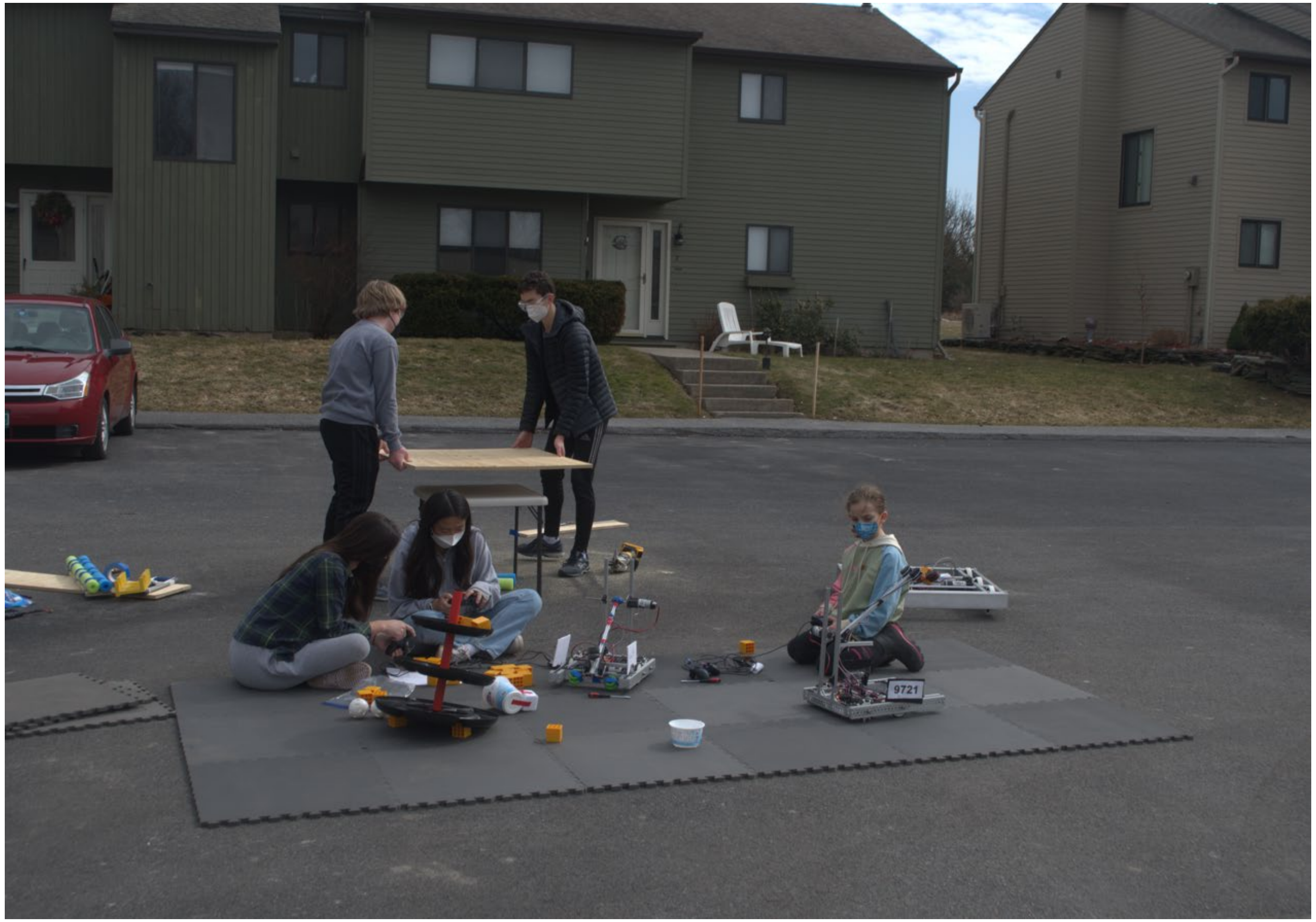
16295
Robot Presentation

About team 16295

- This is our third season of FTC as an all-girls team!
- We incorporated a rookie team from last year that we mentored to have a total of 5 girls on the team. (Kayla, Emma, Sage, Laura and Lisa)
- During our first year, our robot operated fully and delivered cargo to partner teams. We scored points in the teleOp phase. We scored 0-5 pts.
- During our second year, we participated in the virtual event and scored points in both the TeleOp and End Game phases of the competition. We scored 15-20 pts!
- This year, our robot has the potential to score points in teleOp (24 points) and end game (15 pts) for a total of 39 points!!

