Jan Sistonen & Jermu Roivanen
20.02.2026

Technical plan: Automated Waste Sorter

1. Overall Architecture

The system follows a Sense-Think-Act pipeline. The waste is placed into view; the Pi identifies the material and triggers an actuator to move the item into the correct bin.

- **Perception:** USB Camera (Visual) + Ultrasonic Sensor (Distance/Presence detection).
- **Processing:** Raspberry Pi 5 running a Python-based inference engine.
- **Execution:** GPIO-linked actuators (Servos or Solenoids) to divert waste.

2. Component Specifications

Hardware

- **Controller:** Raspberry Pi 5.
- **Vision:** USB Webcam
- **Proximity:** Ultrasonic Sensor (to detect when an object is in the "sorting zone").
- **Actuators:** Servo or solenoid
- **Power: 5V/5A Power Supply** for the Pi and a servo/ solenoid driver

Software

- **OS:** Raspberry Pi OS (64-bit).
- **Languages: Python** for the ML/Logic and **sensors**
- **ROS – architecture** Components communicate with topics, services & nodes

3. Integration & Interfaces

- **Camera Pi:** USB 3.0 / MIPI CSI interface.
- **Ultrasonic Pi:** GPIO pins (Trigger/Echo).
- **Pi Actuators:** PWM (Pulse Width Modulation) signals via GPIO to a motor driver or servo.

4. Software & Machine Learning

- **Model:** We will use a **Pre-trained Model** (like Yolo) fine-tuned on a waste dataset (Plastic, Paper, Metal, Glass, etc.). If we have enough time, we might train our own model.
- **Logic:**
    1. Ultrasonic sensor detects an object.
    2. Camera captures a frame.
    3. ML Model returns a label/ classification (e.g., "Metal").
    4. Pi sends a signal to the Actuator to move to right bin.

5. Environment

- **Operational:** Indoor use with consistent lighting (ML models struggle with changing shadows). A stable power outlet is required for the Pi 5.
- **Development:** SSH remote connection to Pi (coding on your laptop while the code runs on the Pi).

6. Hardware Status & Needs

- **Have:** Raspberry Pi 5, USB Camera (Intel stereo cam)
- **Need:** Ultrasonic sensor, Motor Driver, sorting mechanism (bin cover flip waste to the right bin), and a more robust power solution than 9V batteries.

7. Project Management

- **Version Control:** Git/GitHub.
- **Change Management:** Weekly code reviews
- **Github Actions:** for automated test protocols before code commits

8. Test Plan

1. **Sensor Calibration:** Ensure the Ultrasonic sensor consistently triggers only when an object is present.
2. **Model performance review:** We will test different pre-trained models (YOLO, OpenCV, Lobe), measure the performance for computing speed and object detection accuracy
3. **Inference Accuracy:** Test the camera with 50 different items to calculate the Top 1 accuracy percentage.
4. **Latency Test:** Measure the time from "Detection" to "Actuation" (Target: < 2 seconds).

9. Group

1. **Members**: Jan Sistonen and Jermu Roivanen.
2. **Roles:** No divided roles, doing together all parts of the project.
3. **Estimated time:** Scheduled proximately 10 hours per week (per member)