles to implement this in DelFFi are foreseen here. Nevertheless, in eclipse mode these critical parameters shall not be downlinked by default in order to save power.

ACTION In the DelFFi software ICD FlexTLM design, one AX.25 frame with the real-time status of all critical system parameters easily fits in the transmission sequence.

3.4.9 *The CDHS shall be able to operate in a real-time rather than historical transmission set by telecommand or a pre-defined schedule. [ACCEPTED]*

Source: SAT.2.6-F-13 [22]

DISCUSSION Critical system parameters were already directly fed to the COMMS systems in Delfi-n3xt. No major obstacles to implement this in DelFFi are foreseen here either.

ACTION This requirements can be adopted because the FlexTLM design already supports both real-time and historical transmission set by telecommand.

3.5 SUBSYSTEMS INTERFACES

In this section the requirements with respect to the interfaces of subsystems onboard DelFFi are analyzed.

3.5.1 *All critical CDHS components shall be radiation tolerant if feasible. [REDEFINED]*

Source: SAT.2.6-C-03 [22]

DISCUSSION Making all CDHS components radiation tolerant is hard to achieve. However, due to the shorter duration of the QB50 mission (8 months) than regular nanosatellite missions, radiation plays a less important role. Nevertheless, the CDHS should be able to survive the space environment and cope with possible radiation induced errors for a significant amount of time.

ACTION The CDHS architecture will be designed in such a way that it can deal with Single Event Effects (SEE).

3.5.2 *The CDHS shall verify and test the functionality of subsystems and payloads. [ACCEPTED]*

Source: SAT.2.6-F-02 [22]

DISCUSSION As the central system the CDHS should make sure the subsystems of the nanosatellite perform according to the requirements.

ACTION Checks shall be incorporated in the service and application layer software.

3.5.3 *The OBC shall be the I²C master of the FIPEX payload. [DISCARDED]*

Source: FIPEX ICD (section 2.2) [31]

DISCUSSION The FIPEX payload ICD defines this OBC setting.

ACTION Note that since FIPEX ICD version 1.5.3 [32] I²C is not available anymore and thus for science operations the Universal Asynchronous Receiver/Transmitter (UART) interface has to be used.

3.5.4 *The I^2C interface shall use a 7-bit address field. [DISCARDED]*

Source: FIPEX ICD (section 2.2) [31]

DISCUSSION The FIPEX payload ICD defines this address field.

ACTION Note that since FIPEX ICD version 1.5.3 [32] I^2C is not available anymore and thus for science operations the UART interface has to be used.

3.6 MONITORING

In this section the requirements concerning the monitoring of the CDHS are outlined.

3.6.1 *The CDHS shall collect whole orbit data and log telemetry every minute for the entire duration of the mission, where whole orbit data is defined as the following set of parameters: time, spacecraft mode, battery bus voltage, battery bus current, current on regulated bus 3.3V, current on regulated bus 5.0V, communication subsystem temperature, EPS temperature and battery temperature. The WOD packet format is provided in [SLRxxxx]. [ACCEPTED]*

Source: SAT.2.6-F-15 [22] / QB50-SYS-1.4.1 [23]

DISCUSSION The Whole Orbit Data (WOD) of DelFFi is requested by the QB50 mission. In order to support the science data of the payload and monitor the performance of the satellite, collection of housekeeping data is essential. Part of this requirement states telemetry shall be logged every minute. However, DelFFi system requirement *SAT.2.6.2.1.3-P-01* (3.4) already specifies telemetry shall be logged 30 times per minute and thus not many problems are foreseen here. Nevertheless, if the science customer actually wants whole orbit data with a certain interval this has implications for the telemetry definition. Therefore, the flexible telemetry definition as outlined in section 2.7 becomes all more important.

ACTION In subsection 4.1.2 it was calculated 82 kbits of WOD will be generated every day. Furthermore, in section 4.6 a feasibility analysis concerning the telemetry history was performed. It was determined an AX.25 frame containing WOD only needs to be send maximally once every ten seconds during Delfi ground station passes, to achieve full WOD coverage every day.

3.6.2 *The whole orbit data shall be stored in the OBC until they are successfully downlinked. [REDEFINED]*

Source: SAT.2.6-F-16 [22]

DISCUSSION This requirement is related to the previously stated requirement 3.6.1. However, there is no planned active acknowledgement system of packets planned for DelFFi and therefore the OBC does not know when it can delete transmitted whole orbit data.

ACTION It is most likely not necessary to delete data because more than sufficient storage is available if the anticipated SD cards of at least 2 GB are used, possibly even 16 GB. A functionality to erase (part) of the data from the SD cards shall be available to handle unexpected storage problems.

3.6.3 *The Science Unit Thermistor (SU_TH_GO, SU_TH_RET) can be used to monitor and log the science unit temperature but should not be used to decide on board about science unit command script execution. [DISCARDED]*

Source: FPX-RQ-14 [32]

DISCUSSION Restrictions on the usage of the thermistor to probably ensure the reliability of the instrument.

ACTION Note that this requirement is not in the FIPEX ICD since version 2.0 [24] anymore and thus this requirement has been discarded.

3.7 REVISIONS OF CDHS REQUIREMENTS DURING THE PROJECT

Throughout the DelFFi project several changes in requirements were made. Usually these changes could be attributed to third parties such as the science unit provider and the QB50 mission coordinator the

Von Karman Insitute (VKI). Therefore, in this section some of the main changes in requirements will be discussed.

All requirements which were proposed for modification or new requirements derived from analysis and designs made in this thesis were summarized. This list, together with a rationale for each requirement was provided to the DelFFi project manager at the end of this thesis. The list was discussed and the requirements which were accepted by the project manager were incorporated in the DelFFi requirements overview.

3.7.1 *Supported Data Protocols FIPEX*

Initially, the FIPEX ICD specified both I^2C and UART would be available on the science unit. However, at some point the ICD did not specify the I^2C interface anymore and thus only UART would be supported. It was concluded by the DelFFi team that in principle it should not pose a problem for the CDHS from a technical point of view. Nevertheless, it would imply a considerable effort to change certain parts of the OBC architecture to support communication over UART as well. Therefore, it was decided if this change in protocol support of FIPEX would lead to unacceptable delays in the overall development of the DelFFi satellites, a request to the TU Dresden to support I^2C again for FIPEX should be made. This would have been a reasonable request because previous ICDs claimed to support I^2C. In the end it was chosen to accept the interface change to UART and the current OBC design for DelFFi now supports the FIPEX UART interface.

3.7.2 *Requirement Numbering Inconsistencies*

During the DelFFi project many top level (QB50) and subsystems requirements (FIPEX) changed. In principle, this is not necessarily a problem, but the rules of requirement numbering and management were not always strictly followed in the supplied ICDs. For instance, new requirements were assigned to numbers previously used by completely different requirements. This fact required careful analysis by the DelFFi team to check the differences between ICD versions and always include the source/version number of the ICD for each stated requirement. Most of these problems could have been avoided, if numbers of discarded requirements were not allowed to be used for new requirements.

3.8 SUMMARY & STATISTICS

Note that mostly higher level CDHS requirements are discussed in this chapter. The lower level data bus requirements still need further investigation but were less relevant for the scope of this thesis such as the FlexTLM system. In table 1 the statistics of the DelFFi CDHS requirements analysis performed in this chapter are shown.

Table 1: Summary of DelFFi CDHS requirement analysis statistics.

Category	Accepted	Redefined	Further analysis	Discarded
Telemetry	5	0	1	0
Data acquisition	5	1	1	2
Subsystem interfaces	1	1	0	2
Monitoring	1	1	0	1
Total	**12**	**3**	**2**	**5**

4 FLEXIBLE CDHS & TELEMETRY

Quite some telemetry bandwidth was wasted during the operations of Delfi-n3Xt as was discussed before in section 2.7. Certain telemetry parameters were simply not necessary to monitor anymore after a while, or in the case of specific subsystems which were switched off. Also, some housekeeping parameters were not really critical to monitor and could be updated less frequently (e.g. temperature data), while other parameters would benefit of a higher update frequency (e.g. ADCS data). Efficient usage of telemetry bandwidth is very valuable since the DelFFi mission will generate a significant amount of science data which can be downlinked. Especially if one considers the limited radio amateur participation of Delfi-n3Xt as was discussed in section 2.6 and therefore the most optimal definition of housekeeping data to be downlinked should be used during each phase of the DelFFi mission.

Another desirable function would be to include/mix data generated - both housekeeping and science data - of the previous orbits in the telemetry with a certain interval. In this chapter a more flexible implementation of the telemetry definition onboard the DelFFi nanosatellites will be proposed and designed.

The first section will provide an analysis about the expected telemetry budget for the DelFFi mission. In section 4.2 the best method to incorporate reconfigurable telemetry in DelFFi or nanosatellites in general will be discussed. Special consideration to avoid potential faults in the telemetry due to the reconfigurability will be given. In section 4.3 an analysis will be made about how to downlink the telemetry of previous orbits during a ground station pass. Subsequently, in section 4.4 the design of the DelFFi software ICD will be discussed incorporating the flexible telemetry design. In section 4.5 the implications of FlexTLM for the ground station will be outlined. Hereafter, in section 4.6 the downlink feasibility of telemetry history will be discussed. Finally, in section 4.7 a design for the FlexTLM assembly software will be proposed and a conclusion including FlexTLM related requirements will be provided in section 4.8.

4.1 TELEMETRY BUDGET ANALYSIS FOR DELFFI

In this section a preliminary analysis is made about the estimated average telemetry data size which can be downlinked during the

DelFFi mission. Based on this analysis the need for a flexible telemetry system will be proven. The expected average access times and durations for DelFFi with the Delfi GS were calculated with the Analytical Graphics (AGI) Systems Tool Kit (STK).

In the simulations it is assumed the satellites can only rely on the Delfi GS, which leads to a smaller estimated telemetry data size than will probably be the case if other GS support the DelFFi mission as well. At this phase of the QB50 mission the exact orbit insertion characteristics are unknown. The current QB50 systems requirements document [23] specifies that the expected initial orbit altitude will be between 350 - 400 km ±7 km with an inclination of 98.6 ± 0.08°. Furthermore, in STK the orbit is assumed to be circular and the epoch is set at the first of January 2016. The average contact time per day of DelFFi with the Delfi GS was computed over a two week time span. These calculations were performed for different orbital altitudes due to the orbital decay during the DelFFi mission.

The following assumptions have been made to calculate the expected real content data which can be received successfully at the Delfi GS during an average DelFFi mission day:

- The factor due to **packet losses** and a **weak signal** during (parts) of a pass: η =0.8 [-]
- **Data rate** of TRXVU: r=1200, 4800, 9600 bit/s
- Actual **throughput** of data (AX.25 overhead and packet group identifiers, also some time is required between AX.25 frames for the Phase Lock Loop (PLL) of the demodulator): ϑ =0.7 [-]
- A **minimum elevation** angle of 10 degrees is used. (According to the DelFFi link budget [33])

The following equation can be derived to calculate the average amount of telemetry, which can be received by the Delfi GS during a DelFFi mission day:

$$B = \Lambda \eta r \vartheta \tag{1}$$

In which Λ represents the average contact time per day as was calculated with STK for different initial orbital altitudes. In figure 10 the estimated telemetry budget for DelFFi with respect to varying orbital altitudes and for different data rates can be seen. A DelFFi communications engineer still needs to conduct a trade-off in the future, to decide whether a data rate of 9.6 kbps will be used on the TRXVU, or a lower data rate for radio amateur compatibility as is discussed in subsection 4.2.1. For now, it will be assumed a data rate of 9.6 kbps will be implemented on DelFFi because this would enable the Delfi GS to receive much more telemetry and requirement QB50-SYS-1.5.2 (3.3.5) also specifies 9.6 kbps shall be used on the transmitter.

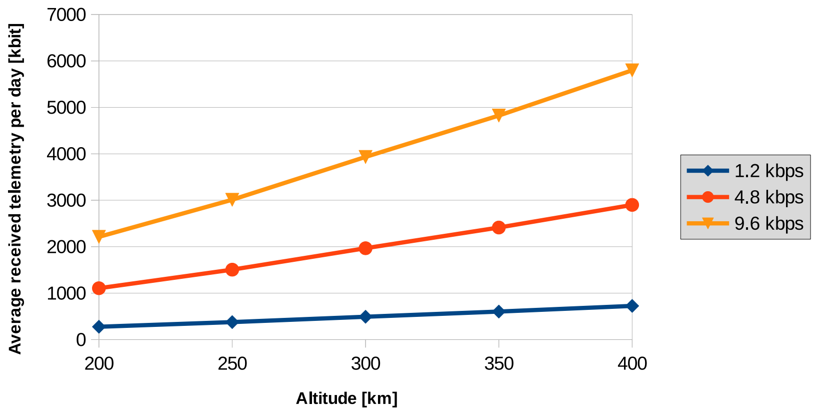

Figure 10: Estimated average received telemetry per day for the DelFFi mission calculated for different orbital altitudes.

According to requirement QB50-SYS-1.5.2 (3.3.3) at least 300 kbits of science data shall be downlinked every day to the ground station of the university operating the satellite. Hence, if this is subtracted from the estimated average received telemetry per day for a 9.6 kbps data rate setting as displayed in figure 10, this will initially take up approximately one twentieth of the total telemetry budget, while nearing the end of the DelFFi mission this will increase to one seventh of the telemetry budget. Therefore, this fact alone requires a FlexTLM solution to receive the most relevant housekeeping data.

4.1.1 *Science Data Volume*

It is stated and assumed 300 kbits of science data needs to be downlinked every day (3.3.3), although in reality there is actually a larger data volume required. The main reason is because for each measurement the time, attitude and position information needs to be tagged to the science data of the FIPEX and Surface Thermal Monitors (STM) as well, as is defined in requirement FPX-SW-0320 [24].

In requirement FPX-SW-0360 [24] the following information about the recommended science operations is stated:

- Science state time of FIPEX t_{FIP}: 25 %
- Measurement interval of FIPEX m_{FIP}: 6 seconds
- Science state time of STM t_{STM}: 50 %
- Measurement interval of STM m_{STM}: 120 seconds

Furthermore, the following data volumes of the different data packages were defined in the DelFFi software ICD [34]:

- Data volume FIPEX d_{FIP}: 56 bits

- Data volume STM d_{STM}: 72 bits
- Time (32 bits), position (16 bits) and attitude (16 bits) data d_{tag}: 64 bits

The amount of science data generated daily by the FIPEX instrument can be calculated by the following equation:

$$D_{FIP} = \frac{(24[hr] \times 3600[s] \times t_{FIP})}{m_{FIP}} (d_{FIP} + d_{TAG}) \tag{2}$$

The amount of data daily generated by the STM can be calculated by the following equation:

$$D_{STM} = \frac{(24[hr] \times 3600[s] \times t_{STM})}{m_{STM}} (d_{STM} + d_{TAG}) \tag{3}$$

Table 2 was generated by calculating equation 2 and 3 using the input parameters which were mentioned earlier in this subsection. It can be seen in table 2 that in total approximately 500 kbits of science and tagging data is generated every day.

Table 2: Estimated daily science and tagging data volume of the FIPEX instrument.

Parameter	**Description**	**Daily data volume [kbit]**
D_{FIP}	FIPEX science data	201.6
D_{STM}	STM science data	25.92
$D_{FIP,TAG}$	FIPEX tag data	252
$D_{STM,TAG}$	STM tag data	23.04
$D_{Science}$	**Total**	**502.56**

4.1.2 *Whole Orbit Data (WOD) Volume*

The WOD requirement QB50-SYS-1.4.1 / 3.6.1 specifies basic housekeeping data of every minute of previous orbits has to be downlinked. Simply multiplying the amount of minutes in a day (24 x 60) with the 57 bits of a basic housekeeping data packet [23] gives a total size for the WOD of 82.08 kbits per day. The WOD volume is limited, so the WOD requirement does not seem to have major implications for the telemetry budget.

4.2 RECONFIGURABLE TELEMETRY

In order to achieve Flexible Telemetry (FlexTLM) capabilities during a nanosatellite mission lifetime, the right method to implement reconfigurable telemetry should be found. There are several possible methods to make this reconfigurable telemetry available for DelFFi. However, one of the key issues is that the reconfigurability should never endanger the mission by permanently corrupting the telemetry and thus no data can be decoded anymore. Therefore, fault prevention will be an essential point of concern when designing the reconfigurable telemetry implementation.

4.2.1 *Data Link Layer Protocol*

One important reconfigurable telemetry constraint is set by QB50 requirement QB50-SYS-1.5.13 [23]. This requirement specifies that QB50 satellites shall use the AX.25 protocol. The AX.25 is a data link layer protocol, which is designed for radio amateurs to ensure link-layer compatibility between stations [15]. In table 3 the information frame structure of the AX.25 protocol is shown. Some of the acronyms used in table 3 are the Protocol Identifier (PID) field and Frame Check Sequence (FCS) field.

Table 3: Information frame structure of the AX.25 protocol. [15]

Flag	Address	Control	PID	Info	FCS	Flag
01111110	112/224 bits	8/16 bits	8 bits	N*8 bits	16 bits	01111110

The design of the reconfigurable telemetry shall adhere to the standards of the AX.25 protocol. If DelFFi would use a custom protocol this could save some data overhead. For Delfi-n3Xt it was considered to use the DelfiX protocol which would save approximately 10 to 20 % in data overhead. However, preference was given to radio amateur compatibility [35].

Nowadays, many radio amateurs do not use Terminal Node Controllers (TNCs) anymore which only support AX.25 packets. It could be considered to use the FX.25 protocol for DelFFi which provides a Forward Error Correction (FEC) capability. This could enable the Delfi GS to receive more decodable frames of the DelFFi satellites. Nevertheless, for the analysis in this chapter it is assumed DelFFi will stick with the AX.25 protocol to comply with QB50 requirement QB50-SYS-1.5.13 [23] as stated earlier. Since the FX.25 protocol encapsulates AX.25 frames with an FEC compatibility with existing AX.25 equipment is maintained [36]. Hence, if at a later stage of the DelFFi project, it is decided to use the FX.25 protocol this has no major implications for the flexible telemetry design.

Another limiting factor is the data bandwidth which will be used for DelFFi. If a too high data bandwidth will be selected this means less radio amateurs will be able to receive telemetry. The main reason is that not all radio amateurs have the equipment to support these data rates as was already discussed in section 2.6 and therefore it might be better to optimize the telemetry definition for 1200 bit/s. However, for DelFFi a UHF downlink (3.3.4 / QB50-SYS-1.5.1) will be used and it needs to provide at least 9.6 kbps (3.3.5 / QB50-SYS-1.5.2).

4.2.2 *Fault-tolerant Telemetry Definition*

The first question is whether all telemetry parameters should be able to be turned on or off. One might consider that some critical parameters such as the EPS status should always be included in the telemetry frames definition. Hence, these critical parameters cannot be overridden either on purpose or by an accidental/corrupt telecommand and thus situations in which essential housekeeping data is lacking in the telemetry frames, can be prevented by implementing a partly fixed telemetry definition.

However, it is questionable if it is really necessary to have a partly fixed telemetry definition, because as long as the receiver is not switched off one can try to recover the telemetry definition to default settings. By allowing a customizable telemetry definition, complexity is avoided and ultimate flexibility is provided to the satellite operators. Nevertheless, it shall always be possible to telecommand the satellite to a default telemetry profile as an extra mechanism to prevent unforeseen failures. Eventually, most reliability can be achieved by testing the FlexTLM software extensively for many possible cases.

4.2.3 *Telemetry Flexibility Architecture*

It is essential to know what data each telemetry frame exactly contains to successfully decode the telemetry. There are several options to implement a reconfigurable telemetry architecture:

- **Time defined frame identifiers:** The telemetry processing system will know the AX.25 frame sequence and AX.25 frame definition of all possible frames at each point in time, and therefore each time a change to the telemetry definition is made, this shall be logged in the database. During processing of received telemetry the ETC data of the AX.25 frames shall be correlated with the telemetry frame definition at that particular point in time according to the database. Note that this method is sensitive for errors if telemetry configuration changes are not properly (automatically) logged in the database. If these configuration changes are not properly logged, the telemetry processor cannot find the appropriate definitions and does not know how to decode the telemetry.

