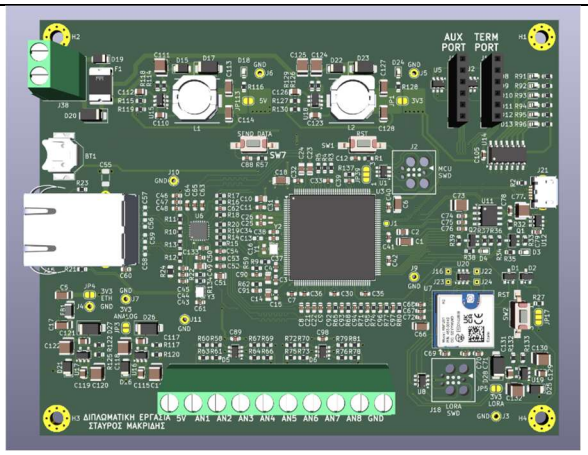
ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Συσκευή κρυπτογράφησης ασύρματης επικοινωνίας μέσω LoRa με χρήση χαοτικών χαρτών»



Του φοιτητή
Σταύρου Μακρίδη
Αρ. Μητρώου: 52111m

Επιβλέπων
Άγγελος Γιακουμής
Αναπληρωτής καθηγητής

Θεσσαλονίκη Μάρτιος 2026

Τίτλος Δ.Ε. Συσκευή κρυπτογράφησης ασύρματης επικοινωνίας μέσω LoRa με χρήση χαοτικών χαρτών

Κωδικός Δ.Ε. 23260

Ονοματεπώνυμο φοιτητή Σταύρος Μακρίδης

Ονοματεπώνυμο εισηγητή Άγγελος Γιακουμής

Ημερομηνία ανάληψης Δ.Ε. 01-10-2023

Ημερομηνία περάτωσης Δ.Ε. 15-2-2026

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Στην οικογένεια μου για την αμέριστη συμπαράσταση της»

Πρόλογος

Η ασφάλεια των ασύρματων επικοινωνιών αποτελεί πλέον βασικό ζήτημα, ιδιαίτερα σε συστήματα χαμηλής κατανάλωσης και μεγάλης εμβέλειας, όπως αυτά που βασίζονται στην τεχνολογία LoRa. Σε πολλές περιπτώσεις, οι συσκευές αυτές καλούνται να λειτουργήσουν σε πραγματικά περιβάλλοντα, με περιορισμένους υπολογιστικούς πόρους και αυξημένες απαιτήσεις αξιοπιστίας, γεγονός που καθιστά τη σχεδίαση μηχανισμών κρυπτογράφησης μια σύνθετη πρόκληση.

Το ενδιαφέρον για το συγκεκριμένο θέμα προέκυψε μέσα από την ενασχόληση με ενσωματωμένα συστήματα και ασύρματες επικοινωνίες, όπου η πρακτική υλοποίηση συχνά αποκαλύπτει περιορισμούς που δεν είναι εμφανείς σε καθαρά θεωρητικό επίπεδο. Στο πλαίσιο αυτό, η χαοτική θεωρία και οι χαοτικοί χάρτες παρουσιάζουν ιδιαίτερο ενδιαφέρον, καθώς προσφέρουν ιδιότητες που μπορούν να αξιοποιηθούν για την παραγωγή ψευδοτυχαίων ακολουθιών και την ανάπτυξη εναλλακτικών κρυπτογραφικών σχημάτων.

Η παρούσα διπλωματική εργασία επικεντρώνεται στη σχεδίαση και υλοποίηση μιας συσκευής κρυπτογράφησης ασύρματης επικοινωνίας μέσω LoRa με χρήση χαοτικών χαρτών, με στόχο τη διερεύνηση της πρακτικής εφαρμογής αυτής της προσέγγισης σε πραγματικά ενσωματωμένα συστήματα.

Σημειώνεται ότι η προτεινόμενη προσέγγιση δεν στοχεύει στην αντικατάσταση καθιερωμένων προτύπων κρυπτογραφίας υψηλής ασφάλειας ούτε αποτελεί πιστοποιημένη υλοποίηση κατά NIST. Αντιθέτως, διερευνάται ως ελαφρύς μηχανισμός απόκρυψης δεδομένων κατάλληλος για περιβάλλοντα περιορισμένων πόρων, όπου προέχει η πρακτική λειτουργικότητα και η χαμηλή υπολογιστική επιβάρυνση.

Περίληψη

Η παρούσα διπλωματική εργασία αφορά τη σχεδίαση και υλοποίηση μιας ενσωματωμένης συσκευής κρυπτογράφησης ασύρματης επικοινωνίας μέσω LoRa, με χρήση χαοτικών χαρτών. Η τεχνολογία LoRa χρησιμοποιείται ευρέως σε εφαρμογές χαμηλής κατανάλωσης και μεγάλης εμβέλειας, όπου η ασφάλεια των δεδομένων και η αποδοτική αξιοποίηση των διαθέσιμων πόρων αποτελούν κρίσιμες παραμέτρους.

Στο πλαίσιο της εργασίας μελετώνται βασικές έννοιες της χαοτικής θεωρίας και εξετάζεται η δυνατότητα αξιοποίησης χαοτικών χαρτών για την ανάπτυξη μηχανισμών κρυπτογράφησης κατάλληλων για ενσωματωμένα συστήματα. Προτείνεται και υλοποιείται ένα σύστημα κρυπτογράφησης το οποίο ενσωματώνεται σε συσκευή ασύρματης επικοινωνίας LoRa, λαμβάνοντας υπόψη περιορισμούς υπολογιστικής ισχύος, μνήμης και κατανάλωσης ενέργειας.

Η αναπτυγμένη συσκευή υποστηρίζει, πέραν της ασύρματης επικοινωνίας LoRa, ενσύρματες διεπαφές USB και Ethernet 10/100, καθώς και οκτώ αναλογικά κανάλια εισόδου ανάλυσης 12 bit. Επιπλέον, περιλαμβάνει σειριακή διεπαφή UART, η οποία αξιοποιείται τόσο για καταγραφή λειτουργικών δεδομένων όσο και για την υλοποίηση περιβάλλοντος διεπαφής χρήστη μέσω κονσόλας.

Η υλοποίηση καλύπτει τόσο το επίπεδο υλικού όσο και το επίπεδο λογισμικού, περιλαμβάνοντας τη διαδικασία κρυπτογράφησης και αποκρυπτογράφησης των δεδομένων πριν και μετά τη μετάδοση. Πραγματοποιούνται πειραματικές δοκιμές για την αξιολόγηση της λειτουργίας του συστήματος και της επίδρασης της κρυπτογράφησης στην επικοινωνία, με τα αποτελέσματα να καταδεικνύουν τη δυνατότητα εφαρμογής χαοτικών χαρτών σε σύγχρονα ενσωματωμένα συστήματα ασύρματης επικοινωνίας.

Η αξιολόγηση της μεθόδου εστιάζει στη λειτουργική επάρκεια και την πρακτική εφαρμοσιμότητα σε συστήματα χαμηλών πόρων και δεν συνιστά πιστοποίηση κρυπτογραφικής ισχύος επιπέδου προτύπων (π.χ. NIST).

«An Embedded Chaotic Encryption System for Secure LoRa Wireless Communication»

«Stavros Makridis»

Abstract

This diploma thesis focuses on the design and implementation of an embedded device for secure wireless communication over LoRa, utilizing chaotic maps. LoRa technology is widely adopted in low-power, long-range applications, where data security and efficient use of limited system resources constitute critical requirements.

Within the scope of this work, fundamental concepts of chaos theory are examined, and the potential use of chaotic maps for the development of cryptographic mechanisms suitable for embedded systems is investigated. A cryptographic scheme based on chaotic behavior is proposed and implemented, taking into account constraints related to computational power, memory availability, and energy consumption. The proposed scheme is integrated into a LoRa-based wireless communication device.

In addition to LoRa communication, the developed device supports multiple wired interfaces, including USB and Ethernet 10/100, as well as eight 12-bit analog input channels. Furthermore, a UART interface is implemented and used both for system logging and for providing a simple console-based user interface. These features enable comprehensive system interaction, configuration, and monitoring during operation and testing.

The implementation covers both hardware and software levels, including the processes of data encryption prior to transmission and decryption upon reception. Experimental evaluations are conducted to verify the correct operation of the system and to assess the impact of the cryptographic process on LoRa communication performance. The results demonstrate that chaotic-map-based cryptographic approaches can be effectively applied to modern low-resource embedded wireless communication systems, offering a practical alternative for enhancing data security.

The evaluation of the proposed method focuses on functional adequacy and practical applicability in resource-constrained systems and does not constitute a certification of cryptographic strength according to established standards (e.g., NIST). The approach is examined as a lightweight data-obfuscation mechanism rather than a replacement for conventional high-security cryptographic algorithms.

Ευχαριστίες

Η καθοδήγηση και η υποστήριξη του καθηγητή κ. Άγγελου Γιακουμή, ο οποίος είχε την καλοσύνη να είναι ο επιβλέπων της παρούσας εργασίας, αποτέλεσαν καθοριστικό παράγοντα για την επιτυχή ολοκλήρωσή της. Τον ευχαριστώ θερμά, όχι μόνο για τη συνεχή υποστήριξη και τις πολύτιμες υποδείξεις του, αλλά και για τις γνώσεις και την εμπειρία που αποκόμισα κατά τη διάρκεια της εκπόνησής της.

Μια διπλωματική εργασία δεν είναι αποκομμένη από την εκπαιδευτική διαδικασία. Το αντίθετο μάλιστα, αποτελεί το επιστέγασμα μιας προσπάθειας μάθησης και μια δύσκολης πορείας προς την γνώση. Σε αυτή την πορεία η βοήθεια όλων των καθηγητών του τμήματος που συμμετείχαν στο πρόγραμμα ήταν καταλυτική και πολύτιμή. Τους ευχαριστώ όλους από βάθος καρδιάς.

Ιδιαίτερες ευχαριστίες οφείλω στην οικογένειά μου, και ειδικότερα στη σύζυγο και την κόρη μου, για την ψυχολογική υποστήριξη, την κατανόηση και τη συμπαράσταση που μου προσέφεραν απλόχερα καθ' όλη τη διάρκεια αυτής της προσπάθειας.

Περιεχόμενα

Πρόλογος iv
Περίληψη v
Abstract vi
Ευχαριστίες vii
Περιεχόμενα viii
Κατάλογος Σχημάτων xi
Κατάλογος Πινάκων xii
Συντομογραφίες xiii
Κεφάλαιο 1ο: Το σύστημα 14
1.1 Το δομικό στοιχείο 14
1.2 Τοπολογία συστήματος 14
1.3 Επεκτάσεις - Δυνατότητες 15
Κεφάλαιο 2ο: Σχεδίαση Υλικού 15
2.1 Διάγραμμα βαθμίδων 15
2.2 Μικροϋπολογιστής 15
2.2.1 Ethernet 17
2.2.2 LoRa 18
2.2.3 USB 19
2.2.4 Αναλογικές είσοδοι 20
2.3 Τμήμα Ethernet 20
2.4 Τμήμα LoRa 21
2.5 Τμήμα USB 23
2.6 Τμήμα αναλογικών εισόδων 24
2.7 Τροφοδοσία 24
2.7.1 Παλμοσταθεροποιητές υποβιβασμού τάσης από 12V σε 5V και 3.3V 26
2.7.2 Γραμμικοί σταθεροποιητές από 5V σε 3,3V 27
2.8 Σχεδίαση τυπωμένου κυκλώματος 28
2.9 Καθορισμός διάταξης στρωμάτων 28
2.10 Σχεδιαστικοί κανόνες 30
Κεφάλαιο 3ο: Θεωρία Χάους και Ασφάλεια Δεδομένων 31
3.1 Ντετερμινιστικότητα και απροσδιοριστία 31
3.2 Ευαισθησία στις αρχικές συνθήκες 31

3.3 Μη γραμμικότητα και πολύπλοκες τροχιές 32
3.4 Σύνδεση με Κρυπτογράφηση 32
3.4.1 Διαδοχική επανάληψη (iterative amplification) 33
3.4.2 Διασύνδεση (cross-coupling) μεταξύ διαφορετικών χαρτών 33
3.4.3 Μεταφορά της χαοτικής κατάστασης σε δυαδική μορφή μέσω μηχανισμών μίξης 33
3.5 Πλαίσιο εφαρμογής στη μετάδοση αναλογικών μετρήσεων 33
Κεφάλαιο 4ο: Χαοτικοί Χάρτες και Ψηφιακή Υλοποίηση 34
4.1 Η έννοια του διακριτού χάρτη 34
4.2 Logistic Map 35
4.2.1 Πειραματική απεικόνιση σε ψηφιακή υλοποίηση 36
4.2.2 Ερμηνεία πειραματικών αποτελεσμάτων 36
4.3 Ο Tent Map 36
4.3.1 Πειραματική απεικόνιση σε ψηφιακή υλοποίηση 37
4.3.2 Ερμηνεία πειραματικών αποτελεσμάτων 37
4.4 Ο Chebyshev Map και άλλες προσεγγίσεις 38
4.4.1 Σχόλια ψηφιακής υλοποίησης 38
4.5 Υβριδικός Logistic–Tent Map 38
4.5.1 Πειραματική απεικόνιση σε ψηφιακή υλοποίηση 39
4.5.2 Ερμηνεία πειραματικών αποτελεσμάτων 40
4.6 Ψηφιακά ζητήματα: από την αναλογική συνέχεια στη διακριτή αναπαράσταση 40
4.7 Κριτήρια Σύγκρισης Χαοτικών Χαρτών για Embedded PRNG 40
4.8 Συγκριτική Αποτίμηση Χαοτικών Χαρτών 41
4.9 Έλεγχοι – Συμπεράσματα 42
4.10 Τεκμηριωμένη Επιλογή Τελικού Αλγορίθμου 43
Κεφάλαιο 5ο: Παραγωγή Ψευδοτυχαίων Ακολουθιών και Χαοτική Κρυπτογράφηση 44
5.1 Χαοτικά συστήματα ως PRNG 44
5.2 Μετατροπή της χαοτικής κατάστασης σε 32-bit τιμές 45
5.3 Διαδικασία παραγωγής keystream 45
5.4 Stream cipher μέσω XOR 46
5.5 Συμβατότητα με συστήματα πραγματικού χρόνου 47
5.6 Σύνοψη κεφαλαίου 47
Κεφάλαιο 6ο: Τεχνικά Χαρακτηριστικά LoRa 48
6.1 Τι είναι το LoRa 48
6.2 Αρχή λειτουργίας – Chirp Spread Spectrum (CSS) 48
6.3 Διαμόρφωση LoRa – Τεχνικές Παράμετροι 48

6.4 Συχνότητες λειτουργίας (ISM Bands) 48
6.5 Ισχύς εκπομπής 48
6.6 Εμβέλεια 49
6.7 LoRa vs LoRaWAN 49
6.8 Πλεονεκτήματα LoRa 49
6.9 Μειονεκτήματα LoRa 49
Κεφάλαιο 7ο: Λογισμικό (Firmware) 50
7.1 Ρύθμιση περιφερειακών και οδηγοί με την χρήση STM32CubeMX 50
7.1.1 LWIP 50
7.1.2 USB Device 50
7.2 Διάγραμμα ροής λειτουργίας 53
7.2.1 Initialization (CubeMX) 55
7.2.2 Initialization (Application) 55
7.2.3 MX_LWIP_Process 55
7.2.4 tcp_server_task 55
7.2.5 udp_server_task 55
7.2.6 usb_process 55
7.2.7 log_process 55
7.2.8 APP_TMR_Process 56
7.2.9 RM126x_Process 56
7.2.10 ADC_Process 56
7.2.11 LoRaP2P_Process 56
7.3 Crypto 56
7.3.1 Αρχικοποίηση (Seeding) 56
7.3.2 Χαοτικό Σύστημα (Hybrid Chaotic Map) 56
7.3.3 Μετατροπή σε Ψευδοτυχαία Λέξη (PRNG Output) 57
7.3.4 Avalanche Hash (Τελική Διασπορά Bit) 57
7.3.5 Παραγωγή Keystream και Κρυπτογράφηση 58
7.3.6 Επεξηγήσεις Σχεδιαστικών Επιλογών 58
7.4 Χειρισμός RM1261 σε P2P mode 58
7.4.1 Αρχικοποίηση για σύνδεση στο δίκτυο P2P 58
7.4.2 Επανασύνδεση 59
7.4.3 Αποστολή και λήψη δεδομένων 60
7.5 Πρωτόκολλο επικοινωνίας 60
7.5.1 Κανάλι LoRa 60

7.5.2 Κανάλι TCP/UDP 60
7.5.3 Κανάλι USB 61
Κεφάλαιο 8ο: Πειραματικός έλεγχος 62
8.1 Διάταξη ανάπτυξης και δοκιμών 62
8.2 Επιβεβαίωση ορθής λειτουργίας 63
ΕΠΙΛΟΓΟΣ: Αποτίμηση – Επεκτάσεις – Συμπεράσματα 66
ΒΙΒΛΙΟΓΡΑΦΙΑ 67
ΠΑΡΑΡΤΗΜΑ Α: Σχηματικά διαγράμματα 73
ΠΑΡΑΡΤΗΜΑ Β: Σχεδίαση παλμοσταθεροποιητών 83
ΠΑΡΑΡΤΗΜΑ Γ: Αρχεία παραγωγής τυπωμένου κυκλώματος 85
ΠΑΡΑΡΤΗΜΑ Δ: Προγράμματα δημιουργίας γραφημάτων 89
ΠΑΡΑΡΤΗΜΑ Ε: Πρόγραμμα καταγραφής επικοινωνίας 96

Κατάλογος Σχημάτων

Σχήμα 1 Τοπολογία συστήματος 14
Σχήμα 2 Διάγραμμα βαθμίδων 15
Σχήμα 3 Συνδεσμολογία SMPS για λειτουργία LDO (SMPS απενεργοποιημένο) 16
Σχήμα 4 Τοπολογία τροφοδοσίας MCU 17
Σχήμα 5 Σύνδεση MCU με τετραγωνικό ταλαντωτή. 17
Σχήμα 6 Εσωτερικό κύκλωμα συνδετήρα Ethernet 21
Σχήμα 7 Δομικό διάγραμμα RM1261 22
Σχήμα 8 Αναλογική είσοδος 24
Σχήμα 9 Διάγραμμα βαθμίδων τμήματος τροφοδοσίας 25
Σχήμα 10 Κύκλωμα προστασίας εισόδου τροφοδοσίας 26
Σχήμα 11 Παλμοσταθεροποιητές υποβιβασμού τάσης από 12V σε 3,3V και 5V 27
Σχήμα 12 Γραμμικός σταθεροποιητές υποβιβασμού τάσης από 5V σε 3,3V 27
Σχήμα 13 Γεωμετρία στρωμάτων τυπωμένου κυκλώματος 29
Σχήμα 14 Ευαισθησία του «Logistic Map» στις αρχικές τιμές 32
Σχήμα 15 Διάγραμμα ροής δεδομένων 34
Σχήμα 16 Ιστόγραμμα τιμών κατάστασης του logistic map σε αριθμητική FP32 και FP64 36
Σχήμα 17 Ιστόγραμμα τιμών κατάστασης του tent map σε αριθμητική FP32 37
Σχήμα 18 Ιστόγραμμα τιμών κατάστασης του υβριδικού logistic–tent map σε FP32 και FP64 39
Σχήμα 19 Κατανομή συχνοτήτων byte εξόδου από υβριδικό PRNG σε FP32 40
Σχήμα 20 Εξέλιξη της μεταβλητής κατάστασης του logistic map σε διαδοχικές επαναλήψεις 44
Σχήμα 21 Εξέλιξη της μεταβλητής κατάστασης του tent map σε διαδοχικές επαναλήψεις 45
Σχήμα 22 Ενδεικτική διαδικασία μετατροπής χαοτικής κατάστασης σε δυαδική αναπαράσταση 45
Σχήμα 23 Ενδεικτική ροή παραγωγής λέξεων από χαοτικό PRNG με ενδιάμεσα βήματα επανάληψης (skipping) μεταξύ διαδοχικών εξόδων 46
Σχήμα 24 Ενδεικτική ροή κρυπτογράφησης stream cipher μέσω πράξης XOR 47

Σχήμα 25 Αντιστοίχιση των σημάτων των περιφερειακών στους ακροδέκτες εισόδου/εξόδου (I/O).. 51
Σχήμα 26 Διάγραμμα ροής Firmware 54
Σχήμα 27 Δομή κρυπτογραφημένου πακέτου δεδομένων 60
Σχήμα 28 Σχηματικό διάγραμμα τμήματος μικροελεγκτή 73
Σχήμα 29 Σχηματικό διάγραμμα τμήματος Ethernet 74
Σχήμα 30 Σχηματικό διάγραμμα τμήματος LoRa 75
Σχήμα 31 Σχηματικό διάγραμμα τμήματος USB 76
Σχήμα 32 Σχηματικό διάγραμμα τμήματος αναλογικών εισόδων..... 77
Σχήμα 33 Σχηματικό διάγραμμα τμήματος τροφοδοσίας 78
Σχήμα 34 Σχηματικό διάγραμμα παλμοσταθεροποιητή 12V/5V (TI's Webench Power Designer).... 83
Σχήμα 35 Σχηματικό διάγραμμα παλμοσταθεροποιητή 12V/3,3V (TI's Webench Power Designer). 83
Σχήμα 36 Silkscreen Top 85
Σχήμα 37 Top Copper (L1) 85
Σχήμα 38 Inner Copper 1 (L2) 86
Σχήμα 39 Inner Copper 2 (L3) 86
Σχήμα 40 Bottom Copper (L4)..... 87
Σχήμα 41 Top Paste Mask..... 87
Σχήμα 42 Top Solder Mask..... 88
Σχήμα 43 Bottom Solder Mask 88

Κατάλογος Πινάκων

Πίνακας 1 Σήματα διασύνδεσης MCU με το τμήμα Ethernet..... 18
Πίνακας 2 Σήματα διασύνδεσης MCU με το τμήμα LoRa 18
Πίνακας 3 Σήματα διασύνδεσης MCU με το τμήμα USB 19
Πίνακας 4 Αναλογικά σήματα..... 20
Πίνακας 5 Διάταξη στρωμάτων τυπωμένου κυκλώματος..... 29
Πίνακας 6 Κανόνες κλάσεων συνδέσεων (Netclass Rules) 30
Πίνακας 7 Συνοπτική ποιοτική αξιολόγηση των logistic, tent και συζευγμένου logistic–tent χαρτών ως προς βασικά tests του NIST SP800-22..... 42
Πίνακας 8 Συχνότητες λειτουργίας LoRa ανά περιοχή..... 48
Πίνακας 9 Αντιστοίχιση των σημάτων των περιφερειακών στους ακροδέκτες εισόδου/εξόδου (I/O). 51
Πίνακας 10 Αλληλουχία εντολών AT αρχικοποίησης..... 59
Πίνακας 11 Αλληλουχία εντολών AT επανασύνδεσης..... 59
Πίνακας 12 Υποστηριζόμενες εντολές πρωτοκόλλου μεταφοράς μετρήσεων..... 61
Πίνακας 13 Λίστα υλικών παλμοσταθεροποιητή 12V/5V..... 83
Πίνακας 14 Λίστα υλικών παλμοσταθεροποιητή 12V/3,3V..... 83

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
LoRa	Long Range
USB	Universal Serial Bus
UART	Universal Asynchronous Receiver/Transmitter
OSI	Open Systems Interconnection
PHY	Physical. Αναφέρεται στο κατώτερο επίπεδο του OSI.
CSS	Chirp Spread Spectrum
VCO	Voltage Controlled Oscillator
PLL	Phase Locked Loop. Τοπολογία που σταθεροποιεί ένα VCO
SMPS	Switching Mode Power Supply
RMII	Reduced Media-Independent Interface
PCB	Printed Circuit Board

Κεφάλαιο 1ο: Το σύστημα

1.1 Το δομικό στοιχείο

Η συσκευή που πραγματεύεται η εργασία αυτή διαθέτει τρείς (3) επικοινωνιακές θύρες, Ethernet [1], USB[2] και LoRa[3] (Κεφάλαιο 5ο). Αυτό την καθιστά ιδιαίτερα ευέλικτη και ικανή να αποτελέσει το δομικό στοιχείο ενός μεγαλύτερου συστήματος αποτελούμενο από έναν κεντρικό κόμβο και πλήθος περιφερειακών κόμβων δικτυωμένων μεταξύ τους. Κάθε περιφερειακός κόμβος ψηφιοποιεί και επεξεργάζεται αναλογικά δεδομένα από μέχρι οκτώ (8) πηγές. Η ανάλυση της αναλογοψηφιακής μετατροπής είναι 12bit για όλα τα κανάλια. Τα δεδομένα στην συνέχεια μεταδίδονται κρυπτογραφημένα μέσω LoRa με σταθερό ρυθμό.

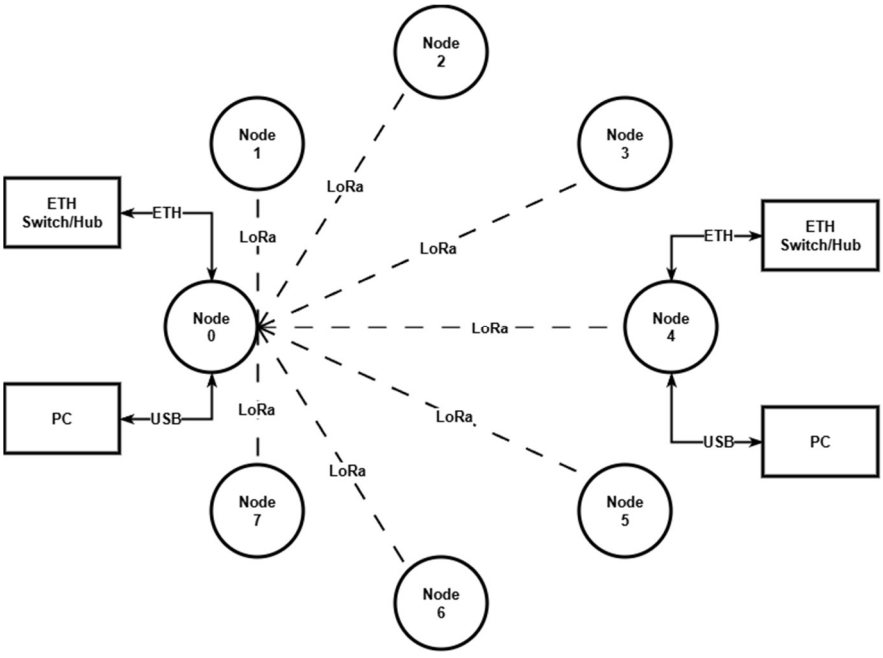
Για τις άλλες δύο θύρες υπάρχει η δυνατότητα τα δεδομένα να αποστέλλονται με έναν από τους παρακάτω τρόπους και η κρυπτογράφηση μπορεί να απενεργοποιηθεί κατά επιλογή και για τους δύο.

1. Μετά από αίτημα σε μια τοπολογία Client/Server με την συσκευή στον ρόλο του Server
2. Με σταθερό ρυθμό οριζόμενο από τον αποδέκτη των δεδομένων

1.2 Τοπολογία συστήματος

Το RM1261[4] παρέχει ένα Peer-to-Peer (P2P)[5] τρόπο λειτουργίας, υλοποιημένο από την eZurio, το οποίο επιτρέπει απλή σημείο-προς-σημείο επικοινωνία μέσω AT-based πρωτοκόλλου, χωρίς να απαιτείται ανάπτυξη firmware με το Silicon Labs SDK.

Το P2P επιτρέπει σε κάθε κόμβο του δικτύου να στέλνει δεδομένα σε οποιοδήποτε κόμβο εντός του ίδιου δικτύου, έναν σε κάθε φορά ή ταυτόχρονα σε όλους (Broadcasting). Κάθε κόμβος έχει μια μοναδική διεύθυνση μέσα στο δίκτυο που είναι συνδεμένος και κάθε δίκτυο έχει ένα μοναδικό αναγνωριστικό. Όλοι οι κόμβοι πρέπει να το γνωρίζουν για να μπορούν να συνδεθούν σε αυτό. Η συνολική τοπολογία του δικτύου απεικονίζεται στο σχήμα που ακολουθεί.



Σχήμα 1 Τοπολογία συστήματος

Ο κόμβος με διεύθυνση 0 στέλνει διαρκώς ένα πακέτο συγχρονισμού (Beacon). Όταν μια συσκευή είναι σε κατάσταση λήψης (listening) και γνωρίζει το αναγνωριστικό του δικτύου, τότε μόλις το λάβει συγχρονίζεται με αυτό αποτελώντας πλέον έναν κόμβο του δικτύου, μέχρις ούτε θελήσει να αποσυνδεθεί από αυτό.

Για τις ανάγκες τις εργασίας έχει υλοποιηθεί η απλούστερη περίπτωση με δύο κόμβους. Ο ένας στο Node-Id = 0 ή Node0 και η δεύτερη στο Node-Id = 1 ή Node1. Στο Node0 έχει συνδεθεί μέσω USB και Ethernet ένας υπολογιστής που εκτελεί λογισμικό και λαμβάνει τα δεδομένα και τα απεικονίζει στην οθόνη του.

1.3 Επεκτάσεις - Δυνατότητες

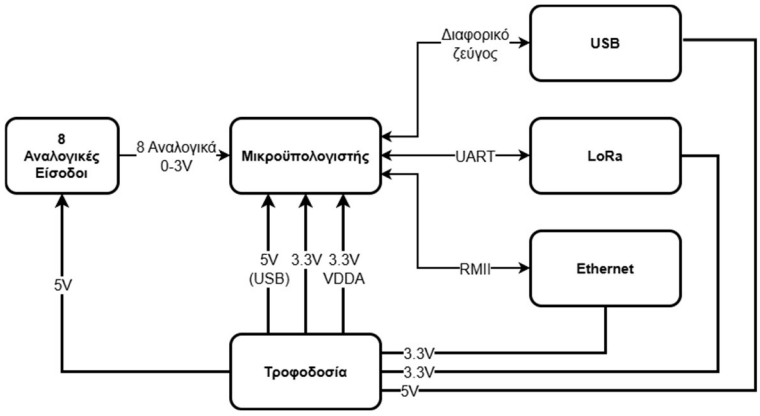
Κάθε κόμβος μπορεί να συνδεθεί μέσω Ethernet[1] σε μεγαλύτερα δίκτυα υπολογιστών και μέσω USB[2] σε άλλες συσκευές εκτός του δικτύου LoRa[3] και να αποτελέσει μια πύλη (gateway) για την μεταφορά των δεδομένων ολόκληρου του ασύρματου δικτύου προς εξωτερικού αποδέκτες. Σχηματίζεται έτσι ένα πλέγμα (grid) με πολλαπλούς δρόμους διασύνδεσης του ασύρματου δικτύου με μεγαλύτερα δίκτυα ακόμα και το Internet.

Ένα σημαντικό πλεονέκτημα αυτής της τοπολογίας είναι ότι το δίκτυο δεν θα βγει εκτός λειτουργίας αν για οποιοδήποτε λόγο ένας ή περισσότεροι κόμβοι τεθούν εκτός λειτουργίας. Συνδυάζοντας την ευελιξία αυτού του δικτύου με την από σχεδιασμού μεγάλη εμβέλεια του LoRa γίνεται εύκολα αντιληπτό ότι μπορεί να εξυπηρετήσει πλήθος εφαρμογών τηλεμετρίας και τηλεχειρισμού.

Κεφάλαιο 2ο: Σχεδίαση Υλικού

2.1 Διάγραμμα βαθμίδων

Στο παρακάτω σχήμα απεικονίζεται σε μορφή διαγράμματος βαθμίδων η διασύνδεση των τμημάτων της συσκευής. Στις ενότητες που ακολουθούν αναλύονται τα επιμέρους τμήματα και η λειτουργία που επιτελούν.



Σχήμα 2 Διάγραμμα βαθμίδων

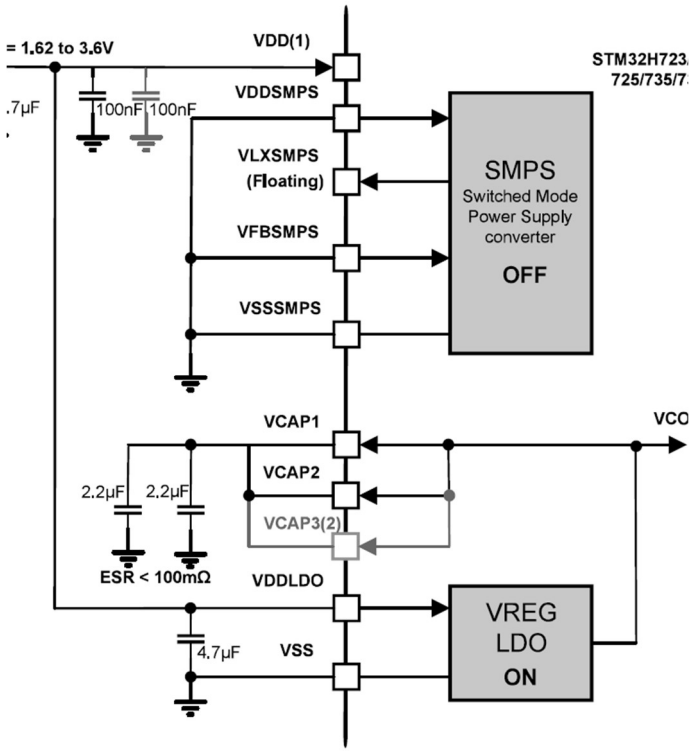
2.2 Μικροϋπολογιστής

Στο κέντρο του βρίσκεται ο μικροελεγκτής (MCU) STM32H735ZG που διαθέτει 1MB μνήμη Flash για την αποθήκευση του εκτελέσιμου κώδικα, 564KB μνήμης RAM για τα δεδομένα του προγράμματος

και μπορεί να χρονιστεί μέχρι 550MHz. Η πλήρης λίστα των περιφερειακών και των δυνατοτήτων περιέχεται στο έγγραφο κατασκευαστικών δεδομένων που παρέχει η ST Microelectronics [6].

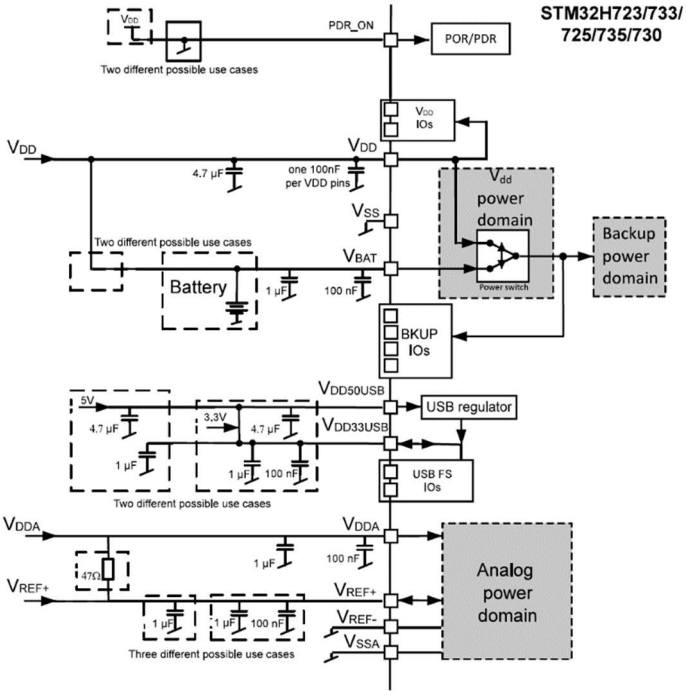
Η οικογένεια που ανήκει ο MCU διαθέτει τα κυκλώματα ελέγχου και ισχύος ενός παλμοτροφοδοτικό (SMPS)[6]. Συνδέοντας εξωτερικά το πηνίο και τον πυκνωτή εξόδου μπορεί να υλοποιηθεί ένα πλήρες και SMPS υποβιβασμού συνεχούς τάσης (Buck). Είναι μια λειτουργία που δίνει λύσεις στον σχεδιαστή ειδικά σε εφαρμογές που το κόστος είναι κρίσιμη σχεδιαστική παράμετρος. Επιλέχθηκε να μην χρησιμοποιηθεί αυτή η δυνατότητα για να αποφευχθεί πιθανή ηλεκτρομαγνητική παρεμβολή (EMI) εκ των έσω στα ευαίσθητα και υψηλής συχνότητας σήματα του RMII[7] και του USB[2] καθώς και τον αναλογικών σημάτων.

Για να απενεργοποιηθεί το SMPS οι ακροδέκτες του συνδέθηκαν σύμφωνα με το έγγραφο χρήσης του STM32H735 της STMicroelectronics [8] και απόσπασμα παραθέτετε στο παρακάτω σχήμα.



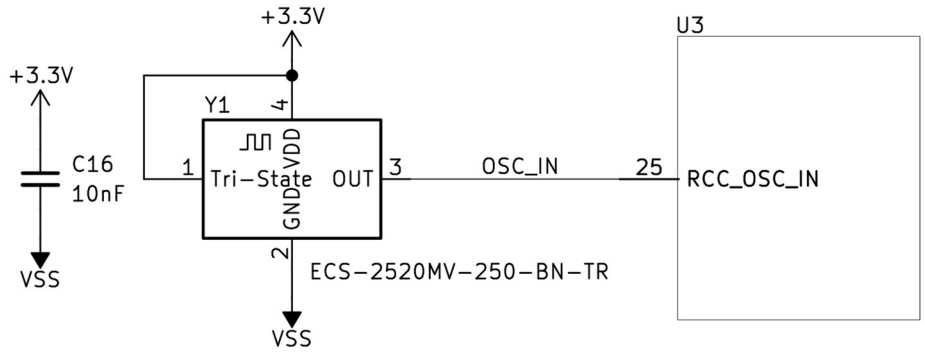
Σχήμα 3 Συνδεσμολογία SMPS για λειτουργία LDO (SMPS απενεργοποιημένο)

Στο παρακάτω απόσπασμα από το ίδιο αρχείο απεικονίζονται οι συνδέσεις του MCU στις γραμμές τροφοδοσίας και οι προτεινόμενες τιμές πυκνωτών απόζευξης. Εκτός από τις τιμές των πυκνωτών προτείνει και τύπο (π.χ. κεραμικός, τανταλίου κ.λπ.).



Σχήμα 4 Τοπολογία τροφοδοσίας MCU

Τον χρονισμό του MCU έχει αναλάβει ο ολοκληρωμένος τετραγωνικός ταλαντωτής ECS-2520MV-250-BN-TR [9] και το τετραγωνικό σήμα που παράγει στην έξοδο του έχει συχνότητα 25MHz. Το εσωτερικό υποσύστημα χρονισμού του MCU διαθέτει ένα σύνολο από διαιρέτες και πολλαπλασιαστές συχνότητας που μαζί με τα τρία PLLs μπορούν να παράγουν την συχνότητα των 550MHz για τον πυρήνα και έναν αριθμό μικρότερων συχνοτήτων για τα επιμέρους περιφερειακά. Η παραμετροποίηση του υποσυστήματος αυτού αναλύεται εκτενώς στο κεφάλαιο του λογισμικού. Το ακόλουθο σχήμα απεικονίζει τον τρόπο σύνδεσης του ταλαντωτή με τον MCU. [6][8][10]



Σχήμα 5 Σύνδεση MCU με τετραγωνικό ταλαντωτή.

2.2.1 Ethernet

Το περιφερειακό MAC εσωτερικά στον MCU είναι ένα από τα πιο σύνθετα υποσυστήματα των περισσοτέρων MCU σήμερα. Περιέχει δεκάδες καταχωρητές και αποκλειστική μνήμη RAM για την διαχείριση των δεδομένων που μεταφέρονται μέσω του Ethernet. Παρόλα αυτά δεν μπορεί να οδηγήσει απ' ευθείας μια σύνδεση Ethernet. Αυτό δεν είναι αδυναμία του STM32H735 αλλά η συνήθεις πρακτική που ακολουθείται από όλους τους κατασκευαστές μικροελεγκτών 32-bit και γίνεται κυρίως για να περιοριστεί η κατανάλωση ισχύος από τον μικροελεγκτή και εξοικονομηθεί το πολύτιμο πυρίτιο στο εσωτερικό του για πιο απαραίτητα κυκλώματα. Μια κατηγορία ολοκληρωμένων κυκλωμάτων που

καλούνται PHY (PHYsical) χρησιμοποιούνται ώστε να καλυφθεί αυτό το κενό. Στην παρούσα σχεδίαση έχει επιλεγεί το LAN8742A [11] της Microchip. Το ολοκληρωμένο καθώς και το τμήμα θα αναλυθούν στην αντίστοιχη ενότητα.

Τα σήματα που χρησιμοποιούνται για την επικοινωνία MCU και PHY που ονομάζεται RMII [7]. Ο πίνακας που ακολουθεί περιέχει τα σχετικά σήματα.

