

PhantomSDR-Plus

Πλήρης τεκμηρίωση

16 έγγραφα · δημιουργήθηκε 2026-10-03

Περιεχόμενα

1. README	8
Βασικά χαρακτηριστικά	8
Υποστηριζόμενο υλικό	10
Γρήγορη εκκίνηση	11
Μετρήσεις επιδόσεων	12
Χρήση	12
Ρυθμίσεις	13
Κατώτατο όριο καταρράκτη — waterfall.sh	14
Προσαρμογή	15
Τεκμηρίωση	17
Διαδικτυακός κατάλογος WebSDR	17
Αντιμετώπιση προβλημάτων	17
Συνεισφορά	18
Άδεια χρήσης	18
Συντελεστές και ευχαριστίες	18
Σύνδεσμοι	18
Υποστήριξη	18
Συμβουλές επιδόσεων	18
2. Οδηγός εγκατάστασης	20
Πίνακας περιεχομένων	20
Απαιτήσεις συστήματος	20
Προετοιμασία πριν την εγκατάσταση	21
Εγκατάσταση εξαρτήσεων	22
Εγκατάσταση Node.js και npm	23
Εγκατάσταση OpenCL (προαιρετικό αλλά συνιστώμενο)	24
Εγκατάσταση του PhantomSDR-Plus	26
Εγκατάσταση του κωδικοποιητή ήχου Opus	31
Autorun Spot Reporter (FT8/FT4/WSPR)	32
Εξομοίωση πελατών KiwiSDR (προαιρετικό)	34
Ρυθμίσεις	35
Ρύθμιση ανά συσκευή SDR	38
Δοκιμή και επαλήθευση	41
Ρύθμιση αυτόματης εκκίνησης	42
Θερμική προστασία CPU	44
Αντιμετώπιση προβλημάτων	46
Βελτιστοποίηση επιδόσεων	50
Ενημέρωση του PhantomSDR-Plus	51
Αντίγραφα ασφαλείας και επαναφορά	53
Θέματα ασφάλειας	54
Λήψη βοήθειας	54
Παράρτημα A: πλήρης κατάλογος εξαρτήσεων	55
Παράρτημα B: παραδείγματα ρυθμίσεων	56
3. Πίνακας διαχείρισης	58
Επισκόπηση	58
Τι σας προσφέρει ο πίνακας διαχείρισης	58

Απαιτήσεις	59
Βήμα 1 — Αρχεία	59
Βήμα 2 — Εκτελέστε το σενάριο ρύθμισης	59
Βήμα 3 — Εκκίνηση / διακοπή / επανεκκίνηση	60
Βήμα 4 — Ανοίξτε τη θύρα στο τείχος προστασίας	63
Βήμα 5 — Λειτουργία Kick Users	63
Ρυθμίσεις — admin_config.json	63
Η ρύθμιση sdr_host	64
proxy.py — τι είναι και πότε το χρειάζεστε	65
Αλλαγή θυρών μετά τη ρύθμιση	65
Χρήσιμες χειροκίνητες εντολές	66
Αρχεία καταγραφής	66
Σύνοψη πρόσβασης	68
Διαγραφή μηνυμάτων συνομιλίας	69
Μηνύματα μετάδοσης στον καταρράκτη	69
Γραφήματα συστήματος	70
Thermal Guard	71
Spot Reporting (Autorun FT8/FT4/JS8/WSPR)	74
Αντιμετώπιση προβλημάτων	80
4. Thermal Guard	83
Περιεχόμενα	83
1. Τι κάνει	83
2. Γρήγορη εκκίνηση	84
3. Πώς αποφασίζει	84
4. Οι τέσσερις λειτουργίες	85
5. Επιλογή λειτουργίας	87
6. Αναφορά ρυθμίσεων	88
7. Λειτουργία χωρίς τον πίνακα διαχείρισης	88
8. Ενεργοποίηση του σταδίου throttle χωρίς root	89
9. Λειτουργία χωρίς επιτήρηση	90
10. Δοκιμή πριν του εμπιστευτείτε το μηχανήμα	91
11. Διαβάζοντας το αρχείο καταγραφής	92
12. Lockout και όριο συχνότητας	92
13. Επίλυση προβλημάτων	93
14. Απενεργοποίηση ή αφαίρεση	94
Δείτε επίσης	94
5. Όρια συνδέσεων	95
Περιεχόμενα	95
1. Από τι αμύνεστε στην πραγματικότητα	95
2. Τα τρία επίπεδα	96
3. Όρια ανά διεύθυνση	96
4. Αδρανείς συνδέσεις	98
5. Η προστασία στον πυρήνα	98
6. Αναφορά ρυθμίσεων	99
7. Τι βλέπει ένας ακροατής που απορρίφθηκε	100
8. Διαβάζοντας τα αρχεία καταγραφής και τους μετρητές	101
9. Επιλέγοντας τους αριθμούς σας	101
10. Παγίδες που αξίζει να ξέρετε	102
11. Απενεργοποίηση όλων	103

6. Πολλοί δέκτες	104
Περιεχόμενα	104
1. Πώς λειτουργεί	104
2. Δύο τρόποι δημοσίευσης των δεκτών	105
3. Προσθήκη δευτέρου δέκτη	105
4. receivers.toml	107
5. Εκκίνηση και διακοπή	108
6. Τι βλέπουν οι ακροατές	108
7. Το S-meter πάνω από τα 30 MHz	109
8. Ασφάλεια	109
9. Πόσοι δέκτες χωράνε	110
10. Παγίδες που αξίζει να γνωρίζετε	110
7. Διακομιστής στατιστικών	111
Επισκόπηση	111
Προαπαιτούμενα	111
Μέθοδοι εγκατάστασης	111
Διαμορφώστε την εφαρμογή σας PhantomSDR	114
Δοκιμή της εγκατάστασης	116
Διαχείριση υπηρεσίας	117
Χειροκίνητη εγκατάσταση (Επιλογή 2)	117
Αλλαγή της θύρας μετά την εγκατάσταση	119
Αντιμετώπιση προβλημάτων	119
Τι θα δείτε	122
Ανίχνευση θερμοκρασίας	122
Σύνοψη χαρακτηριστικών	123
Σημειώσεις ασφαλείας	123
Υποστήριξη και πόροι	123
8. Αποκωδικοποιητές	126
Πίνακας περιεχομένων	126
1. Πώς ξεκινά ένας αποκωδικοποιητής	126
2. FT8	129
3. FT4-FT2	130
4. JS8	131
5. CW — κώδικας Μορς	133
6. QRSS Grabber	134
7. WSPR	138
8. HF FAX / WEFAX	139
9. NAVTEX	140
10. FSK / RTTY — μαζί με PSK31 και Olivia	141
11. SSTV	145
12. Γενικές συμβουλές	147
9. RADE README	149
Η εγκατάσταση — ο σύντομος δρόμος	149
Πίνακας περιεχομένων	149
Πώς λειτουργεί	150
Επισκόπηση αρχιτεκτονικής	151
RADEL έναντι RADEU	151
Προϋποθέσεις	152

Εγκατάσταση με το χέρι	152
Τοποθέτηση των διορθωμένων αρχείων	152
Έλεγχος του sidcar	153
Η θύρα 8074	154
Χρήση του RADE στον περιηγητή	154
Επαλήθευση και αποσφαλμάτωση	155
Μεταβλητές περιβάλλοντος	156
Αρχεία που άλλαξαν	157
Σύνοψη ροής σήματος	157
Αντιμετώπιση προβλημάτων	158
Μέτρηση ταυτοχρονίας στο δικό σας υλικό	160
Ενημέρωση του RADE v1	163
10. RADE — γενική εγκατάσταση (Linux)	165
Βήμα 1 — Πακέτα συστήματος	165
Βήμα 2 — Πακέτα Python	166
Βήμα 3 — Κλωνοποιήστε και μεταγλωττίστε το αποθετήριο rade	167
Βήμα 4 — Επαληθεύστε την αλυσίδα αποκωδικοποίησης	168
Βήμα 5 — Μεταγλωττίστε το frontend του PhantomSDR-Plus	169
Βήμα 6 — Έλεγχος του sidcar	169
Το rade.sh εξακολουθεί να διατίθεται για αυτόνομη χρήση — για να τρέξετε το sidcar χωρίς τον launcher του PhantomSDR-Plus, ή για να το δοκιμάσετε μόνο του. Οι εντολές του, το αρχείο καταγραφής του και οι σχετικές προειδοποιήσεις για τον watchdog βρίσκονται στο Έλεγχος του sidcar.	169
Βήμα 7 — Ανοίξτε τη θύρα 8074	169
Βήμα 8 — Εκκινήστε τον διακομιστή	171
Βήμα 9 — Εκκινήστε το RADE	171
Βήμα 10 — Χρήση του RADE στον περιηγητή	171
Σημείωση για τη CPU	172
Μελλοντική ενημέρωση του RADE	172
11. Επεξεργασία παραλλαγών	173
1. Τι είναι οι παραλλαγές	173
2. Τι κάνει κάθε αρχείο	174
3. Αλλαγή που επηρεάζει μόνο ορισμένες παραλλαγές	176
4. Πού βρίσκονται οι διακλαδώσεις παραλλαγών	177
5. Άλλα ρυθμιστικά που αξίζει να γνωρίζετε	178
6. Μετά την επεξεργασία — ανακατασκευή	178
7. Πώς να ελέγξετε την αλλαγή σας πριν την εμπιστευτείτε	179
12. Οδηγός χρήσης	180
Πίνακας περιεχομένων	180
Εισαγωγή	180
Ξεκινώντας	181
Επισκόπηση της διεπαφής	182
Βασικές λειτουργίες	183
Προηγμένες δυνατότητες	186
Τρόποι αποδιαμόρφωσης	190
Ψηφιακοί αποκωδικοποιητές	193
Συντομεύσεις πληκτρολογίου	196
Σελιδοδείκτες	197
Χρήση σε κινητά	198

Συμβουλές και καλές πρακτικές	199
Αντιμετώπιση προβλημάτων	200
Συχνές ερωτήσεις	202
Πόροι	202
Παράρτημα: συνήθεις συχνότητες	203
Γλωσσάρι	204
13. Εξομοίωση πελάτη KiwiSDR	206
Πίνακας περιεχομένων	206
1. Τι παίρνει ένας πελάτης Kiwi	206
2. Εγκατάσταση της γέφυρας	207
3. Ενεργοποίηση	207
4. Αναφορά ρυθμίσεων	208
5. Σύνδεση ενός πελάτη	208
6. Στάθμη ήχου	208
7. Το S-meter	209
8. Καταρράκτης και φάσμα	210
9. Αν κάτι δεν φαίνεται σωστό	211
10. Μέχρι πού έχει επαληθευτεί	212
14. Διαφορική λήψη	213
Τι κάνει και τι δεν κάνει	213
Γρήγορη εκκίνηση	214
Χρήση WebSDR	215
Ανάγνωση του πίνακα	216
Remote SNR trim	217
Επιλογή δεύτερου δέκτη	218
Επίλυση προβλημάτων	218
Περιορισμοί	219
15. Έλεγχος πομποδέκτη	220
Τι χρειάζεστε	220
Υποστηριζόμενοι δέκτες	221
Τι κάνει και τι δεν κάνει	221
Γρήγορη εκκίνηση	221
Επιλογή τρόπου σύνδεσης	222
Ρυθμίσεις σειριακής θύρας	223
Συγχρονισμός	223
Τρόποι λειτουργίας	224
Εύρος φίλτρου	225
Σίγαση κατά την εκπομπή	225
Μετατόπιση συχνότητας	226
Το μενού Rig	226
Linux	226
Windows	227
TCI-CAT στη σελίδα του δέκτη	227
Για διαχειριστές δεκτών	237
Για προγραμματιστές: η διεπαφή της σελίδας	237
Αντιμετώπιση προβλημάτων	238
Γνωστοί περιορισμοί	238
16. Δομή του έργου	240

Πίνακας περιεχομένων	240
Δέντρο καταλόγων	241
Ριζικός κατάλογος	254
Πηγαίος κώδικας (src/)	257
Frontend (frontend/)	258
Λίστες συχνότητων (frequencylist/)	263
Αρχεία ρυθμίσεων	263
Σύστημα μεταγλώττισης	265
Εξαρτήσεις αρχείων	265
Ροή δεδομένων	266
Οδηγός τροποποίησης αρχείων	266
Σημαντικές διαδρομές	267
Συνήθειες εργασίες με αρχεία	267
Έλεγχος εκδόσεων	268

README

README.md

PhantomSDR-Plus

Ένας βελτιωμένος διακομιστής WebSDR ανοικτού κώδικα με προηγμένες δυνατότητες και φουτουριστικό σχεδιασμό

Το PhantomSDR-Plus είναι ένα fork του PhantomSDR που προσφέρει έναν διακομιστή web υψηλών επιδόσεων για ραδιόφωνο καθορισμένο από λογισμικό (SDR), ικανό να εξυπηρετεί εκατοντάδες ταυτόχρονους χρήστες. Διαθέτει βελτιωμένο περιβάλλον χρήστη, υποστήριξη πολλαπλών αποκωδικοποιητών, οπτικοποίηση του σχεδίου μπαντών και συμβατότητα με διάφορες πλατφόρμες υλικού SDR.

Βασικά χαρακτηριστικά

Επιδόσεις και κλιμάκωση

- **Υποστήριξη πολλών χρηστών:** εκατοντάδες ταυτόχρονοι χρήστες, ανάλογα με το υλικό
- **Υψηλός ρυθμός δειγματοληψίας:** υποστήριξη SDR έως 70 MSPS (πραγματικό) / 35 MSPS (IQ)
- **Επιτάχυνση υλικού:** υποστήριξη OpenCL και CUDA για επεξεργασία με επιτάχυνση GPU
- **Βελτιστοποιημένη ροή:** συμπίεση ήχου FLAC και Opus χαμηλής καθυστέρησης. Ο φυλλομετρητής αποκωδικοποιεί κάθε πακέτο FLAC τη στιγμή που φτάνει, και το buffer αναπαραγωγής του προσαρμόζεται στη σύνδεση του ακροατή, κρατώντας μόνο όσο ήχο χρειάζεται — συνήθως 100-150 ms από τον δέκτη ως το μεγάφωνο σε κινητό μέσω Wi-Fi

Περιβάλλον χρήστη

- **Φουτουριστικός σχεδιασμός:** σύγχρονο, αποκρίσιμο περιβάλλον web
- **Βελτιστοποιημένο για κινητά:** βελτιωμένο γραφικό περιβάλλον για ακρόαση εν κινήσει
- **Οπτικοποίηση σχεδίου μπαντών:** διαδραστικός καταρράκτης με επικαλύψεις ζωνών συχνοτήτων
- **Προσαρμόσιμοι χρωματικοί χάρτες:** πολλαπλά χρωματικά σχήματα καταρράκτη
- **Διπλό S-meter:** επιλογή μεταξύ αναλογικής ή ψηφιακής ένδειξης σήματος
- **Κουμπιά αποκωδικοποιητών με ένα πάτημα:** μια σειρά Decoders στον κύριο πίνακα, ακριβώς κάτω από τον επιλογέα τρόπων λειτουργίας (FT8, FT4, FT2, JS8, CW, WSPR, FAX, SSTV, NAVTX, RTTY), με τα RADEL και RADEU δίπλα στον ίδιο τον επιλογέα τρόπων και να επαναλαμβάνονται στα αναδυόμενα παράθυρα τρόπων και μπαντών. Ένα πάτημα ξεκινά τον αποκωδικοποιητή και ανοίγει το παράθυρό του, ένα δεύτερο τον σταματά. Η παλιά σειρά Bandwidth έχει αφαιρεθεί.
- **Σαρωτής καναλιών:** σαρώνει τη μπάντα, ή ακριβώς ό,τι δείχνει ο καταρράκτης, και σταματά στο πρώτο κανάλι που ανεβαίνει επιλεγμένα dB πάνω από το επίπεδο θορύβου της ίδιας της μπάντας, και συνεχίζει μόνος του μόλις το κανάλι ησυχάσει. Το βήμα ακολουθεί τον τρόπο λειτουργίας και οι στάσεις πέφτουν πάνω στο πλέγμα καναλιών (1 kHz SSB, 100 Hz CW, 5 kHz AM βραχέων, 9/10 kHz μεσαία, 9 kHz μακρά). Μια λειτουργία *Παράλειψη κενών* διαβάζει το φάσμα του καταρράκτη και πηδά κατευθείαν στα σήματα, ενώ ένα κουμπί κλειδώματος βγάζει από τη σάρωση ένα διαρκώς κατειλημμένο κανάλι. Όλα γίνονται τοπικά στον φυλλομετρητή του ακροατή, οπότε η σάρωση δεν μετακινεί τον δέκτη για κανέναν άλλον (βλ. [Οδηγό χρήσης](#))

- **Διαφορική λήψη:** συνδυάζει τον δέκτη με έναν δεύτερο σε άλλο σημείο και αναπαράγει όποιον από τους δύο έχει εκείνη τη στιγμή το καλύτερο σήμα, ώστε ένα fading στον έναν σταθμό να καλύπτεται από τον άλλον. Ο δεύτερος δέκτης μπορεί να είναι άλλο ένα PhantomSDR-Plus, ένα KiwiSDR, ένα UberSDR ή ένα WebSDR· τα τρία πρώτα δεν χρειάζονται τίποτα πέρα από τον browser του ακροατή και μόνο ένα WebSDR απαιτεί ένα μικρό relay στον δικό σας διακομιστή. Ο συγχρονισμός των δύο παίρνει περίπου δεκαπέντε δευτερόλεπτα και μια ρύθμιση απομακρυσμένου SNR τους εξισορροπεί μεταξύ τους. Κάθε ακροατής κρατά στον δικό του browser μια λίστα με ονόματα δεύτερων δεκτών, στη σειρά που θέλει, και μπορεί να την εξάγει σε αρχείο και να την επαναφέρει (δείτε [Διαφορική λήψη](#))
- **Πολλοί δέκτες, μία σελίδα:** ένας υπολογιστής μπορεί να τρέχει ταυτόχρονα δύο ή περισσότερους δέκτες — για παράδειγμα ένα RX-888 για τα βραχέα και ένα RTL-SDR για τα 2 m — και οι ακροατές περνούν από τον έναν στον άλλο με κουμπιά στην κεφαλίδα της σελίδας. Κάθε δέκτης κρατά τον δικό του καταρράκτη, chat, σημάνσεις, λίστα ακροατών και στοιχεία σταθμού, και πάνω από τα 30 MHz το S-meter δείχνει μονάδες S VHF (S9 = -93 dBm) (δείτε [Πολλοί δέκτες](#))

Επεξεργασία σήματος

- **Πολλαπλοί τρόποι αποδιαμόρφωσης:** AM, FM, USB, LSB, CW και άλλοι. Έχει επίσης υλοποιηθεί ένας αποκωδικοποιητής RADE έκδοσης 1.
- **Σύγχρονη ανίχνευση AM:** βελτιωμένη ποιότητα λήψης AM
- **Μείωση θορύβου:** NR (φασματικό), NB (κόπτης κρουστικού θορύβου), NS (καταστολή θορύβου βάθους) και AN (αυτόματο notch) — όλα στη διαδρομή ακρόασης, ποτέ στους αποκωδικοποιητές
- **Επιλογές AGC:** πολλαπλοί τρόποι αυτόματου ελέγχου κέρδους
- **Αυτόματο squelch:** αυτόματο κατώφλι squelch βάσει θορύβου
- **Squelch υποτόνου CTCSS:** ο ήχος FM περνά μόνο όσο λαμβάνεται ο επιλεγμένος υπότονος — ή οποιοσδήποτε τυποποιημένος· ο τόνος επιλέγεται από μικρό παράθυρο δίπλα στο κουμπί CTCSS, που τον εμφανίζει στη συνέχεια
- **Μείωση θορύβου AI:** νευρωνικό δίκτυο RNNoise στον φυλλομετρητή του ακροατή, αφαιρεί τον θόρυβο της μπάντας από την ομιλία (μόνο φωνητικοί τρόποι· δείτε τον [Οδηγό χρήσης](#))

Προηγμένες δυνατότητες

- **Ψηφιακοί αποκωδικοποιητές:** FT8, FT4, FT2, JS8, CW, QRSS Grabber, WSPR, HF FAX, SSTV, NAVTEX, FSK/RTTY, PSK31, Olivia και FreeDV RADE, καθένας σε δικό του νήμα παρασκηνίου ώστε η αποκωδικοποίηση να μη διακόπτει ποτέ τον ήχο (δείτε [Αποκωδικοποιητές](#))
- **Decoder ID:** ονομάζει τον ψηφιακό τρόπο εκπομπής μέσα στο εύρος διέλευσης και προσφέρει τον κατάλληλο αποκωδικοποιητή με ένα κλικ, από μετρημένο εύρος ζώνης, απόσταση τόνων, ρυθμό συμβόλων και χρονισμό ριπών, λαμβάνοντας υπόψη και τη συχνότητα ως πρόσθετη ένδειξη — κάτι που ξεχωρίζει επίσης το **FT8 από το JS8**, δύο τρόπους εκπομπής με πανομοιότυπο σήμα. Μόνο προτείνει, δεν αλλάζει ποτέ μόνο του, και προτιμά να σιωπήσει παρά να μαντέψει όταν το σήμα είναι πολύ ασθενές. Οι διαλείψεις δεν το αποσιωπούν: όταν το QSB κόβει σε κομμάτια μια δίλεπτη εκπομπή WSPR, εξακολουθεί να ονομάζει τον τρόπο εκπομπής, διαβάζοντας τον ρυθμό από ολόκληρο το ιστορικό ακρόασης, και χαμηλώνει τη βεβαιότητα που δείχνει για να δηλώσει ότι η ένδειξη είναι ασθενέστερη. Μεταξύ περίπου 400 Hz και 2700 Hz η θέση μέσα στο εύρος διέλευσης δεν έχει σημασία· εκτός αυτού εμφανίζεται ένα κεχριμπαρένιο κουμπί **Recentre & retry**, που μετακινεί μία φορά τον συντονισμό και ξεκινά νέα μέτρηση. Όταν δεχθείτε μια πρόταση, ο δέκτης ρυθμίζεται για αυτόν τον τρόπο εκπομπής: η συντονισμένη συχνότητα πηγαίνει στο κέντρο του καταρράκτη, η εικόνα γίνεται περίπου 100 kHz γύρω της, και η πλευρική ζώνη και το εύρος διέλευσης γίνονται αυτά με τα οποία δουλεύεται ο τρόπος αυτός — η υποζώνη των 3 kHz για την οικογένεια FT8, 1350–1650 Hz για το WSPR, ± 250 Hz και CW για το Morse, και ούτω καθεξής. Ανενεργό εξ ορισμού, και περίπου 0,5% ενός πυρήνα όταν τρέχει (βλ. [Αποκωδικοποιητές](#))

- **Προεπιλογές ψηφιακών τρόπων:** το FT8, FT4, FT2, JS8 ή WSPR ρυθμίζει ταυτόχρονα την πλευρική ζώνη και το εύρος διέλευσης
- **Πίνακας συνομιλίας JS8:** τα μηνύματα πολλαπλών πλαισίων συναρμολογούνται σε ολόκληρες προτάσεις
- **Πίνακας στατιστικών:** στατιστικά διακομιστή και χρηστών σε πραγματικό χρόνο
- **Λίστα συνδεδεμένων χρηστών:** όλοι όσοι ακούν, με κουμπί συντονισμού σε κάθε γραμμή. Η δική σας συνεδρία σημειώνεται με **you** — τόσο στο παράθυρο της διεπαφής όσο και όταν το `users.html` ανοίγει ως ξεχωριστή σελίδα
- **Αυτόματη αναφορά spot:** ένας δαίμονας `autorun` αποκωδικοποιεί FT8, FT4 και WSPR στον διακομιστή και ανεβάζει spot στο PSK Reporter και στο WSPRnet, με δύο μετρητές στον πίνακα διαχείρισης — πλακίδια ανά αποκωδικοποιητή για την τρέχουσα εκτέλεση και συνολικό άθροισμα δίπλα σε κάθε πλαίσιο ελέγχου μπάντας/τρόπου
- **Γραφήματα διακομιστή:** σελίδα **Γραφήματα** στον πίνακα διαχείρισης με συχνότητα CPU, φόρτο CPU, θερμοκρασία CPU και συνδεδεμένους χρήστες για τα τελευταία 15 λεπτά έως 24 ώρες (δείτε [Πίνακας διαχείρισης](#))
- **Θερμική προστασία:** μια προστασία μέσα στον πίνακα διαχείρισης σταματά τον διακομιστή αν υπερθερμανθεί η CPU και τον επανεκκινεί μόλις κρυώσει, με όρια που προκύπτουν από το κρίσιμο όριο της ίδιας σας της CPU και χωρίς καμία υπόθεση για το πώς ξεκινάτε ή σταματάτε τον διακομιστή· έρχεται σε λειτουργία μόνο καταγραφής, οπότε δεν κάνει τίποτα μέχρι να το ενεργοποιήσετε (δείτε [Thermal Guard](#)) Για δέκτη χωρίς επιτήρηση **συνιστάται έντονα** να τρέχει ο πίνακας ως μονάδα `systemd` — η προστασία ζει μέσα του, οπότε αλλιώς μια επανεκκίνηση την παίρνει μαζί της (δείτε [Πίνακας διαχείρισης](#)).
- **Εξομοίωση πελατών KiwiSDR:** μια προαιρετική γέφυρα που απαντά και στο πρωτόκολλο KiwiSDR, στον ίδιο host και στην ίδια θύρα, ώστε λογισμικό Kiwi όπως το **AetherSDR** και το `kiwiclient` να συνδέεται απευθείας στον δέκτη — με πραγματικό συντονισμό, waterfall και S-meter στην ίδια κλίμακα με αυτό της ιστοσελίδας. Παραμένει κλειστή μέχρι να μπει `[kiwi_emulation] enabled = true` στη διαμόρφωσή σας (δείτε [Εγκατάσταση](#))
- **Έλεγχος πομποδέκτη (CAT):** κρατά τον δικό σας πομποδέκτη και το παράθυρο του δέκτη στην ίδια συχνότητα, τρόπο λειτουργίας και φίλτρο, προς τη μία κατεύθυνση ή και προς τις δύο — μέσω του μενού Rig του [Desktop PhantomSDR+](#) (Icom, Yaesu, Kenwood, Elecraft, FlexRadio και QRP Labs απευθείας, κάθε άλλος πομποδέκτης μέσω Hamlib, ή flrig) ή με το CATsync Tool for WebSDRs σε browser. Οι σελίδες προσφέρουν συναρτήσεις `catsync_*` για συχνότητα, τρόπο λειτουργίας, εύρος φίλτρου και σίγαση. Λειτουργεί με δέκτες PhantomSDR-Plus, KiwiSDR, PA3FWM WebSDR και UberSDR (δείτε [Έλεγχος πομποδέκτη](#))
- **Σύστημα σελιδοδεικτών:** εισαγωγή/εξαγωγή σελιδοδεικτών συχνοτήτων
- **Συντομεύσεις πληκτρολογίου:** αποδοτική πλοήγηση και χειρισμός
- **Υποστήριξη ροδέλας ποντικιού:** διαισθητικός συντονισμός συχνότητας
- **Κατάλογος WebSDR:** ενσωμάτωση με το <https://sdr-list.xyz>

Υποστηριζόμενο υλικό

Το PhantomSDR-Plus υποστηρίζει ευρύ φάσμα δεκτών SDR:

Συσκευή	Ρυθμός δειγματοληψίας	Μορφή	Διεπαφή
RX888 MK2	Έως 64 MHz	16-bit	Εγγενής
RTL-SDR	Έως 3.2 MHz	8-bit	rtl_sdr
HackRF One	Έως 20 MHz	8-bit	hackrf_transfer
Airspy HF+ / Discovery	Έως 912 kHz	16-bit	SoapySDR (SoapyAirspyHF) + rx_sdr
SDRplay RSP1A	Έως 10 MHz	16-bit	SoapySDR (libmirisdr-5 + SoapyMiri, ή το API της SDRplay) + rx_sdr
RigExpert Fobos SDR	50 MHz (HF άμεση δειγματοληψία) / έως 20 MHz (RF)	16-bit / 32-bit float	SoapySDR (SoapyFobosSDR) + rx_sdr
Άλλες συσκευές	Μεταβλητός	Διάφορες	SoapySDR/rx_tools

Γρήγορη εκκίνηση

Απαιτήσεις συστήματος

Ελάχιστες:

- Ubuntu 22.04 LTS (συνιστάται) ή Fedora
- CPU 2 πυρήνων
- 4 GB RAM
- 10 GB χώρου στον δίσκο

Συνιστώμενες:

- Ubuntu 24.04 LTS
- CPU 4+ πυρήνων (Ryzen 5 2600 ή Intel i5-6500T ή καλύτερος)
- 8 GB RAM
- GPU με υποστήριξη OpenCL (προαιρετικό αλλά ιδιαίτερα συνιστώμενο)
- Αποθηκευτικός χώρος SSD

Εγκατάσταση

```
# Κλωνοποίηση του αποθετηρίου
git clone --recursive https://github.com/sv1btl/PhantomSDR-Plus
cd PhantomSDR-Plus

# Να γίνουν εκτελέσιμα τα σενάρια
chmod +x *.sh

# Εκτέλεση της αυτόματης εγκατάστασης
./install.sh
```

Σημείωση: μετά την εκτέλεση του `install.sh`, επανεκκινήστε το τερματικό σας πριν συνεχίσετε.

Τρέχετε ήδη PhantomSDR-Plus; Μην το εγκαταστήσετε ξανά — ενημερώστε το. Κατεβάστε μία φορά το εργαλείο ενημέρωσης και τρέξτε το· η διαμόρφωσή σας, οι σημάνσεις, ο κωδικός διαχειριστή, η λίστα συχνοτήτων και το ιστορικό συνομιλίας δεν θίγονται ποτέ, και ό,τι έχετε αλλάξει μόνοι σας τίθεται υπόψη σας αντί να αντικατασταθεί:

```
bash cd ~/PhantomSDR-Plus curl -fLO https://raw.githubusercontent.com/sv1btl/PhantomSDR-Plus/main/update.sh chmod +x update.sh ./update.sh
```

Η τελευταία γραμμή απλώς *αναφέρει* τι θα αλλάξει και δεν γράφει τίποτα· το `./update.sh --apply` το εκτελεί. Λεπτομέρειες στο κεφάλαιο *Ενημέρωση του PhantomSDR-Plus*. Η επανάληψη του εγκαταστάτη σε μια στάση που λειτουργεί χρειάζεται μόνο όταν η ανακατασκευή αποτύχει από πακέτα του συστήματος που λείπουν.

Χρησιμοποιήστε το `install_fedora.sh` σε Fedora, το `install_arch.sh` σε Arch ή το `install_opensuse.sh` σε openSUSE Tumbleweed — κάνουν την ίδια δουλειά με διαφορετικό διαχειριστή πακέτων. Κάθε έκδοση Debian και Ubuntu καλύπτεται από το ίδιο το `install.sh`.

Τίποτα δεν χρειάζεται να εγκατασταθεί με το χέρι από πριν. Το script εγκαθιστά τις εξαρτήσεις build και το Node.js, χτίζει το backend, χτίζει τον driver του δέκτη που θα επιλέξετε (και τους κανόνες udev μαζί, στον RX888), ανοίγει το `site_information.json` για να το συμπληρώσετε, χτίζει το frontend, εγκαθιστά OpenCL όταν το υλικό το υποστηρίζει, και έπειτα στήνει το **admin panel**, τον **αποκωδικοποιητή FreeDV RADE** και τον **stats server** — και τα τρία από προεπιλογή, με η τα προσπερνάτε — και κλείνει με ένα πλήρες `recompile.sh`. Είναι διαδραστικό από την αρχή ως το τέλος και διαρκεί από είκοσι λεπτά έως πάνω από μία ώρα.

Για την περιγραφή βήμα προς βήμα, δείτε [INSTALLATION.md](#) → [Τι κάνει το πρόγραμμα εγκατάστασης](#). Για αναλυτικές οδηγίες εγκατάστασης, δείτε το [INSTALLATION.md](#).

Μετρήσεις επιδόσεων

Υλικό	Ρυθμός δειγματοληψίας	Χρήση CPU	Χωρητικότητα χρηστών
Ryzen 5 2600 (όλοι οι πυρήνες)	64 MHz (32 MHz IQ)	38-40 %	100+ χρήστες
AMD RX 580 (GPU)	64 MHz (32 MHz IQ)	28-35 %	100+ χρήστες
Intel i5-6500T (με OpenCL)	60 MHz (30 MHz IQ)	10-12 %	100+ χρήστες

Σημείωση: η επιβάρυνση CPU ανά χρήστη είναι ελάχιστη (<1 % ανά χρήστη) όταν είναι ενεργοποιημένη η επιτάχυνση υλικού.

Χρήση

Βασική λειτουργία

- Ρυθμίστε το SDR σας:** επεξεργαστείτε το κατάλληλο αρχείο ρυθμίσεων (π.χ. `config-rtl.toml`)
- Ενημερώστε τις πληροφορίες του σταθμού:** επεξεργαστείτε το `frontend/site_information.json`
- Εκκινήστε τον διακομιστή:** εκτελέστε το κατάλληλο σενάριο εκκίνησης: `bash ./start-rtl.sh #` Για RTL-SDR `./start-rsp1a.sh #` Για SDRplay RSP1A `./start-airspyhf.sh #` Για Airspy HF+ `./start-fobos-hf.sh #` Για RigExpert Fobos SDR, HF1/HF2 (0-25 MHz) `./start-fobos.sh #` Για RigExpert Fobos SDR, RF (25-6000 MHz) `./start-hackrf.sh #` Για HackRF One `./start-rx888mk2.sh #` Για RX888 MK2. Οι RSP1A, Airspy HF+ και Fobos δουλεύουν μέσω SoapySDR και `rx_sdr`· το πρόγραμμα εγκατάστασης στήνει τον driver τους όταν τους επιλέξετε (επιλογές 3, 6 και 5), ή τρέξτε `./setup-`

rspla.sh, ./setup-airspyhf.sh ή ./setup-fobos.sh σε σταθμό ήδη εγκατεστημένο. Δείτε το [INSTALLATION.md](#). Ο HackRF One (επιλογή 7) χρειάζεται μόνο το πακέτο hackrf της διανομής: ./setup-hackrf.sh. Κάθε σενάριο εκκίνησης είναι **αυτόνομο**: σταματά κάθε στιγμιότυπο που εκτελείται, εκκινεί τον δέκτη + τον spectrumserver, **αποσυνδέεται στο παρασκήνιο** (επιβιώνει από το κλείσιμο του τερματικού) και εκτελεί έναν **watchdog** που επανεκκινεί αυτόματα την αλυσίδα αν τερματιστεί. Η πρόοδος καταγράφεται στο logwebsdr.txt — παρακολουθήστε το με tail -f logwebsdr.txt. Η εκ νέου εκτέλεση ενός σεναρίου εκκίνησης αποτελεί καθαρή επανεκκίνηση, ενώ ένα κλείδωμα flock διασφαλίζει ότι εκτελείται μόνο ένας δέκτης κάθε φορά. Προαιρετικά: SPECTRUM_CORES=0-3 για καρφίτσωμα σε CPU, RX_ARGS="" για αντικατάσταση των ορισμάτων του δέκτη χωρίς επεξεργασία του αρχείου (RX888_ARGS για το RX888), RX_DRIVER=miri|sdrplay για να διαλέξετε τον driver του RSP1A.

4. **Αποκτήστε πρόσβαση στη διεπαφή**: ανοίξτε τον περιηγητή σας στο http://localhost:PORT (η προεπιλεγμένη θύρα διαφέρει ανά ρύθμιση)

Τερματισμός του διακομιστή

Ένα κοινό σενάριο τερματισμού λειτουργεί για **κάθε** δέκτη — σταματά πρώτα τον watchdog (ώστε να μην μπορεί να επανεκκινήσει αυτόματα) και έπειτα τον δέκτη και τον spectrumserver:

```
./stop-websdr.sh
```

Ρυθμίσεις

Βασικά αρχεία ρυθμίσεων

1. **config-[συσκευή].toml** – ρυθμίσεις διακομιστή και SDR
2. Ρυθμίσεις διακομιστή (θύρα, νήματα, ριζικός κατάλογος HTML)
3. Ρυθμίσεις εισόδου (ρυθμός δειγματοληψίας, μέγεθος FFT, συχνότητα)
4. Επιλογές εγγραφής στον κατάλογο WebSDR
5. Ρυθμίσεις συμπίεσης ήχου
6. Μετά την τροποποίησή του, επανεκκινήστε τον διακομιστή.
7. **frontend/site_information.json** – δημόσιες πληροφορίες του σταθμού
8. Στοιχεία χειριστή (διακριτικό κλήσης, e-mail, τοποθεσία)
9. Πληροφορίες υλικού και κεραίας
10. Ρυθμίσεις περιοχής και εύρους ζώνης
11. Ενεργοποίηση/απενεργοποίηση συνομιλίας
12. Μετά την τροποποίησή του, εκτελέστε το recompile.sh σε ένα τερματικό.
13. **markers.json** – δείκτες συχνοτήτων και σχέδιο μπαντών
14. **frontend/src/bands-config.js** – ρύθμιση του σχεδίου μπαντών Το αρχείο αυτό βρίσκεται στο frontend/src/bands-config.js και ορίζει τις μπάντες που δημιουργεί ο SysOp.
15. Θα δείτε κάτι σαν το παρακάτω και είστε ελεύθεροι να το επεξεργαστείτε όπως θέλετε: ``
16. const bands =
17. { ITU: 1, name: '40m', min: -30, max: 110, initFreq: '7120000', publishBand: '1', startFreq: 7000000, endFreq: 7200000, step: 1000, color: 'rgba(50, 168, 72, 0.6)', modes: [{ mode: MODES.CW, startFreq: 7000000, endFreq: 7040000 }, { mode: MODES.LSB, startFreq: 7040000, endFreq: 7200000 }] }, { ITU: 2, name: '40m', min: -30, max: 110, initFreq: '', publishBand: '1', startFreq:

```
7000000, endFreq: 7300000, stepi: 1000, color: 'rgba(50, 168, 72, 0.6)', modes: [ { mode:
MODES.CW, startFreq: 7000000, endFreq: 7050000 }, { mode: MODES.LSB, startFreq: 7050000,
endFreq: 7300000 } ] }, { ITU: 3, name: '40m', min: -30, max: 110, initFreq: '', publishBand: '1',
startFreq: 7000000, endFreq: 7200000, stepi: 1000, color: 'rgba(50, 168, 72, 0.6)', modes: [ { mode:
MODES.CW, startFreq: 7000000, endFreq: 7040000 }, { mode: MODES.LSB, startFreq: 7040000,
endFreq: 7200000 } ] ..... etc ``
```

Όπου:

18. **ITU** είναι η περιοχή στην οποία βρίσκεται ο διακομιστής,
19. **min & max** είναι τα όρια φωτεινότητας του καταρράκτη για τη συγκεκριμένη μπάντα,
20. **initFreq** είναι η αρχική συχνότητα συντονισμού όταν επιλέγεται η μπάντα,
21. **publishBand** καθορίζει αν εμφανίζεται η μπάντα: 1 για ερασιτεχνικές μπάντες, 2 για μπάντες εκπομπής,
22. **startFreq & endFreq** ορίζουν τα όρια της μπάντας,
23. **stepi** είναι το προεπιλεγμένο βήμα της ροδέλας του ποντικιού για τη μπάντα, και **modes** είναι ο προτιμώμενος προεπιλεγμένος τρόπος λειτουργίας για τη μπάντα.
24. Αφού κάνετε αλλαγές, **εκτελέστε το recompile.sh σε ένα τερματικό** από τον φάκελο PhantomSDR-Plus.

Για παραδείγματα ρυθμίσεων, δείτε τα συνοδευτικά υποδείγματα αρχείων:

- config-rtl.toml - ρύθμιση RTL-SDR
- config-rspla.toml - ρύθμιση SDRplay RSP1A
- config-airspyhf.toml - ρύθμιση Airspy HF+
- config-fobos-hf.toml / config-fobos.toml - ρύθμιση RigExpert Fobos SDR (HF / RF)
- config-hackrf.toml - HackRF One
- config-rx888mk2.toml - ρύθμιση RX888 MK2

Κατώτατο όριο καταρράκτη — waterfall.sh

Το waterfall.sh (στη ρίζα του αποθετηρίου) αλλάζει το **προεπιλεγμένο ελάχιστο επίπεδο του καταρράκτη** σε dB — δηλαδή την τιμή από την οποία ξεκινά ο καταρράκτης σε μια νέα επίσκεψη.

Η τιμή αυτή βρίσκεται σε δύο αρχεία πηγαίου κώδικα, σε πέντε διαφορετικά σημεία (frontend/src/waterfall.js → this.minWaterfall, και frontend/src/App.svelte → το αρχικό let min_waterfall, οι δύο περιπτώσεις επαναφοράς «min» και η προεπιλογή των σελιδοδεικτών). Η επεξεργασία τους με το χέρι εύκολα πάει στραβά, γι' αυτό το κάνει το σενάριο για εσάς. Δεν χρησιμοποιεί ποτέ σταθερούς αριθμούς γραμμών — εντοπίζει κάθε σημείο με μοτίβο, ώστε να συνεχίζει να δουλεύει και μετά από ενημερώσεις του κώδικα.

- **Υψηλότερη** τιμή (π.χ. -15) δίνει **πιο σκοτεινό** καταρράκτη.
- **Χαμηλότερη** τιμή (π.χ. -45) δίνει **πιο φωτεινό** καταρράκτη.

```
./waterfall.sh          # διαδραστικά – δείχνει τις τρέχουσες τιμές και ζητά τη νέα
./waterfall.sh -v -15   # ορισμός της τιμής χωρίς ερώτηση
./waterfall.sh -v -15 -y # ... και παράλειψη κάθε επιβεβαίωσης
./waterfall.sh -s       # μόνο εμφάνιση των τρεχουσών τιμών, χωρίς αλλαγή
```

Πριν από την επεξεργασία κρατά αντίγραφο ασφαλείας και των δύο αρχείων με χρονική σήμανση (*.bak-EEEEμμη-ΩΩΛΛΔΔ) και τυπώνει την εντολή επαναφοράς. Τροποποιούνται μόνο οι κυριολεκτικές τιμές που είναι ίσες με την τιμή που χρησιμοποιείται τώρα, οπότε άσχετοι αριθμοί σε αυτά τα αρχεία δεν μπορούν ποτέ να ξαναγραφτούν κατά λάθος. Αν η νέα τιμή δεν εμφανιστεί μετά, το σενάριο

επαναφέρει μόνο του τα αντίγραφα ασφαλείας και τερματίζει με σφάλμα. Προειδοποιεί επίσης όταν τα δύο αρχεία δεν συμφωνούν και ρυθμίζει και τα δύο.

- Μετά την αλλαγή πρέπει να ξαναχτιστεί το frontend — το σενάριο προσφέρεται να εκτελέσει το `recompile.sh` για εσάς (με `-y` παραλείπει την ανακατασκευή, οπότε εκτελέστε μόνοι σας το `./recompile.sh` όταν είστε έτοιμοι).

Προσαρμογή

Αλλαγή της εικόνας φόντου

Αντικαταστήστε το `frontend/src/assets/background.jpg` με την εικόνα της προτίμησής σας (διατηρώντας το ίδιο όνομα αρχείου).

- Μετά την τροποποίηση, εκτελέστε το `recompile.sh` σε ένα τερματικό.

Επιλογή κωδικοποιητή ήχου

Επιλέξτε μεταξύ FLAC και Opus στη ρύθμιση `.toml`:

```
[input]
audio_compression="opus" # or "flac"
```

- Μετά την τροποποίηση, επανεκκινήστε τον διακομιστή.

Βαθμονόμηση του S-meter

Τα δύο όργανα βρίσκονται σκόπιμα σε **ξεχωριστές αλυσίδες**, οπότε ρυθμίζονται με δύο διαφορετικά ρυθμιστικά.

Τα δύο ρυθμιστικά

Και τα δύο βρίσκονται στην ενότητα `[input]` του αρχείου ρυθμίσεων με το οποίο ξεκινά ο server (`config.toml`, `config-rx888mk2.toml` κ.λπ.):

```
[input]
smeter_offset=5          # μόνο η ψηφιακή μπάρα
analog_smeter_offset=5  # μόνο ο αναλογικός δείκτης
```

Διαβάζονται στο `src/websocket.cpp` και στέλνονται στον browser με το πρώτο frame `basic_info`, άρα **αρκεί επανεκκίνηση του server** — δεν χρειάζεται νέο build του frontend.

Αναλογική αλυσίδα

```
εμφανιζόμενα dBm = -130 + (rawDb + analog_smeter_offset + 130) x 1,1
```

- Το οπτικό κέρδος 1,1 εφαρμόζεται *μετά* το offset, με σημείο περιστροφής τα -130 dBm. Έτσι 1 μονάδα `analog_smeter_offset` μετακινεί την ένδειξη κατά 1,1 dB. Για μετατόπιση κατά X dB ισχύει `analog_smeter_offset ~= 0,91 x X`. Η τιμή στο TOML διαβάζεται ως ακέραιος, άρα μόνο ακέραια βήματα.
- Η αντιστοίχιση του δείκτη (`powerFromDbm` στο `frontend/src/lib/SMeterAnalog.svelte`) τοποθετεί το **S9 στα -73 dBm**, στις 60 από 100 μονάδες δείκτη, με καμπύλη `pow(...,0,6)` από -130 έως -73 κάτω από το S9 και `pow(...,0,8)` από -73 έως -13 (S9+60) πάνω από αυτό.
- **Δεν** είναι γραμμική κλίμακα 6 dB ανά μονάδα S, οπότε το ουσιαστικό σημείο αναφοράς είναι το S9 — οι υποδιαίρέσεις κάτω από το S9 δεν θα ακολουθήσουν γεννήτρια σήματος σε βήματα των 6 dB.

Ψηφιακή αλυσίδα

```
value    = (rawDb / 150) x 100 + smeter_offset    -> περιορισμός σε [-100, 0]
segments = round((value + 100) x 35 / 100) + DIGITAL_BAR_TRIM
```

- Πρόσεξε το $/150 \times 100$: η ψηφιακή κλίμακα είναι **συμπιεσμένη στα 2/3** και δεν βλέπει ποτέ ούτε το `analog_smeter_offset` ούτε το οπτικό κέρδος 1,1. Γι' αυτό η μπάρα και ο δείκτης δεν συμφωνούν ποτέ ακριβώς — είναι σκόπιμο.
- 35 τμήματα σε 100 μονάδες σημαίνει ότι ένα τμήμα είναι 2,86 μονάδες offset, περίπου 4,3 dB raw. Άρα `smeter_offset=3` αντιστοιχεί περίπου σε ένα τμήμα.
- Το `DIGITAL_BAR_TRIM` στο `frontend/src/lib/SMeterDigital.svelte` (τώρα 0) μετατοπίζει τη μπάρα κατά ακέραια τμήματα. Η αλλαγή του είναι επέμβαση στον κώδικα και **απαιτεί** νέο build του frontend.

Τα παράθυρα LED τροφοδοτούνται αναλογικά και στα δύο όργανα

Τα παράθυρα dBm / dBμV / SNR / NF και στα **δύο** όργανα διαβάζουν τη βαθμονομημένη αναλογική τιμή και προσθέτουν σταθερό `VISUAL_DBM_OFFSET` ίσο με 5, γραμμένο σταθερά στα `SMeterAnalog.svelte` και `SMeterDigital.svelte`. Επομένως:

- Το `smeter_offset` **δεν** αλλάζει κανέναν αριθμό στα παράθυρα, μόνο το μήκος της μπάρας.
- $\text{dB}\mu\text{V} = \text{dBm} + 107$ (50 ohm), και $\text{NF} = \text{dBm} - \text{SNR}$ εξ ορισμού, άρα το NF ακολουθεί αυτόματα.
- Αν θέλεις σωστούς **αριθμούς** χωρίς να πειράξεις τη θέση του δείκτη, το ρυθμιστικό είναι το `VISUAL_DBM_OFFSET` — αλλά είναι σταθερά στον κώδικα, θέλει νέο build και πρέπει να αλλάξει και στα **δύο** αρχεία για να μείνουν συνεπή.

Διαδικασία

1. Δώσε γνωστή στάθμη στην είσοδο κεραίας (γεννήτρια στα -73 dBm = S9), σε SSB/CW με σταθερό εύρος διόδου. Οι ενδείξεις εξαρτώνται από το εύρος και το AGC, οπότε κλείδωσε πρώτα τρόπο λειτουργίας και bandwidth.
2. Δες τι δείχνει το αναλογικό παράθυρο dBm και θέσε `analog_smeter_offset` $\approx 0,91 \times (-73 - \text{ένδειξη})$. Επανεκκίνησε τον server και ξαναέλεγξε. Μία επανάληψη αρκεί· ο δείκτης πρέπει να κάθεται στο S9.
3. Πήγαινε στο ψηφιακό όργανο και ρύθμισε το `smeter_offset` ώστε το μήκος της μπάρας να δείχνει το S9 εκεί που θέλεις — περίπου 3 μονάδες ανά τμήμα. Αυτό το βήμα είναι καθαρά αισθητικό και δεν επηρεάζει τις ενδείξεις.
4. Χωρίς γεννήτρια, καλή εναλλακτική είναι μια ήσυχη μπάνα με φάρο γνωστής στάθμης, ή απλώς να «δέσεις» το επίπεδο θορύβου σε λογική τιμή (π.χ. -120 dBm στα 20 m με αξιοπρεπή κεραία) — με την επίγνωση ότι έτσι βαθμονομείται όλη η αλυσίδα μαζί με το κέρδος της κεραίας, όχι μόνο ο δέκτης.

Παραλλαγές γραφικού περιβάλλοντος

Και οι τέσσερις παραλλαγές (αναλογικό/ψηφιακό S-meter x διάταξη v1/v2) βρίσκονται σε μία δόμηση. Οι επισκέπτες εναλλάσσονται με το μενού **Φπάνω δεξιά** — τίποτα δεν ξαναφορτώνεται, ο ήχος, ο καταρράκτης και όποιος αποκωδικοποιητής τρέχει συνεχίζουν — και η επιλογή θυμάται ανά φυλλομετρητή.

- Εκτελέστε το `recompile.sh` σε ένα τερματικό για να ορίσετε την *αρχική* παραλλαγή, αυτή που βλέπει ένας επισκέπτης την πρώτη φορά.
- **EDITING_VARIANTS.md** – πώς να επεξεργαστείτε τις παραλλαγές του frontend (S-meter και διάταξη) και να τις ξαναχτίσετε

Τεκμηρίωση

- **INSTALLATION.md** - πλήρης οδηγός εγκατάστασης για διαχειριστές συστήματος
 - **ADMIN_PANEL_SETUP.md** - οδηγός εγκατάστασης για διαχειριστές του πίνακα διαχείρισης
 - **USER_GUIDE.md** - οδηγός τελικού χρήστη για τη λειτουργία του WebSDR
 - **THERMAL_GUARD.md** - εγχειρίδιο διαχειριστή για την προστασία υπερθέρμανσης της CPU: οι τέσσερις λειτουργίες και τι πρέπει να κάνετε σε καθεμία
 - **CONNECTION_LIMITS.md** - εγχειρίδιο διαχειριστή για τα όρια συνδέσεων: προστασία ενός δημόσιου δέκτη από πλημμύρα συνδέσεων
 - **MULTI_RECEIVER.md** - εγχειρίδιο διαχειριστή για πολλούς δέκτες σε έναν υπολογιστή: instances, receivers.toml, η επιλογή δέκτη και το start-all.sh
 - **PROJECT_STRUCTURE.md** - δομή καταλόγων και οργάνωση του κώδικα
 - **EDITING_VARIANTS.md** - πώς να επεξεργαστείτε τις παραλλαγές του frontend (S-meter και διάταξη) και να τις ξαναχτίσετε
-

Διαδικτυακός κατάλογος WebSDR

Καταχωρίστε τον WebSDR σας στον επίσημο κατάλογο: <https://sdr-list.xyz>

Ορίστε `register_online=true` στο αρχείο ρυθμίσεων `.toml` για αυτόματη καταχώριση.

Αντιμετώπιση προβλημάτων

Συνήθη προβλήματα

Το OpenCL δεν λειτουργεί

- Βεβαιωθείτε ότι οι οδηγοί έχουν εγκατασταθεί σωστά
- Ελέγξτε με την εντολή `clinfo`
- Δείτε το INSTALLATION.md για αναλυτική εγκατάσταση του OpenCL

Προβλήματα καθυστέρησης ήχου

- Κρατήστε το FLAC ως κωδικοποιητή: είναι αυτός με τη μικρότερη καθυστέρηση, ενώ το Opus κρατά πίσω έως 20 ms ανά πακέτο
- Ανοίξτε τη σελίδα προσθέτοντας `?diag=1` στη διεύθυνση (για παράδειγμα `/mobile?diag=1`) για να δείτε την καθυστέρηση χωρισμένη σε πακέτο, buffer και έξοδο· το buffer μικραίνει μόνο του μέσα σε ένα-δυο λεπτά μετά από μια διαταραχή του δικτύου
- Αν ο ήχος διακόπτεται, ανεβάστε το buffer μία βαθμίδα (δείτε τον Οδηγό χρήσης, *Buffer ήχου και καθυστέρηση*)
- Εξασφαλίστε επαρκείς πόρους CPU/GPU

Αποτυχίες μεταγλώττισης

- Επαληθεύστε ότι έχουν εγκατασταθεί όλες οι εξαρτήσεις
 - Δοκιμάστε να καθαρίσετε τον κατάλογο μεταγλώττισης: `rm -rf build && meson setup build`
 - Ελέγξτε για συγκρουόμενες εκδόσεις βιβλιοθηκών
-

Συνεισφορά

Οι συνεισφορές είναι ευπρόσδεκτες! Πρόκειται για ανεξάρτητο fork με πρόσθετες δυνατότητες. Παρακαλούμε:

1. Κάντε fork το αποθετήριο
2. Δημιουργήστε έναν κλάδο για τη λειτουργία
3. Κάντε commit τις αλλαγές σας
4. Κάντε push στον κλάδο
5. Δημιουργήστε ένα Pull Request

Άδεια χρήσης

Το έργο διατίθεται υπό την άδεια GNU General Public License v3.0 – δείτε το αρχείο [LICENSE](#) για λεπτομέρειες.

Συντελεστές και ευχαριστίες

- **SV1BTL & SV2AMK** – ανάπτυξη και βελτιώσεις του PhantomSDR-Plus
- Βασισμένο στο αρχικό έργο PhantomSDR

Σύνδεσμοι

- **Ζωντανή επίδειξη:** <http://phantomsdr.no-ip.org:8900/>
- **Κατάλογος WebSDR:** <https://sdr-list.xyz>
- **Αποθετήριο GitHub:** <https://github.com/sv1btl/PhantomSDR-Plus>

Υποστήριξη

- **Ζητήματα:** αναφέρετε σφάλματα μέσω των [GitHub Issues](#)
- **E-mail:** επικοινωνία sv1btl@otenet.gr

Συμβουλές επιδόσεων

1. **Ενεργοποιήστε OpenCL/CUDA** για επιτάχυνση GPU – μειώνει δραματικά τη χρήση της CPU
2. **Χρησιμοποιήστε αποθηκευτικό χώρο SSD** για καλύτερες επιδόσεις εισόδου/εξόδου
3. **Διαθέστε επαρκή RAM** – συνιστώνται τουλάχιστον 8 GB
4. **Βελτιστοποιήστε το μέγεθος της FFT** – μεγαλύτερη FFT = καλύτερη ανάλυση αλλά περισσότερη χρήση CPU
5. **Εξετάστε μια αποκλειστική GPU** – AMD ή NVIDIA με υποστήριξη OpenCL

73 de SV1BTL & SV2AMK

Για αναλυτικές οδηγίες εγκατάστασης, δείτε το *INSTALLATION.md* Για τον οδηγό λειτουργίας τελικού χρήστη, δείτε το *USER_GUIDE.md*

ΚΕΦΑΛΑΙΟ 2

Οδηγός εγκατάστασης

INSTALLATION.md

Οδηγός εγκατάστασης PhantomSDR-Plus για διαχειριστές συστήματος

Αυτός ο πλήρης οδηγός θα σας καθοδηγήσει στην εγκατάσταση και ρύθμιση του PhantomSDR-Plus στον διακομιστή σας.

Πίνακας περιεχομένων

1. Απαιτήσεις συστήματος
2. Προετοιμασία πριν την εγκατάσταση
3. Εγκατάσταση του PhantomSDR-Plus — το script και τι κάνει
4. Autorun Spot Reporter (FT8/FT4/WSPR)
5. Εξομοίωση πελατών KiwiSDR (προαιρετικό)
6. Ρυθμίσεις
7. Ρύθμιση ανά συσκευή SDR
8. Δοκιμή και επαλήθευση
9. Ρύθμιση αυτόματης εκκίνησης
10. Θερμική προστασία CPU
11. Αντιμετώπιση προβλημάτων

Μόνο για αναφορά — το πρόγραμμα εγκατάστασης τα κάνει ήδη όλα αυτά για εσάς. Διαβάστε αυτές τις ενότητες αν βρίσκεστε σε διανομή που δεν καλύπτει κανένα από τα scripts, ή αν χρειάζεται να επισκευάσετε ένα βήμα με το χέρι:

- Εγκατάσταση εξαρτήσεων
- Εγκατάσταση Node.js και npm
- Εγκατάσταση OpenCL
- Χειροκίνητο build
- Εγκατάσταση του κωδικοποιητή ήχου Opus

Απαιτήσεις συστήματος

Υποστηριζόμενα λειτουργικά συστήματα

Κύριο (συνιστώμενο):

- Ubuntu 22.04 LTS (Jammy), 24.04 LTS (Noble) και 26.04 LTS (Resolute) — συνιστάται η 24.04
- Debian 12 (Bookworm) και Debian 13 (Trixie)

Εναλλακτικό:

- Fedora (τελευταία σταθερή έκδοση)

- Arch Linux (κυλιόμενη έκδοση)
- openSUSE Tumbleweed (όχι Leap — δείτε τη σημείωση παρακάτω)

Απαιτήσεις υλικού

Ελάχιστη διαμόρφωση:

- CPU: επεξεργαστής διπλού πυρήνα (2+ GHz)
- RAM: 4 GB
- Αποθηκευτικός χώρος: 10 GB ελεύθερα
- Δίκτυο: σύνδεση 100 Mbps

Συνιστώμενη διαμόρφωση:

- CPU: τετραπύρηνος ή καλύτερος (Ryzen 5 2600, Intel i5-6500T ή καλύτερος)
- RAM: 8 GB ή περισσότερα
- Αποθηκευτικός χώρος: SSD 20 GB και άνω
- GPU: AMD/NVIDIA με υποστήριξη OpenCL (ιδιαίτερα συνιστώμενο)
- Δίκτυο: σύνδεση 1 Gbps

Διαμόρφωση υψηλών επιδόσεων:

- CPU: 6+ πυρήνες (Ryzen 7, Intel i7 ή καλύτερος)
- RAM: 16 GB ή περισσότερα
- Αποθηκευτικός χώρος: NVMe SSD
- GPU: αποκλειστική GPU με υποστήριξη OpenCL/CUDA
- Δίκτυο: 1 Gbps ή καλύτερο

Προετοιμασία πριν την εγκατάσταση

1. Ενημερώστε το σύστημά σας

```
# Ubuntu/Debian
sudo apt update
sudo apt upgrade -y
sudo reboot

# Fedora
sudo dnf update -y
sudo reboot
```

2. Επαληθεύστε την έκδοση του Ubuntu

```
lsb_release -a
```

Η αναμενόμενη έξοδος πρέπει να δείχνει: Ubuntu 24.04 LTS

3. Ελέγξτε τον διαθέσιμο χώρο στον δίσκο

```
df -h
```

Βεβαιωθείτε ότι έχετε τουλάχιστον 10 GB ελεύθερα στον προσωπικό σας κατάλογο.

4. Ελέγξτε τις πληροφορίες της CPU

```
lscpu
```

Σημειώστε τον αριθμό των πυρήνων/νημάτων για τη βελτιστοποίηση των ρυθμίσεων.

Εγκατάσταση εξαρτήσεων

ΣΗΜΑΝΤΙΚΟ

Δεν χρειάζεστε αυτή την ενότητα για μια κανονική εγκατάσταση. Το `./install.sh` — ή το script της διανομής σας — τα κάνει όλα αυτά για εσάς· δείτε [Τι κάνει το πρόγραμμα εγκατάστασης](#). Όσα ακολουθούν είναι αναφορά για χειροκίνητη εγκατάσταση, για διανομή που δεν καλύπτει κανένα script, ή για την επισκευή ενός βήματος με το χέρι.

Ubuntu 24.04 LTS

```
sudo apt install -y \
  build-essential \
  cmake \
  pkg-config \
  meson \
  libfftw3-dev \
  libwebsocketpp-dev \
  libflac++-dev \
  zlib1g-dev \
  libzstd-dev \
  libboost-all-dev \
  libopus-dev \
  libliquid-dev \
  git \
  psmisc \
  wget \
  curl
```

Fedora

```
sudo dnf install -y \
  g++ \
  meson \
  cmake \
  fftw3-devel \
  websocketpp-devel \
  flac-devel \
  zlib-devel \
  boost-devel \
  libzstd-devel \
  opus-devel \
  liquid-dsp-devel \
  git \
  psmisc \
  wget \
  curl
```

Επαληθεύστε την εγκατάσταση

```
# Check if key dependencies are installed
pkg-config --modversion fftw3
cmake --version
meson --version
```

Εγκατάσταση Node.js και npm

ΣΗΜΑΝΤΙΚΟ

Δεν χρειάζεστε αυτή την ενότητα για μια κανονική εγκατάσταση. Το `./install.sh` — ή το script της διανομής σας — τα κάνει όλα αυτά για εσάς· δείτε [Τι κάνει το πρόγραμμα εγκατάστασης](#). Όσα ακολουθούν είναι αναφορά για χειροκίνητη εγκατάσταση, για διανομή που δεν καλύπτει κανένα script, ή για την επισκευή ενός βήματος με το χέρι.

Το PhantomSDR-Plus απαιτεί Node.js για τη μεταγλώττιση του frontend. Θα χρησιμοποιήσουμε το NVM (Node Version Manager) για την εγκατάσταση.

1. Εγκαταστήστε το NVM

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.4/install.sh | bash
```

2. Φορτώστε το NVM

ΣΗΜΑΝΤΙΚΟ: κλείστε και ανοίξτε ξανά το τερματικό σας ή εκτελέστε:

```
export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"
```

3. Επαληθεύστε την εγκατάσταση του NVM

```
nvm --version
```

Αναμενόμενη έξοδος: 0.40.4 ή παρόμοια

4. Εγκαταστήστε το Node.js

```
# Install Node.js 22 - the version every install script requires
nvm install 22

# Make it the default, so systemd units and cron see it too
nvm alias default 22

# Verify installation
node --version
npm --version
```

5. Προαιρετικά: εγκαταστήστε επιπλέον εκδόσεις του Node

```
# Install latest version
nvm install node

# Install specific version (if needed)
nvm install 22.22.3

# List installed versions
nvm list

# Use specific version
nvm use 22
```

Εγκατάσταση OpenCL (προαιρετικό αλλά συνιστώμενο)

ΣΗΜΑΝΤΙΚΟ

Δεν χρειάζεστε αυτή την ενότητα για μια κανονική εγκατάσταση. Το `./install.sh` — ή το script της διανομής σας — τα κάνει όλα αυτά για εσάς· δείτε [Τι κάνει το πρόγραμμα εγκατάστασης](#). Όσα ακολουθούν είναι αναφορά για χειροκίνητη εγκατάσταση, για διανομή που δεν καλύπτει κανένα script, ή για την επισκευή ενός βήματος με το χέρι.

Το OpenCL βελτιώνει δραματικά τις επιδόσεις μεταφέροντας τους υπολογισμούς FFT στην GPU. Η ενότητα αυτή καλύπτει τα ενσωματωμένα γραφικά Intel. Για GPU AMD/NVIDIA ανατρέξτε στην τεκμηρίωση του κατασκευαστή.

CPU Intel με ενσωματωμένα γραφικά

1. Εγκαταστήστε τα βασικά στοιχεία OpenCL

```
sudo apt install -y \
  build-essential \
  cmake \
  pkg-config \
  meson \
  libfftw3-dev \
  libwebsocketpp-dev \
  libflac++-dev \
  zlib1g-dev \
  libzstd-dev \
  libboost-all-dev \
  libopus-dev \
  libliquid-dev \
  libclfft-dev \
  ocl-icd-opencl-dev \
  clinfo
```

2. Κατεβάστε το Intel Compute Runtime

```
mkdir -p ~/neo
cd ~/neo

wget https://github.com/intel/compute-runtime/releases/download/19.07.12410/intel-
gmmlib_18.4.1_amd64.deb
wget https://github.com/intel/compute-runtime/releases/download/19.07.12410/intel-igc-
core_18.50.1270_amd64.deb
wget https://github.com/intel/compute-runtime/releases/download/19.07.12410/intel-igc-
opencl_18.50.1270_amd64.deb
wget https://github.com/intel/compute-runtime/releases/download/19.07.12410/intel-
opencl_19.07.12410_amd64.deb
wget https://github.com/intel/compute-runtime/releases/download/19.07.12410/intel-
ocloc_19.07.12410_amd64.deb
```

3. Εγκαταστήστε το Intel OpenCL Runtime

```
sudo dpkg -i *.deb
```

Αν προκύψουν σφάλματα εξαρτήσεων:

```
sudo apt --fix-broken install
```

4. Εγκαταστήστε τον φορτωτή OpenCL ICD

```
sudo apt update
sudo apt install intel-opencl-icd
```

Αν προκύψουν σφάλματα:

```
sudo apt --fix-broken install
sudo apt install intel-opencl-icd
```

5. Επαληθεύστε την εγκατάσταση του OpenCL

```
sudo clinfo
```

Θα πρέπει να δείτε πληροφορίες για την πλατφόρμα και τις συσκευές OpenCL. Αναζητήστε:

- Αριθμό πλατφορμών: 1 (ή περισσότερες)
- Όνομα πλατφόρμας: Intel(R) OpenCL (ή παρόμοιο)
- Τύπο συσκευής: GPU ή CPU

6. Επανεκκινήστε

```
sudo reboot
```

OpenCL σε GPU AMD

Για GPU AMD, εγκαταστήστε το ROCm:

```
# Add ROCm repository
wget -q -O - https://repo.radeon.com/rocm/rocm.gpg.key | sudo apt-key add -
echo 'deb [arch=amd64] https://repo.radeon.com/rocm/apt/debian/ ubuntu main' | sudo tee /etc/apt/sources.list.d/rocm.list

# Install ROCm
sudo apt update
sudo apt install rocm-ocl rocm-clinfo

# Add user to video group
sudo usermod -a -G video $USER

# Reboot
sudo reboot

# Verify
clinfo
```

OpenCL σε GPU NVIDIA

Για GPU NVIDIA, εγκαταστήστε το CUDA:

```
# Install NVIDIA drivers
sudo apt install nvidia-driver-525

# Install CUDA toolkit
wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64/cuda-keyring_1.0-1_all.deb
sudo dpkg -i cuda-keyring_1.0-1_all.deb
sudo apt update
sudo apt install cuda

# Reboot
sudo reboot

# Verify
nvidia-smi
clinfo
```

Εγκατάσταση του PhantomSDR-Plus

Κλωνοποίηση, εκτέλεση scripts, εκτέλεση του προγράμματος εγκατάστασης

```
cd ~
git clone --recursive https://github.com/svlbtl/PhantomSDR-Plus
cd PhantomSDR-Plus
chmod +x *.sh
./install.sh
```

⚠ **ΣΗΜΑΝΤΙΚΟ:** Μόλις τελειώσει η εγκατάσταση, **επανεκκινήστε το terminal** — τα νεοεγκατεστημένα Node.js και Rust δεν βρίσκονται στο `PATH` σας πριν από αυτό.

Τρέχετε ήδη PhantomSDR-Plus; Μην το εγκαταστήσετε ξανά — ενημερώστε το. Κατεβάστε μία φορά το εργαλείο ενημέρωσης και τρέξτε το· η διαμόρφωσή σας, οι σημάνσεις, ο κωδικός διαχειριστή, η λίστα συχνοτήτων και το ιστορικό συνομιλίας δεν θίγονται ποτέ, και ό,τι έχετε αλλάξει μόνοι σας τίθεται υπόψη σας αντί να αντικατασταθεί:

```
bash cd ~/PhantomSDR-Plus curl -fLO https://raw.githubusercontent.com/svlbtl/PhantomSDR-Plus/main/update.sh chmod +x update.sh ./update.sh
```

Η τελευταία γραμμή απλώς αναφέρει τι θα αλλάξει και δεν γράφει τίποτα· το `./update.sh --apply` το εκτελεί. Λεπτομέρειες στο κεφάλαιο *Ενημέρωση του PhantomSDR-Plus*. Η επανάληψη του εγκαταστάτη σε μια στάση που λειτουργεί χρειάζεται μόνο όταν η ανακατασκευή αποτύχει από πακέτα του συστήματος που λείπουν.

Χρησιμοποιήστε το script που ταιριάζει στο σύστημά σας. Κάνουν την ίδια δουλειά και κάνουν τις ίδιες ερωτήσεις· αλλάζει μόνο ο διαχειριστής πακέτων:

Σύστημα	Script
Ubuntu 22.04 / 24.04 / 26.04, Debian 12 / 13	<code>./install.sh</code>
Fedora	<code>./install_fedora.sh</code>
Arch	<code>./install_arch.sh</code>
openSUSE Tumbleweed	<code>./install_opensuse.sh</code>

Ένα πρόγραμμα εγκατάστασης για κάθε έκδοση Debian και Ubuntu. Το `install.sh` διαβάζει το `/etc/os-release` και την εγκατεστημένη έκδοση του Boost και προσαρμόζεται μόνο του· τα παλιά `install_ubuntu22.sh`, `install-Deb12.sh` και `install_ubuntu26.sh` καταργήθηκαν. Θέτει `DEBIAN_FRONTEND=noninteractive`, ώστε το `tzdata` που έρχεται μαζί με το `python3-matplotlib` να μην μπορεί να σταματήσει την εκτέλεση ζητώντας τη ζώνη ώρας σας και στη συνέχεια να καταπιεί την απάντηση που προοριζόταν για την επόμενη ερώτηση. Στο Jammy καλεί απευθείας το `install_rade_ubuntu22.sh`, επειδή το `python3-websockets` του Jammy είναι 10.1 ενώ το RADE χρειάζεται 11.0 ή νεότερο. Γνωρίζει πού βρίσκεται το `intel-ocl-icd` σε κάθε έκδοση: στα αποθετήρια σε 22.04 και 26.04, στο `non-free` στο Debian 12, και στο αποθετήριο γραφικών της Intel σε 24.04 και Debian 13. Και όπου το Boost είναι 1.87 ή νεότερο, το patch των κεφαλίδων `websocketpp` παύει να είναι προαιρετικό — το πρόγραμμα εγκατάστασης επαληθεύει ότι εφαρμόστηκε και αρνείται να χτίσει χωρίς αυτό. Δεν χρειάζεται ποτέ νεότερος μεταγλωττιστής: ο προεπιλεγμένος GCC δέχεται `-std=c++23` και στις πέντε εκδόσεις, οπότε μην εγκαταστήσετε `gcc-12` γι' αυτό.

Η εγκατάσταση εκτελείται σε 17 σαφώς αριθμημένα βήματα, και κάθε σημείο όπου περιμένει εσάς περιβάλλεται από ένα πλαίσιο  **ΧΡΕΙΑΖΕΤΑΙ Η ΕΙΣΟΔΟΣ ΣΑΣ**, ώστε μια ερώτηση να μην μπορεί να περάσει απαρατήρητη μέσα στην κύλιση. Οι επτά ερωτήσεις παρατίθενται από την αρχή, πριν εγκατασταθεί οτιδήποτε. Το `PHANTOM_NONINTERACTIVE=1` τις απαντά όλες με τις προεπιλογές τους· δείτε την κεφαλίδα του `install.sh` για τις μεταβλητές `PHANTOM_*`.

Κάθε εκτέλεση γράφει το `install.txt`. Όταν το πρόγραμμα εγκατάστασης τελειώσει — ή διακοπεί στη μέση — γράφει μια αναφορά στο `install.txt` μέσα στον φάκελο `PhantomSDR-Plus`: το αποτέλεσμα, καθένα από τα 17 βήματα ως `OK` / `SKIPPED` / `PARTIAL` / `FAILED`, ό,τι εντόπισε (διανομή, Boost, μεταγλωττιστή, Node.js), ποια στοιχεία εγκαταστάθηκαν, και κάθε προειδοποίηση που εμφανίστηκε. Μια εκτέλεση που αποτυγχάνει αφήνει αναφορά που σταματά στο βήμα που απέτυχε, με την αιτία και τη σημείωση ότι το πρόγραμμα εγκατάστασης μπορεί να ξανατρέξει με ασφάλεια. Είναι το πρώτο αρχείο που διαβάζετε όταν κάτι δεν δούλεψε, και το πρώτο που επισυνάπτετε σε αναφορά σφάλματος. Κάθε εκτέλεση το αντικαθιστά, οπότε κρατήστε αντίγραφο αν θέλετε να συγκρίνετε δύο εγκαταστάσεις.

Τα Ubuntu 26.04, Arch και openSUSE Tumbleweed μεταγλωττίζονται, αλλά δεν έχουν δοκιμαστεί στον αέρα. Και τα τρία διαθέτουν Boost νεότερο από 1.87, που κατέργησε το API `io_service` για το οποίο γράφτηκε το ενσωματωμένο `websocketpp` 0.8.2. Οι διορθωμένες κεφαλίδες που αντιγράφει το πρόγραμμα εγκατάστασης γεφυρώνουν αυτό το κενό (`io_context`, `executor_work_guard`, `boost::asio::post`, ο σύγχρονος `resolver`), και όπου το Boost είναι 1.87 ή νεότερο το πρόγραμμα θεωρεί το patch υποχρεωτικό και όχι προαιρετικό — επαληθεύει ότι εφαρμόστηκε και αρνείται να χτίσει χωρίς αυτό. Μια πλήρης εγκατάσταση, μαζί με τον οδηγό του δέκτη και όλα τα προαιρετικά στοιχεία, έχει επαληθευτεί από άκρη σε άκρη σε containers με Boost 1.90 (Ubuntu 26.04), 1.91 (openSUSE Tumbleweed) και 1.92 (Arch). Αυτό αποδεικνύει ότι ο διακομιστής μεταγλωττίζεται και ξεκινά — όχι ότι εξυπηρετεί δέκτη για ώρες. Θεωρήστε και τα τρία μη δοκιμασμένα σε παραγωγή μέχρι να αναφέρει κάποιος.

openSUSE εδώ σημαίνει Tumbleweed. Εκεί έχει επαληθευτεί το `install_opensuse.sh`. Το Leap 15.6 δεν δουλεύει: τα αποθετήριά του δεν έχουν καθόλου `liquid-dsp-devel`, που το χρειάζεται το backend, ενώ το Boost υπάρχει μόνο με ονόματα πακέτων που φέρουν έκδοση. Η υποστήριξη του Leap θα απαιτούσε προσθήκη εξωτερικών αποθετηρίων OBS, οπότε προς το παρόν μένει εκτός.

Τι κάνει το πρόγραμμα εγκατάστασης

Τίποτα δεν χρειάζεται να ετοιμαστεί με το χέρι από πριν — καμία λίστα πακέτων για αντιγραφή, κανένα Node.js για κατέβασμα, κανένα κυνήγι πακέτων OpenCL. Εκτελείται σε 19 αριθμημένα βήματα και σταματά για έως δέκα ερωτήσεις, καθεμία μέσα σε πλαίσιο «ΧΡΕΙΑΖΕΤΑΙ Η ΕΙΣΟΔΟΣ ΣΑΣ» — οπότε είτε μέινετε στο πληκτρολόγιο, είτε θέστε `PHANTOM_NONINTERACTIVE=1` και αφήστε το να απαντήσει σε όλα με τις προεπιλογές (δείτε παρακάτω), και υπολογίστε από περίπου είκοσι λεπτά έως αρκετά πάνω από μία ώρα, ανάλογα με το μηχάνημα και το πόσα extras θα κρατήσετε.

#	Βήμα	Τι σας ρωτά
1	Απαριθμεί τις υπηρεσίες PhantomSDR-Plus που τρέχουν αυτή τη στιγμή — admin panel, reverse proxy, stats server, δέκτης — και προσφέρεται να τις σταματήσει πριν αγγίξει οτιδήποτε. Μόνο ένα script εκκίνησης που τρέχει μετράει ως δέκτης· ένα που είναι απλώς ανοιχτό σε επεξεργαστή κειμένου δεν αγγίζεται.	επιβεβαίωση, προεπιλογή ναι
2-6	Εντοπίζει τη διανομή και εγκαθιστά όλες τις εξαρτήσεις build (compiler, meson/ninja, FFTW, Boost, FLAC, Opus, liquid-dsp, zlib/zstd, libcurl ...), εγκαθιστώντας και Node.js 22 μέσω nvm αν το σύστημα δεν έχει ή έχει παλαιότερο	τίποτα
7	Χτίζει το backend με meson	τίποτα
8	Χτίζει τον driver του δέκτη σας — RX888 MkII / RX888, RTL-SDR (το Blog V4 ρωτιέται χωριστά), SDRplay RSP1A, RigExpert Fobos SDR, Airspy HF+, HackRF One ή κανέναν. Με τον RX888 εγκαθιστά και τους κανόνες udev , ώστε ο server να μη χρειάζεται ποτέ sudo για τη συσκευή. Οι RSP1A, Fobos και Airspy HF+ δουλεύουν μέσω SoapySDR: η επιλογή τους τρέχει το setup-rsp1a.sh, το setup-fobos.sh ή το setup-airspyhf.sh, που χτίζει τον driver και το rx_sdr και εγκαθιστά επίσης κανόνα udev (δείτε Δέκτες μέσω SoapySDR (RSP1A, Fobos, Airspy HF+)) Ο HackRF One δεν χρειάζεται SoapySDR: η επιλογή του τρέχει το setup-hackrf.sh, που εγκαθιστά το πακέτο hackrf της διανομής και έναν κανόνα udev.	ποιον SDR έχετε
9	Ανοίγει το frontend/site_information.json στον editor σας	διακριτικό, locator, εξοπλισμός, κεραία — μην το προσπεράσετε
10-11	Εγκαθιστά τις εξαρτήσεις του frontend και χτίζει τις σελίδες desktop και /mobile	τίποτα
12	Εγκαθιστά OpenCL, επιλέγοντας τον provider από το υλικό που βρίσκει (GPU Intel / AMD / NVIDIA, ή το x86 CPU runtime). Αν δεν υπάρχει συσκευή με OpenCL, σας το λέει και προχωράει	επιβεβαίωση, προεπιλογή ναι
13-15	Εγκαθιστά το admin panel , τον αποκωδικοποιητή FreeDV RADE V1 και τον stats server — και τα τρία από προεπιλογή	επιβεβαίωση για καθένα, προεπιλογή ναι· το καθένα έχει δικές του ερωτήσεις
16	Ξαναεφαρμόζει τα πέντε διορθωμένα headers του websocketpp πάνω στο subproject του meson και επαληθεύει ότι εφαρμόστηκαν. Τα τρία από αυτά είναι η δουλειά συμβατότητας με Boost ≥ 1.87, χωρίς την οποία το backend δεν μεταγλωττίζεται σε Boost 1.90· τα άλλα δύο είναι αλλαγές του ίδιου του έργου, η μία εκ των οποίων είναι η διόρθωση που χρειάζεται η εγγραφή στο websdr.org	τίποτα
17	Εγκαθιστά την εξομίωση πελατών KiwiSDR εκτελώντας το kiwi_install.sh, ώστε πελάτες Kiwi όπως το AetherSDR να μπορούν να συνδεθούν σε αυτόν τον δέκτη. Διορθώνει τις πηγές και προσθέτει [kiwi_emulation] στα αρχεία ρυθμίσεων στη ρίζα του αποθετηρίου — δείτε Εξομίωση πελατών KiwiSDR	επιβεβαίωση, προεπιλογή ναι
18	Εκτελεί το recompile.sh, ώστε όλα να χτιστούν από τις διορθωμένες πηγές	[3] Both backend and frontend → αρχική παραλλαγή → [1] build-all.sh
19	Τυπώνει τη σύνοψη: κάθε βήμα με το αποτέλεσμά του και κάθε στοιχείο με το τι εγκαταστάθηκε	τίποτα

Εγκατάσταση χωρίς επίβλεψη

Κάθε ερώτηση έχει μια μεταβλητή περιβάλλοντος που την παρακάμπτει, και το πρόγραμμα εγκατάστασης περνά μόνο του στις προεπιλογές όταν η είσοδος δεν είναι τερματικό (σωλήνας,

container, εργασία CI). Ισχύει το ίδιο και στα τέσσερα: `install.sh`, `install_arch.sh`, `install_fedora.sh` και `install_opensuse.sh`. Θέστε `PHANTOM_NONINTERACTIVE=1` και η εκτέλεση ολοκληρώνεται χωρίς καμία ερώτηση:

Μεταβλητή	Τι κάνει
<code>PHANTOM_NONINTERACTIVE=1</code>	answer every question with its default
<code>PHANTOM_SDR=1...7</code>	RX888 · RTL-SDR · SDRplay RSP1A · skip · Fobos · Airspy HF+ · HackRF (default 4)
<code>PHANTOM_RTLSDR_V4=y\ n</code>	RTL-SDR Blog V4 driver (default n)
<code>PHANTOM_SITE_EDIT=y\ n</code>	open <code>site_information.json</code> in an editor
<code>PHANTOM_OPENCL=y\ n</code>	install OpenCL (default y)
<code>PHANTOM_OPENCL_PROVIDER=1\ 2\ 3</code>	Intel · Mesa/Rusticl · POCL (default: from the detected hardware)
<code>PHANTOM_ADMIN=y\ n</code>	admin panel (y interactive, n unattended)
<code>PHANTOM_RADE=y\ n</code>	RADE / FreeDV (y interactive, n unattended)
<code>PHANTOM_STATS=y\ n</code>	statistics server (y interactive, n unattended)
<code>PHANTOM_KIWI=y\ n</code>	KiwiSDR client emulation (default y)
<code>PHANTOM_RECOMPILE=y\ n</code>	final rebuild (default y)
<code>PHANTOM_CURLPP=y\ n</code>	συνέχεια χωρίς <code>curlpp</code> — μόνο Arch και openSUSE (προεπιλογή y)
<code>PHANTOM_FIX_CLOCK_SKEW=y\ n</code>	επαναφορά των χρονοσημάνσεων των πηγαίων αρχείων που είναι στο μέλλον, ώστε να μπορέσει να χτίσει το <code>meson</code> (προεπιλογή y)

Τα τρία υπο-προγράμματα με σήμανση *n χωρίς επίβλεψη* είναι διαδραστικά από μόνα τους, οπότε μια εκτέλεση χωρίς επίβλεψη τα παραλείπει αντί να κολλήσει στις ερωτήσεις τους. Ονομάστε τα ρητά για να συμπεριληφθούν:

```
# a complete unattended install for an RX888 MkII
PHANTOM_NONINTERACTIVE=1 PHANTOM_SDR=1 ./install.sh

# ... and with the admin panel, RADE and the statistics server too
PHANTOM_NONINTERACTIVE=1 PHANTOM_SDR=1 \
  PHANTOM_ADMIN=y PHANTOM_RADE=y PHANTOM_STATS=y ./install.sh
```

Μετά από μια επανεκκίνηση οι κανόνες udev του RX888 και ο όποιος driver OpenCL είναι πλήρως ενεργοί· το script σας λέει πότε χρειάζεται επανεκκίνηση ή νέα σύνδεση χρήστη.

Όταν στήνει τον πίνακα διαχείρισης, ο installer προσφέρει και τις δύο μονάδες `systemd` και **συνιστά έντονα** να τις δεχτείτε: η προστασία υπερθέρμανσης της CPU τρέχει μέσα στον πίνακα, οπότε χωρίς αυτές μια επανεκκίνηση ή μια κατάρρευση αφήνει το μηχάνημα απροστάτευτο. Το βήμα παραλείπεται αυτόματα όπου δεν τρέχει `systemd` (containers, WSL1, OpenRC) — δείτε τον [οδηγό του πίνακα](#).

Κάθε extra είναι ένα κανονικό script που μπορείτε να τρέξετε και μόνο του, οποτεδήποτε — `./setup_admin.sh` ([οδηγός](#)), `./install_rade.sh` ([οδηγός](#)), `./install-stats-server.sh` ([οδηγός](#)), `./setup-rx888-udev.sh` και τα scripts των δεκτών `./setup-rsp1a.sh`, `./setup-fobos.sh`, `./setup-airspyhf.sh` και `./setup-hackrf.sh` — αυτά ακριβώς καλεί το πρόγραμμα εγκατάστασης.

Χειροκίνητο build (αναφορά)

ΣΗΜΑΝΤΙΚΟ

Δεν χρειάζεστε αυτή την ενότητα για μια κανονική εγκατάσταση. Το `./install.sh` — ή το script της διανομής σας — τα κάνει όλα αυτά για εσάς· δείτε [Τι κάνει το πρόγραμμα εγκατάστασης](#). Όσα ακολουθούν είναι αναφορά για χειροκίνητη εγκατάσταση, για διανομή που δεν καλύπτει κανένα script, ή για την επισκευή ενός βήματος με το χέρι.

Αν ποτέ χρειαστεί να χτίσετε χωρίς το script εγκατάστασης — σε διανομή που δεν καλύπτεται, ή για να επισκευάσετε ένα μισοτελειωμένο build:

Μεταγλώττιση του backend

```
# Clean any previous builds, Configure build with optimization, Compile
rm -rf build
meson setup build --buildtype=release --optimization=3
meson compile -C build

# Verify binary was created
ls -lh build/spectrumserver
```

Μεταγλώττιση του frontend

```
cd frontend

# Install dependencies, Fix any vulnerabilities, Build the frontend
npm install
npm audit fix
npm run build

# Return to project root
cd ..
```

Επαληθεύστε την εγκατάσταση

```
# Check if binary exists
ls -lh build/spectrumserver

# Check if frontend was built
ls -lh frontend/dist/

# List important files
ls -lh *.sh *.toml
```

Εγκατάσταση του κωδικοποιητή ήχου Opus

ΣΗΜΑΝΤΙΚΟ

Δεν χρειάζεστε αυτή την ενότητα για μια κανονική εγκατάσταση. Το `./install.sh` — ή το script της διανομής σας — τα κάνει όλα αυτά για εσάς· δείτε [Τι κάνει το πρόγραμμα εγκατάστασης](#). Όσα ακολουθούν είναι αναφορά για χειροκίνητη εγκατάσταση, για διανομή που δεν καλύπτει κανένα script, ή για την επισκευή ενός βήματος με το χέρι.

Ο Opus προσφέρει καλύτερη ποιότητα ήχου και μικρότερη καθυστέρηση σε σχέση με το FLAC.

1. Εγκαταστήστε τη βιβλιοθήκη συστήματος libopus

```
sudo apt-get update
sudo apt-get install -y libopus0 libopus-dev
```

2. Εγκαταστήστε τον αποκωδικοποιητή Opus για το frontend

```
cd PhantomSDR-Plus/frontend

# Install npm dependencies if not already done, Install Opus WASM decoder, Fix any vulnerabilities,
# Rebuild frontend
npm install
npm install @wasm-audio-decoders/opus-ml
npm audit fix
npm run build

# Return to project root
cd ..
```

3. Επαληθεύστε την εγκατάσταση του Opus

```
# Check if Opus system library is installed
pkg-config --modversion opus

# Check if Opus npm package is installed
cd frontend
npm list @wasm-audio-decoders/opus-ml
cd ..
```

Autorun Spot Reporter (FT8/FT4/WSPR)

Το PhantomSDR-Plus περιλαμβάνει έναν προαιρετικό **autorun spot reporter** (στον κατάλογο `autorun/`). Αποκωδικοποιεί FT8/FT4/WSPR από την πλευρά του διακομιστή, απευθείας από τον δέκτη, και ανεβάζει τα spot στα δίκτυα αναφορών:

- **FT8 / FT4 → PSK Reporter** (IPFIX/UDP)
- **WSPR → wsprnet.org** (HTTP)

Ελέγχεται εξ ολοκλήρου από την καρτέλα **πίνακας διαχείρισης → «Spot Reporting»** και **οι αναφορές είναι ΑΠΕΝΕΡΓΟΠΟΙΗΜΕΝΕΣ εξ ορισμού** — τίποτα δεν μεταδίδεται ή ανεβαίνει μέχρι να το ενεργοποιήσετε εκεί. Ο αποκωδικοποιητής εκτελείται ως ξεχωριστός δαίμονας Node.js και λαμβάνει τον ήχο του δέκτη τοπικά, οπότε δεν προσθέτει επιπλέον υλικό RF.

Προϋποθέσεις

Προϋπόθεση	Σημειώσεις
Node.js 22+	Το ίδιο περιβάλλον εκτέλεσης που ήδη χρειάζεται η μεταγλώττιση του frontend — εγκαθίσταται στο βήμα του Node.js .
Πακέτα <code>npm ws + cbor-x</code>	Επιλύονται μέσω του <code>autorun/node_modules</code> , ενός συμβολικού συνδέσμου προς το <code>node_modules</code> του frontend (και τα δύο πακέτα δηλώνονται στο <code>frontend/package.json</code>).
<code>util-linux</code> (taskset)	Το κουμπί «Start» της διαχείρισης καρφίτσων τον δαίμονα στους E-cores με <code>taskset</code> . Υπάρχει σχεδόν σε κάθε διανομή· τα προγράμματα εγκατάστασης το προσθέτουν ρητά.
Διακριτικό + locator	Διαβάζονται από το <code>frontend/site_information.json</code> (<code>siteSysop</code> / <code>siteGridSquare</code>), εκτός αν αντικατασταθούν στην καρτέλα διαχείρισης. Τα spot ανεβαίνουν με αυτό το διακριτικό.

⚠ **Αναφέρετε μόνο ό,τι πράγματι λαμβάνετε.** Τα spot ανεβαίνουν σε δημόσια δίκτυα με το διακριτικό σας — ενεργοποιήστε μόνο μπάντες/τρόπους που ο δέκτης σας ακούει πραγματικά και χρησιμοποιήστε το σωστό locator.

Αυτόματη εγκατάσταση

Το `install.sh` (και οι παραλλαγές ανά διανομή: `install_arch.sh`, `install_fedora.sh`, `install_opensuse.sh`) κάνουν τα πάντα για εσάς: εγκαθιστούν Node.js 22 και `util-linux`, εκτελούν το `npm install` του frontend και έπειτα δημιουργούν αυτόματα τον συμβολικό σύνδεσμο `autorun/node_modules`. Δεν απαιτούνται επιπλέον βήματα — η δυνατότητα είναι έτοιμη μόλις ολοκληρωθεί το `install.sh`.

Χειροκίνητη εγκατάσταση

ΣΗΜΕΙΩΣΗ

Το πρόγραμμα εγκατάστασης το κάνει ήδη αυτό για εσάς: δημιουργεί το symlink και εγκαθιστά το `util-linux`. Χρησιμοποιήστε τα παρακάτω βήματα μόνο για επισκευή με το χέρι.

Αν κάνατε χειροκίνητη εγκατάσταση (επιλογή B), δημιουργήστε μόνοι σας τον συμβολικό σύνδεσμο μετά το `npm install` του frontend, ώστε ο δαίμονας να μπορεί να επιλύσει τις εξαρτήσεις του:

```
cd ~/PhantomSDR-Plus

# 1. Frontend deps must be installed first (ws + cbor-x live there)
cd frontend && npm install && cd ..

# 2. Link the autorun daemon to the frontend's node_modules
ln -sf ../frontend/node_modules autorun/node_modules

# 3. Ensure taskset is available (Debian/Ubuntu shown; use your package manager)
sudo apt install -y util-linux
```

Σημείωση: το `autorun/node_modules` αγνοείται από το git, οπότε ένα καθαρό `git clone` δεν το περιέχει ποτέ — ο σύνδεσμος πρέπει να (ξανα)δημιουργείται μετά από κάθε καθαρή λήψη. Τα προγράμματα εγκατάστασης το κάνουν για εσάς· η παραπάνω εντολή αφορά μόνο τις χειροκίνητες εγκαταστάσεις.

Επαλήθευση

```
# The symlink should point at ../frontend/node_modules
ls -l autorun/node_modules

# ws + cbor-x must resolve
ls -d frontend/node_modules/ws frontend/node_modules/cbor-x

# taskset must be on PATH
command -v taskset
```

Έπειτα ανοίξτε τον πίνακα διαχείρισης, πηγαίνετε στην καρτέλα **Spot Reporting**, ορίστε τα στοιχεία σας, επιλέξτε τις μπάντες/τρόπους προς αποκωδικοποίηση, ενεργοποιήστε τους προορισμούς και πατήστε **Start**. Ένα σήμα «REPORTING» εμφανίζεται στον κύριο καταρράκτη όσο οι αναφορές είναι ενεργές. (Το PSK Reporter ανεβάζει ομαδικά κάθε 5 λεπτά και το wsprnet κάθε 2 λεπτά, οπότε ένας δαίμονας που μόλις ξεκίνησε δείχνει «0 sent» για τα πρώτα λεπτά — είναι φυσιολογικό.)

Στη συνέχεια εμφανίζονται δύο διαφορετικοί μετρητές, που εύκολα μπερδεύονται μεταξύ τους. Τα πλακίδια **SPOTS UPLOADED PER DECODER** μετρούν μόνο την τρέχουσα εκτέλεση, οπότε το Stop/Start τα μηδενίζει· ο αριθμός δίπλα σε κάθε πλαίσιο ελέγχου μπάντας/τρόπου είναι το **συνολικό** άθροισμα από την αρχή, αποθηκευμένο στο `autorun-totals.json` ώστε να επιβιώνει των επανεκκινήσεων. Δείτε τον [οδηγό του πίνακα διαχείρισης](#) και για τα δύο, καθώς και για τη σελίδα **Γραφήματα**, που σχεδιάζει συχνότητα CPU, φόρτο, θερμοκρασία και συνδεδεμένους χρήστες από 15 λεπτά έως 24 ώρες.

Αν το μηχάνημά σας ζεσταίνεται, ο πίνακας διαθέτει και [θερμική προστασία](#) που σταματά τον διακομιστή όταν η CPU φτάσει σε επικίνδυνη θερμοκρασία και τον επανεκκινεί μόλις κρυώσει. Τα όριά της προκύπτουν από το κρίσιμο όριο της ίδιας σας της CPU, οπότε δεν έχετε τίποτα να υπολογίσετε, και λειτουργεί με όποιον τρόπο εκκίνησης/διακοπής κι αν χρησιμοποιείτε. Εγκαθίσταται μαζί με τον πίνακα αλλά ξεκινά σε λειτουργία μόνο καταγραφής: καταγράφει τι θα έκανε και δεν αλλάζει τίποτα μέχρι να την ενεργοποιήσετε στις Ρυθμίσεις. Η συνεχής αποκωδικοποίηση κρατά τη CPU απασχολημένη όλο το εικοσιτετράωρο, οπότε αξίζει να διαβάσετε την καρτέλα CRASH μετά από μια εβδομάδα autorun για να δείτε πόσο κοντά φτάνει πραγματικά το μηχάνημά σας.

Η θύρα του διακομιστή εντοπίζεται αυτόματα. Ο δαίμονας συνδέεται απευθείας στη `[server]` port του ίδιου του spectrumserver (loopback + διακριτικό, παρακάμπτοντας τον διαμεσολαβητή). Διαβάζει τη θύρα αυτή από το αρχείο ρυθμίσεων με το οποίο ξεκίνησε ο τρέχων spectrumserver, οπότε λειτουργεί σε οποιαδήποτε θύρα χωρίς ρύθμιση — η γραμμή `[autorun] tap backend: ...` στο `autorun.log` δείχνει τι εντόπισε. Αν ο διακομιστής σας δεν εκτελούνταν όταν ξεκίνησε ο δαίμονας ή αν έχετε ασυνήθιστη διάταξη, καθορίστε την στο `autorun.json`: `json "server": { "host": "127.0.0.1", "port": 9002 }` Λανθασμένη θύρα εμφανίζεται ως 504 / `tap closed 1006` στο `autorun.log` με τα `decodes` κολλημένα στο 0.

Εξομίωση πελατών KiwiSDR (προαιρετικό)

Από την έκδοση v4.1.0 το PhantomSDR-Plus μπορεί να απαντά και στο **πρωτόκολλο KiwiSDR**, ώστε λογισμικό γραμμένο για KiwiSDR — το **AetherSDR**, το `kiwiclient` και τα υπόλοιπα — να συνδέεται απευθείας στον δέκτη σας, στον ίδιο υπολογιστή και θύρα που ήδη δημοσιεύετε. Είναι απενεργοποιημένη μέχρι να προστεθεί `[kiwi_emulation] enabled = true` στη ρύθμιση με την οποία τρέχει ο δέκτης σας.

Το πρόγραμμα εγκατάστασης την προσφέρει ως βήμα 17· το `./kiwi_install.sh` την εφαρμόζει σε δέντρο που είναι ήδη εγκατεστημένο.

Πλήρης τεκμηρίωση: Εξομοίωση πελατών KiwiSDR — τι κάνει η γέφυρα, πώς εγκαθίσταται, κάθε κλειδί [kiwi_emulation], η σύνδεση πελάτη, η στάθμη ήχου, το S-meter, ο ρυθμός του καταρράκτη και πίνακας συμπτωμάτων.

Ρυθμίσεις

1. Επιλέξτε το αρχείο ρυθμίσεων

Επιλέξτε το κατάλληλο αρχείο ρυθμίσεων για το SDR σας:

- `config-rtl.toml` - κλειδιά RTL-SDR
- `config-rspla.toml` - SDRplay RSP1A
- `config-airspyhf.toml` - Airspy HF+ Discovery
- `config-fobos-hf.toml` - RigExpert Fobos SDR, HF1/HF2 με άμεση δειγματοληψία (0-25 MHz)
- `config-fobos.toml` - RigExpert Fobos SDR, κλάδος RF (25-6000 MHz)
- `config-rx888mk2.toml` - RX888 MK2
- `config-hackrf.toml` - HackRF One
- `config.example.hackrf.toml` - HackRF One

Σε αυτό το παράδειγμα θα χρησιμοποιήσουμε RTL-SDR.

2. Επεξεργαστείτε το αρχείο ρυθμίσεων

```
nano config-rtl.toml
```

Βασικές ρυθμίσεις

```
[server]
port = 9002                # Web interface port (or everything else)
html_root = "frontend/dist/" # Frontend location
otherusers = 1             # Show other users (1=yes, 0=no)
threads = 2                # Number of server threads

[websdr]
register_online = true      # Register on sdr-list.xyz (true/false)
name = "Your WebSDR Name"  # Display name
antenna = "Your Antenna Type" # e.g., "Vertical", "Loop", "Dipole"
grid_locator = "AB12cd"    # Your Maidenhead grid square
hostname = "your.domain.com" # Your domain or IP address

[input]
sps = 2048000              # Sample rate (adjust for your coverage)
fft_size = 131072          # FFT size (higher = better resolution)
brightness_offset = -10    # Waterfall brightness adjustment
frequency = 145000000       # Base frequency in Hz (145 MHz for 2m)
signal = "iq"              # "iq" for complex, "real" for real sampling
fft_threads = 2            # FFT processing threads
accelerator = "opencl"     # "none", "cuda", or "opencl"
audio_sps = 12000          # Audio sample rate (keep at 12000)
audio_compression = "opus" # "flac" or "opus"
smeter_offset = -2         # S-meter calibration
waterfall_size = 1024      # Waterfall FFT size
waterfall_compression = "zstd" # Waterfall compression

[input.driver]
name = "stdin"             # Input driver
format = "u8"              # Sample format for RTL-SDR

[input.defaults]
frequency = 145500000       # Default tuning frequency
modulation = "FM"          # Default modulation mode
```

Οδηγίες για τον ρυθμό δειγματοληψίας

Κάλυψη	Ρυθμός δειγματοληψίας	Μέγεθος FFT
2 MHz	2048000	131072
3.2 MHz	3200000	131072
10 MHz	10000000	1048576
30 MHz	30000000	2097152
60 MHz	60000000	4194304

3. Ρυθμίστε τις πληροφορίες του σταθμού

```
nano frontend/site_information.json
```

Επεξεργαστείτε τα ακόλουθα πεδία:

```
{
  "siteSysop": "YourCallsign",
  "siteSysopEmailAddress": "your@email.com",
  "siteGridSquare": "AB12cd",
  "siteCity": "Your City, Country",
  "siteInformation": "https://github.com/sv1btl/PhantomSDR-Plus",
  "siteHardware": "Computer specifications",
  "siteSoftware": "PhantomSDR-Plus v5.0.0",
  "siteReceiver": "Your SDR model",
  "siteAntenna": "Antenna description",
  "siteNote": "Additional information",
  "siteIP": "http://your.domain.com:9002",
  "siteSDRBaseFrequency": 0,
  "siteSDRBandwidth": 2048000,
  "siteRegion": 1,
  "siteChatEnabled": true
}
```

Περιοχές IARU:

- **1:** Ευρώπη, Αφρική, Μέση Ανατολή, Βόρεια Ασία
- **2:** Αμερική (Βόρεια, Κεντρική, Νότια), Καραϊβική
- **3:** Ασία-Ειρηνικός, Ωκεανία

4. Προσαρμόστε τους δείκτες συχνότητων (προαιρετικό)

```
nano markers.json
```

Προσθέστε τις αγαπημένες σας συχνότητες, αναμεταδότες και ραδιοφωνικούς σταθμούς.

5. Επεξεργαστείτε το σενάριο εκκίνησης

```
nano start-rtl.sh
```

Το σενάριο εκκίνησης είναι αυτόνομος εκκινητής με ενσωματωμένο watchdog. Επεξεργαστείτε μόνο το τμήμα **RECEIVER CONFIGURATION** κοντά στην αρχή, ώστε τα ορίσματα του δέκτη να ταιριάζουν με την εγκατάστασή σας:

```
# == RECEIVER CONFIGURATION (the only receiver-specific part) ==
RX_LABEL="RTL-SDR"
RX_COMM="rtl_sdr"                # process name to monitor/kill
CONFIG="$PHANTOMDIR/config-rtl.toml" # your .toml
FIFO="$PHANTOMDIR/rtl.fifo"
RX_ARGS="{RX_ARGS:-f 145000000 -s 2048000 -}" # receiver args (edit these)
```

Παράμετροι μέσα στο RX_ARGS:

- **-f 145000000:** κεντρική συχνότητα (145 MHz)
- **-s 2048000:** ρυθμός δειγματοληψίας (2,048 MSPS)

Το CONFIG δείχνει στο δικό σας .toml. **Δεν** χρειάζεται να αλλάξετε τίποτε άλλο στο σενάριο — η λογική εκκίνησης/επανεκκίνησης/watchdog/καταγραφής είναι γενική. Μπορείτε επίσης να αντικαταστήσετε τα ορίσματα του δέκτη κατά την εκκίνηση χωρίς να επεξεργαστείτε το αρχείο: `RX_ARGS="-f 145000000 -s 2048000 -" ./start-rtl.sh` (RX888_ARGS για το `start-rx888mk2.sh`).

Ρύθμιση ανά συσκευή SDR

RTL-SDR

Εγκαταστήστε τα εργαλεία RTL-SDR

```
sudo apt install -y rtl-sdr
```

Δοκιμάστε το RTL-SDR

```
rtl_test
```

Πατήστε Ctrl+C για διακοπή. Θα πρέπει να δείτε πληροφορίες για τον ρυθμό δειγματοληψίας.

Επεξεργαστείτε τις ρυθμίσεις

```
nano config-rtl.toml
```

Συνήθεις ρυθμίσεις για RTL-SDR:

```
sps = 2048000
frequency = 145000000
signal = "iq"
format = "u8"
```

Δέκτες μέσω SoapySDR (RSP1A, Fobos, Airspy HF+)

Ο SDRplay RSP1A, ο RigExpert Fobos SDR και ο Airspy HF+ φτάνουν στον server με τον ίδιο τρόπο: ένας driver SoapySDR για τη συσκευή, και το `rx_sdr` (από το `rx_tools`) που στέλνει τα δείγματα στο `spectrumserver`. Ένα script ανά δέκτη εγκαθιστά όλη την αλυσίδα:

Δέκτης	Script	Επιλογή εγκατάστασης	Script εκκίνησης	Ρυθμίσεις
SDRplay RSP1A	<code>./setup-rspla.sh</code>	3	<code>start-rspla.sh</code>	<code>config-rspla.toml</code>
RigExpert Fobos SDR	<code>./setup-fobos.sh</code>	5	<code>start-fobos-hf.sh</code> (HF) / <code>start-fobos.sh</code> (RF)	<code>config-fobos-hf.toml</code> / <code>config-fobos.toml</code>
Airspy HF+	<code>./setup-airspyhf.sh</code>	6	<code>start-airspyhf.sh</code>	<code>config-airspyhf.toml</code>

Κάθε script:

- εγκαθιστά `git`, `cmake`, `compiler`, `libusb` και **SoapySDR** από τα πακέτα της διανομής σας — αναγνωρίζει `apt`, `dnf`, `pacman` και `zypper`, άρα δουλεύει το ίδιο σε Debian/Ubuntu, Fedora, Arch και openSUSE.
- χτίζει τον driver του δέκτη από τον πηγαίο κώδικα στο `sdr_drivers/` (εκτός `git`).
- χτίζει το **rx_tools** για το `rx_sdr`, εκτός αν το `rx_sdr` είναι ήδη εγκατεστημένο — ένα αντίγραφο εξυπηρετεί όλους τους δέκτες.
- εγκαθιστά έναν **κανόνα udev** και σας προσθέτει στην ομάδα `plugdev`, ώστε το script εκκίνησης να μη χρειάζεται ποτέ `sudo` για τη συσκευή. **Αποσυνδεθείτε και συνδεθείτε ξανά μία φορά** για να ισχύσει η ομάδα.
- στο τέλος ελέγχει ότι το SoapySDR πράγματι εμφανίζει τον driver, αλλιώς σταματά με σφάλμα.

Τρέξτε τα ως κανονικός χρήστης — καλούν μόνα τους το `sudo`. Είναι idempotent: αν ξανατρέξει ένα, ενημερώνει τον κώδικα του driver με `git pull` και ξαναχτίζει. Το πρόγραμμα εγκατάστασης καλεί ακριβώς αυτά τα scripts· σε σταθμό που είναι ήδη εγκατεστημένος, τρέξτε σκέτο αυτό του δέκτη σας. Το κοινό μέρος βρίσκεται στο `setup-sdr-common.sh`, που δεν τρέχει μόνο του. Δύο προαιρετικές μεταβλητές ισχύουν και για τα τρία: το `SDR_USER=<όνομα>` βάζει άλλον λογαριασμό στο `plugdev`, και το `SDR_SKIP_DEPS=1` παραλείπει την εγκατάσταση πακέτων.

Με τη συσκευή συνδεδεμένη, ελέγξτε ότι ο δέκτης εντοπίζεται:

```
SoapySDRUtil --find="driver=soapyMiri"    # RSP1A
SoapySDRUtil --find="driver=fobos"       # Fobos
SoapySDRUtil --find="driver=airspyhf"    # Airspy HF+
```

SDRplay RSP1A

Το `setup-rsp1a.sh` εγκαθιστά τον **driver ανοιχτού κώδικα**: το `libmirisdr-5`, τον driver που προέκυψε από reverse engineering για τα chips Mirics MSi2500/MSi001 μέσα στους RSP1 και RSP1A, με το module `SoapySDR` του, το `SoapyMiri (driver=soapyMiri)`. Δεν χρειάζεται υπηρεσία στο παρασκήνιο ούτε root. Το script βάζει επίσης σε blacklist τους drivers πυρήνα `msi2500` και `msi001` στο `/etc/modprobe.d/blacklist-msi2500.conf`, επειδή δεσμεύουν τον RSP1A ως ραδιόφωνο V4L2 πριν προλάβει το `rx_sdr` να τον ανοίξει. Αν η συσκευή ήταν συνδεδεμένη ενώ ήταν φορτωμένοι, αποσυνδέστε και ξανασυνδέστε την μία φορά (ή κάντε επανεκκίνηση).

Το κλειστό API της SDRplay (`driver=sdrplay`, με το `SoapySDRPlay` και την υπηρεσία `sdrplay`) εξακολουθεί να δουλεύει — το συνοδευτικό αρχείο `instructions-for-rsp1a` περιγράφει εκείνη την εγκατάσταση. Το `start-rsp1a.sh` διαλέγει μόνο του ανάμεσα στα δύο: αν το `SoapySDR` έχει τον driver `sdrplay` χρησιμοποιεί το API και πρώτα επανεκκινεί την υπηρεσία `sdrplay`, αλλιώς χρησιμοποιεί το `libmirisdr-5`. Ένας σταθμός στημένος στο API συνεχίζει λοιπόν να δουλεύει όπως πριν. Για να επιβάλετε έναν, ξεκινήστε με `RX_DRIVER=miri` ή `RX_DRIVER=sdrplay`. Ο driver που επιλέχθηκε γράφεται στο `logwebsdr.txt` ως RSP1A driver: `miri` ή RSP1A driver: `sdrplay`.

RigExpert Fobos SDR

Ο Fobos στέλνει **έναν κλάδο κάθε φορά**, γι' αυτό έχει δύο scripts εκκίνησης που μοιράζονται το κλείδωμα του watchdog όπως κάθε `start-*.sh` — η εκκίνηση του ενός σταματά το άλλο:

- **start-fobos-hf.sh** — οι είσοδοι HF1/HF2 με άμεση δειγματοληψία, 0-25 MHz, στη θύρα 9003. Σε αυτή τη λειτουργία δεν υπάρχει τοπικός ταλαντωτής: ο ADC ψηφιοποιεί συνεχώς 0-25 MHz (το `-f 0` είναι σκόπιμο, και η προειδοποίηση *Failed to set center freq* του `rx_sdr` είναι αναμενόμενη). Το `rx_sdr` δίνει CF32 στα 50 Msps· ο μικρός μετατροπέας `cf32_to_real` (τον χτίζει το `setup-fobos.sh` από το `cf32_to_real.c`) κρατά το κανάλι I και το κλιμακώνει σε s16, γι' αυτό το `config-fobos-hf.toml` έχει `signal="real"` και `format="s16"`.
- **start-fobos.sh** — ο κλάδος RF, 25-6000 MHz, IQ, στη θύρα 9002. Διαλέξτε το παράθυρο πριν την πρώτη εκκίνηση: τα `-f` (κέντρο) και `-s` (ρυθμός δειγματοληψίας) στη γραμμή `RX_ARGS` πρέπει να ταιριάζουν με τα `frequency=` και `sps=` του `config-fobos.toml`. Οι τιμές που έρχονται (97 MHz, 20 Msps) είναι μόνο αφετηρία για να δείτε ότι τρέχει.

ΣΗΜΑΝΤΙΚΟ

Το `config-fobos-hf.toml` έρχεται με `accelerator="opencl"`: μια ροή 50 Msps μέσα από FFT 1M bins είναι τόσο βαριά όσο τα 60 Msps του RX888, και στη CPU χάνονται δείγματα και το μηχάνημα ζεσταίνεται. Σε μηχάνημα χωρίς συσκευή OpenCL το `spectrumserver` δεν ξεκινά με αυτή τη ρύθμιση — βάλτε εκεί `accelerator="none"`.

Το `SoapyFobosSDR` χρειάζεται για να χτιστεί τα headers και του `libfobos` και του `libfobos-sdr-agile`, γι' αυτό το `setup-fobos.sh` χτίζει και τα δύο, ακόμη και για το στάνταρ firmware. Η υποστήριξη του Fobos

προέρχεται από ένα script που έγραψε άλλος σταθμός αφού έστησε έναν Fobos στο PhantomSDR-Plus· οι τιμές HF στο `config-fobos-hf.toml` είναι αυτές που δούλεψαν εκεί.

Airspy HF+

Το `setup-airspyhf.sh` εγκαθιστά το `libairspyhf` από τη διανομή όπου υπάρχει πακέτο (Debian και Ubuntu, Fedora, openSUSE) και το χτίζει από τον πηγαίο κώδικα όπου δεν υπάρχει (το Arch το έχει μόνο στο AUR), και μετά χτίζει το `SoapyAirspyHF` (`driver=airspyhf`). Το `start-airspyhf.sh` δίνει 912 ksps IQ σε CS16, που το `config-airspyhf.toml` διαβάζει ως `format="s16"`. Το συνοδευτικό αρχείο `instructions-for-airspy` περιγράφει τον χειροκίνητο δρόμο.

RX888 MK2

Εγκαταστήστε τα εργαλεία RX888

```
# Install rx_tools
git clone https://github.com/rxseger/rx_tools.git
cd rx_tools
mkdir build && cd build
cmake ..
make -j4
sudo make install
sudo ldconfig

# Install RX888 firmware and support
# Follow manufacturer instructions
```

Εκτέλεση του rx888_stream χωρίς sudo (κανόνες udev)

Εξ ορισμού η συσκευή USB Cypress FX3 του RX-888 είναι προσβάσιμη μόνο από τον root, οπότε το `rx888_stream` θα χρειαζόταν `sudo`. Εκτελέστε μία φορά το συνοδευτικό βοηθητικό σενάριο για να εγκατασταθούν κανόνες udev και να προστεθείτε στην ομάδα `plugdev`:

```
./setup-rx888-udev.sh          # re-runs itself with sudo automatically
```

Γράφει το `/etc/udev/rules.d/99-rx888.rules` και για τα τρία αναγνωριστικά προϊόντος Cypress FX3 (04b4:00f1/00f3 bootloader και 04b4:8613 με φορτωμένο firmware), επαναφορτώνει το udev και δημιουργεί σταθερό σύνδεσμο `/dev/rx888`. **Αποσυνδεθείτε και συνδεθείτε ξανά** (για την αλλαγή ομάδας) και **επανασυνδέστε τη συσκευή** μία φορά· έπειτα το `rx888_stream` — και ο εκκινητής `start-rx888mk2.sh` — τρέχουν χωρίς `sudo`. Είναι ταυτοδύναμο, οπότε μπορείτε να το εκτελέσετε ξανά με ασφάλεια. Για άλλο λογαριασμό: `RX888_USER=<όνομα> ./setup-rx888-udev.sh`.

Επεξεργαστείτε τις ρυθμίσεις

```
nano config-rx888mk2.toml
```

Συνήθεις ρυθμίσεις για RX888:

```
sps = 60000000
frequency = 0
signal = "real"
format = "s16"
fft_size = 4194304
```

HackRF One

Για τον HackRF δεν χτίζεται τίποτα: το `hackrf_transfer`, από το πακέτο `hackrf` της διανομής σας, στέλνει τα δείγματα κατευθείαν στο `spectrumserver`. Το `./setup-hackrf.sh` (επιλογή 7 του

προγράμματος εγκατάστασης) εγκαθιστά αυτό το πακέτο με apt, dnf, pacman ή zypper και προσθέτει κανόνα udev και την ομάδα plugdev, ώστε το start-hackrf.sh να τρέχει χωρίς sudo — αποσυνδεθείτε και συνδεθείτε ξανά μία φορά μετά.

