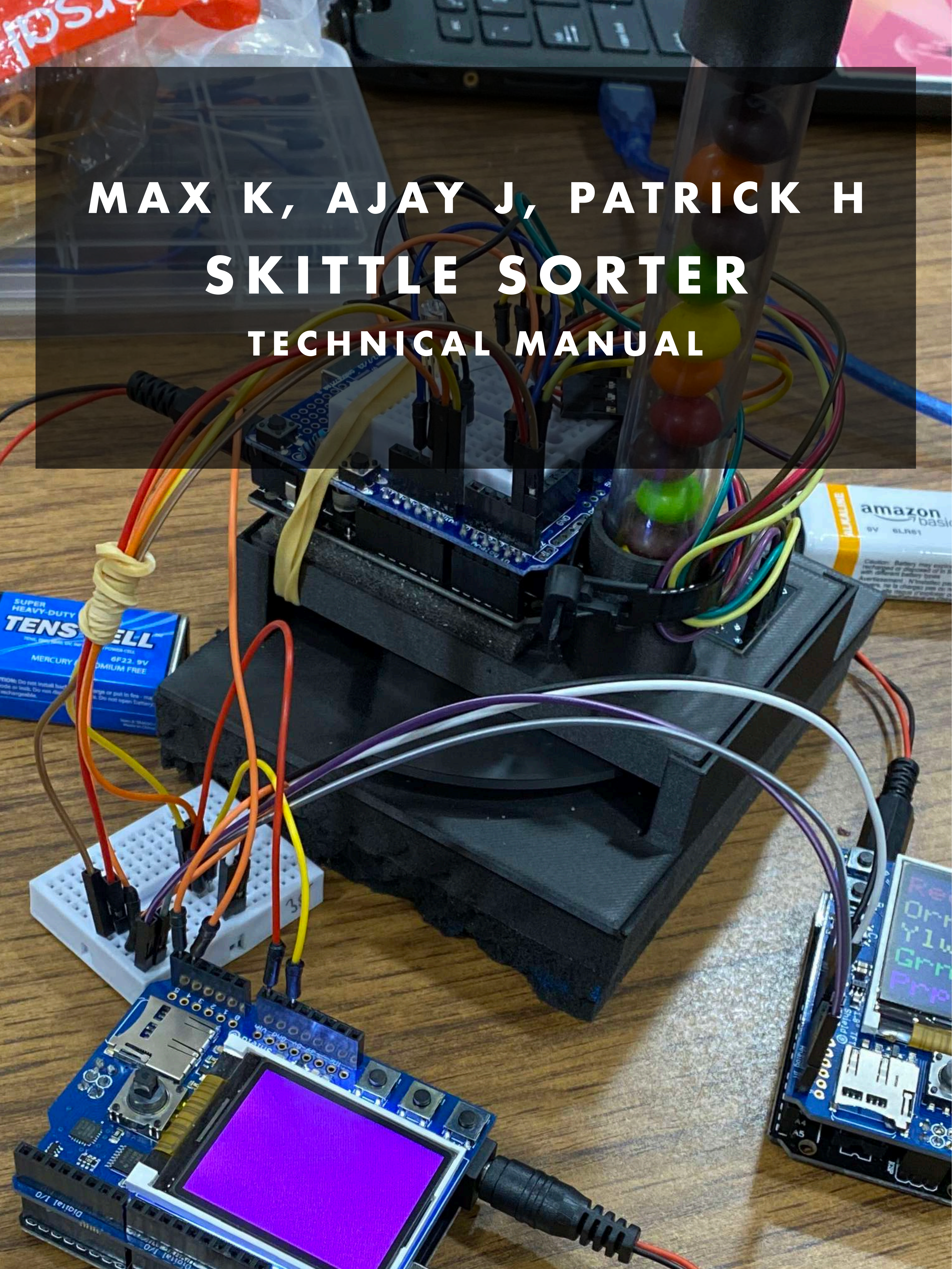
MAX K, AJAY J, PATRICK H

SKITTLE SORTER

TECHNICAL MANUAL



THE SKITTLE SORTER TECHNICAL MANUAL OVERVIEW

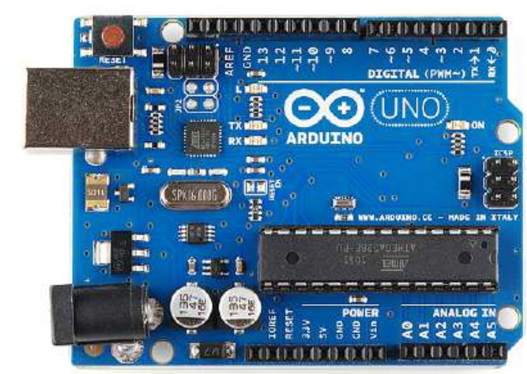
This skittle sorter technical manual serves as an official document that overviews how it works instead of how to use it. Specific Arduino libraries, terms and ideas/strategies will be mentioned so it is recommended these are known well before hand.

