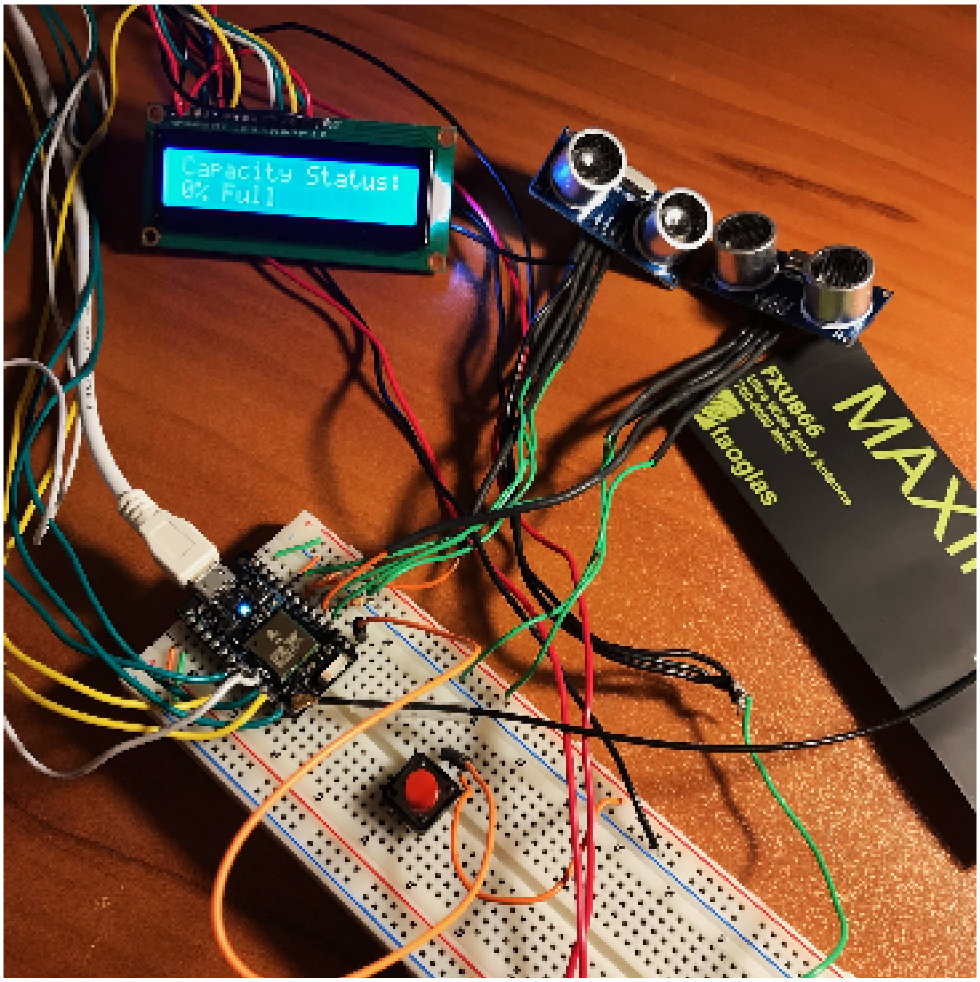
WasteFull



Luke Johnson,
Jonathan Skarda

About the Project

To ask a scary question: How much time, money, plastic and manpower is wasted, every year, on superfluous emptying of waste containers that could've gone another day, or week, before being emptied? This question brings to light a major problem with modern systems of waste management: they rely on someone being physically present in order to assess how full a container is, and if it should be emptied or not. Our device offers a way to break this dependency.

We reframed the question, "How full is a lidded container?" to "How close are the container's contents to its lid (assuming upright positioning)?" The project began with the need to be able to answer this question without opening the lid of said container or even being in any sort of proximity to it. To do this, we built a device that was to be mounted on the underside of a container's lid with distance sensors able to detect how far the top of a pile of trash was from the lid of the trash can containing it. The device's firmware would use the data generated by these sensors to determine, in terms of percentage, how full a container was, it would utilize cloud connectivity to make this data available remotely and even send a reminder to a user when the container needed to be emptied.

