3.3.1 Marble Sorter

Timmy Pollard-Grayson

Balasubramanian Parthasarathy

Owen Benner

Steven Szostak

POE Block 2

3/29/16

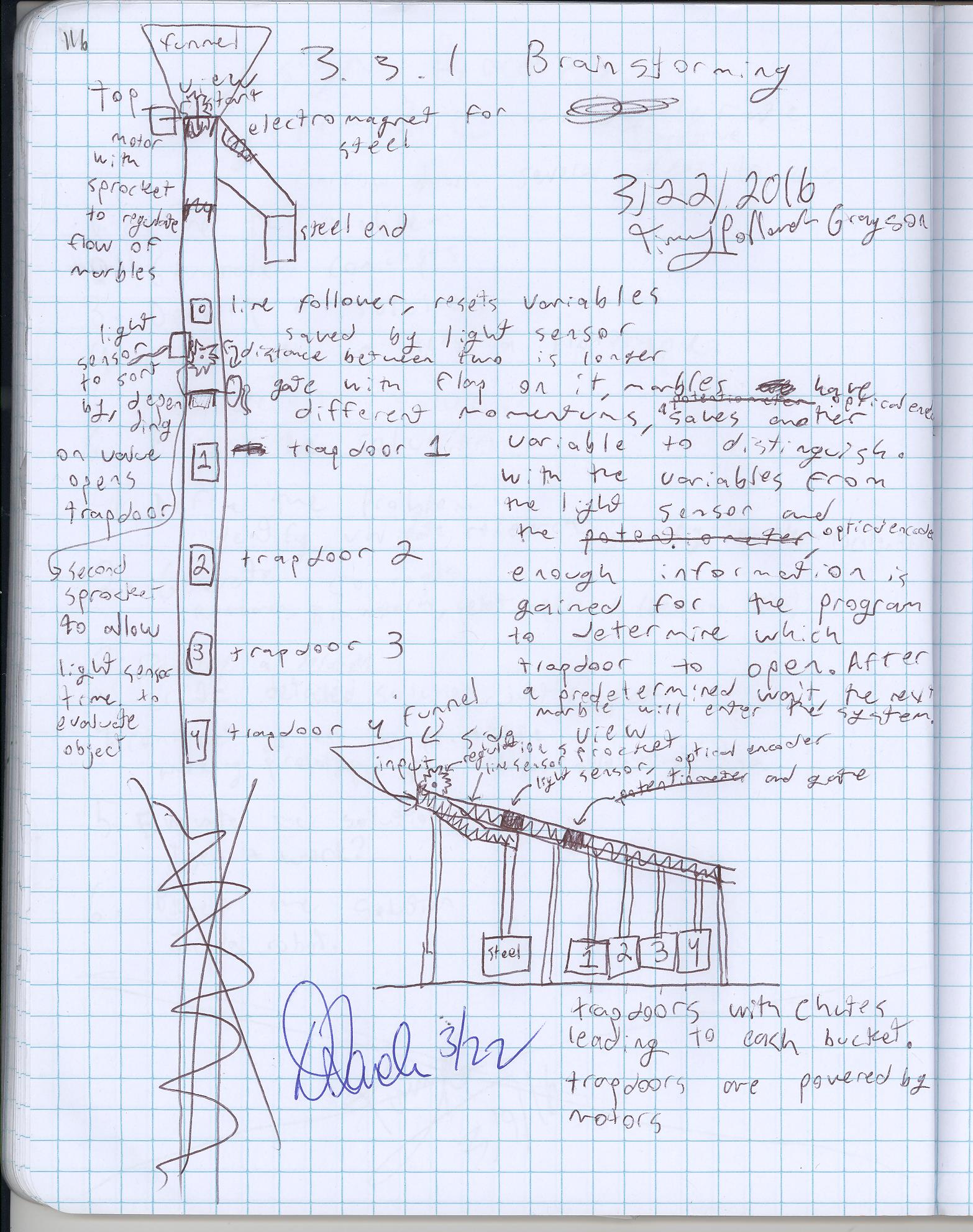
Design Brief and Constraints

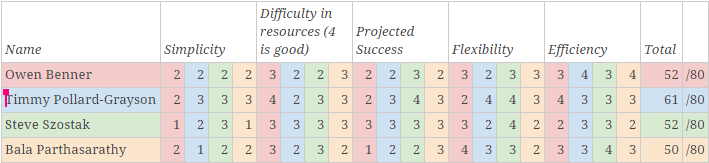
The National Recreation Park Association (NRPA) has asked your team to develop a solution to a growing problem in their parks. The association has placed dumpsters for recyclable material throughout the parks. They have a sorting facility; however, they need the device that will sort the recyclable material.