The skittle sorter is a project made by Max Knutson, Patrick Hultquist and Ajay Jayaraman to do just that: sort arduinos.

MATERIALS

Below is a table of all of the significant materials used, excluding all connecters.

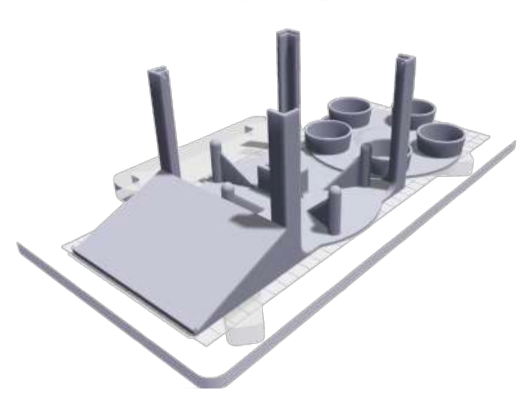
Arduino x3



Breadboard Shield x2

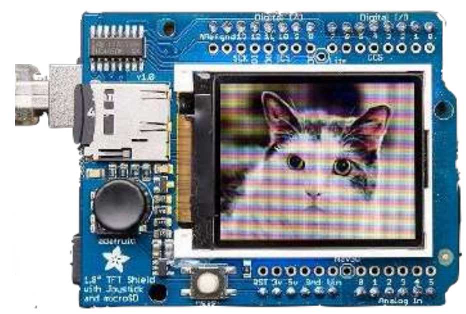
9V Battery x3

3D Printed Structures



Color Sensor x2

Adafruit Display Shield



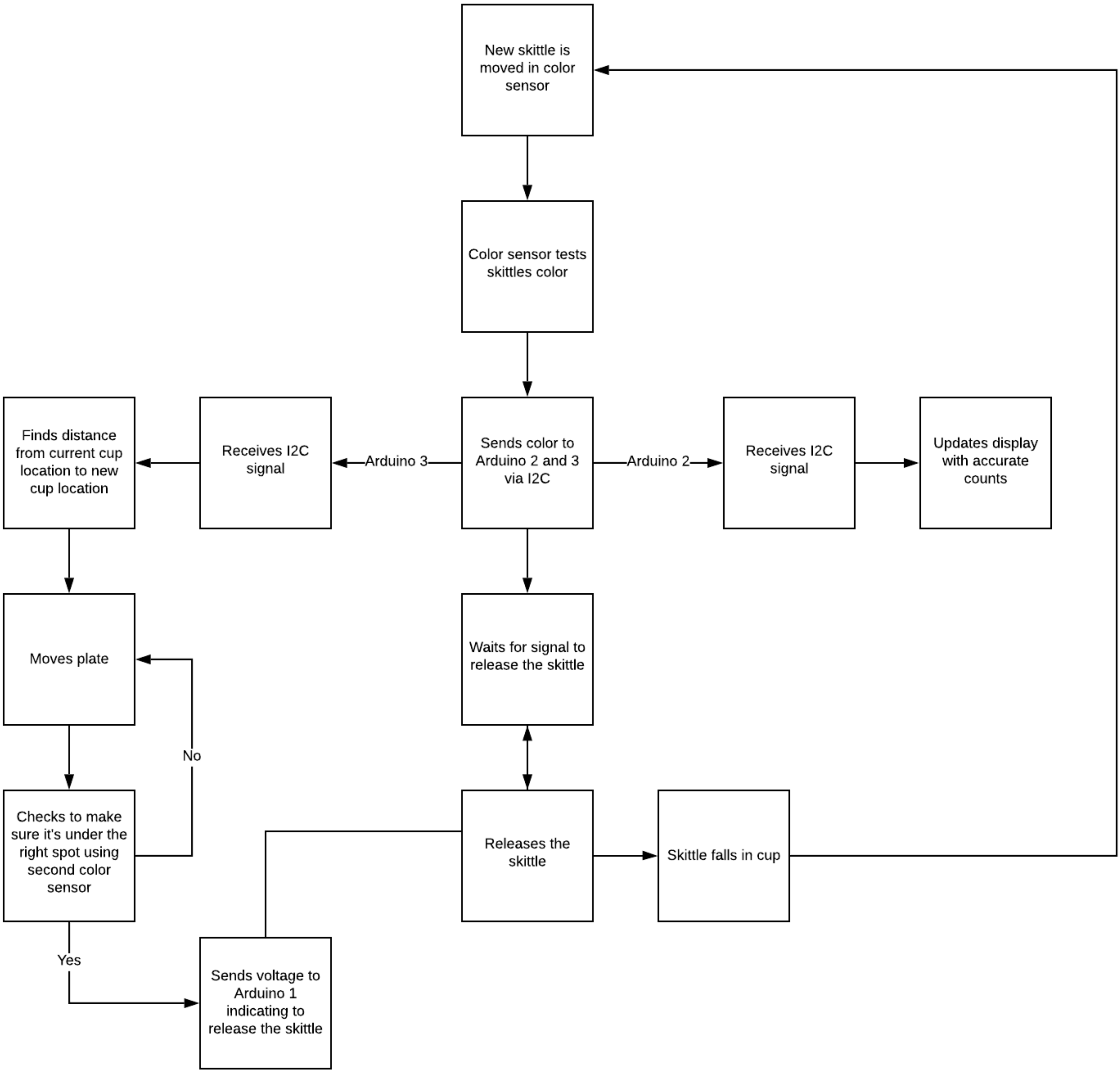
180º Servo Motor

360º Servo Motor

Transistor

Skittles

PROCESS FLOW CHART

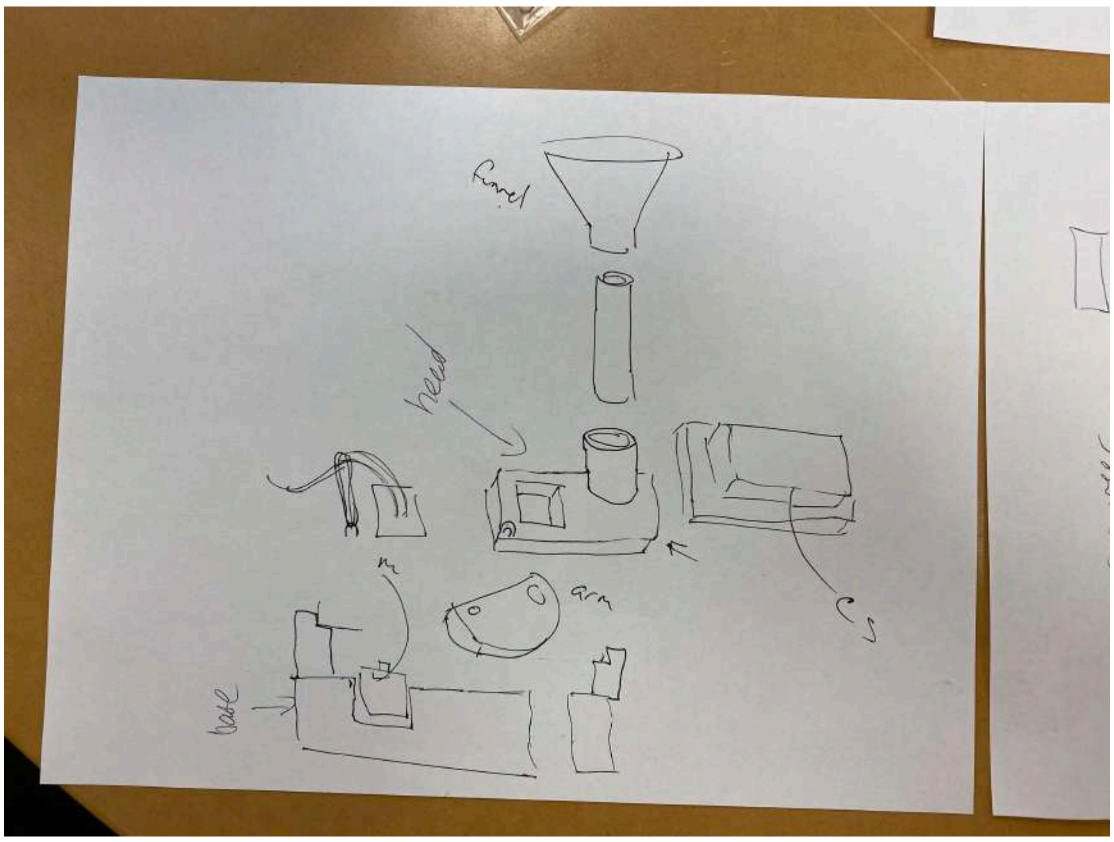


Step Name	Description
New skittle is moved in color sensor	Using the 180° servo motor, the arm is moved under the skittle chute, waits 500 ms, and then is moved under the spot with the color sensor.
Color sensor tests skittles color	The color sensor (TCS230 TCS3200) has a 64-color grid including 16 red, 16 green, 16 blue and 16 blank. Each rgb value is tested using the pulseIn() built-in Arduino function. There is a mode set for each color using pins S2 and S3 on the color sensor. LOW, LOW means red. HIGH, LOW means blue. LOW, HIGH means green. The frequencies are tested 15 times for red, green, and blue, and then averaged.
