University of Southern California

ITP 348/ Professor Parke /Fall 2020

Developer Documentation

Daniel Masters

damaster@usc.edu

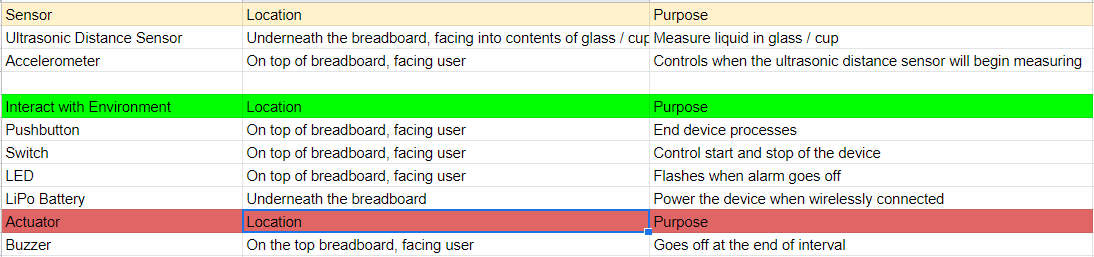
**OVERVIEW**

The fluid consumption tracker and reminder is a device developed around the premise of tracking an individual’s overall consumption of fluid and ensuring a user reaches their goal fluid consumption through consistent reminders. The user begins by utilizing the Particle cloud to input the amount of liquid they want to consume, the interval of time (in minutes) that they want to be reminded to drink (alarm will go off and light will flash at the end of the interval) and be able to control the device via a start/stop/pause feature. An accelerometer determines the orientation of the device and when the device is lying flat, it will allow the ultrasonic distance sensor to measure the distance to the surface of the fluid. With some mathematical calculations one can determine the amount of fluid in the cup. If the user wants to take a sip, they must remove the device sitting atop their cup and lean the device on its side (the accelerometer will register this as not lying flat and will not allow the ultrasonic distance sensor to take any measurements). If the user takes a sip before the end of the interval, the interval (using millis) will reset upon the user placing the device horizontally over the cup. If the user does not take a sip before the end of the interval, an alarm will go off (buzzer plays a jingle) and the LED will flash which alerts the user to take a sip. All information regarding the amount of each sip and the total amount of fluid consumed is sent to the cloud (Initial State) where graphics of this is displayed. If the user would like to refill their cup at any point, they must take the device off of the cup, refill the cup and press and hold the red button for two seconds. Holding the button for two seconds resets the millis timer and will measure the new amount of fluid without changing the overall sum of the fluid that has been consumed. When the goal amount for the user has been reached, exit(0) will be processed and a red blinking light will result and the device will automatically restart. A switch controls the on/off function of the device by allowing or denying power from a lipo battery attached underneath the breadboard of the device. The button allows the user to reset their device.

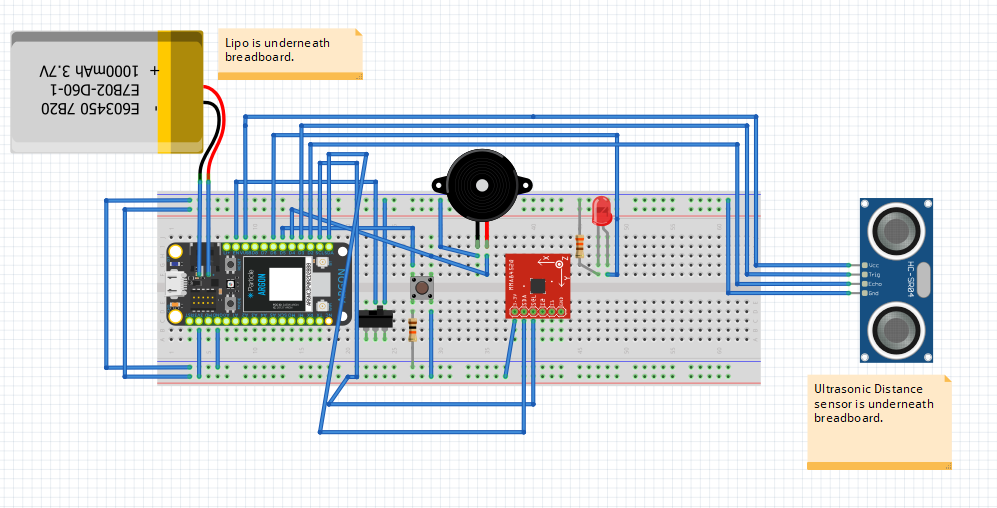
**IMPORTANT NOTE**

This device is only compatible with cups that have a diameter of more than 7.5 centimeters. This diameter measurement must be consistent throughout the entirety of the cup and not just the top of it. Failure to obtain a cup with such measurements will result in inaccurate ultrasonic distance sensor measurements and render the device unusable.

**LIST OF COMPONENTS**



**DIAGRAM**



**MODIFICATIONS TO BETTER THE DEVICE**

-A modification that could better the device overall would be to find a component that could diversify the range of containers that the device could measure the water level from. The device in its current state can only take measurements from one specific cup which is due to the ultrasonic distance sensor. I have done some research and found that the sensor is only accurate with cups that have a diameter of more than 7.5 centimeters which is the reason I had to use a large coffee cup.

-Another potential modification to the components of the device would be to replace the Sparkfun ultrasonic distance sensor with a smaller, more accurate. This will ensure more accurate measurements.

-A third potential modification would be to construct a device that, when placed over the user’s cup/bottle/glass, has a consistent position. This would assure more accurate ultrasonic distance measurements.

**ERRORS AND POTENTIAL FIXES**

-An error that needs to be fixed to better the device would be to alter the code such that the ultrasonic distance sensor begins measuring only when the device is over a cup/bottle/glass.

-Another error that needs to be fixed is the connectivity to the cloud (more specifically InitialState) as it is not immediately responsive when the device uses the LiPo battery as its power source. A potential fix to this would be to scrap the LiPo altogether or only use the LiPo for one specific task.