

# ROS2 Mobile Manipulator

## Project Status Report

### Systems Audit Summary

February 10, 2026

### Contents

---

#### ==== EXECUTIVE SUMMARY ====

The repository has a functional base-motion stack (mecanum kinematics, wheel command bridge, encoder odometry), a defined robot model (base/arm/camera TF structure), and partial perception/manipulation components. However, end-to-end integration is incomplete: perception outputs are not fully consumed by the manipulation planner path, manipulation gating depends on a navigation status topic that is not clearly produced in the current stack, and the manipulation execution sequence remains partially stubbed.

#### ==== SUBSYSTEM STATUS TABLE ====

---

Subsystem	Status	Evidence
Base Motion Stack	COMPLETE	/cmd_vel converted to /wheel_commands; serial bridge exchanges commands/encoders with Arduino; odometry publishes /odom + TF.
Odometry	COMPLETE	Encoder odometry subscribes to /wheel_ticks, publishes /odom, and broadcasts odom to base_link TF.
Navigation (Nav2)	PARTIAL	Nav2 servers and lifecycle manager are launched; full integration with manipulation gating/status signaling is not evident.
Robot Description / TF	COMPLETE	URDF/xacro includes base, manipulator mount, arm joints, and camera_link.
Perception (Object/Table)	PARTIAL	Color+depth object detection and camera-to-base transform exist; table plane outputs exist; integration path to manipulation yaw input is incomplete.
Manipulation Decision Layer	PARTIAL	State machine consumes reachability/yaw and publishes decisions; decision logic present but depends on upstream topic availability.

Manipulation Execution Layer	<b>PARTIAL</b>	Executor sends serial commands to Arduino; descend/grasp/lift are explicitly stubbed.
Task Orchestration	<b>PARTIAL</b>	Task manager queues tasks and publishes feedback but simulates NAVIGATE/PICK/PLACE behavior rather than full subsystem control.

---

## ==== DATA FLOW ANALYSIS ====

### Observed Flow

- Perception publishes `/arm/object_pose_camera`.
- Pose transformer republishes `/arm/object_pose_base`.
- Readiness gate uses `/arm/object_pose_base + /navigation/status` to publish `/arm/object_reachable`.
- Arm state machine consumes `/arm/object_reachable + /arm/desired_base_yaw` and publishes `/arm/command_decision`.
- Manipulation executor consumes command decisions and yaw, then sends serial commands to Arduino.

### Where Flow Stops / Breaks

- Base yaw planner expects `/detected_object_position`, which is not the same interface as the existing perception object-pose topics.
- Readiness gating depends on `/navigation/status`; clear producer wiring is not evident in current integration path.
- Table detection outputs are not shown as consumed by navigation/manipulation logic.

## ==== WHAT IS DONE ====

- Base control, wheel serial bridge, and encoder odometry pipeline are implemented.
- Robot model and TF structure for base+arm+camera are defined.
- Perception nodes for color object detection, pose transform, and table plane detection exist.
- Manipulation state machine and serial executor infrastructure exist.

## ==== WHAT IS NOT DONE ====

- End-to-end perception-to-yaw-to-manipulation wiring is not fully aligned.
- Navigation/manipulation mutual exclusion is not conclusively enforced by a fully wired status source.
- Manipulation execution phases (descend, grasp, lift) remain stubbed.
- Some launch/runtime registrations appear inconsistent with executable availability.

## **==== NEXT REQUIRED STEP ====**

**Single integration step:** establish one authoritative producer for `/navigation/status` and connect it into the currently implemented readiness gate so manipulation permission is enforced only when navigation is stopped.