- **Unique frame identifiers:** In this system unique AX.25 frame identifiers are assigned to each new telemetry profile. Each time a new AX.25 frame is defined a new unique identifier will be assigned. The AX.25 frame identifier which will be replaced in the telemetry will be discarded. In this way, complexity will be reduced and potential faults will be mitigated. Of course one still needs to log the definition of each frame identifier, but at least this system is not sensitive to time errors as is the case with time defined frame identifiers. However, a disadvantage is that one has slightly more overhead in the frames because unique identifier numbers need to be included in the frames.
- **Packet group identifiers:** Packet groups are defined as groups of similar parameters. Including the packet group identifiers right after the header of AX.25 frames implies these frames can contain different content over time. Nonetheless, because the decoding information - the packet group IDs - are included in the info field this shall not pose a problem for the telemetry processing system. This method does create some additional overhead because between 100 to 200 bits are necessary in each AX.25 frame to define the packet groups included.

Although time defined frame identifiers have slightly less overhead than unique frame identifiers they are more sensitive to errors. Therefore, reliability is chosen over data bandwidth and hence unique frame identifiers would be a better option to tag frames (see requirement 1). In the case of unique frame identifiers probably more bits need to be allocated for the AX.25 frame ID to provide sufficient unique frame possibilities during a nanosatellite mission. Even though this method would save data bandwidth it has some implications for the operations of a FlexTLM nanosatellite because each time a new AX.25 frame is defined the frame sequence tables need to be updated as well. If it is assumed five frame sequence tables will be implemented, then in theory five frame sequence table update telecommands have to be sent if new AX.25 frames are defined. More can be learned about the concept of frame sequence tables in subsection 4.2.5.

Packet group identifiers would reduce the risk of a discrepancy between the AX.25 frame definition of the nanosatellite and the telemetry processing system. Although AX.25 frame definitions telecommand updates could in principle be verified by means of telemetry requests, it is unacceptable data cannot be decoded or is wrongly interpreted. Therefore, it is recommended packet group identifiers are used for FlexTLM and the packet group identifier overhead in the AX.25 frames is accepted. If the DelFFi software ICD indicates it is difficult to fulfill the data downlink requirements it might be considered to re-evaluate the use of unique frame identifiers to save on AX.25 frame overhead. Note that in the current analysis of 4.6 this is not the case and thus FlexTLM can stick with packet identifiers. In table 4 a summary of the telemetry flexibility architecture options is provided.

Table 4: Summary of telemetry flexibility architecture options.

	Reliability	**Frame overhead**	**Operations**
Time defined frames IDs	Low	Low	Medium
Unique frame IDs	Medium	Low	High
Packet group IDs (selected)	High	Medium	Low

4.2.4 *Telemetry Profiles*

In order to simplify procedures it is recommended subsystems define different telemetry profiles which will be pre-programmed on the OBC. For example, each subsystem telemetry profile will define the parameters with corresponding update frequencies which shall be downlinked. These telemetry profiles can range from very basic telemetry profiles with just a few parameters and a low update frequency, to detailed telemetry profiles with all parameters and a very high update frequency. To design these telemetry profiles a software ICD was made by requesting the responsible subsystem engineers what parameters there subsystem will generate. Each parameter should also indicate the measurement frequency a mission analyst of the relevant subsystem would eventually need as a minimum to make some sense out of the data. Also, the maximum update frequency which is desired if the subsystem needs to be examined in detail. The normal update frequency is somewhere in the middle of the minimum and maximum update frequency and thus this parameter represents a compromise between the two. In table 5 the concept of these different telemetry profiles is outlined.

Table 5: Subsystems ICD data example micropropulsion.

Parameter			**Value**		**Update frequency [Hz]**		
Name	**Bit size**	**Unit**	**Min.**	**Max.**	**Min.**	**Normal**	**Max.**
Chamber temperature	12	°C	-40	500	0.1	1	30
PCB temperature	8	°C	-40	80	0.1	1	1
Valve voltage	8	V	0	12	0.1	1	30

Telemetry profiles can contain parameters which need to be updated less frequently than the telemetry downlink transmission interval. Hence, some parameters do not need to be included in the AX.25 frame definition for each transmission event. This fact can simply be ignored and it is accepted the same parameter measurement is kept in the telemetry definition for a few consecutive transmission events. However, in this case some data bandwidth is wasted because of non-essential parameter measurements in the telemetry. It can also be solved by alternating between unique AX.25 frames which have slightly different telemetry profiles and are alternated in a sequence of transmission events. Each unique AX.25 frame identifier includes some low update frequency parameters in its definition. Nevertheless, this solution can become much more complex and thus it needs to be determined - when all telemetry parameters for DelFFi are listed - how much data bandwidth can be saved by not including low update frequency parameters in all AX.25 frames. It is recommended to not alternate with different telemetry profiles if it turns out these data bandwidth savings are marginal. In the remainder of this chapter a design to control and alternate between different AX.25 frames will be outlined.

4.2.5 *Frame Sequence Table*

If several AX.25 frames are defined for downlink the satellite needs to be aware in what sequence and how often these frames shall be transmitted. It is envisioned to use a frame sequence table to control this transmission sequence (see requirement 3). This transmission sequence indirectly controls the update frequency of each parameter as well. In table 6 an example of how such a frame sequence table will look like is provided. If it is assumed one AX.25 frame is transmitted every second, a unique frame ID needs to be assigned to each second in the transmit sequence. The transmission of this defined frame sequence will repeat itself after every completed cycle. In the example of table 6 the total defined transmission time of the frame sequence table is 10 seconds based on a transmission frequency of 1 Hz per frame. Note that this total transmission time can be adjusted to the needs of the operators. The contents of the AX.25 frame sequence table can be controlled by means of telecommands during any phase of the mission.

Table 6: AX.25 frame transmit sequence table example for a 1 Hz transmission frequency.

Time [s]	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
AX.25 frame ID [#]	1	1	2	1	1	3	1	1	1	4

4.2.6 *Control Method for Content of AX.25 Frames*

The content of all AX.25 frames has to be controlled according to the mission analysts needs at each point during the mission. There are several methods to control the contents of these AX.25 frames:

1. **Manually:** In case of the manual method, the mission operators have to change the content of an AX.25 frame by means of a telecommand. An example of such a telecommand can be seen in table 7. There is no intelligent decision by the satellite what data needs to be downlinked. This method will be a bit more operation intensive because the operators have to actively control the telemetry definition each time the satellite operates in a different mode.

2. **Autonomously:** The autonomous method lets the satellite decide what telemetry should be transmitted corresponding to the mode of the satellite. This semi-intelligent telemetry control is pre-defined by a set of rules. For example, if the satellite is in science mode some AX.25 frames with payload data will be included. Or in case of the formation flying experiment detailed data about this mode will be included in the AX.25 frames.

3. **Semi-autonomous:** A probably more realistic method of controlling the content of the AX.25 frames is a combination between the manual and autonomous method. This semi-autonomous method still provides full control to the satellite operators but also reduces their workload. Especially since the DelFFi satellites will have to operate according to schedules, it is desirable different modes are auto correlated to telemetry profiles (see requirement 2). The satellite operators can still override the AX.25 configuration if the pre-defined rules do not provide the desired telemetry. It is foreseen each mode has its own frame sequence table which satellite operators can modify. In this way the most optimal telemetry configuration is always defined for each operating mode.

The AX.25 frames should be assembled by packets with similar parameters and desired update frequencies. These parameters are grouped and assigned to different packets with a unique ID when the satellite is developed. Each header of a transmitted AX.25 frame includes the sequence of the packet IDs assembled in the respective AX.25 frame. For both AX.25 frame IDs and packet IDs 8 bits are allocated because this provides 2^8=256 unique IDs which is believed to be sufficient. The exact numbers of allocated bits for IDs should be defined later, by taking into account the actual amount of different packets required by the DelFFi mission and the storage available.

An alternative to explicitly stating the packet IDs which need to be included in an AX.25 frame, as shown in the telecommand example of table 7, is using an initialization vector. This AX.25 frame initialization vector telecommand will simply include a sequence of bits, which is compared with a predefined fixed sequence of packets. Which method will be implemented depends on the preference of the software engineer. In terms of telecommand bit size it probably does not make much of a difference. If it is assumed 20 packets are defined for a typical AX.25 frame then 160 bits are necessary to describe the packets to be included in the AX.25 frame. Consider in total there are about 120 unique packet IDs defined, so 120 bits would be necessary to describe an AX.25 frame by means of a bit sequence. In the end which method is more efficient, depends on the amount of packet IDs which need to be described for each AX.25 frame.

Table 7: Example of a fictitious telecommand for AX.25 frame content control.

	Bit size of content	Telecommand bit size
Destination	0	8
AX.25 frame ID 9	10	8
Packet ID 5	82	8
Packet ID 4	70	8
Packet ID 8	120	8
Packet ID n	510	8
CRC	0	16
Total	**792**	64

It could be decided the parameters included in each packet are also controllable by means of telecommands providing ultimate flexibility. However, in that case time defined packet identifiers similar to the time defined frame identifiers as described in subsection 4.2.3 need to be implemented. Time defined packet identifiers for individual parameter control are the only option otherwise the header information will become unacceptably large. For example, in case if unique packet identifiers would be used, assume about 100 parameters are included in one AX.25 frame and the CDHS gathers 1000 unique parameters. Then 2^{10}=1024 so 10 bits are necessary to describe each parameter in the AX.25 frame or 1000 bits to define all 100 parameters.

This implies there is almost no space in the AX.25 frame left to include the actual content of the parameters. Therefore, unique parameter identifiers are not an option but time defined packet identifiers were already determined to be a reliability risk in subsection 4.2.3. Hence, packet groups shall not be able to be changed in-orbit and packets will be fixed during development of the satellite. In figure 11 the Design Option Tree (DOT) of the flexible architecture as outlined in this section can be seen. This DOT summarizes all evaluated design options of this section and highlights the chosen options in orange.

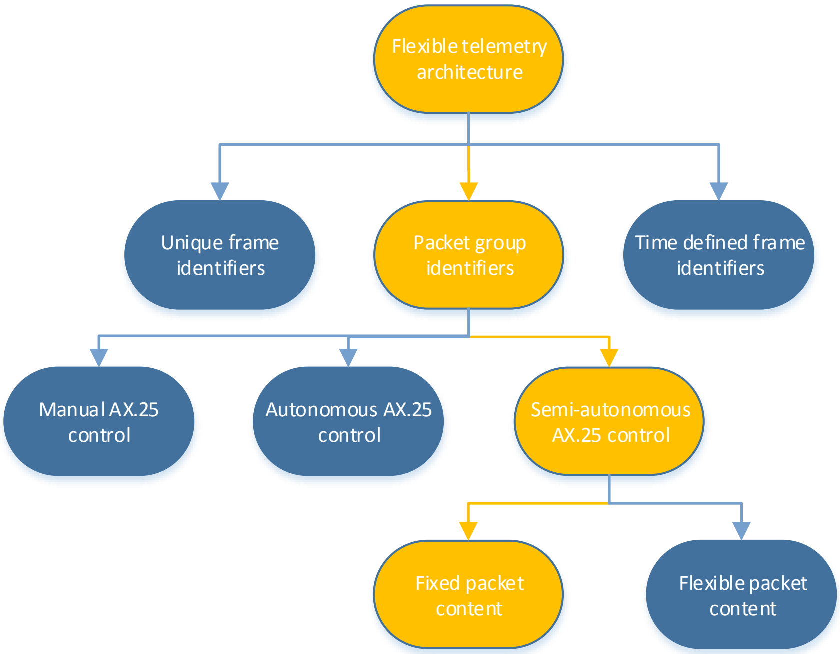

Figure 11: Design Option Tree (DOT) of the flexible telemetry architecture. Items highlighted in orange are chosen for the FlexTLM approach.

4.2.7 *CDHS Flexibility*

Besides the telemetry frame flexibility architecture, another design decision has to be made. Not only the telemetry definition itself can be made flexible, but also the CDHS in terms of the data acquisition loop. For example, it can constantly adapt to the set telemetry definition by the mission operators. In this case some parameters will not be acquired anymore and new parameters are included or updated less/more frequently in the acquisition loop. It is questionable if this is desirable from a reliability and telemetry history capability point of view, because one can simply acquire all possible data continuously with a maximum desired frequency as specified by the subsystem specialists.

Note that the acquisition loop itself does not need to run at the maximum frequency, but rather accepts an array of the parameters history. Higher frequency data has to be gather by the subsystem itself by changing the internal data acquisition loop timing and package this data. For example, this was done by the $T^3\mu PS$ propulsion system on Delfi-n3Xt. The OBC determines what data should be included in the telemetry frames and which data should not be included. However, this might induce a higher load on the system and it might turn out the bandwidth of the bus is not sufficient to acquire the complete data set. Hence, preliminary calculations were made about the anticipated amount of data generated by all subsystems.

From table 4.4 it can be derived a maximum of 6176 bits of data packets will be acquired by the OBC data acquisition loop per second. Note that this is the currently anticipated amount of data and this does not include I^2C overhead. Also, this amount of bits will increase if more parameters are identified for individual subsystems and I^2C overhead is taken into account. Therefore, this should be multiplied with a safety factor of two which leads to 12352 bps. The I^2C bus on Delfi-n3Xt attained speeds of about 50 kbps so the data acquisition loop should be able to acquire all data continuously. Nonetheless, if approximately 12 kbps need to be written to the SD card this will take 45 ms of execution time (writing a block of data to the SD card of 512 bytes takes 13 ms ± 2 ms) [27]. Note that creating and closing files on the SD card will also require execution time, but this does not have to be performed every second. It might be more efficient to temporary store the collected data in a buffer and write it to the SD card every few seconds.

4.3 TELEMETRY HISTORY PLAYBACK DEFINITION

It would be of great value if during a ground station pass data produced in the hours before the pass can be downlinked. In this way data logged while in eclipse, or during orbits when the nanosatellite is not in the visibility of ground stations will be available to mission operators and analysts. It is also specified in requirement SAT.2.6-F-15 (3.6.1) that whole orbit data shall be logged by each participating QB50 nanosatellite. In addition, the CDHS needs to be able to retransmit packets to the COMMS for retransmission as defined by requirement SAT.2.6-F-11 (3.4.7). However, there are many possible ways this telemetry history can be arranged. First of all, the telemetry history shall be an optional feature and not be enabled by default (see requirement 5). This is important because then there is always a capability to monitor the satellite in real-time if desired, without losing any update frequency resolution in the data. In this section the method for controlling the telemetry history playback definition will be outlined.

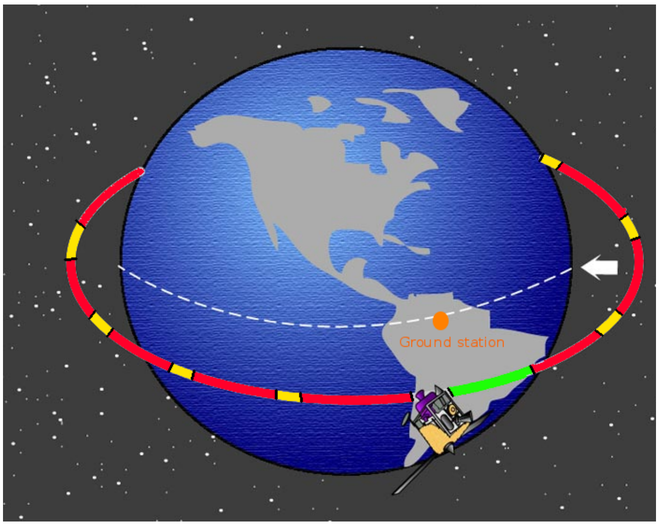

Figure 12: Concept of telemetry history definition orbit. [Adapted, 6]

In figure 12 the concept is illustrated of how the telemetry history definition is envisioned. In the green part of the orbit, when the satellite is in view of a ground station - in this case somewhere in South America - chunks of data of the previous hours of the pass, indicated as yellow (bright) parts of the orbit, will be downlinked. The red (dark) parts in the orbit will be gaps in the telemetry history, which do not fit in the total data bandwidth available during the ground station pass.

4.3.1 *Parameters to be Controlled*

It is desirable that telemetry AX.25 frames of past orbit data are alternated with telemetry AX.25 frames containing real-time telemetry. Depending on the need of the DelFFi mission at specific points in time, different ratios for the amount of past orbit telemetry frames versus real-time telemetry frames can be set by a telecommand. It is unlikely past orbit telemetry spanning a considerable amount of time can be downlinked and therefore certain decisions and compromises need to be made regarding the telemetry history definition. One of the biggest complicating factors is that there are many parameters which need to be set and taken into account. The following questions need to be asked every time when the telemetry history playback is redefined:

- What is the interval required between different telemetry sets? Note that a telemetry set is defined as an array of consecutive AX.25 frames of housekeeping or science data.
- How many AX.25 frames are incorporated in a measurement set?

- What is the amount of time back/start time the data retrieval should go back to?
- How many successful AX.25 frames can typically be received during a ground station pass?

The telemetry history definition is visualized in figure 13.

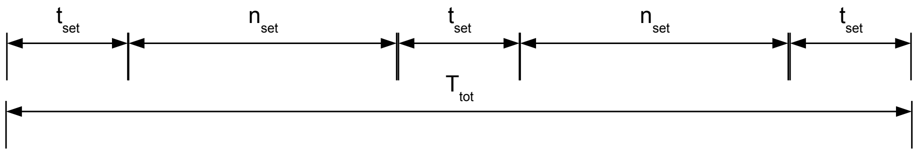

Figure 13: Visualization of telemetry history playback definition.

The following parameters can be identified to control the telemetry history playback:

- ETC start: ETC time the telemetry history playback shall start.
- n_{set}: Amount of time in a telemetry set.
- t_{set}: Interval between telemetry sets expressed in amount of time.
- T_{tot}: Total telemetry timespan, note that this is not the actual amount of time for which telemetry will be downlinked.

The measurement interval and measurement set size should be optimized, while not exceeding the expected amount of AX.25 frames which can be successfully decoded. Of course the expected amount of successfully decoded AX.25 frames is a theoretical number and therefore it is not guaranteed all intended data will be received. In this section the house and science data are assumed to be strictly separated in AX.25 frames. However, it is possible to mix the science and housekeeping data in AX.25 frames.

4.3.2 *Manual Definition Control*

Since the DelFFi nanosatellites will in principle not be aware if/when they are in the visibility of a ground station, manual operation of the telemetry history definition is required for optimal data coverage. Because there is no active acknowledgement of received data by ground stations the DelFFi nanosatellites will not know what telemetry history to send. Or in other words: There is no intelligent telemetry control onboard the DelFFi nanosatellites and thus if a second ground station pass takes place, satellite operators want to define the telemetry history again so no duplicate data is received. Note that this new telemetry history definition telecommand will depend on the last frames received during the first ground station pass.

Table 8: Example of a fictitious telecommand to control the telemetry history playback.

Parameter	Example content
ETC start	6417791
n_{set}	90 s
t_{set}	300 s
T_{tot}	36000 s

It is recommended the control of this telemetry history playback is merged with the flexible telemetry method as outlined in section 4.2. By means of a telecommand AX.25 frame IDs can be designated as either real-time or telemetry history data. In principle the actual content/included parameters of the telemetry history AX.25 frame can be the same as a real-time AX.25 frame. Only the data itself included in the frame is connected to a different time stamp. In table 8 an example of a telemetry history playback telecommand definition is given. In the previous subsection 4.3.1 the description of the parameters which are outlined in table 8 can be found. The actual implementation of these parameters in the FlexTLM assembly software can be found in section 4.7.

4.3.3 *Automatic Definition Control*

It would be beneficial if the ground segment could automatically generate telecommands for the telemetry history playback based on gaps in the telemetry database. This would reduce the mission operations resources of DelFFi significantly. Note that this is a recommendation to the DelFFi ground segment and provides one example of how to approach automatic telecommand history playback generation. The following procedure to implement on the DelFFi server is envisioned:

1. Cron jobs on the server can be scheduled to determine the big gaps of telemetry in the DelFFi database, between the last set of telemetry frames and the previous set of telemetry frames.

2. The access time of the next ground station pass can be calculated by software running on the server. With the access time known and the currently set FlexTLM definition on the DelFFi satellites, it can be estimated how much telemetry can be downlinked during the next pass.

3. Based on this automatic analysis the telemetry history playback ETC start time can be defined. Also the amount of time a telemetry set encompasses (n_{set}) and the interval between telemetry sets (t_{set}) can be determined following predefined rules. For ex-

ample, the right balance between these two parameters shall be found by optimizing the Earth coverage and local data coverage.

4. The radio operators will automatically receive a proposed telemetry history playback telecommand a few hours before the pass. Preferably these telecommands are automatically put in a queue of the new transmitter uplink program, by means of a link with the server.

It would be even better if these telecommands and Two-line Element (TLE) updates can be transmitted automatically as well. However, this feature requires further investigation, because it needs to be assured no humans are present at the roof of the Delfi GS. In addition, an exemption needs to be requested for automatic transmission without a radio operator present in the Delfi GS. Furthermore, it needs to be verified up to how much power can be used for transmission without posing any danger for humans.

