Alex Kosla
EE5003 Hardware Notes
22261039

July 3rd, 2023

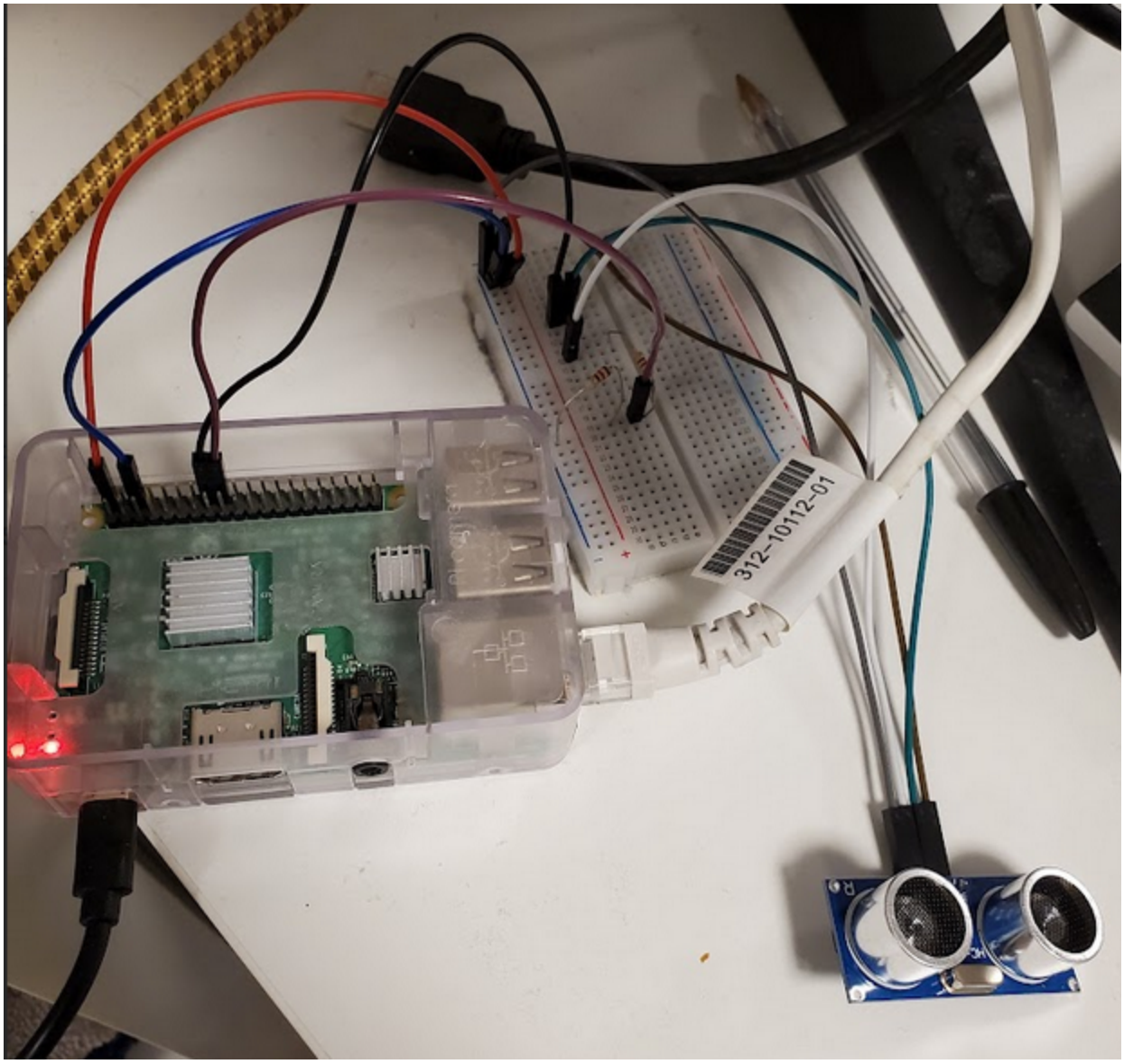
I have obtained the HC-SR04 ultrasonic sensor I'll need from Billy Roarty, my technical officer, along with some additional wires. I should have all of the basic parts I need to begin working with the sensor on my Raspberry Pi 3.

I also completed my end of the setup assessment form and will email it to Martin, my mentor, tomorrow.

July 4th, 2023

Today I am wiring the breadboard. I found a datasheet for the HC-SR04, but it generally lacks the information I'll need on how to wire it, so I have turned to youtube, where there are several existing tutorials on how to connect the sensor to a Raspberry Pi. Here are links to a few of the different videos:
https://www.youtube.com/watch?v=_7drIUmC8Zo
https://www.youtube.com/watch?v=O7x5r0P980U
https://www.youtube.com/watch?v=OTBIXnzcI34

Using the first video, I was able to get the code to more or less work. Here is how I wired it:



I am unsure of exactly how many ohms my resistors are. I obtained them from Dr. Molloy for EE513, and he seemed to imply that the exact resistances didn't matter much and they would be sufficient for protecting the GPIO pins. This circuit does seem to have a pull-up/pull-down setup that attempts to protect the pins, though I'm unsure how effective it is, as I'm not sure how what I have compares to the 1k and 2k ohm resistors specified in the video.

It is able to roughly display the distance, though it seems off by a few centimeters.

