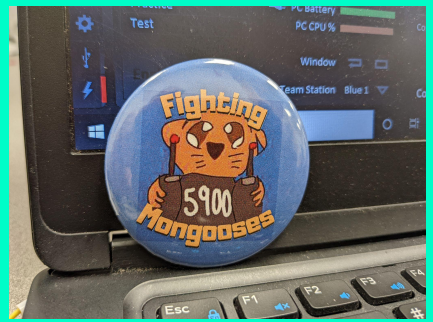
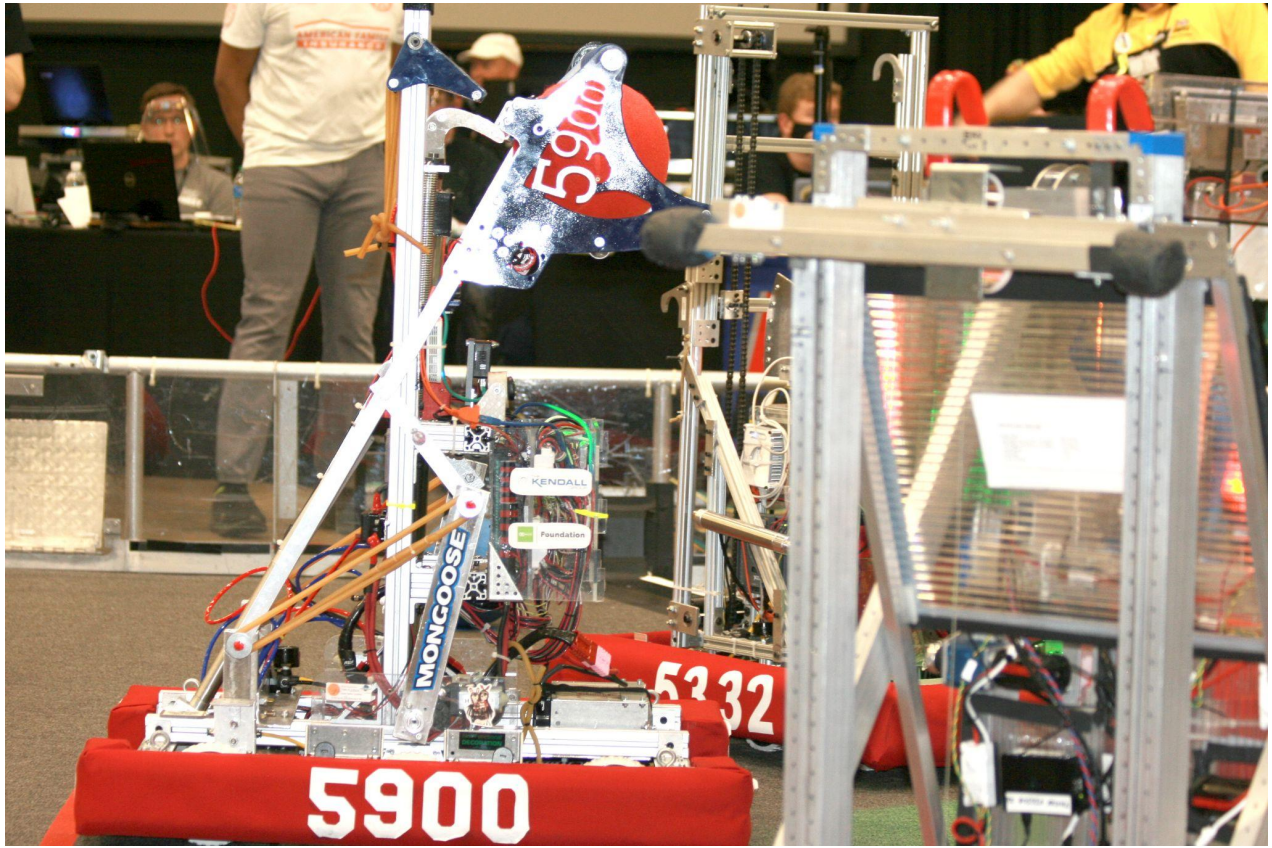
Rapid react 2022
"Miss-aligned"



¡Las Mangostas luchadoras
Team 5900 programming manual
Last update - February 27, 2022

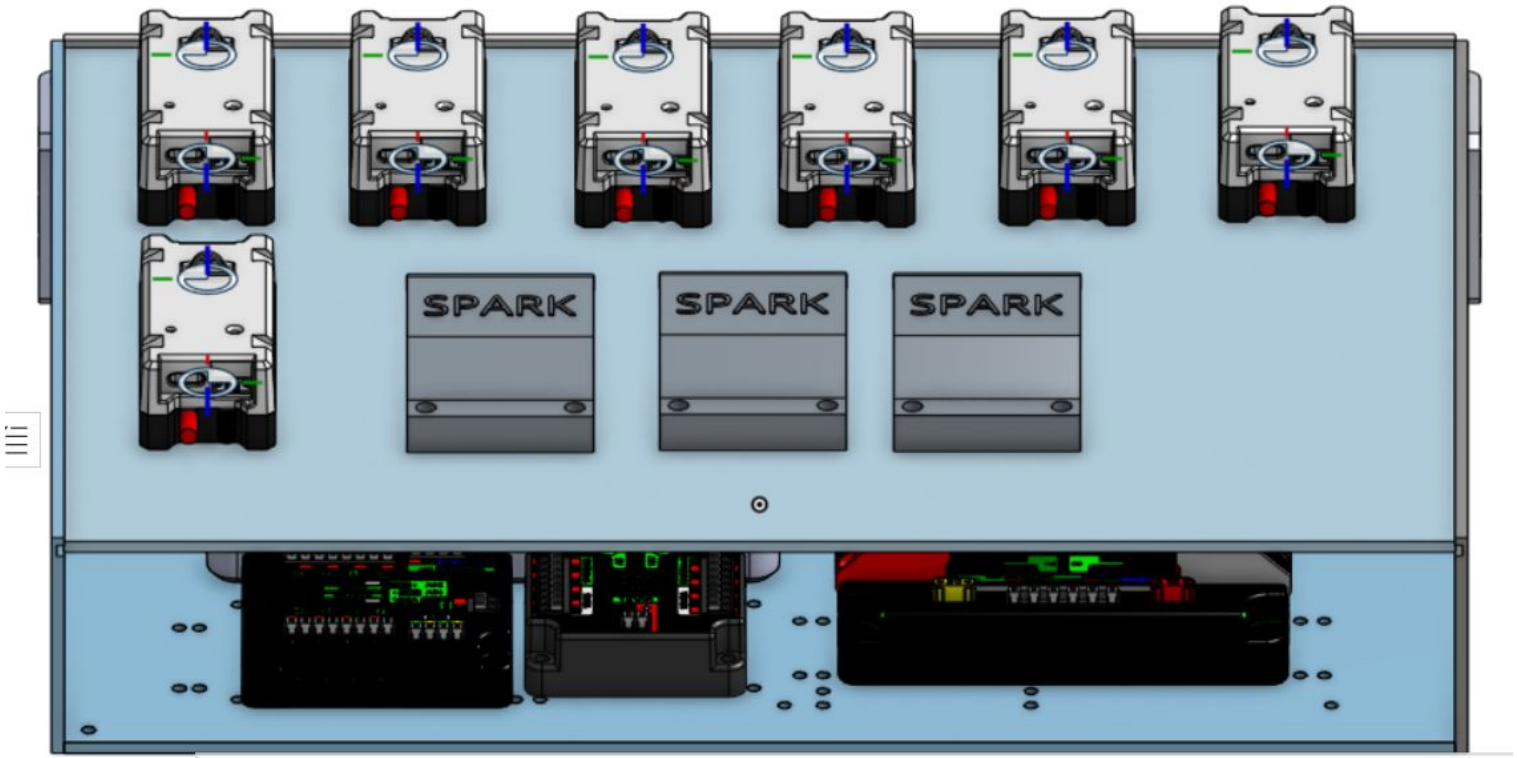
"Miss-aligned"



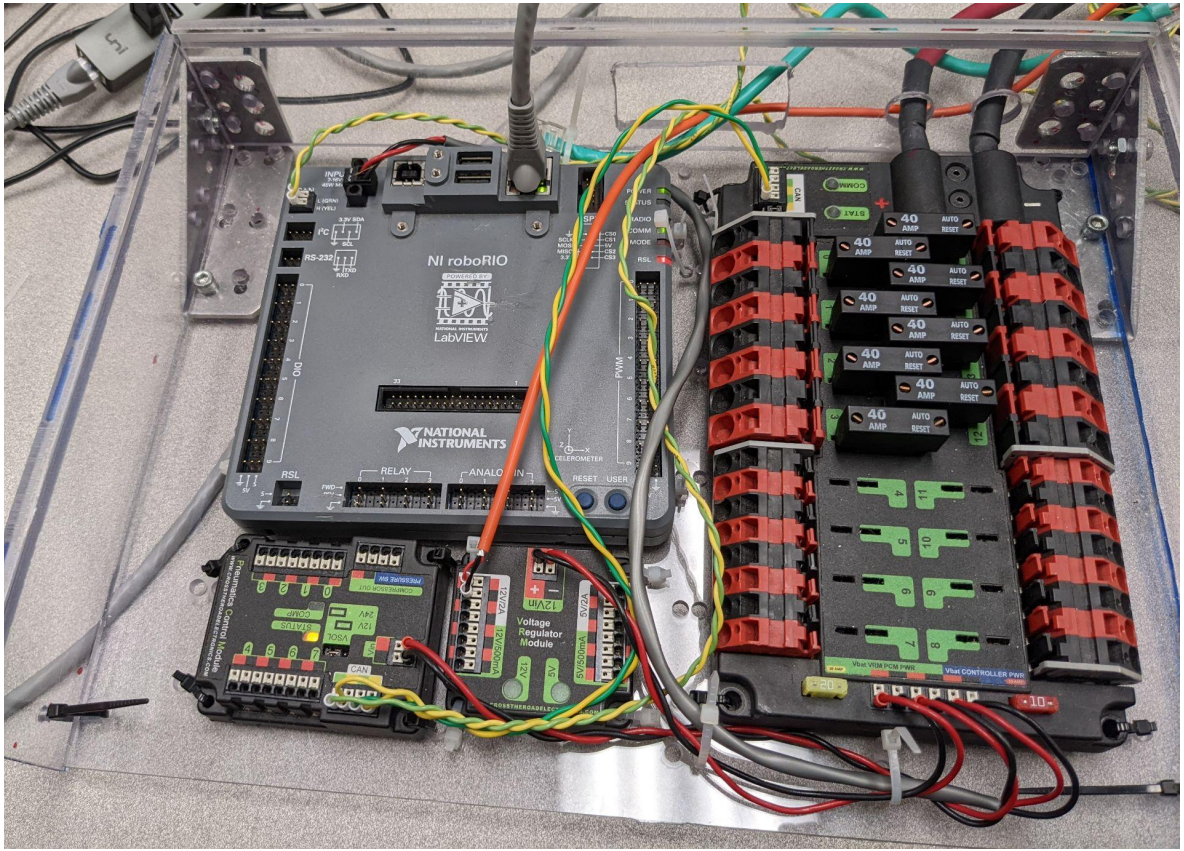
"Miss-aligned"

