



Vision Sensor



Use the Vision Sensor to detect objects!



Discover new hands-on builds and
programming opportunities to further
your understanding of a subject matter.

The Completed Look of the Build



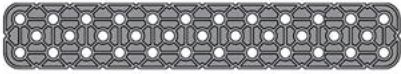
The completed Autopilot robot build.


This robot is designed so that it can be built quickly and drive around either autonomously or with the Controller in a short amount of time.


Build Instructions

1

The section at the top of the step shows important information for the build.


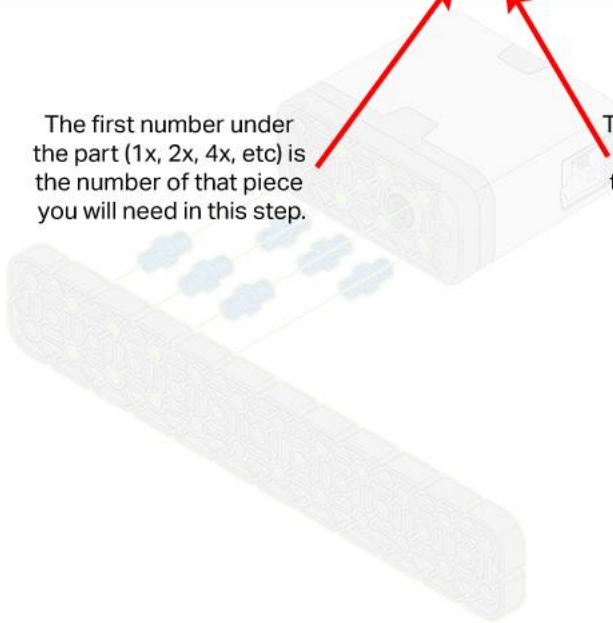
 **1x** - 2x12 Beam

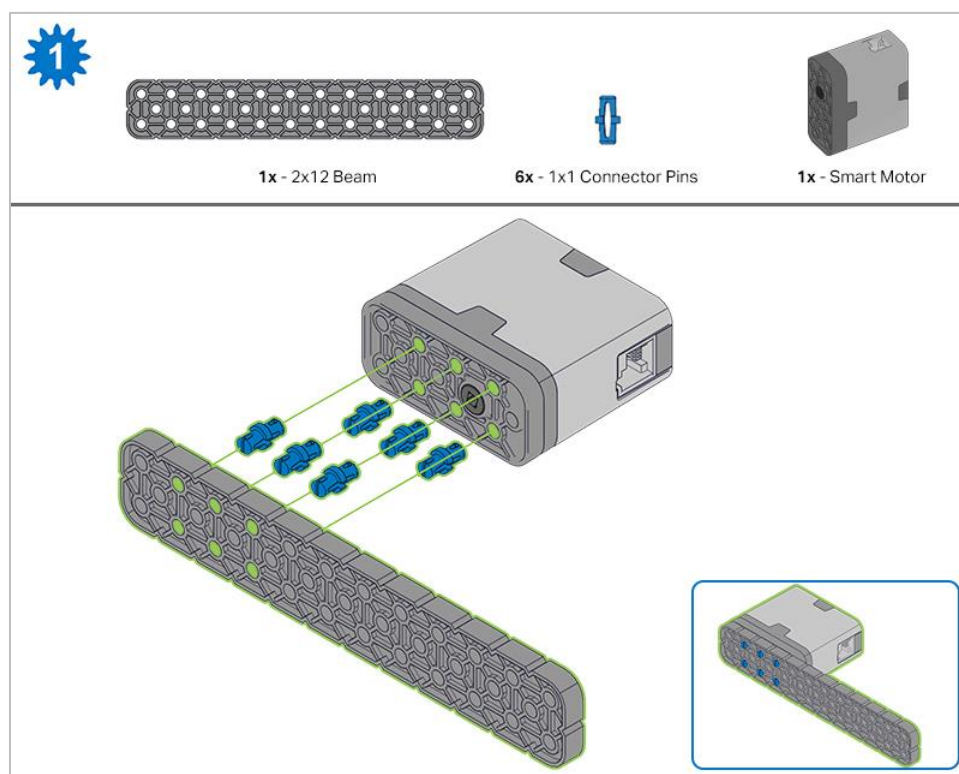
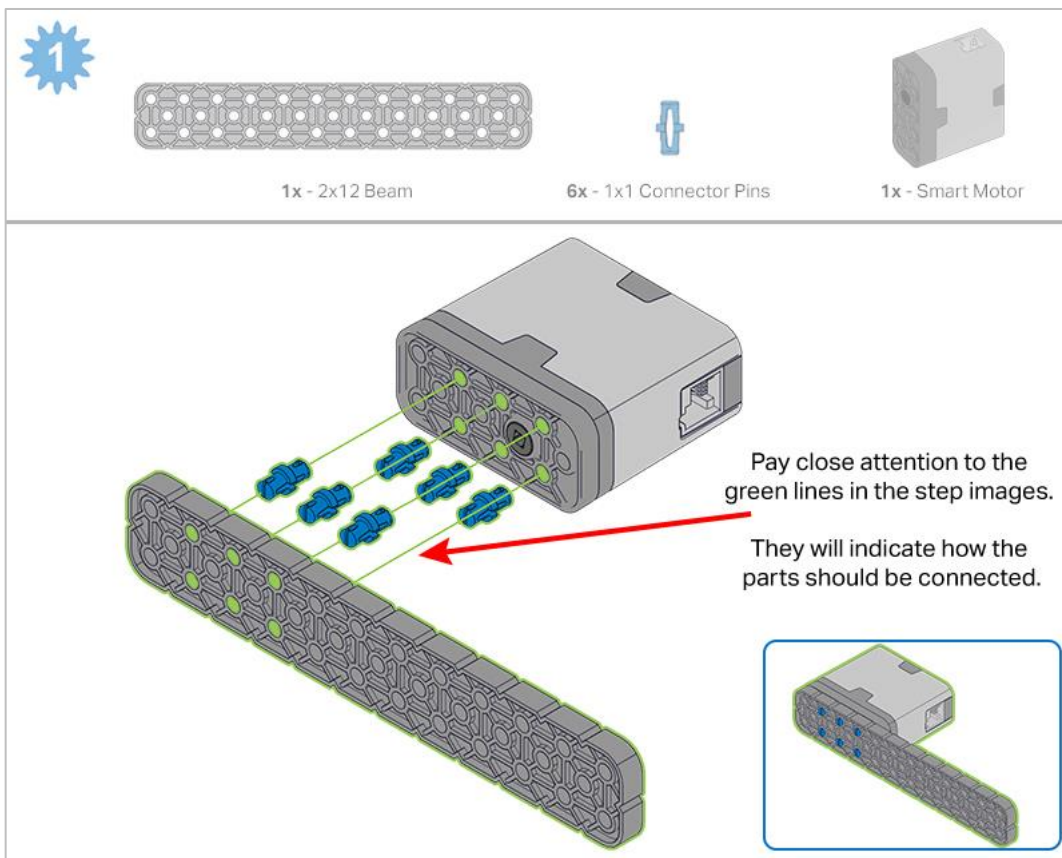
 **6x** - **1x1** Connector Pins

 **1x** - Smart Motor

The first number under the part (1x, 2x, 4x, etc) is the number of that piece you will need in this step.

The next information under the part image is the size (if necessary) and name of the part needed.





*Steps 1-6 will be repeated for steps 7-12, so it's possible to make both at once.
Count all pieces before starting your build and have them readily available.*

2



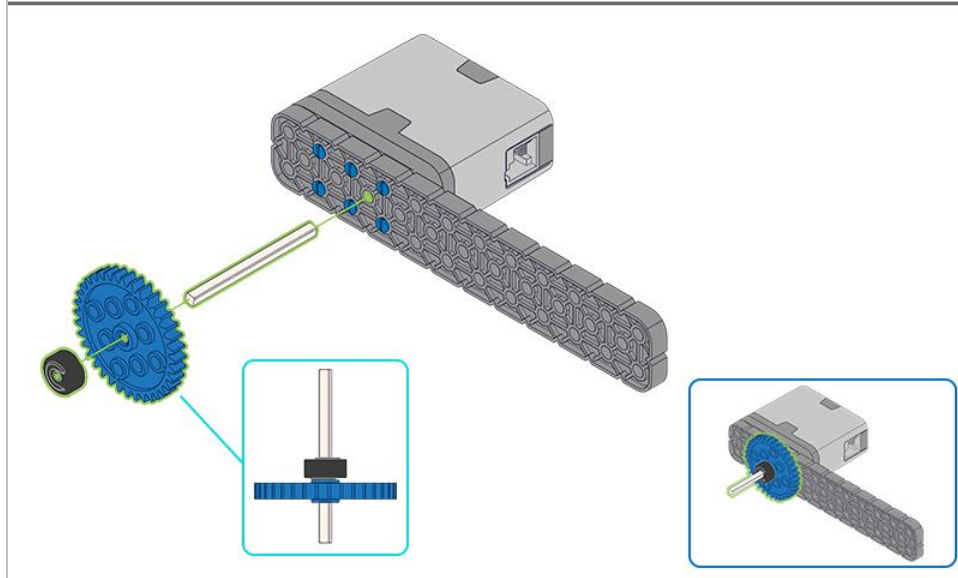
1x - 36 Tooth Gear



1x - Rubber Shaft Collar



1x - 4x Pitch Shaft

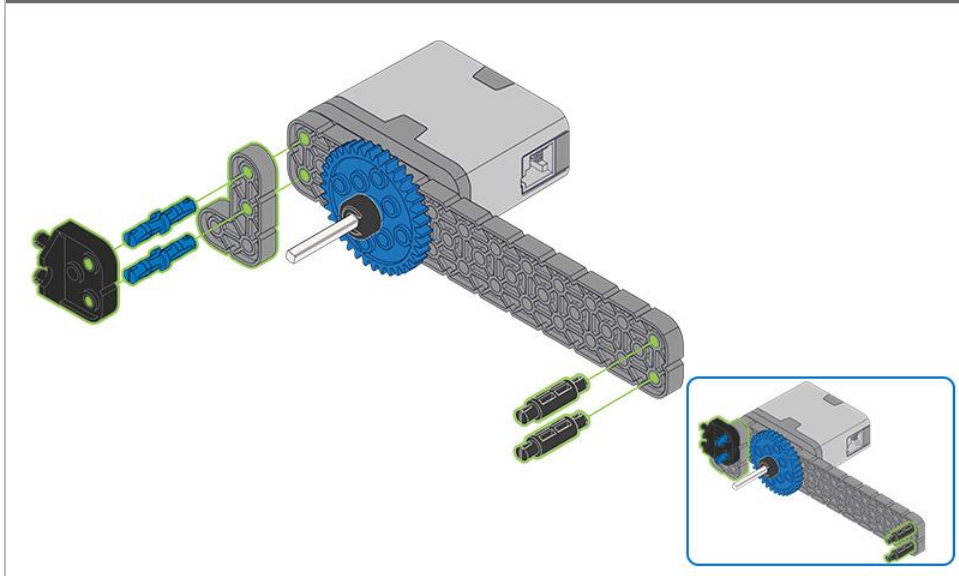


When adding the 4x Pitch Shaft, twist the pitch shaft to check for tension while turning. If it spins freely, it is not properly inserted into the motor.

3



1x - 2x Wide, 1x2 Offset Corner Connector 2x - 2x2 Connector Pins 1x - 2x3 Right Angle Beam 2x - 1x Pitch Standoffs



4



2x - 36 Tooth Gears

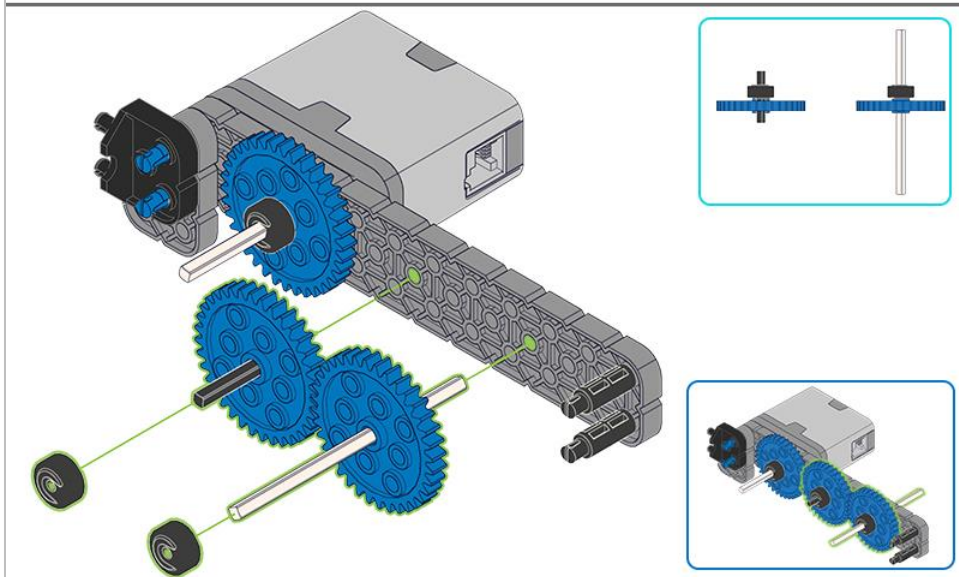


2x - Rubber Shaft Collars

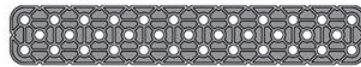
1x - 6x Pitch Shaft



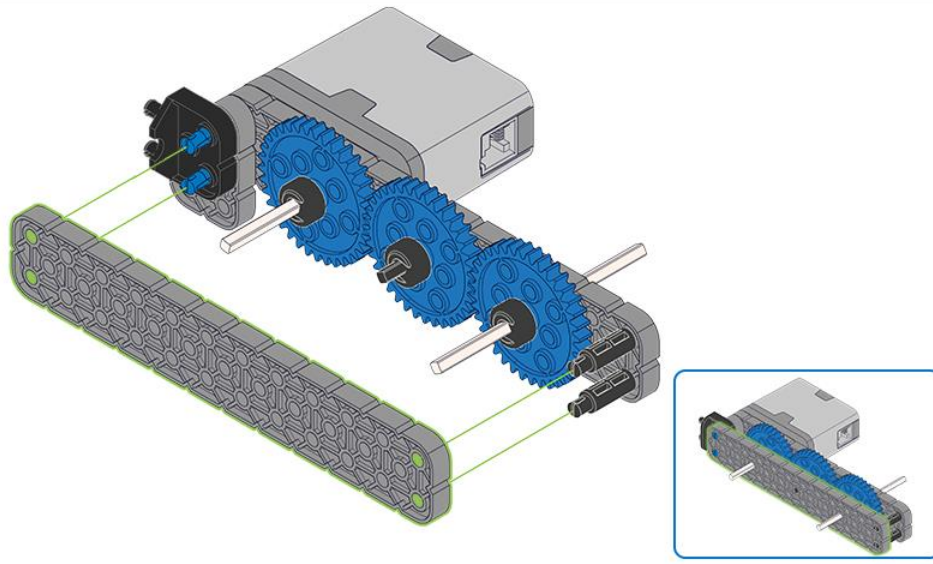
1x - 2x Plastic Pitch Shaft



5



1x - 2x12 Beam



Make sure the gears fit together properly before locking the Beam in place.

