Seeker®
BY PITSCO EDUCATION

TEACHER'S GUIDE
UNIT 4

Unit Overview

In this Unit, Sophia and Lucas introduce students to Borealis Space Systems, a private aerospace company that builds equipment and systems that operate in space such as satellites, probes, habitats, and rovers. The tasks students complete center around adapting and programming Seeker to assist in the clean room, help with station maintenance, and monitor temperature in order to be used in a Mars colony simulation. Students will learn about using for loops, while loops, and nested loops as well as recursive functions. They'll also learn about three sensors – the Ultrasonic, the Time of Flight (TOF), and the environmental (T-H).

In Activity 1, students meet Kumar Patel, an automation engineer who works for Borealis Space Systems. He would like to use Seeker to perform tasks in the clean room because robots generate less dust than humans. Students will use Seeker's Ultrasonic Sensor to make precise deliveries around the clean room in rover mode. They will program Seeker using while and until loops in order to continually analyze the distance between Seeker and a wall. Using this method, Seeker will locate, pick up, and transport canisters around the clean room. The activity ends with students taking a video of Seeker completing the transport challenge to show Kumar.

In Activity 2, students meet Carlee Jackson, a college student who is spending a semester as an analog astronaut. Because Seeker is so adaptable, Kumar would also like Seeker to assist the analog astronauts around the colony station. Students are introduced to the environmental sensors to track Mars' temperature and humidity. They're also introduced to the Time of Flight (TOF) sensors that allow Seeker to be more precise with locating and placing objects. Students program Seeker using the TOF sensors to replace CO_2 scrubbers and oxygen canisters in the Mars simulation station. They also program Seeker to respond to hand gestures. Using all these pieces, the activity concludes with students completing a challenge to program Seeker to exit the facility and collect rock samples all while being aware of Mars' weather conditions.

Careers

- Automation Engineer
- Analog Astronaut

Context and Setting

A private aerospace company wants to use Seeker to perform tasks in a clean room as well as for station maintenance and sample collection in their Mars colony simulation.

Computer Science and Technology Concepts

- For Loops and While Loops
- Nested Loops
- Recursive Functions
- Ultrasonic Sensor
- Time of Flight Sensor
- Environmental Sensor

Unit Prep Instructions

- Students will be using Seeker in rover mode. They'll be instructed on how to reconfigure the robot in Activity 1 and will learn how to add the Time of Flight sensors in Activity 2.
- This Unit includes a lot of floor time for Seeker. In Activity 1, space will need to be created to be the clean room Seeker is working in. In Activity 2, a corridor will need to be created for Seeker using books, cardboard, or another material. For the challenge at the end of Activity 2, students will need a space designated to serve as an airlock and use of the surrounding area to represent Mars' surface.

- Consider giving student pairs different roles. For example, for Activity 1, have one student oversee hardware while the other student oversees software. Then, switch roles for Activity 2.

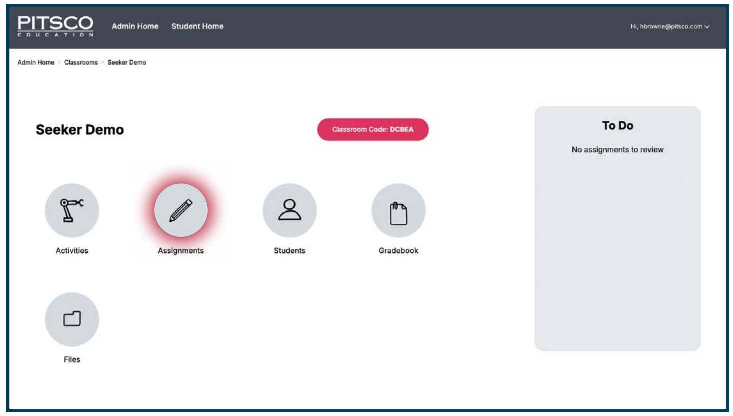
Unit Discussion Questions

- How can robots make space exploration safer?
- How can robots assist aerospace engineers?

Unit Assignments

The following assignments are available to be used with the Unit 4 content. When you set up your classroom, the points possible defaults to zero points. If you choose to use an assignment, you will need to set the number of points the assignment is worth. This can be done in the Assignment section when managing your specific classroom.

When a student submits an assignment, it should show up as a To Do item for you to review and assign points.



Activity 1

- Page 7 – Submit Blockly code for making precise distance measurements: U4A1-MeasureDistance.
- Page 7 – Submit Blockly code for driving toward a workstation: U4A1-FirstDelivery.
- Page 9 – Submit Blockly code for delivering canisters: U4A1-Delivery.
- Page 12 – Submit Blockly code for using the Ultrasonic Sensor to find canisters: U4A1-FindCanister.
- Page 13 – Submit Blockly code for adding to the previous program the ability to spin to the canister: U4A1-FindCanister.
- Page 14 – Submit Blockly code for adding to the previous program the ability to pick up the canister: U4A1-FindCanister.
- Page 15 – Submit Blockly code for transporting canisters: U4A1-TransportChallenge
- Page 15 – Submit a video of Seeker completing the transport challenge.

Activity 2

- Page 3 – Submit Blockly code for collecting weather data: U4A2-EnvironmentalMonitor.
- Page 6 – Submit Blockly code for testing the Time of Flight Sensor: U4A2-TOFSensorTest.
- Page 8 – Submit Blockly code for Seeker traveling down a corridor: U4A2-StationNavigation.
- Page 11 – Submit the updated Blockly code for replacing the CO_2 and oxygen canisters: U4A2-StationMaintenance.
- Page 11 – Submit Blockly code for the gesture sensing program: U4A2-Gesture.
- Page 15 – Submit Blockly code for completing the Mars Rover challenge.
- Page 15 – Submit a video of Seeker completing the Mars Rover challenge.

Global Activities

- Career Activity 1: Career Research – Research a career of their choice and submit information about the career in text field.
  - Consider having students research the careers mentioned in this Unit to make a stronger connection between real careers and how robotics is impacting those careers.
    - Automation Engineer
    - Analog Astronaut
    - Aerospace Engineer
- Career Activity 2: Career Priorities – In a text field, list priorities for choosing a career, three possible career choices, advantages and disadvantages of each career, and how the career priorities are met with each career.
- Career Activity 3: Interest Profile – Complete an interest profile and record results from the profile in a text field.