For this project, we were tasked with creating a machine which will separate marbles from each other. The process needs to be fully automated and made out of VEX components, and the end product will have all the marbles in different, separate bins, while maintaining complete control during the sorting process. There are five different types of marbles (wood, aluminum, steel, opaque plastic, and clear plastic), but we only need to sort four for complete credit. Finally, the machine must be able to sort sixteen (or twenty, for extra credit) marbles in under two minutes to complete the task.

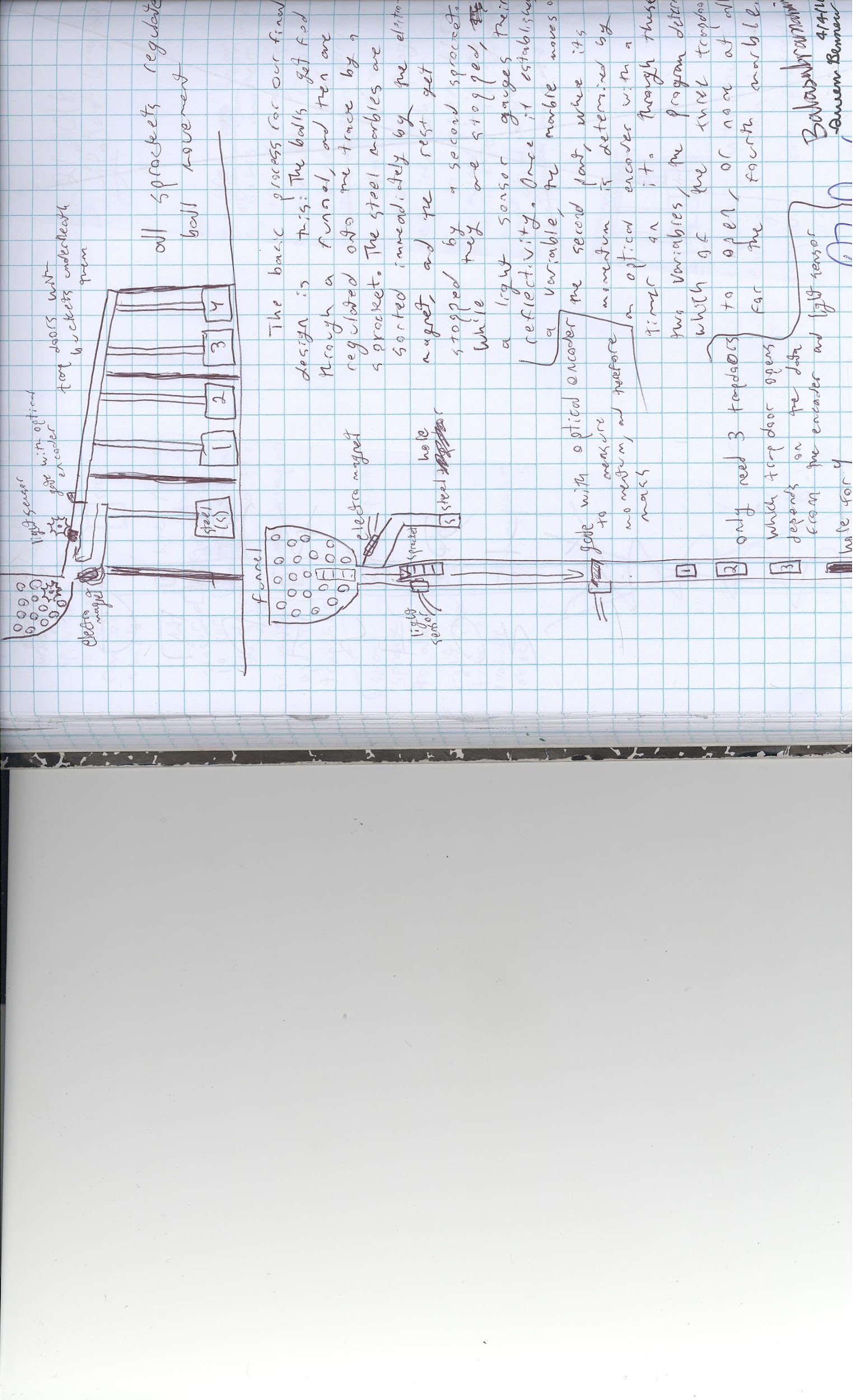
First Brainstorm

|  |  |
| --- | --- |
| This is the first brainstorm that I came up with to solve this problem. The initial solution involves using a gate with a flap on it to measure momentum using the optical encoder (Each marble has a different weight, and therefore momentum, so when the marble hits the flap, each marble will send it up a different amount, which can be measured by the optical encoder), and a light sensor to collect readings (reflectivity of marble, which is specific to each marble), and finally trap doors to guide the balls where to go. Once, the variables are obtained, the program determines exactly which trapdoor to open. The marbles are individually fed into the system by a sprocket attached to a motor.  The description reads: “With the variables from the light sensor and the optical encoder, enough information is gained for the program to determine which trapdoor to open. After a predetermined wait, the next marble will enter the system.” |  |
|  |
|  |  |
|  |



Decision Matrix This is the decision matrix we used to determine which idea to use for our final project. We evaluated each one carefully on simplicity (If the final product would be a simple, yet effective, design), how much material it would use (Less material means a faster build time, and so more time to tweak and fine-tune), projected success (Whether or not we believed it would actually complete the task), flexibility (How easily we could modify the design), and efficiency (Assuming it could complete the task, how long would it take?) to determine which one we wanted. We selected these criteria because we believed that the perfect combination of these factors would yield the best product. In the voting process, we each went through every idea, even our own, and critically analyzed it and rated it from 1-4 on each criteria.

Final Sketch

This is the final sketch that we created after evaluating each one. We removed a trap door and re-evaluated the design, to see if we could spot any subtle changes we felt necessary, or something that could speed up the marble sorting process. 

