**1.**      **Project Implementation**

For the drivetrains we used mecanum wheels to allow for strafing. Our intake uses surgical tubing in which we used 3D printed carriers to hold them in place. Our outtake is a bucket that goes down to drop the pixel. We used the REV Control Hub. As for software, we used Android studio (Java) to code all the autonomous and driver-controlled actions. Our autonomous codes detect where our team prop is using a distance sensor. and place a pixel on the tape it is on. This works about 60% of the time. Then it goes to the backdrop and places the pixel on it.

**2.**      **Results and Data**

Our original autonomous program worked 80% of the time, because in the rear autonomous, the wheels would strafe and would not park. Our current autonomous works 20% of the time. It would rarely drop the pixel on the spike marker tape and put the pixel on the backdrop. During the drive control period, we were able to get 6 pixels on the backdrop during a 2-and-a-half-minute period. Our drone was one of our most consistent parts of our robot. It worked 100% of the time, getting 10 – 20 points every time. Our hanging worked 20% of the time and was not very reliable to the point that parking would get more points.

**3.**      **Lessons Learned**

**Object Detection and Autonomous Improvements**

The distance sensor also is not reliable enough so we will use a touch sensor. The distance sensor would not detect the team prop entirely and detect something else farther away. We were going to use a webcam to detect the team prop. We would do it by checking which spike marker tape had the most color saturation and proceeding the autonomous code from there. To detect the team prop, we used 2 distance sensors, but we found out that they interfered with each other. They would detect both spike marker tapes and would confuse the robot. We then changed to 1 distance sensor. For future autonomous actions, we will want to take from the pixel stacks. This will get more points in autonomous. To detect the team prop, we used 2 distance sensors, but we found out that they interfered with each other. They would detect both spike marker tapes and would confuse the robot. We then changed to 1 distance sensor. For future autonomous actions, we will want to take from the pixel stacks. This will get more points in autonomous.

**Pixel Dropping and Arm**

When we first started out as a team, we used a claw to pick up the pixel. We then used a rack and pinion to extend and put the pixel on the backdrop. As we found out, picking up the pixel with the claw was not consistent enough and sometimes would not even pick up the pixel. When we were able to pick up the pixel, the rack and pinion was not long enough and did not reach the backdrop. Our current bucket is not the most efficient outtake we could do. During competitions the pixels drop from the bucket too slowly. What we will do is use a prong release outtake, so we can use our bucket design but add prongs to where the pixels go in. The prongs will touch the backdrop and push a cover connected to it and let the pixels out. When we were testing the arm, we found out that the arm would go too far back and the pixels would fly out. To fix this, we used zip – ties to make sure the arm would not go too high. In the future, we are thinking about using a PID loop to keep the arm from going too high.

**Intake**

For our intake, we were going to use compliant wheels, but after we tested the surgical tubing and the compliant wheels, we found out that the surgical tubing intake the pixels better. We also learned that using 3 - D printed parts for important parts of the robot was good as we broke our shaft that we mounted our surgical tubing on. We now use a metal shaft. We also changed the way we mounted our surgical tubing. We used to use compliant wheels to mount our surgical tubing, but the surgical tubing would fall off easily. We then 3 – D printed a surgical holder (*18253 Beach Bots*), and our surgical tubing doesn’t fall off anymore.

**Drone Launcher**

Currently, our drone launcher works very well. Before it would only land on the drone tapes 20% of the time. After using smaller rubber bands that provided extra resistance the drone flew farther. At that point, we only had to make and test different paper airplane designs to see which worked the best. We finally made a drone design that worked 100% of the time bar a time when the launcher got stuck.

**Hanging**

Our hanging was easily the least reliable part of our robot. The part that was supposed to make the robot hang would consistently fall off, and we would lose points because of it. It also had a motor that reeled in the string to make it hang.