Challenge

Students will design, build, and code a robot to follow or track a specified object and maintain a constant distance from said object via image recognition from the Pixy2 camera integrated with the TETRIX® PRIZM® controller.

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)
- Pixy2 camera with TETRIX Pixycam Adapter
- Items to create challenge field:
  - At least 2 colored balls at least as big as a golf ball (can be any solid-colored balls that are not white; primary colors are preferable)
  - Painter's tape
  - Metric tape measure or meterstick
- Engineering logbook
- Computer with the following software installed:
  - Arduino Software (IDE) and Arduino libraries for TETRIX PRIZM and Pixy2
    - In order to get all example code, including code created for use with PRIZM and TETRIX, it is recommended you download the Arduino Pixy2 library here: https://www.pitsco.com/TETRIX-PRIZM-Robotics-Controller
  - PixyMon software
    - Download here: https://pixycam.com/downloads-pixy2/

Note of encouragement: Expanding your skill set as a robotics user with unfamiliar hardware or new programming concepts can certainly be daunting! It's perfectly acceptable and even recommended as proper technique to look at sample code to get a feel for how "good" code should look and execute. Please reference the example code found in the TETRIX_Pixy2_Camera library as a starting point for your own code. You can also use the examples as a point of reference to confirm the proper functionality of your hardware. When using provided examples or sample code, always make sure to note the hardware configuration outlined in the code comments to ensure actual robot behavior matches expectations.

Objectives

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

- Build an autonomous robot with TETRIX PRIZM as the controller and the Pixy2 camera mounted as a sensor.
- Train the Pixy2 camera with PixyMon software to recognize a colored object and assign it a unique signature identification.
- Access the information processed by Pixy2 within the Arduino programming environment and utilize it to program autonomous robot behavior.
- Test, analyze, and refine the robot to optimize its performance.
- Demonstrate the robot's ability to use the information from the Pixy2 camera to successfully track or follow a specific object autonomously.
- Reflect on and discuss the challenge including its real-world applications.

Activity

Stuck on You like Glue

Difficulty

Intermediate

Class Time

Seven or more 45-minute class periods

Grade Level

- Middle school
- High school

Learning Focus

- Engineering design
- Robot assembly
- Coding
- Integration of image acquisition and analysis with robot behavior

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

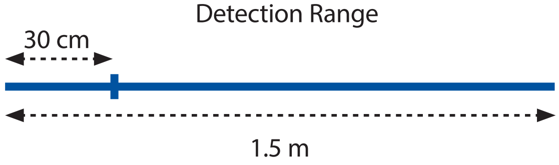
- Share, define, and refine the challenge. Ask questions to help you get a clear understanding of the challenge. Document this information in the engineering logbook.
- Write the challenge in your own words. Record the constraints you should follow, the materials that can be used as part of the solution, and what the final test of a successful solution would be. Discuss the constraints and materials that are allowed.

Step 2: Brainstorm Solutions (30 minutes)

- Brainstorm ideas to solve the challenge. Create quick sketches to help describe or explain potential solutions to the challenge. Think of multiple solutions and discuss merits of each in the given time frame.
- Considerations for your design:
  - Where will you mount the camera? How will its position impact the field of view?
  - What are the lighting conditions? Do they impact the camera's mounting position?
  - How do mobility, turning radius, and speed all play a part in how successful the robot can track a moving object?

Step 3: Set Up (15 minutes)

- Examine the objects to be tracked.
  - Observe and note any and all distinguishing features that would help identify the object (for example, size, shape, and color).
  - Use the painter's tape and measuring device to create detection range parameters by laying out a 1.5 m strip of tape on floor with a mark 30 cm from one end.



Step 4: Formulate a Solution (30 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.