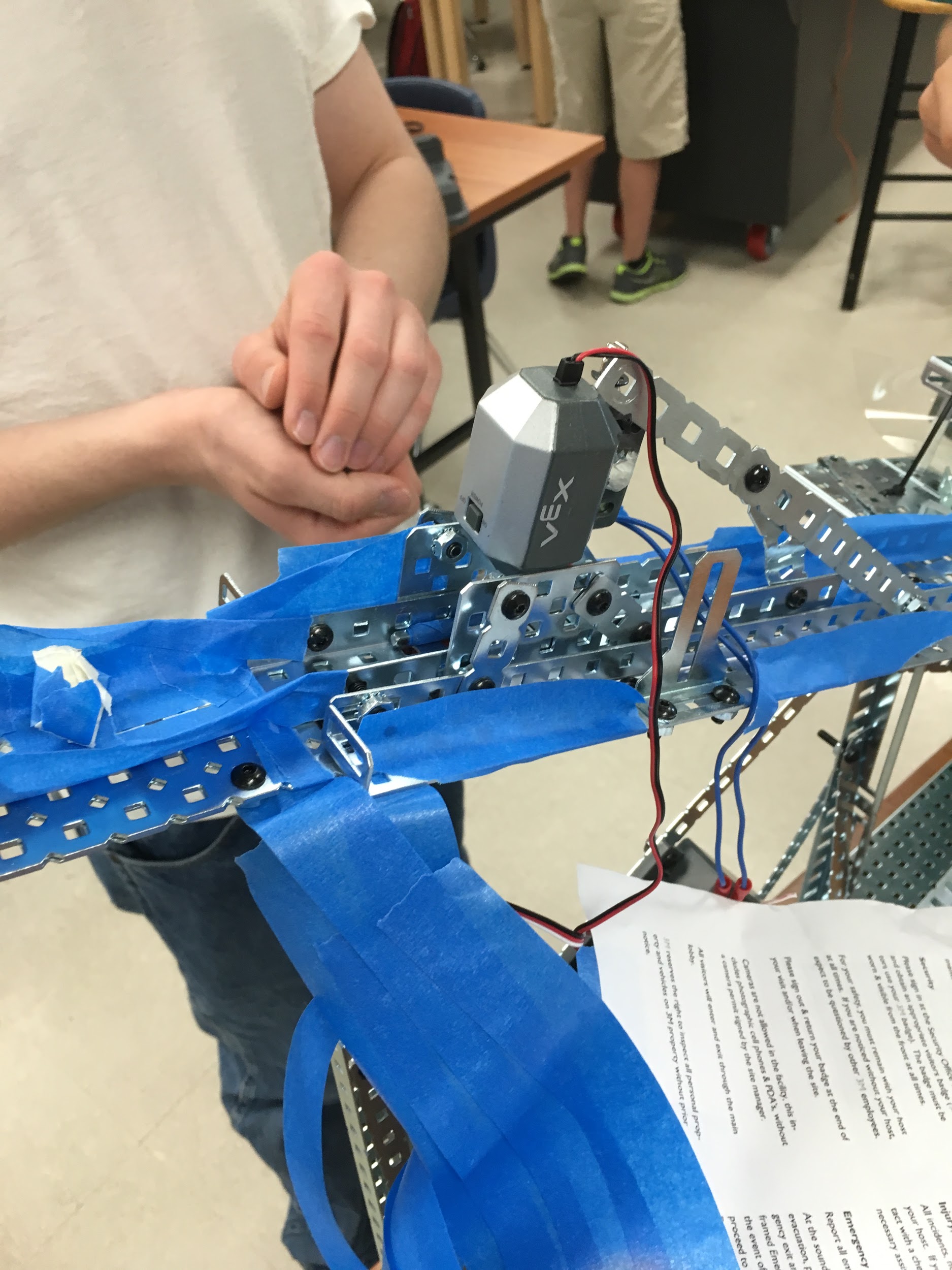
The balls get fed through a funnel, then are regulated onto the track by a sprocket. The steel marbles are diverted onto a separate track by the electromagnet, and the rest get stopped by a second sprocket. While they are stopped, a light sensor gauges the reflectivity. Once it establishes a variable, the marble moves to the second part, where it’s momentum is determined by an optical encoder with a timer on it. Through these two variables, the program determines which of the three trapdoors to open, or to open none for the fourth marble.