6



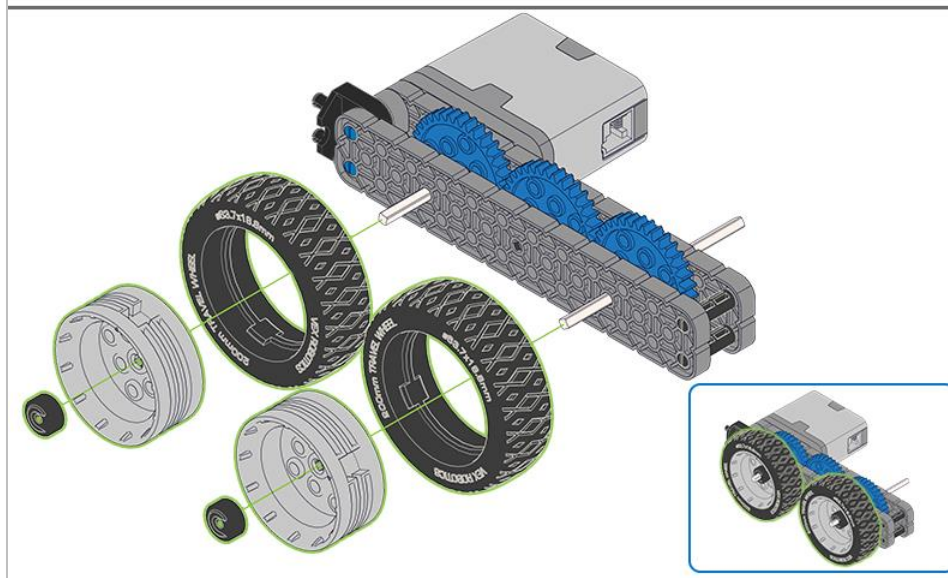
2x - Rubber Shaft Collars



2x - Wheel Hubs

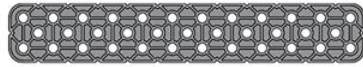


2x - Tires (200mm)



After attaching the wheels, twist the wheel that has the shaft going into the motor. If the wheel spins freely and without tension, the 4x Pitch Shaft has slipped out of place.

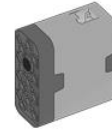
7



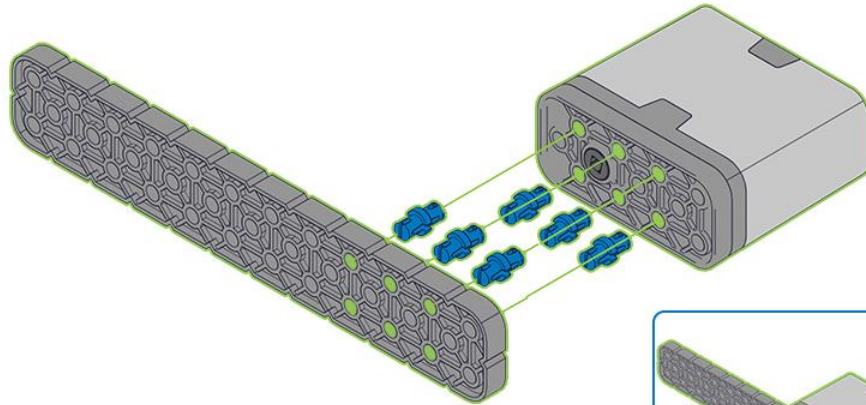
1x - 2x12 Beam



6x - 1x1 Connector Pins



1x - Smart Motor



8



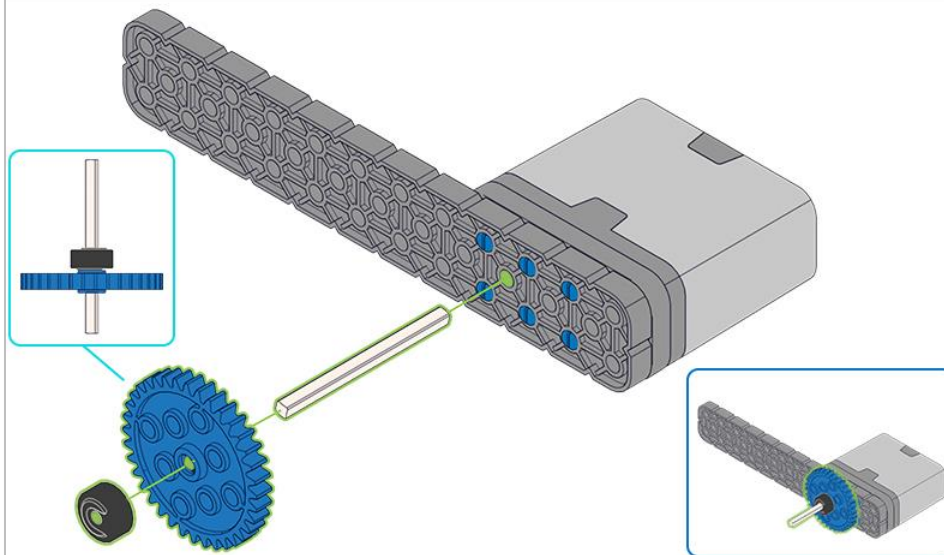
1x - 36 Tooth Gear



1x - Rubber Shaft Collar



1x - 4x Pitch Shaft

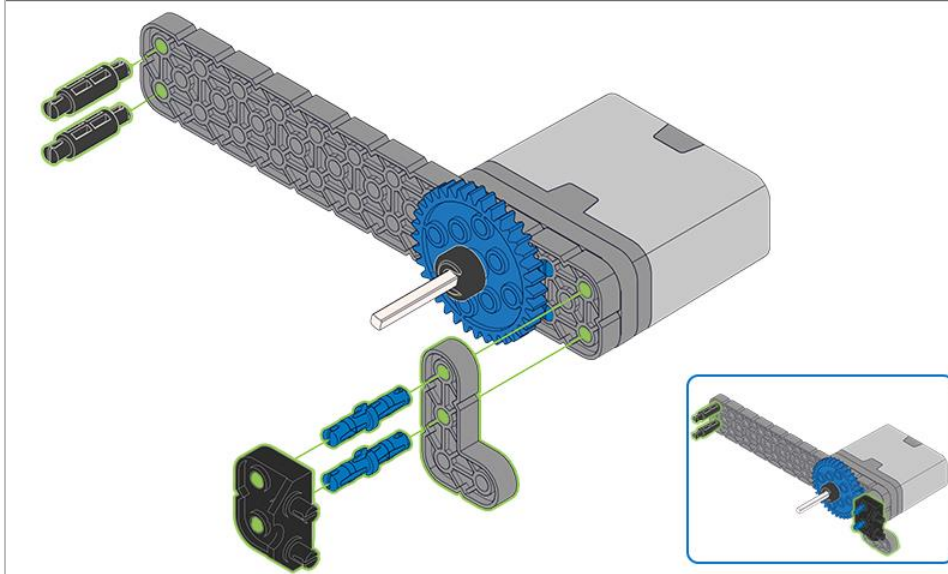


When adding the 4x Pitch Shaft, twist the pitch shaft to check for tension while turning. If it spins freely, it is not properly inserted into the motor.

9



1x - 2x Wide, 1x2 Offset Corner Connector 2x - 2x2 Connector Pins 1x - 2x3 Right Angle Beam 2x - 1x Pitch Standoffs



10



2x - 36 Tooth Gears

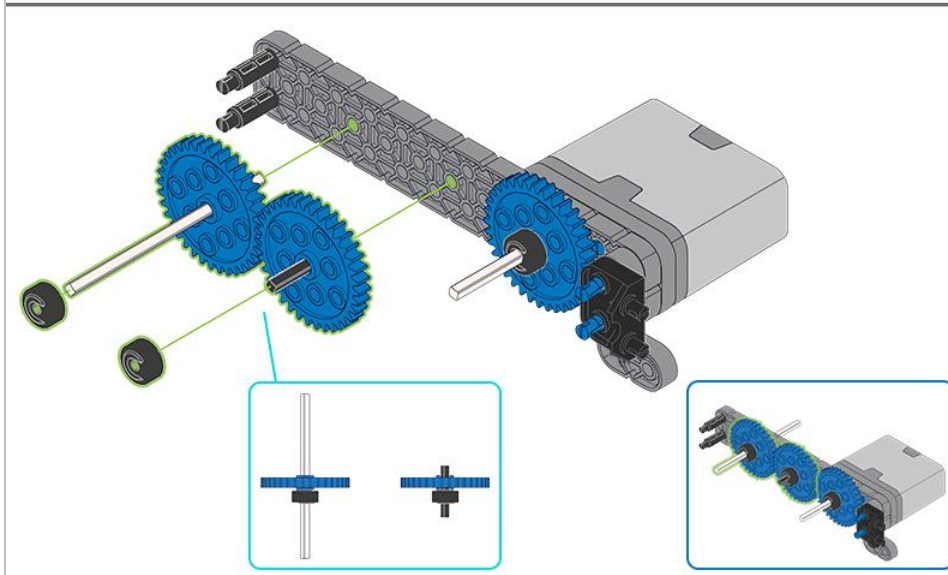


2x - Rubber Shaft Collars

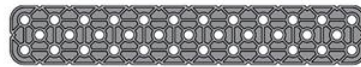
1x - 6x Pitch Shaft



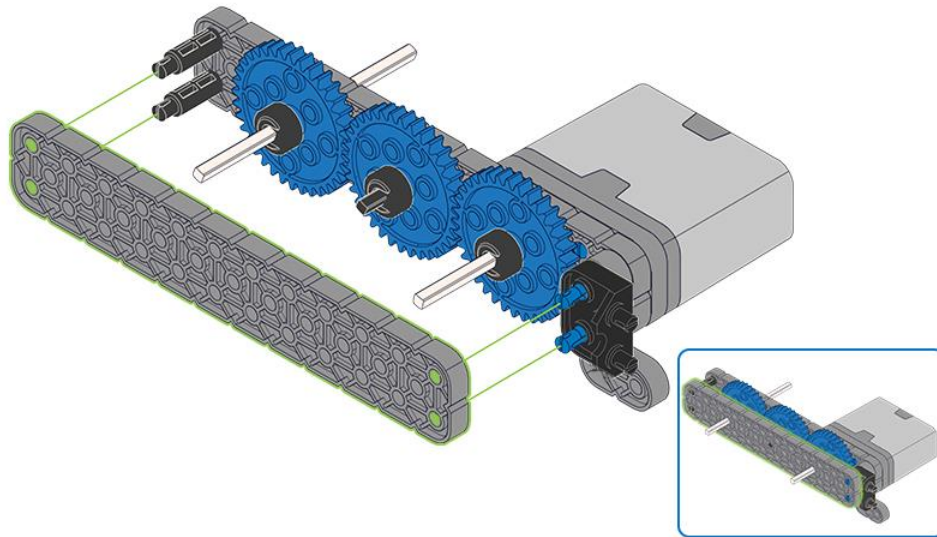
1x - 2x Plastic Pitch Shaft



11



1x - 2x12 Beam



Make sure the gears fit together properly before locking the Beam in place.

12



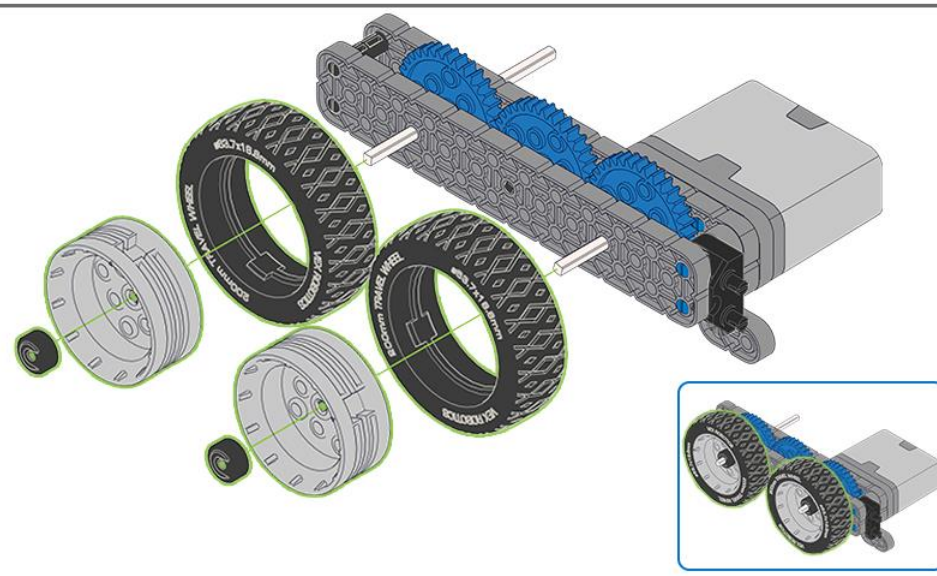
2x - Rubber Shaft Collars



2x - Wheel Hubs



2x - Tires (200mm)



After attaching the wheels, twist the wheel that has the shaft going into the motor. If the wheel spins freely and without tension, the 4x Pitch Shaft has slipped out of place.

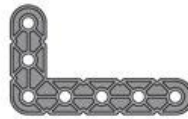
13



1x - 2x8 Beam



2x - 2x Wide, 1x2 Corner Connectors



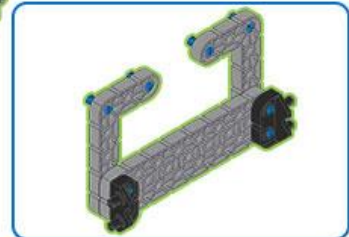
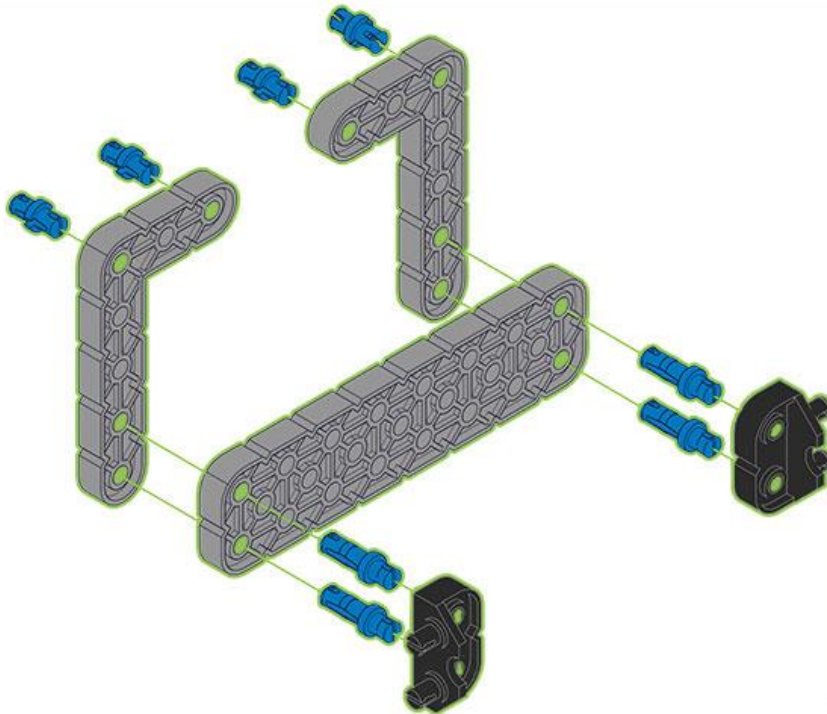
2x - 3x5 Right Angle Beams



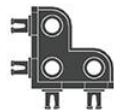
4x - 1x1 Connector Pins



4x - 1x2 Connector Pins



14

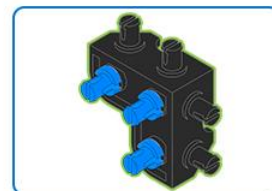
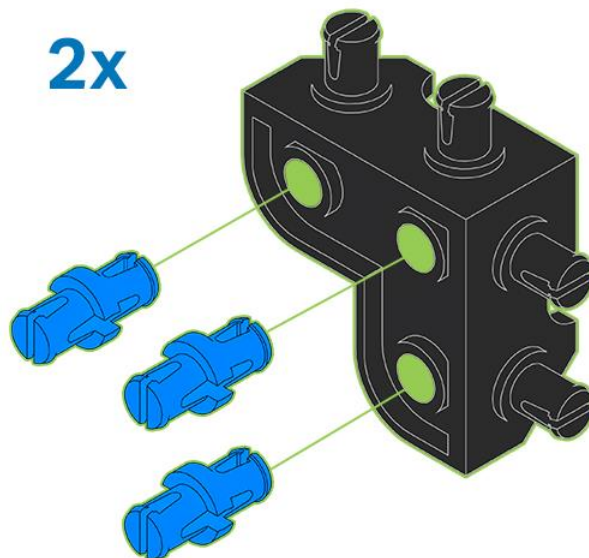