Guide to Activity 1: Satellite Manufacturing Facility

Page 1 – Borealis Space Systems

Students learn about Borealis Space Systems, a private aerospace company. Sophia and Lucas speculate how Seeker could be beneficial to an aerospace company and a Mars colony. They decide to reach out to Borealis Space Systems.

Tasks to Complete

- There are no official tasks to complete.

Teaching Tips

- Seeker robots will be used in rover mode in this Unit. Students will be instructed on how to reconfigure Seeker.
- This Unit includes a lot of floor time for Seeker. In Activity 1, space will need to be created to be the clean room Seeker is working in.
- Consider having students research how robots are being used in the aerospace industry.

Notes:

Icon Key

- Code Saved to Student Profile
- Code Testing Task
- *Innovation Notebook* Task
- Engineering Task
- Guided Programming Task
- Exploration Task
- Sequencing Task
- Debugging/ Troubleshooting Task
- Challenge Task

Page 2 – NDA with Borealis

Students meet Kumar Patil, an automation engineer at Borealis Space Systems. Kumar suggests an NDA to protect both Borealis Space Systems and Adaptibots. He would like to see how Seeker could help with work in the clean room and in transporting chemical canisters.

Tasks to Complete

- Students sign the NDA with Borealis Space Systems.

Teaching Tips

- Consider having students research the careers of an automation engineer. Career activities are provided as assignments that can be assigned through the teacher portal.

Notes:

Page 3 – Rover Mode

Students are led through resetting the servos to 90 degrees, modifying Seeker to be in rover mode, wiring the servos and Ultrasonic Sensors, and readjusting components as needed.

Tasks to Complete

- Modify Seeker from robot mode to rover mode.
- Wire and reset servos with the gripper forks attachments.
- Attach and wire the Ultrasonic Sensor.

Teaching Tips

- If Seeker is being used in more than one class period, the students who first use Seeker will complete the modifications Seeker needs for rover mode. Students in later classes can follow the Technical Guide to check the modifications were made correctly.

Notes:

Page 4 – The Ultrasonic Sensor

Lucas and Sophia explain how the Ultrasonic Sensor works. Students are guided through creating a new program that allows them to test the Ultrasonic Sensor.

Tasks to Complete

- Create a simple program to test the Ultrasonic Sensor.

Teaching Tips

- Students will learn about echolocation. Consider having students research the different animals that use echolocation.
- Students will be testing the Ultrasonic Sensor. They will need to have an object or person within 2 meters of Seeker in order to get a measurement, which will appear in the app's terminal.

Notes:

Page 5 – While Loops

Lucas explains how while loops work and how using them will benefit a distance measuring program. He leads students through making a simple distance measuring program using while loops and the Ultrasonic Sensor.

Tasks to Complete

- Create a distance measuring program using a forever loop.
- Improve the program using a while loop.
- Test the program on Seeker.

Teaching Tips

- This is the first time students are using a while loop. A while loop only repeats while a specific condition is true. A real-life example could be that, at a traffic light, cars continue to drive while their light is green.

Notes:

Page 6 – Until Loops

Lucas describes how students can improve the data Seeker prints by attaching labels using concatenation. Concatenation is when multiple pieces of data are combined into a single string. Lucas explains that until loops repeat until a condition becomes true. Students predict what will happen when the distance measuring program uses an until loop instead of a while loop in their *Innovation Notebook*.

Tasks to Complete

- Improve the distance measuring program using concatenation.
- Predict what will happen when the distance measuring program uses an until loop instead of a while loop.
- Test the program on Seeker.

Teaching Tips

- This is the first time students are using an until loop. An until loop is the opposite of a while loop and repeats until a condition becomes true. A real-life example of this could be how, during a meal, we eat until we are full. When it's true that we are full, we stop.
- See page 12 for the *Innovation Notebook* Answer Key for this task.

Notes:

Page 7 – Drive to Wall

Sophia explains how to switch between a while loop and an until loop. She then has students test and save the program. She's also started code to deliver supplies in the clean room. Sophia explains some of the code, but, ultimately, has students place the given blocks into the correct sequence.

Tasks to Complete

- Save the last program to their profile as U4A1-MeasureDistance and submit.
- Place the given blocks and algorithms in the correct sequence and locations in the program.
- Save the new program to their profile as U4A1-FirstDelivery and submit.
- Test the program on Seeker.

Teaching Tips

- When the Load Code button is clicked to bring up the blocks for this task, some of the blocks are already put together into the correct sequence or algorithm. To move a series of connected blocks, grab the top block and move it to where it needs to go. The other blocks should stay connected to it, allowing you to move the entire sequence.
- If students make mistakes in the provided sequences or algorithms, they can click **Load Code** again to start over.
- Students will know they have a correct solution when Seeker successfully drives toward the wall, slows down, and then stops within twelve centimeters.

Notes:

Page 8 – Satellite Deliveries

Sophia has started a program for Seeker to make deliveries around a Borealis clean room. If prop canisters have not yet been created, Sophia will guide students through making them using the red and green slips of paper found in Seeker's kit. Sophia then has students load her code as she talks through parts of it. There are three bugs that students will need to find and fix using the debugging process.

Tasks to Complete

- Create two cylinders to represent components of a Borealis satellite – propulsion and communications modules.
- Debug the Satellite Deliveries program using the debugging process.
- Document the bugs and fixes in the Satellite Deliveries section of their *Innovation Notebook*.
- Test the program on Seeker to see if the bugs have been fixed and Seeker delivers both cylinders.

Teaching Tips

- The Seeker kit has two red and two green slips of paper that can be used to create cylinders. If they haven't been created yet, students will be prompted to loop one edge of the paper around, so it overlaps the other edge by about two to two and a half inches. The diameter of the cylinder should be about two and a half inches.
    - Note that if the diameter of the cylinder is too big, the canister will drag on the ground. If it's too small, it will make it more difficult for Seeker to pick up.
- See page 14 for the *Innovation Notebook* Answer Key for this task.
- The debugging process can be referenced in the Appendix of the *Innovation Notebook*.
- The correct bug fixes include:

- In **redDelivery** function, Servo 4 position should remain at **armsUp** position.
- In **greenDelivery** function, the until loop should be a while loop.
- In **returnToStart** function, the Ultrasonic Sensor is the wrong port. It should be Port 1.