CONTROLS

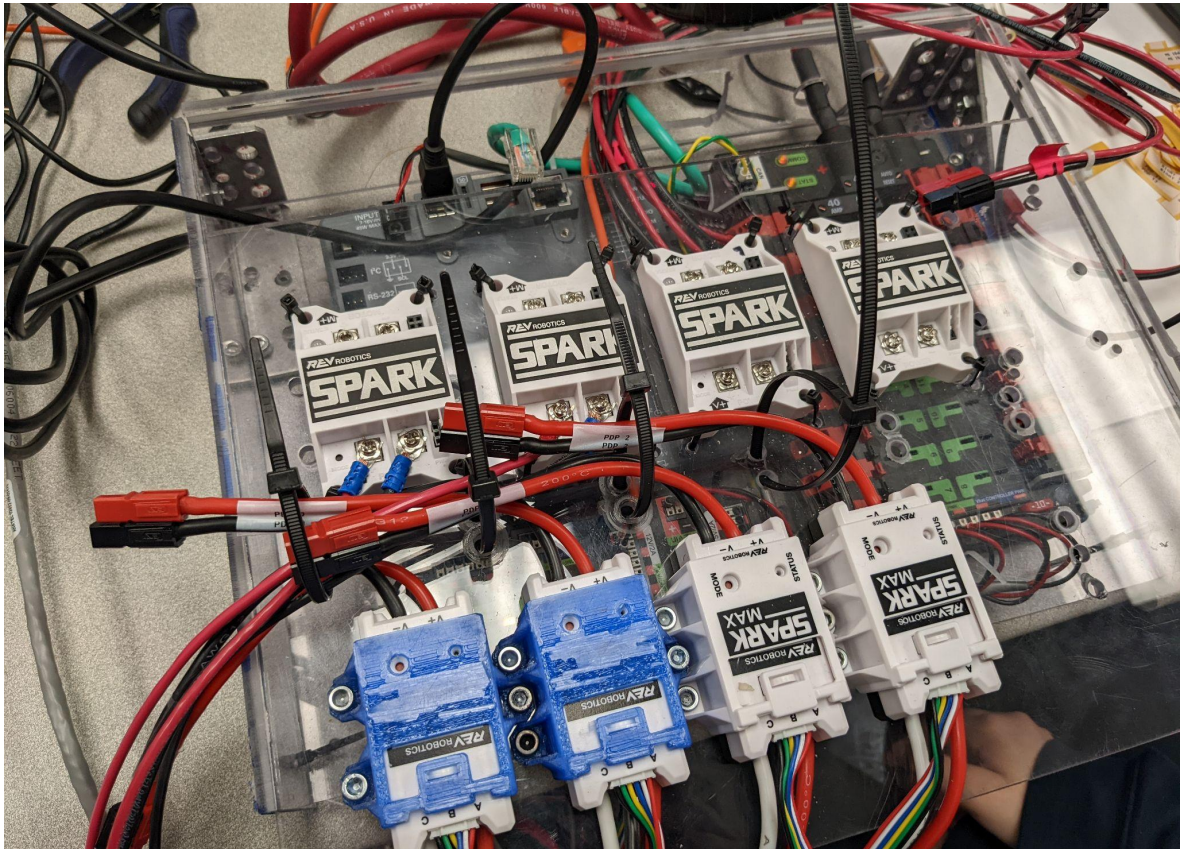
Controls enclosure



Controls enclosure - roboRIO / PDP panel

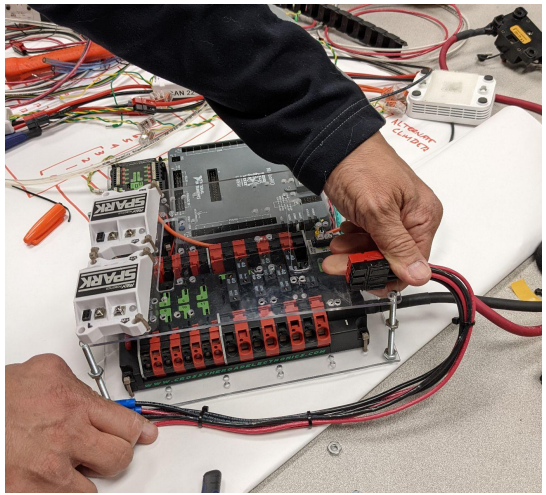


Controls enclosure - top panel for sparkmax and spark drives

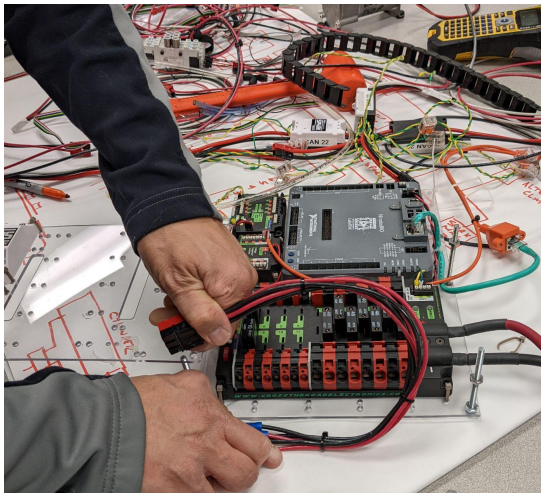


Controls enclosure cable movement

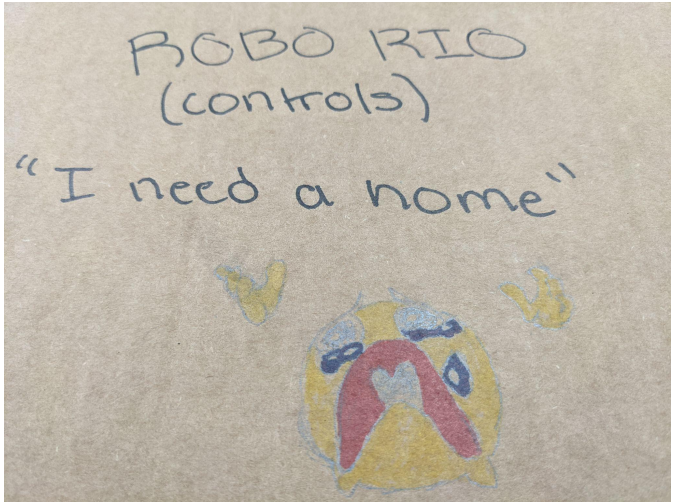
Shelf in position



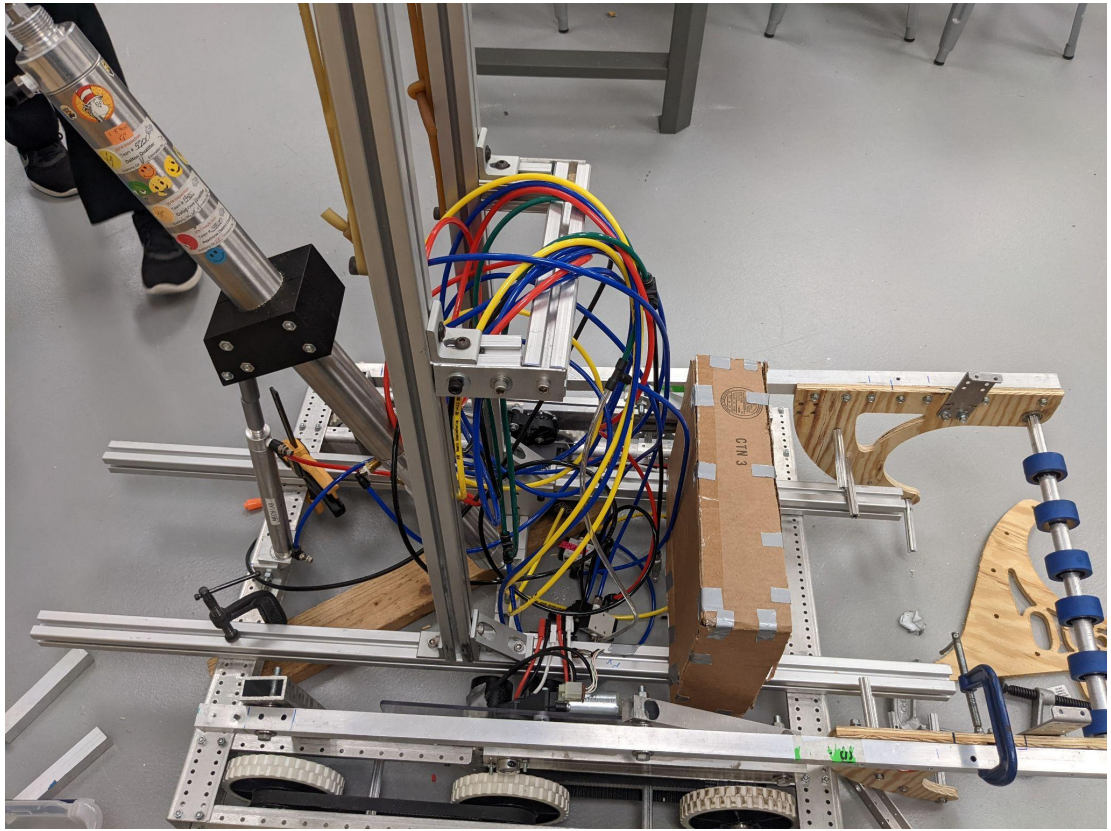
Shelf sliding out



Possible home for the RIO

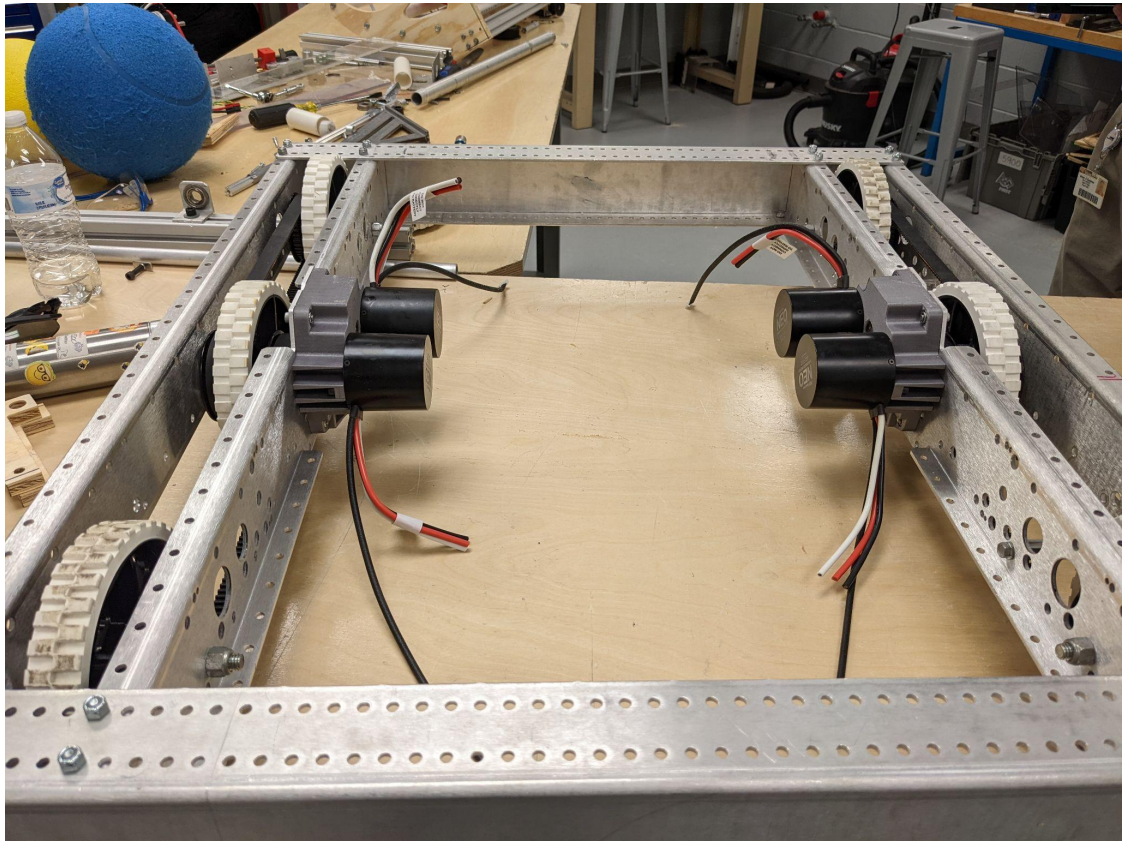


Try to keep battery at center of the robot as low to ground as possible for stability, but easily accessible

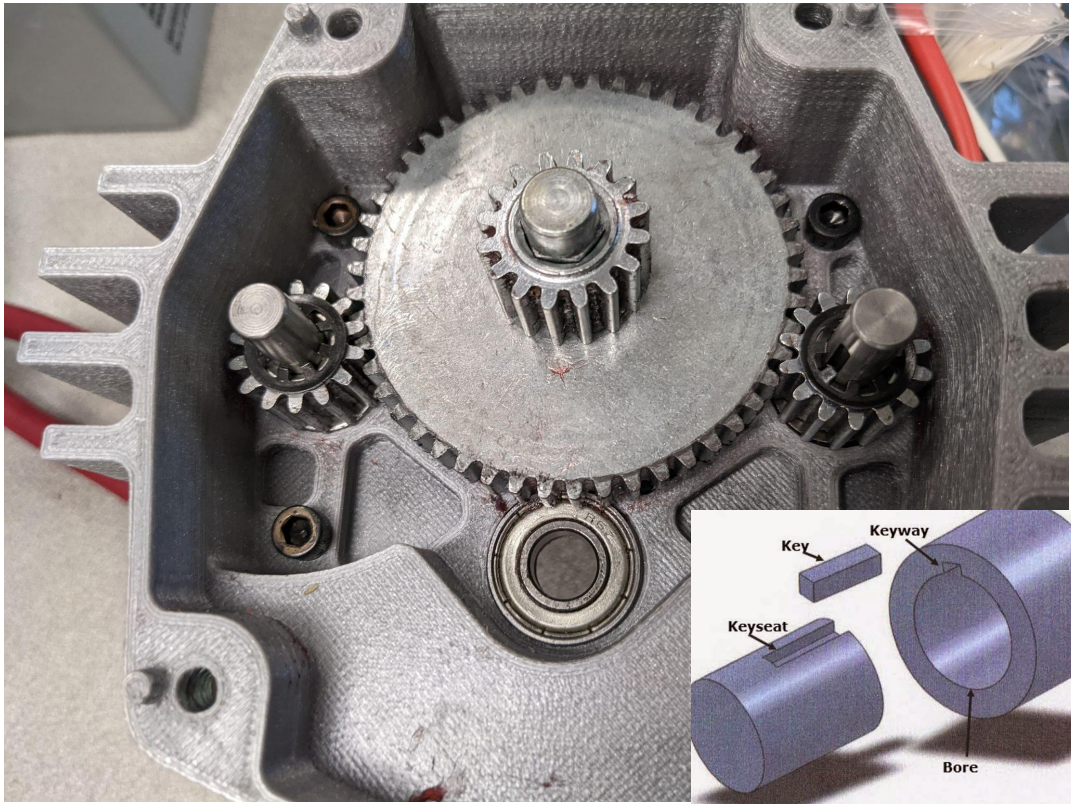


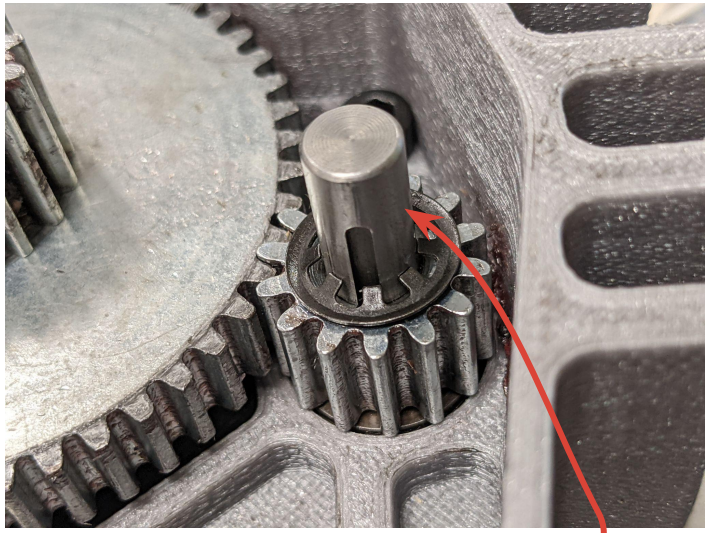
DRIVETRAIN

Drivetrain - 2 Neo MOTORS for Left, 2 NEO MOTORS for Right



Drivetrain - the NEOs connect to this gearbox





Shaft of NEO motor

Needs key to turn gear

Lots of NEO motors this year

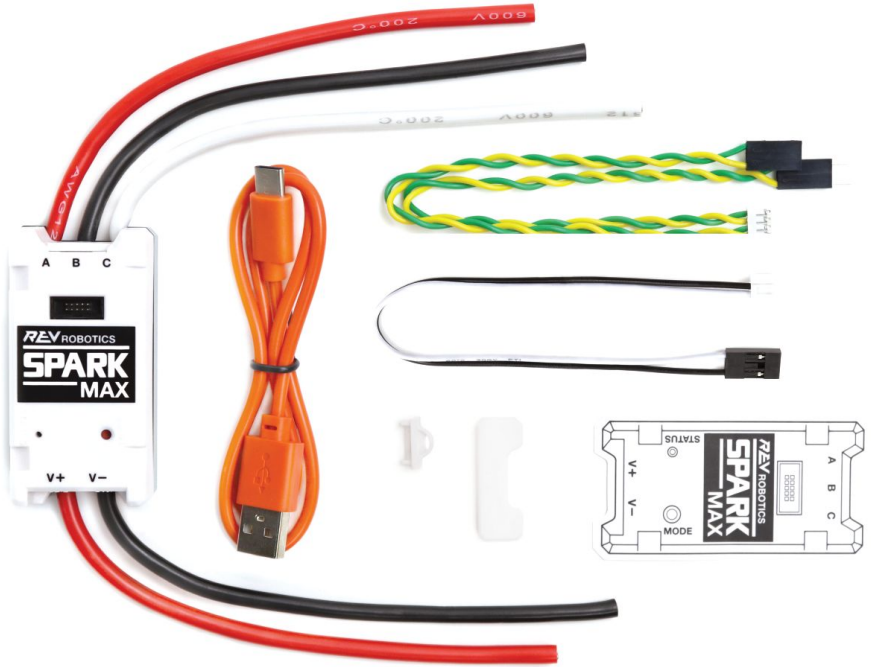
Please do not plug these motors directly to a battery or it will blow URP! Watch out for other teammates who may not be aware and share this information with them. Thanks!