Πίνακας 1 Σήματα διασύνδεσης MCU με το τμήμα Ethernet

Α/Α	Ονομασία	Περιγραφή	Κατεύθυνση
1	ETH_TDX0	Bit0 των δεδομένων που αποστέλλονται προς το PHY	Έξοδος
2	ETH_TDX1	Bit1 των δεδομένων που αποστέλλονται προς το PHY	Έξοδος
3	ETH_TXEN	Εντολή προς το PHY να διαβάσει τα δεδομένα των ETH_TXD1, ETH_TXD0	Έξοδος
4	ETH_RXD0	Bit0 των δεδομένων που αποστέλλονται από το PHY	Είσοδος
5	ETH_RXD1	Bit1 των δεδομένων που αποστέλλονται προς το PHY	Είσοδος
6	ETH_RXER	Το PHY ενημερώνει τον MCU ότι κατά την τελευταία λήψη δεδομένων από το Ethernet υπάρχει σφάλμα	Είσοδος
7	ETH_CRS_DV	Το PHY ενημερώνει τον MCU ότι μπορεί να διαβάσει τα δεδομένα των ETH_RXD1, ETH_RXD0	Είσοδος
8	ETH_MDIO	Δεδομένα της θύρας ελέγχου του PHY	Αμφίδρομο
9	ETH_MDC	Χρονισμός της θύρας ελέγχου του PHY	Έξοδος
10	ETH_nRST	Σήμα reset του PHY	Έξοδος
11	ETH_INT	Το PHY ενημερώνει τον MCU για συμβάντα που συμβαίνουν ασύγχρονα	Είσοδος
12	ETH_REF_CLK	Σήμα συγχρονισμού του MAC με το PHY	Είσοδος

Τα σήματα 1 έως 9 αποτελούν μέρος του διαύλου RMII ενώ τα 10 έως 12 χρήσιμα και συνήθως υπάρχουν σε αντίστοιχες σχεδιάσεις. Τα σήματα ETHMDIO, ETH_MDC αποτελούν μια θύρα σειριακής επικοινωνίας για την παραμετροποίηση του PHY από τον MCU. [6][8][10]

2.2.2 LoRa

Η σύνδεση του MCU με το τμήμα του LoRa γίνεται μέσω σειριακής θύρας UART και μια επιπλέον ψηφιακή έξοδο για τον έλεγχο του σήματος Reset. Το τμήμα αυτό θα αναλυθεί λεπτομερώς στη σχετική ενότητα. Ο πίνακας που ακολουθεί περιέχει τα σήματα διασύνδεσης μεταξύ MCU και τμήματος LoRa.

Πίνακας 2 Σήματα διασύνδεσης MCU με το τμήμα LoRa

Α/Α	Ονομασία	Περιγραφή	Κατεύθυνση
1	COM_RX	Λήψη δεδομένων από τον MCU	Είσοδος
2	COM_TX	Αποστολή δεδομένων από τον MCU	Έξοδος

3	COM_RTS	Αίτημα αποστολής δεδομένων	Έξοδος
4	COM_CTS	Αποδοχή συνέχισης αποστολής δεδομένων	Είσοδος
5	COM_RST	Έλεγχος σήματος Reset του τμήματος LoRa	Έξοδος

Τα COM_RTS και COM_CTS εξασφαλίζουν ότι κανένα από τμήματα που επικοινωνούν δεν θα αποστέλλει δεδομένα με μεγαλύτερο ρυθμό από αυτόν που μπορεί να επεξεργαστεί ο αποδέκτης. [6][8][10]

2.2.3 USB

Το πρωτόκολλο USB [2] ορίζει δύο ρόλους συσκευών και τις χωρίζει σε τρεις κατηγορίες:

- Hosts: Είναι η συσκευή που ελέγχει τον δίαυλο και συγχρονίζει όλες τις συσκευές. Παράδειγμα host συσκευής είναι ένα PC.
- Devices: Είναι οι συσκευές που συνδέονται σε έναν host κάτω από τον έλεγχο του και ορισμένες περιπτώσεις τροφοδοτούνται από τον host (π.χ. USB memory stick)
- OTG ή Dual role: Μπορούν να λειτουργήσουν είτε σαν host είτε σαν device. Η επιλογή γίνεται από ένα επιπλέον 5^{ο} σήμα εκτός των τεσσάρων που έχουν οι ενός ρόλου συσκευές.

Στην παρούσα υλοποίηση αν και έχει προβλεφθεί η λειτουργία USB OTG δεν έχει υλοποιηθεί στο λογισμικό. Προστέθηκε αποκλειστικά σαν πρόβλεψη για μελλοντική χρήση ή για την κάλυψη ανάγκης που θα μπορούσε να προκύψει κατά την διάρκεια της ανάπτυξης.

Ο πίνακας που ακολουθεί περιέχει τα σήματα διασύνδεσης μεταξύ MCU και USB. [6][8][10]

Πίνακας 3 Σήματα διασύνδεσης MCU με το τμήμα USB

Α/Α	Ονομασία	Περιγραφή	Κατεύθυνση
1	USB_DP	Το data+ του διαφορικού ζεύγους	Αμφίδρομο
2	USB_DN	Το data- του διαφορικού ζεύγους	Αμφίδρομο
3	USB_VBUS	Ανιχνεύει την τάση των 5V του VBUS όταν η συσκευή συνδέεται σε έναν Host	Είσοδος
4	USB_ID	Επιλογής τρόπου λειτουργίας μιας USB OTG συσκευής. (USB OTG)	Είσοδος
5	USB_OC	Ενημερώνει τον MCU ότι το ρεύμα με το οποίο τροφοδοτείται η συσκευή USB που έχει συνδεθεί υπερβαίνει το όριο. (USB OTG)	Είσοδος
6	USB_VBUS_EN	Ενεργοποιεί το διακόπτη ισχύος για την τροφοδότηση της USB συσκευής που έχει συνδεθεί. (USB OTG)	Έξοδος

2.2.4 Αναλογικές είσοδοι

Ο STM32H735 διαθέτει τρεις μετατροπείς αναλογικών σημάτων σε ψηφιακή πληροφορία. Οι ADC1 και ADC2 είναι όμοιοι. Ο ADC3 διαφέρει κυρίως στον μέγιστο ρυθμό δειγματοληψίας. Όλοι μπορούν να ψηφιοποιήσουν ένα σήμα κάθε φορά αλλά χάρη στους ενσωματωμένους αναλογικούς πολυπλέκτες μπορούν να είναι ταυτόχρονα συνδεμένοι σε περισσότερα από ένα σήματα. Το λογισμικό μπορεί (α) να επιλέξει ένα κανάλι και να ξεκινήσει μια δειγματοληψία, (β) να ορίσει μέσω καταχωρητών του ADC για ποια κανάλια και με ποια σειρά θα λαμβάνονται ψηφιακά δείγματα. Είναι ιδιαίτερα ευέλικτα υποσυστήματα περιφερειακά και μπορούν να συνεργαστούν με άλλα περιφερειακά όπως χρονιστές (Timers) για να εξασφαλιστεί σταθερός ρυθμός δειγματοληψίας. Επιπλέον η ικανότητα τους να συνεργάζονται με μονάδες απευθείας πρόσβασης στην μνήμη (DMA, Direct Memory Access) τους επιτρέπει να λειτουργούν παράλληλα με την CPU χωρίς να την επιβαρύνουν. Περισσότερες πληροφορίες για τα ADCs καθώς και για όλα τα περιφερειακά του STM32H735 υπάρχουν διαθέσιμα στα αρχεία προδιαγραφών της STMicroelectronics. [6][8][10]

Ο πίνακας που ακολουθεί περιέχει τα σήματα που συνδέονται στον ADC. Το τμήμα των αναλογικών εισόδων θα αναλυθεί στην αντίστοιχη ενότητα.

Πίνακας 4 Αναλογικά σήματα

Α/Α	Ονομασία	Περιγραφή	Κατεύθυνση
1	AIN1	Αναλογική είσοδος 1	Είσοδος
2	AIN2	Αναλογική είσοδος 2	Είσοδος
3	AIN3	Αναλογική είσοδος 3	Είσοδος
4	AIN4	Αναλογική είσοδος 4	Είσοδος
5	AIN5	Αναλογική είσοδος 5	Είσοδος
6	AIN6	Αναλογική είσοδος 6	Είσοδος
7	AIN7	Αναλογική είσοδος 7	Είσοδος
8	AIN8	Αναλογική είσοδος 1	Είσοδος

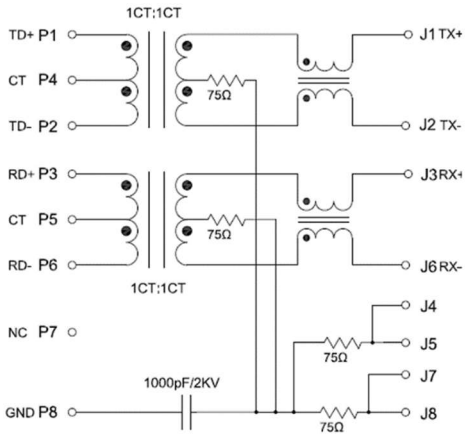
2.3 Τμήμα Ethernet

Το πλήρες σχηματικό διάγραμμα του τμήματος LoRa παρατίθεται στο Παράρτημα Α και Σχήμα 29. Το κεντρικό εξάρτημα του τμήματος είναι το U6, το LAN8742A[11] της Microchip. Για την ενσωμάτωση του στην σχεδίαση εφαρμόστηκε χωρίς καμία παρέκκλιση η ενδεδειγμένη τοπολογία όπως αυτή περιγράφεται στα σχετικά έγγραφα του κατασκευαστή [12][13][14].

Τον χρονισμό του MCU έχει αναλάβει το Y3, ένας τετραγωνικός ταλαντωτής ECS-2520MV-500-BN-TR [9] που παράγει έξοδο συχνότητας 50MHz και πλάτους 3.3V. Εκτός από την είσοδο XTAL1/CLKIN του U6 οδηγεί και την είσοδο ETH_REF_CLK του U3 για να συγχρονιστεί το περιφερειακό MAC του U3 με το U6. Οι αντιστάσεις σειράς R134 και R22 τερματίζουν τις δύο γραμμές για να ελαττωθούν οι ανακλάσεις. Η τιμή των 33Ω είναι ενδεικτική και συνήθως η ακριβής τιμή καθορίζεται πειραματικά γιατί εξαρτάται και από την παρασιτική συμπεριφορά της γραμμής και την αντίσταση εισόδου των οδηγούμενων κυκλωμάτων. Οι μετρήσεις αυτές απαιτούν εξειδικευμένα όργανα που δεν ήταν διαθέσιμα στην διάρκεια της ανάπτυξης.

Το τμήμα τροφοδοτείται από δύο πηγές 3.3V. Η τάση που παράγεται από το κύκλωμα του U18, έναν DC/DC σταθεροποιητή σε τοπολογία Buck συνδέεται στον ακροδέκτη VDDIO του U6 και τροφοδοτεί τα κυκλώματα διασύνδεσης του με το U3, όπως τα σήματα του RMII, nRST και nINT. Η δεύτερη τάση έρχεται από το κύκλωμα του U17, ένα γραμμικό σταθεροποιητή και τροφοδοτεί αποκλειστικά την Ethernet πλευρά του τμήματος. Ο διαχωρισμός της τροφοδοσίας σε ψηφιακή IO και σε Ethernet είναι μια συνήθης πρακτική και προτείνεται από τους κατασκευαστές όπως κάνει και η Microchip στην σχετική τεκμηρίωση για το LAN8742A.

Ο ακροδέκτης Ethernet 100T J15 ARJM11A3-009-AB-ER2-T[15] της Abracon περιέχει, εκτός των επαφών του συνδετήρα, και το κύκλωμα των μετασχηματιστών απομόνωσης (magnetics) όπως ορίζει το πρότυπο του Ethernet [1]. Στο σχήμα που ακολουθεί απεικονίζεται το εσωτερικό κύκλωμα του.



Σχήμα 6 Εσωτερικό κύκλωμα συνδετήρα Ethernet

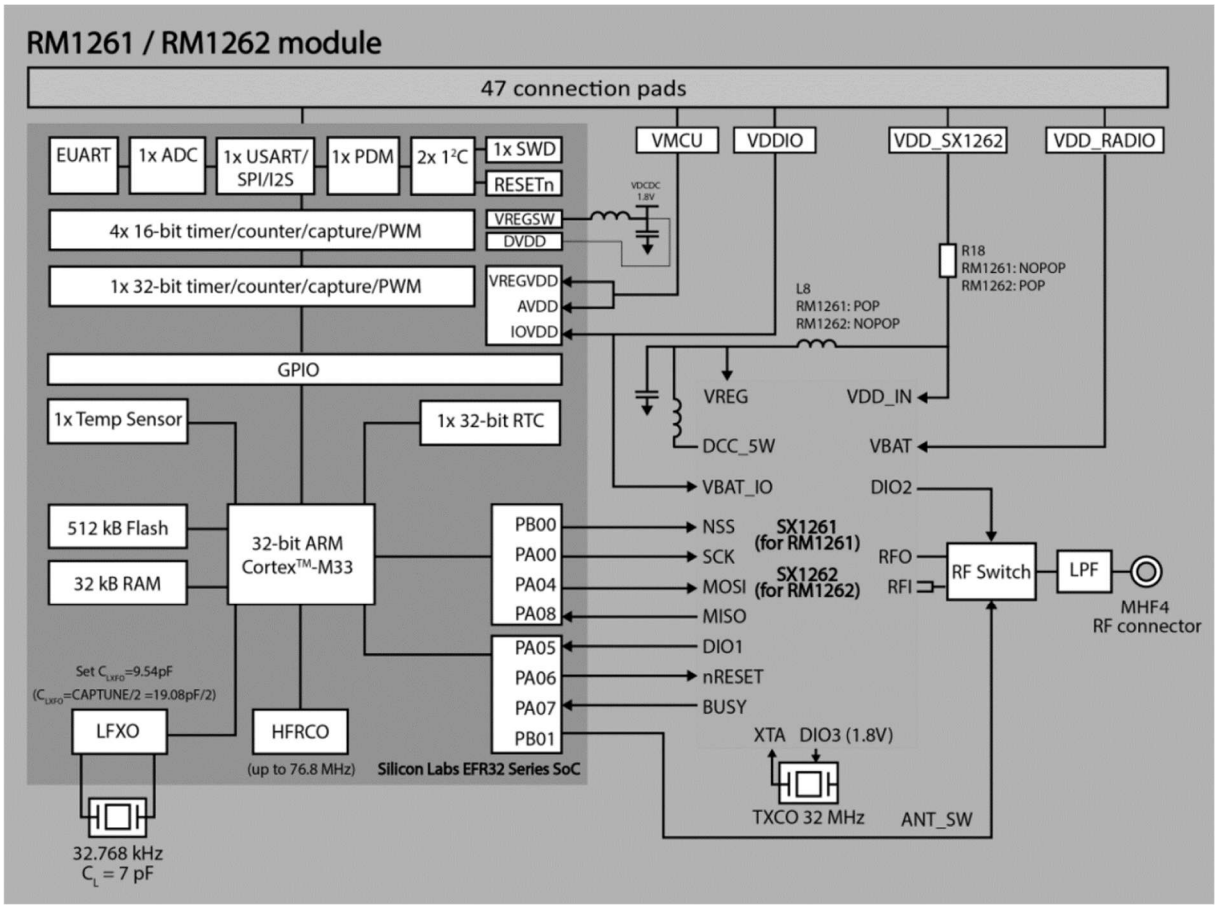
2.4 Τμήμα LoRa

Το πλήρες σχηματικό διάγραμμα του τμήματος LoRa παρατίθεται στο Παράρτημα Α και Σχήμα 30. Τις λειτουργίες του πρωτοκόλλου και την επικοινωνία με τον MCU έχει αναλάβει η αυτόνομη δομική μονάδα (module) RM1261 της eZurio.

Όπως φαίνεται στο Σχήμα 7 εσωτερικά απαρτίζεται από δύο βασικά μέρη:

1. SX1261 της Semtech. Εκτελεί όλες τις λειτουργίες του LoRa καθώς αποτελεί κατοχυρωμένη πνευματική ιδιοκτησία της εταιρείας.
2. EFR32BG22 της Silicon Labs. Είναι ένας γενικής χρήσης ARM Cortex-M33 μικροελεγκτής. Μπορεί να προγραμματιστεί και να λειτουργήσει ως ο βασικός ελεγκτής της εφαρμογής ή να τεθεί υπό τον έλεγχο του κυρίως μικροϋπολογιστή της εφαρμογής. Η δεύτερη περίπτωση έχει επιλεγεί για αυτή την σχεδίαση.

Πρέπει να σημειωθεί εδώ ότι, λόγω της κανονιστικής νομοθεσίας κάθε χώρας ή περιοχής που διέπει τις ασύρματες επικοινωνίες και δει μεγάλης εμβέλειας η εταιρεία κατασκευάζει δύο τύπους. Το RM1261 EU, UKCA, NCC(Taiwan), MIC(Japan), IN(India) και το RM1262 για FCC, ISED, AS/NZS. Εκτός από την κεντρική συχνότητα εκπομπής δεν έχουν καμία διαφορά. [16]



Σχήμα 7 Δομικό διάγραμμα RM1261

Η ενσωμάτωση του module έγινε σύμφωνα με τον προτεινόμενο τρόπο της eZurio όπως αυτός περιγράφετε στα κατασκευαστικά δεδομένα [16] με εξαίρεση το κύκλωμα reset και τον ακροδέκτη προγραμματισμού SWD.

Η είσοδος reset είναι εσωτερικά συνδεμένη μέσω αντίστασης ~35K στην εσωτερική τάση 1.8V η οποία τροφοδοτεί τον πυρήνα του module. Στην είσοδο συνδέθηκαν οι άνοδοι τριών διόδων Schottky που μαζί με την εσωτερική αντίσταση σχηματίζουν την ψηφιακή λογική wired-OR. Οι κάθοδοι των διόδων συνδέονται (α) σε μια έξοδο του MCU, (β) στον ακροδέκτη SWD και (γ) σε ένα διακόπτη πίεσης (push button).

Για τον ακροδέκτη προγραμματισμού χρησιμοποιήθηκε ο αντίστοιχος 6-Pin Needle Adapter της Segger [17]. Επίσης προστέθηκε προστασία από ηλεκτροστατική εκφόρτισης (ESD) στα σήματα του EUART και του SWD. Για τον σκοπό αυτό επιλέχθηκε το USBLC6-4SC6Y[18] της STMicroelectronics. Οι λόγοι που οδήγησαν στην επιλογή του είναι η πολύ χαμηλή χωρητικότητα του (3pF) και το εξαιρετικά μικρό ρεύμα διαρροής (10nA). Μπορεί να προστατεύσει από υπέρταση και βύθιση τέσσερα σήματα και με την δίοδο Zener που περιέχει προστατεύσει την γραμμή τροφοδοσίας από υπερτάσεις. Στο σχηματικό είναι τα U8 και U20 για SWD (J18) και TxD/RxD/CTS/RTS αντίστοιχα.

2.5 Τμήμα USB

Το πλήρες σχηματικό διάγραμμα του τμήματος LoRa παρατίθεται στο Παράρτημα Α και Σχήμα 31.

Τα τέσσερα σήματα VBUS, USB_DN, USB_DP, ID του J21[19] συνδέονται απ' ευθείας με τα USB_VBUS, USB_DN, USB_DP, USB_ID του MCU αντίστοιχα. Το U12[18] παρέχει ESD προστασία και στα τέσσερα σήματα και την τάση VBUS.

Το U11 STMPS2171MTR[20] είναι ένας διακόπτης ισχύος ελεγχόμενος από ένα 3.3V ψηφιακό σήμα. Επιτρέπει την τροφοδότηση της συσκευής που θα συνδεθεί όπως αναλύθηκε στην ενότητα 2.2.3.

Το κύκλωμα του Q1 οδηγείται από την έξοδο FLT του U11 και διεγείρει την δίοδο LED D3 και ταυτόχρονα ενημερώνει τον U3 για την υπέρβαση του ορίου ρεύματος (OC, Over-Current) της συνδεμένης συσκευής. Το σήμα FLT είναι ενεργό σε χαμηλή στάθμη (Active low) και είναι ανοιχτού συλλέκτη. Η R32 κλείνει το κύκλωμα του συλλέκτη και πολώνει το τρανζίστορ. Όταν ενεργοποιείται το FLT τότε η τάση βάση του Q1 γίνεται ίση με 0.4V σύμφωνα με το datasheet του U11 και η VBE γίνεται με την σειρά της αρνητικότερη από -0.7V. Το Q1 οδηγείται σε αγωγή και συγκεκριμένα στο κόρο. Η τάση συλλέκτη γίνεται ίση με 3.3V - Vce(sat) = 3.3V – 0.3V = 3V διεγείροντας την D3 και την είσοδο του U3 που συνδέεται στο σήμα USB_OC. Η R35 περιορίζει το ρεύμα διαμέσου της D3 και η R34 πολώνει τον συλλέκτη του Q1. Οι τιμές των R33, R34 και R35 υπολογίζονται ως εξής.

$$\mathrm{R35} = \frac{VCC - VCE(SAT) - VD3}{Id} \quad (1)$$

για VCC = 3,3V, VCE(SAT) = 0,3V [21], VD3=1.95V [22], Id = 10mA (μισό του ρεύματος δοκιμής) η (1) δίνει R35 = 105Ω ή 100Ω.

$$\mathrm{R33} = \frac{VCC - VBE - VFLT}{Ib} = \frac{VCC - VBE - VFLT}{\frac{Ic}{\beta}} = \frac{\beta \cdot (VCC - VBE - VFLT)}{Id} \quad (2)$$

Επιλέγοντας κατάλληλα την R34 ώστε το ρεύμα διαμέσου της να μπορεί να θεωρηθεί αμελητέο τότε Ic = Id. Για VCC = 3,3V, VBE = 0,7V, β = 100, VFLT=0.4V και Id = 10mA η (2) δίνει R33 = 22K. Επειδή το β ενός τρανζίστορ πυριτίου είναι μια παράμετρος που παρουσιάζει μεγάλη διακύμανση ακόμα και μεταξύ εξαρτημάτων του ίδιου τύπου, επιλέχθηκε μικρότερη τιμή (18K) για να οδηγηθεί το τρανζίστορ πιο βαθιά μέσα στην περιοχή κόρου.

$$R34 = \frac{VCC - VCE(SAT)}{I_{R3}} \quad (3)$$

Για VCC = 3,3V, VCE(SAT) = 0,3V, I_{R34}=150μA η (3) δίνει R34 = 20K (18K).

Η τιμή της R32 δεν είναι κρίσιμη και θα μπορούσε να χρησιμοποιηθεί η 18K αφού υπάρχει ήδη στην λίστα υλικών.

Το κύκλωμα του Q1 οδηγείται από την έξοδο USB_VBUS_EN του U3 και διεγείρει την δίοδο LED D4 και ταυτόχρονα ενεργοποιεί την είσοδο EN του U11 και κατά συνέπεια την έξοδο του. Τα κυκλώματα των Q1 και Q2 είναι πανομοιότυπα και ισχύουν τα ίδια για την ανάλυση της λειτουργίας και των υπολογισμό των τιμών. Επομένως:

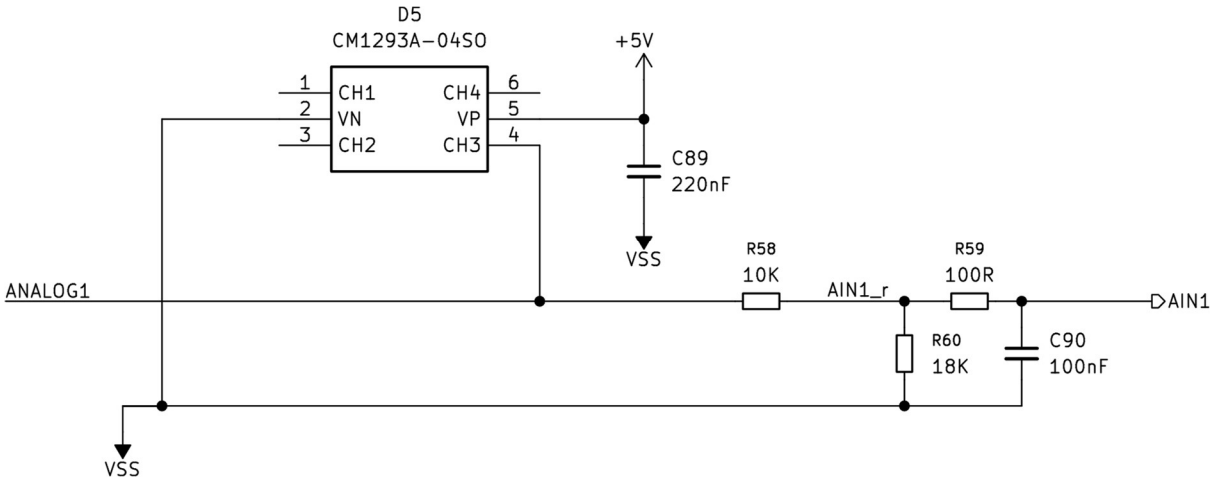
- Q1, Q2 BC857CWT1G [21]
- D3 LTST-C190KRKT [22]
- D4 LTST-C190KGKT [23]
- R32, R33, R34 18K, 1%

- R35 100Ω, 1%
- R36, R37, R39 18K, 1%
- R38 100Ω, 1%

2.6 Τμήμα αναλογικών εισόδων

Το πλήρες σχηματικό διάγραμμα του τμήματος LoRa παρατίθεται στο Παράρτημα Α και Σχήμα 32.

Οι αναλογικές είσοδοι δέχονται σήματα 0-5V. Η μέγιστη τάση που μπορεί να μετρήσει ο MCU είναι 3,3V. Επομένως πρέπει να γίνει κλιμάκωση. Θεωρητικά και στην γενική περίπτωση πρέπει να είναι γνωστά τα χαρακτηριστικά του σήματος αλλά και της πηγής που θα μετρηθεί για να επιλεγεί η καταλληλότερη μέθοδος κλιμάκωσης. Για τις ανάγκες της εργασίας θα χρησιμοποιηθούν ποτενσιόμετρα. Η απλούστερη μέθοδος είναι ο διαιρέτης τάσης με αντιστάσεις. Το τμήμα περιέχει 8 όμοιες βαθμίδες και στην περιγραφή θα γίνει αναφορά μόνο στην πρώτη. Στην συνέχεια παρατίθεται απόσπασμα από το σχηματικό διάγραμμα για την πρώτη είσοδο.



Σχήμα 8 Αναλογική είσοδος

Το D5 παρέχει ESD (CM1293A-04SO[24]) προστασία στις αναλογικές εισόδους και στην τροφοδοσία των 5V που τροφοδοτεί και τις εξωτερικές πηγές που συνδέονται στις εισόδους. Στο Σχήμα 8 οι αντιστάσεις R58 και R60 αποτελούν τον διαιρέτη τάσης για την κλιμάκωση του σήματος εισόδου. Ο λόγος διαίρεσης υπολογίζεται ως εξής:

$$A = \frac{R58}{R58 + R60} = \frac{18K}{10K + 18K} = 0{,}64(4)$$

Επομένως μια τάση 0 - 5V μετατρέπεται σε 0 - 3,2V.

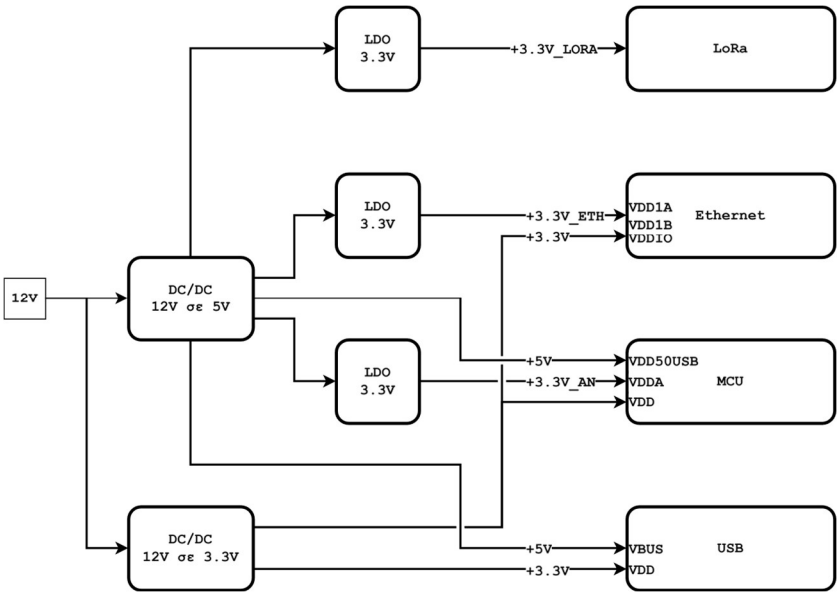
Το RC φίλτρο των R59/C90 έχει συχνότητα αποκοπής 15KHz και τοποθετείται στο τυπωμένο κύκλωμα όσο γίνεται πιο κοντά στον ακροδέκτη του αναλογικού καναλιού του MCU που θα χρησιμοποιηθεί για την δειγματοληψία του. Η συχνότητα του είναι τάξης μεγέθους χαμηλότερη από τις συχνότητες των RMII, USB, LoRa και των dc/dc μετατροπέων που υπάρχουν στο κύκλωμα.

2.7 Τροφοδοσία

Η είσοδος του τμήματος τροφοδοσίας είναι 12VDC και έχει τέσσερις εξόδους 3,3V και μια 5V. Το διάγραμμα βαθμίδων στο Σχήμα 9 που ακολουθεί δίνει μια εποπτική εικόνα της διάρθρωσης του τροφοδοτικού και ποια τμήματα της συσκευής τροφοδοτεί κάθε έξοδος του.

Η μία από τις τέσσερις εξόδους των 3.3V παράγεται από ένα παλμοσταθεροποιητή (SMPS) και αποτελεί την κύρια πηγή τροφοδοσίας των ψηφιακών κυκλωμάτων μεταξύ των οποίων και του μικροελεγκτή. Οι άλλες τρείς δημιουργούνται από γραμμικούς σταθεροποιητές (LDO) και τροφοδοτούν:

- Την πλευρά του Ethernet του PHY (LAN8742A [11]). Ενδείκνυται και συστήνεται από τον κατασκευαστή. Είναι γενικά μια μεθοδολογία που εφαρμόζεται σε σχεδιάσεις που υλοποιούν Ethernet σύνδεση. Ο λόγος είναι η απομόνωση του RMII διαύλου με τα γρήγορα αλλά ασθενή σήματα, σε ότι αφορά το ρεύμα, από τα διαφορικά ζεύγη των εξόδων που οδηγούν τα magnetics του Ethernet.
- Την αναλογική τροφοδοσία του μικροελεγκτή. Εσωτερικά τροφοδοτεί τους ADCs και όλα τα αναλογικά κυκλώματα όπως φαίνεται στο Σχήμα 4. Ο λόγος και εδώ είναι η απομόνωση μεταξύ των κατά κανόνα θορυβωδών ψηφιακών σημάτων από τα χαμηλής τάσης ευαίσθητα αναλογικά. Για παράδειγμα, κυμάτωση της τάξης των 5mV θα μπορούσε να επηρεάσει τις μετρήσεις αν λάβουμε υπόψιν ότι αν η τάση αναφοράς ενός ADC είναι 2,048V και η ανάλυση του 12-bit το 1 LSB είναι 0,5mV.
- Το LoRa Module. Το έγγραφο κατασκευαστικών δεδομένων του module RM1261 [4] αναφέρει ρητά ότι οι τρείς ακροδέκτες τροφοδοσίας που διαθέτει VDD_RADIO, VMCU, VDD_IO πρέπει να συνδεθούν στην ίδια τροφοδοσία. Για να αποφευχθεί η επιστροφή θορύβου προς το δίκτυο των 3,3V κυρίως μέσω του VDD_RADIO, χρησιμοποιήθηκε ένας γραμμικός σταθεροποιητής για να παράγει την τάση αυτή.



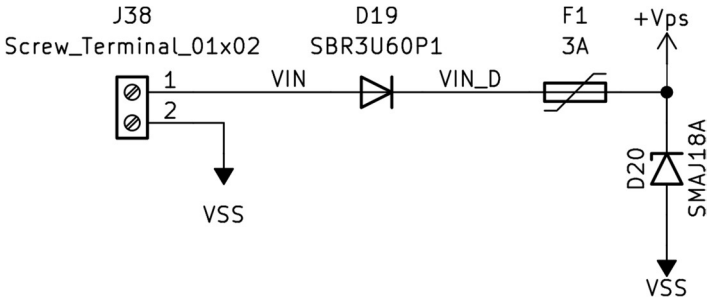
Σχήμα 9 Διάγραμμα βαθμίδων τμήματος τροφοδοσίας

Η τάση των 5V παράγεται επίσης από ένα παλμοσταθεροποιητή και έχει διπλό ρόλο. Πρωτίστως λειτουργεί ως το πρώτο στάδιο υποβιβασμού τάσης για τους LDO των 3,3V και δευτερευόντως αποτελεί την πηγή τάσης παραγωγής της VBUS του USB.

Η κατανάλωση ισχύος ενός γραμμικού σταθερεοποιητή είναι το γινόμενο της διαφοράς τάσης μεταξύ εισόδου και εξόδου και του ρεύματος εξόδου. Μια μεγάλη διαφορά τάσης για ακόμα και ένα σχετικά μικρό ρεύμα μπορούν να μπορεί να οδηγήσει το ολοκληρωμένο κοντά στο όριο του σε ότι αφορά την απαγωγή θερμότητας. Επίσης η ενέργεια αυτή χάνεται αφού μετατρέπεται σε θερμότητα και τελικά ελαττώνεται η απόδοση του κυκλώματος. Αυτοί είναι η λόγοι που επιλέχθηκαν παλμοσταθεροποιητές για τις τροφοδοσίες που πρόκειται να παρέχουν σχετικά μεγάλα ρεύματα. Ο συνδυασμός

παλμοσταθεροποιητή ως πρώτη βαθμίδα που τροφοδοτεί στην συνέχεια γραμμικούς είναι μια συνήθης πρακτική στην βιομηχανία.

Η πηγή τροφοδότησης της συσκευής συνδέεται στον συνδετήρα J38 που μαζί με το κύκλωμα προστασίας της εισόδου απεικονίζονται στο Σχήμα 10. Η δίοδο Schottky D19 (SBR3U60P1 [25]) προστατεύει την συσκευή από λανθασμένη σύνδεση με ανάστροφη πολικότητα, ενώ η ασφάλεια PTC F1 (2920L300/15DR [26]) που ακολουθεί διακόπτει το κύκλωμα αν η ένταση του ρεύματος εισόδου υπερβεί τα 3A. Στην συνέχεια η δίοδος Transil D20 (SMAJ18A [27])προστατεύει από γρήγορες αιχμές τάσης που μπορεί να αναπτυχθούν στην είσοδο τροφοδοσίας είτε μέσω της καλωδίωσης είτε από ESD κατά την επαφή με ηλεκτροστατικά φορτισμένο σώμα.



Σχήμα 10 Κύκλωμα προστασίας εισόδου τροφοδοσίας

Το πλήρες σχηματικό διάγραμμα του τμήματος τροφοδοσίας παρατίθεται στο Παράρτημα Α και Σχήμα 33.

2.7.1 Παλμοσταθεροποιητές υποβιβασμού τάσης από 12V σε 5V και 3.3V

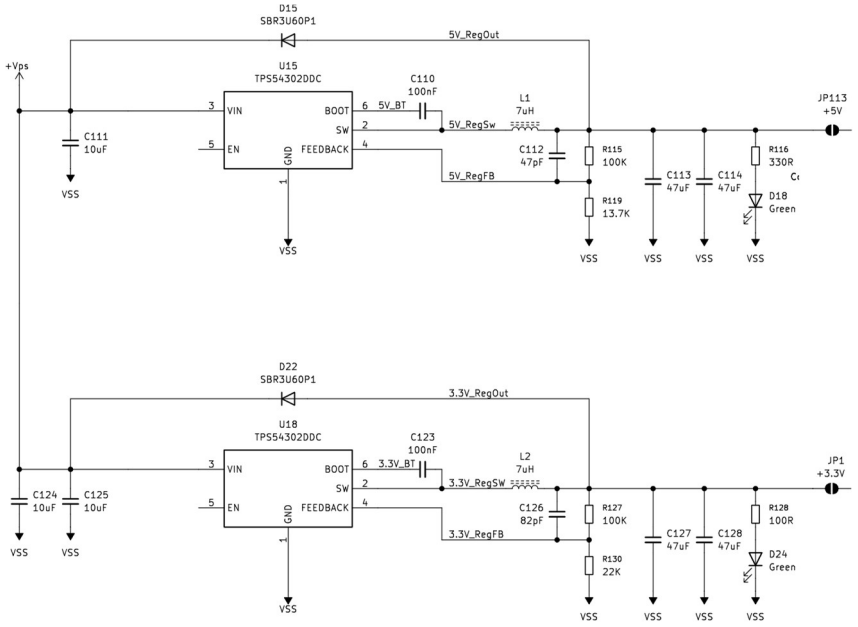
Τα δύο κυκλώματα των U15 και U18 για τα 5V και τα 3,3V αντίστοιχα είναι τοπολογικά όμοια. Διαφέρουν μόνο σε τιμές εξαρτημάτων και συγκεκριμένα:

- Πυκνωτή εισόδου
- Διαιρέτη τάσης στον βρόγχο ανάδρασης
- Πυκνωτή παράκαμψης στον διαιρέτη τάσης του βρόγχου ανάδρασης

Για τον σχεδιασμό και τον υπολογισμό των τιμών των στοιχείων και των δύο κυκλωμάτων χρησιμοποιήθηκε το εργαλείο της Texas Instruments Webench Power Designer.

Η διαδικασία αρχίζει δίνοντας τις επιθυμητές παραμέτρους όπως εύρος τάσης εισόδου, τάση εξόδου και ένταση ρεύματος εξόδου. Το πρόγραμμα προτείνει ένα η περισσότερα κυκλώματα για όσους σταθεροποιητές ικανοποιούν τις απαιτήσεις της σχεδίασης. Επιλέχθηκε το TPS54302DDC [28] με βάση την απλότητα του και το κόστος των παρελκόμενων εξαρτημάτων που το πλαισιώνουν.

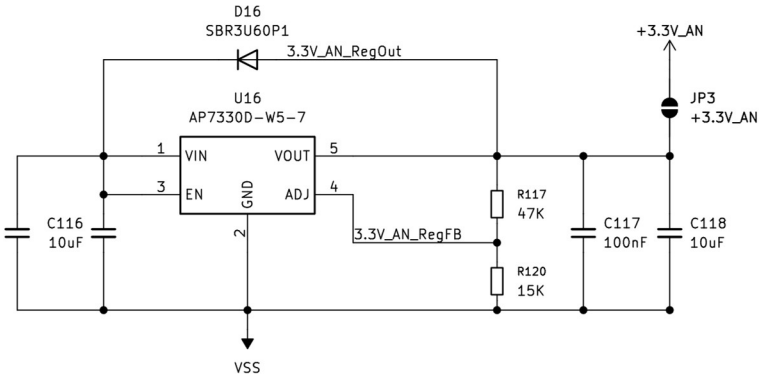
Στο Παράρτημα Β παρατίθεται το σχηματικό διάγραμμα του κυκλώματος όπως παράγεται από το εργαλείο καθώς και οι προτεινόμενη λίστα υλικών. Στη ακόλουθο Σχήμα 11 απεικονίζονται τα δύο τροφοδοτικά όπως αυτά έχουν υλοποιηθεί στην παρούσα εφαρμογή. Για όλα τα στοιχεία έχουν χρησιμοποιηθεί οι προτεινόμενοι τύποι τιμές.



Σχήμα 11 Παλμοσταθεροποιητές υποβιβασμού τάσης από 12V σε 3,3V και 5V

2.7.2 Γραμμικοί σταθεροποιητές από 5V σε 3,3V

Όπως αναφέρθηκε στο 2.7 το τμήμα τροφοδοσίας διαθέτει τα κυκλώματα τριών γραμμικών σταθεροποιητών σε ρόλο πηγών τάσης 3,3V. Τα κυκλώματα είναι πανομοιότυπα και για αυτό θα αναλυθεί μόνο το ένα.



Σχήμα 12 Γραμμικός σταθεροποιητές υποβιβασμού τάσης από 5V σε 3,3V

Ο AP7330D [29] της Diodes Semiconductors επιλέχθηκε για την υψηλή απόρριψη κυμάτωσης που είναι 80dB, τα μόλις 60μVrms θόρυβο εξόδου και τα 15mV ρύθμιση φορτίου (Load Regulation) σε φορτίο 300mA.

Το datasheet ορίζει για τον πυκνωτή εισόδου να είναι κεραμικός και τουλάχιστον 470nF. Οι σταθεροποιητές βρίσκονται μακριά από αυτόν των 5V γιατί είναι τοποθετημένοι κοντά στα τμήματα που τροφοδοτούν και για αυτό ο πυκνωτής εισόδου επιλέχθηκε μεγαλύτερος από την προτεινόμενη τιμή για να δώσει την απαιτούμενη ενέργεια σε απότομες μεταβολές φορτίου.

