



Intelligent IoT- Based Smart Recycling Bin

Student: Eftali Note

Course: BSc Applied Computing

Supervisor: Fahad Ahmet



Introduction

- Recycling contamination remains a significant environmental challenge, as recyclable materials are often disposed of incorrectly, reducing the effectiveness of recycling systems. Existing smart bins primarily monitor bin fill levels but do not automatically identify different types of waste. [1], [2], [3]
- This project proposes an **Intelligent IoT-Based Smart Recycling Bin** that combines **Artificial Intelligence (AI)**, **computer vision**, and **Internet of Things (IoT)** technologies to automatically classify waste, open the correct recycling compartment, monitor the bin's fill level, and record recycling activity in real time. The aim is to improve recycling accuracy, reduce contamination, and promote more sustainable waste management.



Motivation

One day, while throwing away a plastic bottle, I noticed that several people had placed coffee cups, food containers, and general waste into the recycling bin. Although the bin was clearly labelled, many items had been disposed of incorrectly. This made me realise that even when people want to recycle, they are often unsure which bin to use. As a result, recyclable materials become contaminated and may end up in landfill.

This inspired me to think about how technology could help solve this problem. Instead of relying on people to decide where waste belongs, I wanted to design a smart recycling bin that could identify the waste automatically using Artificial Intelligence and open the correct compartment for disposal.

As a Computing student, I also wanted to challenge myself by combining several technologies I have learned throughout my degree, including Artificial Intelligence, computer vision, IoT, embedded systems, and Python programming. This project allows me to apply these skills to develop a practical solution that could improve recycling accuracy and support more sustainable waste management.



Problem Statement

Current Challenges

- **Recycling contamination** – Incorrect waste disposal contaminates recyclable materials, reducing recycling efficiency. [4].
- **Human error** – Many people are unsure which items can be recycled, leading to incorrect waste segregation.
- **Low recycling efficiency** – Contaminated recyclable waste often cannot be processed and is diverted to landfill.
- **Limited functionality of existing smart bins** – Most existing smart bins only monitor bin fill levels and do not identify or classify waste automatically.

UK Statistics

- **64.1%** of UK packaging waste was recycled in **2024**.
- Around **82%** of UK households place at least one item in recycling that is **not accepted locally**, contributing to recycling contamination.



Aim & Objectives

Aim

To design and develop an **Intelligent IoT-Based Smart Recycling Bin** that uses **Artificial Intelligence (AI)** and **Internet of Things (IoT)** technologies to automatically classify waste, improve recycling accuracy, and reduce contamination .

Objectives

- Design and develop the smart recycling bin prototype.
- Develop an AI model for automatic waste classification.
- Integrate the Raspberry Pi [5], Arduino [6]., sensors, and servo motor.
- Develop an IoT dashboard to monitor recycling activity and bin status.
- Evaluate the system based on classification accuracy, response time, and reliability.



Project Innovation

Captures image of waste item for AI classification.

- Runs AI model
- Image processing
- Sends commands to Arduino
- Stores data / connects to dashboard

Detects presence of object
in front of the bin.

Open and close the correct lid/compartment.

Controls servo motors
and reads sensor data
from Raspberry Pi.

Powers Raspberry Pi and Arduino.

