Automatic Pet Feeder

Senior Design Project - Group 12

Devin Besaw - CpE
Fernando Oviedo - CpE
Gershon Prospere - EE
Jose Tapizuent - CpE

Motivation

Our goal is to build an Automatic Pet Feeder that provides the support that pet owners need in order to keep their pets healthy and taken care of.

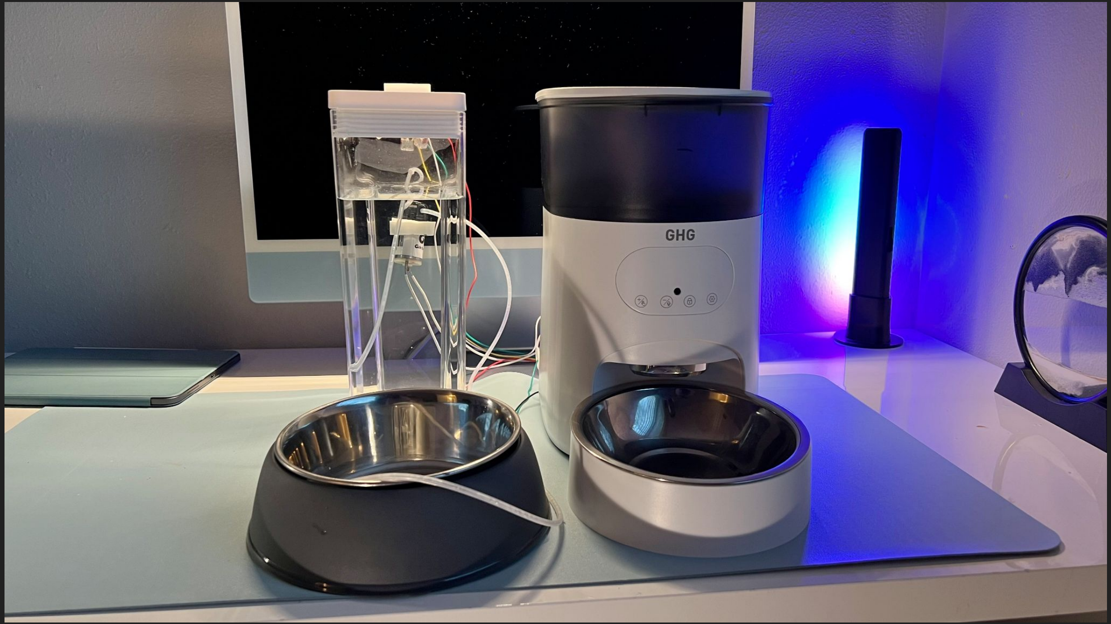
- To offer more convenience when feeding pets.
- To provide food and water in a timely and consistent manner.
- To help pets stay healthy and lose weight.

Project Goals and Objectives

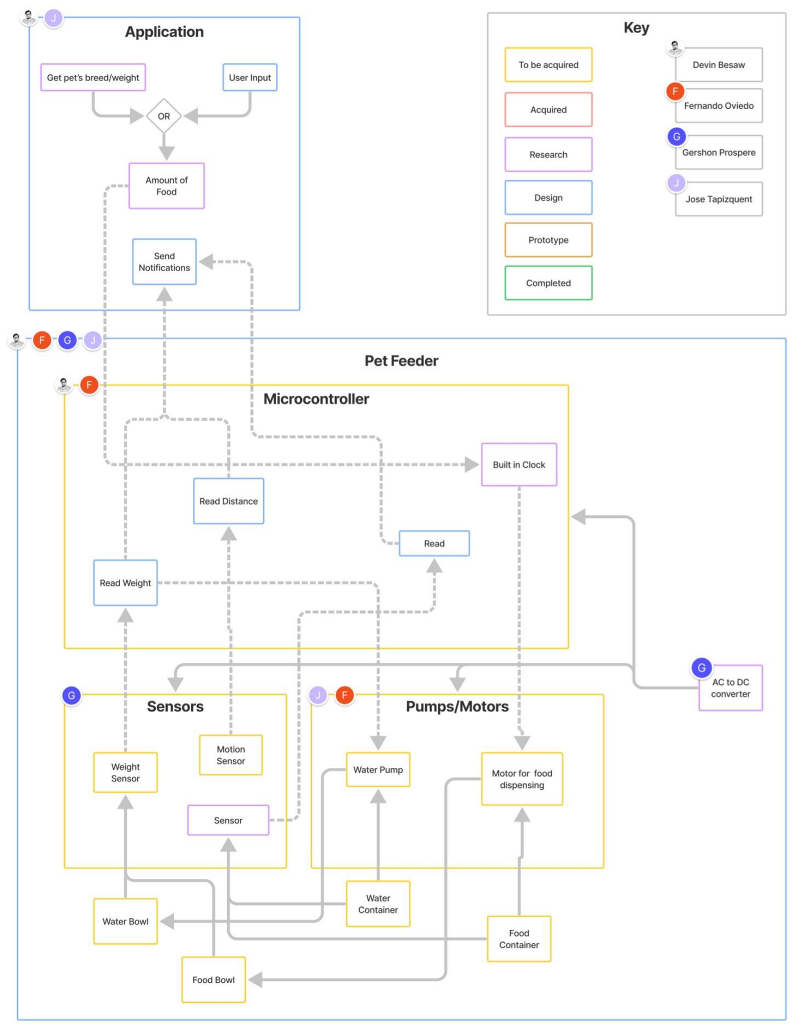
- The dispensing of food in a timely manner
- Automatic refilling of a water bowl when it’s low.
- Gather information from sensors.
    - Ultrasonic sensors to determine how much is left in the food and water reservoirs.
- Camera (Advanced features)

Housing/Casing

We have decided on using an existing housing of an Automatic Pet Feeder and have a separate water containment attached to the main housing that work synchronously.



