**ICM (Interface Control Module) — Design Brief (Updated)**

**0) Role & Placement**

* **Role**: The ICM is the head-end brain for provisioning, configuration UI (AP→STA), node pairing/mapping, event logging, sequence policy, and diagnostics. It now also manages **local user feedback (buzzer)** and **environmental sensing (temperature sensor)**.
* **Placement**: Indoors at the home side. Interfaces to the 48 V trunk only indirectly (via the head-end PSM). All field modules are wireless along the driveway.

**1) High-level Block Diagram (text)**

* 5 V IN (from PSM-HE) → Input filtering & protection → Primary 3.3 V buck → Clean 3.3 V LDO rail for MCU/RF/logic
* ESP32-S3 (with u.FL external antenna)
* USB-C (CDC) for service logs & firmware loading (factory)
* User I/O: Reset/Factory button; BIT (green) & FAULT (red) LEDs
* **Buzzer for audible feedback** (short beep = action confirm; long beep = error)
* **Temperature sensor** input (OneWire DS18B20 type, with pull-up) for monitoring enclosure ambient/thermal safety
* Optional inputs: Day/Night digital input; dry-contact header (future)
* Secure storage: Internal flash (NVS) + secure boot & flash encryption
* Debug access: UART/JTAG (factory only; locked in field)

**2) Core Electronics**

**2.1 MCU/RF**

* SoC: ESP32-S3-WROOM/N8R2
* Clocking: 40 MHz crystal; optional 32.768 kHz footprint for deep sleep
* Antenna: u.FL to 2.4 GHz external whip, with RF keep-out

**2.2 Power Path**

* 5 V input (4.75–5.5 V, ≥1 A) with fuse, TVS, reverse protection, and Pi-filter
* Primary 3.3 V buck → Secondary low-noise LDO for RF section
* Brown-out detection + optional 5 V ADC telemetry

**2.3 User & Service Interfaces**

* USB-C with ESD protection
* Buttons/LEDs: RESET/FACTORY, BIT LED (green), FAULT LED (red)
* **Buzzer (GPIO-controlled, active-high)** for audible UI feedback and alerts
* **Temperature sensor (OneWire DS18B20, GPIO with pull-up)** for monitoring ambient or board temperature
* Optional day/night input (RC + Schmitt + TVS)

**2.4 Storage & Identification**

* Flash partitioned: bootloader, ota\_0, ota\_1, nvs (encrypted), keys, coredump
* Device IDs/QR label with Serial, MAC, HW rev, FW rev, Key slot

**3) Cybersecurity & Tamper Hardening**

* Secure Boot V2, Flash Encryption, JTAG disabled post-provisioning
* NVS secrets encrypted, factory reset wipes config/keys
* OTA A/B update with rollback, golden image fallback
* Access creds persisted securely in NVS

**4) Reliability & Robustness**

* All semiconductors derated ≤70% of max stress
* TVS + polyfuse at input; watchdogs always enabled
* Thermal headroom maintained; coredump + safe boot recovery
* ESD/EFT immunity sized for IEC 61000-4-2/-4/-5

**5) EMC/Safety Targets**

* IEC/EN 62368-1 (safety, SELV)
* CISPR/EN 55032 Class B emissions
* EN 55035 immunity (ESD/EFT/surge)
* IEC 60068-2 environment (–15 °C…+55 °C indoor, humid garage)

**6) Firmware Architecture**

* FreeRTOS tasks:
  + HTTP/UI + captive portal provisioning
  + ESP-NOW key mgmt + peer registry
  + Sequencer policy & schedule hooks
  + Heartbeat (BIT LED), watchdog kicker
  + Logger (ring buffer to UI/USB)
  + **BuzzerManager** task for audible feedback control
  + **TempSensor task** polling DS18B20 and reporting into monitoring API
* NVS schema versioned and checksummed
* Crash safety: coredump + boot loop detector

**7) Detailed Interface & Pin Budgeting**

* Power: 5 V IN terminal → 3.3 V
* USB-C: D+/D-, CC1/CC2, shield to chassis via RC
* LEDs: BIT, FAULT (pins in Config.h)
* Buttons: RESET/FACTORY (pins in Config.h)
* Day/Night IN (GPIO, optional)
* **Buzzer: Pin + polarity stored in Config.h (BUZZER\_PIN\_KEY, BUZZER\_ACTIVE\_HIGH\_KEY)**
* **Temperature sensor: Pin + pull-up stored in Config.h (TEMP\_SENSOR\_PIN\_KEY, TEMP\_SENSOR\_PULLUP\_KEY)**
* Factory pads: UART/JTAG pads, not populated
* RF: u.FL → SMA whip

**8) Manufacturing Test Plan**

* ICT & power rails test
* Bootloader signature check
* USB CDC enumeration
* RF beacon + ESP-NOW self-test
* NVS read/write/erase test (encrypted)
* LED, button, **buzzer tone test**, **temperature sensor readback**
* Secure boot eFuses burned + QR print

**9) Field Diagnostics**

* LEDs: BIT blink healthy, FAULT solid error
* **Buzzer patterns**: short beep = success, long beep = failure, repeating = alarm
* USB console: status/log export
* Recovery: Long press factory reset wipes NVS + returns to AP

**10) Pin & Config Management**

All pin assignments (LEDs, buttons, buzzer, temperature sensor, UART, SD, etc.) are **centrally defined in Config.h** under hardware mapping keys. This allows field hardware revisions or board spins to re-map I/O via config updates without altering core firmware.

**TL;DR**

The ICM is still the same secure, long-life, hard-to-kill indoor controller — now enhanced with **audible feedback via a buzzer** and **thermal monitoring via a temperature sensor**, both with their pins configurable in software through