HC-SR04 → ARDUINO UNO		SERVO MOTORS → ARDUINO UNO		RASPBERRY PI 4 → ARDUINO UNO	
VCC	→ 5V	VCC (Red)	→ 5V	GPIO 17 (TXD)	→ RX (Arduino)
Trig	→ D9	GND (Brown/Black)	→ GND	GPIO 18 (RXD)	→ TX (Arduino)
Echo	→ D10	Signal (Orange/Yellow)	→ D11 (Servo 1)	GND	→ GND
GND	→ GND		D12 (Servo 2)		

```

graph TD
    A[5V Power Supply] --> B[Raspberry Pi 4 (via USB-C)]
    B --> C[Arduino Uno (via USB or Vin)]
    C --> D[Common Ground (GND)]

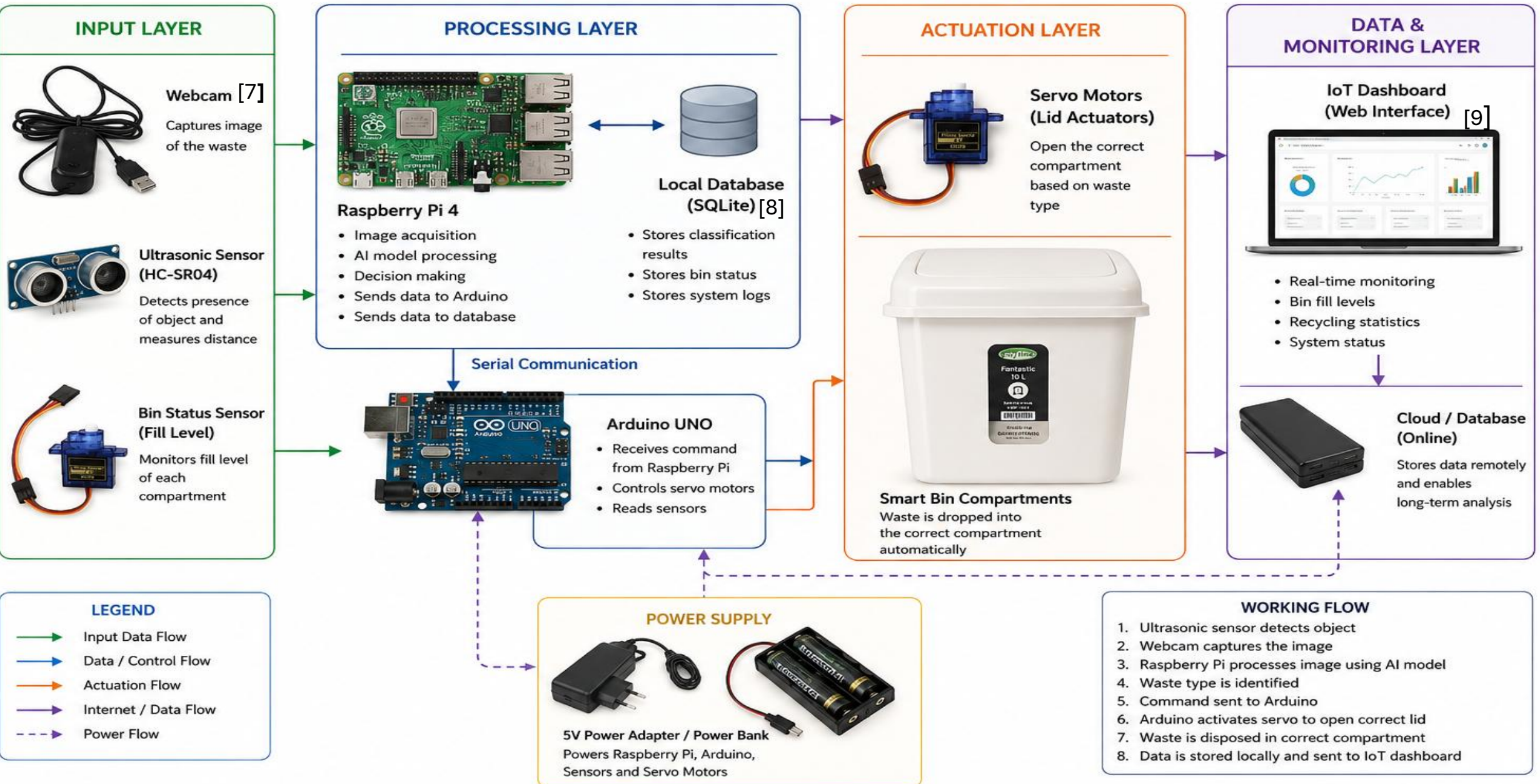
```

- 1 Ultrasonic sensor detects object
- 2 Camera captures image
- 3 Raspberry Pi processes image using AI
- 4 Result sent to Arduino
- 5 Arduino activates servo to open correct lid
- 6 Data stored in database & sent to dashboard

