



MERCURY

A free software HF modem and reliable data link

Email, files and messages, from below -10 dB SNR to 3.1 kbit/s

Rhizomatica • **the modem of HERMES** • **October 2026**

github.com/Rhizomatica/mercury • hermes.radio



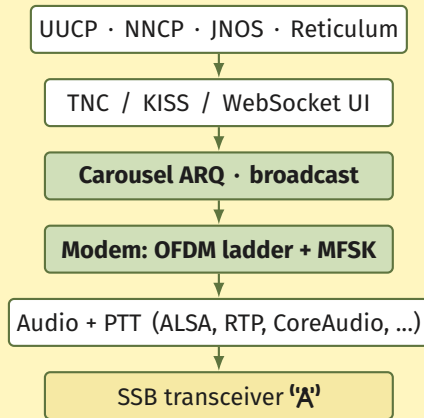
1. Mercury in HERMES
2. Modulation and coding modes
3. MFSK for weak-signal operation
4. Carousel ARQ
5. Formal methods: TLA+ and TLAPS
6. Testing and channel simulation
7. Integration and platforms
8. Outlook



MERCURY IN HERMES

Overview

- ▶ The HF software modem of **HERMES**: it turns an SSB radio into a data link for email, files and messages
- ▶ Hands the application a **TCP connection**: a VARA-compatible TNC (control 8300, data 8301), KISS and broadcast on 8100
- ▶ A clean **rewrite in C** (v2, since March 2026); the C++ v1 lives on as `mercuryv1`
- ▶ **Free software**: GPL-3.0-or-later; vendored codec2 subset LGPL-2.1
- ▶ Built by Rhizomatica for communities beyond the Internet, first in the Amazon rainforest



1109

commits on trunk since the C
rewrite (2026-03-05)

496

commits since 1 September

5

releases since September (1.9.15
→ 1.9.19)

Platforms

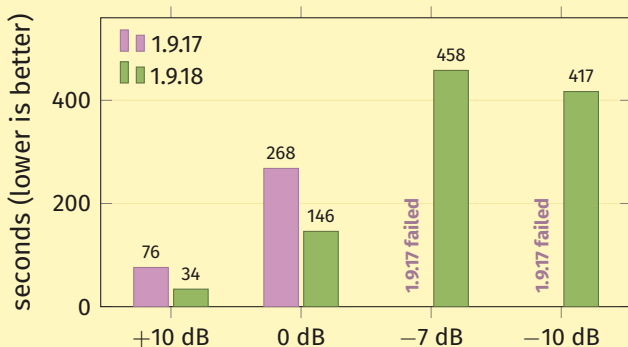
Linux (amd64, arm64, armhf, i386),
Raspberry Pi 4/5, Windows (signed),
macOS (universal, notarised); **iOS and
Android coming soon**

Releases

1.9.18 (12 Oct): the carousel data plane

1.9.19 (13 Oct): MFSK centred on 1500 Hz,
hence air-protocol version 2 (does not
interoperate with 1.9.18); JNOS 2.0 support

Release 1.9.18 compared with 1.9.17: a 2 KB transfer



- ▶ Two real daemons through codec2 ch, 2074 bytes
- ▶ **2× faster** on a good link
- ▶ Links down to **-10 dB** SNR now complete
- ▶ On air, 8 KB each way:
485/509 s → **312/238 s**



