



ARISTOTLE
UNIVERSITY
OF THESSALONIKI

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σχεδιασμός μηχανισμού Ανίχνευσης,
Απομόνωσης και Αντιμετώπισης
Βλαβών

στον

Νανοδορυφόρο AcubeSAT

Συγγραφέας: Κωνσταντίνος ΚΑΝΑΒΟΥΡΑΣ (8824)

Επιβλέπων: Καθ. Αλκιβιάδης ΧΑΤΖΟΠΟΥΛΟΣ

*Η διπλωματική εργασία κατατίθεται για την
εκπλήρωση των υποχρεώσεων για λήψη διπλώματος*

στην

Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Copyright © 2021 Κωνσταντίνος Καναβούρας

ΔΗΜΟΣΙΕΥΤΗΚΕ ΑΠΟ ΤΟ ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

[HTTPS://GITHUB.COM/KONGR45OPEN/THESIS-LATEX](https://github.com/kongr45open/thesis-latex)

Αυτή η εργασία χορηγείται με άδεια Creative Commons Αναφορά Δημουργού 4.0 Διεθνές (CC BY 4.0 – η “Άδεια”)· το κείμενο του παρόντος δεν επιτρέπεται να χρησιμοποιηθεί παρά μόνο με βάση την Άδεια. Για να δείτε ένα αντίγραφο αυτής της άδειας, επισκεφτείτε το <https://creativecommons.org/licenses/by/4.0/legalcode.el>, ή δείτε μια "αναγνώσιμη από άνθρωπο" σύνοψη στο <https://creativecommons.org/licenses/by/4.0/deed.el>.

Η εργασία ετοιμάστηκε χρησιμοποιώντας L^AT_EX με το πρότυπο **tufte-latex** και τις βελτιώσεις του **tufte-latex-thesis**.

Το AcubeSAT project εκτελείται με την υποστήριξη του Education Office του Ευρωπαϊκού Οργανισμού Διαστήματος, στα πλαίσια του προγράμματος Fly Your Satellite!

Οι απόψεις που εκφράζονται στο παρόν από τους συγγραφείς δεν μπορούν σε καμία περιπτώση να θεωρηθεί πως εκφραζούν την επίσημη άποψη, ή υποστήριξη, του Ευρωπαϊκού Οργανισμού Διαστήματος.

Έκδόθηκε στις 28 Ιουνίου 2021 (e16θα3d: Add ALL the code)

Περιεχόμενα

1	<i>Εισαγωγή</i>	14
	<i>1.1 Επιστημονική Περιοχή</i>	14
	<i>1.2 Σκοπός και Συνεισφορά της Εργασίας</i>	16
	<i>1.3 Διάρθωση και Δομή</i>	18
2	<i>Η αποστολή AcubeSAT</i>	19
	<i>2.1 Η αποστολή</i>	19
	<i>2.2 Υποσυστήματα</i>	20
	<i>2.2.1 Υποσύστημα Προσδιορισμού και Ελέγχου Προσανατολισμού (ADCS)</i>	21
	<i>2.2.2 Υποσύστημα Τηλεπικοινωνιών (COMMS)</i>	21
	<i>2.2.3 Υποσύστημα Ενέργειας (EPS)</i>	22
	<i>2.2.4 Υποσύστημα Διαχείρισης Δεδομένων (OBDH)</i>	23
	<i>2.2.5 Υποσύστημα Λογισμικού (OBSW)</i>	23
	<i>2.2.6 Υποσύστημα Χειρισμού (OPS)</i>	24
	<i>2.2.7 Υποσύστημα Δομής (Structural)</i>	25
	<i>2.2.8 Μηχανική Συστήματος (SYE)</i>	25
	<i>2.2.9 Επιστημονική Μονάδα (SU)</i>	25
	<i>2.2.10 Υποσύστημα Θερμικού Ελέγχου (Thermal)</i>	27
	<i>2.2.11 Ανάλυση Τροχιάς (Trajectory)</i>	27
	<i>2.3 Χρονιμοποιούμενα εργαλεία</i>	28
3	<i>Η φιλοσοφία FDIR της πρωτοβουλίας SAVOIR</i>	29
	<i>3.1 Το Πρότυπο Αξιοποίησης Πακέτων ECSS</i>	29
	<i>3.2 Το Εγχειρίδιο FDIR του SAVOIR</i>	32
	<i>3.2.1 Βήματα σχεδιασμού FDIR</i>	32
	<i>3.2.2 Προτεινόμενη αρχιτεκτονική</i>	34
4	<i>FDIR στο AcubeSAT</i>	36
	<i>4.1 Βασικές αρχές του FDIR</i>	36
	<i>4.2 Μελέτη διαφορετικών αρχιτεκτονικών</i>	38
	<i>4.2.1 Μόνιμες βλάβες (ανάλυση αξιοπιστίας)</i>	39

4.2.2	<i>Προσωρινές βλάβες (ανάλυση ρυθμού βλαβών)</i>	41
4.2.3	<i>Συμπεράσματα</i>	42
4.3	<i>Λεπτομερείς Ενέργειες του FDIR</i>	44
4.3.1	<i>Τρόποι εντοπισμού αποτυχιών</i>	45
4.3.2	<i>Προληπτικές δράσεις</i>	46
4.3.3	<i>Διορθωτικές δράσεις</i>	47
4.4	<i>Λειτουργική ροή του FDIR</i>	48
4.4.1	<i>Ασφαλής Λειτουργία</i>	49
4.5	<i>Επόμενα βήματα</i>	51
5	<i>Πρακτική επίδειξη του FDIR</i>	52
5.1	<i>Περιγραφή Συστήματος</i>	52
5.1.1	<i>Λειτουργικότητα</i>	52
5.1.2	<i>Υλικό</i>	54
5.2	<i>Λογισμικό</i>	55
5.2.1	<i>Δορυφορικό Τμήμα</i>	55
5.2.2	<i>Μέγεθος Λογισμικού</i>	59
5.2.3	<i>Εντοπισμός Αποτυχιών I²C</i>	60
5.2.4	<i>Έλεγχος Αποστολής</i>	61
5.2.5	<i>Βάση Δεδομένων PUS</i>	61
5.3	<i>Στήσιμο FDIR</i>	62
5.3.1	<i>Λεπτομερής σχεδιασμός FDIR</i>	63
5.4	<i>Επικύρωση FDIR</i>	70
5.4.1	<i>Κανονική λειτουργία</i>	70
5.4.2	<i>Προσομοίωση αστοχιών</i>	73
5.4.3	<i>Τροποποίηση του FDIR σε πραγματικό χρόνο</i>	88
6	<i>Συμπεράσματα και Μελλοντική Εργασία</i>	94
6.1	<i>Μελλοντική εργασία</i>	95
A	<i>Βιβλιογραφία</i>	97
B	<i>Πηγαίος κώδικας</i>	101

Κατάλογος Σχημάτων

1.1	Είδη δορυφόρων σε λειτουργία το 2018 [1]	14
1.2	Ενεργοί δορυφόροι σε τροχιά γύρω από τη γη το 2021	14
1.3	Γραφική αναπαράσταση του δορυφόρου Jason-3	15
1.4	Το "F-1" CubeSat του FPT Πανεπιστημίου στο Βιετνάμ, μεγέθους 1U	15
1.5	Κατάσταση αποστολών CubeSat από το 2000 [11]	16
1.6	Κατάσταση αποστολών CubeSat από ανεξάρτητους κατασκευαστές	16
1.7	Κύριοι λόγοι αποτυχίας αποστολών CubeSat	17
2.1	Λογότυπο της αποστολής AcubeSAT	19
2.2	Καλλιτεχνική προβολή του νανοδορυφόρου AcubeSAT	19
2.3	Προβολή του εσωτερικού του AcubeSAT, όπου φαίνονται τα διαφορετικά υποσυστήματα και η πειραματική διάταξη	20
2.4	Δοκιμαστικό μοντέλο του τσιπ μικροδρευστομηχανικής	20
2.5	Η πλακέτα SatNOGS COMMS	22
2.6	Δυναμική ανάλυση προύπολογισμού ισχύος	23
2.7	Ο μικροελεγκτής SAMV71Q21RT	23
2.8	Ο COTS σκελετός του CubeSat	25
2.9	Παράδειγμα παραγόμενης εικόνας	25
2.10	Διαφανής προβολή του δοχείου πειράματος και του εσωτερικού του	26
2.11	Το τσιπ μικροδρευστομηχανικής	26
2.12	Επίγειο ίχνος μιας πιθανής τροχιάς του AcubeSAT	27
3.2	Προτεινόμενη ιεραρχία των διαφορετικών επιπέδων FDIR από το SAVOIR	34
3.3	Η ροή δεδομένων για τον εντοπισμό της αποτυχίας	35
4.1	Οπτική απεικόνιση των διαφορετικών μορφών σύνδεσης που θα εξεταστούν	39
4.2	Ανθεκτικότητα σύνθετου συστήματος σε μόνιμες βλάβες (θεωρώντας αλάνθαστο ψηφιοφόρο)	40
4.3	Ανθεκτικότητα σύνθετου συστήματος σε μόνιμες βλάβες (θεωρώντας ευάλωτο ψηφιοφόρο)	41
4.4	Ανθεκτικότητα σύνθετου συστήματος σε παροδικά σφάλματα (θεωρώντας αλάνθαστο ψηφιοφόρο)	43
4.5	Γενική ροή του FDIR σε λειτουργικές καταστάσεις του δορυφόρου	48
4.6	Ροή του FDIR στην Ασφαλή Λειτουργία	50
5.1	Μπλοκ διάγραμμα υψηλού επιπέδου της πειραματικής διάταξης	52
5.2	Το πλήρες αναπτυξιακό σύστημα	53
5.3	Επίσημη φωτογραφία της αναπτυξιακής πλακέτας ATSAMV71-XULT	54
5.4	Ο αισθητήρας θερμοκρασίας MCP9808, κολλημένος επάνω στην πλακέτα του πειράματος	54
5.5	Απλοποιημένο σχήματικό της εφαρμογής	55
5.6	Η ροή δεδομένων μεταξύ των διαφορετικών κομματιών του λογισμικού	56

5.7 Καθαρό σήμα I ² C	61
5.8 Σήμα σε ένα ελαττωματικό δίκτυο I ² C	61
5.9 Προβολές παραμέτρων και αρχείου στο YAMCS	62
5.10 Η διεπαφή "PUS database viewer"	62
5.11 Επισκόπηση της διαδικασίας FDIR του αισθητήρα θερμοκρασίας	63
5.12 Περιοδικοί ραδιοφάροι που παράγονται από τον μικροελεγκτή, όπως φαίνονται στο YAMCS	70
5.13 Εξέλιξη μιας παραμέτρου θερμοκρασίας στο χρόνο, όπως φαίνεται στο YAMCS	70
5.14 Έξοδος καταγραφής σε κανονική λειτουργία	71
5.15 Μεταβάσεις που πραγματοποιούνται κατά την εκκίνηση	71
5.16 Λίστα ορισμών παρακολούθησης στην εκκίνηση, χωρίς καμία αποτυχία	72
5.17 Λίστα ορισμών γεγονότος-αντίδρασης	73
5.18 Το γεγονός έναρξης του μικροελεγκτή, μέσα από το YAMCS	73
5.19 Θερμοκρασιακές παραμέτροι YAMCS σε κανονική λειτουργία	75
5.20 Θερμοκρασιακές παραμέτροι YAMCS υπό την αποτυχία F-010	75
5.21 Πίνακας μετάβασης ορισμών παρακολούθησης (αποτυχία F-010 όπως φαίνεται στη διεπαφή PUS)	76
5.22 Πλήρης λίστα των ορισμών παρακολούθησης κατά την αποτυχία F-010	77
5.23 Το γεγονός προσωρινής αποτυχίας του αισθητήρα όπως φαίνεται στο YAMCS	77
5.24 Προβολή της παραμέτρου θερμοκρασίας στη διάρκεια μιας αποτυχίας	78
5.25 Διαγνωστική έξοδος του μικροελεγκτή κατά την αποτυχία F-010	78
5.26 Προβολή της παραμέτρου θερμοκρασίας στη διάρκεια της αποτυχίας F-040	82
5.27 Οι μετρήσεις θερμοκρασίας στις οποίες έχουν προστεθεί εικονικές τιμές	87
5.28 Λίστα τηλεεντολών στο περιβάλλον του YAMCS	89
5.29 Προβολή της οθόνης εκτέλεσης εντολής του YAMCS	90
5.30 Η τηλεεντολή TC[12,5], ακριβώς όπως παρουσιάζεται στο [35]	92
5.31 Περιβάλλον προσθήκης ορισμού παρακολούθησης από το YAMCS	93

Κατάλογος Πινάκων

2.1 Μέγιστες τιμές σφαλμάτων ADCS μετά τη σταθεροποίηση	21
2.2 Προϋπολογισμός ισχύος AcubeSAT σε κανονική λειτουργία	22
4.1 Σύνοψη αποτελεσμάτων μελέτης διαφορετικών αρχιτεκτονικών	43
4.2 Τρόποι εντοπισμού αποτυχιών FDIR του AcubeSAT	45
4.3 Προληπτικές δράσεις FDIR του AcubeSAT	46
4.4 Διορθωτικές δράσεις FDIR του AcubeSAT	47
4.5 Σύνοψη της λειτουργικότητας του AcubeSAT σε διαφορετικές Λειτουργίες Συστήματος	49
5.1 Λίστα διεργασιών του FreeRTOS που υλοποιούν την πειραματική διάταξη	57
5.2 Λίστα του νέου λογισμικού που αναπτύχθηκε για την παρούσα εργασία	57
5.3 Λίστα των έτοιμων βιβλιοθηκών που χρησιμοποιήθηκαν	58
5.4 Μέγεθος του παραχθέντος κώδικα με διαφορετικές ρυθμίσεις μεταγλωττιστή	59
5.5 Τα αρχεία που καταναλώνουν την περισσότερη μνήμη ROM	59
5.6 Τα αρχεία που καταναλώνουν την περισσότερη μνήμη RAM	59
5.7 Μη εγγράψιμοι καταχωρητές του MCP9808	60
5.8 FMEA στο σύστημα επίδειξης	64
5.9 Πίνακας HSIA	67
5.10 Λίστα παραμέτρων ST[20]	68
5.11 Λίστα ορισμών παρακολούθησης της υπηρεσίας ST[12]	68
5.12 Λίστα ορισμών γεγονότων-αντιδράσεων της υπηρεσίας ST[19]	69
5.13 Σύνοψη μεθόδων δοκιμής αποτυχιών	74
5.14 Πίνακας μετάβασης ορισμών παρακολούθησης (αποτυχία F-010)	76
5.15 Πίνακας μετάβασης ορισμών παρακολούθησης (αποτυχία F-020)	79
5.16 Πίνακας μετάβασης ορισμών παρακολούθησης (αποτυχία F-040)	81
5.17 Πίνακας μετάβασης ορισμών παρακολούθησης (αποτυχία F-050)	83
5.18 Πίνακας μετάβασης ορισμών παρακολούθησης (προσωρινή αποτυχία F-040, διαφορική)	84
5.19 Πίνακας μετάβασης ορισμών παρακολούθησης (μόνιμη αποτυχία F-050, διαφορική)	85
5.20 Πίνακας μετάβασης ορισμών παρακολούθησης (αποτυχία F-130)	86
5.21 Απαιτήσεις αναδιαμόρφωσης FDIR στην πειραματική διάταξη	88
5.22 Πίνακας μετάβασης με απενεργοποιημένη την υπηρεσία event-action	91

Μεταφράσεις ξενόγλωσσης ορολογίας

beacon ραδιοφάρος

bus δίαυλος

check validity condition συνθήκη εγκυρότητας ελέγχου

cold redundancy ψυχρός πλεονασμός (passive redundancy)

detection εντοπισμός

driver οδηγός περιφερειακού

enumeration απαρίθμηση

event γεγονός

event-action γεγονός-αντίδραση

failure αποτυχία (εναλλακτικές μεταφράσεις: βλάβη, αστοχία)

failure mode τρόπος αποτυχίας

float τιμή κινητής υποδιαστολής

hot redundancy ενεργός πλεονασμός (active redundancy)

interface διεπαφή

isolation απομόνωση

microcontroller μικροελεγκτής

monitoring definition ορισμός παρακολούθησης

operator χειριστής

parameter παράμετρος

recovery διόρθωση

reliability αξιοπιστία

requirement προδιαγραφή

server διακομιστής

service υπηρεσία

standard πρότυπο

validation επικύρωση

verification επαλήθευση

voter ψηφοφόρος

warm redundancy θερμός πλεονασμός

Ακρωνύμια

<i>ADCS</i> Attitude Determination and Control Subsystem	20–23, 37, 50
<i>API</i> Application Programming Interface (Διεπαφή Προγραμματισμού Εφαρμογών)	61
<i>CAN</i> Controller Area Network	23
<i>CCSDS</i> The Consultative Committee for Space Data Systems.....	22
<i>CDR</i> Critical Design Review	20
<i>COBS</i> Consistent Overhead Byte Stuffing	54
<i>COMMS</i> Communications	21, 22, 50
<i>COTS</i> Commercial Off-The-Shelf.....	15, 20, 22, 25, 43
<i>DDJF</i> Design Definition and Justification File	
<i>ECSS</i> European Cooperation for Space Standardization .	16, 18, 29, 35, 51, 52, 54, 61, 70, 75, 88, 95, 96, 109
<i>EMC</i> Electromagnetic Compatibility (Ηλεκτρομαγνητική Συμβατότητα)	22
<i>EPS</i> Electrical Power Subsystem	22, 50
<i>ETL</i> Embedded Template Library	56, 59
<i>FDIR</i> Fault Detection, Isolation and Recovery (Εντοπισμός, Απομόνωση και Διόρθωση Αποτυχιών) 15–18, 20, 22, 26, 29–38, 48–53, 55, 61–63, 65, 68, 70, 75, 76, 79–81, 83, 85, 86, 88–91, 94–96, 113	
<i>FMEA</i> Failure Mode and Effects Analysis	16, 25, 33, 44, 51, 63, 65, 94
<i>FMECA</i> Failure Mode, Effects and Criticality Analysis	
<i>GMAT</i> General Mission Analysis Tool	
<i>GNSS</i> Global Navigation Satellite System (Παγκόσμιο Δορυφορικό Σύστημα Πλοϊγησης).....	14
<i>GS</i> Ground Station (Σταθμός Βάσης)	22
<i>HAL</i> Hardware Abstraction Library.....	56
<i>HSIA</i> Hardware/Software Interaction Analysis	33, 51, 63, 65, 66
<i>HTTP</i> Hypertext Transfer Protocol	61
<i>I²C</i> Inter-Integrated Circuit	54, 60, 64, 75, 85, 111
<i>IC</i> Integrated Circuit (Ολοκληρωμένο Κύκλωμα)	
<i>IDE</i> Integrated Development Environment	56
<i>ISM</i> Industrial, Scientific, Medical	22
<i>LEO</i> Low Earth Orbit (Χαμηλή Γύινη Τροχιά).....	15, 25

<i>MAIV</i> Maintenance, Assembly, Integration and Verification (Συντήρηση, Κατασκευή, Συναρμολόγηση και Επαλήθευση)	25
<i>MCU</i> microcontroller (Μικροελεγκτής, MicroController Unit)	54
<i>MPPT</i> Maximum Power Point Tracking	23
<i>MRAM</i> Magnetoresistive Random-Access Memory	23
<i>OBC</i> On-Board Computer	22, 23, 37, 50
<i>OBDH</i> On-Board Data Handling	23
<i>OBSW</i> On-Board Software	23
<i>OPS</i> Operations	23
<i>PA</i> Product Assurance	20
<i>PCB</i> Printed Circuit Board (Τυπωμένη Πλακέτα)	
<i>PCDU</i> Power Conditioning & Distribution Unit	23
<i>PDMS</i> Polydimethylsiloxane (Πολυδιμεθυλοσιλοξάνη)	26
<i>PMON</i> Parameter Monitoring Definition (Ορισμός Παρακολούθησης Παραμέτρου)	
<i>PUS</i> Packet Utilisation Standard	29, 34, 49, 51, 55, 61, 65, 66, 68, 88, 89, 130
<i>RAM</i> Random Access Memory (Μνήμη Τυχαίας Προσπέλασης)	59
<i>RAMS</i> Reliability, Availability, Maintainability and Safety (Αξιοπιστία, Διαθεσιμότητα, Συντηρησιμότητα και Ασφάλεια)	25, 33
<i>RF</i> RadioFrequency	22, 30
<i>ROM</i> Read-Only Memory (Μνήμη Μόνο Ανάγνωσης)	
<i>RTOS</i> Real-Time Operating System (Λειτουργικό Σύστημα Πραγματικού Χρόνου)	23, 49, 55, 80
<i>SAVOIR</i> Space AVionics Open Interface aRchitecture	16, 29, 31, 37, 52, 88, 94
<i>SCL</i> Serial Clock	60, 75, 79, 85
<i>SDA</i> Serial Data	60, 75, 79, 85
<i>SEFI</i> Single Event Functional Interrupt (Διακοπή Λειτουργίας Απλής Προσβολής)	
<i>SEL</i> Single Event Latchup (Μανδάλωση Απλής Προσβολής)	65
<i>SRAM</i> Static Random Access Memory	54
<i>SU</i> Science Unit	22, 25
<i>SYE</i> Systems Engineering	25
<i>TC</i> Telecommands (Τηλεεντολές)	21, 29–31, 54, 86
<i>TM</i> Telemetry (Τηλεμετρία)	21, 29, 31, 35, 54
<i>UART</i> Universal Asynchronous Serial Bus	54, 138
<i>UHF</i> Ultra-High Frequency	22, 50
<i>USB</i> Universal Serial Bus	54
<i>XTCE</i> XML Telemetric and Command Exchange	61
<i>YAMCS</i> Yet Another Mission Control System	61, 70, 79, 88, 91, 130, 138

Περίληψη

Το διάστημα δεν είναι ένα φιλόξενο περιβάλλον: αν και η σημερινή τεχνολογία έχει επιτρέψει τη λειτουργία χιλιάδων τεχνητών δορυφόρων σε τροχιά γύρω από τη γη, οι νανοδορυφόροι που χαρακτηρίζονται ως “CubeSats” έχουν ένα ποσοστό αποτυχίας κοντά στο 50%. Τα χαμπλά κόστη, η έλλειψη αυστηρών τεχνικών προδιαγραφών και η έλλειψη ανοιχτής τεχνικής βιβλιογραφίας συχνά αυξάνει τα ρίσκα για τα εκπαιδευτικά, επιστημονικά και εμπορικά CubeSat. Αυτή η εργασία ερευνά μία απλή παραμετροποίηση αρχιτεκτονική Ανίχνευσης, Απομόνωσης και Αντιμετώπισης Βλαβών (“FDIR”) που βασίζεται σε Ευρωπαϊκά πρότυπα και πρωτόκολλα, ευρέως χρησιμοποιούμενα στον τομέα της αεροδιαστηματικής. Αυτή η δομή διόρθωσης σφαλμάτων, μαζί με τη συνοδεύομενη υλοποίηση σε κώδικα, μπορεί να χρησιμοποιηθεί από οποιαδήποτε αποστολή CubeSat για να αυξηθεί η αξιοπιστία του σχεδιασμού και να μειωθεί η πιθανότητα αποτυχίας, μέσω της αυτόνομης αντιμετώπισης βλαβών κατά την πτήση. Η εργασία επίσης περιλαμβάνει γενικές πληροφορίες σχετικά με την αξιοπιστία των CubeSat, και εξερευνά το λογισμικό και υλικό που χρησιμοποιείται για να υλοποιηθεί το προτεινόμενο FDIR στην αποστολή AcubeSAT, η οποία βρίσκεται υπό σχεδιασμό από φοιτητές του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης.

Abstract

Konstantinos Kanavouras

Design of Fault Detection, Isolation and Recovery in the AcubeSAT NanoSatellite

Supervisor: Prof. Alkiviadis Hatzopoulos

Aristotle University of Thessaloniki

28 June 2021

Space is not a welcoming environment; while the aerospace engineering community has managed to reliably operate thousands of satellites in orbit, CubeSats, the most popular class of nanosatellite, only have a 50% success rate. Low costs, lack of strict technical requirements and scarcity of publicly available documentation often drives up the risks for educational, scientific and commercial CubeSats. This thesis investigates a configurable and modular Fault Detection, Isolation and Recovery (FDIR) architecture that uses the ECSS Packet Utilisation Standard. This FDIR concept, along with the provided open-source software implementation, can be used by CubeSat missions to increase the reliability of their design and chances of mission success, by autonomously responding to on-board errors. The thesis also includes background information regarding CubeSat reliability, and explores the software and hardware used to implement the proposed FDIR design on the AcubeSAT mission, currently under design by students of the Aristotle University of Thessaloniki.

Eυχαριστίες

Κατ' αρχάς, θα ήθελα να ευχαριστήσω τον Καθηγητή Αλκιβιάδη Χατζόπουλο, όχι μόνο για τη βοήθεια και τις ευκαιρίες που μου έχει προσφέρει, αλλά για τη διαρκή υποστήριξη που παρέχει στο πρότζεκτ του CubeSat, σε οργανωτικό και τεχνικό επίπεδο. Επίσης, θα ήθελα να ευχαριστήσω τους δεκάδες συμφοιτητές μου στο πρότζεκτ, και ιδιαίτερα στο υποσύστημα OBC, για τη συνεργασία και το έργο που έχουν προσφέρει και χρησιμοποιείται σε αυτήν τη διπλωματική. Η εργασία αυτή είναι αποτέλεσμα ενός φιλόδοξου πρότζεκτ που ξεκίνησε από το 2016 και έχει απασχολήσει έναν τεράστιο αριθμό φοιτητών, επιστημόνων και μηχανικών που πίστεψαν σε αυτό, από την Ελλάδα αλλά ακόμα και από την υπόλοιπη Ευρώπη, παρά όλες τις δυσκολίες και εμπόδια που αντιμετωπίστηκαν. Τέλος, θα ήθελα να ευχαριστήσω τον Αναστάση Ρετσέλη για τη συνεργασία μας και τις ενδιαφέρουσες συζητήσεις στο κομμάτι του Systems Engineering, αλλά και όλους τους φίλους και τους γονείς μου που στάθηκαν κοντά μου όλα τα τελευταία χρόνια. Δεν θα βρισκόμουν εδώ χωρίς εσάς.

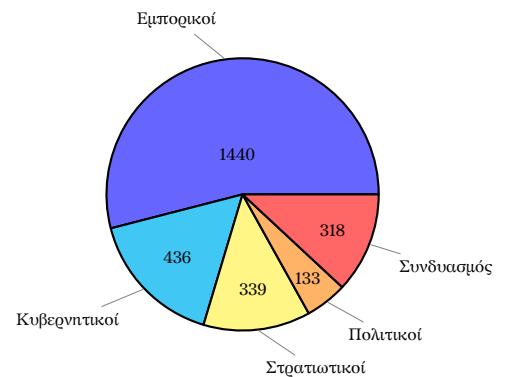
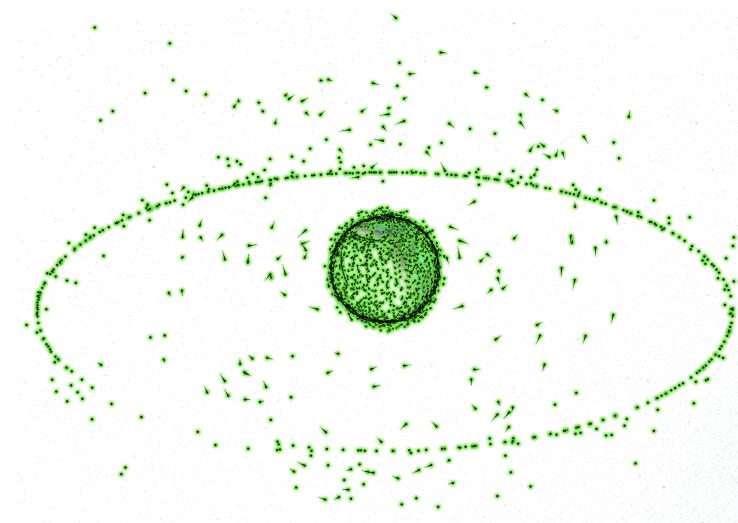
1

Εισαγωγή

1.1 Επιστημονική Περιοχή

Η παρούσα εργασία ασχολείται με την Αεροδιαστημική Μηχανική, και πιο συγκεκριμένα την τεχνολογία κατασκευής και χειρισμού δορυφόρων. Ο πρώτος τεχνητός δορυφόρος "Sputnik 1" εκτοξεύτηκε το έτος 1957, και μέχρι το 2021 έχουν εκτοξευτεί περίπου 8900 δορυφόροι, από τους οποίους περίπου 4000 είναι λειτουργικοί και σε τροχιά [1–3].

Οι εκτοξευμένοι δορυφόροι επιτελούν ένα σύνολο από διαφορετικές λειτουργίες, που μπορεί να περιλαμβάνουν τηλεπικοινωνίες, παρατήρηση γης, σήματα πλοήγησης (GNSS), επιστημονική έρευνα, ανάπτυξη τεχνολογίας ή άλλους σκοπούς, χωρίς να συμπεριλαμβάνονται σε αυτό το κομμάτι επανδρωμένες αποστολές, αποστολές εκτός της τροχιάς της γης, ή διαστημικοί σταθμοί. Τα δεδομένα που παράγει η διαστημική τεχνολογία, με επιδράσεις στις μεταφορές, στην πρόγνωση καιρού και στην τεχνολογία, έχουν εδώ και καιρό γίνει αναπόσπαστο κομμάτι του σύγχρονου πολιτισμού.



Σχήμα 1.1: Είδη δορυφόρων σε λειτουργία το 2018 [1]

[1] Wood, *Visualizing All of Earth's Satellites*.

[2] Union of Concerned Scientists, *Satellite Database*.

[3] Kelso, *NORAD Two-Line Element Sets Current Data*.

Σχήμα 1.2: Ενεργοί τεχνητοί δορυφόροι σε τροχιά γύρω από τη γη το 2021. Κοντά στον πλανήτη βρίσκονται οι δορυφόροι Χαμπλής Γήινης Τροχιάς. Σε σταθερό υψόμετρο 35 786 km βρίσκονται οι δορυφόροι Γεωστατικής τροχιάς, και ανάμεσά τους κινούνται οι δορυφόροι Μέσης Γήινης Τροχιάς [4]

Μετά την εκτόξευσή τους, οι δορυφόροι τυπικά παραμένουν στο διάστημα μέχρι την ολοκλήρωση της αποστολής τους μετά από κάποια χρόνια, οπότε και μπορούν να επιστρέψουν στη γη και να καούν στην ατμόσφαιρα, είτε να παραμείνουν στο διάστημα ως διαστημικά σκουπίδια.

Η ανάπτυξη διαστημικής τεχνολογίας παρουσιάζει έναν σημαντικό βαθμό δυσκολίας, ο οποίος προέρχεται από τις ιδιαίτερα αυξημένες απαιτήσεις αξιοπιστίας στις δυσμενείς συνθήκες του διαστήματος. Πιο συγκεκριμένα, η μεγάλη απόσταση, οι ακραίες θερμοκρασίες, τα μηχανικά φορτία της εκτόξευσης, οι ακτινοβολίες υψηλών ενεργειών, το κενό αέρα, και η αδυναμία αποκατάστασης βλαβών, έχουν εγείρει ένα ευρύ φάσμα προκλήσεων μπροστά στην εξερεύνηση του διαστήματος. Έτσι, πέρα από την τυπική ανάπτυξη τεχνολογίας, τα πρότζεκτ αεροδιαστηματικής συνδέονται αναπόσπαστα με ένα σύνολο κλάδων της μηχανικής, όπως το systems engineering (μηχανική συστημάτων), product assurance (διασφάλιση ποιότητας), reliability engineering (αξιοπιστία συστημάτων) και ζητήματα assembly, integration & verification (κατασκευής, συναρμολόγησης και επαλήθευσης) [5]. Η παρούσα εργασία εστιάζει στο κομμάτι Fault Detection, Isolation and Recovery (Εντοπισμός, Απομόνωση και Διόρθωση Αποτυχιών, FDIR), δηλαδή στη δυνατότητα των διαστημικών συστημάτων να εντοπίζουν και να ανακάμπτουν αυτόματα από σφάλματα και λειτουργικές αστοχίες.

Παρά τις προαναφερθείσες δυσκολίες, η επιστήμη έχει προχωρήσει αρκετά ώστε το διάστημα να αποτελεί έναν εμπορικό στόχο. Στα πλαίσια της φιλοσοφίας "NewSpace", όλο και περισσότεροι κερδοσκοπικοί ή μη φορείς εκμεταλλεύονται τις δυνατότητες που προσφέρονται έξω από τη γη [6]. Start-ups, αστερισμοί δορυφόρων, εκπαιδευτικές αποστολές και εμπορικές υπηρεσίες είναι έννοιες που ήδη έγιναν συνηθισμένες τη δεκαετία του 2010, με περισσότερες χώρες ανά τον κόσμο να αρχίζουν όχι μόνον να έχουν πρόσβαση στο διάστημα, αλλά και να αποκτούν δυνατότητα εκτόξευσης.

Στα πλαίσια του παραπάνω μελετάται η τεχνολογία των όλο και πιο δημοφιλών **νανοδορυφόρων** [7]. Τα CubeSats, η πιο δημοφιλής κατηγορία νανοδορυφόρου που είναι και το αντικείμενο αυτής της εργασίας, δημιουργήθηκαν σαν ιδέα το 1999. Το χαμπλό τους κόστος και μέγεθος, καθώς και η μικρή σχετικά πολυπλοκότητα, έχουν επιτρέψει πανεπιστήμια και οργανισμούς παγκοσμίως να εκτοξεύουν πάνω από 1530 CubeSats [8].

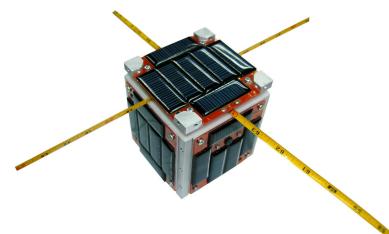
Τα CubeSats κατασκευάζονται από δομικές μονάδες διαστάσεων $10 \times 10 \times 10$ cm που ονομάζονται **Units**. Τυπικά μεγέθη δορυφόρων προκύπτουν "στοιβάζοντας" αυτά τα units, με αποτέλεσμα δορυφόρους 1U, 1.5U, 2U, 3U, 6U και 12U. Το κάθε Unit να μπορεί να ξυγίσει μέρχι 2 kg. Εσωτερικά, τα CubeSat αποτελούνται από εμπο-



Σχήμα 1.3: Γραφική αναπαράσταση του δορυφόρου Jason-3.

[5] Wertz and Larson, *Space Mission Analysis and Design*.

[6] Denis et al., "From New Space to Big Space".



Σχήμα 1.4: Το "F-1" CubeSat του FPT Πανεπιστημίου στο Βιετνάμ, μεγέθους 1U

[7] Sweeting, "Modern Small Satellites-Changing the Economics of Space".

[8] Swartwout, *CubeSat Database*.

ρικά πλεκτρονικά εξαρτήματα χαμηλού κόστους, τα λεγόμενα Commercial Off-The-Shelf (COTS). Τα CubeSats εκτοξεύονται συνήθως σε Χαμηλή Γύινη Τροχιά (LEO) [9, 10], τοποθετούμενα ως δευτερεύοντα φορτία σε εκτοξεύσεις μεγαλύτερων δορυφόρων ("piggyback launch"), και παραμένουν ενεργά για χρονικά διαστήματα περίπου 1 ως 3 χρόνια.

1.2 Σκοπός και Συνεισφορά της Εργασίας

Η παρούσα εργασία αφορά τη βελτίωση της **αξιοπιστίας** των νανοδορυφόρων και συστημάτων CubeSat. Αν και η πλειοψηφία των κλασικών δορυφόρων που έχουν εκτοξευτεί είχαν, έστω μερικώς, επιτυχημένες αποστολές [12, 13], το 39.6% των εκτοξευμένων CubeSats έχουν αποτύχει να ολοκληρώσουν την αποστολή τους, ή δεν έχουν λειτουργήσει σε τροχιά. Το ποσοστό αποτυχίας αυξάνεται στο 57.0% αν εστιάσουμε σε φορείς (όπως Πανεπιστήμια) που εκτοξεύουν τα πρώτα τους CubeSat, και δεν διαθέτουν το διαστημικό υπόβαθρο των μεγάλων κατασκευαστών και των εμπορικών δορυφόρων.

Οι λόγοι της υψηλής αποτυχίας των CubeSat σχετίζονται κυρίως με το συμπεισμένο πρόγραμμα κατασκευής, το οποίο οδηγεί σε σφάλματα και προβλήματα που δεν εντοπίζονται κατά τη διάρκεια του σχεδιασμού. Πηγές βλαβών όπως περιβαλλοντική φθορά (θερμικοί κύκλοι, ακτινοβολία), έλλειψη περιθωρίων ασφαλείας στην ισχύ και τις τηλεπικοινωνίες, και προβλήματα λογισμικού, συνδυάζονται με τη γενικότερη έλλειψη εμπειρίας και αυστηρών διαδικασιών των νέων κατασκευαστών, για να μειώσουν το βαθμό αξιοπιστίας των νανοδορυφόρων [11, 14].

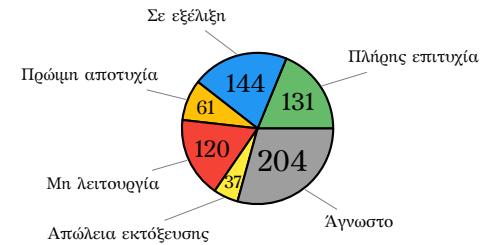
Η προσέγγιση που ακολουθείται για την αύξηση της αξιοπιστίας των CubeSat περιλαμβάνει:

- Μείωση του αριθμού πιθανών βλαβών από την φάση του σχεδιασμού, με συμπερίληψη ζητημάτων αξιοπιστίας στις πρώιμες φάσεις του πρότζεκτ
- Δομημένα εργαλεία εκτίμησης και βελτίωσης της αξιοπιστίας, όπως **Failure Mode and Effects Analysis (FMEA)** και **Fault Detection, Isolation and Recovery (Εντοπισμός, Απομόνωση και Διόρθωση Αποτυχιών, FDIR)** [15, 16]
- Προσαρμογή και υιοθέτηση ανοιχτών προτύπων διαστημικής τεχνολογίας για το σχεδιασμό, την κατασκευή και την επαλήθευση [17]

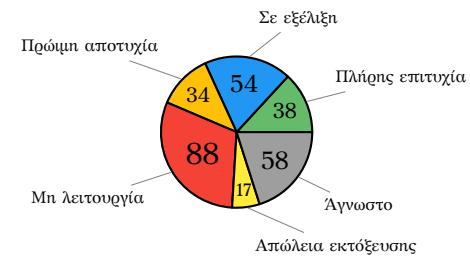
Η παρούσα εργασία σκοπεύει να εμπλουτίσει την εργαλειοθήκη των κατασκευαστών CubeSat αναλύοντας μία δομημένη προσέγγιση FDIR που βασίζεται στα **ευρωπαϊκά πρότυπα ECSS**, και στην

[9] Anthopoulos and Retselis, "Orbital Analysis and the Spin-Orbit Coupling for the AcubeSAT Mission".

[10] Riebeek, *Catalog of Earth Satellite Orbits*.



Σχήμα 1.5: Κατάσταση αποστολών CubeSat από το 2000 [11]



Σχήμα 1.6: Κατάσταση αποστολών CubeSat από "hobbyists" (πανεπιστήμια και άλλα ερευνητικά/εκπαιδευτικά πρότζεκτ) [8]

[12] Kattakuri, "Failures in Spacecraft System".

[13] Jacklin, *Small-Satellite Mission Failure Rates*.

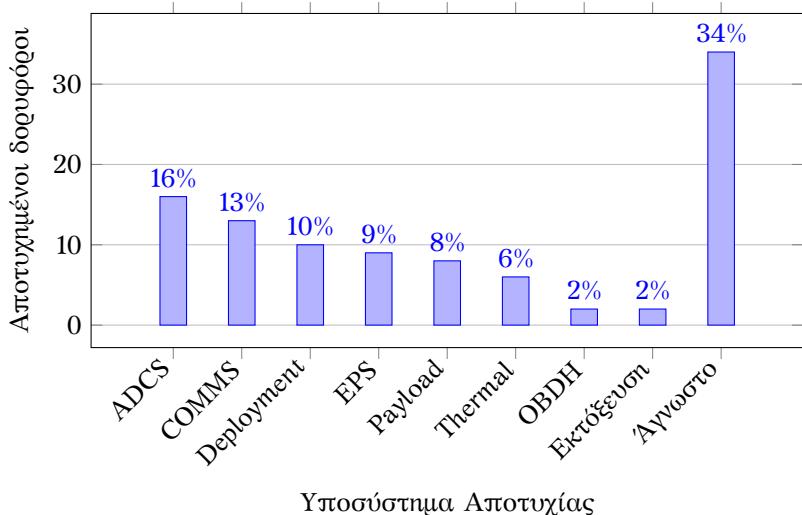
[11] Swartwout, "CubeSat Mission Success: Are We Getting Better?"

[14] Langer *et al.*, "A Reliability Estimation Tool for Reducing Infant Mortality in Cubesat Missions".

[15] Faure *et al.*, "Toward Lean Satellites Reliability Improvement Using HORYU-IV Project as Case Study".

[16] Menchinelli *et al.*, "A Reliability Engineering Approach for Managing Risks in CubeSats".

[17] Scholz, *Handbook of Free and Open Space Standards*.



Σχήμα 1.7: Κύριοι λόγοι αποτυχίας αποστολών CubeSat ($n = 50$) [18]

εργασία της πρωτοβουλίας **SAVOIR**¹ της ευρωπαϊκής διαστημικής βιομηχανίας. Στις επόμενες σελίδες, θα αναλυθεί πώς οι ιδέες αυτές μπορούν να προσαρμοστούν σε μία αποστολή CubeSat μικρού βελτιστού ωστε:

- Να χρησιμοποιηθεί ένας δομημένος τρόπος για να ερευνηθούν οι πιθανές αποτυχίες ενός δορυφόρου, και τρόποι αντιμετώπισής τους
- Να προετοιμαστεί λογισμικό που θα μπορεί να ανταποκριθεί και να διαχειριστεί με ασφάλεια το σύνολο των κρίσιμων αποτυχιών σε τροχιά
- Να αναπτυχθεί μία **παραμετροποιήσιμη** αρχιτεκτονική, που θα επιτρέπει τους χειριστές να μεταβάλλουν εύκολα τη συμπεριφορά του συστήματος FDIR και να το προσαρμόσουν σε διαφορετικά δεδομένα

Το προτεινόμενο σύστημα FDIR στοχεύει στην αύξηση της σιγουριάς κατά την κατασκευή, αλλά και στη γρήγορη ενσωμάτωση νέων δεδομένων που προκύπτουν κατά τις δοκιμές και την εκτόξευση.

Πέρα από τη θεωρητική ανάλυση, περίληψη και σχολιασμό των προτύπων, η εργασία εστιάζει στην εφαρμογή της προαναφερθείσας δομής FDIR στο νανοδορυφόρο *AcubeSAT*, και περιλαμβάνει μία πειραματική υλοποίηση της συνίτας FDIR σε υλικό & λογισμικό.

¹ <https://savoir.estec.esa.int/>

1.3 Διάρθωση και Δομή

Η εργασία ξεκινάει με την περίληψη της αποστολής AcubeSAT, που αποτελεί την κύρια αφορμή και πλαισίο συγγραφής της (Κεφάλαιο 2). Συνεχίζει με μία ανάλυση των κύριων προτύπων και της φιλοσοφίας FDIR, με εστίαση στα κομμάτια που αφορούν τους κατασκευαστές CubeSat και τις ερευνητικές ή εκπαιδευτικές αποστολές (Κεφάλαιο 3). Το επόμενο κομμάτι εξετάζει την εφαρμογή των προηγούμενων προτύπων στο σχεδιασμό του AcubeSAT, και σε μερικές σχεδιαστικές αποφάσεις και λεπτομέρειες υλοποίησης του FDIR στο νανοδορυφόρο (Κεφάλαιο 4).

Το τελευταίο κεφάλαιο περιλαμβάνει την πρακτική επίδειξη του προτεινόμενου FDIR σε ένα απλό δοκιμαστικό σύστημα δύο αισθητήρων (Κεφάλαιο 5). Το σύστημα αναπτύσσεται έτσι ώστε να μπορεί να εντοπίσει και να ανακάψει από όλες τις προβλεπόμενες πιθανές αποτυχίες, χρησιμοποιώντας τις πλήρεις βιβλιοθήκες που καλύπτουν τα πρότυπα ECSS και FDIR. Μετά από την ανάπτυξη του συστήματος, γίνεται πλήρης δοκιμή της απόκρισής του στις αποτυχίες. Το σύστημα είναι διαμορφωμένο έτσι ώστε να εξομοιώσει το σύστημα του δορυφόρου, συμπεριλαμβανομένου του συστήματος μικροελεγκτή αλλά και του Σταθμού Βάσης.

2

Η αποστολή AcubeSAT

Το πλαίσιο ανάπτυξης της παρούσας εργασίας είναι το πρότζεκτ του νανοδορυφόρου "AcubeSAT" που βρίσκεται υπό σχεδιασμό από φοιτητές του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης, και στοχεύει στην εκτέλεση ενός πειράματος βιολογικής φύσεως στο διάστημα.

Το πρότζεκτ ξεκίνησε σε ερευνητική μορφή το 2015, και το 2020 έγινε μέρος του προγράμματος *Fly Your Satellite! 3¹* του Ευρωπαϊκού Οργανισμού Διαστήματος (European Space Agency), με ορίζοντα εκτόξευσης το 2023.

Το AcubeSAT έχει μέγεθος 3 κυβικών μονάδων (3U), διαστάσεις $10 \times 10 \times 34.05$ cm και μάζα 4.26 kg. Περιλαμβάνει το ωφέλιμο φορτίο του πειράματος, που καλύπτει 2U, και τα υπόλοιπα υποστηρικτικά υποσυστήματα (platform) στο υπόλοιπο 1U.

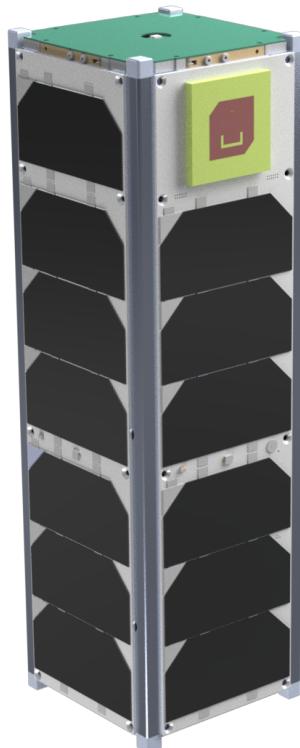
Το πρότζεκτ είναι ανοιχτού κώδικα, με το λογισμικό και τις κατασκευές που έχει ετοιμάσει οι ομάδα να βρίσκονται στο διαδίκτυο για ελεύθερη χρήση από το κοινό.²

2.1 Η αποστολή

Η κύρια αποστολή του AcubeSAT είναι η μελέτη ευκαρυωτικών κυττάρων σε Χαμηλή Γύινη Τροχιά, και πιο συγκεκριμένα η εκτίμηση των αποτελεσμάτων της *ακτινοβολίας* και *μικροβαρύτητας* σε 190 διαφορετικά στελέχη του ζυμομύκητα *Saccharomyces Cerevisiae*. Τα προηγούμενα επιτυγχάνονται με μία πρωτότυπη διάταξη "lab-on-a-chip", η οποία στο κέντρο της έχει ένα τσιπ **μικροβευστομηχανικής** ([Σχήμα 2.4](#)) [19] που επιτρέπει την εξέταση της ρύθμισης της μοριακής συμπεριφοράς των κυττάρων σε πρωτοφανή κλίμακα.



Σχήμα 2.1: Λογότυπο της αποστολής AcubeSAT



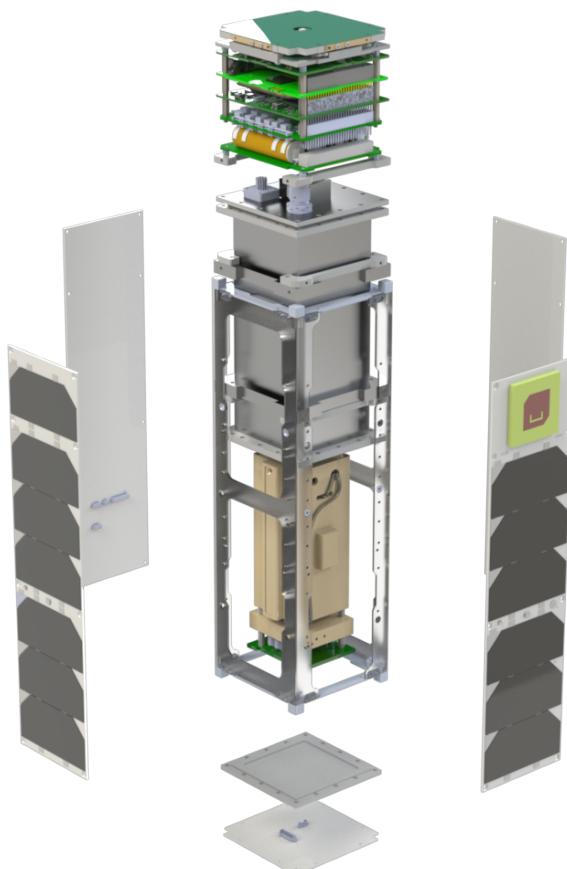
Σχήμα 2.2: Καλλιτεχνική προβολή του νανοδορυφόρου AcubeSAT

¹ https://www.esa.int/Education/CubeSats_-_Fly_Your_Satellite

² <https://gitlab.com/acubesat/>

[19] Volpetti *et al.*, "A Microfluidic Bioprinter".

Τα αποτελέσματα του πειράματος μεταφέρονται στη Γη σε μορφή φωτογραφιών, όπου συγκρίνονται με ένα όμοιο επίγειο πείραμα.

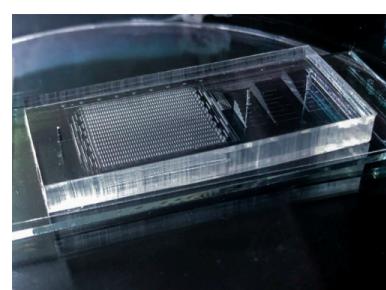


Σχήμα 2.3: Προβολή του εσωτερικού του AcubeSAT, όπου φαίνονται τα διαφορετικά υποσυστήματα και η πειραματική διάταξη

2.2 Υποσυστήματα

Ο νανοδορυφόρος AcubeSAT είναι τεχνικά και οργανωτικά χωρισμένος σε 11 διαφορετικές υποομάδες ή υποσυστήματα, κάθε μία από τις οποίες είναι υπεύθυνη για ένα διαφορετικό τμήμα του δορυφόρου, και αποτελείται από 2 ως 9 αφοσιωμένα μέλη.

Στις επόμενες ενότητες παρουσιάζεται μια σύντομη εισαγωγή στη λειτουργία και το σχεδιασμό κάθε υποσυστήματος. Καθώς η διαδικασία σχεδιασμού συστημάτων και Product Assurance είναι εγγενώς συνδεδεμένη με τη λειτουργία όλων των υποσυστημάτων, αναφέρονται επίσης οι λεπτομέρειες που αφορούν το FDIR. Για πιο λεπτομερείς πληροφορίες, ο αναγνώστης ενθαρρύνεται να ανατρέξει στην ιστοσελίδα του AcubeSAT³, ή στα δημοσίως διαθέσιμα έγγραφα του Critical Design Review (CDR)⁴.



Σχήμα 2.4: Δοκιμαστικό μοντέλο του τσιπ μικρορρευστομηχανικής

³ <https://acubesat.spacedot.gr/subsystems/>

⁴ <https://gitlab.com/acubesat/documentation/cdr-public>

2.2.1 Υποσύστημα Προσδιορισμού και Ελέγχου Προσανατολισμού (ADCS)

Η υποομάδα του ADCS είναι υπεύθυνη για τον έλεγχο του προσανατολισμού του δορυφόρου σε τροχιά. Αυτό επιτυγχάνεται με τη χρήση μιας σειράς ενεργοποιητών, που προέρχονται έτοιμοι από την αγορά (COTS): μία πλακέτα 3 αξόνων και έναν τροχός αντίδρασης 1 άξονα, σε συνδυασμό με αισθητήρες για τον προσδιορισμό της περιστροφής του δορυφόρου (1 γυροσκόπιο και 2 μαγνητόμετρα) και αλγορίθμους φιλτραρίσματος & ελέγχου [20, 21].

Το ADCS μπορεί να λειτουργήσει με τα εξής διαφορετικά “προφίλ στόχευσης” (pointing profiles) για να καλύψει διαφορετικές λειτουργικές ανάγκες:

- 1. Λειτουργία Σταθεροποίησης** (Detumbling), όπου ο δορυφόρος προσπαθεί να φτάσει τη γωνιακή του ταχύτητα κοντά στο 0, για να εξασφαλίσει μια αξιόπιστη θαδιοζεύξη, να αποτρέψει την αποκόλληση εξαρτημάτων και να επιτρέψει αργότερα την ευκολότερη ανάκτηση του επιθυμητού προσανατολισμού.

Η λειτουργία Σταθεροποίησης υλοποιείται με τον απλούστερο δυνατό τρόπο, χρησιμοποιώντας μόνο το ένα από τα 2 μαγνητόμετρα και έναν απλό αλγόριθμο ελέγχου. Ενεργοποιείται όταν δεν υπάρχει ανάγκη εφαρμογής κάποιου άλλου ή όταν η γωνιακή ταχύτητα του νανοδορυφόρου είναι επικίνδυνα υψηλή.

- 2. Στόχευση στο Ναδίρ** (Nadir Pointing), όπου ο δορυφόρος στρέφει την πλευρά +X ώστε να κοιτάζει προς τη Γη. Αυτό το προφίλ χρησιμοποιείται σε περάσματα πάνω από τον Σταθμό Βάσης, όπου η κατευθυντική κεραία χρειάζεται άμεση ορατότητα.
- 3. Στόχευση στον Ήλιο** (Sun Pointing), όπου ο δορυφόρος στρέφεται προς τον ήλιο, προκειμένου να μεγιστοποιηθεί η είσοδος των ηλιακών πάνελ. Αυτό το προφίλ χρησιμοποιείται ανάμεσα στα περάσματα από το Σταθμό Βάσης, προκειμένου να εξασφαλιστεί θετικός προϋπολογισμός ισχύος.

2.2.2 Υποσύστημα Τηλεπικοινωνιών (COMMS)

Το υποσύστημα communications (τηλεπικοινωνιών) είναι υπεύθυνο για τη μετάδοση δεδομένων μεταξύ της Γης και του δορυφόρου σε τροχιά. Τα μεταδιδόμενα δεδομένα χωρίζονται σε 3 διαφορετικές κατηγορίες [22]:

- Telecommands (Τηλεεντολές, TC):** Εντολές από τη Γη προς το δορυφόρο. Μπορούν να χρησιμοποιηθούν για να ζητήσουν πληροφορίες ή για να εκτελέσουν συγκεκριμένες ενέργειες στο CubeSat.

[20] Savvidis *et al.*, *AcubeSAT AOCS DDIF*.

[21] Velentzas and Kotsakiachidis, “Design of the Attitude Control System of the AcubeSAT Nanosatellite”.

Πίνακας 2.1: Μέγιστες τιμές σφαλμάτων ADCS μετά τη σταθεροποίηση

Σφάλμα	Τιμή
Απόλυτη Ακρίβεια Στόχευσης	< 30°
Απόλυτη Ακρίβεια Γνώσης	< 1°

[22] Kapoglou and Chatziargyriou, *AcubeSAT TTC DDIF*.

- Telemetry (Τηλεμετρία, TM):** Πληροφορίες που αποστέλλονται από τον δορυφόρο προς τη Γη, που συνήθως περιλαμβάνουν χωτικές πληροφορίες, όπως τιμές αισθητήρων, κατάσταση συστήματος, τρέχουσα ώρα και συμβάντα.
- Επιστημονικά δεδομένα:** Τα επιστημονικά δεδομένα που παράγονται από το ωφέλιμο φορτίο (payload). Αυτά είναι τα δεδομένα με τον μεγαλύτερο όγκο και αντιπροσωπεύουν τα κύρια αποτελέσματα της αποστολής.

