

Ryan Tao

Electrical Engineering, UCLA '29 | Embedded, RF/Analog, FPGA, PCB, DSP

tyanrao@ucla.edu | 626-363-3145 | Los Angeles, CA

ryanlintao.com | [LinkedIn](#) | [GitHub](#)

Second-year EE student at UCLA, co-founder and EE/firmware lead at Slingshot (a basketball training-load wearable startup). I take hardware from first principles to a finished device and debug it until it works. U.S. citizen.

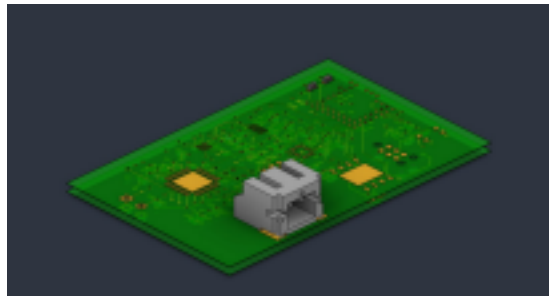
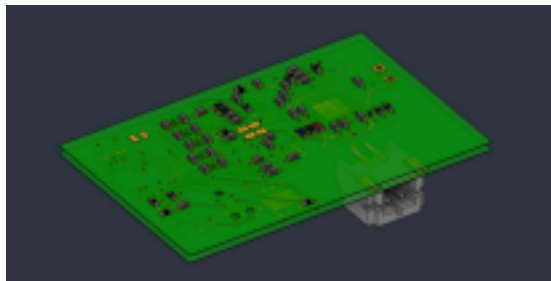
GPA: 3.51 | John Richard Leffler Scholarship (electronic engineering) | Muriel K. & Robert B. Allan Fund | Salutatorian, Glen A. Wilson HS

01 STARTUP

Slingshot: BLE Training-Load Wearable

Co-Founder & EE/Firmware Lead | March 2026 - Present | slingshot-sports.com

A battery-powered sensor pod worn on an upper-arm compression sleeve that fuses motion and heart-rate data into a training-load score. As one of two engineers on a three-person team, I own the board and firmware end to end.



Slingshot V1

PCB 3D renders. Top side (left): nRF52840 module, USB-C. Bottom side (right): dense passive population, power ICs, sensor footprints.

Hardware

- Designed a 4-layer mixed-signal PCB (nRF52840 BLE SoC, five-sensor suite, QSPI NOR flash, full power management) in Autodesk Fusion Electronics.
- Debugged all five V1 boards via DMM continuity testing, root-caused to hand-reflow soldering of the nRF module, and drove the V2 respin to a contract manufacturer: nRF54L15 module, nPM1300 PMIC replacing four discrete power ICs, ~500 DRC violations resolved on 0.35 mm-pitch WLCSP/BGA layout via via-in-pad escape routing.
- Architected the battery-management power tree: USB-C charging, buck-boost regulation to 3.3V, Li-Po fuel gauging. Buck-boost over LDO to hold the rail stable across the full Li-Po discharge curve (3.0-4.2V crosses the 3.3V output).
- V2: 5 units in factory build with flying-probe test.

Firmware

- Embedded C/C++ on PlatformIO: I2C/SPI/QSPI sensor drivers, on-device flash logging, USB-CDC export, BLE telemetry. IMU + barometer sensor fusion for jump-height estimation.
- Caught two pre-fab design errors (power-mode conflict, regulator feedback divider) through ERC and schematic review against datasheets.