```
./setup-hackrf.sh # installs the hackrf package and the udev rule
hackrf_info      # with the HackRF plugged in: must list the board
```

Το start-hackrf.sh στέλνει ένα παράθυρο κάθε φορά. Διαλέξτε το πριν την πρώτη εκκίνηση: τα -f (κέντρο) και -s (ρυθμός δειγματοληψίας) στη γραμμή RX_ARGS πρέπει να ταιριάζουν με τα frequency= και sps= του config-hackrf.toml. Τα -l και -g είναι τα κέρδη του LNA (0-40 dB, βήματα 8 dB) και του VGA (0-62 dB, βήματα 2 dB), και το -a 1 ανάβει τον ενισχυτή RF. Οι τιμές που έρχονται (98 MHz, 20 Msps) είναι μόνο αφετηρία για να δείτε ότι τρέχει.

Το hackrf_transfer γράφει IQ 8 bit **με πρόσημο**, γι' αυτό οι ρυθμίσεις θέλουν format="s8". Το config.example.hackrf.toml είχε u8 μέχρι τον Σεπτέμβριο του 2026· με u8 κάθε δείγμα είναι μετατοπισμένο κατά 128 και ο καταρράκτης δείχνει μόνο θόρυβο — ελέγξτε τυχόν παλαιότερο αντίγραφό του.

Πολλοί δέκτες σε έναν υπολογιστή

Ένας υπολογιστής μπορεί να τρέχει ταυτόχρονα περισσότερους από έναν από τους παραπάνω δέκτες — για παράδειγμα ένα RX-888 για τα βραχεία και ένα RTL-SDR για τα 2 m — τον καθένα ως ξεχωριστό διακομιστή, με επιλογή δέκτη πάνω στη σελίδα. Κάθε επιπλέον δέκτης ξεκινά με INSTANCE=<όνομα> ./start-<δέκτης>.sh και ρυθμίζεται στο instances/<όνομα>/ το receivers.toml τους καταγράφει για τον proxy και για το start-all.sh. Όλα βρίσκονται στο [Πολλοί δέκτες](#). Ο γρηγορότερος τρόπος να προσθέσετε δέκτη είναι το ./add-receiver.sh, που προσφέρουν τα σενάρια εγκατάστασης στο τέλος: εγκαθιστά τον οδηγό (setup-rtlsdr.sh για RTL-SDR, το αντίστοιχο setup-*.sh για τους άλλους) και γράφει για εσάς τον φάκελο του instance και το receivers.toml.

Δοκιμή και επαλήθευση

1. Δοκιμαστική εκτέλεση

```
# For RTL-SDR
./start-rtl.sh
```

Αυτό εκκινεί τον διακομιστή **στο παρασκήνιο** (αποσυνδέεται και επιστρέφει αμέσως) και αρχίζει να καταγράφει στο logwebsdr.txt.

2. Ελέγξτε για σφάλματα

Το σενάριο καταγράφει την πρόοδό του στο logwebsdr.txt — παρακολουθήστε το ζωντανά:

```
tail -f logwebsdr.txt
```

Αναζητήστε:

- Starting the initialization script of the PhantomSDR Server
- <receiver> running (PID ...)
- Server started normally
- Server ... FAILED ... ή ERROR: receiver binary '...' not found (διορθώστε τα ορίσματα του δέκτη ή το .toml, ή εγκαταστήστε το εργαλείο του δέκτη, και ξαναεκτελέστε το σενάριο εκκίνησης)

3. Πρόσβαση στη διεπαφή web

Ανοίξτε τον περιηγητή σας στο:

```
http://localhost:9002
```

(Αντικαταστήστε τον αριθμό θύρας με τη δική σας ρυθμισμένη θύρα)

4. Επαληθεύστε τη λειτουργία

- Ο καταρράκτης πρέπει να εμφανίζεται
- Ο ήχος πρέπει να παίζει όταν κάνετε κλικ σε σήματα
- Το S-meter πρέπει να αντιδρά στα σήματα
- Ο μετρητής χρηστών πρέπει να δείχνει «1»

5. Δοκιμάστε από άλλη συσκευή

Από άλλον υπολογιστή στο δίκτυό σας:

```
http://YOUR_SERVER_IP:9002
```

6. Ελέγξτε τη χρήση πόρων

```
# Monitor CPU usage
htop

# Monitor GPU usage (if OpenCL/CUDA enabled)
nvidia-smi # For NVIDIA
radeontop  # For AMD

# Monitor network usage
iftop
```

7. Σταματήστε τον διακομιστή

Ο διακομιστής εκτελείται στο παρασκήνιο υπό watchdog, οπότε **το Ctrl+C δεν θα τον σταματήσει** (και ο watchdog θα τον επανεκκινούσε). Χρησιμοποιήστε το κοινό σενάριο τερματισμού, που λειτουργεί για κάθε δέκτη — σταματά πρώτα τον watchdog και έπειτα τον δέκτη και τον `spectrumserver`:

```
./stop-websdr.sh
```

Ρύθμιση αυτόματης εκκίνησης

Με systemd (συνιστάται)

1. Δημιουργήστε το αρχείο υπηρεσίας

```
sudo nano /etc/systemd/system/phantomsdr.service
```

Προσθέστε το ακόλουθο περιεχόμενο (προσαρμόστε διαδρομές και χρήστη):

```
[Unit]
Description=PhantomSDR-Plus WebSDR Server
After=network.target

[Service]
Type=simple
User=youruser
WorkingDirectory=/home/youruser/PhantomSDR-Plus
# Run the watchdog in the FOREGROUND (note the --watchdog flag) so systemd can
# track it. Do NOT use the plain "./start-rtl.sh" here – that form detaches into
# the background and exits, which systemd would treat as the service stopping.
ExecStart=/home/youruser/PhantomSDR-Plus/start-rtl.sh --watchdog
ExecStop=/home/youruser/PhantomSDR-Plus/stop-websdr.sh
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target
```

Η διεργασία `--watchdog` ήδη επανεκκινεί μόνη της τον δέκτη και τον `spectrumserver`. Το `Restart=always` είναι απλώς δίχτυ ασφαλείας για τη σπάνια περίπτωση που τερματιστεί ο ίδιος ο `watchdog`. Επειδή εδώ ο διακομιστής εποπτεύεται από το `systemd`, μπορείτε να χρησιμοποιείτε `systemctl start/stop/restart` και `journalctl -u phantomsdr -f` αντί να εκτελείτε τα σενάρια με το χέρι.

2. Ενεργοποιήστε και εκκινήστε την υπηρεσία

```
# Reload systemd
sudo systemctl daemon-reload

# Enable service to start on boot
sudo systemctl enable phantomsdr.service

# Start service now
sudo systemctl start phantomsdr.service

# Check status
sudo systemctl status phantomsdr.service
```

3. Διαχειριστείτε την υπηρεσία

```
# Start
sudo systemctl start phantomsdr

# Stop
sudo systemctl stop phantomsdr

# Restart
sudo systemctl restart phantomsdr

# View logs
sudo journalctl -u phantomsdr -f
```

Με Screen (εναλλακτικά)

Συνήθως περιπτώ: το `./start-rtl.sh` ήδη αποσυνδέεται στο παρασκήνιο (μέσω `setsid`) και συνεχίζει να εκτελείται μετά την αποσύνδεσή σας, με δικό του `watchdog`. Το Screen είναι χρήσιμο μόνο αν θέλετε συγκεκριμένα διαδραστική συνεδρία για την εκτέλεση της μορφής προσκηνίου `./start-rtl.sh --watchdog`.

1. Εγκαταστήστε το Screen

```
sudo apt install -y screen
```

2. Εκκινήστε σε συνεδρία Screen

```
screen -S phantomsdr
./start-rtl.sh
```

Πατήστε `Ctrl+A` και έπειτα `D` για αποσύνδεση.

3. Επανασυνδεθείτε στη συνεδρία

```
screen -r phantomsdr
```

Θερμική προστασία CPU

Πλήρες εγχειρίδιο: [THERMAL_GUARD.md](#) — οι τέσσερις λειτουργίες και τι πρέπει να κάνει ο διαχειριστής σε καθεμία, λειτουργία χωρίς τον πίνακα, το στάδιο `throttle` χωρίς `root`, δοκιμές και επίλυση προβλημάτων.

Το PhantomSDR-Plus περιλαμβάνει μια προστασία που σταματά τον διακομιστή αν η CPU φτάσει σε επικίνδυνη θερμοκρασία και τον ξεκινά ξανά μόλις κρυώσει. Τα όριά της προκύπτουν από το κρίσιμο όριο που δημοσιεύει η ίδια σας η CPU, οπότε δεν έχετε τίποτα να υπολογίσετε, και λειτουργεί με όποιον τρόπο εκκίνησης/διακοπής κι αν χρησιμοποιείτε.

Αν χρησιμοποιείτε τον πίνακα διαχείρισης, την έχετε ήδη — τρέχει μέσα στον πίνακα και ρυθμίζεται από τη σελίδα «Ρυθμίσεις». Δείτε τον [οδηγό του πίνακα διαχείρισης](#). Η υπόλοιπη ενότητα αφορά εγκαταστάσεις **χωρίς** τον πίνακα.

1. Δείτε τι βλέπει η προστασία στο μηχάνημά σας

```
cd ~/PhantomSDR-Plus
python3 thermal_guard.py --once
```

```
sensor      : hwmon:coretemp
temperature: 58.0 C
crit       : 100.0 C
mode       : log
thresholds : warn 88.0 throttle 92.0 stop 95.0 resume 75.0 (C)
cpufreq    : not writable – throttle stage disabled
```

Χρησιμοποιούνται μόνο πραγματικοί αισθητήρες της CPU (`coretemp`, `k10temp`, `zenpower`, `cpu_thermal`). Οι `acpitz` και οι μη επισημασμένες θερμικές ζώνες αγνοούνται σκόπιμα, γιατί συχνά αναφέρουν θερμοκρασία κουτιού ή πλακέτας δεκάδες βαθμούς κάτω από τη CPU. Αν εμφανιστεί `sensor : NONE`, το

μηχάνημα δεν μπορεί να προστατευθεί — συνηθισμένο σε VPS ή μέσα σε container — και η προστασία μένει ανενεργή αντί να προσποιείται.

Δεν χρειάζεται να εγκαταστήσετε τίποτα: το `thermal_guard.py` χρησιμοποιεί μόνο τη standard library της Python.

2. Ρυθμίστε την

Η προστασία διαβάζει το `admin_config.json` δίπλα στο `thermal_guard.py`. Δημιουργήστε το αν δεν έχετε — χωρίς τον πίνακα δεν θα έχετε. Το ελάχιστο χρήσιμο αρχείο:

```
{
  "sdr_process_name": "spectrumserver",
  "stop_script": "stop-websdr.sh",
  "start_script": "start-rx888mk2.sh",
  "thermal_mode": "stop+restart"
}
```

- `thermal_mode` — `log` (μόνο καταγραφή, η προεπιλογή), `throttle`, `stop` ή `stop+restart`. **Αυτό είναι που πρέπει οπωσδήποτε να ορίσετε**, γιατί η προστασία έρχεται αδρανής: αν την αφήσετε ως έχει, μόνο γράφει τι θα έκανε. Μπορείτε επίσης να το δώσετε στη γραμμή εντολών ως `--mode stop+restart`, που υπερισχύει του αρχείου — αν σας βολεύουν όλες οι υπόλοιπες προεπιλογές, δεν χρειάζεστε καθόλου αρχείο ρυθμίσεων.
- `stop_script` / `start_script` — πώς σταματά και ξεκινά τον διακομιστή σας. Χωρίς σενάριο διακοπής καταφεύγει σε `SIGTERM` και μετά από 10 δευτερόλεπτα σε `SIGKILL` προς το `sdr_process_name`. Χωρίς σενάριο εκκίνησης σταματά αλλά δεν επανεκκινεί ποτέ.
- Τα όρια και οι χρόνοι μπορούν επίσης να μπουν εδώ, με τα ίδια ονόματα κλειδιών που χρησιμοποιεί ο πίνακας — ο πλήρης πίνακας βρίσκεται στον [οδηγό του πίνακα διαχείρισης](#).

Αν τον διακομιστή SDR τον επιβλέπει το systemd, βάλτε το `stop_script` να δείχνει σε ένα μικρό σενάριο που εκτελεί `systemctl stop τη-μονάδα-σας`, αντί να αφήσετε την προστασία να στείλει σήμα απευθείας στη διεργασία. Η προστασία θα κερδίσει έτσι κι αλλιώς — όσο η CPU είναι πολύ ζεστή επαναλαμβάνει τη διακοπή κάθε 2 δευτερόλεπτα, οπότε ό,τι επαναφέρει τον διακομιστή αναιρείται — αλλά μια καθαρή διακοπή είναι προτιμότερη από μια διαμάχη κάθε δύο δευτερόλεπτα.

3. Τρέξτε την ως υπηρεσία

Το αποθετήριο περιλαμβάνει έτοιμη μονάδα, το `thermal-guard.service`. Αλλάξτε μέσα το `User=` και τις δύο διαδρομές και μετά:

```
sudo cp thermal-guard.service /etc/systemd/system/
sudo systemctl daemon-reload
sudo systemctl enable --now thermal-guard
systemctl status thermal-guard
```

Το `User=` πρέπει να είναι λογαριασμός που επιτρέπεται να εκτελεί τα σενάρια εκκίνησης/διακοπής σας — κανονικά ο ίδιος χρήστης που τρέχει τον διακομιστή SDR. **Μια προστασία που τρέχει ως χρήστης που δεν μπορεί να σταματήσει τον διακομιστή σας αφήνει απροστάτευτους, ενώ δείχνει ότι σας προστατεύει.**

Για να οπλίσετε την προστασία από τη μονάδα αντί για το αρχείο ρυθμίσεων, προσθέστε το `--mode` στο `ExecStart`:

```
ExecStart=/usr/bin/python3 -u /home/sysop/PhantomSDR-Plus/thermal_guard.py --mode stop+restart
```

Χωρίς systemd:

```
cd ~/PhantomSDR-Plus
nohup python3 -u thermal_guard.py --mode stop+restart >> thermal.log 2>&1 &
```

Το `python3 thermal_guard.py --help` δείχνει όλες τις επιλογές, μαζί με το `--config` για αρχείο ρυθμίσεων αλλού και όχι δίπλα στο σενάριο.

4. Παρακολουθήστε την και μετά εμπιστευτείτε την

Ό,τι κάνει η προστασία γράφεται στο `crash.log` μέσα στον φάκελο PhantomSDR-Plus, ένα συμβάν ανά γραμμή:

```
[THERMAL] warn: 89.0C (warn=88.0 crit=100.0) sustained 31s
[THERMAL] STOPPED server: 96.0C sustained 62s (stop=95.0 crit=100.0) – Started stop-websdr.sh
[THERMAL] lockout: process reappeared at 94.0C – re-stopped [3 time(s) so far]
[THERMAL] recovered: 74.0C held below 75.0 – lockout cleared
```

```
grep THERMAL crash.log
```

Αφήστε πρώτα το `thermal_mode` στο "log" για μια εβδομάδα και διαβάστε το αρχείο μετά τις βαριές σας στιγμές — μια πλήρη μεταγλώττιση, ένα ζεστό απόγευμα. Αν δεν εμφανιστεί τίποτα, το μηχανήμα σας δεν πλησίασε ποτέ. Μετά βάλτε τη λειτουργία σε `stop`.

Πριν βασιστείτε σε αυτήν, επιβεβαιώστε όλη τη διαδρομή μία φορά με μια πλασματική ένδειξη πάνω από το όριο διακοπής σας:

```
"thermal_test_temp": 97
```

Η προστασία τη θεωρεί πραγματική, οπότε τρέχουν κατά παραγγελία τα `warn` → `stop` → κλείδωμα → επάνοδος → επανεκκίνηση. Αφαιρέστε τη γραμμή (ή βάλτε την σε `null`) για να επιστρέψετε στον πραγματικό αισθητήρα. Με τη λειτουργία στο `stop` ο διακομιστής όντως σταματά και οι ακροατές σας αποσυνδέονται, οπότε κάντε το όταν δεν υπάρχει κόσμος — ή δοκιμάστε στο `log`, όπου φαίνεται τι θα γινόταν χωρίς να αγγίζετε τίποτα.

Αντιμετώπιση προβλημάτων

Σφάλματα μεταγλώττισης

Ζητήματα εξαρτήσεων

```
# Ubuntu: Install missing dependencies
sudo apt install --fix-broken

# Fedora: Install missing dependencies
sudo dnf install <package-name>
```

Αποτυχία ρύθμισης του Meson

```
# Clean and reconfigure
rm -rf build
meson setup build --buildtype=release
```

Το meson setup τερματίζει με ModuleNotFoundError: No module named 'mesonbuild'

```
Traceback (most recent call last):
  File "/home/<user>/local/bin/meson", line 3, in <module>
    from mesonbuild.mesonmain import main
ModuleNotFoundError: No module named 'mesonbuild'
```

Δεν φταίει ο πηγαίος κώδικας. Μια παλιά εγκατάσταση με `pip install --user meson` άφησε ένα σενάριο εκκίνησης στο `~/.local/bin`, το οποίο στο `PATH` προηγείται του `/usr/bin` και έτσι κρύβει το αντίγραφο που εγκατέστησε ο διαχειριστής πακέτων. Μια αναβάθμιση διανομής (π.χ. Ubuntu 24.04 → 26.04) μεταφέρει την Python σε νέα έκδοση, το παλιό `site-packages` που είχε το `mesonbuild` δεν βρίσκεται πλέον στη διαδρομή εισαγωγής, και το σενάριο τερματίζει πριν κάνει οτιδήποτε. Το ίδιο μπορεί να συμβεί και στο `ninja`.

Ο εγκαταστάτης το εντοπίζει και το παρακάμπτει για όσο διαρκεί η εκτέλεση, εμφανίζοντας προειδοποίηση, αλλά επιδιορθώστε το σύστημα:

```
rm -f ~/.local/bin/meson
hash -r
meson --version          # πρέπει να τυπώσει έκδοση
```

Αν προτιμάτε να κρατήσετε το meson εγκατεστημένο μέσω pip, εγκαταστήστε το ξανά για την Python που έχει τώρα το σύστημα:

```
python3 -m pip install --user --force-reinstall --break-system-packages meson
```

Το meson setup σταματά με Clock skew detected

```
ERROR: Clock skew detected. File /home/pi/PhantomSDR-Plus/build/./meson.build has a time stamp
10392.2144s in the future.
```

Δεν υπάρχει κανένα πρόβλημα στον κώδικα: το ρολόι του συστήματος πάει πίσω σε σχέση με τα αρχεία. Το meson και το ninja αρνούνται να χτίσουν όταν ένα αρχείο εισόδου είναι νεότερο από την τρέχουσα ώρα, γιατί δεν μπορούν να ξέρουν τι είναι παλιό. Συμβαίνει σε Raspberry Pi χωρίς μπαταρία στην υποδοχή RTC — κάθε εκκίνηση ξεκινά από την τελευταία γνωστή ώρα, και ένα χτίσιμο πριν συγχρονιστεί το NTP βλέπει όλο το δέντρο με ημερομηνία στο μέλλον — καθώς και σε δέντρο που αποσυμπιέστηκε ή αντιγράφηκε από μηχάνημα που πάει μπροστά.

Διορθώστε πρώτα το ρολόι:

```
timedatectl                                # είναι σωστή η ώρα; είναι συγχρονισμένο το NTP;
sudo timedatectl set-ntp true
```

Περιμένετε λίγα δευτερόλεπτα και τρέξτε ξανά τον installer. Το ελέγχει πριν καλέσει το meson και προσφέρεται να επαναφέρει τις χρονοσημάνσεις (`PHANTOM_FIX_CLOCK_SKEW=y|n`). Με το χέρι, μέσα από τον φάκελο του κώδικα:

```
find . -path ./build -prune -o -path ./git -prune -o -newermt now -print0 | xargs -0r touch
```

Το backend χτίστηκε αλλά δεν υπάρχει ιστοσελίδα

Το σύμπτωμα είναι μια εγκατάσταση που μοιάζει να πέτυχε σχεδόν ολόκληρη: το `build/spectrumserver` υπάρχει, ο διακομιστής ξεκινά, και ο φυλλομετρητής δεν παίρνει τίποτα — επειδή ο κατάλογος `frontend/dist/` δεν παρήχθη ποτέ. Ψάξτε στην έξοδο του προγράμματος εγκατάστασης για:

```
[vite-plugin-top-level-await] missing field `type`
  at Compiler.printSync (node_modules/@swc/core/index.js:257:29)
error during build:
  Failed: Default
```

Το vite-plugin-top-level-await ζητά @swc/core ^1.12.14, οπότε μια καθαρή `npm install` καταλήγει στην 1.16.0, της οποίας η `printSync()` απορρίπτει το συντακτικό δέντρο που της δίνει το πρόσθετο. Δεν έχει καμία σχέση με τη διανομή σας — κάθε πρόγραμμα εγκατάστασης πλέον καρφώνει την έκδοση που δουλεύει, όπως και το `frontend/package.json`. Ένα μηχανήμα που εγκαταστάθηκε πριν προστεθεί αυτό το κάρφωμα συνεχίζει να δουλεύει μέχρι να διαγραφεί ο κατάλογος `node_modules` του — γι' αυτό εμφανίζεται από το πουθενά σε μια επανεγκατάσταση.

Αν επιδιορθώνετε παλαιότερο αντίγραφο με το χέρι:

```
cd frontend
npm pkg set 'overrides.@swc/core=1.15.33'
npm install
./build-all.sh
```

Σφάλματα εκτέλεσης

Η θύρα χρησιμοποιείται ήδη

```
# Find process using port
sudo lsof -i :9002

# Kill process
sudo kill -9 <PID>
```

Άρνηση πρόσβασης στο SDR

```
# Add user to plugdev group
sudo usermod -a -G plugdev $USER

# Reload groups (or log out and back in)
newgrp plugdev
```

Για ένα **RX-888 MkII** αυτό από μόνο του δεν αρκεί — η συσκευή Cypress FX3 χρειάζεται επίσης κανόνες `udev`. Εκτελέστε το συνοδευτικό βοηθητικό σενάριο (εγκαθιστά τους κανόνες και σας προσθέτει στην `plugdev`), έπειτα αποσυνδεθείτε/συνδεθείτε και επανασυνδέστε τη συσκευή:

```
./setup-rx888-udev.sh
```

Προβλήματα ήχου

```
# Try different codec
# Edit config file and change:
audio_compression = "flac" # or "opus"

# Rebuild
cd frontend && npm run build && cd ..
```

Προβλήματα OpenCL

Το clinfo δεν δείχνει συσκευές

```
# Verify drivers are loaded
sudo clinfo

# Check OpenCL ICD files
ls /etc/OpenCL/vendors/

# Reinstall drivers (Intel example)
sudo apt remove --purge intel-opencl-icd
sudo apt install intel-opencl-icd
```

Δεν βελτιώθηκαν οι επιδόσεις

```
# Verify OpenCL is enabled in config
accelerator = "opencl"

# Try different device
# Add to config file:
[input.opencl]
device_id = 0 # Try 0, 1, 2, etc.
```

Προβλήματα δικτύου

Δεν υπάρχει πρόσβαση από άλλες συσκευές

```
# Check firewall
sudo ufw status
sudo ufw allow 9002/tcp

# Or disable firewall temporarily for testing
sudo ufw disable
```

Υψηλή καθυστέρηση

```
# Reduce waterfall size in config
waterfall_size = 512

# Increase buffer size
buffer_size = 16384

# Use Opus instead of FLAC
audio_compression = "opus"
```

Προβλήματα συσκευής SDR

Δεν βρέθηκε το RTL-SDR

```
# Check if device is recognized
lsusb | grep Realtek

# Test with rtl_test
rtl_test

# Try blacklisting DVB-T drivers
echo "blacklist dvb_usb_rtl28xxu" | sudo tee /etc/modprobe.d/blacklist-rtl.conf
sudo rmmod dvb_usb_rtl28xxu
```

Δεν βρέθηκε το SDRplay

Το `grep "RSP1A driver" logwebsdr.txt` δείχνει ποιον driver διάλεξε το `start-rspla.sh`.

Με τον ανοιχτό driver (`libmirisdr-5`, `RSP1A driver: miri`):

```
# The kernel's msi2500 driver must not hold the device
lsmod | grep msi2500 && sudo modprobe -r msi2500 msi001

# Test with SoapySDR
SoapySDRUtil --find="driver=soapyMiri"
```

Με το API της SDRplay (`RSP1A driver: sdrplay`):

```
# Check API installation
systemctl status sdrplay

# Restart API
sudo systemctl restart sdrplay

# Test with SoapySDR
SoapySDRUtil --find="driver=sdrplay"
```

Δεν βρέθηκε ο Fobos ή ο Airspy HF+

```
SoapySDRUtil --info | grep -i factories # must list fobos / airspyhf
SoapySDRUtil --find="driver=fobos"      # or driver=airspyhf
id | grep plugdev                       # log out and back in if missing
```

Αν ο driver λείπει από τη λίστα, ξανατρέξτε το `./setup-fobos.sh` ή το `./setup-airspyhf.sh`: σταματά με την αιτία αν αποτύχει το χτίσιμο ή ο έλεγχος του SoapySDR.

Βελτιστοποίηση επιδόσεων

Βελτιστοποίηση CPU

```
# Adjust thread counts based on CPU cores
[server]
threads = 4 # Set to number of cores

[input]
fft_threads = 4 # Set to number of cores
```

Βελτιστοποίηση μνήμης

```
# Reduce memory usage
[input]
waterfall_size = 512 # Lower value = less memory
fft_size = 65536    # Lower value = less memory
```

Βελτιστοποίηση δικτύου

```
# Optimize for limited bandwidth
[input]
audio_compression = "opus" # More efficient than FLAC
waterfall_compression = "zstd" # Efficient compression
```

Ενημέρωση του PhantomSDR-Plus

Από την έκδοση 4.1.0 το αποθετήριο περιλαμβάνει το `update.sh`, ένα εργαλείο που ενημερώνει μια εγκατεστημένη στάση με το δημοσιευμένο δέντρο αρχείων **χωρίς να αγγίζει τα αρχεία που την κάνουν δική σας στάση**. Αντικαθιστά το χειρόγραφο σενάριο με `git pull` που ζητούσαν οι προηγούμενες εκδόσεις αυτού του οδηγού, και δεν χρειάζεται καθόλου `git`: το δημοσιευμένο δέντρο κατεβαίνει ως `tarball` και συγκρίνεται με το δικό σας αρχείο προς αρχείο. Λειτουργεί λοιπόν το ίδιο είτε κλωνοποιήσατε το αποθετήριο, είτε αποσυμπιέσατε ένα `update.zip`, είτε αντιγράψατε το δέντρο από ένα στικάκι USB.

Αν η εγκατάστασή σας δεν έχει ακόμη το `update.sh`

Ένα παλαιότερο δέντρο δεν περιέχει το σενάριο. Κατεβάστε το μία φορά — είναι το μόνο βήμα όλης της διαδικασίας που κάνετε ποτέ με το χέρι:

```
cd ~/PhantomSDR-Plus
curl -fLO https://raw.githubusercontent.com/sv1btl/PhantomSDR-Plus/main/update.sh
chmod +x update.sh
```

Από εκεί και πέρα τα πάντα — πηγαίος κώδικας, frontend, τεκμηρίωση, εγκαταστάτες και το ίδιο το `update.sh` — έρχονται μέσω του εργαλείου.

Βήμα 1 — δείτε τι θα αλλάξει (δεν γράφει τίποτα)

```
cd ~/PhantomSDR-Plus
./update.sh
```

Κατεβάζει το δημοσιευμένο δέντρο, το συγκρίνει με το δικό σας και τυπώνει αναφορά. Δεν γράφει απολύτως τίποτα, οπότε μπορείτε να το τρέξετε οποιαδήποτε στιγμή — ακόμη και με τον δέκτη στον αέρα. Ο κωδικός εξόδου είναι `0` όταν είστε ενημερωμένοι και `10` όταν υπάρχει ενημέρωση που περιμένει, ώστε μια εργασία `cron` να σας ειδοποιεί όταν υπάρχει κάτι να γίνει.

Βήμα 2 — εφαρμόστε την

```
./update.sh --apply
```

Τρία είδη αρχείων αντιμετωπίζονται διαφορετικά, και ακριβώς σε αυτό έγκειται η αξία του:

Αρχεία	Τι γίνεται
<code>config.toml</code> , <code>config-<δέκτης>.toml</code> , <code>markers.json</code> , <code>admin_config.json</code> , <code>autorun.json</code> , <code>frequencylist/</code> , <code>chat_history.txt</code> , <code>frontend/variant.json</code> , <code>frontend/site_information.json</code> , τα αρχεία καταγραφής, <code>build/</code> , <code>frontend/dist/</code>	Δεν αγγίζονται ποτέ , ούτε καν εμφανίζονται σε ερώτηση. Αυτά είναι που κάνουν το μηχανήμα <i>δικό σας</i> δέκτη.
<code>start-*.sh</code> , <code>stop-websdr.sh</code> , οι μονάδες <code>*.service</code> , <code>install*.sh</code> , <code>recompile.sh</code> , <code>setup_admin.sh</code> , <code>smeter_theme.sh</code> , <code>proxy.py</code> , <code>admin_server.py</code> , <code>thermal_guard.py</code> , <code>websdr_relay.py</code> , <code>frontend/src/bands-config.js</code> , τα δείγματα ρυθμίσεων <code>config.example*.toml</code>	Ρωτάει πάντα , γιατί αυτά είναι τα αρχεία που ένας διαχειριστής έχει λόγο να έχει τροποποιήσει.
Όλα τα υπόλοιπα	Ενημερώνονται, αφού πρώτα σωθεί αντίγραφο του παλιού αρχείου στο αρχείο αντιγράφων ασφαλείας (δείτε <i>Αναίρεση μιας ενημέρωσης</i>).

Για κάθε αρχείο της μεσαίας ομάδας βλέπετε τις διαφορές και έχετε τρεις επιλογές:

```
start-rx888mk2.sh [K]eep mine / [u]pstream / [b]oth (ENTER = Keep mine)
```

- **Keep mine** — το δικό σας αρχείο μένει ακριβώς όπως είναι.
- **upstream** — εγκαθίσταται η νέα έκδοση, αφού πρώτα κρατηθεί αντίγραφο της δικής σας.
- **both** — η νέα έκδοση γράφεται δίπλα στη δική σας ως `start-rx888mk2.sh.new`, για να ενσωματώσετε τις δικές σας αλλαγές με την ησυχία σας.

Πώς είναι η πρώτη εκτέλεση. Την πρώτη φορά δεν υπάρχει καταγραφή για το από ποια έκδοση προήλθαν τα αρχεία σας, οπότε κάθε αρχείο της μεσαίας ομάδας τίθεται υπόψη σας — περίπου δέκα ερωτήσεις. Απαντήστε ως εξής:

Η περίπτωση σας	Απάντηση
Δεν πειράξατε ποτέ αυτό το αρχείο	u — πάρτε τη νέα έκδοση. Η συνηθισμένη περίπτωση.
Το πειράξατε (δικά σας RX888_ARGS, καρφίτσωμα CPU, τροποποιημένη μονάδα)	b — μένει το δικό σας και η νέα έκδοση μπαίνει δίπλα ως <code>&lt;αρχείο>.new</code> .
Δεν είστε βέβαιοι	ENTER — μένει το δικό σας, δεν χάνεται τίποτα, συγκρίνετε αργότερα.

Η διαμόρφωσή σας δεν εμπλέκεται ποτέ σε αυτό: οι ερωτήσεις αφορούν μόνο σενάρια και μονάδες υπηρεσιών.

Το `update.sh` καταγράφει στο `.update-state/` την έκδοση κάθε αρχείου που εγκαθιστά. Έτσι, από τη δεύτερη εκτέλεση και μετά ξεχωρίζει ένα αρχείο που **εσείς** αλλάξατε από ένα αρχείο που είναι απλώς παλιό, και σταματά να ρωτήσει μόνο για όσα πράγματι πειράξατε.

Πριν γράψει οτιδήποτε σταματά τον δέκτη, τον πίνακα διαχείρισης, τον αντίστροφο proxy και το relay διαφορικής λήψης WebSDR **της εγκατάστασης που ενημερώνει** — ό,τι εξυπηρετεί άλλον κατάλογο αναφέρεται και αφήνεται να τρέχει, ώστε ένα δεύτερο αντίγραφο να ενημερώνεται όσο το πρώτο μένει στον αέρα — και στο τέλος ξεκινά ξανά ακριβώς ό,τι σταμάτησε. Αν άλλαξαν αρχεία πηγαίου κώδικα ή του frontend, προσφέρεται να τρέξει το `recompile.sh` για εσάς. Δεν διαγράφει ποτέ τίποτα: αρχεία που έχουν φύγει από το αποθετήριο απλώς αναφέρονται, και αφαιρούνται μόνο αν το ζητήσετε με `--prune`.

Αναίρεση μιας ενημέρωσης

```
./update.sh --restore LAST
```

Κάθε αρχείο που αντικαθιστά μια ενημέρωση σώζεται πρώτα, σε ένα αρχείο με ημερομηνία ανά εκτέλεση, στον απλό, ορατό φάκελο `update-backups/`:

```
update-backups/phantomsdr-backup-20260923-164530.zip
```

Το αρχείο περιέχει μόνο τα αρχεία που αντικατέστησε η συγκεκριμένη εκτέλεση, το καθένα στη διαδρομή που έχει στο δέντρο σας, οπότε μπορεί να αντιγραφεί σε άλλο μηχάνημα ως ένα αρχείο, να ανοιχτεί οπουδήποτε ή να αποσυμπίεστεί κατευθείαν πάνω στην εγκατάσταση. Όπου δεν είναι εγκατεστημένο το `zip`, γράφεται ως `.tar.gz`. Κρατιούνται τα τρία πιο πρόσφατα. Οποιαδήποτε από αυτές τις εντολές επαναφέρει ό,τι αντικατέστησε μια ενημέρωση:

```
./update.sh --restore LAST
./update.sh --restore phantomsdr-backup-20260923-164530.zip
```

Κάθε αρχείο περιέχει επίσης το δικό του `restore.sh`, για την περίπτωση που το ίδιο το `update.sh` είναι το πρόβλημα: αποσυμπίεστε το αρχείο και δώστε του τον φάκελο της εγκατάστασής σας, για παράδειγμα `./restore.sh /home/pi/PhantomSDR-Plus`. Κατά την επαναφορά το `update.sh` τρέχει από αντίγραφο έξω από το δέντρο, επειδή συνήθως είναι κι αυτό ένα από τα αρχεία που επαναφέρονται.

Τα αντίγραφα ασφαλείας από πριν τις 23 Σεπτεμβρίου 2026 βρίσκονται στον κρυφό φάκελο `.update-backups/<χρονολόγηση>/` επαναφέρονται με τον ίδιο τρόπο και μετρούν στα τρία που κρατιούνται.

Άλλες επιλογές

```
./update.sh --apply --yes      # δεν ρωτά ποτέ· ό,τι έχετε αλλάξει ΔΙΑΤΗΡΕΙΤΑΙ
./update.sh --ref v5.0.0      # ετικέτα, κλάδος ή commit αντί για το τρέχον δέντρο
./update.sh --list-excludes   # τυπώνει τους κανόνες "μην αγγίζεις" όπως ισχύουν εδώ
./update.sh --verbose         # εμφανίζει όλα τα αρχεία, όχι μόνο τα πρώτα 40
```

Μπορείτε να προσθέσετε δικούς σας κανόνες "μην αγγίζεις" γράφοντας ένα μοτίβο ανά γραμμή στο `update-exclude.txt`, στον ριζικό φάκελο της εγκατάστασης.

Αν η ανακατασκευή αποτύχει σε πολύ παλιά εγκατάσταση

Το `update.sh` ενημερώνει αρχεία, όχι πακέτα του συστήματος. Αν το δέντρο σας είναι τόσο παλιό ώστε η μεταγλώττιση να χρειάζεται πλέον βιβλιοθήκες που δεν έχετε, το `recompile.sh` θα σταματήσει με σφάλμα μεταγλωττιστή ή `meson`. Δεν είναι χαλασμένη ενημέρωση — λείπουν οι εξαρτήσεις:

```
./install.sh
```

Ο εγκαταστάτης ενημερώνεται από την ίδια εκτέλεση, και η διαμόρφωσή σας επιβιώνει και από αυτόν.

Ενημέρωση με το χέρι

Αν προτιμάτε να εφαρμόσετε μόνοι σας ένα πακέτο αρχείων:

```
cd ~
unzip -o update.zip
cd PhantomSDR-Plus
./stop-websdr.sh
cp -a ~/update/. .
chmod +x *.sh
./recompile.sh
./start-rx88mk2.sh      # ή το σενάριο εκκίνησης του δικού σας δέκτη
```

Κρατήστε πρώτα αντίγραφο της διαμόρφωσής σας — `config-*.toml`, `frontend/site_information.json`, `admin_config.json` και `markers.json` — γιατί ένα πακέτο αρχείων δεν μπορεί να ξεχωρίσει τις δικές σας αλλαγές από εκείνες της έκδοσης. Ακριβώς αυτό το πρόβλημα λύνει το `update.sh`.

Αντίγραφα ασφαλείας και επαναφορά

Αρχεία προς αντιγραφή

- Αρχεία ρυθμίσεων: `*.toml`
- Πληροφορίες σταθμού: `frontend/site_information.json`
- Δείκτες: `markers.json`
- Προσαρμοσμένα σενάρια: `start-*.sh`, `stop-*.sh`
- Ιστορικό συνομιλίας: `chat_history.txt`

- Εικόνα φόντου: `frontend/src/assets/background.jpg`

Εντολή αντιγραφής

```
cd ~/PhantomSDR-Plus
tar -czf phantomsdr-backup-$(date +%Y%m%d).tar.gz \
*.toml \
*.sh \
markers.json \
chat_history.txt \
frontend/site_information.json \
frontend/src/assets/background.jpg
```

Εντολή επαναφοράς

```
tar -xzf phantomsdr-backup-YYYYMMDD.tar.gz
```

Θέματα ασφάλειας

Ρύθμιση τείχους προστασίας

```
# Allow only necessary ports
sudo ufw allow 9002/tcp
sudo ufw enable
```

Αντίστροφος διαμεσολαβητής (προαιρετικά)

Εξετάστε τη χρήση nginx ή Apache ως αντίστροφου διαμεσολαβητή για:

- κρυπτογράφηση SSL/TLS
- αντιστοίχιση ονόματος τομέα
- εξισορρόπηση φορτίου
- έλεγχο πρόσβασης

Όρια χρηστών

Επεξεργαστείτε το `config.toml`:

```
[server]
max_users = 100 # Limit concurrent users
```

Λήψη βοήθειας

Πόροι

- **Τεκμηρίωση:** αυτός ο οδηγός, README.md, USER_GUIDE.md
- **GitHub Issues:** <https://github.com/sv1btl/PhantomSDR-Plus/issues>
- **Ζωντανή επίδειξη:** <http://phantomsdr.no-ip.org:8900/>

Αναφορά προβλημάτων

Όταν αναφέρετε προβλήματα, περιλάβετε:

1. Λειτουργικό σύστημα και έκδοση
2. Μοντέλο συσκευής SDR
3. Περιεχόμενο του αρχείου ρυθμίσεων
4. Μηνύματα σφάλματος
5. Χρήση πόρων συστήματος (CPU, RAM, GPU)

Υποστήριξη από την κοινότητα

- Ελέγξτε τα υπάρχοντα GitHub issues πριν δημιουργήσετε νέα
- Δώστε αναλυτικές πληροφορίες για την εγκατάστασή σας
- Συμπεριλάβετε αρχεία καταγραφής και μηνύματα σφάλματος
- Δείξτε υπομονή και σεβασμό

Παράρτημα A: πλήρης κατάλογος εξαρτήσεων

Κατάλογος πακέτων Ubuntu 24.04

```
build-essential
cmake
pkg-config
meson
libfftw3-dev
libwebsocketpp-dev
libflac++-dev
zlib1g-dev
libzstd-dev
libboost-all-dev
libopus-dev
libliquid-dev
git
util-linux (taskset – for the autorun spot reporter)
psmisc
wget
curl
rtl-sdr (for RTL-SDR)
airspy (for Airspy)
hackrf (for HackRF)
libclfft-dev (for OpenCL)
ocl-icd-opencl-dev (for OpenCL)
clinfo (for OpenCL)
```

Παράρτημα Β: παραδείγματα ρυθμίσεων

Παράδειγμα 1: RTL-SDR για VHF/UHF

```
[input]
sps = 2048000
frequency = 145000000
signal = "iq"

[input.driver]
format = "u8"

[input.defaults]
frequency = 145500000
modulation = "FM"
```

Παράδειγμα 2: RX-888 mk2 για HF (0-30 MHz)

Ο δέκτης που χρησιμοποιούν οι περισσότεροι sysop. Εδώ δίνεται ολόκληρη η ενότητα `[input]`, όχι απόσπασμα, με τις τιμές του `config-rx888mk2.toml` που συνοδεύει το αποθετήριο.

```
[input]
sps = 60000000          # 0-30 MHz με απευθείας δειγματοληψία
fft_size = 4194304      # δείτε τη σημείωση παρακάτω
fft_threads = 8
brightness_offset = -9  # πιο αρνητικό αν ο καταρράκτης δείχνει μαύρες περιοχές
frequency = 0           # βασική ζώνη: ο RX-888 δειγματοληπτεί από DC
signal = "real"         # όχι "iq" - η απευθείας δειγματοληψία δίνει πραγματικό σήμα
accelerator = "opencl"  # "none" αν δεν υπάρχει περιβάλλον OpenCL
audio_sps = 12000
audio_compression = "flac"
waterfall_size = 1024
waterfall_compression = "zstd"
smeter_offset = 5
analog_smeter_offset = 5

[input.driver]
name = "stdin"
format = "s16"

[input.defaults]
frequency = 7120000
modulation = "LSB"
```

Τροφοδοτείται από το `rx888_stream`, κάτι που κάνει για εσάς το `start-rx888mk2.sh`:

```
rx888_stream -f /path/to/SDDC_FX3.img -s 60000000 -g 50 -a 0 -m low --pga -d -r -o - \
| build/spectrumserver --config config-rx888mk2.toml
```

Για το `fft_size`: στα 60 MSPS το σωστό μέγεθος είναι 4194304. Το 8388608 διπλασιάζει την ανάλυση του καταρράκτη και μαζί της τη μνήμη και το κόστος CPU κάθε μετασχηματισμού· στα περισσότερα μηχανήματα αγοράζει οξύτερα bin με τμήμα χαμένα καρέ. Ξεκινήστε από 4194304 και ανεβείτε μόνο αν ο διακομιστής έχει άνετο περιθώριο.

Παράδειγμα 3: HackRF για ευρυζωνικό FM

```
[input]
sps = 10000000
frequency = 100900000
signal = "iq"

[input.driver]
format = "s8"

[input.defaults]
frequency = 100900000
modulation = "WBFM"
```

Η εγκατάσταση ολοκληρώθηκε! Θα πρέπει τώρα να έχετε πλήρως λειτουργικό διακομιστή PhantomSDR-Plus.

73 de SV1BTL & SV2AMK

ΚΕΦΑΛΑΙΟ 3

Πίνακας διαχείρισης

ADMIN_PANEL_SETUP.md

Πίνακας διαχείρισης PhantomSDR-Plus — Οδηγός ρύθμισης

⚠Ειδοποίηση ασφαλείας: κρατήστε αυτόν τον πίνακα διαχείρισης στο οικιακό σας δίκτυο ή πίσω από VPN. Η δημόσια έκθεση της θύρας διαχείρισης δεν συνιστάται χωρίς κατάλληλη θωράκιση της ταυτοποίησης.

Επισκόπηση

Ο πίνακας διαχείρισης αποτελείται από δύο υπηρεσίες Python:

Υπηρεσία	Αρχείο	Ρόλος
Πίνακας διαχείρισης	admin_server.py	Εφαρμογή web Flask, δεσμεύεται στο 127.0.0.1 (μόνο εσωτερικά)
Αντίστροφος διαμεσολαβητής	proxy.py	Εκθέτει τόσο το SDR όσο και τον πίνακα διαχείρισης σε μία μόνο δημόσια θύρα

Και οι δύο υπηρεσίες διαβάζουν τις ρυθμίσεις τους από το `admin_config.json`, το οποίο γράφεται από το `setup_admin.sh`. **Μην επεξεργάζεστε απευθείας τις σταθερές θυρών μέσα στα αρχεία Python** — όλες οι θύρες και η IP του SDR αποθηκεύονται στο `admin_config.json`.

Τι σας προσφέρει ο πίνακας διαχείρισης

- **Πίνακας ελέγχου** — ζωντανή κατάσταση διακομιστή, CPU/RAM, κύριες διεργασίες, πρόσφατη έξοδος καταγραφής, τερματικό
- **Επεξεργαστής ρυθμίσεων** — προβολή, επεξεργασία και αποθήκευση οποιουδήποτε αρχείου `.toml`, `.sh`, `.json`, `.h`, `.cpp` της εγκατάστασής σας
- **Προβολέας καταγραφών** — παρακολούθηση οποιουδήποτε αρχείου καταγραφής σε πραγματικό χρόνο (διακομιστής SDR, πίνακας διαχείρισης, RADE, καταρρεύσεις, autorun, proxy)
- **Δείκτες** — προβολή και επεξεργασία δεικτών συχνότητας
- **Ιστορικό συνομιλίας** — προβολή του αρχείου συνομιλίας του WebSDR, πλήρης εκκαθάριση ή διαγραφή μεμονωμένων μηνυμάτων χωρίς επανεκκίνηση του διακομιστή
- **Μήνυμα καταρράκτη** — μετάδοση μόνιμου μηνύματος-πλαίσου σε όλους τους συνδεδεμένους χρήστες, ορατού σε πραγματικό χρόνο στην οθόνη καταρράκτη
- **Χρήστες** — ζωντανή λίστα συνδεδεμένων ακροατών με τη συχνότητα, τον τρόπο, τη διάρκεια και κουμπί Kick
- **Διαγραφή μηνυμάτων συνομιλίας** — κουμπί Delete που αφαιρεί άμεσα αυτό το ένα μήνυμα από το αρχείο
- **Μηνύματα μετάδοσης στον καταρράκτη** — σας επιτρέπει να προωθήσετε μόνιμο πλαίσιο κειμένου στον καταρράκτη κάθε συνδεδεμένου χρήστη

- **Spot Reporting** — εκκίνηση/διακοπή του αποκωδικοποιητή autorun FT8/FT4/J58/WSPR, επιλογή μπαντών/τρόπων και αναφορά spot στα PSK Reporter / wsprnet (απενεργοποιημένο εξ ορισμού), με δύο μετρητές: πλακίδια ανά αποκωδικοποιητή για την τρέχουσα εκτέλεση και ένα συνολικό άθροισμα δίπλα σε κάθε πλαίσιο ελέγχου μπάντας/τρόπου
- **Γραφήματα** — συχνότητα CPU, φορτίο CPU, θερμοκρασία CPU και συνδεδεμένοι χρήστες, σε γράφημα για τα τελευταία 15 λεπτά / 1 ώρα / 4 ώρες / 12 ώρες / 24 ώρες
- **Θερμική προστασία (Thermal Guard)** — σταματά τον διακομιστή αν υπερθερμανθεί η CPU και τον επανεκκινεί μόλις κρυώσει, με όρια που προκύπτουν από την ίδια σας τη CPU· λειτουργεί με οποιονδήποτε τρόπο εκκίνησης/διακοπής και έρχεται σε λειτουργία μόνο καταγραφής, οπότε δεν κάνει τίποτα μέχρι να το ενεργοποιήσετε
- **Ρυθμίσεις** — αλλαγή κωδικού διαχειριστή, βασικού καταλόγου SDR, ονόματος διεργασίας, δημόσιας θύρας

Απαιτήσεις

- Το PhantomSDR-Plus ήδη εγκατεστημένο και σε λειτουργία
- Python 3.8+

ΣΗΜΕΙΩΣΗ

Κανονικά το πρόγραμμα εγκατάστασης το κάνει αυτό για εσάς. Το `./install.sh` — όπως και καθένα από τα τέσσερα scripts των διανομών, και τα δύο scripts του RADE και το script του stats server — στήνει το admin panel από προεπιλογή, εγκαθιστώντας πρώτα το pip με τον διαχειριστή πακέτων του συστήματος αν λείπει. Αυτή η σελίδα περιγράφει τι γίνεται σε εκείνο το βήμα, και τι να κάνετε αν το προσπεράσατε ή θέλετε να αλλάξετε κάτι αργότερα.

Οι βιβλιοθήκες Python του panel (`flask`, `psutil`, `aiohttp`, `tomli-w`) εγκαθίστανται από το `setup_admin.sh`· αν αυτό αποτύχει, σας τυπώνει την ακριβή εντολή.

Βήμα 1 — Αρχεία

Τοποθετήστε αυτά τα τέσσερα αρχεία μέσα στον κατάλογο του PhantomSDR-Plus:

```
admin_server.py
manage_admin.sh
setup_admin.sh
proxy.py
```

Κάντε τα σενάρια κελύφους εκτελέσιμα:

```
chmod +x setup_admin.sh manage_admin.sh
```

Βήμα 2 — Εκτελέστε το σενάριο ρύθμισης

```
./setup_admin.sh
```

Το σενάριο θα:

1. Ελέγξει ότι είναι εγκατεστημένη η Python 3

2. Επαληθεύσει ότι υπάρχουν τα `admin_server.py` και `manage_admin.sh`
3. Καταγράψει το `127.0.0.1` ως `sdr_host` στη διαμόρφωση (δείτε [Η ρύθμιση sdr_host](#))
4. Ζητήσει τρεις αριθμούς θυρών. Κάθε ερώτηση προσφέρει μια προεπιλογή σε αγκύλες που γίνεται δεκτή με ένα σκέτο Enter, οπότε μια συνηθισμένη εγκατάσταση θέλει τρία πατήματα πλήκτρων:
5. **Θύρα του spectrumserver** — η θύρα στην οποία ακούει ο διακομιστής SDR (προεπιλογή 8900, ακριβώς αυτή που έχει κάθε `config-*.toml` του αποθετηρίου)
6. **Εσωτερική θύρα πίνακα διαχείρισης** — όπου δεσμεύεται τοπικά το `admin_server.py` (προεπιλογή 3000)
7. **Δημόσια θύρα διαμεσολαβητή** — η μοναδική εξωτερική θύρα που συνδυάζει SDR + διαχείριση (προεπιλογή 8902)

Οι μη έγκυρες απαντήσεις απορρίπτονται και η ερώτηση επαναλαμβάνεται, αλλά μόνο πέντε φορές — μετά χρησιμοποιείται η προεπιλογή. Μια εκτέλεση της οποίας η είσοδος δεν είναι τερματικό (μέσω `pipe`, `cron`, χωρίς επιτήρηση) παίρνει αμέσως τις προεπιλογές αντί να περιμένει είσοδο που δεν θα έρθει ποτέ.

1. Εγκαταστήσει τα `flask`, `psutil`, `aiohttp` και `tomli-w` μέσω `pip`
2. Παραχωρήσει στο `ss` τη δυνατότητα `cap_net_admin` (εφεδρική διαδρομή της λειτουργίας Kick Users, χρειάζεται μόνο όταν ο πίνακας τρέχει χωρίς τον διαμεσολαβητή)
3. Ρωτήσει ποιο σενάριο ξεκινά και ποιο σταματά τον δέκτη — μέσω αυτών ο πίνακας χειρίζεται τον διακομιστή, το ίδιο και η θερμική προστασία
4. Ρωτήσει πόσο μακριά επιτρέπεται να φτάσει μόνη της η θερμική προστασία της CPU — `log`, `throttle`, `stop` ή `stop+restart` — και να προσφέρει την εκτέλεση του `setup-cpufreq-perms.sh` ώστε το στάδιο `throttle` να δουλεύει χωρίς `root` (δείτε το [THERMAL_GUARD.md](#))
5. Γράψει το `admin_config.json` με όλες τις ρυθμίσεις
6. Προσφέρει την εγκατάσταση δύο μονάδων `systemd` (πίνακας + διαμεσολαβητής) — εκκίνηση με το `boot` και επανεκκίνηση μετά από κατάρρευση. **Συνιστάται και είναι η προεπιλογή** (σκέτο Enter την αποδέχεται): η προστασία υπερθέρμανσης της CPU τρέχει μόνο όσο τρέχει ο πίνακας, οπότε χωρίς τις μονάδες το μηχάνημα μένει απροστάτευτο μετά από επανεκκίνηση. Το βήμα παραλείπεται αυτόματα όπου το `systemd` δεν είναι PID 1 (`containers`, `WSL1`, `OpenRC`) ή όπου δεν έχετε `sudo`.

Μετά τη ρύθμιση, ανοίξτε τον περιηγητή σας μέσω του διαμεσολαβητή:

```
http://YOUR_SERVER_IP:<proxy_port>/admin
```

Προεπιλεγμένος κωδικός: `admin`

⚠ **Αλλάξτε αμέσως τον κωδικό** — πηγαίνετε στις Settings στην πρώτη σύνδεση. Ένας οδηγός πρώτης εκτέλεσης θα σας καθοδηγήσει στον ορισμό καταλόγου SDR, ονόματος διεργασίας, δημόσιας θύρας και νέου κωδικού.

Βήμα 3 — Εκκίνηση / διακοπή / επανεκκίνηση

Υπάρχουν δύο τρόποι να τρέχετε τον πίνακα και τον διαμεσολαβητή. **Διαλέξτε έναν και μείνετε σε αυτόν** — η προειδοποίηση στο τέλος της ενότητας εξηγεί τι συμβαίνει αν τους ανακατέψετε.

Η Μέθοδος B (systemd) συνιστάται έντονα όπου υπάρχει systemd. Η προστασία υπερθέρμανσης της CPU τρέχει μέσα στον πίνακα, οπότε ένας πίνακας που πεθαίνει στις 03:00 ή ένα μηχάνημα που επανεκκινεί αφήνει τη CPU απροστάτευτη μέχρι να τον ξεκινήσει κάποιος ξανά με το χέρι. Το `setup_admin.sh` προσφέρει πλέον τις μονάδες από προεπιλογή. Χρησιμοποιήστε τη Μέθοδο A όταν δεν υπάρχει systemd — containers, WSL1, OpenRC/sysvinit — ή όταν θέλετε σκόπιμα να επιβλέπετε μόνοι σας.

Μέθοδος A — `manage_admin.sh` (χωρίς root, χωρίς εγκατάσταση)

```
./manage_admin.sh start      # start admin panel and proxy
./manage_admin.sh stop      # stop both
./manage_admin.sh status    # show running status and PIDs
```

Σκόπιμα **δεν υπάρχει restart**. Παλιότερα τερμάτιζε τον πίνακα και ξεκινούσε αμέσως το δικό του αντίγραφο, κάτι που συγκρούεται με το systemd σε κάθε μηχάνημα που χρησιμοποιεί τη Μέθοδο B. Για επανεκκίνηση με τη Μέθοδο A, εκτελέστε `stop` και μετά `start` — δείτε [Επανεκκίνηση του πίνακα](#) παρακάτω.

Ένα σενάριο ελέγχει **και τις δύο** διεργασίες. Ελέγχει ότι είναι εγκατεστημένο το `aiohttp` πριν ξεκινήσει τον διαμεσολαβητή και εμφανίζει σαφές σφάλμα με την εντολή διόρθωσης αν λείπει.

Τίποτα δεν ξεκινά μόνο του: μετά από επανεκκίνηση, ή αν καταρρεύσει ο πίνακας, τον ξεκινάτε ξανά με το χέρι. Αυτό είναι εντάξει για έναν δέκτη που παρακολουθείτε, αλλά προσέξτε ότι η θερμική προστασία της CPU τρέχει μέσα στον πίνακα: όσο ο πίνακας είναι κάτω, τίποτα δεν προστατεύει το μηχάνημα (δείτε το [THERMAL_GUARD.md](#)).

Μέθοδος B — μονάδες systemd (εκκίνηση με το boot, επανεκκίνηση μετά από κατάρρευση)

Το αποθετήριο περιλαμβάνει δύο έτοιμες μονάδες, τις `phantomsdr-admin.service` και `phantomsdr-proxy.service`. Προσαρμόστε το `User=` και τις διαδρομές και στις δύο, και μετά:

```
sudo cp phantomsdr-admin.service phantomsdr-proxy.service /etc/systemd/system/
sudo systemctl daemon-reload
./manage_admin.sh stop      # free the ports first
sudo systemctl enable --now phantomsdr-admin
sudo systemctl enable --now phantomsdr-proxy
```

Οι καθημερινές εντολές γίνονται τότε:

```
sudo systemctl start   phantomsdr-admin
sudo systemctl stop    phantomsdr-admin
sudo systemctl restart phantomsdr-admin
systemctl status       phantomsdr-admin
sudo journalctl -u phantomsdr-admin -f
```

Οι ίδιες τέσσερις με `phantomsdr-proxy`, ή και τα δύο μαζί: `sudo systemctl restart phantomsdr-admin phantomsdr-proxy`.

Τα `admin.log` και `proxy.log` δουλεύουν ακριβώς όπως πριν — οι μονάδες γράφουν στα ίδια αρχεία.

Ένα πράγμα να κάνετε μία φορά: εγκαταστήστε τον κανόνα `tmpfiles` από το `tmpfiles/phantomsdr-logs.conf` (αλλάξτε πρώτα τις διαδρομές και τον χρήστη του). Το systemd δημιουργεί αυτά τα δύο αρχεία ως `root` πριν κατέβει στο `User=`, και τότε το κουμπί **Clear Logs** του πίνακα αποτυγχάνει σε αυτά με `Permission denied` — δείτε παρακάτω «To Clear Logs λέει `Permission denied`». Το `setup_admin.sh` εγκαθιστά τον κανόνα για εσάς.

Η μονάδα του πίνακα περιέχει `SupplementaryGroups=cpufreq`. Αυτό επιτρέπει στο στάδιο `throttle` της θερμικής προστασίας να χαμηλώνει τη συχνότητα της CPU χωρίς ο πίνακας να τρέχει ως `root`. Δημιουργήστε πρώτα αυτή την ομάδα με το `./setup-cpufreq-perms.sh`, αλλιώς το `systemd` θα αρνηθεί να ξεκινήσει τη μονάδα· διαγράψτε τη γραμμή αν δεν χρησιμοποιείτε το στάδιο `throttle`.

Επιστροφή στη Μέθοδο Α:

```
sudo systemctl disable --now phantomsdr-admin
sudo systemctl disable --now phantomsdr-proxy
```

⚠ **Μην ανακατεύετε τα δύο.** Με ενεργές τις μονάδες, το `./manage_admin.sh stop` αναιρείται από το `systemd` πέντε δευτερόλεπτα αργότερα, και το `./manage_admin.sh start` σας δίνει ένα δεύτερο, μη διαχειριζόμενο αντίγραφο που μάχεται με εκείνο του `systemd` για την ίδια θύρα. (Το `restart` αφαιρέθηκε από το script ακριβώς γι' αυτόν τον λόγο· τώρα τυπώνει σφάλμα και σας παραπέμπει στο `systemctl`.) Το `./manage_admin.sh status` μόνο διαβάζει, οπότε παραμένει χρήσιμο και στις δύο περιπτώσεις.

Επανεκκίνηση του πίνακα

Στη συνέχεια αυτού του εγγράφου γράφεται σε αρκετά σημεία «επανεκκινήστε τον πίνακα». Εννοείται όποιο από τα δύο ταιριάζει στη μέθοδό σας:

```
# Μέθοδος Β – μονάδες systemd (η κανονική εγκατάσταση)
sudo systemctl restart phantomsdr-admin phantomsdr-proxy

# Μέθοδος Α – χωρίς εγκατεστημένες μονάδες
./manage_admin.sh stop && ./manage_admin.sh start
```

Η μονάδα συνοδεύεται από `KillMode=process`, και αυτή η γραμμή είναι καθοριστική. Αν ξεκινάτε τον SDR από τον πίνακα, ο δέκτης είναι παιδί της μονάδας διαχείρισης και κληρονομεί το `cgroup` της. Με την προεπιλογή του `systemd`, `KillMode=control-group`, η επανεκκίνηση του πίνακα θα κατέβαζε μαζί της τον `spectrumserver`, τον `watchdog` του και τον δαίμονα `autorun` — αφού πρώτα κολλούσε για ολόκληρα τα 90 δευτερόλεπτα του χρονικού ορίου τερματισμού. Με `KillMode=process` το `systemd` στέλνει σήμα μόνο στον ίδιο τον πίνακα, οπότε το `sudo systemctl restart phantomsdr-admin` αφήνει στον αέρα έναν δέκτη με ακροατές.

Αναβάθμιση μονάδας που εγκαταστάθηκε πριν από την v4.1.0. Τα παλιότερα αρχεία μονάδας δεν έχουν αυτή τη γραμμή, και η επανεκκίνηση του πίνακα δεν την προσθέτει: το `systemctl restart` ξανατρέχει το πρόγραμμα, δεν αλλάζει τις ρυθμίσεις του. Επεξεργαστείτε το *εγκατεστημένο* αντίγραφο — αυτό στο αποθετήριο είναι απλώς πρότυπο, το `systemd` δεν το διαβάζει ποτέ:

```
sudo nano /etc/systemd/system/phantomsdr-admin.service
```

Προσθέστε `KillMode=process` στην ενότητα `[Service]`, δίπλα στο `Restart=always`. Έπειτα ζητήστε από το `systemd` να ξαναδιαβάσει το αρχείο και επιβεβαιώστε:

```
sudo systemctl daemon-reload
systemctl show phantomsdr-admin -p KillMode
```

Η δεύτερη εντολή πρέπει να τυπώσει `KillMode=process`. Αν εξακολουθεί να τυπώνει `control-group`, το `systemd` χρησιμοποιεί το αποθηκευμένο αντίγραφό του — το `systemctl show phantomsdr-admin -p NeedDaemonReload` δείχνει `yes` όσο εκκρεμεί η επαναφόρτωση. Τα αρχικά κενά στα αρχεία μονάδας αγνοούνται, οπότε η στοίχιση δεν παίζει ρόλο.

Βήμα 4 — Ανοίξτε τη θύρα στο τείχος προστασίας

Ανοίξτε στο τείχος προστασίας μόνο τη **δημόσια θύρα του διαμεσολαβητή**. Δεν χρειάζεται να εκθέσετε εξωτερικά την εσωτερική θύρα του πίνακα διαχείρισης:

```
sudo ufw allow <proxy_port>
```

Βήμα 5 — Λειτουργία Kick Users

Η σελίδα Users δείχνει κάθε συνδεδεμένο ακροατή. Κάθε γραμμή εμφανίζει την IP του χρήστη, τη συχνότητα συντονισμού, τον τρόπο και τη διάρκεια σύνδεσης. Για να αποσυνδέσετε έναν χρήστη, κάντε κλικ στο κουμπί **Kick** της γραμμής του — τερματίζεται μόνο αυτή η σύνδεση TCP. Όλοι οι άλλοι ακροατές παραμένουν συνδεδεμένοι και δεν ακούν τίποτα.

Η αποσύνδεση είναι άμεση και χειρουργική: ο διακομιστής δεν επανεκκινείται, ο ήχος κανενός άλλου δεν διακόπτεται και ο χρήστης που αποσυνδέθηκε μπορεί να επανασυνδεθεί αμέσως (η λειτουργία αφορά την απομάκρυνση προβληματικών ή κολλημένων συνδέσεων, όχι τον αποκλεισμό).

Στο παρασκήνιο η αποσύνδεση χρησιμοποιεί `ss -K dst <IP> dport <port>` για να τερματίσει τη συγκεκριμένη υποδοχή TCP. Αυτό απαιτεί μια δυνατότητα του Linux που το σενάριο ρύθμισης παραχωρεί αυτόματα. Αν παραλείψατε τη ρύθμιση ή το κουμπί εμφανίζει σφάλμα, εφαρμόστε τη χειροκίνητα:

```
sudo setcap cap_net_admin+ep $(which ss)
# Verify:
getcap $(which ss)    # should show: cap_net_admin=ep
```

Ρυθμίσεις — admin_config.json

Όλες οι ρυθμίσεις εκτέλεσης βρίσκονται στο `admin_config.json` στον κατάλογο του PhantomSDR-Plus. Το `setup_admin.sh` γράφει το αρχικό αρχείο· οι επόμενες αλλαγές μπορούν να γίνουν από τη σελίδα Settings ή με απευθείας επεξεργασία του αρχείου.

Βασικά πεδία που γράφει το `setup_admin.sh`:

Κλειδί	Περιγραφή
<code>port</code>	Εσωτερική θύρα πίνακα διαχείρισης (εκεί δεσμεύεται το <code>admin_server.py</code>)
<code>public_port</code>	Θύρα του spectrumserver (χρησιμοποιείται από τη σελίδα Users για το <code>/users</code>)
<code>proxy_port</code>	Δημόσια θύρα στην οποία ακούει ο διαμεσολαβητής
<code>sdr_host</code>	Ο υπολογιστής μέσω του οποίου ο διαμεσολαβητής φτάνει στον spectrumserver — <code>127.0.0.1</code> (δείτε παρακάτω)
<code>password_hash</code>	Σύνοψη SHA-256 του κωδικού διαχειριστή
<code>sdr_base_dir</code>	Διαδρομή της εγκατάστασης του PhantomSDR-Plus
<code>sdr_process_name</code>	Όνομα διεργασίας προς παρακολούθηση (προεπιλογή: <code>spectrumserver</code>)
<code>start_script</code> / <code>stop_script</code>	Τα σενάρια που εκτελεί ο πίνακας για να ξεκινήσει και να σταματήσει τον διακομιστή — μέσω αυτών ενεργεί και η θερμική προστασία

Πολλοί δέκτες σε έναν υπολογιστή; Ορίστε το `start_script` σε `start-all.sh`. Το σενάριο διακοπής ήδη σταματά κάθε δέκτη, ενώ το Restart και η θερμική προστασία χρησιμοποιούν το σενάριο εκκίνησης· με το `start-all.sh` επαναφέρουν κάθε δέκτη του `receivers.toml` αντί μόνο τον κύριο. Η σελίδα Users, ο αριθμός ακροατών και το kick καλύπτουν τότε επίσης κάθε δέκτη. Δείτε [Πολλοί δέκτες](#).

Κλειδιά θερμικής προστασίας (όλα προαιρετικά — ο πίνακας τα γράφει όταν αποθηκεύσετε τις ρυθμίσεις Thermal Guard, και όσα λείπουν παίρνουν αυτές τις προεπιλογές):

Κλειδί	Προεπιλογή	Περιγραφή
<code>thermal_enabled</code>	<code>true</code>	Γενικός διακόπτης της προστασίας
<code>thermal_mode</code>	<code>"log"</code>	<code>log</code> (μόνο καταγραφή, καμία ενέργεια) · <code>throttle</code> · <code>stop</code> · <code>stop+restart</code>
<code>thermal_warn</code>	<code>null</code>	Θερμοκρασία προειδοποίησης °C· <code>null</code> = υπολογισμός από το κρίσιμο όριο της CPU σας
<code>thermal_throttle</code>	<code>null</code>	Θερμοκρασία στην οποία χαμηλώνει η μέγιστη συχνότητα της CPU· <code>null</code> = αυτόματο
<code>thermal_stop</code>	<code>null</code>	Θερμοκρασία στην οποία σταματά ο διακομιστής· <code>null</code> = αυτόματο
<code>thermal_resume</code>	<code>null</code>	Θερμοκρασία κάτω από την οποία πρέπει να πέσει η CPU για να επανέλθει· <code>null</code> = αυτόματο
<code>thermal_sustain_s</code>	<code>60</code>	Δευτερόλεπτα που πρέπει να διατηρηθεί η θερμοκρασία διακοπής πριν γίνει ενέργεια
<code>thermal_warn_sustain_s</code>	<code>30</code>	Δευτερόλεπτα διατήρησης για τα όρια προειδοποίησης/throttle
<code>thermal_resume_s</code>	<code>300</code>	Δευτερόλεπτα κάτω από το όριο επανόδου πριν αρθεί το κλείδωμα
<code>thermal_max_stops_hour</code>	<code>2</code>	Θερμικές διακοπές ανά ώρα πριν απενεργοποιηθεί η αυτόματη επανεκκίνηση
<code>thermal_test_temp</code>	<code>null</code>	Πλασματική θερμοκρασία για δοκιμή· <code>null</code> = χρήση του πραγματικού αισθητήρα

Οι αλλαγές σε αυτά ισχύουν στο επόμενο δείγμα — μέσα σε λίγα δευτερόλεπτα, χωρίς επανεκκίνηση.

Για να αλλάξετε θύρες μετά την αρχική ρύθμιση, επεξεργαστείτε το `admin_config.json` και επανεκκινήστε:

```
sudo systemctl restart phantomsdr-admin phantomsdr-proxy # B
./manage_admin.sh stop && ./manage_admin.sh start # A
```

Η ρύθμιση `sdr_host`

Το `proxy.py` συνδέεται στον `spectrumserver` μέσω βρόχου (loopback) — το `sdr_host` είναι `127.0.0.1` στο `admin_config.json`, γραμμένο εκεί από το `setup_admin.sh`. Κανονικά δεν χρειάζεται να το αλλάξετε.

Βάλτε πραγματική διεύθυνση μόνο αν ο `spectrumserver` τρέχει σε **άλλο μηχάνημα** από τον διαμεσολαβητή. Μετά κάντε [Επανεκκίνηση του πίνακα](#).

Τι άλλαξε: παλαιότερες εκδόσεις απαιτούσαν το `sdr_host` να είναι η τοπική IP του μηχανήματος, επειδή ο `spectrumserver` έκλεινε τα WebSocket που έφταναν από τη διεύθυνση βρόχου. Αυτό το φίλτρο αφαιρέθηκε, οπότε οι τοπικές συνδέσεις λειτουργούν κανονικά. Δύο συνέπειες: η περιήγηση στο `http://localhost:<port>` από το ίδιο το μηχάνημα του διακομιστή δείχνει πλέον ολόκληρο το περιβάλλον (πριν φόρτωνε η σελίδα αλλά έμενε κενή), και ο διαμεσολαβητής δεν χαλάει πια όταν το DHCP δώσει νέα IP στο μηχάνημα. Αν αναβαθμίζετε και το `admin_config.json` σας έχει ακόμη τοπική IP, αλλάξτε την σε `127.0.0.1` ή εκτελέστε ξανά το `setup_admin.sh`.

proxy.py — τι είναι και πότε το χρειάζεστε

Το `proxy.py` τοποθετεί τόσο τον διακομιστή SDR όσο και τον πίνακα διαχείρισης σε **μία μόνο δημόσια θύρα**, δρομολογώντας βάσει προθέματος διαδρομής:

Διαδρομή	Δρομολογείται σε
<code>/admin*</code>	Πίνακας διαχείρισης (<code>localhost:<port></code>)
Οτιδήποτε άλλο	Spectrumserver (<code><sdr_host>:<public_port></code>)

Χωρίς τον διαμεσολαβητή πρέπει να εκθέσετε δύο θύρες ξεχωριστά. Με αυτόν εκθέτετε μόνο μία.

Ο διαμεσολαβητής αφαιρεί επίσης το όριο των 4 MB στο μέγεθος μηνύματος WebSocket (σημαντικό για μεγάλα πλαίσια FFT από το RX-888 στα 60 MSPS), προωθεί τις πραγματικές IP των πελατών μέσω κεφαλίδων `X-Forwarded-*` και διατηρεί ζωντανές τις μακρόβιες συνδέσεις μέσα από NAT με παλμό κάθε 30 δευτερόλεπτα.

Αλλαγή θυρών μετά τη ρύθμιση

Επεξεργαστείτε απευθείας το `admin_config.json`:

```
{
  "port":      3000,
  "public_port": 9001,
  "proxy_port": 9002,
  "sdr_host":   "127.0.0.1"
}
```

Έπειτα επανεκκινήστε και τις δύο υπηρεσίες:

```
sudo systemctl restart phantomsdr-admin phantomsdr-proxy # B
./manage_admin.sh stop && ./manage_admin.sh start        # A
```

Αν χρειάζεστε επίσης το `admin_server.py` να δεσμεύεται σε άλλη θύρα κατά την εκκίνηση (π.χ. στην υπηρεσία `systemd`), μπορείτε να το αντικαταστήσετε με τη μεταβλητή περιβάλλοντος:

```
ADMIN_PORT=3000 python3 admin_server.py
```

Εξ ορισμού το `admin_server.py` δεσμεύεται μόνο στο `127.0.0.1`. Για αυτόνομη εκτέλεση χωρίς διαμεσολαβητή (ανάπτυξη/δοκιμές), δεσμεύστε το σε όλες τις διεπαφές:

```
ADMIN_BIND=0.0.0.0 python3 admin_server.py
```

Χρήσιμες χειροκίνητες εντολές

```
# Start admin panel manually (foreground)
python3 ~/PhantomSDR-Plus/admin_server.py

# Kill the admin panel
pkill -f admin_server.py

# Kill the proxy
pkill -f proxy.py

# Free a port that is stuck in use
sudo fuser -k 3000/tcp

# Check what is listening on a port
ss -tlnp | grep 3000
```

Αρχεία καταγραφής

Αρχείο	Περιεχόμενο
admin.log	Έξοδος του πίνακα διαχείρισης
proxy.log	Πλαίσιο εκκίνησης διαμεσολαβητή + μία γραμμή πρόσβασης ανά προωθημένο αίτημα
logwebsdr.txt	Log του script εκκίνησης και του watchdog — έναρξη/παύση, driver δέκτη, εγγραφή στο websdr.org
spectrumserver.log	Η έξοδος της ίδιας της διεργασίας του δέκτη — ρύθμιση FFT και OpenCL, συνδέσεις πελατών, σφάλματα. Το μεγαλύτερο από αυτά τα αρχεία περιστρέφεται, οπότε δίπλα του μπορεί να υπάρχει spectrumserver.log.1
rade.log	Αρχείο καταγραφής του sidecar RADE/FreeDV
autorun.log	Αρχείο καταγραφής του δαίμονα αποκωδικοποίησης — spots, SNR, ολίσθηση
crash.log	Αναφορές κατάρρευσης και θερμικά συμβάντα: ίχνη [CRASH] / [TERMINATE], σημειώσεις [EXIT] και γραμμές [THERMAL] από τη θερμική προστασία. Κανονικά κενό — ό,τι υπάρχει μέσα αξίζει να διαβαστεί

Όλα αυτά είναι αναγνώσιμα από τις καρτέλες **Log Viewer** του πίνακα (LOGWEBSDR, SPECTRUMSERVER, ADMIN, RADE, CRASH, AUTORUN, PROXY). Ο viewer δείχνει τις τελευταίες 150 γραμμές της επιλεγμένης καρτέλας και ανανεώνεται κάθε 3 δευτερόλεπτα όταν είναι ενεργή η αυτόματη ανανέωση· διαβάζει μόνο το τέλος του αρχείου, οπότε το μέγεθός του δεν έχει σημασία.

Περιστροφή των proxy.log και admin.log

Το proxy.log καταγράφει κάθε προωθημένο αίτημα και το admin.log κάθε αίτημα που φτάνει στον ίδιο τον πίνακα. Κανένα από τα δύο δεν περιστρέφεται εξ ορισμού. Οι ερωτήσεις του πίνακα ελέγχου κυριαρχούσαν άλλοτε και στα δύο αρχεία — το /admin/api/status εκτελείται κάθε ~3 δευτερόλεπτα όσο μια καρτέλα παραμένει ανοιχτή — αλλά πλέον φιλτράρονται και από τα δύο (δείτε παρακάτω), οπότε η περιστροφή περιορίζει αργή αύξηση και όχι πλημμύρα.

Το αποθετήριο περιλαμβάνει ρύθμιση logrotate στο logrotate/phantomsdr (ημερήσια ή νωρίτερα αν ένα αρχείο ξεπεράσει τα 10 MB· διατηρεί 7 συμπιεσμένα αρχεία στο logproxy/, ώστε να μην γεμίζει η ρίζα του έργου). **Είναι ειδική για κάθε εγκατάσταση — ανοίξτε την και αλλάξτε τις τρεις διαδρομές αρχείων, τη διαδρομή olddir και τον χρήστη su ώστε να ταιριάζουν στο μηχάνημά σας πριν την εγκαταστήσετε**, και έπειτα:

```
sudo cp logrotate/phantomcdr /etc/logrotate.d/phantomcdr
sudo logrotate -d /etc/logrotate.d/phantomcdr # dry run, should list all three logs
```

Το `logrotate/phantomcdr` στο αποθετήριο και το `/etc/logrotate.d/phantomcdr` είναι δύο **ανεξάρτητα αντίγραφα** — η `cp` δεν τα συνδέει. Η επεξεργασία του αρχείου του αποθετηρίου δεν αλλάζει τίποτα σε ζωντανό σύστημα μέχρι να εκτελέσετε ξανά την παραπάνω `sudo cp`. Μόνο το αντίγραφο στο `/etc` διαβάζει πραγματικά το `logrotate`.

Η ρύθμιση χρησιμοποιεί `copytruncate`, που είναι απαραίτητο: το `manage_admin.sh` ξεκινά και τις δύο διεργασίες με `>> <log>` και καμία δεν ξαναανοίγει τη `standard` έξοδό της, οπότε μια περιστροφή βασισμένη σε μετονομασία θα τις άφηνε να γράφουν στο περιστραμμένο αρχείο ενώ το νέο θα έμενε για πάντα άδειο.

Τα αρχεία καταλήγουν στο `logproxy/` (δημιουργείται αυτόματα από το `createolddir`) και ονομάζονται `proxy.log.1.gz ... proxy.log.7.gz`, με το νεότερο πρώτο. Διαβάστε ένα με `zcat logproxy/proxy.log.1.gz | less` ή αναζητήστε σε όλα ταυτόχρονα με `zgrep "pattern" logproxy/*.gz`.

Το κουμπί **Clear Logs** του πίνακα μηδενίζει **κάθε** αρχείο του παραπάνω πίνακα, μαζί και το `autorun.log` και το `proxy.log` (το `CLEAR_EXCLUDE` στο `admin_server.py` είναι κενό). Προσέξτε τι κοστίζει αυτό στο `autorun.log`: το ιστορικό αποκωδικοποίησής του είναι η μόνη καταγραφή του τι ακούστηκε και πότε, και δεν μπορεί να ανακατασκευαστεί — από ένα μηδένισμα σώζεται μόνο ό,τι έχει ήδη περιστρέψει το `logrotate` στο `logproxy/`, οπότε χάνεται καθετί που αποκωδικοποιήθηκε μετά την τελευταία περιστροφή. Ο μηδενισμός με τον αποκωδικοποιητή σε λειτουργία είναι ασφαλής: ο δαίμονας κρατά το αρχείο ανοιχτό με `O_APPEND` και συνεχίζει να γράφει στο ίδιο inode. Η ειδοποίηση αναφέρει τι μηδενίστηκε, και αν κάποιο αρχείο δεν μπόρεσε να γραφτεί — π.χ. ένα `proxy.log` που σε εγκατάσταση με χωριστούς χρήστες ανήκει σε άλλον χρήστη — αναφέρει εκείνο το αρχείο και το σφάλμα αντί να δηλώσει επιτυχία, και αφήνει το παράθυρο εκείνου του αρχείου όπως είναι αντί να το αδειάσει. Τα αρχεία μηδενίζονται επιτόπου, ποτέ δεν διαγράφονται: τόσο το `manage_admin.sh` όσο και οι μονάδες `systemd` τα ανοίγουν με `O_APPEND`, οπότε ένα διαγραμμένο αρχείο θα συνέχιζε να γεμίζει ένα αόρατο inode μέχρι την επανεκκίνηση της υπηρεσίας.

Οι επιτυχείς ερωτήσεις φιλτράρονται και από τα **δύο** αρχεία καταγραφής πρόσβασης, από το `QuietPollFilter` στο `proxy.py` και από το αντίστοιχο στο `admin_server.py`. Ο πίνακας ελέγχου καλεί το `/admin/api/status` κάθε 3 δευτερόλεπτα και τα `/admin/api/logs`, `/admin/api/autorun/status`, `/admin/api/users`, `/admin/api/graph-stats` και `/admin/api/chat` κάθε 5 δευτερόλεπτα· αφιλτράριστα θα αποτελούσαν ~95 % και των δύο αρχείων. Απορρίπτονται μόνο τα απλά 200 σε αυτές τις διαδρομές — οποιαδήποτε άλλη κατάσταση, διαδρομή ή μέθοδος καταγράφεται όπως πριν, ώστε μια αποτυχημένη ερώτηση να παραμένει ορατή και οι ενέργειες που αλλάζουν κατάσταση (`kick`, `autorun/start`, ...) να έχουν ξεχωριστές διαδρομές και να καταγράφονται πάντα. Η μία σκόπιμη εξαίρεση είναι το `/admin/api/logs/clear`, φιλτραρισμένο και στα δύο αρχεία: η δική του γραμμή πρόσβασης γράφεται μετά τον μηδενισμό, οπότε χωρίς το φίλτρο κάθε επιτυχημένος καθαρισμός άφηνε πίσω μία φρέσκια γραμμή στο αρχείο που μόλις είχε αδειάσει, και το κουμπί έμοιαζε χαλασμένο.

Επειδή το `proxy.log` είναι αρχείο πρόσβασης, περιέχει διευθύνσεις IP επισκεπτών και συμβολοσειρές `user-agent` — έχετε το υπόψη πριν το μοιραστείτε ζητώντας βοήθεια.

To «Clear Logs» λέει **Permission denied**

Σε μηχανήμα όπου οι μονάδες `systemd` εγκαταστάθηκαν σε καθαρό δέντρο, το κουμπί μπορεί να επιστρέψει:

```
x Could not clear admin.log ([Errno 13] Permission denied: /home/you/PhantomSDR-Plus/admin.log),
proxy.log ([Errno 13] Permission denied: /home/you/PhantomSDR-Plus/proxy.log)
```

Είναι πάντα αυτά τα δύο αρχεία και ποτέ τα άλλα πέντε, επειδή είναι τα μόνα που δημιουργεί το ίδιο το systemd. Το `StandardOutput=append` ανοίγεται από το PID 1 **πριν** κατέβει στο `User=`, οπότε ένα αρχείο καταγραφής που δεν υπάρχει ακόμη δημιουργείται ως `root:root 0644`. Ο πίνακας τρέχει ως ο χρήστης σας και δεν μπορεί να το αδειάσει. Τα άλλα πέντε αρχεία τα δημιουργούν τα σενάρια εκκίνησης, που ήδη τρέχουν ως ο χρήστης σας, γι' αυτό καθαρίζονται κανονικά. Όπου τα δύο αρχεία προϋπήρχαν των μονάδων — τα έφτιαξε το `manage_admin.sh` με `>>` — ανήκουν ήδη σωστά και το κουμπί δουλεύει· γι' αυτό το πρόβλημα εμφανίζεται μόνο σε νέα εγκατάσταση που ξεκίνησε κατευθείαν με systemd.

Επιβεβαιώστε με `ls -l admin.log proxy.log`: όσα αποτυγχάνουν δείχνουν `root root`. Διορθώστε τα:

```
sudo chown "$(id -un):$(id -gn)" admin.log proxy.log
sudo chmod 664 admin.log proxy.log
```

Δεν χρειάζεται επανεκκίνηση — το systemd συνεχίζει να γράφει μέσω του περιγραφέα αρχείου που ήδη κρατά, και η προσθήκη δεδομένων δεν αλλάζει την ιδιοκτησία. Πατήστε ξανά **Clear Logs** και τα δύο αδειάζουν.

Μη διαγράψετε τα αρχεία. Και τα δύο είναι ανοιχτά με `O_APPEND`· η διαγραφή αφήνει το systemd να γεμίζει έναν αόρατο inode μέχρι την επανεκκίνηση της υπηρεσίας, και το αρχείο που επανεμφανίζεται ανήκει πάλι στον root.

Για να μην ξανασυμβεί, εγκαταστήστε τον κανόνα `tmpfiles` που περιλαμβάνει το αποθετήριο στο `tmpfiles/phantomsdr-logs.conf`. Ξαναδημιουργεί και τα δύο αρχεία με τον σωστό ιδιοκτήτη σε κάθε εκκίνηση, πριν ξεκινήσουν οι μονάδες, ώστε ένα αρχείο που αφαιρέθηκε — χειροκίνητο καθάρισμα, επανεγκατάσταση, μετακίνηση του καταλόγου εγκατάστασης — να μην επιστρέφει ποτέ ως root. **Είναι ειδικός για κάθε μηχανήμα: ανοίξτε τον και αλλάξτε πρώτα τις δύο διαδρομές και το user:group ώστε να ταιριάζουν**, και μετά:

```
sudo cp tmpfiles/phantomsdr-logs.conf /etc/tmpfiles.d/99-phantomsdr-logs.conf
sudo systemd-tmpfiles --create /etc/tmpfiles.d/99-phantomsdr-logs.conf
```

Όπως και με το `logrotate/phantomsdr`, το αρχείο του αποθετηρίου και το αντίγραφο στο `/etc` είναι ανεξάρτητα — η επεξεργασία του πρώτου δεν αλλάζει τίποτα μέχρι να ξανατρέξετε το `cp`. Σε systemd 254 και νεότερο ο κανόνας διορθώνει επίσης την ιδιοκτησία και τα δικαιώματα ενός αρχείου που ήδη υπάρχει, οπότε λύνει και την τρέχουσα αποτυχία· σε παλαιότερες εκδόσεις το `chown` παραπάνω είναι αυτό που ξεμπλοκάρει το κουμπί.

Το `setup_admin.sh` κάνει όλα τα παραπάνω για εσάς όταν εγκαθιστά τις μονάδες, οπότε μια εγκατάσταση μέσω του σεναρίου δεν συναντά ποτέ αυτό το πρόβλημα. Τα βήματα αφορούν χειροκίνητη εγκατάσταση, ή εγκατάσταση παλαιότερη από το βήμα ιδιοκτησίας των logs.

Το `logrotate` δεν επαναφέρει το πρόβλημα: η διαμόρφωση που παρέχεται χρησιμοποιεί `copytruncate`, που διατηρεί τον ίδιο inode και άρα τον ίδιο ιδιοκτήτη.

Σύνοψη πρόσβασης

Διάταξη	SDR	Πίνακας διαχείρισης
Χωρίς διαμεσολαβητή	<code>http://YOUR_IP:<public_port></code>	<code>http://YOUR_IP:<port>/admin</code>
Με διαμεσολαβητή	<code>http://YOUR_IP:<proxy_port></code>	<code>http://YOUR_IP:<proxy_port>/admin</code>

Διαγραφή μηνυμάτων συνομιλίας

Η σελίδα Chat History παραθέτει κάθε μήνυμα του τρέχοντος αρχείου συνομιλίας. Κάθε εγγραφή έχει κουμπί **Delete** που αφαιρεί άμεσα αυτό το ένα μήνυμα από το αρχείο — χωρίς επανεκκίνηση διακομιστή.

Πώς λειτουργεί:

- Ο πίνακας διαχείρισης ξαναγράφει επιτόπου το αρχείο συνομιλίας, αφαιρώντας μόνο τη γραμμή του επιλεγμένου μηνύματος.
- Η αλλαγή ισχύει για όποιον φορτώσει ξανά τη συνομιλία· το ήδη φορτωμένο ιστορικό σε ανοιχτές καρτέλες δεν ενημερώνεται αναδρομικά.
- Η εκκαθάριση όλου του αρχείου (κουμπί **Clear All**) μηδενίζει το αρχείο, κάτι που επίσης ισχύει χωρίς επανεκκίνηση.
- Μια **απάντηση στη συνομιλία** αποθηκεύεται ως συνηθισμένο μήνυμα που ξεκινά με μια σήμανση η οποία ονομάζει το μήνυμα στο οποίο απαντά, για παράδειγμα [RE:2026-09-25 12:00:01|SV1BTL], οπότε σε αυτή τη λίστα εμφανίζεται με αυτή τη σήμανση μπροστά. Η διαγραφή μιας απάντησης αφαιρεί μόνο αυτήν. Η διαγραφή του μηνύματος στο οποίο απαντά αφήνει τις απαντήσεις του στη θέση τους, δείχνοντας μόνο το όνομα και την ώρα στα οποία απαντούν.

Το κουμπί διαγραφής εμφανίζεται με συνέπεια και στις πέντε παραλλαγές της εφαρμογής Svelte. Αν μια διαγραφή φαίνεται να μην ισχύει, επιβεβαιώστε ότι ο πίνακας διαχείρισης έχει δικαίωμα εγγραφής στο αρχείο συνομιλίας:

```
ls -l ~/PhantomSDR-Plus/chat.jsonl # path depends on your config
```

Μηνύματα μετάδοσης στον καταρράκτη

Ο πίνακας **Waterfall Message** σας επιτρέπει να προωθήσετε μόνιμο πλαίσιο κειμένου στην οθόνη καταρράκτη κάθε συνδεδεμένου χρήστη, χωρίς να αγγίξετε τη διεργασία του διακομιστή.

Αποστολή μηνύματος

1. Ανοίξτε τον πίνακα διαχείρισης και πηγαίνετε στο **Waterfall Message**.
2. Πληκτρολογήστε το κείμενο του μηνύματος και επιλέξτε χρώμα (δεκαεξαδικό, π.χ. #ffdd00).
3. Κάντε κλικ στο **Send** — το μήνυμα εμφανίζεται αμέσως σε όλες τις ενεργές προβολές καταρράκτη.

Εκκαθάριση του μηνύματος

Κάντε κλικ στο **Clear** για να αφαιρέσετε το πλαίσιο από όλους τους καταρράκτες. Η κατάσταση του μηνύματος διατηρείται στη μνήμη του πίνακα διαχείρισης· καθαρίζεται αυτόματα αν επανεκκινηθεί η διεργασία του πίνακα.

Πώς λειτουργεί

Ο πίνακας διαχείρισης εκθέτει δύο εσωτερικά σημεία που προωθεί ο διαμεσολαβητής:

Σημείο	Μέθοδος	Σκοπός
/admin/api/waterfall-message	POST	Ορισμός ή εκκαθάριση του τρέχοντος κειμένου και χρώματος
/admin/api/waterfall-message	GET	Επιστροφή της τρέχουσας κατάστασης μηνύματος ως JSON

Το frontend ερωτά την κατάσταση του μηνύματος και το αποδίδει ως επικάλυψη στον καμβά του καταρράκτη. Δεν χρειάζεται επανασύνδεση WebSocket ούτε επαναφόρτωση σελίδας από την πλευρά του πελάτη.

Τυπικές χρήσεις

- Ανακοίνωση προγραμματισμένης συντήρησης: "Server restart in 10 minutes"
- Επισήμανση συνθηκών διάδοσης: "Solar flux 180 – 10m wide open"
- Μήνυμα καλωσορίσματος: "Welcome to SV1BTL WebSDR – Athens, KM17"

Γραφήματα συστήματος

Η σελίδα **Γραφήματα** σχεδιάζει τέσσερα μεγέθη πάνω σε έναν κοινό άξονα χρόνου:

- **Συχνότητα CPU** — μέση συχνότητα σε όλους τους πυρήνες, σε GHz
- **Φορτίο CPU** — συνολική αξιοποίηση, σε ποσοστό
- **Θερμοκρασία CPU** — σε °C, με διακεκομμένες γραμμές στους 70 °C (θερμό) και 80 °C (κρίσιμο)
- **Χρήστες online** — ακροατές συνδεδεμένοι στο WebSDR εκείνη τη στιγμή

Επιλέξτε διάστημα με **15 MIN / 1 HOUR / 4 HOURS / 12 HOURS / 24 HOURS**. Περνώντας τον δείκτη πάνω από το γράφημα εμφανίζεται σταυρόνημα με τις τέσσερις τιμές εκείνης της στιγμής. Οι τέσσερις κάρτες πάνω από το γράφημα δείχνουν πάντα το νεότερο δείγμα.

Πώς γίνεται η δειγματοληψία

Ένα νήμα παρασκηνίου στο `admin_server.py` παίρνει ένα δείγμα κάθε **2 δευτερόλεπτα**. Τα δείγματα κρατούνται σε δύο επίπεδα: η τελευταία **1 ώρα** σε πλήρη ανάλυση 2 δευτερολέπτων (1800 σημεία) και **24 ώρες** με μέσους όρους 30 δευτερολέπτων (2880 σημεία) για τα μεγάλα διαστήματα. Η σελίδα επιλέγει το επίπεδο που καλύπτει το διάστημα που ζητήσατε και γράφει 30s averages στην κεφαλίδα όταν βλέπετε το επίπεδο των μέσων όρων.

Μια ολόκληρη ημέρα σε ανάλυση 2 δευτερολέπτων θα ήταν ~43000 σημεία — δεκάδες MB μνήμης, αρχική μεταφορά πολλών megabyte και περίπου 40 σημεία ανά εικονοστοιχείο, που καμία οθόνη δεν μπορεί να αποδώσει. Ο μέσος όρος των 30 δευτερολέπτων κρατά μια ημέρα σε 2880 σημεία.

Η δειγματοληψία ξεκινά μαζί με τον πίνακα διαχείρισης, όχι όταν ανοίξετε πρώτη φορά τη σελίδα — έτσι η σελίδα ανοίγει με ιστορικό που ήδη υπάρχει.

Ο buffer είναι **μόνο στη μνήμη**: τίποτα δεν γράφεται στον δίσκο και το ιστορικό χάνεται όταν επανεκκινήσει ο πίνακας διαχείρισης. Αυτό είναι σκόπιμο — κρατά τη λειτουργία εκτός περιστροφής αρχείων καταγραφής και εκτός δίσκου.

Τα 2 δευτερόλεπτα δίνουν σχεδόν ζωντανή εικόνα και είναι αρκετά μικρό διάστημα ώστε να μην αλλοιώνεται η συχνότητα CPU, η οποία σε σύγχρονο επεξεργαστή πηδά από τη συχνότητα ηρεμίας στη turbo από δείγμα σε δείγμα. Ένα δείγμα κοστίζει περίπου ένα χιλιοστό του δευτερολέπτου, οπότε η δειγματοληψία είναι αμελητέα δίπλα στον SDR server. Η σελίδα ρωτά με τον ρυθμό του κάθε επιπέδου — κάθε 2 δευτερόλεπτα στα ζωντανά διαστήματα, κάθε 30 δευτερόλεπτα στα μέσων όρων, καθώς συχνότερα ερωτήματα επιστρέφουν απλώς κενές απαντήσεις.

Για να αλλάξετε τον ρυθμό ή τη διάρκεια διατήρησης, επεξεργαστείτε αυτές τις σταθερές στην αρχή του μπλοκ δειγματοληψίας στο `admin_server.py` και κάντε επανεκκίνηση:

```
GRAPH_INTERVAL_S    = 2          # δευτερόλεπτα μεταξύ λεπτομερών δειγμάτων
GRAPH_FINE_S        = 3600       # πόσο πίσω φτάνει το λεπτομερές επίπεδο (1 ώρα)
GRAPH_COARSE_EVERY  = 15         # δείγματα ανά συγκεντρωτικό σημείο (15 x 2s = 30s)
GRAPH_COARSE_S      = 24 * 3600 # πόσο πίσω φτάνει το συγκεντρωτικό επίπεδο (24 ώρες)
```

Από πού προέρχονται οι τιμές

Μέγεθος	Πηγή
Συχνότητα CPU	<code>psutil.cpu_freq()</code> , εναλλακτικά <code>/sys/devices/system/cpu/cpu*/cpufreq/</code> . Είναι τα ίδια δεδομένα του πυρήνα που μορφοποιεί το <code>cpufreq-info</code> — το <code>cpufrequtils</code> δεν απαιτείται .
Φορτίο CPU	<code>psutil.cpu_percent()</code>
Θερμοκρασία CPU	<code>psutil.sensors_temperatures()</code> , εναλλακτικά <code>/sys/class/thermal</code> και μετά η εντολή <code>sensors</code>
Χρήστες online	Το ίδιο το endpoint <code>/users</code> του <code>spectrumserver</code> — η έγκυρη λίστα συνεδριών. Η καταμέτρηση συνδέσεων με <code>ss</code> δεν βλέπει τις πραγματικές IP των πελατών πίσω από το <code>proxy.py</code> .

Αν κάποιο μέγεθος δεν είναι διαθέσιμο στο μηχανήμα σας, το αντίστοιχο πλαίσιο το δηλώνει αντί να σχεδιάσει μηδενική γραμμή. Μια αποτυχημένη μέτρηση διακόπτει τη γραμμή, ώστε ένα κενό να μη μοιάζει ποτέ με μετρημένη τιμή.

Γιατί τέσσερα πλαίσια και όχι ένα γράφημα

Τα GHz, τα ποσοστά, οι βαθμοί και ένας αριθμός χρηστών δεν έχουν κοινή κλίμακα. Σε ένα γράφημα θα χρειάζονταν πολλοί άξονες y, και οι διασταυρώσεις που προκύπτουν είναι τεχνούργημα της κλίμακας, όχι στοιχείο για τον διακομιστή. Τέσσερα στοιβαγμένα πλαίσια σε έναν άξονα χρόνου κρατούν τη σύγκριση τίμια. Κάθε πλαίσιο ξεκινά από το μηδέν, ώστε το ύψος της καμπύλης να παραμένει ανάλογο της τιμής.

Το χρώμα προορίζεται για τη θερμοκρασία, όπου το πορτοκαλί και το κόκκινο σημαίνουν τα παραπάνω όρια. Τα υπόλοιπα πλαίσια μοιράζονται ένα χρώμα, επειδή καθένα έχει μία μόνο σειρά που ο τίτλος του ήδη ονομάζει.

Endpoint

`GET /admin/api/graph-stats?since=<epoch>` (απαιτείται σύνδεση) επιστρέφει τα δείγματα νεότερα από το `since`, ώστε η σελίδα να ρωτά σταδιακά αντί να κατεβάζει ξανά ολόκληρο τον buffer σε κάθε κύκλο.

Thermal Guard

Πλήρες εγχειρίδιο: [THERMAL_GUARD.md](#) — οι τέσσερις λειτουργίες και τι πρέπει να κάνει ο διαχειριστής σε καθεμία, λειτουργία χωρίς τον πίνακα, το στάδιο `throttle` χωρίς `root`, δοκιμές και επίλυση προβλημάτων.

Σταματά τον διακομιστή SDR όταν η CPU ζεσταθεί επικίνδυνα και τον ξεκινά ξανά μόλις κρυώσει. Είναι μέρος του πίνακα — δεν εγκαθιστάτε τίποτα επιπλέον — και εμφανίζεται ως κάρτα **THERMAL GUARD** στον Πίνακα ελέγχου, με τις ρυθμίσεις του στις **Ρυθμίσεις**.

Έρχεται **αδρανής**: η λειτουργία ξεκινά στο `log`, οπότε εξ αρχής μόνο καταγράφει τι θα έκανε. Τίποτα δεν αγγίζει τον διακομιστή σας μέχρι να αλλάξετε τη λειτουργία.

Λειτουργεί με οποιονδήποτε τρόπο εκκίνησης/διακοπής

Η προστασία δεν προσπαθεί ποτέ να καταλάβει τι επιβλέπει τον διακομιστή σας. Ενεργεί μέσω των σεναρίων που ορίσατε ως **Default start script** και **Default stop script** στις Ρυθμίσεις, και όσο η CPU είναι υπερθερμασμένη επαναλαμβάνει την εντολή διακοπής σε **κάθε έλεγχο** (κάθε 2 δευτερόλεπτα). Οτιδήποτε επαναφέρει τον διακομιστή — ο `watchdog` μέσα στα `start-*.sh`, μια μονάδα `systemd`, μια εργασία `cron`, μια συνεδρία `tmux` — αναιρείται μέσα σε λίγα δευτερόλεπτα, μέχρι να κρυώσει το μηχανήμα. Αυτό είναι το *κλείδωμα* (lockout).

Αν δεν έχει οριστεί σενάριο διακοπής, στέλνει SIGTERM στη διεργασία που ονομάζεται στο `sdr_process_name` και SIGKILL μετά από περιθώριο 10 δευτερολέπτων.

Τα όρια προκύπτουν από τη δική σας CPU

Αντί για έναν σταθερό αριθμό που θα ήταν λάθος στα περισσότερα μηχανήματα, η προστασία διαβάζει το κρίσιμο όριο που δημοσιεύει ο πυρήνας για τη CPU σας και δουλεύει προς τα πίσω από αυτό:

Η CPU σας αναφέρει	προειδοποίηση	throttle	διακοπή	επάνοδος
Intel, όριο 100 °C	88	92	95	75
AMD Ryzen, όριο 95 °C	83	87	90	70
Raspberry Pi, όριο 85 °C	73	77	80	60

Αυτό έχει σημασία: ένα σταθερό όριο 95 °C είναι σχεδόν φυσιολογικό σε Intel, αλλά σε Ryzen — όπου η Tctl κάθεται στους 95 °C από σχεδιασμό υπό boost — θα σταματούσε τον διακομιστή σε ένα υγιές μηχάνημα, ενώ σε Raspberry Pi, που κάνει throttle στους 80 °C, δεν θα ενεργοποιούνταν ποτέ. Κάθε όριο μπορεί να οριστεί και με το χέρι στις Ρυθμίσεις· αφήστε το πεδίο κενό για αυτόματο.

Για να δείτε πώς φαίνεται το μηχανήμά σας στην προστασία:

```
python3 thermal_guard.py --once
```

```
sensor      : hwmon:coretemp
temperature: 58.0 C
crit        : 100.0 C
mode        : log
thresholds  : warn 88.0  throttle 92.0  stop 95.0  resume 75.0 (C)
cpufreq     : not writable - throttle stage disabled
```

Τα τέσσερα στάδια

Η διακοπή του διακομιστή είναι η έσχατη λύση, όχι το μοναδικό εργαλείο:

1. **warn** — καταγράφεται, τίποτε άλλο.
2. **throttle** — χαμηλώνει κατά 20 % τη μέγιστη συχνότητα της CPU. Απαιτεί οδηγό `cpufreq` και δικαίωμα εγγραφής στο `scaling_max_freq`, δηλαδή συνήθως `root`· όπου αυτό δεν υπάρχει (η συνηθισμένη περίπτωση για έναν πίνακα χωρίς προνόμια, και στα περισσότερα VPS) το στάδιο **απενεργοποιείται μόνο του** και η προστασία περνά από `warn` απευθείας σε `stop`. Το `./setup-cpufreq-perms.sh` δίνει αυτό το δικαίωμα εγγραφής σε μια ομάδα `cpufreq`, ώστε το στάδιο να δουλεύει χωρίς `root` — δείτε [THERMAL_GUARD.md](#).
3. **stop** — εκτελεί το σενάριο διακοπής σας και κρατά τον διακομιστή κάτω (βλ. κλείδωμα παραπάνω).
4. **resume** — αφού η CPU παραμένει κάτω από τη θερμοκρασία επανόδου για `thermal_resume_s` (5 λεπτά από προεπιλογή), αποκαθίσταται η συχνότητα και, στη λειτουργία `stop+restart`, εκτελείται ξανά το σενάριο εκκίνησης.

Κάθε στάδιο πρέπει να **διατηρηθεί** για τον χρόνο του — 30 δευτερόλεπτα για `warn` και `throttle`, 60 για `stop` — ώστε μια στιγμιαία κορύφωση κατά τη μεταγλώττιση να μην το ενεργοποιεί ποτέ.

Οι επαναλαμβανόμενες διακοπές περιορίζονται από το `thermal_max_stops_hour` (2 από προεπιλογή). Μόλις φτάσει το όριο, απενεργοποιείται η **αυτόματη επανεκκίνηση** ενώ η προστασία συνεχίζει να λειτουργεί, ώστε ένα μηχάνημα που υπερθερμαίνεται να μην ταλαντώνεται ανάμεσα σε λειτουργία και διακοπή.

Ποιοι αισθητήρες χρησιμοποιούνται

Μόνο πραγματικοί αισθητήρες πυριτίου της CPU: `coretemp` (Intel), `k10temp` / `zenpower` (AMD), `cpu_thermal` / `soc_thermal` (ARM, Raspberry Pi). Χρησιμοποιείται η υψηλότερη ένδειξη του τσιπ, όχι ο μέσος όρος.

Οι `acpitz` και οι μη επισημασμένες θερμικές ζώνες **αγνοούνται σκόπιμα** — συχνά αναφέρουν θερμοκρασία κουτιού ή πλακέτας δεκάδες βαθμούς κάτω από την πραγματική CPU, οπότε μια προστασία που τους εμπιστευόταν δεν θα ενεργοποιούνταν ποτέ όταν θα χρειαζόταν. Αν το μηχάνημά σας δεν έχει χρησιμοποιήσιμο αισθητήρα (συνηθισμένο σε VPS ή μέσα σε container), η κάρτα το λέει καθαρά (`no trusted CPU sensor`) και η προστασία μένει ανενεργή αντί να προσποιείται ότι σας προστατεύει.

Τι γράφει

Όλα καταλήγουν στο `crash.log`, δηλαδή στην καρτέλα **CRASH** του Log Viewer, ένα συμβάν ανά γραμμή:

```
[THERMAL] warn: 89.0C (warn=88.0 crit=100.0) sustained 31s
[THERMAL] throttle: 93.0C sustained 30s – cpufreq max lowered 20%
[THERMAL] STOPPED server: 96.0C sustained 62s (stop=95.0 crit=100.0) – Started stop-websdr.sh
[THERMAL] lockout: process reappeared at 94.0C – re-stopped [3 time(s) so far]
[THERMAL] recovered: 74.0C held below 75.0 – lockout cleared
[THERMAL] auto-restart: Started start-rx88mk2.sh
```

Η γραμμή του κλειδώματος καταγράφεται το πολύ μία φορά ανά λεπτό, ώστε ένας επιβλέπων που αντιστέκεται συνεχώς να μην πλημμυρίζει το αρχείο.

Δοκιμάστε το πριν το εμπιστευτείτε

Ορίστε το **TEST TEMPERATURE** στις Ρυθμίσεις σε τιμή πάνω από το όριο διακοπής σας. Η προστασία το εκλαμβάνει ως πραγματική ένδειξη, οπότε εκτελείται όλη η διαδρομή — `warn`, `throttle`, `stop`, κλείδωμα, επάνοδος, επανεκκίνηση — όποτε το θελήσετε. Αδειάστε το πεδίο για να επιστρέψετε στον πραγματικό αισθητήρα.

Αν κάνετε αυτή τη δοκιμή με τη λειτουργία στο `stop`, ο διακομιστής όντως σταματά και οι ακροατές σας αποσυνδέονται. Κάντε τη όταν δεν υπάρχει κόσμος, ή αφήστε τη λειτουργία στο `log`, όπου η δοκιμή δείχνει τι θα γινόταν χωρίς να αγγίξει τίποτα.

Προτεινόμενος τρόπος υιοθέτησης

1. Αφήστε τη λειτουργία στο `log` για μια εβδομάδα και συνεχίστε κανονικά.
2. Διαβάστε την καρτέλα CRASH μετά τις βαριές στιγμές — μια πλήρη μεταγλώττιση, ένα ζεστό απόγευμα. Αν δεν εμφανιστεί τίποτα, το μηχάνημά σας δεν πλησίασε ποτέ.
3. Αν οι καταγεγραμμένες θερμοκρασίες φαίνονται σωστές, βάλτε τη λειτουργία σε `stop` (ή `stop+restart` για να επανέρχεται μόνος του).
4. Επιβεβαιώστε το μία φορά με το πεδίο δοκιμαστικής θερμοκρασίας και μετά αφήστε το.

Άρση του κλειδώματος

Το **CLEAR LOCKOUT** στην κάρτα του Πίνακα ελέγχου καθαρίζει το κλείδωμα και το όριο επαναλήψεων. Δεν αποπλίζει την προστασία: αν η CPU είναι ακόμη πολύ ζεστή, ο επόμενος έλεγχος απλώς θα σταματήσει ξανά τον διακομιστή. Χρησιμοποιήστε το αφού έχετε διορθώσει την ψύξη.

Endpoint

Το `GET /admin/api/thermal` (απαιτεί σύνδεση) επιστρέφει όλη την κατάσταση της προστασίας — αισθητήρα, όρια, στάδιο, κλείδωμα, τελευταίο συμβάν. Ένα `POST` με `{"action": "reset"}` κάνει ό,τι και το κουμπί `CLEAR LOCKOUT`.

Χωρίς τον πίνακα διαχείρισης

Το `thermal_guard.py` χρησιμοποιεί μόνο τη `standard library` και τρέχει και μόνο του, διαβάζοντας το ίδιο `admin_config.json`:

```
python3 thermal_guard.py --once           # εμφάνιση αισθητήρα, κρίσιμου ορίου και ορίων
python3 thermal_guard.py                 # εκτέλεση ως υπηρεσία (μόνο καταγραφή μέχρι να
οπλιστεί)
python3 thermal_guard.py --mode stop+restart # όπλισμα χωρίς κανένα αρχείο ρυθμίσεων
python3 thermal_guard.py --config /path/to/other.json
python3 thermal_guard.py --help
```

Το `--mode` υπερισχύει του `thermal_mode` του αρχείου ρυθμίσεων, οπότε ένα μηχάνημα χωρίς πίνακα μπορεί να οπλιστεί κατευθείαν από τη μονάδα `systemd`, χωρίς καθόλου `JSON`.

Spot Reporting (Autorun FT8/FT4/JS8/WSPR)

Η καρτέλα **Spot Reporting** ελέγχει τον δαίμονα `autorun` (`autorun/index.js`) — μια διεργασία `Node.js` που αποκωδικοποιεί `FT8/FT4/JS8/WSPR` από την πλευρά του διακομιστή, απευθείας από τον δέκτη, και ανεβάζει τα `spot` στα δίκτυα αναφορών:

- **FT8 / FT4 / JS8 → PSK Reporter** (IPFIX/UDP)
- **WSPR → wsprnet.org** (HTTP)

Το JS8 αναφέρεται μόνο σε ταχύτητα Normal — τον κύκλο κλήσης των 15 δευτ., όπου γίνονται τα `heartbeat` και τα `CQ`. Τα `spot` προέρχονται από `heartbeat`, `compound` πλαίσια και κατευθυνόμενα μηνύματα· ομαδικοί προορισμοί όπως το `@ALLCALL` και διακριτικά που δεν μπόρεσε να αναλύσει ο αποκωδικοποιητής (`<...>`) δεν αναφέρονται ποτέ. Το `JS8` μοιράζεται την ουρά του `PSK Reporter` με τα `FT8` και `FT4` και μετριέται ξεχωριστά, όπως ακριβώς και εκείνα.

Οι αναφορές είναι ΑΠΕΝΕΡΓΟΠΟΙΗΜΕΝΕΣ εξ ορισμού. Τίποτα δεν ανεβαίνει μέχρι να ενεργοποιήσετε προορισμό και να πατήσετε **Start**. Η αποκωδικοποίηση και το ανέβασμα είναι ανεξάρτητα: ο δαίμονας μπορεί να αποκωδικοποιεί και να καταγράφει με τις αναφορές κλειστές, ώστε να επαληθεύσετε τη δραστηριότητα πριν δημοσιοποιηθεί οτιδήποτε.

⚠ Τα `spot` ανεβαίνουν σε δημόσια δίκτυα με **το δικό σας διακριτικό**. Ενεργοποιήστε μόνο μπάντες/τρόπους που ο δέκτης σας ακούει πραγματικά και ορίστε το σωστό `locator`.

Προϋποθέσεις

Ο δαίμονας `autorun` χρειάζεται `Node.js 22+`, τα πακέτα `npm ws + cbor-x` (επιλύονται μέσω του συμβολικού συνδέσμου `autorun/node_modules → frontend/node_modules`) και `util-linux (taskset)`. Τα προγράμματα εγκατάστασης τα ρυθμίζουν όλα αυτόματα — δείτε την [ενότητα Autorun Spot Reporter του INSTALLATION.md](#). Αν αποτύχει το **Start**, η συνήθης αιτία είναι ο σύνδεσμος ή το `taskset` (δείτε την Αντιμετώπιση προβλημάτων παρακάτω).

Χρήση της καρτέλας

1. **Ταυτότητα** — διακριτικό και locator. Προσυμπληρώνονται από το `frontend/site_information.json` (`siteSysop` / `siteGridSquare`, κομμένο στους 6 χαρακτήρες)· αντικαταστήστε τα εδώ αν χρειάζεται.
2. **Πίνακας μπάντας × τρόπου** — επιλέξτε τους συνδυασμούς προς αποκωδικοποίηση. Οι γραμμές είναι μπάντες, οι στήλες FT8 / FT4 / JS8 / WSPR· τα μη υποστηριζόμενα κελιά είναι απενεργοποιημένα. Το WSPR καλύπτει επιπλέον τις μπάντες LF/MF (2200 m, 630 m) και ένα επιπλέον κανάλι 80 m EU.
3. **Προορισμοί** — ενεργοποιήστε **PSK Reporter** ή/και **wspnnet** (και τα δύο κλειστά εξ ορισμού).
4. **Max slots** — όριο ασφαλείας για το πόσες θέσεις μπάντας/τρόπου μπορούν να τρέχουν ταυτόχρονα.
5. **Start / Stop** — ξεκινά/τερματίζει τον δαίμονα (καρφίτσωμένο στους ανώτερους πυρήνες CPU με `taskset`, που επιλέγονται αυτόματα για το μηχάνημα — δείτε *Απαιτήσεις πόρων και περιορισμοί παρακάτω*). Τα **Save / Reload** αποθηκεύουν και ξαναδιαβάζουν τις ρυθμίσεις· το **Free All** αδειάζει όλες τις θέσεις.

Η κάρτα κατάστασης ανανεώνεται κάθε 5 s και δείχνει την κατάσταση εκτέλεσης, τους μετρητές αποκωδικοποίησης/αποστολής και την ώρα της τελευταίας αποστολής. Ένα σήμα **REPORTING** εμφανίζεται στον κύριο καταρράκτη όσο οι αναφορές είναι ενεργές.

Το «0 sent» τα πρώτα λεπτά είναι φυσιολογικό. Το PSK Reporter ανεβάζει ομαδικά κάθε **5 λεπτά** και το wspnnet κάθε **2 λεπτά** — τα spot περιμένουν στην ουρά. Η κάρτα κατάστασης δείχνει τον αριθμό των εκκρεμών και τον ρυθμό.

Αλλαγή μπαντών ή τρόπων ενώ τρέχει

Οι αλλαγές δεν ισχύουν μέχρι να επανεκκινήσει ο daemon. Το `autorun/index.js` διαβάζει το `autorun.json` μία μόνο φορά, κατά την εκκίνηση. Δεν υπάρχει παρακολούθηση αρχείου ούτε σήμα επαναφόρτωσης — τα μόνα σήματα που χειρίζεται είναι SIGINT και SIGTERM, που σημαίνουν και τα δύο τερματισμό. Ένας daemon που τρέχει συνεχίζει επομένως να αποκωδικοποιεί τα slot με τα οποία ξεκίνησε, ό,τι κι αν αποθηκεύσετε μετά.

Για να εφαρμοστεί μια αλλαγή:

1. Επιλέξτε / αποεπιλέξτε μπάντες και τρόπους
2. **SAVE CONFIG**
3. **STOP**
4. **START**

Ή απλώς **STOP** → αλλαγή επιλογών → **START**, καθώς το START αποθηκεύει τη διαμόρφωση πριν ξεκινήσει.

⚠ **Πατώντας START χωρίς να κάνετε πρώτα STOP, η αλλαγή δεν εφαρμόζεται.** Το START αποθηκεύει τη διαμόρφωση και μετά το αίτημα εκκίνησης απαντά `already running` και δεν επανεκκινεί τίποτα. Βλέπετε ένα μήνυμα «αποθηκεύτηκε» δίπλα σε σφάλμα «already running», ενώ ο daemon συνεχίζει με τις **παλιές** μπάντες — μοιάζει εύκολα με επιτυχία. Πάντα STOP πρώτα.

Τι κάνει μια επανεκκίνηση στους μετρητές:

- Τα **συνολικά νούμερα δίπλα στα πλαίσια επιλογής διατηρούνται** — βρίσκονται στο `autorun-totals.json`, που δεν διαγράφεται κατά τον τερματισμό.
- Τα **πλακίδια ανά αποκωδικοποιητή μηδενίζονται**, αφορούν κάθε εκτέλεση.
- **Δεν χάνεται τίποτα από την ουρά**: ο τερματισμός στέλνει πρώτα τα εκκρεμή spot.
- Μια **νέα** μπάντα ξεκινά από 0· μια μπάντα που είχατε **ξαναχρησιμοποιήσει** συνεχίζει το προηγούμενο σύνολό της.

- Μια μπάντα που **αποεπιλέγετε κρατά τον αριθμό της** δίπλα στο άδαιο πλέον πλαίσιο· το ιστορικό δεν διαγράφεται.

Η αποκωδικοποίηση σταματά μόνο για τα λίγα δευτερόλεπτα μεταξύ STOP και START.

Μετρητές spot

Εμφανίζονται δύο διαφορετικές μετρήσεις, που απαντούν σε διαφορετικά ερωτήματα.

SPOTS UPLOADED PER DECODER — ένα πλακίδιο ανά αποκωδικοποιητή (FT8 / FT4 / JS8 / WSPR) με τα spot που ανέβηκαν **από την τελευταία εκκίνηση του daemon**, και από κάτω το πλήθος αποκωδικοποιήσεων και πόσα εκκρεμούν. Το Stop/Start τα μηδενίζει. Ένας αποκωδικοποιητής του οποίου ο προορισμός είναι απενεργοποιημένος δείχνει τις αποκωδικοποιήσεις του και **reporting off** αντί για σκέτο 0, γιατί εκεί το μηδέν είναι ρύθμιση, όχι σφάλμα.

Ο αριθμός δίπλα σε κάθε πλαίσιο επιλογής στο BANDS & MODES είναι τα spot που έχουν ανέβει **συνολικά** για αυτόν τον συνδυασμό μπάντας και τρόπου. Κρατούνται στο `autorun-totals.json` και επιβιώνουν των επανεκκινήσεων. Μια τελεία μπροστά (·123) σε πορτοκαλί σημαίνει ότι το slot έχει αποκωδικοποιήσει αλλά δεν έχει ανεβάσει ακόμη. Αυτό είναι φυσιολογικό ανάμεσα στις αποστολές — το PSK Reporter ανεβάζει κάθε 5 λεπτά και το wsprnet κάθε 2 — οπότε οι μετρητές FT8/FT4 μένουν πορτοκαλί τα πρώτα λεπτά μετά την εκκίνηση. Παραμένει πορτοκαλί και όταν ο προορισμός του αποκωδικοποιητή είναι κλειστός. Ένα σκέτο 0 χωρίς τελεία σημαίνει ότι το slot είναι ενεργό αλλά δεν έχει αποκωδικοποιήσει ακόμη τίποτα — το WSPR μένει στο μηδέν για αρκετά λεπτά μετά την εκκίνηση, επειδή δουλεύει σε κύκλο 2 λεπτών, ενώ το FT8 μετρά ήδη σε εκατοντάδες. Ένα slot που ο daemon δεν έχει τρέξει ποτέ δεν δείχνει τίποτα. Περνώντας τον δείκτη βλέπετε ανεβάσματα, αποκωδικοποιήσεις και την ώρα του τελευταίου ανεβάσματος.

Τα FT8 και FT4 μοιράζονται μία ουρά αποστολής προς το PSK Reporter, γι' αυτό ο daemon χρεώνει κάθε spot στον δικό του τρόπο και μπάντα τη στιγμή της αποστολής· ο διαχωρισμός δεν εκτιμάται εκ των υστέρων από τα σύνολα.

Μηδενισμός των μετρητών. Το **FREE ALL SLOTS** αποεπιλέγει όλες τις μπάντες και τους τρόπους, κλείνει και τους δύο προορισμούς, σταματά τον daemon και **σβήνει τους συνολικούς μετρητές** — μετά από αυτό δεν εμφανίζεται κανένας αριθμός δίπλα σε κανένα πλαίσιο επιλογής. Είναι ο μόνος τρόπος να μηδενιστούν και δεν αναιρείται. Το κουμπί σταματά τον daemon και περιμένει να τερματίσει πριν διαγράψει το `autorun-totals.json`, επειδή ο daemon ξαναγράφει αυτό το αρχείο κατά τον τερματισμό· η διαγραφή με ζωντανό daemon απλώς θα επανέφερε τους αριθμούς.

Απαιτήσεις πόρων και περιορισμοί

Ο δαίμονας είναι σκόπιμα ελαφρύς και τρέχει σε υλικό χαμηλών πόρων (ένας i5 τεσσάρων πυρήνων είναι αρκετός), υπάρχουν όμως πραγματικά όρια που πρέπει να γνωρίζετε.

Το καρφίτσωμα CPU του autorun λαμβάνει υπόψη τη διαμόρφωση και είναι αυτόματο. Ο δαίμονας **autorun** εκκινείται πάντα από τον πίνακα διαχείρισης (`admin_server.py`), οπότε το καρφίτσωμά του ορίζεται σε κάθε εγκατάσταση χωρίς ρύθμιση: εξάγει εύρος πυρήνων `taskset` από τον αριθμό των CPU, δεσμεύοντας μερικούς από τους ανώτερους πυρήνες για αποκωδικοποίηση, ή τρέχει χωρίς καρφίτσωμα σε ≤ 4 πυρήνες.

Το καρφίτσωμα του spectrumserver εξαρτάται από τον δικό σας τρόπο εκκίνησης. Ο τρόπος με τον οποίο εκκινείτε τον spectrumserver διαφέρει ανά εγκατάσταση (τύπος SDR, εργαλεία, προσωπικά σενάρια), οπότε το έγγραφο αυτό δεν προϋποθέτει κάποιον συγκεκριμένο εκκινητή. Ο πίνακας διαχείρισης δεν εκκινεί ούτε καρφιτσώνει τον spectrumserver — αυτό εξαρτάται εξ ολοκλήρου από την εντολή ή την υπηρεσία που χρησιμοποιείτε. Αν θέλετε να κρατήσετε τον spectrumserver μακριά από τους πυρήνες που χρησιμοποιεί ο δαίμονας autorun, καρφιτρώστε τον μόνοι σας με `taskset` στη δική σας εντολή εκκίνησης (δείτε την ενότητα αντικατάστασης παρακάτω). Αν δεν το κάνετε, τρέχει

απλώς χωρίς καρφίτσωμα και ο χρονοπρογραμματιστής του λειτουργικού τον εξισορροπεί — σωστό και ασφαλές, απλώς χάνετε τον σκόπιμο διαχωρισμό πυρήνων.

Για αναφορά, ο δαίμονας `autorun` δεσμεύει αυτούς τους ανώτερους πυρήνες (αν λοιπόν καρφίτσώσετε τον `spectrumserver`, κρατήστε τον στους χαμηλότερους για να μην υπάρχει επικάλυψη):

Λογικές CPU	Χρησιμοποιεί ο δαίμονας <code>autorun</code>	Αφήστε για τον <code>spectrumserver</code>
≤ 4	χωρίς καρφίτσωμα	χωρίς καρφίτσωμα
6	5	0-4
8	6-7	0-5
12 (π.χ. υβριδικό 8P+4E)	8-11	0-7
16	12-15	0-11

Σε μηχάνημα **4 πυρήνων (ή λιγότερων)** δεν υπάρχει τίποτα να διαχωριστεί, οπότε και ο δαίμονας `autorun` τρέχει *χωρίς καρφίτσωμα* και μοιράζεται όλους τους πυρήνες — σωστό και ασφαλές, αλλά η FFT του SDR και οι εκρήξεις αποκωδικοποίησης ανταγωνίζονται τους ίδιους πυρήνες.

Γνωστοί περιορισμοί:

- 1. Το πραγματικό σημείο συμφόρησης είναι ο `spectrumserver` + το εύρος ζώνης του SDR, όχι ο δαίμονας.** Ένα RX888 στα 30 MHz χρειάζεται OpenCL/GPU· μια CPU με λίγους πυρήνες χωρίς ικανή GPU δεν μπορεί να συντηρήσει αυτή τη FFT. Συνδυάστε υλικό χαμηλών πόρων με στενότερο SDR ($RSP1A \approx 10 \text{ MHz}$, $RTL\text{-}SDR \approx 2.4 \text{ MHz}$).
- 2. Το WSPR είναι το βαρύτερο σε CPU.** Ο αποκωδικοποιητής Fano είναι μεταφορά σε JS (~30 s ανά μπάντα, μονονηματικός). Η δεξαμενή 4 workers εκτελεί 4 αποκωδικοποιήσεις παράλληλα· η ενεργοποίηση **περισσότερων από ~4 μπαντών WSPR** ταυτόχρονα μπορεί να δημιουργήσει ουρά πέρα από τη θυρίδα των 120 s, ιδίως όσο το SDR ανταγωνίζεται για πυρήνες. Οι αποκωδικοποιήσεις FT8/FT4 είναι συγκριτικά φθηνές.
- 3. Κλιμάκωση αριθμού θέσεων.** Τα ~2-2,5 πυρήνες / 600-700 MB αφορούν τις πλήρεις ~28 θέσεις σε μηχάνημα 8 πυρήνων. Σε 4 κοινόχρηστους πυρήνες περιορίζεστε σε λίγες μπάντες (οδηγία: $\leq 6 \text{ FT8/FT4} + \leq 3 \text{ WSPR}$) και παρακολουθείτε το στατιστικό `queued` της δεξαμενής.
- 4. Το καρφίτσωμα προϋποθέτει υβριδική τοπολογία Intel** (χαμηλότεροι πυρήνες = ταχύτεροι P-cores). Σε AMD ή σε CPU με εναλλασσόμενη αρίθμηση SMT ο αυτόματος διαχωρισμός παραμένει έγκυρος (χωρίς επικάλυψη, χωρίς κατάρρευση), αλλά το «οι χαμηλότεροι πυρήνες είναι ταχύτεροι» μπορεί να μην ισχύει κυριολεκτικά· σε επεξεργαστή 4C/8T με hyperthreading οι ανώτεροι πυρήνες είναι αδέρφια SMT — περιορισμένοι, όχι πλήρως απομονωμένοι. Όπου το αυτόματο εύρος δεν είναι ιδανικό, χρησιμοποιήστε τη **χειροκίνητη αντικατάσταση** παρακάτω.
- 5. Η κάλυψη μπαντών περιορίζεται από τον δέκτη.** Η διαμόρφωση του RX888 ($\text{sps}=60000000 \rightarrow$ Nyquist στα 30 MHz) δεν φτάνει στα 6 m και άνω· ο πίνακας μπαντών είναι 160 m-10 m. Άλλα SDR καλύπτουν ό,τι επιτρέπει το παράθυρο συντονισμού τους.
- 6. Ρυθμός αναφορών.** Το PSK Reporter στέλνει κάθε 5 λεπτά, το `wsprnet` κάθε 2 λεπτά — το «0 sent» τα πρώτα λεπτά είναι φυσιολογικό, όχι σφάλμα.

Χειροκίνητη αντικατάσταση καρφίτσώματος CPU (προχωρημένα)

Ο αυτόματος διαχωρισμός πυρήνων είναι σωστός για τα περισσότερα μηχανήματα, αλλά μπορείτε να επιβάλετε συγκεκριμένο καρφίτσωμα όπου δεν είναι (AMD CCX/CCD, ARM big.LITTLE, εναλλασσόμενο SMT). Δύο ανεξάρτητες αντικαταστάσεις, καθεμία δέχεται λίστα πυρήνων τύπου `taskset -c (2-3, 0,2,4, 0-2,5)` ή `none/off/unpinned` για απενεργοποίηση. Μια άκυρη τιμή αγνοείται και εκτελείται η αυτόματη εξαγωγή — ένα τυπογραφικό λάθος δεν μπορεί ποτέ να σταματήσει την εκκίνηση.

Δαίμονας autorun — μεταβλητή περιβάλλοντος `AUTORUN_CORES` ή πεδίο `"cores"` στο `autorun.json` (υπερισχύει το περιβάλλον). Το πεδίο `"cores"` διατηρείται στο **Save** του πίνακα διαχείρισης, οπότε μια χειροκίνητη επεξεργασία παραμένει. Δύο τρόποι να το ορίσετε:

Επιλογή Α — `autorun.json` (μόνιμο, συνιστάται). Προσθέστε μια γραμμή `"cores"` στη ρύθμιση στη ρίζα του αποθετηρίου και έπειτα κάντε **Stop → Start** στην καρτέλα Spot Reporting:

```
// autorun.json
{ "identity": { "callsign": "SV1BTL", "grid": "KM17VX" },
  "reporting": { "pskreporter": true, "wsprnet": false },
  "slots": [ /* ... */ ],
  "cores": "2-3" }           // or "none" to run unpinned
```

Επιλογή Β — μεταβλητή περιβάλλοντος (μία φορά). Ο δαίμονας εκκινείται από τον πίνακα διαχείρισης, οπότε η μεταβλητή πρέπει να βρίσκεται στο περιβάλλον **του πίνακα** — ορίστε τη και επανεκκινήστε τον πίνακα (το περιβάλλον υπερσχύει της τιμής στο `autorun.json`):

```
# Μέθοδος Α — η μεταβλητή πρέπει να μπει στη διεργασία που ξεκινά:
./manage_admin.sh stop && AUTORUN_CORES=2-3 ./manage_admin.sh start

# Μέθοδος Β — το systemd αγνοεί μεταβλητή που δίνεται στη γραμμή εντολών
# του systemctl, οπότε βάλτε την στη μονάδα:
sudo systemctl set-environment AUTORUN_CORES=2-3
sudo systemctl restart phantomsdr-admin
```

spectrumserver — καρφίτσωστε τον μόνιμο σας προσθέτοντας `taskset -c <πυρήνες>` μπροστά από όποια εντολή χρησιμοποιείτε για να εκκινήσετε τον διακομιστή. Αυτό λειτουργεί με κάθε μέθοδο εκκίνησης (απευθείας εκτέλεση, αλυσίδα SDR, μονάδα υπηρεσίας κ.λπ.):

```
# pin the server to cores 0-3, direct launch:
taskset -c 0-3 ./build/spectrumserver --config <your-config>.toml && <your-input>
# or in an SDR pipeline:
<your-sdr-source> | taskset -c 0-3 ./build/spectrumserver --config <your-config>.toml
```

⚠ Φροντίστε να επιβιώνει των επανεκκινήσεων. Ένα ενσωματωμένο `taskset` ισχύει μόνο για εκείνη τη μία εκκίνηση. Αν κάτι επανεκκινεί αυτόματα τον διακομιστή (watchdog, υπηρεσία systemd, εγγραφή cron/@reboot ...), προσθέστε το πρόθεμα `taskset` μέσα στο σενάριο ή τη μονάδα που τον εκκινεί πραγματικά — αλλιώς η επανεκκίνηση χάνει το καρφίτσωμα.

Επαληθεύστε ότι το καρφίτσωμα εφαρμόστηκε — εμφανίστε την πραγματική συγγένεια CPU των τρεχουσών διεργασιών:

```
taskset -cp "$(pgrep -f 'autorun/index.js')" # autorun daemon
taskset -cp "$(pgrep -x spectrumserver)"    # spectrumserver
```

Αποδεκτές τιμές (και για τις δύο αντικαταστάσεις): λίστα `taskset -c (2-3, 0,2,4, 0-2,5)` ή `none/off/unpinned` για κανένα καρφίτσωμα. Οτιδήποτε κακοδιατυπωμένο αγνοείται και εκτελείται η αυτόματη εξαγωγή, οπότε ένα τυπογραφικό λάθος δεν μπορεί ποτέ να μπλοκάρει την εκκίνηση.

Σε Raspberry Pi / οποιοδήποτε μηχάνημα ≤ 4 πυρήνων κανονικά δεν χρειάζεστε καμία από τις δύο — η αυτόματη διαδρομή ήδη εκτελεί και τις δύο διεργασίες χωρίς καρφίτσωμα, που είναι η σωστή επιλογή σε μικρή, ομοιογενή CPU. Η αντικατάσταση αφορά μεγαλύτερα μηχανήματα που δεν είναι υβριδικά Intel.

Αρχεία και σημεία πρόσβασης

Στοιχείο	Σκοπός
<code>autorun.json</code>	Αποθηκευμένη ρύθμιση (ταυτότητα, θέσεις, προορισμοί, μέγιστες θέσεις, προαιρετικό <code>cores</code>). Γράφεται από την καρτέλα· αγνοείται από το <code>git</code> .
<code>autorun-status.json</code>	Ζωντανή κατάσταση που διαβάζει η κάρτα κατάστασης (<code>pid</code> , μετρητές, τελευταία αποστολή). Γράφεται κάθε 15 s· αφαιρείται στη διακοπή.
<code>frontend/dist/autorun-active.json</code>	Δημόσια δεδομένα σήματος, διαθέσιμα στο <code>/autorun-active.json</code> . Κενό όταν οι αναφορές είναι κλειστές.
<code>autorun.log</code>	Τυπική έξοδος/σφαλμάτων του δαίμονα.
<code>GET/POST /admin/api/autorun/config</code>	Ανάγνωση/εγγραφή του <code>autorun.json</code> (επικυρώνει διακριτικό, <code>locator</code> , συνδυασμούς μπάντας/τρόπου, όριο θέσεων).
<code>GET /admin/api/autorun/status</code>	Κατάσταση εκτέλεσης (<code>pgrep</code>) + <code>autorun-status.json</code> .
<code>POST /admin/api/autorun/start / stop</code>	Εκκίνηση (<code>taskset -c <auto> node autorun/index.js</code>) / <code>SIGTERM</code> . Το εύρος πυρήνων εξάγεται από τη CPU (δείτε παραπάνω).

Σημείωση: μετά από αναβάθμιση του `admin_server.py` πρέπει να κάνετε [Επανεκκίνηση του πίνακα](#) για να φορτωθούν οι νέες διαδρομές `autorun` ή ένας ενημερωμένος πίνακας μπάντας/τρόπου. Ο δαίμονας τρέχει ως ξεχωριστή διεργασία, οπότε επιβιώνει των επανεκκινήσεων του πίνακα διαχείρισης.

Αντιμετώπιση προβλημάτων

Πρόβλημα	Λύση
Ο πίνακας διαχείρισης δεν ξεκινά	Ελέγξτε το <code>admin.log</code> — συνήθως λείπει κάποιο πακέτο Python
Ο διαμεσολαβητής δεν ξεκινά	Εκτελέστε <code>python3 -c "import aiohttp"</code> — αν αποτύχει: <code>pip3 install aiohttp --break-system-packages</code>
Το <code>proxy.log</code> είναι κενό	Ελέγξτε πρώτα ότι ο διαμεσολαβητής όντως τρέχει (<code>./manage_admin.sh status</code>). Αν ναι, έχετε παλιά έκδοση: ο εκκινητής πρέπει να χρησιμοποιεί <code>python3 -u</code> (χωρίς ενταμίευση — αλλιώς το πλαίσιο εκκίνησης δεν φεύγει ποτέ από τον ενταμιευτή των 8 KB) και η <code>main()</code> του <code>proxy.py</code> πρέπει να καλεί <code>logging.basicConfig()</code> (αλλιώς ο καταγραφέας πρόσβασης του <code>aiohttp</code> δεν έχει χειριστή και κάθε γραμμή αιτήματος χάνεται, επειδή το <code>web.AppRunner</code> — σε αντίθεση με το <code>web.run_app</code> — δεν ρυθμίζει την καταγραφή).
Ο διαμεσολαβητής ξεκινά αλλά ο καταρράκτης είναι κενός	Ελέγξτε το <code>sdr_host</code> στο <code>admin_config.json</code> — κανονικά <code>127.0.0.1</code> . Αν έχει ακόμη παλιά τοπική IP από προηγούμενη έκδοση, αλλάξτε την και κάντε Επανεκκίνηση του πίνακα
Το κουμπί Kick λέει «no connections»	Εκτελέστε <code>getcap \$(which ss)</code> — αν λείπει το <code>cap_net_admin</code> , εκτελέστε <code>sudo setcap cap_net_admin+ep \$(which ss)</code>
Η κατάσταση δείχνει πάντα OFFLINE	Πηγαίνετε σε Settings → SDR Process Name — ορίστε το ακριβές όνομα που δείχνει η <code>ps -eo comm,args \ grep -v grep</code>
Η σελίδα Users δεν δείχνει δεδομένα	Επαληθεύστε ότι λειτουργεί το σημείο πρόσβασης: <code>curl http://127.0.0.1:<public_port>/users</code>
Εξωτερικός περιηγητής εμφανίζει NetworkError	Ελέγξτε ότι ο διαμεσολαβητής τρέχει: <code>./manage_admin.sh status</code>
Ο διαμεσολαβητής χάλασε μετά από αλλαγή δικτύου	Αφορά μόνο την περίπτωση που το <code>sdr_host</code> έχει ακόμη τοπική IP — βάλτε το σε <code>127.0.0.1</code> και κάντε Επανεκκίνηση του πίνακα
Το κουμπί διαγραφής συνομιλίας δεν έχει αποτέλεσμα	Ελέγξτε το δικαίωμα εγγραφής στο αρχείο συνομιλίας: <code>ls -l ~/PhantomSDR-Plus/chat.jsonl</code>
Το μήνυμα καταρράκτη δεν εμφανίζεται	Επιβεβαιώστε ότι ο διαμεσολαβητής τρέχει (<code>./manage_admin.sh status</code>) και ότι το frontend είναι σε πρόσφατη έκδοση που περιλαμβάνει τον αποδότη επικάλυψης
Το μήνυμα καταρράκτη χάθηκε μετά από επανεκκίνηση	Αναμενόμενο — η κατάσταση του μηνύματος είναι μόνο στη μνήμη· στείλτε το ξανά μετά την επανεκκίνηση του πίνακα
Το «Start» του Spot Reporting αποτυγχάνει / ο δαίμονας τερματίζεται αμέσως	Ελέγξτε το <code>autorun.log</code> . Συνήθως λείπει ο σύνδεσμος <code>autorun/node_modules</code> (<code>ln -sf ../frontend/node_modules autorun/node_modules</code>) ή δεν είναι εγκατεστημένο το <code>taskset</code> (<code>sudo apt install -y util-linux</code>).
Spot Reporting: ο δαίμονας τρέχει αλλά τα <code>decodes</code> μένουν 0 και το <code>autorun.log</code> δείχνει <code>504 / tap closed 1006</code>	Η λήψη ήχου δεν φτάνει στον <code>spectrumserver</code> . Ο δαίμονας εντοπίζει αυτόματα τη θύρα από τη ρύθμιση του τρέχοντος διακομιστή· η γραμμή <code>[autorun] tap backend: HOST:PORT</code> πρέπει να ταιριάζει με τη <code>[server] port</code> . Αν είναι λάθος (ή ο διακομιστής δεν έτρεχε στην εκκίνηση), καθορίστε την στο <code>autorun.json</code> : <code>"server": { "host": "127.0.0.1", "port": 9002 }</code> , και έπειτα Stop→Start.
Το Spot Reporting δείχνει «0 sent»	Φυσιολογικό τα πρώτα λεπτά — το PSK Reporter στέλνει κάθε 5 λεπτά, το <code>wspnnet</code> κάθε 2. Επιβεβαιώστε ότι είναι ενεργός κάποιος προορισμός και ότι ο δαίμονας τρέχει.
Νέες μπάντες/τρόποι δεν εμφανίζονται στον πίνακα	Επανεκκίνηση του πίνακα — ο πίνακας φορτώνεται στην εκκίνηση.

Πρόβλημα	Λύση
Το <code>autorun.log</code> δείχνει <code>MODULE_TYPELESS_PACKAGE_JSON / «Reparsing as ES module ... performance overhead»</code>	Καλλωπιστική προειδοποίηση (η αποκωδικοποίηση λειτουργεί). Από το <code>frontend/src/modules/package.json</code> λείπει το <code>"type": "module"</code> · προσθέστε τη γραμμή κοντά στην αρχή και έπειτα <code>Stop→Start</code> . Δεν χρειάζεται νέα μεταγλώττιση του frontend — ο δαίμονας εισάγει απευθείας το αρχείο πηγής.
Το «Start» του Spot Reporting αποτυγχάνει με <code>taskset: ... Invalid argument</code> σε παλιό/μικρό PC	Δεν θα έπρεπε να συμβαίνει με τον εκκινητή που λαμβάνει υπόψη τη διαμόρφωση — το εύρος πυρήνων εξάγεται από τον αριθμό CPU και δεν καρφιστώνει σε ≤ 4 πυρήνες. Αν εμφανιστεί, το <code>admin_server.py</code> σας προηγείται αυτής της αλλαγής· ενημερώστε το (<code>_autorun_taskset_prefix</code>) και κάντε Επανεκκίνηση του πίνακα .
Το σήμα <code>REPORTING</code> δεν εμφανίζεται στον καταρράκτη	Οι αναφορές πρέπει να είναι ενεργές με τουλάχιστον μία θέση· το σήμα διαβάζει το <code>/autorun-active.json</code> . Μεταγλωττίστε ξανά το frontend αν το <code>dist/index.html</code> προηγείται του σήματος.

Thermal Guard

THERMAL_GUARD.md

Thermal Guard — Εγχειρίδιο διαχειριστή

Προστασία από υπερθέρμανση της CPU για το PhantomSDR-Plus.

Ένα WebSDR είναι από τα λίγα πράγματα που κρατούν όλους τους πυρήνες απασχολημένους, όλη μέρα, για πάντα. Αν βουλώσει ένας ανεμιστήρας, κολλήσει μια αντλία ή ζεστάνει ο χώρος τον Αύγουστο, δεν σας το λέει κανείς — ο διακομιστής απλώς συνεχίζει να αποκωδικοποιεί μέχρι η CPU να ψήνεται. Το thermal guard είναι το κομμάτι που το αντιλαμβάνεται και τραβάει την πρίζα για λογαριασμό σας.

Έρχεται **ανενεργό**: εξ αρχής μόνο γράφει σε ένα αρχείο καταγραφής. Τίποτα δεν προστατεύεται μέχρι να επιλέξετε λειτουργία. Αυτό το εγχειρίδιο αφορά ακριβώς αυτή την επιλογή.

Βιάζεστε; Τρέξτε `python3 thermal_guard.py --once`, διαβάστε τα όρια που τυπώνει και βάλτε το `thermal_mode` σε `stop+restart` από τον πίνακα διαχείρισης. Αυτή είναι όλη η δουλειά. Όλα τα υπόλοιπα παρακάτω είναι το γιατί.

Περιεχόμενα

1. Τι κάνει
2. Γρήγορη εκκίνηση
3. Πώς αποφασίζει
4. Οι τέσσερις λειτουργίες — τι κάνει η καθεμία και τι πρέπει να κάνετε εσείς
5. Επιλογή λειτουργίας
6. Αναφορά ρυθμίσεων
7. Λειτουργία χωρίς τον πίνακα διαχείρισης
8. Ενεργοποίηση του σταδίου throttle χωρίς root
9. Λειτουργία χωρίς επιτήρηση — όταν δεν βλέπει κανείς
10. Δοκιμή πριν του εμπιστευτείτε το μηχάνημα
11. Διαβάζοντας το αρχείο καταγραφής
12. Lockout και όριο συχνότητας
13. Επίλυση προβλημάτων
14. Απενεργοποίηση ή αφαίρεση

1. Τι κάνει

Κάθε 2 δευτερόλεπτα το guard διαβάζει τη θερμοκρασία του πυριτίου της CPU και τη συγκρίνει με τέσσερα όρια. Όταν η θερμοκρασία παραμείνει πάνω από κάποιο από αυτά αρκετή ώρα, ανεβαίνει μια σκάλα:

κανονικά → warn → throttle → stop → (ψύξη) → restart
 γραμμή στο χαμήλωμα εκτέλεση ο διακομιστής εκτέλεση
 αρχείο συχνότητας stop script μένει κάτω start script

Δύο σχεδιαστικές αποφάσεις το κάνουν να δουλεύει σε *οποιοδήποτε* μηχάνημα — αξίζει να τις καταλάβετε πριν βασιστείτε σε αυτό:

Δεν ρωτά ποτέ πώς τρέχετε τον διακομιστή σας. systemd, βρόχος watchdog, cron, tmux, ένα σκέτο κέλυφος — το guard δεν το ξέρει και δεν το χρειάζεται. Αντ' αυτού, όσο το μηχάνημα είναι υπερθερμασμένο κρατά ένα **lockout**: επαναλαμβάνει την εντολή διακοπής σε *κάθε* κύκλο. Αν ο watchdog σας ξαναξεκινήσει τον διακομιστή, το guard τον σταματά ξανά μέσα σε δύο δευτερόλεπτα. Ό,τι κι αν προσπαθεί να τον κρατήσει ζωντανό χάνει, μέχρι να κρυώσει η CPU. Αυτό ακριβώς το κάνει καθολικό, και είναι δοκιμασμένο απέναντι σε πραγματικό watchdog.

Δεν καρφώνει ποτέ μια σταθερή θερμοκρασία. Το «σταμάτα στους 95 °C» είναι κακός κανόνας: σε έναν Intel με Tjmax 100 αυτό είναι σχεδόν κανονικό φορτίο, σε έναν Ryzen είναι *κάτω* από εκεί που κάθεται η Tctl υπό boost εκ σχεδιασμού, και σε ένα Raspberry Pi δεν φτάνει ποτέ, γιατί το SoC περιορίζεται μόνο του στους 80. Έτσι το guard διαβάζει το κρίσιμο όριο που δημοσιεύει ο δικός σας πυρήνας (tempN_crit) και υπολογίζει προς τα πίσω. Ίδιος κώδικας, λογικοί αριθμοί παντού.

2. Γρήγορη εκκίνηση

1. Δείτε πώς φαίνεται το μηχάνημά σας στο guard. Πάντα πρώτα αυτό:

```
cd ~/PhantomSDR-Plus
python3 thermal_guard.py --once
```

```
sensor      : hwmon:coretemp
temperature: 65.0 C
crit        : 100.0 C
mode        : log
thresholds  : warn 88.0 throttle 92.0 stop 95.0 resume 75.0 (C)
cpufreq     : not writable – throttle stage disabled
```

Αν το sensor λέει NONE, σταματήστε εδώ και διαβάστε την [Επίλυση προβλημάτων](#) — το guard δεν μπορεί να προστατεύσει μηχάνημα που δεν μπορεί να μετρήσει, και σας το λέει αντί να προσποιείται.

2. Αφήστε το λίγες μέρες σε λειτουργία log. Ήδη τρέχει μέσα στον πίνακα διαχείρισης. Αφήστε το να παρακολουθήσει μια κανονική εβδομάδα — καλοκαιρινά απογεύματα, ένα μεγάλο build, ένα πολυσύχναστο Σάββατο με contest — και δείτε αν θα είχε ενεργοποιηθεί ποτέ.

3. Οπλίστε το. Πίνακας διαχείρισης → **Settings** → **THERMAL GUARD** → **MODE** → stop+restart → **SAVE THERMAL SETTINGS**. Ισχύει μέσα σε περίπου 2 δευτερόλεπτα· δεν χρειάζεται επανεκκίνηση.

3. Πώς αποφασίζει

Ο αισθητήρας

Το guard εμπιστεύεται μόνο πραγματικούς αισθητήρες πυριτίου της CPU:

Πηγή	Τυπικό υλικό
coretemp	Intel
k10temp, zenpower, zenpower3	AMD
cpu_thermal, soc_thermal	Raspberry Pi, πλακέτες ARM
Ζώνη x86_pkg_temp	εφεδρεία σε x86

Το `acpitz` και οι ζώνες χωρίς ετικέτα εξαιρούνται **σκόπιμα**. Σε έναν τυπικό σταθερό υπολογιστή το `acpitz` δείχνει περίπου 28 °C ενώ το πακέτο της CPU είναι στους 59 °C — ένα `guard` που το εμπιστευόταν απλώς δεν θα ενεργοποιούνταν ποτέ. Το να αναφέρει «απενεργοποιημένο, κανένας αξιόπιστος αισθητήρας» είναι πιο τίμιο από το να προσποιείται ότι προστατεύει.

Το `psutil` χρησιμοποιείται αν τυχαίνει να είναι εγκατεστημένο, αλλά δεν απαιτείται ποτέ. Το `guard` είναι καθαρή `standard library`.

Τα όρια

Καθένα είναι μια απόσταση κάτω από το κρίσιμο όριο που δημοσιεύει το ίδιο το μηχάνημα:

Όριο	Απόσταση από το <code>crit</code>	Τι σημαίνει
warn	−12 °C	Κάτι δεν πάει καλά, να καταγραφεί
throttle	−8 °C	Δοκίμασε ψύξη χαμηλώνοντας τη συχνότητα
stop	−5 °C	Πολύ κοντά στο όριο — σταμάτα τον διακομιστή
resume	−25 °C	Πραγματικά κρύο ξανά, ασφαλής επιστροφή

Στην πράξη αυτό δίνει:

Η CPU σας αναφέρει	warn	throttle	stop	resume
Intel, crit 100 °C	88	92	95	75
AMD Ryzen, crit 95 °C	83	87	90	70
Raspberry Pi, crit 85 °C	73	77	80	60

Αν το μηχάνημα δεν δημοσιεύει καθόλου κρίσιμο σημείο, το `guard` πέφτει σε ένα συντηρητικό σταθερό σύνολο και το δηλώνει. Μπορείτε να παρακάμψετε οποιοδήποτε όριο με απόλυτη τιμή (δείτε [Αναφορά ρυθμίσεων](#)), αλλά δεν θα έπρεπε να χρειαστεί.

Χρόνοι διατήρησης — γιατί μια κορυφή δεν ενεργοποιεί ποτέ τίποτα

Τίποτα δεν συμβαίνει από μία μόνο ζεστή μέτρηση. Ένα όριο πρέπει να διατηρηθεί **συνεχώς**:

- **warn** και **throttle**: 30 δευτ. (`thermal_warn_sustain_s`)
- **stop**: 60 δευτ. (`thermal_sustain_s`)
- **resume**: 300 δευτ. κάτω από το όριο `resume` πριν από επανεκκίνηση (`thermal_resume_s`)

Μια κορυφή από `compile`, μια ριπή αποκωδικοποίησης, ένας ανεμιστήρας που ανεβάζει στροφές — όλα πολύ σύντομα για να μετρήσουν. Μία μόνο μέτρηση κάτω από το όριο μηδενίζει το ρολόι. Επιπλέον, μετά την εκκίνηση το `guard` απαιτεί 3 συνεχόμενες έγκυρες μετρήσεις πριν κάνει οτιδήποτε, και απορρίπτει κάθε τιμή εκτός 20–125 °C ως σφάλμα αισθητήρα και όχι ως θερμοκρασία.

4. Οι τέσσερις λειτουργίες

Το `thermal_mode` είναι ο ένας διακόπτης που καθορίζει πόσο ψηλά επιτρέπεται να ανέβει το `guard` στη σκάλα. Κάθε λειτουργία **περιλαμβάνει όλα όσα κάνουν οι προηγούμενες**.

Λειτουργία	Προειδοποιεί	Περιορίζει	Σταματά	Επανεκκινεί	Μπορεί να πέσει ο διακομιστής;
log	✓	—	—	—	Όχι — δεν αγγίζει τίποτα
throttle	✓	✓	—	—	Όχι
stop	✓	✓	✓	—	Ναι, μένει κάτω μέχρι να δράσετε
stop+restart	✓	✓	✓	✓	Ναι, επανέρχεται μόνο του

log — παρακολούθηση και αναφορά (η προεπιλογή)

Τι κάνει: όλα όσα κάνουν οι άλλες λειτουργίες, εκτός από το να δρα. Ακολουθεί ολόκληρη τη σκάλα και γράφει στο `crash.log` τι θα είχε κάνει. Δεν εκτελεί ποτέ το `stop script` σας, δεν αγγίζει τη συχνότητα της CPU, δεν διακόπτει κανέναν ακροατή.

Τι πρέπει να κάνετε: τίποτα — ήδη τρέχει. Μετά από μια εβδομάδα ανοίξτε την καρτέλα **CRASH** στον πίνακα διαχείρισης ή τρέξτε:

```
grep THERMAL crash.log
```

- **Δεν υπάρχει τίποτα;** Η ψύξη σας είναι εντάξει και μπορείτε να οπλίσετε το guard με σιγουριά.
- **STOP level reached ... NOT stopping;** Το guard θα είχε κατεβάσει τον διακομιστή σας. Αυτό είναι πραγματικό θερμικό πρόβλημα με το οποίο λειτουργούσατε — διορθώστε την ψύξη και οπλίστε το guard.

Χρησιμοποιήστε την όταν: μόλις εγκαταστήσατε το guard και δεν ξέρετε ακόμη πώς συμπεριφέρεται το μηχανήμα σας. Είναι λειτουργία δοκιμαστικής περιόδου, όχι προορισμός — ένα guard σε `log` δεν προστατεύει τίποτα.

throttle — χαμήλωμα της CPU, καμία διακοπή υπηρεσίας

Τι κάνει: στο όριο `throttle` χαμηλώνει τη μέγιστη συχνότητα της CPU κατά 20 % και την αποκαθιστά μόλις η θερμοκρασία επιστρέψει κάτω από τη γραμμή `warn`. Το WebSDR σας μένει όρθιο σε όλη τη διάρκεια. Οι ακροατές ίσως παρατηρήσουν ελαφρώς χειρότερη απόδοση υπό βαρύ φορτίο· οι περισσότεροι δεν θα παρατηρήσουν τίποτα.

Αν η θερμοκρασία συνεχίσει να ανεβαίνει και περάσει το όριο `stop`, αυτή η λειτουργία **το καταγράφει και δεν κάνει τίποτα** — δεν θα σταματήσει τον διακομιστή.

Τι πρέπει να κάνετε:

1. Δώστε στο guard δικαίωμα εγγραφής στο όριο συχνότητας της CPU — το χρειάζεται, και εξ ορισμού το έχει μόνο ο `root`. Τρέξτε μία φορά `./setup-cpufreq-perms.sh`· δείτε την [ενότητα 8](#).
2. Επιβεβαιώστε με `python3 thermal_guard.py --once` ότι τυπώνει `cpufreq : writable, throttle stage available`. Αν λέει `not writable`, αυτή η λειτουργία δεν κάνει τίποτα παραπάνω από το `log`, και θα το λέει κάθε φορά στο αρχείο καταγραφής.

Χρησιμοποιήστε την όταν: κυνηγάτε ένα οριακό πρόβλημα ψύξης και θέλετε να υπερασπιστείτε το μηχανήμα χωρίς να χάσετε ποτέ ακροατές. **Προσοχή, δεν είναι πλήρης προστασία** — αν ο περιορισμός δεν φτάνει, δεν συμβαίνει τίποτα άλλο. Καλό ως πρώτο βήμα, κακό ως τελική απάντηση.

stop — κατέβασμα του διακομιστή και παραμονή εκτός

Τι κάνει: στο όριο `stop` εκτελεί το `stop script` σας και μετά κρατά το **lockout** — ξανασταματά τον διακομιστή κάθε 2 δευτερόλεπτα όσο το μηχανήμα είναι υπερθερμασμένο, ώστε τίποτα να μην τον

επαναφέρει πίσω από την πλάτη σας. Όταν η CPU μείνει 5 λεπτά κάτω από το όριο resume, το **lockout** λύνεται. **Ο διακομιστής δεν επιστρέφει μόνος του.** Τον ξεκινάτε εσείς όταν βεβαιωθείτε ότι η αιτία διορθώθηκε.

Τι πρέπει να κάνετε:

- Βεβαιωθείτε ότι το guard ξέρει πώς να σταματήσει τον διακομιστή σας. Στον πίνακα διαχείρισης αυτό είναι το ρυθμισμένο stop script (`./stop-websdr.sh` για τους παρεχόμενους εκκινήτες). Αν δεν έχει οριστεί, το guard καταφεύγει σε SIGTERM στο όνομα της διεργασίας και μετά σε SIGKILL ύστερα από 10 δευτερόλεπτα χάριτος — δουλεύει, αλλά ένα κανονικό stop script είναι καθαρότερο.
- Δεχτείτε ότι **το WebSDR σας θα είναι εκτός μέχρι να το πάρετε είδηση**. Φροντίστε να το μαθαίνετε: μια παρακολούθηση διαθεσιμότητας, η καρτέλα CRASH, ή απλώς ένας καθημερινός έλεγχος.

Χρησιμοποιήστε την όταν: το μηχάνημα μετράει περισσότερο από την υπηρεσία, είστε συνήθως εκεί, ή ένα θερμικό συμβάν είναι αρκετά σοβαρό ώστε να θέλετε να ελέγξετε το μηχάνημα πριν συνεχίσει. Επίσης η σωστή επιλογή αν δεν εμπιστεύεστε την αυτόματη επανεκκίνηση στο υλικό σας.

stop+restart — κατέβασμα, ψύξη, αυτόματη επιστροφή

Τι κάνει: ό,τι και το stop, και επιπλέον: μόλις η CPU μείνει 5 λεπτά κάτω από το όριο resume, εκτελεί το start script σας και το WebSDR επιστρέφει μόνο του.

Το «φτερούγισμα» αποτρέπεται από το `thermal_max_stops_hour` (προεπιλογή 2). Μετά από δύο θερμικές διακοπές μέσα σε μία ώρα, το guard **απενεργοποιεί την αυτόματη επανεκκίνηση** και αφήνει τον διακομιστή κάτω για να τον δείτε. Προσέξτε τη σημαντική λεπτομέρεια: το όριο σβήνει την επανεκκίνηση, ποτέ την προστασία — το lockout συνεχίζει να σταματά τον διακομιστή όσο είναι ζεστός, όσες φορές κι αν έχει ήδη ενεργοποιηθεί.

Τι πρέπει να κάνετε:

- Βεβαιωθείτε ότι **και τα δύο** scripts είναι ρυθμισμένα και δουλεύουν αυτόνομα — τόσο το start όσο και το stop. Δοκιμάστε τα με το χέρι: `./stop-websdr.sh` και μετά `./start-rx888mk2.sh`.
- Τίποτα άλλο. Αυτή είναι η λειτουργία «βάλ' το και ξέχνα το».

Χρησιμοποιήστε την όταν: το WebSDR είναι δημόσιο, χωρίς επιτήρηση ή απομακρυσμένο — δηλαδή στις περισσότερες περιπτώσεις. Είναι η προτεινόμενη λειτουργία για μια κανονική εγκατάσταση.

5. Επιλογή λειτουργίας

Η κατάσταση σας	Λειτουργία
Μόλις εγκαταστήσατε το guard, δεν ξέρετε ακόμη το μηχάνημα	log για μια εβδομάδα
Δημόσιο WebSDR, χωρίς επιτήρηση, θέλετε να φροντίζει τον εαυτό του	stop+restart
Απομακρυσμένη θέση όπου δεν πάτε εύκολα	stop+restart
Θέλετε να ελέγχετε το μηχάνημα μετά από κάθε θερμικό συμβάν	stop
Υλικό που δεν εμπιστεύεστε για αυτόματη επανεκκίνηση	stop
Οριακή ψύξη, δεν πρέπει να χάνονται ακροατές	throttle, και μετά stop+restart
Μηχάνημα χωρίς αξιόπιστο αισθητήρα CPU	καμία δεν δουλεύει — πρώτα ο αισθητήρας

Για τους περισσότερους διαχειριστές η απάντηση είναι `stop+restart`, και η τίμια περίληψη των υπολοίπων είναι: το `log` δεν προστατεύει τίποτα και το `throttle` προστατεύει λίγο.

6. Αναφορά ρυθμίσεων

Όλα τα κλειδιά βρίσκονται στο `admin_config.json` δίπλα στο `thermal_guard.py`, και καθένα από αυτά μπορεί επίσης να αλλάξει από **Settings** → **THERMAL GUARD**. Το guard **ξαναδιαβάζει το αρχείο σε κάθε κύκλο**, οπότε οι αλλαγές ισχύουν μέσα σε ~2 δευτερόλεπτα — χωρίς επανεκκίνηση.

Κλειδί	Προεπιλογή	Σημασία
<code>thermal_enabled</code>	<code>true</code>	Γενικός διακόπτης. Το <code>false</code> απενεργοποιεί εντελώς το guard.
<code>thermal_mode</code>	<code>"log"</code>	<code>log</code> <code>throttle</code> <code>stop</code> <code>stop+restart</code> — δείτε την ενότητα 4 .
<code>thermal_warn</code>	<code>null</code>	Απόλυτο όριο warn σε °C. <code>null</code> = υπολογισμός από το <code>crit</code> της CPU.
<code>thermal_throttle</code>	<code>null</code>	Απόλυτο όριο throttle. <code>null</code> = υπολογισμένο.
<code>thermal_stop</code>	<code>null</code>	Απόλυτο όριο stop. <code>null</code> = υπολογισμένο.
<code>thermal_resume</code>	<code>null</code>	Απόλυτο όριο resume. <code>null</code> = υπολογισμένο.
<code>thermal_sustain_s</code>	<code>60</code>	Δευτερόλεπτα διατήρησης του ορίου stop πριν τη διακοπή. Ελάχιστο 5.
<code>thermal_warn_sustain_s</code>	<code>30</code>	Δευτερόλεπτα για τα στάδια warn και throttle. Ελάχιστο 5.
<code>thermal_resume_s</code>	<code>300</code>	Δευτερόλεπτα κάτω από το όριο resume πριν λυθεί το <code>lockout</code> . Ελάχιστο 30.
<code>thermal_max_stops_hour</code>	<code>2</code>	Θερμικές διακοπές ανά ώρα μετά τις οποίες απενεργοποιείται η αυτόματη επανεκκίνηση. Η προστασία συνεχίζεται.
<code>thermal_test_temp</code>	<code>null</code>	Προσποιείται ότι η CPU είναι σε αυτή τη θερμοκρασία. Μόνο για δοκιμές — δείτε την ενότητα 10 .

Το αυτόνομο guard χρησιμοποιεί επιπλέον τα `start_script`, `stop_script`, `sdr_process_name` και `sdr_base_dir` από το ίδιο αρχείο.

Παράκαμψη ορίου. Κάντε το μόνο αν έχετε λόγο. Ένας συνηθισμένος: το μηχάνημά σας ζεσταίνεται δικαιολογημένα στα builds και δεν θέλετε να μετράει. Ανεβάζοντας το `thermal_stop` κερδίζετε περιθώριο εις βάρος της απόστασης από το κρίσιμο σημείο — μην το βάλετε ποτέ πάνω από την τιμή `crit`, αλλιώς η CPU θα φτάσει πρώτη στη δική της τερματική προστασία και το guard γίνεται διακοσμητικό.

7. Λειτουργία χωρίς τον πίνακα διαχείρισης

Πολλοί διαχειριστές δεν εγκαθιστούν ποτέ τον πίνακα. Το guard δουλεύει μια χαρά μόνο του — ίδιο αρχείο, ίδια συμπεριφορά.

Δοκιμάστε το πρώτα στο προσκήνιο:

```
cd ~/PhantomSDR-Plus
python3 thermal_guard.py --once          # τι θα γινόταν εδώ;
python3 thermal_guard.py --mode log      # ζωντανή παρακολούθηση, Ctrl-C για έξοδο
```

Το `--mode` παρακάμπτει το `thermal_mode` από τη γραμμή εντολών, οπότε δεν χρειάζεται ποτέ να γράψετε JSON στο χέρι:

```
python3 thermal_guard.py --mode stop+restart
python3 thermal_guard.py --config /etc/phantomsdr/thermal.json
```

Μετά κάντε το μόνιμο με το υπόδειγμα μονάδας που συνοδεύει το έργο:

```
sudo cp thermal-guard.service /etc/systemd/system/
sudo nano /etc/systemd/system/thermal-guard.service # ορίστε User= και διαδρομές
sudo systemctl daemon-reload
sudo systemctl enable --now thermal-guard
systemctl status thermal-guard
```

Διαβάστε τα σχόλια στην αρχή του `thermal-guard.service` πριν το ενεργοποιήσετε. Δύο πράγματα έχουν τη μεγαλύτερη σημασία:

- Το `User=` **πρέπει να είναι λογαριασμός που μπορεί πραγματικά να εκτελέσει τα start και stop scripts σας**. Το να τρέχει το guard ως root είναι συνήθως λάθος απάντηση: θα εκτελούσε και αυτά τα scripts ως root.
- Είτε δείξτε του ένα `admin_config.json` που περιέχει `start_script` και `stop_script`, είτε δώστε `--mode` στη γραμμή `ExecStart`. Μια ελάχιστη ρύθμιση αρκεί:

```
{
  "sdr_base_dir": "/home/youruser/PhantomSDR-Plus",
  "sdr_process_name": "spectrumserver",
  "start_script": "start-rx88mk2.sh",
  "stop_script": "stop-websdr.sh",
  "thermal_mode": "stop+restart"
}
```

Ό,τι κάνει το guard εξακολουθεί να καταγράφεται στο `crash.log` δίπλα σε αυτό το αρχείο ρυθμίσεων, καθώς και στο journal του systemd (`journalctl -u thermal-guard`).

8. Ενεργοποίηση του σταδίου throttle χωρίς root

Εξ ορισμού το `python3 thermal_guard.py --once` αναφέρει:

```
cpufreq : not writable – throttle stage disabled
```

Αυτό είναι φυσιολογικό και δεν είναι σφάλμα. Ο πυρήνας δημιουργεί το

```
/sys/devices/system/cpu/cpuN/cpufreq/scaling_max_freq
```

με ιδιοκτήτη `root:root` και δικαιώματα `0644`. Ένας πίνακας διαχείρισης που τρέχει ως απλός χρήστης δεν μπορεί να γράψει εκεί, οπότε το guard παραλείπει το στάδιο throttle και το δηλώνει στο αρχείο καταγραφής αντί να αποτυγχάνει σιωπηλά.

Η λάθος λύση είναι να τρέξετε το guard ως root. Εκτελεί επίσης τα start και stop scripts σας, και αυτά πρέπει να μείνουν χωρίς προνόμια.

Η σωστή λύση είναι να δώσετε δικαίωμα εγγραφής σε αυτά ακριβώς τα αρχεία σε μια ειδική ομάδα. Ένα script το κάνει:

```
cd ~/PhantomSDR-Plus
./setup-cpufreq-perms.sh
```

Ζητά τον κωδικό σας μία φορά — και μόνο μία, κατά την εγκατάσταση — και έπειτα:

1. δημιουργεί μια ομάδα συστήματος `cpufreq` και σας προσθέτει σε αυτήν·
2. εγκαθιστά το `/etc/tmpfiles.d/99-phantomsdr-cpufreq.conf` ώστε η ομάδα και τα δικαιώματα `0664` να εφαρμόζονται ξανά **σε κάθε εκκίνηση** — τα δικαιώματα του `sysfs` δεν επιβιώνουν από μόνα τους μιας επανεκκίνησης·
3. εφαρμόζει την αλλαγή αμέσως, ώστε να δοκιμάσετε χωρίς επανεκκίνηση.

Μετά, με αυτή τη σειρά:

```
# 1. αποσυνδεθείτε και συνδεθείτε ξανά (ή κάντε επανεκκίνηση) — μια νέα
# ομάδα φτάνει στις διεργασίες σας μόνο μέσω νέας σύνδεσης
id | grep cpufreq

# 2. επανεκκινήστε τον πίνακα· το guard ελέγχει το δικαίωμα εγγραφής μία φορά, στην εκκίνηση
sudo systemctl restart phantomsdr-admin
# (έτσι δεν χρειάζεται αποσύνδεση — το systemd ξαναχτίζει τη λίστα ομάδων
# σε κάθε εκκίνηση. Μόνο αν ΔΕΝ είναι εγκατεστημένα τα units:
# ./manage_admin.sh stop && ./manage_admin.sh start — ποτέ και τα δύο, διεκδικούν τη θύρα 3000)

# 3. επιβεβαιώστε
python3 thermal_guard.py --once
# cpufreq : writable, throttle stage available
```

Για πλήρη ανάκληση: `sudo ./setup-cpufreq-perms.sh --revoke`.

Σημειώσεις

- Το στάδιο `throttle` *δρα* μόνο αν το `thermal_mode` είναι `throttle` ή υψηλότερο. Μόνη της η άδεια δεν αλλάζει τίποτα.
- Το `guard` χαμηλώνει κατά 20 % όποιο όριο ισχύει *εκείνη τη στιγμή* και αποκαθιστά ακριβώς αυτή την τιμή. Αν ήδη περιορίζετε την CPU σας (π.χ. με `MAX_SPEED` στο `/etc/init.d/cpufrequtils`), αποκαθίσταται ο δικός σας περιορισμός — το `guard` δεν θα σας επιστρέψει στα κρυφά υψηλότερη συχνότητα από όση ζητήσατε.
- Ορισμένα μηχανήματα δεν εκθέτουν καθόλου οδηγό `cpufreq` — πολλά VPS, τα περισσότερα containers. Το script το εντοπίζει και δεν αλλάζει τίποτα. Εκεί χρησιμοποιήστε `stop` ή `stop+restart`· δεν εξαρτώνται από το `cpufreq`.

9. Λειτουργία χωρίς επιτήρηση

Η πιο συχνή ερώτηση για το `guard`: *τι γίνεται όταν δεν βλέπει κανείς*; Η σύντομη απάντηση είναι ότι ακριβώς γι' αυτή την περίπτωση φτιάχτηκε.

Δεν χρειάζεται ανοιχτό browser. Το `guard` είναι ένα νήμα παρασκηνίου μέσα στο `admin_server.py`, που χρονίζεται από τον δειγματολήπτη των γραφημάτων κάθε 2 δευτερόλεπτα. Τρέχει είτε είναι κάποιος συνδεδεμένος στον πίνακα είτε όχι, είτε υπάρχει ανοιχτός `browser` είτε όχι, είτε είστε ξύπνιοι είτε όχι. Στην αυτόνομη λειτουργία είναι υπηρεσία `systemd`, ακόμη πιο ανεξάρτητη. Η κάρτα στο `dashboard` είναι ένα *παράθυρο* προς το `guard`, όχι το ίδιο το `guard`.

Δρα χωρίς να ρωτήσει. Δεν υπάρχει διάλογος επιβεβαίωσης ούτε ειδοποίηση που να περιμένει. Στο όριο `stop` σταματά τον διακομιστή, τελεία. Αυτό ακριβώς είναι το νόημα.

Αλλά μόνο αν του δώσατε άδεια. Σε λειτουργία `log` θα καταγράψει πιστά ένα μηχάνημα που ψήνεται και δεν θα κάνει τίποτα γι' αυτό. Αν διαβάσετε αυτή την ενότητα επειδή «δεν είμαι εκεί για να το βλέπω», η λειτουργία που θέλετε είναι `stop+restart` — μόνο αυτή και προστατεύει το μηχάνημα και επαναφέρει την υπηρεσία χωρίς εσάς.

Βοηθά μόνο όσο ο πίνακας είναι ζωντανός. Η θερμική προστασία είναι μέρος του `admin_server.py`: ένας πίνακας που πέθανε στις 02:00 παίρνει μαζί του και την προστασία, ενώ μετά από επανεκκίνηση το μηχάνημα μένει αψύλακτο μέχρι να συνδεθείτε και να τον ξεκινήσετε ξανά. Αν δεν παρακολουθεί κανείς, αφήστε το `systemd` να παρακολουθεί: το αποθετήριο περιλαμβάνει τα `phantomsdr-admin.service` και `phantomsdr-proxy.service`, που ξεκινούν με το `boot` και επανεκκινούν τον πίνακα μέσα σε πέντε δευτερόλεπτα από μια κατάρρευση. Δείτε το [ADMIN_PANEL_SETUP.md](#). Για δέκτη χωρίς επιτήρηση αυτό **συνιστάται έντονα** — το να οπλίσετε το `stop+restart` χωρίς αυτό σας προστατεύει μόνο μέχρι την επόμενη επανεκκίνηση.

Πώς το μαθαίνετε εκ των υστέρων. Κατά σειρά χρησιμότητας:

Πού	Τι παίρνετε
<code>crash.log</code> / καρτέλα CRASH	Κάθε ενέργεια, με χρονοσήμανση. Το μόνιμο αρχείο.
Κάρτα THERMAL GUARD στο dashboard	Ζωντανή κατάσταση: θερμοκρασία, στάδιο, <code>lockout</code> , διακοπές της τελευταίας ώρας.
Σελίδα Graphs (θερμοκρασία CPU, 24 ώρες)	Το σχήμα του συμβάντος — πόσο γρήγορα ανέβηκε, πόσο άργησε να κρυώσει.
<code>journalctl -u thermal-guard</code>	Μόνο στην αυτόνομη λειτουργία.

Μια λογική ρουτίνα για θέση χωρίς επιτήρηση: `stop+restart` οπλισμένο, και `grep THERMAL crash.log` όποτε τυχαίνει να συνδεθείτε. Αν είναι άδειο, δεν υπάρχει τίποτα να μάθετε.

Ένα παράδειγμα βήμα-βήμα. Βλάβη ανεμιστήρα στις 03:00 σε μηχάνημα με `stop+restart`:

03:14	η θερμοκρασία περνά τους 88 °C, κρατά 30 δευτ.	→ καταγραφή <code>warn</code>
03:16	περνά τους 92 °C, κρατά 30 δευτ.	→ εφαρμογή <code>throttle</code> (αν επιτρέπεται)
03:19	περνά τους 95 °C, κρατά 60 δευτ.	→ <code>STOP</code> – <code>stop script</code> , <code>lockout</code> ενεργό
03:19	ο <code>watchdog</code> ξαναξεκινά τον διακομιστή	→ το <code>guard</code> τον ξανασταματά, καταγραφή
03:26	η CPU 5 λεπτά κάτω από 75 °C	→ λύση <code>lockout</code> , εκτέλεση <code>start script</code>
03:41	υπερθερμαίνεται και σταματά δεύτερη φορά	→ όριο συχνότητας, τέλος οι επανεκκινήσεις ο διακομιστής μένει κάτω, η προστασία
	μένει	

Ξυπνάτε με ένα WebSDR εκτός λειτουργίας, ένα `crash.log` που λέει ακριβώς γιατί και — το σημαντικότερο — μια CPU που δεν πλησίασε ποτέ το κρίσιμο σημείο της.

10. Δοκιμή πριν του εμπιστευτείτε το μηχάνημα

Μην περιμένετε πραγματικό καύσιμα για να μάθετε αν δουλεύει το `stop script` σας.

Το `thermal_test_temp` κάνει το `guard` να πιστεύει ότι η CPU βρίσκεται σε θερμοκρασία που επιλέγετε εσείς. Όλα τα υπόλοιπα συμπεριφέρονται απολύτως κανονικά — οι χρόνοι διατήρησης, η σκάλα, το `lockout`, τα πραγματικά σας `scripts`.

1. Πίνακας διαχείρισης → **Settings** → **THERMAL GUARD** → **TEST TEMPERATURE**.
2. Βάλτε τιμή πάνω από το όριο `stop` (π.χ. 97 σε Intel).
3. Αποθηκεύστε και παρακολουθήστε την κάρτα στο dashboard και την καρτέλα CRASH.

4. Καθαρίστε το πεδίο όταν τελειώσετε. Μια δοκιμαστική θερμοκρασία που ξέμεινε σημαίνει ότι το guard δεν διαβάζει πια τον πραγματικό αισθητήρα.

Σε λειτουργία `log` αυτό σας δείχνει όλη τη σκάλα χωρίς κανένα ρίσκο — η ασφαλής πρώτη δοκιμή, και αυτή που κάνει κανείς πριν οπλίσει οτιδήποτε.

Σε λειτουργία `stop+restart` αυτό είναι **ζωντανή δοκιμή**: ο διακομιστής σας πράγματι θα σταματήσει και πράγματι θα επιστρέψει περίπου πέντε λεπτά αφού καθαρίσετε το πεδίο. Κάντε το όταν δεν ακούει κανείς. Αξίζει να γίνει μία φορά, γιατί είναι ο μόνος τρόπος να ξέρετε ότι τα `stop` και `start scripts` σας δουλεύουν όταν τα καλεί κάτι άλλο εκτός από εσάς.

Αντίστοιχα στην αυτόνομη λειτουργία:

```
python3 thermal_guard.py --once          # μόνο τα όρια, καμία ενέργεια
python3 thermal_guard.py --mode log      # ζωντανή παρακολούθηση της σκάλας
```

11. Διαβάζοντας το αρχείο καταγραφής

Τα πάντα προστίθενται στο `crash.log` δίπλα στο αρχείο ρυθμίσεων, ένα συμβάν ανά γραμμή, με πρόθεμα `[THERMAL]`:

```
grep THERMAL crash.log
```

Οι γραμμές είναι σκόπιμα μονές και φιλικές προς το `grep`. Τι θα δείτε:

Γραμμή	Σημασία
warn: 88.4C (warn=88.0 crit=100) sustained 30s	Πρώτο σκαλί. Δεν έγινε τίποτα.
throttle: 92.1C sustained 30s – cpufreq max lowered 20%	Χαμήλωσε η συχνότητα της CPU.
throttle level reached: ... (no cpufreq write access – stage skipped)	Δείτε την ενότητα 8 .
STOP level reached: ... mode is 'log', NOT stopping	Θα είχε σταματήσει τον διακομιστή. Οπλίστε το.
STOPPED server: 95.2C sustained 60s (stop=95.0 crit=100) – ran stop-websdr.sh	Το αληθινό συμβάν.
lockout: process reappeared at 96.0C – re-stopped (...) [12 time(s) so far]	Κάτι ξαναξεκινά τον διακομιστή σας· το guard κερδίζει. Περιορισμένο σε μία γραμμή ανά λεπτό.
rate limit: 2 thermal stops within the hour – automatic restart is now DISABLED	Μένει κάτω για να ερευνησετε. Η προστασία συνεχίζεται.
recovered: 54.0C held below 75.0 – lockout cleared	Κρύο ξανά.
auto-restart: ran start-rx88mk2.sh	Πάλι online.
back to normal: 59.0C (warn=88.0)	Κατέβηκε από τη σκάλα χωρίς ποτέ να σταματήσει.

12. Lockout και όριο συχνότητας

Αυτές οι δύο συμπεριφορές είναι που εκπλήσσουν, οπότε ας ειπωθούν καθαρά.

To lockout είναι αυτό που κάνει το guard να δουλεύει απέναντι στη δική σας εγκατάσταση. Όταν σταματά τον διακομιστή δεν στέλνει απλώς μία εντολή και ελπίζει. Σημειώνεται ως κλειδωμένο και, σε κάθε κύκλο 2 δευτερολέπτων όσο η θερμοκρασία είναι πάνω από το όριο resume, ελέγχει αν η διεργασία επέστρεψε — και αν ναι, την ξανασταματά. Ο watchdog σας, το `Restart=always` του systemd, μια εργασία cron, ένας ανυπόμονος διαχειριστής: όλοι χάνουν αυτή τη διαμάχη μέχρι να κρυώσει η CPU. Λύνεται αυτόματα μόλις η θερμοκρασία μείνει κάτω από το όριο resume για `thermal_resume_s` (προεπιλογή 5 λεπτά).

Το κουμπί CLEAR LOCKOUT στην κάρτα του dashboard λύνει το lockout και μηδενίζει τον μετρητή διακοπών ανά ώρα, για όταν έχετε διορθώσει την ψύξη και δεν θέλετε να περιμένετε. **Δεν** αποπλίζει το guard: αν το μηχάνημα είναι ακόμη ζεστό, ο αμέσως επόμενος έλεγχος θα σταματήσει ξανά τον διακομιστή. Αυτό είναι σκόπιμο.

Το όριο συχνότητας (`thermal_max_stops_hour`, προεπιλογή 2) εμποδίζει ένα χαλασμένο μηχάνημα να ανεβοκατεβαίνει όλη νύχτα. Μόλις συμπληρωθεί, το guard **απενεργοποιεί την αυτόματη επανεκκίνηση** και αφήνει τον διακομιστή κάτω. Διαβάστε το προσεκτικά: απενεργοποιεί την επανεκκίνηση, όχι την προστασία. Το lockout συνεχίζει να σταματά τον διακομιστή όσο είναι υπερθερμασμένος, όσες φορές κι αν έχει ήδη ενεργοποιηθεί. Ένα guard που θα σιωπούσε ακριβώς όταν το μηχάνημα βρίσκεται στη χειρότερη κατάστασή του θα ήταν χειρότερο από καθόλου guard.

13. Επίλυση προβλημάτων

sensor : NONE – no trusted CPU sensor on this machine Το guard δεν βρήκε επιτρεπόμενο αισθητήρα και αρνείται να μαντέψει. Δοκιμάστε `sensors` (από το `lm-sensors`: `sudo apt install lm-sensors && sudo sensors-detect`). Μέσα σε εικονική μηχανή ή container συχνά πράγματι δεν εκτίθεται αισθητήρας πυριτίου — το guard δεν μπορεί να προστατεύσει τέτοιο μηχάνημα και σωστά δηλώνεται απενεργοποιημένο αντί να προσποιείται.

cpufreq : not writable – throttle stage disabled Αναμενόμενο αν δεν έχετε τρέξει το `./setup-cpufreq-perms.sh` — δείτε την [ενότητα 8](#). Επηρεάζει μόνο το στάδιο throttle· τα stop και stop+restart δεν θίγονται.

Έτρεξα το setup-cpufreq-perms.sh και εξακολουθεί να λέει not writable Δύο πιθανές αιτίες: δεν έχετε αποσυνδεθεί και συνδεθεί ξανά από τότε (μια νέα ομάδα φτάνει στις διεργασίες μόνο μέσω νέας σύνδεσης — ελέγξτε με `id`), ή ο πίνακας δεν έχει επανεκκινηθεί από τότε (το guard ελέγχει το δικαίωμα εγγραφής μία φορά, στην εκκίνηση). Κάντε και τα δύο, με αυτή τη σειρά.

Τα όρια μοιάζουν λάθος για τη CPU μου Δείτε τι δημοσιεύει το μηχάνημα: `cat /sys/class/hwmon/hwmon*/temp*_crit`. Αν το crit σας είναι ασυνήθιστο ή απουσιάζει, ορίστε απόλυτες τιμές στα `thermal_stop` και τα συγγενικά.

Σταμάτησε τον διακομιστή μου και δεν νομίζω ότι ήταν ζεστός Ελέγξτε μήπως έχει μείνει ορισμένο το `thermal_test_temp` από κάποια δοκιμή. Αυτή είναι με διαφορά η συνηθέστερη αιτία.

Δεν ενεργοποιείται ποτέ αν και το μηχάνημα ζεσταίνεται Επιβεβαιώστε ότι η λειτουργία δεν είναι log, ότι το `thermal_enabled` είναι true, και συγκρίνετε τη θερμοκρασία που βλέπετε με τα όρια από το `--once`. Θυμηθείτε ότι το όριο πρέπει να διατηρηθεί συνεχώς για όλο τον χρόνο διατήρησης.

Ο διακομιστής ξαναξεκινά συνεχώς κατά τη διάρκεια θερμικού συμβάντος Αυτό είναι το lockout που δουλεύει, και οι γραμμές `lockout:` είναι η αναφορά επιτυχίας του — ο επιτηρητής σας συνεχίζει να προσπαθεί, το guard συνεχίζει να το αναιρεί. Δεν υπάρχει τίποτα να διορθωθεί.

Ο πίνακας δεν τρέχει· τρέχει το guard; Όχι — στη λειτουργία με πίνακα το guard ζει μέσα στο `admin_server.py`. Αν θέλετε προστασία ανεξάρτητη από τον πίνακα, χρησιμοποιήστε την αυτόνομη υπηρεσία της [ενότητας 7](#).

14. Απενεργοποίηση ή αφαίρεση

- **Παύση:** βάλτε το `thermal_mode` σε `log`. Συνεχίζει να παρακολουθεί και να αναφέρει, χωρίς ποτέ να δρα.
- **Πλήρης απενεργοποίηση:** βάλτε το `thermal_enabled` σε `false`.
- **Αυτόνομο:** `sudo systemctl disable --now thermal-guard`.
- **Αναίρεση των δικαιωμάτων `cpufreq`:** `sudo ./setup-cpufreq-perms.sh --revoke`.

Η διαγραφή του `thermal_guard.py` είναι επίσης ασφαλής — το `admin_server.py` το εισάγει μέσα σε `try` και απλώς αναφέρει το `guard` ως μη διαθέσιμο αν λείπει.

Δείτε επίσης

- [Εγκατάσταση πίνακα διαχείρισης](#) — η καρτέλα Thermal Guard στο πλαίσió της
- [Εγκατάσταση](#) — θερμική προστασία σε νέα εγκατάσταση
- [Δομή έργου](#) — πού βρίσκεται το `thermal_guard.py`

Όρια συνδέσεων

CONNECTION_LIMITS.md

Όρια συνδέσεων — Εγχειρίδιο διαχειριστή

Προστασία ενός δημόσιου δέκτη από πλημμύρα συνδέσεων.

Ένα WebSDR είναι μια δημόσια υπηρεσία που δίνει σε όποιον ζητήσει μια συνεχή ροή ήχου και φάσματος. Αυτός είναι ο σκοπός του, και εκεί ακριβώς βρίσκεται και το πρόβλημα: για τον διακομιστή, τίποτα σε έναν κανονικό ακροατή δεν ξεχωρίζει από κάποιον που ανοίγει συνεδρίες σε βρόχο. Στις 16 Σεπτεμβρίου 2026 αυτός ο δέκτης δέχτηκε ακριβώς αυτό — μία διεύθυνση άνοιξε 134 συνεδρίες ακρόασης μέσα σε ένα λεπτό και κράτησε 181 ταυτόχρονα, ενώ το πραγματικό κοινό του σταθμού ήταν γύρω στους δώδεκα.

Αυτό το εγχειρίδιο αφορά τα τρία επίπεδα που το σταματούν, τι μπορεί και τι δεν μπορεί να κάνει το καθένα, και πώς θα καταλάβετε αν τα δικά σας είναι λογικά ρυθμισμένα.

Βιάζεστε; Το παράδειγμα `config.toml` έρχεται ήδη με λογικές τιμές, οπότε μια καθαρή εγκατάσταση είναι προστατευμένη χωρίς να κάνετε τίποτα. Αν το δικό σας `.toml` είναι παλαιότερο και δεν έχει κλειδιά `[limits]` πέρα από τα `audio/waterfall/events`, τότε τα όρια ανά διεύθυνση είναι απλώς **ανενεργά** — αντιγράψτε το μπλοκ από το παράδειγμα για να τα ενεργοποιήσετε. Η μία εντολή που αξίζει να ξέρετε μετά είναι `grep Refused spectrumserver.log`, που σας λέει αν απορρίπτεται κανείς.

Περιεχόμενα

1. Από τι αμύνεστε στην πραγματικότητα
2. Τα τρία επίπεδα
3. Όρια ανά διεύθυνση — η πολιτική για τους ακροατές
4. Αδρανείς συνδέσεις — η σιωπηλή επίθεση
5. Η προστασία στον πυρήνα — `setup-firewall.sh`
6. Αναφορά ρυθμίσεων
7. Τι βλέπει ένας ακροατής που απορρίφθηκε
8. Διαβάζοντας τα αρχεία καταγραφής και τους μετρητές
9. Επιλέγοντας τους αριθμούς σας
10. Παγίδες που αξίζει να ξέρετε
11. Απενεργοποίηση όλων

1. Από τι αμύνεστε στην πραγματικότητα

Αξίζει να είμαστε ακριβείς, γιατί η λέξη «DDoS» καλύπτει δύο πολύ διαφορετικά πράγματα και μόνο το ένα λύνεται στο δικό σας μηχανήμα.

Πλημμύρα από μία πηγή. Ένας υπολογιστής, ή λίγοι, που ανοίγουν συνδέσεις όσο πιο γρήγορα μπορούν. Αυτό συνέβη εδώ και διορθώνεται πλήρως: οι συνδέσεις έρχονται από μια διεύθυνση που μπορείτε να μετρήσετε, και η μέτρηση αρκεί.

Μια πραγματικά κατανεμημένη επίθεση. Χιλιάδες υπολογιστές, συχνά με πλαστές διευθύνσεις προέλευσης, που γεμίζουν τη γραμμή σας. Όταν τα πακέτα φτάνουν στην κάρτα δικτύου σας, το εύρος ζώνης έχει ήδη ξοδευτεί. **Τίποτα από όσα περιγράφονται εδώ δεν βοηθά**, και τίποτα που εγκαθιστάτε στον διακομιστή δεν μπορεί — η μόνη απάντηση είναι μια υπηρεσία πριν από εσάς, κάτι που είναι άλλη συζήτηση και απαιτεί πραγματικό domain και έναν πάροχο όπως το Cloudflare μπροστά από τον δέκτη.

Όλα τα παρακάτω αφορούν την πρώτη περίπτωση. Δεν είναι περιορισμός που χρειάζεται απολογία: η πρώτη περίπτωση είναι αυτό που συμβαίνει στην πράξη σε ερασιτεχνικούς σταθμούς.

2. Τα τρία επίπεδα

το διαδίκτυο



1. nftables setup-firewall.sh
χονδροειδές, φθηνό, κόβει όγκο πριν το διαβάσει
οτιδήποτε



2. spectrumserver [limits] στο config.toml
ακριβές, ανά ακροατή, ξέρει ποιος είναι ποιος



3. ο browser λέει το γιατί, και δεν ξαναδοκιμάζει ποτέ |

Δεν είναι σκόπιμα εφεδρικά μεταξύ τους. Το επίπεδο του πυρήνα είναι γρήγορο αλλά χαζό — μετράει πακέτα από μια διεύθυνση και δεν ξέρει τίποτα για ακροατές. Το επίπεδο του διακομιστή είναι ακριβές αλλά βλέπει μια σύνδεση μόνο αφού γίνει δεκτή και αναλυθεί, πράγμα που κοστίζει. Το επίπεδο του browser υπάρχει ώστε ένας ακροατής που απορρίφθηκε να μάθει τι έγινε, αντί να κοιτάζει μια σελίδα που δεν φορτώνει ποτέ.

3. Όρια ανά διεύθυνση

Τι μετράει ως ένας ακροατής

Ακροατής είναι **μία σύνδεση /audio** — ή, όπου είναι ενεργό το [kiwi_emulation], **μία σύνδεση ήχου Kiwi**, που μετράει ακριβώς το ίδιο. Αυτή είναι η ίδια μονάδα που χρησιμοποιούν ήδη ο μετρητής χρηστών, το users.json και οι ετικέτες πάνω στον καταρράκτη, οπότε ένας πελάτης Kiwi καταναλώνει θέση και εμφανίζεται στη λίστα χρηστών όπως κι ένας browser.

Αυτό έχει σημασία γιατί μία μόνο καρτέλα browser σε υπολογιστή ανοίγει **τέσσερις** συνδέσεις προς τον διακομιστή σας:

Σύνδεση	Σκοπός
/audio	η ροή αποδιαμορφωμένου ήχου — αυτή είναι ο ακροατής
/waterfall	η απεικόνιση του φάσματος
/events	λίστα χρηστών, συχνότητες των άλλων ακροατών
/chat	το πλαίσιο συνομιλίας

Ένα κινητό στο /mobile ανοίγει δύο (/audio και /chat), γιατί δεν έχει καταρράκτη. Έτσι ένα όριο που μετρούσε σκέτες συνδέσεις θα απέρριπτε τους πάντες: τρεις καρτέλες είναι δώδεκα συνδέσεις.

Τα τρία κλειδιά

```
[limits]
per_ip=3          # ταυτόχρονοι ακροατές από μία διεύθυνση
per_ip_sockets=0  # ταυτόχρονες συνδέσεις κάθε είδους· το 0 δίνει per_ip * 4 + 4
per_ip_rate=40    # νέες συνδέσεις ανά λεπτό από μία διεύθυνση
per_ip_ban_s=600  # για πόσο απορρίπτεται μια διεύθυνση που ξεπέρασε τον ρυθμό
```

Το **per_ip** είναι ο βασικός αριθμός και αυτός που θα σκεφτείτε. Το τρία είναι καλή αφετηρία.

Το **per_ip_sockets** υπάρχει γιατί το /audio δεν είναι το μόνο που αξίζει να πλημμυρίσει κανείς. Το /waterfall είναι η βαρύτερη ροή που παράγει ο διακομιστής σας, και ένα όριο που μετρούσε μόνο ακροατές θα παρακαμπτόταν απλώς ανοίγοντας συνδέσεις καταρράκτη. Αν μείνει στο 0 γίνεται $\text{per_ip} * 4 + 4$, που χωράει per_ip πλήρεις καρτέλες υπολογιστή με λίγο περιθώριο για συνδέσεις που ο πυρήνας δεν έχει προλάβει να καθαρίσει μετά από ανανέωση σελίδας.

Το **per_ip_rate** είναι αυτό που πραγματικά τερματίζει μια πλημμύρα. Το όριο από μόνο του δεν αρκεί: η απόρριψη μιας σύνδεσης είναι φθηνή αλλά όχι δωρεάν, και ο επιτιθέμενος μπορεί να επανασυνδέεται όσο γρήγορα γυρίζουν οι απορρίψεις, καίγοντας CPU σε στήσιμο και ξήλωμα. Το όριο ρυθμού του δίνει αποκλεισμό αντί για απάντηση. Θυμηθείτε ότι μία καρτέλα κοστίζει τέσσερα, οπότε 40/λεπτό είναι περίπου δέκα φορτώσεις σελίδας το λεπτό από μία διεύθυνση.

Μια σκόπιμη σειρά. Ο ρυθμός χρεώνεται πριν ελεγχθεί το όριο ταυτόχρονων. Αυτό μοιάζει ανάποδο μέχρι να σκεφτείτε μια διεύθυνση που είναι ήδη στο όριο της και επανασυνδέεται σε βρόχο: αν της απαντούσε κάθε φορά η φθηνή απόρριψη «είσαι στο όριο», οι παραβιάσεις ρυθμού δεν θα συσσωρεύονταν ποτέ και ο αποκλεισμός που τερματίζει τον βρόχο δεν θα ενεργοποιούταν ποτέ.

Ποιοι εξαιρούνται

Το loopback δεν περιορίζεται ποτέ. Το autorun tap για την αναφορά spots, ο πίνακας διαχείρισης και ένας browser ανοιγμένος στο ίδιο το μηχάνημα του διακομιστή έρχονται όλα από το 127.0.0.1, και το να τους αφήναμε να καταναλώνουν θέσεις θα σήμαινε ότι ο ίδιος σας ο σταθμός κλειδώνει έξω τους ακροατές σας.

Αυτό καλύπτει και το proxy.py: οι συνδέσεις μέσω του proxy φτάνουν στον spectrumserver από loopback, και η πραγματική διεύθυνση του πελάτη έρχεται στην κεφαλίδα X-Forwarded-For, που είναι αυτή στην οποία κλειδώνονται τα όρια.

Από την v5.0.0 η κεφαλίδα αυτή γίνεται πιστευτή **μόνο από ομότιμο loopback** — έναν proxy στον ίδιο υπολογιστή. Όταν τη στέλνει οποιοσδήποτε άλλος αγνοείται και μετρά η διεύθυνση TCP, γιατί αλλιώς ένας απευθείας επισκέπτης θα μπορούσε να γράψει X-Forwarded-For: 127.0.0.1 και να περάσει ως loopback όλα τα όρια αυτής της σελίδας. Το proxy.py επίσης πετά κάθε X-Forwarded-For ή X-Real-IP που στέλνει ένας πελάτης πριν προσθέσει τα δικά του, και απαντά στο kick του διαχειριστή (/~kick) μόνο για πελάτες στον ίδιο υπολογιστή. Ένας σταθμός με πολλούς δέκτες πίσω από τον proxy περιγράφεται στο [Πολλοί δέκτες](#), εκεί τα όρια ισχύουν ανά δέκτη.

4. Αδρανείς συνδέσεις

Υπάρχει μια επίθεση που κανένα από τα παραπάνω δεν μπορεί να δει.

Ανοίγεις μια σύνδεση TCP. Δεν στέλνεις τίποτα. Την κρατάς.

Ένας τέτοιος πελάτης δεν ολοκληρώνει ποτέ αίτημα, οπότε δεν φτάνει ποτέ στον κώδικα όπου ζουν τα όρια ανά διεύθυνση — δεν υπάρχει ακροατής, δεν υπάρχει αντικείμενο πελάτη, τίποτα δεν έχει όνομα για να μετρηθεί. Στο μεταξύ του κοστίζει λίγα bytes και σε εσάς κοστίζει έναν περιγραφέα αρχείου και μια θέση για όσο κρατιέται.

```
[limits]
idle_per_ip=8      # αδρανείς συνδέσεις (πριν από αίτημα) ανά διεύθυνση
idle_total=512     # ανώτατο όριο για όλες τις διευθύνσεις μαζί
```

Το όριο είναι στο **πόσες**, όχι στο **για πόσο**, και αυτή η διάκριση είναι όλη η σχεδίαση. **Δεν υπάρχει προθεσμία σε αδρανή σύνδεση, και αυτό είναι σκόπιμο.** Μια προθεσμία μοιάζει η προφανής απάντηση και είναι παγίδα (δείτε §10): υπάρχουν νόμιμοι πελάτες που συνδέονται και μετά μένουν σιωπηλοί για πολλή ώρα πριν μιλήσουν. Αυτό που τους ξεχωρίζει από μια επίθεση είναι η ποσότητα — ένας νόμιμος πελάτης κρατάει μία τέτοια σύνδεση, ένας επιτιθέμενος χιλιάδες.

Μια αδρανής σύνδεση ζει λοιπόν μέχρι να κάνει αίτημα ή να πεθάνει, και η θέση της ελευθερώνεται και στις δύο περιπτώσεις.

Το `idle_total` φράζει μια πλημμύρα απλωμένη σε πολλές διευθύνσεις προέλευσης. Όταν πιαστεί το ταβάνι, απορρίπτονται οι **νέες** αδρανείς συνδέσεις αντί να διώχνονται οι υπάρχουσες, ώστε μια νόμιμη μακρόβια σύνδεση να μην είναι ποτέ αυτή που θυσιάζεται για να ελευθερωθεί χώρος.

Και τα δύο εφαρμόζονται τη στιγμή που γίνεται δεκτή η σύνδεση, με βάση τη διεύθυνση του πελάτη — δεν υπάρχει ακόμη αίτημα, άρα δεν υπάρχει `X-Forwarded-For` να συμβουλευτούμε. Αυτό δεν πειράζει: το `loopback` δεν μετρείται, πράγμα που καλύπτει ό,τι έρχεται μέσω `proxy.py`, και μια απευθείας πλημμύρα κουβαλάει τη δική της διεύθυνση.

5. Η προστασία στον πυρήνα

Το `setup-firewall.sh` εγκαθιστά έναν πίνακα `nftables` που κόβει όγκο πριν ο `spectrumserver` διαβάσει έστω ένα byte. Είναι σκόπιμα χονδροειδές, και οι αριθμοί του είναι αρκετά πάνω από αυτούς του `config.toml`, ώστε ένας κανονικός ακροατής να μην μπορεί ποτέ να πιαστεί και από τα δύο.

```
./setup-firewall.sh --show      # τύπωσε τους κανόνες, μην αλλάξεις τίποτα
sudo ./setup-firewall.sh --check # έλεγχος εγκυρότητας στον πυρήνα σας
sudo ./setup-firewall.sh --apply # φόρτωση, με αυτόματη επαναφορά σε 60 δευτ.
sudo ./setup-firewall.sh --persist # επαναφόρτωση στην εκκίνηση
sudo ./setup-firewall.sh --status # οι μετρητές πακέτων ανά κανόνα
sudo ./setup-firewall.sh --remove # αναίρεση όλων
```

Καλύπτει τέσσερα πράγματα: ένα ταβάνι ταυτόχρονων συνδέσεων ανά διεύθυνση προέλευσης στις θύρες του `proxy` και του δέκτη, έναν ρυθμό νέων συνδέσεων ανά διεύθυνση, ένα φρένο για επιθέσεις εξαντλητικής δοκιμής στο SSH, και το Windows file sharing κλειστό σε οτιδήποτε εκτός ιδιωτικών περιοχών διευθύνσεων — ο `smbd` ακούει στο `0.0.0.0` σε πολλά μηχανήματα, και το αν αυτό είναι προσβάσιμο εξαρτάται από έναν δρομολογητή που το script δεν μπορεί να δει.

Οι θύρες διαβάζονται από το `admin_config.json`, το ίδιο αρχείο που χρησιμοποιεί το `proxy.py`, οπότε ακολουθεί τη δική σας εγκατάσταση.

Γιατί δεν μπορεί να σας κλειδώσει έξω

Δύο ιδιότητες, και οι δύο σκόπιμες:

- Η πολιτική του πίνακα είναι **accept**, και απορρίπτει μόνο συγκεκριμένα ονομασμένα μοτίβα. Ένας κανόνας που δεν ταιριάζει αφήνει το πακέτο να περάσει σε ό,τι άλλο έχετε ρυθμίσει. Δεν μπορεί να «μαυρίσει» το μηχάνημα.
- **Οι εδραιωμένες συνδέσεις γίνονται δεκτές στον πρώτο κιόλας κανόνα.** Η συνεδρία SSH στην οποία πληκτρολογείτε δεν επηρεάζεται ποτέ από όσα ακολουθούν.

Πάνω από αυτά, το `--apply` οπλίζει αυτόματη επαναφορά: φορτώνει τους κανόνες και μετά περιμένει επιβεβαίωση μέσα σε 60 δευτερόλεπτα, αλλιώς τους αφαιρεί. Μην πείτε τίποτα, κλείστε το τερματικό, χάστε τη σύνδεσή σας — το σύνολο κανόνων εξαφανίζεται από μόνο του.

Χρησιμοποιήστε σωστά αυτό το παράθυρο. Ελέγξτε **από άλλη συσκευή, εκτός του δικού σας δικτύου**, ότι ο δέκτης φορτώνει ακόμη και ότι ανοίγει νέα συνεδρία SSH. Ο έλεγχος με τη συνεδρία που ήδη έχετε δεν αποδεικνύει τίποτα, γιατί έγινε δεκτή από τον πρώτο κανόνα.

6. Αναφορά ρυθμίσεων

Κάθε κλειδί παρακάτω ζει κάτω από το `[limits]` στο αρχείο `.toml` σας. **Όλα έχουν προεπιλογή ανενεργή ή γενναιόδωρη**, οπότε μια ρύθμιση που δεν τα αναφέρει συμπεριφέρεται ακριβώς όπως πάντα.

Κλειδί	Προεπιλογή	Σημασία
<code>per_ip</code>	0 (ανενεργό)	Ταυτόχρονοι ακροατές ανά διεύθυνση
<code>per_ip_sockets</code>	$0 \rightarrow \text{per_ip} * 4 + 4$	Ταυτόχρονες συνδέσεις κάθε είδους ανά διεύθυνση
<code>per_ip_rate</code>	0 (ανενεργό)	Νέες συνδέσεις ανά λεπτό ανά διεύθυνση
<code>per_ip_ban_s</code>	600	Δευτερόλεπτα απόρριψης μετά από παραβίαση ρυθμού
<code>idle_per_ip</code>	8	Αδρανείς συνδέσεις, πριν από αίτημα, ανά διεύθυνση
<code>idle_total</code>	512	Αδρανείς συνδέσεις σε όλες τις διευθύνσεις μαζί

Θέτοντας οποιοδήποτε κλειδί σε 0 απενεργοποιείται ο συγκεκριμένος έλεγχος.

Τα τρία που δεν είναι όρια

Το `[limits]` περιέχει επίσης τα `audio`, `waterfall` και `events`, κληρονομιά από το upstream PhantomSDR, και **κανένα από τα τρία δεν επιβάλλεται**. Τα `waterfall` και `events` δεν διαβάζονται από κανέναν κώδικα. Το `audio` διαβάζεται σε ένα μόνο σημείο: στέλνεται ως `max_users` στο JSON εγγραφής προς τους καταλόγους του `[websdr] register_urls`, οπότε είναι ο αριθμός που δείχνουν το `sdr-list.xyz` και οι υπόλοιποι, και δεν κάνει απολύτως τίποτα όταν `register_online=false`. Αν θέλετε όριο στους ακροατές, το `per_ip` παραπάνω είναι αυτό που δρα.

Άλλο ένα, κάτω από το `[server]`

```
[server]
min_client_version=0
```

Δεν είναι όριο αλλά απόρριψη, και μοιράζεται τον ίδιο κωδικό κλεισίματος, οπότε ανήκει εδώ. Οι σελίδες δηλώνουν την έκδοση από την οποία φορτώθηκαν ως `/audio?v=N`, και ό,τι είναι κάτω από το `min_client_version` διώχνεται με το μήνυμα «αυτή η σελίδα είναι παλιά — ανανεώστε την».

Υπάρχει επειδή ο διακομιστής δεν μπορεί να φτάσει σε JavaScript που ήδη τρέχει στον browser κάποιου. Μια καρτέλα κρατάει τον κώδικα που φόρτωσε μέχρι κάποιος να την ανανεώσει, οπότε μια αλλαγή στη σελίδα φτάνει μόνο σε όσους τύχει να ανανεώσουν — και ένας σταθμός που μόλις άλλαξε τη συμπεριφορά των συνεδριών μπορεί να χρειάζεται τις παλιές σελίδες να φύγουν τώρα, όχι κάποτε.

Ελέγχεται μόνο το `/audio`, γιατί εκεί ζει μια συνεδρία, ενώ το `loorback` και οι διαδρομές Kiwi εξαιρούνται ώστε το `autorun tap` και οι πελάτες Kiwi να μένουν ανέπαφοι. Το `θ` το απενεργοποιεί και είναι η προεπιλογή. Μπορείτε να το αφήσετε ενεργό και αφού στραγγίσουν οι παλιές σελίδες: ένας τρέχων πελάτης `diversity` που απορρίπτεται σε σκέτο `/audio` ξαναδοκιμάζει με τον δείκτη, οπότε απορρίπτονται πλέον μόνο πελάτες `diversity` με παλαιότερη έκδοση του PhantomSDR-Plus και εργαλεία τρίτων που ανοίγουν το `/audio` χωρίς `?v=`. Βάλτε το πάλι σε `θ` αν κάποιος άλλος σταθμός αναφέρει ότι δεν μπορεί πια να σας χρησιμοποιήσει ως συνεργάτη `diversity`.

7. Τι βλέπει ένας ακροατής που απορρίφθηκε

Τίποτα δεν επανασυνδέεται ποτέ

Ξεκινήστε από εδώ, γιατί αυτό διέπει όλα τα υπόλοιπα: **μια σύνδεση `/audio` που πέφτει τερματίζει τη συνεδρία**. Δεν υπάρχει αυτόματη επανασύνδεση, εκ σχεδιασμού.

Η επανασύνδεση είναι το σωστό ένστικτο για μια εφαρμογή συνομιλίας και το λάθος για έναν δέκτη. Τα `/waterfall` και `/events` δεν επέστρεφαν μαζί της, οπότε μια συνεδρία που ξαναδοκίμαζε ήταν μια ζωντανή σύνδεση ήχου βιδωμένη πάνω σε παγωμένο καταρράκτη· και το σταμάτημα του διακομιστή δεν τον άδειαζε πια, γιατί κάθε ακροατής επέστρεφε λίγα δευτερόλεπτα μετά, γεμίζοντας τη λίστα χρηστών με συνεδρίες όπου δεν ήταν πραγματικά κανείς. Η σελίδα σωπαίνει και ο ακροατής την ξαναφορτώνει — που είναι ακριβώς ό,τι έκανε αυτός ο δέκτης πριν προστεθεί ποτέ επανασύνδεση.

Για τα όρια αυτού του εγχειριδίου αυτό αφαιρεί και μια παγίδα: απέναντι σε όριο ρυθμού, μια αυτόματη επανάληψη θα παρέτεινε ακριβώς τον αποκλεισμό που προσπαθούσε να παρακάμψει.

Οι δύο κωδικοί κλεισίματος

Κωδικός	Σημασία	Ποιος τον στέλνει
4003	απόρριψη — πάνω από όριο ανά διεύθυνση, ή παλιά σελίδα	<code>spectrumserver</code>
4001	αποβολή από τον διαχειριστή	ο <code>spectrumserver</code> , όταν ο πίνακας καλεί το <code>/~kick</code>

Και οι δύο είναι στην ιδιωτική περιοχή 4000–4999, οπότε κανείς δεν μπερδεύεται με κωδικό πρωτοκόλλου, και και οι δύο θεωρούνται οριστικοί από κάθε σελίδα. Το κείμενο της αιτίας ταξιθεύει στο πλαίσιο κλεισίματος και όχι ως μήνυμα, γιατί το πρώτο πλαίσιο στο `/audio` είναι δεσμευμένο για τις ρυθμίσεις του δέκτη και οτιδήποτε σταλεί πριν από αυτό θα χαλούσε κάθε κανονική σύνδεση.

Στη σελίδα

Απόρριψη κατά τη φόρτωση της σελίδας — ο ακροατής δεν ξεκινάει καν, οπότε και οι δύο σελίδες εξηγούνται. Ο υπολογιστής δείχνει ένα πλαίσιο: «Αυτός ο δέκτης απέρριψε τη σύνδεση», τον λόγο, και τη σημείωση ότι όλοι όσοι μοιράζονται μια σύνδεση στο διαδίκτυο μετράνε ως μία διεύθυνση. Το `/mobile` δείχνει το ίδιο στην περιοχή ειδοποιήσεών του.

Απόρριψη ή αποβολή στη μέση της συνεδρίας — η σελίδα του υπολογιστή απλώς σταματάει, σκόπιμα χωρίς μήνυμα: ο βρόχος σχεδίασης του καταρράκτη παύει και τίποτα άλλο. Το `/mobile` δείχνει μία γραμμή που λέει στον ακροατή να ανανεώσει, εκτός από την περίπτωση αποβολής, όπου κι αυτό μένει σιωπηλό.

Αυτή η σιωπή μετά από αποβολή είναι σκόπιμη. **Η αποβολή είναι ο διαχειριστής που τερματίζει μια συνεδρία, όχι τιμωρία** — δεν ανακοινώνεται τίποτα, και όποιος θέλει να ξαναμπει φορτώνει τη σελίδα και είναι ένας συνηθισμένος ακροατής ένα δευτερόλεπτο αργότερα. Το μόνο που εξασφαλίζει ο κωδικός 4001 είναι ότι ο browser ξέρει πως αυτό δεν ήταν πεσμένη σύνδεση, ώστε η αποβολή να πιάνει αντί να αναιρείται μόνη της.

Την αποβολή την κάνει ο ίδιος ο `spectrumserver`, όχι το `proxy.py`: οι ακροατές συνδέονται απευθείας στη θύρα του δέκτη, οπότε ο `proxy` δεν κατέχει καμία από τις συνδέσεις τους και η δική του αποβολή δεν φτάνει σε κανέναν. Ο πίνακας καλεί το `/~kick?ip=&secs=`, που δέχεται μόνο `loorback` και ελέγχει τον πραγματικό συνομιλητή TCP και όχι το `X-Forwarded-For`, το οποίο το στέλνει ο πελάτης. Το `secs` είναι εξ ορισμού μηδέν — κανένας αποκλεισμός — αλλά δέχεται διάρκεια αν θέλετε να μείνει η διεύθυνση έξω για κάποιο διάστημα, και αυτό χρησιμοποιεί τον ίδιο μηχανισμό αποκλεισμού με το `per_ip_rate`, οπότε η επανασύνδεση απορρίπτεται στην πόρτα.

8. Διαβάζοντας τα αρχεία καταγραφής και τους μετρητές

Ο διακομιστής

```
grep Refused spectrumserver.log
```

```
Refused /audio from 203.0.113.5: too many simultaneous connections
Refused /audio from 198.51.100.7: too many connection attempts
```

Αν δεν υπάρχει τίποτα εκεί, δεν απορρίπτεται κανείς. Η καταγραφή περιορίζεται σε το πολύ μία γραμμή ανά διεύθυνση κάθε πέντε δευτερόλεπτα — χωρίς αυτό, μια επίθεση θα μετατρεπόταν σε απεριόριστες εγγραφές στον δίσκο, που είναι απλώς ένας πιο αργός τρόπος να πέσει ο σταθμός.

Ο πυρήνας

```
sudo ./setup-firewall.sh --status
```

Διαβάστε το **counter packets N** σε κάθε κανόνα· αυτός είναι ο αριθμός που πράγματι απορρίφθηκε. Όλα μηδενικά σημαίνει ότι δεν έχει μπλοκαριστεί τίποτα και τα όριά σας είναι άνετα χαλαρά.

Μην ανησυχήσετε από τις λίστες `elements = { ... }` στα σύνολα πάνω από τους κανόνες. Αυτοί είναι οι συνηθισμένοι ακροατές σας που παρακολουθούνται σε σχέση με τα όρια — μια εγγραφή εμφανίζεται στην πρώτη σύνδεση μιας διεύθυνσης και λήγει ένα δυο λεπτά αργότερα. Η παρακολούθηση δεν είναι μπλοκάρισμα.

9. Επιλέγοντας τους αριθμούς σας

Δεν χρειάζεται να μαντέψετε· τα ίδια σας τα αρχεία καταγραφής θα σας πουν. Αυτή είναι η μέτρηση που όρισε το `per_ip=3` εδώ, πάνω σε μία εβδομάδα από `logs/users_*.jsonl`:

Ταυτόχρονες συνεδρίες από μία διεύθυνση	Πλήθος διεθύνσεων
1	682
2	54
3	13
πάνω από 3	8

758 διαφορετικές διευθύνσεις σε επτά μέρες, και ένα όριο τριών θα άγγιζε οκτώ από αυτές. Η πλημμύρα καθόταν στις 181.

Για να την επαναλάβετε στον δικό σας σταθμό, αντιστοιχίστε τα ids συνεδρίας ανά διεύθυνση στο `logs/users_*.jsonl` και πάρτε τη μέγιστη επικάλυψη. Προσοχή: το αρχείο καταγράφει γεγονότα `tune` και `disconnect` αλλά δεν έχει γεγονός `connect`, οπότε η αρχή μιας συνεδρίας πρέπει να συναχθεί από το πρώτο της `tune`.

Θυμηθείτε το NAT. Ένας σύλλογος, ένα σχολείο, ένα γραφείο ή ένας πάροχος κινητής εμφανίζονται ως μία διεύθυνση, και όλοι από πίσω μοιράζονται ένα όριο. Αυτό είναι εγγενές σε κάθε όριο κλειδωμένο σε διεύθυνση· η μόνη άμβλυνση είναι να διαλέξετε έναν αριθμό που σας βολεύει. Αν κάποιος ακροατής αναφέρει ποτέ ότι δεν μπορεί να συνδεθεί, αυτό είναι το πρώτο που πρέπει να υποψιαστείτε — ελέγξτε το `grep Refused` πριν από οτιδήποτε άλλο.

10. Παγίδες που αξίζει να ξέρετε

Ποτέ προθεσμία σε αδρανή σύνδεση

Δεν υπάρχει πια ρύθμιση `handshake_s`, και ο λόγος είναι αυτός. Μια προθεσμία 30 δευτερολέπτων σε αδρανείς συνδέσεις μοιάζει η προφανής άμυνα απέναντι σε `slowloris`. Φτιάχτηκε, ενεργοποιήθηκε και μέσα σε ένα λεπτό έβγαλε αυτό:

```
Idle deadline: closed 1 connection(s) idle past 30s: 192.87.173.88
```

Το 192.87.173.88 είναι το `etgd-websdr.ewi.utwente.nl` — ο ίδιος ο υπολογιστής επανάκλησης του **websdr.org**. Συνδέεται πίσω στον διακομιστή σας και μετά μένει αδρανής, για αρκετά πάνω από 30 δευτερόλεπτα, πριν στείλει το `GET /~~orgstatus`. Η προθεσμία τον έκοβε κάθε φορά και η εγγραφή σταμάτησε. Η ρύθμιση αφαιρέθηκε αντί να μείνει ως επιλογή, γιατί ένα αυτοπυροβόλημα που χαλάει την καταχώρησή σας δεν αξίζει τις εξήντα γραμμές που το υλοποιούν.

Το ίδιο σκεπτικό ισχύει για το `open_handshake_timeout` του `websocketpp`, που υπάρχει ακόμη. Η τιμή των δέκα λεπτών στο `spectrumserver.cpp` μοιάζει παράλογη και είναι κρίσιμη: είναι αυτή που επιτρέπει στην επανάκληση του `websdr.org` να μείνει αδρανής. Υπάρχει και δεύτερος λόγος — ο χειριστής του `/~~orgstatus` απαντά από νήμα που κρατάει ένα `dup()` της σύνδεσης, και το `websocketpp` τερματίζει μια σύνδεση που έληξε με `shutdown()`, κλήση σε επίπεδο `socket` που φτάνει μέσα από το αντίγραφο και θα χάλαγε την επανάκληση ακόμη κι αν ο χρόνος αδράνειας δεν ήταν θέμα.

Αν χρειάζεται να περιορίσετε τις σιωπηλές συνδέσεις, περιορίστε το πλήθος τους, όχι την ηλικία τους. Αυτό κάνουν τα `idle_per_ip` και `idle_total`.

Ένα αρχείο `nftables` πρέπει να είναι ιδεμοτικό

Το `nft -f` **προσθέτει** σε έναν πίνακα που ήδη υπάρχει, αντί να τον αντικαθιστά. Ένα αρχείο κανόνων χωρίς πρόλογο `delete table` λοιπόν διπλασιάζει ολόκληρη την αλυσίδα σε κάθε επαναφόρτωση — που είναι ακριβώς ό,τι κάνει μια αφελής μονάδα `systemd` σε κάθε επανεκκίνηση υπηρεσίας. Το `setup-firewall.sh` παράγει τον πρόλογο για εσάς· αν γράψετε δικούς σας κανόνες, κάντε το ίδιο.