Notes:

Page 9 – Sensor Exploration

Students confirm they found the correct bugs and save the program to their profile. Sophia and Lucas are going to think about how they could use the Ultrasonic Sensor to locate an object in the clean room, go to pick it up, and retrieve it. In the meantime, students will explore other ways to use the Ultrasonic Sensor.

Tasks to Complete

- Save the program to their profile as U4A1-Delivery and submit.
- Explore ways to use the Ultrasonic Sensor.
- Test the exploration ideas on Seeker.

Teaching Tips

- Exploration ideas:
  - Program Seeker as a security robot. If the Ultrasonic Sensor detects a person within a certain range, it could then sound an alarm and flash its pixels.
  - Program Seeker as part of a manufacturing assembly line at Borealis and count parts that passed by on conveyor belt. When Seeker detects that a part goes by, it increases the count. When 10 parts go by, it can use its pixels and sound to indicate the group of 10. You could also display the count in the terminal.
  - Program Seeker to avoid other Seeker robots. Determine if the ultrasonic pulses from one sensor interfere with another robot's sensor, causing the distance measurements to be wrong. Investigate these possibilities with another group and their Seeker robot.
- Exploration ideas can also be found in the Sensor Exploration section of their *Innovation Notebook*.

Notes:

Page 10 – Canister Transport: Program Start

Sophia and Lucas introduce students to the canister transport program. They explain what will happen, but this is a program Sophia and Lucas will help students program from pseudocode. Students will establish the variables and complete the Program Start section of the pseudocode.

Tasks to Complete

- Create variables to set servo positions.
- Create variables to set other values that will be used in the program.
- Invert one of the arm servos.
- Set the servo speeds.

Teaching Tips

- Variable names and values can be found in the Canister Pickup: Program Start section of the *Innovation Notebook*.
- The pseudocode for the program can be found in the Appendix of the *Innovation Notebook*.
- Inverting one of the arm servos will allow one servo to rotate clockwise and the other to rotate counterclockwise to lift a canister using the same servo position.
- Students will want to save the program so they can continue working on it in the next section.

Notes:

Page 11 – Canister Transport: Find a Canister

Sophia guides students through building the portion of the program that finds the canister. They'll build the program alongside Sophia but can also reference the pseudocode in the *Innovation Notebook*. Students will use the pseudocode to plan how they would write the if and else-if statements Seeker will need.

Tasks to Complete

- Move sensor to straight forward and lift arms to lift positions.
- Use pixels to indicate the sensor is looking for a canister.
- Use a while loop to allow Seeker to keep scanning for the canister.
- Change the sensor direction by a certain increment to look in a new direction.
- Write the two conditions for the if and else-if statements in the Canister Transport: Find a Canister section of your *Innovation Notebook*.

Teaching Tips

  - The pseudocode for the program can be found in the Appendix of the *Innovation Notebook*.
  - Students will want to save the program so they can continue working on it in the next section.
  - See page 15 for the *Innovation Notebook* Answer Key for this task.

Notes:

Page 12 – Canister Transport: Scanning Conditions

Sophia continues to lead students through building the canister transport program. Students will work on programming Seeker to move the sensor in order to continue to scan for a canister.

Tasks to Complete

- Program the if and else-if statements to determine where Seeker needs to scan.
- Move the sensor on Seeker to look for the canister.
- Save the program to their profile as U4A1-FindCanister and submit.
- Test the program on Seeker. Seeker should scan the area and move forward if it doesn't find anything. When Seeker detects an objective, it will stop.

Teaching Tips

- The pseudocode for the program can be found in the Appendix of the *Innovation Notebook*.
- This submission will only have the code completed for the Program Start section and the **findCanister** procedure. Students will be building two more procedures as they move forward.

Notes:

Page 13 – Canister Transport: Spin to the Canister

Lucas guides students through creating the **spinToCanister** procedure, which runs after Seeker identifies a canister. Students will be programming Seeker to spin left or right, depending on where the canister is, and line up with the canister in order to later pick it up.

Tasks to Complete

- Use pixels to indicate a canister has been detected.
- Spin left or right the appropriate number of degrees.
- Open the lift arms, look straight ahead, and use sound and animations to indicate Seeker is lined up on the canister.
- Save the program to their profile as U4A1-FindCanister and submit.
- Test the program on Seeker.

Teaching Tips

- The pseudocode for the program can be found in the Appendix of the *Innovation Notebook*.
- This submission will have the Program Start and the **findCanister** procedure code from the last submission. The new code will be found in the **spinToCanister** procedure, which should be called at the end of the **findCanister** procedure. Students will be building one more procedure as they move forward to complete the program.
- For the chemical canister, students can use a plastic ball-pit ball, a pickle or Wiffle® ball, or a foam ball. Students can also make their own paper canister using the PDF template and following its cut-and-fold instructions.

Page 14 – Canister Transport: Pick Up Canister

Lucas guides students through creating the **pickUpCanister** procedure, which allows Seeker to pick up a canister and runs after Seeker has lined up with it. Students will be programming Seeker to drive forward a precise distance and move the lift arms to the lift position.

Tasks to Complete

- Use pixels to indicate whether Seeker is driving toward or Seeker has picked up or has lost the canister.
- Drive toward the canister and continually calculate how close Seeker is to the canister in order to ensure a precise pick up.
- Program the lift arms to move to the lift position to pick up the canister.
- Save the program to their profile as U4A1-FindCanister and submit.
- Test the program on Seeker.

Teaching Tips

- The pseudocode for the program can be found in the Appendix of the *Innovation Notebook*.
- The program they submit should include the past code from the Program Start section, the **findCanister** procedure, and the **spinToCanister** procedure. The new code will be found in the **pickUpCanister** procedure, which should be called at the end of the **spinToCanister** procedure.

