Automated Pet Food Dispenser

...

Presenters:
Mansoor Umar - Computer Engineering Technology
mumar9@myseneca.ca, 647-646-0606

Nicholas Hess - Computer Engineering Technology
nhess@myseneca.ca, 647-403-3405

Date: Monday April 3rd, 2023

Agenda

1. Product Introduction
    a. What is our Product?
    b. What our Product offers
2. Technical Overview
    a. Elements of our Product
    b. Specifications of Our Product
    c. The Building Process
3. Conclusion
    a. Viability of our Product
    b. Future Developments
    c. Contact Information

What is our Product?

- The Automated Pet Food Dispenser is a feeding automation product that utilizes bluetooth detection in order to detect the presence of a pet to feed.

- The product has all the automation configured by the user through a globally accessible website which contains both monitoring and manual override options.

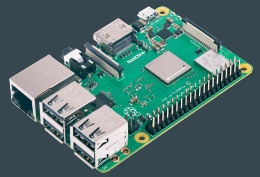
- Our product enables the user to gain more time within their daily schedules while ensuring that their pets remain fed when they want at the portions they want.

What our Product Offers - Key Product Features

- A Port Forwarded Website
- Scheduled Dispense Times linked with Bluetooth
- An Option to Manually Dispense Food
- Pet Detection via Bluetooth Tag

