Team Number - 12311
Tech-No-Logic

Name: Aidan, 10th Grade
Role: Programmer
Interests: robotics, tennis
Reasons for Joining: I wanted to develop my programming by writing code for an actual object

Name: Brandon, 10th Grade
Role: Programmer
Interests: Robotics, software developing
Reasons for joining: I have always had the interest in involving myself in mechanical engineering and robotics. At some point I developed a passion for coding and so I took a summer program which taught me Java for 5 weeks. Having the skills for coding and the interest in robotics, joining the FTC was a fantastic thing for me to participate in.

Name: David, 11th Grade
Role: Builder, Designer, Quality Inspector
Interests: Engineering, Design, Science, Robots in general
Reasons For Joining; I wanted to join the robotics team to enrich my own life and learn about things I'm interested in. I had freetime that wasn't being used for anything important. Feels good saying that I know how to build a robot and all the steps included within the process. During this experience I learned a lot about different types of machines including some that can easily break/bend/cut metal, print material, and cut shapes using lasers.

Name: Matias, 10th Grade
Role: Designer and builder
Interests: Robotics, tennis, puzzle boxes/mechanical furniture, biking
Reasons for joining: I want to learn a variety of skills such as working with wood, metal, 3D printing, and CAD programs

Name: Oliver, 9th Grade
Role: Builder
Interests: Machines, 3-D Design
Reasons for joining: I have participated in FTC in the past and wanted to continue doing it.

Name: Sadiq, 11th Grade
Role: Programer and builder
Interests: Thinking, programming and drawing
Reasons for joining: I wanted to challenge my creativity, improve on my problem solving skills and also get some experience working with a team. Additionally, I wanted to learn what it takes for a team to accomplish a goal or task.

Team 12311: Tech-no-logic

Summary Pages:

Original Plan:

- Robot would be able to suck in the block and pick it up with an arm that is located on the back
- The arm would grab the robot on one of the nubs coming out of the top
- That arm would be able to stack the blocks and place the capstone
- The robot will use mecanum wheels to help drivers stack and grab blocks easier
- The base will be made out 8020 extrusion for strength and stamped aluminum for ease of attaching things

Changing Up Original Plan:

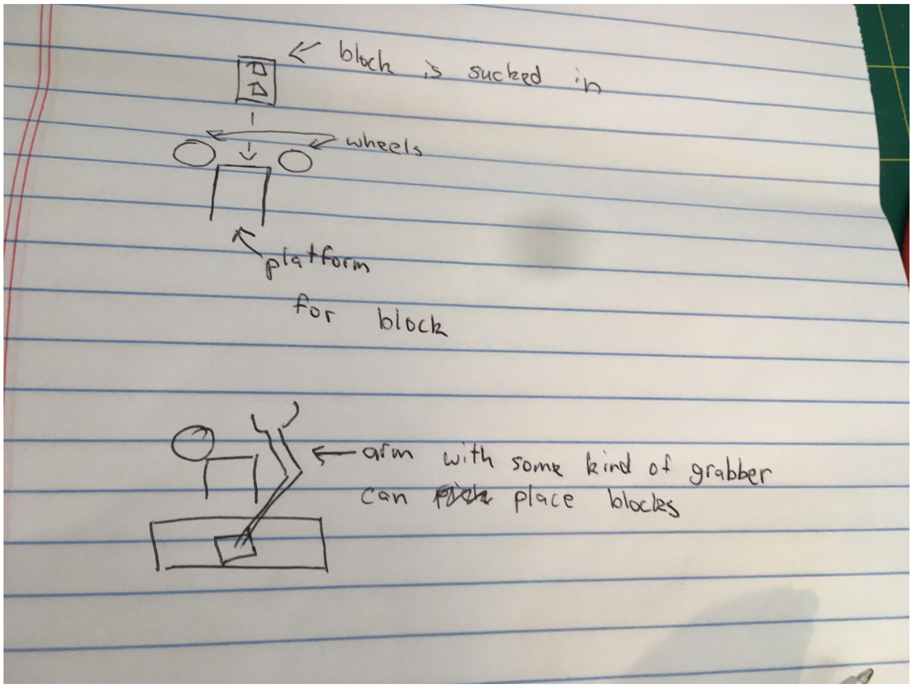
Why?

