Pet Bowl Water Sensor

Nate Gross  
*Team Pontiac*  
*CSU Online Student*

**The “Pet Bowl Water Sensor” will send e-mail notifications to your device that contain water level statuses such as FULL, LOW, and CRITICAL. Additionally, it is setup to receive an input from the user to define the receiving email address in order to enable customization. In this report we will detail the accuracy and response time as well as the costs and marketability for this item.**

# Introduction

The hardware being used for this water-level sensor project is the Raspberry Pi 3 with an ADC (MCP3008) that’s connected to a breadboard to provide us analog to digital conversion. The jumper leads power the breadboard from the Raspberry Pi while the other leads link the MCP3008 ADC to the GPIO (find more technical names in datasheet. Finally, we have the Rain Water Level Sensor that is powered from the breadboard and giving us the 3 readings for water level detection.

The operating system being used is Raspbian which includes Python IDE Thonny that will compile our Python code and communicate with the hardware. E-mail notifications will be transmitted via Wi-Fi that’s built-in to the Raspberry Pi 3.

# Problem characterization

Achieving accuracy and response time of prototype is critical to achieve success. The device will undergo calibrations and repeatability tests that it is reading accurately. The device will also need to be tested for response time to personal mobile device letting user know of status.

Once quality of prototype is achieved, I need to evaluate cost and marketability of item. Costs will be calculated based off of BOM and a price point will be chosen based on proven profit margins while comparing other automated pet product price points. I will then conduct surveys to quantify if price point is valid.

## Accuracy and Response Time

After the breadboard circuit was initially hardwired, I wanted to verify everything was connected correctly. A multimeter was used on my wire leads that would go to the sensor to confirm current was flowing and reaching the sensor, and hat voltage readings were successfully received.

In order to qualify the accuracy of the device, it will be exposed to repeated tests to verify the output is stable to the end user. The requires controlled runs with a timed measurement window for final comparison. In order to record the stability, a CSV file will be generated while the program runs to record the received value from the ADC.

## Cost and Marketability

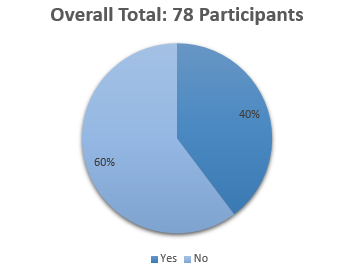
To determine marketability of this item we must first figure out what our cost is to build. Below is the original BOM consisting of quantities to build a single sensing module:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Desc** | **Qty** | **Cost** | **Supplier** |
| 1 | Raspberry Pi 3 | 1 | $35 | Amazon (online) |
| 2 | 5” eTape Liquid Level Sensor  Included:   * 1000Ω resistor * 4 pin connector | 1 | $40 | Adafruit Industries (online) |
| 3 | MCP3008 ADC (Analog to Digital Converter) | 1 | $4 | Adafruit Industries (online) |
| 4 | Breadboard | 1 | $5 | Adafruit Industries (online) |
| 5 | Micro SD card (8GB) | 1 | $10 | Adafruit Industries (online) |
| 6 | Male to Male/Female to Male wires (6”) | 2 | $4 | Adafruit Industries (online) |
| **TOTAL COST** | | | **$98** | |

The demand for this product seems strong as the idea provides convenience for a pet when faced with a busy lifestyle. This pet device falls into a category of other automated pet products that helped gauge a price point of what people are willing to pay for convenience. I also applied what Corporate Finance Institute indicates as a successful profit margin and went above that to ensure I can account for costs that may need to be added before product can be finalized. Since the cost of our original sensing module is around $100, I set the price point at $149.99 which is considered as a high margin at 33%.