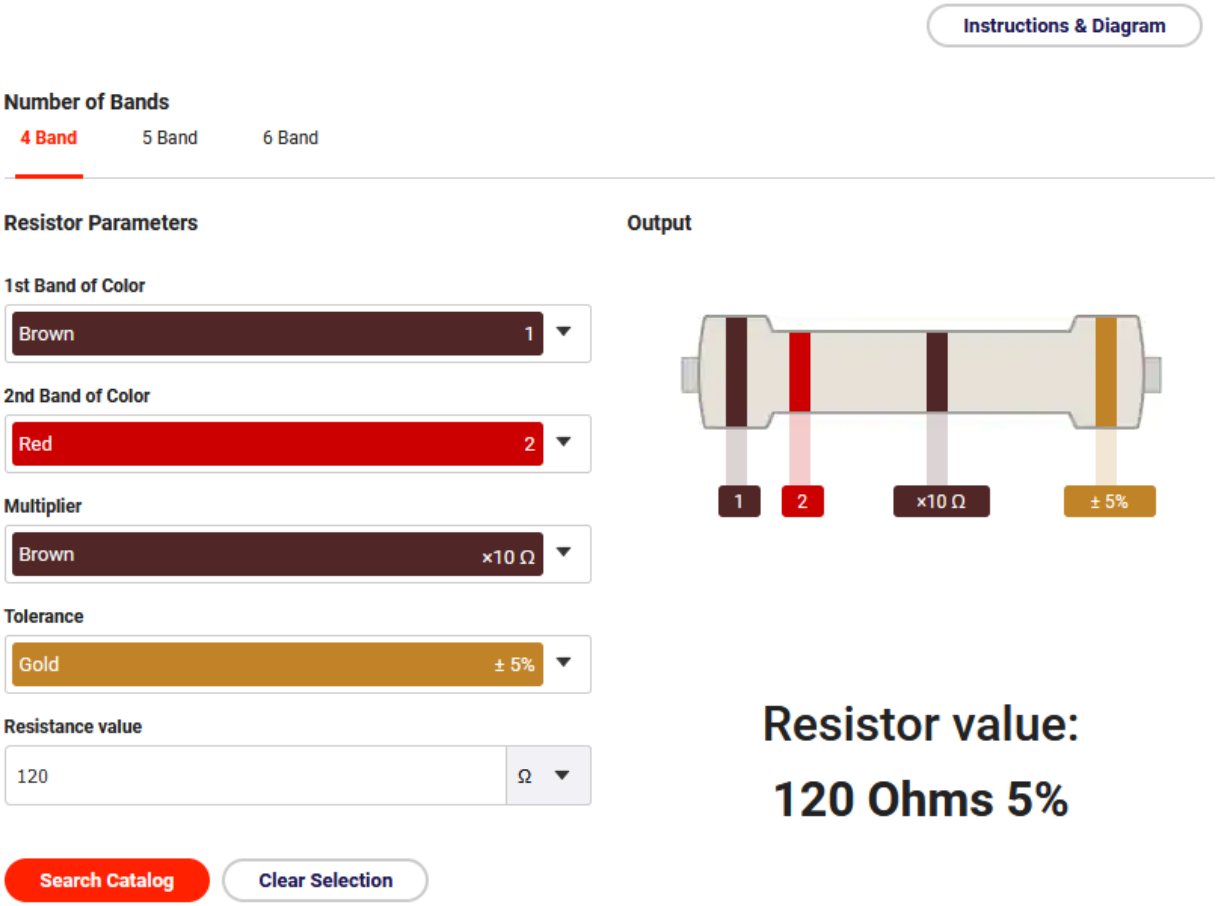
```
tuna@raspberrypi:~/trash/code $ python distance.py
Distance Measurment in Progress
Waiting for Sensor to Settle
Distance: 5.12 cm
```

The first photo is where my hand was for the first reading. The second photo is where my hand was for the second reading.

```
tuna@raspberrypi:~/trash/code $ python distance.py
Distance Measurment in Progress
Waiting for Sensor to Settle
Distance: 18.13 cm
```

However, this code is released without a license, so I'd prefer to find something on github that would be more robust and easier to cite/reference for the final product. Additionally, either the code's method of calculating distance is inaccurate or the sensor itself is inaccurate. I'll need to experiment and compare results with a ruler. [LATER NOTE: I DID NOT USE THE CODE MENTIONED HERE, I USED THE hcsr04sensor LIBRARY]

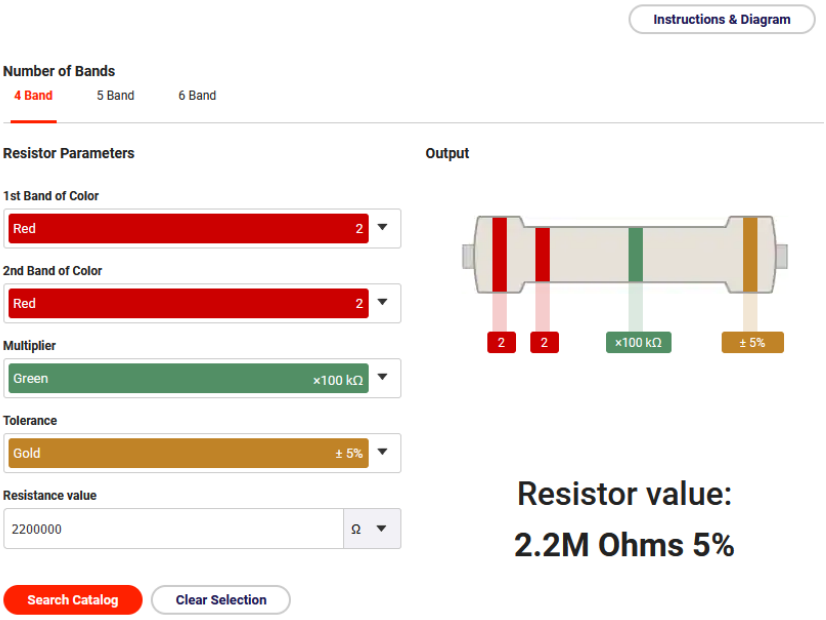
I looked up the specifications of the resistors I had and it seems they're both 120 Ohms, meaning they could possibly be not strong enough to adequately protect the GPIO pins. As such, I immediately turned off the raspberry pi and disconnected the power. I'll need to pay a visit to Billy to obtain 1k and 2k resistors.