Είναι σημαντικό να αναφερθεί ότι η τροχιά του δορυφόρου επιτρέπει μόνο μια πολύ μικρή διάρκεια ορατότητας από τον Σταθμό Βάσης (Ground Station) κάθε μέρα, αιχάνοντας τις ανάγκες για αυτονομία και τη σημασία μιας σωστά εφαρμοσμένης μεθόδου FDIR.

Το κύριο συστατικό του υποσυστήματος COMMS είναι η πλακέτα **SatNOGS COMMS** [23], ένας RF πομποδέκτης ανοικτού κώδικα που αναπτύσσεται από την [LibreSpace Foundation](#), βασισμένη στα τηλεπικοινωνιακά πρότυπα CCSDS.

Η επικοινωνία πραγματοποιείται με τη χρήση 2 ζωνών συχνοτήτων στην περιοχή ISM, 436.5 MHz και 2.425 GHz, μέσω μίας κεραίας διπόλων και μίας κατευθυντικής επιφανειακής κεραίας (patch antenna). Η χρήση των συχνοτήτων ISM επιτρέπει την εύκολη ραδιοεραστεχνική πρόσβαση στο δορυφόρο. Η πρώτη (UHF) ζώνη εκπέμπει επίσης ένα περιοδικό σήμα **beacon** (ραδιοφάρος), που περιέχει κρίσιμες πληροφορίες σχετικά με την κατάσταση του δορυφόρου.

Το υποσύστημα τηλεπικοινωνιών είναι επίσης υπεύθυνο για την ανάλυση πλεκτρομαγνητικής συμβατότητας (EMC), καθώς και για το σχεδιασμό και την κατασκευή του Σταθμού Βάσης. Ο τελευταίος θα αποτελέσει μέρος του **SatNOGS** [24], ενός παγκόσμιου δικτύου δορυφορικών επίγειων σταθμών που βασίζεται σε ανοικτές τεχνολογίες και ανοικτά δεδομένα.

2.2.3 Υποσύστημα Ενέργειας (EPS)

Το EPS είναι το υποσύστημα υπεύθυνο για την παραγωγή, κατανομή και αποθήκευση πλεκτρικής ενέργειας στον δορυφόρο. Είναι ένα κρίσιμο κομμάτι του CubeSat, λόγω της άμεσης εξάρτησης και των σχετικά αυξημένων ενεργειακών αναγκών πολλών υποσυστημάτων του, και θεωρείται πως είναι ο πιο πιθανός λόγος αποτυχίας των CubeSat σε τροχιά [25].

Το AcubeSAT έχει επιλέξει έναν συνδυασμό από COTS προϊόντα για το EPS: [26]

- Τα **πλιακά πάνελ** (solar panels) αγοράζονται από την EnduroSat.



Σχήμα 2.5: Η πλακέτα SatNOGS COMMS

[23] Surligas, “[SatNOGS-COMMS](#)”.

[24] White et al., “[Overview of the Satellite Networked Open Ground Stations \(SatNOGS\) Project](#)”.

Πίνακας 2.2: Προϋπολογισμός ισχύος AcubeSAT σε κανονική λειτουργία

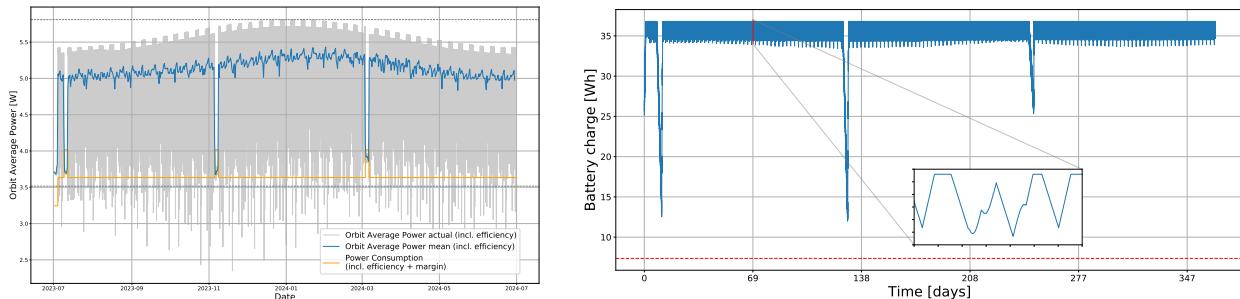
Καταναλωτής	Ισχύς
ADCS	1.10 W
COMMS	0.85 W
EPS	0.99 W
OBC	0.12 W
SU	0.25 W
Σύνολο	3.30 W
Μέση παραγόμενη ισχύς	4.24 W

[25] Langer and Bouwmeester, “[Reliability of CubeSats – Statistical Data, Developers’ Beliefs and the Way Forward](#)”.

[26] Retselis et al., [“AcubeSAT System DDIF”](#).

4 πάνελ διάστασης 3U καλύπτουν τις X και Y πλευρές, και ένα πάνελ 1U καλύπτει την -Z πλευρά.

- Η μονάδα **ελέγχου και διανομής ισχύος** (PCDU) αγοράζεται από την NanoAvionics και προσφέρει 10 προστατευμένα κανάλια με 4 διαφορετικές τιμές τάσης, καθώς και 4 μετατροπείς MPPT.
- Οι **μπαταρίες**, επίσης αγορασμένες από τη NanoAvionics, αποτελούνται από 4 στοιχεία ιόντων λιθίου μιορφής 18650.



2.2.4 Υποσύστημα Διαχείρισης Δεδομένων (OBDH)

Το υποσύστημα OBDH είναι υπεύθυνο για το σχεδιασμό των διεπαφών δεδομένων του CubeSat, καθώς και για τον σχεδιασμό της πλακέτας **On-Board Computer (OBC)**, η οποία είναι επιφορτισμένη με τον έλεγχο των βασικών λειτουργιών του δορυφόρου [27].

Η πλακέτα OBC περιέχει τα βασικά εργαλεία για την εξασφάλιση της αυτονομίας, και βασίζεται σε έναν ανθεκτικό στην ακτινοβολία μικροελεγκτή **Microchip SAMV71Q21RT**⁵, και μια μνήμη MRAM που χρησιμοποιείται για την αποθήκευση κρίσιμων δεδομένων. Η πλακέτα φιλοξενεί επίσης τον μικροελεγκτή του υποσυστήματος ADCS, ως μέτρο εξοικονόμησης χώρου.

Η διεπαφή δεδομένων μεταξύ των υποσυστημάτων του AcubeSAT χρησιμοποιεί ένα Controller Area Network (CAN) bus σε **ψυχρό πλεονασμό** (cold redundancy), το οποίο επιλέχθηκε λόγω της υψηλής αξιοπιστίας του [18]. Οι πλακέτες του AcubeSAT ικανοποιούν τις μηχανικές απαιτήσεις του προτύπου PC/104 [28].

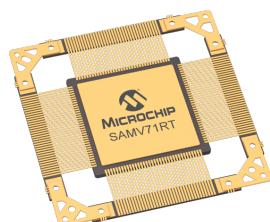
2.2.5 Υποσύστημα Λογισμικού (OBSW)

Το υποσύστημα OBSW είναι υπεύθυνο για το σχεδιασμό και την ανάπτυξη του λογισμικού του νανοδορυφόρου. Η γλώσσα που επιλέχθηκε να χρησιμοποιηθεί στους 4 μικροελεγκτές είναι μια περιορισμένη μιορφή της C++, και ο κώδικας ελέγχεται από μια σειρά προτύπων, στατικών αναλυτών και unit tests [29]. Όλο το λογισμικό εκτελείται επάνω στο λειτουργικό σύστημα πραγματικού χρόνου FreeRTOS.

Σχήμα 2.6: Δυναμική ανάλυση προϋπολογισμού ισχύος.
Αριστερά: Κατανάλωση & παραγωγή ισχύος στην τροχιά.
Δεξιά: Επίπεδο εκφόρτισης μπαταρίας κατά τη διάρκεια της αποστολής

[27] Kanavouras and Pavlakis, *AcubeSAT OBDH DDJF*.

⁵ <https://www.microchip.com/wwwproducts/en/SAMV71Q21RT>



Σχήμα 2.7: Ο μικροελεγκτής SAMV71Q21RT

[18] Bouwmeester et al., “Survey on the Implementation and Reliability of CubeSat Electrical Bus Interfaces”.

[28] PC/104 Embedded Consortium, *PC/104 Specification*.

[29] Kanavouras et al., *AcubeSAT OBSW DDJF*.

2.2.6 Υποσύστημα Χειρισμού (OPS)

Το υποσύστημα Operations είναι υπεύθυνο για το σχεδιασμό των λειτουργιών & διαδικασιών του σκάφους και για τη διασφάλιση της λειτουργικότητας, της δυνατότητας ελέγχου και της παρατηρησιμότητας του δορυφόρου πριν και κατά τη διάρκεια της τροχιάς του.

Κατά τη διάρκεια της αποστολής, το AcubeSAT μπορεί να βρίσκεται σε μία από τις ακόλουθες **λειτουργίες συστήματος**: [30]

- Λειτουργία Εκτόξευσης** (Launch/Off mode): Ο δορυφόρος είναι πλήρως απενεργοποιημένος και κανένα υποσύστημα δεν λαβάνει ισχύ. Αυτή η λειτουργία αναπαριστά την κατάσταση του διαστημικού σκάφους μέσα στον "deployer", όπου δεν επιτρέπεται η ενεργοποίηση κανενός πλεκτρονικού στοιχείου, [31, req. 3.3.3] και το CubeSat πρέπει να βρίσκεται σε πλήρη αδράνεια.
- Λειτουργία Εκκίνησης** (Commissioning mode): Αυτή η λειτουργία ξεκινάει μόλις το CubeSat εξέλθει από τον deployer, δηλαδή όταν ολοκληρωθεί η εκτόξευση. Περιλαμβάνει τις αρχικές ενέργειες εκκίνησης του δορυφόρου, συμπεριλαμβανομένης της εκτύλιξης της κεραίας, και της μείωσης της γωνιακής ταχύτητας. Σε αυτήν τη λειτουργία δεν παράγονται επιστημονικά δεδομένα.
- Κανονική Λειτουργία** (Nominal mode): Εδώ θα βρίσκεται ο δορυφόρος τον περισσότερο χρόνο. Εκτός από τις απαραίτητες λειτουργίες αυτονομίας και τη φόρτιση της μπαταρίας, το CubeSat θα μεταδίδει επίσης τηλεμετρία και επιστημονικά δεδομένα. Εδώ δεν πραγματοποιείται καμία επιστημονική εργασία, εκτός από τυχόν ελέγχους υγείας.
- Επιστημονική Λειτουργία** (Science mode): Εδώ πραγματοποιείται το κύριο πείραμα και παράγονται τα επιστημονικά δεδομένα. Αυτή η λειτουργία περιλαμβάνει τη δράση του υδραυλικού συστήματος, τη λειτουργία του τσιπ μικρορευστομηχανικής, την καλλιέργεια των κυττάρων και την περιοδική λήψη εικόνων με τη χρήση του μικροσκοπίου.

Το AcubeSAT έχει χωρίσει την επιστημονική λειτουργία σε 3 διαφορετικές φάσεις, που ονομάζονται υπο-πειράματα α, β και γ, διαρκούν 72 ώρες το καθένα και εκτελούνται σε διαφορετικά χρονικά σημεία της αποστολής ώστε να διερευνηθεί η χρονική εξάρτηση των παρατηρούμενων αποτελεσμάτων.

- Ασφαλής Λειτουργία** (Safe mode): Συνηθίζεται τα διαστημικά συστήματα να περιλαμβάνουν μια ασφαλή λειτουργία [32, σ. 385], όπου ο δορυφόρος απενεργοποιεί όλα τα μη απαραίτητα συστή-

[30] Zaras et al., *AcubeSAT Mission Description & Operations Plan*.

[31] California Polytechnic State University, *CubeSat Design Specification Rev. 13*.

[32] Aguirre, *Introduction to Space Systems*.

ματα και διαδικασίες, προκειμένου να ανταποκριθεί σε σημαντικές δυσλειτουργίες που δεν μπορούν να διορθωθούν αυτομάτως. Η ασφαλής λειτουργία προορίζεται ως μια καλά καθορισμένη και καλά δοκιμασμένη λειτουργία, η οποία είναι εύκολη στη συντήρηση και μειώνει τον κίνδυνο οποιασδήποτε καταστροφικής βλάβης.

Κάθε λειτουργία συνδέεται με ένα διάγραμμα **λειτουργικής ροής**, το οποίο δείχνει μια υψηλού επιπέδου περιγραφή των διαδικασιών που εκτελούνται κατά τη διάρκεια της [33].

2.2.7 Υποσύστημα Δομής (Structural)

Το υποσύστημα Structural έχει αναλάβει:

- Την ανάλυση και τη διαμόρφωση του COTS σκελετού 3U ([Σχήμα 2.8](#)) που στεγάζει όλα τα εξαρτήματα του CubeSat και είναι κατασκευασμένος από αλουμίνιο. Οι αναλύσεις δονήσεων είναι ιδιαίτερα σημαντικές, καθώς διερευνούν κατά πόσο το CubeSat μπορεί να αντέξει τα μηχανικά φορτία κατά την εκτόξευση.
- Τον πλήρη σχεδιασμό, κατασκευή και συναρμολόγηση του **δοχείου πειράματος** ([Σχήμα 2.10](#)).

2.2.8 Μηχανική Συστήματος (SYE)

Η υποομάδα του Systems Engineering λειτουργεί ως τεχνική αρχή του διορυφόδου. Είναι υπεύθυνη για το συντονισμό των εργασιών και της επικοινωνίας μεταξύ των υποσυστημάτων, τη συμμόρφωση με τα πρότυπα και τις τεχνικές προδιαγραφές, καθώς και τον εντοπισμό & την επίλυση όλων των ζητημάτων που προκύπτουν από τον πολύπλοκο σχεδιασμό ενός διορυφορικού συστήματος.

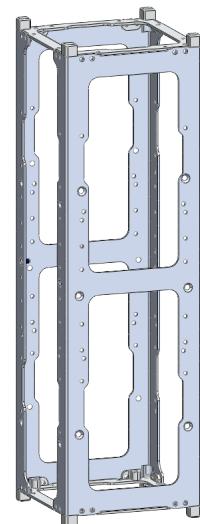
Επιπλέον, η ομάδα SYE είναι υπεύθυνη για ορισμένα τεχνικά κομμάτια που δεν ανήκουν σε κανένα από τα άλλα υποσυστήματα, όπως την ανάλυση αξιοπιστίας (RAMS), την ανάλυση αποτυχιών & αποτελεσμάτων τους (FMEA), την καλωδίωση και το πλάνο κατασκευής & δοκιμών (MAIV).

2.2.9 Επιστημονική Μονάδα (SU)

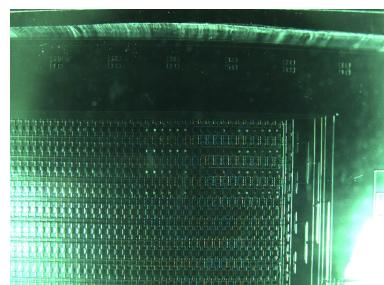
Η υποομάδα Science Unit είναι υπεύθυνη για τη σχεδίαση και την υλοποίηση του επιστημονικού φορτίου της αποστολής, δηλαδή τη μελέτη της επιδράσεων του περιβάλλοντος Χαμηλής Γήινης Τροχιάς (LEO) σε ζυμομύκητες.

Το επιστημονικό φορτίο αποτελείται από τα ακόλουθα λειτουργικά μέρη: [34]

[33] AcubeSAT Team, *AcubeSAT Functional Architecture*.



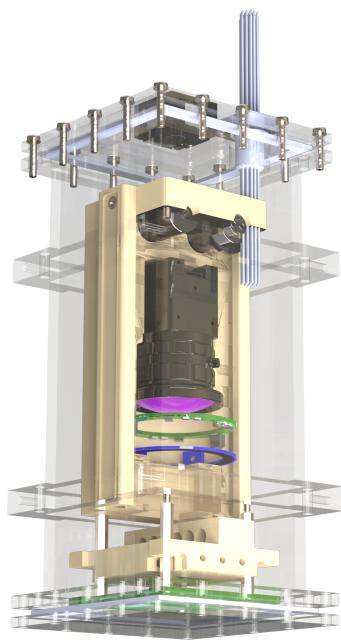
Σχήμα 2.8: Ο COTS σκελετός του CubeSat



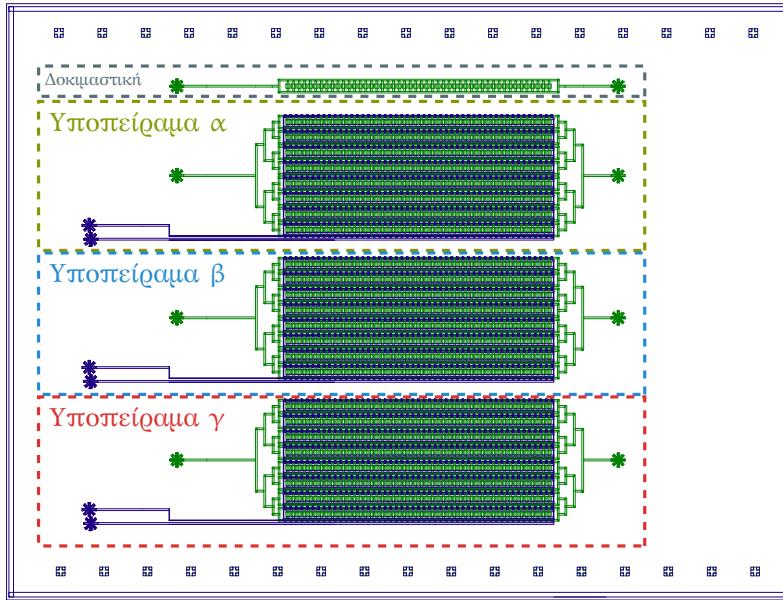
Σχήμα 2.9: Παράδειγμα παραγόμενης εικόνας [34]

[34] Arampatzis et al., *AcubeSAT Payload DDJF*.

- Το **δοχείο πειράματος**, μια δομή αλουμινίου μεγέθους σχεδόν 2U, σε τυπική ατμοσφαιρική πίεση και σχεδιασμένη να φιλοξενεί όλα τα όργανα του πειράματος. Το δοχείο φιλοξενεί επίσης ένα μονόσωμα ("*unibody*") που στηρίζει μηχανικά όλα τα εξαρτήματα.
- Ένα **τσιπ μικρορευστομηχανικής** (microfluidic chip) βασισμένο στο υλικό PDMS [19], που φιλοξενεί 384 θαλάμους ικανούς να εξετάσουν 190 διαφορετικά στελέχη των μυκητών *Saccharomyces Cerevisiae* για κάθε υποπείραμα.
- Ένα **υδραυλικό σύστημα** που αποτελείται από 2 αντλίες, 14 ηλεκτρικές βαλβίδες και 3 σακούλες υγρών.
- Ένα **οπτικό σύστημα** που λειτουργεί ως μικροσκόπιο, το οποίο περιέχει μια κάμερα και μια σειρά από λαμπτήρες, φίλτρα και ένα φακό
- Έναν αριθμό **θερμαντικών σωμάτων** για τον αυστηρό έλεγχο των θερμοκρασιών λειτουργίας των εξαρτημάτων
- Έναν αριθμό πλεοναζόντων **αισθητήρων** για περιβαλλοντικές μετρήσεις
- Μία πλακέτα που περιέχει τον μικροελεγκτή και τα υπόλοιπα εξαρτήματα ελέγχου



Σχήμα 2.10: Διαφανής προβολή του δοχείου πειράματος και του εσωτερικού του



Σχήμα 2.11: Το τσιπ μικρορευστομηχανικής και ο διαχωρισμός του σε 3 επιμέρους υποπειράματα και 1 γραμμή δοκιμής. Οι είσοδοι των υγρών φαίνονται στην αριστερή πλευρά του τσιπ, ενώ οι έξοδοι στα δεξιά. Πράσινο: στρώμα ρούπας. Μπλε: στρώμα ελέγχου.

Το πείραμα επιβάλλει κάποιους περιορισμούς στο σχεδιασμό του FDIR:

1. **Διακοπή** ενός από τα τρία 72ωρα υποπειράματα κατά τη διάρκεια της εκτέλεσης μπορεί να σημαίνει πλήρη απώλεια του υποπειράματος. Ο αντίκτυπος ενός τέτοιου γεγονότος εξαρτάται από τη διάρκεια και τη χρονική στιγμή της εμφάνισής του. Σε κάθε περίπτωση, επαρκής παρατηρησιμότητα θα επιτρέψει στο επίγειο πείραμα να μιληθεί όσο το δυνατόν περισσότερο τις συνθήκες εντός τροχιάς.

2. Το πάγωμα των υγρών στο εσωτερικό του τσιπ και των σωλήνων μπορεί να οδηγήσει σε μόνιμη βλάβη της διάταξης. Ως εκ τούτου, μπορεί να απαιτηθεί η ενεργή θέρμανση ακόμη και κατά τη διάρκεια της ασφαλούς λειτουργίας, μετά την εκτέλεση του πρώτου υπο-πειράματος και την εισροή υγρού στο σύστημα. Αυτά τα χαρακτηριστικά επιβάλλουν περαιτέρω περιορισμούς στην ελάχιστη διαθεσιμότητα του συστήματος, ανάλογα με τις περιβαλλοντικές συνθήκες.

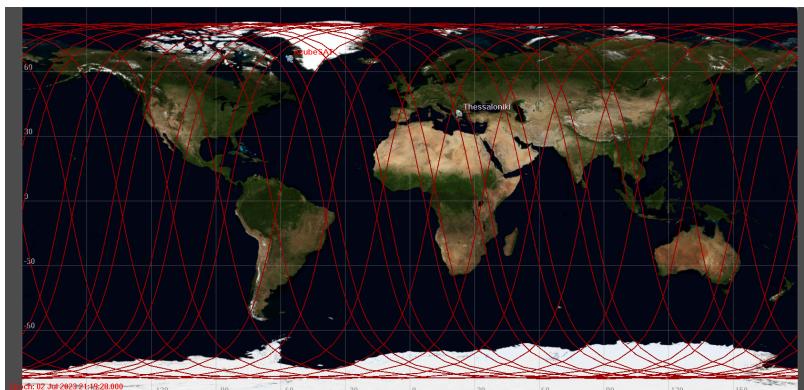
2.2.10 Υποσύστημα Θερμικού Ελέγχου (Thermal)

Η υποομάδα Thermal είναι υπεύθυνη για τη θερμική ανάλυση του δορυφόρου, όπου η εισερχόμενη πλιακή ακτινοβολία συνδυάζεται με την απαγωγή θερμότητας των υποσυστημάτων, με σκοπό τον προσδιορισμό των χειρότερων θερμοκρασιών που αντιμετωπίζει ο δορυφόρος, σε θερμές και ψυχρές συνθήκες.

Τα αποτελέσματα της θερμικής ανάλυσης οδηγούν συνήθως στην εφαρμογή παθητικών ή ενεργητικών μεθόδων θερμικού ελέγχου. Ειδικότερα στο AcubeSAT, χρησιμοποιούνται 3 πλεκτρονικά ελεγχόμενα θερμαντικά σώματα για τις μπαταρίες, το πειραματικό τσιπ και τις βαλβίδες.

2.2.11 Ανάλυση Τροχιάς (Trajectory)

Η υποομάδα Trajectory είναι υπεύθυνη για την ανάλυση της τροχιάς του διαστημικού σκάφους, την αποτίμηση των επιπτώσεων της ακτινοβολίας, τη συμμόρφωση του δορυφόρου με τους κανονισμούς για τα διαστημικά σκουπίδια και την εκτίμηση της διάρκειας ζωής του σε τροχιά.



Σχήμα 2.12: Επίγειο ίχνος μιας πιθανής τροχιάς του AcubeSAT, σχεδιασμένο με χρήση του εργαλείου General Mission Analysis Tool της NASA

Οι απαιτήσεις του AcubeSAT δεν υπαγορεύουν τη χρήση πρωθυπότιχων, πράγμα που σημαίνει ότι η τροχιά του δορυφόρου καθορίζεται αποκλειστικά από την εκτόξευση και δεν μπορεί να τροποποιηθεί κατά την πτήση. Καθώς πληροφορίες για την εκτόξευση είναι άγνωστες μέχρι κάποιο χρονικό διάστημα πριν από την παράδοση του δορυφόρου, εκτελείται ένα σύνολο αναλύσεων ευαισθησίας για

τον προσδιορισμό των επιτρεπόμενων τροχιών [9, 30].

2.3 Χρησιμοποιούμενα εργαλεία

Ως ένα περίπλοκο πρότζεκτ, το AcubeSAT χρησιμοποιεί ένα σύνολο βοηθητικών εργαλειών, έτοιμων ή κατασκευασμένων στην ομάδα, για την εκπλήρωση των τεχνικών και οργανωτικών εργασιών του. Ενδεικτικά παραθέτουμε τα:

- **GitLab**⁶, για διαμοιρασμό κώδικα αλλά και οργάνωση εργασιών & καθηκόντων
- **Open Concurrent Design Tool**⁷, μία πλατφόρμα για διαμοιρασμό τεχνικών δεδομένων και τιμών για τα εξαρτήματα του δορυφόρου
- **L^AT_EX**⁸ και **Overleaf**⁹, για συγγραφή της τεχνικής βιβλιογραφίας και τεκμηρίωσης
- **Requirements Tree**¹⁰, μία διαδραστική προβολή των τεχνικών προδιαγραφών του δορυφόρου
- **Documentation List**¹¹, πλατφόρμα για καταλογοποίηση και πλοήγηση στα έγγραφα που έχει παράξει η ομάδα

[9] Anthopoulos and Retselis, “Orbital Analysis and the Spin-Orbit Coupling for the AcubeSAT Mission”.

[30] Zaras *et al.*, *AcubeSAT Mission Description & Operations Plan*.

⁶ <https://gitlab.com/acubesat>

⁷ <https://ocdt.esa.int/>

⁸ <https://www.latex-project.org/>

⁹ <https://www.overleaf.com/>

¹⁰ <https://acubesat.spacedot.gr/requirements-tree>

¹¹ <https://gitlab.com/acubesat/utilities/mattermost-documentation-bot>

3

Η φιλοσοφία FDIR της πρωτοβουλίας SAVOIR

Στην παρούσα ενότητα θα εξετάσουμε συνοπτικά τον συνδυασμό δύο φιλοσοφιών που χρησιμοποιούνται από ευρωπαϊκούς φορείς με δράση στο χώρο του διαστήματος. Η πρώτη φιλοσοφία συνδέεται με το **πρότυπο ECSS-E-ST-70-41C** [35] του οργανισμού European Cooperation for Space Standardization (ECSS), το οποίο ορίζει το βασικό σύνολο τηλεμετρίας (TM) και τηλεεντολών (TC) που χρειάζεται η πλατφόρμα ενός δορυφόρου για να ελεγθεί από τον σταθμό βάσης.

Η δεύτερη φιλοσοφία βασίζεται στο εγχειρίδιο **SAVOIR-HB-003 "SAVOIR FDIR Handbook"** [36] της πρωτοβουλίας **Space Avionics Open Interface aRchitecture (SAVOIR)**. Το εγχειρίδιο αυτό βασίζεται στις λειτουργίες του προτύπου ECSS-E-ST-70-41C, και προτείνει μία λεπτομερή μεθοδολογία για την ανάπτυξη του FDIR σε ένα διαστημικό πρότζεκτ. Το εγχειρίδιο είναι η βάση της φιλοσοφίας που ακολουθείται στη συνέχεια της εργασίας.

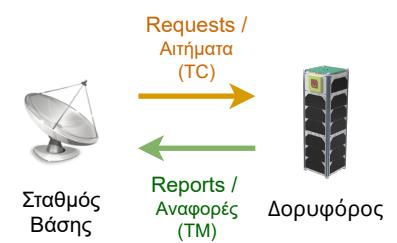
[35] ECSS Secretariat, *ECSS-E-ST-70-41C - Telemetry and Telecommand Packet Utilization*.

[36] Space Avionics Open interface Architecture, *SAVOIR FDIR Handbook*.

3.1 Το Πρότυπο Αξιοποίησης Πακέτων ECSS

Το πρότυπο ECSS-E-ST-70-41C, ή εν συντομίᾳ **PUS** (Packet Utilisation Standard) ορίζει ένα σύνολο από δομημένες **τηλεεντολές** και **τηλεμετρία** που μπορεί να χρησιμοποιήσει ένα διαστημικό σκάφος για να επικοινωνήσει με τη γη. Μπορεί να παρομοιαστεί με μία "γλώσσα" που χρησιμοποιεί ένα σύστημα ώστε οι χειριστές να έχουν πλήρη επίγνωση και πλήρη έλεγχο των λειτουργιών του. Ταυτόχρονα, ορίζει τις βασικές λειτουργίες που πρέπει να πραγματοποιεί ένα αυτόνομο σύστημα με αυστηρούς όρους, και καθορίζει τις έννοιες των **εφαρμογών**, των **παραμέτρων** και των **γεγονότων**.

Το πρότυπο αυτό μπορεί να χρησιμοποιηθεί σε οποιαδήποτε διαστημική αποστολή, και από τους συγγραφείς του συνιστάται η



Σχήμα 3.1: Το μοντέλο μεταφοράς δεδομένων του PUS

προσαρμογή και επιλογή των πιο χρήσιμων "υπηρεσιών" (**services**) από αυτές που προσφέρει. Συνοπτικά, παρέχει τις εξής 20 υπηρεσίες: [35, 37, 38]

- **ST[01]: Request verification**

Παρέχει αναφορές επιβεβαίωσης ή αποτυχίας για εκτελούμενες εντολές. Αυτή η υπηρεσία ουσιαστικά ενημερώνει τους χειριστές για την (επιτυχημένη ή μη) κατάσταση των TC που αποστέλλονται στον δορυφόρο.

- **ST[02]: Device access**

Επιτρέπει την ενεργοποίηση, τον έλεγο και τη ρύθμιση των περιφερειακών συσκευών που βασίζονται σε απλά πρωτόκολλα επικοινωνίας.

- **ST[03]: Housekeeping**

Παράγει περιοδικές αναφορές που περιέχουν τιμές παραμέτρων. Η υπηρεσία αυτή ουσιαστικά συνθέτει τους περιοδικούς ραδιοφάρους (**RF beacons**) του δορυφόρου, χωρίς προηγούμενο ερέθισμα.

- **ST[04]: Parameter statistics reporting**

Επιτρέπει την αναφορά στατιστικών στοιχείων (ελάχιστο, μέγιστο, μέσος όρος, τυπική απόκλιση) για παραμέτρους σε καθορισμένα χρονικά διαστήματα.

- **ST[05]: Event reporting**

Ενημερώνει τους χειριστές όταν λαμβάνουν χώρα γεγονότα.

- **ST[06]: Memory management**

Επιτρέπει την απευθείας ανάγνωση και εγγραφή σε μονάδες μνήμης. Αυτό μπορεί να είναι χρήσιμο για σκοπούς αποσφαλμάτωσης, ανάκτησης δεδομένων της αποστολής, ή φόρτωσης εντελώς νέου λογισμικού. Η υπηρεσία παρέχει επίσης τη δυνατότητα κατεβάσματος και ανεβάσματος αρχείων σε ένα σύστημα αρχείων.

- **ST[07]: Task management (καταργημένη)**

Επιτρέπει τη διακοπή, την αναστολή ή την επανάληψη διεργασιών σε περίπτωση έκτακτης ανάγκης. Η υπηρεσία αυτή έχει αφαιρεθεί από το πρότυπο και παρατίθεται για ιστορικούς λόγους.

- **ST[08]: Function management**

Παρέχει τη δυνατότητα εκτέλεσης προκαθορισμένων ενεργειών που μπορούν να λάβουν περαιτέρω παραμέτρους. Οι ενέργειες αυτές μπορεί να αντιστοιχούν σε πειραματικές διαδικασίες ή γενικότερες λειτουργίες.

- **ST[09]: Time management**

Επιτρέπει την περιοδική αναφορά της τιμής του ρολογιού του CubeSat, για σκοπούς παρατηρησιμότητας και συσχέτισης.

- **ST[10]: Time packet (καταργημένο)**

Χρησιμοποιήθηκε στο παρελθόν για την παροχή πληροφοριών ρολογιού. Η υπηρεσία αυτή έχει αφαιρεθεί από το πρότυπο και παρατίθεται για ιστορικούς λόγους.

- **ST[11]: Time-based scheduling**

[35] ECSS Secretariat, *ECSS-E-ST-70-41C – Telemetry and Telecommand Packet Utilization*.

[37] ECSS Secretariat, *ECSS-E-70-41A – Telemetry and Telecommand Packet Utilization*.

[38] Kaufeler, "The ESA Standard for Telemetry and Telecommand Packet Utilisation".

Εφαρμογή (application process): Κάθε οντότητα (ολικού ή λογισμικού) που μπορεί να λάβει τηλεεντολές και να παράγει τηλεμετρία

Γεγονός (events): Αξιοσημείωτο συμβάν που λαμβάνει χώρα στο δορυφόρο, όπως:

- Αυτόνομη ενέργεια
- Ανιχνευμένη βλάβη ή ανωμαλία
- Προκαθορισμένο βήμα μιας διαδικασίας

Παράμετρος (parameter): Μεταβλητός και αναγνώσιμος περιέκτης δεδομένων που μπορεί να αντιπροσωπεύει:

- Βασικές ρυθμίσεις του συστήματος ή άλλων στοιχείων
- Μετρήσεις αισθητήρων και άλλες τιμές πλημετρίας
- Αποτελέσματα και διαγνωστικά του FDIR

Επιτρέπει στους χειριστές να προγραμματίζουν ("time-tag") τηλεντολές για εκτέλεση σε μελλοντικές χρονικές στιγμές, αντί για αμέσως.

- **ST[12]: On-board monitoring**

Επιτρέπει τον έλεγχο των τιμών των παραμέτρων για να διασφαλιστεί ότι παραμένουν εντός ρυθμιζόμενων ορίων. Κάθε φορά που τα όρια παραβιάζονται, δημιουργείται ένα γεγονός (ST[05]) για περαιτέρω επεξεργασία.

- **ST[13]: Large packet transfer**

Παρέχει μια μέθοδο τμηματοποίησης (segmentation) μηνυμάτων, για συμβολοσειρές που είναι πολύ μεγάλες για να χωρέσουν στο μέγιστο επιτρεπόμενο μήκος για TC ή TM.

- **ST[14]: Real-time forwarding control**

Αυτή η υπηρεσία είναι υπεύθυνη για τον έλεγχο της παραγόμενης τηλεμετρίας που εκπέμπεται άμεσα προς το Σταθμό Βάσης.

- **ST[15]: On-board storage and retrieval**

Επιτρέπει την αποθήκευση των παραγόμενων αναφορών, καθώς και την ανάκτησή τους όταν ο δορυφόρος περνάει επάνω από το Σταθμό Βάσης.

- **ST[16]: On-board traffic management (deprecated)**

Επιτρέπει την παρακολούθηση της κατάστασης και του φόρτου ενός διαύλου δεδομένων. Η υπηρεσία αυτή έχει αφαιρεθεί από το πρότυπο και παρατίθεται για ιστορικούς λόγους.

- **ST[17]: Test**

Η υπηρεσία αυτή παρέχει απλές εντολές για ελέγχους υγείας του δορυφόρου.

- **ST[18]: On-board operations procedure**

Επιτρέπει τη φόρτωση, τον έλεγχο (εκκίνηση, αναστολή, συνέχιση, διακοπή) και τη ρύθμιση των "operations procedures", δλδ. ακολουθιών εντολών σε κάποια ερμηνευμένη γλώσσα προγραμματισμού.

- **ST[19]: Event-action**

Παρέχει τη δυνατότητα αυτόνομης εκτέλεσης TCs όταν λαμβάνει χώρα ένα συμβάν ST[05].

- **ST[20]: On-board parameter management**

Παρέχει τη δυνατότητα ανάγνωσης και αλλαγής τιμών των παραμέτρων του δορυφόρου.

- **ST[21]: Request sequencing**

Επιτρέπει στους χειριστές να φορτώνουν TCs για να εκτελεστούν με διαδοχική σειρά.

- **ST[22]: Position-based scheduling**

Παρέχει τη δυνατότητα εκτέλεσης TC σε συγκεκριμένα σημεία της τροχιάς.

- **ST[23]: File management**

Επιτρέπει την εκτέλεση διαχειριστικών εντολών σε συστήματα αρχείων στον δορυφόρο, όπως αντιγραφή, μετακίνηση, διαγραφή, ή κλείδωμα.

3.2 Το Εγχειρίδιο FDIR του SAVOIR

Το εγχειρίδιο SAVOIR-HB-003 περιγράφει με λεπτομέρεια τη διαδικασία και μεθοδολογίες για εφαρμογή του FDIR, βασισμένο σε αποτελέσματα & διδάγματα από προηγούμενες ευρωπαϊκές αποστολές. Σε αυτό το κεφάλαιο θα σχολιαστούν τα κομμάτια που αφορούν μία αποστολή CubeSat χαμηλού κόστους.

3.2.1 Βήματα σχεδιασμού FDIR

Το εγχειρίδιο αναλύει τα βήματα του σχεδιασμού του FDIR, άρχικα συνδεδεμένα με τις φάσεις μιας τυπικής διαστημικής αποστολής. Έχει σημασία να τονιστεί πως ο σχεδιασμός του FDIR ξεκινάει πριν την υλοποίηση και επαλήθευσή του, από γενικό σε πιο λεπτομερές επίπεδο. Αυτή η λογική "καταρράκτη" που ακολουθείται από τα διαστημικά πρότζεκτ αντιτίθεται στην πιο σύγχρονη και ευέλικτη "agile" λογική των πρότζεκτ μικρού βεληνεκούς. Παρ' όλα αυτά, οι ιδέες του εγχειριδίου μπορούν να εφαρμοστούν σε οποιαδήποτε λογική ανάπτυξης ενός πρότζεκτ.

Τα βήματα της διαδικασίας FDIR ορίζονται ως τα εξής:

- Τεχνικές προδιαγραφές (requirements):** Εδώ, με βάση τις απαιτήσεις της κάθε αποστολής, αναλύονται αντίστοιχα οι απαιτήσεις του συστήματος για αξιοπιστία & διαθεσιμότητα, και εξάγονται οι βασικές προδιαγραφές για το σύστημα FDIR. Σε αυτό το στάδιο δεν λαμβάνονται σχεδιαστικές αποφάσεις και δεν επιλέγονται λύσεις, αλλά καθορίζονται τα προβλήματα που πρέπει να λυθούν και τα βασικά κριτήρια επιτυχίας του συστήματος.

Αν και η διατύπωση τεχνικών προδιαγραφών σε τυχαία γεγονότα και βλάβες για τις οποίες δεν υπάρχει αρκετή διαθέσιμη πληροφορία είναι δύσκολη, το εγχειρίδιο προτείνει ένα σύνολο από τεχνικές προδιαγραφές [36, σ. 42] που μπορούν να χρησιμοποιηθούν από αποστολές. Ένα μεγάλο σύνολο των προδιαγραφών αυτών καλύπτεται και από τη σχεδίαση που παρουσιάζουν με στις επόμενες ενότητες.

[36] Space Avionics Open interface Architecture, *SAVOIR FDIR Handbook*.

- Ορισμός ιδέας (concept):** Εδώ γίνεται σύγκριση των διαθέσιμων γενικών στρατηγικών του FDIR που θα πληρούν τις προδιαγραφές του [Στοιχείου 0](#). Αναλύονται λύσεις όπως πλεονασμός, διασταυρωμένες συνδέσεις, συστήματα τριπλής ψηφοφορίας ή εγγενώς αξιόπιστα (και υψηλού κόστους) εξαρτήματα.
- Ορισμός αρχιτεκτονικής (architecture):** Σε αυτό το βήμα ορίζεται σε ένα γενικό επίπεδο ο τρόπος λειτουργίας του FDIR με βάση τις δυνατότητες του συστήματος, χωρίς τις λεπτομέρειες

υλοποίησης αλλά με αρκετή πληροφορία για τη δομή του συστήματος και τις διεπαφές των στοιχείων του.

3. **Λεπτομερής σχεδιασμός** (detailed design): Προτού ολοκληρωθεί ο πλήρης σχεδιασμός του FDIR, απαιτείται η κατάστρωση του **Failure Mode and Effects Analysis (FMEA)** [39] για το σύστημα και τα υποσυστήματα. Το FMEA πρόκειται για τον κατάλογο όλων των πιθανών αποτυχιών για ένα εξάρτημα του δορυφόρου, όπου καταλογοποιούνται με δομημένο τρόπο οι πιθανές βλάβες και η σοβαρότητά τους.

Στη συνέχεια, για κάθε σημαντικό αντικείμενο του καταλόγου FMEA αναλύεται το **Hardware/Software Interaction Analysis (HSIA)** [40]. Πρακτικά, για κάθε βλάβη ορίζονται σε όρους λογισμικού τα **παρακολουθούμενα δεδομένα** και οι **ενέργειες απομόνωσης & επιδιόρθωσης**.

Πέρα από τα παραπάνω, διατυπώνεται ο ακριβής τρόπος και η θρόνη λειτουργίας του FDIR για κάθε λειτουργία του δορυφόρου, με ιδιαίτερη έμφαση στην Ασφαλή Λειτουργία ([Ενότητα 4.4.1](#)).

4. **Υλοποίηση & επαλήθευση:** Σε αυτό το στάδιο κατασκευάζεται το υλικό και συγγράφεται το λογισμικό που εκτελούν το FDIR. Ακόμα, για τα αντικείμενα της λίστας HSIA, πραγματοποιείται **επαλήθευση** μέσω ανάλυσης ή δοκιμής. Πιο συγκεκριμένα, πρέπει να προσομοιωθεί ή κάθε επίφοβη αποτυχία (feared event), και να επαληθευθεί πως το σύστημα μπορεί να την αντιμετωπίσει με αναμενόμενο τρόπο.
5. **Επικύρωση** (validation): Σε αυτό το στάδιο πραγματοποιούνται δοκιμές του FDIR σε επίπεδο συστήματος, ως μέρος της δοκιμαστικής καμπάνιας¹ ολόκληρης της αποστολής, και πραγματοποιούνται οι τελικές μικρορυθμίσεις.
6. **Προετοιμασία για χειρισμό** (operations preparation): Στο τελευταίο βήμα πραγματοποιούνται οι τελικές αλλαγές στη βιβλιογραφία & τεκμηρίωση για τον απομακρυσμένο χειρισμό της αποστολής.

Αν και όλα τα παραπάνω βήματα έχουν σημασία, για “λιτές” αποστολές πρέπει να εστιαστεί προσοχή στις τεχνικές προδιαγραφές ([Στοιχείο 0](#)), στον ορισμό ιδέας ([Στοιχείο 1](#)), στον λεπτομερή σχεδιασμό ([Στοιχείο 3](#)) και στην επαλήθευση ([Στοιχείο 4](#)). Ένα παράδειγμα εφαρμογής της παραπάνω διαδικασίας βρίσκεται στο [Κεφάλαιο 5](#).

Σε αυτό το σημείο θα πρέπει να τονιστεί πως η διαδικασία σχεδιασμού του FDIR και γενικότερα της μελέτης αξιοπιστίας του συστήματος είναι κάτι που πρέπει να λαμβάνεται υπ' όψιν καθ' όλη

[39] Carlson, *Effective FMEAs*.

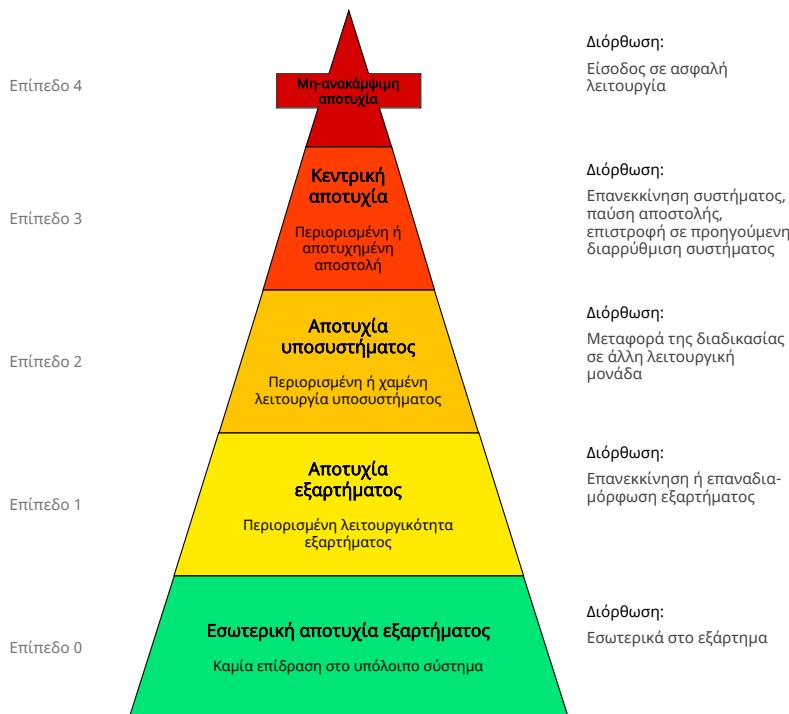
[40] ECSS Secretariat, *ECSS-Q-ST-30-02C - Failure Modes, Effects (and Criticality) Analysis (FMEA/FMECA)*.

¹ Μπορεί να περιέχει τις λειτουργικές δοκιμές συστήματος (ambient tests), ή τις δοκιμές περιβάλλοντος (environmental tests) όπου προσομοιώνονται συνθήκες διαστήματος

τη διάρκεια σχεδιασμού ενός διαστημικού συστήματος, και όχι μόνο στον λεπτομερή σχεδιασμό ή στην επαλήθευση. Ο τομέας του Reliability, Availability, Maintainability and Safety (Αξιοπιστία, Διαθεσιμότητα, Συντηρησιμότητα και Ασφάλεια, RAMS) συνδέεται άμεσα με όλα τα υπόλοιπα υποσυστήματα, λαμβάνοντας τις απαραίτητες εισόδους αλλά και επηρεάζοντας σε μεγάλο βαθμό το σχεδιασμό. Ειδικότερα σε νέα CubeSat projects, ο τομέας της αξιοπιστίας τείνει να μη λαμβάνει τη δέουσα σημασία [14].

3.2.2 Προτεινόμενη αρχιτεκτονική

Το εγχειρίδιο προτείνει κάποιες αρχές για την αρχιτεκτονική ενός συστήματος FDIR, η πρώτη από τις οποίες το εξετάζει ως ένα **ιεραρχικό σύστημα**, όπου μικρής σημασίας αποτυχίες ξεκινούν από **εσωτερικά εξαρτήματα** και διορθώνονται σε επίπεδο εξαρτήματος, ή διαδίδονται στο υπόλοιπο σύστημα και απαιτούν σημαντικότερες ενέργειες ανάκαμψης (Σχήμα 3.2).



[14] Langer et al., “A Reliability Estimation Tool for Reducing Infant Mortality in Cubesat Missions”.

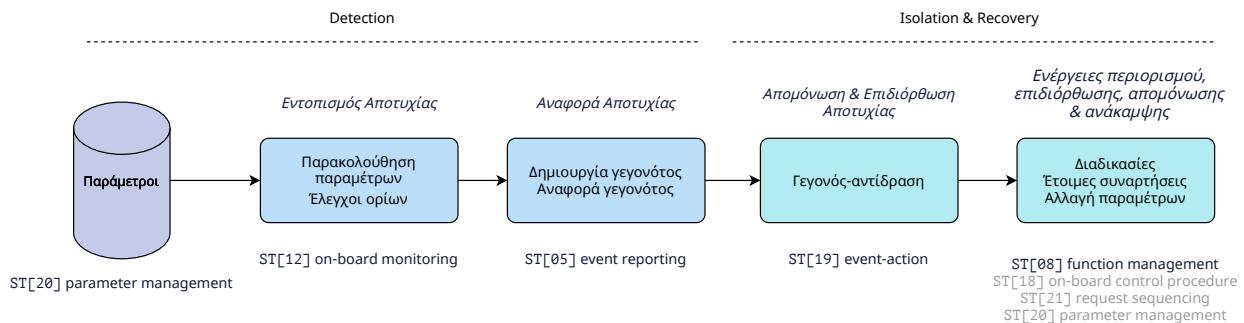
Σχήμα 3.2: Προτεινόμενη ιεραρχία των διαφορετικών επιπέδων FDIR από το SAVOIR. Η κάθε αποστολή μπορεί να επιλέξει διαφορετικά επίπεδα για να απλοποιήσει την υλοποίηση του συστήματος.

Συνεχίζοντας, παρουσιάζεται μία **προσαρμόσιμη** υλοποίηση που εκμεταλλεύεται τη **βάση δεδομένων** του συστήματος και την επικοινωνία με τα **PUS services** ώστε να καλυφθούν πλήρως οι ανάγκες του FDIR. Η λογική αυτή προϋποθέτει τα ακόλουθα:

1. Δεδομένα αξιοπιστίας αποθηκεύονται σε **παραμέτρους** (π.χ. τιμές αισθητήρων, κατάσταση περιφερειακών)
2. Οι παραμέτροι αυτές **παρακολουθούνται** από την υπηρεσία **on-board monitoring**. Όταν κάποια παραμέτρος αποκτήσει τιμή εκτός ορίων, σημαίνει πως έχει εμφανιστεί

κάποια αποτυχία.

3. Κάθε έξοδος από τα όρια δημιουργεί ένα **γεγονός** που τηλεμετράται (TM) στους χειριστές.
4. Κάθε γεγονός είναι συνδεδεμένο με μία **αντίδραση**, αποθηκευμένη στην υπηρεσία event-action.
5. Η αντίδραση αυτή μπορεί να είναι μία ή περισσότερες **διαδικασίες** που διορθώνουν την βλάβη.



Η παραπάνω λογική συνοψίζεται στο **Σχήμα 3.3**. Η σημασία της έγκειται στο ότι προσφέρει έναν **δομημένο** τρόπο το σύστημα να διαχειριστεί οποιοδήποτε πρόβλημα προβλεφθεί, αλλά ταυτόχρονα να δίνει τη δυνατότητα στους χειριστές & κατασκευαστές:

- Να λάβουν όλες τις πληροφορίες για τη λειτουργική κατάσταση των εξαρτημάτων και τιμές των αισθητήρων
- Να παρέμβουν με ασφάλεια εξ' ολοκλήρου στη λειτουργία του FDIR χωρίς μεταβολή του πηγαίου κώδικα
- Να απενεργοποιήσουν εν μέρει ή εντελώς το FDIR αν χρειαστεί για διαγνωστικούς σκοπούς
- Να χρησιμοποιήσουν μία έτοιμη και δοκιμασμένη σουίτα λογισμικού των υπηρεσιών ECSS για να υλοποιήσουν όλα τα παραπάνω, μειώνοντας το κόστος και τον χρόνο ανάπτυξης
- Να αποφευχθεί το ανεξέλεγκτο "hard-coding" διαδικασιών FDIR σε διάφορα σημεία του κώδικα

Σχήμα 3.3: Η ροή δεδομένων για τον εντοπισμό της αποτυχίας

4

FDIR στο AcubeSAT

Στο παρακάτω κεφάλαιο γίνεται ανάλυση περίπτωσης (case study) για την εφαρμογή των παραπάνω (Κεφάλαιο 3) στην αποστολή του AcubeSAT (Κεφάλαιο 2). Πιο συγκεκριμένα, μελετώνται οι σχεδιαστικές αποφάσεις που λήφθηκαν, τα κομμάτια των προτύπων που εφαρμόστηκαν, και οι συγκεκριμένες ενέργειες που εξαρτώνται από την αποστολή [41].

[41] Retselis and Kanavouras, *Acube-SAT FMEA File*.

4.1 Βασικές αρχές του FDIR

Οι απαιτήσεις του FDIR του AcubeSAT προέρχονται κυρίως από τις δυνατότητες του σχεδιασμού του συστήματος (Κεφάλαιο 2) και του πειράματος (Ενότητα 2.2.9). Πιο συγκεκριμένα, οι βασικές αρχές είναι:

1. **Αρθρωτή δομή** (modularity): Το FDIR δεν θα πρέπει να είναι κλειδωμένο (hard-coded) στο λογισμικό, αλλά θα πρέπει να επιτρέπει τις εύκολες μεταβολές και να βασίζεται σε μία δομή μένη βάση δεδομένων.
2. **Αυτονομία**: Ο δορυφόρος θα πρέπει να μπορεί να επιβιώσει σε τροχιά για πάνω από 24h χωρίς επικοινωνία με το σταθμό βάσης, και να μπορεί να συντηρήσει οποιαδήποτε λειτουργία συστήματος, και ειδικότερα το επιστημονικό πείραμα.
3. **Διαμορφωσιμότητα σε τροχιά**: Είναι πολύ δύσκολο να προβλεφθεί η ακριβής συμπεριφορά του δορυφόρου σε τροχιά. Ως εκ τούτου, θα πρέπει να δίνεται η δυνατότητα στους χειριστές για πλήρη διαμόρφωση και ρύθμιση του FDIR αν χρειαστεί. Η διαμορφωσιμότητα αυτή μπορεί να περιλαμβάνει απενεργοποίηση ελέγχων, αλλαγές επικίνδυνων ορίων, ή ακόμα και εγκαστάσταση νέων ελέγχων.

4. **Παρατηρησιμότητα.** προσφέροντας έναν μεγάλο όγκο πληροφορίας στο σταθμό βάσης για τη διάγνωση των προβλημάτων.
5. **Καμία υπόθεση** για λειτουργικότητα εξοπλισμού. Κάθε κατάσταση σφάλματος στο λογισμικό και κάθε μη έγκυρη είσοδος λογισμικού θα πρέπει να θεωρείται πιθανή.
6. **Διαθεσιμότητα:** Το AcubeSAT επωφελείται από μεγαλύτερους χρόνους σε τροχιά, αφού αυτό επιτρέπει το κατέβασμα μεγαλύτερου όγκου δεδομένων και περισσότερα επιστημονικά αποτελέσματα. Επιπλέον, σύντομη διακοπή κάποιου από τα υποστηριζόμενα μπορεί εγγενώς να οδηγήσει σε πλήρη αποτυχία του. Ως εκ τούτου, το AcubeSAT ακολουθεί μία λειτουργική (fail-operational) και όχι ακίνδυνη (fail-safe) λογική, και προσπαθεί να διορθώσει βλάβες χωρίς να διακόψει τη ροή των λειτουργιών του.
7. **Σχεδιασμός** βασισμένος στις αποτυχίες: Το AcubeSAT προσπαθεί να συμπεριλάβει πιθανές αποτυχίες στα βασικά κομμάτια του σχεδιασμού, αυξάνοντας την αξιοπιστία ακόμα και πριν την εφαρμογή του συστήματος FDIR.

Οι παραπάνω απαιτήσεις καλύπτονται από τη λογική του SAVOIR που παρουσιάστηκε στο [Κεφάλαιο 3](#), η οποία υλοποιείται και στο AcubeSAT.

Τα επίπεδα του FDIR στο AcubeSAT είναι τα εξής 3:

Επίπεδο 1 Μονάδα/Εξάρτημα: Εδώ γίνεται εντοπισμός και διόρθωση των αποτυχιών σε επίπεδο ενός εξαρτήματος (π.χ. αισθητήρα θερμοκρασίας). Οι απλές αποτυχίες στο επίπεδο αυτό δεν επηρεάζουν το παραπάνω επίπεδο, και μπορούν να διορθωθούν τυπικά με απλές ενέργειες (π.χ. επανεκκίνηση εξαρτήματος).

Το επίπεδο αυτό το αναλαμβάνει συνήθως ο μικροελεγκτής του κάθε υποσυστήματος. Παρ' όλο που οι αποτυχίες εδώ μπορεί να μη γίνουν αντιληπτές από τα υπόλοιπα υποσυστήματα, η αντίστοιχη τηλεμετρία πρέπει να παραχθεί ώστε για να υπάρχει η κατάλληλη παρατηρησιμότητα από το σταθμό βάσης, ή να προβλεφθούν άλλες βλάβες που δεν έγιναν αντιληπτές από το διορυφόρο.

