Full Metal Beavers

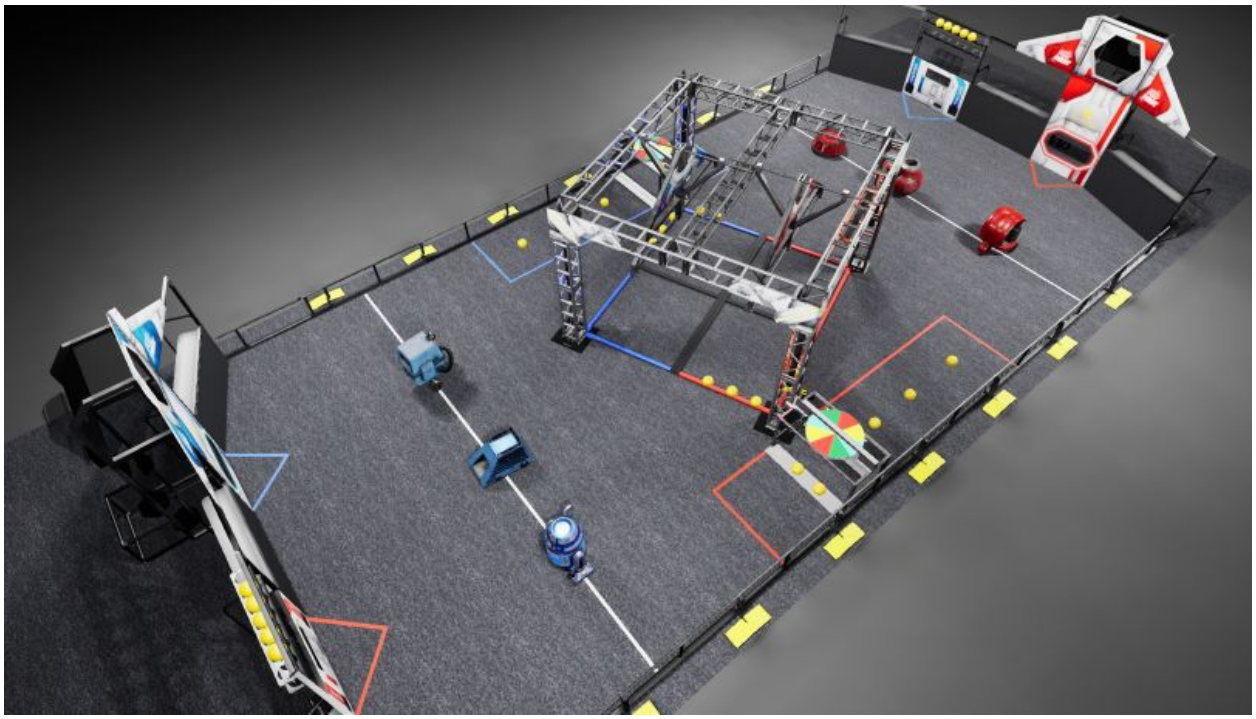
FIRST® Team 6636

Robot Technical Handbook

2020 Revision. This document is subject to periodic review, last revised [April 20th, 2020]

Table of Contents

Robot Functionality Objectives



- **During this season, our robot will have to complete a series of tasks in order to energize and activate a shield to protect FIRST city from asteroids.**
- POWER PORT - Score power cells into their respective entries. Our robot is aiming for the lower level of the POWER PORT by picking up power cells from the Alliance LOADING STATION, opposite of the POWER PORT
- CONTROL PANEL - Spin the control panel a minimum of 3 times but not over 5. Must stay on the color for at least 3 seconds. Our robot aims to move the control panel at least 3 times and then to stay on a given color at least for 3 seconds long. For both Rotation and Position control, the alliance is granted 30 points.
- CLIMB MECHANISM- To hang onto the generator switch efficiently nearing the end of the match to guarantee the alliance 25 points. The robots CLIMB MECHANISM will grant the alliance an extra 20 points over the 5 points given to alliances for PARKING the robot.

