MANIAC MECHANICS

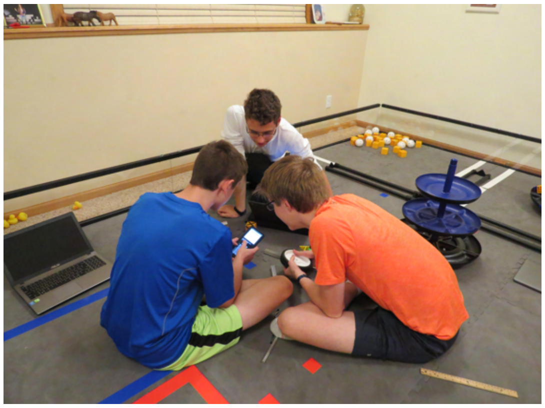
18082

2021-22 FREIGHT FRENZY ENGINEERING PORTFOLIO

Table of Contents

Who Are We?

- We are a second year FTC team with 3 returning and 3 new members.
- Emma is 13 and works on outreach, fundraising, and robot construction. She loves to do aerial silks, which is acrobatics while hanging from a long piece of fabric!
- Noah is 15 and helps with robot brainstorming and programming. He enjoys chess and piano.



- Jason is 16 and does programming, robot design and construction. He also enjoys playing tuba, piano, trumpet and guitar.
- Josh is 16 and is in charge of CAD, and also works on robot design and construction. He enjoys martial arts and playing classical guitar.
- Seth is 15 and is in charge of the Engineering Notebook and Portfolio, and also works on Robot Construction. He loves to read the works of J.R.R. Tolkien.
- Daniel is 15 and works on programming and gives input on robot construction. He has the opposite of a fear of heights.

Team Plan

- We lost two of our team members from last year, but gained three new ones who have become friends and add a lot to our team!
- We are excited for the first time to have in-person competitions with other teams!
- Despite various challenges, we have worked well as a team to brainstorm solutions and implement them
- We made strong goals and milestones for us to achieve, and have strived to meet them
- We especially wanted to keep up on due dates and milestones, because we got almost two months behind on milestones last year (COVID really saved us in terms of timing!)
- Now, if we miss a milestone, we have extra meetings until we are caught up!

How Have We Improved?

- As a 2nd year team, we have had many opportunities to grow in our skills
- Our prior FLL experiences give us a head start in design ideas and in using the core values of FIRST
- Our prior year in FTC & years in FLL gave us good experience presenting our ideas and documenting our progress after each meeting in our notebook
- We have progressed from last year's constructing our robot before doing the CAD to this year's CADing our ideas first, which we've found much more efficient!
- Jason and Daniel have learned how to program in Java and continue to enjoy the challenge of making our robot "Alfred" move in various ways!
- We also have improved in our design process, taking the time to sort through various possibilities
- Instead of coming up with ideas and just prototyping or trying them out, we brainstorm thoroughly and CAD our ideas before constructing them
- We could still get better at meeting our milestones on time :-)

Budget

Starting Balance	**+$107.92**
Dues	**+$1500**
Sponsorships	**+$1030**
Tournaments	**-$700**
Field Kit	**-$277.12**
Parts	**-$1750.19**
Supplies/Misc	**-$194.04**
Total Actual Cost:	**-$2921.35**
Current Deficit:	**-$283.43**

We were very grateful to receive sponsorships from three local companies: SlashBlue ($1500), Advanced Drafting & Residential Design ($1000), and Custom Solutions Manufacturing ($500), to help with our startup costs last year! We will look for more sponsors this year to cover our deficit, or do more fundraisers with local restaurants.

Fundraising

- We wrote and gave a Fundraising presentation for the CEO of a local company, SlashBlue, and received a donation of $1,030 which we are very grateful for!
- We emailed a local Chick-fil-A in hope of performing more fundraisers with them as soon as they begin offering them again. We help out during the dinner rush, and they give us 20% of their online sales during that time!

Game Strategies... lead to ...Design Requirements

Game Strategies...	...Design Requirements
Be able to move in any direction	Use mecanum wheels
Be able to measure movement precisely	Use odometry: 3 pods with small wheels
Be able to drive *around* the barricades	Drivetrain limited to about 12" wide
Be able to drive *over* the barricades if we want to	Protect the wheels and odometry pods in strong chassis frames, with enough ground clearance, and pods pivoting up
Deliver the freight to the correct level of the shipping hub *and* be able to cap the hubs with our Team Shipping Element	Use an extendable gripper arm, but it has to reach high *and* fit in an 18" box!
Have our arm be reasonably balanced, to not stress its main pivoting axle	Mount two of its motors behind its axle as counterweights, like a crane/backhoe
Be able to see freight well to grab it accurately.	Position the grabbers on the side of the gripper (new)
Protect the arm from large sideways impacts, like crashing into other robots	Use a clamped aluminum tube for quick release, like a bicycle seatpost
Keep the robot's center of gravity low, so it won't tip over because it's only 12" wide	Mount the motors and battery as low as possible - this meant goBilda channel
Detect the duck or Team Shipping Element during Autonomous	Use color sensors, at the proper height, and protected inside the chassis frames
Be able to rotate the Carousel during Autonomous and Endgame	Mount two rubber wheels on the outside of our robot, at carousel height
Leave enough space for our arm to pivot	Mount the Hubs in the back. Need to protect them with thick acrylic plastic!

