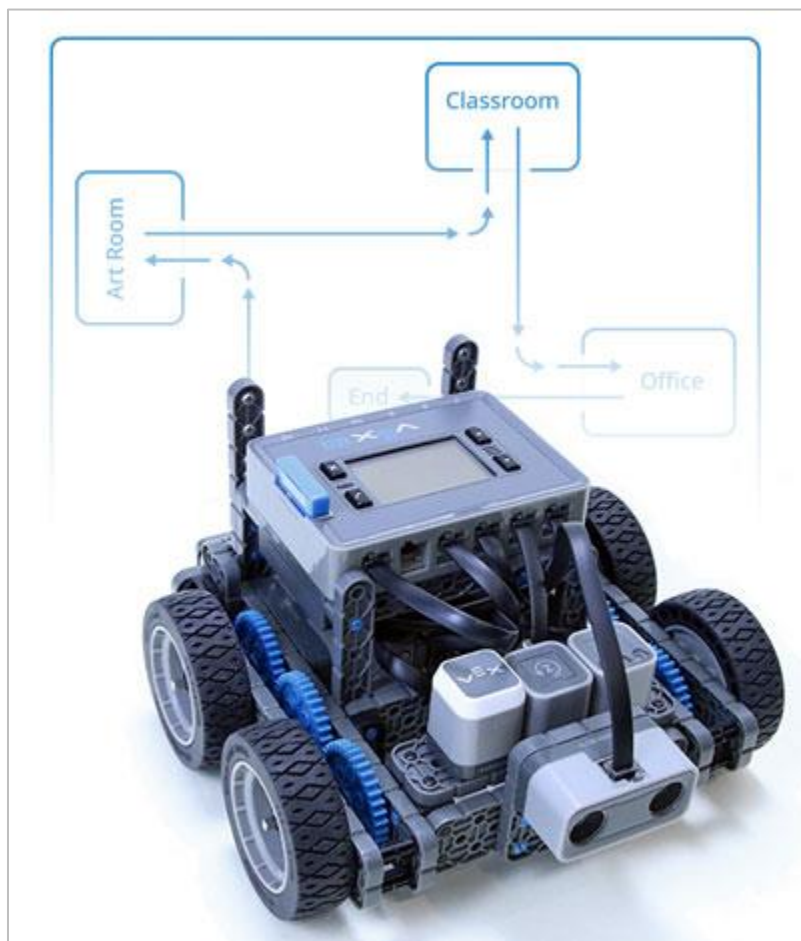




Movement Challenge



Program the Autopilot to drive on a designated path through a sequence of movements.



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

The Completed Look of the Build



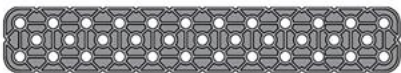
The completed Autopilot robot build.


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.


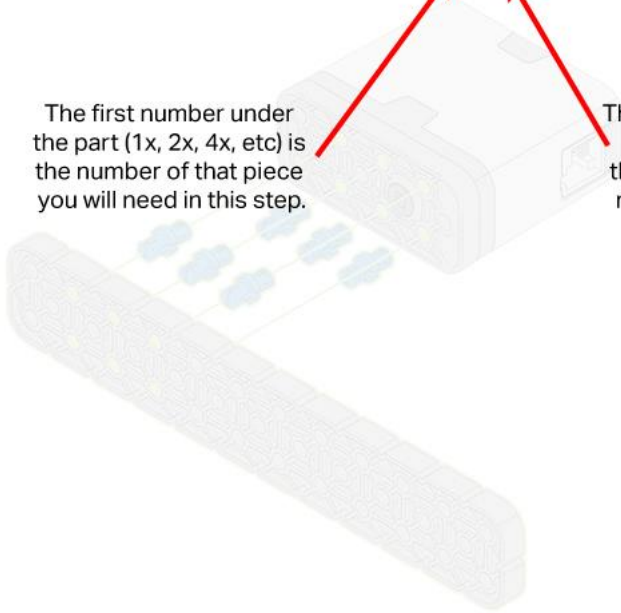
 **1x** - 2x12 Beam

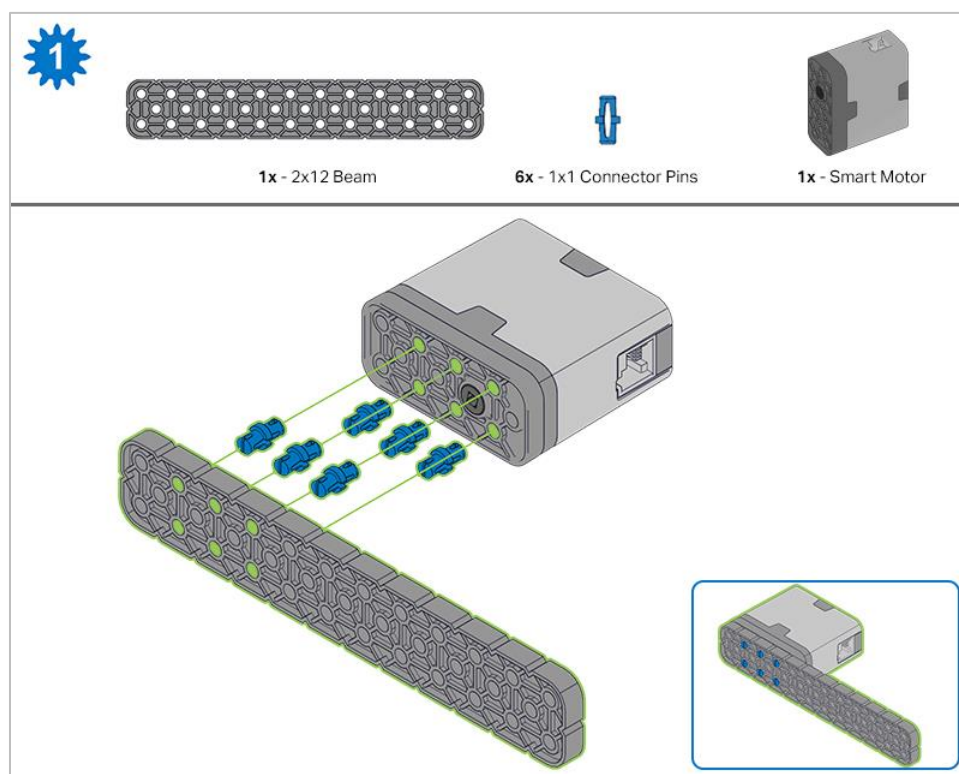
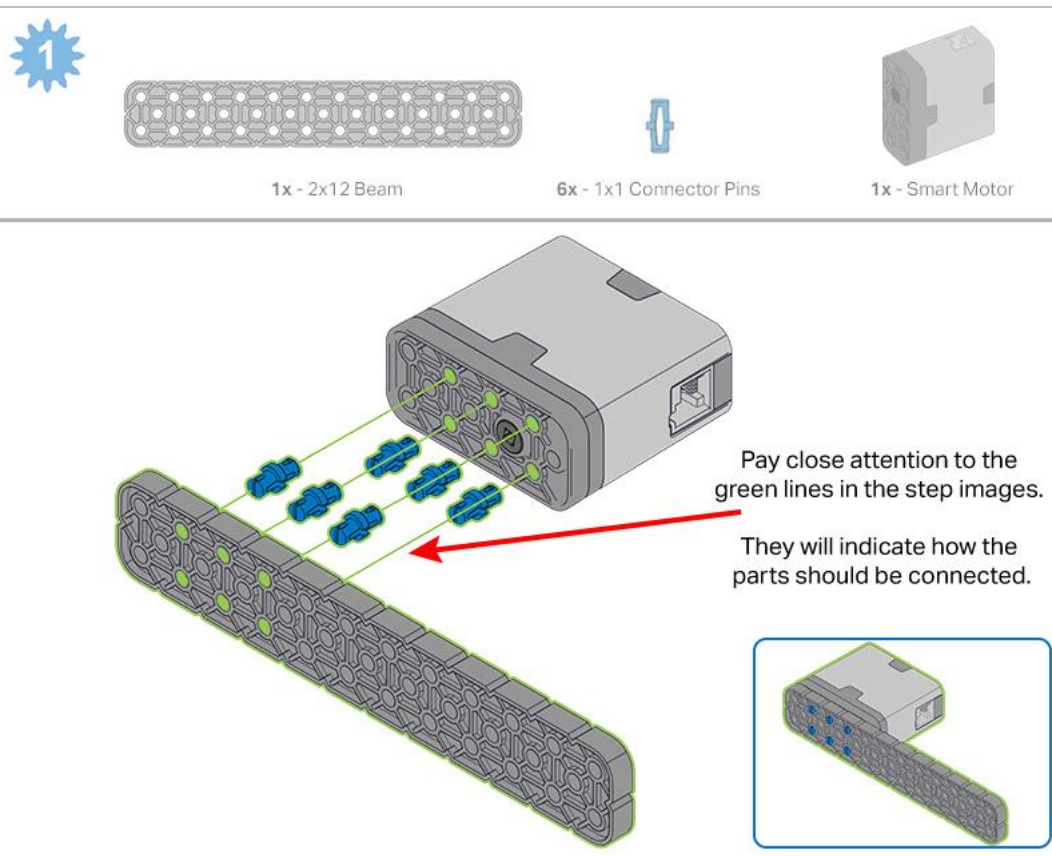
 **6x** - **1x1** Connector Pins