The NEO motors are brushless motors. The NEOs need a Electronic Speed Controller (ESC) to spin.

The CIM motors are dc brush motors and can be connected to a battery.

Here's how brushless motors work:
https://www.youtube.com/watch?v=bCEiOnuODac

The NEO is controlled by a SPARK MAX motor controller

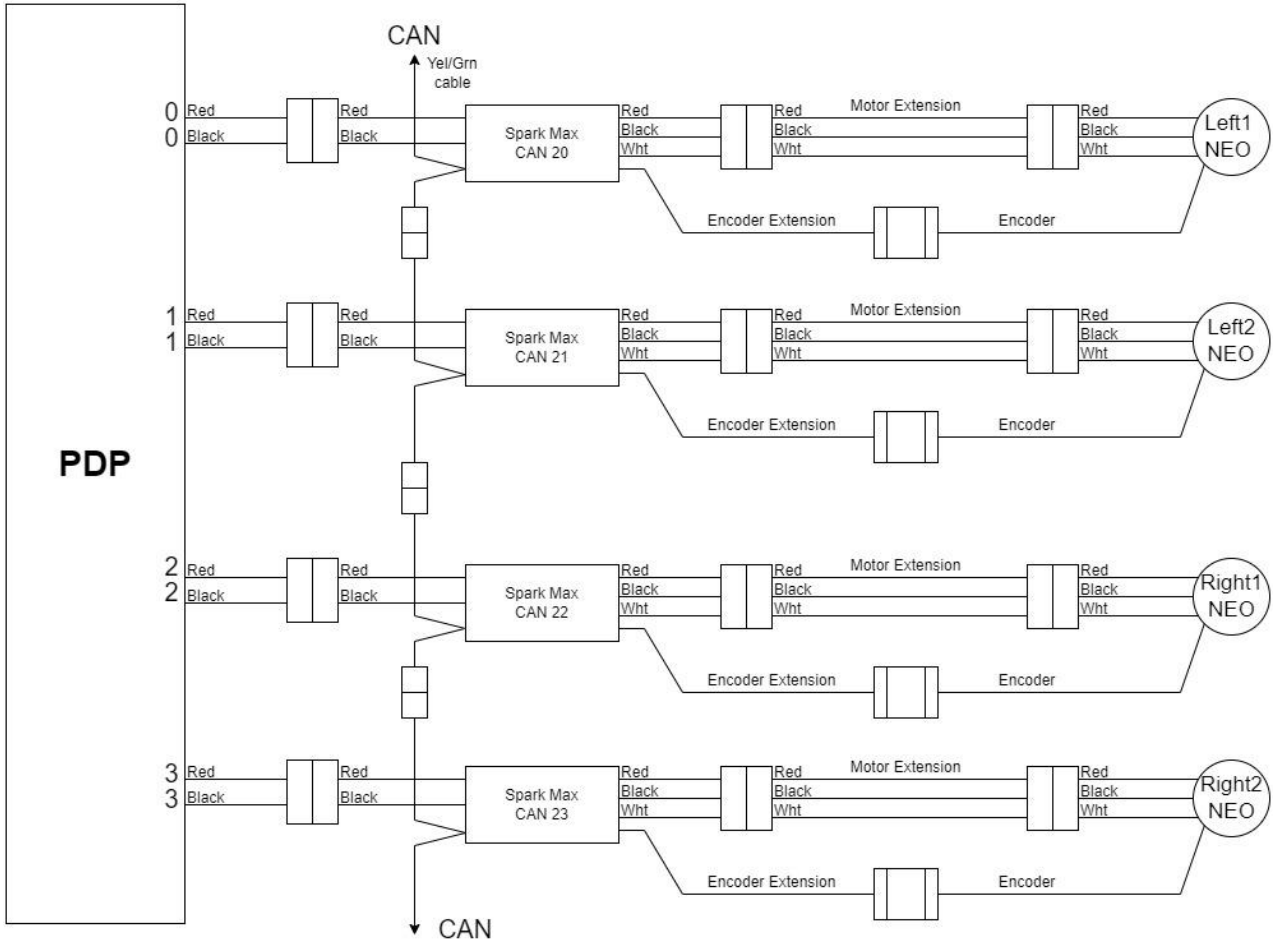


The roboRIO talks to the SPARK MAX using a network called CAN. It's the yellow/green cable that you see.

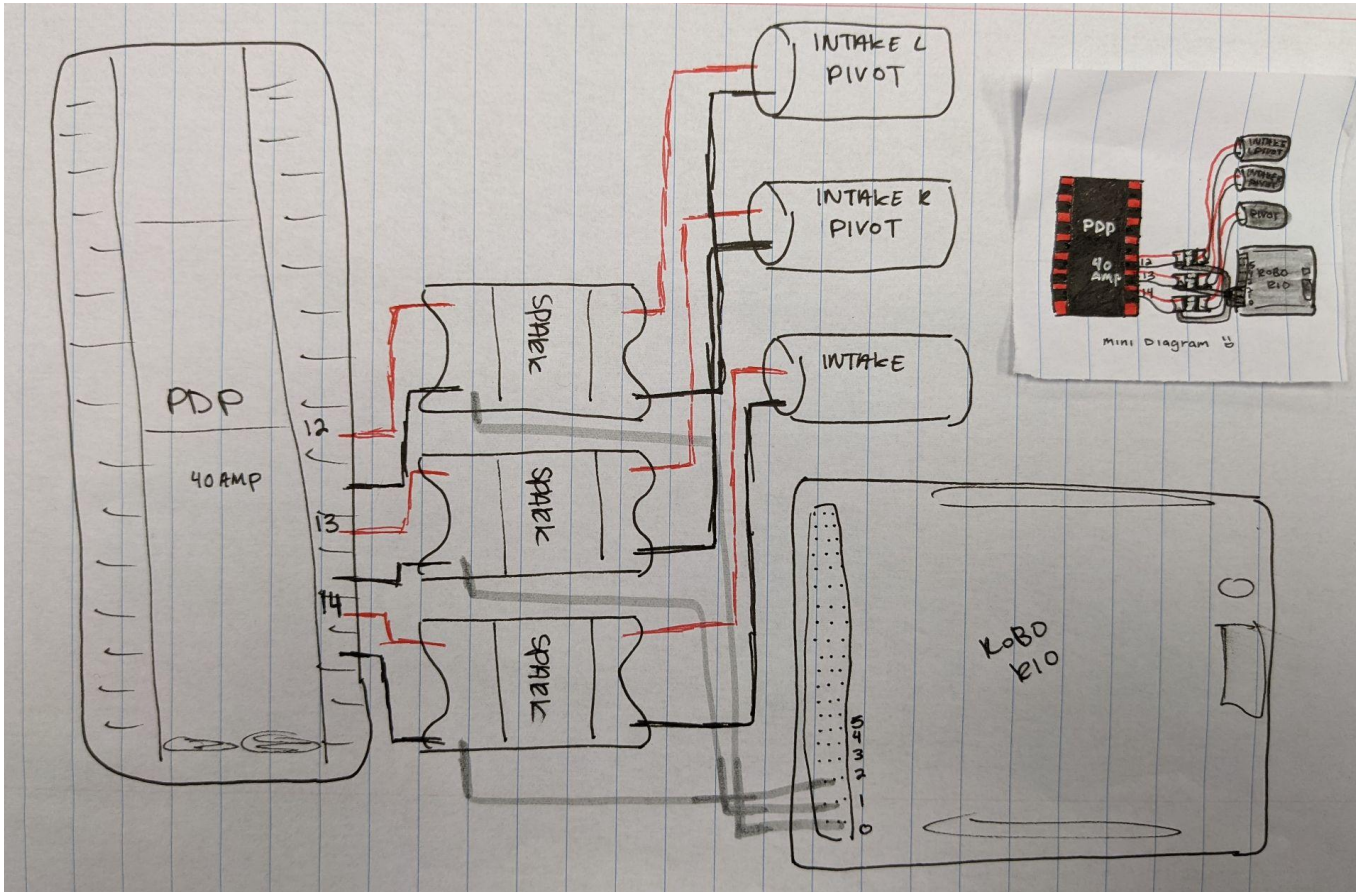
All devices on the CAN NETWORK have an address / ID number. All the devices are listening to messages from the roboRIO, but only respond when the message is directed to its ID.

To configure the SPARK MAX ID and other parameters, use the REV hardware client software.

DriveTrain Wiring



Intake motors - 3 spark controllers using pwm signal



Next, you Need to add vendor library for sparkmax

Online Installation

You can use the online method to install REVLib C++/Java if your development machine is connected to the internet:

1.Open your robot project in VSCode.

2.Click on the WPI icon in the corner to open the WPI Command Pallet.

3.Select **Manage Vendor Libraries**.

4.Select **Install new library (online)**.

5.Enter the following installation URL and press ENTER:
https://software-metadata.revrobotics.com/REVLib.json

Offline Installation

1.Download and unzip the latest REVLib into the *C:\Users\Public\wpilib\2022* directory on Windows and *~/wpilib/2022* directory on Unix-like systems.

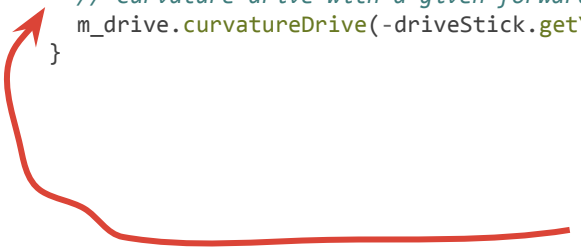
2.Follow the adding an offline-installed library

Now some Sample code for Multi-motor differential drive systems

```
import com.revrobotics.CANSparkMax;
import com.revrobotics.CANSparkMaxLowLevel.MotorType;
import edu.wpi.first.wpilibj.motorcontrol.MotorControllerGroup;


public class Robot {
    CANSparkMax m_left1 = new CANSparkMax(1, MotorType.kBrushless);
    CANSparkMax m_left2 = new CANSparkMax(2, MotorType.kBrushless);
    MotorControllerGroup m_left = new MotorControllerGroup(m_left1, m_left2);

    CANSparkMax m_right1 = new CANSparkMax(3, MotorType.kBrushless);
    CANSparkMax m_right2 = new CANSparkMax(4, MotorType.kBrushless);
    MotorControllerGroup m_left = new MotorControllerGroup(m_right1, m_right2);


    DifferentialDrive m_drive = new DifferentialDrive(m_left, m_right);

    public void robotInit() {
        m_left.setInverted(true); // if you want to invert the entire side
    }
}
```

Do you want to drive a tank, arcade or curvature?

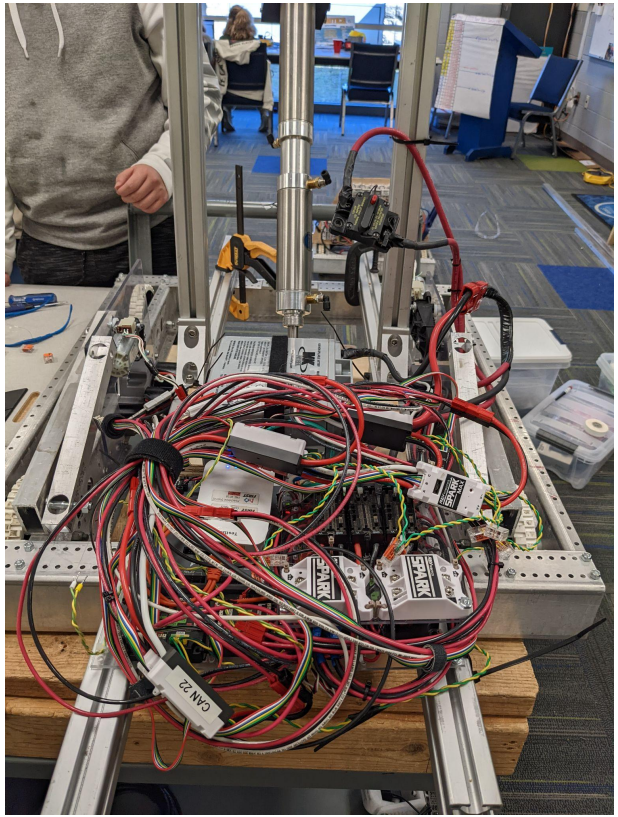
```
public void teleopPeriodic() {
    // Tank drive with a given left and right rates
    m_drive.tankDrive(-leftStick.getY(), -rightStick.getY());

    // Arcade drive with a given forward and turn rate
    m_drive.arcadeDrive(-driveStick.getY(), driveStick.getX());

    // Curvature drive with a given forward and turn rate, as well as a button for turning in-place.
    m_drive.curvatureDrive(-driveStick.getY(), driveStick.getX(), driveStick.getButton(1));
}
```



Let's try all three!!!

Curvature might be cool!

Testing the drivetrain



How do we keep the bot from tipping over?

If you stop too fast, the robot will try to tip over. It might also tip over if you start too fast. One way to reduce the chances of flipping is to accelerate and decelerate at a reasonable amount. Here's one method from Chief Delphi:

timytamy — Jan '15

> **gpetilli:**
>
> We typically do
> x = joyx * abs(joyx); y = joyy * abs(joyy)
>
> this squares the input giving a parabolic response while maintaining sign.

The wpilib actually has this built in. If you call the arcadeDrive (or tankDrive) method, one of the parameters is a boolean "squaredInputs", which does pretty much what your doing.

