
Web Remote — Desktop/Laptop Page Changelog (`web/remote.html`)

Covers only the desktop/laptop remote-control page (browse to `http://<pi-hostname>:8765/`, `/index.html`, or `/remote.html`). This page has no on-screen version label of its own (unlike the phone page); entries below are dated instead. Unless noted otherwise, every entry is **web remote only**: re-sync `PiDisplay/` and hard-refresh the browser — no Pi restart or Teensy reflash needed.

2026-10-04 — CW page: Vol/AGC removed

- Vol:/AGC: removed from the CW settings page; page values show “-” with early V1.0.33 builds.

2026-10-04 — CW settings page

- The Settings box switches to the T41’s CW settings page while Decode is on. Needs firmware V1.0.33.

2026-10-04 — Decoded CW strip

- In CW receive with Decode on, the decoded text shows in a strip along the bottom of the waterfall (newest on the right), with WPM. Needs firmware V1.0.33.

2026-10-04 — Brighter 3D waterfall

- Same 3D changes as the Pi: brighter back rows, brighter bottom blues, thin blue base on every row.

2026-10-04 — Waterfall uses the T41’s own colours

- 2D waterfall coloured from the spectrum message’s new `wf` field (firmware V1.0.32) so its noise blue matches the T41; falls back to the trace bins without it.

2026-10-04 — Waterfall blue much brighter

- `brightenBlue()` strengthened to match the Pi; 3D waterfall bottom colours brightened again.

2026-10-04 — Brighter waterfall blue

- Waterfall blues brightened (same change as the Pi, `brightenBlue()`); 3D waterfall’s bottom colours too. Calibration overlay background is solid black.

2026-10-04 — Calibration screens (needs firmware V1.0.30)

- Menu > Calibration > S meter level, Frequency, Transmit IQ and Power enabled. Transmit IQ / Frequency / Power open a full-screen overlay copied from the T41 screen with buttons for every control; arrow keys adjust the main values. S meter level uses the normal parameter box.

2026-10-03 — PI Spectrum Fill starts Off

- The spectrum fill starts Off until Display > PI Spectrum > PI Spectrum Fill > On is picked (matches the Pi screen).

2026-10-03 — Vol: to settings; Up/Down for Vol

- Vol: moved from the data line to the settings grid.
- After the Vol button, Up/Down arrows change the volume too.

2026-10-03 — Audio / Mic / CW dropdowns; Vol button

- **New Audio, Mic and CW toolbar dropdowns** (Pi, web, phone), next to Display. They are the T41’s own **Audio Options**, **Microphone** and **CW Options** menus, copied item for item and in the same order (e.g. Audio: AGC Off/Long/Slow/Medium/Fast, Adjust Equalizers, Toggle Aut notch, the four Noise Reduction choices; Mic: Mic gain, USB drive, USB RX level; CW: WPM ... CW Manual Mode). Picks send the same menu selection as before (and jump straight there with firmware V1.0.28+). Those three categories are no longer listed under Menu.

- **Vol is now a plain button.** One click sets the radio's Volume knob back to audio volume (from AGC/Mic/STn if needed) and points the mouse wheel and Left/Right arrows at it. Mic gain, sidetone and AGC are reached through the new Mic / CW / Audio dropdowns.
- Pi toolbar: rows are now split by width (the top row runs further right, where there's room), and "Course Inc"/"Fine Inc" are shortened to "Course"/"Fine" like the web, so the second row stays clear of the SWR readout. Phone: dropdowns in the right-hand toolbar columns open toward the left so they stay on screen.
- Pi: `_OWN_DROPDOWN_CATEGORIES` / `_OWN_DROPDOWN_LABELS`, `_make_plain_button()`. Web: `buildMenu(..., {only})` for each, `buildVolButton()`.

2026-10-03 — Arrow keys reach Filter

- **Arrow keys now turn the Filter (and Volume) knob on the web remotes.** After you picked Lo Cut / Hi Cut from the Filter dropdown, the dropdown kept keyboard focus, so the arrows changed its own selection (flipping between Lo and Hi) instead of reaching the radio. Toolbar dropdowns now give focus back after a pick, and the arrow handler no longer skips a closed toolbar dropdown.