 **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



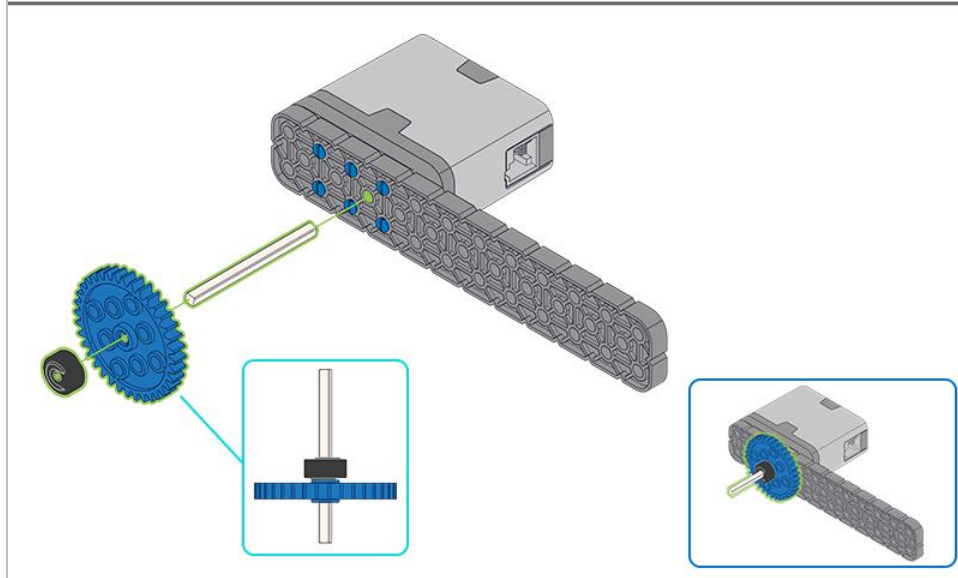
1x - 36 Tooth Gear



1x - Rubber Shaft Collar



1x - 4x Pitch Shaft

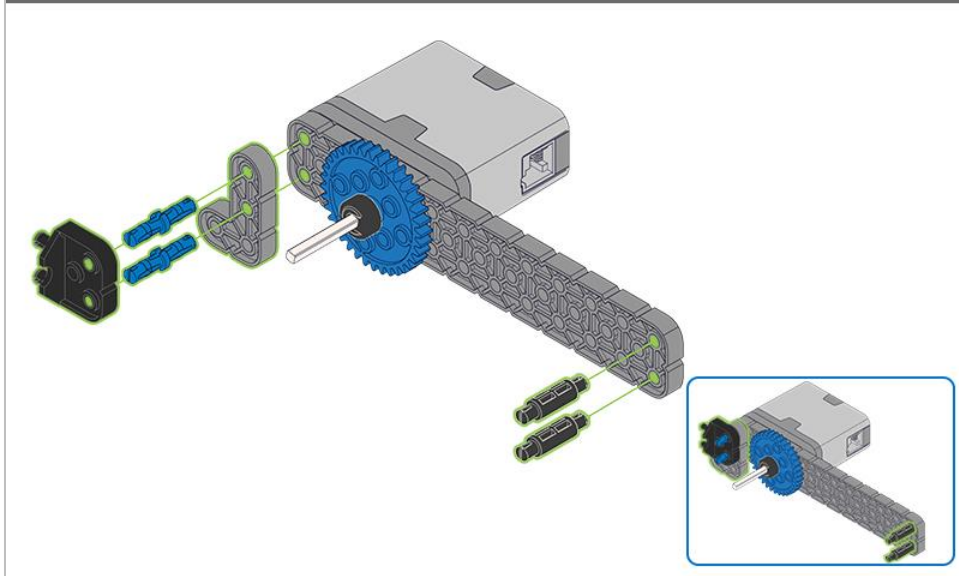


When adding the 4x Pitch Shaft, twist the pitch shaft to check for tension while turning. If it spins freely, it is not properly inserted into the motor.

