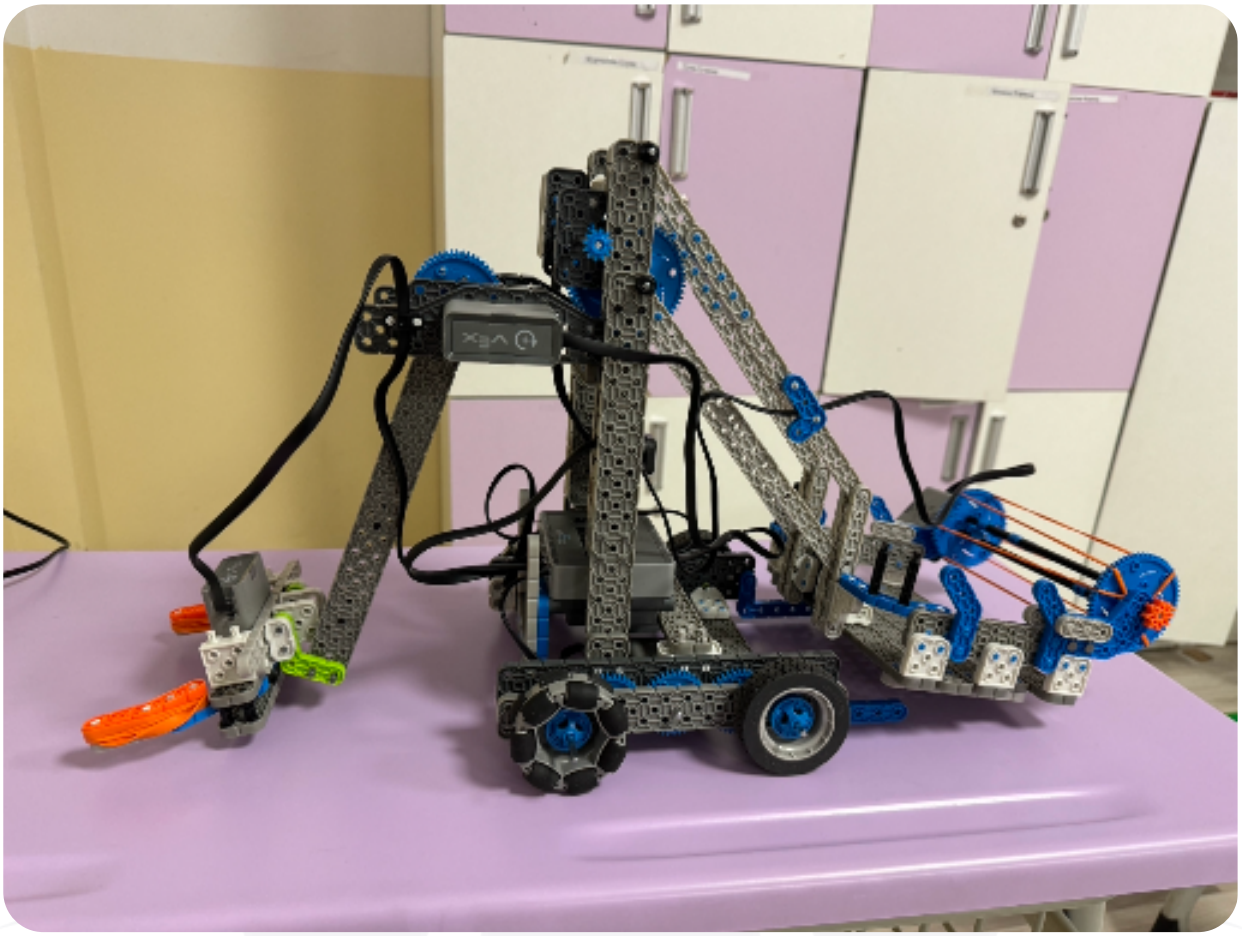
ROBOT-TERA



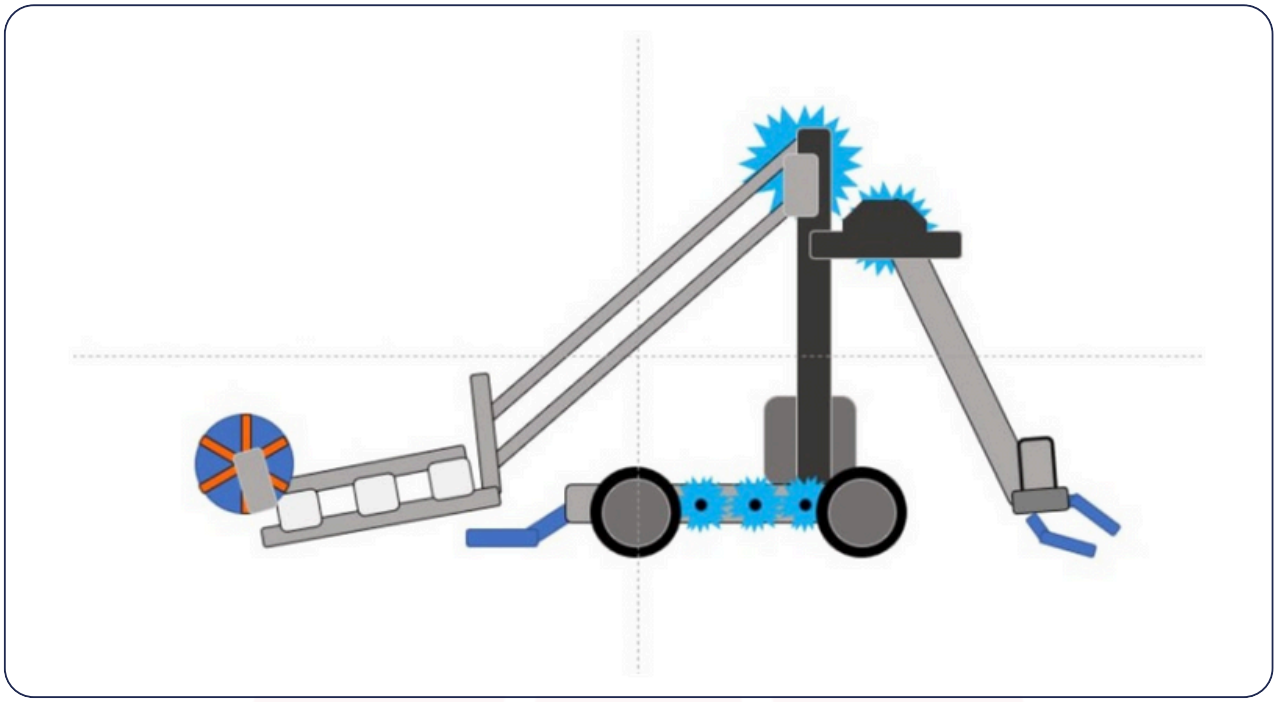
Description of the Design

1. 6 engines from the VEX IQ Generation 2 set
2. (2 motors - transmission, one motor for raising and lowering the robotic arm, 1 motor for grabbing water bottles, 1 motor for rotating the mechanism for grabbing cubes and tennis balls and another motor for lifting the basket with balls and cubes)
3. Microcontroller from the VEX IQ Generation 2
4. Kit control panel from the VEX IQ Generation 2
5. Set As well as parts from the VEX IQ Generation 2 kit for assembling the robot