Elements of our Product - Major Components Used

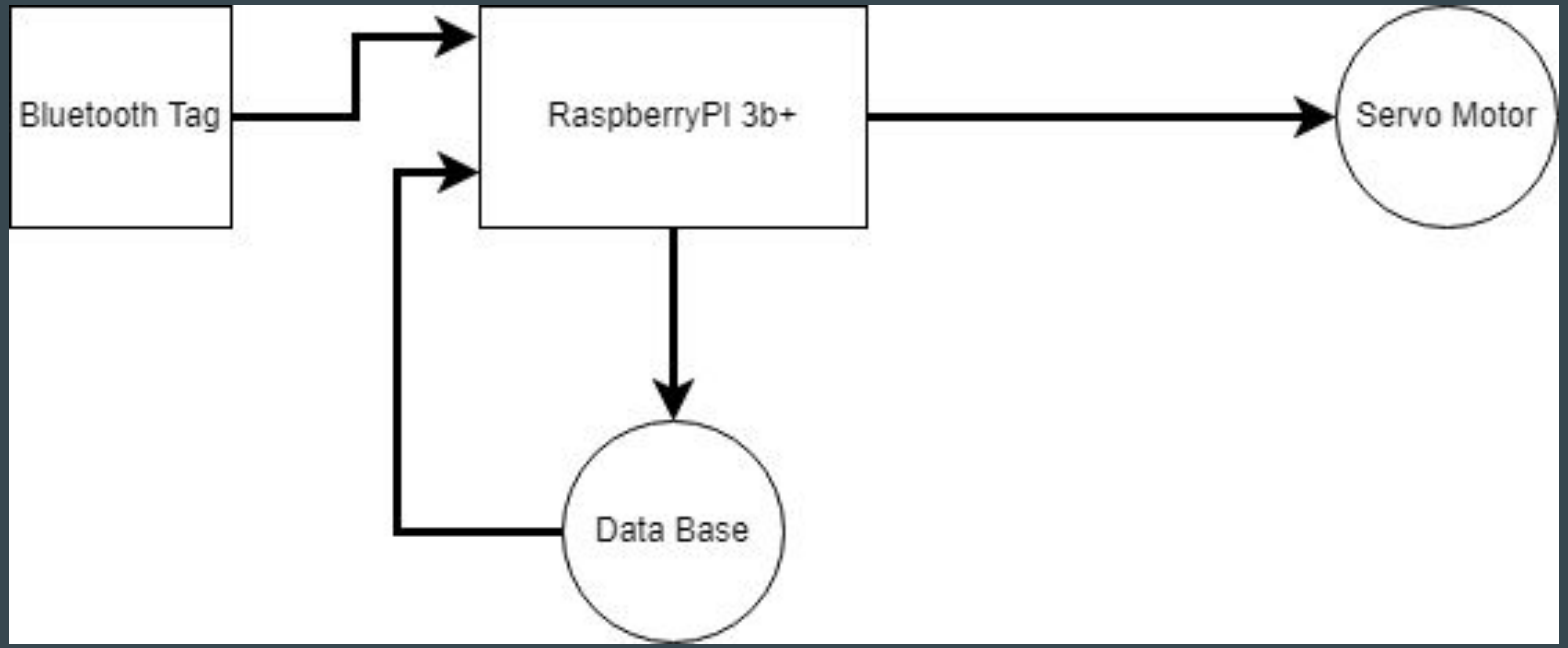
Images of Components



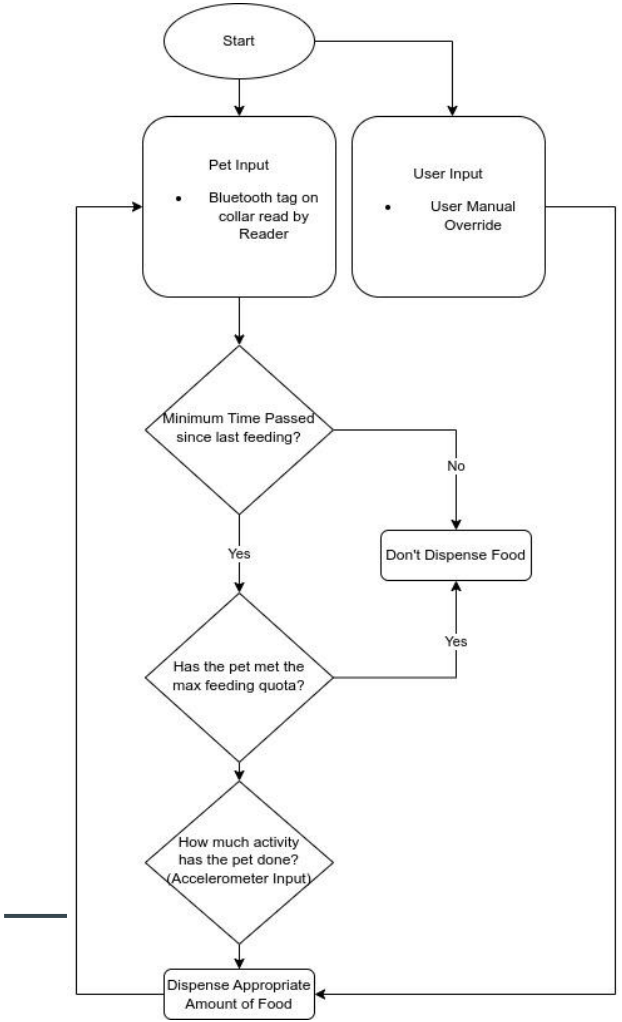
List of Components (No full details)

- Raspberry Pi 3B+
- SPT5525LV-360 25KG Digital Servo Motor
- Anti-Lost Tracker Key Tracker GPS Locator

Specifications of our Product - System Block Diagram



Specifications of our Product
-
Software Diagram



Product Specifications

- Product dimensions: 7.5”D x 7”W x 16”H
- Product weight: 4 Pounds (Full weight)
- Microcontroller: Raspberry Pi 3 B+
- Power Supply: 5.1V 2.5 Amp from Raspberry Pi
- Wireless: 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless

The Building Process - Software

- Web Server/Website Design
- Database Creation/Manipulation
- Bluetooth Detection
- Motor Control
- Port Forwarding

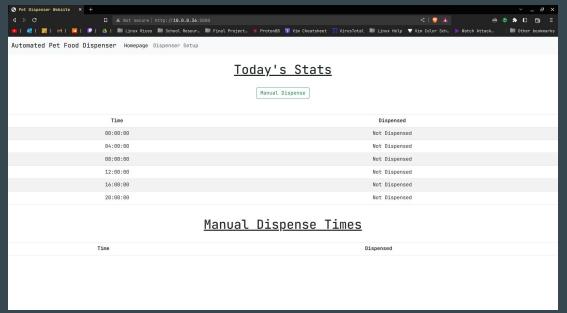
The Building Process - Software - Website Design

- The Website Design process was done through a Flask server that utilized:
  - HTML Pages
  - CSS - Bootstrap Implementation
  - Jinja2 - Python coding within HTML

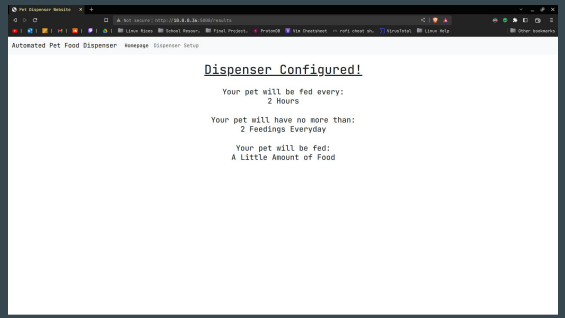
- Along with these Python Libraries:
  - Flask Library
  - Pandas Library
  - Subprocess Library
  - Time Library
  - Gpiozero Library

The Building Process - Software - Website GUI

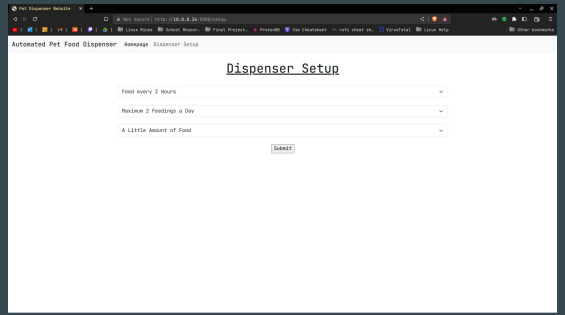
Here are some of the web pages we have on our website



Home Page



Results Page



Config Page

The Building Process - Software - Database Creation/Manipulation

- The website controls databases through the use of excel spreadsheets and the Pandas Python Library.
  - Excel spreadsheets are used in order to have multiple files access the same databases.
- The Dispenser Setup page has the web server copy a template excel spreadsheet to convert to a database for the homepage and has the form values passed into another spreadsheet for later access.
  - The spreadsheet stores the: Feed Intervals, Maximum Feed Quota, and Feeding Amount
- Whenever changes are made through scanning for the pet at the allotted time or reconfiguration of the dispenser, the same process with spreadsheets is utilized to edit the databases.

