## Step 1: Understand and Define the Problem (Analyse)

*Deliverable: Clear problem statement, assumptions, inputs/outputs, and a simple sketch or block diagram of the system.*

This project is for creating a automated pet feeder that has customisable feeding amounts and feeding times. This must be low cost to produce and requires input sensors such as weight sensors, food level sensors, Realtime clock and battery sensors. It also provides outputs such as turning the servo and alert notifications.

## Step 2: Organise and Describe the Data

*Deliverable: Data table with inputs/outputs and operational parameters.*

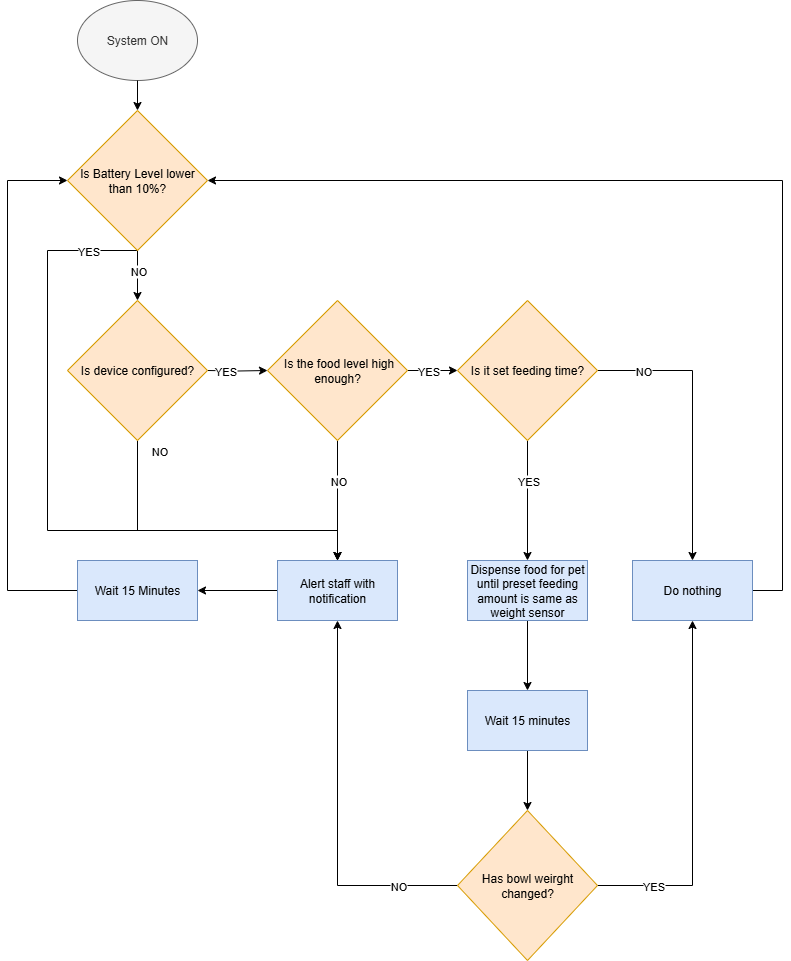
|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Unit | Example |
| Device Configurations | Input | Hours and Minutes/Integer | 8:00 100g |
| Weight sensor | Input | Integer | 100g |
| Level Sensor | Input | Boolean | TRUE |
| Battery Sensor | Input | Boolean | FALSE |
| Realtime Clock | Input | Hours and Minutes | 8:25 |
| Servo | Output | Integer | 45⁰ |
| Alert Notification | Output | String | "Please Configure Device" |

## Step 3: Plan the Solution (Design the Algorithm)

*Create decision logic for dispensing food. Add logic to detect errors.*

1. If battery level sensor is not positive send alert and wait 15 minutes.
2. If feeding times/amounts not configured send alert and wait 15 minutes.
3. If food level sensor is not positive send alert and wait 15 minutes.
4. Check if it is set feeding time. If not loop back to start.
5. If it is set feeding time dispense food until preset feeding amount matches current result.
6. Wait 15 minutes
7. If bowls weight has not changed send alert and wait 15 minutes.