Block Diagram



Overall Design Approach and Implementation

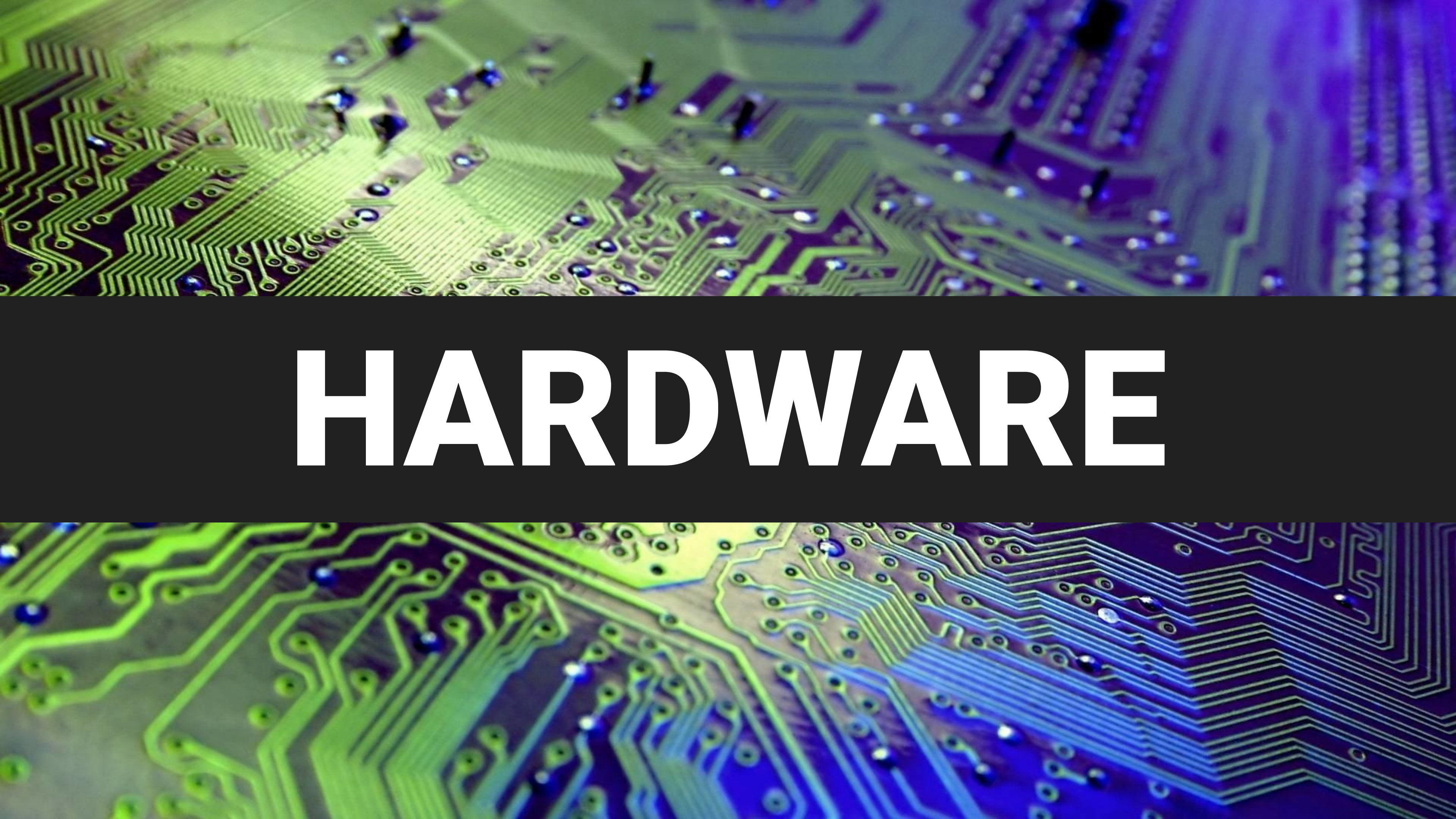
Hardware

- MCU (Arduino, Raspberry Pi)
- Power Supply & Voltage Regulators
- Servo Motor
- Ultrasonic sensors & Pressure sensors
- PCB
- Water Pump
- Camera/Microphone (Advanced Features)

Software

- App (React Native, JS)
- Python, C++ to program the Raspberry Pi and Arduino

HARDWARE



Arduino

- 32 KB of memory
- Operating Voltage of 2.7-5.5 V
- 23 pins

Used for:

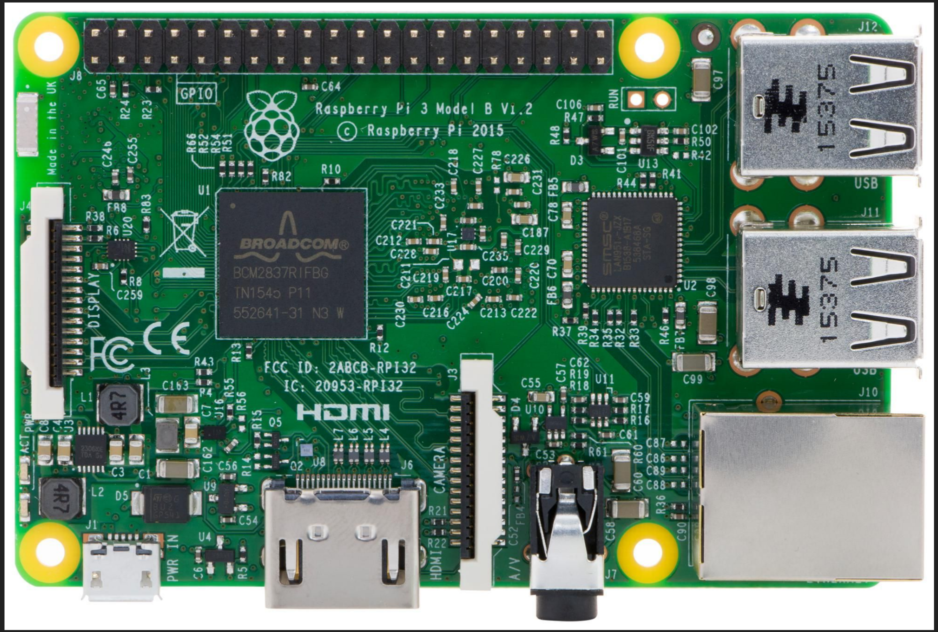
- Taking data from the sensors
- Giving instruction to the pump and motor