4.3.4 *Transceiver Duty Cycling & Control*

The DelFFi mission has significant power requirements and the transceiver might have to be duty cycled during certain phases of the mission. This has serious implications with regard to the ground station operations and the amount of data which can be downlinked every day. If a duty cycle for the transceiver is used, this will lead to no full utilization of the downlink of the DelFFi nanosatellites and this is far from desirable. Duty cycling the transceiver will most likely also result in a conflict with the minimum downlink requirement of the FIPEX instrument of 0.3 Mbits per day (FPX-SW-0360 / 3.3.3) [24] and the whole orbit data requirement of QB50 (QB50-SYS-1.4.1 / 3.6.1) [23]. Also, requirement SAT.2.6-F-10 (3.4.6) imposes this functionality on the DelFFi satellites because the downlink shall be enabled by means of a telecommand or a pre-defined schedule. Hence, a smart design of the downlink needs to be created to avoid these problems and two solutions are envisioned:

- **Telecommand:** Every time a DelFFi nanosatellite comes in view of the Delfi ground station a telecommand has to be transmitted, which puts the nanosatellite in a high speed telemetry mode for a certain time duration. It would be beneficial if partner universities or radio amateurs who will support the DelFFi mission with their ground station, will be provided with the capability to transmit a high speed telemetry mode telecommand request as well (e.g. a dedicated DelFFi telecommand uplink program). One serious implication of this solution is that ground station operations will be seriously intensified. Even if the ground station is fully automated the uplink capability of the Delfi ground station may only be used if an operator is present, because the roof with the antennas has to be checked and cleared for any persons before transmitting. If it is considered there will be two DelFFi satellites which will be separated significantly during the mission - hence different ground station pass times - operators need to be present many times per day.

- **Scheduling:** This solution will require telecommands which include schedule information about the transmitter on times for a few days in advance. The transmitter on times are correlated with the pass times of the participating ground stations. The main benefit of this approach is that no continuous presence of ground station operators is required. However, a downside is that a slightly more complex system to implement the schedule of the transmitter on times is necessary. Also ground stations which incidentally help out cannot receive the DelFFi satellites if they are not in the vicinity of the scheduled ground stations.

Another consideration has to be made about whether the transmitter will be (partially) on during eclipse or not. The Delfi-n3Xt transmitter in nominal mode was off during eclipse to save power. If the transmitter on pass times are scheduled DelFFi could maybe even transmit during eclipse. Nonetheless, this has to be determined at a later stage in the project when a more detailed power budget is available. Operations wise it is strongly recommended implementing transmitter scheduling for the DelFFi mission (see requirement 4). In addition, the transmitter needs to be in a low speed telemetry mode outside the scheduled transmitter on times. In this way it can always be identified if the satellite is still alive and some basic telemetry can be retrieved.

4.4 DELFFI SOFTWARE ICD FLEXTLM IMPLEMENTATION

Implementing FlexTLM requires careful analysis of the DelFFi software ICD [34]. In table 9 a summary of the DelFFi software ICD subsystems can be found. In this table the average total bit size for different telemetry profiles can be seen. This average total bit size is based on the desired update frequency for each telemetry profile and parameter. Note that individual packets containing groups of parameters for each subsystem can be combined in different AX.25 frames to achieve the desired telemetry profile. Therefore, table 9 does not reflect the actual average bit size for a subsystem in all cases. According to the DelFFi ADCS engineers, it will be hard to speed up the ADCS loop from 0.5 to 1 Hz and higher update frequencies than 1 Hz are even out of the question.

Table 9: DelFFi software ICD subsystem summary for different telemetry profiles.

Subsystem	**Average total bit size [$\frac{bit}{s}$]**		
	Minimum	**Normal**	**Maximum**
Attitude Determination & Control System (ADCS)	1121	1121	2242
Onboard Computer (OBC)	188	188	188
Primary Transceiver (PTRX)	96	96	96
Electrical Power System (EPS)	43	432	432
Thermal Control System (TCS)	2	3	396
Science Unit (FIPEX)	376	376	377
Inter Satellite Link (GAMALINK)	128	128	128
Propulsion (mPs+)	9	84	2316
Deployment Antenna Board (DAB)	0	1	2
Bluetooth LE Experiment (BLE)	5	5	5
Total	**1963**	**2428**	**6175**

The telemetry definition is strongly correlated with the specific operating mode of a DelFFi satellite. Especially the ADCS modes will have a major impact on the telemetry definition for each operational mode. In table 10 the possible ADCS modes and other subsystems which are active during each DelFFi mission phase can be seen. Note that not all subsystems are shown in this table because the generic subsystems such as the OBC, EPS and PTRX will be on during all phases of the DelFFi mission.

In order to avoid data gaps, it shall be attempted for each new telemetry definition that the packet groups which require the data acquisition loop update frequency (e.g. 1 Hz), are the core of every AX.25 frame defined in the frame sequence table. The remainder of the AX.25 frame should be filled with packet groups which require a lower update frequency. Of course, this is not possible if the total bit size of the packet groups which require the data acquisition loop update frequency is larger than the content of an AX.25 frame. In that case, it has to be accepted every 10 seconds there is a gap in which the low update frequency packet groups are assembled in an AX.25 frame.

Table 10: Possible ADCS modes and other active subsystems during each DelFFi mission phase. [16] [17]

	Mission phase				
	LEOP	**Natural drift**	**Formation acquisition**	**Formation flying**	**Scientific mode**
	ADCS mode				
De-tumbling	x				
Sun pointing					x
Thrust vector control			x	x	x
Safe	x	x			x
	Subsystem				
FIPEX			x	x	x
GAMALINK			x	x	
mPs+			x	x	
DAB	x				

4.4.1 *Software ICD Design Approach*

The default length of an AX.25 information field is 256 octets [15]. Hence, the definition of each AX.25 frame in the DelFFi software ICD shall not exceed 2048 bits. However, 100 bits are reserved because also an initialization vector defining the telemetry packets to be included in the AX.25 frames needs to be incorporated. Hence, 1948 bits of actual content are available for individual AX.25 frames. But note that probably four AX.25 frames can be send in a transmission event of 1 Hz and hence the packets of the largest data producers, such as the ADCS and mPS, can be distributed over several AX.25 frames, if necessary.

In order to design the software ICD the following design approach has been used:

1. All parameters and their characteristics were gathered of individual subsystems and discussed with the responsible subsystem engineers in detail. In addition, available ICDs were carefully read through to identify telemetry parameters as well.

2. Lists have been made in spreadsheets listing all parameters for a subsystem with the corresponding bit size, description and desired update frequencies.

3. Packet groups have been identified which are often related to I^2C read out requests as specified in subsystem ICDs.

4. FlexTLM packets were defined based on groups of likewise parameters and characteristics such as desired update frequencies. FlexTLM packets can contain the same group of parameters as packet groups, but this is not mandatory.

5. The subtotals of each FlexTLM packet were summarized in a sheet and a matrix was developed relating the possible ADCS modes to the relevant FlexTLM packets. This correlation was merely used as an indication for the AX.25 frames which were defined in the same sheet.

6. The definition of the AX.25 frames is based on an iteration which takes into account the FlexTLM packets, which need to be transmitted for each (ADCS) operating mode, and the maximum bit size of a frame. The FlexTLM packets are spread over several AX.25 frame definitions, if not all FlexTLM packets fit into one AX.25 frame, or in case the desired update frequencies vary a lot.

7. In the final sheet, the frame sequence tables specify the order of the AX.25 frames to be transmitted for the different ADCS modes. Several AX.25 frames can be transmitted in one second, but the sum of the bit size of the AX.25 frames shall not exceed

the data rate. Hence, if the transmission events are defined as 1 Hz the total bit size cannot be higher than 9600 bit. However, because there is some protocol overhead (160 bits) [34] and assuming four AX.25 frames usually fit in a 1 Hz transmission event, the actual content cannot be larger than 7616 bits (1904 bits maximum content x four AX.25 frames).

During the design of the FlexTLM of the DelFFi software ICD, it was discovered the required core data for each mode was significantly smaller than was allocated. Hence, more AX.25 frames could be included which contained telemetry history data such as WOD and FIPEX science data.

4.4.2 *Data Volume Savings of FlexTLM*

It is important to critically reflect on the amount of data volume, which can be saved by implementing the FlexTLM solution, instead of fixed telemetry. This will support a balanced trade and decision about the need for a FlexTLM implementation. Note that ´save´ in this context means more relevant telemetry can be downlinked and not the actual size of the data itself is reduced. It is not obvious how much data volume is saved, because fixed telemetry would already require different AX.25 frames. Much of the data bandwidth is saved by not including certain parameters in the telemetry frames every second. In table 11 an overview is provided for the data volume, which can be saved for different data packets on a daily basis, if FlexTLM is used. Note that only a few examples of data packet savings are shown in this table, but the actual total data savings are even larger than the shown total. The ADCS data volume savings are based on the difference between packets included in the detumbling mode and Thrust Vector Control (TVC) mode. The total data volume savings of these three data packets - TCS, WOD and ADCS - alone are already 64 % of the entire estimated daily received data volume (see subsection 4.1).

Table 11: Overview of daily data volume savings for different telemetry packets as a result of the FlexTLM implementation for different orbital altitudes.

Altitude [km]	**TCS [kbit]**	**WOD [kbit]**	**ADCS [kbit]**	**Total [kbit]**
200	234	709	466	**1409**
250	319	965	634	**1917**
300	417	1260	828	**2505**
350	511	1546	1016	**3073**
400	615	1858	1221	**3694**

4.5 FLEXTLM GROUND SEGMENT IMPLICATIONS

The telemetry server has to process incoming data. For Delfi-n3Xt the decoding of the AX.25 frames was performed directly by the processing software. However, the Delfi-n3Xt telemetry only included two types of AX.25 frames. The FlexTLM approach allows many more AX.25 frame definitions and therefore the dissection of data needs to be more flexible. If open-source software such as Wireshark will be used for the dissection of packets, special attention needs to be given to this modularity [37]. In principle Wireshark supports custom dissectors, but the header of an AX.25 frame needs to define the protocol. Therefore, the AX.25 frame IDs need to be put in the PID field as shown in table 3. This is an 8 bit field so 2^8=256 unique AX.25 frame IDs are possible.

4.6 TELEMETRY HISTORY FEASIBILITY ANALYSIS

Now the FlexTLM system has been outlined and the DelFFi software ICD is designed, it is time to check if all telemetry history can be downlinked. Particularly, it will be verified if the 300 kbits of SU data per day downlink requirement (FPX-SW-0360 / 3.3.3) can be satisfied. In addition, the requirement to receive a log of WOD of every minute of the mission (QB50-SYS-1.4.1 / 3.6.1) needs to be verified. These two requirements need to be achieved by still maintaining a core real-time telemetry profile as well. For the telemetry analysis the recommended operations of the FIPEX as specified before in subsection 4.1.1 are used. In table 12 an overview of the required AX.25 frame inclusion frequency for the WOD and FIPEX data can be found. The calculations are based on the following assumptions:

- One AX.25 WOD frame contains **32 packets** representing 32 minutes of science data. (See subsection 4.1.2)
- One AX.25 FIPEX frame contains **20 FIPEX measurement packets** and **1 STM measurement packet,** which is based on a 120 seconds measurement time span. (See subsection 4.1.1 for the ratio FIPEX - STM)

Table 12: Overview of AX.25 frame inclusion frequency for the WOD and FIPEX science data based on different orbital altitudes based on a transmitter data rate of 9.6 kbps.

Altitude [km]	Average access time/day [s]	WOD [frames/s]	FIPEX [frames/sec]
200	411	0.11	1.75
250	560	0.08	1.29
300	732	0.06	0.98
350	898	0.05	0.80
400	1079	0.04	0.67

It can be seen in table 12 that a WOD AX.25 frame only needs to be included once every twenty seconds at the beginning of the DelFFi mission. At the end of the mission this increases to once every ten seconds. Note that this is based on the assumption the telemetry history AX.25 frame will be transmitted only once. It might be more desirable to transmit it two or three times to increase the probability of receiving the data successfully.

A FIPEX AX.25 frame containing 1596 bits of actual content [34] will need to be transmitted nearly once every second at the beginning of the DelFFi mission. During the end of the mission this FIPEX AX.25 frame transmission increases to two twice every second. The numbers observed in this analysis clearly indicate the advantage of being able to adapt the telemetry during the DelFFi mission and hence the benefits of implementing FlexTLM.

Some concerns were raised if the SD card search and read out time for telemetry history data is fast enough to fit in the data acquisition loop. Since the transmitter data rate will not exceed 9600 bps (see subsection 4.1) or 7616 bps of real data (1904 bits maximum content x four AX.25 frames), no more than 7616 bits of telemetry data needs to be acquired from the SD card every second. Note that some overhead is also present on top of this content. An analysis and design of the anticipated fault-tolerant storage system for DelFFi indicates for a block of 512 bytes 13 ms ± 2 ms of search and read out time for the SD card is required [27]. Four AX.25 frames fit in a 1 Hz transmission event if they are completely filled with data. This means no more than four blocks of telemetry history data need to be read of the SD card. If one considers the maximum read out time of 15 ms then 60 ms is the maximum amount of time which is required by the data acquisition loop for the SD card readout time. Note that more than four blocks of telemetry history data need to be read of the SD card if one considers the data destined for one AX.25 frame is spread over several blocks.

4.7 flextlm assembly software

In figure 14 the activity flow for the assembly of FlexTLM frames is shown. Some parameters used in this flowchart were defined earlier in subsection 4.3.1 to control the telemetry history playback:

- ETC start: ETC time the telemetry history playback shall start.
- n_{set}: Amount of time in a telemetry set.
- t_{set}: Interval between telemetry sets expressed in amount of time.
- T_{tot}: Total telemetry time span, note that this is not the actual amount of time for which telemetry will be downlinked.

A few blocks require a bit of additional explanation for a better understanding of the flow chart:

- **Assembly.FlexTLM.06:** Include a sequence of bits indicating which packets are included in the current frame.
- **Assembly.FlexTLM.12:** In this code block it will be determined which ETC time needs to be selected to acquire the corresponding data block of the SD card.
- **Assembly.FlexTLM.14:** Increase the telemetry history step with one second for the next ETC time data block.
- **Assembly.FlexTLM.15:** Check if the FlexTLM assembly has completed the assembly of frames of the current telemetry set.
- **Assembly.FlexTLM.16:** If the total measurement time span (T_{tot}) has been reached the telemetry history step will be set to zero and the FlexTLM history cycle will be repeated. Unless a new telecommand resets the telemetry history definition.
- **Assembly.FlexTLM.18:** Set the new limit ($n_{set,new}$) for the time when the assembly of the frames has completed another telemetry set (n_{set}).

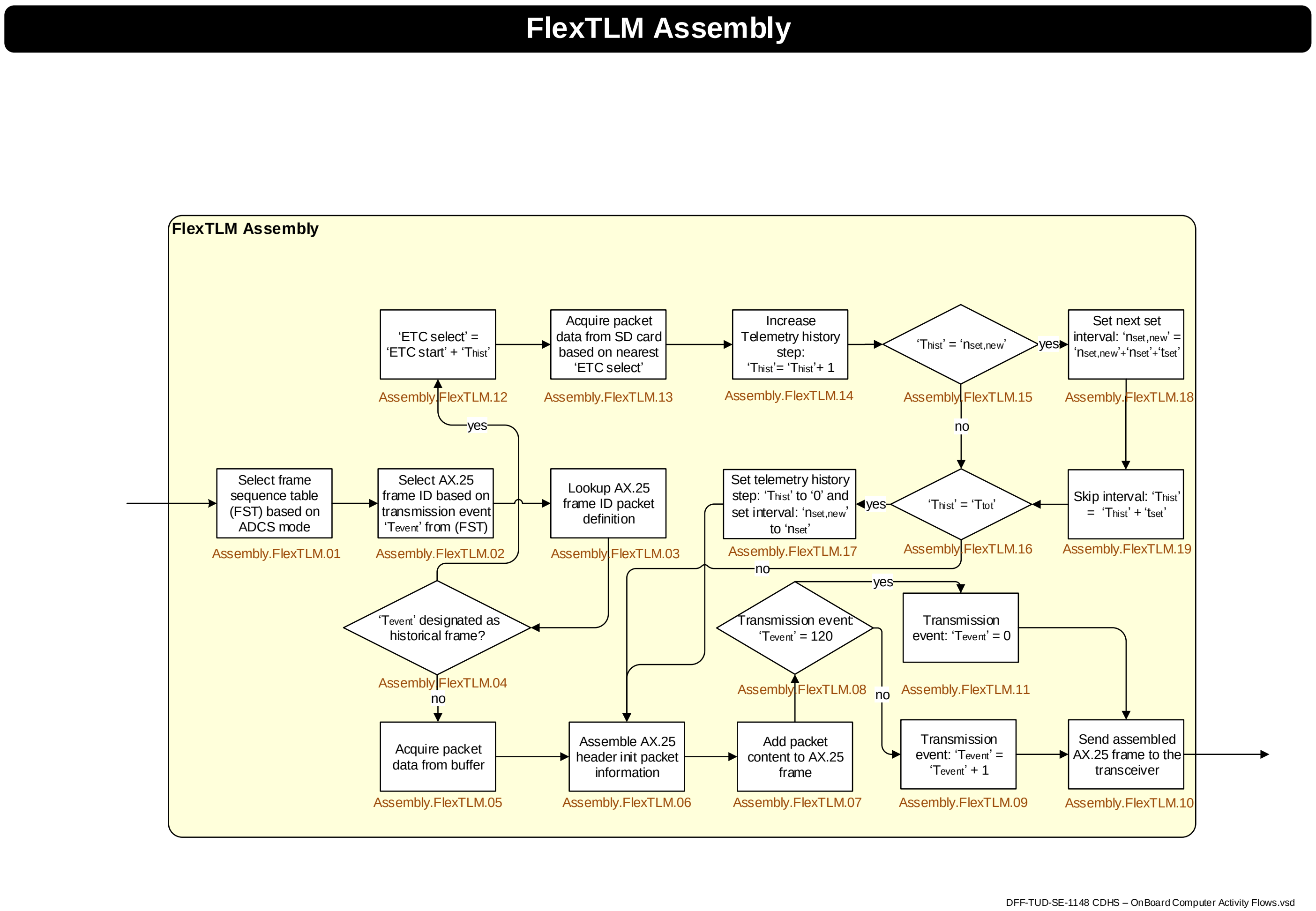

Figure 14: Designed activity flow of the FlexTLM assembly software.

4.8 Conclusion & Requirements

In this chapter a flexible telemetry design and telemetry history playback capability has been proposed. It is shown by means of various examples and calculations throughout this chapter that there is a need for a FlexTLM system on DelFFi. Future DelFFi software engineers might encounter problems implementing specific FlexTLM design choices made in this chapter, because of hardware limitations or other restrictions. The software engineer(s) shall have the freedom to adapt the FlexTLM system to a practical implementation in close cooperation with the CDHS engineer.

Several requirements with respect to the flexible telemetry system based on the analysis in this chapter are proposed:

1. The content of AX.25 frames shall be defined by packet group identifiers. (see subsection 4.2.3)
2. The CDHS shall semi-autonomously control the content of AX.25 frames. (see subsection 4.2.6)
    a) The content of AX.25 frames shall be based on the operating mode of the satellite.
    b) The content of AX.25 frames can be overridden by means of a telecommand.
3. The sequence of transmitting AX.25 frames shall be controlled by means of a telecommand frame sequence table. (see subsection 4.2.5)
4. The transmitter on status shall be scheduled by means of a telecommand. (see subsection 4.3.4)
5. The telemetry history shall be an optional feature and not be enabled by default. (see subsection 4.2.2)

Part II

WIRELESS INTRA-SPACECRAFT COMMUNICATION FOR NANOSATELLITES

5 NANOSATELLITE BLUETOOTH LOW ENERGY EXPERIMENTS

Note that most of the contents of this chapter are submitted as part of a paper to the 4S symposium 2014 [38].

Usually wires are used for intra-spacecraft communication in nanosatellites, but to reduce wiring integration complexity and increase reliability, it would be interesting to consider wireless communication for intra-spacecraft data links [39]. Several wireless nanosatellite architectures are possible, ranging from completely independent subsystems to just a few wireless sensor nodes.

One of the most obvious advantages for wireless intra-spacecraft communication is the prevention of failing wires or connectors. However, maybe an even more important advantage is that the risk of a single shared interface is reduced. For example, a problem with the I^2C interface can freeze the entire data bus. Reduced mass can also be a benefit of wireless communication, however this is probably only the case in larger satellites which use long wires. In smaller satellites such as microsatellites and nanosatellites the mass savings are marginal.

