LOGAN HARTFORD

Bay Area, CA · logan.hartford@outlook.com · (705) 427-5690 · https://lhartford.com/projects/

SKILLS

Firmware: C/++, Go, Zephyr RTOS, Embedded Linux (Yocto), Device Drivers, Bootloaders, A/B OTA, Low-Power
Robotics: Python, ROS2, OpenCV, YOLO, Pose Estimation, EKF, Localization, Sensor Fusion, PID
Hardware: STM32, nRF52/53, PIC16/24, ESP32, i.MX93, BLE, Matter, Thread, NFC, UART/I2C/SPI/CAN-FD
Electrical: Altium, High-Speed Routing, Signal Integrity, Impedance Control, PDN Analysis, Power Converters

WORK HISTORY

Firmware Engineer | Level Home Jul. 2025 – Jun. 2026

- Reduced battery-related support tickets by **60% across 225K** units using a data-informed estimation algorithm.
- Led firmware for a **50K units/yr** embedded Linux networking hub by building a custom i.MX93 Yocto BSP.
- Enabled OTA management of **500 hubs** by adding A/B partitions, bootloader logic, and update services.
- Cut factory validation time by **20%** by developing a Go test app to exercise hub peripherals over UART.
- Raised battery measurement accuracy by **35%** by accounting for usage and stress across the system.
- Trimmed battery health prediction error by **2 weeks** using temperature and product-compensated thresholds.
- Boosted critical battery detection accuracy by **10%** using internal resistance derived from voltage droop.

Firmware Engineer | Level Home May 2024 – Aug. 2024

- Built a modular factory-test app for an nRF53 smart lock, running **20K units/yr** production.
- Reduced NFC polling power by **80%** with no added latency using a presence-sensor driver and app logic.
- Cut magnetometer power consumption by **90%** by gating door-position polling on accelerometer interrupt.
- Developed an E-ink display driver enabling grayscale and animations by designing custom transition waveforms.
- Quantified unwanted magnetizability across **15** product configs by scripting a fixture-based test protocol.

Robotic System Integration Engineer, Optimus | Tesla, Inc. Sep. 2023 – Dec. 2023

- Improved hand repair efficacy by **75%** by building a Linux/CAN test station with touch-sensor visualization.
- Increased hand repairs from **8 to 50/wk** by writing repair & triage procedures and training technicians on them.
- Enabled actuator validation by adding a CAN-command UI and tests that log motor current and speed.
- Reduced repair part usage by **45%** by creating a triage procedure ordered by likelihood, risk, and parts scarcity.
- Captured foot-sensor data for locomotion training by integrating a sensor boot into the collection stack.

Cell Manufacturing Engineering, Can Stamping | Tesla, Inc. Jan. 2023 – Apr. 2023

- Streamlined **200+** line-fault workflows by automating fault ranking, owner routing, and HMI updates in Python.
- Enabled QC trend tracking by automating measurement logging and backfilling **400+** historical PDFs in Python.

Hardware Design & Firmware Engineer | Edge AI Solutions Sep. 2021 – Dec. 2022

- Shipped **500-unit** boom-lift harness alarm, owning PCB and bare-metal PIC firmware for clip-in detection.
- Extended boom-lift alarm battery life from **20 to 128 days** using sensor sleep modes and tap-wake firmware.
- Rejected false motion alarms by fusing barometer and accelerometer data in a fixed-point Kalman filter on a PIC.
- Cut board test time from **20 to 2 min/unit** across 500 units using a bed-of-nails fixture and UART test firmware.
- Enabled Jetson Nano bring-up by implementing PIC-managed rail sequencing and power-good firmware.
- Cut sewer-robot footprint by **60%**, routing GbE and dual MIPI CSI-2 onto one **8-layer** impedance-controlled PCB.
- Reverse-engineered undocumented fiber-Ethernet hardware and stood up in-house pick-and-place production.
- Built a PIC16-managed 4S 18650 BMS enabling in-robot charging, charge/discharge control, and fault cutoff.