Raspberry Pi 3

- 1 GB of memory
- Operating Voltage of 5 V
- 40 pins

Used for:

- Main controller
- Communicating to the Arduino
- Camera



Ultrasonic Sensor

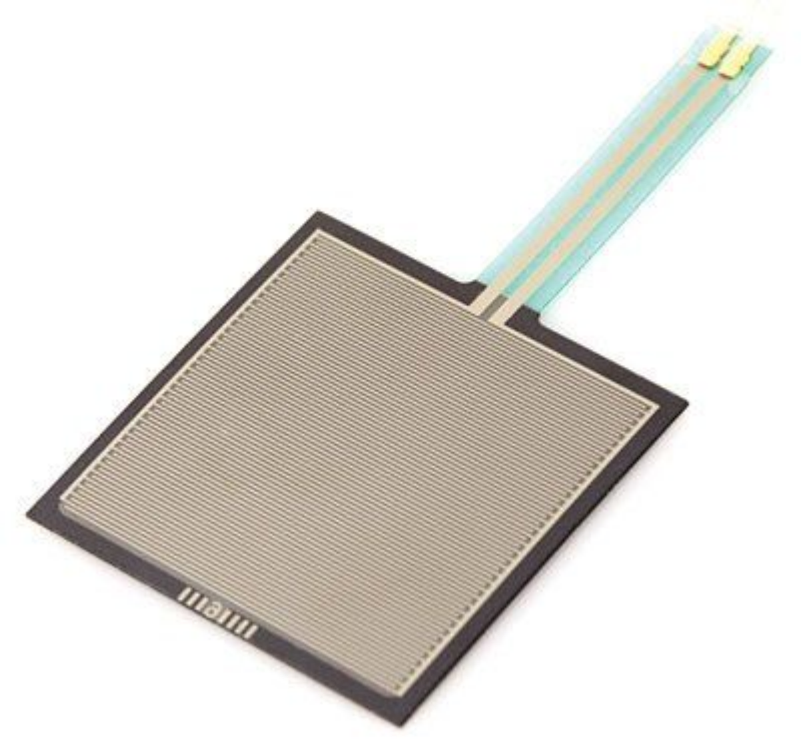
Used to detect the water and food levels in respective tanks

Input Voltage	5 V
Working Current	15 mA
Max Range	2 cm
Min Range	400 cm

Pressure Sensor

Measures the weight of the water bowl

Voltage	5 V
Weight	0.08 ou
Area	4 x 4 cm



Pump & Motor

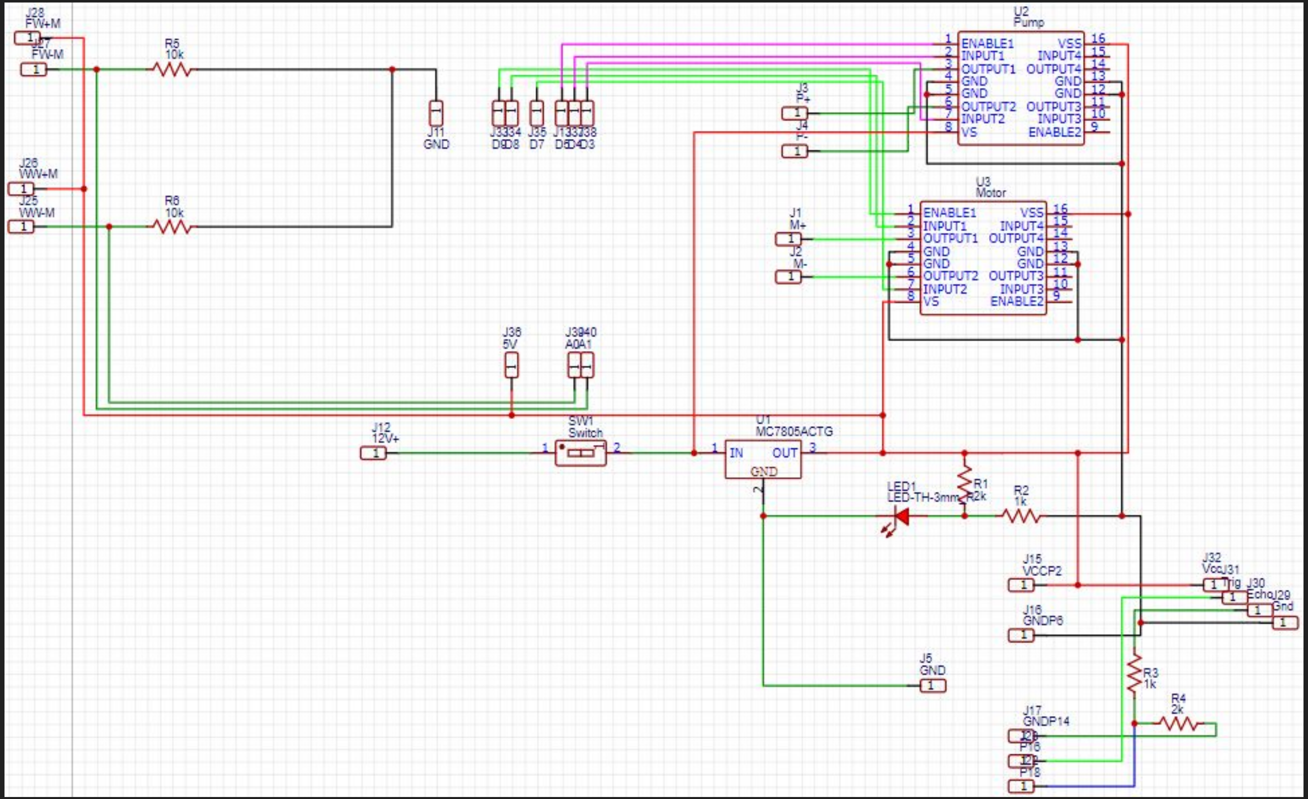
	Pump	Motor
Voltage	12 V	5 V
Weight	2.4 ou	3.2 ou
RPM	5000	6000

