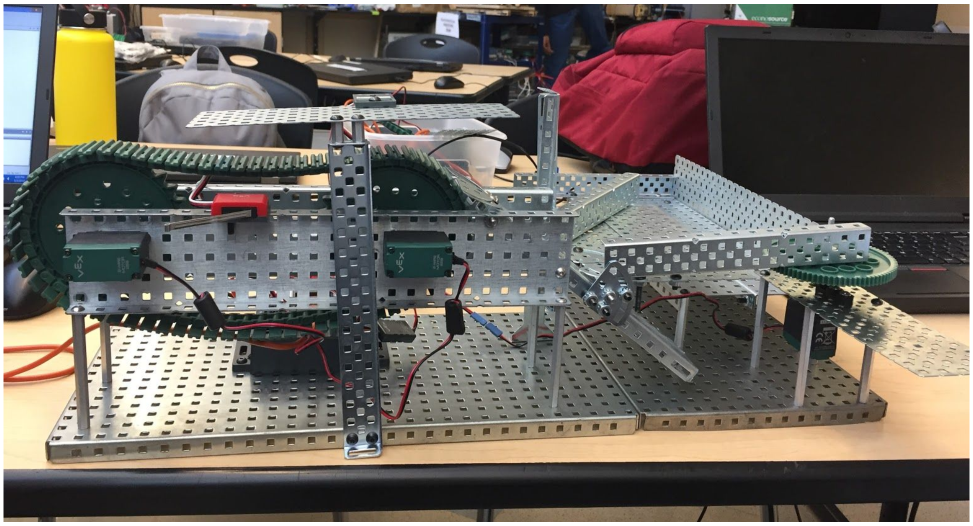
Project 3.2.4 Machine Control Design

MENT to be

By: Megan Fannin, Nitin Indukuri, Emily Que, & Tyler Nguyen

4/23/18

POE 6



Tables of Contents

Design Brief

Client: So Sweet Cookie Factory

Target Consumer: Cookie Factories

Designers: MENT to be (Megan, Nitin, Tyler, Emily)

Problem Statement: The So Sweet Cookie Factory doesn’t have a device that is able to detect a certain number of cookies then transfer it to further stages in the packaging process.

Design Statement: The “MENT” to be company must design an efficient and effective device to help in the factory’s cookie packaging process.

Constraints:

- must first transport cookies to secondary area
- once 3 cookies detected, must be sent to a final location to be packaged

Team Deliverables:

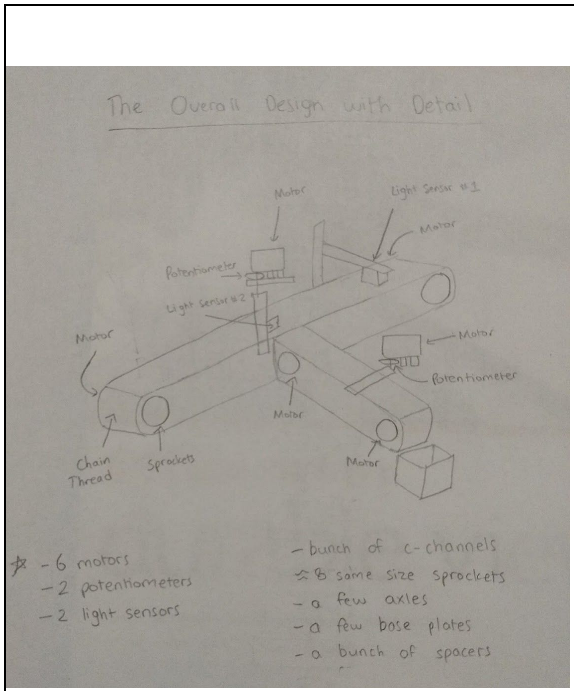
- Documentation
    - Design Brief
    - Brainstorm
    - Decision Matrix Explanation
    - Design Modifications
    - Final Design
    - Key Contributors
- Working Prototype
- Robot C Program
    - Fully commented
    - Pseudocode filled out
    - Task description

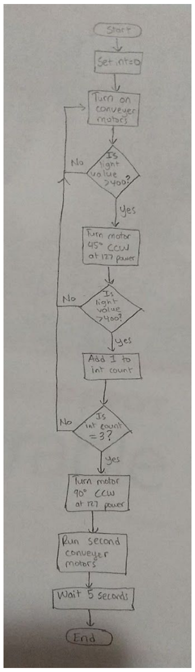
Individual Deliverables:

- Brainstorm Sketches & Flowchart
- Individual Project Log
- Conclusion Questions

Brainstorming Sketches

<table>
<tr>
<td>

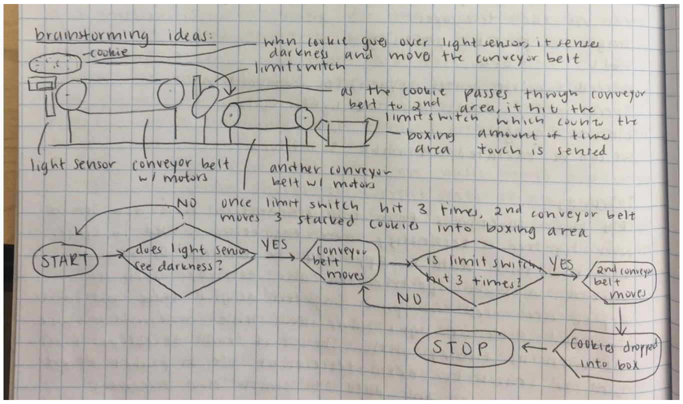




</td>
<td>

For this design, the cookie will first be dropped onto the conveyor belt and once the light sensor detects that there is a change in light, the conveyor belt will start moving. Once the cookie reaches the second light sensor and it has been detected it will add one count and the conveyor belt will stop moving while a potentiometer turns a lever redirecting the cookie onto a second conveyor belt. The conveyor belt will then move the cookie down to a second lever. Once the second light sensor has detected 3 cookies, the first conveyor belt will stop moving and the lever will release the cookies into the package.
X Nitin Indukuri 5/3/18