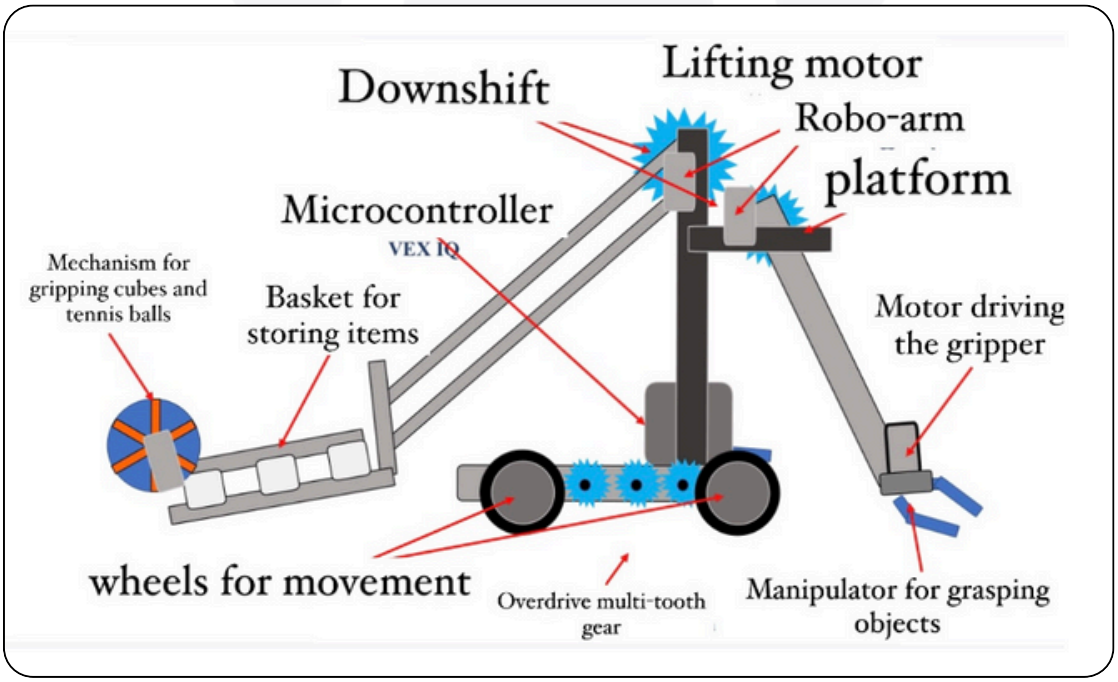
Robot Information
System: Vex IQ Gen 2
Cost: # of motors :6

Robot dimensions:
Height: 39cm
Width: 45cm
Depth: 31 cm

View from the bottom



- Manipulator for grasping objects
- Overdrive multi tooth gear
- Wheels for movement
- Bascer for storing items and mechanism for grapping cubes and tennis balls
- Microcontroller
- Downshift
- Lightning motor
- Robo arm
- Platform
- Motor driving the gripper



Photos of the Designing & Operating

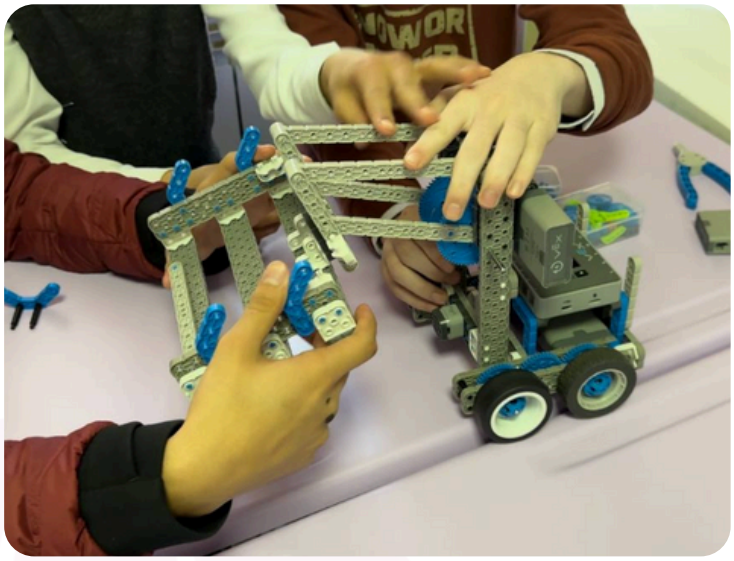
1. Building a robotic transmission with multi-gear overdrive preparing a place for the microcontroller on the top of the wheels

