

Treat Dispenser

Viktor and Assan

Not actual project

>

3/18/2025



INTRODUCTION

We'd like to introduce the pet treat dispenser. The problem being addressed is the inability of pet owners to reward their pets remotely when they are not home. This project aims to develop a smart pet treat dispenser controlled via a website interface hosted on a Raspberry Pi.







Objectives/Features

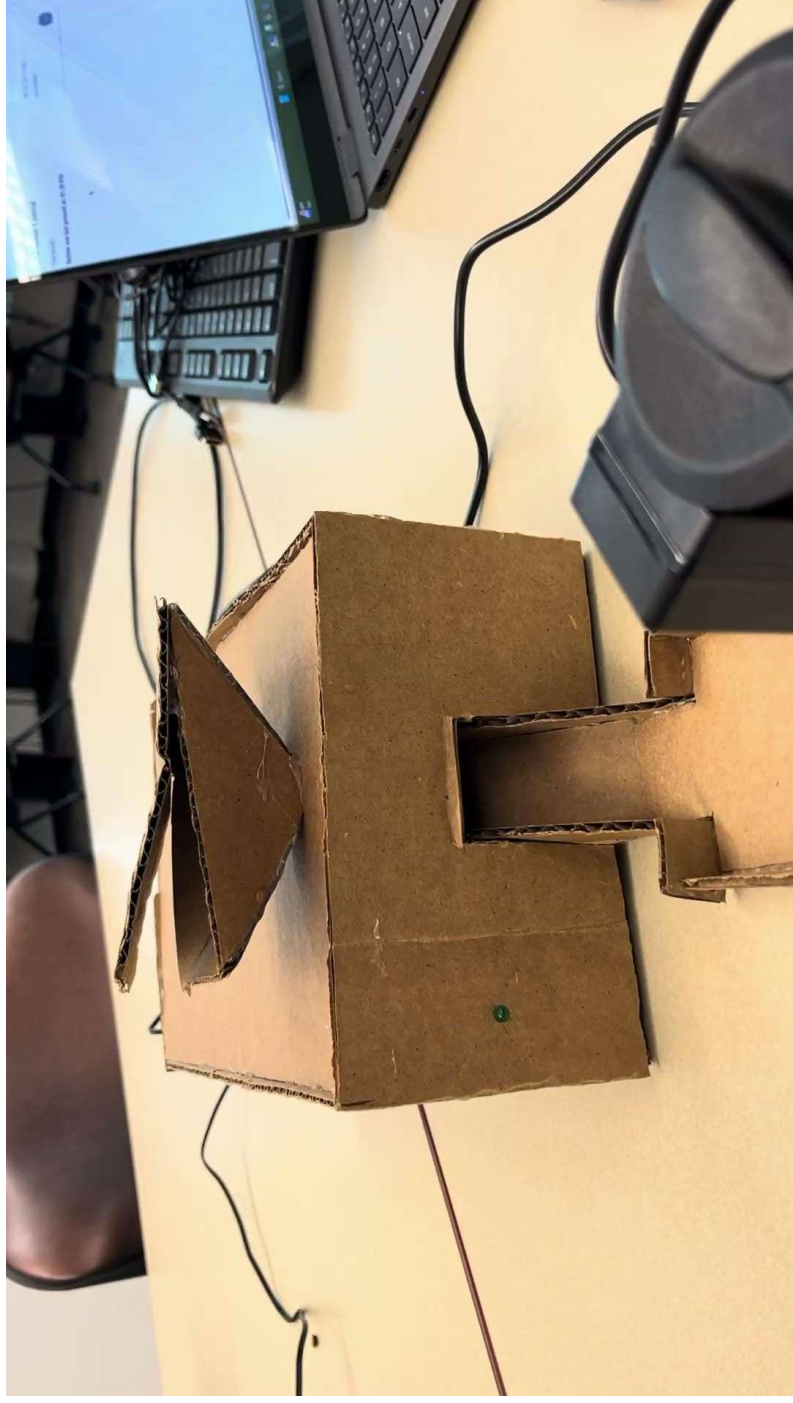
Our solution provides an interactive system where a pet owner can access a website from any device, press a button, and dispense a treat for their pet. The Raspberry Pi acts as the central controller, receiving commands from the website and triggering a motorized dispensing mechanism.