2026-10-03 — Trackpad fix + arrow keys

- **Trackpad fix (web remotes):** filter Lo/Hi Cut (and every other wheel use) jumped around with a trackpad, because each of the trackpad's many tiny scroll events, including sideways ones, sent a full encoder step. Now:
 - vertical movement is added up and sent as one step per 40px of travel (a mouse-wheel notch is still exactly one step);
 - sideways swipes are ignored;
 - a change of direction restarts the count;
 - steps are at least 35 ms apart. The new code is `wheelStep()`.
- **Arrow keys on every version (Pi, web, phone with a keyboard):**
 - **Left/Right** turn whichever knob the wheel is set to (Filter Lo/Hi Cut or Volume, picked from those dropdowns), one step per press. With nothing picked they tune.
 - **Up/Down** always tune (Shift = fine tune).
 - In a Menu parameter box every arrow turns the Filter knob; on the equalizer screen Left/Right pick the band and Up/Down change its level.
 - Pi: new `display.bind_arrow_keys()` / `radio_ui._on_arrow_key()`. These are ignored while typing in the Network page's password box.

2026-10-03 — Nixie tubes redrawn

- **Freq Font > Nixie redrawn** to look like real Nixie tubes, from the user's reference picture:
 - a tall glass tube with dark end caps and a warm halo;
 - a red-amber honeycomb mesh inside;
 - thin wire digits glowing yellow-orange, drawn as smooth strokes (wide and dim through thin and bright) rather than a font;
 - a full tube of its own for the decimal point, with a glowing dot at the bottom.
- Same shapes on the Pi (`freq_fonts.py`, Tk canvas) and the web (`drawNixieTube()`, canvas glow and gradients).

2026-10-03 — DSP: on data line, Freq Font, dBm readout fix

- "DSP:" added to the data line between AGC: and Scale;; "DSP Gain:" removed from the settings pane.
- Display > **Freq Font** (Original / Nixie / VFD / Dot Matrix), drawn on `#freqCanvas` by `drawFreqStyled()`, following the shared `pi_freq_font` state.
- **Web remote: the S-meter's "-XXX.X dBm" reading was cut off on the right.** `marginRight` changed from 70 to 100, the same fix the phone got on 2026-09-27.

2026-10-02 — Menu highlights

- Menu/Display items mark the current / last-selected choice (green check) in groups of two or more. Firmware items follow the T41's current menu item (STATUS) and clicks; the Pi-only flyouts follow `pi_spectrum_fill` / `pi_waterfall_3d` / `pi_waterfall_speed` / `new_pi_brightness`.
- The new Network page is on the Pi screen only (see `CHANGELOG_pidisplay.md`).

2026-10-02 — Toolbar dropdowns moved left

- `#freqBBlock` min-width 300px to 270px: dropdowns sit about 8 spaces further left.

2026-10-02 — Display gets its own dropdown

- **Display has its own toolbar dropdown**, between Menu and Vol, and is no longer a category under Menu. Its items are the same grouped list (Spectrum floor, scale, smoothing, decay, trace smoothing, Spectrum shadow, Spectrum fill, Brightness, PI Spectrum), now one level shallower. Picks still send Display's real primary index, so nothing changes on the Teensy.
- Pi: `_OWN_DROPDOWN_CATEGORIES = ("Display",)`, plus a toolbar entry "Display" after "Menu". Web: `OWN_DROPDOWN_CATEGORIES`, and

`buildMenu(container, {label, id, only})` builds both dropdowns. The `#menuWrap/#menuBtn/#menuPanel` styles became classes, so the two share styling, and opening one closes the other.

2026-10-02 — Display menu regrouped; 3D peaks blended

- **Menu > Display regrouped** into sub-menus. The T41's own menu numbers are unchanged; this only changes how the menu is shown:
 - Spectrum floor > Auto spectrum floor / Manual spectrum floor / Spectrum floor
 - Spectrum scale, Spectrum smoothing, Spectrum decay, Spectrum trace smoothing
 - Spectrum shadow > On / Off
 - Spectrum fill > On / Off
 - Brightness > 25-100%
 - PI Spectrum > PI Spectrum Fill (On/Off) / PI Waterfall (2D/3D) / PI Waterfall Speed Defined once as `_MENU_LAYOUT` (display.py) / `MENU_LAYOUT` (web); every number in it is the item's real `DisplayOptions[]` index. Greyed-out items still work as before.
- **3D waterfall peaks blended:** the Pi now uses 12 colour bands (was 6), sampled from the same colour ramp as the web gradient. Each new row is 1-2-1 smoothed across neighbouring columns (valleys between nearby peaks fill in; peaks keep their height) and mixed 35% with the previous row (`_WF3D_TIME_BLEND`), so ridges look continuous. Row outlines are now dark navy instead of bright blue. The Pi takes about 20 ms per frame in testing.
- Menu built by a generic nested builder (`addCascade/addLeaf/buildNodes`).