2. We stick the microcontroller

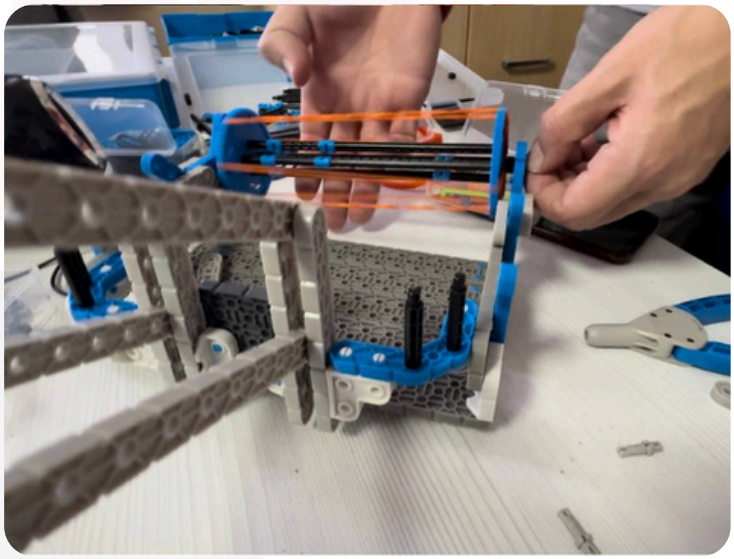
3. Preparing a mount for a basket with storage of objects

Photos of the Designing & Operating

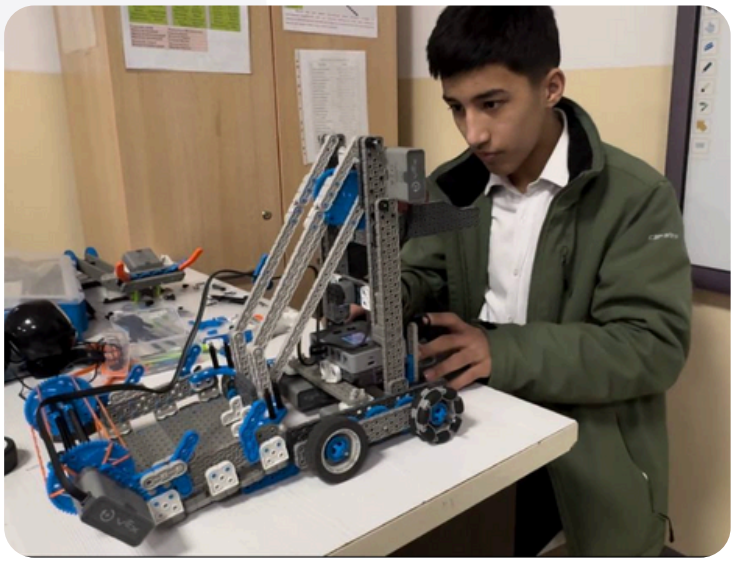
4. Installing a basket for cubes and tennis balls



5. Installing a mechanism for collecting cubes and balls. We are testing whether the claw will lift the load and whether it has enough strength, we came to the conclusion that for greater grip with the cube, it is necessary to use elastic bands on the claw