Our Design Process

1. **Brainstorm ideas**- We think about what kind of design we want to achieve the most points per second.
2. **Research and critique ideas**- We research ideas and come up with a list. Then we critique our ideas to fit within our time and money constraints.
3. **Use pros and cons for deliberation**- After we have narrowed down our list of ideas, we list all the pros and cons of each design. Then we vote for a design(s) to move forward with.
4. **Prototype**- We prototype our design(s) with CAD, seeing if we have the necessary parts and whether or not the parts will fit together.
5. **Assemble**- We purchase the necessary parts and use CAD as instructions for the assembly.

Design Decisions and Iterations

Frame:

- We designed custom channel and had it made by SendCutSend.com, but they were unable to bend it because the flanges were too deep. We called several local machine shops, but they also couldn't. Finally the Cat in the Hat coach bent it for us.
- A bearing that sticks out a few mm from our robot caught on the gap between the glass panels. With the help of Chris (our design reviewer) we decided to sand down the edges of fender washers and mount them around the bearings with strong double-sided tape, like an inclined plane to prevent the bump.

Drivetrain:

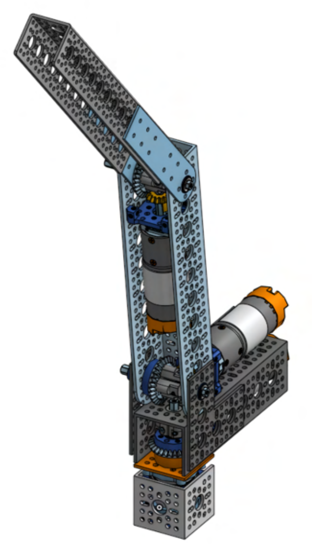
- We already had a working drivetrain from last year with vertical motors that use chain, but we wanted a more compact one that would fit this year's robot game.
- We designed custom chassis channels to fit our new thinner mecanum wheels and the gearing necessary for them.
- We also mounted the motors in goBilda channels that connected the two custom chassis parts.
- The front motors are ahead of their wheels, and the rear motors behind their wheels, by exactly the 40-tooth gear spacing, so the front and rear chassis pieces could be identical.

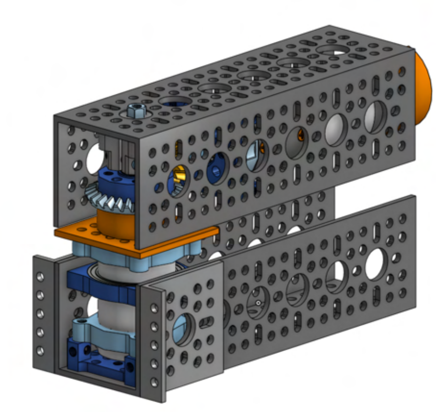
Gripper:

- We made a gripper hand using linkages, a servo, and foam tape (so we can grip all the freight types, even the heaviest cubes).
- The original Gripper was vertical, with the two claws facing downwards to be able to grab the object.
- After using our gripper we determined it would be easier to control if we came at blocks from the side rather than above.
- To accomplish this we added gears which enabled both sides of the gripper to move and the adjusted grippers allow us to target blocks more easily

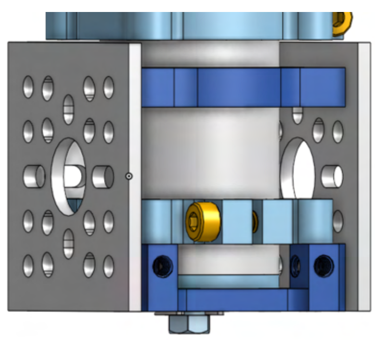
Arm:

- The arm is made of both GoBilda and Actobotics channels that interlock
- We cut and drilled custom 0.1" aluminum plates to facilitate this junction: they have holes for a goBilda bevel gear on one end and Actobotics spacing on the other.
- We put a panning motor on the bottom and another two motors with bevel gears underneath and inside the arm to power the other directions it moves.
- We used an elbow joint and a shoulder joint for greater precision:





- Panning motor (top) mounted on chassis tube (bottom)
- We thought of a unique tubular mount that clamps to the chassis on the bottom and to the bevel gear for panning on top
- If the Arm is ever whacked from the side, this tube can twist within its clamps, like a bike's seatpost, and the motors and gears won't be damaged!
- The first iteration is below; the second is above (better support, with a channel the entire width of the robot).