Η τάση εξόδου καθορίζεται από τις τιμές των αντιστάσεων R117 και R120 σύμφωνα με την σχέση:

$$V_{out} = V_{ref} \cdot \left(1 + \frac{R117}{R120}\right) \Longrightarrow R117 = R120 \cdot \left(\frac{V_{out}}{V_{ref}} - 1\right) (5)$$

Για Vref = 0,8V (από datasheet), Vout = 3,3V, R120 = 18K (επιλογή) η (5) δίνει R177 = 46,875K (47K).

2.8 Σχεδίαση τυπωμένου κυκλώματος

Για τον σχεδιασμό των σχηματικών και του τυπωμένου κυκλώματος (PCB) χρησιμοποιήθηκε το KiCad 9.xx. Ένα πλήρες πακέτο εργαλείων ανοιχτού κώδικα που χρησιμοποιείται διεθνώς από εταιρείες, ερευνητικούς οργανισμούς και εκπαιδευτικά ιδρύματα. Καλύπτει όλη την ροή εργασιών για την σχεδίαση ενός τυπωμένου κυκλώματος από την ιδέα μέχρι την κατασκευή του. Μερικές από τις λειτουργίες του είναι:

- σχεδιασμός μοντέλων εξαρτημάτων
- οργάνωση των μοντέλων σε συλλογές με την μορφή επαναχρησιμοποιήσιμων βιβλιοθηκών
- σχεδίαση σχηματικού διαγράμματος
- σχεδίαση τυπωμένου κυκλώματος
- δημιουργία των απαραίτητων αρχείων για την κατασκευή, όπως για παράδειγμα Gerber, Excellon για την περιγραφή των χαλκοδιαδρόμων (Tracks) και των οπών (Holes) αντίστοιχα.
- επεξεργασία των Gerber αρχείων στις περιπτώσεις που είναι απαραίτητο
- εκτύπωση του σχηματικού και των αρχείων παραγωγής

2.9 Καθορισμός διάταξης στρωμάτων

Στο αρχικό στάδιο κάθε τυπωμένου κυκλώματος ενδείκνυται να επιλεγεί το πλήθος, η φυσική διάταξη και τα χαρακτηριστικά των επιπέδων χαλκού. Στο σχηματικό διάγραμμα τα χαρακτηριστικά των γραμμών που συνδέουν τα στοιχεία του κυκλώματος δεν επηρεάζουν τα κυκλώματα. Αυτό δεν ισχύει στο τυπωμένο κύκλωμα. Οι χαλκοδιάδρομοι είναι μέρος του κυκλώματος και τα φυσικά χαρακτηριστικά τους επιδρούν ουσιαστικά και καθοριστικά στην λειτουργία των κυκλωμάτων. Αυτό καταδεικνύει την σημαντικότητα αυτής της επιλογής.

Το πλήθος και η διάταξη των επιπέδων υπαγορεύεται από:

- την πολυπλοκότητα της σχεδίασης
- την μέγιστη τάση
- την μέγιστη συχνότητα
- τις φυσικές διαστάσεις του τελικού τυπωμένου
- το περιβάλλον λειτουργίας του τυπωμένου
- το πλήθος συνδέσεων

Το τυπωμένο κύκλωμα της παρούσας εργασίας είναι μεσαίας πολυπλοκότητας κυρίως λόγω (α) των εξαρτημάτων του με απόσταση 0,5mm μεταξύ των ακροδεκτών τους και (β) των υψηλών συχνοτήτων του Ethernet (δίαυλος RMII), του USB και του χρονισμού (Clock) των μικροελεγκτή και PHY.

Μια σχεδίαση ελεγχόμενης σύνθετης αντίστασης (Impedance control) επιτρέπει τον σωστό τερματισμό των σημάτων με συνέπεια σημαντικής ποιοτικής βελτίωσης του. Ένας τρόπος για να επιτευχθεί σταθερή και γνωστή σύνθετη αντίσταση για ένα χαλκοδιάδρομο είναι να σχεδιαστεί ως μικρολωρίδα (microstrip). Οι προϋποθέσεις είναι (α) μεταξύ του επιπέδου του και του αμέσως από κάτω του (υπόστρωμα) να υπάρχει ένα στρώμα μονωτικού υλικού γνωστής διηλεκτρικής σταθεράς ε_r και (β) το υπόστρωμα να είναι συνδεμένο στην αναφορά (γείωση) του κυκλώματος. Αν πληρούνται αυτές οι δύο συνθήκες τότε η σύνθετη αντίσταση εξαρτάται μόνο από τα γεωμετρικά χαρακτηριστικά του χαλκοδιαδρόμου. Υπάρχουν μαθηματικοί τύποι που υπολογίζουν με ακρίβεια την σύνθετη αντίσταση αλλά είναι εκτός του σκοπού αυτής της εργασίας. Αντί αυτού υπάρχουν εργαλεία υπολογισμού από κατασκευαστές τυπωμένων κυκλωμάτων και τρίτες πηγές όπως για παράδειγμα της Saturn

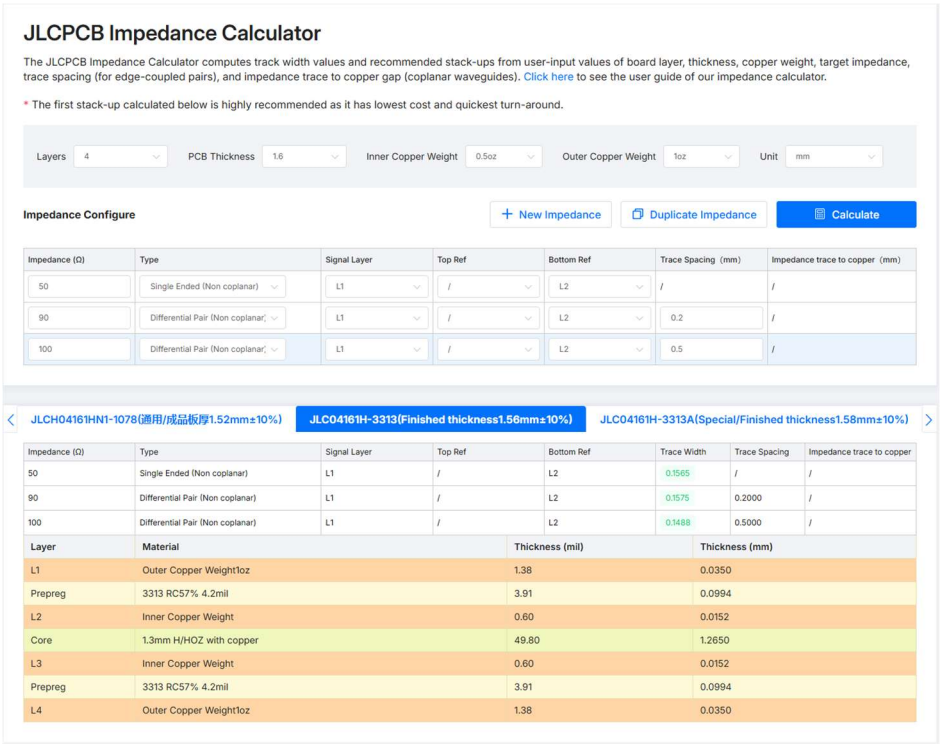
(https://saturnpcb.com/saturn-pcb-toolkit/) που έχει πολύ καλή φήμη για την ακρίβεια των αποτελεσμάτων του. Είναι προτιμότερο και ασφαλέστερο να χρησιμοποιηθεί ο υπολογισμός του κατασκευαστή γιατί λαμβάνονται υπόψιν και παράμετροι σχετικοί με την διαδικασία παραγωγής.

Λαμβάνοντας υπόψιν όλα τα παραπάνω επιλέχθηκε μια διάταξη 4 στρωμάτων σαν μια ισορροπημένη επιλογή μεταξύ απόδοσης και κόστους. Στον πίνακα που ακολουθεί παρατίθενται τα στρώματα και οι κατηγορίες σημάτων που περιέχουν.

Πίνακας 5 Διάταξη στρωμάτων τυπωμένου κυκλώματος

Όνομα	Σήματα
Επάνω Στρώμα	Ελεγχόμενης σύνθετης αντίστασης Χαμηλής συχνότητας
Εσωτερικό 1	Γείωση (Gnd)
Εσωτερικό 2	Γείωση (Gnd)
Κάτω Στρώμα	Αναλογικά Τροφοδοσίας Χαμηλής συχνότητας

Το τυπωμένο κύκλωμα κατασκευάστηκε από την JLC PCB (https://jlcpcb.com/) και το εργαλείο υπολογισμού σύνθετης αντίστασης για το επιλεγμένη διάταξη στρωμάτων (Layer Stackup) έδωσε τα αποτελέσματα της εικόνας που ακολουθεί.



Σχήμα 13 Γεωμετρία στρωμάτων τυπωμένου κυκλώματος

Οι τιμές των στηλών “Trace Width” και “Trace Spacing” μεταφέρθηκαν στους αντίστοιχους σχεδιαστικούς κανόνες στην επόμενη ενότητα.

2.10 Σχεδιαστικοί κανόνες

Το σχηματικό διάγραμμα περιγράφει τον τρόπο με τον οποίο τα εξαρτήματα που απαρτίζουν ένα τυπωμένο κύκλωμα συνδέονται μεταξύ τους. Όσοι ακροδέκτες εξαρτημάτων συνδέονται μεταξύ τους αποτελούν μια σύνδεση (Net) και το σύνολο των Nets αποτελεί το Netlist. Ο σχεδιαστής ορίζει ένα σύνολο κανόνων που διέπουν τα Nets. Συνήθως αυτό γίνεται κατά τον σχεδιασμό του σχηματικού διαγράμματος και μεταφέρεται στο εργαλείο επεξεργασίας του τυπωμένου κυκλώματος μαζί με το Netlist. Για την ευκολότερη διαχείριση των κανόνων και των Nets συχνά τα σχεδιαστικά εργαλεία δίνουν την δυνατότητα ομαδοποίησης των Nets σε κλάσεις (NetClasses). Κάθε κανόνας αντιστοιχίζεται σε ένα μια κλάση και συνέπεια αυτού είναι οποιοδήποτε Net ανήκει σε αυτή διέπετε από όλους τους κανόνες της. Οι κανόνες συνήθως αφορούν το πλάτος των Tracks (Track width), την απόσταση από γειτονικούς που ανήκουν σε άλλο Net (Clearance), την διάμετρο των συνδέσεων μεταξύ των στρωμάτων χαλκού (Via) και την διάμετρο της οπής ενός Via. Ο πίνακας που ακολουθεί περιέχει συγκεντρωμένους όλους τους κανόνες για κάθε Net Class.

Πίνακας 6 Κανόνες κλάσεων συνδέσεων (Netclass Rules)

NetClass	Clearance	Track Width	Via Size	Via Hole	DP Width	DP Gap
USB_DP90	0,35 mm	0,16 mm	0,6 mm	0,3 mm	0,16 mm	0,2 mm
ETH_DP100	0,35 mm	0,15 mm	0,6 mm	0,3 mm	0,15 mm	0,5 mm
HS_Data50	0,35 mm	0,16 mm	0,6 mm	0,3 mm		
LV_Analog	0,50 mm	0,20 mm	0,6 mm	0,3 mm		
HF_Analog	0,50 mm	0,20 mm	0,6 mm	0,3 mm		
Power	0,60 mm	0,30 mm	0,6 mm	0,3 mm		
Ground	0,60 mm	0,30 mm	0,6 mm	0,3 mm		
Default	0,30 mm	0,15 mm	0,6 mm	0,3 mm	0,3 mm	0,3 mm

Οι τιμές της παραμέτρου Clearance υπολογίστηκαν με βάση τον πρακτικό κανόνα του “3W”, όπου W είναι το πλάτος του αγωγού. Δεν είναι πάντα εφικτό να επιτευχθεί αυτό λόγω περιορισμών που προκύπτουν από τις φυσικές διαστάσεις των εξαρτημάτων και την τοποθέτηση τους επάνω στο τυπωμένο κύκλωμα. Σε πραγματικά κυκλώματα βρίσκεται μεταξύ 2W (αποδεκτό όριο) και 2.5W (ισορροπία). Κάτω από αυτό το όριο το ποσοστό αλληλεπίδρασης μεταξύ σημάτων σε γειτονικούς χαλκοδιαδρόμους μπορεί υπό ορισμένες συνθήκες και ανάλογα με την φύση των σημάτων (υψηλής συχνότητας, ισχυρού ρεύματος, ταχέως μεταβαλλόμενα κ.λπ.) να αλλοιώσει τα σήματα.

Στο Παράρτημα Γ παρατίθενται τα παραγόμενα αρχεία παραγωγής του τυπωμένου κυκλώματος.

Κεφάλαιο 3ο: Θεωρία Χάους και Ασφάλεια Δεδομένων

Η θεωρία του χάους αποτελεί έναν από τους πιο ενδιαφέροντες κλάδους της σύγχρονης μαθηματικής φυσικής, κυρίως επειδή καταδεικνύει ότι ένα σύστημα μπορεί να είναι απολύτως ντετερμινιστικό και ταυτόχρονα να εμφανίζει συμπεριφορά που, σε πρακτικό επίπεδο, μοιάζει με τυχαιότητα. Το στοιχείο αυτό έχει οδηγήσει σε πλήθος εφαρμογών σε κλάδους όπως η μοντελοποίηση πολύπλοκων φαινομένων, η επεξεργασία σήματος και, πιο πρόσφατα, η κρυπτογράφηση. Η ιδέα ότι ένα απλό μαθηματικό σύστημα μπορεί να παράγει εξαιρετικά πολύπλοκες ακολουθίες είναι ιδιαίτερα ελκυστική σε περιβάλλοντα όπου απαιτείται υψηλή μεταβλητότητα, χαμηλό κόστος υπολογισμού και καλή κλιμάκωση σε ενσωματωμένες συσκευές.

Στα δυναμικά συστήματα, ο όρος «χάος» δεν συνδέεται με αταξία ή τυχαιότητα με την καθημερινή έννοια. Αντιθέτως, περιγράφει μια κατάσταση όπου η μελλοντική συμπεριφορά εξαρτάται σε τέτοιο βαθμό από τις αρχικές συνθήκες, ώστε μικρές αποκλίσεις να οδηγούν σε τελείως διαφορετική εξέλιξη [30], [31]. Αυτή η ιδιότητα, γνωστή ως ευαισθησία στις αρχικές συνθήκες, αποτελεί το βασικό χαρακτηριστικό που οδήγησε στην αξιοποίηση των χαοτικών χαρτών ως πηγών ψευδοτυχαίων ακολουθιών σε εφαρμογές κρυπτογράφησης. Ειδικά σε ενσωματωμένα συστήματα, όπου ο απαιτούμενος όγκος πράξεων πρέπει να παραμένει χαμηλός, οι χαοτικές συναρτήσεις προσφέρουν μια εναλλακτική λύση στις παραδοσιακές, πιο βαριές κρυπτογραφικές γεννήτριες.

3.1 Ντετερμινιστικότητα και απροσδιοριστία

Ο πυρήνας της θεωρίας χαοτικών συστημάτων βασίζεται στην αντίφαση ότι μια αυστηρά καθορισμένη μαθηματική συνάρτηση μπορεί να παράγει συμπεριφορά που, χωρίς πλήρη γνώση των αρχικών τιμών, μοιάζει αδύνατο να προβλεφθεί. Αυτό δεν σημαίνει ότι το σύστημα είναι τυχαίο. Αντίθετα, είναι πλήρως ντετερμινιστικό: η ίδια αρχική κατάσταση οδηγεί πάντα στα ίδια αποτελέσματα. Το πρόβλημα ανακύπτει στη διάρκεια, καθώς η οποιαδήποτε διαφορά στην αρχική τιμή — ακόμη και σε επίπεδο 10^{-6} — επηρεάζει την εξέλιξη τόσο έντονα ώστε μετά από λίγες επαναλήψεις η τροχιά να έχει αλλάξει ριζικά.

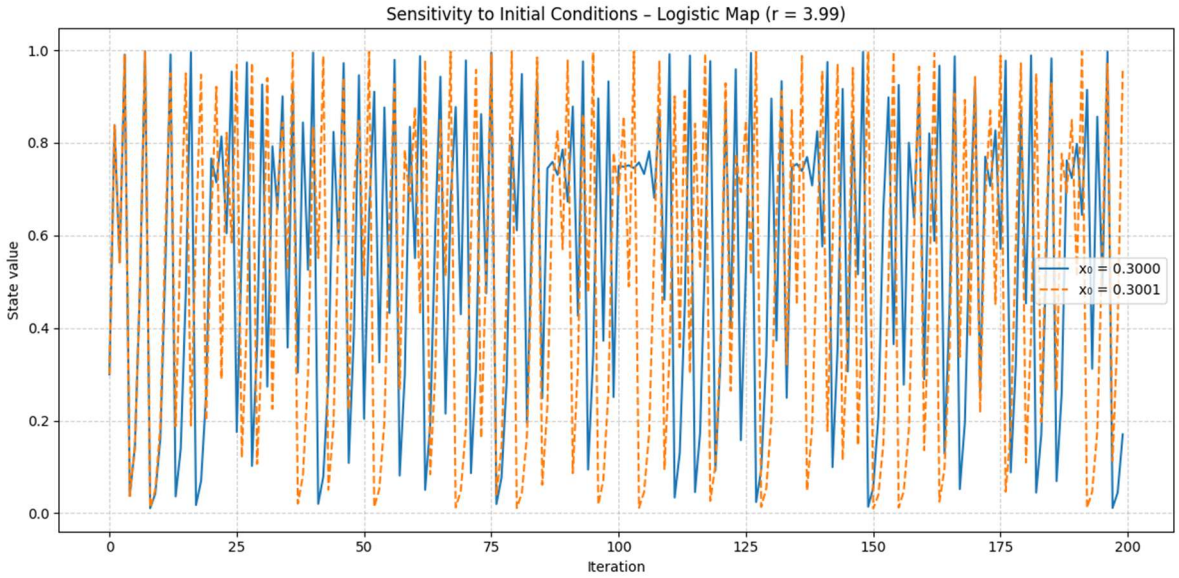
Αυτή η ιδιότητα έχει άμεση αξία στην κρυπτογράφηση. Ένα κλειδί που διαφέρει έστω και κατά ένα bit οδηγεί σε εντελώς διαφορετική ακολουθία, κάτι που ενισχύει σημαντικά τη διάχυση και τη μη προβλεψιμότητα του παραγόμενου keystream.

3.2 Ευαισθησία στις αρχικές συνθήκες

Ένα από τα πιο χαρακτηριστικά παραδείγματα χαοτικού συστήματος είναι ο logistic map. Παρότι ο ορισμός του είναι εξαιρετικά απλός, η συμπεριφορά του στη χαοτική περιοχή παρουσιάζει όλες τις βασικές ιδιότητες που ενδιαφέρουν την κρυπτογραφία.

Παρακάτω παρουσιάζεται ένα διάγραμμα που δείχνει τη διαφορά στην εξέλιξη δύο σχεδόν ταυτόσημων αρχικών τιμών $x_0 = 0.3000$ και $x_0 = 0.3001$.

Η δημιουργία του γραφήματος πραγματοποιήθηκε με χρήση προγράμματος σε γλώσσα Python. Ο κώδικας περιέχεται στο Παράρτημα Δ (ch1_LogisticMap_Sensitivity_InitialConditions.py).



Σχήμα 14 Ευαισθησία του «Logistic Map» στις αρχικές τιμές

Μετά από λίγες επαναλήψεις, οι δύο τροχιές δεν έχουν καμία εμφανή σχέση μεταξύ τους, παρότι η αρχική διαφορά είναι της τάξης του 0.0001. Το φαινόμενο αυτό είναι κεντρικό στην αξιοποίηση των χαοτικών συναρτήσεων ως μηχανισμό παραγωγής ψευδοτυχαίου θορύβου.

Η έντονη απόκλιση των τροχιών εμφανίζεται πολύ νωρίς, κάτι που έχει άμεση χρησιμότητα στις ανάγκες ενός κρυπτογραφικού συστήματος: ακόμη και μικρή τροποποίηση των seeds οδηγεί σε εντελώς διαφορετική παραγόμενη ακολουθία, εμποδίζοντας οποιαδήποτε απόπειρα πρόβλεψης.

3.3 Μη γραμμικότητα και πολύπλοκες τροχιές

Σε αντίθεση με συστήματα που ορίζονται από γραμμικές σχέσεις, όπου η συμπεριφορά μπορεί συνήθως να αναλυθεί με κλασικές μεθόδους, τα χαοτικά συστήματα χαρακτηρίζονται από έντονη μη γραμμικότητα. Αυτό έχει δύο συνέπειες που ενδιαφέρουν την ασφάλεια των δεδομένων:

1. Η έξοδος δεν μπορεί να προβλεφθεί ούτε με απλό ούτε με προχωρημένο γραμμικό μοντέλο.
2. Η στατιστική κατανομή των τιμών τείνει να είναι πιο ομοιόμορφη, γεγονός που μειώνει το ενδεχόμενο επιθέσεων που εκμεταλλεύονται στατιστικά μοτίβα στην παραγόμενη ακολουθία.

Η μη γραμμικότητα προσφέρει ένα επίπεδο προστασίας που δεν επιτυγχάνεται εύκολα με κλασικούς αλγορίθμους χωρίς χρήση πολύπλοκων μετασχηματισμών.

3.4 Σύνδεση με Κρυπτογράφηση

Η κρυπτογράφηση βασίζεται στην παραγωγή ακολουθιών που παρουσιάζουν υψηλό βαθμό εντροπίας και είναι δύσκολο να προβλεφθούν χωρίς γνώση του κλειδιού. Τα χαοτικά συστήματα, παρότι ντετερμινιστικά, παράγουν ακολουθίες που μοιάζουν στατιστικά με τυχαίες, ειδικά όταν ενσωματωθούν τεχνικές όπως, διαδοχική επανάληψη (iterative amplification), cross-coupling μεταξύ διαφορετικών χαρτών και μεταφορά της χαοτικής κατάστασης σε bits μέσω απλών μηχανισμών μίξης. Στα ενσωματωμένα συστήματα, όπου η ταχύτητα, η κατανάλωση και η μνήμη αποτελούν περιορισμούς, η δυνατότητα να επιτευχθεί υψηλή κρυπτογραφική ποιότητα με ελαφριά μαθηματικά είναι σημαντικό πλεονέκτημα. Οι παραπάνω έννοιες αξιοποιούνται στον υβριδικό αλγόριθμο που παρουσιάζεται στο Κεφάλαιο 4.

3.4.1 Διαδοχική επανάληψη (iterative amplification)

Στα χαοτικά συστήματα, η έξοδος κάθε επανάληψης χρησιμοποιείται ως είσοδος της επόμενης. Η διαδικασία αυτή, γνωστή ως διαδοχική επανάληψη, έχει ως αποτέλεσμα την ενίσχυση ακόμη και απειροελάχιστων διαφορών στις αρχικές τιμές του συστήματος.

Στην πράξη, μια πολύ μικρή μεταβολή στην αρχική τιμή οδηγεί, μετά από επαρκή αριθμό επαναλήψεων, σε πλήρως διαφορετική ακολουθία τιμών. Το φαινόμενο αυτό συνδέεται άμεσα με την έννοια της ευαισθησίας στις αρχικές συνθήκες και αποτελεί θεμελιώδες χαρακτηριστικό της χαοτικής συμπεριφοράς.

Από κρυπτογραφικής πλευράς, η διαδοχική επανάληψη λειτουργεί ως μηχανισμός ενίσχυσης της διάχυσης, καθώς μικρές αλλαγές στο κλειδί ή στις αρχικές παραμέτρους επηρεάζουν δραστικά το τελικό αποτέλεσμα.

3.4.2 Διασύνδεση (cross-coupling) μεταξύ διαφορετικών χαρτών

Η χρήση περισσότερων του ενός χαοτικών χαρτών επιτρέπει τη δημιουργία πιο σύνθετης δυναμικής συμπεριφοράς. Όταν η έξοδος ενός χαοτικού χάρτη χρησιμοποιείται ως είσοδος ή παράμετρος σε έναν δεύτερο, τότε προκύπτει διασύνδεση (cross-coupling) μεταξύ των συστημάτων. Η διασύνδεση αυτή:

- αυξάνει τη μη γραμμικότητα του συνολικού συστήματος,
- δυσχεραίνει την αναλυτική πρόβλεψη της εξέλιξής του,
- περιορίζει τη δυνατότητα ανακατασκευής της δυναμικής από μεμονωμένες παρατηρήσεις.

Σε κρυπτογραφικές εφαρμογές, το cross-coupling συμβάλλει στην αύξηση της πολυπλοκότητας του χώρου κλειδιών, καθώς η παραβίαση ενός μεμονωμένου χαρτή δεν επαρκεί για την αναπαραγωγή της συνολικής χαοτικής συμπεριφοράς.

3.4.3 Μεταφορά της χαοτικής κατάστασης σε δυαδική μορφή μέσω μηχανισμών μίξης

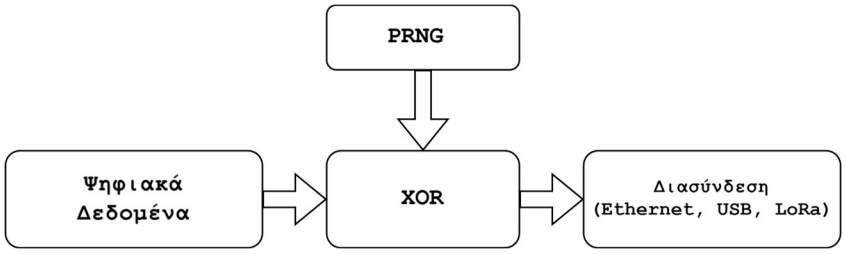
Οι χαοτικοί χάρτες παράγουν τιμές συνεχούς ή υψηλής ακρίβειας αριθμητικής αναπαράστασης. Για να χρησιμοποιηθούν σε ψηφιακά κρυπτογραφικά συστήματα, απαιτείται η μεταφορά της χαοτικής κατάστασης σε δυαδική μορφή. Η μεταφορά αυτή δεν γίνεται άμεσα, αλλά μέσω απλών μηχανισμών μίξης, όπως:

- Διαχωρισμός βάσει ορίου (thresholding)
- Αποκοπή επιλεγμένων bits
- Συνδυασμός χαοτικών τιμών με βασικές λογικές ή αριθμητικές πράξεις.

Οι μηχανισμοί μίξης λειτουργούν ως διεπαφή μεταξύ του συνεχούς χαοτικού χώρου και της ψηφιακής αναπαράστασης, διατηρώντας τη στοχαστική και μη προβλέψιμη φύση του συστήματος. Με τον τρόπο αυτό, η παραγόμενη δυαδική ακολουθία μπορεί να αξιοποιηθεί ως ροή κλειδιού ή ως δυναμική παράμετρος κρυπτογραφικού αλγορίθμου.

3.5 Πλαίσιο εφαρμογής στη μετάδοση αναλογικών μετρήσεων

Στο πλαίσιο συστημάτων που συλλέγουν αναλογικές μετρήσεις και τις μεταδίδουν μέσω διεπαφών όπως USB, Ethernet ή LoRa, η ανάγκη για μια ελαφριά αλλά αποτελεσματική μέθοδο κρυπτογράφησης είναι ιδιαίτερα έντονη. Οι χαοτικές συναρτήσεις μπορούν να ενσωματωθούν άμεσα στο pipeline της επεξεργασίας όπως φαίνεται στο Σχήμα 15.



Σχήμα 15 Διάγραμμα ροής δεδομένων

Η διαδικασία αυτή είναι ταχεία, συνεπώς ταιριάζει στις απαιτήσεις πραγματικού χρόνου του STM32H7, και δεν επηρεάζει την καθυστέρηση της συνολικής μετάδοσης. Επιπλέον, επιτρέπει στον σχεδιαστή να αξιοποιήσει τα χαρακτηριστικά του χάους χωρίς να εφαρμόσει πλήρη, βαριά κρυπτογραφία — κάτι που δεν είναι απαραίτητο σε ένα σύστημα τηλεμετρίας, αλλά η απόκρυψη των δεδομένων είναι υποχρεωτική.

Κεφάλαιο 4ο: Χαοτικοί Χάρτες και Ψηφιακή Υλοποίηση

Η αξιοποίηση χαοτικών συστημάτων στην κρυπτογράφηση βασίζεται σε ορισμένες απλές, αλλά ιδιαίτερα ισχυρές μαθηματικές συναρτήσεις, οι οποίες παράγουν πολύπλοκες ακολουθίες μέσω επαναληπτικών διαδικασιών. Τα συστήματα αυτά, παρότι φαινομενικά απλά στη διατύπωσή τους, εμφανίζουν συμπεριφορές που είναι πρακτικά απρόβλεπτες χωρίς ακριβή γνώση των αρχικών συνθηκών. Στο κεφάλαιο αυτό παρουσιάζονται οι πιο γνωστοί χαοτικοί χάρτες που έχουν χρησιμοποιηθεί σε μελέτες κρυπτογράφησης και ψευδοτυχαίων γεννητριών, καθώς και τα ζητήματα που προκύπτουν κατά τη μετάβασή τους από το αναλογικό στο ψηφιακό περιβάλλον.

4.1 Η έννοια του διακριτού χάρτη

Οι περισσότεροι χαοτικοί αλγόριθμοι που χρησιμοποιούνται σε συστήματα ψηφιακής κρυπτογράφησης στηρίζονται σε συναρτήσεις της μορφής (Σχέση 6):

$$x(n+1) = f\big(x(n)\big) (6)$$

Το χαρακτηριστικό που τους καθιστά τόσο ισχυρούς είναι ότι, αν η συνάρτηση f έχει τις κατάλληλες ιδιότητες, τότε η επαναλαμβανόμενη εφαρμογή της δημιουργεί ακολουθίες που παρουσιάζουν υψηλή ευαισθησία στις αρχικές τιμές. Η κεντρική ιδέα είναι ότι ολόκληρη η πολυπλοκότητα ενός χαοτικού συστήματος μπορεί να αναπαρασταθεί ψηφιακά με μία μόνο μεταβλητή κατάστασης, η οποία ενημερώνεται σε κάθε βήμα βάσει του ίδιου μαθηματικού κανόνα. Οι ακολουθίες που παράγονται μπορούν να χρησιμοποιηθούν απευθείας ως πηγή εντροπίας για κρυπτογραφικούς μηχανισμούς ροής, εφόσον μετασχηματιστούν κατάλληλα σε δυαδική μορφή.

Στην παραπάνω διατύπωση, η μεταβλητή x(n) εκφράζει την κατάσταση του χαοτικού συστήματος στο n-οστό διακριτό χρονικό βήμα, ενώ η συνάρτηση $f(\chi(n))$ ορίζει τον κανόνα με τον οποίο η κατάσταση αυτή εξελίσσεται. Σε κάθε επανάληψη, η προηγούμενη τιμή της κατάστασης χρησιμοποιείται ως είσοδος για τον υπολογισμό της επόμενης, χωρίς την ανάγκη αποθήκευσης επιπλέον ιστορικών δεδομένων.

Πρακτικά, αυτό σημαίνει ότι το σύστημα «θυμάται» το παρελθόν του αποκλειστικά μέσω της τρέχουσας τιμής της μεταβλητής κατάστασης. Παρότι ο μαθηματικός κανόνας παραμένει σταθερός, η

επαναλαμβανόμενη εφαρμογή του οδηγεί σε ακολουθίες τιμών με έντονα μη γραμμική και μη προβλέψιμη συμπεριφορά, ιδίως όταν η συνάρτηση f λειτουργεί σε περιοχές χαοτικής δυναμικής.

Η ιδιαιτερότητα αυτή καθιστά τους χαοτικούς χάρτες ιδιαίτερα κατάλληλους για ψηφιακή κρυπτογράφηση, καθώς μια ελάχιστη μεταβολή στην αρχική τιμή x(0) οδηγεί, μετά από μικρό αριθμό επαναλήψεων, σε πλήρως διαφορετική εξέλιξη της ακολουθίας. Έτσι, ένα σύστημα που περιγράφεται από μία μόνο μεταβλητή κατάστασης μπορεί να παράγει ακολουθίες υψηλής εντροπίας, ικανές να χρησιμοποιηθούν ως βάση για μηχανισμούς κρυπτογράφησης ροής.

Συνοπτικά ο διακριτός χάρτης περιγράφει τον κανόνα μετάβασης από την τρέχουσα κατάσταση x(n) στην επόμενη x(n+1). Η χρονική ακολουθία προκύπτει έμμεσα, μέσω της επαναλαμβανόμενης εφαρμογής του ίδιου χάρτη. Η γραφική απεικόνιση του είναι το σύνολο των σημείων ενός καρτεσιανού συστήματος αξόνων XY τοποθετώντας τις τιμές x(n) στον άξονα X και x(n+1) στον άξονα Y.

4.2 Logistic Map

Ο logistic map αποτελεί έναν από τους πλέον μελετημένους διακριτούς χαοτικούς χάρτες [32] και ορίζεται από τη μη γραμμική επαναληπτική σχέση (Σχέση 7):

$$x(n+1) = r \cdot x(n) \cdot \big(1 - x(n)\big) \quad (7)$$

όπου x(n) είναι η μεταβλητή κατάστασης στο n-οστό διακριτό βήμα και r μία πραγματική παράμετρος ελέγχου της δυναμικής του συστήματος.

Παρά την εξαιρετικά απλή μαθηματική του μορφή, ο logistic map παρουσιάζει πλούσια και σύνθετη συμπεριφορά, η οποία μεταβάλλεται δραστικά ανάλογα με την τιμή της παραμέτρου r. Για μικρές τιμές του r, το σύστημα συγκλίνει σε σταθερό σημείο, ενώ για ενδιάμεσες τιμές εμφανίζει περιοδική συμπεριφορά. Όταν η παράμετρος r προσεγγίζει τιμές κοντά στο 4 (π.χ. r≈3,99), η συμπεριφορά του συστήματος παύει να είναι περιοδική και εισέρχεται στη χαοτική περιοχή, όπου η εξέλιξη της μεταβλητής κατάστασης χαρακτηρίζεται από έντονη ευαισθησία στις αρχικές τιμές και μη προβλεψιμότητα [33].

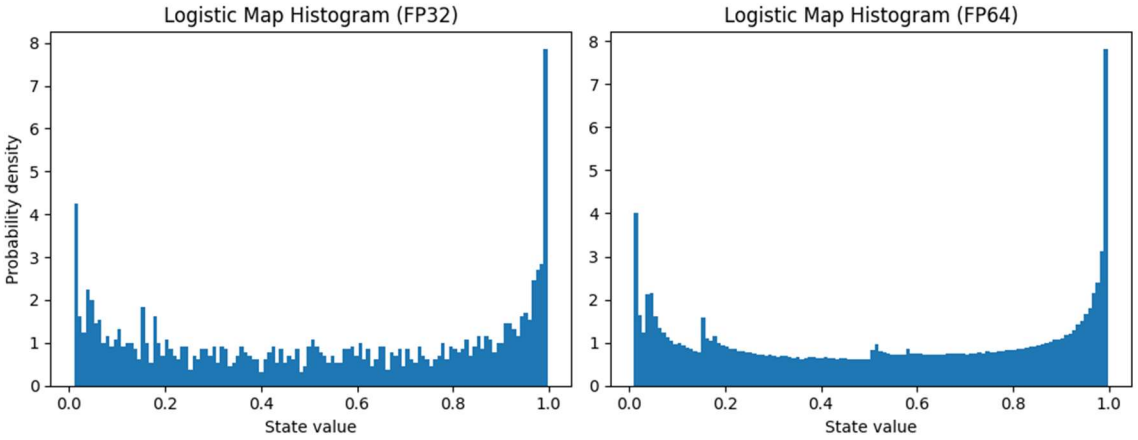
Ένα από τα βασικά πλεονεκτήματα του logistic map είναι ότι η μεταβλητή κατάστασης x(n) παραμένει στο κανονικοποιημένο διάστημα (0,1) για κατάλληλες τιμές της παραμέτρου r. Το περιορισμένο αυτό εύρος ορισμού εξασφαλίζει ότι οι παραγόμενες τιμές δεν εκτρέπονται αριθμητικά, γεγονός που καθιστά τον χάρτη ιδιαίτερα κατάλληλο για ψηφιακή υλοποίηση. Συγκεκριμένα, οι τιμές της μεταβλητής κατάστασης μπορούν να αναπαρασταθούν απευθείας σε μεταβλητές κινητής υποδιαστολής, χωρίς την ανάγκη πρόσθετης κλιμάκωσης ή ελέγχου ορίων.

Η χαοτική ακολουθία που προκύπτει από την επαναλαμβανόμενη εφαρμογή του logistic map μπορεί, υπό κατάλληλο μετασχηματισμό, να αξιοποιηθεί ως πηγή εντροπίας σε κρυπτογραφικούς μηχανισμούς ροής. Ωστόσο, παρά τη χαοτική του φύση, ο logistic map από μόνος του δεν επαρκεί για την κάλυψη αυστηρών κρυπτογραφικών απαιτήσεων, καθώς η απλή δομή του μπορεί να οδηγήσει σε συσχετίσεις ή στατιστικές αδυναμίες όταν χρησιμοποιείται αυτούσιος.

Για τον λόγο αυτό, στη σύγχρονη βιβλιογραφία προτείνεται συχνά η χρήση του logistic map ως δομικού στοιχείου σε πιο σύνθετα ή υβριδικά σχήματα, όπου συνδυάζεται με πρόσθετους χαοτικούς μηχανισμούς ή διαδικασίες μίξης. Με τον τρόπο αυτό, διατηρούνται τα πλεονεκτήματα της απλότητας και της αποδοτικότητας, ενώ παράλληλα ενισχύεται η κρυπτογραφική ισχύς του συνολικού συστήματος.

4.2.1 Πειραματική απεικόνιση σε ψηφιακή υλοποίηση

Η δημιουργία του γραφήματος πραγματοποιήθηκε με χρήση προγράμματος σε γλώσσα Python. Ο κώδικας περιέχεται στο Παράρτημα Δ (ch2_logistic_fp32_vs_fp64_histogram.py).



Σχήμα 16 Ιστόγραμμα τιμών κατάστασης του logistic map σε αριθμητική FP32 και FP64

4.2.2 Ερμηνεία πειραματικών αποτελεσμάτων

Ο logistic map αποτελεί έναν από τους πλέον μελετημένους χαοτικούς μηχανισμούς και συχνά χρησιμοποιείται ως βάση για γεννήτριες ψευδοτυχαίων αριθμών. Θεωρητικά, για κατάλληλες τιμές της παραμέτρου, παρουσιάζει έντονη μη γραμμικότητα και υψηλή ευαισθησία στις αρχικές συνθήκες.

Στο πλαίσιο της παρούσας εργασίας, η συμπεριφορά του logistic map εξετάζεται και πειραματικά μέσω ιστογραμμάτων των τιμών κατάστασης σε αριθμητική κινητής υποδιαστολής FP32 και FP64. Στην περίπτωση FP64, η κατανομή των τιμών είναι ομαλή και συνεχής, γεγονός που συμφωνεί με τη θεωρητική περιγραφή του χάρτη. Αντίθετα, στην περίπτωση FP32 παρατηρούνται έντονες τοπικές συγκεντρώσεις τιμών καθώς και περιοχές με μειωμένη πυκνότητα.

Η συμπεριφορά αυτή υποδεικνύει ότι, σε περιβάλλον πεπερασμένης ακρίβειας, ο logistic map δεν εξερευνά ομοιόμορφα τον χώρο καταστάσεων, αλλά τείνει να ακολουθεί προτιμώμενες τροχιές λόγω αριθμητικής στρογγυλοποίησης. Ως αποτέλεσμα, μειώνεται η αποτελεσματική εντροπία της παραγόμενης ακολουθίας, γεγονός που περιορίζει την καταλληλότητά του ως αυτόνομη πηγή ψευδοτυχαίων αριθμών σε embedded συστήματα.