Voltage Regulator

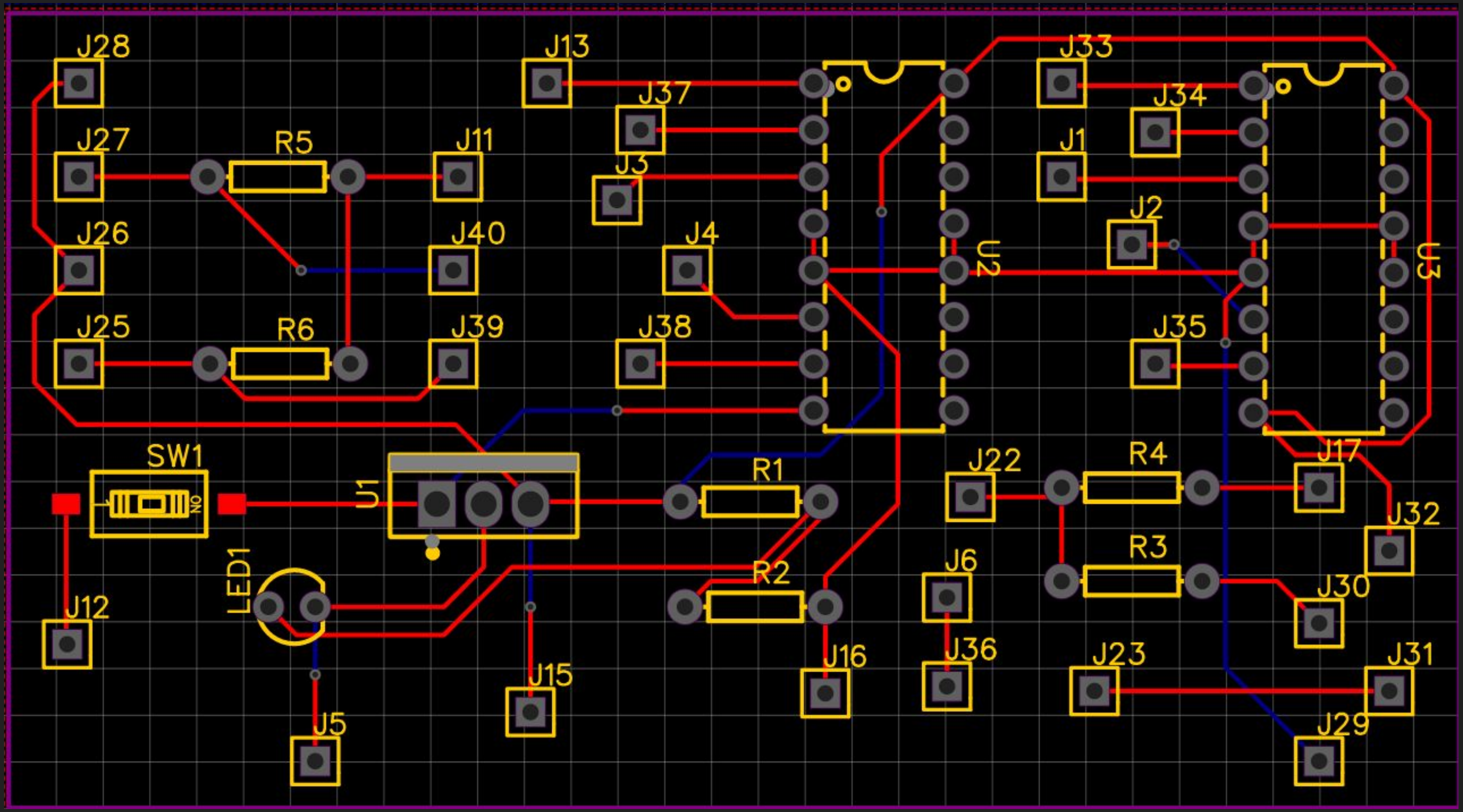
MC7805ACTG Through Hole

Voltage Max Input	35 V
Voltage Min Input	7 V
Voltage Output	4.8 - 5.2 V
Temperature	0 - 125 C

Schematic


J28
FW+M
J27
FW-M
R5
10k
J11
GND
J3334
D9D8
J35
D7
J133738
D5D4D3
U2
Pump
1 ENABLE1
2 INPUT1
3 OUTPUT1
4 GND
5 GND
6 OUTPUT2
7 INPUT2
8 VS
VSS 16
INPUT4 15
OUTPUT4 14
GND 13
GND 12
OUTPUT3 11
INPUT3 10
ENABLE2 9
J3
P+
J4
P-
J26
WW+M
J25
WW-M
R6
10k
U3
Motor
1 ENABLE1
2 INPUT1
3 OUTPUT1
4 GND
5 GND
6 OUTPUT2
7 INPUT2
8 VS
VSS 16
INPUT4 15
OUTPUT4 14
GND 13
GND 12
OUTPUT3 11
INPUT3 10
ENABLE2 9
J1
M+
J2
M-
J36
5V
J3940
A0A1
J12
12V+
SW1
Switch
1
2
U1
MC7805ACTG
1 IN
OUT 3
GND
2
LED1
LED-TH-3mm
R1
R2k
R2
1k
J15
VCCP2
J32
VccJ31
Trig J30
EchoJ29
Gnd
J16
GNDP6
R3
1k
R4
2k
J17
GNDP14
J20
P16
J22
P18
J5
GND