2x - Large Chassis Corner Connectors



6x - 1x1 Connector Pins

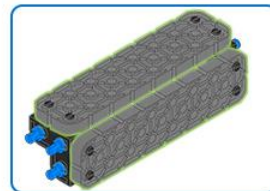
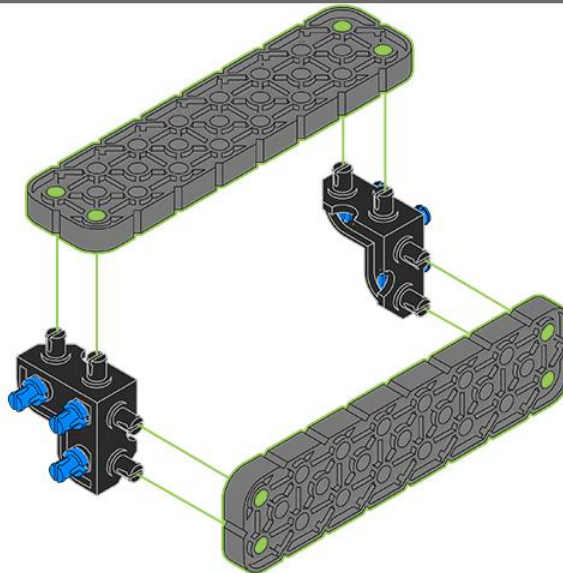
2x



15



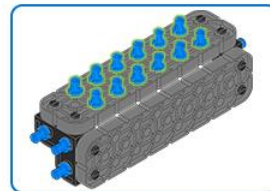
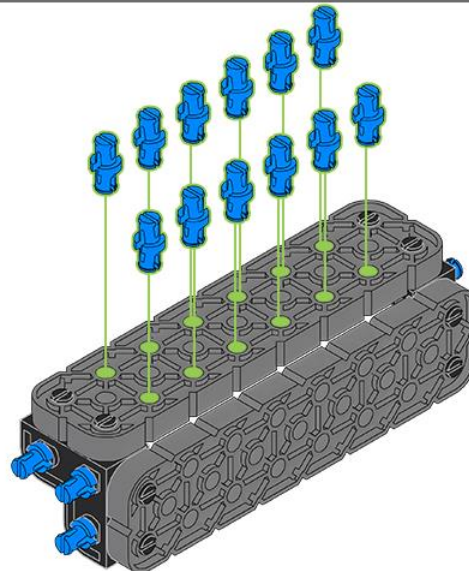
2x - 2x8 Beams



16



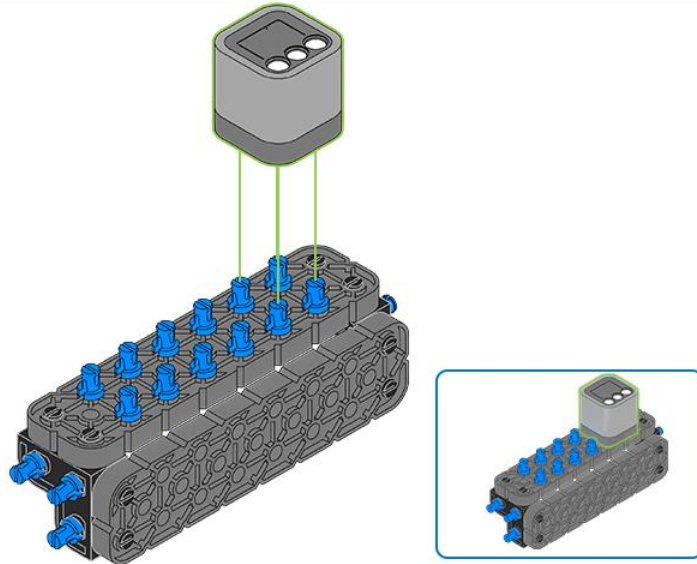
12x - 1x1 Connector Pins



17



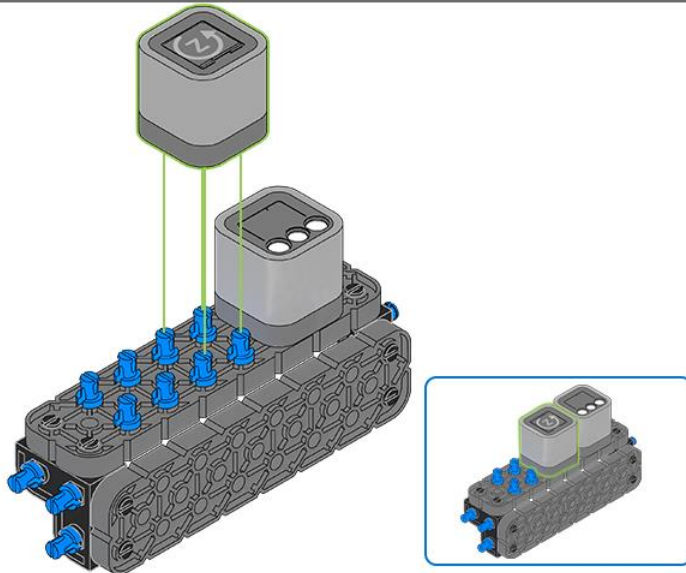
1x - Color Sensor



18



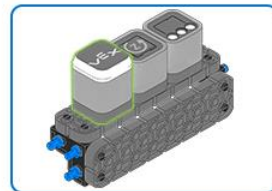
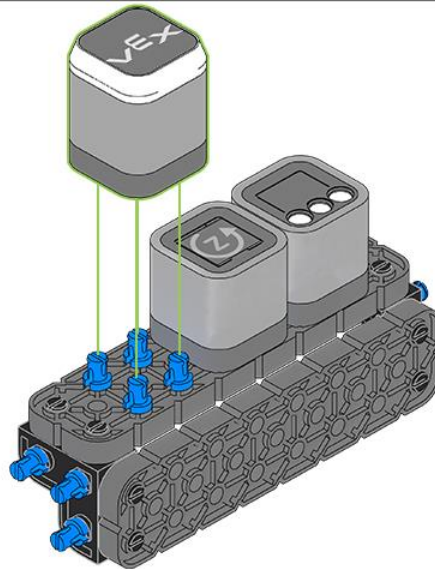
1x - Gyro Sensor



19



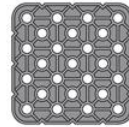
1x - Touch LED



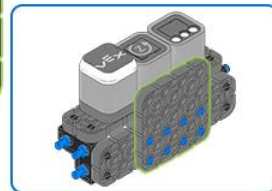
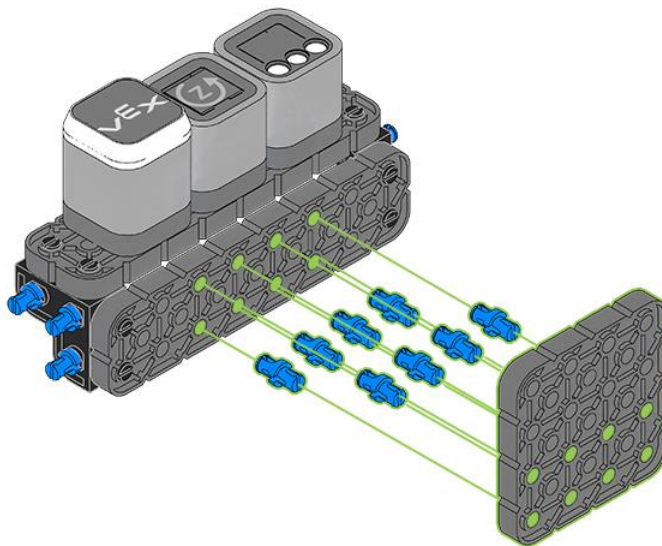
20



8x - 1x1 Connector Pins



1x - 4x4 Plate



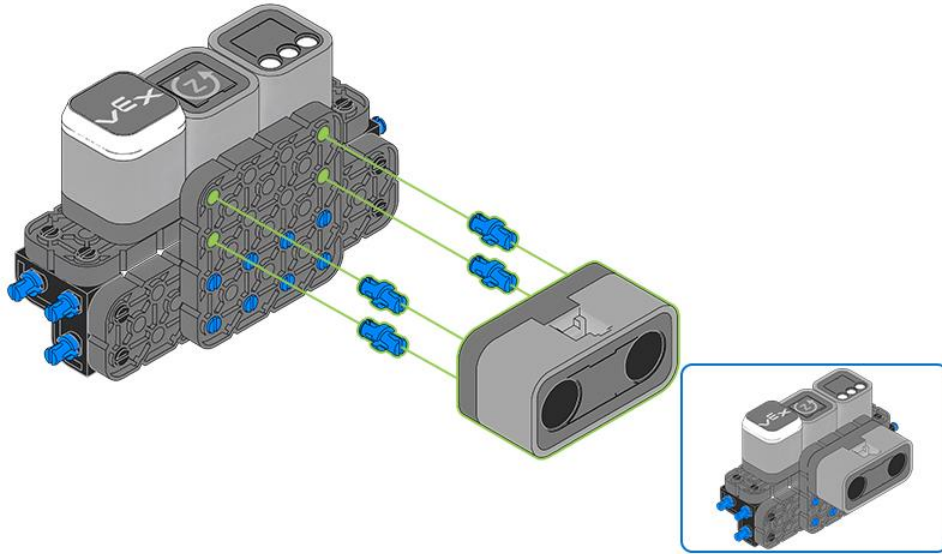
21



1x - Distance Sensor



4x - 1x1 Connector Pins



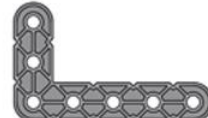
22



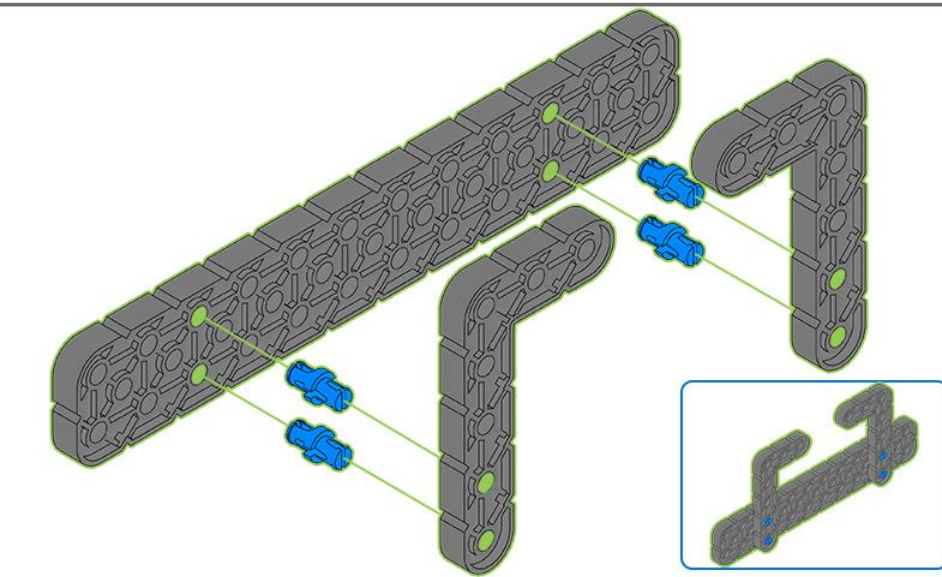
4x - 1x1 Connector Pins



1x - 2x12 Beam



2x - 3x5 Right Angle Beams



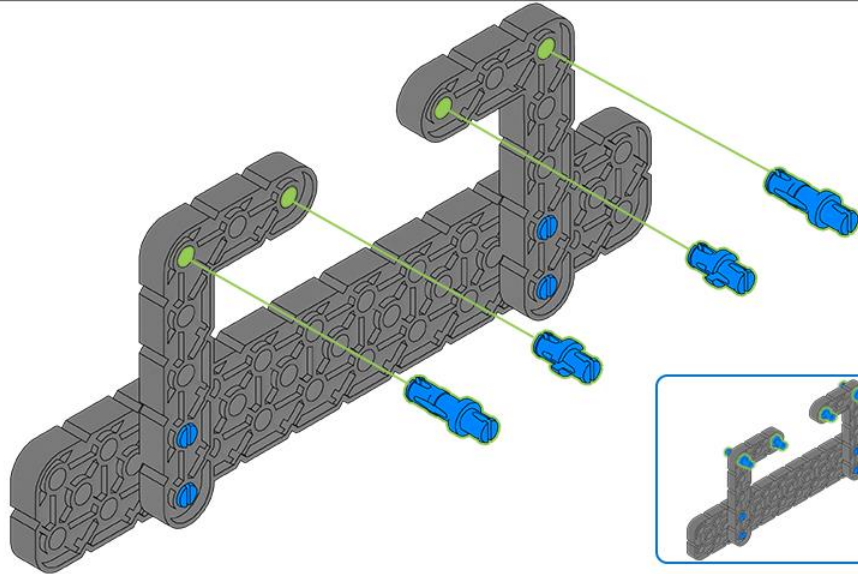
23



2x - 1x2 Connector Pins



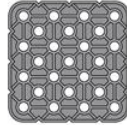
2x - 1x1 Connector Pins



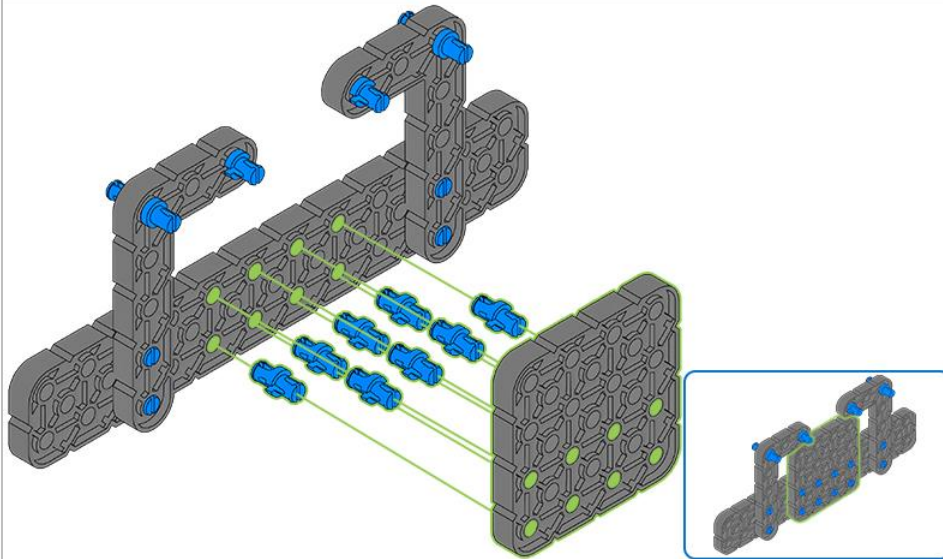
24



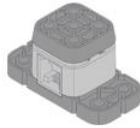
8x - 1x1 Connector Pins



1x - 4x4 Plate



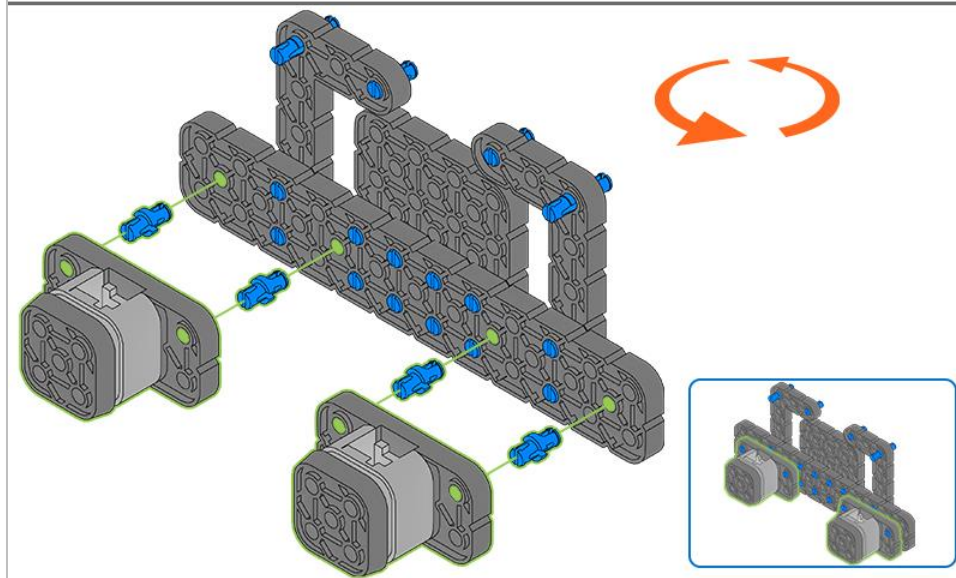
25



2x - Bumper Switch



4x - 1x1 Connector Pins



The orange arrows indicate to turn the assembly around.