2026-10-02 — Waterfall speed setting

- New **Menu > Display > "PI Waterfall Speed"** flyout: **Fast** (one waterfall row per spectrum frame, about 15 rows/s, as before), **Medium** (1 per 2 frames), **Slow** (1 per 4) and **Very slow** (1 per 8). It applies to both the 2D and 3D waterfalls. The frames in between are peak-held into the row, so a short signal is never skipped, and the spectrum trace above stays real-time.
- Shared state across the Pi and every web tab, like PI Waterfall / PI Spectrum Fill. It starts at Fast. Pi: `set_waterfall_speed()`, `set_pi_waterfall_speed()`, web command `pi_waterfall_speed_set {divisor}`, broadcast `pi_waterfall_speed`.

2026-10-02 — 3D waterfall option

- Menu > Display > **"PI Waterfall"** flyout (2D / 3D), sending `pi_waterfall_3d_set`; follows `pi_waterfall_3d` broadcasts.
- New `drawWaterfall3D()` / `wf3dNormalize()`: 24 rows x 256 columns, one gradient fill per row, painted back to front, with a centre marker. Rows are collected every frame, so switching to 3D shows history straight away.

2026-10-01 — Equalizer: Exit button, Edit RX / Edit TX buttons

- "Editing: (switch)" toggle replaced by **Edit RX / Edit TX** buttons. They send `eq_select_side` (the same command a bar click uses), and the side being edited is lit green and reads "Editing ...".
- New **Exit** button (button 14, firmware V1.0.27+): undoes this visit's changes and returns Home without saving. "Save & Exit" unchanged.

2026-10-01 — Audio Equalizer screen added; "TI:" renamed "CT:"

- Full-screen **Audio Equalizer** overlay (Audio Options → Adjust Equalizers), driven by firmware V1.0.26 telemetry (PiLink 0x09): Receive and Transmit panes, 14 bars each with frequency labels, selected-cell readout and increment.
- Controls: Editing RX/TX (button 15), Cell ◀▶ (Volume encoder), Level – + (Filter encoder), Incr 1/10 (button 16), Save & Exit (Home). Click/tap a bar to select it directly, including on the other side.
- While the overlay is up: mouse wheel and Up/Down/PgUp/PgDn change level, Left/Right move the cell.
- Readout label **"TI:" renamed "CT:"**.

2026-10-01 — I2C BIT test screen added; several menu items greyed out

Requires the Teensy to be on firmware V1.0.25 (new `PILINK_MSG_BIT_RESULTS` downlink message) for the new screen to show live data; the menu-greying part needs no firmware change at all.

New `#bitOverlay/#bitPanel` full-screen takeover mirrors the Teensy's own I2C BIT test screen (Menu > Diagnostics > "I2C BIT test") - one PASS/FAIL row per board (front panel, BPF, RF, RF SI5351, K9HZ LPF) built from a new `onBitResults()` handler for the `"bit_results"` WebSocket message, plus a single "Home" button sending `sendButton(17)` (Config.h `HOME_SCREEN`) - the only control that screen's own `Loop.cpp` switch handles. Entering the screen needed no new command at all: "I2C BIT test" was already a normal `menu_select` leaf in this page's own `T41_MENU` mirror.

Also new: a `MENU_DISABLED_ITEMS` set greys out (and removes the click handler from) every Calibration item except Receive IQ, plus Diagnostics > Restore receive EQ/Buffer print and EEPROM > Print Data to Serial - items with no web-side screen/action behind them yet. A new `.submenu-item.disabled` CSS class renders the grey, same `buildMenu()` leaf-building loop that places every other menu item.