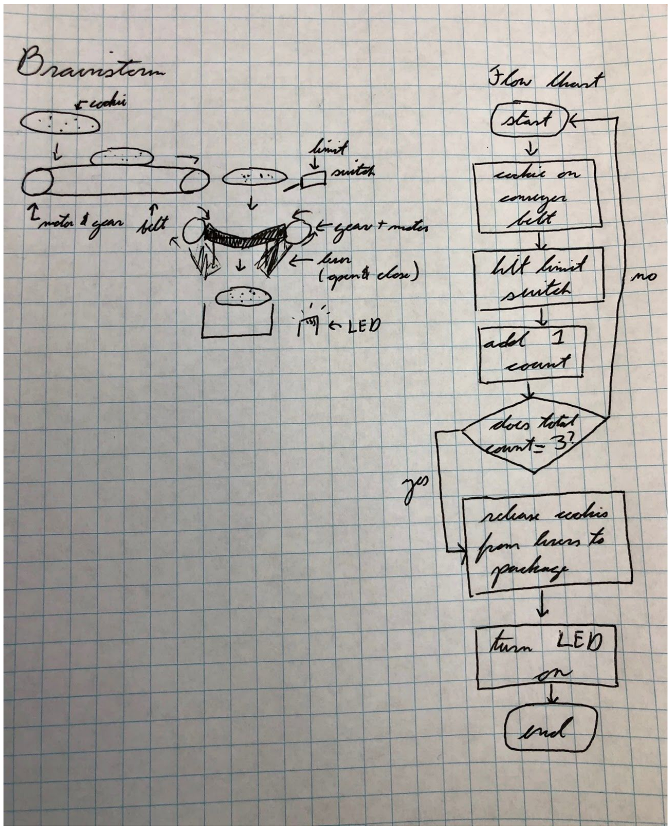
</td>
</tr>
<tr>
<td>


brainstorming ideas:
-cookie
when cookie goes over light sensor, it sense darkness and move the conveyor belt
limitswitch
as the cookie passes through conveyor belt to 2nd area, it hit the limit switch which count the amount of time touch is sensed
boxing area
light sensor
conveyor belt w/ motors
another conveyor belt w/ motors
once limit switch hit 3 times, 2nd conveyor belt moves 3 stacked cookies into boxing area
NO
START
does light sensor see darkness?
YES
conveyor belt moves
is limit switch hit 3 times?
YES
NO
2nd conveyor belt moves
cookies dropped into box
STOP

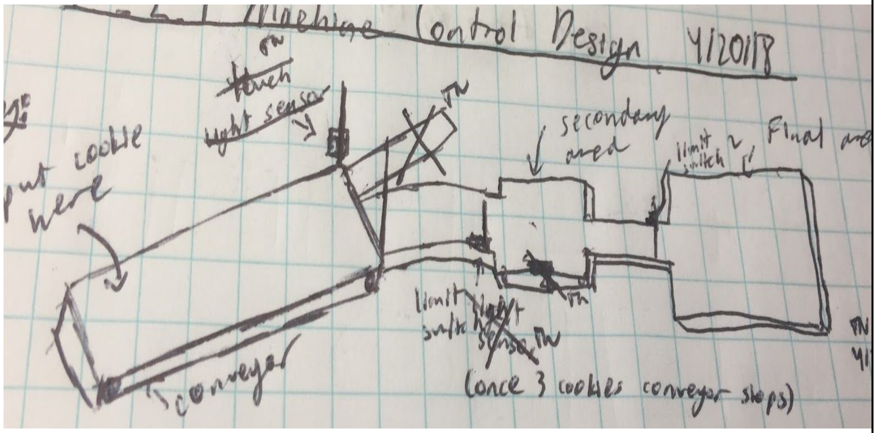
</td>
<td>

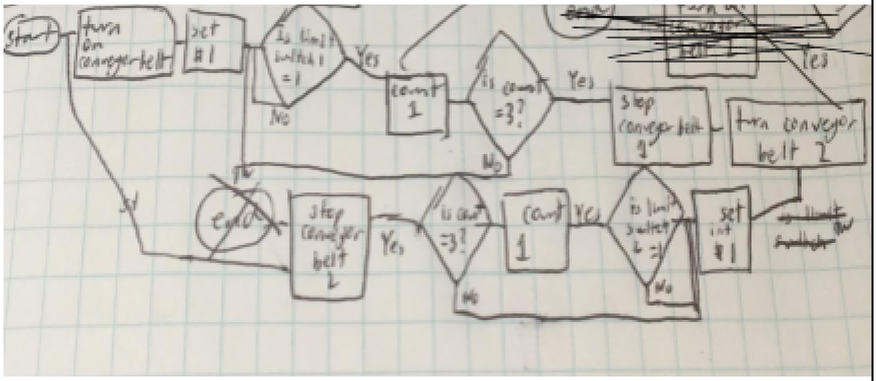
The cookie will first be passed over a light sensor which will turn on the conveyor, the conveyor belt will then move the cookie onto a second conveyor belt that is lower than the previous. As it drops to the lower belt, it will hit a limit switch that sets up a count to detect when 3 cookies have passed it. Once 3 cookies have moved across the system, the 1st conveyor belt will stop moving and the second belt will begin to move. The 3 cookies will then be dropped into a final box for packaging.
X Megan Fannin 5/1/18

</td>
</tr>
</table>



The cookie will first drop onto the belt which is controlled by two motors to move at a certain speed. As the gears turn, the cookie will be transported and fall onto two levers. As the cookie drops, it will hit a limit switch which adds one count to the overall count. When the count has reached three, the levers will open and release the cookies into their package. As the cookies are released, an LED will turn on which signifies that the cookies are ready to be shipped.
X Emily Que 5/3/18





The first thing you have to do is to put the cookie on the conveyor belt. The conveyor belt will be constantly on so when you place it onto the conveyor belt, the cookie will be moved toward forward until it hits the limit switch. When the limit switch is activated three times, the conveyor belt will stop so only three cookies will be on the secondary area. Then the conveyor belt on the secondary area will start moving so it will move the remaining three cookies on the secondary area will be moved toward the final area.
X Tyler Nguyen 5/3/18