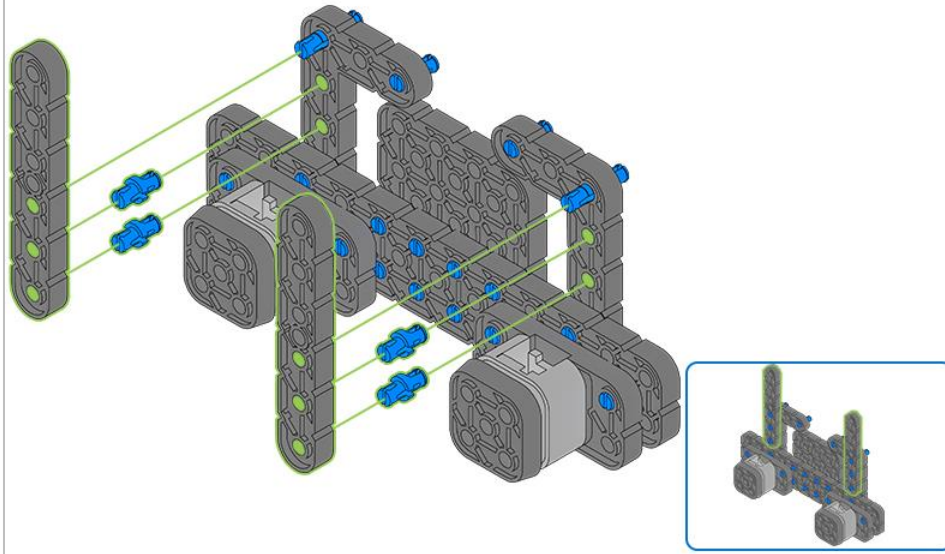
26



2x - 1x6 Beams



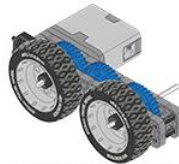
4x - 1x1 Connector Pins



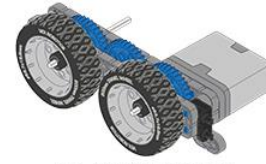
27



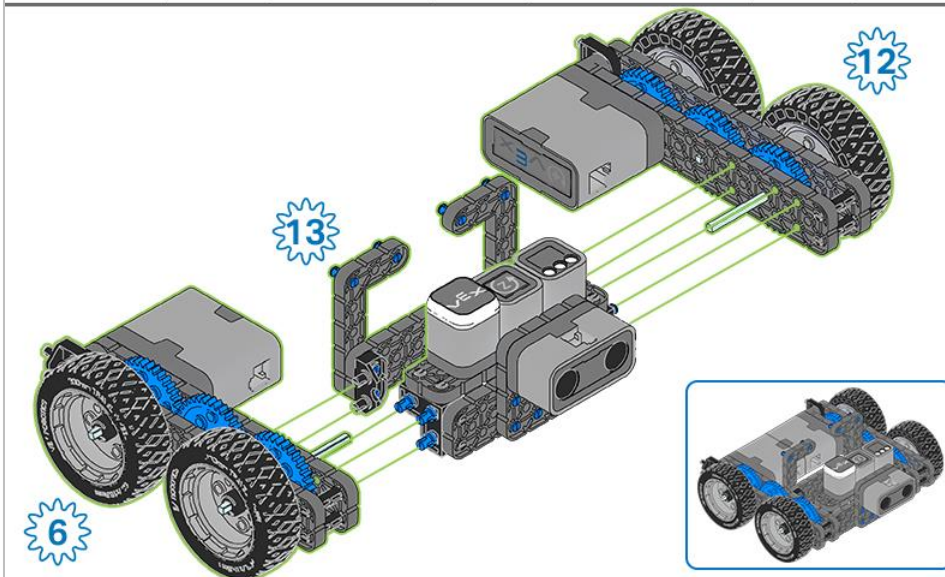
1x - Step 13 Assembly



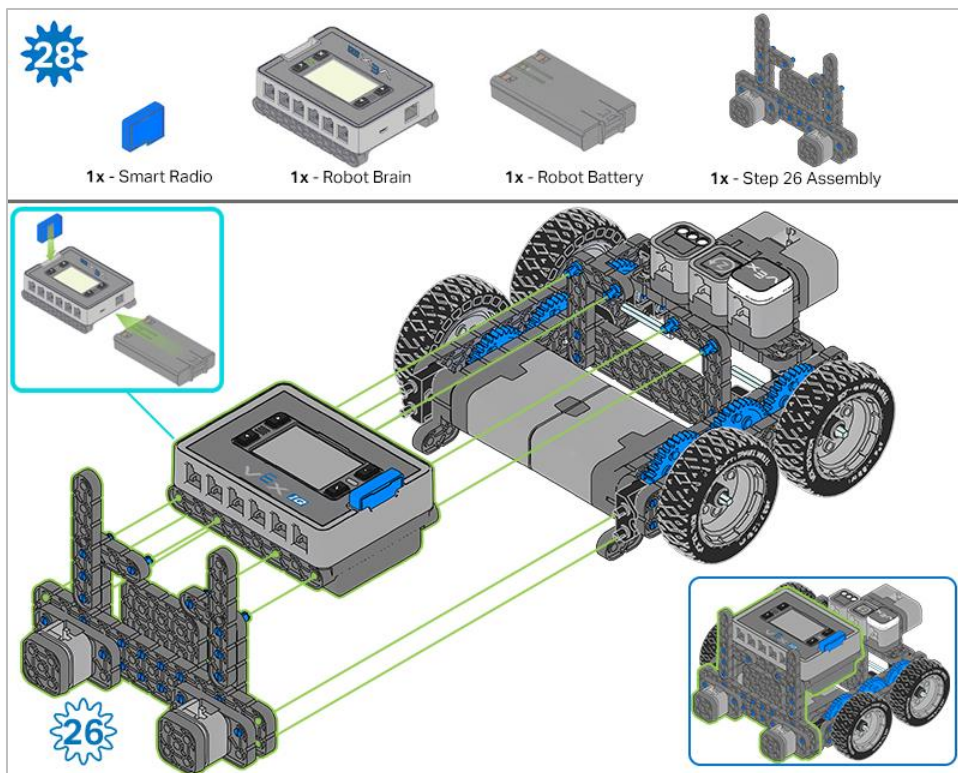
1x - Step 6 Assembly



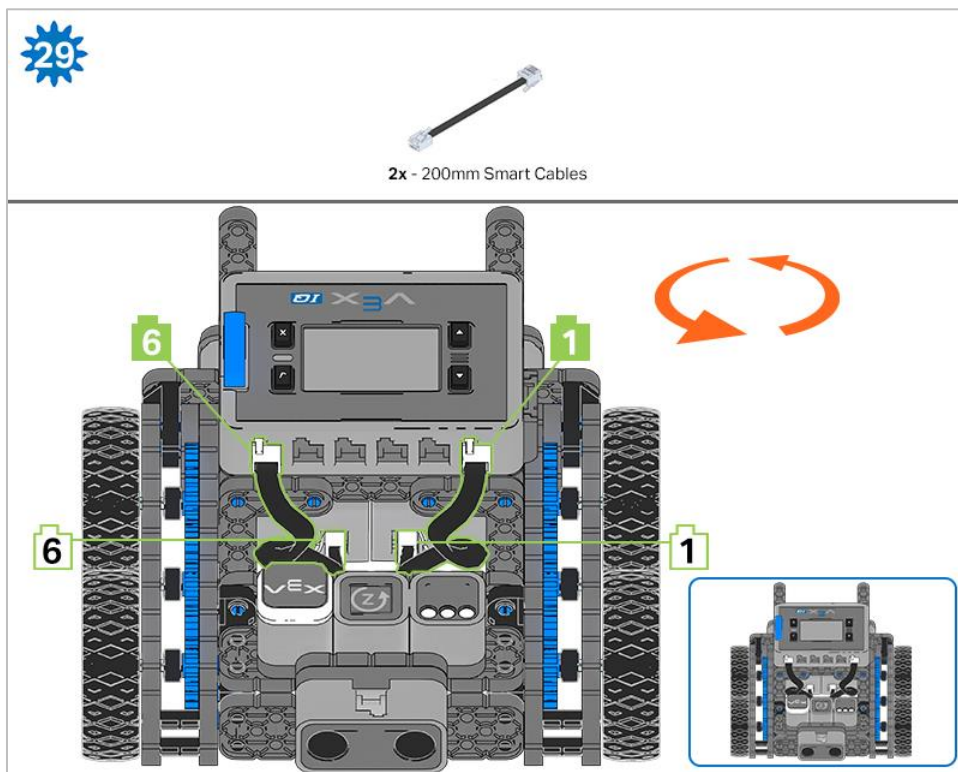
1x - Step 12 Assembly



The highlighted blue numbers placed in gear shapes represent the assemblies completed from those specific steps.



Ensure the Smart Radio and Robot Battery are inserted before attaching the Brain to the rest of the assembly.



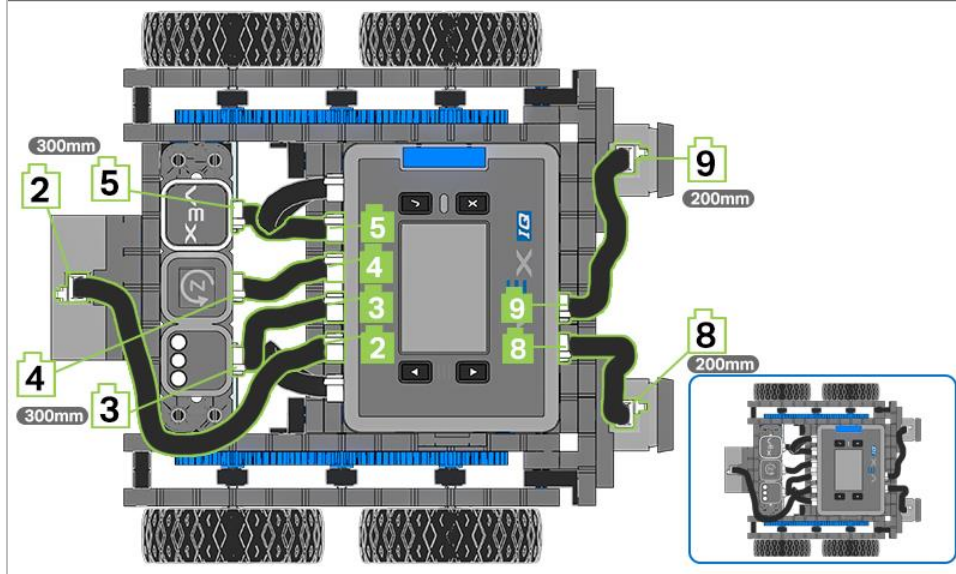
Steps 29-30, when you attach the Smart Cables, make sure they are tucked away so as to not block the Smart Sensors. The orange arrows indicate to turn the robot around.



4x - 300mm Smart Cables



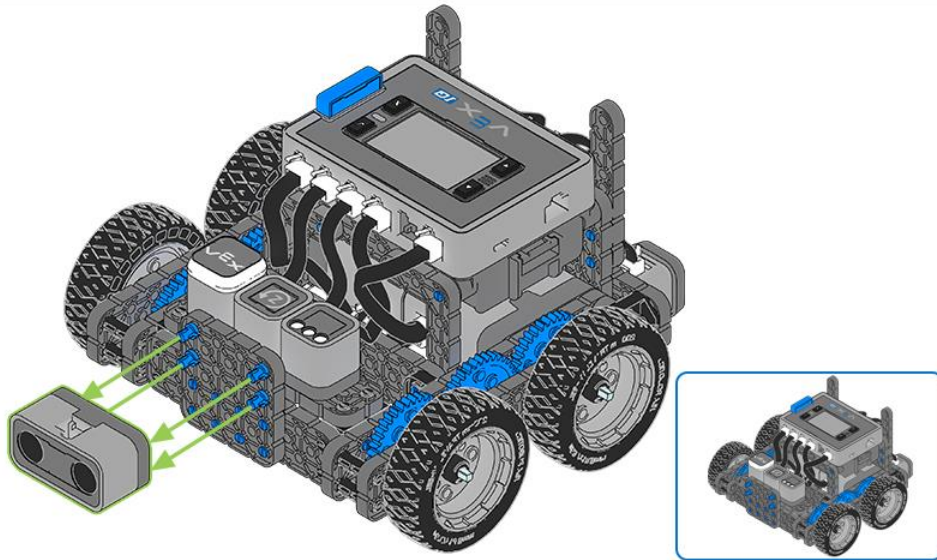
2x - 200mm Smart Cables



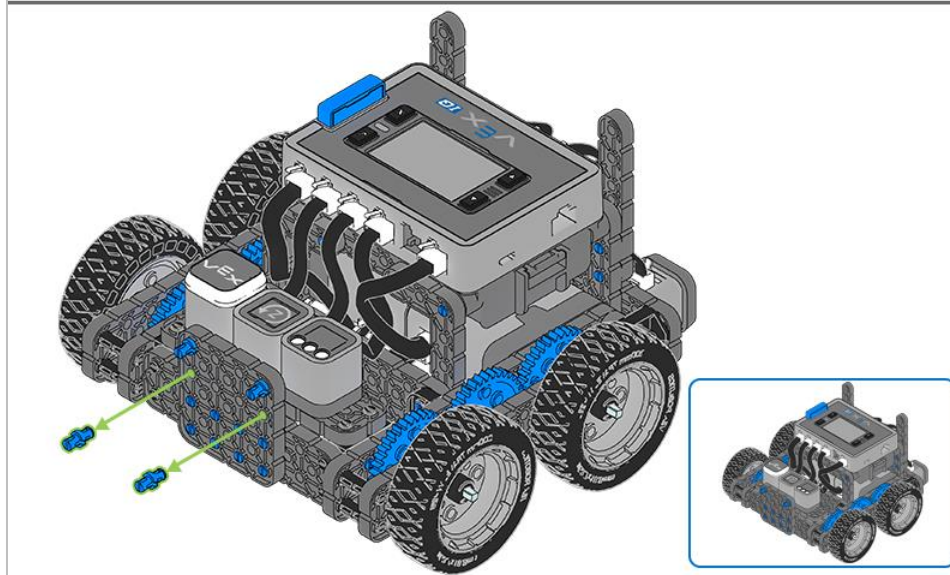
Adding the Vision Sensor

After the Autopilot has been assembled, use the building instructions below to add the Vision Sensor.

1



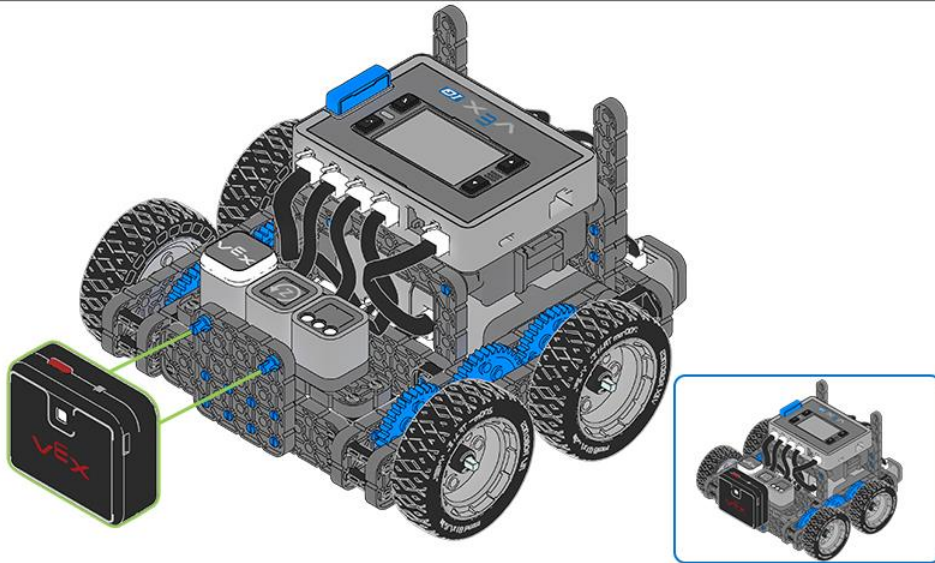
2



3



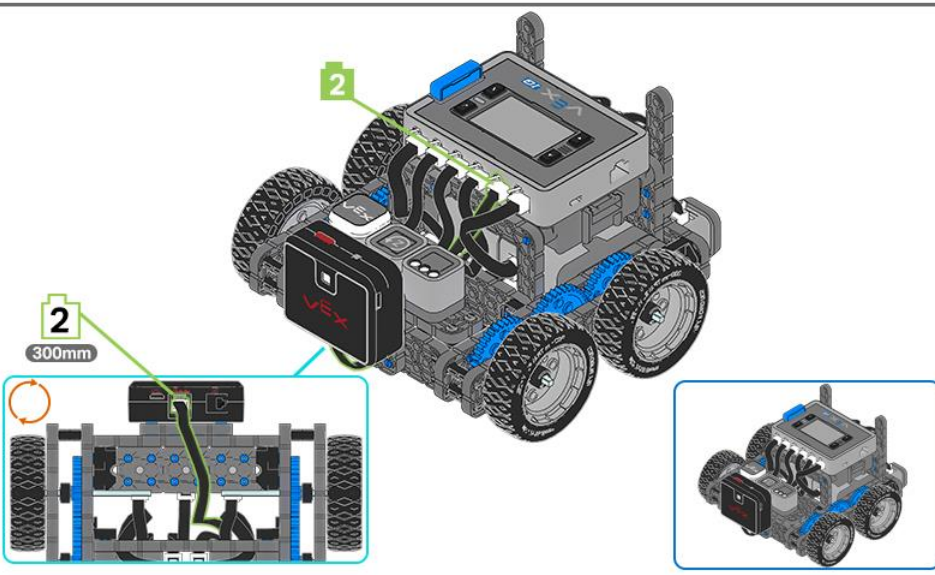
1x - Vision Sensor



4



1x - 300mm Smart Cable



Exploration

Now that the build is finished, explore and see what it can do. Then answer this question in your engineering notebook.