2026-10-01 — Receive IQ calibration: "Stop"/"Exit" buttons + instructions

Brings this page's calibration screen to parity with the Pi's own screen, at the user's request. New `#rxiqCalStop` ("Stop", orange outline) and a plain "Exit" button added to `#rxiqCalControls`, alongside the existing Band/Auto/Save & Exit. Both send the same repurposed virtual-button ids the Pi sends: `sendButton(13)` (`Config.h` `DECODER_TOGGLE`) for Stop, `sendButton(14)` (`Config.h` `DPE`) for Exit - new "stop"/"exit_nosave" cases in the `[data-rxiq]` click handler, no new WebSocket command shapes. New `#rxiqCalInstructions` block below the table, same text as the Pi screen's instructions ("Adjust until Delta > 60 dB... Stop aborts Auto Tune early. Exit leaves without saving. Save & Exit when done."). No reflash needed if already on V1.0.24 (that firmware shipped the button handling); hard-refresh the browser.

2026-09-30 — Receive IQ calibration screen added

New full-screen overlay (`#rxiqCalOverlay`) mirroring the T41's Menu > Calibration > Receive IQ screen, shown/hidden by a new `"rxiq_cal"` WebSocket message (`onRxiqCal()`) - sent whenever `uiSM.state_id == UISM_StateId_CALIBRATE_RX_IQ` on the Teensy. **Requires reflashing the Teensy to V1.0.23** - see `CHANGELOG_firmware.md`.

- Readout: current band, Delta (dB), Amp, Phase, and step-size Increment (0.010/0.001).
- 13-row all-bands table (Band/Amp/Phase/Delta), current band highlighted.
- Full control, not just a mirror: Amp±/Phase± (Volume/Filter encoder ticks), Increment toggle, Auto-tune, Band◀▶, and Save & Exit buttons, all sending the same generic `{cmd:"button",id}/ {cmd:"encoder",encoder,increase}` shapes the toolbar and Menu already use (`sendButton()/sendEncoder()`) - no backend changes needed on the web side for this feature.
- **Not included:** the live two-sideband spectrum mini-plot the T41/Pi screen shows - deliberately left out for this first pass.
- Scope: Receive IQ only, at the user's request - Frequency/Transmit IQ/ Power calibration screens are a future pass.

2026-09-30 — SWR/PWR readout added, “(USB)” demod suffix bug fixed

User report: "Both remotes web and phone are missing the demod reading like the PI screen 20M / DATA (USB) and also the SWR 1.0 PWR 0.0 label/readings." Two separate issues:

1. **SWR/PWR never built on this page** - `status.swr/ status.forward_power_w` were already arriving on every `"status"` message, just never rendered anywhere. New `#swrText` element added to `statusBlock`, right under REC/XMT (same relative position as the Pi's own screen's SWR/PWR pane, which sits flush against `txrx_status`). `onStatus()` updates it with `SWR {swr.toFixed(1)} PWR {forward_power_w.toFixed(1)} W`, red on TX / white on RX - mirrors `display.py`'s `set_swr()` exactly.
2. **“(USB)” suffix silently dropped** - `MODULATION_NAMES` had 0: null instead of 0: "USB", a bug present since this page was first built (modulation 0 is the common USB default, so this showed up immediately once noticed). `display.py`'s own `_MODULATION_NAMES` had the right value the whole time - only this page's separate JS copy of the table was wrong. Fixed to 0: "USB".

No reflash needed - hard-refresh the browser.

2026-09-30 — Independent “PI Spectrum Fill” toggle (supersedes mirroring below)

Follow-up to the mirroring entry directly below: the user tried it, then asked to tune the two independently ("Need to be able to tune it on/off separately. T41 On/Off and PI/Remotes On/Off"). New Menu > Display > "PI Spectrum Fill" > On/Off 3rd-level flyout (same pattern as "Brightness"), sending `{cmd: "pi_spectrum_fill_set", enabled}`. New `state.piSpectrumFillEnabled` (default `true`) updated by a new `onPiSpectrumFill(enabled)` handler for a new `"pi_spectrum_fill"` message type (pushed by `radio_ui.py/ web_relay.py`, cached server-side so a newly-opened tab is synced right after `login_ok`). `drawSpectrumStatic()`'s fill gate switched from `state.settings.spectrum_fill_trace_enabled` to `state.piSpectrumFillEnabled` - this page's fill is now fully decoupled from the T41's own "Spectrum fill on/off" setting, which continues to affect only the T41's own screen. **No reflash needed** - see `CHANGELOG_pidisplay.md` for the full cross-file writeup.