PCB


J28
J13
J33
J37
J34
J27
R5
J11
J3
J1
J2
J26
J40
J4
U2
U3
J25
R6
J39
J38
J35
SW1
ON
U1
R1
J22
R4
J17
J32
LED1
R2
J6
R3
J30
J12
J16
J36
J23
J31
J15
J5
J29

Final PCB Testing

- Connect adapter to ensure enough power passing through
- Test voltage regulator
- Run pcb with all components soldered
- Attempt use of Atmega328

document.getElementById(div).
else if (i==2)
{
var atpos=inputs[i].indexOf("@");
inputs[i].lastIndexOf(".");
else
document.getElementB
}
if (i==5)

SOFTWARE

Software Design

Objectives	Description
Basic	Programming the MCU's among other components to make sure the functionality is optimal
Advanced	Implement camera feed to allow pet parents to monitor pet. Also, by using the speaker, play classical music to calm the pet and preventing separation anxiety.
Stretch	Integrate computer vision to identify that the right pet is fed and enclose the bowls if its an intruder.

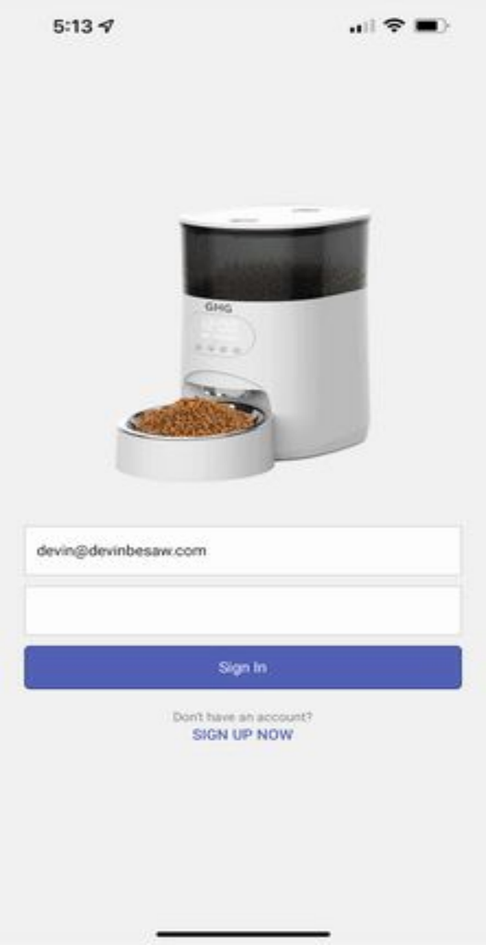
Programming Languages

MCU/Application	Languages/Framework	Percentage used for the completion of the project
Arduino	C++	20%
Raspberry Pi	Flask, Python	45%
App	React Native, Js	35%

User Communication

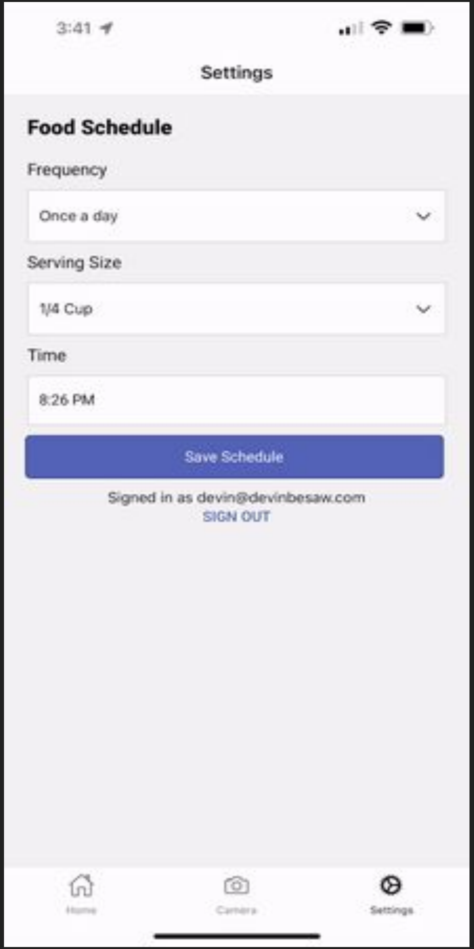
The entire communication happens through a mobile application that pairs with the Automatic Pet Feeder.

This allows the pet owner to customize the timing for the Automatic Pet Feeder and quantity served, to monitor the levels of food and water left in the containers, and view the camera feed



