Challenge

Students will design, program, and build a stationary robotic arm that can pick up a ball and drop it into a cup. The robot must place a ball into cups placed at different spots within the range of the arm.

Materials Needed

Each pair of students will create one robot.

- Use one of these sets:
  - TETRIX® PRIME Programmable Robotics Set (44321)
  - TETRIX PRIME Dual-Control Robotics Set (44322)
- Yellow balls (from robotics set)
- Red cups (from robotics set)
- Painter's tape
- Engineering logbook

Objectives

By the end of the lesson, students will be able to:

- Design and build a challenge field.
- Build a robot within the constraints to meet the challenge.
- Write the steps and create a program for the robot that meets the challenge.
- Test and refine the robot program and design.
- Demonstrate the effectiveness of the robot to meet the challenge.
- Reflect on and share the challenge and its real-world applications.

Activity

Robotic Arm Challenge

Difficulty

Intermediate

Class Time

Six or more 45-minute class periods

Grade Level

- Middle school
- High school

Learning Focus

- Engineering problem-solving
- Robot assembly
- Computer science
- Logistics

Step 1: Introduce (15 minutes)

- Share, define, and refine the challenge. Document this information in the engineering logbook.
- Write the challenge in your own words. Record the constraints you should follow, the materials that can be used for the solution, and what the testing field will look like. Discuss the constraints and materials that are allowed.

Step 2: Brainstorm (30 minutes)

- Brainstorm ideas to solve the challenge. Create quick sketches and describe solutions to the challenge.
- Considerations for your design:
  - What is the maximum range of the robotic arm?
  - How will the robotic arm swivel to place the ball in different locations?
  - How will the robotic arm pick up the ball?

Step 3: Set Up (15 minutes)

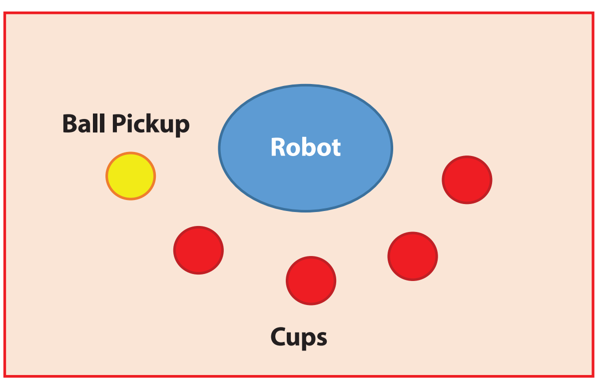
- Build the challenge field following the pictured guide.
  - Determine the range of the arm on your robot and place the cups within range in four different positions.
  - Place tape beneath each cup to mark its location. Also indicate the location where your robot will be placed and where the ball pickup will occur. The base of the robot shouldn't move from its location.

Constraints

The team's robot must:

- Contain parts from only one set.
- Have a robotic arm.
- Have a stationary base.
- Have an arm that can:
  - Move up and down to pick up and drop balls.
  - Swivel to different positions.

Possible Challenge Field



Step 4: Plan (30 minutes)

- Before building, think about the potential design of the robot and draw or record ideas in the engineering logbook. Consider the following:
  - The use of a servo to change the position of the robotic arm
  - The use of the gripper kit to pick up and release the ball
  - Location and orientation of sensor(s), if any
- Create a detailed sketch of your selected solution to the challenge. Label the materials you will use. Write a detailed description of how your solution meets the challenge, constraints, and criteria.

Step 5: Create (45 minutes)

- Design and build the robot. Remember to update the solution in the engineering logbook as the design is improved.
  - **Note:** The creation of the robot could take longer depending on the complexity of the robot solution.

Step 6: Write the Steps (15 minutes)

- Think through the steps or series of actions that the robot will have to complete to meet the challenge. Planning this series of steps is sometimes referred to as creating pseudocode for your robot.
  - Record these steps in the engineering logbook and use them as a guide when operating the robot. Notice that the steps are like writing code for the robot to follow. Make sure the robot performs all the steps required in the challenge.

Step 7: Create the Program (45 minutes)

- When you have completed this process, you are ready to begin programming using your steps as a guide. Remember to track changes in the engineering logbook.
  - When you are coding your robot, it is recommended that you write the code using functions so that each task can be tested and adjusted before it is incorporated into the final program.
- Prepare functions to control your robot, depending upon your solution plan.
- Check each of your functions as you write it to make sure it works as you intend.
- Now, write a test sketch to try them all out.

Step 8: Test (45 minutes)

- Test the solution. Place the robot into the challenge field and press the Start button to execute the code.
- Refine the solution. Adjust the design and code as needed. Document any changes in the engineering logbook.

Sample Steps

1. Move the robotic arm to position above a ball.
2. Open the gripper.
3. Drop the robotic arm to the ball.
4. Close the gripper.
5. Raise the robotic arm.
6. Move the robotic arm to position above a cup.
7. Open the gripper and release the ball into the cup.
8. Repeat the steps until a ball is placed into each cup.
9. Celebrate!

Step 9: Demonstrate (15 minutes)

- When the robot has been tested and successfully operates the program, demonstrate its performance in a final test.

Step 10: Reflect and Share (15 minutes)

- Look back at the prototype. How does it compare to the final design?
- Look back at the original steps. How do they compare to the final steps?
- Discuss the original prototype, the final robot code, the solution as implemented, and how this challenge applies to the real world of robot design and programming.

Step 11: Extensions

- Go Mobile
    - Transform your robot from stationary to mobile. Program the robot to drive to a ball, pick it up, and place it into a cup. Set up a path for the robot to follow and pick up balls and place them into cups.
- Go Full Circle
    - Place the cups around the robot so that it has to swivel around 360 degrees to place the balls in the cups. Adjust your build accordingly to allow for a full range of movement. Update your programming to reflect the new positions of the cups.

To get you started, here are some sample blocks for the PULSE™ controller with the *TETRIX Ardublockly* software.

You could also incorporate the use of functions to name a series of programming steps such as open gripper, close gripper, move arm up, move arm down, swivel arm to a position, and so on.

Here are a couple of examples of functions with programming blocks for their actions: