Team 9973B Robot Introduction

Urumqi No.1 Senior High School

Author:

Zijin Wang

Mentor:

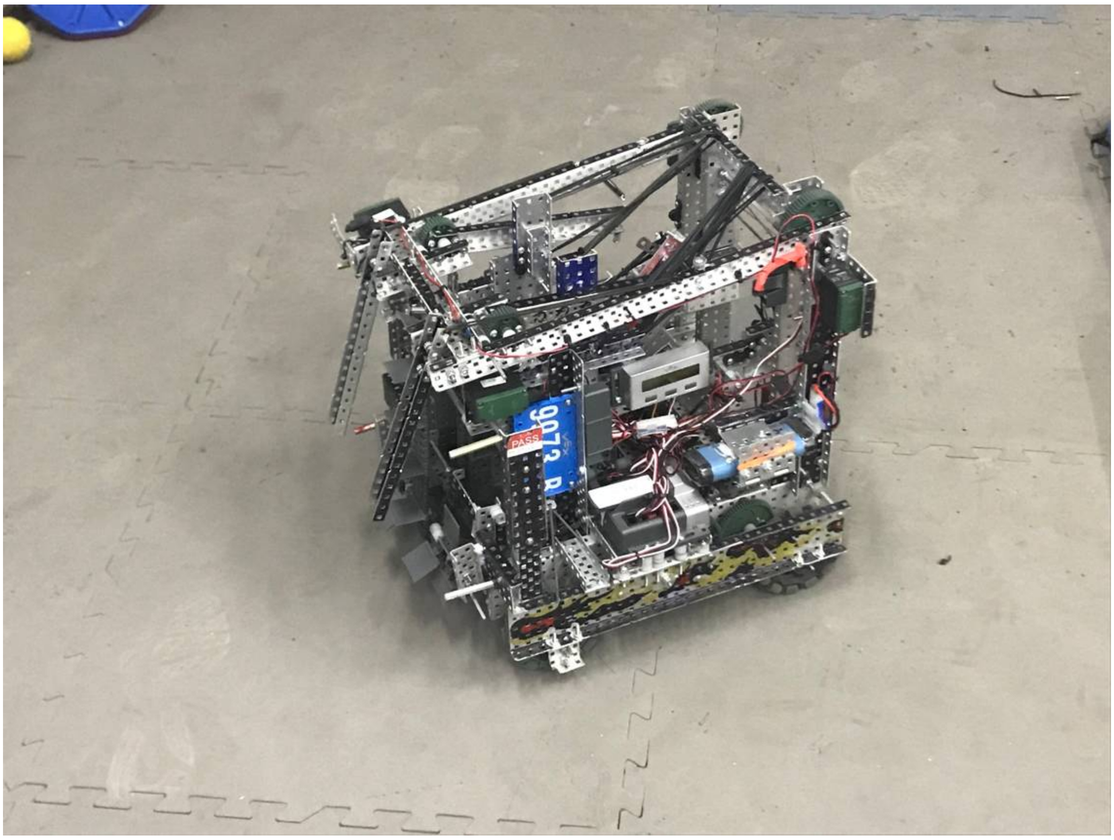
Yifei Han

July 2018

9973B Robot Introduction

Zijin Wang

Here is our robot for the VEX Game 2019-"Turning Point"



Members&Position :

This robot is built mainly by Zijin Wang; improvements and optimization are done by Yueyan Ma; programs are written by Zijin Wang; examinations and repairs are done by Qianhui Zhao.

Automatic strategy :

For the automatic part of the competition, I've programmed 3 different program sets to confront different circumstances in the game.