Notes:

Page 15 – Transport Challenge

Sophia and Lucas video call Kumar to fill him in on what they've programmed Seeker to do. Kumar would like a demonstration of Seeker's capabilities, but, because he's traveling, students will need to create a video to send him. Sophia realizes that the Find Canister program locates and picks up canister, but it never takes them anywhere. Students will need to complete the Transport Challenge by programming Seeker to return to its starting position after picking up the canister. The video they create for Kumar needs to show the completed Transport Challenge.

Tasks to Complete

- Complete the Transport Challenge.
- Save the program to their profile as U4A1-TransportChallenge and submit.
- Submit a video of Seeker completing the Transport Challenge.

Teaching Tips

- The program students submit must include all the code from the Find Canister program.
- Students will need to take a video of Seeker completing the entire program, so each Seeker will need to have space to test the program and a canister to use. Canisters could be represented by a variety of objects including a plastic ball-pit ball, a pickle or Wiffle ball, a foam ball, or a paper canister using the template students were provided.

Notes:

Activity 1 *Innovation Notebook* Answer Key

Until Loops

Predict what will happen when the distance measuring program uses an until loop instead of a while loop.

Instead of taking a distance measurement when the Start button is pressed, the program will continuously take distance measurements until the Start button is pressed. Pressing the Start button will stop Seeker from measuring distance. This is the opposite behavior of when a while loop is used.

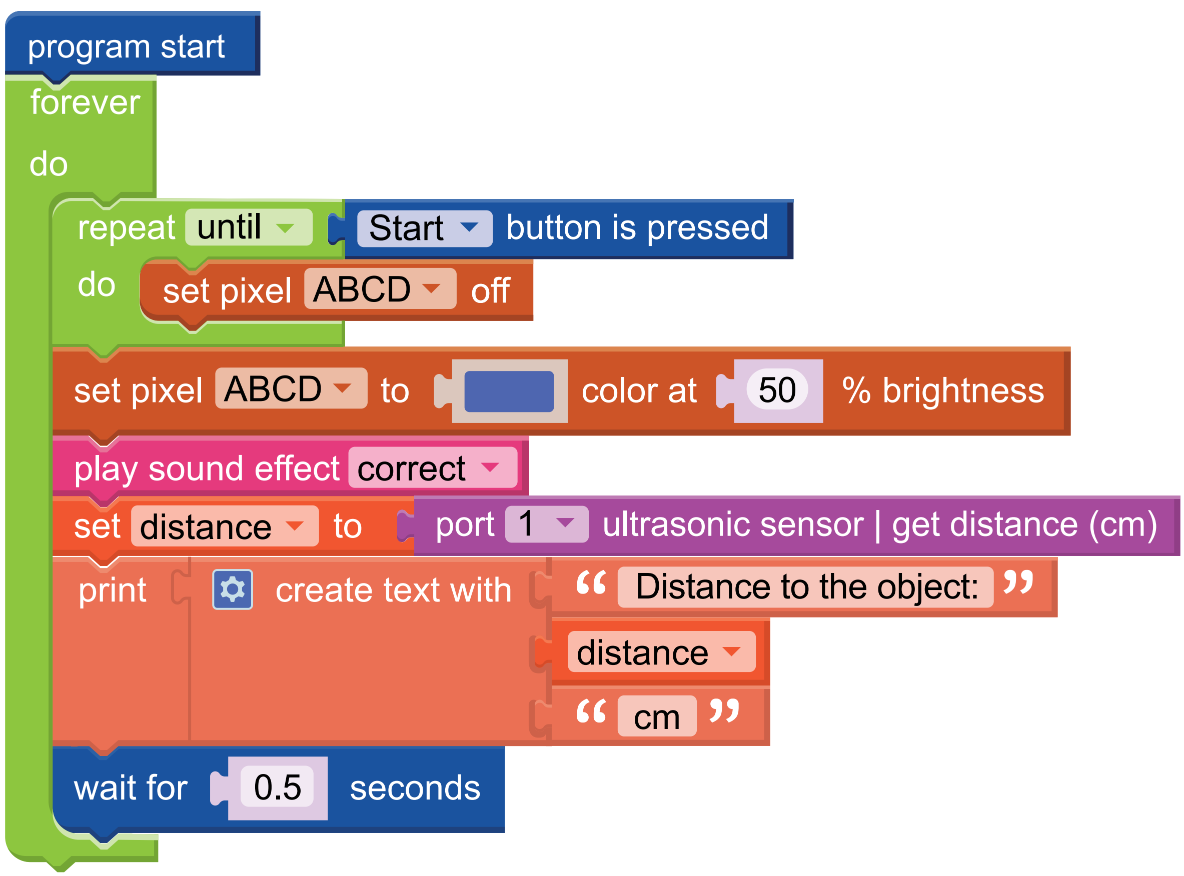
Run the program and test your prediction. Was your prediction correct? If not, explain what Seeker does when an until loop is used rather than a while loop.

Answers will vary depending on student predictions. If their prediction is wrong, students should explain that Seeker continuously takes distance measurements until the Start button is pressed and that pressing the Start button stops Seeker from measuring distance.

Record how you fixed the program so Seeker only takes distance measurements when the Start button is pressed using an until loop. See the code at the bottom of the page.

To fix the program, students simply need to swap what is in the loop with what is outside the loop. In other words, the **set pixel off block** goes inside the loop and everything else goes outside the until loop. When this is done, the pixels are turned off until the Start button is pressed. Then, the until loop stops running and the distance measurement is taken and displayed.

A second solution would be to include a **not block** before the **Start button is pressed block**. This inverts the logical value of the loop condition, making it the same as a while loop (loop until not the Start button is pressed). This solution is more complicated logically. The **not block** will be discussed in another Unit.



Satellite Deliveries

Bug Description	Bug Location	Bug Fix
Both servos are moved to the **armsDown** position, which drops off both the red and green cylinders. Only the red cylinder should be delivered at the first location.	**redDelivery** Function	Change the servo position for servo 4 to the **armsUp** position.
Seeker does not drive to within 20 cm of the wall to make the red cylinder delivery.	**greenDelivery** Function	Change the until loop to a while loop so the loop repeats while Seeker is more than 20 cm away.
The program stops and the terminal reports an error when Seeker tries to return to its start position.	**returnToStart** Function	In the while loop, the wrong port is used for the Ultrasonic Sensor. Change the port to Port 1.