A survey was conducted with the price point of $149.99 that combines participants from social media and in person surveying. The survey consisted of a marketing pitch and a simple Yes/No response to see if participants would pay this price. Below you will find the totals from 78 participants:

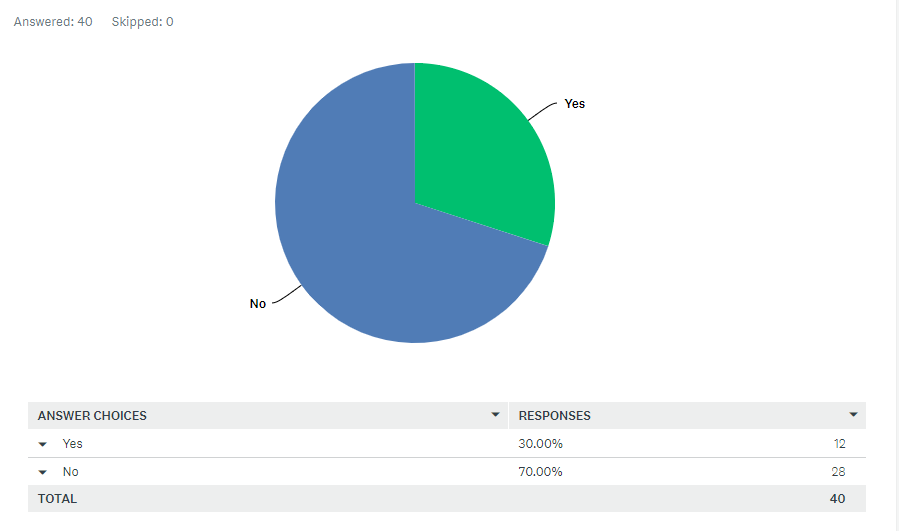
**Marketing pitch:** *Have pets with a busy lifestyle? Find yourself forgetting to check the water bowl or worse… checking on It unnecessarily? Worry no more with the "Pet Bowl Water Sensor"! This one-time purchase will provide push notifications to your mobile device indicating when your pets water bowl is either filled, getting low, or warning you when it reaches critical levels. Be alerted anywhere, at any time, to ensure your pets are properly hydrated for a one-time purchase of $149.99. Would you pay this price?*



Overall people did not favor the price point. Here are the individual totals separated by social media platform and in-person survey:

**Digital Survey- Facebook**

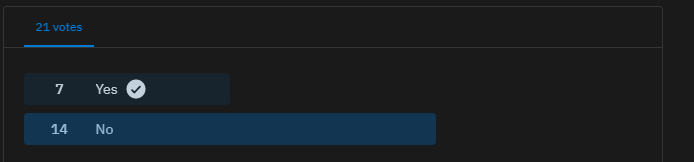
***\*Completed through “Survey Monkey”***