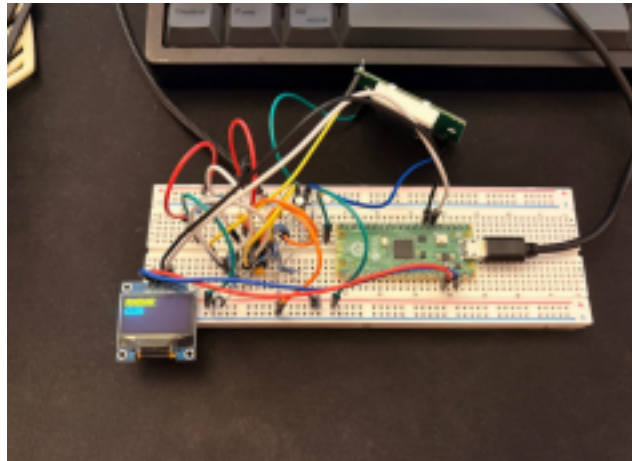
Parts: nRF52840, nRF54L15, nPM1300, BMI323, BMP585, W25Q64 | Tools: Fusion Electronics, PlatformIO, BLE, Web Bluetooth, SolidWorks, JLCPCB

02 PERSONAL BUILD

X-Band Doppler Radar Speed Gun

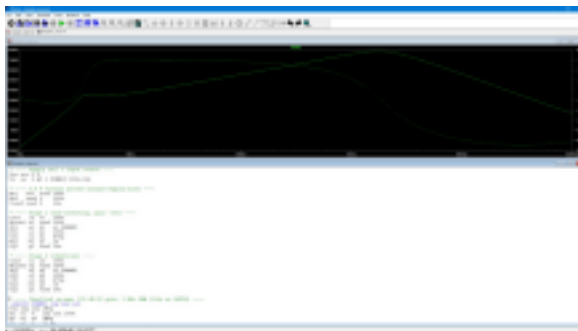
Solo Build | June 2026 | [GitHub](#) | [Build page with demo videos](#)

A handheld radar speed gun that measures vehicle speed in real time from a 10.525 GHz X-band module through a custom analog chain, into a real-time FFT pipeline, to a live OLED readout. Runs standalone on a USB power bank.



Complete radar build: Pico 2 (RP2350), HB100 X-band module (top right), two-stage analog amplifier chain, SSD1306 OLED displaying 'RADAR GUN.' Video demos at ryanlintao.com/doppler.html

- RF front end: HB100 X-band module at 10.525 GHz, Doppler sensitivity of 31.36 Hz/mph. • Analog signal conditioning: multi-stage non-inverting amplifier ($\sim 101\times$ per stage) for microvolt-level Doppler return, DC biasing via resistive divider, AC coupling. Verified in LTspice, then debugged on breadboard to working readout.
- DSP: 1024-point real-time FFT at 8 kHz sampling (~ 0.25 mph resolution, max ~ 128 mph) with peak detection, noise thresholding, and moving-average smoothing.
- Phased build: analog hardware first, Python/NumPy DSP prototype second (validated through PC sound card), C/C++ embedded port on RP2350 third. Each stage verified before integration.



Left:

LTspice simulation, two-stage amplifier frequency response with full netlist visible. Right: Python DSP prototype streaming live mph readings with peak tracking.

Stack: RP2350/Pico 2, HB100 X-band, LM358 op-amp, SSD1306 OLED, C/C++, Python/NumPy, LTspice, PlatformIO, ArduinoFFT, Audacity

03 PERSONAL BUILD | IN PROGRESS

Real-Time FPGA Spectrum Analyzer

Solo Build | July 2026 - Present | [GitHub](#) | [Full build page](#)

A live spectrum analyzer in Verilog on a Sipeed Tang Nano 9K. An external SPI ADC samples an analog signal, the FPGA computes frequency content via a fixed-point Goertzel filterbank, and renders the spectrum to a display. No

processor in the signal path. Endgame: feed the radar gun's amplifier output into this board for all-hardware DSP.

- Dual-clock-domain architecture: ADC at ~20 kHz, display at 25 MHz pixel clock. Bridged by a gray-code async FIFO (one pointer bit changes at a time; mid-transition reads safe).
- Custom SPI master FSM for MCP3008 ADC with two-edge discipline (MOSI on falling, MISO sampled on rising). Caught an off-by-one-bit shift ($0x14A = 0x2A5 \gg 1$) from sampling phase misalignment.
- VGA/DVI timing generation from first principles: 640x480 @ 60 Hz, verified at exactly 307,200 visible pixels (640x480) and 420,000 total (800x525).