devin@devinbesaw.com

Don't have an account?
SIGN UP NOW



Settings

Food Schedule

Frequency: Once a day

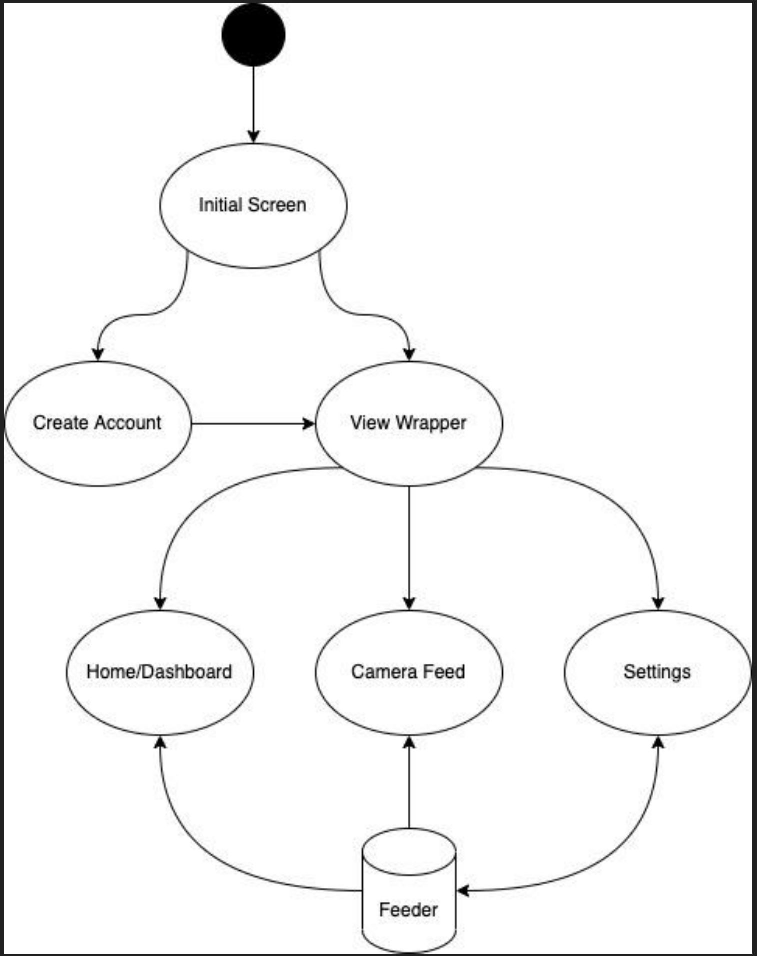
Serving Size: 1/4 Cup

Time: 8:26 PM

Signed in as devin@devinbesaw.com

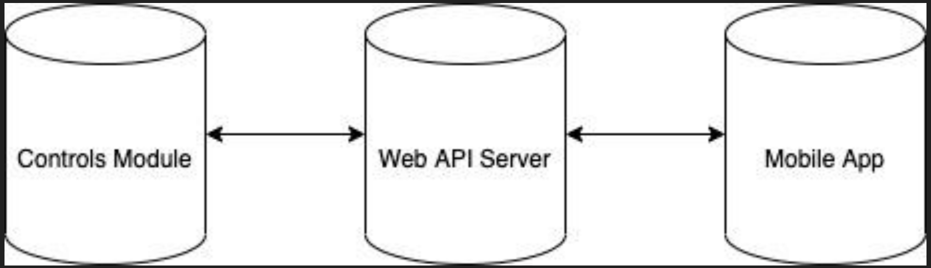
Mobile Application Flow

Describes the flow the user’s experiences when interacting with the mobile application



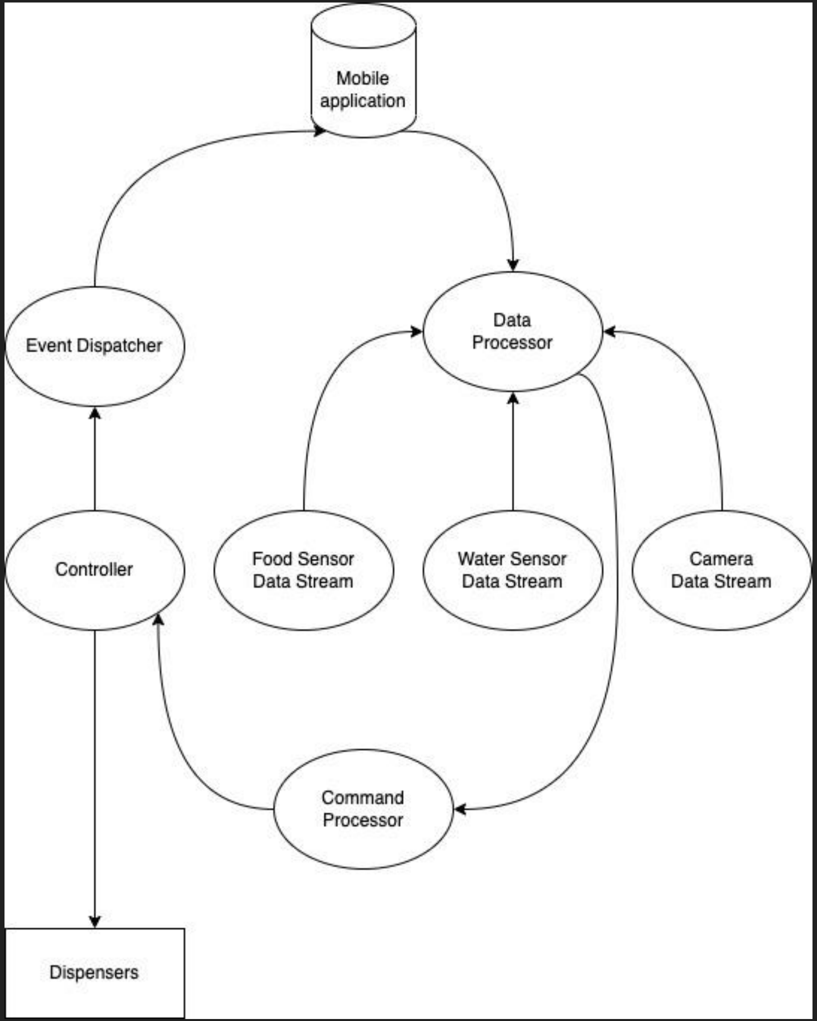
Communications

Web API server to handle communications between mobile application and the Automatic Pet Feeder