Επίπεδο 2 Υποσύστημα: Εδώ ανίκουν σοβαρές αποτυχίες που επηρεάζουν την πλήρη λειτουργία ενός υποσυστήματος (π.χ. υπερθέρμανση, καμία απάντηση από το υποσύστημα κλπ.). Οι αποτυχίες αυτές δεν οδηγούν το διορυφόρο απαραίτητα σε παύση, αλλά μπορούν να διορθωθούν με επανεκκινήσεις ή αλλαγές των ρυθμίσεων του υποσυστήματος.

Οι αποτυχίες σε επίπεδο υποσυστήματος και πάνω αναλαμβάνονται από τον μικροελεγκτή του On-Board Computer (OBC), που περιέχει το λογισμικό για τον εντοπισμό και επιδιόρθωσή τους. Σε περίπτωση όμως αποτυχίας του On-Board Computer (OBC), το Attitude Determination and Control Subsystem (ADCS) αναλαμβάνει την ευθύνη του Fault Detection, Isolation and Recovery (Εντοπισμός, Απομόνωση και Διόρθωση Αποτυχιών, FDIR).

Επίπεδο 3 Σύστημα: Εδώ ανίκουν αποτυχίες οι οποίες δεν μπορούν να διορθωθούν αυτόνομα από το σύστημα (π.χ. ελλιπής ισχύς, υπερβολικά μεγάλη γωνιακή ταχύτητα). Ο τρόπος επιδιόρθωσης εδώ είναι συνήθως η είσοδος σε ασφαλή λειτουργία, όπου πραγματοποιούνται μόνο οι πιο βασικές και ζωτικές διαδικασίες του δορυφόρου, μέχρι το σφάλμα να επιδιορθωθεί από τον σταθμό βάσης.

4.2 Μελέτη διαφορετικών αρχιτεκτονικών

Προτού μελετήσουμε τη δομή του FDIR, έχει νόημα να μελετήσουμε διαφορετικές αρχιτεκτονικές αποφάσεις που μπορούν να ληφθούν κατά το σχεδιασμό ενός μικρού δορυφορικού συστήματος. Πιο συγκεκριμένα, θα συγκρίνουμε τις δομές [42]:

[42] Birolini, *Reliability Engineering*.

1. **Απλού εξαρτήματος**
2. **Διπλού πλεονασμού**, όπου έχουμε δύο όμοια εξαρτήματα που πραγματοποιούν την ίδια λειτουργία, και το ένα μπορεί να αντικαταστήσει το άλλο. Η συνδεσμολογία αυτή μπορεί να ακολουθεί με τη λογική του **ψυχρού πλεονασμού** ή του **ενεργού πλεονασμού**.

Στον **ψυχρό πλεονασμό**, μόνο ένα από τα εξαρτήματα λειτουργεί κάθε φορά, και ένα εξωτερικό κύκλωμα¹ ελέγχει ποιό από τα δύο δεν βρίσκεται σε κατάσταση σφάλματος, για να του παραδώσει τον έλεγχο.

¹Η τα ίδια τα εξαρτήματα, στην περίπτωση του θερμού πλεονασμού [36, σ. 20]

Στον **ενεργό πλεονασμό**, και τα δύο εξαρτήματα είναι λειτουργικά ταυτόχρονα, πραγματοποιώντας τις ίδιες διαδικασίες, και το ένα εξάρτημα μπορεί να αναλάβει τις λειτουργικές ευθύνες σε περίπτωση βλάβης του άλλου.

3. **Τριπλού πλεονασμού με ψηφοφορία** (triple modular redundancy), όπου έχουμε τρία εξαρτήματα που πραγματοποιούν συνεχώς τους ίδιους υπολογισμούς, και ένα κύκλωμα ψηφοφορίας επιλέγει την έξοδο που έχει την πλειοψηφία από τα 3 εξαρτήματα.

Η σύγκριση των συστημάτων αυτών θα γίνει σε επίπεδο μόνιμων

και προσωρινών βλαβών:

- Στις **μόνιμες βλάβες** εξετάζουμε συμβάντα (π.χ. ακτινοβολία, απώλεια μνήμης) τα οποία δεν μπορούν να διορθωθούν και οδηγούν σε απώλεια της αποστολής. Η μελέτη εδώ γίνεται με βάση την έννοια της αξιοπιστίας (reliability) R , δηλαδή της πιθανότητας το σύστημα να επιζήσει εντός μιας χρονικής περιόδου t .
- Στις **προσωρινές βλάβες** εξετάζουμε συμβάντα (π.χ. αλλαγές bit, bugs λογισμικού, λάθο στη μεταφορά μπονυμάτων) που είναι παροδικά και επηρεάζουν συνήθως μόνο μία μέτρηση, μία ενέργεια ή ένα αποτέλεσμα. Εδώ η μελέτη γίνεται με βάση την έννοια του ρυθμού βλαβών (failure rate) λ , δηλαδή της συχνότητας των αποτυχιών.

Στην παρούσα ανάλυση δεν είναι εύκολο να χρησιμοποιηθούν ακριβείς τιμές αξιοπιστίας και ρυθμού αποτυχιών για τα εξαρτήματα, καθώς δεν υπάρχουν στη βιβλιογραφία διαθέσιμα στοιχεία μεγάλης κλίμακας για τις αποτυχίες εξαρτημάτων στο διάστημα [43]. Επομένως, η ανάλυση θα γίνει σε ένα εύρος τιμών αξιοπιστίας και ρυθμού σφαλμάτων.

Ενδεικτικά, αναφέρουμε ότι η τάξη μεγέθους του ρυθμού αποτυχιών μιας συσκευής λόγω ακτινοβολίας μπορεί να εκτείνεται μεταξύ 1×10^{-11} ως $1 \times 10^{-1} \frac{\text{αποτυχίες}}{\text{δευτερόλεπτο}}$, εξαρτώμενη από το περιβάλλον ακτινοβολίας, τη σοβαρότητα της βλάβης, και την κατασκευή της κάθε συσκευής [44, σ. 158–159].

Ο κώδικας που χρησιμοποιήθηκε για όλα τα παρακάτω παραδείγματα βρίσκεται στο [Παράρτημα B](#).

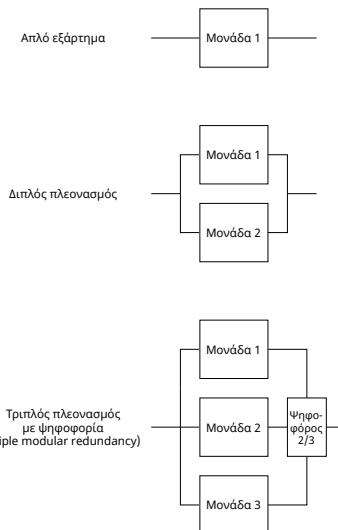
4.2.1 Μόνιμες βλάβες (ανάλυση αξιοπιστίας)

Ιδεατός ψηφοφόρος

Αν θεωρήσουμε ότι η αξιοπιστία ενός εξαρτήματος είναι R , η αξιοπιστία δύο εξαρτημάτων συνδεδεμένων παράλληλα είναι [42, σ. 31]:

$$R_2 = 2R - R^2 = 1 - (1 - R)^2 \quad (4.1)$$

Η (4.1) ισχύει τόσο στον ψυχρό, όσο και στον ενεργό πλεονασμό, καθώς θεωρούμε ότι το σύστημα μπορεί να ανακάμψει από μόνιμες βλάβες και στις δύο περιπτώσεις, μετακινώντας τη "δικαιοδοσία" ελέγχου στο ανεπηρέαστο εξάρτημα.



Σχήμα 4.1: Οπτική απεικόνιση των διαφορετικών μορφών σύνδεσης που θα εξεταστούν

[43] ESA TEC-QCD, *Effective Reliability Prediction for Space Applications*.

[44] Gupta, "Analysis of Single Event Radiation Effects and Fault Mechanisms in SRAM, FRAM and NAND Flash".

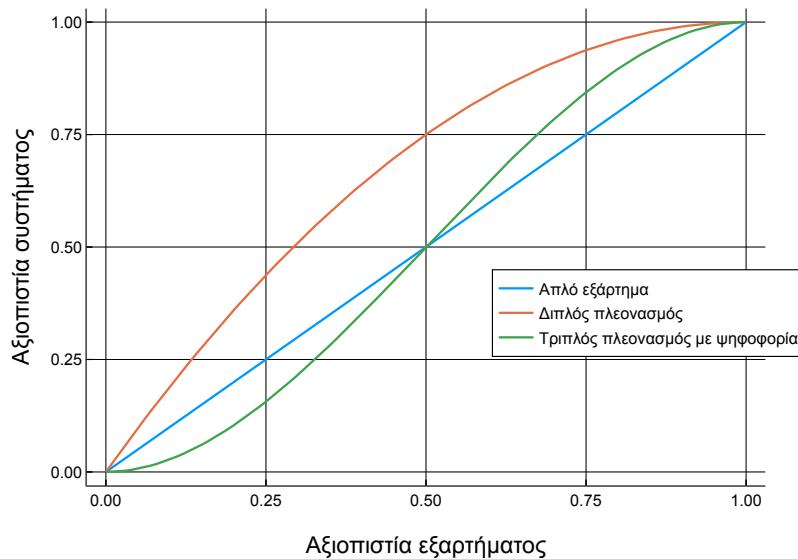
[42] Birolini, *Reliability Engineering*.

Σε όλες τις αναλύσεις υποθέτουμε ότι τα ευάλωτα εξαρτήματα είναι πανομούτυπα και έχουν τα ίδια χαρακτηριστικά αξιοπιστίας.

Η αξιοπιστία του συστήματος τριπλής ψηφοφορίας είναι [42, σ. 31]:

$$\begin{aligned}
 R_3 &= \sum_{i=k}^n \binom{n}{i} R^i (1-R)^{n-i} \\
 &= \sum_{i=2}^3 \binom{3}{i} R^i (1-R)^{3-i} \\
 &= \binom{3}{2} R^2 (1-R) + \binom{3}{3} R^3 (1-R)^0 \\
 &= 3R^2 - 3R^3 + R^3 = 3R^2 - 2R^3
 \end{aligned} \tag{4.2}$$

Εδώ αξίζει να σημειωθεί πως το σύστημα τριπλής ψηφοφορίας θεωρεί έναν απλό ψηφοφόρο, και σε περίπτωση μόνιμης βλάβης και των δύο ευάλωτων εξαρτημάτων, το σύστημα θεωρείται πως δεν μπορεί να ανακάμψει.



Σχήμα 4.2: Ανθεκτικότητα σύνθετου συστήματος σε μόνιμες βλάβες (θεωρώντας αλάνθαστο ψηφοφόρο — μεγαλύτερο R είναι καλύτερο)

Τα αποτελέσματα εφαρμογής των (4.1) και (4.2) φαίνονται στο Σχήμα 4.2. Μπορεί να διακριθεί πως η χρήση δύο πλεονάζοντων εξαρτημάτων αυξάνει σε έναν βαθμό την αξιοπιστία του συστήματος, σε σχέση με το απλό εξάρτημα, καθώς οι βλάβες του ενός εξαρτήματος μπορούν να διορθωθούν, αλλά οι βλάβες και στα δύο είναι καταστροφικές. Αντίθετα, ο τριπλός πλεονασμός εισάγει περισσότερα σημεία αποτυχίας. Καθώς η αποτυχία οποιουδήποτε συνδυασμού 2 εκ των 3 εξαρτημάτων είναι καταστροφική, ο τριπλός πλεονασμός υστερεί στις μόνιμες βλάβες σε σχέση με τον διπλό.

Ευάλωτος ψηφοφόρος

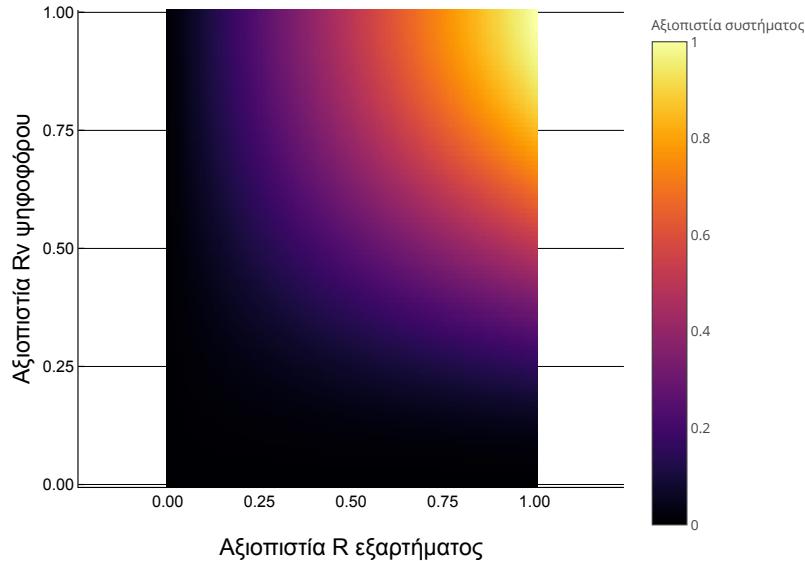
Ιδιαίτερο ενδιαφέρον έχει η μελέτη της περίπτωσης του ευάλωτου κυκλώματος ψηφοφορίας στην περίπτωση τριπλού πλεονασμού.

Στα πλαίσια της ανάλυσης, όταν θεωρήσουμε μία ανεξάρτητη αξιοπιστία R_v για αυτό το κύκλωμα. Από [42, σ. 31], έχουμε:

$$R_{3+v} = (3R^2 - 2R^3)R_v \quad (4.3)$$

$$= R_3 R_v \quad (4.4)$$

και η αξιοπιστία του συστήματος απλώς πολλαπλασιάζεται με την αξιοπιστία του ψηφοφόρου, που μπορεί να μειώσει γραμμικά την αξιοπιστία του συστήματος τριπλού πλεονασμού με ψηφοφορία.



[42] Birolini, *Reliability Engineering*.

Σχήμα 4.3: Ανθεκτικότητα σύνθετου συστήματος σε μόνιμες βλάβες (θεωρώντας ευάλωτο ψηφοφόρο — μεγαλύτερο R είναι καλύτερο)

4.2.2 Προσωρινές βλάβες (ανάλυση ρυθμού βλαβών)

Αν θεωρήσουμε λ τον ρυθμό αποτυχίας του ενός εξαρτήματος, στον ψυχρό πλεονασμό ο ρυθμός αποτυχίας είναι επίσης λ, καθώς μόνο το ένα εξάρτημα είναι ενεργό κάθε φορά, και το απενεργοποιημένο δεν μπορεί να εμφανίσει βλάβη.

Στην περίπτωση ενεργού πλεονασμού, και τα δύο εξαρτήματα μπορεί να αποτύχουν, με ρυθμό λ το καθένα. Θεωρούμε ότι οι αποτυχίες των εξαρτημάτων ακολουθούν κατανομή Poisson [42, σ. 35]. Τότε η πιθανότητα να συμβούν k αποτυχίες σε ένα χρονικό διάστημα δίνεται από τη σχέση [45, σ. 60]:

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \quad (4.5)$$

Για απλοποίηση της ανάλυσης, θα θεωρήσουμε ότι στο χρονικό διάστημα που ορίζεται από την παραμέτρο λ μπορεί να συμβεί μόνο 1 βλάβη. Αυτή η υπόθεση είναι ασφαλής, καθώς μπορούμε να μελετήσουμε μικρά χρονικά διαστήματα (συνήθως δευτερόλεπτα), όπου είναι υπερβολικά ασυνήθιστο να συμβούν περισσότερες από 1 βλάβες.

$$P = P(X) = \lambda e^{-\lambda} \quad (4.6)$$

Θεωρούμε πως οι αποτυχίες των εξαρτημάτων είναι ανεξάρτητα γεγονότα.

[42] Birolini, *Reliability Engineering*.

[45] Yates and Goodman, *Probability and Stochastic Processes*.

Θα αναζητήσουμε το σημείο όπου η πιθανότητα 2 αποτυχιών είναι ίση με το 1% της πιθανότητας 1 αποτυχίας, δηλαδή $P(X = 2) = 0.01P(X = 1) \implies \frac{\lambda^2 e^{-\lambda}}{2!} = 0.01\lambda e^{-\lambda} \implies \lambda = 0.02$. Άρα για $\lambda < 2 \times 10^{-2}$ η υπόθεση είναι ασφαλής.

Στον ενεργό διπλό πλεονασμό δεν είναι πάντα δυνατός ο εντοπισμός της λάθος και της σωστής εξόδου, επομένως μία βλάβη σε οποιοδήποτε από τα δύο εξαρτήματα θεωρείται βλάβη του συστήματος. Άρα, αν X_1 και X_2 τα ενδεχόμενα βλάβης του 1^{ου} και του 2^{ου} εξαρτήματος αντίστοιχα:

$$\begin{aligned} P_2 &= P(X_1 \text{ ή } X_2) = 1 - (1 - P(X))^2 \\ &= 1 - (1 - \lambda e^{-\lambda})^2 \\ &= 2\lambda e^{-\lambda} - \lambda^2 e^{-2\lambda} \end{aligned} \quad (4.7)$$

Η (4.7) δεν ακολουθεί την κατανομή Poisson, άρα δεν είναι εύκολο να εξάγουμε άμεσα τον ρυθμό αποτυχίας του συστήματος. Μπορούμε όμως καταχρηστικά να εξάγουμε τον μέσο ρυθμό αποτυχίας, έστω λ_2 , προσπαθώντας να προσαρμόσουμε την κατανομή Poisson στο μη-Poisson αποτέλεσμα:

$$\lambda_2 e^{-\lambda_2} = P_2 \quad (4.8)$$

Η (4.8) μπορεί να λυθεί αξιοποιώντας τη συνάρτηση Lambert W [46] που ορίζεται ως εξής:

$$\text{για } w, z \in \mathbb{R}: \quad we^w = z \iff w = W_k(z) \text{ για κάποια } k \in \mathbb{Z} \quad (4.9)$$

Επομένως μπορούμε να υπολογίσουμε το λ_2 :

$$\lambda_2 = -W_k(-P_2) \quad (4.10)$$

ΣΤΟ ΣΥΣΤΗΜΑ με τριπλό πλεονασμό έχουμε η πιθανότητα συνολικής αποτυχίας προκύπτει αν αποτύχουν τουλάχιστον 2 από τα εξαρτήματα. Άρα, αντίστοιχα με την (4.2) :

$$\begin{aligned} P_3 &= 3P^2 - 2P^3 \\ &= 3(\lambda e^{-\lambda})^2 - 2(\lambda e^{-\lambda})^3 \end{aligned} \quad (4.11)$$

και, αντίστοιχα με την (4.10):

$$\lambda_3 = -W_k(-P_3) \quad (4.12)$$

Τα αποτελέσματα των παραπάνω μπορούν να υπολογιστούν αριθμητικά και φαίνονται στο Σχήμα 4.4.

4.2.3 Συμπεράσματα

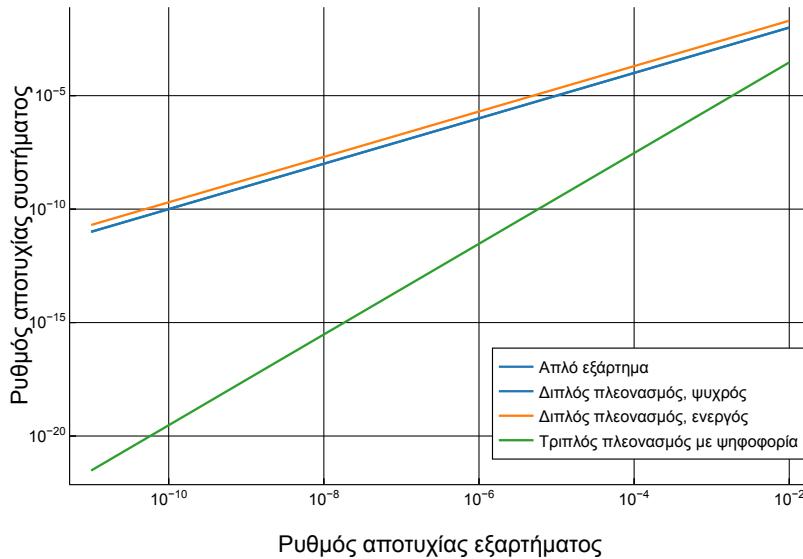
Τα αποτελέσματα των Σχημάτων 4.2 και 4.4 συνοψίζονται στον Πίνακα 4.1, μαζί με την πολυπλοκότητα πρακτικής υλοποίησης της κάθε αρχιτεκτονικής.

Εναλλακτικά μπορούμε να αγνοήσουμε τον αμελητέο 2^ο όρο της (4.7), και να έχουμε $P_2 = 2\lambda e^{-\lambda}$. Επειδή $e^x \approx 1$ πολύ κοντά στο $x = 0$, η (4.8) γίνεται:

$$\lambda_2 = 2\lambda$$

Το παραπάνω επιβεβαιώνεται και διασθητικά. Αφού εισάγουμε 2 εξαρτήματα, ο ρυθμός σφάλματος διπλασιάστηκε.

[46] Weisstein, *Lambert W-Function*.



Σχήμα 4.4: Ανθεκτικότητα σύνθετου συστήματος σε παροδικά σφάλματα (θεωρώντας αλάνθαστο ψηφιοφόρο)

Όπως φαίνεται, η κάθε λύση συνοδεύεται από ένα διαφορετικό σύνολο προτερημάτων και αρνητικών. Η χρήση της ψηφιοφορίας μπορεί να μειώσει συντριπτικά τον ρυθμό αποτυχιών για προσωρινές βλάβες, ενώ η χρήση του καθαρού πλεονασμού βελτιώνει τη συμπεριφορά στις μόνιμες βλάβες και επιτρέπει την ανάκαμψη από καταστροφικά σφάλματα.

Ανθεκτικότητα σε			
Αρχιτεκτονική	Μόνιμες βλάβες	Προσωρινές βλάβες	Πολυπλοκότητα
Απλό εξάρτημα	Κανονική	Κανονική	Μικρή
Διπλός πλεονασμός, ψυχρός	Πολύ καλή	Κανονική	Μέτρια
Διπλός πλεονασμός, ενεργός	Πολύ καλή	Χειρότερη	Μέτρια
Τριπλός πλεονασμός με ψηφιοφορία	Καλύτερη (αν $R > 0.5$)	Εξαιρετική	Μεγάλη

Η τελική επιλογή αρχιτεκτονικής εξαρτάται από τις απαρτίσεις & προδιαγραφές του εκάστοτε πρότζεκτ. Για παράδειγμα, σε επανδρωμένες αποστολές είναι συνηθισμένο να επιλέγεται χρήση τριπλού πλεονασμού με ψηφιοφορία στον υπολογιστή πτήσης, ώστε κανένα σφάλμα να μπορεί να διακινδυνεύσει την ασφάλεια των επιβατών.

Πίνακας 4.1: Σύνοψη αποτελεσμάτων μελέτης διαφορετικών αρχιτεκτονικών

Στην αποστολή AcubeSAT, όπου χρησιμοποιούνται έτοιμα εξαρτήματα της αγοράς (COTS) ευάλωτα σε ακτινοβολία, η έμφαση δίνεται στην αποφυγή των καταστροφικών επιπτώσεων. Οι προσωρινές βλάβες, αν και μπορούν δυνητικά να μειώσουν το επιστημονικό αποτέλεσμα και τη διαθεσιμότητα του συστήματος, θεωρούνται λιγότερο κρίσιμες από τις μόνιμες. Ειδικά στα προγράμματα CubeSat, πρέπει να λαμβάνεται υπ' όψιν και η **πολυπλοκότητα** της επιλεγμένης αρχιτεκτονικής, αρχικά σε σχέση με τους διαθέσιμους πόρους και χρόνο υλοποίησης, αλλά και επειδή μία περίπλοκη

αρχιτεκτονική μπορεί από μόνη της να μειώσει την αξιοπιστία του συστήματος, αν δεν υλοποιηθεί και δεν δοκιμαστεί σωστά.

Η αρχιτεκτονική που επιλέχθηκε στο AcubeSAT είναι η εξής:

- Σε δίκτυα **αισθητήρων** (π.χ. θερμοκρασίας, πίεσης, μαγνητικού πεδίου) εφαρμόζεται διπλός πλεονασμός, και δίνεται η δυνατότητα διόρθωσης της απώλειας κάποιου αισθητήρα.
- Σε εξαρτήματα με **εμπειρία πτήσης** (flight heritage) που έχουν αποδείξει τη λειτουργικότητά τους σε άλλες αποστολές, θεωρείται πως το R είναι αρκετά υψηλό, και χάριν ευκολίας δεν υλοποιείται κάποια μορφή πλεονασμού.
- Ειδικά για τους μικροελεγκτές, η λογική που προτιμήθηκε είναι η χρήση ενός **ακριβότερου** μικροελεγκτή **ανθεκτικού στην ακτινοβολία** με εγγενώς μεγαλύτερο R ([Ενότητα 2.2.4](#)), αντί για προσπάθεια αύξησης του R με χρήση πλεονασμού.

4.3 Λεπτομερείς Ενέργειες του FDIR

Στην παρακάτω ενότητα παρουσιάζεται η λίστα με τους πιθανούς τρόπους εντοπισμού, πρόληψης και αντιμετώπισης των σφαλμάτων που εμφανίζονται στον δορυφόρο. Για την κάθε πιθανή αποτυχία που εντοπίζεται στο FMEA [47], επιλέγεται ένας ή περισσότεροι τρόποι εντοπισμού και διόρθωσης από αυτούς τους πίνακες, και προσαρμόζονται στο εκάστοτε στοιχείο.

[47] Retselis and Kanavouras, *Acube-SAT FMEA Worksheet*.

4.3.1 Τρόποι εντοπισμού αποτυχιών

Επίπεδο	Μέθοδος	Περιγραφή
Μονάδα	Αυτο-δοκιμή δεν επιστρέφει σωστή τιμή	Τα συνδεδεμένα περιφερειακά μπορεί να διαθέτουν δυνατότητες αυτο-δοκιμής (self-test) ή να έχουν αποθηκευμένες προκαθορισμένες τιμές που δεν μπορούν να μεταβληθούν.
Μονάδα	Τιμή εκτός ορίων	Μη αναμενόμενες τιμές των αισθητήρων μπορεί να σημαίνουν επικίνδυνες περιβαλλοντικές συνθήκες, ή απλά ελαττωματικούς (π.χ. SEFI) αισθητήρες. Ο διπλός ή τριπλός πλεονασμός μπορεί να αυξήσει την παρατηρησιακή της σε αυτή την περίπτωση.
Μονάδα	Τιμές κοντινών αισθητήρων πολύ διαφορετικές	
Μονάδα	Καμία απάντηση ή ασυνάρτητες τιμές	Η μονάδα ή η διεπαφή μπορεί να είναι ελαττωματική
Μονάδα	Υπερβολικά μεγάλη κατανάλωση ρεύματος	Μπορεί να παρατηρηθεί σε μονάδες που πάσχουν από βραχυκυκλώματα ή SEL
Μονάδα	Λάθος σε αλγόριθμο εντοπισμού & διόρθωσης σφαλμάτων	
Υποσύστημα	Καμία απάντηση σε εντολές CAN	Μπορεί να οφείλεται σε αποτυχία του υποσυστήματος, ή του πομποδέκτη CAN
Σύστημα	Καμία τηλεεντολή δεν έχει ληφθεί για κάποιο χρονικό διάστημα	Ο GS στέλνει περιοδικές εντολές "ανανέωσης" στον δορυφόρο σε κάθε πέρασμα. Η μη λήψη σημαίνει απώλεια επικοινωνίας.
Λογισμικό	Εξαίρεση υλικού	Μπορεί να οφείλεται σε διαίρεση με το 0, προσπάθεια πρόσβασης σε μη προσβάσιμη μνήμη κλπ.
Λογισμικό	Αποτυχημένος έλεγχος ορισμάτων	
Λογισμικό	Μη αναμενόμενη επανεκκίνηση μικροελεγκτή	
Λογισμικό	Διαφορά στις τιμές μεταβλητών με πλεονασμό	

Πίνακας 4.2: Τρόποι εντοπισμού αποτυχιών FDIR του AcubeSAT

4.3.2 Προληπτικές δράσεις

#	Επίπεδο	Δράση
1	Υλικό	Μικροελεγκτής ανθεκτικός στην ακτινοβολία
2	Υλικό	Κυκλώματα παρακολούθησης μικροελεγκτή (watchdogs)
3	Υλικό	Κυκλώματα παρακολούθησης συστήματος (watchdogs)
4	Υλικό	Αλγόριθμοι εντοπισμού & διόρθωσης σφαλμάτων
5	Υλικό	Διπλός πλεονασμός στη μνήμη λογισμικού
6	Υλικό	Κυκλώματα περιορισμού ρεύματος
7	Υλικό	Προστασία υπέρτασης και υπερέντασης υποσυστημάτων
8	Υλικό	Χρήση προσωρινής μνήμης MRAM ανθεκτικής στην ακτινοβολία
9	Υλικό	Διπλός πλεονασμός στους αισθητήρες
10	Υλικό	Υποβιβασμός προδιαγραφών σε πλεκτρονικά ισχύος
11	Υλικό	Τοποθέτηση ευάλωτων εξαρτημάτων στο κέντρο, ώστε να μειωθεί η επίδραση της ακτινοβολίας
12	Υλικό	Επιλογή εξαρτημάτων με αποτελέσματα δοκιμών ακτινοβολίας
13	Υλικό	Διπλός πλεονασμός στον δίαυλο CAN
14	Υλικό	Βίτιστοι στον δίαυλο CAN
15	Λογισμικό	Αλγόριθμοι εντοπισμού & διόρθωσης σφαλμάτων
16	Λογισμικό	Σύστημα αρχείων ανθεκτικό σε βλάβες και απώλεια ισχύος
17	Λογισμικό	Φιλτράρισμα εισόδων ADCS
18	Διαχειριστικά	Αυτόματες περιοδικές επανεκκινήσεις
19	Διαχειριστικά	Αποφυγή επικίνδυνων διεργασιών σε περιοχές με υψηλό υπόβαθρο ακτινοβολίας
20	Διαχειριστικά	Αξιολόγηση των διαθέσιμων πόρων & προϋπολογισμών πριν από την έναρξη υποεργάματος

Πίνακας 4.3: Προληπτικές δράσεις FDIR του AcubeSAT

4.3.3 Διορθωτικές δράσεις

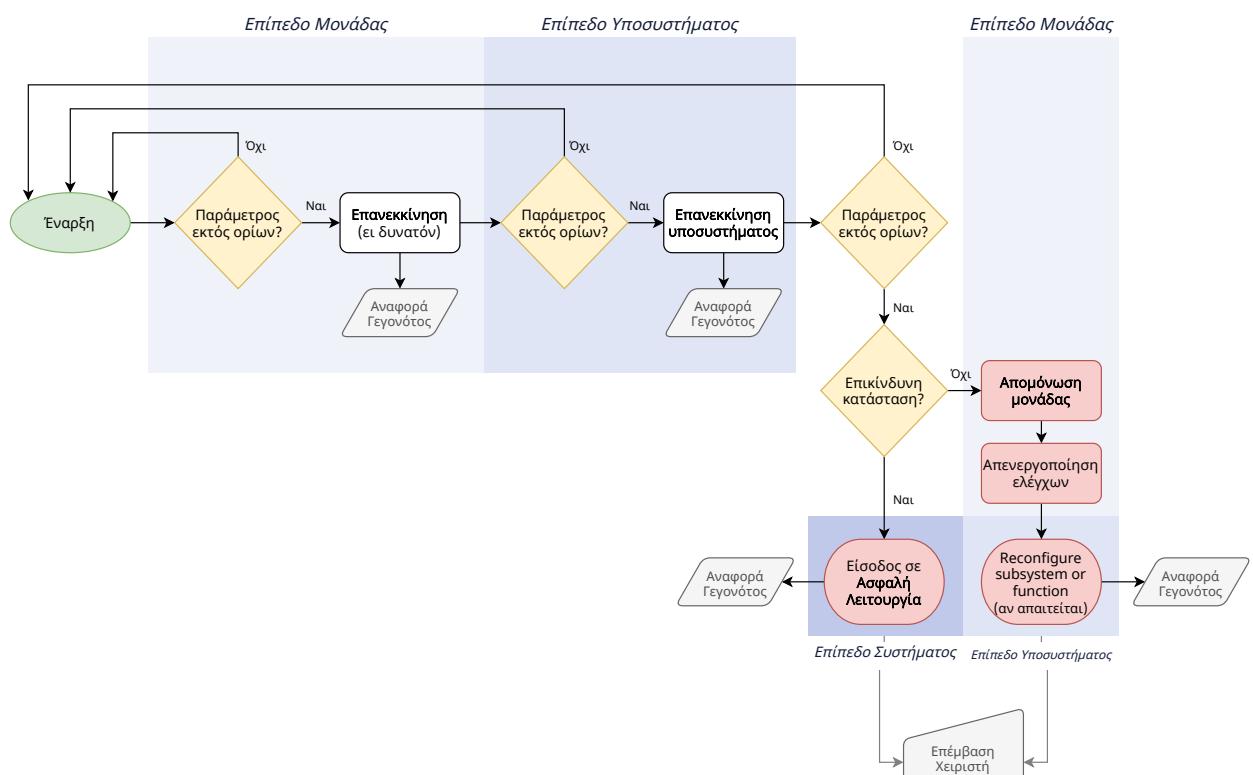
Επίπεδο	Δράση	Περιγραφή
	Καμία	Πολλές φορές υπάρχουν προβλήματα τα οποία διορθώνονται αυτόνομα, ή δεν υπάρχει τρόπος διόρθωσής τους (και δεν επηρεάζουν σημαντικά τη λειτουργία του υπόλοιπου δορυφόρου).
Μονάδα	Επανεκκίνηση μονάδας	Αυτή η μέθοδος ανάκαμψης συνδέεται με προσωρινά προβλήματα που μπορεί να εμφανιστούν στο λογισμικό ή το υλικό μιας συσκευής. Τέτοιες βλάβες μπορεί να προέρχονται από λάθη σχεδιασμού (bugs) που δεν δοκιμάστηκαν εκτενώς, ή από αλλαγές κατάστασης των τρανζίστορ που οφείλονται σε ακτινοβολία.
Μονάδα	Διακοπή και ανάκτηση ρεύματος	Μερικά προβλήματα μπορεί να μην διορθώνονται παρά μόνον αν διακοπεί η παροχή ρεύματος στην πληγείσα μονάδα. Φαινόμενα όπως SEL μπορούν να απαιτούν τέτοιες δράσεις.
Μονάδα	Απομόνωση μονάδας	Αν μία μονάδα δεν μπορεί να επιδιορθωθεί, το επόμενο βήμα είναι να αγνοηθούν οι τιμές της ή να αποσυνδεθεί από το ρεύμα, ώστε να μην επηρεαστεί το υπόλοιπο σύστημα από τις λανθασμένες ενέργειές της.
Υποσύστημα	Επανεκκίνηση υποσυστήματος	Φαινόμενα ακτινοβολίας ή άλλες αποτυχίες μπορεί να οδηγήσουν ένα υποσύστημα σε μη αναμενόμενη κατάσταση, από την οποία να μπορεί να ξεφύγει μόνο με επανεκκίνηση.
Υποσύστημα	Αναδιαμόρφωση υποσυστήματος	Αν υπάρχει μια μόνιμη βλάβη που δεν μπορεί να διορθωθεί με επανεκκίνηση, ο δορυφόρος πρέπει να προσπαθήσει να συνεχίσει την κανονική λειτουργία. Αυτό μπορεί να γίνει αλλάζοντας τις προεπιλεγμένες ρυθμίσεις ενός υποσυστήματος, π.χ. μέσω επιλογής διαφορετικών αισθητήρων, μέσω αλλαγής της λειτουργίας του υποσυστήματος, ή μέσω ανάθεσης των καθηκόντων του σε μια άλλη λειτουργική μονάδα.
Σύστημα	Διακοπή τρέχουσας λειτουργίας	Μπορεί να διακοπεί η εκτελούμενη διαδικασία (π.χ. έναρξη ενός πειράματος, αποστολή TM), εφ' όσον δεν επηρεάζει την ασφάλεια του συστήματος.
Σύστημα	Είσοδος σε ασφαλή λειτουργία (safe mode)	Αυτό είναι το τελευταίο βήμα στις διορθωτικές προσπάθειες. Ο δορυφόρος εισέρχεται στην ασφαλή λειτουργία ώστε να αποτρέψει περαιτέρω διάδοση της βλάβης.

Πίνακας 4.4: Διορθωτικές δράσεις FDIR του AcubeSAT

4.4 Λειτουργική ροή του FDIR

Σε αυτήν την ενότητα παρουσιάζονται οι ροές και διαδικασίες του FDIR που εκτελούνται σε τροχιά. Αν και η διαδικασία για την κάθε δομική μονάδα και την κάθε διαδικασία είναι διαφορετική, ακολουθείται ένας γενικότερος ιεραρχικός κανόνας που ξεκινάει από τη χαμηλότερη βαθμίδα (μονάδας/εξαρτήματος), και προχωρά στο επίπεδο συστήματος, αν η εξεταζόμενη βλάβη συνεχίζει να το επηρεάζει.

Μετά από την επιδιόρθωση (ή προσπάθεια επιδιόρθωσης) της βλάβης, η ευθύνη μεταφέρεται πλέον στους χειριστές, που πρέπει να εντοπίσουν την βασική πηγή του προβλήματος, και να πραγματοποιήσουν τις απαραίτητες ενέργειες, αν χρειάζεται, για τη μόνιμη επιδιόρθωσή του.



Η γενική ροή που παρουσιάζεται στο Σχήμα 4.5 περιγράφει την τυπική διαδικασία επίλινσης μιας αποτυχίας στο διορυφόρο. Η διαδικασία που ακολουθείται είναι ιεραρχική, ξεκινάει από το χαμηλό επίπεδο, και μπορεί να καταλήξει μέχρι και το επίπεδο συστήματος. Πιο συγκεκριμένα:

1. Αν μια παρατηρούμενη παράμετρος είναι εκτός ορίων, πρώτα επιχειρείται επιδιόρθωση σε επίπεδο μονάδας με απλές ενέργειες (π.χ. reset, επανεκκίνηση).

Σχήμα 4.5: Γενική ροή του FDIR σε λειτουργικές καταστάσεις του διορυφόρου

2. Αν η αποτυχία συνεχίσει να υφίσταται, τότε γίνεται προσπάθεια επίλυσής της σε επίπεδο υποσυστήματος. Για κρίσιμα εξαρτήματα, αυτό μπορεί να σημαίνει επανεκκίνηση ολόκληρου του υποσυστήματος. διαφορετικά, μπορεί να γίνει απλή απενεργοποίηση της μονάδας.

Τεχνικά, ο διαχωρισμός μεταξύ των επιπέδων μονάδας και υποσυστήματος επιτυγχάνεται με κατάλληλη αξιοποίηση του **ST[12] on-board monitoring service** ([Ενότητα 3.1](#)). Με την παραμέτρου **repetition number** (αριθμός επαναλήψεων), ο χειριστής μπορεί να ορίσει το κατώφλι στη διάρκεια ύπαρξης της αποτυχίας που σηματοδοτεί απλή επιδιόρθωση ή βαρύτερα μέτρα. Για παράδειγμα, 2 μετρήσεις εκτός ορίων οδηγούν σε επανεκκίνηση του αισθητήρα, ενώ 5 μετρήσεις εκτός ορίων μπορεί να οδηγήσουν σε πλήρη απομόνωση/απενεργοποίησή του.

3. Αν η αποτυχία συνεχίσει να υφίσταται αλλά δεν είναι κρίσιμη, απενεργοποιείται η μονάδα μόνιμα, καθώς και οι έλεγχοι του FDIR για αυτήν, και αναδιαμορφώνεται αν χρειάζεται το υποσύστημα.

Σε αυτό το σημείο πρέπει να τονιστεί πως η απενεργοποιημένη μονάδα δεν μπορεί να συνεισφέρει σε οποιαδήποτε λειτουργία του δορυφόρου. Η υπηρεσία ST[12] επιτρέπει τη χρήση των **check validity conditions** (συνθηκών εγκυρότητας ελέγχου), που μπορούν να απενεργοποιήσουν έναν έλεγχο αν αυτός αναφέρεται σε απενεργοποιημένη συσκευή ή άλλες συνθήκες που τον αποτρέπουν.

4. Αν η αποτυχία συνεχίσει να υφίσταται, είναι κρίσιμη και δεν υπάρχει άλλος τρόπος διόρθωσής της, το σύστημα εισέρχεται σε **Ασφαλή Λειτουργία**.

Όλα τα παραπάνω εκτελούνται στο υπόβαθρο του κάθε μικροελεγκτή σαν μία διεργασία του RTOS, και, βασισμένα στις υπηρεσίες PUS, είναι πλήρως προσαρμόσιμα από τον σταθμό βάσης.

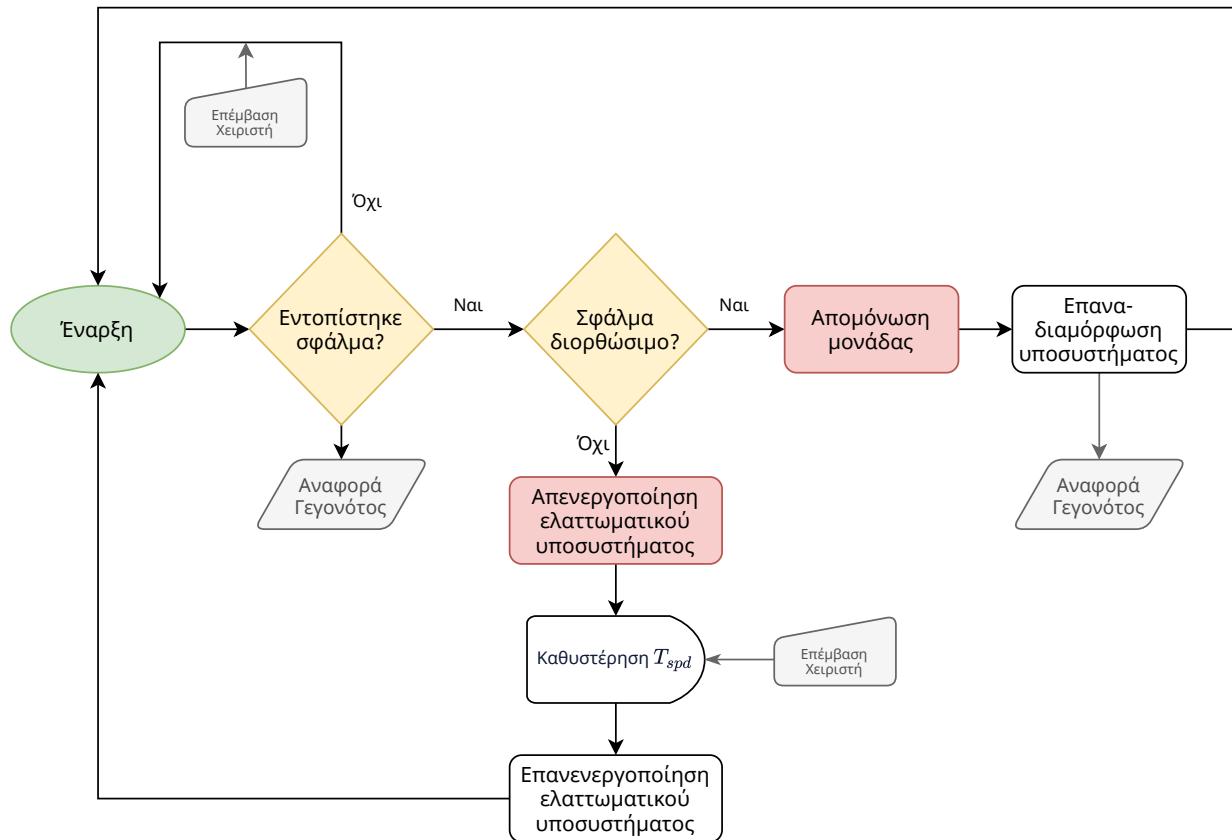
Μία εναλλακτική μέθοδος υλοποίησης είναι να οριστεί μια παράμετρος που δείχνει την κατάσταση ή το τρέχον βίντα του FDIR για την κάθε μονάδα, και οι ορισμοί παρακολουθίσης να πυροδοτούνται ανάλογα με την τιμή αυτής της παραμέτρου.

Υποσύστημα	Εκτόξευση	Εκκίνηση	Κανονική	Επιστημονική	Ασφαλής
ADCS	Off	Σταθεροποίηση	Στόχευση	Σταθεροποίηση	Σταθεροποίηση
COMMS	Off	UHF μόνο	UHF και S-Band	UHF μόνο	UHF μόνο
EPS	Off	On	On	On	On
OBC	Off	On	On	On	On
SU	Off	Off	Συντήρηση & δεδομένα	On	Συντήρηση

4.4.1 Ασφαλής Λειτουργία

Ο στόχος της ασφαλούς λειτουργίας είναι να παρέχει μια ισχυρή, καλά δοκιμασμένη και ντετερμινιστική κατάσταση, όπου ο δορυ-

πίνακας 4.5: Σύνοψη της λειτουργικότητας του AcubeSAT σε διαφορετικές Λειτουργίες Συστήματος



Σχήμα 4.6: Ροή του FDIR στην Ασφαλή Λειτουργία

φόρος μπορεί να λειτουργήσει αυτόνομα, μέχρι οι χειριστές μπορέσουν να αποκαταστήσουν την απειλητική για την αποστολή βλάβη. Το FDIR της Ασφαλούς Λειτουργίας ακολουθεί επίσης το πρότυπο PUS και τις προδιαγραφές του AcubeSAT ([Ενότητα 4.1](#)), συνεχίζοντας να επιτρέπει τις αλλαγές από τους χειριστές.

Στο AcubeSAT, η λειτουργικότητα του διορυφόρου είναι σημαντικά μειωμένη: το προφίλ στόχευσης προσπαθεί να πετύχει μόνο σταθεροποίηση, δεν εκτελούνται περίπλοκες λειτουργίες, και τα υποσυστήματα πραγματοποιούν μόνο συντήρηση. Ωστόσο, η επικοινωνία UHF και η μετάδοση του περιοδικού ραδιοφάρου (βεαζον) εξακολουθούν να είναι ενεργές για λόγους παρατηρησιμότητας, και για να επιτραπεί η διόρθωση του εντοπισμένου προβλήματος.

Η ΓΕΝΙΚΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ του FDIR για την ασφαλή λειτουργία παρουσιάζεται στο [Σχήμα 4.6](#). Οι ενέργειες αποκατάστασης περιλαμβάνουν:

- Απενεργοποίηση ολόκληρων προβληματικών υποσυστημάτων. Στην περίπτωση των ADCS και OBC, η ενέργεια αυτή συνοδεύεται από σημαντικά μειωμένη λειτουργικότητα, αλλά δεν επηρεάζει τη δυνατότητα ελέγχου του διορυφόρου. Από την

άλλη πλευρά, η απενεργοποίηση των υποσυστημάτων COMMS και EPS εμποδίζει οποιαδήποτε επικοινωνία του σταθμού βάσης με το CubeSat. Η μόνη δράση που μπορεί να γίνει είναι η προσπάθεια επανεκκίνησης των υποσυστημάτων μετά από μία προκαθορισμένη χρονική καθυστέρηση, μέχρι ο δορυφόρος να βρεθεί για παράδειγμα σε πιο ευνοϊκές περιβαλλοντικές συνθήκες.

- Απομόνωση εξαρτημάτων και ανάλογη αναδιαιρόφωση των υποσυστημάτων. Ο μικρότερος αριθμός ενεργών εξαρτημάτων σε Ασφαλή Λειτουργία σημαίνει μια πιο ασφαλή διαδικασία σχεδιασμού και δοκιμών.

Οι χειριστές του AcubeSAT είναι υπεύθυνοι για τον προσδιορισμό της αιτίας της βλάβης και των διορθωτικών ενεργειών αμέσως μόλις λάβουν την τηλεμετρία που ανακοινώνει το σφάλμα, και πρέπει να επέμβουν άμεσα αφού αποφασίστεί μια ασφαλής πορεία δράσης. Η έξοδος από την ασφαλή λειτουργία μπορεί να γίνει μόνο μετά από εντολή του χειριστή, και μόνο αφού διορθωθεί το αρχικό σφάλμα και δεν είναι ορατά άλλα σφάλματα (λόγω π.χ. διάδοσης της βλάβης). Οι δράσεις που μπορούν να λάβουν οι χειριστές είναι:

- Άλλαγή των ρυθμίσεων του δορυφόρου μέσω των υπηρεσιών ECSS
- Πιο συχνά, άλλαγή του **λογισμικού** του δορυφόρου, ανεβάζοντας ανανεωμένες εικόνες δυαδικής μορφής²
- Μόνιμη απενεργοποίηση ελαττωματικών μονάδων
- Αναμονή μέχρι ο δορυφόρος να βρεθεί σε ευνοϊκή κατάσταση
- Καμία, αν η βλάβη εντοπίστηκε από λάθος, ή διορθώθηκε αυτόματα

² Η διαδικασία αυτή απαιτεί την εκτέλεση πλήρους καμπάνιας δοκιμών λογισμικού στη γη, και μία αξιόπιστη διαδικασία επαναπρογραμματισμού του δορυφόρου σε τροχιά [29, σ. 45]

4.5 Επόμενα βήματα

Ο συνδυασμός FMEA και HSIA του AcubeSAT περιέχει 928 πιθανές αποτυχίες, εκ των οποίων οι 185 θεωρούνται υψηλής κρισιμότητας [47].

Μετά από τη διατύπωση της αρχιτεκτονικής και του σχεδιασμού του FDIR, έπειτα η υλοποίησή του. Για αυτόν τον σκοπό, η ομάδα έχει αναπτύξει μία βιβλιοθήκη που υλοποιεί τις υπηρεσίες PUS³ και μπορεί να καλύψει τις ανάγκες της παρούσας ενότητας.

Η επαλήθευση του FDIR είναι το τελευταίο κομμάτι της διαδικασίας, και περιλαμβάνει έλεγχο της κάθε επίφοβης αποτυχίας, μέσω προσομοίωσης με υλικό ή λογισμικό.

[47] Retselis and Kanavouras, *AcubeSAT FMEA Worksheet*.

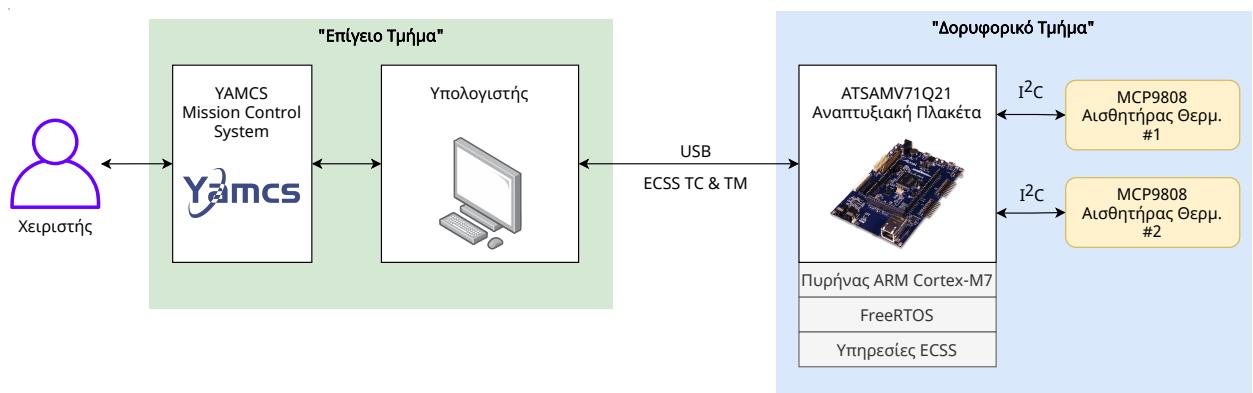
³ <https://gitlab.com/acubesat/obc/ecss-services>

5

Πρακτική επίδειξη του FDIR

Ως πείραμα "proof-of-concept" για την υλοποίηση του FDIR του AcubeSAT, προετοιμάσαμε μια πρακτική διάταξη για την προσομοίωση της συμπεριφοράς του δορυφόρου (Σχήμα 5.1). Η διάταξη αποτελείται από:

- Έναν Cortex-M7 μικροελεγκτή, που χρησιμοποιείται για την προσομοίωση ενός υποσυστήματος δορυφόρου
- Έναν αριθμός πλεονάζοντων αισθητήρων θερμοκρασίας, που προσομοιώνουν πιθανά σημεία αστοχίας.
- Το συνοδευτικό λογισμικό που περιλαμβάνει την υλοποίηση των υπηρεσιών ECSS και της λογικής SAVOIR FDIR.
- Έναν υπολογιστή, που χρησιμεύει ως σταθμός βάσης για την παροχή των απαραίτητων δυνατοτήτων ελέγχου και παρατήρησης.



5.1 Περιγραφή Συστήματος

Σχήμα 5.1: Μπλοκ διάγραμμα υψηλού επιπέδου της πειραματικής διάταξης

5.1.1 Λειτουργικότητα

Προκειμένου να εξομιλούσαμε τις πιο βασικές λειτουργίες ενός υποσυστήματος διαστημικού σκάφους, υλοποίήσαμε ένα σύστημα με

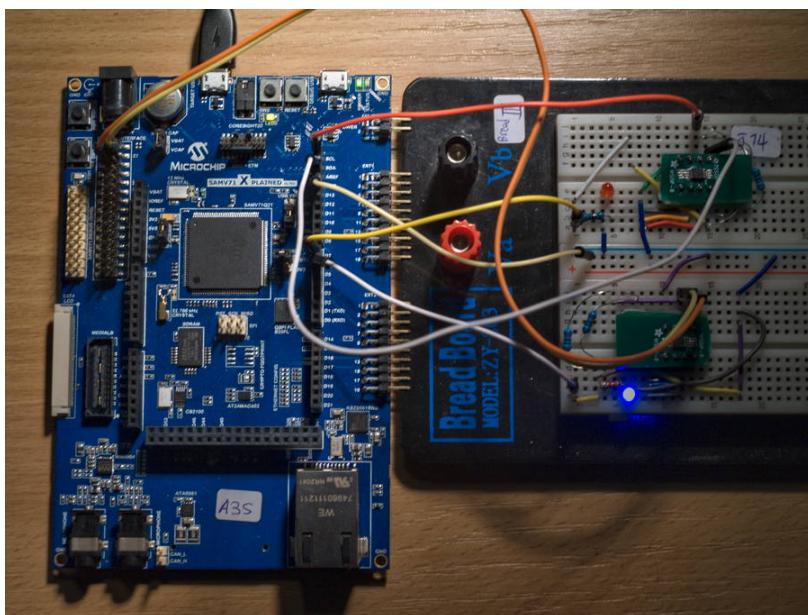
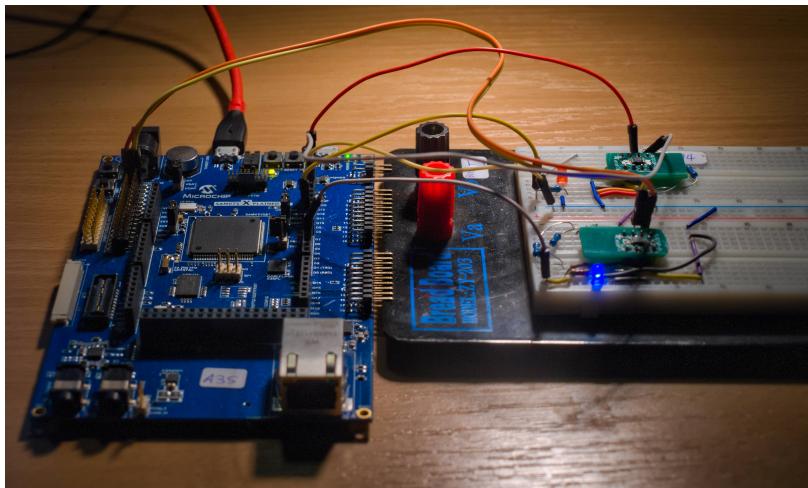
μία μόνο λειτουργική προδιαγραφή:

RQ-010: Το σύστημα πρέπει να μετρά και να μεταδίδει τη θερμοκρασία περιβάλλοντος.