- Six RTL modules simulation-verified with self-checking Icarus Verilog testbenches, checked against Python/NumPy fixed-point reference models. Nothing touches the board until its testbench passes.
- 16-bin fixed-point Goertzel filterbank (Q1.15) in progress. Pipelined 256-point radix-2 FFT planned.

Stack: Verilog, Gowin EDA, Icarus Verilog, GTKWave, Python/NumPy, MCP3008 ADC, MAX4466 mic, Tang Nano 9K, Git

04 RESEARCH

Ferrofluid Computer Vision

Research Intern, i2bl Lab, UCLA | November 2025 - Present | [Project page with lab footage](#)

Real-time computer vision for microfluidic control. The system detects and tracks ferrofluid droplets on microfluidic chips, estimating per-droplet volume from vision alone, and closes a control loop that steers droplet trajectories by adjusting magnetic field placement.

- Developed and trained a real-time CV model to segment and localize ferrofluid flow and magnetic sources from live camera data.
- Closed a vision-in-the-loop control system: model output drives magnetic field placement to steer ferrofluid trajectories on microfluidic chips in real time.
- Per-droplet volume estimation in microliters from vision alone.

Stack: Python, OpenCV, real-time inference, microfluidics

05 PERSONAL BUILD

ESP32-S3 Keychain Mini Camera

Solo Build | 2026 | [Project page](#)

A pocket-sized WiFi-transfer keychain camera on a XIAO ESP32-S3 Sense. Tap to snap, hold to broadcast a WiFi AP for phone-side photo transfer through a browser gallery. No app, no cable. 3D-printed enclosure designed in Fusion 360.



ESP32-S3 Sense module with camera (left), 3D-printed enclosure front (center) and back (right).

Technical Skills

Hardware & PCB Design

Schematic capture, 4-layer mixed-signal PCB layout, fine-pitch BGA/WLCSP & via-in-pad escape routing, power electronics (PMIC, DC-DC buck-boost, Li-Po charging, fuel gauging), analog circuit design, analog signal conditioning, RF /

2.4 GHz layout, I2C / SPI / QSPI, board bring-up (SWD / J-Link), design for manufacturing (DFM), board debug (DMM, continuity/flying-probe test)

Digital & FPGA

Verilog, RTL design (FSMs), clock domain crossing (gray-code async FIFO), fixed-point DSP (Q-format, Goertzel), video timing generation, UART/SPI from scratch, Field-Programmable Gate Arrays (FPGA), self-checking testbenches vs. NumPy golden models

Embedded Systems & DSP

Embedded C/C++, firmware development, Nordic nRF52/nRF54, RP2350, ESP32-S3, Bluetooth Low Energy (BLE), real-time FFT, digital signal processing, sensor fusion, low-power design, PlatformIO

RF & Analog

Radio frequency (RF) engineering, Doppler radar, X-band, analog circuit design, signal conditioning (biasing, AC coupling), LTspice simulation, ADC sampling, Nyquist & resolution

Software & Languages

C/C++, Python, NumPy, Verilog, Java, JavaScript, React, HTML/CSS, Git

CAD & Tools

Autodesk Fusion Electronics (ECAD), Autodesk Fusion 360 (MCAD), SolidWorks, PlatformIO, LTspice, Gowin EDA, Icarus Verilog, GTKWave, JLCPCB, Git

Domains

Mixed-signal IC design, computer vision, machine learning, embedded systems, signal processing

Education

University of California, Los Angeles | B.S. Electrical Engineering | Expected June 2029 | GPA: 3.51

Relevant coursework: Linear Algebra, Differential Equations, Multivariable Calculus, Physics (Mechanics; Oscillations, Waves & Fields), Intro to Computer Science I & II

Honors: John Richard Leffler Scholarship (electronic engineering), Muriel K. & Robert B. Allan Fund (Samueli School of Engineering)

Full interactive portfolio with build photos, 3D renders, demo videos, and source code:

ryanlintao.com

Ryan Tao | tyanrao@ucla.edu | 626-363-3145 | Los Angeles, CA