EDUCATION | UNIVERSITY OF WATERLOO

BASc Mechanical Engineering (Distinction) · Mechatronics, Computing & AI Minors Sep. 2020 – Apr. 2025
Electrical Lead, Team Captain · University of Waterloo Robotics Team Sep. 2021 – Jan. 2024

PROJECTS

WEEDWARDEN - WINNER OF BEST PROTOTYPE AWARD AT CAPSTONE SYMPOSIUM



Removal of a Fake Dandelion

Overview:

- Goal: Create an autonomous weed control robot.
- Raspberry Pi 5 for **ROS2** decisions and CV, with a **NUCLEO-STM32** for commutation and encoder capture.
- Pi Camera v3 w/ **YOLOv11n-Pose** model locating weeds.
- Cartesian system with 2 axes of motion and a drill motor.
- Locomotion system providing third axes of motion.

Contributions:

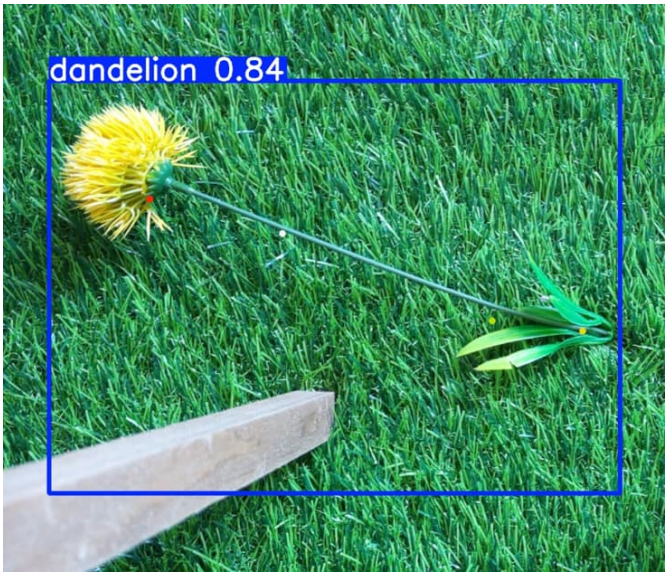
- Technical Lead, Software Lead.
- Defined high-level system and software architecture.
- Developed drivers for sensors and actuators in Python.
- Trained a deep CV model on custom dataset, optimized **EKF localization** and developed autonomous logic.
- Designed and cast rubber tires to improve autonomy.

Computer Vision:

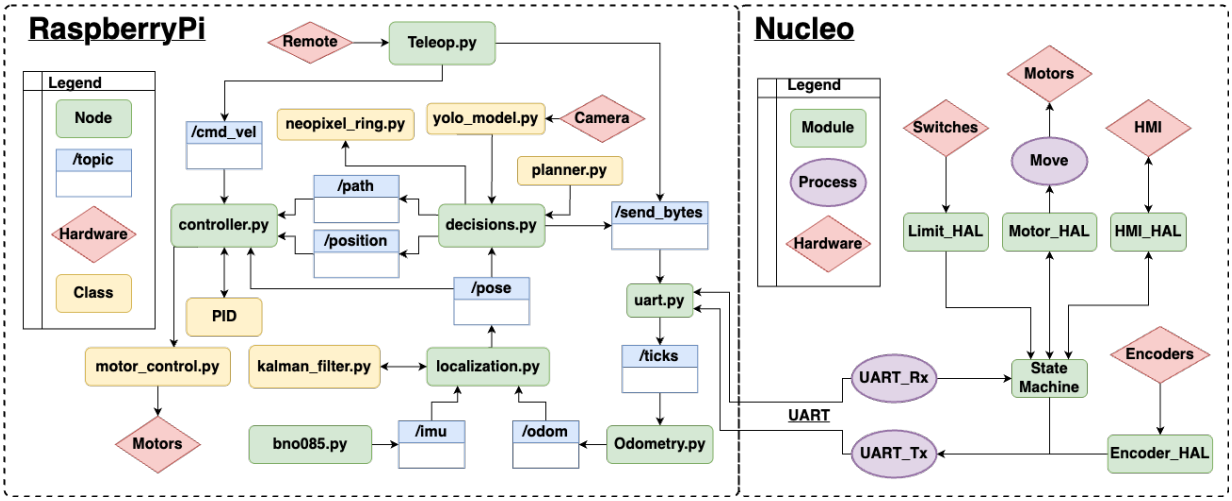
- **1600** custom images, augmented to **4800**, GPU-accelerated training of YOLOv11n-Pose model.
- Calibrated **homography** transform to convert from camera to robot frame.
- Camera and inference used to iteratively adjust the position of the drill in x and y directions.