Another benefit of wireless nodes is easier Assembly, Integration & Testing (AIT) [40]. For example, it is possible to monitor the links with external nodes outside the nanosatellite and thus no physical connection is required to debug certain wireless nodes. Also, there is more flexibility in the placement of nodes in the nanosatellite and complexity of the wiring is reduced. Mainly because the wires connecting the nodes which are mounted on the body (e.g. sun sensors and temperature sensors), should not have too much slack and therefore the connectors are fixed through the integration holes after integration of the body panels.

Of course, there are not only benefits for using wireless intra-spacecraft communication, but also some potential obstacles. One of the main disadvantages of implementing wireless intra-spacecraft communication is Electromagnetic Interference (EMI) [40]. Mainly the Printed Circuit Boards (PCB) and nanosatellite structure/walls can induce Radio Frequency (RF) problems because they can act as shields or reflectors causing signal degradation due to losses and multipath effects.

It is considered to implement wireless sensor nodes on the DelFFi nanosatellites as a technology demonstration. Body temperature sensors are not mission critical and are difficult to integrate with wires, so they are chosen for an in-orbit experiment. In order to test the performance of wireless internal data communication for nanosatellites on-ground experiments were conducted in a representative setup.

In section 5.1 of this chapter some past wireless intra-spacecraft communication experiments are discussed. A short analysis about the hardware and protocol selection of the wireless link is given in section 5.2. Subsequently, in section 5.3 an overview of the experimental setup used for the wireless communication tests is given.

5.1 PAST WIRELESS INTRA-SPACECRAFT COMMUNICATION EXPERIMENTS

In the past several experiments with wireless intra-spacecraft communication were conducted. Some of these experiments were conducted in orbit and other experiments evaluated on ground. In this section some of the main findings of these experiments are outlined to obtain a good basic insight about satellite internal wireless communication in order to define the experiments.

5.1.1 *In-orbit*

In the Delfi-C3 nanosatellite a wireless intra-spacecraft communication link has already been tested. The Autonomous Wireless Sun Sensor (AWSS) is a quadrant detector which is powered by a triple junction solar cell [41]. In figure 15 the front and back side of the AWSS can be seen. Generated data is relayed by a Nordic semiconductor nRF9E5 System on a Chip (SoC). This chipset is a low power multi-band sub 1 GHz RF SoC. The main goal of implementing the AWSS in Delfi-C3 was to demonstrate the feasibility of a wireless link and that the sensor could operate under a variable power supply. Two AWSS systems were integrated in Delfi-C3 and one functioned properly. It is not certain what was the cause of the malfunctioning AWSS but there is a strong indication the sun sensor itself was not working properly [41].

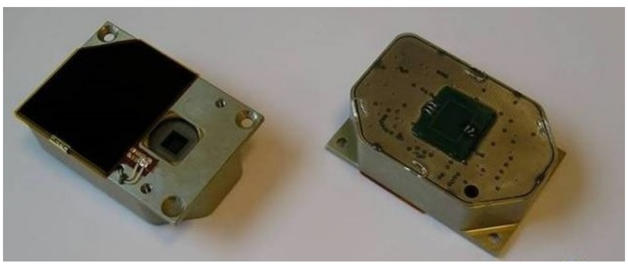

Figure 15: Delfi-C3 Autonomous Wireless Sun Sensor (AWSS). [7]

There have also been tests with non RF wireless intra-spacecraft communication experiments in satellites. Several Optical Wireless Links to Intra-Spacecraft (OWSL) communication experiments have been conducted in space such as the Nanosat-01 and the current ongoing OPTOS (All-Optical Satellite) mission [42]. In figure 16 an image of OPTOS can be seen. There was quite some positive experience with OWSL for Nanosat-01 and especially the measured Bit Error Rate (BER) was favorable. The BER for data rates up to 200 kbps did not exceed 10^{-8} [-] for the major part of the mission [42]. However, at the South Atlantic Anomaly (SAA) the BER was in the order of 10^{-6} [-] due to Single Event Transients (SETs).

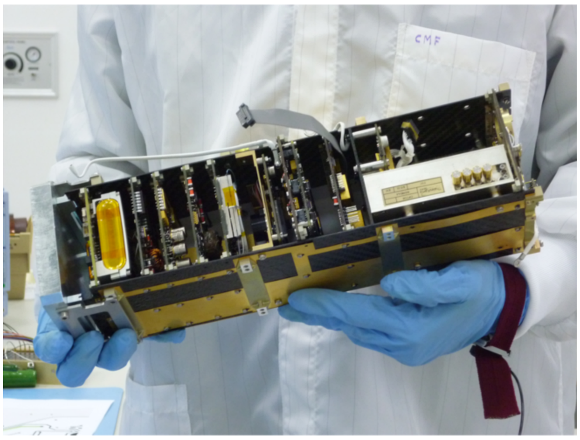

Figure 16: Picture of the all-optical satellite OPTOS. [8]

A wireless Controller Area Network (CAN) bus is implemented on the OPTOS satellite. No data wires are present and all units are connected to a wireless CAN network, which operates on 950 nm at 125 kbps [42]. OPTOS was launched in November 2013 [43] and no detailed flight results are publicly available yet. The satellite was ending its commissioning phase as of March 19th, 2014 and operating in nominal mode [44].

5.1.2 *On-ground*

An internal wireless bus for a CubeSat has been implemented in the Norwegian University of Technology Test Satellite (NUTS). The experimental wireless bus of this satellite implements nRF24L01 ultra low power transceivers which use the Enhanced Shockburst™ protocol [45]. Each packet supports up to 32 bytes of payload data and the maximum raw data rate of these modules is 2 Mbps. At the time of publication of this paper the satellite still has to be launched and hence in-orbit validation has yet to be performed.

Another test of wireless intra-spacecraft communication was conducted by use of the Zigbee PRO protocol and ZigBit hardware [46]. Some problems were encountered during these tests. For example, when several sensor nodes at fast duty cycles were used in the network sensor data arrival delays of 30 ms occurred frequently. This is not necessarily a problem unless a subsystem requires a high sample frequency and accuracy. It was also observed that end devices stayed awake for a relatively long duration up till 120 ms which leads to a higher power consumption.

5.2 HARDWARE & WIRELESS PROTOCOL SELECTION

There are several state-of-the-art wireless communication protocols which had been analyzed. An extensive trade-off has been conducted in which ZigBee, ANT, the Infrared Data Association (IrDA), Low Power Wi-Fi and Bluetooth Low Energy (BLE) were compared. It turned out that BLE was the protocol which performs best based on the criteria power consumption and achievable data rates [47]. Simulations indicate the additional power consumption of BLE compared to wired solutions seems to be moderate if efficient use is made of the built-in sleep mode capabilities of the BLE modules.

The main reason why IrDA was not selected is because it requires a clear line of sight. For some applications in a nanosatellite this might not be a problem, but it would limit the freedom of other implementations as well. Otherwise IrDA transceivers perform really well in terms of power consumption and support of high data rates. Low power Wi-Fi also has favorable characteristics for higher data rates but it is overkill for lower power consuming nodes such as sensors. Only ZigBee is most similar to BLE in terms of data rates and power consumption. However, the power consumed per transmitted bit is slightly worse than BLE. Therefore, it was decided to perform experiments with BLE modules to determine the suitability to implement wireless data links for sensors or even full subsystems in a nanosatellite.

5.2.1 *Hardware Selection*

The BLE module used in the experiments was the BLE113 of Bluegiga Technologies. It is one of the latest BLE module designs available on the market and is very power efficient and small. It integrates a Bluetooth radio and software stack with easy access to the Application Programming Interface (API) by the so called BGScript. This BGScript is a custom scripting language used for applications running on the module which takes advantage of the onboard microcontroller. In the transmitting mode (TX) the module consumes 18.2 mA of current and in receiving mode (RX) 14.3 mA is consumed [11] with an operating voltage of about 3 V. Three sleep modes/low power modes are used on the BLE113 and these take place based on activity, soft timers and interrupts [48]. In sleep mode the current consumption ranges from 0.5 to 270 μA depending on the specific power mode.

The BLE protocol has an RF symbol modulation rate of 1 Mbps but this includes layers of protocol overhead. The maximum theoretical effective throughput of the BLE protocol is 270 kbps. However, this throughput cannot be achieved in practical situations because of hardware limitations and the RF environment [49]. The best throughput between two BLE modules Bluegiga has observed is 100 kbps [50]. Due to the reliability that is required this data rate will be lower because packets have to be acknowledged. In comparison with the raw data rate of 100 kbps for standard mode I^2C and approximately 50 kbps throughput these modules perform rather well. Nevertheless, BLE cannot be used to interface all current CubeSat systems and components available and hence BLE should be seen as a complementary technology to either wired solutions or other wireless protocols for higher data rates which can be used in certain situations. If one wants to achieve higher data rates other wireless protocols can better be considered. For example, in the case of a wireless high end payload normal Bluetooth or even Wi-Fi is more suitable.

5.3 EXPERIMENTAL SETUP

To determine the suitability of implementing a wireless link as an experiment in the DelFFi nanosatellites the BLE wireless communication experiment for nanosatellites had to be conducted. In this section the test setup used for the experiments is explained. The main focus of the experiments presented in this paper is on the performance of the wireless link within a CubeSat. A stack of unpopulated PCBs with grounded layers and a battery pack in a nanosatellite structure was used as a reference test model. In this test model several parameters were controlled such as the distance, location and the amount of PCBs. For the EMI tests a spare model of the Delfi-C3 nanosatellite was used to conduct the experiments and monitor possible actively generated interference on the wireless link caused by the electronics of other systems.

5.3.1 *Objectives*

The main objectives of the nanosatellite wireless data link experiment are the following:

- Determine the connection stability of wireless modules in a nanosatellite.
- Measure the Packet Error Rate (PER) for various conditions: Distance, location, amount of PCBs and EMI effects.
- Measure the power consumption characteristics and corresponding achievable data rates of the wireless modules.
- Experience the complexity and extra development effort required to implement a wireless link solution in a nanosatellite.

5.3.2 *Arduino Due Board*

Initial testing was done by connecting the BLE113 modules to Arduino Dues through the UART interface. In figure 17 the Arduino Due board which was used for the BLE experiments can be seen. The Arduino Due board version has been selected because it is more powerful and the PCB is better populated than the Arduino Pro version. Power is supplied to the Arduino board through a USB interface. This same USB Interface is used for serial communication with a computer. The supply voltage required by the BLE113 modules is between 2 V to 3.6 V. Therefore the 3.3 V Arduino Due version was selected instead of the standard 5 V supply boards or the in-house T-minus Arduino board.

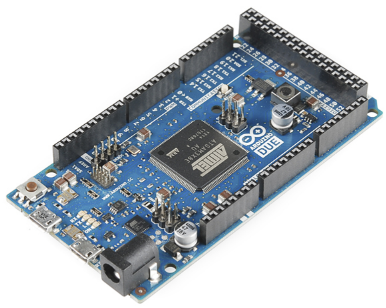

Figure 17: Arduino Due board. [9]

5.3.3 *Measuring Characteristics*

Several characteristics about the performance of the BLE113 modules needed to be measured to fulfill the goals of the experiments as outlined in section 5.3.1. One of these characteristics is the BER of the BLE module connections. Information about the BER of a BLE113 module cannot be retrieved directly. However, information about received and sent packets can be retrieved from the modules. For instance, the amount of packets in which the Cyclic Redundancy Check (CRC) has failed and succeeded. In addition, the

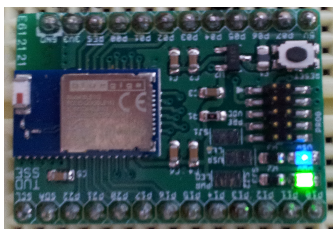

Figure 18: BLE113 module breakout board.

amount of packets retransmitted and acknowledgements received can be requested from the modules. Based on the foregoing mentioned parameters the Packet Error Rate (PER) and Packet Sent Error Rate (PSER) was calculated. The PER and PSER can be distinguished by the fact that the PSER is the PER plus the ratio of packets which were never received at the collector side and have to be sent again. It is important to realize the PER does not represent permanently lost data because packets with failed CRCs are simply transmitted by the server again as well.

To determine the PER with a certain confidence level (CL) some simple calculations had to be made to determine the minimum measurement duration time. The CL is defined as the percentage of cases in which the BER is found to be lower than the specified BER. The BER can be calculated by using equation 4 [51].

$$CL = 1 - e^{-N_{bits}BER} \tag{4}$$

To determine the PER equation 4 can be rewritten to equation 5.

$$CL = 1 - e^{-N_{packets}PER} \tag{5}$$

This equation can be modified to calculate the required measurement duration which can be calculated with equation 6.

$$t = \frac{-\ln(1 - CL)}{f_p \cdot PER} \tag{6}$$

Assuming a CL of less than 0.95 [-], a packet frequency (f_p) of 4 Hz and a maximum PER of 0.1 [-] then a measurement has to last for 8 seconds. The actual measurements were conducted for 200 seconds to determine the PER with a higher accuracy. A PER of 0.1 [-] is deemed to be acceptable as long as it is randomly distributed and does not occur too many times for one single packet transmission attempt. The packet frequency is set at 4 Hz because that was found to be the maximum packet frequency possible during the measurements due to software limitations of the UART interface of the Arduinos. However, if higher update frequencies are required, such as in the case of an ADCS, faster packet frequencies should be tested as well. Especially because the use of smaller connection intervals might affect the overall performance of the BLE modules.

Another property which was logged during the tests in the modules was the Received Signal Strength Indication (RSSI). In addition, it was also desired that the power consumption of the BLE113 modules is known for different TX power settings and power profiles. A data acquisition (DAQ) system with a sample frequency of 4 kS/s was used to measure the voltage over a resistor put in series with the power supply. Hence, by measuring the voltage over this resistor the current could be derived and the average power consumption was calculated. Some bias was present in the power consumption measurements because of the breakout boards overhead (e.g. status indicator LEDs) and therefore the 8.2 mW base power consumption of the LEDs was subtracted from the measurements. These breakout boards (see figure 18) were manufactured for ease of access to the modules and are not supposed to be used in an end application.

5.3.4 *Communication Architecture*

It is convenient to have a basic idea of the different data transfer modes the BLE modules can use to understand the data exchange procedure. Basically four operating modes are available on the BLE113 modules [52]:

- **Write:** Write operations are requested by the client (master) and the client writes data to the server (slave). A master can be a server, but usually it is a client. If the OBC BLE client module wants to send certain data to a subsystem or sensor node it can act as a server as well.
- **Read:** The client requests to read data of the server.
- **Notify:** Notifications are performed by the server by pushing data to the client. The client does not acknowledge data.
- **Indicate:** Indications are again performed by pushing data of the server to the client. However, indications are confirmed by the client to the server contrary to notifications. Hence, indications are more reliable but notifications are faster because no acknowledgement needs to be sent by the client and received by the server.

These operating modes make several data exchange architectures possible to implement for internal nanosatellite wireless communication. Which architecture is the most suitable implementation depends on the specific situation. But generally it can be said that for a multi-node network the best solution in terms of efficiency and reliability is that the client performs read operations on each server sequentially. This data exchange flow limits the power consumption on the server while the client has a higher duty cycle. Another possibility is that the slaves notify or indicate data to the Generic Attribute Profile (GATT) server of the client. The GATT profile defines all parameters and its characteristics which can be exchanged over the BLE link. However, in this case timing is really sensitive because attempts from multiple slaves to write data to the client at the same time can lead to collisions and therefore many packet transmit retries. Active and precise time synchronization between the modules could reduce these packet collisions. If a network with only a few nodes is developed one can best use indications because this is the most reliable method of data transfer and can achieve sufficient data rates for small sensor nodes.

During the measurements only indications were used to perform the experiments. No real sensor data was used because the main goal of the experiments was to test the connection performance of the wireless links. All tests were conducted with attribute data with the maximum possible payload size of 20 bytes if indications are enabled and a single attribute represents several sensor measurements. Note that encryption and authentication were not enabled during the experiments. These functionalities of the BLE modules impose overhead and are probably not likely to be needed in the eventual application. Security issues can be handled by using a white list for all trusted BLE module Media Access Control (MAC) addresses.

5.3.5 *Module Settings*

Several BLE113 module settings are available to optimize power and performance, according to the user needs. The following parameters can be tuned to achieve a certain performance [53]:

- **Power:** Only a single packet is sent during each connection interval. This setting minimizes power consumption, but might limit throughput.
- **Balanced:** Sends only packets that fit in transmission buffer which is 128B. Normally 3-4 packets will fit, depending on user payload and overhead.
- **Performance:** Maximize throughput and loads new packets to the transmission buffer and sends them as soon as previous packets have been successfully transmitted. Increases power consumption.

The TX power of the modules can also be set in the configuration files of the modules. The highest power setting is equal to approximately +0 dBm. The lowest power setting is -24 dBm [11]. Different parameter setting combinations are possible and were explored. The corresponding power consumption characteristics, data rates achieved, PER, PSER, RSSI and observations made were logged. This information and the analysis hereof is presented in section 6.1.

5.3.6 *Measurements Test Setup & Cases*

For development purposes and easy access the BLE modules were mounted on breakout boards during the tests. These breakout boards were powered by the 3.3 V outputs of the Arduino Dues and basic parameters of the BLE modules were monitored through the UART interface. In the nanosatellite these breakout boards were placed at various locations to measure the performance under EMI influence of PCBs and the aluminum walls of the nanosatellite. The PCBs were connected to the nanosatellite structure and connected to the ground of one Arduino. Measurements were conducted for the following cases:

1. **Nanosatellite model unpowered - no obstacles:** The BLE modules were placed in-plane with the unpopulated PCBs in the test model but the satellite was turned off. During these communication experiments the BLE modules had an almost clear line of sight. This case was used as a reference for the other cases to compare the performance of the modules in different conditions.
2. **Nanosatellite model unpowered - obstacles:** The purpose of this case was to measure the influence of PCBs as shields and potential multipath effects on the wireless link. See figure 19 for an impression of this test model.
3. **Nanosatellite powered:** Again the BLE modules were placed at various locations, but now in the Delfi-C3 satellite as a representative model and in this case the satellite was turned on. The goal of this case was to determine potential EMI effects of electronic circuits.
4. **Different module orientations:** Different orientations of the BLE modules were tried in the test model to identify the optimal configuration and performance difference. More information about the RF gain of the BLE113 modules in different directions can be found in subsection 5.3.11.

Figure 19: Nanosatellite test model with unpopulated PCBs.

Two BLE modules were used to conduct the experiments. One module was acting as the master (client) and the other as a slave (server). The master represents the module connected to the OBC while the slave is a sensor node or subsystem.

5.3.7 *BLE113 Module Specifications*

In figure 20 the schematics of the BLE113 module can be found. The pins are configured as follows: [10]

- **Pin 9 to 33:** Configurable I/O port.
- **Pin 23 and 24:** Configurable I/O port with 20 mA driving capability.
- **Pin 14 (SCL):** Can be used as I²C clock pin or digital I/O. Leave floating if not used. If grounded disable pull up.
- **Pin 15 (SDA):** Can be used as I²C data pin or digital I/O. Leave floating if not used. If grounded disable pull up.
- **Pin 8 and 17:** Supply voltage 2V - 3.6V

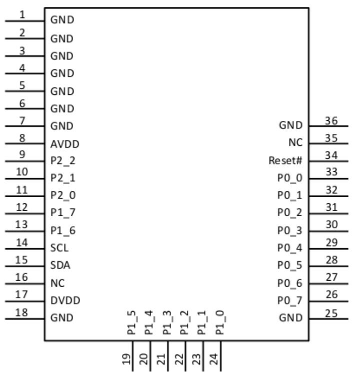

Figure 20: Schematics of the BLE113 module. [10]

5.3.8 *Measuring Error Rates*

The BER of the BLE module connections needed to be measured. According to the Bluegiga API [54] counters reflecting information about sent and received packets can be retrieved by the ´Get Counters´ command. Once the request has been made the counters are reseted and the following response is given:

- **txok:** Acknowledgements received for sent packets.
- **txretry:** Number of packets retransmitted.
- **rxok:** Packets received were CRC was ok.
- **rxfail:** Packets received were CRC failed.

- **mbuf:** Available packet buffers.

The PER is easily calculated by the following equation:

$$PER = \frac{rx_{fail}}{(rx_{ok} + rx_{fail})} \tag{7}$$

The PSER can be calculated by:

$$PSER = \frac{tx_{retry}}{(tx_{ok} + tx_{retry})} \tag{8}$$

These error numbers can be retrieved at the end of an experiment or after every package burst and save them in storage. There is no easy way to retrieve the BER through the module API. Only by means of comparison of the transmitted bits and the actual received bits one can determine the BER of a connection event.