*Deliverable: A flowchart representing your automated pet feeder logic (exported from Draw.io and included in your assignment report). You must also include in your submission the actual Draw.io file.*



## Step 4: Implement the Solution (Word Coding)

*Deliverable: Sequence of tasks with suitable explanations.*

SystemOn

Loop every minute (

if BatteryLevel > 10%

if DeviceConfigured == True

if FoodLevel == True

if RealTime == SetFeedingTime

DispenseFood

BowlWeightBefore = weight of bowl now

Wait 30 minutes

BowlWeightAfter = weight of bowl now

if BowlWeightBefore = BowlWeightAfter

Finish

else

SendAlert("Please fill Device")

Wait 15 minutes

else

SendAlert("Please Configure Device")

Wait 15 minutes

else

SendAlert("Battery Level Low")

Wait 15 minutes

)

## Step 5: Test and Refine the Solution (Debug and Verify)

*Deliverable: Test outputs, discussion of logic, and system refinements.*

If the pet eats as expected no alert will be sent.

If the pet does not eat an alert will be sent once.

If the food bin is empty an alert will be sent every 15 minutes.

This is all expected

An improvement that I would suggest would be to change the food empty alert to a longer wait, to avoid spamming staff devices.

**Your Challenge PART 2: On the Use of Technology.**

To promote professional reporting and personal development practices, you are now required to use [**GitHub**](https://github.com/) to manage your project files and collaborate efficiently when needed.

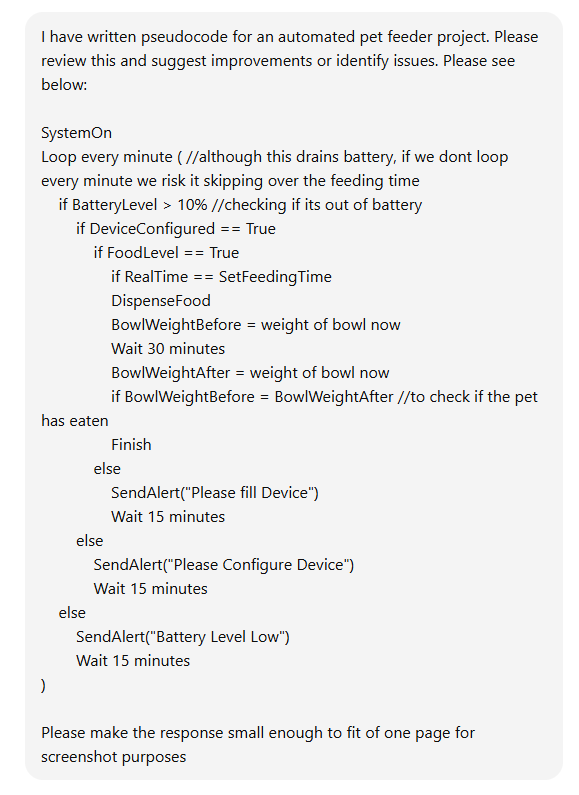
**Submission Requirement:**

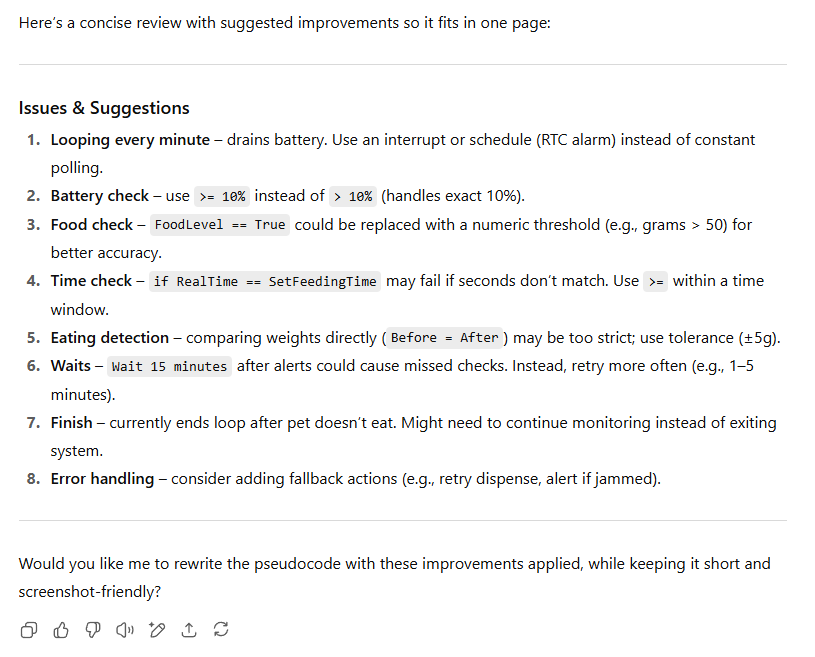
* Include the GitHub repository link in your report.