Sends color to Arduino 2 and 3 via I2C	Using the include <Wire.h> tag at the top includes the I2C communication library which allows two Arduinos to communicate using analog pins A4 and A5. Additionally, they can share power by connecting the VIN pin. Because I2C is easiest using integers, a code was developed. 1: red, 2: orange, 3: yellow, 4: green, 5: purple.
Receives I2C signal	In setup(), using Wire.receiveEvent(onReceive) ensures that any time something is sent to the slave Arduino, the onReceive function is called. To read the signal, Wire.read() is used. The skittle color is still in integer form.
Finds distance from current cup location to new cup	The last position of the cup is saved using the variable lastcolor. To determine how to get there, the Arduino finds the number of steps required to get to the cup using a clockwise rotation.
Moves plate	Using the 360° continuous rotation motor, the plate is moved. This one is complicated — the 360° motor is hard to control so I used a transistor to switch the 5V power on and off
Checks to make sure it's under the right spot using second color sensor	There is another color sensor (same make) underneath the plate, which measures the current position of the plate. This is done because the position of the motor is not entirely accurate. The second color sensor measures the position by looking at a colored disk underneath the plate. If it’s not in the right position, the plate will move a very small distance
Sends voltage to Arduino 1 indicating to release the skittle	After it has been proved that the plate is in the right spot, using digitalWrite(), a small voltage is sent to the main Arduino indicating that the movement is done and the plate is ready.
Waits for signal to release the skittle	The main Arduino is stuck in a while loop until Arduino 3 sends the mentioned voltage to pin A0 in the main arduino.
Arduino 1 releases the skittle	By moving the arm from underneath the color sensor to the little hole in the 3D printed piece, the skittle is released and dropped into the cup. It next goes back to the beginning
Updates display with accurate counts	There is a very useful display that shows the counts of each color of skittle. This Adafruit display is updated every time a skittle is measured. It can be reset by using the reset button on the display shield.

PROJECT EVOLUTION

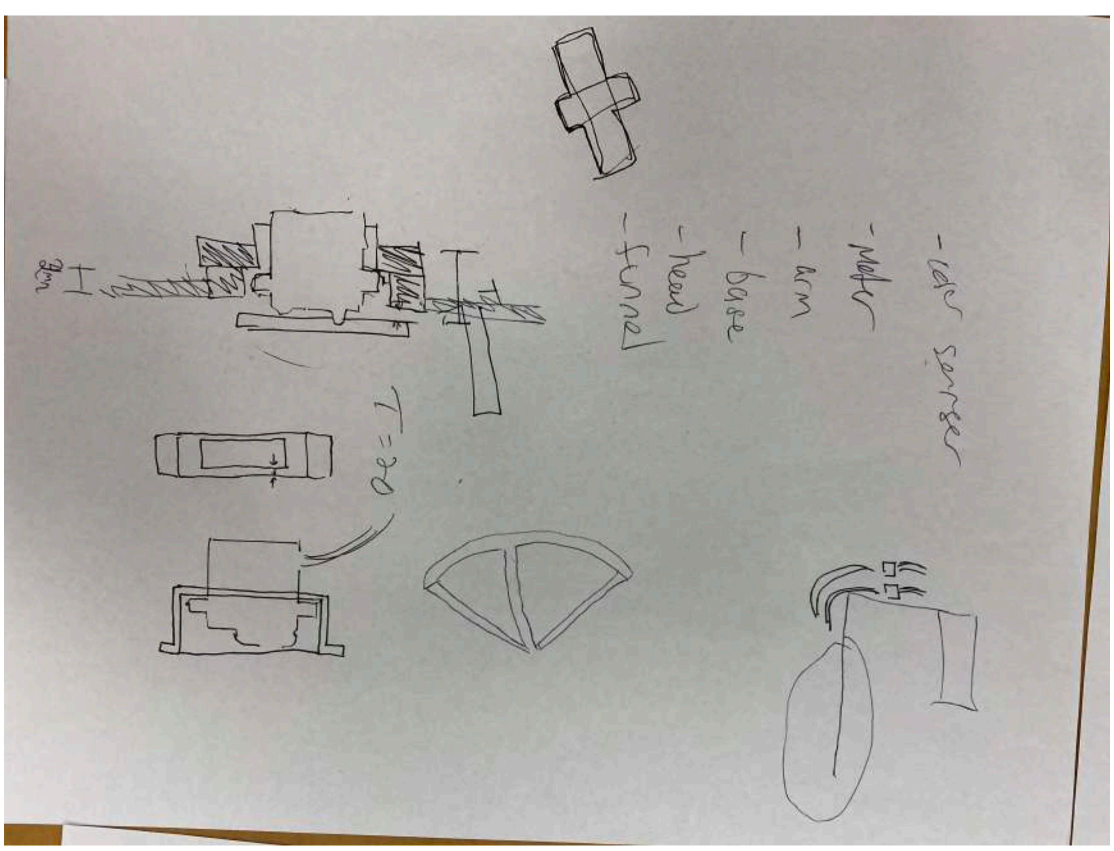
This project wasn't easy to complete, and was done in steps. Looking at each step one by one can tell a lot about how it was built in total.