5.3.9 *Measuring the Power Consumption*

It was desired that the power consumption of the BLE113 modules is continuously monitored during the tests. However, because of the low power levels of the BLE113 modules, special consideration needed to be given to measure the power consumption accurately. In order to achieve a high power consumption measurement resolution, special equipment was needed. For example, high sensitivity current probes could have been used to achieve these accurate measurements. These probes were unfortunately not available in the Delfi clean room. Therefore, the NI USB-6008 DAQ was used to effectively monitor the power consumption of the BLE modules. A small LabVIEW program was written to measure the power consumption and store it on a computer. This stored data was correlated with the actual transmission characteristics to determine the overall performance.

5.3.10 *Configuration Settings of the Modules*

The BLE113 module needs to be configured and the application configurations need to be set. In the ¨Bluetooth Smart – Configuration guide¨ [53] supporting documentation can be found. First of all, the Bluetooth Smart Software Development Kit (SDK) should be installed on a computer to have the appropriate drivers to compile BLE projects into hex files which can be flashed onto the BLE113. Also a CC debugger is necessary to be able to flash the BLE projects onto the modules.

The first file that needs to be defined is the *project.bgproj* file which specifies the basic settings of the module and refers to the other files which need to be included in the project. An Over-the-air (OTA) flash update option is available for the BLE113 module, but will not be used to avoid additional complexity and potential errors. The UART interface will be used and therefore the following XML code needs to be included in the project file for example:

Listing 1: XML code in the project.bgproj file.

```
<?xml version="1.0" encoding="UTF-8" ?>
<project>
      <gatt in="gatt.xml" />
      <hardware in="hardware.xml" />
      <config in="config.xml" />
      <device type="ble113" />
      <boot fw="bootuart" />
      <image out="BLE113.hex" />
</project>
```

In order to configure the hardware features a hardware.xml file needs to be created as well:

Listing 2: XML code in the hardware.xml file.

```
<?xml version="1.0" encoding="UTF-8" ?>
<hardware>
      <sleeposc enable="true" ppm="30" />
      <usart channel="0" alternate="2" baud="19200"
          endpoint="none" flow="false" />
      <usb enable="false" endpoint="api" />
      <txpower power="15" bias="5" />
      <script enable="true" />
      <port index="1" tristate="0" pull="up" />
</hardware>
```

The software features can be configured by defining the config.xml file:

Listing 3: XML code in the config.xml file.

```
<?xml version="1.0" encoding="UTF-8" ?>
<config>
      <connections value="1" />
      <script_timeout="0" />
      <throughput optimize="balance" />
</config>
```

Furthermore, the BLE application code has to be developed for both server and client. This BGscript projects can be compiled and flashed onto the BLE modules by means of a CC debugger.

5.3.11 *Module Placement & Orientation in the Nanosatellite*

Once basic experience with the modules had been established, it was time to test the performance of the modules when they have to operate in various orientations attached to or in a nanosatellite. Before locations and orientations of the BLE113 module were chosen the radiation profile/pattern of the BLE113 antenna needed to be analyzed first. In order to have a basic insight to the RF gains of the modules the radiation patterns of the BLE113 antenna characteristics specifications [11] are shown in figure 21, 22 and 23. The white rectangle in the figures represents the antenna of the module.

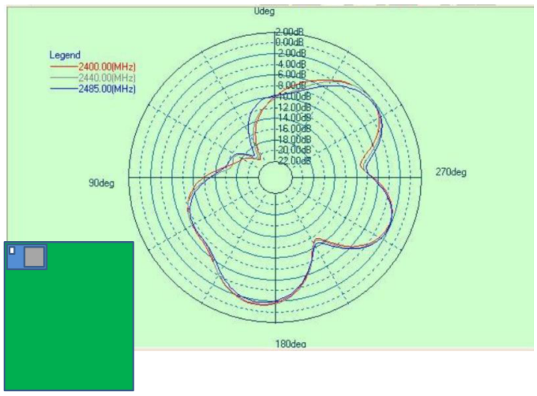

Figure 21: Top view of the BLE113 radiation pattern. [11]

In figure 21 it can be seen that the modules have the weakest gain in the second quadrant of the ground plane (northeast). The other quadrants have better gains except for some small dips.

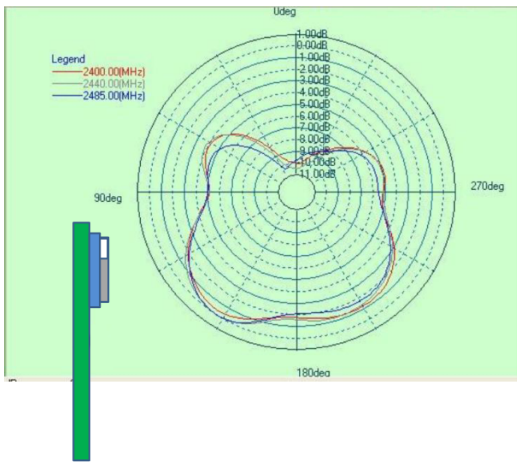

Figure 22: Front view of the BLE113 radiation pattern. [11]

The highest gains can be found parallel to the ground plane as displayed in figure 22. The gains are higher if the other module is located in the third or fourth (south) quadrants.

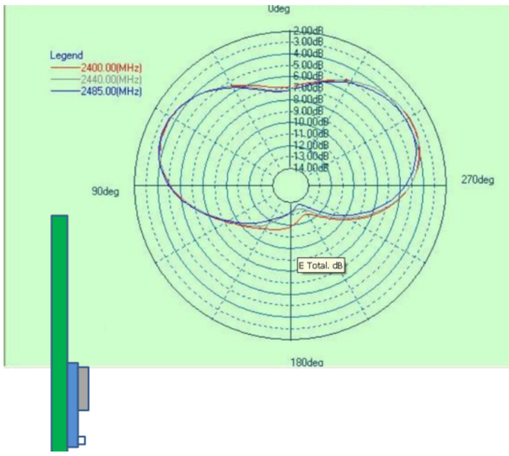

Figure 23: Side view of the BLE113 radiation pattern. [11]

If only two nodes are used the optimal location and orientation of the BLE modules can be defined. Assume the master BLE is located in the horizontal plane on the OBC in the center of the nanosatellite. Then the optimal orientation of a BLE slave module mounted on the side of a nanosatellite wall - perpendicular orientation with respect to the master BLE module - depends on whether it is located below or above the OBC as can be seen in figure 23. The results of the tests with different module orientations are given in subsection 6.1.5.

6 BLUETOOTH LOW ENERGY EXPERIMENTS RESULTS

Note that most of the contents of this chapter are submitted as part of a paper to the 4S symposium 2014 [38].

In this chapter the results of the on ground wireless intra-spacecraft communication experiments for nanosatellites as proposed in chapter 5 will be discussed. The problems encountered, observations made and performance with respect to different aspects during the experiments will be treated. In section 6.1 the results of the conducted experiments are presented and discussed. Hereafter, in section 6.2 the advantages and disadvantages of wireless versus wired data links for nanosatellites are compared. Finally, in section 6.3 conclusions are made and recommendations for future wireless intra-spacecraft communication studies are provided.

6.1 RESULTS & ANALYSIS OF EXPERIMENTS

In this section the measurements of the conducted experiments as outlined in the previous section are presented and discussed. An analysis related to the goals stated in section 5.3.1 is made.

6.1.1 *Connection Stability*

No special observations were made about the connection stability of the BLE connection in the nanosatellite test model. The connection was quickly established and no lost connections were noticed. Nevertheless, even if the connection is lost, it should automatically re-establish itself. Also, the integrated timers prevent a stuck module if an error arises.

6.1.2 *Power Consumption*

Power consumption measurements of the modules were conducted to identify the implications of various module settings. The first power measurement set was used to identify the difference in power consumption for different TX settings. In table 13 an overview of the average power consumption by both the server and collector for different modes can be found. As one can see the difference in power consumption is marginal for different TX settings and power profiles. Also, it is important to realize the TX settings only affect the peak power consumption for very short time durations (~1 ms) and because of the duty cycle it does not affect the average power consumption to a significant extent. In addition the TX peak current consumption of the BLE113 for different TX settings only varies with ~2 mA between the lowest and highest TX power setting.

Table 13: Average power consumption for different TX settings and power profiles.

	Average power consumption [mW]				
Setting	**Power**	**Performance**	**Balanced**		
TX	**-24 dBm**	**-24 dBm**	**-24 dBm**	**-12 dBm**	**0 dBm**
Server	7.1	7.1	6.2	7.1	7.1
Collector	16.5	16.4	16.5	16.3	16.6

It might be concluded the different TX settings and power profiles do not make much of a difference in the used case of the experiments (100 bytes/sec). However, in an actual implementation for which higher throughputs are required, the difference in average power consumption for the different power profiles and TX settings might increase because the module is simply transmitting more often. On the other hand, for the anticipated implementation of a wireless temperature node with an update frequency of 0.1 Hz on DelFFi the total average power consumption for the collector and server is estimated to be 0.6 mW.

6.1.3 *Influence of Module Distance*

In the first PER measurements the influence of the distance between two BLE modules in the nanosatellite test model was analyzed. The amount of PCBs (eight) plus battery pack was kept constant and the experiments were conducted in two different BLE113 TX power settings. Note that the influence of distance was also measured for just one PCB in order to have a reference measurement available. Unless specified differently all tests were conducted in the balanced profile mode.

One measurement set was taken in the almost lowest TX power (-24 dBm) setting possible on the modules while the other was logged at a TX power setting of -12.5 dBm, which is a medium power level for the BLE113. The highest TX power setting (0 dBm) was not monitored in these tests because it was already determined almost no packet errors (< 0.03 %) arose on the collector side if this power setting is used.

In figure 24 and 25 the results of these distance related measurements for the PER and PSER can be found respectively. There is no apparent correlation between distance and PER/PSER. It can be seen that in some cases the PER and PSER are lower if only one PCB is present. There are some exceptions though, which can be found mainly in case of the measured PER at 20 cm and PSER at 25 cm and hence no conclusions can be drawn from these results.

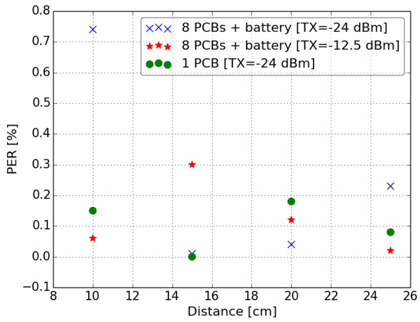

Figure 24: Distance versus measured Packet Error Rate of the client in the test model.

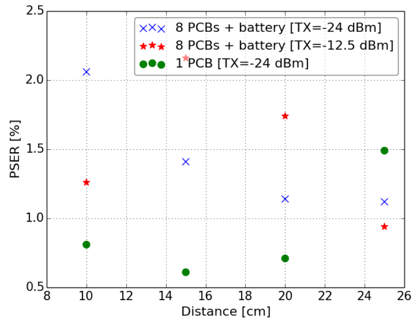

Figure 25: Distance versus measured Packet Sent Error Rate of the server in the test model.

6.1.4 *Amount of PCBs & Power Setting*

In the second set of PER measurements the influence of the amount of PCBs between two BLE modules in the test model was monitored. During these tests the distance (25 cm) was kept constant, the PCBs were equally spaced and no battery was included. In figure 26 a plot of both the PER and PSER can be found in case of different amount of PCBs. No obvious relationship between the amount of PCBs and PER/PSER can be identified. The PER is just as good for two PCBs as it is for eight PCBs, but the PSER is worse. However, the PSER is better for six PCBs than if two PCBs are used and hence no clear relationship with the amount of PCBs can be derived.

The goal of the third set of PER measurements was to determine the influence of the TX power setting. In figure 27 a plot of the PER and PSER for different TX power settings can be seen. It can be seen that the PER for the lowest TX power setting (0.2 %) is higher than the PER for the other TX power settings (< 0.1 %). For the highest TX power setting the PER is actually smaller than 0.03 %. The PSER varies between 1.0 % and 1.8 % and no strong conclusions can be made about this parameter.

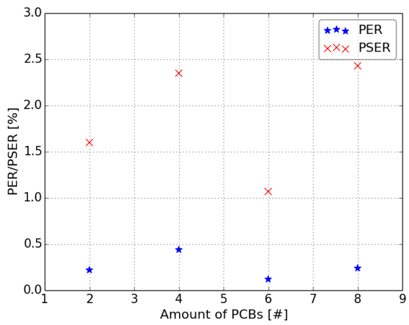

Figure 26: Amount of PCBs versus Packet (Sent) Error Rate of the client and server in the test model with a distance of 25 cm. (without battery)

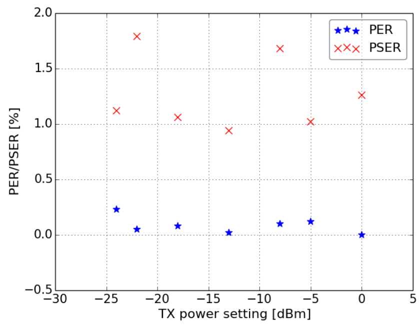

Figure 27: TX power setting versus Packet (Sent) Error Rate of the client and server in the test model for 8 PCBs + battery pack with a distance of 25 cm.

6.1.5 *Influence of Power Profiles*

In addition to all the foregoing measurements a few tests concerning different power profiles were conducted as well. The influence of the different power profiles: balanced, power and performance on the PER and PSER was checked. The definitions of the different power profiles were stated before in subsection 5.3.5. In table 14 the PER and PSER for the different power profiles can be viewed. It can be derived from the table that the PER and PSER are indeed higher for the performance mode than if the power or balanced modes are used. Nonetheless, this is only the case if TX is set to -24 dBm because if TX is set to -12 dBm the performance mode has actually a very low PER and PSER. This observation might indicate a correlation between TX power setting and profile mode used for the BLE113.

Table 14: Influence of power profiles on PER and PSER.

	TX = -24 dBm			**TX = -12 dBm**		
Setting	**Power**	**Balanced**	**Performance**	**Power**	**Balanced**	**Performance**
PSER [%]	1.73	1.4	2.12	2.06	1.66	0.35
PER [%]	0.22	0.22	0.40	0.10	0.11	0.02

The RSSI for different orientations of the BLE113 modules was monitored such as perpendicular with respect to each other and different rotations as well. These different orientations were tested because of the non omnidirectional gain of the BLE antennas as was explained in subsection 5.3.11. Note that the radiation pattern depends on the layout of the mother board and other parameters. For example, the PCB substrate and the impedance pairing between the microstrip and the output stage of the amplifier affects the radiation pattern as well. Only a few dBm difference for the RSSI was found for some orientations in the measurements but no orientation affected the PER or PSER significantly.

6.1.6 *Electromagnetic Interference Compatibility*

For a reliable nanosatellite all subsystems need to be checked for Electromagnetic Interference Compatibility (EMC). There are a few EMI related mechanisms which can cause interoperability problems with wireless systems onboard a spacecraft [55]:

- **Out-of-band emissions:** Radiating systems can emit some radiation outside their band. For example, harmonics of the used band, local oscillators or leakage of intermediate frequencies.
- **Out-of-band sensitivity:** Receivers usually have some sensitivity outside their operating band. Any receivers onboard a nanosatellite whether it concerns BLE modules or space to ground transceivers can experience interference if not checked properly for EMC.
- **Intermodulation products:** These are often caused by poorly screened PCB tracks or RF stubs which are coupled with mixers and generate other frequencies.

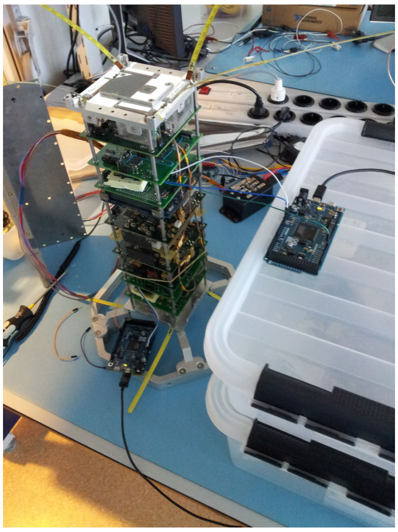

Figure 28: Picture of Delfi-C3 spare model Bluetooth Low Energy test setup.

The influence of EMI effects on the wireless BLE link were determined by using the Delfi-C3 spare satellite (see figure 28). The same BLE server and collector configuration was used as in the other tests. In table 15 the PER and PSER measurements for these tests can be found. The PER and PSER were clearly higher if the Delfi-C3 spare satellite was turned on than when the Delfi-C3 was unpowered. It is especially interesting to see that the PER/PSER increased significantly and was higher for the closer distance/fewer PCBs case when the satellite was turned on. Most likely this can be attributed to the fact that the transceiver was located very close to the place where the modules were placed for the close distance case. Hence, harmonics of the Delfi-C3 spare satellite transceiver might have been a disturbing factor which caused this effect. Nevertheless, a PSER of 3.44 % is still acceptable and does not have a major impact on the performance of the modules. Note that after a faulty transmission a new attempt to transmit the data is made and thus the PSER is not actual lost payload data as was already explained in subsection 5.3.3.

Table 15: Delfi-C3 EMI influence on PER and PSER.

Power	Distance [cm]	PCBs [#]	PER [%]	PSER [%]
Off	20	7	0.24	2.05
On	20	7	0.34	2.41
Off	5	3	0.11	0.86
On	5	3	0.88	3.44

Another point of concern is the fact that BLE operates at a frequency of 2.4 GHz and thus might interfere with any S-band systems on a nanosatellite [55]. The BLE protocol makes use of Adaptive Frequency Hopping (AFH), which ensures occupied frequencies are avoided. Besides the mitigation of co-channel interference also frequency selective fading effects can be prevented by AFH. Even though AFH is implemented in BLE, it is recommended tests are conducted with an S-band transceiver in the vicinity of BLE113 modules to exclude any serious RF interference between these systems.

6.1.7 *Development Complexity*

The development of the software for the experiments was done by using BGScript. It makes programming the modules relatively easy although some low level code programming is still required. Though libraries could be written to handle typical low level procedures if efforts are combined. It takes about a week for an experienced software developer to understand the protocol properly but once developers gain experience with the protocol applications are developed rapidly. However, for a flight model implementation the use of the C API is highly recommended. This will provide more functionalities in the software such as the ability to use functions instead of procedures.

6.2 WIRED VERSUS WIRELESS NODES

In this section an evaluation is made about whether wireless links do have significant advantages over wired links in CubeSats and for which type of nodes, it can best be considered to implement a wireless link. The main findings of the conducted experiments as outlined in the previous section 6.1 are taken into consideration for this evaluation. These characteristics are compared with development experience of the Delfi-n3Xt and Delfi-C3 nanosatellites.

6.2.1 *Connection Stability*

A stable data connection is essential for a properly functioning satellite. During the development of Delfi-C3 and Delfi-n3Xt stuck bus events were often experienced with the I^2C interface. In-orbit experience with Delfi-n3Xt also learned that several OBC reboots per week occurred. These OBC reboots might indicate issues with the I^2C interface as well. Experience has shown that making I^2C reliable is a very labor intensive process. Nonetheless, a wireless interface for some nodes might be easier to implement because of the automatic re-establishment capability of the connection by BLE modules. If one wireless node has an error the other nodes can still exchange information without the need to reset the entire data bus.

6.2.2 *Power Consumption*

A big advantage of BLE modules is that they have built-in sleep mode capabilities. If these modules are smartly duty cycled a signifcant amount of power can be saved. Certainly if no frequent sensor measurements are required or the measured data does not need to be transmitted to the OBC immediately the power savings can be substantial. As was already presented in subsection 6.1.2 the average power consumption of a BLE113 server used for the experiments (1 kbps) was 7.1 mW and 16.5 mW for the collector, adding up to a total power consumption of 23.6 mW. An I^2C implementation would require 17 mW if a combination of I^2C pull-up resistors (2 x 3.3 V over 10 kΩ) and I^2C buffers (PCA9515 - 2.3 mA on 3.3 V [56]) are used.

A low data rate, low update frequency (0.1 Hz) wireless temperature sensor network only consumes 0.6 mW of power as was discussed in subsection 6.1.2. The server for this case consumes about 0.2 mW of power on average. If a CR2032 - 20 mm in diameter, 3.2 mm thickness - coin cell battery of 250 mAh operating on 3 V [57] is connected to a server node it could last for six months in the ideal case. To extend the lifetime of a wireless node the update frequency can be reduced or an extra coin cell can be added. Smaller coin (button) cells can be used as well but this would drastically reduce the lifetime of the wireless node to a few months.

