






# MAHIR MAHOTA

 [linkedin.com/in/mahir-mahota](https://www.linkedin.com/in/mahir-mahota)  [github.com/mahir-mahota](https://github.com/mahir-mahota)  [mahirmahota.com](https://mahirmahota.com)  
 [mmahota@uwaterloo.ca](mailto:mmahota@uwaterloo.ca)  +1 226-899-6987

## EDUCATION

**University of Waterloo** – Candidate for BAsC in Mechatronics Engineering Sept. 2022 – May 2027  
GPA: 3.99 (93.33% Cumulative) – Dean's Honours List and Academic Representative  
Relevant courses – MTE 241 (RTOS), MTE 220 (Sensors), MTE 140 (Data Structures and Algorithms)

## SKILLS

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

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

**Electrical:** Altium, Soldering (THT/SMD), Oscilloscope, Logic Analyzer, DMM, Hot Air Reflow Station

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

## WORK EXPERIENCE

**Firmware/Embedded Engineering Intern** Sept. 2024 – Present  
Siemens – Enlighted Inc

- Developing firmware in **C** and **Python** for IR sensors, communication gateways and management systems

**Embedded Software Developer Intern** Jan. 2024 – Apr. 2024  
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 dichroic wheel RPMs through a **CLI**, allowing for finer sound tuning
- 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 Intern** May 2023 – Aug. 2023  
onsemi

- 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

## DESIGN TEAMS

**Vehicle Platform Director** May 2024 – Present  
WATonomous

- Managing **15+** active members to develop hardware and firmware to turn a Kia Soul EV fully autonomous
- Developing **C++** interface layer to translate **ROS2** messages from the autonomy stack into **CAN** frames
- Integrating power systems infrastructure to connect the **sensor** and **compute** racks with the car battery
- Soldered and mounted boards to communicate with the **CAN** network, enabling remote joystick control

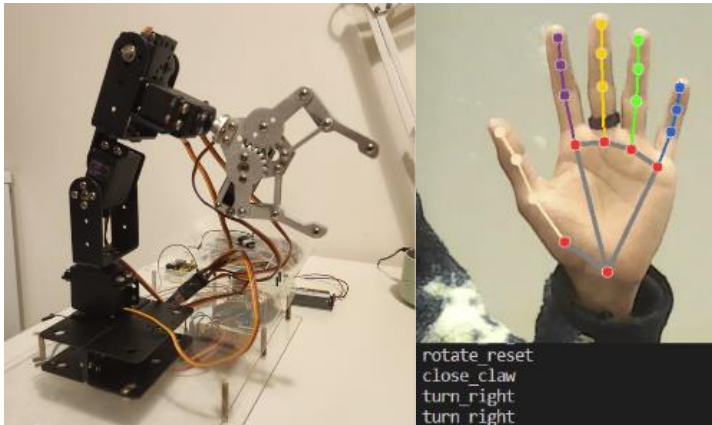
**Firmware Team Lead** Jun. 2023 – Aug. 2024  
Waterloop

- Directed **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

**Firmware Team Member** Nov. 2022 – Apr. 2023  
Waterloo Aerial Robotics Group

- Wrote motor test driver in **C** with a NUCLEO-F401RE for driving a servo by varying a potentiometer value
- Configured onboard timer for a set **PWM** duty cycle and implemented **SPI** communication with an **ADC**