2026-09-30 — Mirrors T41's “Spectrum fill on/off” setting (superseded, see above)

`drawSpectrumStatic()` now reads `state.settings.spectrum_fill_trace_enabled` (defaults to `true`) and skips `drawTraceFill()` when it's false, so this page's spectrum trace goes outline-only to match whatever the T41 itself is set to (Menu > Display > "Spectrum fill on/off", firmware V1.0.21/22) - added at the user's request, purely cosmetic. The outline (`drawTrace()`) is drawn either way. **Requires reflashing the Teensy to V1.0.22** (new `PiLink` field) in addition to the usual re-sync/hard-refresh - see `CHANGELOG_pidisplay.md` for the full cross-file writeup.

2026-09-30 — Choppy-audio bug found and fixed (mis-clocked playback rate)

First tried raising `AUDIO_LEAD_S` (the playback jitter buffer) from 0.15s to 0.4s, at the user's report: "Sound on remotes is choppy" - the theory being WiFi timing wobble was outrunning the buffer. The user reported that made it "even worst", which is backwards for a jitter problem: a bigger buffer should only ever help or do nothing against pure network jitter. That pointed at a different kind of bug - drift, not jitter - and `onAudioPcm()` had one: `audioCtx.createBuffer(1, frameCount, audioCtx.sampleRate)` labeled every chunk with the *AudioContext's* sample rate, not the rate the PCM was actually captured at (`AUDIO_SAMPLE_RATE`, 44.1kHz). `ensureAudioContext()` only *requests* 44.1kHz when creating the context - browsers are free to hand back their own hardware-native rate instead (48kHz is a common default), and when that happens every chunk was mislabeled: played back pitch/speed-shifted, and worse, `buffer.duration` (used to advance `audioNextPlayTime`) was computed from the wrong rate, so the playback schedule slowly drifted out of sync with real chunk-arrival timing and had to periodically resync (an audible glitch each time). A bigger `AUDIO_LEAD_S` just let more of that drift accumulate before each resync - exactly the "even worse" symptom reported. Fixed by labeling the buffer with its true, fixed source rate (`AUDIO_SAMPLE_RATE`) instead - `AudioBufferSourceNode` resamples an `AudioBuffer` to the context's actual rate automatically on playback, so this is correct regardless of

what rate the context landed on. `AUDIO_LEAD_S` reverted back to 0.15s once the real bug was fixed.

2026-09-30 — Spectrum pane shrunk, New Line 2 strip moved above it

At the user's request: "reduce the spectrum's pane height, there is a lot of wasted space as the signals never reach that high. Then move the readouts that are currently in the spectrum display to above the top line for the spectrum display." - `specGeometry()`'s `traceH` share of the canvas's remaining height (after the ruler gap/labelH) cut from 40% to 25% - the trace box was taking far more height than the live signal ever used; the freed space now goes to the waterfall. - The Vol:/AGC:/Scale:/Floor:/Zoom:/TI:/FT:/Noise: strip moved OUT of the spectrum canvas entirely - was `drawNewLine2()`, drawn on the canvas across the inside top of the trace box; now a real DOM row, `#newLine2Row` (plain flex, `justify-content: space-between, 14px`), positioned above `#mainRow`. `#mainRow` is `flex:1` so it automatically gives up this row's height to `#specCanvas` below it - no extra resize logic needed. Updated via the new `updateNewLine2()` (plain DOM `textContent`, called from `onSettings()`) instead of being redrawn on every spectrum frame.

2026-09-30 — "Brightness" combined into one flyout, "Scale:" centered

Two follow-up changes, same day: - The 4 flat "Brightness NN%" entries (see the entry below) became a single "Brightness" entry with its own nested percentage flyout, at the user's request. `buildMenu()` special-cases "Display" > "Brightness" into a 3rd-level cascade reusing the same `.menu-item/.submenu/.submenu-open` CSS as the top-level category flyout - clearing is now scoped to each flyout's own immediate children (`Array.from(...children)` instead of `panel.querySelectorAll(...)`) so opening a nested flyout doesn't also close its parent. Sends a new `"brightness_set"` command instead of `"menu_select"` - see `CHANGELOG_pidisplay.md` for the full writeup, including the corresponding `radio_ui.py` changes. - Scale: moved right and centered between AGC: and Floor:, still with full room for "Scale: XX.XdB" - the previous overlap fix (below) left it crowded against AGC:. New fractions: `0.00/0.09/0.23/0.44/0.58/0.68/0.78/0.89`.

