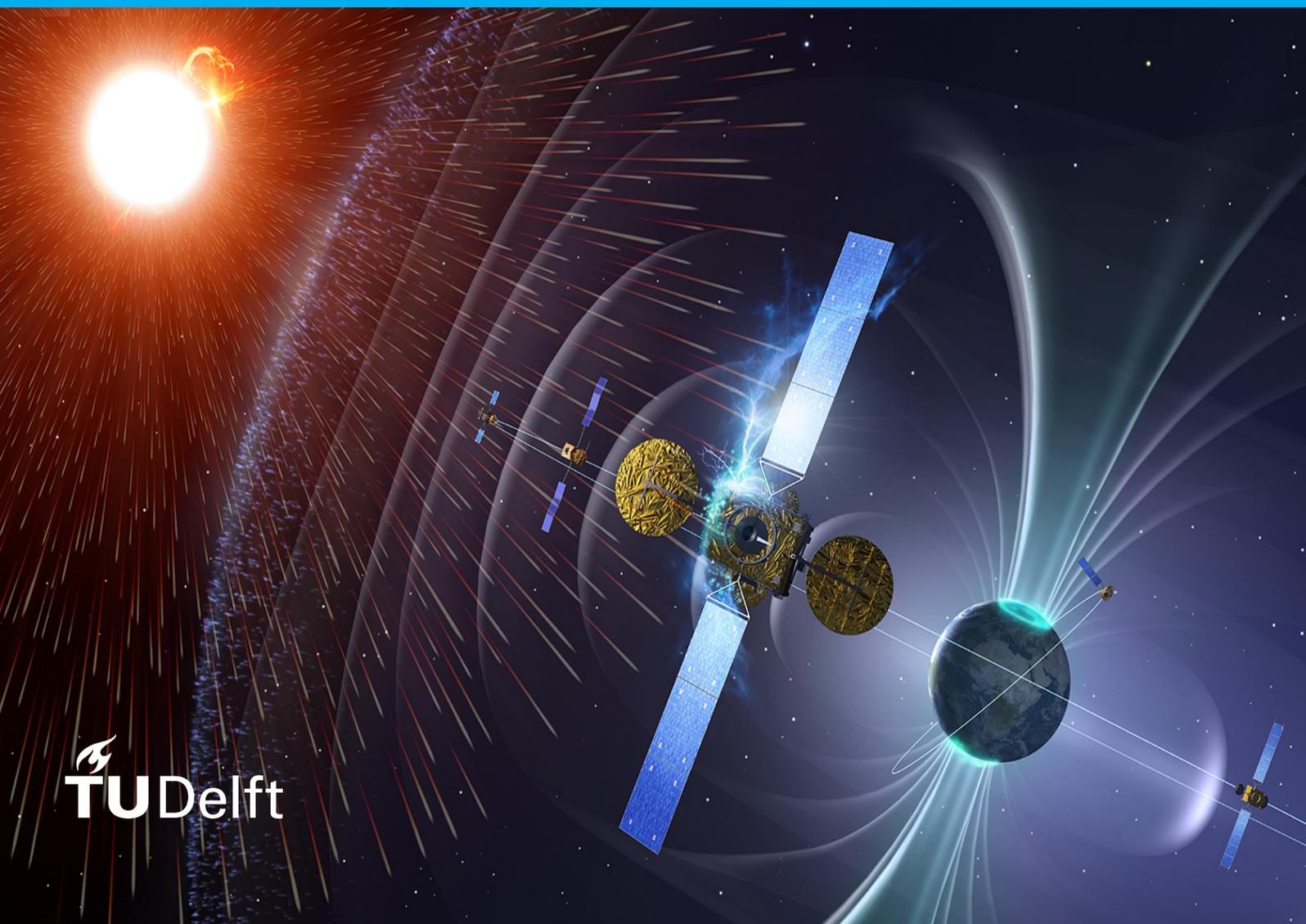


# Characterisation of a Floating Gate Dosimeter for Space Applications

William De Meyere





# Characterisation of a Floating Gate Dosimeter for Space Applications

by

William De Meyere

to obtain the degree of Master of Science  
at the Delft University of Technology,  
to be defended publicly on Thursday March 17, 2022 at 09:00 AM.

*A little learning is a dangerous thing;  
Drink deep, or taste not the Pierian spring:  
Alexander Pope*

Student number: 5137861  
Project duration: May 25, 2021 – March 17, 2022  
Thesis committee: Prof. dr. E.K.A. Gill, TU Delft, Chair  
Dr. A. Menicucci, TU Delft, Supervisor  
Dr. D.M. Stam, TU Delft, Examiner

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Front cover image: European Space Agency



# Summary

Solid state dosimetry is nowadays crucial in providing high-quality real-time measurement of radiation dose. In collaboration with the European Organization for Nuclear Research (CERN), the Málaga based company Sealicon has developed a sensor chip for solid state dosimetry based upon transistor technology. The advantages of Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET) dosimeters include compact size, good resolution and large measurement range. The disadvantage of temperature dependence can be overcome by using a second reference transistor. The Sealicon developed chip differs from most other solid state dosimeters in that it uses a floating gate transistor. The floating gate is isolated from the rest of the transistor and is extended over the field oxide. A charge can then be injected into this isolated gate. When ionising radiation hits the sensor, it creates electron-hole pairs in the field oxide and, depending on the sign of the charge in the floating gate, either the holes or the electrons migrate towards the gate. These charges partially neutralise the charge in the gate and hence influence the transistor's drain current. By measuring this current, the amount of ionising radiation can be determined. Advantages of floating gate dosimeters are their low supply voltage requirements, the possibility to integrate read-out electronics and the fact that they can be manufactured using conventional processes. Furthermore, gate bias voltage and continuous power supply are not required.

Although a wide array of radiation sensors exists, new compact and precise sensor types are needed by the space community. Classical space missions and radiation sensor systems are bulky and expensive, limiting data collection. Additional data with good spatial and temporal coverage are necessary to improve engineering design models, augment forecasting methods, track environmental changes and perform scientific research. A cheap and compact radiation monitoring payload will allow the design of miniature space missions, making collection of data at a relatively low cost possible. This payload can also be integrated in larger satellites, deep space missions or constellations. If the sensor is simple enough, only a minimal modification of the existing satellite designs is needed for this integration. The main objective of this thesis was thus to design and characterise a simple, compact, cheap but yet precise radiation monitoring system.

In the first phase of this study, an interface between the sensor chip developed by Sealicon and a CubeSat testbed was designed by using the FlatSat at the European Space Research and Technology Centre (ESTEC). This testbed is a flattened out CubeSat, and includes on-board computers, a telemetry unit, a power distribution unit and attitude control systems. To verify functionality of the interface, the on-board computer of the testbed was programmed to gather dosimeter data at regular intervals and send it, via the telemetry unit, to the ground station. On the ground station, mission control software was used to consult the data being transferred in real-time. In the second phase, the performance and characteristics of the sensor were verified by means of proton beam radiation tests. Although the sensor chip was able to consistently detect radiation, the sensitivity of the sensor was lower than expected. Additional tests were performed and the results hereof indicate that the low sensitivity might be caused by low beam energies. The sensor also showed a degradation of its sensitivity in function of total ionising dose. Annealing effects were noted, though these might be less of an issue considering the relatively low dose rates encountered in space. The electrical noise in the circuit was determined as well. Compared to previous research data, the noise was found to be more than three times as high. This noise can be reduced by means of filtering. Future research is needed to detect possible causes of the high electrical noise and investigate solutions. In conclusion, it was shown that it is possible to integrate the sensor chip into a satellite system. Additionally, proton beam tests have demonstrated that the floating gate dosimeter can detect radiation. Future research is needed to verify the performance of the radiation sensor, especially at lower particle energies.



# Preface

During my master's course *Space Embedded Systems*, I came into contact with the space radiation environment and the design of embedded systems. Fortunately, the course lecturer Prof. Dr. Alessandra Menicucci, now my thesis promotor, had good ideas for a thesis subject. I would first like to thank Alessandra for the continued support throughout the whole project: setting up collaborations with the European Space Agency (ESA) and the European Organization for Nuclear Research (CERN), arranging radiation tests and, of course, providing constructive feedback on my work.

Next, I would also like to thank all the engineers and researchers I had the opportunity to work with. My fellow student Zan Kupljenik was working on the same test setup at ESA, and working together was an amazing experience. Tomasz Szewczyk and Malte Bargholz, the engineers at ESA, have been there everyday to help us with both hardware and software issues. Abhimanyu Shanbhag, researcher at TU Delft, has helped me greatly with dosimeter research. The meetings we had with the researchers at CERN were always enlightening. Dr. Salvatore Danzeca, Dr. Matteo Brucoli and Alessandro Zimmaro were a great help in understanding the sensor and providing feedback on the radiation test results. I would also like to thank Lukasz Pakula, engineer at the Mechanical, Maritime and Materials Engineering department of TU Delft, for the assistance in PCB design and Wouter van Burik, researcher at HollandPTC, for the extensive assistance before, during and after the radiation tests.

Finally, I would like to thank my friends and family. To my parents, who have always been an example to me and taught me so much, thank you for offering me this amazing opportunity and for the unconditional support. To my brothers, without whom I would not be the person I am today, thank you for being the rocks I can build upon. To my friends, thank you for the never-ending supply of joie de vivre. Thank you Julie, for all your love and encouragement.

*William De Meyere  
Delft, March 2022*



# Contents

<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xiii</b>
<b>Acronyms</b>	<b>xv</b>
<b>Symbols</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Literature Study</b>	<b>3</b>
2.1 CubeSats . . . . .	3
2.1.1 CubeSat Reliability . . . . .	4
2.2 Radiation . . . . .	4
2.2.1 Space Radiation Environment . . . . .	5
2.2.2 Radiation Effects on Electronic Equipment . . . . .	6
2.3 Solid-State Detectors . . . . .	10
2.3.1 Metal-Oxide-Semiconductor Field-Effect Transistor Dosimeters . . . . .	11
2.3.2 Floating Gate Dosimeters . . . . .	12
2.4 Smart & Distributed Radiation Monitoring . . . . .	16
2.4.1 Machine Learning in Space . . . . .	17
2.4.2 Satellite Constellations . . . . .	18
2.5 Research Goals & Question . . . . .	18
<b>3 Floating Gate Dosimeter Payload Design</b>	<b>19</b>
3.1 System Design . . . . .	19
3.1.1 Stakeholder Requirements . . . . .	19
3.1.2 Verification & Validation . . . . .	21
3.2 Radiation Environment Simulation . . . . .	23
3.2.1 Low-Earth Orbit . . . . .	23
3.2.2 Mars . . . . .	27
3.2.3 Conclusion . . . . .	30
3.3 Thermal Limitations . . . . .	31
3.3.1 Heat Balance . . . . .	31
3.3.2 Conclusion . . . . .	31
3.4 FlatSat . . . . .	32
3.4.1 FlatSat Software . . . . .	33
3.4.2 FlatSat & Arduino . . . . .	35
3.4.3 Yet Another Mission Control System . . . . .	36
3.5 Floating Gate Dosimeter . . . . .	36
3.5.1 FGD-03F Setup . . . . .	36
3.5.2 Post-Processing . . . . .	37
3.6 Arduino . . . . .	37
3.6.1 Software Setup . . . . .	38
3.6.2 Software Loop . . . . .	39
3.7 From Dosimeter to Mission Control . . . . .	39
<b>4 Radiation Testing</b>	<b>41</b>
4.1 Test Preparation . . . . .	41
4.1.1 Test Facility . . . . .	41
4.1.2 Printed Circuit Board Design . . . . .	42
4.1.3 Test Setup . . . . .	44

4.1.4	Arduino & SAMV71 Programming . . . . .	45
4.1.5	Data Acquisition . . . . .	46
4.1.6	Test Plan . . . . .	46
4.2	Results . . . . .	47
4.2.1	Sensor Sensitivity . . . . .	47
4.2.2	Flux & Total Ionising Dose Effects . . . . .	49
4.2.3	Linear Range . . . . .	53
4.2.4	Passive Mode & Power Consumption . . . . .	53
4.2.5	Recharging Process . . . . .	54
4.2.6	Annealing & Charge Retention. . . . .	55
4.2.7	Resolution & Temperature . . . . .	56
4.3	Supplementary Tests . . . . .	63
<b>5</b>	<b>Conclusion</b>	<b>67</b>
5.1	Floating Gate Dosimeter Payload Design . . . . .	67
5.2	Radiation Testing . . . . .	68
5.3	Research Goals & Question . . . . .	69
5.4	Future Work. . . . .	69
<b>A</b>	<b>SPENVIS Simulations</b>	<b>71</b>
A.1	Low-Earth Orbit. . . . .	71
A.2	Areostationary orbit. . . . .	74
<b>B</b>	<b>Code</b>	<b>75</b>
B.1	FlatSat . . . . .	75
B.1.1	FreeRTOS . . . . .	75
B.1.2	Yamcs. . . . .	75
B.1.3	On-Board Computer . . . . .	78
B.2	Arduino . . . . .	79
B.2.1	Basic Functions. . . . .	79
B.2.2	FlatSat-Dosimeter Interface . . . . .	80
B.2.3	Radiation Tests . . . . .	83
B.3	SAMV71 . . . . .	86
B.3.1	Radiation Tests . . . . .	86
	<b>Bibliography</b>	<b>89</b>

# List of Figures

2.1	Amount of satellites launched per year according to an online database [97]. . . . .	4
2.2	Proton and electron intensities in the Van Allen belts according to the AP-8 and AE-8 models [63]. . . . .	6
2.3	Satellite anomalies caused by the space environment [9, 32, 33]. . . . .	7
2.4	TID effect on a MOS transistor [79]. . . . .	9
2.5	Annual dose received with 4 mm aluminium shielding in function of altitude for an equatorial circular orbit [25]. . . . .	9
2.6	A telescope used for particle identification consisting of a $\Delta E$ and an $E$ detector [62]. . .	11
2.7	Illustration of a pixel in a hybrid pixel detector [73]. . . . .	11
2.8	Cross section of the gate area of a MOSFET dosimeter. The acronyms ot and it stand for oxide traps and interface traps [52]. . . . .	12
2.9	Cross-section of a floating gate transistor [81]. . . . .	13
2.10	Cross-section and top view of a floating gate dosimeter developed by Sealicon [43]. . .	14
2.11	A schematic representation of the chip developed at Sealicon [89]. . . . .	15
2.12	Comparison of the dose responses of the Varadis VT03 1 m RADFET [102], and a Sealicon prototype FGDOS sensor chip [12]. . . . .	16
2.13	TID measured by RADFET sensors, with and without bias gate voltage, and an FGDOS sensor. The sensors were placed in a mixed field generated by a 24 GeV proton beam and a copper target [13]. . . . .	17
3.1	Typical SPENVIS workflow [30]. . . . .	23
3.2	A world map with mission averaged proton fluxes. Output retrieved from SPENVIS for a one year mission in an ISS-like orbit. . . . .	24
3.3	Results of the SPENVIS simulation for solar proton fluence and GCR proton flux for a one year CubeSat mission in an ISS-like orbit. . . . .	25
3.4	TID in Si in function of shielding depth according to a SPENVIS simulation for a one year CubeSat mission in an ISS-like orbit. . . . .	26
3.5	Comparison of the total dose for a one year CubeSat mission on an ISS like orbit using different models. The H in the legend represents hydrogen ions, i.e. protons. The terms cubic, powerlaw and linear in the legend represent the chosen interpolation method. . .	27
3.6	Flux and dose rates measured by the Liulin-MO dosimeter during a circular orbit around Mars with an altitude of 400 km and an inclination of 72° [90]. . . . .	28
3.7	Dose rate measured by MARIE during an SPE that took place in July 2002 [107]. . . . .	29
3.8	Comparison between the GCR $^1\text{H}$ isotope fluxes for a 10 year Mars orbiter mission during either the 1996 solar minimum or the current mission epoch solar maximum. . . . .	29
3.9	TID in function of shielding depth according to SPENVIS simulations for a 10 year mission in an areostationary orbit using both SHIELDDOSE and MULASSIS. . . . .	30
3.10	Picture of the FlatSat with basic avionics units installed and the ground station, all provided by Space Inventor. . . . .	33
3.11	Schematic of the Arduino UNO microcontroller [4]. . . . .	35
3.12	Life output of Yamcs showing the dosimeter frequency in function of time. Data go from the dosimeter to the ground station via the Arduino and the FlatSat CSP network. . . . .	40
4.1	Beam arrangement and test setup positioning for the radiation tests at HollandPTC on the 16 <sup>th</sup> of December. . . . .	42
4.2	Construction of the FGD-03F PCB in EasyEDA. . . . .	43
4.3	Test setup for the radiation tests at HollandPTC. . . . .	44
4.4	Schematic representation of the setup for the radiation tests at HollandPTC. . . . .	45

4.5	Sensor frequencies in function of time with a beam energy of $70\text{ MeV}$ at DUT, a flux of $7.07e6\text{ \#/cm}^2/\text{s}$ and the sensors in high and low sensitivity mode. Linear regression lines are added, as well as the sensitivity calculated from the linear regression coefficients and dose rates. These plots correspond with test runs 03 and 04 respectively. . . . .	48
4.6	Sensor temperature and reference frequencies in function of time with a beam energy of $70\text{ MeV}$ at DUT, a flux of $7.07e6\text{ \#/cm}^2/\text{s}$ and the sensors in high and low sensitivity mode. These plots correspond with test runs 03 and 04. . . . .	50
4.7	Sensor frequency in function of time with a beam energy of $70\text{ MeV}$ at DUT, a flux of $3.69e7\text{ \#/cm}^2/\text{s}$ and the sensor in high sensitivity mode. Linear regression lines are added, as well as the sensitivity calculated from the linear regression coefficients and dose rates. The plot corresponds with test run 07. . . . .	51
4.8	Threshold voltage shift and recovery of an irradiated N-channel MOSFET [79]. . . . .	52
4.9	Typical progress of sensitivity and frequency output of an FGDOS in function of received dose [15]. . . . .	52
4.10	Sensor and reference frequencies in function of time with a beam energy of $70\text{ MeV}$ at DUT, a flux of $7.07e6\text{ \#/cm}^2/\text{s}$ and the sensors in high sensitivity mode. The plot corresponds with test run 09. . . . .	54
4.11	Power consumption of the sensors in four different steps, each lasting about five minutes: high sensitivity mode, low sensitivity mode, standby mode and passive mode. Since passive mode was only available for sensor 2, sensor 1 was put back into high sensitivity mode during the last step. . . . .	54
4.12	Sensor frequency overshoot in function of time with a beam energy of $70\text{ MeV}$ at DUT, a flux of $1.40e7\text{ \#/cm}^2/\text{s}$ and the sensor in high sensitivity mode. The plot corresponds with test run 05. . . . .	55
4.13	Electric field generated by the floating gate charge, left, and the effect on the electric field caused by charge traps in the field oxide [14]. . . . .	56
4.14	Sensor output in function of time for a noise test in high sensitivity mode using a window of 4096 pulses. The figures correspond with test 01 in table 4.5. . . . .	58
4.15	Histograms of the sensor frequency outputs for a noise test in high sensitivity mode. A normal distribution is fitted on top of the histograms. The figures correspond with test 01 in table 4.5. . . . .	59
4.16	Sensor frequency output in function of time and filtering of the signal using a moving average with a window of $20\text{ s}$ for a noise test in high sensitivity mode. The figures correspond with test 01 in table 4.5. . . . .	60
4.17	Histograms of the sensor frequency outputs for a noise test in low sensitivity mode. A non-parametric distribution, determined via a Gaussian KDE, is fitted on top of the histograms. The figures correspond with test 02 in table 4.5. . . . .	61
4.18	Sensor output in function of time for a noise test in high sensitivity mode using a window of 32768 pulses. The figures correspond with test 03 and 04 in table 4.5. . . . .	62
4.19	Sensor output in function of time without radiation but with varying temperature and in low sensitivity mode. . . . .	64
4.20	Arduino $5\text{ V}$ power supply output measured during four different phases, each lasting about $2\text{ min}$ : external power supply without sensor interaction, external power supply with sensor interaction, USB cable power supply with sensor interaction, USB cable power supply without sensor interaction. . . . .	65
4.21	Sensor frequencies in function of time for an experiment performed on the $24^{\text{th}}$ December with a beam energy of $150\text{ MeV}$ and a dose rate at DUT in air of $0.5\text{ Gy/min}$ . . . . .	65
4.22	Sensor frequencies in function of time for an experiment performed on the $24^{\text{th}}$ December with a beam energy of $175\text{ MeV}$ and a dose rate at DUT in air of $1.84\text{ Gy/min}$ . Linear regression lines and associated sensitivities are depicted as well. . . . .	66
A.1	Orbit specifications for a one year CubeSat mission in LEO in SPENVIS based on the orbital characteristics of the ISS [82]. . . . .	71
A.2	Settings for the trapped radiation models in SPENVIS for a one year CubeSat mission in LEO. . . . .	71

---

A.3	Settings for the solar particle fluence model in SPENVIS for a one year CubeSat mission in LEO. . . . .	72
A.4	Settings for the galactic cosmic ray flux model in SPENVIS for a one year CubeSat mission in LEO. . . . .	72
A.5	Settings for the total ionising dose model in SPENVIS for a one year CubeSat mission in LEO. . . . .	72
A.6	Settings in SPENVIS for MULASSIS simulations for different sources of radiation particles for a one year CubeSat mission in LEO. . . . .	72
A.7	Settings for the geometry in SPENVIS for MULASSIS simulations for a one year CubeSat mission in LEO. . . . .	73
A.8	Analysis settings in SPENVIS for MULASSIS simulations for a one year CubeSat mission in LEO. . . . .	73
A.9	Orbit specifications for a 10 year areostationary mission in SPENVIS. . . . .	74
A.10	GCR model settings in SPENVIS for an areostationary orbit. . . . .	74



# List of Tables

2.1	Comparison of RADFET, FOXFET and FGDOS sensors, where $V_T$ represents the threshold voltage and $F$ the frequency output [58]. . . . .	16
3.1	Overview of the main stakeholder requirements. Priorities are either essential (E), conditional (C) or optional (O). . . . .	20
3.2	Overview of the radiation monitoring system requirements. The last column identifies the requirement verification method: T (testing), A (analysis), I (Inspection), S (Similarity). . . . .	22
3.3	Overview of TID in function of shielding thickness for a one year ISS-like orbit and for a 10 year areostationary orbit. . . . .	30
3.4	Overview of sensor temperatures calculated from the heat balance equation 3.3 for different situations. . . . .	32
4.1	Overview of the test runs for the radiation tests on the 16 <sup>th</sup> of December at HollandPTC in Delft. . . . .	47
4.2	Sensitivities of the sensors in $Hz/Gy$ in function of flux, TID and sensitivity setting. The TID is the amount of radiation already received before the start of that particular experiment. . . . .	49
4.3	Sensitivities calculated using linear regression for different parts of the frequency range for test run 07 with a beam energy of 70 MeV at DUT and a flux of $3.69e7 \# / cm^2 / s$ . The relative difference between each partial sensitivity and the sensitivity calculated over the nominal linear range is given as well. . . . .	53
4.4	Frequency degradation between tests, during a test break or during an RF trip of the cyclotron. The duration of each break is noted, as well as the frequency change per hour. . . . .	56
4.5	Overview of the resolution determined via noise tests. Sensor 1 is the prototype and sensor 2 is the FGD-03F chip. The filtered signal was determined using a moving average window of about 20 s. . . . .	57
5.1	Overview of the radiation monitoring system requirements. The last column states if the requirement has been met (V), not met (X) or still has to be verified (-). . . . .	68



# Acronyms

**ADC** Analog-to-Digital Converter 15  
**AI** Artificial Intelligence 17  
**API** Application Programming Interface 34  
**BIRA-IASB** Royal Belgian Institute for Space Aeronomy 22  
**CAN** Controller Area Network 21, 32–34  
**CCSRM** CubeSats Constellation for Space Radiation Measurements 17  
**CERN** European Organization for Nuclear Research 11, 14, 32, 47, 51, 54, 56, 63, 70  
**CHARM** CERN High energy AcceleRator Mixed field 14  
**CIPO** Controller In Peripheral Out 38  
**CME** Coronal Mass Ejection 5  
**CMOS** Complementary Metal–Oxide–Semiconductor 8, 10, 12, 14  
**COPI** Controller Out Peripheral In 38  
**COTS** Commercial Off-The-Shelf 3, 4, 18, 19, 21, 67  
**CRAND** Cosmic Ray Albedo Neutron Decay 6  
**CS** Chip Select 38  
**CSP** CubeSat Space Protocol 32, 34, 36, 39  
**CSV** Comma Separated Values 45, 47  
**DD** Displacement Damage 7  
**DUT** Device Under Test 41, 46, 63  
**ECSS** European Cooperation for Space Standardisation 22  
**ELDRS** enhanced low dose rate sensitivity 8  
**ESA** European Space Agency 22, 27, 36  
**ESP** Emission of Solar Protons 24  
**ESTEC** European Space Research and Technology Centre 32, 43, 67  
**FGDOS** Floating Gate Dosimeter 10, 13–15, 18, 19, 32, 33, 68, 69  
**FOXFET** Field Oxide Field Effect Transistor 14  
**GCR** Galactic Cosmic Ray 5, 23, 24, 26–29  
**GNSS** Global Navigation Satellite System 17  
**I/O** Input/Output 35, 38  
**I2C** Inter-Integrated Circuit 21, 32, 35, 38, 39, 45, 69  
**IDE** Integrated Development Environment 35  
**ISR** Interrupt Service Routine 38, 39, 45  
**ISS** International Space Station 5, 11, 22, 23, 26  
**JAXA** Japan Aerospace Exploration Agency 6  
**KDE** Kernel Density Estimation 57  
**LEO** Low Earth Orbit 3, 6, 17, 23, 28, 29  
**LET** Linear Energy Transfer 4, 41, 46, 63  
**MARIE** Martian Radiation Environment Experiment 28  
**MBU** Multiple Bit Upset 10  
**MEO** Medium Earth Orbit 17  
**MERLIN** Medium Energy Electron Flux in Earth’s Outer Radiation Belt 17  
**ML** Machine Learning 3, 16, 17, 69  
**MOS** Metal–Oxide–Semiconductor 7, 8, 10, 13, 53  
**MOSFET** Metal–Oxide–Semiconductor Field-Effect Transistor 8, 11, 13–15, 43, 51  
**MULASSIS** Multi-Layered Shielding Simulation Software 26, 28, 29  
**NASA** National Aeronautics and Space Administration 3  
**OBC** On-Board Computer 32–35, 39  
**PCB** Printed Circuit Board 37, 38, 42, 43, 63  
**PCDU** Power Conditioning and Distribution Unit 32  
**PSYCHIC** Prediction of Solar particle Yields for Characterisation of Integrated Circuits 24

**PWM** Pulse Width Modulation 38, 45, 63  
**RADECS** RADiation and its Effects on Components and Systems 70  
**RADFET** Radiation-Sensitive Field-Effect Transistor 11, 13–15, 20–22, 67, 68  
**RF** Radio Frequency 4, 39, 41, 46, 49, 51, 55  
**RMS** Radiation Monitoring System 19  
**RS** Recommended Standard 21  
**RTOS** Real-Time Operating System 33, 34  
**SAA** South Atlantic Anomaly 6, 23, 26, 29  
**SC** Satellite Constellation 3, 16, 17, 69  
**SCK** Serial Clock 38  
**SCL** Serial Clock 35  
**SDA** Serial Data 35  
**SEB** Single Event Burnout 10  
**SEE** Single Event Effect 9, 17, 19, 21, 46, 48, 69  
**SEFI** Single Event Functional Interrupt 10  
**SEGR** Single Event Gate Rupture 10  
**SEL** Single Event Latch-Up 10, 21, 67  
**SET** Single Event Transient 10  
**SEU** Single Event Upset 10, 21, 67  
**SPE** Solar Particle Event 5, 17, 27, 28  
**SPENVIS** SPace ENViroment Information System 22–24, 26, 28, 29, 31  
**SPI** Serial Peripheral Interface 13–15, 21, 32, 36–39, 45, 63  
**SRIM** Stopping and Range of Ions in Matter 41, 46  
**TBD** To Be Determined 20, 21, 67  
**TID** Total Ionising Dose 7, 8, 13, 19, 26, 28, 29, 46, 47, 49, 51, 53, 55, 68, 70  
**TT&C** Telemetry, Tracking and Command 32  
**UART** Universal Asynchronous Receiver-Transmitter 21, 32  
**UDP** User Datagram Protocol 36  
**USB** Universal Serial Bus 21, 32  
**XML** Extensible Markup Language 36  
**Yamcs** Yet Another Mission Control System 36, 39

# Symbols

$A$  Area [ $m^2$ ] 31  
 $Al$  Albedo factor [-] 31  
 $D$  Dose [ $Gr$ ] 41  
 $E$  Energy [ $J$ ] 4, 10  
 $F$  Frequency [ $Hz$ ] xiii, 14, 37, 47  
 $Gy$  Gray [ $J/kg$ ] 4  
 $J$  Flux [ $1/m^2/s$ ] 4, 41  
 $J$  Radiative flux [ $W/m^2$ ] 31  
 $J'$  Fluence [ $1/m^2$ ] 4, 41  
 $P$  Power [ $W$ ] 31  
 $R$  Radius [ $m$ ] 31  
 $S$  Stopping power [ $N$ ] 4  
 $T$  Temperature [ $K$ ] 31  
 $V_T$  Threshold voltage [ $V$ ] xiii, 14  
 $\alpha$  Absorptivity [-] 31  
 $\epsilon$  Emissivity [-] 31  
 $\sigma$  Standard Deviation [-] 56  
 $\sigma$  Stefan Boltzmann Constant [ $W/(m^2K^4)$ ] 31  
 $m$  Mass [ $kg$ ] 10  
 $rad$  Absorbed radiation dose [ $0.01 J/kg$ ] 4  
 $t$  Time [ $s$ ] 47  
 $x$  Length or distance [ $m$ ] 4, 10  
 $z$  Electrical charge [-] 10



# Introduction

The Explorer 1 mission from the 1950s demonstrated that large amounts of radiation can be encountered in space [71]. Since then, countless space missions have been launched to investigate the space radiation environment. Understanding this part of the space environment is crucial, since radiation can greatly affect both satellite equipment and humans. Although this topic has been investigated intensely, knowledge gaps still exist. Not every region of the radiation environment has been investigated and radiation models require more data in order to reduce error margins. Furthermore, continuous monitoring is necessary, since the environment is very dynamic. Researchers also need real-time data for warning systems. However, classic space missions are expensive and require specialised engineering teams, making space radiation monitoring a non-trivial task. The main goal of the thesis was thus to design and characterise a simple and cheap space radiation monitoring tool. The hope is that such a system will be widely deployed, providing scientists with dosimetry data with a unique spatial and temporal coverage.

Several recent advancements in satellite and dosimeter technology could facilitate the design of novel radiation monitoring tools. These advancements are discussed in chapter 2. In section 2.1 CubeSat technology is described and some general statistics with regard to their popularity and reliability are shown. Section 2.2 deals with the space radiation environment. Here, the general concepts of radiation are explained and the space environment is analysed. Furthermore, the effects of this environment on satellite electronic equipment are discussed. Section 2.3 deals with the subject of radiation monitoring. Several types of radiation sensors are discussed, with a focus on the floating gate dosimeter chip developed by Sealicon. Section 2.4 touches on the subjects of machine learning and satellite constellations. These two technologies are described because the characteristics of floating gate dosimeters make them suitable for mass deployment in constellations, resulting in more available data for machine learning algorithms. The chapter concludes with a description of the research goals and questions that form the basis of the subsequent thesis research.

Based on the literature study, it was concluded that combining the advancements in small satellite systems and space dosimeters could enhance current radiation monitoring capabilities. A first step in the thesis project was thus to demonstrate that a floating gate dosimeter can be integrated in a small satellite system. For this, the FlatSat testbed at the European Space Research and Technology Centre was used. This testbed resembles a flattened out CubeSat satellite and provides researchers with the opportunity to demonstrate space system functionality without the need of a specific satellite prototype. The design of an interface between the dosimeter chip and testbed is discussed in chapter 3.

The next step in the thesis research was to characterise a floating gate dosimeter and verify its operating principles. Radiation tests have been performed at HollandPTC, a medical facility in Delft. Their cyclotron was used to irradiate the floating gate dosimeter with protons at various energies. The radiation tests and the results are discussed in chapter 4, and serve to prove that the sensor is in fact an improvement over previous dosimeter types. Finally, the conclusions of the FlatSat integration and the radiation tests are discussed in chapter 5.



# 2

## Literature Study

CubeSats and radiation monitoring are very broad terms, each containing a plethora of possible research fields. Hence, a literature study was performed to help in defining research goals, research questions and the pathways that can be followed to answer those questions. Additionally, the literature study provides the necessary background information necessary to fully comprehend the subsequent chapters. First, section 2.1 gives a basic explanation on CubeSats and highlights the relatively high failure rate of CubeSat missions. Second, in section 2.2, the space radiation environment and its effects on electronics and satellites in general are discussed. Next, section 2.3 examines the properties of solid-state dosimeters. Finally, some aspects of Machine Learning (ML) and Satellite Constellations (SCs), as well as their link with space radiation monitoring, are discussed in section 2.4. Using the knowledge gathered during the literature study, a research question for the thesis was formulated, which is discussed in section 2.5.

### 2.1. CubeSats

As far back as the nineties, and following the concept of 'smaller, better, faster, stronger', engineers have been exploring the idea of tiny satellites, weighing only a few kg or less and having dimensions in the order of several inches. Eventually, engineers at Stanford and California Polytechnic State University devised the concept of a CubeSat, which was originally intended for educational purposes and to facilitate experimental space missions [18, 50, 106]. According to the online database maintained by Dr. M. Swartwout, 1184 CubeSats have been launched between 2000 and 2019, with over 100 CubeSats being launched every year for the past three years [97], indicating their ever increasing presence in space and their usage for different applications. A graphical depiction of the amount of CubeSats launched each year is shown in figure 2.1.

Today, the CubeSat standard is used to describe small cubic satellites, with a 10 *cm* edge and weighing up to 2 *kg*. This single unit cube then describes a 1*U* satellite, but can be combined with other cubes to form larger sizes, for example, 3*U*, 6*U* and 12*U*, consisting of 3, 6 and 12 units respectively [17]. The standardisation of the CubeSat format, as well as their small size and relative simplicity, combined with the use of more Commercial Off-The-Shelf (COTS) components, results in a significant decrease in the efforts needed to develop a satellite. Project times are reduced, no extensive engineering teams are necessary and the low cost means that the market is open to universities and private companies that want to launch their own satellites. Additionally, the fact that these lightweight satellites can be launched in groups and piggyback on larger satellites means that the launch costs are reduced as well. Most CubeSats are launched into Low Earth Orbit (LEO) to keep costs low and because this is often dictated by their payload [18, 50, 106].

CubeSats have very wide applicability, and their success and further development is mainly driven by several factors: low initial investment cost, easy accessibility, standardisation of formats, large active community and technological advances allowing for ever smaller components [21]. Thanks to the relative ease with which a CubeSat mission can be executed, they have been employed for more

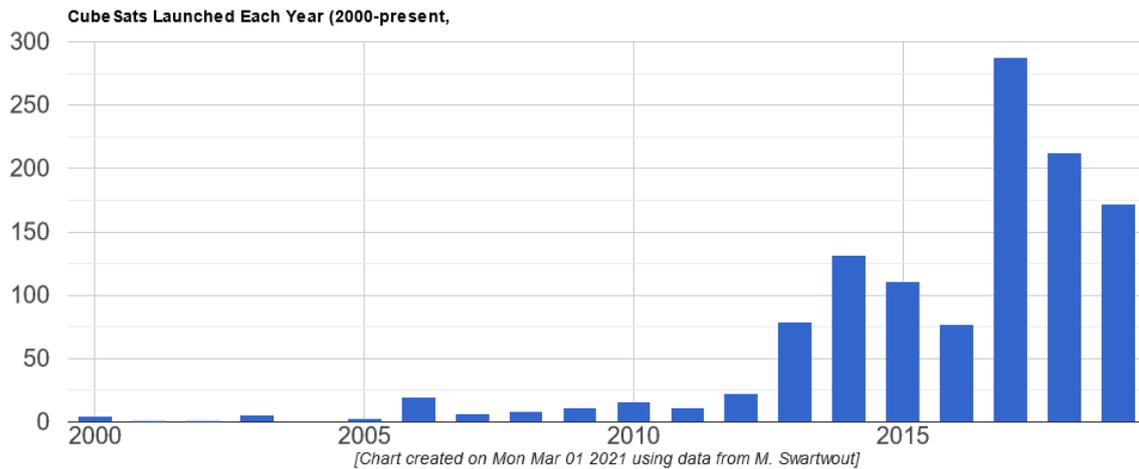


Figure 2.1: Amount of satellites launched per year according to an online database [97].

than educational purposes only. According to Villela et al. [106], who analysed a database including CubeSats launches up to 2018, most CubeSats are being used for technological development. Other important fields that make use of CubeSats are the Earth sciences and physics and astronomy. Despite their small size, CubeSats could also enhance deep space exploration and research. In 2018, National Aeronautics and Space Administration (NASA) launched its first two interplanetary CubeSats: MarCO-A and B hitched a ride with InSight, a Mars lander. The two small partners of the lander were used to monitor its landing and communicate with Earth. The success of this mission proves that CubeSats could be used for other deep space missions [64].

### 2.1.1. CubeSat Reliability

Reliable design of engineering systems is always important, but space systems have far more stringent requirements. Repairs in space are almost always unfeasible and unreliable systems might result in a loss of the mission and the millions of dollars used to design, construct and launch the satellite. In other words, "there are no second chances in space missions" [35]. Hazards in space include heavy thermal stresses, intense radiation related to the Sun, Van Allen belts or other galactic rays, magnetic forces, space debris, Radio Frequency (RF) interference and illegitimate commands [84]. Previously in this chapter, CubeSats have been described as a fast, cheap, simple alternative to conventional larger satellites. Besides the benefits mentioned before, these properties also come with a drawback, namely their low reliability. The database created by Dr. M. Swartwout shows that almost a fifth of CubeSats were death-on-arrival, meaning that they were ejected from their launcher successfully, but communication could never be established. Another 9.4% suffered from an early loss. By combining these data with other sources, Langer and Boumeester [65] created a database comprising of 178 individual CubeSats in order to investigate CubeSat reliability issues. As the researchers point out, the use of COTS and short development times are part of what make CubeSats attractive, and using more space-grade components or performing extensive testing is not ideal to improve reliability. However, according to the authors, many failures could be prevented by performing more functional tests before launching the CubeSats. The authors also conducted a survey, in which they asked developers' opinions on the reasons for this high failure rate. Most developers believe software design errors could be at the basis of a critical failure in the first six months of operations. Also high energy radiation effects, although to a lesser degree, are stated as one of the reasons of early critical failure of CubeSats.

## 2.2. Radiation

Satellites have to endure a harsh environment whilst being launched and in orbit. It was already stated that these conditions influence the satellite's reliability and can result in mission failure. This section starts off with a short explanation on radiation in general, then continues with an explanation on the space environment, with a focus on radiation and its effects on satellite electronics.

Radiation is generally divided in ionising and non-ionising radiation. Non-ionising radiation does not have enough energy to remove electrons from atoms, but can cause atoms to move or vibrate. Ionising radiation is more energetic, more penetrating and can ionise the atoms it hits [77, 85]. An important concept when discussing charged particle radiation is the amount of energy deposited in the irradiated material per unit of length, called Linear Energy Transfer (LET) and expressed in  $MeV/\mu m$ . The LET is closely related to the stopping power, but is not exactly equal. The stopping power  $S$  is the amount of energy lost by the particle per unit of length, and since part of this energy can for example leave the irradiated material as Bremsstrahlung, not all of this energy is transferred to the material itself. However, this difference is usually small, resulting in equation 2.1. Here,  $E$  represents energy and  $x$  unit of length. Bremsstrahlung, also called braking radiation, is the result of part of the kinetic energy of particles getting converted into photons when they are suddenly braking [33, 100]. An additional important concept is dose, or the amount of energy absorbed per mass irradiated material. Several units can be used to describe the amount of energy or dose, of which the most popular ones are  $rad$ , defined as  $0.01 J/kg$ , and  $Gy$ , defined as  $1 J/kg$ . Note that energy absorption depends on material type, so the irradiated material type has to be mentioned as well [49]. Since silicon is a material widely used in electronics, this material is often used as a reference. Consequently, unless specified otherwise, the doses in this thesis project always refer to the absorbed dose in silicon. Other important mathematical concepts include the flux  $J$ , expressed in particles per unit area per unit time, and fluence  $J'$ , which is the flux integrated over time and thus expressed in particles per unit area [49, 85].

$$LET = -\frac{dE_{deposited}}{dx} \approx S = -\frac{dE}{dx} \quad (2.1)$$

### 2.2.1. Space Radiation Environment

The space radiation environment can be interpreted in more than one manner. From the point of view of the environment itself, it can be described by looking at types of particles and their energy levels. Alternatively, it can also be described based on the source of the radiation. From the satellite's point of view, the radiation environment can be described by looking at the different effects it has on its equipment. It can also be analysed by looking at different satellite orbits and their specific environments. In the next paragraphs, the radiation environment is described by looking at different radiation sources, types of particles and energy levels. In section 2.2.2 the different effects radiation can have on satellites and electronics are discussed.

For a satellite orbiting Earth, there are three main sources of highly energetic particles: the Sun, Galactic Cosmic Rays (GCRs) and trapped radiation. The Sun is characterised by sunspots, i.e. concentrations of strong magnetic fields in the Sun's outer layers. Their prevalence varies with a period of approximately 11 years, named the solar cycle. These spots sometimes result in solar flares, emitting high-energy particles and electromagnetic radiation in a broad spectrum. Powerful flares can cause disruptions in the Earth's magnetic field, satellites, radio transmissions and terrestrial power lines. These solar flares can be followed by a Coronal Mass Ejection (CME), during which massive amounts of plasma are ejected into space. They form one of the most important concepts in space weather since they can greatly influence the near-Earth space environment, resulting in geomagnetic storms and the disruptions mentioned before [9, 20, 74, 104]. An event associated with either solar flares or CMEs are Solar Particle Events (SPEs). These events are hard to predict, vary greatly, can last up to more than a week and result in high amounts of high-energy ionising radiation reaching Earth. Since they are mostly composed of protons, they are also called solar proton events. But besides protons they contain electrons, alpha particles and heavier ions as well. In vicinity of the Earth and at low inclinations, these particles are deflected by the Earth's magnetic field. Consequently, only high energy particles can reach lower altitudes, whilst low energy particles can get trapped in the radiation belts [49, 56, 85]. Contrary to intuition, these events might also induce a reduction in near-Earth radiation levels called a Forbush decrease: the magnetic field associated with these events can deflect GCRs, effectively reducing the amount of radiation that can reach, for example, the International Space Station (ISS) [77, 90].

The second source of highly energetic particles are GCRs. These rays, with energies ranging from  $MeV$  to  $GeV$  and above, arrive at Earth from somewhere in our galaxy or from even further away. Some of these have the highest energies ever observed for a particle, and their origins remain enigmatic. They

mostly consist of protons, over 85%, helium and, to a smaller degree, heavier ions and  $\beta$  particles. The intensity of the rays arriving at Earth is negatively correlated with solar activity due to the Forbush effect described above. The Earth's magnetic field and atmosphere provide protection against these particles, with the most protection being provided around the equator. At ground-level they can be detected by studying secondary particles, mostly neutrons, originating from their interaction with the atmosphere [31, 49, 77, 85].

The final main source of radiation are the two toroid shaped belts surrounding the Earth, called the Van Allen belts. These are regions in which charged particles are trapped as a result of their interaction with the Earth's magnetic field [36, 49]. Figure 2.2 depicts flux densities for protons and electrons above a certain energy level in function of distance from the Earth expressed in L-shells. Such a shell is a set of magnetic field lines that crosses the Earth's magnetic equatorial plane at a certain amount of Earth radii measured from the centre of the Earth [68]. As can be seen from the figure, the inner belt mostly consists of high energy protons and electrons, but since the protons contribute most to the dose received at these altitudes, because they have higher energies, the inner belt is sometimes also called the proton belt [31]. Typically, electrons have energies in the order of 1 MeV and the protons in the order of 10 MeV up to 100 MeV and more. Some minor fractions of heavier ions are present as well. Most of the inner belt particles are a result of the decay of secondary neutrons produced by cosmic rays hitting the atmosphere, a mechanism called Cosmic Ray Albedo Neutron Decay (CRAND). Solar events also contribute to the composition of the inner belt [31, 33, 49, 68]. An important aspect of the Earth's magnetic field is that it is tilted with respect to the Earth's axis of rotation and that the centre of the field does not coincide with the centre of the Earth. As a result, the inner radiation belt extends to very low altitudes over the South Atlantic Ocean. This region, called the South Atlantic Anomaly (SAA), is of particular importance to satellites orbiting in LEO, since it can greatly influence normal satellite operations. Additionally, the region also poses a danger to human spaceflight [25, 33, 77]. The outer belt is called the electron belt because, although there are protons and some heavier ions present, most of the dose received here is due to electrons [31]. Electron fluxes here are higher than in the inner belt, with the electrons having a wider distribution of energy levels as well. Particles trapped here come from both solar and atmospheric origins. Because of the influence of solar winds and events, the outer belt is much less stable compared to the inner belt.

