Testing

- Outputs
- Logic discussion
- Refinements

1. Preparation

Before running tests:

- Disconnect the **motor's food dispenser part** so it spins freely without dropping food.
- Have your **Pi powered on** and the code saved (e.g., `pet_feeder.py`).
- Open a terminal on the Pi so you can run:

```
python3 pet_feeder.py
```

- Keep a **multimeter** or basic tester ready to check wiring if something doesn't respond.

---

2. Step-by-Step Testing Plan

Step 1 — Sensor Calibration

Goal: Make sure the hopper distance sensor and bowl weight sensor are giving realistic numbers.

Hopper Sensor Test

```
print("Hopper distance:", read_hopper_distance(), "cm")
```

- Put your hand inside the hopper → distance should shrink.
- Remove your hand → distance should increase.

Load Cell Test

```
print("Bowl weight:", read_bowl_weight(), "grams")
```

- Place an empty bowl → should be near `0 g`.
- Place 50g of rice/dry beans → should show close to 50g.
- Adjust `hx.set_reference_unit()` until weights look correct.

Step 2 — Motor Test (No Food)

Goal: Verify that the motor spins when commanded.

In Python shell:

```
dispense_food(2)  # should spin for 2 seconds
```

- Motor should turn smoothly.
- If it doesn't spin:
  - Check wiring from Pi → motor driver → motor.
  - Check power supply voltage (motors need more than Pi's 3.3V).

Step 3 — Alert Test

Goal: Ensure the buzzer and logging work.

In Python shell:

```
alert_staff("Test alert")
```

- Should beep and write a log entry to `/home/pi/pet_feeder_log.txt`.

Step 4 — Hopper Empty Detection

Goal: Simulate an empty hopper.

- Point the hopper sensor into open space (no food).
- Run the code and confirm it logs “Hopper empty!” instead of trying to feed.

Step 5 — Jam Simulation

Goal: Make sure retry logic works.

- Block the bowl sensor with something so it always reads “empty” even after dispensing.
- The code should:
    - Try to dispense 3 times.
    - Then give a “Dispense jam detected” alert.

Step 6 — Full Dry Run

Goal: Run the system without food but with all components active.

- Set `FEED_TIME` in the code to 1–2 minutes from now.
- Run the program.
- Watch the sequence:
    - Time reached → hopper check → motor spin → weight check → retry if needed.

Step 7 — Real Food Test

Goal: Feed actual dry kibble.

- Attach the dispensing mechanism to the motor.
- Fill hopper with enough food.
- Run at scheduled time.
- Ensure the weight change is detected.

Refinement Tips

- **Timing Adjustments**
  - If too much food drops → reduce `seconds` in `dispense_food()`.
  - If too little food drops → increase it.
- **Sensor Sensitivity**
  - If hopper says “empty” when it’s not → adjust `HOPPER_MIN_DISTANCE`.
  - If bowl says “empty” when it’s not → adjust `BOWL_EMPTY_WEIGHT`.
- **Logging for Debugging**
  - Check `/home/pi/pet_feeder_log.txt` after each run to verify sequence.
  - Add more `print()` statements if you want live feedback.
- **Dashboard Integration**
  - Replace `log_event()` with an MQTT or HTTP POST to send logs to your dashboard for real-time monitoring.