MAYUKH GHOSH

+65 8168 0170 • mayukh.ghosh@u.nus.edu • ascendedsky.github.io• www.linkedin.com/in/mayukh-ghosh-

EDUCATION

National University of Singapore **Aug 2024 - Present**
Bachelor of Engineering (Honours)

- Major in Electrical Engineering, Second Major in Innovation and Design
- Expected Date of Graduation: May 2028
- GPA: 4.80/5.00
- Relevant course work: Electronic Circuits, Signals and Systems, Fundamentals of System Design, Electromagnetics, Microcontroller Programming and Interfacing, Electrical Energy Systems

WORK EXPERIENCE

NUS Engineering Design and Innovation Centre, Robotics Engineer **May 2025 - Aug 2025**

- Integrated Open-RMF middleware with two TurtleBot3 robots on Jazzy, establishing a robust fleet coordination framework with autonomous task execution, leveraging ROS2 topics, actions, and services for real-time multi-robot state synchronisation, building a reliable reference set-up for future student use
- Evaluated and bench marked three Navigation2 controllers with Free-Fleet adapter, enforcing lane discipline and improving multi-agent path consistency through detailed analysis of controller performance and robot motion behaviours
- Designed two Gazebo simulations of the EDIC building, accurately replicating real-world infrastructure to validate Open-RMF coordination workflows, sensor data flow, and autonomous navigation logic
- Created an ESP32-based door adapter through Micro-ROS, executing Wi-Fi transport at 10–20 Hz message rates with ROS2 pub/sub endpoints and real-time GPIO actuation with less than 300 ms latency to connect facility-management hardware into Open-RMF workflows

NUS Mars Rover, Communications and Global Navigation Satellite System Lead **Jan 2025 - Present**

- Implemented a point-to-point wireless communication network for the rover leveraging 2 high-power digital data links over UDP, and utilised FFmpeg to compress raw video data using H. 264 before transmission to attain low-latency, low-bandwidth, long-range video streaming between the rover and control station
- Devised the rover's dual-band 900 MHz / 2.4 GHz communication architecture through link-budget modelling, RF propagation analysis, and hardware evaluation, and engineered a stepper-motor control system to automatically steer the 900 MHz antenna, maintaining stable links beyond 2 km
- Developed a PyQt-based GUI to automate rover video-pipeline configuration via Paramiko SSH, reducing command modification and execution time to under 10 seconds and adding real-time telemetry displays (RSSI, latency, frame-drop metrics) with basic analytics to boost remote and autonomous situational awareness
- Integrated GNSS module into rover embedded system firmware, enabling real-time MapViz visualisation of position and trajectory, and programmed behavior-tree based autonomous recovery to the last recorded GNSS coordinates if signal is lost for more than 30 seconds

PROJECTS

Design of a 12U Lunar Cubesat **Aug 2025 - Nov 2025**

- Proposed mission concept for a polar lunar orbit focused on mapping volatile deposits in near-PSRs and penumbras through multispectral sensing
- Developed end-to-end system architecture, including payload integration, communications subsystem requirements, and thermal/mechanical constraints for a 12U platform
- Designed the electrical architecture, covering power budgeting, EPS layout, sensor interfaces, and redundancy strategies for deep-space operation
- Proposed an AI-based framework for spectrometer calibration-drift correction, power-fault prediction from onboard telemetry, and adaptive data compression for low-bandwidth down link

STM32 Microcontroller Game & IIR Filter Implementation **Sep 2025 - Nov 2025**

- Implemented real-time microcontroller game control and input handling, coordinating HAL, system clock, GPIO, ADC, timers, DMA, UART, and I²C to facilitate TIM16/TIM17-driven LED blinking, buzzer toggling, and environmental sensor monitoring
- Coded GPIO interrupt logic for single and rapid/slow double-press detection to switch or replay games, and configured DMA-based UART for non-blocking environmental status updates
- Constructed and optimised a 10th-order IIR filter in assembly, through circular buffers, modular arithmetic, and register reuse to reduce memory movement and execution cycles by >75% vs C, with scalable circular indexing and efficient arithmetic via MLA/MLS instructions

Active Edge Coverage for Multi-Robot Collaboration **Aug 2025 - Nov 2025**

- Devised and combined a novel goal selection algorithm for exploring edges in a graph-map with the Open-RMF framework, hence decreasing full map exploration time by 15% compared to frontier algorithms in a multi-robot environment
- Coded an obstacle detection system using a Raspberry Pi camera to dynamically open and close lanes in multi-robot maps with upto 70 nodes, enhancing collaborative navigation
- Programmed autonomous robot docking, utilising AprilTag detection via Raspberry Pi camera with ROS2 Navigation Docking Server and Open-RMF battery management to allow precise, real-time docking and charging coordination

Opamp-based Signal Processing and Filtering **Sep 2025 - Nov 2025**

- Designed and created a 3rd-order active Chebyshev band-stop filter using op-amps, accomplishing 10 dB audio noise reduction at the target frequency of 5 kHZ while preserving signal amplitude and minimising distortion
- Constructed a modulated signal transmission and reception pipeline, building a 4th-order Butterworth low-pass filter with a cutoff frequency at 937 Hz for demodulation, achieving accurate signal recovery with <2% amplitude error

Rescue TurtleBot3 with Thermal Imaging and Flare Mechanism **Jan 2025 - Apr 2025**

- Implemented autonomous navigation with a customised Navigation 2 Stack of the robot through the ROS 2 framework on Ubuntu 22.04 on Raspberry Pi
- Programmed a system coordinator Python programme along with a C++ Wavefront Frontier Detection algorithm, along with a backup navigation algorithm, thus improving system scalability and robustness
- Devised a modified electrical architecture for combining a thermal camera and 2 motors into existing TurtleBot3 hardware for mission success

AI Agent Based Pursuit Evasion Games **Dec 2024 - Jan 2025**

- Developed a classic pursuer-evader game through PyTorch by employing model-free training with multi-agent deep Q Neural Networks, achieving adaptive and competitive agent behaviour
- Decreased training time for convergence by 30% using prioritised experience replay and target Q networks

CERTIFICATES & SKILLS

- Proficient in Python, C++, Java, Bash, LaTeX, ROS2, PyTorch, Linux, Git
- Basic proficiency in Altium Designer, LTSpice, Fusion 360, MATLAB, GNU Octave, OpenCV, Docker, CI/CD, STM32, ESP32, UART, I²C, SPI, DMA, ARM Cortex-M, PCB debugging, oscilloscopes, MQTT, TCP/UDP networking, Quantum Computing (Qiskit)