Sensor Exploration

- Program Seeker as a security robot. If the Ultrasonic Sensor detects a person within a certain range, it could then sound an alarm and flash it pixels.
- Program Seeker as part of a manufacturing assembly line at Borealis and count parts that pass by on conveyor belt. When seeker detects that a part goes by, it increases the count. When 10 parts go by, it could use its pixels and sound to indicate the group of 10. You could also display the count in the terminal.
- Program Seeker to avoid other Seeker robots. Determine if the ultrasonic pulses from one sensor interfere with another robot's sensor causing the distance measurements to be wrong. Investigate these possibilities with another group and their Seeker robot.

Canister Transport: Program Start

Variable Name	Variable Value	Variable Purpose
limitRight	40	Limits the distance that the Ultrasonic Sensor can rotate to the right to 40 degrees.
limitLeft	140	Limits the distance that the Ultrasonic Sensor can rotate to the left to 140 degrees.
posLift	180	Sets the position of the lift arms for lifting a canister off the ground.
posOpen	90	Sets the open position of the lift arms for capturing a canister.
lookRange	50	Seeker will look for canister that is within 50 cm of its sensor.
Increment	5	Sets the number of degrees the sensor will rotate to look in a new direction.
lookDirection	90	Sets the initial direction the sensor will look to 90 degrees, which is straight forward.

Canister Transport: Find a Canister

Use the pseudocode for the Canister Transport program to determine how to write the conditions for the if and else-if statements for Steps 2.B.c and 2.B.d. The entire pseudocode can be found in the Appendix or you can use the partial pseudocode shown here. Use the variable names you created for the program when writing your conditions.

if: **lookDirection** = **limitLeft**

else if: **lookDirection** = 90

Partial Pseudocode for the Canister Transport Program

2. Find a canister
   A. Move sensor to straight forward and lift arms to lift positions.
   B. While no canister is detected within the looking range of the sensor . . .
      a. Use pixels to indicate the sensor is looking for a canister.
      b. Change the sensor direction by a certain increment to look in a new direction.
      c. If the look direction is greater than the left range of the sensor servo, move the sensor to the right side.
      d. If sensor didn't see a canister as it rotated through its different position and the look direction is back to forward, drive forward a short distance so Seeker can look again.
      e. Move the sensor to the new looking direction.
      f. Repeat these steps while no canister is detected.
   C. Store the distance to the canister.

Guide to Activity 2: Mars Simulation Colony

Page 1 – Simulated Mars Colony

Students begin this activity by meeting Carlee, a college student who is spending a semester as an analog astronaut. Kumar and Carlee would like to see how Seeker could be of use in the simulated Mars colony with station maintenance and collecting rock samples.

Tasks to Complete

- There are no official tasks to complete.

Teaching Tips

- During this Activity, students will be testing programs that require them to have a corridor set up. Materials such as boxes, books, or an existing wall can be used to help create one. Students won't be asked to set up the corridor until Video 8, but, as the teacher, begin planning and gathering materials as needed.
- Simulating a Mars colony is not a new concept, consider having students research what aerospace companies have already done.

Notes:

Icon Key

- Code Saved to Student Profile
- Code Testing Task
- *Innovation Notebook* Task
- Engineering Task
- Guided Programming Task
- Exploration Task
- Sequencing Task
- Debugging/ Troubleshooting Task
- Challenge Task

Page 2 – Environmental Sensor

Carlee shows the Adaptibots team the simulated Mars colony crater. They discuss Mars' atmosphere and the possibility of using Seeker to monitor the weather. Back at Adaptibots, Sophia and Lucas guide students through attaching an environmental sensor – or T-H sensor – to Seeker.

Tasks to Complete

Mount and attach the T-H sensor to Seeker.

Teaching Tips

- Students need to make sure the sensor is mounted using a sensor mount. They'll need to use a hexlock fastener and thumbscrew to attach the sensor to the back of Seeker's chassis and tighten the thumbscrew to hold it in place. The T-H sensor needs to have the blue plastic part of the sensor centered directly behind the controller. Students should use a short sensor cable to connect the sensor to sensor Port 3 on the controller and tuck any extra sensor cable under Seeker's chassis.
- Consider having students research the weather in your area including the temperature and humidity.

Notes:

Page 3 – Weather Reporting

Sophia guides students through creating a new program to collect weather data. Students will save and test the program.

Tasks to Complete

- Create a program to collect weather data.
- Save the program to their profile as U4A2-EnvironmentalMonitor and submit.
- Test the program on Seeker.

Teaching Tips

- Consider having special stations set up for students to test their program. One station could have a hair dryer to make the "weather" conditions warm, while another has a fan to make the conditions cooler. Changes in temperature may be subtle.
  - **Warnings:**
    - Hair dryers should not be placed directly on to the sensor but held 6-12 inches away.
    - If you are using a hair dryer or other tool to manipulate the temperature, do not allow the temperature to climb above 130°F. Going higher than this could potentially damage the robot or the sensor.
    - Do not use ice to try to bring Seeker's temperature down. Electronics and water are usually not a good mix.

Notes:

Page 4 – Environmental Exploration

Lucas introduces students to the program he created to graph the weather data. He explains why graphs and charts are useful.

Tasks to Complete

- Test the program to graph temperature.
- Complete the Environmental Exploration section of their *Innovation Notebook*.

Teaching Tips

- Exploration Ideas:
  - Seeker sounds an alarm and flashes its pixels when the temperature gets higher than 80 degrees.
  - Change Seeker's pixels based on the temperature or humidity. Use blue colors to indicate cooler temperatures or high moisture levels. Use red colors to indicate warm temperatures or dry humidity levels.
  - Allow the user to switch how the temperature is displayed between Fahrenheit and Celsius.
  - Seeker only takes a temperature reading with a certain event occurs, such as a button being pressed.

Notes:

Page 5 – Time of Flight Sensors

Sophia and Lucas meet Carlee back at the crater and fill her in on what they've done with Seeker's temperature reporting. Carlee then gives them the next task of Seeker working as a maintenance robot – specifically replacing oxygen canisters and CO_2 scrubbers in the station corridor. Back at Adaptibots, Lucas and Sophia decide to use two Time of Flight (TOF) sensors, which use an infrared laser to measure distance and are more precise than an Ultrasonic Sensor.