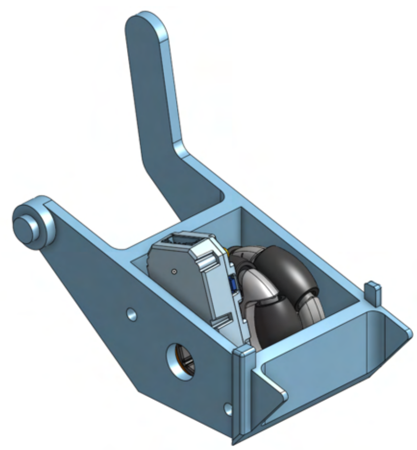


Color sensors:

- We use color sensors to sense for our team shipping element so that we can determine the correct level to put the freight in.

Hubs and cables:

- Our control and expansion hub are mounted on the back of our robot, so they are easy to access and out of Arm's way.
- We used zip ties, metal tape, and spiral wrap to make sure our cables stay in place.



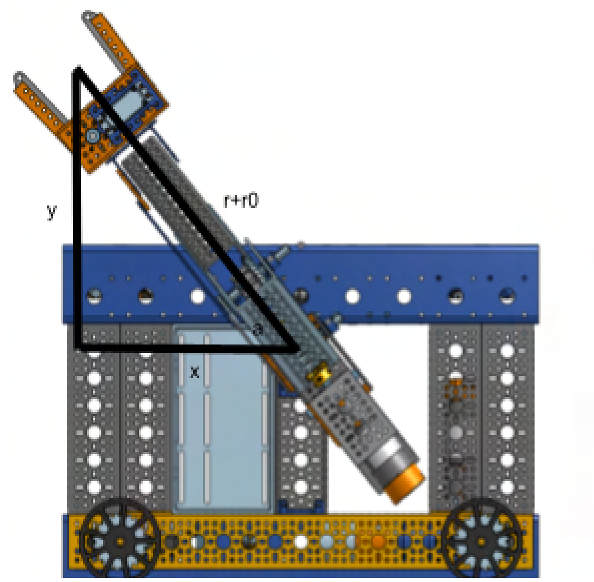
Odometry:

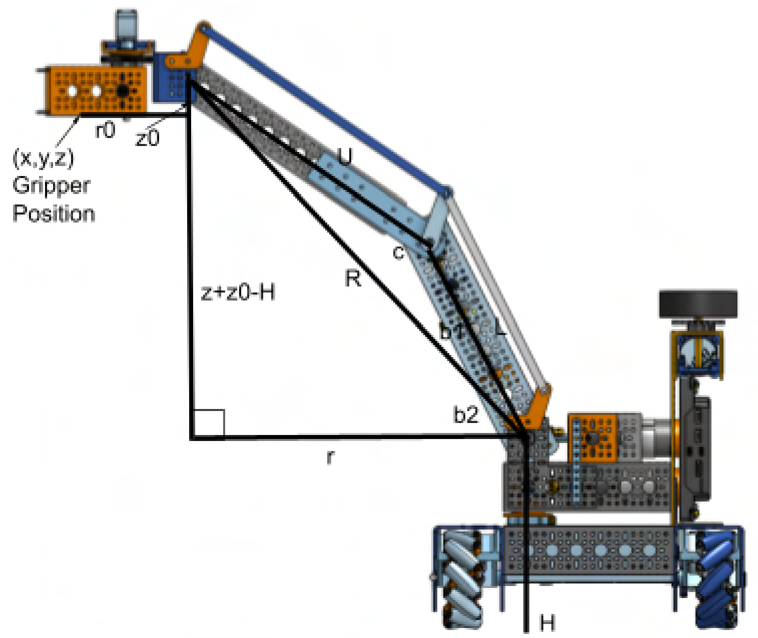
- We designed and 3D printed odometry pods, and made sure they can go over the barriers, but the code is not quite done.
- There were other things we had to get done in order for the robot to function, and we can get a decent score without odometry.
- The odometry pods are sloped, and pivot, so that we can go over the barriers and gravity will pull the pods back to the mat.

Discarded Design Ideas:

- Last year's drivetrain was too big/complex, and not well designed for this year's game.
- Our new geared drivetrain is faster, more compact, has a lower center of gravity, and is able to drive over the barriers.
- We almost decided to use side intakes and an elevator-like arm with different levels.