The low power consumption can also be suitable for an external sensor with its own power supply. For example, a 1 Hz sun sensor with a power consumption of 2 mW can be powered by a 30 mm^2 solar cell.

6.2.3 *Packet Error Rates*

The data bus implemented on Delfi-n3Xt did not contain a standard CRC algorithm. It was reasoned a CRC was not essential because the BER on the Delfi-n3Xt data bus was lower than 10^{-6} [19]. Some parameters could be corrupted due to the lack of a CRC but this can be filtered out because usually large sets of data are analyzed. Although a zero PER is not required for many nanosatellites parameters the 24 bit CRC of the BLE113 modules would ensure that in real practical cases no parameters are erroneous.

6.2.4 *Throughput*

The throughput of BLE was already partially discussed in subsection 5.2.1. According to Bluegiga the highest reliable throughput which can be achieved with acknowledged packets is 8-10 kbps [49]. If non-acknowledged data transmission operations are used peak throughputs of about 100 kbps can be attained [50]. It can be concluded that the throughput depends on the required reliability of the link whether BLE performs better than I^2C in terms of throughput. If a high reliability is required (acknowledgements) the BLE113 modules can achieve only 10 kbps while I^2C can achieve 50 kbps because BERs are much lower and therefore no acknowledgements are necessary. However, if a lower reliability (no acknowledgements) is acceptable BLE113 modules can achieve 100 kbps because no acknowledgements have to be send.

6.3 CONCLUSION

The performance of Bluetooth Low Energy BLE113 modules in a representative nanosatellite has been monitored and evaluated. During the tests conducted the connection stability of the BLE113 modules was excellent. In case the highest transmit (TX) power setting (0 dBm) was used a very low Packet Error Rate (PER) was generated (< 0.03 %). Note that the PER does not represent actual lost data because an error is detected and the packet is transmitted again. No correlation between PER or Packet Sent Error Rate (PSER) and the amount of shields such as Printed Circuit Boards was found. Neither a relation between module distance in a nanosatellite and PER could be determined. Also, there was no impact of different power settings on the average power consumption of the BLE113 modules in the conducted tests. Different power profiles did affect the PER with the performance mode generating higher PERs (0.40 %) in low TX power settings of -24 dBm. Using different orientations did not alter the Received Signal Strength Indication (RSSI) significantly and no big difference in PERs was noticed as a result of these varying orientations.

Final tests with a Delfi-C3 spare satellite confirmed EMI effects disturbed the BLE signal and have a negative influence on the PER. Nonetheless, the PSER was still tolerable with maximum observed PSERs of 3.4 %. Note that after a faulty transmission a new attempt to transmit the data is made and thus the PSER does not represent actual lost payload data.

Implementing BLE113 module nodes as an alternative to an I^2C interface or other wired interfaces in a nanosatellite can be advantageous in specific cases. For low data rate, low update frequency sensor nodes BLE113 modules can be very power efficient in comparison with I^2C. A coin cell of 250 mAh could power the wireless node for six months and then the need for an electrical and data wire would be completely eliminated. It is therefore best to consider BLE as a replacement for nodes which are difficult to integrate.

The next step will be the development of the wireless BLE temperature boards for the DelFFi mission for which a proposal is made in the next chapter (7). To check if the BLE113 modules can survive the launcher and space environment some thermal and mechanical tests shall be conducted with the modules. In-orbit wireless temperature nodes experiments onboard DelFFi should eventually proof the radiation tolerance of the modules as well.

7 PROPOSAL FOR A WIRELESS BLE TEMPERATURE SENSOR EXPERIMENT ON DELFFI

In the previous two chapters the case for wireless intra-spacecraft communication and the results of the conducted experiments with BLE were presented. In this chapter a proposal for a wireless BLE temperature sensor experiment onboard DelFFi will be provided, based on the experience with the conducted on-ground tests. A wireless sun sensor experiment could be proposed for DelFFi as well, but is discarded to focus efforts on one wireless experiment only. The goals of the wireless BLE temperature sensor experiment are the following:

- Verify the reliability of a wireless BLE sensor node network in a nanosatellite in-orbit.
- Demonstrate the integration advantages of a wireless sensor node.
- Validate the space environmental compatibility of the wireless BLE113 module (e.g. radiation tolerance).

In the first section of this chapter the recommended communication architecture will be outlined. Subsequently, in section 7.2 the different options of possible power supplies for the wireless nodes are discussed. Hereafter, in section 7.3 considerations about the placement and integration of the wireless temperature nodes are given. Besides the technical aspects of the wireless BLE temperature sensor experiment also considerations with respect to the project schedule, risks and resources will be given in section 7.4. Finally, in section 7.5 a conclusion is made and requirements with respect to a wireless BLE temperature sensor experiment on DelFFi are proposed.

7.1 COMMUNICATION ARCHITECTURE

Several communication architectures are possible to implement between the wireless modules as listed in subsection 5.3.4. Only a few temperature measurements with a low update frequency are required for the wireless BLE temperature sensors and thus no high throughput is required. Therefore, indications can be used, which provide the highest reliability by acknowledging every successfully received packet. The alternative is to use burst transmissions in which no acknowledgements are used and therefore total radio on time can be lower, hence saving power. However, there is not sufficient experience/performance evaluation with burst transmission in the DelFFi team yet and thus this is left out of consideration for now.

The master module (client) should be integrated on the OBC board and the slave (server) modules shall be connected to non-critical independent temperature sensors on the nanosatellite. In order to save power the server modules need to be removed from their sleep mode. To preserve the maximum amount of power for the server modules they should only wake up every 60 seconds to listen for incoming requests of the client. The client on the OBC can be triggered by means of a telecommand to turn on the server. By another telecommand the client can try to turn off the server again. In this way, no unnecessary power is consumed before the satellite is launched. This method allows a sensor node with a completely independent power supply as will be discussed in subsection 7.2. Note that this method is solely used to switch on and off the experiment because during the experiments the sleep mode will be tuned differently. In figure 29 the communication architecture for the wireless BLE temperature sensor experiment can be found.

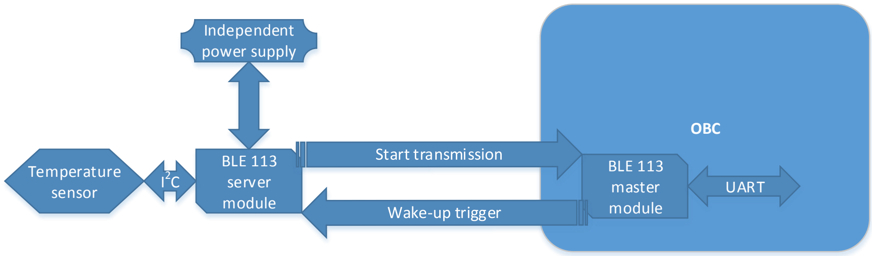

Figure 29: Wireless BLE temperature sensor experiment communication architecture.

In addition to the temperature measurements, it is also interesting to log various parameters of the modules for analysis of the wireless link. Mainly parameters which are used to derive the PER and PSER of the modules which were stated in subsection 5.3.8 need to be retrieved. Also, the internal temperature of the BLE modules shall be stored and have the possibility to be downlinked to the ground. This will allow mission analysts to correlate the performance of the BLE modules with the internal temperature of the modules. The interface of the BLE client module with the OBC will be performed through the UART interface because all the other BLE113 interfaces such as I²C only support peripherals [58].

7.2 POWER SUPPLY & STORAGE

There are three identified options for the power supply of the wireless temperature nodes:

1. Powered by a coin cell.
2. Powered by a small solar cell.
3. A hybrid solution by using a solar cell and:
    a) super capacitor
    b) rechargeable coin cell

Powering the wireless temperature node by a coin cell is a straightforward solution. However, adding a small solar cell as well can prolong the lifetime of the wireless temperature node. Using only a small solar cell will limit the operations of the wireless experiment in eclipse and only to sensors mounted outside the nanosatellite unless a wire of the server node is connected to a small solar cell mounted on the nanosatellite body. This small solar cell can even charge the coin cell somewhat so it takes a longer time to run out of power. Instead of a coin cell a super capacitor can be used as well which offers a low Equivalent Series Resistance (ESR) compared to a coin cell and therefore super capacitors deliver a high power density. In figure 30 a conceptual circuit diagram outlining the concept of using a super capacitor and a solar cell to power the BLE module can be found.

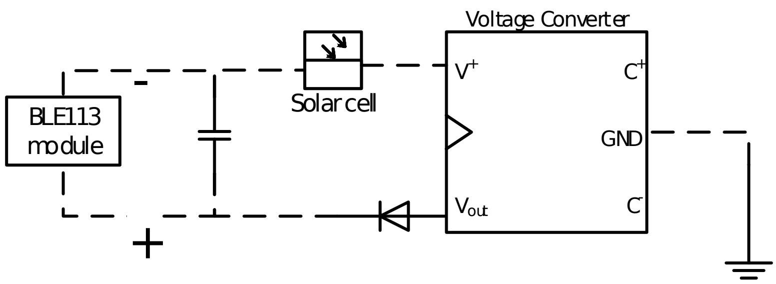

Figure 30: Conceptual circuit diagram of the wireless BLE experiment with a super capacitor and solar cell.

In earlier power consumption calculations which can be traced back to subsection 6.2.2, it was estimated one CR2032 coin cell can power a server node for about six months. If it is chosen to power the wireless temperature experiment with a coin cell, it is recommended two CR2032 coin cells are used, because the actual power consumption might be a bit higher and loss of battery charge will occur. Also, if the DelFFi nanosatellites are in storage for a few months before the launch some power will be drawn from the coin cells and therefore a second coin cell could theoretically extend the lifetime of the server nodes to one year.

7.2.1 *Super Capacitor Sizing*

Some further analysis is required in order to determine if super capacitors are a good alternative to coin cells for the wireless node experiment. An approximation to calculate the capacitance required can be made by performing an energy balance [59]:

$$C = 2\left(\frac{E_{load}}{V_{init}^2 - V_{final}^2}\right) \tag{9}$$

In which C is the capacitance (F), V_{init} is the initial super capacitor voltage (V) and V_{final} the minimum voltage (V) that the super capacitor is able to discharge after the peak load. The E_{load} represents the load energy (J) and can be defined as follows:

$$E_{load} = P_{avg}t_{load} \tag{10}$$

Where P_{avg} is the average load power in Watts and t_{load} the load duration in seconds. Note that equation 9 does not take ESR (Ω) into account and hence can only be used if the following statement for the voltage drop holds:

$$I_{load}ESR \ll V_{final} \tag{11}$$

The I_{load} is the current (A) drawn by the load so in this case the BLE113 module and temperature sensor. Typical super capacitors which are available on the market have an ESR of 100 mΩ. Now to verify if the approximation for the capacitance of equation 9 can be used statement 11 needs to be checked:

$$0.010[A] \cdot 0.1[\Omega] \ll 2.7[V] \tag{12}$$

It can be seen this statement easily holds and thus the approximation is valid. First, the total BLE radio on time during eclipse needs to be determined to calculate the load energy. According to the latest orbit insertion specifications of the QB50 mission (350 - 400 km altitude) [23] the worst case/longest duration eclipse will be t_{ecl}=2160 seconds and the worst case/shortest duration sun illumination is calculated to be t_{sun}=3148 seconds. The foregoing parameters were determined by using ESATAN output. Now the load duration can be calculated by using the following equation:

$$t_{load} = f_{meas}t_{ecl}t_{radio} \tag{13}$$

Where f_{meas} is the measurement frequency in Hertz and t_{radio} the BLE radio on time in seconds for each transmit event. As defined earlier the system will be designed for a measurement frequency of 0.1 Hz and the BLE radio on time was found to be approximately 0.18 seconds. Substituting these values in equation 13 indicates a total load duration of 39 seconds. Note that most of the time the BLE module is in sleep mode and has a very low power consumption which is in the order of micro Ampères. Nevertheless, to also take this sleep mode into account and leakages the total load duration is assumed to be 60 seconds. In earlier BLE power measurements of which the results are outlined in subsection 6.1.2 the power consumption for one BLE radio on time event was found to be 30 mW. By using equation 10 it was determined that a load energy of 1.8 J is required.

Furthermore, by means of practical experience it is known the BLE modules function well between 2.7 and 3.3 V. Therefore the V_{final} is set at 2.7 V and the V_{init} is equal to 5 V to decrease the capacitance need. This 5 V exceeds the maximum operating voltage of the BLE113 of 3.3 V and hence a voltage regulator needs to be incorporated in the electrical circuit. Calculating the capacitance with equation 9 by using all the previously defined values results in a required capacitance of 0.2 Farad or 200 mF. Note that in this subsection merely a preliminary analysis for the required capacitance for the wireless temperature node experiment has been made. More detailed investigation is desired for a more accurate and optimal determination of the required capacitance.

7.2.2 *Super Capacitor Selection*

There is a tremendous amount of super capacitors available on the market today and they come in all types, shapes and sizes. For this analysis only electrolytic and tantalum capacitors are taken into consideration. Tantalum capacitors are widely used in electronics because they have a few advantages over electrolytic capacitors [60]:

- Very low electrical leakage.
- Retain a charge for a long duration.
- More tolerant of high temperature environments.

Nevertheless, tantalum capacitors also have some disadvantages [60]:

- Relatively expensive.
- Less tolerant of heavy charge and discharge currents.
- The ESR can also be somewhat higher than aluminum electrolytics.

Although tantalum capacitors may seem the way to go there is one major problem for this wireless node application. The tantalum capacitor found with the highest capacitance available on the market is the STE10000-10T4MI [61] which has a capacitance of 10 mF. It was already determined in subsection 7.2.1 that a capacitance of about 200 mF is required so this means 20 of these tantalum capacitors need to be placed in parallel. If it is considered the STE10000-10T4MI is 10 mm in diameter and 27 mm in length it can be concluded tantalum capacitors are not feasible for the wireless node experiment.

A small search for suitable capacitors resulted in the two options as outlined in table 16.

Table 16: Selection of super capacitors potentially suitable for the wireless temperature experiment.

	EDLC252520-351-2F-21 [62]	**DMT334R2S474-M3DTA0 [63]**
Capacitance [F]	0.35	0.47
ESR [$m\Omega$]	70	130
Nominal voltage [V]	3.2	4.2
Rated voltage [V]	5.5	4.2
Dimensions [mm]	25 x 20 x 2.5 **(l x w x h)**	21 x 14 x 3.5 **(l x w x h)**

Note that these two super capacitors were selected to determine the approximate volume required. The selection of the super capacitor for the eventual application requires a more extensive search and trade-off. Also note that the placement of several lower capacitance super capacitors in parallel or series can turn out to be more efficient. In figure 31 and 32 the EDLC252520-351-2F-21 and DMT334R2S474M3DTA0 respectively can be found.

Figure 31: Picture of the EDLC252520-351-2F-21 super capacitor. [12]

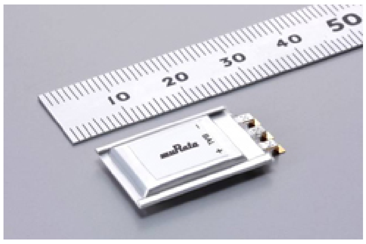

Figure 32: Picture of the DMT334R2S474M3DTA0 super capacitor. [13]

The volume of the selected super capacitors is about the same of a CR2032 coin cell. Therefore, it can be a nice replacement for a coin cell, allowing easier charging and more discharge/charge cycles. However, a rechargeable coin cell is probably able to support the wireless node as well and will have a much higher initial charge than a super capacitor. Nonetheless, rechargeable coin cells will not offer the capacity a non-rechargeable coin cell offers. Also a typical rechargeable coin cell such as the LiR2450 only has a cycle life of 500 times [64].

7.2.3 *Solar Cell Sizing*

Limited space is available on the DelFFi body panels for an independent additional solar cell to power the wireless temperature node experiment. Mainly because space is already allocated for the primary solar cells, sensors and GPS patch antennas. In figure 33 the four areas which are still available (green / A, B, F, H) for a small solar cell can be seen. As can be derived from figure 33 and summarized in table 17 the body panel with a hinge has very restricted dimensions in available due to the GPS patch antenna. Location options one and two only allow solar cells with a maximum width of 15 and 11 mm respectively. Option three on the body panel with a hinge has more proportionate dimensions of 33 x 18 mm. Therefore it is preferred to place the additional solar cell on the body panel with a hinge. However, two wireless server nodes are planned on each satellite and hence two body panels are required. The current body panel configuration defines that one body panel with a hinge is in full illumination and the other is not. This will imply the incoming power for this panel will be approximately one third in comparison with the one in full illumination mostly due to albedo effects. Nevertheless, this is probably still enough because the solar cell will be oversized anyway. It still needs to be evaluated whether a solar cell also fits in location option four because this allows more freedom in the placement of the other wireless experiment. It does not make sense to place the two wireless sensor experiments very close to each other.

Table 17: Space available for an additional independent solar cell on the DelFFi body panels.

Option [#]	Length [mm]	Width [mm]
1 / A	15	17
2 / B	47	11
3 / H	33	18
4 / F	45	13

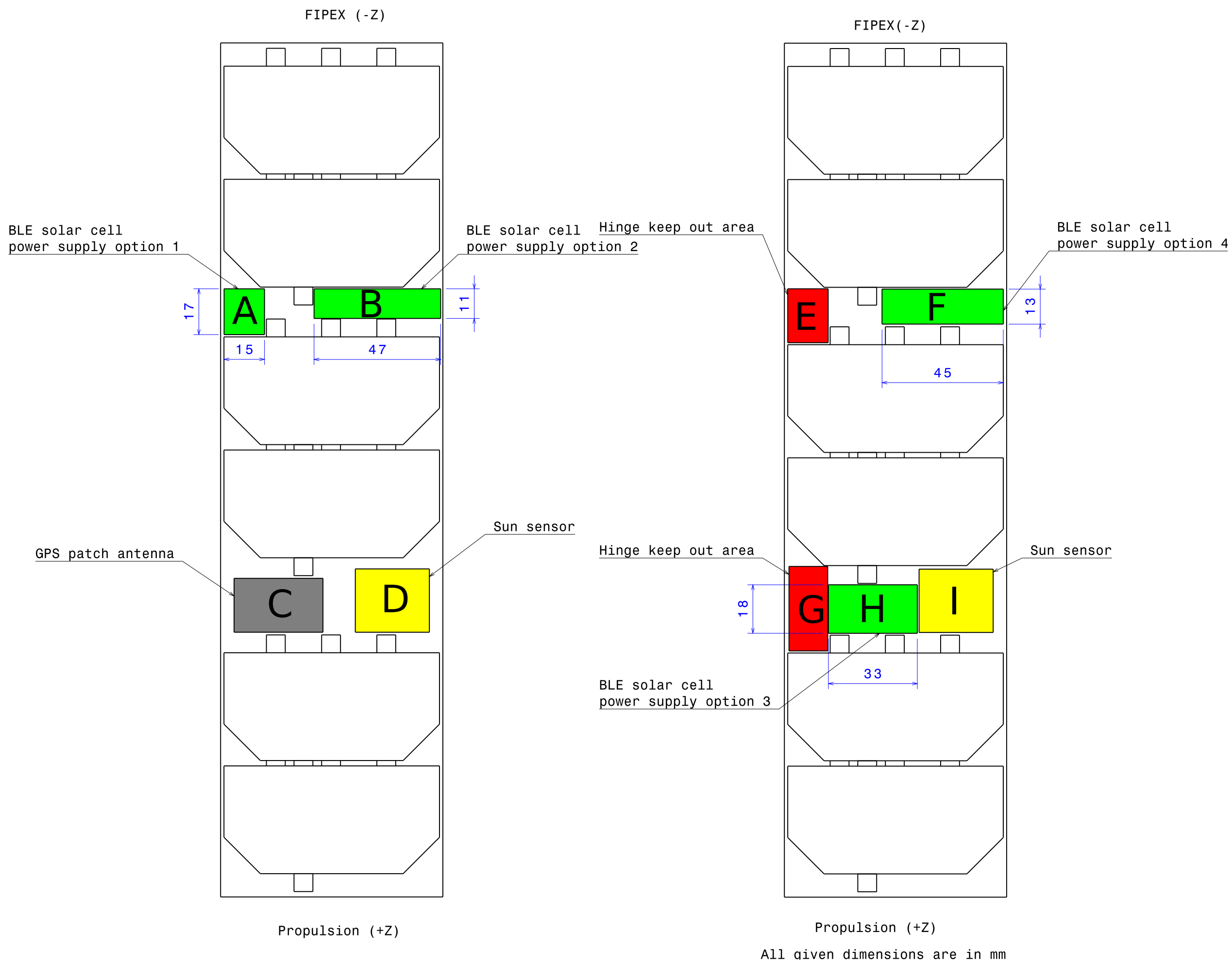

Figure 33: Physical layout and keep out areas of the current DelFFi body panel configurations. The left panel represents the configuration without a hinge and the right panel one with a hinge.

