## Challenge

Students will design, program, and build a robot that works with one other robot to compete in an obstacle robot relay race without dropping a baton.

## Materials Needed

Each pair of students will create one robot. You will work with another pair as a team to compete in the relay race.

- Use one of these sets:
  - TETRIX® PRIME Programmable Robotics Set (44321)
  - TETRIX PRIME Dual-Control Robotics Set (44322)
- Items to create challenge field: painter's tape, objects to represent the baton
- 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

Relay Race 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

### **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:
  - Your robot needs the ability to grab a baton and let go.
  - Your robot should stay within its lane during the relay.
  - Your robot can use the Line Finder Sensor to follow lines and detect messages.
  - Notice that Robot 1 and Robot 2 have different programming requirements.
  - Make sure your program does not require an exact distance to run before coming to a line, because the distances might change.
  - Robot 2 must sense where Robot 1 left the baton to be picked up.
  - It is not OK for Robot 1 to carry the baton on its first run.
  - Robot 1 must sense where to pick up the baton to take it down the track.
  - Rules of play:
    - Robot 1 picks up the baton, goes down to Line 1, drops the baton, and returns, tagging its partner. Tagging occurs when the robot arm touches the other robot.
    - Robot 2 goes down to Line 1, picks up the baton, drops it at Line 2, and returns to the start line.
    - When Robot 2 tags Robot 1, Robot 1 goes to Line 2, picks up the baton, and drops it off at Line 1.
      - Robot 1 then returns to the start line and tags Robot 2.
    - Robot 2 then goes to Line 1 and picks up the baton and brings it back to the start line.
    - The first team to cross the start line with its baton wins.
  - Penalties:
    - Starting before being tagged means the team must start over.
    - Losing the baton means that robot must start over.
    - Straying out of bounds or across into the other team's lane means the team must start over.

#### Constraints

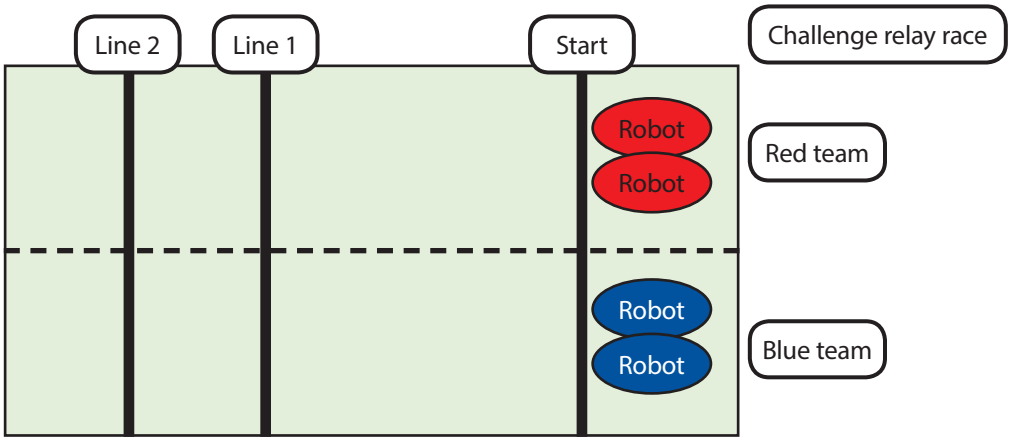
The team's robots must:

- Contain parts from only one set.
- Measure less than 30 cm x 40 cm x 40 cm.
- Use a Line Finder Sensor to detect a line and detect messages from another robot.
- Send messages to another robot's Line Finder Sensor by alternately holding and removing a black flag in front of the other robot's sensor.
- Pick up and drop off a small object.

### **Step 3: Set Up** (15 minutes)

- Build the challenge field following the pictured guide.
- The field should be about 1 m x 3 m with a start line and two goal lines in colors that contrast with the field color. See the sample diagram.
- Have a small, lightweight object to represent a baton, such as a 5 cm cube, for each team.