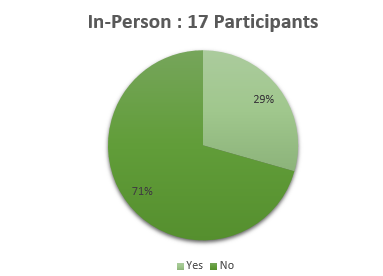
**Digital Survey**

***Reddit***





**In-person Survey of Employees : 17 participants**

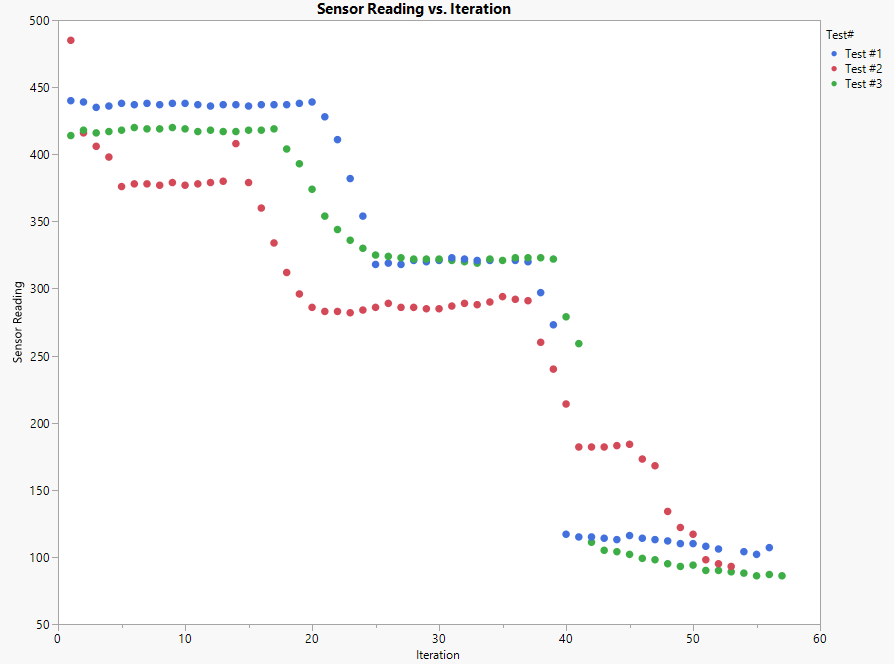


The data concludes that digital surveys were an overwhelming “No” at 69% with a total of while the in-person survey did well with 71% voting “Yes”.

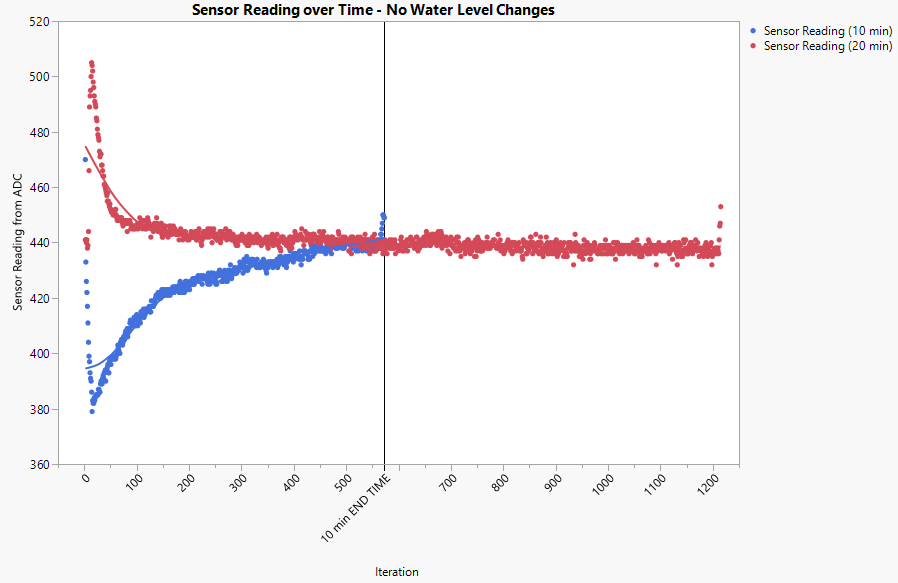
# PROPOSED SOLUTION AND IMPLEMENTATION STRATEGY

## Reliability

After a successful use of the device, it was subjected to repeated tests to ensure an accurate response. Trial runs were conducted in 30 second intervals for each of the 3 states. This correlated to roughly 20 iterations per state. Each test showed a step function response to the new state, and an email was successfully generated.



In addition, an extended time period testing was performed to ensure the decay over time was minimal and to ensure accuracy. Both a 10-minute and 20-minute interval were observed for this test. It was noted that there were outlier points upon initialization that stabilized over time to a consistent reading that matched between the two time periods within the same bowl of water. Below graph highlights this observation, the red data points are from the 20-minutes trial and the blue data represents the 10-minute trial, along with a notation for when the 10-minute trial completed.



