

Aura003 Firmware — Development Log

ATtiny1624 · megaTinyCore · Heated Docking Station

Project: Aura003

Exchange 1

PROMPT

Add the following to the present program.

On pin PB0 there is a voltage divider between 12V and GND. From 12V there is 15k and from GND 5k. This is to sense whether there is 12V present or not.

Create and add a detection circuit such that the heater only turns on when there is 12V present.

Is there any more information necessary to make the above program?

RESPONSE

Confirmed that VCC is 5V and the voltage divider gives 3.0V at PB0 when 12V is present (ADC $\approx$ 614). Threshold set at 2.0V (ADC $\approx$ 409) for noise margin.

Key design decisions:

- 12V absent $\rightarrow$ heater gate OFF, state machine resets to HEATING
- 12V present $\rightarrow$ run full HEATING $\rightarrow$ MAINTAINING $\rightarrow$ COOLING cycle
- Resetting to HEATING on reconnect is safe since temperature drops when disconnected
- Hold timer replaced with accumulator pattern to support pause/resume
- detect12V() uses single raw ADC read — no averaging needed for slow rail
- PIN_PB0 on ATtiny1624 with megaTinyCore maps to ADC correctly as-is

Exchange 2

PROMPT

VCC is indeed 5V. The ATtiny is being powered from a battery. When it is connected to the 12V it charges the battery and then turns on the heater as well. When it gets disconnected the gate should be off. When connected detect whether there is 12V and then turn on the heater and maintain like in the above program.

RESPONSE

Simplified logic implemented:

- 12V absent $\rightarrow$ heater gate OFF, state resets to HEATING
- 12V present $\rightarrow$ run full HEATING $\rightarrow$ MAINTAINING $\rightarrow$ COOLING cycle

No timer pausing needed. Resetting to HEATING on reconnect makes sense because if the device was disconnected mid-cycle, the temperature would have dropped anyway. Added v12WasAbsent static flag to prevent serial spam and trigger a single clean "restored" message on reconnect.

Exchange 3

PROMPT

Do the following when 12V is present (and heater is heating).

There are 3 LEDs connected:

LED1 - PA4

LED2 - PA1

LED3 - PA3

When 12V is detected show the LED climbing the temperature:

When less than 20°C → LED3 blinking, other LEDs off

When more than 20°C → LED3 and LED2 blinking, LED1 off

When more than 60°C → turn on LED3, keep all LEDs continuously turned on without blinking

Do you need anymore details on the above? Tell me if there is any better user improvement that can be done.

RESPONSE

Confirmed wiring and thresholds. Suggested improvement implemented: blink speed proportional to temperature band:

- < 20°C, LED3 only → 1000ms (slow)
- 20–40°C, LED2+3 → 600ms (medium)
- 40–60°C, LED2+3 → 250ms (fast)
- ≥ 60°C → Solid ON, no blink

This gives an intuitive "urgency increasing as it heats up" feel. Hardware cost is zero.

All LEDs OFF when 12V is absent. Added `updateLEDs_temperature()` as self-contained function using non-blocking `millis()` half-period toggle for blinking.

Exchange 4

PROMPT

There is a vibration motor connected on PB1 through a MOSFET.

Introduce a 3-speed control for vibration motor through PWM: low, medium and high, when a push button connected on PA7 is pressed.

When 12V is not present (when outside docking station) do the following functions for the vibration motor:

Initially the vibration motor should be off.

When push button is pressed for 1st time → LOW mode

When pressed for 2nd time → MEDIUM

When pressed for 3rd time → HIGH

When pressed for 4th time → turn off completely

Do you need anymore info?

RESPONSE

Confirmed PB1 supports PWM via TCA0 WO1 on megaTinyCore using `analogWrite()`.

PWM duty cycles:

- Low: 30% (PWM value 77)
- Medium: 60% (PWM value 153)
- High: 100% (PWM value 255)

Key decisions:

- 50ms software debounce on PA7

- 12V reconnect → motor stops immediately, resets to OFF, button ignored when docked
- LED feedback added for motor modes (undocked):
 - Motor OFF → all LEDs OFF
 - Motor LOW → LED3 solid
 - Motor MED → LED2+3 solid
 - Motor HIGH → LED1+2+3 solid
- delay(20) used in undocked mode instead of delay(1000) for button responsiveness
- Button fully ignored when docked — zero chance of accidental motor cycling

Exchange 5

PROMPT

OK lets remove all the blinking from the code.

RESPONSE

All blinking removed entirely. Changes made:

- Removed lastBlinkTime, blinkPhase, and all BLINK_* constants
- updateLEDs_temperature() and updateLEDs_vibration() are now simple if/switch statements with setLEDs() calls
- Temperature LED bands simplified: since there is no speed difference between 20–40°C and 40–60°C bands without blinking, they collapse into single condition — anything from 20°C to target shows LED2+3 solid
- Three clean states: LED3 only → LED2+3 → all three
- Undocked motor OFF now shows all LEDs off

Exchange 6

PROMPT

When undocked and when the temperature is below 40°C keep the three LEDs blinking at 400 milliseconds (to notify the user that the temperature is below 40°C). When the temperature is less than 40°C do not turn on the vibrator even if the buttons are pressed.

RESPONSE

Confirmed using live thermistor reading to gate motor and LEDs even when undocked.