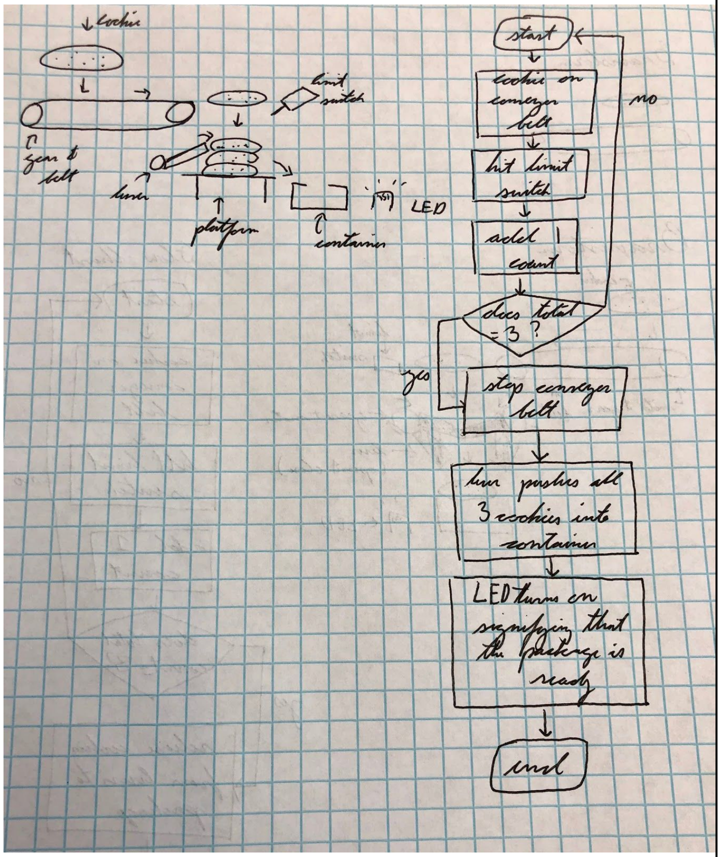
Decision Matrix

Points: 1-5	Functionality	Efficiency	Aesthetics	Simplicity	Total Points
Tyler	5	4	3	3	15
Nitin	5	3	5	2	15
Megan	5	4	3	5	17
Emily	5	5	4	5	19

X Eugene Chou 04/25/2018

We decided to rate our design matrix on a scale from 1-5 with 5 being the highest score. The categories we chose were functionality, efficiency, aesthetics, and simplicity. Functionality depends upon if the system can work properly for the intended purpose. The lowest score at 1 would represent a design that doesn't work at all as a cookie factory and a 5 being a design that perfectly moves 3 cookies into multiple areas to be packaged. Efficiency measures how well the prototype can work with 1 being the lowest score while 5 is the highest. In this category, we looked for if the prototype would be able to work properly without any troubles and setbacks with the most efficient systems/parts. Aesthetics used the same grading category but was looking for how appealing the design was. We wanted a design that would be clear in every aspect of its functions and altogether looked nice. The last category we created was simplicity which was also rated on a scale from 1-5. Simplicity measured how simple and straightforward a design was. Having a clear layout for the design would make the coding, building, and understanding process much more easy. At the end of rating, Emily's design won because it fulfilled each category. Her design had the required components and was laid out and used materials that would make it very efficient and simple to create.

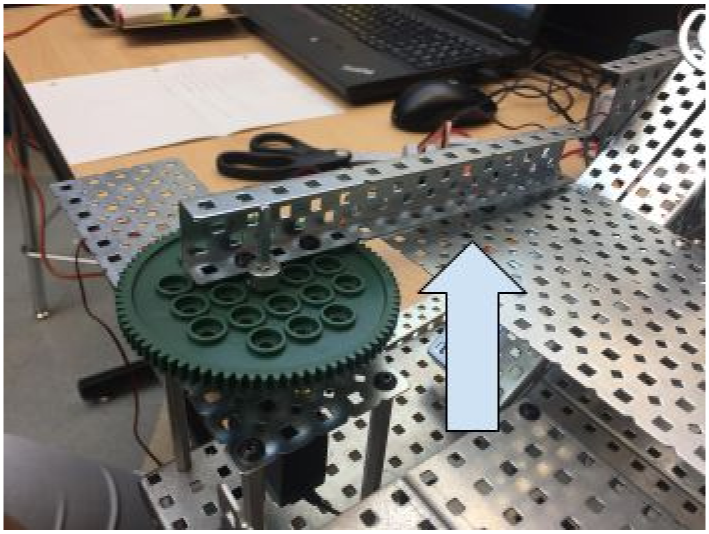
Design Modifications



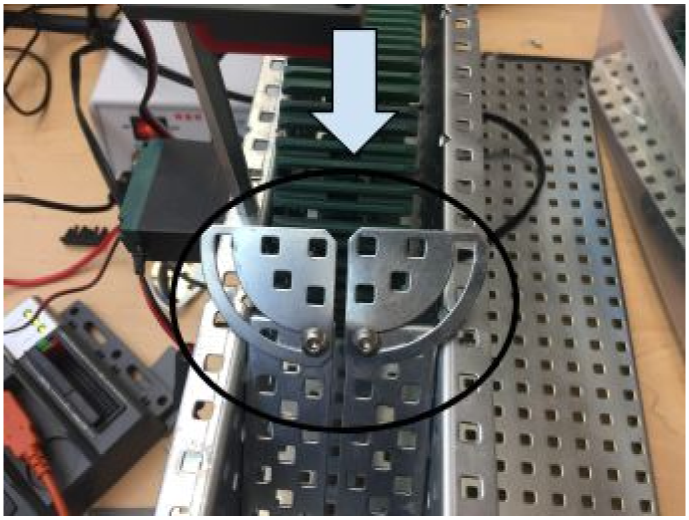
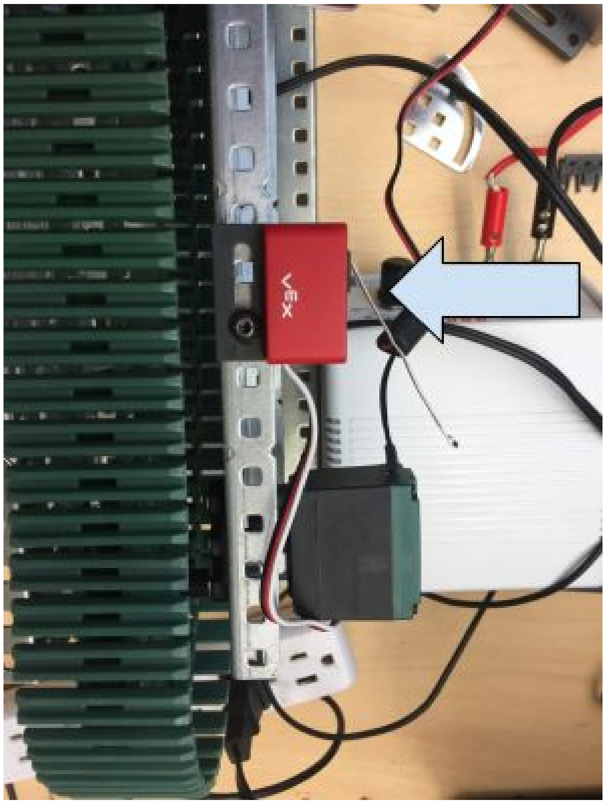
5/1
Another modification that we did was to stop the conveyor belt once the limit switch hit 3. This was to ensure that no more cookies would compile onto the platform before being pushed into the package. Before this, the conveyor belt would continue moving and functioning, possibly messing up the entire system with an excess of cookies. With this modification, the machine is now ensured to have 3 cookies in every single package, no more, no less.