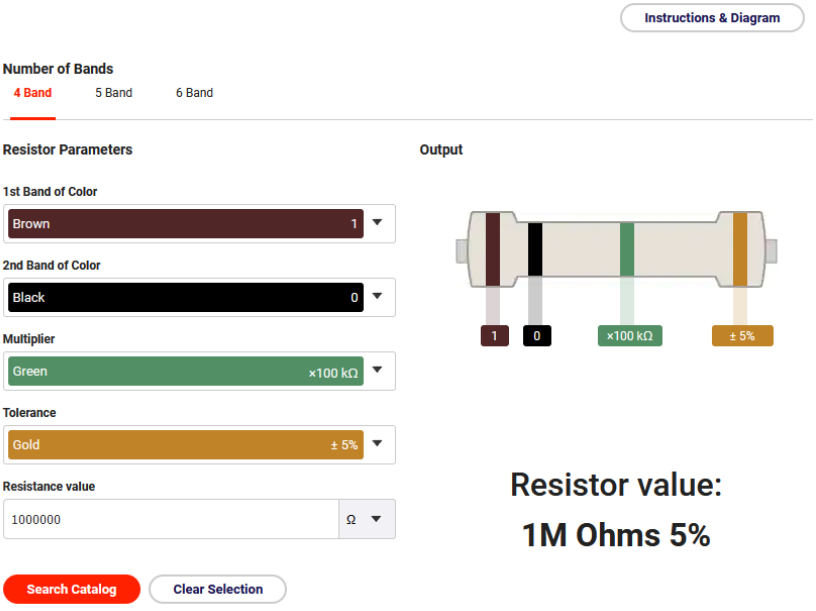


https://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-resistor-color-code