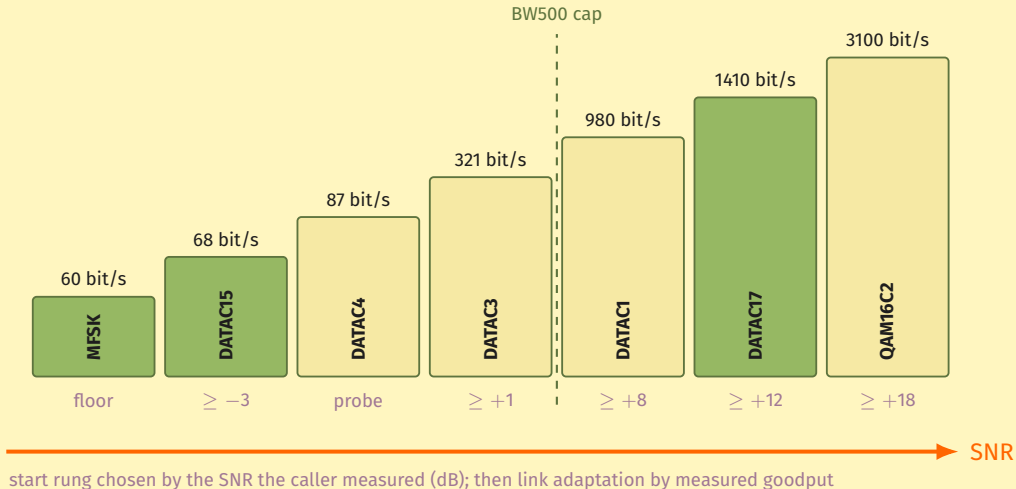
MODULATION AND CODING MODES

Mode overview

Mode	Origin	BW, Hz	Payload, B	bit/s	FEC (n,k)	Role
MFSK	Mercury	1000	98	~60	LDPC (1600,800)	rung 0 (floor)
DATA15	Mercury	200	30	68	(768,256)	rung 1
DATA4	codec2	250	54	87	(1472,448)	rung 2
DATA3	codec2	500	126	321	(2048,1024)	rung 3 (BW500 top)
DATA1	codec2	1700	510	980	(8192,4096)	rung 4
DATA17	Mercury	2100	1180	1410	(15936,9456)	rung 5
QAM16C2	codec2	2100	1213	3100	(16200,9720)	rung 6
DATA16	Mercury	200	14	42	(640,128)	ARQ control
DATA0/13/14	codec2	200–500	3–14	58–291	—	broadcast, tests

green = designed by us: three of the seven rungs, and the control mode

Adaptive mode ladder: seven rungs from 60 to 3100 bit/s



OFDM modes designed by the project

Robust: DATAC15 and DATAC16

- ▶ Rate-1/3 LDPC (H_256_768_22), 200 Hz
- ▶ DATAC15: 30 bytes, 3 carriers, the first OFDM rung
- ▶ DATAC16: 14 bytes, calls, polls and ACKs
- ▶ 70/100 frames at -7 dB on MPP fading

Fast: DATAC17, on codec2's QAM16C2 code

- ▶ 2.1 kHz, about 1.2 KB a frame
- ▶ codec2's QAM16C2: 16-QAM, (16200,9720), 3100 bit/s
- ▶ DATAC17: QPSK on that codeword, 1410 bit/s
- ▶ DATAC17 97/100 at $+10$ dB (MPP)

Accuracy of link adaptation

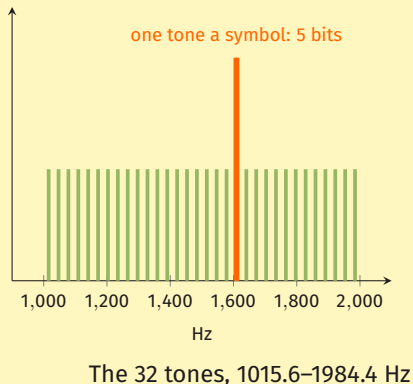
Per-mode **SNR calibration** (QAM16C2 read 9 dB low); a 16-QAM estimator above 12 dB; the **TX filter** no longer cuts our outer carriers (6.5 dB); **FFT-correlation** acquisition, $2.2\times$ cheaper