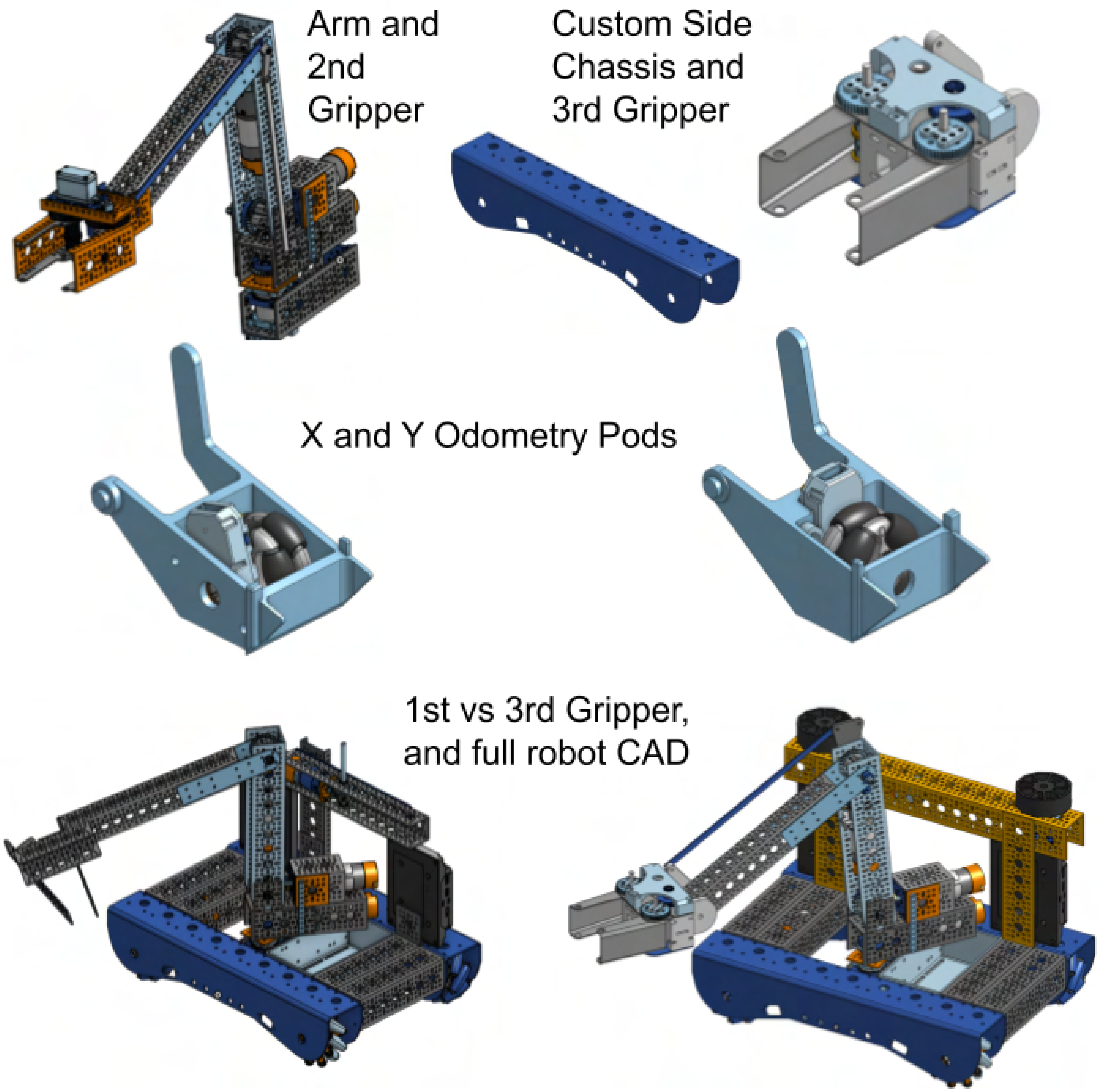
Math





- We want to find angles a, b=b1+b2, and c to program our motors, and angle a is just $a=\tan^{-1}(y/x)$, if we use the atan2 function to be careful with the signs.
- We can measure distances L, U, and H, the height of the shoulder pivot.
- We can compute R using r, z+z0-H, and the Pythagorean Theorem.
- We can compute $b2=\tan^{-1}[(z+z0-H)/r]$.
- Then we use the Law of Cosines which states $C^2=A^2+B^2-2AB\cos(c)$, where angle c is opposite side C.
- For our robot this translates to $R^2=L^2+U^2-2LU\cos(c)$, so $c=\cos^{-1}[(R^2-L^2-U^2)/(-2LU)]$ is our equation..
- To find angle b1, we again use the Law of Cosines and get $b1=\cos^{-1}[(L^2-U^2-R^2)/(-2UR)]$.
- This is how we convert the freight position x, y, and z to the arm angles a, b, and c! The gripper-hand always stays level because of the linkages attached to the shoulder, elbow, and wrist.
- We wrote Java functions moveArmToPosition(x, y, z) and moveArmToAngle(a, r, z).
- We wrote TeleOp code where joysticks move r and z, and the code converts to the proper angles b and c in real time!