- The arm that picked up the nub was generally inconsistent and gave very little room for error
- Sucking in the block would only work if the robot was perfectly lined up and rotated correctly
- Because of the way the blocks are lined up in autonomous the robot would struggle getting a capstone out of the line up
- The 8020 extrusion was rotating causing the wheels to not touch the ground equally

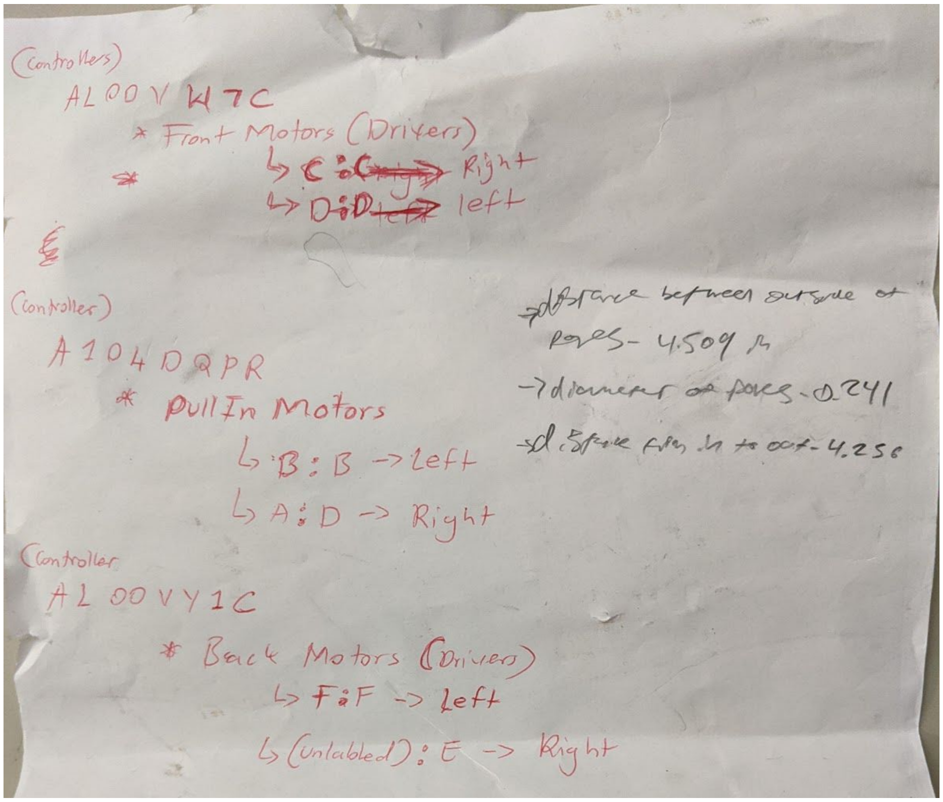
New Plan

- Use one arm for picking up and stacking blocks
- The arm will utilize a forklift with extension general form for getting to the block
- The grabber will be pincher, needing a servo for moving the pincher
- Adding cross sections connecting the two pieces of 8020 to stabilize the base

Sketching ideas to pick up the blocks:



Trying to learn how to program the Modern Robotics platform we started with:



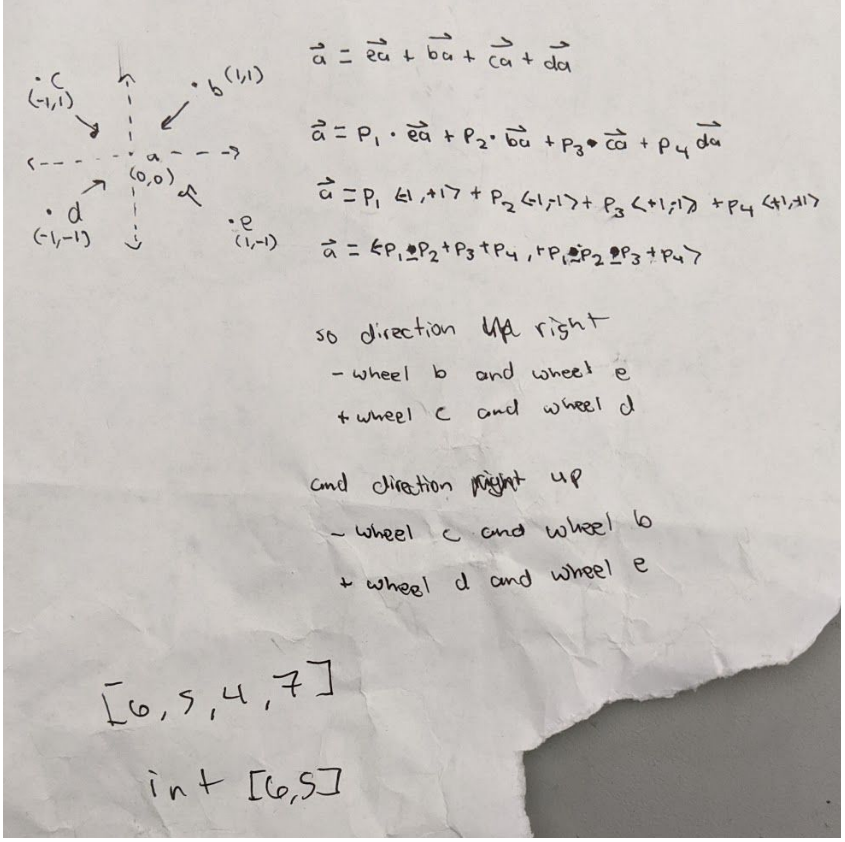
(Controllers)
AL00VW7C
* Front Motors (Drivers)
↳ C : C ~~Right~~ -> Right
↳ D : D ~~left~~ -> left

(Controller)
A104DQPR
* Pull In Motors
↳ B : B -> Left
↳ A : D -> Right

(Controller)
AL00VY1C
* Back Motors (Drivers)
↳ F : F -> Left
↳ (Unlabled) : E -> Right

-> distance between outside of poles - 4.509 in
-> diameter of poles - 0.241
-> distance from in to out - 4.256

Learning vectors to drive the mecanum wheels:



c
(-1,1)
b (1,1)
a
(0,0)
d
(-1,-1)
e
(1,-1)

$\vec{a} = \vec{ea} + \vec{ba} + \vec{ca} + \vec{da}$

$\vec{a} = P_1 \cdot \vec{ea} + P_2 \cdot \vec{ba} + P_3 \cdot \vec{ca} + P_4 \vec{da}$

$\vec{a} = P_1 \langle -1,+1\rangle + P_2 \langle -1,-1\rangle + P_3 \langle +1,-1\rangle + P_4 \langle +1,+1\rangle$