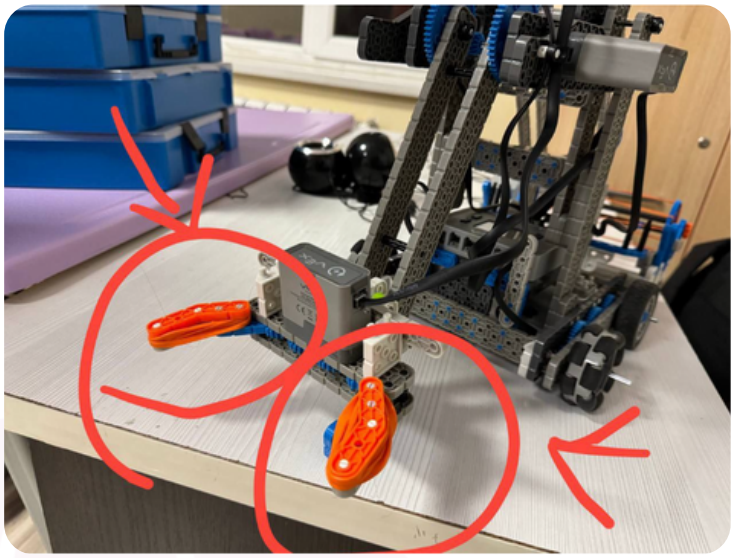


6. We are testing whether the claw will lift the load and whether it has enough strength, we came to the conclusion that for greater grip with the cube, it is necessary to use elastic bands on the claw

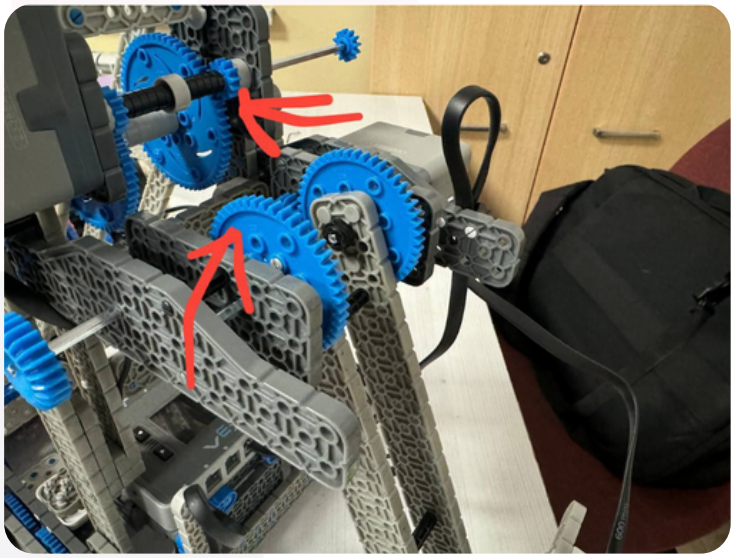


Photos of the Designing & Operating

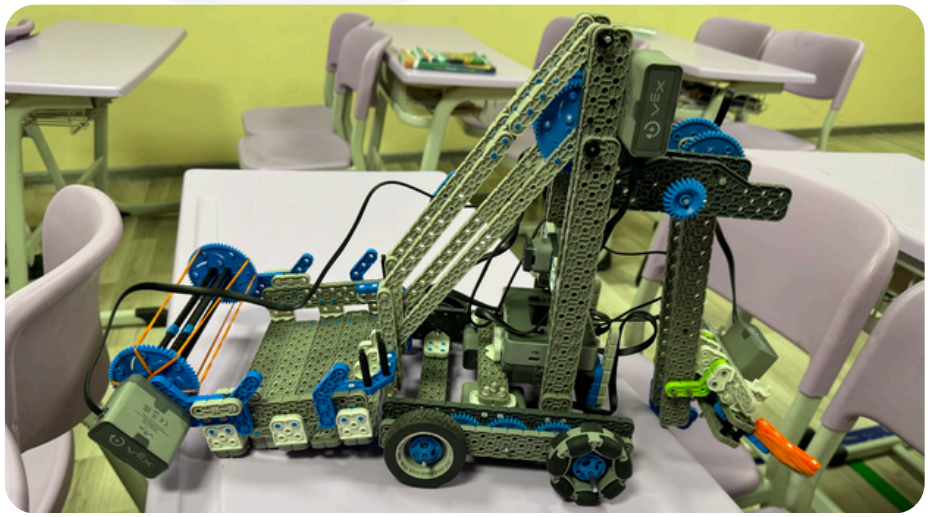
7. Installed a rear manipulator with a large number of rubber bands for gripping bottles.