CAD


Arm and 2nd Gripper
Custom Side Chassis and 3rd Gripper
X and Y Odometry Pods
1st vs 3rd Gripper, and full robot CAD

Programming: some of our TeleOp Code

```
// Send calculated power to wheels
if (gamepad1.left_bumper)
  powerScale = 0.5;
else
  powerScale = 1.0;
leftFront.setPower(powerScale * (-x - y - xTurn));
rightFront.setPower(powerScale * (x - y - xTurn));
leftRear.setPower(powerScale * (-x + y - xTurn));
rightRear.setPower(powerScale * (x + y - xTurn));

// This will open or close the gripper hand
if (gamepad2.x && (handTime.seconds() > 0.25)) {
  if (_isHandOpen) {
     hand.setPosition(0.57); //closed
     _isHandOpen = false;
  }
  else {
     hand.setPosition(0.425);
     _isHandOpen = true;
  }
  handTime.reset();
}

// Right bumper: “Precise” mode
if (rb){
  leftFront.setPower(0.05);
  leftRear.setPower(0.05);
  rightFront.setPower(-0.05);
  rightRear.setPower(-0.05);
}
```

```
// Spin the carousel, with ramping so ducks don’t fly
off: increase power by only 0.1 each loop!
double MAX_CHG = 0.1;
if (gamepad1.b){
   if (carousel.getPower() + MAX_CHG < 1.0) {
      carousel.setPower(carousel.getPower() +
MAX_CHG);
   }
   else{
      carousel.setPower(1.0);
   }
} else {
   carousel.setPower(0.0);
}

// TELEMETRY EXAMPLE
telemetry.addData("Colors", "Left (%d,%d,%d) Right
(%d,%d,%d)",
leftColor.red(),leftColor.green(),leftColor.blue(),
rightColor.red(),rightColor.green(),rightColor.blue());
telemetry.update();

// PANNING MOTOR
  peter.setPower(pb2);
// SHOULDER MOTOR
if (sb2 > 0.2) {
  shoulder.setPower(0.7 * sb2);
}
else {
  shoulder.setPower(sb2);
}
```

Programming: our new Robot class

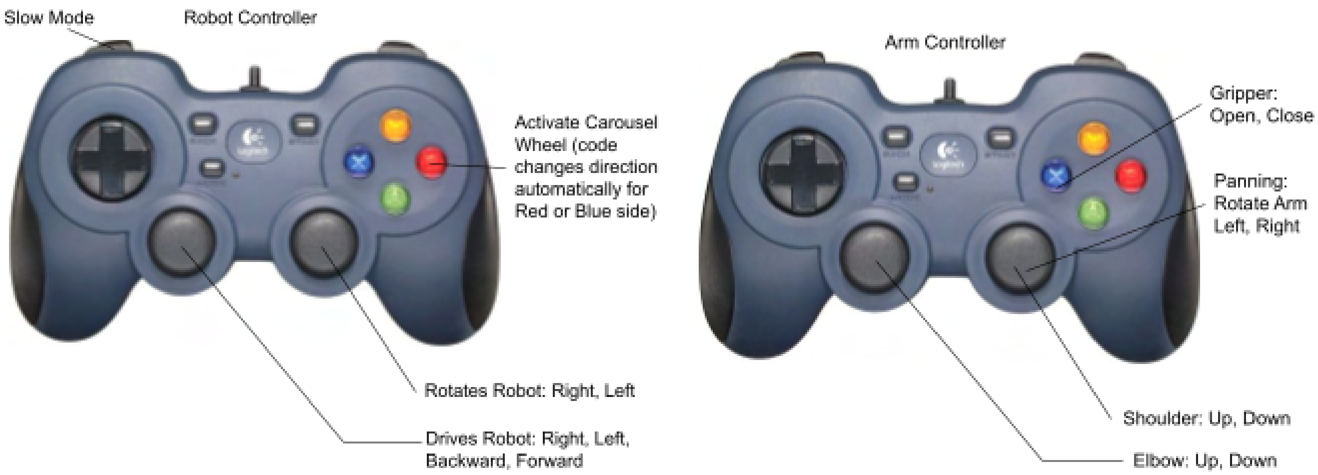
```
public Robot(HardwareMap hardwareMap, boolean
resetEncoders, int isRed) {  // red = 1, blue = -1
   … // initialize motors from hardwareMap
   _aPower = _rPower = _zPower = 0.0;
   STARTUP_PAN_ANGLE = (90 + (40*isRed));
   for (int i = 0; i < 7; i++) {
      _motor[i].setDirection(FORWARD);
      _motor[i].setMode(RUN_USING_ENCODER);
   }
   shoulder.setDirection(REVERSE); // so encoder counts
increase with angle
   elbow.setDirection(REVERSE);    // "
   if (resetEncoders) {
   for (int i = 0; i < 7; i++)
        _motor[i].setMode(STOP_AND_RESET);
   }
   for (int i = 4; i < 8; i++) {
        _motor[i].setPower(0);
        _motor[i].setZeroPowerBehavior(BRAKE);
     }
     computeArmRZ();
 }
```

```
// MOVE THE ARM TO A POSITION RELATIVE TO
ITS ORIGIN (which is on the mat directly below the
shoulder pivot)
public void moveArmToPosition(double x, double
y, double z, double power) {
   double a = Math.atan2(y, x);
   double rr = Math.sqrt(x * x + y * y); // r + r0
   Log.d("", String.format("moveArmPos: x %f, y %f, z
%f, a %f deg, rr %f", x, y, z, Math.toDegrees(a), rr));
   moveArmToAngle(a, rr, z, power);
}
```