4.3 Ο Tent Map

Ο tent map αποτελεί έναν από τους απλούστερους και θεωρητικά αναλύσιμους χαοτικούς χάρτες, λόγω της κατά τμήματα γραμμικής δομής του (Σχέση 8) [34]:

$$x(n+1) = \begin{cases} \mu \cdot x(n) & για\ x(n) < 0.5 \\ \mu \cdot \big(1 - x(n)\big) & για x(n) \geq 0.5 \end{cases} \quad (8)$$

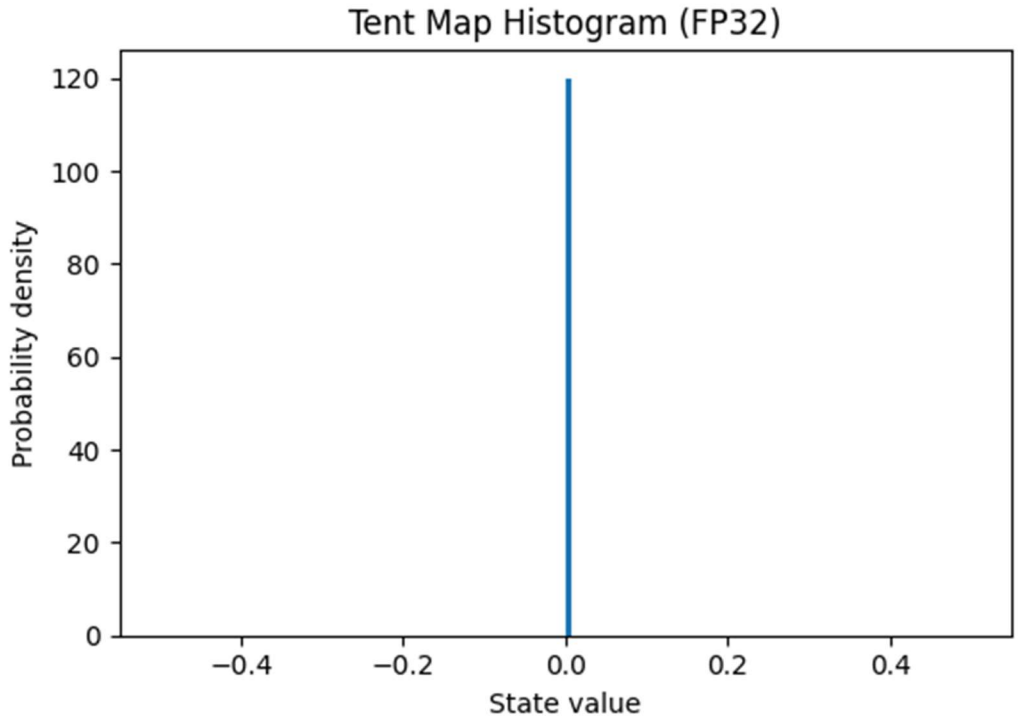
όπου x(n) είναι η μεταβλητή κατάστασης στο n-οστό διακριτό βήμα και μ μία πραγματική παράμετρος ελέγχου της δυναμικής του συστήματος.

Για την τιμή μ = 2, το σύστημα βρίσκεται στο μέγιστο επίπεδο χάους. Η μορφή του tent map τον καθιστά θεωρητικά κατάλληλο για ψηφιακά περιβάλλοντα· ωστόσο, η ψηφιακή υλοποίηση σε περιορισμένη αριθμητική ακρίβεια μπορεί να παρουσιάσει σημαντικές αποκλίσεις, όπως θα φανεί στη συνέχεια. Αυτό

τον διαχωρίζει από τον logistic map, όπου η μη γραμμικότητα μπορεί να συμπιεστεί σε μικρότερο αριθμό διακριτών τιμών όταν οι υπολογισμοί γίνονται σε FP32. Ένα ακόμη πλεονέκτημα είναι ότι η στατιστική του κατανομή τείνει να είναι πιο ομοιόμορφη, κάτι που είναι επιθυμητό όταν οι παραγόμενες τιμές χρησιμοποιούνται για παραγωγή κρυπτογραφικού keystream.

4.3.1 Πειραματική απεικόνιση σε ψηφιακή υλοποίηση

Η δημιουργία του γραφήματος πραγματοποιήθηκε με χρήση προγράμματος σε γλώσσα Python. Ο κώδικας περιέχεται στο Παράρτημα Δ (ch2_tent_map_fp32_histogram.py).



Σχήμα 17 Ιστόγραμμα τιμών κατάστασης του tent map σε αριθμητική FP32

Δεν παρουσιάζεται αντίστοιχο γράφημα του tent map σε αριθμητική FP64, καθώς σε συνθήκες υψηλής ακρίβειας η συμπεριφορά του χάρτη είναι αναμενόμενη και δεν διαφοροποιείται ουσιαστικά από τη θεωρητική του περιγραφή. Η ανάλυση εστιάζει στη δυσμενέστερη και πρακτικά σημαντική περίπτωση της FP32, η οποία αντιπροσωπεύει το πραγματικό περιβάλλον χρήσης σε embedded συστήματα.

4.3.2 Ερμηνεία πειραματικών αποτελεσμάτων

Ο tent map αποτελεί έναν απλό piecewise γραμμικό χαοτικό χάρτη, ο οποίος θεωρητικά εμφανίζει καλή χαοτική συμπεριφορά και συχνά προτείνεται ως εναλλακτική λύση σε σχέση με τον logistic map. Ωστόσο, η ψηφιακή του υλοποίηση παρουσιάζει ιδιαίτερες δυσκολίες.

Το ιστόγραμμα των τιμών κατάστασης του tent map σε FP32 αποκαλύπτει ταχεία κατάρρευση της δυναμικής συμπεριφοράς σε μία ή ελάχιστες τιμές. Το φαινόμενο αυτό υποδηλώνει ότι το σύστημα έχει εισέλθει σε σταθερό σημείο ή πολύ μικρό κύκλο, με αποτέλεσμα την ουσιαστική απώλεια της χαοτικής φύσης του.

Η συμπεριφορά αυτή αποδίδεται στον piecewise γραμμικό χαρακτήρα του tent map, ο οποίος σε συνδυασμό με την πεπερασμένη ακρίβεια της FP32 αριθμητικής οδηγεί σε γρήγορη απώλεια πληροφορίας και περιοδικότητα.

4.4 Ο Chebyshev Map και άλλες προσεγγίσεις

Ο Chebyshev map αποτελεί έναν λιγότερο διαδεδομένο, αλλά θεωρητικά ισχυρό χαοτικό χάρτη, ο οποίος έχει μελετηθεί στη βιβλιογραφία σε εφαρμογές γεννητριών ψευδοτυχαίων αριθμών και κρυπτογραφικών μηχανισμών. Η μαθηματική του διατύπωση βασίζεται σε πολυωνυμικές και τριγωνομετρικές συναρτήσεις και συνδέεται άμεσα με την οικογένεια των πολυωνύμων Chebyshev και τη θεωρία αριθμών. Ο ορισμός του βασίζεται στη Σχέση 9:

$$x(n+1) = \cos\big(k \cdot \cos^{-1}\big(x(n)\big)\big) \quad (9)$$

όπου x(n) είναι η μεταβλητή κατάστασης στο n-οστό διακριτό βήμα και k μία πραγματική παράμετρος ελέγχου της δυναμικής του συστήματος.

Ένα από τα βασικά θεωρητικά πλεονεκτήματα του Chebyshev map είναι η έντονη μη γραμμικότητα και η καλή ιδιότητα μίξης, στοιχεία που μπορούν να οδηγήσουν σε χαοτικές ακολουθίες με ικανοποιητική στατιστική συμπεριφορά. Για τον λόγο αυτό, ο χάρτης έχει προταθεί σε ορισμένες μελέτες ως εναλλακτική λύση σε απλούστερους χαοτικούς μηχανισμούς.

Ωστόσο, η πρακτική του υλοποίηση σε embedded συστήματα παρουσιάζει σημαντικούς περιορισμούς. Ο υπολογισμός του Chebyshev map απαιτεί τη χρήση τριγωνομετρικών συναρτήσεων (όπως arccos και cos), οι οποίες συνεπάγονται αυξημένο υπολογιστικό κόστος, εξάρτηση από βιβλιοθήκες μαθηματικών συναρτήσεων και λιγότερο προβλέψιμο χρόνο εκτέλεσης. Τα χαρακτηριστικά αυτά καθιστούν τον χάρτη λιγότερο κατάλληλο για εφαρμογές πραγματικού χρόνου σε μικροελεγκτές περιορισμένων πόρων, όπως ο STM32H7.

Για τον λόγο αυτό, η παρούσα εργασία εστιάζει σε χαοτικούς χάρτες που βασίζονται αποκλειστικά σε απλές αριθμητικές πράξεις (πολλαπλασιασμούς και αφαιρέσεις), οι οποίοι προσφέρουν καλύτερη ισορροπία μεταξύ θεωρητικής χαοτικής συμπεριφοράς και πρακτικής αποδοτικότητας σε embedded περιβάλλοντα.

4.4.1 Σχόλια ψηφιακής υλοποίησης

Δεν περιλαμβάνονται πειραματικά γραφήματα λόγω αυξημένης υπολογιστικής πολυπλοκότητας και μειωμένης καταλληλότητας για embedded υλοποίηση.

4.5 Υβριδικός Logistic–Tent Map

Ο συνδυασμός logistic και tent map εκμεταλλεύεται τα πλεονεκτήματα και των δύο. Ο logistic map εισάγει έντονη μη γραμμικότητα, ενώ ο tent map προσφέρει σταθερό χάος σε ψηφιακές υλοποιήσεις και πιο ομοιόμορφη κατανομή. Το coupling των δύο μειώνει αισθητά την πιθανότητα να δημιουργηθούν περιοδικοί κύκλοι, ενώ το υπολογιστικό κόστος παραμένει πολύ χαμηλό.

Ο υβριδικός χαοτικός μηχανισμός περιγράφεται από δύο μεταβλητές κατάστασης, x(n) και y(n), οι οποίες λαμβάνουν τιμές στο διάστημα (0,1). Η μεταβλητή x(n) ακολουθεί τη δυναμική του logistic map, ενώ η μεταβλητή y(n) ακολουθεί τη δυναμική του tent map. Ο αρχικός κανόνας εξέλιξης των δύο χαρτών ορίζεται ως:

$$x(n+1)' = r \cdot x(n) \cdot \big(1 - x(n)\big) \quad (10)$$

$$y(n+1)' = \mu \cdot y(n), για y(n) < 0.5 (11)$$

$$y(n+1)' = \mu \cdot \big(1 - y(n)\big), για y(n) \geq 0.5 (12)$$

όπου r και μ είναι οι παράμετροι ελέγχου της δυναμικής των χαρτών. Στην παρούσα εργασία επιλέγονται τιμές r ≈ 3,99 και μ = 2, οι οποίες αντιστοιχούν στη χαοτική περιοχή λειτουργίας των δύο συστημάτων.

Στη συνέχεια εφαρμόζεται μηχανισμός διασύνδεσης (cross-coupling) μεταξύ των δύο καταστάσεων, ώστε η εξέλιξη κάθε μεταβλητής να εξαρτάται και από την κατάσταση του άλλου χάρτη. Ο τελικός κανόνας εξέλιξης του υβριδικού συστήματος δίνεται από:

$$x(n+1) = [x(n+1)' + \alpha \cdot y(n+1)' + \varepsilon(n)] \, mod \, 1 \quad (13)$$

$$y(n+1) = [y(n+1)' + \beta \cdot x(n+1)' + \varepsilon(n)] \, mod \, 1 \quad (14)$$

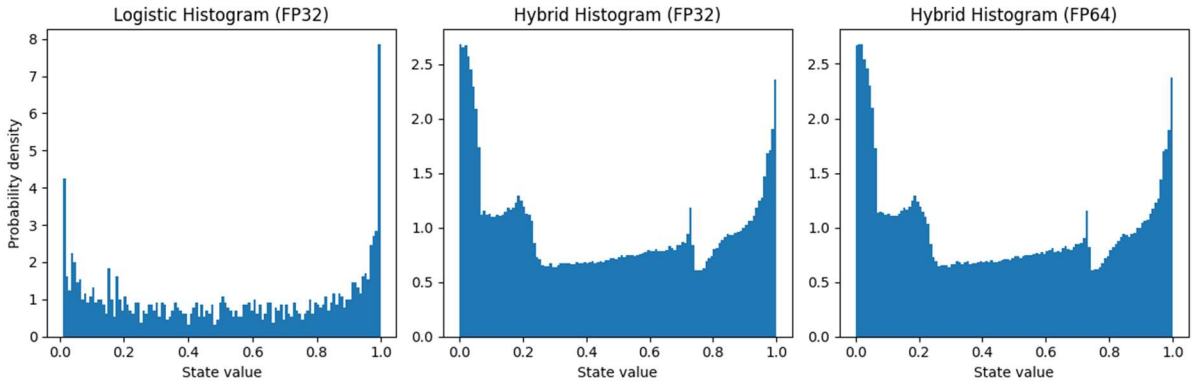
όπου α και β είναι σταθεροί συντελεστές σύζευξης που ρυθμίζουν τον βαθμό αλληλεπίδρασης μεταξύ των δύο χαρτών.

Ο όρος ε(n) αντιπροσωπεύει μία πολύ μικρής τάξης, χρονικά μεταβαλλόμενη διαταραχή, η οποία εξαρτάται από το διακριτό βήμα n και προέρχεται από έναν απλό μετρητή (counter). Η διαταραχή αυτή εισάγεται με σκοπό την αποφυγή αριθμητικών artefacts που οφείλονται στην πεπερασμένη ακρίβεια της αριθμητικής κινητής υποδιαστολής (FP32). Η παρουσία του ε(n) δεν αλλοιώνει τη βασική χαοτική δυναμική του συστήματος, αλλά βελτιώνει τη συμπεριφορά του σε πρακτικές ψηφιακές υλοποιήσεις. Η επιλογή σχετίζεται με την ύπαρξη hardware FPU στον STM32H7, όπου η FP32 αριθμητική εκτελείται ταχύτερα από fixed-point scaling.

Η έξοδος του υβριδικού χαοτικού συστήματος σε κάθε χρονικό βήμα είναι το ζεύγος (x(n), y(n)), το οποίο αποτελεί την πλήρη εσωτερική κατάσταση του συστήματος και χρησιμοποιείται ως βάση για την περαιτέρω ψηφιακή επεξεργασία και παραγωγή ψευδοτυχαίων ακολουθιών, όπως θα παρουσιαστεί στο επόμενο κεφάλαιο.

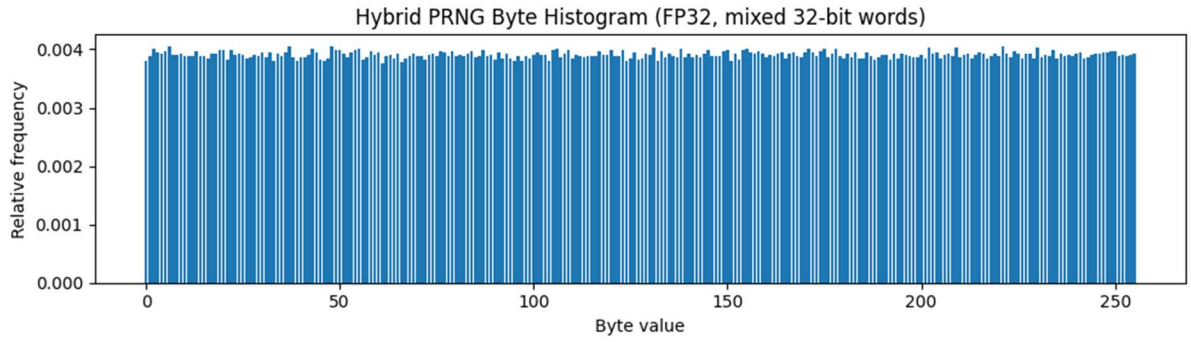
4.5.1 Πειραματική απεικόνιση σε ψηφιακή υλοποίηση

Η δημιουργία του γραφήματος του σχήματος πραγματοποιήθηκε με χρήση προγράμματος σε γλώσσα Python. Ο κώδικας περιέχεται στο Παράρτημα Δ (ch2_hybrid_state_histograms_fp32_fp64.py).



Σχήμα 18 Ιστόγραμμα τιμών κατάστασης του υβριδικού logistic–tent map σε FP32 και FP64

Η δημιουργία του γραφήματος του σχήματος πραγματοποιήθηκε με χρήση προγράμματος σε γλώσσα Python. Ο κώδικας περιέχεται στο Παράρτημα Δ (ch2_hybrid_prng_byte_histogram.py).



Σχήμα 19 Κατανομή συχνοτήτων byte εξόδου από υβριδικό PRNG σε FP32

4.5.2 Ερμηνεία πειραματικών αποτελεσμάτων

Με βάση τις παρατηρήσεις από τη μεμονωμένη χρήση των logistic και tent maps, καθίσταται σαφές ότι κανένας από τους δύο δεν είναι επαρκής ως αυτόνομη λύση σε περιβάλλοντα περιορισμένης αριθμητικής ακρίβειας. Για τον λόγο αυτό υιοθετείται μία υβριδική προσέγγιση, η οποία συνδυάζει τους δύο χαοτικούς μηχανισμούς.

Τα ιστογράμματα της μεταβλητής κατάστασης του υβριδικού σχήματος σε FP32 και FP64 δείχνουν ότι, ακόμη και σε πεπερασμένη ακρίβεια, η κατανομή των τιμών παραμένει αισθητά πιο ομοιόμορφη σε σύγκριση με τον logistic map FP32, χωρίς την κατάρρευση που παρατηρείται στον tent map. Η σύγκριση με την περίπτωση FP64 επιβεβαιώνει ότι τα υπολειπόμενα αριθμητικά artefacts οφείλονται αποκλειστικά στην ακρίβεια και όχι στον ίδιο τον υβριδικό μηχανισμό.

Επιπλέον, η κατανομή συχνοτήτων των byte εξόδου του hybrid PRNG παρουσιάζει σχεδόν ομοιόμορφη συμπεριφορά. Το αποτέλεσμα αυτό υποδεικνύει ότι τα αριθμητικά φαινόμενα της FP32 αναπαράστασης δεν μεταφέρονται στο τελικό keystream, καθιστώντας τον υβριδικό αλγόριθμο κατάλληλο για χρήση ως πηγή κλειδιού σε stream cipher για embedded εφαρμογές.

4.6 Ψηφιακά ζητήματα: από την αναλογική συνέχεια στη διακριτή αναπαράσταση

Τα αναλογικά χαοτικά συστήματα λειτουργούν σε πραγματικούς αριθμούς και έχουν άπειρη ακρίβεια. Στον αντίποδα, τα ψηφιακά συστήματα έχουν πεπερασμένη αναπαράσταση αριθμών, είτε μέσω fixed-point είτε μέσω floating-point μοντέλων [36]. Αυτό σημαίνει ότι το πλήθος των διαφορετικών τιμών που μπορεί να πάρει η μεταβλητή κατάστασης είναι πεπερασμένο και μικρότερο από το θεωρητικό συνεχές. Στην πράξη, αυτό δημιουργεί δύο πιθανά προβλήματα:

1. Περιοδικότητα
2. Συγκέντρωση τιμών

Οι δύο τεχνικές που χρησιμοποιούνται σε αυτή την εργασία για να αποφευχθούν τα παραπάνω είναι:

1. επαναληπτική εφαρμογή πολλών βημάτων του χάρτη σε κάθε παραγωγή εξόδου,
2. συνδυασμός δύο χαρτών (logistic και tent) μέσω coupling.

4.7 Κριτήρια Σύγκρισης Χαοτικών Χαρτών για Embedded PRNG

Η επιλογή ενός χαοτικού χάρτη ως βάση για γεννήτρια ψευδοτυχαίων αριθμών σε embedded περιβάλλον δεν μπορεί να στηρίζεται αποκλειστικά σε θεωρητικά μαθηματικά χαρακτηριστικά. Απαιτείται η συνεκτίμηση τόσο της χαοτικής συμπεριφοράς όσο και των πρακτικών περιορισμών της

ψηφιακής υλοποίησης. Στο πλαίσιο της παρούσας εργασίας, τα κριτήρια σύγκρισης των εξεταζόμενων χαοτικών χαρτών ορίζονται ως εξής:

- Ευαισθησία στις αρχικές συνθήκες. Ο χάρτης θα πρέπει να εμφανίζει έντονη απόκλιση τροχιών ακόμη και για ελάχιστες μεταβολές της αρχικής τιμής, ώστε μικρές αλλαγές στο seed να οδηγούν σε πλήρως διαφορετικές ακολουθίες.
- Διατήρηση χαοτικής συμπεριφοράς σε πεπερασμένη ακρίβεια. Ιδιαίτερη σημασία δίνεται στη συμπεριφορά του χάρτη σε αριθμητική FP32, καθώς αυτή αποτελεί το πραγματικό περιβάλλον υλοποίησης. Η εμφάνιση περιοδικότητας ή κατάρρευσης της δυναμικής θεωρείται κρίσιμη αδυναμία.
- Ομοιομορφία κατανομής τιμών. Οι παραγόμενες τιμές κατάστασης ή εξόδου θα πρέπει να κατανέμονται όσο το δυνατόν πιο ομοιόμορφα, χωρίς έντονες συγκεντρώσεις ή στατιστικές ανισορροπίες που θα μπορούσαν να οδηγήσουν σε προβλέψιμα μοτίβα.
- Καταλληλότητα για παραγωγή keystream. Ο χάρτης θα πρέπει να μπορεί να χρησιμοποιηθεί ως αξιόπιστη πηγή keystream για stream cipher, χωρίς να απαιτεί πολύπλοκους μετασχηματισμούς ή βαριά post-processing στάδια.
- Υπολογιστικό κόστος και προβλεψιμότητα χρόνου εκτέλεσης. Η υλοποίηση πρέπει να βασίζεται σε απλές αριθμητικές πράξεις, με σταθερό και προβλέψιμο χρόνο εκτέλεσης, ώστε να είναι συμβατή με απαιτήσεις πραγματικού χρόνου.
- Καταλληλότητα για embedded υλοποίηση. Συνεκτιμώνται παράγοντες όπως η ανάγκη για βιβλιοθήκες μαθηματικών συναρτήσεων, η χρήση FPU και η συνολική πολυπλοκότητα του αλγορίθμου.

Τα παραπάνω κριτήρια συνδέονται άμεσα με τα πειραματικά αποτελέσματα που παρουσιάστηκαν στις προηγούμενες ενότητες και αποτελούν τη βάση για τη συγκριτική αποτίμηση και την τελική επιλογή του χαοτικού μηχανισμού που χρησιμοποιείται στην παρούσα εργασία.

4.8 Συγκριτική Αποτίμηση Χαοτικών Χαρτών

Με βάση τα κριτήρια της προηγούμενης ενότητας, πραγματοποιείται συγκριτική αποτίμηση των χαοτικών χαρτών που εξετάστηκαν. Η αξιολόγηση δεν αποσκοπεί σε απόλυτη ιεράρχηση θεωρητικής ασφάλειας, αλλά στη σύγκριση της καταλληλότητας κάθε προσέγγισης για ψηφιακή υλοποίηση σε embedded περιβάλλον.

Logistic map:

- Υψηλή θεωρητική ευαισθησία στις αρχικές συνθήκες
- Απλή μαθηματική μορφή και χαμηλό υπολογιστικό κόστος
- Εμφανίζει αριθμητικά artefacts και συγκέντρωση τιμών σε FP32

Tent map:

- Απλή υλοποίηση με γραμμικές πράξεις
- Καλή θεωρητική ομοιομορφία κατανομής
- Ταχεία κατάρρευση χαοτικής συμπεριφοράς σε FP32

Chebyshev map:

- Ισχυρή θεωρητική μη γραμμικότητα και ιδιότητες μίξης
- Χρήση σε εξειδικευμένες κρυπτογραφικές μελέτες
- Αυξημένο υπολογιστικό κόστος και μειωμένη καταλληλότητα για real-time embedded υλοποίηση

Hybrid logistic–tent map:

- Συνδυασμός πλεονεκτημάτων logistic και tent map
- Βελτιωμένη συμπεριφορά σε FP32
- Καλή ομοιομορφία keystream και χαμηλό υπολογιστικό κόστος

4.9 Έλεγχοι – Συμπεράσματα

Όπως αναφέρθηκε ήδη σε προηγούμενα κεφάλαια, η παρούσα εργασία δεν επιχειρεί πλήρη στατιστική πιστοποίηση των χαοτικών χαρτών, αλλά ένα ελαφρύ, ενδεικτικό έλεγχο τυχαιότητας. Τα αποτελέσματα ευθυγραμμίζονται με τη βιβλιογραφία: οι πολύ γνωστοί και μελετημένοι logistic και tent χάρτες παρουσιάζουν σημαντικές αδυναμίες. Αποτυγχάνουν στα περισσότερα tests, ενώ ένας συζευγμένος logistic–tent χάρτης εμφανίζει σαφώς βελτιωμένη συμπεριφορά δίνοντας ενδείξεις ότι μπορεί να περάσει αρκετά από τα βασικά NIST SP800-22 tests. Τα ευρήματα αυτά δεν αποτελούν αυστηρή απόδειξη, αλλά παρέχουν επαρκές κίνητρο ώστε η συνέχεια της εργασίας να εστιάσει στην πρακτική υλοποίηση του υβριδικού χάρτη σε embedded περιβάλλον χαμηλών πόρων. Στον Πίνακας 7 ακολουθεί αποτυπώνονται με εποπτικό τρόπο τα συμπεράσματα.

Πίνακας 7 Συνοπτική ποιοτική αξιολόγηση των logistic, tent και συζευγμένου logistic–tent χαρτών ως προς βασικά tests του NIST SP800-22.

NIST Test (SP800-22)	Logistic	Tent	Citations	Logistic + Tent
Συχνότητας (Frequency)	Fail	Fail	[50],[51],[52]	Pass
Συχνότητας σε κομμάτι (Block Frequency)	Fail	Fail	[50],[52]	Pass
Δοκιμή εκτελέσεων (Runs)	Fail	Fail	[51],[52]	Pass
Μεγαλύτερης σειράς ίδιου ψηφίου (Longest Run of Ones)	Fail	Fail	[50]	Pass
Κατάταξη δυαδικού πίνακα (Binary Matrix Rank)	Fail	Fail	[51]	Fail
Fourier (FFT)	Fail	Fail	[50],[52]	Pass
Αντιστοίχιση μη επικαλυπτόμενων προτύπων (Non-overlapping Templates)	Fail	Fail	[51]	Fail
Αντιστοίχιση επικαλυπτόμενων προτύπων (Overlapping Templates)	Fail	Fail	[51]	Fail
Καθολικό στατιστικό (Maurer's Universal)	Fail	Fail	[50]	Fail
Γραμμική πολυπλοκότητα (Linear Complexity)	Fail	Fail	[51]	Fail
Σειριακή δοκιμή (Serial)	Fail	Fail	[52]	Pass
Προσεγγιστική εντροπία (Approximate Entropy)	Fail	Fail	[50]	Pass
Συσσωρευτικών αθροισμάτων (Cumulative Sums)	Inconsistent	Inconsistent	[51]	Pass
Τυχαίων περιηγήσεων (Random Excursions)	Inconsistent	Inconsistent	[52]	Pass
Τυχαίων περιηγήσεων με παραλλαγή (Random Excursions Variant)	Inconsistent	Inconsistent	[52]	Pass

4.10 Τεκμηριωμένη Επιλογή Τελικού Αλγορίθμου

Η συγκριτική αποτίμηση καταδεικνύει ότι καμία μεμονωμένη χαοτική προσέγγιση δεν καλύπτει πλήρως τις απαιτήσεις ενός embedded PRNG όταν λαμβάνονται υπόψη οι περιορισμοί της ψηφιακής υλοποίησης. Ο logistic map, παρότι θεωρητικά ισχυρός και υπολογιστικά αποδοτικός, εμφανίζει αριθμητικές αδυναμίες σε FP32. Ο tent map, αν και απλός, παρουσιάζει ταχεία απώλεια χαοτικής συμπεριφοράς σε περιβάλλον πεπερασμένης ακρίβειας. Ο Chebyshev map προσφέρει ενδιαφέρουσες θεωρητικές ιδιότητες, αλλά το υπολογιστικό του κόστος τον καθιστά λιγότερο κατάλληλο για εφαρμογές πραγματικού χρόνου σε μικροελεγκτές.

Αντίθετα, ο υβριδικός logistic–tent map επιτυγχάνει ισορροπία μεταξύ θεωρητικής χαοτικής συμπεριφοράς και πρακτικής αποδοτικότητας. Ο συνδυασμός δύο διαφορετικών χαοτικών μηχανισμών μειώνει την πιθανότητα εμφάνισης περιοδικότητας και περιορίζει τις αριθμητικές αδυναμίες της FP32 αναπαράστασης, ενώ το συνολικό υπολογιστικό κόστος παραμένει χαμηλό και προβλέψιμο.

Με βάση τα παραπάνω, ο υβριδικός logistic–tent map επιλέγεται ως ο καταλληλότερος χαοτικός μηχανισμός για την υλοποίηση της γεννήτριας ψευδοτυχαίων αριθμών που χρησιμοποιείται στην παρούσα εργασία. Ο στόχος δεν είναι η αντικατάσταση κλασικών αλγορίθμων κρυπτογράφησης, αλλά η επίτευξη επαρκούς απόκρυψης δεδομένων με ελάχιστο υπολογιστικό κόστος σε περιορισμένο υλικό.

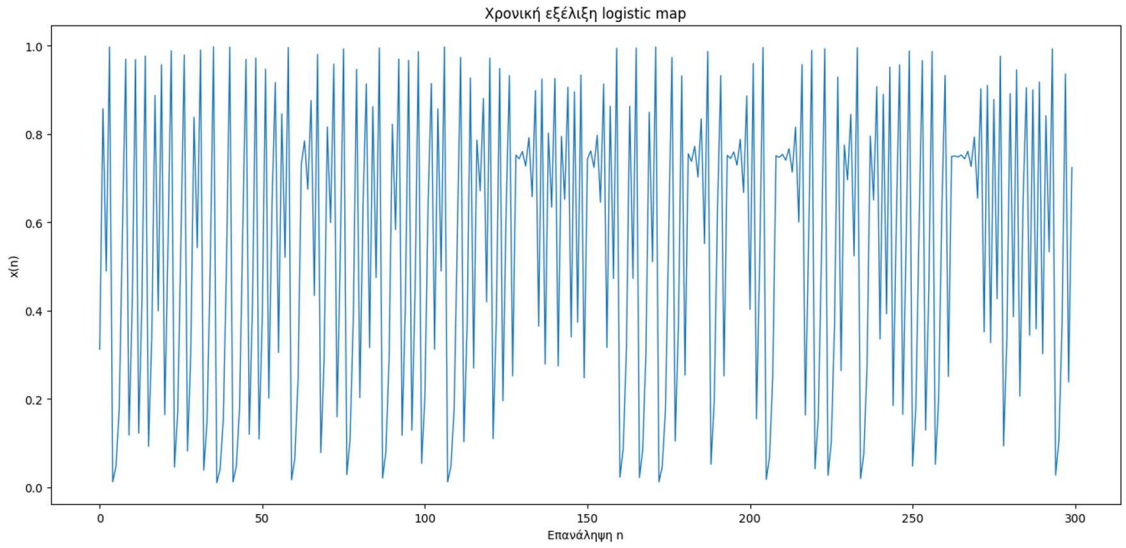
Κεφάλαιο 5ο: Παραγωγή Ψευδοτυχαίων Ακολουθιών και Χαοτική Κρυπτογράφηση

Η αξιοποίηση χαοτικών συστημάτων για την παραγωγή ψευδοτυχαίων ακολουθιών αποτελεί έναν ιδιαίτερα ελκυστικό μηχανισμό σε εφαρμογές όπου απαιτείται υψηλή εντροπία, μικρό υπολογιστικό κόστος και δυνατότητα λειτουργίας σε πραγματικό χρόνο. Στο κεφάλαιο αυτό εξετάζεται πώς οι χαοτικοί χάρτες μπορούν να χρησιμοποιηθούν ως βάση για μια γεννήτρια ψευδοτυχαίων αριθμών (PRNG) και πώς το παραγόμενο υλικό μπορεί να αξιοποιηθεί στην υλοποίηση κρυπτογράφησης τύπου stream cipher.

5.1 Χαοτικά συστήματα ως PRNG

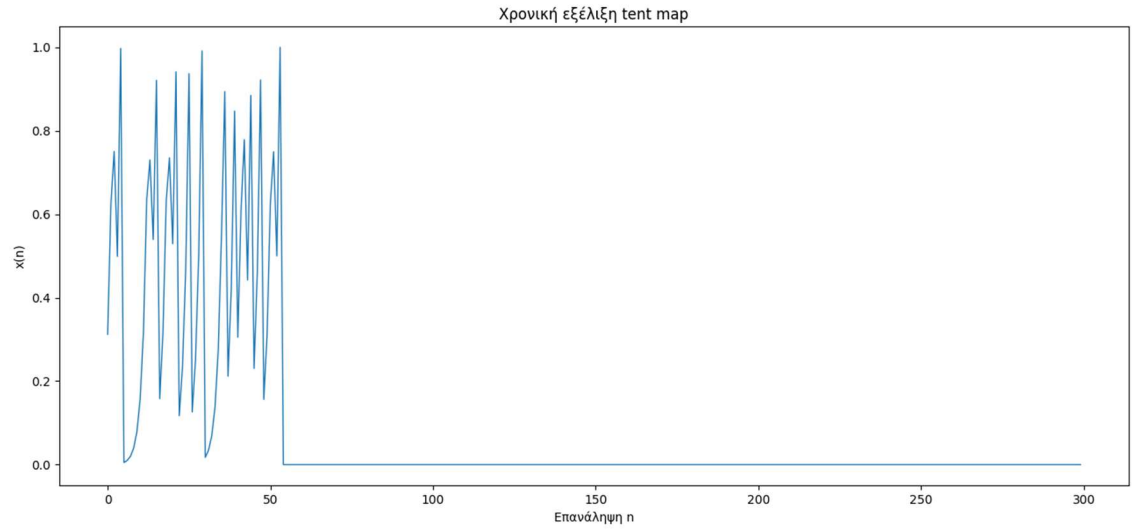
Όπως αναφέρθηκε στα προηγούμενα κεφάλαια, τα χαοτικά συστήματα παράγουν ακολουθίες που εμφανίζουν υψηλή ευαισθησία στις αρχικές συνθήκες και μη γραμμική εξέλιξη. Οι δύο αυτές ιδιότητες συνιστούν βασικές προϋποθέσεις για την κατασκευή ενός αξιόπιστου PRNG [35], [36]. Το παραγόμενο bitstream πρέπει να μεταβάλλεται δραματικά ακόμη και για ελάχιστες τροποποιήσεις της αρχικής τιμής [35], [36], [40] (seed).

Πριν εξεταστεί η διαδικασία παραγωγής του bitstream, είναι χρήσιμο να αποτυπωθεί γραφικά η συμπεριφορά μιας χαοτικής ακολουθίας ως προς την πάροδο των επαναλήψεων. Ενδεικτικά, η εξέλιξη της μεταβλητής κατάστασης του logistic map αποδίδει μια χαρακτηριστική εικόνα της συνολικής μεταβλητότητας του συστήματος, όπως φαίνεται στο Σχήμα 20.



Σχήμα 20 Εξέλιξη της μεταβλητής κατάστασης του logistic map σε διαδοχικές επαναλήψεις

Αντίστοιχη απεικόνιση μπορεί να πραγματοποιηθεί και για τον tent map, όπως παρουσιάζεται στο Σχήμα 21.



Σχήμα 21 Εξέλιξη της μεταβλητής κατάστασης του tent map σε διαδοχικές επαναλήψεις

Παρότι οι δύο χάρτες παράγουν τροχιές διαφορετικού ύφους, και οι δύο διαθέτουν τα απαιτούμενα χαρακτηριστικά για χρήση ως πηγή ψευδοτυχαίων ακολουθιών [35], [37].

Τα παραπάνω σχήματα έχουν ενδεικτικό χαρακτήρα και αποσκοπούν στην ποιοτική απεικόνιση της χαοτικής συμπεριφοράς των χαρτών. Οι παράμετροι και οι διαδικασίες που χρησιμοποιούνται στην τελική υλοποίηση του συστήματος παρουσιάζονται αναλυτικά στα επόμενα κεφάλαια.

5.2 Μετατροπή της χαοτικής κατάστασης σε 32-bit τιμές

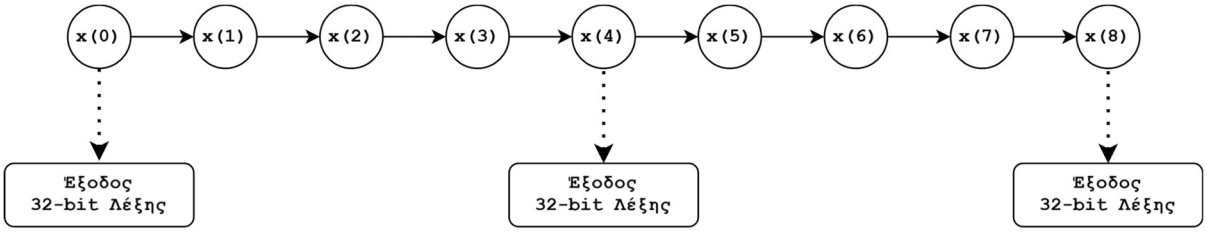
Για να λειτουργήσει ένας χαοτικός αλγόριθμος ως PRNG, το κρίσιμο βήμα είναι η μετάβαση από τη συνεχή χαοτική κατάσταση (η οποία βρίσκεται στο διάστημα (0,1)) σε μια διακριτή δυαδική αναπαράσταση. Στη συγκεκριμένη εργασία χρησιμοποιείται η αναπαράσταση FP32 (float), και έτσι το περιεχόμενο της μεταβλητής κατάστασης μπορεί να μετατραπεί απευθείας σε 32-bit λέξη ερμηνεύοντας ξανά τα bit της. Η διαδικασία αυτή απεικονίζεται στα ακόλουθα διαγράμματα/σχήματα (Σχήμα 22).

Σχήμα 22 Ενδεικτική διαδικασία μετατροπής χαοτικής κατάστασης σε δυαδική αναπαράσταση

Η επιλογή αναπαράστασης floating-point αντί fixed-point οφείλεται στην ύπαρξη μονάδας κινητής υποδιαστολής (FPU) στον STM32H7, η οποία καθιστά το κόστος πράξεων FP32 πρακτικά αμελητέο.

5.3 Διαδικασία παραγωγής keystream

Μετά την παραγωγή κάθε 32-bit λέξης, ο αλγόριθμος δεν παράγει άμεσα την επόμενη έξοδο, αλλά εκτελεί έναν προκαθορισμένο αριθμό επιπλέον επαναλήψεων της χαοτικής συνάρτησης. Το διάστημα αυτό, το οποίο μπορεί ενδεικτικά να κυμαίνεται από 4 έως 8 επαναλήψεις, αυξάνει τη χρονική απόσταση μεταξύ των καταστάσεων που χρησιμοποιούνται για την παραγωγή εξόδου. Με τον τρόπο αυτό ενισχύεται η μη προβλεψιμότητα της παραγόμενης ακολουθίας και μειώνεται η πιθανότητα εμφάνισης περιοδικότητας. Η συνολική ροή της διαδικασίας παρουσιάζεται ενδεικτικά στο Σχήμα 23.



Σχήμα 23 Ενδεικτική ροή παραγωγής λέξεων από χαοτικό PRNG με ενδιάμεσα βήματα επανάληψης (skipping) μεταξύ διαδοχικών εξόδων

Στο συγκεκριμένο σύστημα αξιοποιούνται:

- δύο χαοτικοί χάρτες (logistic και tent)
- μηχανισμός cross-coupling μεταξύ των καταστάσεών τους
- μικρό ποσοστό «θορύβου» προερχόμενο από μετρητή (counter), με σκοπό την αποφυγή παρενεργειών που οφείλονται στην πεπερασμένη ακρίβεια της αριθμητικής κινητής υποδιαστολής 32-bit (FP32), οι οποίες είναι γνωστές ως αριθμητικά τεχνουργήματα (artefacts) [36]
- στάδιο μη γραμμικής μίξης (bit mixing) πριν από την τελική παραγωγή της 32-bit τιμής

Η συνδυαστική αυτή προσέγγιση οδηγεί στη δημιουργία μίας γεννήτριας ψευδοτυχαίων αριθμών που παραμένει ελαφριά και ταχεία, ενώ παράλληλα παρουσιάζει στατιστική συμπεριφορά επαρκή για τις ανάγκες κρυπτογράφησης τηλεμετρικών δεδομένων [38], [39]. Αξίζει να σημειωθεί ότι η διαδικασία αυτή δεν μεταβάλλει τον ίδιο τον χαοτικό χάρτη, αλλά αποκλειστικά τον ρυθμό με τον οποίο η εσωτερική κατάσταση δειγματοληπτείται για την παραγωγή εξόδου.

5.4 Stream cipher μέσω XOR

Σύμφωνα με τη θεωρία της μυστικότητας του Shannon, η ασφάλεια ενός stream cipher εξαρτάται άμεσα από την τυχαιότητα της ροής κλειδιού [41], [42], [43]. Το PRNG χρησιμοποιείται στη συνέχεια ως πηγή ροής κλειδιού (keystream) για την υλοποίηση ενός stream cipher. Η βασική ιδέα είναι ότι κάθε byte της ροής κλειδιού συνδυάζεται με το αντίστοιχο byte των αρχικών δεδομένων (plaintext) μέσω της λογικής πράξης αποκλειστικής διάζευξης (XOR), σύμφωνα με τη (Σχέση 15) [41]:

$$\text{ciphertext}[i] = \text{plaintext}[i] \oplus \text{keystream}[i] (15)$$

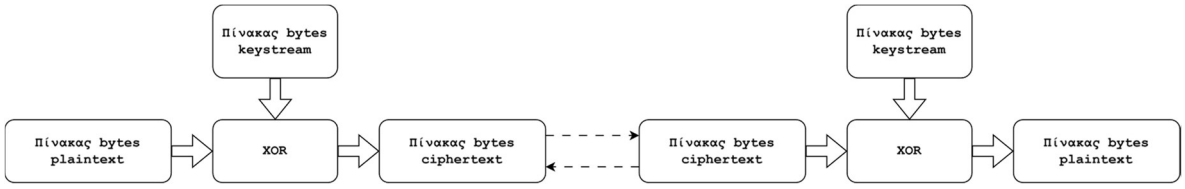
όπου:

- ciphertext: ο εξερχόμενος κρυπτογραφημένος πίνακας bytes
- plaintext: ο εισερχόμενος πίνακας bytes
- keystream: ακολουθία bytes που παράγει το PRNG
- i: ο δείκτης του byte μέσα στον πίνακα

Η διαδικασία αυτή είναι πλήρως αναστρέψιμη χωρίς πρόσθετο υπολογιστικό κόστος, καθώς η ίδια πράξη XOR χρησιμοποιείται και για την αποκρυπτογράφηση (Σχέση 16):

$$\text{plaintext}[i] = \text{ciphertext}[i] \oplus \text{keystream}[i] (16)$$

Συνεπώς, η ίδια ρουτίνα μπορεί να χρησιμοποιηθεί τόσο για κρυπτογράφηση όσο και για αποκρυπτογράφηση, υπό την προϋπόθεση ότι η ροή κλειδιού είναι απολύτως ίδια και στις δύο πλευρές της επικοινωνίας. Η συνολική ροή της διαδικασίας απεικονίζεται ενδεικτικά στο ακόλουθο διάγραμμα Σχήμα 24.