- Predict and describe how the Autopilot robot's behavior would change if the shaft in Step 2 of the robot Build Instructions was not inserted into the motor on one side of the robot, provide a diagram and a discussion of what purpose the rubber shaft collar on the motor shaft serves with your description.



Test your build, observe how it functions,
and fuel your logic and reasoning skills
through imaginative, creative play.

What is a Vision Sensor?

Description

The Vision Sensor allows your robot to collect visual data from a live feed. A live feed is a streaming transmission of what a video camera is capturing. The Vision Sensor is like a smart camera that can observe, select, adjust, and store colors and objects that appear in its visual field.



Vision Sensor 276-4850

Capabilities:

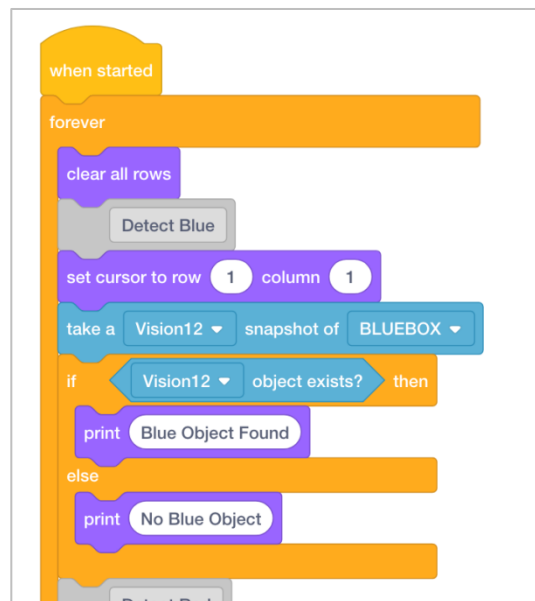
- This sensor can be used for recognizing colors and color patterns.
- This sensor can be used to follow an object.
- This sensor can be used to collect information about the environment.

The Vision Sensor allows the robot to use visual input data from its environment. The project can then determine how the visual input data should affect the robot's behavior. For example, the robot could perform actions (output) such as spinning motors or displaying results on the LCD screen.

The Vision Sensor can also capture a snapshot of what is in front of it and analyze it

according to what the user is asking. For example, a user can gather data from the snapshot such as, what color is the object? Is there an object detected at all? How large is the object (width and height)?

The robot can then make decisions based off of this data. The partial example project below shows how this is done. In this first part of the example project, the robot will print "Blue Object Found" if a blue object is detected and "No Blue Object" otherwise. That is the first of three decisions within the example project but the second and third decisions are not shown here.



Using the Vision Sensor

The Builder in each group should get the hardware required. The Recorder should get the group's engineering notebook. The Programmer should open VEXcode IQ Blocks.

Hardware/Software Required:

Quantity	Hardware/Other Items
1	VEX IQ Super Kit
1	VEXcode IQ Blocks (latest version, Windows, MacOS, Chromebook, iPad)
1	Engineering Notebook
1	Configuring the Vision Sensor (Tutorial)
1	Tuning the Vision Sensor (Tutorial)
1	Detecting Objects example project

This activity will give you the tools to use the Vision Sensor.

You can use the Help information inside of VEXcode IQ Blocks to learn about the blocks. For guidance in using the Help feature, see the Using Help tutorial.



1. Preparing for the activity

Before you begin the activity, do you have each of these items ready? The Builder should check each of the following:

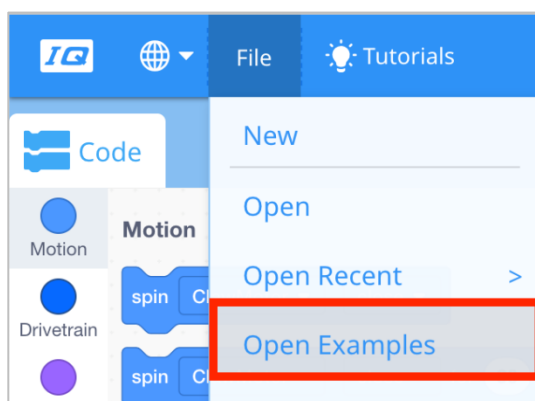
- Are all the motors and sensors plugged into the correct port?
- Are the smart cables fully inserted into all of the motors and sensors?
- Is the Brain turned on?
- Is the battery charged?

2. Open an Example Project.

- VEXcode IQ Blocks contains many different example projects. You'll use one of them in this exploration. For help and tips on using example projects, check out the Using Examples and Templates tutorial.

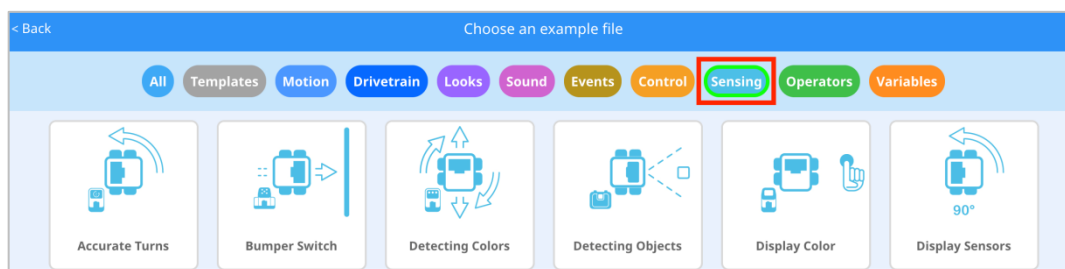


Then, open the Detecting Objects example project.



The Programmer should complete the following steps:

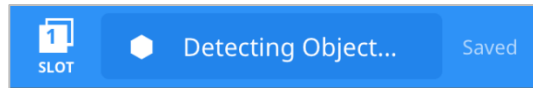
- Open the File menu.
- Select **Open Examples**.
- Use the filter bar at the top of the application and select "Sensing."



- Select and open the **Detecting Objects** example project.



- **Save** your project as Detecting Objects.

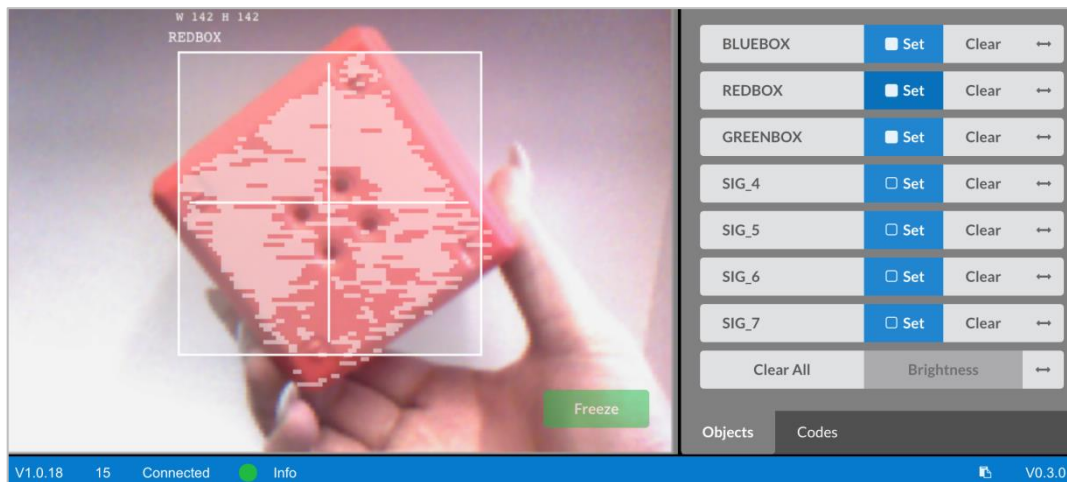


- Check to make sure the project name **Detecting Objects** is now in the window in the center of the toolbar.
- For addition help, view the Use Example Projects and Templates tutorial video.



3. Configuring and Using the Vision Sensor

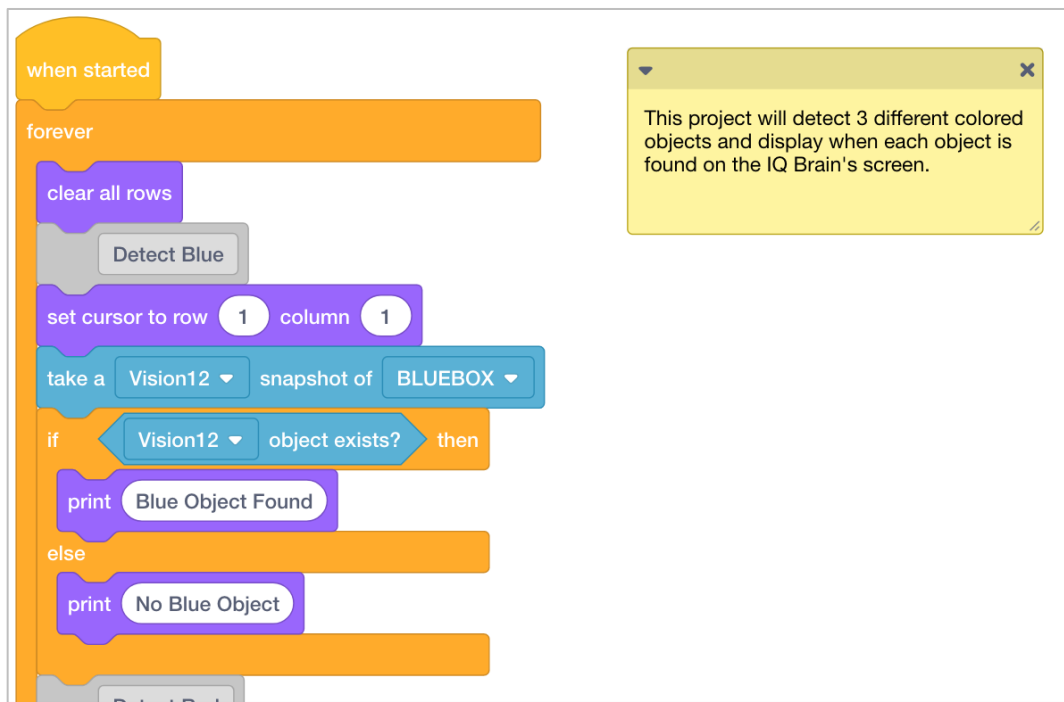
- Begin by watching the Configuring the Vision Sensor tutorial video.
- Next, configure the Vision Sensor for three colored objects: red, green, and blue.



- Have the Programmer open the previously saved Detecting Objects example project.



- What is this project actually doing? Predict what the Autopilot will do and have the Recorder write down the predictions in your engineering notebook.



- Have the Driver Download and Run the project. Have the Builder place different colored objects in front of the Vision Sensor and observe the robot's behavior. Have the Recorder record in your engineering notebook how your prediction was different or correct compared to what you actually observed from the project.

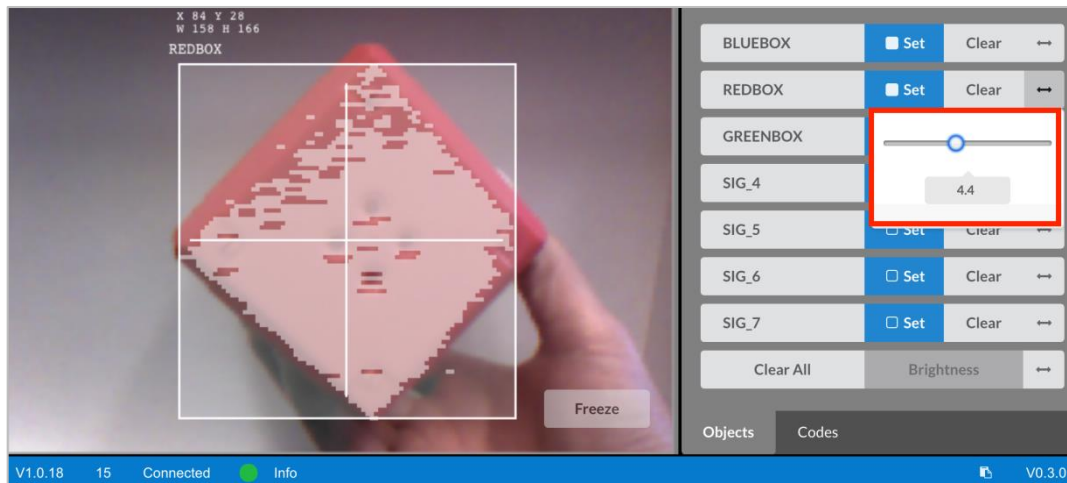
For additional help, view the Download and Run a Project tutorial video.



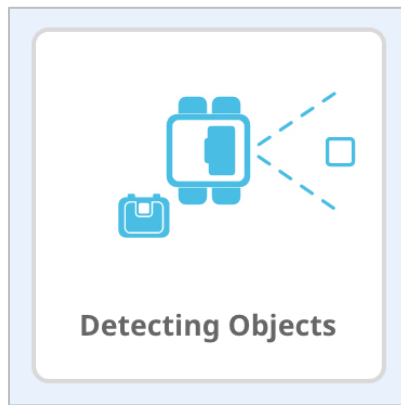
4. Tuning the Vision Sensor

Often times an object is configured to be recognized by the Vision Sensor in one environment, for example, in a classroom. When the Vision Sensor is then taken into a different environment such as a competition setting, the object may not be recognized by the Vision Sensor. This is often due to a change in lighting after the Vision Sensor has already been configured. To solve this problem, you may have to tune your Vision Sensor.

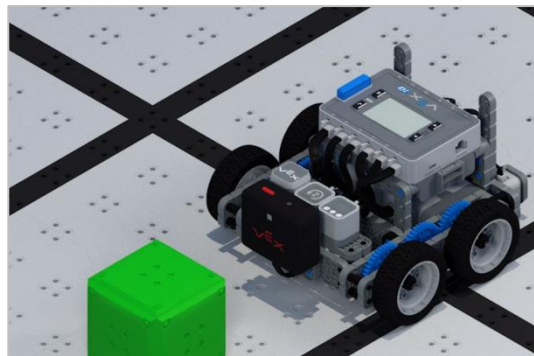
- Begin by watching the Tuning the Vision Sensor tutorial video.
- Next, Tune the Vision Sensor for the three colored objects: red, green, and blue.



- Have the Programmer open the previously saved Detecting Objects example project.



- How will tuning the Vision Sensor affect how well it can detect objects? Have the Builder take the Autopilot to a different part of the room with more or less light.



