A FIVE-DISPLAY DESK STATION

Auroraeon DIY

Project status, operating concept, and future direction

CURRENT PHASE	PARTS ARE ON THE WAY	NEXT: BENCH VALIDATION

Version 1.0 | 22 August 2026

SECTION 01

The project right now

The product vision is defined, the software and presentation model are prepared, and the core physical parts are in transit. The next milestone is proving the design on real hardware.

AREA	CURRENT POSITION	NEXT PROOF
Core parts	On the way	Inspect exact revisions and electrical ratings
Industrial design	Rendered concept complete	Turn it into a measured, printable enclosure
Windows control	EXE built and visually checked	Test over USB with the assembled clock
macOS control	Native source prepared	Build the app and DMG on a Mac
ESP32 firmware	Feature scaffold written	Compile, flash, and tune on hardware
Online services	Integration plan defined	Add Wi-Fi setup, NTP, AccuWeather, and AQI

What this means

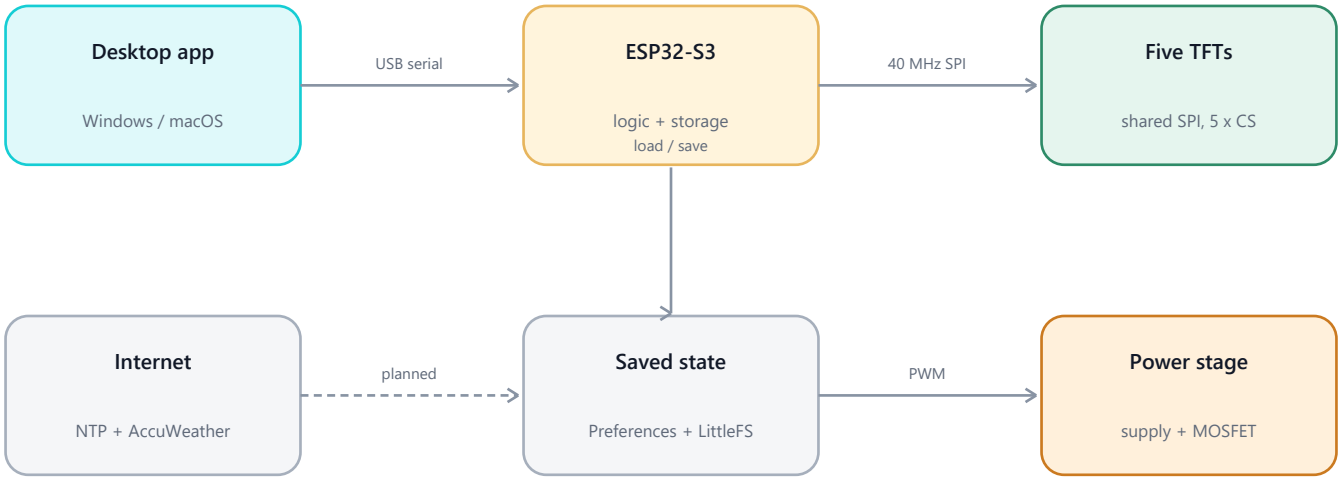
- The project is not being presented as a finished commercial device. It is at the pre-assembly prototype stage.
- The render shows the intended form and screen arrangement; exact fit, current draw, thermals, and signal integrity still need measurements.
- The Windows controller is runnable now. The ESP32 firmware and macOS port still need real-device builds and tests.

The goal is a desk object that combines a sculptural clock face, colored matrix weather art, and the flexibility of a programmable display system.

SECTION 02

How the station works

One ESP32-S3 coordinates all five displays. Four screens become a large four-digit clock; the left screen becomes a compact weather and air-quality panel.



0	**WEATHER**	Colored dot-matrix symbol, temperature, condition, and AQI
1	**HOUR TENS**	First digit of 24-hour time
2	**HOUR ONES**	Second hour digit
3	**MINUTE TENS**	First minute digit
4	**MINUTE ONES**	Final minute digit

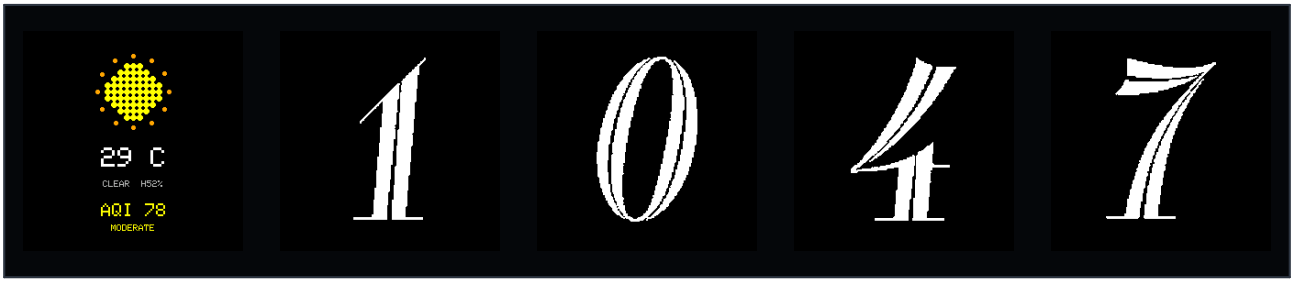
Display bus

MOSI, clock, data/command, and reset are shared. Each TFT receives a separate chip-select line, so the ESP32 can address one round panel at a time. Settings live in non-volatile Preferences storage; uploaded GIFs live in LittleFS.

SECTION 03

The visual system

The clock is designed around true-black screens, high-contrast decorative numerals, and restrained color on the weather display.



DEFAULT

OG Sculpted Serif

The favorite original style is the default: ornate white numerals on black, shared by all four time screens.

CUSTOM

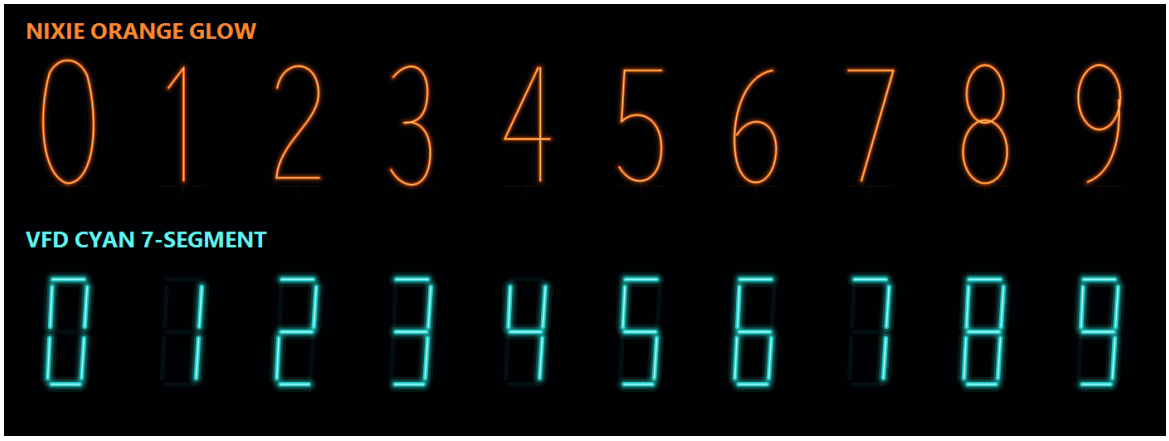
Nixie Orange Glow

Curved wire-like numerals use layered amber strokes to suggest a lit Nixie cathode.

CUSTOM

VFD Cyan 7-Segment

Cyan segments, dim inactive elements, and a small bloom recreate a VFD-like display.



The desktop preview above is generated from the same code-drawn Nixie and VFD elements used by the controller design.

SECTION 04

Hardware and power

The core parts are on the way. Their exact revisions must be checked before final wiring because marketplace GC9A01 boards can differ in voltage and backlight circuitry.

EXPECTED CORE PART	ROLE	WHEN IT ARRIVES
ESP32-S3 board	Clock logic, USB, storage, future Wi-Fi	Confirm board pinout, USB mode, flash, and PSRAM
5 x 1.28-inch GC9A01 TFT	Weather plus four time digits	Confirm VCC, BL circuit, connector order, and physical dimensions
Regulated supply	Powers displays and backlights	Select only after measuring one module and adding at least 25% margin
Logic-level MOSFET	Global PWM backlight control	Verify full enhancement at a 3.3 V gate drive
Base, harness, connectors	Hides pins and distributes power/data	Finalize from measured production parts

POWER RULE Five display backlights must never be driven directly from an ESP32 GPIO. Use a properly sized external regulated supply, shared ground, and a suitable MOSFET switching stage.

Current firmware pin plan

MOSI 11	SCLK 12	DC 9	RST 10	BL PWM 13	CS 4,5,6,7,8

SECTION 05

Software and control

The clock is intended to be customizable over USB: one shared digit font, editable element colors, brightness control, host time sync, profiles, and custom GIFs.

BUILT

Windows controller

A native WinForms EXE has been built and visually checked. It finds COM ports, previews five screens, applies styles, saves JSON profiles, synchronizes time, and uploads GIF data at 921600 baud.

SOURCE READY

macOS controller

A native SwiftUI source port is prepared with the same controls and serial protocol. It still needs to be compiled and USB-tested on macOS with Xcode.

Why there is not a drag-and-drop DMG yet

A DMG is a macOS disk image that normally contains a compiled .app and an Applications-folder shortcut. This project was created in Windows, which cannot run Xcode, produce the final macOS application binary, sign it with an Apple identity, notarize it, or test Gatekeeper behavior. Packaging unbuilt source inside a DMG would look finished but would not give you an installable application.

1	**Build**	Run the Swift package on a real Mac with Xcode 15+.
2	**Test**	Connect the ESP32 and verify serial, profiles, previews, and GIF upload.
3	**Sign**	Use an Apple Developer ID and hardened runtime for distribution.
4	**Notarize**	Submit to Apple so Gatekeeper accepts it on other Macs.
5	**Package**	Create the polished DMG with the app and Applications shortcut.

The intended final Mac experience is still exactly the familiar drag-to-Applications DMG; it is simply a later release step that must be completed on macOS.

SECTION 06

Time, Wi-Fi, and weather

The finished station should run without a computer: it will join a saved Wi-Fi network, correct its clock from the internet, and refresh weather information on a controlled schedule.

PLANNED

Wi-Fi setup

On first use, the clock should create a temporary setup network. A phone or computer selects the home Wi-Fi and stores its credentials safely.

PLANNED

Accurate time

NTP should synchronize at boot and about every hour. The local clock continues running between checks, so drift is corrected without constant traffic.

PLANNED

Live conditions

AccuWeather supplies temperature and icon codes. A chosen AQI source supplies air-quality data. The last good values remain visible during outages.

	Load saved appearance and last known data
	Join the saved Wi-Fi; open setup mode only when needed
	Correct system time and apply the local time-zone rule
	Request weather and AQI at an API-safe interval
	Render the latest valid data; retry failures without freezing the clock

Current truth

The firmware already knows how to draw weather values and map AccuWeather icon numbers 1-44 into colored dot-matrix symbols. It does not yet contain Wi-Fi provisioning, NTP, HTTPS requests, an API key, or AQI acquisition. Those are the main software additions after hardware bring-up.

SECTION 07

The physical design

Five raw modules stand exposed above a low black wedge. Their pins and wiring disappear into the base, preserving the floating instrument-like look.

BASE	DISPLAY PITCH	LCD HOUSING	MODULE INSERTION
about 235 x 65 x 24 mm	about 43 mm	about 35.58 mm	about 3.2 mm deeper

Before fabrication

- Measure the delivered PCBs and headers.
- Add hidden slots or clamps, a removable lid, screw posts, cable channels, ventilation, and ESP32/power mounts.
- Keep USB, reset, and service access reachable.

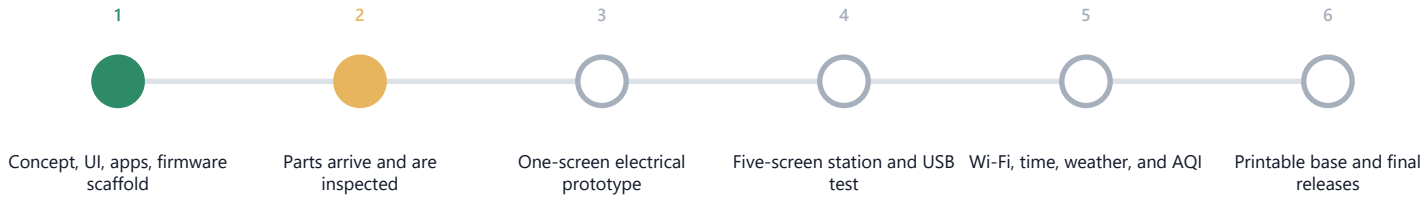
Long-running display care

- GC9A01 panels are TFT LCDs, not OLEDs; classic OLED burn-in is not the expected effect.
- Use only the brightness needed, dim when idle, and keep heat away from the displays and regulator.
- Run multi-day soak tests before unattended use.

SECTION 08

Future plan

The path forward is intentionally staged: prove one part at a time, keep the visual identity, and only call the project finished after the complete station survives real use.



NEAR TERM	AFTER HARDWARE WORKS	LONGER TERM
Inspect the delivered boards; confirm voltage, pinout, and physical dimensions. Compile the firmware and prove one display with a current-limited supply.	Add all five panels, tune SPI speed and screen alignment, validate power and temperatures, then test every Windows USB command and GIF upload.	Add setup-portal Wi-Fi, hourly NTP correction, AccuWeather and AQI, a production enclosure, a signed Windows release, and a signed/notarized drag-to-Applications DMG.

Definition of success

A finished Auroraeon station will power up cleanly, show accurate time and fresh weather without a computer, accept customization from Windows or macOS, recover gracefully from network or USB interruptions, remain thermally stable for days, and preserve the exposed five-display design that made the project distinctive from the start.

Selected technical references

ESP32-S3: https://documentation.espressif.com/esp32-s3_datasheet_en.pdf
Arduino-ESP32 USB CDC: https://docs.espressif.com/projects/arduino-esp32/en/latest/tutorials/cdc_dfu_flash.html
GC9A01A library: https://github.com/adafruit/Adafruit_GC9A01A
AccuWeather icons: https://developer.accuweather.com/documentation/weather-icons