Προκειμένου να δικαιολογήσουμε μια υλοποίηση FDIR για αυτό το σύστημα, θα εισάγουμε μια προδιαγραφή αξιοπιστίας:

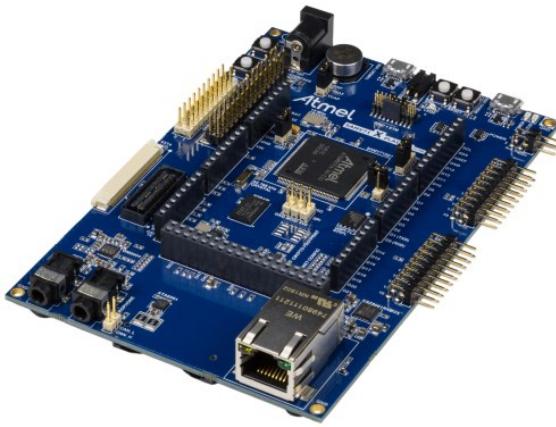
RQ-020: Καμία μεμονωμένη αποτυχία σε κάποιο μετρητικό στοιχείο δεν πρέπει να οδηγήσει σε απώλεια της λειτουργικότητας του συστήματος

Ο λεπτομερής σχεδιασμός αυτού του απλού συστήματος επίδειξης παρουσιάζεται στις επόμενες ενότητες, και καταστρώθηκε έτσι ώστε να ταιριάζει όσο το δυνατόν περισσότερο με τη λειτουργικότητα, τις διεπαφές, το σχεδιασμό και το λογισμικό του νανοδορυφόρου AcubeSAT.



Σχήμα 5.2: Το πλήρες δοκιμαστικό σύστημα. Αριστερά, διακρίνεται η μπλε αναπτυξιακή πλακέτα του μικροελεγκτή. Δεξιά βρίσκεται το breadboard, όπου οι είναι συνδεδεμένοι οι δύο αισθητήρες θερμοκρασίες. Διακρίνεται ακόμα η καλωδίωση από την πλακέτα προς το breadboard. Οι δύο ενδεικτικές λυχνίες δίνουν μία οπτική ένδειξη του αν το εξάρτημα λαμβάνει ισχύ ή όχι.

5.1.2 Υλικό



Το κέντρο του συστήματος επίδειξης είναι ο microcontroller (Μικροελεγκτής, MCU) που χρησιμοποιείται για την προσομοίωση του σχεδιασμού και της λειτουργικότητας ενός από τα υποσυστήματα του AcubeSAT (Ενότητα 2.2.4). Ο επιλεγμένος μικροελεγκτής είναι ο Atmel ATSAMV71Q21 που φιλοξενείται στην αναπτυξιακή πλακέτα **ATSAMV71-XULT**¹. Αυτός ο μικροελεγκτής είναι λειτουργικά πανομοιότυπος με αυτόν που θα χρησιμοποιηθεί στην τροχιά, διαθέτοντας έναν 32-bit πυρήνα ARM Cortex-M7 με μνήμες 2 MiB flash και 384 KiB SRAM, και μέγιστη ταχύτητα ρολογιού 300 MHz.

Ο μικροελεγκτής συνδέεται με δύο **αισθητήρες θερμοκρασίας** οι οποίοι χρησιμοποιούνται για την προσομοίωση εξαρτημάτων του υποσυστήματος που είναι επιφερεπή σε βλάβες. Οι επιλεγμένοι αισθητήρες είναι οι **MCP9808**² της Microchip, οι οποίοι προσφέρουν ακριβή και ταχεία ανάγνωση της θερμοκρασίας μέσω ενός διαύλου Inter-Integrated Circuit (I^2C). Η μέγιστη δυνατή περίοδος δειγματοληψίας για τον αισθητήρα είναι 250 ms.

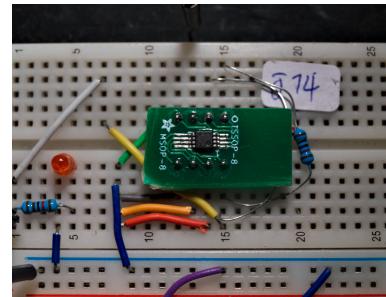
Οι δύο αισθητήρες είναι συνδεδεμένοι σε διάταξη ενεργού πλεονασμού, λόγω της εξαιρετικά χαμηλής ισχύος λειτουργίας τους. Οι δύο αισθητήρες είναι συνδεδεμένοι σε διαφορετικούς διαύλους I^2C , έτσι ώστε πιθανή βλάβη του ενός διαύλου να μην επηρεάζει τη λειτουργία του άλλου αισθητήρα.

Διεπαφή USB Για τη λήψη Telemetry και τη μετάδοση Telecommands στο υποσύστημα επίδειξης, ο σχεδιασμός περιλαμβάνει μια σύνδεση Universal Serial Bus με έναν τυπικό υπολογιστή. Η σύνδεση αυτή χρησιμοποιεί το περιφερειακό UART του μικροελεγκτή, το οποίο μεταφέρει αποκλειστικά μηνύματα ECSS μεταξύ του μικροελεγκτή και του υπολογιστή. Τα μηνύματα αυτά περιλαμβάνουν τις τυπικές αναφορές TM και τα αιτήματα TC, αλλά και μηνύματα καταγραφής που προορίζονται για διαγνωστικούς σκοπούς.

Καθώς το πρωτόκολλο UART δεν προσφέρει κάποιο μηχανισμό για διαχωρισμό πακέτων, εφαρμόζουμε κωδικοποίηση COBS [48] για όλα τα μεταδόμενα μηνύματα.

Σχήμα 5.3: Επίσημη φωτογραφία της αναπτυξιακής πλακέτας ATSAMV71-XULT

¹ <https://www.microchip.com/Developmenttools/ProductDetails/ATSAMV71-XULT>



Σχήμα 5.4: Ο αισθητήρας θερμοκρασίας MCP9808, κολλημένος επάνω στην πλακέτα του πειράματος

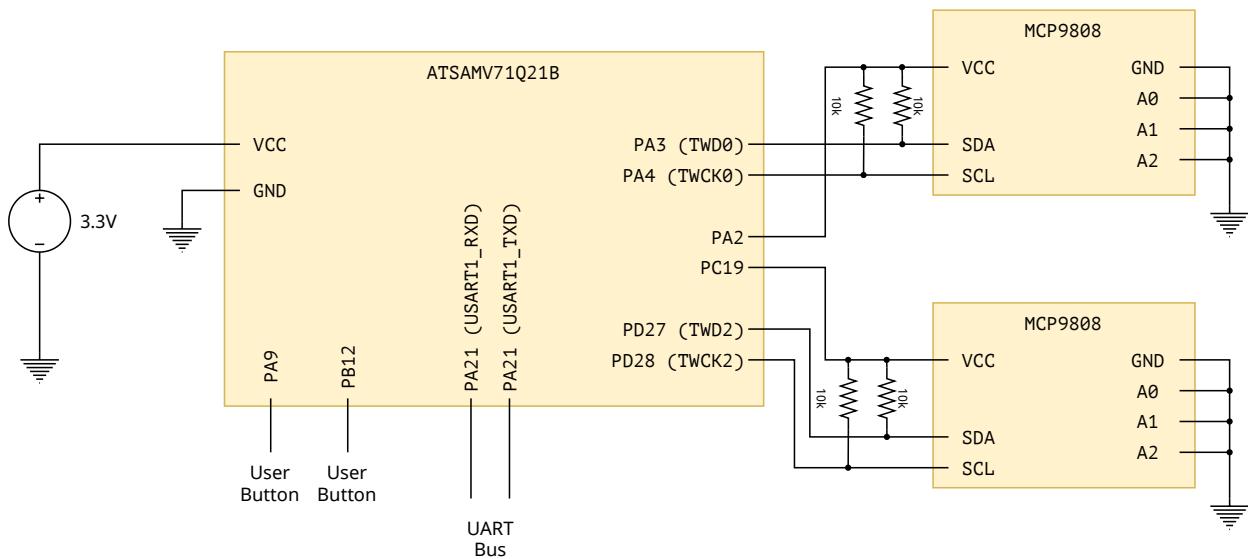
² <https://www.microchip.com/wwwproducts/en/MCP9808>

Ψυχρός πλεονασμός (cold redundancy): Μόνο το ένα εξάρτημα λειτουργεί πλήρως, ενώ τα άλλα έχουν μειωμένη λειτουργικότητα.

Θερμός πλεονασμός (warm redundancy): Ένα εξάρτημα λειτουργεί πλήρως, ενώ τα άλλα έχουν μειωμένη λειτουργικότητα.

Ενεργός πλεονασμός (hot redundancy): Δύο ή περισσότερα ταυτόχρονα ενεργά εξάρτηματα λειτουργούν παράλληλα [36].

[48] Cheshire and Baker, "Consistent Overhead Byte Stuffing".



Συνδέσεις Η πλακέτα ανάπτυξης και οι αισθητήρες τοποθετήθηκαν και συνδέθηκαν χρησιμοποιώντας ένα breadboard. Τα ολοκληρωμένα κυκλώματα κολλήθηκαν επάνω σε απλές πλακέτες μικρού μεγέθους, που κατασκευάστηκαν με βάση το σχέδιο του Jaroslav Sýkora.³

5.2 Λογισμικό

5.2.1 Δορυφορικό Τμήμα

Όλες οι λειτουργίες του FDIR αναπτύχθηκαν χρησιμοποιώντας τη γλώσσα C++.

Επιλέχθηκε το μοντέρνο πρότυπο C++17. Ολόκληρη η εργασία βασίζεται σε ελεύθερο και ανοικτό λογισμικό το οποίο διατίθεται ελεύθερα για λύψη και τροποποίηση. Όλος ο κώδικας αναπτύχθηκε σε Git⁴.

Η λειτουργικότητα του FDIR είναι δομημένη γύρω από την υλοποίηση του ECSS-E-ST-70-41C PUS που γράφτηκε από την ομάδα AcubeSAT⁵, η οποία προσφέρει μια παραμετροποιήσιμη υλοποίηση του προτύπου χρησιμοποιώντας σύγχρονη C++.

Το λογισμικό του μικροελεγκτή και η βασική λογική βασίζονται στο FreeRTOS⁶, ένα λειτουργικό σύστημα πραγματικού χρόνου με χαμηλές απαιτήσεις που απευθύνεται σε ενσωματωμένες συσκευές. Το FreeRTOS παρέχει ανεξάρτητες διεργασίες και δομές που επιτρέπουν τον ασφαλή συγχρονισμό δεδομένων μεταξύ παράλληλων παραγωγών & καταναλωτών.

Για να διασφαλιστεί υψηλή αξιοπιστία και χαμηλή κατανάλωση

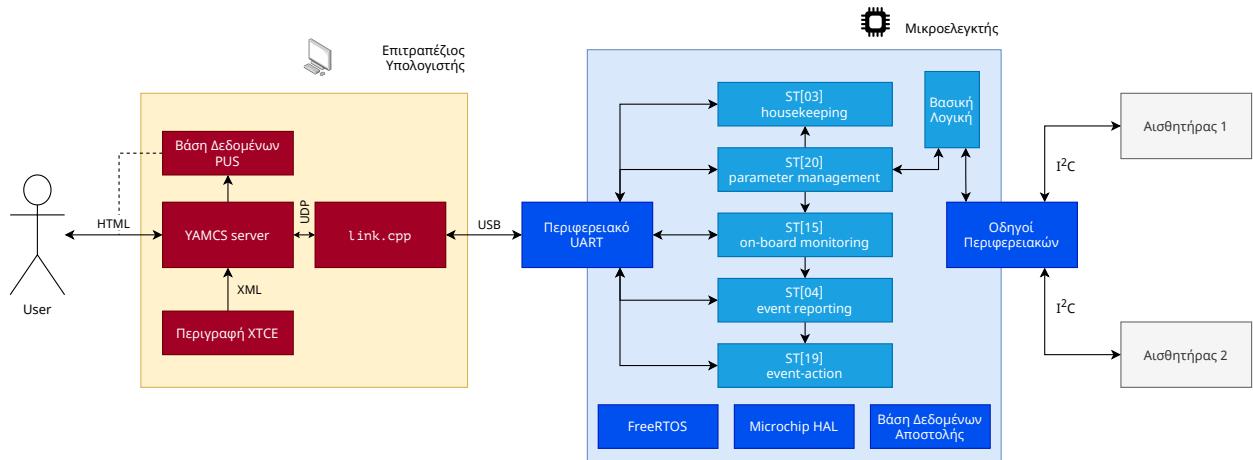
Σχήμα 5.5: Απλοποιημένο σχηματικό της εφαρμογής. Οι δύο αισθητήρες είναι συνδεδεμένοι μέσω I²C με τις απαραίτητες αντιστάσεις ανύψωσης στα 3.3 V. Η τροφοδοσία τους ελέγχεται απευθείας από τις εξόδους του μικροελεγκτή, επιτρέποντας στο χρήστη να τους αποσυνδέσει εντελώς αν χρειαστεί. Οι ακροδέκτες A των αισθητήρων ωριμάζονται αυθαίρετα ώστε να επιλέγεται η διεύθυνση I²C της κάθε συσκευής. Χρησιμοποιήσαμε την ίδια διεύθυνση, καθώς οι δύο δίαυλοι I²C είναι πλήρως ανεξάρτητοι. Πυκνωτές αποσύζευξης, κρύσταλλοι και άλλα παθητικά εξαρτήματα δεν φαίνονται, καθώς περιλαμβάνονται ήδη στην έτοιμη αναπτυξιακή πλακέτα.

³ <https://bit.ly/3vZwuH1>

⁴ <https://git-scm.com/>

⁵ <https://gitlab.com/acubesat/obc/ecss-services>

⁶ <https://www.freertos.org/>



πόρων, οι ακόλουθοι **περιορισμοί** λαμβάνονται υπ' όψιν την ανάπτυξη σε C++:

1. Η δυναμική κατανομή μνήμης⁷ απαγορεύεται εντελώς.
2. Το **Στοιχείο 1** σημαίνει ότι οι κλασικές δομές δεδομένων της C++⁸ δεν μπορούν να χρησιμοποιηθούν. Αντίθετα, η βιβλιοθήκη **Embedded Template Library (ETL)**⁹ περιλαμβάνει δομές που αποθηκεύονται σε στατική μνήμη, ή στην στοίβα.
3. "Ακριβά" χαρακτηριστικά, όπως το *Run-Time Type Inference*, οι δυναμικές μετατροπές τύπων (dynamic casts) ή οι εξαιρέσεις (exceptions) απαγορεύονται επίσης.

Η βιβλιοθήκη Hardware Abstraction Library (HAL) και η σουίτα λογισμικού **MPLAB Harmony**¹⁰ χρησιμοποιούνται για την επικοινωνία με όλα τα ενσωματωμένα περιφερειακά του μικροελεγκτή. Το **CLion**¹¹ έχει επιλεγεί ως το IDE, μαζί με το σύστημα "χτισίματος" κώδικα **CMake**¹².

Όλο το λογισμικό που αναπτύχθηκε στο πλαίσιο της παρούσας διατριβής είναι διαθέσιμο στο διαδίκτυο και έχει καταγραφεί στον Πίνακα 5.2. Τα πιο κρίσιμα μέρη του λογισμικού που αναπτύχθηκε έχουν αναταραχθεί στο **Παράρτημα B**. Μια λίστα των έτοιμων ή τροποποιημένων βιβλιοθηκών που χρησιμοποιήθηκαν μπορεί επίσης να βρεθεί στον Πίνακα 5.3.

Σχήμα 5.6: Η φοί δεδομένων μεταξύ των διαφορετικών κομματιών του λογισμικού

⁷ Χρήση των `malloc`, `new` κλπ.

⁸ `std::vector`, `std::string`, `std::map` κλπ.

⁹ <https://www.ETLcpp.com/>

¹⁰ <https://www.microchip.com/en-us/development-tools-tools-and-software/embedded-software-center/mplab-harmony-v3>

¹¹ <https://www.jetbrains.com/clion/>

¹² <https://cmake.org/>

Όνομα	Μέγεθος στοίβας	Επεξήγηση
Internal_Temp	2.5 kB	Λαμβάνει διαγνωστικές μετρήσεις από τον εσωτερικό αισθητήρα θερμοκρασίας του μικροελεγκτή, και αναβοσβίνει ένα LED για σκοπούς παρατηρησιών.
ECSS	3 kB	Εκτελεί τις περιοδικές διαδικασίες που ορίζουν οι υπηρεσίες ECSS. Αυτό περιλαμβάνει την εκπομπή ραδιοφάρων και τον έλεγχο των ορισμένων παρακολούθησης. Αυτή η διεργασία υλοποιεί το FDIR.
UART_Tx	3 kB	Αποστολή μηνυμάτων προς τον υπολογιστή. Η διεργασία αυτή λειτουργεί ως "gatekeeper task", δηλαδή είναι η μόνη διεργασία που μπορεί να στείλει μηνύματα UART. Όλες οι άλλες διεργασίες επικοινωνούν με αυτήν μέσω μίας στοίβας.
UART_Rx	6 kB	Λήψη μηνυμάτων UART από τον υπολογιστή. Για απλότητα, αυτή η διεργασία, πέρα από λήψη τηλεεντολών, πραγματοποιεί και την άμεση εκτέλεσή τους.
T1	1.5 kB	Επικοινωνία με τον αισθητήρα θερμοκρασίας 1, και ενημέρωση των αντίστοιχων παραμέτρων
T2	1.5 kB	Επικοινωνία με τον αισθητήρα θερμοκρασίας 2, και ενημέρωση των αντίστοιχων παραμέτρων

Πίνακας 5.1: Λίστα διεργασιών του FreeRTOS που υλοποιούν την πειραματική διάταξη

Όνομα	Περιγραφή	URL
fdir-demo	Πλήρης κώδικας για τον μικροελεγκτή	https://github.com/kongr45open/fdir-demo
fdir-demo-yamcs	Λογισμικό Σταθμού Βάσης: 1. Διεπαφή με το YAMCS 2. Περιγραφή ECSS πακέτων, αναγνώσιμη από μηχανή 3. Διεπαφή UART-USB με τον μικροελεγκτή 4. Πρόγραμμα εμφάνισης της βάσης δεδομένων PUS	https://github.com/kongr45open/fdir-demo-yamcs

Πίνακας 5.2: Λίστα του νέου λογισμικού που αναπτύχθηκε για την παρούσα εργασία

Όνομα	Περιγραφή	Μεταβολές
FreeRTOS	Λειτουργικό σύστημα πραγματικού χρόνου	
ecss-services	C++ υλοποίηση του προτύπου ECSS-E-ST-70-41C	Προστέθηκαν υλοποίησεις υπηρεσιών που δεν υπήρχαν, και βελτιώθηκαν συναρτήσεις διεπαφής με τον μικροελεγκτή https://gitlab.com/kongr45gpen/ecss-services/-/tree/fdir
ETL	Βιβλιοθήκη C++ (περιλαμβάνει δομές δεδομένων, αλγορίθμους και άλλα εξαρτήματα) για κρίσιμες εφαρμογές με περιορισμένη μνήμη	
YAMCS	Σουίτα λογισμικού για έλεγχο διαστημικών αποστολών	
cobs-c	Υλοποίηση του πρωτοκόλλου COBS	
Material Design Lite	Σουίτα έτοιμων σχεδιαστικών στοιχείων σε γλώσσα CSS (για σταθμό βάσης)	
lodash	Βοηθητικές συναρτήσεις σε Javascript (για σταθμό βάσης)	

Πίνακας 5.3: Λίστα των έτοιμων βιβλιοθηκών που χρησιμοποιήθηκαν

5.2.2 Μέγεθος Λογισμικού

Καθώς αναπτύσσουμε ένα ενσωματωμένο σύστημα σε μικροελεγκτή με περιορισμούς χώρου, έχει νόημα η ανάλυση του μεγέθους του κώδικα που παράγεται. Ο μεταγλωττιστής gcc παρέχει τη δυνατότητα μέτρησης των καταναλισκόμενων πόρων. Το συνολικό μέγεθος του προγράμματος παρουσιάζεται στον [Πίνακα 5.4](#), και τα αρχεία C++ με την μεγαλύτερη κατανάλωση μνήμης λογισμικού και μνήμης δεδομένων φαίνονται στους [Πίνακες 5.5](#) και [5.6](#).¹³

Κώδικας χωρίς βελτιστοποίηση (-O0)

Περιοχή μνήμης	Χρονισμοποιημένη	Συνολικό μέγεθος	Ποσοστό
ROM	254 324 B	2 MB	12.13%
RAM	205 992 B	384 kB	52.39%

Βελτιστοποίηση ελάχιστου μεγέθους (-Ofast)

Περιοχή μνήμης	Χρονισμοποιημένη	Συνολικό μέγεθος	Ποσοστό
ROM	228 384 B	2 MB	10.89%
RAM	205 992 B	384 kB	52.39%

Μέγεθος (B) Αρχείο

22 515	ServicePool.cpp
12 350	main.cpp
9271	TimeBasedSchedulingService.cpp
8684	SystemParameters.cpp
7708	EventActionService.cpp
6546	OnBoardMonitoringService.cpp
5952	ECSSTask.cpp
4451	FreeRTOS_tasks.c
4404	MCP9808-internal.cpp
3281	Logger.cpp

Μέγεθος (B) Αρχείο

131104	heap_4.c
38 088	ServicePool.cpp
11192	UARTTask.cpp
1064	SystemParameterMonitoring.cpp
340	UARTRXTask.cpp
260	FreeRTOS_tasks.c
200	sys_time.c
188	SystemParameters.cpp
50	main.cpp

Να σημειωθεί πως δεν έχει γίνει καμία προσπάθεια για βελτιστοποίηση της κατανάλωσης μνήμης ή γενικότερα των επιδόσεων του συστήματος. Επίσης, να τονιστεί πως λόγω της χρήσης της βιβλιοθήκης ETL και της έλλειψης δυναμικής κατανάλωσης μνήμης,

¹³ Τα δεδομένα παράχθησαν με χρήση της βιβλιοθήκης [LinkerMapView](#).

Πίνακας 5.4: Μέγεθος του παραχθέντος κώδικα με διαφορετικές ρυθμίσεις μεταγλωττισή

Πίνακας 5.5: Τα αρχεία που καταναλώνουν την περισσότερη μνήμη ROM (δεν περιλαμβάνονται βιβλιοθήκες συστήματος)

Πίνακας 5.6: Τα αρχεία που καταναλώνουν την περισσότερη μνήμη RAM (δεν περιλαμβάνονται βιβλιοθήκες συστήματος)

οι καταναλώσεις του **Πίνακα 5.6** αναφέρονται στην πλειοψηφία σχεδόν της κατανάλωσης μνήμης RAM. Μάλιστα, η στοίβα (stack) όπου αρχικοποιούν μεταβλητές οι διεργασίες του FreeRTOS, είναι καταχωριμένη εξ' ολοκλήρου στο αρχείο `heap_4.c`.

5.2.3 Εντοπισμός Αποτυχιών I^2C

Έχει ενδιαφέρον να μελετήσουμε πώς το περιφερειακό I^2C του μικροελεγκτή μπορεί να χρησιμοποιηθεί για τον εντοπισμό διαφόρων ειδών σφαλμάτων στον δίαυλο και τα περιφερειακά:

1. Σφάλματα περιφερειακού

Το περιφερειακό I^2C του μικροελεγκτή είναι ικανό να παράγει σφάλμα κάθε φορά που το συνδεδεμένο περιφερειακό δεν θέτει το bit "acknowledge". Η αποτυχία να θέσει αυτό το bit μπορεί να είναι αποτέλεσμα αποσύνδεσης των γραμμών ρολογιού (SCL) ή δεδομένων (SDA), βλάβης του διαύλου ή κακής λειτουργίας του ίδιου του περιφερειακού.

2. Χρονικό περιθώριο απάντησης

Σε ορισμένες δυσμενείς για το I^2C πλεκτρικές συνθήκες, που μπορεί να προκληθούν από λανθασμένα επιλεγμένες τιμές αντίστασης pull-up ή μεγάλο χωροπικό φορτίο, η ακεραιότητα του σήματος I^2C μπορεί να χαθεί (**Σχήμα 5.7**). Αυτό το σφάλμα δεν ανιχνεύεται άμεσα από το περιφερειακό του μικροελεγκτή, αλλά μπορεί να διαγνωστεί με την προσθήκη ενός χρονικού ορίου (timeout) στις αναγνώσεις δεδομένων του αισθητήρα.

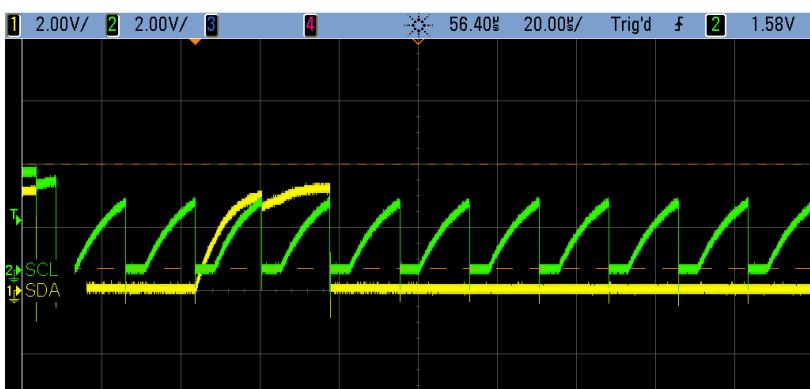
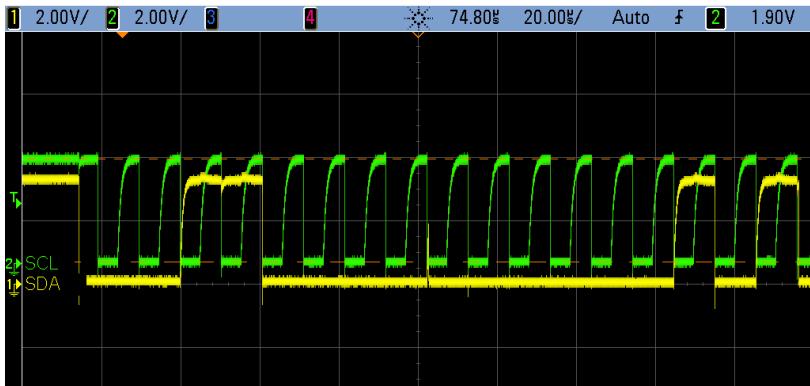
3. Chip ID

Είναι πιθανό ο αισθητήρας να υποστεί σοβαρή βλάβη, αλλά να είναι ακόμα σε θέση να ανταποκριθεί στις εντολές I^2C ή να θέσει το bit *acknowledge*. Για τον λόγο αυτό, πριν από κάθε ανάγνωση της θερμοκρασίας, πραγματοποιείται ένας πιο αυστηρός έλεγχος της κατάστασης του αισθητήρα: οι μόνιμοι καταχωριτές του περιφερειακού περιέχουν τιμές οι οποίες, υπό κανονικές συνθήκες, δεν μπορούν να αλλάξουν εκτός αν υπάρχει βλάβη υλικού. Εάν επιστραφεί οποιαδήποτε τιμή διαφορετική από την αναμενόμενη, θεωρείται ότι το περιφερειακό ή ο δίαυλος έχει εμφανίσει σφάλμα.

Για να εξασφαλιστεί ο εντοπισμός των σφαλμάτων του περιφερειακού, είναι σημαντικό να βεβαιωθούμε ότι η κατάσταση σφάλματος του περιφερειακού αναγνωρίζεται μετά από κάθε επικοινωνία. Τα interrupts του μικροελεγκτή και η εντολή `[[nodiscard]]` της C++ συνδυάζονται για να διασφαλιστεί ότι όλα τα σφάλματα του υλικού λαμβάνονται υπόψιν.

Πίνακας 5.7: Μη εγγράψιμοι καταχωριτές του MCP9808

Διεύ/ση	Καταχωριτής	Τιμή
0x06	Manufacturer ID	0x0054
0x07	Device ID	0x0400



5.2.4 Έλεγχος Αποστολής

Ο υπολογιστής του επίγειου τμήματος πρέπει να αναλύει τα λαμβανόμενα πακέτα και να στέλνει εντολές, καθώς και να εμφανίζει πληροφορίες σχετικά με τις μετρήσεις των αισθητήρων, την κατάσταση του FDIR, και την υγεία του συνδεδεμένου συστήματος πτήσης.

Η σουίτα λογισμικού YAMCS [49] επιλέχθηκε για να καλύψει τις παραπάνω ανάγκες, και προσαρμόστηκε ώστε να παρέχει τις δυνατότητες που απαιτούνται για την παρούσα επίδειξη.

Το πρωτόκολλο ECSS [35] δεν υποστηρίζεται εγγενώς από το YAMCS. Ωστόσο, οι απαραίτητες εντολές έχουν εισαχθεί στο σύστημα μέσω αρχείων μορφής XML Telemetric and Command Exchange (XTCE) [50]. Ο πιγαίος κώδικας της ενσωμάτωσης μπορεί να βρεθεί στο Παράρτημα Β.

5.2.5 Βάση Δεδομένων PUS

Η σουίτα YAMCS δεν διαθέτει έτοιμη υποστήριξη και συγχρονισμό με τη βάση δεδομένων PUS του δορυφόρου. Αυτό σημαίνει πως δεν υπάρχει έτοιμος εύκολος τρόπος για εποπτική προβολή των γεγονότων, παραμέτρων, ορισμών παρακολούθησης και ορισμών γεγονότος-αντίδρασης.

Σχήμα 5.7: Καθαρό σήμα I²C. Η πράσινη κυματομορφή (Serial Clock (SCL)) έχει σχήμα όπου μπορούν εύκολα να ξεχωριστούν οι παλμοί, και τα δεδομένα της κίτρινης κυματομορφής (Serial Data (SDA)) συνδέονται αντίστοιχα με τους παλμούς ρολογιού. Το πλάτος των κυματομορφών είναι περίπου 3.3 V

Σχήμα 5.8: Σήμα σε ένα ελαττωματικό δίκτυο I²C, όπου υπάρχει πολύ μεγάλο χωροποικό φορτίο. Ο χρόνος ανόδου και των δύο σημάτων είναι υπερβολικά μεγάλος, και η επικοινωνία στη συγκεκριμένη συχνότητα δεν είναι δυνατή. Το σήμα δεν προλαβαίνει να φτάσει την τιμή των 3.3 V. Σε αυτήν την περίπτωση, δεν είναι δυνατή η μεταφορά δεδομένων στον δίαυλο, και το δίκτυο ή το εξάρτημα θεωρείται πως έχει αποτύχει.

[49] Sela, “Yamcs - A Lightweight Open-Source Mission Control System”.

[35] ECSS Secretariat, *ECSS-E-ST-70-41C - Telemetry and Telecommand Packet Utilization*.

[50] Simon et al., “XTCE: A Standard XML-Schema for Describing Mission Operations Databases”.

The PUS Database Viewer table shows the following PMON definitions:

PMON ID	Monitored Parameter	Validity condition	Interval	PMON Status	Repetition #	Check	Acquisition time
0	Temp. 1 Status	Temp. 1 0 <= 0x0	500 ms	OK	2	x = Nominal 0x2 Temp1_Timeout_Temporary	2021-06-25T16:09:06.453Z
1	Temp. 2 Status	Temp. 1 0 <= 0x0	500 ms	OK	2	x = Nominal 0x2 Temp2_Timeout_Temporary	2021-06-25T16:09:06.461Z
2	Temp. 1 Status	Temp. 1 0 <= 0x0	500 ms	OK	5	x = Nominal 0x2 Temp1_Timeout_Permanent	2021-06-25T16:09:06.468Z
3	Temp. 2 Status	Temp. 1 0 <= 0x0	500 ms	OK	5	x = Nominal 0x2 Temp2_Timeout_Permanent	2021-06-25T16:09:06.476Z
4	Temp. 1	Temp. 1 Status 1 <= 0xff	500 ms	OK	2	-40 <= x <= 100 Temp1_UnexpectedValue_Temporary Temp1_UnexpectedValue_Temporary	2021-06-25T16:09:06.484Z
5	Temp. 2	Temp. 2 Status 1 <= 0xff	500 ms	OK	2	-40 <= x <= 100 Temp2_UnexpectedValue_Temporary Temp2_UnexpectedValue_Temporary	2021-06-25T16:09:06.491Z
6	Temp. Δ	Temp. 1+2 Status 1 <= 0xff	500 ms	OK	6	-5 ≤ x ≤ 5 TempDelta_UnexpectedValue_Temporary TempDelta_UnexpectedValue_Temporary	2021-06-25T16:09:06.499Z

Για να διευκολυνθεί η γρήγορη προβολή της διαδικασίας του FDIR, υλοποιήσαμε μία online διεπαφή που δείχνει τα παραπάνω (Σχήμα 5.10). Η διεπαφή ενημερώνεται σε πραγματικό χρόνο μέσω του YAMCS HTTP API και προβάλλει τους ορισμούς αλλά και την κατάσταση του FDIR.

Η εφαρμογή γράφτηκε σε HTML και Javascript, εκμεταλλευόμενη WebSockets, και το γραφικό περιβάλλον της βασίζεται στην βιβλιοθήκη Material Design Lite.¹⁴ Βρίσκεται στο αποθετήριο fdir-demo-yamcs¹⁵, μαζί με τα υπόλοιπα κομμάτια του σταθμού βάσης της επίδειξης.

5.3 Στήσιμο FDIR

Ο σκοπός της πειραματικής διάταξης είναι να παρατηρηθεί η α-

Σχήμα 5.9: Προβολές παραμέτρων και αρχείου στο YAMCS

Σχήμα 5.10: Η διεπαφή "PUS database viewer". Αριστερά βρίσκεται η λίστα ορισμών παρακολούθησης, πάνω δεξιά η λίστα μεταβάσεων, και κάτω δεξιά η λίστα ορισμών γεγονότος-αντίδρασης.

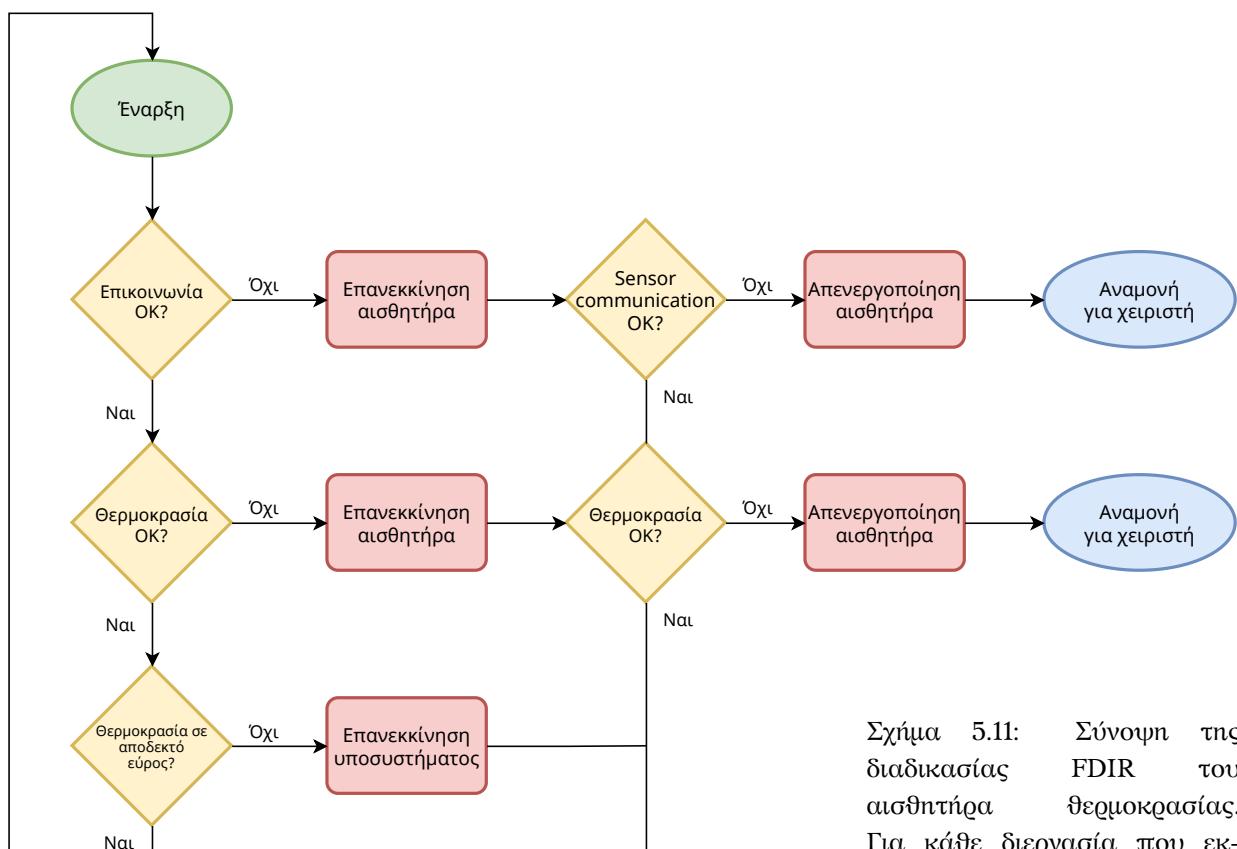
¹⁴ <https://getmdl.io/>

¹⁵ <https://github.com/kongr45open/fdir-demo-yamcs/tree/master/ecss-interface>

πόκριση του συστήματος σε τυχόν αποτυχία των δύο "ευπρόσβλητων" αισθητήρων θερμοκρασίας. Για το σκοπό αυτό, θα προσπαθήσουμε να προσομοιώσουμε κάθε τρόπο αποτυχίας κάθε αισθητήρα, και να σχεδιάσουμε μια υλοποίηση FDIR που να προβλέπει όλες αυτές τις αστοχίες.

Το πρώτο βήμα για το λεπτομερή σχεδιασμό του FDIR είναι η προετοιμασία της **ανάλυσης FMEA** για το σύστημα. Όλες οι FMEA που εκτελούνται ακολουθούν τις απαιτήσεις του προτύπου ECSS-Q-ST-30-02C [40]. Η ανάλυση βασίζεται στην FMEA που πραγματοποιήθηκε για τον νανοδιορυφόρο AcubeSAT [47], και παρατίθεται στον **Πίνακα 5.8** για την πειραματική διάταξη. Διερευνώνται όλες οι αναμενόμενες αποτυχίες για τους δύο αισθητήρες, καθώς και μία αποτυχία για ολόκληρο το σύστημα που μπορεί να ανιχνευθεί από τους αισθητήρες που λειτουργούν σωστά. Κάθε ένας από αυτούς τους τρόπους αποτυχίας θα επαληθευτεί αργότερα με την εισαγωγή τροποποιήσεων λογισμικού ή υλικού.

5.3.1 Λεπτομερής σχεδιασμός FDIR



Σχήμα 5.11: Σύνοψη της διαδικασίας FDIR του αισθητήρα θερμοκρασίας. Για κάθε διεργασία που εκτελείται και για κάθε σφάλμα που εντοπίζεται, παράγεται και η αντίστοιχη τηλεμετρία.

Το FDIR των αισθητήρων ακούλουθει την ιεραρχική διαδικασία που περιγράφεται στο **Ενότητα 4.4**, που εμφανίζεται στο **Σχήμα 5.11** όπως έχει προσαρμοστεί για τους αισθητήρες.

Οι διπλές αποτυχίες δεν διερευνώνται στο πεδίο εφαρμογής του παρόντος εγγράφου, καθώς εξετάζονται κυρίως μόνο για συστήματα εκτόξευσης και επανδρωμένες αποστολές [36].

[40] ECSS Secretariat, *ECSS-Q-ST-30-02C - Failure Modes, Effects (and Criticality) Analysis (FMEA/FMECA)*.

[47] Retselis and Kanavouras, *Acube-SAT FMEA Worksheet*.

ID	Τρόπος Αποτυχίας	Αίτιο Αποτυχίας	Φάση αποστολής	Συνέπειες αποτυχίας:	Συνέπειες αποτυχίας: Τελικές	Εντοπισμός Αποτυχίας/Παρατηρήσιμα Συμπτώματα	Επίπεδο σοβαρότητας	Προβλέψεις αντιστάθμισης
Αισθητήρας θερμοκρασίας MCP9808 #1								
F-010	Προσωρινή απώλεια λειτουργικότητας	Τυχαίο, Ακτινοβολία	Όλες	Δεν λαμβάνονται μετρήσεις από τον αισθητήρα	Καμία	Καμία επικοινωνία μέσω I ² C	4	Πλεονασμός με 2 συνολικά αισθητήρες
F-020	Μόνιμη απώλεια λειτουργικότητας	Τυχαίο, Ακτινοβολία	Όλες	Δεν λαμβάνονται μετρήσεις από τον αισθητήρα	Καμία	Καμία επικοινωνία μέσω I ² C	4	Πλεονασμός με 2 συνολικά αισθητήρες
F-030	Βραχυκύκλωμα ακροδεκτών ισχύος	Τυχαίο, Ακτινοβολία	Όλες	Δεν λαμβάνονται μετρήσεις από τον αισθητήρα	Καμία	Καμία επικοινωνία μέσω I ² C	4	Αντίσταση περιορισμού ρεύματος
F-040	Προσωρινή αλλαγή τιμής	Τυχαίο, Ακτινοβολία	Όλες	Λανθασμένες θερμοκρασιακές μετρήσεις	Καμία	Μεγάλη διαφορά θερμοκρασίας μεταξύ 2 αισθητήρων	4	Πλεονασμός με 2 συνολικά αισθητήρες
F-050	Μόνιμη αλλαγή τιμής	Τυχαίο, Ακτινοβολία	Όλες	Λανθασμένες θερμοκρασιακές μετρήσεις	Καμία	Μεγάλη διαφορά θερμοκρασίας μεταξύ 2 αισθητήρων	4	Πλεονασμός με 2 συνολικά αισθητήρες
F-060	Κολλημένος ακροδέκτης εξόδου I ² C	Τυχαίο, Ακτινοβολία	Όλες	Αδυναμία επικοινωνίας και με τους 2 αισθητήρες	Καμία	Καμία επικοινωνία μέσω I ² C για όλους τους αισθητήρες	4	Αισθητήρες συνδεδεμένοι σε διαφορετικούς διαύλους I ² C
Αισθητήρας θερμοκρασίας MCP9808 #2								
F-070	Προσωρινή απώλεια λειτουργικότητας	Τυχαίο, Ακτινοβολία	Όλες	Δεν λαμβάνονται μετρήσεις από τον αισθητήρα	Καμία	Καμία επικοινωνία μέσω I ² C	4	Πλεονασμός με 2 συνολικά αισθητήρες
F-080	Μόνιμη απώλεια λειτουργικότητας	Τυχαίο, Ακτινοβολία	Όλες	Δεν λαμβάνονται μετρήσεις από τον αισθητήρα	Καμία	Καμία επικοινωνία μέσω I ² C	4	Πλεονασμός με 2 συνολικά αισθητήρες
F-090	Βραχυκύκλωμα ακροδεκτών ισχύος	Τυχαίο, Ακτινοβολία	Όλες	Δεν λαμβάνονται μετρήσεις από τον αισθητήρα	Καμία	Καμία επικοινωνία μέσω I ² C	4	Αντίσταση περιορισμού ρεύματος
F-100	Προσωρινή αλλαγή τιμής	Τυχαίο, Ακτινοβολία	Όλες	Λανθασμένες θερμοκρασιακές μετρήσεις	Καμία	Μεγάλη διαφορά θερμοκρασίας μεταξύ 2 αισθητήρων	4	Πλεονασμός με 2 συνολικά αισθητήρες
F-110	Μόνιμη αλλαγή τιμής	Τυχαίο, Ακτινοβολία	Όλες	Λανθασμένες θερμοκρασιακές μετρήσεις	Καμία	Μεγάλη διαφορά θερμοκρασίας μεταξύ 2 αισθητήρων	4	Πλεονασμός με 2 συνολικά αισθητήρες
F-120	Κολλημένος ακροδέκτης εξόδου I ² C	Τυχαίο, Ακτινοβολία	Όλες	Αδυναμία επικοινωνίας και με τους 2 αισθητήρες	Καμία	Καμία επικοινωνία μέσω I ² C για όλους τους αισθητήρες	4	Αισθητήρες συνδεδεμένοι σε διαφορετικούς διαύλους I ² C
Υποσύστημα								
F-130	Υπερθέρμανση	Βραχυκύκλωμα, περιβάλλον	Όλες	Αποτυχία ευάλωτων εξαρτημάτων	Απώλεια λειτουργικότητας υποστήματος	Μετρημένη θερμοκρασία εκτός του αναψευνόμενου εύρους	3	Θερμική ανάλυση με περιθώριο ασφαλείας, προστασία υπερέντασης

Η προσέγγιση ακολουθά τα παρακάτω βήματα, τα οποία εκτελούνται διαρκώς στο υπόβαθρο:

- Παρακολουθείται η κατάσταση υγείας του διαύλου I²C (**Ενότητα 5.2.3**). Οποιαδήποτε αποτυχία ή έλλειψη επικοινωνίας υποδεικνύει βλάβη στα πλεκτρονικά του αισθητήρα.
- Παρακολουθείται η έξοδος θερμοκρασίας των αισθητήρων. Τυχόν τιμές που βρίσκονται εκτός των ορίων των φυσικώς δυνατών θερμικών συνθηκών θεωρούνται ότι υποδομήλωνουν βλάβες του αισθητήρα ή της επικοινωνίας. Τυχόν τιμές εκτός των ορίων λειτουργίας των πλεκτρονικών του υποστήματος θεωρούνται ότι υποδομήλωνουν υπερθέρμανση και οδηγούν σε επανεκκίνηση (power-cycle) του αισθητήρα. Εάν οι τιμές εξακο-
- Για να διορθωθούν οι αποτυχίες, επιχειρείται πρώτα επανεκκίνηση (power-cycle) του αισθητήρα. Εάν οι τιμές

Πίνακας 5.8: FMEA στο σύστημα επίδειξης

Για αυτήν την επίδειξη, δεν θα εργευνήσουμε αποτυχίες των εσωτερικών στοιχείων του μικροελεγκτή. Δείτε στο [47] για το πλήρες FMEA και HSIA στους μικροελεγκτές.

λουθούν να είναι εσφαλμένες, τότε θεωρείται πως ο αισθητήρας δεν μπορεί να ανακάμψει από την βλάβη, και απλώς απομονώνεται και απενεργοποιείται.

Hardware/Software Interaction Analysis Αφού έχουμε καθορίσει τους πιθανούς τρόπους αποτυχίας και τη λογική λειτουργίας του συστήματος, είμαστε έτοιμοι να αναλύσουμε κάθε πιθανή αποτυχία για να ορίσουμε τις αναμενόμενες εισόδους λογισμικού για το FDIR. Πιο συγκεκριμένα, για κάθε αποτυχία πρέπει να οριστούν τα εξής [36, σ. 84]:

- **Παραμέτροι προς παρακολούθηση** για εντοπισμό
- **Εύρη τιμών** για να διαχωρίσουν τις καταστάσεις ύπαρξης ή μη κάποιας αποτυχίας
- Ενέργειες απομόνωσης και ανάκαμψης για την αποτροπή της διάδοσης της αποτυχίας και, εάν είναι δυνατόν, για την επαναφορά του συστήματος σε κανονική λειτουργία.

[36] Space Avionics Open interface Architecture, *SAVOIR FDIR Handbook*.

Οι παραπάνω πληροφορίες παρατίθενται στον λεγόμενο πίνακα Hardware/Software Interaction Analysis (HSIA), ο οποίος συνδέει κάθε αποτυχία (που προσδιορίζεται στο Failure Mode and Effects Analysis) με τις αντίστοιχες παραμέτρους λογισμικού χαριτλού επιπέδου (Πίνακας 5.9).

Για τους σκοπούς αυτής της επίδειξης, θα θεωρήσουμε ότι όλα τα στοιχεία FMEA μπορούν να οδηγήσουν σε επίφοβες αποτυχίες (feared events), δηλαδή ότι το FDIR θα καλύπτει όλες τις αναγνωρισμένες πιθανές αποτυχίες.

Πιο συγκεκριμένα, το HSIA προσθέτει τις ακόλουθες σημαντικές πληροφορίες στο FMEA [40]:

- **Παραπορούμενες παραμέτροι και συνθήκες** που πυροδοτούν την ενέργεια ανάκτησης.

[40] ECSS Secretariat, *ECSS-Q-ST-30-02C – Failure Modes, Effects (and Criticality) Analysis (FMEA/FMECA)*.

Αυτές παρατίθενται εδώ σε όρους παραμέτρων PUS που μπορούν εύκολα να διαβαστούν και να συγκριθούν από το λογισμικό. Αυτός ο κατάλογος συνδέεται στη συνέχεια με τους αντίστοιχους ορισμούς της υπηρεσίας ST[12] *on-board monitoring*, που παρατίθενται στον Πίνακα 5.11.

Προσωρινή αποτυχία: Προκαλείται από γεγονότα όπως bit flips, θόρυβο ή μεταβατικά φαινόμενα και μπορεί να διορθωθεί χωρίς παρέμβαση ή με μια απλή επανεκκίνηση/κύκλο τροφοδοσίας.

Είναι σημαντικό να σημειωθεί ότι καμία μεμονωμένη εκτός ορίων τιμή δεν θα πρέπει να προκαλέσει ενέργεια ανάκτησης. Αυτό γίνεται ώστε πρόσκαιρα σφάλματα ή προσωρινά προβλήματα επικοινωνίας να μπν μπορούν να οδηγήσουν σε κάποια άσκοπη ενεργοποίηση του FDIR, που θα μπορούσε να απενεργοποιήσει ένα καλά λειτουργικό εξάρτημα.

Μόνιμη αποτυχία: Μια αποτυχία (συνήθως στο υλικό, π.χ. λόγω SELs [51] ή βραχυκυκλωμάτων) που δεν μπορεί να διορθωθεί αυτόνομα.

Η διάκριση μεταξύ προσωρινών και μόνιμων αποτυχιών γίνεται

μετρώντας τον αριθμό των διαδοχικών αποτυχιών. Προτού απενεργοποιηθεί μόνιμα ένας αισθητήρας, πρέπει ενδεικτικά να έχουν αποτύχει τουλάχιστον 5 προσπάθειες επικοινωνίας.

Η χρήση 2 αισθητήρων που λειτουργούν σε ενεργό πλεονασμό επιτρέπει επίσης τη διερεύνηση της διαφοράς μεταξύ των δύο θερμοκρασιών για την εξαγωγή συμπερασμάτων σχετικά με την υγεία των παραγόμενων δεδομένων. Εάν η απόλυτη διαφορά μεταξύ των 2 τιμών είναι πολύ υψηλή, υποθέτοντας ότι οι αισθητήρες βρίσκονται αρκετά κοντά μεταξύ τους, θεωρείται ότι υπάρχει κατάσταση σφάλματος. Ωστόσο, καθώς οι αισθητήρες είναι μόνο 2, δεν είναι δυνατόν να συμπεράνουμε ποιος αισθητήρας έχει αποτύχει.

- **Δράσεις αποκατάστασης και απομόνωσης βλάβης.**

Αυτές απαριθμούνται σε όρους γεγονότων PUS (*ST[05] event reporting*) και ενεργειών (*ST[19] eventaction*), καταγεγραμμένες στον [Πίνακα 5.12](#).

Οι επιλεγμένες ενέργειες αποκατάστασης αντιστοιχούν στην προσέγγιση του [Σχήματος 5.11](#). Επιπλέον, όταν δεν είναι γνωστό ποιος αισθητήρας έχει αποτύχει, χρησιμοποιείται μια ασφαλής προσέγγιση επανεκκίνησης ή απενεργοποίησης και των δύο αισθητήρων.

Παράμετροι Μετά την ολοκλήρωση της Hardware/Software Interaction Analysis, είμαστε έτοιμοι να ορίσουμε τις αριθμητικές παραμέτρους της υπηρεσίας *ST[20] on-board parameter management* [35] ([Πίνακας 5.10](#)).

Πέρα από τις εξόδους των αισθητήρων, έχουμε συμπεριλάβει τη **διαφορά** (Δ) μεταξύ των δύο θερμοκρασιών, καθώς το πρότυπο PUS δεν παρέχει τρόπους για την εκτέλεση αριθμητικών υπολογισμών. Επιπλέον, περιλαμβάνεται η κατάσταση λειτουργίας του κάθε αισθητήρα, η οποία είναι μια απαριθμημένη (enumerated) τιμή.¹⁶

Αυτές οι παράμετροι τηλεμετρώνται στον σταθμό βάσης και εμφανίζονται στο χρήστη περιοδικά, μέσω της υπηρεσίας *ST[03] housekeeping* PUS και των ορδιοφάρων που αυτή εκπέμπει. Ωστόσο, όταν το CubeSat βρίσκεται στο διάστημα, ο σταθμός βάσης δεν θα έχει συνεχή πρόσβαση σε αυτά τα δεδομένα, καθώς ο δορυφόρος δεν θα είναι ορατός από τους επίγειους σταθμούς καθ' όλη τη διάρκεια της τροχιάς του.

Ορισμοί παρακολούθησης Μετά τον ορισμό των παραμέτρων, θα πρέπει να περιγραφούν οι ορισμοί παρακολούθησης που καθορίζουν τα αποδεκτά και μη εύρον τιμών για κάθε παράμετρο. Η διαχείριση αυτών γίνεται από την υπηρεσία *ST[12] on-board moni-*

[35] ECSS Secretariat, *ECSS-E-ST-70-41C – Telemetry and Telecommand Packet Utilization*.

¹⁶ Η τιμή μπορεί να είναι μία από τις **NOMINAL** (ο αισθητήρας λειτουργεί κανονικά), **TIMEOUT** (έχει εντοπιστεί κάποιο σφάλμα) ή **DISABLED** (ο αισθητήρας είναι θρησκευμένος)

ID	Τρόπος Αποτυχίας	Παρατηρούμενες Παράμετροι	ST[12] ID	ST[05] ID	Ενέργεια αποκατάστασης
Αισθητήρας θερμοκρασίας MCP9808 #1					
F-010	Προσωρινή απώλεια λειτουργικότητας	T1_Status = TIMEOUT 2 φορές	0	0	Επανεκκίνηση αισθητήρα 1
F-020	Μόνιμη απώλεια λειτουργικότητας	T1_Status = TIMEOUT 5 φορές	2	2	Αγνόστη αισθητήρα 1
F-030	Βραχυκύκλωμα ακροδεκτών ισχύος	T1_Status = TIMEOUT 5 φορές	2	2	Αγνόστη αισθητήρα 1
F-040	Προσωρινή αλλαγή τιμής	$\begin{cases} \Delta T > 20^\circ\text{C} \text{ ή} \\ T_1 > 100^\circ\text{C} \text{ ή} \\ T_1 < -40^\circ\text{C} \end{cases}$ 2 φορές	4, 6	0, 4	Επανεκκίνηση αισθητήρα 1
F-050	Μόνιμη αλλαγή τιμής	$\begin{cases} \Delta T > 20^\circ\text{C} \text{ ή} \\ T_1 > 100^\circ\text{C} \text{ ή} \\ T_1 < -40^\circ\text{C} \end{cases}$ 5 φορές	7, 9	2, 5	Αγνόστη αισθητήρα 1
F-060	Κολλημένος ακροδέκτης εξόδου I ² C	T1_Status = TIMEOUT 5 φορές	2	2	Αγνόστη αισθητήρα 1
Αισθητήρας θερμοκρασίας MCP9808 #2					
F-070	Προσωρινή απώλεια λειτουργικότητας	T2_Status = TIMEOUT 2 φορές	1	1	Επανεκκίνηση αισθητήρα 2
F-080	Μόνιμη απώλεια λειτουργικότητας	T2_Status = TIMEOUT 5 φορές	3	3	Αγνόστη αισθητήρα 2
F-090	Βραχυκύκλωμα ακροδεκτών ισχύος	T2_Status = TIMEOUT 5 φορές	3	3	Αγνόστη αισθητήρα 2
F-100	Προσωρινή αλλαγή τιμής	$\begin{cases} \Delta T > 20^\circ\text{C} \text{ ή} \\ T_2 > 100^\circ\text{C} \text{ ή} \\ T_2 < -40^\circ\text{C} \end{cases}$ 2 φορές	5, 6	1, 4	Επανεκκίνηση αισθητήρα 2
F-110	Μόνιμη αλλαγή τιμής	$\begin{cases} \Delta T > 20^\circ\text{C} \text{ ή} \\ T_2 > 100^\circ\text{C} \text{ ή} \\ T_2 < -40^\circ\text{C} \end{cases}$ 5 φορές	8, 9	3, 5	Αγνόστη αισθητήρα 2
F-120	Κολλημένος ακροδέκτης εξόδου I ² C	T2_Status = TIMEOUT 5 φορές	3	3	Αγνόστη αισθητήρα 2
Υποσύστημα					
F-130	Υπερθέρμανση	T ₁ > 40 °C or T ₂ > 40 °C	10, 11	6	Επανεκκίνηση υποσυστήματος

Πίνακας 5.9: Πίνακας HSIA

ID	Παράμετρος	Μονάδα	Τύπος	Εγγραφή
0	Θερμοκρασία 1	°C	float	R
1	Θερμοκρασία 2	°C	float	R
2	Δ Θερμοκρασίας	°C	float	R
3	Κατάσταση Θερμοκρασία 1		Απαρίθμηση	RW
4	Κατάσταση Θερμοκρασία 2		Απαρίθμηση	RW
5	Κατάσταση Θερμοκρασία 1+2		Απαρίθμηση	R

Πίνακας 5.10: Λίστα παραμέτρων ST[20]

R: Μόνο ανάγνωση

RW: Ανάγνωση και εγγραφή

toring του PUS.