Team of Teams

- FRC 883(making bumpers(
- FTC 16295(fixing servo wire that came loose)
- FTC 9721(testing after tightening motor fasteners)



Season Plan

- We planned to re-use equipment from last year's teams.
- We planned to mentor team 9721 (next generation of a veteran team)
- We plan to score points during teleOp and End Game.
- We plan to use the workshop model, where we build on last year's robot and add specific new features to score more points.

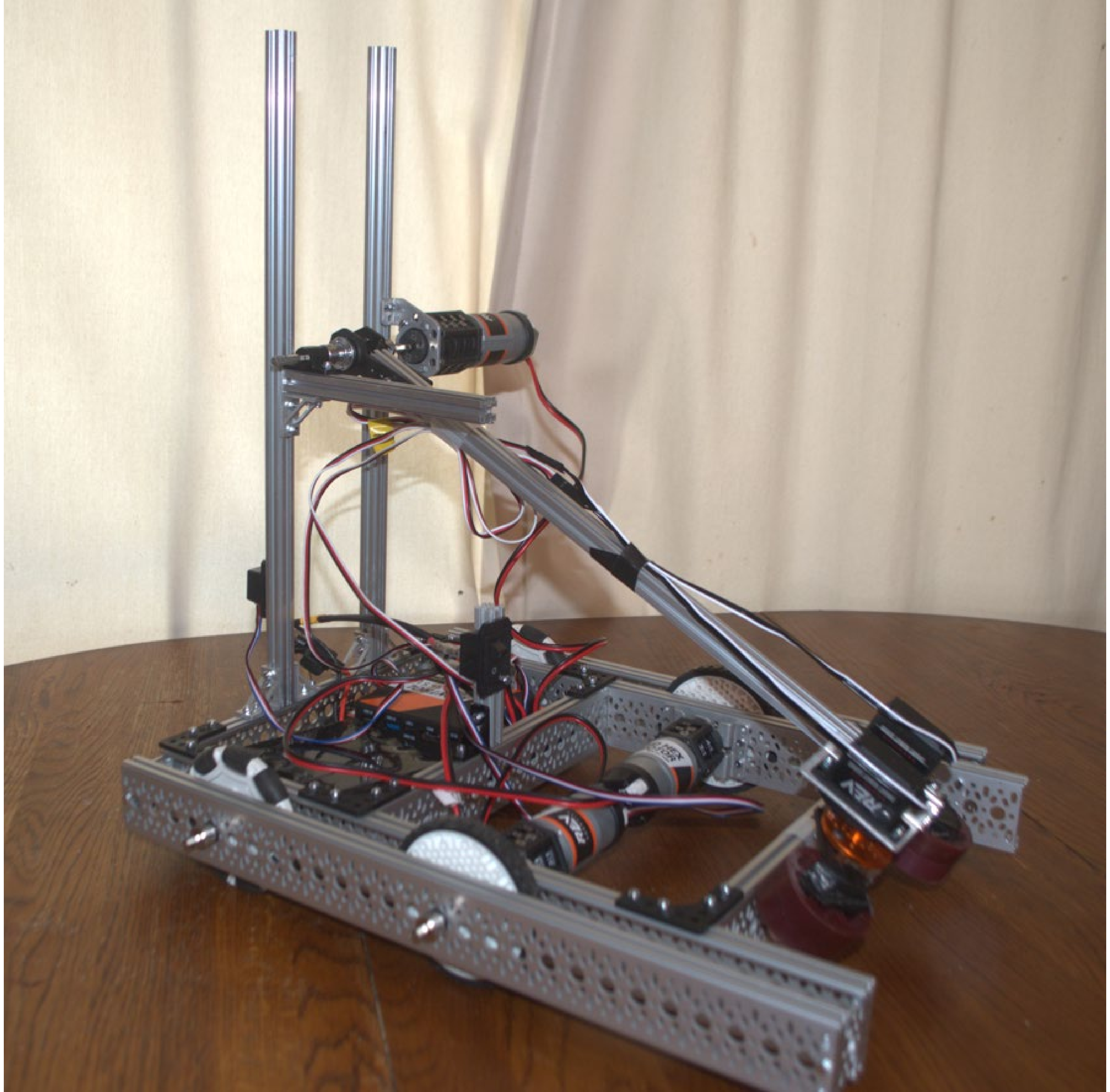
Mentoring Team 9721

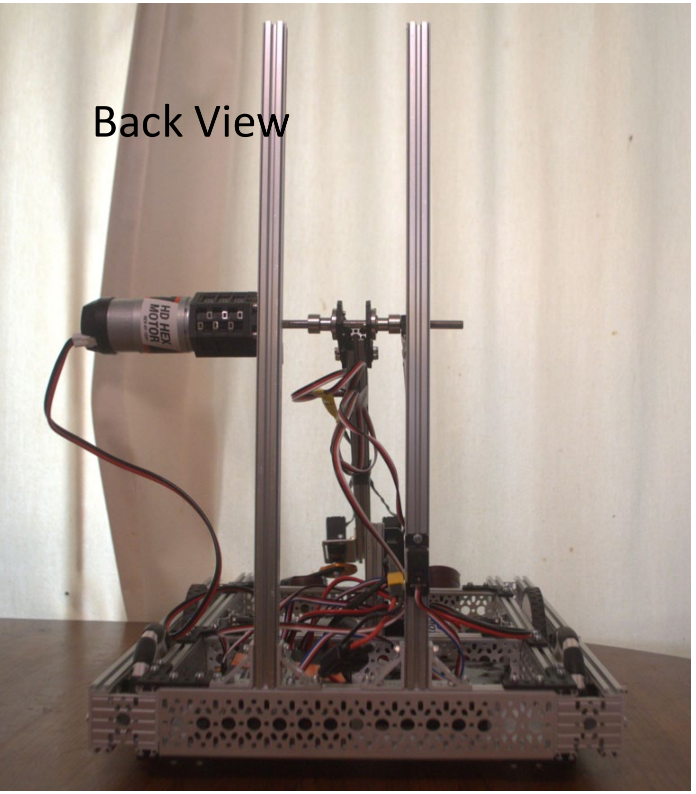
- This team is transitioning from FLL (2022 State Champs! to FTC)

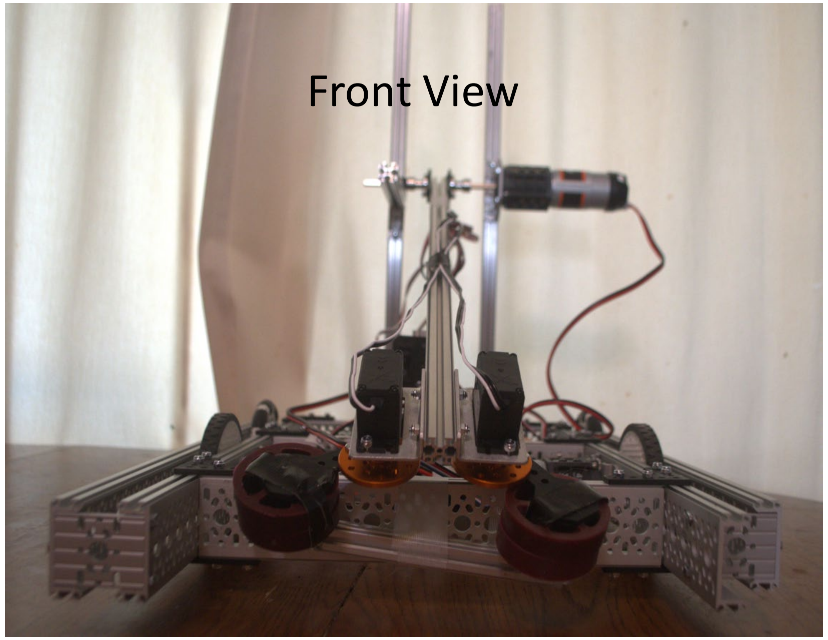
Freight Frenzie Goals

- First Plan(43)
- End Game
  - Cap the alliance tower (place it instead of cube at the start) (15)
- TeleOp
  - Place 3 cubes (from start) on the alliance tower top level (18)