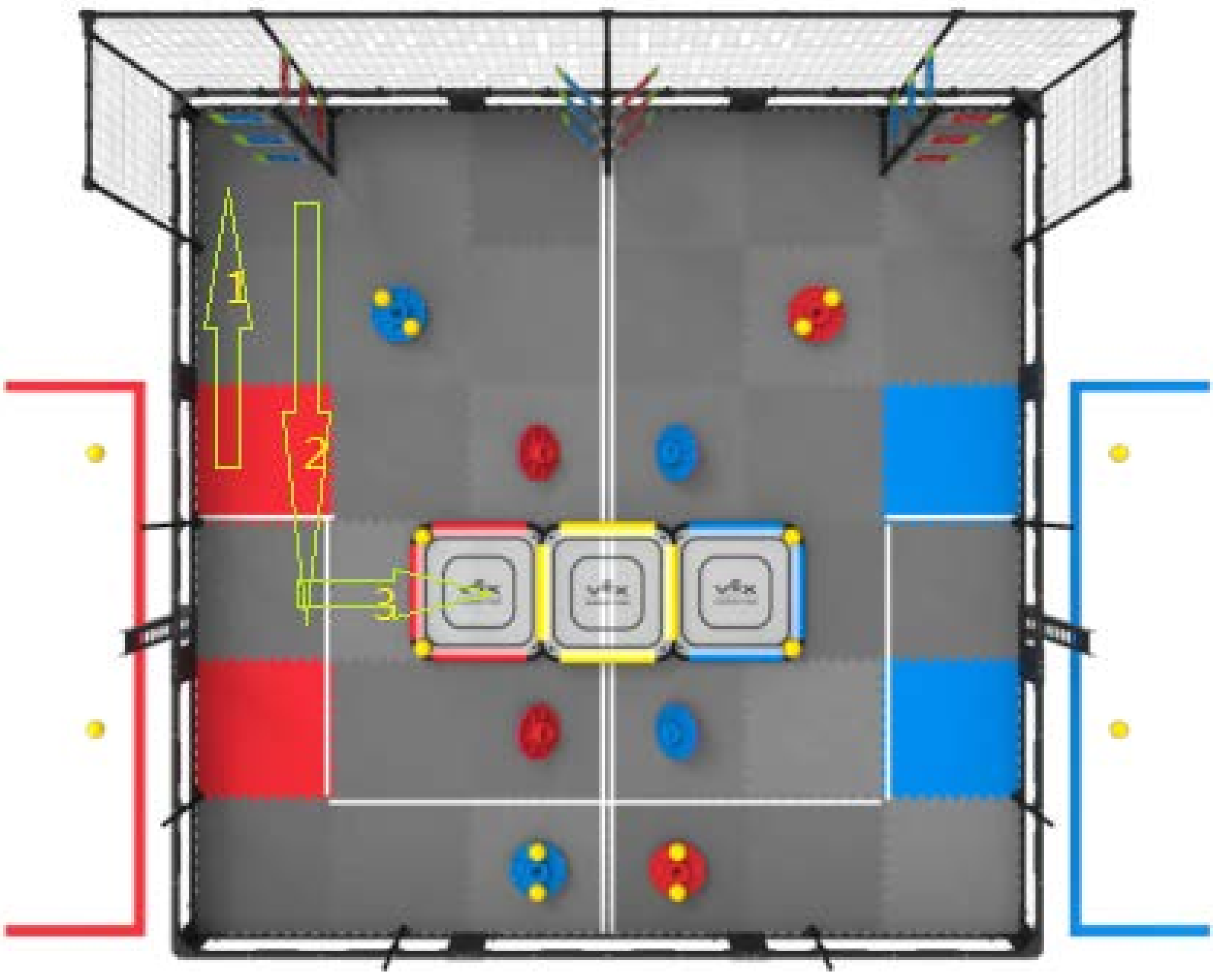
Plan 1 :

1. Hit the middle flag

2. Hit the lowest flag

3. Get back

4. Go onto the platform



Code :

void auto_near(char side,bool platform)

{

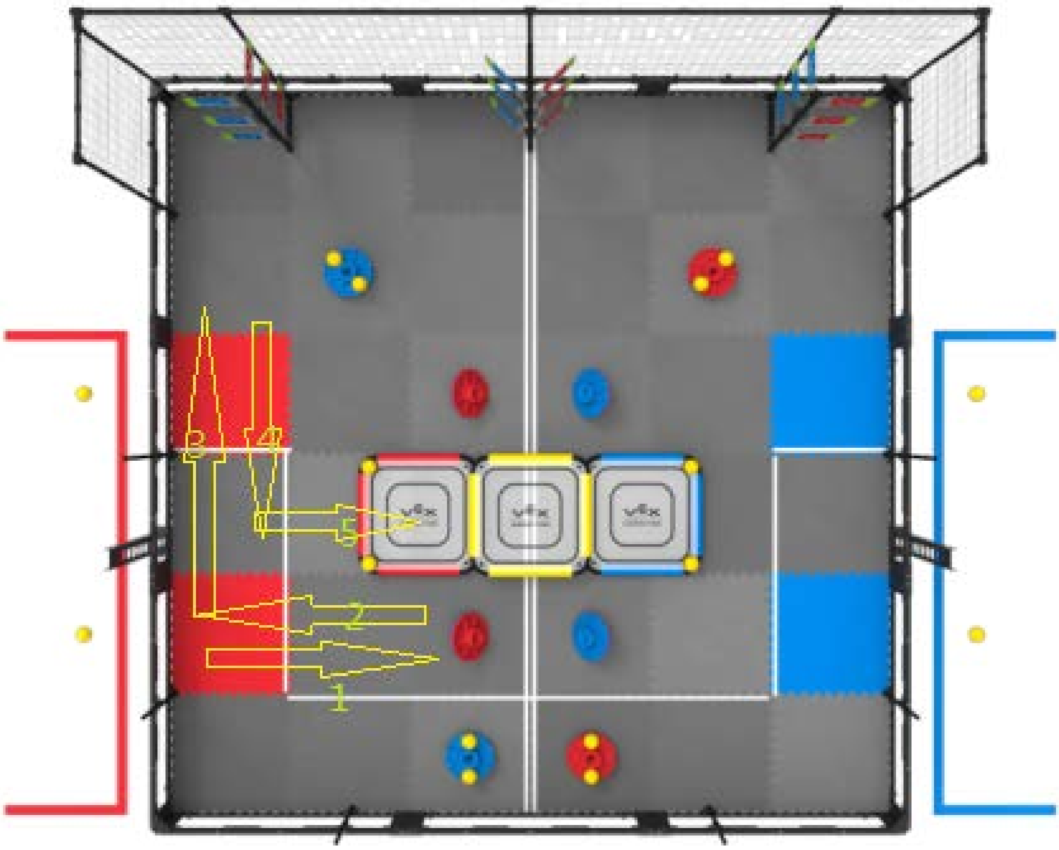
```
short n=1;
if(slide=='b') n = -1;
Gyro = 0; //Clear the Gyroscope Sensor

if(platform==false)
{
        Roller=127;
        move(100,100,1800)
        wait1Msec(1000);Roller=40;
        move(-100,-100,1700);
        trun(-n*60,n*60,n*90);
        hit_flag();
        Roller=127;
        move(100,100,1200);wait1Msec(200);
        hit_flag();
        move(100,100,400);//the second flag
        move(-100,-100,1000);
}
else
{
        move(100,100,1200);wait1Msec(200);
        hit_flag();
        move(100,100,400);//the second flag
        move(-100,-100,3300);
        turn(-n*60,n*60,n*90);
        wheel(60,60);wait1Msec(500);
        wheel(0,0);wait1Msec(500);
```

```
        wheel(-127,-127);wait1Msec(3000);
        wheel(0,0);
    }
}
```

