MAHIR MAHOTA

EDUCATION

University of Waterloo – Candidate for BASc in Mechatronics Engineering Sept. 2022 – May 2027 GPA: 3.99 (94.27% Cumulative) – Dean's Honours List and Academic Representative Relevant courses – MTE 262 (Digital Logic), MTE 140 (Data Structures and Algorithms), MTE 120 (Circuits)

SKILLS

Software: C, C++, Python, Linux, Bash, RTOS, ROS2, Git, GDB, CMake, MATLAB/Simulink, VHDL

Tools and Technologies: STM32, PIC, ESP32, Raspberry Pi, Arduino, OpenCV, TensorFlow, Pandas, NumPy

Electrical: Altium, Soldering, Oscilloscope, Logic Analyzer, DMM, Hot Air Reflow Station

Protocols: CAN, I2C, SPI, UART, SMBus/PMBus

EXPERIENCE

Embedded Software Developer

Christie Digital Systems

- Designed embedded software in C/C++ for the master control and regulation board on venue projectors
- Generated varying software PWM with a PIC24, enabling sinusoidal wave creation to drive piezo actuator
- Enabled manual control and display of peripheral wheel RPMs through the CLI, allowing quieter operation
- Updated laser calibration on start-up by modifying projector initialisation state, enabling colour filter use
- Wrote automation script in **Python** to parse schematic netlist files and error check 16K+ pin connections

Firmware Developer

onsemi

May 2023 - Aug. 2023

Jan. 2024 - Present

- Developed firmware in C/C++ for a multi-phase voltage controller, collaborating in an Agile environment
- Implemented a shared SMBus access layer in C using circular buffers, eliminating global variable reliance
- Multithreaded in C++ to test driver functionality concurrently, using semaphores for resource protection
- Verified I2C state machine transitions, using bit masking and bitwise operations to check register values
- Conducted tests for 15+ drivers and a PMBus library with 94% coverage, using GDB for debugging issues
- Edited CMake and JSON files to include tests so they could be built and then output results successfully

Firmware Team Lead

Waterloop

June 2023 - Present

- Directing 14 active members to develop software for a custom-built hyperloop pod used in competition
- Developed a CAN driver and config files for the NUCLEO-F767ZI to communicate using the STM32 HAL
- Created a KAC-8080N motor controller driver in C for closed-loop PID control of the LIM using a DAC
- Designed two-layer PCB in **Altium** to multiplex 48 thermistors, reducing **ADC** channels in **BMS** by 87.5%
- Ideated Python state machine architecture for main RPi to unpack CAN messages and send error codes
- Used DMA to gather data from ADC pin for motor thermistors and set watchdog timer to ensure control

PROJECTS

% Eye Controlled Trolley

- Programmed an ESP32 to drive 4 DC motors using L298N H-bridges, enabling speed control over Wi-Fi
- Interacted with AdHawk eye tracking glasses and Python API to monitor line of sight with 80% accuracy
- Processed MPU6050 gyroscope data to drive trolley with commands received from socket interfacing

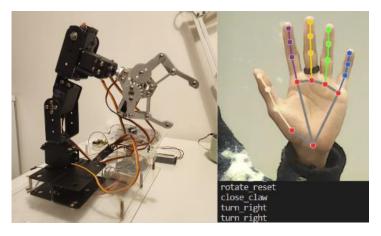
% OpenCV Robotic Arm

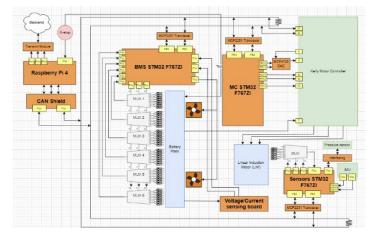
- Implemented C++ drivers for UART communication with the 6DOF arm through an HC-05 BT module
- Interfaced between Python program and Arduino using serial transmission to allow sending instructions
- Detected hand gestures with 21 coordinates generated using OpenCV and Google's MediaPipe library

