

ROS 2. Odometry. LiDAR. SLAM. Nav2.

A university-level autonomous robotics journey built around ZAS Robotics Car-5 — from low-level encoder motor control to mapping, localisation, path planning and fully autonomous navigation.

ZAS ROBOTICS CAR-5

ROS2 Autonomy Stack + Dedicated Controller Architecture

Product platform: high-precision encoder drive · MPU6050 IMU · Raspberry Pi ROS2 compute · 2D LiDAR

CAR-5 PRODUCT IMAGE: use official ZAS Robotics website image in final artwork

YOUR HARDWARE, MAPPED TO THE CURRICULUM

Raspberry Pi 4 / 5

Ubuntu + ROS 2 compute for localisation, SLAM, planning and Nav2.

2× Encoder DC Motors

Wheel velocity feedback, closed-loop motion and differential-drive odometry.

MPU6050 IMU

Angular velocity and acceleration for motion/orientation estimation.

Dedicated Controller

Real-time motor control, encoder feedback and robust sensor interfacing.

TB6612 Motor Driver

Independent bidirectional left/right motor control.

2D LiDAR

360° ranging for obstacle perception, mapping and localisation.

THE PROMISE

Students do not simply learn ROS 2 commands. Every stage adds one working capability to the same Car-5 — motor control, encoder feedback, ROS 2 communication, odometry, LiDAR, SLAM, localisation and finally autonomous navigation.

Build the Robot Foundation

STAGES 01–05 · MAKE IT MOVE → MEASURE IT → CONNECT IT TO ROS 2

01 Differential Drive & Encoder Motors

Motor direction, PWM control, encoder ticks, wheel speed and differential-drive motion.

DEMO Command forward/reverse motion, left/right turns and an accurate 360° rotation using encoder feedback.

02 Closed-Loop Motor Control

Measure wheel velocity from encoders and introduce feedback/PID wheel-speed control. Explore why equal PWM does not always produce equal wheel speed.

DEMO Compare open-loop and encoder-controlled straight-line motion and measure trajectory improvement.

03 Robust Raspberry Pi <-> Controller Communication

Build a reliable command/status layer with packet framing, validation, heartbeat, watchdog and communication-loss safety.

DEMO Drive Car-5 from the Raspberry Pi, interrupt communication and verify automatic safe-stop behaviour.

04 ROS 2 Foundations

Learn nodes, topics, messages, publishers/subscribers and geometry_msgs/Twist through the physical robot. Introduce /cmd_vel as the standard mobile-robot velocity interface.

DEMO Publish linear and angular velocity commands from the ROS 2 terminal and watch Car-5 execute them.

05 ROS 2 Motor Bridge

Build the chain: /cmd_vel → differential-drive kinematics → wheel velocities → motor commands. Connect high-level ROS commands to physical wheel movement.

DEMO Command straight, curved and in-place rotational trajectories using /cmd_vel.

Give the Robot Position & Perception

STAGES 06–09 · KNOW WHERE IT IS → SEE THE WORLD → BUILD THE MAP

06 Encoder Odometry

Convert encoder ticks into wheel displacement and estimate robot pose (x, y, θ). Publish /odom and investigate dead-reckoning error.

DEMO Drive a measured square and compare the ROS 2 odometry trail with the robot's actual finishing position.

07 MPU6050 IMU & Robot Orientation

Publish ROS-compatible IMU data and explore angular velocity, acceleration, gyro bias, noise and drift. Compare inertial and encoder-based rotation estimates.

DEMO Rotate Car-5 through a measured angle and compare encoder-estimated heading with IMU-estimated heading.

08 TF & Robot Coordinate Frames

Build the frame structure required by autonomous navigation: map → odom → base_link → laser. Understand how sensors and robot pose relate in ROS 2.

DEMO Visualise Car-5 and its TF tree in RViz and verify that the LiDAR frame moves correctly with the robot.

09 LiDAR Integration & SLAM

Publish 2D laser scans on /scan, visualise obstacle geometry in RViz, then use SLAM to create a 2D occupancy-grid map of the environment.

DEMO Manually drive Car-5 around a laboratory/classroom and watch the map appear live in RViz; save the completed map.

AUTONOMY DATA FLOW

ENCODERS + MPU6050 + LiDAR → ODOMETRY + PERCEPTION → ROS 2 + TF → SLAM MAP

Make Car-5 Autonomous

STAGE 10 · LOCALISE → PLAN → AVOID → NAVIGATE

10 Localisation & Nav2 Autonomous Navigation

Launch Car-5 inside the saved map. Introduce localisation, global planning, local control, costmaps and obstacle avoidance. Connect the complete autonomous navigation pipeline.

DEMO Select a goal in RViz. Car-5 localises, calculates a route, moves autonomously, responds to an unexpected obstacle and reaches the destination.

CAPSTONE DEMO

Goal in RViz → Localisation → Global Path → Local Controller → /cmd_vel → Car-5

Students place an unexpected obstacle in the route and observe Nav2 dynamically respond while the robot continues toward its goal.

TRAINING FORMAT

Format	Suggested Duration	Team Size
10 progressive stages	20–30 hours hands-on	2–3 students per Car-5
Prerequisite	Platform	Outcome
Basic Python/C/C++ + Linux	Ubuntu 24.04 + ROS 2 Jazzy	Working autonomous navigation stack

STUDENT OUTPUTS

ROS 2 Car-5 workspace · robust motor bridge · encoder odometry · MPU6050 integration · LiDAR integration · TF tree · SLAM-generated map · Nav2 configuration · autonomous navigation demonstration

FROM /cmd_vel TO AUTONOMOUS NAVIGATION

Make it move → Measure its motion → Connect ROS 2 → Know where it is → See the world → Build a map → Navigate autonomously