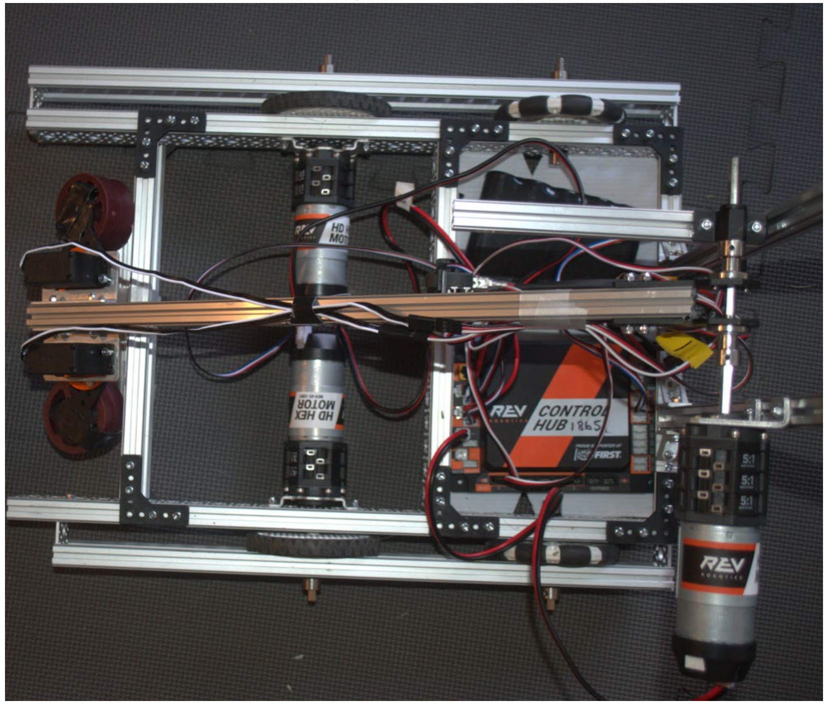
- Alternate Plan (36)
  - Tip the shared tower (20)
- TeleOp
  - Place cubes on the shared tower (6)

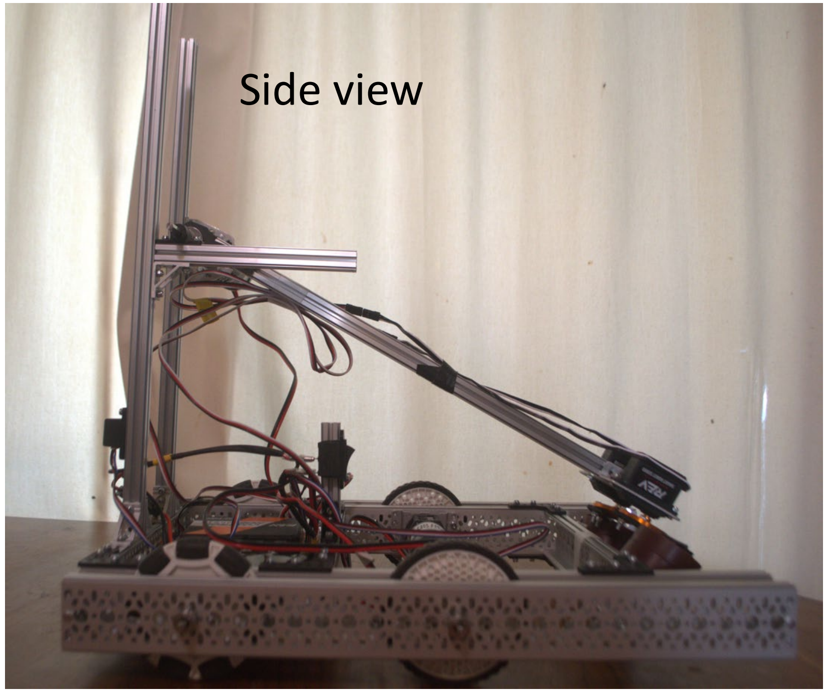






Top View





Robot Design-Chassis

- Robot Chassis was based on 2021 REV prototype (no longer available) that was used by team 18651 last year.
- We modified the chassis by removing the front omni wheels. We made this change to allow the robot to push freight around the game field.
- We used two ultra-planetary motors. We had planned to use the older 40:1 gear motors but the axle was not long enough to reach to the second C channel of the robot design.
- The motors were geared using a 12:1 ratio (3:1 and 4:1 cartridges). A longer axle was placed using the set screw so that the axle could span the distance across the channels to keep the wheels stable.
- The front bumper was moved forward to put the cubes in the correct position for collection.
- The side bumpers prevent material from getting stuck under the robot.

Robot Design-Electronics

- A floor was built by placing plastic corrugated board between the back C Channels. This creates a safe space for the control hub and battery
- The motors are attached left to right, 0-2 ports.
- The servos are attached left to right, 0-1 ports.
- A switch was mounted to turn the robot off and on.