Οι συνθήκες εγκυρότητας ελέγχου είναι σημαντικό να αναφερθούν σε αυτή την περίπτωση, καθώς χρησιμοποιούν τις τρεις παραμέτρους Κατάστασης του Πίνακα 5.10 για να αποτρέψουν την εκτέλεση ελέγχων σε απενεργοποιημένα περιφερειακά. Ως εκ τούτου, οι αισθητήρες που δεν παρέχουν τιμές λόγω προβλημάτων υλικού, ενεργειών FDIR ή κάποιας ρωτής πλευρούς από το Σταθμό Βάσης, δεν μπορούν να ενεργοποιήσουν ενέργειες FDIR.

ID	Παρακολούθηση παράμετρος	Συνθήκες εγκυρότητας ελέγχου			Έλεγχος		
		Παράμετρος εγκυρότητας	Ανα- μενόμενη τιμή	Περίοδος	Αρ. μετρήσεων	Είδος	Κριτήριο
0	Κατάσταση Θερμ. 1			500 ms	2	Ισότητα	$\neq \text{TIMEOUT}$
1	Κατάσταση Θερμ. 2			500 ms	2	Ισότητα	$\neq \text{TIMEOUT}$
2	Κατάσταση Θερμ. 1			500 ms	5	Ισότητα	$\neq \text{TIMEOUT}$
3	Κατάσταση Θερμ. 2			500 ms	5	Ισότητα	$\neq \text{TIMEOUT}$
4	Θερμοκρασία 1	Κατάσταση Θερμ. 1	NOMINAL	500 ms	2	Εύρος	$-40^{\circ}\text{C} < t < 100^{\circ}\text{C}$
5	Θερμοκρασία 2	Κατάσταση Θερμ. 2	NOMINAL	500 ms	2	Εύρος	$-40^{\circ}\text{C} < t < 100^{\circ}\text{C}$
6	Δ Θερμοκρασία	Κατάσταση Θερμ. 1+2	NOMINAL	500 ms	2	Εύρος	$-20^{\circ}\text{C} < \Delta < 20^{\circ}\text{C}$
7	Θερμοκρασία 1	Κατάσταση Θερμ. 1	NOMINAL	500 ms	5	Εύρος	$-40^{\circ}\text{C} < t < 100^{\circ}\text{C}$
8	Θερμοκρασία 2	Κατάσταση Θερμ. 2	NOMINAL	500 ms	5	Εύρος	$-40^{\circ}\text{C} < t < 100^{\circ}\text{C}$
9	Δ Θερμοκρασία	Κατάσταση Θερμ. 1+2	NOMINAL	500 ms	5	Εύρος	$-20^{\circ}\text{C} < \Delta < 20^{\circ}\text{C}$
10	Θερμοκρασία 1	Κατάσταση Θερμ. 1	NOMINAL	500 ms	5	Εύρος	$t < 50^{\circ}\text{C}$
11	Θερμοκρασία 2	Κατάσταση Θερμ. 2	NOMINAL	500 ms	5	Εύρος	$t < 50^{\circ}\text{C}$

Πίνακας 5.11: Λίστα ορισμών παρακολούθησης της υπηρεσίας ST[12]. Τα σύμβολα t και Δ αναπαριστούν τις εκάστοτε τιμές των παραμέτρων.

- PUS γεγονότα (υπηρεσία ST[05]), τα οποία εκπέμπονται αυτόματα όταν ένας ορισμός παρακολούθησης ξεπερνά τα όριά του
- PUS ορισμοί γεγονότων-αντιδράσεων (υπηρεσία ST[19]), οι οποίοι συνδέονται κάθε εμφάνιση ενός γεγονότος με μία αποθηκευμένη πλευρού που εκτελείται αμέσως και αναπαριστά τις διαδικασίες ανάκτησης του FDIR.

Όλοι αυτοί οι ορισμοί μπορούν εύκολα να τροποποιηθούν από τους χειριστές με τη χρήση μεμονωμένων τηλεεντολών, οι οποίες επιδρούν επάνω στην βάση δεδομένων του διαστημικού εξοπλισμού.

ID γεγονότος	Ορισμός Παρακολούθησης	Αντίδραση
0	0, 4	Επανεκκίνηση αισθητήρα 1
1	1, 5	Επανεκκίνηση αισθητήρα 2
2	2, 7	Κατάσταση αισθητήρα 1 = DISABLED
3	3, 8	Κατάσταση αισθητήρα 2 = DISABLED
4	6	Επανεκκίνηση αισθητήρων 1 + 2
5	9	Κατάσταση αισθητήρων 1 + 2 = DISABLED
6	10, 11	Επανεκκίνηση συστήματος

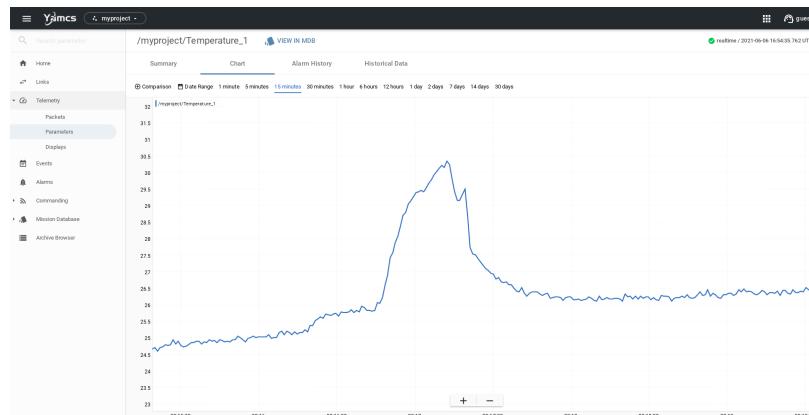
Πίνακας 5.12: Λίστα ορισμών γεγονότων-αντιδράσεων της υπηρεσίας ST[19]

5.4 Επικύρωση FDIR

5.4.1 Κανονική λειτουργία

Στην κανονική λειτουργία, δεν προσομοιώνεται καμία βλάβη. Το σύστημα είναι ενεργοποιημένο και παράγει τηλεμετρία που υπακούει στα πρότυπα ECSS (Σχήμα 5.12), η οποία περιλαμβάνει τιμές παραμέτρων που αναλύονται από το YAMCS (Σχήμα 5.13). Επιπλέον, διατηρείται ένα αρχείο καταγραφής για διαγνωστικούς σκοπούς (Σχήμα 5.14). Το σύστημα συμμορφώνεται με τις ορισμένες απαιτήσεις (Ενότητα 5.1.1) και μπορεί να μεταδώσει τη θερμοκρασία στον Σταθμό Βάσης, όπως είναι αναμενόμενο.

Σχήμα 5.12: Περιοδικοί φάσματα παραγόνται από τον μικροελεγκτή, όπως φαίνονται στο YAMCS



Σχήμα 5.13: Εξέλιξη μιας παραμέτρου θερμοκρασίας στο χρόνο, όπως φαίνεται στο YAMCS

Ιδιαίτερο ενδιαφέρον έχει να παρατηρήσουμε την κατάσταση του συστήματος FDIR κατά την εικόνα. Η πρώτη πληροφορία που λαμβάνουμε φαίνεται στην Σχήμα 5.15. Η εικόνα αυτή δείχνει τις μεταβάσεις (transitions) της υπηρεσίας *ST[12] on-board monitoring*, όπως φαίνονται στην διεπαφή που αναπτύξαμε (Ενότητα 5.2.5). Πρακτικά, αυτή η προβολή δείχνει το αποτέλεσμα του κάθε ελέγχου, και αν η τιμή είναι εντός ορίων, εκτός ορίων, ή αν ο έλεγχος είναι απενεργοποιημένος.

Στο συγκεκριμένο παράδειγμα, παρατηρούμε ότι και οι 12 καταχωριμένοι έλεγχοι μεταβαίνουν από την κατάσταση **Unchecked** στην

Γενικά είδη κατάστασης [35]:

Unchecked: Ο έλεγχος είναι απενεργοποιημένος.

OK: Η μεταβλητή βρίσκεται εντός ορίων.

Invalid: Η συνθήκη εγκυρότητας του έλεγχου είναι ψευδής. Ο έλεγχος δεν πραγματοποιείται.

```

909601 [debug] T1 = 28.18
909701 [debug] T1 = 28.08
909801 [debug] T1 = 28.14
909901 [debug] T1 = 27.97
910000 [trace] New TM [3,25]
910001 [debug] T1 = 28.10
910101 [debug] T1 = 28.03
910201 [debug] T1 = 28.12
910301 [debug] T1 = 27.90
910401 [debug] T1 = 28.08

```

Σχήμα 5.14: Έξοδος καταγραφής σε κανονική λειτουργία. Ο πρώτος αριθμός δηλώνει τους χτύπους ρολογιού του μικροελεγκτή, που μετρώνται σε χιλιοστά του δευτερολέπτου από την εκκίνηση.

Check Transitions					CLEAR
Timestamp	PMON	Parameter	Value	Transition	
6500	11	Temp. 2	26.4375	Unchecked → OK	
6500	10	Temp. 1	26.3125	Unchecked → OK	
6000	9	Temp. Δ	0.125	Unchecked → OK	
3500	6	Temp. Δ	0.125	Unchecked → OK	
3000	8	Temp. 2	26.375	Unchecked → OK	
3000	7	Temp. 1	26.25	Unchecked → OK	
3000	3	Temp. 2 Status	1	Unchecked → OK	
3000	2	Temp. 1 Status	1	Unchecked → OK	
1500	5	Temp. 2	26.375	Unchecked → OK	
1500	4	Temp. 1	26.25	Unchecked → OK	
1500	1	Temp. 2 Status	1	Unchecked → OK	
1500	0	Temp. 1 Status	1	Unchecked → OK	

Σχήμα 5.15: Μεταβάσεις που πραγματοποιούνται κατά την εκκίνηση

κατάσταση OK. Αυτό συμβαίνει καθώς, με την εκκίνηση του συστήματος, όλοι οι ορισμοί παρακολούθησης βρίσκονται στην αρχική κατάσταση Unchecked, που δηλώνει πως δεν έχουν πραγματοποιηθεί ποτέ. Μετά από την πάροδο της περιόδου μετρήσεων επί του αριθμού επαναλήψεων του κάθε ορισμού, η μεταβλητή θεωρείται πως έχει τιμή εντός ορίων, και ο ορισμός είναι OK.

Ο διαφορετικός χρόνος μετάβασης του κάθε ορισμού οφείλεται στον διαφορετικό **αριθμό επαναλήψεων**. Για να καταχωριθεί η μετάβαση, πρέπει η μεταβλητή να έχει μετρηθεί όσες φορές δηλώνει αυτός ο αριθμός, που ορίζεται από τους χειριστές.

Εφ' όσον οι μεταβάσεις οδηγούν σε μια κατάσταση εντός ορίων, δεν παράγεται κάποια άλλη αναφορά, ούτε πυροδοτείται κάποιος άλλος μπχανισμός επιδιόρθωσης. Το σύστημα λειτουργεί σωστά.

PMON ID	Monitored Parameter	Validity condition	Interval	PMON Status	Repetition #	Check	Acquisition time
0	Temp. 1 Status	Temp. 1 0 0x0	500 ms	OK	2	x = Nominal 0x2 Temp1_Timeout_Temporary	2021-06-25T17:37:49.086Z
1	Temp. 2 Status	Temp. 1 0 0x0	500 ms	OK	2	x = Nominal 0x2 Temp2_Timeout_Temporary	2021-06-25T17:37:49.094Z
2	Temp. 1 Status	Temp. 1 0 0x0	500 ms	OK	5	x = Nominal 0x2 Temp1_Timeout_Permanent	2021-06-25T17:37:49.102Z
3	Temp. 2 Status	Temp. 1 0 0x0	500 ms	OK	5	x = Nominal 0x2 Temp2_Timeout_Permanent	2021-06-25T17:37:49.109Z
4	Temp. 1	Temp. 1 Status 1 0xffff	500 ms	OK	2	-40 ≤ x ≤ 100 Temp1_UnexpectedValue_Temporary Temp1_UnexpectedValue_Temporary	2021-06-25T17:37:49.117Z
5	Temp. 2	Temp. 2 Status 1 0xffff	500 ms	OK	2	-40 ≤ x ≤ 100 Temp2_UnexpectedValue_Temporary Temp2_UnexpectedValue_Temporary	2021-06-25T17:37:49.124Z
6	Temp. Δ	Temp. 1+2 Status 1 0xffff	500 ms	OK	6	-5 ≤ x ≤ 5 TempDelta_UnexpectedValue_Temporary TempDelta_UnexpectedValue_Temporary	2021-06-25T17:37:49.132Z
7	Temp. 1	Temp. 1 Status 1 0xffff	500 ms	OK	5	-40 ≤ x ≤ 100 Temp1_UnexpectedValue_Permanent Temp1_UnexpectedValue_Permanent	2021-06-25T17:37:49.140Z
8	Temp. 2	Temp. 2 Status 1 0xffff	500 ms	OK	5	-40 ≤ x ≤ 100 Temp2_UnexpectedValue_Permanent Temp2_UnexpectedValue_Permanent	2021-06-25T17:37:49.147Z
9	Temp. Δ	Temp. 1+2 Status 1 0xffff	500 ms	OK	11	-5 ≤ x ≤ 5 TempDelta_UnexpectedValue_Permanent TempDelta_UnexpectedValue_Permanent	2021-06-25T17:37:49.155Z
10	Temp. 1	Temp. 1 Status 1 0xffff	500 ms	OK	12	-Infinity ≤ x ≤ 50 Subsystem_Overheat_Temp1 Subsystem_Overheat_Temp1	2021-06-25T17:37:49.163Z
11	Temp. 2	Temp. 2 Status 1 0xffff	500 ms	OK	12	-Infinity ≤ x ≤ 50 Subsystem_Overheat_Temp2 Subsystem_Overheat_Temp2	2021-06-25T17:37:49.170Z

Σχήμα 5.16: Λίστα ορισμών παρακολούθησης στην εκκίνηση, χωρίς καμία αποτυχία. Τα δεδομένα αυτά λαμβάνονται ως τηλεμετρία από τον μικροελεγκτή και αντιστοιχούν ακριβώς στις τιμές του Πίνακα 5.11.

Ακόμα, κατά την εκκίνηση, δημιουργείται ένα γεγονός της υπηρεσίας *ST[05] event reporting* που ανακοινώνει την εκκίνηση του μικροελεγκτή ([Σχήμα 5.18](#)).

Event-Action Definitions				
ID	Event	Status	Request	
1	Temp1_Timeout_Temporary	On	TC[8,1] restart_sensor 1	
2	Temp2_Timeout_Temporary	On	TC[8,1] restart_sensor 2	
3	Temp1_Timeout_Permanent	On	TC[20,3]	
4	Temp2_Timeout_Permanent	On	TC[20,3]	
5	Temp1_UnexpectedValue_Temporary	On	TC[8,1] restart_sensor 1	
6	Temp2_UnexpectedValue_Temporary	On	TC[8,1] restart_sensor 2	
7	TempDelta_UnexpectedValue_Temporary	On	TC[8,1] restart_sensor 3	
8	Temp1_UnexpectedValue_Permanent	On	TC[20,3]	
9	Temp2_UnexpectedValue_Permanent	On	TC[20,3]	
10	TempDelta_UnexpectedValue_Permanent	On	TC[20,3]	
11	TempDelta_UnexpectedValue_Permanent	On	TC[20,3]	
12	Subsystem_Overheat_Temp1	On	TC[8,1] restart	
13	Subsystem_Overheat_Temp2	On	TC[8,1] restart	

The screenshot shows a YAMCS event log interface. At the top, there's a header with 'Events' and a '+ CREATE EVENT' button. To the right is a timestamp 'realtime / 2021-06-25 19:02:59.623 UTC'. Below the header are several filter options: 'Filter by text search' (with a placeholder), 'Columns -', 'Info level - Any source -', 'Last hour -', and 'Jump to now'. A note says 'Showing events from the last hour ending at 2021-06-25 19:02:54.623 UTC (Mission Time)'. On the right, there's a 'Download events' link. The main table has four columns: 'Generation Time', 'Message', 'Type', and 'Source'. One row is visible: '2021-06-25 18:11:02.140 UTC' (Generation Time), 'ECSS informative event 3 MCU_Start' (Message), 'ECSS_Informative_Event' (Type), and 'ECSS_Event_Parser' (Source).

5.4.2 Προσομοίωση αστοχιών

Για την προσομοίωση κάθε αποτυχίας χρησιμοποιήθηκαν οι ακόλουθες μέθοδοι:

- Για τις αποτυχίες επικοινωνίας μεταξύ εξαρτημάτων, κάθε αισθητήρας απλά αποσυνδέθηκε φυσικά από το σύστημα.
- Για μικρές αλλαγές στη θερμοκρασία, ένα πιστόλι θερμού αέρα ή άλλη πηγή θερμότητας στρεφόταν προς κάθε αισθητήρα, με κατάλληλη φροντίδα ώστε να μη ξεπεραστούν τα λειτουργικά όρια των εξαρτημάτων.
- Για μεταβολές της θερμοκρασίας που δεν ήταν πρακτικά δυνατό να προσομοιωθούν, η βλάβη εισήχθη στο λογισμικό, τροποποιώντας χειροκίνητα τις αντίστοιχες μεταβλητές.

Τα δύο κουμπιά που υπάρχουν στην αναπτυξιακή πλακέτα χρησιμοποιήθηκαν ένα απλό μέσο για την "εισαγωγή" ακραίων τιμών θερμοκρασίας.

Σχήμα 5.17: Λίστα ορισμών γεγονότος-αντίδρασης στην εκκίνηση, χωρίς καμία αποτυχία. Τα δεδομένα αυτά λαμβάνονται ως τηλεμετρία από τον μικροελεγκτή και αντιστοιχούν ακριβώς στις τιμές του [Πίνακα 5.12](#).

Σχήμα 5.18: Το γεγονός έναρξης του μικροελεγκτή, μέσα από το YAMCS

ID	Τρόπος Αποτυχίας	Ενέργεια αποκατάστασης	Τρόπος προσομοίωσης
Αισθητήρας θερμοκρασίας MCP9808 #1			
F-010	Προσωρινή απώλεια λειτουργικότητας	Επανεκκίνηση αισθητήρα 1	Υλικό: Προσωρινή αποσύνδεση καλωδίου
F-020	Μόνιμη απώλεια λειτουργικότητας	Αγνόηση αισθητήρα 1	Υλικό: Μόνιμη αποσύνδεση καλωδίου
F-030	Βραχυκύλωμα ακροδεκτών ισχύος	Αγνόηση αισθητήρα 1	<i>Τίδιο με F-020</i>
F-040	Προσωρινή αλλαγή τιμής	Επανεκκίνηση αισθητήρα 1	Λογισμικό: Προσομοίωση μεγάλης τιμής (προσωρινά) Υλικό: Αύξηση θερμοκρασιακής διαφοράς (προσωρινά)
F-050	Μόνιμη αλλαγή τιμής	Αγνόηση αισθητήρα 1	Λογισμικό: Προσομοίωση μεγάλης τιμής (μόνιμα) Υλικό: Αύξηση θερμοκρασιακής διαφοράς (μόνιμα)
F-060	Κολλημένος ακροδέκτης εξόδου I ² C	Αγνόηση αισθητήρα 1	Υλικό: Σύνδεση ακροδέκτη στη γείωση
Αισθητήρας θερμοκρασίας MCP9808 #2			
<i>Τίδιο με τον αισθητήρα #1</i>			
Υποσύστημα			
F-130	Υπερθέρμανση	Επανεκκίνηση υποσυστήματος	Λογισμικό: Προσομοίωση μεγάλης τιμής (προσωρινά)

F-010: Προσωρινή απώλεια λειτουργικότητας

Η πρώτη αποτυχία που θα προσομοιώσουμε θα είναι η απώλεια λειτουργίας του διαύλου I²C ή του περιφερειακού. Η προσομοίωση γίνεται στο υλικό μέσω της προσωρινής αποσύνδεσης καλωδίου.

Διαδικασία

1. Αποσυνδέουμε το καλώδιο SDA ή SCL από το breadboard
2. Μετά από 2 δευτερόλεπτα, επανασυνδέουμε το καλώδιο, ώστε να αναπαραστήσουμε μία προσωρινή (και όχι μόνη) βλάβη

Οπτική ένδειξη Η 1^η ενδεικτική λυχνία αναβοσβήνει μία φορά, και παραμένει ανοιχτή.

Αποτελέσματα Για αυτήν την ανάλυση, θα παρουσιαστούν με λεπτομέρεια όλες οι διαδικασίες που συμβαίνουν στο υπόβαθρο και πώς φαίνονται σε έναν χειριστή. Στις επόμενες αναλύσεις, τα αποτελέσματα αυτά θα παρουσιαστούν μόνο συνοπτικά, χρησιμοποιώντας το εργαλείο του πίνακα μεταβάσεων.

Προτού μελετήσουμε τα αποτελέσματα της αποτυχίας και του FDIR, θα κοιτάξουμε τις παραμέτρους και τις τιμές αυτών.

⌚ Status_Temperature_1	enumeration	Telemetered	Nominal
⌚ Status_Temperature_2	enumeration	Telemetered	Nominal
⌚ Temperature_1	float	Telemetered	25.5 celsius
⌚ Temperature_2	float	Telemetered	25.75 celsius

⌚ Status_Temperature_1	enumeration	Telemetered	Timeout
⌚ Status_Temperature_2	enumeration	Telemetered	Nominal
⌚ Temperature_1	float	Telemetered	0 celsius
⌚ Temperature_2	float	Telemetered	25.75 celsius

Στο Σχήμα 5.19, φαίνονται οι τιμές θερμοκρασίας και κατάστασης σε κανονική λειτουργία. Οι τιμές λαμβάνονται κανονικά και οι αισθητήρες αποστέλλουν δεδομένα. Τη στιγμή που αποσυνδέουμε το καλώδιο, το σύστημα¹⁷ θέτει την τιμή κατάστασης σε TIMEOUT, καθώς αναγνωρίζει την αποτυχία της ζεύξης (Σχήμα 5.20). Ταυτόχρονα, η τιμή της θερμοκρασίας γίνεται 0, καθώς δεν υπάρχει κάποια ενεργή μέτρηση.¹⁸

Μετά από την αλλαγή των τιμών των παραμέτρων, σειρά έχει η υπηρεσία ST[12] *on-board monitoring* και οι ορισμοί παρακολούθησης (Σχήμα 5.21 και Πίνακας 5.14). Εδώ με τη σειρά γίνονται τα εξής:

Σχήμα 5.19: Θερμοκρασιακές παραμέτροι YAMCS σε κανονική λειτουργία

Σχήμα 5.20: Θερμοκρασιακές παραμέτροι YAMCS υπό την αποτυχία F-010

¹⁷ Η διαδικασία αυτή πραγματοποιείται από τον οδηγό του περιφερειακού, και όχι από την υλοποίηση των ECSS services.

¹⁸ Σε ένα πραγματικό σύστημα θα είχε νόημα και να διατηρείται η προηγούμενη τιμή της θερμοκρασίας. Σε εμάς η τιμή τέθηκε στο 0 για να επιβεβαιώσουμε πως ένα αποτυχημένο περιφερειακό που επιστρέφει λανθασμένες τιμές δεν μπορεί να επηρεάσει αρνητικά το υπόλοιπο σύστημα.

Check Transitions					CLEAR
Timestamp (ms)	PMON	Parameter	Value	Transition	
3622500	10	Temp. 1	25.5	Invalid → OK	
3622000	9	Temp. Δ	0.25	Invalid → OK	
3619500	6	Temp. Δ	0.3125	Invalid → OK	
3619000	7	Temp. 1	25.5	Invalid → OK	
3617500	4	Temp. 1	25.5	Invalid → OK	
3617500	0	Temp. 1 Status	1	Unexpected_Value → OK	
3616000	0	Temp. 1 Status	2	OK → Unexpected_Value	
3615000	10	Temp. 1	0	OK → Invalid	
3615000	9	Temp. Δ	25.6875	OK → Invalid	
3615000	7	Temp. 1	0	OK → Invalid	
3615000	6	Temp. Δ	25.6875	OK → Invalid	
3615000	4	Temp. 1	0	OK → Invalid	

Σχήμα 5.21: Πίνακας μετάβασης ορισμένων παρακολούθησης (αποτυχία F-010 όπως φαίνεται στη διεπαφή PUS)

Χρόνος (ms)	PMON	Παράμετρος	Μετάβαση
0	4	Θερμ. 1 = 0	OK → Invalid
0	6	Θερμ. Δ = 25.6875	OK → Invalid
0	7	Θερμ. 1 = 0	OK → Invalid
0	9	Θερμ. Δ = 25.6875	OK → Invalid
0	10	Θερμ. 1 = 0	OK → Invalid
1000	0	Κατάστ. Θερμ. 1 = TIMEOUT	OK → Unexpected_Value
Αποκατάσταση βλάβης			
2500	0	Κατάστ. Θερμ. 1 = NOMINAL	Unexpected_Value → OK
2500	4	Θερμ. 1 = 25.5	Invalid → OK
4000	7	Θερμ. 1 = 25.5	Invalid → OK
4500	6	Θερμ. Δ = 0.3125	Invalid → OK
7000	9	Θερμ. Δ = 0.25	Invalid → OK
7000	10	Θερμ. 1 = 25.5	Invalid → OK

- Με το που αποσυνδέεται ο αισθητήρας, οι ορισμοί 4, 6, 7, 9 και 10 (Πίνακας 5.11) αποκτούν κατάσταση **Invalid**. Αυτό συμβαίνει καθώς για αυτούς τους ορισμούς υπάρχει **συνθήκη εγκυρότητας ελέγχου**. Αν ο αισθητήρας 1 είναι απενεργοποιημένος,¹⁹ δεν έχει κανένα νόημα η λίγη των τιμών του για εύρεση λαθών.

Επομένως, όλοι οι ορισμοί που συνδέονται με τον αισθητήρα 1 απενεργοποιούνται και δεν μπορούν να πυροδοτήσουν κάποια αντίδραση. Αυτό γίνεται άμεσα.

- Μετά από δύο μετρήσεις και χρόνο $2 \cdot 500 = 1000$ ms, ο ορισμός παρακολούθησης 0 επιβεβαιώνει πως η παράμετρος **Κατάσταση Θερμοκρασία 1** έχει ανεπιθύμητη τιμή. Ως αποτέλεσμα, ο ορισμός παρακολούθησης μεταβαίνει σε κατάσταση **Unexpected_Value**, και πυροδοτεί όλη τη διαδικασία αναφοράς και διόρθωσης της βλάβης. Η αποτυχία του αισθητήρα 1 έχει πλέον εντοπιστεί, και το σύστημα FDIR έχει αρχίσει να δρα.

Πίνακας 5.14: Πίνακας μετάβασης ορισμένων παρακολούθησης (αποτυχία F-010)

¹⁹ είτε λόγω προβλήματος, είτε λόγω εντολής

3. Οι ορισμοί παρακολούθησης δεν μεταβάλλονται μέχρι να διορθωθεί η βλάβη, οπότε και επιστρέφουν στην κατάσταση OK όπου δεν υπάρχει κανένα πρόβλημα. Εδώ, όπως θα περιγραφεί αργότερα, το σύστημα αποκατέστησε τη βλάβη χωρίς την επέμβαση του χειριστή, και έχει επιστρέψει στην αρχική του κατάσταση.

PMON ID	Monitored Parameter	Validity condition	Interval	PMON Status	Repetition #	Check	Acquisition
0	Temp. 1 Status	Temp. 1 0  500 ms	500 ms	Unexpected_Value	2	x = Nominal  Temp1_Timeout_Temporary	2021-06-25T19:12:56
1	Temp. 2 Status	Temp. 1 0  500 ms	500 ms	OK	2	x = Nominal  Temp2_Timeout_Temporary	2021-06-25T19:12:56
2	Temp. 1 Status	Temp. 1 0  500 ms	500 ms	OK	5	x = Nominal  Temp1_Timeout_Permanent	2021-06-25T19:12:56
3	Temp. 2 Status	Temp. 1 0  500 ms	500 ms	OK	5	x = Nominal  Temp2_Timeout_Permanent	2021-06-25T19:12:56
4	Temp. 1	Temp. 1 Status 1  500 ms	500 ms	Invalid	2	-40 ≤ x ≤ 100 Temp1_UnexpectedValue_Temporary Temp1_UnexpectedValue_Temporary	2021-06-25T19:12:56
5	Temp. 2	Temp. 2 Status 1  500 ms	500 ms	OK	2	-40 ≤ x ≤ 100 Temp2_UnexpectedValue_Temporary Temp2_UnexpectedValue_Temporary	2021-06-25T19:12:56
6	Temp. Δ	Temp. 1+2 Status 1  500 ms	500 ms	Invalid	6	-5 ≤ x ≤ 5 TempDelta_UnexpectedValue_Temporary TempDelta_UnexpectedValue_Temporary	2021-06-25T19:12:56
7	Temp. 1	Temp. 1 Status 1  500 ms	500 ms	Invalid	5	-40 ≤ x ≤ 100 Temp1_UnexpectedValue_Permanent Temp1_UnexpectedValue_Permanent	2021-06-25T19:12:56
8	Temp. 2	Temp. 2 Status 1  500 ms	500 ms	OK	5	-40 ≤ x ≤ 100 Temp2_UnexpectedValue_Permanent Temp2_UnexpectedValue_Permanent	2021-06-25T19:12:56
9	Temp. Δ	Temp. 1+2 Status 1  500 ms	500 ms	Invalid	11	-5 ≤ x ≤ 5 TempDelta_UnexpectedValue_Permanent TempDelta_UnexpectedValue_Permanent	2021-06-25T19:12:56
10	Temp. 1	Temp. 1 Status 1  500 ms	500 ms	Invalid	12	-Infinity ≤ x ≤ 50 Subsystem_Overheat_Temp1 Subsystem_Overheat_Temp1	2021-06-25T19:12:56
11	Temp. 2	Temp. 2 Status 1  500 ms	500 ms	OK	12	-Infinity ≤ x ≤ 50 Subsystem_Overheat_Temp2 Subsystem_Overheat_Temp2	2021-06-25T19:12:56

Μετά από τον εντοπισμό της αποτυχίας, δημιουργείται το αντίστοιχο γεγονός και αποστέλλεται στους χειριστές (Σχήμα 5.23).

■■■■■ 2021-06-25 19:12:56.369 UTC ECSS high severity anomaly 100 Temp1_Timeout_Temporary ECSS_HiSev_Anomaly ECSS_Event_Parser

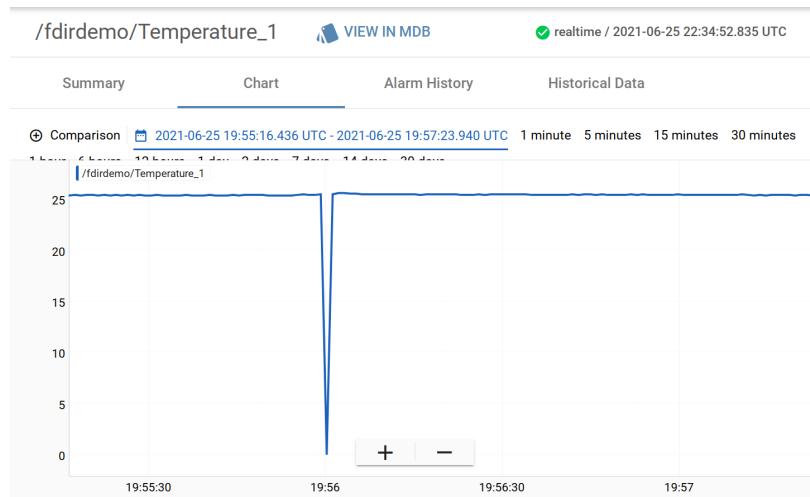
Με την αποστολή του γεγονότος, πυροδοτείται ο μηχανισμός γεγονότος-αντίδρασης. Η αντίδραση που συνδέεται με την προσωρινή αποτυχία του αισθητήρα είναι η κλίση της συνάρτησης restart_sensor²⁰ με όρισμα 1, που σηματοδοτεί την επανεκκίνηση του 1^{ου} αισθητήρα. Στους ανθρώπινους παρατηρητές, η ενδεικτική λυχνία αναβοσβήνει, καθώς το σύστημα επανεκκίνησε την τροφοδοσία του αισθητήρα. Αυτή ήταν η πρώτη προσπάθεια για επιδιόρθωση του προβλήματος.

Σε αυτό το σημείο, στα πλαίσια του πειράματος, ο εκτελεστής του πειράματος επανασυνδέει τον απομακρυσμένο αισθητήρα, προσμοιώνοντας έτσι την "επιδιόρθωση" του προβλήματος. Οι προσπάθειες επικοινωνίας με τον αισθητήρα είναι πετυχημένες, και το σύστημα θεωρεί πως η βλάβη έχει αποκατασταθεί.

Σχήμα 5.22: Πλήρης λίστα των ορισμένων παρακολούθησης κατά την αποτυχία F-010, όπως φαίνονται στην διεπαφή PUS. Ο ορισμός θ που περιλαμβάνει την αποτυχία του αισθητήρα 1 έχει ενεργοποιηθεί, ενώ άλλοι ορισμοί που εξαρτώνται από τον αισθητήρα είναι απενεργοποιημένοι (Invalid).

Σχήμα 5.23: Το γεγονός προσωρινής αποτυχίας του αισθητήρα όπως φαίνεται στο YAMCS

²⁰ Υπηρεσία ST[08] function management



```

3622076 [debug] T [T1]: 25.56
3622176 [debug] T [T2]: 25.56
3622206 [error] Error in TWI: 1
3622206 [debug] T [T1]: 0.00
3622307 [debug] T [T2]: 25.56
3622336 [error] Error in TWI: 1
3622336 [debug] T [T1]: 0.00
3622438 [debug] T [T2]: 25.63
3622467 [error] Error in TWI: 1
3622467 [debug] T [T1]: 0.00
3622500 [error] Monitoring status 4 changed from 0 to 2
3622500 [error] Monitoring status 6 changed from 0 to 2
3622501 [error] Monitoring status 7 changed from 0 to 2
3622502 [error] Monitoring status 9 changed from 0 to 2
3622502 [error] Monitoring status 10 changed from 0 to 2
3622503 [trace] New TM [3,25]
Received TM [25]
3622569 [debug] T [T2]: 25.63
3622597 [error] Error in TWI: 1
3622597 [debug] T [T1]: 0.00
3622655 [trace] New TM [12,12]
Received TM [82]
3622700 [debug] T [T2]: 25.56
3622728 [error] Error in TWI: 1
3622728 [debug] T [T1]: 0.00
3622831 [debug] T [T2]: 25.56
3622859 [error] Error in TWI: 1
3622859 [debug] T [T1]: 0.00
...

```

Σχήμα 5.24: Προβολή της παραμέτρου θερμοκρασίας στη διάρκεια μιας αποτυχίας. Η τιμή είναι φυσιολογική, πέφτει στους 0 °C για 1 δευτερόλεπτο, και επανέρχεται μετά την επιδιόρθωση.

Σχήμα 5.25: Διαγνωστική έξοδος του μικροελεγκτή κατά την αποτυχία F-010. Μόλις το καλώδιο αποσυνδεθεί, ο μικροελεγκτής εντοπίζει το σφάλμα στον δίαυλο I²C, και η τιμή της αναγνώσματος θερμοκρασίας γίνεται υποδεικνύτη. Η κατάσταση των ορισμένων παρακούμησης αλλάζει στον αντίστοιχο χτύπο ρολογιού. Ταυτόχρονα, δημιουργούνται οι τηλεμετρίες ραδιοφάρου ([3,25]) και αναφοράς των μεταβάσεων (ST[12,12]). Τα διαγνωστικά αυτά δεδομένα παράγονται σε μορφή κειμένου από το μικροελεγκτή και δεν θα είναι διαθέσιμα σε τροχιά.

F-020: Μόνιμη απώλεια λειτουργικότητας

Αυτή η βλάβη είναι παρόμοια με την F-010, αλλά υποθέτει πως το εξάρτημα δεν δύναται να επιδιορθωθεί, ακόμα και μετά την επανεκκίνηση.

Διαδικασία

1. Αποσυνδέουμε το καλώδιο SDA ή SCL από το breadboard

Οπτική ένδειξη Η 1^η ενδεικτική λυχνία αναβοσβήνει μια φορά, παραμένει ανοιχτή, και μετά σβήνει μόνιμα.

Χρόνος (s)	PMON	Παράμετρος	Μετάβαση
0	4	Θερμ. 1 = 0	OK → Invalid
0	6	Θερμ. Δ = 29.9375	OK → Invalid
0	7	Θερμ. 1 = 0	OK → Invalid
0	9	Θερμ. Δ = 29.9375	OK → Invalid
0	10	Θερμ. 1 = 0	OK → Invalid
1	0	Κατάστ. Θερμ. 1 = TIMEOUT	OK → Unexpected_Value
2.5	2	Κατάστ. Θερμ. 1 = TIMEOUT	OK → Unexpected_Value
Απομόνωση βλάβης			
4	0	Κατάστ. Θερμ. 1 = DISABLED	Unexpected_Value → OK
5.5	2	Κατάστ. Θερμ. 1 = DISABLED	Unexpected_Value → OK

Αποτελέσματα Στη δοκιμή αυτή το σύστημα αναγνωρίζει κανονικά την αποτυχία του αισθητήρα, και, όπως φαίνεται στον [Πίνακα 5.15](#), προσπαθεί να την διορθώσει δύο φορές:

Πίνακας 5.15: Πίνακας μετάβασης ορισμών παρακολούθησης (αποτυχία F-020)

1. ($t = 1\text{ s}$) Όπως και στην αποτυχία F-010, γίνεται μία επανεκκίνηση του αισθητήρα. Μετά την επανεκκίνηση, η παράμετρος **Κατάσταση Θερμοκρασία 1** συνεχίζει να έχει μια αναμενόμενη τιμή, οπότε η διαδικασία του FDIR δεν διακόπτεται.
2. ($t = 2.5\text{ s}$) Ο ορισμός παρακολούθησης 2 έχει μετρήσει την κατάσταση του αισθητήρα 5 φορές (repetition number), χωρίς καμία επιτυχία. Το σύστημα θεωρεί πως η βλάβη δεν μπορεί να διορθωθεί, και **απενεργοποιεί** πλήρως την τροφοδοσία του αισθητήρα.

Με αυτήν τη μέθοδο, η βλάβη απομονώνεται και δεν μπορεί να επηρεάσει το υπόλοιπο σύστημα. Όλοι οι ορισμοί του FDIR που συνδέονται με τον αισθητήρα είναι απενεργοποιημένοι. Ο σταθμός βάσης γνωρίζει μέσω τηλεμετρίας την κατάσταση του αισθητήρα, και δεν λαμβάνει υπ' όψιν τις λανθασμένες θερμοκρασιακές μετρήσεις του.

Σε αυτό το σημείο, ο ανενεργός αισθητήρας μπορεί να επαναφερθεί

μόνο με κάποια τηλεεντολή, ή με επανεκκίνηση του συστήματος.

Μέσα από τη διεπαφή του YAMCS, ο χρήστης μπορεί να στείλει την εντολή `/fdirdemo/Set_Temp1_Status(Temperature_Status: "Nominal")`

και ο αισθητήρας να ενεργοποιηθεί ξανά.²¹ Αν ο αισθητήρας συνεχίσει να είναι αποσυνδεμένος, παρατηρούμε πως η διαδικασία FDIR επαναλαμβάνεται όπως αναμένεται, και απενεργοποιεί ξανά την τροφοδοσία.

Γενικότερα, παρατηρούμε πως η υπηρεσία του FDIR δεν προκαλεί κάποιον υποβιβασμό του υπόλοιπου συστήματος, καθώς εκτελείται ως ξεχωριστή διεργασία του RTOS.

²¹ Εσωτερικά, αυτή η εντολή καλεί την υπηρεσία `ST[20] parameter management`, και θέτει την παράμετρο Κατάσταση Θερμοκρασία 1 σε τιμή NOMINAL

F-040: Προσωρινή αλλαγή τιμής

Αυτή η βλάβη απαιτεί την αλλαγή της τιμής της θερμοκρασίας για ένα χρονικό διάστημα. Καθώς η θερμοκρασία πρέπει να πέσει κάτω των -40°C ή να ανέβει άνω των 100°C για να πυροδοτηθεί ο μηχανισμός FDIR (Πίνακας 5.9), η μόνη πρακτική μέθοδος για προσομοίωση αυτής της αποτυχίας είναι η **εισαγωγή μιας φεύγτικης τιμής μέσω λογισμικού**.

Ο τρόπος με τον οποίο εισάγουμε αναλογίες τιμών είναι **ένα από τα δύο μπουτόν** της αναπτυξιακής πλακέτας.²² Πιο συγκεκριμένα, πατώντας το *button 0*, στην θερμοκρασία 1 προστίθεται η τιμή 80°C . Πατώντας το *button 1* γίνεται το αντίστοιχο για τη θερμοκρασία 2.

²² Εναλλακτικά, μπορούν να χρησιμοποιηθούν δοκιμαστικές πιλεεντολές

Διαδικασία

1. Κρατάμε πατημένο το *button 0* για μόνο 2 δευτερόλεπτα.

Οπτική ένδειξη Η 1^η ενδεικτική λυχνία αναβοσβήνει μία φορά, και παραμένει ανοιχτή.

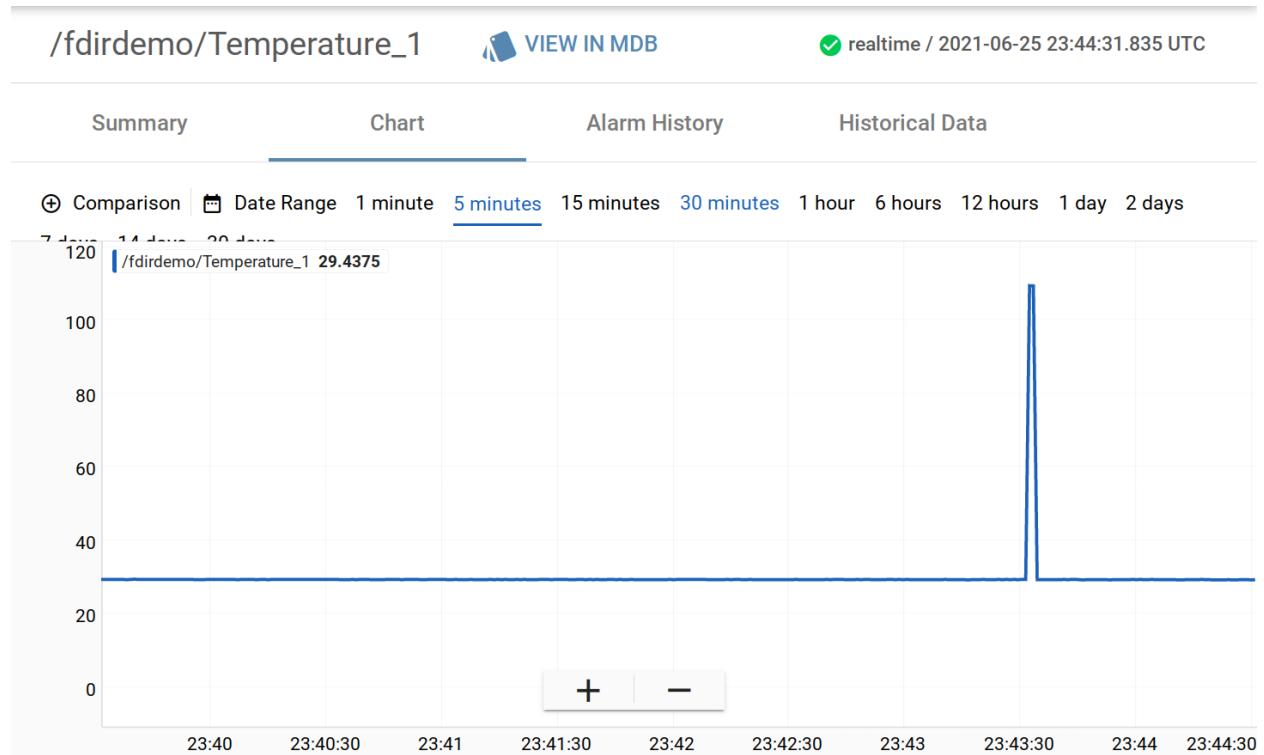
Χρόνος (ms)	PMON	Παράμετρος	Μετάβαση
0	4	Θερμ. 1 = 109.4375	0K → Above_High_Limit
			Αποκατάσταση βλάβης
1.5	4	Θερμ. 1 = 29.5625	Above_High_Limit → 0K

Αποτελέσματα Στο παρόν παράδειγμα, μόνο ένας ορισμός παρακολούθησης βγαίνει εκτός ορίων λόγω της λάθος τιμής. Με το που η έξοδος από τα όρια εντοπιστεί, πραγματοποιείται επανεκκίνηση του αισθητήρα, και η βλάβη αποκαθίσταται.

Πίνακας 5.16: Πίνακας μετάβασης ορισμών παρακολούθησης (αποτυχία F-040)

Η λογική που χρησιμοποιήσαμε στην αρχιτεκτονική του FDIR σημαίνει πως οι υπόλοιποι ορισμοί συνεχίζουν να παρακολουθούν την τιμή της θερμοκρασίας του αισθητήρα, χωρίς να λαμβάνουν τιμή **Invalid**. Για να αποφευχθεί τυχόν σύγκρουση μεταξύ των ορισμών, θα πρέπει ο αριθμός επαναλήψεων των μετρήσεών τους να είναι μεγαλύτερος από αυτόν που πυροδοτήθηκε τώρα. Πράγματι, στο πείραμα αυτό, δεν μεταβλήθηκε κανένας άλλος ορισμός πριν και μετά τη διόρθωση του σφάλματος.²³

²³ Εναλλακτικά, ο έλεγχος εγκυρότητας των υπόλοιπων ορισμών μπορεί να βεβαιώνεται πως η τιμή δεν παραβιάζει τα όρια κάποιου άλλου ορισμού.



Σχήμα 5.26: Προβολή της παραμέτρου θερμοκρασίας στη διάρκεια της αποτυχίας F-040. Φαίνεται η υπερβολικά μεγάλη τιμή που εισήγαγε το λογισμικό.

F-050: Μόνιμη αλλαγή τιμής

Αυτή η βλάβη απαιτεί είναι αντίστοιχη της F-040, χωρίς δυνατότητα επιδιόρθωσης του αισθητήρα.

Διαδικασία

- Κρατάμε πατημένο το *button 0* μέχρι να απενεργοποιηθεί η ενδεικτική λυχνία του 1^{ου} αισθητήρα.

Οπτική ένδειξη Η 1^η ενδεικτική λυχνία αναβοσβήνει μια φορά, παραμένει ανοιχτή, και μετά σβήνει μόνιμα.

Χρόνος (ms)	PMON	Παράμετρος	Μετάβαση
0	4	Θερμ. 1 = 109.4375	OK → Above_High_Limit
1.5	7	Θερμ. 1 = 109.4375	OK → Above_High_Limit
Απομόνωση βλάβης			
1.5	9	Θερμ. Δ = -79.9375	OK → Invalid
1.5	10	Θερμ. 1 = 109.4375	OK → Invalid
2	4	Θερμ. 1 = 109.4375	Above_High_Limit → Invalid
2	6	Θερμ. Δ = -79.9375	OK → Invalid
2	7	Θερμ. 1 = 109.4375	Above_High_Limit → Invalid

Αποτελέσματα Η διαδικασία FDIR εδώ ακολουθεί ξανά δύο στάδια:

- Μετά τον πρώτο εντοπισμό (ορισμός 4), γίνεται επανεκκίνηση του αισθητήρα, και η ενδεικτική λυχνία αναβοσβήνει. Παρ' όλα αυτά, το πρόβλημα δεν αποκαθίσταται.
- Μετά από 1.5 s επιπλέον δευτερόλεπτα, αναλαμβάνει δράση ο ορισμός 7. Αφού ο αισθητήρας έχει ακόμα μία αφύσικη τιμή θερμοκρασίας, η αντίδραση είναι ο αισθητήρας να απενεργοποιηθεί. Έχει ξεκινήσει έτσι η απομόνωση της αποτυχίας.

Πίνακας 5.17:
μετάβασης
παρακολούθησης (αποτυχία F-050)

Μετά από την απομόνωση, όπου η κατάσταση του περιφερειακού τίθεται σε DISABLED, οι ορισμοί που εξαρτώνται από αυτό το περιφερειακό απενεργοποιούνται (**Invalid**). Όπως είναι αναμενόμενο, η απομόνωση έχει ολοκληρωθεί.

Τα βήματα διόρθωσης και απομόνωσης είναι αντίστοιχα με αυτά της αποτυχίας F-020, και απαιτείται ξανά επέμβαση από τους χειριστές για οποιαδήποτε άλλη κίνηση.

F-040 και F-050: Διαφορική μέτρηση

Οι αποτυχίες F-040 και F-050 έχουν έναν δεύτερο τρόπο εντοπισμού: Τη μεγάλη διαφορά μεταξύ μετρήσεων των 2 αισθητήρων. Δεδομένου του μικρού κατωφλίου που ορίζεται στον **Πίνακα 5.9** ($|\Delta| \leq 5^{\circ}\text{C}$), η δοκιμή αυτή γίνεται στο υλικό, θερμαίνοντας έναν από τους 2 αισθητήρες.

Διαδικασία

- Θερμαίνομε έναν αισθητήρα με κάποιο θερμαντικό σώμα. Φροντίζουμε ώστε να μην θερμανθεί πολύ ($< 50^{\circ}\text{C}$), και η θερμότητα να είναι εστιασμένη μόνο σε αυτόν.

Οπτική ένδειξη (προσωρινή αποτυχία) Και οι δύο ενδεικτικές λυχνίες αναβοσβήνουν ταυτόχρονα μια φορά, και παραμένουν ανοιχτές.

Οπτική ένδειξη (μόνιμη αποτυχία) Και οι δύο ενδεικτικές λυχνίες αναβοσβήνουν ταυτόχρονα μια φορά, παραμένουν ανοιχτές, και μετά σβήνουν μόνιμα.

Χρόνος (ms)	PMON	Παραμέτρος	Μετάβαση
0	6	Θερμ. $\Delta = -6.1875$	OK → Below_Low_Limit Αποκατάσταση βλάβης
7.5	6	Θερμ. $\Delta = -4.8125$	Below_Low_Limit → OK

Αποτελέσματα Οι ενέργειες διόρθωσης που σημειώνονται στην διαφορική περίπτωση είναι παρόμοιες με αυτές τις απλής περίπτωσης (F-040, F-050), με τη διαφορά πως δεν είναι γνωστό ποιος από τους 2 αισθητήρες έχει αποτύχει. Η προσέγγιση που ακολουθεί η αρχιτεκτονική εδώ είναι *fail-safe*, και απενεργοποιεί και τους 2 αισθητήρες. Ως συνέπεια, οι 7 ορισμοί παρακολούθησης που εξαρτώνται από θερμοκρασιακές τιμές απενεργοποιούνται (**Invalid**).

Πίνακας 5.18: Πίνακας μετάβασης ορισμών παρακολούθησης (προσωρινή αποτυχία F-040, διαφορική)

Χρόνος (ms)	PMON	Παράμετρος	Μετάβαση
0	6	Θερμ. Δ = -5.8125	OK → Below_Low_Limit
2.5	9	Θερμ. Δ = -6	OK → Below_Low_Limit
Απομόνωση βλάβης			
2.5	10	Θερμ. 1 = 38.3125	OK → Invalid
2.5	11	Θερμ. 2 = 32.3125	OK → Invalid
3	4	Θερμ. 1 = 38.3125	OK → Invalid
3	5	Θερμ. 2 = -32.3125	OK → Invalid
3	6	Θερμ. Δ = -6	OK → Invalid
3	7	Θερμ. 1 = 38.3125	OK → Invalid
3	8	Θερμ. 2 = 32.3125	OK → Invalid
3	9	Θερμ. Δ = -6	OK → Invalid

F-060: Κολλημένος ακροδέκτης εξόδου I²C Αυτή η αποτυχία προσομοιώνεται με σύνδεση υλικού. Σε τέτοιες περιπτώσεις πρέπει να δίνεται προσοχή ώστε να μη γίνονται συνδέσεις (π.χ. βραχυκυκλώματα) που θα βλάψουν το δοκιμαστικό σύστημα, ειδικά αν δεν υπάρχουν κατάλληλα κυκλώματα προστασίας.

Διαδικασία

- Συνδέουμε έναν από τους ακροδέκτες SDA, SCL στη γείωση με ένα επιπλέον καλώδιο.²⁴

Αποτελέσματα Παρατηρούμε πως τα αποτελέσματα είναι ακριβώς ίδια με αυτά τις αποτυχίας F-020 (μόνιμη απώλεια λειτουργικότητας). Αυτό είναι φυσιολογικό, καθώς ουσιαστικά καταστρέφουμε τον δίαιυλο I²C του περιφερειακού.

F-070 μέχρι F-120: Αισθητήρας 2

Τα αποτελέσματα των δοκιμών FDIR για τον 2^ο αισθητήρα είναι ίδια με αυτά του 1^{ου} αισθητήρα που παρουσιάστηκαν παραπάνω, καθώς τα εξαρτήματα έχουν ίδια συμπεριφορά και σχεδιασμό FDIR.

Σε αυτό το σημείο σημειώνεται πως ακόμα κι αν ο σχεδιασμός και το λογισμικό είναι πανομοιότυπα, σε περιπτώσεις σαν και αυτήν υπάρχει νόημα επανάληψης των δοκιμών για τα επιπλέον εξαρτήματα, καθώς συνεχίζει να υπάρχει η πιθανότητα προβλημάτων υλικού, λάθος συνδέσεων, ατυχών ρυθμίσεων και bugs.

Πίνακας 5.19: Πίνακας μετάβασης ορισμών παρακολούθησης (μόνιμη αποτυχία F-050, διαφορική)

²⁴ Καθώς το I²C λειτουργεί με βάση τη λογική ανοιχτής εκροής (open drain), η σύνδεση στη γείωση δεν θα προκαλέσει πρόβλημα στα πλεκτρονικά.

F-130: Υπερθέρμανση υποσυστήματος Στη δοκιμή αυτή δεν μελετάμε την αποτυχία ενός από τους αισθητήρες, αλλά την αποτυχία του ίδιου του υποσυστήματος, που υποθέτουμε πως έχει υπερθερμανθεί. Για να διευκολύνουμε την επαναληψιμότητα τις δοκιμής, προσθέσαμε ξανά μία εικονική θερμοκρασία μέσω λογισμικού.

Διαδικασία

- Καλούμε μέσω TC τη συνάρτηση `bogus_temp` που προκαλεί μία σταδιακή άνοδο της θερμοκρασίας μέχρι τους 90°C .

Σε περιπτώσεις δύσκολων δοκιμών, μπορούν να γίνουν τροποποιήσεις στους ορισμούς της υπηρεσίας ώστε να διευκολυνθεί το τεστ. Παρ' όλα αυτά, συνιστάται το σύστημα να βρίσκεται όσο πιο κοντά γίνεται στο πραγματικό, έτσι ώστε οι συνθήκες δοκιμής να συμπίπτουν με τις συνθήκες πτίσεις. Ακόμα, πρέπει να λαμβάνονται υπ' όψιν ο αλληλοεπηρεασμός των διαφορετικών ορισμάν.