MFSK FOR WEAK-SIGNAL OPERATION

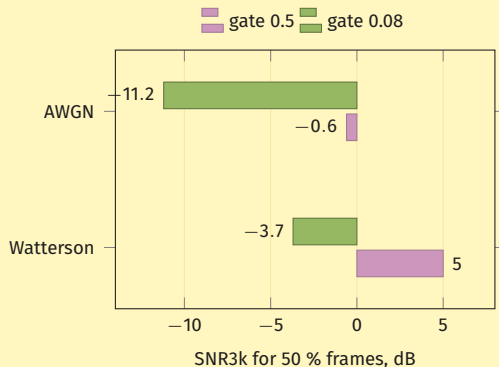
MFSK: a non-coherent waveform below the OFDM modes

- ▶ Mercury's own modem: **32-FSK on OFDM subcarriers**, non-coherent
- ▶ 8 kHz, Nfft 256: tones 31.25 Hz apart, 40 ms symbols (25 baud)
- ▶ Centred on **1500 Hz** since 1.9.19: tones 1015.6–1984.4 Hz
- ▶ Preamble *and* postamble: a burst is found from either end
- ▶ LDPC N = 1600; five rates ported, **rate 1/2 ships** (98-byte frames, 13.5 s)
- ▶ Reducing the code rate to 1/16 gains only 2.1 dB: sensitivity is limited by synchronisation



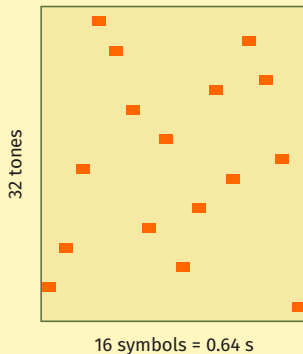
Sensitivity limited by synchronisation, not by coding

- ▶ All five code rates gave **identical** frame error rates: frames near the threshold never reached the decoder
- ▶ The sync accept gate was 0.5, which demands SNR per symbol ≥ 0 dB
- ▶ At **0.08**, with false anchors released:
 - AWGN 50 % point: $-0.6 \rightarrow -11.2$ dB
 - Watterson 2 ms / 1 Hz: $+5.0 \rightarrow -3.7$ dB
- ▶ Dual-ended acquisition contributes about 2 dB; codeword interleaving adds 19 % throughput at the margin



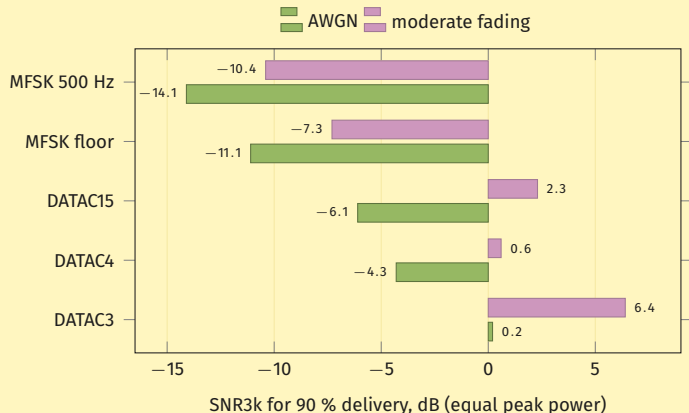
Pattern acknowledgements: 0.64 s of airtime

- ▶ An acknowledgement as a **Welch-Costas pattern**: 16 MFSK symbols, **0.64 s**
- ▶ $5.8\times$ less airtime than a 3.74 s DATAC16 frame
- ▶ Detected every time down to about -9 dB on fading and -22 dB on AWGN
- ▶ **0 false alarms** in 30 trials down to -26 dB; 0 in 2 h of noise after the half-subcarrier search
- ▶ Frequency search over ± 78 Hz: an IC-7100 and an sBitx went from 0/5 to 17/17 patterns on air



(illustrative layout: one tone per symbol, no repeats)