8. Reduction gears were used to lift loads.

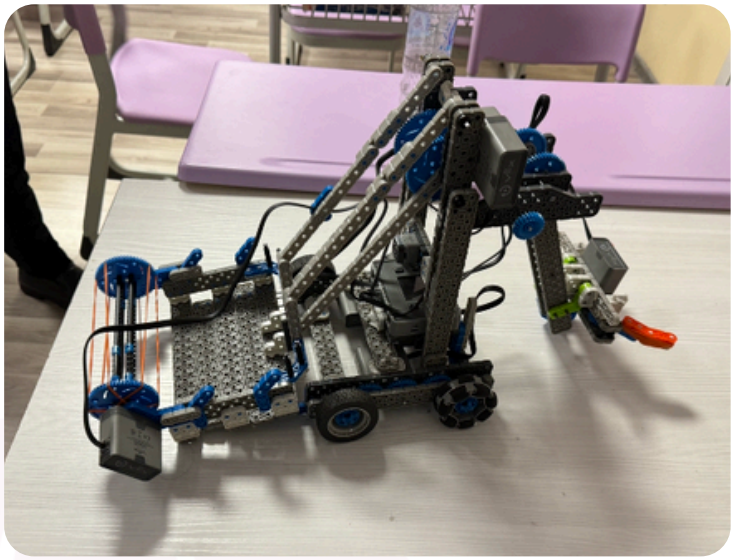


OUR ROBOT



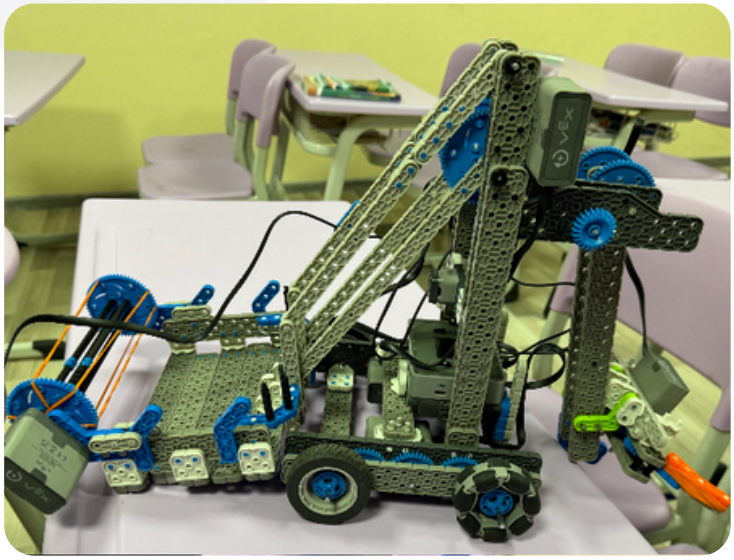
Description of Robot by Photos

1. Robot from the top



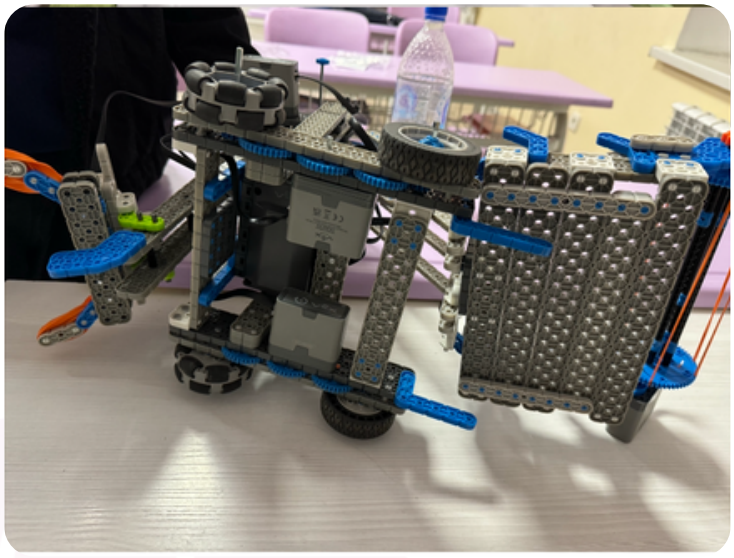
2. From the right

3. From the left

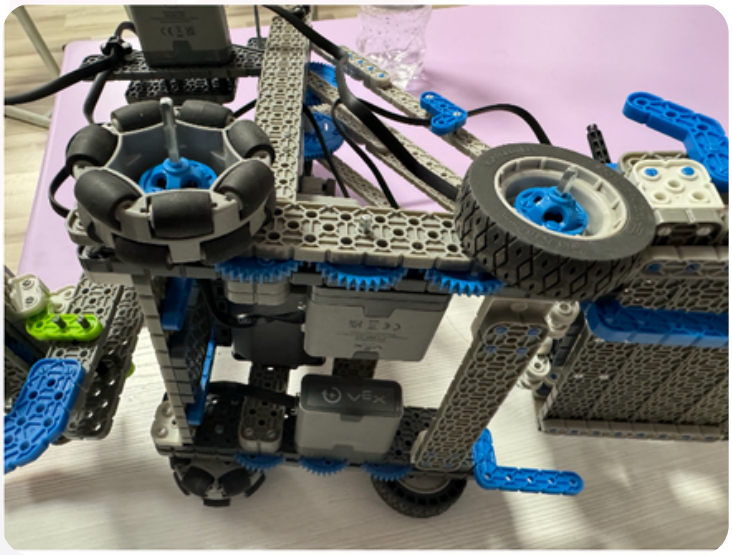


Description of Robot by Photos

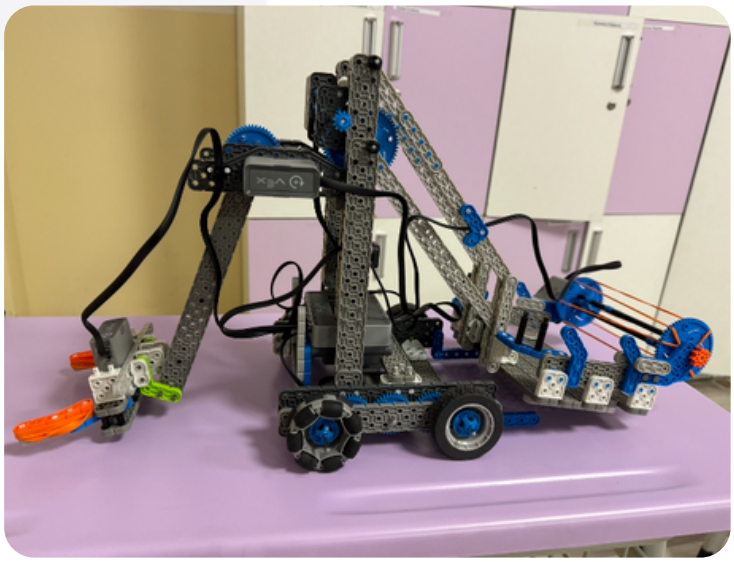
4. Mechanism of the wheels



5. Robot from below

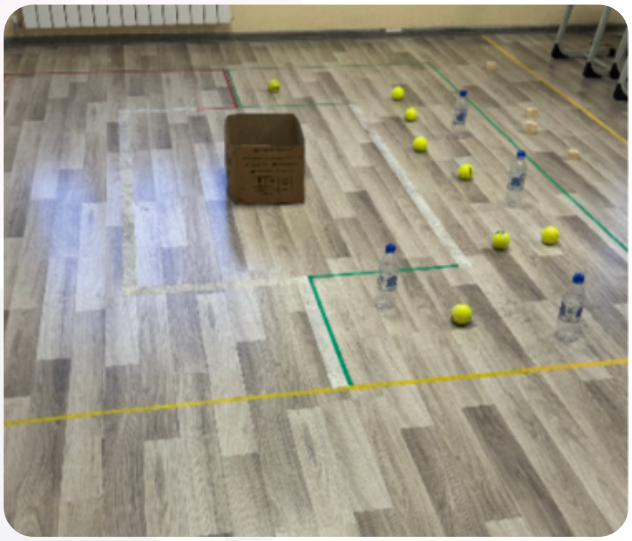


6. Robot arm



Making the Field for Practicing



```
def when_started1():
  global myVariable
  while True:
    drivetrain.set_drive_velocity(100, PERCENT)
    if controller.axisA.position() > 50:
      drivetrain.drive(FORWARD)
    if -50 > controller.axisA.position():
      drivetrain.drive(REVERSE)
    if controller.axisC.position() > 50:
      drivetrain.turn(RIGHT)
    if -50 > controller.axisC.position():
      drivetrain.turn(LEFT)
    if controller.axisA.position() == 0 and controller.axisC.position() == 0:
      drivetrain.stop()
    wait(20, MSEC)
```

```
def onevent_controllerbuttonLUp_pressed_0():
  global myVariable
  motor_4.set_velocity(100, PERCENT)