To assess this distance, between the lid and receptacle contents, WasteFULL utilizes two HC-SRO4 ultrasonic distance sensors. An HC-SR04 sensor emits an ultrasonic pulse and then detects the sound waves that bounce off any surfaces directly in front of the sensor. When the sensor detects this echo, a signal travels from its echo pin to a digital input. A program can be written to calculate the distance between the sensor and the surface reflecting the sound waves by recording the amount of time between a pulse and an echo, then dividing this value in half and multiplying it by the speed of sound (Last Minute Engineers, 2019). This information, however, is useless unless it can be communicated to a human. For that, there is our device's QC1602A LCD display. An LCD display forms images of characters by selectively illuminating pixels. A backlight, when powered, illuminates beneath a panel containing liquid crystals. A separate route for electricity is directed to this panel. When the liquid crystals within a pixel space are electrified, their shape and chemical properties are manipulated in a way that allows light from the backlight to shine through that pixel. By selectively electrifying certain pixels, a user (with the help of a digital device and software he/she probably didn't write) can generate letters, numbers or other desired imagery. Additionally, a potentiometer may be used to regulate voltages between the power source and pixels to determine display contrast (Woodford, 2020). Our device utilizes its LCD display to communicate to a nearby user how full the container is. Finally, to reset our device, we use a push button, which when powered, and then pressed, touches two pieces of metal together to conduct a signal to a pin that interprets this as a high signal (if wired active high). Software then responds to this event accordingly.

WasteFull utilizes cloud connectivity to access an interactive html website and interface with IFTTT. The website offers its user two ways of customizing WasteFull to their needs. First, it allows the user to enter their trash can height. This ensures that the photon converts distances read by the sensors to percentages of can height correctly. Second, is the ability for the user to determine how full the trash bin can get before the photon sends them an email, notifying them that their trash is full (for example, a user might want to be notified when their trash is 70% full instead of 90%). WasteFull employs IFTTT to send the email to the user once the trash has reached this user-defined maximum capacity.

Software Functionality

As mentioned earlier, the HTML website needed to have two functions. One, to allow the user to enter their own trash bin height, and two, to allow the user to change the maximum capacity. To accomplish this, the HTML website was set up with two forms to call the particle functions s***etBinHgt(string args)*** and ***setMaxCap(string args)*** respectively. The form allowed the user to set the ***args*** with a textbox. The style of the website was set up using a simple CSS file which simply set the font to georgia and added a background gradient.

The device's embedded program can be broken down into 4 fundamental sections: *data collection, data conversion, assessment and actuation*. Encompassed in *data collection*, the member function "getDistanceCM" of two objects of class HC_SR04 from the *HC_SR04.h* library gets the readings from the distance sensors. Web variables for the container's height (binHeight) and max capacity (maxCap) are updated from the cloud. In *data conversion*, distance sensor readings are converted to percentages of the container's height by "int setCurrCap()," while "int setMaxCap" converts the user's desired max capacity from percentage to a distance-from-sensor value. In *assessment*, the function "int WasteFull'' accepts the raw distance readings and compares them to "maxCap" to determine if the container is "full" and if appropriate responses are to be triggered or if the program should continue reading capacity and updating as normal. In *actuation*, an "if" block in "void loop" iterates once per second, printing the current capacity ("currCap") of the container, along with surrounding display message text, to the LCD display's screen (setCurrCap is actually called in "WasteFull" and fed an average of the sensors's readings). If "full" mode has been triggered, actuation remains within "int WasteFull" and only the "Max Capacity Reached!" message can be printed until the device is reset ("full" mode isn't a literal feature of the program, it just refers to what happens when the counter gets to 6).

When writing the code for this project we decided we wanted to make it modular so that it could be used with different sizes of trash bins. For example, we had to write a function that would take in a user-entered trash bin height and use that value to convert distance-from-sensor values to accurate percentages for the user's interpretation. We had to come up with a formula that could do this conversion. We also had to take into account that the sensors' distance readings were constantly fluctuating ± 2 cm. To account for this, we made the output percentage display in increments of 5% to stabilize the values printed to the LCD display. For this, we developed the formula:

Capacity_Percent = 100 - (100.0 / (Bin_Height - 10)) * (Sensor_Value - 10);