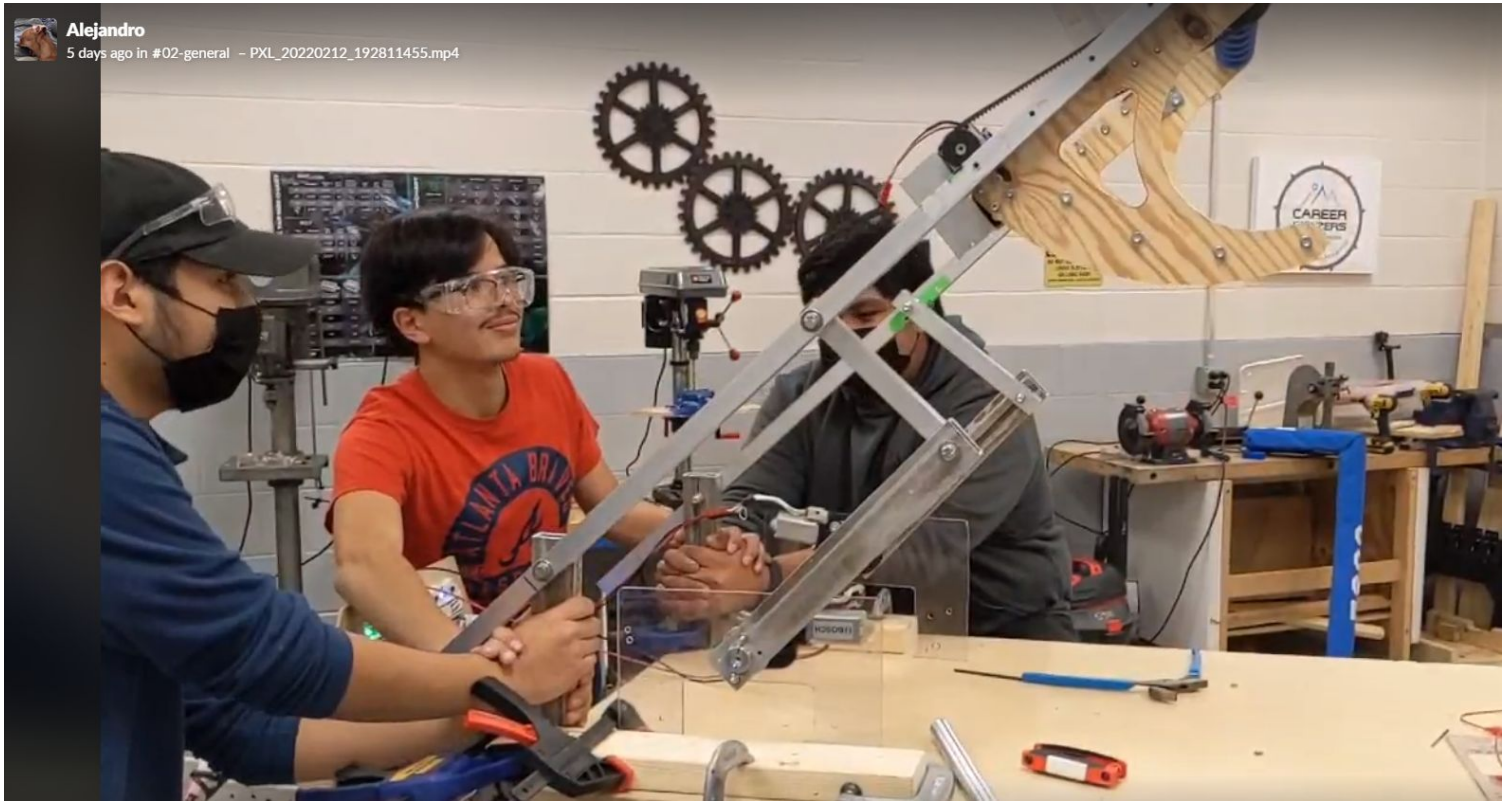
eg:

```
driveSys.arcadeDrive(joyL.getY(), joyR.getX(), true);
```

Another option to try is a Slew Rate Limiter.

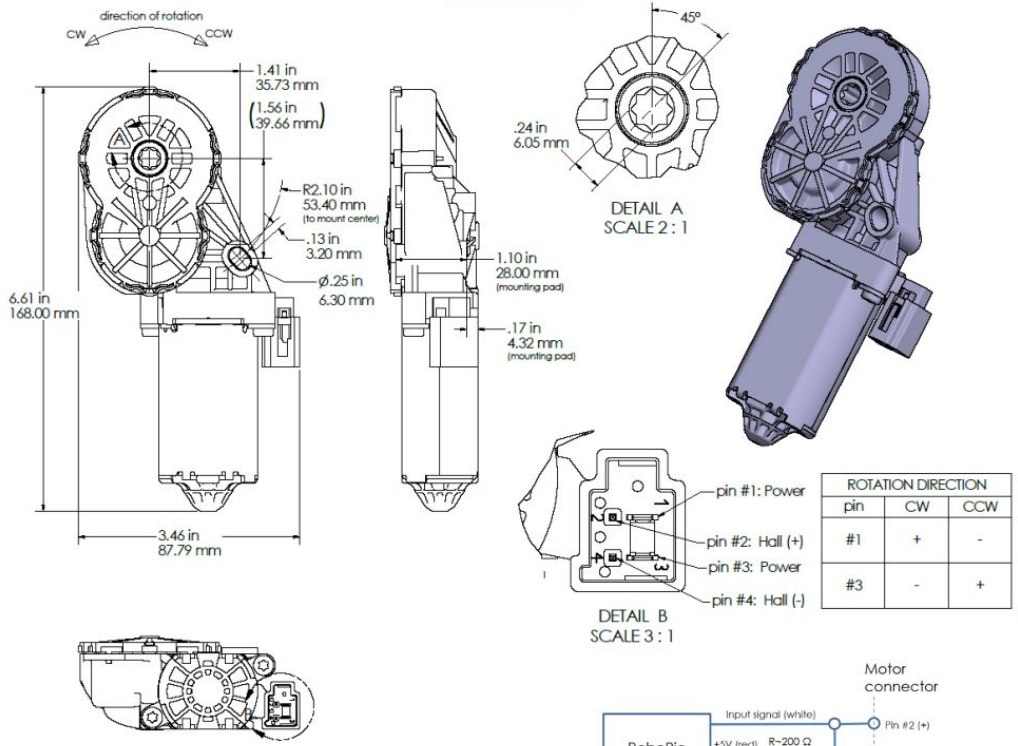
INTAKE

INTAKE + INTAKE ACTUATOR



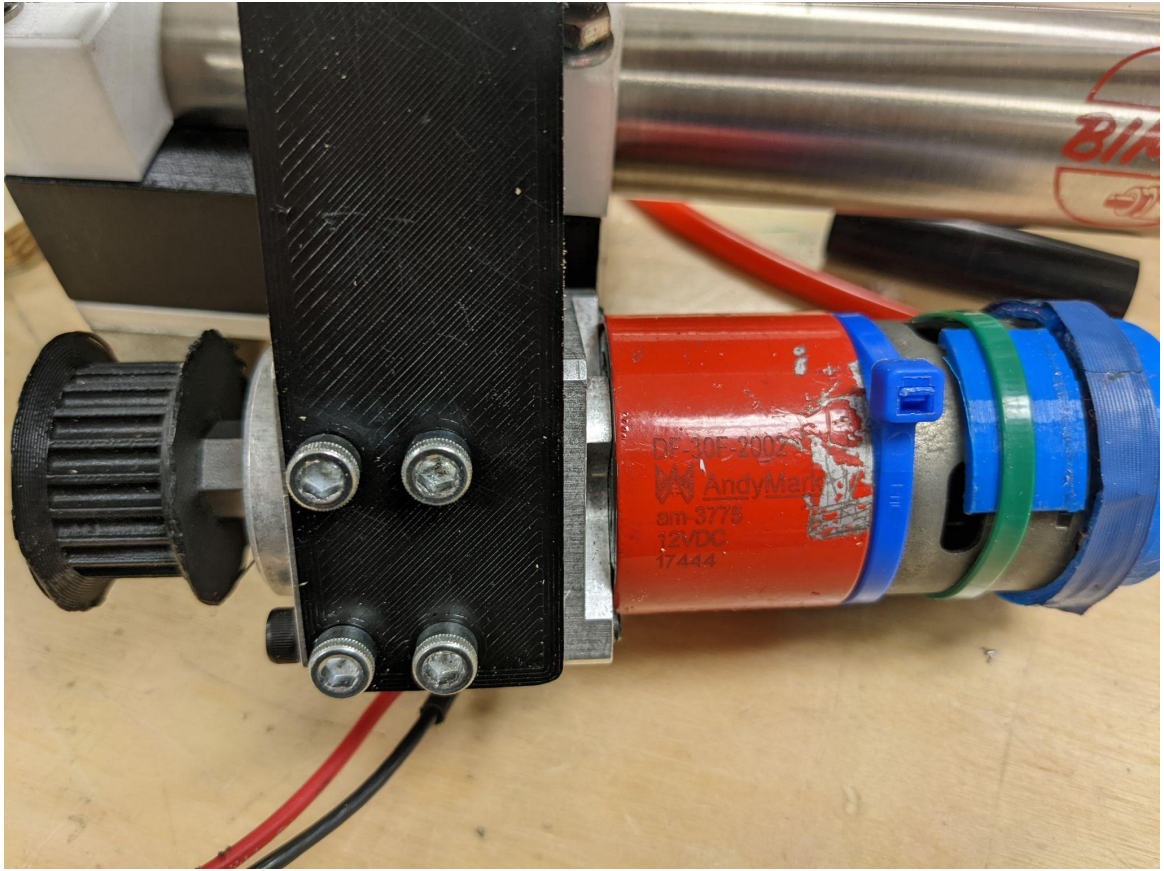
INTAKE ACTUATOR - LOWERS AND LIFTS INTAKE, 2 BOSCH SEAT MOTORS

www.bosch.us



Performance values at 13V, 0Ω wire resistance

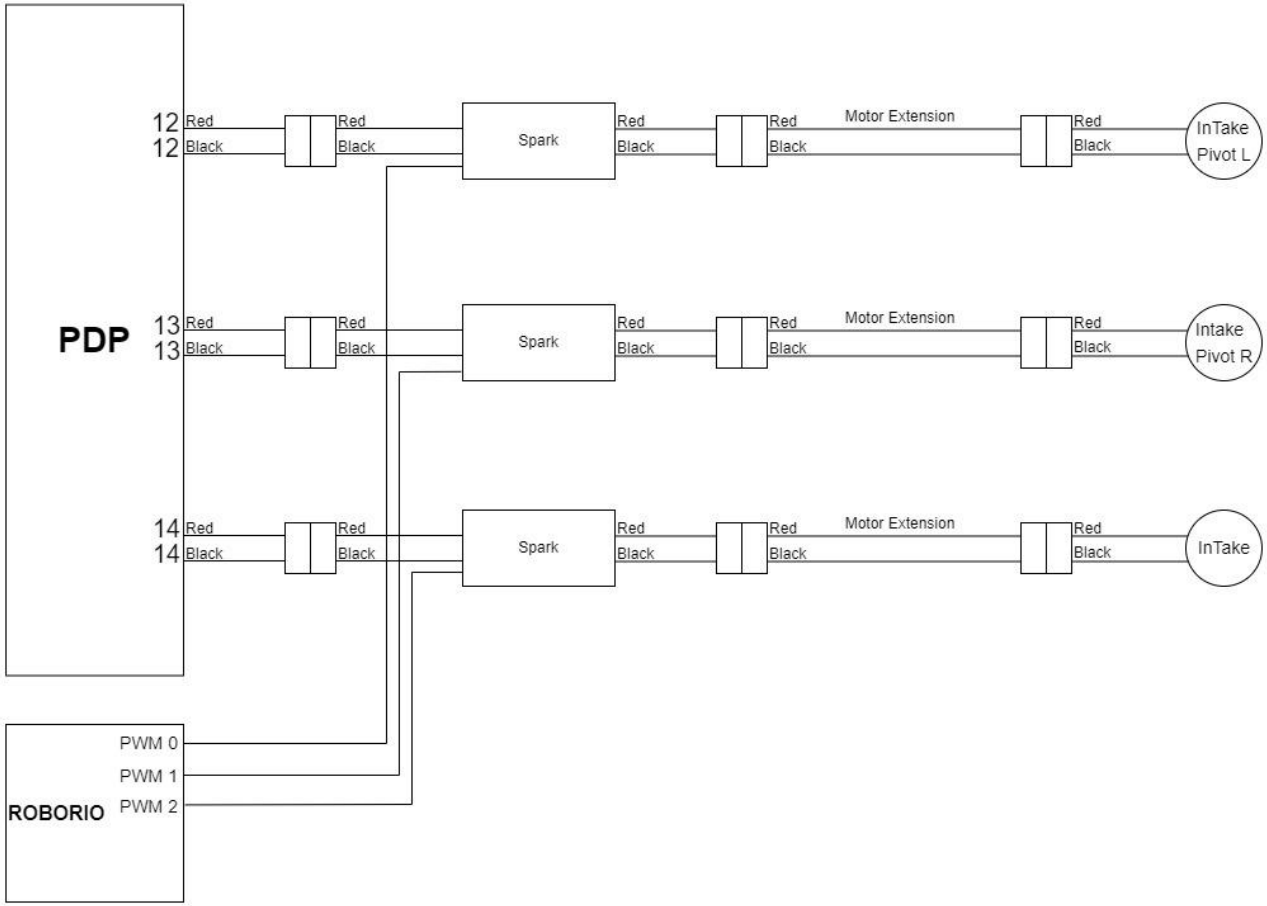
INTAKE REDLINE MOTOR - RETRIEVE AND DELIVER CARGO



Specifications

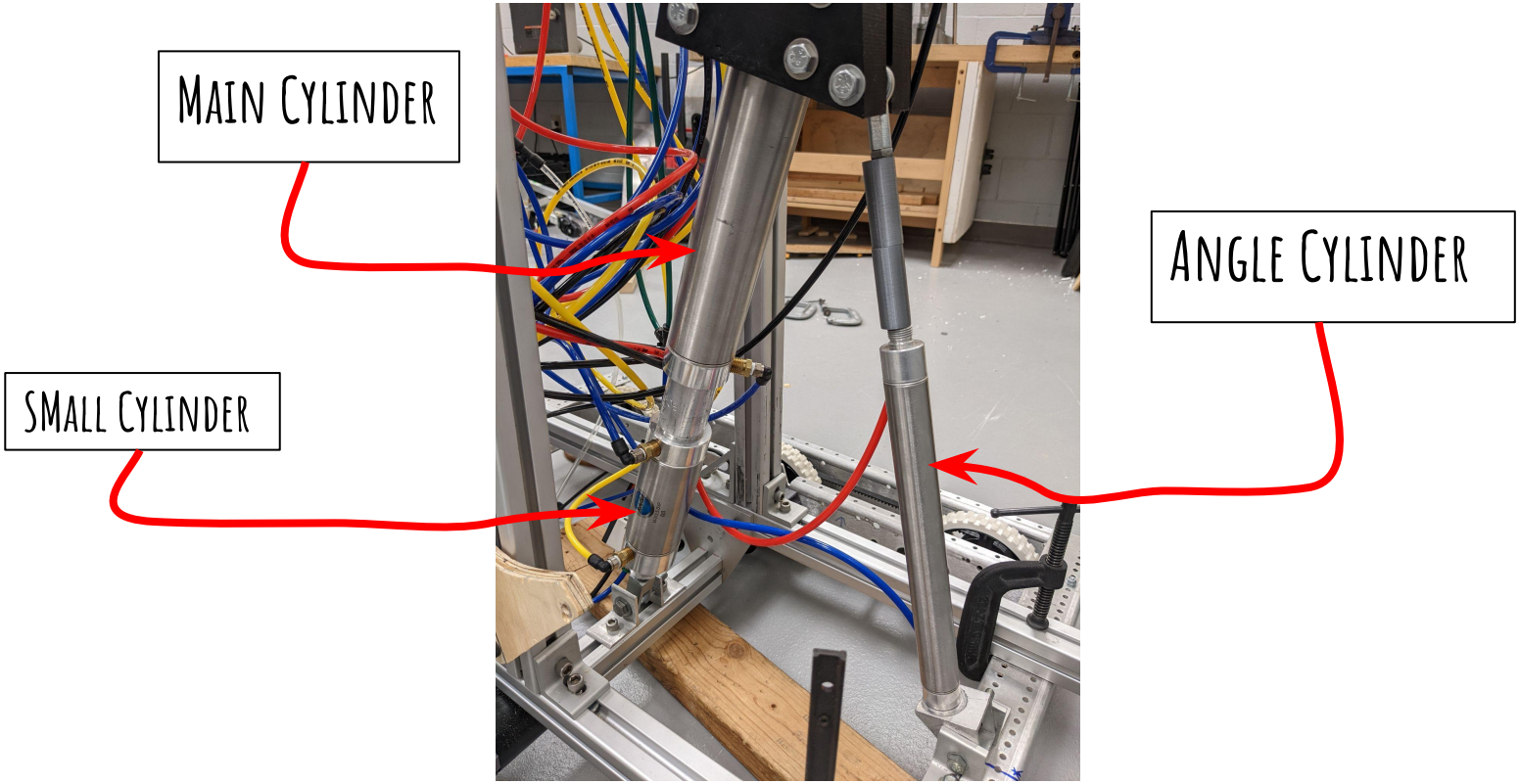
- Bolt Pattern: 2 M4x0.7 Holes on a 29 mm diameter bolt circle
- Diameter: 1.744 in.
- Length: 3.455 in.
- Maximum Power: 380 Watts (+/- 10%)
- No Load Current: 3.8 A
- No Load RPM: 21020 RPM (+/- 10%)
- Output Shaft Diameter: 5 mm
- Output Shaft Length: 0.346 in.
- Stall Current: 130 A
- Stall Torque: 0.7 N-m (6.2 in-lbs) (+/- 10%)
- Voltage: 12 VDC
- Weight: 0.806 lb