- Have the Driver Download and Run the project. Have the Builder place different colored objects in front of the Vision Sensor and observe the robot's behavior. Have the Recorder document in your engineering notebook how well the Vision Sensor detects objects. Does the Vision Sensor need tuned after it changed locations?

For additional help, view the Download and Run a Project tutorial video.



- Tune the Vision Sensor as necessary. Test the Vision Sensor after it has been tuned to determine if it can detect objects better and make adjustments as needed.



Become a 21st century problem solver
by applying the core skills and concepts
you learned to other problems.

Perception with Self-Driving Vehicles

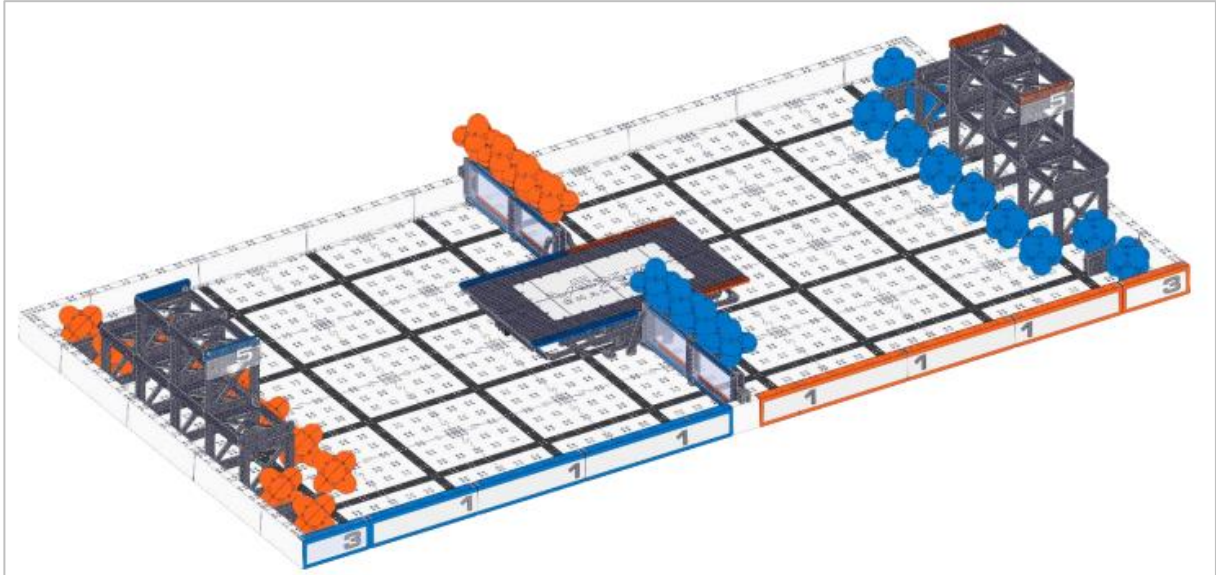


A self-driving vehicle with an array of sensors

Perception with Self-Driving Vehicles

Self-driving vehicles use a wide array of sensors to perceive their surroundings, on-board computers to combine and process the data from all of those sensors, and one or more motors to safely move the vehicle along the road using AI (artificial intelligence). Self-driving vehicles often make use of high resolution cameras to be able to recognize things such as pedestrians, road signs, and other vehicles. Vision data from the cameras is often combined with sensors that use lasers to measure distance between the vehicle and other objects. This allows the on-board computers to make decisions based on what types of objects are in their environment, and how far away each object is.

Competition Connection: Crossover



VRC 2016-2017 Crossover Field

Robot Capabilities

The 2016 - 2017 VEX Robotics Competition game Crossover required players to place their colored Hexballs in their scoring zones. There were 28 Hexballs in total: 14 of each color (blue and orange) per team. Points were earned by placing the team's correctly colored Hexball in their goal zone.

Although it was illegal to use a Vision Sensor in a VEX IQ Robotics Competition in 2016-2017, you can imagine that competition teams could have benefited from using the Vision Sensor to detect the team-specific colored Hexball. It would be easier for the robot to pick up the correctly colored Hexball during the autonomous period if the robot was programmed to move towards and grab a certain colored object. If the robot selects the incorrect colored Hexball during the autonomous period, there is a possibility that the robot would not score as many points.

Similarly, for the Driving Skills Challenge, it may be hard for teams to manually line up the robot enough to place the Hexball in the goal. The Vision Sensor could be used to align the

robot more accurately.

Overall, the Vision Sensor would have greatly helped skilled teams to program the robot to use the Vision Sensor to detect Hexballs and align the robot properly in order to place the Hexballs into the goals.



Is there a more efficient way to come to
the same conclusion? Take what you've
learned and try to improve it.

Prepare for the Vision Data Challenge

The Vision Sensor's Sensing Blocks

VEXcode IQ Blocks has Sensing blocks for the Vision Sensor. The first two you already used in the Play section to take a snapshot and to check if the object exists.

In the figure below, you see that the *snapshot* block captured the GREENBOX snapshot. The object, GREENBOX, was identified in the snapshot and so the answer of whether it exists is TRUE.

Let's look at these other Sensing blocks and what their values tell us.

The screenshot shows a VEXcode IQ Blocks workspace. At the top left, a 'take a snapshot of' block is configured with 'Vision12' and 'GREENBOX'. An arrow points from this block to a camera view. The camera view shows a green rectangular object labeled 'GREENBOX' on a grey background. The object is centered within a white bounding box. The coordinates (0,0) and (315,0) are shown at the top, and (0,211) and (315,211) at the bottom. To the right of the camera view, there are six 'Vision12' object sensing blocks, each with a dropdown menu set to 'object'. The first block is 'object exists?' with a value of 'TRUE'. The second is 'object count' with a value of '1'. The third is 'object centerX' with a value of '154'. The fourth is 'object centerY' with a value of '105'. The fifth is 'object width' with a value of '140'. The sixth is 'object height' with a value of '142'.

Block	Value
Vision12 object exists?	TRUE
Vision12 object count	1
Vision12 object centerX	154
Vision12 object centerY	105
Vision12 object width	140
Vision12 object height	142

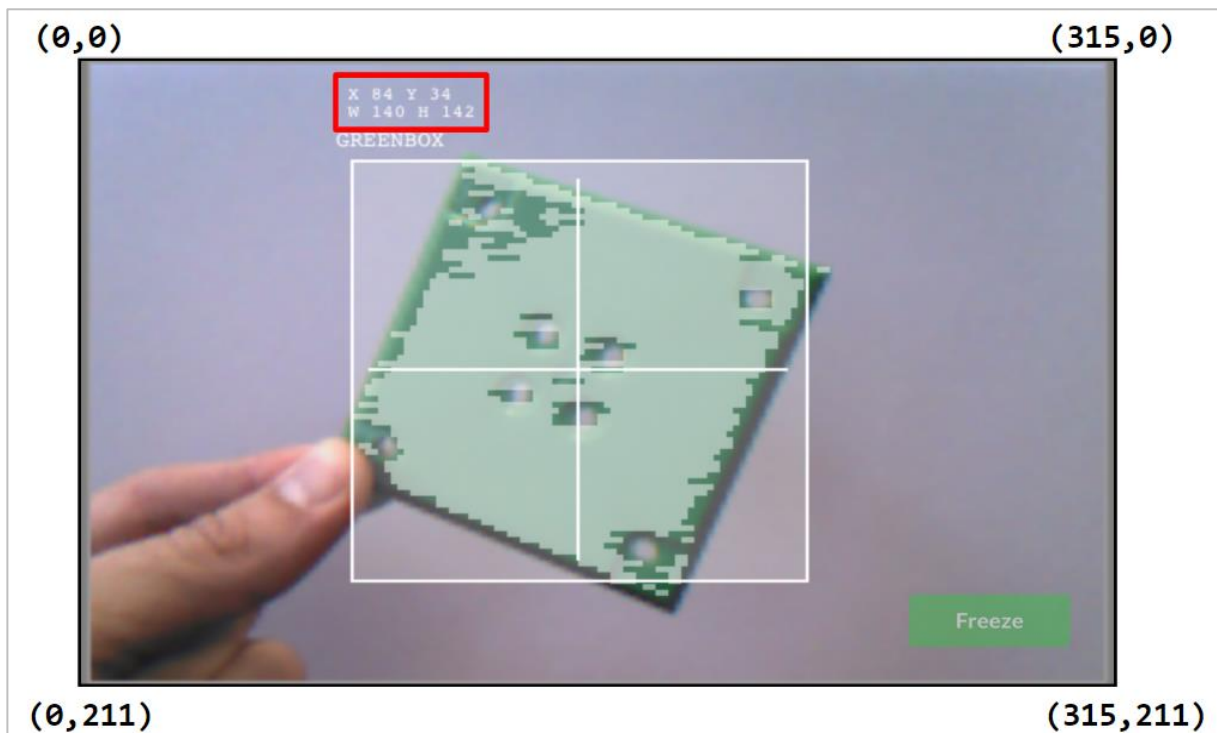
- The *object count* block tells us how many GREENBOX objects are in the snapshot. Here, there is only 1 detected.
- The center X value tells us whether the GREENBOX object is to the left or right of the robot's center point. Remember, the Vision Sensor is mounted in the middle of the front of the robot and so the snapshot's view is the robot's view.
 - If center X is greater than 157.5, the object is to the right of the robot's center point.
 - If center X is less than 157.5, the object is to the left of the robot's center point.
- The center Y value tells us whether the GREENBOX is higher or lower than the robot's center point.

- If center Y is greater than 105.5, the object is lower than the robot's center point.
- If center Y is less than 105.5, the object is higher than the robot's center point.
- The width and height values tell us how close the GREENBOX is to the robot.
 - The same-sized object will be larger in width and height as it gets closer to the robot.

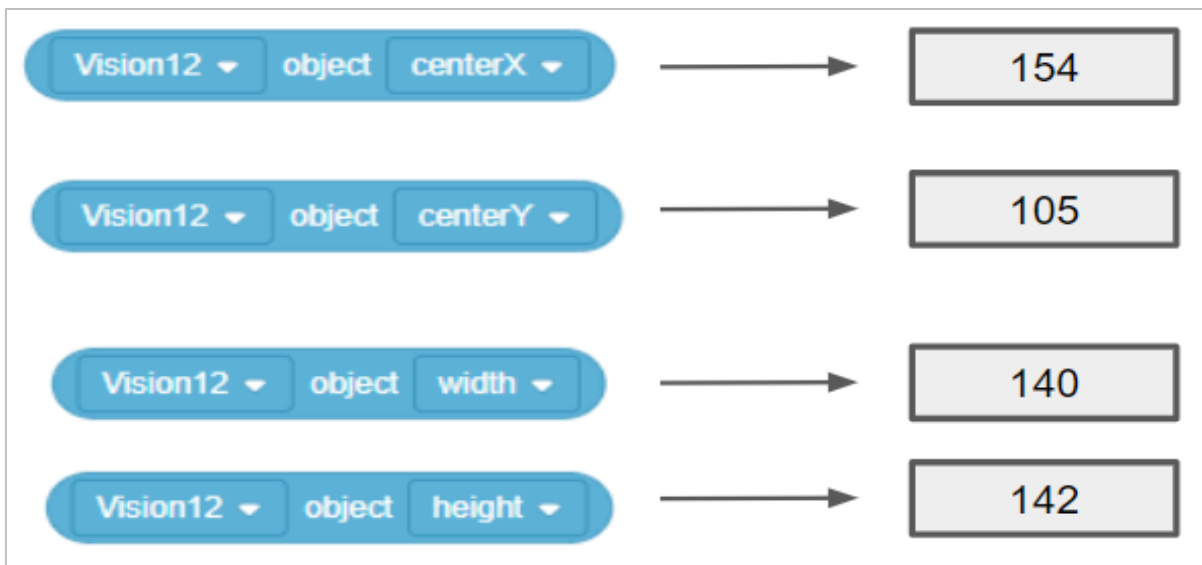
How are the center X and center Y values calculated?

The values are calculated based on the coordinates within the snapshot. The width and height of the object are already calculated.

The Vision Sensor tracks the X and Y values of the upper left corner of the object. Below, those coordinates are (84, 34).

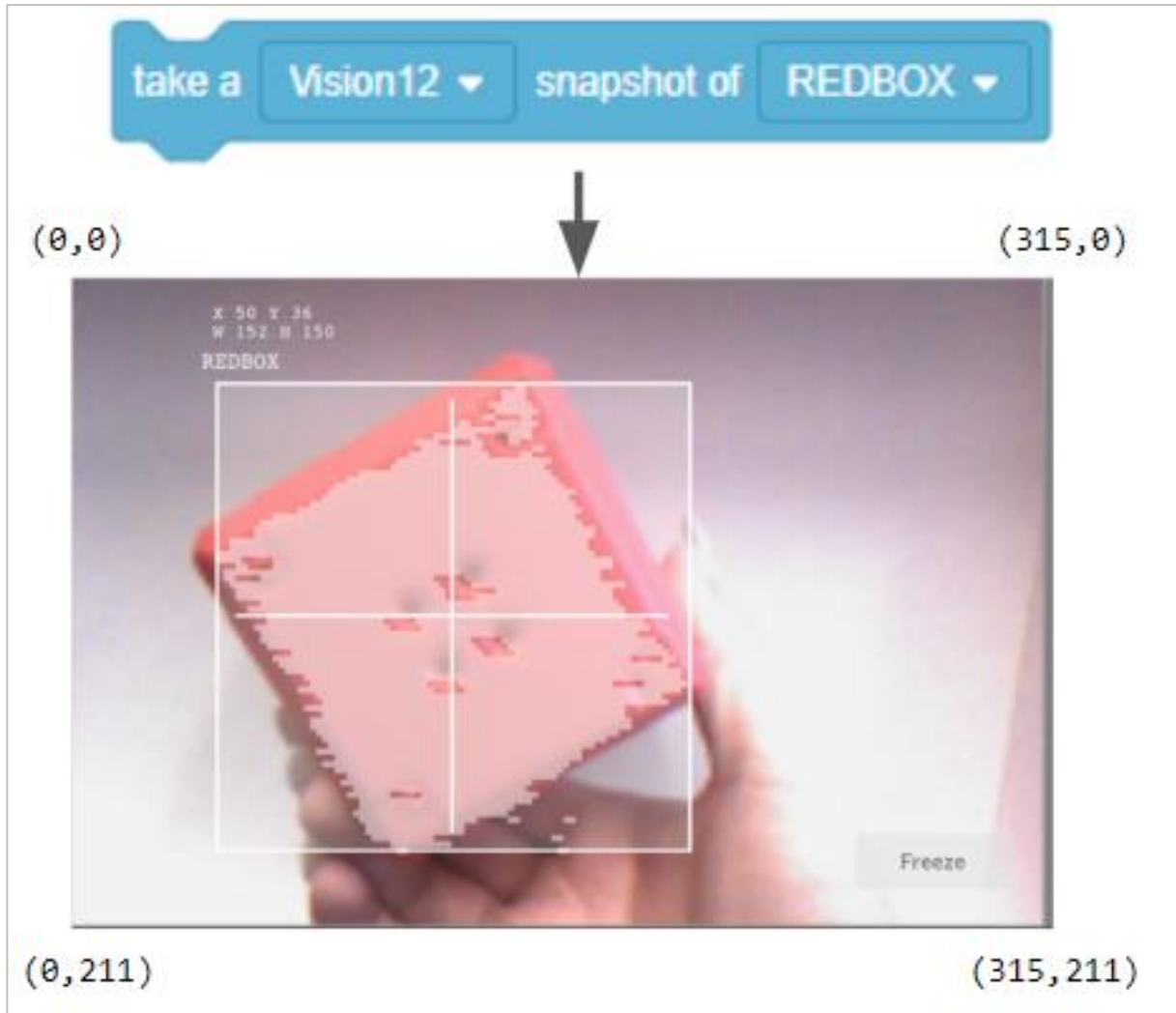


The center X and center Y values can be calculated based off of the coordinates of the upper left corner (84, 34), and the width (W 140) and height (H 142) values provided.