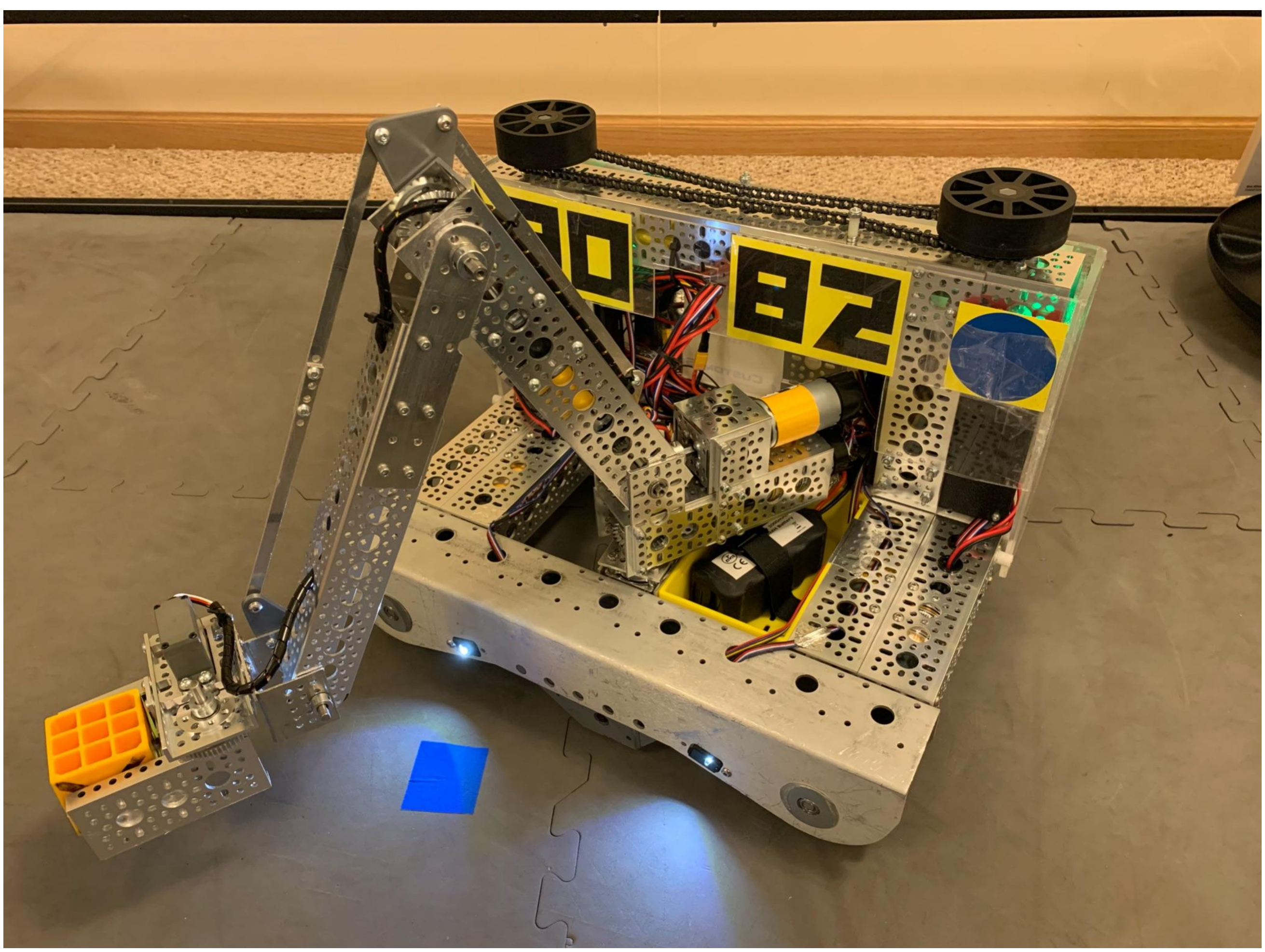
```
// MOVE THE ARM TO AN ANGLE/POSITION , more
intuitive for driving.  a is the panning angle in
// degrees clockwise from +y, r the distance from the seat
post to the drop point (includes r0),
// and z the height above the mat.
public void moveArmToAngle(double a, double rr,
double z, double power) {
    pan.setTargetPosition((int) ((STARTUP_PAN_ANGLE -
Math.toDegrees(a)) / ARM_DEGREES_PER_COUNT));
    pan.setMode(RUN_TO_POSITION);
    pan.setZeroPowerBehavior(BRAKE);
    pan.setPower(power);
    Point bc = computeArmBC_Radians(rr, z);
    shoulder.setTargetPosition((int) ((Math.toDegrees(bc.x) -
STARTUP_SHOULDER_ANGLE) /
ARM_DEGREES_PER_COUNT));
    shoulder.setMode(RUN_TO_POSITION);
    shoulder.setZeroPowerBehavior(BRAKE);
    shoulder.setPower(power);
    elbow.setTargetPosition((int) ((Math.toDegrees(bc.y) -
STARTUP_ELBOW_ANGLE) /
ARM_DEGREES_PER_COUNT));
    elbow.setMode(RUN_TO_POSITION);
    elbow.setZeroPowerBehavior(BRAKE);
    elbow.setPower(power);
}

public void computeArmRZ() {
    // Compute angles b and c, and then positions r and z,
from current shoulder and elbow positions
    _angleB =
Math.toRadians(STARTUP_SHOULDER_ANGLE +
shoulder.getCurrentPosition() *
ARM_DEGREES_PER_COUNT);  // upward from horizontal
    _angleC = Math.toRadians(STARTUP_ELBOW_ANGLE +
elbow.getCurrentPosition() *
ARM_DEGREES_PER_COUNT); // outward from zero
(which is unreachable)
    _armR = ARM_U * Math.cos(_angleB) + ARM_L *
Math.sin(_angleB + _angleC - 0.5 * Math.PI) + ARM_R0;
    _armZ = ARM_U * Math.sin(_angleB) -
ARM_L * Math.cos(_angleB + _angleC - 0.5
* Math.PI) + ARM_H - ARM_Z0;
}

public Point computeArmBC_Radians(double rr, double
z) {
    double r = rr - ARM_R0;
    if (z < 0.0)
        z = 0.0;  // can't go below mat!
    double zTmp = z + ARM_Z0 - ARM_H;
    double b2 = Math.atan2(zTmp, r);
    double R = Math.sqrt(r * r + zTmp * zTmp);
    double c = Math.acos((R * R - ARM_L * ARM_L - ARM_U
* ARM_U) / (-2 * ARM_L * ARM_U));
    // TODO: check that these angles are reachable!
    double b1 = Math.acos((ARM_L * ARM_L - ARM_U *