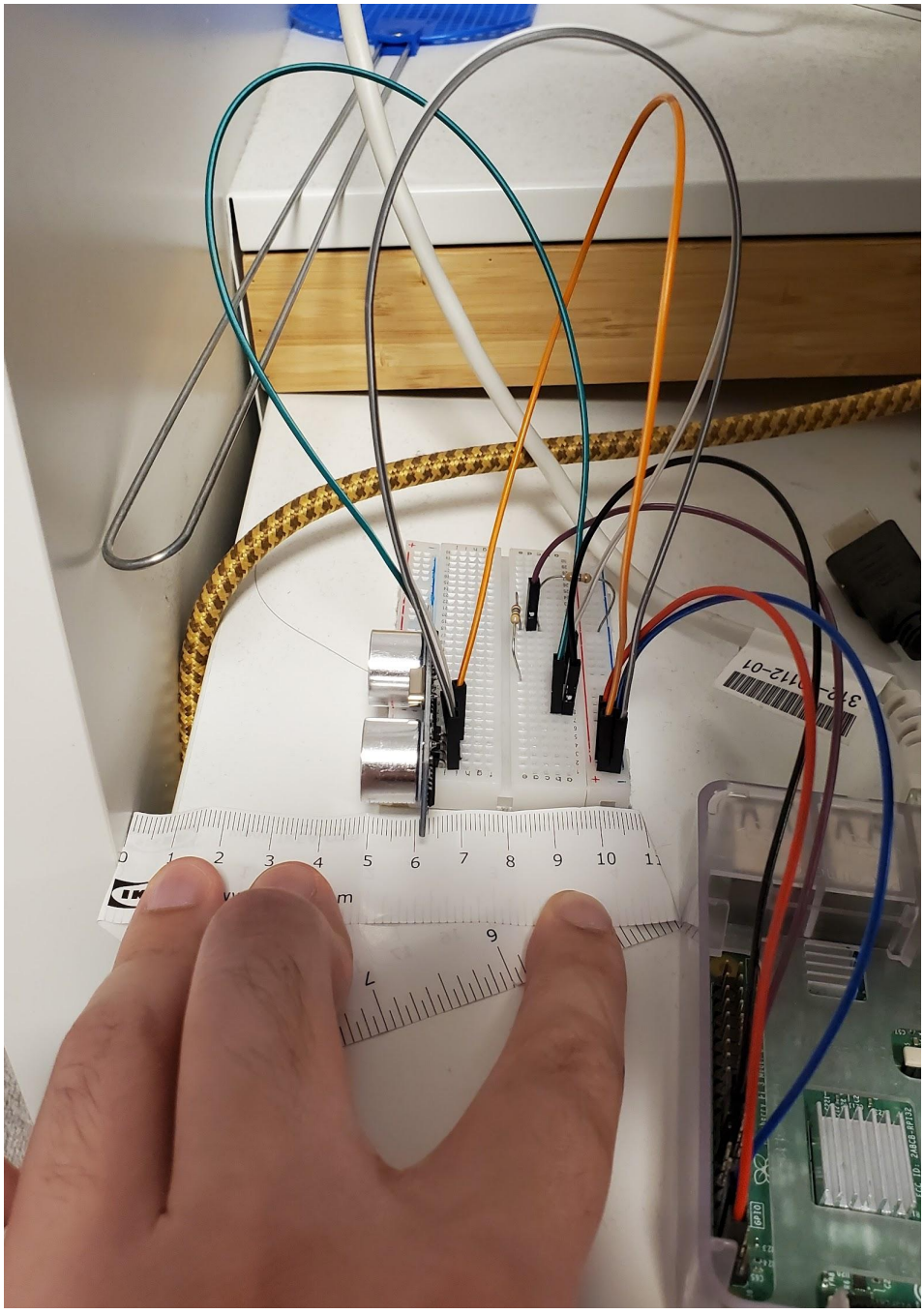
July 12th 2023

I obtained new sensors from Billy, but it seems I may have accidentally gotten ones a little too powerful: they're 1M and 2.2M rather than 1K and 2K. However, the sensor still seems to work, so I suppose the extra protection doesn't hurt.





I ‘mounted’ the sensor this time by attaching it to the breadboard directly. I used some measuring tape to reference how accurate it is. Here’s how it’s wired:



This seems accurate enough for my needs. It’s within 1cm accuracy.

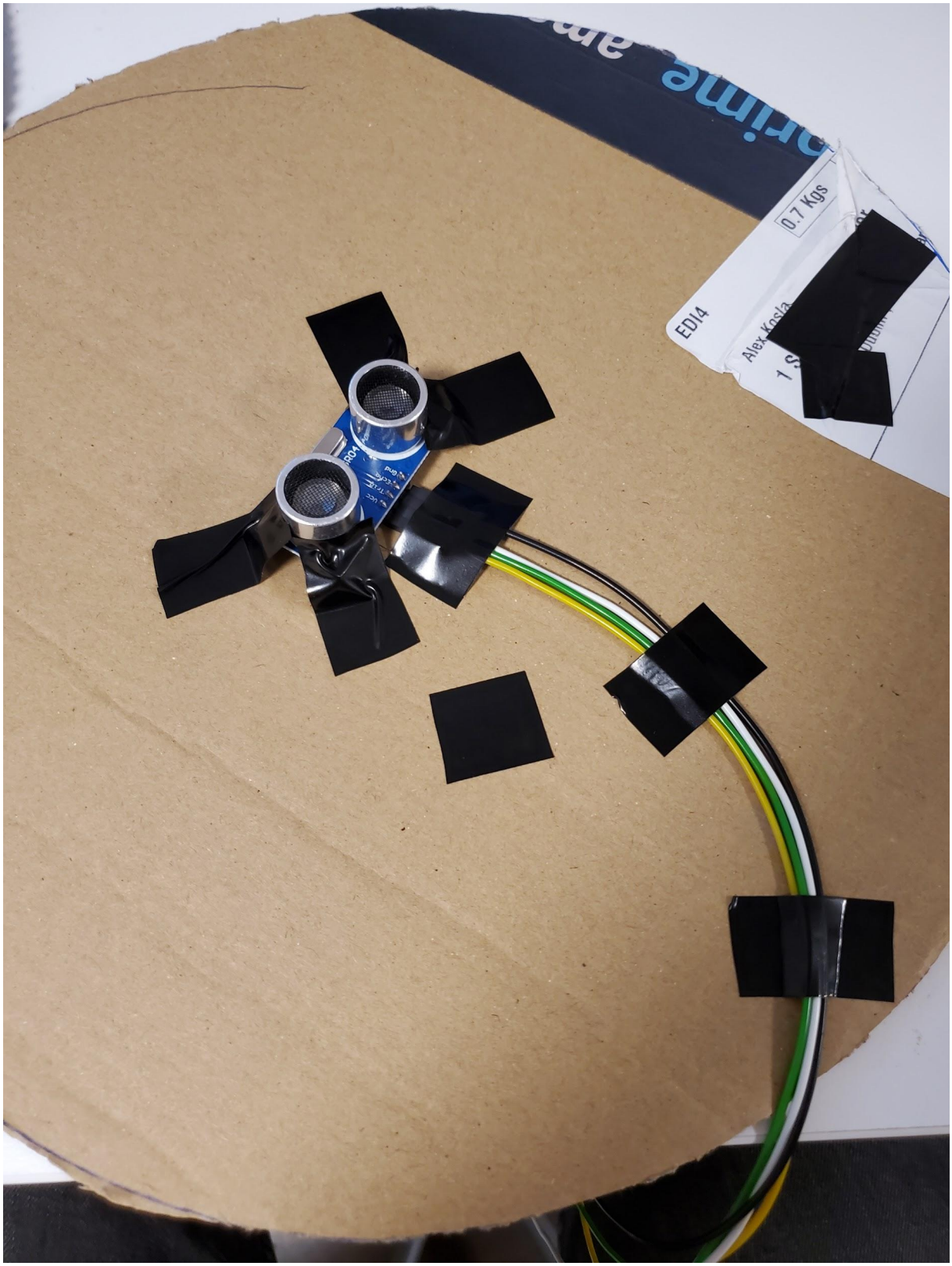
```
tuna@raspberrypi:~/trash/code $ python distance.py
Distance Measurment in Progress
Waiting for Sensor to Settle
Distance: 5.65 cm
```

Now, I'll need to find some github code if available that does this and extend it to add getting a timestamp and sending a post request.