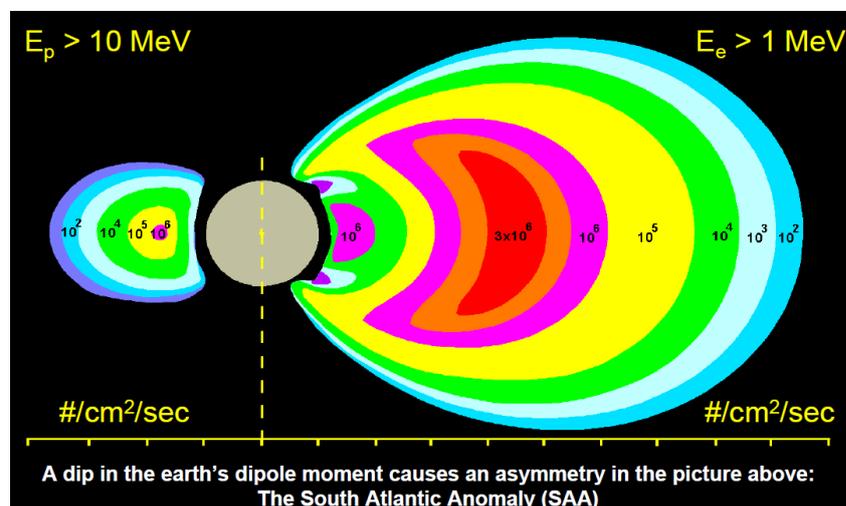


Figure 2.2: Proton and electron intensities in the Van Allen belts according to the AP-8 and AE-8 models [63].

### 2.2.2. Radiation Effects on Electronic Equipment

Environmental effects have been causing anomalies in satellites since the early space age. An interesting distribution of anomalies caused by the space environment is shown in figure 2.3 and shows the importance of understanding radiation in space. The eventual impact of the aforementioned anomalies can vary greatly, from short outages or minor inconveniences, to complete mission failures. For instance, the Hubble Space Telescope can only make observations for half of its orbital period, since

measurements are not feasible whilst Hubble is passing the SAA [95]. Radiation also affected Hubble's CCD detector, used for imaging and spectroscopy, causing changes in the sensor's operating parameters. If exact models could be build relating the progress of these effects to the amount of radiation, an on-board radiation sensor could indicate how the retrieved data were affected. Scientists could then retrieve the original data from the modified ones by using these models [61]. Improved measurements of the radiative environment could also lead to more precise downtimes and thus more measurement time. Another example of possible serious implications of radiation damage is the Globalstar satellite constellation used for communications. A couple of years after the their launches, in the period 1998 – 2000, several satellites of the constellation began experiencing failures in their receivers. It was later found that this was caused by displacement damage in an on-board optocoupler. Note that not all the satellites in the constellation experienced this, indicating the variability between what are essentially identical components [33]. A more recent example of radiation damage is the Hitomi satellite, part of the astronomy program of Japan Aerospace Exploration Agency (JAXA), which was lost after an unfortunate cascade of events that started with a radiation induced failure when flying through the SAA [75].

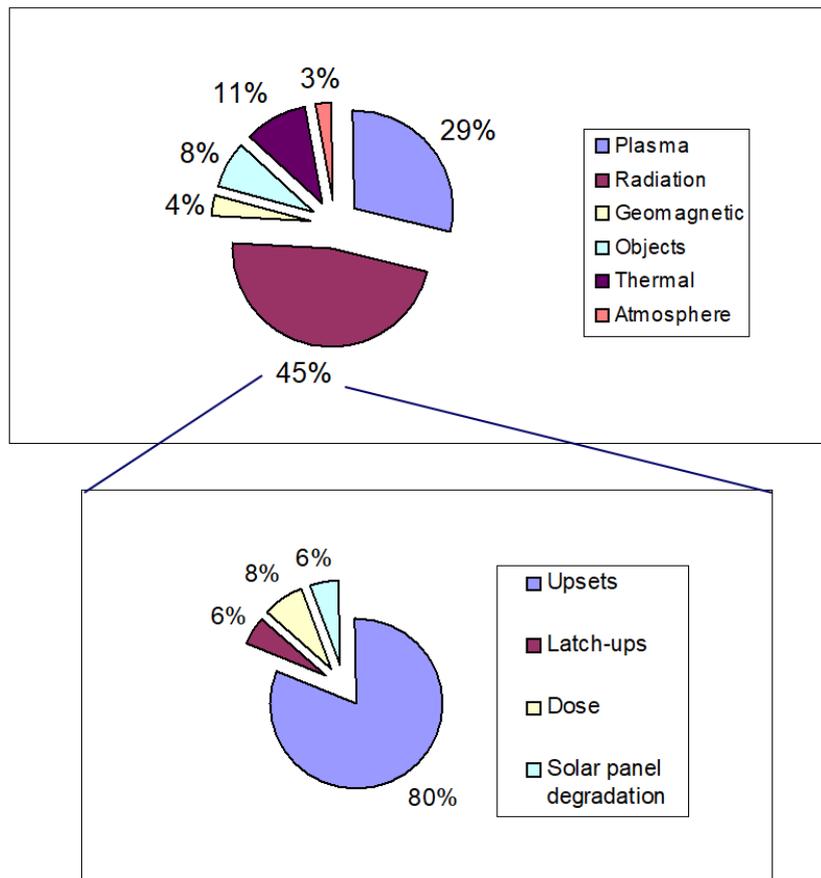


Figure 2.3: Satellite anomalies caused by the space environment [9, 32, 33].

Energetic particles change the behaviour of electronic equipment via several mechanisms. First, when particles interact with atomic nuclei, they can displace them out of their position in the lattice structure of the material, creating defects. These defects, called Displacement Damage (DD), are the result of accumulated non-ionising radiation caused by protons, electron, neutrons, ions and even photons. The displaced nuclei can knock back multiple other nuclei in their turn, or cause ionisation effects along their path. In semiconductors this new state establishes new energy levels in the semiconductor band gap. These effects cause the material's bulk properties to change, leading to the degradation of components, i.e. a change in operational behaviour of electric components, and possibly an eventual failure. Also mechanical and optical properties can be changed via DD [10, 31, 33, 51, 85, 104].

Second, when charged particles have enough energy, they can ionise the atoms in the irradiated material. In semiconductors and insulators, this results in the creation of electron-hole pairs that can affect the behaviour of circuits in satellite components. The effects of ionising radiation can accumulate over time, resulting in Total Ionising Dose (TID). The damage can be illustrated by considering a Metal–Oxide–Semiconductor (MOS) device. Four principal processes lie at the basis of the radiation effects on these devices [79]. First, electron-hole pairs are generated in the gate oxide as a result of ionising radiation. Part of these pairs recombine, but, under influence of an externally applied gate voltage, the electrons quickly travel to the gate. The fraction of holes that did not recombine is called the charge yield. Second, the leftover holes, which are much less mobile, slowly travel to the oxide-silicon interface. The third process starts when these holes arrive near the interface. Here, the holes can get trapped in the oxide, causing a shift in the threshold voltage of the device. These trapped charges can anneal over time, meaning that the voltage threshold shift might disappear over time as the trapped holes recombine with electrons. The final process is the accumulation of traps in the interface with state energy levels in the silicon band gap, resulting again in a shift of the threshold voltage. For P-channel transistors, these interface trapped charges are predominantly positive, whilst for N-channel transistors they are mostly negative [31]. These processes depend on several parameters, including the applied electric field, temperature, oxide thickness and manufacturing methods. Other parameters that influence radiation sensitivity include: type and rate of irradiation, device geometry, post-irradiation conditions, ageing, construction materials and impurities [53].

The TID effect is illustrated for a Metal–Oxide–Semiconductor Field-Effect Transistor (MOSFET) in figure 2.4. Trapped charges in the oxide cause a shift in the threshold voltage, which can lead to threshold voltage changes, current leakage or even the inability to switch the transistor off, i.e. a failure of the component [79, 85]. However, the thinning of the gate oxide has as a result that the quantum tunnelling of electrons can negate the trapping of charges in Complementary Metal–Oxide–Semiconductor (CMOS) devices [79, 88]. Consequently, the field oxide has become the main issue with regard to total dose effects. Trapped charges in the thick field oxides of MOS-technology can result in the formation of conducting paths, leading to leakage currents in the transistors or between adjacent transistors in an integrated circuit. Another effect that influences the received dose is dose enhancement. When radiation particles travel through a thin oxide layer packed between two other materials, the amount of secondary electrons generated by the interaction between this particle and the materials depends on the material itself. The thin oxide layer is subjected to a charge equilibrium unbalance because of this, receiving a higher dose than when it would be a standalone oxide [31, 79]. Attention should be paid to the dose rate, i.e. dose per unit of time, as well. The apparent increase in dose effects with higher dose rates can be attributed to the difference in exposure time, since several dose effects are time dependent [79]. Furthermore, bipolar transistors experience what is called enhanced low dose rate sensitivity (ELDRS), meaning that these devices exhibit more damage at low dose rates when compared to higher dose rates [57, 87]. A final observed effect is called the rebound effect. In an N-channel device, there are positive oxide traps and negative interface traps, resulting in an overall negative voltage threshold shift. The oxide traps can, however, anneal, and over time the threshold voltage shift becomes positive [79]. For a more comprehensive study on the effects of ionising radiation on transistors, including more complex mechanisms, the reader is referred to more specialised literature [53]. Also note that, sometimes, a single highly energetic particle can already deposit enough energy for the aforementioned effects to take place. This is called a micro-dose effect and could be seen as the overlap of TID effects and the events discussed in the next paragraphs. It should be clear that the concept of TID is complex, and effects highly depend on different environmental and situational parameters. When considering a complete integrated circuit based on MOS technology, different transistors can behave in different manners, depending on, for example, their respective gate voltage bias. As explained in , Additionally, radiation can affect a circuit's performance, by reducing its operating speed, and its reliability, by causing functional failures [31, 53, 87, 92]. The main causes of these failures are "an inability to switch from one state to another and increases in standby power" as a consequence of the radiation effects on the transistors [53].

An image depicting an estimation of the annual dose received from different particles in function of orbital altitude is shown in figure 2.5. The results are retrieved using the AE-8 and AP-8 models, which are models that can be used to simulate the trapped radiation environment. Orbits with higher inclination experience higher doses as a consequence of the diminished protection from Earth's magnetic

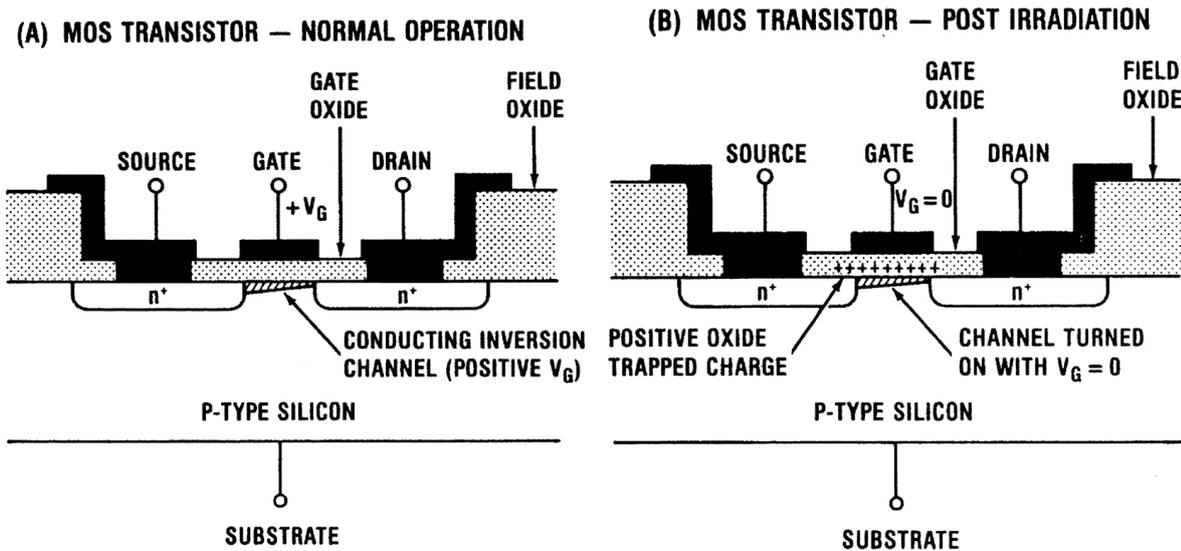


Figure 2.4: TID effect on a MOS transistor [79].

field. Commercial parts can tolerate about 5 *krad*, although this can be higher, and thus even more than enough for use in, for example, CubeSats. However, radiation tolerance of a device can differ significantly per lot, and caution is advised [93]. Radiation hardened parts can handle doses of 100 *krad* and even doses of several orders of magnitude higher [93]. Despite the effects TID can cause, there are few satellites that have experienced anomalies caused by these effects. This might be the result of excessive design margins [33, 87].

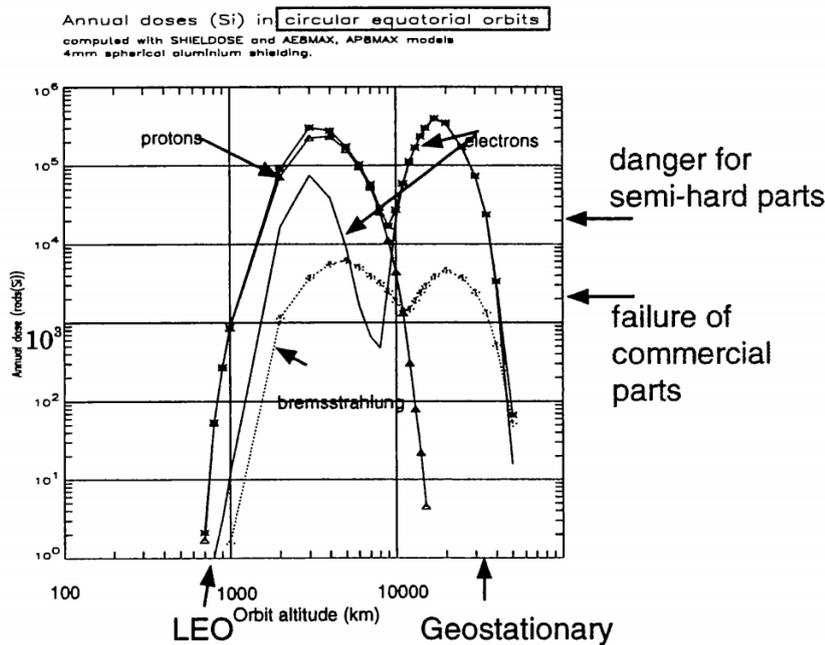


Figure 2.5: Annual dose received with 4 mm aluminium shielding in function of altitude for an equatorial circular orbit [25].

The third and final effect that radiation can have on electronic equipment occurs when a single highly energetic particle induces enough ionisation along its path to disturb the proper functionalities of electronic components. These Single Event Effects (SEEs) have a variety of different outcomes and can be divided in hard and soft effects. Hard effects are permanent, whereas soft effects can be resolved by, for example, a reset or a power cycle. The mechanisms behind these effects are similar to those

mentioned in the previous section, i.e. ionising radiation creates electron-hole pairs which can travel in the semiconductors resulting in abnormal voltages and currents in the affected transistor [31, 99]. The different types SEEs can be classified as shown in the list below. The details in this list are an amalgamation of the information found in [31, 48, 72, 85, 99]. Note that different literature sources might use slightly different classifications of these effects.

- Single Event Upset (SEU) (soft): A change of state, for example a bit flip in a memory component. A Multiple Bit Upset (MBU) happens when multiple bits are flipped in neighbouring memory cells by a single particle.
- Single Event Transient (SET) (soft): Current or voltage transient in a component's circuitry.
- Single Event Functional Interrupt (SEFI) (soft): Fault in the component that stops normal operation of the device.
- Single Event Latch-Up (SEL) (soft/hard): Short circuit in CMOS technologies that can possibly lead to permanent damage in a component if it is not mitigated.
- Single Event Gate Rupture (SEGR) (hard): Breakdown of the gate oxide in, for example; MOS transistors.
- Single Event Burnout (SEB) (hard): Activation of a parasitic structure in a power transistor resulting in a burnout because of high currents.

### 2.3. Solid-State Detectors

Over the decades numerous methods have been devised to measure radiation, from the older ionisation chamber detectors, to the more recent technologies employing several semiconductor radiation effects to measure radiation. Several books on the matter have been written, and a whole overview of detector technologies would be out of scope for this project. Because of their specific advantages mentioned in this section, the focus is on solid-state detectors, more specifically, semiconductor based dosimeters. Some available sensor types are explained, and a few practical examples of such sensors are given, with the main focus being on Floating Gate Dosimeters (FGDOSs).

Although the term solid state detector could technically refer to any type of radiation detector that is based on a solid-state medium, it is most often used to refer to those detectors that are based on electron-hole pair creation in a solid-state semiconductor, mostly silicon or germanium. Main advantages of such detectors are their compact size, excellent energy resolution and fast timing, whilst a disadvantage is their sensitivity to radiation damage [62, 101]. Note how their advantages make them particularly suited for use in small satellite missions. How this radiation sensing process works might be best explained by looking at a simple example, the PN-junction. Here, an N-type and a P-type doped semiconductor are merged together, causing the excess electrons to diffuse from the N-type to the P-type material, and vice versa for the P-type material, resulting in a potential difference between the two materials. The region in which the electrons and holes have diffused, called the depletion layer, can be further enlarged by applying an external voltage. The junction is then said to be in reverse bias. As already explained in section 2.2.2, ionising radiation produces electron-hole pairs in the semiconductor. Under influence of the internal electric field, these pairs are separated, and the charge collected at the external ends of the junction can be recorded by appropriate electronics. The applied electric field should be chosen so that the created pairs do not have the opportunity to recombine, since this would result in an undetected amount of radiation. Additionally, these electrons or holes, termed charge carriers, can get trapped by imperfections in the lattice structure of the semiconductor, affecting the junction's behaviour. Using different materials, layouts and processing techniques, different types of solid-state detectors can be constructed [101].

Silicon detectors can be used to determine the type of radiation particle via a  $\Delta E$   $E$ -detector, also called a telescope. In its simplest configuration, such a telescope contains two silicon detectors: a thin detector for  $dE/dx$  detection and a thick one for  $E$  detection, with  $E$  representing energy and  $x$  the distance travelled. In figure 2.6 a basic telescope setup is depicted. For non-relativistic particles, Bethe's formula can be written as shown in equation 2.2. The constants  $C_1$  and  $C_2$  are known, and the factor  $mz^2$ , with  $m$  the mass and  $z$  the charge of the ion, is specific to the type of particle that struck the detector. By letting the sensor measure  $dE/dx$  and  $E$ , the latter being the sum of the energy deposited in both sensor layers, the type of particle and its energy level can be determined [28, 62].

$$\frac{dE}{dx} = C_1 \frac{mz^2}{E} \ln C_2 \frac{E}{m} \quad (2.2)$$

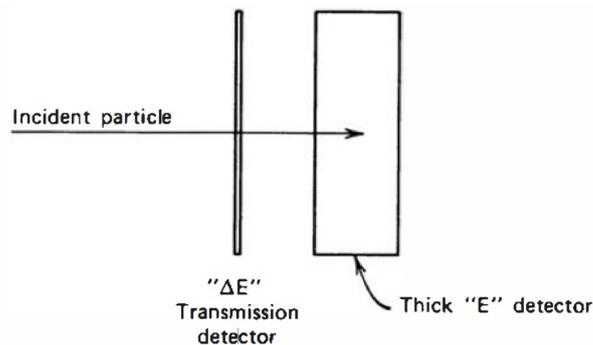


Figure 2.6: A telescope used for particle identification consisting of a  $\Delta E$  and an  $E$  detector [62].

For some applications it might be necessary to determine the location of irradiation. Several semiconductor sensor types exist for this purpose, one of which are silicon strips. For this type of sensor, the electrode on one side of the PN-junction is substituted by several electrode strips. The location of irradiation is then where the strongest signal of the electrodes is being detected. With this method resolutions as small as  $10 \mu m$  are possible. Configurations exist where strips are placed orthogonally to obtain two dimensional location measurements [62]. Another way to get location determination is by means of so-called pixel detectors. Here, the electrode is fabricated in a matrix-like pattern, and each cell of that matrix is electrically insulated from the other. Each pixel is accompanied by its own electrical connection and readout channels. In hybrid pixel detectors, this is done by separately manufacturing the pixel detector chip and the readout chip and soldering them together at the end of the process. A depiction of one such pixel is shown in figure 2.7. The Medipix family of sensors, developed since 1997 by collaborations led by European Organization for Nuclear Research (CERN), are hybrid pixel detectors and have been used in space [34]. More specifically, the Timepix sensor has been successfully used for space radiation measurements on board the ISS and CubeSats [7].

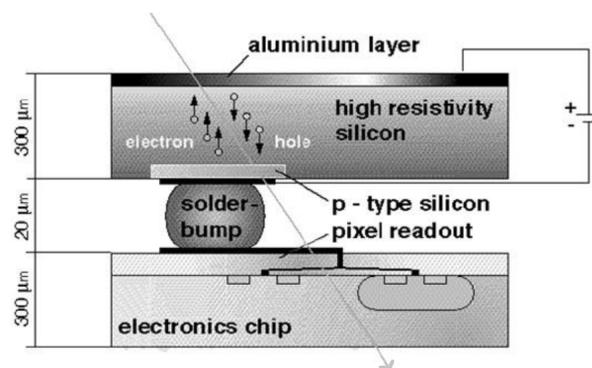


Figure 2.7: Illustration of a pixel in a hybrid pixel detector [73].

### 2.3.1. Metal-Oxide-Semiconductor Field-Effect Transistor Dosimeters

In section 2.2.2 it is described how radiation can affect the threshold voltage of a MOSFET. Since this shift is proportional to the radiation's deposited energy, a MOSFET can effectively be used as a type of semiconductor dosimeter. One advantage of this type of detector is their large range. For example, Varadis, a spin-out from the Tyndall National Institute, offers several types of sensors with a gate oxide thickness varying between  $0.1 - 1 \mu m$ , resulting in a total detectable range of  $0.3 - 1e6 rad$ . The ranges and sensitivity levels of the sensors can be further extended by combining different sensors

with modified gate oxide thicknesses. Another major benefit is their small size. A disadvantage is their temperature dependence, but this can be overcome by using dual transistors on a single chip. Other disadvantages include their dependence on the applied bias and the susceptibility to radiation damage [2, 62]. The sensors are now commonly referred to as Radiation-Sensitive Field-Effect Transistors (RADFETs) and have been used on countless space missions [41, 52].

Besides the aforementioned disadvantages, several other limitations of the sensor have to be taken into account. First, with increasing dose the oxide trapped charges influence the applied electric field and the sensor becomes saturated, having a reduced sensitivity. However, several mechanisms have been proposed that could extend the range of these sensors. The trapped charge can be neutralised by letting electrons tunnel from the gate to the substrate. Alternatively, the trapped charge can be eliminated by using the sensor in reverse bias. These mechanisms can be used to reset the threshold voltage of the sensor. A second disadvantage is that the oxide trapped charge can be neutralized by subsequent irradiation or by thermal annealing. The neutralisation of these traps is also known as fading. Furthermore, inherent noise limitations and the fact that fabrication imperfections cause slight performance variations have to be taken into account. Another disadvantage is that the sensors are not fully symmetrical and the angle of irradiation thus influences measured values. Note that the working of the sensor can be improved by applying a bias. The sensor can thus be used in measuring mode, with a bias applied, and in reading mode, where a reference current is forced through the sensor to indirectly measure the shift in threshold voltage [41].

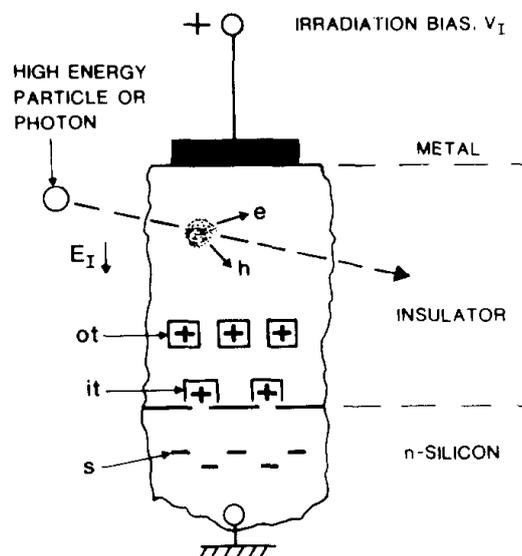


Figure 2.8: Cross section of the gate area of a MOSFET dosimeter. The acronyms ot and it stand for oxide traps and interface traps [52].

### 2.3.2. Floating Gate Dosimeters

Modern non-volatile memories are based on floating gate transistors, see figure 2.9. The gate of these transistors consists of multiple layers, with the layer closest to the substrate, called the floating gate, being electrically insulated from the rest of the transistor, thus acting as a potential well. When charge is forced into this well, it can retain its charge as long as no other charge is forced into it. Several methods exist to add charge to the floating gate. For instance, by applying a significant voltage between drain and source, causing avalanche breakdown, electrons become sufficiently energised to cross the isolating barrier towards or from the floating gate, depending on the bias applied on the top gate [81]. The device thus basically serves as a memory cell, but can also be used as radiation detector. Ionising radiation creates electron-hole pairs in the oxide around the floating gate, and the electrons or the holes, depending on the charge in the gate, migrate towards the gate, effectively discharging it. Another effect that has to be taken into account is photoemission of charge carriers in the floating gate, i.e. the carriers can get enough energy to cross the insulation boundary when irradiated. Since the threshold voltage of these transistors depends on the charge in the floating gate, the amount of radiation dose can be

determined by measuring the change in threshold voltage [3, 41, 42]. The monitoring of this shift can be done in two different ways: either by measuring the drain current of the transistor or by comparison with a conventional transistor [42]. A benefit of this type of dosimeter is that it can be manufactured in a conventional CMOS process. Other benefits that make them a good candidate for space dosimetry are their small size, low supply voltage and the possibility to integrate read-out electronics [3, 42].

A drawback of the sensors is the decrease in sensitivity with increasing doses, similar to MOSFET dosimeters. This decrease also depends on the recharge rate of the floating gate [14]. The measurement range of these sensors is also limited and during manufacturing a trade-off needs to be made between range and sensitivity. However, the FGDOS can easily be reset by recharging the floating gate, thus effectively extending its range and restoring its sensitivity [42]. Note that it is also possible to extend the measurement range of MOSFET based sensors, but the process for recharging the floating gate dosimeter is more straightforward and can be easily implemented, as demonstrated by the sensor chip discussed in the next paragraphs. Additionally, these sensors suffer from fading due to annealing or charge leakage [12]. The sensor has flown on the 4M Lunar flyby mission, after which the gathered data were compared to simulation results. Analysis has shown a good correlation between sensor and simulation data. Furthermore, the sensors worked without bias applied, which means power saving could be increased on future missions [22].

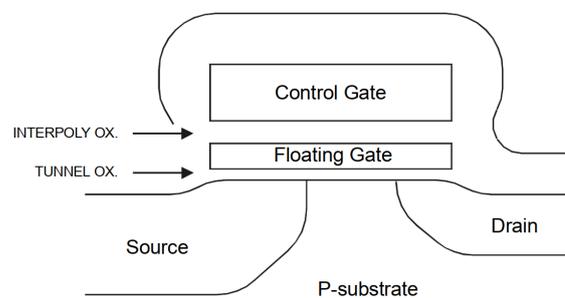


Figure 2.9: Cross-section of a floating gate transistor [81].

### Sealicon FGD-03F

At Sealicon, formerly iC-Málaga, a chip for radiation measurements based upon an FGDOS has been developed. Sealicon itself offers different versions of this chip, and both the FGD-02F demo board and the FGD-03F sensor chip have been used throughout the thesis. Since version 03F has been used for the radiation tests, only this version is further discussed here. A top view of a prototype of such a sensor is depicted in figure 2.10. The floating gate is extended over the field oxide where the electron-hole pairs are generated by radiation. From an electric point of view, the sensor can be divided in three parts: the extended gate acting as a capacitor, the N-channel MOS transistor that is used to read out the charge stored in the floating gate and the injector circuitry used to inject charge into the floating gate. A separate readout circuit is used to measure the transistor's drain current and convert this signal to one whose frequency depends on the floating gate charge, or in other words on the amount of dose [3, 14, 26].

The FGD-03F chip comes in a QFN32 chip package measuring  $5 \times 5 \times 0.9 \text{ mm}$  and contains two sensors. Each sensor can be programmed to function with high or low sensitivity, with values of about  $70 \text{ kHz/Gy}$  and  $10 \text{ kHz/Gy}$  respectively. The exact sensitivities have to be determined via calibration. Selecting a different sensitivity means that different readout circuitry is being used. For example, in low sensitivity mode, the readout circuitry incorporates less compensation circuitry but has a better linearity in the measurement range. The maximum TID that can be measured according to the sensor datasheet is  $500 \text{ Gy}$  or  $50 \text{ krad}$  [89]. A functional block diagram of the sensor chip is depicted in figure 2.11. Note that the diagram only shows one sensor, whilst the actual chip contains two sensors. This results in the actual chip having 30 pins instead of the 15 shown on the figure. Consequently, two pins of the standard QFN32 package are unused. The chip contains, of course, the floating gate dosimeter. The chip also includes a reference oscillator generating a reference frequency that can be used to compensate temperature effects, which is a tactic also applied to RADFET sensors. Next, it includes a

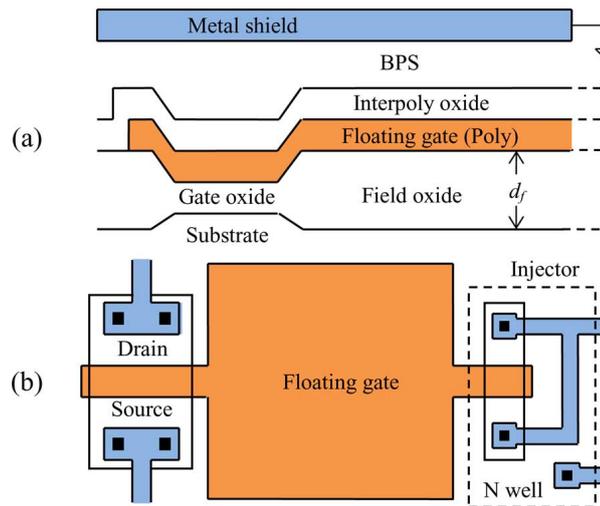


Figure 2.10: Cross-section and top view of a floating gate dosimeter developed by Sealicon [43].

charge pump and recharging circuit block, which are used to recharge the floating gate of the sensor, thus extending its TID range. The counters are used to count the amount of pulses generated by the sensor's frequency signal output. Here, the analogue frequency output is converted into a digital value. The chip also includes a block that regulates Serial Peripheral Interface (SPI) connections and stores data in registers. The communication interface is used to connect the chip to, for example, a microcontroller. Finally, the sensor has an internal temperature sensor as well [26, 89].

The GND, GNDD, VCC and VCCD pins in the figure are used for analogue and digital power supply and ground. The SCK, NCS, MOSI and MISO pins are used for SPI communication. The CK pin is used for the input of an external clock signal, which is used as a reference by the internal counter. The NSTBY pin is used to put the sensor in standby mode. In this mode, the sensor's power consumption is reduced to a minimum and no measurements are being performed. The NIRQ pin generates an interrupt signal whenever new measurement data are available in the sensor's registers. The VB and VCHP pins are shorted together, so the internal charge pump provides the necessary voltage to recharge the floating gate. Alternatively, an external voltage supply can be used as well. The other pins were not used during this project [26, 89].

### Comparison with RADFET

As stated before, the RADFET is a MOSFET based dosimeter that has been used on numerous space missions. Additionally, it is being used at CERN to monitor dose along the Large Hadron Collider. However, there are investigations going on to see whether their measurement system can be improved by using FGDOs [11, 13]. Depending on the oxide thickness of the sensor, different ranges and sensitivities can be achieved. The smallest range offered by Varadis, a RADFET with an oxide thickness of  $1 \mu\text{m}$ , is from  $0.3 \text{ rad}$  to  $1 \text{ krad}$ , whilst the sensor with an oxide thickness of  $100 \text{ nm}$  has a range of  $100 \text{ rad}$  to  $1 \text{ Mrad}$  [102]. In order to determine the dose received from the threshold voltage shift, the sensors need to be calibrated. From each batch of sensors a sensor has to be irradiated in a test facility to determine its calibration curve, depicted on the left in figure 2.12. This is different for the FGDOs. These sensors can be operated purely in their linear range by keeping the charge in the floating gate between between frequency thresholds that mark the limits of the linear range, as depicted on the right in figure 2.12. This eliminates the need of calibration curves over the complete range. Furthermore, when comparing the FGDOs with the RADFET with the thickest oxide, thus with a sensitivity as close as possible to the FGDOs, the FGDOs has a larger range.

The resolution of the sensors, i.e. the minimal detectable dose, depends not only their sensitivity but also on the noise in the circuitry around the sensor. For RADFET sensors, the resolution further depends on the applied gate voltage, and the sensors are more precise when biased with a gate voltage. Brucoli et al. [13] stated that RADFETs with a gate oxide thickness of  $400 \text{ nm}$  and a bias of  $5 \text{ V}$  can reach a resolution of  $0.4 \text{ rad}$ . In other research by the same principal author, the sensors with an

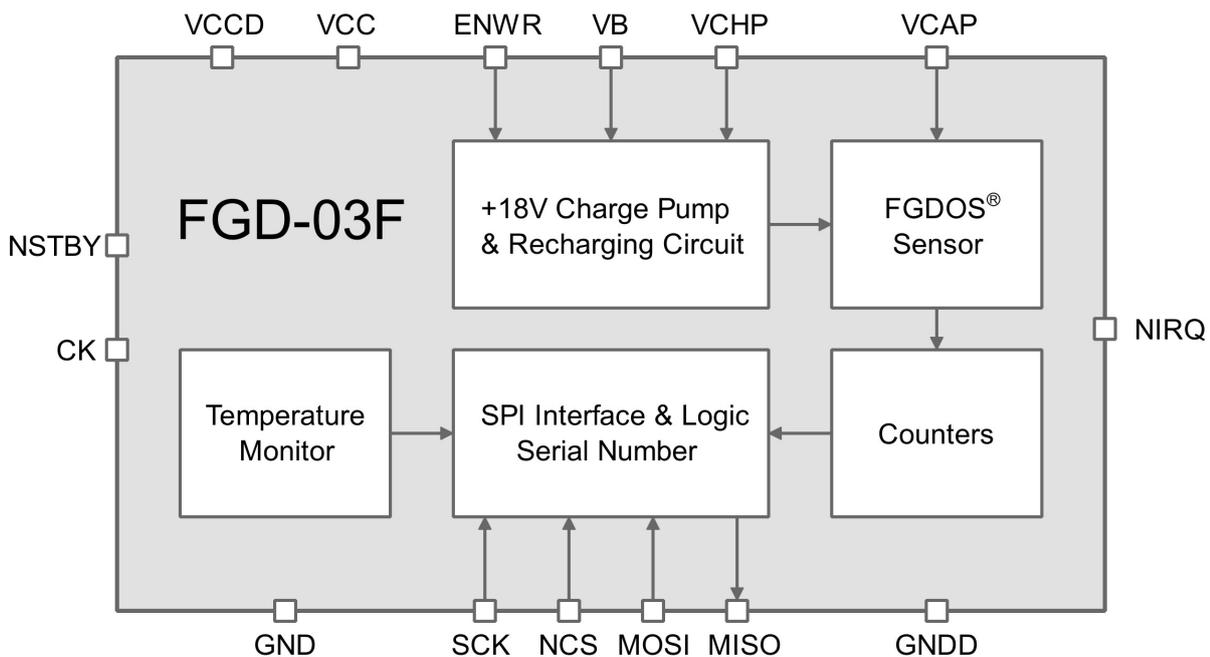
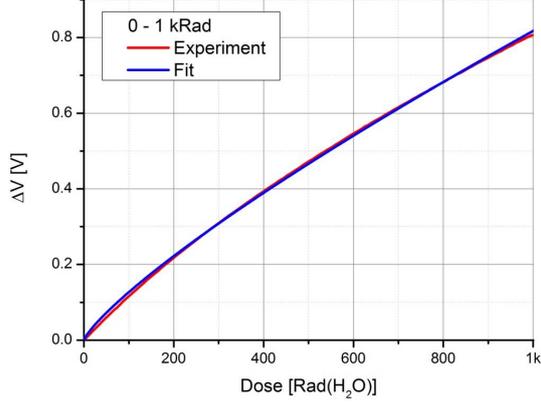


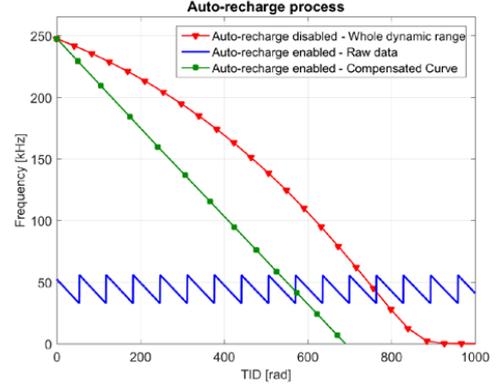
Figure 2.11: A schematic representation of the chip developed at Sealcon [89].

oxide thickness of  $100\text{ nm}$  have been used for testing and a comparison with FGDOSs because they had a higher reliability [11]. Here, the resolution that could be achieved with an FGDOS is  $16\text{ mrad}$ , more than a factor two difference with the RADFET. Additionally, when using the RADFET with a bias, a different calibration curve is needed. For RADFETs with an oxide thickness of  $1\text{ }\mu\text{m}$ , Varadis itself mentions a smallest detectable dose of  $0.3\text{ rad}$ , still less precise than the FGDOS. A comparison between the response of a  $100\text{ nm}$  RADFET, with and without gate bias, and an FGDOS during a test at the CERN High energy AcceleraRator Mixed field (CHARM) facility is depicted in figure 2.13. In this facility mixed radiation fields are generated by letting proton beams impinge on different possible targets. The figure clearly demonstrates the effect of the higher sensitivity of the FGDOS, resulting in more precise measurements and less noise. Although RADFETs can be constructed with thicker gate oxides for increased sensitivity, this comes with a major drawback. Since CMOS production technologies today require ever smaller dimensions and gate oxide thicknesses for increased performance, these RADFETs have to be produced by custom processes. This results in more complicated production processes and the impossibility to directly integrate them with other CMOS circuitry. To resolve this, the gate oxide can be replaced with a thicker field oxide, resulting in what is called a Field Oxide Field Effect Transistor (FOXFET). These sensors can be constructed in standard processes and have a gate, now constructed of a field oxide, of about  $400 - 600\text{ nm}$ . For larger sensitivities with these MOSFETs, the gate oxides have to be enlarged and custom processes have to be used. Furthermore, using RADFETs with thicker gate oxides causes them to be more dependent on dose rates and energies [26]. As a result, the FGDOS receives more and more attention because of its high sensitivity and its straightforward fabrication process [13, 41, 58]. In table 2.1 an overview of the three different sensors mentioned above is shown.

Other characteristics like weight, size and power consumption are harder to compare using the example models above, since these parameters depend a lot on the circuitry that has to be used to operate and read out the sensors. The Varadis sensors come in a 6L SOT 23 package, which measures  $2.8 \times 1.6 \times 1.15\text{ mm}$ , which is smaller than the QFN32 package of the FGD-03F. However, to read out the RADFET, external tools like a current source, a voltage sensor and an Analog-to-Digital Converter (ADC) are needed. The ADC is used to convert the voltage reading to a digital signal that can be send to a processor or stored into memory. This is not the case for the FGDOS, since the signal is internally converted and ready to be send over SPI. The same argument holds for the small difference in weight. Although the Varadis sensor only weighs about  $1\text{ g}$  and the FGD-03F about  $2\text{ g}$ , the weight of the additional readout circuitry still has to be included for the RADFET. With regard to the power



(a) RADFET calibration curve showing threshold voltage shift in function of dose.



(b) FGDOS chip frequency output in function of dose. The blue curve shows how recharging the floating gate keeps the sensor operating in the linear range.

Figure 2.12: Comparison of the dose responses of the Varadis VT03 1 m RADFET [102], and a Sealicon prototype FGDOS sensor chip [12].

Table 2.1: Comparison of RADFET, FOXFET and FGDOS sensors, where  $V_T$  represents the threshold voltage and  $F$  the frequency output [58].

Sensor	Readout Complexity	Power Overhead	Dose Sensitivity	Technology	Sensing Principle
RADFET	low	medium	medium	custom CMOS	$V_{Th}$ shift
FOXFET	low	medium	medium	custom CMOS	$V_{Th}$ shift
FGDOS	medium	low / medium	high	Standard CMOS	$V_{Th}$ shift / $F$

consumption, both sensors can be used in passive mode, meaning that they do not need power supply until the user wants to retrieve a measurement. However, both sensor types experience drawbacks when used in passive mode. Measurements should be retrieved from the FGDOS at shorter intervals, compared to the RADFET, to prevent the sensor operating outside of its linear region. Operating the RADFET in passive mode, thus without bias, results in a lower resolution. Power consumption for the FGDOS in active mode, thus continuously powered, is between about 5 – 25 mW, depending on the exact settings and usage of the sensor [15]. It would be inaccurate to compare this to the power consumption of a single MOSFET, but Varadis also offers a readout module RM-VT02, which includes a current source and outputs an analog voltage signal from which the radiation dose can be determined. The board measures 38x15x3.5 mm, has a typical supply voltage of 24 V and a typical supply current of 2 mA, which results in a power consumption of 48 mW [103], still not including any ADC. This suggests that power consumption of a RADFET with readout circuitry, including conversion to digital signals, would be higher when compared to the FGDOS chip. Additionally, the supply voltage of the FGD-03F is only 5 V. This low supply voltage is preferred since it is easier to provide in small space missions. The price of the most precise Varadis RADFET is €90. This is cheaper than an FGD-03F, which costs €140. But again, the RADFET does not include any readout components.

## 2.4. Smart & Distributed Radiation Monitoring

Two recent technological developments in the space industry include the use of machine learning and the deployment of satellite constellations. Both ML and SCs can contribute to the exploration of space and provide improved space based services, such as enhanced Earth observation data gathering and analysis. ML is the process in which a computer builds a model based upon a dataset. The model can then be used to, for instance, get new insights into the data or to process newly gathered data. For a

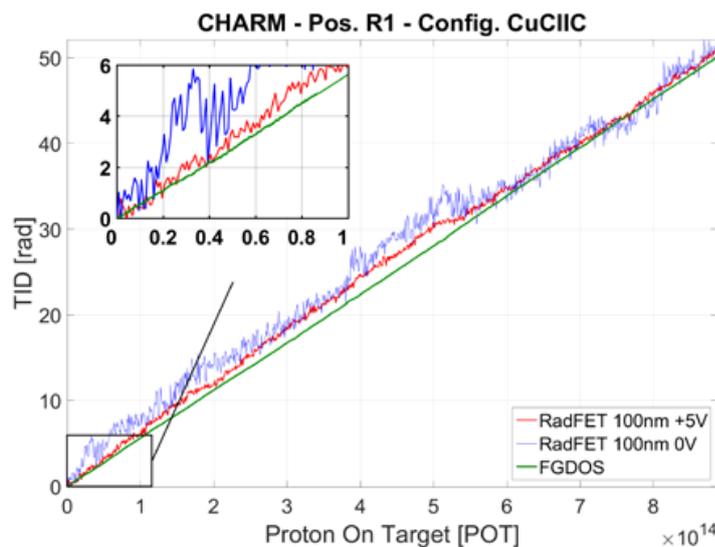


Figure 2.13: TID measured by RADFET sensors, with and without bias gate voltage, and an FGDS sensor. The sensors were placed in a mixed field generated by a 24 GeV proton beam and a copper target [13].

complete overview of ML, the reader is referred to more specialised literature [47]. Benefits of constellations are "an improved temporal resolution and ubiquitous availability of service when compared with traditional "monolithic" systems" [21].

### 2.4.1. Machine Learning in Space

Although ML algorithms have been around for about three decades, only more recently the field of ML, and the more general Artificial Intelligence (AI), have been expanding significantly. One application of ML in space is space weather research. Here, scientists try to predict the exact behaviour of the near-Earth space environment. Space weather events can seriously impact the operation of modern technologies, possibly having large socio-economic consequences. A first example hereof are the disturbances caused in power grids by geomagnetically induced currents, like substorm currents and the ring current. As a consequence of these currents, transformers can fail, resulting in disturbances or even complete power outages, like in Quebec in 1989. Another example is the influence space weather has on Global Navigation Satellite Systems (GNSSs). The harsh radiation environment, especially SPEs at the altitude of navigation satellites, can affect satellite operations and cause errors or failures via SEEs. ML predictions can help to prevent damages by warning the relevant parties that necessary actions have to be taken [20]. For instance, power grid operators can be warned to switch to safe mode before outages can occur. Furthermore, prediction models can improve navigation accuracy or radio communications and geomagnetic or auroral predictions [19].

Another possible application is radiation environment analysis. Finn et al. [37] demonstrated the possibilities of combining radiation measurements with ML to improve mission performance. In this particular case, radiation belt predictions are being made. As mentioned before, satellites that cross the radiation belts might need to shut down upon entry of these belts to prevent emergency shut-downs of electronic equipment caused by the radiation. These entries and exits are usually predicted using regression techniques or entirely theoretical models. In their work the authors employ a different approach using an ML algorithm to better predict these entry and exit times for the INTEGRAL satellite. According to the authors, the complex radiation environment can be approximated using ML algorithms, without having to fully comprehend the underlying physical mechanisms, provided an adequate dataset is available. ML methods can also be used to replace or improve the classical radiation environment models used for, amongst others, satellite design. Smirnov et al. [94], for instance, proposed the Medium Energy Electron Flux in Earth's Outer Radiation Belt (MERLIN) model, which focuses on reconstructing the medium electron flux in Medium Earth Orbit (MEO) based on data provided by GPS satellites. The authors concluded that their model is well capable of reconstructing the radiation environment, stating that their model can be used for informative purposes, satellite design or environment nowcasting.

### 2.4.2. Satellite Constellations

That SCs are becoming more popular because of their unique characteristics is demonstrated by the growing amount of satellites being deployed as part of a constellation. Despite the failures and rather unsuccessful business plans many earlier concepts, like Globalstar, had to endure, there are today dozens of companies or agencies that have proposed satellite constellation missions [24]. A well known example is the Starlink constellation of SpaceX, which hopes to bring over 4000 satellites into orbit by 2024 and will be used for broadband internet services [60]. An example of an operational CubeSat constellation is the Earth observation constellation operated by Planet, providing high frequency global coverage with quality images that can be used in sectors like forestry, agriculture and security [86]. An example of a proposed CubeSat constellation with respect to radiation monitoring is the CubeSats Constellation for Space Radiation Measurements (CCSRM) mission designed by the Skolkovo Institute of Science and Technology. The constellation consists of 12 CubeSats that would monitor the proton fluxes in LEO to provide information on the short term fluctuations of these fluxes [29]. As mentioned before, constellations have unique spatial and temporal coverage. This resulting increase in data availability could, for instance, result in new insights in the space radiation environment.