Σχήμα 24 Ενδεικτική ροή κρυπτογράφησης stream cipher μέσω πράξης XOR

Το βασικό πλεονέκτημα της μεθόδου είναι ότι επιτρέπει κρυπτογράφηση σε πραγματικό χρόνο, χωρίς την ανάγκη μεγάλων περιοχών μνήμης (buffers) και χωρίς την απαίτηση υπολογιστικά απαιτητικών μαθηματικών πράξεων.

5.5 Συμβατότητα με συστήματα πραγματικού χρόνου

Ο μικροελεγκτής STM32H7 διαθέτει μονάδα κινητής υποδιαστολής (FPU) που εκτελεί πράξεις FP32 σε έναν κύκλο, γεγονός που καθιστά τις επαναλήψεις των logistic και tent maps ιδιαίτερα αποδοτικές από πλευράς χρόνου εκτέλεσης. Ως αποτέλεσμα, ο προτεινόμενος αλγόριθμος μπορεί να ενσωματωθεί χωρίς δυσκολία σε σενάρια πραγματικού χρόνου, όπως:

- επεξεργασία δεδομένων ADC με υψηλό ρυθμό ανανέωσης
- εκτέλεση εντός callbacks διεπαφών USB CDC
- κρυπτογράφηση δεδομένων πριν από την αποστολή πακέτων UDP και TCP
- χρήση σε payloads LoRa, όπου κάθε byte και κάθε κύκλος επεξεργασίας είναι κρίσιμος

Η ποσοτική αξιολόγηση της χρονικής επιβάρυνσης που εισάγει το επίπεδο κρυπτογράφησης παρουσιάζεται στο κεφάλαιο ανάλυσης του λογισμικού (firmware), μέσω μετρήσεων επεξεργασίας buffers συγκεκριμένου μεγέθους.

5.6 Σύνοψη κεφαλαίου

Η προσέγγιση που παρουσιάστηκε στο παρόν κεφάλαιο συνδυάζει χαοτικά συστήματα με απλούς μηχανισμούς μίξης για την παραγωγή ροής κλειδιού και την υλοποίηση stream cipher σε περιβάλλοντα περιορισμένων πόρων. Τα βασικά πλεονεκτήματα της μεθόδου συνοψίζονται στα εξής:

- ελάχιστο υπολογιστικό κόστος,
- υψηλή εντροπία σε ψηφιακές υλοποιήσεις,
- ικανοποιητική στατιστική συμπεριφορά για εφαρμογές embedded κρυπτογράφησης,
- απλή και κατανοητή αρχιτεκτονική [44],[45],[46],[47].

Για τους σκοπούς της παρούσας διπλωματικής εργασίας και για εφαρμογές τηλεμετρίας χωρίς αυστηρές απαιτήσεις υψηλού επιπέδου ασφάλειας, η μέθοδος αυτή κρίνεται επαρκής και πρακτικά εφαρμόσιμη. Ζητήματα που αφορούν περαιτέρω ενίσχυση της ασφάλειας και επέκταση των μηχανισμών προστασίας εξετάζονται σε επόμενα κεφάλαια και στα συνολικά συμπεράσματα της εργασίας.

Κεφάλαιο 6ο: Τεχνικά Χαρακτηριστικά LoRa

6.1 Τι είναι το LoRa

Το LoRa (Long Range) είναι μια φυσική διαμόρφωση RF που αναπτύχθηκε από τη Semtech. Δεν είναι πρωτόκολλο δικτύου· αυτό είναι το LoRaWAN. Το LoRa είναι το PHY layer που επιτρέπει πολύ μεγάλη εμβέλεια με πολύ χαμηλή κατανάλωση.

6.2 Αρχή λειτουργίας – Chirp Spread Spectrum (CSS)

Το LoRa βασίζεται στη διαμόρφωση Chirp Spread Spectrum (CSS), όπου το σήμα αποτελείται από "chirps" που αυξάνουν ή μειώνουν γραμμικά τη συχνότητά τους.

Αυτό προσφέρει:

- Υψηλή ανοχή σε θόρυβο (robustness)
- Μεγάλη εμβέλεια λόγω πολύ χαμηλού απαιτούμενου SNR
- Αντοχή σε παρεμβολές
- Ευκολία αποκωδικοποίησης ακόμα και κάτω από το επίπεδο θορύβου

6.3 Διαμόρφωση LoRa – Τεχνικές Παράμετροι

- Spreading Factor (SF)
- Τιμές: SF7 έως SF12
- Μεγαλύτερο SF → μεγαλύτερη ευαισθησία, μεγαλύτερη εμβέλεια, μικρότερο data rate.
- Bandwidth (BW)
- Συνήθεις τιμές: 125 kHz, 250 kHz, 500 kHz
- Μικρότερο BW → μεγαλύτερη ευαισθησία, μικρότερο data rate.
- Coding Rate (CR)
- CR = 4/5, 4/6, 4/7, 4/8
- Μεγαλύτερο CR → καλύτερη διόρθωση λαθών, μικρότερο data rate.
- Data Rates
- Από ~0.3 kbps (SF12, BW125)
- Έως ~50 kbps (SF7, BW500)

6.4 Συχνότητες λειτουργίας (ISM Bands)

Το LoRa χρησιμοποιεί μη αδειοδοτημένες ISM μπάντες. Οι συχνότητες εξαρτώνται από την περιοχή.

Πίνακας 8 Συχνότητες λειτουργίας LoRa ανά περιοχή

Περιοχή	Συχνότητα
Ευρώπη	863 - 870 MHz
ΗΠΑ	902 - 928 MHz
Ασία	433, 920 – 925 MHz

Στην Ευρώπη ισχύουν και περιορισμοί duty cycle (π.χ. 1% σε πολλά sub-bands).

6.5 Ισχύς εκπομπής

- Τυπική ισχύς: +14 dBm (25 mW) στην Ευρώπη
- Μέγιστο επιτρεπόμενο EIRP: +16 έως +20 dBm, ανάλογα με το sub-band και τους κανονισμούς

- Στις ΗΠΑ επιτρέπεται υψηλότερη ισχύς λόγω διαφορετικών κανόνων FCC

6.6 Εμβέλεια

Η εμβέλεια εξαρτάται από SF, BW, ισχύ, κεραία και περιβάλλον:

- Αστικό περιβάλλον: 2–5 km
- Αγροτικό/ανοιχτό: 10–15 km
- Ιδανικές συνθήκες: έως 20+ km

6.7 LoRa vs LoRaWAN

Συχνά συγχέονται οι δύο όροι ή χρησιμοποιούνται ως ισοδύναμες τεχνολογίες. Υπάρχει μια θεμελιακή διαφορά που κάνει την σύγκριση αδύνατη. Το LoRa είναι φυσική διαμόρφωση RF, είναι το physical layer ενώ το LoRaWAN είναι λογισμικό, ένα πρωτόκολλο. Συνεργάζονται όπως το Ethernet και το TCP/IP. Ο πίνακας που ακολουθεί παρουσιάζει τις διαφορές μεταξύ LoRa και LoRaWAN.

Χαρακτηριστικό	LoRa	LoRaWAN
Ρόλος	Modulation (PHY)	MAC, network, security
Ανάπτυξη	Semtech	LoRa Alliance
Ασφάλεια	Δεν ορίζει	AES-128 end-to-end
Δίκτυο	Point-to-point ή gateway	Πλήρες star-of-stars δίκτυο

6.8 Πλεονεκτήματα LoRa

- Πολύ μεγάλη εμβέλεια
- Εξαιρετικά χαμηλή κατανάλωση
- Λειτουργία σε μη αδειοδοτημένες συχνότητες
- Υψηλή ανοχή σε θόρυβο
- Ιδανικό για IoT αισθητήρες

6.9 Μειονεκτήματα LoRa

- Χαμηλό data rate
- Περιορισμοί duty cycle στην Ευρώπη
- Όχι κατάλληλο για real-time εφαρμογές υψηλού bandwidth

Κεφάλαιο 7ο: Λογισμικό (Firmware)

Το παρόν κεφάλαιο περιγράφει την αρχιτεκτονική και τη λειτουργική οργάνωση του ενσωματωμένου λογισμικού (firmware) που αναπτύχθηκε για την παρούσα διπλωματική εργασία. Το firmware αποτελεί τον συνδετικό κρίκο μεταξύ του υλικού συστήματος, των μηχανισμών κρυπτογράφησης και των διεπαφών επικοινωνίας, υλοποιώντας μια ντετερμινιστική και επεκτάσιμη ροή δεδομένων πραγματικού χρόνου.

Η ανάλυση πραγματοποιείται σε επίπεδο αρχιτεκτονικής και ροής λειτουργίας, χωρίς αναφορά σε λεπτομέρειες υλοποίησης συναρτήσεων. Εντούτοις ο πλήρης πηγαίος κώδικας αποτελεί μέρος των παραδοτέων και κατά συνέπεια συνοδεύει την τεκμηρίωση της εργασίας.

7.1 Ρύθμιση περιφερειακών και οδηγοί με την χρήση STM32CubeMX

Οι μικροελεγκτές ARM Cortex-M, όπως ο STM32H735, είναι σύνθετα συστήματα σε ολοκληρωμένη μορφή (SoC, System on Chip) με πλήθος περιφερειακών και δεκάδες συνδυασμούς χρήσης τους. Για να διευκολύνουν τους μηχανικούς οι κατασκευαστές παρέχουν εργαλεία λογισμικού, ώστε μέσα από ένα γραφικό περιβάλλον φιλικό προς τον χρήστη να μπορεί να γίνει η παραμετροποίηση ταχύτερα, ευκολότερα και ασφαλέστερα με πολύ λιγότερα λάθη. Ένα τέτοιο εργαλείο είναι το STM32CubeMX (https://www.st.com/en/development-tools/stm32cubemx.html). Εκτός από την παραμετροποίηση μπορεί να δημιουργήσει κώδικα για την οδήγηση των περιφερειακών (drivers) και να ενσωματώσει υποσυστήματα που χρησιμοποιούν τους drivers.

Είναι διαθέσιμες δύο τεχνικές για τους drivers, HAL (Hardware Abstraction Layer) ή LL (Low Level). Η πρώτη επιτρέπει στον προγραμματιστή να την χρησιμοποιήσει ακόμα και αν δεν γνωρίζει τις λεπτομέρειες της εσωτερικής λειτουργίας του μικροελεγκτή, να χρησιμοποιήσει μελλοντικά άλλο μικροελεγκτή με ελάχιστες αλλαγές στον κώδικα με το τίμημα της πιο αργής εκτέλεσης. Ο δεύτερος όπως μπορεί να γίνει αντιληπτό επιτυγχάνει ταχύτερη εκτέλεση αλλά απαιτεί βαθύτερη γνώση του μικροελεγκτή και η αλλαγή του μπορεί να οδηγήσει σε εκτεταμένο ανασχεδιασμό της εφαρμογής. Στην παρούσα εφαρμογή έχει επιλεγεί το HAL.

Εκτός από τους drivers το εργαλείο μπορεί να ενσωματώσει και ολόκληρα υποσυστήματα. Δύο από αυτά αξιοποιούνται από την εφαρμογή, (α) το LWIP (Light Weight IP) και (β) USB Device.

7.1.1 LWIP

Είναι μια ανοιχτού κώδικα υλοποίηση των πρωτοκόλλων που χρησιμοποιούνται σε δίκτυα Ethernet αλλά και όχι μόνο. Συνήθως ονομάζονται όλα μαζί σαν TCP/IP stack από τα ονόματα των δύο θεμελιακών πρωτοκόλλων, TCP (Transmission Control Protocol) και IP (Internet Protocol). Η εφαρμογή χρησιμοποιεί δύο από τα πρωτόκολλα του, TCP και UDP (User Datagram Protocol).

7.1.2 USB Device

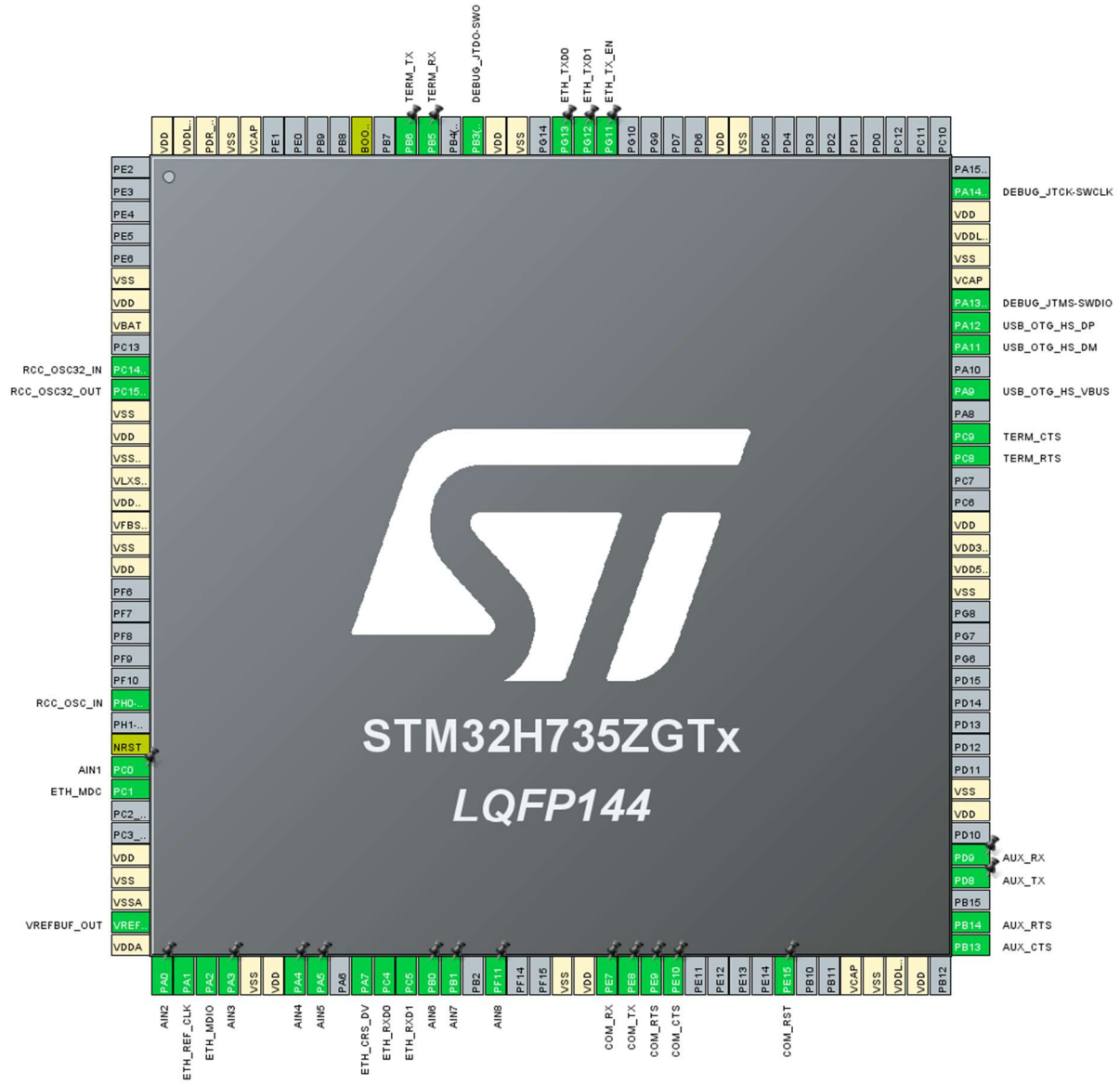
Κάθε συσκευή που αναλαμβάνει τον ρόλο του Device (2.2.3) μπορεί να υλοποιήσει μια ή περισσότερες από τις ακόλουθες κλάσεις.

- Audio Device Class (ADC)
- Communication Device Class (CDC)
- Download Firmware Update Class (DFU)
- Human Interface Device Class (HID)
- Custom Human Interface Device Class (CHID)

- Mass Storage Class

Η εφαρμογή υλοποιεί την κλάση CDC (Communication Device Class) και λειτουργεί εικονική σειριακή θύρα (Virtual Port Com). Η μόνη κατάλληλη ως εναλλακτική είναι η CHID αλλά προϋποθέτει την ανάπτυξη driver για περιβάλλον με το οποίο θα λειτουργήσει και αυτό είναι εκτός του σκοπού της εργασίας αυτής. Επιπλέον η STMicroelectronics παρέχει driver για Windows που αναγνωρίζει οποιαδήποτε συσκευή με firmware που χρησιμοποιεί το USB Device υποσύστημα της.

Στο ακόλουθο σχήμα απεικονίζεται ο μικροελεγκτής και η αντιστοίχιση των σημάτων των ενεργοποιημένων περιφερειακών στους ακροδέκτες εισόδου/εξόδου (I/O).



Σχήμα 25 Αντιστοίχιση των σημάτων των περιφερειακών στους ακροδέκτες εισόδου/εξόδου (I/O)

Συγκεκριμένα ο πίνακας που ακολουθεί περιέχει τις αντιστοιχίες περιφερειακό → σήμα → ακροδέκτης IO.

Πίνακας 9 Αντιστοίχιση των σημάτων των περιφερειακών στους ακροδέκτες εισόδου/εξόδου (I/O)

Περιφερειακό	Λειτουργία	Σήμα	Ακροδέκτης
ADC1	IN2 Single-ended	ADC1_INP2	PF11

ADC1	IN5 Single-ended	ADC1_INP5	PB1
ADC1	IN9 Single-ended	ADC1_INP9	PB0
ADC1	IN10 Single-ended	ADC1_INP10	PC0
ADC1	IN15 Single-ended	ADC1_INP15	PA3
ADC1	IN16 Single-ended	ADC1_INP16	PA0
ADC1	IN18 Single-ended	ADC1_INP18	PA4
ADC1	Single-ended	ADC1_INP19	PA5
ADC3	Vbat Channel	ADC3_Vbat_Input	VP_ADC3_Vbat_Input
ADC3	Temperature Sensor Channel	ADC3_TempSens_Input	VP_ADC3_TempSens_Input
ADC3	Vrefint Channel	ADC3_Vref_Input	VP_ADC3_Vref_Input
DEBUG	Trace Asynchronous Sw	DEBUG_JTMS-SWDIO	PA13(JTMS/SWDIO)
DEBUG	Trace Asynchronous Sw	DEBUG_JTCK-SWCLK	PA14(JTCK/SWCLK)
DEBUG	Trace Asynchronous Sw	DEBUG_JTDO-SWO	PB3(JTDO/TRACESWO)
ETH	RMII	ETH_CRS_DV	PA7
ETH	RMII	ETH_MDC	PC1
ETH	RMII	ETH_MDIO	PA2
ETH	RMII	ETH_REF_CLK	PA1
ETH	RMII	ETH_RXD0	PC4
ETH	RMII	ETH_RXD1	PC5
ETH	RMII	ETH_TXD0	PG13
ETH	RMII	ETH_TXD1	PG12
ETH	RMII	ETH_TX_EN	PG11
RCC	BYPASS Clock Source	RCC_OSC_IN	PH0-OSC_IN
RCC	Crystal/Ceramic Resonator	RCC_OSC32_IN	PC14-OSC32_IN
RCC	Crystal/Ceramic Resonator	RCC_OSC32_OUT	PC15-OSC32_OUT
RCC	USB_HS	RCC_USB	VP_RCC_USB
SYS	SysTick	SYS_VS_Systick	VP_SYS_VS_Systick
UART5	Asynchronous	UART5_RX	PB5
UART5	Asynchronous	UART5_TX	PB6
UART5	CTS/RTS	UART5_CTS	PC9
UART5	CTS/RTS	UART5_RTS	PC8
UART7	Asynchronous	UART7_RX	PE7
UART7	Asynchronous	UART7_TX	PE8
UART7	CTS/RTS	UART7_CTS	PE10
UART7	CTS/RTS	UART7_RTS	PE9
USART3	Asynchronous	USART3_RX	PD9
USART3	Asynchronous	USART3_TX	PD8
USART3	CTS/RTS	USART3_CTS	PB13
USART3	CTS/RTS	USART3_RTS	PB14
USB_OTG_HS	Device_Only	USB_OTG_HS_DM	PA11
USB_OTG_HS	Device_Only	USB_OTG_HS_DP	PA12
USB_OTG_HS	Activate-VBUS	USB_OTG_HS_VBUS	PA9
VREFBUF	Internal voltage reference	VREFBUF_OUT	VREF+

7.2 Διάγραμμα ροής λειτουργίας

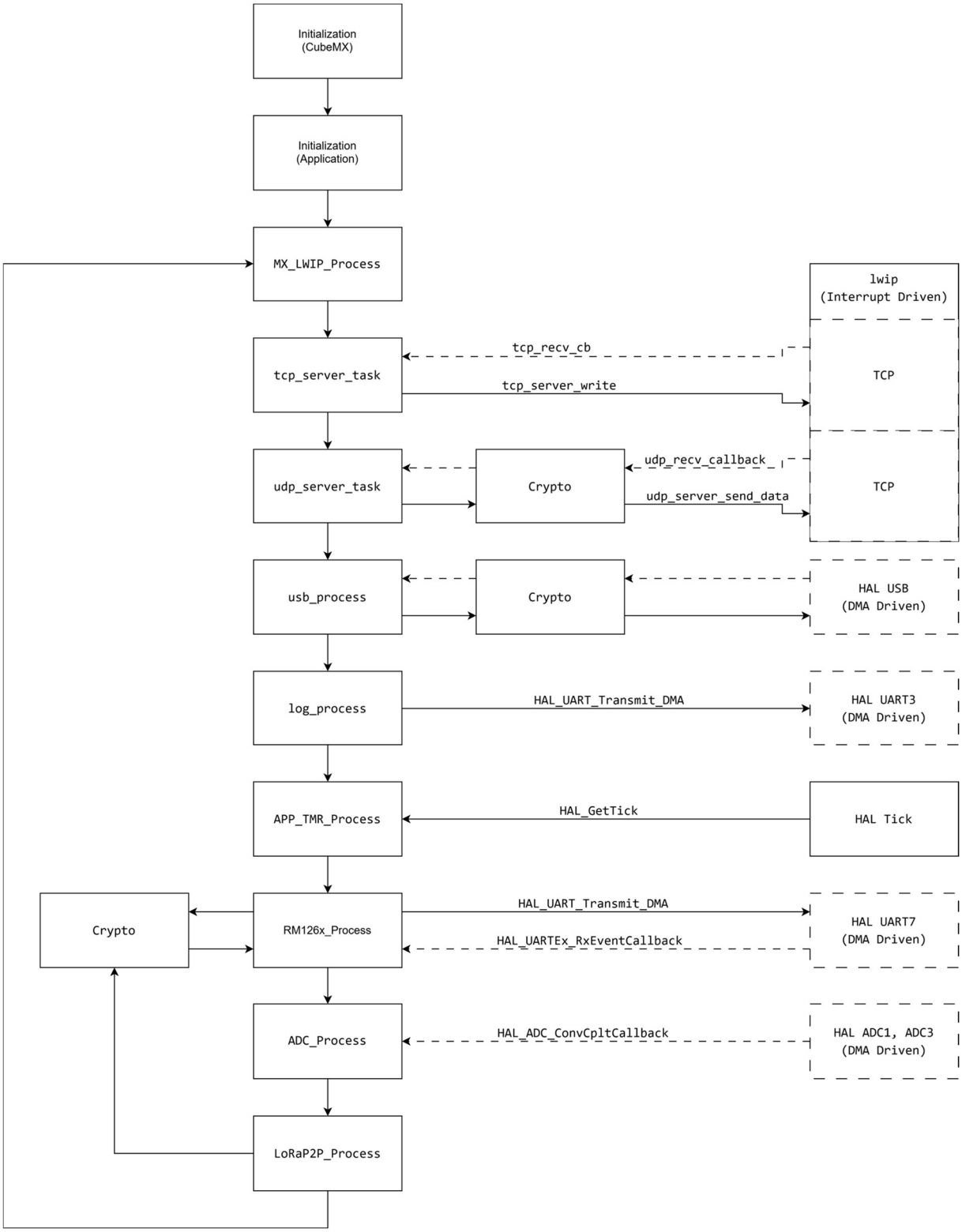
Το firmware ακολουθεί αρθρωτή (modular) αρχιτεκτονική, όπου κάθε υποσύστημα υλοποιείται ως διακριτή λειτουργική ενότητα. Η προσέγγιση αυτή επιτρέπει:

1. καθαρό διαχωρισμό αρμοδιοτήτων
2. ευκολότερη αποσφαλμάτωση και επαλήθευση
3. μελλοντική επέκταση χωρίς ανασχεδιασμό

Οι βασικές ενότητες του firmware είναι:

1. συλλογή και προεπεξεργασία αναλογικών δεδομένων
2. κρυπτογράφηση δεδομένων
3. δημιουργία και διαχείριση πακέτων
4. διεπαφές επικοινωνίας (USB, Ethernet, LoRa)
5. έλεγχος χρονισμού και ροής

Στο σχήμα που ακολουθεί απεικονίζεται το διάγραμμα ροής και οι αλληλεπίδραση της εφαρμογής με το σύστημα.



Σχήμα 26 Διάγραμμα ροής Firmware

Τα τμήματα κώδικα που βρίσκονται στην δεξιά πλευρά του διαγράμματος (Σχήμα 26) και απεικονίζονται με διακεκομμένο περίγραμμα εκτελούνται ασύγχρονα σε σχέση με την κυρίως ροή γιατί συνεργάζονται με τα περιφερειακά μέσω του HAL και ενημερώνουν το κύριο πρόγραμμα για συμβάντα όπως σφάλματα αποσύνδεσης και λήψη δεδομένων. Οι ειδοποιήσεις στο σχεδιάγραμμα συμβολίζονται με διακεκομμένα βέλη. Στις επόμενες ενότητες περιγράφεται ο ρόλος κάθε τμήματος του διαγράμματος.

7.2.1 Initialization (CubeMX)

Εκτελεί τον κώδικα που έχει παραχθεί από το CubeMX για την αρχικοποίηση των περιφερειακών.

7.2.2 Initialization (Application)

Αρχικοποιεί τα τμήματα:

1. tcp_server_task
2. udp_server_task
3. usb_process
4. log_process
5. APP_TMR_Process
6. RM126x_Process
7. ADC_Process

7.2.3 MX_LWIP_Process

Είναι κομμάτι του LWIP και πρέπει να εκτελείται περιοδικά ώστε να χειρίζεται της εσωτερικές λειτουργίες του TCP/IP stack.

7.2.4 tcp_server_task

Περιμένει για εισερχόμενα αιτήματα σύνδεσης. Αν δεν υπάρχει ήδη ενεργή σύνδεση την αποδέχεται και εγκαθίσταται η σύνδεση. Στην αντίθετη περίπτωση απορρίπτει το αίτημα.

Για όσο είναι ενεργή η σύνδεση δέχεται τα εισερχόμενα δεδομένα, τα επεξεργάζεται και στέλνει την κατάλληλη απάντηση στον αποστολέα.

Όταν κλείσει η σύνδεση με οποιοδήποτε τρόπο τότε επιστρέφει στην κατάσταση αναμονής για νέο αίτημα.

7.2.5 udp_server_task

Είναι υπό τον έλεγχο του tcp_server_task. Όταν υπάρχει ενεργή σύνδεση τότε αναλαμβάνει να στείλει τα δεδομένα που ζητήθηκαν μέσω του tcp_server_task.

7.2.6 usb_process

Δέχεται τα δεδομένα που λαμβάνονται μέσω της θύρας USB, τα επεξεργάζεται και στέλνει δεδομένα αν αυτό έχει ζητηθεί.

7.2.7 log_process

Εκτελεί μια βοηθητική αλλά ιδιαιτέρως χρήσιμη λειτουργία. Συνεργάζεται μέσω του HAL με το UART3 σε συνδυασμό με DMA (Direct Memory Access), για να στείλει μηνύματα με την μορφή κειμένου για συμβάντα ή σφάλματα που συνέβησαν στην εφαρμογή.

7.2.8 APP_TMR_Process

Εκτελεί μια βοηθητική λειτουργία παρέχοντας στην εφαρμογή έναν σύνολο από μετρητές χρόνο με ανάλυση 1ms για την περιοδική εκτέλεσης εργασιών με ευέλικτο τρόπο.

7.2.9 RM126x_Process

Παραμετροποιεί το LoRa module RM1261 και μορφοποιεί τα δεδομένα που πρέπει να σταλούν στο δίκτυο.

7.2.10 ADC_Process

Η αναλογοψηφιακή μετατροπή των αναλογικών σημάτων υλοποιήθηκε με την συνεργασία τριών περιφερειακών. Ένας μετρητής (TIM6) εκκινεί την ακολουθιακή μετατροπή των οκτώ αναλογικών καναλιών. Κάθε φορά που ολοκληρώνεται η μετατροπή ένα κανάλι DMA μεταφέρει το αποτέλεσμα σε μια προκαθορισμένη περιοχή μνήμης για να αρχίσει η επόμενη μετατροπή. Ο κύκλος αρχίζει πάλι με τον επόμενο παλμό πυροδότησης από τον μετρητή. Όταν έχουν ολοκληρωθεί οκτώ κύκλοι το DMA ειδοποιεί την «ADC_Process». Για κάθε κανάλι υπολογίζεται η μέση τιμή των οκτώ δειγμάτων που έχουν ληφθεί και αυτή είναι η τιμή του καναλιού.

7.2.11 LoRaP2P_Process

Στέλνει περιοδικά τις τρέχουσες τιμές των οκτώ καναλιών κρυπτογραφημένες προς το δίκτυο LoRa.

7.3 Crypto

Ο μηχανισμός κρυπτογράφησης που υλοποιήθηκε ανήκει στην κατηγορία των stream ciphers. Για κάθε πακέτο δεδομένων παράγεται μία ψευδοτυχαία ακολουθία (keystream), η οποία συνδυάζεται με τα δεδομένα μέσω πράξης XOR.

Η βασική ιδέα είναι ότι το keystream δεν προέρχεται από γραμμικό αλγόριθμο αλλά από ένα χαοτικό δυναμικό σύστημα δύο διαστάσεων. Το σύστημα αυτό παράγει τιμές υψηλής ευαισθησίας στις αρχικές συνθήκες και στη συνέχεια οι τιμές μετατρέπονται σε 32-bit λέξεις με μη γραμμική επεξεργασία.

Κάθε πακέτο κρυπτογραφείται ανεξάρτητα. Ο μετρητής πακέτων χρησιμοποιείται ως seed, επιτρέποντας στον δέκτη να αποκρυπτογραφεί ακόμη και αν έχουν χαθεί ενδιάμεσα πακέτα.

7.3.1 Αρχικοποίηση (Seeding)

Για κάθε πακέτο:

$$seed1 = packetCounter$$
$$seed2 = packetCounter \text{ ^ } 0xA5A5A5A5$$

Οι τιμές αυτές μετατρέπονται σε πραγματικούς αριθμούς στο διάστημα (0,1):

$$x = \frac{(seed + 0.5)}{2^{32}}$$

Έτσι αποφεύγονται ακραίες τιμές (0 ή 1) που θα κατέστρεφαν τη χαοτική συμπεριφορά.

7.3.2 Χαοτικό Σύστημα (Hybrid Chaotic Map)

Χρησιμοποιείται συνδυασμός δύο χαοτικών χαρτών:

Logistic Map

$$x_{log} = \mathrm{r} \cdot x \cdot (1 - x), \mathrm{r} = 3.99 (17)$$

Tent Map

$$x_{tent} = f(x) = \begin{cases} 2 \cdot x, x < 0.5 \\ 2 \cdot (1 - x), x \geq 0.5 \end{cases} (18)$$

Οι δύο χάρτες συζευγνύονται (cross-coupling) και προστίθεται μικρός θόρυβος (noise) από έναν ακέραιο μετρητή.

Noise από counter

$$\text{counter} \mathrel{+}= 0\text{x9E3779B9} \quad (19)$$

$$noise = \left(\left(\frac{counter}{2^{32}} \right) - 0.5 \right) \cdot 10^{-3} (20)$$

Cross coupling

$$x_{log} = x_{log} + 0.5 \cdot x_{tent} + noise (21)$$

$$x_{tent} = x_{tent} + 0.5 \cdot x_{log} - noise \quad (22)$$

Το αποτέλεσμα περιορίζεται στο διάστημα (0,1) κρατώντας μόνο το κλασματικό μέρος. Αυτή η διαδικασία επαναλαμβάνεται αρκετές φορές πριν την παραγωγή κάθε λέξης ώστε να ενισχυθεί η αποσυσχέτιση.

7.3.3 Μετατροπή σε Ψευδοτυχαία Λέξη (PRNG Output)

Οι τιμές των χαοτικών μεταβλητών δεν χρησιμοποιούνται άμεσα. Αντί αυτού, χρησιμοποιείται το IEEE-754 bit pattern τους. Αν:

$$A = bits(x_{log})$$

$$B = bits(x_{tent})$$

$$C = bits(x_{counter})$$

τότε εφαρμόζεται μη γραμμική μίξη:

$$mixed = A \text{ ^ } (B \ll 1) \text{ ^ } (C \ll 17) \text{ ^ } (A + B \cdot 0x9E3779B9) \quad (23)$$

7.3.4 Avalanche Hash (Τελική Διασπορά Bit)

Για την αποφυγή στατιστικών συσχετίσεων εφαρμόζεται hash τύπου Murmur finalizer [53]:

$$x \mathrel{\widehat{=}} x \gg 16$$

$$x \mathrel{*}= 0x7FEB352D$$

$$x \mathrel{\widehat{=}} x \gg 15$$

$$x \mathrel{*}= 0x846CA68B$$

$$x \mathrel{\widehat{=}} x \gg 16$$

Η διαδικασία αυτή διασφαλίζει ότι μικρή αλλαγή στην είσοδο προκαλεί μεγάλη αλλαγή στην έξοδο (avalanche effect).

7.3.5 Παραγωγή Keystream και Κρυπτογράφηση

Η 32-bit λέξη διασπάται σε bytes (little-endian) και εφαρμόζεται XOR:

$$\text{ciphertext} = \text{plaintext} \text{ ^ } \text{keystream} \quad (24)$$

Η αποκρυπτογράφηση είναι η ίδια πράξη:

$$\text{plaintext} = \text{ci}\textit{pher}\text{text} \text{ ^ } \text{keystream} \quad (25)$$

7.3.6 Επεξηγήσεις Σχεδιαστικών Επιλογών

Ο logistic map έχει ισχυρή μη γραμμικότητα αλλά εμφανίζει περιοδικότητα σε πεπερασμένη ακρίβεια. Ο tent map έχει καλύτερη ομοιομορφία κατανομής. Ο συνδυασμός τους αυξάνει την εντροπία και μειώνει τα στατιστικά ίχνη.

Η χρήση του counter ως seed καθιστά το κάθε frame γίνεται ανεξάρτητο κρυπτόγραμμα. Έτσι:

- δεν απαιτείται συγχρονισμός ροής
- δεν επηρεάζει η απώλεια πακέτων
- επιτρέπεται αποκρυπτογράφηση εκτός σειράς

Οι χαοτικοί χάρτες παράγουν πραγματικούς αριθμούς που μπορεί να έχουν τοπικές συσχετίσεις στα bits. Ο avalanche hash εξασφαλίζει πλήρη διάχυση πληροφορίας: αλλαγή 1 bit στην είσοδο επηρεάζει περίπου το 50% των bits της εξόδου.

Οι αριθμοί που χρησιμοποιήθηκαν δεν είναι τυχαίοι. Συγκεκριμένα:

- 0x9E3779B9 ή 2654435761. Είναι ο αριθμός της χρυσής τομής σε μορφή 32-bit. Χρησιμοποιείται ευρέως σε hash και PRNG γιατί κατανέμει ομοιόμορφα τα bits και αποφεύγει επαναλήψεις.
- 0x7FEB352D και 0x846CA68B. Σταθερές από τον MurmurHash3 finalizer. Έχουν επιλεγεί πειραματικά ώστε να επιτυγχάνουν ισχυρή διάχυση bit χωρίς μεγάλο υπολογιστικό κόστος.
- Προσθήκη θορύβου 10^{-3}. Σπάει τις συμμετρίες του συστήματος σε πεπερασμένη ακρίβεια κινητής υποδιαστολής και αποτρέπει την είσοδο σε μικρούς κύκλους (limit cycles).

7.4 Χειρισμός RM1261 σε P2P mode

Όπως αναφέρθηκε στις ενότητες 1.2, 1.3 το RM1261 διατίθεται με εγκατεστημένο firmware για την υλοποίηση ενός Peer-to-Peer δικτύου. Η κατασκευάστρια εταιρεία, eZurio, παρέχει οδηγίες με την μορφή application note [48] καθώς και για το σύνολο των εντολών που υποστηρίζονται [49].

7.4.1 Αρχικοποίηση για σύνδεση στο δίκτυο P2P

Η προτεινόμενη αλληλουχία AT εντολών, όπως αυτή υποδεικνύεται [48], ώστε να μπορέσει το module να συνδεθεί στο δίκτυο είναι αυτή που παρουσιάζεται με τον ακόλουθο πίνακα. Για κάθε εντολή υπάρχει και μια σύντομη εξήγηση για την λειτουργία που εκτελεί. Περισσότερες πληροφορίες για αυτές και για όλες τις υποστηριζόμενες υπάρχουν στο εγχειρίδιο χρήσης [49].

Πίνακας 10 Αλληλουχία εντολών AT αρχικοποίησης

Εντολή	Περιγραφή
AT&F	Φορτώνει τις εργοστασιακές ρυθμίσεις
ATZ	Soft Reset
ATS603=3	Κλάση του δικτύου. 3 για P2P [49]
ATS611=1	Το region που θα λειτουργήσει. 1 για Ευρώπη [49]
AT&W	Αποθήκευση των αλλαγών
ATZ	Soft Reset
ATS700=1	Διεύθυνση του κόμβου στο δίκτυο
ATS701=4	Το μέγεθος του δικτύου
ATS703=5	Ρυθμός μετάδοσης δεδομένων. 5 είναι η ανώτερη τιμή [49]
ATS704=0	Διάρκεια του «listening»
ATS705=0	Συχνότητα του «listening»
ATS706=5	Ρυθμός μετάδοσης δεδομένων για το beacon. 5 είναι η ανώτερη τιμή [49]
ATS708=1	Εκπεμπόμενη ισχύς. Αυτή είναι η ελάχιστη τιμή. Προσοχή: Μεγαλύτερη τιμή για μικρές αποστάσεις μπορεί να καταστρέψει τους δέκτες.
ATS707=18	Μέγιστη τιμή μήκους πακέτου
ATS702=8	Διάρκεια παραθύρου [49]
AT+P2PS	Θέτει το module σε κατάσταση αναμονής για να λάβει πακέτα beacon και να συνδεθεί στο δίκτυο

7.4.2 Επανασύνδεση

Εάν για οποιοδήποτε λόγο το node0 (1.2, 1.3) σταματήσει να στέλνει beacon τότε το module αποσυνδέεται και ενημερώνει σχετικά τον κύριο μικροελεγκτή για το συμβάν. Η διαδικασία επανασύνδεσης είναι να αποσυνδεθεί και να έλθει σε κατάσταση αναμονής για να λάβει πακέτα beacon και να συνδεθεί στο δίκτυο. Δεν απαιτείται εκ νέου αρχικοποίηση. Στον ακόλουθο πίνακα παρατίθενται οι σχετικές εντολές.

Πίνακας 11 Αλληλουχία εντολών AT επανασύνδεσης

Εντολή	Περιγραφή
AT+P2PE	Αποσύνδεση από το δίκτυο
AT+P2PS	Θέτει το module σε κατάσταση αναμονής για να λάβει πακέτα beacon και να συνδεθεί στο δίκτυο

7.4.3 Αποστολή και λήψη δεδομένων

Η αποστολή δεδομένων προς το δίκτυο γίνεται μέσω κατάλληλης εντολής AT, ενώ η λήψη στέλνεται από το module ασύγχρονα μέσω ειδοποίησης. Τα δεδομένα στην αποστολή και λήψη είναι μορφοποιημένα σε δεκαεξαδική απεικόνιση. Αυτό σημαίνει ότι για την αναπαράσταση ενός byte απαιτούνται δύο χαρακτήρες. Για παράδειγμα: ο αριθμός 12h ή 0x12 θα απεικονιστεί ως «12» και σε δυαδική μορφή είναι οι αριθμοί 0x31, 0x32 σύμφωνα με τον πίνακα ASCII. Αναλυτικά περιγράφονται στην συνέχεια.

