

# Joseph Mi

437-929-3943 | [jm.josephmi@gmail.com](mailto:jm.josephmi@gmail.com) | [linkedin.com/in/joseph-mi](https://linkedin.com/in/joseph-mi) | [github.com/Joseph-Mi](https://github.com/Joseph-Mi)

## EDUCATION

### University of Toronto

*Bachelor of Electrical and Computer Engineering, Minor in Robotics*

Toronto, ON

Sept. 2023 – May 2027

- **Deep Learning and App Development Researcher under Professor Ervin Sejdic, Vector Institute**
- Notable courses: Elec. & Mag. Fields 2, Signals and Systems, Data Structures and Algorithms, and Electronics, Circuits 1&2, PCB Design, Princeton Computer Architecture II via Coursera

## EXPERIENCE

### PCB and Network Design Researcher

*University of Toronto Network Lab*

Aug 2024 – Present

Toronto, ON

- Implemented **FreeRTOS task scheduling** into sensor node state to include receive and transmit threads
- Led **power testing using PPK2** to discover MOSFET leakage problem
- **Completed PCB schematic/layout** for ESP32 cotton candy ECU, including 4 I2C and UART sensors
- Optimized RabbitMQ messaging queue for Cotton Candy node
- Managing **RPi host state**, dependencies, messaging, data storage and node organization
- Exploring HULP library in C to base project on, instead of Assembly, to handle data upkeep during Deep Sleep

### University of Toronto Formula Racing

*Electrical and Firmware Engineer*

Dec 2023 – Present

Toronto, ON

- Implemented **Brake System Plausibility control** firmware to activate emergency shut down circuit
- Led **PCB bring up** and validated 16+ sensors, LCD, and CAN on 6 layer 200+ component custom Front ECU
- Designing **battery pack charge state monitoring**: CAN definitions, fault detection, and RT cell interfacing
- Implemented **EEPROM and Internal Board Temperature** readings onto Accum., Rear, and Front ECUs
- Developed BMS monitoring algorithms using CAN protocol, **reducing system faults by 30%**
- Contributing to **Driverless-Firmware integration**(ON/OFF global driverless states and task assignments via FreeRTOS) Driverless testing(CAN, DAQ, state validations)
- Contributed to **Low Voltage bench** development (Bosch CAN, sensors, and ECU)

### Mapping Software Developer

Dec 2023 - Present

- Implemented C# script to generate **Unity bezier splines from CSV files**, which hold the asphalt path
- Integrating **Lidar perception pipeline** within simulation
- Developed Docker container to integrate ROS endpoints, dependencies, and DV source code
- Managed **GPS readings using RTKlib** for vehicle mapping

### Full Stack Developer

*Confidens Analytics*

May 2024 – Sept 2024

North York, ON

- Developed **REST APIs using Laravel and MySQL** to manage user records, including handling Google and SpyFu API calls to retrieve competitor data within progressive radii
- Built service **purchasing pipeline** and optimized SQL database to record transactions
- Improved competitor analysis data requests and user display load times by **53%**

## PROJECTS

### CAPTR | C++, Platformio, Arduino, Data Processing, Onshape, Teensy, FreeRTOS

March 2023 – Present

- Developed **data logging thread** to retrieve/store gyro, baro, mag, acceleration, and timestamp data into flash
- Developed **schematic and layout** to include LDOs, Buck, UART, I2C, SPI sensors, RPP, and fire readiness
- Implementing **telemetry communication** and monitoring using RFM95W LoRa Transceiver
- Contributed to **Unscented Kalman Filter (UKF)** for state estimation and accept state updates

### DE1-SoC Vision | Verilog, ModelSim, Drivers, Tensorflow, Linux Development

Sept 2024 – Nov 2024

- Trained ASL **TFLite model** and integrated with **custom Linux Image**
- Managing **ARM Processor/FPGA interactions** and active task offloading
- Constructed sequential and combinatory digital circuits in Verilog, using **TCL scripts, Testbenches, and ModelSim for verification**
- Utilized **on-board RAM and ROM** to store I/O greyscale letter outputs
- Developed and integrated **VGA, PS2, and Audio drivers** for interfacing and functionalities
- Interlacing staggered Moore and Mealy FSMs into single state and overall software state management

## TECHNICAL SKILLS

**Languages:** Python, C/C++, C#, JavaScript, MATLAB, PHP, ROS2, SQL, Java, R

**Development Frameworks and Management:** React, Vue.js, Laravel, Bootstrap, React Native, NoSQL, MySQL

**Electrical Design and Hardware:** MCUs, RPi, Altium, Kicad, Verilog, System Verilog, ModelSim, Platformio

**Developer Tools:** Git, Github, Docker, BitBucket, Kubernetes, RabbitMQ

**Machine Learning and Data:** Tensorflow, CNNs, Kaggle, Google Colab, Jupyter, Matplotlib