Plan 2 :

1. Get the ball under the cap
2. Go forward
3. Hit the highest flag
4. Go onto the platform

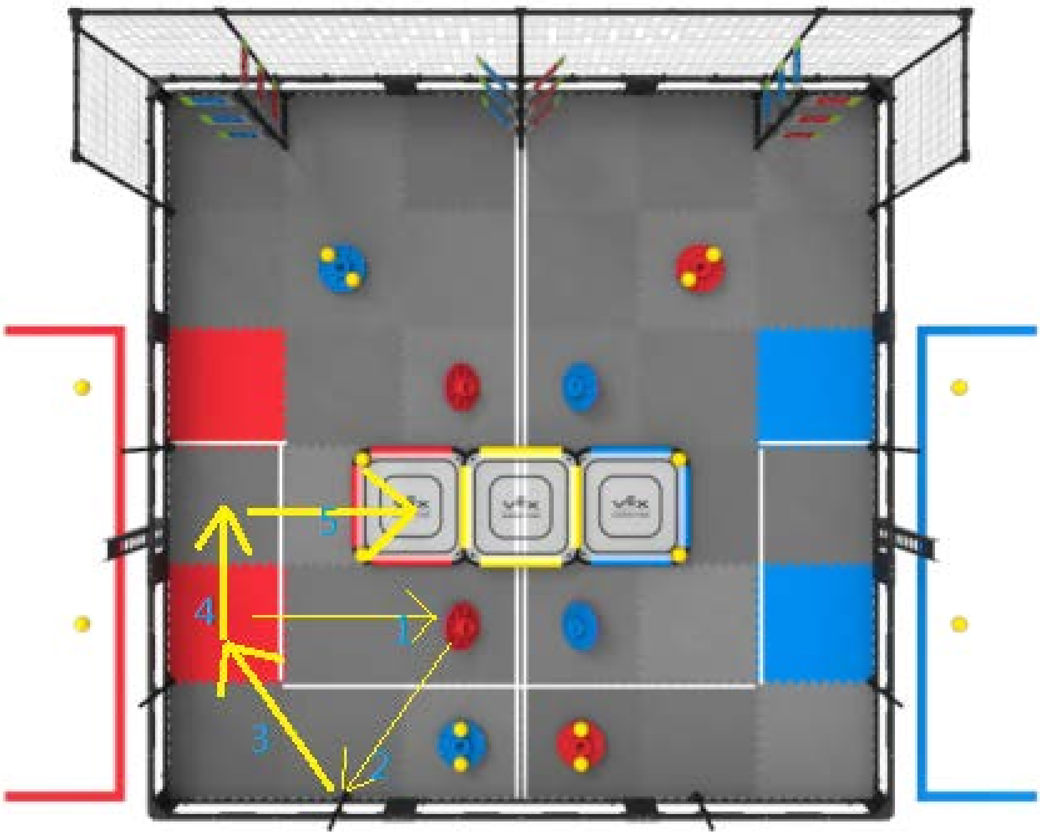


Code:

```
void auto_far(char side)
{
    short n=1;
    if(side=='b') n = -1;
```

```
    Gyro=0;
    Roller=127;
    move(100,100,1800);
    wair1Msec(1000);ROller=0;
    move(-100,-100,1700);
    turn(-n*60,n*60,n*90)；
    move(-100,-100,1100);
    turn(-n*60,n*60,n*180);
    wheel(60,60);wait1Msec(500);
    wheel(0,0);wait1Msec(500);
    wheel(-127,-127);wait1Msec(3000);
    wheel(0,0);
}
```

Plan 3:



1. Flip the cap
2. Get the ball under the cap
3. Go onto the platform

Code：

```
void auto_cap(char side)
{
    short n=1;
    if(side=='b') n=-1;

    Lifter=-127;while(Back_limit==0);Lifter=0;
    move(-100,-100,1600);

    Claw=127;waitMsec(500);
    Ecd=0;wheel(-n*60,n*60);while(abs(Ecd)<120);
    wheel(n*60,-n*60);wait1Msec(50);wheel(0,0);

    move(60,60,1550);
    wheel(30,30);
    Lifter=127;wait1Msec(2000);Lifter=-30;

    Gyro=0;
    Claw=-127;wait1Msec(250);move(-100,-100,300);wait1Msec(250);
    turn(n*60,-n*60,-n*90);
    Lifter=127;wait1Msec(500);
    move(100,100,1300);
    Claw=Lifter=0;
```

```
turn(-n*60,n*60,n*0);
wheel(60,60);wait1Msec(1000);
wheel(0,0)wait1Msec(500);
wheel(-127,-127);wait1Msec(3000);
wheel(0,0);
}
```

Structure Features:

1. Base System

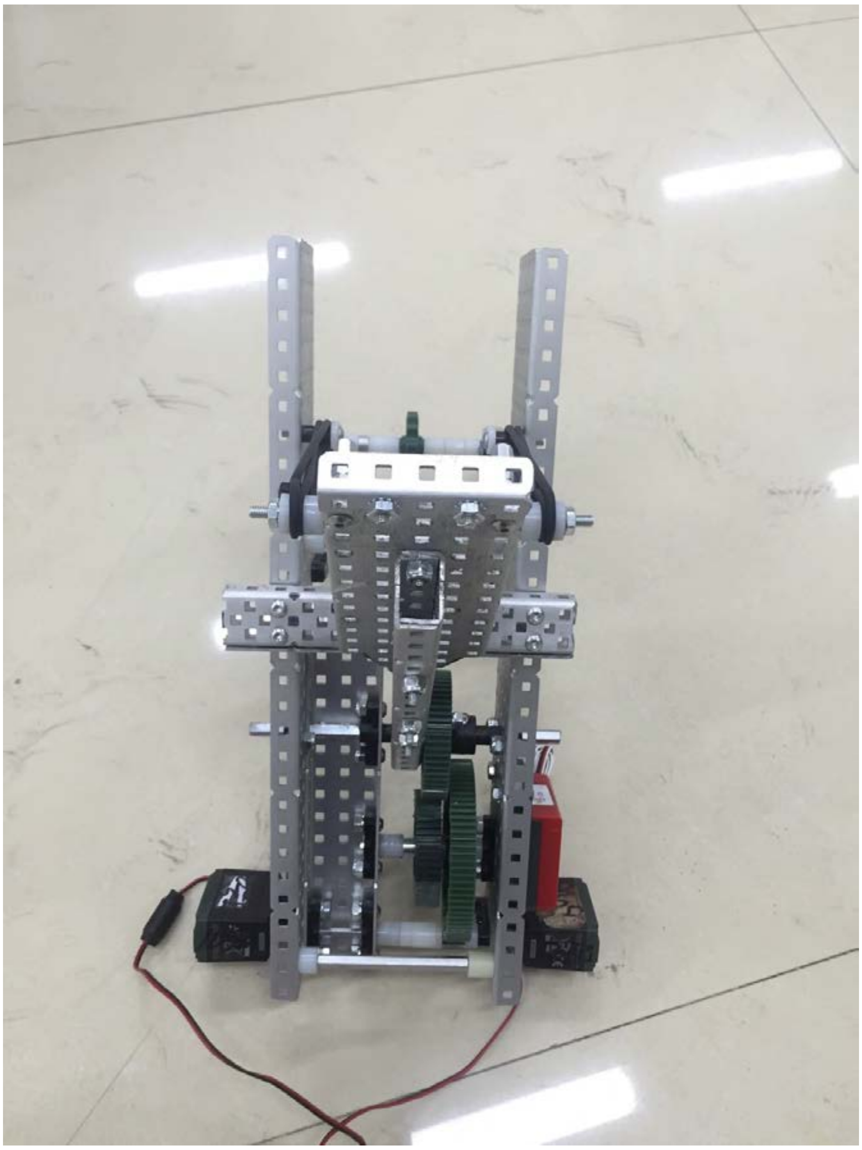
Result from our experiments proves that the base needs to be built taller .To prevent the robot from motor over-heating , 6-motor-base system is necessary.

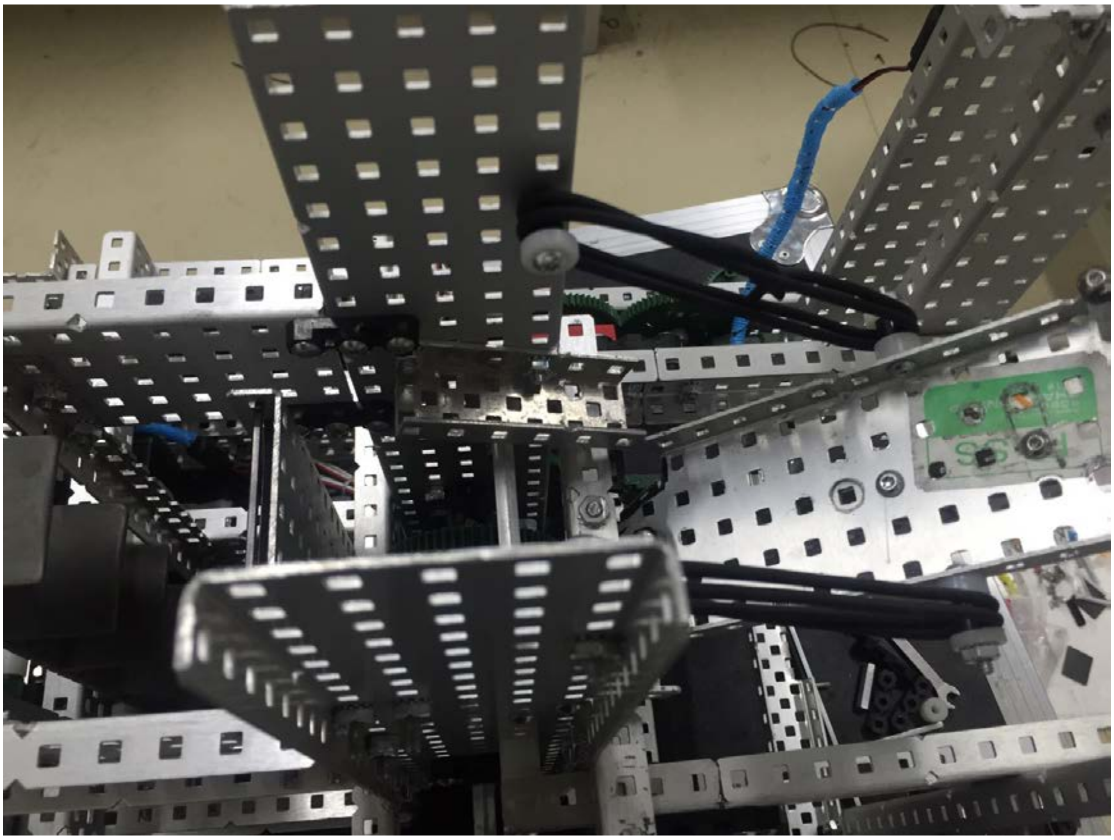
If the base, as the core of a robot, malfunctions, the whole robot turns into a frozen wall. Therefore, balancing performance and battery and preventing over-heating are extremely important.