InTake Wiring



Climb system

CLIMB CYLINDERS



Joystick controls

Driver controls

Driver and pit checklist

Before Match

- Change out Battery (Use Battery Beak check)
- Pre-Charge Air Tanks
- Change bumpers to correct Color
- Joystick connected to correct Port
- Set Robot Preferences to color Color
- Check Robot and make sure nothing falls off
- Have Flag ready
- Send human player to queue with Flag
- **(NO PHONES ALLOWED)**
- Meet with Alliance Teams to discuss strategy

AFTER MATCH

Review with Programming Crew

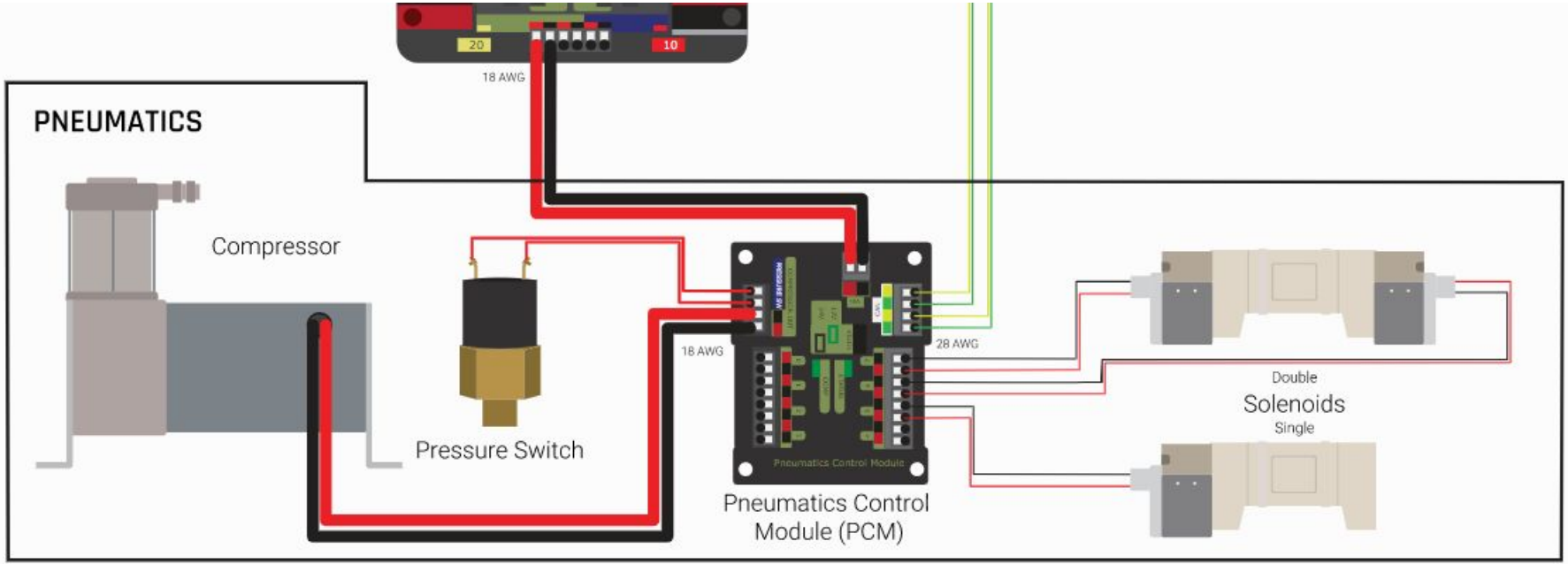
Discuss what went wrong and what went right with Bot

What's pNeu, Matics?

Most of what you need to know to use pneumatics on your bot!

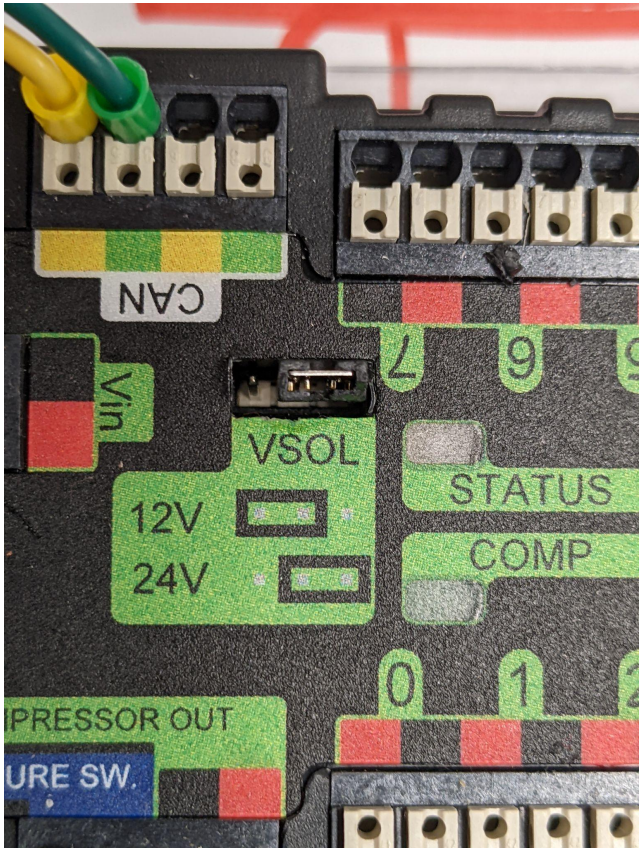
Cylinder and solenoid valve

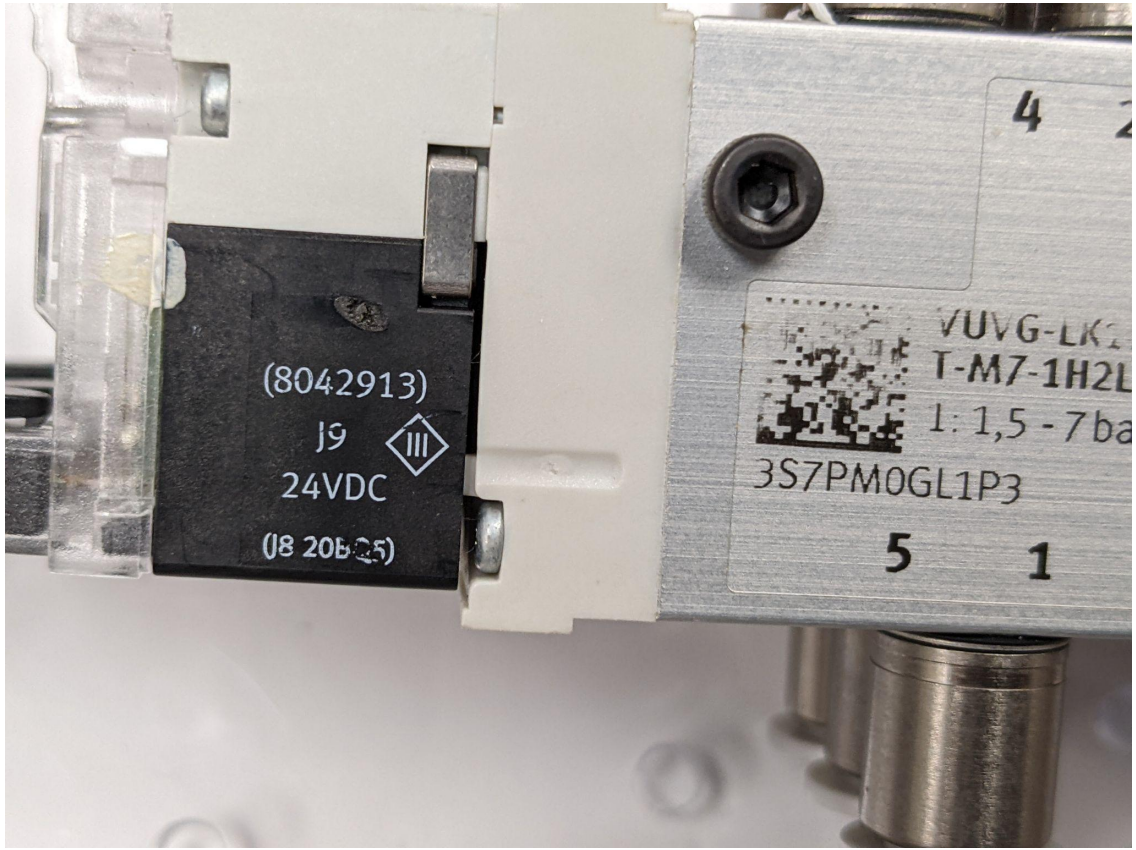
Pneumatics control module (PCM)



The first PCM on your robot can be wired from the PDP VRM/PCM connectors on the end of the PDP. The PCM is connected to the roboRIO via CAN and can be placed anywhere in the middle of the CAN chain (or on the end with a custom terminator). For more details on wiring a single PCM, see Pneumatics Control Module Power (Optional)

ARE YOU USING 12V SOLENOIDS OR 24V SOLENOIDS?





Let's dissect some code for pneumatics control

First add libraries for solenoids and pneumatic controller

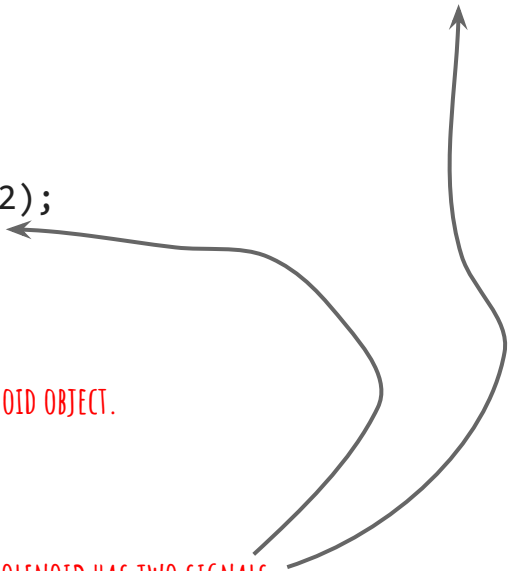
```
import edu.wpi.first.wpilibj.DoubleSolenoid;
import edu.wpi.first.wpilibj.Joystick;
import edu.wpi.first.wpilibj.PneumaticsModuleType;
import edu.wpi.first.wpilibj.Solenoid;
import edu.wpi.first.wpilibj.TimedRobot;
```

The doublesolenoid library is for valves that require two signals. One signal to extend the cylinder and another signal to retract the cylinder.

The solenoid library is for valves that require only one signal to extend and retract the cylinder. Usually, there is as spring that physically returns the valve to the off state when the signal is off.

Creating a solenoid and a doublesolenoid object

```
// Solenoid corresponds to a single solenoid.
private final Solenoid m_solenoid = new Solenoid(PneumaticsModuleType.CTREPCM, 0);

// DoubleSolenoid corresponds to a double solenoid.
private final DoubleSolenoid m_doubleSolenoid =
    new DoubleSolenoid(PneumaticsModuleType.CTREPCM, 1, 2);
```



The word new says that we are creating an object. In this case it is a solenoid object and a doublesolenoid object.
We name the solenoid object m_solenoid.
We name the doublesolenoid object m_doublesolenoid

these numbers are for the channels that are used to control these solenoids. Notice that the doublesolenoid has two signals.

CONTROLLING THE VALVES IN TELEOP MODE

```
@Override
 public void teleopPeriodic() {
   /*
    * The output of GetRawButton is true/false depending on whether
    * the button is pressed; Set takes a boolean for whether
    * to use the default (false) channel or the other (true).
    */
   m_solenoid.set(m_stick.getRawButton(kSolenoidButton));

   /*
    * In order to set the double solenoid, if just one button
    * is pressed, set the solenoid to correspond to that button.
    * If both are pressed, set the solenoid will be set to Forwards.
    */
   if (m_stick.getRawButton(kDoubleSolenoidForward)) {
     m_doubleSolenoid.set(DoubleSolenoid.Value.kForward);
   } else if (m_stick.getRawButton(kDoubleSolenoidReverse)) {
     m_doubleSolenoid.set(DoubleSolenoid.Value.kReverse);
   }
 }
```

THIS ACTIVATES THE SOLENOID VALVE IF BUTTON IS PRESSED.

ONE BUTTON EXTENDS ANOTHER BUTTON RETRACTS THE DOUBLESOLENOID.