2026-09-30 — "Brightness" added to the Display menu

`T41_MENU`'s "Display" category gained 4 new entries — "Brightness 25%", "50%", "75%", "100%" — a Pi-only backlight control for the Raspberry Pi Touch Display 2's own screen, at the user's request. No new websocket command needed: these are sent as an ordinary `menu_select` message, same as any other menu item, and intercepted server-side in `radio_ui.py` before anything is forwarded to the Teensy — see `CHANGELOG_pidisplay.md` for the full writeup. **This page only** for the menu entries themselves; the actual brightness write happens on the Pi regardless of which surface (Pi screen, this page, or the phone page) sent the selection.

2026-09-30 — "Scale:"/"Floor:" columns no longer overlap

Scale:'s widest value ("12.5dB") was overwriting part of Floor:'s label, reported right after Floor: shipped (see the entry below). `drawNewLine2()`'s column fractions recomputed by actual character-count budget per column (label + widest possible value) instead of even spacing: `0.00/0.10/0.18/0.35/0.49/0.60/0.73/0.86`. Same fix applied to `remote_mobile.html`.

2026-09-30 — "Floor:" reading added to the New Line 2 strip

New column in `drawNewLine2()`, right after `Scale:`, showing the current Menu > Display > "Spectrum floor" setting — "Auto" while "Auto spectrum floor" is on, otherwise the signed manual value (a raw screen-pixel baseline shift, -100..100, not dB). **Not web-remote-only** - needs the same-date `PiLink.h/PiLink.cpp` firmware change (see `CHANGELOG_firmware.md` V1.0.18) reflashed before `s.spectrum_floor/s.spectrum_floor_auto` are real data. Row fractions recomputed for 8 columns: `0.00/0.12/0.25/0.37/0.50/0.62/0.75/0.87`.

2026-09-30 — "RX DSP Gain" added to the RF Options menu

`T41_MENU`'s "RF Options" entry gained a 5th item, "RX DSP Gain", matching the newly-un-hidden firmware menu entry (see `CHANGELOG_firmware.md` V1.0.17) and `display.py`'s same-date `T41_MENU` update — index order must match the firmware's `RFSet[]` 1:1 since indices, not labels, are sent over `{cmd: "menu_select"}`. **Needs a Teensy reflash** before the entry does anything on the real radio; see `CHANGELOG_pidisplay.md` for the companion `_MENU_REAL_LENGTH_OVERRIDE` wraparound fix in `radio_ui.py`.

2026-09-29 — Audio Filter tick labels enlarged

The "0k/1k/.../6k" labels along the bottom of the Audio Filter pane bumped from 8px to 11px, at the user's request (also applied to the Pi's own screen). **This page only** — the user confirmed the phone page (`remote_mobile.html`) is fine as-is at the current size, so it's unchanged. Still comfortably inside the bottom margin already reserved below the pane's content box.

2026-09-27 — Audio Filter ruler labels missing their leading/trailing character

User report: "0K" showed only "K", and the rightmost tick showed only its digit, not "NK". Root cause: `drawAudioSpectrumStatic()`'s 1kHz-tick ruler always drew each label `textAlign: "center"` - the leftmost tick sits at `x=0` and the rightmost can land at/near the canvas's own right edge, so centering there draws half the label's width off-canvas, which the canvas silently clips. Now switches to `"left"` within 8px of the left edge and `"right"` within 8px of the right edge, `"center"` everywhere in between. Same fix applied to `remote_mobile.html`.

2026-09-27 — “Power: 20 W” changed to “Power: 20W”

No space before the unit anymore, at the user’s request to “move the W over”. Also applied to `remote_mobile.html` and the Pi’s own Settings pane (`display.py`) for consistency. Cosmetic only.