Tasks to Complete

- Attach the Time of Flight sensors to either side of Seeker.

Teaching Tips

- Students will need to locate the two TOF sensors from their kits. They will use hexlock fasteners to attach them to the 90° angle brackets at the front of the robot's chassis and vertically to the angle brackets. Students will use short sensor cables to connect the sensor on Seeker's right side to sensor Port 2 and the sensor on the left side to sensor Port 4. Students should tuck the excess cable out of the way.

Notes:

Page 6 – TOF Sensor Test

Sophia guides students through creating a new program to test the TOF sensors. Students will test the program on Seeker.

Tasks to Complete

- Create a program to test the TOF sensors.
- Save the program to their profile as U4A2-TOFSensorTest and submit.
- Test the program on Seeker.

Teaching Tips

- The program students create will display the distance in millimeters (mm) in the terminal. It will also play a sound – the farther away an object is, the higher the frequency of the sound Seeker plays.
- Students can use any object that's around, including themselves, to test the program.

Notes:

Page 7 – Station Navigation Pseudocode

In order to replace the oxygen canisters and CO_2 scrubbers in the station corridor, Sophia decomposes the problem into two parts. Students will first focus on navigating the corridor before they worry about the canisters and scrubbers. They will need to start by writing pseudocode in their *Innovation Notebooks*.

Tasks to Complete

- Complete pseudocode for Seeker to navigate the central corridor using TOF sensors in the Station Navigation Pseudocode section of their *Innovation Notebook*.

Teaching Tips

- Students should use the TOF sensors to compare the distance on the left and the right to help Seeker stay in the middle of the corridor.

Notes:

Page 8 – Station Navigation Puzzle

Sophia shows students her pseudocode and walks them through the program she's started. There are blocks not connected to the code. Students will use the pseudocode to complete the program.

Tasks to Complete

- Review Sophia's pseudocode and the started program.
- Complete the program by sequencing the blocks in the correct order based on Sophia's pseudocode.
- Save the program to their profile as U4A2-StationNavigation and submit.
- Create a corridor to test Seeker.
- Test the program on Seeker.

Teaching Tips

- Students will need to set up a corridor. Materials such as boxes, books, or an existing wall can be used to help create one. The corridor should be between 1/2 to 1 meter wide and at least 1-1/2 meter long, but no longer than 3 meters. Make sure that whatever you use as walls sticks up higher than the TOF sensors on the side of Seeker.

Notes:

Page 9 – CO_2 Scrubbers and Oxygen Canisters

Lucas explains Sophia's completed could to students. He asks them to brainstorm ways to modify the program so Seeker can locate the oxygen canisters and CO_2 scrubbers.

Tasks to Complete

- Brainstorm program modifications in the CO_2 Scrubbers and Oxygen Canisters section of their *Innovation Notebook*.

Teaching Tips

- It may be helpful for students to place their Seeker in the created corridor to think through possibilities.
- See page 26 for the *Innovation Notebook* Answer Key for this task.

Notes:

Page 10 – Maintenance Programming

Lucas guides students through adding to the program to allow Seeker to find the oxygen canisters and CO_2 scrubbers. Students will need to finish the program by adding in the code for the **replaceItem** procedure. They can use the Maintenance Programming section of their *Innovation Notebook*.

Tasks to Complete

- Program Seeker to locate the oxygen canisters and CO_2 scrubbers.
- Program Seeker to replace the oxygen canisters and CO_2 scrubbers using the pseudocode in the Maintenance Programming section of their *Innovation Notebook*.
- Test the program on Seeker using their corridor.

Teaching Tips

- Students are prompted to save this program in the next video. If students are ending for the day, prompt them to save it before leaving the classroom. They can save the program to their profile as U4A2-StationMaintenance but will submit it during Video 11.
- Students will be testing their program to replace both the CO_2 filter and oxygen canister. For now, replacing a filter or canister is done by touching the object with at least one of Seeker's arms.
- Students may need to tweak the parameters until Seeker is successful.
- When designing the corridor:
  - The gap for a missing canister should be about a 30 to 40 centimeters.
  - Gaps should be staggered and not directly across from each other.
  - The arrangement of the corridor may need to be adjusted as students test based on the results.
- Students may need to experiment with placing the canister in order for Seeker to sense both the gap and then the canister.

Notes:

Page 11 – Gesture Sensing

Lucas has students save the station maintenance program. He then gets a video call from Sophia and Carlee in the crater. They developed a program to allow Seeker to respond to hand gestures. This is to allow Seeker to assist astronauts outside of the station. Hands gestures are used instead of voice commands because the air on Mars is thinner and sound travels differently than it does here on Earth. Sophia walks Lucas and students through the program. Students will need to save the program and reposition the TOF sensors.

Tasks to Complete

- Review Sophia's Gesture Sensing program.
- Save the program to their profile as U4A2-StationMaintenance and submit.
- Reposition the TOF sensors.
- Save the program to their profile as U4A2-Gesture and submit.

Teaching Tips

- Students will need to reposition the TOF sensors.
  - The sensor connected to Port 2 is moved so it faces up and is just to the right of the controller. The sensor should hang off the edge of the chassis so the cable can pass under the chassis and then up through one of the larger holes to Port 2. Students will use a hexlock fastener and thumbscrew to hold the sensor in place.
  - The sensor connected to Port 4 is moved to the other side facing up. It should be positioned as far back as it can go on the chassis, hanging over the edge like the first sensor. Students will use a hexlock fastener and thumbscrew to hold it to the chassis.
  - There needs to be enough space between the two sensors that they trigger separately.
- When testing, if students wave their hands forward – the back sensor triggers first and then the front – Seeker will drive forward. If students wave their hands backward – the front sensor triggers first and then the back – Seeker will move in reverse.
  - It may be necessary to briefly pause their hand on the first sensor and then wave slowly toward the second sensor. This will ensure the sensors are triggered in the desired order.
- Consider having students research how sound travels differently on Mars than on Earth.

Notes:

Page 12 – Gesture Improvements

Lucas checks back in with Sophia and Carlee after testing out the gesture sensing program. Sophia and Carlee ask students to brainstorm ways to modify Seeker's hardware and software to allow for other gestures that Seeker could recognize, allowing Seeker to make turns, start and stop, move servos, or perform other tasks.