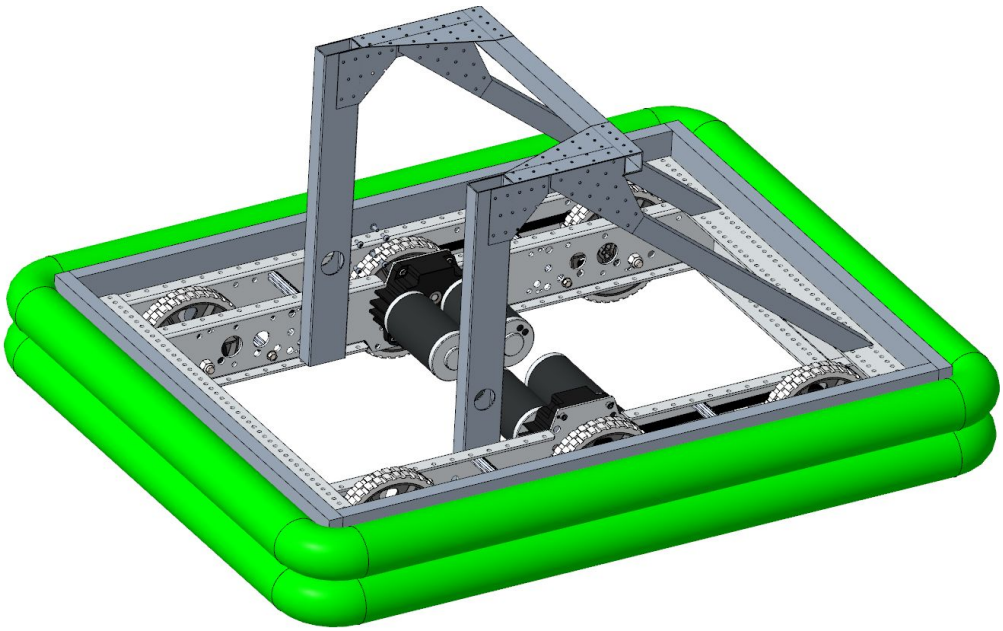
DriveTrain

The drivetrain that we will be using for our 2020 Robot will be the 6 wheel tank drivetrain which is 7 in. x 7 in. x 34 in. (The term "long chassis" is another name for the frame of the drivetrain) . It was created by Andymark which comes with all the parts and a manual to help you assemble the chassis. It weighs 25 lbs and it features HiGrip Wheels, Toughbox Mini Gearboxes, a Gates HTD Belts, and powered by CIM motors

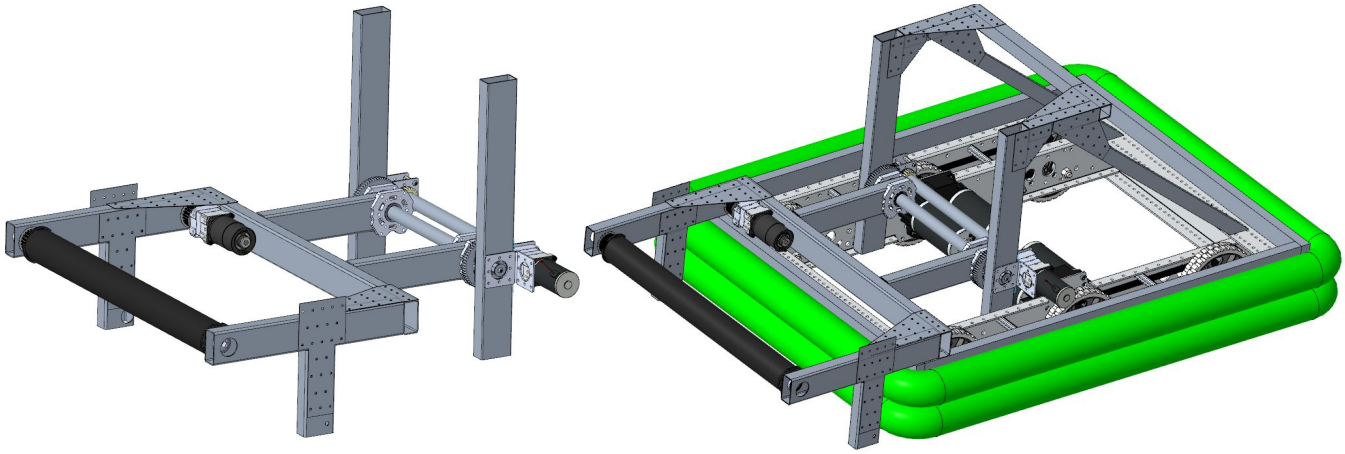
The image above is called the CIM motor. There are 4 CIM Motors in the drivetrain and they are essential to the drivetrain since the motors themselves power the drivetrain.