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

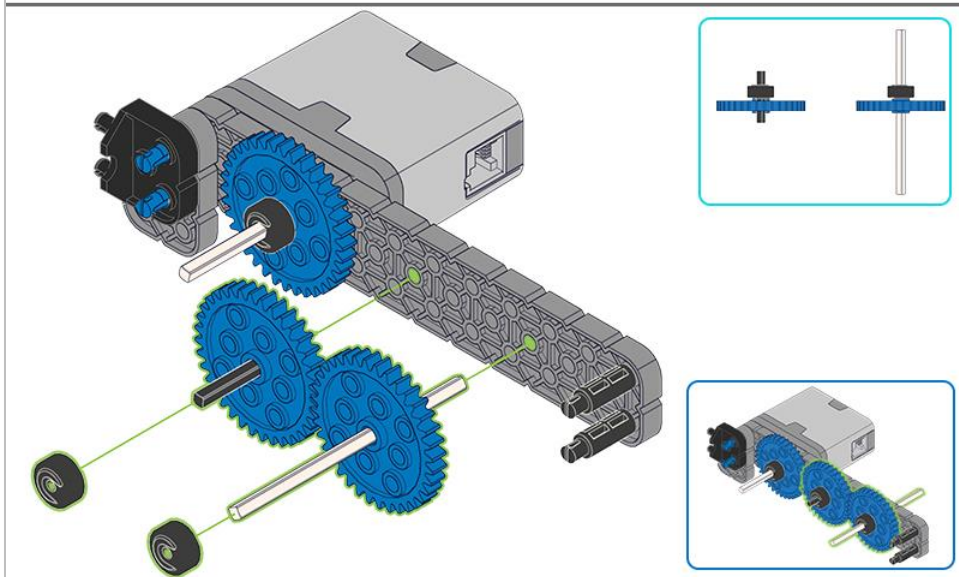


2x - Rubber Shaft Collars

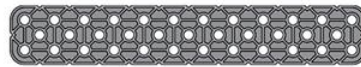
1x - 6x Pitch Shaft



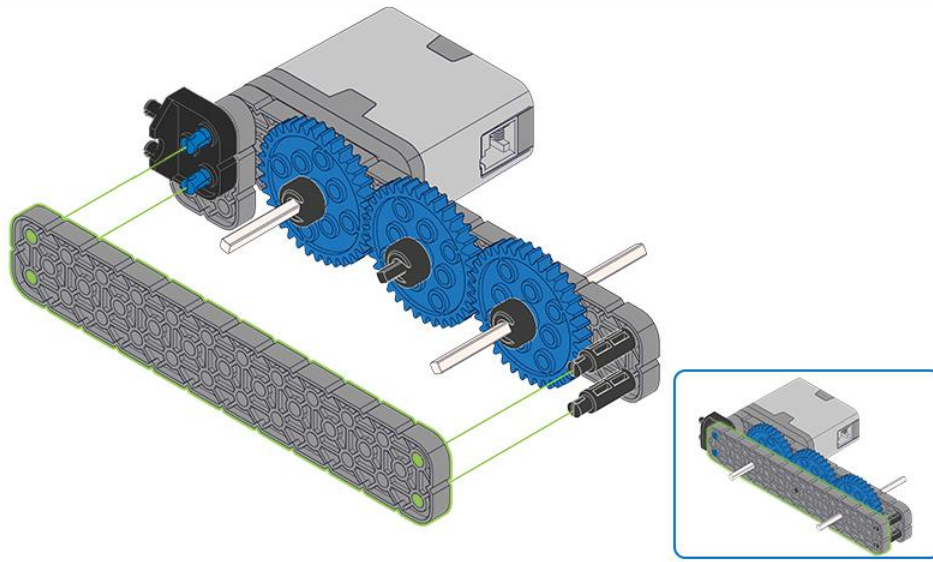
1x - 2x Plastic Pitch Shaft



5



1x - 2x12 Beam



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

6



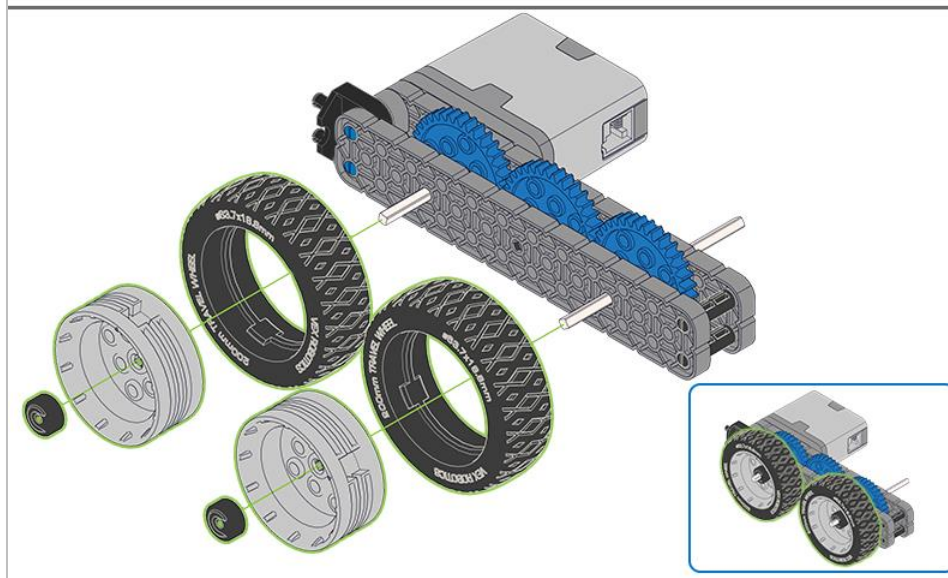
2x - Rubber Shaft Collars



2x - Wheel Hubs

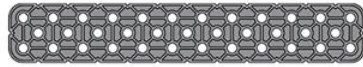


2x - Tires (200mm)



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.

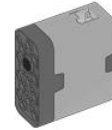
7



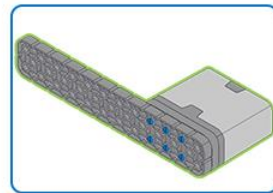
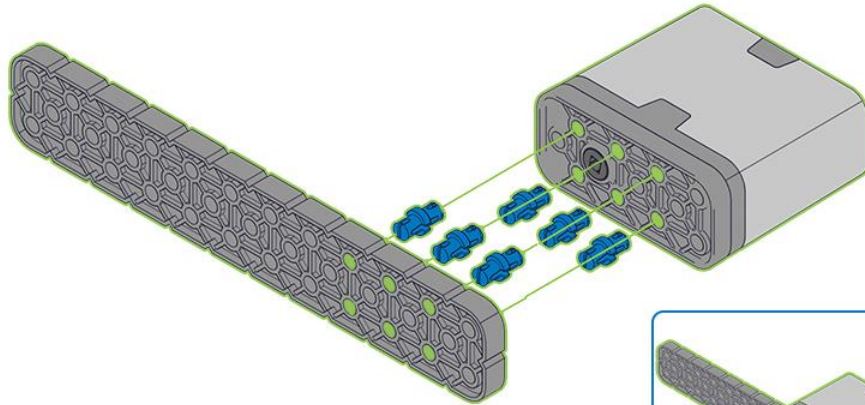
1x - 2x12 Beam



6x - 1x1 Connector Pins



1x - Smart Motor



8



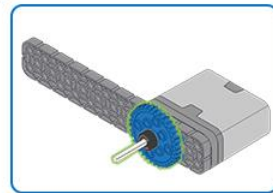
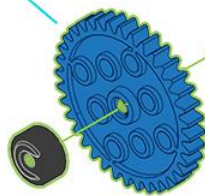
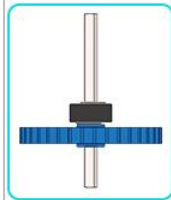
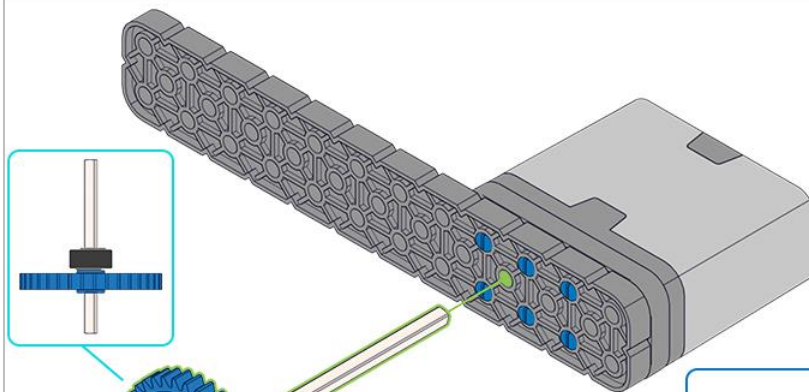
1x - 36 Tooth Gear