The Building Process - Software - Bluetooth Detection

- In order to scan for the Bluetooth Tag we utilize the Linux's built in bluetoothctl/hcitool commands and file scanning in python to achieve Bluetooth Detection

- We have a bash script that empties a text file and echos the output of a 5 second bluetooth scan into that text file. The bash script then runs a python program that scans the file for the bluetooth tags mac address within the file. Depending on the result of the scan the appropriate python files are run to manipulate the homepage database.

- In order to run the scans at the allotted times we have the OS check hourly to see whether the current time is present within the excel spreadsheet and then run the bluetooth scan script if it is.
    - This is done hourly through the use of cron jobs within crontab which can be utilized as a linux equivalent to windows task scheduler

The Building Process - Software - Motor Control

- The motor control was done using the Raspberry Pi's GPIO pins and python files that contain the time and gpiozero libraries.
- Once the bluetooth scan detects the tag within the file, a bash script that runs the appropriate motor file for different dispense amounts is run.
- The motor files run a command that send a signal to the servo to run at maximum speed and is only run until the sleep time is over. The differing feed amounts are based on the time the motor spends running during a feed.
  - The more food the user wants to dispense the higher the sleep time for the motor to run
- The pin that the signal is sent from is GPIO12 of the Raspberry Pi.
  - That pin is used as the signal sent to the motor through wires

The Building Process - Software - Port Forwarding

- In order to port forward our Flask server we grabbed the public IPV4 of the Raspberry Pi through the website: https://www.whatismyip.com/, internal network address of the Raspberry Pi using ifconfig in a terminal, and used our service providers router management app to configure port forwarding.

- On our router management app we went into the port forward settings and port forwarded port :5000 on the internal network address to allow for traffic to go both in and out through that port on the ip

- Now we were able to connect to the Flask server outside of the local network by typing the public IPV4 of the Raspberry Pi followed by ":5000".

The Building Process - Hardware

- The main base is the Zevro Dry Food Dispenser
- To connect the servo motor to the dispenser shaft an adapter coupling was used
  - To mount the motor, a piece of PVC pipe was cut and the motor was slotted in
    - To mount the PVC pipe hose clamps were connected around the pipe to the dispenser
- Holes were drilled in the dispenser to allow wire routing to the Raspberry Pi
  - The wires were glued in place to insure they would not be unplugged
- The Raspberry Pi was secured to the back of the dispenser to keep it out of the way and balance weight
- A second PVC pipe was added to direct the food being dispensed

Viability of our Product

- Complete Automation of Pet Feeding Process
- Remote Access and Control over how much your pet eats
- Low Network and Power Requirements
- Compact and lightweight

Future Developments

- A full enclosure
  - This would provide more security against pets moving the dispenser
- An activity detection system
  - Adding an accelerometer onto the tag would allow for the portioning to be controlled through activity detection
- Custom Server URL
  - Purchasing and configuring a domain name for the user to utilize instead of going through the port
- Accommodation for multiple pets
  - Opening up the system to allow for multiple pets to be configured to eat at different times

References

360 Continuous Rotation SPT5525LV-360 25KG Digital Servo. (n.d.). ebay. Retrieved from https://www.ebay.com/itm/203881216784.

Anti-lost Bluetooth-compatible Tracer. (n.d.). Walmart. Retrieved from https://www.walmart.ca/en/ip/Mymisisa-Anti-lost-Bluetooth-compatible-Tracer-Alarm-Locator-Tracker-Device-Black/630XU59SCW15?skuId=5RDVQ2SY62O0&offerId=7D907407975143D68E1D0EBBDD15CC64.

Raspberry Pi 3 Png. (n.d.). PNGITEM. Retrieved from https://www.pngitem.com/middle/mbmmxT_raspberry-pi-3-png-transparent-png/.

Sandra Manson. (n.d.). *Futuristic Laptop*. Lera Blog. Retrieved from https://lerablog.org/technology/software/discovery-phase-in-the-software-development-process-key-deliverables/.

Website Gui popping out of computer. (n.d.). eliNEXT. Retrieved from https://www.elinext.com/blog/modular-web-design/.

Automated Pet Food Dispenser

Contact Information:

- Mansoor Umar
  - Email: mumar9@myseneca.ca
  - Telephone: 647-646-0606

- Nicholas Hess
  - Email: nhess@myseneca.ca
  - Telephone: 647-403-3405