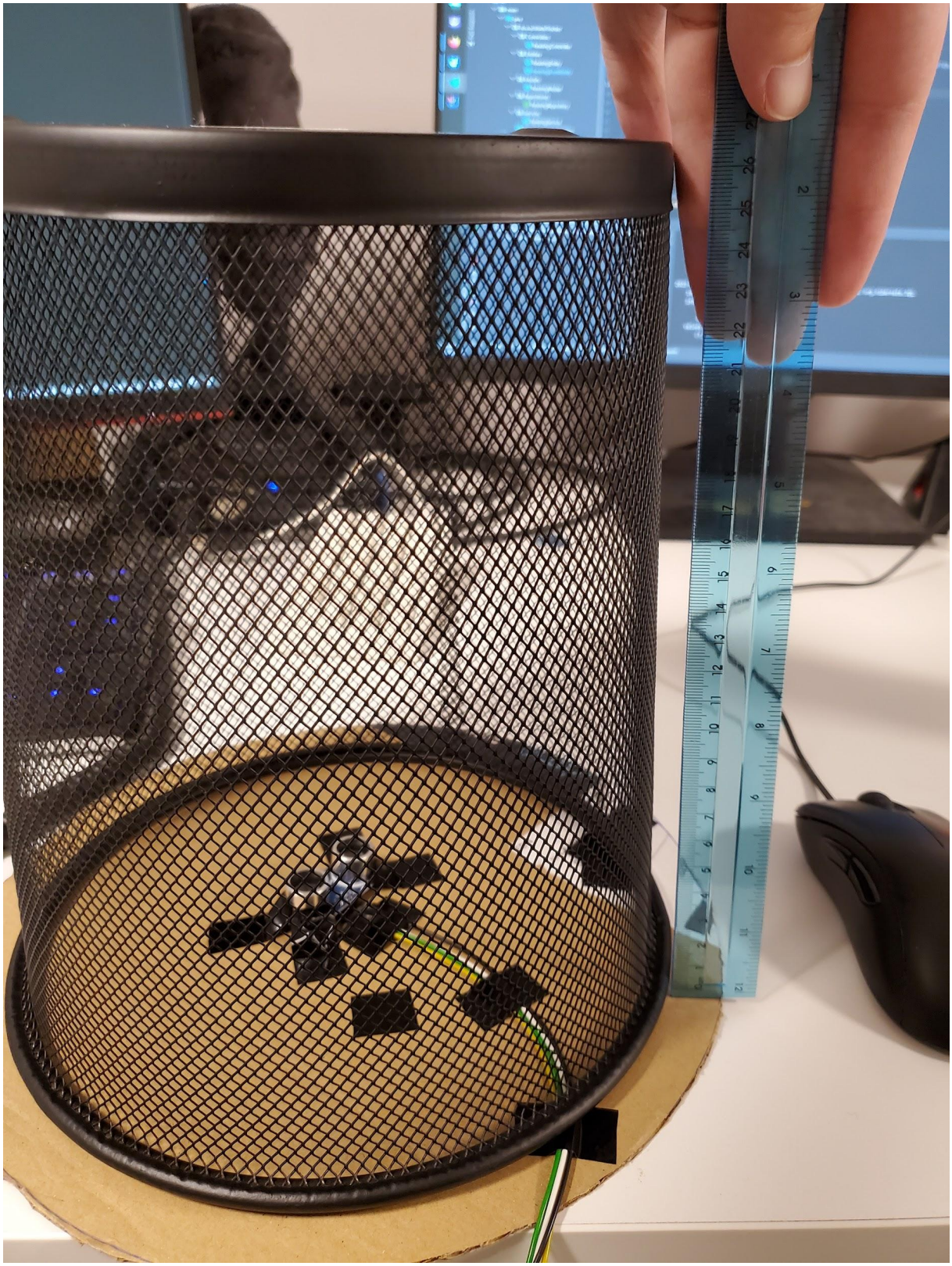
July 21st, 2023

I began constructing the smart bin today. I cut a cardboard circle using a bowl as a guide, so that the circle is slightly larger than the brim of the bin.

Next, I secured the sensor with insulated electrical tape to the cardboard lid. I connected the sensor to the breadboard via 3m long cables.



Next, to test the sensor, I placed the bin on top of it and measured the distance between the cardboard lid and the base of the bin, which appears to be approximately 26.5cm.



I tested that my reading.py code still functioned, and it does indeed.

```
the current time is 2023-07-21T21:25:37
{"distance": 24.970040824791337, "time": "2023-07-21T21:25:37"}
{"readingId":"503e1c39-68e8-4d00-83ef-e8c0d050cdcc","time":"2023-07-21T21:25:37","distance":24.970041}
```

After taking several readings, the sensor appears to be 25cm away from the bottom of the bin, so the bin fullness level will have to be recalibrated to match this.