  while not controller.buttonLUp.pressing():
    motor_4.spin(FORWARD)
    wait(20, MSEC)
  motor_4.stop()

def onevent_controllerbuttonLDown_pressed_0():
  global myVariable
  motor_4.set_velocity(100, PERCENT)
  while not controller.buttonLDown.pressing():
    motor_4.spin(REVERSE)
    wait(20, MSEC)
  motor_4.stop()

def onevent_controllerbuttonRUp_pressed_0():
  global myVariable
  motor_4.set_velocity(100, PERCENT)
  while not controller.buttonRUp.pressing():
    motor_10.spin(FORWARD)
    wait(20, MSEC)
  motor_10.stop()

def onevent_controllerbuttonRDown_pressed_0():
  global myVariable
  motor_4.set_velocity(100, PERCENT)
  while not controller.buttonRDown.pressing():
    motor_10.spin(REVERSE)
    wait(20, MSEC)
  motor_10.stop()

def onevent_controllerbuttonEUp_pressed_0():
  global myVariable
  motor_5.set_velocity(100, PERCENT)
  while not controller.buttonEUp.pressing():
    motor_5.spin(FORWARD)
    wait(20, MSEC)
  motor_5.stop()
```

```
def onevent_controllerbuttonEDown_pressed_0():
  global myVariable
  motor_5.set_velocity(100, PERCENT)
  while not controller.buttonEDown.pressing():
    motor_5.spin(REVERSE)
    wait(20, MSEC)