```
sudo ./setup-firewall.sh --status | grep -c dport # πρέπει να δίνει 8, όχι 16
```

11. Απενεργοποίηση όλων

Τα όρια του διακομιστή: βάλτε τα κλειδιά σε 0, ή σβήστε τα από το `.toml` σας, και επανεκκινήστε τον δέκτη. Δεν υπάρχει άλλη κατάσταση· τίποτα δεν επιβιώνει μιας επανεκκίνησης.

Το τείχος προστασίας:

```
sudo ./setup-firewall.sh --remove
```

Αυτό σβήνει τον πίνακα, αφαιρεί το αποθηκευμένο σύνολο κανόνων και απενεργοποιεί την υπηρεσία εκκίνησης. Το μηχάνημά σας επιστρέφει ακριβώς στην κατάσταση που ήταν πριν — που, στις περισσότερες εγκαταστάσεις, σημαίνει καθόλου τείχος προστασίας. Αξίζει να το σκεφτείτε πριν το αφαιρέσετε.

Πολλοί δέκτες

MULTI_RECEIVER.md

Πολλοί δέκτες — Εγχειρίδιο διαχειριστή

Δύο ή περισσότεροι δέκτες σε έναν υπολογιστή, με επιλογή δέκτη πάνω στη σελίδα.

Από την v5.0.0 ένας υπολογιστής PhantomSDR-Plus μπορεί να τρέχει ταυτόχρονα πολλούς δέκτες — για παράδειγμα ένα RX-888 για τα βραχέα και ένα RTL-SDR για τα 2 m — και οι ακροατές περνούν από τον έναν στον άλλο με κουμπιά στην κεφαλίδα της σελίδας, όπως το OpenWebRX προσφέρει τα προφίλ του. Κάθε δέκτης κρατά τον δικό του καταρράκτη, chat, σημάνσεις, λίστα ακροατών και στοιχεία σταθμού· κανείς σε έναν δέκτη δεν ενοχλείται ποτέ από κάποιον σε άλλον.

Βιάζεστε; Ένας σταθμός με έναν δέκτη δεν χρειάζεται τίποτα από αυτά και δεν βλέπει καμία αλλαγή. Για δεύτερο δέκτη: δημιουργήστε το `instances/<όνομα>/` με δικό του `config.toml`, ξεκινήστε τον με `INSTANCE=<όνομα> ./start-<δέκτης>.sh`, καταχωρίστε και τους δύο δέκτες στο `receivers.toml` και επανεκκινήστε τον proxy. Το παράδειγμα στην [ενότητα 3](#) τα δείχνει γραμμή προς γραμμή.

Ο ευκολότερος τρόπος: τρέξτε το `./add-receiver.sh` (τα σενάρια εγκατάστασης το προσφέρουν στο τέλος). Ρωτά ποιον δέκτη και τι θα καλύπτει, εγκαθιστά τον οδηγό του, γράφει όλα όσα περιγράφονται στην [ενότητα 3](#) και στην [ενότητα 4](#), και προσφέρεται να ξεκινήσει τον δέκτη και να επανεκκινήσει τον proxy. Το υπόλοιπο εγχειρίδιο εξηγεί τι κάνει, για όταν θέλετε να αλλάξετε κάτι με το χέρι.

Περιεχόμενα

1. Πώς λειτουργεί
2. Δύο τρόποι δημοσίευσης των δεκτών
3. Προσθήκη δεύτερου δέκτη
4. `receivers.toml`
5. Εκκίνηση και διακοπή
6. Τι βλέπουν οι ακροατές
7. Το S-meter πάνω από τα 30 MHz
8. Ασφάλεια
9. Πόσοι δέκτες χωράνε
10. Παγίδες που αξίζει να γνωρίζετε

1. Πώς λειτουργεί

Κάθε δέκτης είναι ένας πλήρης, δικός του spectrumserver, με δικό του SDR, δική του ρύθμιση και δική του εσωτερική θύρα. Δεν μοιράζονται ποτέ δείγματα, ρυθμίσεις ή ακροατές. Αυτό που τους ενώνει είναι το `proxy.py`, ο reverse proxy που ήδη σερβίρει τον πίνακα διαχείρισης: διαβάζει το `receivers.toml` και στέλνει κάθε αίτημα σε έναν δέκτη.

```

ακροατές → proxy.py → :8900 spectrumserver RX-888 (HF, ο κύριος δέκτης)
                    → 127.0.0.1:9002 spectrumserver RTL-SDR (2 m, instance "vhf")

```

Ο proxy αποφασίζει με τα εξής, με αυτή τη σειρά:

1. `?rx=<id>` στη διεύθυνση — `http://your.host:8899/?rx=vhf`. Η σελίδα το προσθέτει σε κάθε socket, ερώτημα και σύνδεσμο που ανοίγει, ώστε μια καρτέλα να μένει στον δέκτη της.
2. Το cookie `rx`, που ορίζει ο proxy όποτε ένα αίτημα φέρει έγκυρο `?rx=`.
3. Την κεφαλίδα Host, αν ένας δέκτης έχει αυτό το όνομα στα `hostnames`.
4. Αλλιώς τον προεπιλεγμένο δέκτη.

Κύριος δέκτης είναι αυτός που ξεκινά με τον συνηθισμένο τρόπο (`./start-rx888mk2.sh`). Κάθε άλλος δέκτης είναι ένα **επώνυμο instance**: ξεκινά με `INSTANCE=<όνομα>`, ρυθμίζεται στο `instances/<όνομα>` και έχει δικό του log (`logwebsdr-<όνομα>.txt`), log διακομιστή, κλείδωμα watchdog και FIFO. Οι διεργασίες ξεχωρίζουν από μια ετικέτα `PHANTOMSDR_INSTANCE` στο περιβάλλον τους, όχι από το όνομά τους· έτσι δύο δέκτες μπορούν να τρέχουν το ίδιο ακριβώς πρόγραμμα — δύο `rtl_sdr`, ή `rx_sdr` για ένα RSP1A και ένα Airspy — και η επανεκκίνηση του ενός δεν αγγίζει ποτέ τον άλλον.

2. Δύο τρόποι δημοσίευσης των δεκτών

A — ο κύριος δέκτης κρατά τη δική του θύρα. Ο κύριος δέκτης μένει ακριβώς όπου ήταν (π.χ. : 8900), οι ακροατές του δεν καταλαβαίνουν τίποτα, και οι άλλοι δέκτες φτάνονται μέσω της θύρας του proxy (`proxy_port` στο `admin_config.json`, π.χ. : 8899) με `?rx=`. Τίποτα δεν μετακινείται και τίποτα δεν βγαίνει από τον αέρα· σημαίνει όμως δύο δημόσιες θύρες.

```

HF: http://your.host:8900/
2 m: http://your.host:8899/?rx=vhf

```

B — μία δημόσια θύρα για όλα. Ο proxy αναλαμβάνει τη δημόσια θύρα και ο κύριος δέκτης μετακινείται σε εσωτερική. Ορίστε `[front] port` στο `receivers.toml`, μετακινήστε το `[server] port` του κύριου δέκτη και ορίστε σε αυτόν `host = "127.0.0.1"`, βάλτε στο `public_port` του `admin_config.json` τη νέα εσωτερική θύρα και προσθέστε `public_port = <η δημόσια θύρα>` κάτω από το `[websdr]`, ώστε ο κατάλογος να συνεχίσει να ανακοινώνει τη σωστή. Τότε όλα βρίσκονται πίσω από μία θύρα· το κόστος είναι μια σύντομη διακοπή κατά τη μετακίνηση.

```

HF: http://your.host:8900/
2 m: http://your.host:8900/?rx=vhf

```

Η επιλογή δέκτη δουλεύει το ίδιο και στις δύο περιπτώσεις. Η διάταξη A είναι η ασφαλέστερη αρχή· η B μπορεί να ακολουθήσει αργότερα χωρίς καμία αλλαγή στον δεύτερο δέκτη.

3. Προσθήκη δεύτερου δέκτη

Το `add-receiver.sh` κάνει όλα τα παρακάτω βήματα — τρέξτε το και απαντήστε στις ερωτήσεις του· εγκαθιστά και τον οδηγό (`setup-rtlsdr.sh` για RTL-SDR, το αντίστοιχο `setup-*.sh` για τους άλλους), διαλέγει ελεύθερη εσωτερική θύρα και ονομάζει τον δέκτη ανάλογα με το τι καλύπτει. Τα βήματα γράφονται εδώ αναλυτικά ώστε να μπορείτε να ελέγξετε τη δουλειά του ή να την αλλάξετε αργότερα.

Το παράδειγμα προσθέτει ένα RTL-SDR Blog V4 για τα 2 m ως instance `vhf`, δίπλα σε ένα RX-888 που κρατά τη θύρα 8900 (διάταξη A).

1. Ο οδηγός. Το RTL-SDR Blog V4 χρειάζεται τον οδηγό της RTL-SDR Blog αντί για το πακέτο `rtl-sdr` της διανομής, και ο οδηγός DVB-T του πυρήνα πρέπει να αφήσει το `stick`. Το `install.sh` (επιλογή δέκτη 2, «RTL-SDR Blog V4: ναι») κάνει και τα δύο. Ελέγξτε με `rtl_test -t`, που πρέπει να γράφει `RTL-SDR Blog V4 Detected`.

2. Ο φάκελος του instance. Ό,τι ανήκει στον δέκτη βρίσκεται στο `instances/vhf/`, που δεν γίνεται ποτέ commit και δεν το αντικαθιστά καμία ενημέρωση:

```
instances/vhf/
├─ config.toml           η ρύθμιση του spectrumserver του (υποχρεωτικό)
├─ instance.env         προαιρετικό: παράμετροι δέκτη και δέσμευση πυρήνων
├─ markers.json         οι δικές του σημάνσεις
└─ www/                 δικά του αντίγραφα των αρχείων σταθμού της σελίδας
    ├─ site_information.json
    └─ wf-message.json
```

Ο `spectrumserver` τρέχει με φάκελο εργασίας το `instances/vhf/`, οπότε και το ιστορικό chat, οι σημάνσεις, το FFTW wisdom και το `logs/` είναι δικά του.

3. config.toml. Ξεκινήστε από το `config-rtl.toml` και αλλάξτε:

```
[server]
port=9002                # η δική του εσωτερική θύρα
host="127.0.0.1"         # προσβάσιμος μόνο μέσω του proxy
html_root="www/"         # τα λίγα δικά του αρχεία...
html_fallback_root="../../frontend/dist/" # ...και η κοινή σελίδα για τα υπόλοιπα

[input]
sps=2400000              # ίδιο με το -s στο instance.env
frequency=145000000       # ίδιο με το -f στο instance.env

[input.defaults]
frequency=145500000
modulation="FM"

[kiwi_emulation]
enabled = false          # δείτε τις παγίδες παρακάτω
```

Το `html_fallback_root` επιτρέπει στο instance να κρατά μόνο τα δικά του `site_information.json` και `wf-message.json`, ενώ όλα τα άλλα — η ίδια η χτισμένη σελίδα — έρχονται από το `frontend/dist`. Έτσι ένα χτίσιμο του frontend φτάνει σε κάθε δέκτη. Ένα αντίγραφο του `frontend/dist` με symlink δεν λειτουργεί: ο διακομιστής αρνείται αρχεία που καταλήγουν έξω από τον δικό του φάκελο.

4. instance.env. Το διαβάζει το σενάριο εκκίνησης μετά τις δικές του ρυθμίσεις:

```
RX_ARGS="-f 145000000 -s 2400000 -g 29.7 -" # τα -f και -s ίδια με το config.toml
SPECTRUM_CORES=8-11                       # μακριά από τους πυρήνες του κύριου δέκτη
```

5. Τα στοιχεία της σελίδας του. Αντιγράψτε το `frontend/site_information.json` στο `instances/vhf/www/` και διορθώστε: `siteReceiver`, `siteAntenna`, `siteSDRBaseFrequency` και `siteSDRBandwidth` (αυτά ορίζουν ποια κουμπιά μπάντας εμφανίζονται), καθώς και τα κλειδιά της [ενότητας 6](#).

6. Εκκίνηση.

```
INSTANCE=vhf ./start-rtl.sh
```

7. Καταχωρίστε τον στο `receivers.toml` και επανεκκινήστε τον proxy (`sudo systemctl restart phantomsdr-proxy`). Ο δεύτερος δέκτης βρίσκεται τώρα στο `http://your.host:8899/?rx=vhf`, και οι δύο σελίδες δείχνουν την επιλογή.

4. `receivers.toml`

Αντιγράψτε το `receivers.toml.example` σε `receivers.toml`. Χωρίς αυτό το αρχείο ο proxy εξυπηρετεί έναν μόνο δέκτη, όπως πάντα.

```
# [front]
# port = 8900          # μόνο διάταξη B: ο proxy ακούει και σε αυτή τη δημόσια θύρα

[[receiver]]
id      = "hf"
name    = "HF 0-30 MHz (RX-888 MkII)"
port    = 8900
default = true
launcher = "start-rx888mk2.sh"
url      = "http://your.host:8900/"

[[receiver]]
id      = "vhf"
name    = "2 m (RTL-SDR Blog V4)"
port    = 9002
launcher = "start-rtl.sh"
instance = "vhf"
url      = "http://your.host:8899/?rx=vhf"
```

Κλειδί	Σημασία
<code>id</code>	Σύντομο όνομα για το <code>?rx=</code> · γράμματα, ψηφία, <code>-</code> και <code>_</code>
<code>name</code>	Το κείμενο στο κουμπί του δέκτη
<code>port</code>	Το <code>[server] port</code> του spectrumserver του
<code>host</code>	Προαιρετικό· πού τρέχει αυτός ο spectrumserver (προεπιλογή <code>127.0.0.1</code>)
<code>default</code>	Ο δέκτης για αιτήματα που δεν ονομάζουν κανέναν· το πολύ ένας
<code>launcher</code>	Το <code>start-*.sh</code> του, για το <code>start-all.sh</code> · παραλείψτε το για δέκτη που δεν ξεκινά αυτός ο υπολογιστής
<code>instance</code>	Το όνομα του instance· παραλείψτε το για τον κύριο δέκτη
<code>url</code>	Πού στέλνει τον ακροατή το κουμπί του· χωρίς αυτό, <code>/?rx=<id></code> στον proxy
<code>hostnames</code>	Προαιρετικά ονόματα DNS που οδηγούν κατευθείαν σε αυτόν τον δέκτη

Ο proxy σερβίρει τη λίστα και ως `/receivers.json` (ids, ονόματα και συνδέσμους — ποτέ τις εσωτερικές θύρες)· αυτό διαβάζει η επιλογή δέκτη.

5. Εκκίνηση και διακοπή

Εντολή	Τι κάνει
<code>./start-rx88mk2.sh</code>	Ξεκινά ή επανεκκινεί τον κύριο δέκτη
<code>INSTANCE=vhf ./start-rtl.sh</code>	Ξεκινά ή επανεκκινεί τον δέκτη vhf
<code>./start-all.sh</code>	Ξεκινά κάθε δέκτη του <code>receivers.toml</code> που έχει <code>launcher</code>
<code>./stop-websdr.sh vhf</code>	Σταματά μόνο τον vhf
<code>./stop-websdr.sh main</code>	Σταματά μόνο τον κύριο δέκτη
<code>./stop-websdr.sh</code>	Σταματά κάθε δέκτη, όπως πάντα

Όσο ένας δέκτης είναι σταματημένος, το κουμπί του εξαφανίζεται από την επιλογή — ο proxy καταγράφει μόνο τους δέκτες που απαντούν, και οι ανοιχτές σελίδες ξαναδιαβάζουν τη λίστα μία φορά το λεπτό — και επανέρχεται όταν ο δέκτης ξεκινήσει ξανά· αν μείνει να τρέχει μόνο ένας δέκτης, η γραμμή *Receivers*: εξαφανίζεται εντελώς. Ένας σταματημένος δέκτης όμως επανέρχεται μόνος του όποτε ξεκινούν όλοι οι δέκτες: το `./start-all.sh`, το **Restart** του πίνακα διαχείρισης και μια επανεκκίνηση από τη θερμική προστασία ξεκινούν όλους όσους υπάρχουν στο `receivers.toml`. Για να μείνει κάποιος μόνιμα κλειστός, αφαιρέστε (ή σχολιάστε) το μπλοκ `[[receiver]]` του και επανεκκινήστε τον proxy.

Στον πίνακα διαχείρισης ορίστε το **Default start script** σε `start-all.sh`. Το Stop του ήδη σταματά κάθε δέκτη, ενώ το Restart και η θερμική προστασία χρησιμοποιούν το σενάριο εκκίνησης· με το `start-all.sh` επαναφέρουν κάθε δέκτη αντί μόνο τον κύριο.

Ο βοηθός RADE ανήκει μόνο στον κύριο δέκτη. Ένα επώνυμο instance δεν τον ξεκινά ούτε τον σταματά ποτέ.

6. Τι βλέπουν οι ακροατές

Η επιλογή δέκτη. Σε σταθμό με περισσότερους από έναν δέκτες η κεφαλίδα της σελίδας αποκτά δική της γραμμή, *Receivers*·, με ένα κουμπί ανά δέκτη· ο τρέχων εμφανίζεται κίτρινος. Η σελίδα `/mobile` έχει τα ίδια κουμπιά ως δεύτερη σειρά στην επάνω μπάρα της, και ανοίγουν τη σελίδα `/mobile` του άλλου δέκτη.

Τα δικά του στοιχεία για κάθε δέκτη. Μια σελίδα που ανοίγει σε δεύτερο δέκτη φορτώνει πριν ξεκινήσει το `site_information.json` εκείνου του δέκτη· έτσι τα κουμπιά μπάντας, η αρχική συχνότητα, η λίστα ακροατών, τα στοιχεία σταθμού και οι σύνδεσμοι Receiver και Antenna είναι όλα δικά του. Μια απλή επίσκεψη στον κύριο δέκτη δεν κάνει κανένα επιπλέον αίτημα.

Τα κλειδιά στο `site_information.json` για αυτό:

Κλειδί	Σημασία
<code>siteReceiverId</code>	Ο δέκτης στον οποίο ανήκει αυτή η σελίδα (το <code>id</code> του)· παραλείψτε το στον κύριο δέκτη
<code>siteReceiversList</code>	Από πού διαβάζει η επιλογή τη λίστα, π.χ. <code>http://your.host:8899/receivers.json</code> · χρειάζεται σε δέκτη του οποίου η σελίδα δεν σερβίρεται μέσω του proxy
<code>siteReceiverURL</code>	Πού οδηγεί το όνομα <i>Receiver</i> στο <i>Open Additional Info</i>
<code>siteAntennaURL</code>	Πού οδηγεί το όνομα <i>Antenna</i> · το <code>""</code> δείχνει το όνομα χωρίς σύνδεσμο

Πότε η επιλογή μένει κρυφή. Η επιλογή δείχνει μόνο μια λίστα που αναφέρει τον host με τον οποίο άνοιξε η σελίδα. Αν ανοίξει με διεύθυνση LAN (`http://192.168.1.10:8900/`) μένει κρυφή· ως `http://`

`your.host:8900/` εμφανίζεται. Αυτό είναι σκόπιμο: ένας σταθμός που αντέγραψε το `site_information.json` κάποιου άλλου χωρίς να το διορθώσει δεν δείχνει ποτέ τους δέκτες εκείνου ως δικούς του.

7. Το S-meter πάνω από τα 30 MHz

Το πρότυπο της IARU Region 1 ορίζει το S9 στα -73 dBm κάτω από τα 30 MHz και στα **-93 dBm πάνω από αυτά**, με έξι dB ανά μονάδα S και στις δύο περιπτώσεις. Τα S-meter (η αναλογική βελόνα, η ψηφιακή μπάρα και η μπάρα του /mobile) ακολουθούν τη συχνότητα συντονισμού: πάνω από τα 30 MHz δείχνουν μονάδες S VHF, από κάτω όπως πριν. Οι τιμές σε dBm και dBμV δεν αλλάζουν ποτέ. Αν προτιμάτε τα όργανα να μένουν στο 0 σε άδειο κανάλι, όπως σε πομποδέκτη VHF/UHF, προσθέστε `"siteSMeterGateDb": 6` στο `site_information.json` του δέκτη (χωρίς νέο χτίσιμο): πάνω από τα **60 MHz** η βελόνα και η μπάρα ανεβαίνουν τότε μόνο όταν ένα σήμα ξεπερνά κατά τόσα dB τον θόρυβο της δικής του γειτονιάς (το ισχυρότερο σημείο της ζώνης διέλευσης έναντι του φάσματος ± 100 kHz γύρω του). Είναι **απενεργοποιημένο από προεπιλογή** — ένα βαθμονομημένο όργανο που δείχνει τον πραγματικό θόρυβο της μπάντας είναι η πιο τίμια ένδειξη. Η σελίδα /mobile, που δεν έχει καταρράκτη, εκτιμά τον θόρυβο από την ίδια τη λαμβανόμενη στάθμη.

Βαθμονομήστε τα dBm ενός δέκτη VHF με το `analog_smeter_offset` (βελόνα και αριθμοί) και το `smeter_offset` (ψηφιακή μπάρα) στο `[input]` του `config.toml` του, και μετά επανεκκινήστε αυτόν τον δέκτη. Ορίστε πρώτα την ενίσχυση: ένα RTL-SDR αναφέρει στάθμες σε σχέση με τη δική του πλήρη κλίμακα, οπότε κάθε αλλαγή ενίσχυσης μετακινεί τις ενδείξεις. Χωρίς γεννήτρια σήματος, ένα φορτίο 50 Ω στη θέση της κεραίας, σε USB με φίλτρο 2,7 kHz, πρέπει να δείχνει περίπου -136 dBm (ο θερμικός θόρυβος σε 2,7 kHz είναι $-139,7$ dBm, συν το noise figure του stick).

Για ένα RTL-SDR, το `add-receiver.sh` γράφει μια αρχική βαθμονόμηση: σταθερή ενίσχυση `-g 29.7` στο `instance.env` (η αυτόματη ενίσχυση του tuner δεν βαθμονομείται) και `analog_smeter_offset=-55`, `smeter_offset=-38`, μετρημένα σε RTL-SDR Blog V4 με αυτή την ενίσχυση και σήμα -71 dBm (63 μV). Ένα άλλο stick του ίδιου μοντέλου δείχνει συνήθως λίγα μόνο dB διαφορά· ελέγξτε το με γνωστό σήμα και βαθμονομήστε ξανά αν αλλάξετε την ενίσχυση.

8. Ασφάλεια

Η τοποθέτηση του proxy μπροστά από τους ακροατές έκλεισε δύο κενά, που διορθώθηκαν και τα δύο στην v5.0.0:

- **Το kick του διαχειριστή μέσω του proxy.** Ο `spectrumserver` επιτρέπει το `/~kick` μόνο από τον ίδιο υπολογιστή, και γι' αυτόν κάθε αίτημα που προωθεί ο proxy έρχεται από τον ίδιο υπολογιστή. Όποιος έφτανε στη θύρα του proxy μπορούσε επομένως να αποσυνδέσει και να αποκλείσει οποιονδήποτε ακροατή. Τώρα ο proxy απαντά στο `/~kick` μόνο για πελάτες στον ίδιο υπολογιστή.
- **Πλαστές διευθύνσεις πελατών.** Ο `spectrumserver` πίστευε την κεφαλίδα `X-Forwarded-For` από οποιονδήποτε, οπότε ένας επισκέπτης μπορούσε να δηλώσει `127.0.0.1` και να περάσει τα όρια ανά διεύθυνση. Τώρα η κεφαλίδα γίνεται πιστευτή μόνο από τοπικό proxy, και ο proxy πετά κάθε αντίγραφο που στέλνει ο πελάτης πριν προσθέσει το δικό του.

Δέστε κάθε δέκτη που φτάνεται μέσω του proxy στο `127.0.0.1` (`[server] host`), ώστε η θύρα του να μη χρησιμοποιείται παρακάμπτοντας τον proxy. Τα όρια ανά διεύθυνση των **Ορίων συνδέσεων** ισχύουν ανά δέκτη.

9. Πόσοι δέκτες χωράνε

Το λογισμικό δεν βάζει όριο· το υλικό βάζει. Οι στενοζωνικοί δέκτες είναι οικονομικοί: ένα RTL-SDR στα 2,4 Msps χρειάζεται περίπου 5-10 % ενός πυρήνα, μερικές δεκάδες MB μνήμης και περίπου 38 Mbit/s στο USB.

- **Το USB 2.0 είναι το συνηθισμένο όριο.** Όλες οι συσκευές USB 2 μοιράζονται έναν δίαυλο 480 Mbit/s, ακόμη και σε μπλε υποδοχή USB 3. Τρία έως τέσσερα RTL-SDR ανά δίαυλο είναι ασφαλές νούμερο· προσθέστε τα ένα-ένα και παρακολουθήστε κάθε καταρράκτη για κενά.
- **Τροφοδοσία:** πάνω από δύο sticks χρησιμοποιήστε hub με δική του τροφοδοσία.
- **Ένας δεύτερος ευρυζωνικός δέκτης** (άλλο RX-888, ή HackRF στα 20 Msps) είναι άλλη υπόθεση: ανταγωνίζεται τον πρώτο για USB 3, CPU και GPU.
- **Οι ακροατές κοστίζουν περισσότερο από τους δέκτες.** Κάθε ακροατής κοστίζει CPU στον δέκτη του και εύρος ζώνης ανεβάσματος, οπότε μετρά περισσότερο ο συνολικός αριθμός ακροατών από τον αριθμό των δεκτών.

10. Παγίδες που αξίζει να γνωρίζετε

- **Δύο sticks του ίδιου μοντέλου έχουν τον ίδιο σειριακό αριθμό.** Κάθε RTL-SDR Blog V4 δηλώνει 00000001. Δώστε σε καθένα δικό του με `rtl_eeprom -s <σειριακός>` (με ένα μόνο stick συνδεδεμένο κάθε φορά) και ονομάστε τον στο `instance.env` (`RX_ARGS="-d <σειριακός> ..."`), αλλιώς μετά από επανεκκίνηση τα sticks μπορεί να αλλάξουν δέκτη.
- **Οι πελάτες Kiwi δεν μπορούν να διαλέξουν δέκτη.** Ένας πελάτης KiwiSDR καλεί σκέτη διεύθυνση και θύρα· μέσω του proxy καταλήγει πάντα στον προεπιλεγμένο δέκτη. Κρατήστε το `[kiwi_emulation]` κλειστό στους άλλους.
- **Τα cookies ισχύουν ανά host, όχι ανά θύρα.** Ένας επισκέπτης που χρησιμοποίησε τον δεύτερο δέκτη στο :8899 φέρνει το `rx=vhf` και στο :8900. Γι' αυτό μια σελίδα αποφασίζει ποιος δέκτης είναι από το `siteReceiverId`, ποτέ από το cookie.
- **Καταχωρίσεις σε καταλόγους.** Δώστε σε κάθε δέκτη δικό του όνομα στο `[websdr]`, και σε δέκτη πίσω από τον proxy ορίστε το `[websdr] public_port` στη θύρα που χρησιμοποιούν οι ακροατές, αλλιώς η καταχώριση διαφημίζει την εσωτερική.
- **Ένα χτίσιμο του frontend δείχνει για λίγο «Not Found».** Το `frontend/dist` ξαναχτίζεται επί τόπου, οπότε για λίγα δευτερόλεπτα κατά το χτίσιμο η σελίδα λείπει από κάθε δέκτη.
- **Τίποτα δεν ξεκινά τους δέκτες κατά την εκκίνηση του υπολογιστή,** εκτός αν το ρυθμίσετε. Μετά από επανεκκίνηση, το `./start-all.sh` τους επαναφέρει όλους.

ΚΕΦΑΛΑΙΟ 7

Διακομιστής στατιστικών

sdr-stats/README.md

Οδηγός εγκατάστασης του System Stats

Ένας ολοκληρωμένος οδηγός για την προσθήκη παρακολούθησης διακομιστή σε πραγματικό χρόνο στην εφαρμογή σας PhantomSDR χρησιμοποιώντας το αυτοματοποιημένο σενάριο εγκατάστασης.

Επισκόπηση

Αυτή η λειτουργία προσθέτει ένα κουμπί **Stats** στη διεπαφή σας PhantomSDR που εμφανίζει πληροφορίες συστήματος σε πραγματικό χρόνο:

- Χρήση CPU, πυρήνες, συχνότητα, θερμοκρασία και κορυφαίες διεργασίες
- Χρήση μνήμης (χρησιμοποιούμενη/συνολική/ποσοστό)

Τα στατιστικά ενημερώνονται αυτόματα κάθε 5 δευτερόλεπτα όσο το παράθυρο διαλόγου είναι ανοιχτό.

Δεν είναι το ίδιο με τη σελίδα «Γραφήματα» του πίνακα διαχείρισης. Αυτό το παράθυρο δείχνει στιγμιότυπο που ανανεώνεται όσο είναι ανοιχτό και δεν κρατά ιστορικό. Ο πίνακας διαχείρισης έχει ξεχωριστή σελίδα **Γραφήματα** που δειγματοληπτεί συχνότητα CPU, φόρτο, θερμοκρασία και συνδεδεμένους χρήστες κάθε 2 δευτερόλεπτα και τα σχεδιάζει για τα τελευταία 15 λεπτά έως 24 ώρες — δείτε τον [οδηγό του πίνακα διαχείρισης](#).

Προαπαιτούμενα

- **Διακομιστής Linux** (συνιστάται Ubuntu/Debian)
- **Σύνδεση στο διαδίκτυο** (για τη λήψη του Node.js και των πακέτων)
- **Πρόσβαση Sudo** (για τη ρύθμιση της υπηρεσίας systemd)
- **PhantomSDR-Plus** ήδη εγκατεστημένο και σε λειτουργία
- **Im-sensors** (συνιστάται για τη θερμοκρασία CPU Intel/AMD - εγκαθίσταται αυτόματα από το σενάριο)

Μέθοδοι εγκατάστασης

ΣΗΜΕΙΩΣΗ

Το κύριο πρόγραμμα εγκατάστασης το προσφέρει ήδη. Το `./install.sh` — και τα τέσσερα scripts των διανομών — τρέχουν το `install-stats-server.sh` από προεπιλογή στο πλαίσιο της κανονικής εγκατάστασης, οπότε σε ένα φρέσκο μηχάνημα ο stats server είναι συνήθως ήδη στη θέση του. Χρησιμοποιήστε αυτή τη σελίδα για να τον εγκαταστήσετε ξεχωριστά, ή για να αλλάξετε αργότερα θύρα, διεύθυνση ή υπηρεσία.

Επιλογή 1: Αυτοματοποιημένη εγκατάσταση (Συνιστάται)

Το αυτοματοποιημένο σενάριο τα φροντίζει όλα για εσάς.

Βήμα 1: Εκτελέστε το σενάριο

```
cd ~/PhantomSDR-Plus
chmod +x install-stats-server.sh # only if it is not already executable
./install-stats-server.sh
```

ΣΗΜΑΝΤΙΚΟ: ΜΗΝ το εκτελέσετε ως root ή με sudo. Εκτελέστε το ως ο κανονικός σας χρήστης.

Βήμα 2: Ακολουθήστε τις διαδραστικές προτροπές

Το σενάριο θα σας καθοδηγήσει μέσα από:

1. Εγκατάσταση Node.js (αν χρειάζεται)

```
△ Node.js / npm is not installed
Would you like to install Node.js 22 via nvm now? (y/n): y
```

Το σενάριο εγκαθιστά το Node.js 22 μέσω nvm, σε οποιαδήποτε διανομή Linux — χωρίς root και χωρίς πηγή apt. Κάνει την ίδια ερώτηση όταν το Node.js υπάρχει αλλά είναι παλαιότερο από 22.

2. Κατάλογος εγκατάστασης

```
Installation directory [/home/user/sdr-stats-server]:
```

Πατήστε Enter για την προεπιλογή ή καθορίστε μια προσαρμοσμένη διαδρομή.

3. Διαμόρφωση θύρας (Βελτιωμένη!)

```
Port Configuration:
  ▫ Default port: 3001
  ▫ Common alternatives: 8080, 5000, 8888
Enter port number [3001]:
```

Το σενάριο επικυρώνει τη θύρα σας:

- Ελέγχει αν είναι έγκυρος αριθμός
- Προειδοποιεί αν η θύρα < 1024 (απαιτεί δικαιώματα root)
- Επικυρώνει το εύρος (1-65535)
- Ελέγχει αν η θύρα χρησιμοποιείται ήδη
- Δείχνει ποια διεργασία τη χρησιμοποιεί αν είναι κατειλημμένη
- Σας επιτρέπει να επιλέξετε διαφορετική θύρα

Παραδείγματα σεναρίων:**Χρήση προεπιλεγμένης θύρας:**

```
Enter port number [3001]: ↵
✓ Port 3001 selected
```

Επιλογή προσαρμοσμένης θύρας:

```
Enter port number [3001]: 8080
✓ Port 8080 selected
```

Θύρα ήδη σε χρήση:

```

Enter port number [3001]: 3001
△ Port 3001 is currently in use
Process using port 3001:
node 12345 user 20u IPv6 123456 TCP *:3001 (LISTEN)
Choose a different port? (y/n): y
Enter port number [3001]: 3002
✓ Port 3002 selected

```

Μη έγκυρη θύρα:

```

Enter port number [3001]: abc
x Port must be a number
Enter port number [3001]: 99999
x Port must be between 1 and 65535
Enter port number [3001]: 3001
✓ Port 3001 selected

```

4. Διεύθυνση διακομιστή

```

What is your server's public address?
Examples: Your_site_IP, 192.168.1.100, localhost
Server address: Your_site_IP

```

Εισαγάγετε τον δημόσιο τομέα ή τη διεύθυνση IP του διακομιστή σας.

5. Επιβεβαίωση

```

Please confirm your settings:
Installation directory: /home/user/sdr-stats-server
Port: 3001
Server address: Your_site_IP
Stats URL will be: http://Your_site_IP:3001
Continue with these settings? (y/n): y

```

6. Ρύθμιση υπηρεσίας Systemd (Προαιρετικό)

```

Would you like to set up the server as a system service (auto-start on boot)? (y/n): y

```

Συνιστάται: Επιλέξτε **y** για να ξεκινά ο διακομιστής αυτόματα κατά την εκκίνηση.

Βήμα 3: Η εγκατάσταση ολοκληρώθηκε! ✓

Το σενάριο θα εμφανίσει μια σύνοψη:

```

Installation Complete! ✓
Open the port in the router!

```

```

Configuration Summary:
Installation: /home/user/sdr-stats-server
Port: 3001
Server: Your_site_IP
API URL: http://Your_site_IP:3001/api/system-stats

```

Διαμορφώστε την εφαρμογή σας PhantomSDR

Αφού ολοκληρωθεί το σενάριο, πρέπει να ενημερώσετε την εφαρμογή σας Svelte.

Βήμα 1: Ενημερώστε το `site_information.json`

Δεδομένου ότι το `site_information.json` έχει ήδη επεξεργαστεί κατά την αρχική ρύθμιση, χρειάζεται μόνο να προσθέσετε τη νέα γραμμή `siteStats` με τη θύρα σας:

```
{
  "siteSysop": "your name or callsign",
  "siteSysopEmailAddress": "mail@mail.net",
  "siteGridSquare": "QTH locator",
  "siteCity": "City Country",
  "siteInformation": "https://github.com/svlbtl/PhantomSDR-Plus",
  "siteHardware": "Hardware you are using, ",
  "siteSoftware": "Software you are using",
  "siteReceiver": "Receiver model",
  "siteAntenna": "Receiving Antenna.",
  "siteNote": "This is a bright new open-source WebSDR project, under active development.",
  "siteIP": "http://Your_site_IP:port",
  "siteStats": "http://Your_site_IP:3001", ← ΠΡΟΣΘΕΣΤΕ ΑΥΤΗ ΤΗ ΓΡΑΜΜΗ (χρησιμοποιήστε τη θύρα σας)
  "siteSDRBaseFrequency": 0,
  "siteSDRBandwidth": 30000000,
  "siteRegion": 1,
  "siteChatEnabled": true
}
```

Σημείωση: Χρησιμοποιήστε τη θύρα που επιλέξατε κατά την εγκατάσταση!

Βήμα 2: Ενημερώστε το `App.svelte` (η τελευταία έκδοση του GitHub περιλαμβάνει ήδη το ενημερωμένο αρχείο)

Κάντε αυτές τις **4 αλλαγές** στο αρχείο σας `App.svelte`:

Αλλαγή 1: Βεβαιωθείτε ότι το `siteStats` δηλώνεται (γύρω από τη γραμμή 50)

```
import {
  siteSysop,
  siteSysopEmailAddress,
  siteInformation,
  siteGridSquare,
  siteCity,
  siteHardware,
  siteSoftware,
  siteReceiver,
  siteAntenna,
  siteNote,
  siteIP,
  siteStats, // ← ΠΡΟΣΘΕΣΤΕ ΑΥΤΗ ΤΗ ΓΡΑΜΜΗ
  siteSDRBaseFrequency,
  siteSDRBandwidth,
  siteRegion,
  siteChatEnabled,
} from "../site_information.json";
```

Αλλαγή 2: Ενημερώστε το αντικείμενο systemStats (γύρω από τη γραμμή 530)

```
let systemStats = {
  cpu: { usage: 0, cores: 0, temperature: null, frequency: null, topProcesses: [] }, // ←
  ΠΡΟΣΘΕΣΤΕ topProcesses: []
  memory: { used: 0, total: 0, percent: 0 }
};
```

Αλλαγή 3: Ενημερώστε το URL fetch (γύρω από τη γραμμή 541)

```
async function fetchSystemStats() {
  try {
    const response = await fetch(`${siteStats}/api/system-stats`); // ← ΑΛΛΑΞΤΕ ΑΥΤΗ ΤΗ ΓΡΑΜΜΗ
    if (response.ok) {
      systemStats = await response.json();
    } else {
      console.error('Failed to fetch system stats:', response.statusText);
    }
  } catch (error) {
    console.error('Error fetching system stats:', error);
  }
}
```

Αλλαγή 4: Προσθέστε την εμφάνιση κορυφαίων διεργασιών (στην ενότητα CPU, γύρω από τη γραμμή 3838)

Προσθέστε αυτόν τον κώδικα μετά την ενότητα θερμοκρασίας:

```
{#if systemStats.cpu.frequency}
<div style="display: flex; justify-content: space-between; margin-bottom: 0.25rem;">
  <span>Frequency:</span>
  <span style="color: #4ade80;">{systemStats.cpu.frequency.current} GHz{#if
systemStats.cpu.frequency.max} <span style="opacity: 0.7;">(max {systemStats.cpu.frequency.max})</
span>{/if}</span>
</div>
{/if}

{#if systemStats.cpu.temperature !== null}
<div style="display: flex; justify-content: space-between; margin-bottom: 0.25rem;">
  <span>Temperature:</span>
  <span style="color: {systemStats.cpu.temperature > 70 ? '#fbbf24' :
'#4ade80'};">{systemStats.cpu.temperature}°C</span>
</div>
{/if}

<!-- ΠΡΟΣΘΕΣΤΕ ΟΛΟΚΛΗΡΗ ΑΥΤΗ ΤΗΝ ΕΝΟΤΗΤΑ: -->
{#if systemStats.cpu.topProcesses && systemStats.cpu.topProcesses.length > 0}
<div style="margin-top: 0.75rem; padding-top: 0.75rem; border-top: 1px solid
rgba(255,255,255,0.1);">
  <h4 style="margin: 0 0 0.5rem 0; font-size: 0.85rem; color: rgba(0, 225, 255, 0.8);">Top
Processes:</h4>
  {#each systemStats.cpu.topProcesses as process}
    <div style="display: flex; justify-content: space-between; margin-bottom: 0.25rem; font-size:
0.85rem;">
      <span style="overflow: hidden; text-overflow: ellipsis; white-space: nowrap; max-width:
70%;">{process.name}</span>
      <span style="color: {process.cpu > 50 ? '#ef4444' : process.cpu > 25 ? '#fbbf24' :
'#4ade80'};">{process.cpu}%</span>
    </div>
  {/each}
</div>
{/if}
```

Βήμα 3: Επαναμεταγλωττίστε την εφαρμογή σας

```
cd /path/to/your/phantomsdr-app
npm ./recompile.sh
```

Δοκιμή της εγκατάστασης

Δοκιμή 1: Έλεγχος του σημείου πρόσβασης API

```
curl http://localhost:3001/api/system-stats
```

Αναμενόμενη έξοδος (JSON):

```
{
  "cpu": {
    "usage": 45.2,
    "cores": 8,
    "coresUsed": 1.3,
    "temperature": 62.5,
    "frequency": {
      "current": 3.0,
      "max": 3.9,
      "limit": 4.4
    },
  },
  "topProcesses": [
    {"name": "node", "cpu": 12.5},
    {"name": "phantomsdr", "cpu": 8.3}
  ],
  "memory": {
    "used": 8.5,
    "total": 16,
    "percent": 53
  }
}
```

Δοκιμή 2: Έλεγχος κατάστασης υπηρεσίας

Αν το εγκαταστήσατε ως υπηρεσία:

```
sudo systemctl status sdr-stats.service
```

Θα πρέπει να εμφανίσει: Active: active (running)

Δοκιμή 3: Δοκιμή στο πρόγραμμα περιήγησης

1. Ανοίξτε τη διαδικτυακή διεπαφή σας PhantomSDR
2. Μετακινηθείτε στην ενότητα **Additional Info**
3. Κάντε κλικ στο κουμπί **Open Additional Info**
4. Αναζητήστε το κουμπί **Stats** δίπλα στις πληροφορίες PC
5. Κάντε κλικ πάνω του για να ανοίξετε το παράθυρο διαλόγου στατιστικών
6. Επαληθεύστε ότι τα στατιστικά εμφανίζονται και ενημερώνονται

Διαχείριση υπηρεσίας

Αν το εγκαταστήσατε ως υπηρεσία systemd, χρησιμοποιήστε αυτές τις εντολές:

```
# Start the service
sudo systemctl start sdr-stats.service

# Stop the service
sudo systemctl stop sdr-stats.service

# Restart the service (after updating files)
sudo systemctl restart sdr-stats.service

# Check status
sudo systemctl status sdr-stats.service

# View real-time logs
sudo journalctl -u sdr-stats.service -f

# View last 50 log entries
sudo journalctl -u sdr-stats.service -n 50

# Enable auto-start on boot
sudo systemctl enable sdr-stats.service

# Disable auto-start
sudo systemctl disable sdr-stats.service

# Edit service
sudo nano /etc/systemd/system/sdr-stats.service

# Delete service
sudo rm /etc/systemd/system/sdr-stats.service
```

Χειροκίνητη εγκατάσταση (Επιλογή 2)

ΣΗΜΑΝΤΙΚΟ

Δεν χρειάζεστε αυτή την ενότητα για μια κανονική εγκατάσταση. Το `install-stats-server.sh` τα κάνει όλα αυτά για εσάς και το τρέχει το κύριο πρόγραμμα εγκατάστασης από προεπιλογή. Ακολουθήστε την μόνο για χειροκίνητη εγκατάσταση ή για να επισκευάσετε κάτι με το χέρι.

⚠ **Προειδοποίηση:** Η χειροκίνητη εγκατάσταση μπορεί να χρησιμοποιήσει παρωχημένα αρχεία. Το **αυτοματοποιημένο σενάριο συνιστάται ανεπιφύλακτα** καθώς:

- Εγκαθιστά την τελευταία έκδοση με βελτιωμένη ανίχνευση θερμοκρασίας
- Εγκαθιστά και διαμορφώνει αυτόματα το `lm-sensors`
- Διαχειρίζεται αυτόματα όλες τις εξαρτήσεις

Αν εξακολουθείτε να προτιμάτε τη χειροκίνητη εγκατάσταση:

1. Εγκαταστήστε το Node.js

```
# Any distribution - nvm needs no root and no apt source.
# NOT NodeSource: deb.nodesource.com now answers HTTP 403 on every path.
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.4/install.sh | bash
export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"
nvm install 22 && nvm use 22 && nvm alias default 22
```

2. Εγκαταστήστε το lm-sensors (Απαιτείται για τη θερμοκρασία Intel/AMD)

```
sudo apt-get update
sudo apt-get install -y lm-sensors
sudo sensors-detect --auto
```

3. Δημιουργήστε τον κατάλογο

```
mkdir ~/sdr-stats-server
cd ~/sdr-stats-server
```

4. Λήψη των τελευταίων αρχείων

Σημαντικό: τα αρχεία του διακομιστή δεν φυλάσσονται πλέον ως αντίγραφα στο αποθετήριο - τα δημιουργεί το σενάριο εγκατάστασης.

Αντ' αυτού, εκτελέστε το αυτοματοποιημένο σενάριο μία φορά για να δημιουργήσετε τα τελευταία αρχεία και, στη συνέχεια, αντιγράψτε τα:

```
# Run the automated script from the repository root
cd ~/PhantomSDR-Plus
./install-stats-server.sh
# When prompted, use a temporary port like 9999
# After installation completes, copy the generated files
cp ~/sdr-stats-server/system-stats-server.js ~/your-manual-install-dir/
cp ~/sdr-stats-server/package.json ~/your-manual-install-dir/
```

Ή δημιουργήστε χειροκίνητα τα αρχεία χρησιμοποιώντας τον πιο πρόσφατο κώδικα από το σενάριο εγκατάστασης.

5. Επεξεργαστείτε τη θύρα (αν χρειάζεται)

```
nano ~/sdr-stats-server/system-stats-server.js
```

Αλλάξτε τη γραμμή 6:

```
const PORT = process.env.PORT || 3001; // Change this number
```

6. Εγκαταστήστε τις εξαρτήσεις

```
npm install
```

7. Δοκιμαστική εκτέλεση

```
npm start
```

8. Ρύθμιση υπηρεσίας (Προαιρετικό)

Ακολουθήστε τη ρύθμιση υπηρεσίας systemd από την ενότητα αυτοματοποιημένης εγκατάστασης.

Αλλαγή της θύρας μετά την εγκατάσταση

Αν χρειαστεί να αλλάξετε τη θύρα μετά την εγκατάσταση:

Βήμα 1: Επεξεργαστείτε το αρχείο του διακομιστή

```
nano ~/sdr-stats-server/system-stats-server.js
```

Αλλάξτε τη γραμμή 6:

```
const PORT = process.env.PORT || 3001; // Change this number
```

Βήμα 2: Ενημερώστε το site_information.json

```
"siteStats": "http://Your_site_IP:NEW_PORT"
```

Βήμα 3: Επανεκκινήστε την υπηρεσία

```
sudo systemctl restart sdr-stats.service
```

Βήμα 4: Επαναδημιουργήστε την εφαρμογή σας Svelte

```
cd /path/to/phantomsdr-app
npm run build
```

Αντιμετώπιση προβλημάτων

Πρόβλημα: Η θύρα χρησιμοποιείται ήδη

Σφάλμα:

```
Error: listen EADDRINUSE: address already in use :::3001
```

Λύση:

1. Βρείτε τι χρησιμοποιεί τη θύρα:

```
sudo lsof -i :3001
```

1. Σταματήστε αυτήν τη διεργασία:

```
kill <PID>
# or
sudo systemctl stop sdr-stats.service
```

1. Ξεκινήστε ξανά την υπηρεσία:

```
sudo systemctl start sdr-stats.service
```

Εναλλακτική: Εκτελέστε ξανά το σενάριο εγκατάστασης και επιλέξτε διαφορετική θύρα όταν σας ζητηθεί.

Πρόβλημα: Η υπηρεσία δεν ξεκινά

Σφάλμα:

```
status=217/USER
```

Λύση:

1. Ελέγξτε το αρχείο υπηρεσίας:

```
cat /etc/systemd/system/sdr-stats.service
```

1. Επαληθεύστε ότι η γραμμή `User=` ταιριάζει με το όνομα χρήστη σας:

```
whoami
```

1. Επεξεργαστείτε αν χρειάζεται:

```
sudo nano /etc/systemd/system/sdr-stats.service
```

1. Επαναφορτώστε και επανεκκινήστε:

```
sudo systemctl daemon-reload
sudo systemctl restart sdr-stats.service
```

Πρόβλημα: Τα στατιστικά δεν εμφανίζονται στο πρόγραμμα περιήγησης

Συμπτώματα: Το παράθυρο διαλόγου ανοίγει αλλά χωρίς δεδομένα, ή εμφανίζει τιμές 0

Λύσεις:

1. Ελέγξτε την κονσόλα του προγράμματος περιήγησης (F12) για σφάλματα
2. Επαληθεύστε ότι το API είναι προσβάσιμο:

```
curl http://localhost:3001/api/system-stats
```

1. Ελέγξτε ότι η υπηρεσία εκτελείται:

```
sudo systemctl status sdr-stats.service
```

1. Επαληθεύστε ότι το `siteStats` στο `site_information.json` ταιριάζει με τη διεύθυνση και τη θύρα του διακομιστή σας
2. Ελέγξτε για σφάλματα CORS - βεβαιωθείτε ότι ο διακομιστής επιτρέπει τον τομέα σας

Πρόβλημα: Η θερμοκρασία εμφανίζει null

Αιτία: Η ανάγνωση θερμοκρασίας δεν είναι διαθέσιμη στο σύστημά σας

Λύσεις:

1. **Για τα περισσότερα συστήματα Linux:** Ελέγξτε αν υπάρχει η θερμική ζώνη:

```
cat /sys/class/thermal/thermal_zone0/temp
```

1. Εγκαταστήστε το lm-sensors:

```
sudo apt-get install lm-sensors
sudo sensors-detect
sensors
```

1. **Σημείωση:** Η θερμοκρασία μπορεί να μην είναι διαθέσιμη σε:
2. Εικονικές μηχανές
3. Ορισμένους παρόχους VPS
4. Windows Subsystem for Linux (WSL)
5. Μη Linux συστήματα

Αυτό είναι φυσιολογικό - τα υπόλοιπα στατιστικά θα εξακολουθούν να λειτουργούν!

Πρόβλημα: Οι τιμές των στατιστικών δεν ταιριάζουν με το top

Αιτία: Η περίοδος δειγματοληψίας CPU ήταν πολύ σύντομη σε παλαιότερες εκδόσεις

Λύση: Η τρέχουσα έκδοση χρησιμοποιεί δειγματοληψία 1 δευτερολέπτου για ακριβείς μετρήσεις. Αν έχετε μια παλιά εγκατάσταση:

1. Ενημερώστε το αρχείο του διακομιστή:

```
nano ~/sdr-stats-server/system-stats-server.js
```

1. Βρείτε τη γραμμή 18 και αλλάξτε:

```
}, 100); // Old value
```

σε:

```
}, 1000); // New value (1 second)
```

1. Επανεκκινήστε:

```
sudo systemctl restart sdr-stats.service
```

Ή απλώς εκτελέστε ξανά το σενάριο εγκατάστασης για να λάβετε την τελευταία έκδοση.

Πρόβλημα: Το Node.js δεν βρέθηκε

Σφάλμα:

```
node: command not found
```

Λύση:

1. Εγκαταστήστε το Node.js:

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.4/install.sh | bash
export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"
nvm install 22 && nvm use 22 && nvm alias default 22
```

1. Επαληθεύστε:

```
node --version
npm --version
```

Τι θα δείτε

Μόλις εγκατασταθεί και διαμορφωθεί, το παράθυρο διαλόγου στατιστικών εμφανίζει:

Ενότητα CPU

- **Χρήση:** Συνολικό ποσοστό αξιοποίησης CPU
- **Πυρήνες:** Πόση CPU χρησιμοποιείται πραγματικά, εκφρασμένη σε ολόκληρους πυρήνες — π.χ. 1.3 / 12 in use σημαίνει ότι το μηχάνημα κάνει δουλειά 1,3 πυρήνων. Είναι ακριβές μέγεθος και δεν χρειάζεται κατώφλι. Σκόπιμα δεν είναι καταμέτρηση απασχολημένων πυρήνων: ο χρονοπρογραμματιστής απλώνει τη δουλειά ενός πυρήνα σε πολλούς, οπότε μια τέτοια καταμέτρηση θα προσέγγιζε απλώς αυτόν τον αριθμό.
- **Συχνότητα:** Μέση τρέχουσα συχνότητα CPU σε όλους τους πυρήνες σε GHz, με τον ταχύτερο πυρήνα σε παρένθεση (αν είναι διαθέσιμη)
- **Θερμοκρασία:** Θερμοκρασία CPU σε Κελσίου (αν είναι διαθέσιμη), με χρωματική κωδικοποίηση: πράσινο έως 69 °C, πορτοκαλί 70-79 °C, κόκκινο από 80 °C και πάνω
- **Κορυφαίες διεργασίες:** Οι 5 διεργασίες με τη μεγαλύτερη κατανάλωση CPU με ποσοστά κωδικοποιημένα με χρώματα:
 - Πράσινο: έως 50 % CPU
 - Πορτοκαλί: 51-80 % CPU
 - Κόκκινο: πάνω από 80 % CPU

Ενότητα μνήμης

- **Χρησιμοποιούμενη:** Μνήμη σε χρήση (GB)
- **Συνολική:** Συνολική μνήμη συστήματος (GB)
- **Χρήση:** Ποσοστό χρήσης μνήμης (κωδικοποιημένο με χρώματα)

🔄Αυτόματη ενημέρωση

- Τα στατιστικά ανανεώνονται κάθε **5 δευτερόλεπτα** όσο το παράθυρο διαλόγου είναι ανοιχτό
- Σταματούν να ενημερώνονται όταν κλείνετε το παράθυρο διαλόγου (εξοικονομεί πόρους)

Ανίχνευση θερμοκρασίας

Ο διακομιστής στατιστικών χρησιμοποιεί προηγμένη ανίχνευση θερμοκρασίας πολλαπλών μεθόδων:

Υποστηριζόμενα συστήματα:

- **Επεξεργαστές Intel:** Διαβάζει από αισθητήρες coretemp μέσω lm-sensors

- **Επεξεργαστές AMD:** Διαβάζει από αισθητήρες k10temp (Tdie/Tctl)
- **Επεξεργαστές ARM:** Διαβάζει από θερμικές ζώνες (Raspberry Pi, κ.λπ.)

Απαιτήσεις:

- **Intel/AMD:** Απαιτεί το πακέτο `lm-sensors` (εγκαθίσταται αυτόματα από το σενάριο)
- **ARM/Raspberry Pi:** Λειτουργεί άμεσα, δεν χρειάζονται επιπλέον πακέτα

Προτεραιότητα εγκατάστασης:

1. Δοκιμάζει την εντολή `sensors` για τη θερμοκρασία Package/Core (Intel/AMD)
2. Επιστρέφει στην άμεση ανάγνωση `coretemp` (Intel)
3. Επιστρέφει στο `thermal_zone0` (ARM/Raspberry Pi)

Αν η θερμοκρασία εμφανίζει `null`, εγκαταστήστε το `lm-sensors`:

```
sudo apt-get install lm-sensors
sudo sensors-detect --auto
sudo systemctl restart sdr-stats.service
```

Σύνοψη χαρακτηριστικών

Παρακολούθηση σε πραγματικό χρόνο - Ζωντανά στατιστικά συστήματος από τον διακομιστή σας **Μετρήσεις CPU** - Χρήση, πυρήνες, θερμοκρασία, κορυφαίες διεργασίες **Παρακολούθηση μνήμης** - Χρησιμοποιούμενη, συνολική και ποσοστό **Αυτόματες ενημερώσεις** - Ανανεώνεται κάθε 5 δευτερόλεπτα **Κωδικοποιημένο με χρώματα** - Οπτικές ενδείξεις για προειδοποιήσεις **Ελαφρύ** - Ελάχιστη χρήση πόρων **Εύκολη εγκατάσταση** - Το αυτοματοποιημένο σενάριο τα φροντίζει όλα **Επικύρωση θύρας** - Έξυπνη επιλογή θύρας με ανίχνευση διενέξεων **Διαχείριση υπηρεσίας** - Αυτόματη εκκίνηση κατά την εκκίνηση

Σημειώσεις ασφαλείας

- Ο διακομιστής στατιστικών δέχεται συνδέσεις από οποιαδήποτε προέλευση (CORS: *)
- Για παραγωγικό περιβάλλον, εξετάστε τον περιορισμό του CORS μόνο στον τομέα σας
- Η επιλογή θύρας επικυρώνει την είσοδο και ελέγχει για διενέξεις
- Η ανάγνωση της θερμοκρασίας CPU είναι μόνο για ανάγνωση και ασφαλής
- Δεν εκτίθενται ευαίσθητες πληροφορίες συστήματος

Υποστήριξη και πόροι

Περιλαμβανόμενα αρχεία:

Σε αυτόν τον φάκελο (το ίδιο το σενάριο εγκατάστασης βρίσκεται στη ρίζα, ως `PhantomSDR-Plus/install-stats-server.sh` - υπάρχει πάντα ένα μόνο αντίγραφο):

- `package.json` - Εξαρτήσεις, ίδιες με αυτές που γράφει το σενάριο
- Αυτό το έγγραφο, διαθέσιμο επίσης σε `de`, `el`, `es`, `fr`, `hr` και `ru`

Δημιουργούνται από το σενάριο στο `~/sdr-stats-server/`:

- `system-stats-server.js` - Διακομιστής backend, με τη θύρα που επιλέξατε
- `package.json` - Εξαρτήσεις
- `node_modules/` - εγκαθίστανται από το `npm install`

Λήψη βοήθειας:

- Ελέγξτε την ενότητα αντιμετώπισης προβλημάτων παραπάνω
- Εξετάστε τα αρχεία καταγραφής της υπηρεσίας: `sudo journalctl -u sdr-stats.service -n 50`
- Δοκιμάστε το API χειροκίνητα: `curl http://localhost:3001/api/system-stats`
- Επαληθεύστε τα αρχεία διαμόρφωσης

Απεγκατάσταση του διακομιστή στατιστικών:

- Διαγράψτε τον φάκελο `/sdr-stats-server`
- Αν έχετε δημιουργήσει μια υπηρεσία, απλώς διαγράψτε την χρησιμοποιώντας:

```
sudo rm /etc/systemd/system/sdr-stats.service`
```

- Τροποποιήστε το `App.svelte`: Βρείτε το τμήμα του κώδικα:

```
<!-- In case you don't want the Stats Button to appear, please comment this
button section (12 lines)-->
<!-- System Stats Button -->
<button
  type="button"
  class="glass-button text-white py-1 px-2 ml-2 rounded text-xs"
  on:click={openSystemStats}
  title="System Resources"
  aria-haspopup="dialog"
  aria-expanded={showSystemStats}
  aria-controls="system-stats-dialog"
  style="color:rgba(0, 225, 255, 0.993); font-size: 0.75rem;"
>
  Stats
</button>
...
και αντικαταστήστε με:
```

```
<!-- In case you don't want the Stats Button to appear, please comment this button
section (12 lines)-->
<!-- System Stats Button -->
<!--
<button
  type="button"
  class="glass-button text-white py-1 px-2 ml-2 rounded text-xs"
  on:click={openSystemStats}
  title="System Resources"
  aria-haspopup="dialog"
  aria-expanded={showSystemStats}
  aria-controls="system-stats-dialog"
  style="color:rgba(0, 225, 255, 0.993); font-size: 0.75rem;"
>
  Stats
</button>
-->
```

...

- Επαναμεταγλωττίστε χρησιμοποιώντας την εντολή:

```
cd PhantomSDR-Plus
./recompile.sh
...

- Επανεκκινήστε τον υπολογιστή και ξεκινήστε τον διακομιστή

---

**Συγγραφέας:** Δημιουργήθηκε για το PhantomSDR-Plus
**Έκδοση:** 1.1 (Βελτιωμένη ανίχνευση θερμοκρασίας CPU για Intel/AMD) **Ενημερώθηκε:** Ιανουάριος
2026
**Άδεια:** MIT

---

## Γρήγορη αναφορά εντολών

```bash
Installation
cd ~/PhantomSDR-Plus
chmod +x install-stats-server.sh
./install-stats-server.sh

Service Management
sudo systemctl start sdr-stats.service # Start
sudo systemctl stop sdr-stats.service # Stop
sudo systemctl restart sdr-stats.service # Restart
sudo systemctl status sdr-stats.service # Status
sudo journalctl -u sdr-stats.service -f # Logs

Testing
curl http://localhost:3001/api/system-stats # Test API
curl http://localhost:3001/api/health # Health check

Troubleshooting
sudo lsof -i :3001 # Check port
whoami # Check username
node --version # Check Node.js
```

Καλή παρακολούθηση!

# Αποκωδικοποιητές

DECODERS.md

## PhantomSDR-Plus — Εγχειρίδιο χρήσης αποκωδικοποιητών

Ο οδηγός αυτός καλύπτει κάθε ενσωματωμένο αποκωδικοποιητή του PhantomSDR-Plus. Όλοι οι αποκωδικοποιητές ενεργοποιούνται με την ίδια διαδικασία, που περιγράφεται παρακάτω, και ακολουθούν οι οδηγίες ρύθμισης για τον καθένα.

### Πίνακας περιεχομένων

1. Πώς ξεκινά ένας αποκωδικοποιητής
2. FT8
3. FT4-FT2
4. JS8
5. CW — κώδικας Μορς
6. QRSS Grabber
7. WSPR
8. HF FAX / WEFAX
9. NAVTEX
10. FSK / RTTY — μαζί με PSK31 και Olivia
11. SSTV
12. Γενικές συμβουλές

### 1. Πώς ξεκινά ένας αποκωδικοποιητής

Όλοι οι αποκωδικοποιητές βρίσκονται στην ενότητα **Decoder Options**, κάτω από τα χειριστήρια του φασματογραφήματος ήχου στον κύριο πίνακα.

#### Βήματα:

1. Κάντε κλικ στο κουμπί **Decoder: OFF** για να γίνει **Decoder: ON** (γίνεται μπλε όταν είναι ενεργό).
2. Ανοίξτε το αναπτυσσόμενο μενού που εμφανίζεται δεξιά από το κουμπί και επιλέξτε τον αποκωδικοποιητή που θέλετε.
3. Ο πίνακας του επιλεγμένου αποκωδικοποιητή θα εμφανιστεί κάτω από τα χειριστήρια — ακολουθήστε τις οδηγίες του συγκεκριμένου αποκωδικοποιητή στην αντίστοιχη ενότητα αυτού του οδηγού.
4. Για να σταματήσετε την αποκωδικοποίηση, επιλέξτε — **Select decoder** — από το μενού ή κάντε κλικ στο κουμπί **Decoder: ON** για να το επαναφέρετε σε OFF.

#### Κουμπιά αποκωδικοποιητών με ένα πάτημα

Το αναπτυσσόμενο μενού δεν είναι ο μόνος τρόπος. Ο κύριος πίνακας διαθέτει μια σειρά κουμπιών **Decoders** — ακριβώς κάτω από τα **Wheel Tuning Steps** — με ένα κουμπί για κάθε αποκωδικοποιητή:

Κουμπί	Αποκωδικοποιητής	Κουμπί	Αποκωδικοποιητής
<b>FT8</b>	FT8	<b>SSTV</b>	SSTV
<b>FT4</b>	FT4	<b>NAVTEX</b>	NAVTEX
<b>FT2</b>	FT2	<b>RTTY</b>	FSK / RTTY
<b>CW</b>	CW		
<b>WSPR</b>	WSPR	<b>FAX</b>	HF FAX / WEFAX

Το πάτημα ενός κουμπιού κάνει όλη τη διαδικασία με τη μία: επιλέγει τον αποκωδικοποιητή, ανάβει τον διακόπτη Decoder (ON) και φέρνει το παράθυρό του στην οθόνη. Το κουμπί γίνεται μπλε όσο τρέχει ο αποκωδικοποιητής του. **Πατήστε το ίδιο κουμπί ξανά για να σταματήσει ο αποκωδικοποιητής** — το παράθυρό του κλείνει μαζί.

Τα **RADEL** και **RADEU** δεν περιλαμβάνονται σκόπιμα σε αυτή τη σειρά. Είναι τρόποι ψηφιακής φωνής και όχι αποκωδικοποιητές κειμένου, γι' αυτό έχουν το δικό τους ζευγάρι κουμπιών δίπλα στην επικεφαλίδα **Modes selector**, καθώς και μέσα στα αναδυόμενα παράθυρα **Modes** και **Bands**. Λειτουργούν ακριβώς με τον ίδιο τρόπο — πάτημα για εκκίνηση, δεύτερο πάτημα για διακοπή. Δείτε το [εγχειρίδιο RADE](#).

Τα κουμπιά, το αναπτυσσόμενο μενού και το κουμπί ON/OFF ελέγχουν την ίδια κατάσταση, οπότε ό,τι κι αν χρησιμοποιήσετε, τα υπόλοιπα ακολουθούν.

Μόνο ένας αποκωδικοποιητής μπορεί να είναι ενεργός κάθε φορά. Η μετάβαση σε άλλον αποκωδικοποιητή σταματά αυτόματα τον προηγούμενο.

Ο **QRSS grabber** δεν ανήκει σε αυτό το μενού — έχει δικό του τμήμα **QRSS** και μπορεί να τρέχει ταυτόχρονα με έναν αποκωδικοποιητή.

**Η αποκωδικοποίηση εκτελείται στο παρασκήνιο.** Τα SSTV, HF FAX, NAVTEX, FSK/RTTY και CW εκτελούνται το καθένα σε ξεχωριστό νήμα Web Worker, ώστε η αποκωδικοποίηση να μην ανταγωνίζεται ποτέ την αναπαραγωγή ήχου ή τον καταρράκτη. Η εκκίνηση ή διακοπή ενός αποκωδικοποιητή δεν διακόπτει τον ήχο και η διεπαφή παραμένει αποκρίσιμη ενώ λαμβάνεται μια εικόνα ή μια σελίδα. Όλοι οι αποκωδικοποιητές τροφοδοτούνται με ακατέργαστο ήχο που λαμβάνεται πριν από το AGC, τη μείωση θορύβου και τη σίγαση — έτσι, αν σιγάσετε τον δέκτη ή αλλάξετε αυτές τις ρυθμίσεις κατά το γούστο σας, η αποκωδικοποίηση δεν επηρεάζεται.

**Ένας αποκωδικοποιητής που τρέχει κατέχει τον τρόπο λήψης και το εύρος διέλευσης.**

Κανονικά ο τρόπος λήψης ακολουθεί τον χάρτη μπαντών στο `bands-config.js`: μετακινήστε τον συντονισμό σε τμήμα σημειωμένο ως LSB ή AM και ο δέκτης αλλάζει σε αυτό. Όσο τρέχει ένας αποκωδικοποιητής, αυτό δεν συμβαίνει πλέον. Ο αποκωδικοποιητής κρατά τον τρόπο λήψης που χρειάζεται (USB για τους περισσότερους, δικό του για το RADE) και το εύρος διέλευσης που χρειάζεται — PSK31 περίπου  $\pm 100$  Hz, Olivia όλο του το εύρος, RTTY το shift του — και τα δύο επιβιώνουν του επανασυντονισμού, ακόμη και ενός άλματος σε άλλη μπάντα. Χωρίς αυτό, το FT8 στα 40 m θα γύριζε τον δέκτη σε LSB μόλις κινούνταν ο συντονισμός, και τα στενά εύρη των αποκωδικοποιητών θα άνοιγαν πάλι στο πλήρες φίλτρο SSB.

Ο τρόπος λήψης της ίδιας της μπάντας επιστρέφει τη στιγμή που σβήνετε τον αποκωδικοποιητή. Μπορείτε πάντα να επιβάλετε τρόπο λήψης με το χέρι: τα κουμπιά των modes είναι συνειδητή επιλογή και υπερισχύουν πάντα. Ο αποκωδικοποιητής CW είναι η εξαίρεση σε όλα αυτά — αποκωδικοποιεί στον τρόπο λήψης που ακούτε ήδη και ποτέ δεν παίρνει τον έλεγχο του δέκτη.

## Decoder ID — αυτόματη αναγνώριση του τρόπου εκπομπής

Πάνω από τα κουμπιά των αποκωδικοποιητών βρίσκεται το πλαίσιο **Decoder ID**. Απαντά στην ερώτηση «τι ακούω;» όταν πέσετε πάνω σε ένα ψηφιακό σήμα που δεν αναγνωρίζετε, και στη συνέχεια σας προσφέρει τον κατάλληλο αποκωδικοποιητή με ένα κλικ.

Είναι **απενεργοποιημένο εξ ορισμού** και δεν κοστίζει τίποτα όσο είναι κλειστό. Πατήστε **On** και αφήστε το σήμα συντονισμένο. Για περίπου μισό λεπτό δείχνει *Listening...* με μπάρα προόδου — οι μετρήσεις χρονισμού πρέπει να δουν δύο πλήρεις κύκλους FT8 για να έχουν νόημα — και μετά ονομάζει τον τρόπο εκπομπής:

Decoder ID [On] NAVTEX / SIT0R-B 99% 170 Hz shift · 100.00 Bd · 43 dB S/N [ Use NAVTEX ]

- Ο **τρόπος εκπομπής** και ένα ποσοστό **βεβαιότητας**. Πράσινο πάνω από 75%, κίτρινο πάνω από 50%, πορτοκαλί χαμηλότερα — μια πορτοκαλί απάντηση είναι ένδειξη, όχι ετυμηγορία.
- **Τι μετρήθηκε πραγματικά:** καταλαμβανόμενο εύρος ζώνης, απόσταση τόνων, ρυθμός συμβόλων, διάρκεια ριπής, το πλέγμα UTC στο οποίο ταιριάζει, λόγος σήματος προς θόρυβο. Αυτά είναι τα τεκμήρια πίσω από την ετυμηγορία, ώστε να κρίνετε και μόνοι σας.
- Το **Use <τρόπος>** ξεκινά τον σωστό αποκωδικοποιητή, ακριβώς σαν να είχατε πατήσει το κουμπί του — και ρυθμίζει τον δέκτη για να δουλέψετε αυτόν τον τρόπο εκπομπής. Για τους τρόπους που βρίσκονται στο πλαίσιο FSK επιλέγει επιπλέον τη σωστή παραλλαγή — RTTY, Weather RTTY, PSK31 ή Olivia.
- Οι επόμενοι υποψήφιοι εμφανίζονται μετά το **or**. Ένα κλικ δοκιμάζει εκείνον, οπότε μια λανθασμένη πρώτη απάντηση δεν σας κοστίζει τίποτα.
- Το **\*\*\*\*** σημαίνει ότι η συχνότητα στην οποία είστε συντονισμένοι είναι γνωστή συχνότητα κλήσης για αυτόν τον τρόπο, και ότι αυτό μέτρησε στην κατάταξη.

Δεν αλλάζει ποτέ μόνο του τον αποκωδικοποιητή σας. Πάντα απλώς προτείνει. Και μόλις ξεκινήσετε έναν αποκωδικοποιητή — από τη σειρά κουμπιών, από την αναπτυσσόμενη λίστα ή από το ίδιο το κουμπί **Use** του Decoder ID — το Decoder ID απενεργοποιείται μόνο του: η δουλειά του ήταν να επιλέξει έναν, και τώρα ο αποκωδικοποιητής χρειάζεται τον ίδιο ήχο.

**Ρυθμίζει τον δέκτη για τον τρόπο εκπομπής.** Η επιλογή ενός τρόπου εκπομπής είναι απόφαση να τον δουλέψετε, οπότε ο δέκτης ακολουθεί: η συντονισμένη συχνότητα πηγαίνει στο κέντρο του καταρράκτη και η εικόνα γίνεται περίπου 100 kHz γύρω της, η πλευρική ζώνη γίνεται αυτή στην οποία δουλεύεται ο τρόπος αυτός και το εύρος διέλευσης στενεύει στο παράθυρο όπου ζει ο τρόπος αυτός — ολόκληρη η υποζώνη των 3 kHz για FT8, FT4, FT2 και JS8, 1350–1650 Hz για WSPR, 250–750 Hz για NAVTEX, 800–2700 Hz για HF FAX, 900–2600 Hz για SSTV και  $\pm 250$  Hz εκατέρωθεν της συχνότητας για CW, όπου ο δέκτης περνά επιπλέον σε CW. Ένα στενό εύρος διέλευσης κρατά τα γειτονικά σήματα έξω από τον αποκωδικοποιητή — ακριβώς ό,τι θα κάνατε με το χέρι. Δύο εξαιρέσεις: το SSTV κρατά την πλευρική ζώνη που επέλεξε μόνο του — LSB στα 80 και 40 m κατά σύμβαση, όπου η επιβολή USB θα αντέστρεφε την αντιστοιχία των τόνων — και οι τρόποι FSK κρατούν το παράθυρο που παράγει το ίδιο τους το πλαίσιο από το shift ή το εύρος της παραλλαγής, το οποίο είναι στενότερο από κάθε σταθερή τιμή. Το πάτημα του **Use** για τον αποκωδικοποιητή που ήδη τρέχει εξακολουθεί να τον σβήνει, και τότε ο δέκτης μένει ακριβώς όπως ήταν.

**Οι αργοί τρόποι θέλουν περισσότερο χρόνο.** Το WSPR βρίσκεται σε πλέγμα δύο λεπτών και οι εικόνες SSTV διαρκούν ένα με δύο λεπτά, οπότε αυτά τα δύο αναγνωρίζονται αργότερα από τα υπόλοιπα — δώστε τους τρία ή τέσσερα λεπτά στη συχνότητα. Το HF FAX και το SSTV αναγνωρίζονται επίσης μόνο όταν είναι συντονισμένα με τον συνηθισμένο τρόπο — η ζώνη της εικόνας να βρίσκεται περίπου στα 1500 με 2300 Hz στο ακουστικό φάσμα, εκεί δηλαδή που την περιμένουν και οι αποκωδικοποιητές τους. Κάθε αλλαγή συχνότητας ξεκινά τη μέτρηση από την αρχή, γιατί ό,τι έχει συλλεχθεί ως τότε περιγράφει τη συχνότητα που μόλις αφήσατε.

**Τι αναγνωρίζει:** FT8, FT4, JS8, WSPR, CW, NAVTEX/SITOR-B, RTTY στα 45,45 Bd, Weather RTTY (DWD), PSK31, Olivia, HF FAX, SSTV και DSC. Σε αυτή την έκδοση δεν υπάρχει αποκωδικοποιητής DSC, οπότε το DSC ονομάζεται και σημειώνεται ως *no decoder*.

**Recentre & retry.** Το πού βρίσκεται το σήμα μέσα στη ζώνη διέλευσης ήχου δεν έχει σημασία μεταξύ περίπου 400 Hz και 2700 Hz. Εκτός αυτού του εύρους η μέτρηση χειροτερεύει και εμφανίζεται ένα κίτρινο κουμπί **Recentre & retry**. Πατώντας το, ο δέκτης μετακινείται ώστε το σήμα να βρεθεί στο μέσο της ζώνης διέλευσης και ξεκινά νέα μέτρηση. Είναι μία υπολογισμένη κίνηση, όχι σάρωση.

### Τρία ειλικρινή όρια:

- **Τα FT8, JS8 και FT2 είναι το ίδιο σήμα.** Ίδια διαμόρφωση 8-FSK, ίδια απόσταση τόνων, και το JS8 σε ταχύτητα Normal χρησιμοποιεί τον ίδιο κύκλο 15 δευτερολέπτων με το FT8. Τίποτα μέσα στο ίδιο το σήμα δεν τα ξεχωρίζει. Η συχνότητα όμως τα ξεχωρίζει: σε συχνότητα κλήσης FT8 αναφέρει FT8, σε συχνότητα JS8 αναφέρει JS8, και οπουδήποτε αλλού αναφέρει ειλικρινά την οικογένεια — *FT8 / JS8 / FT2*. Ξεκινήστε όποιον από τους δύο και δείτε ποιος βγάζει κείμενο.
- **Σιωπά αντί να μαντεύει.** Κάτω από περίπου 10 dB λόγο σήματος προς θόρυβο αναφέρει *Nothing above the noise* ή *Not recognised*, αντί να ονομάσει έναν τρόπο εκπομπής που δεν βλέπει πραγματικά. Αυτό είναι σκόπιμο: μια σίγουρη λανθασμένη απάντηση είναι χειρότερη από καμία απάντηση.
- **Ένα σήμα με διαλείψεις διαβάζεται αλλιώς.** Το WSPR εκπέμπει 110 δευτερόλεπτα χωρίς διακοπή, δηλαδή περισσότερο από όσο κρατά μια συνηθισμένη διάλειψη QSB. Έτσι μια βαθιά διάλειψη κόβει την εκπομπή σε κομμάτια και ο δίλεπτος ρυθμός της δεν διαβάζεται πλέον από αυτά. Τότε ανακτάται με άλλο τρόπο: ολόκληρο το ιστορικό ακρόασης συγκρίνεται με το δίλεπτο πλέγμα αντί για τις μεμονωμένες εκπομπές, και η γραμμή των μετρήσεων γράφει *120 s grid (through fading)*. Αυτό είναι ασθενέστερη ένδειξη από μια εκπομπή που ακούστηκε ολόκληρη, γι' αυτό η βεβαιότητα που εμφανίζεται είναι σκόπιμα χαμηλότερη. Μια διάλειψη αρκετά βαθιά ώστε να θάψει το σήμα στον θόρυβο δεν ανακτάται καθόλου. Το Olivia είναι αυστηρότερο από τα υπόλοιπα: δεν ονομάζεται πριν μετρηθεί ο ρυθμός συμβόλων του, και έτσι σιωπά νωρίτερα.

## 2. FT8

**Τι είναι:** το FT8 είναι ένας δημοφιλής ψηφιακός τρόπος ασθενούς σήματος που χρησιμοποιούν ραδιοερασιτέχνες παγκοσμίως. Οι εκπομπές διαρκούν ακριβώς 15 δευτερόλεπτα και το πρωτόκολλο μπορεί να λάβει σήματα έως 20-25 dB κάτω από τη στάθμη θορύβου. Είναι ο πιο διαδεδομένος τρόπος για επαφές μεγάλων αποστάσεων (DX).

## Συνιστώμενες συχνότητες (USB)

Μπάντα	Συχνότητα
160 m	1.840 MHz
80 m	3.573 MHz
40 m	7.074 MHz
30 m	10.136 MHz
20 m	14.074 MHz
17 m	18.100 MHz
15 m	21.074 MHz
12 m	24.915 MHz
10 m	28.074 MHz

## Ρύθμιση

1. Συντονιστείτε σε μια από τις παραπάνω συχνότητες FT8 και θέστε τον τρόπο σε **USB**.
2. Ενεργοποιήστε τον αποκωδικοποιητή και επιλέξτε **FT8** από το μενού. Ή απλώς πατήστε το κουμπί **FT8**.
3. Ο πίνακας **FT8 Messages** εμφανίζεται αυτόματα από κάτω.

## Ανάγνωση των αποτελεσμάτων

Η λίστα μηνυμάτων δείχνει τις αποκωδικοποιημένες εκπομπές καθώς φτάνουν. Κάθε κύκλος 15 δευτερολέπτων παράγει μια νέα παρτίδα μηνυμάτων. Το πεδίο **Farthest** (πάνω δεξιά στον πίνακα) δείχνει τη μεγαλύτερη απόσταση που αποκωδικοποιήθηκε στην τρέχουσα συνεδρία, σε χιλιόμετρα.

Τυπική μορφή μηνύματος: **CQ DX AA1BB FN31** — κλήση CQ από το διακριτικό AA1BB που βρίσκεται στο τετράγωνο FN31.

Το **SNR** που φαίνεται σε κάθε αποκωδικοποίηση είναι πραγματικός λόγος σήματος προς θόρυβο, μετρημένος πάνω στο αποκωδικοποιημένο σήμα σε σχέση με τον θόρυβο γύρω του και αναγόμενος σε εύρος 2500 Hz, όπως το αναφέρει το WSJT-X. Οι ασθενείς σταθμοί δείχνουν αρνητικές τιμές, ως περίπου -20 dB, και οι ισχυροί θετικές· στις ίδιες ηχογραφήσεις τα δύο προγράμματα συμφωνούν κατά μέσο όρο, συνήθως μέσα σε περίπου 3 dB για μια μεμονωμένη αποκωδικοποίηση. Τα FT4 και FT2 χρησιμοποιούν την ίδια μέθοδο.

**Μετά από ενημέρωση:** ο αποκωδικοποιητής FT8/FT4/FT2 είναι ξεχωριστό αρχείο που οι φυλλομετρητές κρατούν για πολύ καιρό. Αν όλα τα SNR μένουν ανάμεσα σε περίπου 0 και +5 dB, φορτώστε ξανά τη σελίδα με πλήρη ανανέωση (**Ctrl+Shift+R**, ή **Cmd+Shift+R** σε Mac).

**Συμβουλή:** το FT8 είναι αυστηρά συγχρονισμένο χρονικά. Ο περιηγητής σας χρησιμοποιεί το ρολόι του υπολογιστή σας· αν αυτό αποκλίνει περισσότερο από λίγα δευτερόλεπτα, η αποκωδικοποίηση θα αποτύχει. Κρατήστε την ώρα του συστήματος συγχρονισμένη με NTP.

## 3. FT4-FT2

**Τι είναι:** το FT4 είναι μια ταχύτερη παραλλαγή του FT8, σχεδιασμένη για λειτουργία τύπου διαγωνισμού. Κάθε κύκλος εκπομπής διαρκεί 7,5 δευτερόλεπτα (το μισό του FT8), γεγονός που το κάνει διπλάσια γρήγορο αλλά απαιτεί ελαφρώς ισχυρότερο σήμα. Το FT2 είναι ακόμη ταχύτερη παραλλαγή

του FT8. Είναι ένας εξαιρετικά γρήγορος τρόπος 77 bit με περιόδους TR 3,75 δευτερολέπτων (= το μισό του FT4) — ακόμη πειραματικός.

### Συνιστώμενες συχνότητες (USB)

Μπάντα	Συχνότητα FT4
80 m	3.575 MHz
40 m	7.047 MHz
30 m	10.140 MHz
20 m	14.080 MHz
15 m	21.140 MHz
10 m	28.180 MHz

Μπάντα	Συχνότητα FT2
160 m	1.843 έως 1.846
80 m	3.578 έως 3.581
60 m	5.360 (ελέγξτε τους τοπικούς κανονισμούς)
40 m	7.052 έως 7.062
30 m	10.144
20 m	14.084
17 m	18.108
15 m	21.144
12 m	24.923
10 m	28.184

### Ρύθμιση

1. Συντονιστείτε σε μια συχνότητα FT4 και θέστε τον τρόπο σε **USB**.
2. Ενεργοποιήστε τον αποκωδικοποιητή και επιλέξτε **FT4** ή **FT2** από το μενού. Ή απλώς πατήστε το κουμπί **FT4 / FT2**.
3. Ο πίνακας **FT4 Messages** ή **FT2 Messages** εμφανίζεται από κάτω, με πανομοιότυπη διάταξη με τον πίνακα FT8.

**Σημείωση:** τα FT4 και FT8 χρησιμοποιούν διαφορετικές φασματικές μορφές και δεν είναι εναλλάξιμα. Βεβαιωθείτε ότι βρίσκεστε σε συχνότητα FT4 όταν χρησιμοποιείτε αυτόν τον αποκωδικοποιητή.

## 4. JS8

**Τι είναι:** Το JS8 (ο τρόπος λειτουργίας του JS8Call) παίρνει τη μηχανή ασθενούς σήματος του FT8 και τη μετατρέπει σε τρόπο συνομιλίας πληκτρολογίου. Ενώ το FT8 στέλνει σταθερά μηνύματα 13 χαρακτήρων, το JS8 στέλνει ελεύθερο κείμενο λίγους χαρακτήρες κάθε φορά και ενώνει τις διαδοχικές εκπομπές σε ολόκληρες προτάσεις — έτσι οτιδήποτε ξεπερνά τους δώδεκα χαρακτήρες φτάνει σε

πολλούς κύκλους των 15 δευτερολέπτων. Αποκωδικοποιεί περίπου τόσο βαθιά όσο το FT8, οπότε δουλεύει εκεί που φωνή και CW δεν περνούν.

## Προτεινόμενες συχνότητες (USB)

Μπάντα	Συχνότητα
160m	1.842 MHz
80m	3.578 MHz
40m	7.078 MHz
30m	10.130 MHz
20m	14.078 MHz
17m	18.104 MHz
15m	21.078 MHz
12m	24.922 MHz
10m	28.078 MHz

## Ταχύτητες

Το JS8 έχει πέντε ταχύτητες. Όλοι οι σταθμοί σε μια συνομιλία πρέπει να χρησιμοποιούν την ίδια. Η **Normal** είναι η ταχύτητα κλήσης και εκεί γίνεται σχεδόν όλη η κίνηση — ξεκινήστε από εκεί.

Ταχύτητα	Κύκλος	Εύρος	Πότε χρησιμοποιείται
Slow	30 δευτ.	25 Hz	Πολύ ασθενείς διαδρομές· η πιο ευαίσθητη
Normal	15 δευτ.	50 Hz	Η κανονική ταχύτητα κλήσης
Fast	10 δευτ.	80 Hz	Ταχύτερες ανταλλαγές, θέλει ισχυρότερο σήμα
Turbo	6 δευτ.	160 Hz	Ισχυρά τοπικά σήματα
Ultra	4 δευτ.	250 Hz	Πειραματική, σπάνια στον αέρα

## Ρύθμιση

1. Συντονιστείτε σε μια συχνότητα JS8 και βάλτε **USB**.
2. Ενεργοποιήστε τον αποκωδικοποιητή και επιλέξτε **JS8** από τη λίστα, ή απλώς πατήστε το κουμπί **JS8**.
3. Εμφανίζεται το παράθυρο **JS8 Decoder**. Αφήστε την **Speed** στο *Normal*, εκτός αν ξέρετε ότι ο σταθμός που θέλετε χρησιμοποιεί άλλη.

## Πώς διαβάζεται η έξοδος

Το παράθυρο έχει δύο λίστες.

**Μηνύματα που ακόμη έρχονται** εμφανίζονται επάνω με πράσινο χρώμα και δρομέα που αναβοσβήνει. Ένα μήνυμα JS8 μπορεί να χρειαστεί τέσσερις κύκλους — ένα ολόκληρο λεπτό στο Normal — οπότε εκεί βλέπετε την πρόταση να χτίζεται. Δεν έχει πάει κάτι στραβά αν μείνει εκεί για λίγο.

**Ολοκληρωμένα μηνύματα** γεμίζουν την κύρια λίστα, μία γραμμή το καθένα:

Στήλη	Σημασία
Mode	Πάντα JS8
Hz	Συχνότητα ακρόασης του σήματος μέσα στο passband
dB	Λόγος σήματος προς θόρυβο σε εύρος αναφοράς 2500 Hz
Message	Το αποκωδικοποιημένο κείμενο· <b>τα διακριτικά εμφανίζονται πράσινα</b>

Τυπικά μηνύματα:

- SV1BTL KM17: HB — heartbeat: ο σταθμός ανακοινώνει ότι είναι στον αέρα, με το τετράγωνό του.
- KN4CRD: K00G SNR -05 — κατευθυνόμενο μήνυμα: ο KN4CRD λέει στον K00G ότι τον λαμβάνει στα -5 dB.
- MP 100W 8M/BALUN JN58KH AUGSBURG, MARTIN — ελεύθερο κείμενο, εδώ μια περιγραφή σταθμού που ήρθε σε τέσσερις κύκλους.

Μήνυμα σε **σβησμένα πλάγια** σημαίνει ότι έληξε ο χρόνος πριν φτάσει το τελευταίο του πλαίσιο. Το κείμενο είναι αληθινό, αλλά μπορεί να είναι κομμένο.

## Sync offset

Το ρυθμιστικό **Sync offset** ορίζει πόσο μετά το όριο του κύκλου UTC αρχίζει η καταγραφή, αντισταθμίζοντας την καθυστέρηση της αλυσίδας ήχου. Αφήστε το **Auto** επιλεγμένο: ο αποκωδικοποιητής μετρά τον χρονισμό των σημάτων που ακούει και προσαρμόζεται μόνος του.

## Σημάνσεις φάσματος

Η μικρή λωρίδα φάσματος στο παράθυρο σχεδιάζει μια κίτρινη κατακόρυφη γραμμή στη συχνότητα κάθε σήματος που αποκωδικοποιήθηκε στον προηγούμενο κύκλο, ώστε η στήλη Hz να διαβάζεται κατευθείαν από την οθόνη.

## Δείτε τα δικά σας spot

Αν ο δέκτης ανεβάξει spot JS8 στο PSK Reporter (δείτε την καρτέλα Spot Reporting στο admin panel), το κουμπί **JS8 map** στο πλαίσιο **Note**: ανοίγει τον χάρτη του PSK Reporter φιλτραρισμένο στο διακριτικό αυτού του δέκτη και στον τρόπο JS8 — ακριβώς ό,τι κάνουν τα **FT8 map** και **FT4 map** για εκείνους τους τρόπους.

**Συμβουλή:** Το JS8 είναι πολύ πιο ήσυχο από το FT8. Ένα απόγευμα μπορεί να δείτε μία εκπομπή κάθε δύο λεπτά, και μεγάλα διαστήματα χωρίς τίποτα είναι φυσιολογικά. Δώστε του πέντε με δέκα λεπτά πριν συμπεράνετε ότι κάτι δεν πάει καλά. Τα 20m (14.078) και τα 40m (7.078) είναι τα συνηθισμένα σημεία.

**Συμβουλή:** Όπως και το FT8, το JS8 είναι αυστηρά συγχρονισμένο και χρησιμοποιεί το ρολόι του υπολογιστή σας. Αν η ώρα του συστήματος αποκλίνει πάνω από ένα-δύο δευτερόλεπτα, δεν θα αποκωδικοποιηθεί τίποτα. Κρατήστε την συγχρονισμένη με NTP.

## 5. CW — κώδικας Μορς

**Τι είναι:** ο αποκωδικοποιητής CW ακούει σήματα κώδικα Μορς (συνεχές κύμα) και τα μετατρέπει σε κείμενο σε πραγματικό χρόνο. Παρακολουθεί αυτόματα τη συχνότητα του σήματος και προσαρμόζεται στην ταχύτητα εκπομπής του χειριστή.

## Συνιστώμενες συχνότητες

Το CW είναι ενεργό σε όλες τις ερασιτεχνικές μπάντες, συνήθως στο κατώτερο τμήμα κάθε μπάντας. Συνήθη σημεία:

Μπάντα	Τμήμα
40 m	7.000–7.040 MHz
20 m	14.000–14.070 MHz
15 m	21.000–21.080 MHz

## Ρύθμιση

1. Συντονιστείτε σε ένα σήμα CW χρησιμοποιώντας τον τρόπο αποδιαμόρφωσης **CW** ή **CW-L** κατά περίπτωση.
2. Ενεργοποιήστε τον αποκωδικοποιητή και επιλέξτε **CW** από το μενού. Ή απλώς πατήστε το κουμπί **CW**.
3. Ο πίνακας **CW Decoder** εμφανίζεται από κάτω.

## Ανάγνωση των αποτελεσμάτων

- Η κεφαλίδα του πίνακα δείχνει την εντοπισμένη συχνότητα του σήματος σε Hz (π.χ.  $\approx 700$  Hz) και την εκτιμώμενη ταχύτητα εκπομπής σε λέξεις ανά λεπτό (π.χ.  $\cdot 22$  WPM).
- Αν δεν εντοπιστεί σήμα, η κεφαλίδα δείχνει **scanning...**
- Σε άδειο κανάλι δεν τυπώνεται τίποτα, και η ταχύτητα και ο τόνος δεν εμφανίζονται. Ο αποκωδικοποιητής αρχίζει να τυπώνει μόνο όταν υπάρχει σαφής διαφορά ανάμεσα σε πατημένο και ελεύθερο πλήκτρο, κάτι που ο σκέτος θόρυβος δεν παράγει, ενώ ένα σήμα ακόμη αναγνώσιμο στα περίπου  $-10$  dB το έχει άφθονο.
- Το αποκωδικοποιημένο κείμενο κυλά με κεχριμπαρένιους χαρακτήρες σταθερού πλάτους. Ο δρομέας που αναβοσβήνει (■) δείχνει πού γράφεται αυτή τη στιγμή το κείμενο.
- Κάντε κλικ στο **Clear** για να διαγράψετε τον ενταμιευτή εξόδου.

**Συμβουλές:** - Κεντράρετε το εύρος διέλευσης στον τόνο CW. Ο αποκωδικοποιητής αποδίδει καλύτερα όταν το σήμα CW βρίσκεται περίπου μεταξύ 400 και 900 Hz στο ηχητικό φάσμα. - Διαβάζονται ταχύτητες από περίπου 8 έως 40 WPM, και η εκπομπή Farnsworth (χαρακτήρες σταλμένοι γρήγορα με μεγαλύτερα κενά ανάμεσά τους): το κλείδωμα ταχύτητας κρατά και τα γράμματα δεν σπάνε σε χωριστές λέξεις. Το έντονα χειροκίνητο (ακανόνιστο) Morse μπορεί ακόμη να μειώσει την ακρίβεια. - Ο αποκωδικοποιητής αποδίδει καλύτερα σε ένα μοναδικό καθαρό σήμα. Ισχυρό QRM από γειτονικά σήματα στην ίδια μπάντα μπορεί να τον μπερδέψει.

## 6. QRSS Grabber

**Τι είναι:** QRSS είναι CW σταλμένο τόσο αργά ώστε μια τελεία να διαρκεί δευτερόλεπτα αντί για χιλιοστά του δευτερολέπτου. Σε αυτή την ταχύτητα το σήμα καταλαμβάνει μόνο ένα κλάσμα του hertz, οπότε μια αρκετά στενή ανάλυση μπορεί να βγάλει το ήχος από 20–30 dB κάτω από τη στάθμη θορύβου. Δεν υπάρχει τίποτα να διαβαστεί με το αυτί — το QRSS **το βλέπεις**, δεν το ακούς. Ο grabber τρέχει δικό του πολύ μακρύ FFT πάνω στον ήχο του δέκτη και ζωγραφίζει την κλασική εικόνα grabber: συχνότητα στον κατακόρυφο άξονα, χρόνος να κυλάει από αριστερά προς τα δεξιά, με τη νεότερη στήλη πάντα στη δεξιά άκρη.

Τα περισσότερα από όσα θα δείτε προέρχονται από φάρους MEPT (Manned Experimental Propagation Transmitter) — πομπούς πολύ χαμηλής ισχύος, συχνά λίγων εκατοντάδων milliwatt σε ένα σύρμα, που αναγνωρίζονται συνεχώς σε πολύ αργό Μορς ή με μοτίβα FSK.

## Πού να ψάξετε

Οι φάροι QRSS ζουν σε στενά παράθυρα 100–200 Hz κοντά στο κάτω άκρο κάθε μπάντας. Δεν χρειάζεται να τα θυμάστε: ο grabber έχει λίστα **Band** με τα παρακάτω παράθυρα, και η επιλογή ενός συντονίζει εκεί και ετοιμάζει τον δέκτη γι' αυτό (βλ. *Άνοιγμα του grabber*).

Μπάντα	Κέντρο παραθύρου	Σημείωση
30m	10.140,00 kHz	Το κύριο παράθυρο QRSS — με διαφορά το πιο πολυσύχναστο, μέρα και νύχτα
40m	7.039,90 kHz	
40m	7.000,85 kHz	Παράθυρο Knights, 7.000,8–7.000,9 kHz
20m	14.096,90 kHz	
80m	3.569,90 kHz	
80m	3.568,60 kHz	Παλαιότερο παράθυρο
80m	3.500,85 kHz	Παράθυρο Knights, 3.500,8–3.500,9 kHz
160m	1.837,90 kHz	
160m	1.843,30 kHz	Παλαιότερο παράθυρο
630m	476,10 kHz	
2200m	137,70 kHz	QRSS / DFCW στα LF
60m	5.288,55 kHz	
17m	18.105,90 kHz	
15m	21.095,90 kHz	
12m	24.925,90 kHz	
10m	28.125,70 kHz	
10m	28.000,85 kHz	Παλαιότερο παράθυρο
10m	28.322,00 kHz	Εναλλακτικό
6m	50.294,30 kHz	

Αρκετές μπάντες εμφανίζονται δύο φορές επειδή όντως χρησιμοποιούνται δύο συμβάσεις. Τα νεότερα παράθυρα βρίσκονται 200 Hz κάτω από τη συχνότητα WSPR της μπάντας· τα παλαιότερα παράθυρα *Knights* βρίσκονται εντελώς αλλού, και στα 40 και 80 m πολύ χαμηλότερα. Αν μια μπάντα είναι ήσυχη στο ένα, δοκιμάστε το άλλο.

Πρόκειται για συμβάσεις, όχι κανονισμούς, και κάποιες είναι τοπικές — ιδίως οι τιμές των 10 m διαφέρουν. Πάρτε τη λίστα ως αφετηρία και ακολουθήστε την τοπική πρακτική.

## Άνοιγμα του grabber

Ο QRSS grabber **δεν** βρίσκεται στο μενού των αποκωδικοποιητών. Έχει δικό του τμήμα **QRSS**, δίπλα στα χειριστήρια του φασματογράφου και των αποκωδικοποιητών.

1. Πατήστε **Show**. Εμφανίζεται ο καμβάς του grabber και τα χειριστήριά του ξεδιπλώνονται δίπλα στο κουμπί.

2. Διαλέξτε παράθυρο από τη λίστα **Band** και πατήστε **Tune**. Αυτό κάνει με μία κίνηση ό,τι χρειάζεται ο τρόπος λήψης: συντονίζει στο παράθυρο, γυρίζει τον δέκτη σε **CW** και θέτει εύρος διέλευσης όσο ακριβώς χρειάζεται για τη λωρίδα που παρακολουθείτε. Η ένδειξη δείχνει τότε την πραγματική συχνότητα QRSS, και τα ίχνη πέφτουν στην κεντρική γραμμή της οθόνης.
3. Ρυθμίστε την **Speed** ώστε να ταιριάζει με τη διάρκεια τελείας του φάρου.
4. Τα **Centre** και **Span** δουλεύουν όπως πριν· αλλάζοντας οποιοδήποτε από τα δύο αναδιαμορφώνεται και το εύρος διέλευσης, ώστε οθόνη και δέκτης να μένουν συντονισμένα μεταξύ τους.
5. Πατήστε **Hide** για διακοπή. Ο κανονικός τρόπος λήψης της μπάντας επιστρέφει.

Μπορείτε ακόμη να το κάνετε και με το χέρι — συντονίστε σε **USB** περίπου 1 kHz κάτω από το παράθυρο ώστε τα ίχνη να πέσουν κοντά στα 800 Hz ήχου. Η λίστα **Band** απλώς σας γλιτώνει τους υπολογισμούς.

Όσο είναι συντονισμένο παράθυρο από τη λίστα, ο grabber κρατά τον δέκτη όπως κάνει ένας αποκωδικοποιητής: το CW και το εύρος του επιβιώνουν του επανασυντονισμού, και το παράθυρο ακολουθεί τον συντονισμό, ώστε το ίχνος να μένει στην κεντρική γραμμή καθώς ψάχνετε κατά μήκος της μπάντας. Παραδίδει τον δέκτη όταν κρύψετε τον grabber, πατήσετε κουμπί mode ή ξεκινήσετε αποκωδικοποιητή.

## Ταχύτητες

Ρύθμιση	Διάρκεια τελείας	Διακριτική ικανότητα	Παράθυρο ανάλυσης	Νέα στήλη κάθε
QRSS 3	3 s	0,73 Hz	1,4 s	0,7 s
QRSS 6	6 s	0,37 Hz	2,7 s	1,4 s
QRSS 10	10 s	0,18 Hz	5,5 s	2,7 s
QRSS 30	30 s	0,09 Hz	10,9 s	5,5 s
QRSS 60	60 s	0,05 Hz	21,8 s	10,9 s

Ρύθμιση ταχύτερη από τον φάρο σπαταλά ευαισθησία — το ίχνος βγαίνει λεπτό και θορυβώδες. Ρύθμιση πιο αργή από τον φάρο μπουτσοιώνει διαδοχικές τελείες και παύλες σε μία μπάρα. Το πάνελ ανοίγει στο **QRSS 6**, μια λογική αφετηρία για άγνωστο σήμα: πιο ευαίσθητο από το QRSS 3, ενώ ανανεώνεται γρηγορότερα και ανέχεται καλύτερα την ολίσθηση από το QRSS 10. Μόλις διακρίνετε το σχήμα του χειρισμού, πηγαίνετε στην ταχύτητα που πράγματι στέλνει ο φάρος — κάθε βήμα προς ταχύτερη ρύθμιση κοστίζει 3 dB ευαισθησίας.

## Χειριστήρια

Χειριστήριο	Τι κάνει
<b>Band</b>	Τα παράθυρα QRSS του παραπάνω πίνακα. Το <b>Tune</b> πηγαίνει στο επιλεγμένο, σε CW με ανάλογο εύρος διέλευσης.
<b>Speed</b>	Ορίζει το μήκος του μετασχηματισμού — τον συμβιβασμό ανάμεσα σε διακριτική ικανότητα συχνότητας και χρόνου (πίνακας παραπάνω).
<b>Centre</b>	Ακουστική συχνότητα στο μέσο της οθόνης, 100–3000 Hz.
<b>Span</b>	Ύψος της εμφανιζόμενης λωρίδας: 20, 50, 100 ή 200 Hz, με αρχική τιμή 100 Hz — το πλάτος μιας υποζώνης QRSS. Μικρότερο span απλώνει κάθε ίχνος σε περισσότερα pixels. Στα μεγαλύτερα span χωρούν στο πάνελ περισσότερα bin από όσες γραμμές pixels έχει· τότε κάθε γραμμή δείχνει το ισχυρότερο bin που καλύπτει, ώστε να μην μπορεί να κρυφτεί τίποτα ανάμεσα στις γραμμές.
<b>Gain</b>	−10 έως +40 dB. Μετατοπίζει τον χρωματικό χάρτη σε σχέση με τη μετρημένη στάθμη θορύβου· αυξήστε το για να φωτίσετε αμυδρά ίχνη.
<b>Color</b>	Rainbow, Green ή Grayscale.
<b>Clear</b>	Καθαρίζει τον καμβά και μηδενίζει την αναφορά θορύβου.

## Πώς διαβάζεται η οθόνη

Οι υποδιαίρέσεις συχνότητας τρέχουν στην αριστερή άκρη, με την υψηλότερη συχνότητα πάνω. Κάτω από τον καμβά μια γραμμή κατάστασης δείχνει την επιλεγμένη ταχύτητα, τη διακριτική ικανότητα που όντως χρησιμοποιείται (π.χ.  $0.183 \text{ Hz/bin} \cdot 5.5 \text{ s window}$ ) και τον ρυθμό στηλών (π.χ.  $2.7 \text{ s/column}$ ).

Ο χρωματικός χάρτης αναφέρεται στον **τρέχοντα** θόρυβο, που μετριέται συνεχώς από τα άδεια τμήματα κάθε στήλης, και όχι σε απόλυτη στάθμη. Έτσι αλλαγές σε ένταση, AGC, κεραία ή μπάντα δεν απαιτούν εκ νέου ρύθμιση του Gain — η οθόνη κρατά σταθερή αντίθεση απέναντι σε όποιον θόρυβο έχει η μπάντα. Όπως και στους αποκωδικοποιητές, ο ήχος λαμβάνεται πριν από AGC, μείωση θορύβου και σίγαση, οπότε μπορείτε να έχετε τον δέκτη στη σίγαση και να συνεχίζετε να βλέπετε.

Τι σημαίνουν τα σχήματα:

- Ένα **σταθερό φέρον** ζωγραφίζει ευθεία οριζόντια γραμμή.
- Το **αργό Μορς** ζωγραφίζει αυτή τη γραμμή ως αλυσίδα κοντών και μακριών τμημάτων — τελείες και παύλες που διαβάζονται από αριστερά προς τα δεξιά.
- **Καμπύλα ή μετατοπιζόμενα ίχνη** είναι ένας μη σταθεροποιημένος ταλαντωτής φάρου που ζεσταίνεται ή κρυώνει. Είναι φυσιολογικό, και συχνά αυτή ακριβώς η υπογραφή ολίσθησης είναι που κάνει έναν τακτικό παρατηρητή να αναγνωρίσει τον σταθμό.
- **Κατακόρυφες γραμμώσεις** σε όλη τη λωρίδα είναι στατικά ή τοπικές ριπές θορύβου, όχι σήμα.

**Συμβουλές:** - Κάντε υπομονή. Στο QRSS 30 μία στήλη παίρνει 5,5 δευτερόλεπτα, οπότε ένα ολόκληρο διακριτικό μπορεί να θέλει δέκα λεπτά ή περισσότερο για να διασχίσει την οθόνη. Αφήστε το να δουλεύει. - Ο συντονισμός από τη λίστα **Band** έχει ήδη στενέψει το εύρος γύρω από το παράθυρο. Αν συντονίσατε με το χέρι, κάντε το εσείς: δεν αλλάζει αυτό που αναλύει ο μετασχηματισμός, αλλά εμποδίζει ισχυρούς γείτονες να δουλεύουν το AGC του δέκτη. - Ο grabber είναι ανεξάρτητος από τους αποκωδικοποιητές — μπορείτε να τον αφήσετε αναμμένο ενώ ένας αποκωδικοποιητής δουλεύει σε κάτι άλλο. - Μεγάλο span με αργή ταχύτητα είναι ο βαρύτερος συνδυασμός· όλος ο μετασχηματισμός τρέχει στον browser, οπότε σε μέτριο μηχάνημα προτιμήστε span 50 Hz.

## 7. WSPR

**Τι είναι:** το WSPR (Weak Signal Propagation Reporter, προφέρεται «whisper») είναι τρόπος φάρου εξαιρετικά ασθενών σημάτων που χαρτογραφεί τις διαδρομές διάδοσης HF παγκοσμίως. Κάθε εκπομπή διαρκεί περίπου 110 δευτερόλεπτα και χωρά σε θυρίδα εύρους 200 Hz. Ο αποκωδικοποιητής περιμένει μια πλήρη θυρίδα 2 λεπτών ευθυγραμμισμένη με το UTC πριν αποκωδικοποιηθεί.

### Συνιστώμενες συχνότητες (USB, ένδειξη)

Μπάντα	Συχνότητα ένδειξης
160 m	1.836.600 MHz
80 m	3.568.600 MHz
40 m	7.038.600 MHz
30 m	10.138.700 MHz
20 m	14.095.600 MHz
17 m	18.104.600 MHz
15 m	21.094.600 MHz

### Ρύθμιση

1. Συντονιστείτε σε μια από τις παραπάνω συχνότητες ένδειξης WSPR και θέστε τον τρόπο σε **USB**.
2. Το σήμα WSPR καταλαμβάνει την ηχητική περιοχή 1400-1600 Hz. Η επιλογή του **WSPR** κάνει τα υπόλοιπα για εσάς: ο δέκτης περνά σε USB και το εύρος διέλευσης στενεύει στα 1350-1650 Hz, που καλύπτει ολόκληρη την περιοχή στην οποία ψάχνει ο αποκωδικοποιητής. Δεν χρειάζεται περαιτέρω ρύθμιση.
3. Ενεργοποιήστε τον αποκωδικοποιητή και επιλέξτε **WSPR** από το μενού. Ή απλώς πατήστε το κουμπί **WSPR**.
4. Ο πίνακας **WSPR-2 Decoder** εμφανίζεται από κάτω.

### Ανάγνωση των αποτελεσμάτων

Ο πίνακας δείχνει μια γραμμή προόδου για την τρέχουσα θυρίδα 2 λεπτών:

- **Κυανή γραμμή που γεμίζει** — συλλογή δεδομένων σήματος (0-116 s μέσα στη θυρίδα).
- **Κεκριμπαρένια γραμμή που πάλλεται** — αποκωδικοποίηση σε εξέλιξη (τα τελευταία ~4 s της θυρίδας).
- **Κενή γραμμή** — αναμονή για το επόμενο άρτιο λεπτό UTC.

Κάθε επιτυχώς αποκωδικοποιημένο spot εμφανίζεται σε πίνακα με τις εξής στήλες:

Στήλη	Σημασία
UTC	Ωρα του spot (άρτιο λεπτό)
Callsign	Ο σταθμός που εξέπεμψε
Grid	Locator Maidenhead του πομπού
Power	Ισχύς εκπομπής σε dBm
Freq	Ακριβής ηχητική συχνότητα (Hz) εντός του εύρους διέλευσης WSPR
SNR	Λόγος σήματος προς θόρυβο σε dB

Κάντε κλικ στο **Clear** για να διαγράψετε τη λίστα των spot.

**Συμβουλή:** η αποκωδικοποίηση WSPR απαιτεί πολύ ακριβή ώρα συστήματος (εντός  $\pm 1$  δευτερολέπτου από το UTC). Η πρώτη θυρίδα μετά την ενεργοποίηση του αποκωδικοποιητή θα ξεκινήσει στο επόμενο άρτιο λεπτό UTC — μια σύντομη αναμονή είναι φυσιολογική.

## 8. HF FAX / WEFAX

**Τι είναι:** το HF Radiofax (γνωστό και ως WEFAX) χρησιμοποιείται από ακτοφυλακές και μετεωρολογικές υπηρεσίες παγκοσμίως για τη μετάδοση μετεωρολογικών χαρτών, χαρτών κατάστασης θάλασσας και αναλύσεων επιφανείας μέσω βραχέων κυμάτων. Ο αποκωδικοποιητής ανασυνθέτει την εικόνα γραμμή προς γραμμή καθώς λαμβάνεται.

### Ρύθμιση

1. Ενεργοποιήστε τον αποκωδικοποιητή και επιλέξτε **HF FAX / WEFAX** από το μενού. Ή απλώς πατήστε το κουμπί **FAX**.
2. Ο πίνακας **HF FAX / WEFAX Receiver** εμφανίζεται από κάτω.
3. **Επιλέξτε σταθμό** από το αναπτυσσόμενο μενού Station. Είναι διαθέσιμοι πάνω από 20 σταθμοί που καλύπτουν Ευρώπη, Ασία, Ωκεανία και Αμερική (π.χ. DDH3/DDK3 Γερμανία, SVJ4/GR Ελλάδα, JMH Ιαπωνία, NMG ΗΠΑ Νέα Ορλεάνη). Περιλαμβάνεται επίσης ο CBM Punta Arenas (Χιλή). Τα προγράμματα ακολουθούν το παγκόσμιο πρόγραμμα radiofax της NOAA (έκδοση της 7ης Μαρτίου 2025)· το Charleville (VMC, Αυστραλία) περιλαμβάνει τις επιπλέον εκπομπές του που γίνονται μόνο στα 11030 kHz.
4. Αν ο σταθμός εκπέμπει σε περισσότερες από μία συχνότητες, επιλέξτε την επιθυμητή από το υπομενού **Frequency**.
5. Κάντε κλικ στο ► **Tune** για αυτόματο συντονισμό του καταρράκτη σε αυτόν τον σταθμό.
6. Ο τρόπος τίθεται αυτόματα σε **USB**.

### Πρόγραμμα εκπομπών

Όταν επιλέγεται σταθμός, εμφανίζεται πίνακας αντίστροφης μέτρησης **Next Transmissions** με τις επόμενες 4 προγραμματισμένες εκπομπές σε UTC και ζωντανή αντίστροφη μέτρηση:

- Κανονικό (γκρι/πράσινο) — η εκπομπή πλησιάζει.
- **Κεχριμπαρένιο** — η εκπομπή ξεκινά σε λιγότερο από 3 λεπτά· οπλίστε τώρα τον αποκωδικοποιητή.
- **Κόκκινο** ● — η εκπομπή ξεκινά σε λιγότερο από 30 δευτερόλεπτα· η λήψη είναι άμεση.

### Παράμετροι

Οι περισσότεροι σταθμοί χρησιμοποιούν τις τυπικές προεπιλογές (με ★). Αλλάξτε τις μόνο αν γνωρίζετε ότι ο σταθμός χρησιμοποιεί μη τυπικές ρυθμίσεις.

Παράμετρος	Προεπιλογή	Περιγραφή
LPM	120 ★	Γραμμές ανά λεπτό — η ταχύτητα περιστροφής του τυμπάνου
IOC	576 ★	Δείκτης συνεργασίας — καθορίζει τα εικονοστοιχεία ανά γραμμή
Shift	800 Hz ★	Μετατόπιση συχνότητας μεταξύ των τόνων μαύρου και λευκού

## Χειριστήρια

- ⇌ **Auto-align** — ενεργό εξ ορισμού. Συγχρονίζεται αυτόματα με το σήμα φάσης στην αρχή κάθε εικόνας. Απενεργοποιήστε το μόνο αν αντιμετωπίζετε προβλήματα ευθυγράμμισης σε σήμα που ξέρετε ότι είναι καλό.
- ⌛ **Invert** — αντιστρέφει μαύρο και λευκό. Χρησιμοποιήστε το αν η εικόνα εμφανίζεται ως αρνητικό (λευκές περιοχές εκεί που θα έπρεπε να είναι μαύρες).
- ↻ **Refresh** — καθαρίζει τον καμβά και επαναφέρει τον αποκωδικοποιητή. Χρησιμοποιήστε το μεταξύ εκπομπών ή αν η εικόνα σκίζεται ή παρασύρεται.
- ⬇ **Save PNG** — αποθηκεύει τον τρέχοντα καμβά ως αρχείο PNG στον υπολογιστή σας.

## Ενδείξεις κατάστασης

Στο κάτω μέρος της εικόνας, δύο ενδείξεις τόνου δείχνουν:

- **300 Hz phasing** — ανάβει κυανό όταν εντοπιστεί ο τόνος φάσης έναρξης εικόνας.
- **450 Hz stop** — ανάβει κόκκινο όταν εντοπιστεί ο τόνος τερματισμού εικόνας.

Και τα δύο σήματα αναγνωρίζονται από τον ρυθμό τους σε όλο το πλάτος μιας γραμμής, που τα πραγματικά σήματα έναρξης και λήξης το γεμίζουν πάντα ενώ το περιεχόμενο ενός χάρτη πρακτικά ποτέ, οπότε οι γραμμοσκιασμένες περιοχές ενός χάρτη δεν περνιούνται για σήμα έναρξης. Αναγνωρίζονται έως περίπου 0 dB SNR, και σε πολύ ασθενή σήματα η εικόνα ευθυγραμμίζεται από τον μέσο όρο αρκετών γραμμών phasing, ώστε ο χάρτης να ξεκινά και πάλι από το αριστερό άκρο.

**Σημείωση:** η εικόνα κυλά προς τα πάνω — η πιο πρόσφατη γραμμή εμφανίζεται πάντα στο κάτω μέρος του καμβά. Αν δείτε **[PHASING]** στην κεφαλίδα, ο αποκωδικοποιητής κλείδωσε σε νέα έναρξη εικόνας.

## 9. NAVTEX

**Τι είναι:** το NAVTEX είναι το διεθνές ναυτιλιακό σύστημα εκπομπής πληροφοριών παράκτιας ασφάλειας — προειδοποιήσεις ναυσιπλοΐας, μετεωρολογικές προγνώσεις και ειδοποιήσεις έρευνας και διάσωσης. Χρησιμοποιεί FSK 100 baud (SITOR-B με FEC) και λαμβάνεται σε ειδικά κανάλια παγκοσμίως.

### Διαθέσιμα κανάλια

Κανάλι	Συχνότητα	Χρήση
Διεθνές	518 kHz	Αγγλόφωνο, διεθνές
Εθνικό	490 kHz	Εκπομπές στην εθνική γλώσσα
HF (×5)	4209.5 / 6314 / 8416.5 / 12579 / 16806.5 kHz	NAVTEX HF μεγάλης εμβέλειας

### Ρύθμιση

1. Ενεργοποιήστε τον αποκωδικοποιητή και επιλέξτε **NAVTEX** από το μενού. Ή απλώς πατήστε το κουμπί **NAVTEX**. Ο δέκτης μεταβαίνει αυτόματα σε **USB**.
2. Εμφανίζεται ο πίνακας **NAVTEX Receiver**.
3. Επιλέξτε το επιθυμητό κανάλι από το μενού **Station** (π.χ. International – 518 kHz).
4. Κάντε κλικ στο ⇌ **Tune & Set IF** για αυτόματο συντονισμό του καταρράκτη και στένωση του εύρους διέλευσης στο σωστό ηχητικό παράθυρο. Ο τρόπος τίθεται αυτόματα σε **USB**. Η ένδειξη τοποθετείται 500 Hz κάτω από το κέντρο του καναλιού, ώστε το σήμα NAVTEX να εμφανίζεται στα 500 Hz στον ήχο.

## Πρόγραμμα εκπομπών

Ο πίνακας προγράμματος παραθέτει όλους τους γνωστούς σταθμούς του επιλεγμένου καναλιού με:

- το αναγνωριστικό γράμμα ITU και τη σημαία της χώρας
- την ώρα της επόμενης εκπομπής σε UTC
- ζωντανή αντίστροφη μέτρηση έως την επόμενη εκπομπή
- **Κεχριμπαρένιο** σε λιγότερο από 2 λεπτά / **Κόκκινο** ● σε λιγότερο από 30 δευτερόλεπτα — οπλίστε τώρα τον αποκωδικοποιητή

## Ανάγνωση των αποτελεσμάτων

Το αποκωδικοποιημένο κείμενο εμφανίζεται με γαλαζοπράσινους χαρακτήρες σταθερού πλάτους. Τα όρια των μηνυμάτων σημειώνονται καθαρά:

```
— ZCZC MA12 —
... message content ...
— NNNN —
```

Το ZCZC σημαίνει την αρχή ενός μηνύματος. Οι τέσσερις χαρακτήρες που ακολουθούν δηλώνουν τον σταθμό (M), το θέμα (A = ναυτιλιακές προειδοποιήσεις) και τον αύξοντα αριθμό (12). Η επικεφαλίδα περιμένει ολόκληρη την κεφαλίδα, οπότε γράφει ZCZC MA12 και όχι ZCZC ????. Το NNNN σημαίνει το τέλος.

Αν συντονιστείτε αφού έχει αρχίσει ένα μήνυμα, ή αν η κεφαλίδα του χαθεί στον θόρυβο, το κείμενο εμφανίζεται κανονικά, κάτω από μια επικεφαλίδα — **(header missed)** —. Αυτή ανοίγει μόνο όταν φτάσουν αρκετοί συνεχόμενοι χαρακτήρες στους οποίους συμφωνούν και οι δύο εκπομπές NAVTEX, κάτι που ο θόρυβος πρακτικά ποτέ δεν παράγει. Κλείνει κανονικά στο NNNN, με — **(signal lost)** — αν το σήμα χαθεί νωρίτερα, ή όταν αρχίσει νέο μήνυμα.

Κάντε κλικ στο **Clear** για να διαγράψετε τον ενταμιευτή μηνυμάτων.

**Σημείωση:** το NAVTEX λειτουργεί σε συχνότητες MF και LF (518/490 kHz). Η εμβέλεια λήψης είναι συνήθως 200-400 ναυτικά μίλια από τον πομπό. Τα κανάλια HF (4-17 MHz) προσφέρουν πολύ μεγαλύτερη εμβέλεια.

## 10. FSK / RTTY — μαζί με PSK31 και Olivia

**Τι είναι:** ένας αποκωδικοποιητής γενικής χρήσης για στενής ζώνης ψηφιακά κειμένου, με πέντε παραλλαγές λειτουργίας που επιλέγονται από ένα μενού. Οι τρεις είναι πραγματικό FSK (Frequency-Shift Keying): ναυτιλιακό FSK (SITOR), μετεωρολογικό RTTY και ερασιτεχνικό RTTY. Οι άλλες δύο δεν είναι καθόλου FSK αλλά μοιράζονται το ίδιο παράθυρο: το **PSK31**, που είναι διαμόρφωση φάσης, και το **Olivia**, που είναι πολυτονικό FSK με διόρθωση σφαλμάτων. Κάθε παραλλαγή συνοδεύεται από προρύθμιση προσαρμοσμένη στις τυπικές παραμέτρους της.

## Παραλλαγές και προρυθμίσεις

Παραλλαγή	Κέντρο	Shift	Baud	Πλαισίωση	Κωδικοποίηση
Ναυτιλιακό FSK / SITOR	500 Hz	170 Hz	100	7N1	CCIR-476
Μετεωρολογικό RTTY	1000 Hz	450 Hz	50	5N1.5	ITA2
Ερασιτεχνικό RTTY	1000 Hz	170 Hz	45.45	5N1.5	ITA2
PSK31 (BPSK)	1000 Hz	—	31.25	—	Varicode
Olivia (MFSK)	1000 Hz	—	βλ. Mode	—	7 bit + FEC

Ο πίνακας προσαρμόζεται στην παραλλαγή που επιλέγετε. Τα **Shift**, **Baud**, **Framing**, **Encoding**, **Invert mark / space** και **Auto shift detect** αποκρύπτονται για PSK31 και Olivia, επειδή καμία από τις δύο δεν έχει ζεύγος τόνων mark/space ούτε πλαισίωση τύπου UART. Στη θέση τους, το **Center audio** γίνεται πεδίο ελεύθερης καταχώρισης (το φέρον μπορεί να βρίσκεται οπουδήποτε μέσα στο εύρος διέλευσης), ενώ το Olivia αποκτά επιλογή **Mode** και ρυθμιστικό **Squelch**.

## Ρύθμιση

1. Ενεργοποιήστε τον αποκωδικοποιητή και επιλέξτε **FSK / RTTY** από το μενού. Ή απλώς πατήστε το κουμπί **RTTY**.
2. Εμφανίζεται ο πίνακας του αποκωδικοποιητή. Ο τίτλος του ακολουθεί την παραλλαγή — *FSK / RTTY Decoder*, *PSK31 Decoder* ή *Olivia Decoder*.
3. Επιλέξτε τη **Variant**. Οι παράμετροι ενημερώνονται αυτόματα.
4. Για το Olivia, ορίστε το **Mode** (τόνοι / εύρος ζώνης) ώστε να ταιριάζει με την εκπομπή — δείτε τις σημειώσεις για το Olivia παρακάτω.
5. Χρησιμοποιήστε το μενού **Known frequency** για να επιλέξετε μια συνήθη συχνότητα για την παραλλαγή που επιλέξατε και έπειτα κάντε κλικ στο **Tune** για να μεταβείτε σε αυτήν.
6. Κάντε λεπτό συντονισμό μέχρι το αποκωδικοποιημένο κείμενο να γίνει σταθερό και ευανάγνωστο.

## Γνωστές συχνότητες ανά παραλλαγή

### Ναυτιλιακό FSK / SITOR

- 518.0 kHz — διεθνές NAVTEX
- 490.0 kHz — εθνικό NAVTEX
- 4209.5 / 6314.0 / 8416.5 / 12579.0 / 16806.5 / 22376.0 kHz — SITOR HF

### Μετεωρολογικό RTTY

- 4583.0 / 7646.0 / 10100.8 / 11039.0 / 14467.3 kHz — DWD (Γερμανική Μετεωρολογική Υπηρεσία)

### Ερασιτεχνικό RTTY

- 3590 kHz (80 m), 7043 kHz (40 m), 10143 kHz (30 m), 14083 kHz (20 m), 21083 kHz (15 m), 28083 kHz (10 m)

### PSK31

- 3580.15 kHz (80 m), 7040.15 kHz (40 m), 10142.15 kHz (30 m), 14070.15 kHz (20 m), 18100.15 kHz (17 m), 21080.15 kHz (15 m), 24920.15 kHz (12 m), 28120.15 kHz (10 m)

### Olivia

- 3577.75 kHz (80 m), 7073.75 kHz (40 m), 10142.25 kHz (30 m), 14075.5 kHz (20 m), 18103.75 kHz (17 m), 21075.75 kHz (15 m), 24921.75 kHz (12 m), 28123.75 kHz (10 m)

## Παράμετροι

Παράμετρος	Ισχύει για	Περιγραφή
Center audio (Hz)	όλες	Η ηχητική συχνότητα του μέσου μεταξύ mark και space· για το PSK31 το φέρον, για το Olivia το κέντρο της ομάδας τόνων. Μενού για τις παραλλαγές FSK, πεδίο ελεύθερης καταχώρισης για PSK31 και Olivia
Shift (Hz)	μόνο FSK	Διαφορά συχνότητας μεταξύ των τόνων mark και space
Baud	μόνο FSK	Ρυθμός συμβόλων
Framing	μόνο FSK	Bit δεδομένων, ισοτιμία, bit τερματισμού (π.χ. 7N1 = 7 δεδομένων, χωρίς ισοτιμία, 1 stop)
Encoding	μόνο FSK	Σύνολο χαρακτήρων (CCIR-476, ITA2/Baudot ή ASCII)
Invert mark / space	μόνο FSK	Αντιστρέφει τους τόνους mark και space
Auto shift detect	μόνο FSK	Επιχειρεί αυτόματη μέτρηση του shift από το εισερχόμενο σήμα
Mode (τόνοι / Hz)	μόνο Olivia	Αριθμός τόνων και εύρος ζώνης — πρέπει να ταιριάζει ακριβώς με την εκπομπή
Squelch (FEC S/N)	μόνο Olivia	Πόσο ισχυρή πρέπει να είναι η διόρθωση σφαλμάτων πριν εκτυπωθεί κείμενο
Squelch (SNR)	Ερασιτεχνικό και μετεωρολογικό RTTY	Εμποδίζει τον θόρυβο να τυπώνεται ως τυχαία γράμματα. Ξεκινά στα $-5$ dB για το ερασιτεχνικό RTTY και στα $-8$ dB για το μετεωρολογικό· τέρμα αριστερά ( <b>off</b> ) απενεργοποιείται. Η ένδειξη SNR γίνεται γκρι όσο το squelch είναι κλειστό

## Μετρήσεις σήματος

Η γραμμή κατάστασης δείχνει ζωντανές μετρήσεις, και τα πεδία αλλάζουν ανάλογα με την παραλλαγή:

Παραλλαγή	Πεδία που εμφανίζονται
Παραλλαγές FSK	<b>Mark / Space</b> (μετρημένες συχνότητες τόνων), <b>SNR</b> , <b>Lock</b> , <b>Timing</b>
PSK31	<b>Carrier</b> (Hz, μετά την αυτόματη διόρθωση συχνότητας), <b>IMD</b> (dB), <b>SNR</b> , <b>Lock</b> , <b>Timing</b>
Olivia	<b>Centre</b> (Hz), <b>Mode</b> , <b>S/N</b> , <b>FEC</b> (%), <b>Sync</b>

Το Timing/Sync δείχνει LOCKED/SYNCED μόλις ο αποκωδικοποιητής παρακολουθεί το σήμα, και SEARCH όσο ακόμη αναζητά.

Το **SNR** των παραλλαγών FSK και του PSK31 μετρείται πάνω στο ίδιο το σήμα και δίνεται σε εύρος 3 kHz, ώστε και τα δύο να διαβάζονται στην ίδια κλίμακα: με ακρίβεια περίπου 1 dB από +20 dB έως  $-5$  dB. Μια άδεια συχνότητα δείχνει περίπου  $-12$  έως  $-15$  dB, και στο PSK31 τα 0 dB είναι ήδη ισχυρό σήμα. Το **S/N** του Olivia είναι κάτι άλλο: ο δείκτης ποιότητας της διόρθωσης σφαλμάτων.

Το ερασιτεχνικό RTTY (45,45 baud, 170 Hz) διαβάζεται χωρίς λάθη έως περίπου 0 dB, με περίπου 1,5% λαθασμένους χαρακτήρες στα  $-3$  dB και 20% στα  $-6$  dB.

## Πρόσθετα χειριστήρια

- ⇒ **Set IF Band-Pass** — στενεύει τη ζώνη διέλευσης του δέκτη ώστε να περικλείει στενά το σήμα. Το εύρος ακολουθεί την παραλλαγή: 300 Hz για το ερασιτεχνικό RTTY (αρκετά στενό ώστε να αφαιρεί έναν ισχυρότερο σταθμό 250 Hz δίπλα, διαβάζοντας ακόμη σήμα αποσυντονισμένο κατά 40 Hz), 594 Hz για το μετεωρολογικό RTTY, mark/space συν περιθώριο για το Maritime FSK, περίπου  $\pm 100$  Hz για το PSK31 και όλο το εύρος του μπλοκ τόνων συν περιθώριο για το Olivia. Η ζώνη

διέλευσης ακολουθεί τον αποκωδικοποιητή όταν το **Auto-tune** ή το **Auto shift detect** μετακινεί τους τόνους, και όταν αλλάζετε τη ρύθμιση **Baud**.

- **Auto-tune Center** — αυτόματη αναζήτηση του σήματος. Για τις παραλλαγές FSK αναζητά ισορροπημένο ζεύγος τόνων, για το PSK31 εντοπίζει το φέρον, και για το Olivia βρίσκει την ισχυρότερη ομάδα του επιλεγμένου εύρους ζώνης. Το **Auto shift detect** αλλάζει το shift μόνο όταν υπάρχει πραγματικό σήμα δύο τόνων, οπότε δεν πηδά πια πάνω στον θόρυβο.

## Σημειώσεις για το PSK31

Το PSK31 είναι ο πιο διαδεδομένος τρόπος επικοινωνίας πληκτρολογίου σε πληκτρολόγιο στα HF. Καταλαμβάνει μόλις 62 Hz, οπότε αρκετές επαφές συνυπάρχουν δίπλα δίπλα μέσα σε λίγες εκατοντάδες hertz γύρω από τη συνήθη συχνότητα συνάντησης, και ο συντονισμός είναι θέμα επιλογής ενός ίχνους μέσα από την ομάδα στον καταρράκτη.

- Ο αποκωδικοποιητής διορθώνει μόνος του το σφάλμα συντονισμού σε εύρος περίπου **±25 Hz**, οπότε αρκεί να πλησιάσετε. Το **Carrier** στη γραμμή μετρήσεων δείχνει πού τελικά σταθεροποιήθηκε.
- Το **IMD** μετρά την ποιότητα του πομπού, όχι της λήψης: ένα καθαρό σήμα δείχνει περίπου **-20 dB** ή καλύτερα. Κακή ένδειξη σημαίνει ότι ο άλλος σταθμός υπερδιεγείρει τον πομπό του, όχι ότι εσείς είστε εκτός συντονισμού.
- Το κείμενο εμφανίζεται χαρακτήρα προς χαρακτήρα χωρίς διόρθωση σφαλμάτων, οπότε ένα ασθενές σήμα υποβαθμίζεται σε περιστασιακά λάθος γράμματα αντί να σταματά.

## Σημειώσεις για το Olivia

Το Olivia θυσιάζει ταχύτητα για ανθεκτικότητα. Είναι πολύ πιο αργό από το PSK31 αλλά αποκωδικοποιεί σήματα που δεν ακούγονται με το αυτί, γι' αυτό και προτιμάται για ασθενή σήματα και επαφές μεγάλων αποστάσεων.

- **Η ρύθμιση Mode πρέπει να ταιριάζει ακριβώς με την εκπομπή.** Λανθασμένος συνδυασμός τόνων/εύρους ζώνης δεν αποκωδικοποιεί απολύτως τίποτε — όχι ακατάληπτο κείμενο, αλλά σιωπή. Ο πίνακας ανοίγει στο **8 / 250**, τη στενή διαμόρφωση που μένει ανοιχτή στις συχνότητες κλήσης· τα **16 / 500** και **32 / 1000** είναι τα άλλα δύο σε κοινή χρήση, ενώ προσφέρεται και το **16 / 1000**.
- Το Olivia δεν στέλνει προοίμιο, οπότε ο αποκωδικοποιητής πρέπει να αναζητήσει τον συγχρονισμό. **Αφήστε λίγα δευτερόλεπτα** μετά τον συντονισμό πριν εμφανιστεί κείμενο. Το πεδίο **Sync** δείχνει SEARCH μέχρι να κλειδώσει.
- Η διόρθωση σφαλμάτων λειτουργεί σε μπλοκ, οπότε το κείμενο καταφθάνει σε **ριπές αντί για συνεχή ροή**, με καθυστέρηση αρκετών μπλοκ μεταξύ εκπομπής και εμφάνισης.
- Το **Squelch (FEC S/N)** ορίζει πόσο σίγουρη πρέπει να είναι η διόρθωση σφαλμάτων πριν εκτυπώσει. Η προεπιλογή 4.0 κρατά τον θόρυβο έξω· το 3.0 είναι το κατώτατο όριο, κάτω από το οποίο ο τυχαίος θόρυβος αρχίζει να τυπώνει περιστασιακούς χαρακτήρες. Ένα καλό σήμα δείχνει 8-9 στο μετρητή **FEC**, οπότε υπάρχει άνετο περιθώριο να ανεβάσετε το squelch σε πολυσύχναστη μπάνα.

**Σημείωση για τον τρόπο λειτουργίας:** ο αποκωδικοποιητής αναλαμβάνει τον έλεγχο του τρόπου αποδιαμόρφωσης και του εύρους διέλευσης IF όσο είναι ενεργός. Και τα δύο αποκαθίστανται αυτόματα όταν τον απενεργοποιήσετε. Και οι πέντε παραλλαγές χρησιμοποιούν **USB**.

**Σημείωση για την πολικότητα (μόνο για τις παραλλαγές FSK):** για μετεωρολογικό RTTY συνήθως χρειάζεται να επιλέξετε το **Invert mark / space**. Για ναυτιλιακό FSK (τύπου SITOR/NAVTEX) και ερασιτεχνικό RTTY αφήστε το ξεμαρκάριστο — το ερασιτεχνικό RTTY στέλνει το mark ως την υψηλότερη ραδιοσυχνότητα, και το USB το διατηρεί ως τον υψηλότερο ηχητικό τόνο, που είναι ακριβώς η ξεμαρκάριστη περίπτωση. Αν το αποκωδικοποιημένο κείμενο είναι ακατάληπτο, το πρώτο πράγμα που πρέπει να δοκιμάσετε είναι να αλλάξετε αυτό το πλαίσιο. Το πλαίσιο αποκρύπτεται για PSK31 και Olivia, που δεν έχουν ζεύγος mark/space.

**Σημείωση για γράμματα/ψηφία (μόνο για τις παραλλαγές FSK):** ο κώδικας Baudot διατηρεί γράμματα και ψηφία σε δύο ξεχωριστές καταστάσεις, και ο θόρυβος μπορεί να μεταφέρει τον αποκωδικοποιητή στη λάθος — κάτι που αλλοιώνει κάθε επόμενο χαρακτήρα, όχι μόνο τον χαλασμένο. Γι' αυτό ο αποκωδικοποιητής επιστρέφει στα γράμματα σε κάθε κενό διάστημα· είναι καθιερωμένη πρακτική και αποκαθιστά μια χαλασμένη μετατόπιση μέσα σε μία ή δύο λέξεις αντί για ολόκληρη γραμμή. Το τίμημα είναι ότι οι ομάδες ψηφίων που χωρίζονται με κενά απαιτούν από τον πομπό να επαναλαμβάνει τη μετατόπιση ψηφίων μετά από κάθε κενό — όπως συνήθως κάνουν οι πομποί.

## 11. SSTV

**Τι είναι:** η τηλεόραση αργής σάρωσης μεταδίδει ακίνητες εικόνες μέσω ενός συνηθισμένου φωνητικού καναλιού SSB, μία γραμμή τη φορά, ως τόνο διαμορφωμένο κατά συχνότητα μεταξύ 1500 Hz (μαύρο) και 2300 Hz (λευκό). Μια πλήρης εικόνα διαρκεί από 36 δευτερόλεπτα έως 4,5 λεπτά, ανάλογα με τον τρόπο.

### Συνιστώμενες συχνότητες (USB)

Μπάντα	Συχνότητα	Τρόπος	Σημειώσεις
20 m	14.230 MHz	USB	Η κύρια διεθνής συχνότητα κλήσης SSTV — μακράν η πιο ενεργή
20 m	14.233 MHz	USB	Δευτερεύουσα, όταν η 14.230 είναι κατειλημμένη
15 m	21.340 MHz	USB	
10 m	28.680 MHz	USB	Ενεργή σε ανοίγματα της μπάντας
40 m	7.171 MHz	LSB	
80 m	3.845 MHz	LSB	Περιφερειακή, τα βράδια

**Η πλευρική ζώνη επιλέγεται για εσάς.** Η εκκίνηση του αποκωδικοποιητή επιλέγει **LSB κάτω από 10 MHz** και **USB πάνω από αυτά**, σύμφωνα με τη συνήθη ερασιτεχνική πρακτική. Σε λάθος πλευρική ζώνη η αντιστοίχιση των τόνων είναι αντεστραμμένη και η εικόνα δεν θα αποκωδικοποιηθεί. Αν συναντήσετε σταθμό που αγνοεί τη σύμβαση, απλώς αλλάξετε την πλευρική ζώνη χειροκίνητα — ο αποκωδικοποιητής συνεχίζει να λειτουργεί, καθαρίζει το καρέ και ξεκινά από την αρχή με τη νέα ρύθμιση.

### Ρύθμιση

1. Ενεργοποιήστε τον αποκωδικοποιητή και επιλέξτε **SSTV** από το μενού. Ή απλώς πατήστε το κουμπί **SSTV**. Ο δέκτης αλλάζει πλευρική ζώνη αυτόματα — USB πάνω από 10 MHz, LSB κάτω από αυτά.

2. Συντονιστείτε ώστε οι τόνοι της εικόνας να πέφτουν στο μέσο του εύρους διέλευσης. Ένα σωστά συντονισμένο σήμα έχει τους παλμούς συγχρονισμού στα 1200 Hz και το περιεχόμενο της εικόνας μεταξύ 1500 και 2300 Hz.
3. Αφήστε το **Mode** στο **Auto** εκτός αν γνωρίζετε ήδη τι εκπέμπεται. Η εικόνα χτίζεται γραμμή προς γραμμή καθώς λαμβάνεται.

## Τρόποι

Ρύθμιση	Εικόνα	Διάρκεια
<b>Auto</b>	Εντοπίζεται αυτόματα	—
Martin M1 / M2	320×256 έγχρωμη	114 s / 58 s
Scottie S1 / S2	320×256 έγχρωμη	110 s / 71 s
Scottie DX	320×256 έγχρωμη	269 s
Robot 36 / 72	320×240 έγχρωμη	36 s / 72 s

Το **Auto** λειτουργεί με δύο τρόπους: διαβάζει την **κεφαλίδα VIS** — τον ψηφιακό κωδικό τρόπου που στέλνεται στα πρώτα 300 ms μιας εκπομπής — και, αν η κεφαλίδα χάθηκε (συντονιστήκατε αργά ή χάθηκε λόγω QSB), αναγνωρίζει τον τρόπο από τον χρονισμό των παλμών συγχρονισμού. Η επιλογή συγκεκριμένου τρόπου τον επιβάλλει, αλλά ο αποκωδικοποιητής επαληθεύει και πάλι τον συγχρονισμό πριν σχεδιάσει οτιδήποτε, ώστε μια λανθασμένη επιλογή να μη δίνει εικόνα αντί για θόρυβο.

## Ανάγνωση των αποτελεσμάτων

Η γραμμή κατάστασης κάτω από τα χειριστήρια αναφέρει τι κάνει ο αποκωδικοποιητής:

Κατάσταση	Σημασία
Waiting for VIS / AUTO lock	Ακούει· τίποτα δεν έχει αναγνωριστεί ακόμη
Martin M1? verifying sync...	Βρέθηκε υποψήφιος και επιβεβαιώνεται με τις επόμενες γραμμές
Martin M1 lock (VIS)	Κλείδωσε από την κεφαλίδα VIS
Martin M1 lock (AUTO)	Κλείδωσε από τον χρονισμό συγχρονισμού
Martin M1 lock (MANUAL)	Ξεκίνησε με το κουμπί ● <b>Force</b>
Candidate rejected (no sync)	Ο υποψήφιος δεν επιβεβαιώθηκε — φυσιολογικό σε θόρυβο
— sync lost, resetting	Το σήμα εξαφανίστηκε στη μέση της εικόνας
Frame complete	Ελήφθη ολόκληρη η εικόνα

Ο εντοπισμένος τρόπος εμφανίζεται επίσης με πράσινο δίπλα στον τίτλο **SSTV**, ενώ ο μετρητής γραμμών δείχνει την πρόοδο.

## Χειριστήρια

Κουμπί	Ενέργεια
► <b>Start</b>	Οπλίζει τον αποκωδικοποιητή. Δεν σχεδιάζει εικόνα από μόνος του — απαιτείται κλείδωμα από την κεφαλίδα VIS ή από τον εντοπισμό συγχρονισμού.
■ <b>Stop</b>	Σταματά την αποκωδικοποίηση.
↺ <b>Reset</b>	Καθαρίζει τον καμβά και επανοπλίζει επιτόπου. Χρησιμοποιήστε το μεταξύ εικόνων ή μετά από νέο συντονισμό.
● <b>Force</b>	Ξεκινά να σχεδιάζει <b>αμέσως</b> με τον τρόπο που έχει επιλεγεί στο μενού, παρακάμπτοντας εντελώς τον VIS και τον εντοπισμό συγχρονισμού. Διαθέσιμο μόνο όταν το Mode δεν είναι <b>Auto</b> .
<b>Save</b>	Αποθηκεύει την τρέχουσα εικόνα ως PNG.

**Πότε να χρησιμοποιήσετε το Force.** Αν βλέπετε εικόνα στον καταρράκτη αλλά ο αποκωδικοποιητής δεν κλειδώνει σε αυτήν — ασυνήθιστος τρόπος, σήμα πολύ ασθενές ή πολύ παραμορφωμένο για τους ανιχνευτές ή κεφαλίδα που χάσατε — επιλέξτε τον τρόπο χειροκίνητα και πατήστε ● **Force**. Το καρέ αγκυρώνεται τη στιγμή που κάνετε κλικ και η σήμανση τρόπου δείχνει **MANUAL**, ώστε να το ξεχωρίζετε από κλείδωμα VIS ή AUTO. Ο αποκωδικοποιητής και πάλι προσαρμόζεται σε πραγματικό παλμό συγχρονισμού αν βρει κάποιον, οπότε ένα κλικ λίγο νωρίτερα ή αργότερα διορθώνεται.

Επειδή το Force παρακάμπτει κάθε έλεγχο ασφαλείας, θα ζωγραφίσει ευχαρίστως θόρυβο αν το πατήσετε σε άδειο κανάλι, και δεν σταματά από μόνο του — πατήστε ■ **Stop** ή ↺ **Reset**.

## Σημειώσεις

- **Ο θόρυβος δεν ξεκινά τον αποκωδικοποιητή.** Οι ατμοσφαιρικές εκκενώσεις, οι σπινθήρες και ο θόρυβος του δικτύου αρκούσαν κάποτε για να ενεργοποιήσουν αποκωδικοποίηση. Κάθε στάδιο ανίχνευσης ελέγχει πλέον ότι ο τόνος είναι γνήσιος τόνος και επιβεβαιώνει τον υποψήφιο με τις επόμενες γραμμές πριν σχεδιαστεί έστω ένα εικονοστοιχείο. Περιμένετε ο αποκωδικοποιητής να παραμένει ήσυχος σε άδεια μπάντα.
- **Η εικόνα παρασύρεται διαγώνια αν είστε εκτός συχνότητας.** Το SSTV δεν συγχωρεί σφάλματα συντονισμού. Αν οι γραμμές γέρνουν, ρυθμίστε τη συχνότητα με μικρά βήματα και αφήστε να ξεκινήσει η επόμενη εικόνα.
- Η ισορροπία χρωμάτων και η ευκρίνεια είναι σταθερές· δεν υπάρχουν ρυθμίσεις να γίνουν.

## 12. Γενικές συμβουλές

**Το Decoder: ON πρέπει να ενεργοποιηθεί πρώτα.** Το αναπτυσσόμενο μενού είναι απενεργοποιημένο (γκριζαρισμένο) μέχρι να κάνετε κλικ στο κουμπί Decoder.

**Ένας αποκωδικοποιητής τη φορά.** Η επιλογή νέου αποκωδικοποιητή από το μενού σταματά αυτόματα τον προηγούμενος ενεργό και επαναφέρει τυχόν αλλαγές τρόπου ή εύρους διέλευσης που είχε κάνει.

**Ο τρόπος λειτουργίας ρυθμίζεται αυτόματα.** Το HF FAX και το NAVTEX θέτουν τον δέκτη σε USB μόλις τα επιλέξετε, το SSTV επιλέγει την πλευρική ζώνη ανάλογα με τη μπάντα (LSB κάτω από 10 MHz, USB πάνω), ενώ τα FAX, NAVTEX και FSK ρυθμίζουν επίσης το εύρος διέλευσης όταν πατήσετε το κουμπί Tune τους. Τα FT8, FT4, FT2, JS8 και WSPR κάνουν το ίδιο μόλις τα επιλέξετε — από τη σειρά κουμπιών ή από το αναπτυσσόμενο μενού: ο δέκτης περνά σε USB και παίρνει το δικό του εύρος διέλευσης — ολόκληρη η υποζώνη των 3 kHz για την οικογένεια FT8, 1350-1650 Hz για το WSPR. Όταν σταματήσετε τον αποκωδικοποιητή, αποκαθίσταται ο προεπιλεγμένος τρόπος της μπάντας.

**Η ενεργοποίηση ενός αποκωδικοποιητή δεν διακόπτει πλέον τον ήχο.** Οι αποκωδικοποιητές εκτελούνται σε δικά τους νήματα, οπότε δεν υπάρχει κενό, κλικ ή διακοπή όταν κάποιος ξεκινά, σταματά ή αντικαθίσταται από άλλον.

**Η ακρίβεια του ρολογιού του συστήματος έχει σημασία.** Τα FT8, FT4 και WSPR είναι κρίσιμα ως προς τον χρόνο. Αποκωδικοποιούν σε σταθερά παράθυρα ευθυγραμμισμένα με το UTC. Αν το ρολόι του υπολογιστή σας αποκλίνει περισσότερο από 1-2 δευτερόλεπτα, τα ποσοστά αποκωδικοποίησης θα πέσουν σημαντικά. Χρησιμοποιήστε πελάτη NTP για να διατηρείτε το ρολόι ακριβές.

**Τα φίλτρα θορύβου δεν φτάνουν στους αποκωδικοποιητές.** Τα NR, NB, NS και AN είναι βοηθήματα ακρόασης, μόνο για τα αυτιά σας. Κάθε αποκωδικοποιητής — FT8, FT4/FT2, CW, WSPR, FAX, NAVTEX, FSK/RTTY/PSK31/Olivia, SSTV και ο QRSS grabber — παίρνει τον ήχο πριν από αυτά τα φίλτρα, οπότε ρυθμίστε τα όπως ακούγεται καλύτερα, χωρίς να ανησυχείτε για την ποιότητα αποκωδικοποίησης. Για τον ίδιο λόγο οι αποκωδικοποιητές συνεχίζουν να δουλεύουν όσο ο δέκτης είναι σε σίγαση ή squelch: μπορείτε να κλείσετε το ηχείο και να αφήσετε έναν αποκωδικοποιητή ή μια ολονύκτια λήψη QRSS να συνεχίσει. Το μόνο που ακολουθεί όντως αυτό που ακούτε είναι το Φασματογράφημα Ήχου, που σκόπιμα δείχνει τον φιλτραρισμένο ήχο.

**Η ποιότητα του σήματος υπερσχύει της ισχύος του.** Οι περισσότεροι από αυτούς τους αποκωδικοποιητές σχεδιάστηκαν για ασθενή σήματα. Μια πιο ήσυχη μπάντα με λιγότερο θόρυβο είναι συχνά πιο παραγωγική από ένα δυνατό σήμα γεμάτο παρεμβολές. Χρησιμοποιήστε τον καταρράκτη και τα χειριστήρια εύρους διέλευσης για να εντοπίσετε και να αποφύγετε το QRM πριν ενεργοποιήσετε αποκωδικοποιητή.

**Χρησιμοποιείτε άφοβα τα κουμπιά Refresh ή Clear.** Οι εικόνες FAX παρασύρονται αν η συχνότητα ένδειξης είναι ελαφρώς εκτός, ενώ οι αποκωδικοποιητές κειμένου συσσωρεύουν χαρακτήρες θορύβου. Ένα καθαρό ξεκίνημα μετά από ρυθμίσεις συντονισμού δίνει συχνά πολύ καθαρότερο αποτέλεσμα.

## Αυτόματη αναφορά spot και γραφήματα του διακομιστή

Οι αποκωδικοποιήσεις FT8, FT4 και WSPR μπορούν να ανεβαίνουν αυτόματα και από τον ίδιο τον διακομιστή — FT8/FT4 στο PSK Reporter, WSPR στο WSPRnet — μέσω ενός δαίμονα autorun που ξεκινά ο διαχειριστής από τον πίνακα διαχείρισης. Λειτουργεί ανεξάρτητα από τους αποκωδικοποιητές του φυλλομετρητή σας: συνεχίζει είτε ακούει κάποιος είτε όχι, και τίποτα από όσα αποκωδικοποιείτε στον φυλλομετρητή δεν αναφέρεται.

Ο διαχειριστής τον παρακολουθεί με δύο μετρητές, που εύκολα μπερδεύονται μεταξύ τους: τα πλακίδια ανά αποκωδικοποιητή μετρούν τα spot που ανέβηκαν από την τελευταία εκκίνηση του δαίμονα, ενώ ο αριθμός δίπλα σε κάθε πλαίσιο ελέγχου μπάντας/τρόπου είναι το συνολικό άθροισμα της θέσης από την αρχή και επιβιώνει των επανεκκινήσεων. Ο ίδιος πίνακας διαθέτει σελίδα **Γραφήματα** που σχεδιάζει συχνότητα CPU, φόρτο CPU, θερμοκρασία CPU και συνδεδεμένους χρήστες για τα τελευταία 15 λεπτά έως 24 ώρες. Και τα δύο περιγράφονται στον [οδηγό του πίνακα διαχείρισης](#).

## ΚΕΦΑΛΑΙΟ 9

## RADE README

RADE\_README.md

## Ψηφιακή φωνή RADE v1 για το PhantomSDR-Plus

Το **RADE** (Radio AutoencoDEr) είναι ο κορυφαίος τρόπος ψηφιακής φωνής HF του FreeDV. Χρησιμοποιεί έναν υβριδικό συνδυασμό μηχανικής μάθησης / DSP (τον νευρωνικό κωδικοποιητή φωνής FARGAN) για να προσφέρει ομιλία υψηλής ποιότητας στα HF με SNR έως και  $-2$  dB, σε εύρος ζώνης RF μόλις 1500 Hz — στενότερο από ένα σήμα SSB.

Το έγγραφο αυτό καλύπτει την πλήρη ενσωμάτωση της λήψης RADE v1 στο PhantomSDR-Plus, που υλοποιείται ως sidecar σε Python (`rade_helper.py`) και γεφυρώνει τον περιηγητή με την αλυσίδα αποκωδικοποίησης `radac_rxe.py` + `lpcnet_demo`.

## Η εγκατάσταση — ο σύντομος δρόμος

```
cd ~/PhantomSDR-Plus
./install_rade.sh
```

Αυτή είναι όλη η εγκατάσταση. Το script εγκαθιστά τα πακέτα συστήματος μέσω όποιου διαχειριστή πακέτων έχει το μηχάνημα (`apt`, `pacman`, `dnf` ή `zypper`), καταφεύγει στο `pip` για ό,τι δεν έχει η διανομή, αναβαθμίζει το `websockets` αν η έκδοση της διανομής είναι παλαιότερη από την 11.0 που χρειάζεται το sidecar, κλωνοποιεί και χτίζει το `radac`, ελέγχει τα βάρη του μοντέλου, ξαναχτίζει το frontend και ξεκινά το sidecar. Στο Ubuntu 22.04 παραδίδει τη σκυτάλη στο `install_rade_ubuntu22.sh`. Το `./install.sh` προσφέρεται να το τρέξει στο πλαίσιο της κανονικής εγκατάστασης, οπότε σε ένα φρέσκο μηχάνημα το RADE υπάρχει ήδη.

**Αυτό το έγγραφο εξηγεί πώς λειτουργεί το RADE και πώς να το τρέχετε — δεν είναι η διαδικασία εγκατάστασης.** Αν χρειάζεται να κάνετε την εγκατάσταση με το χέρι, σε σύστημα που δεν καλύπτει το script ή για να επισκευάσετε ένα βήμα, ακολουθήστε τον οδηγό που συνδέεται παρακάτω· εκεί και μόνο εκεί είναι γραμμένα τα βήματα. Όλα όσα ακολουθούν εδώ — η αρχιτεκτονική, τα διορθωμένα αρχεία, η θύρα, η χρήση του RADE στον περιηγητή, οι μεταβλητές περιβάλλοντος, η αντιμετώπιση προβλημάτων και η μέτρηση ταυτοχρονίας — ισχύουν για κάθε εγκατάσταση, όπως κι αν γίνει.

- **Χειροκίνητη εγκατάσταση σε Linux** - ο δρόμος με το χέρι, βήμα βήμα, για Ubuntu, Debian, Fedora, Arch και Raspberry Pi OS

## Πίνακας περιεχομένων

1. Η εγκατάσταση — ο σύντομος δρόμος
2. Πώς λειτουργεί
3. Επισκόπηση αρχιτεκτονικής
4. RADEL έναντι RADEU
5. Προϋποθέσεις
6. Εγκατάσταση με το χέρι

7. Τοποθέτηση των διορθωμένων αρχείων
8. Έλεγχος του sidecar
9. Η θύρα 8074
10. Χρήση του RADE στον περιηγητή
11. Επαλήθευση και αποσφαλμάτωση
12. Μεταβλητές περιβάλλοντος
13. Αρχεία που άλλαξαν
14. Σύνοψη ροής σήματος
15. Αντιμετώπιση προβλημάτων
16. Μέτρηση ταυτοχρονίας στο δικό σας υλικό
17. Ενημέρωση του RADE v1

## Πώς λειτουργεί

Το RADE v1 δεν μπορεί να εκτελεστεί στον περιηγητή — απαιτεί PyTorch και τον νευρωνικό κωδικοποιητή φωνής FARGAN, που είναι υπερβολικά μεγάλα για WASM. Η λύση είναι μια διεργασία sidecar σε Python (`rade_helper.py`) που εκτελείται στον ίδιο διακομιστή με το PhantomSDR-Plus.

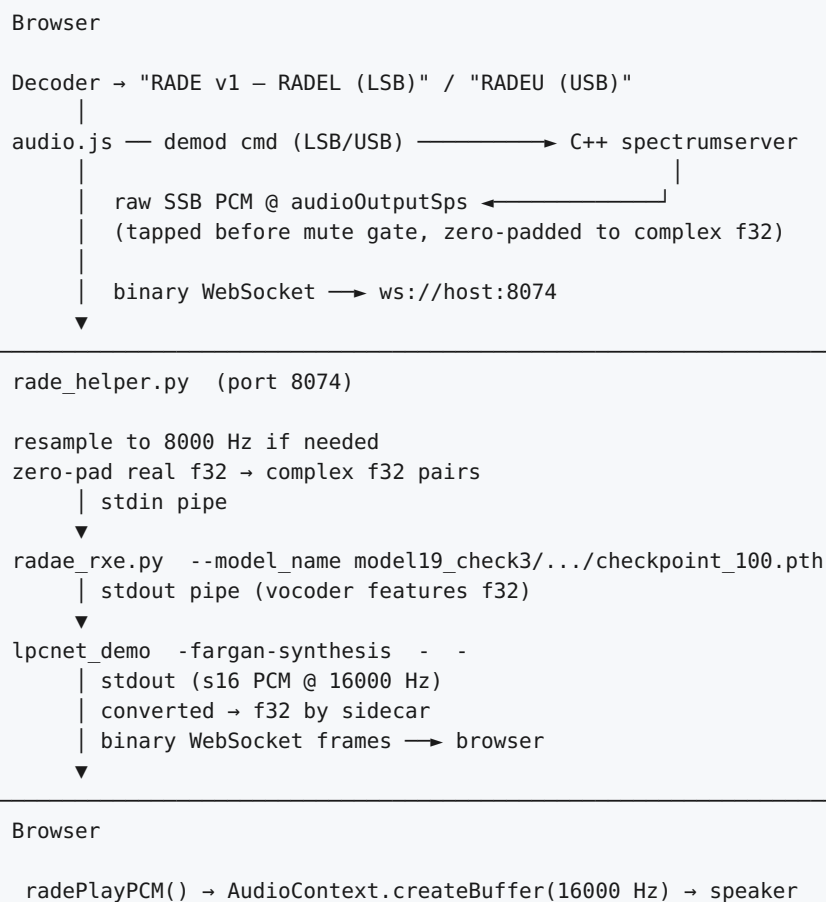
**Σημαντικό:** το `freedv_rx` του αποθετηρίου codec2 **δεν** υποστηρίζει RADEV1. Η αλυσίδα αποκωδικοποίησης RADE βρίσκεται εξ ολοκλήρου στο ξεχωριστό αποθετήριο `radae` του David Rowe (VK5DGR). Μην επιχειρήσετε να χρησιμοποιήσετε το `freedv_rx` του codec2 για RADE.

Όταν επιλέγετε RADE στον περιηγητή:

1. Το frontend θέτει την υποκείμενη αποδιαμόρφωση σε USB ή LSB — ο διακομιστής C++ αποδιαμορφώνει κανονικά το σήμα SSB.
2. Το ακατέργαστο αποδιαμορφωμένο PCM λαμβάνεται στο `audio.js` **πριν** από οποιαδήποτε σίγαση ή πύλη squelch — το `radae_rxe.py` χρειάζεται συνεχή είσοδο για να διατηρεί τον συγχρονισμό πλαισίων.
3. Κάθε τμήμα PCM συμπληρώνεται με μηδενικά από πραγματικό f32 σε μιγαδικό f32 (πραγματικό + 0.0 φανταστικό) — αυτό αναμένει το `radae_rxe.py` στην τυπική είσοδο.
4. Ο sidecar διοχετεύει αυτά τα δείγματα στο `radae_rxe.py`, το οποίο παράγει χαρακτηριστικά κωδικοποιητή φωνής.
5. Το `lpcnet_demo -fargan-synthesis` μετατρέπει τα χαρακτηριστικά σε ομιλία s16 στα 16000 Hz.
6. Ο sidecar μετατρέπει s16 → f32 και τα στέλνει πίσω στον περιηγητή ως δυαδικά πλαίσια WebSocket.
7. Ο περιηγητής αναπαράγει την αποκωδικοποιημένη ομιλία μέσω του Web Audio API στα 16000 Hz.

Ο διακομιστής C++ (`spectrumserver.cpp`) δεν τροποποιείται καθόλου.

## Επισκόπηση αρχιτεκτονικής



## RADEL έναντι RADEU

Το RADE v1 μεταδίδεται πάντα μέσω SSB. Κατά σύμβαση:

Τρόπος	Πλευρική ζώνη	Χρήση στις μπάντες
RADEL	LSB	160 m, 80 m, 40 m ( $\leq 10$ MHz)
RADEU	USB	20 m, 17 m, 15 m, 12 m, 10 m ( $> 10$ MHz)

## Προϋποθέσεις

Απαίτηση	Έκδοση	Σημειώσεις
PhantomSDR-Plus	οποιαδήποτε	με frontend Vite/Svelte
Linux	Ubuntu 24.04+ / Debian Bookworm+	δοκιμασμένο
Python	3.8+	για τα <code>rade_helper.py</code> και <code>radar_rxe.py</code>
αποθετήριο radar	τελευταίο master	παρέχει τα <code>radar_rxe.py</code> και <code>lpcnet_demo</code>
cmake	3.10+	για μεταγλώττιση του <code>lpcnet_demo</code> από το radar
PyTorch	2.0+	απαιτείται από το <code>radar_rxe.py</code>
Node.js	16+	για το <code>npm run build</code>
websockets (Python)	10-16+	<code>pip3 install websockets</code> — υποστηρίζονται όλες οι εκδόσεις
matplotlib	οποιαδήποτε	απαιτείται από το <code>radar_rxe.py</code> κατά την εισαγωγή
numpy	1.23+	απαιτείται· επιτρέπει επίσης την επαναδειγματοληψία
scipy	οποιαδήποτε	προαιρετικό, επιτρέπει ακριβή επαναδειγματοληψία

## Εγκατάσταση με το χέρι

Το `./install_rade.sh` κάνει ολόκληρη την εγκατάσταση. Όταν θέλετε να την κάνετε μόνοι σας — σε σύστημα που δεν καλύπτει το script, ή για να επισκευάσετε ένα βήμα — η διαδικασία βρίσκεται στον **Οδηγό χειροκίνητης εγκατάστασης RADE v1**, και μόνο εκεί. Καλύπτει Ubuntu, Debian, Fedora, Arch και Raspberry Pi OS σε ένα πέρασμα:

Βήμα	Περιγραφή
Βήμα 1	Πακέτα συστήματος και ο έλεγχος για Node.js 22+
Βήμα 2	Πακέτα Python, μαζί με τη σημαία του PEP 668 και το torch wheel μόνο για CPU
Βήμα 3	Κλωνοποίηση και μεταγλώττιση του αποθετηρίου radar, επαλήθευση του <code>lpcnet_demo</code> και των βαρών του μοντέλου
Βήμα 4	Επαλήθευση της αλυσίδας αποκωδικοποίησης εκτός σύνδεσης, πριν συνδεθεί με τον περιηγητή
Βήμα 5	Μεταγλώττιση του frontend του PhantomSDR-Plus
Βήμα 6	Έλεγχος του sidcar
Βήμα 7	Άνοιγμα της θύρας 8074
Βήματα 8-10	Εκκίνηση του διακομιστή, επαλήθευση του RADE, χρήση του στον περιηγητή

Το υπόλοιπο αυτού του εγγράφου θεωρεί ότι αυτό έχει ολοκληρωθεί.

## Τοποθέτηση των διορθωμένων αρχείων

Αντιγράψτε τα ακόλουθα αρχεία από το σύνολο διορθώσεων στο αποθετήριό σας PhantomSDR-Plus.

## Νέο αρχείο — τοποθετήστε το στη ρίζα του αποθετηρίου, δίπλα στο σενάριο εκκίνησης που χρησιμοποιείτε:

```
rade_helper.py
```

## Διορθωμένα αρχεία frontend — τοποθετήστε τα στο frontend/src/:

```
audio.js
App.svelte
```

## Τι κάνουν οι διορθώσεις

**audio.js** — 5 αλλαγές:

- Κατασκευαστής: 5 νέα πεδία κατάστασης RADE (`decodeRADE`, `_radeSideband`, `_radeSocket`, `_radeCallback`, `_radeReady`, `_radeNextTime`)
- Αποθήκευση PCM πριν την ενίσχυση: το `pcmArrayPreBoost` αποθηκεύεται πριν από την ενίσχυση 300× του FLAC, ώστε το RADE να λαμβάνει το αρχικό πλάτος (η ενίσχυση θα κορέσει το `radar_xe.py`)
- `playAudio()`: λήψη PCM για το RADE με τον ήχο πριν την ενίσχυση, πριν από την πύλη σίγασης/`squelch`
- `playAudio()`: φύλακας `if (this.decodeRADE) return` — καταστέλλει την αναπαραγωγή του ακατέργαστου SSB όσο παίζει η αποκωδικοποιημένη από το RADE ομιλία
- Νέες μέθοδοι: `setRADEDecoding()`, `setRADECallback()`, `_radePlayPCM()` με προγραμματισμένη αναπαραγωγή χωρίς κενά μέσω του ρολογιού `_radeNextTime` (αναπαραγωγή στα **16000 Hz** — ο ρυθμός εξόδου του `lpcnet_demo`)

**Κάθε παραλλαγή Svelte** — 6 αλλαγές:

- `demodulationDefaults`: `RADEL {type: 'LSB', offsets: [2200, -700]}`, `RADEU {type: 'USB', offsets: [-700, 2200]}` — το εύρος διέλευσης ξεκινά 700 Hz από το φέρον, με πλάτος 1500 Hz
- Μεταβλητές κατάστασης: `radeEnabled`, `radeConnected`, `radeSynced`, `radeSnr`, `_radeDeactivate()`
- `_radeDeactivate()`: σταματά τον αποκωδικοποιητή και **επαναφέρει τον προεπιλεγμένο τρόπο της μπάντας** από το `bands-config.js` — συμπεριφορά πανομοιότυπη με την απενεργοποίηση των FAX, NAVTEX, FSK
- `_deactivateAll()`: γραμμή καθαρισμού RADE
- `activateSelectedDecoder()`: κλάδοι `radel` και `radeu`
- Αναπτυσσόμενο μενού αποκωδικοποιητών: δύο νέες καταχωρίσεις `<option>`
- Πίνακας κατάστασης: κουκκίδα σύνδεσης, κατάσταση συγχρονισμού/SNR, πλαίσιο σφάλματος

## Σενάρια εκκίνησης

Τα σενάρια εκκίνησης του δέκτη (`start-rx888mk2.sh`, `start-airspyhf.sh`, `start-rtl.sh`, `start-rspla.sh`, `start-fobos-hf.sh`, `start-fobos.sh`, `start-hackrf.sh`) ξεκινούν μόνα τους το `sidecar` μόλις ο διακομιστής λειτουργήσει, και το `stop-websdr.sh` το σταματά μαζί με τον διακομιστή. Το `rade.sh` παραμένει για αυτόνομη εκτέλεση του `sidecar` — δείτε [Έλεγχος του sidecar](#).

## Έλεγχος του sidecar

**Κανονικά δεν χρειάζεστε ξεχωριστό σενάριο ελέγχου για το RADE.** Τα σενάρια εκκίνησης του δέκτη (`start-rx888mk2.sh`, `start-airspyhf.sh`, `start-rtl.sh`, `start-rspla.sh`, `start-fobos-hf.sh`, `start-fobos.sh`, `start-hackrf.sh`) ξεκινούν μόνα τους το `sidecar` μόλις επιβεβαιωθεί ότι ο διακομιστής λειτουργεί, το επανεκκινούν αν τερματιστεί, και αναφέρουν την κατάστασή του στη σύνοψη που

τυπώνουν στο τέλος — ✓ `RADE sidecar running`, ή μια σημείωση · που εξηγεί γιατί δεν ενεργοποιήθηκε (δεν είναι εγκατεστημένο, `RADE_ENABLED=0`, ή δεν υπάρχει `python3`). Το `./stop-websdr.sh` το σταματά μαζί με τον διακομιστή. Για να τρέξετε τον διακομιστή χωρίς RADE, ξεκινήστε τον με `RADE_ENABLED=0`.

Το `rade.sh` παραμένει για αυτόνομη χρήση — για να τρέξετε το sidecar χωρίς τον launcher του PhantomSDR-Plus, ή για να το δοκιμάσετε μόνο του:

```
chmod +x ~/PhantomSDR-Plus/rade.sh
cd ~/PhantomSDR-Plus
./rade.sh start # start sidecar with self-restarting watchdog
./rade.sh stop # stop sidecar + watchdog + all lpcnet_demo children
./rade.sh restart # stop then start cleanly
./rade.sh status # show running / stopped + process info
```

Η δραστηριότητα του sidecar καταγράφεται στο `~/PhantomSDR-Plus/rade.log`:

```
tail -f ~/PhantomSDR-Plus/rade.log
```

**Μην ανακατεύετε τα δύο όσο ο διακομιστής τρέχει.** Αν ο watchdog ενός σεναρίου εκκίνησης είναι ενεργός, θα επαναφέρει το sidecar μέσα σε περίπου πέντε δευτερόλεπτα, οπότε το `./rade.sh stop` θα φαίνεται ότι απέτυχε. Για να σταματήσετε το RADE ενώ ο διακομιστής τρέχει, είτε ξεκινήστε τον διακομιστή με `RADE_ENABLED=0`, είτε σταματήστε τα πάντα με `./stop-websdr.sh`.

Αν όντως τρέξετε το `rade.sh` αυτόνομα, **χρησιμοποιείτε πάντα `./rade.sh stop`** — ένα σκέτο `pkill -f rade_helper.py` δεν νικά τον δικό του watchdog, ο οποίος επανεκκινεί τη διεργασία μέσα σε 3 δευτερόλεπτα.

## Η θύρα 8074

Το sidecar ακούει στη θύρα TCP **8074** και ο περιηγητής συνδέεται απευθείας σε αυτήν, οπότε η θύρα πρέπει να είναι προσβάσιμη από έξω. Το άνοιγμά της — `ufw`, `firewalld`, `iptables`, ο κανόνας NAT του δρομολογητή, και το proxy Nginx για την περίπτωση που ο πάροχος μπλοκάρει εντελώς τη θύρα — είναι το **Βήμα 7 του οδηγού χειροκίνητης εγκατάστασης**.

Δύο πράγματα αξίζει να επαναληφθούν εδώ, γιατί κοστίζουν τον περισσότερο χρόνο:

```
Is the sidecar listening?
ss -tlnp | grep 8074
```

**Προσοχή στο NAT hairpin:** μια δοκιμή με `curl` από τον ίδιο τον διακομιστή προς το δημόσιο όνομά του συχνά επιστρέφει `Connection refused` ακόμη κι όταν η θύρα είναι ανοιχτή — πολλοί δρομολογητές δεν επιστρέφουν την κίνηση. Δοκιμάζετε πάντα από εξωτερικό μηχάνημα, ή χρησιμοποιήστε το **<https://portchecker.co>**.

## Χρήση του RADE στον περιηγητή

1. Ανοίξτε τη διεπαφή web του PhantomSDR-Plus
2. Βρείτε ενεργούς σταθμούς στο **[qso.freedv.org](https://qso.freedv.org)**
3. Συντονιστείτε στη συχνότητα ένδειξης του σταθμού
4. Στις **Decoder Options** επιλέξτε:

5. **RADE v1 — RADEL (LSB)** για 40 m / 80 m / 160 m
6. **RADE v1 — RADEU (USB)** για 20 m / 17 m / 15 m / 12 m / 10 m
7. Κάντε κλικ στο **Decoder: ON**

## Τα κουμπιά RADEL / RADEU (με ένα πάτημα)

Τα βήματα 4 και 5 μπορούν να αντικατασταθούν από ένα μόνο πάτημα. Ένα ζευγάρι κουμπιών **RADEL / RADEU** υπάρχει σε τρία σημεία:

- δίπλα στην επικεφαλίδα **Modes selector** του κύριου πίνακα·
- μέσα στο αναδυόμενο παράθυρο **Modes**·
- μέσα στο αναδυόμενο παράθυρο **Bands**.

Το πάτημα ενός από αυτά επιλέγει τον αποκωδικοποιητή, ανάβει τον διακόπτη Decoder, κλείνει το αναδυόμενο παράθυρο μέσα από το οποίο το πατήσατε και φέρνει το πάνελ του RADE στην οθόνη. Το κουμπί γίνεται μπλε όσο τρέχει το RADE. **Πατήστε το ίδιο κουμπί ξανά για να σταματήσει το RADE** — το πάνελ κλείνει και ο τρόπος λειτουργίας με το passband επανέρχονται όπως περιγράφεται παρακάτω. Τα κουμπιά, το αναπτυσσόμενο μενού **Decoder Options** και το κουμπί ON/OFF μοιράζονται την ίδια κατάσταση.

## Καταστάσεις της ένδειξης του πίνακα

Ένδειξη	Σημασία
Κόκκινο — «Connecting to sidecar...»	Η θύρα 8074 δεν είναι προσβάσιμη ή ο sidecar δεν εκτελείται
Κίτρινο — «Searching for signal...»	Ο sidecar συνδέθηκε, δεν εντοπίστηκε ακόμη πλαίσιο RADE
Πράσινο — «Synced · SNR x.x dB»	Αποκωδικοποίηση — ακούγεται ομιλία

## Όταν απενεργοποιείτε τον αποκωδικοποιητή

Ο τρόπος και το εύρος διέλευσης επανέρχονται αυτόματα στη σωστή προεπιλογή για την τρέχουσα συχνότητα, όπως ορίζεται στο `bands-config.js` — όπως ακριβώς με τα FAX, NAVTEX, FSK.

## Επαλήθευση και αποσφαλμάτωση

### Εκτελείται ο sidecar;

```
ps aux | grep rade_helper | grep -v grep
ss -tlnp | grep 8074
```

### Παρακολούθηση ζωντανής δραστηριότητας

```
tail -f ~/PhantomSDR-Plus/rade.log
```

Όταν συνδέεται περιηγητής:

```
[RADE] client connected: x.x.x.x:XXXXX
[RADE] x.x.x.x:XXXXX sps=12000 sideband=LSB
[RADE] x.x.x.x:XXXXX spawning pipeline (torch_threads=1)
```

## Γρήγορη δοκιμή WebSocket

```
python3 - << 'EOF'
import asyncio, websockets, json

async def test():
 async with websockets.connect('ws://localhost:8074') as ws:
 await ws.send(json.dumps({'type': 'init', 'sps': 8000, 'sideband': 'LSB'}))
 print(await ws.recv()) # expect: {"type": "status", "connected": true}

asyncio.run(test())
EOF
```

## Κονσόλα περιηγητή (F12)

[RADE] ► ENABLED LSB @ 12000 Hz → helper ws://localhost:8074

Καλό — το WebSocket άνοιξε. Η τιμή σε Hz ταιριάζει με το `audioOutputSps` του διακομιστή σας (συνήθως 8000–12000 Hz).

[RADE] sidecar socket error

Η θύρα 8074 δεν είναι προσβάσιμη — ελέγξτε το τείχος προστασίας και τον κανόνα NAT του δρομολογητή.

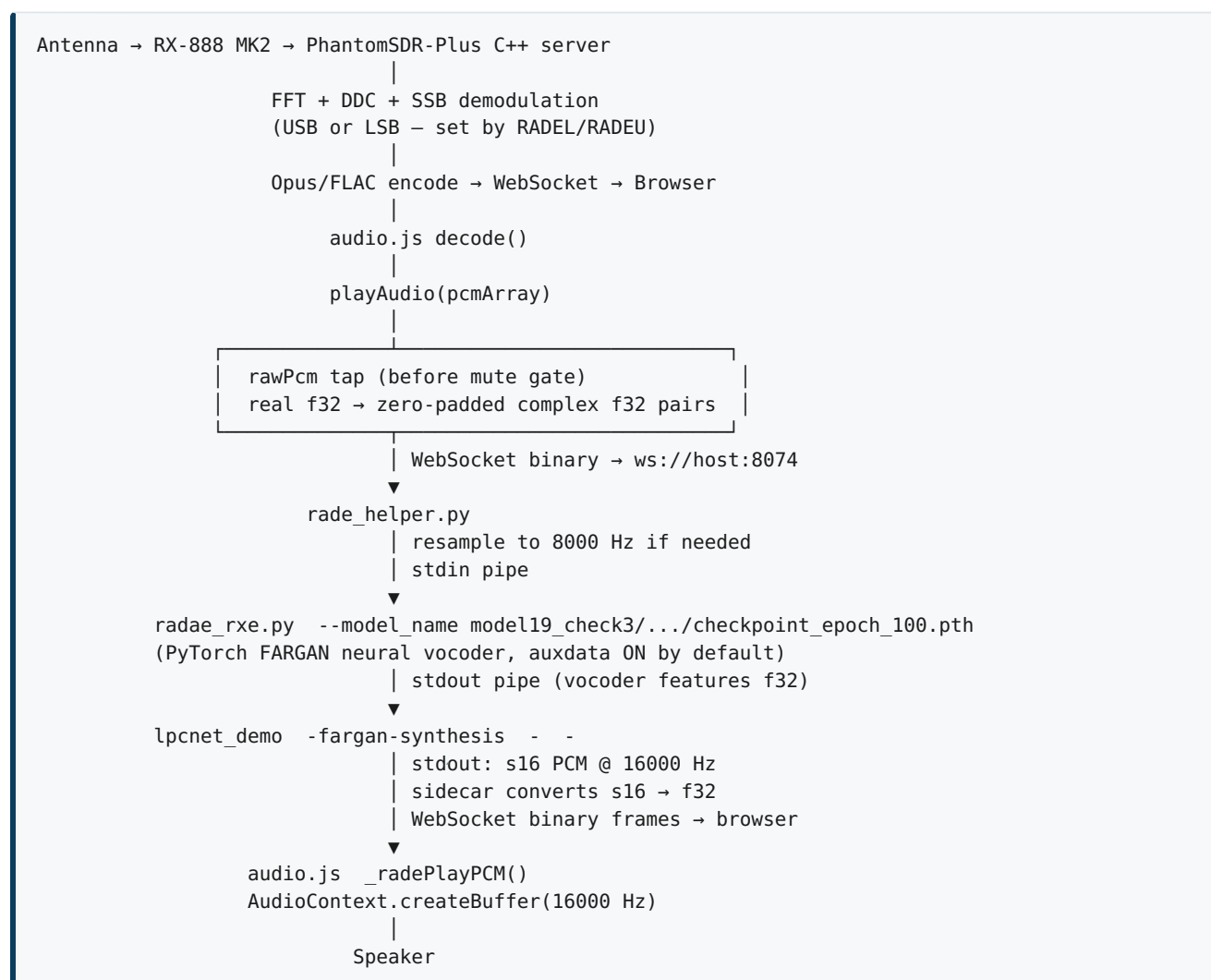
## Μεταβλητές περιβάλλοντος

Μεταβλητή	Προεπιλογή	Περιγραφή
<code>RADE_HELPER_PORT</code>	8074	Θύρα TCP στην οποία ακούει ο sidecar
<code>RADE_HELPER_HOST</code>	0.0.0.0	Διεύθυνση δέσμευσης (127.0.0.1 πίσω από διαμεσολαβητή)
<code>RADAE_DIR</code>	~/radae	Ρίζα του αποθετηρίου radae
<code>RADE_MODEL</code>	<code>RADAE_DIR/model19_check3/checkpoints/checkpoint_epoch_100.pth</code>	Βάρη μοντέλου
<code>LPCNET_DEMO</code>	<code>RADAE_DIR/build/src/lpcnet_demo</code>	Εκτελέσιμο <code>lpcnet_demo</code>
<code>RADE_AUXDATA</code>	1	Θέστε το σε 0 για να περάσει --noauxdata στο <code>radae_rxe.py</code>
<code>RADE_TORCH_THREADS</code>	1	Νήματα PyTorch/OpenBLAS ανά στιγμίοτυπο <code>radae_rxe.py</code> — περιορίζει τη χρήση CPU
<code>RADE_PIN_CORES</code>	1	Καρφίτσώνει κάθε αλυσίδα αποκωδικοποίησης σε δικό της σύνολο πυρήνων· το 0 απενεργοποιεί το καρφίτσωμα
<code>RADE_CORES_PER_CLIENT</code>	2	Πυρήνες που αποδίδονται ανά πελάτη όταν το καρφίτσωμα είναι ενεργό

## Αρχεία που άλλαξαν

Αρχείο	Τύπος	Σημειώσεις
rade_helper.py	<b>Νέο</b>	Sidecar Python: διακομιστής WebSocket + αλυσίδα αποκωδικοποίησης δύο διεργασιών
frontend/src/audio.js	Τροποποιημένο	4 διορθώσεις
frontend/src/App.svelte	Τροποποιημένο	6 διορθώσεις
rade.sh	<b>Νέο</b>	Σενάριο ελέγχου του sidecar RADE για αυτόνομη χρήση (start/stop/restart/status)
start-rx88mk2.sh	Τροποποιημένο	ξεκινά, επιτηρεί και αναφέρει το sidecar· το ίδιο και τα start-airspyhf.sh, start-rtl.sh, start-rsp1a.sh, start-fobos-hf.sh, start-fobos.sh, start-hackrf.sh
stop-websdr.sh	Τροποποιημένο	σταματά το sidecar μαζί με τον διακομιστή
spectrumserver.cpp	<b>Αμετάβλητο</b>	Δεν απαιτούνται τροποποιήσεις σε C++

## Σύνοψη ροής σήματος



## Αντιμετώπιση προβλημάτων

### Κόκκινο πλαίσιο — «Sidecar not reachable»

```
Is sidecar running?
ps aux | grep rade_helper | grep -v grep

Start it manually for testing
python3 ~/PhantomSDR-Plus/rade_helper.py &

Check port is open externally – use portchecker.co NOT curl from the server
(curl from the server uses NAT loopback and gives false "Connection refused")
```

### Η θύρα 8074 εμφανίζεται κλειστή στο portchecker.co παρά τον κανόνα iptables

Ο πάροχος ίσως φιλτράρει τη θύρα ή λείπει ο κανόνας NAT του δρομολογητή. Επιλογές:

1. Δοκιμάστε άλλη θύρα: `RADE_HELPER_PORT=8080 python3 rade_helper.py`
2. Proxy μέσω της υπάρχουσας δημόσιας θύρας σας με Nginx (δείτε το [Βήμα 7 του οδηγού χειροκίνητης εγκατάστασης](#))

### `radae_rxe.py: error: unrecognized arguments: model_path`

Η διαδρομή του μοντέλου πρέπει να δίνεται με `--model_name`, όχι ως θεσιακό όρισμα:

```
WRONG – positional argument
python3 radae_rxe.py model19_check3/checkpoints/checkpoint_epoch_100.pth

CORRECT – named argument
python3 radae_rxe.py --model_name model19_check3/checkpoints/checkpoint_epoch_100.pth
```

### `ModuleNotFoundError: No module named 'matplotlib'`

```
pip3 install matplotlib
```

Το `radae_rxe.py` εισάγει το `matplotlib` χωρίς όρους στην αρχή του αρχείου.

### `ModuleNotFoundError: No module named 'torch'`

```
pip3 install torch
```

### `lpcnet_demo: No such file or directory`

Η μεταγλώττιση του `radae` δεν ολοκληρώθηκε. Μεταγλωττίστε ξανά:

```
cd ~/radae/build
cmake .. && make -j$(nproc)
ls src/lpcnet_demo # should exist now
```

## Σφάλμα `size mismatch` ΣΤΟ `inference.sh`

Το `model19_check3` απαιτεί `--auxdata` κατά την κωδικοποίηση με το `inference.sh`:

```
./inference.sh model19_check3/checkpoints/checkpoint_epoch_100.pth \
wav/brian_g8sez.wav /dev/null \
--rate_Fs --pilots --pilot_eq --eq_ls --cp 0.004 \
--write_rx rx.f32 --auxdata # ← required for model19_check3
```

Σημείωση: κατά την **αποκωδικοποίηση** με το `radae_rxe.py`, το `--auxdata` είναι η προεπιλογή — μην το περνάτε.

## Κίτρινη ένδειξη — «Searching for signal» — δεν συγχρονίζεται ποτέ

- Επιβεβαιώστε τη σωστή πλευρική ζώνη: RADEL για  $\leq 10$  MHz, RADEU για  $> 10$  MHz
- Ελέγξτε στο [qso.freedv.org](https://qso.freedv.org) ότι κάποιος σταθμός εκπέμπει αυτή τη στιγμή
- Το RADE v1 χρησιμοποιεί 30 φέροντα σε εύρος 1500 Hz — εμφανίζεται ως συμπαγής ομάδα στον καταρράκτη
- Αφήστε έως 1,5 δευτερόλεπτο για την απόκτηση

## `underrun!!!` από το `arplay` κατά τη δοκιμή της αλυσίδας (Βήμα 3)

Αναμενόμενο μόνο σε δοκιμές εκτός σύνδεσης με αρχεία. Το `radae_rxe.py` επεξεργάζεται πιο αργά από την είσοδο/έξοδο του αρχείου, με αποτέλεσμα να αδειάζει ο ενταμιευτής ήχου. Αυτό δεν συμβαίνει σε ζωντανή λήψη, γιατί ο περιηγητής τροφοδοτεί τον ήχο σε πραγματικό χρόνο.

## `ConnectionClosedError: received 1011 (internal error)`

Ο sidcar δέχτηκε τη σύνδεση WebSocket αλλά κατέρρευσε εσωτερικά πριν απαντήσει. Αιτία είναι μια μη συμβατή έκδοση του `rade_helper.py` — μια παλαιότερη έκδοση προσπαθούσε να περάσει έναν `StreamReader` του `asyncio` ως τυπική είσοδο υποδιεργασίας, κάτι που αποτυγχάνει σιωπηλά και κλείνει με σφάλμα 1011.

**Λύση:** αντικαταστήστε το `rade_helper.py` με την τρέχουσα έκδοση από το σύνολο διορθώσεων. Η τρέχουσα έκδοση χρησιμοποιεί `os.pipe()` για τον διαδιεργασιακό αγωγό και είναι συμβατή με `websockets` 10.x έως 16.x+.

## Πολλοί ταυτόχρονοι χρήστες

Κάθε σύνδεση περιηγητή δημιουργεί το δικό της ανεξάρτητο ζεύγος `radae_rxe.py` + `lpcnet_demo`, οπότε κάθε χρήστης μπορεί να συντονίζεται ελεύθερα σε διαφορετική συχνότητα.

Εξ ορισμού το `radae_rxe.py` χρησιμοποιεί **όλους τους διαθέσιμους πυρήνες CPU** για τις πράξεις πινάκων του PyTorch, προκαλώντας χρήση CPU ~900 % ανά στιγμιότυπο. Το `rade_helper.py` το περιορίζει με δύο βελτιστοποιήσεις:

1. `OMP_NUM_THREADS=1` (και τα αντίστοιχα MKL/OpenBLAS) — περιορίζει το PyTorch σε 1 νήμα
2. Συναρτήσεις μετατροπής ήχου διανυσματοποιημένες με `numpy` — εξαλείφουν το κόστος των βρόχων της Python

Αποτέλεσμα: κάθε στιγμιότυπο καταναλώνει περίπου **8-10 %** ενός πυρήνα, αρκετά για αποκωδικοποίηση RADE σε πραγματικό χρόνο. Αν ακούτε διακοπές ήχου, αυξήστε σε 2 νήματα:

```
RADE_TORCH_THREADS=2 ./start-rx888mk2.sh # or your receiver's launcher
standalone: RADE_TORCH_THREADS=2 ./rade.sh restart
```

Ταυτόχρονοι χρήστες RADE	Κατά προσέγγιση CPU
1	~9 %
5	~45 %
10	~90 %
20	~180 %

Σε έναν i5-12450H (12 νήματα) με τον spectrumserver στο ~42 % και το rx888\_stream στο ~11 %, έχετε περίπου **1000 % περιθώριο** — αρκετό για **20+ ταυτόχρονους** χρήστες RADE προτού η CPU γίνει πρόβλημα.

**Ο παραπάνω πίνακας υποθέτει ότι το κόστος CPU κλιμακώνεται γραμμικά με τους χρήστες. Η μέτρηση λέει ότι δεν συμβαίνει αυτό.** Μια εκτέλεση του `rade_loadtest.py` στον ίδιο i5-12450H έδειξε κόστος CPU ανά ακροατή αρκετά κάτω από αυτόν τον πίνακα έως ~24 ακροατές και έπειτα απότομη άνοδο, με το μηχάνημα να σταματά από τη **θερμοκρασία του πακέτου στους 32 ακροατές** — όχι από τη CPU ούτε από το εύρος ζώνης. Θεωρήστε αυτά τα νούμερα πρόχειρο οδηγό μόνο για μικρούς αριθμούς και μετρήστε το δικό σας μηχάνημα: δείτε [Μέτρηση ταυτοχρονίας στο δικό σας υλικό](#). Σημειώστε ότι η δοκιμή φόρτου τροφοδοτεί την αλυσίδα με θόρυβο και όχι με πραγματικό σήμα RADE, οπότε η απόλυτη χρήση CPU με πραγματική κίνηση μπορεί να διαφέρει· το αξιόπιστο στοιχείο είναι το *σχήμα* της καμπύλης.

## Μέτρηση ταυτοχρονίας στο δικό σας υλικό

Τα παραπάνω νούμερα προέρχονται από ένα συγκεκριμένο μηχάνημα. Το `rade_loadtest.py` (στη ρίζα του αποθετηρίου) μετρά το (ίδιο πράγμα στο **δικό σας** — αυξάνει σταδιακά συνθετικούς ακροατές RADE απέναντι στον sidcar και αναφέρει τον τελευταίο αριθμό ακροατών που άντεξε το μηχάνημά σας πριν κάτι υποχωρήσει.

Μετά από κάθε βήμα αφήνει το μηχάνημα να ηρεμήσει και έπειτα δειγματοληπτεί τη θερμοκρασία πακέτου, τη συνολική CPU, τη διαθέσιμη RAM, το swap και το συνολικό RSS όλων των διεργασιών `radae_rxe.py` και `lpcnet_demo`. Σταματά στο πρώτο **γόνατο** — όποιο όριο παραβιαστεί πρώτο — και εμφανίζει τον τελευταίο σταθερό αριθμό.

## Προϋποθέσεις

```
sudo apt install python3-websockets python3-psutil
```

Εκτελέστε το **στον υπολογιστή του SDR** (διαβάζει τοπικούς αισθητήρες και μνήμη διεργασιών), με τον spectrumserver και τον sidcar RADE ήδη σε λειτουργία:

```
cd ~/PhantomSDR-Plus
python3 rade_loadtest.py
```

Το Ctrl-C κλείνει καθαρά όλες τις συνδέσεις οποιαδήποτε στιγμή.

## Πριν την πρώτη εκτέλεση

Ανοίξτε το αρχείο και ελέγξτε τα δύο τμήματα στην αρχή:

- **CONFIG** — το `WS_URL` έχει προεπιλογή `ws://127.0.0.1:8074`· αλλάξτε το αν μετακινήσατε τον `sidecar` από την προεπιλεγμένη θύρα. Τα `RADE_SPS` (12000) και `RADE_SIDEHAND` (USB) πρέπει να ταιριάζουν με ό,τι αναφέρει το frontend σας.
- **handshake\_messages()** — το μοναδικό πλαίσιο `init` που στέλνει πρέπει να ταιριάζει με την αρχικοποίηση RADE στο `frontend/src/audio.js`. Αν αυτή η χειραψία έχει αποκλίνει, κάθε ακροατής αποτυγχάνει να συνδεθεί και η δοκιμή τερματίζεται αμέσως με αποτυχίες σύνδεσης.

## Κατώφλια γονάτου

Ρυθμίστε τα στο CONFIG κατά βούληση — είναι σκόπιμα συντηρητικά:

Κατώφλι	Προεπιλογή	Σταματά όταν
TEMP_KNEE_C	85.0	Η θερμοκρασία πακέτου φτάσει αυτό το όριο
MIN_AVAIL_MB	800	Η διαθέσιμη RAM πέσει κάτω από αυτό
SWAP_GROWTH_MB	50	Το swap ξεπεράσει τη βάση που καταγράφηκε στην εκκίνηση
FAIL_LIMIT	3	Τόσοι ακροατές αποτύχουν να συνδεθούν σε ένα βήμα
MAX_LISTENERS	60	Απόλυτο σταμάτημα ακόμη κι αν δεν εμφανιστεί ποτέ γόνατο

Το σχήμα της κλιμάκωσης ελέγχεται από τα `STEP` (ακροατές που προστίθενται ανά βήμα, 2 εξ ορισμού), `SETTLE_S` (25 s ηρεμίας πριν τη δειγματοληψία — αυτό επιτρέπει να συσσωρευτεί θερμότητα) και `SAMPLE_S` (παράθυρο μέσου όρου CPU 5 s). Μια πλήρης εκτέλεση έως το προεπιλεγμένο όριο των 60 ακροατών διαρκεί επομένως περίπου 15 λεπτά.

## Ανάγνωση των αποτελεσμάτων

Κάθε βήμα τυπώνει μια γραμμή και προσθέτει μια εγγραφή στο `rade_loadtest.csv`:

Στήλη	Σημασία
listeners	Συνθετικοί ακροατές συνδεδεμένοι σε αυτό το βήμα
connect_fails	Ακροατές που δεν κατάφεραν να δημιουργήσουν ή να διατηρήσουν σύνδεση
pkg_c	Θερμοκρασία πακέτου, διάμεσος 5 δειγμάτων (ανθεκτική σε αιχμές)
cpu_pct	Ποσοστό CPU <b>ολόκληρου του συστήματος</b> , όχι ανά ακροατή
avail_mb	Διαθέσιμη RAM
swap_mb	Swap σε χρήση
dec_rss_mb	Συνολικό RSS όλων των διεργασιών αποκωδικοποίησης
dec_procs	Πλήθος διεργασιών αποκωδικοποίησης — αναμένονται <b>2 ανά ακροατή</b> ( <code>radac_rxe.py</code> + <code>lpcnet_demo</code> )

## Μια μετρημένη εκτέλεση

Στον i5-12450H που αναφέρθηκε παραπάνω, μια πλήρης εκτέλεση τελείωσε έτσι:

```
KNEE at n=32: temp 91.0°C ≥ 85.0
Last stable concurrency: 30 RADE listeners
```

**Η θερμότητα ήταν ο δεσμευτικός περιορισμός**, και μάλιστα με μεγάλη διαφορά. Στο γόνατο υπήρχαν ακόμη 7004 MB διαθέσιμης RAM — 6,2 GB πάνω από το όριο `MIN_AVAIL_MB` — και μηδέν αποτυχίες σύνδεσης σε κάθε βήμα.

Δύο στήλες αξίζουν προσεκτική ανάγνωση:

**Το `dec_rss_mb` υπερεκτιμά το πραγματικό κόστος μνήμης.** Αυξάνεται πολύ σταθερά κατά 295 MB ανά ακροατή (292, 294, 294 ... 295 σε όλα τα 16 βήματα), αλλά η διαθέσιμη RAM πέφτει μόνο περίπου **202 MB ανά ακροατή**. Οι δύο διεργασίες πίσω από κάθε ακροατή μοιράζονται σελίδες βιβλιοθηκών, και το RSS τις μετρά μία φορά ανά διεργασία. Προεκτείνοντας την κλίση του `avail_mb`, το γόνατο μνήμης θα ερχόταν κοντά στους **63 ακροατές** — πέρα από το απόλυτο όριο των 60, οπότε σε αυτό το μηχανήμα δεν μπορεί ποτέ να ενεργοποιηθεί. Διαστασιολογήστε τη μνήμη με βάση το `avail_mb`, όχι το `dec_rss_mb`.

**Το `cpu_pct` είναι επίπεδο — και μετά δεν είναι.** Παραμένει στο ~5,3 % ολόκληρου του μηχανήματος έως τους 24 ακροατές και έπειτα γίνεται υπεργραμμικό:

Ακροατές	24	26	28	30	32
<code>cpu_pct</code>	6,7 %	18,4 %	30,1 %	49,6 %	65,2 %

Πρόκειται για δεκαπλάσια άνοδο της CPU ενώ ο αριθμός ακροατών αυξάνεται κατά ένα τρίτο. Το σημείο καμπής επαναλαμβάνεται στο ίδιο σημείο σε διαφορετικές εκτελέσεις, οπότε αντιμετωπίστε το ως ιδιότητα του μηχανήματος και όχι ως θόρυβο — το κόστος CPU ανά ακροατή είναι περίπου 3,4 % ενός πυρήνα κάτω από το σημείο καμπής και περίπου 20 % πάνω από αυτό. Μην προεκτείνετε νούμερο CPU ανά χρήστη που μετρήθηκε σε χαμηλό αριθμό ακροατών.

**Η δοκιμή δεν έχει γόνατο CPU.** Τα τέσσερα όρια είναι θερμοκρασία, διαθέσιμη RAM, αύξηση `swar` και αποτυχίες σύνδεσης — η CPU καταγράφεται αλλά ποτέ δεν τερματίζει την εκτέλεση. Στην παραπάνω εκτέλεση η CPU έφτασε το 65 % και θα συνέχιζε να ανεβαίνει αν δεν είχε πρώτα ενεργοποιηθεί η θερμοκρασία. Αν σας ενδιαφέρει ένα ανώτατο όριο CPU, παρακολουθήστε μόνοι σας τη στήλη ή προσθέστε κατώφλι.

**Τι αποδεικνύει και τι όχι.** Κάθε συνθετικός ακροατής μεταδίδει τυχαίο θόρυβο χαμηλού πλάτους, όχι πραγματικό σήμα RADE. Αυτό αρκεί ώστε να τρέξει ολόκληρη η αλυσίδα αποκωδικοποίησης και να καταναλωθούν πόροι, οπότε τα νούμερα των πόρων είναι ουσιαστικά — αλλά η δοκιμή δεν λέει τίποτα για την ποιότητα αποκωδικοποίησης ή τη συνέχεια του ήχου υπό φόρτο. Επιβεβαιώστε τα με πραγματικούς ακροατές σε πραγματικό σήμα.

## Αν οι πυρήνες κορεστούν πριν το γόνατο

Το RADE καρφισώνει έντονα ένα νήμα ανά στιγμιότυπο, οπότε μεμονωμένοι πυρήνες μπορούν να φτάσουν τους ~90 °C ενώ η συνολική CPU φαίνεται αδρανής. Το `rade_helper.py` κατανέμει τις αλυσίδες στους πυρήνες εκ περιτροπής για να το αντισταθμίσει. Αν μια εκτέλεση δείχνει πρόωρα θερμικά γόνατα, διευρύνετε την κατανομή πυρήνων ανά πελάτη:

```
RADE_CORES_PER_CLIENT=3 ./rade.sh restart
```

Δεν χρειάζεται επεξεργασία αρχείου — πρόκειται για μεταβλητή περιβάλλοντος (δείτε [Μεταβλητές περιβάλλοντος](#)). το `RADE_PIN_CORES=0` απενεργοποιεί εντελώς το καρφίτσωμα.

## Ενημέρωση του RADE v1

Η ενσωμάτωση στο PhantomSDR-Plus (`rade_helper.py`, διορθώσεις frontend) είναι μόνο γέφυρα — όλη η λογική αποκωδικοποίησης RADE βρίσκεται στο αποθετήριο `radae`. Οι ενημερώσεις είναι επομένως σχεδόν πάντα ένα απλό `git pull` + νέα μεταγλώττιση, χωρίς αλλαγές στο ίδιο το PhantomSDR-Plus.

### Τυπική ενημέρωση (νέος κώδικας, ίδιο μοντέλο)

```
1. Pull latest radae code
cd ~/radae
git pull

2. Rebuild lpcnet_demo (in case C code changed)
cd build
cmake ..
make -j$(nproc)

3. Restart the sidecar – no server restart needed
cd ~/PhantomSDR-Plus
./rade.sh restart
```

Αυτό είναι όλο. Καμία νέα μεταγλώττιση του frontend, καμία επανεκκίνηση διακομιστή, καμία επεξεργασία αρχείων.

### Μόνο νέα βάρη μοντέλου

Αν κυκλοφορήσει νέο σημείο ελέγχου (π.χ. `model20`) χωρίς αλλαγές κώδικα, στρέψτε τον sidecar στα νέα βάρη με τη μεταβλητή περιβάλλοντος — δεν απαιτείται επεξεργασία αρχείου:

```
One-off: start with new model
RADE_MODEL=~/.radae/model20/checkpoints/checkpoint_epoch_100.pth ./rade.sh start

Or permanently – add to your shell profile (~/.bashrc):
export RADE_MODEL=~/.radae/model20/checkpoints/checkpoint_epoch_100.pth
```

### Τι απαιτεί κάθε είδος αλλαγής

Τι άλλαξε στο <code>radae</code>	Απαιτούμενη ενέργεια
Νέα βάρη μοντέλου (νέο αρχείο <code>.pth</code> )	Ορίστε τη μεταβλητή <code>RADE_MODEL</code> , <code>./rade.sh restart</code>
Αλλαγή κώδικα στο <code>radae_rxe.py</code>	<code>git pull</code> , <code>./rade.sh restart</code>
Άλλαξε ο κώδικας C του <code>lpcnet_demo</code>	<code>git pull</code> , νέα μεταγλώττιση, <code>./rade.sh restart</code>
Το <code>radae_rxe.py</code> μετονομάστηκε ή μετακινήθηκε	Ενημερώστε τη διαδρομή <code>RADAE_RX</code> στο <code>rade_helper.py</code> (μία γραμμή)
Το όρισμα <code>--model_name</code> μετονομάστηκε	Ενημερώστε το <code>radae_cmd</code> στο <code>rade_helper.py</code> (μία γραμμή)
Νέος ρυθμός δειγματοληψίας εξόδου ( $\neq 16000$ Hz)	Ενημερώστε το <code>SPS_OUT</code> στο <code>rade_helper.py</code> + το <code>createBuffer()</code> στο <code>audio.js</code>
Το RADE v2 χρησιμοποιεί άλλο εκτελέσιμο	Ενημερώστε το <code>RADAE_RX</code> στο <code>rade_helper.py</code> (μία γραμμή)

### Επαλήθευση ότι η ενημέρωση πέτυχε

Μετά την επανεκκίνηση, επιβεβαιώστε ότι τρέχει ο νέος κώδικας:

```
Check sidecar picked up new radae_rxe.py
./rade.sh status

Tail log to see startup lines
tail -20 ~/PhantomSDR-Plus/rade.log
```

Το αρχείο καταγραφής πρέπει να δείχνει τη διαδρομή μοντέλου που περιμένετε:

```
[RADE] model : /home/sv1btl/radae/model19_check3/checkpoints/checkpoint_epoch_100.pth
```

## Παραμένετε ενημερωμένοι

Εγγραφείτε στη σελίδα εκδόσεων του αποθετηρίου radae για να ειδοποιείστε για νέα μοντέλα και ενημερώσεις κώδικα:

```
https://github.com/drowe67/radae/releases
```

Ελέγξτε το ιστολόγιο του FreeDV για ανακοινώσεις σχετικά με νέες κυματομορφές και μοντέλα RADE:

```
https://freedv.org/blog/
```

Αναπτύχθηκε και δοκιμάστηκε στο *PhantomSDR-Plus* (fork *sv1btl/PhantomSDR-Plus*) με *RX-888 MK2* σε *Ubuntu 24.04*. Ο διακομιστής C++ δεν τροποποιείται.

Το RADE αναπτύσσεται από τον *David Rowe VK5DGR* και την ομάδα *FreeDV*. Δείτε το [freedv.org/radio-autoencoder](https://freedv.org/radio-autoencoder).

## Πλήρης απεγκατάσταση και καθαρή επανεγκατάσταση

Να η πλήρης αποδόμηση πριν εκτελέσετε ξανά το `install_rade.sh`:

1. Σταματήστε τον sidecar `cd ~/PhantomSDR-Plus && ./rade.sh stop`
2. Αφαιρέστε το αποθετήριο radae και τη μεταγλώττιση `rm -rf ~/radae`
3. Αφαιρέστε το torch (εγκατεστημένο στον τοπικό κατάλογο χρήστη) `pip3 uninstall -y torch rm -rf ~/.local/lib/python3.11/site-packages/torch*`
4. Αφαιρέστε τα πακέτα Python του apt (προαιρετικό — παραλείψτε το αν τα χρησιμοποιούν άλλα) `sudo apt-get remove -y python3-numpy python3-scipy python3-matplotlib python3-websockets sudo apt-get autoremove -y`
5. Επαληθεύστε ότι όλα αφαιρέθηκαν `python3 -c "import torch" 2>&1 # should say ModuleNotFoundError ls ~/radae 2>&1 # should say No such file or directory`
6. Καθαρή εγκατάσταση `chmod +x ~/PhantomSDR-Plus/install_rade.sh ~/PhantomSDR-Plus/install_rade.sh`

## ΚΕΦΑΛΑΙΟ 10

# RADE — γενική εγκατάσταση (Linux)

RADE\_General\_INSTALL\_MANUAL\_LINUX.md

## RADE v1 — Οδηγός χειροκίνητης εγκατάστασης

**Όλα τα Linux — Ubuntu / Debian / Fedora / Arch / Raspberry Pi OS · PhantomSDR-Plus**

**Προϋπόθεση:** τα διορθωμένα αρχεία (`rade_helper.py`, `rade.sh`, `audio.js`, το `App.svelte`) βρίσκονται ήδη στο δέντρο καταλόγων του PhantomSDR-Plus. Ο οδηγός αυτός χτίζει τα υπόλοιπα γύρω από αυτά.

**ΣΗΜΑΝΤΙΚΟ**

**Δεν χρειάζεστε αυτόν τον οδηγό για μια κανονική εγκατάσταση.** Το `./install_rade.sh` στον φάκελο PhantomSDR-Plus κάνει για εσάς κάθε βήμα που ακολουθεί — πακέτα συστήματος, βιβλιοθήκες Python, το `build` του `radare`, τον έλεγχο των βαρών του μοντέλου και την εκκίνηση του `sidcar` — και δουλεύει εξίσου σε συστήματα `apt`, `pacman`, `dnf` και `zypper`. Το `./install.sh` και τα τέσσερα `scripts` των διανομών προσφέρονται να το τρέξουν στο πλαίσιο της κανονικής εγκατάστασης. Ακολουθήστε αυτόν τον οδηγό μόνο για χειροκίνητη εγκατάσταση, για σύστημα που δεν καλύπτει το `script`, ή για να επισκευάσετε ένα βήμα με το χέρι.

**Χρήστες Raspberry Pi:** αυτός είναι και ο δικός σας οδηγός. Το Raspberry Pi OS είναι Debian, οπότε ακολουθήστε παντού τα τμήματα για Debian και διαβάστε τις σημειώσεις **Raspberry Pi / Bookworm** όπου εμφανίζονται — καλύπτουν τα δύο σημεία που διαφέρουν σε ένα Pi: το PEP 668 και το `torch wheel` μόνο για CPU.

## Βήμα 1 — Πακέτα συστήματος

Επιλέξτε το τμήμα που αντιστοιχεί στη διανομή σας.

### Ubuntu / Debian / Raspberry Pi OS

```
sudo apt update
sudo apt install -y \
 build-essential cmake git \
 python3 python3-pip \
 nodejs npm \
 alsa-utils
```

### Fedora / RHEL / Rocky

```
sudo dnf install -y \
 gcc gcc-c++ cmake git \
 python3 python3-pip \
 nodejs npm \
 alsa-utils
```

## Arch / Manjaro

```
sudo pacman -Sy --needed \
 base-devel cmake git \
 python python-pip \
 nodejs npm \
 alsa-utils
```

### Έλεγχος έκδοσης Node.js (όλες οι διανομές)

Η μεταγλώττιση του frontend του RADE απαιτεί Node.js 22 ή νεότερο:

```
node --version
```

Αν είναι κάτω από 22.x, εγκαταστήστε το με `nvm` (οποιαδήποτε διανομή, χωρίς root):

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.4/install.sh | bash
export NVM_DIR="$HOME/.nvm"
[-s "$NVM_DIR/nvm.sh"] && \. "$NVM_DIR/nvm.sh"
nvm install 22 && nvm use 22 && nvm alias default 22
```

**Όχι μέσω NodeSource.** Το `deb.nodesource.com` και το `rpm.nodesource.com` απαντούν πλέον με HTTP 403 σε κάθε διαδρομή αποθετηρίου, οπότε η παλιά εντολή `curl -fsSL https://deb.nodesource.com/setup_NN.x | sudo -E bash -` δεν λειτουργεί πια — και σε Debian/Ubuntu αφήνει πίσω μια πηγή `apt` που χαλάει κάθε επόμενο `apt update`. Για τον ίδιο λόγο τα σενάρια εγκατάστασης χρησιμοποιούν `nvm`.

## Βήμα 2 — Πακέτα Python

### Ubuntu 24.04 και παλαιότερα / Fedora / Arch

Σε αυτά τα συστήματα λειτουργεί το σκέτο `pip3 install`:

```
pip3 install websockets matplotlib numpy scipy

torch — CPU-only build (~150–250 MB, avoids the ~3 GB CUDA wheel)
Use this if the server has no GPU, which is the typical case for a WebSDR
pip3 install torch --index-url https://download.pytorch.org/whl/cpu
```

Αν ο διακομιστής σας **διαθέτει** GPU NVIDIA με εγκατεστημένο CUDA, μπορείτε να παραλείψετε το `--index-url` για να λάβετε την πλήρη έκδοση με CUDA. Δεν υπάρχει όφελος επιδόσεων για το RADE — το `radea_rxe.py` χρησιμοποιεί το PyTorch μόνο για πράξεις πινάκων στη CPU.

### Ubuntu 23.04+ / Debian Bookworm+ / Raspberry Pi OS (συστήματα PEP 668)

Οι διανομές αυτές μπλοκάρουν το σκέτο `pip3 install`. Προσθέστε τη σημαία:

```
All packages except torch
pip3 install --break-system-packages websockets matplotlib numpy scipy

torch — CPU-only build
pip3 install --break-system-packages torch \
 --index-url https://download.pytorch.org/whl/cpu
```

**Raspberry Pi / Bookworm — δύο κανόνες που δαγκώνουν:**

1. Κάθε `pip3 install` απαιτεί `--break-system-packages` (εφαρμογή του PEP 668). Χωρίς αυτό η εγκατάσταση μπλοκάρει εντελώς.
2. Ένα σκέτο `pip3 install torch` κατεβάζει την έκδοση CUDA (~3 GB). Σε ένα Pi δεν υπάρχει CUDA — χρησιμοποιήστε το ευρετήριο μόνο για CPU ώστε να λάβετε την ελαφριά έκδοση ARM64 (~150 MB).

Εναλλακτικά, χρησιμοποιήστε εικονικό περιβάλλον για να αποφύγετε εντελώς τη σημαία:

```
python3 -m venv ~/rade-venv
source ~/rade-venv/bin/activate
pip install websockets matplotlib numpy scipy torch \
 --index-url https://download.pytorch.org/whl/cpu
```

Αν χρησιμοποιείτε `venv`, προτάξτε σε όλες τις επόμενες κλήσεις `python3` αυτού του οδηγού το `source ~/rade-venv/bin/activate` ή χρησιμοποιήστε την πλήρη διαδρομή `~/rade-venv/bin/python3`.

**Επαλήθευση**

```
python3 -c "import websockets, matplotlib, torch, numpy, scipy; print('All OK!')"
```

Αναμενόμενη έξοδος:

```
All OK
```

**Βήμα 3 — Κλωνοποιήστε και μεταγλωττίστε το αποθετήριο `radeae`**

Ο αποκωδικοποιητής RADE βρίσκεται σε ξεχωριστό αποθετήριο από το `codec2`. Το `freedv_rx` του `codec2` **δεν** υποστηρίζει RADE v1.

```
Remove any previous incomplete clone
rm -rf ~/radeae

Clone
git clone https://github.com/drowe67/radae.git ~/radeae
cd ~/radeae

Build
mkdir build && cd build
cmake ..
make -j$(nproc)
```

**Επαληθεύστε ότι μεταγλωττίστηκε το `lpcnet_demo`**

```
ls -la ~/radeae/build/src/lpcnet_demo
```

Αναμενόμενο: το εκτελέσιμο υπάρχει και είναι εκτελέσιμο.

## Επαληθεύστε ότι υπάρχουν τα βάρη του μοντέλου

```
ls ~/radeae/model19_check3/checkpoints/checkpoint_epoch_100.pth
```

Αναμενόμενο: εμφανίζεται το αρχείο `.pth`. Τα βάρη συνοδεύουν το αποθετήριο — δεν απαιτείται ξεχωριστή λήψη.

## Βήμα 4 — Επαληθεύστε την αλυσίδα αποκωδικοποίησης

Το βήμα αυτό επιβεβαιώνει ότι η πλήρης αλυσίδα εκτός σύνδεσης λειτουργεί πριν τη συνδέσετε με τον περιηγητή. Εκτελέστε από το `~/radeae`:

```
cd ~/radeae
```

### 4α — Δημιουργήστε δοκιμαστικό σήμα κωδικοποιημένο σε RADE

```
./inference.sh model19_check3/checkpoints/checkpoint_epoch_100.pth \
wav/brian_g8sez.wav /dev/null \
--rate_Fs --pilots --pilot_eq --eq_ls --cp 0.004 \
--write_rx rx.f32 --auxdata
```

Περιμένετε να ολοκληρωθεί. Οι τελευταίες γραμμές που εμφανίζονται πρέπει να είναι:

```
loss: 0.741 Auxdata BER: 0.012
```

### 4β — Αποκωδικοποιήστε και αναπαράγετε

```
cat rx.f32 \
| python3 radeae_rxe.py \
--model_name model19_check3/checkpoints/checkpoint_epoch_100.pth \
| ./build/src/lpcnet_demo -fargan-synthesis - - \
| aplay -f S16_LE -r 16000
```

Θα πρέπει να ακούσετε φωνή. Η έξοδος δείχνει την απόκτηση συγχρονισμού:

```
1 state: search ...
5 state: sync ... SNRdB: 4.03 uw_err: 0
Playing raw data 'stdin' : Signed 16 bit Little Endian, Rate 16000 Hz, Mono
```

**Τα μηνύματα `underrun!!!` σε αυτή τη δοκιμή είναι αναμενόμενα και ακίνδυνα.** Εμφανίζονται επειδή το `radeae_rxe.py` επεξεργάζεται πιο αργά από την είσοδο/έξοδο του αρχείου. Δεν εμφανίζονται σε ζωντανή λήψη — ο περιηγητής τροφοδοτεί τον ήχο σε πραγματικό χρόνο.

### 4γ — Επαληθεύστε ότι ξεκινά σωστά ο `sidecar`

```
python3 ~/PhantomSDR-Plus/rade_helper.py
```

Αναμενόμενη έξοδος (χωρίς γραμμές WARNING):

```
[RADE] helper starting on ws://0.0.0.0:8074
[RADE] rade_rx.py : /home/<user>/rade/rade_rxe.py
[RADE] model : /home/<user>/rade/model19_checkpoints/checkpoint_epoch_100.pth
[RADE] lpcnet_demo : /home/<user>/rade/build/src/lpcnet_demo
[RADE] auxdata : ON (default)
[RADE] torch threads : 1 per instance (RADE_TORCH_THREADS to override)
[RADE] architecture : per-connection (each user tunes independently)
[RADE] listening - waiting for PhantomSDR-Plus clients
```

Πατήστε **Ctrl+C** για διακοπή.

Το `<user>` στις παραπάνω διαδρομές είναι ο λογαριασμός με τον οποίο έχετε συνδεθεί — σε μια τυπική εικόνα Raspberry Pi OS γίνονται `/home/pi/rade/...`.

## Βήμα 5 — Μεταγλωττίστε το frontend του PhantomSDR-Plus

```
cd ~/PhantomSDR-Plus/frontend

Only if node_modules is missing:
npm install

cd ~/PhantomSDR-Plus
./recompile.sh
```

Προσέξτε για σφάλματα του αναλυτή Vite/acorn. Τα διορθωμένα αρχεία αποφεύγουν σκόπιμα τα `?.`, `??` και τα σκέτα `catch {}`, ώστε να συμμορφώνονται με τον περιορισμό του acorn.

## Βήμα 6 — Έλεγχος του sidocar

**Κανονικά δεν χρειάζεστε ξεχωριστό σενάριο ελέγχου για το RADE.** Τα σενάρια εκκίνησης (`start-rx888mk2.sh`, `start-airspyhf.sh`, `start-rtl.sh`, `start-rspla.sh`, `start-fobos-hf.sh`, `start-fobos.sh`, `start-hackrf.sh`) ξεκινούν μόνα τους το sidocar μόλις σηκωθεί ο διακομιστής, το επανεκκινούν αν τερματιστεί, και αναφέρουν την κατάστασή του· το `stop-websdr.sh` το σταματά μαζί με τον διακομιστή. Πηγαίνετε κατευθείαν στο Βήμα 7.

**Το `rade.sh` εξακολουθεί να διατίθεται για αυτόνομη χρήση — για να τρέξετε το sidocar χωρίς τον launcher του PhantomSDR-Plus, ή για να το δοκιμάσετε μόνο του. Οι εντολές του, το αρχείο καταγραφής του και οι σχετικές προειδοποιήσεις για τον watchdog βρίσκονται στο Έλεγχος του sidocar.**

## Βήμα 7 — Ανοίξτε τη θύρα 8074

Ο sidocar ακούει στη θύρα TCP **8074**. Ο περιηγητής συνδέεται απευθείας σε αυτή τη θύρα. Πρέπει να την ανοίξετε χειροκίνητα.

## ufw (Ubuntu / Debian / Raspberry Pi OS)

```
sudo ufw allow 8074/tcp && sudo ufw reload
```

## firewalld (Fedora / RHEL / Rocky)

```
sudo firewall-cmd --add-port=8074/tcp --permanent
sudo firewall-cmd --reload
```

## iptables (οποιαδήποτε διανομή, μόνιμα)

```
sudo iptables -I INPUT -p tcp --dport 8074 -j ACCEPT

Ubuntu / Debian / Raspberry Pi OS – persist across reboots:
sudo apt install iptables-persistent
sudo netfilter-persistent save

Fedora / RHEL – persist:
sudo service iptables save
```

## Δρομολογητής

Προσθέστε κανόνα NAT/προώθησης θύρας: **TCP 8074 → τοπική IP διακομιστή : 8074**

## Δοκιμή από έξω

Χρησιμοποιήστε το <https://portchecker.co> και ελέγξτε τη θύρα 8074 στο δημόσιο όνομα κεντρικού υπολογιστή σας. **Μη** δοκιμάζετε με `curl` από τον ίδιο τον διακομιστή — το NAT hairpin δίνει ψευδή αποτελέσματα «Connection refused» ακόμη και όταν η θύρα είναι ανοιχτή.