Resources on using Pneumatics

Great video on Pneumatics

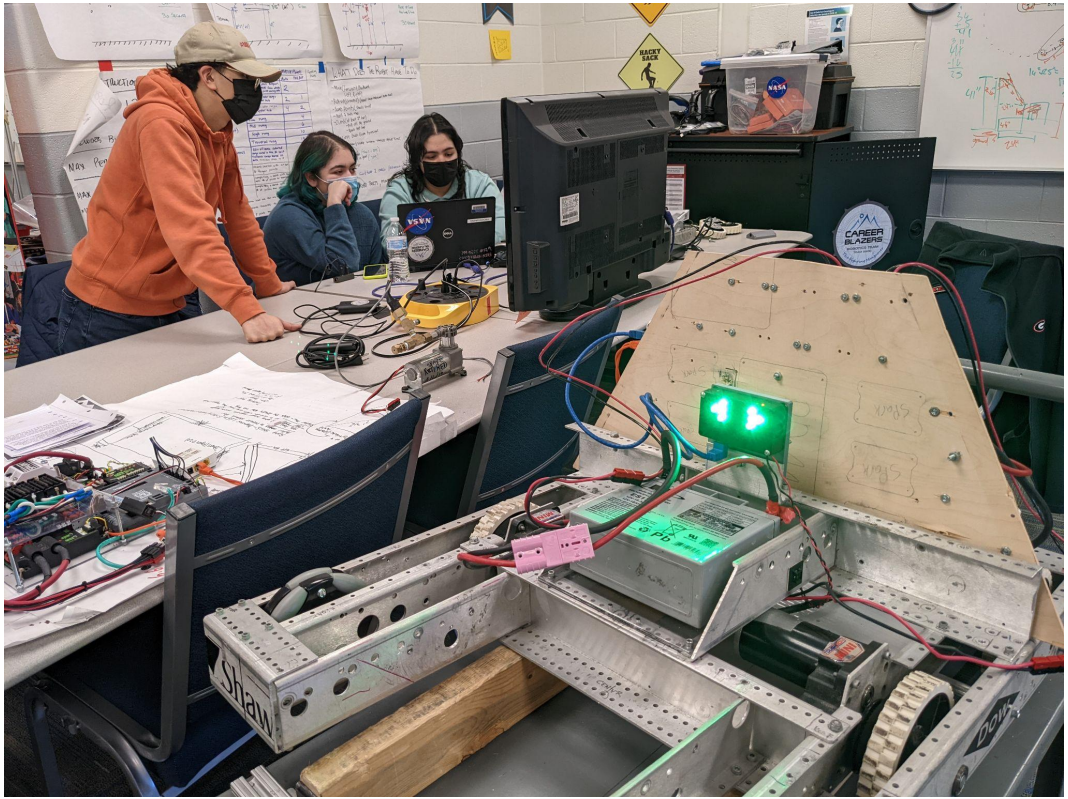
WPILib section on Pneumatics

Camera

Want some lime wit dat?

Stuff you might want to know about the limelight

Limelight camera



How do I select a different pipeline for the limelight?

First, What's a pipeline and what can I do with it?

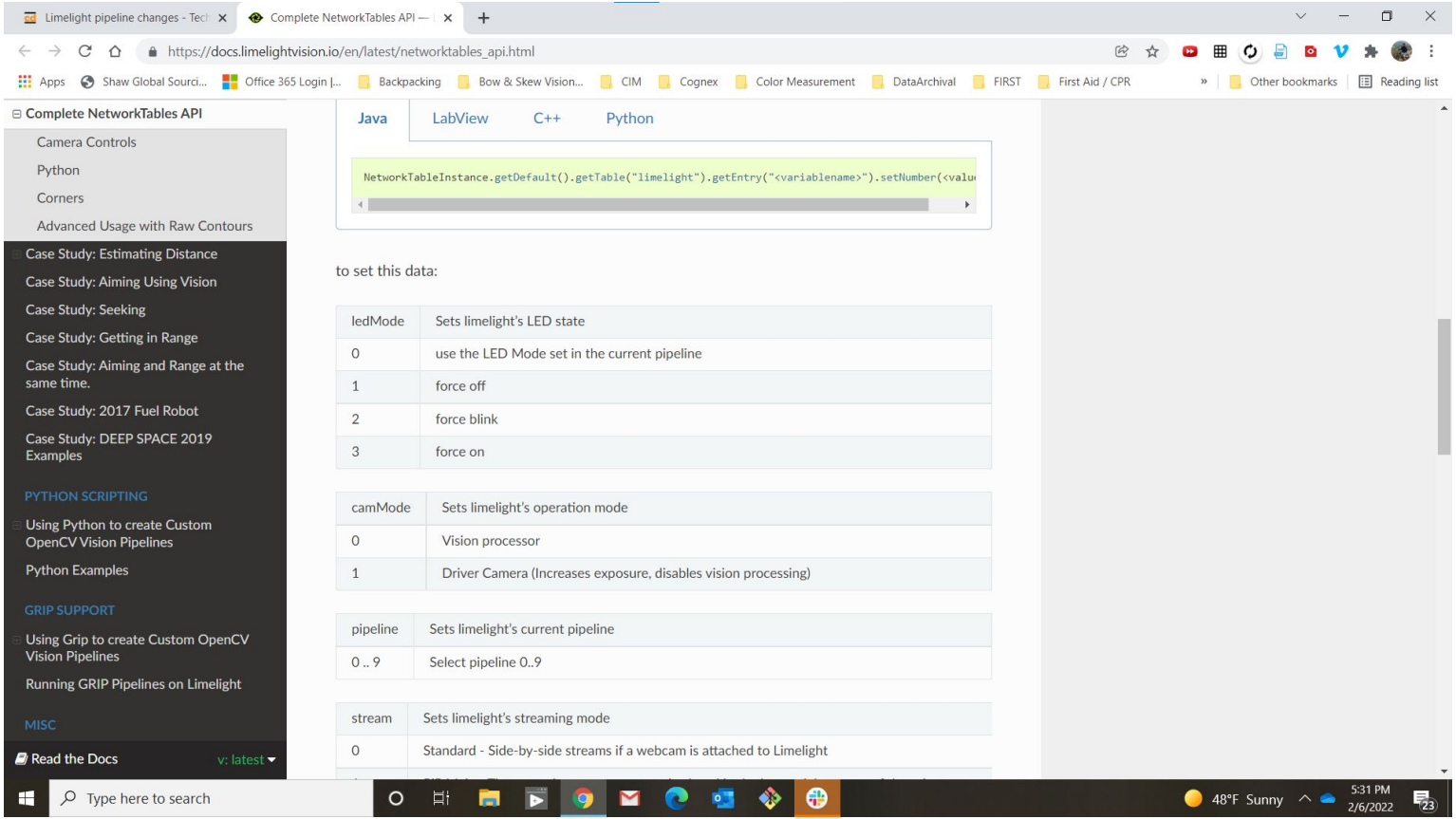
The limelight can have up to 10 pipelines (0 to 9)

Think of a pipeline as a program or configuration of the camera to look for a certain target.

For rapid-react, we could have a pipeline for detecting a red cargo, a pipeline for blue cargo, a pipeline to look for upper hub goal, etc.

How do we select which pipeline to run on the camera?

How do we select which pipeline to run on the limelight?



Using the networktableinstance function

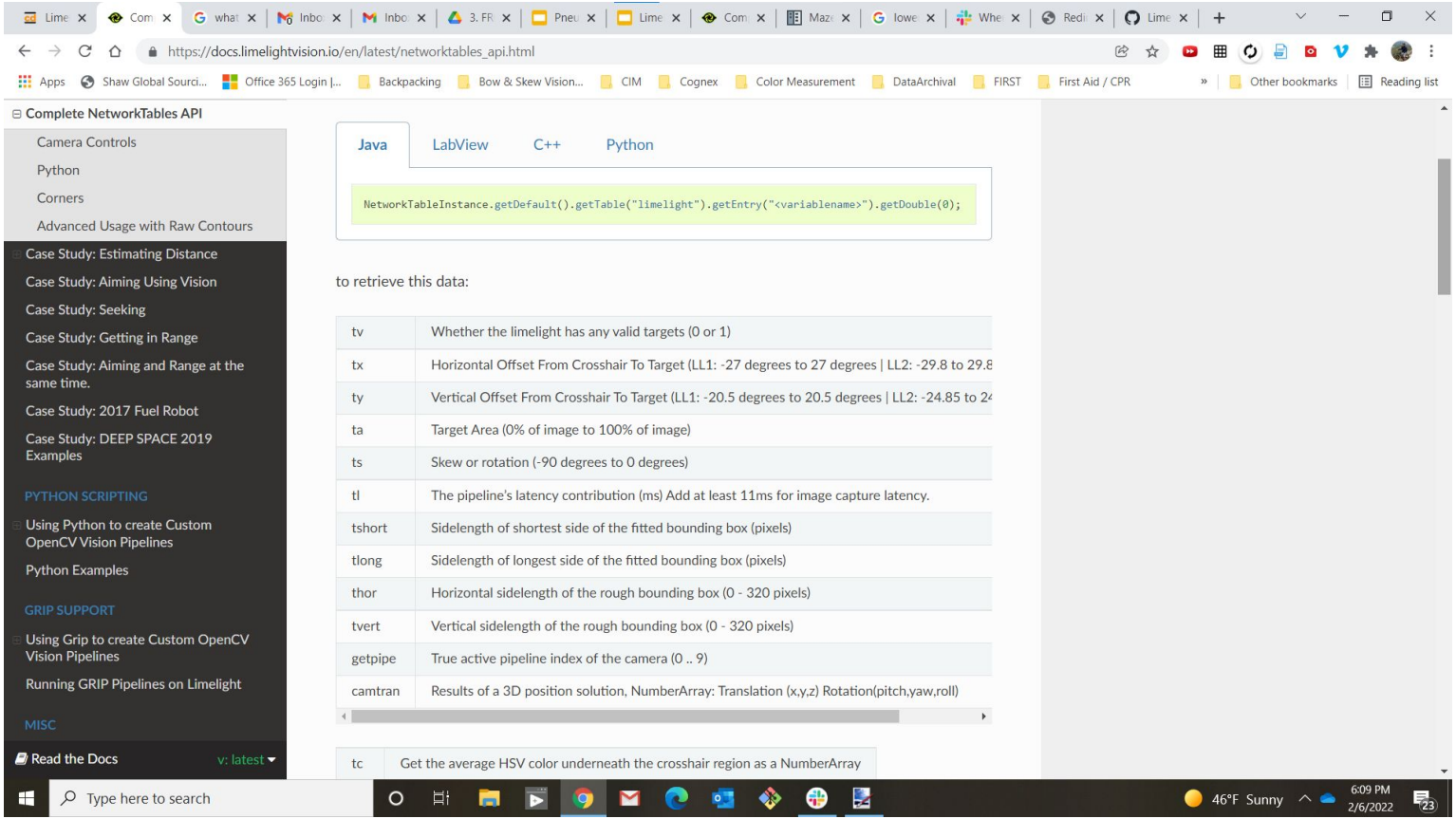
```
NetworkTableInstance.getDefault().getTable("limelight").getEntry("<variablename>").setNumber(<value>);
```

```
NetworkTableInstance.getDefault().getTable("limelight").getEntry("pipeline").setNumber(0); //select pipeline 0

NetworkTableInstance.getDefault().getTable("limelight").getEntry("pipeline").setNumber(1); //select pipeline 1

NetworkTableInstance.getDefault().getTable("limelight").getEntry("ledMode").setNumber(1);  //turn off LEDs

NetworkTableInstance.getDefault().getTable("limelight").getEntry("ledMode").setNumber(3);  //turn on LEDs
```

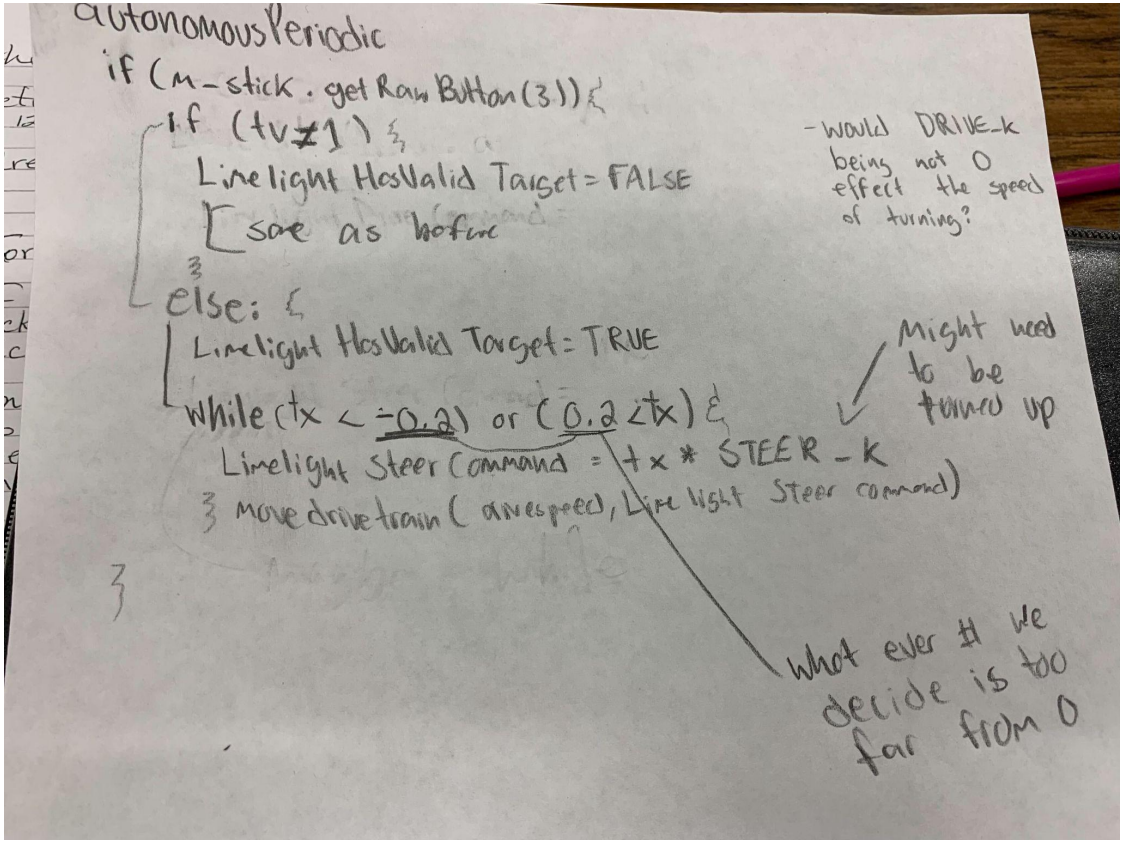
See the pattern?

What values can we get from the limelight?

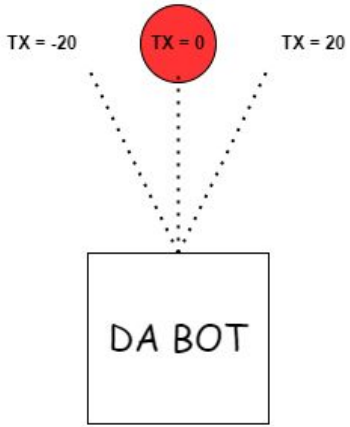
LIMELIGHT DATA



Now, write down how you think it should work. Then, start Sperimenting!!!



Limelight TX data



If TX is zero-ish, do not move DA BOT!