  motor_5.stop()

def onevent_controllerbuttonFUp_pressed_0():
  global myVariable
  motor_11.set_velocity(100, PERCENT)
  while not controller.buttonFUp.pressing():motor_11.spin(FORWARD)
  wait(20, MSEC)
motor_11.stop()

def onevent_controllerbuttonFDown_pressed_0():
  global myVariable
  motor_11.set_velocity(100, PERCENT)
  while not controller.buttonFDown.pressing():
    motor_11.spin(REVERSE)
    wait(20, MSEC)
  motor_11.stop()

# Calibrate the Drivetrain Gyro
calibrate_drivetrain()

# system event handlers
controller.buttonLUp.pressed(onevent_controllerbuttonLUp_pressed_0)
controller.buttonLDown.pressed(onevent_controllerbuttonLDown_pressed_0)
controller.buttonRUp.pressed(onevent_controllerbuttonRUp_pressed_0)
controller.buttonRDown.pressed(onevent_controllerbuttonRDown_pressed_0)
controller.buttonEUp.pressed(onevent_controllerbuttonEUp_pressed_0)
controller.buttonEDown.pressed(onevent_controllerbuttonEDown_pressed_0)
controller.buttonFUp.pressed(onevent_controllerbuttonFUp_pressed_0)
controller.buttonFDown.pressed(onevent_controllerbuttonFDown_pressed_0)
# add 15ms delay to make sure events are registered correctly.
wait(15, MSEC)

when_started1()
```