Tasks to Complete

- Complete the Gesture Improvements section of their *Innovation Notebook*.

Teaching Tips

- See page 27 for the *Innovation Notebook* Answer Key for this task.

Notes:

Page 13 – Mars Rover Challenge

Sophia introduces students to the Mars Rover Challenge. In this challenge, students will create a rover that can go out into harsh Mars environment to search for rocks and then bring them back to the station for analysis. Seeker will start its exploration from the airlock. The rover will need to monitor Mars' temperature and humidity and report these values back to the astronauts through the terminal. Seeker needs to automatically return to the airlock if the temperature is below 65°F or above 80°F. Seeker will need to collect two rock samples using its Ultrasonic Sensor and Time of Flight sensors to navigate, detect rock samples, and collect them for analysis. Students will need to build some kind of collection mechanism for Seeker.

Tasks to Complete

- Complete the Mars Rover Challenge.
- Submit a video of Seeker completing the Mars Rover Challenge.

Teaching Tips

- See pages 29-31 for the *Innovation Notebook* Answer Key for this task.
- Students will need to establish an area that is the airlock and scatter "rocks" around the area that can be collected.

Notes:

Evaluation Rubric

This rubric shows how you will be graded at completing the challenge according to the criteria and constraints.

Category	0 Points	1 Point	2 Points	3 Points	4 Points	Points Earned
Seeker's Features	Seeker's structure is missing all but one of the following: the Ultrasonic Sensor, the TH sensor, the TOF sensor, servo arms, and a collection device. Not all parts on Seeker are used in the code.	Seeker's structure includes only includes two of the following: the Ultrasonic Sensor, the TH sensor, the TOF sensor, servo arms, and a collection device. Not all parts on Seeker are used in the code.	Seeker's structure includes all but two of the following: the Ultrasonic Sensor, the TH sensor, the TOF sensor, servo arms, and a collection device. Not all parts on Seeker are used in the code.	Seeker's structure includes all but one of the following: the Ultrasonic Sensor, the TH sensor, the TOF sensor, servo arms, and a collection device. All parts are included, but not all parts are used in the code.	Seeker's structure includes all of the following: the Ultrasonic Sensor, the TH sensor, the TOF sensor, servo arms, and a collection device. All parts are used in the code.	
Rock Sample Collection	Seeker does not successfully locate a rock sample to collect.	Seeker attempts to collect one rock sample. The collection mechanism cannot secure the rock sample to take it back to the airlock.	Seeker collects one rock sample but doesn't attempt a second. The collection mechanism has trouble securing the rock sample until back in the airlock.	Seeker tracks the temperature outside the airlock and prints the data with labels to the terminal, but then does not automatically return to the airlock if the temperature gets outside the 65-80°F range.	Seeker tracks the temperature outside the airlock, prints the data with labels to the terminal, and automatically returns to the airlock if the temperature gets outside the 65-80°F range.	
Temperature Tracking	Seeker doesn't track the temperature.	Seeker tracks the temperature outside of the airlock but doesn't print the data.	Seeker tracks the temperature outside the airlock but prints data to the terminal without labels and does nothing with the data.	Seeker tracks the temperature outside the airlock and prints the data with labels to the terminal, but then does not automatically return to the airlock if the temperature gets outside the 65-80°F range.	Servos were always positioned correctly to pick up and deliver bags of groceries and to turn robot's head to indicate turning direction.	
Procedures	The program incorporates zero procedures to break the program into sections.	The program incorporates one or more procedures that are incorrectly used to break the program into sections.	The program incorporates one correctly used procedure to break the program into sections.	The program incorporates two correctly used procedures to break the program into sections.	The program incorporates three or more correctly used procedures to break the program into sections.	
Conditions	The program does not include conditions.	The program includes conditions, but the logic of how the conditions are used does not make sense.	The program includes one condition, and the logic of how the condition is used makes sense.	The program includes two conditions, and the logic of how the conditions are used makes sense.	The program includes at least three conditions, and the logic of how the conditions are used makes sense.	
Video	Video meets none of the aspects of a quality explanatory video: • The video clearly shows Seeker completing the challenge. • The explanation is clearly spoken. • The explanation fully explains how the rover works. • The video is 4-8 minutes long.	Video meets one aspect of a quality explanatory video: • The video clearly shows Seeker completing the challenge. • The explanation is clearly spoken. • The explanation fully explains how the rover works. • The video is 4-8 minutes long.	Video meets two aspects of a quality explanatory video: • The video clearly shows Seeker completing the challenge. • The explanation is clearly spoken. • The explanation fully explains how the rover works. • The video is 4-8 minutes long.	Video meets three aspects of a quality explanatory video: • The video clearly shows Seeker completing the challenge. • The explanation is clearly spoken. • The explanation fully explains how the rover works. • The video is 4-8 minutes long.	Video meets all four aspects of a quality explanatory video: • The video clearly shows Seeker completing the challenge. • The explanation is clearly spoken. • The explanation fully explains how the rover works. • The video is 4-8 minutes long.	
					Total Score:	

Bonus Points

1. Seeker keeps count of how many rock samples have been collected and displays the information in the terminal.
2. If possible, work with another group and Seeker robot to create a second program for the second Seeker to retrieve the rock samples from the airlock as they're delivered and organize them inside the station. Remember, though, only one airlock door can be open at a time, meaning only one Seeker should be in the airlock at a time.

Page 14 – Borealis Contract

Sophia and Lucas video call with Kumar and Carlee after sending them the video of Seeker completing the Mars Rover Challenge. Kumar shares that they want to purchase a number of Seekers and plans to send them the purchase order by the end of the day.

Tasks to Complete

- There are no official tasks to complete.

Teaching Tips

- Sophia mentioned earlier that they will use the profits from this contract to expand the company. Consider a discussion with students about what expansion possibilities would be most beneficial to Adaptibots.