1x - Rubber Shaft Collar



1x - 4x Pitch Shaft

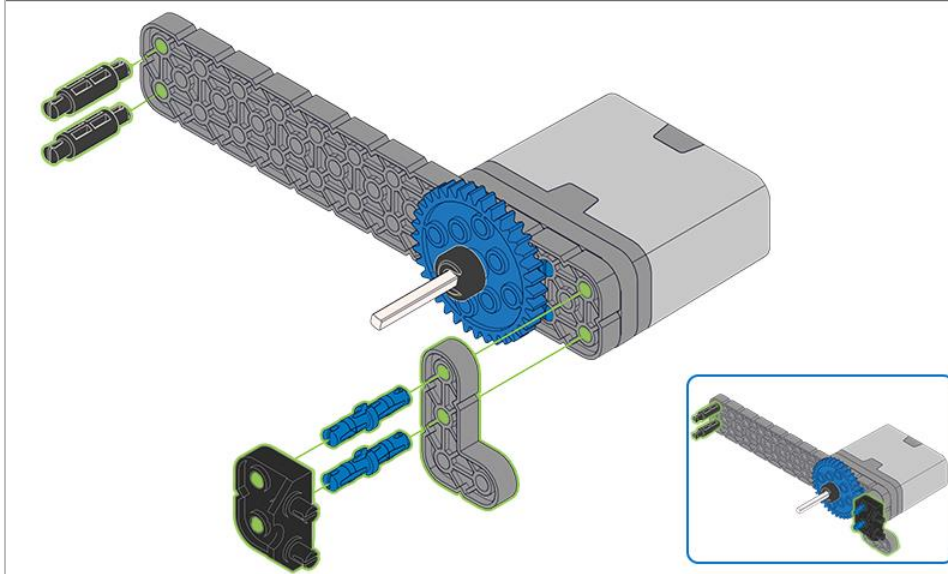


When adding the 4x Pitch Shaft, twist the pitch shaft to check for tension while turning. If it spins freely, it is not properly inserted into the motor.

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

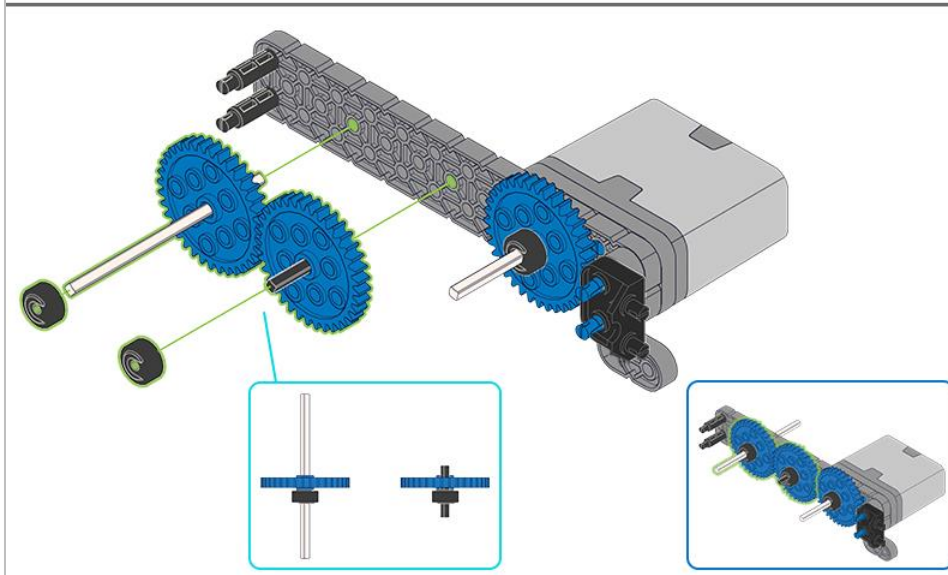


2x - Rubber Shaft Collars

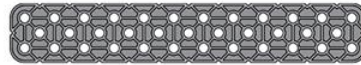
1x - 6x Pitch Shaft



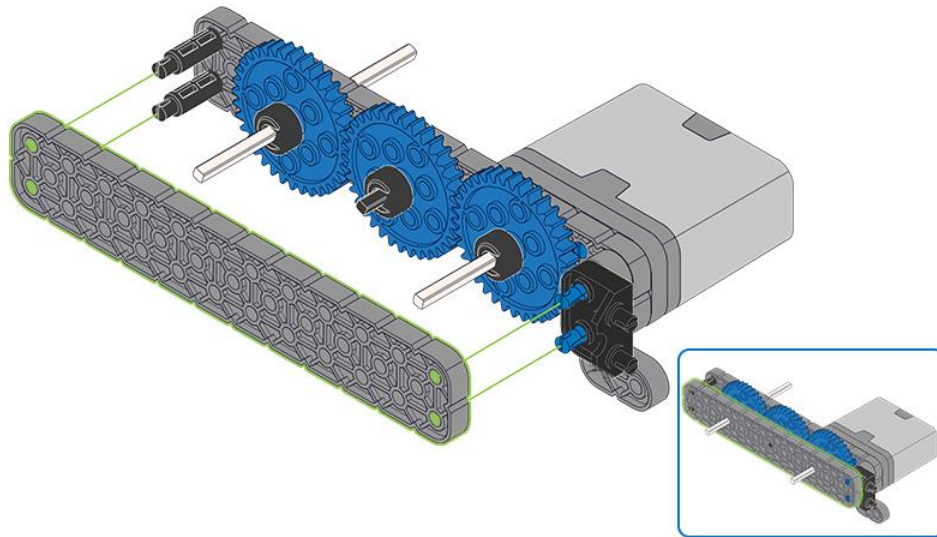
1x - 2x Plastic Pitch Shaft



11



1x - 2x12 Beam



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

12



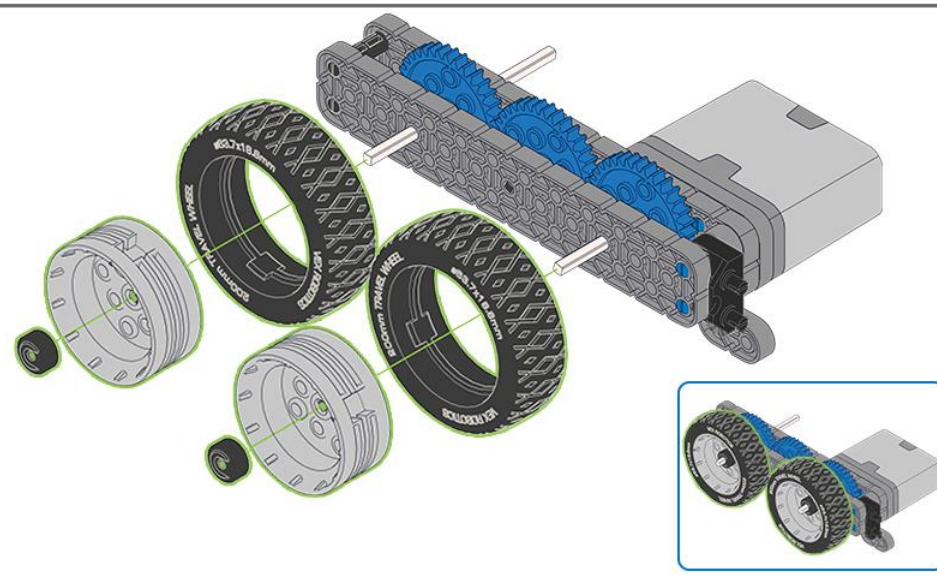
2x - Rubber Shaft Collars



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.