What is zero-ish? whatever you like it to be!
We can start with +- 1. TX has to be greater than 1 or less than -1 for motor to steer to cargo

What's the minimum voltage needed to move the motors?
I think this is around 0.35 Volts. So we need the speed voltage to be greater than 0.35 Volts or less than -0.35 Volts. Remember the speed voltage can be from -1.0 to +1.0

If TX is maxed out at -20 or +20, how fast do we want the motor to run? Let's start with 0.5 Volts

Vmin = 0.35
Vmax = 0.50
TXmax = 20
TXmin = 1

Vdelta = (Vmax - Vmin) / (TXmax-TXmin)
= 0.008

```
If TX >= 1.0
  SteerCommand = 0.35 + TX * Vdelta
ELSE IF TX <=-1.0
  SteerCommand = -0.35 + TX * Vdelta
ELSE
  SteerCommand = 0
```

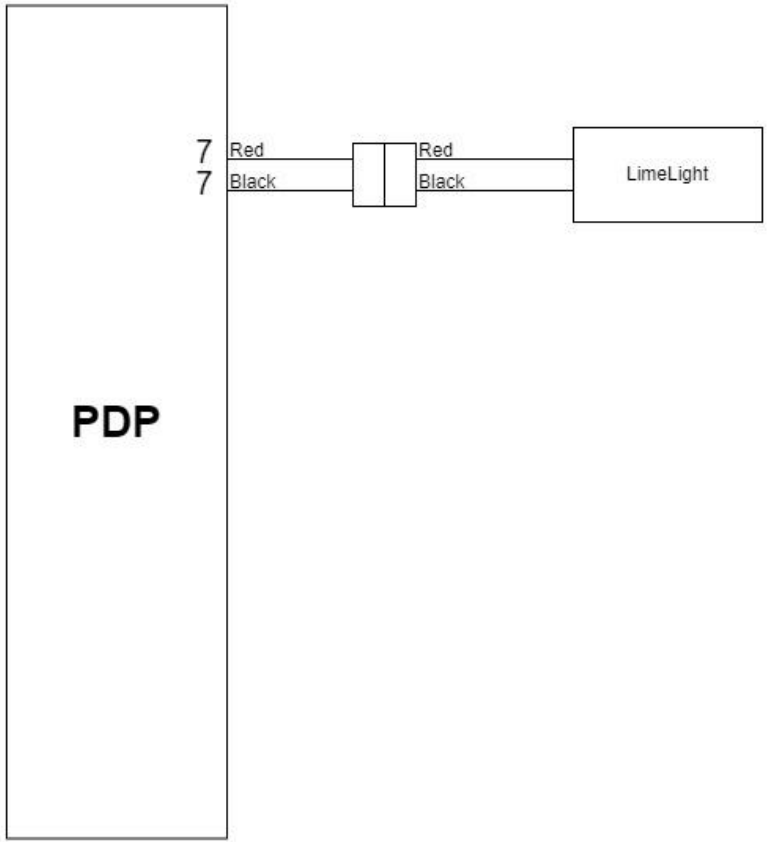
Check the math:
When TX = 1.0,
SteerCommand = 0.358

When TX = 20.0
SteerCommand = 0.51

When TX = -1.0,
SteerCommand = -0.358

When TX = -20.0
SteerCommand = -0.51

LimeLight Wiring



Resources on using the limelight

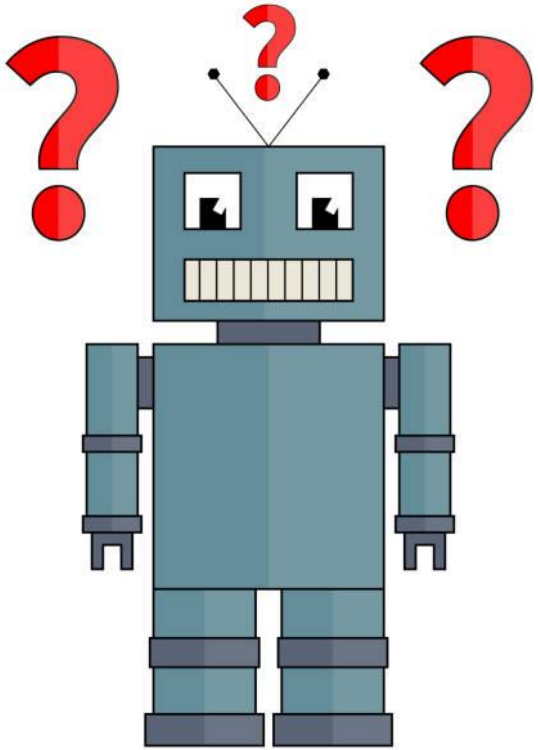
Limelight Functions

Limelight video from team 3244

What is your preference?

Learn about robot preferences

TWEAK SETTINGS ON THE ROBOT WITHOUT REWRITING CODE



Why use robot preferences?

set maximum speed for drivetrain

tell the robot where it is placed at the start of the match - run autonomous sequence based on start location

tell the robot what color alliance it is for the match - use to select pipeline for limelight

Instead of having to recompile / deploy new code everytime you change a setting, you can store in robot preferences

Retains preferences when turning off power to the robot - you don't have to re-enter the settings.

Use SmartDashboard to create robot preferences

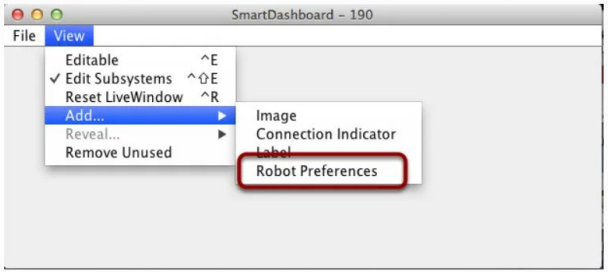
Robot Preferences - Google Slide | Setting Robot Preferences — FIR | Setting Robot Preferences — FIR

https://docs.wpilib.org/en/stable/docs/software/dashboards/smartdashboard/setting-robot-preferences-from-smartdashboard.html

Apps | Shaw Global Sourci... | Office 365 Login |... | Backpacking | Bow & Skew Vision... | CIM | Cognex | Color Measurement | DataArchival | FIRST | First Aid / CPR | Other bookmarks | Reading list

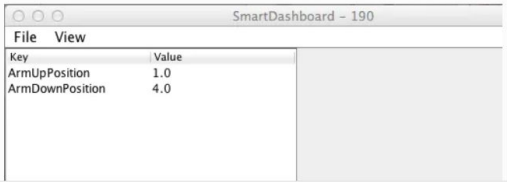
Read the Docs v: stable

Displaying Preferences in SmartDashboard



In the SmartDashboard, the Preferences display can be added to the display revealing the contents of the preferences file stored in the roboRIO flash memory.

Viewing and Editing Preferences



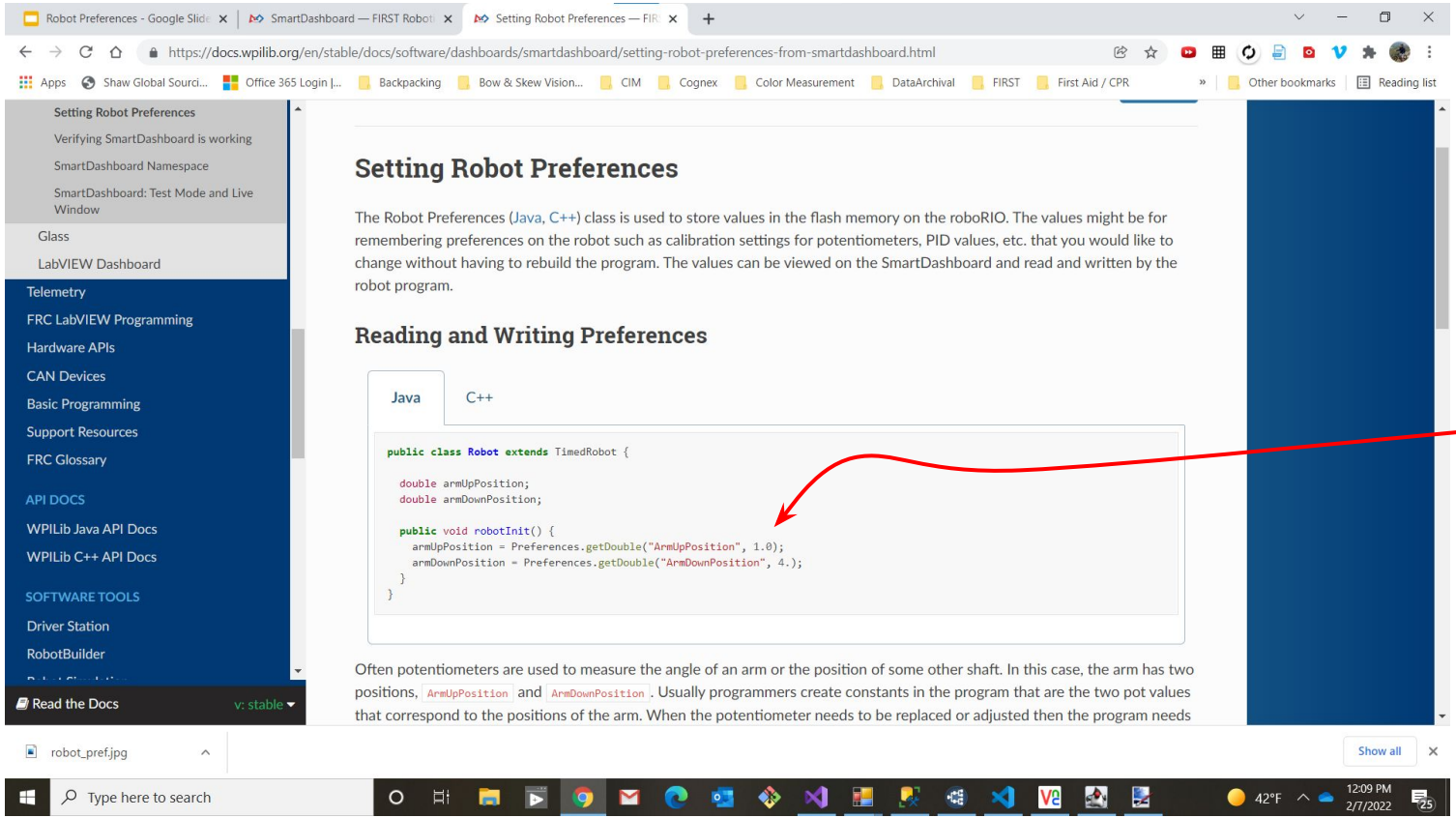
robot_pref.jpg

Show all

Type here to search

42°F 12:03 PM 2/7/2022

READ ROBOT PREFERENCES



DEFAULT VALUE IF PREFERENCE IS NOT FOUND

Resources on using robot preferences

Setting robot preferences

A-Ton-of-Moose

Learn how to use enum and switch in Java

ENUM - PROVIDES A DESCRIPTIVE NAME TO CONSTANT VALUES

```
enum AutoStates
{
  wait_for_activation,
  initialize,
  wait_for_turn,
  wait_for_elapsetime
};
```

So what?

Instead of AutoStates equalling 0, 1, 2, 3

The AutoStates values can be given a descriptive name.

In this example,

AutoState.wait_for_activation has a value of 0.

AutoState.initialize has a value of 1.

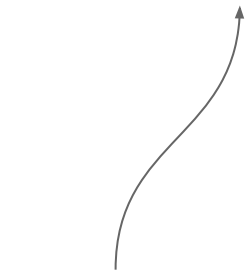
AutoState.wait_for_turn has a value of 2..

AutoState.wait_for_elapsetime has a value of 3.

Let's see how this makes ATonOfMoose code more readable!

Create a variable for The state of the Autonomous code.

```
private AutoStates atonomous_state;
```



The type of data that the variable we will be storing

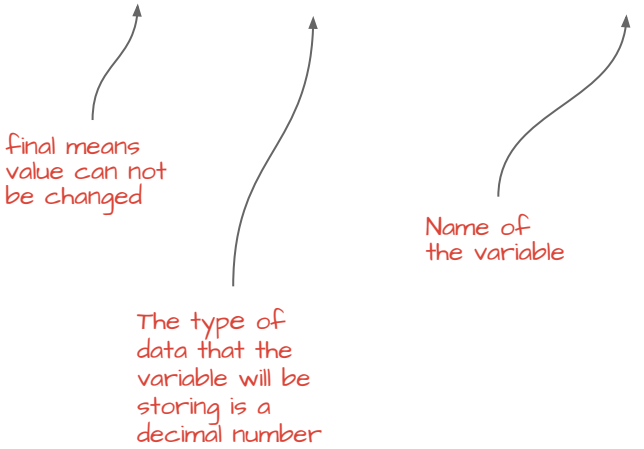
Name of the variable

The variable autonomous_state is of the type enum AutoStates that we created in the previous slide.

What is a variable? A variable is a placeholder. A location in the roboRIO's memory where we want to store a value that we will need to read or change later in the program.

Create a constant value for timing a drivetrain move

```
final double MoveCountDown = 2.0;
```



The variable called MoveCountDown is of type double which is a decimal number.

What is final mean? It says that this value cannot be changed.

We will use this value to tell us how long to move the robot.

In this example, the 2.0 is the number of seconds for our move.

Set the atonomous _state variable for the start of the match

```
public void StartGameClock()
{
  atonomous_state = AutoStates.initialize;
}
```

When the game starts, the robot will be placed in ATonOfMoose mode for 15seconds.

The variable atonomous_state is set to the initialize state.

During Autonomous - This routine will be called every 20milliseconds.

```
public void autonomousPeriodic()
{
  switch( atonomous_state )
  {
    case wait_for_activation:
      break;

    case initialize:
      aclock.reset();                                   // reset timer
      aclock.start();                                   // start timing
      Atonomous_state = AutoStates.wait_for_turn;
      break;

    case wait_for_turn:
      if( aclock.get() > 0.8)                           // wait 0.8 seconds
        MoveDriveTrain( 0, 0);                          // turn off both motors
        atonomous_stat = AutoStates.wait_for_elapsetime;
      else
        MoveDriveTrain( 0, -0.5);                       // CCW left motor off and right motor at half speed
      break;

    case wait_for_elapsetime:
      if( aclock.get() > MoveCountDown )                // wait till time is greater than MoveCountDown
      {
        MotorsOff();                                    // Turn off the drivetrain motors
        Atonomous_state = AutoStates.wait_for_activation;   // We are done, just wait now for teleOp mode
      }
      else
        MoveDriveTrain( -0.5, 0 );                      // CW left motor at half speed and right motor off
      break;

    default:
      atonomous_state = AutoStates.wait_for_activation;     // if atonomous_state is unknown, set to wait_for_activation
      break;
  }
}
```

Show a little class!

Basic class definition - a class is a like a blueprint describing an object

```
// Name of the package must be same as the directory as our project folder


package frc.robot;  //We need this to say this class is part of our frc robot code

public class MyClass
{
  MyClass(){     //This is a constructor - when you use new keyword to create object

  }

  public void func_one(String s)
  {
  }


  public void another_func()
  {
  }
}
```

MORE INFO ON JAVA CLASS HERE!!!

Todo list

To do list

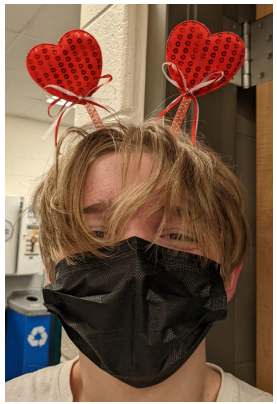
- Test drivetrain
- Test climb code
- Wire up solenoid valves, compressor, pressure sensor
- Finish autonomous code and test
- Build controls enclosure
- Find home for controls box enclosure
- Find home for limelight, compressor and battery
- Work with drivers on how they want to control da bot
- Setup robot preferences for alliance color
- Separate code for different parts of the bot - in java we call these classes

Da programming and electrical crew

Rapid react programming / electrical crew

Alejandro miranda Jr
Isaac jones
Vannessa garcia
Austen Melendez
Crystal melendez
Giselle rodriguez
Zero Hardy
Wesley santana
Mia Mendoza
Jazmine ayabar
Ivan franco
Hugo lozano

WE



Keep experimenting, keep Learning, and
Most of all, Have fun!!!!