Notes:

Activity 2

Environmental Exploration

Exploration Ideas:

- *Seeker sounds an alarm and flashes its pixels when the temperature gets higher than 80 degrees.*
- *Change Seeker's pixels based on the temperature or humidity. Use blue colors to indicate cooler temperatures or high moisture levels. Use red colors to indicate warm temperatures or dry humidity levels.*
- *Allow the user to switch how the temperature is displayed between Fahrenheit and Celsius.*
- *Seeker only takes a temperature reading when a certain event occurs, such as a button being pressed.*

Students should record any other ideas they try.

Station Navigation Pseudocode

Develop pseudocode for Seeker to navigate the central corridor using the TOF sensors.

Pseudocode will vary depending on how they go about the program but should use the TOF sensors to compare the distance on the left and the right to help Seeker stay in the middle of the corridor.

CO_2 Scrubbers and Oxygen Canisters

Brainstorm ideas for how the Station Navigation program will need to be modified so Seeker can locate the CO_2 scrubber and oxygen canister in your station's central corridor.

Answers will vary based on student ideas and their understanding of the Station Navigation program. Following are some examples ideas that students could list.

- When a TOF sensor detects a large distance that would represent the gap in the wall, pivot that direction, either left or right, to perform maintenance on the object located there.
- Use the Ultrasonic Sensor to scan back and forth like it did in the first Activity looking for the CO_2 scrubber and the oxygen canister. When one is detected, turn that direction to perform maintenance on it.
- Use the TOF sensor on the right to look for the CO_2 scrubber on the right. Use the TOF sensor on the left to maintain an equal distance from the left wall. When it detects the CO_2 scrubber, turn and perform maintenance on it. After the robot turns around, repeat this same process to locate the oxygen canister, which is now on Seeker's right side.

Maintenance Programming

*Use the following steps to complete the **replaceItems** procedure for the maintenance robot program. Follow the instructions carefully. Parameters that are listed in red might need to be adjusted to work for your specific course. Inside the **replaceItem** procedure:*

- *Stop Seeker from driving.*
- *Play a sound effect to indicate it has detected an item to replace.*
- *Pivot right for **90 degrees** at the **speed stored in the speed variable** so Seeker faces the item to replace.*
- *Drive forward at the **speed variable** for **10 centimeters** to the item.*
- *Set the servo positions for the arms to **180 degrees** to cause the arms to rotate inward. This represents replacing the CO_2 filter or oxygen canister. The arm servos are connected to servo Ports 2 and 4.*
- *Wait for **1 second** to give the servos time to move.*
- *Move all the servos back to their **90-degree positions** to move the arms back out.*
- *Drive in reverse at the **speed variable** for a distance of **20 centimeters** to return to the center of the corridor.*
- *Spin **left** at the speed variable for 90 degrees so Seeker faces the end of the corridor.*
- *Drive forward at the **speed variable** for 10 centimeters to drive past the item that was just replaced.*
- *While the TOF sensor on the right (Port 2) is greater than the **range variable**, start driving forward at the **speed variable speed**. This will cause Seeker to drive forward until it no longer detects the gap in the wall on its right side. It can now resume navigating the corridor in the main part of the program.*

At the beginning of the main part of the program, after declaring the variables:

- *Invert the direction of Servo 4 so the arms work together.*
- *Set the speeds of all the servos to **100%**.*

Gesture Improvements

*Brainstorm ideas for improving Seeker's ability to recognize gestures using **only the hardware provided in the kit**. As you brainstorm, make sure to record what hardware is needed, how it will recognize the gesture, and what task Seeker can do based on the gesture. Try to come up with at least three ideas.*

Answers will vary but some examples are shown here.

What hardware is required?	How will the hardware recognize the gesture?	What task will be completed?
Ultrasonic Sensor and at least one TOF sensor.	Place the Ultrasonic Sensor on the opposite side of Seeker's chassis from the TOF sensors facing up to detect left- to right-hand movements.	Seeker can spin left or right depending on if the Ultrasonic Sensor is triggered or a TOF sensor is triggered first.
TOF sensor	Use the sensor to detect if the hand is moving closer or farther away from the robot up and down.	Moving the hand closer could stop the robot. Moving the hand farther away could start the robot moving forward. You could even set the speed based on how close the hand is to the sensor.
TOF sensor or Ultrasonic Sensor	Place one of the sensors facing forward to detect the distance to the hand.	Program Seeker to keep a certain distance from the hand in front of it. If the hand moves farther away, Seeker speeds up, and, if the hand moves closer, it slows down.

*Brainstorm ideas for improving Seeker's ability to recognize gestures using **any hardware**. As you brainstorm, make sure to record what hardware is needed, how it will recognize the gesture, and what task Seeker can do based on the gesture. Try to come up with at least three ideas.*

Answers will vary but some examples are shown here.

What hardware is required?	How will the hardware recognize the gesture?	What task will be completed?
Two more TOF sensors	Using four TOF sensors placed on each corner of Seeker's chassis, Seeker could recognize multiple directions included left-right, front-back, diagonal, or a z-type pattern.	Seeker could move in any direction based on which gesture was given.
Two more TOF sensors	Four TOF sensors placed in a diamond pattern on Seeker's chassis to detect a hand within a certain range.	Each sensor could represent a button. This could allow you to "press a button" by placing your hand placing your hand over a particular sensor without actually having to bend over and press the button.
AI Camera	Use the camera to track hand motion.	Seeker could open or close its arms based on the hand signal.

Challenge: Mars Rover

Plan

Brainstorm names for your Mars rover.

Names should be school appropriate but otherwise have no guidelines unless established by the teacher. NASA doesn't have strict guidelines for naming rovers and will often use public contests and themes.

Use the following space to create a sketch of your Mars rover including the mechanism you'll use for collecting rock samples.

Sketches will vary depending on what kind of mechanisms students think they can build.

Develop pseudocode for Seeker to leave the airlock, detect a rock sample, navigate to the sample, collect it, and return to the airlock.

Answers will vary depending on how students approach solving the challenge.

Put to the Test

Program Observation Notes	Program Modification Notes
Observations will vary.	Modifications will vary.

Present

Use the following space to plan your video presentation of your Mars rover.

Video Presentation Plans:

Students should include a full script of their narration.

Find more cool stuff at

PITSCO.COM

PITSCO EDUCATION

Pitsco Education • P.O. Box 1708, Pittsburg, KS 66762 • 800-835-0686 • Pitsco.com

PE•0425•1125•01 79429