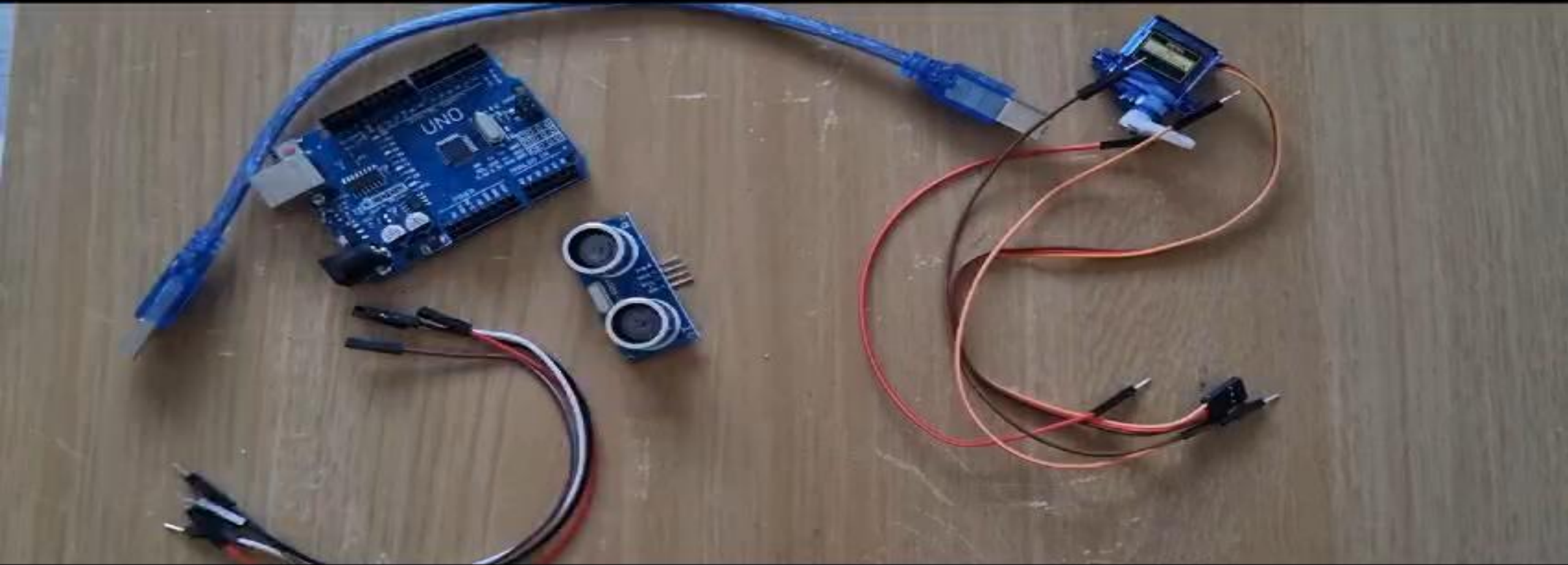
AI Classification
Auto lid opening
IoT Dashboard
Data Logging

SYSTEM ARCHITECTURE

Intelligent IoT-Based Smart Recycling Bin



AUTOMATIC SWING LID



Arduino Connection

sketch_jun28a | Arduino IDE 2.3.10

File Edit Sketch Tools Help

Arduino UNO

BOARDS MANAGER

Filter your search...

Type: All

Arduino AVR Boards by Arduino

1.8.8 installed

Boards included in this package: Arduino Industrial 101, Arduino UNO Mini, Arduino Nano,...

More info

1.8.8 REMOVE

Arduino ESP32 Boards by Arduino

Boards included in this package: Arduino Nano ESP32

More info

2.0.18- INSTALL

Arduino Mbed OS Edge Boards by Arduino

Boards included in this package:

sketch_jun28a.ino

```
1 #include <Servo.h>
2
3 Servo binServo;
4
5 // Pins
6 const int trigPin = 9;
7 const int echoPin = 10;
8 const int servoPin = 6;
9
10 // Variables
11 long duration;
12 int distance;
13
14 void setup()
15 {
16     Serial.begin(9600);
17
18     pinMode(trigPin, OUTPUT);
19     pinMode(echoPin, INPUT);
20
21     binServo.attach(servoPin);
22
23     // Lid closed
24     binServo.write(0);
25 }
```