## Εναλλακτικά — διαμεσολαβητής Nginx (αν η θύρα 8074 φράσσεται από τον πάροχο)

Προσθέστε μέσα στο υπάρχον μπλοκ `server {}`:

```
location /rade {
 proxy_pass http://127.0.0.1:8074;
 proxy_http_version 1.1;
 proxy_set_header Upgrade $http_upgrade;
 proxy_set_header Connection "upgrade";
 proxy_set_header Host $host;
 proxy_read_timeout 3600s;
}
```

```
sudo nginx -t && sudo nginx -s reload
```

Έπειτα επεξεργαστείτε το `audio.js` μέσα στη `setRADEDecoding()`:

```
// Change:
var helperUri = uri || ('ws://' + window.location.hostname + ':8074');
// To:
var helperUri = uri || ('ws://' + window.location.host + '/rade');
```

Μεταγλωττίστε ξανά το frontend μετά από αυτή την αλλαγή (`./recompile.sh`).

## Βήμα 8 — Εκκινήστε τον διακομιστή

Εκκινήστε το PhantomSDR-Plus ως συνήθως. Το RADE **δεν** ξεκινά αυτόματα:

```
cd ~/PhantomSDR-Plus
./start-rx888mk2.sh # or start-airspyhf.sh,
 # start-rtl.sh, start-rspla.sh,
 # start-fobos-hf.sh, start-fobos.sh,
 # start-hackrf.sh
```

Χρησιμοποιήστε τον εκκινητή που αντιστοιχεί στον δέκτη σας. Καθένας περιλαμβάνει δικό του watchdog και αρχείο καταγραφής· το `./stop-websdr.sh` σταματά όποιον εκτελείται.

## Βήμα 9 — Εκκινήστε το RADE

```
cd ~/PhantomSDR-Plus
./rade.sh start
./rade.sh status
tail -f rade.log
```

Αναμενόμενη καταγραφή:

```
[RADE] sidecar starting at ...
[RADE] helper starting on ws://0.0.0.0:8074
[RADE] rade_rx.py : /home/<user>/rade/rade_rxe.py
[RADE] model : /home/<user>/rade/model19_checkpoint3/checkpoints/checkpoint_epoch_100.pth
[RADE] lpcnet_demo : /home/<user>/rade/build/src/lpcnet_demo
[RADE] auxdata : ON (default)
[RADE] torch threads : 1 per instance
[RADE] architecture : per-connection (each user tunes independently)
[RADE] listening : waiting for PhantomSDR-Plus clients
```

## Βήμα 10 — Χρήση του RADE στον περιηγητή

1. Ανοίξτε τη διεπαφή web του PhantomSDR-Plus
2. Βρείτε ενεργούς σταθμούς στο <https://qso.freedv.org>
3. Συντονιστείτε στη συχνότητα ένδειξης του σταθμού
4. Στις **Decoder Options** επιλέξτε:
5. **RADE v1 — RADEL (LSB)** για 40 m / 80 m / 160 m ( $\leq 10$  MHz)
6. **RADE v1 — RADEU (USB)** για 20 m / 17 m / 15 m / 12 m / 10 m ( $> 10$  MHz)
7. Κάντε κλικ στο **Decoder: ON**

Τα βήματα 4 και 5 μπορούν να αντικατασταθούν από ένα μόνο πάτημα: τα κουμπιά **RADEL / RADEU** βρίσκονται δίπλα στην επικεφαλίδα **Modes selector** και μέσα στα αναδυόμενα παράθυρα **Modes** και **Bands**. Ένα πάτημα επιλέγει τον αποκωδικοποιητή, τον ανάβει και φέρνει το παράθυρο RADE στην οθόνη· δεύτερο πάτημα σταματά το RADE. Δείτε το [εγχειρίδιο RADE](#).

Ένδειξη	Σημασία
Κόκκινο — «Connecting to sidecar...»	Η θύρα 8074 δεν είναι προσβάσιμη ή ο sidecar δεν εκτελείται
Κίτρινο — «Searching for signal...»	Ο sidecar συνδέθηκε, δεν εντοπίστηκε ακόμη πλαίσιο RADE (αφήστε ~1,5 s)
Πράσινο — «Synced · SNR x.x dB»	Αποκωδικοποίηση — ακούγεται ομιλία

## Σημείωση για τη CPU

Κάθε χρήστης RADE καταναλώνει ~8-10 % ενός πυρήνα (τα νήματα PyTorch περιορίζονται σε 1 από τον sidecar). Αν ακούτε διακοπές ήχου, αυξήστε σε 2 νήματα:

```
RADE_TORCH_THREADS=2 ./rade.sh restart
```

Ταυτόχρονοι χρήστες	Κατά προσέγγιση CPU
1	~9 %
5	~45 %
10	~90 %
20	~180 %

Οι τιμές αυτές μετρήθηκαν σε x86\_64. Ένα Raspberry Pi αποκωδικοποιεί RADE μια χαρά, αλλά το κόστος ανά χρήστη είναι μεγαλύτερο και ο παραπάνω πίνακας δεν μεταφέρεται — μετρήστε τη δική σας πλακέτα με `top` ενώ ένας χρήστης είναι συγχρονισμένος, πριν διαφημίσετε όριο χρηστών.

## Μελλοντική ενημέρωση του RADE

```
cd ~/radae
git pull
cmake -S . -B build && make -C build -j$(nproc)

cd ~/PhantomSDR-Plus
./rade.sh restart
```

Δεν χρειάζεται νέα μεταγλώττιση του frontend ούτε επανεκκίνηση του διακομιστή, εκτός αν άλλαξε το ίδιο το `rade_helper.py`.

*Δοκιμασμένο σε Ubuntu 24.04 (x86\_64) και Raspberry Pi 4 / Raspberry Pi OS Bookworm (ARM64, Python 3.11). Fork PhantomSDR-Plus: [sv1bt/PhantomSDR-Plus](https://github.com/sv1bt/PhantomSDR-Plus). Το RADE αναπτύχθηκε από τον David Rowe VK5DGR και την ομάδα FreeDV — <https://freedv.org>*

## ΚΕΦΑΛΑΙΟ 11

## Επεξεργασία παραλλαγών

EDITING\_VARIANTS.md

## PhantomSDR-Plus — Πώς να επεξεργαστείτε τις παραλλαγές του frontend

(αφού τα τέσσερα αρχεία `App__*_smeter_.svelte` συγχωνεύτηκαν σε ένα `App.svelte`)

## 1. Τι είναι οι παραλλαγές

Δύο ανεξάρτητες επιλογές, που δίνουν τέσσερις συνδυασμούς:

<code>smeter = "analog"</code>	το όργανο με κινούμενη βελόνα
<code>= "digital"</code>	το τμηματοποιημένο όργανο ράβδου
<code>layout = "v1"</code>	επιλογές τρόπων λειτουργίας πάνω, επιλογές μπάντας κάτω
<code>= "v2"</code>	επιλογές μπάντας πάνω, επιλογές τρόπων λειτουργίας κάτω (μετακινεί επίσης τη σειρά κουμπιών λεπτού συντονισμού στο κινητό)

Και οι τέσσερις βρίσκονται σε **μία** δόμηση, που σερβίρεται στο `/`. Ο επιλογέας `☞`πάνω δεξιά εναλλάσσει μεταξύ τους μέσα στη σελίδα που ήδη τρέχει: θέτει τα props `smeter/layout` του `App.svelte`, οπότε τίποτα δεν ξαναφορτώνεται και ο ήχος, ο καταρράκτης και οι αποκωδικοποιητές συνεχίζουν κανονικά. Η επιλογή αποθηκεύεται ανά φυλλομετρητή στο `localStorage` με κλειδί `phantom.variant`, και υπερισχύει της προεπιλογής της δόμησης στην επόμενη επίσκεψη.

Επομένως η παραλλαγή που επιλέγει ο sysop στο `recompile.sh` είναι η **αρχική** παραλλαγή — αυτό που βλέπει ένας επισκέπτης την πρώτη φορά, πριν αγγίξει το μενού.

URL	τι είναι
<code>/</code>	η σελίδα για υπολογιστή· και οι τέσσερις παραλλαγές, εναλλάξιμες σε χρόνο εκτέλεσης
<code>/mobile/</code>	ξεχωριστή σελίδα, χτίζεται από το <code>build-mobile.sh</code> — δεν είναι παραλλαγή

Μέχρι τις 10/08/2026 κάθε παραλλαγή ήταν δική της δόμηση στα `/analog/`, `/digital/`, `/v2-analog/` και `/v2-digital/`, και ο επιλογέας πλοηγούσε ανάμεσά τους — γι' αυτό η εναλλαγή ξαναφόρτωνε τη σελίδα και έκοβε τον ήχο. Αυτές οι δομήσεις δεν υπάρχουν πια· σε καθεμία από τις τέσσερις διαδρομές μένει μόνο μια μικρή `index.html` που ανακατευθύνει στο `/`, την οποία γράφει το `frontend/make-redirect-stubs.sh` σε κάθε δόμηση ώστε οι παλιοί σελιδοδείκτες να συνεχίσουν να δουλεύουν.

Η παραλλαγή **δεν** γράφεται σε κανένα αρχείο πηγαίου κώδικα. Το `src/main.js` είναι ένα σταθερό σημείο εισόδου που δίνει στο component δύο τιμές που ορίζονται κατά τη δόμηση:

```
const app = new App({
 target: document.getElementById('app'),
 props: { smeter: __PHANTOM_SMETER__, layout: __PHANTOM_LAYOUT__ }
})
```

Το `vite.config.js` τις συμπληρώνει κατά τη δόμηση, παίρνοντας το πρώτο από τα παρακάτω που είναι ορισμένο:

1. **PHANTOM\_SMETER / PHANTOM\_LAYOUT** — μεταβλητές περιβάλλοντος. Τώρα που καταργήθηκαν οι επιμέρους δομήσεις παραλλαγών, κανείς δεν τις θέτει· εξακολουθούν όμως να λειτουργούν για μια εφάπαξ δόμηση με άλλη αρχική παραλλαγή.
2. **frontend/variant.json** — η προεπιλογή του ιστότοπου, την οποία γράφει το `recompile.sh`. Η δόμηση της ρίζας (/) δεν περνά παραλλαγή, οπότε καταλήγει εδώ.
3. **analog / v1** — η εφεδρική τιμή.

Αυτές οι δύο τιμές είναι μόνο η αφητηρία. Το `App.svelte` τις αντικαθιστά κατά την αρχικοποίηση από το `localStorage` αν ο επισκέπτης έχει ήδη επιλέξει παραλλαγή, και ο επιλογέας τις θέτει ξανά σε κάθε εναλλαγή.

Μέσα στο `App.svelte` τα props μετατρέπονται σε δύο σημαίες στις γραμμές 20-21:

```
$: isAnalog = smeter !== "digital";
$: isV2 = layout === "v2";
```

Χρησιμοποιείτε αυτές τις δύο σημαίες — ποτέ τα `smeter/layout` απευθείας — ώστε όλο το αρχείο να διαβάζεται με τον ίδιο τρόπο.

## 2. Τι κάνει κάθε αρχείο

### `frontend/src/main.js` — 12 γραμμές

Το σημείο εισόδου. Τίποτα άλλο εκτός από τη δημιουργία του App και τις δύο τιμές `__PHANTOM_*` που περνά ως props. Είναι **σταθερό** — ούτε το `recompile.sh` ούτε τα σενάρια δόμησης το ξαναγράφουν πλέον, και ακριβώς αυτό επιτρέπει στις παραλλαγές να χτίζονται ταυτόχρονα.

**Μη βάζετε κώδικα εφαρμογής εδώ.** Είναι το αρχείο εισόδου, όχι χώρος για λογική.

### `frontend/src/App.svelte` — ~17130 γραμμές

Ολόκληρη η εφαρμογή: καταρράκτης, συντονισμός, αποκωδικοποιητές, σελιδοδείκτες, πάνελ, διάταξη υπολογιστή ΚΑΙ η προσαρμοστική διάταξη κινητού. Σχεδόν κάθε αλλαγή που θα κάνετε πάει εδώ.

### `frontend/src/lib/SMeterAnalog.svelte` — 824 γραμμές

Το όργανο με κινούμενη βελόνα, πλήρες: σχεδίαση της πλάκας, βελόνα, η εναλλαγή φωτεινής/σκοτεινής όψης με το κλειδί της στο `localStorage`, και η εξομάλυνση της βελόνας.

- Prop εισόδου: `dbm` — η ΒΑΘΜΟΝΟΜΗΜΕΝΗ ισχύς σε dBm.
- Δέχεται επίσης: `mobile` — μικρότερα χαρακτηριστικά canvas για τη διάταξη κινητού.
- Ρυθμιστικό: `SMOOTH_TIME_MS` στη γραμμή 55. Μικρότερο = ταχύτερη βελόνα, μεγαλύτερο = πιο ομαλή. Τώρα 16 (ένα καρέ στα 60 fps, πρακτικά το κατώτατο όριο).

### Οι τρεις όψεις του οργάνου — και το `smeter_theme.sh`

Το αναλογικό όργανο έχει τρεις όψεις (φόντα), όλες σχεδιασμένες από τον ίδιο κώδικα:

<code>dark</code>	το αρχικό σκούρο βουρτσισμένο μέταλλο
<code>amber</code>	ανοιχτή, απαλή γκρίζα όψη
<code>vintage</code>	ζεστή όψη σε παλαιωμένο κεχριμπαρένιο χρώμα

**Μην το υπερδύετε με την παραλλαγή `smeter = analog / digital` παραπάνω.** Η παραλλαγή επιλέγει ποιο εξάρτημα οργάνου εμφανίζεται· η όψη επιλέγει πώς σχεδιάζεται το αναλογικό. Αποθηκεύονται σε διαφορετικά κλειδιά `localStorage` (`phantom.variant` και `smeterTheme`) και ορίζονται σε διαφορετικά σημεία.

**Η πλευρά του επισκέπτη.** Ο καμβάς του οργάνου είναι κουμπί (`role="button"`, επίσης `Enter/Space` από το πληκτρολόγιο, με την υπόδειξη «click to switch style» από κάτω). Κάθε κλικ εναλλάσσει `dark` → `amber` → `vintage` και αποθηκεύει την επιλογή στο `localStorage.smeterTheme`. Διαβάζεται ξανά σε κάθε φόρτωση σελίδας, άρα η επιλογή του επισκέπτη επιβιώνει από ανανεώσεις και επανεκκινήσεις του φυλλομετρητή επ' αόριστον. Σκόπιμα **δεν** είναι cookie: ένα cookie θα στελνόταν στον διακομιστή σε κάθε αίτημα καταρράκτη και ήχου, για μια τιμή που ο διακομιστής δεν διαβάζει ποτέ.

**Η πλευρά του διαχειριστή.** Δύο πράγματα στο `SMeterAnalog.svelte` ορίζουν τι βλέπουν όλοι οι υπόλοιποι:

Γραμμή	Τι κάνει
<code>let smeterTheme = 'dark';</code>	Η όψη που παίρνει ένας φυλλομετρητής <b>χωρίς αποθηκευμένη επιλογή</b> .
<code>const SMETER_PREF_VERSION = 2;</code>	Ένας μετρητής. Όταν η αποθηκευμένη έκδοση ενός φυλλομετρητή είναι πίσω από αυτόν, ξεχνά την αποθηκευμένη όψη <b>μία φορά</b> και παίρνει την παραπάνω προεπιλογή. Αυξήστε τον για να μετακινήσετε επισκέπτες που έχουν ήδη επιλέξει.

Επομένως, αλλάζοντας μόνο την προεπιλογή φτάνετε μόνο σε ολοκαίνουργιους επισκέπτες. Και οι δύο αλλαγές μαζί μετακινούν τους πάντες — γι' αυτό υπάρχει το `smeter_theme.sh` στη ρίζα του αποθετηρίου:

```
./smeter_theme.sh # αναφορά της τρέχουσας προεπιλογής και επιλογή από μενού
./smeter_theme.sh vintage # άμεσος ορισμός, μετά προσφορά για ανακατασκευή
./smeter_theme.sh dark --build # εκτέλεση του frontend/build-all.sh χωρίς ερώτηση
./smeter_theme.sh amber --no-reset # μόνο νέοι επισκέπτες — οι υπάρχοντες μένουν ως έχουν
```

Αναφέρει την τρέχουσα προεπιλογή, ορίζει τη νέα, αυξάνει τον μετρητή επαναφοράς (πάντα προς τα πάνω — **ποτέ μην τον μειώνετε**, ένας μικρότερος αριθμός απλώς εμποδίζει την επαναφορά) και μετά προσφέρει την ίδια επιλογή δόμησης frontend με το `recompile.sh`. **Δεν** αγγίζει το backend και **δεν** ξαναγράφει το `variant.json`.

Μια αλλαγή φτάνει σε έναν συνδεδεμένο ακροατή μόνο όταν ξαναφορτώσει τη σελίδα του — η προεπιλογή είναι μεταγλωττισμένη μέσα στο πακέτο JS και η επαναφορά τρέχει κατά τη φόρτωση. Δεν υπάρχει ζωντανή προώθηση. Μια απλή ανανέωση αρκεί, αφού το vite βάζει hash στα ονόματα αρχείων του πακέτου.

Η εφάπαξ επαναφορά του μετρητή υπερσχύει για όλους, ακόμη και για επισκέπτες που επέλεξαν συνειδητά τη δική τους όψη. Αυτό είναι σκόπιμο: είναι ο μόνος τρόπος να επανέλθει όλος ο σταθμός σε μία εμφάνιση. Χρησιμοποιήστε `--no-reset` αν προτιμάτε να τους αφήσετε ήσυχους.

## frontend/src/lib/SMeterDigital.svelte — 184 γραμμές

Το τμηματοποιημένο όργανο ράβδου, πλήρες.

- Props εισόδου: `rawDb` — η ΑΚΑΤΕΡΓΑΣΤΗ τιμή του `audio.getPowerDb()`
- `smeterOffset` — `audio.smeter_offset`
- `mobile` — μικρότερα χαρακτηριστικά canvas

**ΣΗΜΕΙΩΣΗ:** αυτό το όργανο τροφοδοτείται σκόπιμα από την ΑΚΑΤΕΡΓΑΣΤΗ ισχύ συν το offset, ΟΧΙ από τη βαθμονομημένη τιμή dBm που χρησιμοποιεί η αναλογική όψη. Πάντα είχε τη δική του αντιστοίχιση. Δίνοντάς του τη βαθμονομημένη τιμή θα μετατοπίζονταν όλες οι ενδείξεις σε κάθε ψηφιακό σταθμό. Μην «ενοποιήσετε» αυτές τις δύο εισόδους.

#### `frontend/src/lib/BandSelector.svelte` — 69 γραμμές

Το πλέγμα κουμπιών των μπαντών. Αποδίδεται δύο φορές στο `App.svelte` (μία ανά διάταξη).

#### `frontend/src/lib/ModesSelector.svelte` — 75 γραμμές

Το πλέγμα κουμπιών των τρόπων λειτουργίας. Αποδίδεται δύο φορές στο `App.svelte`.

#### `frontend/src/lib/StatusIndicators.svelte` — 122 γραμμές

Οι λυχνίες κατάστασης (mute, squelch, NR, NB, NS, AN, CTCSS). Δέχεται ένα `prop wide`: η ψηφιακή διάταξη χρησιμοποιεί πλατύτερες λυχνίες από την αναλογική. Και οι δύο κλάσεις Tailwind (`w-8` και `w-10`) γράφονται ολόκληρες μέσα στο component — ο σαρωτής του Tailwind βλέπει μόνο κυριολεκτικά ονόματα κλάσεων, οπότε ποτέ μην τα φτιάχνετε με συνένωση συμβολοσειρών.

Όλα τα υπόλοιπα αρχεία στο `frontend/src/lib/` (Spectrogram, PassbandTuner, FrequencyInput, VersionSelector, ...) προϋπήρχαν αυτής της εργασίας και είναι αμετάβλητα.

### 3. Αλλαγή που επηρεάζει μόνο ορισμένες παραλλαγές

Τυλίξτε την στη σημαία. Στο markup:

```
{#if isAnalog}
 ...μόνο οι αναλογικοί σταθμοί το βλέπουν αυτό...
{:else}
 ...μόνο οι ψηφιακοί σταθμοί το βλέπουν αυτό...
{/if}

{#if isV2} ... {/if} {#if !isV2} ... {/if}
```

Σε ένα χαρακτηριστικό class:

```
class="p-4 {isAnalog ? 'text-xs' : 'text-sm'} rounded-md"
```

Γράψτε και τις δύο εναλλακτικές ως πλήρη κυριολεκτικά ονόματα, όπως παραπάνω. Το Tailwind σαρώνει τον πηγαίο κώδικα για πλήρη ονόματα κλάσεων· το `text-{size}` ή ένα συνενωμένο όνομα αφαιρείται από το CSS και το στυλ εξαφανίζεται σιωπηλά.

Στο μπλοκ `<script>`: προσπαθήστε να ΜΗΝ διακλαδίζετε καθόλου. Το σενάριο υπολογίζει σκόπιμα τις εισόδους ΚΑΙ ΤΩΝ ΔΥΟ οργάνων σε κάθε κύκλο —

```
smeterDbm = powerDb; // αναλογική όψη
smeterRawDb = audio.getPowerDb(); // ψηφιακή όψη
smeterOffset = audio.smeter_offset; // ψηφιακή όψη
```

— ώστε μόνο το markup να χρειάζεται να επιλέξει. Γι' αυτό υπάρχουν μόλις 15 διακλαδώσεις παραλλαγών σε ένα αρχείο 17000 γραμμών. Κρατήστε το έτσι· η διακλάδωση σε επίπεδο script ήταν αυτό που έκανε τα τέσσερα παλιά αρχεία να αποκλίνουν.

## 4. Πού βρίσκονται οι διακλαδώσεις παραλλαγών

Δεκαπέντε σημεία στο App.svelte. Οι αριθμοί γραμμών μετατοπίζονται όταν επεξεργάζεστε — ο αξιόπιστος τρόπος να τα βρείτε όλα είναι:

```
cd frontend/src && grep -n 'isAnalog\\|isV2' App.svelte
```

Κατά τη σύνταξη του κειμένου:

Γραμμή	Σημαία	Τι εναλλάσσει
20		δήλωση του isAnalog
21		δήλωση του isV2
8062	isV2	υπολογιστής: επιλογέας μπάντας πρώτος (σειρά v2)
8090	isV2	υπολογιστής: μέγεθος επικεφαλίδας του πάνελ
8686	isV2	υπολογιστής: επιλογέας τρόπων λειτουργίας πρώτος (σειρά v1)
8717	isAnalog	ελάχιστο πλάτος του πάνελ του οργάνου (η βελόνα είναι στενότερη)
8730	isAnalog	μόνο αναλογικό: γραμμή ημερομηνίας/ώρας πάνω από το όργανο
8819	isAnalog	μόνο αναλογικό: λυχνίες κατάστασης + διαχωριστικό
8834	isAnalog	μόνο ψηφιακό: γραμμή ώρας, πλατιές λυχνίες, και η ίδια η επιλογή <SMeterDigital> / <SMeterAnalog>
8861	isAnalog	κατακόρυφες αποστάσεις του μπλοκ συχνότητας
8864	isAnalog	άνω περιθώριο του μπλοκ λεπτού συντονισμού
12288	isV2	κινητό: σειρά λεπτού συντονισμού επάνω (σειρά v2)
12347	isAnalog	κινητό: ποιο όργανο αποδίδεται
12433	isV2	κινητό: σειρά λεπτού συντονισμού κάτω (σειρά v1)
13204	isAnalog	μέγεθος κειμένου του κουμπιού noise gate

Όλα τα υπόλοιπα στο αρχείο είναι κοινά και για τις τέσσερις παραλλαγές.

## 5. Άλλα ρυθμιστικά που αξίζει να γνωρίζετε

Θέση	Σταθερά	Σκοπός
App.svelte γραμμή 5034	<code>const visualGain = 1.1;</code>	Βαθμονόμηση του αναλογικού οργάνου. Κάθε 0,10 αντιστοιχεί περίπου σε 5 dBm στην ένδειξη.
SMeterAnalog.svelte γραμμή 55	<code>const SMOOTH_TIME_MS = 16;</code>	Σταθερά χρόνου εξομάλυνσης της βελόνας, σε χιλιοστά του δευτερολέπτου.
SMeterAnalog.svelte γραμμή 64	<code>let smeterTheme = 'dark';</code>	Προεπιλεγμένη όψη του αναλογικού οργάνου για φυλλομετρητή χωρίς αποθηκευμένη επιλογή. Ορίστε την με <code>./smeter_theme.sh</code> , όχι με το χέρι.
SMeterAnalog.svelte γραμμή 69	<code>const SMETER_PREF_VERSION = 2;</code>	Μετρητής εφάπαξ επαναφοράς της αποθηκευμένης όψης. Αυξήστε τον για να μετακινήσετε υπάρχοντες επισκέπτες στην προεπιλογή· πηγαίνει μόνο προς τα πάνω.
SMeterDigital.svelte γραμμή 149	<code>const DIGITAL_BAR_TRIM = 0;</code>	Μετατοπίζει τη ράβδο κατά ακέραια τμήματα.
SMeterDigital.svelte γραμμή 31	<code>const numberOfDots = 35;</code>	Αριθμός τμημάτων της ράβδου.

## 6. Μετά την επεξεργασία — ανακατασκευή

Ο ευκολότερος τρόπος, από τη ρίζα του αποθετηρίου:

```
./recompile.sh
```

Η επιλογή μενού **[2]** ξαναχτίζει το frontend και ρωτά ποια παραλλαγή θα βλέπει ένας επισκέπτης την πρώτη φορά. Η επιλογή καταγράφεται στο `frontend/variant.json` — κανένα αρχείο πηγαίου κώδικα δεν αντιγράφεται ούτε ξαναγράφεται. (Παλιότερα ξαναδημιουργούσε και το `VersionSelector.svelte` από heredoc· αυτό αφαιρέθηκε όταν ο επιλογέας έπαψε να πλοηγεί, γιατί θα ανέτρεπε την εναλλαγή σε χρόνο εκτέλεσης σε κάθε εκτέλεση.) Η επιλογή **[1]** αφορά μόνο το backend, η **[3]** και τα δύο.

Ή απευθείας, από το `frontend/`:

```
./build-all.sh χτίζει τη σελίδα για υπολογιστή KAI το /mobile/ <-- συνήθως αυτό
./build-default.sh μόνο τη σελίδα για υπολογιστή
./build-mobile.sh μόνο το /mobile/
```

Το `build-default.sh` αφήνει το vite να αδειάσει τον `dist/`, πράγμα που αφαιρεί τον `dist/mobile/` μέχρι το επόμενο `build-all.sh` ή `build-mobile.sh`. Γι' αυτό το `build-all.sh` είναι η κανονική επιλογή.

Το `build-all.sh` έχει το δικό του αντίγραφο της λογικής δόμησης (`build_version SMETER LAYOUT NAME OUTDIR BASE LOG`) αντί να καλεί το `build-default.sh`. Αν αλλάξετε τον τρόπο που χτίζεται η σελίδα, αλλάξτε τον KAI ΣΤΑ ΔΥΟ σημεία, αλλιώς οι δύο διαδρομές θα διαφωνήσουν.

Μερικές λεπτομέρειες που αξίζει να ξέρετε:

- Θέτει `PHANTOM_KEEP_OUTDIR=1` ώστε το Vite να μην αδειάζει τον κατάλογο εξόδου, και καθαρίζει το ίδιο το `dist/` — σβήνει τα περιεχόμενα αλλά κρατά το `users.json`, το οποίο ο ζωντανός `spectrumserver` ξαναγράφει σε κάθε σύνδεση και αποσύνδεση ακροατή.

- Χτίζει το `/mobile` τελευταίο, αφού ολοκληρωθεί η έξοδος για υπολογιστή.
- Ο μηχανισμός παράλληλης δόμησης (`PHANTOM_BUILD_JOBS`, προεπιλογή 3) είναι κατάλοιπο από την εποχή των πέντε δομήσεων για υπολογιστή. Με μία, τίποτα δεν περιμένει στην ουρά.

**Δικαιώματα εκτέλεσης:** το `recompile.sh` εκτελεί `chmod +x` σε ένα σενάριο πριν το τρέξει, και το `build-all.sh` καλεί σκόπιμα `bash build-mobile.sh` αντί για `./build-mobile.sh`, οπότε ένα bit που λείπει δεν χαλάει πια τη δόμηση. Η μεταφόρτωση ενός σεναρίου μέσω της διεπαφής ιστού του GitHub το επαναφέρει πάντως σε 644, οπότε αν τρέξετε κάποιο απευθείας:

```
chmod +x frontend/build-*.sh
```

## 7. Πώς να ελέγξετε την αλλαγή σας πριν την εμπιστευτείτε

Ένα πράσινο `vite build` **ΔΕΝ** αρκεί. Η δόμηση δεν μπορεί να δει ένα αναγνωριστικό που χάθηκε όταν μετακινήθηκε κώδικας μεταξύ αρχείων — αυτό είναι `ReferenceError` κατά την εκτέλεση, και η σελίδα απλώς δεν φορτώνει. Έχει συμβεί δύο φορές.

### 1) Έλεγχος μη ορισμένων αναγνωριστικών (πιάνει ακριβώς αυτή την αστοχία)

```
cd frontend
npx eslint --no-eslintrc \
 --parser svelte-eslint-parser \
 --rule '{"no-undef":"error"}' \
 --env browser,es2022 \
 src/App.svelte src/lib/SMeter*.svelte
```

Ένα γνωστό, αθώο εύρημα αναμένεται:

```
'dBmCalOffset' is not defined
```

Το `typeof x === 'number'` σε μη δηλωμένο όνομα δεν προκαλεί ποτέ σφάλμα, οπότε η έκφραση ήδη αποτιμάται σε 0. Οτιδήποτε άλλο είναι πραγματικό πρόβλημα.

### 2) Χτίστε, μετά ΦΟΡΤΩΣΤΕ ΤΗ ΣΕΛΙΔΑ ΚΑΙ ΠΕΡΑΣΤΕ ΚΑΙ ΤΙΣ ΤΕΣΣΕΡΙΣ ΠΑΡΑΛΛΑΓΕΣ

Όχι μόνο αυτή που αλλάξατε — μια αλλαγή κάτω από το `isAnalog/isV2` μπορεί να χαλάσει τις άλλες τρεις ενώ η δική σας δουλεύει. Ανοίξτε το `/`, περάστε και τις τέσσερις επιλογές στο μενού ⚙, και μετά ανοίξτε το `/mobile/`.

Ανοίξτε την κονσόλα του φυλλομετρητή σε καθεμία. Μια κενή ή μισοσχεδιασμένη σελίδα με `ReferenceError` στην κονσόλα είναι η υπογραφή του παραπάνω σφάλματος.

### 3) Ποτέ μην αφήνετε το `frontend/dist/` σε ημιτελή κατάσταση

Ένα σκέτο `npx vite build` αδειάζει τον κατάλογο εξόδου και παίρνει μαζί του το `dist/mobile/` και το ζωντανό `dist/users.json` του διακομιστή — η σελίδα για κινητά απαντά 404 μέχρι την επόμενη επιτυχημένη δόμηση. Χτίζετε πάντα μέσω των σεναρίων: το `build-all.sh` θέτει `PHANTOM_KEEP_OUTDIR=1` και καθαρίζει μόνο του, διατηρώντας το `dist/users.json`. Αν έχετε αμφιβολία, απλώς ξανατρέξτε το `./build-all.sh`.

## ΚΕΦΑΛΑΙΟ 12

# Οδηγός χρήσης

USER\_GUIDE.md

## Οδηγός χρήσης PhantomSDR-Plus

Καλώς ορίσατε στο PhantomSDR-Plus! Ο οδηγός αυτός θα σας βοηθήσει να αξιοποιήσετε στο έπακρο την εμπειρία ακρόασης μέσω WebSDR.

### Πίνακας περιεχομένων

1. Εισαγωγή
2. Ξεκινώντας
3. Επισκόπηση της διεπαφής
4. Βασικές λειτουργίες
5. Προηγμένες δυνατότητες
6. Τρόποι αποδιαμόρφωσης
7. Ψηφιακοί αποκωδικοποιητές
8. Συντομεύσεις πληκτρολογίου
9. Σελιδοδείκτες
10. Χρήση σε κινητά
11. Συμβουλές και καλές πρακτικές
12. Αντιμετώπιση προβλημάτων
13. Συχνές ερωτήσεις

### Εισαγωγή

#### Τι είναι το PhantomSDR-Plus;

Το PhantomSDR-Plus είναι ένα ραδιόφωνο καθορισμένο από λογισμικό (SDR) που λειτουργεί μέσω web και σας επιτρέπει να ακούτε ραδιοφωνικά σήματα μέσω του διαδικτύου. Δεν απαιτείται ειδικό λογισμικό ή υλικό από τη δική σας πλευρά — μόνο ένας σύγχρονος περιηγητής!

#### Τι μπορείτε να ακούσετε;

Ανάλογα με τη διαμόρφωση του WebSDR, μπορείτε να συντονιστείτε σε:

- **Ραδιοερασιτεχνικά:** ραδιοερασιτέχνες σε όλο τον κόσμο
- **Ραδιοφωνικούς σταθμούς:** ραδιόφωνο AM/FM, εκπομπές βραχέων κυμάτων
- **Αεροπορία:** έλεγχος εναέριας κυκλοφορίας, επικοινωνίες αεροσκαφών
- **Ναυτιλία:** επικοινωνίες πλοίου-ακτής, ναυτιλιακά δελτία καιρού
- **Μετεωρολογικούς δορυφόρους:** NOAA, METEOR-M
- **Ψηφιακούς τρόπους:** FT8, RTTY, PSK31 και άλλους
- **Σταθμούς υπηρεσιών:** σήματα ώρας, στρατιωτικοί, κυβερνητικοί

## Απαιτήσεις συστήματος

- **Περιηγητής:** Chrome/Edge (συνιστάται), Firefox, Safari
  - **Σύνδεση:** ευρυζωνικό διαδίκτυο (συνιστάται 1+ Mbps)
  - **Ήχος:** ηχεία ή ακουστικά που λειτουργούν
  - **Προαιρετικά:** ποντίκι με ροδέλα για ευκολότερο συντονισμό
- 

## Ξεκινώντας

---

### 1. Πρόσβαση στο WebSDR

Ανοίξτε τον περιηγητή σας και μεταβείτε στη διεύθυνση του WebSDR που σας έδωσε ο διαχειριστής.

Παράδειγμα: `http://websdr.example.com:9002`

### 2. Αρχική φόρτωση της σελίδας

Όταν φορτώσει η σελίδα, θα δείτε:

- μια πολύχρωμη οθόνη καταρράκτη που δείχνει τη ραδιοφωνική δραστηριότητα
- πίνακα ελέγχου με ένδειξη συχνότητας και κουμπιά
- S-meter που δείχνει την ισχύ του σήματος
- ένδειξη αριθμού χρηστών

### 3. Ξεκινήστε την ακρόαση

1. **Κάντε κλικ σε ένα σήμα** στην οθόνη καταρράκτη
  2. **Ο ήχος θα ξεκινήσει αυτόματα**
  3. **Ρυθμίστε την ένταση** με τον έλεγχο έντασης του περιηγητή ή τον ρυθμιστή στην οθόνη
  4. **Κάντε λεπτό συντονισμό** κάνοντας κλικ με ακρίβεια πάνω στο σήμα
-

## Επισκόπηση της διεπαφής

### Κύρια στοιχεία

WebSDR <callsign>, located in <QTH> ⓘ [ Analog S-Meter ▼ ] email · Keyboard Shortcuts · Users · Stats · Other servers 1 2 3 Frequency Search: MW SW HamDash PSK Reporter WSPRnet · Callsign Search [ Open Additional Info ]			
Spectrum + Waterfall band-plan strip along the bottom			
Audio & Buffer volume · SQ · zoom AGC Auto Fast Med Slow Compressor Equalizer	Frequency 7,120.00 band · VFO	S-Meter dBm dBμV SNR · NF	Waterfall Controls Min · Max · zoom colour-map strip Zoom In Out Max Min
Filters NR NB NS AN CTCSS Modes selector ... RADEL RADEU USB LSB CW AM QUAM FM Band selector	Fine Tuning (kHz) VFO A · Modes · Bands · IF Filters Wheel Tuning Steps Decoders FT8 FT4 FT2 JS8 CW WSPR FAX SSTV NAVTEX RTTY ...	Waterfall Spectrum Auto Adj. Height [ profile ▼ ] [ Bookmarks ]	

Τα τρία πάνελ κάτω από τον καταρράκτη κυλούν ως μία σελίδα· σε στενό παράθυρο στοιβάζονται αντί να μπαίνουν δίπλα-δίπλα. Τα **RADEL** και **RADEU** βρίσκονται πάνω στην ίδια τη γραμμή του τίτλου *Modes selector*, στα δεξιά του.

### 1. Οθόνη καταρράκτη

Ο καταρράκτης είναι μια οπτική αναπαράσταση των ραδιοφωνικών σημάτων:

- **Οριζόντιος άξονας:** η συχνότητα
- **Κατακόρυφος άξονας:** ο χρόνος (κυλά προς τα κάτω)
- **Χρώματα:** η ισχύς του σήματος
- **Σκούρο μπλε/μαύρο:** κανένα σήμα (στάθμη θορύβου)
- **Πράσινο/κίτρινο:** ασθενή έως μέτρια σήματα
- **Πορτοκαλί/κόκκινο:** ισχυρά σήματα
- **Λευκό:** πολύ ισχυρά σήματα

### 2. Ένδειξη συχνότητας

Δείχνει την τρέχουσα συχνότητα συντονισμού σε διάφορες μορφές:

- **MHz:** 7.100.000 MHz (μπάντες HF)
- **kHz:** 14200 kHz
- **Hz:** 145500000 Hz (VHF/UHF)

### 3. S-meter (αναλογικό ή ψηφιακό)

Εμφανίζει την ισχύ του σήματος:

- **S1-S9:** τυπική κλίμακα ισχύος σήματος
- **+dB:** σήματα πάνω από S9 (π.χ. S9+20dB)
- **Αναλογικό ή ψηφιακό:** ανάλογα με τη διαμόρφωση

## 4. Κουμπιά τρόπου λειτουργίας

Επιλέξτε τον τρόπο αποδιαμόρφωσης:

- **AM**: διαμόρφωση πλάτους
- **FM**: διαμόρφωση συχνότητας
- **USB**: άνω πλευρική ζώνη
- **LSB**: κάτω πλευρική ζώνη
- **CW**: κώδικας Μορς (συνεχές κύμα)
- **WBFM**: ευρυζωνικό FM (ραδιοφωνικές εκπομπές)
- **QUAM**: στερεοφωνικό AM C-QUAM — η ετικέτα γίνεται πράσινη όταν ανιχνευθεί στερεοφωνικός πιλότος

Δίπλα στην επικεφαλίδα **Modes selector** βρίσκονται τα κουμπιά **RADEL** και **RADEU**, που ξεκινούν την ψηφιακή φωνή RADE v1 με ένα πάτημα — δείτε [Ψηφιακοί αποκωδικοποιητές](#).

## 5. Πίνακας ελέγχου

Πρόσθετα χειριστήρια:

- **AGC**: αυτόματος έλεγχος κέρδους
- **NR**: μείωση θορύβου (φασματική)
- **NB**: κόπτης κρουστικού θορύβου
- **NS**: καταστολή θορύβου βάθους
- **AN**: αυτόματο notch
- **CTCSS**: squelch υποτόνου (FM). Πατώντας το ανοίγει δίπλα στο κουμπί ένα μικρό παράθυρο με *Any tone* και τους 49 τυποποιημένους υποτόνους (67,0–254,1 Hz)· διαλέγετε έναν και το κουμπί τον εμφανίζει (π.χ. 88.5, ή ANY), και ο ήχος περνά μόνο όσο λαμβάνεται αυτός ο υπότονος — το άνοιγμα παίρνει περίπου 0,7 s. Πατώντας ξανά το κουμπί το CTCSS απενεργοποιείται· το Escape ή ένα κλικ έξω κλείνει το παράθυρο χωρίς αλλαγή. Οι σελιδοδείκτες κρατούν τον επιλεγμένο τόνο.
- **SQL**: squelch
- **AI**: μείωση θορύβου με τεχνητή νοημοσύνη (μόνο φωνή)
- **Zoom**: επίπεδο μεγέθυνσης του καταρράκτη
- **Wheel Tuning Steps**: βήμα συντονισμού της ροδέλας του ποντικιού
- **Decoders**: κουμπιά που ξεκινούν και σταματούν κάθε αποκωδικοποιητή με ένα πάτημα

## 6. Επικάλυψη σχεδίου μπαντών

Χρωματιστές λωρίδες που δείχνουν τις κατανομές συχνοτήτων:

- διαφορετικά χρώματα για διαφορετικές υπηρεσίες
- βοηθούν να διακρίνετε τι επιτρέπεται σε κάθε συχνότητα

## Βασικές λειτουργίες

### Συντονισμός σε μια συχνότητα

#### Μέθοδος 1: κλικ στον καταρράκτη

1. Κάντε κλικ απευθείας σε ένα σήμα στον καταρράκτη
2. Ο δέκτης θα συντονιστεί σε αυτή τη συχνότητα
3. Ο ήχος θα ξεκινήσει

## Μέθοδος 2: πληκτρολόγηση συχνότητας

1. Κάντε κλικ στην ένδειξη συχνότητας
2. Πληκτρολογήστε την επιθυμητή συχνότητα
3. Πατήστε Enter

Παραδείγματα:

- 7100 → 7.100 MHz
- 14200.0 → 14.200 MHz
- 145.500 → 145.500 MHz

## Μέθοδος 3: χρήση της ροδέλας του ποντικιού

1. Τοποθετήστε τον δείκτη πάνω από την ένδειξη συχνότητας
2. Ροδέλα προς τα πάνω για αύξηση της συχνότητας
3. Ροδέλα προς τα κάτω για μείωση της συχνότητας

## Μέθοδος 4: κουμπιά βήματος

1. Χρησιμοποιήστε τα κουμπιά ▲ και ▼ δίπλα στη συχνότητα
2. Το βήμα διαφέρει ανά τρόπο λειτουργίας:
3. **AM/FM**: βήματα του 1 kHz
4. **SSB**: βήματα των 100 Hz
5. **CW**: βήματα των 10 Hz

## Σαρωτής (Scanner)

Ο σαρωτής μετακινεί τον δέκτη κατά μήκος μιας περιοχής καναλιών και σταματά στο πρώτο που έχει σήμα. Βρίσκεται στη γραμμή **Fine Tuning (kHz)**, τέρμα δεξιά:



Χειριστήριο	Τι κάνει
◀ ▶	Σάρωση κάτω / πάνω. Όταν έχει σταματήσει σε σήμα, ένα βέλος συνεχίζει τη σάρωση.
■	Σταμάτημα.
🔍	Αποκλεισμός αυτού του καναλιού — για παράσιτο ή μόνιμο φέρον.
30 dB · 4 ▼	Το κατώφλι, και δίπλα η στάθμη που μετριέται τώρα. Ανοίγει τις ρυθμίσεις.

Το κείμενο αριστερά δείχνει τι κάνει ο σαρωτής: Scanner όταν είναι ανενεργός, τη συχνότητα όπου έχει φτάσει κατά τη σάρωση, ● 3s όσο μετράει αντίστροφα για να συνεχίσει, ● 30·22s όσο το κανάλι είναι ακόμα κατειλημμένο, ● hold όταν θα μείνει εκεί.

## Ρυθμίσεις

- **Range** — το *Scan Band* σαρώνει την μπάντα όπου ξεκινά η σάρωση, το *Scan Visible* σαρώνει ακριβώς ό,τι δείχνει ο καταρράκτης και τον ακολουθεί όταν κάνετε ζουμ ή τον μετακινείτε.
- **Scan** — το *Every channel* συντονίζει και ακούει κάθε κανάλι με τη σειρά, το *Skip empty* διαβάζει το φάσμα και πηγαίνει κατευθείαν στα σήματα.
- **Stop at** — πόσο πάνω από το θόρυβο βάθους της μπάντας πρέπει να είναι ένα κανάλι για να σταματήσει η σάρωση, σε dB. Το θόρυβο παρακολουθείται συνεχώς, ώστε η ίδια ρύθμιση δουλεύει μέρα και νύχτα. Ένα άδειο κανάλι δεν δείχνει 0 dB — δείτε για μια στιγμή την ένδειξη στο κουμπί και βάλτε το κατώφλι πιο πάνω.

- **Resume after** — πόση ώρα πρέπει να μένει ήσυχο ένα κανάλι πριν συνεχίσει η σάρωση. Οι παύσεις της ομιλίας δεν την ξεκινούν ξανά. Το *Hold* μένει μέχρι να πατήσετε κουμπί.
- **Max stay** — συνεχίζει μετά από αυτόν τον χρόνο ακόμα και αν το σήμα υπάρχει ακόμα, ώστε ένα μόνιμο φέρον να μην κρατάει τη σάρωση για πάντα.

Το βήμα ακολουθεί τον τρόπο λειτουργίας — 1 kHz στο SSB, 0,1 kHz στο CW, 5 kHz στο AM, 9 ή 10 kHz στα μεσαία και 9 kHz στα μακρά — και οι στάσεις πέφτουν πάνω στο πλέγμα των καναλιών. Η σάρωση μένει μέσα στην περιοχή της και μέσα σε όσα μπορεί να συντονίσει ο δέκτης· στην άκρη συνεχίζει από το άλλο άκρο.

Οι ρυθμίσεις και τα αποκλεισμένα κανάλια αποθηκεύονται στον φυλλομετρητή σας.

## Επιλογή τρόπου αποδιαμόρφωσης

Επιλέξτε τον κατάλληλο τρόπο για το σήμα:

### Για φωνητικές επικοινωνίες:

- **AM**: αεροπορία, ραδιοφωνικές εκπομπές AM, ορισμένες ερασιτεχνικές
- **FM**: αναμεταδότες VHF/UHF, ραδιοφωνικές εκπομπές FM
- **USB**: ραδιοερασιτεχνικά HF (20 m, 17 m, 15 m, 12 m, 10 m)
- **LSB**: ραδιοερασιτεχνικά HF (160 m, 80 m, 40 m, 30 m)

### Για δεδομένα/ψηφιακά:

- **USB**: οι περισσότεροι ψηφιακοί τρόποι (FT8, PSK31, RTTY)
- **LSB**: ορισμένοι ψηφιακοί τρόποι στις χαμηλότερες μπάντες HF

### Για κώδικα Μορς:

- **CW**: σήματα τηλεγραφίας/Μορς

## Ρύθμιση της έντασης

- **Ρυθμιστής στην οθόνη**: σύρετε τον ρυθμιστή έντασης
- **Ένταση περιηγητή**: χρησιμοποιήστε τα χειριστήρια πολυμέσων του περιηγητή
- **Ένταση συστήματος**: ρυθμίστε την ένταση του υπολογιστή σας
- **Πληκτρολόγιο**: χρησιμοποιήστε τα πλήκτρα + και - (αν υποστηρίζονται)

## Χρήση του S-meter

Το S-meter δείχνει την ισχύ του σήματος:

- **S0-S3**: πολύ ασθενές σήμα, δύσκολο στη λήψη
- **S4-S6**: ασθενές έως μέτριο σήμα
- **S7-S9**: καλό έως ισχυρό σήμα
- **S9+**: εξαιρετικά ισχυρό σήμα

**Συμβουλή**: για καλύτερο ήχο, συντονιστείτε σε σήματα που δείχνουν S7 ή περισσότερο.

**Αλλαγή της όψης του οργάνου**: στο αναλογικό όργανο (με βελόνα), κάντε κλικ πάνω στο ίδιο το όργανο — ή εστιάστε το και πατήστε Enter ή Space — για να εναλλάξετε τρία φόντα: σκούρο βουρτσισμένο μέταλλο, μια ανοιχτή απαλή γκριζα όψη και μια ζεστή vintage κεχριμπαρένια όψη. Ο φυλλομετρητής σας θυμάται την επιλογή, οπότε παραμένει μετά από ανανέωση ή επανεκκίνηση. Ισχύει ανά φυλλομετρητή και ανά διεύθυνση: αν ανοίξετε τον δέκτη με το όνομα υπολογιστή και με την IP, θα έχετε δύο ξεχωριστές ρυθμίσεις, ενώ ένα ιδιωτικό παράθυρο ξεκινά πάντα από την προεπιλογή του σταθμού.

**Πάνω από τα 30 MHz** το S-meter ακολουθεί το πρότυπο VHF της IARU Region 1: εκεί το S9 είναι στα  $-93$  dBm αντί για  $-73$  dBm στα βραχέα, πάλι με έξι dB ανά μονάδα S. Η βελόνα και η μπάρα αλλάζουν μόνες τους με τη συχνότητα συντονισμού· οι τιμές σε dBm και dBμV δείχνουν πάντα τη μετρημένη στάθμη. Ορισμένοι σταθμοί ρυθμίζουν επίσης τους δέκτες VHF/UHF ώστε τα όργανα να μένουν στο **0 σε άδειο κανάλι**, όπως σε πομποδέκτη VHF/UHF: πάνω από τα 60 MHz η βελόνα και η μπάρα ανεβαίνουν τότε μόνο όταν ένα σήμα ξεχωρίζει από τον θόρυβο. Οι τιμές dBm, dBμV, SNR και NF δείχνουν πάντα τις πραγματικές τιμές.

## Επιλογή δέκτη

Ορισμένοι σταθμοί τρέχουν περισσότερους από έναν δέκτες — για παράδειγμα έναν για τα βραχέα και έναν για τη μπάντα των 2 m. Δείχνουν στην κεφαλίδα της σελίδας μια γραμμή *Receivers*: με ένα κουμπί ανά δέκτη· αυτός στον οποίο βρίσκεστε εμφανίζεται κίτρινος. Πατώντας άλλο κουμπί ανοίγει η σελίδα εκείνου του δέκτη, με δικό του καταρράκτη, μπάντες, chat και λίστα ακροατών. Στη σελίδα /mobile τα ίδια κουμπιά βρίσκονται στην επάνω μπάρα. Ένας σταθμός με έναν μόνο δέκτη δεν δείχνει τέτοια γραμμή.

## Προηγμένες δυνατότητες

### Αυτόματος έλεγχος κέρδους (AGC)

Το AGC ρυθμίζει αυτόματα τις στάθμες ήχου:

- **Off**: καμία αυτόματη ρύθμιση κέρδους
- **Slow**: σταδιακές μεταβολές στάθμης (ιδανικό για SSB)
- **Medium**: ισορροπημένη απόκριση
- **Fast**: γρήγορη προσαρμογή (ιδανικό για AM)

**Σύσταση**: ξεκινήστε με «Fast» για AM και «Slow» για SSB.

### Τα τέσσερα χειριστήρια θορύβου

Τα NR, NB, NS και AN είναι ξεχωριστά κουμπιά on/off, καθένα για διαφορετικό είδος θορύβου. Είναι ανεξάρτητα μεταξύ τους — ανάβοντας το ένα δεν ανάβει άλλο — και συνδυάζονται ελεύθερα.

Κανένα τους δεν φτάνει στους αποκωδικοποιητές: τα FT8, CW, WSPR, SSTV, FAX, NAVTEX, RTTY/PSK31/Olivina και ο QRSS grabber διαβάζουν τον ήχο πριν από αυτά τα φίλτρα, οπότε μπορείτε να τα ρυθμίζετε καθαρά με το αυτί χωρίς να επηρεάζετε την αποκωδικοποίηση. Δείτε το [εγχειρίδιο αποκωδικοποιητών](#).

### Μείωση θορύβου (NR)

Φασματική μείωση θορύβου. Εκτιμά τη στάθμη θορύβου σε κάθε περιοχή του φάσματος ήχου και χαμηλώνει αυτές τις περιοχές, αφήνοντας ήσυχο ό,τι ξεχωρίζει πάνω από τον θόρυβο.

**Χρησιμοποιήστε το όταν**: ακούτε σταθερό σφύριγμα ή λευκό θόρυβο πίσω από το σήμα.

Οι σταθεροί τόνοι — μια νότα CW, ένα φέρον — αναγνωρίζονται ως σήμα και προστατεύονται, οπότε το NR δεν τρώει το σήμα CW όπως θα έκανε ένα αφελές φίλτρο. Τυπικό αποτέλεσμα: 10 dB πτώση του θορύβου ανάμεσα στα λόγια, με κόστος λίγα δέκατα του dB από την ίδια την ομιλία.

### Κόπτης κρουστικού θορύβου (NB)

Αφαιρεί τον κρουστικό θόρυβο: κλικ, κρότους, ατμοσφαιρικά, θόρυβο ανάφλεξης και γραμμών ρεύματος. Παρακολουθεί την περιβάλλουσα του ήχου και σβήνει μόνο τα δείγματα που εκτινάσσονται πολύ πάνω από αυτήν — περίπου ένα χιλιοστό του δευτερολέπτου ανά κρότο, με ομαλή είσοδο και έξοδο ώστε το ίδιο το σβήσιμο να μην κάνει κλικ.

**Χρησιμοποιήστε το όταν:** ακούτε κλικ ή κρότους από γραμμές ρεύματος, μοτέρ, καταιγίδες ή ανάφλεξη οχημάτων.

Τα notch δικτύου στα 50 Hz και 60 Hz ακολουθούν αυτό το κουμπί.

## Καταστολή θορύβου βάθους (NS)

Μετρά για αρκετά δευτερόλεπτα το ίδιο το πάτωμα θορύβου της μπάντας και κόβει σταθερά ό,τι κάθεται σε αυτό, βαθύτερα στις ψηλές συχνότητες ήχου παρά στις χαμηλές. Εκεί που το NR αντιδρά στιγμή προς στιγμή, το NS είναι το αργό, σταθερό χέρι: χαμηλώνει τη βουή της μπάντας χωρίς να αλλάζει τον ήχο του ίδιου του σήματος.

**Χρησιμοποιήστε το όταν:** η μπάντα είναι ήσυχη αλλά βουερή και θέλετε να πέσει ο θόρυβος χωρίς την «υποβρύχια» χροιά του επιθετικού NR.

Ενεργό μόνο σε USB, LSB και AM — CW, FM και ψηφιακοί τρόποι μένουν ανέγγιχτοι. Ξαναμετρά το πάτωμα σε κάθε αλλαγή συχνότητας ή τρόπου και σταθεροποιείται μέσα σε λίγα δευτερόλεπτα.

## Αυτόματο notch (AN)

Βρίσκει και αφαιρεί μόνο του σταθερούς ενοχλητικούς τόνους — ετεροδύνες, φέροντα, σφυρίγματα — χωρίς να χρειάζεται να τοποθετήσετε notch με το χέρι. Προσαρμόζεται συνεχώς, οπότε αφαιρεί πολλούς τόνους ταυτόχρονα και κρατά σβηστό έναν τόνο που ολισθαίνει.

**Χρησιμοποιήστε το όταν:** ακούτε σφύριγμα ή τόνο πάνω από το σήμα που θέλετε.

Παρακάμπτεται στο CW, όπου το επιθυμητό σήμα είναι σταθερός τόνος.

**Σημείωση καθυστέρησης:** τα NR και NS προσθέτουν το καθένα περίπου 40 ms καθυστέρηση ήχου όσο είναι αναμμένα (και τα δύο μαζί, περίπου 80 ms). Τα NB και AN δεν προσθέτουν καθόλου. Αυτό αφορά μόνο την ακρόαση, ποτέ τους αποκωδικοποιητές.

## Αυτόματο squelch (SQL)

Σιγάζει τον ήχο όταν δεν υπάρχει σήμα:

- **Off:** πάντα ανοιχτό (ακούγεται στατικός θόρυβος)
- **Auto:** ορίζει αυτόματα το κατώφλι
- **Manual:** χειροκίνητη ρύθμιση κατωφλίου

**Χρησιμοποιήστε το όταν:** παρακολουθείτε μια συχνότητα για δραστηριότητα.

## Μείωση θορύβου με τεχνητή νοημοσύνη (AI)

Το κουμπί **AI**, με ένα ρυθμιστικό έντασης δίπλα του, βρίσκεται στο πλαίσιο Audio & Buffer ακριβώς κάτω από το SQ. Στη σελίδα /mobile βρίσκεται στην καρτέλα Audio, κάτω από το Squelch. Αφαιρεί τον θόρυβο της μπάντας από την ομιλία με το RNNoise, ένα μικρό νευρωνικό δίκτυο εκπαιδευμένο σε φωνή.

Όλα γίνονται μέσα στον δικό σας φυλλομετρητή: τίποτα δεν στέλνεται σε εξωτερικό διακομιστή, δεν χρειάζεται λογαριασμός και ο δέκτης δεν επιβαρύνεται καθόλου.

- **Ενεργοποίηση:** Πατήστε **AI**: το κουμπί γίνεται μπλε. Με το πρώτο πάτημα κατεβαίνει η μονάδα (περίπου 1,3 MB) και το κουμπί αναβοσβήνει όσο φορτώνει.
- **Ένταση:** ορίζει πόσο έντονα δουλεύει όσο κάποιος μιλά — οι χαμηλότερες τιμές κρατούν περισσότερο από τον αρχικό ήχο. Το πάνω άκρο της κλίμακας είναι σκόπιμα πιο ήπιο από μια πλήρη αποκοπή, ώστε η ομιλία να μην αλλοιώνεται ούτε στο 100%, που και πάλι καθαρίζει περισσότερο από το 50%. Ανάμεσα στις λέξεις δουλεύει μόνο του πιο έντονα, οπότε ο θόρυβος πέφτει ακόμη περισσότερο εκεί. Η προεπιλογή είναι 50%. Μπορείτε να το ρυθμίσετε και με το AI κλειστό.

- **Ένδειξη κατάστασης:** Η ένδειξη AI κάτω από τη συχνότητα ανάβει κυανή όσο το AI λειτουργεί, και εμφανίζεται μισοσβησμένη όταν το AI είναι ανοιχτό αλλά ο τρόπος λήψης δεν είναι φωνητικός.

**Χρήση όταν:** Ακούτε φωνή SSB ή AM σε μπάνα με θόρυβο. Το σφύριγμα ανάμεσα στις λέξεις πέφτει συνήθως περίπου 10 dB στην προεπιλογή 50%, και περισσότερο στο 100%, έως περίπου 20 dB, ενώ η ομιλία διατηρεί τη στάθμη της.

Μόνο φωνητικοί τρόποι — USB, LSB, AM και SAM. Σε CW, FM, ψηφιακούς τρόπους και C-QUAM ο ήχος περνά ανέγγιχτος, επειδή το δίκτυο θεωρεί θόρυβο έναν τόνο CW ή τη μουσική. Όπως και τα τέσσερα χειριστήρια θορύβου, δεν φτάνει ποτέ στους αποκωδικοποιητές. Σε ασθενείς σταθμούς (κάτω από περίπου +6 dB SNR) μετρά μόνο του τον λόγο σήματος προς θόρυβο και εξομαλύνει περισσότερο τις αποφάσεις του, ώστε η ομιλία να μην παίρνει τον «υδαρή», γουργουριστό ήχο που μπορεί να δώσει ένα τέτοιο φίλτρο· οι ισχυροί σταθμοί επεξεργάζονται ακριβώς όπως πριν. Αν ένα πολύ ασθενές σήμα εξακολουθεί να ακούγεται επεξεργασμένο, χαμηλώστε το ρυθμιστικό. Προσθέτει περίπου 50 ms καθυστέρηση στον ήχο.

## Buffer ήχου και καθυστέρηση

Ο ήχος φτάνει στον φυλλομετρητή σε μικρά πακέτα, 35 ms το καθένα σε έναν τυπικό δέκτη, και ο φυλλομετρητής κρατά λίγο απόθεμα ώστε ένα πακέτο που αργεί να μη δημιουργεί κενό. Αυτό το απόθεμα είναι το **buffer**, και αποτελεί το μεγαλύτερο μέρος της καθυστέρησης ανάμεσα στον δέκτη και το μεγάφωνό σας.

- **Desktop:** στο πάνελ Audio & Buffer, το κουμπί με το εικονίδιο δικτύου ανεβάζει το buffer μία βαθμίδα με κάθε κλικ (γίνεται πράσινο πάνω από την κατώτερη βαθμίδα), και το ρυθμιστικό δίπλα του επιλέγει απευθείας βαθμίδα από 1 έως 6. Η κατώτερη βαθμίδα ταιριάζει σε ενσύρματη σύνδεση· τα κινητά ξεκινούν από τη βαθμίδα 2.
- **/mobile:** το κουμπί **BUF** και το ρυθμιστικό **Buffer** στην καρτέλα Audio κάνουν το ίδιο.

Το buffer ρυθμίζεται μόνο του. Η σελίδα μετρά πόσο ακανόνιστα παραδίδει τον ήχο η σύνδεσή σας και κρατά μόνο όσο απόθεμα έδειξε ότι χρειάζεται το τελευταίο λεπτό, συν το περιθώριο ασφαλείας της βαθμίδας που επιλέξατε: 20 ms στη βαθμίδα 1, 50 ms στη βαθμίδα 2, περισσότερο στις υψηλότερες. Μετά από μια στιγμιαία διαταραχή του δικτύου η επιπλέον καθυστέρηση απορροφάται ξανά μέσα σε ένα-δυο λεπτά, και μια μικρή διαφορά ανάμεσα στο ρολόι της κάρτας ήχου σας και του δέκτη δεν προκαλεί πια διακοπή κάθε λίγα λεπτά. Οι διορθώσεις είναι λίγα χιλιοστά του δευτερολέπτου τη φορά, γίνονται εκεί όπου ο ήχος επαναλαμβάνεται και δεν ακούγονται. Οι αποκωδικοποιητές και οι ηχογραφήσεις σας δεν επηρεάζονται ποτέ: παίρνουν κάθε δείγμα.

Αν ο ήχος διακόπτεται, ανεβάστε το buffer μία βαθμίδα. Σε κινητό μέσω οικιακού Wi-Fi η καθυστέρηση από τον δέκτη ως το μεγάφωνο είναι συνήθως περίπου 100–150 ms· το δίκτυο ανάμεσα σε εσάς και τον δέκτη προστίθεται σε αυτήν.

## Συνομιλία

Όταν ο διαχειριστής έχει ενεργοποιήσει τη συνομιλία, αυτή έχει δικό της πάνελ στη σελίδα desktop και καρτέλα **Chat** στο /mobile. Δώστε το όνομα ή το διακριτικό σας, γράψτε το μήνυμά σας και στείλτε το.

Για να απαντήσετε σε συγκεκριμένο μήνυμα, πατήστε ➔ **Reply** δίπλα στην ώρα του. Μια γραμμή πάνω από το πεδίο εισαγωγής δείχνει σε ποιον και σε τι απαντάτε· γράψτε και στείλτε όπως πάντα, ή πατήστε × ή Esc για ακύρωση. Η απάντηση εμφανίζεται με εσοχή κάτω από το μήνυμα στο οποίο απαντά, με μια σύντομη γραμμή παράθεσης. Μια απάντηση σε απάντηση μπαίνει στο ίδιο νήμα. Ο διακομιστής κρατά τα τελευταία 20 μηνύματα· όταν το αρχικό έχει ήδη φύγει από αυτό το ιστορικό, η απάντηση δείχνει μόνο το όνομα και την ώρα στα οποία απαντά.

## Λειτουργία μεγέθυνσης

Μεγεθύνει την οθόνη καταρράκτη:

- **1x**: κανονική προβολή (ευρεία κάλυψη)
- **2x**: μεγέθυνση  $\times 2$
- **4x**: μεγέθυνση  $\times 4$
- **8x**: μεγέθυνση  $\times 8$

**Χρησιμοποιήστε το όταν:** χρειάζεται να δείτε τα σήματα καθαρότερα ή να συντονιστείτε με ακρίβεια.

## Έλεγχος πομποδέκτη (CAT)

Ο δικός σας πομποδέκτης και η σελίδα του δέκτη μπορούν να μένουν στην ίδια συχνότητα: γυρίστε το κουμπί συντονισμού του πομποδέκτη και ο καταρράκτης ακολουθεί, ή κάντε κλικ σε ένα σήμα στον καταρράκτη και ο πομποδέκτης συντονίζεται εκεί. Αφορά μόνο τη δική σας συνεδρία ακρόασης· κανείς άλλος στον δέκτη δεν επηρεάζεται.

**Με το Desktop PhantomSDR+ (4.0 ή νεότερο).** Η εφαρμογή υπολογιστή έχει μενού **Rig**. Το *Rig* → *Rig control...* ανοίγει ένα παράθυρο όπου επιλέγετε τον πομποδέκτη σας και τον τρόπο σύνδεσης, ενώ το ίδιο το μενού ανοίγει και κλείνει τον συγχρονισμό. Συγχρονίζει συχνότητα, τρόπο λειτουργίας και εύρος φίλτρου, προς τη μία κατεύθυνση ή και προς τις δύο, και μπορεί να σιγεί τον δέκτη όσο εκπέμπετε. Εκτός από το PhantomSDR-Plus χειρίζεται με τον ίδιο τρόπο και δέκτες **KiwiSDR**, **PA3FWM WebSDR** και **UberSDR**. Φτάνει στον πομποδέκτη με έναν από τέσσερις τρόπους:

- **Ενσωματωμένοι οδηγοί** — χωρίς άλλο λογισμικό: Icom (CI-V), Yaesu (νέο CAT και FT-817/857/897), Kenwood, Elecraft, FlexRadio SmartSDR CAT, QRP Labs και άλλοι συμβατοί με Kenwood πομποδέκτες.
- **Hamlib** — κάθε πομποδέκτης που γνωρίζει το Hamlib, πάνω από 300, από λίστα με αναζήτηση. Τα προγράμματα εγκατάστασης για Windows περιέχουν το Hamlib· σε Linux εγκαταστήστε το `libhamlib-utils`.
- **rigctld στο δίκτυο** — ένα `rigctld` που ήδη τρέχει.
- **f1rig** — για πομποδέκτη που το f1rig ήδη μοιράζεται με fldigi, WSJT-X ή πρόγραμμα ημερολογίου.

Η εφαρμογή, τα προγράμματα εγκατάστασης και το πλήρες εγχειρίδιό της βρίσκονται στο [Desktop PhantomSDR+ \(Dropbox\)](#).

**Με φυλλομετρητή (browser).** Το [CATsync Tool for WebSDRs](#) (Windows) συνδέει έναν πομποδέκτη με τη σελίδα του δέκτη που είναι ανοιχτή στον browser σας. Συγχρονίζει συχνότητα και τρόπο λειτουργίας.

**Το εύρος φίλτρου και η σίγαση κατά την εκπομπή** λειτουργούν σε δέκτες KiwiSDR, WebSDR και UberSDR, και σε δέκτη PhantomSDR-Plus με την 4.1.0 ή νεότερη. Σε παλαιότερο PhantomSDR-Plus η συχνότητα και ο τρόπος λειτουργίας συγχρονίζονται κανονικά· το φίλτρο όχι.

Το πλήρες εγχειρίδιο — κάθε ρύθμιση, οι υποστηριζόμενοι πομποδέκτες, πώς αποφεύγεται να «μαλώνουν» οι δύο πλευρές και η αντιμετώπιση προβλημάτων — είναι το **Έλεγχος πομποδέκτη**.

**Για προγραμματιστές.** Κάθε σελίδα δέκτη προσφέρει τις παρακάτω συναρτήσεις στο window, και αυτές χρησιμοποιούν και τα δύο εργαλεία:

Συνάρτηση	Λειτουργία
<code>catsync_getFrequency()</code> / <code>catsync_setFrequency(hz)</code>	Συχνότητα συντονισμού, σε Hz
<code>catsync_getMode()</code> / <code>catsync_setMode(mode)</code>	USB, LSB, CW, CW-L, AM, FM, WBFM ...
<code>catsync_getBandwidth()</code> / <code>catsync_setBandwidth(hz)</code>	Συνολικό εύρος διέλευσης, σε Hz. Ορίστε το μετά τον τρόπο λειτουργίας: η αλλαγή τρόπου επαναφέρει τη ζώνη διέλευσης
<code>catsync_getMute()</code> / <code>catsync_setMute(on)</code>	Σίγαση, μέσω του κουμπιού σίγασης της σελίδας
<code>catsync_ready</code>	<code>true</code> μόλις εγκατασταθούν οι συναρτήσεις

Οι τρεις τελευταίες γραμμές είναι νέες στην 4.1.0· ελέγξτε ότι μια συνάρτηση υπάρχει πριν την καλέσετε.

## Τρόποι αποδιαμόρφωσης

### AM (διαμόρφωση πλάτους)

Χρησιμοποιείται για:

- αεροπορικές επικοινωνίες
- ραδιοφωνικές εκπομπές AM
- ορισμένες ραδιοερασιτεχνικές επικοινωνίες
- ναυτιλιακές επικοινωνίες

Χαρακτηριστικά:

- ευρύ εύρος ζώνης (συνήθως 10 kHz)
- ευάλωτο στον θόρυβο
- εύκολος συντονισμός (αρκεί ένα κλικ στο σήμα)

Καλές πρακτικές:

- χρησιμοποιήστε γρήγορο AGC
- ενεργοποιήστε τον κόπτη θορύβου αν ακούτε κλικ
- συντονιστείτε στην κορυφή του σήματος στον καταρράκτη

### QUAM (στερεοφωνικό AM C-QUAM)

Χρησιμοποιείται για:

- Σταθμούς μεσαίων κυμάτων που εκπέμπουν στερεοφωνικό AM

Χαρακτηριστικά:

- Εύρος 10 kHz, αποκωδικοποιείται ως στερεοφωνικό ζεύγος αντί για μονοφωνικό AM
- Η ετικέτα του κουμπιού γίνεται **πράσινη** από μόνη της όταν υπάρχει ο στερεοφωνικός πιλότος των 25 Hz, ώστε να βλέπετε ποιοι σταθμοί εκπέμπουν πράγματι στερεοφωνικά πριν τον επιλέξετε
- Από το AM, πατώντας ξανά το κουμπί περνάτε σε σύγχρονη ανίχνευση AM στο φέρον· η ετικέτα γίνεται **κίτρινη** και δείχνει SAM. Τρίτο πάτημα επιστρέφει στο απλό AM
- Ο ήχος C-QUAM μεταφέρεται με Opus· κάθε άλλη λειτουργία χρησιμοποιεί FLAC

**Καλές πρακτικές:**

- Ψάξτε για την πράσινη ετικέτα σε ισχυρά σήματα μεσαίων κυμάτων τη νύχτα
- Αν το στερεοφωνικό ακούγεται ασταθές, το SAM είναι πιο σταθερή επιλογή σε αδύναμο φέρον

**FM (διαμόρφωση συχνότητας)****Χρησιμοποιείται για:**

- ραδιοερασιτεχνικούς αναμεταδότες VHF/UHF
- υπηρεσίες κοινής ωφέλειας (αστυνομία, πυροσβεστική, ΕΚΑΒ)
- εμπορικές αμφίδρομες επικοινωνίες
- ορισμένες δορυφορικές επικοινωνίες

**Χαρακτηριστικά:**

- στενό εύρος ζώνης (συνήθως 12,5 ή 25 kHz)
- εξαιρετική ανοσία στον θόρυβο
- «φαινόμενο κατάληψης» (επικρατεί το ισχυρότερο σήμα)

**Καλές πρακτικές:**

- συντονιστείτε με ακρίβεια στο κέντρο του σήματος
- χρησιμοποιήστε squelch για σίγαση στην ηρεμία
- απενεργοποιήστε τη μείωση θορύβου (δεν χρειάζεται)

**USB (άνω πλευρική ζώνη)****Χρησιμοποιείται για:**

- ραδιοερασιτεχνικά HF (πάνω από 10 MHz)
- τους περισσότερους ψηφιακούς τρόπους HF
- ναυτιλιακές επικοινωνίες (πάνω από 8 MHz)

**Χαρακτηριστικά:**

- στενό εύρος ζώνης (συνήθως 2.4 kHz)
- αποδοτική χρήση του φάσματος
- απαιτεί ακριβή συντονισμό

**Καλές πρακτικές:**

- χρησιμοποιήστε αργό AGC
- συντονιστείτε στο κάτω άκρο του σήματος στον καταρράκτη
- ενεργοποιήστε τη μείωση θορύβου αν χρειάζεται

**LSB (κάτω πλευρική ζώνη)****Χρησιμοποιείται για:**

- ραδιοερασιτεχνικά HF (κάτω από 10 MHz)
- ορισμένους ψηφιακούς τρόπους HF
- ναυτιλιακές επικοινωνίες (κάτω από 8 MHz)

**Χαρακτηριστικά:**

- ίδιο με το USB αλλά κατοπτρικά
- σύμβαση: LSB στις χαμηλότερες μπάντες HF

**Καλές πρακτικές:**

- χρησιμοποιήστε αργό AGC
- συντονιστείτε στο άνω άκρο του σήματος στον καταρράκτη
- ενεργοποιήστε τη μείωση θορύβου αν χρειάζεται

**CW (συνεχές κύμα / κώδικας Μορς)****Χρησιμοποιείται για:**

- ραδιοερασιτεχνική τηλεγραφία
- ραδιοφάρους ναυσιπλοΐας
- σταθμούς σημάτων ώρας

**Χαρακτηριστικά:**

- πολύ στενό εύρος ζώνης (100-500 Hz)
- υψηλή απόδοση
- απαιτεί γνώση του κώδικα Μορς για κατανόηση

**Καλές πρακτικές:**

- χρησιμοποιήστε στενό φίλτρο (400-500 Hz)
- συντονιστείτε με ακρίβεια στο κέντρο του τόνου
- ενεργοποιήστε το φίλτρο ήχου για καλύτερο τόνο

**CW-L (CW, κάτω πλευρική ζώνη)**

Το ίδιο φίλτρο Μορς  $\pm 250$  Hz με το **CW**, αλλά ο τόνος λαμβάνεται από κάτω από το φέρον αντί από πάνω. Χρησιμοποιήστε το όταν ένα σήμα ακούγεται καθαρότερα από την κάτω πλευρά, ή όταν ένα παρεμβάλλον φέρον βρίσκεται ακριβώς πάνω από αυτό που θέλετε.

**Δεν υπάρχει κουμπί CW-L στη σελίδα υπολογιστή.** Η σειρά λειτουργιών της είναι

USB · LSB · CW · AM · QUAM · FM. Το CW-L προσφέρεται στην απλοποιημένη σελίδα /mobile, ενώ στον υπολογιστή μπορεί να οριστεί από σύνδεσμο ή σελιδοδείκτη που το ονομάζει.

**WBFM (ευρυζωνικό FM)****Χρησιμοποιείται για:**

- ραδιοφωνικές εκπομπές FM (88-108 MHz)
- ορισμένες δορυφορικές καθοδικές ζεύξεις

**Χαρακτηριστικά:**

- πολύ ευρύ εύρος ζώνης (200 kHz)
- εξαιρετική ποιότητα ήχου
- υψηλή πιστότητα

**Καλές πρακτικές:**

- συντονιστείτε ακριβώς στην κεντρική συχνότητα
- δεν χρειάζεται squelch για ραδιοφωνικές εκπομπές
- απολαύστε ήχο υψηλής ποιότητας!

## Ψηφιακοί αποκωδικοποιητές

Το PhantomSDR-Plus περιλαμβάνει ενσωματωμένους αποκωδικοποιητές για ψηφιακούς τρόπους. Για πλήρη οδηγό, δείτε τους [Αποκωδικοποιητές](#).

Ο γρηγορότερος τρόπος είναι η σειρά κουμπιών **Decoders** στον κύριο πίνακα, ακριβώς κάτω από τα **Wheel Tuning Steps**: δέκα κουμπιά — **FT8, FT4, FT2, JS8, CW, WSPR, FAX, SSTV, NAVTX, RTTY** — που το καθένα, με ένα μόνο πάτημα, ξεκινά τον αποκωδικοποιητή του, ανάβει τον κεντρικό διακόπτη Decoder και φέρνει το παράθυρο του αποκωδικοποιητή στην οθόνη. Το κουμπί παραμένει μπλε όσο τρέχει ο αποκωδικοποιητής· πατήστε το ξανά για να τον σταματήσετε και να κλείσει το παράθυρό του. Το αναπτυσσόμενο μενού **Decoder Options** λειτουργεί ακριβώς όπως πριν και παραμένει συγχρονισμένο με τα κουμπιά.

Τα **RADEL** και **RADEU** (ψηφιακή φωνή RADE v1) έχουν δικά τους κουμπιά δίπλα στην επικεφαλίδα **Modes selector** και μέσα στα αναδυόμενα παράθυρα **Modes** και **Bands**, και λειτουργούν με τον ίδιο τρόπο: πάτημα για εκκίνηση, δεύτερο πάτημα για διακοπή.

Οι αποκωδικοποιητές SSTV, HF FAX, NAVTEX, FSK/RTTY και CW εκτελούνται ο καθένας σε δικό του νήμα παρασκηνίου, ώστε η ενεργοποίηση ή απενεργοποίηση ενός αποκωδικοποιητή να μη διακόπτει ποτέ τον ήχο και ο καταρράκτης να παραμένει ομαλός κατά την αποκωδικοποίηση. Τροφοδοτούνται με ήχο που λαμβάνεται πριν από το AGC, τη μείωση θορύβου και τη σίγαση — μπορείτε να σιγάσετε τον δέκτη και η αποκωδικοποίηση συνεχίζεται ανεπηρέαστη.

**Όσο τρέχει ένας αποκωδικοποιητής, κρατά τον τρόπο λήψης και το εύρος διέλευσης.**

Κανονικά ο τρόπος λήψης ακολουθεί τον χάρτη μπαντών — συντονίστε σε τμήμα σημειωμένο ως LSB ή AM και ο δέκτης αλλάζει. Ένας αποκωδικοποιητής που τρέχει υπερισχύει, κρατώντας τον τρόπο λήψης και το στενό εύρος που χρειάζεται, ακόμη κι αν πάτε σε άλλη μπάντα. Ο τρόπος λήψης της μπάντας επιστρέφει μόλις σβήσετε τον αποκωδικοποιητή, και τα κουμπιά των modes λειτουργούν πάντα αν θέλετε να πάρετε εσείς τον έλεγχο. Ο αποκωδικοποιητής CW είναι η εξαίρεση: αποκωδικοποιεί στον τρόπο λήψης που ήδη ακούτε.

### Αποκωδικοποιητής FT8, FT4

**Τι είναι το FT8 και το FT4;**

- δημοφιλής ψηφιακός ραδιοερασιτεχνικός τρόπος
- επικοινωνία με ασθενή σήματα
- εκπομπές 15 δευτερολέπτων για FT8, 7,5 δευτερολέπτων για FT4

**Πώς να το χρησιμοποιήσετε:**

1. Συντονιστείτε σε συχνότητες FT8 ή FT4
2. Επιλέξτε τρόπο USB
3. Ενεργοποιήστε τον αποκωδικοποιητή FT8 — πατήστε το κουμπί **FT8** ή επιλέξτε τον από το μενού
4. Παρακολουθήστε τα αποκωδικοποιημένα μηνύματα να εμφανίζονται

**Συνήθεις συχνότητες FT8 (USB):**

- 160 μέτρα: 1.840 MHz
- 80 μέτρα: 3.573 MHz
- 60 μέτρα: 5.357 MHz
- 40 μέτρα: 7.074 MHz
- 30 μέτρα: 10.136 MHz
- 20 μέτρα: 14.074 MHz
- 17 μέτρα: 18.100 MHz

- 15 μέτρα: 21.074 MHz
- 12 μέτρα: 24.915 MHz
- 10 μέτρα: 28.074 MHz
- 6 μέτρα: 50.313 MHz (50.323 MHz για DX)

#### Συνήθεις συχνότητες FT4 (USB):

- 80 m: 3.575 MHz
- 40 m: 7.0475 MHz
- 30 m: 10.140 MHz
- 20 m: 14.080 MHz
- 17 m: 18.104 MHz
- 15 m: 21.140 MHz
- 12 m: 24.919 MHz
- 10 m: 28.180 MHz
- 6 m: 50.318 MHz

## Αποκωδικοποιητής JS8

### Τι είναι το JS8;

- Η μηχανή ασθενούς σήματος του FT8, για συνομιλία από πληκτρολόγιο σε πληκτρολόγιο
- Ελεύθερο κείμενο αντί για σταθερά μηνύματα: τα μεγάλα μηνύματα έρχονται σε πολλούς κύκλους
- Πέντε ταχύτητες· η **Normal** (15 δευτ.) είναι η ταχύτητα κλήσης και έχει σχεδόν όλη την κίνηση

### Πώς χρησιμοποιείται:

1. Συντονιστείτε σε συχνότητα JS8
2. Επιλέξτε **USB**
3. Ενεργοποιήστε τον αποκωδικοποιητή JS8 — κουμπί **JS8** ή από το μενού
4. Αφήστε την **Speed** στο *Normal* και το sync offset στο **Auto**

### Πώς διαβάζεται το παράθυρο:

- Τα μηνύματα που ακόμη έρχονται εμφανίζονται επάνω με πράσινο και δρομέα που αναβοσβήνει — στο Normal ένα μήνυμα μπορεί να θέλει ένα ολόκληρο λεπτό
- Τα ολοκληρωμένα μπαίνουν από κάτω ως Mode | Hz | dB | Message, με **τα διακριτικά πράσινα**
- Γραμμή σε σβησμένα πλάγια σημαίνει ότι έληξε ο χρόνος πριν έρθει το τελευταίο πλαίσιο: το κείμενο είναι αληθινό αλλά ίσως κομμένο

#### Συνήθεις συχνότητες JS8 (USB):

- 160m: 1.842 MHz
- 80m: 3.578 MHz
- 40m: 7.078 MHz
- 30m: 10.130 MHz
- 20m: 14.078 MHz
- 17m: 18.104 MHz
- 15m: 21.078 MHz
- 12m: 24.922 MHz
- 10m: 28.078 MHz

Το JS8 είναι πολύ πιο ήσυχο από το FT8 — μία εκπομπή κάθε λίγα λεπτά είναι φυσιολογική και τα κενά δεν είναι βλάβη. Πλήρης οδηγός: [Αποκωδικοποιητές](#).

## Αποκωδικοποιητής CW

Απλώς πατήστε το κουμπί CW και θα εμφανιστεί το αποκωδικοποιημένο κείμενο. Πατήστε ξανά το κουμπί CW για να καθαρίσετε το παράθυρο και να επανεκκινήσετε τον αποκωδικοποιητή.

## WSPR

Το WSPR (Weak Signal Propagation Reporter, προφέρεται «whisper») είναι ένας τρόπος φάρου εξαιρετικά ασθενών σημάτων που χαρτογραφεί τις διαδρομές διάδοσης HF παγκοσμίως. Κάθε εκπομπή διαρκεί περίπου 110 δευτερόλεπτα και χωρά σε μια θυρίδα εύρους 200 Hz. Ο αποκωδικοποιητής περιμένει μια πλήρη θυρίδα 2 λεπτών ευθυγραμμισμένη με το UTC πριν αποκωδικοποιήσει. Ενεργοποιήστε τον αποκωδικοποιητή και επιλέξτε **WSPR** από την αναπτυσσόμενη λίστα. Ή απλώς πατήστε το κουμπί **WSPR**.

## HF FAX / WEFAX

Το HF Radiofax (γνωστό και ως WEFAX) χρησιμοποιείται από ακτοφυλακές και μετεωρολογικές υπηρεσίες παγκοσμίως για τη μετάδοση μετεωρολογικών χαρτών, χαρτών κατάστασης θάλασσας και αναλύσεων επιφανείας μέσω βραχέων κυμάτων. Ο αποκωδικοποιητής ανασυνθέτει την εικόνα γραμμή προς γραμμή καθώς λαμβάνεται.

## NAVTEX

Το NAVTEX είναι το διεθνές ναυτιλιακό σύστημα εκπομπής πληροφοριών παράκτιας ασφάλειας — προειδοποιήσεις ναυσιπλοΐας, μετεωρολογικές προγνώσεις και ειδοποιήσεις έρευνας και διάσωσης.

## FSK / RTTY, PSK31 και Olivia

Ένας αποκωδικοποιητής γενικής χρήσης για στενής ζώνης ψηφιακά κειμένου, με πέντε παραλλαγές σε ένα παράθυρο: ναυτιλιακό FSK (SITOR), μετεωρολογικό RTTY, ερασιτεχνικό RTTY, **PSK31** (διαμόρφωση φάσης, 31,25 baud) και **Olivia** (πολυτονικό FSK με διόρθωση σφαλμάτων). Ο πίνακας προσαρμόζεται στην παραλλαγή — τα χειριστήρια shift, baud και πλαισίωσης εξαφανίζονται για PSK31 και Olivia, ενώ το Olivia προσθέτει επιλογή Mode και ρυθμιστικό squelch.

Το ερασιτεχνικό και το μετεωρολογικό RTTY έχουν ρυθμιστικό **Squelch (SNR)** που εμποδίζει τον θόρυβο να τυπώνεται ως τυχαία γράμματα, και όλες οι παραλλαγές εκτός από το Olivia δείχνουν πραγματικό SNR στην ίδια κλίμακα των 3 kHz. Δύο πράγματα αξίζει να ξέρετε: το PSK31 διορθώνει μόνο του το σφάλμα συντονισμού σε εύρος περίπου  $\pm 25$  Hz, οπότε αρκεί να πλησιάσετε· το Olivia απαιτεί το **Mode** (τόνοι / εύρος ζώνης) να ταιριάζει ακριβώς με την εκπομπή, δεν στέλνει προοίμιο και επομένως χρειάζεται λίγα δευτερόλεπτα για να συγχρονιστεί πριν εμφανιστεί κείμενο. Ο πίνακας ανοίγει στο Olivia **8 / 250**.

## QRSS Grabber

Το QRSS είναι CW σταλμένο τόσο αργά ώστε μια τελεία να διαρκεί δευτερόλεπτα, και το παρακολουθείς με το μάτι αντί με το αυτί — ο grabber ζωγραφίζει το ίχνος στη δική του οθόνη. Δεν βρίσκεται στο μενού των αποκωδικοποιητών· έχει δικό του τμήμα **QRSS** και μπορεί να τρέχει ταυτόχρονα με έναν αποκωδικοποιητή.

Πατήστε **Show**, μετά διαλέξτε παράθυρο από τη λίστα **Band** και πατήστε **Tune**: ο δέκτης πηγαίνει σε εκείνη τη συχνότητα σε **CW** με εύρος διέλευσης κομμένο για τη λήψη, και το ίχνος πέφτει στην κεντρική γραμμή της οθόνης. Τα 30 m (10.140,00 kHz) είναι το πιο πολυσύχναστο παράθυρο. Ρυθμίστε την **Speed** ώστε να ταιριάζει με τον φάρο — **QRSS 10** αν δεν ξέρετε — και κάντε υπομονή: ένα διακριτικό μπορεί να χρειαστεί δέκα λεπτά για να διασχίσει την οθόνη. Λεπτομέρειες στους [Αποκωδικοποιητές](#).

## SSTV

Η τηλεόραση αργής σάρωσης στέλνει ακίνητες εικόνες μέσω ενός συνηθισμένου καναλιού SSB, μία γραμμή τη φορά. Ενεργοποιήστε τον αποκωδικοποιητή, επιλέξτε **SSTV** και συντονιστείτε σε μια συχνότητα SSTV — τα 14.230 MHz είναι η κύρια διεθνής συχνότητα κλήσης. Αφήστε το **Mode** στο **Auto**: ο αποκωδικοποιητής διαβάζει την κεφαλίδα VIS της εκπομπής και, αν συντονιστήκατε μετά την κεφαλίδα, αναγνωρίζει τον τρόπο από τον χρονισμό συγχρονισμού. Υποστηρίζονται οι τρόποι Martin, Scottie και Robot, και η εικόνα χτίζεται γραμμή προς γραμμή καθώς φτάνει. Ή απλώς πατήστε το κουμπί **SSTV**.

## Αυτόματη αναφορά spot και γραφήματα του διακομιστή

Οι αποκωδικοποιήσεις FT8, FT4 και WSPR μπορούν να ανεβαίνουν αυτόματα και από τον ίδιο τον διακομιστή — FT8/FT4 στο PSK Reporter, WSPR στο WSPRnet — μέσω ενός δαίμονα autorun που ξεκινά ο διαχειριστής από τον πίνακα διαχείρισης. Λειτουργεί ανεξάρτητα από τους αποκωδικοποιητές του φυλλομετρητή σας: συνεχίζει είτε ακούει κάποιος είτε όχι, και τίποτα από όσα αποκωδικοποιείτε στον φυλλομετρητή δεν αναφέρεται.

Ο διαχειριστής τον παρακολουθεί με δύο μετρητές, που εύκολα μπερδεύονται μεταξύ τους: τα πλακίδια ανά αποκωδικοποιητή μετρούν τα spot που ανέβηκαν από την τελευταία εκκίνηση του δαίμονα, ενώ ο αριθμός δίπλα σε κάθε πλαίσιο ελέγχου μπάντας/τρόπου είναι το συνολικό άθροισμα της θέσης από την αρχή και επιβιώνει των επανεκκινήσεων. Ο ίδιος πίνακας διαθέτει σελίδα **Γραφήματα** που σχεδιάζει συχνότητα CPU, φόρτο CPU, θερμοκρασία CPU και συνδεδεμένους χρήστες για τα τελευταία 15 λεπτά έως 24 ώρες. Και τα δύο περιγράφονται στον [οδηγό του πίνακα διαχείρισης](#).

## Συντομεύσεις πληκτρολογίου

Συντομεύσεις πληκτρολογίου για ταχύτερο χειρισμό.

### Έλεγχος συχνότητας

Η ένδειξη συχνότητας είναι μια σειρά ψηφίων που χειρίζεστε απευθείας. **Κάντε πρώτα κλικ σε ένα ψηφίο** — αυτό το επιλέγει, και όλα τα παρακάτω ενεργούν στην επιλογή.

- **Βέλος αριστερά / δεξιά**: μετακινεί την επιλογή στο επόμενο ψηφίο (οκτώ ψηφία, από 100 MHz έως 10 Hz)
- **Βέλος πάνω / κάτω**: αυξάνει ή μειώνει το *επιλεγμένο* ψηφίο κατά την αξία της θέσης του — άρα το βήμα είναι 1 MHz στο ψηφίο των MHz, 10 Hz στο τελευταίο
- **0 - 9**: πληκτρολογεί το ψηφίο απευθείας στην επιλεγμένη θέση

**Ροδέλα ποντικιού πάνω από την ένδειξη συχνότητας**: βηματίζει με το βήμα συντονισμού της μπάντας (1 kHz εκτός αν το σχέδιο μπάντας ορίζει άλλο). Κρατήστε **Shift** για 1 kHz, **Alt** για 10 kHz.

**Ροδέλα ποντικιού πάνω από τον καταρράκτη**: κάνει ζουμ. Κρατήστε **Ctrl** (ή **Cmd**) ή **Shift** για συντονισμό αντί για ζουμ, και **Shift + Ctrl** μαζί για στρογγυλοποίηση στο πλησιέστερο ακέραιο kHz.

Δεν υπάρχουν συντομεύσεις Page Up / Page Down.

## Σελιδοδείκτες

Αποθηκεύστε τις αγαπημένες σας συχνότητες για γρήγορη πρόσβαση. Μπορείτε επίσης να εξαγάγετε τη λίστα σελιδοδεικτών και να την αποθηκεύσετε τοπικά, και έπειτα να την εισαγάγετε σε οποιοδήποτε άλλο PhantomSDR.

Όταν σελιδοδείκτες και δείκτες επικαλύπτονται:

Οι μπλε σελιδοδείκτες εμφανίζονται από πάνω

Οι κίτρινοι δείκτες εμφανίζονται από κάτω

Τα κλικ στους σελιδοδείκτες έχουν προτεραιότητα

### Προσθήκη σελιδοδείκτη

1. Συντονιστείτε στην επιθυμητή συχνότητα
2. Κάντε κλικ στο κουμπί «Bookmarks»
3. Κάντε κλικ στο «Add Bookmark»
4. Εισαγάγετε περιγραφή
5. Κάντε κλικ στο «Save»

				Bookmark name		Label (optional)		[Add]	
--	--	--	--	---------------	--	------------------	--	-------	--

### Πώς χρησιμοποιείται:

#### Προσθήκη σελιδοδείκτη:

- Όνομα: «Local News Station»
- Ετικέτα: «NEWS»
- Κάντε κλικ στο Add

#### Προβολή στον καταρράκτη:

- κάντε μεγέθυνση μέχρι να εμφανιστούν οι δείκτες
- θα δείτε ένα σκούρο μπλε πλαίσιο με το «NEWS» σε έντονο κίτρινο

#### Κλικ σε σελιδοδείκτη:

- συντονίζει στη συχνότητα
- ορίζει τον τρόπο αποδιαμόρφωσης
- λειτουργεί ακριβώς όπως το κλικ σε έναν δείκτη

### Διαχείριση σελιδοδεικτών

- **Επεξεργασία:** κάντε κλικ στο εικονίδιο μολυβιού δίπλα στον σελιδοδείκτη
- **Διαγραφή:** κάντε κλικ στο εικονίδιο κάδου δίπλα στον σελιδοδείκτη
- **Εξαγωγή:** κατεβάστε τους σελιδοδείκτες ως αρχείο JSON
- **Εισαγωγή:** ανεβάστε σελιδοδείκτες από αρχείο JSON

### Κοινή χρήση σελιδοδεικτών

1. Κάντε κλικ στο «Export Bookmarks»

2. Μοιραστείτε το αρχείο JSON με άλλους
3. Οι παραλήπτες κάνουν κλικ στο «Import Bookmarks»
4. Επιλέγουν το αρχείο σας

## Χρήση σε κινητά

Το PhantomSDR-Plus λειτουργεί εξαιρετικά σε κινητές συσκευές!

### Δύο προβολές για κινητά

Υπάρχουν δύο τρόποι να χρησιμοποιήσετε τον δέκτη από κινητό:

- **http://o\_διακομιστής\_σας:PORT/mobile** — η απλοποιημένη σελίδα. Χωρίς καταρράκτη, άρα περίπου η μισή κατανάλωση δεδομένων. Εισαγωγή συχνότητας, βήματα συντονισμού, τρόποι λειτουργίας, S-meter, μπάντες, σελιδοδείκτες, χρήστες και συνομιλία.
- **Η εκτεταμένη προβολή** — η διάταξη της κύριας διεπαφής για κινητά, με καταρράκτη και όλα τα χειριστήρια.

Η εναλλαγή γίνεται με τα κουμπιά στο κάτω μέρος του **/mobile** (**Mobile extended view, Full desktop view**) και με το κουμπί **Simplified mobile** στην εκτεταμένη προβολή.

**Η συχνότητά σας σας ακολουθεί.** Η αλλαγή προβολής σας κρατά στο ίδιο σήμα — η συχνότητα και ο τρόπος λειτουργίας ταξιδεύουν μέσα στον σύνδεσμο, οπότε δεν καταλήγεται πια στην προεπιλεγμένη συχνότητα του δέκτη. Και η γραμμή διευθύνσεων ακολουθεί τον συντονισμό σας: η ανανέωση της σελίδας, ένας σελιδοδείκτης ή ένας σύνδεσμος που στέλνετε σε κάποιον επιστρέφουν ακριβώς σε αυτή τη συχνότητα.

**Ο τρόπος λειτουργίας ακολουθεί το σχέδιο μπαντών και στις δύο σελίδες.** Συντονιστείτε σε τμήμα που δηλώνεται ως AM, LSB, USB ή CW — πληκτρολογώντας συχνότητα, με βηματικό συντονισμό ή με κουμπί μπάντας — και ο δέκτης αλλάζει σε αυτόν τον τρόπο, τόσο στην απλοποιημένη σελίδα όσο και στην πλήρη διεπαφή. Ένας τρόπος που επιλέγετε με το χέρι παραμένει όσο κινείστε μέσα στο ίδιο τμήμα, ενώ έξω από τις καθορισμένες μπάντες ο τρόπος σας δεν πειράζεται. Όσο τρέχει το **RADE** (RADEL/RADEU) κρατά τον δέκτη, οπότε ο συντονισμός δεν το διακόπτει.

Όταν αλλάζετε προβολή, ο τρέχων τρόπος λειτουργίας έρχεται μαζί σας· αν η προβολή στην οποία πηγαίνετε δεν τον διαθέτει, αποφασίζει το σχέδιο μπαντών για τη συγκεκριμένη συχνότητα — μια συχνότητα εκπομπής φτάνει σε AM, τα 40 m σε LSB, ένα τμήμα CW σε CW. Το **SAM** της απλοποιημένης σελίδας γίνεται AM με τον σύγχρονο ανιχνευτή στην εκτεταμένη προβολή, και αντίστροφα. Τα **RADEL** και **RADEU** υπάρχουν μόνο στην απλοποιημένη σελίδα: φεύγοντας από αυτά, η συχνότητα διατηρείται και τον τρόπο λειτουργίας τον επιλέγει το σχέδιο μπαντών.

Μια συχνότητα εκτός της κάλυψης του δέκτη επαναφέρεται στο πλησιέστερο άκρο, ώστε ένας παλιός σύνδεσμος να μη σας αφήνει ποτέ εκτός μπάντας.

### Δυνατότητες ειδικά για κινητά

- **Χειριστήρια φιλικά στην αφή:** μεγάλα κουμπιά και ρυθμιστές
- **Σύρσιμο για συντονισμό:** σύρετε αριστερά/δεξιά στον καταρράκτη
- **Τσιμπίμα για μεγέθυνση:** τσιμπίστε τον καταρράκτη για μεγέθυνση/σμίκρυνση
- **Οριζόντιος προσανατολισμός:** περιστρέψτε για καλύτερη προβολή

### Συμβουλές για κινητά

1. **Χρησιμοποιήστε Wi-Fi:** η ροή ήχου καταναλώνει δεδομένα
2. **Οριζόντιος προσανατολισμός:** καλύτερη προβολή του καταρράκτη

3. **Ακουστικά:** καλύτερη ποιότητα ήχου
4. **Αποθηκεύστε αγαπημένα:** ευκολότερη επιστροφή σε σταθμούς
5. **Κλείστε άλλες εφαρμογές:** εξασφαλίζει ομαλή απόδοση

### Συνιστώμενοι περιηγητές για κινητά

- **Android:** Chrome ή Samsung Internet
- **iOS:** Mozilla
- **Και στα δύο:** φροντίστε ο περιηγητής να είναι ενημερωμένος

## Συμβουλές και καλές πρακτικές

### Για την καλύτερη λήψη

1. **Επιλέξτε ισχυρά σήματα:** αναζητήστε πορτοκαλί/κόκκινο στον καταρράκτη
2. **Συντονιστείτε με ακρίβεια:** κάντε κλικ ακριβώς στο κέντρο του σήματος
3. **Επιλέξτε τον σωστό τρόπο:** ταιριάξτε τον με τον τύπο του σήματος
4. **Ρυθμίστε το AGC:** γρήγορο για AM, αργό για SSB
5. **Χρησιμοποιήστε NR/NB:** βοηθούν σε θορυβώδεις συνθήκες

### Εντοπισμός δραστηριότητας

1. **Παρακολουθήστε τον καταρράκτη:** τα χρώματα δείχνουν την ισχύ του σήματος
2. **Ακούστε σε δημοφιλείς συχνότητες:**
3. 40 m: 7.100-7.300 MHz (LSB)
4. 20 m: 14.200-14.350 MHz (USB)
5. 2 m: 145.200-145.600 MHz (FM)
6. **Ελέγξτε την επικάλυψη του σχεδίου μπαντών:** δείχνει τις κατανομές συχνοτήτων
7. **Χρησιμοποιήστε σελιδοδείκτες:** γρήγορη πρόσβαση σε ενεργές συχνότητες

### Κατανόηση των συνθηκών διάδοσης

#### HF την ημέρα (υψηλές συχνότητες):

- οι υψηλότερες μπάντες αποδίδουν καλύτερα (20 m, 15 m, 10 m)
- είναι δυνατή η επικοινωνία μεγάλων αποστάσεων (DX)
- ακούγονται ραδιοφωνικοί σταθμοί

#### HF τη νύχτα:

- οι χαμηλότερες μπάντες αποδίδουν καλύτερα (80 m, 40 m)
- διαφορετικά μοτίβα διάδοσης
- ακούγονται διαφορετικοί σταθμοί

#### VHF/UHF:

- κυρίως οπτική επαφή
- τοπικές επικοινωνίες
- πιο σταθερές συνθήκες

## Δεοντολογία

1. **Μη δεσμεύετε τον δέκτη:** και άλλοι θέλουν να ακούσουν
2. **Χρησιμοποιήστε τη συνομιλία με σεβασμό:** να είστε ευγενικοί με τους άλλους χρήστες
3. **Αναφέρετε προβλήματα:** βοηθήστε τον διαχειριστή να συντηρήσει τον σταθμό
4. **Μη ζητάτε τεχνική υποστήριξη:** πρόκειται για πλατφόρμα ακρόασης. Στείλτε μήνυμα στον SysOp για βοήθεια.

## Αντιμετώπιση προβλημάτων

### Δεν υπάρχει ήχος

#### Πιθανές αιτίες:

1. Σίγαση στον περιηγητή → ελέγξτε τα χειριστήρια έντασης του περιηγητή
2. Σίγαση στο σύστημα → ελέγξτε την ένταση του υπολογιστή
3. Ασθενές σήμα → συντονιστείτε σε ισχυρότερο σήμα (S7+)
4. Λάθος τρόπος → δοκιμάστε άλλους τρόπους αποδιαμόρφωσης

#### Λύσεις:

1. Κάντε κλικ σε ένα ισχυρό σήμα (πορτοκαλί/κόκκινο στον καταρράκτη)
2. Ελέγξτε ότι ο περιηγητής δεν είναι σε σίγαση (εικονίδιο σίγασης στην καρτέλα)
3. Δοκιμάστε άλλη συχνότητα
4. Ανανεώστε τη σελίδα (F5)

### Παραμορφωμένος ήχος

#### Πιθανές αιτίες:

1. Υπερβολικά ισχυρό σήμα → κορεσμός
2. Λάθος τρόπος → σήμα AM σε τρόπο SSB κ.λπ.
3. Παρεμβολές → διαρροή από γειτονικά σήματα

#### Λύσεις:

1. Χαμηλώστε την ένταση
2. Δοκιμάστε άλλον τρόπο αποδιαμόρφωσης
3. Χρησιμοποιήστε στενότερο εύρος φίλτρου
4. Απομακρυνθείτε από τα σήματα που παρεμβάλλονται

### Ο ήχος διακόπτεται ή καθυστερεί

#### Πιθανές αιτίες:

1. Ακανόνιστη σύνδεση → τα πακέτα φτάνουν αργά και το buffer αδειάζει
2. Φορτωμένο κινητό ή υπολογιστής → ο φυλλομετρητής δεν προλαβαίνει

#### Λύσεις:

1. Ανεβάστε το buffer μία βαθμίδα (δείτε *Buffer ήχου και καθυστέρηση*)
2. Προτιμήστε Wi-Fi ή καλώδιο από τα δεδομένα κινητής
3. Κλείστε άλλες καρτέλες και εφαρμογές

Για πιο προσεκτική ματιά, ανοίξτε τη σελίδα προσθέτοντας `?diag=1` στη διεύθυνσή της (για παράδειγμα `http://your_server:PORT/mobile?diag=1`). Η σελίδα `/mobile` δείχνει τότε κάτω από τα χειριστήριά της την καθυστέρηση, πόσο ακανόνιστη είναι η σύνδεση και πώς τα πάει το buffer· η σελίδα `desktop` γράφει την ίδια γραμμή στην κονσόλα του φυλλομετρητή (F12) κάθε δέκα δευτερόλεπτα. Χρήσιμο όταν αναφέρετε ένα πρόβλημα στον διαχειριστή του σταθμού.

## Ο καταρράκτης δεν ανανεώνεται

### Πιθανές αιτίες:

1. Πρόβλημα δικτύου → αργή ή διακοπτόμενη σύνδεση
2. Απόδοση περιηγητή → πολλές ανοιχτές καρτέλες
3. Υπερφόρτωση διακομιστή → πολλοί χρήστες

### Λύσεις:

1. Ελέγξτε τη σύνδεση στο διαδίκτυο
2. Κλείστε τις περιττές καρτέλες
3. Ανανεώστε τη σελίδα (F5)
4. Δοκιμάστε αργότερα, όταν υπάρχουν λιγότεροι χρήστες

## Δεν είναι δυνατός ο συντονισμός σε συχνότητα

### Πιθανές αιτίες:

1. Συχνότητα εκτός εύρους → το SDR δεν καλύπτει αυτή τη συχνότητα
2. Λάθος μορφή πληκτρολόγησης → χρησιμοποιήστε τη σωστή μορφή (π.χ. «14200» και όχι «14.200.000»)

### Λύσεις:

1. Ελέγξτε την κάλυψη συχνοτήτων του SDR (εμφανίζεται στη σελίδα)
2. Χρησιμοποιήστε τα παραδείγματα μορφής συχνότητας που δίνονται
3. Κάντε κλικ στον καταρράκτη αντ' αυτού

## Ήχος με διακοπές

### Πιθανές αιτίες:

1. αργή σύνδεση στο διαδίκτυο
2. υψηλό φορτίο διακομιστή
3. προβλήματα απόδοσης του περιηγητή

### Λύσεις:

1. Κλείστε άλλες εφαρμογές που καταναλώνουν εύρος ζώνης
2. Δοκιμάστε ξανά σε ώρες μειωμένης κίνησης
3. Κλείστε τις περιττές καρτέλες
4. Χρησιμοποιήστε ενσύρματη σύνδεση αντί για Wi-Fi

## Συχνές ερωτήσεις

---

### Γενικές ερωτήσεις

**Ε: Χρειάζομαι ειδικό εξοπλισμό για να χρησιμοποιήσω το WebSDR;** Α: Όχι! Μόνο έναν υπολογιστή ή μια κινητή συσκευή με πρόσβαση στο διαδίκτυο.

**Ε: Η χρήση του WebSDR είναι δωρεάν;** Α: Ναι, τα περισσότερα WebSDR είναι δωρεάν. Λειτουργούν από εθελοντές.

**Ε: Μπορώ να εκπέμψω με το WebSDR;** Α: Όχι, το WebSDR είναι μόνο για λήψη. Δεν μπορείτε να εκπέμψετε.

**Ε: Σε ποιες συχνότητες μπορώ να ακούσω;** Α: Εξαρτάται από τη διαμόρφωση του WebSDR. Ελέγξτε τις πληροφορίες του σταθμού.

**Ε: Μπορώ να ηχογραφήσω;** Α: Ορισμένοι περιηγητές επιτρέπουν την εγγραφή. Ελέγξτε τις δυνατότητες του περιηγητή σας.

### Τεχνικές ερωτήσεις

**Ε: Τι ρυθμό δειγματοληψίας χρησιμοποιεί το SDR;** Α: Διαφέρει ανά σταθμό. Ελέγξτε τη σελίδα πληροφοριών του σταθμού.

**Ε: Ποια είναι η καθυστέρηση;** Α: Συνήθως μερικά δέκατα του δευτερολέπτου ανάμεσα στο ραδιοσήμα και τα ηχεία σας: η επεξεργασία στον δέκτη, το δίκτυο και το buffer αναπαραγωγής του φυλλομετρητή, που προσαρμόζεται στη σύνδεσή σας (περίπου 100-150 ms σε κινητό μέσω οικιακού Wi-Fi). Δείτε *Buffer ήχου και καθυστέρηση*.

**Ε: Μπορώ να χρησιμοποιήσω πολλαπλά στιγμιότυπα;** Α: Συνήθως ναι, αλλά μπορεί να επιβαρύνει τον διακομιστή. Δείτε κατανόηση.

**Ε: Λειτουργεί εκτός σύνδεσης;** Α: Όχι, το WebSDR απαιτεί σύνδεση στο διαδίκτυο.

**Ε: Ποιοι περιηγητές υποστηρίζονται;** Α: Chrome, Firefox, Edge, Safari (όλες οι πρόσφατες εκδόσεις)

### Ερωτήσεις χρήσης

**Ε: Πόσα άτομα μπορούν να ακούν ταυτόχρονα;** Α: Εξαρτάται από τη χωρητικότητα του διακομιστή. Συχνά 50-200+ χρήστες.

**Ε: Μπορώ να δω τι ακούν οι άλλοι;** Α: Αν είναι ενεργοποιημένο, ναι. Αναζητήστε τις ενδείξεις «άλλοι χρήστες».

**Ε: Μπορώ να συνομιλήσω με άλλους ακροατές;** Α: Αν το έχει ενεργοποιήσει ο διαχειριστής. Αναζητήστε το πλαίσιο συνομιλίας· το **↔ Reply** απαντά σε συγκεκριμένο μήνυμα (δείτε *Συνομιλία παραπάνω*).

**Ε: Γιατί σε ορισμένες συχνότητες δεν εμφανίζεται τίποτα;** Α: Δεν υπάρχουν σήματα σε αυτή τη συχνότητα αυτή τη στιγμή. Δοκιμάστε άλλες!

**Ε: Τι είναι οι χρωματιστές ζώνες στον καταρράκτη;** Α: Η επικάλυψη του σχεδίου μπαντών που δείχνει τις κατανομές συχνοτήτων.

## Πόροι

---

### Μάθετε περισσότερα για τον ραδιοφωνικό κόσμο

- **Σχέδια μπαντών:** αναζητήστε «amateur radio band plan» + την περιοχή σας

- **Διάδοση:** μάθετε για τη διάδοση των ραδιοκυμάτων HF
- **Ψηφιακοί τρόποι:** ερευνήστε τα FT8, PSK31, RTTY
- **Ραδιοερασιτεχνισμός:** σκεφτείτε να αποκτήσετε άδεια ραδιοερασιτέχνη!

## Εύρεση περισσότερων WebSDR

- **Κατάλογος WebSDR:** <http://sdr-list.xyz>
- **WebSDR.org:** <http://websdr.org>
- **KiwiSDR:** <http://kiwisdr.com/public/>

## Λήψη βοήθειας

1. **Διαχειριστής του σταθμού:** δείτε τα στοιχεία επικοινωνίας στη σελίδα
2. **Συνομιλία χρηστών:** ρωτήστε άλλους ακροατές (αν είναι διαθέσιμη)
3. **Διαδικτυακά φόρουμ:** αναζητήστε κοινότητες WebSDR
4. **Τεκμηρίωση:** ανατρέξτε σε αυτόν τον οδηγό!

