Challenge

Students will design, build, and program a robot that can approach an object and grab it using a tactile gripper. The gripper should apply enough force to hold the object without crushing it. Tactile sensors embedded in the gripper fingers tell the robot when to stop squeezing the object. The robot must then transport the object to a drop zone and release the object. Finally, the robot must return to the starting location.

Materials Needed

Each pair of students will create one robot.

- Use one of these sets:
  - TETRIX® MAX Programmable Robotics Set (43053)
  - TETRIX MAX Dual-Control Robotics Set (43054)
- At least two tactile sensors. Examples include:
  - Grove Micro Switches: https://www.seeedstudio.com/Grove-Micro-Switch.html
  - Grove Button(P)s: https://www.seeedstudio.com/Grove-Button-P.html
  - Grove LED Buttons: https://www.seeedstudio.com/Grove-Red-LED-Button.html
- Rubber bands or springs
- Items to create challenge field:
  - Painter's tape
  - Meterstick or tape measure
  - Items for grabbing that are soft and delicate. Examples include:
    - Small paper cups
    - Empty plastic water bottles
    - Sponges
    - Small balloons
    - Toilet paper tubes
    - Paper cylinders
- Computer with the following installed:
  - Arduino Software (IDE) with Arduino libraries for TETRIX PRIZM® and for TETRIX Tactile Sensor
    - Both libraries can be downloaded from the Pitsco website at: https://www.pitsco.com/TETRIX-PRIZM-Robotics-Controller
- Engineering logbook

Objectives

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

- Build a challenge field.
- Write the steps for the robot to follow to complete the challenge.
- Design, build, and program a robot to meet the criteria and constraints of the challenge.
- Test, analyze, and refine the robot to improve its performance.
- Demonstrate the robot's effectiveness at meeting the challenge.
- Reflect on and discuss the challenge including its real-world applications.

Activity

Tactile Gripper Challenge

Difficulty

Intermediate

Class Time

Five or more 45-minute class periods

Grade Level

- Middle school
- High school

Learning Focus

- Engineering problem-solving
- Programming/coding
- Digital sensors

Step 1: Determine the Challenge and Specifications (15 minutes)

- Share, define, and refine the challenge. Ask questions to help you get a clear understanding of 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.

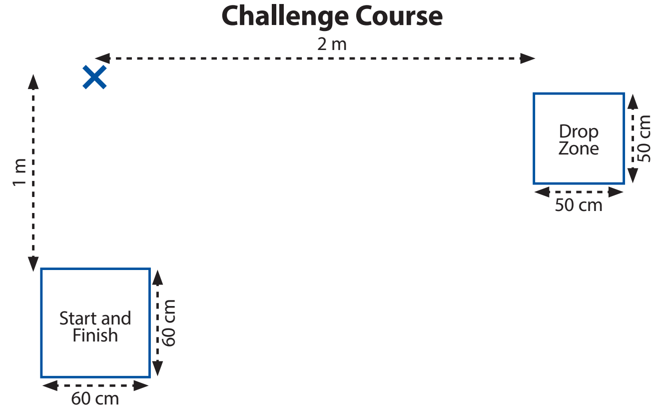
Step 2: Brainstorm Solutions (30 minutes)

- Brainstorm ideas to solve the challenge. Think of as many possible solutions as you can in the given time frame.
- Considerations for your design:
    - How can tactile sensors be attached to the fingers of a gripper to sense when an object is grabbed?
    - How can you create a bumper mechanism that presses against the tactile switch when an object is grabbed rather than using the sensor to grab an object?
    - What items can you use to spring-load the bumpers so that the bumpers don't press the tactile sensors when an object isn't grabbed?
    - How will the tactile sensors be used to control the movement of the gripper?
    - How many servos will be used in your gripper mechanism?
    - Can a line finder sensor or ultrasonic sensor aid the robot in completing the challenge?

Step 3: Set Up (15 minutes)

- Build the challenge field:
    1. Use painter's tape to create a 60 cm x 60 cm starting square for the robot. The wheels of the robot must start and finish inside this square to complete the challenge.
    2. Create a small X with painter's tape one meter from the starting square. This is where the object to be transported will be placed. Depending on the size of the object, you might need to build a platform to raise the object off the ground a bit to make the object easier for the robot to grab.
    3. Create a 50 cm x 50 cm square with painter's tape two meters to one side of the object. This will act as the drop zone for the object.

Challenge Course



Criteria and Constraints

The team's robots must:

- Other than the additional items listed previously, utilize parts from only one set.
- Contain no bent, cut, or broken pieces.
- Be mobile and able to transport an object without damaging it.
- Be able to grab an object without damaging it.
- Incorporate at least two tactile sensors in the gripper mechanism to limit the crushing force of the gripper.
- Include bumper mechanisms on the gripper fingers that press against the tactile sensors when an object is gripped rather than using the sensors themselves to grab an object.
- Be completely autonomous with no human intervention.

Step 4: Formulate a Solution (45 minutes)

- Consider the ideas you brainstormed in Step 2. Which of these ideas do you think will have the most success on the challenge field?
- Turn your best ideas into a design for your robot.
- In your engineering logbook:
  - Create a detailed sketch of your chosen solution to the challenge.
  - List materials you will use.
  - Write a detailed description of how your solution meets the challenge criteria and constraints.