Structure

The Structure is a series of 2" x 1" box tubes, 1"x1" box tubes and 1" x 1" with galvanized plates that are connected to the chassis that create the surface to place the mechanisms on. The structure also involves bumpers that provide extra protection to the chassis. The bumpers are foldable so the robot can be used in both teams in the competition and it is all around the chassis making no cracks or open spaces that makes the chassis stick out. The structure also involved the drive train(which is described above). The bumpers are made out of fabric with wood and pool noodles.



Power Cell Mechanism



Arms:

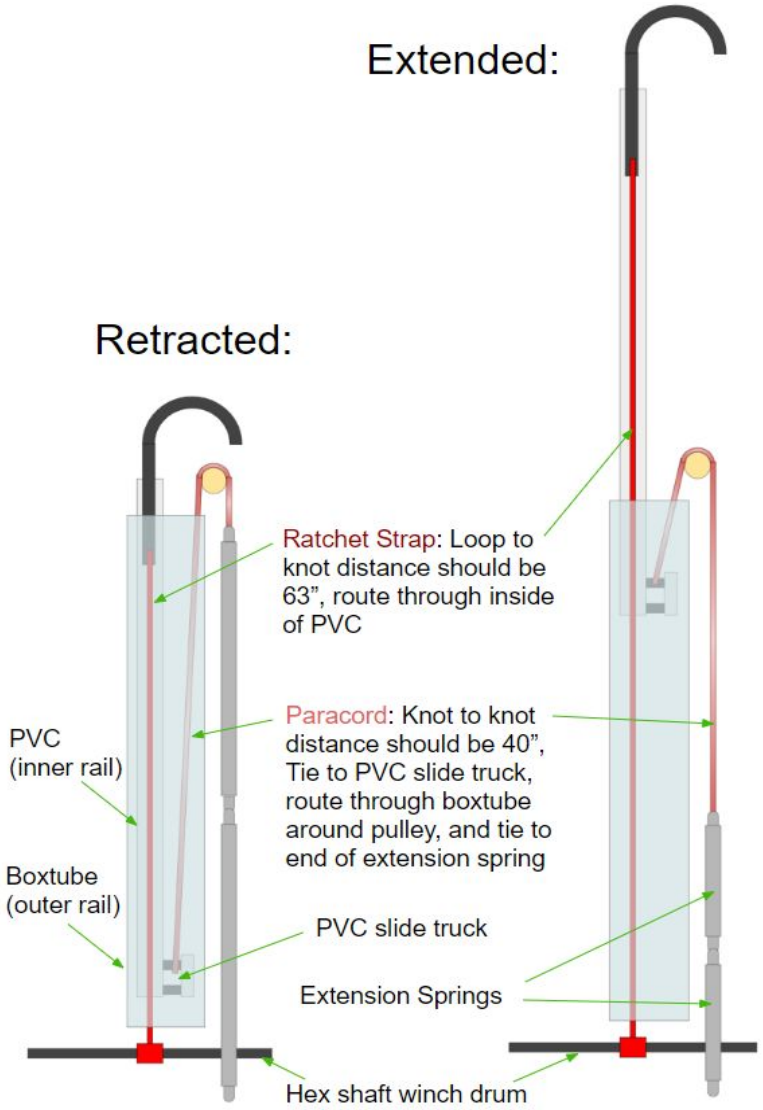
2 2"x 1"x 16.75" box tubes, each is surrounded by ½" hex bearings which are connected to Versablocks that are connected to the chassis. A BAG motor is used to power this mechanism which helps to pull in the POWER CELLS.