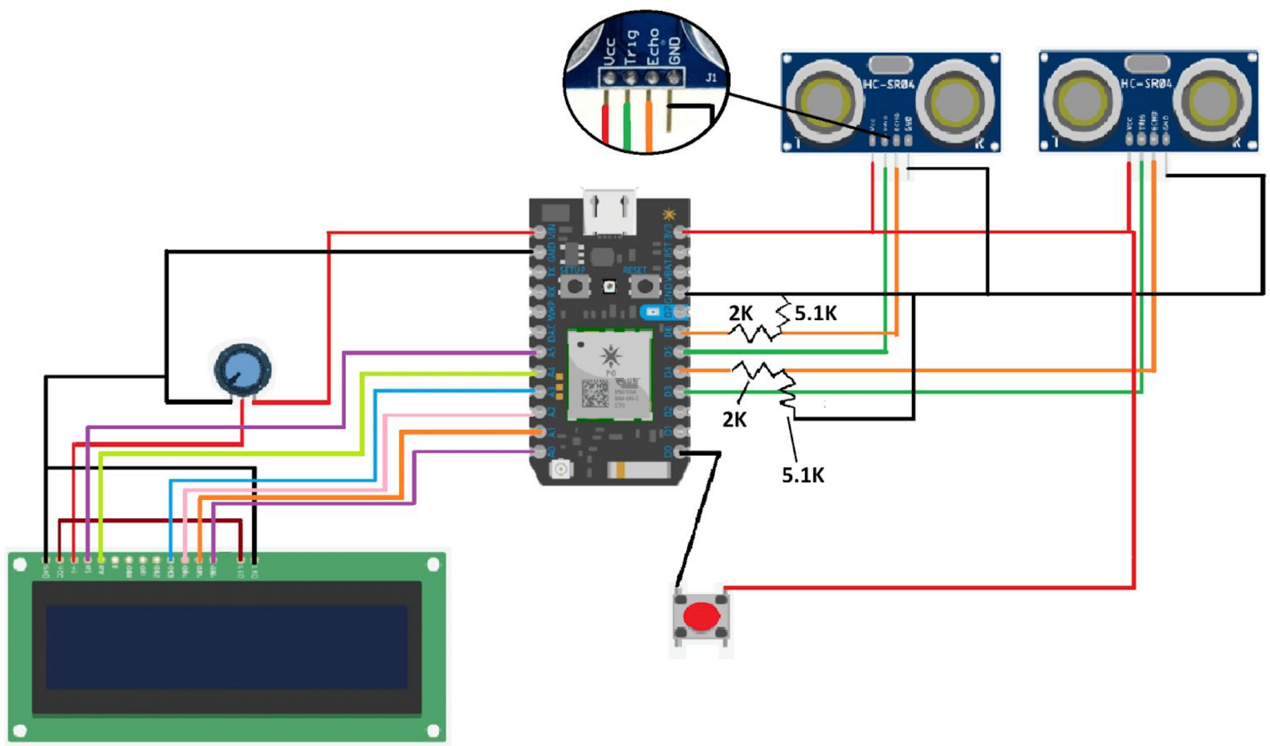
Capacity_Percent -= (Capacity_Percent % 5);

(the "%" in the formula refers to modulus)

We also had to find a way to prevent the device from sending more than one "full" notification email per instance of being full. To address this, we implemented a timer that would increase by 1 per second while the trash was being read as "full." If the timer reached 6 (6 consecutive seconds of "full" readings) a "full" alert was triggered. Otherwise, the counter would decrease by 1 until reaching zero for every second the sensors weren't reading "full," thus allowing for the lid to be opened and trash bag to be pulled out without triggering a "full" alert or without making the next alert-trigger period shorter. To prevent the device from sending multiple notifications per "full" instance, we wrote the program's timer mechanism to no longer increase or decrease after a "full." This way, the only way for another email to be sent after a "full" would be if the reset button was pressed, setting the counter back to zero.

WasteFull Schematic

Top Right:	Two HC-SR04 sensors
Bottom Right:	Reset Button
Bottom Left:	LCD display
Center Left:	Potentiometer
Center:	Photon



References

Woodford, C. (2020, July 08). *How do LCDs (liquid crystal displays) work?* Retrieved December 18, 2020, from *https://www.explainthatstuff.com/lcdtv.html*

Last Minute Engineers. (2019, December 13). *How HC-SR04 Ultrasonic Sensor Works & How to Interface It With Arduino.* Retrieved December 18, 2020, from *https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/*