ELEC 291 Project 1 – Reflow Oven Controller Current Progress & Next Actions (Keypad + Thermocouple + FSM)

Date: February 08, 2026 (America/Vancouver)

1. Project objective (system-level)

Build a microcontroller-based controller that measures oven temperature using a K-type thermocouple with cold-junction compensation, drives a 1500W toaster oven through an SSR (e.g., PWM power control), provides a user interface (keypad/LCD/7-seg), streams temperature to a PC at ~1 sample/sec for real-time plotting, and enforces safety abort conditions.

2. What the current codebase already has (confirmed from the provided snippets)

- **Keypad input module** (4x4 scan, 25 ms debounce, digit shift-left to append, shift-right as backspace, store-variable by field selection).
- **Thermocouple/ADC pipeline**: ADC read → math32 conversion to temperature → hex2bcd → stored as BCD for display/serial output.
- **Serial output**: outputs temperature as ASCII digits + CR/LF; suitable for 1 Hz plotting.
- **FSM skeleton**: Settings → Ramp to Soak → Hold Soak → Ramp to Reflow → Hold Reflow → Shutdown, driven by a 1-second flag from Timer2.
- **Safety checks**: abort if temperature stays below 50°C at 60 seconds; abort if temperature exceeds 240°C.
- **SSR control hook**: OVEN_POWER pin is controlled; placeholder PWM logic exists in Timer2 ISR for Hold states.

3. Gaps / issues that will block a reliable demo (highest priority first)

- **Keypad parameters are stored as BCD nibbles, but control logic needs binary integers.** Soak/Reflow times and temperatures must be committed into Soak_Temp_Bin / Soak_Time_Bin / Reflow_Temp_Bin / Reflow_Time_Bin with consistent units (°C and seconds).
- **Timing model is inconsistent.** Timer2 ISR currently updates only a single-byte BCD Time_Elapsed and does not maintain a true 32-bit Time_Elapsed_Bin. Hold-state comparisons are therefore fragile and can fail for values beyond simple ranges.
- **Temperature representation is BCD-only in many places.** Control comparisons repeatedly convert BCD→hex every second. Cleaner design is to keep a binary temperature (Current_Temp_Bin) updated by Read_ADC and keep BCD only for display/serial.
- **PWM is a placeholder.** Current ISR pattern effectively produces a fixed, coarse duty cycle in Hold states and does not regulate to target. For a stable soak/reflow hold, PWM must be unified and driven by either a fixed duty or a simple feedback rule.
- **Mixed ownership of OVEN_POWER.** Some states set OVEN_POWER in main loop while ISR also toggles it. This can cause conflicts; choose one control path (recommended: ISR PWM only).

4. Recommended next actions (concrete tasks)

Task A — Parameter commit (Keypad → binary)

- Define a clear 'commit' event (e.g., keypad 'D' or '#') and a field selector (current SW0–SW3 works for bring-up).
- On commit: Store_BCD(field_BCD) → bcd2hex → Store_X_var(field_Bin).
- Apply range clamps: temperature <= 240°C; enforce sane min/max for soak/reflow time.

Task B — Fix elapsed-time counter (1 Hz, binary)

- In Timer2 ISR when 1000 ms elapses: increment Time_Elapsed_Bin (32-bit) by 1.
- On state transitions: clear Time_Elapsed_Bin (all 4 bytes).
- Replace hold comparisons with x_gteq_y against Soak_Time_Bin / Reflow_Time_Bin.

Task C — Maintain binary temperature for control

- Modify Read_ADC to store the computed temperature into Current_Temp_Bin (x) before hex2bcd.
- Keep curr_temp (BCD) only for 7-seg/LCD/serial formatting.
- Update ramp/abort logic to compare using Current_Temp_Bin directly (no repeated bcd2hex conversions).

Task D — Make PWM demo-stable, then improve

- Minimum: Ramp states duty=100% until target threshold; Hold states use fixed duty (e.g., 30–60%).
- Better: compute error = target - current and use simple piecewise duty (no PID needed for demo).
- Unify ownership: ISR writes OVEN_POWER; main loop writes desired duty variable only.

5. Minimal demo checklist (what to verify end-to-end)

- Keypad: enter each of the 4 parameters; commit; verify binary variables change (via temporary 7-seg/LCD debug view).
- Timer: elapsed seconds increments correctly and resets on each state transition.
- Thermocouple: temperature reading matches multimeter within ±3°C across 25–240°C range (record validation data).
- FSM: transitions occur at correct temperature/time thresholds.
- Serial plot: exactly ~1 sample/sec during the whole run.
- Safety: unplug thermocouple and confirm abort at 60 s; force over-temp (if safe) and confirm shutdown.