**PART 1: Solving Problem Process**

**A. Understand and Define the problem:**

1. **Problem statement:** An animal shelter needs assistance in designing a low-cost and automatic scheduled pet feeder for cats and dogs. The system can check if the pet has eaten and can also alert the staff member when there are issues with food not being dispensed properly or there is not enough in the bowl.
2. **Features of the Feeder:**

* Scheduled food dispensing for cats and dogs
* Monitoring amount of food consumption (in container and on bowl)
* Alerting staff on system issues

1. **Inputs and outputs:**

* Input:
  + Feeding times (from a clock/timer)
  + Food level from sensor (in container)
  + Food weight from scale (in bowl)
* Output:
  + Food dispensing signal to servo motor
  + Alert from sensor to notify staff when issue occurs

1. **Assumptions and limitations:**

* Electricity supply is stable
* Sensors are consistently monitoring correctly
* There is only one food type in the dispenser
* The food container sensor is weight-based and food level is considered to be Low if there is less than 150g of food in the container
* The staff are available to troubleshoot for the system and feed the pet manually when nessesary

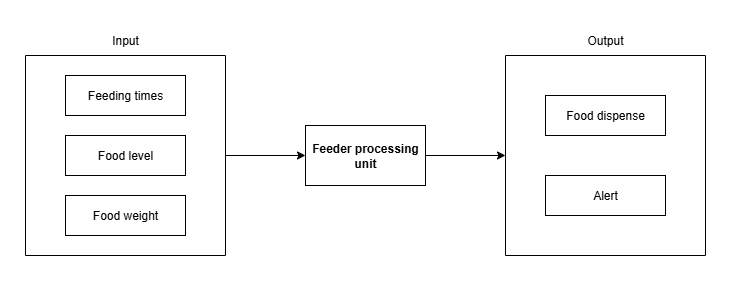


Figure 1. Block diagram of the system.

**B. Organise and Describe the Data**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data** | **Data type** | **Data description** | **Sample values** | **Operational constraints** |
| Clock | Input | Provides current time | 08:00, 18:00 | Trigger scheduled feeding time |
| Food level sensor | Input | Detects food level in the container | Low, Not Low | When detect Low level of food, send out alert |
| Food weight sensor | Input | Measures the weight of food in the feeding bowl | 0g, 50g, 100g | Detect when the food is dispensed or has the pet eaten the food |
| Servo motor | Output | Dispenses portion of food | ON, OFF | Turn on to open the food dispenser |
| Alert | Output | Sends out alerts to staff | ON, OFF | Notify staff of errors like empty container and jammed motor |

**C. Plan the solution**

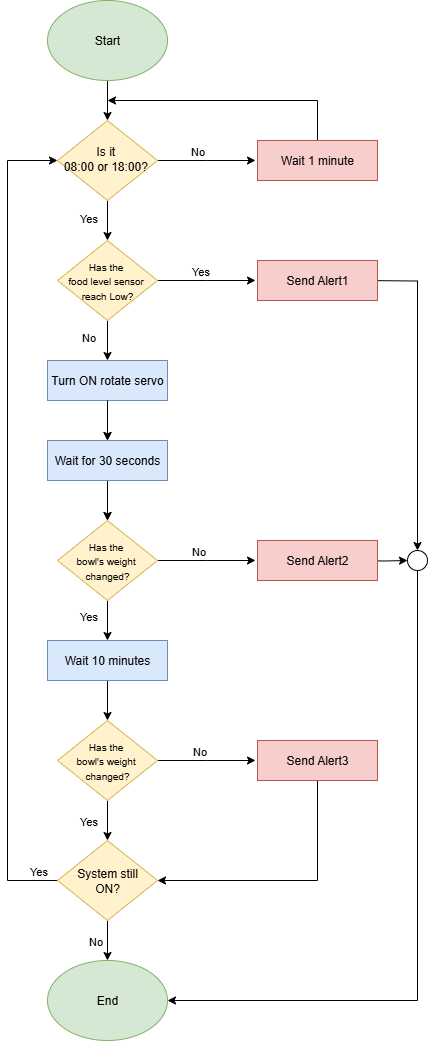


Figure 2. Flow chart logic for food dispenser

**D. Implement the Solution**

Variable names:

|  |  |  |
| --- | --- | --- |
| **Variables** | **Names** | **Unit** |
| Current time on the clock | CurrentTime | Hour (24-hour format) |
| Feeding time 1 | FeedingTime1 | 08:00 |
| Feeding time 2 | FeedingTime2 | 18:00 |
| Feeding portion | FoodPortion | 100g |
| Food level in container | FoodLevel | Not Low, Low |
| Alert “Food bin is Empty” | Alert1 | Food bin is Empty |
| Alert “Food failed to dispense” | Alert2 | Food failed to dispense |
| Alert “Food has not been eaten” | Alert3 | Food has not been eaten |
| Bowl weight before dispense | PreDispenseBowlWeight | g (gram) |
| Bowl weight after dispense | PostDispenseBowlWeight | g (gram |
| Bowl weight 10 minutes after dispense | FinalBowlWeight | g (gram |

* Check for CurrentTime on the clock:
  + If the clock CurrentTime is 08:00 or 18:00, then the rotate servo will turn ON and dispense the food.
  + If the clock CurrentTime is not 08:00 or 18:00, then wait for 1 minute and repeat the loop.
* Check for FoodLevel sensor of the container:
  + If FoodLevel is Low, then send out Alert1.
  + If FoodLevel is not Low, measure PreDispenseBowlWeight then Turn ON the rotate servo to dispense food.
* Wait for 30 seconds for food dispensing process to finish, then measure PostDispenseBowlWeight, then:
  + If PostDispenseBowlWeight is less than (PreDispenseBowlWeight+FoodPortion), then send out Alert2.
  + Else, wait for 10 minutes, then measure FinalBowlWeight, if FinalBowlWeight is equal to PostDispenseBowlWeight, send out Alert3.
* Check for system status:
  + If the system is still ON, repeat the process by returning to Check for CurrentTime
  + If the system is not ON, End the process

**E. Test and Refine the Solution**

* Scenario 1: Pet eats as expected
  + Inputs:
    - CurrentTime is 08:00
    - FoodLevel is not Low
    - PreDispenseBowlWeight is 0g
  + Process:
    - System dispenses 100g of FoodPortion
    - After 10 minutes, FinalBowlWeight is less than 100g
  + Outputs:
    - No alerts sent
    - Pets are fed normally
    - Process completed
* Scenario 2: Pet does not eat
  + Inputs:
    - CurrentTime is 08:00
    - FoodLevel is not Low
    - PreDispenseBowlWeight is 0g
  + Process:
    - System dispenses 100g of FoodPortion
    - After 10 minutes, FinalBowlWeight is still 100g
  + Outputs:
    - Alert3 sent out
    - Pet did not eat the food
    - Process completed with Alert3 sent out
* Scenario 3: Food bin is empty
  + Inputs:
    - CurrentTime is 08:00
    - FoodLevel is Low
  + Process:
    - System check with sensor before dispensing food
  + Outputs:
    - Alert2 sent out
    - Food did not dispense
    - Process did not complete with Alert2 sent out

**Suggest improvements:**

* **Food consumption tracking log:** To track pet’s eating amount and habit to monitor their health condition.
* **Food portion adjustment:** To change portion bigger/smaller depends on pet’s need, age, food type.
* **Tolerance margin for weighing:** To avoid Alert2 being sent out by food spilling out or pet eating right after food was dispensed from the container.

**PART 3: AI Agent Integration**

From my Copilot interaction, I asked it to Refine my Word code logic and implementing the system in the real world. For the first prompt, Copilot pointed out that my time check system could potentially be looped if it starts and ended in the same minute. However, I have implemented a 10-minute wait time to check for food amount after feeding the pets, so the proposed potential issue is not a problem here. Also, it suggested me to add a margin of error for the sensor to activate and check for the food eaten multiple times after dispensing the food so that the system would not send out false alarm too many times. For the second prompt, it suggested that by using Raspberry Pi, a real-time digital clock, weight sensor, and a power supply, my system can realistically be implemented. Both prompts have assisted me in shaping up my system, I find the second prompt being more useful to me as hardware implementation is something that I am currently more curious about. I have since added the solution to my Suggest improvement part of this assignment, though not directly integrated into the setup of the system because I still think my implementation makes this device work more efficiently with rational steps. Moreover, from the interaction in my first prompt, Copilot still shows some limitation in understanding and giving suggestions to my system. Finally, I asked Copilot to write a README.md file for my GitHub page as well, it made the page better than I currently can!

A white background with black text

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.