## 2.5. Research Goals & Question

The literature study demonstrated that there are still gaps in space radiation monitoring data and capabilities. Space radiation monitoring is necessary to improve engineering design models, augment forecasting methods, track environmental changes, classify and analyse on-board anomalies and perform scientific research. In the area of radiation monitoring, the FGDOS is a sensor design that offers new possibilities. Its low power consumption and small size might lay at the basis of innovative engineering concepts that are not feasible using classical sensors. It can, for instance, replace more expensive and bulkier sensors to decrease the costs of continuous radiation monitoring. These properties are particularly suited for the CubeSat format. The small satellites can then either be used as a technology demonstrator or to perform specific tasks that would be impractical or too costly using traditional satellite concepts. Furthermore, because of their compact size, low price and straightforward integration, these sensors could be widely deployed on future missions or even complete constellations, offering unique spatial and temporal coverage. The increase in data gathering might also prove useful for machine learning algorithms, resulting in improved modelling software or better space weather forecasting.

This thesis research aims to demonstrate that a functioning, simple and cheap system can be made that incorporates the technological advances in radiation sensors and CubeSat design. It is hoped that by building and testing a CubeSat compatible system that allows for COTS radiation monitoring, future missions can be designed based on this research and open questions in space radiation monitoring can be addressed. The system is also validated using radiation test facilities. The tests serve to prove that the system, i.e. sensor and readout electronics, can function as a compact and simple radiation monitoring system. They also help in understanding FGDOSs characteristics and might provide useful insights for future radiation research or mission design. In future research, the system can be further explored to demonstrate its functionalities as part of a constellation or to provide data for machine learning models. As a result, the following research question has been formulated, with accompanying subquestions:

*Can a relatively simple, lightweight and cheap space radiation monitoring system be designed that exploits the properties of floating gate dosimeters and that, combined with CubeSat technology, is able to answer the open questions in space radiation and/or improve continuous radiation monitoring?*

- Can the floating gate dosimeter be implemented into a microcontroller based system, resulting in a basic payload design for use in future space missions?*
- What would the characteristics of such a system be and what would be its main benefits for space missions, i.e. reliability, measurement range and resolution, cost, weight, power consumption and ease of implementation?*

# 3

## Floating Gate Dosimeter Payload Design

In the previous chapter, it was shown that an FGDOS can be used to improve current radiation monitoring capabilities. Furthermore, because of its characteristics, it can be used in conjunction with other miniature space technologies like CubeSats. This chapter starts with a discussion on stakeholder and system requirements for the design of an FGDOS CubeSat payload. Next, the radiation and thermal space environments in which the sensor would be required to operate are discussed in sections 3.2 and 3.3. To test the dosimeter payload interface, the testbed developed by Space Inventor, called FlatSat, was used, the details of which are explained in section 3.4. Because of an incompatibility between the testbed and the sensor with respect to communication interfaces, an Arduino microcontroller was used as well. The FGDOS chip developed by Sealicon has been introduced in section 2.3, but is described into more detail in section 3.5, where its programming is explained as well. The necessary Arduino programming for interaction with the FGDOS is explained in section 3.6. Finally, the full setup's functionalities, thus FlatSat with Arduino and FGDOS, are discussed in section 3.7. The descriptions and interactions presented in this section are referring to the FGD-03F sensor chip provided by Sealicon, unless specified otherwise.

### 3.1. System Design

In the literature study, a need for improved radiation monitoring capabilities was identified. Several research fields would benefit from a new dosimeter that is precise, small, cheap and easy to use. The correct design of a radiation monitoring system should start with identifying the people and entities that have a stake in the project and their specific requirements, see section 3.1.1. Section 3.1.2 discusses the verification and validation methods used throughout the design of the payload to make sure these requirements are met and the dosimeter payload answers stakeholder's needs.

#### 3.1.1. Stakeholder Requirements

A multitude of parties could benefit from an improved radiation monitoring system: research institutes and universities that want to investigate space radiation, space weather agencies that need continuous radiation monitoring for safety and forecasting, companies that require radiation data to improve models or design satellites and nuclear or medical research centers that require precise radiation level control. In this project, the focus is on the space related stakeholders. Another notable stakeholder is the TU Delft Lunar Zebro student project. The team has selected the FGD-03F as a payload for their small rover that is planned to be launched to the moon in 2022 [27]. Passive stakeholders include standards and protocols used in satellite design and communication, which have to be taken into account when designing the radiation monitoring system.

The aforementioned stakeholders have specific requirements with regard to the system. Several relevant stakeholder requirements and corresponding child requirements were identified and are listed in table 3.1. Defining proper and clear requirements before the start of the design avoids fuzzy definitions and misunderstandings throughout the project, possibly resulting in an unsatisfactory design. The Radiation Monitoring System (RMS) requirements RMS-01 deal with the performance of the dosimeter

when compared to other solid-state detectors. One of the main goals of the system is to outperform current sensor capabilities by having an improved sensitivity. Additionally, the range of the sensor should be appropriate for different types of space missions. The RMS-02 requirements define the necessary thresholds for some of the system's general characteristics, so that it can be used on future small satellite missions. Third, the RMS-03 requirements make sure that the system can be used in space and are based on the information found in literature [66]. Next, RMS-04 deals with the requirements to make the system a future COTS part. This is necessary to make sure the sensor can be easily implemented as a widespread payload or even scaled to satellite constellations. Finally, the RMS-05 requirements state the system's optional flexibility. This could be useful for different use cases, for example the use in nuclear reactors versus space missions.

Table 3.1: Overview of the main stakeholder requirements. Priorities are either essential (E), conditional (C) or optional (O).

Label	Description	Priority
RMS-01	The system shall outperform current solid-state dosimeters	
RMS-01.1	The system shall have a better resolution than current solid-state detectors	E
RMS-01.2	The system shall have a range adequate for near-Earth missions	E
RMS-01.3	The system shall have a range adequate for deep space missions	C
RMS-02	The system shall be future-proof with regard to the ongoing miniaturisation of space missions	
RMS-02.1	The system shall have an adequate size	E
RMS-02.2	The system shall have an adequate weight	E
RMS-02.3	The system shall have adequate power characteristics	E
RMS-03	The system shall be space grade	
RMS-03.1	The system shall have adequate temperature operating conditions	E
RMS-03.2	The system shall be able to function in vacuum	E
RMS-03.3	The system shall be radiation resistant	E
RMS-03.3	The system shall be able to handle adequate TID	E
RMS-03.4	The system shall be able to handle SEEs	E
RMS-03.5	The system shall be plasma resistant	E
RMS-03.6	The system shall have an adequate reliability	E
RMS-03.7	The system shall be able to be able to operate in zero-gravity	E
RMS-03.7	The system shall be able to be able to withstand typical launch stresses and vibrations	E
RMS-03.9	The system shall withstand interaction with the high altitude atmosphere	E
RMS-04	The system shall be a potential COTS part	
RMS-04.1	The system shall be easily produced	C
RMS-04.2	The system shall be cheap	E
RMS-04.3	The system shall be plug and play for common CubeSat layouts	C
RMS-05	The system shall be flexible	
RMS-05.1	The system shall have different sensitivities	O

From the stakeholder or customer requirements, different more specific system requirements were deduced that state how exactly the customer requirements can be fulfilled. An overview is given in table 3.2. Note that some requirements mention To Be Determined (TBD), which means no exact values were determined with regard to system specifications during this thesis project. This is considered acceptable given the exploratory nature of this work. Each requirement should be VALID: verifiable, achievable, logical, integral and definitive. This means that each requirement 1) can be verified, preferably via a quantitative method, 2) should be able to be achieved with the available resources, 3) should follow logically from the stakeholder requirements, 4) should be complete and 5) should be unambiguous [44]. The reasoning behind each of the requirements shown in table 3.2 is explained in the following paragraphs.

The resolution of RADFET solid-state detectors is described in section 2.3. The smallest detectable dose for the Varadis RADFETs is  $0.3 \text{ rad}$ . Assuming an increase of a factor 10 is desired, the minimum resolution of the system should be  $30 \text{ mrad}$ . The range requirements follow from the simulations discussed in section 3.2. These requirements are referred to with PERF in their label. The next requirements, labelled with MINI, state how the system shall be made compatible with current and future small satellite formats. For this purpose, several other assumptions have to be made. Besides the now widely used CubeSat format, several other concepts have been proposed and are being developed that are even smaller and cheaper. One example are PocketQubes, which consist of a number of p units

with each unit measuring  $5 \times 5 \times 5$  cm and weighing less than 180 g. An even more compact example are ChipSats or Sprites used for the KickSat mission mentioned in section 2.4.2. These satellites measure  $3.2 \times 3.2$  cm and have a thickness of a few millimeter [21]. Since the goal is to have a widespread presence of the dosimeter in space, it would be preferable that the dosimeter can be added as a secondary payload to virtually any CubeSat mission. Hence, the Sprite format is chosen as reference for size, mass and power requirements. For size, it is assumed that 10% of a Sprite's surface area is preserved for payload. Assuming a linear relation between mass and size of the PocketQube and the Sprite formats and a Sprite thickness of 3.2 mm, a Sprite would weigh about 4.7 g, which is indeed roughly the weight of the satellite units used for the KickSat project [83]. A weight budget of 15 – 50% of the spacecraft dry mass is allotted to the payload, resulting in the weight limit mentioned in table 3.2 [67]. With regard to energy restrictions, two factors are considered: power consumption and supply voltage. A power consumption of the communications unit in the order of milliwatts is mentioned for the Sprites, and a similar power consumption is assumed for the payload [83]. Since popular micro-processor boards often use a supply voltage of 3 – 5.5 V, this is considered an ideal requirement for the dosimeter system, since this would then prevent the need of additional voltage converters.

The readout electronics surrounding the dosimeter should be able to resist the conditions in space, including radiation, plasma, temperature, vacuum, zero gravity, launch forces and the Earth's outer atmosphere. These requirements are labelled with SPAC and are mostly retrieved from literature [66]. The expected dose was determined via simulations, see section 3.2. Additionally, the system should be able to tolerate SEEs. Operating temperature range of the equipment and payload are mission specific, but a first estimate was made based on the results of a heat balance calculation, see section 3.3. A necessary payload reliability of 0.89 is assumed [67]. To limit cost and make the system easily available, it would be ideal if the system can be produced in standard manufacturing processes or consists of COTS components. The system also needs to include a digital interface with a common communication protocol, so that it can be readily connected to different types of on-board computers. Because of its simplicity and reliability, only serial communication is considered. Common serial communication protocols include SPI, Inter-Integrated Circuit (I2C), Controller Area Network (CAN), Universal Asynchronous Receiver-Transmitter (UART), Universal Serial Bus (USB) and different Recommended Standard (RS) versions. These requirements are all labelled with COTS. An optional functionality, labelled FLEX, would be to be able to switch between sensitivities with the same system so that it can be easily programmed to fit a specific use case.

### 3.1.2. Verification & Validation

In order to make sure the designed system complies with the specifications put forward by the requirements, verification is necessary. Verifying that a (sub)system complies with a requirement can be done via four different methods: testing, analysis, inspection or similarity [44]. Testing is the most preferred method, as it quantitatively proves that the system complies with the requirement. However, this option is often the most expensive and is not always feasible. The second option is to provide analytical proof, through for example simulations or modelling, that the requirements are met. Third, some specifications can be verified by performing a standard quality control method, during which, for example, datasheets can be consulted to verify system characteristics. Finally, one can proof compliance with a requirement by describing how the system is similar to another system that has been proven to comply [44, 66].

In table 3.2, the verification methods that were used in this thesis project are shown for each requirement. Some requirements are not investigated or verified in this exploratory project, indicated by '-'. Additionally, some requirements were partially verified but would need more conclusive and precise verification in the future. The advised future verification method is then put between brackets. First, the PERF parameters were verified by testing the dosimeter in a radiation laboratory. More details with regard to test procedures and results can be found in chapter 4. Second, the MINI requirements were verified by physical inspection or by consulting datasheets. The power usage was also measured during the radiation tests. Requirements RMS-SPAC-01 to 03.2 were all checked by inspecting relevant datasheets or documents. In the future, however, more conclusive verification methods like testing should be performed. Alternatively, simulations can be used or the system can be compared to a similar one. Note that these last methods might be sufficient for an engineering model, but are not acceptable for a flight model. Requirement RMS-SPAC-03.03 with regard to SEEs could be verified

Table 3.2: Overview of the radiation monitoring system requirements. The last column identifies the requirement verification method: T (testing), A (analysis), I (Inspection), S (Similarity).

Label	Description	Priority	Verification
RMS-PERF-01	The system shall have a resolution of at least 30 <i>mrad</i>	E	T
RMS-PERF-02	The system shall have a measurement range of minimum 120 <i>krad</i>	C	T
RMS-MINI-01	The system shall have a maximum size of 3.2x3.2x3.2 <i>mm</i>	E	I
RMS-MINI-02	The system shall have a maximum weight of 2.4 <i>g</i>	E	I
RMS-MINI-03	The system shall have a maximum power usage of 10 <i>mW</i>	E	T
RMS-MINI-04	The system shall have a supply voltage of 3 – 5.5 <i>V</i>	E	I
RMS-SPAC-01	The system shall work in a temperature range of 60 – 380 <i>K</i>	E	I (T)
RMS-SPAC-02.1	The system shall consist of materials that can withstand exposure to vacuum and UV	E	I (T)
RMS-SPAC-02.2	The system shall consist of materials that generate little contamination	E	I (T)
RMS-SPAC-03.1	The system shall be radiation hardened to doses of at least 120 <i>krad</i>	E	I (T)
RMS-SPAC-03.2	The system's hardware shall be SEL and SEU resistant	E	- (T)
RMS-SPAC-03.3	The system shall have software algorithms dedicated to the detection and recovery of SEUs	E	- (T)
RMS-SPAC-04.1	The system shall prevent differential charging of surfaces by means of conductive coatings	E	- (I)
RMS-SPAC-04.2	The system shall have a common ground	E	I
RMS-SPAC-04.3	The system's electronics and wiring shall be shielded physically and electrically	E	I
RMS-SPAC-04.4	The system shall include electric filtering to protect circuits from discharge-induced currents	E	I (T)
RMS-SPAC-04.5	The system's exterior surfaces shall be able to withstand dielectric breakdown	E	- (T)
RMS-SPAC-05	The system shall have a reliability of 0.89 or higher	E	- (T)
RMS-SPAC-06	The system shall be able to operate in zero gravity	E	I
RMS-SPAC-07	The system shall be able to withstand launch stresses and vibrations TBD	E	I (T)
RMS-SPAC-08	The system shall be constructed with materials that resist atomic oxygen erosion	E	- (T)
RMS-COTS-01	The dosimeter shall be able to be produced in standard manufacturing processes	E	I
RMS-COTS-02	The system, besides dosimeter, shall consist of only COTS parts	C	I
RMS-COTS-03	The system shall cost as much as a system containing a RADFET dosimeter or less	E	I
RMS-COTS-04	The system shall communicate via a common serial communication interface	C	I
RMS-FLEX-01	The system shall have multiple sensitivity options	O	I

by software simulations in the future, e.g. using fault injection programs. Additionally, testing of the system to determine its response to SEEs should be done. However, SEEs are not further considered in this thesis project and are thus left for future research. Requirement RMS-SPAC-04.1 is not verified now but can be inspected in future research. Requirements RMS-SPAC-04.2 up to 04.4 were checked by inspecting the final system design and relevant datasheets. The dielectric breakdown RMS-SPAC-04.5, reliability RMS-SPAC-05 and atomic oxygen erosion RMS-SPAC-07 requirements should all be tested for in the future. The zero gravity RMS-SPAC-06 and launch stresses RMS-SPAC-07 were verified by means of inspection, but the last one should also be tested for in the future. All COTS and FLEX requirements were verified by inspection of the final system.

After the aforementioned verification techniques, validation is necessary to demonstrate that a product meets stakeholders needs and is ready for a particular use. The focus here is less on the requirements and more on how the system needs to act during real-life operations. This is needed because system requirements can only approximate reality and verification methods are imperfect [44, 66]. First, the stakeholders expect a system ready to be implemented on small satellite systems. The validation hereof, performed using a CubeSat testbed, is discussed in sections 3.4 to 3.7. Additionally, the stakeholders require a system that outperforms current miniature dosimeters, like the RADFET. This was validated by means of radiation tests and is discussed in chapter 4. In future research, additional validation steps still need to be performed. For example, stress tests should be performed to validate the system's robustness to the harsh space conditions and software errors.

## 3.2. Radiation Environment Simulation

In order to determine a required dose detection range for the sensor system, simulations in SPace ENViroment Information System (SPENVIS) were performed. SPENVIS is an online interface provided by the Royal Belgian Institute for Space Aeronomy (BIRA-IASB) under an European Space Agency (ESA) contract that allows for modelling of the space environment and its effects [38]. In this section the setup and results of these simulations are presented. Two different mission profiles were considered. When radiation doses are mentioned, the reader can assume the absorbing material is silicon, unless specified otherwise. First, the orbit of the ISS was chosen as an example mission profile because many radiation monitoring experiments have been flown on the ISS and their in-flight data could provide valuable validation datasets. Next, in order to simulate expected doses for deep space missions and future interplanetary missions, an orbit around Mars was considered. For both missions the start date was chosen to be January 2022. The specific date does not matter, but the year could be important because it determines whether the simulations are run during a solar maximum or a solar minimum, as explained in the sections 3.2.1 and 3.2.2. Note that the Sun has entered its 25<sup>th</sup> cycle around the end of 2020, with an expected peak in solar activity in 2025 [1]. Regardless of mission start date, the simulations settings have all been chosen to be conservative, resulting in more stringent system requirements that are automatically valid for periods of less radiation intensive periods. Screenshots showing the exact settings used for the simulations are added in appendix A. For the radiation models and settings, the guidelines from the European Cooperation for Space Standardisation (ECSS) were followed [39]. In figure 3.1, the typical workflow for a simulation in SPENVIS is depicted. First, a mission specific orbit is defined. Second, the different radiation sources and their models are defined. Finally, depending on what exactly needs to be simulated, different types of simulations can be run to determine total dose, solar cell degradation or other parameters. For validation of the different models, for example by comparing them with space mission data, the reader is referred to more specialised literature [105].

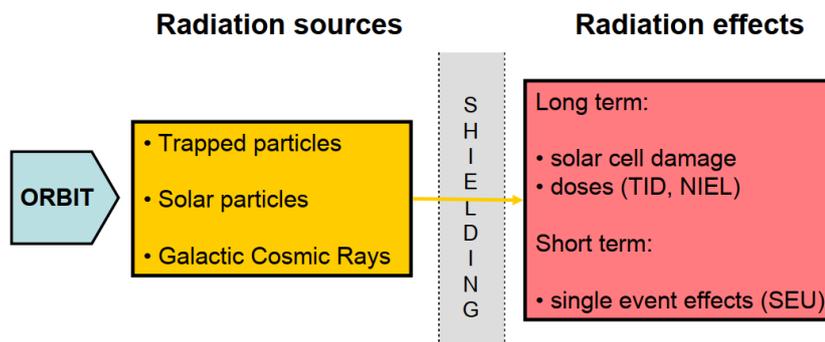


Figure 3.1: Typical SPENVIS workflow [30].

### 3.2.1. Low-Earth Orbit

For LEO, data from the ISS orbit were taken and entered into SPENVIS, see figure A.1 [82]. For simplicity, solar radiation pressure and atmospheric drag were not accounted for. A mission duration of one year was chosen, since this is a typical mission duration of a CubeSat at this altitude considering orbital decay [80]. An orbital period of 30 days was chosen as representative for the whole mission duration. Shorter time periods allow for faster simulations but then parts of the coverage of Earth's surface might be missed. After defining the orbit, the radiation models were set. For the Van Allen belts, the AP-8 and AE-8 models for trapped protons and electrons respectively were chosen, see figure A.2. These models are appropriate to determine cumulative effects for missions over 6 months, but the models do not include temporal flux variations or directionality. An error of about a factor two, or even more for the AE-8 model, should be taken into account as well [38, 39]. There have been several attempts to overcome these limitations, resulting in the AP-9 and AE-9 models, but these are no industry standard yet [45]. To be conservative, the solar minimum for the AP-8 and solar maximum for the AE-8 models were set, since this results in the highest proton and electron fluxes. The exact solar activity depends on the mission timeline, as explained before. The threshold settings are only used

to determine which fraction of the orbit passes the threshold and do not influence further simulations. Although the AP-8 and AE-8 models are static, local time variations can be selected to include time variations in the outer radiation belt in the AE-8 model down to the third L-shell [38, 105]. These outer zone variations result in the flux being a random variable, and thus confidence levels can be set. In SPENVIS, a confidence level can be set at 50% plus or minus a certain number of standard deviations. Again, to be conservative, a confidence level of 97,725% was chosen, meaning that there is a 2.275% chance that the fluxes are exceeded. A higher confidence level of 99.865 can be chosen as well, but this was deemed excessive. Additionally, for the GCR model, this level of confidence is not available, thus not allowing for a consistent choice of confidence level if higher levels would be set for the trapped radiation particles. An example of the output of this part of the simulation is shown in figure 3.2, where a world map with the mission average flux of protons is depicted. On the figure, the SAA as explained in section 2.2.1, is clearly visible.

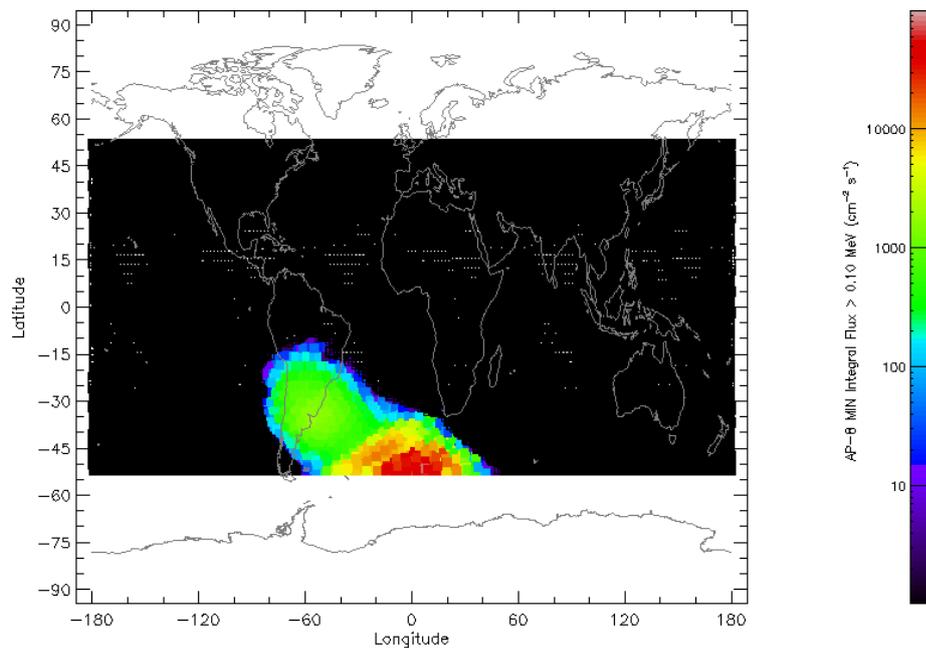
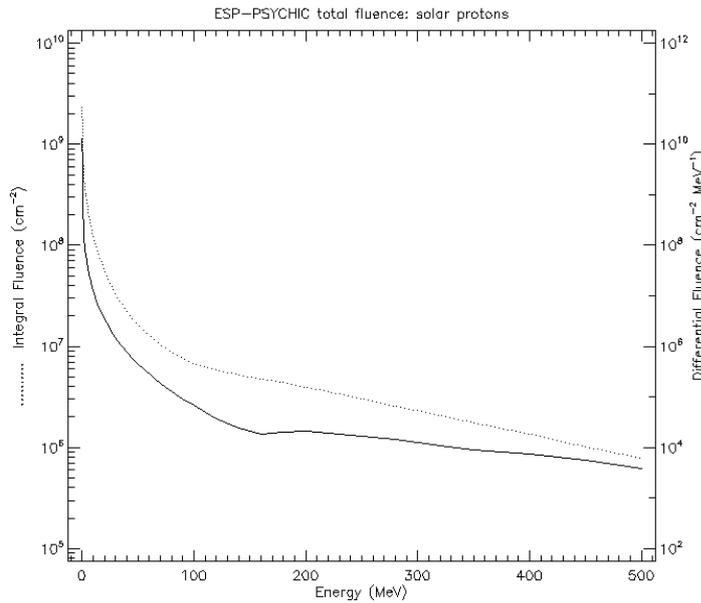


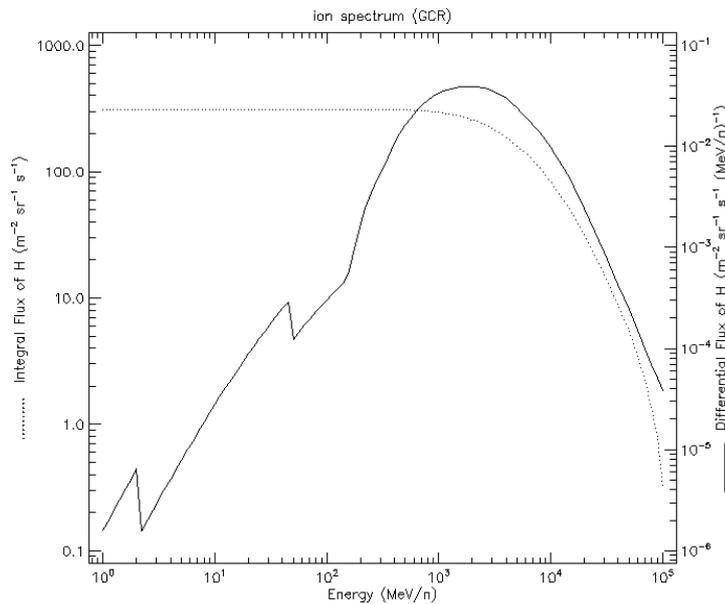
Figure 3.2: A world map with mission averaged proton fluxes. Output retrieved from SPENVIS for a one year mission in an ISS-like orbit.

Next, the solar particle fluence model was set up, see figure A.3. As advised by the standard, the Emission of Solar Protons (ESP) model was used [39]. In SPENVIS, this has been combined with the Prediction of Solar particle Yields for Characterisation of Integrated Circuits (PSYCHIC) model, which includes heavy ions and extends the energy range of the solar protons modelled in ESP. The complete ion spectrum was incorporated and the prediction period and solar cycle offset were set to automatically coincide with the mission time period. Again, to be conservative, a confidence level of 97.725% was set. The model does not include electrons, but this is only necessary when considering, for example, internal charging effects [39]. The magnetic shielding provided by Earth was included in the simulations and a quiet magnetosphere was chosen to be conservative. Finally, GCRs were modelled, with the settings as shown in figure A.4. The ISO-15390 model is advised by the standard [39]. This model takes variations in solar activity and the heliospheric model into account [38]. SPENVIS allows for selection of the standard model, which gives the average flux, or two other models that return the fluxes for other confidence levels based on an added standard deviation of the mean. The mean plus two standard deviations was chosen to be conservative, resulting in a confidence level of 97,725%. The total mission fluence of solar protons and GCR protons in function of energy is depicted in figure 3.3, and shows the different orders of magnitude for the peak fluxes and their respective energy levels. The fluences of heavier ions, like helium and lithium, are several orders of magnitude smaller. Note

that in the total dose calculations done by SPENVIS, the GCR fluxes are not taken into account. In the next paragraphs it is shown that this only has a minor influence on the total dose. Additionally, there are jumps in the graph for GCR protons. This effect was only noticed when choosing mission epoch as a time period in the GCR settings and not when a fixed solar minimum period was selected. This might suggest a bug in SPENVIS and has been reported on the SPENVIS forum. The solar minimum period was used in further GCR simulations to be more conservative, since the flux of GCR particles is negatively correlated with solar activity [77, 90].



(a) Fluence of solar protons in function of energy according to the ESP-PSYCHIC model.



(b) Flux of GCR protons in function of energy according to the ISO-15390 model.

Figure 3.3: Results of the SPENVIS simulation for solar proton fluence and GCR proton flux for a one year CubeSat mission in an ISS-like orbit.

After setting up and running the orbit and radiation sources models, the total ionising dose for simple geometries was determined. The settings for the TID model are depicted in figure A.5. Either default values for the shielding thickness or tabulated values can be used. Furthermore, three different dose models can be used: SHIELDOSE, SHIELDOSE2 and SHIELDOSE2Q. Each version is an update of the previous model and the latest model was chosen for the simulations. Updates include improved simulations and having additional options for detector materials. The shielding configuration was chosen to be the centre of an aluminium sphere, which best represents the system's possible future configuration in a satellite with radiation coming from all directions. The shield was considered to be composed of aluminium, because this is the most popular construction material for satellites. Finally, the target material was chosen to be silicon, the main constituent of semiconductor devices. The results of the simulation are depicted in figure 3.4. A typical CubeSat wall thickness is  $0.2\text{ cm}$ , although custom materials and additive manufacturing are being used today as well [76]. The SPENVIS report file mentions a total dose of  $11.25\text{ krad}$  for  $2\text{ mm}$  of shielding, with most of the dose being caused by trapped electrons. It should be noted that when setting the confidence levels of the AE-8 and ESP models to 50%, this value drops down to  $874.5\text{ rad}$ , indicating the high uncertainty in the models. The expected TID is much larger compared to the results presented in figure 2.5. This difference can be attributed to the high confidence levels set in the simulations presented here and to the difference in inclination between the orbit of the ISS and the equatorial orbit used for figure 2.5, causing the ISS to fly through the high particle fluxes of the SAA. Note that the simulations do not take into account the heavier ions present in the solar winds. Since these have fluxes several orders of magnitude smaller than the solar protons, their effects were considered negligible.

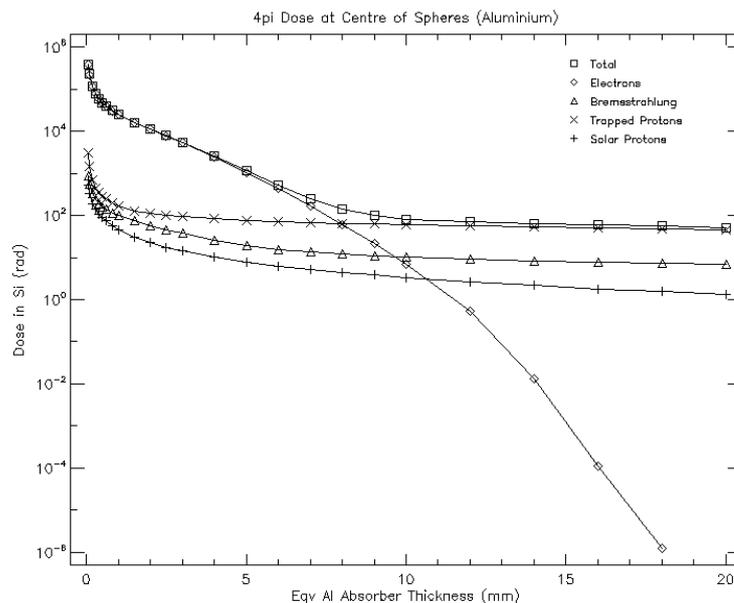


Figure 3.4: TID in Si in function of shielding depth according to a SPENVIS simulation for a one year CubeSat mission in an ISS-like orbit.

In order to determine the effect of the GCRs particles on the total dose, additional SPENVIS tools were used. There is the possibility to rerun the simulations with a user defined trapped particle environment, but this approach does not allow to include heavy ions or the high particle energy levels associated with GCRs. Instead, the integrated Multi-Layered Shielding Simulation Software (MULASSIS) package was used, which is a Monte-Carlo based simulation tool for the analysis of fluences and doses for more complex shielding and sensor geometries than what can be defined in SHIELDOSE. First, the trapped proton environment was simulated using the default geometry, consisting of 26 slabs of aluminium with the boundaries set as to equal the default shielding thicknesses from the SHIELDOSE simulations. The source settings for the MULASSIS simulations are depicted in figure A.6, the geometry settings in figure A.7 and the analysis settings in figure A.8. A total of 10.000 particles was chosen to be simulated in the Monte-Carlo simulations. Energy biasing can be used if faster simulations would be

needed. With energy biasing, there is a higher probability of low flux particles being generated in the Monte Carlo simulations. It is an event biasing technique that replaces the actual flux probability distribution function with an artificial one. It increases simulation efficiency and is especially useful for thick shielding simulations [38]. For full comparison, the SHIELDOSE simulation was rerun with a finite slab geometry, instead of a sphere, since this setup better represents the slabs of shielding used in the MULASSIS simulations. In the case of GCRs, different simulations were run for different isotopes of the same type of ion which then all have to be added one by one. A comparison of the output of SHIELDOSE and MULASSIS simulations is shown in figure 3.5. The results from the MULASSIS simulations agree well with those from SHIELDOSE for the dose caused by trapped protons, except for very thin shield thicknesses where the MULASSIS simulations give higher doses. This effect might be caused by, for example, interpolation errors or differences in Monte-Carlo simulation data, but is not further investigated. The results of the MULASSIS simulations also depend on the chosen method of interpolation, with the powerlaw interpolation method not agreeing with the SHIELDOSE simulations. The interpolation method does not matter for GCR simulations, since no interpolation method is then being used. The figure also depicts the estimated dose caused by protons originating from GCRs, which are clearly less influenced by aluminium shielding because of their high energies. Despite their high energies, GCR ions contribute several orders of magnitude less dose when compared to other dose sources because of their low flux. Although the comparison was only made for protons, heavier ions have even lower fluxes, and these contribute even less to the TID over the mission lifetime. When looking at figure 3.4, for a typical CubeSat shielding thickness of  $2\text{ mm}$ , the trapped protons are not even the most important dose source, and the GCR protons represent thus only a fraction of the total dose. In conclusion, for this particular mission, the effect of GCR particles on the total dose can be neglected, but it should be taken into account when more precision is required, certainly when working with thick shielding. Furthermore, the effect of heavier solar ions should be incorporated if more precision is required.

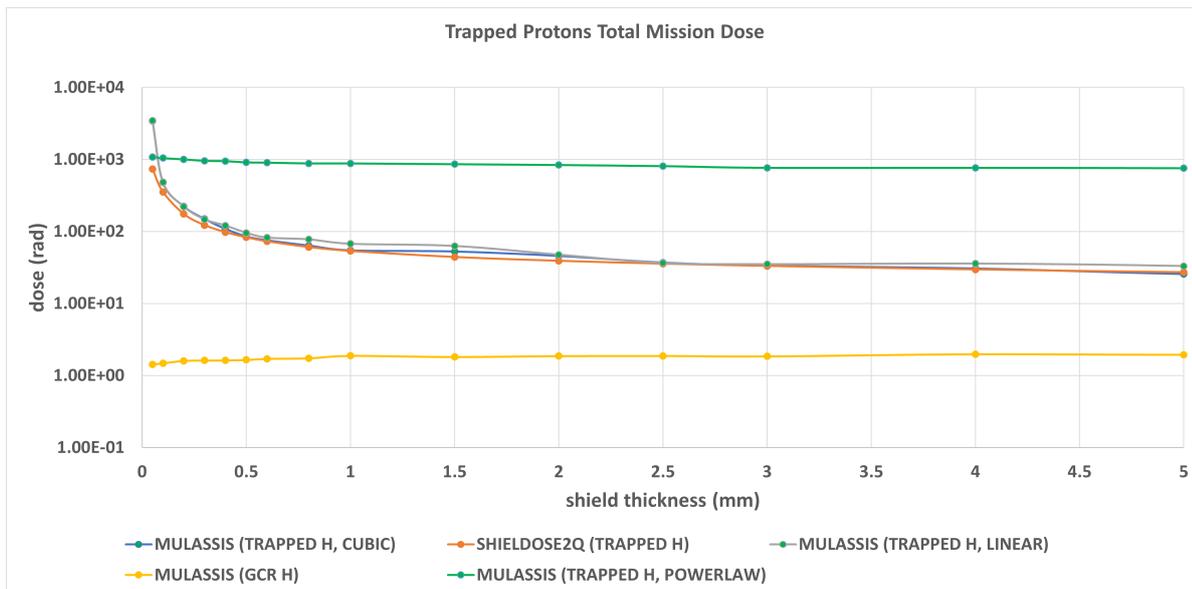


Figure 3.5: Comparison of the total dose for a one year CubeSat mission on an ISS like orbit using different models. The H in the legend represents hydrogen ions, i.e. protons. The terms cubic, powerlaw and linear in the legend represent the chosen interpolation method.

### 3.2.2. Mars

To investigate the possible use of the radiation monitoring system for deep space missions, an areo-centric orbit is considered. Both experimental data and simulations were used to determine the dose levels that can be expected in such an orbit. An example of a recent mission is the Trace Gas Orbiter, part of ExoMars, a joint ESA - Roscosmos programme to explore Mars. One of the scientific instruments aboard the satellite is the Liulin-MO dosimeter, consisting of two pairs of silicon PIN-diodes.

The measurements whilst in orbit around Mars between May 2018 and December 2019 can be found in literature [90]. The orbit is a circular orbit with an altitude of  $400\text{ km}$  and an inclination of  $74^\circ$ . The sensors itself are shielded from the space environment by  $1.2\text{ mm}$  of steel and  $1\text{ mm}$  of aluminium, resulting in a minimum energy threshold of  $30\text{ MeV}$  for protons and  $1.7\text{ MeV}$  for electrons. Results for fluxes and hourly dose rates of GCRs are depicted in figure 3.6. The time period of the measurements coincides with a solar minimum and no SPEs were recorded. Assuming a dose rate of  $15.5\text{ }\mu\text{Gy/hr}$ , based upon figure 3.6, the yearly dose for a mission in a low altitude areocentric orbit due to GCRs is  $13.58\text{ rad}$  considering the aforementioned shielding configuration.

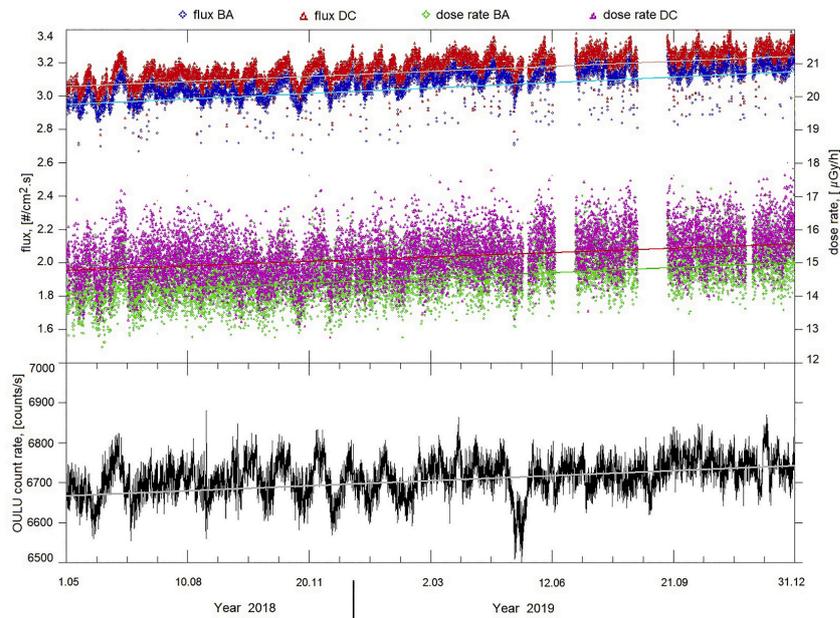


Figure 3.6: Flux and dose rates measured by the Liulin-MO dosimeter during a circular orbit around Mars with an altitude of  $400\text{ km}$  and an inclination of  $72^\circ$  [90].

During times of higher solar activity, the GCR dose decreases, but daily doses can increase significantly as a consequence of SPEs. In figure 3.7 an example of the effect of an SPE is depicted. The dose is measured by the Martian Radiation Environment Experiment (MARIE) aboard the 2001 Mars Odyssey spacecraft, orbiting Mars since 2001. The event was measured shortly after the solar maximum of 2001 and indicates a temporary increase of the dose rate by a factor of about 50 [107]. These high dose rates for solar events were also measured by the radiation experiment aboard the Mars Curiosity rover, which noted a total dose of  $1.2 - 19.5\text{ mGy}$  added per event [107].

An interesting mission profile that can be simulated in SPENVIS is an areostationary orbit, similar to a geostationary orbit but around Mars. These types of orbits are considered to be interesting for future telecommunications between Earth and Mars bases [70]. Because of the difficulties of entering an areostationary orbit compared to LEO, these missions would have a duration longer than one year. Hence, a mission duration of 10 years was chosen. The settings for the mission and orbit are depicted in figure A.9. For this simulation, a shorter representative time period was chosen, since there is less variability in the orbital parameters. For the solar fluence and for the TID simulations, the same settings as described in section 3.2.1 were set. A major difference between the results of the simulations for Mars concerning the TID and the results represented in figure 3.4 for LEO can be noticed: only solar protons were modelled by SPENVIS for an areostationary orbit. This is because there are no radiation belts present around Mars. In fact, Mars does not have an internally generated magnetic field like Earth, but instead is surrounded by a magnetic field induced by solar winds [98]. One of the consequences hereof is that Mars is much less protected against GCRs. However, GCRs are not included in the SPENVIS SHIELDDOSE TID simulation, thus the same approach as described in section 3.2.1 was used to model the GCR particles by means of MULASSIS. The settings for the GCR model are depicted in figure A.10. Notice that these settings differ from figure A.4. For an areostationary orbit, no magnetic

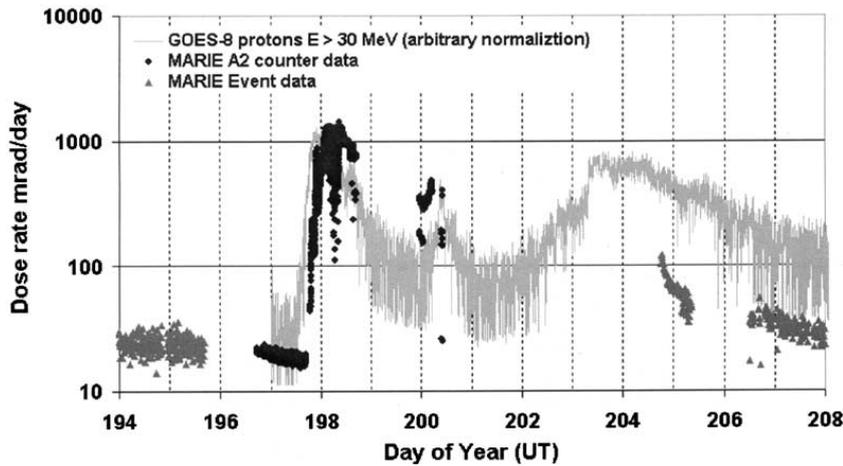


Figure 3.7: Dose rate measured by MARIE during an SPE that took place in July 2002 [107].

shielding can be chosen and a propagation factor has to be set. Without propagation, the GCR flux is taken constant throughout the heliosphere, which, according to the SPENVIS help files, is precise enough. As mentioned before, instead of mission epoch a solar minimum time period was chosen, which results in higher GCR fluxes. The difference in flux in function of energy level between the 1996 solar minimum and the chosen mission epoch is depicted in figure 3.8.

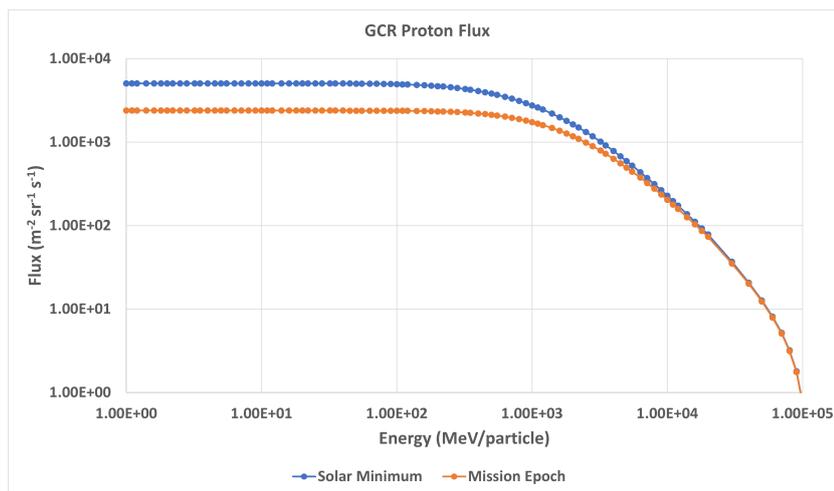


Figure 3.8: Comparison between the GCR <sup>1</sup>H isotope fluxes for a 10 year Mars orbiter mission during either the 1996 solar minimum or the current mission epoch solar maximum.

The results of both the SHIELDOSE and the MULASSIS simulations are depicted in figure 3.9 and show that, compared to LEO, the total dose due to solar protons per year has increased by about two orders of magnitude. The dose caused by GCR protons per year has increased by about two orders of magnitude as well. Additionally, the dose caused by helium ions was included. As mentioned before, in MULASSIS the dose for different types of isotopes is calculated separately. According to the SPENVIS reports, both helium isotopes contribute an approximately equal dose, although the <sup>3</sup>He particles should be 10 times less prevalent in solar winds [8]. The contribution of heavier ions decreases quickly and is not depicted. Previously, a yearly measured dose of about 13.58 rad due to GCRs was mentioned for a low altitude areocentric orbit, which agrees with the simulated total dose range for 10 years caused by GCR protons and helium ions of about 120 – 160 rad, depending on shielding. Note that these results were obtained for a 10 year mission during solar maximum activity for solar protons and solar minimum for GCRs, which is a conservative approach.