As was discussed in subsection 6.2.2 the average power consumption of a 0.1 Hz BLE server node is 0.2 mW and hence this will be the driving factor for the power supply. Assume 0.2 mW is generated by the solar cells for near term usage and a 0.2 mW surplus to charge the super capacitors or coin cells for eclipse mode, which makes a total required power of 0.4 mW. This is more power than required because the sun mode lasts one third of the time longer than the eclipse mode as was stated in subsection 7.2.1 but this margin will cope for inefficiencies and uncertainties. A preliminary solar cell size area A_{cell} (m^2) estimation can be calculated with the following equation [65]:

$$A_{cell} = \frac{P_{req}}{S \cos i \eta_{cell} (1 - D)} \tag{14}$$

In which P_{req} (W) is the desired power delivery of the solar cell, S is the solar irradiance (W/m^2), i is the solar incidence angle (degrees) with respect to a perpendicular axis to the solar cell, η_{cell} the efficiency of the solar cell (-) and D is the degradation factor (-). The P_{req} was already defined as 0.4 mW, the solar irradiance is 1367 W/m^2 and a typical degradation factor for solar cells in space is 0.20 [-] [66]. An average worst case incidence angle of 70 degrees is assumed and a solar cell efficiency of 27 %, which is one of the best efficiencies available for a space tolerant small solar cell [67]. Substituting all these values in equation 14 results in a required solar cell area of 4 mm^2 to power one wireless server node. Note that this is a preliminary analysis but it at least gives an indication of the solar cell size required.

7.2.4 *Solar Cell Selection*

In the previous subsection 7.2.3 it was determined that the required solar cell size is 4 mm^2 to power one wireless sensor node. The super capacitors selected in subsection 7.2.2 operate between 2.7 V and 5 V. Therefore, the solar cell needs to be able to provide a voltage of at least 4 V or placed in series to achieve this voltage as well. A quick search on the websites of major electronics suppliers such as Mouser, Farnell and Digi-Key learns that there are no suitable COTS solar cells available. First of all, the dimensions really restrict the amount of options available and the voltage at Maximum Power Point Tracking (MPPT) is generally also too low to meet the specifications of the super capacitors. Another and maybe even more important point of concern is that the behavior and performance of these solar cells in a space environment are unknown. It is thus recommended one of the solar cells, which is already analyzed by DelFFi engineers will be used for this application.

The solar cell type which seems to be most suitable to power the wireless experiment are the Triangular Advanced Solar Cells (TASC) of

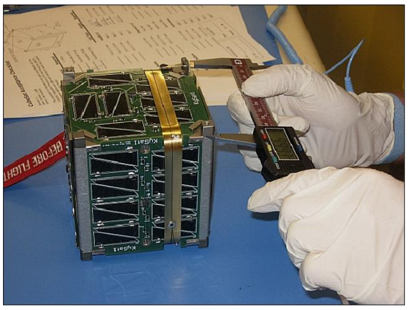

Figure 34: KySat-1 during CubeSat acceptance testing. [14]

Spectrolab [67]. These are triple-junction gallium arsenide solar cells, which cope relatively well with the space environment. In figure 34 it can be seen how these TASCs are attached to the KySat-1. Two of these TASCs need to be placed in series to achieve a voltage of 4 V (V_{mp} = 2.19 V) and this will deliver a current of 56 mA (I_{mp} = 28 mA). Although this is the ideal case the solar cell delivers way more power than required. Nevertheless, with all the restrictions and considerations stated earlier, it is still recommended to implement these solar cells as a power supply support for the wireless experiment on DelFFi. Perhaps for future nanosatellite missions more sensors can be placed on one wireless node or a higher update frequency can be used to increase the utilization of the power generated. If two of these TASCs are placed in series a total area of 16 x 32 mm [67] is required and this would fit in location option three as defined in subsection 7.2.3.

7.3 PLACEMENT & INTEGRATION

The DelFFi nanosatellites will be packed with subsystems and sensors. Therefore, a suitable location to place one or two BLE temperature sensor nodes shall be determined. In addition, mission critical sensors should not be hindered by the experimental BLE temperature sensor nodes. For Delfi-n3Xt the TMP275 temperature sensors [68] were used. Maybe another temperature sensor needs to be selected for DelFFi but for the analysis in this chapter that is left out of consideration. The TMP275 can operate at a voltage supply range of 2.7 V to 5.5 V and has an I^2C digital output interface. These characteristics are compatible with the pin outs of the BLE113 module.

Different power supply options would require different placement options:

- As recommended in the previous subsection (7.2) two CR2032 coin cells should be used if there is no recharging capability (solar cells) present in the server node. The dimensions of the CR2032 are 20 mm in diameter and 3.2 mm. [57] Preferably the coin cells are stacked on each other without being directly connected, but in parallel to double the capacity and not the voltage. Stacking the coin cells on top of each other would reduce the total surface area required for the wireless temperature node.

- If rechargeable coin cells or super capacitors are used the solar cells shall be placed at location option three (see figure 33). Wires should be pulled through small holes in the body panel and preferably not too long wires shall be used. The power storage and BLE module plus sensor can best be placed in the vicinity of the solar cell to avoid new integration complexity by again introducing wires.
- The dimensions of the BLE113 module are 9.15 x 15.75 x 2.1 mm (l x w x h). [11] In principle the BLE module can be stacked on top of the two coin cells or a super capacitor as well but then the overall thickness will increase to 8.5 mm and more likely 12 mm if the PCB is taken into account as well. Inside the nanosatellite body this might pose a problem if the satellites do not have much interspacing. Because in that case there only is a margin of 4 mm offset of the nanosatellite wall available. On the outside nanosatellite body panels this should not exceed the maximum 9 mm extension thickness as set in QB50-SYS-1.1.3 [23].

7.4 ESTIMATED DEVELOPMENT EFFORT & RESOURCES

The DelFFi project experiences time constraints and therefore resources should be allocated carefully. In this section the development effort is estimated and specified. Two clear tasks can be identified:

1. The development of the wireless temperature nodes including power supply, temperature sensor and BLE module server software.
2. The integration of the master (client) BLE module on the OBC and the software to connect to the servers of the wireless temperature sensors. Also the code on the client to send data to the OBC and the OBC software to handle this data needs attention.

The costs of the wireless BLE temperature sensors is very low in comparison with the costs of the entire satellite. At least the hardware costs are not significant and mainly the amount of man hours can be considered to be costly. An estimation of the hardware costs for the BLE experiment is outlined in table 18. These costs are based on a client and two server nodes on each DelFFi satellite. Note that the TASCs need to be ordered in bulk and therefore the total cost of such an order is allocated in this overview for sake of completeness. Nonetheless, the remainder of these cells can be used for other Delfi projects as well.

Table 18: Estimated hardware costs for the wireless BLE temperature sensor experiment.

Item	Cost [EUR]	Amount [#]	Total cost [EUR]
BLE113 module	11.16 [69]	6	66.96
TMP275	1.70 [70]	4	6.80
Supporting circuitry	100.00	-	100.00
TASC-1 [71]	Minimum order 50	4	110.00
TASC-2 [71]	Minimum order 50	4	110.00
CC debugger	43.52 [72]	1	43.52
			437.28
CR2032 coin cell	0.22 [73]	8	1.76
EDLC252520-351-2F-21	10.46 [62]	4	41.84

After consultation with the involved BLE hardware and software engineers - which supported the on-ground experiments as well - the development time was estimated. In table 19 an overview of the different tasks and the estimated time required for one engineer to finish each task can be found. Note that most tasks can be conducted in parallel.

Table 19: Estimated development time & resources required for the wireless BLE temperature sensor experiment. The time required is based on the work of one full time engineer.

Task	Time required [weeks]
OBC - BLE master module software	3
BLE temperature sensor PCB development	1
BLE client - BLE server software	3
Additional application layer & testing	2
Electrical testing	1
	10

Another task which needs to be performed is to test the EMC of the BLE modules with the GAMALINK device. As discussed before in subsection 6.1.6 the BLE modules can create certain EMI effects which might create interoperability issues with other RF devices and S-band systems in particular. Therefore, some compatibility tests should be performed as soon as the GAMALINK device is in-house. Other S-band equipment might be used in replacement for the GAMALINK device to conduct the EMC tests. However, it is recommended the exact hardware is used in the final tests to avoid any hardware specific problems.

7.5 conclusion & requirements

If the wireless experiment was to be specifically designed for the DelFFi mission, only two coin cells should be used. These two coin cells would be enough to power the experiment for the design lifetime of one year. In terms of simplicity and ultimate freedom in positioning the wireless sensor this would be the best option. However, if future applications are kept into consideration the wireless experiment needs to be designed for longer duration missions and more power intensive applications as well. Therefore, a renewable power supply by means of a solar cell is recommended. Whether super capacitors or rechargeable coin cells shall be used can be decided by the PCB design engineer.

Based on the analysis in this chapter the following requirements for the wireless BLE temperature sensor experiment for DelFFi can be defined:

1. The wireless temperature sensors shall not endanger the operation of other subsystems.
2. If a wireless node is placed outside the nanosatellite it shall not exceed 9 mm thickness. (QB50-SYS-1.1.3 [23])
3. If a wireless node is placed inside the nanosatellite it shall not exceed 4 mm thickness.
4. The BLE113 modules shall not be exposed to temperatures outside their operational temperature range of -40 °C to +85 °C [11].
5. The wireless temperature sensor servers shall be able to operate for at least six months by means of an independent power supply.
6. The wireless server node shall be able to survive a storage time of eight months.
7. The wireless server nodes shall be connected to a small solar cell.
8. The wireless experiment shall be able to operate in both sun and eclipse.

8 CONCLUSIONS

In this thesis several topics related to the Command & Data Handling System (CDHS) for nanosatellites have been treated. In the following sections a summary of the main conclusions for each part of the thesis is given. In addition, answers to the relevant research questions are provided and some recommendations are made.

8.1 Delfi-n3Xt operations lessons learned

First, an analysis about the Delfi-n3Xt operations has been conducted. The main research question concerning the Delfi-n3Xt operations was:

"What aspects of the CDHS can be improved for DelFFi?"

Although Delfi-n3Xt experienced several daily Onboard Computer (OBC) reboots, this itself proved the data bus was sufficiently reliable and it was concluded the CDHS of Delfi-n3Xt performed well. However, one of the main lessons learned of Delfi-n3Xt operations is that the telecommand preparation and storage infrastructure on the ground for future Delfi missions needs to be automated. This automatic telecommand storage will enable the traceability of telecommands for the mission analysts.

Also, based on experience with Delfi-n3Xt telemetry which was not relevant anymore or could be updated less frequently, such as temperature and deployment data, it was concluded there is a need for a more flexible telemetry system on future Delfi missions. This reconfigurable telemetry system would enable the downlink of more relevant data for each phase or need of a nanosatellite mission. In addition, this functionality will provide telemetry history as well, which can greatly enhance the understanding of the performance of the nanosatellite. Based on predefined criteria for Delfi-n3Xt a mission success of approximately 85 % has been achieved. The main reason for this high success rate despite the satellite malfunctioning is that most experiments had already been conducted before the error occurred.

8.2 DelFFi CDHS requirements

The CDHS requirements for DelFFi have been continuously evaluated during the project. Some of these requirements were discarded because of changing interface definitions or conflicts with basic requirement rules. Other externally defined requirements, by for example the QB50 system requirements, sometimes needed clarification and this was managed in cooperation with the project manager of DelFFi. In total 24 requirements were proposed or requested for modification based on the thesis work. Reasons for modification of the requirements were changing interface definitions or several requirements which were listed in one requirement and were split to provide clarity. In addition, some requirements were discarded because they did not have a clear goal or rationale.

8.3 FLEXIBLE TELEMETRY

Different options to implement Flexible Telemetry (FlexTLM) were designed and evaluated. The designed FlexTLM solution will provide the DelFFi mission with the capability to efficiently use the data bandwidth available to downlink the relevant data of each particular operating mode. The telemetry assembly software onboard DelFFi will be able to flexibly include packets in AX.25 frames. Also, a telemetry history playback functionality is incorporated in the FlexTLM approach, to further enhance the knowledge and understanding of the entire DelFFi mission.

The research question related to flexible telemetry, which was stated in the introduction is:

> "Can a reliable flexible telemetry system increase the scientific return of the DelFFi mission and to what extent?"

Based on post analysis of the designed FlexTLM system it was concluded more relevant (science) data can be received by the DelFFi mission. Especially because the AX.25 frames can be redefined if the average access time per day decreases during the mission and thus more data requirements can be satisfied. Calculations have shown at least 64 % of the daily received data volume at the Delfi ground station can be saved by FlexTLM.

The next step will be the creation of the final engineering solution which can be used to develop the software implementation of FlexTLM. The eventual time and resources available to the DelFFi project will determine which functionalities of FlexTLM can be implemented in the final product. In case of the ground segment most software needs to be developed to interpret which packets are included in the received AX.25 frames. The satellite AX.25 frame assembly software needs to be developed and extensively tested as well. Also, software to interpret the different telecommands to control the FlexTLM has to be created.

8.4 wireless intra-spacecraft communication

Additional innovation has been applied by conducting research into wireless intra-spacecraft communication for nanosatellites. In a literature study by Schoemaker [47] an extensive trade-off was performed and it was already concluded Bluetooth Low Energy (BLE) is the most suitable wireless protocol for internal data communication in nanosatellites. Especially in terms of power consumed per transmitted bit BLE performs well.

On-ground experiments with BLE modules in a representative nanosatellite environment and the Delfi-C3 spare model have been conducted. The performance of these modules was very good and the observed packet error rates were low (<0.3 %). Note that packet errors do not actually represent lost data because acknowledgements are used to retransmit data. Low PERs can rather be seen as a small reduction in performance.

The research question concerning wireless intra-spacecraft communication, which was stated in the introduction is:

> "Does a wireless data link in a nanosatellite have benefits over traditional wired data links?"

After analysis of wired versus wireless nodes, it was concluded wireless solutions can be beneficial for certain difficult to integrate nodes. However, nodes with a high power consumption still have to be connected to the main power bus and hence the advantage of a wireless data bus for these nodes is reduced. Nevertheless, the excellent results of the wireless experiments has led to a proposal, to implement a completely independent wireless BLE temperature sensor experiment on DelFFi, in order to perform in-orbit validation of the technology.

It is recommended before the actual wireless experiment hardware is produced, electromagnetic compatibility (EMC) tests with some DelFFi flight hardware will be conducted. Especially the transmitter and GAMALINK device need to be tested for coexistence with the BLE hardware. In addition, the influence of the BLE signal on the performance of electronic circuits needs to be monitored.

Part III

APPENDIX

A PROJECT MANAGEMENT

In this appendix a discussion about how the thesis work was managed is performed. Initially, only a rough planning about what had to be done was listed. Once more was known about the delivery of hardware or the need of the DelFFi project, more detailed plannings were made by means of a Gantt chart. These Gantt charts were evaluated with the supervisor and updated every now and then. In appendix B the work breakdown structure of the entire thesis planning can be found.

In the thesis entrance permit form the start date of the thesis project was set at 11-11-2013 and the expected graduation was estimated to be at 05-06-2014. However, in this planning no small holidays and unplanned conferences were taken into account. Especially because the author was not aware yet about new regulations with regard to thesis planning and grading. Also, some work what was done could not always be effectively attributed to thesis work. For example, the involvement of the author in the ground station operations and mission analysis of Delfi-n3Xt took some time resources of the author. Although these ground station operations were a very valuable and learnful experience it was not possible to translate all of this work to the thesis.

Some hardware delivery delays shuffled the thesis schedule a bit as well. However, this was handled by shifting some tasks to the start of the thesis and therefore a waste of time was avoided.

Another task which took a considerable amount of time for the author was to get acquainted with the Bluetooth Low Energy hardware and software protocol. Nevertheless, a lot of development time was saved by making use of the in-house expertise of other department members (software engineer and electrical engineer).

After a summation of all small holidays, conferences and other activities a total of five weeks can be attributed to non-thesis time. If the 22^{nd} of July 2014 is set as the graduation date the nominal thesis time was exceeded by three weeks.

B THESIS PLANNING

WBS	Name	Start	Finish	Duration	Slack
1	Start of thesis	Nov 11	Nov 11	N/A	182d
2	**Preparatory phase**	**Nov 11**	**Jan 3**	**40d**	**142d**
2.1	Thesis definition	Nov 11	Nov 15	5d	177d
2.2	Delfi-n3Xt preparations and LEOPS	Nov 18	Dec 6	15d	162d
2.3	Delfi-n3xt operations chapter	Dec 2	Dec 12	9d	158d
2.4	DelFFi CDHS requirements analysis	Dec 2	Dec 12	9d	158d
2.5	Developed BLE experiments plan and hardware procurement	Dec 13	Dec 20	6d	152d
2.6	Christmas holidays	Dec 23	Jan 3	10d	142d
3	**General thesis work**	**Jan 6**	**Feb 14**	**30d**	**112d**
3.1	Read Bluegiga BLE documentation	Jan 6	Jan 10	5d	137d
3.2	Worked on FIPEX and GAMALINK analysis	Jan 13	Jan 24	10d	127d
3.3	Analyzed various DelFFi subsystem ICDs	Jan 27	Jan 31	5d	122d
3.4	Attended 7th QB50 symposium	Jan 28	Jan 29	2d	124d
3.5	Finished research methodologies course	Feb 3	Feb 7	5d	117d
3.6	Preparing code for BLE experiments	Feb 10	Feb 14	5d	112d
4	**BLE experiments**	**Feb 17**	**Apr 4**	**35d**	**77d**
4.1	Establish communication between BLED112 dongles	Feb 17	Feb 28	10d	102d
4.2	De Delftse Bedrijvendagen	Feb 18	Feb 19	2d	109d
4.3	Assemble and test BLE113 breakout boards	Feb 24	Feb 28	5d	102d
4.4	Interface BLE113 modules with Arduinos	Mar 3	Mar 7	5d	97d
4.5	Set up power measurement equipment	Mar 10	Mar 14	5d	92d
4.6	Prepare test programs for experiments	Mar 10	Mar 14	5d	92d
4.7	Perform experiments	Mar 17	Apr 4	15d	77d
5	**Write paper**	**Mar 31**	**Apr 18**	**15d**	**67d**
5.1	First draft paper	Mar 31	Apr 4	5d	77d
5.2	Small holiday	Apr 7	Apr 11	5d	72d
5.3	Second draft paper	Apr 7	Apr 11	5d	72d
5.4	Final version	Apr 14	Apr 18	5d	67d
6	4S paper submission deadline	Apr 21	Apr 21	N/A	67d
7	**Thesis report**	**Apr 21**	**May 16**	**20d**	**47d**
7.1	Incorporate BLE experiment results chapter	Apr 21	Apr 25	5d	62d
7.2	Wireless BLE experiment proposal	Apr 28	May 9	10d	52d
7.3	Work on flexible telemetry chapter	May 12	May 16	5d	47d
8	Prepare 4S Symposium presentation etc.	May 19	May 23	5d	42d
9	4S Symposium	May 26	May 30	5d	37d
10	**Finish thesis report**	**Jun 2**	**Jun 20**	**15d**	**22d**
10.1	Process 4S Symposium feedback	Jun 2	Jun 4	3d	34d
10.2	Finish flexible telemetry chapter and software ICD	Jun 5	Jun 18	10d	24d
10.3	Wrap up report	Jun 18	Jun 20	3d	22d
11	**Graduation process**	**Jun 23**	**Jul 22**	**22d**	
11.1	Greenlight review	Jun 23	Jun 23	N/A	22d
11.2	Thesis will be reviewed	Jun 23	Jun 27	5d	17d
11.3	Improve report based on feedback	Jun 30	Jul 4	5d	12d
11.4	Finish final version of report	Jul 7	Jul 7	N/A	12d
11.5	Prepare thesis presentation	Jul 9	Jul 22	10d	
11.6	Final preparations for graduation	Jul 16	Jul 21	4d	1d
11.7	Graduation	Jul 22	Jul 22	N/A	1d

C

DELFI-N3XT MISSION SUCCESS RATE

[Adapted, 74]

