

BRADY LIN

bradylin2027@u.northwestern.edu | (224) 417-5308 | [linkedin.com/in/brady-lin/](https://www.linkedin.com/in/brady-lin/)
github.com/Bylin-code | Portfolio at bradylin.com

EDUCATION

Northwestern University

B.S. Electrical Engineering | Segal Design Certificate

Evanston, IL

September 2023 – June 2027

- GPA: 3.94 (Major)

SKILLS

Embedded development: C, C++, Python, embedded Linux, RTOS, I²C, SPI, UART, various microcontrollers

Hardware debugging: multimeter, serial monitor, oscilloscope, logic analyzer

CAD and EDA: SolidWorks, Onshape, Fusion360, AutoCAD, LTspice, MATLAB, KiCAD, EasyEDA, PCB layout

Fabrication and prototyping: 3D printing, soldering, laser cutting, waterjetting, milling, drill pressing, bandsawing, lathing

Language: English (native), Mandarin (native), experienced in communicating with manufacturers in both languages

EXPERIENCES

Formlabs

Somerville, MA

Incoming Mechatronics Intern

Jan 2026 - March 2026

- Building electromechanical and embedded control systems for next-generation SLS and SLA 3D printers.

Northwestern University | Center for Robotics and Biosystems

Evanston, IL

Robotic Systems Engineer

March 2025 – Present

- Developed firmware on ESP32-S3 and Raspberry Pi Pico for closed-loop stepper control and I²C networking across 20+ nodes, enabling autonomous operation of a 1.5 m × 1.5 m CoreXY gantry for a 150-drone battery-swap array
- Improved system accuracy by reducing X-axis racking error from ~8° to less than 0.5° through PID tuning and kinematics analysis
- Increased manufacturability and reliability by designing 30+ parts in SolidWorks CAD following DFM/DFA principles, then fabricating them via 3D printing, laser cutting, and machining.
- Validated embedded-mechanical integration by testing all custom parts across motion accuracy, load handling, and responsiveness.

Northwestern University | Creative Engineering & Robotic Arts Studio

Evanston, IL

Founder & Lead Engineer

April 2024 – Present

bradylin.com/projects/pulse/

- Led design and development of a six-armed robotic drummer integrating mechanical arms, animatronics, and LCD feedback, powered by a cross-platform Raspberry Pi + Teensy 4.1 system running embedded Linux and C++.
- Developed real-time embedded C++ firmware to coordinate servos, solenoids, and stepper motors, achieving zero effective latency synchronization with MIDI playback parsed by an external Python program.
- Optimized hardware-software integration by designing and validating 40+ mechanical, grounding, and thermal components in CAD.
- Produced a live performance for 100+ attendees, overseeing cross-team collaboration to showcase the robot playing with a band.

PROJECTS

Expressive Violin-Style MIDI Instrument | Designer & Hardware Engineer

Nov 2024 – July 2025

bradylin.com/projects/stradex/

- Built a violin-inspired MIDI controller with interrupt-driven C++ firmware on a custom PCB and Onshape-modeled enclosure
- Designed a compact PCB with 70+ components in KiCAD, integrating multiple I²C chips, sensors, and an RP2350 microcontroller
- Achieved sub-6 ms system latency by leveraging multicore firmware, optimizing USB-MIDI bandwidth, and debugging with oscilloscopes, logic analyzers, and UART, maximizing the instruments' responsiveness
- Validated reliability of 3D printed moving parts through 30+ rapid prototyping iterations and automated >10,000 actuation test cycles
- Released a project video that gained 15,000+ YouTube views, was featured in tech media such as Hackaday, and led to 20 unit sales

Low Level Computer-Vision Robotics Project | Mechatronics Engineer

March 2025 – May 2025

<https://bradylin.com/projects/kermit/>

- Engineered a low-level computer vision line-following robot by writing Raspberry Pi Pico C-SDK firmware with PID control for dual DC motors, winning 1st place in the project-based graduate level class *Advanced Mechatronics, MECH_ENG 433*
- Optimized real-time image processing by implementing an SPI camera pipeline, tuning a PID controller, and building a centroid-based tracking algorithm, enabling fast lap times and stable path-following performance
- Debugged inter-module communication using logic analyzers and oscilloscopes, catching multiple algorithm-breaking bugs