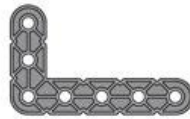
13



1x - 2x8 Beam



2x - 2x Wide, 1x2 Corner Connectors



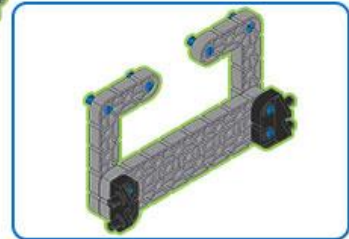
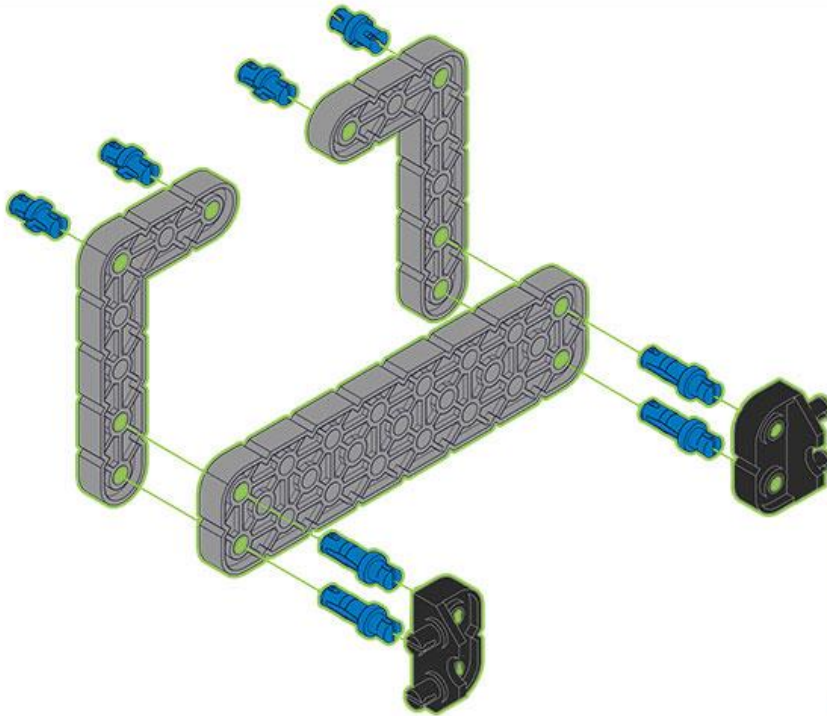
2x - 3x5 Right Angle Beams



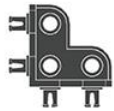
4x - 1x1 Connector Pins



4x - 1x2 Connector Pins



14

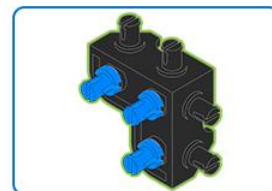
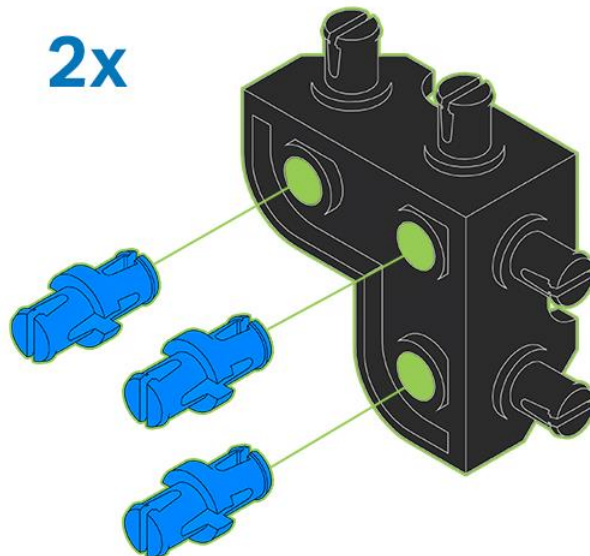


2x - Large Chassis Corner Connectors



6x - 1x1 Connector Pins

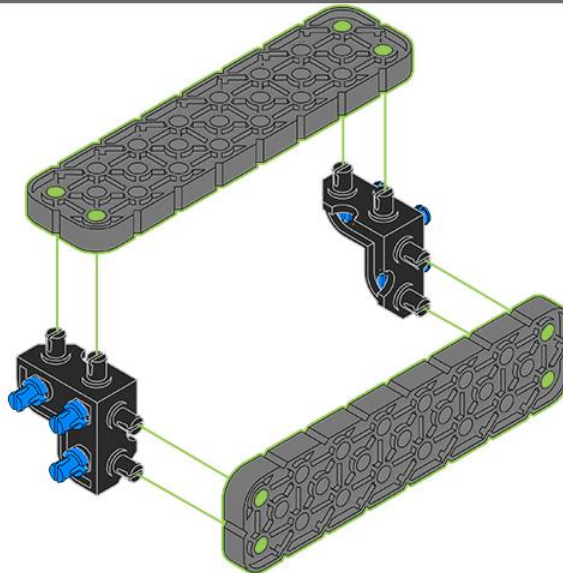
2x



15



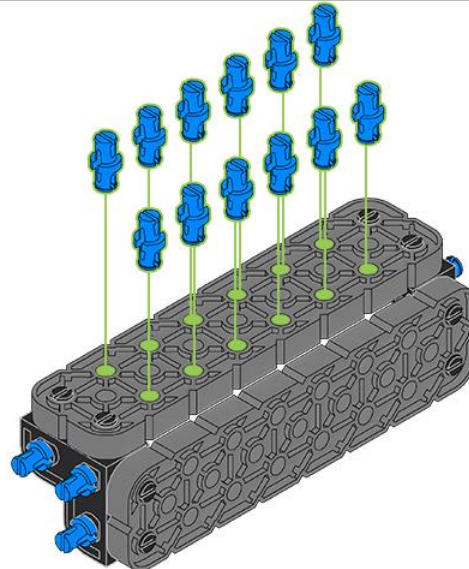
2x - 2x8 Beams



16



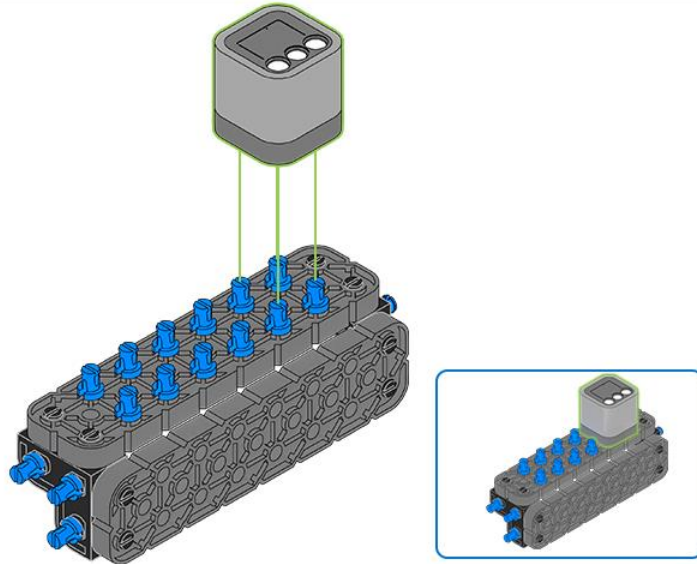
12x - 1x1 Connector Pins



17



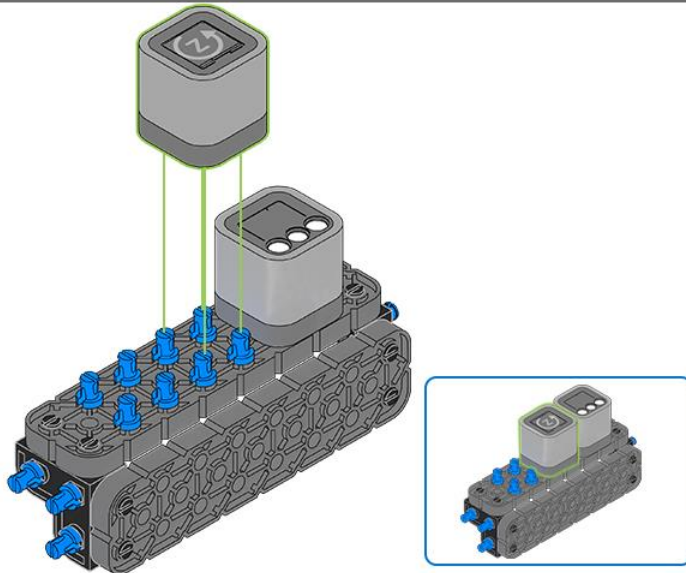
1x - Color Sensor



18



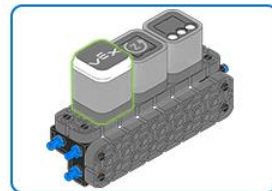
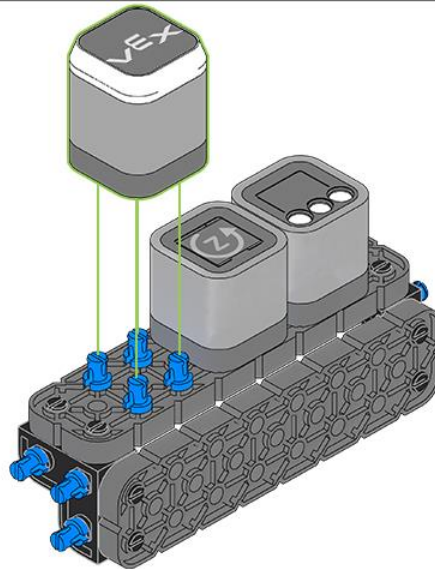
1x - Gyro Sensor



19



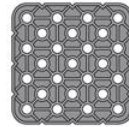
1x - Touch LED



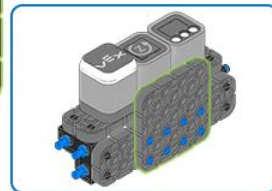
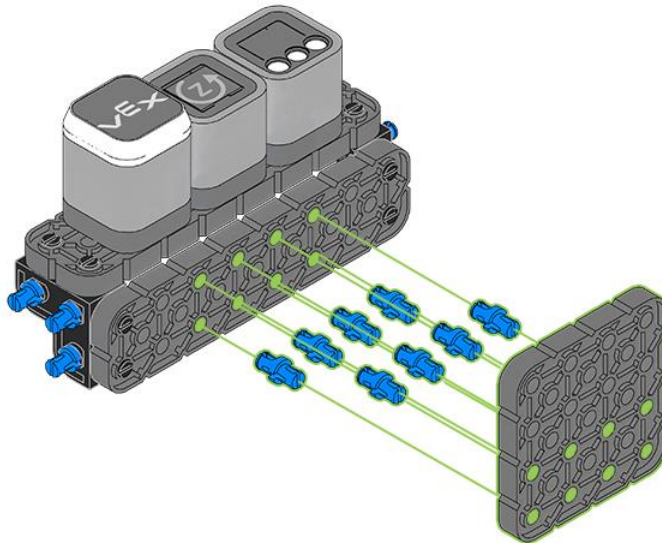
20