As for the controller, a "turbo" button on the joystick allows the base speed to increase from it’s original speed(85) to it’s maximum speed(127).

2.Shooting System

We designed a stone-shooter model as our shooting prototype.

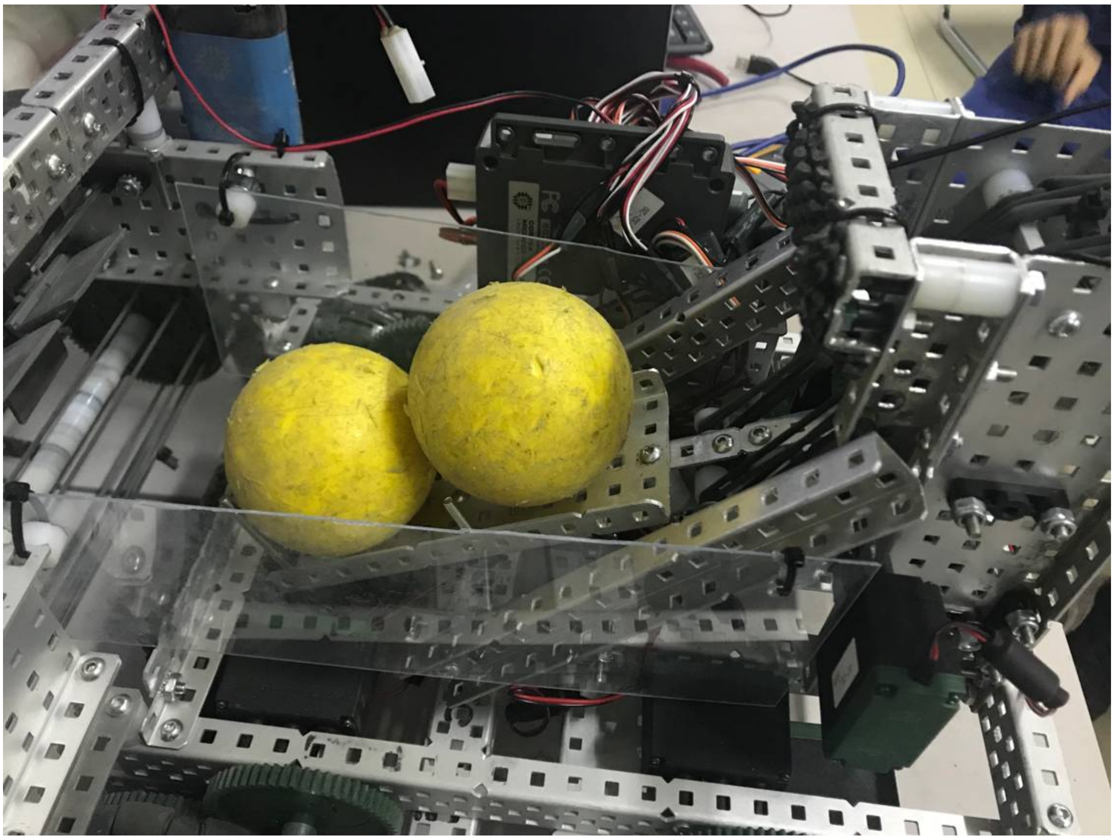




Other vital details:

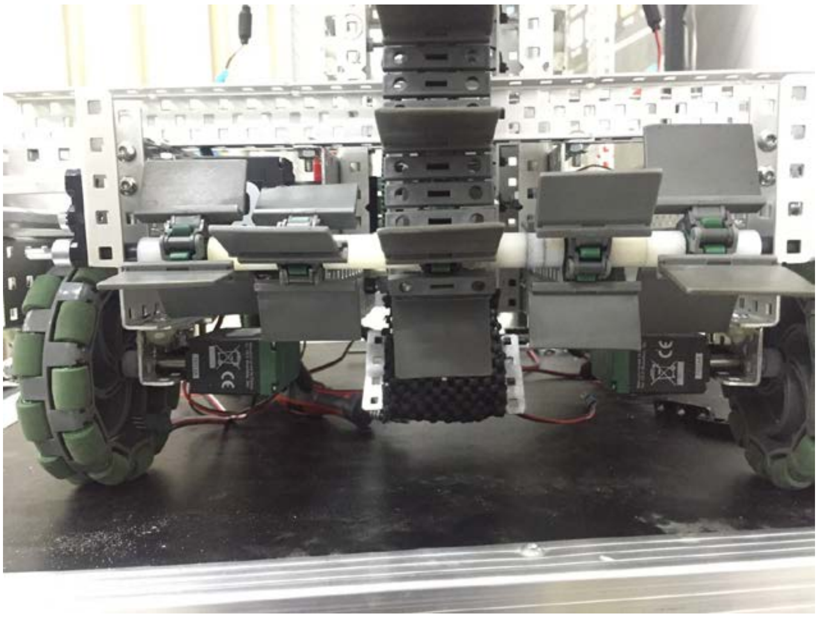
To make it easier for the shooting system to obtain balls, we equipped it with a Potentiometer modifying the platform at the appropriate place to stable its trajectory.