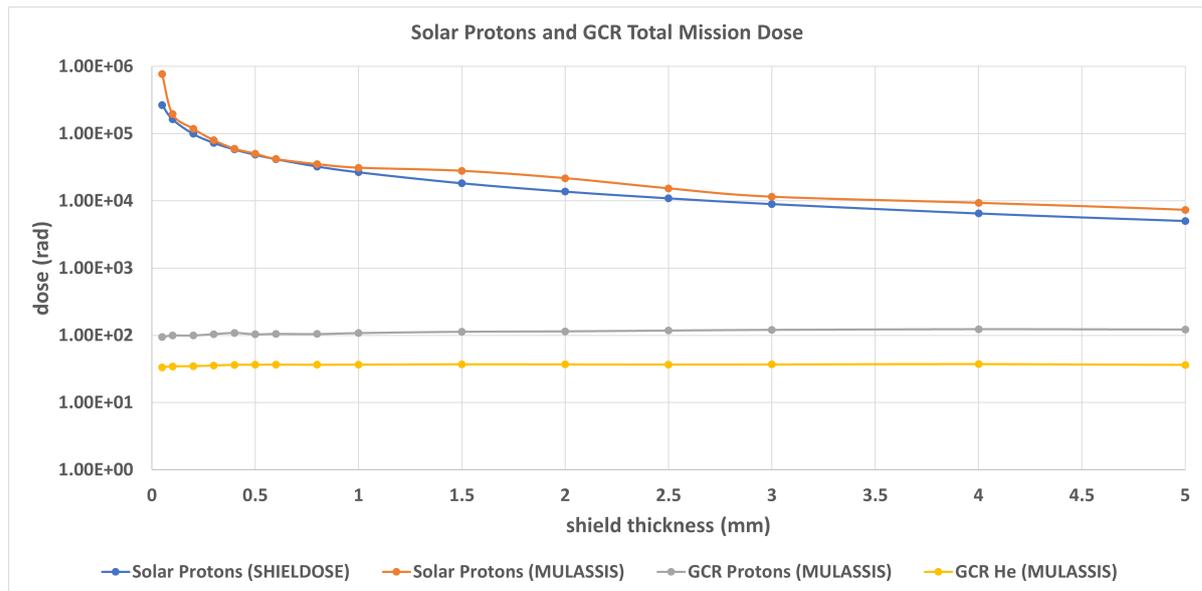


Figure 3.9: TID in function of shielding depth according to SPENVIS simulations for a 10 year mission in an areostationary orbit using both SHIELDDOSE and MULASSIS.

### 3.2.3. Conclusion

In table 3.3 the TID caused by a combination of trapped particles, solar protons and GCR protons for both mission profiles simulated above and for different shielding thicknesses is given. The values were retrieved from the SPENVIS report files. For the trapped protons, trapped electrons and solar protons, the results from the MULASSIS simulations were used, since these values were more conservative in almost all cases. In case of the Mars mission, GCR helium ions are included as well. Referring to the sensor datasheet, the FGD-03F has a range up to 50 *krad* [89]. This means that the sensor should be suitable for short LEO missions in the worst conditions, even if they pass through the SAA and there is only very limited shielding being used. Without much shielding, the total dose for a 10 year areostationary orbit would however exceed this limit. This indicates that for these missions a minimum level of shielding is necessary. As mentioned before, typical CubeSat shielding thickness is 2 *mm*. Taking this as a reference, the minimum dose the dosimeter should be able to withstand is about 22 *krad*. However, higher values are preferred as they allow more flexibility for future deep space missions with regard to shielding. Hence, a more strict minimum of 120 *krad* is chosen for the system requirements.

Table 3.3: Overview of TID in function of shielding thickness for a one year ISS-like orbit and for a 10 year areostationary orbit.

Mission	Thickness (mm)	Trapped (rad)	Solar (rad)	GCR (rad)	Total TID (rad)
ISS	0.05	1.78e5	419	1.43	1.79e5
	0.2	3.66e4	68.3	1.60	3.67e4
	2	3.69e3	9.16	1.87	3.70e3
Mars	0.05	0	7.67e5	128	7.68e5
	0.2	0	1.19e5	135	1.19e5
	2	0	2.18e4	151	2.19e4

### 3.3. Thermal Limitations

It would be beneficial that no active thermal control systems are needed when integrating the sensor in a satellite, since this would limit the applicability of the sensor and affect overall mission cost and complexity. To get a general idea, a few heat balances were constructed to determine the required operating temperature range for the dosimeter payload under design. Several assumptions were made with regard to the heat balance in space as to keep the complexity to a minimum, see section 3.3.1.

#### 3.3.1. Heat Balance

For the sensor in a circular orbit around a planet in our solar system, the following terms have to be taken into account for the heat balance: internal power generation, the solar radiative flux, the infrared radiative flux of the planet, the solar flux reflected by the planet called Albedo and the amount of heat generated by the sensor. Only the cuboid sensor in space is considered, with dimensions  $5 \times 5 \times 0.9 \text{ mm}$ , resulting in an external surface area  $A$  of  $68 \text{ mm}^2$ , and an average power usage,  $P$ , of  $20 \text{ mW}$  [15, 89]. The sensor is covered in black paint with a solar absorptivity  $\alpha$  of 0.96 and an infrared emissivity  $\epsilon$  of 0.88 [59]. The other surface areas in the heat balance, indicating the part of the sensor that experiences for example solar flux, depend on the orientation of the sensor and are determined on a case-by-case basis. The solar radiative flux at a distance of one astronomical unit from the Sun, which is roughly the distance from Earth to the Sun, equals  $1366 \text{ W/m}^2$ . Radiative fluxes at other distances from the Sun can be calculated by multiplying this flux with the ratio of the distances to the Sun squared. The distance between Mars and the Sun is  $1.5237 \text{ AU}$ . Radiative fluxes are indicated by  $J$ . The Bond albedo factors  $Al$  of Earth and Mars are 0.306 and 0.25 respectively. The Albedo flux at a certain orbit is then calculated using equation 3.1, with  $R_{planet}$  and  $R_{orbit}$  the radii of the planet and the orbit respectively. The infrared radiative flux calculation is shown in equation 3.2. The radii  $R$  of Earth and Mars are  $6371$  and  $3389.9 \text{ km}$  respectively. The surface temperatures  $T$  of Earth and Mars are  $255$  and  $210 \text{ K}$  [69]. Using these values, the heat balance as shown by equation 3.3 was constructed, where  $\sigma$  represents the Stefan-Boltzmann constant.

$$J_{Albedo} = Al * J_{Sun} * R_{planet}^2 / R_{orbit}^2 \quad (3.1)$$

$$J_{infrared} = \sigma * T_{planet}^4 * R_{planet}^2 / R_{orbit}^2 \quad (3.2)$$

$$A_{external} * \epsilon_{infrared} * \sigma * T_{sensor}^4 = J_{Sun} * A_{Sun} * \alpha_{Sun} + J_{Albedo} * A_{Albedo} * \alpha_{Sun} + J_{infrared} * A_{infrared} * \epsilon_{infrared} + P_{internal} \quad (3.3)$$

Just as for the SPENVIS simulations, two different orbits were considered. One is an ISS-like orbit around Earth, the other is an areostationary orbit. For each orbit, two solar illumination cases were taken into account. In one case, the sensor is illuminated by the Sun on one side and receives Albedo and planetary radiation on the other side. This means that surface areas for solar, Albedo and infrared radiation all equal  $5 \times 5 \text{ mm}$ . In the other case, the sensor finds itself in the shadow of the planet, is thus not illuminated by the Sun and only receives infrared radiation from the planet. Additionally, the power consumption of the sensor is either  $20$  or  $0 \text{ mW}$ . Using this information, the steady-state temperature of the sensor in each of these cases was calculated using the heat balance. Note that temporal variations due to the continuous variation in conditions were not taken into account.

#### 3.3.2. Conclusion

The results of the calculations for each separate case are shown in table 3.4. If the system is required to operate in all specified conditions without any thermal control, the most extreme values should be used to define a required operating temperature range. Although this range of over  $300 \text{ K}$  might seem excessive, it makes sure that the dosimeter can operate in extreme temperatures with ease and can be deployed in even more extreme conditions with only little thermal control measures.

With regard to the FGD-03F, with an operating temperature range of  $253 - 358 \text{ K}$ , the table shows that for most cases, the sensor does not need extensive thermal control measures [89]. For an ISS-like orbit, temperature can be manipulated when needed by powering on or off the sensor. Another option is to provide the sensor with some louvers, which keep the heat in during unlit conditions and keep the heat out during lit conditions. Insulation can also be an effective approach for the overall cooler areostationary orbit. Finally, in real-life, these sensors would be part of a larger satellite system. The

sensor can then be placed either close or far from other power dissipating components to keep the sensor temperature within its limits. In conclusion, the sensor should be able to function on most types of space missions with very little to no thermal control required.

Table 3.4: Overview of sensor temperatures calculated from the heat balance equation 3.3 for different situations.

Orbit	Power (mW)	Illumination	Temperature (K)	Range
ISS	20	sunlit	373.1	X
		shadow	294.0	V
	0	sunlit	341.7	V
		shadow	192.3	X
Areostationary	20	sunlit	317.1	V
		shadow	277.7	V
	0	sunlit	254.7	V
		shadow	66.6	X

### 3.4. FlatSat

As explained on the website of Space Inventor, the FlatSat is basically "a large motherboard where satellite avionics modules can be installed and connected as if it was inside a real satellite" [55]. This testbed, depicted in figure 3.10 and available for testing at the European Space Research and Technology Centre (ESTEC) facilities in Noordwijk, is basically a flattened out CubeSat with different types of electronic connections. The setup is designed to make it possible to test avionics units, payloads, software components and demonstrate complete satellite operations without having to construct a mission specific prototype. This allows for more straightforward functional testing of components and complete satellite setups earlier in the design phase with easy access to all the subsystems [55]. This testbed was used to design an interface between an FGDOS and a small satellite system, thus demonstrating that the dosimeter can be deployed as a satellite payload and, consequently, can be used for future space missions.

The different subsystems on the FlatSat interact via CubeSat Space Protocol (CSP), which is essentially a network protocol similar to the TCP/IP protocol widely used to communicate over the world wide web [46]. Each subsystem on the network, called a node, can send and receive data to other nodes on the network. Interaction with the network can be done via either the ground station, depicted above the FlatSat in figure 3.10, or a computer. These can be connected to the testbed via D-subminiature connectors that make use of the CAN standard on top of which CSP is implemented, resulting in the computer being just another node on the FlatSat system network. Instead of using the connectors, there is also the possibility to connect to the network via the telecommunications subsystem using antennae and radio-frequency signals. All the explanations below are using the computer terminal as an example, but the exact same operations can be performed using the ground station terminal. Several modules developed by Space Inventor were already mounted on the testbed by ESTEC, including two On-Board Computers (OBCs) in a single enclosure, a Telemetry, Tracking and Command (TT&C) unit and a Power Conditioning and Distribution Unit (PCDU).

Different types of ports are available to which external payloads can be attached. The testbed itself has, for example, debugging ports for both CAN and I2C. The OBC itself has an Harwin M80 connector with the possibility to connect devices via CAN, I2C, USB or UART. However, the FGDOS can only communicate via SPI, which is not available on the testbed or any of the mounted modules. The signal coming from the sensor thus has to be converted to a FlatSat compatible signal. Keeping in mind that, in the future, multiple sensors might have to be connected, it was decided to convert the signal to I2C, since this standard allows to connect 128 slave devices when using 7-bit addressing, and even more when 10 bit addressing is being used. Also note that a sensor chip that uses I2C instead of SPI is currently being developed by Sealicon. For conversion of the signal between SPI and I2C, an Arduino microcontroller was used, since this was readily available and the Arduino software has libraries for both SPI and I2C communication standards. Additionally, in a private conversation, CERN researchers

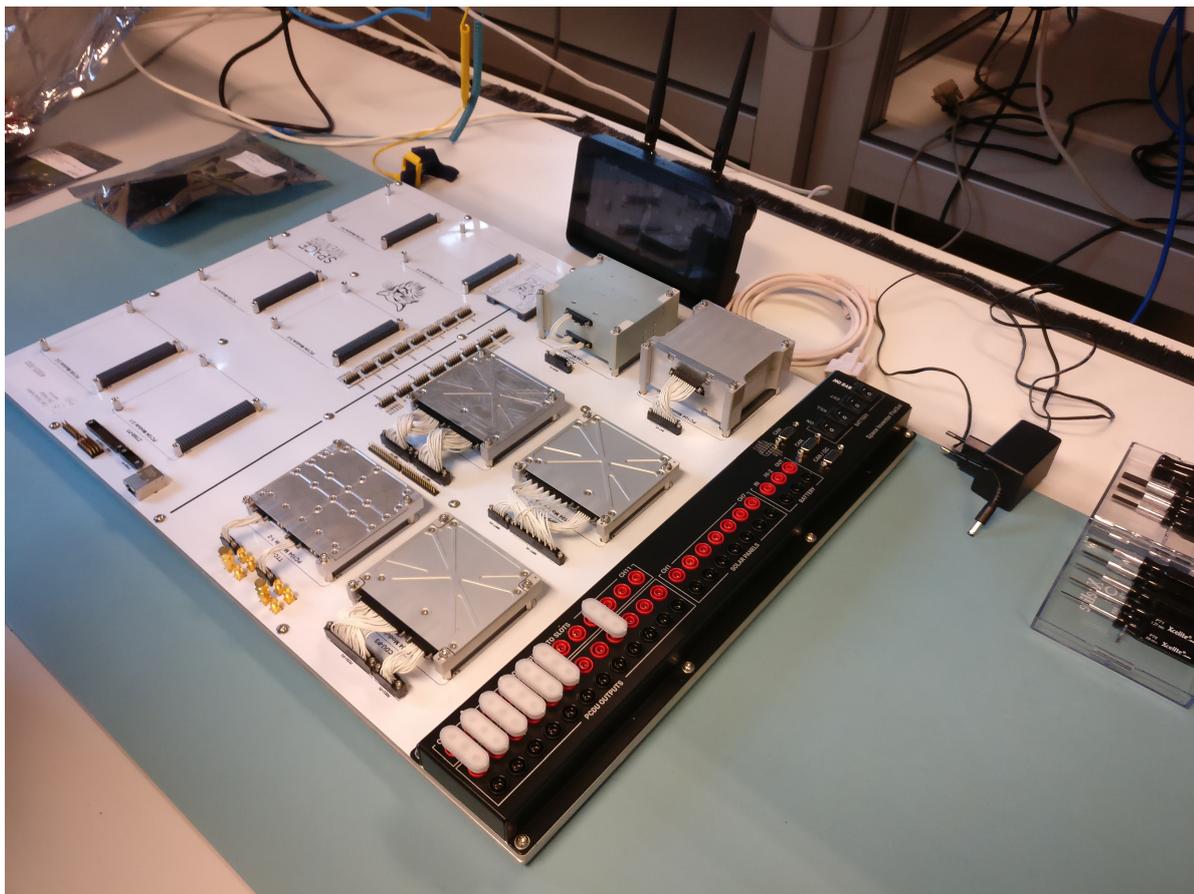


Figure 3.10: Picture of the FlatSat with basic avionics units installed and the ground station, all provided by Space Inventor.

stated that they also make use of an Arduino to develop FGDOS systems, which makes it easier to exchange code and expertise.

Because of the lockdown measures that were in place during the execution of this thesis as a consequence of the COVID-19 pandemic, the FlatSat test setup was not always readily available. To avoid major loss of time in setting up the software and communication with the sensor, an extra development board was used, i.e. the SAMV71 Xplained Ultra Evaluation Kit by Microchip Technology. This board can run the same operating system as the OBC and has several pins available that allow connection with the Arduino and the FGDOS. Since the setup of the operating system and the physical connections is similar to that of the FlatSat OBC, only the setup of the OBC itself is discussed. In a later phase, this board was also used to monitor the sensor chip's power consumption, see chapter 4.

### 3.4.1. FlatSat Software

The FlatSat modules run a Real-Time Operating System (RTOS) that can be programmed to complete mission specific tasks. To interact with the FlatSat OBC, additional software tools are needed. Space Inventor provides several Linux programs to facilitate this interaction with the FlatSat and its subsystems, the first one being the command shell Satctl and the second one being Spaceboot. How to program the OBC using these tools is explained in the following paragraphs.

#### Satctl

To explain the command shell's functionalities, the process of connecting a computer with a Linux operating system to the FlatSat network is described. It is assumed that the physical connection has been realised and the computer has an active connection via the CAN line using the SocketCAN package. After installing SatCtl, the computer can be initialised on the FlatSat network by entering the command shown in listing 3.1 in the Linux terminal.

```
1 satctl -n0 -v1 -ccan0
```

Listing 3.1: Terminal command to start up Satctl.

The computer is now part of the network as node 0, using parameter version 1 and CAN interface 0. The parameter version defines the communication protocol for the parameter system and the last option selects which communication interface to use. Next, the commands shown in listing 3.2 can be used to retrieve a parameter value from a specific node. The first command downloads a list of parameter definitions from a specified node to the computer or ground station node, and the second command then retrieves the current value for that specific parameter. More commands are available and can be found in the documentation provided with the testbed.

```
1 list download <node> [timeout] [version]
2 get <parameter name> [@<node>]
```

Listing 3.2: Satctl commands to retrieve parameter values.

### Spaceboot

The next tool provided by Space Inventor, called Spaceboot, is used to upload a software image to a flash storage slot of a specified subsystem on the network. Typically, the flash memory of a subsystem is divided into multiple slots and each slot can store a different firmware image. It is possible to upload an image to a slot whilst a system is running from a different slot. The code in listing 3.3 gives an example on how to upload a new image to the OBC via the terminal on the computer. The parameter version and computer node are defined as before. The two OBCs can be found on nodes 12 and 13 on the network. The `-r` option tells the OBC from which flash slot to restart and how many times it should reboot from this slot. The `-f` option is used to define which binary file to upload to which slot. Finally, the OBC is restarted from the same slot to which the new firmware was uploaded, in this case slot 1.

```
1 spaceboot -v1 -n0 12 -r2,1 -f1,<firmware-file>.bin -r1,1
```

Listing 3.3: Spaceboot terminal command example.

### FreeRTOS

The firmware binary file is build from several programming code libraries, which include the parameter version, the CSP library and user written source files that define the tasks to be performed by the RTOS installed on the OBC. This type of operating system is used in applications with very strict time constraints, thus where the software has to adhere to important deadlines for the execution of tasks. The operating system being used here is FreeRTOS, an open source RTOS. A short explanation on how to create a task for the OBC to perform is given here, but for a detailed explanation the reader is referred to the FreeRTOS manual, which can be found on the FreeRTOS website [40]. Additionally, it is assumed that the user has basic programming skills, especially C, and knows how to use build software, such as Meson, to generate binary files from larger software projects. The user first has to create source and header files and make sure they are included by the project build software. In the new source file, the user first creates a new task by calling upon the task-create Application Programming Interface (API) function, an example of which is given in listing B.1. The first argument of the API function refers to the actual task having to be performed. Other arguments then define, amongst others, the stack depth of the task and its priority. The task itself includes an infinite loop which contains the code to be run repeatedly by the OBC. In the example shown in listing B.1, a string is printed to the standard output with a 5 second interval.

After creating this task and including it in the build system, the binary file can be generated and uploaded to the OBC via Spaceboot. Next, the computer is connected to the network via Satctl. The standard output of the OBC can be monitored via the Satctl command `stdbuf 12`, which results in the standard output of node 12 now directly being printed into the terminal of the computer.

### 3.4.2. FlatSat & Arduino

A first step in integrating the dosimeter payload into the FlatSat was connecting the FlatSat to an Arduino microcontroller over I2C. The physical connection was made by connecting the clock and data pins of the Arduino to those of the OBC, as well as the ground pin, so both systems operate at the same reference voltage level. To program the Arduino the 'Wire' library was used, which includes a set of functions to set up and control I2C connections. The FreeRTOS library includes several general Input/Output (I/O) functions as well as specific I2C functions.

#### Arduino

Arduino is an open-source platform that provides both hardware and software tools. For this project, the Arduino UNO microcontroller was used, together with the Arduino Integrated Development Environment (IDE) for programming the microcontroller. A schematic of the hardware is depicted in figure 3.11, with on the top right the Serial Clock (SCL) and Serial Data (SDA) pins for I2C communication. Using the 'Wire' library the Arduino can be set up to make communication possible with the functions shown in listing 3.4. First, the address of the Arduino, here used as a peripheral, is being set. Next, it is defined which functions the Arduino has to execute when receiving write or read request via the I2C lines. The 'receiveEvent' and 'requestEvent' functions can then contain any code the user wishes to execute when communication by the controller, in this case the FlatSat OBC, is started. In these functions the read and write functions of the 'Wire' library can be used to read data from or write data to the I2C lines.

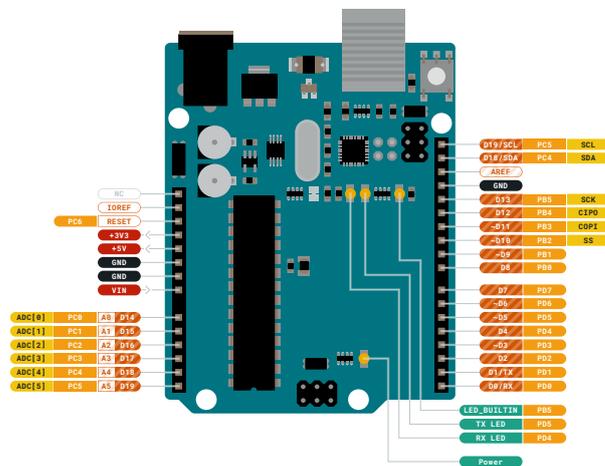
```

1 void setup(){
2   Wire.begin(0x25); // set peripheral address
3   Wire.onReceive(receiveEvent);
4   Wire.onRequest(requestEvent);
5 }
    
```

Listing 3.4: Example of Wire library usage for Arduino.



ARDUINO  
UNO REV3  
STORE: ARDUINO.CC/UNO-REV3



<ul style="list-style-type: none"> <li><span style="display: inline-block; width: 10px; height: 10px; background-color: black; margin-right: 5px;"></span> Ground</li> <li><span style="display: inline-block; width: 10px; height: 10px; background-color: red; margin-right: 5px;"></span> Power</li> <li><span style="display: inline-block; width: 10px; height: 10px; background-color: green; margin-right: 5px;"></span> LED</li> <li><span style="display: inline-block; width: 10px; height: 10px; background-color: grey; margin-right: 5px;"></span> Internal Pin</li> <li><span style="display: inline-block; width: 10px; height: 10px; background-color: brown; margin-right: 5px;"></span> SWD Pin</li> </ul>	<ul style="list-style-type: none"> <li><span style="display: inline-block; width: 10px; height: 10px; border: 1px solid black; margin-right: 5px;"></span> Digital Pin</li> <li><span style="display: inline-block; width: 10px; height: 10px; border: 1px solid black; margin-right: 5px;"></span> Analog Pin</li> <li><span style="display: inline-block; width: 10px; height: 10px; border: 1px solid black; margin-right: 5px;"></span> Other Pin</li> <li><span style="display: inline-block; width: 10px; height: 10px; border: 1px solid black; margin-right: 5px;"></span> Microcontroller's Port</li> <li><span style="display: inline-block; width: 10px; height: 10px; border: 1px solid black; margin-right: 5px;"></span> Default</li> </ul>	<ul style="list-style-type: none"> <li><span style="color: red;">▲</span> <b>MAXIMUM</b> current per I/O pin is 20mA</li> <li><span style="color: red;">▲</span> <b>MAXIMUM</b> current per +3.3V pin is 50mA</li> </ul>	<ul style="list-style-type: none"> <li><b>VIN</b> 6-20 V input to the board.</li> <li>NOTE: CIPO/COPI have previously been referred to as MISO/MOSI</li> </ul>
--	---	--	--

ARDUINO.CC  
Last update: 13/10/2021

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. To see a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1888, Mountain View, CA 94042, USA.

Figure 3.11: Schematic of the Arduino UNO microcontroller [4].

## FreeRTOS

The setup of an I2C connection in FreeRTOS is depicted in listing B.2. At first an I/O descriptor is defined that contains the pointers to the I/O read and write functions. This descriptor is then linked to the I2C driver descriptor, after which the communication channel is enabled and the slave address is set. In the infinite task loop, the user can send and receive data bytes using the depicted write and read functions, each of them triggering the respective I2C read and write functions on the Arduino defined earlier. The process is straightforward for sending and receiving bytes, but some functionalities have to be added for transmitting integers or other multiple-byte variable types. FreeRTOS receives the bytes and puts them into memory in the received order. The user can, for example, define an array of two bytes in which incoming bytes are stored and reconstruct an integer from those bytes using bitwise operations. Alternatively, the user can just use a pointer to the integer's memory location. When FreeRTOS then has to fetch the integer's value from memory, it reads the two bytes in the order they were placed there by the read function. The Arduino library cannot send or receive integers and the user has to use arrays of bytes and re- or deconstruct multiple-byte variable types from those arrays. The bytes itself are transferred most significant bit first as per I2C standards.

### 3.4.3. Yet Another Mission Control System

The open-source software framework Yet Another Mission Control System (Yamcs) can be used to communicate with satellites, payloads or ground stations. This mission control system makes it easy to send and receive telecommands and telemetry to and from a satellite. For more information the user is referred to the Yamcs website [91]. The framework is used to demonstrate that radiation measurements from the dosimeter can be received on the ground station via the FlatSat as if it were a satellite orbiting in space. Setting up a connection between the FlatSat and the computer or ground station has been discussed before. However, some additional steps are required in order to use Yamcs. First, a CSP server application is activated on the ground station. This is needed to connect to the satellite's CSP network because the tools provided by Space Inventor, like Satctl, are not being used. Additionally, a CSP bridge has to be setup to translate the CSP packets to User Datagram Protocol (UDP) packets that are understood by the mission control software. The correct settings for these setups were provided by ESA engineers. Next, after installing Yamcs according to their website, the telemetry settings have to be changed in order to understand how the CSP data packets are constructed. This is done via the Extensible Markup Language (XML) code shown in listing B.3. In this file, the data bytes that belong to the variables retrieved from the dosimeter are defined, thus allowing Yamcs to understand the data packets it retrieves from the FlatSat. As can be seen from the listing, each packet contains the necessary CSP identifiers, including priority, destination and encryption. The telemetry packet contains five variables: sensor number, temperature, sensor frequency, reference frequency and window factor. What these variables represent is explained in section 3.6.

After setting up the local Yamcs server using, for example, the ground station, the user can open a browser and connect to the local server which, if using the default settings, can be found at `http://localhost:8090`. Here, a user interface can be found with different functionalities: plotting and storing of telemetry data, sending telecommands or managing alerts. An example output of the dosimeter data is shown in section 3.7.

## 3.5. Floating Gate Dosimeter

A concise overview of the dosimeter chip, programming and basic functionalities is given in this section. However, for an extensive overview on characteristics, pin layout, register settings and more, the reader is referred to the Sealicon website and dosimeter datasheet [89]. A functional overview of the sensor chip is given in section 2.3.2. In the current section, the process of setting up the sensor and changing the necessary registers is discussed. Additionally, the process of converting the frequency values obtained from the sensor to radiation doses is explained, based on the information retrieved from the sensor datasheet.

### 3.5.1. FGD-03F Setup

It is assumed that the sensor has power and is connected to a controller, for example an Arduino, using SPI. For details with regard to this part of the setup, see section 3.6. Additionally, a clock signal is applied at pin CK. Sealicon advises a signal of  $32.768\text{ kHz}$ . Next, using the SPI connection, registers can

be set to put the sensor in high or low sensitivity mode, select the internal charge pump for recharges and set the measurement window. This measurement window determines for how long the counters in the sensor count the frequency pulses coming from the dosimeter. Depending on the register settings, the measurement window is determined by either an amount of pulses at the CK pin or by the duration of a pulse signal at the CK pin. To avoid possible timing issues with applying a constant signal, the first approach is taken in this project. As a result, the sensor frequency can be calculated by means of equation 3.4:

$$F = \frac{\text{Register Count}}{\text{Window Pulses}} * F_{CK} = \text{window factor} * F_{CK} \quad (3.4)$$

Here,  $F$  represents frequency and the register count signifies the number of pulses measured by the counter, which is stored in the sensor's registers as an 18 bit value. The register value can either be the amount of pulses generated by the dosimeter or the amount of pulses generated by the reference oscillator, depending on which register is read. As explained before, the external clock frequency  $F_{CK}$  is used as a reference by the sensor to determine the duration of the measurement window. The window pulses factor is set by the user and determines the length of the measurement window. Next, the threshold and target frequencies can be set. When the sensor frequency goes below the threshold a recharge is triggered and the floating gate is being recharged until it reaches the target frequency. If needed, this automatic recharge procedure can be switched off and controlled manually. Since the target and threshold values are only five bit values, only the five most significant bits of the frequency registers are used for comparison. For added precision, the settings can be changed, resulting in the three most significant bits of the frequency values being disregarded for the comparison and the five next bits being used. This is needed when, for example, short measurement windows are being used. As a consequence of this short window, the frequency registers hold low values and the five most significant bits might all be zero. For the FGD-02F, the threshold and target are actually eight bit values and they cannot be shifted. The recharge voltage generated by the internal charge pump can be set as well. Although higher voltages result in shorter recharge times, and thus less time lost on recharging, it was noticed that the highest voltages result in such short recharge times that the sensor does not have enough time to compare the sensor frequency output to the target value, thus resulting in overshoots. Furthermore, the sensor is set to ignore frequency measurements during SPI communication, since these communication signals might interfere with the measurements.

### 3.5.2. Post-Processing

After the initial setup, the counter registers can be read and the frequencies can be calculated using equation 3.4. The amount of received dose between two timestamps  $t$  and  $t-1$  can then be determined via equation 3.5:

$$\text{Dose} = \frac{F_{\text{sensor},t} - F_{\text{sensor},t-1}}{\text{Sensitivity}} \quad (3.5)$$

Because the sensor is sensitive to temperature, temperature compensation might be necessary to accurately determine radiation doses. This can only be done if the sensor has been temperature characterised before. Using this characterisation a relation between the sensor and reference frequencies is established in function of temperature. This relation is then assumed to be valid under radiative conditions as well. Finally, either a lookup table or a linear approximation of this relation can be used to determine the sensor frequency at the reference temperature from the sensor and reference frequency at the measured temperature. Since the sensor used in the tests was not temperature characterised, this method has not been tested in this thesis work.

## 3.6. Arduino

After setting up the connection between the FlatSat and the Arduino microcontroller, the next step is to connect the microcontroller to the radiation sensor. To get familiar with Sealicon's sensor functionalities, a Sealicon demo board has been used, which consists of the chip mounted on a Printed Circuit Board (PCB) and a USB adapter, which allows the user to plug the sensor directly into the computer. Sealicon also provides a software application that plots the data coming from the evaluation kit and allows the user to change the sensor's settings. The demo board uses an FGD-02F dosimeter, which differs slightly from the one described in section 2.3.2. The main differences are that this version does not include an internal charge pump and cannot be put into standby mode. The pinouts for both sensors are

the same, except for the pins related to the charge pump. Besides some small differences in register settings, as explained in section 3.5, the program explained below is valid for both types of sensors.

To connect the chip on the board to the Arduino via the sensor's SPI interface, some adaptations to the evaluation kit had to be made. Most importantly, the Controller Out Peripheral In (COPI) and Controller In Peripheral Out (CIPO) lines of both sensors inside the chip are all connected to each other, which makes it impossible to connect it to the Arduino microcontroller. Hence, the traces on the PCB connecting these pins have been cut. Unfortunately, only the traces for one sensor could be cut, so only one sensor of the two present in the chip was used. Next, some extra wiring had to be added to make sure all the necessary pins for chip functionality could be connected to their respective Arduino pins. For SPI communication, the COPI, CIPO, Chip Select (CS) and Serial Clock (SCK) pins of the Arduino and the sensor have to be connected. Next, the sensor on the chip also needs to be provided with an external clock signal. For this purpose the Arduino's digital I/O pins can output a Pulse Width Modulation (PWM) signal. Finally, the sensor's power and ground pins should be connected to the board's 5 V supply and ground pins respectively.

### 3.6.1. Software Setup

An Arduino program basically consists of two functions. First, a setup function can be used to initialise the necessary functions, followed by an infinite loop in which the microcontroller performs the user's defined tasks. For interaction with the sensor, the setup function has to initialise four different functionalities: serial communication with the computer, clock signal generation, SPI communication with the sensor and set up of the sensor's registers. The code of this setup is depicted in listing B.7.

The first command sets up communication with the computer using a specified baudrate. Next, a digital pin for the generation of a clock signal is chosen by the user and set to an output pin. The easiest way to generate a PWM signal is to switch the pin output between low and high with a delay in between, thus effectively generating a signal with a specified frequency and duty rate. However, since this is not the only piece of code running in the main loop, timing would become very difficult and the clock signal would vary over time. Furthermore, high frequency signals are not possible using this method. It is thus more precise to program the Arduino's processing chip's registers so that an accurate PWM signal is being generated in the background whilst the other tasks can continue to run. For a detailed explanation about the registers the reader is referred to the ATmega328P chip's datasheet [6]. Basically, by setting the registers, different digital pins can be connected to the internal timers, which in turn use the chip's system clock as a reference to generate timing and counting functionalities. This practise is, however, not perfect either. The system's clock relies on the internal oscillator to generate a clock signal and this oscillator has a calibration accuracy of only 14%. After the desired pin is connected to the timer, the frequency and duty cycle of the PWM can be set by setting the appropriate prescaler and compare register values. The resulting PWM has a frequency of 31.25 kHz, which differs only slightly from the frequency recommended by Sealicon.

The third part is initialising the SPI communication by means of the SPI library provided by Arduino. The SPI settings function sets the basic protocols for communication: SPI clock frequency, most significant bit first data transfer and SPI mode. The maximum communication frequency that can be used for SPI is 5 MHz, as specified in the sensor datasheet. There are a total of four SPI communication modes available, based on the different options available for the clock's phase and polarity settings. The polarity determines whether the clock signal is active low or active high and the phase determines whether bits should be read at the falling or rising edge of the clock signal. Next, the CS pin is set to output and put to high. Pin 10 is also set to output, since this pin could otherwise cause the microcontroller to think it is a peripheral on the SPI network instead of a controller.

The final function shown in the setup function in listing B.7 is a user defined function in which the sensor is set up according to the specific use case. This function is depicted in listing B.5 and uses basic SPI read and write functions to set the registers, and thus the settings, of the sensor chip. The read and write functions are depicted in listing B.6. They basically set the first two bits of the register addressing byte to 10 or 01 depending on whether a read or write commands needs to be performed. The mapping of the variable names to the hexadecimal register addresses, definitions of pins and constant values used by the Arduino programs are all shown in listing B.8. The setup of the sensor follows the same procedure described in the sensor datasheet [89].

### 3.6.2. Software Loop

After the software setup, the main Arduino program loop was defined. This loop is repeated continuously by the Arduino. It takes care of function flags and can be used, if needed, to print data to the computer for debugging purposes. Every time the I2C connection triggers an Interrupt Service Routine (ISR) and requests data, the Arduino retrieves information from the sensor and stores this into a data array. This data array is always constructed in the same manner, otherwise the different parts of the system would be unable to communicate clearly and consistently. As mentioned in section 3.4.3, this data array contains five variables for a total of 13 bytes:

1. sensor number, 1 byte
2. temperature, 2 bytes
3. sensor frequency, 3 bytes
4. reference frequency, 3 bytes
5. recharge register, 1 byte
6. window factor, 4 bytes

The sensor number is useful for when more than one sensor is connected. The temperature and frequencies are measured by the sensor. These frequencies are the number of counts registered by the internal counters multiplied by the window factor, as depicted by equation 3.4. The recharge register is an 8 bit value of which the most significant bit indicates whether or not a recharge is ongoing, whilst the other 7 bits state how many recharges have been performed.

## 3.7. From Dosimeter to Mission Control

Both the connections between the FlatSat and the Arduino and the Arduino and the dosimeter have been put forward. Also the link between the FlatSat and Yamcs has been explained. Putting everything together, it is possible to follow the dosimeter frequency outputs life via Yamcs.

First, the OBC was programmed so that its main task is retrieving sensor data at regular intervals. This task sends read and write requests over I2C to the Arduino. It also interacts with Yamcs by sending CSP data packets over the network to the ground station via the RF antennae. The complete OBC code is displayed in listing B.4. The extra delays throughout the code are added to avoid that there are too many simultaneously open CSP connections. Next, the Arduino receives these signals and, via an I2C ISR, sends the same request over SPI to the dosimeter, which takes the necessary actions. Sensor data then take the same route back, until eventually they can be visualised in Yamcs. The complete Arduino code is displayed in listing B.7 Attention has to be paid to the deconstruction and reconstruction of the variables send over the different interfaces. When, for example, a float variable is transferred, this variable is then decomposed into four bytes, which are send between the components using the appropriate communication protocol. Whilst the I2C protocol sends data most significant bit first, the SPI bus does not necessarily follow this setting. For instance, to avoid possible clashes in what bits and bytes are send first or last, the bytes that the Arduino receives via protocol X are first used to reconstruct the associated variables using the protocol's settings and conventions. Next, these variables are used to determine the data structure to send via protocol Y to another part of the system, again using the protocol's specific characteristics. The final step is then to start the CSP server and bridge on the ground station, as well as the Yamcs application. Once connected to the Yamcs server via the browser, the user can look at the data from the sensor being retrieved in real-time.

A screenshot of a plot of the data gathered by Yamcs is shown in figure 3.12. The data arrays being send from the sensor to Yamcs are structured as described in section 3.6.2. From the browser, all the data updates can be followed. Alerts and telecommands can be included, but this should then be added to the rest of the data exchange chain as well. This proves that the sensor chip can be included in a typical small satellite system. However, this does not say anything about actual radiation detection. The characterisation of the sensor using proton beam radiation tests is discussed in chapter 4.

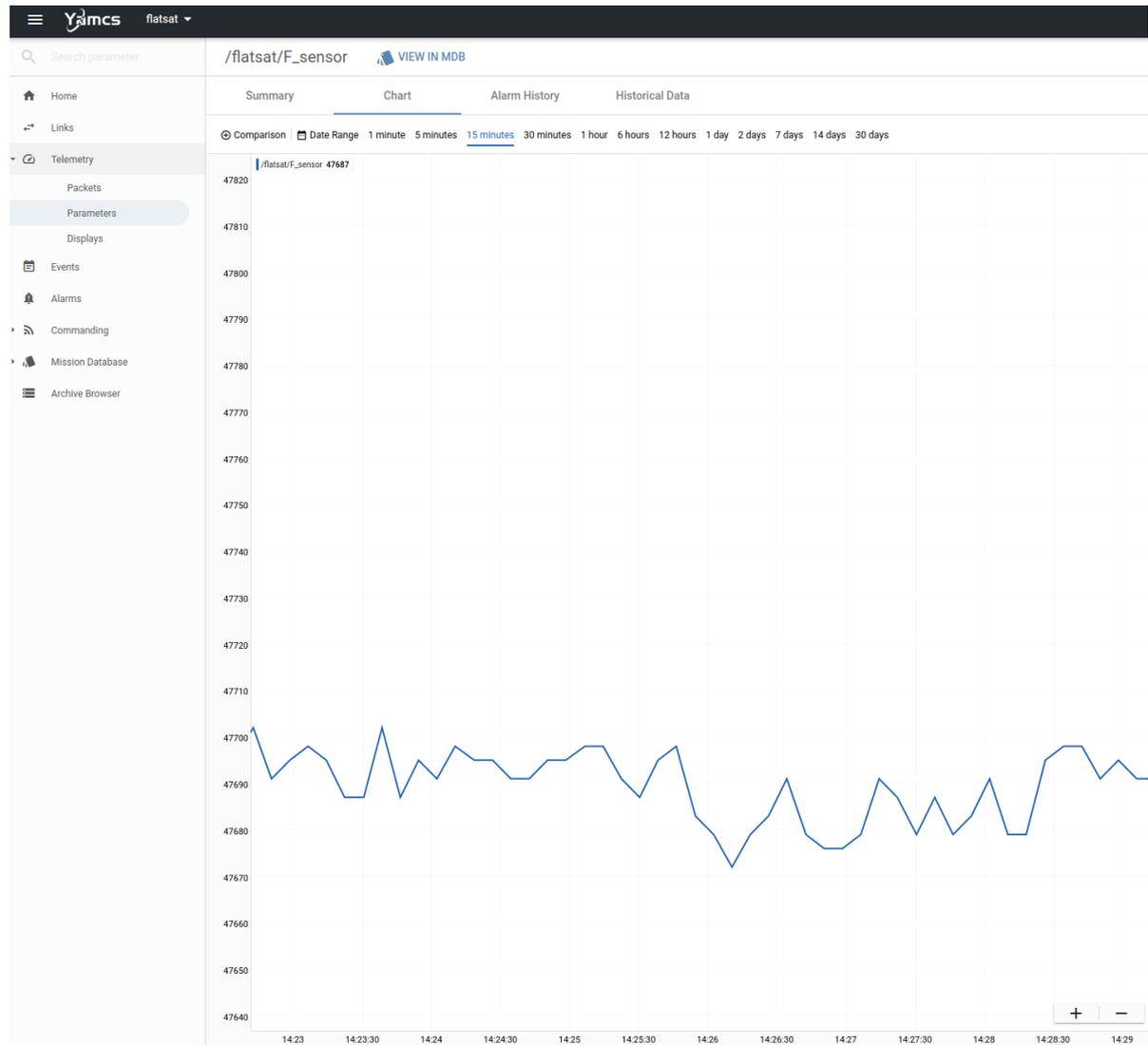


Figure 3.12: Life output of Yamcs showing the dosimeter frequency in function of time. Data go from the dosimeter to the ground station via the Arduino and the FlatSat CSP network.

# 4

## Radiation Testing

The floating gate dosimeter chip developed by Sealicon was proven to be compatible with a typical CubeSat setup, as discussed in chapter 3. Several papers exist discussing the results of radiation tests carried out with the dosimeter [15, 23, 26]. However, there are still some open questions with regard to its behaviour. Additionally, the sensor should be well characterised if it will be used on future projects like the Lunar Zebro. Hence, radiation tests were performed to better understand the dosimeter's features. First, the necessary preparation for the tests are described in section 4.1. This includes a detailed test plan and a description of the test facilities and test setup. Next, the results of the radiation tests are discussed in section 4.2.

### 4.1. Test Preparation

#### 4.1.1. Test Facility

Different radiation test facilities were contacted, each having their own specifications with regard to particle type, energy and flux. However, because of difficulties with planning and supplementary regulations due to the pandemic, not all facilities were available for testing. Eventually, radiation tests were performed at HollandPTC on the 16<sup>th</sup> of December. HollandPTC is a medical facility near the TU Delft campus. Their cyclotron can generate a proton beam with an energy between 70 – 250 MeV. Depending on the setup, different field shapes, sizes, homogeneities, energy levels and dose rates can be provided. Via calibration, the dose rate in air at Device Under Test (DUT) can be related to the beam current for the specific setup. This relation is linear and the calibration values can thus be used to relate any desired dose rate to the required beam current settings. Using formula 4.1, these dose rates can be further converted to flux levels. The specific setup for the experiments is shown in figure 4.1, in which, from left to right, the following elements can be seen: DUT, two collimators, dual ring setup with the thin ring being used, lead foil for scattering, NextQ beam monitor and beam outlet. The scattering sheet, ring and collimators were used to get the desired beam shape and profile. This was a 4x4 cm<sup>2</sup> square beam profile with a 96% homogeneity. Each test run can only take up to about 12 min because of safety restrictions related to the cyclotron. The cyclotron also suffered from RF trips, meaning that the RF alternating electric field failed and the cyclotron had to be started again. Only a beam energy of 100 MeV was used, resulting in a beam of 70 MeV at DUT.

For each test run, HollandPTC provided beam current, flux at target and experiment duration. The dose rate in silicon can be calculated using equation 4.1. Here, LET is expressed in MeV cm<sup>2</sup>/mg, the flux  $J$  in particles/cm<sup>2</sup> and the last factor is a conversion factor with implicit units J/MeV \* mg/kg, resulting in a dose rate with units J/kg/s. The fluence  $J'$  and the total dose  $D$  can be calculated by integrating equation 4.1 over time, see equation 4.2. The LET from different types of ions impinging on a material can be derived from the Stopping and Range of Ions in Matter (SRIM) package, which is a widely used Monte-Carlo simulation based software package that is available for free online [96]. Another option is to consult the tables provided by the National Institute of Standards and Technology [78].