## Παράρτημα: συνήθειες συχνότητες

### Ραδιοερασιτεχνικές μπάντες HF

Μπάντα	Εύρος συχνοτήτων	Τρόπος	Δραστηριότητα
160 m	1.800-2.000 MHz	LSB	Νύχτα/τοπική
80 m	3.500-4.000 MHz	LSB	Νύχτα/περιφερειακή
40 m	7.000-7.300 MHz	LSB	Ημέρα/νύχτα/DX
30 m	10.100-10.150 MHz	USB	Μόνο δεδομένα/CW
20 m	14.000-14.350 MHz	USB	Ημέρα/DX
17 m	18.068-18.168 MHz	USB	Ημέρα/DX
15 m	21.000-21.450 MHz	USB	Ημέρα/DX
12 m	24.890-24.990 MHz	USB	Ημέρα/DX
10 m	28.000-29.700 MHz	USB	Σποραδική/DX

### Ραδιοερασιτεχνικές μπάντες VHF/UHF

Μπάντα	Εύρος συχνοτήτων	Τρόπος	Δραστηριότητα
6 m	50.000-54.000 MHz	USB/FM	Σποραδική
2 m	144.000-148.000 MHz	FM	Πολύ ενεργή
70 cm	420.000-450.000 MHz	FM	Ενεργή

## Μπάντες ραδιοφωνικών εκπομπών

Υπηρεσία	Εύρος συχνοτήτων	Τρόπος
Ραδιόφωνο AM	530-1710 kHz	AM
Βραχεία κύματα	2.3-26.1 MHz	AM
Ραδιόφωνο FM	88-108 MHz	WBFM

## Αεροπορία

Υπηρεσία	Εύρος συχνοτήτων	Τρόπος
Έλεγχος εναέριας κυκλοφορίας	118-137 MHz	AM
ACARS (δεδομένα)	130-136 MHz	Δεδομένα

## Ναυτιλία

Υπηρεσία	Εύρος συχνοτήτων	Τρόπος
Ναυτιλιακά VHF	156-162 MHz	FM
Ναυτιλιακά HF	2-22 MHz	USB

## Γλωσσάρι

**AGC:** αυτόματος έλεγχος κέρδους – ρυθμίζει αυτόματα τις στάθμες ήχου

**AM:** διαμόρφωση πλάτους – τρόπος φωνής για την αεροπορία και τις ραδιοφωνικές εκπομπές

**Εύρος ζώνης:** η περιοχή συχνοτήτων ενός σήματος

**CW:** συνεχές κύμα – σήματα κώδικα Μορς

**DX:** επικοινωνία μεγάλης απόστασης

**FFT:** ταχύς μετασχηματισμός Φουριέ – μετατρέπει το πεδίο του χρόνου σε πεδίο συχνοτήτων

**FM:** διαμόρφωση συχνότητας – τρόπος φωνής για VHF/UHF

**HF:** υψηλές συχνότητες (3-30 MHz) – μπάντες μεγάλων αποστάσεων

**kHz:** κιλοχέρτζ (1.000 Hz)

**LSB:** κάτω πλευρική ζώνη – τρόπος φωνής για τις χαμηλότερες μπάντες HF

**MHz:** μεγαχέρτζ (1.000.000 Hz)

**NB:** κόπτης κρουστικού θορύβου – αφαιρεί τον κρουστικό θόρυβο

**NR:** μείωση θορύβου – μειώνει τον θόρυβο υποβάθρου

**PSK:** διαμόρφωση μετατόπισης φάσης – ψηφιακός τρόπος

**RTTY:** ραδιοτηλετυπία – ψηφιακός τρόπος κειμένου

**S-meter:** όργανο ένδειξης ισχύος σήματος

**SDR:** ραδιόφωνο καθορισμένο από λογισμικό

**SQL:** squelch – σιγάζει τον ήχο όταν δεν υπάρχει σήμα

**SSB:** μονή πλευρική ζώνη (USB ή LSB)

**USB:** άνω πλευρική ζώνη – τρόπος φωνής για τις υψηλότερες μπάντες HF

**VHF:** πολύ υψηλές συχνότητες (30-300 MHz) – οπτική επαφή

**UHF:** υπερυψηλές συχνότητες (300-3000 MHz) – οπτική επαφή

**Καταρράκτης:** οπτική απεικόνιση του ραδιοφωνικού φάσματος στον χρόνο

---

**Καλή εξερεύνηση του ραδιοφωνικού φάσματος με το PhantomSDR-Plus!**

**73 (θερμούς χαιρετισμούς) de SV1BTL & SV2AMK**

Για οδηγίες εγκατάστασης, δείτε το [INSTALLATION.md](#). Για τεχνικές λεπτομέρειες, δείτε το [PROJECT\\_STRUCTURE.md](#).

## ΚΕΦΑΛΑΙΟ 13

## Εξομοίωση πελάτη KiwiSDR

Aether\_config.md

## Εξομοίωση πελατών KiwiSDR

## Σύνδεση του AetherSDR, του kiwiclient και άλλου λογισμικού KiwiSDR στο PhantomSDR-Plus

Από την έκδοση v4.1.0 το PhantomSDR-Plus μπορεί να απαντά και στο **πρωτόκολλο KiwiSDR**, ώστε λογισμικό γραμμένο για KiwiSDR — το **AetherSDR**, το **kiwiclient** και τα υπόλοιπα — να συνδέεται απευθείας στον δέκτη σας. Πρόκειται για μια γέφυρα μέσα στην ίδια διεργασία του διακομιστή, στον ίδιο υπολογιστή και στην ίδια θύρα που ήδη δημοσιεύετε: χωρίς δεύτερη υπηρεσία, χωρίς δεύτερη θύρα, χωρίς proxy.

**Είναι απενεργοποιημένη μέχρι να την ενεργοποιήσετε.** Τίποτα από τη γέφυρα δεν τρέχει — ούτε socket, ούτε έλεγχος — όσο το `[kiwi_emulation]` λείπει ή είναι `false`, που είναι η κατάσταση ενός φρέσκου δέντρου.

## Πίνακας περιεχομένων

1. Τι παίρνει ένας πελάτης Kiwi
2. Εγκατάσταση της γέφυρας
3. Ενεργοποίηση
4. Αναφορά ρυθμίσεων
5. Σύνδεση ενός πελάτη
6. Στάθμη ήχου
7. Το S-meter
8. Καταρράκτης και φάσμα
9. Αν κάτι δεν φαίνεται σωστό
10. Μέχρι πού έχει επαληθευτεί

## 1. Τι παίρνει ένας πελάτης Kiwi

Όνομα	Περιγραφή
Ήχος	<code>ws://&lt;host&gt;:&lt;port&gt;/kiwi/&lt;id&gt;/SND</code> — αποδιαμορφωμένος ήχος στην πλαίσωση SND του Kiwi
Καταρράκτης	<code>ws://&lt;host&gt;:&lt;port&gt;/kiwi/&lt;id&gt;/W/F</code> — το φάσμα, σε πλαίσια W/F του Kiwi
Συντονισμός	πραγματικός, όχι διακοσμητικός: το <code>SET mod=...</code> από τον πελάτη αλλάζει συχνότητα, πλευρική ζώνη και εύρος διέλευσης στην πλευρά του PhantomSDR, ακριβώς όπως θα έκανε ένας ακροατής από τον browser
S-meter	οδηγείται από την ισχύ ανά μπλοκ του ίδιου του αποδιαμορφωτή, στην ίδια κλίμακα με το S-meter της ιστοσελίδας

Ο συντονισμός υποστηρίζεται για **πραγματική** είσοδο (`signal = "real"` στο `.toml` σας). Για είσοδο IQ η σχέση bin προς συχνότητα διαφέρει και δεν έχει υλοποιηθεί, οπότε ένας δέκτης IQ θα δίνει ήχο και καταρράκτη αλλά δεν θα ακολουθεί τον συντονισμό ενός πελάτη Kiwi.

## 2. Εγκατάσταση της γέφυρας

Η γέφυρα είναι ένα σύνολο διορθώσεων στις πηγές του backend συν ένα νέο αρχείο κεφαλίδας, το `src/kiwi_bridge.h`. Υπάρχουν δύο τρόποι να μπουν.

**Με το πρόγραμμα εγκατάστασης.** Το `install.sh` (και τα `install_arch.sh`, `install_fedora.sh`, `install_opensuse.sh`) έχουν δικό τους βήμα γι' αυτό — το βήμα 17, *Kiwi client emulation*. Προσφέρεται, δεν επιβάλλεται: απαντήστε `n` και δεν αλλάζει τίποτα. Για ανεπιτήρητη εκτέλεση αποφασίζει το `PHANTOM_KIWI=y|n`, με προεπιλογή το `ναι`.

**Σε δέντρο που είναι ήδη εγκατεστημένο,** τρέξτε το script που συνοδεύει τη ρίζα του αποθετηρίου:

```
cd ~/PhantomSDR-Plus
./kiwi_install.sh
./recompile.sh # επιλέξτε [1] Μόνο backend
```

Το `kiwi_install.sh` κρατά αντίγραφο κάθε αρχείου που αγγίζει στο `backup_kiwi_bridge_<χρονοσήμανση>/` και εφαρμόζει κάθε διόρθωση με ακριβή αντιστοίχιση κειμένου — αν δεν βρει μια άγκυρα εκεί που την περιμένει (τοπικά τροποποιημένο δέντρο, άλλη έκδοση), σταματά εκεί, ονομάζει το αρχείο και το αφήνει ανέγγιχτο. Είναι ταυτοδύναμο: σε δέντρο που έχει ήδη τη γέφυρα αναφέρει κάθε διόρθωση ως ήδη εφαρμοσμένη και δεν αλλάζει τίποτα.

**Αν μόλις εφαρμόσατε αρχείο ενημέρωσης που περιέχει ήδη τη γέφυρα, δεν χρειάζεται να το τρέξετε.** Οι διορθωμένες πηγές είναι μέσα στο πακέτο· το `kiwi_install.sh` απλώς θα σας πει ότι κάθε διόρθωση υπάρχει ήδη.

Τα αρχεία που διορθώνει είναι τα `src/client.h`, `src/signal.cpp`, `src/waterfall.cpp`, `src/spectrumserver.h`, `src/spectrumserver.cpp`, `src/websocket.cpp` και `src/http.cpp`, και αντιγράφει το `kiwi_bridge.h` στο `src/`.

**Εγκαθιστά τη γέφυρα, δεν την αναβαθμίζει.** Αν κάποια διόρθωση αποτύχει σε δέντρο όπου η γέφυρα ήδη δουλεύει, ο κώδικας έχει προχωρήσει μπροστά από το script — και το script χρειάζεται ενημέρωση, όχι το δέντρο σας.

## 3. Ενεργοποίηση

Προσθέστε το εξής στο `.toml` με το οποίο τρέχει πραγματικά ο δέκτης σας — το ίδιο αρχείο που δίνετε στον `spectrumserver`, αυτό που αναφέρεται στο `start-<radio>.sh` σας. Όχι στο `config.example.*`:

```
[kiwi_emulation]
enabled = true
```

Έπειτα επανεκκινήστε τον δέκτη:

```
./stop-websdr.sh
./start-<ο-δεκτης-σας>.sh
```

Το `kiwi_install.sh` προσθέτει το μπλοκ για εσάς, με `enabled = true`, σε όποια από τα `config.toml`, `config-rx888mk2.toml`, `config-airspyhf.toml`, `config-rtl.toml`, `config-rspla.toml`, `config-fobos.toml`, `config-fobos-hf.toml` και `config-hackrf.toml` υπάρχουν στη ρίζα του αποθετηρίου. Δεν μπορεί να ξέρει για ρύθμιση που κρατάτε αλλού ή με άλλο όνομα, και **τα αρχεία ενημέρωσης δεν στέλνουν ποτέ .toml**, οπότε μετά από ενημέρωση το μπλοκ το βάζετε εσείς. Τα `config.example.*.toml` που συνοδεύουν το έργο το έχουν τεκμηριωμένο και σε `false`.

## 4. Αναφορά ρυθμίσεων

Ό,τι διαβάζει η γέφυρα βρίσκεται σε μία ενότητα `[kiwi_emulation]`. Μόνο το `enabled` είναι υποχρεωτικό· και οι τέσσερις ρυθμίσεις ακριβείας έχουν ως προεπιλογή τη βαθμονομημένη τιμή, οπότε ένα μπλοκ με μόνο `enabled = true` είναι ήδη σωστό.

Κλειδί	Προεπιλογή	Τι κάνει
<code>enabled</code>	<code>false</code>	Απάντηση στο πρωτόκολλο KiwiSDR. Απενεργοποιημένο = εντελώς αδρανές.
<code>audio_gain</code>	0	dB. Ενίσχυση εξόδου, μόνο για πελάτες Kiwi. Βλ. <a href="#">Στάθμη ήχου</a> .
<code>smeter_offset</code>	<code>input.analog_smeter_offset</code>	dB. Παρακάμπτει τη μετατόπιση του S-meter, μόνο για πελάτες Kiwi. Βλ. <a href="#">To S-meter</a> .
<code>wf_cal</code>	0	dB. Ρύθμιση εμφάνισης για καταρράκτη και φάσμα. Βλ. <a href="#">Καταρράκτης και φάσμα</a> .
<code>wf_fps_max</code>	23	Καρέ καταρράκτη ανά δευτερόλεπτο. Βλ. <a href="#">Καταρράκτης και φάσμα</a> .

Ένα πλήρες μπλοκ, με τις τιμές που τρέχει ο δέκτης αυτού του έργου:

```
[kiwi_emulation]
enabled = true
audio_gain = 60.0 # dB, μόνο πελάτες Kiwi. 0 = ανέπαφο.
wf_cal = -10.0 # dB, θέμα γούστου. 0 = συμφωνεί με το S-meter.
wf_fps_max = 28 # καρέ/s. Το 23 είναι το μέγιστο του πρωτοκόλλου.
smeter_offset = 5.0 # dB. Προεπιλογή είναι το input.analog_smeter_offset.
```

**Κάθε κλειδί εδώ διαβάζεται κατά την εκκίνηση.** Η αλλαγή τους είναι **επανεκκίνηση**, όχι επαναμεταγλώττιση — `./stop-websdr.sh` και μετά το script εκκίνησής σας. Δεν χρειάζεται ποτέ να μεταγλωττίσετε ξανά για να τα ρυθμίσετε.

Δεκτές είναι τόσο η ακέραιη όσο και η δεκαδική μορφή: το 28 και το 28.0 διαβάζονται το ίδιο. Μόνο ένα κλειδί που πραγματικά λείπει πέφτει στην προεπιλογή του.

## 5. Σύνδεση ενός πελάτη

Στρέψτε τον πελάτη στην ίδια διεύθυνση και θύρα που χρησιμοποιούν οι ακροατές σας — χωρίς διαδρομή, χωρίς πρόθεμα. Το AetherSDR ζητά υπολογιστή και θύρα· το `kiwiclnt` παίρνει `-s` και `-p`:

```
kiwiclnt, καταγραφή 30 δευτερολέπτων στους 7100 kHz LSB
python3 kiwirecorder.py -s o.dektis.sas.example -p 8073 -f 7100 -m lsb --tlimit=30
```

Ο πελάτης συντονίζεται, ο καταρράκτης γεμίζει και το S-meter δείχνει. Αν δεν συμβαίνει απολύτως τίποτα, η γέφυρα είναι σχεδόν σίγουρα ακόμη απενεργοποιημένη — βλ. [ενότητα 3](#).

## 6. Στάθμη ήχου

Το πρώτο που προσέχουν οι περισσότεροι είναι ότι ένας πελάτης Kiwi ακούγεται πιο ισχνός από την ιστοσελίδα του PhantomSDR. Δεν χάνει τίποτα η γέφυρα: ένας κοινός buffer τροφοδοτεί κάθε

κωδικοποιητή, και η ίδια συχνότητα με το ίδιο εύρος διέλευσης, μετρημένη μέσα από τη διαδρομή ήχου του browser και μέσα από το /SND, δίνει πανομοιότυπα δείγματα. Η ιστοσελίδα ακούγεται δυνατά επειδή το `audio.js` ξαναχτίζει τον ήχο μέσα στον browser — ενίσχυση μπάσων, ζωνοπερατό φίλτρο, ανάδειξη παρουσίας, συμπίεστής με `makeup gain` και το ποτενσιόμετρο έντασης. Ένας πελάτης Kiwi δεν έχει τίποτα από αυτά και δεν μπορεί να αποκτήσει.

Το `audio_gain` κλείνει το κενό, μόνο για πελάτες Kiwi:

```
[kiwi_emulation]
enabled = true
audio_gain = 55.0 # dB, μόνο πελάτες Kiwi. 0 = ανέπαφο.
```

**Ξεκινήστε γύρω στα 55-60 dB.** Τα 55 dB δεν είναι αριθμητικό αποτέλεσμα: εκεί μετρήθηκαν στην ίδια στάθμη ένας πελάτης Kiwi και η ιστοσελίδα του PhantomSDR, σε ισχυρό τοπικό σήμα στους 729 kHz. Στην πράξη συχνά προτιμώνται λίγο περισσότερα — αυτός ο δέκτης έμεινε στα 60 — οπότε διαλέξτε με το αυτί σας.

Δεν μπορείτε να χαλάσετε τον ήχο με αυτό. Ένας **περιοριστής κορυφών με πρόβλεψη** βρίσκεται ανάμεσα στην ενίσχυση και στον περιορισμό των 16 bit: κάθε δείγμα καθυστερεί 4 ms όσο η ενίσχυση υπολογίζεται από ήχο που δεν έχει ακόμη σταλεί, ώστε ένα δυνατό πέρασμα να συναντά ενίσχυση που έχει ήδη κατέβει γι' αυτό, και οι κορυφές να διπλώνουν αντί να ισοπεδώνονται. Οδηγημένο στα 60 και ακόμη στα 70 dB — πολύ πέρα από το περιθώριο της σκέτης ενίσχυσης — δεν ψαλιδίστηκε ούτε ένα δείγμα στα τετρακόσιες χιλιάδες. Κάτω από το κατώφλι του περιοριστή η ενίσχυση είναι ακριβώς 1,0, οπότε ένας δέκτης που δεν πιέζει τον ήχο παίρνει τα δείγματα ανέπαφα.

Μια πολύ υψηλή τιμή κοστίζει ένταση, όχι ποιότητα. Πάνω από τα 60 dB περίπου, η επιπλέον ενίσχυση απλώς περιορίζεται: μετρημένα εδώ, η στάθμη ανέβηκε περίπου 1 dB από τα 55 στα 60 και σχεδόν δεν κουνήθηκε από τα 60 στα 70. Πέρα από εκεί αγοράζετε συμπίεση, όχι ένταση.

Τίποτα από αυτά δεν αγγίζει την ιστοσελίδα, και τίποτα δεν μετακινεί το S-meter — αυτό προκύπτει από την ισχύ του αποδιαμορφωτή, όχι από τα δείγματα ήχου, οπότε ανεβάζοντας την ένταση δεν μπορείτε να κάνετε το όργανο να πει ψέματα.

## 7. To S-meter

Το S-meter του Kiwi δεν βαθμονομείται χωριστά: αναπαράγει ό,τι δείχνει η ιστοσελίδα σας, στάδιο προς στάδιο. Η ισχύς ανά μπλοκ του αποδιαμορφωτή είναι η αφητηρία και για τα δύο, και μετά η σελίδα εφαρμόζει το `input.analog_smeter_offset`, διαστέλλει το αποτέλεσμα γύρω από τα -130 dBm και προσθέτει τη δική της μετατόπιση εμφάνισης. Η γέφυρα κάνει το ίδιο, οπότε ένας πελάτης Kiwi και η σελίδα δείχνουν το ίδιο σήμα με την ίδια ένταση — επαληθευμένο σε ισχυρό τοπικό σταθμό, όπου και τα δύο έδειξαν -37 dBm.

Αν, και μόνο αν, θέλετε οι πελάτες Kiwi να δείχνουν διαφορετικά από τη σελίδα σας, δώστε στη γέφυρα δική της μετατόπιση:

```
[kiwi_emulation]
enabled = true
smeter_offset = 5.0 # αντικαθιστά το input.analog_smeter_offset μόνο για πελάτες Kiwi
```

**Συγκρίνετε τα δύο όργανα με το ίδιο φίλτρο, αλλιώς καθόλου.** Η ένδειξη προκύπτει από την ισχύ μέσα στο εύρος διέλευσης, οπότε ένα πλατύτερο φίλτρο μαζεύει περισσότερο θόρυβο και δείχνει υψηλότερα — σε ήσυχη μπάντα η ίδια συχνότητα μετρήθηκε 5,6 dB ισχυρότερη στα 9 kHz απ' ό,τι στα 2,4 kHz. Ένας πελάτης Kiwi με φίλτρο 6 kHz απέναντι σε ιστοσελίδα στα 2,4 kHz θα διαφωνεί κατά αρκετά dB όπως και να είναι βαθμονομημένα. Ταιριάξτε πρώτα τον τρόπο εκπομπής και το εύρος.

**Αυτή είναι η κλίμακα της σελίδας, όχι φυσική.** Το όργανο της σελίδας διαστέλλει το εύρος του για ευκρίνεια, ώστε 10 dB πραγματικής μεταβολής να εμφανίζονται ως περίπου 11. Το να την ταιριάξουμε σημαίνει ότι οι πελάτες Kiwi συμφωνούν με τον δέκτη σας και διαφέρουν από έναν γνήσιο KiwiSDR κατά ποσό που μεγαλώνει με την ένταση του σήματος. Αυτός είναι ο σωστός συμβιβασμός όταν η δική σας σελίδα είναι το μέτρο σύγκρισης για όλους· αν προτιμάτε η γέφυρα να μένει φυσικά τίμια, αυτή είναι η ενότητα που αλλάζετε.

## 8. Καταρράκτης και φάσμα

### Η κλίμακα dB — `wf_cal`

Κάθε bin του καταρράκτη μετατρέπεται σε πραγματικά dBm πριν σταλεί, στην ίδια κλίμακα με την οποία είναι βαθμονομημένο το S-meter. Ένα φέρον διαβάζεται επομένως **στην ίδια στάθμη σε κάθε ζουμ**, όπως ακριβώς δίνει ένας γνήσιος KiwiSDR, και το φάσμα συμφωνεί με το S-meter της ιστοσελίδας πάνω στο ίδιο εύρος διέλευσης.

```
[kiwi_emulation]
enabled = true
wf_cal = 0.0 # dB, μόνο πελάτες Kiwi
```

**Το `wf_cal` δεν είναι βαθμονόμηση.** Το 0 είναι η τιμή στην οποία το φάσμα συμφωνεί με το S-meter σας, και εκεί πρέπει να μέινει αν θέλετε οι αριθμοί να σημαίνουν κάτι. Υπάρχει επειδή το πόσο δυνατό πρέπει να δείχνει ένα φάσμα είναι θέμα γούστου και του τι κάνει ο πελάτης σας με το εύρος — ο δέκτης αυτού του έργου τρέχει με -10 απλώς επειδή έτσι φαίνεται σωστό στο AethersDR. Μία τιμή μετακινεί μαζί φάσμα και καταρράκτη: είναι τα ίδια bytes στη γραμμή και το πρωτόκολλο Kiwi δεν μπορεί να τα κλιμακώσει χωριστά. Αν θέλετε να διαφέρουν, αυτό πρέπει να έρθει από τα χειριστήρια εμφάνισης του ίδιου του πελάτη.

Για να ελέγξετε την κλίμακα, συγκρίνετε τον καταρράκτη αθροισμένο πάνω στο εύρος διέλευσης με το S-meter σε μια **ήσυχη** συχνότητα. Πάνω σε φέρον, ένα εύρος SSB αποκλείει το ίδιο το φέρον και τα δύο δεν συγκρίνονται.

### Ο ρυθμός καρτέ — `wf_fps_max`

Ένας πελάτης Kiwi ζητά τον γρηγορότερο καταρράκτη που μπορεί να έχει (`SET wf_speed=4`) και ρυθμίζει την κύλισή του — και τη μέση τιμή του φάσματος — στον ρυθμό που ο διακομιστής λέει ότι θα δώσει. Αν οι δύο διαφωνούν, η εικόνα φαίνεται νωθρή αν και τίποτα δεν καθυστερεί πραγματικά.

Ο δέκτης παράγει  $2 \times \text{sps} / \text{fft\_size}$  φάσματα το δευτερόλεπτο. Οι πελάτες Kiwi τροφοδοτούνται από καθένα από αυτά και αραιώνονται στον ρυθμό που ζήτησαν· η ιστοσελίδα κρατά τον δικό της, πιο αργό ρυθμό και δεν επηρεάζεται.

```
[kiwi_emulation]
enabled = true
wf_fps_max = 23 # καρτέ ανά δευτερόλεπτο
```

Το 23 είναι το μέγιστο που δηλώνει το ίδιο το πρωτόκολλο KiwiSDR και η ασφαλής προεπιλογή.

**Κερδίζει πάντα η μικρότερη από τις τιμές `wf_fps_max` και τον ρυθμό του ίδιου του δέκτη,** οπότε η αύξηση κάνει κάτι μόνο αν η FFT σας είναι αρκετά γρήγορη ώστε να περισσεύουν καρέ.

Παράδειγμα για δέκτη στα 60 MS/s με FFT 4194304 σημείων —  $2 \times 60000000 / 4194304 = 28,6$  καρέ το δευτερόλεπτο:

<code>wf_fps_max</code>	παραδίδεται
απουσιάζει (προεπιλογή 23)	23,0 fps
28	28,0 fps
40	28,6 fps — κερδίζει το όριο του δέκτη

Τηρείται και το SET `wf_speed` από τον πελάτη: 0 κλειστό, 1 = 1 fps, 2 το ένα τέταρτο του μεγίστου, 3 το μισό, 4 το μέγιστο. Ένας πελάτης σε αδύναμη γραμμή μπορεί έτσι να ζητήσει λιγότερα.

**Το αν βοηθά η αύξηση το αποφασίζει ο πελάτης, όχι ο διακομιστής.** Ένας πελάτης που σχεδιάζει κάθε καρέ μόλις φτάνει σας δίνει πιο ομαλή κύλιση· ένας πελάτης που ρυθμίζει τον ρυθμό του στα 23 που περιμένει απλώς θα βάλει τα επιπλέον καρέ σε ουρά, και η καθυστέρηση θα μεγαλώσει. Δοκιμάστε το, και αν ο καταρράκτης μοιάζει πιο νωθρός αντί για πιο ομαλός, γυρίστε το πίσω στο 23.

## Τι καθορίζει την υπόλοιπη καθυστέρηση

Αν κυνηγάτε την καθυστέρηση, το μεγαλύτερο μέρος της δεν βρίσκεται στη γέφυρα. Το παράθυρο ανάλυσης έχει πλάτος `fft_size / sps` — 70 ms σε FFT 4194304 σημείων στα 60 MS/s — και ένα δείγμα περιμένει επιπλέον έως το μισό αυτού μέχρι να γεμίσει το βήμα. Το να μισέψετε το `fft_size` μισεύει και τα δύο και διπλασιάζει τον ρυθμό καρέ, αλλά μισεύει και τη φασματική σας ανάλυση, από την οποία ζουν οι στενής ζώνης αποκωδικοποιητές (WSPR, FT8, CW). Στους περισσότερους δέκτες αυτό είναι κακή ανταλλαγή για μερικές δεκάδες χιλιοστά του δευτερολέπτου. Είναι απόφαση στο `[input]` και όχι κάτι που μπορεί να αλλάξει η γέφυρα.

## 9. Αν κάτι δεν φαίνεται σωστό

Κάθε εντολή που στέλνει ένας πελάτης Kiwi γράφεται στο `/tmp/kiwi_retune.log` — τα αιτήματα συντονισμού, οι αλλαγές τρόπου εκπομπής, η πιστοποίηση και ό,τι δεν αναγνώρισε η γέφυρα. Δεν κοστίζει τίποτα μετρήσιμο και προορίζεται να μείνει εκεί μόνιμα. Όταν ένας πελάτης συμπεριφέρεται περίεργα, αυτό το αρχείο δείχνει τι ζήτησε πραγματικά — που είναι συνήθως όλη η απάντηση.

Σύμπτωμα	Κοιτάξτε
Ο πελάτης συνδέεται και πέφτει αμέσως	λείπει το <code>[kiwi_emulation] enabled = true</code> από τη ρύθμιση με την οποία <b>ξεκίνησε</b> ο διακομιστής
Ούτε ήχος, ούτε καταρράκτης, ούτε γραμμές στο log	το backend δεν ξαναμεταγλωττίστηκε μετά το <code>kiwi_install.sh</code> — τρέξτε <code>./recompile.sh</code> , επιλογή [1]
Το <code>kiwi_install.sh</code> σταματά σε μια διόρθωση	το μήνυμα ονομάζει το αρχείο και την άγκυρα· το δέντρο έχει αποκλίνει από ό,τι περιμένει η διόρθωση, και το αρχείο έμεινε ανέγγιχτο
Υπάρχει ήχος αλλά ισχνός και χαμηλός	αναμενόμενο — ρυθμίστε το <code>audio_gain</code> , βλ. <a href="#">ενότητα 6</a>
Ο καταρράκτης κυλά νωθρά	<code>wf_fps_max</code> , βλ. <a href="#">ενότητα 8</a>
Ο συντονισμός στον πελάτη δεν κάνει τίποτα	είσοδος IQ ( <code>signal = "iq"</code> )· ο συντονισμός υλοποιείται μόνο για πραγματικές εισόδους
Μια αλλαγμένη ρύθμιση δεν είχε αποτέλεσμα	αυτά τα κλειδιά διαβάζονται κατά την εκκίνηση — επανεκκινήστε τον δέκτη

**Το AGC είναι υπόθεση του πελάτη.** Το `SET agc=` από πελάτη Kiwi γίνεται δεκτό και αγνοείται — ο ήχος φτάνει με το AGC του ίδιου του PhantomSDR εφαρμοσμένο, και τα χειριστήρια AGC του πελάτη δρουν πάνω σε ό,τι λαμβάνει.

## 10. Μέχρι πού έχει επαληθευτεί

Η μορφή στη γραμμή έχει ελεγχθεί απέναντι σε ζωντανό δέκτη και διαβαστεί γραμμή προς γραμμή απέναντι στον κώδικα των δύο πελατών στους οποίους απευθύνεται — το `KiwiSdrProtocol.cpp` του AetherSDR και το `kiwi/client.py` του `kiwiclient`. Η διάταξη και το μέγεθος των πλαισίων, οι αριθμοί ακολουθίας, η σειρά των bytes, η κλίμακα του S-meter, ο συντονισμός και ο καταρράκτης συμφωνούν όλα. Η γέφυρα έχει επίσης **επιβεβαιωθεί ότι δουλεύει με την ίδια την εφαρμογή AetherSDR** για υπολογιστή, από τον συντηρητή της, απέναντι σε αυτόν τον δέκτη.

Μετρημένα σε αυτόν τον δέκτη, με πελάτη δοκιμής στις ζωντανές συνδέσεις: ο συντονισμός εφαρμόζεται σε 35–50 ms, ο ήχος τρέχει σε πραγματικό χρόνο χωρίς ολίσθηση, και ο καταρράκτης δίνει τον ρυθμό που έχει ρυθμιστεί χωρίς κολλήματα και χωρίς χαμένα καρέ.

Αν παρ' όλα αυτά κάτι δείχνει λάθος στη δική σας εγκατάσταση, το `/tmp/kiwi_retune.log` είναι το πρώτο μέρος να κοιτάξετε, και μια αναφορά είναι ευπρόσδεκτη έτσι κι αλλιώς.

**Δείτε επίσης:** [Οδηγός εγκατάστασης](#) · [Δομή του έργου](#) · [Οδηγός χρήσης](#)

# Διαφορική λήψη

RECEIVE\_DIVERSITY.md

## PhantomSDR-Plus — Διαφορική λήψη (Receive Diversity)

Συνδυάστε τον δέκτη σας με **έναν δεύτερο δέκτη σε άλλη τοποθεσία** και ακούστε όποιον από τους δύο έχει εκείνη τη στιγμή το καλύτερο σήμα. Όταν η μία τοποθεσία πέφτει σε fading, η άλλη συνήθως όχι, και η λήψη συνεχίζεται.

Ο δεύτερος δέκτης μπορεί να είναι ένας άλλος PhantomSDR-Plus, ένας KiwiSDR, ένας UberSDR ή ένας WebSDR. Για τους τρεις πρώτους όλα εκτελούνται στον browser: καμία αλλαγή στον server, κανένα αρχείο ρυθμίσεων, τίποτα προς εγκατάσταση και καμία ειδική πρόσβαση στον άλλο δέκτη. Ο WebSDR είναι η εξαίρεση και χρειάζεται ένα μικρό πρόγραμμα στον δικό σας server — δείτε [Χρήση WebSDR](#) παρακάτω.

### Τι κάνει και τι δεν κάνει

Εδώ «diversity» σημαίνει **επιλογή**: κάθε στιγμή ακούτε τη μία ή την άλλη τοποθεσία, με ένα σύντομο crossfade στην εναλλαγή. Δεν είναι phased array και δεν προσθέτει τα δύο σήματα μεταξύ τους.

Αυτή είναι συνειδητή επιλογή, και ο λόγος είναι η φυσική, όχι το λογισμικό. Δύο δέκτες σε απόσταση εκατοντάδων χιλιομέτρων ακούν την ίδια εκπομπή μέσα από διαφορετικές ιονοσφαιρικές διαδρομές. Οι δύο κυματομορφές φτάνουν με άσχετη φάση και ελαφρώς διαφορετικό Doppler, οπότε η πρόσθεσή τους ακούγεται κούφια και «χτενισμένη» — το κλασικό phasey αποτέλεσμα. Η πραγματική σύμφωνη σύνθεση απαιτεί δύο δέκτες σε **κοινό ρολόι**, ευθυγραμμισμένους στο δείγμα· δύο ανεξάρτητοι δέκτες μέσω internet δεν μπορούν να το προσφέρουν ποτέ.

Αυτό που κερδίζετε είναι λοιπόν **συνέχεια, όχι ένταση σήματος**:

- Σε σταθερή διαδρομή όπου ο ένας δέκτης είναι απλώς καλύτερος, θα ακούτε εκείνον και δεν θα κερδίσετε τίποτα. Αυτό είναι φυσιολογικό.
- Σε διαδρομή με έντονο QSB, τα fadings σε δύο απομακρυσμένες τοποθεσίες είναι σε μεγάλο βαθμό ασυσχέτιστα. Η τοποθεσία Α χάνεται για τρία δευτερόλεπτα, η Β όχι, και ο ήχος συνεχίζεται. Περιμένετε «*έπαψα να χάνω λέξεις*», όχι «*από S3 έγινε S7*».
- Το δεύτερο όφελος είναι συχνά ο **τοπικός θόρυβος και τα QRM**. Δύο τοποθεσίες έχουν διαφορετικούς γείτονες, διαφορετικά δίκτυα ρεύματος και διαφορετικό splatter. Σε θορυβώδη μπάνα αυτό μπορεί να μετράει περισσότερο από το fading.

Υπάρχει κόστος: ακούτε **ένα έως τρία δευτερόλεπτα πίσω από τον πραγματικό χρόνο**, επειδή η ροή που προηγείται πρέπει να καθυστερήσει ώστε να συμπέσει με την άλλη. Αδιάφορο για ακρόαση και αποκωδικοποίηση, ενοχλητικό αν προσπαθείτε να κάνετε QSO.

#### ΣΗΜΕΙΩΣΗ

Οι ψηφιακοί αποκωδικοποιητές (FT8, JS8, WSPR, RADE κ.λπ.) χρησιμοποιούν σκόπιμα τον **τοπικό** σας δέκτη, όχι τον συνδυασμένο ήχο. Αυτά τα modes ολοκληρώνουν σύμφωνα σε ολόκληρη τη χρονοθυρίδα, και μια εναλλαγή τοποθεσίας στη μέση της θυρίδας είναι ασυνέχεια φάσης που μπορεί να σας στοιχίσει ακριβώς την αποκωδικοποίηση που το diversity ήθελε να σώσει. Το diversity εξυπηρετεί το ηχείο· οι αποκωδικοποιητές κρατούν τη συνεχή τοπική ροή.

## Γρήγορη εκκίνηση

1. Ανοίξτε τον πίνακα **Receive Diversity** — βρίσκεται ακριβώς κάτω από τα παράθυρα των αποκωδικοποιητών, κλειστός, με την ένδειξη **Receive Diversity** — off. Κάντε κλικ για να ανοίξει.
2. Επιλέξτε τι είναι ο δεύτερος δέκτης: **PhantomSDR+**, **KiwiSDR**, **UberSDR** ή **WebSDR**.
3. Πληκτρολογήστε τη διεύθυνσή του — ή επιλέξτε μία από τις αποθηκευμένες — και πατήστε **Start**.
4. Συντονίστε κανονικά τον δικό σας δέκτη. Ο δεύτερος **ακολουθεί αυτόματα** — συχνότητα, mode και passband — σε κάθε αλλαγή συντονισμού.

Αυτή είναι όλη η ρύθμιση. Δεν χρειάζεται τίποτα στον άλλο δέκτη και δεν υπάρχει ξεχωριστός συντονισμός: σας ακολουθεί πάντα.

## Διευθύνσεις

Δεύτερος δέκτης	Τι πληκτρολογείτε	Σημειώσεις
PhantomSDR+	host:8900	Η κανονική θύρα του δέκτη
KiwiSDR	host:8073	Η θύρα του KiwiSDR
UberSDR	host	Η <b>κανονική θύρα web</b> , όχι η 8073
WebSDR	host:8901	Η θύρα του ίδιου του σταθμού. Χρειάζεται το relay — δείτε παρακάτω

Γίνονται δεκτά σκέτο hostname, διεύθυνση `http://` ή πλήρες `ws://`. Αν ο άλλος δέκτης σερβίρεται μέσω HTTPS, γράψτε το hostname και θα χρησιμοποιηθεί ασφαλής σύνδεση.

## Αποθηκευμένοι δέκτες

Κάθε διεύθυνση που χρησιμοποιείτε απομνημονεύεται, σε ξεχωριστή λίστα για κάθε τύπο δέκτη — μια διεύθυνση KiwiSDR δεν είναι ποτέ χρήσιμη πρόταση όταν ο επιλογέας δείχνει WebSDR. Οι λίστες μένουν στον δικό σας browser: δεν στέλνονται πουθενά και κάθε ακροατής έχει τη δική του.

Το κουμπί  δίπλα στο πλαίσιο της διεύθυνσης ανοίγει τη λίστα.

- **Ονόματα.** Κάθε καταχώρηση μπορεί να πάρει όνομα — το `Twente` διαβάζεται καλύτερα από το `websdr.ewi.utwente.nl:8901`. Οι ονομασμένες καταχωρήσεις εμφανίζονται με πράσινο πάνω από τη διεύθυνσή τους, και το όνομα φαίνεται δίπλα στη διεύθυνση στην αναδυόμενη λίστα καθώς πληκτρολογείτε. Μια καταχώρηση ταυτοποιείται από τη διεύθυνσή της, οπότε το όνομα δεν βάζει ποτέ τον ίδιο σταθμό δύο φορές στη λίστα. Όσο τρέχει η διαφορική λήψη, ο δέκτης με τον οποίο είναι συνδεδεμένη δείχνει το όνομά του **έντονο και να αναβοσβήνει αργά**, ώστε η λίστα να λείει και ποιον ακούτε πραγματικά.
- Το **add** προσθέτει δέκτη με το χέρι, χωρίς να συνδεθείτε πρώτα σε αυτόν. Το  διορθώνει όνομα και διεύθυνση επί τόπου, το **x** αφαιρεί μία καταχώρηση και το **clear all** αδειάζει τη λίστα του τύπου που βλέπετε. Το Enter αποθηκεύει, το Escape ακυρώνει.
- Το **\*\*\*\*** ανοίγει την ίδια τη σελίδα εκείνου του δέκτη σε νέα καρτέλα. Η αποθηκευμένη διεύθυνση είναι αυτή που καλεί το λογισμικό, οπότε μετατρέπεται πρώτα σε διεύθυνση περιήγησης — το `ws://` γίνεται `http://` και αφαιρείται η διαδρομή `/audio`, `/ws` ή `/kiwi/.../SND` που χρειάζεται κάθε είδος δέκτη, αφού η σελίδα του σταθμού βρίσκεται στη ρίζα. Είναι πραγματικός σύνδεσμος, οπότε το μεσαίο κλικ ή το παρατεταμένο πάτημα λειτουργούν ως συνήθως.
- **Η σειρά είναι δική σας.** Μετακινήστε μια γραμμή με τα  , ή σύρετέ την και αφήστε την εκεί που της αρμόζει. Με αυτή τη σειρά προτείνει τις διευθύνσεις και η αναδυόμενη λίστα.
- **export και import.** Οι λίστες βρίσκονται στην αποθήκευση του ίδιου του browser, η οποία χάνεται όταν καθαρίζετε τα δεδομένα του ιστότοπου — ακριβώς από αυτό προστατεύει το export. Το export

γράφει και τις τέσσερις λίστες, μαζί με τα ονόματα, σε ένα αρχείο JSON που μπορείτε να κρατήσετε ή να μεταφέρετε σε άλλον browser ή σε άλλο μηχάνημα. Το import το διαβάζει πίσω και ρωτά αν θα γίνει **merge** με τα ήδη αποθηκευμένα ή **replace** των πάντων.

-  **QR**. Σχεδιάζει όλες τις λίστες ως κωδικό QR στην οθόνη, για να τις μεταφέρετε σε κινητό — δείτε «Στο κινητό» παρακάτω. Μια λίστα πολύ μεγάλη για σάρωση καταφεύγει στο κείμενο δίπλα της, που μπορείτε να αντιγράψετε και να επικολλήσετε οπουδήποτε.

Δεν υπάρχει όριο στο πόσους δέκτες κρατάτε.

## Στο κινητό

Η σελίδα /mobile έχει την ίδια λειτουργία σε μια καρτέλα **Div**, δίπλα στις Audio, Bands, Marks, Users και Chat: τύπος δέκτη, διεύθυνση, Start και Stop, οι αποθηκευμένοι δέκτες ως λίστα που επιλέγετε με ένα άγγιγμα, οι ζωντανές ενδείξεις και το SNR trim. Κάθε γραμμή έχει επίσης τον σύνδεσμο \*\*\*\* προς τη σελίδα εκείνου του δέκτη. Ο δεύτερος δέκτης ακολουθεί μόνος του τον συντονισμό του κινητού, ακριβώς όπως και στη σελίδα του υπολογιστή. Αυτό που λείπει είναι η επεξεργασία — μετονομασία, αλλαγή σειράς, διαγραφή — που παραμένει στη σελίδα του υπολογιστή, όπου υπάρχει χώρος γι' αυτήν.

Η αποθηκευμένη λίστα ανήκει σε έναν browser, οπότε ένα κινητό ξεκινά με άδεια λίστα όσους δέκτες κι αν έχετε αποθηκευμένους στον υπολογιστή. Γι' αυτό υπάρχει ο κωδικός QR: πατήστε  και μετά  **QR** στον υπολογιστή, σαρώστε τον κωδικό με την κάμερα του κινητού και επικολλήστε το κείμενο στο **Import** της καρτέλας Div. Ρωτά αν θα γίνει merge ή replace, ενώ το Export στο κινητό στέλνει μια λίστα προς την αντίθετη κατεύθυνση.

## Χρήση WebSDR

Ένας WebSDR — το λογισμικό του Pieter-Tjerk de Boer, PA3FWM, που λειτουργεί στο Twente και σε αρκετές εκατοντάδες άλλους σταθμούς — μπορεί να χρησιμοποιηθεί ως δεύτερος δέκτης, αλλά όχι απευθείας από τον browser.

Ο λόγος είναι ένας σκόπιμος έλεγχος από τη δική τους πλευρά. Ένας WebSDR αρνείται τη σύνδεση ήχου αν η κεφαλίδα `Origin` δεν αναφέρει τον δικό του ιστότοπο, και το `Origin` είναι *απαγορευμένο όνομα κεφαλίδας*: ο browser το θέτει από τη σελίδα στην οποία βρίσκεστε και κανένα script δεν επιτρέπεται να το αλλάξει. Δεν υπάρχει τρόπος παράκαμψης από την πλευρά του client — ούτε θα έπρεπε να υπάρχει.

Έτσι, τη σύνδεση την κάνει ένα μικρό πρόγραμμα στον δικό σας server, το `websdr_relay.py`. Ο browser σας μιλάει στο relay, και το relay μιλάει στον WebSDR.

## Εγκατάσταση του relay

Οι τέσσερις installers των διανομών το προσφέρουν ως προαιρετικό βήμα. Για μια υπάρχουσα εγκατάσταση:

```
cd ~/PhantomSDR-Plus
./setup_websdr_relay.sh
```

Ζητά μια θύρα, παίρνει το διακριτικό και τη διεύθυνση του σταθμού σας από το `frontend/site_information.json` και προσφέρεται να εγκαταστήσει υπηρεσία systemd ώστε το relay να ξεκινά με την εκκίνηση. Οι ρυθμίσεις βρίσκονται στο `websdr_relay.json` και αλλάζουν οποιαδήποτε στιγμή.

**Ένα πράγμα δεν μπορεί να κάνει ο installer για εσάς: να προωθήσει τη θύρα του relay στον router σας.** Οι browsers των ακροατών συνδέονται απευθείας στο relay — δεν περνά μέσα από τον δέκτη — οπότε χωρίς αυτή την προώθηση η διαφορική λήψη με WebSDR λειτουργεί μόνο μέσα από το δικό σας δίκτυο. Οι άλλοι τρεις τύποι δεκτών δεν επηρεάζονται.

Αν είναι εγκατεστημένος ο πίνακας διαχείρισης, το dashboard του δείχνει μια κάρτα **WebSDR Diversity Relay** με την κατάσταση, τη θύρα και πόσες συνεδρίες τρέχουν και προς ποιους σταθμούς.

## Να είστε καλός επισκέπτης

Το relay συνδέεται σε δέκτες άλλων από τον server σας και όχι από τη διεύθυνση του κάθε ακροατή, γι' αυτό είναι φτιαγμένο να συμπεριφέρεται σωστά:

- **Όριο δέκα ταυτόχρονων συνεδριών προς έναν WebSDR**, και εξήντα συνολικά. Αυτό εμποδίζει μια πολυάσχολη μέρα στον σταθμό σας να μοιάζει με επίθεση στον σταθμό κάποιου άλλου.
- **Ένα User-Agent που ονομάζει τον σταθμό σας και τον χειριστή του**, ώστε ένας χειριστής που δεν το επιθυμεί να ξέρει ακριβώς σε ποιον να γράψει. Συμπληρώστε το τίμιο.
- **Πρωθούνται μόνο εντολές συντονισμού**. Η σύνδεση δεν μπορεί να χρησιμοποιηθεί για τίποτε άλλο.
- **Ιδιωτικές, loopback και carrier-NAT διευθύνσεις απορρίπτονται**, ώστε κανείς επισκέπτης να μη στρέψει το relay σε κάτι μέσα στο μηχάνημα ή στο τοπικό σας δίκτυο.

### ΣΗΜΑΝΤΙΚΟ

Ο έλεγχος Origin υπάρχει επειδή οι δημιουργοί του WebSDR δεν ήθελαν οι δέκτες τους να οδηγούνται από σελίδες τρίτων. Η χρήση του relay είναι συνειδητή απόφαση, όχι παράλειψη δική τους. Κρατήστε το όριο εκεί που είναι, κρατήστε το User-Agent τίμιο, και σταματήστε αν σας το ζητήσει ένας χειριστής.

## Τι διαφέρει σε έναν WebSDR

- **Πολλοί σταθμοί καλύπτουν λίγες στενές λωρίδες φάσματος**, όχι συνεχές εύρος — π.χ. ένα παράθυρο 256 kHz στα 40 m. Η διαφορική λήψη ενεργοποιείται μόνο μέσα σε αυτές, και ο πίνακας διαβάζει τη λίστα μπαντών του σταθμού για να ξέρει πού βρίσκονται.
- **Ο ρυθμός δειγματοληψίας του ήχου αλλάζει** με το πλάτος του φίλτρου και απέχει περισσότερο από την ονομαστική τιμή απ' ό,τι στους άλλους τύπους, οπότε η ευθυγράμμιση μπορεί να αργήσει λίγο περισσότερο.
- **Το CW δεν προσαρμόζεται** στη σύμβαση του WebSDR, που τοποθετεί τη ζώνη διέλευσης εξ ολοκλήρου κάτω από τον φέροντα. Το SSB και το AM είναι σωστά.

## Ανάγνωση του πίνακα

Όταν τρέχει, ο πίνακας δείχνει έναν μικρό πίνακα τιμών:

Γραμμή	Σημασία
<b>link</b>	Η σύνδεση με τον δεύτερο δέκτη. Το ζητούμενο είναι <code>ready</code> . Σε αποτυχία εμφανίζεται το κείμενο σφάλματος του άλλου server και ο κωδικός τερματισμού
<b>aligned</b>	Η μετρημένη καθυστέρηση μεταξύ των δύο ροών μόλις κλειδώσουν, ή <code>searching</code>
<b>corr</b>	Πόσο ισχυρά συσχετίζονται οι δύο ροές. Έχει νόημα μόνο μετά την ευθυγράμμιση
<b>remote audio</b>	Αποκωδικοποιημένος ήχος που φτάνει από τον δεύτερο δέκτη. Το <code>none</code> σε πορτοκαλί σημαίνει ότι δεν αποκωδικοποιείται τίποτα
<b>SNR local / remote</b>	Οι δύο εκτιμήσεις λόγου σήματος προς θόρυβο που συγκρίνονται
<b>switches</b>	Πόσες φορές άλλαξε τοποθεσία

Δίπλα στον τίτλο, μία λέξη σας λέει πού βρίσκεστε:

off (γκρι)	δεν εκτελείται
Please wait... (πορτοκαλί)	συνδέεται, ή συνδέθηκε και ευθυγραμμίζεται ακόμη
Ready (πράσινο)	συνδεδεμένο, κλειδωμένο και ακολουθεί την καλύτερη τοποθεσία
το μήνυμα σφάλματος του δέκτη (κόκκινο)	απέτυχε και δεν θα διορθωθεί μόνο του

Η διάκριση πορτοκαλί/κόκκινου είναι αυτή που γλιτώνει χρόνο: πορτοκαλί σημαίνει συνεχίστε να περιμένετε, κόκκινο σημαίνει σταματήστε να περιμένετε και διαβάστε το μήνυμα.

Η χρωματιστή κουκκίδα προσθέτει ποια τοποθεσία ακούτε πραγματικά — πράσινο ο δικός σας δέκτης, κυανό ο απομακρυσμένος.

## Η ευθυγράμμιση θέλει περίπου 15 δευτερόλεπτα

Αυτό είναι φυσιολογικό. Οι δύο ροές ευθυγραμμίζονται με συσχέτιση των **περιβαλλουσών του ήχου**, και αυτό χρειάζεται ένα παράθυρο ήχου. Η αναζήτηση γίνεται σε δύο στάδια: μια στενή αναζήτηση  $\pm 1,5$  δευτερολέπτου μπορεί να ξεκινήσει μόλις μαζευτούν περίπου 9 δευτερόλεπτα ήχου, και μια δεύτερη, ανεξάρτητη μέτρηση πέντε δευτερόλεπτα αργότερα πρέπει να **συμφωνήσει** με την πρώτη για να γίνει αποδεκτή η ευθυγράμμιση. Όταν οι δύο σταθμοί απέχουν χρονικά περισσότερο από αυτό — ένα ή δύο δευτερόλεπτα επιπλέον buffering κάπου στη διαδρομή — αναλαμβάνει η πλήρης αναζήτηση  $\pm 4$  δευτερολέπτων στα 14 δευτερόλεπτα και η επιβεβαίωση έρχεται στα 19.

Το βήμα επιβεβαίωσης έχει σημασία. Όταν δύο τοποθεσίες κάνουν fading σε αντίθετη φάση — χωρίς ποτέ να έχουν και οι δύο το σήμα ταυτόχρονα — μια μόνο συσχέτιση θα κλειδώσει με βεβαιότητα σε εντελώς λάθος καθυστέρηση, που ακούγεται σαν ηχώ. Η απαίτηση δύο ανεξάρτητων μετρήσεων που συμφωνούν το απορρίπτει αυτό.

Ένα δεύτερο πράγμα πρέπει να είναι σωστό πριν κρατήσει το κλείδωμα: οι ρυθμοί δειγματοληψίας των δύο δεκτών. Δύο δέκτες σημαίνει δύο ρολόγια και δύο αλυσίδες αποδεκατισμού, οπότε οι ροές ήχου τους **δεν** φτάνουν με ακριβώς τον ίδιο ρυθμό ακόμη κι όταν και οι δύο δηλώνουν 12 kHz — διαφορά 2% είναι φυσιολογική, δηλαδή ολίσθηση 240 δειγμάτων ανά δευτερόλεπτο, πολύ περισσότερο από την ανοχή ευθυγράμμισης. Γι' αυτό η απομακρυσμένη ροή επαναδειγματοληπτείται συνεχώς ώστε να ταιριάζει με τον ρυθμό του δικού σας δέκτη. Ο λόγος μετρίεται **μετρώντας δείγματα**, όχι με συσχέτιση, οπότε δεν χρειάζεται δικό του κλείδωμα και σταθεροποιείται μέσα στα πρώτα δευτερόλεπτα — πριν καν τρέξει η πρώτη αναζήτηση ευθυγράμμισης. Ένα αδύναμο ή διακοπτόμενο σήμα μπορεί να χρειαστεί δύο ή τρεις προσπάθειες, οπότε τα 30 δευτερόλεπτα δεν είναι λόγος ανησυχίας· ένα λεπτό χωρίς αποτέλεσμα είναι.

Όσο εμφανίζεται **searching**, και οι δύο τιμές SNR δείχνουν 0.0. Αυτό είναι εσκεμμένο: οι δύο εκτιμήσεις μετρώνται σε **ευθυγραμμισμένα ως προς το περιεχόμενο** δείγματα, άρα τίποτα δεν μετρίεται πριν υπάρξει ευθυγράμμιση. Το 0.0 / 0.0 σημαίνει «δεν κλείδωσε ακόμη», όχι «δεν υπάρχει σήμα».

## Remote SNR trim

Αυτό το ποτενσιόμετρο μεροληπτεί την επιλογή μεταξύ των δύο τοποθεσιών. Προστίθεται στο μετρημένο SNR του απομακρυσμένου δέκτη πριν από τη σύγκριση — θετικές τιμές ευνοούν τον απομακρυσμένο, αρνητικές τον δικό σας. Δεν αλλάζει **τίποτα άλλο**: ούτε τη στάθμη του ήχου ούτε τον συνδυασμό, μόνο ποια τοποθεσία κερδίζει.

Υπάρχει επειδή οι δύο τιμές SNR δεν είναι πάντα συγκρίσιμες. Και οι δύο μετρώνται με τον ίδιο τρόπο — ως εκατοστημόριο εύρος του ήχου — αλλά ένας δέκτης του οποίου ο codec ανεβάξει το δικό του πάτωμα θορύβου δείχνει χαμηλότερα απ' όσο του αξίζει. Ένας KiwiSDR εφαρμόζει ισχυρή ενίσχυση και

limiter, που συμπιέζει το εύρος· ένας απωλεστικός codec θέτει πάτωμα θορύβου κάτω από το οποίο το σήμα δεν κατεβαίνει ποτέ.

### Πώς να το ρυθμίσετε:

1. Συντονίστε και τους δύο δέκτες σε ένα σήμα που υπάρχει σταθερά και περιμένετε το `aligned`.
2. Παρακολουθήστε τις δύο τιμές SNR για μισό λεπτό. Σημειώστε τη συνήθη διαφορά.
3. Χρησιμοποιήστε τα άκρα ως δοκιμή ακρόασης: **+15** επιβάλλει τον απομακρυσμένο, **-15** τον δικό σας. Ακούστε τον καθένα για λίγα δευτερόλεπτα. Είναι ο μόνος τρόπος να ακούσετε μία τοποθεσία μεμονωμένα.
4. Αν ο απομακρυσμένος δείχνει π.χ. 5 dB χαμηλότερα αλλά ακούγεται εξίσου καλά, βάλτε **+5**. Κάνετε τις δύο ενδείξεις να συμφωνούν όταν οι δύο τοποθεσίες ακούγονται ίδιες.
5. Παρακολουθήστε τον μετρητή **switches** για δέκα λεπτά ακρόασης. Συνεχείς εναλλαγές σημαίνουν ότι το trim είναι πολύ κοντά στην ισοπαλία· καμία εναλλαγή ενώ ο δικός σας δέκτης ακούγεται να σβήνει σημαίνει ότι είναι πολύ μακριά προς την άλλη κατεύθυνση.

Σημεία εκκίνησης: **0** για έναν άλλο PhantomSDR-Plus, που έχει πανομοιότυπη διαδρομή ήχου και είναι εξ ορισμού συγκρίσιμος· μια μικρή **θετική** τιμή για KiwiSDR ή UberSDR.

#### ΣΥΜΒΟΥΛΗ

Αν χρειάζεστε πάνω από  $\pm 8$  dB περίπου, σταματήστε. Σε εκείνο το σημείο η τίμια εξήγηση είναι συνήθως ότι ο ένας δέκτης πράγματι είναι χειρότερος για τη συγκεκριμένη διαδρομή, και το να μεροληπτείτε πέρα από μια πραγματική διαφορά σημαίνει απλώς ότι ακούτε τον χειρότερο.

Η ρύθμιση εφαρμόζεται αμέσως και απομνημονεύεται στον browser σας.

## Επιλογή δεύτερου δέκτη

**Η απόσταση μετράει.** Τα *fadings* αποσυσχετίζονται με την απόσταση — μερικές εκατοντάδες χιλιόμετρα στα HF είναι καλός στόχος. Δύο δέκτες στην ίδια πόλη σβήνουν μαζί και δεν προσφέρουν τίποτα. Πολύ μακριά, και ο δεύτερος δέκτης μπορεί να μην ακούει καθόλου το σήμα σας.

**Και οι δύο πρέπει όντως να ακούν το σήμα.** Αυτή είναι η απαίτηση που κανένα λογισμικό δεν παρακάμπτει. Αν ο δεύτερος δέκτης δεν ακούει αυτό που ακούτε εσείς, η συσχέτιση μένει χαμηλή και δεν θα κλειδώσει — σκόπιμα, γιατί μια λάθος ευθυγράμμιση ακούγεται χειρότερα από καθόλου *diversity*.

**Κάλυψη.** Ένας δεύτερος PhantomSDR-Plus ή ένας KiwiSDR ανακοινώνει το εύρος συχνοτήτων που καλύπτει, και ο πίνακας σας προειδοποιεί αν συντονιστείτε εκτός του. Ένας WebSDR ανακοινώνει τα μπαντ του, που συχνά είναι στενές λωρίδες αντί για συνεχές εύρος. Ο UberSDR δεν ανακοινώνει κάλυψη, οπότε εκεί η άρνηση κλειδώματος είναι η μόνη ένδειξη.

#### ΣΗΜΑΝΤΙΚΟ

Οι δημόσιοι δέκτες λειτουργούν από εθελοντές και έχουν περιορισμένο αριθμό θέσεων ακροατών. Μια συνεδρία *diversity* καταλαμβάνει μία από αυτές για όσο τρέχει, ακριβώς όπως ένας ακροατής. Δείξτε κατανόηση πριν αφήσετε μία συνδεδεμένη επ' αόριστον, και ρωτήστε τον υπεύθυνο αν σκοπεύετε να χρησιμοποιείτε τον δέκτη του εντατικά.

## Επίλυση προβλημάτων

**Το link δεν φεύγει από το connecting ή κάνει κύκλους σύνδεσης/κλεισίματος** Πιθανότατα λάθος διεύθυνση ή θύρα. Δείτε τον πίνακα παραπάνω — ιδίως ότι ο UberSDR χρησιμοποιεί την κανονική θύρα

web, όχι την 8073. Ο πίνακας δείχνει τον κωδικό τερματισμού και η κονσόλα του browser (F12) το υποκείμενο σφάλμα.

**Το link δείχνει σφάλμα με κείμενο** Αυτό το κείμενο προέρχεται από τον άλλο δέκτη και συνήθως είναι συγκεκριμένο: απορριφθείσα συνεδρία, γεμάτος server ή δέκτης που δεν δέχεται εξωτερικούς πελάτες.

**Ο KiwiSDR αρνείται τη σύνδεση** Κάποιοι KiwiSDR απαιτούν κωδικό, και σε έναν UberSDR το συμβατό με KiwiSDR άκρο είναι απενεργοποιημένο εξ ορισμού. Για UberSDR χρησιμοποιήστε τον τύπο **UberSDR** — αυτή η διαδρομή είναι πάντα διαθέσιμη.

**WebSDR: the WebSDR relay is not reachable** Το relay δεν τρέχει, ή ο browser σας δεν φτάνει στη θύρα του. Πρέπει να είναι προσβάσιμο στο **ίδιο hostname από το οποίο σερβίρεται η σελίδα του δέκτη**, επειδή ο browser συνδέεται απευθείας σε αυτό και όχι μέσω του δέκτη. Ανοίξτε το `http://<αυτό το hostname>:<θύρα relay>/status` στον ίδιο browser για να δείτε ποιο από τα δύο ισχύει. Αν απαντά στο τοπικό δίκτυο αλλά όχι από έξω, η θύρα δεν είναι προωθημένη στον router.

**WebSDR: σφάλμα που αναφέρει απόρριψη σύνδεσης** Ο σταθμός αυτός έχει αυστηρότερες ρυθμίσεις από τον συνήθη έλεγχο `Origin`, ή ο χειριστής του έχει αποκλείσει αυτόν τον σταθμό.

**Το link είναι ready αλλά το remote audio δείχνει none** Ο ήχος δεν αποκωδικοποιείται. Ανανεώστε τη σελίδα· αν επιμένει, η κονσόλα του browser θα δείξει ποιος αποκωδικοποιητής απέτυχε.

**Φτάνει ήχος αλλά δεν ευθυγραμμίζεται ποτέ** Οι δύο δέκτες δεν ακούν το ίδιο πράγμα. Δοκιμάστε έναν ισχυρό ραδιοφωνικό σταθμό ή μια μπάντα όπου και οι δύο τοποθεσίες έχουν καλή διάδοση. Μια ήσυχη μπάντα με μόνο θόρυβο και στις δύο πλευρές δεν θα συσχετιστεί ποτέ — και σωστά.

**Εναλλάσσεται συνεχώς** Οι δύο τοποθεσίες είναι πολύ κοντά σε SNR. Ρυθμίστε ελαφρώς το trim ώστε η μία να προτιμάται σταθερά.

**Δεν χρησιμοποιεί ποτέ τον δεύτερο δέκτη** Ελέγξτε ότι το `remote audio` ανεβαίνει και ότι οι τιμές SNR φαίνονται λογικές. Αν ο απομακρυσμένος δείχνει πολύ χαμηλότερα απ' ό,τι ακούγεται, γι' αυτό υπάρχει το trim.

## Περιορισμοί

- **Δεν είναι σύμφωνο.** Κανένα phasing, κανένα nulling, καμία ραδιογωνιομετρία. Η ακύρωση μιας πηγής θορύβου απαιτεί δύο κεραίες σε κοινό ρολόι στην ίδια τοποθεσία· αυτή είναι άλλη τεχνική για άλλο πρόβλημα.
- **Ένα έως τρία δευτερόλεπτα καθυστέρηση,** αναπόφευκτα.
- **Μόνο δύο δέκτες.**
- **Μονοφωνικό.** Αν ο δέκτης σας είναι σε στερεοφωνικό mode όπως το C-QUAM, ο ήχος περνά ανέπαφος και το diversity δεν ενεργοποιείται.
- **Η ευθυγράμμιση θέλει σήμα.** Κάτω από περίπου 10 dB SNR στη μία από τις δύο τοποθεσίες, περιμένετε να παραμείνει στο `searching`.
- **Ένας WebSDR χρειάζεται το relay** στον δικό σας server, με τη θύρα του προωθημένη. Οι άλλοι τρεις τύποι δεκτών δεν χρειάζονται τίποτα από τα δύο.

# Έλεγχος πομποδέκτη

RIG\_CONTROL.md

## PhantomSDR-Plus — Έλεγχος πομποδέκτη (CAT)

Κρατήστε **τον δικό σας πομποδέκτη** και έναν **δέκτη PhantomSDR-Plus** στην ίδια συχνότητα, τρόπο λειτουργίας και φίλτρο. Γυρίστε το κουμπί συντονισμού του πομποδέκτη και ο καταρράκτης ακολουθεί· κάντε κλικ σε ένα σήμα στον καταρράκτη και ο πομποδέκτης συντονίζεται εκεί. Όταν εκπέμπετε, ο δέκτης μπορεί να σιγήσει, ώστε να μη σας ξαναπαίξει το δικό σας σήμα.

Λειτουργεί με δέκτες **PhantomSDR-Plus, KiwiSDR, PA3FWM WebSDR και UberSDR**, δικούς σας ή άλλων, και μετακινεί πάντα μόνο τη *δική σας* συνεδρία ακρόασης — κανείς άλλος στον δέκτη δεν ακούει ούτε βλέπει τίποτα. Ο διαχειριστής του δέκτη δεν χρειάζεται να εγκαταστήσει ή να ρυθμίσει τίποτα.

### Τι χρειάζεστε

Δύο τρόποι σύνδεσης πομποδέκτη και μία προϋπόθεση στον δέκτη:

Στοιχείο	Τι είναι	Συγχρονίζει
<b>Desktop PhantomSDR+ 4.0 ή νεότερο</b>	Η εφαρμογή υπολογιστή, με μενού <b>Rig</b> . Linux (PC και Raspberry Pi) και Windows.	Συχνότητα, τρόπο λειτουργίας, εύρος φίλτρου, σίγαση κατά την εκπομπή — προς τη μία κατεύθυνση ή και τις δύο
<b>CATsync Tool for WebSDRs</b>	Ξεχωριστό πρόγραμμα Windows που συνδέει πομποδέκτη με τη σελίδα του δέκτη στον browser σας.	Συχνότητα και τρόπο λειτουργίας
<b>TCI-CAT στη σελίδα του δέκτη</b>	Ένα κουμπί πάνω στην ίδια τη σελίδα PhantomSDR-Plus, σε οποιονδήποτε browser. Μιλά TCI με το ExpertSDR, το AetherSDR ή το Thetis, ή με οποιονδήποτε πομποδέκτη υποστηρίζει το Hamlib μέσω μιας μικρής γέφυρας. Μόνο σε δέκτες PhantomSDR-Plus με την έκδοση 4.1.0 ή νεότερη.	Συχνότητα, τρόπος λειτουργίας και εύρος φίλτρου και προς τις δύο κατευθύνσεις· σίγαση κατά την εκπομπή
<b>Ο δέκτης</b>	PhantomSDR-Plus, KiwiSDR, PA3FWM WebSDR ή UberSDR. Ένας δέκτης PhantomSDR-Plus χρειάζεται την <b>4.1.0</b> ή νεότερη για εύρος φίλτρου και σίγαση.	Παλαιότερος PhantomSDR-Plus συγχρονίζει κανονικά συχνότητα και τρόπο λειτουργίας

Το μεγαλύτερο μέρος του εγχειριδίου περιγράφει το Desktop PhantomSDR+. Το κουμπί TCI-CAT έχει δική του ενότητα, [TCI-CAT στη σελίδα του δέκτη](#). Το CATsync Tool έχει τη δική του τεκμηρίωση στον ιστότοπό του.

## Υποστηριζόμενοι δέκτες

Η εφαρμογή αναγνωρίζει το είδος του δέκτη σε ένα παράθυρο σταθμού και τον χειρίζεται μέσω των δικών του χειριστηρίων της σελίδας. Το παράθυρο Rig control δείχνει το είδος που βρήκε δίπλα στο όνομα του σταθμού.

Δέκτης	Συχνότητα και τρόπος	Εύρος φίλτρου	Σίγαση κατά την εκπομπή
PhantomSDR-Plus	Ναι	Με την 4.1.0 ή νεότερη	Με την 4.1.0 ή νεότερη
KiwiSDR (και Web-888)	Ναι	Ναι	Ναι
PA3FWM WebSDR	Ναι, με αλλαγή μπάντας σε σταθμούς πολλών μπαντών	Ναι	Ναι
UberSDR	Ναι	Ναι	Ναι

Ένας web δέκτης έχει λιγότερους τρόπους λειτουργίας από τους περισσότερους πομποδέκτες, οπότε κάποιος τρόπος του πομποδέκτη μοιράζονται έναν τρόπο του δέκτη: το KiwiSDR και το WebSDR έχουν ένα μόνο CW για CW και CW-R. Οι τρόποι δέκτη που αναφέρει αυτό το εγχειρίδιο είναι του PhantomSDR-Plus· οι άλλοι δέκτες χρησιμοποιούν τον πλησιέστερο αντίστοιχο. Ένα WebSDR κρατά φίλτρο CW κάτω από 1 kHz και τα άλλα φίλτρα 1 kHz ή ευρύτερα, γιατί έτσι ξεχωρίζει η ίδια η σελίδα το CW, και μια συχνότητα έξω από κάθε μπάντα του σταθμού WebSDR αγνοείται. Το UberSDR ρυθμίζει το φίλτρο του σε βήματα ολισθητή, οπότε ένα εύρος μπορεί να απέχει έως περίπου 50 Hz από του πομποδέκτη. Κάθε άλλο είδος σελίδας — π.χ. OpenWebRX — δείχνει *not a receiver this app can drive* και δεν συγχρονίζεται τίποτα.

## Τι κάνει και τι δεν κάνει

- Συγχρονίζει **έναν πομποδέκτη με ένα παράθυρο δέκτη** κάθε φορά.
- Διαβάζει και τις δύο πλευρές αρκετές φορές το δευτερόλεπτο και, όταν διαφωνούν, ρυθμίζει τη μία ώστε να ταιριάζει με την άλλη. **Δεν** εκπέμπει, δεν ενεργοποιεί το PTT και δεν στέλνει ήχο πουθενά.
- Η μετακίνηση του δέκτη αλλάζει μόνο τη δική σας συνεδρία. Οι άλλοι ακροατές του ίδιου δέκτη δεν επηρεάζονται και ο διαχειριστής δεν χρειάζεται να το επιτρέψει.
- Μια σειριακή θύρα μπορεί να την ανοίξει **ένα πρόγραμμα τη φορά**. Αν το WSJT-X, ένα πρόγραμμα ημερολογίου ή ένα εργαλείο του κατασκευαστή έχει ήδη τη θύρα, χρησιμοποιήστε την επιλογή **flrig** ή **rigctl on network** για να μοιραστείτε τον πομποδέκτη αντί να μαλώνουν για τη θύρα.

## Γρήγορη εκκίνηση

1. Ανοίξτε έναν σταθμό στο Desktop PhantomSDR+ ως συνήθως.
2. **Rig → Rig control...**
3. Στο **Connection** επιλέξτε **Built-in** αν ο πομποδέκτης σας είναι στη λίστα, αλλιώς **Hamlib (all rigs)**.
4. Επιλέξτε τον πομποδέκτη, τη σειριακή θύρα και την ταχύτητα στην οποία είναι ρυθμισμένο το μενού CAT ή CI-V του πομποδέκτη.
5. Στο **Sync** αφήστε επιλεγμένο το **Both directions**.

6. Πατήστε **Connect**. Οι δύο ενδείξεις στην κορυφή — πομποδέκτης και δέκτης — πρέπει να δείξουν την ίδια συχνότητα μέσα σε ένα δευτερόλεπτο.

Κάθε ρύθμιση αποθηκεύεται μόλις την αλλάξετε. Την επόμενη φορά αρκεί **Rig → Connect**, ή τσεκάρετε **Connect when the app starts**.

## Επιλογή τρόπου σύνδεσης

Επιλογή	Χρησιμοποιήστε την όταν	Χρειάζεται
<b>Built-in</b>	Ο πομποδέκτης σας είναι στη λίστα παρακάτω.	Τίποτα άλλο
<b>Hamlib (all rigs)</b>	Ο πομποδέκτης σας είναι οτιδήποτε άλλο — το Hamlib γνωρίζει πάνω από 300. Η εφαρμογή ξεκινά για εσάς το <code>rigctld</code> του Hamlib σε ιδιωτική τοπική θύρα και το σταματά όταν αποσυνδεθείτε.	Windows: τίποτα, το Hamlib περιλαμβάνεται. Linux: <code>sudo apt install libhamlib-utils</code>
<b>rigctld on network</b>	Ήδη τρέχει ένα <code>rigctld</code> , σε αυτόν ή σε άλλον υπολογιστή του δικτύου σας.	Host και θύρα (προεπιλογή 4532)
<b>flrig</b>	Το flrig ήδη ελέγχει τον πομποδέκτη για fldigi, WSJT-X ή πρόγραμμα ημερολογίου.	Να τρέχει το flrig, με τη θύρα XML-RPC του (προεπιλογή 12345)

## Πομποδέκτες με ενσωματωμένο οδηγό

Η ταχύτητα και η διεύθυνση CI-V είναι οι εργοστασιακές τιμές που συμπληρώνει η εφαρμογή. **Είναι μόνο αφετηρία — βάλτε ό,τι λέει το μενού του πομποδέκτη σας.**

Οικογένεια	Πομποδέκτες	Προεπιλεγμένη ταχύτητα	Σημειώσεις
<b>Icom CI-V</b>	IC-7300, IC-7610, IC-705, IC-9700, IC-905, IC-7760, IC-7851, IC-7100, IC-7410, IC-9100, IC-7600, IC-7200, IC-7700, IC-7000, IC-7800, IC-756PROIII, IC-756PROII, IC-R8600 και κάθε άλλος με CI-V	19200	Διεύθυνση CI-V ανά μοντέλο (IC-7300 94, IC-705 A4, IC-9700 A2, IC-7610 98 ...)
	IC-746PRO, IC-718, IC-R75	9600	
<b>Xiegu (CI-V)</b>	G90, X6100	19200	Διεύθυνση 70· ελέγξτε το μενού
<b>Yaesu νέο CAT</b>	FTDX101D/MP, FTDX10, FT-710, FT-991/A, FT-891, FTDX5000, FTDX3000, FTDX1200, FT-950, FT-2000, FT-450/450D	38400	
<b>Yaesu κλασικό CAT</b>	FT-817/818, FT-857/857D, FT-897/897D	38400	2 stop bits· συντονίζει σε βήματα 10 Hz
<b>Kenwood</b>	TS-990S, TS-890S, TS-590S/SG	115200	
	TS-480, TS-2000, TS-870S	57600	
<b>Elecraft</b>	K4, K3/K3S, KX3, KX2	38400	Συγχρονίζεται και το φίλτρο
<b>Συμβατοί με Kenwood</b>	FlexRadio SmartSDR CAT (εικονική θύρα), QRP Labs QMX/QMX+/QDX, (tr)uSDX, Lab599 Discovery TX-500, άλλοι συμβατοί με Kenwood	9600–38400	

Ένας πομποδέκτης που θα έπρεπε να είναι συμβατός αλλά δεν συνεννοείται με ενσωματωμένο οδηγό, συνήθως δουλεύει με **Hamlib**, που αντέχει πολύ περισσότερες παραλλαγές.

## Ρυθμίσεις σειριακής θύρας

Ρύθμιση	Τι βάζετε
<b>Serial port</b>	Τη θύρα του πομποδέκτη. Οι μετατροπείς USB και οι πομποδέκτες με θύρα USB εμφανίζονται πρώτοι. Το <b>Other / network address...</b> δέχεται θύρα που δεν φαίνεται στη λίστα — COM7, /dev/ttyUSB1 — ή tcp://host:port για σειριακή θύρα που δίνεται μέσω δικτύου από ser2net ή παρόμοιο.
<b>Speed (baud)</b>	Ακριβώς ό,τι λέει το μενού ταχύτητας CAT / CI-V του πομποδέκτη. Λάθος ταχύτητα μοιάζει με πομποδέκτη που δεν απαντά ποτέ.
<b>Stop bits</b>	1 για σχεδόν όλους· 2 για την οικογένεια FT-817/857/897.
<b>CI-V address</b>	Μόνο Icom, σε δεκαεξαδικό (94, όχι 148). Πρέπει να ταιριάζει με το μενού διεύθυνσης CI-V.
<b>DTR / RTS</b>	Αφήστε τα <b>κλειστά</b> εκτός αν τα χρειάζεται η διεπαφή σας. Πολλά καλώδια CAT ενεργοποιούν τον πομπό ή κάνουν επανεκκίνηση του πομποδέκτη σε μία από αυτές τις γραμμές.
<b>Hardware flow control</b>	Κλειστό, εκτός αν το εγχειρίδιο ζητά RTS/CTS.

Στο **Hamlib** οι ίδιες ρυθμίσεις περνούν στο `rigctl`. Εκεί τα stop bits έχουν και επιλογή *Rig default*, και το **Extra rigctl options** δέχεται οτιδήποτε άλλο καταλαβαίνει το `rigctl`, π.χ. `--set-conf=post_write_delay=10`. Το **rigctl program** δείχνει σε συγκεκριμένο `rigctl` αν έχετε εγκαταστήσει περισσότερα από ένα.

## Συγχρονισμός

### Κατεύθυνση

Επιλογή	Τι συμβαίνει
<b>Rig → receiver</b>	Το παράθυρο του δέκτη ακολουθεί τον πομποδέκτη. Μια αλλαγή στον καταρράκτη επιστρέφει στη συχνότητα του πομποδέκτη.
<b>Receiver → rig</b>	Ο πομποδέκτης ακολουθεί το παράθυρο του δέκτη. Το γύρισμα του κουμπιού του πομποδέκτη αναιρείται.
<b>Both directions</b>	Κερδίζει η πλευρά που αγγίξατε <b>τελευταία</b> . Τη στιγμή της σύνδεσης, πριν αγγιχτεί καμία, κερδίζει ο πομποδέκτης.

Η κατεύθυνση αλλάζει και από το μενού **Rig** ενώ είστε συνδεδεμένοι.

### Πώς αποφεύγεται να «μαλώνουν» οι δύο πλευρές

Κάθε τιμή που γράφει η εφαρμογή εμφανίζεται λίγο μετά ως αλλαγή στην άλλη πλευρά. Αν αυτό λαμβανόταν κατά γράμμα, πομποδέκτης και δέκτης θα κυνηγούσαν ο ένας τον άλλον επ' άπειρον. Η εφαρμογή το αποφεύγει με τρεις τρόπους:

- Η σελίδα του δέκτη εφαρμόζει μια αλλαγή αμέσως, οπότε διαβάζεται ξανά αμέσως μετά την εγγραφή και αυτή η ένδειξη γίνεται η νέα αφετηρία.

- Ο πομποδέκτης εφαρμόζει μια αλλαγή λίγο αργότερα, οπότε κάθε τιμή που του στέλνεται απομνημονεύεται. Όταν ο πομποδέκτης αναφέρει αυτή την τιμή, αναγνωρίζεται ως εγγραφή της εφαρμογής και όχι ως χέρι στο κουμπί.
- Μια τιμή που ο πομποδέκτης αρνείται — π.χ. ευρυζωνικό FM σε πομποδέκτη HF — στέλνεται **δύο φορές** και μετά αφήνεται ήσυχη ώσπου να αλλάξει η πλευρά-πηγή, αντί να επαναλαμβάνεται πολλές φορές το δευτερόλεπτο.

Ο επανασυντονισμός της σελίδας του δέκτη μπορεί να την κάνει να διαλέξει τον προεπιλεγμένο τρόπο της μπάντας (π.χ. LSB κάτω από τα 10 MHz). Όταν οδηγεί ο πομποδέκτης, η εφαρμογή επαναφέρει αμέσως τον τρόπο του πομποδέκτη, ώστε πομποδέκτης σε USB στα 40 m να κρατά και τον δέκτη σε USB.

## Ποιο παράθυρο δέκτη

Το **Receiver window** ορίζει ποιος σταθμός ακολουθεί τον πομποδέκτη:

- **The station window last in front** (προεπιλογή) — με δύο σταθμούς ανοιχτούς, κάντε κλικ σε έναν και ο πομποδέκτης ακολουθεί αυτόν.
- **Συγκεκριμένος σταθμός** — ο πομποδέκτης μένει σε αυτόν, είτε είναι μπροστά είτε όχι. Αν ο σταθμός δεν είναι ανοιχτός, δεν συγχρονίζεται τίποτα ώσπου να ανοίξει.

## Ρυθμός ενημέρωσης

Το **Update every** ορίζει πόσο συχνά διαβάζονται οι δύο πλευρές: 150 ms, 300 ms (προεπιλογή), 500 ms ή 1 s. Γρηγορότερα νιώθεται πιο άμεσο στο κουμπί· αργότερα είναι πιο ήπιο για παλιό πομποδέκτη στα 4800 ή 9600 baud, όπου κάθε ανάγνωση κοστίζει πραγματικό χρόνο στη γραμμή.

## Τρόποι λειτουργίας

Τρόπος στον πομποδέκτη	Ο δέκτης ακούει σε
USB, LSB	USB, LSB
CW	CW
CW-R (αντίστροφο)	CW-L
AM, σύγχρονο AM, DSB	AM
FM, στενό FM	FM
Ευρυζωνικό FM	WBFM
RTTY / FSK	LSB
RTTY-R / FSK-R	USB
Ψηφιακοί τρόποι δεδομένων (USB-D, DATA-U, PKTUSB, DIG)	USB
Δεδομένα LSB, δεδομένα FM	LSB, FM

Τρόπος στον δέκτη	Ο πομποδέκτης ρυθμίζεται σε
USB, LSB	USB, LSB
CW	CW
CW-L	CW-R
AM, QUAM	AM
FM	FM
WBFM	WFM — οι περισσότεροι πομποδέκτες HF το αρνούνται και αφήνονται ήσυχοι μετά από δύο προσπάθειες
RADE (άνω / κάτω)	USB / LSB

Πομποδέκτης σε **τρόπο δεδομένων** παραμένει σε αυτόν: το USB του δέκτη θεωρείται ότι συμφωνεί με το USB-D του πομποδέκτη, ώστε ο δέκτης να μη βγάζει ποτέ τον πομποδέκτη από τον τρόπο δεδομένων.

## Εύρος φίλτρου

Τσεκάρετε **Sync filter width** για να ταιριάζουν οι ζώνες διέλευσης. Διαφορές κάτω από 60 Hz θεωρούνται ίσες, αφού κανένα φίλτρο δεν έχει τα ίδια βήματα με άλλο.

Σύνδεση πομποδέκτη	Εύρος φίλτρου
Hamlib	Ναι, όπου το Hamlib το υποστηρίζει για τον συγκεκριμένο πομποδέκτη
flrig	Ναι
Ενσωματωμένο Icom CI-V	Ναι — βήματα 50 Hz έως 500 Hz, μετά βήματα 100 Hz έως 3,6 kHz· AM σε βήματα 200 Hz έως 10 kHz· όχι σε FM
Ενσωματωμένο Elecraft	Ναι, σε βήματα 10 Hz
Ενσωματωμένο Kenwood, Yaesu, οικογένεια FT-817	Όχι — αυτοί οι πομποδέκτες επιλέγουν φίλτρα από πίνακες ανά μοντέλο. Για το φίλτρο χρησιμοποιήστε Hamlib

Οι δέκτες KiwiSDR, WebSDR και UberSDR έχουν πάντα έλεγχο φίλτρου. Ένας δέκτης PhantomSDR-Plus πρέπει να έχει την **4.1.0** ή νεότερη· σε παλαιότερο η συχνότητα και ο τρόπος λειτουργίας συγχρονίζονται κανονικά, και το παράθυρο Rig control εξηγεί γιατί δεν ακολουθεί το φίλτρο.

## Σίγαση κατά την εκπομπή

Τσεκάρετε **Mute receiver while transmitting**. Όσο ο πομποδέκτης εκπέμπει, το παράθυρο του δέκτη είναι σε σίγαση, και όταν σταματήσει ο ήχος επιστρέφει. Το κουμπί σίγασης του δέκτη το δείχνει και μπορείτε πάντα να καταργήσετε τη σίγαση με το χέρι.

Αν είχατε ήδη βάλει τον δέκτη σε σίγαση μόνοι σας, μένει σε σίγαση και μετά.

Χρειάζεται σύνδεση που αναφέρει κατάσταση εκπομπής — όλοι οι ενσωματωμένοι οδηγοί, το flrig και το Hamlib για τους περισσότερους πομποδέκτες — και, σε δέκτη PhantomSDR-Plus, την 4.1.0 ή νεότερη.

## Μετατόπιση συχνότητας

Το **Frequency offset** προστίθεται στη συχνότητα του πομποδέκτη για να δώσει του δέκτη:

συχνότητα δέκτη = συχνότητα πομποδέκτη + μετατόπιση

Διάταξη	Μετατόπιση
Transverter 2 m σε πομποδέκτη 10 m (τα 144,100 MHz φαίνονται ως 28,100 MHz)	116000000
Transverter 70 cm σε πομποδέκτη 2 m (432 → 144)	288000000
Χωρίς transverter	0

## Το μενού Rig

Στοιχείο	Λειτουργία
<b>Rig control...</b>	Ανοίγει το παράθυρο Rig control
<b>Connect / Disconnect</b> όνομα πομποδέκτη	Ξεκινά ή σταματά τον συγχρονισμό· στο Hamlib ξεκινά ή σταματά και το rigctl
<b>Rig to receiver / Receiver to rig / Both directions</b>	Κατεύθυνση συγχρονισμού
<b>Sync filter width</b>	Ναι / όχι
<b>Mute receiver while transmitting</b>	Ναι / όχι
Γραμμή κατάστασης	<i>Not connected, Connecting..., Connected: όνομα</i> , ή το τελευταίο σφάλμα

Η ζωντανή ένδειξη συχνότητας βρίσκεται στο παράθυρο Rig control και όχι στο μενού, που αλλιώς θα έκλεινε μόνο του σε κάθε αλλαγή.

## Linux

**Δικαίωμα στη σειριακή θύρα.** Οι σειριακές θύρες ανήκουν στην ομάδα `dialout`. Χρήστης εκτός αυτής παίρνει *Could not open ttyUSB0*. Προσθέστε τον εαυτό σας μία φορά και μετά αποσυνδεθείτε και ξανασυνδεθείτε:

```
sudo usermod -aG dialout $USER
```

**Hamlib.** Εγκαταστήστε το από τη διανομή σας:

```
sudo apt install libhamlib-utils
```

Το πακέτο `.deb` του Desktop PhantomSDR+ το προτείνει, οπότε το `sudo apt install ./phantomsdr-plus-desktop_4.0.0_amd64.deb` το φέρνει μαζί· το `dpkg -i` δεν εγκαθιστά προτεινόμενα πακέτα. Οι ενσωματωμένοι οδηγοί και το flrig δεν χρειάζονται Hamlib.

## Windows

Το `rigctl.exe` του ίδιου του Hamlib περιλαμβάνεται και στο 64-bit και στο 32-bit πρόγραμμα εγκατάστασης. Οι θύρες COM εμφανίζονται στη λίστα με το όνομά τους (COM3). Αν ο πομποδέκτης θέλει οδηγό USB, εγκαταστήστε πρώτα τον οδηγό του κατασκευαστή — ως τότε η θύρα δεν υπάρχει.

## TCI-CAT στη σελίδα του δέκτη

Μια σελίδα δέκτη PhantomSDR-Plus μπορεί να ακολουθεί έναν πομποδέκτη και **μόνη της**, χωρίς την εφαρμογή desktop: ο browser μιλά απευθείας με έναν διακομιστή **TCI** στον δικό σας υπολογιστή. Το TCI είναι το πρωτόκολλο ελέγχου μέσω WebSocket των ExpertSDR2/ExpertSDR3 (SunSDR), AetherSDR (FlexRadio) και Thetis (Apache Labs ANAN, Hermes). Για πομποδέκτη χωρίς TCI, μια μικρή γέφυρα που περιλαμβάνεται στο PhantomSDR-Plus κάνει οποιονδήποτε πομποδέκτη του Hamlib να μοιάζει με διακομιστή TCI.

Χρειάζεται δέκτη PhantomSDR-Plus με την έκδοση **4.1.0 ή νεότερη**. Οι σελίδες KiwiSDR, WebSDR και UberSDR δεν το έχουν.

### Το κουμπί TCI-CAT

Το κουμπί βρίσκεται στη σειρά με τα **VFO**, **Modes**, **Bands** και **IF Filters** — **TCI-CAT** στη φαρδιά διάταξη, **CAT** στη συμπαγή. Η κουκκίδα του είναι **πράσινη** όσο υπάρχει σύνδεση με διακομιστή TCI και **γκρι** αλλιώς. Ανοίγει ένα παράθυρο με: Μέχρι την 4.2.0 το κουμπί λεγόταν **QRG Sync**.

Στοιχείο	Τι κάνει
Κατάσταση	<i>Active</i> — TCI (port 50001) όταν υπάρχει σύνδεση· <i>Inactive</i> με <i>Wait</i> όσο γίνεται αναζήτηση
<b>CAT Sync</b>	Ενεργοποιεί ή απενεργοποιεί τον συγχρονισμό συχνότητας, τρόπου λειτουργίας και φίλτρου. Η σίγαση κατά την εκπομπή λειτουργεί σε κάθε περίπτωση
<b>Host</b>	Ο υπολογιστής με τον διακομιστή TCI: <code>localhost</code> για αυτόν τον υπολογιστή, αλλιώς η διεύθυνσή του στο τοπικό δίκτυο, π.χ. <code>192.168.1.42</code> . Με Enter ή κλικ έξω από το πεδίο γίνεται νέα σύνδεση

Δεν χρειάζεται να επιλέξετε τίποτε άλλο: οι θύρες **50001** (ExpertSDR3, AetherSDR) και **40001** (ExpertSDR2, Thetis) δοκιμάζονται ταυτόχρονα και ξαναδοκιμάζονται κάθε λίγα δευτερόλεπτα, ώστε η σελίδα να συνδέεται μόλις ξεκινήσει το πρόγραμμα. Τα **CAT Sync** και **Host** τα θυμάται ο browser.

Οι browsers που ζητούν άδεια πριν μια ιστοσελίδα φτάσει στο τοπικό σας δίκτυο — οι πρόσφατες εκδόσεις των Chrome και Edge το κάνουν — ρωτούν μία φορά, την πρώτη φορά που η σελίδα συνδέεται. Επιτρέψτε το, αλλιώς η κουκκίδα μένει γκρι.

## Τι συγχρονίζεται

	Συμπεριφορά
<b>Πομποδέκτης → δέκτης: συχνότητα</b>	Κινείται μόνο η κλίμακα. Zoom, φωτεινότητα και αντίθεση μένουν όπως τα αφήσατε· ο καταρράκτης κυλά μόνο όταν η συχνότητα βγαίνει από την οθόνη, και το bandplan δεν αλλάζει τον τρόπο λειτουργίας
<b>Πομποδέκτης → δέκτης: τρόπος λειτουργίας</b>	Ακολουθεί όταν αλλάζει ο τρόπος του πομποδέκτη — USB, LSB, CW, AM, FM· οι ψηφιακοί τρόποι ως USB ή LSB. Τρόπος που ο δέκτης δεν έχει τον αφήνει ως έχει. Ένας αποκωδικοποιητής σε λειτουργία κρατά τον δικό του τρόπο
<b>Πομποδέκτης → δέκτης: εύρος φίλτρου</b>	Η ζώνη διέλευσης του δέκτη παίρνει το εύρος φίλτρου του πομποδέκτη: η αλλαγή του φίλτρου ή το πλήκτρο <b>FIL</b> στον πομποδέκτη αλλάζει τη ζώνη διέλευσης. Ένας αποκωδικοποιητής σε λειτουργία κρατά τη δική του ζώνη
<b>Δέκτης → πομποδέκτης: συχνότητα</b>	Η πληκτρολόγηση συχνότητας, μια ετικέτα ή σελιδοδείκτης, ένα κλικ στον καταρράκτη και το σύρσιμο της ζώνης διέλευσης μετακινούν το VFO του πομποδέκτη. Στο σύρσιμο στέλνονται μόνο συχνότητες που άλλαξαν
<b>Δέκτης → πομποδέκτης: τρόπος λειτουργίας</b>	Κάθε αλλαγή τρόπου στη σελίδα — ένα πλήκτρο τρόπου, το bandplan, ένας αποκωδικοποιητής, ένας σελιδοδείκτης — ορίζει τον τρόπο του πομποδέκτη (USB, LSB, CW, AM, FM· RADE ως USB ή LSB). Πομποδέκτης σε ψηφιακό τρόπο (USB-D) παραμένει σε αυτόν
<b>Δέκτης → πομποδέκτης: εύρος φίλτρου</b>	Τα πλήκτρα <b>IF Filters</b> , ο ολισθητής IF, το σύρσιμο της ζώνης διέλευσης και η αλλαγή τρόπου ορίζουν το φίλτρο του πομποδέκτη· στο σύρσιμο στέλνεται μόνο το τελικό εύρος. Στους πομποδέκτες Icom η γέφυρα επιλέγει αντί γι' αυτό FIL1, FIL2 ή FIL3 — βλ. <a href="#">Φίλτρα Icom</a>
<b>Χωρίς αντίλαλο</b>	Μια αλλαγή που ήρθε από τον πομποδέκτη δεν στέλνεται ποτέ πίσω σε αυτόν, και μια αναφορά του προηγούμενου τρόπου ή φίλτρου που φτάνει αμέσως μετά από αλλαγή στη σελίδα αγνοείται
<b>Σίγαση κατά την εκπομπή</b>	Πάντα ενεργή, ακόμη και με το CAT Sync κλειστό: η σελίδα σιωπάίνει όσο ο πομποδέκτης εκπέμπει. Αν μετακινήσετε τον ολισθητή έντασης κατά την εκπομπή, μένει σιωπηλή· στη λήψη ο ήχος επιστρέφει στην τρέχουσα θέση του ολισθητή
<b>Δεν υποστηρίζονται</b>	Split, VFO B, RIT/XIT και μετατόπιση transverter. Ακολουθείται μόνο το VFO A του πρώτου δέκτη

Ο τρόπος λειτουργίας και το εύρος φίλτρου στέλνονται με τις εντολές του ίδιου του TCI (`modulation`, `rx_filter_band`), οπότε και τα ExpertSDR, AetherSDR και Thetis θα πρέπει να τα ακολουθούν· έχει δοκιμαστεί μόνο η γέφυρα Hamlib. Για μετατόπιση transverter χρησιμοποιήστε το Desktop PhantomSDR+. Μη χρησιμοποιείτε και τα δύο ταυτόχρονα στον ίδιο πομποδέκτη — δύο ελεγκτές τσακώνονται για την κλίμακα.

## Με ποιους πομποδέκτες λειτουργεί

Πομποδέκτης	Λειτουργεί	Πώς
SunSDR (ExpertSDR2/3), FlexRadio (AetherSDR), ANAN/Hermes (Thetis)	Ναι	Απευθείας — ενεργοποιήστε τον διακομιστή TCI στο πρόγραμμα. Δεν έχει δοκιμαστεί ακόμη με αυτά τα προγράμματα· δοκιμάστηκε με προσομοιωμένο διακομιστή TCI
Πομποδέκτης με θύρα CAT που υποστηρίζει το Hamlib — οι περισσότεροι των Icom, Yaesu, Kenwood, Elecraft, Xiegu, QRP Labs	Ναι	Μέσω της γέφυρας Hamlib παρακάτω. Δοκιμάστηκε σε <b>Icom IC-7300</b> σε Linux και Windows· άλλοι οδηγοί μπορεί να διαφέρουν στα ονόματα τρόπων ή στην αναφορά PTT
Πομποδέκτης χωρίς θύρα CAT, ή που δεν υποστηρίζεται από το Hamlib	Όχι	Δεν υπάρχει από πού να διαβαστεί η συχνότητα

## Η γέφυρα Hamlib (IC-7300 και άλλοι πομποδέκτες χωρίς TCI)

Το `tci-bridge/tci-rigctld.mjs` στον φάκελο του PhantomSDR-Plus κάνει οποιονδήποτε πομποδέκτη μπορεί να ελέγξει το Hamlib διακομιστή TCI για τη σελίδα. Τρέχει στον **δικό σας** υπολογιστή — αυτόν που είναι συνδεδεμένος με τον πομποδέκτη, δίπλα στον browser — όχι στον δέκτη. Τέσσερις φορές το δευτερόλεπτο ρωτά το Hamlib για συχνότητα, τρόπο λειτουργίας, εύρος φίλτρου και κατάσταση εκπομπής, στέλνει στη σελίδα μόνο ό,τι άλλαξε, και περνά στον πομποδέκτη τις αλλαγές συχνότητας, τρόπου και φίλτρου της σελίδας.

Φτάνει στο Hamlib με έναν από δύο τρόπους:

Τρόπος	Αλυσίδα	Πότε
<code>--rigctl</code> (προτείνεται)	πομποδέκτης → <code>rigctl</code> → <code>tci-rigctld.mjs</code> → σελίδα	Κανονικά. Η γέφυρα ξεκινά μόνη της το <code>rigctl</code> του Hamlib: ένα παράθυρο, καμία θύρα δικτύου. <b>Χρησιμοποιήστε αυτόν στα Windows</b> , όπου το <code>rigctld.exe</code> συχνά απορρίπτεται με <i>Access is denied</i>
<code>rigctld</code>	πομποδέκτης → <code>rigctld</code> → <code>tci-rigctld.mjs</code> → σελίδα	Ένα άλλο πρόγραμμα, όπως το WSJT-X, πρέπει να χρησιμοποιεί τον πομποδέκτη ταυτόχρονα — βλ. <a href="#">Κοινή χρήση του πομποδέκτη</a>

### Τι χρειάζεται

	Linux	Windows
Hamlib	<code>sudo apt install libhamlib-utils</code> , ή το πακέτο hamlib της διανομής σας	<code>hamlib-w64-...zip</code> από το <a href="https://github.com/Hamlib/Hamlib/releases">github.com/Hamlib/Hamlib/releases</a> , αποσυμπίεσμένο στο <code>C:\hamlib</code> — το <code>rigctl.exe</code> είναι στο <code>C:\hamlib\bin</code>
Node.js	18 ή νεότερο	Ο LTS Windows Installer (.msi) από το <a href="https://nodejs.org">nodejs.org</a> , με τις προεπιλογές
Το πακέτο <code>ws</code>	Βρίσκεται αυτόματα μέσα στον φάκελο PhantomSDR-Plus	Αντιγράψτε το <code>tci-rigctld.mjs</code> σε φάκελο όπως <code>C:\tci-bridge</code> και τρέξτε εκεί μία φορά <code>npm install ws</code>
Η θύρα του πομποδέκτη	<code>ls /dev/serial/by-id/</code> . Μπείτε μία φορά στην ομάδα <code>dialout</code> : <code>sudo usermod -aG dialout \$USER</code> , μετά αποσύνδεση και ξανά σύνδεση	Εγκαταστήστε τον οδηγό USB του κατασκευαστή· ο αριθμός COM φαίνεται στη <b>Διαχείριση Συσκευών → Θύρες (COM &amp; LPT)</b>

Μόνο ένα πρόγραμμα μπορεί να κρατά τη θύρα CAT του πομποδέκτη. Κλείστε το WSJT-X, το flrig, το JS8Call, το RS-B41 ή τη σύνδεση πομποδέκτη του Desktop PhantomSDR+ πριν ξεκινήσετε τη γέφυρα.

### Παράδειγμα: Icom IC-7300

Στον πομποδέκτη: **MENU → SET → Connectors → CI-V — CI-V USB Baud Rate 115200, CI-V Transceive ON**. Ο αριθμός μοντέλου του Hamlib για το IC-7300 είναι 3073.

**Linux.** Ο πομποδέκτης είναι η γραμμή που περιέχει IC-7300 στο `ls /dev/serial/by-id/`, συνήθως και `/dev/ttyUSB0`.

- Ελέγξτε ότι το Hamlib φτάνει στον πομποδέκτη — πρέπει να τυπώσει τη συχνότητα, π.χ. 14280000:  
`bash rigctl -m 3073 -r /dev/ttyUSB0 -s 115200 f`
- Ξεκινήστε τη γέφυρα και αφήστε την να τρέχει: `bash cd ~/PhantomSDR-Plus/tci-bridge node tci-rigctld.mjs --rigctl rigctl -m 3073 -r /dev/ttyUSB0 -s 115200`

**Windows.** Ο πομποδέκτης εμφανίζεται στη Διαχείριση Συσκευών ως *Silicon Labs CP210x USB to UART Bridge (COM4)* — χρησιμοποιήστε τον δικό σας αριθμό COM. Σε Γραμμή Εντολών:

1. Ελέγξτε ότι το Hamlib φτάνει στον πομποδέκτη — πρέπει να τυπώσει τη συχνότητα: `bat C:\hamlib\bin\rigctl.exe -m 3073 -r COM4 -s 115200 f`
2. Ξεκινήστε τη γέφυρα και αφήστε το παράθυρο ανοιχτό: `bat cd C:\tci-bridge node tci-rigctld.mjs --rigctl C:\hamlib\bin\rigctl.exe -m 3073 -r COM4 -s 115200` Η γραμμή είναι μεγάλη· βεβαιωθείτε ότι τελειώνει πράγματι σε `-s 115200`, αλλιώς το `rigctl` σταματά με *Type: rigctl --help*.

Και στα δύο συστήματα η γέφυρα τυπώνει μέσα σε ένα δευτερόλεπτο:

```
rig answered through rigctl
rig -> page 14280000 Hz
rig -> page mode USB
rig -> page RX
```

Ανοίξτε μετά τη σελίδα του δέκτη σε browser στον ίδιο υπολογιστή, ανοίξτε το **TCI-CAT** και ενεργοποιήστε το **CAT Sync**. Η κουκκίδα γίνεται πράσινη και η γέφυρα τυπώνει `page connected`. Γυρίστε το VFO και ο δέκτης ακολουθεί· κάντε κλικ στον καταρράκτη και ο πομποδέκτης ακολουθεί (`page -> rig ... Hz`)· πατήστε PTT και η σελίδα σωπαίνει. Το **Ctrl+C** σταματά τη γέφυρα και μαζί της το `rigctl`.

Ό,τι ακολουθεί το `--rigctl` είναι το πρόγραμμα `rigctl` με τις δικές του επιλογές — ακριβώς αυτές που λειτούργησαν στον έλεγχο. Αυτός είναι ο κανόνας για κάθε πομποδέκτη: **όταν το `rigctl ... f` τυπώνει τη συχνότητα, η γέφυρα λειτουργεί με τις ίδιες επιλογές.**

### Φίλτρα Icom (FIL1, FIL2, FIL3)

Οι πομποδέκτες Icom όπως το IC-7300 δεν δέχονται οποιοδήποτε εύρος φίλτρου: έχουν τρία φίλτρα, **FIL1**, **FIL2** και **FIL3**, το καθένα με εύρος που ορίζεται στο μενού του πομποδέκτη. Με ένα εύρος, το Hamlib θα επέλεγε ένα από αυτά και θα έγραφε και το εύρος του — και στο IC-7300 αυτό το εύρος μπορεί να καταλήξει στο φίλτρο που ήταν επιλεγμένο πριν, ανακατεύοντας τις ρυθμίσεις. Γι' αυτό στους πομποδέκτες Icom η γέφυρα δεν στέλνει ποτέ εύρος: παίρνει το φίλτρο του οποίου το εύρος αναφοράς είναι πιο κοντά στη ζώνη διέλευσης της σελίδας και απλώς το **επιλέγει**, με την εντολή CI-V του ίδιου του πομποδέκτη. Τα εύρη που έχουν οριστεί στον πομποδέκτη δεν αλλάζουν ποτέ.

Τρόπος	FIL1	FIL2	FIL3
USB, LSB (και οι ψηφιακοί τους τρόποι)	2700 Hz	2400 Hz	1800 Hz
CW	1200 Hz	500 Hz	250 Hz
AM	9000 Hz	6000 Hz	3000 Hz
FM	15000 Hz	10000 Hz	7000 Hz

Στο IC-7300 (`-m 3073`) γίνεται αυτόματα. Ορίστε τα φίλτρα SSB του πομποδέκτη αντίστοιχα — FIL1 2,7 kHz, FIL2 2,4 kHz, FIL3 1,8 kHz (κρατήστε πατημένο το **FIL** στον πομποδέκτη) — και η επιλογή 2,7 / 2,4 / 1,8 kHz στα **IF Filters** επιλέγει FIL1 / FIL2 / FIL3, ενώ το πλήκτρο **FIL** στον πομποδέκτη βάζει τη ζώνη διέλευσης της σελίδας στο εύρος εκείνου του φίλτρου. Η γέφυρα γράφει για παράδειγμα `page -> rig filter 2398 Hz LSB → FIL2 sent`.

Επιλογή	Χρήση
<code>--filters 3000,2400,1800</code>	Άλλα εύρη αναφοράς SSB, με τη σειρά FIL1,FIL2,FIL3· ενεργοποιεί επίσης την επιλογή φίλτρου για άλλον πομποδέκτη Icom
<code>--civ A4</code>	Η διεύθυνση CI-V του πομποδέκτη σε δεκαεξαδική μορφή, όταν δεν είναι 94 (IC-705 A4, IC-9700 A2, IC-7610 98)
<code>--filters off</code>	Αποστολή εύρους μέσω Hamlib, όπως για τους άλλους κατασκευαστές

Και οι δύο επιλογές μπαίνουν πριν από το `--rigctl`, π.χ. `node tci-rigctld.mjs --filters 3000,2400,1800 --civ A4 --rigctl rigctl -m 3085 -r /dev/ttyACM0 -s 19200`. Δοκιμάστηκε μόνο σε IC-7300. Οι άλλοι κατασκευαστές — Yaesu, Kenwood, Elecraft και οι υπόλοιποι — παίρνουν το εύρος της σελίδας μέσω Hamlib, που το ορίζει όσο πιο κοντά επιτρέπει ο πομποδέκτης.

## Άλλοι πομποδέκτες

Τα ίδια βήματα ισχύουν για κάθε πομποδέκτη που υποστηρίζει το Hamlib· αλλάζουν μόνο οι επιλογές.

- Προετοιμάστε τον πομποδέκτη.** Στο μενού του σημειώστε την ταχύτητα CAT (ή CI-V) και ενεργοποιήστε ρύθμιση όπως *CAT μέσω USB* ή *CI-V transceive*, αν υπάρχει. Στα Windows εγκαταστήστε τον οδηγό USB του κατασκευαστή.
- Βρείτε τον αριθμό μοντέλου του Hamlib** στη λίστα που τυπώνει το `rigctl -l`:
- Linux: `rigctl -l | grep -i 991`
- Windows: `C:\hamlib\bin\rigctl.exe -l | findstr /i 991`
- Βρείτε τη θύρα.** Linux: `ls /dev/serial/by-id/`. Windows: Διαχείριση Συσκευών. Μερικοί πομποδέκτες δημιουργούν **δύο** θύρες — τα FT-991A, FTDX10 και FT-710 της Yaesu τις ονομάζουν *Enhanced* και *Standard*· το CAT είναι στην **Enhanced**.
- Ελέγξτε και μετά ξεκινήστε τη γέφυρα** με `-m <μοντέλο> -r <θύρα> -s <ταχύτητα>`, όπως στο παράδειγμα του IC-7300: πρώτα `rigctl -m ... -r ... -s ... f`, μετά τις ίδιες επιλογές μετά το `--rigctl`.

Μερικοί αριθμοί μοντέλων, από το Hamlib 4.5 — επιβεβαιώστε τους με `rigctl -l`, γιατί άλλη έκδοση του Hamlib μπορεί να αριθμεί έναν πομποδέκτη διαφορετικά:

Πομποδέκτης	-m	Πομποδέκτης	-m
Icom IC-7300	3073	Yaesu FT-991 / FT-991A	1035
Icom IC-705	3085	Yaesu FTDX10	1042
Icom IC-7610	3078	Yaesu FT-710	1049
Icom IC-9700	3081	Yaesu FT-891	1036
Xiegu G90	3088	Yaesu FT-817	1020
Xiegu X6100	3087	Kenwood TS-590SG	2037
Elecraft K3 / K3S	2029	Kenwood TS-890S	2041
Elecraft KX3	2045	Kenwood TS-2000	2014
Elecraft K4	2047	QRP Labs QCX / QDX	2052

Επιπλέον επιλογές, μόνο όταν χρειάζονται:

- Πομποδέκτης Icom με αλλαγμένη διεύθυνση CI-V:** προσθέστε `-c` με τη διεύθυνση σε δεκαδική μορφή — το 94h είναι `-c 148`.

- **Ρύθμιση που προσφέρει το Hamlib για τον πομποδέκτη:** το `rigctl -m <μοντέλο> -L` τις δείχνει· ορίστε μία με `-C όνομα=τιμή`, π.χ. `-C post_write_delay=10` για αργή διεπαφή.
- **Πομποδέκτης που εκπέμπει ή κάνει επανεκκίνηση όταν ανοίγει η θύρα:** μερικά καλώδια CAT χρησιμοποιούν το DTR ή το RTS για PTT· προσθέστε `-C dtr_state=OFF -C rts_state=OFF`.

### Παράδειγμα: Yaesu FT-991A

Δεν έχει δοκιμαστεί σε πραγματικό FT-991A — ακολουθεί τις ρυθμίσεις του Hamlib για τον πομποδέκτη· ο έλεγχος `rigctl ... f` δείχνει αμέσως αν λειτουργεί.

Στον πομποδέκτη ορίστε το **CAT RATE** (μενού 031) σε 38400. Ο αριθμός μοντέλου του Hamlib για τα FT-991 και FT-991A είναι 1035.

