Challenge

Students will design, program, and build a robot that communicates with Morse code. The robot must use its communication system to tell the operator when the robot completes each task in a series.

Materials Needed

- 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, blocks or books of various sizes, lightweight objects
- 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 Morse code communication functions using the robot's LED.
- Have the robot use the communication system to tell the operator when the robot completes each task in a series.
- 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 and share on the challenge and its real-world applications.

Activity

Morse Code Challenge

Difficulty

Intermediate

Class Time

Six 45-minute class periods

Grade Level

- Middle school
- High school

Learning Focus

- Engineering problem-solving
- Robot assembly
- Computer science
- Runtime error signaling
- Social responsibility and community norms
- Morse code communication

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 set of tasks does a standard-type vehicle robot perform?
  - The robot doesn't have to pick up the object to move it; the object may merely be pushed or pulled to its drop location.
  - This challenge is designed to be relatively easy. For example, notice there is no restriction on the size of the catcher that could be used to make sure the object is picked up. Focus on communicating the Morse code correctly.
  - Remember that accurate turns are best accomplished at a very low power (slow speed) to avoid skidding of the wheels.

Step 3: Set Up (15 minutes)

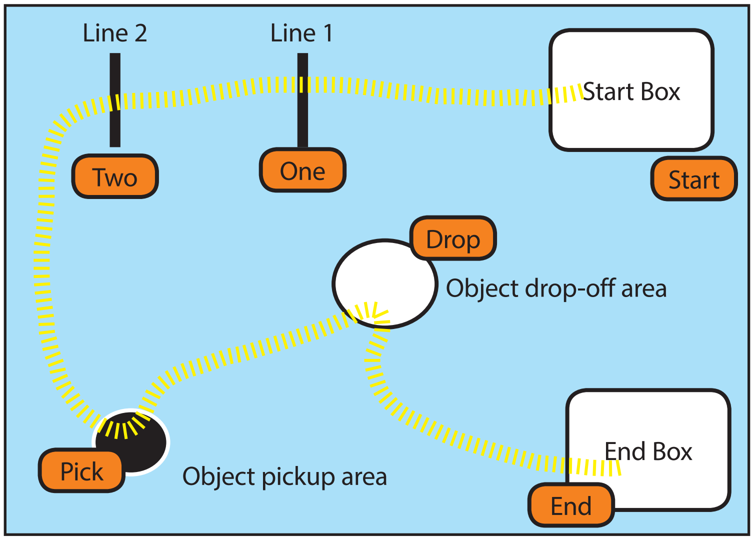
- Build the challenge field following the pictured guide.
  - The field should be about 2 m by 2 m with a start box and an end box, two black tape lines to stop at, an object pickup area, and an object drop-off area.
  - Have one small lightweight object. It starts in the center of the object pickup area and is to be moved to the object drop-off area.
  - Suggested signal words are given on the sample challenge field in orange squares.

Constraints

The team's robot must:

- Contain parts from only one set.
- Have a Line Finder Sensor in a position to detect a line.
- Have an Ultrasonic Sensor in a position to detect predetermined distances.
- Be able to pick up and drop off a small object.
- Signal completion of each task via Morse code before moving on to the next task.

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
    - Location of the LED so it can be easily seen
    - Location and orientation of the Line Finder Sensor
    - Location and orientation of the Ultrasonic Sensor
    - Size, shape, and movement of the arm that will move the object
- 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 function 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 onto 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. Signal START.
2. Go forward until I see a black line.
3. Signal ONE.
4. Go forward until I see a black line.
5. Signal TWO.
6. Make a left turn to head for the object pickup area.
7. Locate the object and stop there.
8. Pick up the object or collect it in a catcher.
9. Signal PICK.
10. Make a left turn to aim for the object drop-off area.
11. Travel far enough to get to the object drop-off area.
12. Drop off the object.
13. Signal DROP.
14. Go backward a short distance so I don’t bump the object.
15. Turn right to aim for the end box.
16. Travel far enough to get into the end box.
17. Signal END.
18. Celebrate.

Step 9: Demonstrate (15 minutes)

- When the robot has been tested and successfully navigates the challenge field, demonstrate its performance in a final test.