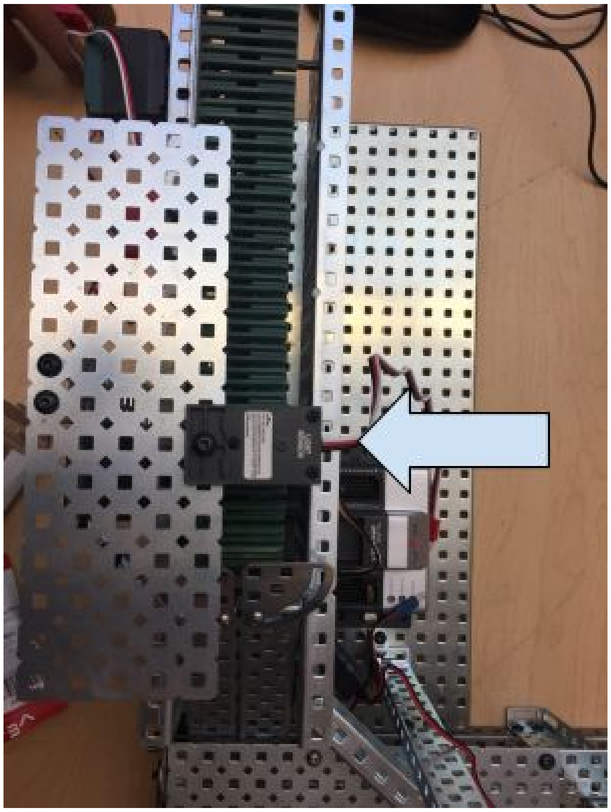
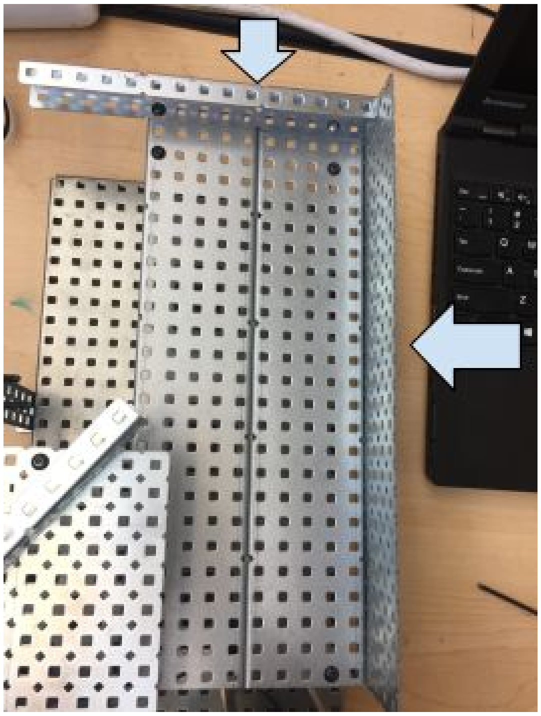
Another modification that we did was to stop the conveyor belt once the limit switch hit 3. This was to ensure that no more cookies would compile onto the platform before being pushed into the package. Before this, the conveyor belt would continue moving and functioning, possibly messing up the entire system with an excess of cookies. With this modification, the machine is now ensured to have 3 cookies in every single package, no more, no less.

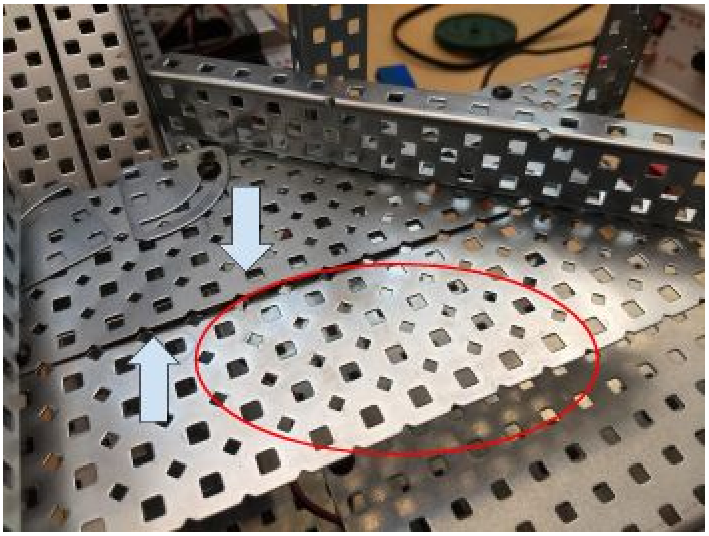
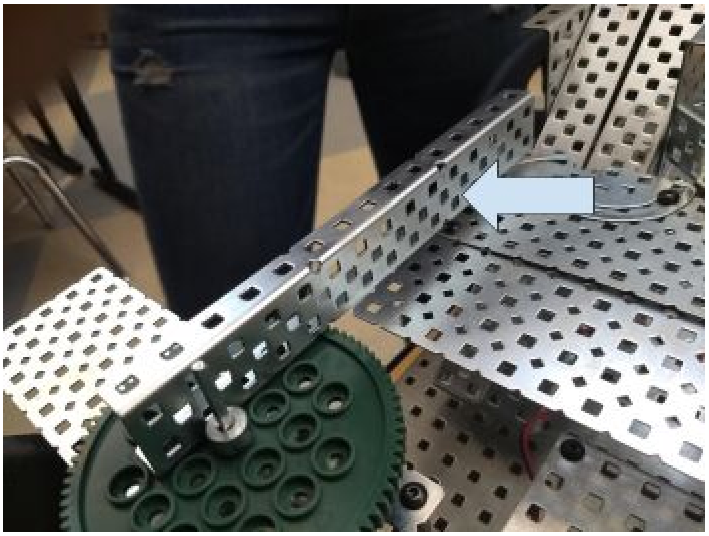
X Emily Que 5/1/18