8x - 1x1 Connector Pins



1x - 4x4 Plate



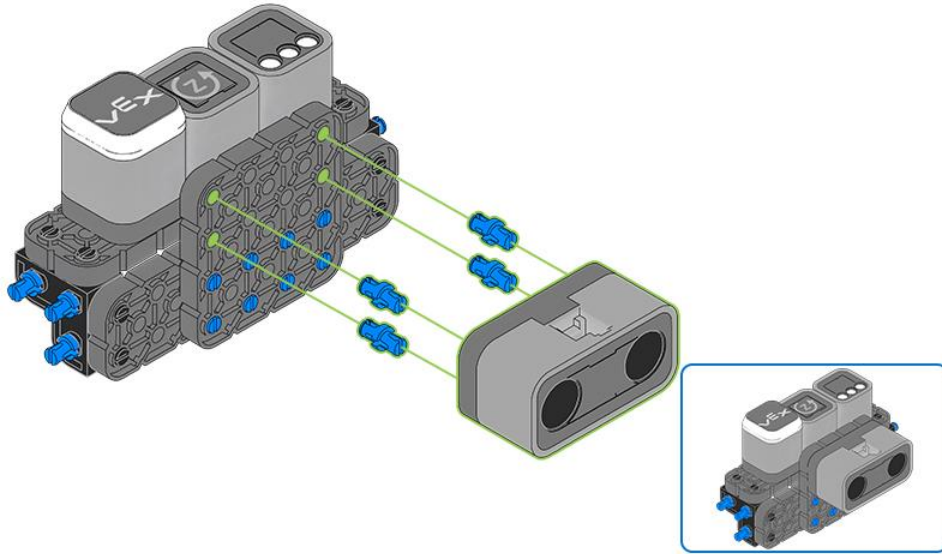
21



1x - Distance Sensor



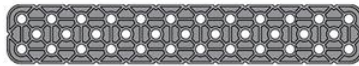
4x - 1x1 Connector Pins



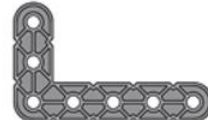
22



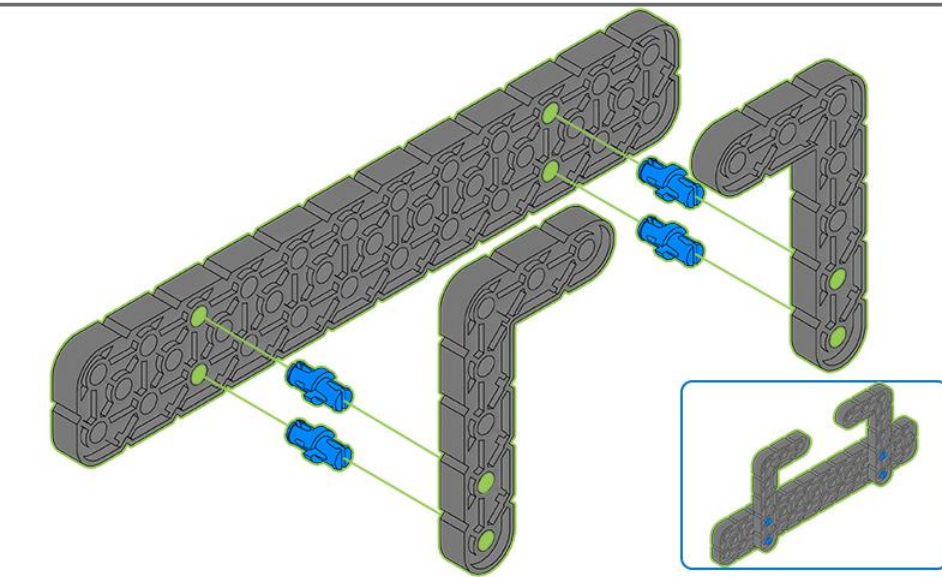
4x - 1x1 Connector Pins



1x - 2x12 Beam



2x - 3x5 Right Angle Beams



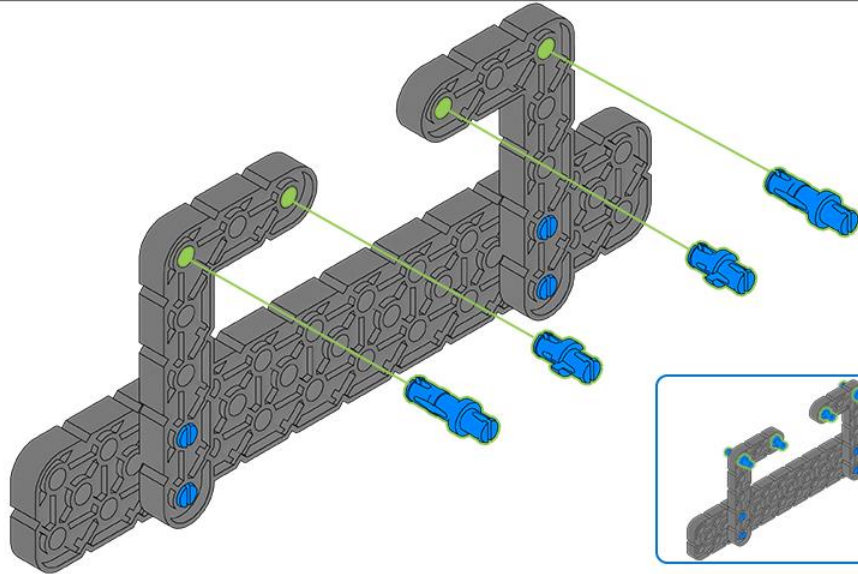
23



2x - 1x2 Connector Pins



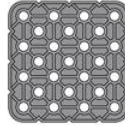
2x - 1x1 Connector Pins



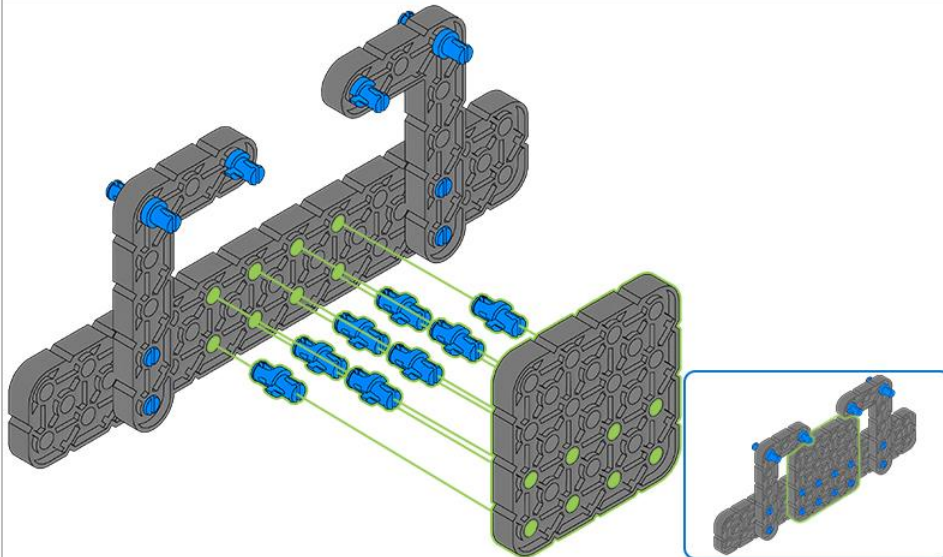
24



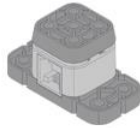
8x - 1x1 Connector Pins



1x - 4x4 Plate



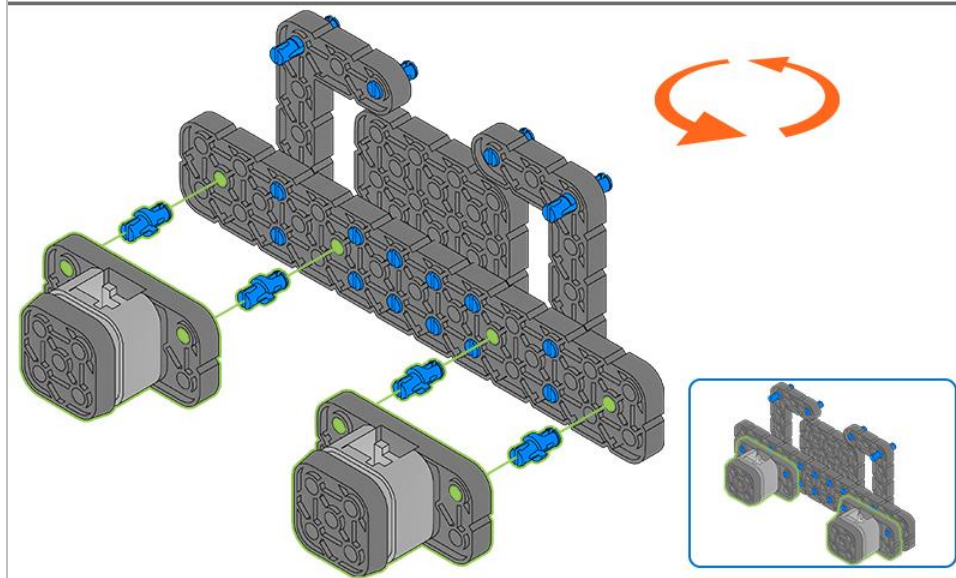
25



2x - Bumper Switch



4x - 1x1 Connector Pins



The orange arrows indicate to turn the assembly around.