- $\text{centerX} = 140/2 + 84 = \mathbf{154}$
 - centerX = half the width of the object added to its leftmost X coordinate
- $\text{centerY} = 142/2 + 34 = \mathbf{105}$
 - centerY = half the height of the object added to its topmost Y coordinate

Practice for the Vision Data Challenge

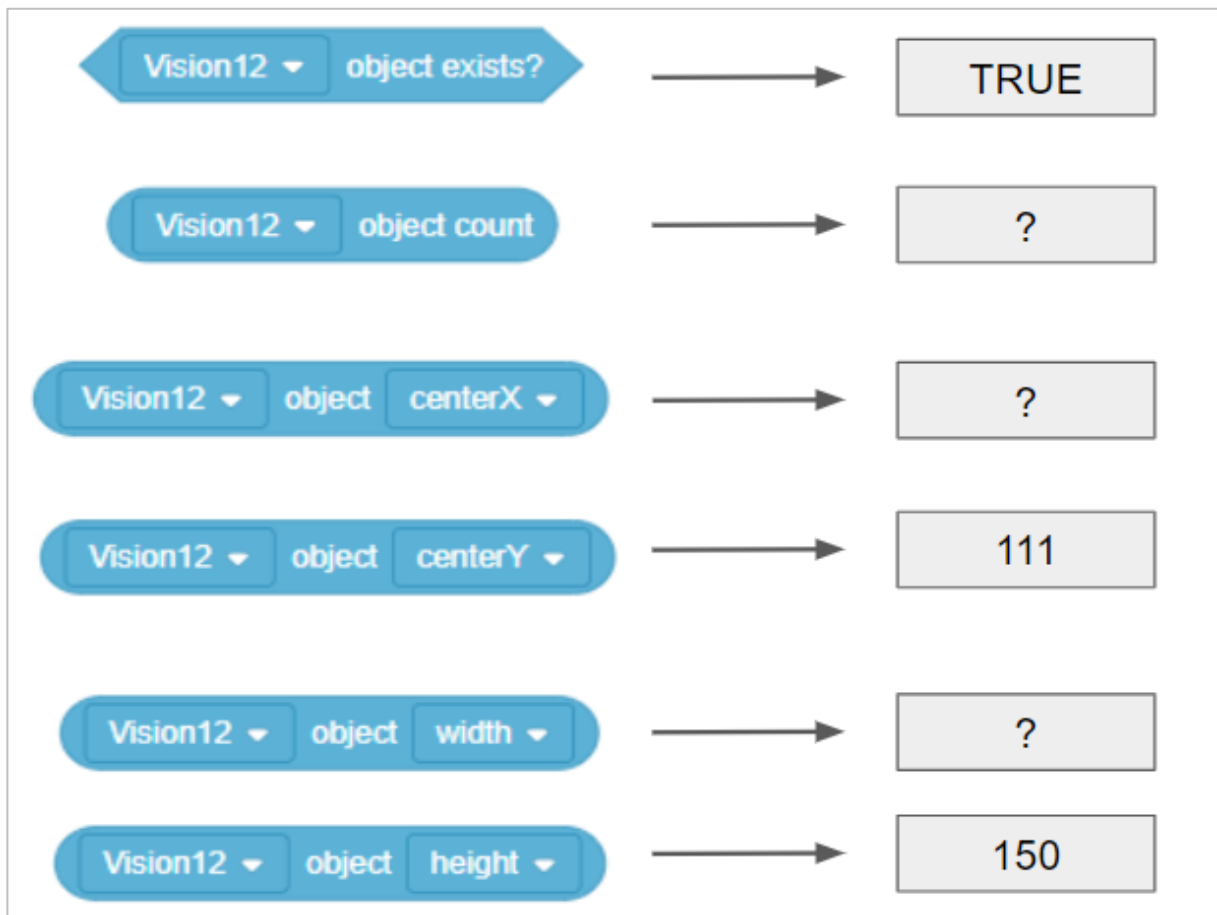


Add the missing values below in your engineering notebook.

Here are the provided data from the snapshot:

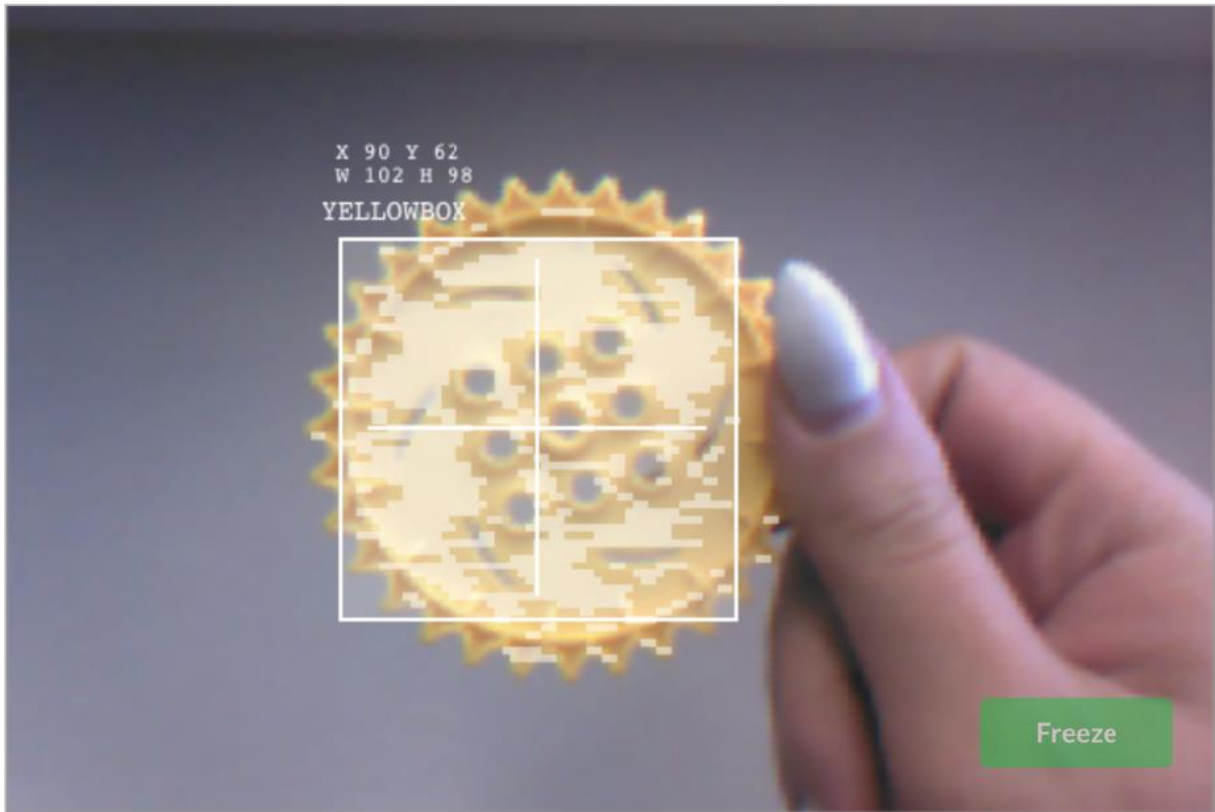
- $X = 50$
- $Y = 36$
- $W = 152$

- $H = 150$



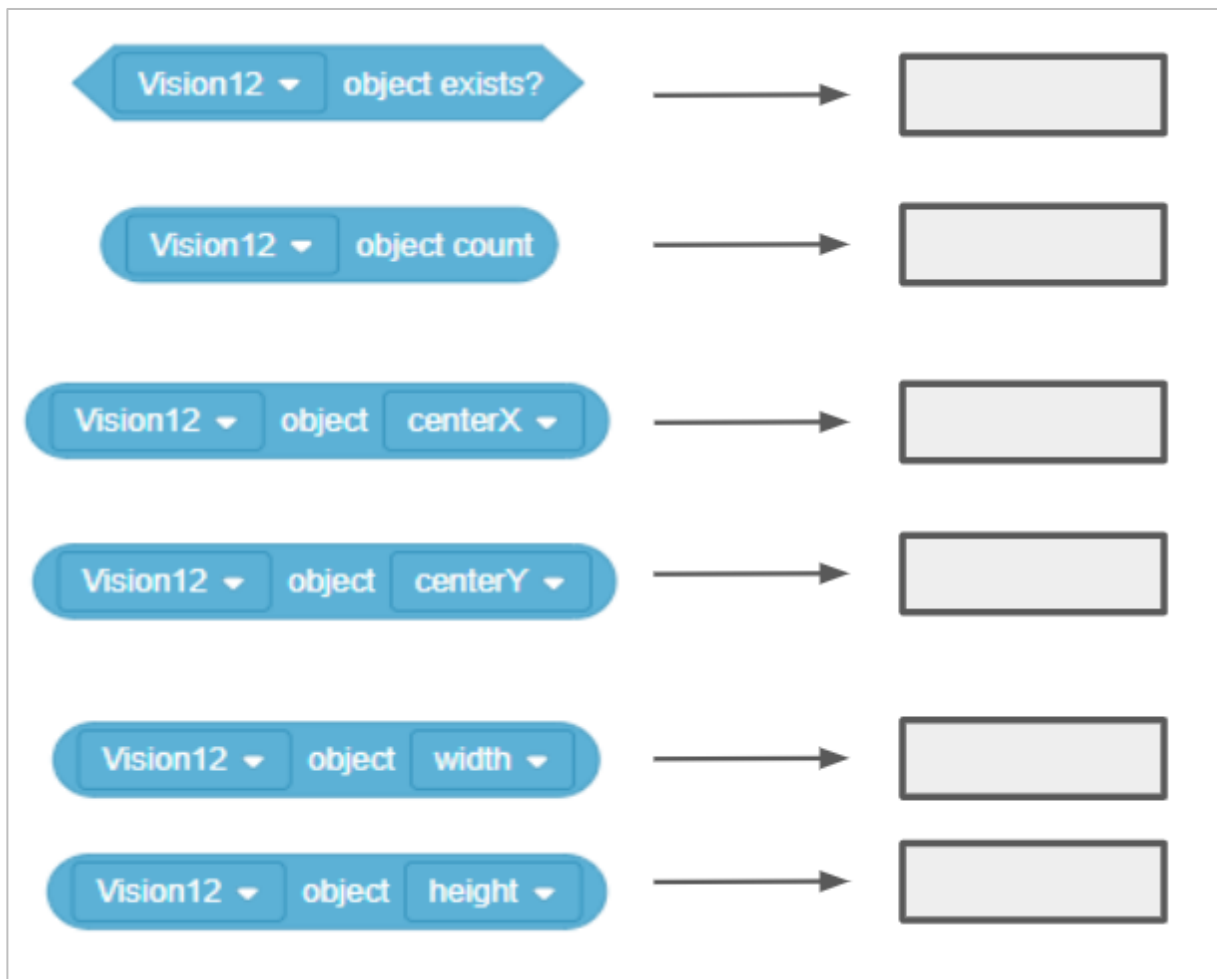
- Is the REDBOX to the left or to the right of the robot's center point?
- Is the REDBOX higher or lower than the robot's center point?

The Vision Data Challenge



Complete the Vision Data Challenge by answering the questions and filling in the missing data in your engineering notebook.

- Which of these blocks was used to take the snapshot above?
 -
 -
- Fill in these values:



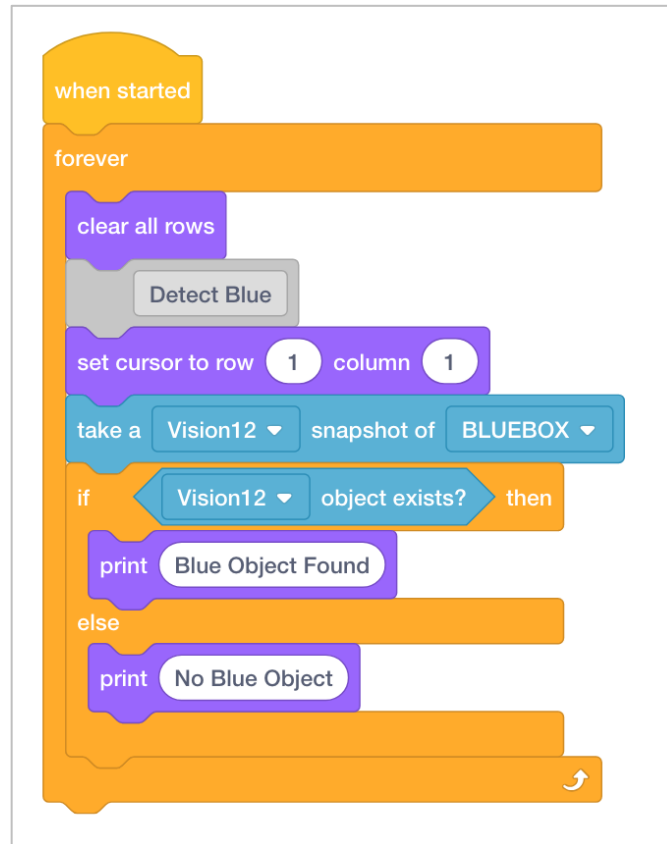
- Is YELLOWBOX to the left or to the right of the robot's center point?
- Is YELLOWBOX above or below the robot's center point?
- YELLOWBOX is **NOT** the best name to give this object if you want to easily recognize which color signature is which. Which of these is a better name? Why?
 - YELLOWGEAR
 - YELLOWCUBE



Understand the core concepts and how
to apply them to different situations.
This review process will fuel motivation
to learn.

Review

1. What does the **snapshot** block do in this example?



- Streams a continuous video of what the Vision Sensor sees
- Prints the color of the object on the Brain's screen
- Determines if an object exists or not
- Takes a snapshot of the current image from the Vision Sensor so that it can be analyzed

2. Which of the following should come first when configuring a signature for the Vision Sensor?

- Still the image so an area of color can be selected
- Select the "Set" button

- Clear the signature
- Place the object in view of the Vision Sensor

3. Which of the following is NOT an example of tuning the Vision Sensor?

- Adjusting the color signature slider
- Adjusting the brightness
- Resetting the color signature of adjusting the slider and brightness does not help
- Calculating the center x

4. Why is a `forever` block used in the Detecting Objects example project?

- "Blue object found" should be printed forever
- The snapshot block only takes one current image of what the Vision Sensor sees. Using the forever block allows the Vision Sensor to take multiple snapshots so that it can continuously check for different objects.
- The blocks inside should only repeat a certain number of times
- The if-then-else block needed to be contained inside another looping block

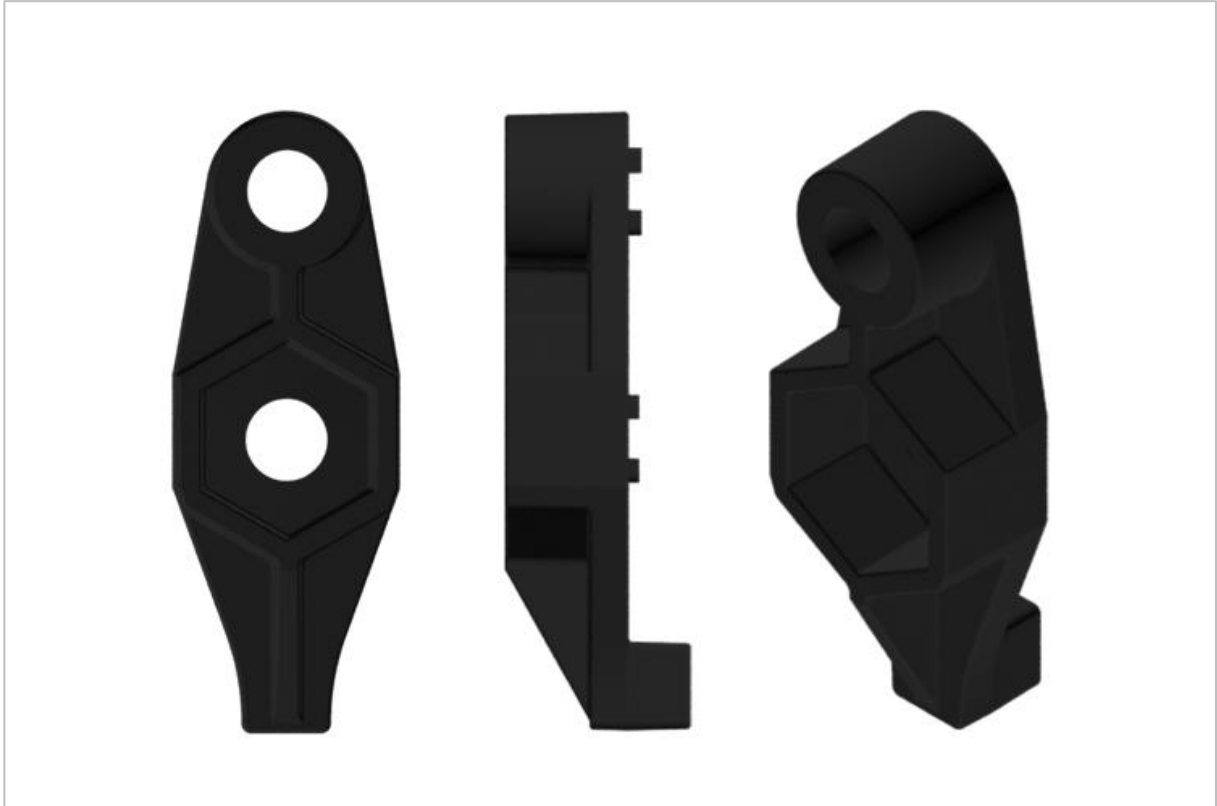
5. The leftmost X value of an object is 30 pixels and the width is 40 pixels in total. What is true about this object?