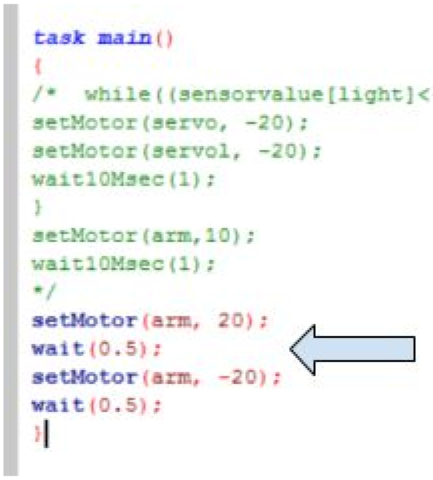
5/1
The first modification done was to change the flaps to a lever that would push the cookies off a platform after three have accumulated together. This lever would then push the cookies into the desired package or container, making it ready to ship off.
This modification was done as the group was unable to attain the proper materials to create the flaps that the original design required. This design is also much more simplistic and requires less parts and energy to create the same motion.
X Emily Que 5/1/18

	5/4 This modification we added angle gussets to ensure that the cookies would not be stuffed between the c-channel and the conveyor belt. This modification would allow the cookie to move down the slope without getting caught anywhere. We also did this at the bottom of the slope to also make sure that the cookie wouldn't be stuck there either. This allowed the machine to push the cookies down toward the secondary area a hundred percent of the time. X Tyler Nguyen 5/4/18
	5/3 This was another modification we had made. This limit switch was added to keep the program running forever and to keep restarting the process by pressing it. It helps us start the process until the 3 cookies are in the secondary area, then the program stops. This allows us control over how many batches of 3 cookies went through our system, because it resets every time we press it. X Nitin Indukuri 5/3/18

	5/3 At first we had a limit switch that would be pushed to detect the cookies. However it was not a very efficient way to detect the cookies, so we opted to use a light sensor instead. In the beginning we put the light sensor on the already existing c-channel to detect the cookies. However there wouldn't be enough space for a cookie to pass underneath it. To fix this we had to use a c-channel, standoffs, and a steel plate to extend the light sensor over the conveyor belt perfectly. In this position as shown in the picture, every single cookie that had gone by the sensor had been sensed unless it was the very edge. This would then allow the machine to detect how many cookies has passed the sensor. When three cookies pass, this would tell the machine to stop the conveyor belt and move the cookies to the final packaging area. X Tyler Nguyen 5/4/18
	5/3 This is another modification we had made while finding some errors in our design. We have added a c-channel and a steel plate at the ends of our secondary area to prevent the cookies from flying out. Since our cookie machine was pretty "aggressive," we had to add some metal to keep the cookies in the secondary area for later shipping. This was a pretty easy modification and we had originally made the steel plate much taller, but our arm kept hitting it, so we made the plate shorter. These two parts were enough to keep all the cookies in. X Nitin Indukuri 5/3/18

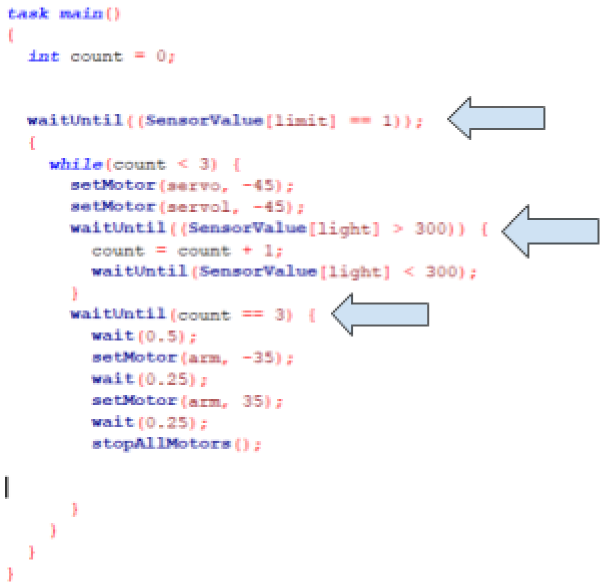
	5/4 This was a very last minute modification because our cookies were being stuck more consistently in front of the steel plates. Previously, the plate inside the red circle to the right, was above the other plate and the cookies kept hitting that small edge, and preventing the arm to move all the way around. By switching the plate under the other plate, we were able to have a smooth flow and the cookies went smoothly without obstructing anything. This eventually allowed all the 3 cookies to easily be pushed into the secondary area. X Nitin Indukuri 5/4/18
	5/4 The c-channel in the picture was used to push the cookies into the final packaging area. Originally it was flipped around so the edges would be facing the inside of the machine. This would make it harder for it to push the cookies because sometimes they would get caught on the small edge of the c-channel. The cookies would then stay on there and not be pushed down to the final area. In response, we flipped the c-channel so there would be nothing for the cookies to get stuck on. X Tyler Nguyen 5/4/18