Criteria and Constraints

The team's robots must:

- Beyond the Pixy2 camera, utilize parts from only one set or as specified by the teacher.
- Be mobile and capable of following a moving object.
- Be able to detect and react to an object within a range of 30 cm to 1.5 m.
- Be able to follow and maintain a constant distance from a trackable object as long as it is within the specified range.
- Be able to distinguish between two objects of similar size and shape but of different hue or color.
- Be able to clearly exhibit autonomous following behavior without interfacing with a human controller after the program is initiated.
- Be enabled with a fail-safe programming function that prevents runaway robots.

Step 5: Prototype the Solution (90 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 the process for your robot in your engineering logbook. Use this process as a guide when operating the robot and completing the challenge.

Sample Process for Following a Ball

1. Turn on and be ready to detect the correct color ball.
2. Detect when the correct color ball comes into range.
3. Move so that the ball comes into the center of the camera's field of view.
4. Continue to move so the ball stays in the center of the camera field of view.
5. Stop if the ball suddenly is moved outside the specified range.

Step 7: Train the Pixy2 (15 minutes)

- Your process has been created, but before you can move on to the next step of actually programming your robot, you must train the Pixy2 camera to recognize an object so it can interact with the program you create.
  - With the Pixy2 mounted on your robot, connect the Pixy2 to the computer via the USB cable that comes with the Pixy2. (**Note:** Keep in mind that ambient lighting will impact vision performance. Try to train the Pixy2 in as similar lighting conditions to the final test as possible.)
  - Launch the PixyMon software and reference the Pixy2 documentation to learn exactly how to train the Pixy2 camera to recognize an object. The documentation can be found here: https://docs.pixycam.com/wiki/doku.php?id=wiki:v2:start
  - Reference the examples in the Arduino Pixy2 library to confirm the Pixy2 can communicate successfully to the PRIZM controller.

Step 8: Program Your Robot (45 minutes)

- With the process created and the Pixy2 trained, you are ready to begin programming using your process or pseudocode as a guide. Develop code for each of the tasks that your robot must complete. Remember to record important information and track changes in the engineering logbook.
  - When you are coding your robot, keep in mind coding techniques like the effective use of functions so that smaller bits of code can be tested and adjusted as needed.
- When you're finished coding, verify the code and then upload the code to the PRIZM controller on your robot.
  - **Note:** The first experience coding with new hardware or trying to create new and unfamiliar behaviors can often be daunting. It's recognized and accepted as solid technique to look at sample code to get a feel for how "good" code should look and execute. Please reference the example code provided in the TETRIX_Pixy2_Camera library as a starting point for your own code. You can also use the example code as a point of reference to confirm the proper functionality of your hardware. Make sure to note the hardware configuration outlined in the code comments to ensure actual robot behavior matches expectations.

Step 9: Test and Analyze (30 minutes)

- Test your robotic solution. Place the robot in your challenge field and, using the PRIZM controller, execute the program you developed in Step 8.
    - Does the robot properly recognize and react to the trackable ball within the specified range? Does it follow the ball and maintain a constant distance from the ball if it moves in any direction within the specified range? Does the robot react to only the correct ball and ignore any other objects? Does the robot stop if the ball is suddenly moved outside the specified range?
    - **Hint:** If the idea is to keep the ball within the center of the camera frame of view, changing the field of view by adjusting the way the camera is mounted or angled will affect the distance from the robot to the object in the center of the camera frame of view.
- As you test your robot, record observations and data in your engineering logbook. Determine if your robot meets the requirements of the challenge.

Step 10: Redesign or Improve the Solution (30 minutes)

- If necessary, refine your challenge solution. Make any adjustments to robot design or program code as needed and be sure to document any changes in your engineering logbook.
- Make any physical changes to your robot according to the design modifications determined necessary.

Step 11: Demonstrate (15 minutes)

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

Step 12: 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 13: Extensions

- Color Keyed
    - Create unique alternate behaviors when the robot detects multiple balls or multiple balls of different hues or colors.
- Conga Line
    - Attach a trackable object to a lead robot or a person and create a robot or multiple robots capable of following the leader.
- Discussion
    - Discuss what it would take to create a scanning mode where the robot actively searches for a trackable object instead of stopping and waiting for an object to come into range or view.