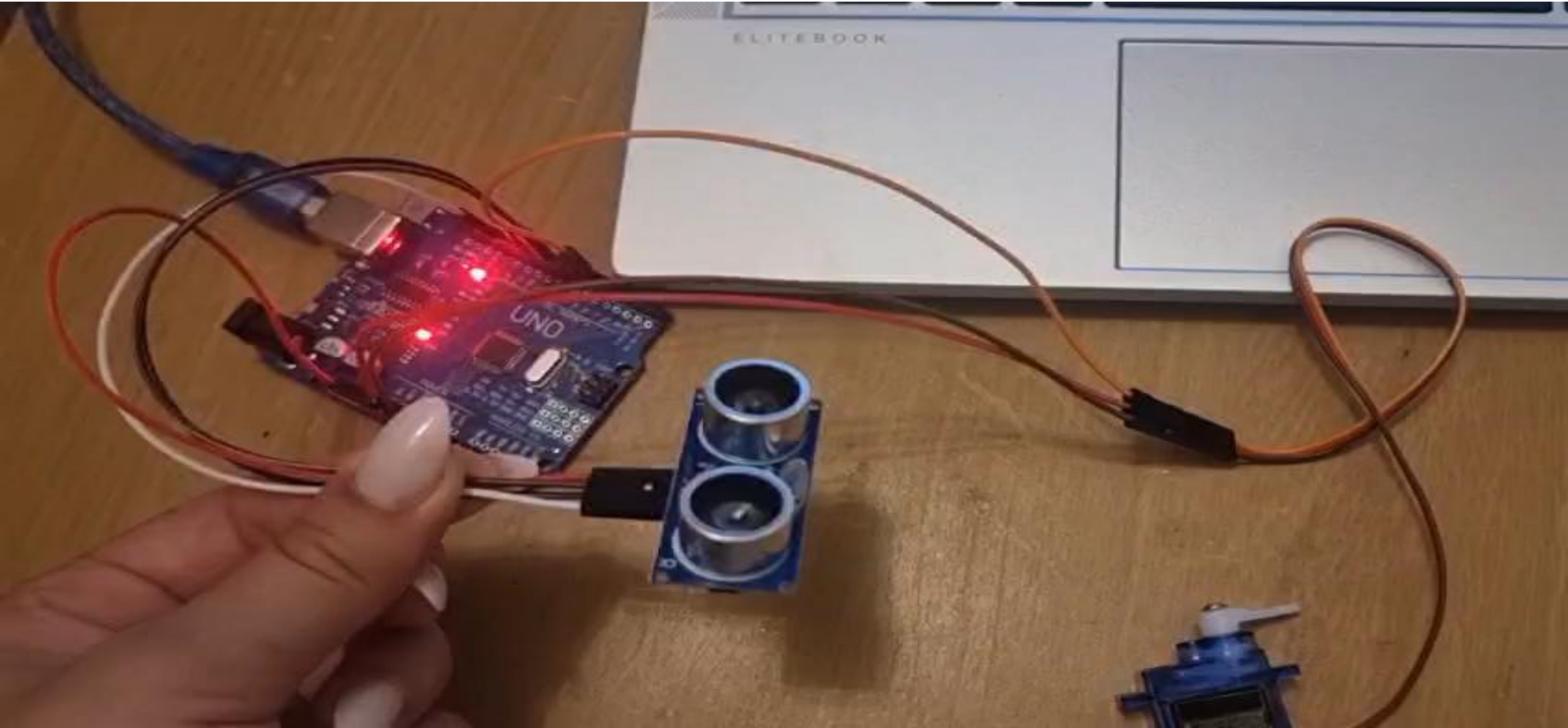
Output

Global variables use 269 bytes (13%) of dynamic memory, leaving 1779 bytes for local variables. Maximum is 2048 bytes.