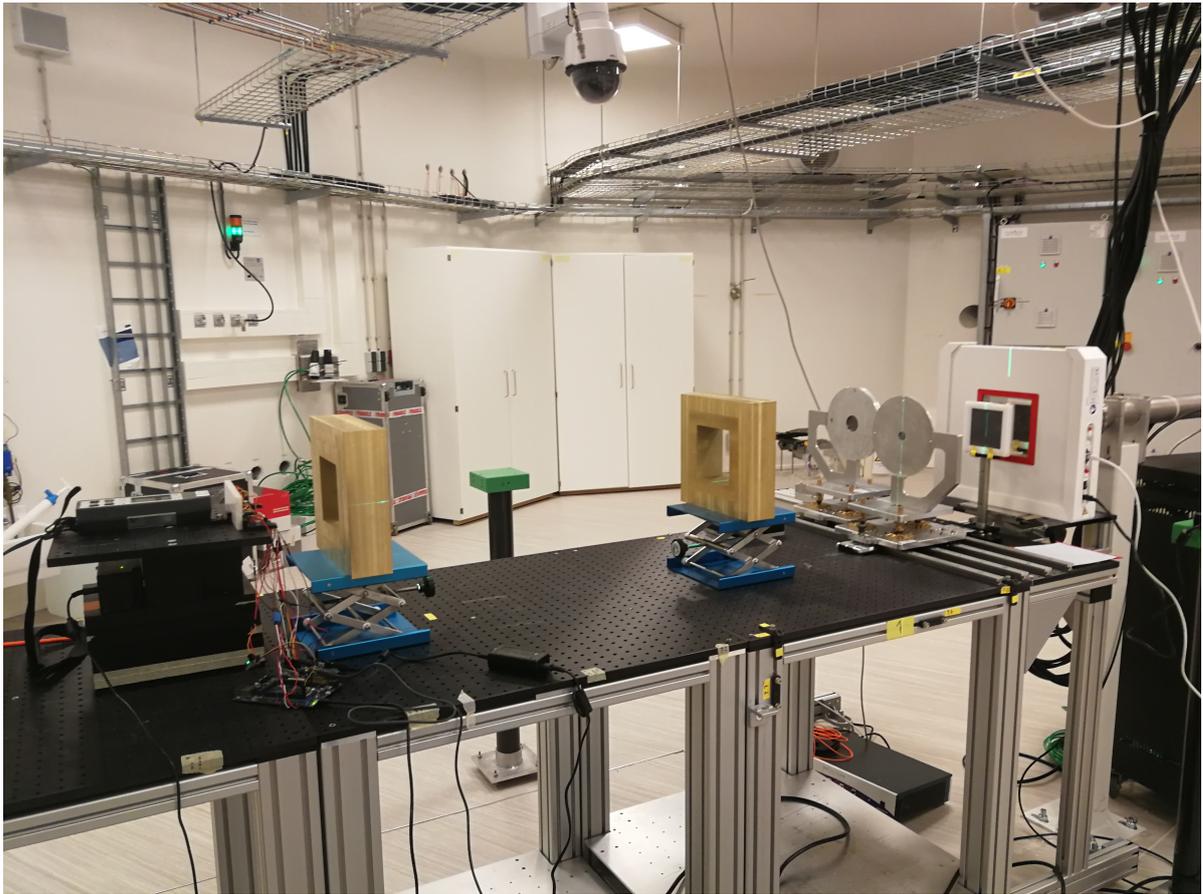


Figure 4.1: Beam arrangement and test setup positioning for the radiation tests at HollandPTC on the 16<sup>th</sup> of December.

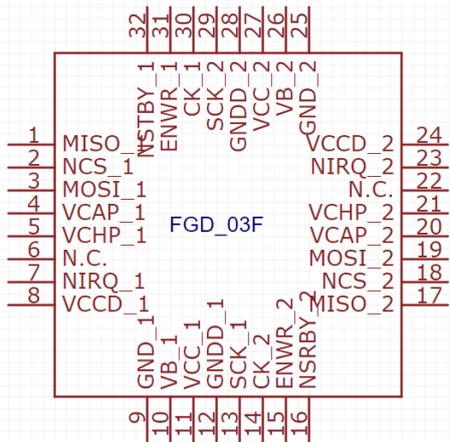
$$Dose\ rate = LET \times J \times 1.6e - 5 \quad (4.1)$$

$$D = \int_t Dose\ rate = LET \times J' \times 1.6e - 5 \quad (4.2)$$

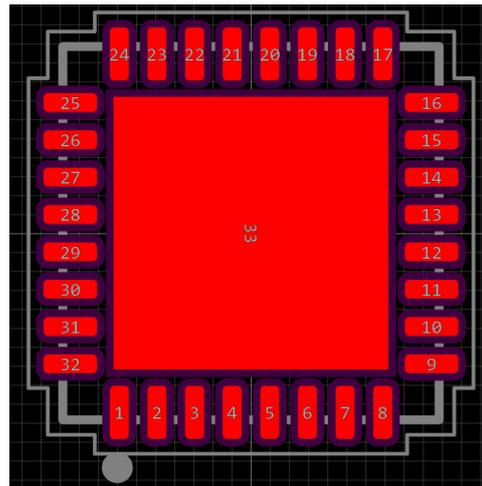
#### 4.1.2. Printed Circuit Board Design

For the radiation tests, the demo board, which was used for interaction with the FlatSat, is not ideal, because not all pins can be accessed separately. Additionally, for future projects like the Lunar Zebro, the sensor chip by itself will be used. Remember that the chip comes in a QFN32 package and, in order to interact with the chip, it has to be mounted on a PCB. Although a more complete PCB could be designed that includes noise limiting capacitors and other electronic components, a board was designed that only includes the chip itself. This allows for more flexibility in the future, since different test setups might require different additional electronic components.

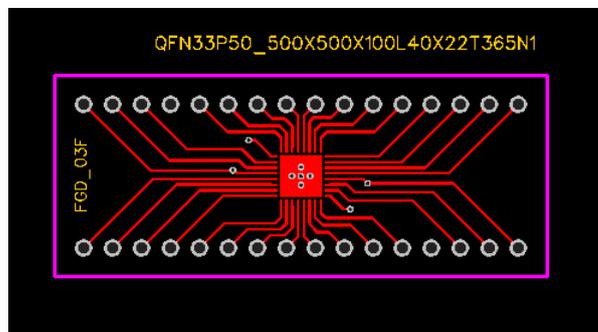
The PCB was designed using EasyEDA, a free browser based tool for electronic design automation. The first step is to define the FGD-03F chip as a component, since it is not available in the existing component libraries. A schematic of the sensor chip is depicted in figure 4.2a. This schematic is coupled to the footprint depicted in figure 4.2b. The footprint determines how the surface of the PCB should be constructed, i.e. which areas have to be covered by solder mask so that the chip can be soldered onto the PCB. Next, a schematic of the PCB is drawn. Since no other components are included on the board, this schematic is simply the chip with the two GND pins connected to the ground. Finally, based on this schematic, the board is designed. A board with only one copper layer could be used, since no copper lines have to cross each other. However, to minimise noise effects, the return current should be allowed to follow a path similar to the path of the supply current. If this is not possible, the



(a) FGD-03F schematic representation in EasyEDA.



(b) FGD-03F footprint representation in EasyEDA.



(c) PCB top view.

Figure 4.2: Construction of the FGD-03F PCB in EasyEDA.

supply and return current create a loop that can act as an antenna when radio frequency signals are being used, causing electromagnetic interference. Consequently, PCBs are generally designed with a dedicated ground layer. This ground layer also diminishes the impedance between different grounding points, thus reducing the possibility of ground loops that could cause interference.

The top view of the two-layer PCB is shown in figure 4.2c. The grounds of the two sensors in the chip are connected to the ground plane. The bottom central pad of the chip, the thermal pad, is also connected to the ground plane using vias, as advised by Sealcon. The digital grounds are connected to each other but are kept separate from the analogue ground plane. The trace connecting the digital grounds on the bottom layer is surrounded by a wide non-conducting area to avoid cross-talk between the analogue and digital circuits. The traces itself should not have 90° angles to avoid additional capacitive noise. Although this PCB proved to work fine, several design choices could be improved. First, the thermal pad is a relatively large area. When soldering paste is applied, the large amount of solder tends to form a bulge as a consequence of surface tension. This could cause the chip to be tilted and not well connected to all pads after soldering. To avoid this, the thermal pad should be divided into several smaller ones. Next, the pads of the pins could be extended outwards, since this makes it easier to visually check that all pins are correctly soldered. Furthermore, the silkscreen on the top layer should indicate where pin number 1 is located and should include a marking around the chip's position, making it easier to manually place the chip correctly onto the board. After design, the board was ordered from Eurocircuits, a Belgian company specialised in PCB manufacturing. The chip was finally mounted onto the board using the facilities of the Faculty of Mechanical, Maritime and Materials Engineering at TU Delft.

### 4.1.3. Test Setup

The complete test setup is depicted in figure 4.3. The Arduino was used to control the FGD-03F sensor chip and steer a MOSFET switch. The switch was used to turn off the power supply to one of the sensors. This allowed for testing with a sensor completely switched off, an operating mode called passive mode. The sensor itself also incorporates a standby mode, during which power consumption is reduced to a minimum. The difference between passive and standby mode is further explained in section 4.2.4. The Arduino was connected to a prototype FGD-03F sensor provided by ESTEC as well. The working principles of this prototype chip are exactly the same, but it contains only one sensor instead of two. Additionally, it comes in a DIL16 packages, thus no PCB design is needed to use it on a breadboard. For the remainder of this thesis, the prototype is referred to as sensor 1 and the QFN chip as sensor 2. There were two reasons for including this prototype. First, it provided additional measurement data that were used to verify the results with of newer QFN package. Second, during programming tests, the sensors of the QFN chip were recharged and then discharged again. Although the guidelines in the sensor datasheets were carefully followed, this discharging resulted in a sensor breaking down, after which it gave an output frequency of  $0\text{ Hz}$ . It is not exactly known what causes this, but it might be the result of the charge injector pump breaking down, caused by a voltage spike or electrostatic discharge and resulting in the floating gate being shorted to the ground. The SAMV71 board was used to control two INA219 current sensors. Each current sensor can measure supply voltage and voltage drop across a shunt resistor, from which the sensor chip's power consumption can be derived. The sensors were connected in series between the Arduino's  $5\text{ V}$  power output and the FGD-03F chips. Around the chips some additional red coated capacitors can be noticed. These decoupling capacitors were used to limit noise in the circuit. Their capacitance is  $100\text{ nF}$ , based on the capacitors used in the demo boards provided by Sealicon.

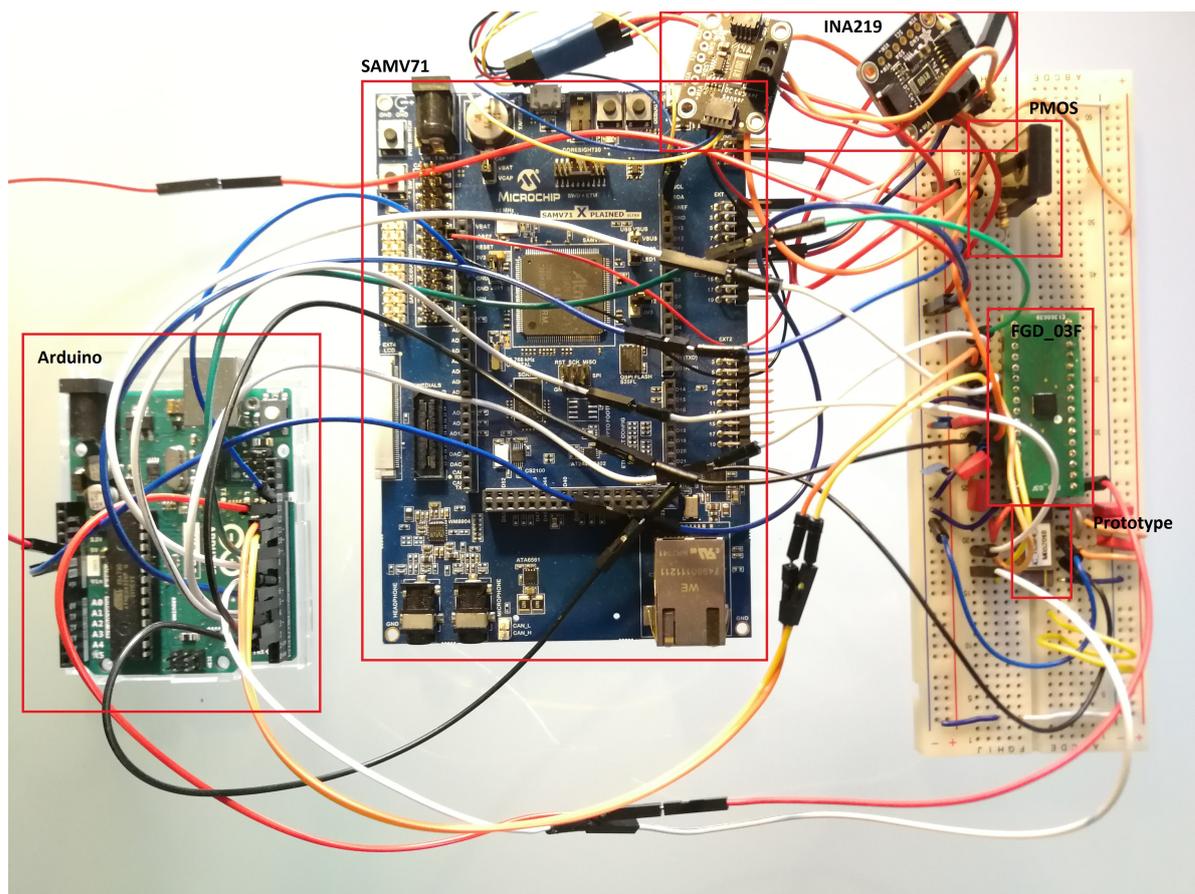


Figure 4.3: Test setup for the radiation tests at HollandPTC.

Both the Arduino and the SAMV71 boards were connected to a computer via USB. The programming of both microcontrollers is explained in section 4.1.4. Both controllers can be powered via their USB connection. However, since the dosimeter output depends on the input voltage, the Arduino was powered via a 10 V power plug. This activates the Arduino's internal voltage regulator [5]. As a result, a more stable 5 V voltage was supplied to the sensor chips. This is further discussed in section 4.2.7. The controllers sent data to the computer that were collected using Python scripts, see section 4.1.5. Since nobody is allowed in the test room during the tests, the computer was controlled remotely via TeamViewer. The remote control was done from within the cyclotron control room. Finally, the breadboard with the dosimeters was aligned with the beam, as already shown in figure 4.1. A schematic representation of the complete setup is shown in figure 4.4.

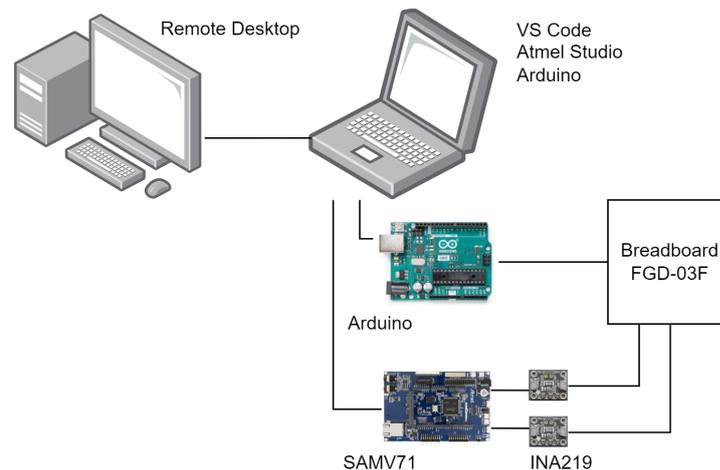


Figure 4.4: Schematic representation of the setup for the radiation tests at HollandPTC.

#### 4.1.4. Arduino & SAMV71 Programming

Several pieces of code that can be used to interact with the FGD-03F chip have been highlighted before. These programs were slightly adapted for the radiation tests. The main differences are that there is no interaction with the FlatSat over I2C and that the SAMV71 board is used for current measurements. The Arduino program is depicted in listing B.9. The setup of the pins, PWM clock signal and SPI were put into separate functions. To collect data from the sensor, a loop could be used that checks for new data at regular intervals. However, in order to gather new data as fast as possible, the chip's interrupt pins were connected to the Arduino's ISR enabled pins. In the program, functions were defined that are executed whenever the chip pulls an interrupt pin low. These functions collect a data array including the following variables: user defined sensor number to be able to distinguish between the two sensors, sensor temperature, sensor frequency, reference frequency and recharge count. In the main loop of the program, three additional functionalities were added. First, if no interrupts are used, the main loop collects data at intervals of two times the window length plus 10%, as advised by Sealicon. Second, because the interrupts do not work during sensor recharges, the program detects, by means of the recharge register, when a recharge is ongoing and enters a data collecting loop until the recharge is finished. Third, the option was added for the user to use commands to change the sensor's settings. By means of single letter commands, the user can put the sensor in passive mode, standby mode, change sensitivity or perform a sensor reset, thus allowing for real-time control of the sensor chip's settings. The command function is depicted in listing B.10.

The SAMV71 code to interact with the INA219 sensors is depicted in listing B.11. For more details with regard to the current sensors, the reader is referred to the sensor datasheet [54]. The program starts with setting up a serial connection with the computer in order to make data transfer possible. Next, I2C communication is started between the controller and the current sensors, and the sensors are configured by setting the registers as defined in the datasheet. The controller then retrieves bus and shunt voltages from the sensors in an infinite loop. The data array sent to the computer contains the bus and shunt voltages of both sensors.

### 4.1.5. Data Acquisition

In sections 3.6 and 4.1.4, it is explained how the microcontrollers send data to the computer and how this stream of data is constructed. These data have to be gathered and stored by the computer consistently, so they can be processed and analysed afterwards. Python scripts were written in Visual Studio Code, one for interaction with the Arduino and one for interaction with the SAMV71 controller. These scripts read incoming information, add timestamps to the data arrays and store them into Comma Separated Values (CSV) files. The script for interaction with the Arduino also includes the possibility to send data back to the controller. This is necessary to be able to control the settings of the FGD-03F sensor chip, as described in section 4.1.4.

### 4.1.6. Test Plan

A choice had to be made with regard to which parameters and sensor characteristics to test, since only a limited amount of testing time was available. The primary goal was to verify the setup's functionalities and determine the sensor's sensitivity, both in low and high sensitivity mode. Other sensor properties that could be tested include:

- Influence of flux on sensitivity
- Influence of particle beam energy on sensitivity
- Influence of TID on sensitivity
- Influence of particle type on sensitivity
- Determine optimal target and threshold frequencies
- Temperature characterisation
- Verify passive and standby mode
- Effects of recharge process and recharge rate on sensor behaviour
- Maximum TID before breakdown
- Influence of the angle of incidence of particles on sensitivity
- SEEs
- Effect of different window factors on sensor behaviour
- Effect of different clock frequencies on sensor behaviour

This list is not exhaustive but includes the most interesting properties. For the first tests, the same energy level was used and only the flux was varied. An overview of the test runs and their main goal is given in table 4.1. Each test run took about 10 minutes. In each test run, both sensors, i.e. the one sensor in the FGD-03F chip that works labelled sensor 2 and the prototype labelled sensor 1, always had the same settings for sensitivity. A measurement window of 4096 pulses was used during all the tests. This is the shortest window that can be selected and was chosen as to retrieve as much data as possible. With a clock frequency of 31.25 *kHz* generated by the Arduino, this results in a measurement window of 131 *ms*. The chip thus counts the amount of pulses coming from the sensor for 131 *ms* and then does the same for the reference oscillator. For the first four tests, a recharge voltage of 16 *V* was used. It was noticed that this causes recharges to take about 30 *s* or more, during which no radiation measurements can be performed. In all later tests, a recharge voltage of 16.5 *V* was used, causing the recharges to take only about 10 *s*. The energy of the proton beam was 100 *MeV* for all the tests, resulting in an energy of 70 *MeV* at DUT. The dose rates and total doses in Si in the table were calculated using equations 4.1 and 4.2 with an LET of 7.614 *MeV/cm<sup>2</sup>/g*, which is the value given by SRIM for hydrogen ions, i.e. protons, with an energy of 70 *MeV* and silicon as the target material. Not all testing times are equal, because some tests were interrupted by RF trips or by the user. For the last experiment, the HollandPTC monitors malfunctioned and not all information is available. The lowest flux level was chosen so that, based on the sensitivity estimation in the sensor datasheet, the whole linear range of the sensor would be covered during a single test run in high sensitivity mode. The second flux level was chosen as to see the influence of recharges in high sensitivity mode, the third flux level to cover the linear range in low sensitivity mode and the highest flux level to see the effect of recharges in low sensitivity mode.

It should be noted that some of the aforementioned testing objectives cannot be performed at HollandPTC. Flux and energy levels can be changed in between test runs, but, due to the radioactivity in the test room, the test setup cannot be moved in between tests. Furthermore, since a cyclotron is being used, only protons are available. Temperature characterisation would require precise temperature

Table 4.1: Overview of the test runs for the radiation tests on the 16<sup>th</sup> of December at HollandPTC in Delft.

Test	Start	Energy (MeV)	Flux (#/s/cm <sup>2</sup> )	Rate (rad/s)	Time (s)	TID (krad)	Settings	Goal
01	19:04	70	1.71e6	0.209	600	0.125	HIGHSENS	sensitivity determination
02	19:16	70	1.71e6	0.209	600	0.250	HIGHSENS	sensitivity determination
03	19:31	70	7.07e6	0.863	600	0.768	HIGHSENS	sensitivity determination
04	19:44	70	7.07e6	0.863	600	1.28	LOWSENS	sensitivity determination
05	19:58	70	1.40e7	1.71	600	2.31	HIGHSENS	influence of flux
06	20:10	70	1.40e7	1.71	900	3.85	LOWSENS	influence of flux
07	20:36	70	3.69e7	4.50	538	6.27	HIGHSENS	influence of flux
08	20:56	70	3.69e7	4.50	600	8.97	LOWSENS	influence of flux
09	21:13	70	7.07e6	0.863	555	9.45	HIGHSENS	passive mode
10	21:35	70	7.07e6	0.863	600	9.97	HIGHSENS	passive mode
11	21:49	70	1.40e7	1.71	600	11.0	HIGHSENS	passive mode
12	22:09	70	1.40e7	1.71	559	11.9	LOWSENS	passive mode
13	22:25	70	7.07e6	0.863	567	12.4	HIGHSENS	passive mode
14	22:40	70	7.07e6	0.863	–	–	HIGHSENS	influence of TID

control and additional equipment. The main goals of these tests were then, as can be seen from table 4.1, to determine the sensor's sensitivity and test the influence of flux, TID and operating mode. After determining the sensitivities, the highest priority was put on testing the influence of flux, passive mode, beam energy and TID. However, an operating error resulted in most of the passive mode test data being stored without timestamps, thus making precise data analysis impossible. The TID mentioned in the table is the cumulative dose.

## 4.2. Results

In this section the results of the radiation tests are presented. In order to be able to analyse the data, they needed to be prepared in files of CSV format. This means, for example, removing invalid entries. Next, the files were analysed using Python scripts. In the following subsections, several properties of the sensor are presented based on the analysis of these files. The results of the tests were also discussed with researchers from CERN to verify correct interpretation of the results.

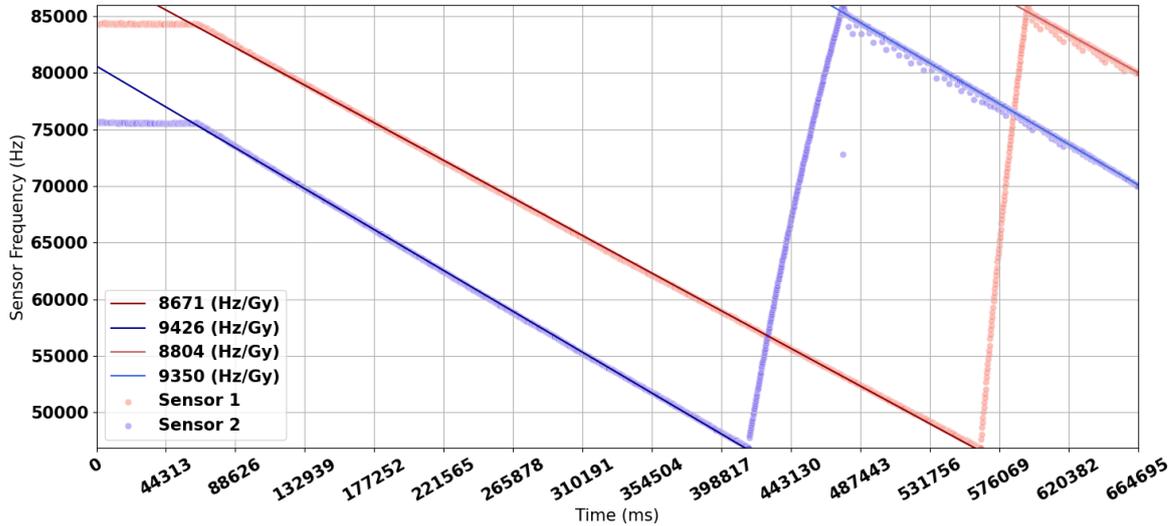
### 4.2.1. Sensor Sensitivity

During the radiation tests it already became clear that the sensitivities of the sensors were lower than expected. The sensor datasheet, provided by the manufacturer Sealicon, mentions a sensitivity of 10 kHz/Gy for the low sensitivity setting and 70 kHz/Gy for the high sensitivity setting. These values seem, however, optimistic. Both Danzeca et al. [26] and Brucoli et al. [13] reported sensitivities of between 25 – 35 kHz/Gy, determined using experiments with a <sup>60</sup>Co source. Danzeca et al. [26] performed experiments with a proton beam as well. Here, sensor sensitivity using a 230 MeV proton beam was measured to be 35 kHz/Gy and 7.6 kHz/Gy for high and low sensitivity mode respectively. These sensitivities went down to 24 kHz/Gy and 5.1 kHz/Gy when the researchers tested with a 60 MeV proton beam. It is, however, still not clear whether part this effect could have been caused by TID. In figures 4.5a and 4.5b, the sensor frequencies measured during test runs 03 and 04 are plotted in function of time. The sensitivity can be calculated by taking the endpoints of the experiment and calculating the difference, resulting in a value for  $\Delta F$ . Dividing this by the total dose then results in a sensitivity value in kHz/Gy. However, the sensor is assumed to be working in a linear range, meaning that a first order polynomial can be determined from the data points via linear regression methods. This approach eliminates the effect of noise that is inherent to the circuit and that could influence the endpoints. The function of this linear behaviour is depicted in equation 4.3, with  $F$  the frequency and  $t$  the time. The coefficient  $dF/dt$  and constant  $c$  in this equation can be determined using simple linear regression. Dividing the derivative  $dF/dt$  by the dose rate then results in a sensitivity in Hz/Gy:

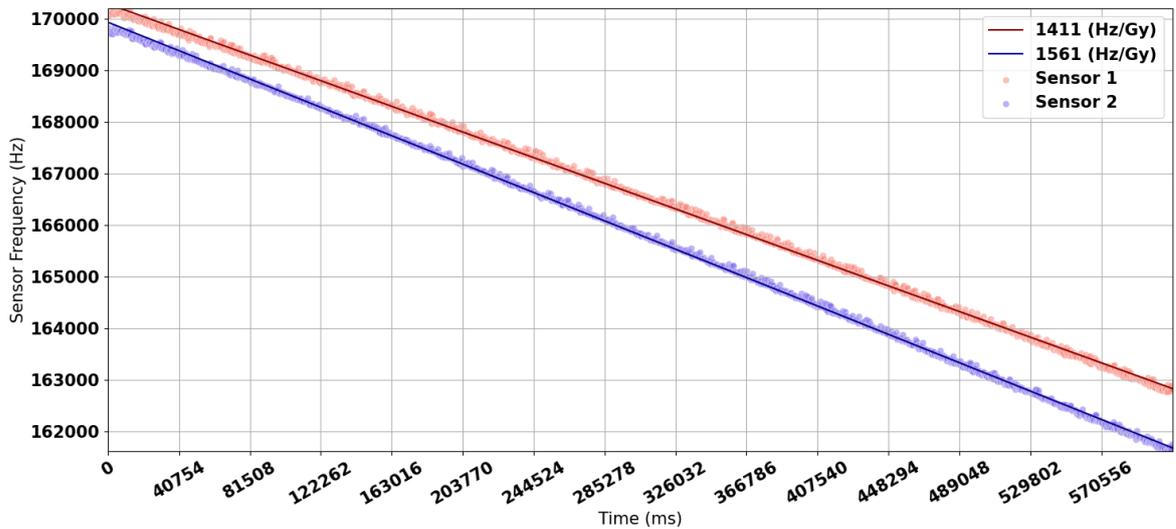
$$F = \frac{dF}{dt} * t + c \quad (4.3)$$

Linear regression lines are depicted in figure 4.5a, where the sensitivities calculated from the linear regression coefficients are shown in the legend as well. Note that this experiment was performed with

the sensor in high sensitivity mode, meaning that both sensors have a sensitivity of about a factor nine smaller than advertised by Sealicon. Compared to the sensitivities reported in literature for a  $60\text{ MeV}$  beam, the measured sensitivity is about a factor three smaller [26]. This was also noticed for the experiments in low sensitivity mode, see figure 4.5b. The linear regression lines in the figures all have p-values of 0 and R-squared values of over 0.999, indicating a very good fit of the linear regression models. In figure 4.5a a recharge of the sensor can be noticed. After the recharge, the frequencies show a larger spread, but this was the result of a programming error and not caused by the sensor.



(a) Test run 03 with the sensors in high sensitivity mode.



(b) Test run 04 with the sensors in low sensitivity mode.

Figure 4.5: Sensor frequencies in function of time with a beam energy of  $70\text{ MeV}$  at DUT, a flux of  $7.07e6\text{ \#/cm}^2/\text{s}$  and the sensors in high and low sensitivity mode. Linear regression lines are added, as well as the sensitivity calculated from the linear regression coefficients and dose rates. These plots correspond with test runs 03 and 04 respectively.

The temperature and reference frequency measurements from the sensors that correspond with figure 4.5a, and thus test run 03, are depicted in figures 4.6a and 4.6b. It shows that the temperature, measured by the sensor, is constant. However, it should be noted that the resolution of the internal temperature sensor is only  $1^\circ\text{C}$ , so small temperature variations are not registered. The reference frequency plot shows that there are small internal temperature variations present. Furthermore, it shows that the reference frequency values are not internally updated during recharges. The reference frequency

during test run 04 is depicted in figure 4.6c and shows that, in low sensitivity mode, the reference frequency needs time to settle to a steady value. This has no influence on the sensor frequency, but should be taken into account when temperature compensation is necessary. It should also be noted that, from time to time, the measurement values jump to an extreme value, for example 500 kHz, and back again. It is assumed that this is caused by SEEs and is not further investigated. A more significant event was noticed during test run 06. At the end of the experiment, the measurements for sensor 2 stopped without explanation, whilst the sensor and reference frequencies for sensor 1 showed irregular values. This could be caused by, for example, an electrostatic discharge in the sensor, but an exact cause is hard to determine. The issue was solved after a reset of the sensors.

#### 4.2.2. Flux & Total Ionising Dose Effects

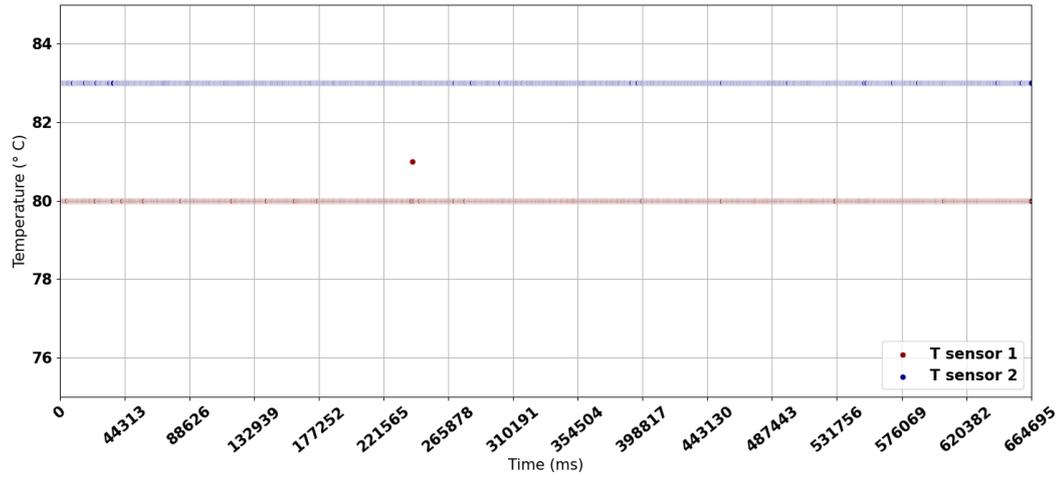
An overview of the sensitivities for different experiments is given in table 4.2. Here, the sensitivities are given for different flux levels, each time for both sensitivity modes. The TID that the sensors already received before each separate test run is also given too, since this influences the sensitivity. Because the temperatures were noticed to vary with maximum 1 °C and the sensors were not temperature characterised, meaning that no temperature compensation can be done, the temperature influence is not taken into account. For some experiments, only data for part of the linear range was available to determine the sensitivity, as noted in the comments in the table. For some fluxes multiple values are available. This is because the sensitivity calculation was split to determine the sensitivity both before and after an interruption. Such an interruption could be a recharge or an RF trip of the cyclotron. The duration of such an RF trip is about 5 min.

From literature it was already known that TID causes a degradation of the sensitivity as a result of oxide traps in the field oxide [15, 79]. These traps cause a reduction in the local electric field, thus leading to higher recombination rates. Additionally, interface traps might influence transistor transconductance, also resulting in a sensitivity drop. The influence of dose rate on the sensitivity is, however, less clear. Danzeca et al. [26] noticed that the sensitivity in the high sensitivity mode decreases with increasing dose rates and the other way around for the low sensitivity mode when testing with a  $^{60}\text{Co}$  source. This was also noticed by Alvarez et al. [3], who attributed this effect to the lower recombination times at higher doses. As mentioned before, Danzeca et al. [26] also performed tests with a proton beam. They observed that the sensitivity tends to decrease for higher dose rates, i.e. a decrease of 30% was noticed when increasing the dose rate from 500 rad/h to 30 krad/h with the sensor in low sensitivity mode. Bruccoli et al. [13] compared batch variability with dose rate variability and noticed that the variability between dose rates of 32 rad/h and 320 rad/h using a  $^{60}\text{Co}$  source is comparable to batch variability.

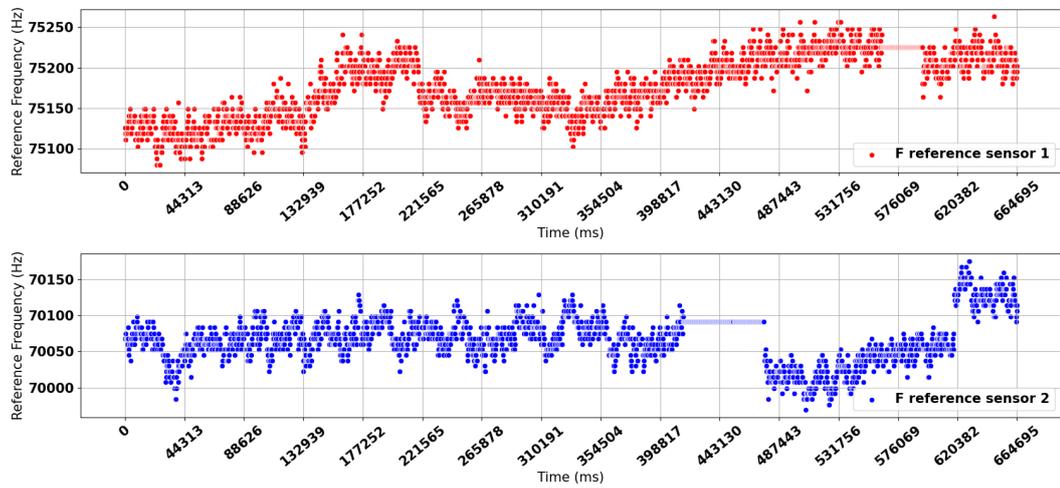
Table 4.2: Sensitivities of the sensors in Hz/Gy in function of flux, TID and sensitivity setting. The TID is the amount of radiation already received before the start of that particular experiment.

Flux (#/cm <sup>2</sup> /s)	Comment	S1	S2	TID	S1	S2	TID
		Low	Low	(krad)	High	High	(krad)
1.71e6	HIGH: part range				7339	7978	0.125
7.07e6	LOW: part range	1425	1577	0.768	8758	9521	0.250
	HIGH: part range after recharge				8893	9444	
					4090	4389	12.4
1.40e7	LOW: RF trip	1812	1744	2.31	8940	9905	1.28
		2258	2202				
3.69e7	HIGH: recharge and RF trip	1381	1617	6.27	8955	9684	3.85
	LOW: RF trip	1725	2017		8918	9641	
					9522	10189	

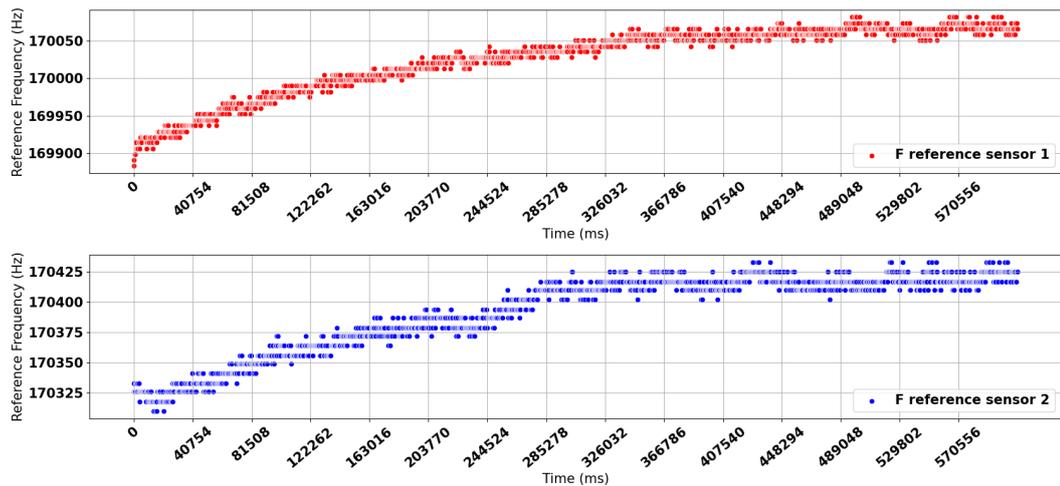
The experiments at HollandPTC were performed with the lowest flux first, as can be seen from table 4.1. From table 4.2 it can be noticed that, although the TID is higher for the higher fluxes, the sensitivities tend to increase with dose rate both for the high and the low sensitivity mode. This is not the case



(a) Sensor temperature measurements for test run 03



(b) Reference frequency measurements for test run 03



(c) Reference frequency measurements for test run 04

Figure 4.6: Sensor temperature and reference frequencies in function of time with a beam energy of 70 MeV at DUT, a flux of  $7.07e6 \text{ \#/cm}^2/\text{s}$  and the sensors in high and low sensitivity mode. These plots correspond with test runs 03 and 04.

for the highest dose rate, which might be caused by TID effects having a more significant influence at this point, although more experiments are needed to further clarify this phenomenon. The table also shows that the sensitivity after a recharge tends to be slightly lower, except for one case: sensor 1 at a flux level of  $7.07e6\#/cm^2/s$ . However, at this flux level, the sensitivities after the recharge were calculated based on part of the linear range only. This should not matter, since the frequency change over time should be linear, but it was noticed from other experiments that this is only an approximation and slight variations over the linear range are present. In figure 4.7 the sensor frequencies are given for test run 07. The part of the figure where the frequency is almost constant is the result of an RF trip. When the linear range before the recharge is split into two parts using the frequency level of the RF trip as a threshold, the resulting sensitivities for the higher and lower frequency range are 9035 and 8933 Hz/Gy respectively. This was similar for the other sensor, indicating that the sensitivity tends to be slightly higher during the upper part of the linear range. This could explain the increase in sensitivity after a recharge for the particular case highlighted before.

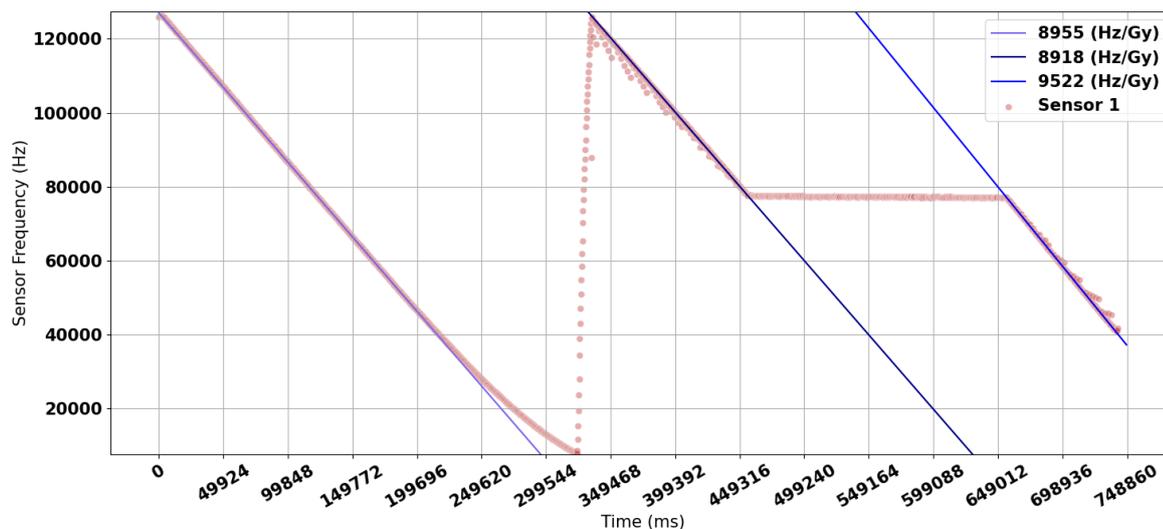


Figure 4.7: Sensor frequency in function of time with a beam energy of 70 MeV at DUT, a flux of  $3.69e7\#/cm^2/s$  and the sensor in high sensitivity mode. Linear regression lines are added, as well as the sensitivity calculated from the linear regression coefficients and dose rates. The plot corresponds with test run 07.

It was noticed that the sensor frequency gradually decreases during the RF trips, which can be seen on figure 4.7. This is probably caused by an annealing effect, i.e. the trapped holes in the tunnel oxide of the reading transistor anneal over time, thus causing a reduction in the electric field and a decrease in output frequency [13]. The sensitivity before and after an RF trip also differs. As explained before, the sensitivity during the lower part of the linear range tends to be slightly lower. However, when the linear range was split by a short outage of the cyclotron, the lower part of the linear range had a higher sensitivity. This could be caused by a short term annealing effect. Alvarez et al. [3] already noticed that, in between test runs, the sensitivity tends to recover after TID degradation, but the time in between their experiments was in the order of hours. Annealing of the trapped charges in the field oxide resulting in a partial recovery of the sensor's initial sensitivity has also been demonstrated by Danzeca et al. [26] by means of isochronal and isothermal annealing. Assuming that heating up the sensor for annealing to  $300^\circ C$  takes significantly longer than the 5 min RF trips, these papers thus all discuss relatively longer time periods for annealing. Nevertheless, short term annealing effects could be possible, as can be seen from figure 4.8. Here, the recovery of the threshold voltage for an N-channel MOSFET after irradiation is depicted, showing that initial recovery can occur within milliseconds.

The last test run, number 14, was performed to assess the influence of TID at a flux level of  $7.07\#/cm^2/s$ , see table 4.2. Danzeca et al. [26] reported a decrease in sensitivity of 21.6% with a TID of 11 krad using a 230 MeV proton beam. In the experiments at HollandPTC however, a decrease of about 54% was measured for a TID of 12.4 krad. In the future more experiments should be performed to verify the sensitivity decrease with TID and assess whether or not this decrease is particle energy dependent.

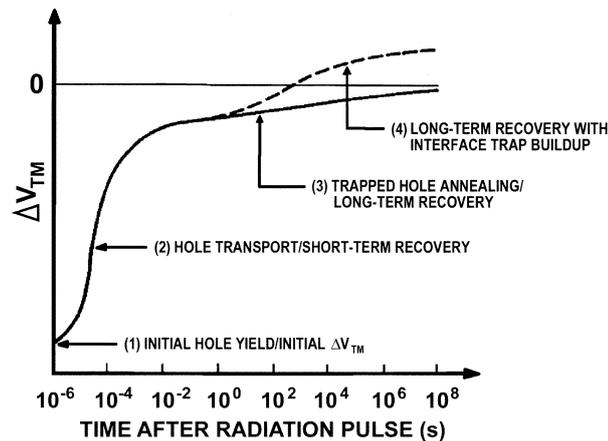


Figure 4.8: Threshold voltage shift and recovery of an irradiated N-channel MOSFET [79].

The fact that both sensors show a lower sensitivity than expected, despite coming from completely different batches, could indicate that there were problems with the test setup itself. The programming of the Arduino itself was verified using the demo board from Sealicon. Remember that in a first step, code was written to interact with the FGD-02F demo board, and that this demo board comes with a software application to read out the sensor. The output of this application, i.e. sensor frequencies, temperatures and other variables, and the output of the Arduino software were found to match, already indicating that the software works as intended. Furthermore, the output of the Arduino code was compared against the output of Arduino code written by CERN researchers for their development of FGD-03F based systems. These outputs matched as well. Additionally, the frequency output of the reference MOSFETs and of the sensors is what can be expected based on the sensor datasheet and previous research [13, 89]. Assume, for example, an error was made in the window factor calculation and the actual sensitivity is about  $35 \text{ kHz/Gy}$  for the high sensitivity mode [26]. This means that the total frequency change in the graphs is actually about a factor four larger. However, this implies that, in figure 4.7, the highest frequencies are over  $500 \text{ kHz}$ , which is far outside of the linear range, as indicated by figure 4.9. This would also increase the reference frequencies by the same factor, since these are calculated using the window factor as well, which would be improbable since the reference frequencies are expected to be around the measured values. In the future more experiments should be performed, for example including the demo board, to verify the measured sensitivities.

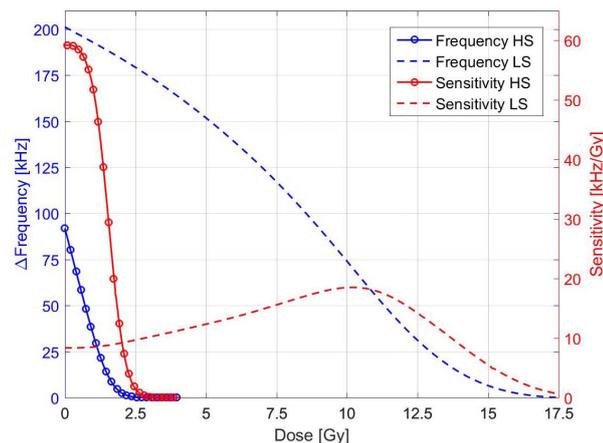


Figure 4.9: Typical progress of sensitivity and frequency output of an FGDOS in function of received dose [15].