Sensor Fusion and Control:

- Fused IMU (BNO085) and odometry data in an **Extended Kalman Filter**.
- Updated with acceleration, gyroscope and magnetometer outputs.
- Used motion capture system to tune filter.
- Implemented adaptive **PID control** for drive motors.



YOLOv11n-Pose Inference on Fake Dandelion



Software Architecture

PROJECTS

SCARA ROBOT

Overview:

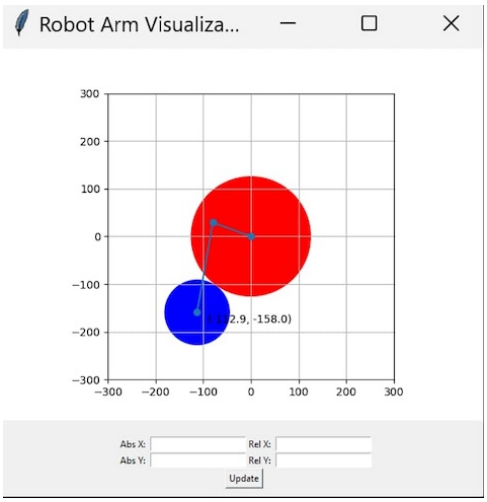
- SCARA pick-and-place robot designed for school design challenge.
- NUCLEO-STM32 with PlatformIO SDK.
- Two stepper-motor-driven planar revolute joints.
- Rack/pinion vertical axis with servo gripper.

Contributions:

- Technical Lead, Firmware and Controls Lead.
- Led system integration and debugging.
- Defined electrical and firmware architecture.
- Implemented **hardware abstraction** layers.
- Implemented controls functional layer.
- Simulated and validated control algorithms.



Final Assembly



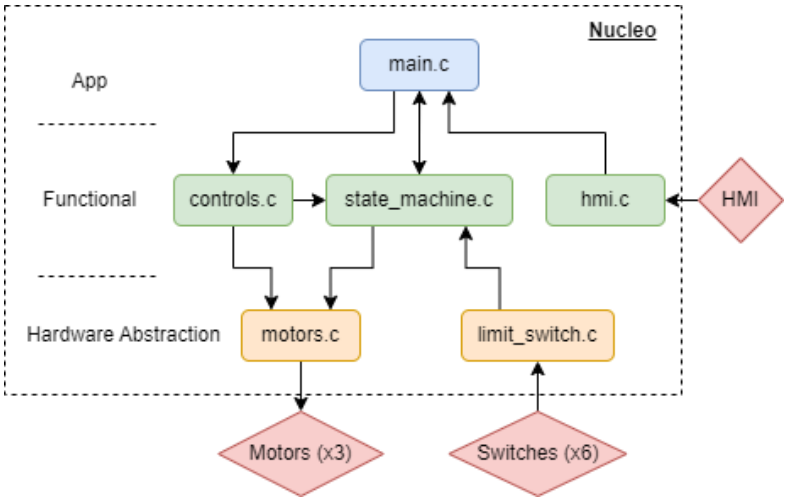
Python Kinematics Simulation

Other Features:

- Developed **asynchronous motor control** from scratch using hardware timers.
- Added gain scheduling to motor control for smoother, faster operations.
- Added **path planning** to joint control to make robot movements smooth and intuitive.
- Limit switches serviced asynchronously via **hardware interrupts**.
- Created manual and autonomous operating modes.

Control Algorithm:

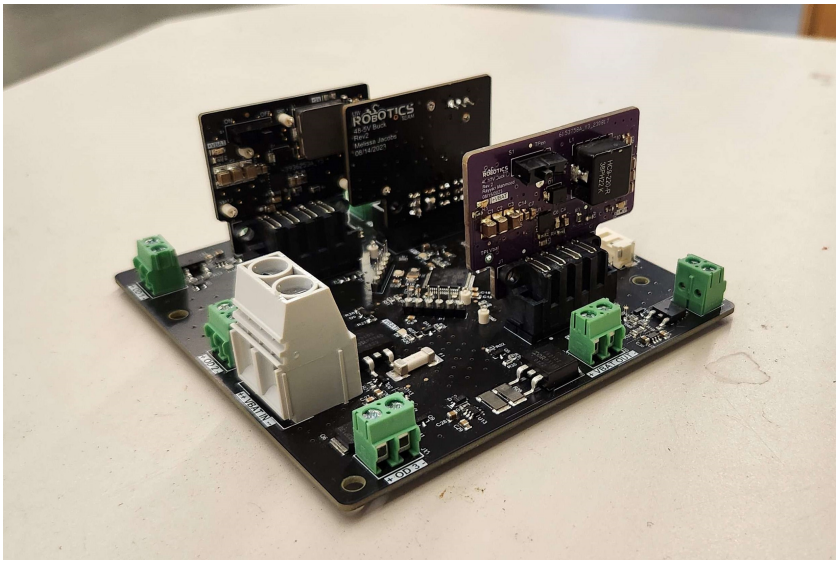
- Developed **IK and control** algorithms from scratch in Python and then implemented in C on the STM32.
- Created a simple simulation in Python using Tkinter.
- Added in physical parameters of the base and end effector to validate the hardware and software safety limits.
- Implemented logic to manually control the robot with joystick.
- Identified and resolved corner cases in the algorithm.
- Added ability to evaluate the quality of IK solutions.
- Ensured smooth operation of the robot throughout its full operating range.



Firmware Architecture

PROJECTS

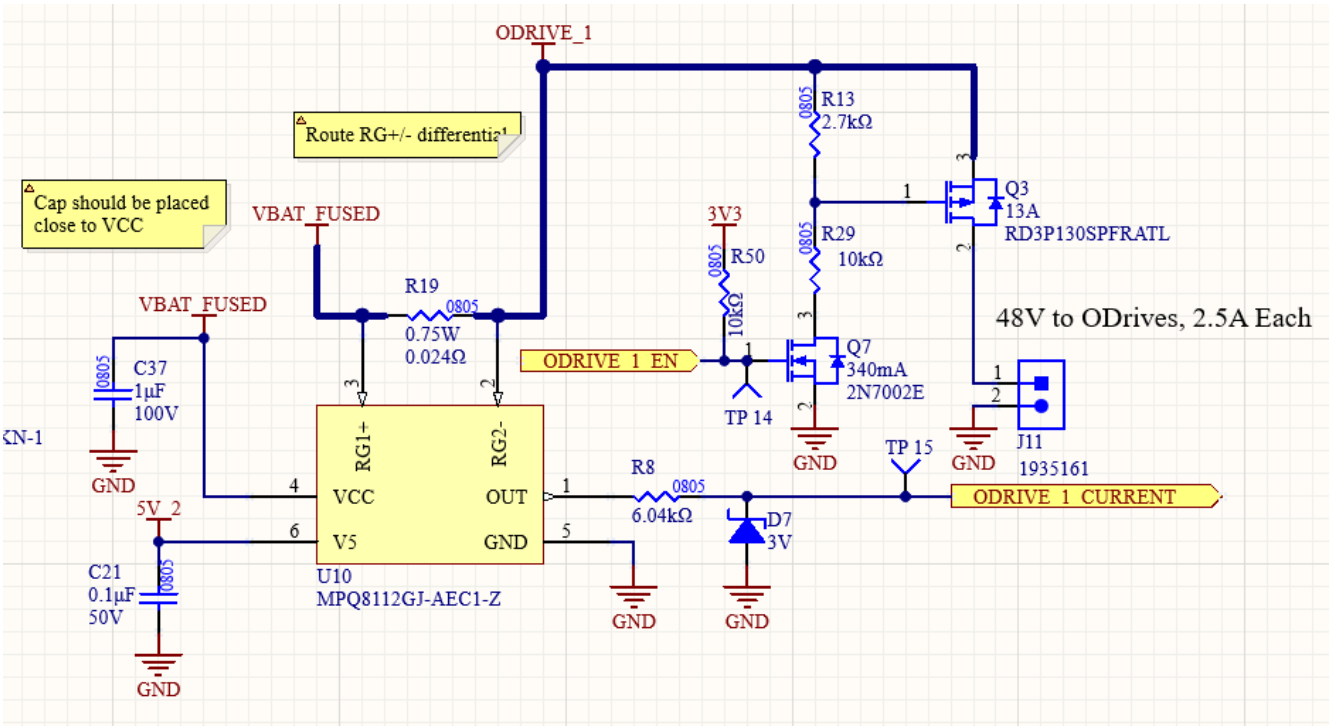
POWER DISTRIBUTION BOARD



The power distribution baseboard with buck converter modules

Overview:

- Reduced the scope of the system and removed unnecessary circuits.
- Modularized the 5V and 17V converters and integrated with vertical connectors.
- Implemented 5V buck converters in parallel to reduce the difficulty of the buck design.
- Added load monitoring and protection to all rails.
- New compact layout for the baseboard.
- Assembled and tested 6 boards across two revisions.