## Marketability

The survey data conducted above shows that we need to improve on affordability. After reviewing the BOM, I decided to use a different and more affordable sensor that will provide the same results:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Desc** | **Qty** | **Cost** | **Supplier** |
| 1 | Raspberry Pi 3 | 1 | $35 | Amazon (online) |
| 2 | Rain Water Level Sensor | 10 | $6 | Adafruit Industries (online) |
| 3 | MCP3008 ADC (Analog to Digital Converter) | 1 | $4 | Adafruit Industries (online) |
| 4 | Breadboard | 1 | $5 | Adafruit Industries (online) |
| 5 | Micro SD card (8GB) | 1 | $10 | Adafruit Industries (online) |
| 6 | Male to Male/Female to Male wires (6”) | 2 | $4 | Adafruit Industries (online) |
| **TOTAL COST** | | | **$64** | |

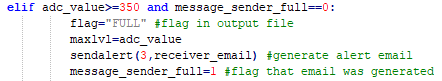
The new BOM now reflects a total cost of $64 and a price point of $149.99 would put us at a 57% profit margin. The product can now be advertised as affordable while generating a profit, creating what I believe to be favorable opportunities to consumer and manufacturer.

## Python Script Implementation Methodology

The setup utilized Raspberry Pi connected to a breadboard for wiring the Analog-to-Digital Converter (i.e. ADC) and water sensor. The methodology for wiring followed similar example of analog sensing signal devices: I/O pins for sensor reading, Vcc for power (3.3V), GROUND, and clock (CLK) for timing to the operating system. The ADC connected to the GPIO pins, CLK, 3.3V, and GROUND on the Raspberry Pi. The water sensor was wired to the I/O PIN on the ADC connected via Channel #0 as well as Vcc and GROUND to enable sensing.

Within the Water Sensor python program, the GPIO python library was utilized to initialize all the pin values and define their input locations. The readings from the ADC were then decoded and returned to the main function for evaluation: the expected range for consideration was 0-500 for the sensor reading. The code was setup to auto-calibrate the max water level reading based on the first refill of the water bowl to avoid premature alerts. To enable this: the max reading is not set until a reading of over 350 is received from the sensor indicating the water level has risen to a healthy level.

Once the FULL water level has been achieved, an email is generated to the user and a flag is toggled to indicate the alert has been transmitted. The setup for the LOW and EMPTY states are similar, using flags to indicate the state has been achieved. There are also conditions set to ensure no toggling between states can occur in order to prevent an overflow of alerts to the user. The example code below illustrates the FULL state checks and flags.



Since the consumption rate is expected to be slow (i.e. one day for full use) the system is designed to continuously run, and the states will be reset only once the water level has been refilled. Alerts are sent via email from an automatic message account to the email address initially setup by the user during startup. This allows customization for the user as needed.

# Conclusions

The repeatability tests showed a stable output on multiple tests. One notable observation is that when the device initially powered up, there was an outlier data point observed, but within one minute the readings stabilized. Overall, the product was deemed stable and output was as expected to the end-user after replacing the hardware to allow for cost-savings.

Cost and marketability of the item after replacing the sensor should provide opportunity to make a “high” profit margin on the product. There is also opportunity to scale price down in affordability to gain more favorable results if or when product comes to market.

# BIBLIOGRAPHY

[1] David Beazley, and Brian K. Jones “Python Cookbook, Third Edition”

[2] Raspberry Pi 3 Model B+  Datasheet. Available; <https://static.raspberrypi.org/files/gproduct-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf>

[3]  OSYOO “How to read Raspberry Pi i/o pin diagram (GPIO  pin graph”. Available: <https://osoyoo.com/2015/10/04/how-to-read-raspberry-pi-io-pin-diagram-gpio-pin-graph/>

[4]  Microchip MCP3004/3008 DataSheet. Available: <https://cdn-shop.adafruit.com/datasheets/MCP3008.pdf>

[5] CFI “Profit Margin”. Available: <https://corporatefinanceinstitute.com/resources/knowledge/accounting/profit-margin/>