Οπτική ένδειξη Το LED κατάστασης του μικροελεγκτή σταματά να αναβοσβήνει για λίγο, και έπειτα επανέρχεται. Τα LED των αισθητήρων αναβοσβήνουν μία φορά.

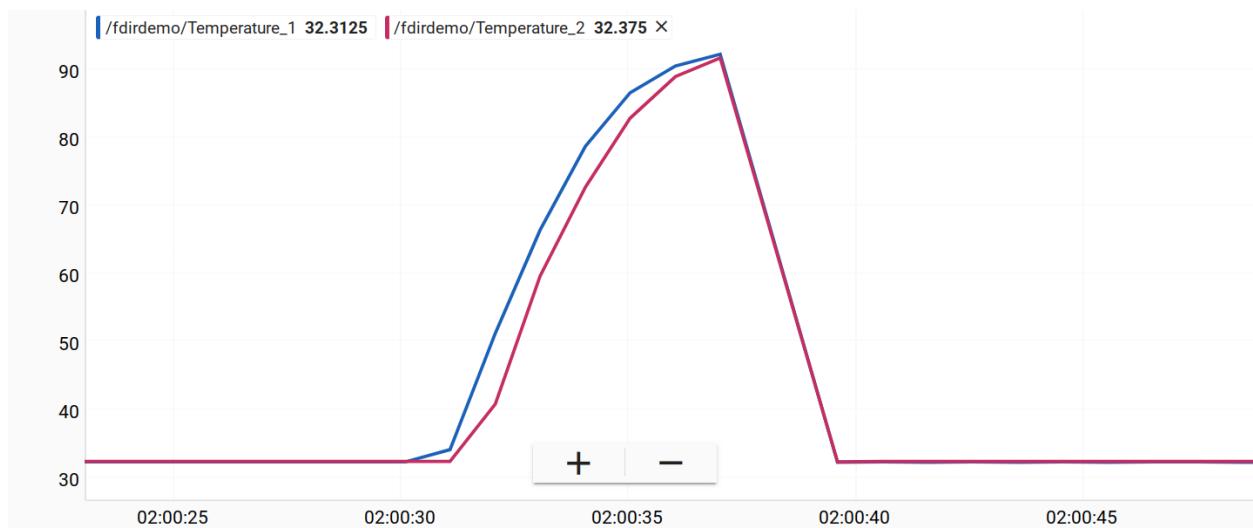
Χρόνος (ms)	PMON	Παράμετρος	Μετάβαση
0	10	Θερμ. 1 = 92.288	OK → Above_High_Limit
Αυτόνομη πανεκκίνηση μικροελεγκτή			

Αποτελέσματα Εδώ ενεργοποιείται ο ορισμός παρακολούθησης που συνδέεται με τον αισθητήρα που έφτασε στους 50°C . Μόλις το σύστημα περάσει αρκετή ώρα σε αυτήν τη θερμοκρασία, γίνεται άμεσα μία επανεκκίνηση του μικροελεγκτή. Η επανεκκίνηση αυτή θα συνεχίσει να γίνεται, μέχρι η θερμοκρασία να επανέλθει.

Η αποτυχία αυτή ουσιαστικά ανήκει στο επίπεδο υποσυστήματος και όχι υλικού που μελετούσαμε προηγουμένως. Σημειώνουμε επίσης πως, λόγω του μεγάλου χρονικού διαστήματος μεταξύ των επανεκκινήσεων, οι χειριστές έχουν αρκετό χρόνο στη διάθεσή τους ώστε να απενεργοποιήσουν το FDIR ή να λάβουν και να επεξεργαστούν την τηλεμετρία.

Εναλλακτικά, μπορούμε να χρησιμοποιήσουμε ένα πιστόλι θερμού αέρα ή κάποια άλλη πηγή θερμότητας ώστε να θερμάνουμε και τους δύο αισθητήρες στους 50°C . Φροντίζουμε ώστε η αύξηση της θερμοκρασίας να είναι ομοιόμορφη, για να μην ενεργοποιηθεί ο έλεγχος διαφοράς.

Πίνακας 5.20: Πίνακας μετάβασης ορισμών παρακολούθησης (αποτυχία F-130)



Σχήμα 5.27: Οι μετρήσεις θερμοκρασίας στις οποίες έχουν προστεθεί εικονικές τιμές, μέσα από το YAMCS. Η θερμοκρασία φτάνει μέχρι τους 90°C , μέχρι που ο μικροελεγκτής κάνει επανεκκίνηση, και η θερμοκρασία επανέρχεται σε κανονικό επίπεδο

5.4.3 Τροποποίηση του FDIR σε πραγματικό χρόνο

Η δύναμη της προσέγγισης FDIR του SAVOIR έγκειται στο ότι ολόκληρη η δομή FDIR μπορεί να τροποποιηθεί με ασφάλεια μέσω τηλεεντολών, χωρίς να απαιτούνται τροποποιήσεις ή επαναφοράς της του κάθικα και του λογισμικού. Η εφαρμογή του μοντέλου υπηρεσιών ECSS επιτρέπει την προσαρμογή όλων των υπηρεσιών και λειτουργιών FDIR, χρησιμοποιώντας προκαθορισμένες δομές τηλεεντολών. Σε αυτή την ενότητα, θα εκτελεστούν εντολές ECSS για την επίδειξη αυτών των δυνατοτήτων και του ελέγχου τους από τον σταθμό βάσης.

Γενικότερα, το επίπεδο της δυνατότητας αναδιαμόρφωσης που παρέχει το σύστημα καθορίζεται από τις απαιτήσεις που παρατίθενται στα [36, 52]. Η προσαρμοσμένη έκδοση που χρησιμοποιήθηκε για τα πειράματά μας παρατίθεται στον [Πίνακα 5.21](#).

#	Προδιαγραφή	Υπηρεσία	Λεπτομέρειες
RC-010	Οι αναφορές FDIR πρέπει να περιέχουν όλες τις πληροφορίες για την ανάλυση αποτυχιών (π.χ. ώρα εμφάνισης, παραμέτροι εκτός ορίων, πραγματοποιημένες μεταγωγές).	ST[12] on-board monitoring	Μέσω περιοδικών αναφορών μεταγωγής
RC-020	Πρέπει να παρέχεται η δυνατότητα ενεργοποίησης και απενεργοποίησης οποιασδήποτε λειτουργίας FDIR μέσω τηλεεντολών.	ST[12] on-board monitoring	Με εντολές <i>enable/disable parameter monitoring definitions</i>
RC-030	Πρέπει να παρέχεται η δυνατότητα προσθήκης και διαγραφής παραμέτρων από τον κατάλογο παρακολούθησης.	ST[12] on-board monitoring	Με εντολές <i>add/delete parameter monitoring definitions</i>
RC-040	Πρέπει να παρέχεται η δυνατότητα τροποποίησης των ενεργειών αποκατάστασης βλάβης που συνδέονται με κάθε ορισμό παρακολούθησης.	ST[19] event-action	Με εντολές <i>add, delete, enable και disable event-action definitions</i>

Η πλατφόρμα YAMCS χρησιμοποιήθηκε ξανά για τη δημιουργία των επιδυμητών εντολών ([Σχήμα 5.28](#)). Παρακάτω, θα αναλυθεί ο τρόπος με τον οποίο μπορεί να επιτευχθεί η κάθε μία από τις παραπάνω απαιτήσεις

RC-010 Παρατηρησιμότητα FDIR

Ο βαθμός παρατηρησιμότητας που απαιτείται, σε πραγματικό και μη χρόνο, εξασφαλίζεται μέσω της διεπαφής που παρουσιάστηκε στην [Ενότητα 5.2.5](#). Οι έτοιμες δομές τηλεμετρίας που υλοποιούνται από το πρότυπο PUS [35] είναι:

Πίνακας 5.21: Απαιτήσεις αναδιαμόρφωσης FDIR στην πειραματική διάταξη

[35] ECSS Secretariat, *ECSS-E-ST-70-41C – Telemetry and Telecommand Packet Utilization*.

Send a command		
Name	Significance	Description
↳ Log		Output a string via UART
↳ MCU_Reset		Perform a soft reset of the MCU
↳ OnBoardMonitoring_Disable		Disable the entire on-board monitoring function
↳ OnBoardMonitoring_Enable		Enable the entire on-board monitoring function
↳ Sensor1_Restart		Restart Sensor 1
↳ Sensor2_Restart		Restart Sensor 2
↳ Set_Temp1_Status		Set the status of temperature sensor 1
↳ Set_Temp2_Status		Set the status of temperature sensor 2
↳ ST03_Disable		Disable the generation of housekeeping reports
↳ ST03_Enable		Enable the generation of housekeeping reports
↳ ST03_SetInterval		Set the generation interval of housekeeping reports
↳ ST12_Add_Definition_Expected_Value		Add a parameter monitoring definition: Expected value, uint64_t type
↳ ST12_Add_Definition_Limit		Add a parameter monitoring definition: Limit check, float type
↳ ST12_Delete_Definition		Delete a parameter monitoring definition
↳ ST12_Disable_Definition		Disable one on-board parameter monitoring definition
↳ ST12_Enable_Definition		Enable one on-board parameter monitoring definition
↳ ST12_ListAllDefinitions		Report the list of all parameter monitoring definitions
↳ ST19_Add_Definition		Create one event-action definition
↳ ST19_Delete_Definition		Delete one event-action definition
↳ ST19_Disable		Disable the event-action function
↳ ST19_Disable_Definition		Disable one event-action definition
↳ ST19_Enable		Enable the event-action function
↳ ST19_Enable_Definition		Enable one event-action definition
↳ ST19_ListAllEventAction		Report the list of all event-action definitions
↳ ST19_ListEventActionRequest		Show an event-action definition with TC requests
↳ Test		-
↳ TestError		An erroneous command to test parsing errors

- TM[3,25]: Περιοδική αναφορά τιμών παραμέτρων (ραδιοφάρος)
- TM[5,1] μέχρι TM[5,4]: Αναφορά γεγονότων όταν συμβαίνουν
- TM[12,9]: Λίστα των ορισμών παρακολούθησης
- TM[12,12]: Περιοδική αναφορά μεταβάσεων κατάστασης σε ορισμούς παρακολούθησης²⁵
- TM[19,7] και TM[19,11]: Λίστα των ορισμών γεγονότος-αντίδρασης
- TM[20,2]: Αναφορά της τιμής οποιασδήποτε παραμέτρου του συστήματος

Με το παραπάνω σύνολο τηλεμετρίας, που κατεβαίνει είτε αυτόνομα είτε μετά από εντολή, οι χειριστές έχουν τη δυνατότητα να μελετήσουν την κατάσταση του συστήματος και του FDIR με λεπτομέρεια. Όλα τα αποτελέσματα στην [Ενότητα 5.4.2](#) εξάχθηκαν χρησιμοποιώντας τις παραπάνω δομές τηλεμετρίας.

RC-020 Ενεργοποίηση/απενεργοποίηση FDIR

Οι δυνατότες που υλοποιήθηκαν και δοκιμάστηκαν για τον έλεγχο του FDIR είναι οι εξής:

- TC[12,16]: *disable the parameter monitoring function*

Αυτή η τηλεεντολή ορίζεται στο πρότυπο PUS και επιτρέπει την

Σχήμα 5.28: Λίστα τηλεεντολών στο περιβάλλον του YAMCS. Αυτές οι εντολές μπορούν να εκκινήσουν διαδικασίες στον δορυφόρο, να ενεργοποιήσουν/απενεργοποιήσουν αισθητήρες, ή να αλλάξουν τις ρυθμίσεις των εξαρτημάτων του δορυφόρου. Η λίστα αυτών των εντολών παρέχεται στο YAMCS μέσω των αρχείων XTCE ([Παράρτημα B](#)).

²⁵ Πιο απλά, ενημερωνόμαστε πότε μια παράμετρος βγαίνει εκτός ορίων, και πότε επανέρχεται σε κανονική κατάσταση

πλήρη απενεργοποίηση της παρακολούθησης παραμέτρων, άρα, κατ' επέκτασιν, την **πλήρη απενεργοποίηση του FDIR**. Μετά την αποστολή της, οποιαδήποτε αποτυχία στο σύστημα δεν γίνεται αντιληπτή, και δεν εκτελείται καμία πράξη ανάκαμψης.

Γίνεται αντιληπτό πως μια τέτοια εντολή είναι αρκετά επικίνδυνη για το σύστημα του δορυφόρου, καθώς μία λάθος εκτέλεσή της θα μειώσει κατά πολύ τη δυνατότητα ανάκαμψης από μία βλάβη.²⁶ Παρ' όλα αυτά, είναι χρήσιμη σε περίπτωση που το FDIR εμφανίζει προβλήματα.

Εδώ σημειώνουμε πως αυτή η εντολή δεν αποτρέπει τα ερωτήματα κατάστασης και τις επιπλέον ρυθμίσεις στην υπηρεσία παρακολούθησης.

- TC[12,15]: enable the parameter monitoring function

Η τηλεεντολή αυτή επιτρέπει την επαναλειτουργία της παρακολούθησης παραμέτρων, αν απενεργοποιήθηκε προηγουμένως.

- TC[12,1]: enable parameter monitoring definitions

Πέρα από απενεργοποίηση ολόκληρης της λειτουργίας FDIR, το πρότυπο δίνει τη δυνατότητα απενεργοποίησης (μαζικής ή μη) μόνο συγκεκριμένων κομματιών του.

Send a command

realtime / 2021-06-26 01:46:45.213 UTC

1. Select command 2. Configure command 3. View report

Command	ST12_Disable_Definition
Space system	/fdirdemo
Description	Disable one on-board parameter monitoring definition
Significance	■■■■■

Arguments

PMON_ID	integer
2	

Options

Comment (optional)

Cancel **> Send**

²⁶ Για αυτό προτείνεται σε κρίσιμες διαδικασίες να απαιτείται να στέλνονται δύο εντολές σε σειρά (λογική ARM-EXECUTE)

Σχήμα 5.29: Προβολή της οθόνης εκτέλεσης εντολής του YAMCS. Ο χρήστης καλείται να εισάγει το ένα όρισμα της εντολής TC[12,1], που είναι ο ορισμός παρακολούθησης που επιθυμεί να απενεργοποιήσει.

Αυτή η δυνατότητα είναι ιδιαίτερα χρήσιμη, ειδικότερα όταν φαίνεται ότι το FDIR είναι υπερβολικά αυστηρό για κάποιο εξάρτημα

- TC[12,2]: *disable parameter monitoring definitions*
- TC[19,5]: *disable event-action definitions*

Πέρα από την απενεργοποίηση ολόκληρου του FDIR για ένα εξάρτημα, δίνεται η δυνατότητα απενεργοποίησης μόνο του "IR" κομματιού του. Αυτό σημαίνει πως ενώ γίνεται εντοπισμός των αποτυχιών, δεν γίνεται **καμία προσπάθεια για απομόνωση ή επιδιόρθωσή τους**. Με αυτόν τον τρόπο μπορούμε να απενεργοποιήσουμε το FDIR, αλλά να διατηρήσουμε την παρατηρούμένη την παραμέτρων και των ορίων.

Πιο τεχνικά, η εντολή αυτή απενεργοποιεί τους ορισμούς γεγονότος-αντίδρασης για συγκεκριμένα γεγονότα. Η συμπεριφορά μιας τυπικής αποτυχίας αφού έχει εκτελεστεί αυτή η εντολή φαίνεται στον [Πίνακα 5.22](#).

Χρόνος (ms)	PMON	Παράμετρος	Μετάβαση
0	4	Θερμ. 1 = 110.625	OK → Above_High_Limit
1.5	7	Θερμ. 1 = 110.6875	OK → Above_High_Limit
2	6	Θερμ. Δ = -79.9375	OK → Below_Low_Limit
4.5	9	Θερμ. Δ = -79.875	OK → Below_Low_Limit
5	10	Θερμ. Δ = 110.625	OK → Above_High_Limit

- TC[19,4]: *enable event-action definitions*

RC-030 Προσθήκη/διαγραφή ορισμών παρακολούθησης

Η δυνατότητα διαγραφής προσθήκης νέων FDIR ορισμών γίνεται μέσω των τηλεεντολών της υπηρεσίας ST[12] *on-board monitoring*:

- TC[12,5]: *add parameter monitoring definitions*

Μέσω της τηλεεντολής TC[12,5] μπορούμε να παρέχουμε στο δορυφόρο όλα τα δεδομένα που χρειάζονται για να προστεθεί ένας ορισμός παρακολούθησης, για οποιονδήποτε τύπο παραμέτρου, ελέγχου τιμής και ελέγχου εγκυρότητας.

Τα ορίσματα της συγκεκριμένης εντολής μπορούν να δοθούν μέσα από το περιβάλλον του YAMCS.

Σημειώνουμε πως η ευκολία στη χρήση αυτής της εντολής μειώνεται λόγω της ύπαρξης πολλών διαφορετικών τύπων παραμέτρων και ελέγχων, που οδηγούν σε διαφορετική μορφή μηνύματος. Για

Πίνακας 5.22: Πίνακας μετάβασης με απενεργοποιημένη την υπηρεσία event-action. Εδώ δεν απενεργοποιούνται οι αισθητήρες, αλλά οι ορισμοί, ο ένας μετά τον άλλον, βγαίνουν εκτός ορίων.

repeated N times									
N	PMON ID	monitored parameter ID	check validity condition			monitoring interval	repetition number	check type	check type dependent criteria (see below)
			validity parameter ID	mask	expected value				
unsigned integer	enumerated	enumerated	enumerated	bit-string (deduced size)	deduced	unsigned integer	unsigned integer	enumerated	
optional					optional				

NOTE 1 For the check type enumerated values, refer to requirement 8.12.3.1a.
 NOTE 2 In the check validity condition field, the size of the mask field and the format of the expected value field are specific to the validity parameter identified by its parameter ID field.
 NOTE 3 The structure of the check type dependent criteria field is driven by requirement 8.12.2.5c for expected-value-checking, requirement 8.12.2.5e for limit-checking and requirement 8.12.2.5f for delta-checking.

Σχήμα 5.30: Η τηλεεντολή TC[12,5], ακριβώς όπως παρουσιάζεται στο [35]

αυτόν το λόγο, στο παράδειγμα αυτό υλοποιήθηκαν οι τύποι **uint64_t** για ελέγχους ισότητας και **float** για ελέγχους ορίων. Ορισμοί με τους προηγούμενους τύπους ρυθμίστηκαν ώστε να είναι συμβατοί με όλες τις υπόλοιπες αριθμητικές παραμέτρους στον κώδικα του διαστημικού τμήματος.

- TC[12,6]: *delete parameter monitoring definitions*

Η εντολή αυτή μπορεί να διαγράψει ορισμούς παρακολούθησης, αλλά και, σε συνδυασμό με την TC[12,5], να μεταβάλλει ορισμούς παρακολούθησης.

Παρ' όλα αυτά, η μεταβολή μέσω διαγραφής και προσθήτης δεν συνιστάται από το [36], καθώς διατρέχει το ρίσκο να μην εκτελεστεί η μία από τις δύο εντολές, και το σύστημα να βρεθεί σε μία ανεπιθύμητη κατάσταση.

[36] Space Avionics Open interface Architecture, *SAVOIR FDIR Handbook*.

RC-040 Προσθήκη/διαγραφή αντιδράσεων

Αντίστοιχα με τους ορισμούς παρακολούθησης, οι χειριστές μπορούν να συνδέσουν τις κατάλληλες αντιδράσεις με αυτούς τους ορισμούς, εκμεταλλευόμενοι τις συναρτήσεις της υπηρεσίας ST[19] *eventaction*. Σε όλες τις παρακάτω συναρτήσεις, η πρόσβαση στους ορισμούς γίνεται με βάση το αναγνωριστικό της εφαρμογής (application process ID) και του γεγονότος (event ID).

- TC[19,1]: *add event-action definitions*

Σημειώνουμε πως το κομμάτι της αντίδρασης ορίζεται από τον χρήστη ως ένα **πακέτο τηλεεντολής**. Έτσι, δίνοντας μία εμφωλευμένη αυθαιρέτη τηλεεντολή, οι χειριστές μπορούν να καθοδηγήσουν τον δορυφόρο ώστε να εκτελέσει πρακτικά οποιαδήποτε ενέργεια μετά από μία αποτυχία.

- TC[19,2]: *delete event-action definitions*

Command ST12_Add_Definition_Limit
Space system /fdirdemo
Description Add a parameter monitoring definition: Limit check, float type
Significance ■■■■■

Arguments

PMON_ID	integer
Monitored_Parameter_ID	integer
Validity_Parameter_ID	integer
Validity_Mask	integer
Validity_Expected_Value	integer
Monitoring_Interval	milliseconds integer
Repetition_Number	integer
Low_Limit	float
Low_Event	integer
High_Limit	float
High_Event	integer

Options**Comment** (optional)
Cancel**Send**

Σχήμα 5.31: Περιβάλλον προσθήκης ορισμού παρακολούθησης από το YAMCS

6

Συμπεράσματα και Μελλοντική Εργασία

Η διπλωματική εργασία παρουσίασε την αρχιτεκτονική και την υλοποίηση μιας μεθόδου ανίχνευσης, απομόνωσης και αντιμετώπισης βλαβών σε διαστημικά συστήματα χαμπλού κόστους. Στο [Κεφάλαιο 2](#), έγινε μία σύντομη παρουσίαση της αποστολής AcubeSAT και των λεπτομερειών της που συνδέονται άμεσα με την αξιοπιστία του δορυφόρου. Στο [Κεφάλαιο 3](#) έγινε μία παρουσίαση των προτύπων ECSS-E-ST-70-41C και SAVOIR-HB-003, τα οποία προτείνουν μεθόδους για ανάπτυξη ενός δομημένου διαχειριστικού πλαισίου και μίας παραμετροποίησης αρχιτεκτονικής διόρθωσης σφαλμάτων. Παρουσιάστηκαν ακόμα τα 7 βήματα της διαδικασία υλοποίησης του FDIR, ξεκινώντας από τη συγγραφή των προδιαγραφών, συνεχίζοντας με τον σχεδιασμό, και καταλήγοντας στην υλοποίηση και τις δοκιμές αξιοπιστίας. Ακόμα, παρουσιάστηκε η ιδέα του Failure Mode and Effects Analysis (FMEA) και η σύνδεσή της με το λογισμικό. Στο [Κεφάλαιο 4](#), οι προηγούμενες πληροφορίες έγιναν πιο συγκεκριμένες, αφού εφαρμόστηκαν στα σχέδια του νανοδορυφόρου AcubeSAT. Πιο συγκεκριμένα, πρώτα τέθηκαν οι απαιτήσεις του συστήματος FDIR και πραγματοποιήθηκε θεωρητική ανάλυση διαφορετικών αρχιτεκτονικών με σκοπό την αύξηση αξιοπιστίας. Τέλος, παρουσιάστηκαν όλες οι μέθοδοι εντοπισμού, πρόληψης και επιδιόρθωσης βλαβών που είναι διαθέσιμες στον νανοδορυφόρο, και προετοιμάστηκαν τα διαγράμματα φοίνικας που δείχνουν τις συγκεκριμένες ενέργειες του FDIR σε κάθε λειτουργία συστήματος.

Στο [Κεφάλαιο 5](#) εφαρμόστηκαν τα 7 βήματα της διαδικασίας FDIR σε μία μικρογραφία ενός δορυφορικού συστήματος. Το φυσικό σύστημα που περιείχε έναν μικροελεγκτή και 2 αισθητήρες θερμοκρασίας εμπλουτίστηκε με υλοποίησεις των προαναφερθέντων προτύπων, και με μία πλατφόρμα εξομοίωσης του σταθμού βάσης. Ο σχεδιασμός του FDIR βασίστηκε σε ένα πλήρες FMEA του συστήματος, βάσει του οποίου εξήχθησαν οι παρακολουθούμενες παραμετροί, τα κατώφλια επικινδυνότητάς τους, και οι διορθωτικές ενέργειες. Η διαδικασία αυτή πραγματοποιήθηκε για κάθε πιθανή βλάβη του αισθητήριου συστήματος. Στη συνέχεια, κάθε μία από τις βλάβες προσομοιώθηκε σε υλικό ή λογισμικό, και αποδείχθηκε πως το σύστημα μπορούσε να ανακάψει σε κάθε περίπτωση. Τέλος, έγινε επίδειξη της δυνατότητας παραμετροποίησης του FDIR σε τροχιά χωρίς τροποποίηση του λογισμικού.

Το αποτέλεσμα της παραπάνω μελέτης και των δοκιμών δείχνει πως είναι δυνατή η εφαρμογή της μεθόδου SAVOIR για την επιδιόρθωση βλαβών σε έναν νανοδορυφόρο. Η λογική που προτάθηκε μπορεί να υποστηριχθεί από έναν μικροελεγκτή χαμπλών πόρων, και το λογισμικό μπορεί να χρησιμοποιηθεί αυτούσιο για οποιοδήποτε ευάλωτο σύστημα, κάτι που γίνεται δυνατό χάρη στην κλιμακούμενη αρχιτεκτονική που χρησιμοποιήθηκε. Η χρήση μιας τέτοιας δομημένης προσέγγισης κρίνεται πως μπορεί να αυξήσει σε μεγάλο βαθμό τις πιθανότητες επιβίωσης ενός CubeSat σε τροχιά, αλλά και να μειώσει τον χρόνο ανάπτυξης μιας αποστολής, καθώς μπορούν να χρησιμοποιηθούν έτοιμα πρότυπα και λογισμικό. Η βιβλιοθήκης που χρησιμοποιήθηκαν, αν και είναι ανοιχτού κώδικα και διαθέσιμες προς χρήση,

μπορούν ελεύθερα να αντικατασταθούν με αντίστοιχες που προσφέρουν την ίδια λειτουργικότητα.

Στο σημείο αυτό πρέπει να τονιστεί η σημασία της επαλήθευσης και των δοκιμών της υλοποίησης που εφαρμόζεται. Όσο περισσότερο οι συνθήκες των δοκιμών αναπαριστούν τις πραγματικές συνθήκες, τόσο βελτιώνεται η εμπιστοσύνη στους υλοποιημένους μηχανισμούς. Στις δικές μας δοκιμές διαπιστώσαμε έναν σημαντικό αριθμό ελλείψεων, οριακών περιπτώσεων και συνθηκών συναγωνισμού (race conditions) που δεν ήταν προφανείς κατά τον σχεδιασμό και τις απλές δοκιμές, αλλά εμφανίστηκαν άμεσα στις δοκιμές κάθε αποτυχίας ξεχωριστά.

Παρά τις δυνατότητες που δίνονται από ένα τέτοιο σύστημα, σημειώσαμε τις εξής λεπτομέρειες υλοποίησης που χρήζουν προσοχή:

- Μία βιαστική υλοποίηση του FDIR μπορεί να οδηγήσει σε μεγάλη αύξηση της πολυπλοκότητας, και δυσκολία διαχείρισης όλων των ορισμών αξιοπιστίας. Η πολυπλοκότητα αυξάνει την πιθανότητα σφαλμάτων που δεν εντοπίζονται, ανθρώπινου λάθους, αλλά και κακής λειτουργίας του συστήματος όταν αποστέλλονται πολλές εντολές που μπορεί να μην εκτελεστούν. Σε μη κρίσιμα συστήματα, προτείνεται ο αριθμός των ορισμών να περιοριστεί σε έναν διαχειρίσιμο αριθμό. Ενδεικτικά στο παρόδειγμά μας, θα μπορούσαν να αφαιρεθούν οι παρακολουθήσεις της διαφοράς Δ των θερμοκρασιών, αντικαθιστώμενες από έλεγχο ώστε και οι 2 αισθητήρες να είναι εκτός ορίων για την επανεκκίνηση του υποσυστήματος.
- Οι διαφορετικοί τύποι μεταβλητών των παραμέτρων (**uint16_t**, **uint32_t**, **float** κ.ά.) δημιουργούν δυσκολίες στη διαχείριση σε περιβάλλοντα με ασφαλείες τύπους (type safe) όπως αυτό της C++, και ίσως κρίθει απαραίτητη η χρήση πιο "ανασφαλών" δομών.
- Είναι σημαντικό να εξεταστούν οι συσχετίσεις μεταξύ των ελέγχων και να βεβαιωθεί πως ένας έλεγχος δεν προκαλεί την γρηγορότερη ή πιο αργή πυροδότηση κάποιου άλλου (π.χ. λόγω επανεκκίνησης αισθητήρα που επαναφέρει στο 0 όλους τους υπόλοιπους ελέγχους)
- Το πρότυπο ECSS-E-ST-70-41C δεν παρέχει ευελιξία στον ορισμό παραμέτρων, και δεδομένα όπως διαφορά μεταξύ τιμών πρέπει να οριστούν χειροκίνητα στον κώδικα. Επιπλέον, δεν υπάρχει ευελιξία στον ορισμό γεγονότων-αντιδράσεων, καθώς κάθε γεγονός μπορεί να συνδεθεί με μία ρυθμιζόμενη αντίδραση.

6.1 Μελλοντική εργασία

Αυτή η διπλωματική εργασία παρουσίασε μόνον μία μελέτη περίπτωσης και μία πειραματική διάταξη για επαλήθευση ενός υποσυνόλου των λειτουργιών του FDIR. Ως επόμενο βήμα μπορεί να χαρακτηριστεί η συνέχεια της εφαρμογής αυτής της φιλοσοφίας στην κατασκευή του νανοδιορυφόρου, η δοκιμή του FDIR στα υποσυστήματα και το τελικό συναρμολογημένο σύστημα, αλλά κυριότερα, η εξαγωγή αποτελεσμάτων από την τροχιά.

Πέρα από τη συνέχιση της εφαρμογής του FDIR στον νανοδιορυφόρο, μερικοί πιο συγκεκριμένοι στόχοι είναι οι εξής:

- Συνέχιση της υλοποίησης των υπηρεσιών ECSS. Συγγραφή του κώδικα για επιπλέον δυνατότητες, όπως:

- Μεταβολή ορισμών παρακολούθησης
 - Παρακολούθηση πολλών μεταβλητών με συνθήκες OR ή AND ("functional monitoring definitions")
 - Υποστήριξη αυθαίρετων τύπων δεδομένων για παραμέτρους
 - Συγχραφή δοκιμών μονάδας (unit tests) για επαλήθευση σε επίπεδο λογισμικού
- Ολοκλήρωση του λογισμικού Ελέγχου Αποστολής (Mission Control) για τον κεντρικό σταθμό βάσης του δορυφόρου, και ενσωμάτωση με τη βάση δεδομένων των υπηρεσιών ECSS
 - Επέκταση του λεπτομερή σχεδιασμού του FDIR στην Ασφαλή Λειτουργία
 - Σύγκριση με άλλες αρχιτεκτονικές ελέγχου του δορυφόρου και αρχιτεκτονικές FDIR που χρησιμοποιούνται στη βιομηχανία [53, 54]
 - Επέκταση της θεωρητικής μελέτης (Ενότητα 4.2) και ενοποίηση παραμέτρων όπως κόστος [55], πολυπλοκότητα, προηγούμενη εμπειρία και αρχιτεκτονικές πλεονασμού
 - Μελέτη μοντέρνων τεχνικών αξιολόγησης αξιοπιστίας και εφαρμογή τους στο νανοδορυφόρο, κατά τον σχεδιασμό ή σε τροχιά
 - Οργάνωση δοκιμών για εξαγωγή πραγματικών αποτελεσμάτων αξιοπιστίας εξαρτημάτων (π.χ. δοκιμές ακτινοβολίας με βαρέα ιόντα)

A

Βιβλιογραφία

- [1] T. Wood. (Oct. 20, 2020). Visualizing All of Earth's Satellites: Who Owns Our Orbit?, Visual Capitalist, [Online]. Available: <https://www.visualcapitalist.com/visualizing-all-of-earths-satellites/> (visited on 06/24/2021).
- [2] Union of Concerned Scientists. (Jan. 1, 2021). Satellite Database, [Online]. Available: <https://www.ucsusa.org/resources/satellite-database> (visited on 06/24/2021).
- [3] T. Kelso, *NORAD Two-Line Element Sets Current Data*. [Online]. Available: <https://www.celestrak.com/NORAD/elements/> (visited on 06/24/2021).
- [4] AGI: Analytical Graphics, *Satellite Viewer*. [Online]. Available: <http://apps.agi.com/SatelliteViewer/> (visited on 06/24/2021).
- [5] J. R. Wertz and W. J. Larson, *Space Mission Analysis and Design*, 3rd edition. El Segundo, Calif. : Dordrecht ; Boston: Springer, Sep. 30, 1999, 996 pp., ISBN: 978-0-7923-5901-2.
- [6] G. Denis, D. Alary, X. Pasco, N. Pisot, D. Texier, and S. Toulza, "From new space to big space: How commercial space dream is becoming a reality", *Acta Astronautica*, vol. 166, σσ. 431–443, Jan. 2020, ISSN: 0094-5765. doi: [10.1016/j.actaastro.2019.08.031](https://doi.org/10.1016/j.actaastro.2019.08.031).
- [7] M. N. Sweeting, "Modern Small Satellites-Changing the Economics of Space", *Proceedings of the IEEE*, vol. 106, no. 3, σσ. 343–361, Mar. 2018, ISSN: 1558-2256. doi: [10.1109/JPROC.2018.2806218](https://doi.org/10.1109/JPROC.2018.2806218).
- [8] M. Swartwout. (Feb. 11, 2021). CubeSat Database, [Online]. Available: <https://sites.google.com/a/slu.edu/swartwout/home/cubesat-database> (visited on 05/20/2021).
- [9] A. Anthopoulos and A. F. Retselis, "Orbital analysis and the spin-orbit coupling for the AcubeSAT mission", Aristotle University of Thessaloniki, Oct. 13, 2020.
- [10] H. Riebeek. (Sep. 4, 2009). Catalog of Earth Satellite Orbits, [Online]. Available: <https://earthobservatory.nasa.gov/features/OrbitsCatalog/page1.php> (visited on 06/24/2021).
- [11] M. Swartwout, "CubeSat Mission Success: Are We Getting Better?", presented at the 2019 Cube-Sat Developers' Workshop, Apr. 23, 2019. [Online]. Available: <http://mstl.atl.calpoly.edu/~workshop/archive/2019/Spring/Day%201/Session%202/MichaelSwartwout.pdf>.
- [12] V. Kattakuri, "Failures in Spacecraft Systems: An Analysis from the Perspective of Decision Making", thesis, Purdue University Graduate School, Aug. 14, 2019. doi: [10.25394/PGS.9037379.v1](https://hammer.psu.edu/articles/thesis/FAILURES_IN_SPACECRAFT_SYSTEMS_AN_ANALYSIS_FROM_THE_PERSPECTIVE_OF_DECISION_MAKING/9037379/1). [Online]. Available: https://hammer.psu.edu/articles/thesis/FAILURES_IN_SPACECRAFT_SYSTEMS_AN_ANALYSIS_FROM_THE_PERSPECTIVE_OF_DECISION_MAKING/9037379/1 (visited on 06/24/2021).

- [13] S. A. Jacklin, “Small-Satellite Mission Failure Rates”, NASA/TM-2018-220034, Mar. 1, 2019. [Online]. Available: <https://ntrs.nasa.gov/citations/20190002705> (visited on 06/24/2021).
- [14] M. Langer, M. Weisgerber, J. Bouwmeester, and A. Hoehn, “A reliability estimation tool for reducing infant mortality in Cubesat missions”, Mar. 1, 2017, σσ. 1-9. doi: [10.1109/AERO.2017.7943598](https://doi.org/10.1109/AERO.2017.7943598).
- [15] P. Faure, A. Tanaka, and M. Cho, “Toward lean satellites reliability improvement using HORYU-IV project as case study”, *Acta Astronautica*, vol. 133, σσ. 33–49, Apr. 1, 2017, issn: 0094-5765. doi: [10.1016/j.actaastro.2016.12.030](https://doi.org/10.1016/j.actaastro.2016.12.030). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S009457651630724X> (visited on 05/20/2021).
- [16] A. Menchinelli, F. Ingiosi, L. Pamphili, P. Marzoli, R. Patriarca, F. Costantino, and F. Piergentili, “A Reliability Engineering Approach for Managing Risks in CubeSats”, *Aerospace*, vol. 5, no. 4, σ. 121, 4 Dec. 2018. doi: [10.3390/aerospace5040121](https://doi.org/10.3390/aerospace5040121). [Online]. Available: <https://www.mdpi.com/2226-4310/5/4/121> (visited on 06/24/2021).
- [17] A. Scholz, *Handbook of Free and Open Space Standards*. LibreCube Initiative, Apr. 2021. [Online]. Available: <https://gitlab.com/artur-scholz/handbook-space-standards/-/raw/master/main.pdf?inline=false>.
- [18] J. Bouwmeester, M. Langer, and E. Gill, “Survey on the implementation and reliability of CubeSat electrical bus interfaces”, *CEAS Space Journal*, vol. 9, no. 2, σσ. 163–173, Jun. 1, 2017, issn: 1868-2510. doi: [10.1007/s12567-016-0138-0](https://doi.org/10.1007/s12567-016-0138-0). [Online]. Available: <https://doi.org/10.1007/s12567-016-0138-0> (visited on 05/20/2021).
- [19] F. Volpetti, E. Petrova, and S. J. Maerkli, “A Microfluidic Biodisplay”, *ACS Synthetic Biology*, vol. 6, no. 11, σσ. 1979–1987, Nov. 17, 2017. doi: [10.1021/acssynbio.7b00088](https://doi.org/10.1021/acssynbio.7b00088). [Online]. Available: <https://doi.org/10.1021/acssynbio.7b00088> (visited on 06/24/2021).
- [20] G. Savvidis, R. Voulgarakis, T. Papafotiou, V. Moustakas, E. Mylonas, and V. Pappa, *AcubeSAT AOCS DDJF*, 2021. [Online]. Available: https://gitlab.com/acubesat/documentation/cdr-public/-/blob/master/DDJF/DDJF_AOCS.pdf.
- [21] I.-G. Velentzas and I. Kotsakiachidis, “Design of the Attitude Control System of the AcubeSAT Nanosatellite”, Aristotle University of Thessaloniki, Mar. 29, 2021. [Online]. Available: <https://ikee.lib.auth.gr/record/329273/>.
- [22] K. Kapoglou and E. Chatziargyriou, *AcubeSAT TTC DDJF*, 2021. [Online]. Available: https://gitlab.com/acubesat/documentation/cdr-public/-/blob/master/DDJF/DDJF_TTC.pdf.
- [23] M. Surligas, “SatNOGS-COMMS”, presented at the CubeSat Developers Workshop, Apr. 12, 2021.
- [24] D. White, C. Shields, P. Papadeas, A. Zisisimatos, M. Surligas, M. Papamatthaiou, D. Papadeas, and E. Kosmas, “Overview of the Satellite Networked Open Ground Stations (SatNOGS) Project”, *Small Satellite Conference*, Aug. 8, 2018. [Online]. Available: <https://digitalcommons.usu.edu/smallsat/2018/all2018/313>.
- [25] M. Langer and J. Bouwmeester, “Reliability of CubeSats – Statistical Data, Developers’ Beliefs and the Way Forward”, *Proceedings of the 30th Annual AIAA/USU Conference on Small Satellites*, 2016. [Online]. Available: <https://repository.tudelft.nl/islandora/object/uuid%3A4c6668ff-c994-467f-a6de-6518f209962e> (visited on 05/20/2021).
- [26] A.-F. Retselis, K. Kanavouras, and G. Pavlakis, *AcubeSAT System DDJF*, 2021. [Online]. Available: https://gitlab.com/acubesat/documentation/cdr-public/-/blob/master/DDJF/DDJF_SYS.pdf.
- [27] K. Kanavouras and G. Pavlakis, *AcubeSAT OBDH DDJF*, 2021. [Online]. Available: https://gitlab.com/acubesat/documentation/cdr-public/-/blob/master/DDJF/DDJF_OBDH.pdf.
- [28] PC/104 Embedded Consortium, “PC/104 Specification”, Aug. 13, 2008. [Online]. Available: https://pc104.org/wp-content/uploads/2015/02/PC104_Spec_v2_6.pdf.

- [29] K. Kanavouras, I. Kozaris, A. Theocharis, G. Pavlakis, and D. Stoupis, *AcubeSAT OBSW DDJF*, 2021. [Online]. Available: https://gitlab.com/acubesat/documentation/cdr-public/-/blob/master/DDJF/DDJF_OBSW.pdf.
- [30] A. Zaras, K. Kapoglis, M. Georgousi, T. Papafotiou, M. Chadolias, A. Anthopoulos, A.-F. Retselis, A. Arampatzis, K.-O. Xenos, and E. Christidou, *AcubeSAT Mission Description & Operations Plan*, 2021. [Online]. Available: <https://gitlab.com/acubesat/documentation/cdr-public/-/blob/master/MDO%20file/MDO.pdf>.
- [31] California Polytechnic State University, “CubeSat Design Specification Rev. 13”, Feb. 20, 2014. [Online]. Available: https://www.cubesat.org/s/cds_rev13_final2.pdf.
- [32] M. A. Aguirre, *Introduction to Space Systems: Design and Synthesis*, ser. Space Technology Library. New York: Springer-Verlag, 2013, ISBN: 978-1-4614-3757-4. doi: [10.1007/978-1-4614-3758-1](https://doi.org/10.1007/978-1-4614-3758-1). [Online]. Available: <https://www.springer.com/gp/book/9781461437574> (visited on 05/23/2021).
- [33] AcubeSAT Team. (May 8, 2021). AcubeSAT Functional Architecture, [Online]. Available: <https://gitlab.com/acubesat/systems-engineering/functional-architecture>.
- [34] A. Arampatzis, A. Zaras, P. Matsatsos, O. Ousoultzoglou, D. Nikolopoulou, and E. Sandaltzopoulou, *AcubeSAT Payload DDJF*, 2021. [Online]. Available: https://gitlab.com/acubesat/documentation/cdr-public/-/blob/master/DDJF/DDJF_PL.pdf.
- [35] ECSS Secretariat, “ECSS-E-ST-70-41C – Telemetry and telecommand packet utilization”, European Space Agency, Apr. 15, 2016. [Online]. Available: <https://ecss.nl/standard/ecss-e-st-70-41c-space-engineering-telemetry-and-telecommand-packet-utilization-15-april-2016/>.
- [36] Space Avionics Open interface Architecture, “SAVOIR FDIR Handbook”, European Space Agency, SAVOIR-HB-003, Oct. 2019. [Online]. Available: <https://essr.esa.int/project/savoir>.
- [37] ECSS Secretariat, “ECSS-E-70-41A – Telemetry and telecommand packet utilization”, European Space Agency, Jan. 30, 2003. [Online]. Available: <https://ecss.nl/standard/ecss-e-70-41a-ground-systems-and-operations-telemetry-and-telecommand-packet-utilization/> (visited on 05/24/2021).
- [38] J.-F. Kaufeler, “The ESA standard for telemetry and telecommand packet utilisation: PUS”, Nov. 1, 1994. [Online]. Available: <https://core.ac.uk/download/pdf/42783096.pdf> (visited on 05/24/2021).
- [39] C. S. Carlson, *Effective FMEAs: Achieving Safe, Reliable, and Economical Products and Processes Using Failure Mode and Effects Analysis*. John Wiley & Sons, Ltd, Apr. 12, 2012, 459 pp., ISBN: 978-1-118-31257-5. doi: [10.1002/9781118312575](https://doi.org/10.1002/9781118312575).
- [40] ECSS Secretariat, “ECSS-Q-ST-30-02C – Failure modes, effects (and criticality) analysis (FMEA/FMECA)”, European Space Agency, Mar. 6, 2009. [Online]. Available: <https://ecss.nl/standard/ecss-q-st-30-02c-failure-modes-effects-and-criticality-analysis-fmeafmeca/>.
- [41] A.-F. Retselis and K. Kanavouras, *AcubeSAT FMEA File*, 2021. [Online]. Available: <https://gitlab.com/acubesat/documentation/cdr-public/-/blob/master/FMEA%20file/FMEA.pdf>.
- [42] A. Birolini, *Reliability Engineering: Theory and Practice*, 4th ed. Berlin Heidelberg: Springer-Verlag, 2004, ISBN: 978-3-662-05409-3. doi: [10.1007/978-3-662-05409-3](https://doi.org/10.1007/978-3-662-05409-3). [Online]. Available: <https://www.springer.com/gp/book/9783662054093> (visited on 06/25/2021).
- [43] ESA TEC-QQD, “Effective Reliability Prediction for Space Applications – White Paper”, European Space Agency, White Paper ESA-TECQCD-WP-0969, May 25, 2016. [Online]. Available: https://www.reliability.space/app/download/14847209524/ESA_WhitePaper_2016.pdf.
- [44] V. Gupta, “Analysis of single event radiation effects and fault mechanisms in SRAM, FRAM and NAND Flash : Application to the MTcube nanosatellite project”, These de doctorat, Montpellier, Jul. 6, 2017. [Online]. Available: <https://tel.archives-ouvertes.fr/tel-01954572/document> (visited on 06/27/2021).

- [45] R. D. Yates and D. J. Goodman, *Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers*, 3rd edition. Hoboken, NJ: Wiley, Jan. 28, 2014, 512 pp., ISBN: 978-1-118-32456-1.
- [46] E. W. Weisstein. O. Lambert W-Function, [Online]. Available: <https://mathworld.wolfram.com/LambertW-Function.html> (visited on 06/27/2021).
- [47] A.-F. Retselis and K. Kanavouras. (Dec. 1, 2020). AcubeSAT FMEA Worksheet, GitLab, [Online]. Available: <https://gitlab.com/acubesat/systems-engineering/fmea> (visited on 06/05/2021).
- [48] S. Cheshire and M. Baker, “Consistent overhead byte stuffing”, in *Proceedings of the ACM SIGCOMM '97 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, ser. SIGCOMM '97, New York, NY, USA: Association for Computing Machinery, Oct. 1, 1997, σσ. 209–220, ISBN: 978-0-89791-905-0. doi: [10.1145/263105.263168](https://doi.org/10.1145/263105.263168). [Online]. Available: <https://doi.org/10.1145/263105.263168> (visited on 06/02/2021).
- [49] A. Sela, “Yamcs - A Lightweight Open-Source Mission Control System”, in *SpaceOps 2012 Conference*, ser. SpaceOps Conferences, 0 vols., American Institute of Aeronautics and Astronautics, Jun. 11, 2012. doi: [10.2514/6.2012-1280790](https://doi.org/10.2514/6.2012-1280790). [Online]. Available: <https://arc.aiaa.org/doi/10.2514/6.2012-1280790> (visited on 06/05/2021).
- [50] G. Simon, E. Shaya, K. Rice, S. Cooper, J. Dunham, and J. Champion, “XTCE: A standard XML-schema for describing mission operations databases”, in *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No.04TH8720)*, vol. 5, Mar. 2004, 3313–3325 Vol.5. doi: [10.1109/AERO.2004.1368138](https://doi.org/10.1109/AERO.2004.1368138).
- [51] Q. Huang and J. Jiang, “An overview of radiation effects on electronic devices under severe accident conditions in NPPs, rad-hardened design techniques and simulation tools”, *Progress in Nuclear Energy*, vol. 114, σσ. 105–120, Jul. 1, 2019, ISSN: 0149-1970. doi: [10.1016/j.pnucene.2019.02.008](https://doi.org/10.1016/j.pnucene.2019.02.008). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0149197019300563> (visited on 06/26/2021).
- [52] ECSS Secretariat, “ECSS-E-ST-70-11C – Space segment operability (31 July 2008)”, European Space Agency, Jul. 31, 2008. [Online]. Available: <https://ecss.nl/standard/ecss-e-st-70-11c-space-segment-operability/> (visited on 06/21/2021).
- [53] NASA, “Fault Management Handbook (Draft 2)”, NASA-HDBK-1002, Apr. 2, 2012. [Online]. Available: https://www.nasa.gov/pdf/636372main_NASA-HDBK-1002_Draft.pdf.
- [54] A. Zolghadri, “Advanced model-based FDIR techniques for aerospace systems: Today challenges and opportunities”, *Progress in Aerospace Sciences*, vol. 53, σσ. 18–29, Aug. 1, 2012, ISSN: 0376-0421. doi: [10.1016/j.paerosci.2012.02.004](https://doi.org/10.1016/j.paerosci.2012.02.004). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0376042112000292> (visited on 04/03/2021).
- [55] Y. K. Chang, H. Kim, and J. S. Kang, “Development of reliability-corrected cost model for Small Earth Observation satellites”, *Acta Astronautica*, vol. 88, σσ. 163–175, Jul. 1, 2013, ISSN: 0094-5765. doi: [10.1016/j.actaastro.2012.10.004](https://doi.org/10.1016/j.actaastro.2012.10.004). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0094576512003827> (visited on 04/04/2021).

B

Πηγαίος κώδικας

Στο Παράρτημα αυτό παρατίθεται ο πηγαίος κώδικας που αναπαριστά την κύρια λογική της πειραματικής διάταξης. Για τον πλήρη κώδικα που περιλαμβάνει τις βιβλιοθήκες και τα υποστηρικτικά αρχεία, ανατρέξτε στους Πίνακες 5.2 και 5.3. Ο κώδικας αυτού του παραρτήματος γράφτηκε εξ' ολοκλήρου στα πλαίσια της διπλωματικής, και διατηρείται στα αποθετήρια των προαναφερθέντων πινάκων.