Intake:

The Power Cell Mechanism uses a 775 Motor to power this system. It uses galvanized steel plates to make the structure of the system and uses a 20" self-seal foam pipe. This structure uses the foam pipe to pull in the power cell from the floor, using a bag motor.

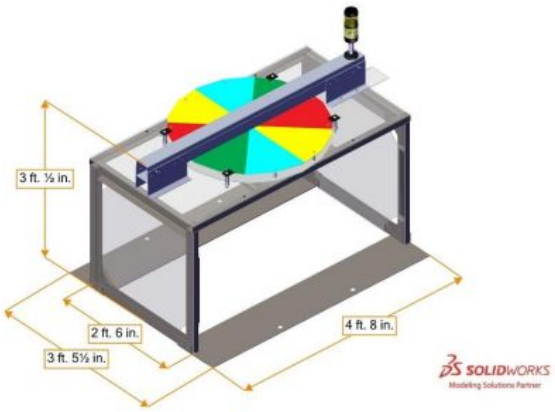
Climbing Mechanism

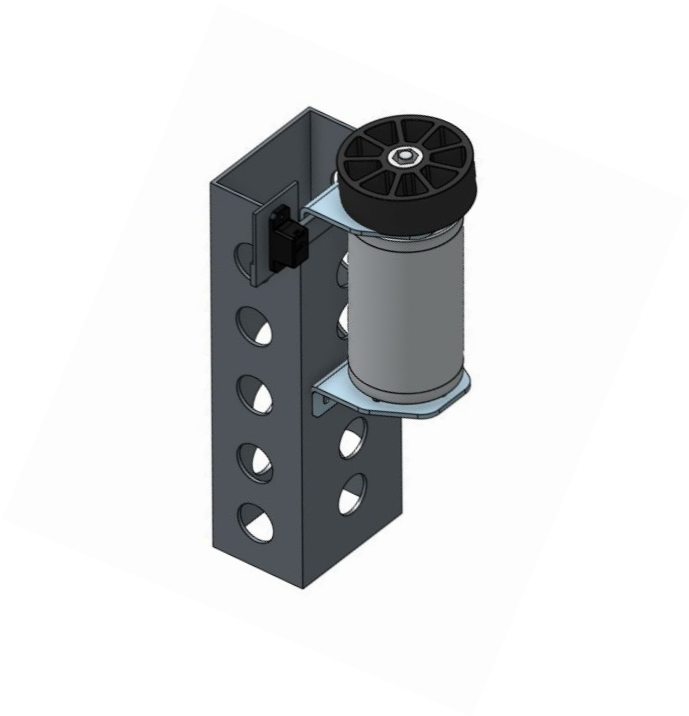
For our robot, it uses a 2" by 1" box tube as an outer system similar to a telescoping system to lift a hook to the generator switch bar. This mechanism uses a hook to grab the bar, a ratchet strap and a paracord to push and PVC pipe up and down. For the pipe to extend, a versaplanetary gearbox that consists of a miniCIM motor and a universal female shaft is used.



Control Panel Mechanism

The control panel mechanism is used in order to rotate the control panel to a specific color. Players will be assigned different colors and must rotate the control panel to that color to gain position control. The mechanism contains a color sensor to detect the specific color, and then has a motor and a wheel in order to rotate the control panel when necessary.





Programming

This season, we used the Limelight for aiding our turning and aligning of our target. It makes it easier for the driver to shoot at a target using reflective tape to score points. We also used the Color Sensor for the robot to detect the color and react with motors using various colors of this year's game.

Vision Targeting:

- For our vision target system, we use the targets area and horizontal position using limelight values to calculate the distance and the angle at which is best for our shooting.
- We used Smart Dashboard as a way to give us feedback and to track the area, x- and y- angles at real time.

Color Detection:

- For our color detection, we used four different colors (blue, red, green, and yellow) for this year's 2020 competition and tested it with the Color Sensor V3.
- Similar to the Limelight, we used Smart Dashboard to track the colors using hex decimals and have it match the color it sees. After it sees the color, the motors will react based on the color it needs to be activated.