Goals

-  **1** Smart Pet Treat Dispenser
-  **2** Remotely controlled through a website
-  **3** Last time treat was dispensed on the site and LED when treat is dispensed
-  **4** Servoblaster code for smooth servo control

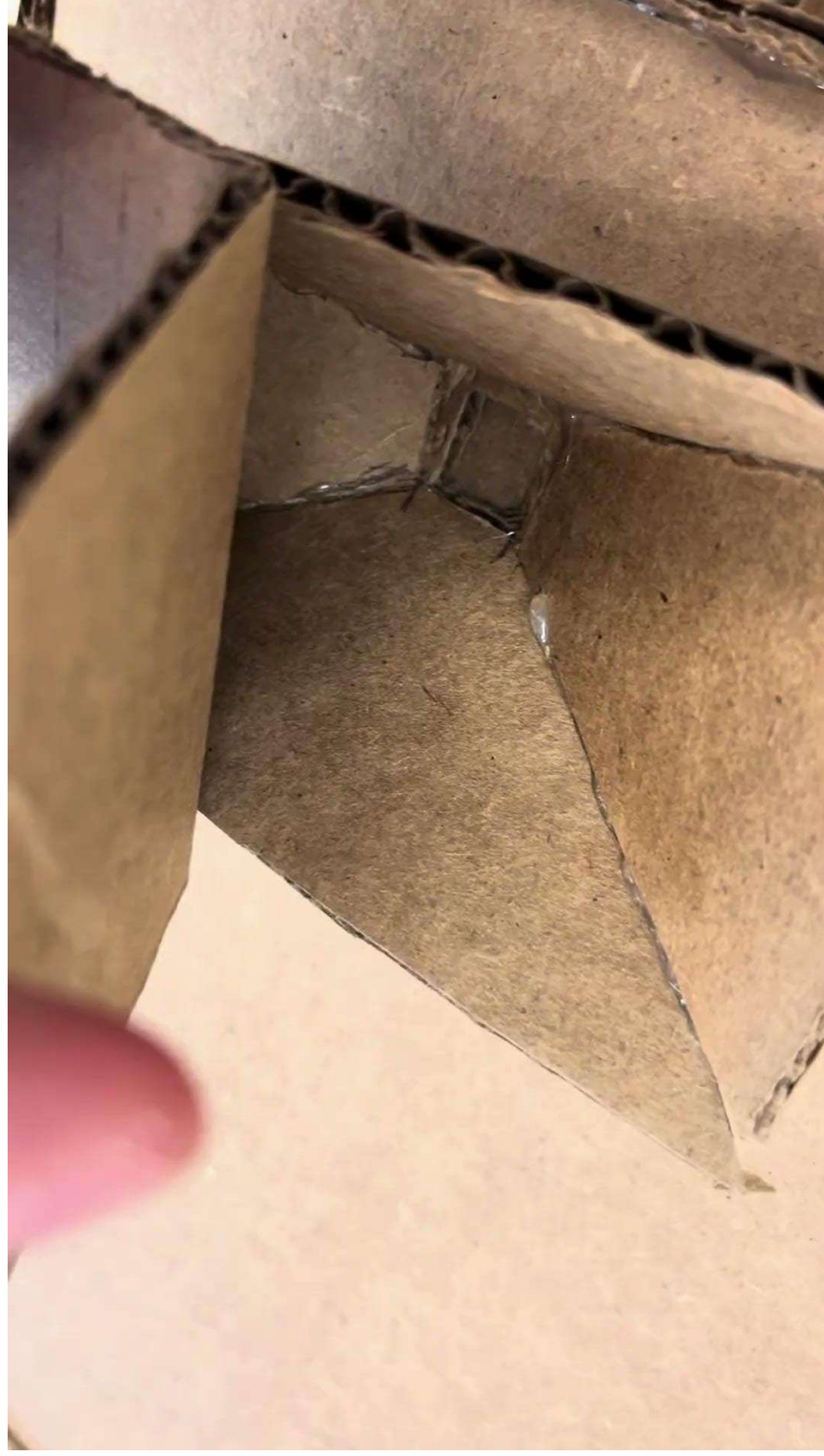
Demonstration



Separate LED Demo



Separate Servo Motor Demo



Hardware/Software

Hardware: Raspberry Pi, Bread Board,
1k Ω Resistor, 470 Ω Resistor, LED,
Servo, 5V Power Supply

Software: ServoBlaster, Bottle,
Datetime



Code

This part of the code imports some of the software components needed and initializes other variables. The `set_angle` function moves the servo to the specified angle using `servoblaster`. `Move_servo` function moves the servo and is used when the button is pressed on the website.

```
import os
import time
from bottle import route, run
from datetime import datetime
from gpiozero import LED

# Servoblaster Configuration
servo_min = 500 # Minimum pulse width (us)
servo_max = 2500 # Maximum pulse width (us)
servo = 2 # Servoblaster channel (linked to GPIO 18)

# Variable to store last pressed time
last_pressed_time = "Never"

# Create led variable
led = LED(26)

def map_value(value, from_low, from_high, to_low, to_high):
    """Maps an angle range to a servo pulse width range."""
    from_range = from_high - from_low
    to_range = to_high - to_low
    scale_factor = float(from_range) / float(to_range)
    return to_low + (value / scale_factor)

def set_angle(angle):
    """Moves the servo to the specified angle using Servoblaster."""
    pulse = int(map_value(angle + 90, 0, 180, servo_min, servo_max))
    command = "echo {}={us} > /dev/servoblaster".format(servo, pulse)
    os.system(command)

def move_servo():
    """Moves the servo to 90 degrees, waits 1 second, then returns to 0."""