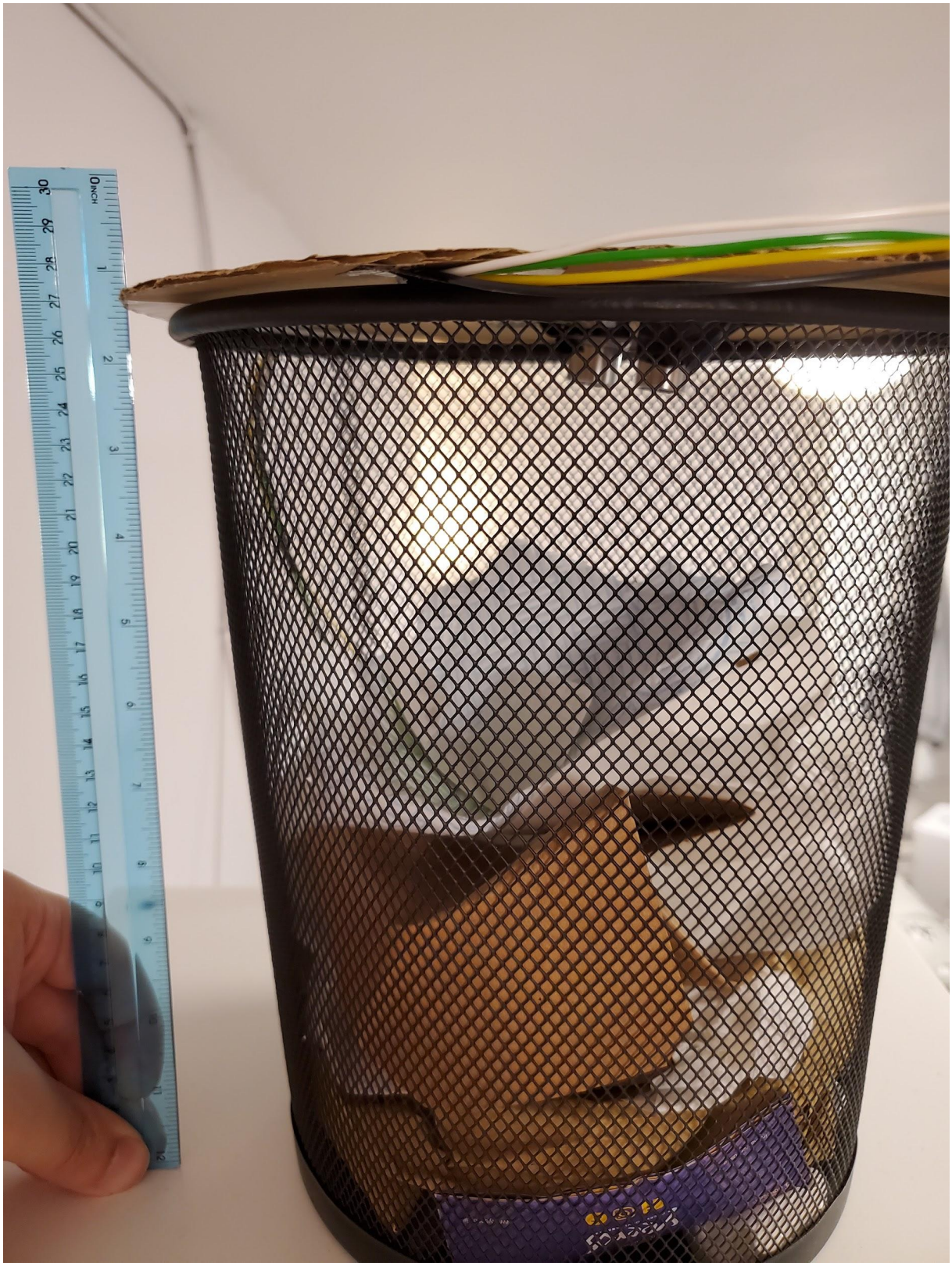
Reading_Id	Distance	Time
9f7cd7a3-f7fe-420f-bbf2-a0b61a3dd57e	25.0028	2023-07-21 21:30:02
2a9816df-e06d-472a-bfce-b9b3563abe36	24.97	2023-07-21 21:25:37

August 8th

Consistency of the readings

I am going to test the accuracy of the sensor. It has been running for several hours and consistently seems to be getting a reading of ~13.4 across 12 different time intervals today, all of them pretty close. The bin has not been touched manually, but hypothetically the contents could've shifted due to doors opening/closing and the breeze of an open window. The contents of the bin are all paper-based.

Reading_Id	Distance	Time
33bcde0b-324c-412a-9f22-a7b3a449464f	13.4608	2023-08-08 22:00:02
85829bfd-e2f7-4639-91d3-5fb168890067	13.4445	2023-08-08 21:30:02
90c722d3-22bc-4c8a-80df-2aba454b85c6	13.4363	2023-08-08 21:00:02
495c0825-60a5-4fc7-bb56-123e1c5a0c14	13.4199	2023-08-08 20:30:02
0b2a80c8-4794-4fc2-92e4-448fcf06aeaa	13.4854	2023-08-08 20:00:02
891f50b7-0a0b-4239-a77a-2b3de043c3b1	13.4199	2023-08-08 19:30:02
4cf85346-6a6a-4aa5-ba07-bbdd3303293c	13.424	2023-08-08 19:00:02
0349eed1-4ec3-463a-8b55-78ac72bf9c71	13.424	2023-08-08 18:30:02
b0dc9040-4569-48d4-b282-4b57e13078e9	13.4854	2023-08-08 18:00:02
f2886e73-244d-41aa-bbe4-a17d6a161809	13.4363	2023-08-08 17:30:02
fd006838-2831-40b4-a506-246b83b95a18	13.4363	2023-08-08 17:00:02
355b6045-a8db-4023-a3b6-620dac00e272	13.4772	2023-08-08 16:30:02