ARM_U - R * R) / (-2 * ARM_U * R));
       Point bc = new Point(b1 + b2, c);  // return the total arm
angles in a pair
    return bc;
}
```

```
public void update(boolean overrideGuards) { //set
all motor powers
    // Set the panning speed (easy):
    ...
    // Set the elbow and shoulder speeds (hard).
    // first compute where the arm is right now:
    computeArmRZ();
    // and make sure it doesn't go into the mat unless
we're overriding:
    if ( _armZ < 0.0 && !overrideGuards ) {
        _armZ = 0.0;
        if (_zPower < 0.0)
            _zPower = 0.0;
    }
    // then compute the direction we want the arm to go:
r + delta r, z + delta z:
    Point bc = computeArmBC_Radians(_armR +
_rPower, _armZ + _zPower );
    double b = bc.x;
    double c = bc.y;
    // but don't let the shoulder go past vertical:
    if ( b > Math.toRadians(90) && !overrideGuards ) {
        b = Math.toRadians(90);
    }
    // or the elbow go outside the range [40, 180]
degrees:
    if ( c > Math.toRadians(180) && !overrideGuards ) {
        c = Math.toRadians(180);
    }
    if ( c < Math.toRadians(40) && !overrideGuards ) {
        c = Math.toRadians(40);
    }

    // Now compute the changes we want in the b and c
angles,
    double bChange = b - _angleB;
    double cChange = c - _angleC;
    // normalize them so they're never too small,
    double maxChange = Math.sqrt(bChange * bChange
+ cChange * cChange);
    if ( maxChange != 0.0 ){
        bChange /= maxChange;
        cChange /= maxChange;
    }
    // and scale them by the maximum of r and z power:
    double maxPower = Math.min(1.0,
Math.sqrt(_rPower * _rPower + _zPower * _zPower));
    shoulder.setPower(maxPower * bChange); // always
in [-1,1]
    elbow.setPower(maxPower * cChange);    // "
    if ( _aPower != 0.0 || _rPower != 0.0 || _zPower !=
0.0 ) {
        Log.d("MM", String.format("aP %.2f, rP %.2f, zP
%.2f, r %.2f, z %.2f, b %.2f deg, c %.2f, bCh %.2f, cCh
%.2f",
                _aPower, _rPower, _zPower, _armR, _armZ,
Math.toDegrees(_angleB), Math.toDegrees(_angleC),
                maxPower * bChange, maxPower *
cChange));
    }
} // end update()
```