### 4.2.3. Linear Range

During some of the tests, the linear range, defined by the threshold and target frequency limits, was extended to test sensor behaviour outside of the nominal values recommended by Sealicon. Figure 4.7 shows that the linear range could be extended upwards, but that the sensitivity quickly deteriorates below the linear range. To quantify this extension, the sensitivity calculated using linear regression for parts of the measured range can be compared with the sensitivity over the linear range, as shown in table 4.3. It shows that extending the linear range upwards would only result in about a 0.5% sensitivity difference. Additionally, the partial sensitivities for the frequencies upwards of 90 kHz show a difference with the average comparable to the partial sensitivities inside of the nominal linear range.

Table 4.3: Sensitivities calculated using linear regression for different parts of the frequency range for test run 07 with a beam energy of 70 MeV at DUT and a flux of  $3.69e7 \text{ \#/cm}^2/\text{s}$ . The relative difference between each partial sensitivity and the sensitivity calculated over the nominal linear range is given as well.

Frequency range (kHz)	S1 (Hz/Gy)	Difference (%)	S2 (Hz/Gy)	Difference (%)
90-50	8955		9684	
120-50	9007	0.58	9749	0.67
120-110	8986	-0.23	9687	-0.64
110-100	9135	1.66	9697	0.10
100-90	9018	-1.28	9866	1.74
90-80	9026	0.09	9825	-0.42
80-70	8888	-1.53	9763	-0.63
70-60	8975	0.98	9635	-1.31
60-50	8898	-0.86	9552	-0.86
50-40	8486	-4.63	9382	-1.78
40-30	8206	-3.30	8863	-5.53
30-20	7254	-11.60	8114	-8.45
20-10	5821	-19.75	6458	-20.41

### 4.2.4. Passive Mode & Power Consumption

As stated before, most of the tests in passive mode had a programming issue. For test run 09, however, correct data are available and displayed in figure 4.10. Again, the data points that do not follow the linear trend were caused by a programming issue. It shows that the sensor is able to detect TID without the need of a power supply. The sensitivity over the passive part of the curve, calculated using the measurements right before and after passive mode, is  $8771 \text{ Hz/Gy}$ , which is slightly higher than the sensitivity over the active part of the curve, calculated using linear regression, which equals  $8615 \text{ Hz/Gy}$ . This could be caused by the fact that the passive part is outside of the nominal linear range. Active or passive mode should not influence the sensor's sensitivity, but more experiment could be performed to confirm this.

The power consumption for the sensors in different modes is shown in figure 4.11. During the power measurements, the sensors were put in different operating modes, with each operating mode lasting for about five minutes. The modes were, in chronological order: high sensitivity mode, low sensitivity mode, standby mode and passive mode. As stated before, in standby mode the sensor's power consumption is reduced to a minimum. However, to completely turn off the sensor, it should be disconnected from any power supply. This is called passive mode and was achieved using a P-channel MOS as a switch. The passive mode was only available for sensor 2. Hence, during the last step, sensor 1 was put back into high sensitivity mode. The power consumption during high and low sensitivity mode was measured to be around  $20 \text{ mW}$  and  $10 \text{ mW}$  respectively, which agrees with previous research [15]. In standby mode the power consumption was almost zero, as expected. Power consumption could be even further reduced by putting the sensor in passive mode. The jumps in between the steps were caused by sensor resets in between the different modes.

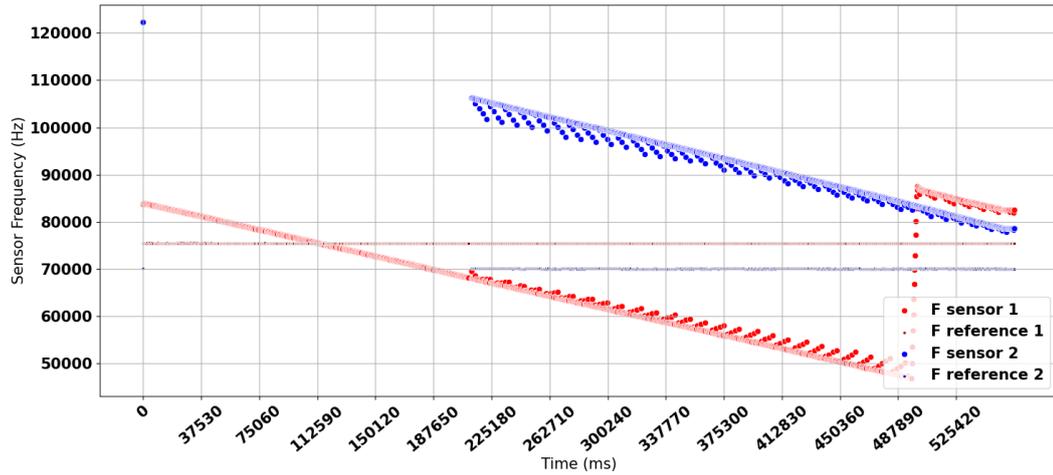


Figure 4.10: Sensor and reference frequencies in function of time with a beam energy of 70 MeV at DUT, a flux of  $7.07e6 \text{ \#/cm}^2/\text{s}$  and the sensors in high sensitivity mode. The plot corresponds with test run 09.

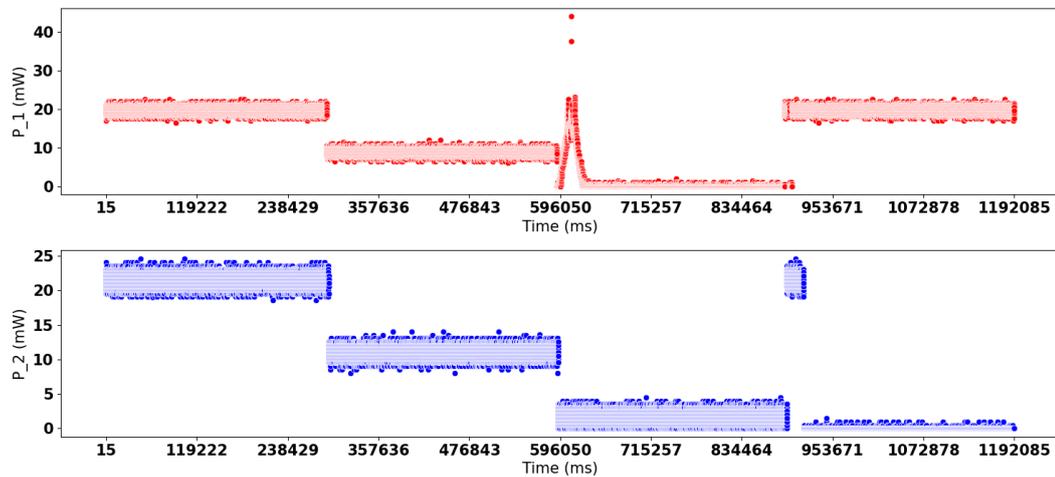


Figure 4.11: Power consumption of the sensors in four different steps, each lasting about five minutes: high sensitivity mode, low sensitivity mode, standby mode and passive mode. Since passive mode was only available for sensor 2, sensor 1 was put back into high sensitivity mode during the last step.

#### 4.2.5. Recharging Process

The recharging process is depicted in figures 4.5a and 4.7. The amount of recharge lost during the recharge can be easily calculated by multiplying the dose rate with the time needed for a recharge. The figures also show the influence the recharge voltage, 16 V versus 16.5 V, has on the needed recharge time. The fact that only a limited amount of bits is used to store threshold and target frequencies can be noticed from the figures too. For example, for figure 4.5a, the threshold was set at the lower end of the linear range, i.e. 50 kHz. Dividing this by a window factor of  $31.25 \text{ kHz}/4096 \text{ pulses} = 7.63$  results in a register value of 6554. The binary representation of this value is depicted in equation 4.4. Remember that, for comparison with the threshold value, the three most significant bits are dropped and the five next ones are taken to determine the start of the recharge process. Hence, the value represented by equation 4.5 is the register value below which a recharge is triggered. Multiplying this by the window factor results in a frequency of 46875 Hz, which is exactly the value at which the sensors started the recharge process during the radiation tests.

$$50 \text{ kHz} \rightarrow 6554 \rightarrow 00|00011001|10011010 \quad (4.4)$$

$$46875 \text{ Hz} \leftarrow 6144 \leftarrow 00|00011000|00000000 \quad (4.5)$$

Previous research at CERN, discussed during private conversations, indicated that the sensor frequency shows an overshoot immediately after the recharge process. This was confirmed in the experiments and is shown in figure 4.12. The vertical line in the figure indicates where the recharge process stopped. The sensor frequency still went up although the recharge ceased, after which it dropped and eventually showed linear behaviour again. It is assumed that this is caused by an internal capacitive load incorporated in the recharging circuit that has to unload after the end of the recharge process. Finally, Brucoli et al. [14] stated that the recharge rate has a strong influence on sensitivity degradation. As a result, the dose rate also indirectly influences sensitivity degradation. This effect was not investigated during the radiation tests.

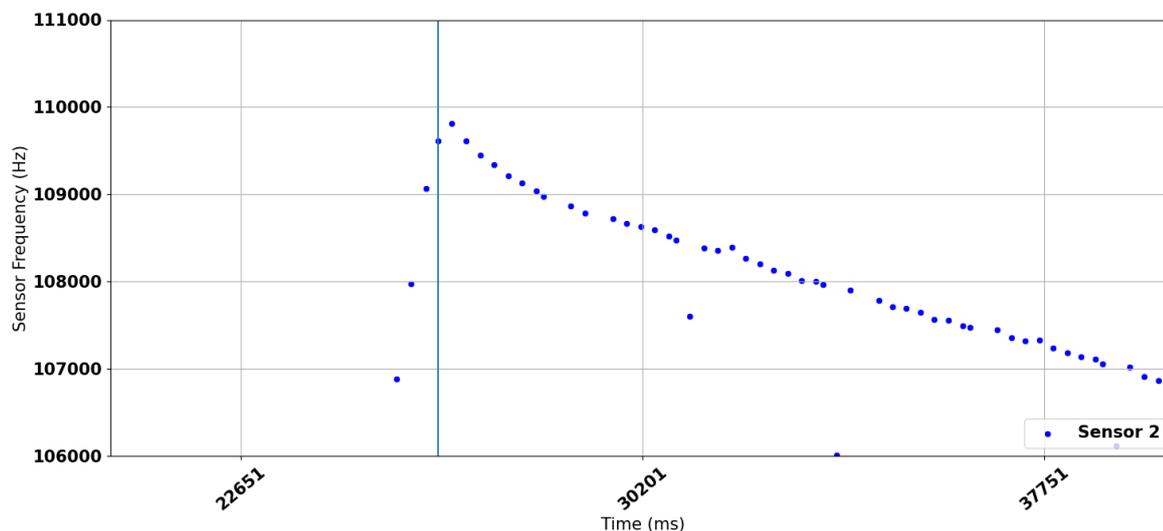


Figure 4.12: Sensor frequency overshoot in function of time with a beam energy of 70 MeV at DUT, a flux of  $1.40e7 \text{ \#/cm}^2/\text{s}$  and the sensor in high sensitivity mode. The plot corresponds with test run 05.

#### 4.2.6. Annealing & Charge Retention

In section 4.2.2 two mechanisms of annealing are described. First, a decrease of the frequency was noticed during short breaks in between tests or during RF trips of the cyclotron. This is probably caused by the annealing of charge traps in the reading transistor. The traps travel to the substrate, lowering the gate voltage, and thus cause a decrease in frequency output. In table 4.4 an overview of the decrease noticed during these breaks is given in  $\text{Hz/hr}$ . The frequency decrease in between tests was calculated using the last and first frequency outputs of the respective tests. The degradation during a test, caused by either an RF trip or by a user defined break, was calculated using linear regression. For the first tests and at low flux levels, no significant degradation was noticed, which can be expected since the low fluxes cause fewer oxide traps. The table shows that higher decreases can be expected right after higher fluxes were used. Additionally, the decrease is lower in low sensitivity mode, because the annealing traps have a relatively lower impact on the frequency output. The sensors were also tested again after 29 days. For sensor 2 no comparison in degradation over this time period can be made, since the frequency after the last test is unknown because of an electrostatic discharge. For sensor 1 a decrease from 82.2 to 75.6 kHz was noticed. This seems to agree with the 4 kHz degradation after five days mentioned in the sensor datasheet [89]. In the future, this decrease should be measured more gradually to see the effect over time and make sure there is no charge leakage.

The second effect is the annealing of trapped charges in the field oxide, as already explained in section 4.2.2. These traps cause a reduction of the electric field, as depicted in figure 4.13. Consequently, the electron-hole pairs generated by ionising radiation have a higher recombination rate, leading to a lower sensitivity of the sensor. Annealing of these traps results in a partial recovery of the sensitivity. The threshold voltage shift of the reading transistor caused by TID has an indirect influence on the sensitivity. Because of this shift, the floating gate contains less charge after a sensor recharge, resulting in a lower electric field and higher recombination rate. An additional effect to take into account, as

Table 4.4: Frequency degradation between tests, during a test break or during an RF trip of the cyclotron. The duration of each break is noted, as well as the frequency change per hour.

Test	Sens	Flux (#/s/cm <sup>2</sup> )	Sensor	Time (min)	dF/dt (Hz/hr)	Comment
02 – 03	HIGH	1.71e6	S1 S2	2.0	0 0	Break
06	LOW	1.40e7	S1 S2	3.4	607 592	RF trip
07	HIGH	3.69e7	S1 S2	3.3	8181 9403	RF trip
08	LOW	3.69e7	S1 S2	2.6	2233 2400	Break
13 – 14	HIGH	7.07e6	S1 S2	5.6 5.6	2703 2703	Break

reported by Brucoli et al. [14], is that the interface traps in the reading transistor might decrease the channel mobility, also reducing sensitivity. The authors state that this can be resolved by implementing a radiation tolerant reading structure. In a private conversation with the authors, they confirmed that this has been done but that it did not resolve the issue.

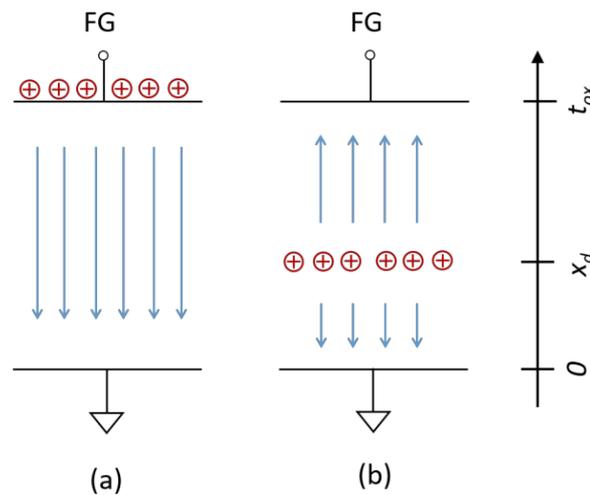


Figure 4.13: Electric field generated by the floating gate charge, left, and the effect on the electric field caused by charge traps in the field oxide [14].

#### 4.2.7. Resolution & Temperature

To determine the resolution of the setup, i.e. the minimal detectable dose, additional tests were performed without any radiation source. The resulting data were used to investigate noise in the circuit and temperature effects. This investigation is presented in the subsequent paragraphs. Considerations on how to reduce noise in the circuitry are also discussed. In the sensor datasheet a noise level of 75 Hz when using a 250 ms window is mentioned. Brucoli et al. [15] noted a minimal detectable frequency of 30 Hz. Brucoli [11] stated that part of the frequency fluctuations are caused by variations in temperature. The author filtered the signal using a moving average and subtracted the filtered signal from the original one to evaluate the noise. This resulted in a minimum detectable frequency of about 5 Hz. The frequency resolutions can be converted to dose resolutions by dividing them with the sensor sensitivity.

### Noise Levels

In figures 4.14a and 4.14b, the sensor and reference frequency of a noise test in high sensitivity mode are depicted. For the experiments the amount of window pulses was set to 4096, resulting in a measurement window of 262 s, see section 4.1.6. The internal temperature sensor indicated a constant temperature throughout the test. Nevertheless, the reference frequency of sensor 1 shows a gradual decrease, indicating a slight temperature variation over time. For an analysis of the noise, a more constant part of the curve, from 500 s to 1100 s, was used. The distributions of the sensor frequency outputs are depicted in figures 4.15a and 4.15b, with a normal distribution fitted on top of the histograms. A normality test, like a Shapiro-Wilk test, is not possible because the frequency output is a discrete and not a continuous variable. Based on conversations with researchers at CERN, the noise of the sensor should be determined using a  $4\sigma$  range, with  $\sigma$  the standard deviation of the normal distribution of the sensor output frequency. Applying this to the figures results in the values mentioned in table 4.5 for test 01. The table includes the total duration of the test, the number of gathered data points, the average frequency during the test, the sensitivity mode and the measurement window setting. In case of a normal distribution, the resolution is determined using a  $4\sigma$  range. For other distributions the resolution calculation is explained in the following paragraphs. The filtered resolution is calculated in the same way, but uses the filtered signal, explained in the next paragraph, as an input.

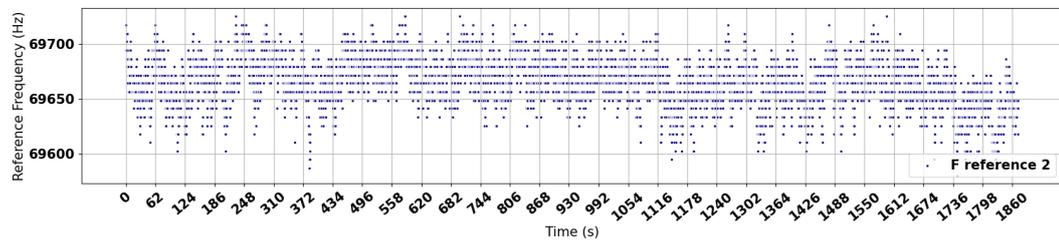
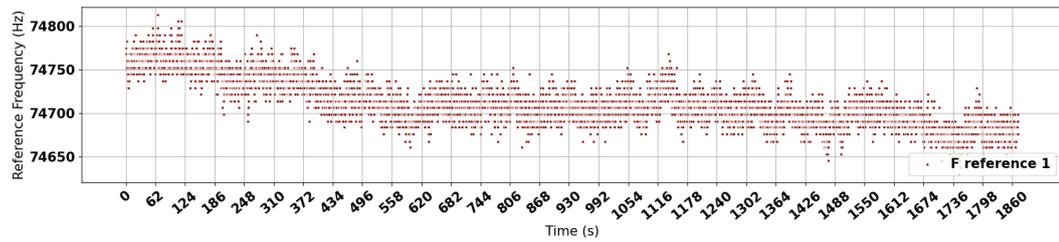
Table 4.5: Overview of the resolution determined via noise tests. Sensor 1 is the prototype and sensor 2 is the FGD-03F chip. The filtered signal was determined using a moving average window of about 20 s.

Test	Time (s)	Sensor	Data Points	Average (Hz)	Resolution (Hz)	Filtered Resolution (Hz)	Sensitivity	Window
01	600	1	1960	87634	115	95	HIGH	4096
		2	1911	87168	109	88	HIGH	4096
02	1800	1	6282	169622	162	156	LOW	4096
		2	6172	169405	154	148	LOW	4096
03	600	1	282	120588	79	58	HIGH	32768
		2	280	130906	82	60	HIGH	32768
04	600	1	280	120548	77	56	HIGH	32768
		2	281	130928	100	58	HIGH	32768

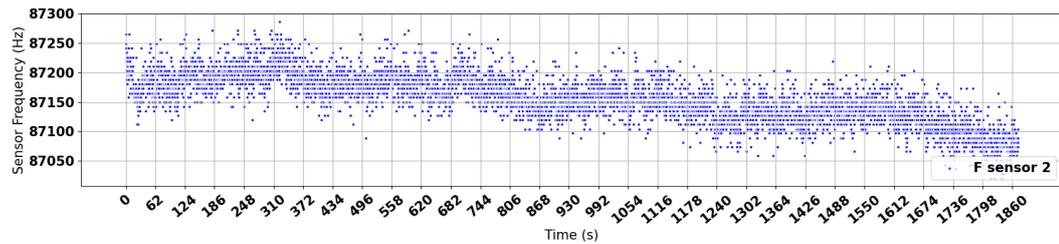
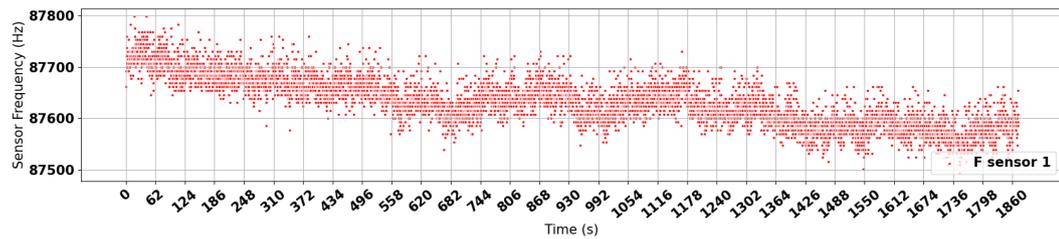
A moving average was applied to the frequency to eliminate the effect of small temperature variations [11]. First, the duration of a single measurement window of the sensor can be calculated from the clock frequency and the amount of pulses during which the sensor is counting the sensor and reference frequencies, as described in section 4.1.6. This results in a value of  $2 * 4096 / 31.250 \text{ Hz} \approx 250 \text{ ms}$ . By setting a moving average over 80 data points, a moving average window of about 20 s was obtained. Next, temperature fluctuations were eliminated by subtracting the filtered signal from the original signal [11]. These steps are depicted in figures 4.16a and 4.16b. Again, a  $4\sigma$  range was determined, which is also displayed in table 4.5.

The distribution of the frequency for low sensitivity mode, based on the test data depicted in figure 4.19 between 600 s and 2400 s, is depicted in figures 4.17a and 4.17b. It shows that the frequency does not follow a normal distribution but looks more like a bimodal distribution. To calculate the resolution, a Gaussian Kernel Density Estimation (KDE) is used to estimate the probability density function of the sensor frequency outputs, also shown in the figures. Although a  $4\sigma$  range could be taken to determine the resolution of the sensor, a different approach was taken. Since, for a normal distribution, 95.45% of values lie within a range of  $4\sigma$ , the Gaussian KDE is used to determine the frequency range that encompasses 95.45% of the values of the distribution. This results in the range mentioned in table 4.5 for test 02. This signal was also filtered using a moving average, see table 4.5.

An additional test was performed to analyse the influence of selecting a larger measurement window. This test, during which the sensor's counter measured for 32768 pulses, or 8x longer than the previous window selection, is depicted in figures 4.18a and 4.18b. The frequencies show a slight jump in their values after about 30 min. At this point the external power supply of the Arduino was disconnected to see the influence on the sensor frequency. This is further explained in the following paragraphs.

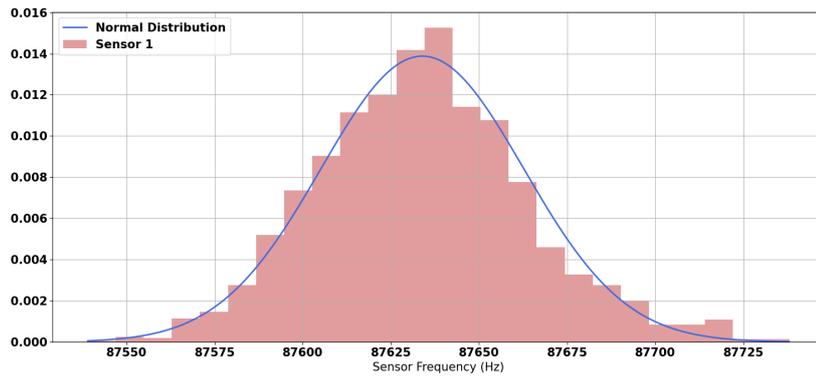


(a) Reference frequency in function of time.

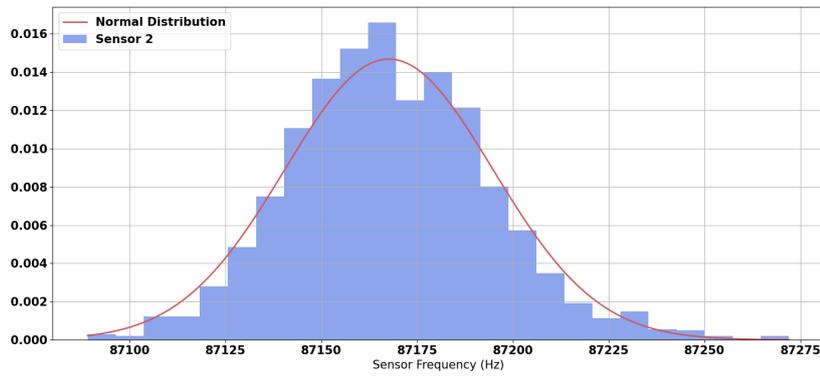


(b) Sensor frequency in function of time.

Figure 4.14: Sensor output in function of time for a noise test in high sensitivity mode using a window of 4096 pulses. The figures correspond with test 01 in table 4.5.

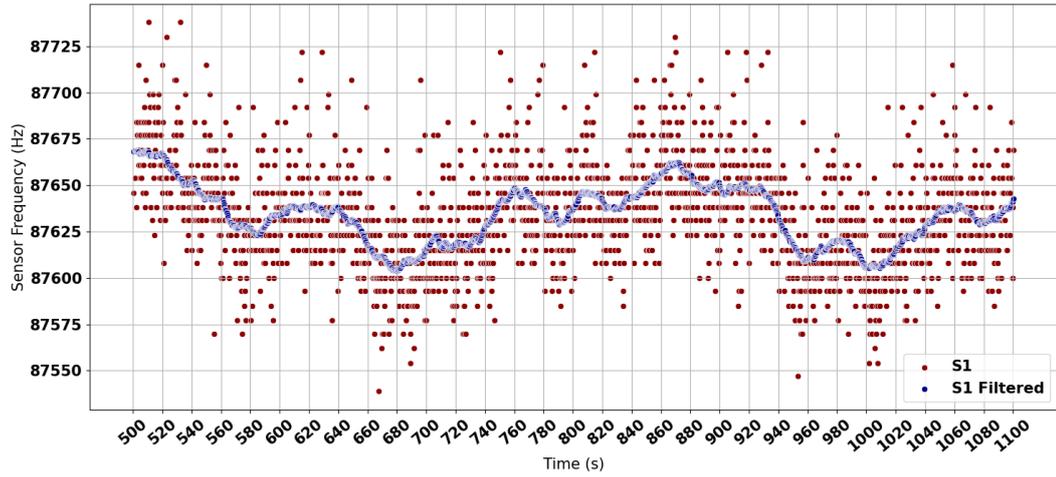


(a) Histogram of the sensor frequency of sensor 1.

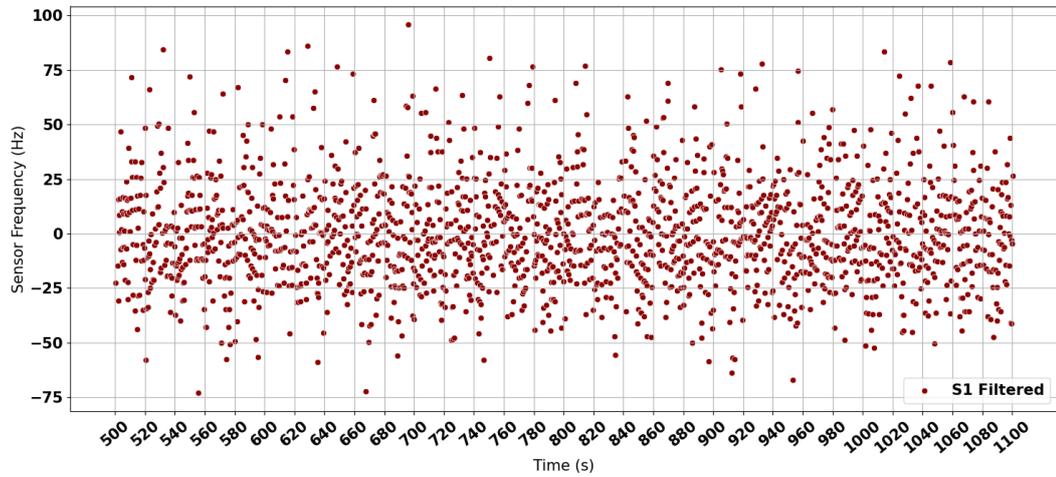


(b) Histogram of the sensor frequency of sensor 2.

Figure 4.15: Histograms of the sensor frequency outputs for a noise test in high sensitivity mode. A normal distribution is fitted on top of the histograms. The figures correspond with test 01 in table 4.5.

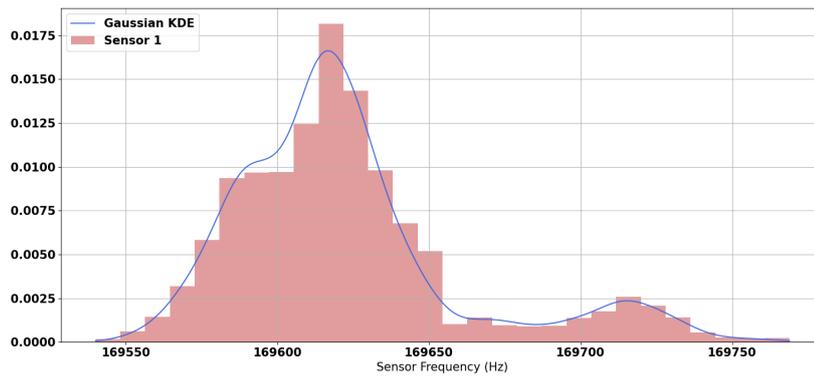


(a) Original and filtered signal of the sensor frequency in function of time.

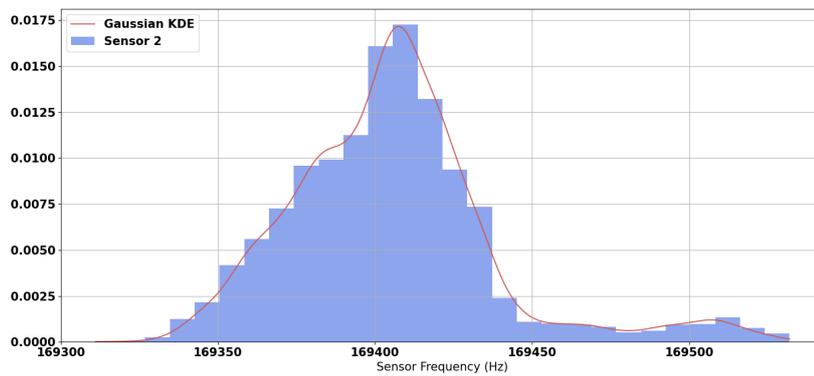


(b) Sensor noise remaining after subtracting the filtered signal from the original signal.

Figure 4.16: Sensor frequency output in function of time and filtering of the signal using a moving average with a window of 20 s for a noise test in high sensitivity mode. The figures correspond with test 01 in table 4.5.

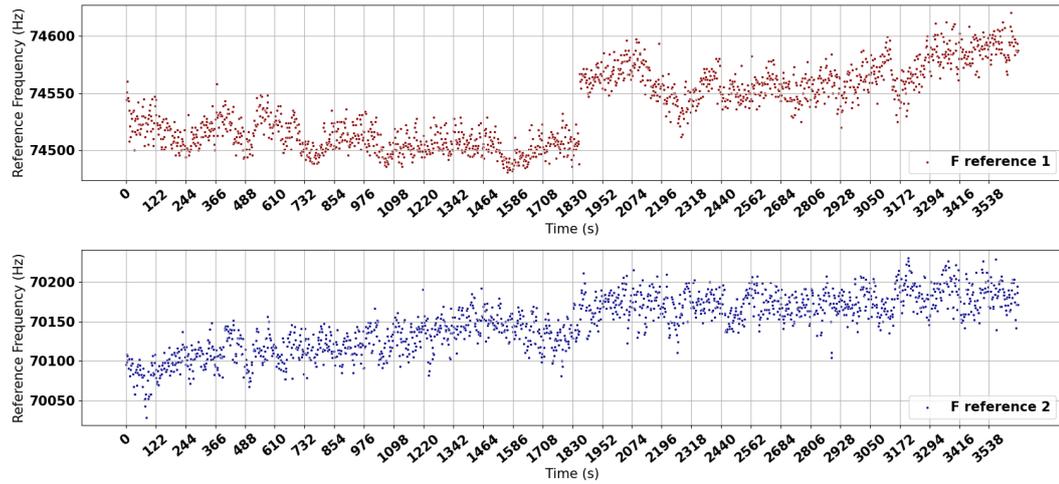


(a) Histogram of the sensor frequency of sensor 1.

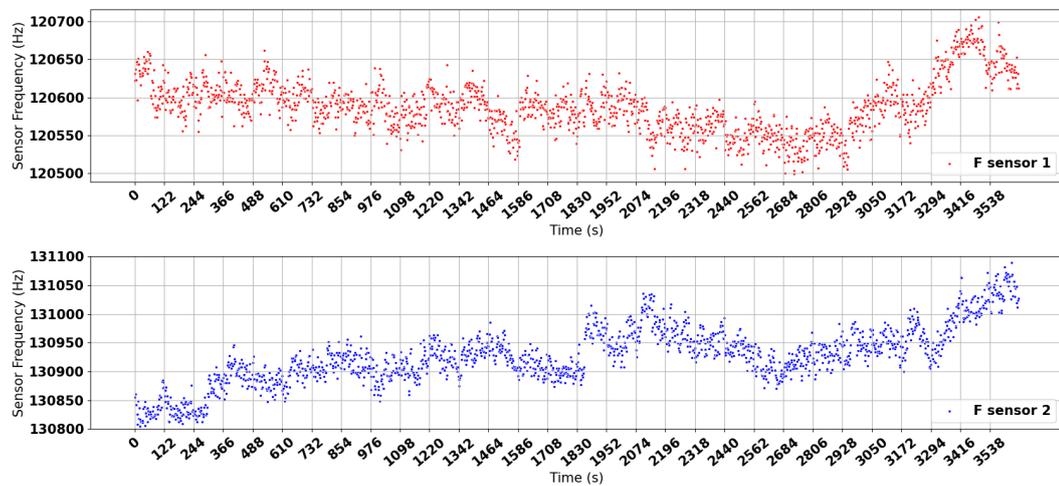


(b) Histogram of the sensor frequency of sensor 2.

Figure 4.17: Histograms of the sensor frequency outputs for a noise test in low sensitivity mode. A non-parametric distribution, determined via a Gaussian KDE, is fitted on top of the histograms. The figures correspond with test 02 in table 4.5.



(a) Reference frequency in function of time.



(b) Sensor frequency in function of time.

Figure 4.18: Sensor output in function of time for a noise test in high sensitivity mode using a window of 32768 pulses. The figures correspond with test 03 and 04 in table 4.5.

Although the tests were run for about an hour and the temperature was measured to be constant, the frequencies still show temporal fluctuations, making it impractical to select a region for noise analysis during which the frequencies were more constant. The noise follows a normal distribution and the  $4\sigma$  limits are displayed in table 4.5. In the table, test run 03 corresponds with the time frame between 620 – 1220 s, thus before the external power supply of the Arduino was unplugged, and test run 04 corresponds with the time frame between 2400 – 3000 s, thus after the external power supply was unplugged. A moving average window of 20 s was applied here as well, but this window now only contains 10 data points. Also note that, due to previous recharges, the frequencies are actually outside of the linear range for the high sensitivity mode, but it is assumed that this does not affect the signal noise. As can be expected, the table shows that using a larger measurement window results in an improved resolution. Assuming a sensitivity of 9 kHz/Gy in high sensitivity mode and a resolution of 80 Hz, the sensor's minimal detectable dose is 8.9 mGy, which is significantly higher than the 0.16 – 0.5 mGy mentioned in literature [13, 15]. This is caused by a combination of the relatively high noise levels and low sensitivity measured during the experiments.

### Temperature Influence

Although it was not possible to perform detailed temperature tests, some tests were performed to get a general idea about the effect of temperature on sensor frequency. In figures 4.19a, 4.19b and 4.19c, the temperature, reference frequency and sensor frequency are given as a function of time for a changing temperature. It is clear that the sensor frequency goes down with increasing temperature. However, as stated before, precise characterisation was not possible, due to the lack of precise temperature control and measurement equipment. Figure 4.19a shows that both sensors display a different temperature change. This is caused by the fact that both temperature sensors have a different offset and a precision of only  $1^\circ\text{C}$ . Thus, it can only be concluded that the actual temperature change is somewhere between  $1 - 2^\circ\text{C}$ . With a frequency change of approximately  $1370\text{ Hz}$  and  $1570\text{ Hz}$  for sensors 1 and 2 respectively, this results in a change of  $685 - 1370\text{ Hz}/^\circ\text{C}$  for sensor 1 and of  $785 - 1570\text{ Hz}/^\circ\text{C}$  for sensor 2. In figure 4.19b the complete output of this test run is shown, thus not only the part where the temperature was changed. The figure shows again that, when using low sensitivity mode, the reference frequency needs several minutes to stabilise.

### Noise Reduction

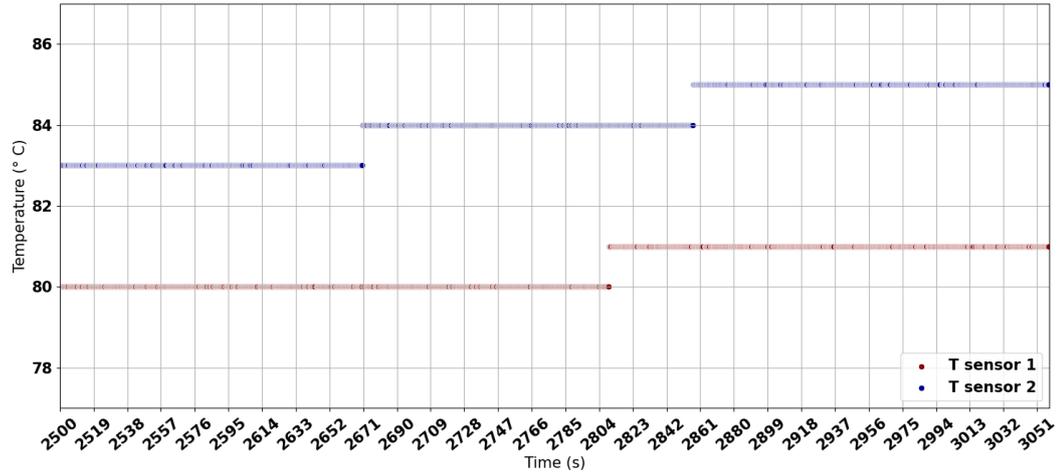
The levels of noise mentioned in table 4.5 are slightly higher than what is mentioned in the sensor datasheet, i.e.  $75\text{ Hz}$  for a window of  $250\text{ ms}$ . Moreover, they are significantly higher than the  $30\text{ Hz}$  mentioned in literature, thus suggesting that noise could be further reduced [15]. Brucoli et al. [15] mentioned in a private conversation that the  $30\text{ Hz}$  was derived using a measurement window of  $1\text{ s}$ . Besides the sensor itself, several other parts of the setup can influence the electric noise levels. The Arduino power supply was already briefly mentioned before. The Arduino can be powered via either the same USB cable it uses for data exchange or via an external power supply that triggers the internal voltage regulator. This was verified by measuring the bus voltage and is depicted in figure 4.20. During the first  $2\text{ min}$  in the figure, the Arduino used an external power supply and was doing nothing. During the next  $2\text{ min}$ , the Arduino was actually interacting with the sensor and sending data to the computer. Next, the same two steps were repeated in reverse order, but now with the external power supply disconnected. Although the sensor's supply voltage shows less variation using the Arduino's external power supply, the effect was not noticed in the sensor output resolution, see table 4.5.

The other subsystems of the Arduino board are also possible origins of noise. For example, as mentioned in section 3.6, the Arduino's internal oscillator is not very precise. As a result, the PWM and SPI clock signals are affected as well. In an attempt to limit noise in the circuit, decoupling capacitors were placed on the breadboard, as mentioned in section 4.1.3. Furthermore, the PCB was designed to include a ground plane, further limiting interference and thus noise levels, see section 4.1.2. In the future, more tests can be performed to determine the main origins of noise in the circuit.

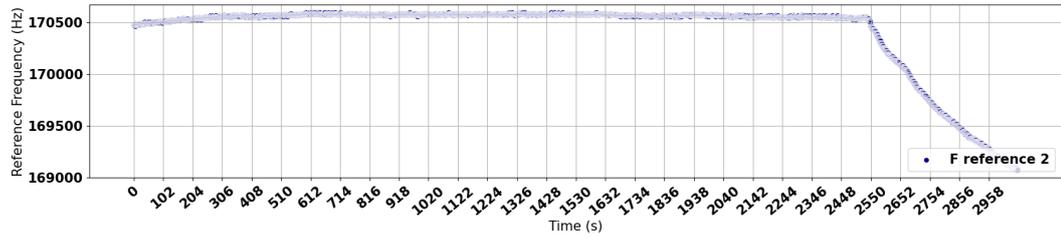
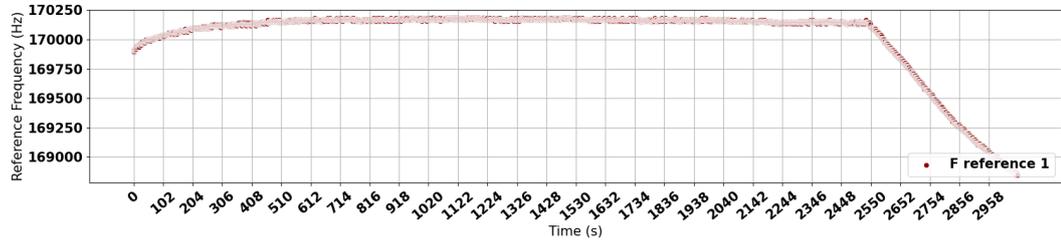
## 4.3. Supplementary Tests

On the 24<sup>th</sup> of February, additional radiation tests were performed. Although most of the data will be analysed by another TU Delft researcher, the most interesting results are already presented here. After a meeting with the researchers at CERN, they advised to test at higher energies to verify the sensor's low sensitivity values. In figure 4.21 the result of a test using a beam energy of  $150\text{ MeV}$  is demonstrated. For these tests HollandPTC did not provide the flux, but the dose rate in air at DUT, from which the flux and the dose rate in Si can be calculated using the appropriate LET values. For this conversion, the beam energy at target has to be used, which was approximately  $120\text{ MeV}$ . Throughout different tests, the sensitivity was measured to be around  $35\text{ kHz/Gy}$ . This is close to the values noted by, for instance, Danzeca et al. [26], and suggests that the low beam energy might have been the cause of the low sensitivities measured before. Using this sensitivity and a frequency resolution of  $80\text{ Hz}$ , the sensor's minimal detectable dose is  $2.3\text{ mGy}$ .

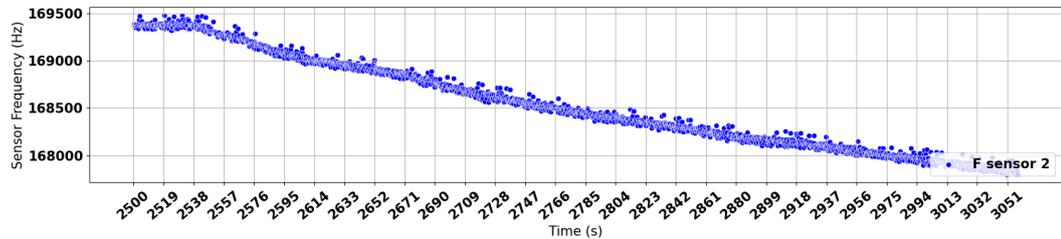
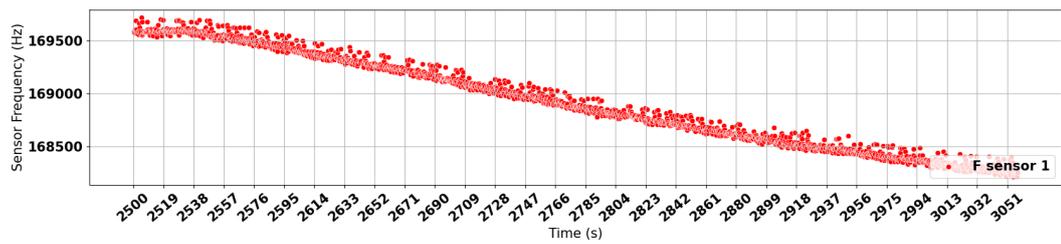
Another interesting result is depicted in figure 4.22. It shows the sensor's behaviour over a very wide range of frequency values. The region in which the sensors show linear behaviour can be clearly distinguished. Additionally, the figure shows the saturation of the floating gate charge for the given recharge voltage. The experiment depicted in the figure was performed with a beam energy of  $175\text{ MeV}$  and a dose rate at DUT in air of  $1.84\text{ Gy/min}$ . Again, the sensitivities are more in line with previous research [26].



(a) Temperature in function of time.



(b) Reference frequency in function of time for a changing temperature.



(c) Sensor frequency in function of time for a changing temperature.

Figure 4.19: Sensor output in function of time without radiation but with varying temperature and in low sensitivity mode.

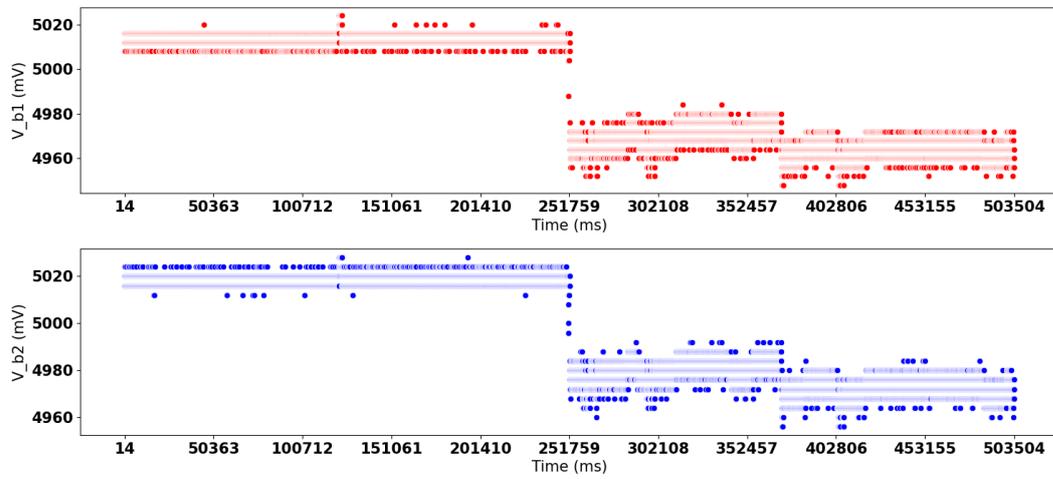


Figure 4.20: Arduino 5 V power supply output measured during four different phases, each lasting about 2 min: external power supply without sensor interaction, external power supply with sensor interaction, USB cable power supply with sensor interaction, USB cable power supply without sensor interaction.