- Its centerX is 50 and it is to the left of the robot's center point.
- Its centerX is 70 and it is to the left of the robot's center point.
- Its centerX is 50 and it is to the right of the robot's center point.
- Its centerX is 70 and it is to the right of the robot's center point.

APPENDIX

Additional information, resources, and materials.

Using the 1 Post Hex Nut Retainer w/ Bearing Flat



1 Post Hex Nut Retainer w/ Bearing Flat

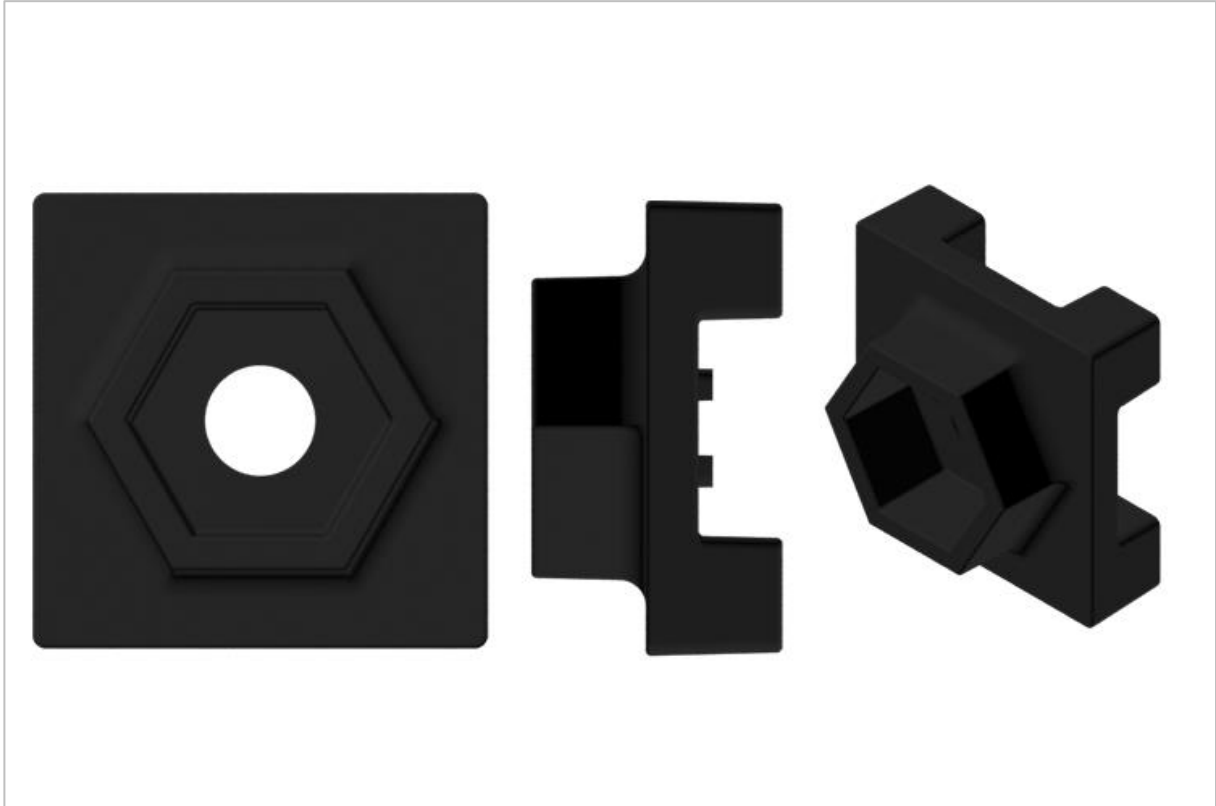
Using the 1 Post Hex Nut Retainer w/ Bearing Flat

The 1 Post Hex Nut Retainer w/ Bearing Flat allows shafts to spin smoothly through holes in structural components. When mounted, it provides two points of contact on structural components for stability. One end of the retainer contains a post sized to securely fit in the square hole of a structural component. The center hole of the retainer is sized and slotted to securely fit a hex nut, allowing a 8-32 screw to easily be tightened without the need for a wrench or pliers. The hole on the end of the Retainer is intended for shafts or screws to pass through.

To make use of the retainer:

- Align it on a VEX structural component such that the end hole is in the desired location, and the center and end sections are also backed by the structural component.
- Insert the square post extruding from the retainer into the structural component to help keep it in place.
- Insert a hex nut into the center section of the retainer so that it is flush with the rest of the component.
- Align any additional structural components to the back of the main structural component, if applicable.
- Use an 8-32 screw of appropriate length to secure the structural component(s) to the retainer through the center hole and hex nut.

Using the 4 Post Hex Nut Retainer



4 Post Hex Nut Retainer

Using the 4 Post Hex Nut Retainer

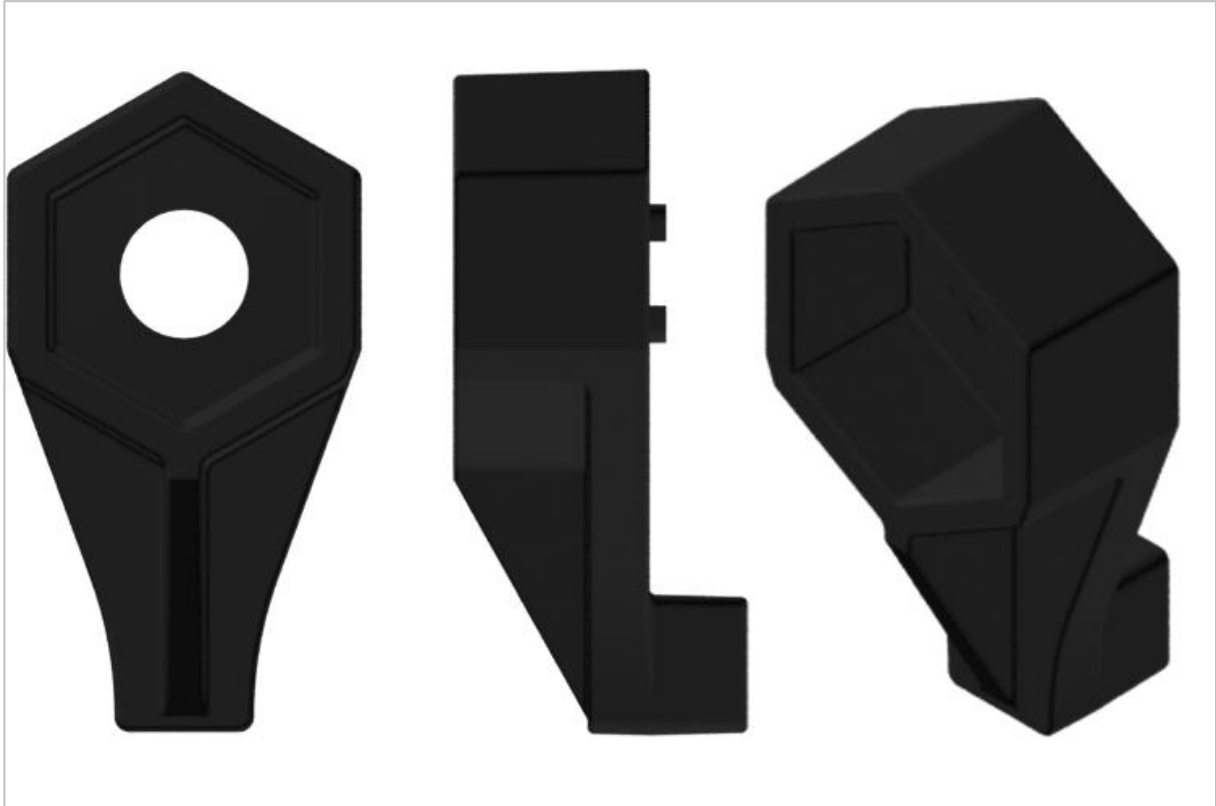
The 4 Post Hex Nut Retainer provides five points of contact for creating a strong connection between two structural components using one screw and nut. Each corner of the retainer contains a post sized to securely fit in a square hole within a structural component. The center of the retainer is sized and slotted to securely fit a hex nut, allowing a 8-32 screw to easily be tightened without the need for a wrench or pliers.

To make use of the retainer:

- Align it on a VEX structural component such that the center hole is in the desired location, and each corner is also backed by the structural component.

- Insert the square posts extruding from the retainer into the structural component to help keep it in place.
- Insert a hex nut into the center section of the retainer so that it is flush with the rest of the component.
- Align any additional structural components to the back of the main structural component, if applicable.
- Use an 8-32 screw of appropriate length to secure the structural component(s) to the retainer through the center hole and hex nut.

Using the 1 Post Hex Nut Retainer



1 Post Hex Nut Retainer

Using the 1 Post Hex Nut Retainer

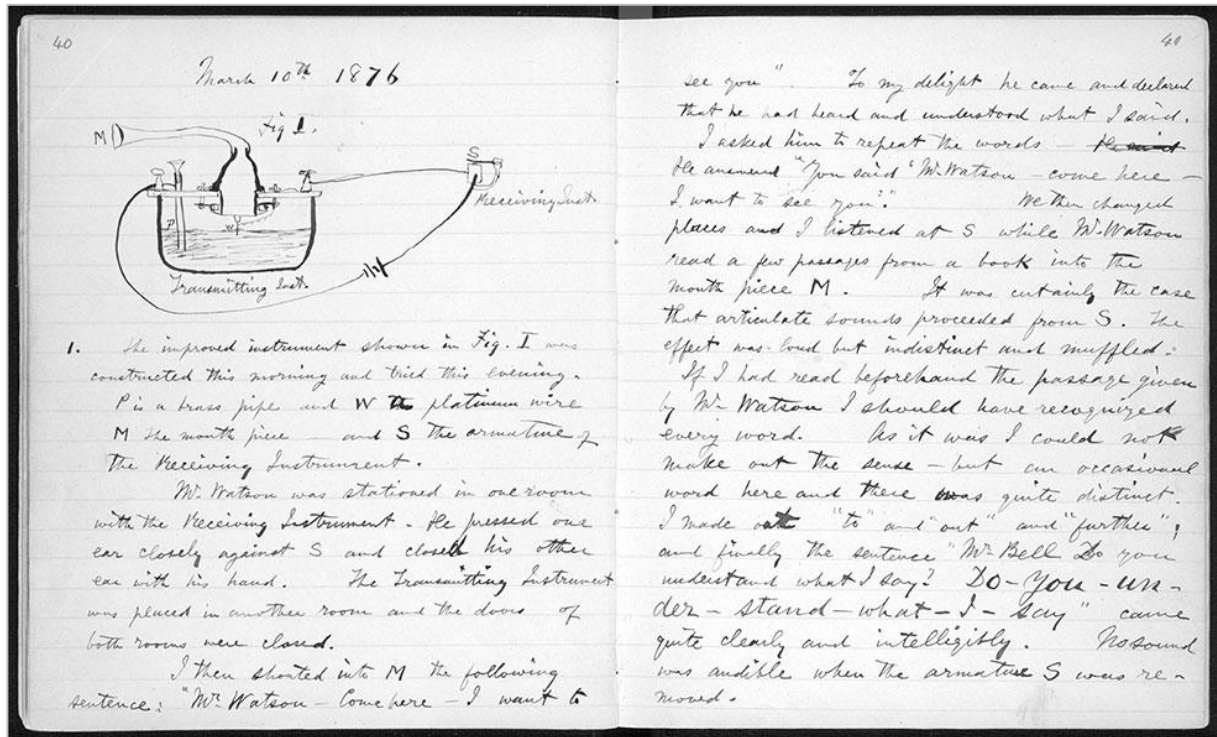
The 1 Post Hex Nut Retainer provides two points of contact for connecting a structural component to another piece using one screw and nut. One end of the retainer contains a post sized to securely fit in the square hole of a structural component. The other end of the retainer is sized and slotted to securely fit a hex nut, allowing a 8-32 screw to easily be tightened without the need for a wrench or pliers.

To make use of the retainer:

- Align it on a VEX structural component such that both ends are backed by the structural component and positioned to secure the second piece.

- Insert the square post extruding from the retainer into the structural component to help keep it in place.
- If the retainer is being used to secure two structural components, insert a hex nut into the other end of the retainer so that it is flush with the rest of the component. If used to secure a different type of component, such as a standoff, it may be appropriate to insert the screw through this side.
- Align any additional components to the back of the main structural component, if applicable.
- If the retainer is being used to connect two structural components, use an 8-32 screw of appropriate length to secure the structural components through the hole and hex nut. If used to connect a different type of component, such as a standoff, secure it directly or with a hex nut.

Engineering Notebooks



Alexander Graham Bell's notebook entry from a successful experiment with his first telephone

An Engineering Notebook Documents your Work

Not only do you use an engineering notebook to organize and document your work, it is also a place to reflect on activities and projects. When working in a team, each team member will maintain their own journal to help with collaboration.

Your engineering notebook should have the following:

- An entry for each day or session that you worked on the solution
- Entries that are chronological, with each entry dated
- Clear, neat, and concise writing and organization
- Labels so that a reader understands all of your notes and how they fit into your iterative design process

An entry might include:

- Brainstorming ideas

- Sketches or pictures of prototypes
- Pseudocode and flowcharts for planning
- Any worked calculations or algorithms used
- Answers to guiding questions
- Notes about observations and/or conducted tests
- Notes about and reflections on your different iterations

Building the Autopilot Robot with a Team

Teacher Toolbox

The build instructions will show students step-by-step instructions on how to build the Autopilot Robot. The Build Instruction Tips section will point out additional information for specific steps which will help students be successful with their build, so be sure to point out that section to students. There is an optional rubric to evaluate the robot build on this page. If any rubrics are used to evaluate students, review the rubric or pass out copies before students begin working so they are clear on how they will be assessed.

Before starting the build, consider how your students will be organized. Will each student have their own robot, or will they work in pairs or teams? If working in teams, each student could build a portion of steps or each student could be given a role. The following roles can be utilized during the building of Autopilot:

- **Right wheel** — This person follows steps 1-6 to build the right wheel of Autopilot. This person is also responsible for making sure that the motor gets plugged into the correct port (port 6).
- **Left wheel** — This person follows steps 7-12 to build the left wheel of Autopilot. This person is also responsible for making sure that the motor gets plugged into the correct port (port 1).
- **Sensors** — This person follows steps 13-26 to build the frame and attach the sensors.
- **Robot Brain** — This person follows steps 27-30 to connect all of the components including the Robot Brain and making sure the sensors are attached to the correct ports. This person is also responsible for making sure that the battery is charged and ready.
 - Port 2: Distance Sensor
 - Port 3: Color Sensor
 - Port 4: Gyro Sensor
 - Port 5: Touch LED
 - Port 8: Bumper Switch
 - Port 9: Bumper Switch

If there are two students in each group, the students can each choose two roles. If there are three students in a group, one of the students can choose to do two roles. If there are four

students in a group, each student can have one role.

Provide the list of roles and their responsibilities to the students. Once students are in their groups, allow the members to choose their role. Circulate the classroom and make sure that every student has a role. There is an optional collaboration rubric on this page.

Remind the students of roles throughout the exploration. For roles to work, students have to feel as though they will be held accountable for fulfilling those roles. Therefore, interject if you see a student taking over someone else's role or not fulfilling their assigned role. Reminders about who is supposed to be doing what can be useful interventions.