AT+P2PT n, “data”	Αποστέλλει στον κόμβο με διεύθυνση n τα δεδομένα “data”
RX: [P2P] S:NN, R:XY, S:ZW, data	Στο πακέτο αυτό οι πληροφορίες που περιέχονται είναι: • data: Τα δεδομένα που μεταφέρει το πακέτο • (S)ender: Η διεύθυνση στο δίκτυο του αποστολέα • (R)SSI: Ο δείκτης ισχύς σήματος λήψης • (S)NR: Ο λόγος σήματος προς θόρυβο

7.5 Πρωτόκολλο επικοινωνίας

Η επικοινωνία γίνεται ομαδοποιώντας τα δεδομένα σε μικρά πακέτα των 18-bytes. Αυτό το όριο τέθηκε από το LoRa σε λειτουργία Peer-to-Peer. Γενικά όπως αναφέρθηκε σε προηγούμενο κεφάλαιο η χωρητικότητα του LoRa είναι μικρή συγκρινόμενη με άλλα μέσα, όπως για παράδειγμα το Ethernet.

Η δομή ενός πακέτου απεικονίζεται στο ακόλουθο σχήμα.

Σχήμα 27 Δομή κρυπτογραφημένου πακέτου δεδομένων

Το CTR είναι ένας 16-bit αριθμός που αυξάνεται με την αποστολή ενός πακέτου και ταυτόχρονα είναι και το seed με το οποίο έγινε η κρυπτογράφηση και που πρέπει να χρησιμοποιηθεί ως seed στην αποκρυπτογράφηση για την ανάκτηση των δεδομένων.

7.5.1 Κανάλι LoRa

Στο κανάλι LoRa τα δεδομένα στέλνονται περιοδικά προς όλα τα nodes στο δίκτυο (Broadcasting) και είναι πάντα κρυπτογραφημένα διότι κατά κανόνα τα ασύρματα μέσα έχουν μεγαλύτερη βαθμό επισφάλειας. Ο ρυθμός με τον οποίο στέλνονται καθορίζεται από την παραμετροποίηση του Peer-to-Peer δικτύου. Στην παρούσα υλοποίηση είναι 8s.

7.5.2 Κανάλι TCP/UDP

Η συσκευή περιμένει για αιτήματα TCP σύνδεσης στην πόρτα 49152 και αν δεν υπάρχει ήδη ενεργή σύνδεση την αποδέχεται διαφορετικά απορρίπτει το αίτημα. Ο δίαυλος που ανοίγει μετά την σύνδεση εξυπηρετεί ως κανάλι ελέγχου και καθορίζει πώς θα μεταφέρονται τα δεδομένα μέσω πόρτας 49152 του UDP με την διεύθυνση IP που αιτήθηκε την σύνδεση στο TCP.

Οι εντολές που υποστηρίζονται είναι οι εξής:

Πίνακας 12 Υποστηριζόμενες εντολές πρωτοκόλλου μεταφοράς μετρήσεων

<START,MM,TT>	Επιστρέφει τις τρέχουσες τιμές των αναλογικών μετρήσεων, όπου MM ο τρόπος απεικόνισης των τιμών 01 = Δυαδική κρυπτογραφημένη (7.5) 02 = Δυαδική μη κρυπτογραφημένη (7.5) 03 = Κείμενο της μορφής <V1,V2,V3,V4,V5,V6,V7,V8>. Οι V1 έως V8 αριθμοί μήκους 4ch που εκφράζουν τις τιμές των μετρήσεων σε mV TT ο ρυθμός αποστολής των δεδομένων σε βήμα 100ms. Αν είναι 0 στέλνει μόνο ένα πακέτο.
<END>	Σταματάει την αποστολή δεδομένων αν η τιμή του TT που την ξεκίνησε δεν ήταν μηδέν.

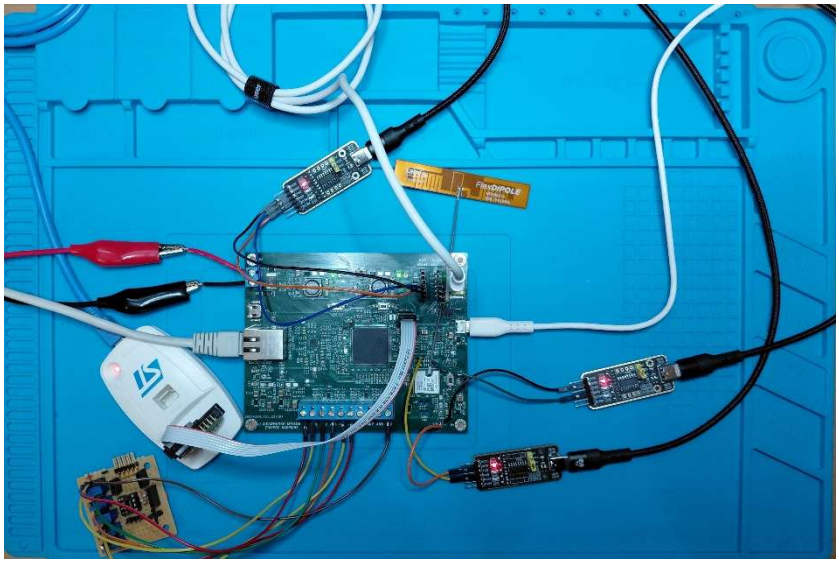
7.5.3 Κανάλι USB

Το USB λειτουργεί ως εικονική σειριακή. Όταν ανοίξει η θύρα από κάποια εφαρμογή που εκτελείται στον host τότε αρχίζει η αποστολή δεδομένων με όμοια δομή με τα άλλα τρία κανάλια επικοινωνίας.

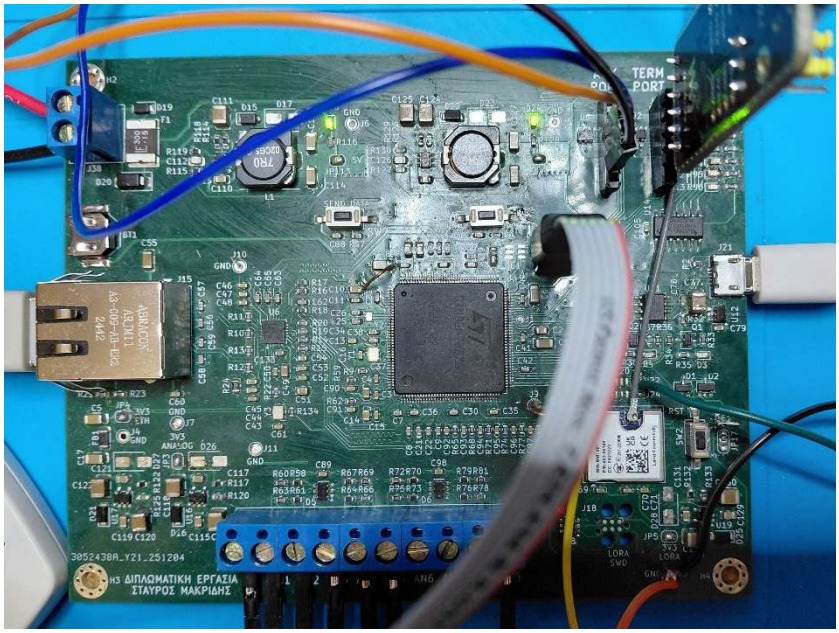
Κεφάλαιο 8ο: Πειραματικός έλεγχος

8.1 Διάταξη ανάπτυξης και δοκιμών

Κατά την διάρκεια της ανάπτυξης του firmware τέσσερις μετατροπής από UART σε USB για την παρακολούθηση της επικοινωνίας μεταξύ MCU και LoRa module, για καταγραφή σφαλμάτων κατά την διάρκεια της εκτέλεσης του προγράμματος. Δεν θα μπορούσε να λείπει ένας programmer/debugger από την διαδικασία ανάπτυξης οποιασδήποτε εφαρμογής μικροελεγκτή ανεξαρτήτως μεγέθους και πολυπλοκότητας αυτής. Στην εργασία αυτή το εργαλείο που χρησιμοποιήθηκε είναι το δημοφιλές ST-LINK/V2. Η συνολική διάταξη απεικονίζεται στις επόμενες εικόνες.

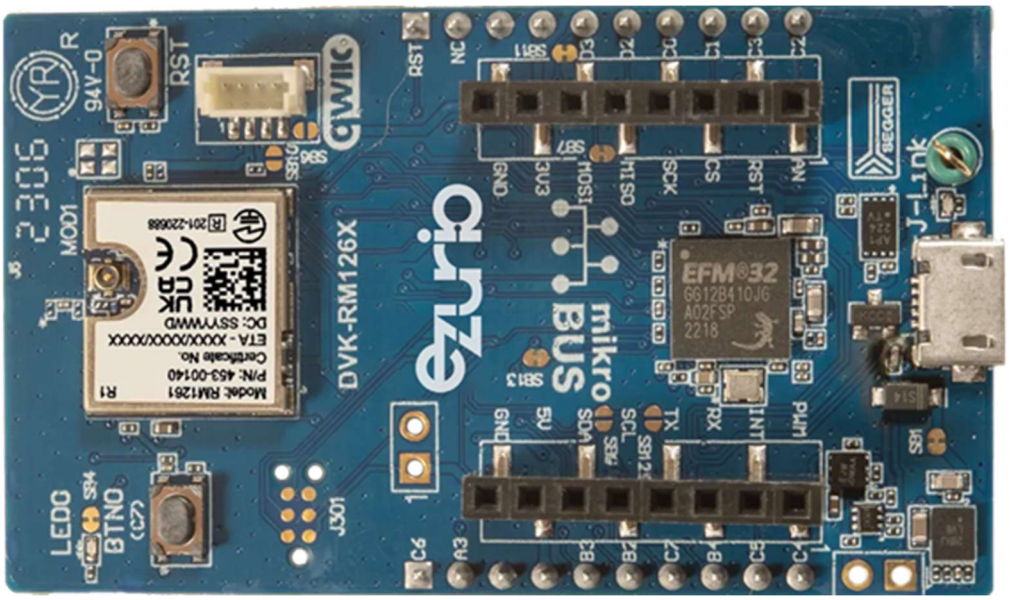


Εικόνα 1 Διάταξη ανάπτυξης και δοκιμών



Εικόνα 2 Η πλακέτα από κοντά

Τον ρόλο του δέκτη και κεντρικού κόμβου εξυπηρετεί το αναπτυξιακό της eZurio, 453-00140-K1 Development Kit (https://www.ezurio.com/part/453-00140-k1) που δεν φαίνεται στις παραπάνω εικόνες διότι πρέπει να βρίσκεται μερικά μέτρα μακριά για την προστασία των δεκτών. Στον υπολογιστή εμφανίζεται ως μια εικονική σειριακή θύρα και υποστηρίζει τις ίδιες AT εντολές με το RM1261. Στην ακόλουθη εικόνα απεικονίζεται το 453-00140-K.



Εικόνα 3 453-00140-K1

8.2 Επιβεβαίωση ορθής λειτουργίας

Για να ελεγχθεί και να διαπιστωθεί ότι η εφαρμογή λειτουργεί σύμφωνα με τον σχεδιασμό χρησιμοποιήθηκε μια θύρα UART του μικροελεγκτή και πρόσθετες λειτουργίες στο επίπεδο του λογισμικού. Για κάθε πακέτο δεδομένων που αποστέλλεται μέσω της σύνδεσης LoRa στέλνεται μια εγγραφή μέσω της UART σε ένα πρόγραμμα κονσόλας (Tera Term, https://teratermproject.github.io/index-en.html) που εκτελείται στον υπολογιστή. Η εγγραφή αυτή έχει την μορφή:

```
Sent packet #00474, Raw Data: AD06980AE30AE30A870A000000000000,
                Encrypted Data: 138BC4E4CBEABE0B0B5ED9171C2BD7F4
```

Όπου:

#00474, ο αύξων αριθμός της εγγραφής. Χρησιμοποιείται μόνο για συσχέτιση των δεδομένων

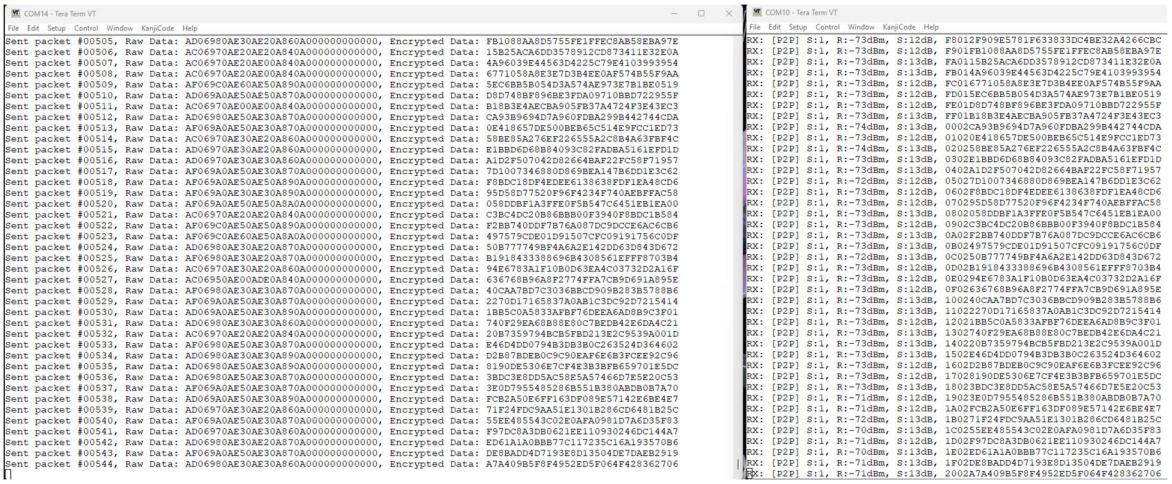
Raw Data, τα αρχικά δεδομένα πριν την κρυπτογράφηση

Encrypted Data, τα δεδομένα μετά την κρυπτογράφηση που αποστάλθηκαν

Κατά όμοιο τρόπο τα δεδομένα καταγράφονται και για τον δέκτη. Η μορφή κάθε εγγραφής είναι ως εξής όπως περιεγράφηκε στην ενότητα 7.4.3:

```
RX: [P2P] S:1, R:-73dBm, S:13dB, DA01138BC4E4CBEABE0B0B5ED9171C2BD7F4
```

Στην εικόνα που ακολουθεί είναι ένα στιγμιότυπο από τις εφαρμογές καταγραφής.



Εικόνα 4 Στιγμιότυπο καταγραφής από Tera Term

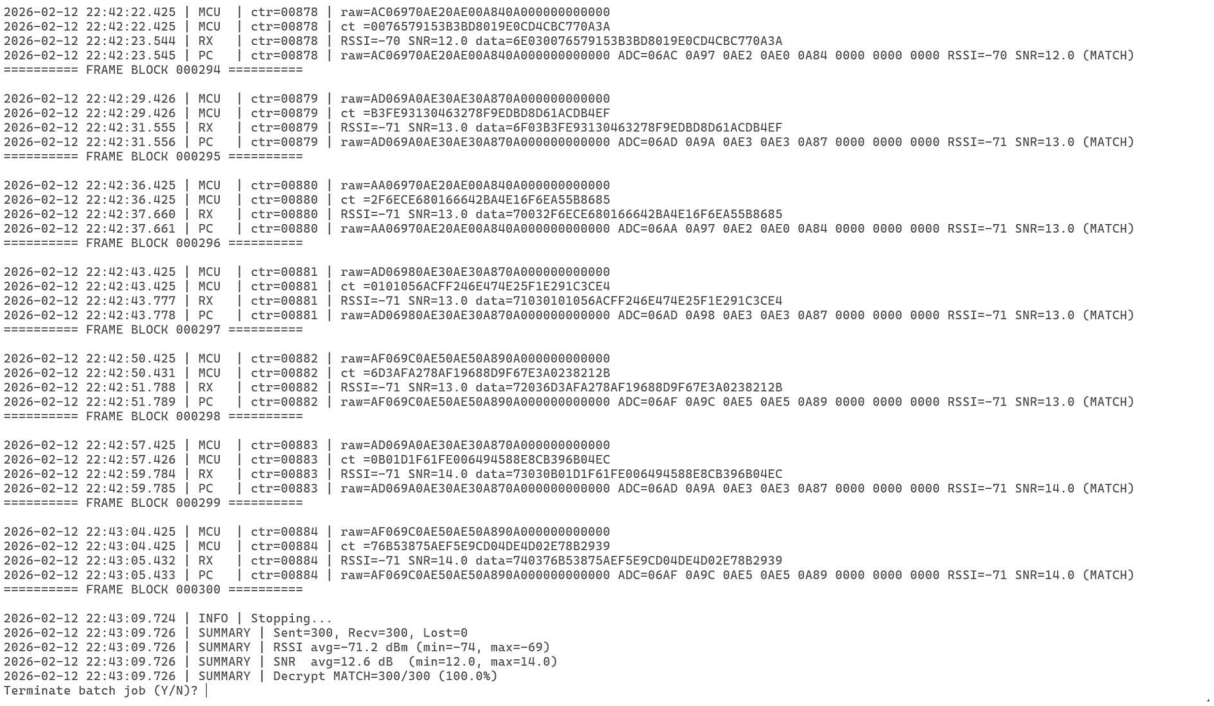
Αυτός ο τρόπος δεν είναι παραγωγικός όταν χρειάζεται να γίνει αποκρυπτογράφηση των λαμβανομένων δεδομένων ώστε να διαπιστωθεί ότι το κανάλι λειτουργεί από άκρη σε άκρη. Για τον λόγο αυτό έπρεπε να αναπτυχθεί μια εφαρμογή που να δέχεται τα εκπεμπόμενα πακέτα καθώς και τα λαμβανόμενα, να εκτελεί αποκρυπτογράφηση σε αυτά και να τα συγκρίνει με αυτά που εκπέμφθηκαν και τέλος να αποφασίζει αν το πακέτο πέρασε από το κανάλι χωρίς απώλειες. Το πρόγραμμα γράφτηκε σε Python και ο πλήρης κώδικας του βρίσκεται στο Παράρτημα Ε.

Η εφαρμογή συνδέεται σε δύο σειριακές θύρες, λαμβάνει δεδομένα και από τις δύο και τα απεικονίζει ως μια εγγραφή συσχετίζοντας τες. Στην εικόνα που ακολουθεί φαίνεται τι απεικονίζει όσο διαρκεί η καταγραφή.

```
C:\Users\smakr\MastersProgram\Courses\Dissertation\Software\CommCorrelator>python comm_correlator.py --mcu COM14 --rx COM10 --out unified_log.txt --print_adc
2026-02-12 22:08:05.774 | INFO | Started. MCU=COM14@115200 RX=COM10@115200 OUT=unified_log.txt
2026-02-12 22:08:11.422 | MCU  | ctr=00585 | raw=AC06970AE20AE00A840A000000000000
2026-02-12 22:08:11.422 | MCU  | ct =1BE722CE5820A051270CE9FB1B552ED0
2026-02-12 22:08:13.501 | RX   | ctr=00585 | RSSI=-71 SNR=13.0 data=49021BE722CE5820A051270CE9FB1B552ED0
2026-02-12 22:08:13.502 | PC   | ctr=00585 | raw=AC06970AE20AE00A840A000000000000 ADC=06AC 0A97 0AE2 0AE0 0A84 0000 0000 0000 RSSI=-71 SNR=13.0 (MATCH)
========== FRAME BLOCK 000001 ==========

2026-02-12 22:08:18.422 | MCU  | ctr=00586 | raw=AF069C0AE60AE50A8A0A000000000000
2026-02-12 22:08:18.422 | MCU  | ct =56EA2BFAAD81369831FF15997F590828
2026-02-12 22:08:19.595 | RX   | ctr=00586 | RSSI=-71 SNR=13.0 data=4A0256EA2BFAAD81369831FF15997F590828
2026-02-12 22:08:19.596 | PC   | ctr=00586 | raw=AF069C0AE60AE50A8A0A000000000000 ADC=06AF 0A9C 0AE6 0AE5 0A8A 0000 0000 0000 RSSI=-71 SNR=13.0 (MATCH)
========== FRAME BLOCK 000002 ==========

2026-02-12 22:08:25.422 | MCU  | ctr=00587 | raw=AD06980AE50AE30A860A000000000000
2026-02-12 22:08:25.422 | MCU  | ct =10FA5B4F32A0CCCA2026B9ED1C9099E0
2026-02-12 22:08:27.602 | RX   | ctr=00587 | RSSI=-71 SNR=13.0 data=4B0210FA5B4F32A0CCCA2026B9ED1C9099E0
2026-02-12 22:08:27.604 | PC   | ctr=00587 | raw=AD06980AE50AE30A860A000000000000 ADC=06AD 0A98 0AE5 0AE3 0A86 0000 0000 0000 RSSI=-71 SNR=13.0 (MATCH)
========== FRAME BLOCK 000003 ==========

2026-02-12 22:08:32.422 | MCU  | ctr=00588 | raw=AD06980AE30AE30A860A000000000000
2026-02-12 22:08:32.422 | MCU  | ct =E08BAE1262FA82A45A1BD26A360D66D2
2026-02-12 22:08:33.250 | RX   | ctr=00588 | RSSI=-71 SNR=13.0 data=4C02E08BAE1262FA82A45A1BD26A360D66D2
2026-02-12 22:08:33.251 | PC   | ctr=00588 | raw=AD06980AE30AE30A860A000000000000 ADC=06AD 0A98 0AE3 0AE3 0A86 0000 0000 0000 RSSI=-71 SNR=13.0 (MATCH)
========== FRAME BLOCK 000004 ==========

2026-02-12 22:08:39.422 | MCU  | ctr=00589 | raw=AD06980AE30AE30A870A000000000000
2026-02-12 22:08:39.422 | MCU  | ct =39C47B17430AB906141CF515AD31EB7F
2026-02-12 22:08:41.247 | RX   | ctr=00589 | RSSI=-71 SNR=13.0 data=4D0239C47B17430AB906141CF515AD31EB7F
2026-02-12 22:08:41.248 | PC   | ctr=00589 | raw=AD06980AE30AE30A870A000000000000 ADC=06AD 0A98 0AE3 0AE3 0A87 0000 0000 0000 RSSI=-71 SNR=13.0 (MATCH)
========== FRAME BLOCK 000005 ==========

2026-02-12 22:08:46.422 | MCU  | ctr=00590 | raw=AD06980AE30AE30A860A000000000000
2026-02-12 22:08:46.422 | MCU  | ct =0B43D664B97A19E4571C4B15E44EDEA1
2026-02-12 22:08:47.370 | RX   | ctr=00590 | RSSI=-71 SNR=13.0 data=4E020B43D664B97A19E4571C4B15E44EDEA1
2026-02-12 22:08:47.371 | PC   | ctr=00590 | raw=AD06980AE30AE30A860A000000000000 ADC=06AD 0A98 0AE3 0AE3 0A86 0000 0000 0000 RSSI=-71 SNR=13.0 (MATCH)
========== FRAME BLOCK 000006 ==========
```

Εικόνα 5 Εκκίνηση συγκεντρωτικής καταγραφής

Η εκτέλεση διακόπτεται πατώντας Ctrl+C και εμφανίζει στην οθόνη το ποσοστό των πακέτων για τα οποία επιβεβαιώθηκε η ορθή λήψη τους, όπως στην εικόνα που ακολουθεί.



Εικόνα 6 Εκκίνηση συγκεντρωτικής καταγραφής και εμφάνιση αποτελέσματος

Με τον τρόπο αυτό αποδεικνύεται και πειραματικά ότι τα δεδομένα περνούν ακέραια από το κρυπτογραφημένο κανάλι ξεκινώντας από την πηγή μέχρι τον τελικό αποδέκτη.

ΕΠΙΛΟΓΟΣ: Αποτίμηση – Επεκτάσεις – Συμπεράσματα

Ο στόχος της εργασίας ήταν να συνδυάσει την βασισμένη στους χαοτικούς χάρτες κρυπτογραφία με την ασύρματη επικοινωνία LoRa. Δύο πολύ ενδιαφέροντα πεδία που μπορούν να βρουν ακόμα ευρύτερη εφαρμογή σε ποικίλους τομείς της τεχνολογίας. Αποδείχθηκε ότι μπορούν να συνεργαστούν με αξιώσεις ακόμα και στον γοητευτικό κόσμο των μικροϋπολογιστών και στον ανερχόμενο ραγδαία τομέα του IoT (Internet of Things).

Το firmware που αναπτύχθηκε δεν εξαντλεί σε καμία περίπτωση τις δυνατότητες του συγκεκριμένου hardware, αλλά είναι μια επίδειξη των δυνατοτήτων του. Μερικές προσθήκες και βελτιώσεις θα μπορούσαν να είναι:

- Η ταυτοποίηση μέσω όνομα χρήστη και κωδικού πριν την σύνδεση και την αποστολή δεδομένων.
- Η αμφίδρομη επικοινωνία για εφαρμογές ασφαλούς τηλεχειρισμού και τηλεμετρίας.
- Η λειτουργία ενδιάμεσου (Proxy) μεταξύ συσκευών που δεν απαιτούν ιδιαιτέρως μεγάλο όγκο δεδομένων και δεν έχουν εν γένει δυνατότητα για ασφαλή ασύρματη επικοινωνία (π.χ. PLCs).
- Βελτίωση της σχεδίασης με στόχο την πολύ χαμηλή κατανάλωση ενέργειας για εφαρμογές απομακρυσμένων εκτός δικτύου σταθμών συλλογής δεδομένων.
- Η προσθήκη αποθηκευτικού μέσου για την αποθήκευση δεδομένων ώστε να αποσταλούν σε μεταγενέστερο χρόνο.

Κλείνοντας δεν θα μπορούσα να μην αναφέρω ότι ήταν μια πολύ ενδιαφέρουσα δύσκολη και ανηφορική διαδρομή που όμως ενίσχυσε και εμπλούτισε τις γνώσεις μου και την εμπειρία μου και σίγουρα δεν μετάνιωσα για την απόφαση μου να την διαβώ.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] IEEE, “IEEE Standard for Ethernet,” IEEE Std 802.3-2022, 2022. Available: https://standards.ieee.org/standard/802_3-2022.html

[2] USB Implementers Forum, “Universal Serial Bus Specification, Revision 2.0,” 2000. Available: https://usb.org/document-library/usb-20-specification

[3] Semtech Corporation, “AN1200.22: LoRa® Modulation Basics,” Rev. 2, 2015. Available: https://www.semtech.com/uploads/documents/an1200.22.pdf

[4] Ezurio, “RM126x Series Product Brief,” 2023. Available: https://www.ezurio.com/documentation/product-brief-rm126x-series

[5] Ezurio, “RM126x Series P2P User Guide (Application Note),” 2023. Available: https://www.ezurio.com/documentation/application-note-p2p-user-guide-rm126x-series

[6] STMicroelectronics, “DS13312: STM32H735xG Datasheet,” Rev. 4, 2023. Available: https://www.st.com/resource/en/datasheet/stm32h735vg.pdf

[7] The RMII Consortium, “Reduced Media Independent Interface (RMII) Specification,” Rev. 1.2, 1998. Available: http://ebook.pldworld.com/_eBook/-Telecommunications,Networks-/TCPIP/RMII/rmii_rev12.pdf (online mirror)

[8] STMicroelectronics, “AN5419: Getting Started with STM32H723/733, STM32H725/735 and STM32H730 Value Line Hardware Development,” Rev. 3, 2026. Available: https://www.st.com/resource/en/application_note/an5419-getting-started-with-stm32h723733-stm32h725735-and-stm32h730-mcu-hardware-development-stmicroelectronics.pdf

[9] ECS Inc. International, “ECS-2520MV Series Crystal Oscillator Datasheet,” 2023. Available: https://ecsxtal.com/datasheet/ECS-2520MV.pdf

[10] STMicroelectronics, “RM0468: STM32H723/733, STM32H725/735 and STM32H730 Value Line Advanced Arm®-Based 32-Bit MCUs Reference Manual,” Rev. 3, 2023. Available: https://www.st.com/resource/en/reference_manual/rm0468-stm32h723733-stm32h725735-and-stm32h730-value-line-advanced-armbased-32bit-mcus-stmicroelectronics.pdf

[11] Microchip Technology Inc., “LAN8742A/LAN8742Ai: Small Footprint RMII 10/100 Ethernet Transceiver with HP Auto-MDIX and flexPWR® Technology – Datasheet,” 2023. Available: https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/DataSheets/DS_LAN8742_00001989A.pdf

[12] Microchip Technology Inc., "SC471240: Schematic Checklist for the LAN8742, 24-pin QFN Package," Rev. A, 2012. Available: https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/SupportingCollateral/LAN8742_QFN_Rev_A_Placement_Checklist.pdf

[13] Microchip Technology Inc., "CC524365: Component Placement Checklist for the LAN8742, 24-pin QFN Package," Rev. A, 2012. Available: https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/SupportingCollateral/LAN8742_QFN_Rev_A_Placement_Checklist.pdf

[14] Microchip Technology Inc., "RC614947: Routing Checklist for the LAN8742, 24-pin QFN Package," Rev. A, 2012. Available: https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/SupportingCollateral/LAN8742_QFN_Rev_A_Routing_Checklist.pdf

[15] Abracon, "Datasheet: RJ45, Single Port, 10/100/1G/5G Base-T Magnetics Module," Rev. A, 2025. Available: https://abracon.com/Magnetics/ARJM11.pdf

[16] Ezurio, “RM126x LoRaWAN Module Datasheet,” 2023. Available: https://www.ezurio.com/documentation/datasheet-rm126x-lorawan-module

[17] Segger, "6-Pin Needle Adapter," 2025. Available: https://www.segger.com/products/debug-probes/j-link/accessories/adapters/6-pin-needle-adapter/

[18] STMicroelectronics, "023199: USBLC6-4SC6Y Automotive very low capacitance ESD protection," Rev. 2, 2012. Available: https://www.st.com/resource/en/datasheet/usblc6-4sc6y.pdf

[19] GCT, "USB3090: Micro USB Receptable, Type AB, 5 Pin, SMT, Horizontal, Bottom Mount, without Peg, with Outer Shell Stakes," Rev. D, 2011. Available: https://gct.co/download?type=PDFSpecification&name=usb3090-spec.pdf

[20] STMicroelectronics, "013793: STMPS2141, STMPS2151, STMPS2161, STMPS2171 Enhanced single channel power switches," Rev. 7, 2022. Available: https://www.st.com/resource/en/datasheet/stmps2141.pdf

[21] onsemi, "BC856BWT1/D: BC856B BC857B BC858A, General Purpose Transistors PNP Silicon," Rev. 6, 2024. Available: https://www.onsemi.com/pdf/datasheet/bc856bwt1-d.pdf

[22] LITE-ON, DS-22-99-0151: LTST-C190KRKT datasheet," Rev. H, 2021. Available: https://optoelectronics.liteon.com/upload/download/DS-22-99-0151/LTST-C190KRKT.PDF

[23] LITE-ON, DS22-2000-074: LTST-C190KGKT datasheet," Rev. O, 2021. Available: https://optoelectronics.liteon.com/upload/download/DS22-2000-074/LTST-C190KGKT.PDF

[24] onsemi, "CM1293A−04SO/D: 4-Channel Low Capacitance ESD Protection Array," Rev. 1, 2025. Available: https://www.onsemi.com/download/data-sheet/pdf/cm1293a-04so-d.pdf

[25] Diodes, DS35272: 3A SBR Super Barrier Rectifier PowerDI," Rev. 8-2, 2018. Available: https://www.diodes.com/datasheet/download/SBR3U60P1.pdf

[26] Littelfuse, "Datasheet: 2920L Series Surface Mount PolySwitch® Resettable PPTC," Rev. 1, 2025. Available: https://www.littelfuse.com/assetdocs/2920l_datasheet_update.pdf?assetguid=f237e8c2-1ed9-4c13-a738-dbe0738b3d2c

[27] Littelfuse, "Datasheet: SMAJ Series Surface Mount – 400W TVS Diodes," Rev. 1, 2025. Available: https://www.littelfuse.com/assetdocs/tvs-diodes-smaj-datasheet?assetguid=13c2a823-03b8-4d1f-9ddc-9b44670aed9d

[28] Texas Instruments Inc., "SLVSDG6B: TPS54302 4.5-V to 28-V Input, 3-A Output, EMI-Friendly Synchronous Step-Down Converter," Rev. B, 2021. Available: https://www.ti.com/lit/gpn/tps54302

[29] Diodes, DS40022: 300mA High PSRR, Low Noise LDO with Enable," Rev. 2 - 2, 2018. Available: https://www.diodes.com/datasheet/download/AP7330.pdf

[30] S. H. Strogatz, Nonlinear Dynamics and Chaos, 2nd ed. Boulder, CO, USA: Westview Press, 2015.

[31] K. T. Alligood, T. D. Sauer, and J. A. Yorke, Chaos: An Introduction to Dynamical Systems. New York, NY, USA: Springer, 1996.

[32] R. M. May, “Simple mathematical models with very complicated dynamics,” Nature, vol. 261, pp. 459–467, 1976.

[33] T. Y. Li and J. A. Yorke, “Period three implies chaos,” American Mathematical Monthly, vol. 82, no. 10, pp. 985–992, 1975.

[34] P. Collet and J.-P. Eckmann, Iterated Maps on the Interval as Dynamical Systems. Boston, MA, USA: Birkhäuser, 2009.

[35] L. Kocarev, “Chaos-based cryptography: a brief overview,” IEEE Circuits and Systems Magazine, vol. 1, no. 3, pp. 6–21, 2001.

[36] G. Alvarez and S. Li, “Some basic cryptographic requirements for chaos-based cryptosystems,” International Journal of Bifurcation and Chaos, vol. 16, no. 8, pp. 2129–2151, 2006.

[37] M. S. Baptista, “Cryptography with chaos,” Physics Letters A, vol. 240, no. 1–2, pp. 50–54, 1998.

[38] N. K. Pareek, V. Patidar, and K. K. Sud, “Image encryption using chaotic logistic map,” Image and Vision Computing, vol. 24, no. 9, pp. 926–934, 2006.

[39] G. Chen, Y. Mao, and C. K. Chui, “A symmetric image encryption scheme based on 3D chaotic cat maps,” Chaos, Solitons & Fractals, vol. 21, no. 3, pp. 749–761, 2004.

[40] P. L’Ecuyer, “Random number generation,” in Handbook of Computational Statistics, 2nd ed. Berlin, Germany: Springer, 2012, pp. 35–71.

[41] C. E. Shannon, “Communication theory of secrecy systems,” Bell System Technical Journal, vol. 28, pp. 656–715, 1949.

[42] B. Schneier, Applied Cryptography, 2nd ed. New York, NY, USA: Wiley, 1996.

[43] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography. Boca Raton, FL, USA: CRC Press, 1996.

[44] L. Kocarev and G. Jakimoski, “Logistic map as a block encryption algorithm,” Physics Letters A, vol. 289, no. 4–5, pp. 199–206, 2001.

[45] J. Fridrich, “Symmetric ciphers based on two-dimensional chaotic maps,” International Journal of Bifurcation and Chaos, vol. 8, no. 6, pp. 1259–1284, 1998.

[46] X. Wang, L. Teng, and X. Qin, “A novel colour image encryption algorithm based on chaos,” Signal Processing, vol. 92, no. 4, pp. 1101–1108, 2012.

[47] A. Rukhin et al., A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications (NIST SP 800-22 Rev.1a), National Institute of Standards and Technology, 2010.

[48] Ezurio, “RM126x Series AT Interface Application Quick Start Guide,” 2023. Available: [https://www.ezurio.com/documentation/quick-start-guide-at-interface-application-rm126x-

series](https://www.ezurio.com/documentation/quick-start-guide-at-interface-application-rm126x-series)

[49] Ezurio, “RM126x Series AT Interface Application User Guide,” 2023. Available: https://www.ezurio.com/documentation/user-guide-rm126x-at-interface-application

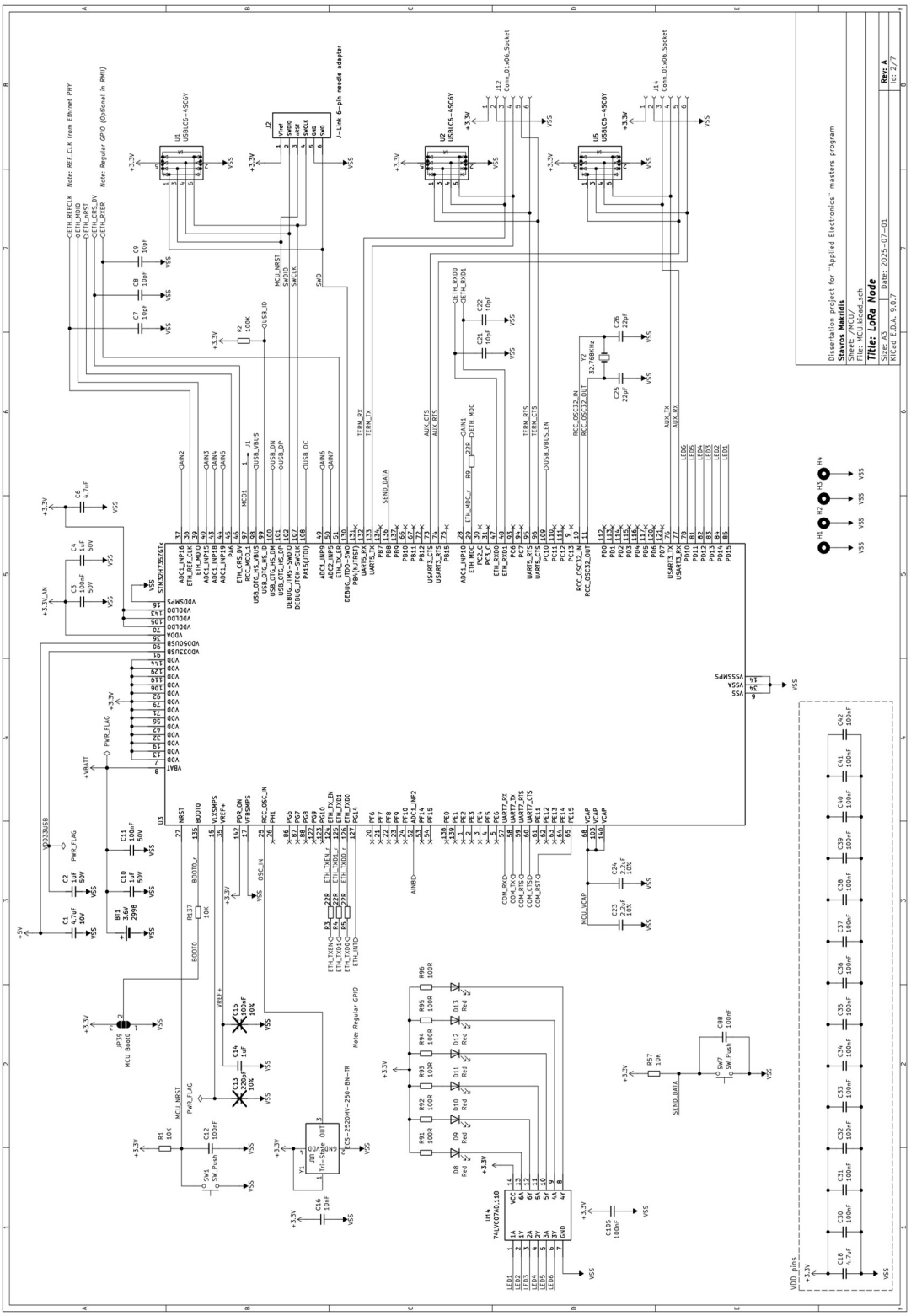
[50] G. Alvarez and S. Li, “Some basic cryptographic requirements for chaos-based cryptosystems,” International Journal of Bifurcation and Chaos, vol. 16, no. 8, pp. 2129–2151, 2006.

[51] M. Suneel, “Chaotic sequences for cryptographic applications,” arXiv preprint, arXiv:cs/0603024, 2006.