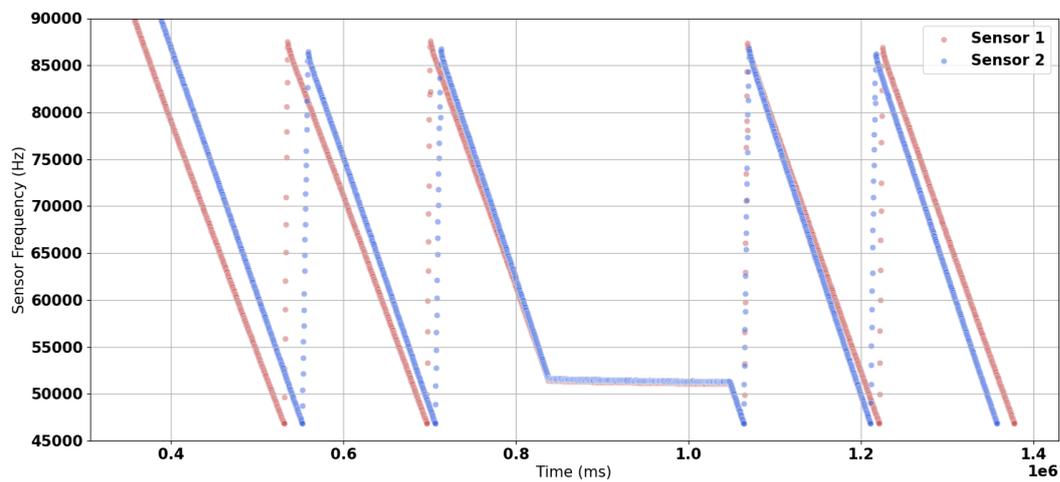


Figure 4.21: Sensor frequencies in function of time for an experiment performed on the 24<sup>th</sup> December with a beam energy of 150 MeV and a dose rate at DUT in air of 0.5 Gy/min.

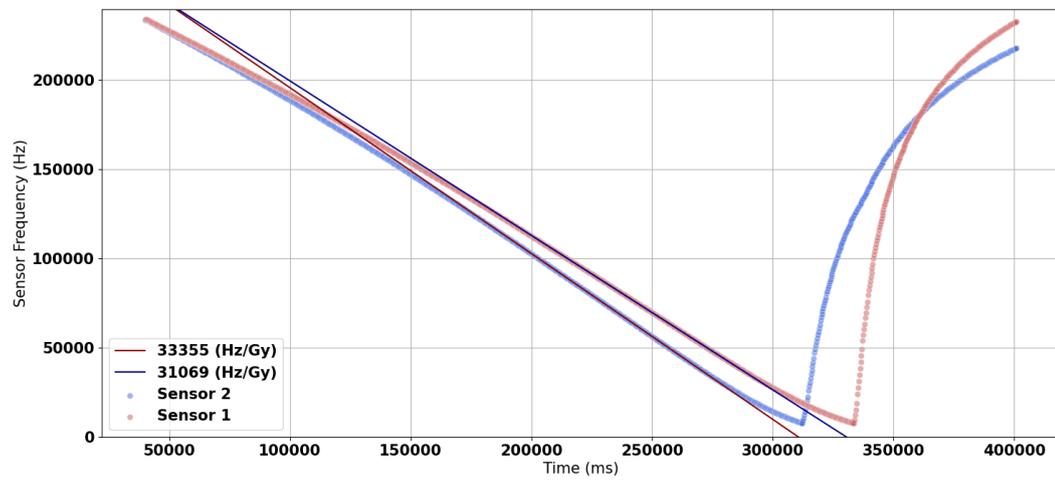


Figure 4.22: Sensor frequencies in function of time for an experiment performed on the 24<sup>th</sup> December with a beam energy of 175 MeV and a dose rate at DUT in air of 1.84 Gy/min. Linear regression lines and associated sensitivities are depicted as well.

# 5

## Conclusion

Space radiation monitoring is essential to improve engineering models, augment forecasting methods, track environmental changes, classify and analyse on-board anomalies and perform scientific research. Despite extensive space radiation research, more space radiation monitoring data and capabilities are needed. An innovative radiation monitoring system of small size, low power consumption and high precision would offer new possibilities for future missions.

The aim of this study was to design and characterise such a simple, compact, cheap but yet precise radiation monitoring system. To develop such a system, a sensor chip for solid state dosimetry was used. The interface between the sensor chip and a CubeSat system was designed by using the FlatSat at ESTEC. The performance and characteristics of the sensor were verified by means of proton beam radiation tests. In this chapter the final conclusions on the FlatSat integration and radiation tests are presented, the verification of the research goals are described and finally, suggestions for future research are made.

### 5.1. Floating Gate Dosimeter Payload Design

In chapter 3 the integration of a floating gate dosimeter in a typical small satellite system is discussed. In the first part of this chapter, the system requirements of a novel radiation monitoring tool are determined. These requirements are repeated in table 5.1. The table demonstrates which of the requirements are met. Requirement RMS-PERF-01 has not been met because, compared to previous research, the noise in the circuit was relatively high [11]. Additional research is needed to resolve this issue. Requirement RMS-PERF-02 was not experimentally verified in this thesis. The sensor datasheet mentions a maximum dose of 50 *krad*, and according to previous research, the sensor stops working around 28 *krad* when using a  $^{60}\text{Co}$  source [26, 89]. Consequently, it is expected that this requirement cannot be met, but this should be verified in future experiments. All the RMS-MINI requirements are met, except for the power requirement. However, the sensor can be used in passive mode, i.e. detect radiation without any power supply. This means that the average power consumption can be reduced in order to meet this requirement. The operating temperature range mentioned in the sensor datasheet is 232–357 K [89]. The temperature requirement RMS-SPAC-01 has not been met, but it should be noted that this requirement was based on hypothetical extreme situations. Requirements RMS-SPAC-02.1 to 04.1 require additional testing. It is expected that these requirements, except RMS-SPAC-03.1, will be met, since an earlier version of the sensor has already been demonstrated to work in space [22]. No specific software dedicated to SEUs, as required by RMS-SPAC-03.3, has been implemented. Future research on the implementation of this software is needed. Requirement RMS-SPAC-04.2 is met, since all parts of the system in both the FlatSat and the radiation test setup were connected to a common ground. Requirement RMS-SPAC-04.3 still needs verification. Despite the internal excess current protections of the sensor chip, several sensors broke down when an attempt was made to discharge them. Furthermore, some of the errors encountered during the tests were assumed to be caused by discharge currents, but these were soft errors, i.e. they could be resolved by a sensor reset. Hence, requirement RMS-SPAC-04.4 should be further investigated. Requirements RMS-SPAC-04.5 and 05

should be verified in the future, whilst requirement 06 is assumed to be met based upon a simple inspection of the chip. Requirements RMS-SPAC-07 and 08 should also be verified in subsequent research. Requirements RMS-COTS-01 to FLEX-01 are all met based upon an inspection of the sensor chip.

Table 5.1: Overview of the radiation monitoring system requirements. The last column states if the requirement has been met (V), not met (X) or still has to be verified (-).

Label	Description	OK
RMS-PERF-01	The system shall have a resolution of at least 30 <i>mrad</i>	X
RMS-PERF-02	The system shall have a measurement range of minimum 120 <i>krad</i>	-
RMS-MINI-01	The system shall have a maximum size of 3.2x3.2x3.2 <i>mm</i>	V
RMS-MINI-02	The system shall have a maximum weight of 2.4 <i>g</i>	V
RMS-MINI-03	The system shall have a maximum power usage of 10 <i>mW</i>	V
RMS-MINI-04	The system shall have a supply voltage of 3 – 5.5 <i>V</i>	V
RMS-SPAC-01	The system shall work in a temperature range of 60 – 380 <i>K</i>	X
RMS-SPAC-02.1	The system shall consist of materials that can withstand exposure to vacuum and UV	-
RMS-SPAC-02.2	The system shall consist of materials that generate little contamination	-
RMS-SPAC-03.1	The system shall be radiation hardened to doses of at least 120 <i>krad</i>	-
RMS-SPAC-03.2	The system's hardware shall be SEL and SEU resistant	-
RMS-SPAC-03.3	The system shall have software algorithms dedicated to the detection and recovery of SEUs	-
RMS-SPAC-04.1	The system shall prevent differential charging of surfaces by means of conductive coatings	-
RMS-SPAC-04.2	The system shall have a common ground	V
RMS-SPAC-04.3	The system's electronics and wiring shall be shielded physically and electrically	-
RMS-SPAC-04.4	The system shall include electric filtering to protect circuits from discharge-induced currents	X
RMS-SPAC-04.5	The system's exterior surfaces shall be able to withstand dielectric breakdown	-
RMS-SPAC-05	The system shall have a reliability of 0.89 or higher	-
RMS-SPAC-06	The system shall be able to operate in zero gravity	V
RMS-SPAC-07	The system shall be able to withstand launch stresses and vibrations TBD	-
RMS-SPAC-08	The system shall be constructed with materials that resist atomic oxygen erosion	-
RMS-COTS-01	The dosimeter shall be able to be produced in standard manufacturing processes	V
RMS-COTS-02	The system, besides dosimeter, shall consist of only COTS parts	V
RMS-COTS-03	The system shall cost as much as a system containing a RADFET dosimeter or less	V
RMS-COTS-04	The system shall communicate via a common serial communication interface	V
RMS-FLEX-01	The system shall have multiple sensitivity options	V

As mentioned before, several of these requirements take the most extreme possible operating conditions into account. Although some requirements have not been met, the sensor chip still shows great potential. Several requirements have not been verified, but based on data from previous literature, it is expected that they will be satisfied in the future [15, 22]. It can be concluded that the system is relatively cheap, straightforward to use, easy to produce and able to withstand harsh space environment conditions. The radiation tests indicate that the sensor does not outperform current miniature solid state detectors with regard to precision, despite the levels of sensitivity and minimal detectable radiation dose mentioned in previous research [15, 26].

## 5.2. Radiation Testing

In chapter 4 the floating gate dosimeter was irradiated with protons to verify its characteristics. The sensor showed linear behaviour as long as some limits with regard to the output frequency range were respected, which makes it more straightforward to use than a RADFET sensor. Additionally, FGDOs do not need a bias specific calibration curve. The sensor can also be used in passive mode, reducing power consumption. Finally, the tests prove the floating gate can be very easily recharged, extending the detectable dose range of the sensor chip.

The TID effect on the sensor was demonstrated, albeit only qualitatively. This affects the precision of the sensor over time. A precise quantitative description of this effect could be applied by sensor users to ensure a minimal level of accuracy. The influence of flux is not clear and seems to be limited. A clear recovery of the sensitivity after TID caused by annealing was noted during the tests, which should be taken into account by future users of the sensor. This may be less of a concern for space applications, because of the low dose rates in space compared to the dose rates employed during the radiation tests.

It was also shown that the sensor output degrades over time. Thus, in real-life applications, the sensor output should be consulted regularly in order to avoid misinterpretations.

The resolution has shown to be lower than expected, whereas it should actually be one of the advantages of this type of sensor [11, 13]. Part of this low resolution can be explained by inaccuracies in the Arduino and surrounding circuitry. The noise can be reduced by using longer measurement windows. Furthermore, it was demonstrated that some fluctuations in the output signal can be mitigated by applying a moving average filter. The origin of the fluctuations can be found in the temperature analysis [11]. This analysis shows that the sensor's output frequency can change by about 700 – 1400 Hz before the sensor's internal temperature sensor actually detects a difference, demonstrating a too low resolution of the internal temperature sensor.

Previous research indicated a sensitivity that was low compared to the sensitivity advertised by Sealicon [13, 26]. It has been noted by other researchers that the sensitivity tends to decrease with decreasing particle energy. The decrease indicated by the experiments carried out in this thesis was larger than the one mentioned in literature [26]. The correct functionality of the sensor and the setup was verified by performing experiments at higher particle beam energies. The results hereof indicate a sensitivity comparable to the sensitivity described in the literature [26]. The sensitivity at lower particle beam energy levels thus requires further research, especially because most protons in space have an energy level below 100 MeV [31, 49].

### 5.3. Research Goals & Question

At the start of the thesis, the following research question and subquestions were formulated:

*Can a relatively simple, lightweight and cheap space radiation monitoring system be designed that exploits the properties of floating gate dosimeters and that, combined with CubeSat technology, is able to answer the open questions in space radiation and/or improve continuous radiation monitoring?*

- *Can the floating gate dosimeter be implemented into a microcontroller based system, resulting in a basic payload design for use in future space missions?*
- *What would the characteristics of such a system be and what would be its main benefits for space missions, i.e. reliability, measurement range and resolution, cost, weight, power consumption and ease of implementation?*

The thesis work proves that a simple and cheap system can indeed be built. The integration with the FlatSat was successful and shows that the FGDOS chip can be integrated into existing small satellite systems, thus answering the first subquestion. The literature study and the radiation tests determined the characteristics of the dosimeter chip, albeit that some properties need further investigation. It was shown that the sensor chip performs very well with respect to cost, power consumption, size and ease of use. The radiation tests showed, however, that the performance of the dosimeter compared to other systems is not as outstanding as previously reported in the literature [13, 26]. Nevertheless, the sensor shows great potential and should be investigated further to clearly determine its capabilities. This part of the thesis answers the second subquestion.

In conclusion, this study demonstrates that it is possible to design a simple, lightweight and cheap space radiation monitoring system based upon floating gate dosimeter technology that can be integrated into small satellite systems. These characteristics make it possible to widely deploy the sensor in constellations or as a main or secondary payload on both near-Earth and deep space missions. Other dosimeter systems with similar characteristics exist, and in order to outperform those, the issues related to sensitivity and noise should be resolved.

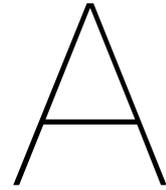
### 5.4. Future Work

Throughout the thesis report, several possible topics for future research were identified. First, several requirements listed in table 5.1 still require verification. For example, although the sensor chip already communicates via a widespread protocol, direct communication with the FlatSat setup was not possible. Since the goal is to deploy the sensor in as many satellites as possible, it would be beneficial to have the option to choose from several communication protocols. In fact, a sensor that uses the I2C protocol

is currently being developed by Sealicon. The deployment of the sensor as part of a SC and radiation data analysis using ML algorithms could also be investigated in the future. Another interesting topic for future research is the sensor's response to SEEs.

Several questions on radiation detection remain to be answered. As mentioned before, the sensors have been irradiated up to doses of 28 *krad*, but such a high total dose was only achieved with radioactive sources such as  $^{60}\text{Co}$ , and never with protons. It would be useful to investigate how properties change under increasing proton radiation and at which point the sensor breaks down. Another interesting experiment would be to document the maximum dose the sensor can withstand with an empty floating gate. This could be useful information for deep space missions. One sensor could be used until it breaks down because of high doses, after which the system can switch to a second sensor that was stored with an empty floating gate. The influence of dose on the sensitivity of a sensor with an empty floating gate should then be investigated as well. Furthermore, the effect of TID could be further investigated. Despite the fact that this effect is quite well understood, it might be interesting to quantify the sensitivity degradation caused by proton TID. Also, the influence of flux is not clear yet and should be further investigated. From the experiments carried out in this thesis, it was clear that the beam energy significantly influences the sensitivity, and more experiments can be performed to determine the relationship between sensitivity and beam energy over a wide energy range. Despite the fact that experiments have been performed with  $\gamma$  rays, protons and mixed fields, no detailed comparison exists between the effects caused by different types of particles at similar dose rates and energy levels. Hence, it would be interesting to investigate this. In none of the previous papers, the directionality dependence of the sensor has been discussed. It is thus unknown if the angle of incidence of the radiation has a significant influence on sensitivity. Floating gate dosimeters might prove useful for other research as well. Very recently, researchers at CERN have presented their results of testing the floating gate dosimeter as a single ion detector at the RADiation and its Effects on Components and Systems (RADECS) conference, and further investigations are ongoing [16].

The noise tests made clear that the setup being used suffers from relatively high noise levels. In the future, the cause of these high noise levels should be determined, and solutions should be studied. These tests also showed fluctuations caused by temperature variations that could not be compensated for. In order to improve accuracy, the integration of a more precise temperature sensor could be investigated.



# SPENVIS Simulations

## A.1. Low-Earth Orbit

<b>Trajectory generation:</b> use orbit generator <input type="button" value="v"/> <b>Number of mission segments:</b> 1 <input type="button" value="v"/> <b>Mission end:</b> total mission duration <input type="button" value="v"/> <b>Mission duration:</b> 1 <input type="text"/> years <input type="button" value="v"/> <b>Satellite orientation:</b> one axis parallel to the velocity vector <input type="button" value="v"/> <b>Account for solar radiation pressure:</b> no <input type="button" value="v"/> <b>Account for atmospheric drag:</b> no <input type="button" value="v"/>	<b>Segment title:</b> ISS <b>Orbit type:</b> general <input type="button" value="v"/> <b>Orbit start:</b> calendar date <input type="button" value="v"/> <input type="text" value="01"/> <input type="text" value="Jan"/> <input type="text" value="2022"/> <input type="text" value="00"/> : <input type="text" value="00"/> : <input type="text" value="00"/> <b>Representative trajectory duration [days]:</b> 30 <b>Altitude specification:</b> altitude for a circular orbit <input type="button" value="v"/> <b>Altitude [km]:</b> 420 <b>Inclination [deg]:</b> 52 <b>R. asc. of asc. node [deg w.r.t. gamma50]:</b> 193 <b>Argument of perigee [deg]:</b> 33 <b>True anomaly [deg]:</b> 130 <b>Output resolution</b> 1. 60.0 <input type="text"/> s below 20000.0 <input type="text"/> km 2. 240.0 <input type="text"/> s below 80000.0 <input type="text"/> km 3. 3600.0 <input type="text"/> s elsewhere
---	--

(a) Mission settings in SPENVIS.

(b) Orbit settings in SPENVIS.

Figure A.1: Orbit specifications for a one year CubeSat mission in LEO in SPENVIS based on the orbital characteristics of the ISS [82].

Trapped radiation models	
<b>Proton model:</b> AP-8 <input type="button" value="v"/> <b>Model version:</b> solar minimum <input type="button" value="v"/> <b>Threshold flux for exposure(/cm2/s):</b> 0.1	<b>Electron model:</b> AE-8 <input type="button" value="v"/> <b>Model version:</b> solar maximum <input type="button" value="v"/> <input type="button" value="v"/> include <input type="button" value="v"/> local time variation <b>Confidence level:</b> 97.725% <input type="button" value="v"/> <b>Threshold flux for exposure(/cm2/s):</b> 0.1
<b>Model developed by:</b> 	<b>Model developed by:</b> 

Figure A.2: Settings for the trapped radiation models in SPENVIS for a one year CubeSat mission in LEO.

<b>Solar particle model:</b>	ESP-PSYCHIC (total fluence) ▼
<b>Ion range:</b>	H ▼ to U ▼
<b>Prediction period:</b>	automatic ▼
<b>Offset in solar cycle:</b>	automatic ▼
<b>Confidence level [%]:</b>	97.725
<b>Magnetic shielding:</b>	on (quiet magnetosphere) <input type="button" value="edit"/>

Figure A.3: Settings for the solar particle fluence model in SPENVIS for a one year CubeSat mission in LEO.

<b>Ion range:</b>	H ▼ to U ▼
<b>GCR model at 1 AU:</b>	ISO 15390 ▼
	ISO-15390 standard model + 2 sigma ▼
<b>solar activity data:</b>	mission epoch ▼
<b>Magnetic shielding:</b>	on (quiet magn.) <input type="button" value="edit"/>

Figure A.4: Settings for the galactic cosmic ray flux model in SPENVIS for a one year CubeSat mission in LEO.

<b>Shielding depths:</b>	default values ▼
<b>Dose model:</b>	SHIELDOSE-2Q ▼
<b>Shielding configuration:</b>	centre of Al spheres ▼
<b>Shield composition:</b>	Aluminium ▼
<b>Target material:</b>	Silicon ▼

Figure A.5: Settings for the total ionising dose model in SPENVIS for a one year CubeSat mission in LEO.

<b>Source particle type and spectrum</b>	
Environment:	Mission based ▼ average trapped particle fluence ▼
Number of primary particles to simulate:	10,000 ▼
Warning: Particle track visualisation will be disabled!	
Incident particle type:	proton ▼
<b>Incident energy spectrum</b>	
Mission average spectrum	
Don't use ▼ energy biasing	
Interpolation type:	linear ▼
<b>Angular distribution</b>	
The angular distribution is omnidirectional.	

(a) Settings in SPENVIS for MULASSIS simulations with trapped protons as a source.

<b>Source particle type and spectrum</b>	
Environment:	Mission based ▼ GCR particle fluence ▼
Number of primary particles to simulate:	10,000 ▼
Warning: Particle track visualisation will be disabled!	
Incident particle type:	ion ▼
<b>Ion definition</b>	
Atomic number:	1 ▼
Isotope:	H1 ▼
<b>Incident energy spectrum</b>	
Mission average spectrum	
Don't use ▼ energy biasing	
<b>Angular distribution</b>	
The angular distribution is omnidirectional.	

(b) Settings in SPENVIS for MULASSIS simulations with GCRs as a source.

Figure A.6: Settings in SPENVIS for MULASSIS simulations for different sources of radiation particles for a one year CubeSat mission in LEO.

<b>Geometry:</b> <input type="text" value="Default"/>	
The default geometry is a single planar slab with 26 aluminium layers and boundaries equivalent to the default SHIELDOSE thicknesses.	
<b>Visualisation</b>	
Format:	<input type="text" value="Encapsulated PostScript (EPS)"/>
Particle tracks:	<input type="text" value="Do not display"/>

Figure A.7: Settings for the geometry in SPENVIS for MULASSIS simulations for a one year CubeSat mission in LEO.

<b>Analysis type:</b> <input type="text" value="Total ionizing dose"/>	
<b>Energy deposition / TID</b>	
Output units:	<input type="text" value="rad"/>
Select layers for energy deposition/total ionising dose analysis:	
1	<input checked="" type="checkbox"/>
2	<input checked="" type="checkbox"/>
3	<input checked="" type="checkbox"/>
4	<input checked="" type="checkbox"/>
5	<input checked="" type="checkbox"/>
6	<input checked="" type="checkbox"/>
7	<input checked="" type="checkbox"/>
8	<input checked="" type="checkbox"/>
9	<input checked="" type="checkbox"/>
10	<input checked="" type="checkbox"/>
11	<input checked="" type="checkbox"/>
12	<input checked="" type="checkbox"/>
13	<input checked="" type="checkbox"/>
14	<input checked="" type="checkbox"/>
15	<input checked="" type="checkbox"/>
16	<input checked="" type="checkbox"/>
17	<input checked="" type="checkbox"/>
18	<input checked="" type="checkbox"/>
19	<input checked="" type="checkbox"/>
20	<input checked="" type="checkbox"/>
21	<input checked="" type="checkbox"/>
22	<input checked="" type="checkbox"/>
23	<input checked="" type="checkbox"/>
24	<input checked="" type="checkbox"/>
25	<input checked="" type="checkbox"/>
26	<input checked="" type="checkbox"/>

Figure A.8: Analysis settings in SPENVIS for MULASSIS simulations for a one year CubeSat mission in LEO.

## A.2. Areostationary orbit

**Trajectory generation:** use orbit generator ▼

**Mission end:** total mission duration ▼

**Mission duration:** 10 years ▼

**Solar radiation pressure parameter [ $m^2 kg^{-1}$ ]:** 0.0

**Segment title:**  
Mars

**Orbit type:** general ▼

**Mars-centric coordinate system**

- **z-axis:** along mean rotational north pole of Mars
- **x-axis:** out along the ascending node "of date" of Mars ICRF/J2000 mean equator

**Orbit start**

01 ▼ Jan ▼ 2022 ▼ 00 ▼ : 00 ▼ : 00 ▼

**Representative trajectory duration ▼ [days]:** 5

**Altitude specification:** semi-major axis and eccentricity ▼

**Semi-major axis [km]:** 17032

**Eccentricity:** 0

**Inclination [deg]:** 0

**R. asc. of asc. node [deg w.r.t. gamma50]:** 0

**Argument of periareion [deg]:** 0

**True anomaly [deg]:** 0

**Output resolution**

1. 60.0 s below 20000.0 km
2. 240.0 s below 80000.0 km
3. 3600.0 s elsewhere

(a) Mission settings in SPENVIS. (b) Orbit settings in SPENVIS.

Figure A.9: Orbit specifications for a 10 year areostationary mission in SPENVIS.

**Ion range:** H ▼ to U ▼

**GCR model at 1 AU:** ISO 15390 ▼

ISO-15390 standard model + 2 sigma ▼

**solar activity data:** Solar Minimum (May 1996) ▼

**Propagation from 1 AU:** no scaling ▼

*reasonable engineering approximation (ECSS)*

**Magnetic shielding:** no

Figure A.10: GCR model settings in SPENVIS for an areostationary orbit.

# B

## Code

### B.1. FlatSat

#### B.1.1. FreeRTOS

```
1 void init_hello_world() {
2     static StaticTask_t hello_tcb;
3     static StackType_t hello_stack[500];
4     xTaskCreateStatic(hello_world_task, "HELLO", 500, NULL, 1, hello_stack, &hello_tcb);
5 }
6
7 TickType_t task_interval = 5; // the task interval in seconds
8 void hello_world_task(void * pvParams) {
9     TickType_t xLastWakeTime = xTaskGetTickCount();
10
11     for(;;) {
12         printf("Hello World \n");
13         vTaskDelayUntil(&xLastWakeTime, task_interval * configTICK_RATE_HZ);
14     }
15 }
```

Listing B.1: Example on how to create a task for the OBC in FreeRTOS.

```
1 /* set up I2C connection */
2 struct io_descriptor *I2C_1_io;
3 i2c_m_sync_get_io_descriptor(&I2C_1, &I2C_1_io);
4 i2c_m_sync_enable(&I2C_1);
5 i2c_m_sync_set_slaveaddr(&I2C_1, 0x25, I2C_M_SEVEN);
6
7 for(;;) {
8     io_write(I2C_1_io, <*variable>, <bytes>);
9     io_read(I2C_1_io, <*variable>, <bytes>);
10 }
```

Listing B.2: FreeRTOS I2C functions.

#### B.1.2. Yamcs

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <SpaceSystem name="flatsat"
3     xmlns="http://www.omg.org/space/xtce"
4     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.omg.org/spec/XTCE/20180204">
5     <TelemetryMetaData>
6         <ParameterTypeSet>
7             <EnumeratedParameterType name="OBCTypeType">
8                 <UnitSet />
9                 <IntegerDataEncoding sizeInBits="8" signed="false" />
10                <EnumerationList>
11                    <Enumeration value="1" label="Beacon" />
12                    <Enumeration value="2" label="Command" />

```

```

13         <Enumeration value="3" label="Data" />
14     </EnumerationList>
15 </EnumeratedParameterType>
16 <EnumeratedParameterType name="OBCEDataPacketTypeType">
17     <UnitSet />
18     <IntegerDataEncoding sizeInBits="8" signed="false" />
19     <EnumerationList>
20         <Enumeration value="1" label="Telemetry" />
21     </EnumerationList>
22 </EnumeratedParameterType>
23 <IntegerParameterType name="CSPPriorityType" signed="false" initialValue="0">
24     <UnitSet />
25     <IntegerDataEncoding sizeInBits="2" />
26 </IntegerParameterType>
27 <IntegerParameterType name="CSPAddressType" signed="false" initialValue="0">
28     <UnitSet />
29     <IntegerDataEncoding sizeInBits="5" />
30 </IntegerParameterType>
31 <IntegerParameterType name="CSPPortType" signed="false" initialValue="0">
32     <UnitSet />
33     <IntegerDataEncoding sizeInBits="6" />
34 </IntegerParameterType>
35 <IntegerParameterType name="CSPReservedFlagsType" initialValue="0">
36     <UnitSet />
37     <IntegerDataEncoding sizeInBits="3" />
38 </IntegerParameterType>
39 <BooleanParameterType name="CSPFragmentFlagType" initialValue="0" zeroStringValue
="NotFragmented" oneStringValue="Fragmented">
40     <UnitSet />
41     <IntegerDataEncoding sizeInBits="1" />
42 </BooleanParameterType>
43 <BooleanParameterType name="CSPAuthenticationFlagType" initialValue="0"
zeroStringValue="NotAuthenticated" oneStringValue="HMAC">
44     <UnitSet />
45     <IntegerDataEncoding sizeInBits="1" />
46 </BooleanParameterType>
47 <BooleanParameterType name="CSPEncryptionFlagType" initialValue="0"
zeroStringValue="Unencrypted" oneStringValue="XTEA">
48     <UnitSet />
49     <IntegerDataEncoding sizeInBits="1" />
50 </BooleanParameterType>
51 <BooleanParameterType name="CSPProtocolFlagType" initialValue="0" zeroStringValue
="UDP" oneStringValue="RDP">
52     <UnitSet />
53     <IntegerDataEncoding sizeInBits="1" />
54 </BooleanParameterType>
55 <BooleanParameterType name="CSPIntegrityFlagType" initialValue="0"
zeroStringValue="NotChecked" oneStringValue="CRC32">
56     <UnitSet />
57     <IntegerDataEncoding sizeInBits="1" />
58 </BooleanParameterType>
59 <IntegerParameterType name="uint8_t" signed="false">
60     <UnitSet />
61     <IntegerDataEncoding encoding="unsigned" sizeInBits="8" byteOrder="
leastSignificantByteFirst" />
62 </IntegerParameterType>
63 <IntegerParameterType name="int16_t" signed="true">
64     <UnitSet />
65     <IntegerDataEncoding encoding="TwosComplement" sizeInBits="16" byteOrder="
leastSignificantByteFirst" />
66 </IntegerParameterType>
67 <IntegerParameterType name="uint16_t" signed="false">
68     <UnitSet />
69     <IntegerDataEncoding encoding="unsigned" sizeInBits="16" byteOrder="
leastSignificantByteFirst" />
70 </IntegerParameterType>
71 <ArrayParameterType name="uint16_t_array" numberOfDimensions="1" arrayTypeRef="
uint16_t" />
72 <ArrayParameterType name="int16_t_array" numberOfDimensions="1" arrayTypeRef="
int16_t" />
73 <IntegerParameterType name="uint32_t" signed="false">

```

```

74         <UnitSet />
75         <IntegerDataEncoding encoding="unsigned" sizeInBits="32" byteOrder="
leastSignificantByteFirst" />
76     </IntegerParameterType>
77     <FloatParameterType name="float_t">
78         <UnitSet />
79         <FloatDataEncoding sizeInBits="32" byteOrder="leastSignificantByteFirst" />
80     </FloatParameterType>
81     <ArrayParameterType name="float_array" numberOfDimensions="1" arrayTypeRef="
float_t" />
82     <BooleanParameterType name="bool_t">
83         <UnitSet />
84         <IntegerDataEncoding sizeInBits="8" />
85     </BooleanParameterType>
86 </ParameterTypeSet>
87 <ParameterSet>
88     <Parameter name="CSP_Priority" parameterTypeRef="CSPPriorityType" />
89     <Parameter name="CSP_Source" parameterTypeRef="CSPAddressType" />
90     <Parameter name="CSP_Destination" parameterTypeRef="CSPAddressType" />
91     <Parameter name="CSP_SourcePort" parameterTypeRef="CSPPortType" />
92     <Parameter name="CSP_DestinationPort" parameterTypeRef="CSPPortType" />
93     <Parameter name="CSP_ReservedFlags" parameterTypeRef="CSPReservedFlagsType" />
94     <Parameter name="CSP_Fragmentation" parameterTypeRef="CSPFragmentFlagType" />
95     <Parameter name="CSP_Authentication" parameterTypeRef="CSPAAuthenticationFlagType"
/ >
96     <Parameter name="CSP_Encryption" parameterTypeRef="CSPEncryptionFlagType" />
97     <Parameter name="CSP_Protocol" parameterTypeRef="CSPProtocolFlagType" />
98     <Parameter name="CSP_Integrity" parameterTypeRef="CSPIntegrityFlagType" />
99
100     <Parameter name="sensor" parameterTypeRef="uint8_t" />
101     <Parameter name="temp" parameterTypeRef="uint8_t" />
102     <Parameter name="F_sensor" parameterTypeRef="uint32_t" />
103     <Parameter name="F_reference" parameterTypeRef="uint32_t" />
104     <Parameter name="recharge_count" parameterTypeRef="uint8_t" />
105     <Parameter name="window_factor" parameterTypeRef="float_t" />
106 </ParameterSet>
107 <ContainerSet>
108     <SequenceContainer abstract="true" name="CSPPacket">
109         <EntryList>
110             <ParameterRefEntry parameterRef="CSP_Priority" />
111             <ParameterRefEntry parameterRef="CSP_Source" />
112             <ParameterRefEntry parameterRef="CSP_Destination" />
113             <ParameterRefEntry parameterRef="CSP_DestinationPort" />
114             <ParameterRefEntry parameterRef="CSP_SourcePort" />
115             <ParameterRefEntry parameterRef="CSP_ReservedFlags" />
116             <ParameterRefEntry parameterRef="CSP_Fragmentation" />
117             <ParameterRefEntry parameterRef="CSP_Authentication" />
118             <ParameterRefEntry parameterRef="CSP_Encryption" />
119             <ParameterRefEntry parameterRef="CSP_Protocol" />
120             <ParameterRefEntry parameterRef="CSP_Integrity" />
121         </EntryList>
122     </SequenceContainer>
123     <SequenceContainer name="TelemetryPacket">
124         <EntryList>
125             <ParameterRefEntry parameterRef="sensor" />
126             <ParameterRefEntry parameterRef="temp" />
127             <ParameterRefEntry parameterRef="F_sensor" />
128             <ParameterRefEntry parameterRef="F_reference" />
129             <ParameterRefEntry parameterRef="recharge_count" />
130             <ParameterRefEntry parameterRef="window_factor" />
131         </EntryList>
132     <BaseContainer containerRef="CSPPacket">
133         <RestrictionCriteria>
134             <ComparisonList>
135                 <Comparison value="20" parameterRef="CSP_DestinationPort" />
136             </ComparisonList>
137         </RestrictionCriteria>
138     </BaseContainer>
139 </SequenceContainer>
140 </ContainerSet>
141 </TelemetryMetaData>

```

```

142
143 <CommandMetaData>
144 </CommandMetaData>
145</SpaceSystem>

```

Listing B.3: Yamcs XML code describing the telemetry coming from the floating gate dosimeter via the FlatSat.

### B.1.3. On-Board Computer

```

1 #include "FGDOS_YAMCS.h"
2
3 /*
4 set up the FreeRTOS task and allocate to it a specific memory stack
5 */
6 void init_FGDOS() {
7     static StaticTask_t FGDOS_tcb;
8     static StackType_t FGDOS_stack[500];
9     xTaskCreateStatic(FGDOS_task, "HELLO", 500, NULL, 1,FGDOS_stack, &FGDOS_tcb);
10 }
11
12 /*
13 main task to perform
14 */
15 void FGDOS_task(void * pvParams) {
16     TickType_t xLastWakeTime = xTaskGetTickCount();
17     TickType_t task_interval = 4; // the task interval in seconds
18
19     /* local variables for sending and receiving I2C and CSP */
20     uint8_t I2C_receive_arr[BYTES_I2C], I2C_send_arr[10], CSP_message[CSP_MESSAGE_LENGTH];
21     uint8_t fgdos_sensor, fgdos_rech_count, fgdos_temp;
22     uint32_t fgdos_f_sens, fgdos_f_ref;
23     float fgdos_window_factor;
24     float *p_fgdos_window_factor; // pointer used to reconstruct float from received bytes
25
26     /* set up I2C connection */
27     struct io_descriptor *I2C_1_io;
28     i2c_m_sync_get_io_descriptor(&I2C_1, &I2C_1_io);
29     i2c_m_sync_enable(&I2C_1);
30     i2c_m_sync_set_slaveaddr(&I2C_1, SLAVE_ADDRESS, I2C_M_SEVEN);
31
32     /* watchdog setup */
33     WDT_0_init();
34
35     for (;;) {
36         /*
37          * Main task: Reading data from the FGDOS:
38          * Send I2C command to the Arduino triggering it to send SPI commands to FGDOS to read
39          * array of registers to send over I2C
40          * data_received = sensor, temperature, F_sensor, F_reference, recharge_count,
41          * window_factor
42          * data_received types = byte, byte, 3*byte, 3*byte, byte, float (4*byte) -> 13 bytes
43          */
44         if(io_read(I2C_1_io, I2C_receive_arr, BYTES_I2C)==BYTES_I2C){
45             if((I2C_receive_arr[8] & 0x80)==0x80){
46                 printf("recharge ongoing... WAIT \n");
47             } else{
48                 fgdos_sensor = I2C_receive_arr[0];
49                 fgdos_temp = I2C_receive_arr[1];
50                 fgdos_rech_count = I2C_receive_arr[8] & 0x0F; // only 4 last bits determine
51                 recharge count
52                 p_fgdos_window_factor = &I2C_receive_arr[9];
53                 fgdos_window_factor = *p_fgdos_window_factor;
54                 // frequencies consist of 18 bits (hence 0x3FFFF mask)
55                 fgdos_f_sens = (((unsigned long)(I2C_receive_arr[2] << 8 | I2C_receive_arr[3]
56 ) <<8 | I2C_receive_arr[4] ) & 0x3FFFF ) * fgdos_window_factor;
57                 fgdos_f_ref = (((unsigned long)(I2C_receive_arr[5] << 8 | I2C_receive_arr[6]
58 ) <<8 | I2C_receive_arr[7] ) & 0x3FFFF ) * fgdos_window_factor;
59             }
60             } else{
61                 printf("wrong numbers of I2C bytes transferred ... \n");
62             }
63         }
64     }
65 }
66
67

```

```

58     vTaskDelayUntil(&xLastWakeTime, task_interval*configTICK_RATE_HZ);
59     wdt_feed(&WDT_0); // reset watchdog
60
61     /*
62     * Send CSP packets via TT&C to ground stations for use in Yamcs.
63     */
64     CSP_message[0] = fgdos_sensor;
65     CSP_message[1] = fgdos_temp;
66     memcpy(&CSP_message[2], &fgdos_f_sens, 4);
67     memcpy(&CSP_message[6], &fgdos_f_ref, 4);
68     CSP_message[10] = fgdos_rech_count;
69     memcpy(&CSP_message[11], &fgdos_window_factor, 4);
70     csp_conn_t * connection = csp_connect(CSP_PRIO_NORM, /* the priority of the
71     connection */
72                                         25, /* the target CSP address */
73                                         20, /* the target port (NOTE: YAMCS does not
74     care about this and just takes all CSP packets, whatever the port) */
75                                         1000, /* the timeout for this connection. */
76                                         CSP_O_NONE /* protocol options */
77                                         );
78     if (connection == NULL) { printf("FAILED \n"); }
79     else {
80         csp_packet_t * pkt = csp_buffer_get(CSP_MESSAGE_LENGTH);
81         pkt->length = CSP_MESSAGE_LENGTH;
82         memcpy(pkt->data, CSP_message, CSP_MESSAGE_LENGTH);
83         csp_send(connection, pkt, 1000);
84         csp_close(connection);
85     }
86     vTaskDelayUntil(&xLastWakeTime, task_interval*configTICK_RATE_HZ);
87 }

```

Listing B.4: FreeRTOS task on the OBC for retrieving dosimeter data and sending it to the ground station for plotting in Yamcs.

## B.2. Arduino

### B.2.1. Basic Functions

```

1 void fgdos_init(byte sensor){
2     // set the register settings
3     write_reg(sensor, xB_RECHARGE_WINDOW, xB_settings_a);
4     write_reg(sensor, xC_CHARGE_SENS, xC_settings);
5     write_reg(sensor, xE_NIRQ_ENGATE, xE_settings);
6     write_reg(sensor, xD_RECHARGE_REF, xD_settings_off);
7     // wait for reference to stabilise, e.g. 4 measurement windows
8     wait(4/WINDOW_FACTOR * 1000);
9     // set target and threshold registers by selecting 5 MSBs (/BITSHIFT) and set them to the
10    appropriate registers
11    byte reg_target = floor(TARGET_FREQ/WINDOW_FACTOR/BITSHIFT);
12    write_reg(sensor, x9_TARGET, reg_target);
13    byte reg_threshold = floor(THRESHOLD_FREQ/WINDOW_FACTOR/BITSHIFT);
14    write_reg(sensor, xA_THRESHOLD, reg_threshold);
15    write_reg(sensor, xD_RECHARGE_REF, xD_settings_on);
16    // wait for registers to update ( 2 measurement windows +10 recommended by Sealicon
17    )wait(2.1/WINDOW_FACTOR * 1000);

```

Listing B.5: FGD-03F sensor chip setup function.

```

1 // Function to read a register from the sensor. First, send the address, then read what is
2 // being send back
3 unsigned int read_reg(byte sensor, byte reg){
4     byte data = 0;
5     byte adr = reg|RD;
6     digitalWrite(sensor, LOW);
7     SPI.transfer(adr);
8     //delayMicroseconds(10);
9     data = SPI.transfer(0x00);
10    digitalWrite(sensor, HIGH);
11    return data;
12 }
13 // write to a register

```

```

13 void write_reg(byte sensor, byte reg, byte data){
14     byte adr = reg|WR;
15     digitalWrite(sensor, LOW);
16     SPI.transfer(adr);
17     SPI.transfer(data);
18     digitalWrite(sensor, HIGH);
19 }

```

Listing B.6: Read and write functions to interact with dosimeter registers via SPI.

## B.2.2. FlatSat-Dosimeter Interface

```

1 void setup() {
2     Serial.begin(250000);
3     while(!Serial);
4
5     //----- PWM GENERATION -----
6     pinMode(9, OUTPUT);
7     TCCR1A = _BV(COM1A0) | _BV(WGM11) | _BV(WGM10);
8     TCCR1B = _BV(WGM13) | _BV(WGM12) | _BV(CS11); // set CS2x to the desired bit to change
9     // divider for frequency (CS21 = 8)
10    OCR1A = 31; // freq = 16MHz / divider / (OCR2A +1) (= 31.25 kHz)
11    OCR1B = 15; // duty cycle = OCR2B+1 / OCR2A+1
12
13    // ----- SPI SETUP -----
14    SPI.beginTransaction(SPISettings(500000, MSBFIRST, SPI_MODE0));
15    SPI.begin();
16    pinMode(SS1, OUTPUT); //chip enable, active low
17    pinMode(SS2, OUTPUT);
18    pinMode(10,OUTPUT); //set SS pin to output to prevent Arduino from going into slave mode
19    digitalWrite(SS1, HIGH);
20    digitalWrite(SS2, HIGH);
21
22    // ----- I2C SETUP -----
23    Wire.begin(0x25); // set slave address
24    Wire.onReceive(receiveEvent);
25    Wire.onRequest(requestEvent);
26
27    // ----- SENSOR SETUP -----
28    fgdos_init(SS2);
29 }
30
31 void loop() {
32     if (flag_receive == true){
33         reg = I2C_in[0];
34         switch(i){
35             case 2:
36                 data8 = I2C_in[1];
37                 write_reg(SS2,reg,data8);
38                 break;
39             default:
40                 Serial.println("data type not recognised...");
41         }
42         flag_receive = false;
43     }
44 }
45
46 void receiveEvent()
47 {
48     i = 0;
49     while( Wire.available()){
50         I2C_in[i] = Wire.read();
51         ++i;
52     }
53     flag_receive = true;
54 }
55
56 void requestEvent()
57 {
58     temperature = read_reg(SS2,TEMP);
59     recharge_count = read_reg(SS2,RECHARGE_COUNT);

```

```

59 collect_freq(SS2,&sens_freq,&ref_freq);
60 I2C_out[0] = 2;
61 I2C_out[1] = temperature;
62 I2C_out[8] = recharge_count;
63 float_to_bytes(WINDOW_FACTOR, &I2C_out[9]);
64
65 Wire.write(I2C_out,BYTES_SEND);
66 flag_request = true;
67 }
68
69 /*
70  * read the frequency registers and reconstruct their values
71  * frequency = registers / window pulses amount * ck frequency
72  */
73 bool collect_freq(byte sensor, long int *sens_freq, long int *ref_freq){
74     bool have_sens_freq = false;
75     bool have_ref_freq = false;
76     byte freq_reg[3];
77     unsigned int time_start=millis();
78
79     if((read_reg(sensor,RECHARGE_COUNT) & 0x80) == 0x80){
80         Serial.println("recharge in progress... BREAK");
81         return false;
82     }
83     while(!have_sens_freq || !have_ref_freq){
84         if(!have_sens_freq){
85             // sensor frequency
86             freq_reg[0] = read_reg(sensor,0x08);
87             freq_reg[1] = read_reg(sensor,0x07);
88             freq_reg[2] = read_reg(sensor,0x06);
89             *sens_freq = (((unsigned long)(freq_reg[0] << 8 | freq_reg[1] ) << 8 | freq_reg[2] ) &
FREQ_MASK );
90             have_sens_freq = true;
91             *sens_freq = *sens_freq * WINDOW_FACTOR;
92
93             I2C_out[2] = freq_reg[0];
94             I2C_out[3] = freq_reg[1];
95             I2C_out[4] = freq_reg[2];
96         }
97         if(!have_ref_freq){
98             // reference frequency
99             freq_reg[0] = read_reg(sensor,0x05);
100            freq_reg[1] = read_reg(sensor,0x04);
101            freq_reg[2] = read_reg(sensor,0x03);
102            *ref_freq = (((unsigned long)(freq_reg[0] << 8 | freq_reg[1] ) << 8 | freq_reg[2] ) &
FREQ_MASK );
103            have_ref_freq = true;
104            *ref_freq = *ref_freq * WINDOW_FACTOR;
105
106            I2C_out[5] = freq_reg[0];
107            I2C_out[6] = freq_reg[1];
108            I2C_out[7] = freq_reg[2];
109        }
110        if ((millis()-time_start>(4*1/WINDOW_FACTOR*x) & !have_ref_freq & !have_sens_freq)){
111            Serial.println("reading failed, timeout");
112            return false;
113        }
114    }
115    return true;
116 }

```

Listing B.7: Arduino program main loop to act as an intermediary between the FlatSat and the FGD-02F dosimeter sensor chip.