Step 1: Sketching it up



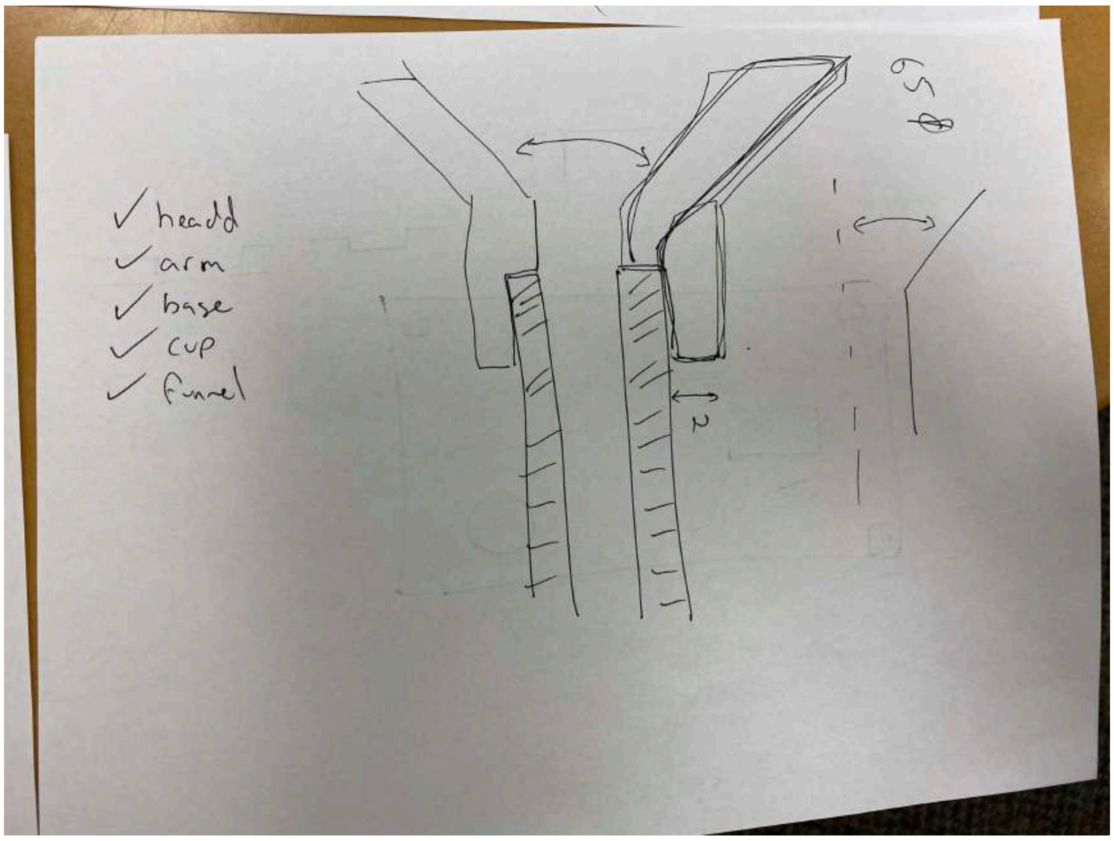
Sketch 1:

The whole system as it fits together including all of the parts. Pictured are early versions of the head, arm, and motor assemblies.



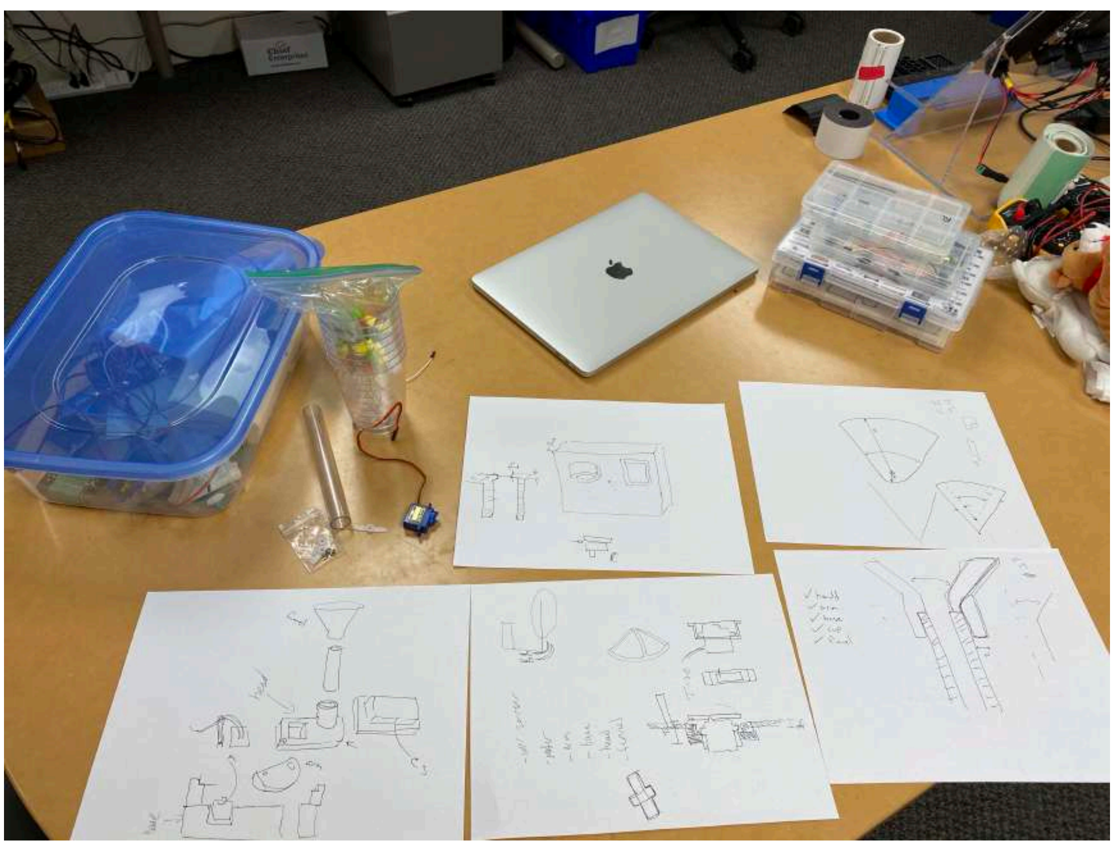
Sketch 2:

The top half of the project. Pictured on the bottom center is an early version of the arm. How the motor is to be positioned is on the left side.



Sketch 3:

The simplest part: the funnel. This sketch shows how the funnel fits into the system. Also note the checkmarks on the left for what had been designed in CAD.



Sketch 4:

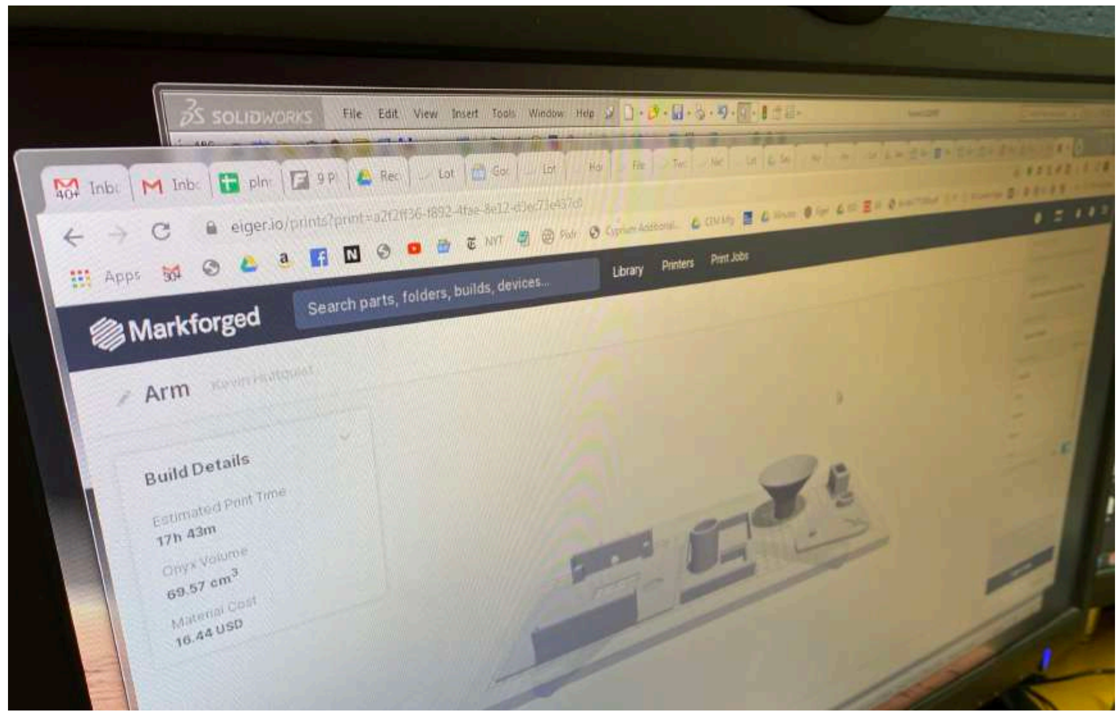
All the 5 original sketches put together on the workbench on November 17, 2019. Note many of these are double sided and therefore could not be recorded.

Step 2: Designing the parts in CAD and printing

Every part of the system — from the small motors to the big base — were designed for 3D printing. SolidWorks was used for the modeling and the Markforged 3D printer.