2026-09-27 — spectrum trace fill matched to the waterfall’s own colors

Matches the same fix made on the Pi’s own screen (`display.py`), at the user’s follow-up request (“Yes match for the remote.html and remote_mobile.html”). The trace’s fill-under-the-curve used to fade from green to yellow at a fixed 1/3 of the pane height and to red at 1/2 — close to, but not exactly, where the waterfall’s own `WATERFALL_GRADIENT_HEX` table shows those same colors at the same amplitude (the trace’s y-position is a direct function of the same 0-255 byte the waterfall colors, so a height fraction here already is an amplitude fraction). - `SPECTRUM_COLOR_BREAK1/SPECTRUM_COLOR_BREAK2` changed from 1/3 and 1/2 to 32/116 and 38/116 — the exact fractions where the waterfall’s own gradient table stops being pure green and reaches pure yellow. - New `SPECTRUM_COLOR_BREAK3` (64/116, where the table reaches pure red) added as an extra `createLinearGradient()` color stop, so the fill clamps to solid red there instead of only reaching full red at the very top of the pane.

2026-09-26 — waterfall disappeared after the ruler fix

(Shared fix, made the same day as the phone page’s V1.0.13.) The previous day’s ruler fix traded one bug for another: the frequency numbers became visible but the waterfall display disappeared. Root cause — `specGeometry()`’s `traceH` was set to `h - gap - labelH`, i.e. 100% of whatever was left over after the ruler’s fixed-pixel band, leaving nothing for the waterfall. - The leftover space now splits between the trace and the waterfall (`traceH` = ~40% of what’s left after `gap/labelH`, waterfall gets the rest) instead of handing all of it to the trace.

2026-09-26 — frequency-scale ruler fixed for real this time

(Shared fix, made the same day as the phone page’s V1.0.12.) The ruler-into-waterfall problem (frequency labels spilling into the waterfall) was traced to `specGeometry()`: the ruler’s band height (`gap/labelH`) was a proportional *fraction* of the canvas height, but `drawFreqScale()`’s own tick-mark/label pixel offsets are fixed constants that never scaled with it. At smaller canvas heights the proportional band shrank below what those fixed offsets need. - `gap/labelH` are now fixed pixel values (5px/20px) regardless of canvas height.

2026-09-26 — REC shown in green

(Shared fix, made the same day as the phone page’s V1.0.9.) `txrxText`’s color when receiving was plain white; now `#00cc44`. XMT stays red, unchanged — a plain color preference, not phone-specific.

2026-09-26 — remaining toolbar dropdown labels + S-meter clipping fixed

(Shared fix, made the same day as the phone page’s V1.0.7.) - **Band/Mode/Demod/Filter labels**: now show a fixed “Band”/“Mode”/“Demod”/“Filter” label, centered, alongside the five fixed earlier the same day (below). - **Vol and Home centered too**: generalized `makeDropdown()/setSelectSilently()` so every toolbar dropdown uses the same invisible-select-over-a-centered-label-span technique — a native `<select>`’s own centering is unreliable across browsers (notably ignored on iOS Safari). - **S-meter tick-label/“S”/“+20dB” text tops cut off**: baseline moved down 6px so an 11px font’s ascenders have real clearance above the canvas edge.

2026-09-26 — dropdown labels overwritten by current value fixed

(Shared fix, made the same day as the phone page’s V1.0.6.) The Course Inc/Fine Inc/Zoom/Noise/Decode toolbar buttons were displaying their *current* selected value instead of a fixed label, since a plain `<select>` always shows its selected option’s own text. - New `.fixedLabelWrap/.fixedLabelText/.fixedLabelSelect` CSS: the real `<select>` stays fully functional but is stretched invisible over a plain, centered label span that never changes text. - `makeDropdown()` gained an optional `fixedLabel` parameter, used for Course/Fine/Zoom/Noise/Decode (renamed from “Course Inc”/“Fine Inc” to the shorter “Course”/“Fine” at the same time).

2026-09-26 — split into a separate desktop-only page

The `@media (pointer: coarse)` rules this page had grown for phone support (jog sliders, reordered status block, 4-column toolbar grid, shortened lower panes — see the mobile page’s own changelog for that history) turned out to also match the user’s own laptop, reflowing the normal desktop screen by mistake. Media-query pointer detection isn’t a reliable phone-vs-laptop signal on all hardware, so the two experiences became two separate files: - `web/remote.html` **reverted to a plain desktop/laptop page** with no pointer-detection dependency at all — every phone-only rule was removed. Desktop users see exactly the pre-phone layout again after a hard refresh. - Everything phone-specific moved into the new `web/remote_mobile.html` (its own changelog covers that page from here on). - The S-meter responsive-canvas fix (below) is the one piece kept in **both** files, since it’s a universal correctness fix, not a phone-only layout choice. - `web_relay.py` gained a `/mobile` route serving the new file; `/`, `/index.html`, `/remote.html` are unchanged and still serve this page.