The distance between each limit made from nails requires a lot of tests to achieve its perfection.

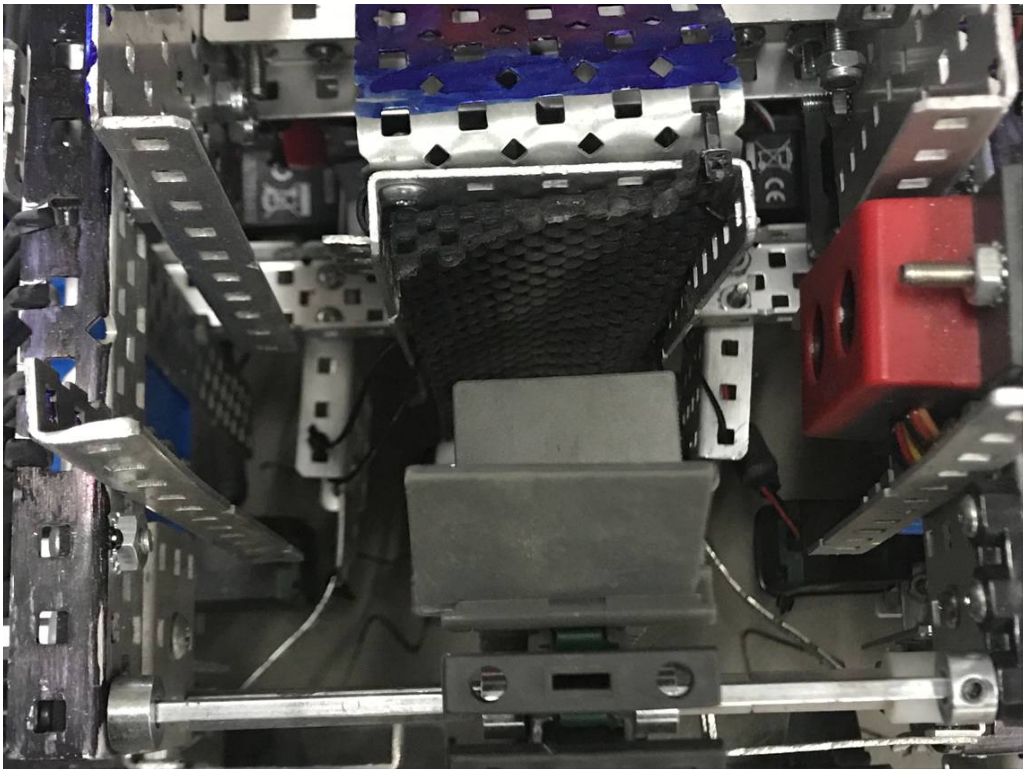


3. Elevator System

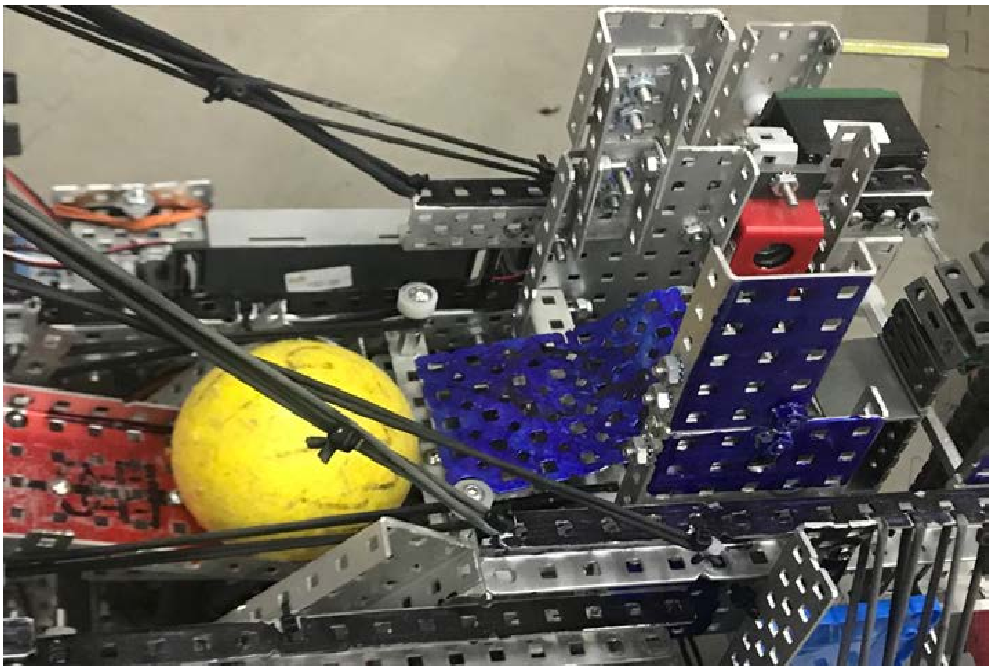
When the balls reach the lower roller, they will be sucked into the elevator eyetem. The motor is directly connected to a bigger clutch and a smaller clutch by chains.



Anti-slip mat is stuck on the another surface of our elevator system to make the transport of balls fast and smooth.