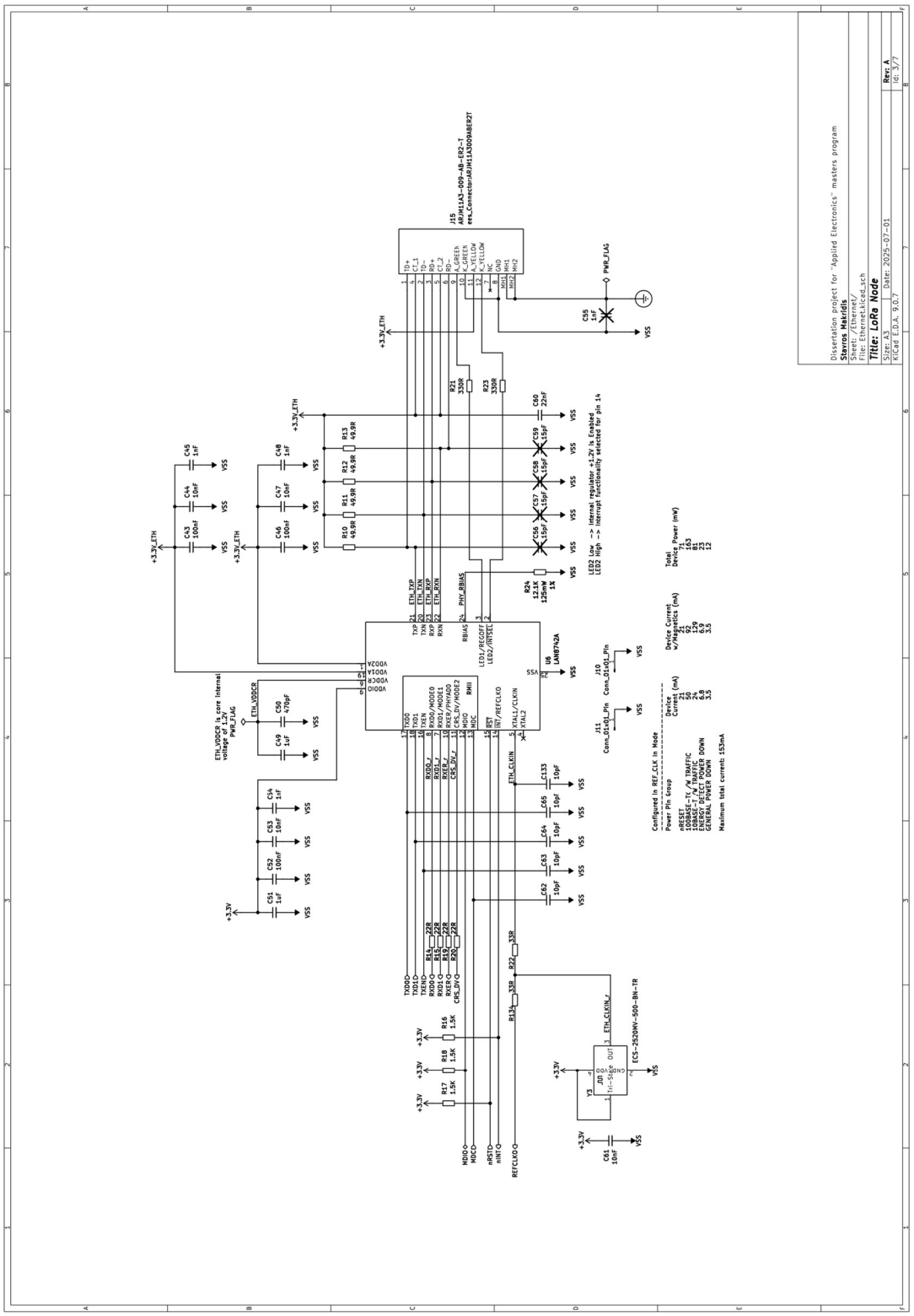
[52] S. Li, X. Mou, and Y. Cai, “Pseudo-random bit generator based on coupled chaotic systems,” International Journal of Bifurcation and Chaos, vol. 15, no. 10, pp. 3367–3377, 2005.

[53] A. Appleby, “MurmurHash3,” 2011. [Online]. Available: https://github.com/aappleby/smhasher

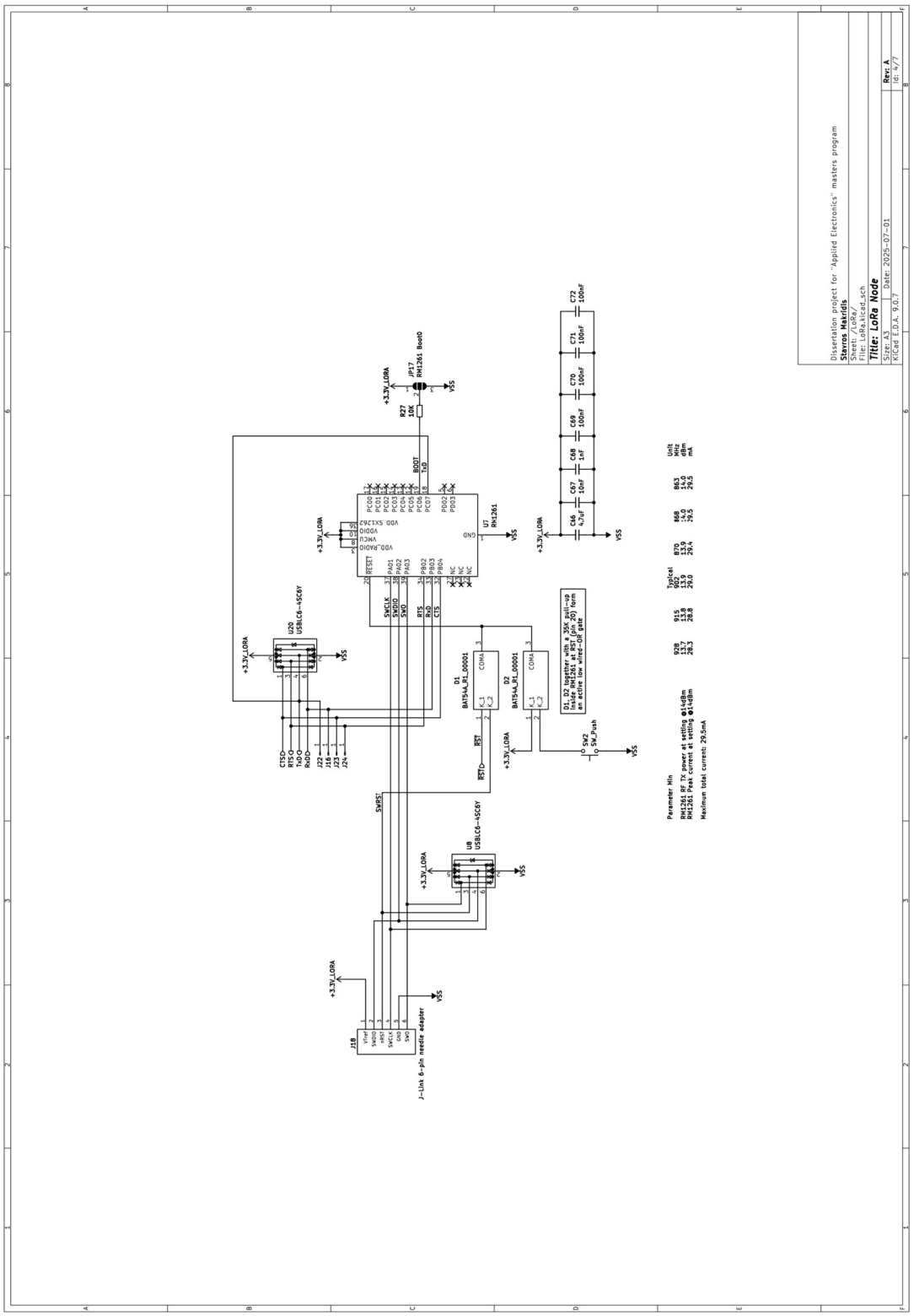
ΠΑΡΑΡΤΗΜΑ Α: Σχηματικά διαγράμματα



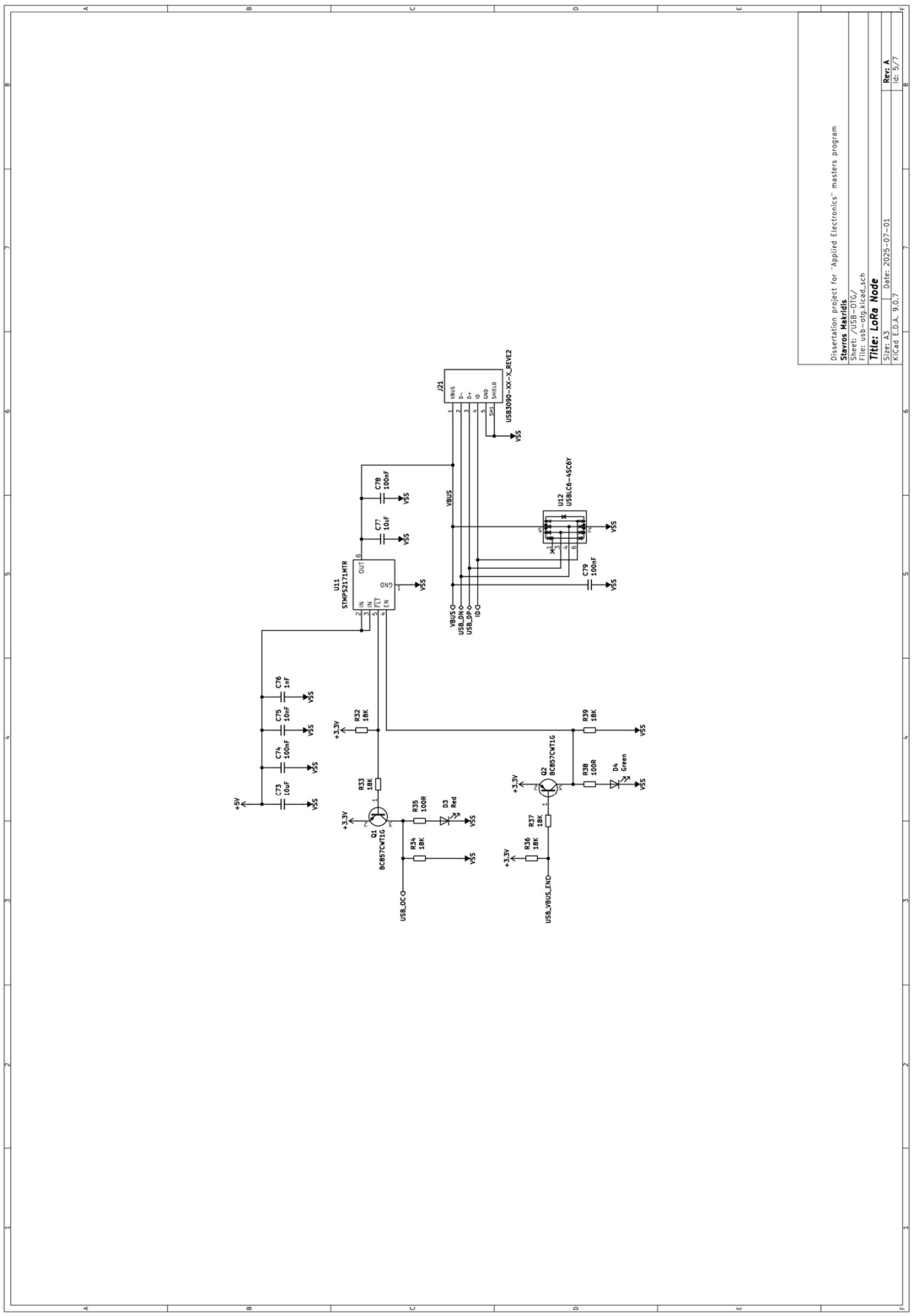
Σχήμα 28 Σχηματικό διάγραμμα τμήματος μικροελεγκτή



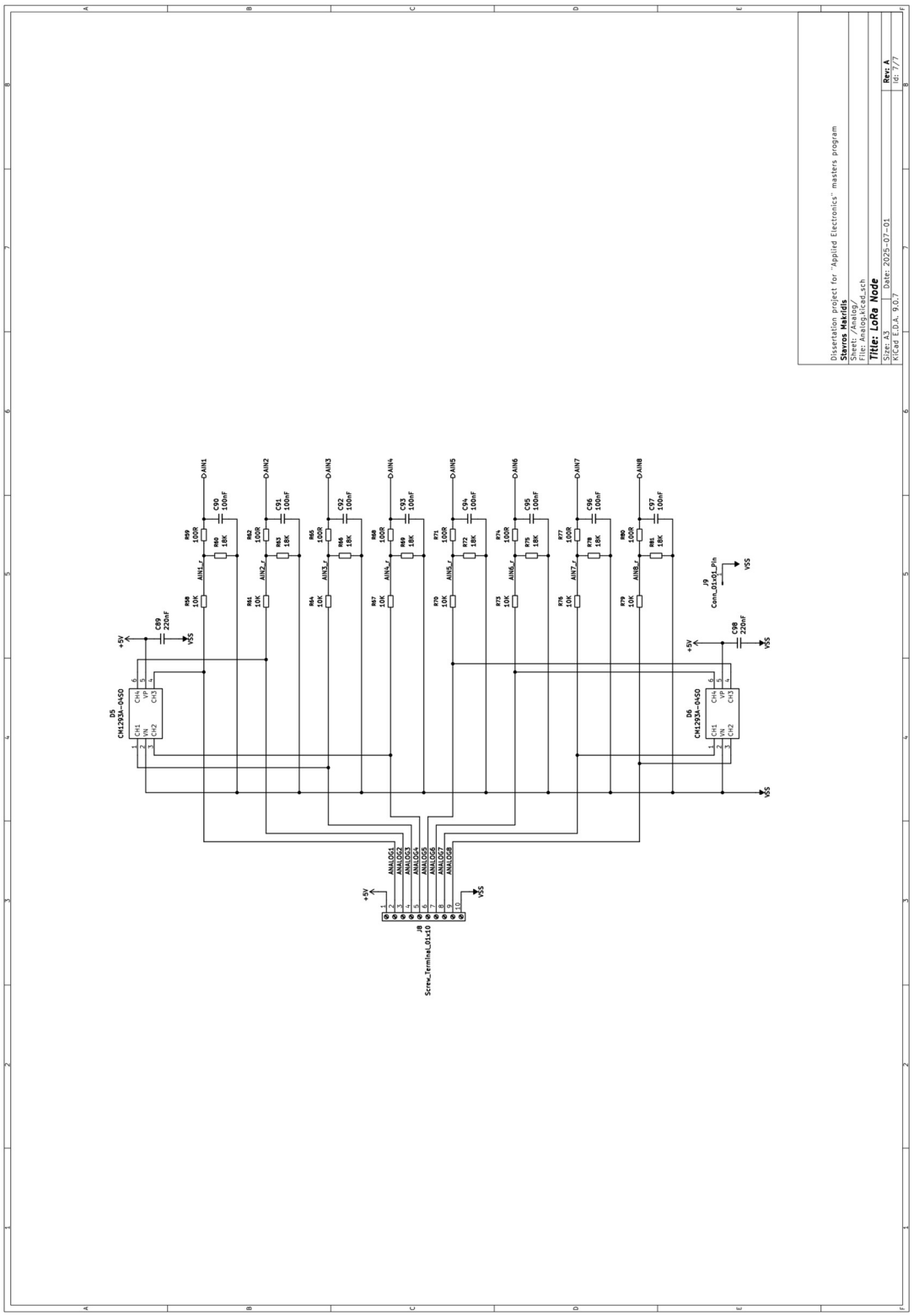
Σχήμα 29 Σχηματικό διάγραμμα τμήματος Ethernet



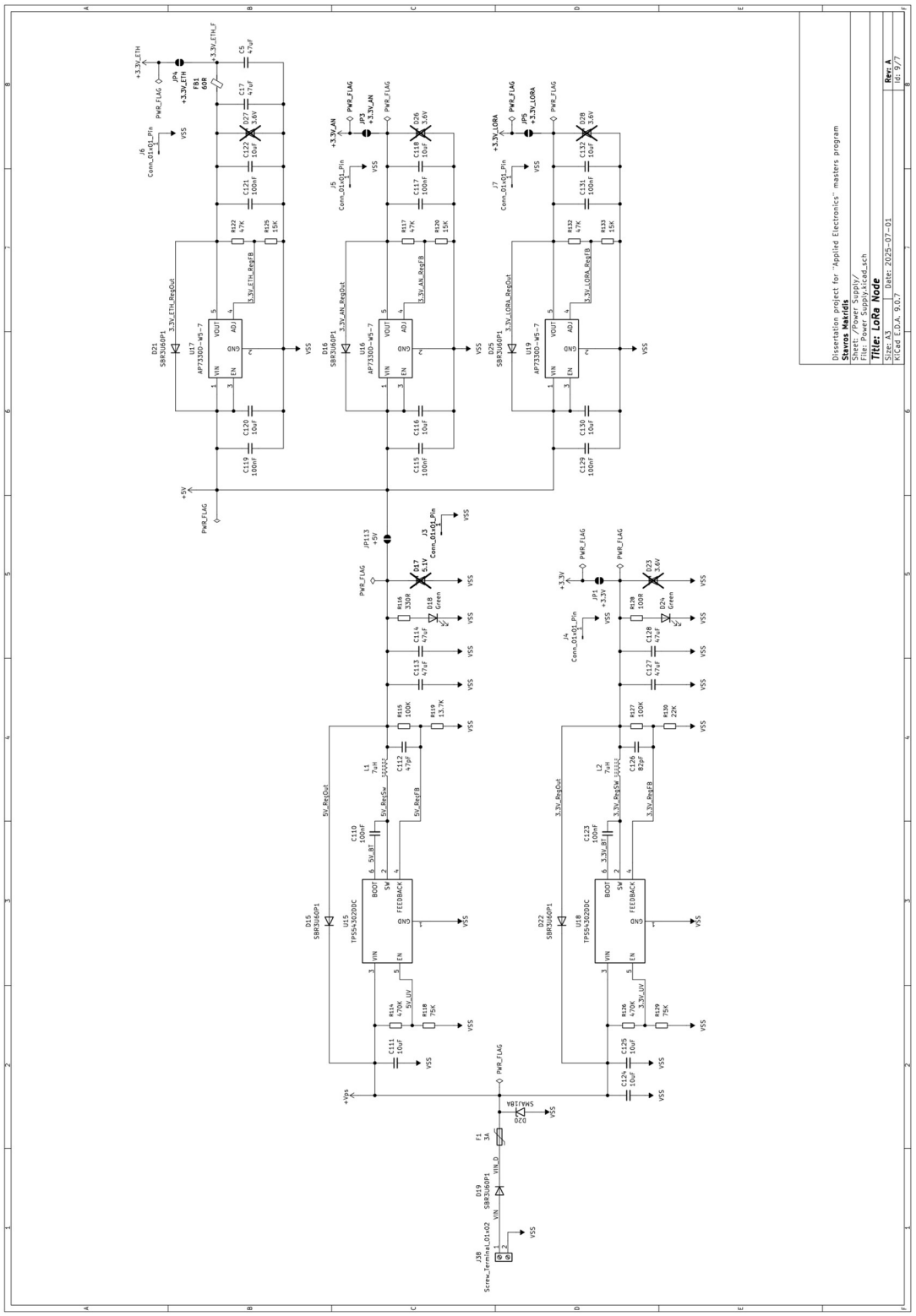
Σχήμα 30 Σχηματικό διάγραμμα τμήματος LoRa



Σχήμα 31 Σχηματικό διάγραμμα τμήματος USB



Σχήμα 32 Σχηματικό διάγραμμα τμήματος αναλογικών εισόδων



Σχήμα 33 Σχηματικό διάγραμμα τμήματος τροφοδοσίας

Λίστα Υλικών

#1 Value: 3.6V, Ref: BT1
MPN: 2998, MNF: Keystone, Qty: 1
Footprint: Battery:BatteryHolder_Keystone_2998_1x6.8mm

#2 Value: 3A, Ref: F1
MPN: 2920L300/15DR, MNF: LittleFuse, Qty: 1
Footprint: Fuse:Fuse_2920_7451Metric

#3 Value: RM1261, Ref: U7
MPN: 453-00140R, MNF: Ezurio, Qty: 1
Footprint: ees_RF_Module_LoRa:RM1261

#4 Value: 74LVC07AD,118, Ref: U14
MPN: 74LVC07AD,118, MNF: Nexperia, Qty: 1
Footprint: Package_SO:SO-14_3.9x8.65mm_P1.27mm

#5 Value: AP7330D-W5-7, Ref: U16,U17,U19
MPN: AP7330D-W5-7, MNF: Diodes Incorporated, Qty: 3
Footprint: Package_TO_SOT_SMD:SOT-23-5

#6 Value: ARJM11A3-009-AB-ER2-T, Ref: J15
MPN: ARJM11A3-009-AB-ER2-T, MNF: ABRACON, Qty: 1
Footprint: ees_Connector:ARJM11A3009ABER2T

#7 Value: BAT54A_R1_00001, Ref: D1,D2
MPN: BAT54A_R1_00001, MNF: PANJIT, Qty: 2
Footprint: Package_TO_SOT_SMD:SOT-23-3

#8 Value: BC857CWT1G, Ref: Q1,Q2
MPN: BC857CWT1G, MNF: On Semi, Qty: 2
Footprint: Package_TO_SOT_SMD:SOT-23-3

#9 Value: 3.6V, Ref: D23,D26,D27,D28
MPN: BZG05C3V6-HM3-08, MNF: Vishay, Qty: 4
Footprint: Diode_SMD:D_SMA

#10 Value: 5.1V, Ref: D17
MPN: BZG05C5V1-HM3-08, MNF: Vishay, Qty: 1
Footprint: Diode_SMD:D_SMA

#11 Value: 100nF, Ref: C110,C115,C117,C119,C121,C123,C129,C131
MPN: C0805C104M5RACTU, MNF: Kemet, Qty: 8
Footprint: Capacitor_SMD:C_0805_2012Metric

#12 Value: 47pF, Ref: C112
MPN: C0805C470J5GACTU, MNF: Kemet, Qty: 1
Footprint: Capacitor_SMD:C_0805_2012Metric

#13 Value: 1nF, Ref: C55
MPN: C1206C102KGRAC, MNF: Kemet, Qty: 1
Footprint: Capacitor_SMD:C_1206_3216Metric

#14 Value: 82pF, Ref: C126
MPN: CC0805JRNPO9BN820, MNF: Yageo, Qty: 1
Footprint: Capacitor_SMD:C_0805_2012Metric

#15 Value: 1nF, Ref: C45,C48,C54,C68,C76
MPN: CL10B102KB8WPNC, MNF: Samsung, Qty: 5
Footprint: Capacitor_SMD:C_0603_1608Metric

#16 Value: 10nF, Ref: C16,C44,C47,C53,C61,C67,C75
   MPN: CL10B103KB8NNWC, MNF: Samsung, Qty: 7
   Footprint: Capacitor_SMD:C_0603_1608Metric

#17 Value: 100nF, Ref:
C3,C11,C12,C15,C30,C31,C32,C33,C34,C35,C36,C37,C38,C39,C40,C41,C42,C43,C46,C52,
C69,C70,C71,C72,C74,C78,C79,C88,C90,C91,C92,C93,C94,C95,C96,C97,C105
   MPN: CL10B104KB8NNNC, MNF: Samsung, Qty: 37
   Footprint: Capacitor_SMD:C_0603_1608Metric

#18 Value: 220pF, Ref: C13
   MPN: CL10B221KB8NNNC, MNF: Samsung, Qty: 1
   Footprint: Capacitor_SMD:C_0603_1608Metric

#19 Value: 22nF, Ref: C60
   MPN: CL10B223KB8NNNC, MNF: Samsung, Qty: 1
   Footprint: Capacitor_SMD:C_0603_1608Metric

#20 Value: 220nF, Ref: C89,C98
   MPN: CL10B224KB8VPNC, MNF: Samsung, Qty: 2
   Footprint: Capacitor_SMD:C_0603_1608Metric

#21 Value: 10pF, Ref: C7,C8,C9,C21,C22,C62,C63,C64,C65,C133
   MPN: CL10C100JB8NNND, MNF: Samsung, Qty: 10
   Footprint: Capacitor_SMD:C_0603_1608Metric

#22 Value: 15pF, Ref: C56,C57,C58,C59
   MPN: CL10C150JB8NNND, MNF: Samsung, Qty: 4
   Footprint: Capacitor_SMD:C_0603_1608Metric

#23 Value: 22pF, Ref: C25,C26
   MPN: CL10C220JB8NNND, MNF: Samsung, Qty: 2
   Footprint: Capacitor_SMD:C_0603_1608Metric

#24 Value: 470pF, Ref: C50
   MPN: CL10C471JB8NFNC, MNF: Samsung, Qty: 1
   Footprint: Capacitor_SMD:C_0603_1608Metric

#25 Value: 1uF, Ref: C2,C4,C10,C14,C49,C51
   MPN: CL21B105KBFNNNE, MNF: Samsung, Qty: 6
   Footprint: Capacitor_SMD:C_0805_2012Metric

#26 Value: 2.2uF, Ref: C23,C24
   MPN: CL21B225KPFNNNG, MNF: Samsung, Qty: 2
   Footprint: Capacitor_SMD:C_0805_2012Metric

#27 Value: 4.7uF, Ref: C1,C6,C18,C66
   MPN: CL31B475KPHNNNE, MNF: Samsung, Qty: 4
   Footprint: Capacitor_SMD:C_1206_3216Metric

#28 Value: 10uF, Ref: C73,C77,C111,C116,C118,C120,C122,C124,C125,C130,C132
   MPN: CL32A106KBJNNNE, MNF: Samsung, Qty: 11
   Footprint: Capacitor_SMD:C_1210_3225Metric

#29 Value: CM1293A-04SO, Ref: D5,D6
   MPN: CM1293A-04SO, MNF: ON Semi, Qty: 2
   Footprint: Package_TO_SOT_SMD:SOT-23-6

#30 Value: 100K, Ref: R2,R115,R127
   MPN: CRCW0603100KFKEA, MNF: Vishay, Qty: 3
   Footprint: Resistor_SMD:R_0603_1608Metric

#31 Value: 100R, Ref: R35,R38,R59,R62,R65,R68,R71,R74,R77,R80,R91,R92,R93,R94,R95,R96,R128
   MPN: CRCW0603100RFKEBC, MNF: Vishay, Qty: 17
   Footprint: Resistor_SMD:R_0603_1608Metric

#32 Value: 330R, Ref: R21,R23,R116
   MPN: CRCW0603330RFKEBC, MNF: Vishay, Qty: 3
   Footprint: Resistor_SMD:R_0603_1608Metric

#33 Value: 470K, Ref: R114,R126
   MPN: CRCW0603470KFKEA, MNF: Vishay, Qty: 2
   Footprint: Resistor_SMD:R_0603_1608Metric

#34 Value: 10K, Ref: R1,R27,R57,R58,R61,R64,R67,R70,R73,R76,R79,R137
   MPN: CRCW060310K0FKEE, MNF: Vishay, Qty: 12
   Footprint: Resistor_SMD:R_0603_1608Metric

#35 Value: 13.7K, Ref: R119
   MPN: CRCW060313K7FKEA, MNF: Vishay, Qty: 1
   Footprint: Resistor_SMD:R_0603_1608Metric

#36 Value: 18K, Ref: R32,R33,R34,R36,R37,R39,R60,R63,R66,R69,R72,R75,R78,R81,R120,R125,R133
   MPN: CRCW060318K0FKEAC, MNF: Vishay, Qty: 17
   Footprint: Resistor_SMD:R_0603_1608Metric

#37 Value: 1.5K, Ref: R16,R17,R18
   MPN: CRCW06031K50FKEA, MNF: Vishay, Qty: 3
   Footprint: Resistor_SMD:R_0603_1608Metric

#38 Value: 22K, Ref: R130
   MPN: CRCW060322K0FKEA, MNF: Vishay, Qty: 1
   Footprint: Resistor_SMD:R_0603_1608Metric

#39 Value: 22R, Ref: R3,R4,R5,R9,R14,R15,R19,R20
   MPN: CRCW060322R0FKEAC, MNF: Vishay, Qty: 8
   Footprint: Resistor_SMD:R_0603_1608Metric

#40 Value: 33R, Ref: R22,R134
   MPN: CRCW060333R0FKEAC, MNF: Vishay, Qty: 2
   Footprint: Resistor_SMD:R_0603_1608Metric

#41 Value: 47K, Ref: R117,R122,R132
   MPN: CRCW060347K0FKEA, MNF: Vishay, Qty: 3
   Footprint: Resistor_SMD:R_0603_1608Metric

#42 Value: 49.9R, Ref: R10,R11,R12,R13
   MPN: CRCW060349R9FKEAC, MNF: Vishay, Qty: 4
   Footprint: Resistor_SMD:R_0603_1608Metric

#43 Value: 75K, Ref: R118,R129
   MPN: CRCW060375K0FKEA, MNF: Vishay, Qty: 2
   Footprint: Resistor_SMD:R_0603_1608Metric

#44 Value: 12.1K, Ref: R24
   MPN: CRCW080512K1FKEAC, MNF: Vishay, Qty: 1
   Footprint: Resistor_SMD:R_0805_2012Metric

#45 Value: 7uH, Ref: L1,L2
   MPN: DR1040-7R0-R, MNF: Eaton, Qty: 2
   Footprint: ees_Inductor_SMD:L_Eaton_DR1040

#46 Value: 32.768KHz, Ref: Y2
   MPN: ECS-.327-12.5-16-TR3, MNF: ECS Inc., Qty: 1
   Footprint: ees_Crystal:XTAL_ECS-16x10

#47 Value: ECS-2520MV-250-BN-TR, Ref: Y1
   MPN: ECS-2520MV-250-BN-TR, MNF: ECS, Qty: 1
   Footprint: Oscillator:Oscillator_SMD_ECS_2520MV-xxx-xx-4Pin_2.5x2.0mm

#48 Value: ECS-2520MV-500-BN-TR, Ref: Y3
   MPN: ECS-2520MV-500-BN-TR, MNF: ECS, Qty: 1
   Footprint: Oscillator:Oscillator_SMD_ECS_2520MV-xxx-xx-4Pin_2.5x2.0mm

#49 Value: SW_Push, Ref: SW1,SW2,SW7
   MPN: FSMSM, MNF: TE Connectivity, Qty: 3
   Footprint: ees_Button_Switch_SMD:FSMSM

#50 Value: 47uF, Ref: C5,C17,C113,C114,C127,C128
   MPN: GRM31CR61A476KE15L, MNF: MuRata, Qty: 6
   Footprint: Capacitor_SMD:C_1206_3216Metric

#51 Value: LAN8742A, Ref: U6
   MPN: LAN8742A-CZ, MNF: Microchip, Qty: 1
   Footprint: Package_DFN_QFN:VQFN-24-1EP_4x4mm_P0.5mm_EP2.5x2.5mm_ThermalVias

#52 Value: Green, Ref: D4,D18,D24
   MPN: LTST-C190KGKT, MNF: LiteOn, Qty: 3
   Footprint: LED_SMD:LED_0603_1608Metric

#53 Value: Red, Ref: D3,D8,D9,D10,D11,D12,D13
   MPN: LTST-C190KRKT, MNF: LiteOn, Qty: 7
   Footprint: LED_SMD:LED_0603_1608Metric

#54 Value: 60R, Ref: FB1
   MPN: MFBM1V3216-600-R, MNF: Eaton, Qty: 1
   Footprint: Inductor_SMD:L_1206_3216Metric

#55 Value: SBR3U60P1, Ref: D15,D16,D19,D21,D22,D25
   MPN: SBR3U60P1-7, MNF: Diodes, Qty: 6
   Footprint: Diode_SMD:D_PowerDI-123

#56 Value: SMAJ18A, Ref: D20
   MPN: SMAJ28A, MNF: LittleFuse, Qty: 1
   Footprint: Diode_SMD:D_SMA

#57 Value: STM32H735ZGTx, Ref: U3
   MPN: STM32H735ZGT6, MNF: ST, Qty: 1
   Footprint: Package_QFP:LQFP-144_20x20mm_P0.5mm

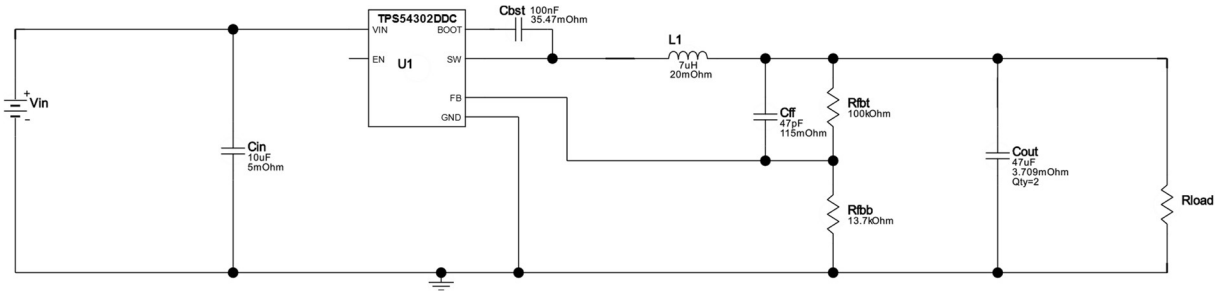
#58 Value: STMPS2171MTR, Ref: U11
   MPN: STMPS2171MTR, MNF: ST, Qty: 1
   Footprint: Package_SO:SOIC-8_3.9x4.9mm_P1.27mm

#59 Value: TPS54302DDC, Ref: U15,U18
   MPN: TPS54302DDC, MNF: Texas Instruments, Qty: 2
   Footprint: Package_TO_SOT_SMD:SOT-23-6

#60 Value: USB3090-XX-X_REVE2, Ref: J21
   MPN: USB3090-30-A, MNF: Global Connector Technology, Qty: 1
   Footprint: ees_Connector_USB:GCT_USB3090-XX-X_REVE2

#61 Value: USBLC6-4SC6Y, Ref: U1,U2,U5,U8,U12,U20
   MPN: USBLC6-4SC6Y, MNF: ST, Qty: 6
   Footprint: Package_TO_SOT_SMD:SOT-23-6

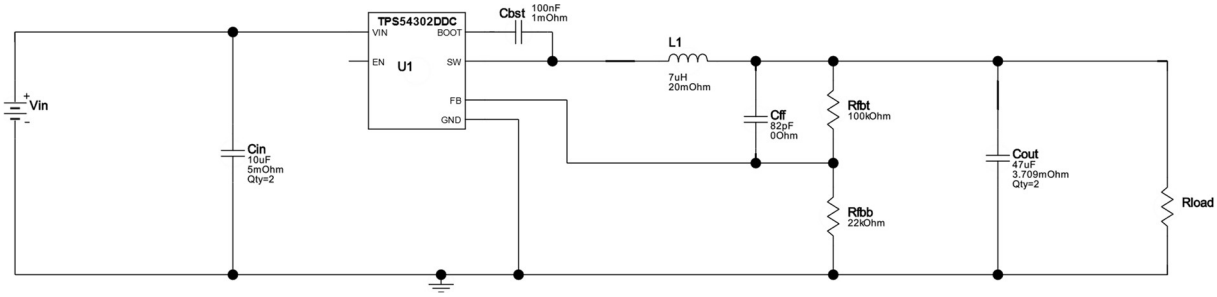
ΠΑΡΑΡΤΗΜΑ Β: Σχεδίαση παλμοσταθεροποιητών



Σχήμα 34 Σχηματικό διάγραμμα παλμοσταθεροποιητή 12V/5V (TI's Webench Power Designer)

Πίνακας 13 Λίστα υλικών παλμοσταθεροποιητή 12V/5V

Part	Manufacturer	Part Number	Quantity
Cbst	Kemet	C0805C104M5RACTU	1
Cin	Samsung Electro-Mechanics	CL32A106KBJNNNE	1
L1	Coiltronics	DR1040-7R0-R	1
Rfbb	Vishay-Dale	CRCW060313K7FKEA	1
Rfbt	Vishay-Dale	CRCW0603100KFKEA	1
U1	Texas Instruments	TPS54302DDCR	1
Cout	MuRata	GRM31CR61A476KE15L	2
Cff	Kemet	C0805C470J5GACTU	1



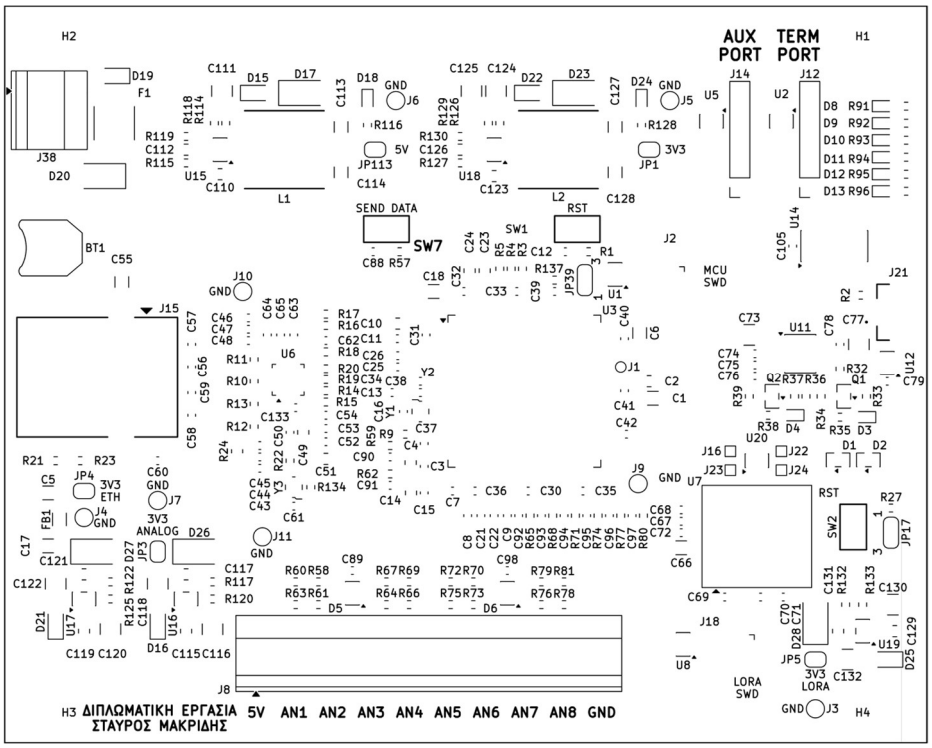
Σχήμα 35 Σχηματικό διάγραμμα παλμοσταθεροποιητή 12V/3,3V (TI's Webench Power Designer)

Πίνακας 14 Λίστα υλικών παλμοσταθεροποιητή 12V/3,3V

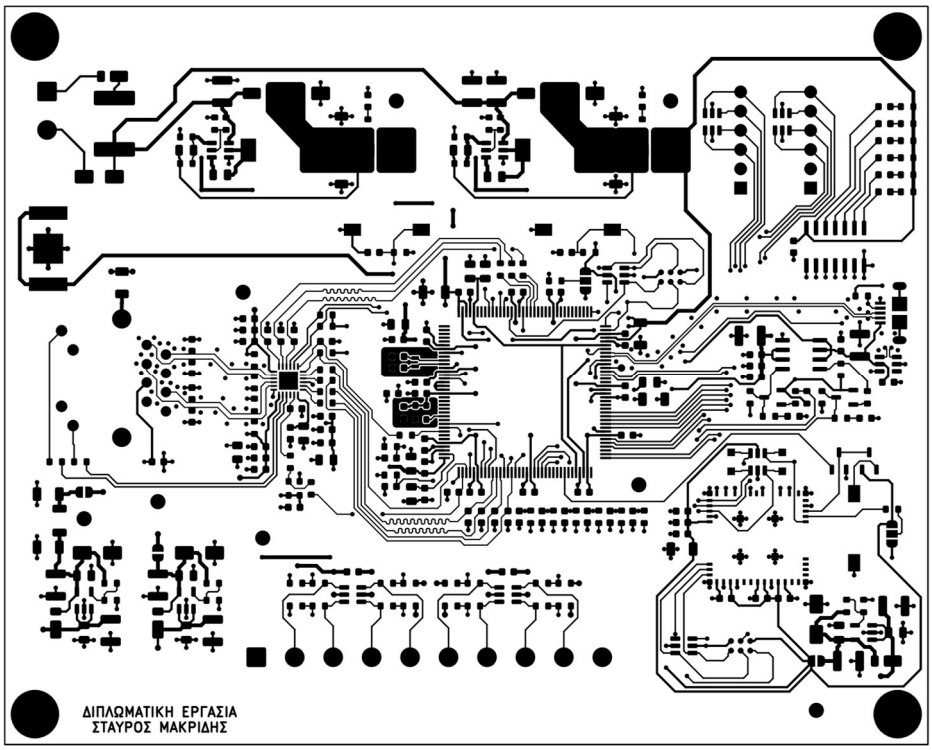
Part	Manufacturer	Part Number	Quantity
U1	Texas Instruments	TPS54302DDCR	1
Cin	Samsung Electro-Mechanics	CL32A106KBJNNNE	2
L1	Coiltronics	DR1040-7R0-R	1
Rfbb	Yageo	RC0603FR-0722KL	1

Cbst	Yageo	CC0805KRX7R9BB104	1
Cout	MuRata	GRM31CR61A476KE15L	2
Rfbt	Vishay-Dale	CRCW0603100KFKEA	1
Cff	Yageo	CC0805JRNPO9BN820	1