2026-09-26 — S-meter bar fixed to actually fit the screen

`#smeterCanvas` never had a resize handler, so its `width="900"` HTML attribute rendered literally, overflowing any viewport narrower than 900px. New `resizeSmeterCanvas()` sizes it to its container's actual rendered width on load and on window resize. Not phone-specific — this fix applies on desktop too, though the bug was only ever visible on a narrow screen (kept in both files after the split above).

Same-day work later moved to the phone page (not part of this file today)

Touch jog sliders for Tune/Fine Tune/Volume/Filter, the hidden Wheel indicator, the reordered `#statusBlock`, and the 4-column toolbar grid were all added directly to this file first (gated behind `@media (pointer: coarse)`), then reverted out of it entirely and moved into `web/remote_mobile.html` at the split above. They are not part of `remote.html` as it stands today — see that page's own changelog for their full history.

2026-09-26 (current / unreleased) — Settings & top-strip parity with the Pi's own screen

- **Vol:/AGC:/Zoom:/TI:/FT:/Noise: moved from the Settings pane into an always-visible strip across the inside top of the spectrum display**, matching the Pi's own screen. New `drawNewLine2()` draws the same six fields as an overlay across the top of the spectrum trace box, redrawn on every SETTINGS message. Removed from the Settings pane (now redundant): the old multiplexed top Vol/AGC row and the Tune Inc:/FT Inc:/Zoom:/Noise: fields — the remaining 8 fields reflow into 4 rows.
- **Filter** (filter bandwidth) now shows in the Audio Filter pane's own upper-right corner, drawn last so it paints on top of the live bars/markers.
- **Power** added to the Settings pane for parity with the Pi's own screen, since that value already reaches the browser. (PI Temp was *not* added — it's a local Pi CPU reading with no PiLink field, so there's nothing for a browser on another machine to show.)
- The Vol/AGC/Zoom/TI/FT/Noise strip and the Filter: readout were bumped from 13px/11px to 14px to match the Settings pane's own font size.

V1.0.14 (2026-09-25) — original build of the LAN web remote

The initial version of this page, added the same day as the underlying `PiDisplay/web_relay.py` WebSocket relay. - **LAN-only browser remote control** (mouse + USB rotary encoder): mirrors the Pi's VFO/toolbar/band-mode/TX-RX/S-meter/spectrum+waterfall+ frequency-scale panes and the Menu parameter-adjustment overlay, mouse-only. Gated by a passkey (`config.WEB_PASSKEY`). - **VFO/fine-tune driven from a mouse wheel** over the frequency/spectrum area (Shift = fine), or from a USB rotary encoder emitting arrow/page keys, regardless of mouse position. Required a Teensy reflash for the new `PILINK_ENCODER_TUNE/PILINK_ENCODER_FINETUNE` virtual-encoder ids. - **Audio Spectrum pane, Settings pane, and a real cascading Menu**: live magenta audio-FFT bars plus passband/CW-filter markers; a Settings pane mirroring the Pi's Vol/AGC/Tune Inc/FT Inc/Atten/AutoNotch/Noise/DSP Gain/Antenna/Zoom/Rate/Temp/Load/version fields; the Menu control rebuilt from a flat `<select>` into a nested/cascading popup closer to the Pi's own Tkinter cascade widget. - **BW (Filter) dropdown, wheel-target fix, and a wheel deselect control**: a "BW" dropdown (Hi Cut/Lo Cut) alongside the existing Vol dropdown; fixed the mouse wheel being hardcoded to always emulate Volume; picking from either dropdown now "claims" the wheel for that knob, with a clickable "Wheel: Tune only / Volume / Filter" status pill to release it. - **Live RX audio, plus a Mute button**: plays back 44.1kHz mono PCM streamed from the Pi as raw binary WebSocket frames, scheduled as back-to-back `AudioBuffers`. A red "Mute" toolbar button is local to the browser tab only — it doesn't stop the Pi from sending audio or affect the Pi's own speaker. - **Blue filter-bandwidth highlight on the Spectrum pane**: a solid rect spanning the current receive passband, centered on the fixed screen-center tuning marker, drawn before the live trace so it only shows through above the curve — matching the real hardware's own visual effect.