

Project #5: Smart Automatic Cat Feeder with Adaptive Behavior and Monitoring

Project Description

This project aims to design and implement a smart automatic feeder system for cats using an Arduino Nano. It supports both scheduled and manual feeding modes, provides interactive feedback via an OLED display and buzzer, and adapts feeding behavior based on historical data. It also includes logging and optional real-time bowl monitoring for overfeeding prevention.

Estimated Item List

- **1x Arduino Nano** – main controller (use ESP32 instead)
- **1x OLED Display** – status and interaction
- **1x Slide Switch** – to toggle feeding mode
- **1x Servo Motor** – to control feeder mechanism (might need relay instead)
- **1x Potentiometer** - to control kibble amount
- **1x Keypad** - to set feeding time in scheduled mode (use serial input instead)
- **1x Push Button** – manual feed trigger
- **2x Weight Sensors (HX711 + load cell)** – monitor food amount in tank and on plate
- **1x RGB LED** – system status indication (use regular/built-in LED instead of RGB)
- **Wires, resistors, etc.**

Software Specifications / Feature List

Core Functional Features

1. Feeding Modes

- **Scheduled Mode:** Dispenses food at fixed intervals (2 portions a day). Feeding time is set by the user using the keypad. A day cycle is defined as 2 min (for easy debugging and displaying).
- **Manual Mode:** Dispenses food on button press. The amount of food will be decided by a potentiometer from min to max per portion (30g-75g).

2. Screen

- Welcome screen with mode and kibble amount info.
- Current time (clock) and next feed time.

3. Servo Control

- Opens feeder lid for the amount of kibble set by the potentiometer per dispense cycle.

4. Eating mechanism

- Implement a mechanism for the pet “eating” to reduce the amount of kibble in the bowl.

Difficulty levels: (implement all)

Level 1 – Feeding History & Adaptive Behavior

- Feeding events (timestamp, quantity) will be shown in the serial interface in the following format:

Feeding: <Timestamp>, <Mode>, <Quantity>

- System analyzes last 24h usage and dynamically adjusts next feeding time or amount (e.g., reduce feeding if bowl is above pre-decided percentage).
 - If the bowl is still full at the end of the day (2 min cycle), reduce to portion by 1.
 - If the bowl is emptied by the first 5 seconds, increase the portion by 1.
- Presents trends or alerts on the OLED if the cat didn't eat in order to indicate an illness or abnormal behavior (less than 0.5 portion a day).

Level 2 – Safety & Monitoring with Weight Sensor

- The weight sensor under the bowl detects food level:
 - Prevents dispensing if the bowl is still full, to prevent overflow.
 - Indicates how long the bowl is empty to indicate underfeeding.
- Logs food consumption per event and shows trends over time, and shows on the serial interface.
- The second weight sensor will trigger an alert when the main food reservoir is at 25% capacity.

Add setup of a MQTT server and publish major events until the simulation is over.

What Makes This Project Challenging

- **Multi-modal Behavior:** Real-time mode switching, input sampling, and servo control coordination.
- **Adaptive Logic:** Requires implementing decision-making based on past behavior and real-time input (sensor data + logs).
- **Concurrency:** Periodic scheduled tasks must run alongside manual interaction with timing precision.
- **Memory Management:** Logging, RTC handling, and weight sensor integration on a limited-memory microcontroller.
- **Error Handling:** Must detect and recover from hardware or feeding anomalies. For example keep the current state and perform reset.