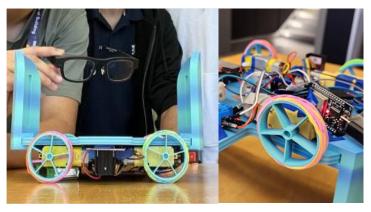
% Movie Reviews Discord Bot

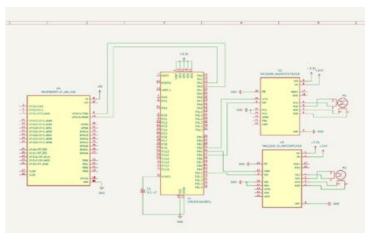
- Designed neural network with TensorFlow using text vectorisation to analyse reviews with 98% accuracy
- Created interactive bot in **Python** with the Discord API, analysing web-scraped articles using the model

PORTFOLIO









OpenCV Robotic Arm



- Transferred OpenCV hand landmark coordinates into an array to detect when finger or hand positioning changed. These defined servo positioning commands for the Arduino sent with the Bluetooth module.
- Wired six servo motors to properly support their stall current. Soldered servo control wires and the HC-05 to the microcontroller, enabling wireless communication in Python.

Waterloop Pod



- Led team to develop competition ready firmware for motor controller, BMS and sensor sub-systems to interface together.
- Developed CAN communication frameworks for STM32 7676ZI boards to send messages and warnings to a central Raspberry Pi.
- Worked on driver to control the 140V-600A motor controller through an external DAC.

Eye Controlled Trolley

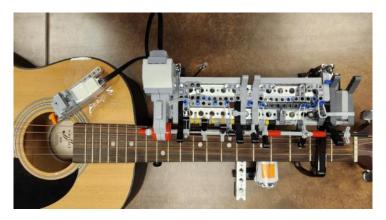


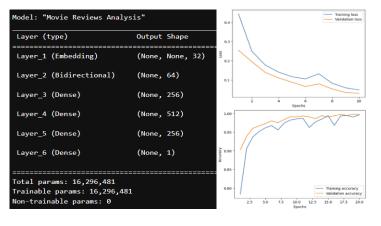
- Cleaned accelerometer and gyroscope data by polling at fixed intervals, averaging values and correcting for drift inaccuracies.
- Had the ESP32 drive four DC motors through two H-bridges, using the onboard WiFi module to send commands directly from the Python script wirelessly.

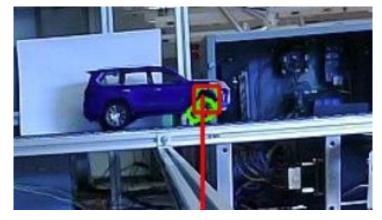
Brick Scanner



- Wrote object-oriented C++ driver for controlling a NEMA-17 stepper motor with an STM32 NUCLEO-F401RE. Added micro stepping capabilities for increased resolution. Two motor objects were defined to move a camera arm.
- Set up communication with RPI using UART and interrupts for important messages such as when the next measurement was urgent.







Guitar Playing Robot



- Wrote drivers in C for colour reading, fretting, and strumming. Used colour and ultrasonic sensors to detect progress of tasks. Designed for first year design project.
- Built camshaft system to press on the string at different frets depending on the angle a motor is rotated. The rotation of the motor was tracked using onboard motor encoders

Movie Reviews Discord Bot



- Created NLP model with TensorFlow using 50K IMDb reviews. Processed data by vectorising and creating a pipeline with separate training and validation sets.
- Interacted with the Discord API to get access to user messages. Programmed a web-scraping script that collects URLs from a Google query search and extracts text.

OpenCV Wheel Tracking



- Developed program in sub 12 hours for wheel detection instead of current standard of mechanical triggers, improving accuracy.
 Applied Hough transforms and contour detection snippets.
- Adapted depth camera for Python instead of C++ using WSL, allowing easier detection of stickered holes in the car chassis.