Programming (cont.)

Drivetrain Code:

For our drivetrain code, we used TalonSRX motor controllers for controlling the speeds of which the motors steer and drive. We used TalonSRX because, unlike other motor controllers, we could communicate with them by using CAN protocols. It also has its interface called the Phoenix Tuner to calibrate them.

```
// Motor Controllers
 WPI_TalonSRX frontLeft = new WPI_TalonSRX(3);
 WPI_TalonSRX rearLeft = new WPI_TalonSRX(2);
 WPI_TalonSRX frontRight = new WPI_TalonSRX(0);
 WPI_TalonSRX rearRight = new WPI_TalonSRX(1);

//DriveTrain
SpeedControllerGroup lSide = new SpeedControllerGroup(frontLeft, rearLeft);
SpeedControllerGroup rSide = new SpeedControllerGroup(frontRight, rearRight);
DifferentialDrive driveTrain = new DifferentialDrive(lSide, rSide);

//Driving
final double move = driveController.getRawAxis(1);
final double turn = driveController.getRawAxis(4);
driveTrain.arcadeDrive(-move * 0.6, turn * 0.7);
```

How we did it:

1) List every TalonSRX used for driving.
2) Then we group the Talons depending on which controller they will be controlling on the chassis.
3) Combine the groups using "DiffrentrialDrive" to control the whole drivetrain.
4) We bounded the "move" and "turn" to the controller's joystick.
5) Lastly, we used arcadeDrive for driving and steering purposes.

Limelight Code:

```
Update_Limelight_Tracking();
 if (auto)
{
  if (m_LimelightHasValidTarget)
  {
driveTrain.arcadeDrive(m_LimelightDriveCommand,m_LimelightSteerCommand);
  }
  else
  {
  driveTrain.arcadeDrive(0.0,0.0);
  }
 }
 else
 {
  driveTrain.arcadeDrive(-move*0.6, turn*0.7);
 }
}
```

This part of the code is the robot goes up to the target with your given values.

- "*m_LimelightHasValidTarget*" is when the Limelight detects the target in its field of view.
- "*m_LimelightDriveCommand*" is when the limelight goes up to the target with the given ta (target area) with your given speed.
- "*m_LimelightSteerCommand*" is the turning speed when the center of the target is off the limelight view.