Robot Controls



Robot Photo



Outreach

Visited Spontaneous Construction, 9/7/21

- We ran an Ultimate Goal match together
- We received many recommendations, ideas and critiques from them including:
    - Use way more 3D parts
    - Use spring-loaded odometry pods
    - Use PETG for high-stress parts and PLA (polylactic acid) for the rest
    - Communicate with your alliance partners about autonomous and game strategy
    - Make multiple autonomous programs

Planned to tour a local manufacturer, Boeser Sheet Metal, 11/19/21

- They canceled that morning, probably due to rising COVID cases, and have not been willing to reschedule

Hand in Hand Christian Montessori and St. Croix Preparatory Academy, 1/5/22

- We did robot demonstrations for two of our schools, bringing part of our mat and wall, and one Carousel
- Jason attends St. Croix Preparatory Academy
- Noah, Emma, and Seth attend Hand in Hand
- At Hand in Hand, we talked about FIRST and FTC in general to over 50 students
- We demoed our robot and autonomous programs
- We answered questions from teachers and students
- The students were very interested in the mecanum wheels and enjoyed watching the robot delivering freight
- At St. Croix Prep, we demoed the robot and autonomous to a dozen students and Jason's former Science teacher
- We talked about our robot and a lot about how it works

Scrimmage hosted by SponCon at Irondale HS, 2/5/22

- We went to a scrimmage with teams 14779, 19706, and 13787
- We tested our programming and practiced driving in 6 matches
- There were 2 refs who took our questions
- They told us that we can't plow freight out of the warehouse, and warned us to not bump the hubs
- Our highest unofficial robot run was 188, with team 13787

(Hand in Hand, ~50 students and teachers, during lunch)

(St. Croix Prep, ~15 students and teachers, after school)

Qualifying Tournaments: Dec 4 & Jan 15

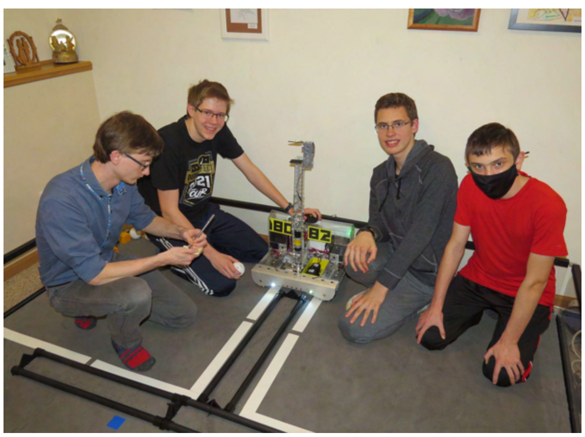
- Our first was at Maple Grove High School
- We finished with the 7th best robot score
- We were thrilled to earn the 3rd place Inspire Award, Connect Award, and we qualified for State

- Our second was at Burnsville High School
- We had the 10th best robot score
- We earned the 1st place Inspire Award and were on the second place alliance

Design Reviews

Chris Bankers, Systems Verification Engineer, 11/30:

- A Design Review helps you reprioritize your prototype to better fit the requirements
- We shared important features of the robot that had strategic implications like the arm, carousel motor, and size to get his feedback on their effectiveness
- He helped us work through a problem: the drive shaft bearings stick 2mm out of the chassis, and “catch” on gaps in the field wall. We decided to solve it by sanding down the edges of some big washers and taping them around the bearings with 3M Kapton tape with 966 adhesive



Heidi Koopman, Graphic Designer, 11/23:

- She reviewed our Portfolio and gave us several pointers and ideas including:
  - Have the Table of Contents on its own page
  - Have the Team Profile on its own page
  - Have more space between the headers and the beginning of the text
  - Less space between lines
  - More space between bullet points
  - Use 10 or 11 pt font
  - Use 2 columns with text and pictures
  - Use ¾ to 1 inch margins
  - Use color on headers and title page
  - Shrink header
  - Put page numbers next to header
  - Use bold font for sub-headers
  - Use sketches for pictures

Season Milestones

By 9/14	Improve the Autonomous driving accuracy;, take old robot apart.
By 9/18	Have a physical drivetrain; CAD at least two new designs.
9/28	Brainstorm; work with new team members
10/5	Decide on season timeline and milestones; brainstorm
10/19	Have all needed parts, begin construction on Chassis and Gripper
10/26	Drivetrain complete (10/30)
11/2	Gripper ARM and hand complete (11/9)
11/9	Programming complete (12/2)
11/16	Robot assembly complete (12/2)
11/23	Final Robot Complete. Zoom with graphic designer.
11/30	Practice driving
12/2	Dress rehearsal! Presentations, questions, robot runs.
12/4	First tournament, at Maple Grove High School 7:30. Leave 7 am!
12/7	Debrief, decide tasks, and set milestones until 2nd tournament
12/14	Finish fundraiser slides
1/4	Prepare for HIH demo, Horizontal Hand, get Odometry working
1/11	Finish Portfolio
1/13	Dress Rehearsal, Scouting Sheets
1/15	Second tournament, at Burnsville High School. Leave 7:10 am!
1/25	Finish Website and Team Video
2/1	Prep for 2/5 scrimmage with Spontaneous Construction
2/8	Finish Portfolio and make decision on Odometry
2/10	Dress rehearsal for State, and practice robot driving!