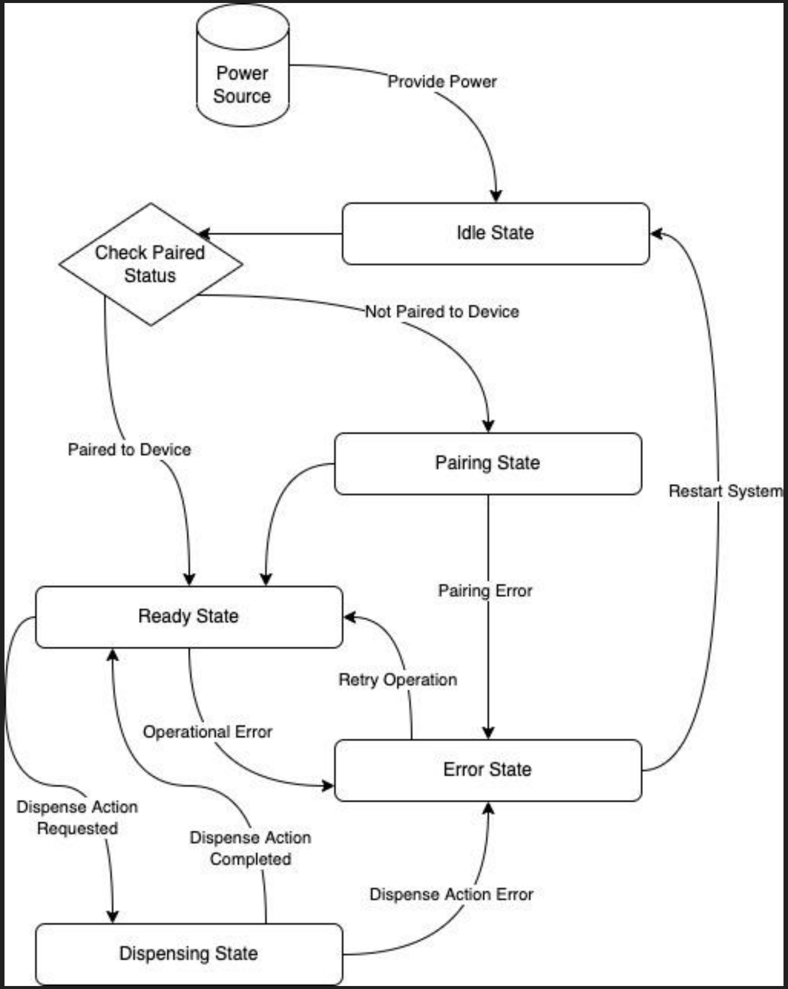
Pet Feeder Operations Flow

Describes the processing of data and operating interpreting



Operating States

Different states that the system might be in when turned on and enabled



Budget and Financing

Item	**Price**
Casing	~$50
12V DC Peristaltic Pump	$12.78
Microcontrollers	$0
PCBs	$22.80
Power Supply	$19.76
Motors/Motor Drivers	$9.62
Ultrasonic Sensor	$14.54
Raspberry Pi Cooling Fan/Arducam	$27.8
Pressure Sensor	$33.79
Other Electrical Components	~$87
Total	**~ $270.00**