Robotics Archives

Programmers for 2021 Infinite Recharge 2

Isaac Jones

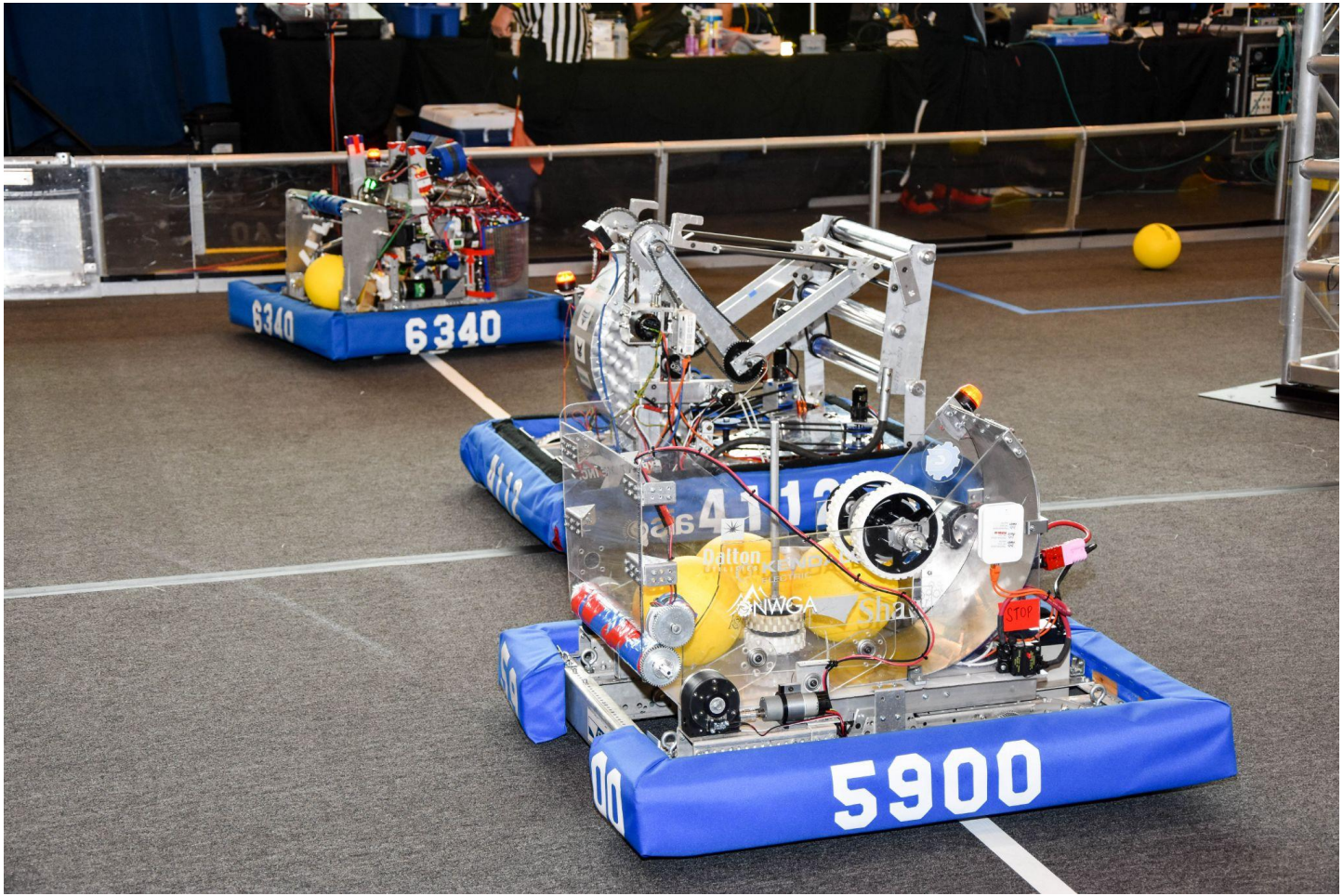
Alejandro Miranda

Dylan McClure

Caleb Henricks

(Java)

Bot: Chonk 2.0 and Wyoming


6340
6340
5900
NWGA
STOP

Programmers for 2020 Infinite Recharge

Troy Johnson

Alejandro Miranda

Jazmine Colindres

Trevor Silvers

Gage Godfrey

(Java)

Bot: No Quema Cuh


7883
6905
6905
5900
5900
7451

Programmers for 2019 Deep Space

Gage Godfrey

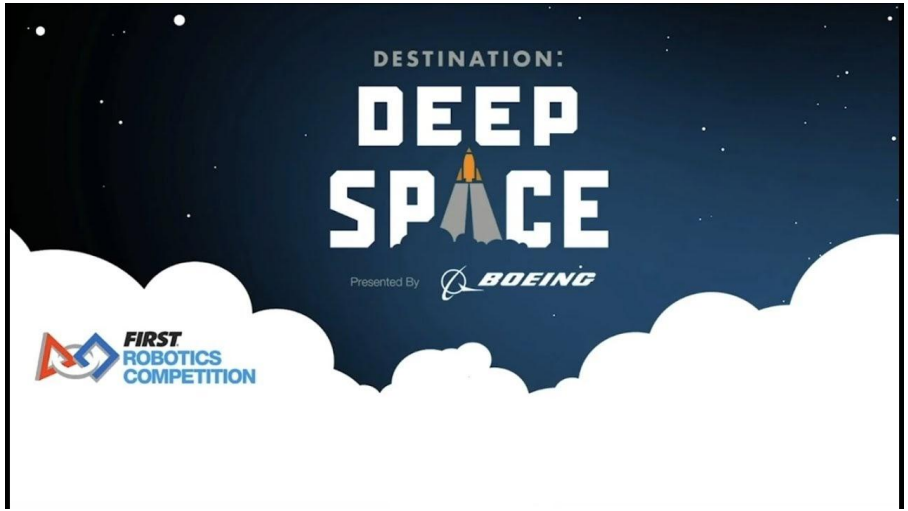
Conner Henderson

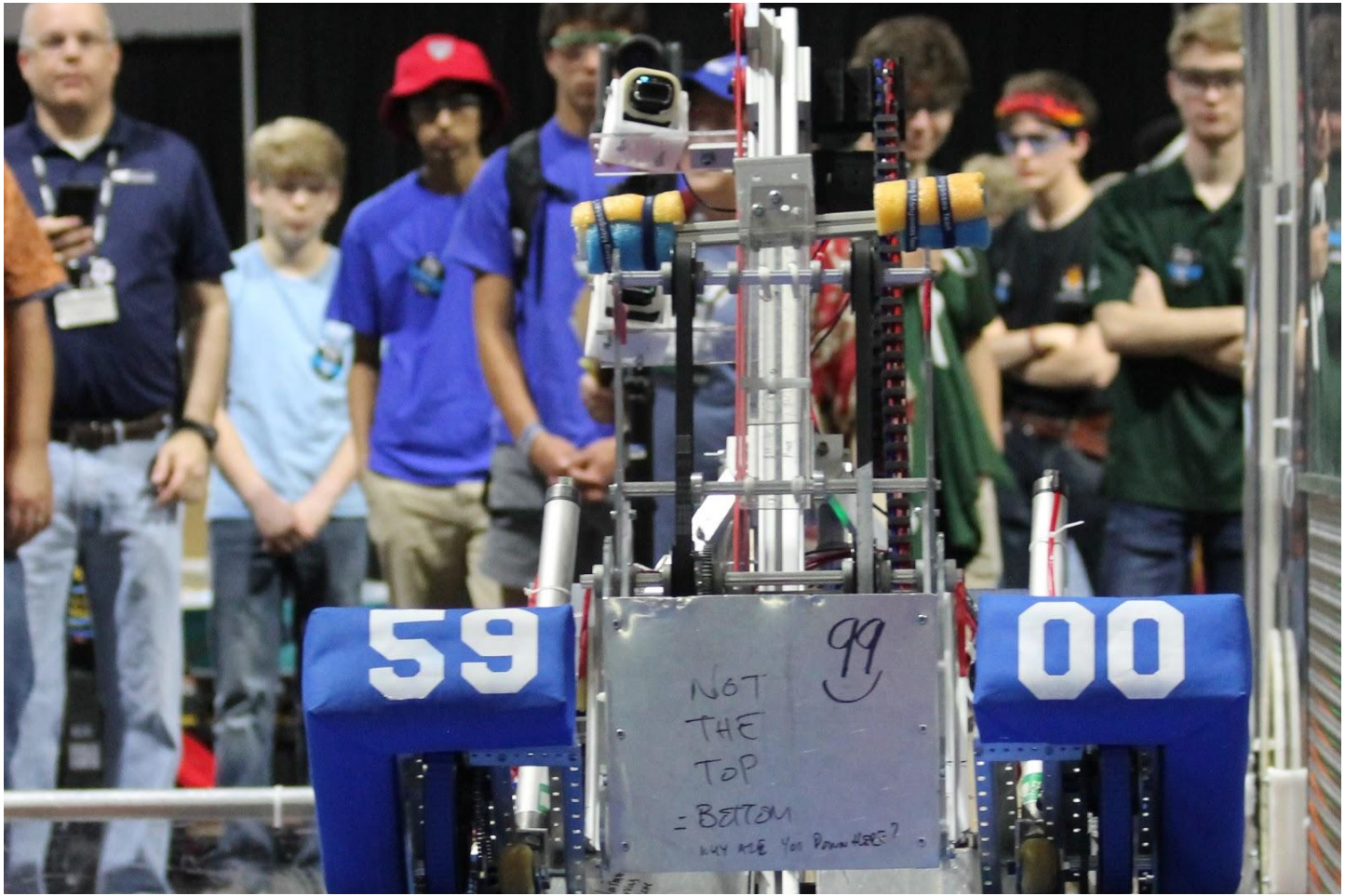
Trevor Silvers

Ian Scaife

(Java)

Bot: Rocket




59
00
99
NOT
THE
TOP
- BETTOM
WHY ARE YOU DOWN HERE?

Programmer for 2018 Power Up

Morrison Outlaw

(Java)

Bot: ZotsBot


aisc
Shaw
NWGA
Dalton
SIEMENS
KENDALL
ELECTRIC
B & M
ALUMINUM
COMPANY
ALROL
Visual
Print Group
ATLANTIC
Shaw
FLOORS
00
59

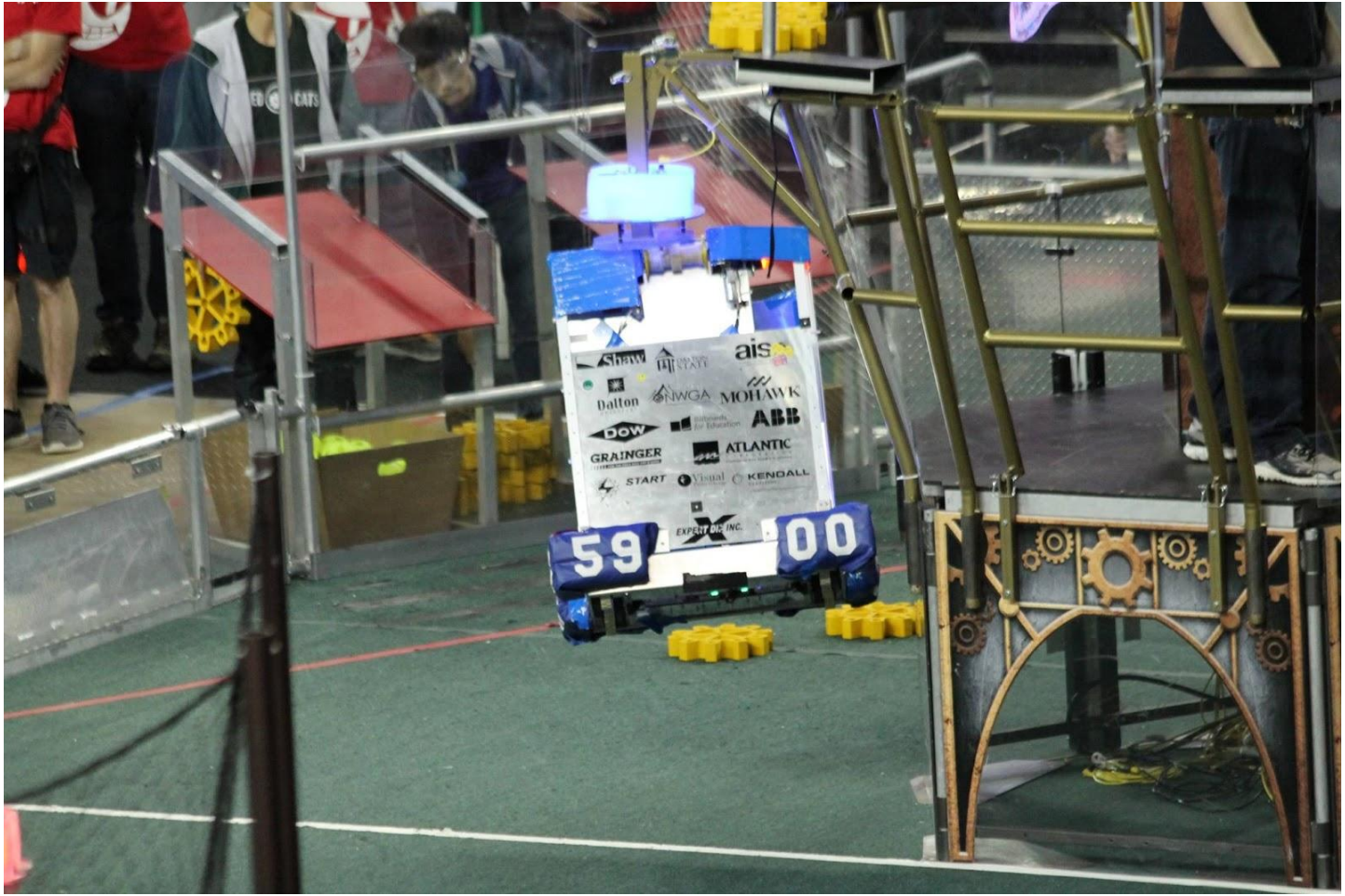
Programmers for 2017 Steamworks

Gage Godfrey

Jon Tan

(Java)

Bot: Spoons


Shaw
ais
Dalton
NWGA
MOHAWK
DOW
ABB
GRAINGER
ATLANTIC
START
KENDALL
59
00

Programmer for 2016 Stronghold

Elijah Pemberton

(LabView)

Bot: LemonSweets

[redacted]

[redacted]


CHARGERS
TEAM 4189
FEEL THE
MONGOOSE
59
00
Safety:
FOOD or
INKS in
work area