SYSC 4907 – Fall 2026/Winter 2027

<table>
<tr><td>Supervisor</td><td>Richard Dansereau</td></tr>
<tr><td>Co-supervisor</td><td>Chao Shen</td></tr>
<tr><td>Course section</td><td>TBD</td></tr>
<tr><td>Project ID</td><td>TBD</td></tr>
<tr><td>Project title</td><td>Autonomous Vehicles</td></tr>
<tr><td>Project description</td><td>

This capstone project involves building and advancing the autonomous systems for **two real autonomous vehicle platforms** housed at Carleton:

1. **Ecolo ET-4-Cruise Conversion**
   A 4-wheeled electric bike originally designed for manual operation. We are transforming this platform into a **fully autonomous research vehicle**, making it an ideal hands-on capstone project for students interested in robotics, embedded systems, controls, and AI.

   

   Some remote-control functionality is already in place, providing a starting point for further development. Students will extend this foundation by designing, integrating, testing, and refining a broad range of autonomous vehicle subsystems.

2. **Aurrigo PodZero**
   A purpose-built autonomous shuttle platform donated to Carleton for research, teaching, and development. Used internationally for short-range autonomous transportation, the PodZero provides a robust mechanical foundation but requires a full redevelopment of its **software, sensing, and computing stack** to reach autonomous operation in our environment.

   

   This capstone project invites students to help rebuild the autonomous system from the ground up and offers an exceptional opportunity to work on an industry-grade autonomous vehicle platform.

Students will work in multidisciplinary teams spanning **software, hardware, artificial intelligence, electronics, embedded systems, controls, sensors, user interfaces, mapping, navigation, and simulation**. The scope is intentionally broad to allow groups to focus on components matching their interests and skill sets.

</td></tr>
</table>

	Possible Technical Areas and Subprojects Below is an expanded list of topics with concrete example tasks students could own. **1. Sensors and Perception** Students can work on selecting, configuring, calibrating, and integrating sensors. **Core Sensors** • **GNSS/GPS + RTK positioning** High-precision localization for autonomous navigation; integrating RTK corrections. • **mmWave radar** Short- and long-range obstacle detection; Doppler velocity detection; fusing radar with other sensors. • **3D LiDAR (e.g., Velodyne VLP-16)** Point cloud processing, object detection, segmentation, and mapping. • **Laser range sensors / proximity sensors** Short-range collision avoidance and docking. • **Camera systems (mono/stereo)** Vision processing, semantic segmentation, lane detection, and object tracking. • **IMUs / accelerometers / gyros** Sensor fusion for stable pose estimation. • **Wheel encoders** Dead-reckoning and motion modelling. **Example student deliverables** • Multi-sensor calibration pipeline • Perception algorithms for traffic lights, pedestrians, lane markings • Radar–LiDAR fusion for improved obstacle detection • Camera-based sign and marker detection **2. Computing and Embedded Software** Work on the edge-computing platforms inside both vehicles. **Computing Hardware** • **Raspberry Pi 4 and 5 boards** For sensor acquisition, auxiliary compute, low-level tasks. • **NVIDIA Jetson AGX Orin** High-performance GPU compute for deep learning and perception. • **Onboard networking (Ethernet, CAN, serial, USB)** **Software Systems** • **Linux / Ubuntu** administration • **ROS2** node development and system integration • **QNX / RTOS** for safety-critical subsystems • **AI/ML pipelines** for detection, classification, and tracking **Example student deliverables** • ROS2 architecture for all vehicle components • Real-time perception nodes on the Jetson platform • Docker-based development and deployment pipelines • Low-latency sensor drivers and communication interfaces

3. Electronics, Wiring, and Hardware Integration
Students can design and build the physical electrical systems supporting autonomy.
Tasks include

- Wiring harness design and installation
- High- and low-voltage routing and protection
- PCB design for sensor breakout boards
- Power distribution and noise mitigation
- Interfacing with motor controllers and steering actuators

Example deliverables

- Custom PCB for sensor integration
- Modular wiring harness for the Ecolo conversion
- Testing and debugging of electrical subsystems

4. Control Systems
Design the algorithms enabling the vehicles to move safely and smoothly.
Sub-areas

- **Motor control** (acceleration, torque management)
- **Steering and turning control**
- **Braking control**
- **Trajectory following**
- **Low-level control loops (PID, MPC, etc.)**
- **Ackerman steering**