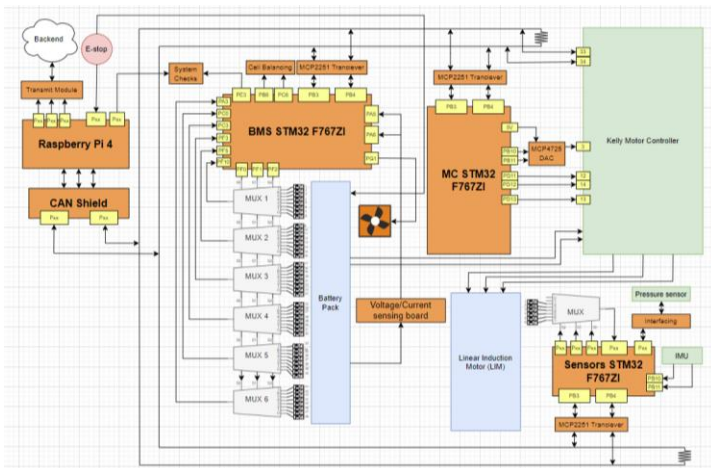
# PORTFOLIO



## OpenCV Robotic Arm



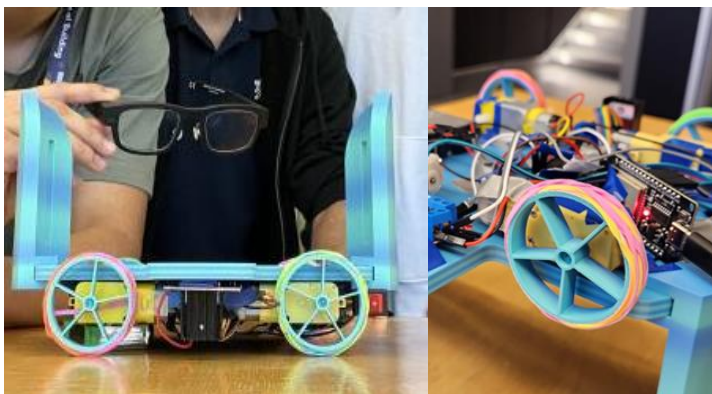
- Transferred OpenCV hand landmark coordinates into an array to detect when finger or hand positioning changed. These defined servo positioning commands for the Arduino sent with the Bluetooth module.
- Wired six servo motors to properly support their stall current. Soldered servo control wires and the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



## Waterloop Pod



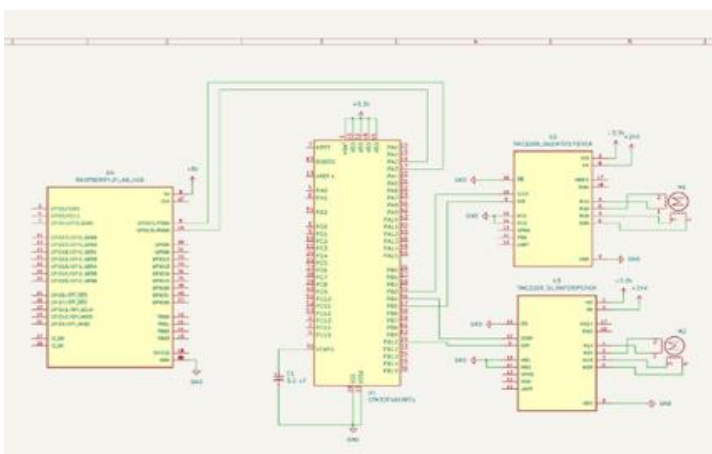
- Led team to develop competition ready firmware for motor controller, BMS and sensor sub-systems to interface together.
- Developed CAN communication frameworks for STM32 7676ZI boards to send messages and warnings to a central Raspberry Pi.
- Worked on driver to control the 140V-600A motor controller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

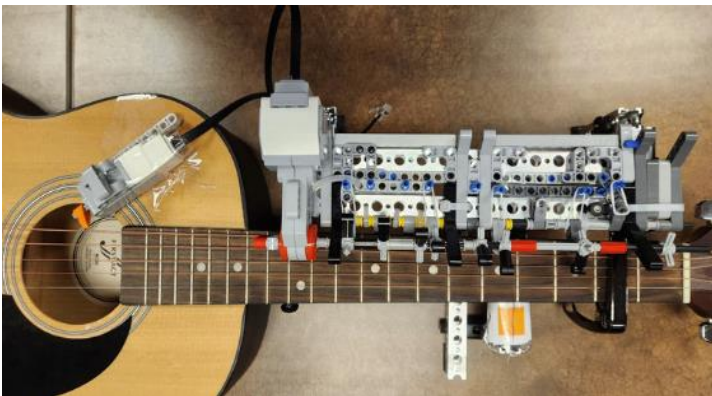


- Wrote object-oriented C++ driver for controlling a NEMA-17 stepper motor with an STM32 NUCLEO-F401RE. Added micro stepping capabilities for increased resolution. Two motor objects were defined to move a camera arm.
- Set up communication with RPI using UART and interrupts for important messages such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

## WATonomous Car



- Designed architecture for the physical interface and the ROS2 nodes allowing CAN communication with the Kia to allow for joystick and autonomous control.
- Implemented power system infrastructure to power the sensor and compute racks using the car battery and DC/AC inverters.



## Guitar Playing Robot

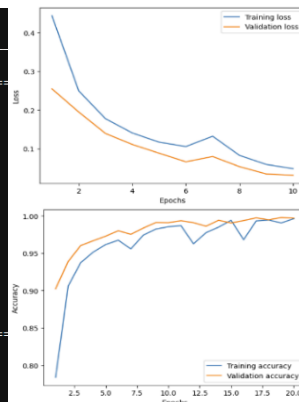


- Wrote drivers in C for colour reading, fretting, and strumming. Used colour and ultrasonic sensors to detect progress of tasks. Designed for first year design project.
- Built camshaft system to press on the string at different frets depending on the angle a motor is rotated. The rotation of the motor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

Model: "Movie Reviews Analysis"

Layer (type)	Output Shape
Layer_1 (Embedding)	(None, None, 32)
Layer_2 (Bidirectional)	(None, 64)
Layer_3 (Dense)	(None, 256)
Layer_4 (Dense)	(None, 512)
Layer_5 (Dense)	(None, 256)
Layer_6 (Dense)	(None, 1)

Total params: 16,296,481  
Trainable params: 16,296,481  
Non-trainable params: 0



## 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.