Το καλώδιο USB του FT-991A δημιουργεί **δύο** σειριακές θύρες. Το CAT είναι στην **Enhanced**:

- **Linux:** το `ls /dev/serial/by-id/` δείχνει δύο γραμμές για τον πομποδέκτη· αυτή που τελειώνει σε `-if00-port0` είναι η Enhanced, συνήθως `/dev/ttyUSB0`.
- **Windows:** η Διαχείριση Συσκευών δείχνει *Silicon Labs Dual CP2105 USB to UART Bridge: Enhanced COM Port (COM5)* και ένα *Standard COM Port*· χρησιμοποιήστε τον αριθμό της Enhanced.

#### Linux:

```
rigctl -m 1035 -r /dev/ttyUSB0 -s 38400 f
cd ~/PhantomSDR-Plus/tci-bridge
node tci-rigctld.mjs --rigctl rigctl -m 1035 -r /dev/ttyUSB0 -s 38400
```

#### Windows:

```
C:\hamlib\bin\rigctl.exe -m 1035 -r COM5 -s 38400 f
cd C:\tci-bridge
node tci-rigctld.mjs --rigctl C:\hamlib\bin\rigctl.exe -m 1035 -r COM5 -s 38400
```

Αν ο έλεγχος δεν τυπώσει κάτι χρήσιμο, η συνήθης αιτία είναι η Standard θύρα αντί της Enhanced, ή CAT RATE διαφορετικό από το `-s`.

### Παράδειγμα: Kenwood TS-590SG

Δεν έχει δοκιμαστεί σε πραγματικό TS-590SG — ακολουθεί τις ρυθμίσεις του Hamlib για τον πομποδέκτη· ο έλεγχος `rigctl ... f` δείχνει αμέσως αν λειτουργεί.

Στο μενού του πομποδέκτη ορίστε την ταχύτητα της θύρας **USB** σε 115200 (ο αριθμός μενού είναι στο εγχειρίδιο του TS-590SG). Ο αριθμός μοντέλου του Hamlib είναι 2037· το παλαιότερο TS-590S είναι 2031. Στα Windows εγκαταστήστε πρώτα τον οδηγό εικονικής θύρας COM της Kenwood για τη θύρα USB.

Το καλώδιο USB δημιουργεί **μία** σειριακή θύρα:

- **Linux:** η γραμμή του πομποδέκτη στο `ls /dev/serial/by-id/`, συνήθως `/dev/ttyUSB0`.
- **Windows:** Διαχείριση Συσκευών → Θύρες (COM & LPT), για παράδειγμα `COM6`.

#### Linux:

```
rigctl -m 2037 -r /dev/ttyUSB0 -s 115200 f
cd ~/PhantomSDR-Plus/tci-bridge
node tci-rigctld.mjs --rigctl rigctl -m 2037 -r /dev/ttyUSB0 -s 115200
```

#### Windows:

```
C:\hamlib\bin\rigctl.exe -m 2037 -r COM6 -s 115200 f
cd C:\tci-bridge
node tci-rigctld.mjs --rigctl C:\hamlib\bin\rigctl.exe -m 2037 -r COM6 -s 115200
```

Αν ο έλεγχος αποτύχει, η συνήθης αιτία είναι ταχύτητα USB διαφορετική από το `-s`, ή καλώδιο στην υποδοχή RS-232 (COM) ενώ το `-r` δείχνει τη θύρα USB.

### Περισσότερα παραδείγματα

Κανένα από αυτά δεν έχει δοκιμαστεί σε πραγματικό υλικό. Κάθε γραμμή δίνει τις επιλογές που μπαίνουν μετά το `rigctl` στον έλεγχο και μετά το `--rigctl rigctl` στη γέφυρα· ορίστε στο μενού του πομποδέκτη την ίδια ταχύτητα. Η θύρα Linux είναι η συνηθισμένη — επιβεβαιώστε την με `ls /dev/serial/by-id/`. Στα Windows αντικαταστήστε τη θύρα με τον αριθμό COM από τη Διαχείριση Συσκευών και το `rigctl` με `C:\hamlib\bin\rigctl.exe`.

Πομποδέκτης	Μενού πομποδέκτη	Επιλογές (Linux)	Σημειώσεις
Icom IC-705 (USB)	CI-V USB Baud Rate 19200	-m 3085 -r /dev/ttyACM0 -s 19200	Εμφανίζονται δύο θύρες· το CI-V είναι η πρώτη (ttyACM0)
Icom IC-7100 (USB)	CI-V USB Baud Rate 19200	-m 3070 -r /dev/ttyUSB0 -s 19200	Η μέγιστη ταχύτητα του Hamlib για αυτόν τον πομποδέκτη είναι 19200
Icom IC-7610	CI-V USB Baud Rate 115200	-m 3078 -r /dev/ttyUSB0 -s 115200	
Icom IC-9700	CI-V USB Baud Rate 38400	-m 3081 -r /dev/ttyUSB0 -s 38400	
Yaesu FTDX101D / FTDX101MP	CAT RATE 38400	-m 1040 -r /dev/ttyUSB0 -s 38400	-m 1044 για το FTDX101MP. Δύο θύρες· χρησιμοποιήστε την Enhanced
Yaesu FTDX10	CAT RATE 38400	-m 1042 -r /dev/ttyUSB0 -s 38400	Δύο θύρες· χρησιμοποιήστε την Enhanced, όπως στο FT-991A
Yaesu FT-710	CAT RATE 38400	-m 1049 -r /dev/ttyUSB0 -s 38400	Δύο θύρες· χρησιμοποιήστε την Enhanced, όπως στο FT-991A
Yaesu FT-891	CAT RATE 38400	-m 1036 -r /dev/ttyUSB0 -s 38400	Δύο θύρες· χρησιμοποιήστε την Enhanced, όπως στο FT-991A
Yaesu FT-450D	CAT RATE 38400	-m 1046 -r /dev/ttyUSB0 -s 38400	Υποδοχή RS-232: χρησιμοποιήστε μετατροπέα USB-σειριακής
Yaesu FT-817 / FT-818	CAT RATE 38400	-m 1020 -r /dev/ttyUSB0 -s 38400	-m 1041 για το FT-818. Χρειάζεται καλώδιο CAT στην υποδοχή ACC
Yaesu FT-857 / FT-897	CAT RATE 38400	-m 1022 -r /dev/ttyUSB0 -s 38400	-m 1023 για το FT-897. Χρειάζεται καλώδιο CAT
Kenwood TS-890S (USB)	Ταχύτητα USB 115200	-m 2041 -r /dev/ttyUSB0 -s 115200	
Kenwood TS-990S (USB)	Ταχύτητα USB 115200	-m 2039 -r /dev/ttyUSB0 -s 115200	
Kenwood TS-590SG / TS-590S (USB)	Ταχύτητα USB 115200	-m 2037 -r /dev/ttyUSB0 -s 115200	Αναλυτικό παράδειγμα παραπάνω
Kenwood TS-480	Ταχύτητα θύρας COM 57600	-m 2028 -r /dev/ttyUSB0 -s 57600	Υποδοχή RS-232: χρησιμοποιήστε μετατροπέα USB-σειριακής
Kenwood TS-2000	Ταχύτητα θύρας COM 57600	-m 2014 -r /dev/ttyUSB0 -s 57600	Υποδοχή RS-232· η μέγιστη ταχύτητα του Hamlib για αυτόν τον πομποδέκτη είναι 57600
Elecraft K4 (USB)	Ταχύτητα RS232 115200	-m 2047 -r /dev/ttyUSB0 -s 115200	
Elecraft K3 / K3S	Ταχύτητα RS232 38400	-m 2029 -r /dev/ttyUSB0 -s 38400	K3S: USB· K3: σειριακή θύρα ή καλώδιο KUSB
Elecraft KX3	Ταχύτητα RS232 38400	-m 2045 -r /dev/ttyUSB0 -s 38400	Καλώδιο KXUSB
Elecraft KX2	Ταχύτητα RS232 38400	-m 2044 -r /dev/ttyUSB0 -s 38400	Καλώδιο KXUSB
Xiegu G90			

Πομποδέκτης	Μενού πομποδέκτη	Επιλογές (Linux)	Σημειώσεις
	Ταχύτητα CI-V 19200	<code>-m 3088 -r /dev/ttyUSB0 -s 19200</code>	Το Hamlib χρησιμοποιεί την προεπιλεγμένη διεύθυνση CI-V του G90
Xiegu X6100	Ταχύτητα CI-V 19200	<code>-m 3087 -r /dev/ttyUSB0 -s 19200</code>	Η μέγιστη ταχύτητα του Hamlib για αυτόν τον πομποδέκτη είναι 19200
Lab599 TX-500	—	<code>-m 2050 -r /dev/ttyUSB0 -s 9600</code>	Το Hamlib χρησιμοποιεί μόνο 9600
ELAD FDM-DUO	—	<code>-m 33001 -r /dev/ttyUSB0 -s 115200</code>	
QRP Labs QDX	—	<code>-m 2052 -r /dev/ttyACM0 -s 9600</code>	Σειριακή θύρα USB· η ταχύτητα δεν έχει σημασία, αλλά πρέπει να δοθεί

Πλήρης εντολή Linux για το IC-705, ως παράδειγμα για το πώς διαβάζεται μια γραμμή:

```
rigctl -m 3085 -r /dev/ttyACM0 -s 19200 f
node tci-rigctld.mjs --rigctl rigctl -m 3085 -r /dev/ttyACM0 -s 19200
```

### Πομποδέκτες που ελέγχονται από άλλο πρόγραμμα

Μερικοί πομποδέκτες ελέγχονται ήδη από ένα πρόγραμμα που μπορεί να κάνει τη δουλειά — μερικές φορές χωρίς καθόλου γέφυρα. Κανένας από αυτούς τους τρόπους δεν έχει δοκιμαστεί.

Πομποδέκτης και πρόγραμμα	Τι να κάνετε
<b>SunSDR</b> με ExpertSDR2/3, <b>FlexRadio</b> με AetherSDR, <b>Apache Labs ANAN / Hermes</b> με Thetis	Χωρίς γέφυρα. Ενεργοποιήστε τον <b>διακομιστή TCI</b> του προγράμματος και χρησιμοποιήστε απευθείας το <b>TCI-CAT</b> — οι θύρες 50001 και 40001 βρίσκονται αυτόματα
<b>FlexRadio</b> με SmartSDR (Windows)	Προσθέστε μια θύρα στο <b>SmartSDR CAT</b> . Μιλά Kenwood CAT, οπότε μια σειριακή θύρα εκεί λειτουργεί ως TS-2000: <code>-m 2014 -r COM8 -s 57600</code> . Μια θύρα TCP λειτουργεί με το μοντέλο FlexRadio του Hamlib: <code>-m 2036 -r 127.0.0.1:&lt;θύρα&gt;</code>
<b>Κάθε πομποδέκτης που ελέγχει το flrig</b>	Αφήστε το flrig να τρέχει και χρησιμοποιήστε το μοντέλο flrig του Hamlib· ο πομποδέκτης μένει κοινός με fldigi, WSJT-X και προγράμματα ημερολογίου: <code>-m 4 -r 127.0.0.1:12345</code>
<b>Ένα rigctld σε λειτουργία</b> , ή πρόγραμμα με διακομιστή <i>Hamlib NET rigctl</i>	Ξεκινήστε τη γέφυρα χωρίς <code>--rigctl</code> , με αυτή τη διεύθυνση: <code>node tci-rigctld.mjs 50001 127.0.0.1:4532</code>

Όπως παντού, οι επιλογές μπαίνουν μετά το `rigctl` στον έλεγχο και μετά το `--rigctl rigctl` στη γέφυρα. Μια διεύθυνση δικτύου όπως `127.0.0.1:12345` δεν χρειάζεται `-s`.

### Κοινή χρήση του πομποδέκτη: ο τρόπος rigctld

Όταν το WSJT-X ή ένα πρόγραμμα ημερολογίου πρέπει να χρησιμοποιεί τον πομποδέκτη όσο τρέχει η γέφυρα, ξεκινήστε τον διακομιστή του Hamlib `rigctld` με τις ίδιες επιλογές, αφήστε τον να τρέχει και συνδέστε και τα δύο προγράμματα σε αυτόν:

```
rigctld -m 3073 -r /dev/ttyUSB0 -s 115200
node tci-rigctld.mjs
```

Χωρίς `--rigctl` η γέφυρα ψάχνει το `rigctld` στο `127.0.0.1:4532` και τυπώνει `rigctld connected`· το `node tci-rigctld.mjs 50001 192.168.1.50:4532` χρησιμοποιεί `rigctl` σε άλλον υπολογιστή. Στο WSJT-X

επιλέξτε `rig Hamlib NET rigctl` στην ίδια διεύθυνση. Στα Windows ο διακομιστής είναι το `C:\hamlib\bin\rigctld.exe` με τις ίδιες επιλογές· αν τα Windows απαντήσουν *Access is denied*, χρησιμοποιήστε `--rigctl` και κλείστε το άλλο πρόγραμμα όσο τρέχει η γέφυρα.

**Θύρες.** Η γέφυρα προσφέρει TCI στη θύρα 50001· άλλη θύρα μπαίνει πρώτη στη γραμμή εντολών, π.χ. `node tci-rigctld.mjs 40001 --rigctl ...`. Αν η γέφυρα και ο browser είναι σε διαφορετικούς υπολογιστές, βάλτε στο **Host** τη διεύθυνση του υπολογιστή της γέφυρας.

## Αντιμετώπιση προβλημάτων TCI-CAT

Σύμπτωμα	Πιθανή αιτία	Τι να κάνετε
Η κουκκίδα μένει γκρι	Ο διακομιστής TCI ή η γέφυρα δεν τρέχει· λάθος <b>Host</b> · αρνηθήκατε την άδεια τοπικού δικτύου του browser	Ενεργοποιήστε το TCI στο ExpertSDR/Thetis/AetherSDR ή ξεκινήστε τη γέφυρα· ελέγξτε το <b>Host</b> · επιτρέψτε την πρόσβαση στο τοπικό δίκτυο στις ρυθμίσεις του ιστότοπου
Το <code>rigctl ... f</code> βγάζει σφάλμα ή περιμένει	Λάθος θύρα ή ταχύτητα· χωρίς δικαίωμα dialout (Linux) ή χωρίς οδηγό USB (Windows)· άλλο πρόγραμμα κρατά τη θύρα	Ταιριάξτε το <code>-s</code> με το μενού του πομποδέκτη· ελέγξτε τη θύρα ( <code>ls /dev/serial/by-id/</code> , Διαχείριση Συσκευών)· κλείστε άλλα προγράμματα πομποδέκτη
Η γέφυρα γράφει <code>rigctl stopped ... Type: rigctl --help</code>	Η εντολή είναι ελλιπής — συνήθως λείπει η ταχύτητα μετά το <code>-s</code>	Πληκτρολογήστε ξανά ολόκληρη τη γραμμή
Η γέφυρα γράφει <i>The value after -s is missing</i>	Η γραμμή εντολών κόπηκε κατά την επικόλληση	Πληκτρολογήστε ξανά το τέλος της γραμμής, π.χ. <code>-s 115200</code>
Το φίλτρο δεν ακολουθεί προς τη μία ή την άλλη κατεύθυνση	Το <b>CAT Sync</b> είναι κλειστό, τρέχει αποκωδικοποιητής στη σελίδα, ή (IC-7300) τα εύρη FIL του πομποδέκτη διαφέρουν από 2700 / 2400 / 1800 Hz	Ανοίξτε το CAT Sync· σταματήστε τον αποκωδικοποιητή· ορίστε τα εύρη FIL στον πομποδέκτη, ή δώστε <code>--filters</code> με τα δικά του εύρη
Η γέφυρα γράφει <code>rigctl stopped ... rig_open: error</code>	Το <code>rigctl</code> δεν μπορεί να ανοίξει τη θύρα	Όπως για το <code>rigctl ... f</code> παραπάνω
Η γέφυρα γράφει <i>the rig is not answering</i>	Η θύρα ανοίγει αλλά ο πομποδέκτης δεν απαντά: λάθος ταχύτητα ή μοντέλο, CAT κλειστό στο μενού, ή αλλαγμένη διεύθυνση CI-V της Icom	Επαναλάβετε τον έλεγχο <code>rigctl ... f</code> · προσθέστε <code>-c</code> για αλλαγμένη διεύθυνση CI-V
Η γέφυρα γράφει <code>rigctld not reachable</code>	Ξεκίνησε χωρίς <code>--rigctl</code> και δεν τρέχει <code>rigctld</code>	Προσθέστε <code>--rigctl ...</code> , ή ξεκινήστε πρώτα το <code>rigctld</code>
Το <code>rigctld.exe</code> λέει <i>Access is denied</i> (Windows)	Τα Windows αρνούνται να το εκτελέσουν	Χρησιμοποιήστε <code>--rigctl</code> — χρειάζεται μόνο το <code>rigctl.exe</code>
Συνδεδεμένο, αλλά ο δέκτης δεν κινείται	Το <b>CAT Sync</b> είναι κλειστό	Ανοίξτε το — η σίγαση κατά την εκπομπή λειτουργεί και χωρίς αυτό
Ο τρόπος του πομποδέκτη δεν ακολουθεί	Τρόπος χωρίς αντίστοιχο στον δέκτη, ή ο οδηγός Hamlib αναφέρει ασυνήθιστο όνομα	Η συχνότητα συγχρονίζεται κανονικά· ορίστε τον τρόπο στη σελίδα
Δεν γίνεται σίγαση κατά την εκπομπή	Ο πομποδέκτης ή ο οδηγός Hamlib του δεν αναφέρει κατάσταση εκπομπής	Δεν υπάρχει κάτι να ρυθμιστεί στη σελίδα
Η κλίμακα πηγαίνει οριζόντια	Το Desktop PhantomSDR+ ή άλλο πρόγραμμα συγχρονίζει επίσης τον πομποδέκτη	Ένας ελεγκτής κάθε φορά

## Για διαχειριστές δεκτών

Τίποτα προς ρύθμιση. Ο έλεγχος πομποδέκτη χρησιμοποιεί μια μικρή διεπαφή JavaScript που κάθε σελίδα PhantomSDR-Plus ήδη έχει· δεν χρειάζεται ρύθμιση server, ανοιχτή θύρα ή δικαίωμα διαχειριστή. Οι λειτουργίες φίλτρου και σίγασης ήρθαν με την έκδοση 4.1.0 — μετά την εφαρμογή της, ξαναχτίστε το frontend (./recompile.sh, επιλογή 2)· ο δέκτης δεν χρειάζεται να σταματήσει. Ούτε οι δέκτες KiwiSDR, WebSDR και UberSDR χρειάζονται κάτι: η εφαρμογή χρησιμοποιεί τα χειριστήρια που ήδη έχουν οι σελίδες τους.

Το κουμπί **TCI-CAT** ήρθε επίσης με την 4.1.0, και πάλι μόνο με νέο χτίσιμο του frontend. Δεν ανοίγει καμία θύρα στον δέκτη: η σύνδεση γίνεται από τον browser κάθε ακροατή προς τον δικό του υπολογιστή. Η γέφυρα Hamlib, tci-bridge/tci-rigctl.js, είναι για να την τρέχουν οι ακροατές στο σπίτι τους· ο δέκτης δεν τη χρησιμοποιεί.

## Για προγραμματιστές: η διεπαφή της σελίδας

Και το Desktop PhantomSDR+ και το CATsync Tool χρησιμοποιούν αυτές τις συναρτήσεις, που κάθε σελίδα PhantomSDR-Plus βάζει στο window μόλις φορτώσει (οι σελίδες KiwiSDR, WebSDR και UberSDR χειρίζονται μέσω των δικών τους, διαφορετικών χειριστηρίων):

Συνάρτηση	Επιστρέφει / κάνει
catsync_ready	true μόλις εγκατασταθούν οι παρακάτω συναρτήσεις
catsync_getFrequency()	Συχνότητα συντονισμού, Hz
catsync_setFrequency(hz)	Συντονισμός στο hz
catsync_getMode()	USB, LSB, CW, CW-L, AM, QUAM, FM, WBFM, RADEU, RADEL
catsync_setMode(mode)	Ορίζει τον τρόπο· επαναφέρει τη ζώνη διέλευσης στην προεπιλογή του τρόπου
catsync_getBandwidth()	Συνολικό εύρος διέλευσης, Hz
catsync_setBandwidth(hz)	Ορίζει το εύρος — μεγαλώνει προς τα πάνω στο USB, προς τα κάτω στο LSB, συμμετρικά αλλού. Καλέστε τη <b>μετά</b> το catsync_setMode
catsync_getMute()	true όταν είναι σε σίγαση
catsync_setMute(on)	Σίγαση ή κατάργηση, μέσω του κουμπιού σίγασης της σελίδας

Οι τέσσερις τελευταίες ήρθαν με την 4.1.0, οπότε ελέγξτε πριν τις καλέσετε:

```
if (window.catsync_ready) {
 window.catsync_setFrequency(7074000)
 window.catsync_setMode('USB')
 if (typeof window.catsync_setBandwidth === 'function') window.catsync_setBandwidth(2400)
}
```

Ο ορισμός συχνότητας επανασυντονίζει τον ήχο, οπότε καλέστε έναν setter μόνο όταν η τιμή πράγματι αλλάξει — ένας setter που καλείται συνεχώς με την ίδια τιμή ακούγεται. Τα παλαιότερα σημεία εισόδου τύπου KiwiSDR/WebSDR (setfreq, set\_mode, freqset\_complete) υπάρχουν ακόμη για εργαλεία που τα περιμένουν.

## Αντιμετώπιση προβλημάτων

Σύμπτωμα	Πιθανή αιτία	Τι να κάνετε
<i>Could not open ttyUSB0</i> (Linux)	Δεν είστε στην ομάδα <code>dialout</code> , ή άλλο πρόγραμμα έχει τη θύρα	<code>sudo usermod -aG dialout \$USER</code> , αποσύνδεση και σύνδεση· κλείστε WSJT-X, προγράμματα ημερολογίου, εργαλεία πομποδέκτη
Η ένδειξη δέκτη λέει <i>not a receiver this app can drive</i>	Άλλο είδος web δέκτη, ή η σελίδα φορτώνει ακόμη	Υποστηρίζονται PhantomSDR-Plus, KiwiSDR, PA3FWM WebSDR και UberSDR· δώστε σε αργή σελίδα λίγα δευτερόλεπτα
<i>The rig did not answer</i>	Λάθος ταχύτητα, λάθος τύπος πομποδέκτη, λάθος διεύθυνση CI-V, πομποδέκτης κλειστός	Ταιριάξτε την ταχύτητα με το μενού· ελέγξτε τη διεύθυνση CI-V· δοκιμάστε Hamlib
<i>Hamlib is not installed</i>	Δεν βρέθηκε <code>rigctld</code>	Linux: <code>sudo apt install libhamlib-utils</code> . Ή ορίστε τη διαδρομή του στο <b>rigctld program</b>
<i>rigctld stopped: ...</i>	Το Hamlib δεν άνοιξε τον πομποδέκτη — ακολουθεί το δικό του μήνυμα	Συνήθως θύρα ή ταχύτητα· το κείμενο μετά την άνω-κάτω τελεία είναι η αιτία κατά το Hamlib
<i>flrig is not running at ...</i>	Το flrig είναι κλειστό, ή η θύρα XML-RPC διαφέρει	Ξεκινήστε το flrig· ελέγξτε τη θύρα στις ρυθμίσεις του
Συνδεδεμένο, αλλά ο δέκτης δεν κινείται	Κανένα παράθυρο σταθμού ανοιχτό, ή το <b>Receiver window</b> δείχνει σε κλειστό σταθμό	Ανοίξτε τον σταθμό ή επιλέξτε <i>The station window last in front</i>
Ο πομποδέκτης εκπέμπει κατά τη σύνδεση	Το DTR ή το RTS ενεργοποιεί τον πομπό μέσω της διεπαφής σας	Ξετσεκάρτε <b>DTR on</b> και <b>RTS on</b>
Το φίλτρο δεν ακολουθεί	Δέκτης παλαιότερος από την 4.1.0, ή ενσωματωμένος οδηγός Kenwood/Yaesu	Συχνότητα και τρόπος συγχρονίζονται κανονικά· για φίλτρο σε Kenwood/Yaesu χρησιμοποιήστε Hamlib
Η σίγαση κατά την εκπομπή δεν κάνει τίποτα	Δέκτης χωρίς την ενημέρωση, ή ο πομποδέκτης δεν αναφέρει κατάσταση εκπομπής	Όπως παραπάνω
<i>Lost the rig ... reconnecting</i>	Αποσυνδέθηκε το καλώδιο, έκλεισε ο πομποδέκτης ή σταμάτησε το <code>rigctld</code>	Τίποτα — ξαναδοκιμάζει κάθε 3 δευτερόλεπτα και συνεχίζει μόλις επιστρέψει
Οι δύο πλευρές πηδούν συνεχώς	Δύο προγράμματα ελέγχουν τον πομποδέκτη ταυτόχρονα	Αφήστε μόνο ένα πρόγραμμα να ρυθμίζει τον πομποδέκτη, ή μοιραστείτε τον μέσω flrig

## Γνωστοί περιορισμοί

- Ένας πομποδέκτης, ένα παράθυρο δέκτη κάθε φορά.
- Δέκτες εκτός από PhantomSDR-Plus, KiwiSDR, PA3FWM WebSDR και UberSDR — π.χ. OpenWebRX — δεν υποστηρίζονται.
- Οι ενσωματωμένοι οδηγοί ακολουθούν τα δημοσιευμένα πρωτόκολλα των κατασκευαστών και δοκιμάστηκαν σε προσομοιωμένους πομποδέκτες και πραγματικό Hamlib· για πομποδέκτη που συμπεριφέρεται διαφορετικά, λύση είναι το Hamlib.
- Λειτουργία split, VFO B, RIT/XIT και κανάλια μνήμης δεν συγχρονίζονται — μόνο η συχνότητα του ενεργού VFO.

- Το TCI-CAT στη σελίδα του δέκτη δεν συγχρονίζει split, VFO B, RIT/XIT ή μετατόπιση transverter, και έχει δοκιμαστεί μόνο σε IC-7300 μέσω της γέφυρας Hamlib.
- Σε Linux τα πακέτα χρησιμοποιούν το Hamlib της διανομής· δεν περιλαμβάνεται δικό τους.

## ΚΕΦΑΛΑΙΟ 16

# Δομή του έργου

---

PROJECT\_STRUCTURE.md

## Δομή του έργου PhantomSDR-Plus

Το έγγραφο αυτό παρέχει πλήρη επισκόπηση της δομής καταλόγων του PhantomSDR-Plus, της οργάνωσης των αρχείων και των σχέσεων μεταξύ των στοιχείων.

---

### Πίνακας περιεχομένων

---

1. Δέντρο καταλόγων
  2. Ριζικός κατάλογος
  3. Πηγαίος κώδικας (`src/`)
  4. Frontend (`frontend/`)
  5. Λίστες συχνοτήτων (`frequencylist/`)
  6. Αρχεία ρυθμίσεων
  7. Σύστημα μεταγλώττισης
-

## Δέντρο καταλόγων

---

```

PhantomSDR-Plus
├── add-receiver.sh # προσθέτει έναν ακόμη δέκτη: οδηγός, instances/<όνομα>/,
receivers.toml – δείτε docs/MULTI_RECEIVER.md
├── admin_config.json
├── admin_server.py
├── autorun
│ ├── audiotap.js
│ ├── bandplan.js
│ ├── decodeworker.js
│ ├── index.js
│ ├── manager.js
│ ├── package.json
│ ├── pool.js
│ ├── probe-ft8.js
│ ├── probe-js8.js
│ ├── pskreporter.js
│ ├── spotparse.js
│ ├── wasm-shim.js
│ └── wsprnet.js
├── cf32_to_real.c # Fobos HF: κρατά το I από το CF32 του rx_sdr, κλιμακωμένο σε s16
(το χτίζει το setup-fobos.sh)
├── chat_history.txt
├── check-go.sh # παλιό: watchdog της αλυσίδας go.sh
├── config-airspyhf.toml
├── config.example.hackrf.toml
├── config.example.rtlsdr.toml
├── config-fobos-hf.toml
├── config-fobos.toml
├── config-hackrf.toml
├── config-rspla.toml
├── config-rtl.toml
├── config-rx888mk2.toml
├── config.toml
├── connection_impl.hpp
├── docs
│ ├── ADMIN_PANEL_SETUP.md
│ ├── Aether_config.md
│ ├── CONNECTION_LIMITS.md
│ ├── de
│ │ ├── ADMIN_PANEL_SETUP.md
│ │ ├── Aether_config.md
│ │ ├── CONNECTION_LIMITS.md
│ │ ├── DECODERS.md
│ │ ├── EDITING_VARIANTS.md
│ │ ├── INSTALLATION.md
│ │ ├── MULTI_RECEIVER.md
│ │ ├── PhantomSDR-Plus-Documentation-DE.pdf
│ │ ├── PROJECT_STRUCTURE.md
│ │ ├── RADE_General_INSTALL_MANUAL_LINUX.md
│ │ ├── RADE_README.md
│ │ ├── README.md
│ │ ├── RECEIVE_DIVERSITY.md
│ │ ├── RIG_CONTROL.md
│ │ ├── THERMAL_GUARD.md
│ │ └── USER_GUIDE.md
│ ├── DECODERS.md
│ ├── EDITING_VARIANTS.md
│ └── el
│ ├── ADMIN_PANEL_SETUP.md
│ ├── Aether_config.md
│ ├── CONNECTION_LIMITS.md
│ ├── DECODERS.md
│ ├── EDITING_VARIANTS.md
│ └── INSTALLATION.md

```

- └─ MULTI\_RECEIVER.md
- └─ PhantomSDR-Plus-Documentation-EL.pdf
- └─ PROJECT\_STRUCTURE.md
- └─ RADE\_General\_INSTALL\_MANUAL\_LINUX.md
- └─ RADE\_README.md
- └─ README.md
- └─ RECEIVE\_DIVERSITY.md
- └─ RIG\_CONTROL.md
- └─ THERMAL\_GUARD.md
- └─ USER\_GUIDE.md
- └─ es
  - └─ ADMIN\_PANEL\_SETUP.md
  - └─ Aether\_config.md
  - └─ CONNECTION\_LIMITS.md
  - └─ DECODERS.md
  - └─ EDITING\_VARIANTS.md
  - └─ INSTALLATION.md
  - └─ MULTI\_RECEIVER.md
  - └─ PhantomSDR-Plus-Documentation-ES.pdf
  - └─ PROJECT\_STRUCTURE.md
  - └─ RADE\_General\_INSTALL\_MANUAL\_LINUX.md
  - └─ RADE\_README.md
  - └─ README.md
  - └─ RECEIVE\_DIVERSITY.md
  - └─ RIG\_CONTROL.md
  - └─ THERMAL\_GUARD.md
  - └─ USER\_GUIDE.md
- └─ fr
  - └─ ADMIN\_PANEL\_SETUP.md
  - └─ Aether\_config.md
  - └─ CONNECTION\_LIMITS.md
  - └─ DECODERS.md
  - └─ EDITING\_VARIANTS.md
  - └─ INSTALLATION.md
  - └─ MULTI\_RECEIVER.md
  - └─ PhantomSDR-Plus-Documentation-FR.pdf
  - └─ PROJECT\_STRUCTURE.md
  - └─ RADE\_General\_INSTALL\_MANUAL\_LINUX.md
  - └─ RADE\_README.md
  - └─ README.md
  - └─ RECEIVE\_DIVERSITY.md
  - └─ RIG\_CONTROL.md
  - └─ THERMAL\_GUARD.md
  - └─ USER\_GUIDE.md
- └─ hr
  - └─ ADMIN\_PANEL\_SETUP.md
  - └─ Aether\_config.md
  - └─ CONNECTION\_LIMITS.md
  - └─ DECODERS.md
  - └─ EDITING\_VARIANTS.md
  - └─ INSTALLATION.md
  - └─ MULTI\_RECEIVER.md
  - └─ PhantomSDR-Plus-Documentation-HR.pdf
  - └─ PROJECT\_STRUCTURE.md
  - └─ RADE\_General\_INSTALL\_MANUAL\_LINUX.md
  - └─ RADE\_README.md
  - └─ README.md
  - └─ RECEIVE\_DIVERSITY.md
  - └─ RIG\_CONTROL.md
  - └─ THERMAL\_GUARD.md
  - └─ USER\_GUIDE.md
- └─ INSTALLATION.md
- └─ MULTI\_RECEIVER.md
- └─ PhantomSDR-Plus-Documentation-EN.pdf
- └─ PROJECT\_STRUCTURE.md

```

├── RADE_General_INSTALL_MANUAL_LINUX.md
├── RADE_README.md
├── README.md
├── RECEIVE_DIVERSITY.md
├── RIG_CONTROL.md
├── ru
│ ├── ADMIN_PANEL_SETUP.md
│ ├── Aether_config.md
│ ├── CONNECTION_LIMITS.md
│ ├── DECODERS.md
│ ├── EDITING_VARIANTS.md
│ ├── INSTALLATION.md
│ ├── MULTI_RECEIVER.md
│ ├── PhantomSDR-Plus-Documentation-RU.pdf
│ ├── PROJECT_STRUCTURE.md
│ ├── RADE_General_INSTALL_MANUAL_LINUX.md
│ ├── RADE_README.md
│ ├── README.md
│ ├── RECEIVE_DIVERSITY.md
│ ├── RIG_CONTROL.md
│ ├── THERMAL_GUARD.md
│ └── USER_GUIDE.md
├── sdr-stats
│ ├── package.json
│ ├── readme_de.md
│ ├── readme_el.md
│ ├── readme_es.md
│ ├── readme_fr.md
│ ├── readme_hr.md
│ ├── README.md
│ └── readme_ru.md
├── THERMAL_GUARD.md
├── USER_GUIDE.md
├── websdr2.png
├── websdr3.png
├── websdr.png
├── favicon.ico
├── fftw_wisdom
├── fix_local_geo.py
├── frequencylist
│ ├── 0.TXT
│ ├── A26all00.TXT
│ └── a26allx2.zip
│ # δεν είναι στο repo: το αποσυμπιέζει το update-markers.sh από το
├── a26allx2.zip
│ ├── a26allx2.zip
│ ├── admin.txt
│ ├── antenna.txt
│ ├── broadcas.txt
│ ├── curl-output.txt
│ ├── currentUpdateFile.txt
│ ├── fmorg.txt
│ ├── generate-current-shortwave.py
│ ├── language.txt
│ ├── mymarkers.json
│ ├── README.md
│ ├── shortwavestations.json
│ ├── site.txt
│ └── update-markers.sh
├── frontend
│ ├── build-all.sh
│ ├── build-default.sh
│ ├── build-mobile.sh
│ ├── debug-title.sh
│ ├── favicon.ico
│ ├── fix-title-python.py
│ └── index.html

```

```

├─ jsconfig.json
├─ LICENSE
├─ make-redirect-stubs.sh
├─ mobile
│ └─ index.html
├─ package.json
├─ package-lock.json
├─ pnpm-lock.yaml
├─ postcss.config.cjs
├─ .prettierrc.json # κανόνες μορφοποίησης Prettier για τον κώδικα του frontend
├─ public
│ ├── analyze_users.py
│ ├── decoders
│ │ ├── ft8_lib.wasm
│ │ ├── js8_dict.bin # λεξικό λέξεων JS8, φορτώνεται μόνο όταν χρειαστεί
│ │ └─ js8.wasm # αποκωδικοποιητής JS8, από το jsdsp/js8_wasm
│ ├── logo.jpg
│ ├── stats.html
│ ├── users.html
│ └─ wf-message.json
├─ README.md
├─ site_information.json
├─ src
│ ├── app.css
│ ├── App.svelte
│ ├── assets
│ │ ├── amateurfrequencies.json
│ │ ├── background.jpg
│ │ ├── shortwavestations.json
│ │ ├── SSTV.png
│ │ ├── SSTV.svg
│ │ └─ svelte.png
│ ├── audio.js
│ ├── audio-stream-worklet.js
│ ├── bands-config.js
│ ├── broadcastSchedules.js
│ ├── clientVersion.js
│ ├── cwDecoder.js
│ ├── cw.worker.js
│ ├── cwWorkerProxy.js
│ ├── decoder.worker.js
│ ├── diversity.js
│ ├── diversityList.js
│ ├── eventBus.js
│ ├── events.js
│ ├── fax.js
│ ├── fax.worker.js
│ ├── faxWorkerProxy.js
│ ├── fft.js
│ ├── flacLowLatency.js # αποκωδικοποιητής FLAC που παίζει κάθε πακέτο μόλις φτάσει
│ ├── fsk.js
│ ├── fsk.worker.js
│ ├── fskWorkerProxy.js
│ ├── kiwiSource.js
│ ├── lib
│ │ ├── backend.js
│ │ ├── BandSelector.svelte
│ │ ├── catsync.js
│ │ ├── chatReply.js # σήμανση και νήματα απαντήσεων στη συνομιλία
│ │ ├── CheckButton.svelte
│ │ ├── colormaps.js
│ │ ├── Counter.svelte
│ │ ├── DiversityPanel.svelte
│ │ ├── fftRadix2.js
│ │ └─ freedv-reporter.js

```

```

├─ FreeDVReporter.svelte
├─ FrequencyInput.svelte
├─ FrequencyMarkers.svelte
├─ FtxSpectrum.svelte
├─ hammeractions.js
├─ LineThroughButton.svelte
├─ Logger.svelte
├─ MagicEyeIndicator.svelte
├─ ModeIdChip.svelte
├─ ModesSelector.svelte
├─ opusMlDecoder.js
├─ PassbandTuner.svelte
├─ Popover.svelte
├─ QrssPanel.svelte
├─ receivers.js # δεδομένα της επιλογής δέκτη (/receivers.json)
├─ rnnnoise.js # φορτώνει το RNNnoise για τη μείωση θορύβου AI
├─ rx.js # σε ποιον δέκτη ανήκει μια σελίδα (?rx=)
├─ SMeterAnalog.svelte
├─ SMeterDigital.svelte
├─ Spectrogram.svelte
├─ StatusIndicators.svelte
├─ storage.js
├─ sUnits.js # κλίμακα μονάδων S ανά συχνότητα (S9 = -93 dBm πάνω από 30
MHz)
├─ Tooltip.svelte
├─ VersionSelector.svelte
├─ VideoAreaSelector.svelte
├─ wrappers.js
├─ main.js
├─ mobile
├─ └─ backend.js
├─ └─ bookmarks.js
├─ └─ main.js
├─ └─ Mobile.svelte
├─ └─ tuning.js
├─ modeId.js
├─ modeId.worker.js
├─ modeIdWorkerProxy.js
├─ modePriors.js
├─ modules
├─ └─ decode.wasm
├─ └─ encode.wasm
├─ └─ ft4.js
├─ └─ ft8.js
├─ └─ js8-decoder.js
├─ └─ js8-format.js
├─ └─ js8.js
├─ └─ js8-reassembler.js
├─ └─ js8-slots.js
├─ └─ js8-tables.js
├─ └─ package.json
├─ └─ phantomsdrdsp_bg_fallback.js
├─ └─ phantomsdrdsp_bg.js
├─ └─ phantomsdrdsp_bg.wasm
├─ └─ phantomsdrdsp_bg.wasm.d.ts
├─ └─ phantomsdrdsp.d.ts
├─ └─ phantomsdrdsp.js
├─ └─ phantomsdrdsp_router.js
├─ └─ wspr.js
├─ olivia.js
├─ playoutControl.js # μετρά τη σύνδεση, κρατά μικρό το buffer αναπαραγωγής
├─ psk31.js
├─ refused.js
├─ remoteSource.js
├─ scanner.js

```

```

| | | | | siteInfo.js # site_information.json, με επικάλυψη από το αντίγραφο του
δέκτη
| | | | | sstv.js
| | | | | sstv.worker.js
| | | | | sstvWorkerProxy.js
| | | | | uberSource.js
| | | | | unused
| | | | | | AudioProcessor.js
| | | | | | decoder.js
| | | | | | decoding.js
| | | | | | modules-emsripten
| | | | | | | dav1d.js
| | | | | | | dav1dnoWasm.js
| | | | | | | dav1dnoWasm.js.mem
| | | | | | | dav1d.wasm
| | | | | | | decode_ft8.js
| | | | | | | decode_ft8.wasm
| | | | | | | FoxenFlac.js
| | | | | | | jsDSP.js
| | | | | | | jsDSPnoWasm.js
| | | | | | | jsDSPnoWasm.js.mem
| | | | | | | jsDSPnoWasm.wasm
| | | | | | | jsDSP.wasm
| | | | | | | libzstd.js
| | | | | | | LiquidDSP.js
| | | | | | | opus.js
| | | | | | | opusnoWasm.js
| | | | | | | opusnoWasm.js.mem
| | | | | | | opus.wasm
| | | | | | | redsea.js
| | | | | | | redsea.wasm
| | | | | | unused.js
| | | | | | wrappers.js
| | | | | | videoRecorder.js
| | | | | | vite-env.d.ts
| | | | | | waterfall.js
| | | | | | webSdrCodec.js
| | | | | | webSdrSource.js
| | | | | stats.html
| | | | | svelte.config.js
| | | | | tailwind.config.cjs
| | | | | vite.config.js
| | | | go.sh # παλιά αλυσίδα εκκίνησης, αντικαταστάθηκε από τα start-<radio>.sh
| | | | install_arch.sh
| | | | install_fedora.sh
| | | | install_opensuse.sh
| | | | install_rade.sh
| | | | install_rade_ubuntu22.sh
| | | | install.sh
| | | | install-stats-server.sh
| | | | instances # επιπλέον δέκτες, ένας φάκελος ο καθένας (όχι στο git) – δείτε
docs/MULTI_RECEIVER.md
| | | | instructions-for-airspy
| | | | instructions-for-rspla
| | | | jsdsp
| | | | | compilejs.sh
| | | | | configureredsea.sh
| | | | | extract_EXPORTED_FUNCTIONS.js
| | | | | ft8_wasm
| | | | | | build_ft8_wasm.sh
| | | | | | README.md
| | | | | | wasm_wrapper.c
| | | | | include
| | | | | | avif
| | | | | | | avif.h

```

```

├── internal.h
├── liquid
│ └── liquid.h
├── js8_wasm # αποκωδικοποιητής JS8 σε C, μεταγλωττισμένος σε WebAssembly, με τα
τεστ και τον έλεγχο αναφοράς του – δείτε το README του
├── build_js8_wasm.sh
├── check_against_reference.sh
├── extract_js8_constants.py
├── extract_js8_varicode.py
├── js8_ab_test.mjs
├── js8_autorun_test.mjs
├── js8_chain_test.mjs
├── js8_constants.c
├── js8_constants.h
├── js8_decode.c
├── js8_decode.h
├── js8_decode_test.c
├── js8_encode.c
├── js8_encode.h
├── js8_frames_test.mjs
├── js8_osd.c
├── js8_osd.h
├── js8_reassembly_test.mjs
├── js8_roundtrip.c
├── js8_slots_test.mjs
├── js8_subtract.c
├── js8_subtract.h
├── js8_wasm_test.mjs
├── js8_wasm_wrapper.c
├── README.md
├── reference
│ ├── build_in_container.sh
│ ├── build_reference_decoder.sh
│ ├── build_reference_frames.sh
│ ├── Containerfile
│ ├── Containerfile.decoder
│ ├── ref_decode.f90
│ ├── ref_decode.txt
│ ├── ref_frames.cpp
│ └── ref_frames.txt
├── ref_gen.f90
├── run_tests.sh
├── lib
│ ├── ANR.c
│ ├── arm_funcs.h
│ ├── CMSIS_DSP
│ │ ├── BUILDING.txt
│ │ └── LICENSE.txt
│ ├── dav1d.cpp
│ ├── NB.c
│ ├── NR_spectral.c
│ └── types.h
├── redsea.js
├── redsea.wasm
├── src
│ ├── index.js
│ ├── libzstd.js
│ ├── LiquidDSP.js
│ ├── NoiseProcessing.js
│ └── wbfmpll.cpp
├── kill.sh # παλιό: τερματίζει τις διεργασίες, καλείται από το go.sh
├── kiwi_install.sh # εγκαθιστά τη γέφυρα KiwiSDR σε υπάρχον δέντρο
├── LICENSE
├── logproxy # εναλλασσόμενα αντίγραφα των logs panel/proxy/autorun
├── logrotate

```

```

├── phantomsdr # ρύθμιση logrotate για proxy.log + admin.log (διορθώστε τις
διαδρομές πριν την εγκατάσταση)
├── manage_admin.sh
├── markers.json
├── meson.build
├── meson_options.txt
├── phantom_fftw_wisdom
├── phantomsdr-admin.service # δείγμα μονάδας systemd για τον πίνακα διαχείρισης (εκκίνηση με το
boot, επανεκκίνηση μετά από κατάρρευση)
├── phantomsdr-proxy.service # δείγμα μονάδας systemd για τον διαμεσολαβητή, εγκαθίσταται μαζί με
τη μονάδα του πίνακα
├── phantomsdr-websdr-relay.service # δείγμα systemd unit για το relay diversity WebSDR
├── proxy.py
├── rade_helper.py
├── rade_loadtest.csv # έξοδος του rade_loadtest.py (μία γραμμή ανά βήμα φόρτου)
├── rade_loadtest.py # τεστ φόρτου RADE: πόσοι ταυτόχρονοι αποκωδικοποιητές αντέχει το
μηχάνημα – δείτε docs/RADE_README.md
├── rade.sh
├── README.md
├── recompile.sh
├── receivers.toml.example # λίστα δεκτών για το proxy.py – αντιγράψτε σε receivers.toml
├── _relaunch.sh # παλιό: βοηθός καθυστερημένης επανεκκίνησης της αλυσίδας go.sh
├── request.hpp
├── setup_admin.sh
├── setup-airspyhf.sh # αλυσίδα driver Airspy HF+: libairspyhf + SoapyAirspyHF + rx_sdr +
udev
├── setup-cpufreq-perms.sh # δίνει στην ομάδα δικαίωμα εγγραφής στο scaling_max_freq, ώστε η
προστασία να περιορίζει τη συχνότητα χωρίς root
├── setup-firewall.sh # προαιρετική προστασία nftables – δείτε docs/CONNECTION_LIMITS.md
├── setup-fobos.sh # αλυσίδα driver Fobos: libfobos + SoapyFobosSDR + rx_sdr +
cf32 to real + udev
├── setup-hackrf.sh # HackRF: πακέτο hackrf της διανομής + udev (τίποτα δεν χτίζεται)
├── setup-rspla.sh # αλυσίδα driver RSP1A: libmirisdr-5 + SoapyMiri + rx_sdr +
blacklist msi2500 + udev
├── setup-rtlsdr.sh # οδηγός RTL-SDR: πακέτο της διανομής, ή οδηγός Blog V4 από τον
πηγαίο κώδικα + αποκλεισμός DVB + udev
├── setup-rx888-udev.sh
├── setup-sdr-common.sh # κοινές βοηθητικές συναρτήσεις των τριών setup scripts παραπάνω
├── setup_websdr_relay.sh # εγκαθιστά το relay diversity για WebSDR (θύρα, ταυτότητα, systemd)
├── smeter_theme.sh
├── src
│ ├── audio.cpp
│ ├── audio.h
│ ├── chat.cpp
│ ├── chat.h
│ ├── client.cpp
│ ├── client.h
│ ├── compression.cpp
│ ├── compression.h
│ ├── crash_handler.cpp
│ ├── crash_handler.h
│ ├── events.cpp
│ ├── events.h
│ ├── fft.cpp
│ ├── fft_cuda.cu
│ ├── fft.h
│ ├── fft_impl.cpp
│ ├── fft_mkl.cpp
│ ├── http.cpp
│ ├── kiwi_bridge.h # γέφυρα πρωτοκόλλου KiwiSDR – βλ. docs/Aether_config.md
│ ├── listing
│ │ ├── software_info.cpp
│ │ └── software_info.h
│ ├── samplerreader.cpp
│ └── samplerreader.h

```

```

├─ signal.cpp
├─ signal.h
├─ spectrumserver.cpp
├─ spectrumserver.h
├─ utils
│ ├── audioprocessing.cpp
│ ├── audioprocessing.h
│ ├── dsp.cpp
│ └── dsp.h
├─ utils.cpp
├─ utils.h
├─ waterfallcompression.cpp
├─ waterfallcompression.h
├─ waterfall.cpp
├─ waterfall.h
├─ websocket.cpp
└─ websocket.h
├─ start-airspyhf.sh
├─ start-all.sh # ξεκινά κάθε δέκτη του receivers.toml
├─ start-fobos-hf.sh
├─ start-fobos.sh
├─ start-hackrf.sh
├─ start-rsp1a.sh
├─ start-rtl.sh
├─ start-rx888mk2.sh
├─ stop-websdr.sh
├─ subprojects
│ ├── fftw3.wrap
│ ├── flac.wrap
│ ├── glaze.wrap
│ ├── libcds.wrap
│ ├── libflac.wrap
│ ├── libvolk.wrap
│ ├── ogg.wrap
│ ├── opus.wrap
│ └─ tomlplusplus-3.4.0
│ ├── CHANGELOG.md
│ ├── cmake
│ │ ├── install-rules.cmake
│ │ ├── project-is-top-level.cmake
│ │ ├── tomlplusplusConfig.cmake
│ │ ├── tomlplusplusConfig.cmake.meson.in
│ │ ├── tomlplusplusConfigVersion.cmake.meson.in
│ │ └─ variables.cmake
│ ├── CMakeLists.txt
│ ├── CODE_OF_CONDUCT.md
│ ├── CONTRIBUTING.md
│ ├── cpp.hint
│ ├── docs
│ │ ├── images
│ │ │ ├── badge-awesome.svg
│ │ │ ├── badge-C++17.svg
│ │ │ ├── badge-gitter.svg
│ │ │ ├── badge-license-MIT.svg
│ │ │ ├── badge-TOML.svg
│ │ │ ├── banner.ai
│ │ │ ├── banner.png
│ │ │ ├── banner.svg
│ │ │ ├── favicon.ico
│ │ │ ├── logo.ai
│ │ │ └─ logo.svg
│ │ ├── pages
│ │ │ └─ main_page.md
│ │ └─ poxy.toml
│ └─ examples

```

```

├─ benchmark_data.toml
├─ CMakeLists.txt
├─ error_printer.cpp
├─ error_printer.vcxproj
├─ examples.hpp
├─ example.toml
├─ merge_base.toml
├─ merge_overrides.toml
├─ meson.build
├─ parse_benchmark.cpp
├─ parse_benchmark.vcxproj
├─ simple_parser.cpp
├─ simple_parser.vcxproj
├─ toml_generator.cpp
├─ toml_generator.vcxproj
├─ toml_merger.cpp
├─ toml_merger.vcxproj
├─ toml_to_json_transcoder.cpp
├─ toml_to_json_transcoder.vcxproj
├─ include
├─ meson.build
├─ toml++
├─ impl
├─ array.hpp
├─ array.inl
├─ at_path.hpp
├─ at_path.inl
├─ date_time.hpp
├─ formatter.hpp
├─ formatter.inl
├─ forward_declarations.hpp
├─ header_end.hpp
├─ header_start.hpp
├─ json_formatter.hpp
├─ json_formatter.inl
├─ key.hpp
├─ make_node.hpp
├─ node.hpp
├─ node.inl
├─ node_view.hpp
├─ parse_error.hpp
├─ parse_result.hpp
├─ parser.hpp
├─ parser.inl
├─ path.hpp
├─ path.inl
├─ preprocessor.hpp
├─ print_to_stream.hpp
├─ print_to_stream.inl
├─ simd.hpp
├─ source_region.hpp
├─ std_except.hpp
├─ std_initializer_list.hpp
├─ std_map.hpp
├─ std_new.hpp
├─ std_optional.hpp
├─ std_string.hpp
├─ std_string.inl
├─ std_utility.hpp
├─ std_variant.hpp
├─ std_vector.hpp
├─ table.hpp
├─ table.inl
├─ toml_formatter.hpp
├─ toml_formatter.inl

```

```

├───┬─── unicode_autogenerated.hpp
│ ├─── unicode.hpp
│ ├─── unicode.inl
│ ├─── value.hpp
│ ├─── version.hpp
│ ├─── yaml_formatter.hpp
│ └─── yaml_formatter.inl
├─── toml.h
├─── toml.hpp
├── LICENSE
├── meson.build
├── meson_options.txt
├── README.md
├── src
│ ├─── meson.build
│ └─── toml.cpp
├── tests
│ ├─── at_path.cpp
│ ├─── conformance_burntsushi_invalid.cpp
│ ├─── conformance_burntsushi_valid.cpp
│ ├─── conformance_iarna_invalid.cpp
│ ├─── conformance_iarna_valid.cpp
│ ├─── cpp.hint
│ ├─── for_each.cpp
│ ├─── formatters.cpp
│ ├─── impl_toml.cpp
│ ├─── leakproof.hpp
│ ├─── lib_catch2.hpp
│ ├─── main.cpp
│ ├─── manipulating_arrays.cpp
│ ├─── manipulating_parse_result.cpp
│ ├─── manipulating_tables.cpp
│ ├─── manipulating_values.cpp
│ ├─── meson.build
│ ├─── odr_test_1.cpp
│ ├─── odr_test_2.cpp
│ ├─── parsing_arrays.cpp
│ ├─── parsing_booleans.cpp
│ ├─── parsing_comments.cpp
│ ├─── parsing_dates_and_times.cpp
│ ├─── parsing_floats.cpp
│ ├─── parsing_integers.cpp
│ ├─── parsing_key_value_pairs.cpp
│ ├─── parsing_spec_example.cpp
│ ├─── parsing_strings.cpp
│ ├─── parsing_tables.cpp
│ ├─── path.cpp
│ ├─── settings.hpp
│ ├─── tests.cpp
│ ├─── tests.hpp
│ ├─── user_feedback.cpp
│ ├─── using_iterators.cpp
│ ├─── visit.cpp
│ └─── vs
│ ├─── odr_test.vcxproj
│ ├─── test_debug_x64_cpplatest_noexcept_unrel.vcxproj
│ ├─── test_debug_x64_cpplatest_noexcept.vcxproj
│ ├─── test_debug_x64_cpplatest_unrel.vcxproj
│ ├─── test_debug_x64_cpplatest.vcxproj
│ ├─── test_debug_x64_noexcept_unrel.vcxproj
│ ├─── test_debug_x64_noexcept.vcxproj
│ ├─── test_debug_x64_unrel.vcxproj
│ ├─── test_debug_x64.vcxproj
│ ├─── test_debug_x86_cpplatest_noexcept_unrel.vcxproj
│ └─── test_debug_x86_cpplatest_noexcept.vcxproj

```

```

├── test_debug_x86_cpplatest_unrel.vcxproj
├── test_debug_x86_cpplatest.vcxproj
├── test_debug_x86_noexcept_unrel.vcxproj
├── test_debug_x86_noexcept.vcxproj
├── test_debug_x86_unrel.vcxproj
├── test_debug_x86.vcxproj
├── test_release_x64_cpplatest_noexcept_unrel.vcxproj
├── test_release_x64_cpplatest_noexcept.vcxproj
├── test_release_x64_cpplatest_unrel.vcxproj
├── test_release_x64_cpplatest.vcxproj
├── test_release_x64_noexcept_unrel.vcxproj
├── test_release_x64_noexcept.vcxproj
├── test_release_x64_unrel.vcxproj
├── test_release_x64.vcxproj
├── test_release_x86_cpplatest_noexcept_unrel.vcxproj
├── test_release_x86_cpplatest_noexcept.vcxproj
├── test_release_x86_cpplatest_unrel.vcxproj
├── test_release_x86_cpplatest.vcxproj
├── test_release_x86_noexcept_unrel.vcxproj
├── test_release_x86_noexcept.vcxproj
├── test_release_x86_unrel.vcxproj
├── test_release_x86.vcxproj
├── windows_compat.cpp
├── toml++.code-workspace
├── toml.hpp
├── toml++.natvis
├── toml++.props
├── toml++.sln
├── toml-test
│ ├── meson.build
│ ├── README.md
│ ├── tt_decoder.cpp
│ ├── tt_decoder.vcxproj
│ ├── tt_encoder.cpp
│ ├── tt_encoder.vcxproj
│ └── tt.hpp
├── toml++.vcxproj
├── toml++.vcxproj.filters
├── tools
│ ├── ci_single_header_check.py
│ ├── clang_format.bat
│ ├── generate_conformance_tests.py
│ ├── generate_single_header.bat
│ ├── generate_single_header.py
│ ├── generate_windows_test_targets.py
│ ├── requirements.txt
│ ├── utils.py
│ └── version.py
├── vendor
│ ├── catch.hpp
│ ├── json.hpp
│ └── README.md
├── tomlplusplus.wrap
├── websocketpp.wrap
├── zlib.wrap
├── zstd.wrap
├── tci-bridge
│ └── tci-rigctld.mjs # διακομιστής TCI για πομποδέκτες Hamlib (IC-7300...), τρέχει στον
υπολογιστή του ακροατή – βλ. docs/RIG_CONTROL.md
├── thermal_guard.py # Προστασία υπερθέρμανσης CPU για τον πίνακα διαχείρισης (τρέχει και
αυτόνομα)
├── thermal-guard.service # δείγμα μονάδας systemd για την προστασία, χωρίς πίνακα διαχείρισης
├── tmpfiles
│ └── phantomsdr-logs.conf # κρατά τα admin.log + proxy.log στην ιδιοκτησία του χρήστη του
πίνακα (διορθώστε τις διαδρομές πριν την εγκατάσταση)

```

```

└─ update.sh
└─ waterfall.sh
└─ websdr_relay.json.example # πρότυπο ρυθμίσεων (θύρα, όρια, ταυτότητα σταθμού)
└─ websdr_relay.py # το ίδιο το relay – δείτε docs/RECEIVE_DIVERSITY.md
└─ websocketpp_asio_connection.hpp # shim για Boost >= 1.87: ws_post αντί για io_service::post
└─ websocketpp_asio_endpoint.hpp # shim για Boost >= 1.87: ws_work / ws_restart,
max_listen_connections
└─ websocketpp_asio.hpp # shim για Boost >= 1.87 στο websocketpp (io_context,
executor_work_guard)
└─ xgo.sh # παλιό: ξεκινά τον spectrumserver, καλείται από το check-go.sh

```

## Ριζικός κατάλογος

### Αρχεία ρυθμίσεων

Αρχείο	Σκοπός	Πότε το τροποποιείτε
config.toml	Προεπιλεγμένες ρυθμίσεις	Αρχική εγκατάσταση, δοκιμές
config-rtl.toml	Ρυθμίσεις ειδικά για RTL-SDR	Χρήση συσκευής RTL-SDR
config-rspla.toml	Ρυθμίσεις SDRplay RSP1A	Χρήση συσκευής RSP1A
config-airspyhf.toml	Ρυθμίσεις Airspy HF+	Χρήση συσκευής Airspy
config-fobos-hf.toml	RigExpert Fobos SDR, HF1/HF2 με άμεση δειγματοληψία (0-25 MHz)	Χρήση Fobos στα HF
config-fobos.toml	RigExpert Fobos SDR, κλάδος RF (25-6000 MHz)	Χρήση Fobos πάνω από 25 MHz
config-hackrf.toml	Ρυθμίσεις HackRF One	Χρήση HackRF
config-rx888mk2.toml	Ρυθμίσεις RX888 MK2	Χρήση συσκευής RX888
config.example.hackrf.toml	Παράδειγμα HackRF One	Χρήση συσκευής HackRF

### Σενάρια εκκίνησης/διακοπής και συντήρησης

Καθένα από τα παρακάτω `start-*.sh` είναι **αυτόνομος εκκινητής + watchdog + καταγραφέας**: σταματά κάθε στιγμιότυπο που εκτελείται, ανεβάζει τον δέκτη + τον `spectrumserver`, αποσυνδέεται στο παρασκήνιο, επανεκκινεί αυτόματα την αλυσίδα αν τερματιστεί και καταγράφει στο `logwebsdr.txt`. Μοιράζονται ένα κοινό σενάριο διακοπής και ένα μοναδικό κλείδωμα `flock` (εκτελείται μόνο ένας δέκτης κάθε φορά). Επεξεργαστείτε μόνο το τμήμα **RECEIVER CONFIGURATION** στην αρχή καθενός (ορίσματα δέκτη / ρυθμίσεις / όνομα διεργασίας). Τα έξι scripts εκκίνησης από το ίδιο πρότυπο (`start-airspyhf.sh`, `start-rtl.sh`, `start-rspla.sh`, `start-fobos.sh`, `start-fobos-hf.sh`, `start-hackrf.sh`) είναι πανομοιότυπα κάτω από αυτό το τμήμα· το `start-rx888mk2.sh` είναι ξεχωριστό script. Από την v5.0.0 πολλοί δέκτες μπορούν να τρέχουν δίπλα-δίπλα: αν ένα σενάριο εκκίνησης ξεκινήσει με `INSTANCE=<όνομα>`, ο δέκτης παίρνει δικό του φάκελο στο `instances/<όνομα>/` και δικό του log, κλείδωμα και FIFO — δείτε [Πολλοί δέκτες](#).

Σενάριο	Σκοπός
<code>install.sh</code>	Αυτοματοποιημένη εγκατάσταση και μεταγλώττιση
<code>add-receiver.sh</code>	Προσθέτει έναν ακόμη δέκτη σε αυτόν τον υπολογιστή: ρωτά ποιον δέκτη και τι καλύπτει, εγκαθιστά τον οδηγό του, δημιουργεί το <code>instances/&lt;όνομα&gt;/</code> και την καταχώριση στο <code>receivers.toml</code> , και προσφέρεται να τον ξεκινήσει και να επανεκκινήσει τον <code>proxy</code> . Τα σενάρια εγκατάστασης το προσφέρουν στο τέλος — δείτε <a href="#">Πολλοί δέκτες</a>
<code>start-rtl.sh</code>	Εκκίνηση + watchdog του διακομιστή με RTL-SDR ( <code>rtl_sdr</code> )
<code>start-rspla.sh</code>	Εκκίνηση + watchdog του διακομιστή με SDRplay RSP1A ( <code>rx_sdr</code> )· χρησιμοποιεί το <code>libmirisdr-5</code> , ή το API της SDRplay όταν είναι εγκατεστημένο (το <code>RX_DRIVER</code> επιβάλλει έναν)
<code>start-airspyhf.sh</code>	Εκκίνηση + watchdog του διακομιστή με Airspy HF+ ( <code>rx_sdr</code> )
<code>start-fobos-hf.sh</code>	Εκκίνηση + watchdog του διακομιστή με RigExpert Fobos SDR, HF1/HF2 με άμεση δειγματοληψία ( <code>rx_sdr \ cf32_to_real</code> )
<code>start-fobos.sh</code>	Εκκίνηση + watchdog του διακομιστή με RigExpert Fobos SDR, κλάδος RF ( <code>rx_sdr</code> )
<code>start-hackrf.sh</code>	Εκκίνηση + watchdog του διακομιστή με HackRF One ( <code>hackrf_transfer</code> )
<code>start-rx888mk2.sh</code>	Εκκίνηση + watchdog του διακομιστή με RX888 MK2 ( <code>rx888_stream</code> )
<code>start-all.sh</code>	Ξεκινά κάθε δέκτη του <code>receivers.toml</code> μέσω του δικού του σεναρίου εκκίνησης — ορίστε το ως σενάριο εκκίνησης του πίνακα διαχείρισης σε σταθμό με πολλούς δέκτες· δείτε <a href="#">Πολλοί δέκτες</a>
<code>stop-websdr.sh</code>	Διακοπή του διακομιστή και του watchdog του — κοινό για όλους τους δέκτες
<code>stop-websdr.sh &lt;instance&gt;</code>	Σταματά μόνο έναν δέκτη ( <code>main</code> για τον κύριο)· χωρίς όρισμα σταματούν όλοι οι δέκτες, όπως πριν
<code>setup-rx888-udev.sh</code>	Εγκαθιστά κανόνες udev ώστε το <code>rx888_stream</code> του RX-888 να τρέχει χωρίς <code>sudo</code>
<code>setup-rspla.sh /</code> <code>setup-fobos.sh /</code> <code>setup-airspyhf.sh</code>	Εγκαθιστούν την αλυσίδα driver SoapySDR για RSP1A, Fobos ή Airspy HF+ — SoapySDR από τη διανομή, driver και <code>rx_tools</code> χτισμένα στο <code>sdr_drivers/</code> , κανόνας udev — σε οποιαδήποτε από τις τέσσερις υποστηριζόμενες διανομές. Τα καλούν οι επιλογές 3, 5 και 6 του προγράμματος εγκατάστασης· τρέχουν και μόνα τους. Κοινός κώδικας στο <code>setup-sdr-common.sh</code> — δείτε τον <a href="#">Οδηγό εγκατάστασης</a>
<code>setup-hackrf.sh</code>	Εγκαθιστά το πακέτο <code>hackrf</code> της διανομής και κανόνα udev για τον HackRF One, σε οποιαδήποτε από τις τέσσερις υποστηριζόμενες διανομές· το καλεί η επιλογή 7 του προγράμματος εγκατάστασης
<code>setup-rtlsdr.sh</code>	Εγκαθιστά τον οδηγό RTL-SDR σε οποιαδήποτε από τις τέσσερις υποστηριζόμενες διανομές — το πακέτο <code>rtl-sdr</code> της διανομής, ή με <code>RTL_V4=y</code> τον οδηγό του RTL-SDR Blog V4 χτισμένο από τον πηγαίο κώδικα — αποκλείει τον οδηγό DVB-T και προσθέτει κανόνα udev· το καλεί το <code>add-receiver.sh</code>
<code>setup-firewall.sh</code>	Προαιρετική προστασία από πλημμύρα σε επίπεδο πυρήνα: φορτώνει έναν πίνακα <code>nftables</code> με ταβάνι ταυτόχρονων συνδέσεων και ρυθμό ανά διεύθυνση προέλευσης στις θύρες του δέκτη, φρένο για επιθέσεις στο SSH, και το Windows file sharing κλειστό εκτός ιδιωτικών περιοχών. Χρειάζεται <code>root</code> , δεν μπορεί να σας κλειδώσει έξω ( <code>policy accept</code> , οι εδραιωμένες συνδέσεις γίνονται δεκτές πρώτες) και το <code>--apply</code> αναιρείται μόνο του αν δεν επιβεβαιωθεί μέσα σε 60 δευτ. — δείτε <a href="#">Όρια συνδέσεων</a>
<code>setup-cpufreq-perms.sh</code>	Δίνει σε μια ομάδα <code>cpufreq</code> δικαίωμα εγγραφής στο όριο συχνότητας της CPU, ώστε το στάδιο <code>throttle</code> της θερμικής προστασίας να δουλεύει χωρίς <code>root</code> . Εγκαθιστά κανόνα <code>tmpfiles.d</code> ώστε να επιβιώνει επανεκκίνησης· το <code>--revoke</code> το αναιρεί
<code>update.sh</code>	

Σενάριο	Σκοπός
	Ενημέρωση της εγκατάστασης από το δημοσιευμένο δέντρο, χωρίς να θιγούν η διαμόρφωση, οι σημάνσεις, η λίστα συχνοτήτων και οι δικές σας αλλαγές — δείτε τον <a href="#">Οδηγό εγκατάστασης</a>
<code>recompile.sh</code>	Ανακατασκευή backend ή/και frontend, και επιλογή της παραλλαγής που σερβίρεται στο /
<code>smeter_theme.sh</code>	Ορισμός της προεπιλεγμένης όψης του αναλογικού S-meter (dark / amber / vintage) για όλους τους χρήστες, με προσφορά ανακατασκευής του frontend — δείτε <a href="#">Επεξεργασία παραλλαγών</a>
<code>waterfall.sh</code>	Αλλάζει το προεπιλεγμένο ελάχιστο επίπεδο καταρράκτη (dB) σε <code>waterfall.js</code> + <code>App.svelte</code> — δείτε το <a href="#">README</a>
<code>kiwi_install.sh</code>	Εγκαθιστά την εξομίωση πελατών KiwiSDR σε δέντρο που δεν την έχει: διορθώνει τις πηγές του backend, αντιγράφει το <code>src/kiwi_bridge.h</code> και προσθέτει τεκμηριωμένο μπλοκ <code>[kiwi_emulation]</code> στα αρχεία ρυθμίσεων της ρίζας. Ταυτοδύναμο, και κρατά αντίγραφο κάθε αρχείου που αγγίζει — βλ. <a href="#">Εξομίωση πελατών KiwiSDR</a>
<code>tci-bridge/tci-rigctld.mjs</code>	Δεν χρησιμοποιείται από τον δέκτη. Ένα μικρό πρόγραμμα Node.js που τρέχει ο ακροατής δίπλα στον δικό του πομποδέκτη: διαβάζει συχνότητα, τρόπο λειτουργίας και κατάσταση εκπομπής από το <code>rigctld</code> του Hamlib και τα προσφέρει ως TCI στη θύρα 50001, ώστε το κουμπί <b>TCI-CAT</b> της σελίδας να ελέγχει πομποδέκτη χωρίς δικό του TCI, όπως το IC-7300 — βλ. <a href="#">Έλεγχος πομποδέκτη</a>

**Παλιά αλυσίδα εκκίνησης.** Τα `go.sh`, `xgo.sh`, `check-go.sh`, `kill.sh` και `_relaunch.sh` είναι η προηγούμενη γενιά scripts εκκίνησης, επιτήρησης και τερματισμού. Ό,τι έκαναν βρίσκεται πλέον μέσα σε κάθε `start-<radio>.sh`, και αυτό είναι που πρέπει να χρησιμοποιείτε. Παραμένουν στον δίσκο επειδή υπάρχουσες εγκαταστάσεις τα αναφέρουν, και δεν συντηρούνται.

## Αρχεία δεδομένων

Αρχείο	Σκοπός	Μορφή
<code>markers.json</code>	Σελιδοδείκτες και δείκτες συχνοτήτων	JSON
<code>chat_history.txt</code>	Μηνύματα συνομιλίας χρηστών	Απλό κείμενο
<code>favicon.ico</code>	Εικονίδιο ιστότοπου	Εικόνα ICO
<code>fftw_wisdom</code>	Δεδομένα βελτιστοποίησης FFT	Δυαδικό FFTW
<code>phantom_fftw_wisdom</code>	Πρόσθετη βελτιστοποίηση FFT	Δυαδικό FFTW

## Πίνακας διαχείρισης και αναφορά spot

Αρχείο	Σκοπός
<code>admin_server.py</code>	Ο ίδιος ο πίνακας διαχείρισης. Πέρα από τις σελίδες διαχείρισης εκτελεί τον δειγματολήπτη της σελίδας <b>Γραφήματα</b> : ένα νήμα παρασκηνίου καταγράφει κάθε 2 δευτερόλεπτα συχνότητα CPU, φόρτο, θερμοκρασία και συνδεδεμένους χρήστες και τα κρατά μόνο στη μνήμη — 1 ώρα σε πλήρη ανάλυση συν 24 ώρες σε μέσους όρους 30 δευτερολέπτων. Τίποτα δεν γράφεται στον δίσκο, οπότε το ιστορικό χάνεται με την επανεκκίνηση.
<code>admin_config.json</code>	Ρυθμίσεις του πίνακα διαχείρισης (hash κωδικού, θύρες, επιλογές, όρια θερμικής προστασίας)
<code>thermal_guard.py</code>	Προστασία υπερθέρμανσης της CPU. Σταματά τον διακομιστή όταν η CPU ζεσταθεί υπερβολικά και τον επανεκκινεί μόλις κρυώσει, με όρια που προκύπτουν από το κρίσιμο όριο που δημοσιεύει η ίδια σας η CPU αντί για έναν σταθερό αριθμό. Δεν προσπαθεί ποτέ να αναγνωρίσει τι επιβλέπει τον διακομιστή — όσο η CPU είναι υπερθερμασμένη επαναλαμβάνει την εντολή διακοπής κάθε 2 δευτερόλεπτα, ώστε ένας watchdog, μια μονάδα systemd ή μια εργασία cron που τον επαναφέρει να αναιρείται μέχρι να κρυώσει το μηχάνημα. Χρησιμοποιεί μόνο τη standard library· εισάγεται από το <code>admin_server.py</code> (με χρονισμό από τον δειγματολήπτη των Γραφημάτων) και τρέχει και αυτόνομα σε εγκαταστάσεις χωρίς τον πίνακα. Από προεπιλογή μόνο καταγράφει, οπότε δεν ενεργεί σε τίποτα μέχρι να ενεργοποιηθεί — δείτε [Πίνακας διαχείρισης] δείτε το <a href="#">εγχειρίδιο Thermal Guard</a>
<code>autorun/</code>	Ο δαίμονας αναφοράς spot — δείτε <a href="#">Autorun Spot Reporter</a>
<code>autorun.json</code>	Μπάντες/τρόποι προς αποκωδικοποίηση, στοιχεία ταυτότητας και προορισμοί
<code>autorun-status.json</code>	Τρέχουσα κατάσταση του δαίμονα — τροφοδοτεί τους μετρητές <b>της τρέχουσας εκτέλεσης</b> (τα πλακίδια ανά αποκωδικοποιητή), που μηδενίζονται με Stop/Start
<code>autorun-totals.json</code>	<b>Συνολικά</b> ανεβασμένα spot ανά μπάντα και τρόπο — ο αριθμός δίπλα σε κάθε πλαίσιο ελέγχου· γράφεται από τον δαίμονα ώστε να επιβιώνει των επανεκκινήσεων

## Αρχεία του συστήματος μεταγλώττισης

Αρχείο	Σκοπός
<code>meson.build</code>	Κύρια ρύθμιση μεταγλώττισης
<code>meson_options.txt</code>	Παραμετροποιήσιμες επιλογές μεταγλώττισης
<code>.gitattributes</code>	Ιδιότητες του αποθετηρίου Git

## Πηγαίος κώδικας (`src/`)

Ο κατάλογος `src/` περιέχει την υλοποίηση του backend σε C++.

### Βασικά στοιχεία

#### 1. Κύρια εφαρμογή (`main.cpp`)

- Αναλύει τα ορίσματα της γραμμής εντολών
- Φορτώνει το αρχείο ρυθμίσεων
- Αρχικοποιεί τα στοιχεία του διακομιστή
- Ξεκινά τον βρόχο συμβάντων

#### 2. Διακομιστής φάσματος (`spectrumserver.cpp`)

- Συντονίζει όλα τα στοιχεία

- Διαχειρίζεται τις συνδέσεις χρηστών
- Διανέμει τα δεδομένα φάσματος
- Χειρίζεται τα αιτήματα χρηστών

### 3. Οδηγοί SDR (drivers/)

- Αφηρημένη διεπαφή για το υλικό SDR
- Ανάγνωση και μορφοποίηση δεδομένων δειγμάτων
- Χειρισμός λειτουργιών ειδικών ανά συσκευή

### 4. Μηχανή DSP (dsp/)

- Υπολογισμός FFT (με επιτάχυνση CPU/GPU)
- Αποδιαμόρφωση (AM, FM, SSB, CW κ.λπ.)
- Φιλτράρισμα και επαναδειγματοληψία ήχου
- AGC και μείωση θορύβου

### 5. Διακομιστής web (server/)

- Επικοινωνία WebSocket
- Εξυπηρέτηση στατικών αρχείων HTTP
- Διαχείριση συνεδριών χρηστών
- Ροή δεδομένων σε πραγματικό χρόνο

### 6. Κωδικοποίηση ήχου (audio/)

- Συμπίεση FLAC
- Συμπίεση Opus
- Βελτιστοποίηση ροής

## Frontend (frontend/)

Η διεπαφή χρήστη βασισμένη στο web, κατασκευασμένη με Svelte και Vite.

### 1. Κύρια εφαρμογή (App.svelte)

- Στοιχείο ανώτατου επιπέδου
- Δομή διάταξης
- Ενορχήστρωση στοιχείων
- **Σειρά κουμπιών αποκωδικοποιητών** — ένα κουμπί ανά αποκωδικοποιητή στον κύριο πίνακα, ακριβώς κάτω από τον επιλογέα τρόπων· ένα πάτημα ξεκινά τον αποκωδικοποιητή και ανοίγει το παράθυρό του, δεύτερο πάτημα τον σταματά. Αντικατέστησε την παλιά σειρά Bandwidth. Τα RADEL/RADEU βρίσκονται στο `lib/ModesSelector.svelte`, δίπλα στον επιλογέα τρόπων και μέσα στα αναδυόμενα παράθυρα **Modes** και **Bands**.

### 2. Οθόνη καταρράκτη (waterfall.js + lib/)

- Σχεδίαση φάσματος και καταρράκτη σε canvas, χρωματικοί χάρτες και η προσαρμοστική αυτόματη ρύθμιση, όλα στο `waterfall.js` (καθαρή JS, όχι στοιχείο)
- Διαδραστικός συντονισμός μέσω του `lib/PassbandTuner.svelte`
- Επικαλύψεις σχεδίου μπαντών και δεικτών μέσω του `lib/FrequencyMarkers.svelte`
- Το φασματογράφημα ήχου είναι ξεχωριστό στοιχείο: `lib/Spectrogram.svelte`

### 3. Χειριστήρια (App.svelte + lib/)

- Εισαγωγή/ένδειξη συχνότητας — `lib/FrequencyInput.svelte`
- Επιλογή τρόπου (AM/FM/SSB/CW) — `lib/ModesSelector.svelte`, αλλαγή μπάντας — `lib/BandSelector.svelte`
- Τα AGC/NR/NB και τα υπόλοιπα χειριστήρια βρίσκονται στο ίδιο το `App.svelte`· δεν υπάρχει ξεχωριστό `Controls.svelte`

#### 3a. Σαρωτής (scanner.js)

- Σαρωτής καναλιών: μετακινεί τον δέκτη σε μια περιοχή και σταματά στο πρώτο κανάλι με σήμα. Καθαρή JS, όχι component — το `App.svelte` δίνει τον VFO, τον τρόπο λειτουργίας, το σχέδιο μπαντών και την κλήση συντονισμού, και παίρνει πίσω την κατάσταση του UI μέσω μίας συνάρτησης επιστροφής
- Το κατώφλι είναι dB πάνω από τον θόρυβο βάθους της μπάντας αντί για απόλυτη στάθμη, χρησιμοποιώντας τον θόρυβο που ήδη παρακολουθεί το `waterfall.js` (`snrNoiseDb`), οπότε δεν χρειάζεται δική του βαθμονόμηση
- Δύο τρόποι διάσχισης: συντονισμός και ακρόαση κάθε καναλιού, ή πρώτα έλεγχος του φάσματος και συντονισμός μόνο σε όσα δεν αποκλείονται. Το φάσμα μπορεί να παραλείψει κανάλι, ποτέ όμως να σταματήσει σε ένα — κάθε στάση προκύπτει από πραγματική παραμονή
- Η περιοχή είναι είτε η μπάντα του σχεδίου μπαντών είτε ακριβώς ό,τι δείχνει ο καταρράκτης· η αυτόματη συνέχιση, ο μέγιστος χρόνος παραμονής και η λίστα αποκλεισμού βρίσκονται επίσης εδώ και αποθηκεύονται στο `localStorage`

### 4. Σύστημα ήχου (audio.js)

- Ροή ήχου μέσω WebSocket
- Αποκωδικοποίηση FLAC/Opus
- Έλεγχος αναπαραγωγής ήχου
- Διανέμει ακατέργαστο PCM (που λαμβάνεται πριν από AGC, μείωση θορύβου και σίγαση) στους αποκωδικοποιητές τρόπων
- Μείωση θορύβου AI: το `lib/rnnoise.js` φορτώνει τη μονάδα WebAssembly του RNNoise (`@jitsi/rnnoise-wasm`) με την πρώτη χρήση· το `audio.js` την εφαρμόζει μόνο σε φωνητικούς τρόπους, μετά το σημείο λήψης των αποκωδικοποιητών, ώστε αυτοί να μην την «ακούν» ποτέ
- Το FLAC περνά από το `flacLowLatency.js`, που αποκωδικοποιεί κάθε πακέτο τη στιγμή που φτάνει και φέρνει τον πραγματικό ρυθμό της ροής (για παράδειγμα 12016,3 Hz) στα 12000 Hz με έναν μετασχηματιστή ρυθμού 64 συντελεστών (καθυστερήση 2,7 ms). Ο αποκωδικοποιητής WASM στο `modules/` απελευθερώνει τον ήχο σε σταθερά κομμάτια των 1024 δειγμάτων και κρατούσε πίσω περίπου 160 ms· χρησιμοποιείται ακόμη για το Opus, για τις πηγές διαφορεικής λήψης (`diversity`) και για το FLAC όταν η σελίδα ανοίγει με `?flac=wasm`
- Έλεγχος του buffer αναπαραγωγής (`playoutControl.js`): το `ArrivalMeter` μετρά πόσο ακανόνιστα φτάνουν τα πακέτα σε σχέση με τον χρονικό άξονα της ίδιας της ροής, καθώς και την απόκλιση ανάμεσα στο ρολόι δειγματοληψίας του δέκτη και την κάρτα ήχου του ακροατή· το `CushionController` κρατά το μικρότερο απόθεμα του τελευταίου λεπτού κοντά στο περιθώριο της επιλεγμένης βαθμίδας buffer, κόβοντας ή επαναλαμβάνοντας μία περίοδο ταιριασμένη στο τονικό ύψος (`spliceOut` / `spliceIn`) από τον ήχο που πάει στο μεγάφωνο. Οι αποκωδικοποιητές και οι ηχογραφήσεις παίρνουν την ανέγγιχτη ροή. Ισχύει για τη διαδρομή με προγραμματισμένα buffer που χρησιμοποιεί κάθε ακροατής ενός σταθμού σκέτου http· το `?drain=0` τον απενεργοποιεί, το `?diag=1` δείχνει τις τιμές του

#### 4α. Αποκωδικοποιητές τρόπων και οι workers τους

Κάθε ένας από τους βαρείς αποκωδικοποιητές τρόπων εκτελείται σε δικό του Web Worker, ώστε η αποκωδικοποίηση να μη μπλοκάρει ποτέ την αναπαραγωγή ήχου ή τον καταρράκτη. Ακολουθούν κοινό μοτίβο — τρία αρχεία ανά αποκωδικοποιητή:

Αποκωδικοποιητής	Μηχανή	Worker	Proxy στο κύριο νήμα
SSTV	sstv.js	sstv.worker.js	sstvWorkerProxy.js
HF FAX	fax.js	fax.worker.js	faxWorkerProxy.js
NAVTEX + FSK/RTTY + PSK31 + Olivia	fsk.js, psk31.js, olivia.js	fsk.worker.js	fskWorkerProxy.js
CW	cwDecoder.js	cw.worker.js	cwWorkerProxy.js

- Η **μηχανή** είναι καθαρός κώδικας DSP που αγνοεί την ύπαρξη workers, οπότε μπορεί να εκτελεστεί και απευθείας (δοκιμές μονάδας ή εφεδρική εκτέλεση στο ίδιο νήμα).
- Ο **worker** κρατά ένα στιγμιότυπο της μηχανής και προωθεί αυτούσια τα συμβάντά της.
- Το **proxy** αντικατοπτρίζει τις μεθόδους της μηχανής, ώστε το `audio.js` να το καλεί ακριβώς όπως θα καλούσε τον αποκωδικοποιητή. Δημιουργεί τον worker με καθυστέρηση, στην πρώτη ενεργοποίηση, και επιστρέφει σε εκτέλεση στο ίδιο νήμα αν οι workers δεν είναι διαθέσιμοι.

Δύο λεπτομέρειες είναι κρίσιμες: το PCM **αντιγράφεται** σε νέο ενταμιευτή πριν μεταφερθεί στον worker (η μεταφορά μιας όψης του συσσωρευτή ήχου θα τον αποσπούσε και θα σκότωνε την αναπαραγωγή), και το μήνυμα `init` του worker επανεφαρμόζει τις ρυθμίσεις σε μια ήδη εκτελούμενη μηχανή αντί να υποθέτει καινούργια.

Το `fsk.js` εξυπηρετεί τόσο το NAVTEX όσο και το FSK/RTTY από μία μηχανή, με επιλογή ανά στιγμιότυπο μέσω ενός πεδίου `role`· κάθε στιγμιότυπο έχει δική του κατάσταση, οπότε τα δύο μπορούν να τρέχουν ανεξάρτητα.

Ο ρόλος `fsk` φιλοξενεί επιπλέον δύο αποκωδικοποιητές που δεν είναι καθόλου FSK. Επιλέγοντας την παραλλαγή `psk31` ή `olivia`, το `fsk.js` παραδίδει τον ήχο στο `psk31.js` ή στο `olivia.js` αντί για τη δική του αλυσίδα διευκρινιστή, δανειζόμενο ωστόσο τη διαμόρφωση, τον worker και τη διαχείριση συμβάντων του — έτσι τα `fsk.worker.js`, `fskWorkerProxy.js` και `audio.js` δεν χρειάζεται να γνωρίζουν τίποτε για τους δύο τρόπους, και το περιβάλλον χρήστη καταναλώνει παντού τα ίδια συμβάντα `char/status/metrics`.

- `psk31.js` — BPSK31: μιγαδική βασική ζώνη, προσαρμοσμένο φίλτρο, διαφορική ανίχνευση και `varicode`, με φασματική χονδρική απόκτηση και λεπτό AFC εύρους περίπου  $\pm 25$  Hz.
- `olivia.js` — Olivia MFSK: μεταφορά του δέκτη MFSK του Pawel Jalocho από το `fldigi` (`pj_mfsk.h`, GPL-3, όπως και αυτό το έργο), μαζί με τη διόρθωση σφαλμάτων Walsh/Hadamard και την τυφλή αναζήτηση συγχρονισμού σε φάση μπλοκ και μετατόπιση συχνότητας.
- `broadcastSchedules.js` — τα ωράρια UTC που προσφέρουν ως προεπιλογές οι αποκωδικοποιητές FAX, NAVTEX και RTTY, από τα προγράμματα ναυτιλιακού fax της NOAA/NWS και τους δημοσιευμένους καταλόγους σταθμών NAVTEX

#### 4β. Διαφορική λήψη (`diversity.js`)

- Συνδυάζει τον τοπικό δέκτη με έναν δεύτερο αλλού και ακολουθεί όποια τοποθεσία έχει εκείνη τη στιγμή το καλύτερο σήμα. Απλό JS, όχι component — βρίσκεται σε ένα σημείο του `audio.js`, που του δίνει το τοπικό PCM και αναπαράγει ό,τι επιστρέφει
- **Επιλογή, όχι πρόσθεση.** Δύο τοποθεσίες ακούν την ίδια εκπομπή μέσα από διαφορετικές ιονοσφαιρικές διαδρομές, άρα οι κυματομορφές τους έχουν άσχετη φάση· η πρόσθεσή τους

ακούγεται «χτενισμένη». Η σύμφωνη σύνθεση θα απαιτούσε κοινό ρολόι, που δύο δέκτες μέσω internet δεν μοιράζονται. Οι δύο ροές αναμειγνύονται μόνο κατά τη διάρκεια ενός crossfade 30 ms

- Η ευθυγράμμιση συσχετίζει τις δύο **περιβάλλουσες ήχου** (λογαριθμική ισχύς στα 100 Hz), ποτέ τις κυματομορφές — η περιβάλλουσα επιβιώνει και της διαδρομής και οποιουδήποτε codec. Ένα κλείδωμα γίνεται αποδεκτό μόνο όταν μια δεύτερη, ανεξάρτητη αναζήτηση συμφωνήσει, κάτι που απορρίπτει τη «σίγουρη αλλά λάθος» καθυστέρηση που θα έδιναν δύο τοποθεσίες με fading σε αντίθετη φάση
- Ο απομακρυσμένος **κλειδώνεται σε ρυθμό** με την τοπική ροή πρώτα. Δύο δέκτες σημαίνει δύο ρολόγια και δύο αλυσίδες αποδεκατισμού, οπότε ο ήχος τους φτάνει έως και 2% διαφορετικά ακόμη κι όταν και οι δύο δηλώνουν 12 kHz — 240 δείγματα ανά δευτερόλεπτο ολίσθησης, που καμία συσχέτιση δεν συγκρατεί. Ο λόγος μετριέται από το πόσα δείγματα παραδίδει πραγματικά η κάθε πλευρά και εφαρμόζεται από resampler που μεταφέρει την κλασματική του φάση μεταξύ μπλοκ, ώστε οποιοσδήποτε λόγος να κρατά επ' αόριστον
- Η επιλογή τοποθεσίας χρησιμοποιεί SNR εκατοστημορίων μετρημένο σε **ευθυγραμμισμένα ως προς το περιεχόμενο** δείγματα, με υστέρηση, χρονικό dwell και γρήγορη διέξοδο όταν η ενεργή τοποθεσία καταρρεύσει. Οι στάθμες ταιριάζονται θόρυβο-προς-θόρυβο, ώστε η εναλλαγή να μην αλλάζει το υπόβαθρο
- Οι αποκωδικοποιητές κρατούν την **τοπική** ροή: FT8, JS8, WSPR και RADE ολοκληρώνουν σύμφωνα σε μια χρονοθυρίδα, και μια εναλλαγή στη μέση της είναι ασυνέχεια φάσης που μπορεί να στοιχίσει την αποκωδικοποίηση

#### 4γ. Πηγές diversity (remoteSource.js, kiwiSource.js, uberSource.js, webSdrSource.js)

- Ένα συμβόλαιο — onPcm / onState / tune / canReceive — ώστε το diversity.js να μη μαθαίνει ποτέ τι υπάρχει στην άλλη άκρη. Η προσθήκη τύπου δέκτη είναι ένα νέο αρχείο
- remoteSource.js — άλλος PhantomSDR-Plus μέσω /audio (cbor + FLAC), με επαναχρησιμοποίηση της createDecoder() από το lib/wrappers.js
- kiwiSource.js — KiwiSDR: πλαίσια SND, PCM big-endian, με τη χρονοσήμανση GPS 10 byte που εισάγει ένα στερεοφωνικό πακέτο πριν από τον ήχο
- uberSource.js — UberSDR μέσω του δικού του /ws: Opus σε κεφαλίδα 21 byte, με επανασυντονισμό πάνω στην ανοιχτή σύνδεση. Το session id πρέπει πρώτα να δηλωθεί με POST /connection και πρέπει να είναι UUID
- webSdrSource.js — WebSDR, μέσω του websdr\_relay.py σε αυτόν τον server: ο browser δεν μπορεί να συνδεθεί απευθείας, επειδή ο WebSDR ελέγχει την κεφαλίδα Origin και κανένα script δεν επιτρέπεται να την αλλάξει. Ο συντονισμός γίνεται με πλαίσιο κειμένου στην ίδια σύνδεση· την κάλυψη μπαντών τη δίνει το relay
- webSdrCodec.js — η μορφή ήχου του WebSDR: ροή με ετικέτες ανά byte, όπου τα συμπιεσμένα μπλοκ τροφοδοτούν προβλεπτή leaky-LMS 20 λήψεων. Μεταφέρθηκε από τον client του WebSDR και επαληθεύτηκε δείγμα προς δείγμα
- diversityList.js — η λίστα αποθηκευμένων δεκτών και οι κανόνες διευθύνσεων για τους τέσσερις τύπους, κοινά για τον πίνακα του υπολογιστή και τη σελίδα κινητού, ώστε μία μορφή να εξυπηρετεί και τα δύο. Επίσης η μεταφορά — ένα συμπαγές blob, ο κωδικός QR του, και ένας parser που δέχεται κάθε μορφή που έχουν γράψει ποτέ οι δύο σελίδες — και η browseUrl(), που μετατρέπει μια διεύθυνση κλήσης πίσω σε μία που ανοίγει ο browser
- lib/DiversityPanel.svelte — το UI: διεύθυνση, τύπος πηγής, SNR trim και ζωντανή κατάσταση, μαζί με τους αποθηκευμένους δέκτες — με ονόματα, με ελεύθερη σειρά, και με export/import σε αρχείο JSON, ξεχωριστά ανά τύπο πηγής στο localStorage. Όλες οι διορθώσεις γίνονται μέσα στον πίνακα: τα prompt() και confirm() μπλοκάρουν το main thread, εκεί όπου προωθούνται τα δείγματα ήχου προς το playback worklet. Ένα κουμπί  QR σχεδιάζει τη λίστα ως κωδικό προς σάρωση, επειδή το localStorage ανήκει σε έναν browser και ένα κινητό ξεκινά άδριο. Το mobile/

Mobile.svelte έχει την ίδια λειτουργία σε μια καρτέλα **Div**, με το δικό της απλό CSS. Δείτε [Διαφορική λήψη](#)

#### 46. Αναγνώριση τρόπου εκπομπής (modeId.js, modePriors.js)

- Απαντά στο «τι ακούω;». Διαβάζει την ίδια ακατέργαστη λήψη PCM με τους αποκωδικοποιητές και κατατάσσει τους πιθανούς τρόπους, ώστε ο χειριστής να διαλέξει τον σωστό αποκωδικοποιητή αντί να δοκιμάσει και τους δέκα. Δεν αποκωδικοποιεί ποτέ: μετρά φυσικά χαρακτηριστικά του σήματος και τα βαθμολογεί σε πίνακα γνωστών τρόπων
- Το κατειλημμένο εύρος είναι το συνεχές πλάτος στα  $-15$  dB γύρω από την κορυφή, σκόπιμα όχι τιμή 99 % ισχύος: τα κλικ χειρισμού αφήνουν μακριές ουρές στο ολοκλήρωμα ισχύος και έκαναν κάθε στενό τρόπο να φαίνεται πολλαπλάσια πλατύς. Ο ρυθμός συμβόλων προκύπτει από τη **στιγμιαία συχνότητα** και όχι από τις ενέργειες των τόνων, επειδή καμία διάρκεια ολοκλήρωσης δεν αναλύει ταυτόχρονα μετατόπιση 170 Hz και σύμβολο 100 Bd. Ο ρυθμός χειρισμού προκύπτει από την περιβάλλουσα on/off και χρησιμεύει και ως ένδειξη ταχύτητας CW
- Το modePriors.js προσθέτει τη μία ένδειξη που δεν μπορεί να φέρει ο ήχος: πού είστε συντονισμένοι. Σήμα 100 Bd / 170 Hz στους 518 kHz είναι NAVTEX· το ίδιο σήμα στους 14,070 MHz δεν είναι. Απλώς αναβαθμολογεί ό,τι ήδη στήριξε το σήμα και ποτέ δεν επινοεί υποψήφιο
- Τα FT8, JS8 και FT2 αναφέρονται σκόπιμα ως μία ομάδα: δεν διαχωρίζονται μόνο από εύρος και απόσταση τόνων, και το αντίθετο θα ήταν μια σίγουρη λάθος απάντηση
- Κάτω από περίπου 10 dB SNR σιωπά αντί να μαντεύει
- Τρέχει σε δικό του Web Worker (modeId.worker.js + modeIdWorkerProxy.js), με το ίδιο μοτίβο engine/worker/proxy όπως οι αποκωδικοποιητές παραπάνω· το αποτέλεσμα είναι το chip στο lib/ModeIdChip.svelte

#### 4ε. Απορρίψεις συνδέσεων (refused.js, clientVersion.js)

- Το refused.js είναι το κοινό λεξιλόγιο για μια σύνδεση που ο διακομιστής διώχνει: κωδικός κλεισίματος **4003** (πάνω από όριο ανά διεύθυνση, ή η σελίδα είναι παλαιότερη από το [server] min\_client\_version) και **4001** (αποβολή από τον διαχειριστή). Και οι δύο είναι οριστικοί. Τα audio.js, waterfall.js και events.js εισάγουν όλα το isRefusal() από εκεί, γιατί και τα τρία ανοίγουν σύνδεση που μπορεί να απορριφθεί και πρέπει να τερματίσουν την υπόσχεση αρχικοποίησής τους όταν συμβεί — αλλιώς η σελίδα περιμένει για πάντα μια σύνδεση που δεν θα έρθει ποτέ
- Τίποτα δεν ξαναδοκιμάζει. Μια πεσμένη σύνδεση /audio τερματίζει τη συνεδρία εκ σχεδιασμού: τα /waterfall και /events δεν επέστρεφαν ποτέ μαζί με μια επανασύνδεση, οπότε μια συνεδρία που ξαναδοκιμάζε ήταν ζωντανός ήχος πάνω σε παγωμένο καταρράκτη, και απέναντι σε όριο ρυθμού κάθε επανάληψη θα παρέτεινε ακριβώς την απόρριψη που προσπαθούσε να παρακάμψει. Η σελίδα υπολογιστή εξηγεί μια απόρριψη που συμβαίνει κατά τη φόρτωση και απλώς σταματάει όταν συμβεί στη μέση· το /mobile δείχνει μία γραμμή που ζητά από τον ακροατή να ανανεώσει
- Το clientVersion.js κρατάει έναν μόνο ακέραιο, το CLIENT\_VERSION, που η σελίδα προσθέτει στη σύνδεση ήχου ως /audio?v=N. Ο διακομιστής απορρίπτει ό,τι είναι κάτω από το [server] min\_client\_version, και έτσι ένας σταθμός αναγκάζει καρτέλες που τρέχουν ακόμη παλιότερη έκδοση να ανανεώσουν — ο μόνος μοχλός που υπάρχει, αφού ο διακομιστής δεν μπορεί να φτάσει σε JavaScript που ήδη τρέχει σε browser. Αυξήστε τον όταν μια αλλαγή στο frontend δεν πρέπει να συνεχίσει να αγνοείται από ανοιχτές καρτέλες
- Πλήρης αναφορά: [Όρια συνδέσεων](#)

#### 4στ. Απαντήσεις στη συνομιλία (lib/chatReply.js)

- Μια απάντηση είναι ένα συνηθισμένο μήνυμα συνομιλίας που ξεκινά με μια σήμανση η οποία ονομάζει το μήνυμα στο οποίο απαντά, [RE:<χρονοσήμανση>|<όνομα>]· ο διακομιστής την αποθηκεύει και τη μεταδίδει όπως κάθε άλλη γραμμή και δεν χρειάζεται καμία αλλαγή

- Το `chatReply.js` δημιουργεί και διαβάσει αυτή τη σήμανση και ομαδοποιεί τις απαντήσεις κάτω από το αρχικό μήνυμα, σε ένα επίπεδο βάθους. Το χρησιμοποιούν και το `App.svelte` και το `mobile/Mobile.svelte`, ώστε οι δύο σελίδες να οργανώνουν τη συνομιλία με τον ίδιο τρόπο
- Όταν το αρχικό μήνυμα έχει ήδη φύγει από το ιστορικό των 20 μηνυμάτων του διακομιστή, η απάντηση μένει εκεί όπου έφτασε και δείχνει μόνο το όνομα και την ώρα στα οποία απαντά

## 5. Διαχείριση κατάστασης (stores/)

- Αντιδραστικοί χώροι δεδομένων
- Κοινή κατάσταση εφαρμογής
- Χειρισμός συμβάντων

## Βασικές δυνατότητες

- Επεξεργασία ήχου σε πραγματικό χρόνο
- Αποκωδικοποίηση ψηφιακών τρόπων (FT8, RTTY κ.λπ.)
- Φιλτράρισμα ήχου
- Φασματική ανάλυση

## Λίστες συχνотήτων (frequencylist/)

Οι δείκτες συχνотήτων πάνω στον καταρράκτη. Το `mymarkers.json` είναι η λίστα που δείχνει πραγματικά ο δέκτης· τα υπόλοιπα είναι η πρώτη ύλη που το `update-markers.sh` μετατρέπει σε αυτήν, ανανεωμένη από τα διαδικτυακά προγράμματα. Το `README.md` σε εκείνον τον φάκελο εξηγεί την ενημέρωση και στις επτά γλώσσες.

```
frequencylist/
├─ mymarkers.json # οι δείκτες που δείχνει ο δέκτης – επεξεργάζονται με το χέρι,
 επιβιώνουν της ενημέρωσης
├─ shortwavestations.json # σταθμοί βραχέων, παράγεται από το generate-current-shortwave.py
├─ 0.TXT # το παγκόσμιο πρόγραμμα HF A26, αποσυμπιέζεται και αναλύεται από το
update-markers.sh
├─ a26allx2.zip # το αρχείο-πηγή που κατεβάζει το update-markers.sh
├─ admin.txt · antenna.txt · broadcas.txt · fmorg.txt · language.txt · site.txt
├─ # βοηθητικές λίστες από την ίδια πηγή (τοποθεσίες, γλώσσες,
διαχειριστές, κεραίες)
├─ generate-current-shortwave.py
├─ update-markers.sh # ανανεώνει όλα τα παραπάνω από τα διαδικτυακά προγράμματα
└─ README.md
```

## Μορφή (mymarkers.json, shortwavestations.json)

```
[
 { "frequency": 77500, "name": "DCF77", "mode": "CW" },
 { "frequency": 2485000, "name": "Vanuatu Broadcasting", "mode": "AM" }
]
```

## Αρχεία ρυθμίσεων

### Ρυθμίσεις διακομιστή (αρχεία .toml)

Δομή των αρχείων ρυθμίσεων:

```
[server]
Web server settings
port = 9002
html_root = "frontend/dist/"
threads = 2
otherusers = 1

[limits]
Όρια συνδέσεων ανά διεύθυνση – όλα είναι εξ ορισμού ανενεργά ή σε λογική τιμή,
οπότε μια ρύθμιση χωρίς αυτά συμπεριφέρεται όπως πάντα. Δείτε CONNECTION_LIMITS.md.
per_ip = 3 # ταυτόχρονοι ακροατές από μία διεύθυνση
per_ip_rate = 40 # νέες συνδέσεις ανά λεπτό από μία διεύθυνση

[websdr]
Online registration
register_online = true
name = "WebSDR Name"
antenna = "Antenna Type"
grid_locator = "AB12cd"
hostname = "domain.com"

[input]
SDR input settings
sps = 2048000 # Sample rate
fft_size = 131072 # FFT size
frequency = 145000000 # Base frequency
signal = "iq" # Signal type: "iq" or "real"
audio_sps = 12000 # Audio sample rate
audio_compression = "opus" # "flac" or "opus"
accelerator = "opencl" # "none", "cuda", "opencl"

[input.driver]
Driver settings
name = "stdin"
format = "u8" # Sample format

[input.defaults]
User interface defaults
frequency = 145500000
modulation = "FM"
```

## Πληροφορίες σταθμού (site\_information.json)

```
{
 "siteSysop": "Operator Callsign",
 "siteSysopEmailAddress": "email@example.com",
 "siteGridSquare": "AB12cd",
 "siteCity": "City, Country",
 "siteInformation": "https://github.com/sv1btl/PhantomSDR-Plus",
 "siteHardware": "Hardware specs",
 "siteSoftware": "Software version",
 "siteReceiver": "SDR model",
 "siteAntenna": "Antenna description",
 "siteNote": "Additional notes",
 "siteIP": "http://domain.com:9002",
 "siteSDRBaseFrequency": 0,
 "siteSDRBandwidth": 2048000,
 "siteRegion": 1,
 "siteChatEnabled": true
}
```

## Σύστημα μεταγλώττισης

### Ρύθμιση μεταγλώττισης Meson

#### `meson.build` (ρίζα)

Ορίζει:

- Μεταδεδομένα του έργου
- Εξαρτήσεις
- Επιλογές μεταγλωττιστή
- Λίστες αρχείων πηγαιού κώδικα
- Στόχους μεταγλώττισης

#### `meson_options.txt`

Διαθέσιμες επιλογές:

```
option('opencl', type: 'boolean', value: false, description: 'Enable OpenCL support')
option('cuda', type: 'boolean', value: false, description: 'Enable CUDA support')
option('optimization', type: 'string', value: '3', description: 'Optimization level')
```

## Εξαρτήσεις αρχείων

### Εξαρτήσεις μεταγλώττισης του backend

```
spectrumserver binary depends on:
├─ C++ source files (src/**/*.cpp)
├─ External libraries:
│ ├── FFTW3
│ ├── WebSocket++
│ ├── FLAC
│ ├── Opus
│ ├── Liquid-DSP
│ ├── Boost
│ ├── zlib
│ ├── zstd
│ └─ OpenCL/CUDA (optional)
└─ Subproject headers:
 ├── nlohmann/json
 └─ toml11
```

### Εξαρτήσεις μεταγλώττισης του frontend

```
frontend/dist/ depends on:
├─ Source files (frontend/src/**)
├─ npm packages (node_modules/):
│ ├── Svelte
│ ├── Vite
│ ├── @wasm-audio-decoders/opus-ml
│ └─ ...
└─ Static assets (frontend/public/)
```

## Ροή δεδομένων

### Ροή λειτουργίας του διακομιστή

1. SDR Hardware → rtl\_sdr/hackrf\_transfer/etc.  
↓
2. Sample Stream → stdin → spectrumserver  
↓
3. spectrumserver:
  - FFT calculation (waterfall)
  - Demodulation (audio)
  - Compression (FLAC/Opus)
 ↓
4. WebSocket → Browser Client  
↓
5. Browser:
  - Render waterfall
  - Decode and play audio
  - Display controls

### Ροή αλληλεπίδρασης χρήστη

1. User clicks on waterfall  
↓
2. JavaScript sends frequency change request  
↓
3. WebSocket → spectrumserver  
↓
4. spectrumserver:
  - Updates demodulator frequency
  - Sends new audio stream
 ↓
5. Browser receives and plays new audio

## Οδηγός τροποποίησης αρχείων

### Όταν τροποποιείτε κώδικα του backend (src/\*\*):

```
cd PhantomSDR-Plus
meson compile -C build
Server restart required
```

### Όταν τροποποιείτε κώδικα του frontend (frontend/src/\*\*):

```
cd PhantomSDR-Plus/frontend
npm run build
cd ..
Server restart required (for static files)
```

### Όταν τροποποιείτε ρυθμίσεις (.toml, .json):

```
Restart server
./stop-websdr.sh
./start-rtl.sh # (or appropriate start script)
```

**Όταν τροποποιείτε τους δείκτες (markers.json):**

```
Reload page in browser
No server restart needed
```

## Σημαντικές διαδρομές

### Διαδρομές εκτέλεσης

- Ρυθμίσεις: `./config-*.toml`
- Ριζικός κατάλογος HTML: `./frontend/dist/`
- Δείκτες: `./markers.json`
- Ιστορικό συνομιλίας: `./chat_history.txt`
- FFTW wisdom: `./fftw_wisdom`, `./phantom_fftw_wisdom`

### Διαδρομές μεταγλώττισης

- Παραγόμενο εκτελέσιμο: `./build/spectrumserver`
- Έξοδος του frontend: `./frontend/dist/`
- Node modules: `./frontend/node_modules/`

### Διαδρομές πηγαίου κώδικα

- Πηγαίος κώδικας backend: `./src/`
- Πηγαίος κώδικας frontend: `./frontend/src/`
- Βιβλιοθήκες DSP: `./jsdsp/`

## Συνήθειες εργασίες με αρχεία

### Προσθήκη νέας ρύθμισης SDR

1. Αντιγράψτε υπάρχουσα ρύθμιση: `cp config-rtl.toml config-mydevice.toml`
2. Επεξεργαστείτε τις παραμέτρους: `nano config-mydevice.toml`
3. Δημιουργήστε σενάριο εκκίνησης: `cp start-rtl.sh start-mydevice.sh`
4. Επεξεργαστείτε το σενάριο εκκίνησης: `nano start-mydevice.sh` — αλλάξτε μόνο το τμήμα **RECEIVER CONFIGURATION** στην αρχή (`RX_LABEL`, `RX_COMM` = το όνομα διεργασίας του δέκτη, `RX_ARGS`, `CONFIG`, `FIFO` και το άγκιστρο `prestart` αν το χρειάζεται η συσκευή). Η λογική εκκίνησης/`watchdog`/καταγραφής από κάτω είναι γενική και δεν χρειάζεται αλλαγές.
5. Κάντε το εκτελέσιμο: `chmod +x start-mydevice.sh`

Το `stop-websdr.sh` ήδη σταματά οποιοδήποτε `start-*.sh --watchdog`· αν ο δέκτης σας χρησιμοποιεί όνομα διεργασίας διαφορετικό από `rx888_stream/rx_sdr/rtl_sdr/hackrf_transfer/cf32_to_real`, προσθέστε και εκεί μια γραμμή `killall -9 <όνομα>`.

### Προσαρμογή του frontend

1. Τροποποιήστε τον πηγαίο κώδικα: `nano frontend/src/App.svelte`
2. Μεταγλωττίστε ξανά: `cd frontend && npm run build && cd ..`
3. Επανεκκινήστε τον διακομιστή: `./stop-websdr.sh && ./start-rtl.sh`

## Προσθήκη προσαρμοσμένων δεικτών

1. Επεξεργαστείτε το αρχείο δεικτών: `nano markers.json`
  2. Μορφή: `json { "markers": [ { "frequency": 145500000, "label": "2m Calling", "mode": "FM" } ] }`
  3. Ανανεώστε τον περιηγητή (δεν χρειάζεται επανεκκίνηση διακομιστή)
- 

## Έλεγχος εκδόσεων

---

### Αρχεία που παρακολουθούνται στο Git

- Πηγαίος κώδικας (`src/`, `frontend/src/`, `jsdsp/`)
- Παραδείγματα ρυθμίσεων (`config.example.*.toml`)
- Σύστημα μεταγλώττισης (`meson.build`, `meson_options.txt`)
- Τεκμηρίωση (`*.md`, `docs/`)
- Σενάρια (`*.sh`)

### Αρχεία που αγνοούνται (`.gitignore`)

- Αποτελέσματα μεταγλώττισης (`build/`, `frontend/dist/`)
  - Εξαρτήσεις (`frontend/node_modules/`)
  - Δεδομένα χρηστών (`chat_history.txt`)
  - Προσωπικές ρυθμίσεις (`config-rtl.toml` αν έχει προσαρμοστεί)
  - Δυναμικά δεδομένα (`*.o`, `*.so`)
- 

**Αυτή η τεκμηρίωση της δομής θα σας βοηθήσει να περιηγηθείτε και να κατανοήσετε τον κώδικα του PhantomSDR-Plus.**

Για οδηγίες εγκατάστασης, δείτε το [INSTALLATION.md](#). Για πληροφορίες χρήσης, δείτε το [USER\\_GUIDE.md](#).

**73 de SV1BTL & SV2AMK**