Custom load monitoring and protection circuit

- **Current sense amplifier** with a gain of 50, outputs 0-3V when current through the resistor is 0-2.5A.
- High-side NMOS-PMOS **load switch** to disconnect the load from power, keeping the load directly connected to ground and protecting against shorts to ground.
- Zener diode to protect the STM32 from signals larger than 3V3.
- At an output of 3V3 or higher from the amplifier, the current through the Zener diode is sufficient for break-down to occur, clamping the voltage to 3V.

PROJECTS

A PCB CHRISTMAS





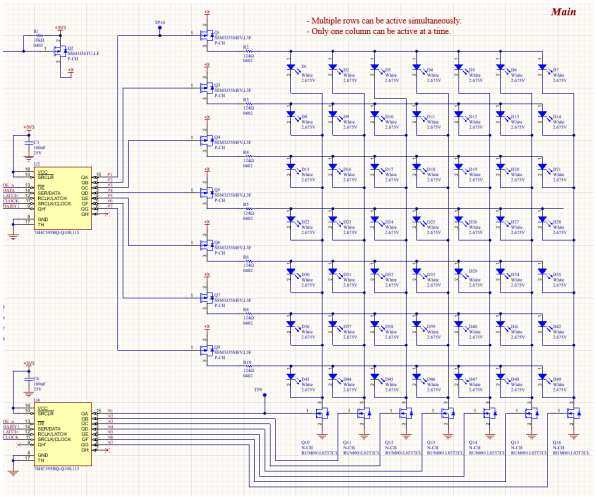
A device playing a song and the layout in Altium

Overview:

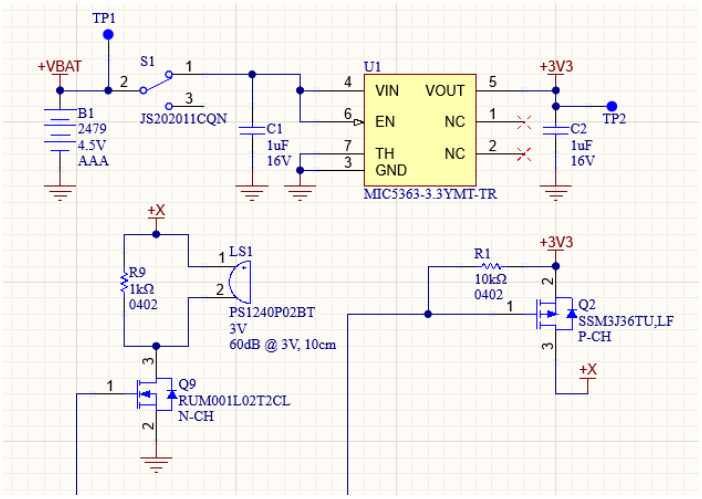
- Created using a **PIC16 MCU**, shift registers, an LDO, and a lot of FETs and LEDs.
- Designed with low-power principles, giving the device a **2+ year idle shelf life**.
- Large copper test points double as ornaments on the tree.
- Designed using smallest possible packages to obscure the inner workings of the device.
- Hand-assembled and tested 20 boards.

LED Matrix Design:

- Addressed **49 LEDs** individually using only **3 MCU pins** through multiplexing and shift registers.
- Matrix driven by 7 high-side PCHs and 7 low-side NCHs.
- State of the FETs controlled by two daisy-chained shift registers.
- Designed to have multiple rows but only one column active at a time, using high-frequency updates to give the illusion of individual addressing.



The 49 LED matrix



Low-power design components

Low-Power Design:

- Toggle switch to provide a hard disconnect from the batteries during storage.
- Load switch disconnects the matrix and buzzer from the main power rail to reduce **quiescent current**.
- When not in use, PIC MCU shuts off the load switch, puts the shift registers into a **high-impedance state** and then sleeps until awoken by an external interrupt generated by the button.