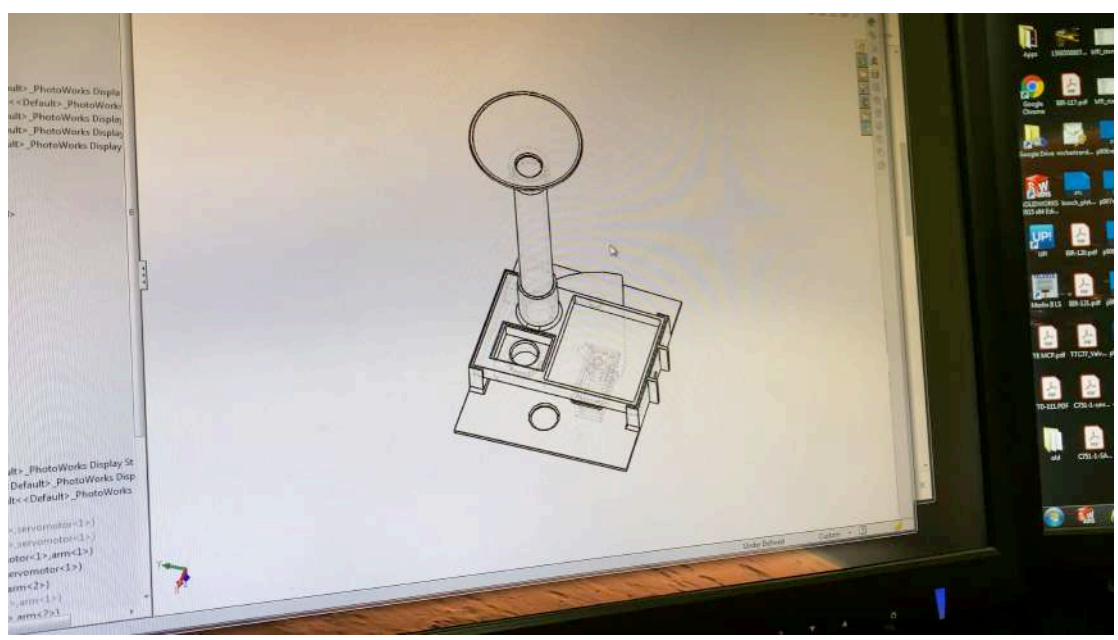
Sketch 5:

Some of the parts lay out on the 3D printer software to be 3D printed. This was the first batch of 3D prints so it does not include the bottom assembly which is the plate and the foot



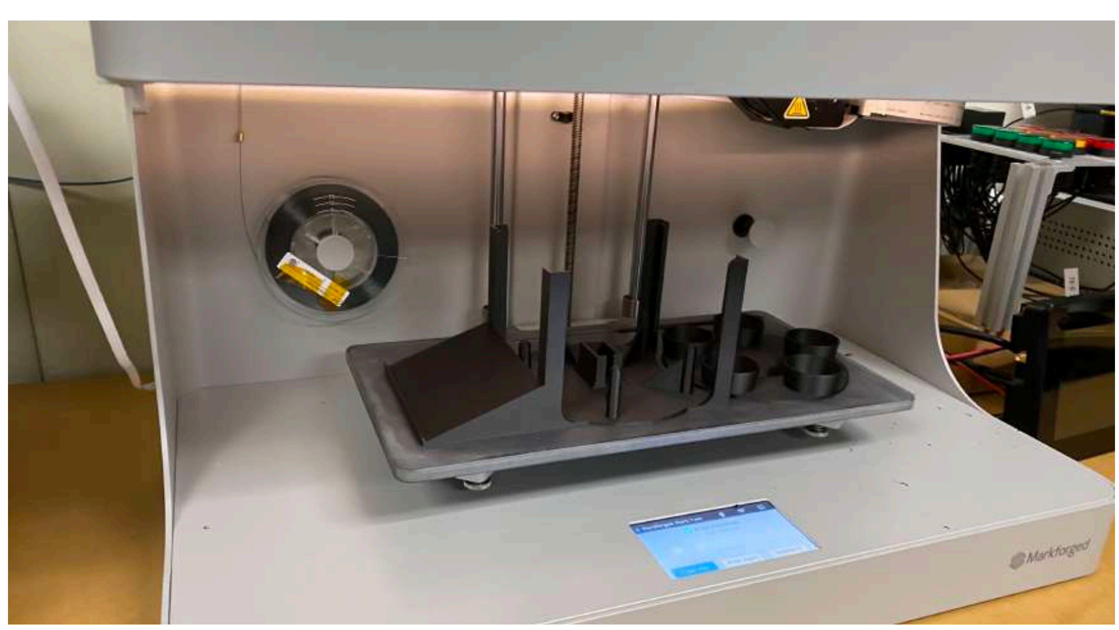
Sketch 6:

The top assembly modeled to ensure everything is all set before 3D printing



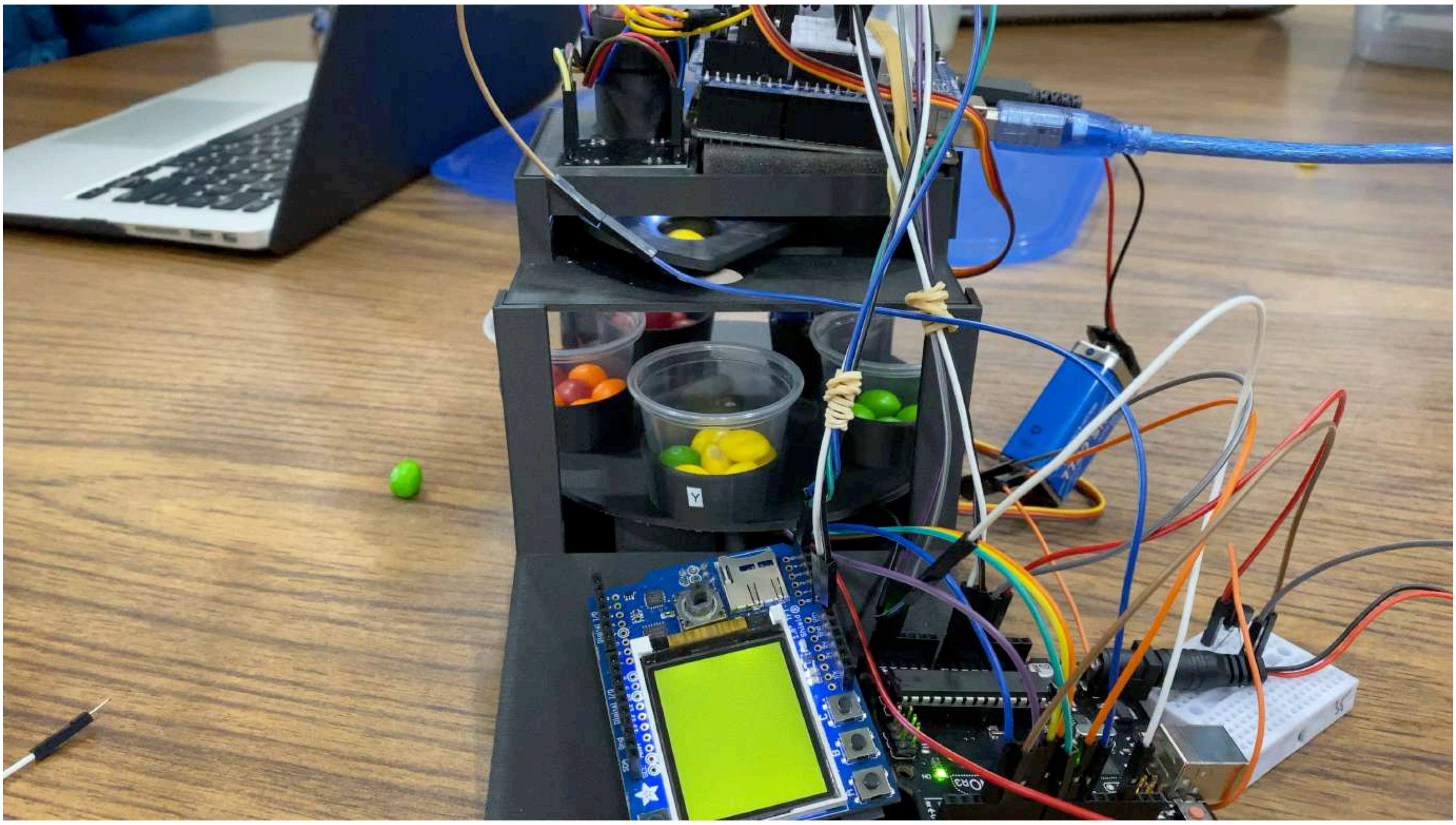
Sketch 7:

The bottom assembly laid out on the 3D printer after printing



Step 3: Putting it all together with code

The code started out as just reading a value from a color sensor and progressively became more advanced as time went on, controlling multiple motors and interacting with several Arduinos, as well as one of them controlling a display.



Sketch 8: One of the prototypes (with error, of course) where the yellow skittle is mid-fall after detecting the right color. The correct display had not been set up.

APPROXIMATE COST BREAKDOWN

Considering everything that has gone into this prototype, it costs a lot. Of course, it could all be optimized down to one Arduino using a larger one with more pins. An Arduino Mega, for example, has 30 digital pins compared to our 13.

Arduino x3	$45
Breadboard Shield x2	$15
9V Battery x3	$6
3D Printed Structures*	$100
Color Sensor x2	$20
Adafruit Display Shield	$35
180° Servo Motor	$2
360° Servo Motor	$10
Skittles	$2
Transistor	$0.50

*3D Print cost about $50 in material and $50 in maintenance and printer costs

This brings the total cost of the prototype to about **$236!** If this were to be mass produced, I would expect we would be able to get the cost down to about $20/piece using a real circuit board. We would also bring the cost down using a mold instead of 3D printing. There are a lot of changes we could make to bring the cost down, but this prototype works well.

OPERATIONAL SPECIFICATIONS

The skittle sorter takes up a lot of power. There should be 3 9V batteries plugged into each one of the Arduinos. Otherwise, three computers plugged into the three Arduinos should work the same way.

Safe operating temperatures are from -25 °C to 70 °C.

SOURCES

Images and STL Files
https://drive.google.com/open?id=1EUO62IufS50-IFspevoKs43_qVaB29Eh

Code
https://github.com/phultquist/skittlesorter