Objective		Success Criteria	Categorial Weight	Project Weight	Criteria Success Score	Project Success Score
1. Educational		Fullfil all educational success criteria.	100.0%	33.3%		33.3%
1.1	Education	Continue the Delfi-n3Xt project up-to launch and operations.	25.0%	8.3%	100%	8.3%
1.2		At least 2/3 of the project will be done by students.	10.0%	3.3%	100%	3.3%
1.3		At least 20 MSc theses resulting from Delfi-n3Xt.	30.0%	10.0%	100%	10.0%
1.4		Enable contact between students and project partners.	10.0%	3.3%	100%	3.3%
1.5		Find a suitable project topic for *every* MSc applicant.	10.0%	3.3%	100%	3.3%
1.6		Involve at least 10 Beng students (internships, theses).	15.0%	5.0%	100%	5.0%
2. Technology Demonstration		Fullfill a majority of the technology demonstration objectives.	125.0%	33.3%		25.5%
2.1.1	T3µPS (TNO)	Receive >25 frames with T3µPS data during and after a CGG ignition.	5.0%	1.7%	100%	1.7%
2.1.2		Receive >25 frames with T3µPS data during and after a thrust maneuver	5.0%	1.7%	100%	1.7%
2.1.3		Receive sufficient data from T3µPS of at least 5 CGG ignitions and subsequent thrust maneuvres to be able to analyze the performance of the system.	20.0%	6.7%	100%	6.7%
2.1.4		Proof a thrust maneuver through TLE change determination.	10.0%	3.3%	0%	0.0%
2.2.1	ITRX (ISIS)	Use the ITRX as primary transceiver for > 100 hours.	5.0%	1.7%	100%	1.7%
2.2.2		Use the ITRX as primary transceiver for 3 months.	20.0%	6.7%	50%	3.3%
2.2.3		Receive >200 frames of ITRX measurement data.	5.0%	1.7%	100%	1.7%
2.2.4	Transponder (TUD, ISIS, UTwente)	Test the transponder functionality and determine performance.	3.0%	1.0%	20%	0.2%
2.2.5		Successful transponder functionality of 3 months.	7.0%	2.3%	0%	0.0%
2.3.1	SDM (Dimes)	Receive > 200 frames of SDM data.	5.0%	1.7%	100%	1.7%
2.3.2		Receive daily frames with SDM data for 3 months i.c.w. sun sensor data.	15.0%	5.0%	100%	5.0%
2.4.1	EPS (Systematic)	Receive > 200 frames of EPS data.	5.0%	1.7%	100%	1.7%
2.4.2		Receive daily frames of EPS data for 3 months.	20.0%	6.7%	100%	6.7%
3. Platform Development & Research		Fullfil a majority of the platform development and testing objectives.	125.0%	33.3%		25.5%
3.1.1	ADCS	Complete development & AIT of triple axis active ADCS.	15.0%	5.0%	100%	5.0%
3.1.2		Receive & analyze at least 100 hours of ADCS telemetry.	10.0%	3.3%	100%	3.3%
		Demonstrate functioning of reaction wheels	3.0%	1.0%	100%	1.0%
		Demonstrate functioning of sun sensors	3.0%	1.0%	100%	1.0%
3.1.3		Demonstrate sun pointing mode.	3.0%	1.0%	50%	0.5%
3.1.4		Demonstrate thruster pointing mode.	3.0%	1.0%	0%	0.0%
3.1.5		Demonstrate ground station tracking.	3.0%	1.0%	100%	1.0%
3.2.1	Reliability	Complete design of a reliable satellite bus system, taking lessons learned of Delfi-C^3 into account.	15.0%	5.0%	100%	5.0%
3.2.2		Establish a 90% uptime of the transmitter during the sunlit period of operations for three months.	15.0%	5.0%	100%	5.0%
		Develop and establish a satellite and ground segment which processes data with proper time tagging and a directly usable data format for operations and analysis.	10.0%	3.3%	100%	3.3%
3.2.3		Establish two years of operational life time of the satellite.	20.0%	6.7%	0%	0.0%
3.3.1	S-band transmitter	Develop a high speed data downlink of >9.6 kbit/s.	5.0%	1.7%	100%	1.7%
3.3.2		Receive > 1000 frames through the high speed data link.	5.0%	1.7%	0%	0.0%
3.4	Structure	Develop a new structure with focus on integration and verify integrity in vibration test.	10.0%	3.3%	100%	3.3%
3.5	Thermal research	Receive at least three months of thermal sensor data from 90% of the thermal onboard sensors.	5.0%	1.7%	100%	1.7%
			Total Mission Succes Rate:			**84.3%**

[1] Giorgos Galatis. CATIA drawings DelFFi, 2014.

[2] Alfred North Whitehead. British mathematician and Philosopher. (Cited on page ix.)

[3] TU Delft. Artist impression of Delfi-n3Xt in space., 2013. URL http://www.lr.tudelft.nl/uploads/RTEmagicC_Delfi-n3Xt_poster_2011_5_m.jpg.jpg. (Cited on pages xiii and 5.)

[4] Steve Snowden and (US ROSAT Science Data Center). South Atlantic Anomaly, 1990. URL http://heasarc.gsfc.nasa.gov/docs/rosat/gallery/display/saa.html. (Cited on pages xiii and 8.)

[5] Pedro Rodrigues. GAMANET - Interface Control Document (v2.0). Technical Report 2, 2013. (Cited on pages xiii, 19, 20, and 21.)

[6] Global Microwave. Equitorial orbit. URL http://globalmicrowave.org/content/equitorial_orbit_geo_thumb.jpg. (Cited on pages xiii and 48.)

[7] Delfispace and TNO. Delfi-C3 AWSS Payload (Front and Back View). URL http://www.delfispace.nl/images/Delfi-C3/AWSS/Delfi-C3AWSSPayload(FrontandBackView).jpg. (Cited on pages xiii and 66.)

[8] CALSENS. OPTOS, 2013. URL http://cal-sens.com/?p=2347&lang=en. (Cited on pages xiii and 67.)

[9] Arduino. Arduino Due, 2013. URL http://www.digibay.in/image/cache/data/se/101-a-arduino-due-atmel-sam3x8e-arm-cortex-m3-84mhz-600x600.jpg. (Cited on pages xiii and 70.)

[10] Bluegiga. BLE113 - Preliminary data sheet, 2013. URL http://datasheets.ru/docs/NEC/001199.pdf. (Cited on pages xiii and 75.)

[11] Bluegiga. BLE113 - data sheet (v1.4), 2014. (Cited on pages xiii, 69, 73, 78, 79, 101, and 103.)

[12] Mouser. EDLC252520-351-2F-21, . URL http://nl.mouser.com/Search/include/LargeProductImage.aspx?path=tdk/lrg/EDLC1.jpg&mfrName=TDK&mfrPartNum=EDLC252520-351-2F-21. (Cited on pages xiv and 96.)

[13] Murata. DMT334R2S474M3DTA0, 2013. URL http://www.murata.com/new/news_release/2013/0909/images/ind_img01.jpg. (Cited on pages xiv and 96.)

[14] Kentucky Space. KySat-1. URL https://eoportal.org/image/image_gallery?uuid=c804341c-3e84-43bd-849f-41d3560ebc4e&groupId=163813&t=1339001533377. (Cited on pages xiv and 100.)

[15] WA Beech, DE Nielsen, and J Taylor. AX.25 Link Access Protocol for Amateur Packet Radio. Technical Report July, 1998. URL http://www.tapr.org/pdf/AX25.2.2.pdf. (Cited on pages xiv, 39, and 55.)

[16] Jian Guo. QB50 Critical Design Review Data Package Document 1 - CubeSat Design Overview Report. Technical report, 2014. (Cited on pages xv and 54.)

[17] Jian Guo. DFF-TUD-BU-1113 [4.0] DelFFi Power Budget, 2014. (Cited on pages xv and 54.)

[18] SA Asundi and NG Fitz-Coy. Design of command, data and telemetry handling system for a distributed computing architecture CubeSat. *Aerospace Conference, 2013 IEEE*, 2013. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6496901. (Cited on page 2.)

[19] Sander van den Berg. MSc THESIS Fault-Tolerant On-Board Computer Software for the Delfi-n3Xt Nanosatellite. 2012. URL http://ce-publications.et.tudelft.nl/publications/1301_faulttolerant_onboard_computer_software_for_the_delfin3.pdf. (Cited on pages 9 and 88.)

[20] J. Bouwmeester, L. Rotthier, C. Schuurbiers, W. Wieling, G. van der Horn, F. Stelwagen, E. Timmer, and M. Tijssen. PRELIMINARY RESULTS OF THE DELFI-N3XT MISSION. *4S Symposium*, pages 1–15, 2014. (Cited on page 9.)

[21] Jasper Bouwmeester. TRXVU feature requests, 2014. (Cited on page 15.)

[22] Jian Guo. DFF-TUD-SE-1110 [3.1] DelFFi Requirements & Configuration Item List, 2014. (Cited on pages 16, 22, 23, 24, 25, 27, 28, 29, 30, 31, and 32.)

[23] Fiona Singarayar. QB50 System Requirements and Recommendations (issue 5). Technical Report 5, 2013. URL https://www.qb50.eu/images/sampledata/reqicds/QB50_Systems_Requirements_20131011_Annex.pdf. (Cited on pages 19, 23, 24, 31, 36, 38, 39, 51, 94, 101, and 103.)

[24] Paul Roß mann, Jörg Heisig, and Tino Schmiel. QB50 FIPEX Science Unit Interface Control Document (v2.0.10). Technical report, 2014. (Cited on pages 19, 22, 23, 25, 32, 37, and 51.)

[25] Tekever. 7th QB50 workshop, von Karman Institute (Brussels), 2014. (Cited on pages 20 and 21.)

[26] ISIS. TrxVU Interface Control Document. page 2013, 2013. (Cited on page 24.)

[27] Maher A Sallam. Analysis and design of a reliable, fault-tolerant storage system for DelFFis nano-satellites. (June), 2014. (Cited on pages 26, 47, and 58.)

[28] Jian Guo. DFF-TUD-SE-0167 [2.2] DelFFi Requirements & Configuration Item List, 2013. (Cited on pages 26 and 27.)

[29] Fiona Singarayar. QB50 System Requirements and Recommendations (issue 3), 2013. (Cited on pages 26 and 27.)

[30] Fiona Singarayar. QB50 System Requirements and Recommendations (issue 4). Technical Report 4, 2013. (Cited on pages 26 and 27.)

[31] A Weber, J. Heisig, and T. Schmiel. QB50 FIPEX Science Unit Interface Control Document (v1.4.77). Technical report, 2013. (Cited on pages 30 and 31.)

[32] A Weber, J. Heisig, and T. Schmiel. QB50 FIPEX Science Unit Interface Control Document (v1.5.3). Technical report, 2013. (Cited on pages 30, 31, and 32.)

[33] Jian Guo. DFF-TUD-BU-1114 [3.0] DelFFi Link Budget, 2013. (Cited on page 36.)

[34] Remco Schoemaker. DFF-TUD-IC-1147 [1.8] CDHS Software ICD, 2014. (Cited on pages 37, 52, 56, and 58.)

[35] Martijn de Milliano, Arthur Tindemans, and Lieuwe S. Boersma. DNX-TUD-TN-0014 [4.7] COMMS - Top Level Design of Communication System. Technical report, 2012. (Cited on page 39.)

[36] Stensat. FX.25 On-Air Performance. Accessed: 21-07-2014. URL `http://www.stensat.org/projects/FX-25/FX-25_performance.htm`. (Cited on page 39.)

[37] Anatoly Ilin. Literature study: Telemetry Systems. 2014. (Cited on page 57.)

[38] Remco Schoemaker. Evaluation of Bluetooth Low Energy Wireless Internal Data Communication for Nanosatellites. 4S Symposium, 2014. (Cited on pages 65 and 81.)

[39] Rouzbeh Amini, Eberhard Gill, and Georgi Gaydadjiev. The challenges of intra-spacecraft wireless data interfacing. *57th International Astronautical Congress*, pages 1–6, 2007. URL `http://www.sse.lr.tudelft.nl/fileadmin/Faculteit/LR/Organisatie/Afdelingen_en_Leerstoelen/Afdeling_SpE/Space_Systems_Eng./Publications/2007/doc/IAC-07-B4708_Amini.pdf`. (Cited on page 65.)

[40] W.H. Zheng and J.T. Armstrong. Wireless intra-spacecraft communication: The benefits and the challenges. *NASA/ESA Conference on Adaptive Hardware and Systems Wireless*, pages 75–78, June 2010. doi: 10.1109/AHS.2010.5546219. URL http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5546219&isnumber=5546215. (Cited on page 65.)

[41] Mark J Bentum, Johan Leijtens, Chris Verhoeven, Hans Van Der Marel, and Bradford Engineering. Measurements on an autonomous wireless payload at 635 km distance using a sensitive radio telescope. *33rd ESA Antenna Workshop on Challenges for Space Antenna Systems 2011*, (4):1–5, 2011. URL http://doc.utwente.nl/78985/1/2304053Bentum.pdf. (Cited on page 66.)

[42] I Arruego and H Guerrero. OWLS: A ten-year history in optical wireless links for intra-satellite communications. *IEEE Journal on Selected Areas in Communications*, 27(9):1599–1611, 2009. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5342320. (Cited on page 67.)

[43] Bryan Klofas. CubeSat Communications System Table. Accessed: 03-04-2014, 2014. URL http://www.klofas.com/comm-table/table.pdf. (Cited on page 67.)

[44] EoPortal. OPTOS (Optical Nanosatellite). Accessed: 03-04-2014, 2014. URL https://directory.eoportal.org/web/eoportal/satellite-missions/o/optos. (Cited on page 67.)

[45] Jordi Frances Matas. Internal Wireless Bus for a CubeSat. (July), 2013. URL http://www.diva-portal.org/smash/record.jsf?pid=diva2:656728. (Cited on page 67.)

[46] M Beekema. Fault-Tolerant Platform for Intra-Spacecraft Modular Wireless Sensor Network. Technical report, 2011. URL http://repository.tudelft.nl/view/ir/uuid:ec871698-8a96-4836-ad74-da71c9231d29/. (Cited on page 68.)

[47] Remco Schoemaker. Robust and Flexible Command & Data Handling on board the DelFFi Formation Flying Mission: Evaluation of Internal Wireless Connection Protocols for Future Nanosatellites. Technical report, 2013. (Cited on pages 68 and 107.)

[48] Jeff Rowberg. [REFERENCE]: BLE11x low power and sleep modes. Accessed: 15-04-2014, 2013. URL https://bluegiga.zendesk.com/entries/23173106--REFERENCE-BLE11x-low-power-and-sleep-modes. (Cited on page 69.)

[49] Jeff Rowberg. [HOW-TO]: Maximize throughput with the BLE112/BLED112. Accessed: 15-03-2014, 2012. URL https://bluegiga.zendesk.com/entries/22400867--HOW-TO-Maximize-throughput-with-the-BLE112-BLED112. (Cited on pages 69 and 88.)

[50] Mikko Savolainen. Throughput with Bluetooth Smart technology. Accessed: 30-03-2014, 2013. URL https://bluegiga.zendesk.com/entries/24646818-Throughput-with-Bluetooth-Smart-technology. (Cited on pages 69 and 88.)

[51] Agilent. How do I measure Bit Error Rate (BER) to a given confidence level on N490xA/B Serial BERTs? Accessed: 19-03-2014. URL http://www.home.agilent.com/agilent/editorial.jspx?cc=US&lc=eng&ckey=1481106&nid=-11143.0.00&id=1481106. (Cited on page 71.)

[52] Jeff Rowberg. [REFERENCE]: BLE master/slave, GATT client/server, and data RX/TX basics. Accessed: 15-03-2014, 2013. URL https://bluegiga.zendesk.com/entries/25053373--REFERENCE-BLE-master-slave-GATT-client-server-and-data-RX-TX-basics. (Cited on page 72.)

[53] Bluegiga. Bluetooth smart module - Configuration guide (v3.1), 2013. (Cited on pages 73 and 76.)

[54] Bluegiga. Bluegiga bluetooth smart software - API documentation (v1.2), 2014. (Cited on page 75.)

[55] CCSDS. Wireless Network Communications Overview for Space Mission Operations. Technical Report December, 2010. URL http://public.ccsds.org/publications/archive/880x0g1.pdf. (Cited on pages 85 and 86.)

[56] NXP. PCA9515 I2C-bus repeater. Accesse, 2009. URL http://www.nxp.com/documents/data_sheet/PCA9515.pdf. (Cited on page 87.)

[57] Sparkfun. Coin Cell Battery - 20mm (CR2032). Accessed: 14-04-2014. URL https://www.sparkfun.com/products/338. (Cited on pages 87 and 100.)

[58] Bluegiga. BLE113 Bluetooth Smart Module, 2014. (Cited on page 92.)

[59] Pierre Mars. Using a Small Solar Cell and a Supercapacitor in a Wireless Sensor. Accessed: 08-05-2014, 2010. URL http://www.sensorsmag.com/networking-communications/energy-harvesting/using-a-small-solar-cell-and-a-supercapacitor-a-wireless-sen-7310. (Cited on page 94.)

[60] Semec. Tantalum Capacitors. Accessed: 08-05-2014. URL http://www.semectech.com/Tantalum-Capacitors/. (Cited on page 95.)

[61] Vishay. SuperTan Extended (STE) Capacitors, Wet Tantalum Capacitors with Hermetic Seal. 2013. URL http://www.vishay.com/docs/43009/ste.pdf. (Cited on page 96.)

[62] Mouser. EDLC252520-351-2F-21. Accessed: 08-05-2014, . URL http://nl.mouser.com/ProductDetail/TDK/EDLC252520-351-2F-21/?qs=sGAEpiMZZMsCu9HefNWqphZcOC/uFu2ZmdCtgvekVTg=. (Cited on pages 96 and 102.)

[63] Mouser. DMT334R2S474M3DTA0. Accessed: 08-05-2014, . URL http://nl.mouser.com/ProductDetail/Murata-Electronics/DMT334R2S474M3DTA0/?qs=sGAEpiMZZMsCu9HefNWqpuD%2bL0qKryH3cC7v5j9lLEp%2bxj5GbDn3Vg==. (Cited on page 96.)

[64] Multicomp. Lithium-ion Battery. URL mwww.adafruit.com/datasheets/LIR2450.pdf. (Cited on page 97.)

[65] Peter Fortescue, John Stark, and Graham Swinerd. *Space System Engineering*. 2002. (Cited on page 99.)

[66] M Raja Reddy. Space solar cells - tradeoff analysis. 77(July 2002): 175–208, 2003. URL http://www.sciencedirect.com/science/article/pii/S0927024802003203#. (Cited on page 99.)

[67] Spectrolab. Triangular Advanced Solar Cells (TASC), 2002. URL http://www.spectrolab.com/DataSheets/PV/PV_NM_TASC_ITJ.pdf. (Cited on pages 99 and 100.)

[68] Paul M.C. Beckers. DNX_TUD_ED_0999 TCS Temperature sensor. 2012. (Cited on page 100.)

[69] Mouser. Bluegiga Technologies BLE113-A-v1. 23-04-2014, . URL http://nl.mouser.com/ProductDetail/Bluegiga-Technologies/BLE113-A-v1/?qs=sGAEpiMZZMsGelYiB%2bjhZpS9PJgNT5YFf7JrVEBvX8iq4UNDNpHamQ==. (Cited on page 102.)

[70] Mouser. TMP275AIDG4. Accessed: 09-05-2014, . URL http://nl.mouser.com/ProductDetail/Texas-Instruments/TMP275AIDG4/?qs=sGAEpiMZZMucenltShoSnvEsemSI3a5kkdnDjBuqz4o=. (Cited on page 102.)

[71] Kevin West. TASC Spectrolab, 2013. (Cited on page 102.)

[72] Mouser. Texas Instruments CC-DEBUGGER. Accessed: 23-04-2014, . URL http://nl.mouser.com/ProductDetail/Texas-Instruments/CC-DEBUGGER/?qs=03wcJz4o8Id/Ma87rj4LcA==. (Cited on page 102.)

[73] Digi-Key. CR2032. Accessed: 23-04-2014. URL http://www.digikey.nl/product-detail/en/CR2032/P189-ND/31939. (Cited on page 102.)

[74] Delfi. Delfi-n3Xt Mission Succes Input Sheet, 2013. (Cited on page 115.)

colophon

This document was typeset using the typographical look-and-feel classicthesis developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography *"The Elements of Typographic Style"*. classicthesis is available for both LaTeX and LyX:

http://code.google.com/p/classicthesis/

Happy users of classicthesis usually send a real postcard to the author, a collection of postcards received so far is featured at:

http://postcards.miede.de/

Final Version as of July 7, 2014 (classicthesis version 1.0 - final version).