$\vec{a} = \langle P_1 - P_2 + P_3 + P_4 , +P_1 - P_2 - P_3 + P_4\rangle$

so direction ~~up~~ right

- wheel b and wheel e

+ wheel c and wheel d

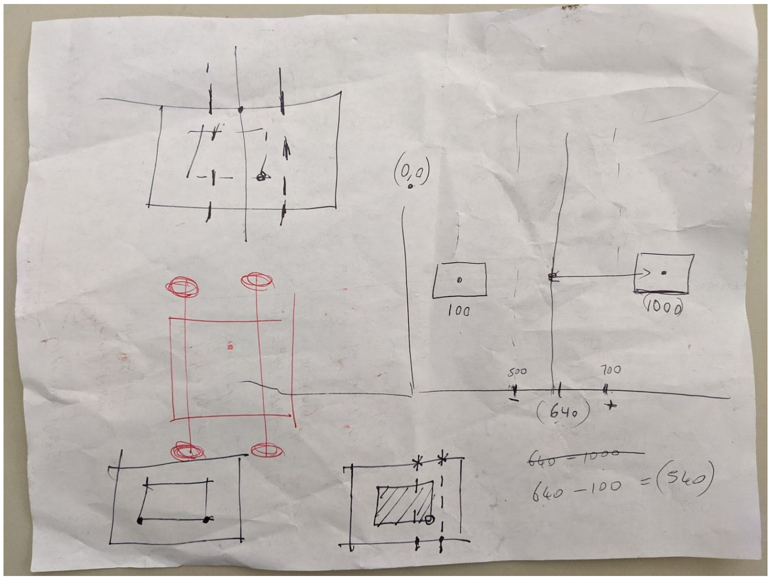
and direction ~~right~~ up

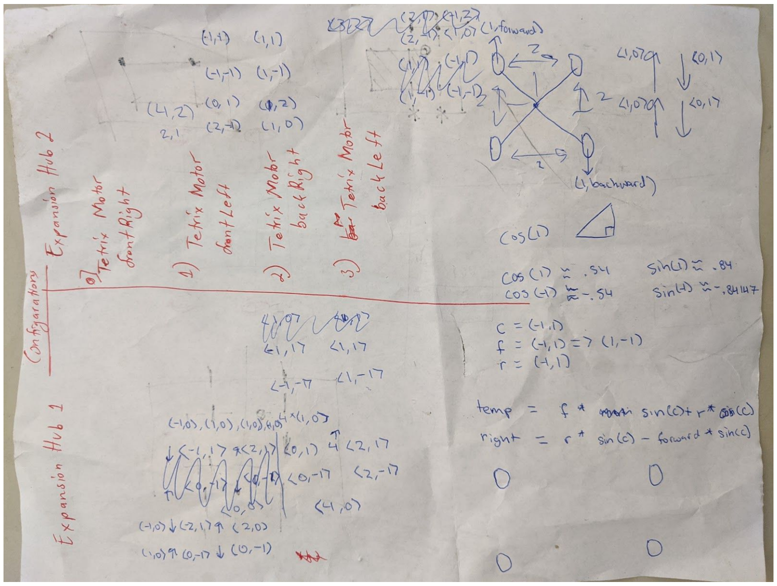
- wheel c and wheel b

+ wheel d and wheel e

[6, 5, 4, 7]

int [6,5]





First Week Brainstorming:

Pull block into robot: would provide a constant position for programmable access

Uses wheels to pull block into center area

Arm on back: would pick up block from position

Arm- How is the arm going to be able to pick up the block

- Suction - would be able to grab from many positions
  -might have problems grabbing a nub
  -would the suction device be strong enough?
  We have set motors we have to use, are they enough
  -how would you make a suction device?

- Grabbing one of the nubs on top of the block
  -would be able to grab from very specified positions
  Making it complicated in autonomous but the center platform for the block could make it reliable
  The complication might make it hard for stacking and grabbing to take place quickly
  -easier to design, could 3-D print the piece for grabbing

Decided on the nub grabber for ease of creation

Stuff not working:

-The block won't get sucked in unless it is in a very specific position
-The arm doesn't work/can't stack and pick up block
-the platform in the middle takes up to much space

New Design:

Forklift with extension: separate vertical and horizontal movement, pincher for grabbing and holding block

-Allows for easy use during autonomous

-more room for error

-simple to operate

10/28: Monday

Stuff We Did

Decided to redesign the arm. The arm will work like a forklift stack with reach.

Motor Concerns

In order to do so we will remove motors from intake device at the front. We will need one motor at the bottom to extend the arm upward. A second motor can be possibly added if the weight of the forklift is too much for the single motor. We will need another motor to extend the reach outward. We will need a servo for the grabber.