    global last_pressed_time
    led.on() #turn on LED
    set_angle(90) # Rotate to 90°
    time.sleep(.2) # Hold for .2 second
    set_angle(0) # Return to 0°
    time.sleep(.8) #hold for .8 second
    led.off() #turn off LED

# Update last pressed time
last_pressed_time = datetime.now().strftime("%I:%M %p") # Format: HH:MM AM/PM
```

More Code

This next part of the code is used to generate the webpage using HTML. The website has a button that can be pressed to move the servo, and the website also notifies you if the servo moved successfully. The website also keeps track of when the last treat was dispensed in Chicago time.

```
@route('/')
def index():
    """Generates the main webpage with a button to control the servo."""
    response = f"""
    <html>
    <head>
        <title>Servo Control</title>
        <script>
            function rotateServo() {{
                fetch('/rotate_servo')
                .then(response => response.text())
                .then(data => {{
                    document.getElementById("lastPressed").innerHTML = data;
                    alert("Servo Rotated!");
                }})
                .catch(error => alert("Error rotating servo"));
            }}
        </script>
    </head>
    <body>
        <h1>Servo Motor Control</h1>
        <button onclick="rotateServo()">Rotate Servo 90°</button>
        <h2>Button was last pressed at: <span id="lastPressed">{last_pressed_time}</span></h2>
    </body>
    </html>
    """
    return response

@route('/rotate_servo')
def rotate_servo():
    """Handles the servo movement when the button is clicked."""
    move_servo()
    return last_pressed_time # Send updated time to webpage

# Run the web server on Raspberry Pi's local network
run(host='0.0.0.0', port=80)
```

Q&A

Any questions?



Thank You!