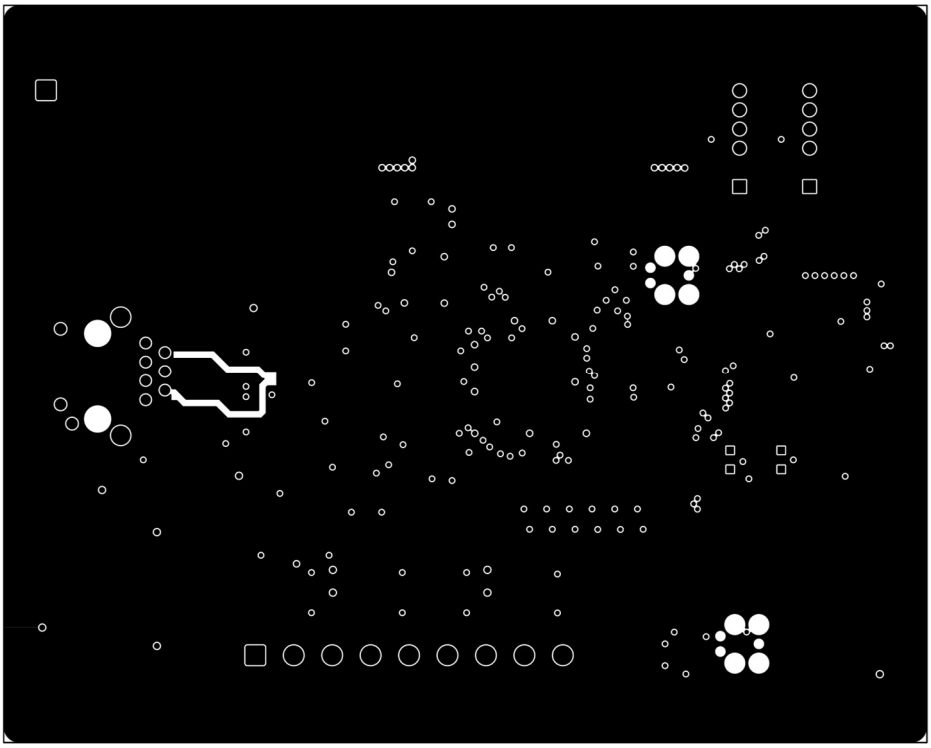
ΠΑΡΑΡΤΗΜΑ Γ: Αρχεία παραγωγής τυπωμένου κυκλώματος



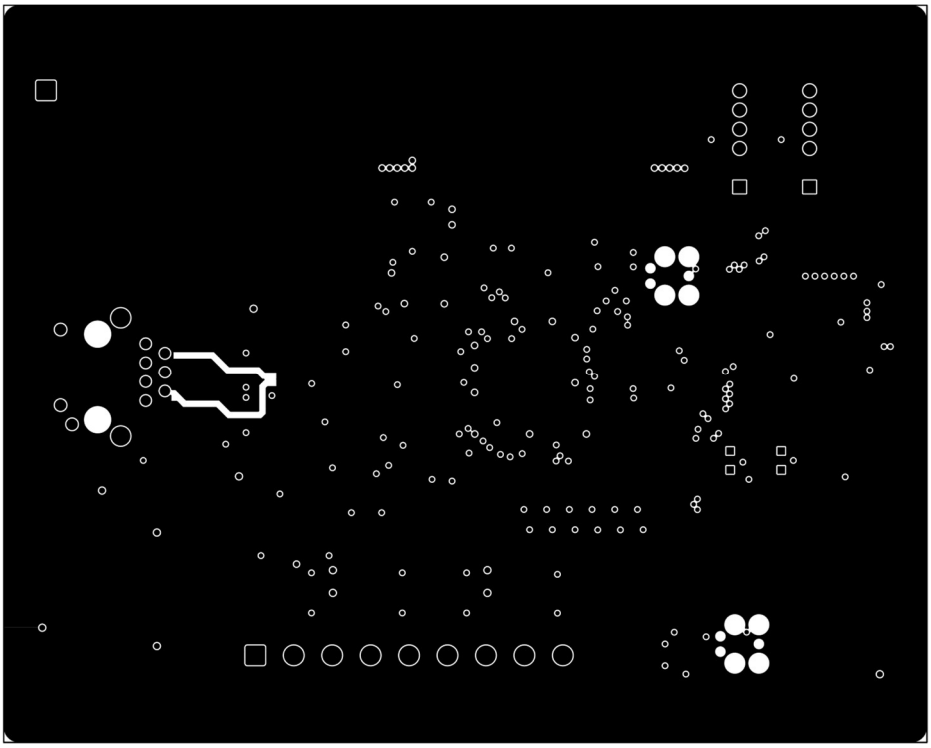
Σχήμα 36 Silkscreen Top



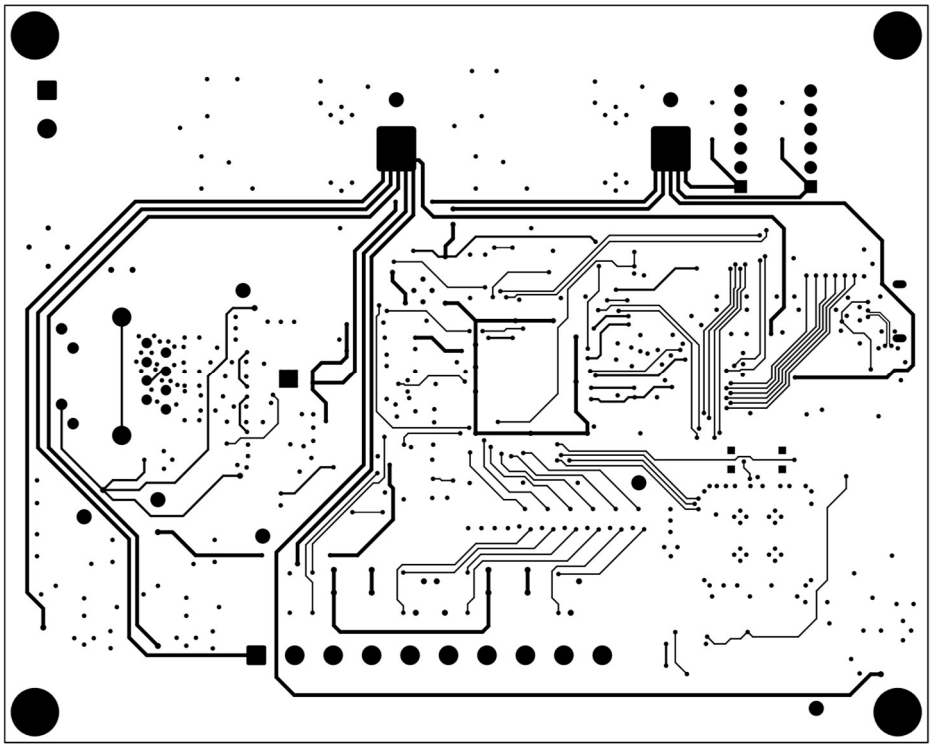
Σχήμα 37 Top Copper (L1)



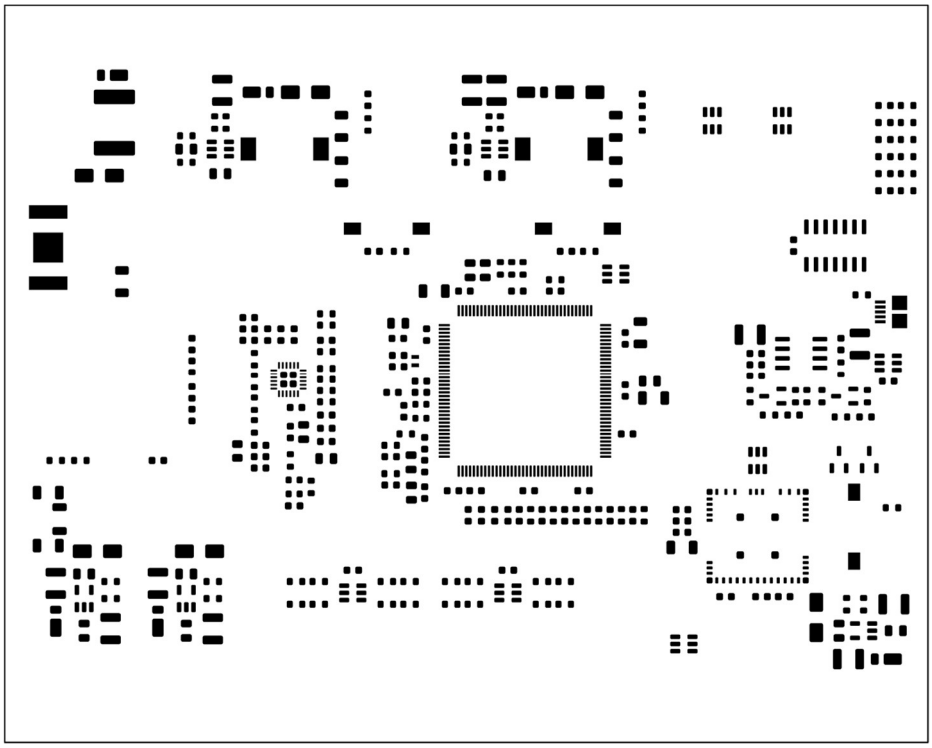
Σχήμα 38 Inner Copper 1 (L2)



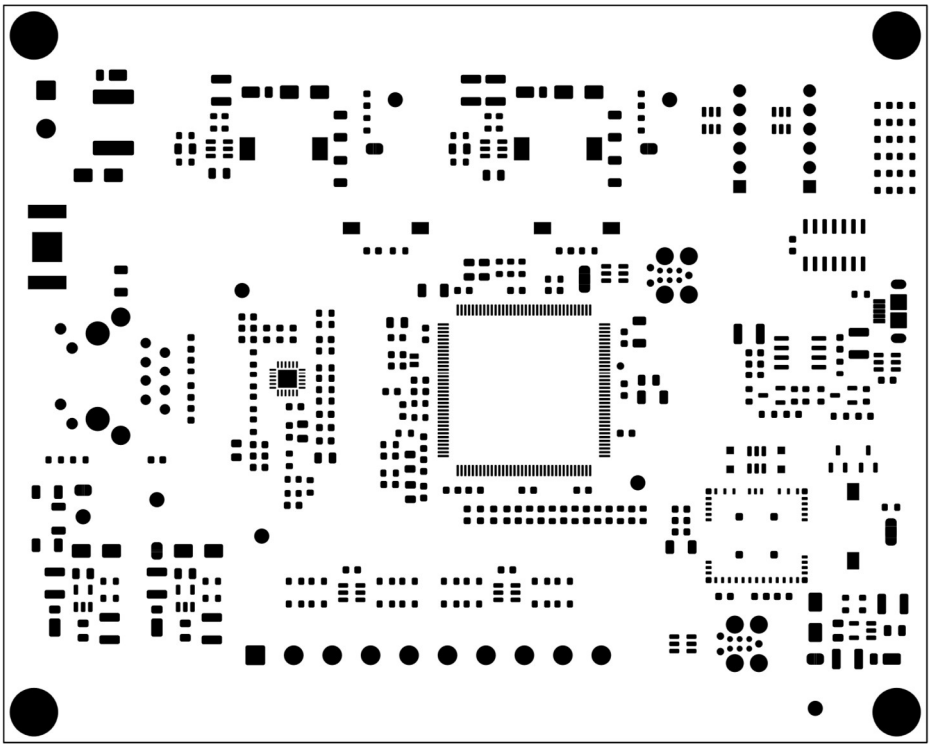
Σχήμα 39 Inner Copper 2 (L3)



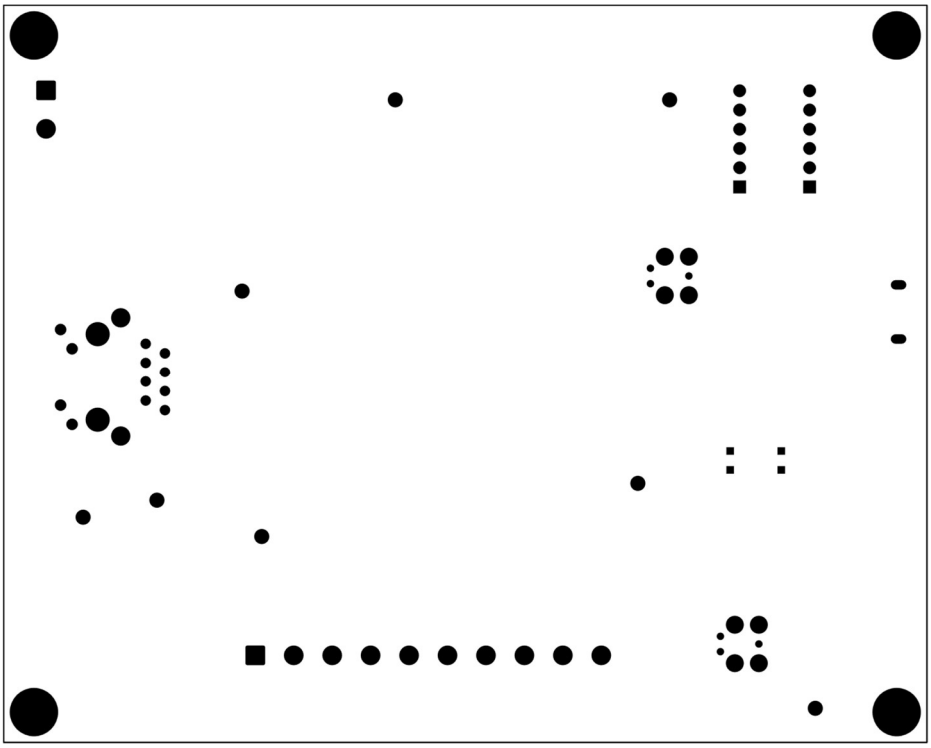
Σχήμα 40 Bottom Copper (L4)



Σχήμα 41 Top Paste Mask



Σχήμα 42 Top Solder Mask



Σχήμα 43 Bottom Solder Mask

ΠΑΡΑΡΤΗΜΑ Δ: Προγράμματα δημιουργίας γραφημάτων

```
#
# ch1_LogisticMap_Sensitivity_InitialConditions.py
#

import numpy as np
import matplotlib.pyplot as plt

# Logistic map parameters
r = 3.99
iterations = 200

# Initial conditions
x0_1 = 0.3000
x0_2 = 0.3001

def logistic_map(r, x0, n):
    x = np.zeros(n)
    x[0] = x0
    for i in range(1, n):
        x[i] = r * x[i - 1] * (1 - x[i - 1])
    return x

# Generate sequences
x1 = logistic_map(r, x0_1, iterations)
x2 = logistic_map(r, x0_2, iterations)

# Plot
plt.figure(figsize=(12, 6))
plt.plot(x1, label=f"x₀ = {x0_1:.4f}", linewidth=1.5)
plt.plot(x2, "--", label=f"x₀ = {x0_2:.4f}", linewidth=1.5)

plt.title("Sensitivity to Initial Conditions – Logistic Map (r = 3.99)")
plt.xlabel("Iteration")
plt.ylabel("State value")
plt.legend()
plt.grid(True, linestyle="--", alpha=0.6)

plt.tight_layout()
plt.show()
```

```
#
# ch2_logistic_fp32_vs_fp64_histogram.py
#

import numpy as np
import matplotlib.pyplot as plt

# Logistic map FP32 vs FP64 histogram (state value distribution)
# Produces a 1x2 figure: left FP32, right FP64.

r = 3.99
iterations = 1_000_000
discard = 2000
x0 = 0.123456
bins = 120


def logistic_fp32_series(r=3.99, x0=0.123456, iterations=1_000_000, discard=2000):
    r32 = np.float32(r)
    x = np.float32(x0)
    out = []
    for i in range(iterations):
        x = r32 * x * (np.float32(1.0) - x)
        if i > discard:
            out.append(float(x))
    return out


def logistic_fp64_series(r=3.99, x0=0.123456, iterations=1_000_000, discard=2000):
    x = float(x0)
    out = []
    for i in range(iterations):
        x = r * x * (1.0 - x)
        if i > discard:
            out.append(x)
    return out


fp32_vals = logistic_fp32_series(r=r, x0=x0, iterations=iterations, discard=discard)
fp64_vals = logistic_fp64_series(r=r, x0=x0, iterations=iterations, discard=discard)

plt.figure(figsize=(10, 4))

plt.subplot(1, 2, 1)
plt.hist(fp32_vals, bins=bins, density=True)
plt.title("Logistic Map Histogram (FP32)")
plt.xlabel("State value")
plt.ylabel("Probability density")

plt.subplot(1, 2, 2)
plt.hist(fp64_vals, bins=bins, density=True)
plt.title("Logistic Map Histogram (FP64)")
plt.xlabel("State value")

plt.tight_layout()
plt.show()
```

```
#
# ch2_tent_map_fp32_histogram.py
#

import numpy as np
import matplotlib.pyplot as plt

# Tent map FP32 histogram (state value distribution)
# Focuses on FP32 because that is the embedded implementation regime.

mu = 2.0
iterations = 1_000_000
discard = 2000
x0 = 0.312345
bins = 120

def tent_step_fp32(x: np.float32, mu32: np.float32) -> np.float32:
    if x < np.float32(0.5):
        return mu32 * x
    return mu32 * (np.float32(1.0) - x)

mu32 = np.float32(mu)
x = np.float32(x0)
vals = []

for i in range(iterations):
    x = tent_step_fp32(x, mu32)
    if i > discard:
        vals.append(float(x))

plt.figure(figsize=(5.5, 4))
plt.hist(vals, bins=bins, density=True)
plt.title("Tent Map Histogram (FP32)")
plt.xlabel("State value")
plt.ylabel("Probability density")
plt.tight_layout()
plt.show()
```

```
#
# ch2_hybrid_state_histograms_fp32_fp64.py
#

import numpy as np
import matplotlib.pyplot as plt

# Hybrid (logistic + tent) state histograms:
# - Logistic FP32 baseline
# - Hybrid FP32
# - Hybrid FP64

iterations = 1_000_000
discard = 2000
bins = 120

def logistic_fp32_series(r=3.99, x0=0.123456, iterations=1_000_000, discard=2000):
    r32 = np.float32(r)
    x = np.float32(x0)
    out = []
    for i in range(iterations):
        x = r32 * x * (np.float32(1.0) - x)
        if i > discard:
            out.append(float(x))
    return out

def hybrid_step_fp32(
    xl,
    xt,
    r=np.float32(3.99),
    mu=np.float32(2.0),
    eps=np.float32(1e-6),
    ctr=np.uint32(0),
):
    xl_next = r * xl * (np.float32(1.0) - xl)
    if xt < np.float32(0.5):
        xt_next = mu * xt
    else:
        xt_next = mu * (np.float32(1.0) - xt)

    xl_c = (xl_next + np.float32(0.07) * xt_next).astype(np.float32)
    xt_c = (xt_next + np.float32(0.05) * xl_next).astype(np.float32)

    noise_u = np.float32((ctr & np.uint32(0xFFFF)) / np.float32(65536.0))
    noise = eps * noise_u

    xl_out = np.mod(xl_c + noise, np.float32(1.0)).astype(np.float32)
    xt_out = np.mod(xt_c + noise, np.float32(1.0)).astype(np.float32)

    xl_out = np.maximum(xl_out, np.float32(np.finfo(np.float32).tiny))
    xt_out = np.maximum(xt_out, np.float32(np.finfo(np.float32).tiny))
    return xl_out, xt_out

def hybrid_step_fp64(xl, xt, r=3.99, mu=2.0, eps=1e-12, ctr=0):
    xl_next = r * xl * (1.0 - xl)
    xt_next = mu * xt if xt < 0.5 else mu * (1.0 - xt)

    xl_c = xl_next + 0.07 * xt_next
    xt_c = xt_next + 0.05 * xl_next

    noise = eps * ((ctr & 0xFFFF) / 65536.0)

    xl_out = (xl_c + noise) % 1.0
    xt_out = (xt_c + noise) % 1.0

    tiny = np.finfo(np.float64).tiny
```

```
    xl_out = max(xl_out, tiny)
    xt_out = max(xt_out, tiny)
    return xl_out, xt_out

def hybrid_series_fp32(
    xl0=0.123456, xt0=0.654321, iterations=1_000_000, discard=2000, eps=1e-6
):
    xl = np.float32(xl0)
    xt = np.float32(xt0)
    out = []
    ctr = np.uint32(0)
    for i in range(iterations):
        ctr = np.uint32(ctr + 1)
        xl, xt = hybrid_step_fp32(xl, xt, eps=np.float32(eps), ctr=ctr)
        if i > discard:
            out.append(float(xl))
    return out

def hybrid_series_fp64(
    xl0=0.123456, xt0=0.654321, iterations=1_000_000, discard=2000, eps=1e-12
):
    xl = float(xl0)
    xt = float(xt0)
    out = []
    ctr = 0
    for i in range(iterations):
        ctr += 1
        xl, xt = hybrid_step_fp64(xl, xt, eps=eps, ctr=ctr)
        if i > discard:
            out.append(xl)
    return out

log_fp32 = logistic_fp32_series()
hyb_fp32 = hybrid_series_fp32()
hyb_fp64 = hybrid_series_fp64()

plt.figure(figsize=(12, 4))

plt.subplot(1, 3, 1)
plt.hist(log_fp32, bins=bins, density=True)
plt.title("Logistic Histogram (FP32)")
plt.xlabel("State value")
plt.ylabel("Probability density")

plt.subplot(1, 3, 2)
plt.hist(hyb_fp32, bins=bins, density=True)
plt.title("Hybrid Histogram (FP32)")
plt.xlabel("State value")

plt.subplot(1, 3, 3)
plt.hist(hyb_fp64, bins=bins, density=True)
plt.title("Hybrid Histogram (FP64)")
plt.xlabel("State value")

plt.tight_layout()
plt.show()
```

```
#
# ch2_hybrid_prng_byte_histogram.py
#

import numpy as np
import matplotlib.pyplot as plt

# Hybrid PRNG byte histogram from mixed 32-bit words (FP32)
# Produces a bar chart for byte values 0..255.

def float_to_u32_bits_fp32(x_fp32: np.float32) -> np.uint32:
    return x_fp32.view(np.uint32)

def mix32(x: np.uint32) -> np.uint32:
    x = np.uint32(x ^ (x >> np.uint32(16)))
    x = np.uint32((x * np.uint32(0x7FEB352D)) & np.uint32(0xFFFFFFFF))
    x = np.uint32(x ^ (x >> np.uint32(15)))
    x = np.uint32((x * np.uint32(0x846CA68B)) & np.uint32(0xFFFFFFFF))
    x = np.uint32(x ^ (x >> np.uint32(16)))
    return x

def hybrid_step_fp32(
    xl,
    xt,
    r=np.float32(3.99),
    mu=np.float32(2.0),
    eps=np.float32(1e-6),
    ctr=np.uint32(0),
):
    xl_next = r * xl * (np.float32(1.0) - xl)
    if xt < np.float32(0.5):
        xt_next = mu * xt
    else:
        xt_next = mu * (np.float32(1.0) - xt)

    xl_c = (xl_next + np.float32(0.07) * xt_next).astype(np.float32)
    xt_c = (xt_next + np.float32(0.05) * xl_next).astype(np.float32)

    noise_u = np.float32((ctr & np.uint32(0xFFFF)) / np.float32(65536.0))
    noise = eps * noise_u

    xl_out = np.mod(xl_c + noise, np.float32(1.0)).astype(np.float32)
    xt_out = np.mod(xt_c + noise, np.float32(1.0)).astype(np.float32)

    xl_out = np.maximum(xl_out, np.float32(np.finfo(np.float32).tiny))
    xt_out = np.maximum(xt_out, np.float32(np.finfo(np.float32).tiny))
    return xl_out, xt_out

def generate_hybrid_words_fp32(
    num_words=300_000, discard=5000, eps=1e-6, xl0=0.123456, xt0=0.654321
):
    xl = np.float32(xl0)
    xt = np.float32(xt0)
    ctr = np.uint32(0)

    for _ in range(discard):
        ctr = np.uint32(ctr + 1)
        xl, xt = hybrid_step_fp32(xl, xt, eps=np.float32(eps), ctr=ctr)
    words = np.zeros(num_words, dtype=np.uint32)
    for i in range(num_words):
        ctr = np.uint32(ctr + 1)
        xl, xt = hybrid_step_fp32(xl, xt, eps=np.float32(eps), ctr=ctr)
```

```
        w = (
            float_to_u32_bits_fp32(x1)
            ^ (float_to_u32_bits_fp32(xt) << np.uint32(1))
            ^ ctr
        )
        words[i] = mix32(w)

    return words

def byte_histogram(words_u32: np.ndarray):
    b = words_u32.view(np.uint8)
    counts = np.bincount(b, minlength=256)
    return counts / counts.sum()

probs = byte_histogram(generate_hybrid_words_fp32())

plt.figure(figsize=(10, 3))
plt.bar(np.arange(256), probs)
plt.xlabel("Byte value")
plt.ylabel("Relative frequency")
plt.title("Hybrid PRNG Byte Histogram (FP32, mixed 32-bit words)")
plt.tight_layout()
plt.show()
```

ΠΑΡΑΡΤΗΜΑ Ε: Πρόγραμμα καταγραφής επικοινωνίας

```
"""
comm_correlator.py

2x UART correlator + Chaotic decrypt verification (bit-exact with your firmware):
- Reads MCU TX log port: extracts ctr, raw16, ct16
- Reads RX receiver port (RM1261): extracts RSSI/SNR + received data (hex)
- Correlates by ctr primarily; falls back to ct matching if needed
- Decrypts RX ciphertext using reseed-per-packet (seed = ctr, seed2 = seed ^ A5A5A5A5)
- Builds unified log and prints end summary

Dependencies: pyserial, numpy
pip install pyserial numpy
"""

from __future__ import annotations

import argparse
import re
import time
import threading
from dataclasses import dataclass, field
from typing import Dict, Optional, List, Tuple

import serial
import numpy as np
import struct

# ============================================================
#  Chaotic decrypt (bit-exact with CryptoChaotic)
# ============================================================

INV_2_32_F32 = np.float32(2.3283064365386963e-10)
A5 = 0xA5A5A5A5
GOLDEN_INC = 0x9E3779B9

def u32(x: int) -> np.uint32:
    return np.uint32(x & 0xFFFFFFFF)

def rotl32(x: int, r: int) -> np.uint32:
    x &= 0xFFFFFFFF
    return np.uint32(((x << r) | (x >> (32 - r))) & 0xFFFFFFFF)

def float_to_u32_bits_f32(x: np.float32) -> np.uint32:
    return np.uint32(struct.unpack("<I", struct.pack("<f", float(x)))[0])

def seed_to_unit_open(seed_u32: np.uint32) -> np.float32:
    s = np.uint32(seed_u32)
    if int(s) == 0:
        s = np.uint32(1)
    x = (np.float32(s) + np.float32(0.5)) * INV_2_32_F32
    if x <= np.float32(0.0):
        x = np.float32(1.0e-7)
    elif x >= np.float32(1.0):
        x = np.float32(1.0 - 1.0e-7)
    return np.float32(x)

def logistic_next(x: np.float32, r: np.float32) -> np.float32:
    xn1 = np.float32(r * x * (np.float32(1.0) - x))
    if xn1 <= np.float32(0.0):
        xn1 = np.float32(1.0e-7)
    elif xn1 >= np.float32(1.0):
        xn1 = np.float32(1.0 - 1.0e-7)
    return np.float32(xn1)
```

```
def tent_next(x: np.float32, mu: np.float32) -> np.float32:
    if x < np.float32(0.5):
        xn1 = np.float32(mu * x)
    else:
        xn1 = np.float32(mu * (np.float32(1.0) - x))
    if xn1 <= np.float32(0.0):
        xn1 = np.float32(1.0e-7)
    elif xn1 >= np.float32(1.0):
        xn1 = np.float32(1.0 - 1.0e-7)
    return np.float32(xn1)

def xor_bytes(a: bytes, b: bytes) -> bytes:
    return bytes(x ^ y for x, y in zip(a, b))

def parse_hex_bytes(s: str) -> bytes:
    s = re.sub(r"[^0-9a-fA-F]", "", s)
    if len(s) % 2:
        raise ValueError("Odd number of hex digits")
    return bytes.fromhex(s)

def adc_u16_le_from_16bytes(raw16: bytes) -> List[int]:
    return [int.from_bytes(raw16[i : i + 2], "little") for i in range(0, 16, 2)]

@dataclass
class ChaoticState:
    r_log: np.float32 = np.float32(3.99)
    mu_tent: np.float32 = np.float32(2.0)
    x_log: np.float32 = np.float32(0.0)
    x_tent: np.float32 = np.float32(0.0)
    counter: np.uint32 = np.uint32(0)

    def init(self, seed1: int, seed2: int) -> None:
        s1 = u32(seed1)
        s2 = u32(seed2)
        self.r_log = np.float32(3.99)
        self.mu_tent = np.float32(2.0)
        self.x_log = seed_to_unit_open(s1)
        self.x_tent = seed_to_unit_open(s2)
        self.counter = np.uint32(
            (int(s1) ^ ((int(s2) * 2654435761) & 0xFFFFFFFF)) & 0xFFFFFFFF
        )

    def step(self) -> None:
        x_log_next = logistic_next(self.x_log, self.r_log)
        x_tent_next = tent_next(self.x_tent, self.mu_tent)

        self.counter = np.uint32((int(self.counter) + GOLDEN_INC) & 0xFFFFFFFF)

        noise = np.float32(np.float32(self.counter) * INV_2_32_F32)
        noise = np.float32((noise - np.float32(0.5)) * np.float32(1.0e-3))

        x_log_h = np.float32(x_log_next + np.float32(0.5) * x_tent_next + noise)
        x_tent_h = np.float32(x_tent_next + np.float32(0.5) * x_log_next - noise)

        # wrap using truncation (matches positive-floor behavior)
        x_log_h = np.float32(x_log_h - np.float32(int(float(x_log_h))))
        x_tent_h = np.float32(x_tent_h - np.float32(int(float(x_tent_h))))

        if x_log_h <= np.float32(0.0):
            x_log_h = np.float32(1.0e-7)
        elif x_log_h >= np.float32(1.0):
            x_log_h = np.float32(1.0 - 1.0e-7)

        if x_tent_h <= np.float32(0.0):
            x_tent_h = np.float32(1.0e-7)
```

```
        elif x_tent_h >= np.float32(1.0):
            x_tent_h = np.float32(1.0 - 1.0e-7)

        self.x_log = np.float32(x_log_h)
        self.x_tent = np.float32(x_tent_h)

    def skip(self, steps: int) -> None:
        for _ in range(int(steps)):
            self.step()

@dataclass
class ChaoticPRNG:
    state: ChaoticState = field(default_factory=ChaoticState)

    def init(self, seed1: int, seed2: int) -> None:
        self.state.init(seed1, seed2)
        self.state.skip(16)

    @staticmethod
    def avalanche(x: np.uint32) -> np.uint32:
        v = np.uint32(x)
        v ^= v >> np.uint32(16)
        v = np.uint32((int(v) * 0x7FEB352D) & 0xFFFFFFFF)
        v ^= v >> np.uint32(15)
        v = np.uint32((int(v) * 0x846CA68B) & 0xFFFFFFFF)
        v ^= v >> np.uint32(16)
        return np.uint32(v)

    def next_word(self) -> np.uint32:
        for _ in range(6):
            self.state.step()

        a = float_to_u32_bits_f32(self.state.x_log)
        b = float_to_u32_bits_f32(self.state.x_tent)
        c = self.state.counter

        mixed = np.uint32(
            (
                int(a)
                ^ int(rotl32(int(b), 11))
                ^ int(rotl32(int(c), 17))
                ^ ((int(a) + (int(b) * 0x9E3779B9)) & 0xFFFFFFFF)
            )
            & 0xFFFFFFFF
        )

        return self.avalanche(mixed)

    def fill_bytes(self, n: int) -> bytes:
        out = bytearray()
        word = 0
        byte_index = 4
        for _ in range(n):
            if byte_index >= 4:
                word = int(self.next_word()) & 0xFFFFFFFF
                byte_index = 0
            out.append(word & 0xFF)
            word >>= 8
            byte_index += 1
        return bytes(out)

def decrypt_ct16_with_ctr(ct16: bytes, ctr: int) -> bytes:
    seed = ctr & 0xFFFFFFFF
    seed1 = seed
    seed2 = seed ^ A5
    prng = ChaoticPRNG()
```

```
    prng.init(seed1, seed2)
    ks = prng.fill_bytes(16)
    return xor_bytes(ct16, ks)

# ============================================================
#  Correlator
# ============================================================

def ts_now() -> str:
    t = time.time()
    lt = time.localtime(t)
    ms = int((t - int(t)) * 1000)
    return time.strftime("%Y-%m-%d %H:%M:%S", lt) + f".{ms:03d}"

@dataclass
class Frame:
    ctr: int
    raw16: Optional[bytes] = None
    ct16: Optional[bytes] = None
    rx_ct16: Optional[bytes] = None
    rssi: Optional[int] = None
    snr: Optional[float] = None
    rx_payload: Optional[bytes] = None
    decrypted16: Optional[bytes] = None
    match: Optional[bool] = None
    created: float = field(default_factory=time.time)
    printed: bool = False

@dataclass
class Stats:
    sent: int = 0
    received: int = 0
    matched: int = 0
    rssi_vals: List[int] = field(default_factory=list)
    snr_vals: List[float] = field(default_factory=list)

    def summary_lines(self) -> List[str]:
        lost = self.sent - self.received
        lines = [f"Sent={self.sent}, Recv={self.received}, Lost={lost}"]
        if self.received:
            rssi_avg = sum(self.rssi_vals) / len(self.rssi_vals)
            snr_avg = sum(self.snr_vals) / len(self.snr_vals)
            lines.append(
                f"RSSI avg={rssi_avg:.1f} dBm (min={min(self.rssi_vals)},
max={max(self.rssi_vals)})"
            )
            lines.append(
                f"SNR  avg={snr_avg:.1f} dB  (min={min(self.snr_vals):.1f},
max={max(self.snr_vals):.1f})"
            )
        if self.received:
            lines.append(
                f"Decrypt MATCH={self.matched}/{self.received}
({100.0*self.matched/self.received:.1f}%)"
            )
        return lines

# MCU line example:
# Sent packet #00388, Raw Data: A806..., Encrypted Data: 1410...
RE_MCU = re.compile(
    r"Sent\s+packet\s+#(\d+)\s*,\s*Raw\s+Data:\s*([0-9A-Fa-
f]{32})\s*,\s*Encrypted\s+Data:\s*([0-9A-Fa-f]{32})"
)

# RM1261 RX example:
# RX:[P2P] S:source, R:rssi, S:snr, data
```

```
RE_RX = re.compile(
    r"RX:\s*\[P2P\]\s+S:([^,]+),\s*R:([-\d]+)\s*dBm,\s*S:([-\d\.]+)\s*dB,\s*([0-9A-Fa-f]+)"
)

def main():
    ap = argparse.ArgumentParser()
    ap.add_argument("--mcu", required=True, help="MCU log COM port (e.g. COM7)")
    ap.add_argument("--rx", required=True, help="Receiver log COM port (e.g. COM9)")
    ap.add_argument("--baud_mcu", type=int, default=115200)
    ap.add_argument("--baud_rx", type=int, default=115200)
    ap.add_argument("--out", default="unified_log.txt")
    ap.add_argument(
        "--ttl_s", type=float, default=120.0, help="Drop frames older than TTL seconds"
    )
    ap.add_argument(
        "--print_adc",
        action="store_true",
        help="Print ADC values (hex) when decrypting",
    )
    args = ap.parse_args()

    frames: Dict[int, Frame] = {}
    lock = threading.Lock()
    stats = Stats()
    completed_frames = 0

    out_f = open(args.out, "w", encoding="utf-8")

    def log(line: str):
        out_f.write(line + "\n")
        out_f.flush()
        print(line)

    def cleanup():
        now = time.time()
        with lock:
            stale = [ctr for ctr, fr in frames.items() if now - fr.created > args.ttl_s]
            for ctr in stale:
                del frames[ctr]

    def finalize_if_ready(ctr: int):
        fr = frames.get(ctr)

        if not fr:
            return

        if fr.printed:
            return

        # need RX ciphertext to decrypt
        if fr.rx_ct16 is None:
            return

        # decrypt
        fr.decrypted16 = decrypt_ct16_with_ctr(fr.rx_ct16, ctr)

        # compare if TX raw exists
        if fr.raw16 is not None:
            fr.match = fr.decrypted16 == fr.raw16
        else:
            fr.match = None

        # log final line
        adc_hex = ""
        if args.print_adc:
            vals = adc_u16_le_from_16bytes(fr.decrypted16)
```

```
            adc_hex = " ADC=" + " ".join(f"{v:04X}" for v in vals)

        match_txt = ""
        if fr.match is True:
            match_txt = " (MATCH)"
        elif fr.match is False:
            match_txt = " (MISMATCH)"

        rssi_snr = ""
        if fr.rssi is not None and fr.snr is not None:
            rssi_snr = f" RSSI={fr.rssi} SNR={fr.snr:.1f}"

        log(
            f"{ts_now()} | PC   | ctr={ctr:05d} |
raw={fr.decrypted16.hex().upper()}{adc_hex}{rssi_snr}{match_txt}"
        )

        nonlocal completed_frames
        completed_frames += 1

        block_no = completed_frames
        # log("")
        log(f"========== FRAME BLOCK {block_no:06d} ==========")
        log("")

        # stats
        stats.received += 1
        if fr.rssi is not None:
            stats.rssi_vals.append(fr.rssi)
        if fr.snr is not None:
            stats.snr_vals.append(fr.snr)
        if fr.match is True:
            stats.matched += 1

        fr.printed = True
        # done
        del frames[ctr]

    # --- Readers ---
    def reader_mcu():
        with serial.Serial(args.mcu, args.baud_mcu, timeout=0.2) as ser:
            while True:
                line = ser.readline().decode(errors="ignore").strip()
                if not line:
                    cleanup()
                    continue

                m = RE_MCU.search(line)
                if m:
                    ctr = int(m.group(1))
                    raw16 = bytes.fromhex(m.group(2))
                    ct16 = bytes.fromhex(m.group(3))

                    with lock:
                        fr = frames.get(ctr)
                        if fr is None:
                            fr = Frame(ctr=ctr)
                            frames[ctr] = fr
                        fr.raw16 = raw16
                        fr.ct16 = ct16
                        fr.created = time.time()

                    stats.sent += 1
                    log(
                        f"{ts_now()} | MCU  | ctr={ctr:05d} | raw={raw16.hex().upper()}"
                    )
```

```
                    log(f"{ts_now()} | MCU  | ctr={ctr:05d} | ct ={ct16.hex().upper()}")
                    continue

                log(f"{ts_now()} | MCU  | {line}")

    def reader_rx():
        with serial.Serial(args.rx, args.baud_rx, timeout=0.2) as ser:
            while True:
                line = ser.readline().decode(errors="ignore").strip()
                if not line:
                    cleanup()
                    continue

                m = RE_RX.search(line)
                if m:
                    rssi = int(m.group(2))
                    snr = float(m.group(3))
                    data_str = m.group(4)
                    payload = parse_hex_bytes(data_str)

                    # Heuristic:
                    # If payload is 18 bytes and starts with ctr (u16 LE), use that ctr and
ct16=payload[2:18]
                    # Else if payload is 16 bytes, we try to match it to a recent TX
ciphertext
                    ctr: Optional[int] = None
                    rx_ct16: Optional[bytes] = None

                    if len(payload) == 18:
                        ctr = int.from_bytes(payload[0:2], "little")
                        rx_ct16 = payload[2:18]
                    elif len(payload) == 16:
                        rx_ct16 = payload
                        # attempt correlation by ciphertext lookup:
                        # find most recent frame whose ct16 equals this rx_ct16
                        with lock:
                            for c, fr in list(frames.items()):
                                if fr.ct16 is not None and fr.ct16 == rx_ct16:
                                    ctr = c
                                    break
                    else:
                        # unknown size: log and move on
                        log(
                            f"{ts_now()} | RX   | len={len(payload)} RSSI={rssi}
SNR={snr:.1f} DATA={payload.hex().upper()}"
                        )
                        continue

                    if ctr is None or rx_ct16 is None:
                        log(
                            f"{ts_now()} | RX   | RSSI={rssi} SNR={snr:.1f}
DATA={payload.hex().upper()} (UNMATCHED)"
                        )
                        continue

                    with lock:
                        fr = frames.get(ctr)
                        if fr is None:
                            fr = Frame(ctr=ctr)
                            frames[ctr] = fr
                        fr.rssi = rssi
                        fr.snr = snr
                        fr.rx_payload = payload
                        fr.rx_ct16 = rx_ct16
                        fr.created = time.time()
```

```
                    log(
                        f"{ts_now()} | RX   | ctr={ctr:05d} | RSSI={rssi} SNR={snr:.1f}
data={payload.hex().upper()}"
                    )
                    finalize_if_ready(ctr)
                    continue

                log(f"{ts_now()} | RX   | {line}")

    t1 = threading.Thread(target=reader_mcu, daemon=True)
    t2 = threading.Thread(target=reader_rx, daemon=True)
    t1.start()
    t2.start()

    log(
        f"{ts_now()} | INFO | Started. MCU={args.mcu}@{args.baud_mcu}
RX={args.rx}@{args.baud_rx} OUT={args.out}"
    )
    try:
        while True:
            time.sleep(2.0)
    except KeyboardInterrupt:
        log(f"{ts_now()} | INFO | Stopping...")
        for s in stats.summary_lines():
            log(f"{ts_now()} | SUMMARY | {s}")
        out_f.close()

if __name__ == "__main__":
    main()
```