Important Dimensions

Upward Length of forklift stack: roughly 32 inches

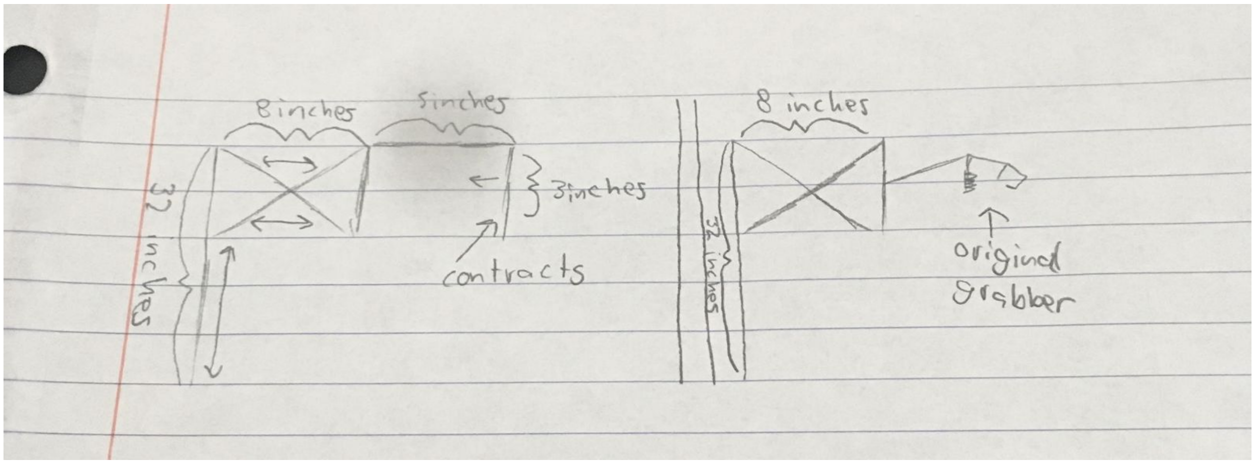
Outward reach: TBD

Basic Sketch-ish?

The arm will have a grabber that grabs both long sides of the block. Looks like |---|. Then will contract to look like |-| to pick up the block. Only one side of the grabber will move inward.

Better Sketch-ish

Would be one of the two options



10/30: Wednesday

Deconstructed Robot

Took apart the wiring and put the rev controllers on the robot.

Measured Speed

Robot has speed of 2.2 ft/s including acceleration for 10 ft.

Gripper

Tried to find usable silicone caulk for the gripper.

11/14

New IDE setup

We have been using the online IDE provided by the FTC, but now we will be switching to Android studio and the Github desktop UI for version control, the two will make it easy to code and manage our code changes. Below are the steps on how to get things running.

Android Studio (writing code)

- Open android studio and on the left sidebar of the screen Navigate to TeamCode/java/org.firstinspires.ftc.teamcode/.... In this directory is where we put all our code.
- To create a new file, the easy way is to copy and paste one of the existing files in your code directory and rename (file names begin with capital letters) the new file or you could navigate to FtcRobotController/java/org.firstinspires.ftc.robotcontroller/external.samples and look for one of the samples(template) that best fit your needs, usually we use BasicOpMode_Iterative, copy and paste it in your team code area after you can edit it.
- After coding remember to submit your changes to the GitHub account

Submitting Changes To GitHub

We are going to use GitHub Desktop UI instead of Git, this will help us push and pull our changes to our GitHub repository just like Git but in a user interface way instead of typing commands, we will have the chance to see our code history. So for the GitHub repository had to be changed to a new one because we needed an FTC SDK so that it could be recognised in the android studio, to get the SDK we had to fork the FTC GitHub repository called ftc_app found at https://github.com/ftctechnh/ftc_app . ftc_app is our new repository and all code changes must be sent there. Below is how to submit the code change:

- After coding in android studio open GitHub UI (found on desktop screen) and make sure your current repository is ftc_app
- On the left side bar it should show changes made (if you changed or added code) and at the bottom left in the summary section you must write about what you changed
- And then the commit blue button should activate, click on it to commit changes.
- In the main area after commiting,a push changes button should appear. Click on the push button to send the changes to the online (origin) repository.