**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:
  - Drivetrain for speed and control
  - Robot chassis for size
  - Movement that will signify a tag
  - Baton placement for pickup by Robot 1 and Robot 2
  - Location and orientation of the Line Finder Sensor
  - The way your robot will both make the tag and look for the line
    - Is the arm going to move, keeping the sensor in one place?
    - Will the sensor move?
    - Can you think of a way to do it without moving either?
- 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.

### Step 9: Demonstrate (15 minutes)

- When the robot has been tested and successfully navigates the challenge field, 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

- Design a relay race that follows a different sequence of steps. Each robot could perform a different action before the baton is passed. Robots could be required to avoid obstacles along the relay path or form a larger relay team.

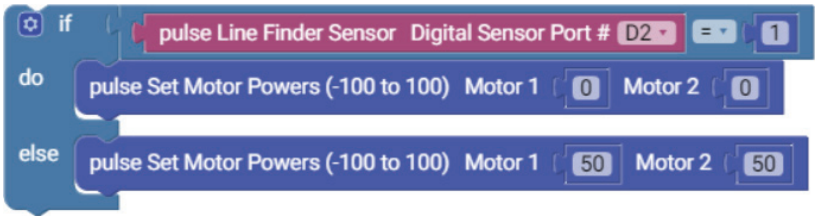
### Robot 1 Sample Steps

1. Pick up the baton at the start line.
2. Go forward until I see Line 1.
3. Drop off the baton.
4. Return to the start line.
5. Tag my partner.
6. Wait to be tagged by my partner.
7. Go forward until I see Line 2.
8. Pick up the baton.
9. Return to Line 1.
10. Drop off the baton.
11. Return to the start line.
12. Tag my partner.
13. Wait until my partner returns.
14. Celebrate.

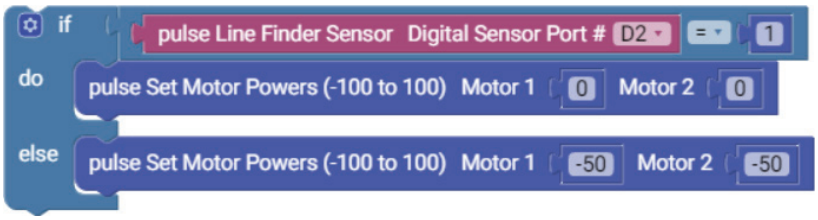
### Robot 2 Sample Steps

1. Wait to be tagged by my partner.
2. Pick up the baton at Line 1.
3. Go forward until I see Line 2.
4. Drop off the baton.
5. Go backward until I see the start line.
6. Tag my partner.
7. Wait to be tagged by my partner.
8. Go forward to Line 1.
9. Pick up the baton.
10. Return to the start line.
11. Celebrate.

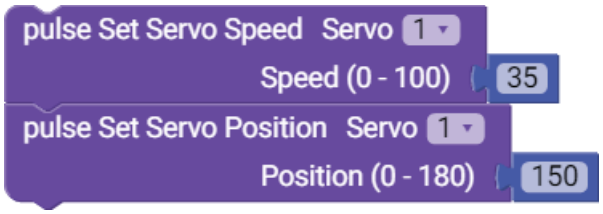
To get you started, here are some sample blocks for the PULSE™ controller with the *TETRIX Ardublockly* software.



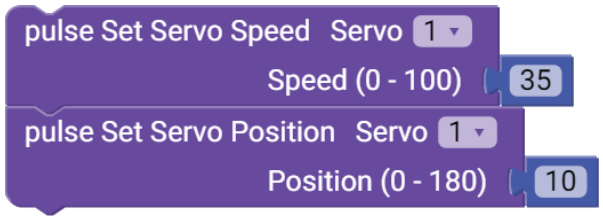

The robot will drive forward until it detects a black line and then stop.




The robot will drive backward until it detects a line and then stop.




Open gripper




Close gripper