CSC 591: Internet of Things: Architectures, Applications, and Implementation

Submission Date: 04/27/22

Andrew Sonnier

Component	**Component Weight**	**Andrew Sonnier**
Ultrasonic sensor demo	.2	100%
Manual water drinking program	.3	100%
Automatic water drinking program	.2	100%
Phone Notification Demo	.15	100%
Web app Mockup	.15	100%
Aggregate Contribution	1	100%

Introduction:

The objective of this project is to create a system to keep track of users' water drinking habits from a water bottle. Similar products in this market do exist, such as the HidrateSpark Bottle which are full water bottles that integrate with your smartphone to keep track of drinking habits. These bottles use accelerometers combined with weight sensors which would be able to tell how much water you have drunk. This project set out to prototype a self contained system in the cap of a bottle which would be able to fit onto existing water bottles.

Design:

When originally proposed, this project was designed to be an end-to-end minimal viable product application which would be usable.

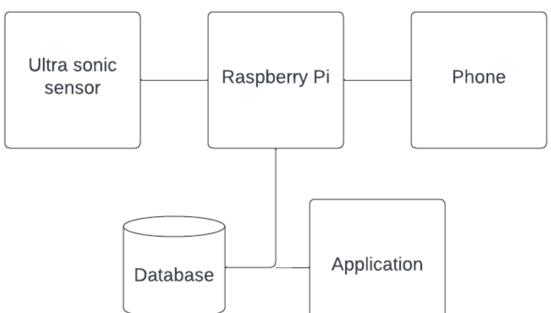

Figure 1

Shown in figure one, the general design was to have an ultrasonic sensor embedded in the cap which would communicate to a microcontroller, in this case a raspberry pi which then would handle storage of information of the database and communication to the phone for data visualization and notifications. Users of this system would need to have the height and width of their water bottle tuned to the system in order to get accurate measurements.

Implementation:

All programming was done in python for its simplicity for prototyping and support available for the raspberry pi's hardware. Gpiozero was used as an alternative to the built-in RPI.GPIO library for its added functionality and the pigpio daemon was used to improve performance and consistency of the GPIO interface. Requests was used to make http requests in order to phone notifications to be sent.

For this project, IFTTT was decided to be used to send phone notifications using a websocket and the IFTTT app.

Deviation from the original design was necessary to ensure completion by the end of semester. In the modified design, four scripts were created all showcasing a different part of the system: ultrasonic_test.py, manual_measurement.py, notification_test.py and auto_measurement.py. Ultrasonic_test is a script designed to test the ultrasonic sensor and its capabilities by printing out the recorded distance every half a second.

Manual_measurement builds on ultrasonic_test and takes two distinct measurements and calculates the ounces drunk between the measurements. Notification_test is a script

To test sending notifications to the phone and ensuring that its capabilities are proper.

Auto_measurement combines all these things into one script in which it automatically takes measurements and will send a notification to the phone if the bottle sits still for long enough

Results and Discussions:

The first complication discovered was the selection of the ultrasonic sensor. It was selected to use the JSN-SR04T sonar sensor for this project due to its low cost, remote sensor board which made prototyping easy and its waterproof qualities. Figure two shows how this sensor was wired into the raspberry pi.

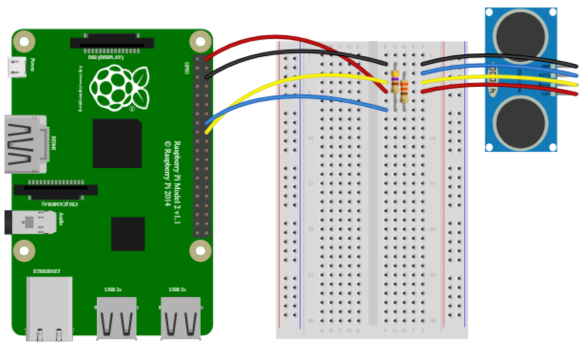

Figure 2

The issue with this sensor, however, is its minimum view distance. For the sensor to operate properly, it must be at a minimum distance of 20 cm. The remedy to this problem was to print a custom water bottle which moves the sensor up above the water bottle shown below in figure three.

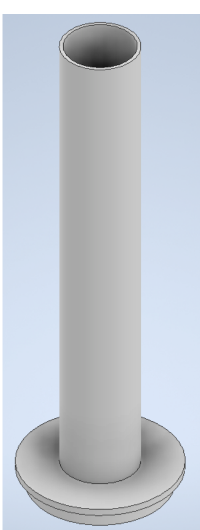

Figure 3

This in theory would fix the issue by moving the sensor physically up to within its viewing distance. In reality the sensor would only function if it was within the bottom 25% of the tube. A redesign into a cone shape would solve these problems.

As for the measurement system itself, the ultrasonic sensor was found to be incredibly accurate down to sub millimeter precision. For detection of automatic drinking, we wait until the water bottle has been stable for 5 seconds before deciding if the water level has changed or not. In the future, combining this with accelerometer data would allow for better detection of more situations, such as carrying the bottle in a bag and when the cap is off the bottle.

The web app design follows one familiar to all health apps, allowing one to view statistics about habits over varying periods of time. In figure four, I show a mock up allowing habits across the week as well as more exact daily results including a log of when water was drunk and comparison to a daily goal.

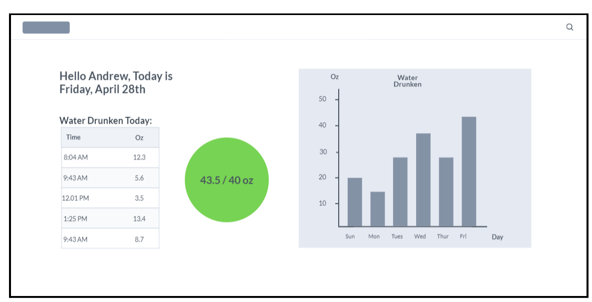

Figure 4

Overall, I would consider this project a success. I set out to showcase the potential for a lower cost, more flexible smart water bottle solution. Whether there is a market for a product like this is a different question but a solution is possible.

Related Works:

"Smart Water Bottles: The Ultimate In-Depth Guide - Impacx". Impacx, 2022, https://impacx.io/blog/smart-water-bottles/.