Program Modifications



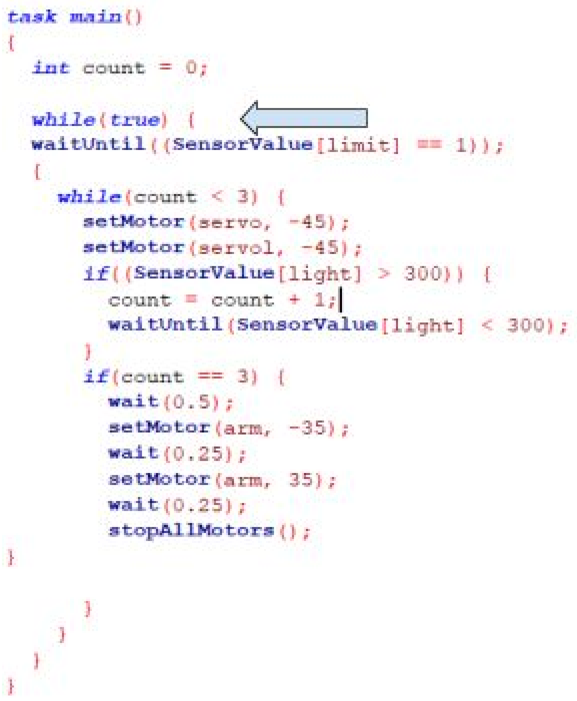
```
task main()
{
/*  while((sensorvalue[light]< 190)) {
setMotor(servo, -20);
setMotor(servol, -20);
wait10Msec(1);
}
setMotor(arm,10);
wait10Msec(1);
*/
setMotor(arm, 20);
wait(0.5);
setMotor(arm, -20);
wait(0.5);
}
```

5/1
This was the first version of the code that we began with. This code was an attempt to test the movement of the lever we created that would push the cookies to the final packaging area. The above code was also an attempt to move the conveyor belt under a normal amount of light. The problem that arose from this code was that the lever arm would move too much in one direction and it was constantly malfunctioning because there was no stop command. Also at this point there were no comments and no code to control the light sensor for when cookies would be detected so the code was nowhere near completion.
X Megan Fannin 5/1/18



```
task main()
{
  int count = 0;


  waitUntil((SensorValue[limit] == 1));
  {
    while(count < 3) {
      setMotor(servo, -45);
      setMotor(servol, -45);
      waitUntil((SensorValue[light] > 300)) {
        count = count + 1;
        waitUntil(SensorValue[light] < 300);
      }
      waitUntil(count == 3) {
        wait(0.5);
        setMotor(arm, -35);
        wait(0.25);
        setMotor(arm, 35);
        wait(0.25);
        stopAllMotors();


      }
    }
  }
}
```

5/3
This was the second version of the code where we got stuck and there were a few errors that we had to fix. The problem with this code was that we kept using "waitUntil" for every command using the counting integer. We realized that "waitUntil" is only used to stop the code until something happens. Once whatever you want happens, then it runs through the code after it. "waitUntil" wasn't the ideal solution in our code because it just stopped our program at every "waitUntil" and won't move on in the code. We had fixed it later by changing a few of them to "if" statements which allow the code in them to run only if whatever we want happens.
X Nitin Indukuri 5/3/18



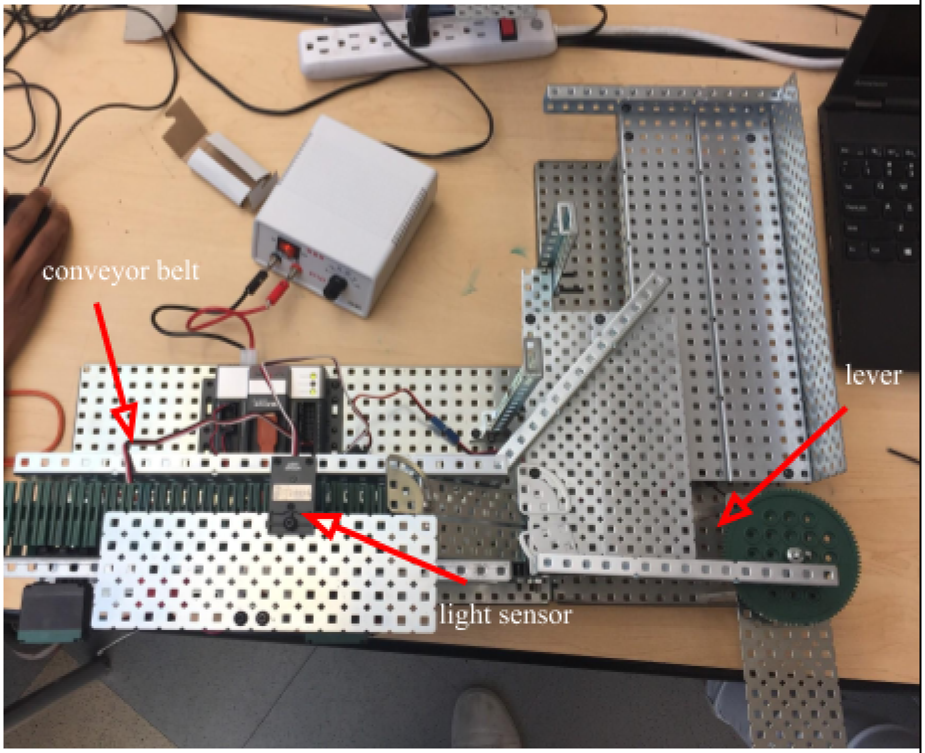
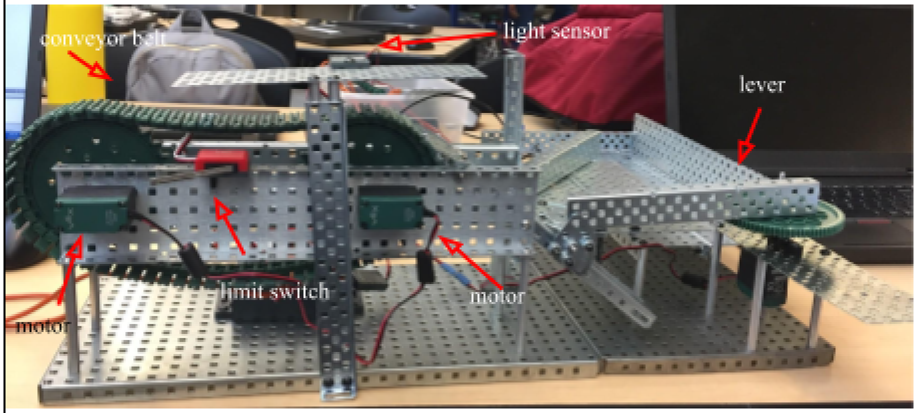
```
task main()
{
  int count = 0;

  while(true) {
  waitUntil((SensorValue[limit] == 1));
  {
    while(count < 3) {
      setMotor(servo, -45);
      setMotor(servol, -45);
      if((SensorValue[light] > 300)) {
        count = count + 1;
        waitUntil(SensorValue[light] < 300);
      }
      if(count == 3) {
        wait(0.5);
        setMotor(arm, -35);
        wait(0.25);
        setMotor(arm, 35);
        wait(0.25);
        stopAllMotors();
}


      }
    }
  }
}
```