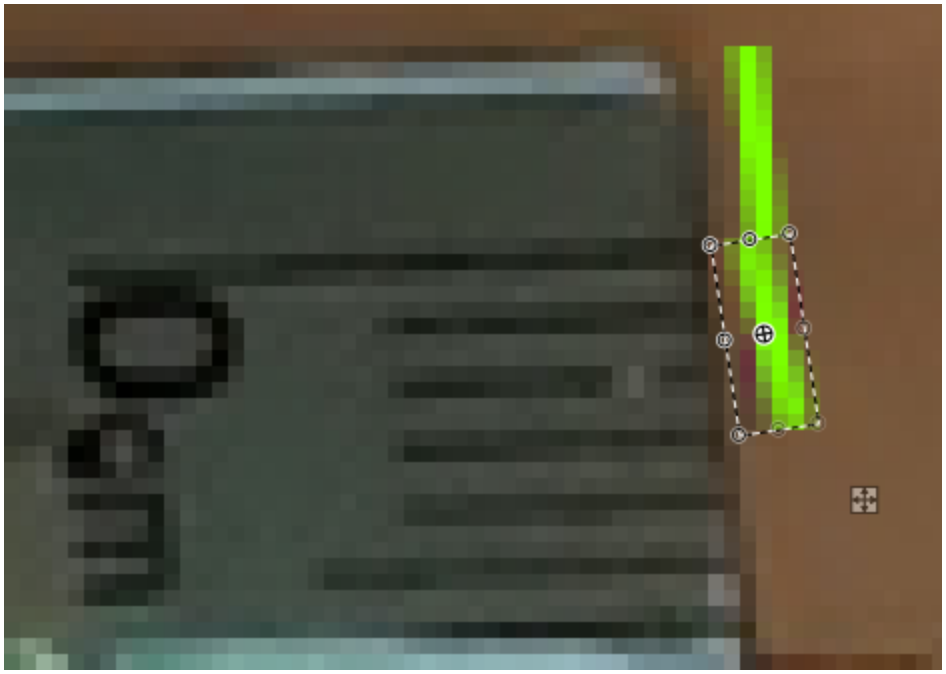
Avg	13.44583333
Stdev	0.02506683188

Median	13.4363
Size	12
Alpha	0.05
Confidence	0.01418263469
Confidence interval (lower bound)	13.4316507
Confidence interval (upper bound)	13.46001597
Variance	0.0006283460606

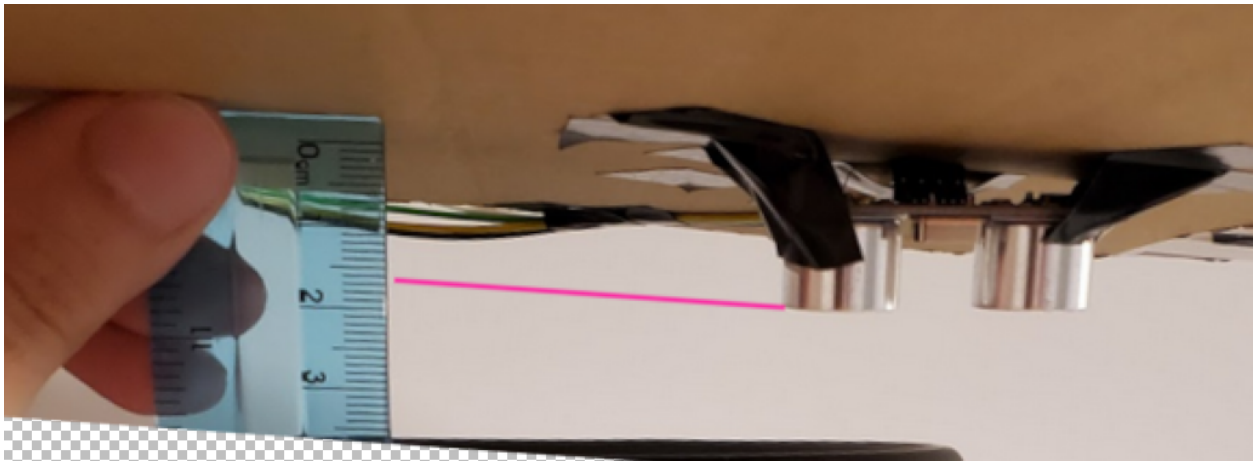
Adjusting the sensor height code

Due to the ‘sag’ of the sensor, the distance calculation probably needs to be adjusted to be 1.75cm for the sensor distance from the lid, rather than 1.5. I have changed that in the code now.

Using paint.net I estimated the 0cm line is 3mm after the edge of the ruler.



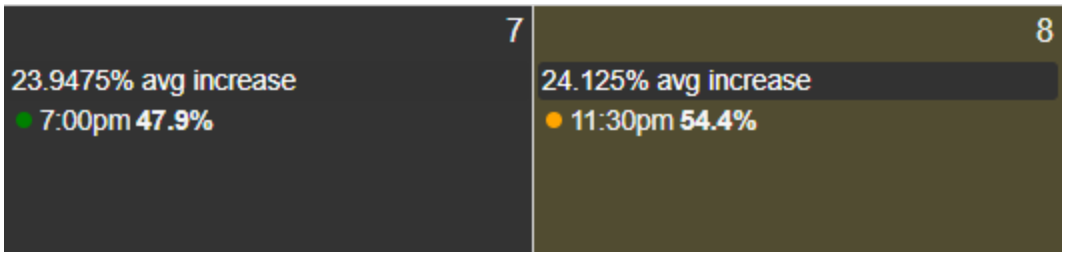
Using that information and rotating the image, I believe the sensor is now 2cm from the lid.