Sensitivity of MFSK compared with the OFDM modes



- ▶ MFSK operates **5 to 8 dB** below DATAC15 on AWGN, **10 to 13 dB** under moderate fading
- ▶ Against modem73's coherent RFDM modes: 2.7 dB better per bit on AWGN, a tie under fading

Under review: a 500 Hz MFSK floor for BW500 sessions

MFSK500: the BW500 floor (PR #390)

- ▶ The 1 kHz MFSK cannot be decoded behind a 500 Hz filter
- ▶ 32 tones 15.625 Hz apart, 80 ms symbols, 26 s frames
- ▶ 90 %: **-14.1 dB** AWGN, -10.4 dB moderate fading, behind the filter too
- ▶ Searches at 2 kHz: idle receive **cheaper** than the 1 kHz floor
- ▶ BW2300 and BW2750 keep the 1 kHz floor; air-protocol version 3

Receiver fixes, all MFSK modes

- ▶ Dial offset to an eighth of a subcarrier
- ▶ Pattern search in half-subcarrier steps (removes a 3–4 dB loss)
- ▶ LLR noise from the tones' own bins
- ▶ A burst with a **gap in its audio** (a playback underrun at the transmitter) is read across the gap

On air, two bench stations

Every complete MFSK500 frame was decoded in both directions, including frames hit by underruns. BW500 sessions on the floor were answered by its patterns.



CAROUSEL ARQ

Motivation for a new ARQ

Previous design: stop-and-wait

- ▶ SNR-driven: one frame, one ACK, one turnaround
- ▶ Rungs chosen from SNR estimates that were wrong per mode
- ▶ Lost ACKs on the weak reverse path stalled the session
- ▶ Sim NVIS: **0 of 20** runs completed

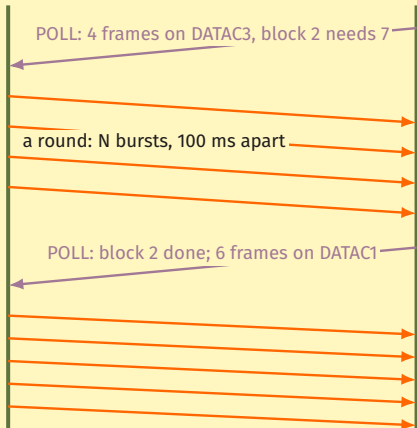
Carousel design

- ▶ **Receiver-driven:** the receiver requests what it still needs
- ▶ **Erasure-coded:** any k of the pieces rebuild a block, so no retransmission of a particular frame
- ▶ No per-frame ACKs, no sequence numbers
- ▶ Rungs chosen by **measured goodput**
- ▶ Merged 3 Oct (#322), the only data plane since 11 Oct (#364)

Carousel operation

Sender

Receiver



- ▶ Systematic Cauchy **Reed-Solomon** over $GF(256)$, blocks of up to 96 pieces
- ▶ **23–24 byte pieces**: four fit one MFSK frame
- ▶ Up to 8 blocks open; each ends in a seed-keyed CRC-32 and fails closed
- ▶ Four consecutive losses mark a rung unusable; it is probed again after 60 s, the interval doubling up to 8 min

Channel access, connection set-up and versioning

Single timing master

The caller owns every timer; the callee keys only in its slots. **0 collisions** in 960 two-way deep-fade sim runs (#351)

Connection at low SNR

After two unanswered DATAC16 CALLs, the CALL alternates with MFSK. The ACCEPT names the start rung and carries the callee's nonce

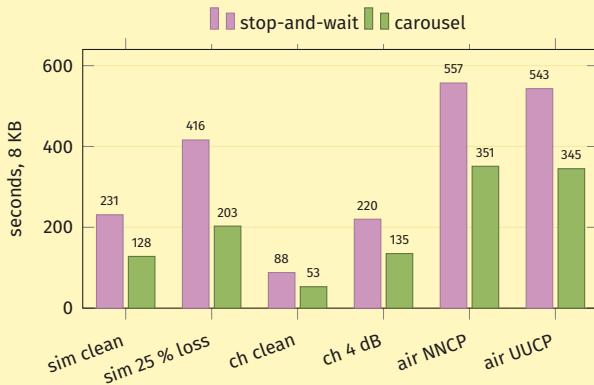
Session binding

Seed = CRC16(session id, nonce, both calls), mixed into every frame's CRC. CALL and ACCEPT carry the air-protocol version (1.9.18: 1, 1.9.19: 2); a CALL of another version is logged, not answered