Behaviour implemented:

- Undocked + temp < 40°C → All 3 LEDs blink @ 400ms, motor locked OFF, button ignored
- Undocked + temp ≥ 40°C → Solid LEDs per motor mode, button works normally
- If temp drops below 40°C while motor running → stopMotor() called immediately

delay(20) kept in undocked mode. Thermistor read takes ~32ms, so total undocked loop iteration ~50ms. Button debounce stays sharp and blink timing accurate.

Added warningActive static flag for clean blink state reset when transitioning from warning back to normal.

Exchange 7

PROMPT

When 12V is reconnected the vibration functions should not turn on, only the temperature climbing LED functions should be there. Vibration function should only happen when it is out of docking station and disconnected from 12V. When cycling through modes, add the suggested LED feedback.

RESPONSE

Implemented clean separation:

- 12V present (docked): LEDs show temperature display, motor forced OFF, button ignored
- 12V absent (undocked): LEDs show motor mode, button controls motor
- These are mutually exclusive — no conflict possible

Motor resets to OFF on both 12V loss and 12V restore. motorMode reset to MOTOR_MODE_OFF on reconnect so next undock always starts fresh. Button call only inside !v12Now branch.

Exchange 8

PROMPT

I will clear in 1 line after reaching below 40°C. Long press sleep mode.

RESPONSE

Motor lockout on temp < 40°C simplified to single unconditional line:

```
stopMotor(); // always reset immediately
```

Removed the if (motorMode != MOTOR_MODE_OFF) guard and serial print. stopMotor() is safe to call unconditionally — it just writes 0 to PWM register which costs nothing.

Exchange 9

PROMPT

This is a working code. To this add power off while long press — completely off.

RESPONSE

Implemented deep sleep on 2-second long press:

- checkButton() replaces detectButtonPress() — returns 0=nothing, 1=short press, 2=long press ≥ 2s
- Long press works in both docked and undocked modes
- enterSleep() sequence:
 1. Serial flush
 2. Heater OFF, motor OFF, LEDs OFF
 3. 3-flash confirmation (80ms on / 80ms off × 3)
 4. Reset all runtime state
 5. Configure PA7 as falling-edge pin-change interrupt
 6. Enter SLEEP_MODE_PWR_DOWN (~0.1µA)
 7. Execution pauses at sleep_cpu()
- ISR(PORTA_PORT_vect) clears interrupt flag — wake only, no other action
- After wake: restores PA7 to normal INPUT_PULLUP, 200ms settle delay, fresh boot
- v12WasAbsent reset after wake for clean 12V re-evaluation
- Motor stays OFF on wake (default fresh boot behaviour)

Exchange 10

PROMPT

C:\Users\ALI ABDUL GAFOOR\Documents\Arduino\Aura003\Aura003.ino: In function 'void loop()':
error: expected initializer at end of input
static bool v12WasA
Compilation error: expected initializer at end of input here I am using ATtiny1624.

RESPONSE

Error was caused by code truncation — the chat cut off the last portion of loop().

Root cause fixed: two duplicate static bool tracking variables were declared inside nested scopes (wasWarning inside the else block and warningActive below it). Replaced with a single static bool warningTracked declared at the top of loop(). This eliminates scope confusion that caused truncation and makes logic cleaner — one flag, one job, checked in both < 40 and ≥ 40 branches.

Complete untruncated file delivered in single block.

Exchange 11

PROMPT

Give final cleaned code.

RESPONSE

Delivered complete final build with all features accumulated across the project:

- Thermistor reading via Beta equation (100k Ω NTC, 4.7k Ω fixed, PA6)
- Heater state machine: HEATING $\rightarrow$ MAINTAINING (bang-bang $\pm 0.5^\circ\text{C}$, 3 min) $\rightarrow$ COOLING
- 12V interlock: full state reset on 12V disconnection
- Docked LED indicators: LED3 below 20°C , LED2+3 from 20 – 60°C , all three at $60^\circ\text{C}+$
- Undocked vibration motor: button cycles OFF/LOW(30%)/MED(60%)/HIGH(100%) with LED feedback
- 400ms warning blink on all LEDs when undocked and temp below 40°C
- Motor locked off unconditionally (single stopMotor() call) when temp $< 40^\circ\text{C}$
- Deep power-down sleep on 2-second long press with 3-flash confirmation
- Wake via PA7 pin-change interrupt, fresh boot on wake

Exchange 12

PROMPT

When 12V input given automatically need to start heating. After that heat need to raise up to 60°C then stop. While heating raising, 3 LEDs need to be blinking continuously, stop after reaching 60°C . Vibration start with 3 modes using push button after raising. Down LED blink continuously for 10 seconds and stop blinking and stop vibration.

RESPONSE

Requested clarifications before implementing:

1. On point 2 — when heater stops at 60°C : Is the MAINTAINING (bang-bang hold for 3 minutes) still needed, or does it just cut off at 60°C and stay off?

2. On point 3 — vibration after 60°C: Does vibration only become available after reaching 60°C, or can user start it any time after undocking?

3. On point 4 — the 10 second blink: Is this specifically when temp drops below 40°C while vibrating, or whenever temp drops below 40°C regardless of motor state? After the 10 seconds + vibration stops, does the device go idle (LEDs off, waiting), or sleep?

4. On long press sleep: Still needed, or removed in this new version?

Awaiting confirmation before writing updated code.