[dubsinnit-cyber/auto-pet-feeder: An auto pet feeder project.](https://github.com/dubsinnit-cyber/auto-pet-feeder/tree/main)

# Your Challenge PART 3: On AI Agent Integration

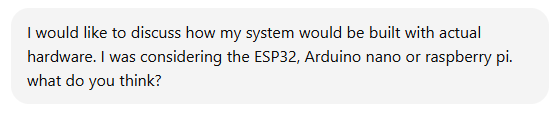
1. **Refine your logic or Word Code**: Ask Copilot to review your Step 4 implementation and suggest improvements or identify potential issues.

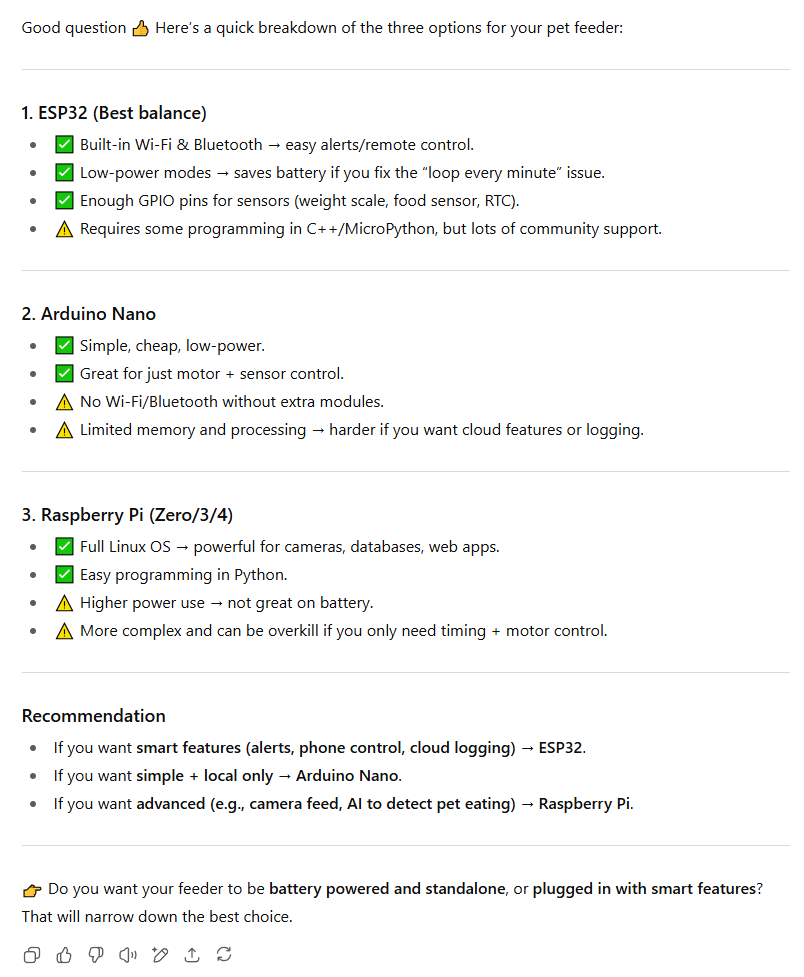




I asked ChatGPT for improvements on my pseudocode and was surprised by how well it analysed and suggested concise corrections. The first point it brought up is that looping every minute would drain battery, which I was aware of and commented in the code but did not know how to solve at the time. It also mentioned some threshold improvements to avoid miss confirmations. It also mentioned that the wait 15 minutes could cause missed checks and while I agree, I added this to avoid spamming the staff members device with notifications every minute. the last things it suggested was that if the pet doesn’t eat the loop ends, which I didn’t expect. I assumed if anything in the loop ends it goes back to the start and repeats. ChatGPT also brought up some errors that I had not considered before. This made me understand that there are a number of improvements that I could have made and that with external input I can greatly improve my system. An important lesson in collaboration.

1. **Explore real-world implementation**: Use Copilot to discuss how your system could be built using actual hardware (e.g., Arduino, Raspberry Pi).





I decided to act ChatGPT how I could build my device in real life. I already had some knowledge about creating IOT projects so I asked for a comparison between the Arduino nano, raspberry pi and ESP32. It mentioned that the Arduino doesn’t have Wi-Fi/Bluetooth capabilities without an extra module which would be an issue for production costs. It also mentioned that the raspberry pi would be overkill for this simple of a project and would have a higher power usage rate, which is a dealbreaker as the code itself right now would drain battery and it needs to be battery operatable. It presented The ESP32 as the best choice for me as it has built in wifi/Bluetooth, needs little power to run and has support for everything I need. Also given that the ESP32 is much cheaper than the other options which supports my low production cost limitation I would pursue that route compared to the alternatives.