Step 5: Prototype the Solution (135 minutes)

- Build the robot according to the designs you created in Step 4. If you modify the design as you build your robot prototype, remember to change the design in your engineering logbook.
  - **Note:** The creation of the robot could take longer depending on the complexity of the robot solution.

Step 6: Develop a Process (15 minutes)

- Robotic challenges often require robots to complete a series of tasks in a certain order. This series of steps is called a process. Think through the process your robot needs to complete to be successful in the challenge. Planning this series of steps is sometimes referred to as creating pseudocode for your robot.
  - Record your robot's process in your engineering logbook. Use this process as a guide when programming the robot to complete the challenge.

Sample Process for Gripping the Object

1. Open the gripper.
2. Drive forward to the object.
3. Close the gripper until the first tactile sensor is triggered.
4. Slow down the speed of the gripper.
5. Continue to close the gripper until the second tactile sensor is triggered.
6. Stop closing the gripper.

Step 7: Program Your Robot (15 minutes)

- When you have created your process, you are ready to begin programming using your process steps as a guide. Develop code for each of the tasks your robot must complete. Remember to record important information and 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 as needed.
- Launch the Arduino Software (IDE) and start a new sketch.
- Include the TETRIX PRIZM and TETRIX Tactile libraries in your sketch.
- Tactile sensors are digital sensors that return either a 1 or 0 depending on whether the sensor switch is pressed or not. For most tactile sensors, a 0 (false) value is returned when the button on the sensor is not pressed. When the button is pressed, a 1 (true) value is returned.
  - The TETRIX Tactile library command for reading the sensor is **getSwitch (*port*)** where port is the digital port that the sensor is connected to on PRIZM.

```
getSwitch(port);
```

- Usually, the **getSwitch()** command is used in one of two types of statements:
  - A command to set a variable to the status of the sensor

```
#include <PRIZM.h>                  // include the PRIZM controller library
#include <Tactile.h>                // include Tactile sensor library
PRIZM prizm;                        // create a class named prizm

int variableName = 0;               // create a variable to store the status of the sensor

void setup() {
  prizm.PrizmBegin();               // initialize the PRIZM controller
  Serial.begin(9600);               // start the serial monitor
}

void loop() {
  variableName = getSwitch(D2);     // store the status of the sensor connected to port D2
  Serial.print(variableName);       // print the status of the sensor
}
```

  - The conditional part of an if statement or while loop that performs a certain process when the sensor is pressed or not pressed

```
#include <PRIZM.h>                  // include the PRIZM controller library
#include <Tactile.h>                // include Tactile sensor library
PRIZM prizm;                        // create a class named prizm

void setup() {
  prizm.PrizmBegin();               // initialize the PRIZM controller
  Serial.begin(9600);               // start the serial monitor
}

void loop() {
  if(getSwitch(D2)==1){             // if the switch is pressed:
    Serial.print("Pressed");          // print the status of the sensor;
  }
  else{                             // otherwise:
    Serial.print("Not pressed");      // print the status of the sensor;
  }
}
```

- Because tactile sensors are digital, you can also use a **digitalRead(*port*)** command to read the status of a sensor instead of the **getSwitch()** command from the TETRIX Tactile library.
  - One benefit of using **digitalRead()** is that this command is a standard Arduino command and doesn't require the TETRIX Tactile library to be installed to read the sensor.
- Included with the TETRIX Tactile library are example sketches that you can use to get a better understanding of how the tactile sensors provide input to the PRIZM controller. To view these examples in the Arduino IDE, go to **File** > **Examples** > **TETRIX TACTILE** and choose an example sketch to look at.

- Convert the process that you developed in Step 6 into code in the Arduino IDE. Make sure to consider each step of the process and what commands need to be written to complete each step.
- Keep the criteria and constraints in mind as you program your robot.
- When you're finished coding, verify the code and upload the code to the PRIZM controller on your robot.

Step 8: Test and Analyze (15 minutes)

- Test your robotic solution. Place the robot in the challenge field and, using the PRIZM controller, execute the program you developed in Step 7.
- As you test your robot, record observations and data in your engineering logbook. Determine if your robot meets the requirements of the challenge.

Step 9: Redesign or Improve the Solution (45 minutes)

- Refine your challenge solution. Adjust the robot design, process, and program as needed. Document changes in the engineering logbook.
- Make the physical changes to your prototype robot according to your design modifications.
- Continue to modify the robot's design and program until it can successfully complete the challenge.

Step 10: Demonstrate (15 minutes)

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

Step 11: Reflect and Share (15 minutes)

- Reflect on the changes your robot went through from the original idea to final design.
- Reflect on the results of the challenge. What elements of your robot design brought you success or failure?
- Discuss the roles and responsibilities each team member fulfilled. How did teamwork and collaboration help you complete the challenge?
- Discuss how this challenge relates to robot design in the real world.

Step 12: Extensions

- Multiple Objects
  - Add more objects to the challenge field and program the robot to retrieve each object and drop them in the drop zone.
- Sorting Hard and Soft Objects
  - Use the gripper to distinguish hard and soft objects. Program the gripper to squeeze an object a certain amount and determine what (if any) tactile sensors are triggered. Place hard objects to one side and soft objects to the other.
- Tele-Op Control
  - Attach a TETRIX Tele-Op Control Module and program your robot to accept controls from a Sony PS4 DUALSHOCK 4 controller.