```

1
2 // CHOOSE SENSITIVITY $1&$ ISR $1&$ WINDOW
3 #define HIGHSENS
4 // #define LOWSENS
5 #define FGD_ISR true
6 //#define FGD_ISR false
7 #define WINDOW_PULSES 4096
8 //#define WINDOW_PULSES 8192
9 //#define WINDOW_PULSES 16384

```

```

10 //#define WINDOW_PULSES 32768
11
12 // Register definitions
13 #define x0_TEMP 0x00
14 #define x1_RECHARGE_COUNT 0x01
15 #define x9_TARGET 0x09
16 #define xA_THRESHOLD 0x0A
17 #define xB_RECHARGE_WINDOW 0x0B
18 #define xC_CHARGE_SENS 0x0C
19 #define xD_RECHARGE_REF 0x0D
20 #define xE_NIRQ_ENGATE 0x0E
21
22 // sensor constant definitions (these depend on settings in Arduino and sensor!)
23 #define CK_FREQ 31250.0f // depends on the settings of the PWM
24 #define WINDOW_FACTOR (CK_FREQ/WINDOW_PULSES) // one should be defined as float, otherwise
rounding errors
25
26 // register overall valid settings
27 // nirq setting to push-pull (nirqoc bit 0) or open collector (1), measurement window to
count clocks (engate bit 0)
28 // edirt bit to 1 to only measure when no SPI is ongoing
29 #define xE_settings 0x06
30 // set the reference oscillator and window measurement amount of pulses settings
31 // bits (8:4) for recharging (enable autorech, internal pump at VB, rech by pump, 0) and
(3:2) for window (bit counting from lsb to msb)
32 // windows set to 11 (4096 pulses)(00=32768 ck pulses per window)
33 // TDIV (bit 0) to 1 for more precise frequency range, ONLY FOR LOWEST 2 AMOUNTS OF PULSES
USED
34 // bitshift is used for threshold and target register comparison, depends on TDIV (if 1, then
10 shifts, if 0 then 13)
35 // Cx default, 4x to disable auto recharge (called active and passive but is confusing, I
know...)
36 #if WINDOW_PULSES == 4096
37 // TDIV to 1
38 #define xB_settings_a 0xCD
39 #define xB_settings_p 0x4D
40 #define BITSHIFT 1024 // is 10 bit shifts
41 #elif WINDOW_PULSES == 8192
42 // TDIV to 1
43 #define xB_settings_a 0xC9
44 #define xB_settings_p 0x49
45 #define BITSHIFT 1024
46 #elif WINDOW_PULSES == 16384
47 // TDIV to determine
48 #define xB_settings_a 0xC5
49 #define xB_settings_p 0x45
50 #define BITSHIFT 8192
51 #elif WINDOW_PULSES == 32768
52 // TDIV to 0
53 #define xB_settings_a 0xC0
54 #define xB_settings_p 0x40
55 #define BITSHIFT 8192
56 #endif
57
58 #ifdef HIGHSENS
59 #define SENS "high"
60 #define THRESHOLD_FREQ 50000
61 #define TARGET_FREQ 90000
62 // manual recharge off and sensitivity to high
63 // MSB to switch on or off manual recharge, 3 LSBs to set sensitivity (100 low, 001 high)
64 #define xC_settings 0x79
65 // disconnect recharging system before setting targets, set pump level (for shortest recharge
time bit 3:0 to 111)
66 // the recharge voltage of 100 (16.5 V, ...4) seems to be good, otherwise to fast and
overshoots in recharging
67 // set reference close to target, according to sensitivity (bit 3 to 1 for lower values)
68 #define xD_settings_off 0x04
69 // allow recharges again
70 #define xD_settings_on 0x44
71 // for enabling manual recharges
72 #define xC_settings_manual 0xF9

```

```

73 #define xD_settings_manual 0x64
74 #endif
75
76 #ifndef LOWSENS
77 #define SENS "low"
78 #define THRESHOLD_FREQ 140000
79 #define TARGET_FREQ 180000
80 #define xC_settings 0x7C
81 #define xD_settings_off 0x04
82 #define xD_settings_on 0x44
83 #define xC_settings_manual 0xFC
84 #define xD_settings_manual 0x64
85 #endif

```

Listing B.8: Arduino code in which the register addresses and values are mapped onto more comprehensible variable names for the FGD-03F sensor chip.

### B.2.3. Radiation Tests

```

1 void setup() {
2   // set up serial connection with PC
3   Serial.begin(250000);
4   while(!Serial);
5
6   standby_passive_pins_setup();
7   pwm_setup();
8   spi_setup();
9   wait(1000);
10
11  fgdos_init(SS2);
12  fgdos_init(SS1);
13
14  // ISR upon new data, interrupt request data ready, active low
15  // Arduino pullup used and sensor set to open collector (can only pull low)
16  if (flag_isr){
17    pinMode(NIRQ_1, INPUT_PULLUP);
18    pinMode(NIRQ_2, INPUT_PULLUP);
19    attachInterrupt(digitalPinToInterrupt(NIRQ_1), collect_data_SS1, LOW);
20    attachInterrupt(digitalPinToInterrupt(NIRQ_2), collect_data_SS2, LOW);
21  }
22 }
23
24 void loop() {
25   // Collect data in a loop when ISR is not used
26   // NIRQ does not work during recharges, collect data via loop during recharges
27   if (!flag_isr){
28     collect_data(SS1, &temperature_1, &recharge_count_1, &sens_freq_1, &ref_freq_1,
29     WINDOW_FACTOR);
30     collect_data(SS2, &temperature_2, &recharge_count_2, &sens_freq_2, &ref_freq_2,
31     WINDOW_FACTOR);
32     //wait at least 2.1 windows for new data (in milliseconds 2/WINDOW_FACTOR * 1000)
33     wait(2.1/WINDOW_FACTOR * 1000);
34   } else if (((recharge_count_1 & 0x80) == 0x80) || ((recharge_count_2 & 0x80) == 0x80)){
35     if ((recharge_count_1 & 0x80) == 0x80){collect_data(SS1, &temperature_1, &
36     recharge_count_1, &sens_freq_1, &ref_freq_1, WINDOW_FACTOR);}
37     if ((recharge_count_2 & 0x80) == 0x80){collect_data(SS2, &temperature_2, &
38     recharge_count_2, &sens_freq_2, &ref_freq_2, WINDOW_FACTOR);}
39     wait(2.1/WINDOW_FACTOR*1000);
40     Serial.println("extra loop running");
41   }
42 }
43
44 // Option to change sensor settings
45 command = get_command();
46 set_command(command, &sens_freq_1, &sens_freq_2, &target_freq, &threshold_freq, &
47   ref_freq_1, &ref_freq_2,
48   &temperature_1, &temperature_2, &recharge_count_1, &recharge_count_2, flag_isr);
49 }
50
51 void collect_data_SS1(){
52   collect_data(SS1, &temperature_1, &recharge_count_1, &sens_freq_1, &ref_freq_1,
53   WINDOW_FACTOR);

```

```

47 }
48 void collect_data_SS2(){
49     collect_data(SS2, &temperature_2, &recharge_count_2, &sens_freq_2, &ref_freq_2,
50     WINDOW_FACTOR);

```

Listing B.9: Arduino program to interact with the FGD-03F sensor chip during the radiation tests.

```

1 char get_command(){
2     String command_str;
3     while(Serial.available()){
4         command_str = Serial.readString();
5     }
6     return command_str[0];
7 }
8
9 void set_command(char command, unsigned long int *sens_freq_1, unsigned long int *sens_freq_2
, unsigned long int *target_freq,
10     unsigned long int *threshold_freq, unsigned long int *ref_freq_1, unsigned
long int *ref_freq_2,
11     int *temperature_1, int *temperature_2, byte *recharge_count_1, byte *
recharge_count_2, bool flag_isr)
12 {
13     // flags to remember on or off states
14     static bool flag_discharge_1 = true, flag_discharge_2 = true, flag_active_1 = true,
flag_on_1 = true, flag_on_2 = true,
15     flag_active_2 = true, flag_recharge_1 = true, flag_recharge_2 = true;
16     //deactivate interrupt before to prevent collision
17     if (flag_isr){
18         detachInterrupt(digitalPinToInterrupt(NIRQ_1));
19         detachInterrupt(digitalPinToInterrupt(NIRQ_2));
20         pinMode(NIRQ_1,INPUT);
21         pinMode(NIRQ_2,INPUT);
22     }
23     switch (command){
24         case '1':
25             collect_data(SS1, temperature_1, recharge_count_2, sens_freq_1, ref_freq_1,
WINDOW_FACTOR);
26             break;
27         case '2':
28             collect_data(SS2, temperature_2, recharge_count_2, sens_freq_2, ref_freq_2,
WINDOW_FACTOR);
29             break;
30         case 't':
31             // T for target and threshold
32             collect_range(SS1, target_freq, threshold_freq);
33             Serial.print("TARGET 1: "); Serial.print(*target_freq);
34             Serial.print(", THRESHOLD 1: "); Serial.println(*threshold_freq);
35             break;
36         case 'T':
37             collect_range(SS2, target_freq, threshold_freq);
38             Serial.print("TARGET 2: "); Serial.print(*target_freq);
39             Serial.print(", THRESHOLD 2: "); Serial.println(*threshold_freq);
40             break;
41         case 'o':
42             if (!flag_on_1){
43                 digitalWrite(NSTBY_1,HIGH);
44                 Serial.println("ON 1");
45                 flag_on_1 = true;
46             } else {
47                 digitalWrite(NSTBY_1,LOW);
48                 Serial.println("STANDBY 1");
49                 flag_on_1 = false;
50             }
51             break;
52         case 'O':
53             if (!flag_on_2){
54                 digitalWrite(NSTBY_2,HIGH);
55                 Serial.println("ON 2");
56                 flag_on_2 = true;
57             } else {
58                 digitalWrite(NSTBY_2,LOW);

```

```
59     Serial.println("STANDBY 2");
60     flag_on_2 = false;
61 }
62 break;
63 case 'a':
64     if (!flag_active_1){
65         digitalWrite(PASSIVE,LOW);
66         Serial.println("ACTIVE 1");
67         flag_active_1 = true;
68     } else {
69         digitalWrite(PASSIVE,HIGH);
70         Serial.println("PASSIVE 1");
71         flag_active_1 = false;
72     }
73     break;
74 case 'A':
75     if (!flag_active_2){
76         digitalWrite(PASSIVE,LOW);
77         Serial.println("ACTIVE 2");
78         flag_active_2 = true;
79     } else {
80         digitalWrite(PASSIVE,HIGH);
81         Serial.println("PASSIVE 2");
82         flag_active_2 = false;
83     }
84     break;
85 case 'r':
86     if (flag_recharge_1){
87         recharge_enable(SS1);
88         Serial.println("RECHARGING STARTED 1");
89         flag_recharge_1 = false;
90     } else {
91         recharge_disable(SS1);
92         Serial.println("RECHARGING STOPPED 1");
93         flag_recharge_1 = true;
94     }
95     break;
96 case 'R':
97     if (flag_recharge_2){
98         recharge_enable(SS2);
99         Serial.println("RECHARGING STARTED 2");
100        flag_recharge_2 = false;
101    } else {
102        recharge_disable(SS2);
103        Serial.println("RECHARGING STOPPED 2");
104        flag_recharge_2 = true;
105    }
106    break;
107 case 'd':
108     if (flag_discharge_1){
109         discharge_enable(SS1);
110         Serial.println("discharging ... 1");
111         flag_discharge_1 = false;
112     } else {
113         discharge_disable(SS1);
114         Serial.println("DISCHARGING STOPPED 1");
115         flag_discharge_1 = true;
116     }
117     break;
118 case 'D':
119     if (flag_discharge_2){
120         discharge_enable(SS2);
121         Serial.println("discharging ... 2");
122         flag_discharge_2 = false;
123     } else {
124         discharge_disable(SS2);
125         Serial.println("DISCHARGING STOPPED 2");
126         flag_discharge_2 = true;
127     }
128     break;
129 case 'i':
```

```

130     // all settings reset
131     Serial.print("RESET: ");
132     fgdos_init(SS1);
133     break;
134 case 'I':
135     // all settings reset
136     Serial.print("RESET: ");
137     fgdos_init(SS2);
138     break;
139 case 'I':
140     fgdos_init_variable(SS1,"low","active");
141     break;
142 case 'L':
143     fgdos_init_variable(SS2,"low","active");
144     break;
145 case 'h':
146     fgdos_init_variable(SS1,"high","active");
147     break;
148 case 'H':
149     fgdos_init_variable(SS2,"high","active");
150     break;
151 default:
152     break;
153 }
154 if (flag_isr){
155     if (flag_on_1){
156         pinMode(NIRQ_1, INPUT_PULLUP);
157         attachInterrupt(digitalPinToInterrupt(NIRQ_1), collect_data_SS1, LOW);
158     }
159     if (flag_on_2){
160         pinMode(NIRQ_2, INPUT_PULLUP);
161         attachInterrupt(digitalPinToInterrupt(NIRQ_2), collect_data_SS2, LOW);
162     }
163 }
164 }

```

Listing B.10: Arduino functions to make commands possible and control the sensor chip's settings.

## B.3. SAMV71

### B.3.1. Radiation Tests

```

1 #include <atmel_start.h>
2 #include <stdio.h>
3
4 #define INA219_1_ADDRESS 0x40
5 #define INA219_2_ADDRESS 0x41
6
7 #define INA219_CONFIG_BVOLTAGERANGE_16V 0x0000 // 16 V bus range
8 #define INA219_CONFIG_GAIN_1_40MV 0x0000 // Gain 1, 40 mV
9 #define INA219_CONFIG_BADCRES_12BIT 0x0180 // 1 x 12-bit shunt sample
10 #define INA219_CONFIG_SADCRES_12BIT_1S_532US 0x0018 // 1 x 12-bit shunt sample
11 #define INA219_CONFIG_MODE_SANDBVOLT_CONTINUOUS 0x07 // < shunt and bus voltage continuous
12
13 #define INA219_REG_SHUNTVOLTAGE (0x01)
14 #define INA219_REG_BUSVOLTAGE (0x02)
15 #define INA219_REG_POWER (0x03)
16 #define INA219_REG_CURRENT (0x04)
17 #define INA219_REG_CONFIG 0x00
18 #define INA219_REG_CALIBRATION 0x05
19
20 #define INA219_VSHUNT_PGA1_MASK 0x8FFF
21 #define INA219_VBUS_MASK 0xFFF8
22
23 #define STR_DATA_SIZE 60
24 #define STR_INFO_SIZE 80
25
26 void retrieve_voltage(uint8_t sensor, uint16_t *V_bus, int16_t *V_shunt);
27
28 int main(void)
29 {

```

```

30  /* Initializes MCU, drivers and middleware */
31  atmel_start_init();
32
33  /* initialize USART for PC communication, variable definitions */
34  struct io_descriptor *io_serial;
35  usart_sync_get_io_descriptor($1&$SERIAL, $1&$io_serial);
36  usart_sync_enable($1&$SERIAL);
37  char str_data[STR_DATA_SIZE], str_info[STR_INFO_SIZE], vbus_str[10], vshunt_str[10];
38
39  /* I2C (TWI) setup */
40  struct io_descriptor *I2C_0_io;
41  i2c_m_sync_get_io_descriptor($1&$I2C_0, $1&$I2C_0_io);
42  i2c_m_sync_enable($1&$I2C_0);
43  int16_t V_shunt_1 = 0, V_shunt_2 = 0;
44  uint16_t V_bus_1 = 0, V_bus_2 = 0;
45  bool flag_shunt = true, flag_bus = true;
46  /* set configuration register for both sensors
47   * also set the calibration register if you want to measure current and power directly
48   */
49  uint16_t config = INA219_CONFIG_BVOLTAGERANGE_16V |
50  INA219_CONFIG_GAIN_1_40MV | INA219_CONFIG_BADCRES_12BIT |
51  INA219_CONFIG_SADCRES_12BIT_1S_532US |
52  INA219_CONFIG_MODE_SANDBVOLT_CONTINUOUS;
53  i2c_m_sync_set_slaveaddr($1&$I2C_0, INA219_1_ADDRESS, I2C_M_SEVEN);
54  i2c_m_sync_cmd_write($1&$I2C_0, INA219_REG_CONFIG, $1&$config, 2);
55  i2c_m_sync_set_slaveaddr($1&$I2C_0, INA219_2_ADDRESS, I2C_M_SEVEN);
56  i2c_m_sync_cmd_write($1&$I2C_0, INA219_REG_CONFIG, $1&$config, 2);
57
58  while (1) {
59      // sensor 1 $1&$ 2 measurements
60      retrieve_voltage(INA219_1_ADDRESS, $1&$V_bus_1, $1&$V_shunt_1);
61      retrieve_voltage(INA219_2_ADDRESS, $1&$V_bus_2, $1&$V_shunt_2);
62      snprintf(str_data, STR_DATA_SIZE, "Vs1(uV),d,Vb1(mV),d,Vs2(uV),d,Vb2(mV),d\n", V_shunt_1,
63      V_bus_1, V_shunt_2, V_bus_2);
64      io_write(io_serial, str_data, STR_DATA_SIZE);
65      while(!usart_sync_is_tx_empty(io_serial)){}
66      //when string is shorter than previous one, end still contains values, so reset string
67      memset(str_data, 0, STR_DATA_SIZE);
68  }
69
70 void retrieve_voltage(uint8_t address, uint16_t *V_bus, int16_t *V_shunt){
71  i2c_m_sync_set_slaveaddr($1&$I2C_0, address, I2C_M_SEVEN);
72  i2c_m_sync_cmd_read($1&$I2C_0, INA219_REG_SHUNTVOLTAGE, V_shunt, 2);
73  delay_us(1); // otherwise bytes are not correctly registered
74  *V_shunt = ((((*V_shunt $1&$ 0x00FF) << 8) | ((*V_shunt $1&$ 0xFF00) >> 8)) $1&$
75  INA219_VSHUNT_PGA1_MASK) *10; // put MSByte first and convert to uV
76  i2c_m_sync_cmd_read($1&$I2C_0, INA219_REG_BUSVOLTAGE, V_bus, 2);
77  delay_us(1);
78  *V_bus = ((((*V_bus $1&$ 0x00FF) << 8) | ((*V_bus $1&$ 0xFF00) >> 8)) >> 3) *4; // put
79  MSByte first and convert to mV

```

Listing B.11: SAMV71 code to measure shunt and bus voltages using two INA219 sensors.



# Bibliography

- [1] European Space Agency. Solar cycle 25: the Sun wakes up, October 2020. URL [https://www.esa.int/Safety\\_Security/Solar\\_cycle\\_25\\_the\\_Sun\\_wakes\\_up](https://www.esa.int/Safety_Security/Solar_cycle_25_the_Sun_wakes_up).
- [2] Syed Naeem Ahmed. *Physics and Engineering of Radiation Detection*. Elsevier Inc., 2015. ISBN 978-0-12-801363-2.
- [3] Maite Alvarez, Carlos Hernando, Joan Cesari, Alvaro Pineda, and Eugeni Garcia-Moreno. Total Ionizing Dose Characterization of a Prototype Floating Gate MOSFET Dosimeter for Space Applications. *IEEE Transactions on Nuclear Science*, 60(6):4281–4288, 2013. ISSN 00189499. doi: 10.1109/TNS.2013.2288573.
- [4] Arduino, n.d.. URL <https://www.arduino.cc/>. Retrieved October 2021.
- [5] Arduino. *ARDUINO UNO REV3*, n.d.. URL <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>. Retrieved August 2021.
- [6] Atmel. *Atmega328p*, n.d.. URL [https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf). Retrieved November 2021.
- [7] T. Baca, M. Jilek, I. Vertat, M. Urban, O. Nentvich, R. Filgas, C. Granja, A. Inneman, and V. Daniel. Timepix in LEO Orbit onboard the VZLUSAT-1 Nanosatellite: 1-year of Space Radiation Dosimetry Measurements. *Journal of Instrumentation*, 13(11), 2018. ISSN 17480221. doi: 10.1088/1748-0221/13/11/C11010.
- [8] J. J. Beatty, D. J. Ficenec, S. Tobias, J. W. Mitchell, S. McKee, S. Nutter, G. Tarle, A. Tomasch, J. Clem, T. G. Guzik, M. Lijowski, J. P. Wefel, C. R. Bower, R. M. Heinz, S. L. Mufson, J. Musser, J. J. Pitts, G. M. Spiczak, S. P. Ahlen, and B. Zhou. The cosmic-ray He-3/He-4 ratio from 100 to 1600 MeV/amu. *The Astrophysical Journal*, 413:268, 8 1993. ISSN 0004-637X. doi: 10.1086/172994.
- [9] K. L. Bedingfield, R. D. Leach, and M. B. Alexander. Spacecraft System Failures and Anomalies Attributed to the Natural Space Environment. Technical report, National Aeronautics and Space Administration, 1996. URL <https://ntrs.nasa.gov/citations/19960050463>.
- [10] M. J. Boschini, C. Consolandi, M. Gervasi, S. Giani, D. Grandi, V. Ivanchenko, S. Pensotti, P. G. Rancoita, and M. Tacconi. Nuclear and Non-Ionizing Energy-Loss for Coulomb Scattered Particles from Low Energy up to Relativistic Regime in Space Radiation Environment. In *Cosmic Rays for Particle and Astroparticle Physics - Proceedings of the 12th ICATPP Conference*, number 11, pages 9–23, 2011. ISBN 9814329029. doi: 10.1142/9789814329033\_0002.
- [11] M. Brucoli. *Total Ionizing Dose Monitoring For Mixed Field Environments*. PhD thesis, Université de Montpellier, 2018.
- [12] M. Brucoli, S. Danzeca, M. Brugger, A. Masi, A. Pineda, J. Cesari, and L. Dusseau. A complete qualification of floating gate dosimeter for CERN applications. In *Radiation and its Effects on Components and Systems (RADECS)*, volume 2016-September, pages 1–4. Institute of Electrical and Electronics Engineers Inc., October 2016. ISBN 9781509043668. doi: 10.1109/RADECS.2016.8093162.
- [13] M. Brucoli, S. Danzeca, M. Brugger, A. Masi, A. Pineda, J. Cesari, L. Dusseau, and F. Wrobel. Floating Gate Dosimeter Suitability for Accelerator-Like Environments. *IEEE Transactions on Nuclear Science*, 64:2054–2060, August 2017. ISSN 00189499. doi: 10.1109/TNS.2017.2681651.

- [14] M. Brucoli, S. Danzeca, J. Cesari, M. Brugger, A. Masi, S. Gilardoni, A. Pineda, L. Dusseau, and F. Wrobel. Investigation on the Sensitivity Degradation of Dosimeters based on Floating Gate Structure. In *Radiation and its Effects on Components and Systems (RADECS)*, 2017.
- [15] M. Brucoli, J. Cesari, S. Danzeca, M. Brugger, A. Masi, A. Pineda, L. Dusseau, and F. Wrobel. Investigation on Passive and Autonomous Mode Operation of Floating Gate Dosimeters. *IEEE Transactions on Nuclear Science*, 66:1620–1627, July 2019. ISSN 15581578. doi: 10.1109/TNS.2019.2895366.
- [16] M. Brucoli, S. Danzeca, A. Waage, A. Masi, R. Garcia Alia, B. Severa Mas, A. Pineda, and V. Ferlet-Cavrois. Heavy-Ion Charge Yield Measurement by Floating Gate Dosimeters. In *Radiation and its Effects on Components and Systems (RADECS)*, 2021.
- [17] California Polytechnic State University. Cubesat Design Specification, 2020. URL <https://www.cubesat.org/cds-announcement>.
- [18] California Polytechnic State University. CubeSat, n.d. URL <https://www.cubesat.org/>. Retrieved May 2021.
- [19] E. Camporeale. The Challenge of Machine Learning in Space Weather: Nowcasting and Forecasting. *Space Weather*, 17(8):1166–1207, 2019. ISSN 15427390. doi: 10.1029/2018SW002061.
- [20] Enrico Camporeale, Jay Johnson, and Simon Wing. *Machine Learning Techniques for Space Weather*. Elsevier, 2018. ISBN 978-0-12-811788-0.
- [21] Chantal Cappelletti, Simone Battistini, and Benjamin Malphrus. *CubeSat Handbook: From Mission Design to Operations*. Academic Press, sep 2020. ISBN 9780128178843.
- [22] J. Cesari, A. Barbancho, A. Pineda, G. Ruy, and H. Moser. Floating Gate Dosimeter Measurements at 4M Lunar Flyby Mission. *IEEE Radiation Effects Data Workshop*, 2015-Novem:15–18, 2015. doi: 10.1109/REDW.2015.7336710.
- [23] Joan Cesari, Matteo Brucoli, Salvatore Danzeca, Álvaro Pineda, Alessandro Masi, Markus Brugger, Simone Gilardoni, Eugeni Isern, Miquel Roca, and Enrique García-Moreno. Study of Floating Gate MOS Structures to improve the noise and sensitivity as Radiation Dosimeter. *17th European Conference on Radiation and Its Effects on Components and Systems (RADECS)*, pages 1–4, 2017. doi: 10.1109/RADECS.2017.8696254.
- [24] Giacomo Curzi, Dario Modenini, and Paolo Tortora. Large Constellations of Small Satellites: A Survey of Near Future Challenges and Missions. *Aerospace*, 7(9), 2020. ISSN 22264310. doi: 10.3390/AEROSPACE7090133.
- [25] E. J. Daly, A. Hilgers, G. Drolshagen, and H. D.R. Evans. Space Environment Analysis: Experience and Trends. *European Space Agency, (Special Publication) ESA SP*, (392):15–22, 1996. ISSN 03796566.
- [26] S. Danzeca, J. Cesari, M. Brugger, L. Dusseau, A. Masi, A. Pineda, and G. Spiezia. Characterization and Modeling of a Floating Gate Dosimeter with Gamma and Protons at Various Energies. *IEEE Transactions on Nuclear Science*, 61(6):3451–3457, 2014. ISSN 00189499. doi: 10.1109/TNS.2014.2364274.
- [27] TU Delft. Lunar Zebro, n.d. URL <https://zebro.space/>. Retrieved November 2021.
- [28] L. Desorgher, W Hajdas, K Egli, X Guo, Y Luo, F Chastellain, C Pereira, D Meier, R. Muff, D. Boscher, and G. Maehlum. The Next Generation Radiation Monitor- NGRM. In *2013 IEEE Nuclear Science Symposium and Medical Imaging Conference (2013 NSS/MIC)*, pages 1–6. IEEE, 2013. doi: 10.1109/NSSMIC.2013.6829497.
- [29] Mikhail Dobynde, Behnoosh Meskoob, Kir Latyshev, Gleb Lavrinov, Elizaveta Perchenko, Anton Ivanov, and Tatiana Podladchikova. Improving Trapped Proton Model on the Low-Earth Orbit with CCSR CubeSats Measurements. In *70th International Astronautical Congress*, pages 1–8, 2019.

- [30] Erwin De Donder. SPENVIS Tutorial Radiation Package, 2010. SPENVIS Workshop Presentation.
- [31] Paul V. Dressendorfer, Joe E. Mazur, James R. Schwank, Todd R. Weatherford, and Christian Poivey. Radiation Effects – From Particles to Payloads. In *IEEE Nuclear and Space Radiation Effects*. Institute of Electrical and Electronics Engineers, Inc., 2002.
- [32] Robert Ecoffet. On-Orbit Anomalies Investigations and Root Cause Determination, 2010. URL <https://slideplayer.com/slide/4612472/>.
- [33] Robert Ecoffet. Overview of In-Orbit Radiation Induced Spacecraft Anomalies. *IEEE Transactions on Nuclear Science*, 60(3):1791–1815, 2013. ISSN 00189499. doi: 10.1109/TNS.2013.2262002.
- [34] European Organization for Nuclear Research. Medipix, n.d. URL <https://medipix.web.cern.ch/home>. Retrieved June 2021.
- [35] European Space Agency. ESA - Dependability, n.d. URL [https://www.esa.int/Enabling\\_Support/Space\\_Engineering\\_Technology/Dependability](https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Dependability). Retrieved May 2021.
- [36] Federal Aviation Administration. The Space Environment, 2018. URL <https://www.faa.gov/about/office{ }org/headquarters{ }offices/avs/offices/aam/cami/library/online{ }libraries/aerospace{ }medicine/tutorial/section3/spacecraft{ }design/>.
- [37] Timothy Finn, Antigoni Georgiadou, Redouane Boumghar, and Jose Antonio Martinez Heras. Machine Learning Methods for Radiation Belts Profile Predictions. In *15th International Conference on Space Operations, 2018*, pages 1–11, June 2018. ISBN 9781624105623. doi: 10.2514/6.2018-2639.
- [38] Belgian Institute for Space Aeronomy. The Space Environment Information System, n.d. URL <https://www.spennis.oma.be/>. Retrieved June 2021.
- [39] European Cooperation for Space Standardization. Space engineering: Space environment. Technical report, ESA Requirements and Standards Office, 6 2020. Standard Revision 1.
- [40] FreeRTOS, n.d. URL <https://freertos.org/>. Retrieved June 2021.
- [41] Mariano Garcia-Inza, Sebastian Carbonetto, and Adrian Faigon. MOS Devices and Integrated Circuits for Ionizing Radiation Dosimetry: A Review. *2020 Argentine Conference on Electronics, CAE 2020*, pages 31–40, 2020. doi: 10.1109/CAE48787.2020.9046369.
- [42] E. Garcia-Moreno, E Isern, M Roca, R Picos, J Font, J Cesari, and A Pineda. Floating Gate CMOS Dosimeter With Frequency Output. *IEEE Transactions on Nuclear Science*, 59(2):373–378, 2012.
- [43] E. Garcia-Moreno, E. Isern, M. Roca, R. Picos, J. Font, J. Cesari, and A. Pineda. Temperature Compensated Floating Gate MOS Radiation Sensor With Current Output. *IEEE Transactions on Nuclear Science*, 60:4026–4030, 2013. ISSN 00189499. doi: 10.1109/TNS.2013.2277605.
- [44] Eberhard Gill. Space Systems Engineering. Delft University of Technology Lecture, 2018.
- [45] G.P. Ginet, T.P. O'Brien, S.L. Huston, W.R. Johnston, T.B. Guild, R. Friedel, C.D. Lindstrom, C.J. Roth, R.A. Quinn, D. Madden, S. Morley, and Yi-Jiun Su. AE9, AP9 and SPM: New Models for Specifying the Trapped Energetic Particle and Space Plasma Environment. *Space Science Reviews*, 2013. doi: 10.1007/978-1-4899-7433-4\_18.
- [46] GomSpace. *CubeSat Space Protocol*, December 2011. URL <https://bytebucket.org/bbruner0/albertasat-on-board-computer/wiki/1.%20Resources/1.1.%20DataSheets/CSP/GS-CSP-1.1.pdf?rev=316ebd49bed49fdbb1d74efdeab74430e7cc726a>.

- [47] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [48] Viyas Gupta. *Analysis of single event radiation effects and fault mechanisms in SRAM, FRAM and NAND Flash: application to the MTCube nanosatellite project*. PhD thesis, Université Montpellier, 2017.
- [49] Daniel Hastings and Henry Garrett. *Spacecraft - Environment Interactions*. Cambridge University Press, 1996. ISBN 0 521 47128 1.
- [50] Hank Heidt, Jordi Puig-Suari, Augustus S Moore, Shinichi Nakasuka, and Robert J Twiggs. CubeSat: A new Generation of Picosatellite for Education and Industry Low-Cost Space Experimentation. *AIAA/USU Conference on Small Satellites*, (10):1–19, 2000.
- [51] Keith E. Holbert. Radiation Effects and Damage, 2008. URL [holbert.faculty.asu.edu/eee560/RadiationEffectsDamage.pdf](http://holbert.faculty.asu.edu/eee560/RadiationEffectsDamage.pdf).
- [52] Andrew Holmes-Siedle and Leonard Adams. RADFET: A review of the use of metal-oxide-silicon devices as integrating dosimeters. *International Journal of Radiation Applications and Instrumentation. Part, 28*(2):235–244, 1986. ISSN 13590197. doi: 10.1016/1359-0197(86)90134-7.
- [53] H. L. Hughes and J. M. Benedetto. Radiation effects and hardening of MOS technology: Devices and circuits. *IEEE Transactions on Nuclear Science*, 50 III(3):500–521, 2003. ISSN 00189499. doi: 10.1109/TNS.2003.812928.
- [54] Adafruit Industries. *INA219 High Side DC Current Sensor Breakout*, n.d. URL <https://www.adafruit.com/product/9043>. Retrieved August 2021.
- [55] Space Inventor, n.d. URL <https://www.space-inventor.com/>. Retrieved October 2021.
- [56] Johnson Space Center. Space Radiation Analysis Group, Johnson Space Center, n.d. URL <https://srag.jsc.nasa.gov/SpaceRadiation/What/What.cfm>. Retrieved May 2021.
- [57] A. H. Johnston and S. M. Guertin. The Effects of Space Radiation on Linear Integrated Circuits. *IEEE Aerospace Conference Proceedings*, 5:363–369, 2000. ISSN 1095323X. doi: 10.1109/aero.2000.878509.
- [58] Arijit Karmakar, Jialei Wang, Jeffrey Prinzie, Valentijn De Smedt, and Paul Leroux. A Review of Semiconductor Based Ionising Radiation Sensors Used in Harsh Radiation Environments and Their Applications. *Radiation*, 1:194–217, August 2021. doi: 10.3390/radiation1030018.
- [59] Lonny Kauder. Spacecraft Thermal Control Coatings References. Technical report, National Aeronautics and Space Administration, 2005. URL <https://ntrs.nasa.gov/api/citations/20070014757/downloads/20070014757.pdf>.
- [60] Philip Kean. SpaceX Starlink Constellation, November 2018. URL <https://www.engineering.com/story/spacex-starlink-constellation>.
- [61] Randy A. Kimble, Paul Goudfrooij, and Ronald L. Gilliland. Radiation Damage Effects on the CCD Detector of the Space Telescope Imaging Spectrograph. *UV, Optical, and IR Space Telescopes and Instruments*, 4013(7):532–544, 2000. doi: 10.1117/12.393973.
- [62] Glenn F. Knoll. *Radiation Detection and Measurement*. John Wiley & Sons, Inc., 2010. ISBN 978-0-470-13148-0.
- [63] Kenneth A Label. Space Radiation Effects on Electronics:, 2004. URL [https://nepp.nasa.gov/files/25295/MRS04\\_LaBel.pdf](https://nepp.nasa.gov/files/25295/MRS04_LaBel.pdf).
- [64] Jet Propulsion Laboratory. JPL | Cubesat | MarCO, n.d. URL <https://www.jpl.nasa.gov/cubesat/missions/marco.php>. Retrieved May 2021.

- [65] Martin Langer and J Boumeester. Reliability of CubeSats – Statistical Data, Developers’ Beliefs and the Way Forward. *Proceedings of 30th Annual AIAA/USU Conference on Small Satellites*, pages 1–12, 2016. URL <https://repository.tudelft.nl/islandora/object/uuid:4c6668ff-c994-467f-a6de-6518f209962e?collection=research>.
- [66] Wiley J. Larson. *Applied Space Systems Engineering*. The McGraw-Hill Companies, Inc, 2009. ISBN 978-0-07-340886-6.
- [67] Wiley J. Larson and James R. Wertz. *Space Mission Analysis and Design*. Microcosm Press; Kluwer Academic Publishers, 3 edition, 2005. ISBN 0-7923-5901-1.
- [68] X. Li, R. S. Selesnick, D. N. Baker, A. N. Jaynes, S. G. Kanekal, Q. Schiller, L. Blum, J. Fennell, and J. B. Blake. Upper limit on the inner radiation belt MeV electron intensity. *Journal of Geophysical Research: Space Physics*, 120(2):1215–1228, 2015. ISSN 21699402. doi: 10.1002/2014JA020777.
- [69] J.J. Lissauer and I. de Pater. *Fundamental Planetary Science: Physics, Chemistry and Habitability*. Cambridge University Press, 2019. doi: 10.1017/9781108304061.
- [70] Robert E Lock, Charles D Edwards Jr, Austin Nicholas, Ryan Woolley, and David J Bell. Small Aerostationary Telecommunications Orbiter Concepts for Mars in the 2020s. In *IEEE Aerospace Conference*, 2016. ISBN 8183932525.
- [71] Sarah Loff. Explorer 1 Overview, 2017. URL [https://www.nasa.gov/mission\\_pages/explorer/explorer-overview.html](https://www.nasa.gov/mission_pages/explorer/explorer-overview.html).
- [72] Alessandra Menicucci. Space Embedded Systems. Delft University of Technology Lecture, 2020.
- [73] Stefano Meroli. *Silicon pixel detectors for high precision measurements*. PhD thesis, Università degli Studi di Perugia, 2011.
- [74] Mark Moldwin. *An Introduction to Space Weather*. Cambridge University Press, 2008. ISBN 978-0-521-86149-6.
- [75] Mariella Moon. Japan’s most powerful X-ray satellite is dead, 2016. URL <https://www.engadget.com/2016-04-29-rip-hitomi.html>.
- [76] NASA Ames Research Center, Small Spacecraft Systems Virtual Institute. State of the Art Small Spacecraft Technology. Technical report, 10 2021. URL <https://www.nasa.gov/smallsat-institute/sst-soa>.
- [77] Carol Norberg. *Human Spaceflight and Exploration*. Springer-Verlag, 2013. ISBN 9783642237256. doi: 10.1007/978-3-642-23725-6.
- [78] National Institute of Standards and Technology. Stopping-Power and Range Tables for Electrons, Protons, and Helium Ions, n.d. URL [nist.gov/pml/stopping-power-range-tables-electrons-protons-and-helium-ions](http://nist.gov/pml/stopping-power-range-tables-electrons-protons-and-helium-ions). Retrieved November 2021.
- [79] T. R. Oldham and F. B. McLean. Total Ionizing Dose Effect in MOS Oxides and Devices. *IEEE Transactions on Nuclear Science*, 50(3):483–499, 2003.
- [80] Daniel L Oltrogge and Kyle Leveque. An Evaluation of CubeSat Orbital Decay. In *25th Annual AIAA/USU Conference on Small Satellites*, 2011. ISBN 6508594621.
- [81] Paolo Pavan, Luca Larcher, and Andrea Marmioli. *Floating Gate Devices: Operation and Compact Modeling*. Kluwer Academic Publishers, 2004. ISBN 1-4020-7731-9.
- [82] Chris Peat. ISS - Orbit, November 2021. URL <https://www.heavens-above.com/orbit.aspx?satid=25544>.
- [83] Mason Peck. Exploring Space with Chip-sized Satellites. *IEEE Spectrum*, July 2011. URL <https://spectrum.ieee.org/exploring-space-with-chipsized-satellites>.

- [84] Joseph N. Pelton, Scott Madry, and Sergio Camacho-Lara. *Handbook of Satellite Applications*. Springer New York, 2013. ISBN 9781441976703. doi: 10.1007/978-1-4419-7671-0\_91.
- [85] Vincent L. Pisacane. *The Space Environment and Its Effects on Space Systems*. American Institute of Aeronautics and Astronautics, Inc, 2008. ISBN 978-1-56347-926-7.
- [86] Planet, n.d. URL <https://www.planet.com/>. Retrieved June 2021.
- [87] Marc Poizat. Total Ionizing Dose (TID): Radiation Environment and its Effects in EEE Components and Hardness Assurance for Space Applications, 2017. URL <https://indico.cern.ch/event/635099/contributions/2570674/attachments/1456398/2249969/Radiation{ }Effects{ }and{ }RHA{ }ESA{ }Course{ }9-10{ }May{ }2017{ }TID{ }MP{ }FINAL{ }WIN.pdf{% }0A>.
- [88] Jim Schwank. Radiation Effects in MOS Oxides. *IEEE Transactions on Nuclear Science*, 55(4):1833–1853, 2008. URL <http://ieeexplore.ieee.org/xpls/abs{ }all.jsp?arnumber=4636929>.
- [89] Sealicon. *FGD-03F*, n.d. URL <https://www.sealiconmicro.com/sealicon-products-documentation-3>. Datasheet Revision 11. Retrieved June 2021.
- [90] Jordanka Semkova, Rositza Koleva, Victor Benghin, Tsvetan Dachev, Yuri Matviichuk, Borislav Tomov, Krasimir Krastev, Stephan Maltchev, Plamen Dimitrov, Nikolay Bankov, Igor Mitrofanov, Alexey Malakhov, Dmitry Golovin, Maxim Mokrousov, Anton Sanin, Maxim Litvak, Alexander Kozyrev, Sergey Nikiforov, Denis Lisov, Artem Anikin, Vyacheslav Shurshakov, and Sergey Drobyshev. Results from radiation environment measurements aboard ExoMars Trace Gas Orbiter in Mars science orbit in May 2018–December 2019. *Icarus*, 361, 6 2021. ISSN 10902643. doi: 10.1016/j.icarus.2020.114264.
- [91] Space Application Services. Yamcs, n.d. URL <https://yamcs.org/>. Retrieved November 2021.
- [92] Deepkumar R Shah. Radiation Effects on MOSFETs, 2016. doi: 10.13140/RG.2.2.12167.29602.
- [93] Doug Sinclair and Jonathan Dyer. Radiation Effects and COTS Parts in SmallSats. In *27th Annual AIAA/USU Conference on Small Satellites*, pages 1–12, 2013. URL <http://digitalcommons.usu.edu/smallsat/2013/all2013/69/>.
- [94] A. G. Smirnov, M. Berrendorf, Y. Y. Shprits, E. A. Kronberg, H. J. Allison, N. A. Aseev, I. S. Zhelavskaya, S. K. Morley, G. D. Reeves, M. R. Carver, and F. Effenberger. Medium Energy Electron Flux in Earth’s Outer Radiation Belt (MERLIN): A Machine Learning Model. *Space Weather*, 18(11):1–20, 2020. ISSN 15427390. doi: 10.1029/2020SW002532.
- [95] Charles Steidel. Hubble Achieves Milestone: 100,000th Exposure, 1996. URL <https://hubblesite.org/contents/news-releases/1996/news-1996-25.html>.
- [96] R. E. Stoller, M. B. Toloczko, G. S. Was, A. G. Certain, S. Dwaraknath, and F. A. Garner. On the Use of SRIM/TRIM for Computing Radiation Damage Exposure. *Nuclear Instruments and Methods in Physics Research, Section B: Beam Interactions with Materials and Atoms*, 310: 75–80, 2013. ISSN 0168583X. doi: 10.1016/j.nimb.2013.05.008.
- [97] Michael Swartwout. CubeSat Database, n.d. URL <https://sites.google.com/a/slu.edu/swartwout/home/cubesat-database>. Retrieved May 2021.
- [98] Brann Tamsyn. MAVEN Maps Electric Currents around Mars that are Fundamental to Atmospheric Loss, May 2020. URL <https://www.nasa.gov/press-release/goddard/2020/mars-electric-currents>.
- [99] B. Todd and S. Uznanski. Radiation Risks and Mitigation in Electronic Systems. *CERN Accelerator School: Power Converters, CAS 2014 - Proceedings*, 003:245–263, May 2018. doi: 10.5170/CERN-2015-003.245.

- [100] P. Truscott, Fan Lei, C. Dyer, B. Quaghebeur, D. Heyndericks, P. Nierninen, H. Evans, E. Daly, A. Mohammadzadeh, and G. Hopkinson. MULASSIS - Monte Carlo Radiation Shielding Simulation for Space Applications Made Easy. In *Proceedings of the 7th European Conference on Radiation and Its Effects on Components and Systems, 2003. RADECS 2003.*, pages 191–196, 2003.
- [101] Nicholas Tsoulfanidis and Sheldon Landsberger. *Measurement and Detection of Radiation*. CRC Press, 4th edition, 2015. ISBN 9781482215489.
- [102] Varadis, n.d.. URL <https://www.varadis.com/>. Retrieved October 2021.
- [103] Varadis. *TECHNICAL DATA RM-VT02 VT02 RADFET Readout Module*, n.d.. URL [https://www.varadis.com/wp-content/uploads/2021/06/RM-VT02\\_Datasheet\\_rev1p1.pdf](https://www.varadis.com/wp-content/uploads/2021/06/RM-VT02_Datasheet_rev1p1.pdf). Datasheet Revision 1. Retrieved October 2021.
- [104] Raoul Velazco, Dale Mcmorrow, and Jaime Estela. *Radiation Effects on Integrated Circuits and Systems for Space Applications*. Springer Nature Switzerland, 2019. ISBN 9783030046590.
- [105] James I. Vette. The AE-8 Trapped Electron Model Environment. Technical report, NASA Goddard Space Flight Center, 1991. URL <https://ntrs.nasa.gov/citations/19920014985>.
- [106] Thyrso Villela, Cesar A. Costa, Alessandra M. Brandão, Fernando T. Bueno, and Rodrigo Leonardi. Towards the Thousandth CubeSat: A Statistical Overview. *International Journal of Aerospace Engineering*, 2019. ISSN 16875974. doi: 10.1155/2019/5063145.
- [107] C. Zeitlin, T. Cleghorn, F. Cucinotta, P. Saganti, V. Andersen, K. Lee, L. Pinsky, W. Atwell, R. Turner, and G. Badhwar. Overview of the Martian Radiation Environment Experiment. *Advances in Space Research*, 33:2204–2210, 2004. ISSN 02731177. doi: 10.1016/S0273-1177(03)00514-3.