**Food Dispenser for Animals**

1. **Introduction**

**Purpose and Motivation**

The purpose of this project is to develop a cost-effective, automated pet feeder using the ESP32 microcontroller. Feeding pets on a timely basis is a common challenge for pet owners, especially when they are not home for extended periods. This project addresses this issue by automating the feeding process while also allowing remote control and monitoring via a web interface. The motivation behind this project stems from the need for a flexible, customizable solution that is more affordable than commercial automated feeders while offering greater functionality.

**Scope**

This project provides the following features:

* Automated food dispensing controlled by a timer.
* Real-time monitoring of food levels using an infrared (IR) sensor.
* A web-based interface to control the feeder remotely and check food status.
* Visual feedback through an LCD display to show feeding schedules and food status.

This system is designed to be user-friendly and can be easily adapted to different use cases, such as feeding small animals or managing food portions for multiple pets.

1. **Bibliographic Research**

**Existing Solutions**

Several automated pet feeders are available in the market, but they often lack customization and are expensive. DIY Arduino-based solutions exist, but they require significant technical expertise to implement. This project leverages the ESP32 microcontroller's Wi-Fi capabilities to create a feeder that is both low-cost and highly customizable.

**Example: "Easy-to-Build Pet Feeder" Project**

One notable example is the "Easy-to-Build Pet Feeder" project on Arduino Project Hub ([source](https://projecthub.arduino.cc/edr1924/easy-to-build-pet-feeder-0a4493)). This project uses an Arduino Uno to control a servo motor for dispensing food. The feeder includes:

* **Manual Button Control**: Users press a button to dispense food.
* **Servo Motor Mechanism**: Used for precise food dispensing.
* **No Internet Connectivity**: The feeder does not support remote control or monitoring.

**Comparison of Solutions**

| **Feature** | **"Easy-to-Build Pet Feeder"** | **DIY ESP32 Feeder** |
| --- | --- | --- |
| Cost | Low | Low (<$50) |
| Customizability | Moderate | High |
| Remote Control Capability | No | Yes |
| Food Monitoring | No | Yes |
| Internet Connectivity | No | Yes |
| Difficulty to Implement | Low | Moderate |

The ESP32-based feeder offers significant advantages, including remote control, food monitoring, and internet connectivity, making it a more versatile solution compared to the "Easy-to-Build Pet Feeder."

1. **Proposed Solution and Implementation**

**Overall Description**

The proposed solution is an automated pet feeder controlled by an ESP32 microcontroller. The feeder uses a stepper motor to dispense food at regular intervals or on demand. An IR sensor detects the food level in the bowl, while an LCD provides real-time feedback. The system includes a web-based interface for remote control and monitoring, allowing users to rotate the motor or check the food status through a browser.

**Hardware Design**

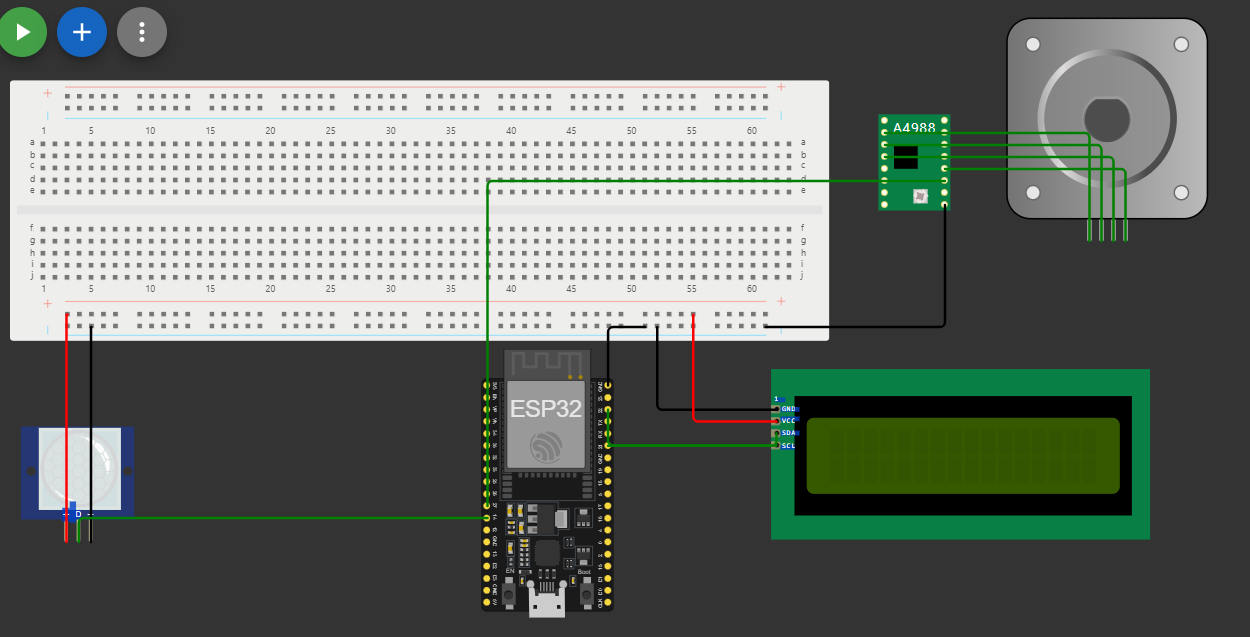
The hardware components include:

1. **ESP32 microcontroller**: Handles Wi-Fi connectivity, motor control, and web server functionality.
2. **Stepper Motor**: Dispenses food by rotating for a specified duration.
3. **IR Sensor**: Monitors food levels in the bowl.
4. **LiquidCrystal I2C LCD**: Displays feeding schedule and food status.
5. **Power Supply**: Provides power to the ESP32 and other components.
6. **Other Components**: Breadboard, jumper wires, and resistors.

**Circuit Diagram**

The circuit connects the ESP32 to the stepper motor, IR sensor, and LCD via the following pins:

* **Stepper Motor**: Pins 27, 26, 25, and 33.
* **IR Sensor**: Pin 14.
* **LCD**: SDA (Pin 21) and SCL (Pin 22).



**Software Implementation**

The software includes the following key components:

1. **Web Server**:
   * Hosts a web page with buttons for controlling the motor and displaying food status.
   * Uses HTTP requests to trigger motor rotation or update food level status.
2. **Motor Control**:
   * Implements stepper motor rotation using a state machine.
   * Allows the motor to rotate for 2 seconds when triggered.
3. **Timer Functionality**:
   * Checks if 12 hours have passed since the last feeding.
   * Automatically triggers the motor to dispense food at the scheduled time.
4. **Food Monitoring**:
   * Reads data from the IR sensor to determine if food is present.
   * Updates the LCD and web interface with the food status.
5. **LCD Updates**:
   * Displays the time remaining until the next feeding.
   * Shows real-time food level information.

**Debugging**

The debugging process involved:

* Testing motor timing to ensure precise food dispensing.
* Calibrating the IR sensor for accurate food detection.
* Verifying Wi-Fi connectivity and web server response times.

1. **Testing and Validation**

**Problems and Solutions**

* **Motor Alignment Issues**:
  + Problem: Inconsistent motor rotation caused uneven food dispensing.
  + Solution: Adjusted motor timing and added delays to ensure proper alignment.
* **IR Sensor Calibration**:
  + Problem: False readings when food bowl was partially filled.
  + Solution: Adjusted sensor position and added software filtering to improve accuracy.
* **Wi-Fi Connectivity**:
  + Problem: Occasional disconnects from the access point.
  + Solution: Improved error handling in the code to reconnect automatically.

**Test Results**

| **Test Scenario** | **Expected Result** | **Actual Result** | **Notes** |
| --- | --- | --- | --- |
| Motor rotation timing | 2 seconds | 2 seconds | Worked as expected |
| Food detection (with food) | IR sensor reads LOW | IR sensor reads LOW | No issues |
| Wi-Fi connectivity | Stable web server | Stable web server | Worked as expected |

**Final Adjustments**

* Improved the LCD display layout for better readability.
* Fine-tuned motor control to minimize power consumption.

1. **Conclusion**

**Purpose Fulfillment**

The automated pet feeder successfully fulfills its purpose of providing a cost-effective, customizable solution for automated feeding. It ensures timely feeding, monitors food levels, and offers remote control through a web interface.

**Impact**

This project can help pet owners manage feeding schedules more efficiently, especially when they are away from home. Its modular design allows it to be adapted for different types of animals and feeding requirements.

**Future Improvements**

* **Mobile App Integration**: Develop a dedicated mobile app for better user experience.
* **Battery Backup**: Add a backup power source to ensure functionality during power outages.
* **Enhanced Sensors**: Integrate additional sensors, such as weight sensors, to measure the exact amount of food dispensed.

By addressing these improvements, the feeder can become even more robust and user-friendly, making it a practical solution for a wider audience.