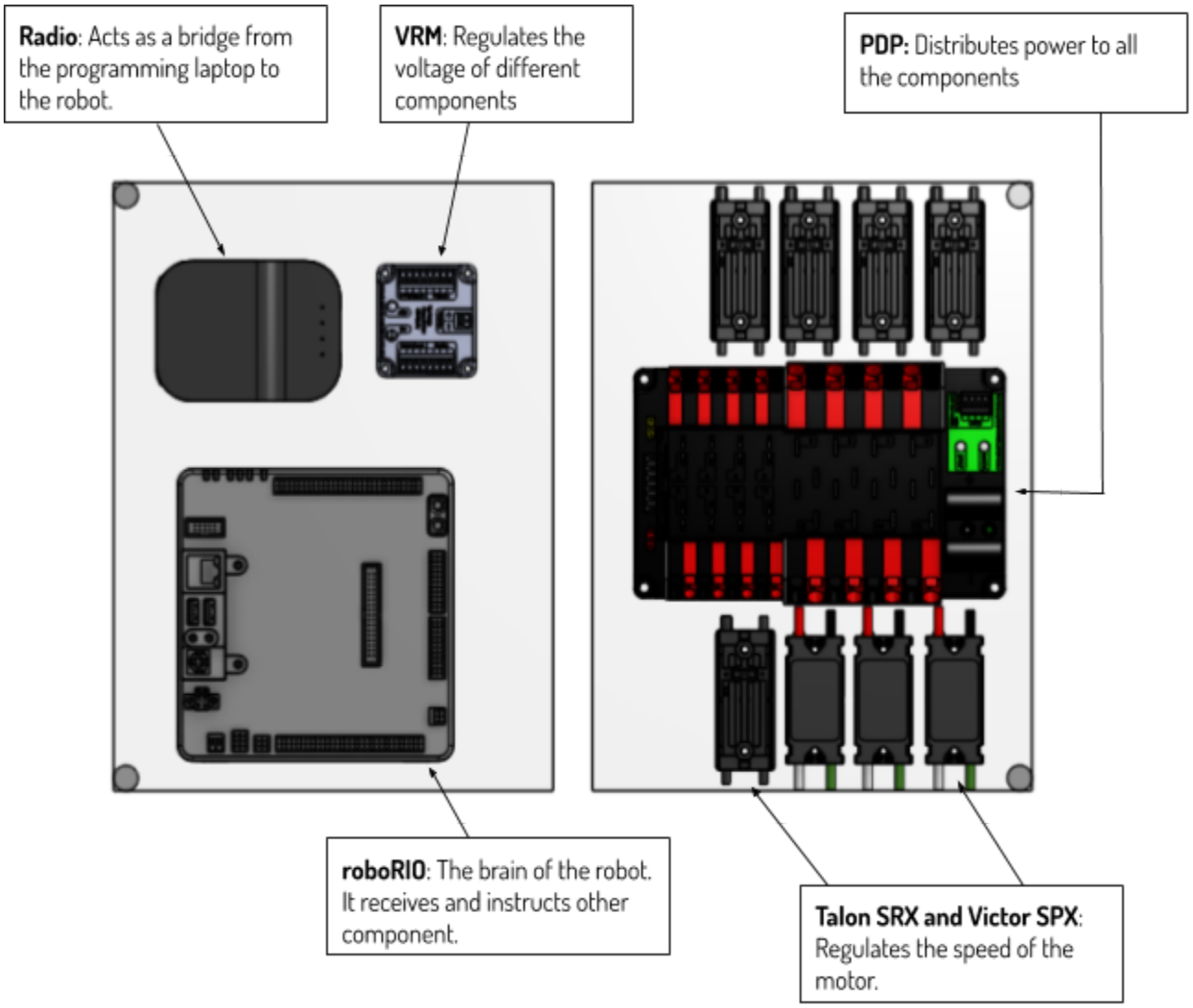
Color Sensor V3 Code:

```
//Blue
 if (match.color != kBlueTarget & m_Controller.getRawButtonPressed(2)) {
  controlpVictor.set(0.5);
  }
  else
{
  controlpVictor.set(0.0);
  }
//Red
  if (match.color != kRedTarget & m_Controller.getRawButtonPressed(3)) {
  controlpVictor.set(0.5);
  }
  else
 {
  controlpVictor.set(0.0);
  }
```

For the first part, when the Color Sensor does not see the color blue "*!=*" and the controller presses the 2nd button, the motor will move until it sees the color blue, then it will stop. The same goes for the color red, but instead of the 2nd button, you will need to press the third button, and it will find the color red. Both parts of the code still have the same function and find the color on the Control Panel Mechanism.

Electrical

The electrical box consists of all the electrical components on our robot. Our electrical box is broken up into two pieces; one part consists of the Roborio, Radio, and the VRM, while the other is PDP and all the motor controllers. The purpose of doing this is to have quicker access to work with the components.



Build Documentation

For the 2020 robot, we started to work on its chassis of which we had a spare one but we needed a few changes. It took a while to fully make the new chassis since there were a few problems along the way.

- The gear boxes needed to be disassembled and reassembled since the gears were getting stuck or were in the wrong place.
- The chassis also had a few problems with its dimensions so we used the table saw and the dremel to cut some of the metal to make the chassis into a long chassis.
- After that, we started to work on the chassis from scratch by reassembling it. After adding all of the modifications, the chassis was ready to be used and tested.
- After making and fixing the new chassis, we worked on creating the power cell mechanism and the bumpers.
- For the construction of the bumpers, we first needed to cut four pieces of wood.. Since the chassis had different measurements, the wood had to be cut into different sizes. After that was done, we then started to cut pool noodles that matched the sizes of the wood and then hot glued them onto the corresponding piece.
- Afterwards, we attached the bumper fabric (With help from online tutorials) the bumpers were complete. At the same time, we started to work on the power cell mechanism.