`analysis.jl`

Το αρχείο αυτό, γραμμένο στη γλώσσα `Julia`, δημιουργεί όλα τα διαγράμματα της Ενότητας 4.2. Μπορεί να βρεθεί επίσης στη διεύθυνση <https://github.com/kongr45open/thesis-latex/tree/main/analysis>.

```
1  using Plots
2  using Plots.PlotMeasures
3  using LambertW
4  import PlotlyBase
5
6  plotly()
7  Plots.PlotlyBackend()
8
9  Plots.scalefontsizes()
10 Plots.scalefontsizes(1.3)
11
12 R = 0:0.01:1;
13
14 ## No-repair reliability
15 reliability_single = R;
16 reliability_dr = @. 1 - (1 - R)^2;
17 #reliability_3r = @. 1 - (1 - R)^3;
18 reliability_tmr = @. 3 * R^2 - 2 * R^3
19
20 plot(
21     R,
22     [reliability_single, reliability_dr, reliability_tmr],
23     label = ["Απλό εξάρτημα" "Διπλός πλεονασμός" "Τριτλός πλεονασμός με ψηφοφορία"],
24     # title = "Ανθεκτικότητα σύνθετου συστήματος σε μόνιμες βλάβες",
25     legend = :outerright,
26     xlabel = "Αξιοπιστία εξαρτήματος",
27     ylabel = "Αξιοπιστία συστήματος",
28     lw = 2,
29 )
30 savefig("reliability_norepair.pdf")
31
32 ## No-repair reliability with vulnerable voter
33 Rv = 0:0.01:1;
34 reliability_tmrv = reliability_tmr * permutedims(Rv)
35
```

```

36 heatmap(
37   R,
38   Rv,
39   reliability_tmr,
40   xlabel = "Αξιοπιστία R εξαρτήματος",
41   ylabel = "Αξιοπιστία Rv ψηφοφόρου",
42   colorbar_title = "Αξιοπιστία συστήματος",
43   aspect_ratio = 1,
44   c = :inferno
45 )
46 savefig("reliability_norepair_voter.pdf")
47
48 ## Prepair reliability
49 λ = 10 .^ (LinRange(-11, -2, 1000))
50
51 λ_single = λ;
52 λ_dr = @. -lambertw(- (2λ * exp(-λ) - λ^2 * exp(-2λ)));
53 λ_tmr = @. -lambertw(- (3(λ * exp(-λ))^2 - 2(λ * exp(-λ))^3));
54
55 plot(
56   λ,
57   [λ_single, λ_single, λ_dr, λ_tmr],
58   label = permutedims([
59     "Απλό εξάρτημα"
60     "Διπλός πλεονασμός, ψυχρός"
61     "Διπλός πλεονασμός, ενεργός"
62     "Τριπλός πλεονασμός με ψηφοφορία"]),
63   #title = "Ανθεκτικότητα σύννετου συστήματος σε παροδικά σφάλματα",
64   legend = (0.6,0.35),
65   xlabel = "Ψυθμός αποτυχίας εξαρτήματος",
66   ylabel = "Ψυθμός αποτυχίας συστήματος",
67   yaxis = :log,
68   xaxis = :log,
69   left_margin = -150px,
70   lw = 2,
71   linecolor=[palette(:tab10)[1] palette(:tab10)[1] palette(:tab10)[2] palette(:tab10)[3]]
72 )
73 savefig("reliability_repair.pdf")

```

main.cpp (fdir-demo)

Πρόκειται για το σημείο έναρξης της εφαρμογής του μικροελεγκτή, όπου ορίζονται οι διεργασίες του λειτουργικού συστήματος, και εκκινείται ο προγραμματιστής.

```

1  ****
2  Main Source File
3
4  Description:
5  This file contains the "main" function for a project. The
6  "main" function calls the "SYS_Initialize" function to initialize the state
7  machines of all modules in the system
8  ****
9
10 // ****
11 // ****
12 // Section: Included Files
13 // ****
14 // ****
15
16 #include <stddef.h>           // Defines NULL
17 #include <stdbool.h>          // Defines true
18 #include <stdlib.h>            // Defines EXIT_FAILURE
19 #include <memory.h>
20 #include <Logger.hpp>
21 #include <Parameters/SystemParameters.hpp>
22 #include <ServicePool.hpp>
23 #include <Parameters/SystemParameterMonitoring.hpp>
24 #include <Tasks/UARTTask.hpp>
25 #include <Tasks/UARTRXTask.hpp>
26 #include <Tasks/ECSSTask.hpp>
27 #include <Peripherals/MCP9808.hpp>
28 #include <Tasks/TemperatureTask.hpp>
29 #include <Tasks/InternalTemperatureTask.hpp>
30 #include "definitions.h"        // SYS function prototypes
31 #include "FreeRTOS.h"
32 #include "task.h"
33 #include "queue.h"
34 #include "list.h"
35
36
37 // ****
38 // ****
39 // Section: Main Entry Point
40 // ****
41 // ****
42
43 /**
44 * Just calls the operator() function of a task
45 * @param pvParameters Pointer to object of type Task
46 */
47 template<class Task>
48 static void vClassTask(void *pvParameters) {
49     (static_cast<Task *>(pvParameters))->operator();
50 }
51
52 std::optional<TemperatureTask> templtask;
53 std::optional<TemperatureTask> temp2task;
54 std::optional<InternalTemperatureTask> tempInternal;
55
56 int main ( void )
57 {
58     // Initialize all modules
59     SYS_Initialize ( NULL );
60
61     // Disable interrupts to prevent RTOS SysTick from crashing the system
62     SysTick->CTRL &= ~(SysTick_CTRL_CLKSOURCE_Msk | SysTick_CTRL_ENABLE_Msk);
63
64     Logger::format.precision(2);
65
66     systemParameterMonitoring.emplace();
67     uartTask.emplace();
68     uartRXtask.emplace();

```

```

69      ecssTask.emplace();
70
71      templtask.emplace(systemParameters.temperature1,
72                         systemParameters.temperature1Status, 0, SENS1_PIN, BTN0_PIN);
73      temp2task.emplace(systemParameters.temperature2,
74                         systemParameters.temperature2Status, 2, SENS2_PIN, BT1_PIN);
75
76      xTaskCreate(vClassTask<InternalTemperatureTask>,
77                   "Internal_Temp", 2500, &tempInternal, tskIDLE_PRIORITY + 1, nullptr);
78      xTaskCreate(vClassTask<ECSSTask>, "ECSS", 3000, &ecssTask, tskIDLE_PRIORITY + 1, nullptr);
79
80      xTaskCreate(vClassTask<UARTTask>, "UART_Tx", 3000, &uartTask, tskIDLE_PRIORITY + 1, nullptr);
81      xTaskCreate(vClassTask<UARTRXTask>, "UART_Rx", 6000, &uartRXtask, tskIDLE_PRIORITY + 1, nullptr);
82
83      xTaskCreate(vClassTask<TemperatureTask>, "T1", 1500, &templtask, tskIDLE_PRIORITY + 1, nullptr);
84      xTaskCreate(vClassTask<TemperatureTask>, "T2", 1500, &temp2task, tskIDLE_PRIORITY + 1, nullptr);
85
86      SysTick->CTRL |= SysTick_CTRL_CLKSOURCE_Msk | SysTick_CTRL_ENABLE_Msk;
87      vTaskStartScheduler();
88 #pragma clang diagnostic push
89 #pragma ide diagnostic ignored "EndlessLoop"
90     while ( true )
91     {
92         SYS_Tasks ( );
93     }
94 #pragma clang diagnostic pop
95
96 /* Execution should not come here during normal operation */
97 return ( EXIT_FAILURE );
98 }
```

SystemParameters.hpp (fdir-demo)

Εδώ ορίζονται όλες οι παράμετροι που διατηρούνται στον μικροελεγκτή, σε συνδυασμό με τους τύπους δεδομένων τους, και συναρτήσεις που καλούνται κατά την ανάγνωση και την εγγραφή τους.

```

1 #ifndef FDIR_DEMO_SYSTEMPARAMETERS_HPP
2 #define FDIR_DEMO_SYSTEMPARAMETERS_HPP
3
4 #include <cstdint>
5 #include <Services/Parameter.hpp>
6 #include <Logger.hpp>
7
8 class SystemParameters {
9 public:
10     enum class TemperatureStatus : uint8_t {
11         Nominal = 0b01,
12         Timeout = 0b10,
13         Disabled = 0b00
14     };
15
16     Parameter<float> temperature1{0.0};
17     Parameter<float> temperature2{0.0};
18     FunctionParameter<float> temperatureDelta{[this]() -> float {
19         return temperature2.getValue() - temperature1.getValue();
20     }};
21     Parameter<float> temperatureInternal{0.0};
22
23     CallbackParameter<TemperatureStatus> temperature1Status{
24         TemperatureStatus::Nominal, temperature1StatusCallback };
25     CallbackParameter<TemperatureStatus> temperature2Status{
26         TemperatureStatus::Nominal, temperature2StatusCallback };
27     FunctionParameter<TemperatureStatus> temperature12Status{[this]() {
28         if (
29             temperature1Status.getValue() == TemperatureStatus::Nominal &&
30             temperature2Status.getValue() == TemperatureStatus::Nominal) {
31             return TemperatureStatus::Nominal;
32         } else {
33             return TemperatureStatus::Disabled;
34         }
35     }};
36
37     FunctionParameter<uint32_t> tick{SystemParameters::getTick};
38
39 /**
40 * The key of the array is the ID of the parameter as specified in PUS
41 */
42 etl::array<std::reference_wrapper<ParameterBase>, ECSS_PARAMETER_COUNT> parametersArray = {
43     std::ref<ParameterBase>(temperature1),
44     std::ref<ParameterBase>(temperature2),
45     std::ref<ParameterBase>(temperatureDelta),
46     std::ref<ParameterBase>(temperature1Status),
47     std::ref<ParameterBase>(temperature2Status),
48     std::ref<ParameterBase>(temperature12Status),
49     std::ref<ParameterBase>(temperatureInternal),
50     std::ref<ParameterBase>(tick)
51 };
52
53 SystemParameters();
54
55 template<typename Value>
56 Value getParameterValue(uint16_t parameterId) {
57     return static_cast<ReadableParameter<Value>*>(&(parametersArray[parameterId].get()))->getValue();
58 }
59
60 uint64_t convertParameterValueToInt(uint16_t parameterId) {
61     return parametersArray[parameterId].get().convertValue();
62 }
63
64 float convertParameterValueToFloat(uint16_t parameterId) {
65     return parametersArray[parameterId].get().convertFloat();
66 }
67 private:
68     static uint32_t getTick();

```

```
69     static void temperature1StatusCallback(TemperatureStatus& status);
70     static void temperature2StatusCallback(TemperatureStatus& status);
71 };
72
73 extern SystemParameters systemParameters;
74
75 #endif //FDIR_DEMO_SYSTEMPARAMETERS_HPP
```

SystemParameterMonitoring.hpp (fdir-demo)

Εδώ ορίζονται οι προεπιλεγμένοι ορισμοί παρακολούθησης. Οι τιμές αυτού του αρχείου συμπίπτουν με αυτές του [Πίνακας 5.11](#).

```

1 #ifndef FDIR_DEMO_SYSTEMPARAMETERMONITORING_HPP
2 #define FDIR_DEMO_SYSTEMPARAMETERMONITORING_HPP
3
4 #include <Services/OnBoardMonitoringService.hpp>
5 #include <Services/OnBoardMonitoringChecks.hpp>
6 #include <ServicePool.hpp>
7
8 class SystemParameterMonitoring {
9     template<class Check>
10    using MonitoringDefinition =
11        OnBoardMonitoringService::ParameterMonitoringDefinition<typename Check::type, Check>;
12
13    MonitoringDefinition<ExpectedValueCheck<SystemParameters::TemperatureStatus>> sensor1timeoutCheck{
14        0, 3, 500, 2, {
15            0b10, SystemParameters::TemperatureStatus::Nominal, 100
16        }, {}
17    };
18
19    MonitoringDefinition<ExpectedValueCheck<SystemParameters::TemperatureStatus>> sensor2timeoutCheck{
20        1, 4, 500, 2, {
21            0b10, SystemParameters::TemperatureStatus::Nominal, 101
22        }, {}
23    };
24
25    MonitoringDefinition<ExpectedValueCheck<SystemParameters::TemperatureStatus>> sensor1permanentTimeoutCheck{
26        2, 3, 500, 5, {
27            0b10, SystemParameters::TemperatureStatus::Nominal, 102
28        }, {}
29    };
30
31    MonitoringDefinition<ExpectedValueCheck<SystemParameters::TemperatureStatus>> sensor2permanentTimeoutCheck{
32        3, 4, 500, 5, {
33            0b10, SystemParameters::TemperatureStatus::Nominal, 103
34        }, {}
35    };
36
37    MonitoringDefinition<LimitCheck<float>> sensor1check{
38        4, 0, 500, 2, {
39            -40, 104, 100, 104
40        }, {{
41            3, 0xff, static_cast<uint64_t>(SystemParameters::TemperatureStatus::Nominal)
42        }}
43    };
44
45    MonitoringDefinition<LimitCheck<float>> sensor2check{
46        5, 1, 500, 2, {
47            -40, 105, 100, 105
48        }, {{
49            4, 0xff, static_cast<uint64_t>(SystemParameters::TemperatureStatus::Nominal)
50        }}
51    };
52
53    MonitoringDefinition<LimitCheck<float>> sensor1permamentCheck{
54        7, 0, 500, 5, {
55            -40, 107, 100, 107
56        }, {{
57            3, 0xff, static_cast<uint64_t>(SystemParameters::TemperatureStatus::Nominal)
58        }}
59    };
60
61    MonitoringDefinition<LimitCheck<float>> sensor2permanentCheck{
62        8, 1, 500, 5, {
63            -40, 108, 100, 108
64        }, {{
65            4, 0xff, static_cast<uint64_t>(SystemParameters::TemperatureStatus::Nominal)
66        }}
67    };
68

```

```

69  MonitoringDefinition<LimitCheck<float>> sensorDeltaCheck{
70      6, 2, 500, 6, {
71          -5, 106, 5, 106
72      }, {{
73          5, 0xff, static_cast<uint64_t>(SystemParameters::TemperatureStatus::Nominal)
74      }}
75  };
76
77  MonitoringDefinition<LimitCheck<float>> sensorDeltaPermanentCheck{
78      9, 2, 500, 11, {
79          -5, 109, 5, 109
80      }, {{
81          5, 0xff, static_cast<uint64_t>(SystemParameters::TemperatureStatus::Nominal)
82      }}
83  };
84
85  MonitoringDefinition<LimitCheck<float>> subsystemSafety1check{
86      10, 0, 500, 12, {
87          -std::numeric_limits<float>::infinity(), 110, 50, 110
88      }, {{
89          3, 0xff, static_cast<uint64_t>(SystemParameters::TemperatureStatus::Nominal)
90      }}
91  };
92
93  MonitoringDefinition<LimitCheck<float>> subsystemSafety2check{
94      11, 1, 500, 12, {
95          -std::numeric_limits<float>::infinity(), 111, 50, 111
96      }, {{
97          4, 0xff, static_cast<uint64_t>(SystemParameters::TemperatureStatus::Nominal)
98      }}
99  };
100
101
102 public:
103     SystemParameterMonitoring() {
104         OnBoardMonitoringService &service = Services.onBoardMonitoring;
105
106         service.addParameterMonitoringDefinition(sensor1timeoutCheck);
107         service.addParameterMonitoringDefinition(sensor2timeoutCheck);
108         service.addParameterMonitoringDefinition(sensor1permanentTimeoutCheck);
109         service.addParameterMonitoringDefinition(sensor2permanentTimeoutCheck);
110         service.addParameterMonitoringDefinition(sensor1check);
111         service.addParameterMonitoringDefinition(sensor2check);
112         service.addParameterMonitoringDefinition(sensorDeltaCheck);
113         service.addParameterMonitoringDefinition(sensor1permamentCheck);
114         service.addParameterMonitoringDefinition(sensor2permanentCheck);
115         service.addParameterMonitoringDefinition(sensorDeltaPermanentCheck);
116         service.addParameterMonitoringDefinition(subsystemSafety1check);
117         service.addParameterMonitoringDefinition(subsystemSafety2check);
118     }
119 };
120
121 extern std::optional<SystemParameterMonitoring> systemParameterMonitoring;
122
123 #endif //FDIR_DEMO_SYSTEMPARAMETERMONITORING_HPP

```

ECSTask.cpp (fdiR-demo)

Εδώ ορίζονται δεδομένα των ECSS υπηρεσιών που δεν ορίστηκαν σε άλλα αρχεία. Ακόμα, εδώ εκτελείται ο κύριος βρόχος των υπηρεσιών ECSS.

```

1 #include <optional>
2 #include <FreeRTOS.h>
3 #include <definitions.h>
4 #include <ServicePool.hpp>
5 #include <task.h>
6 #include <main.hpp>
7 #include "Tasks/ECSTask.h"
8
9 std::optional<ECSTask> ecstask;
10
11 void ECSTask::operator()() {
12     Services.eventReport.informativeEventReport(EventReportService::MCUStart, "");
13
14     for (;;) {
15         pinval = PIO_PinRead(PIO_PIN_PA23);
16         vTaskDelay(pdMS_TO_TICKS(1));
17         Services.onBoardMonitoring.checkAll(xTaskGetTickCount());
18         Services.housekeeping.checkAndSendHousekeepingReports(xTaskGetTickCount());
19     }
20 }
21
22 #pragma clang diagnostic push
23 #pragma ide diagnostic ignored "performance-unnecessary-value-param"
24
25 ECSTask::ECSTask() {
26     auto &Function = Services.functionManagement;
27     auto &Parameter = Services.parameterManagement;
28
29     auto compose = [] (auto &&PH1) {
30         return MessageParser::composeArbitraryECSS<64>(std::forward<decltype(PH1)>(PH1), 0);
31     };
32
33     Services.functionManagement.include("restart", functionRestart);
34     Services.functionManagement.include("restart_sensor", functionRestartSensor);
35     Services.functionManagement.include("log", functionLog);
36     Services.functionManagement.include("bogus_temp", functionBogusTemperature);
37
38     Services.eventAction.addEventActionDefinition(
39         {0, 100, 1, compose(Function.callFromGround("restart_sensor", "1"))});
40     Services.eventAction.addEventActionDefinition(
41         {0, 101, 2, compose(Function.callFromGround("restart_sensor", "2"))});
42     Services.eventAction.addEventActionDefinition(
43         {0, 102, 3, compose(Parameter.setParameterFromGround(
44             systemParameters.temperature1Status, SystemParameters::TemperatureStatus::Disabled))});
45     Services.eventAction.addEventActionDefinition(
46         {0, 103, 4, compose(Parameter.setParameterFromGround(
47             systemParameters.temperature2Status, SystemParameters::TemperatureStatus::Disabled))});
48     Services.eventAction.addEventActionDefinition(
49         {0, 104, 5, compose(Function.callFromGround("restart_sensor", "1"))});
50     Services.eventAction.addEventActionDefinition(
51         {0, 105, 6, compose(Function.callFromGround("restart_sensor", "2"))});
52     Services.eventAction.addEventActionDefinition(
53         {0, 106, 7, compose(Function.callFromGround("restart_sensor", "3"))});
54     Services.eventAction.addEventActionDefinition(
55         {0, 107, 8, compose(Parameter.setParameterFromGround(
56             systemParameters.temperature1Status, SystemParameters::TemperatureStatus::Disabled))});
57     Services.eventAction.addEventActionDefinition(
58         {0, 108, 9, compose(Parameter.setParameterFromGround(
59             systemParameters.temperature2Status, SystemParameters::TemperatureStatus::Disabled))});
60     Services.eventAction.addEventActionDefinition(
61         {0, 109, 10, compose(Parameter.setParameterFromGround(
62             systemParameters.temperature1Status, SystemParameters::TemperatureStatus::Disabled))});
63     Services.eventAction.addEventActionDefinition(
64         {0, 109, 11, compose(Parameter.setParameterFromGround(
65             systemParameters.temperature2Status, SystemParameters::TemperatureStatus::Disabled))});
66     Services.eventAction.addEventActionDefinition(
67         {0, 110, 12, compose(Function.callFromGround("restart"))});
68     Services.eventAction.addEventActionDefinition()

```

```

69             {0, 111, 13, compose(Function.callFromGround("restart"))});
70
71 }
72
73 void ECSSTask::functionRestart(String<16> args) {
74     vTaskDelay(100);
75     NVIC_SystemReset();
76 }
77
78 void ECSSTask::functionRestartSensor(String<16> args) {
79     char sensor = args[0];
80
81     if (sensor == '1' || sensor == '3') {
82         if (temp1task) {
83             temp1task->restart();
84         }
85     }
86
87     if (sensor == '2' || sensor == '3') {
88         if (temp2task) {
89             temp2task->restart();
90         }
91     }
92 }
93
94 void ECSSTask::functionLog(String<16> log) {
95     char data[17]{'\0'};
96     memcpy(data, log.data(), 16);
97     data[16] = '\0';
98
99     LOG_INFO << data;
100 }
101
102 void ECSSTask::functionBogusTemperature(String<16>) {
103     temp1task->addBogusTemperature();
104     temp2task->addBogusTemperature(500);
105 }
106
107 #pragma clang diagnostic pop

```

TemperatureTask.cpp (fdir-demo)

Η διεργασία αυτή είναι η κύρια διεπαφή του μικροελεγκτή με έναν αισθητήρα θερμοκρασίας. Λαμβάνει την τιμή του κάθε 130 ms και ενημερώνει την αντίστοιχη παράμετρο. Ταυτόχρονα, ελέγχει την υγεία του διαύλου I²C και προσθέτει εικονικές τιμές αν ξητείται.

```

1 #include <Logger.hpp>
2 #include <ServicePool.hpp>
3 #include "Tasks/TemperatureTask.hpp"
4
5 TemperatureTask::TemperatureTask(Parameter<float> &parameter,
6                                 CallbackParameter<SystemParameters::TemperatureStatus> &statusParameter,
7                                 uint8_t sensorI2c, PIO_PIN sensorPin, PIO_PIN buttonPin)
8     : parameter(parameter), statusParameter(statusParameter), mcp9808(sensorI2c), sensorPin(sensorPin),
9       buttonPin(buttonPin) {
10    buttonPin(buttonPin);
11    PIO_PinWrite(sensorPin, true);
12}
13
14 void TemperatureTask::operator()() {
15    taskHandle = xTaskGetCurrentTaskHandle();
16
17    while (true) {
18        wasSuspended = false;
19
20        vTaskDelay(130);
21
22        float temperature = 0;
23        bool status = mcp9808.isIDok();
24
25        if (status) {
26            status = mcp9808.getTemp(temperature);
27        }
28
29        if (!status && wasSuspended) {
30            // I2C timeout, but the task was suspended so the I2C peripheral didn't get enough information.
31            // Retry again after a delay
32            continue;
33        }
34
35        if (status) {
36            statusParameter.setValue(SystemParameters::TemperatureStatus::Nominal);
37        } else {
38            statusParameter.setValue(SystemParameters::TemperatureStatus::Timeout);
39        }
40
41        if (!PIO_PinRead(buttonPin)) {
42            temperature += 80;
43        }
44
45        if (bogusTemperatureTime != 0) {
46            temperature += bellFunction();
47        }
48
49        // TODO: Set parameter on timeout?
50        parameter.setValue(temperature);
51
52        LOG_DEBUG << "T [" << pcTaskGetName(nullptr) << "]": " << parameter.getValue();
53    }
54
55    void TemperatureTask::setOutput(bool output) {
56        PIO_PinWrite(sensorPin, output);
57
58        if (taskHandle != nullptr) {
59            if (output) {
60                wasSuspended = true;
61                vTaskResume(taskHandle);
62            } else {
63                vTaskSuspend(taskHandle);
64            }
65        }
66    }

```

```

67
68 void TemperatureTask::restart() {
69     if (statusParameter.getValue() != SystemParameters::TemperatureStatus::Disabled) {
70         setOutput(false);
71         vTaskDelay(100);
72     }
73
74     // Second check, in case the parameter was somehow set during the delay
75     if (statusParameter.getValue() != SystemParameters::TemperatureStatus::Disabled) {
76         setOutput(true);
77     }
78 }
79
80 float TemperatureTask::bellFunction() {
81     const float MaxBogusTemperature = 60;
82     const float BogusDuration = 7000;
83
84     // Apply error for t in (0, 2*BogusDuration)
85     float t = xTaskGetTickCount() - bogusTemperatureTime;
86
87     // Apply error for ot in (-BogusDuration, BogusDuration)
88     float shiftedT = t - BogusDuration;
89
90     if (abs(shiftedT) >= BogusDuration) {
91         return 0;
92     } else {
93         return MaxBogusTemperature * expf(1) *
94                 expf(powf(BogusDuration, 2) / (powf(shiftedT, 2) - powf(BogusDuration, 2)));
95     }
96 }
97
98 void TemperatureTask::addBogusTemperature(int32_t shift) {
99     bogusTemperatureTime = xTaskGetTickCount() + shift;
100 }
```

OnBoardMonitoringService.hpp (fdir-demo)

Η καρδιά της λογικής του FDIR, αυτή η κλάση δημιουργήθηκε στα πλαίσια αυτής της διπλωματικής, και εκτελεί όλους τους ελέγχους για τον εντοπισμό βλαβών.

```

1 #ifndef FDIR_DEMO_ONBOARDMONITORINGSERVICE_HPP
2 #define FDIR_DEMO_ONBOARDMONITORINGSERVICE_HPP
3
4 #include <cstdint>
5 #include "Service.hpp"
6 #include <etl/bitset.h>
7 #include <etl/variant.h>
8 #include <etl/vector.h>
9 #include <etl/pool.h>
10 #include "Parameters/SystemParameters.hpp"
11 #include "Logger.hpp"
12 #include "Services/EventReportService.hpp"
13
14 class OnBoardMonitoringService : public Service {
15 public:
16     inline static const uint8_t ServiceType = 12;
17
18     enum class CheckTypeID : uint8_t {
19         ExpectedValue = 0,
20         Limit = 1,
21         Delta = 2
22     };
23
24     struct CheckValidityCondition {
25         uint16_t parameterId;
26         uint64_t mask;
27         uint64_t expectedValue;
28
29         bool check() {
30             return (systemParameters.convertParameterValueToInt(parameterId) & mask) ==
31                 (expectedValue & mask);
32         }
33     };
34
35     struct CheckTransitionEntry {
36         uint16_t pmonId;
37         uint16_t monitoredParameterId;
38         CheckTypeID checkType;
39         float parameterValue;
40         uint8_t previousStatus;
41         uint8_t currentStatus;
42         uint32_t transitionTime;
43     };
44
45     OnBoardMonitoringService() {
46         serviceType = 12;
47     }
48
49     class ParameterMonitoringDefinitionBase {
50     protected:
51         uint16_t pmonId; ///< Parameter monitoring ID as defined in the standard
52         uint16_t monitoredParameterId;
53         uint32_t monitoringInterval;
54         uint8_t repetitionNumber; ///< The number of checks needed to switch the parameter status
55
56         std::optional<CheckValidityCondition> validityCondition;
57
58         ParameterMonitoringDefinitionBase(uint16_t pmonId, uint16_t monitoredParameterId,
59                                         uint32_t monitoringInterval,
60                                         uint16_t repetitionNumber,
61                                         const std::optional<CheckValidityCondition> &condition)
62             : pmonId(pmonId), monitoredParameterId(monitoredParameterId),
63               monitoringInterval(monitoringInterval),
64               repetitionNumber(repetitionNumber), validityCondition(condition) {}
65
66     public:
67         virtual std::optional<CheckTransitionEntry> check(uint64_t timestamp) = 0;
68

```

```

69     virtual void setEnabled(bool enabled) = 0;
70
71     virtual void appendDefinitionReport(Message &message) = 0;
72
73     uint16_t getPmonId() const {
74         return pmonId;
75     }
76
77
78     template<typename T, class CheckType>
79     class ParameterMonitoringDefinition : public ParameterMonitoringDefinitionBase {
80     public:
81         ParameterMonitoringDefinition(uint16_t pmonId, uint16_t monitoredParameterId,
82                                         uint32_t monitoringInterval,
83                                         uint16_t repetitionNumber, CheckType checkParameters,
84                                         const std::optional<CheckValidityCondition> &condition)
85             : ParameterMonitoringDefinitionBase(pmonId, monitoredParameterId, monitoringInterval,
86                                                 repetitionNumber, condition),
87             checkParameters(checkParameters) {}
88
89         std::optional<CheckTransitionEntry> check(uint64_t timestamp) override;
90
91         void setEnabled(bool enabled) override {
92             this->enabled = enabled;
93         }
94
95         void appendDefinitionReport(Message &message) override;
96
97     private:
98         CheckType checkParameters;
99
100        bool enabled = true;
101
102        uint64_t lastCheckTimestamp = 0;
103        uint8_t repetitionsCounted = 0;
104
105        using CheckingStatus = typename CheckType::CheckingStatus;
106
107        CheckingStatus previousStatus = CheckingStatus::Unchecked;
108        CheckingStatus currentStatus = CheckingStatus::Unchecked;
109        CheckingStatus futureStatus = CheckingStatus::Unchecked;
110    };
111
112     template<class T, class CheckType>
113     void addParameterMonitoringDefinition(ParameterMonitoringDefinition<T, CheckType> &definition) {
114         parameterMonitoringDefinitions.push_back(std::ref(definition));
115     }
116
117     void checkAll(uint64_t currentTimestamp) {
118         if (not monitoringEnabled) {
119             return;
120         }
121
122         for (auto &it: parameterMonitoringDefinitions) {
123             auto transition = it.get().check(currentTimestamp);
124
125             if (transition) {
126                 checkTransitionList.push_back(transition.value());
127             }
128         }
129
130         if (checkTransitionList.full() || currentTimestamp > lastTransitionTimestamp + TransitionListPeriod) {
131             lastTransitionTimestamp = currentTimestamp;
132             checkTransitionReport();
133         }
134     }
135
136     /**
137      * TC[12,5]
138      */
139     void addParameterMonitoringDefinition(Message &message);
140
141     /**
142      * TC[12,6]
143      */
144     void deleteParameterMonitoringDefinition(Message &message);

```

```

145
146     /**
147      * TC[12,1]
148      */
149  void enableParameterMonitoringDefinitions(Message &message);
150
151 /**
152  * TC[12,2]
153 */
154  void disableParameterMonitoringDefinitions(Message &message);
155
156 /**
157  * TC[12,15]
158 */
159  void enableParameterMonitoringFunction(Message &message);
160
161 /**
162  * TC[12,16]
163 */
164  void disableParameterMonitoringFunction(Message &message);
165
166  void reportParameterMonitoringDefinitions(Message &message);
167
168 /**
169  * TM[12,12]
170 */
171  void checkTransitionReport();
172
173  void execute(Message &message);
174
175 private:
176     inline static const int CheckTransitionListItems = 12;
177     inline static const uint64_t TransitionListPeriod = 200;
178
179     uint64_t lastTransitionTimestamp = 0;
180
181     etl::vector<std::reference_wrapper<ParameterMonitoringDefinitionBase>, 20> parameterMonitoringDefinitions;
182
183     bool monitoringEnabled = true;
184
185 /**
186  * @todo This variable should be atomic
187 */
188     etl::vector<CheckTransitionEntry, CheckTransitionListItems> checkTransitionList;
189
190     etl::generic_pool<100, 8, 10> monitoringDefinitionPool;
191 };
192
193 template<typename T, class CheckType>
194 inline void OnBoardMonitoringService::ParameterMonitoringDefinition<T, CheckType>::appendDefinitionReport(
195     Message &message) {
196     message.append(uint16_t{pmonId});
197     message.append(uint16_t{monitoredParameterId});
198     if (validityCondition) {
199         message.append(uint16_t{validityCondition->parameterId});
200         message.append(uint64_t{validityCondition->mask});
201         message.append(uint64_t{validityCondition->expectedValue});
202     } else {
203         message.append(uint16_t{0});
204         message.append(uint64_t{0});
205         message.append(uint64_t{0});
206     }
207     message.append(uint32_t{monitoringInterval});
208     message.append(static_cast<uint8_t>(currentStatus));
209     message.append(uint8_t{repetitionNumber});
210     message.append(static_cast<uint8_t>(CheckType::ID));
211     checkParameters.appendDefinitionReport(message);
212 }
213
214 template<typename T, class CheckType>
215 std::optional<OnBoardMonitoringService::CheckTransitionEntry>
216 OnBoardMonitoringService::ParameterMonitoringDefinition<T, CheckType>::check(uint64_t timestamp) {
217     // TODO: Split the "monitoring interval" and "repetition count" capabilities of this function
218
219     if (timestamp - lastCheckTimestamp < monitoringInterval) {
220         return std::nullopt;

```

```

221 }
222 lastCheckTimestamp = timestamp;
223 T parameterValue;
224 if constexpr (std::is_same_v<T, float>) {
225     parameterValue = systemParameters.convertParameterValueToFloat(monitoredParameterId);
226 } else {
227     parameterValue = systemParameters.getParameterValue<T>(monitoredParameterId);
228 }
229
230 CheckingStatus newStatus = checkParameters.check(parameterValue);
231
232 bool transition = false;
233 bool needsCheck = true;
234
235 previousStatus = currentStatus;
236
237 if (!enabled) {
238     currentStatus = futureStatus = CheckType::CheckingStatus::Unchecked;
239     repetitionsCounted = 0;
240     transition = currentStatus != previousStatus;
241     needsCheck = false;
242 } else if (validityCondition && !validityCondition->check()) {
243     currentStatus = futureStatus = CheckType::CheckingStatus::Invalid;
244     repetitionsCounted = 0;
245     transition = currentStatus != previousStatus;
246     needsCheck = false;
247 }
248
249 if (needsCheck) {
250     if (repetitionNumber <= 1) {
251         currentStatus = futureStatus = newStatus;
252
253         if (currentStatus != previousStatus) {
254             transition = true;
255         }
256     } else {
257         if (newStatus == futureStatus) {
258             if (repetitionsCounted + 1 >= repetitionNumber) {
259                 currentStatus = futureStatus;
260                 repetitionsCounted = 0;
261
262                 if (currentStatus != previousStatus) {
263                     transition = true;
264                 }
265             } else {
266                 ++repetitionsCounted;
267             }
268         } else {
269             repetitionsCounted = 0;
270             futureStatus = newStatus;
271         }
272     }
273 }
274
275 if (transition) {
276     LOG_ERROR << "Monitoring status " << pmonId << " changed from " << static_cast<int>(previousStatus)
277     << " to " << static_cast<int>(currentStatus);
278     checkParameters.generateEvent(currentStatus);
279
280     return std::optional<OnBoardMonitoringService::CheckTransitionEntry>({
281         pmonId,
282         monitoredParameterId,
283         CheckType::ID,
284         static_cast<float>(parameterValue),
285         static_cast<uint8_t>(previousStatus),
286         static_cast<uint8_t>(currentStatus),
287         timestamp
288     });
289 } else {
290     return std::nullopt;
291 }
292 }
293
294 #endif //FDIR_DEMO_ONBOARDMONITORINGSERVICE_HPP

```

xtce.xml (fdir-demo-yamcs)

Το αρχείο αυτό ορίζει τη μορφή των πακέτων τηλεμετρίας και τηλεεντολών που μπορεί να αντιληφθεί και να εκπέμψει αντίστοιχα ο σταθμός βάσης.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <SpaceSystem name="fdirdemo" xmlns="http://www.omg.org/space/xtce"
3 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4 xsi:schemaLocation="http://www.omg.org/spec/XTCE/20180204 https://www.omg.org/spec/XTCE/20180204/SpaceSystem.xsd">
5
6 <TelemetryMetaDataSet>
7   <ParameterTypeSet>
8     <BooleanParameterType name="bool8_t">
9       <UnitSet />
10      <IntegerDataEncoding sizeInBits="8" />
11    </BooleanParameterType>
12    <IntegerParameterType name="uint8_t" signed="false">
13      <UnitSet />
14      <IntegerDataEncoding sizeInBits="8" />
15    </IntegerParameterType>
16    <IntegerParameterType name="uint16_t" signed="false">
17      <UnitSet />
18      <IntegerDataEncoding sizeInBits="16" />
19    </IntegerParameterType>
20    <IntegerParameterType name="uint32_t" signed="false">
21      <UnitSet />
22      <IntegerDataEncoding sizeInBits="32" />
23    </IntegerParameterType>
24    <IntegerParameterType name="uint64_t" signed="false">
25      <UnitSet />
26      <IntegerDataEncoding sizeInBits="64" />
27    </IntegerParameterType>
28    <FloatParameterType name="float_t">
29      <UnitSet />
30      <FloatDataEncoding sizeInBits="32" />
31    </FloatParameterType>
32    <FloatParameterType name="double_t">
33      <UnitSet />
34      <FloatDataEncoding sizeInBits="64" />
35    </FloatParameterType>
36    <FloatParameterType name="Temperature_Type">
37      <UnitSet>
38        <Unit>celsius</Unit>
39      </UnitSet>
40      <FloatDataEncoding sizeInBits="32" />
41    </FloatParameterType>
42    <IntegerParameterType name="Tick_Type" signed="false">
43      <UnitSet>
44        <Unit>ms</Unit>
45      </UnitSet>
46      <IntegerDataEncoding sizeInBits="32" />
47    </IntegerParameterType>
48    <IntegerParameterType name="Packet_Version_Type" signed="false">
49      <UnitSet />
50      <IntegerDataEncoding sizeInBits="3" />
51    </IntegerParameterType>
52    <IntegerParameterType name="APID_Type" signed="false">
53      <UnitSet />
54      <IntegerDataEncoding sizeInBits="11" />
55    </IntegerParameterType>
56    <IntegerParameterType name="Packet_Sequence_Count_Type" signed="false">
57      <UnitSet />
58      <IntegerDataEncoding sizeInBits="14" />
59    </IntegerParameterType>
60    <EnumeratedParameterType name="Packet_Type_Type">
61      <UnitSet />
62      <IntegerDataEncoding sizeInBits="1" />
63      <EnumerationList>
64        <Enumeration label="TM" value="0" />
65        <Enumeration label="TC" value="1" />
66      </EnumerationList>
67    </EnumeratedParameterType>
68    <EnumeratedParameterType name="Temperature_Status_Type">
69      <UnitSet />
70      <IntegerDataEncoding sizeInBits="8" />
71      <EnumerationList>
72        <Enumeration label="Nominal" value="1" />
73        <Enumeration label="Timeout" value="2" />
74        <Enumeration label="Disabled" value="0" />
75      </EnumerationList>
76    </EnumeratedParameterType>
77    <EnumeratedParameterType name="Event_Definition_ID_Type">
78      <UnitSet />
79      <IntegerDataEncoding sizeInBits="16" />
80      <EnumerationList>
81        <Enumeration label="Informative_Unknown" value="0" />
82        <Enumeration label="WDG_Reset" value="1" />
83        <Enumeration label="Assertion_Fail" value="2" />
84        <Enumeration label="MCU_Start" value="3" />
85        <Enumeration label="LowSeverity_Unknown" value="4" />
86        <Enumeration label="MediumSeverity_Unknown" value="5" />
87        <Enumeration label="HighSeverity_Unknown" value="6" />

```

```

88 <Enumeration label="Failed_Start_Of_Execution" value="7" />
89 <Enumeration label="Temp1_Timeout_Temporary" value="100" />
90 <Enumeration label="Temp2_Timeout_Temporary" value="101" />
91 <Enumeration label="Temp1_Timeout_Permanent" value="102" />
92 <Enumeration label="Temp2_Timeout_Permanent" value="103" />
93 <Enumeration label="Temp1_UnexpectedValue_Temporary" value="104" />
94 <Enumeration label="Temp2_UnexpectedValue_Temporary" value="105" />
95 <Enumeration label="TempDelta_UnexpectedValue_Temporary" value="106" />
96 <Enumeration label="Temp1_UnexpectedValue_Permanent" value="107" />
97 <Enumeration label="Temp2_UnexpectedValue_Permanent" value="108" />
98 <Enumeration label="TempDelta_UnexpectedValue_Permanent" value="109" />
99 <Enumeration label="Subsystem_Overheat_Temp1" value="110" />
100 <Enumeration label="Subsystem_Overheat_Temp2" value="111" />
101 </EnumerationList>
102 </EnumeratedParameterType>
103 <EnumeratedParameterType name="PMON_Definition_Check_ID_Type">
104   <UnitSet />
105   <IntegerDataEncoding sizeInBits="8" />
106   <EnumerationList>
107     <Enumeration label="Expected_Value_Check" value="0" />
108     <Enumeration label="Limit_Check" value="1" />
109     <Enumeration label="Delta_Check" value="2" />
110   </EnumerationList>
111 </EnumeratedParameterType>
112 <EnumeratedParameterType name="PMON_Check_Status_Type">
113   <UnitSet />
114   <IntegerDataEncoding sizeInBits="8" />
115   <EnumerationList>
116     <Enumeration label="OK" value="0" />
117     <Enumeration label="Unchecked" value="1" />
118     <Enumeration label="Invalid" value="2" />
119     <Enumeration label="Unexpected_Value" value="3" />
120     <Enumeration label="Below_Low_Limit" value="4" />
121     <Enumeration label="Above_High_Limit" value="5" />
122     <Enumeration label="Below_Low_Threshold" value="6" />
123     <Enumeration label="Above_High_Threshold" value="7" />
124   </EnumerationList>
125 </EnumeratedParameterType>
126 <AggregateParameterType name="PMON_Definition_Type">
127   <MemberList>
128     <Member name="PMON_ID" typeRef="uint16_t" />
129     <Member name="Monitored_Parameter_ID" typeRef="uint16_t" />
130     <Member name="Validity_Parameter_ID" typeRef="uint16_t" />
131     <Member name="Mask" typeRef="uint64_t" />
132     <Member name="Expected_Value" typeRef="uint64_t" />
133     <Member name="Monitoring_Interval" typeRef="uint32_t" />
134     <Member name="Check_Status" typeRef="PMON_Check_Status_Type" />
135     <Member name="Repetition_Number" typeRef="uint8_t" />
136     <Member name="Check_Type" typeRef="PMON_Definition_Check_ID_Type" />
137   </MemberList>
138 </AggregateParameterType>
139 <AggregateParameterType name="PMON_Expected_Value_Check_uint64_Type">
140   <MemberList>
141     <Member name="Mask" typeRef="uint64_t" />
142     <Member name="Expected_Value" typeRef="uint64_t" />
143     <Member name="Event_Definition_ID" typeRef="Event_Definition_ID_Type" />
144   </MemberList>
145 </AggregateParameterType>
146 <AggregateParameterType name="PMON_Expected_Value_Check_Temperature_Status_Type">
147   <MemberList>
148     <Member name="Mask" typeRef="uint64_t" />
149     <Member name="Expected_Value" typeRef="Temperature_Status_Type" />
150     <Member name="Event_Definition_ID" typeRef="Event_Definition_ID_Type" />
151   </MemberList>
152 </AggregateParameterType>
153 <AggregateParameterType name="PMON_Limit_Check_Float_Type">
154   <MemberList>
155     <Member name="Low_Limit" typeRef="float_t" />
156     <Member name="Low_Event" typeRef="Event_Definition_ID_Type" />
157     <Member name="High_Limit" typeRef="float_t" />
158     <Member name="High_Event" typeRef="Event_Definition_ID_Type" />
159   </MemberList>
160 </AggregateParameterType>
161 <AggregateParameterType name="PMON_Limit_Check_Double_Type">
162   <MemberList>
163     <Member name="Low_Limit" typeRef="double_t" />
164     <Member name="Low_Event" typeRef="Event_Definition_ID_Type" />
165     <Member name="High_Limit" typeRef="double_t" />
166     <Member name="High_Event" typeRef="Event_Definition_ID_Type" />
167   </MemberList>
168 </AggregateParameterType>
169 <StringParameterType name="TC_Request_Type">
170   <UnitSet />
171   <StringDataEncoding>
172     <SizeInBits>
173       <Fixed>
174         <FixedValue>512</FixedValue>
175       </Fixed>
176     </SizeInBits>
177   </StringDataEncoding>
178 </StringParameterType>
179 <AggregateParameterType name="EventAction_Definition_Type">
180   <MemberList>
181     <Member name="APID" typeRef="uint16_t" />
182     <Member name="Event_Definition_ID" typeRef="Event_Definition_ID_Type" />
183     <Member name="EventAction_Definition_ID" typeRef="uint16_t" />
184     <Member name="Enabled" typeRef="bool8_t" />

```

```

185     <Member name="TC" typeRef="TC_Request_Type" />
186   </MemberList>
187 </AggregateParameterType>
188 <AggregateParameterType name="EventAction_Status_Type">
189   <MemberList>
190     <Member name="APID" typeRef="uint16_t" />
191     <Member name="Event_Definition_ID" typeRef="Event_Definition_ID_Type" />
192     <Member name="EventAction_Definition_ID" typeRef="uint16_t" />
193     <Member name="Enabled" typeRef="bool8_t" />
194   </MemberList>
195 </AggregateParameterType>
196 <ArrayParameterType name="EventAction_List_Type" arrayTypeRef="EventAction_Status_Type"
197   numberofDimensions="1">
198   <DimensionList>
199     <Dimension>
200       <StartingIndex>
201         <FixedValue>0</FixedValue>
202       </StartingIndex>
203       <EndingIndex>
204         <DynamicValue>
205           <ParameterInstanceRef parameterRef="EventAction_Count" />
206           <LinearAdjustment intercept="-1" />
207         </DynamicValue>
208       </EndingIndex>
209     </Dimension>
210   </DimensionList>
211 </ArrayParameterType>
212 <AggregateParameterType name="PMON_Check_Transition_Type">
213   <MemberList>
214     <Member name="PMON_ID" typeRef="uint16_t" />
215     <Member name="Monitored_Parameter_ID" typeRef="uint16_t" />
216     <Member name="Check_Type" typeRef="PMON_Definition_Check_ID_Type" />
217     <Member name="Value" typeRef="float_t" />
218     <Member name="Previous_Check_Status" typeRef="PMON_Check_Status_Type" />
219     <Member name="Current_Check_Status" typeRef="PMON_Check_Status_Type" />
220     <Member name="Timestamp" typeRef="uint32_t" />
221   </MemberList>
222 </AggregateParameterType>
223 <ArrayParameterType name="PMON_Check_Transition_List_Type" arrayTypeRef="PMON_Check_Transition_Type"
224   numberofDimensions="1">
225   <DimensionList>
226     <Dimension>
227       <StartingIndex>
228         <FixedValue>0</FixedValue>
229       </StartingIndex>
230       <EndingIndex>
231         <DynamicValue>
232           <ParameterInstanceRef parameterRef="PMON_Check_Transition_Count" />
233           <LinearAdjustment intercept="-1" />
234         </DynamicValue>
235       </EndingIndex>
236     </Dimension>
237   </DimensionList>
238 </ArrayParameterType>
239 </ParameterTypeSet>
240 <ParameterSet>
241   <Parameter name="ECSS_Version" parameterTypeRef="uint8_t" />
242   <Parameter name="ECSS_Service_Type" parameterTypeRef="uint8_t" />
243   <Parameter name="ECSS_Message_Type" parameterTypeRef="uint8_t" />
244   <Parameter name="ECSS_MessageTypeCounter" parameterTypeRef="uint16_t" />
245   <Parameter name="ECSS_03_Housekeeping_Structure_ID" parameterTypeRef="uint8_t" />
246   <Parameter name="Temperature_1" parameterTypeRef="Temperature_Type" />
247   <Parameter name="Temperature_2" parameterTypeRef="Temperature_Type" />
248   <Parameter name="Status_Temperature_1" parameterTypeRef="Temperature_Status_Type" />
249   <Parameter name="Status_Temperature_2" parameterTypeRef="Temperature_Status_Type" />
250   <Parameter name="Temperature_Internal" parameterTypeRef="Temperature_Type" />
251   <Parameter name="MCU_Ticks" parameterTypeRef="Tick_Type" />
252   <Parameter name="Event_Definition_ID" parameterTypeRef="Event_Definition_ID_Type" />
253   <Parameter name="Verification_ECSS_Version" parameterTypeRef="Packet_Version_Type" />
254   <Parameter name="Verification_Packet_Type" parameterTypeRef="Packet_Type_Type" />
255   <Parameter name="Verification_APID" parameterTypeRef="APID_Type" />
256   <Parameter name="Verification_Packet_Sequence_Count" parameterTypeRef="Packet_Sequence_Count_Type" />
257   <Parameter name="Verification_Code" parameterTypeRef="uint16_t" />
258   <Parameter name="PMON_Monitoring_Definition_Count" parameterTypeRef="uint8_t" />
259   <Parameter name="PMON_Monitoring_Definition" parameterTypeRef="PMON_Definition_Type" />
260   <Parameter name="PMON_Expected_Value_uint64"
261     parameterTypeRef="PMON_Expected_Value_Check_uint64_Type" />
262   <Parameter name="PMON_Expected_Value_Check_Temperature_Status"
263     parameterTypeRef="PMON_Expected_Value_Check_Temperature_Status_Type" />
264   <Parameter name="PMON_Limit_Check_Float" parameterTypeRef="PMON_Limit_Check_Float_Type" />
265   <Parameter name="PMON_Limit_Check_Double" parameterTypeRef="PMON_Limit_Check_Double_Type" />
266   <Parameter name="PMON_Check_Transition_Count" parameterTypeRef="uint8_t" />
267   <Parameter name="PMON_Check_Transition_List" parameterTypeRef="PMON_Check_Transition_List_Type" />
268   <Parameter name="EventAction_Count" parameterTypeRef="uint16_t" />
269   <Parameter name="EventAction_List" parameterTypeRef="EventAction_List_Type" />
270   <Parameter name="EventAction_Definition" parameterTypeRef="EventAction_Definition_Type" />
271 </ParameterSet>
272 <AlgorithmSet>
273   <CustomAlgorithm name="EventTranscriberAlgorithm">
274     <AncillaryDataSet>
275       <AncillaryData name="Yamcs:AlgorithmMandatoryInput">eventID</AncillaryData>
276       <AncillaryData name="Yamcs:AlgorithmMandatoryInput">messageType</AncillaryData>
277     </AncillaryDataSet>
278     <AlgorithmText language="JavaScript">if (eventID == null) return; if (messageType.value == 1) {
279       Yamcs.watch("ECSS_Event_Parser", "ECSS_Informative_Event", "ECSS informative event "
280       eventID.rawValue + " " + eventID.value); } else if (messageType.value == 2) {
281       Yamcs.warning("ECSS_Event_Parser", "ECSS_LowSev_Anomaly", "ECSS low severity anomaly " +

```

```

282     eventID.rawValue + " " + eventID.value); } else if (messageType.value == 3) {
283         Yamcs.distress("ECSS_Event_Parser", "ECSS_MedSev_Anomaly", "ECSS medium severity anomaly " +
284         eventID.rawValue + " " + eventID.value); } else if (messageType.value == 4) {
285         Yamcs.critical("ECSS_Event_Parser", "ECSS_HiSev_Anomaly", "ECSS high severity anomaly " +
286         eventID.rawValue + " " + eventID.value); }</AlgorithmText>
287     <InputSet>
288         <InputParameterInstanceRef parameterRef="Event_Definition_ID" inputName="eventID" />
289         <InputParameterInstanceRef parameterRef="ECSS_Message_Type" inputName="messageType" />
290     </InputSet>
291     <TriggerSet>
292         <OnParameterUpdateTrigger parameterRef="Event_Definition_ID" />
293     </TriggerSet>
294     </CustomAlgorithm>
295 </AlgorithmSet>
296 <ContainerSet>
297     <SequenceContainer abstract="true" name="ECSSTM">
298         <EntryList>
299             <ParameterRefEntry parameterRef="ECSS_Version" />
300             <ParameterRefEntry parameterRef="ECSS_Service_Type" />
301             <ParameterRefEntry parameterRef="ECSS_Message_Type" />
302             <ParameterRefEntry parameterRef="ECSS_MessageTypeCounter" />
303         </EntryList>
304     </SequenceContainer>
305     <SequenceContainer name="Housekeeping">
306         <EntryList>
307             <ParameterRefEntry parameterRef="ECSS_03_Housekeeping_Structure_ID" />
308         </EntryList>
309         <BaseContainer containerRef="ECSSTM">
310             <RestrictionCriteria>
311                 <ComparisonList>
312                     <Comparison value="3" parameterRef="ECSS_Service_Type" />
313                     <Comparison value="25" parameterRef="ECSS_Message_Type" />
314                 </ComparisonList>
315             </RestrictionCriteria>
316         </BaseContainer>
317     </SequenceContainer>
318     <SequenceContainer name="Housekeeping1">
319         <EntryList>
320             <ParameterRefEntry parameterRef="Temperature_1" />
321             <ParameterRefEntry parameterRef="Status_Temperature_1" />
322             <ParameterRefEntry parameterRef="Temperature_2" />
323             <ParameterRefEntry parameterRef="Status_Temperature_2" />
324             <ParameterRefEntry parameterRef="Temperature_Internal" />
325             <ParameterRefEntry parameterRef="MCU_Ticks" />
326         </EntryList>
327         <BaseContainer containerRef="Housekeeping">
328             <RestrictionCriteria>
329                 <ComparisonList>
330                     <Comparison value="1" parameterRef="ECSS_03_Housekeeping_Structure_ID" />
331                 </ComparisonList>
332             </RestrictionCriteria>
333         </BaseContainer>
334     </SequenceContainer>
335     <SequenceContainer name="EventReporting">
336         <EntryList>
337             <ParameterRefEntry parameterRef="Event_Definition_ID" />
338         </EntryList>
339         <BaseContainer containerRef="ECSSTM">
340             <RestrictionCriteria>
341                 <ComparisonList>
342                     <Comparison value="5" parameterRef="ECSS_Service_Type" />
343                     <Comparison comparisonOperator=">=" value="1" parameterRef="ECSS_Message_Type" />
344                     <Comparison comparisonOperator="<=" value="4" parameterRef="ECSS_Message_Type" />
345                 </ComparisonList>
346             </RestrictionCriteria>
347         </BaseContainer>
348     </SequenceContainer>
349     <SequenceContainer name="VerificationReport">
350         <EntryList>
351             <ParameterRefEntry parameterRef="Verification_ECSS_Version" />
352             <ParameterRefEntry parameterRef="Verification_Packet_Type" />
353             <ParameterRefEntry parameterRef="Verification_APID">
354                 <LocationInContainerInBits referenceLocation="previousEntry">
355                     <FixedValue>1</FixedValue>
356                 </LocationInContainerInBits>
357             </ParameterRefEntry>
358             <ParameterRefEntry parameterRef="Verification_Packet_Sequence_Count">
359                 <LocationInContainerInBits referenceLocation="previousEntry">
360                     <FixedValue>2</FixedValue>
361                 </LocationInContainerInBits>
362             </ParameterRefEntry>
363             <ParameterRefEntry parameterRef="Verification_Code" />
364         </EntryList>
365         <BaseContainer containerRef="ECSSTM">
366             <RestrictionCriteria>
367                 <ComparisonList>
368                     <Comparison value="1" parameterRef="ECSS_Service_Type" />
369                 </ComparisonList>
370             </RestrictionCriteria>
371         </BaseContainer>
372     </SequenceContainer>
373     <SequenceContainer name="MonitoringDefinitions">
374         <EntryList>
375             <ParameterRefEntry parameterRef="PMON_Monitoring_Definition_Count" />
376             <ParameterRefEntry parameterRef="PMON_Monitoring_Definition" />
377         </EntryList>
378     <BaseContainer containerRef="ECSSTM">
```

```

379 <RestrictionCriteria>
380   <ComparisonList>
381     <Comparison value="12" parameterRef="ECSS_Service_Type" />
382     <Comparison value="9" parameterRef="ECSS_Message_Type" />
383   </ComparisonList>
384 </RestrictionCriteria>
385 </BaseContainer>
386 </SequenceContainer>
387 <SequenceContainer name="MonitoringDefinition_ExpectedValue_TemperatureStatus">
388   <EntryList>
389     <ParameterRefEntry parameterRef="PMON_Expected_Value_Check_Temperature_Status" />
390   </EntryList>
391 <BaseContainer containerRef="MonitoringDefinitions">
392   <RestrictionCriteria>
393     <ComparisonList>
394       <Comparison value="1" parameterRef="PMON_Monitoring_Definition_Count" />
395       <Comparison value="Expected_Value_Check"
396         parameterRef="PMON_Monitoring_Definition.Check_Type" />
397       <Comparison comparisonOperator="&gt;=" value="3"
398         parameterRef="PMON_Monitoring_Definition.Monitored_Parameter_ID" />
399       <Comparison comparisonOperator="&lt;=" value="5"
400         parameterRef="PMON_Monitoring_Definition.Monitored_Parameter_ID" />
401     </ComparisonList>
402   </RestrictionCriteria>
403 </BaseContainer>
404 </SequenceContainer>
405 <SequenceContainer name="MonitoringDefinition_ExpectedValue_uint64">
406   <EntryList>
407     <ParameterRefEntry parameterRef="PMON_Expected_Value_Check_uint64" />
408   </EntryList>
409 <BaseContainer containerRef="MonitoringDefinitions">
410   <RestrictionCriteria>
411     <ComparisonList>
412       <Comparison value="1" parameterRef="PMON_Monitoring_Definition_Count" />
413       <Comparison value="Expected_Value_Check"
414         parameterRef="PMON_Monitoring_Definition.Check_Type" />
415     </ComparisonList>
416   </RestrictionCriteria>
417 </BaseContainer>
418 </SequenceContainer>
419 <SequenceContainer name="MonitoringDefinition_Limit_Float">
420   <EntryList>
421     <ParameterRefEntry parameterRef="PMON_Limit_Check_Float" />
422   </EntryList>
423 <BaseContainer containerRef="MonitoringDefinitions">
424   <RestrictionCriteria>
425     <ComparisonList>
426       <Comparison value="1" parameterRef="PMON_Monitoring_Definition_Count" />
427       <Comparison value="Limit_Check" parameterRef="PMON_Monitoring_Definition.Check_Type" />
428     </ComparisonList>
429   </RestrictionCriteria>
430 </BaseContainer>
431 </SequenceContainer>
432 <SequenceContainer name="PMON_Check_Transition_Report">
433   <EntryList>
434     <ParameterRefEntry parameterRef="PMON_Check_Transition_Count" />
435     <ArrayParameterRefEntry parameterRef="PMON_Check_Transition_List">
436       <DimensionList>
437         <Dimension>
438           <StartingIndex>
439             <FixedValue>0</FixedValue>
440           </StartingIndex>
441           <EndingIndex>
442             <DynamicValue>
443               <ParameterInstanceRef parameterRef="PMON_Check_Transition_Count" />
444               <LinearAdjustment intercept="-1" />
445             </DynamicValue>
446           </EndingIndex>
447         </Dimension>
448       </DimensionList>
449     </ArrayParameterRefEntry>
450   </EntryList>
451 <BaseContainer containerRef="ECSSTM">
452   <RestrictionCriteria>
453     <ComparisonList>
454       <Comparison value="12" parameterRef="ECSS_Service_Type" />
455       <Comparison value="12" parameterRef="ECSS_Message_Type" />
456     </ComparisonList>
457   </RestrictionCriteria>
458 </BaseContainer>
459 </SequenceContainer>
460 <SequenceContainer name="EventAction_List">
461   <EntryList>
462     <ParameterRefEntry parameterRef="EventAction_Count" />
463     <ArrayParameterRefEntry parameterRef="EventAction_List">
464       <DimensionList>
465         <Dimension>
466           <StartingIndex>
467             <FixedValue>0</FixedValue>
468           </StartingIndex>
469           <EndingIndex>
470             <DynamicValue>
471               <ParameterInstanceRef parameterRef="EventAction_Count" />
472               <LinearAdjustment intercept="-1" />
473             </DynamicValue>
474           </EndingIndex>
475         </Dimension>

```

```

476      </DimensionList>
477    </ArrayParameterRefEntry>
478  </EntryList>
479  <BaseContainer containerRef="ECSSSTM">
480    <RestrictionCriteria>
481      <ComparisonList>
482        <Comparison value="19" parameterRef="ECSS_Service_Type" />
483        <Comparison value="7" parameterRef="ECSS_Message_Type" />
484      </ComparisonList>
485    </RestrictionCriteria>
486  </BaseContainer>
487 </SequenceContainer>
488 <SequenceContainer name="EventAction_Request_List">
489   <EntryList>
490     <ParameterRefEntry parameterRef="EventAction_Definition">
491       <LocationInContainerInBits referenceLocation="previousEntry">
492         <FixedValue>8</FixedValue>
493       </LocationInContainerInBits>
494     </ParameterRefEntry>
495   </EntryList>
496   <BaseContainer containerRef="ECSSSTM">
497     <RestrictionCriteria>
498       <ComparisonList>
499         <Comparison value="19" parameterRef="ECSS_Service_Type" />
500         <Comparison value="11" parameterRef="ECSS_Message_Type" />
501       </ComparisonList>
502     </RestrictionCriteria>
503   </BaseContainer>
504 </SequenceContainer>
505 </ContainerSet>
506 </TelemetryMetaDataSet>
507 <CommandMetaDataSet>
508   <ArgumentTypeSet>
509     <IntegerArgumentType name="uint4_t" signed="false">
510       <UnitSet />
511       <IntegerDataEncoding sizeInBits="4" />
512     </IntegerArgumentType>
513     <IntegerArgumentType name="uint8_t" signed="false">
514       <UnitSet />
515       <IntegerDataEncoding sizeInBits="8" />
516     </IntegerArgumentType>
517     <IntegerArgumentType name="uint16_t" signed="false">
518       <UnitSet />
519       <IntegerDataEncoding sizeInBits="16" />
520     </IntegerArgumentType>
521     <IntegerArgumentType name="uint64_t" signed="false">
522       <UnitSet />
523       <IntegerDataEncoding sizeInBits="64" />
524     </IntegerArgumentType>
525     <FloatArgumentType name="float_t">
526       <UnitSet />
527       <FloatDataEncoding sizeInBits="32" />
528     </FloatArgumentType>
529     <IntegerArgumentType name="interval_ms_16" signed="false">
530       <UnitSet>
531         <Unit>milliseconds</Unit>
532       </UnitSet>
533       <IntegerDataEncoding sizeInBits="16" />
534     </IntegerArgumentType>
535     <EnumeratedArgumentType name="AcubeSAT_ECSS_03_Definitions">
536       <UnitSet />
537       <IntegerDataEncoding sizeInBits="8" />
538       <EnumerationList>
539         <Enumeration value="1" label="Temperature Housekeeping" />
540       </EnumerationList>
541     </EnumeratedArgumentType>
542     <EnumeratedArgumentType name="Event_Definition_ID_Type">
543       <UnitSet />
544       <IntegerDataEncoding sizeInBits="16" />
545       <EnumerationList>
546         <Enumeration label="Informative_Uncertain" value="0" />
547         <Enumeration label="WWDG_Reset" value="1" />
548         <Enumeration label="Assertion_Fail" value="2" />
549         <Enumeration label="MCU_Start" value="3" />
550         <Enumeration label="LowSeverity_Uncertain" value="4" />
551         <Enumeration label="MediumSeverity_Uncertain" value="5" />
552         <Enumeration label="HighSeverity_Uncertain" value="6" />
553         <Enumeration label="Failed_Start_Of_Execution" value="7" />
554         <Enumeration label="Temp1_Timeout_Temporary" value="100" />
555         <Enumeration label="Temp2_Timeout_Temporary" value="101" />
556         <Enumeration label="Temp1_Timeout_Permanent" value="102" />
557         <Enumeration label="Temp2_Timeout_Permanent" value="103" />
558         <Enumeration label="Temp1_UnexpectedValue_Temporary" value="104" />
559         <Enumeration label="Temp2_UnexpectedValue_Temporary" value="105" />
560         <Enumeration label="TempDelta_UnexpectedValue_Temporary" value="106" />
561         <Enumeration label="Temp1_UnexpectedValue_Permanent" value="107" />
562         <Enumeration label="Temp2_UnexpectedValue_Permanent" value="108" />
563         <Enumeration label="TempDelta_UnexpectedValue_Permanent" value="109" />
564         <Enumeration label="Subsystem_Overheat_Temp1" value="110" />
565         <Enumeration label="Subsystem_Overheat_Temp2" value="111" />
566       </EnumerationList>
567     </EnumeratedArgumentType>
568     <StringArgumentType name="ECSS_Function_Name_Type">
569       <UnitSet />
570       <StringDataEncoding>
571         <SizeInBits>
572           <Fixed>

```