Example deliverables

- Path-tracking with closed-loop feedback
- Braking control with emergency-stop logic
- Drive-by-wire subsystem for the Ecolo ET-4

5. Batteries and Power Systems
Students interested in power engineering can work on:

- Battery monitoring systems
    - State-of-charge (SOC) and cell health assessment
- Power conversion for computation hardware
    - DC–DC converters, voltage regulation, and distribution for onboard compute
- Charge state monitoring and balancing
- Safe shutdown and power sequencing

6. Remote Monitoring and Telemetry
Students can build dashboards and backend systems for monitoring vehicle state.
Components

- Live telemetry streaming
- Vehicle health reporting (faults, temperatures, voltages)
- Logging autonomous decisions for debugging
- Safety event recording
- Remote observer tools

Example deliverables

- Web dashboard showing real-time vehicle status
- Remote “mission control” interface

7. User Interface and Human–Machine Interaction
Work on interfaces that allow a rider or operator to:

- Enter destinations
- View vehicle status
- See warnings or planned routes
- Approve autonomous actions
- Perform testing and diagnostics

8. Mapping and Routing
Students interested in algorithms and spatial computing can develop:

- Global mapping and offline route planning
- Local real-time routing and obstacle negotiation
- SLAM (Simultaneous Localization and Mapping)
- Map storage and update systems

9. Safety and Collision Avoidance
Critical subsystems including:

- Redundant stop systems
- Safe state fallbacks
- Obstacle detection and stopping logic
- Risk assessment and decision logic

10. Digital Twin and Simulation Systems
Work on simulation environments to support development without needing the physical vehicles.
Simulation Tools

- **CARLA** autonomous vehicle simulator
- **Hardware-in-the-loop testing**
- **Remote activation & sensing pipelines**

Example deliverables

- Digital twin of the PodZero
- Simulation scenarios for autonomous testing
- Automated training pipelines for perception models

11. Real-Time Operating Systems (RTOS)
Certain low-level and safety-critical subsystems in an autonomous vehicle require deterministic timing and guaranteed response deadlines. Students interested in embedded systems and real-time robotics can work with **QNX, FreeRTOS, or other RTOS platforms** to implement these components.
Tasks include

- **Deterministic control loops**
    - Developing steering, throttle, and braking controllers that operate at fixed frequencies with minimal jitter.
- **Timing-critical sensor acquisition**
    - Ensuring LiDAR, radar, IMU, and encoder data are captured and delivered within strict deadlines for stable sensor fusion.
- **Safety and fail-safe mechanisms**
    - Implementing watchdog timers, heartbeat monitors, emergency-stop routines, and safe-state fallback logic that must run reliably under all conditions.
- **Real-time communication interfaces**

	◦ Developing CAN bus, UART, I^2C, and SPI drivers scheduled under RTOS constraints for predictable communication. • **RTOS–ROS2 integration** ◦ Building bridges where low-level real-time tasks run in QNX/FreeRTOS while high-level perception and planning run in Linux/ROS2. • **System performance benchmarking** ◦ Measuring latency, jitter, and task-execution timing and tuning the system for autonomous driving needs. **Example deliverables** Real-time control module managing steering, throttle, and braking • RTOS-based sensor driver for IMU or wheel encoders • Watchdog and emergency-stop safety subsystem • Hybrid RTOS–ROS2 communication architecture • Timing-analysis report comparing Linux vs. RTOS performance **What Students Will Gain** Students will graduate with real, hands-on experience in: • Autonomous vehicle design • Robotics and ROS2 development • Hardware–software integration • Real sensor data collection and processing • AI and computer vision • Control systems engineering • Testing and validation of safety-critical systems • Large-team engineering project management These are highly sought-after skills in automotive, robotics, aerospace, AI, and embedded systems industries. **Suggested Project Teams** Subteams of **2–6 students**, each focusing on a domain such as: • Perception and AI • Controls and drive-by-wire • Sensor integration and electronics • Systems and software • Mapping and localization • Simulation and digital twin • UI/telemetry and operator tools Teams will collaborate to build a complete autonomous vehicle software stack.
Program(s)	Computer Systems, Software, Electrical, (Tele)Communications, Mechanical
Maximum number of students	25
Meeting time with supervisor (optional)	
Do you want the student to contact you before the office assign this project to them ? (Yes/No)	Yes