Project Success and Challenges

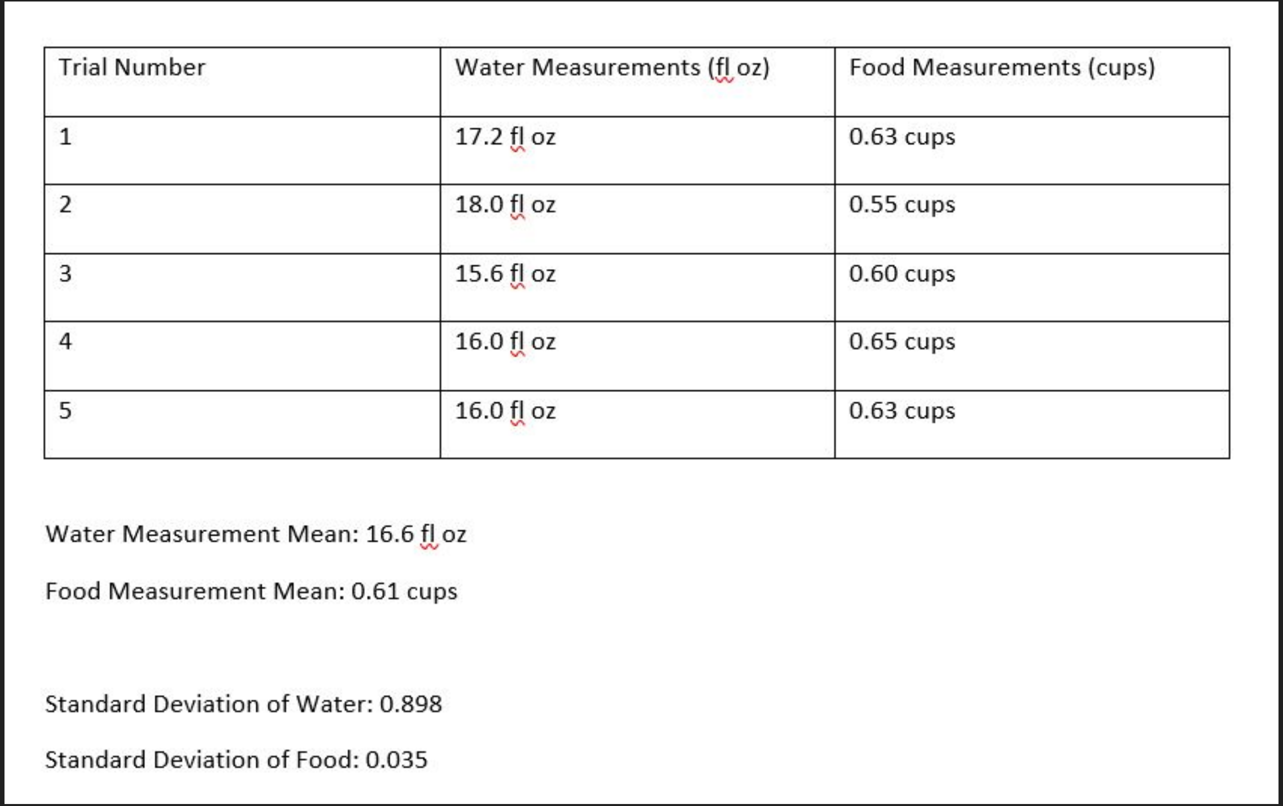
Successes

- The availability of the PCB.
- Having most components in our disposal.
- The app and schedule of the food dispensing works smoothly.
- Pre-existing housing for the food provided enough space for the electrical components.

Challenges

- PCB was printed with the wrong size motor pinout, have to re-order.
- Time Constraint due to work schedule not meshing.
- Limited space inside the housing.
- Force pressure sensor gives inaccurate data/measurements.

Testbench the Food/water



Trial Number	Water Measurements (fl oz)	Food Measurements (cups)
1	17.2 fl oz	0.63 cups
2	18.0 fl oz	0.55 cups
3	15.6 fl oz	0.60 cups
4	16.0 fl oz	0.65 cups
5	16.0 fl oz	0.63 cups

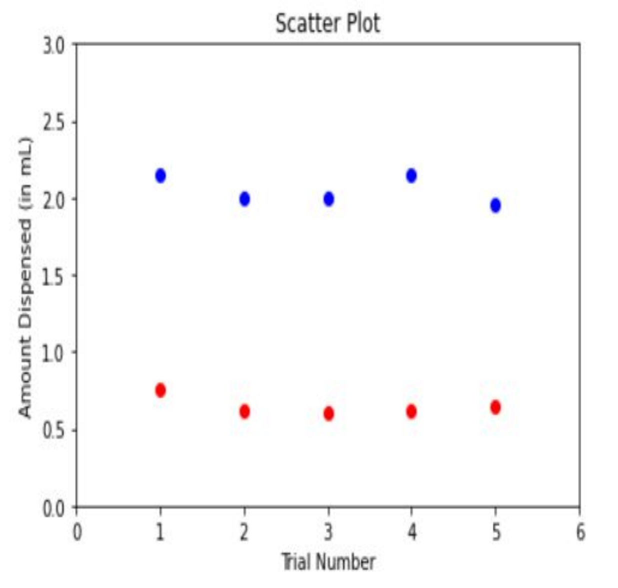
Water Measurement Mean: 16.6 fl oz

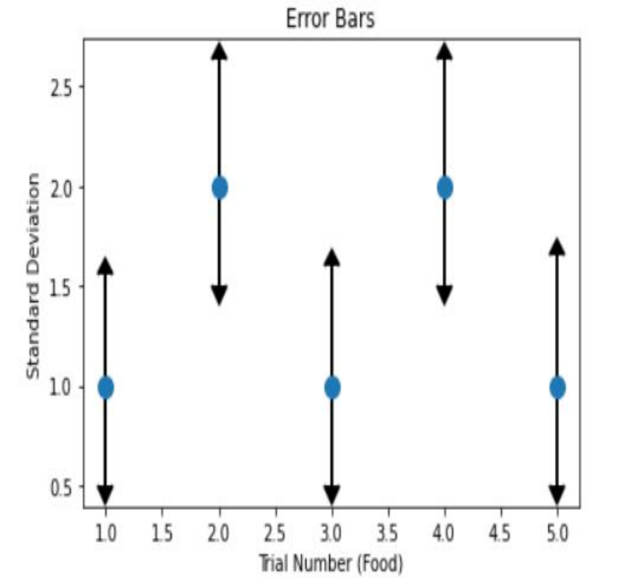
Food Measurement Mean: 0.61 cups

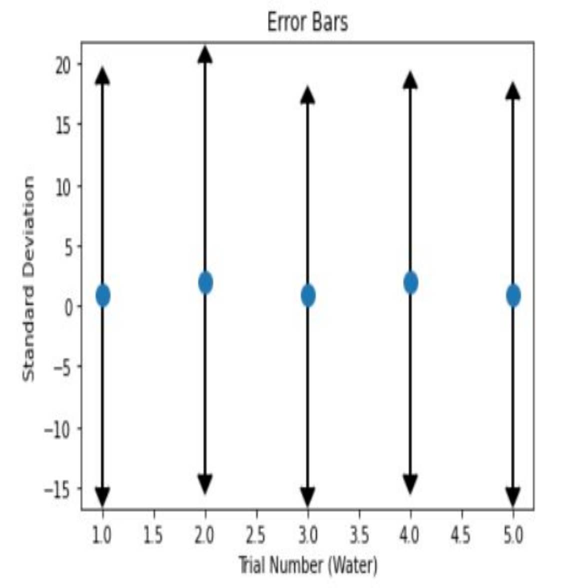
Standard Deviation of Water: 0.898

Standard Deviation of Food: 0.035

Standard Deviation/Error Bar Graph







Conclusion/Alternative Endings

Conclusion:

- Most Automatic Pet Feeders in the market do not use sensors nor dispense both food/water autonomously.
- The communication between the MCU’s and the API worked smoothly.
- All components were able to be applied to the PCB with no problem.

What we could’ve done differently?

- Chosen a different type of sensor for weight.
- Possibly cutting off the water dispensing and fully focus on going limitlessly on the food dispensing.
- Having more time to dive into the stretched goals such as implementing music to soothe the pets and possibly DoS for intruding pets and blocking the food.

QUESTIONS