NOTE: DO NOT PUSH CHANGES DIRECTLY TO THE MASTER BRANCH. Push the changes to the robo-setup branch (we will merging the two branches later after making sure no bugs in the code)

11/18
Hi everyone, today me and Chris realized that the ftc SDK for Android studio was not up to date, so I switched it to the updated version because the tensorflow sample code was for last year, for those who are in panic mode, don't panic all your code is safe and the structure is still the same.

Also Chris is really amazed how we are very good at coming up with new weekly structural ideas for our robot.

11/24
Hi everyone, today we got the drive controls working for now and the object detection, but the problem is that we don't have the arm ready.

Our next goal should be finishing up the arm so that we can implement the object detection to the Autonomous mode.
See you one tomorrow.

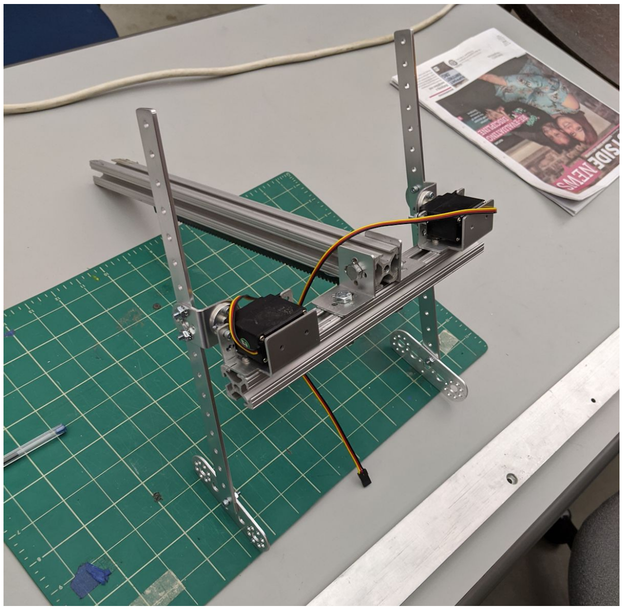
11/27
Hello all, today (Wednesday 11/27) we worked on

- understanding how vuforia, the program we are using to process images, works. We can currently use the software to output a direction that the robot should go based on the position of the block. However, we still don't completely understand vuforia, so if you are interested in learning more about it check out this series https://www.youtube.com/watch?v=2z-o9Ts8XoE (from 4-7)
- Creating a stable lift system, we 3-D printed a part for keeping the metal together and with some small adjustments the lift system should work
- Tidied up the driving code

12/15
Hello everyone, here is a quick status report of what we completed today
-Robot now can functional drive forward/backward and turn clockwise/counterclockwise
-There is a slow mode completely finished (Pressing B makes the robot go half speed)
-We have a piece for the rope on the arm to sit on while the motor moves the rope

12/23
We spent most of the time working on attaching the horizontal part of the arm to the rest of it (the pulley system) and we did not really connect the arms totally because a new design on how the arm was going to be put into place was established. We took two sliders and but the with their backs together but one of them was turned in another direction. That way the vertical part of the arm can easily attach to the horizontal part and it'll be smaller than using the extra piece of metal we had before. Next steps are to attach the arm and add the mechanism that holds blocks :0 . Oliver also had the great idea that we should have a group chat, so we are going to make the robotics group chat on snapchat! This way we'll all be more informed about the robot (since I know not many people really pay attention to emails). Competition is only a few weeks away! Oh and those of you who do see this email send me your username on snapchat or add me @blazingkai and I'll make the group chat

12/29
Arm Progress Report:
It's not done yet, there are still a few things to attach to it, but hopefully tomorrow it will be done with some help. The details on how the arm works will be explained tomorrow. Since we have like one week left our goal tomorrow is to finish the arm and make sure it works, probably our main goal this week is to fix the autonomous mode and adjust the drive controls.



12/30

	Pros	Cons
Grab on long side (no)	Already have a prototype developed	Hard to use during autonomous because you would have to do two steps to pick up a block
Grab on short side (yes)	More room for error	Nothing developed