

Word Algorithm (Step 4)

Automatic Pet Feeder: Step-by-Step Operational Algorithm

Step 1: System Initialization

The system begins by setting up all key components — the real-time clock, servo motor, food storage sensor, and bowl weight sensor.

Comment: This ensures that the device knows the time, can dispense food when required, confirm that the hopper has food available, and measure the weight of food in the bowl.

Step 2: Verify Feeding Schedule or Manual Request

- `current_time = get_current_time()`
- If `current_time` equals a programmed schedule (e.g., 8:00 AM or 6:00 PM), OR if a manual feed button is pressed, move to Step 3.
- Otherwise, the system loops back and continues waiting.
- Comment: Feeding is triggered either by pre-set times or user input, while at all other times the system remains idle.

Step 3: Confirm Food Availability

- If `food_storage_sensor == FULL`, continue to Step 4.
- If not, send a “Refill Food” alert and return to the monitoring state.
- Comment: Prevents the motor from running unnecessarily when no food is available and alerts staff to restock.

Step 4: Dispense Food and Record Initial Bowl Weight

- `rotate_servo(3 seconds)`
- `bowl_weight_before = get_bowl_weight()`
- Comment: The servo operates for around three seconds to release a fixed portion of food. Immediately after, the bowl’s baseline weight is recorded for later comparison.

Step 5: Wait and Measure Bowl Again

- Wait 10 minutes.
- `bowl_weight_after = get_bowl_weight()`
- `weight_difference = bowl_weight_before – bowl_weight_after` Comment: A decrease in weight suggests the pet has eaten some food.

Step 6: Assess Result and Notify Staff

- If `weight_difference >= 5g`, feeding is considered successful.

- Otherwise, trigger an “Uneaten Food” alert. Comment: The 5g threshold is used to judge whether the food has been consumed, though this may not work well for smaller pets.

Step 7: Return to Monitoring State After evaluation

- the system returns to standby mode and continues monitoring for the next feeding schedule or manual request.
- Comment: This looping design ensures continuous operation without interruption.