Listen before talk

Every keying path waits for a clear channel; the busy detector is opt-in (off by default)

Results: simulation, modem bench and on-air tests



- ▶ **35–45 % shorter** transfer times in every measurement
- ▶ Simulated NVIS: stop-and-wait never completed; the carousel did
- ▶ On air: mean of two runs in each direction, sBitx test bench



FORMAL METHODS: TLA+ AND TLAPS

Formal models of the protocol layers

Model	What it proves	Checked
CarouselBlocks	InOrder, Integrity, RetireSafe, NoPhantom: delivery in order, nothing invented, nothing retired early	TLC up to 63.4 M states ; an adversary config at 12.1 M
Liveness	Completes: every block is delivered, under a lossy adversary	2.58 M states, 14.2 h
CarouselProofs	Spec $\Rightarrow$ $\square$ Safety for any number of blocks	TLAPS: 823 obligations (Z3, Zenon)
CarouselTurns (timed)	NoOverlap: the two stations never key together; MNotShutOut	18.5 M states; MFSK config 27.1 M
RungAgreement	Agree, Wasted, Delivered: both ends agree on the rung	1458 states; 5592 at 4 rungs

Validation of the models

Mutation testing

- ▶ Turns: five mutants (no grid, late, long, blind, reuse) each violate NoOverlap
- ▶ Rungs: five more (no request, no bind, deaf, unnamed, previous rung)
- ▶ Blocks: the no-check config violates Integrity, as it must
- ▶ Before the integrity work, the model found **two real bugs**, 9 steps each

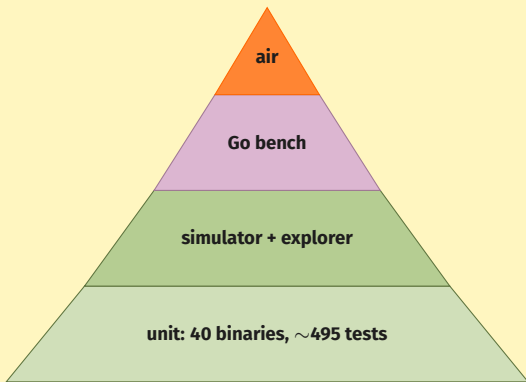
Conformance of the implementation

- ▶ `car_explore` runs the **real** `carousel.c` under every set of up to 3 losses or sensing failures, plus spurious BREAKs and corrupt frames
- ▶ **19 765 runs in 1.5 s** in CI; 9.09 M in the long sweep
- ▶ Overrun and slot oracles in the simulator check the timed model's assumptions



TESTING AND CHANNEL SIMULATION

Test framework



- ▶ **Unit:** modems, LDPC, carousel, ARQ, TNC, JSON, rings, channel model
- ▶ **Simulator:** two state machines on a virtual clock; A/B benches on 120 seeds
- ▶ **Go bench:** two real daemons over FIFO, socket or RTP audio through `ch` or Watterson; two different builds can be paired
- ▶ **On air:** sBitx, IC-7100 and FT-710 stations, UUCP / NNCP / JNOS transfers
- ▶ **One framework:** no second copy of test data, no tool CI cannot run

Continuous integration on every pull request

Builds

Linux amd64, arm64, armhf; Windows x64 (cross); macOS arm64; the Fyne GUI on Linux and Windows

