**Step 4: 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