Luke Trusheim

San Francisco Bay Area
linkedin.com/in/luketrusheim github.com/ltrush

Education

California Polytechnic State University, San Luis Obispo **Sep. 2022 – Jun. 2026**
Bachelor of Science in Computer Engineering | 3.92 GPA, Summa Cum Laude, 12x Dean's List *San Luis Obispo, CA*

Experience

NZXT, Inc. **Jul. 2025 – Sep. 2025**
Firmware Engineer Intern *San Francisco Bay Area*

- Designed, implemented, and validated a wireless lighting control system on Arduino Nano 33 BLEs
- Implemented multi-protocol firmware integrating BLE, UART, and SPI with a coprocessor handling 800 kHz timing
- Evaluated system architecture tradeoffs (BLE star vs. mesh topology, streaming data vs. command-based animation)
- Authored technical report with architecture recommendations to move prototype toward manufacturable product

Computer Engineering Department **Nov. 2024 – Jun. 2026**
Computer Engineering Tutor *San Luis Obispo, CA*

- Help students debug and understand microcontrollers, firmware, computer architecture, circuits, and digital design

Projects

Buddy Finder Compass | *ESP32, SPI, I2C* **Jan. 2026 – Jun. 2026**

- Developing handheld GPS compass devices on ESP32 that locate each other over point-to-point LoRa
- Architected multi-core FreeRTOS firmware with dedicated radio and navigation tasks communicating via queues
- Integrated SPI (LoRa), I2C (IMU), and UART (GNSS) peripherals with interrupt-driven async receive

Minix Character Device Driver | *C, Linux* **Mar. 2026**

- Built character device driver with owner-gated read/write access and ioctl-based ownership transfer between processes
- Maintained driver state across live updates using the Minix System Event Framework

Mesh-Connected Wildlife Camera Using Computer Vision | *Raspberry Pi, Python* **Sep. 2025 – Mar. 2026**

- Rewrote open-source mesh network protocol in Python to work on ARM-based Linux systems (R. Pi Zero 2 W)
- Integrated motion sensor, camera, solar panel and batteries with Raspberry Pi and computer vision models
- Built multithreaded control software with watchdog recovery managing GPIO, inference, and LoRa packet transfer
- Configured systemd services for reliable bring-up and automatic restart after fault or power loss

Lightweight Process Library | *C, x86-64 Linux* **Jan. 2026**

- Implemented a user-level threading library in C with cooperative context switching and a round-robin scheduler
- Managed full thread lifecycle including creation, yielding, termination, and blocking with FIFO scheduling queues

Autonomous Mobile Robot for Human Detection and Tracking | *ROS2, C++* **Nov. 2025 – Dec. 2025**

- Built ROS2 nodes that filtered detections by class ID and confidence threshold ($\geq$65%) before publishing targets
- Implemented coordinate frame transformations using tf2 to convert detected pose into the robot's base link frame
- Tuned a proportional controller with velocity clamping and deadzones to maintain a stable 1m following distance

Bare-Metal Digital Camera on STM32 | *STM32, SPI, I2C* **May. 2025 – Jun. 2025**

- Developed bare-metal firmware for STM32-based digital camera system integrating camera, LCD screen, and SD card
- Implemented interrupt-driven DMA pipeline to capture and transfer real-time image data without frame loss
- Configured peripheral registers based on datasheet specs and validated SPI/I2C bus signals on oscilloscope

Reliable File Transfer Protocol Over UDP | *C, UDP* **May. 2025**

- Built a reliable file-transfer protocol over UDP in C using selective-reject sliding window and state machines
- Verified correctness under injected packet loss and bit-error rates across varying window and file sizes

2025 WERC Environmental Engineering Competition | *Python, Arduino* **Oct. 2024 – Apr. 2025**

- Collaborated with a multi-disciplinary team of 8 engineering students for the 2025 WERC Environmental Design Contest, winning first place by researching a DERMS solution that integrates solar power and hydrogen fuel cells
- Implemented a bench-scale by using Arduino microcontroller to monitor and control peripheral devices
- Designed Python algorithm to simulate solution on real data, demonstrating stable energy output and cost savings

Pipelined and Cached RISC-V CPU on FPGA Board | *RISCV, SystemVerilog* **Sep. 2024 – Dec. 2024**

- Implemented a five-stage pipelined RISC-V core in SystemVerilog handling data/control hazards via stall, forward, flush
- Validated instruction correctness in control and data hazard scenarios and benchmarked CPU operations

Technical Skills

Languages: Python, C, C++, Bash, SystemVerilog
Developer Tools: VS Code, GitHub, Git, Make, Wireshark, Vivado, LTSpice
Technologies/Frameworks: Linux, FreeRTOS, ROS2, BLE, LoRa, GNSS
Electronics and Hardware: ESP32, STM32L4 (ARM), NRF52, DMA, SPI, I2C, UART, GPIO, PWM, oscilloscope, multimeter, logic analyzer, waveform generator, BJTs, MOSFETs, op-amps, datasheets, soldering