Ln 11, Col 15 Arduino UNO on COM8 [not connected] 3

22:00 02/07/2026

Automatic Bin Lid



SWING LID BIN – SERVO INSTALLATION GUIDE

FOR LIDS THAT SWING (LIKE THE BIN SHOWN)

1. OVERVIEW – INSIDE VIEW

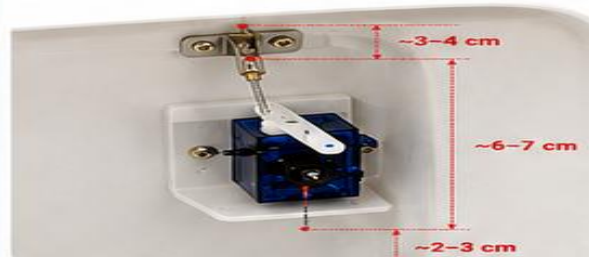


2. LID ARM BRACKET (ON LID – INSIDE)



Fix the flat lid arm bracket on the inside of the lid, near the top center, using 2 screws.

3. SERVO MOUNTING (ON BACK WALL – INSIDE)



Mount the servo on the back wall using the L-bracket and 2 screws. Keep it high enough so the lid can swing freely.

PARTS YOU NEED



RECOMMENDED POSITIONS

- Keep the servo high and centered to allow full swing motion.
- The higher the servo, the larger the opening angle.
- Ensure the linkage rod moves straight without hitting the bin wall.

4. HOW IT WORKS

When the servo rotates, the linkage rod pulls the lid arm, causing the swing lid to open. When the servo returns, the lid swings closed by gravity.

CLOSED (SERVO 0°)

OPEN (SERVO ~90°)



💡 You can adjust the opening angle in code (e.g. 70°, 90°, 100°, 110°, 120°) to get the best opening.

5. LINKAGE CONNECTION (DETAIL)



⚠ Use a tight but smooth connection so the lid moves freely without too much force.

6. WIRING REMINDER



🔌 Use a stable 5V supply.
For best results, power the servo from an external 5V supply if you are also using a Raspberry Pi.

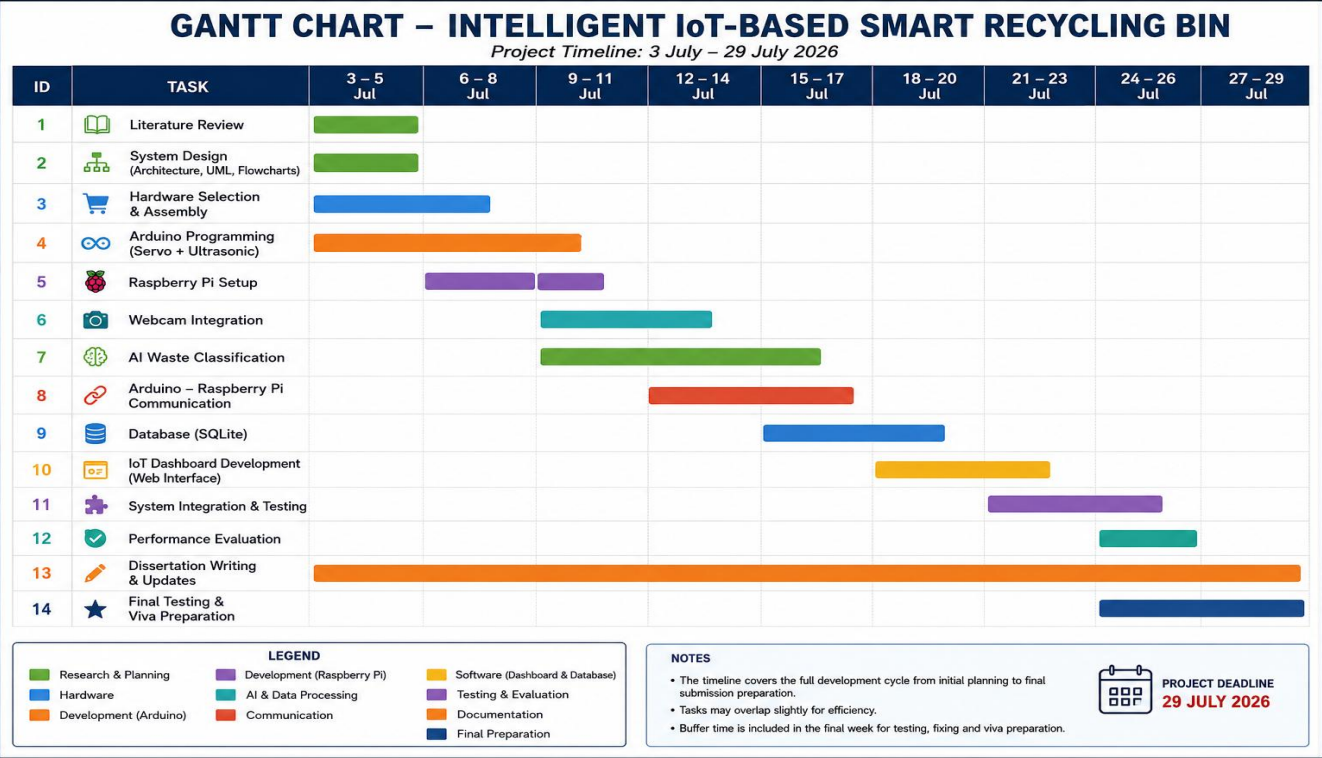
7. SIDE VIEW (CROSS SECTION)