5/3

This was the last step in our programming that we had struggled a lot and had wasted a lot of time on. The problem was that once the integer variable had reached 3, it kept repeating whatever code was in that "if" statement. This made our arm go crazy and just keep rotating in circles without stopping. We had realized that the only problem with our code was that the while(true) statement kept repeating everything after it. We never had reset our integer count after the first 3 cookies went through the conveyor, so our light sensor was counting infinite integers, and our arm never stopped rotating. This was the easiest fix to make because all we had to do was to move our while(true) statement to before setting up our integer variable. This way, whenever the code in our last "if" statement was completed, it went back to the top and reset the count integer back to 0, which allows another 3 cookies to go through our system to the secondary area.

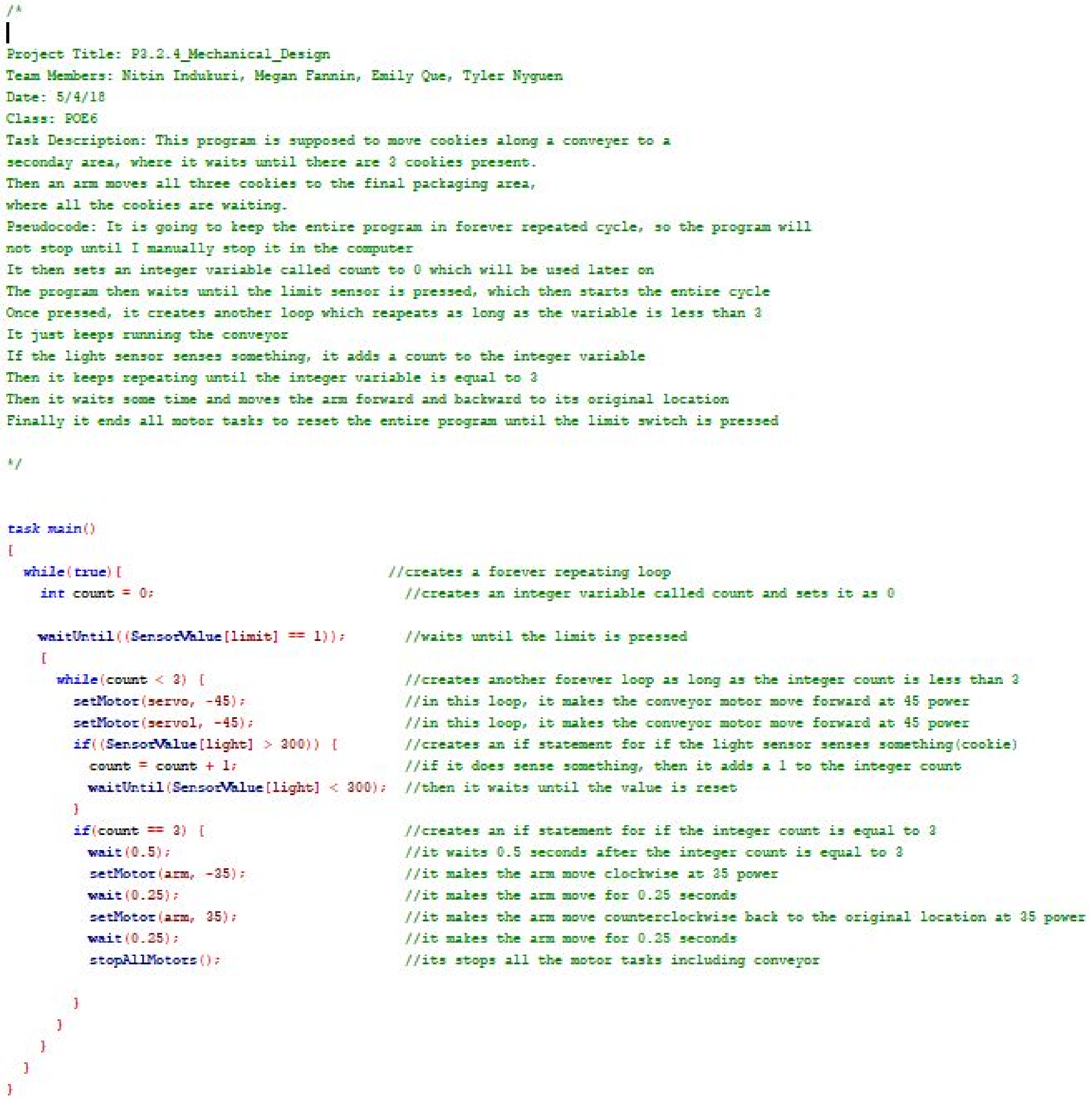
X Nitin Indukuri 5/3/18

Final Design Solution

	This is the top view of our cookie factory prototype. From above, you can see all components of the design, including the conveyor belt, light sensor, ramp, lever and cortex. You can additionally see the multiple platforms we built to act as different loading areas. The cookie would begin on the conveyor belt which is activated by the limit switch on the side of the mechanism. The conveyor belt then moves the cookie down the ramp and onto the lower platform. Once 3 cookies have been sensed by the light sensor (which detects changes in darkness), the "lever door" controlled by the 3rd motor swings 90 degrees to push the cookies on the platform onto another lower platform that acts as the final packaging area.
 conveyor belt light sensor lever limit switch motor motor	This is the front view of our final prototype. This angle better shows how each component that creates the system is connected together. The motors that move the conveyor belt, the limit switch, and the light sensor are all held up by 2 horizontal c-channels lifted by standoffs. This is then connected to a lower platform with seat channels angeled like ramp/inclined plane. The platform is held up by another c-channels which also has a diagonal c-channel on the surface which acts like a boundary for the cookies. Creating an extra wall makes sure the cookies move in a direct path to the lowest platform. The lower platform is also made of

	flat metal sheets that have elevated walls to act as boundaries. Lastly in this picture, you can see that the lever made out of a c-channel connected to a gear moved by a motor. As the motor moves the gear, it powers the arm to push the cookies down.