Tests

Unit tests on amd64 and **i386**; the Go integration bench

Sanitizers

ThreadSanitizer on the unit tests; AddressSanitizer + UBSan on the unit tests and a full integration session

Additional tooling

libFuzzer targets for the ARQ header, frame header, KISS, arithmetic and WebSocket JSON parsers; signed and notarised release pipeline

Channel model compliant with ITU-R F.1487

- ▶ Watterson model with the **Gaussian Doppler spectrum** of **ITU-R F.1487** / **CCIR 520-2**: spread = 2σ
- ▶ Named presets
`ccir-good/moderate/poor/flutter` and `iturn-*` (low to high latitude, NVIS)
- ▶ An exact delay line; the former Butterworth presets are retained for reproducibility
- ▶ The former “poor” preset faded **twice as fast** as CCIR Poor: 1.007 against 0.515 fades per second

Acceptance tests in the CI suite

	Measured	Theory
Power beyond 3σ	0.273 %	0.27 %
Power beyond the spread	4.68 %	4.55 %
Fades below -10 dB, s^{-1}	0.5067	0.5072

Reference values: the Gaussian spectrum and Rice's closed-form level-crossing rate. Records of 900 s with fixed seeds.

Cross-validation with MATLAB

Every preset compared with the R2026a Communications Toolbox (`stdchan`, `comm.RayleighChannel`): **26 of 26 checks passed**; a second run reproduced the results byte for byte.

Independent measurement tools

Channel simulators

- ▶ codec2 ch: AWGN and MPP
- ▶ Our Watterson model (above)
- ▶ AE4JY's **PathSim**, calibrated against ours to ± 0.2 dB
- ▶ OpenARQ's **skywave** harness, with a Mercury adapter

Probes and sweeps

- ▶ Pattern, acquisition and burst-train probes
- ▶ MFSK channel sweeps by file, through any channel tool
- ▶ An opt-in RX audio dump: a frame missed on air can be decoded offline
- ▶ On-air recordings with noise added offline: the bench's own fringe curve



INTEGRATION AND PLATFORMS

Supported applications

Mail and files

UUCP is the HERMES default (-Y probe race fixed upstream); NNCP works too; JNOS 2.0 BBS-to-BBS and telnet with no IP on air

Networks

Reticulum over KISS on the broadcast port; broadcast files by a RaptorQ fountain, wire-compatible with hermes-broadcast

Encryption (in review)

Optional Noise_KK sessions (X25519, ChaCha20-Poly1305), XChaCha20 broadcast files; off by default (PR #306)

Interfaces and platforms

- ▶ **Our own WebSocket server:** 28 820 vendored GPL-2.0-only lines replaced by about 1500 of ours (RFC 6455, TLS through OpenSSL)
- ▶ Status: modes, the peer's view of our signal, frequency, **carrier offset**
- ▶ **TUNE:** a 1 kHz tuning carrier, hard 60 s unkey deadline
- ▶ Audio: ALSA, Pulse, CoreAudio, AAudio, WASAPI, DirectSound, **RTP** to hermes-radio-daemon
- ▶ PTT: Hamlib, serial RTS, CM108 GPIO, HERMES shared memory
- ▶ **Fyne GUI** with the engine linked in
- ▶ **Windows:** Authenticode-signed (Certum)
- ▶ **macOS:** universal x86_64 + arm64, static Hamlib, notarised
- ▶ **iOS and Android:** coming soon, the modem and its GUI on the phone
- ▶ **Raspberry Pi 4 and 5,** the HERMES station computers



OUTLOOK

Mercury on iOS and Android

The modem and its GUI on the phone:
coming soon

Optional encryption

Encrypted sessions on the carousel
plane, off by default (Noise_KK, PR
#306)



THANK YOU

 github.com/Rhizomatica/mercury

 hermes.radio

 docs/ in the repository: modes, carousel, specs, channel model

Mercury is free software (GPL-3.0-or-later).

This presentation: CC BY-SA 4.0.