TIPS

- ✓ Make sure the lid swings freely by hand before final fixing.
- ✓ If the lid opens too little, increase the angle in code (e.g. write(110) or write(120)).
- ✓ If the lid opens too much or hits the back, reduce the angle.
- ✓ Keep the electronics away from water and moisture.

Final Development Flow



A realistic timeline to finish by 29th of July

Task	Estimated time
Finish Arduino lid system	Done / 1 day
Install Raspberry Pi OS	1 day
Connect webcam	1 day
Python camera script	1–2 days
Train or use a waste classification AI model	2–3 days
Connect Raspberry Pi → Arduino	1 day
Store results in SQLite	1 day
Build a simple dashboard	2–3 days
Test the whole system	2 days
Fix bugs	2–3 days
Final report writing and screenshots	4–5 days

TO DO

- ✓

Automatic lid opens (Completed)
- 🔄

Raspberry Pi captures image
- 🔄

AI classifies waste
- 🔄

Raspberry Pi sends result to Arduino
- 🔄

Arduino opens the correct recycling compartment
- 🔄

Data is stored in SQLite
- 🔄

Dashboard displays recycling statistics
- 🔄

Fill-level sensors monitor each compartment
- 🔄

Cloud synchronises data (optional)
- 🔄

Final testing and evaluation

References

- [1] L. Atzori, A. Iera and G. Morabito, "The Internet of Things: A Survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
- [2] S. Madakam, R. Ramaswamy and S. Tripathi, "Internet of Things (IoT): A Literature Review," *Journal of Computer and Communications*, vol. 3, no. 5, pp. 164–173, 2015.
- [3] H. Boyes, B. Hallaq, J. Cunningham and T. Watson, "The Industrial Internet of Things (IIoT): An Analysis Framework," *Computers in Industry*, vol. 101, pp. 1–12, 2018.
- [4] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [5] Raspberry Pi Foundation, *Raspberry Pi Documentation*. [Online]. Available: <https://www.raspberrypi.com/documentation/>. [Accessed: Jul. 2026].
- [6] Arduino, *Arduino Uno Rev3 Documentation*. [Online]. Available: <https://docs.arduino.cc/hardware/uno-rev3>. [Accessed: Jul. 2026].
- [7] OpenCV, *Open Source Computer Vision Library*. [Online]. Available: <https://opencv.org/>. [Accessed: Jul. 2026].
- [8] TensorFlow, *TensorFlow Documentation*. [Online]. Available: <https://www.tensorflow.org/>. [Accessed: Jul. 2026].
- [9] SQLite Consortium, *SQLite Documentation*. [Online]. Available: <https://www.sqlite.org/docs.html>. [Accessed: Jul. 2026].



THANK YOU!

Any questions?

Eftali Note
University of Wales Trinity Saint David,
London