Final Program



```
/*

Project Title: P3.2.4_Mechanical_Design
Team Members: Nitin Indukuri, Megan Fannin, Emily Que, Tyler Nyguen
Date: 5/4/18
Class: POE6
Task Description: This program is supposed to move cookies along a conveyer to a
seconday area, where it waits until there are 3 cookies present.
Then an arm moves all three cookies to the final packaging area,
where all the cookies are waiting.
Pseudocode: It is going to keep the entire program in forever repeated cycle, so the program will
not stop until I manually stop it in the computer
It then sets an integer variable called count to 0 which will be used later on
The program then waits until the limit sensor is pressed, which then starts the entire cycle
Once pressed, it creates another loop which reapeats as long as the variable is less than 3
It just keeps running the conveyor
If the light sensor senses something, it adds a count to the integer variable
Then it keeps repeating until the integer variable is equal to 3
Then it waits some time and moves the arm forward and backward to its original location
Finally it ends all motor tasks to reset the entire program until the limit switch is pressed

*/


task main()
{
  while(true){                                //creates a forever repeating loop
    int count = 0;                              //creates an integer variable called count and sets it as 0

    waitUntil((SensorValue[limit] == 1));       //waits until the limit is pressed
    {
      while(count < 3) {                        //creates another forever loop as long as the integer count is less than 3
        setMotor(servo, -45);                   //in this loop, it makes the conveyor motor move forward at 45 power
        setMotor(servo1, -45);                  //in this loop, it makes the conveyor motor move forward at 45 power
        if((SensorValue[light] > 300)) {        //creates an if statement for if the light sensor senses something(cookie)
          count = count + 1;                    //if it does sense something, then it adds a 1 to the integer count
          waitUntil(SensorValue[light] < 300);  //then it waits until the value is reset
        }
        if(count == 3) {                        //creates an if statement for if the integer count is equal to 3
          wait(0.5);                            //it waits 0.5 seconds after the integer count is equal to 3
          setMotor(arm, -35);                   //it makes the arm move clockwise at 35 power
          wait(0.25);                           //it makes the arm move for 0.25 seconds
          setMotor(arm, 35);                    //it makes the arm move counterclockwise back to the original location at 35 power
          wait(0.25);                           //it makes the arm move for 0.25 seconds
          stopAllMotors();                      //its stops all the motor tasks including conveyor

        }
      }
    }
  }
}
```

Key Contributors

Nitin Indukuri

During this project, I have contributedty a lot to the building and programming of our entire cookie factory. Although I was at the Vex Robotics Worlds Competition for half our work-time and on a field trip for another day, I still managed to help contribute to the documentation when I was gone. Once I came back, I was able to help my team finish building the remaining half of our cookie factory. I finished building the arm and ramp to the area with the arm. Next I had started working on the code to make sure our cookie factory will actually work. Programming the cookie factory was my major contribution to the team because I had helped fix and perfect it before we had to present in class. In the code, I had failed many different times, but I had thought through everything and had eventually perfected it and made sure it worked. Some unfortunate failures I had with my code, was when I typed “sensorvalue” instead of “SensorValue.” Many of these types of mistakes had prolonged the amount of time it was supposed to take to program the cookie factory. However in the end, I finished and everything worked. Finally I did all the documentation for all the parts that I have contributed to the team such as the explanation of my code, and all programming modifications I had made throughout the time. I had also made a few last minute modifications which I had updated into our documentation, by adding pictures and labels to clearly describe what I had changed. Overall, I had contributed a lot of my effort to everything in this project, but my major contribution was the cookie factory code.

Tyler Nguyen

My main contribution to this project was helping build the machine and improving it's design. I put together the main conveyor belt and made sure that it was stable. I found ways to connect the motors and incorporate them into the design. I would improve how the machine worked, such as adding the borders on the finals areas to make sure that no cookies would fall off. Another modification I changed was moving the line sensor so that it would be able to detect the cookies. I would build the base of the machine and made sure that all of our parts were stable and not loose. When not building the machine, I would make sure that the right materials are being used and that we have them. I also advised if a modification would work or not and see if we could have expanded on that idea. Another thing that I did was help on the documentation. I made sure that the pictures and their descriptions were properly formatted and uniform. I also wrote several sections of the design modifications section. I also inputted my own brainstorming picture and flow chart and wrote my own description of the idea. I also helped with some ideas for the decision matrix and the final design solution sections. For the whole documentation, I made sure that everything was on the same page and recorded that into the table of contents.

Megan Fannin

In this project, I mainly worked on the documentation and building of the actual prototype. The first day we started building, I helped choose the parts that we were going to use in the actual design. I also helped lay out the design plus build it. My first responsibility in building was to make sure we could create a layout that would use a limit switch, 3 motors, and 2 additional areas for the cookies to fall onto. During this process we ran into many problems in which I had to help create solutions. For example the cookies kept flying in different directions so I came up with the idea to add a diagonal c-channel to regulate the direction the cookies traveled. In terms of documentation, I specifically wrote the paragraph for my brainstorming sketch, the paragraph describing the design matrix, the description of the final design, and paragraphs describing modifications made. In the beginning of the project I also filled out the google responsibilities form as the project manager which also made me responsible to submit this document. Overall me and my team completed an equal amount of work to create our prototype.

Emily Que

For this project, I contributed both on documentation and building. For documentation, I inserted and formatted the design brief while making it more specific to our project. Formatting both the design brainstorms and that decision matrix, I was able to begin the decision matrix explanation while contributing the initial design that was chosen. My design had moved passed the initial decision and into the modification process to become our final design. Along the way to creating the final the design, the group had run into a multitude of issues that I had contributed ideas to resolve. One main problem I worked on was that the cookies were getting stuck in certain areas of the machine because of rough transitions between different sections. One modification I had come up with was to implement a faster moving conveyor belt along with adding gussets to create a smoother transition for the cookies. The higher speed belt allowed for the cookies to be pushed into an area that was reachable for the lever arm while the gussets created a slope which made it easier for the cookies to slide down the ramp after being transported by the belt. These implications were vital to the productivity of our final design and allowed for an overall smoother transition between stages in the system. Back in documentation, I had inserted and explained the implemented changes into the design modifications page while annotating/labeling pictures for the final design solution. Along with that, throughout the entire process I had created and continued updating the gantt chart with planned and completed tasks for each day the project had been worked on.