Driver Control of Chassis

- The drive code was used both teams 16295 and 18651 last year.
- The method to drive the robot places the controls on the dpad and bumpers. This was simple and intuitive for the operators.
- The control pattern using a series of if else statements to tell the robot what to do.

Robot Design-Arm and Gripper

- The arm design was taken from the design of the arm on team 16295 from last year. The arm used three 5:1 cartridges along with setting the default 0 behavior to brake to operate the arm.
- The arm design was modified to add a second post to stabilize the arm.
- The second post was modified to align the axle on both posts.
- The gripper uses two servos to grab cubes. Cubes were selected because they are more numerous in the game and they do not roll away when you try to grab them.
- The arm collects cubes on one side of the robot and drops them on the other side of the robot. This means that the robot does not need to turn to collect items.

Driver Control Code

Code is taken from samples and modified to suit our needs.

Most of the driver code was mapped to gamepad 1 and was developed last year, including the following methods: DriveAxialForward, DriveAxialBackward, DriveRotateLeft,DriveRotateRight

The arm is controlled with direct commands, which are mapped to gamepad2.

```
//This organized the app and places the files in the teamcode section.
//This would be important if you wanted to do some more complicated
//programming and wanted access to a environment such as Android Studio.

package org.firstinspires.ftc.teamcode;

//These next few statements add specific coding to this class so that you
//can access the build-in methods of these other classes.

import com.qualcomm.robotcore.eventloop.opmode.LinearOpMode;
import com.qualcomm.robotcore.eventloop.opmode.Autonomous;
import org.firstinspires.ftc.robotcore.external.navigation.DistanceUnit;
import com.qualcomm.robotcore.hardware.DistanceSensor;
import org.firstinspires.ftc.robotcore.external.navigation.Position;
import com.qualcomm.robotcore.hardware.Servo;
import com.qualcomm.robotcore.eventloop.opmode.TeleOp;
import com.qualcomm.robotcore.hardware.DcMotor;


//copied and pasted from sample imu program

import com.qualcomm.hardware.bosch.BNO055IMU;
import com.qualcomm.hardware.bosch.JustLoggingAccelerationIntegrator;

import org.firstinspires.ftc.robotcore.external.Func;
import org.firstinspires.ftc.robotcore.external.navigation.Acceleration;
import org.firstinspires.ftc.robotcore.external.navigation.AngleUnit;
import org.firstinspires.ftc.robotcore.external.navigation.AxesOrder;
```

Arm Control

- The arm controls were placed on the buttons. This allows for a single gamepad to control the whole robot if necessary.
- The up/down controls are on the vertical buttons. The open close gripper controls are on the horizontal buttons.
- The main issue for operators is that the controls determine the movement of the motor. When the arm is past vertical, the controls that used to make the motor lift cause the motor to fall and vice versa. This confused the operators.

Cap Design

- The team design a cap using the cube as a template because the robot arm could already pick up a cube.
- The remaining shape resulted from experiments with recycled containers. The team wanted something that would easily work.
- The only issue with the cap design is that the attached cube makes the cap lean to one side. This could be a problem depending upon the cap designs of other teams. It makes sense for the robot to not double cap.

Chassis Testing

https://youtu.be/MrkUghVfw0Q

- The robot drives through obstacles
- The robot can drive inside the barriers (it was designed to be less than 13.5” wide) but it cannot drive over them.
- The robot needs at least .4 power to operate because of friction with the rubber floor.

Arm Testing

- The arm can lift all of the cubes to the highest alliance tower
- The arm can lift the cap and place it on the towers
- The arm needs the tower to be leaning toward it to place it while drive forward
- The arm can place the cap on the tower from the backwards position it needed

Gripper Testing

- Gripper uses two servos with round servo horn and compliant wheels
- Gripper can hold cube or ball
- Ball tends to roll away when we try to grab it

TeleOp testing(18 pts)

A single operator can place (3) cubes on the Highest tower for 18 points

TeleOp Testing https://youtu.be/V-nR0_SCOS4

End Game and Cap

Robot places the team designed
Element on the alliance tower

https://youtu.be/e5_u1xZvJXs

New Tool-Grinder to Cut Steel

We used a long steel axle for prototyping but were short on time.
We cut the steel using a grinder to make the axle fit inside the plane of the robot.

https://youtu.be/HlCtBIVsvmg