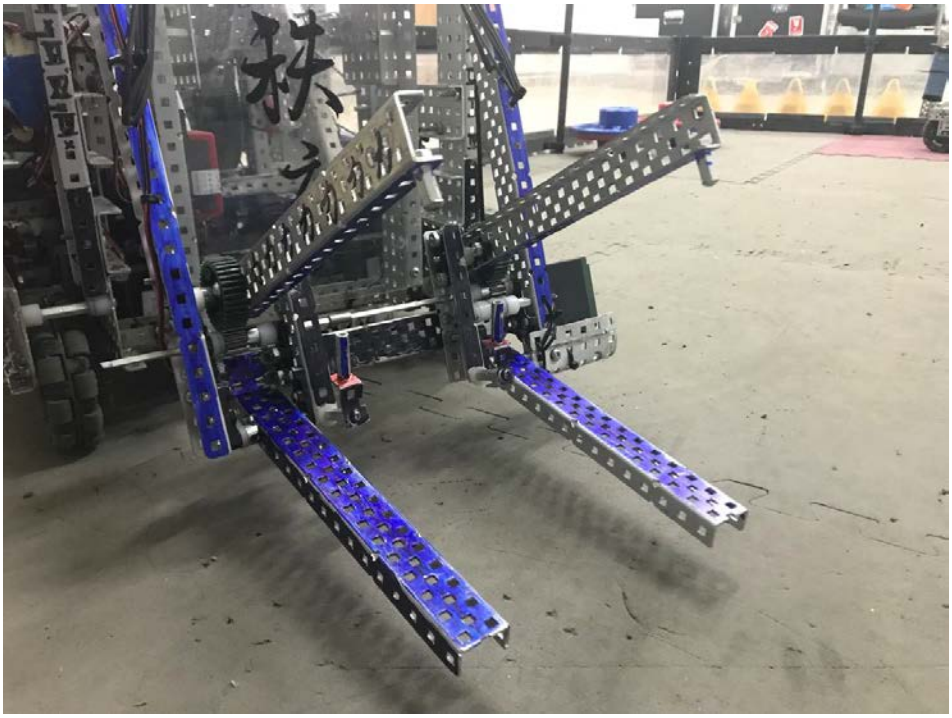
When a ball is lifted to the top of the system, it rolls down to the platform of the shooting system. The board at the end of the shooting platform is designed to prevent balls from falling by accident at high speed.



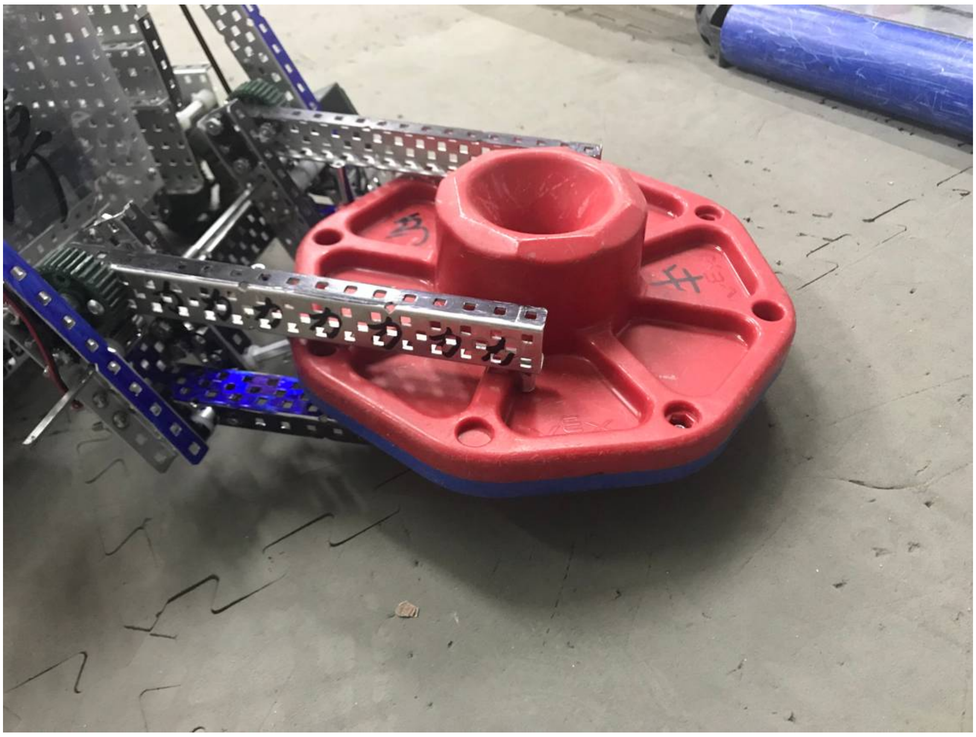
4. Lift&Claw System

The Lift&Claw system is designed to catch caps and place them on posts around the competition field.