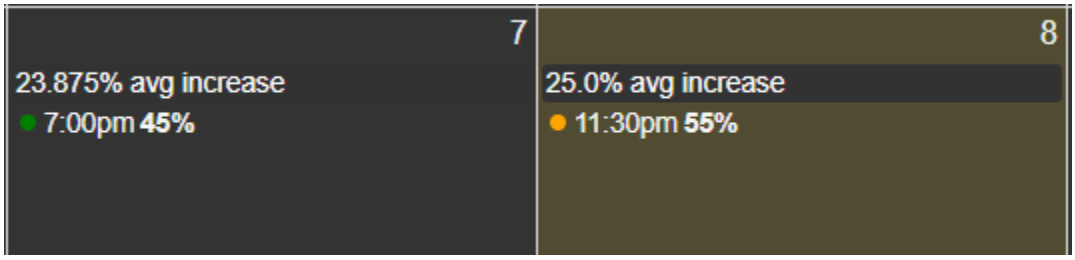
That means that the sensor waste I tested was approximately 12.5 - 2 cm from the sensor (10.5cm), but the sensor reading mean was 13.45cm. This could be due to the fact the sensor is not perfectly flat and could be pointing at a part of the bin at a different depth.

Rounding

I've decided to round the % full to account for adjustments due to noise. It will be to the nearest 5%. Before:



After:



Replacing the lid test

As I had taken off the lid and put it back on, I was curious how readings would be affected by it. I did this 15 times, trying to center the lid by eye, while attempting to not rotate it, as a way to measure human-induced error.

Avg	11.99718
Stdev	1.469787385
Median	11.4192
Size	15
Alpha	0.05
Confidence	0.7438013757
Confidence interval (lower bound)	11.25337862
Confidence interval (upper bound)	12.74098138
Variance	2.160274956

Rotating the lid test

Because the sensor is pointed at an angle, I next tested for how rotating the lid could affect the reading results. I performed the lid replacement test 15 times again, but this time I rotated the lid 180 degrees, trying to place the lid in the exact same position each time.

Avg	14.35439333
Stdev	1.249954267
Median	14.5041
Size	15
Alpha	0.05
Confidence	0.632552513
Confidence interval (lower bound)	13.72184082
Confidence interval (upper bound)	14.98694585
Variance	1.562385671

Lid replacement/rotation conclusions

You can see that while the stdev is relatively similar in both directions the lid is facing for the lid replacements, the averages are noticeably different: ~12cm vs ~14.4cm. In terms of my code, this means a difference between 51.06% full and 41.43% full, respectively.

```
>>> 26.5-2
24.5
>>> diff_dir_a = 24.5-11.99
>>> max = 24.5
>>> diff_dir_a / max
0.5106122448979592
>>> diff_dir_b = max - 14.35
>>> diff_dir_b / max
0.4142857142857143
>>>
```

The waste in the bin itself is neither particularly flat nor particularly uneven and thus could be more inaccurate or less inaccurate depending on the orientation of the waste in the bin. The waste in the bin was not attempted to be placed in a particularly flat/even manner, in an attempt to simulate a more realistic surface.

Plate lid replacement test

To see if smoothing out the surface of the waste to something more even would reduce variance, I took a ceramic dinner plate and used it to press down on the waste, to see if it would reduce the variance. I did this by putting the plate on top of the waste, pressing it down with my hand, and then placing a weight on top of the plate for 45 minutes.

Note that compacting the waste may not be a feasible solution, as doing so could be dangerous if the bin contains high-pressure waste, such as hairspray cans.

For this test, all measurements were between 13 and 17.5cm with a single outlier of 26cm.

Avg	15.31125333
Stdev	3.153022585
Median	14.3569
Size	15
Alpha	0.05
Confidence	1.595620265
Confidence interval (lower bound)	13.71563307
Confidence interval (upper bound)	16.9068736
Variance	9.941551421

Plate lid rotation test

Here I performed the same lid replacement test, except I did so after turning the lid 180 degrees a single time. The statistics widely vary between the two rotations, because in these readings there were more outliers, with readings ranging from 15 to 50cm.

Avg	20.2878
Stdev	8.444806162
Median	18.5547
Size	15
Alpha	0.05
Confidence	4.273583041
Confidence interval (lower bound)	16.01421696
Confidence interval (upper bound)	24.56138304
Variance	71.31475111

Plate test conclusions

These outliers seem to generally be incorrect, and are likely a result of the ultrasonic wave bouncing off a surface at an angle. The primary surface in the bin the ultrasonic wave would be bouncing off of was likely the folded-up paper towel in the center, which sits at an angle of approximately 30 degrees. Because the sensor is mounted at an angle, rotating the bin 180 degrees seems to have the sensor go from reading the surface at the top of the angled paper towel to a surface towards the bottom of the folded paper towel.

A more consumer-ready construction likely would have the sensor be mounted more evenly, but I should have at least some idea of how multiple sensors might ‘vote’ to average out and correct the reading of a very uneven surface. I’d probably put some object in there at a steep-ish angle and move the sensor around to various spots to see what it’d read.

Offset test

I cut a 23.8 cm circle of cardboard by tracing the outline of the rim of the bin. Next, I divided it thirds, marking a spot every 7.9cm. Using a 6.6cm glass for reference, I cut holes in which I would place the sensor in the cardboard. I spaced these holes to be 0.6cm from the markings dividing the cardboard into thirds.

Next, I placed the sensor into the middle hole of the newly constructed frame, with the intention of improving reproducibility. I took a reading in this position, then moved the sensor into the left hole, then the right hole, taking a reading each time. I repeated this process 15 times.

Takeaways:
Have multiple sensors “vote” and take the median of the sensor readings.