```

573      <FixedValue>128</FixedValue>
574    </Fixed>
575  </SizeInBits>
576  </StringDataEncoding>
577</StringArgumentType>
578<StringArgumentType name="ECSS_Function_Data_Type">
579  <UnitSet />
580  <StringDataEncoding>
581    <SizeInBits>
582      <Fixed>
583        <FixedValue>128</FixedValue>
584      </Fixed>
585    </SizeInBits>
586  </StringDataEncoding>
587</StringArgumentType>
588<BinaryArgumentType name="TC_Request_Type">
589  <UnitSet />
590  <BinaryDataEncoding>
591    <SizeInBits>
592      <Fixed>
593        <FixedValue>512</FixedValue>
594      </Fixed>
595    </SizeInBits>
596  </BinaryDataEncoding>
597</BinaryArgumentType>
598<EnumeratedArgumentType name="Temperature_Status_Type">
599  <UnitSet />
600  <IntegerDataEncoding sizeInBits="8" />
601  <EnumerationList>
602    <Enumeration label="Nominal" value="1" />
603    <Enumeration label="Disabled" value="0" />
604  </EnumerationList>
605</EnumeratedArgumentType>
606</ArgumentTypeSet>
607<MetaCommandSet>
608  <MetaCommand name="ECSSTC" abstract="true">
609    <ArgumentList>
610      <Argument argumentTypeRef="uint4_t" name="ECSS_Version" />
611      <Argument argumentTypeRef="uint4_t" name="ECSS_Acknowledgement_Flags" />
612      <Argument argumentTypeRef="uint8_t" name="ECSS_Service_Type" />
613      <Argument argumentTypeRef="uint8_t" name="ECSS_Message_Type" />
614      <Argument argumentTypeRef="uint16_t" name="ECSS_Source_ID" />
615    </ArgumentList>
616    <CommandContainer name="ECSSTCCContainer">
617      <EntryList>
618        <ArgumentRefEntry argumentRef="ECSS_Version" />
619        <ArgumentRefEntry argumentRef="ECSS_Acknowledgement_Flags" />
620        <ArgumentRefEntry argumentRef="ECSS_Service_Type" />
621        <ArgumentRefEntry argumentRef="ECSS_Message_Type" />
622        <ArgumentRefEntry argumentRef="ECSS_Source_ID" />
623      </EntryList>
624    </CommandContainer>
625  </MetaCommand>
626  <MetaCommand name="AcubeSATTC" abstract="true">
627    <BaseMetaCommand metaCommandRef="ECSSTC">
628      <ArgumentAssignmentList>
629        <ArgumentAssignment argumentName="ECSS_Version" argumentValue="2" />
630        <ArgumentAssignment argumentName="ECSS_Acknowledgement_Flags" argumentValue="0b1111" />
631        <ArgumentAssignment argumentName="ECSS_Source_ID" argumentValue="0" />
632      </ArgumentAssignmentList>
633    </BaseMetaCommand>
634    <ArgumentList />
635    <CommandContainer name="AcubeSATTC">
636      <EntryList />
637      <BaseContainer containerRef="ECSSTCCContainer" />
638    </CommandContainer>
639    <VerifierSet>
640      <FailedVerifier>
641        <ContainerRef containerRef="VerificationReport" />
642        <CheckWindow timeToStopChecking="PT2S" />
643      </FailedVerifier>
644    </VerifierSet>
645  </MetaCommand>
646  <MetaCommand name="Test">
647    <BaseMetaCommand metaCommandRef="AcubeSATTC">
648      <ArgumentAssignmentList>
649        <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="17" />
650        <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="1" />
651      </ArgumentAssignmentList>
652    </BaseMetaCommand>
653    <CommandContainer name="TestContainer">
654      <EntryList />
655      <BaseContainer containerRef="AcubeSATTC" />
656    </CommandContainer>
657    <DefaultSignificance consequenceLevel="none" />
658  </MetaCommand>
659  <MetaCommand name="ST03_Enable" shortDescription="Enable the generation of housekeeping reports">
660    <BaseMetaCommand metaCommandRef="AcubeSATTC">
661      <ArgumentAssignmentList>
662        <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="3" />
663        <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="5" />
664      </ArgumentAssignmentList>
665    </BaseMetaCommand>
666    <ArgumentList>
667      <Argument argumentTypeRef="AcubeSAT_ECSS_03_Definitions" name="Definition"
668        shortDescription="Housekeeping Definition ID" />
669    </ArgumentList>

```

```

670 <CommandContainer name="ST03_Enable">
671   <EntryList>
672     <FixedValueEntry name="N" binaryValue="01" sizeInBits="8" />
673     <ArgumentRefEntry argumentRef="Definition" />
674   </EntryList>
675   <BaseContainer containerRef="AcubeSATTC" />
676 </CommandContainer>
677 <DefaultSignificance consequenceLevel="warning" />
678 </MetaCommand>
679 <MetaCommand name="ST03_Disable" shortDescription="Disable the generation of housekeeping reports">
680   <BaseMetaCommand metaCommandRef="AcubeSATTC">
681     <ArgumentAssignmentList>
682       <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="3" />
683       <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="6" />
684     </ArgumentAssignmentList>
685   </BaseMetaCommand>
686   <ArgumentList>
687     <Argument argumentTypeRef="AcubeSAT_ECSS_03_Definitions" name="Definition"
688       shortDescription="Housekeeping Definition ID" />
689   </ArgumentList>
690   <CommandContainer name="ST03_Disable">
691     <EntryList>
692       <FixedValueEntry name="N" binaryValue="01" sizeInBits="8" />
693       <ArgumentRefEntry argumentRef="Definition" />
694     </EntryList>
695     <BaseContainer containerRef="AcubeSATTC" />
696   </CommandContainer>
697   <DefaultSignificance consequenceLevel="warning" />
698 </MetaCommand>
699 <MetaCommand name="ST03_SetInterval"
700   shortDescription="Set the generation interval of housekeeping reports">
701   <BaseMetaCommand metaCommandRef="AcubeSATTC">
702     <ArgumentAssignmentList>
703       <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="3" />
704       <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="31" />
705     </ArgumentAssignmentList>
706   </BaseMetaCommand>
707   <ArgumentList>
708     <Argument argumentTypeRef="AcubeSAT_ECSS_03_Definitions" name="Definition"
709       shortDescription="Housekeeping Definition ID" />
710     <Argument argumentTypeRef="interval_ms_16" name="Collection_Interval" />
711   </ArgumentList>
712   <CommandContainer name="ST03_SetInterval">
713     <EntryList>
714       <FixedValueEntry name="N" binaryValue="01" sizeInBits="8" />
715       <ArgumentRefEntry argumentRef="Definition" />
716       <ArgumentRefEntry argumentRef="Collection_Interval" />
717     </EntryList>
718     <BaseContainer containerRef="AcubeSATTC" />
719   </CommandContainer>
720   <DefaultSignificance consequenceLevel="watch" />
721 </MetaCommand>
722 <MetaCommand name="TestError" shortDescription="An erroneous command to test parsing errors">
723   <BaseMetaCommand metaCommandRef="AcubeSATTC">
724     <ArgumentAssignmentList>
725       <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="95" />
726       <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="60" />
727     </ArgumentAssignmentList>
728   </BaseMetaCommand>
729   <CommandContainer name="TestError">
730     <EntryList />
731     <BaseContainer containerRef="AcubeSATTC" />
732   </CommandContainer>
733   <DefaultSignificance consequenceLevel="none" />
734 </MetaCommand>
735 <MetaCommand name="ST08_Function" abstract="true" shortDescription="Run an ST[08] on-board function">
736   <ArgumentList>
737     <Argument argumentTypeRef="ECSS_Function_Name_Type" name="ECSS_Function_Name"
738       shortDescription="Function definition name" />
739     <Argument argumentTypeRef="ECSS_Function_Data_Type" name="ECSS_Function_Data"
740       shortDescription="Auxiliary data to include" initialValue="0" />
741   </ArgumentList>
742   <BaseMetaCommand metaCommandRef="AcubeSATTC">
743     <ArgumentAssignmentList>
744       <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="8" />
745       <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="1" />
746     </ArgumentAssignmentList>
747   </BaseMetaCommand>
748   <CommandContainer name="ST08_Function">
749     <EntryList>
750       <ArgumentRefEntry argumentRef="ECSS_Function_Name" />
751       <ArgumentRefEntry argumentRef="ECSS_Function_Data" />
752     </EntryList>
753     <BaseContainer containerRef="AcubeSATTC" />
754   </CommandContainer>
755   <DefaultSignificance consequenceLevel="none" />
756 </MetaCommand>
757 <MetaCommand name="Log" shortDescription="Output a string via UART">
758   <BaseMetaCommand metaCommandRef="ST08_Function">
759     <ArgumentAssignmentList>
760       <ArgumentAssignment argumentName="ECSS_Function_Name" argumentValue="log" />
761     </ArgumentAssignmentList>
762   </BaseMetaCommand>
763   <CommandContainer name="Log">
764     <EntryList />
765     <BaseContainer containerRef="ST08_Function" />
766   </CommandContainer>

```

```

767 <DefaultSignificance consequenceLevel="none" />
768 </MetaCommand>
769 <MetaCommand name="MCU_Reset" shortDescription="Perform a soft reset of the MCU">
770   <BaseMetaCommand metaCommandRef="ST08_Function">
771     <ArgumentAssignmentList>
772       <ArgumentAssignment argumentName="ECSS_Function_Name" argumentValue="restart" />
773     </ArgumentAssignmentList>
774   </BaseMetaCommand>
775   <CommandContainer name="MCU_Reset">
776     <EntryList />
777     <BaseContainer containerRef="ST08_Function" />
778   </CommandContainer>
779   <DefaultSignificance consequenceLevel="warning" />
780 </MetaCommand>
781 <MetaCommand name="Sensor1_Restart" shortDescription="Restart Sensor 1">
782   <BaseMetaCommand metaCommandRef="ST08_Function">
783     <ArgumentAssignmentList>
784       <ArgumentAssignment argumentName="ECSS_Function_Name" argumentValue="restart_sensor" />
785       <ArgumentAssignment argumentName="ECSS_Function_Data" argumentValue="1" />
786     </ArgumentAssignmentList>
787   </BaseMetaCommand>
788   <CommandContainer name="Sensor1_Restart">
789     <EntryList />
790     <BaseContainer containerRef="ST08_Function" />
791   </CommandContainer>
792   <DefaultSignificance consequenceLevel="watch" />
793 </MetaCommand>
794 <MetaCommand name="Sensor2_Restart" shortDescription="Restart Sensor 2">
795   <BaseMetaCommand metaCommandRef="ST08_Function">
796     <ArgumentAssignmentList>
797       <ArgumentAssignment argumentName="ECSS_Function_Name" argumentValue="restart_sensor" />
798       <ArgumentAssignment argumentName="ECSS_Function_Data" argumentValue="2" />
799     </ArgumentAssignmentList>
800   </BaseMetaCommand>
801   <CommandContainer name="Sensor2_Restart">
802     <EntryList />
803     <BaseContainer containerRef="ST08_Function" />
804   </CommandContainer>
805   <DefaultSignificance consequenceLevel="watch" />
806 </MetaCommand>
807 <MetaCommand name="Add_Bogus_Temperature"
808 shortDescription="Start adding a fake value to the temperature sensors">
809   <BaseMetaCommand metaCommandRef="ST08_Function">
810     <ArgumentAssignmentList>
811       <ArgumentAssignment argumentName="ECSS_Function_Name" argumentValue="bogus_temp" />
812       <ArgumentAssignment argumentName="ECSS_Function_Data" argumentValue="0" />
813     </ArgumentAssignmentList>
814   </BaseMetaCommand>
815   <CommandContainer name="Add_Bogus_Temperature">
816     <EntryList />
817     <BaseContainer containerRef="ST08_Function" />
818   </CommandContainer>
819   <DefaultSignificance consequenceLevel="watch" />
820 </MetaCommand>
821 <MetaCommand name="OnBoardMonitoring_Enable"
822 shortDescription="Enable the entire on-board monitoring function">
823   <BaseMetaCommand metaCommandRef="AcubeSATTC">
824     <ArgumentAssignmentList>
825       <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="12" />
826       <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="15" />
827     </ArgumentAssignmentList>
828   </BaseMetaCommand>
829   <CommandContainer name="OnBoardMonitoring_Enable">
830     <EntryList />
831     <BaseContainer containerRef="AcubeSATTC" />
832   </CommandContainer>
833   <DefaultSignificance consequenceLevel="distress" />
834 </MetaCommand>
835 <MetaCommand name="OnBoardMonitoring_Disable"
836 shortDescription="Disable the entire on-board monitoring function">
837   <BaseMetaCommand metaCommandRef="AcubeSATTC">
838     <ArgumentAssignmentList>
839       <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="12" />
840       <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="16" />
841     </ArgumentAssignmentList>
842   </BaseMetaCommand>
843   <CommandContainer name="OnBoardMonitoring_Disable">
844     <EntryList />
845     <BaseContainer containerRef="AcubeSATTC" />
846   </CommandContainer>
847   <DefaultSignificance consequenceLevel="distress" />
848 </MetaCommand>
849 <MetaCommand name="ST20_SetOne" abstract="true" shortDescription="Set one on-board parameter">
850   <BaseMetaCommand metaCommandRef="AcubeSATTC">
851     <ArgumentAssignmentList>
852       <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="20" />
853       <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="3" />
854     </ArgumentAssignmentList>
855   </BaseMetaCommand>
856   <CommandContainer name="ST20_SetOne">
857     <EntryList>
858       <FixedValueEntry name="N" binaryValue="0001" sizeInBits="16" />
859     </EntryList>
860     <BaseContainer containerRef="AcubeSATTC" />
861   </CommandContainer>
862 </MetaCommand>
863 <MetaCommand name="Set_Temp1_Status" shortDescription="Set the status of temperature sensor 1">

```

```

864 <BaseMetaCommand metaCommandRef="ST20_SetOne" />
865 <ArgumentList>
866   <Argument argumentTypeRef="Temperature_Status_Type" name="Temperature_Status"
867   shortDescription="Temperature sensor status" />
868 </ArgumentList>
869 <CommandContainer name="Set_Temp1_Status">
870   <EntryList>
871     <FixedValueEntry name="N" binaryValue="0003" sizeInBits="16" />
872     <ArgumentRefEntry argumentRef="Temperature_Status" />
873   </EntryList>
874   <BaseContainer containerRef="ST20_SetOne" />
875 </CommandContainer>
876 <DefaultSignificance consequenceLevel="watch" />
877 </MetaCommand>
878 <MetaCommand name="Set_Temp2_Status" shortDescription="Set the status of temperature sensor 2">
879   <BaseMetaCommand metaCommandRef="ST20_SetOne" />
880   <ArgumentList>
881     <Argument argumentTypeRef="Temperature_Status_Type" name="Temperature_Status"
882     shortDescription="Temperature sensor status" />
883   </ArgumentList>
884   <CommandContainer name="Set_Temp2_Status">
885     <EntryList>
886       <FixedValueEntry name="N" binaryValue="0004" sizeInBits="16" />
887       <ArgumentRefEntry argumentRef="Temperature_Status" />
888     </EntryList>
889     <BaseContainer containerRef="ST20_SetOne" />
890   </CommandContainer>
891   <DefaultSignificance consequenceLevel="watch" />
892 </MetaCommand>
893 <MetaCommand name="ST12_ListAllDefinitions"
894 shortDescription="Report the list of all parameter monitoring definitions">
895   <BaseMetaCommand metaCommandRef="AcubeSATTC">
896     <ArgumentAssignmentList>
897       <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="12" />
898       <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="8" />
899     </ArgumentAssignmentList>
900   </BaseMetaCommand>
901   <CommandContainer name="ST12_ListAllDefinitions">
902     <EntryList>
903       <FixedValueEntry name="N" binaryValue="00" sizeInBits="8" />
904     </EntryList>
905     <BaseContainer containerRef="AcubeSATTC" />
906   </CommandContainer>
907   <DefaultSignificance consequenceLevel="none" />
908 </MetaCommand>
909 <MetaCommand name="ST19_ListAllEventAction"
910 shortDescription="Report the list of all event-action definitions">
911   <BaseMetaCommand metaCommandRef="AcubeSATTC">
912     <ArgumentAssignmentList>
913       <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="19" />
914       <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="6" />
915     </ArgumentAssignmentList>
916   </BaseMetaCommand>
917   <CommandContainer name="ST19_ListAllEventAction">
918     <EntryList />
919     <BaseContainer containerRef="AcubeSATTC" />
920   </CommandContainer>
921   <DefaultSignificance consequenceLevel="none" />
922 </MetaCommand>
923 <MetaCommand name="ST19_Enable" shortDescription="Enable the event-action function">
924   <BaseMetaCommand metaCommandRef="AcubeSATTC">
925     <ArgumentAssignmentList>
926       <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="19" />
927       <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="8" />
928     </ArgumentAssignmentList>
929   </BaseMetaCommand>
930   <CommandContainer name="ST19_Enable">
931     <EntryList />
932     <BaseContainer containerRef="AcubeSATTC" />
933   </CommandContainer>
934   <DefaultSignificance consequenceLevel="distress" />
935 </MetaCommand>
936 <MetaCommand name="ST19_ListEventActionRequest"
937 shortDescription="Show an event-action definition with TC requests">
938   <BaseMetaCommand metaCommandRef="AcubeSATTC">
939     <ArgumentAssignmentList>
940       <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="19" />
941       <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="10" />
942     </ArgumentAssignmentList>
943   </BaseMetaCommand>
944   <ArgumentList>
945     <Argument argumentTypeRef="uint16_t" name="Event_Action_Definition_ID" />
946   </ArgumentList>
947   <CommandContainer name="ST19_ListEventActionRequest">
948     <EntryList>
949       <FixedValueEntry name="N" binaryValue="01" sizeInBits="8" />
950       <ArgumentRefEntry argumentRef="Event_Action_Definition_ID" />
951     </EntryList>
952     <BaseContainer containerRef="AcubeSATTC" />
953   </CommandContainer>
954   <DefaultSignificance consequenceLevel="none" />
955 </MetaCommand>
956 <MetaCommand name="ST19_Disable" shortDescription="Disable the event-action function">
957   <BaseMetaCommand metaCommandRef="AcubeSATTC">
958     <ArgumentAssignmentList>
959       <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="19" />
960       <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="9" />

```

```

961      </ArgumentAssignmentList>
962  </BaseMetaCommand>
963  <CommandContainer name="ST19_Disable">
964    <EntryList />
965    <BaseContainer containerRef="AcubeSATTC" />
966  </CommandContainer>
967  <DefaultSignificance consequenceLevel="distress" />
968 </MetaCommand>
969 <MetaCommand name="ST19_Enable_Definition" shortDescription="Enable one event-action definition">
970   <BaseMetaCommand metaCommandRef="AcubeSATTC">
971     <ArgumentAssignmentList>
972       <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="19" />
973       <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="4" />
974     </ArgumentAssignmentList>
975   </BaseMetaCommand>
976   <ArgumentList>
977     <Argument argumentTypeRef="Event_Definition_ID_Type" name="Event_Definition_ID" />
978   </ArgumentList>
979   <CommandContainer name="ST19_Enable_Definition">
980     <EntryList>
981       <FixedValueEntry name="N" binaryValue="0001" sizeInBits="16" />
982       <FixedValueEntry name="APID" binaryValue="0000" sizeInBits="16" />
983       <ArgumentRefEntry argumentRef="Event_Definition_ID" />
984       <FixedValueEntry name="Event_Action_Definition_ID" binaryValue="0000" sizeInBits="16" />
985     </EntryList>
986     <BaseContainer containerRef="AcubeSATTC" />
987   </CommandContainer>
988   <DefaultSignificance consequenceLevel="warning" />
989 </MetaCommand>
990 <MetaCommand name="ST19_Disable_Definition" shortDescription="Disable one event-action definition">
991   <BaseMetaCommand metaCommandRef="AcubeSATTC">
992     <ArgumentAssignmentList>
993       <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="19" />
994       <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="5" />
995     </ArgumentAssignmentList>
996   </BaseMetaCommand>
997   <ArgumentList>
998     <Argument argumentTypeRef="Event_Definition_ID_Type" name="Event_Definition_ID" />
999   </ArgumentList>
1000  <CommandContainer name="ST19_Disable_Definition">
1001    <EntryList>
1002      <FixedValueEntry name="N" binaryValue="0001" sizeInBits="16" />
1003      <FixedValueEntry name="APID" binaryValue="0000" sizeInBits="16" />
1004      <ArgumentRefEntry argumentRef="Event_Definition_ID" />
1005      <FixedValueEntry name="Event_Action_Definition_ID" binaryValue="0000" sizeInBits="16" />
1006    </EntryList>
1007    <BaseContainer containerRef="AcubeSATTC" />
1008  </CommandContainer>
1009  <DefaultSignificance consequenceLevel="warning" />
1010 </MetaCommand>
1011 <MetaCommand name="ST19_Add_Definition" shortDescription="Create one event-action definition">
1012   <BaseMetaCommand metaCommandRef="AcubeSATTC">
1013     <ArgumentAssignmentList>
1014       <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="19" />
1015       <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="1" />
1016     </ArgumentAssignmentList>
1017   </BaseMetaCommand>
1018   <ArgumentList>
1019     <Argument argumentTypeRef="uint16_t" name="APID" initialValue="0" />
1020     <Argument argumentTypeRef="uint16_t" name="Event_Definition_ID" />
1021     <Argument argumentTypeRef="uint16_t" name="Event_Action_Definition_ID" />
1022     <Argument argumentTypeRef="TC_Request_Type" name="TC" />
1023   </ArgumentList>
1024   <CommandContainer name="ST19_Add_Definition">
1025     <EntryList>
1026       <ArgumentRefEntry argumentRef="APID" />
1027       <ArgumentRefEntry argumentRef="Event_Definition_ID" />
1028       <ArgumentRefEntry argumentRef="Event_Action_Definition_ID" />
1029       <ArgumentRefEntry argumentRef="TC" />
1030     </EntryList>
1031     <BaseContainer containerRef="AcubeSATTC" />
1032   </CommandContainer>
1033   <DefaultSignificance consequenceLevel="warning" />
1034 </MetaCommand>
1035 <MetaCommand name="ST19_Delete_Definition" shortDescription="Delete one event-action definition">
1036   <BaseMetaCommand metaCommandRef="AcubeSATTC">
1037     <ArgumentAssignmentList>
1038       <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="19" />
1039       <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="2" />
1040     </ArgumentAssignmentList>
1041   </BaseMetaCommand>
1042   <ArgumentList>
1043     <Argument argumentTypeRef="uint16_t" name="APID" initialValue="0" />
1044     <Argument argumentTypeRef="uint16_t" name="Event_Definition_ID" />
1045     <Argument argumentTypeRef="uint16_t" name="Event_Action_Definition_ID" />
1046   </ArgumentList>
1047   <CommandContainer name="ST19_Delete_Definition">
1048     <EntryList>
1049       <FixedValueEntry name="N" binaryValue="0001" sizeInBits="16" />
1050       <ArgumentRefEntry argumentRef="APID" />
1051       <ArgumentRefEntry argumentRef="Event_Definition_ID" />
1052       <ArgumentRefEntry argumentRef="Event_Action_Definition_ID" />
1053     </EntryList>
1054     <BaseContainer containerRef="AcubeSATTC" />
1055   </CommandContainer>
1056   <DefaultSignificance consequenceLevel="warning" />
1057 </MetaCommand>

```

```

1058 <MetaCommand name="ST12_Enable_Definition"
1059 shortDescription="Enable one on-board parameter monitoring definition">
1060   <BaseMetaCommand metaCommandRef="AcubeSATTC">
1061     <ArgumentAssignmentList>
1062       <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="12" />
1063       <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="1" />
1064     </ArgumentAssignmentList>
1065   </BaseMetaCommand>
1066   <ArgumentList>
1067     <Argument argumentTypeRef="uint16_t" name="PMON_ID" />
1068   </ArgumentList>
1069   <CommandContainer name="ST12_Enable_Definition">
1070     <EntryList>
1071       <FixedValueEntry name="N" binaryValue="01" sizeInBits="8" />
1072       <ArgumentRefEntry argumentRef="PMON_ID" />
1073     </EntryList>
1074     <BaseContainer containerRef="AcubeSATTC" />
1075   </CommandContainer>
1076   <DefaultSignificance consequenceLevel="warning" />
1077 </MetaCommand>
1078 <MetaCommand name="ST12_Disable_Definition"
1079 shortDescription="Disable one on-board parameter monitoring definition">
1080   <BaseMetaCommand metaCommandRef="AcubeSATTC">
1081     <ArgumentAssignmentList>
1082       <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="12" />
1083       <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="2" />
1084     </ArgumentAssignmentList>
1085   </BaseMetaCommand>
1086   <ArgumentList>
1087     <Argument argumentTypeRef="uint16_t" name="PMON_ID" />
1088   </ArgumentList>
1089   <CommandContainer name="ST12_Disable_Definition">
1090     <EntryList>
1091       <FixedValueEntry name="N" binaryValue="01" sizeInBits="8" />
1092       <ArgumentRefEntry argumentRef="PMON_ID" />
1093     </EntryList>
1094     <BaseContainer containerRef="AcubeSATTC" />
1095   </CommandContainer>
1096   <DefaultSignificance consequenceLevel="warning" />
1097 </MetaCommand>
1098 <MetaCommand name="ST12_Add_Definition" abstract="true"
1099 shortDescription="Add a parameter monitoring definition">
1100   <BaseMetaCommand metaCommandRef="AcubeSATTC">
1101     <ArgumentAssignmentList>
1102       <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="12" />
1103       <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="5" />
1104     </ArgumentAssignmentList>
1105   </BaseMetaCommand>
1106   <ArgumentList>
1107     <Argument argumentTypeRef="uint16_t" name="PMON_ID" />
1108     <Argument argumentTypeRef="uint16_t" name="Monitored_Parameter_ID" />
1109     <Argument argumentTypeRef="uint16_t" name="Validity_Parameter_ID" />
1110     <Argument argumentTypeRef="uint64_t" name="Validity_Mask" />
1111     <Argument argumentTypeRef="uint64_t" name="Validity_Expected_Value" />
1112     <Argument argumentTypeRef="interval_ms_16" name="Monitoring_Interval" />
1113     <Argument argumentTypeRef="uint8_t" name="Repetition_Number" />
1114     <Argument argumentTypeRef="uint8_t" name="Check_Type" />
1115   </ArgumentList>
1116   <CommandContainer name="ST12_Add_Definition">
1117     <EntryList>
1118       <FixedValueEntry name="N" binaryValue="01" sizeInBits="8" />
1119       <ArgumentRefEntry argumentRef="PMON_ID" />
1120       <ArgumentRefEntry argumentRef="Monitored_Parameter_ID" />
1121       <ArgumentRefEntry argumentRef="Validity_Parameter_ID" />
1122       <ArgumentRefEntry argumentRef="Validity_Mask" />
1123       <ArgumentRefEntry argumentRef="Validity_Expected_Value" />
1124       <ArgumentRefEntry argumentRef="Monitoring_Interval" />
1125       <ArgumentRefEntry argumentRef="Repetition_Number" />
1126       <ArgumentRefEntry argumentRef="Check_Type" />
1127     </EntryList>
1128     <BaseContainer containerRef="AcubeSATTC" />
1129   </CommandContainer>
1130   <DefaultSignificance consequenceLevel="warning" />
1131 </MetaCommand>
1132 <MetaCommand name="ST12_Add_Definition_Limit"
1133 shortDescription="Add a parameter monitoring definition: Limit check, float type">
1134   <BaseMetaCommand metaCommandRef="ST12_Add_Definition">
1135     <ArgumentAssignmentList>
1136       <ArgumentAssignment argumentName="Check_Type" argumentValue="1" />
1137     </ArgumentAssignmentList>
1138   </BaseMetaCommand>
1139   <ArgumentList>
1140     <Argument argumentTypeRef="float_t" name="Low_Limit" />
1141     <Argument argumentTypeRef="uint16_t" name="Low_Event" />
1142     <Argument argumentTypeRef="float_t" name="High_Limit" />
1143     <Argument argumentTypeRef="uint16_t" name="High_Event" />
1144   </ArgumentList>
1145   <CommandContainer name="ST12_Add_Definition_Limit">
1146     <EntryList>
1147       <ArgumentRefEntry argumentRef="Low_Limit" />
1148       <ArgumentRefEntry argumentRef="Low_Event" />
1149       <ArgumentRefEntry argumentRef="High_Limit" />
1150       <ArgumentRefEntry argumentRef="High_Event" />
1151     </EntryList>
1152     <BaseContainer containerRef="ST12_Add_Definition" />
1153   </CommandContainer>
1154   <DefaultSignificance consequenceLevel="warning" />

```

```

1155
1156 </MetaCommand>
1157 <MetaCommand name="ST12_Add_Definition_Expected_Value"
1158   shortDescription="Add a parameter monitoring definition: Expected value, uint64_t type">
1159   <BaseMetaCommand metaCommandRef="ST12_Add_Definition">
1160     <ArgumentAssignmentList>
1161       <ArgumentAssignment argumentName="Check_Type" argumentValue="0" />
1162     </ArgumentAssignmentList>
1163   </BaseMetaCommand>
1164   <ArgumentList>
1165     <Argument argumentTypeRef="uint64_t" name="Mask" />
1166     <Argument argumentTypeRef="uint64_t" name="Expected_Value" />
1167     <Argument argumentTypeRef="uint16_t" name="Event_ID" />
1168   </ArgumentList>
1169   <CommandContainer name="ST12_Add_Definition_Expected_Value">
1170     <EntryList>
1171       <ArgumentRefEntry argumentRef="Mask" />
1172       <ArgumentRefEntry argumentRef="Expected_Value" />
1173       <ArgumentRefEntry argumentRef="Event_ID" />
1174     </EntryList>
1175     <BaseContainer containerRef="ST12_Add_Definition" />
1176   </CommandContainer>
1177   <DefaultSignificance consequenceLevel="warning" />
1178 </MetaCommand>
1179 <MetaCommand name="ST12_Delete_Definition" shortDescription="Delete a parameter monitoring definition">
1180   <BaseMetaCommand metaCommandRef="AcubeSATTC">
1181     <ArgumentAssignmentList>
1182       <ArgumentAssignment argumentName="ECSS_Service_Type" argumentValue="12" />
1183       <ArgumentAssignment argumentName="ECSS_Message_Type" argumentValue="6" />
1184     </ArgumentAssignmentList>
1185   </BaseMetaCommand>
1186   <ArgumentList>
1187     <Argument argumentTypeRef="uint16_t" name="PMON_ID" />
1188   </ArgumentList>
1189   <CommandContainer name="ST12_Delete_Definition">
1190     <EntryList>
1191       <FixedValueEntry name="N" binaryValue="01" sizeInBits="8" />
1192       <ArgumentRefEntry argumentRef="PMON_ID" />
1193     </EntryList>
1194     <BaseContainer containerRef="AcubeSATTC" />
1195   </CommandContainer>
1196   <DefaultSignificance consequenceLevel="warning" />
1197 </MetaCommand>
1198 </MetaCommandSet>
1199 </CommandMetaData>
</SpaceSystem>

```

ecss-interface.js (fdir-demo-yamcs)

Το αρχείο αυτό είναι το κύριο κομμάτι της εφαρμογής προβολής της βάσης δεδομένων PUS. Συνδέεται με το YAMCS και προβάλλει τα δεδομένα στον χρήστη. Επίσης έχει τη δυνατότητα αποστολής πιλευντολών, όταν χρειάζεται ο μικροελεγκτής να στείλει τα απαραίτητα δεδομένα.

```

1  console.log("hello")
2
3  const Parameters = [
4      "Temp. 1",
5      "Temp. 2",
6      "Temp. Δ",
7      "Temp. 1 Status",
8      "Temp. 2 Status",
9      "Temp. 1+2 Status",
10     "Temp. Internal",
11     "Tick time",
12 ]
13
14 var timeRequest = {
15     type: "time",
16     id: 1,
17     options: {
18         instance: "fdirdemo",
19     },
20 }
21
22 var parameterRequest = {
23     type: "parameters",
24     id: 2,
25     options: {
26         instance: "fdirdemo",
27         processor: "realtime",
28         id: [
29             { name: "/fdirdemo/PMON_Monitoring_Definition" },
30             { name: "/fdirdemo/PMON_Limit_Check_Double" },
31             { name: "/fdirdemo/PMON_Limit_Check_Float" },
32             { name: "/fdirdemo/PMON_Expected_Value_Check_uint64" },
33             { name: "/fdirdemo/PMON_Expected_Value_Check_Temperature_Status" },
34             { name: "/fdirdemo/EventAction_List" },
35             { name: "/fdirdemo/EventAction_Definition" },
36             { name: "/fdirdemo/PMON_Check_Transition_List" },
37         ],
38     },
39 }
40
41 var EventEnumeration = {}
42 var CheckStatusEnumeration = {}
43
44 var websocket = new WebSocket("ws://localhost:8090/api/websocket")
45
46 websocket.onopen = function (event) {
47     setTimeout(function () {
48         websocket.send(JSON.stringify(timeRequest))
49         websocket.send(JSON.stringify(parameterRequest))
50     }, 300)
51 }
52
53 const $timestamp = document.getElementById("timestamp")
54 const $pmonTable = document.getElementById("pmon-table")
55 const $eventActionTable = document.getElementById("event-action-table")
56 const $transitionTable = document.getElementById("check-transition-table")
57
58 let pm ons = {}
59 let events = {}
60 let transitions = []
61
62 findkey = function (array) {
63     return function (key) {
64         var index = _.findIndex(array.name, function (e) {
65             return e == key
66         })
67     }
68 }
```

```

67     return array.value[index]
68   }
69 }
70
71 getEnumerations = function () {
72   var http = new XMLHttpRequest()
73   http.addEventListener("load", function () {
74     var json = JSON.parse(this.responseText)
75     for (var value of json.type.enumValue) {
76       EventEnumeration[parseInt(value.value)] = value.label
77     }
78   })
79   http.open(
80     "GET",
81     "http://localhost:8090/api/mdb/fdirdemo/parameters/fdirdemo/Event_Definition_ID"
82   )
83   http.send()
84
85   var http2 = new XMLHttpRequest()
86   http2.addEventListener("load", function () {
87     var json = JSON.parse(this.responseText)
88     for (var value of _.find(json.type.member, { name: "Check_Status" }).type
89       .enumValue) {
90       CheckStatusEnumeration[parseInt(value.value)] = value.label
91     }
92   })
93   http2.open(
94     "GET",
95     "http://localhost:8090/api/mdb/fdirdemo/parameters/fdirdemo/PMON_Monitoring_Definition"
96   )
97   http2.send()
98 }
99 getEnumerations()
100
101 getST12definitions = _.throttle(
102   function () {
103     pm ons = {}
104     var http = new XMLHttpRequest()
105     http.open(
106       "POST",
107       "http://localhost:8090/api/processors/fdirdemo/realtime/commands/fdirdemo/ST12_ListAllDefinitions"
108     )
109     http.send()
110   },
111   200,
112   { leading: false, trailing: true }
113 )
114
115 getST19definitions = function () {
116   events = {}
117   var http = new XMLHttpRequest()
118   http.open(
119     "POST",
120     "http://localhost:8090/api/processors/fdirdemo/realtime/commands/fdirdemo/ST19_ListAllEventAction"
121   )
122   http.send()
123 }
124
125 getST19requests = function () {
126   for (key in events) {
127     console.log(key)
128     var http = new XMLHttpRequest()
129     http.open(
130       "POST",
131       "http://localhost:8090/api/processors/fdirdemo/realtime/commands/fdirdemo/ST19_ListEventActionRequest"
132     )
133     http.send(
134       JSON.stringify({
135         assignment: [{ name: "Event_Action_Definition_ID", value: key }],
136       })
137     )
138   }
139 }
140
141 clearTransitions = function () {
142   transitions = []

```

```

143     createTransitionTable()
144 }
145
146 colourStatus = function (element) {
147   text = element.innerText.trim()
148   if (text == "Invalid") {
149     element.style.color = "#ff8f00"
150     element.style.fontWeight = 600
151   } else if (text == "Unchecked") {
152     element.style.colour = "#757575"
153     element.style.fontWeight = 500
154   } else if (text != "OK") {
155     element.style.color = "#c62828"
156     element.style.fontWeight = 600
157   }
158   return element
159 }
160
161 createPmonTable = _.throttle(
162   function () {
163     $pmontable.innerHTML = ""
164
165     for (const [pmontId, pmon] of Object.entries(pmmons)) {
166       var tr = document.createElement("tr")
167
168       var tds = _.map(new Array(8), function (e) {
169         return document.createElement("td")
170       })
171
172       tds[0].appendChild(document.createTextNode(pmontId))
173       tds[1].appendChild(document.createTextNode(pmon.parameter))
174       tds[1].classList.add("table-parameter")
175
176       if (pmon.validity) {
177         var lis = _.map(new Array(2), function (e) {
178           return document.createElement("p")
179         })
180         lis[0].appendChild(document.createTextNode(pmon.validity.parameter))
181         lis[1].appendChild(document.createTextNode(pmon.validity.value))
182         lis[1].appendChild(document.createElement("span"))
183         lis[1].childNodes[1].classList.add("mdl-chip")
184         lis[1].childNodes[1].classList.add("mdl-chip-table")
185         lis[1].childNodes[1].appendChild(document.createElement("code"))
186         lis[1].childNodes[1].childNodes[0].classList.add("mdl-chip__text")
187         lis[1].childNodes[1].childNodes[0].appendChild(
188           document.createTextNode(pmon.validity.mask)
189         )
190         tds[2].appendChild(lis[0])
191         tds[2].appendChild(lis[1])
192     }
193
194     tds[3].appendChild(document.createTextNode(pmon.monitoring_interval))
195     tds[4].appendChild(document.createTextNode(pmon.status))
196
197     colourStatus(tds[4])
198
199     tds[5].appendChild(document.createTextNode(pmon.repetition_number))
200
201     if (pmon.check_type == "Limit_Check") {
202       nodes = []
203       nodes[0] = document.createElement("p")
204       nodes[0].appendChild(
205         document.createTextNode(
206           pmont.check.low + " <= " + "x" + " <= " + pmont.check.high
207         )
208       )
209       nodes[1] = document.createElement("span")
210       nodes[1].classList.add("mdl-chip")
211       nodes[1].classList.add("mdl-chip-long")
212       nodes[1].appendChild(document.createElement("span"))
213       nodes[1].childNodes[0].classList.add("mdl-chip__text")
214       nodes[1].childNodes[0].appendChild(
215         document.createTextNode(pmon.check.low_event)
216       )
217       nodes[2] = document.createElement("span")
218       nodes[2].classList.add("mdl-chip")

```

```

219     nodes[2].classList.add("mdl-chip-long")
220     nodes[2].appendChild(document.createElement("span"))
221     nodes[2].childNodes[0].classList.add("mdl-chip__text")
222     nodes[2].childNodes[0].appendChild(
223       document.createTextNode(pmon.check.high_event)
224     )
225
226     tds[6].appendChild(nodes[0])
227     tds[6].appendChild(nodes[1])
228     tds[6].appendChild(nodes[2])
229   } else {
230     nodes = []
231     nodes[0] = document.createElement("p")
232     nodes[0].appendChild(
233       document.createTextNode("x" + " = " + pmon.check.value)
234     )
235     nodes[1] = document.createElement("span")
236     nodes[1].classList.add("mdl-chip")
237     nodes[1].classList.add("mdl-chip-table")
238     nodes[1].appendChild(document.createElement("span"))
239     nodes[1].childNodes[0].classList.add("mdl-chip__text")
240     nodes[1].childNodes[0].appendChild(
241       document.createTextNode(pmon.check.mask)
242     )
243     nodes[0].appendChild(nodes[1])
244     nodes[2] = document.createElement("span")
245     nodes[2].classList.add("mdl-chip")
246     nodes[2].classList.add("mdl-chip-long")
247     nodes[2].appendChild(document.createElement("span"))
248     nodes[2].childNodes[0].classList.add("mdl-chip__text")
249     nodes[2].childNodes[0].appendChild(
250       document.createTextNode(pmon.check.event)
251     )
252
253     tds[6].appendChild(nodes[0])
254     tds[6].appendChild(nodes[2])
255   }
256
257   tds[7].appendChild(document.createTextNode(pmon.date))
258
259   for (const td of Object.values(tds)) {
260     tr.appendChild(td)
261   }
262
263   $pmonTable.appendChild(tr)
264 }
265 },
266 50,
267 { leading: false, trailing: true }
268 )
269
270 createEventActionTable = _.throttle(
271   function () {
272     $eventActionTable.innerHTML = ""
273
274     for (const [eventActionID, eventAction] of Object.entries(events)) {
275       var tr = document.createElement("tr")
276       var tds = _.map(new Array(4), function (e) {
277         return document.createElement("td")
278       })
279
280       tds[0].appendChild(document.createTextNode(eventActionID))
281
282       tds[1].appendChild(document.createElement("span"))
283       tds[1].childNodes[0].classList.add("mdl-chip")
284       // tds[1].childNodes[0].classList.add('mdl-chip-long');
285       tds[1].childNodes[0].appendChild(document.createElement("span"))
286       tds[1].childNodes[0].classList.add("mdl-chip__text")
287       tds[1].childNodes[0].childNodes[0].appendChild(
288         document.createTextNode(eventAction.event)
289       )
290
291       if (eventAction.enabled) {
292         tds[2].appendChild(document.createTextNode("On"))
293         tds[2].classList.add("mdl-color-text--teal-300")
294       } else {

```

```

295     tds[2].appendChild(document.createTextNode("Off"))
296     tds[2].classList.add("mdl-color-text--red-300")
297   }
298
299   if (eventAction.request) {
300     tds[3].appendChild(
301       document.createTextNode(
302         "TC[" +
303           eventAction.request.service +
304           "," +
305           eventAction.request.message +
306           "] "
307         )
308       )
309     tds[3].appendChild(document.createElement("span"))
310     tds[3].childNodes[1].classList.add("mdl-typography--font-light")
311     tds[3].childNodes[1].appendChild(
312       document.createTextNode(eventAction.request.content1 + " "))
313     )
314     tds[3].appendChild(document.createElement("span"))
315     tds[3].childNodes[2].classList.add("mdl-typography--font-light")
316     tds[3].childNodes[2].appendChild(
317       document.createTextNode(eventAction.request.content2)
318     )
319   }
320
321   for (const td of Object.values(tds)) {
322     tr.appendChild(td)
323   }
324   $eventActionTable.appendChild(tr)
325 }
326 },
327 50,
328 { leading: false, trailing: true }
329 )
330
331 createTransitionTable = _.throttle(
332   function () {
333     $transitionTable.innerHTML = ""
334
335     for (const transition of _.reverse([...transitions])) {
336       var tr = document.createElement("tr")
337       var tds = _.map(new Array(5), function (e) {
338         return document.createElement("td")
339       })
340
341       tds[0].appendChild(document.createTextNode(transition.timestamp))
342       tds[1].appendChild(document.createTextNode(transition.pmonId))
343       tds[2].appendChild(document.createTextNode(transition.parameter))
344       tds[3].appendChild(document.createTextNode(transition.value))
345
346       tds[4].appendChild(document.createElement("span"))
347       tds[4].appendChild(document.createElement("i"))
348       tds[4].appendChild(document.createElement("span"))
349       tds[4].childNodes[0].appendChild(
350         document.createTextNode(transition.previous)
351       )
352       colourStatus(tds[4].childNodes[0])
353       tds[4].childNodes[1].appendChild(
354         document.createTextNode("arrow_right_alt")
355       )
356       tds[4].childNodes[1].classList.add("material-icons")
357       tds[4].childNodes[2].appendChild(
358         document.createTextNode(transition.current)
359       )
360       colourStatus(tds[4].childNodes[2])
361
362       for (const td of Object.values(tds)) {
363         tr.appendChild(td)
364       }
365       $transitionTable.appendChild(tr)
366     }
367   },
368   50,
369   { leading: false, trailing: true }
370 )

```

```

371
372 websocket.onmessage = function (event) {
373   var json = JSON.parse(event.data)
374   console.log(json)
375
376   if (json.type == "time") {
377     $timestamp.innerText = json.data.value
378   } else if (json.type == "parameters") {
379     var monitoringRaw = .find(json.data.values, { numericId: 1 })
380     var eventActionRaw = .find(json.data.values, { numericId: 6 })
381     var eventActionRequestRaw = .find(json.data.values, { numericId: 7 })
382     var checkListRaw = .find(json.data.values, { numericId: 8 })
383
384   if (eventActionRaw) {
385     var eventActionList = eventActionRaw.engValue.arrayValue
386
387     for (var definitionRaw of eventActionList) {
388       var eventaction = findkey(definitionRaw.aggregateValue)
389
390       var entry = {
391         event:
392           EventEnumeration[eventaction("Event_Definition_ID").uint32Value],
393         ID: eventaction("EventAction_Definition_ID").uint32Value,
394         enabled: eventaction("Enabled").uint32Value,
395       }
396
397       events[entry["ID"]] = entry
398     }
399
400     createEventActionTable()
401   }
402   if (eventActionRequestRaw) {
403     var eventActionRequest = findkey(
404       eventActionRequestRaw.engValue.aggregateValue
405     )
406     var id = eventActionRequest("EventAction_Definition_ID").uint32Value
407     var text = eventActionRequest("TC").stringValue
408     var request = {
409       text: text,
410       service: text.charCodeAt(1),
411       message: text.charCodeAt(2),
412       content: text.substr(5),
413       content1: text.substr(5, 16),
414       content2: text.substr(5 + 16, 16),
415     }
416     createEventActionTable = .throttle(
417       function () {
418         $eventActionTable.innerHTML = ""
419
420         for (const [eventActionID, eventAction] of Object.entries(events)) {
421           var tr = document.createElement("tr")
422           var tds = .map(new Array(4), function (e) {
423             return document.createElement("td")
424           })
425
426           tds[0].appendChild(document.createTextNode(eventActionID))
427
428           tds[1].appendChild(document.createElement("span"))
429           tds[1].childNodes[0].classList.add("mdl-chip")
430           // tds[1].childNodes[0].classList.add('mdl-chip-long');
431           tds[1].childNodes[0].appendChild(document.createElement("span"))
432           tds[1].childNodes[0].childNodes[0].classList.add("mdl-chip__text")
433           tds[1].childNodes[0].childNodes[0].appendChild(
434             document.createTextNode(eventAction.event)
435           )
436
437           if (eventAction.enabled) {
438             tds[2].appendChild(document.createTextNode("On"))
439             tds[2].classList.add("mdl-color-text--teal-300")
440           } else {
441             tds[2].appendChild(document.createTextNode("Off"))
442             tds[2].classList.add("mdl-color-text--red-300")
443           }
444
445           if (eventAction.request) {
446             tds[3].appendChild(

```

```

447     document.createTextNode(
448         "TC[ " +
449             eventAction.request.service +
450             ", " +
451             eventAction.request.message +
452             "] "
453         )
454     )
455     tds[3].appendChild(document.createElement("span"))
456     tds[3].childNodes[1].classList.add("mdl-typography--font-light")
457     tds[3].childNodes[1].appendChild(
458         document.createTextNode(eventAction.request.content1 + " "))
459     )
460     tds[3].appendChild(document.createElement("span"))
461     tds[3].childNodes[2].classList.add("mdl-typography--font-light")
462     tds[3].childNodes[2].appendChild(
463         document.createTextNode(eventAction.request.content2)
464     )
465   }
466
467   for (const td of Object.values(tds)) {
468     tr.appendChild(td)
469   }
470   $eventActionTable.appendChild(tr)
471 }
472 },
473 50,
474 { leading: false, trailing: true }
475 )
476
477 events[id]["request"] = request
478
479 createEventActionTable()
480 }
481 if (checkListRaw) {
482   var checkList = checkListRaw.engValue.arrayValue
483
484   for (var checkRaw of checkList) {
485     var check = findkey(checkRaw.aggregateValue)
486
487     var transition = {
488       pmonId: check("PMON_ID").uint32Value,
489       parameter: Parameters[check("Monitored_Parameter_ID").uint32Value],
490       check_type: check("Check_Type").uint32Value,
491       value: check("Value").floatValue,
492       previous:
493         CheckStatusEnumeration[check("Previous_Check_Status").uint32Value],
494       current:
495         CheckStatusEnumeration[check("Current_Check_Status").uint32Value],
496       timestamp: check("Timestamp").uint32Value,
497     }
498
499     transitions.push(transition)
500   }
501
502   createTransitionTable()
503   getST12definitions()
504 }
505 if (monitoringRaw) {
506   var checkRaw = _.find(json.data.values, function (e) {
507     return e.numericId != 1
508   })
509
510   monitoring = findkey(monitoringRaw.engValue.aggregateValue)
511   check = findkey(checkRaw.engValue.aggregateValue)
512
513   if (check("Mask") !== undefined) {
514     checkData = {
515       mask: "0x" + check("Mask").uint32Value.toString(16),
516       value: check("Expected_Value").stringValue
517         ? check("Expected_Value").stringValue
518         : check("Expected_Value").uint32Value,
519       event: check("Event_Definition_ID").stringValue,
520     }
521   } else {
522     checkData = {

```

```

523     low: check("Low_Limit").floatValue,
524     low_event: check("Low_Event").stringValue,
525     high: check("High_Limit").floatValue,
526     high_event: check("High_Event").stringValue,
527   }
528 }
529
530 pmoms[monitoring("PMON_ID").uint32Value] = {
531   parameter: Parameters[monitoring("Monitored_Parameter_ID").uint32Value],
532   validity:
533     Parameters[monitoring("Mask").uint32Value] == 0
534     ? null
535     : {
536       parameter:
537         Parameters[monitoring("Validity_Parameter_ID").uint32Value],
538       mask: "0x" + monitoring("Mask").uint32Value.toString(16),
539       value: monitoring("Expected_Value").uint32Value,
540     },
541   monitoring_interval:
542     monitoring("Monitoring_Interval").uint32Value + " ms",
543   status: monitoring("Check_Status").stringValue,
544   repetition_number: monitoring("Repetition_Number").uint32Value,
545   check: checkData,
546   check_type: monitoring("Check_Type").stringValue,
547   date: monitoringRaw.generationTime,
548 }
549
550   createPmonTable()
551 }
552 }
553 }
```

link.cpp (fdir-demo-yamcs)

Το αρχείο αυτό λειτουργεί ως το χαμηλότερο επίπεδο στην επικοινωνία υπολογιστή-μικροελεγκτή. Μεταφέρει πακέτα από το YAMCS προς το UART του μικροελεγκτή και αντίθετα.

```

1 #include <iostream>
2 #include <boost/asio.hpp>
3 #include <boost/array.hpp>
4 #include <boost/bind.hpp>
5 #include <string>
6 #include "cobs.h"
7
8 using namespace std::literals;
9 using boost::asio::ip::udp;
10
11 const auto port = "/dev/ttyACM0"s;
12 const int baudRate = 115200;
13
14 const auto host = "127.0.0.1"s;
15 const int udpOutPort = 10015;
16 const int udpInPort = 10025;
17
18 int main() {
19     try {
20         // Initialise serial connection
21         boost::asio::io_service io;
22         boost::asio::serial_port serial(io, port);
23         serial.set_option(boost::asio::serial_port_base::baud_rate(baudRate));
24
25         boost::asio::streambuf buf;
26         std::istream is(&buf);
27         std::istringstream iss;
28
29         // Initialise UDP connection
30         udp::socket txSocket(io);
31         txSocket.open(udp::v4());
32         udp::endpoint endpoint(boost::asio::ip::make_address_v4(host), udpOutPort);
33
34         udp::endpoint rxEndpoint(boost::asio::ip::make_address_v4(host), udpInPort);
35         udp::socket rxSocket(io, rxEndpoint);
36         boost::array<unsigned char, 1024> rdbuf;
37
38         std::function<void(const boost::system::error_code&, std::size_t)> udpHandler =
39             [&udpHandler, &rxSocket, &rxEndpoint, &rdbuf, &serial]
40             (const boost::system::error_code& error, std::size_t bytes_transferred) {
41                 if (error.failed()) {
42                     std::cerr << error.category().name() << ":" << error.message() << std::endl;
43                     std::this_thread::sleep_for(500ms);
44                 } else {
45                     std::array<unsigned char, 1024> cobs_buffer;
46
47                     if (bytes_transferred > 4) {
48                         std::cout << "Received TC["
49                             << int{rdbuf[1]}
50                             << "," << int{rdbuf[2]}
51                             << "] (" << bytes_transferred
52                             << " bytes)" << std::endl;
53                 }
54
55                 // COBS encoding
56                 auto result = cobs_encode(
57                     cobs_buffer.data(), cobs_buffer.size() - 1, rdbuf.data(), bytes_transferred);
58
59                 if (result.status != COBS_ENCODE_OK) {
60                     std::cerr << "COBS encode status returned "
61                         << static_cast<int>(result.status) << std::endl;
62                 } else {
63                     // Send packet via serial
64                     cobs_buffer[result.out_len] = '\0';
65                     boost::asio::write(serial, boost::asio::buffer(cobs_buffer.data(), result.out_len + 1));
66                 }
67             }
68     }

```

```

69     rxSocket.async_receive_from(boost::asio::buffer(rxbuf), rxEndpoint, udpHandler);
70 };
71
72 rxSocket.async_receive_from(boost::asio::buffer(rxbuf), rxEndpoint, udpHandler);
73
74 std::function<void(const boost::system::error_code&,std::size_t)> serialHandler =
75 [&serial, &serialHandler, &buf, &txSocket, &endpoint]
76 (const boost::system::error_code& error, std::size_t bytes_transferred) -> void {
77
78     if (error.failed()) {
79         std::cerr << error.category().name() << ":" << error.message() << std::endl;
80         std::this_thread::sleep_for(500ms);
81     } else {
82         std::string receivedAll(reinterpret_cast<const char *>(buf.data().data()), buf.size());
83
84         // Find the first occurrence of a zero
85         size_t zeroLocation = receivedAll.find('\0');
86         std::string receivedRaw(reinterpret_cast<const char *>(buf.data().data()), zeroLocation);
87         buf.consume(zeroLocation + 1);
88
89         // COBS decoding
90         std::vector<char> buffer(receivedRaw.size());
91         auto result = cobs_decode(buffer.data(), buffer.size(), receivedRaw.c_str(),
92                                  receivedRaw.size()); // strip the last byte
93
94         if (result.status != COBS_DECODE_OK) {
95             std::cerr << "COBS status returned " << static_cast<int>(result.status) << std::endl;
96         } else {
97             // Send correct packet via UDP
98
99             if (buffer[1] == -1 && buffer[2] == -1) {
100                 std::cout << std::string(buffer.data(), result.out_len).substr(4);
101             } else {
102                 txSocket.send_to(boost::asio::buffer(buffer.data()), result.out_len, endpoint);
103                 std::cout << "\u001b[38;5;243m"
104                     << "Received TM (" << receivedRaw.size() << ")" << "\u001b[0m" << std::endl;
105             }
106         }
107     }
108
109     // Read next data packet
110     boost::asio::async_read_until(serial, buf, '\0', serialHandler);
111 };
112
113 boost::asio::async_read_until(serial, buf, '\0', serialHandler);
114
115 io.run();
116
117
118 } catch (boost::system::system_error &e) {
119     std::cerr << e.what() << std::endl;
120 }
121
122 std::cout << "endl" << std::endl;
123
124 }
```