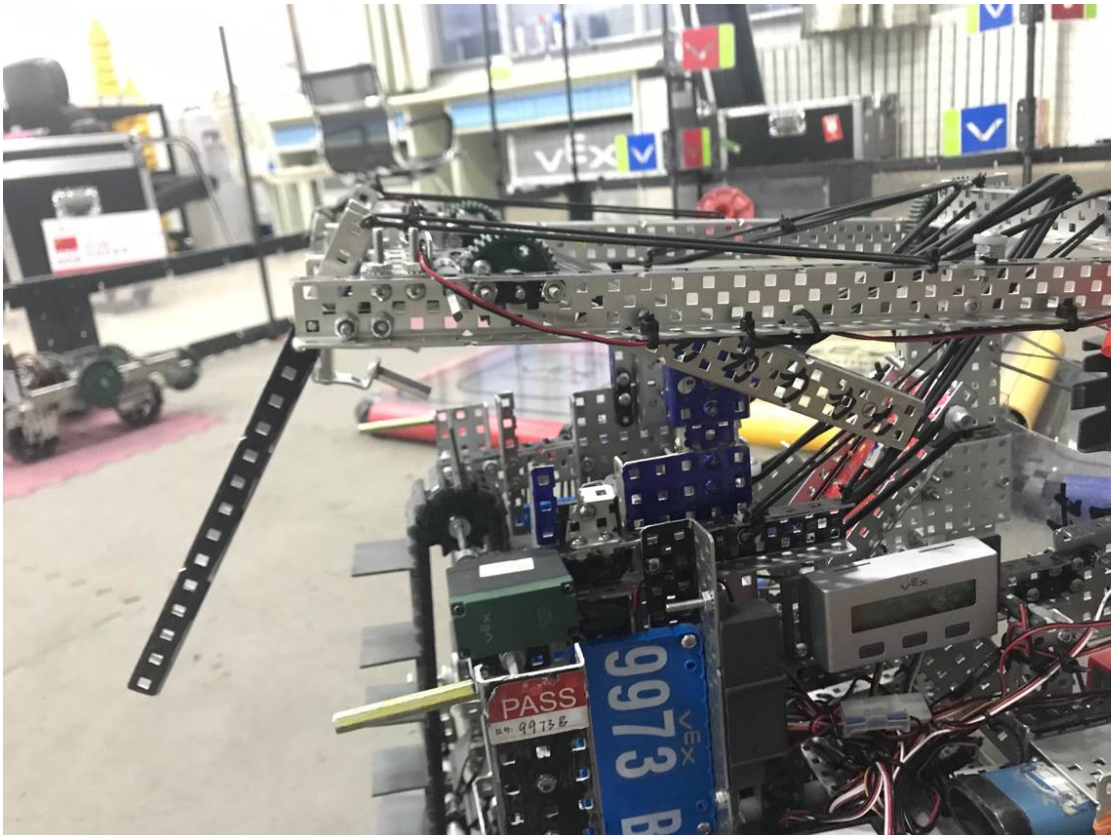
The claw is designed to hold the cap tightly enough from falling down while it is hanging in the air. The two parts of the claw are designed wide enough to hold the cap as stable as possible. Two metal bars are sticked out from the bottom of the arm, producing a flat structure to hold the cap.



Another two metal bars are connected by two gears and a long axle. When the claw program is activated, the upper bars get down and press on the cap with appropriate speed in order to hold the cap tightly avoid over-heating of the motor.



The arms are made of two long metal bars driven by two separate motors. A small gear is connected to another bigger gear fixed on the arm to slow down the speed to provide more force.(Based on the formula P=F*V) The arms are designed to be folded when they are not operating in order to move the center of the gravity close to the center of the robot’s base,making the robot more stable while moving.



5. PID control and its application

Reference from Google:

PID is an acronym for the mathematical terms Proportional, Integral, and Derivative.

Proportional means a constant n such that $y = nx$. This n can be positive or negative, greater or less than one. To make sure the formula more accurate by PID controller standards, proportion is given by Kp and the x term is the control loop error ε: $y = Kp(\varepsilon)$ The term Integral means the summation of a function over a given interval. In case of controller PID that is the sum of error over time: $y = \int f(\varepsilon)dt$. Finally, Derivative is the rate of change during a given interval.

Interpreted by a PID controller: All three of these PID controller components create output based on measured error of the process being regulated. If a control loop function properly, any changes in error caused by setpoint changes or process disturbances are quickly eliminated by the combination of the three factors P, I and D.

Field mounted PID controllers can be placed close to the sensor or the control regulation device and be monitored centrally using a SCADA (supervisory control and data acquisition) system.

With the PID control, the robot can move more smoothly without unnecessary waste of energy and over-heating problems.

Idea: The closer the robot get to the set point, the slower it moves.

1) Moving with PID control

$$\text{Speed} = \text{Kp*full} - \text{speed}$$

$$\text{Kp} = \begin{cases} -0.7 + 1.7 \times \dfrac{\text{x}}{200}, & 0 \leq \text{x} \leq 200 \\ 1, & x > 200 \end{cases}$$

X is the distance to the set point.

Code：

```
float kp_move(float x)
{
	if(x>200)return 1;
	else return -0.7+1.7*x/200;
}
void move(short l,short r,short distance)
{
	Ecd=0;float k=1;
	while(abs(Ecd)<distance)
	{
		k=kp_move(abs(Ecd-distance));
		wheel(l*k,r*k);
	}
```

```
    wheel(0,0);wait1Msec(50);
}
```

2)Turning with PID control

Speed = Kp*full – speed;

$$\mathrm{Kp} = \begin{cases} -0.35 + 1.35 \times \dfrac{\mathrm{x}}{450}, & 0 \le \mathrm{x} < 450 \\ 1, & x \ge 450 \end{cases}$$

x = $|\mathrm{target} - \mathrm{now}|$

Code：

```
float kp_turn(float x)
{
    if(x>450) return 1;
    else return -0.35+1.35*x/450;
}
void turn(short l,short r,short direction)
{
    direction=10;
    direction=direction*19/18;
    float k=1;
    while(abs(Gyro-direction)>100)
    {
        k=kp_turn(abs(Gyro-direction));
        wheel(l*k,r*k);
    }
```

```
        wheel(0,0);wait1Msec(50);
    }
```