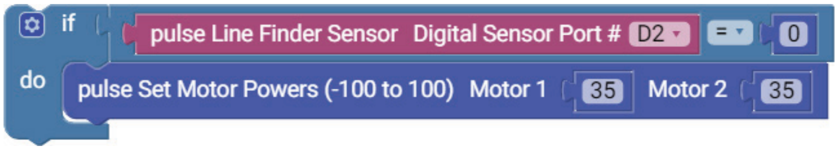
Step 10: Reflect and Share (15 minutes)

- 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

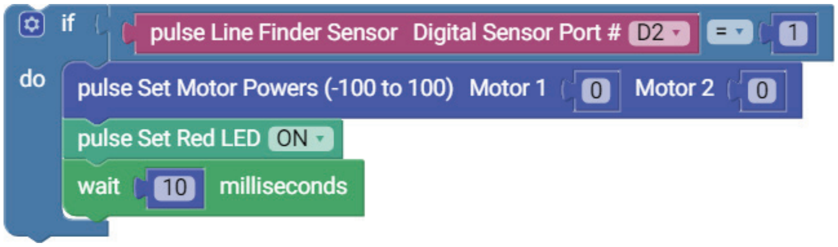
- Morse Code Parser
    - Increase the flexibility of your robot's Morse code ability by creating a function for each letter in the alphabet and then creating a function that interprets any word given to it and uses the letter functions to flash the Morse code for the word, sentence, or paragraph.
- Secret Passage
    - Write code so that when a person comes to a locked door, he or she enters a password into the robot's Line Finder Sensor by flashing a series of black and white cards in a specific order and length of time. Then, the robot flashes the person's name in Morse code to those watching on the other side of the door, who then open the door. If the password is wrong, then the robot flashes a warning signal to those behind the door.

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



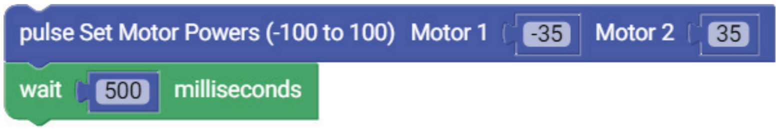
```
if  pulse Line Finder Sensor  Digital Sensor Port # D2  =  0
do  pulse Set Motor Powers (-100 to 100)  Motor 1  35  Motor 2  35
```

In this set of blocks, the motors will drive forward as long as the Line Finder Sensor doesn't detect a black line.



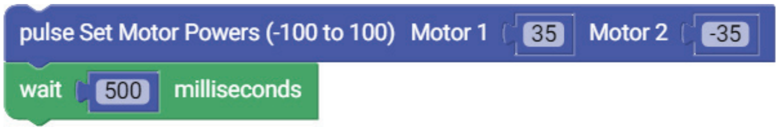
```
if  pulse Line Finder Sensor  Digital Sensor Port # D2  =  1
do  pulse Set Motor Powers (-100 to 100)  Motor 1  0  Motor 2  0
    pulse Set Red LED  ON
    wait  10  milliseconds
```

In this set of blocks, the motors will stop when the Line Finder Sensor detects a black line. Then, the LED is turned on for 10 ms. This is where you can insert the LED patterns for the Morse code you need to signal.



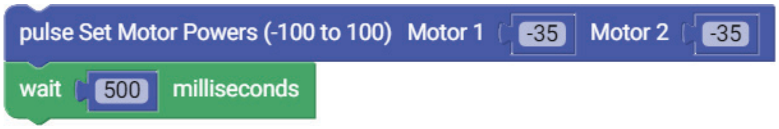
```
pulse Set Motor Powers (-100 to 100)  Motor 1  -35  Motor 2  35
wait  500  milliseconds
```

Turn left



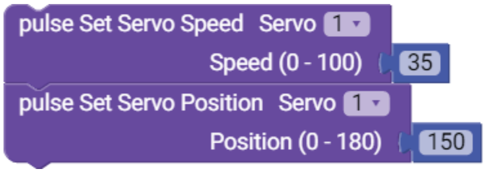
```
pulse Set Motor Powers (-100 to 100)  Motor 1  35  Motor 2  -35
wait  500  milliseconds
```

Turn right



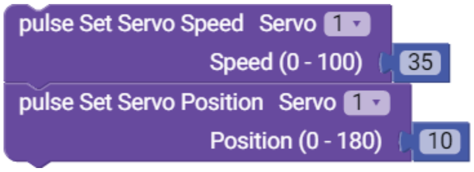
```
pulse Set Motor Powers (-100 to 100)  Motor 1  -35  Motor 2  -35
wait  500  milliseconds
```

Go backward



```
pulse Set Servo Speed  Servo  1
                 Speed (0 - 100)  35
pulse Set Servo Position  Servo  1
                 Position (0 - 180)  150
```

Open gripper



```
pulse Set Servo Speed  Servo  1
                 Speed (0 - 100)  35
pulse Set Servo Position  Servo  1
                 Position (0 - 180)  10
```

Close gripper