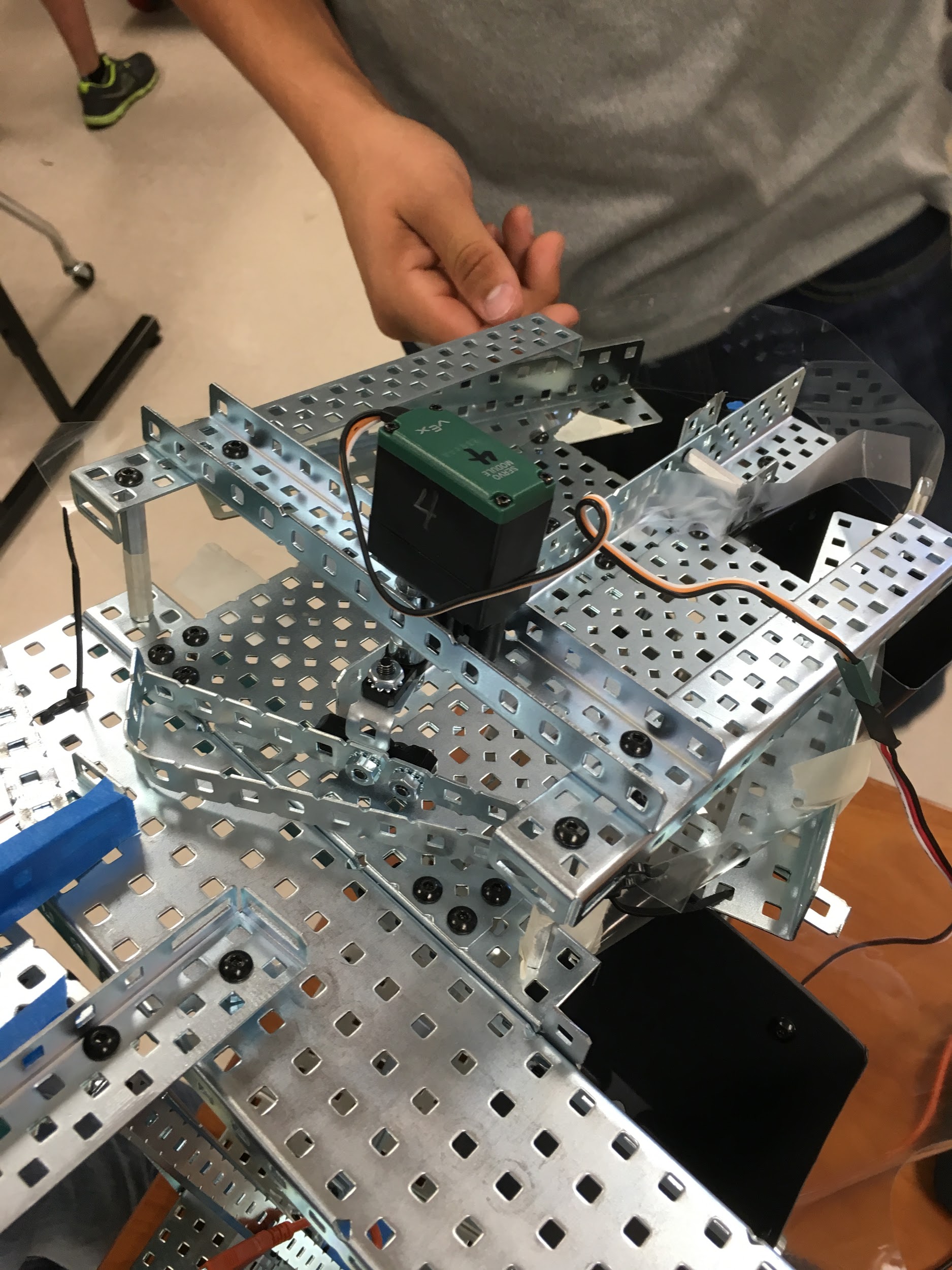
Design Modifications

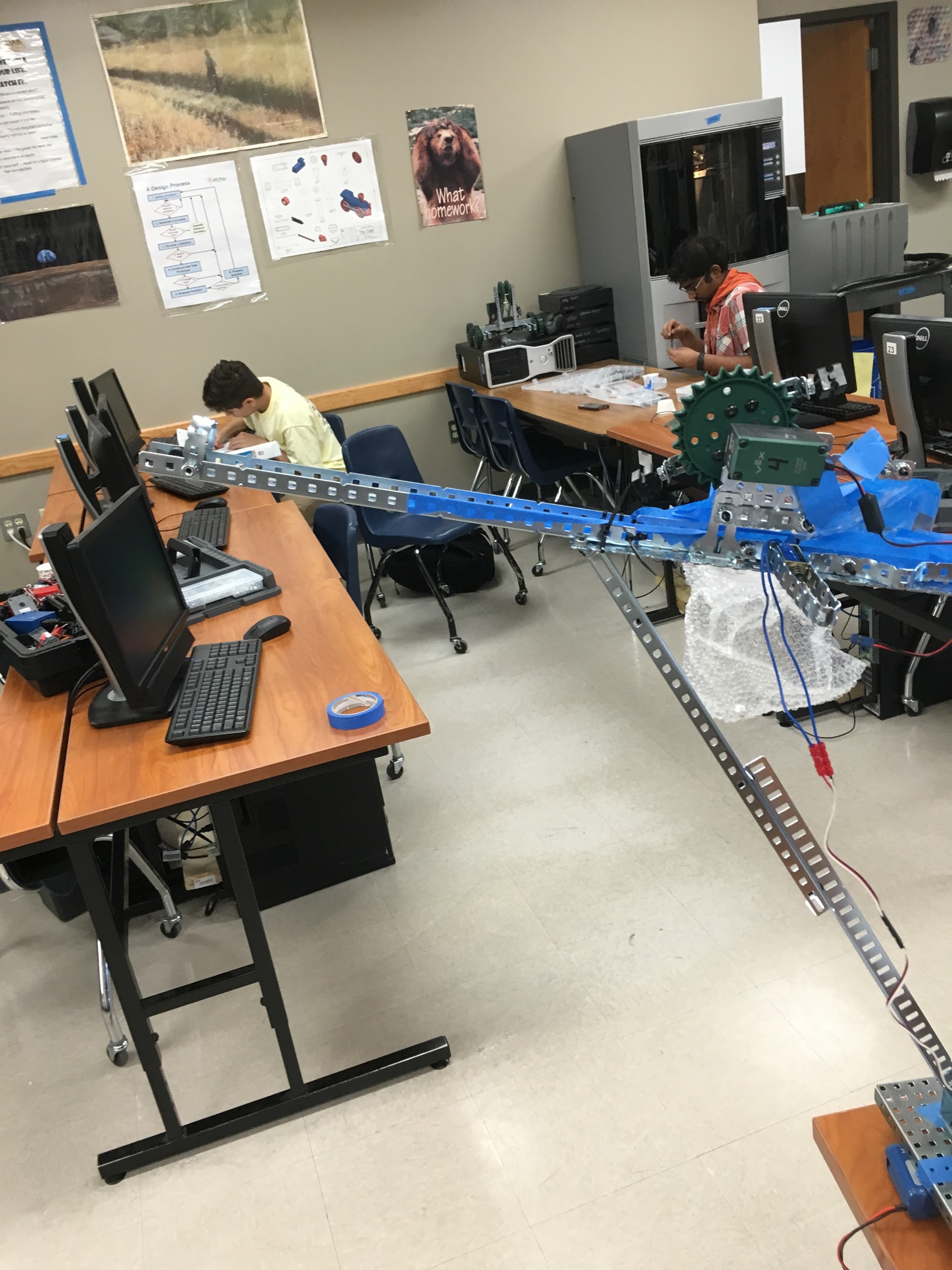
This project underwent a lot of modifications. In the beginning, we were aiming for sorting all five balls, but were only able to efficiently produce methods for four. Also, we originally had attempted to use trap doors, and decided that that wouldn’t work, since we didn’t have an efficient way to construct the mechanisms. We then switched to a gate system using a servo motor, which sorts the marbles into two channels, or remains closed and the marble rolls into a third. Third, the electromagnet is nowhere near as powerful as we had anticipated, so we turned to using a magnet attached to a sprocket to sort out the steel marble. Fourth, our original design had the bins attached to the base, but we didn’t really want to use any more parts, so they are just attached to the bottom of the track. Fifth, we scrapped the optical encoder idea, as there were too many factors at play to accurately determine which marble it was. Sixth, we had a bare-bones structure in the final design, which was much too flimsy, so we added supports to fix that and extended the base. Seventh, we replaced the sprocket on the outside of the light sensor with a solenoid, and the other sprocket on the outside of the funnel too. Eighth, and finally, we didn’t plan for the cortex’s location, so we just attached it to the cross-braces.

Last minute, we attached the new and improved electromagnet, and we coated the entire thing in blue tape to hopefully guide the marbles better.

Final Design

This is the main sensor for the entire design, this is what determines which of the three bins each marble should go into. It turned out to work some of the time, we just got the timing messed up in the program.

This is the final sorting mechanism we used. I’m pretty proud of it, it worked exactly as it was supposed to. The only problems we had with it were code based.

This is the entrance to the machine and the electromagnet sorting system we had to remove the steel balls. This is the part of the machine that we had the most trouble with, as we had to design it in the shortest amount of time and we never got a chance to test it. It had several problems, such as spinning the wrong way in the code (which was easily fixed by just reversing the wires) and it pushed marbles over the blocker we had to regulate which marbles went to the light sensor.

During the official test, a lot of our marbles fell over the edge and went straight to the floor, and most of the remaining clumped up at the light sensor, so more than one went in a cup when only one was supposed to. I don’t have the exact numbers, but I do know that any balls that got into the right container were either accidental or accompanied by several wrong ones.

Code

Here is the code used on our final design:

#pragma config(Sensor, in1, LightSensor, sensorReflection)  
#pragma config(Sensor, in2, LineFollower, sensorLineFollower)  
#pragma config(Motor, port2, Flashlight, tmotorVexFlashlight, openLoop, reversed)  
#pragma config(Motor, port3, Sprocket0, tmotorVex269\_MC29, openLoop)  
#pragma config(Motor, port4, Servo0, tmotorServoStandard, openLoop)  
#pragma config(Motor, port5, Servo1, tmotorServoStandard, openLoop)  
#pragma config(Motor, port6, Sprocket1, tmotorVex269\_MC29, openLoop)  
#pragma config(Motor, port7, ElectroMagnet, tmotorVex393\_HBridge, openLoop)  
//\*!!Code automatically generated by 'ROBOTC' configuration wizard!!\*//  
  
int sprockDelay = 250;  
int whitePlasticThreshold = 225;  
int aluminumWoodThreshold = 500;  
int woodThreshold = 1500;  
int type; //Clear plastic: 0, white plastic: 1, wood or aluminum: 2  
int clearAngle = 60;  
int whiteAngle = -60;  
int aluminumAngle = 0;  
bool wood;  
  
void sprockPrep(){ //Closes solenoid gates.  
 motor[port3] = 127;  
 motor[port6] = 127;  
}  
  
void magnet(){ //Starts steel separater.  
 motor[port7] = -31;  
}  
  
void sprock(){ //Lets one marbel into sorter.  
 motor[port3] = 0;  
 delay(sprockDelay);  
 motor[port3] = 127;  
 delay(2500);  
}  
  
void findType(){ //Determines marbel type using lightSensor values.  
 int x;  
 motor[port2] = 127;  
 delay(200);  
 x = SensorValue[in1];  
 for(int f = 0; f <= 2000; f++){  
 if(SensorValue[in1] > x){  
 x = SensorValue[in1];  
 }  
 delay(1);  
 }  
 motor[port2] = 0;  
 if(x > whitePlasticThreshold){  
 if(x > aluminumWoodThreshold){  
 type = 2;  
 }else{  
 type = 1;  
 }  
 }else{  
 type = 0;  
 }  
}  
  
void sortA(int t){ //Sorts the marbel into the appropriate bin.  
 if(type == 0){  
 motor[port4] = clearAngle;  
 }  
 if(type == 1){  
 motor[port4] = whiteAngle;  
 }  
 if(type == 2){  
 motor[port4] = aluminumAngle;  
 }  
 motor[port6] = 0;  
 delay(1000);  
 motor[port6] = 127;  
}  
  
void findWood(){  
}  
  
void sortB(bool w){  
}  
  
task main(){  
 magnet();  
 sprockPrep();  
 delay(5000);  
 while(true){  
 sprock();  
 delay(500);  
 findType();  
 sortA(type);  
 if(type == 2){  
 findWood();  
 sortB(wood);  
 }  
 }  
}

This code mostly worked, after our presentation we realized that there were a few timing errors, but those were understandable, since we never had time to test the code before our presentation.

Reflection

a. What would your team do differently with your design solution and why?

If we could have more time to work on it, we would fix the electromagnet, the ball regulators, and the code, since all tended to not work as well as we had expected.

b. What was the most challenging aspect of this design problem?

Figuring out how we would sort the marbles was tricky, because many of the marbles had similar variables, and the system we ended up using was a little finicky.

c. What did you learn?

I learned about how we should try to have a good design first, since we had to go back and change it, which took far more time than we could afford.

d. What were some of the challenge**s** of working in a design team?

When one person of the group was absent, which was often, the entire group got a little more behind. Also, it became harder to work in parallel, since we all required the same screwdriver at some point.

Design Process

1. Define problem

For this project, we were given a design brief that clearly defined the problem for us.

1. Generate concepts

Individually, we each thought of an idea to solve the problem.

1. Develop a solution

Then, we met back together and elected one idea and refined it to build.

1. Construct and test prototype

In this project, we had no time to prototype or test, so our construction work was only on the final design.

1. Evaluate solution

Well, we evaluated our solution during the presentation, and we decided that it wouldn’t work.

1. Present solution

As far as successful presentations go, this was not one of them. Ultimately, our design failed, but we did learn what to improve if we were to continue this project.