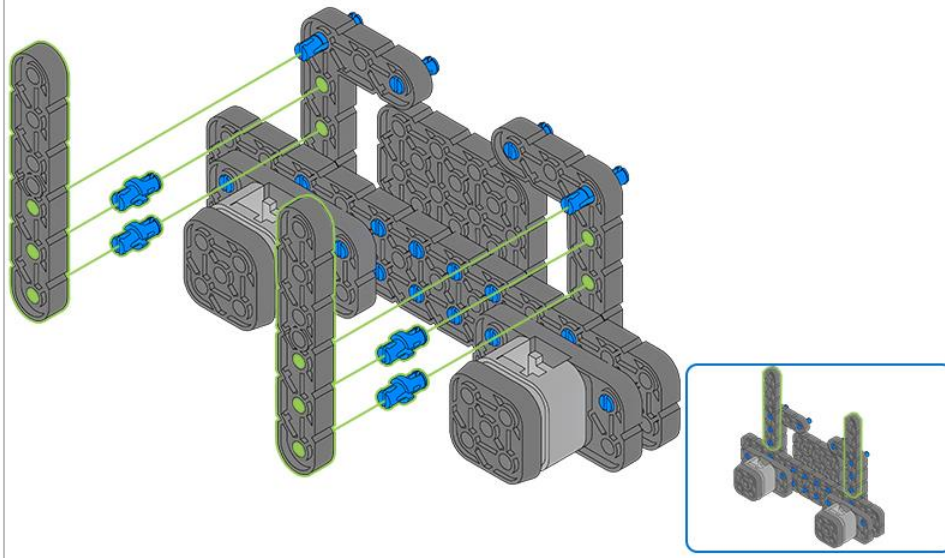
26



2x - 1x6 Beams



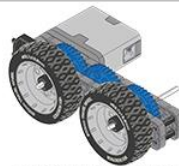
4x - 1x1 Connector Pins



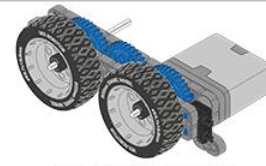
27



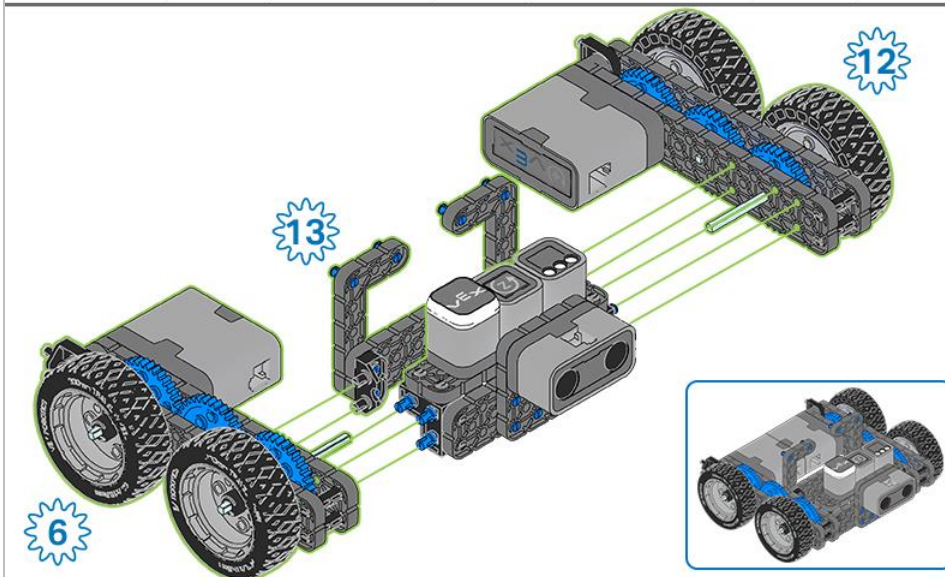
1x - Step 13 Assembly



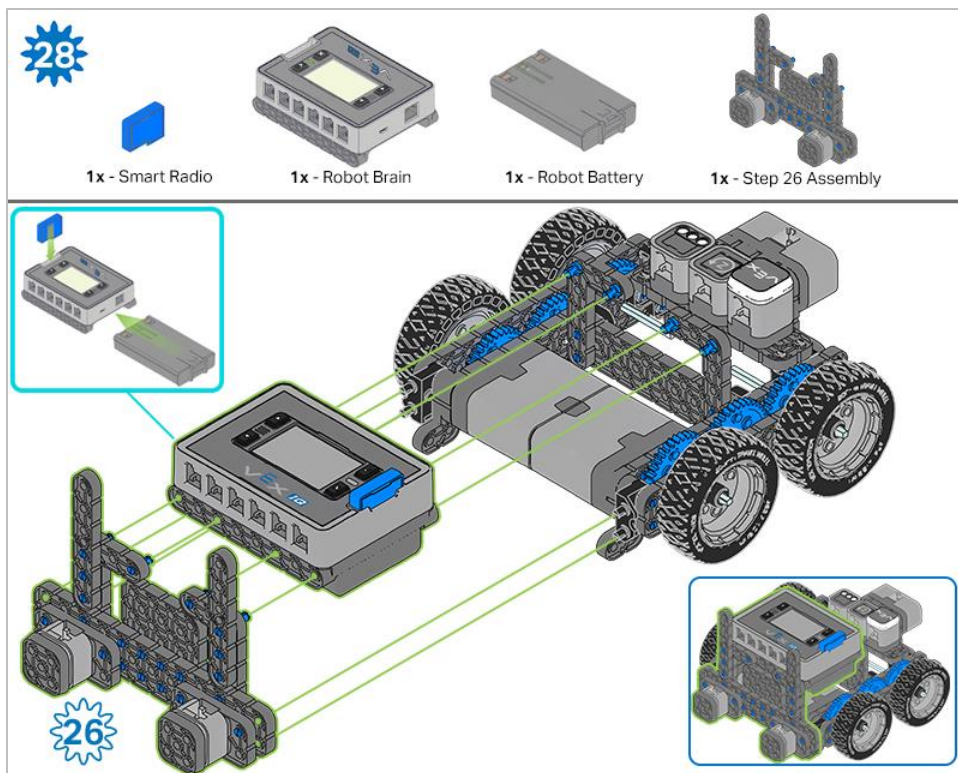
1x - Step 6 Assembly



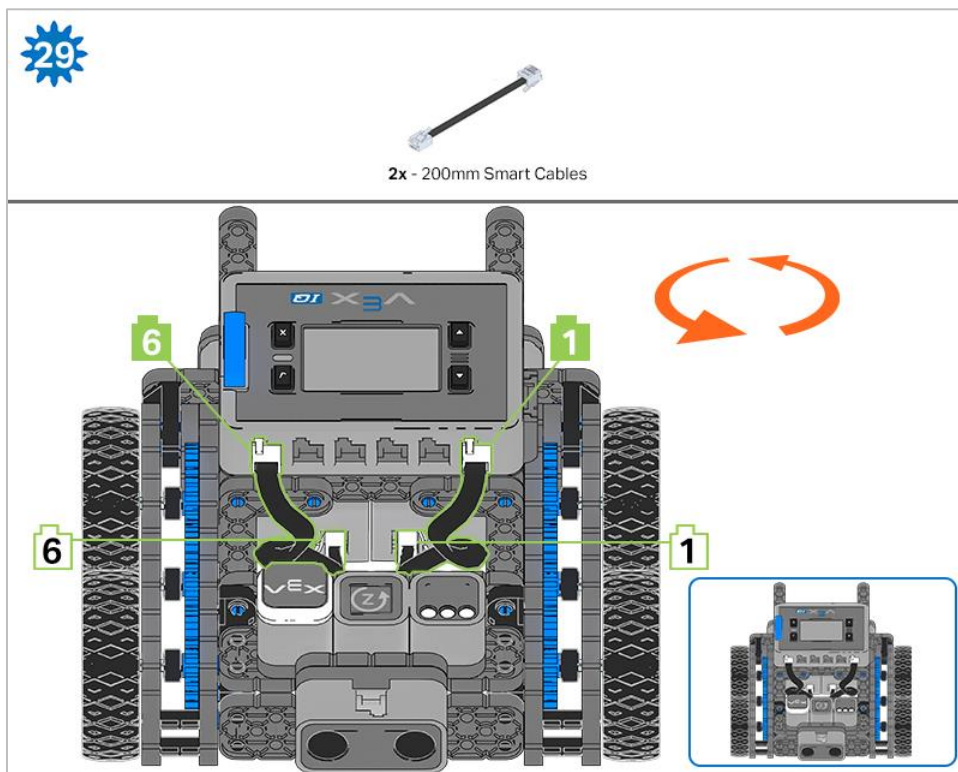
1x - Step 12 Assembly



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.



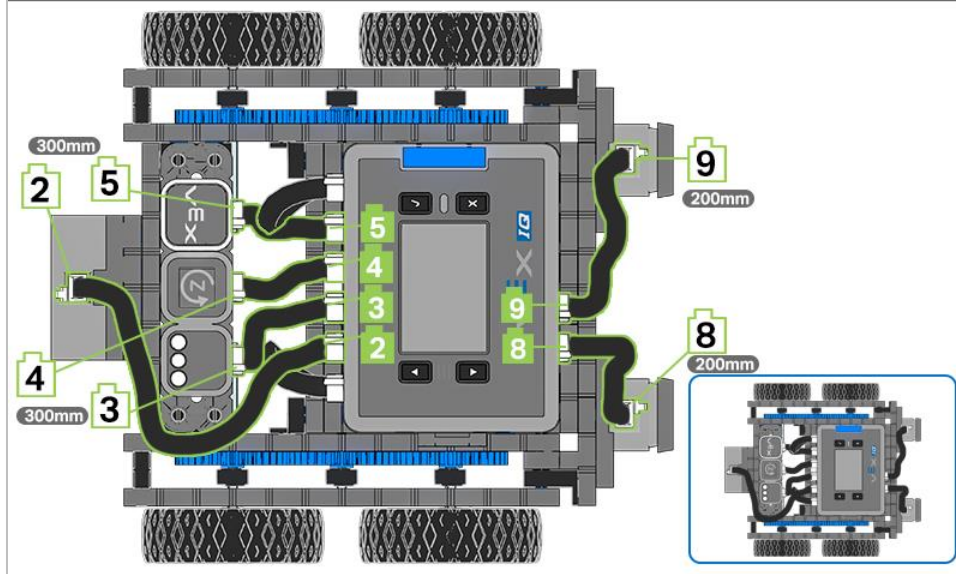
Steps 29-30, when you attach the Smart Cables, make sure they are tucked away so as to not block the Smart Sensors. The orange arrows indicate to turn the robot around.



4x - 300mm Smart Cables



2x - 200mm Smart Cables



Exploration

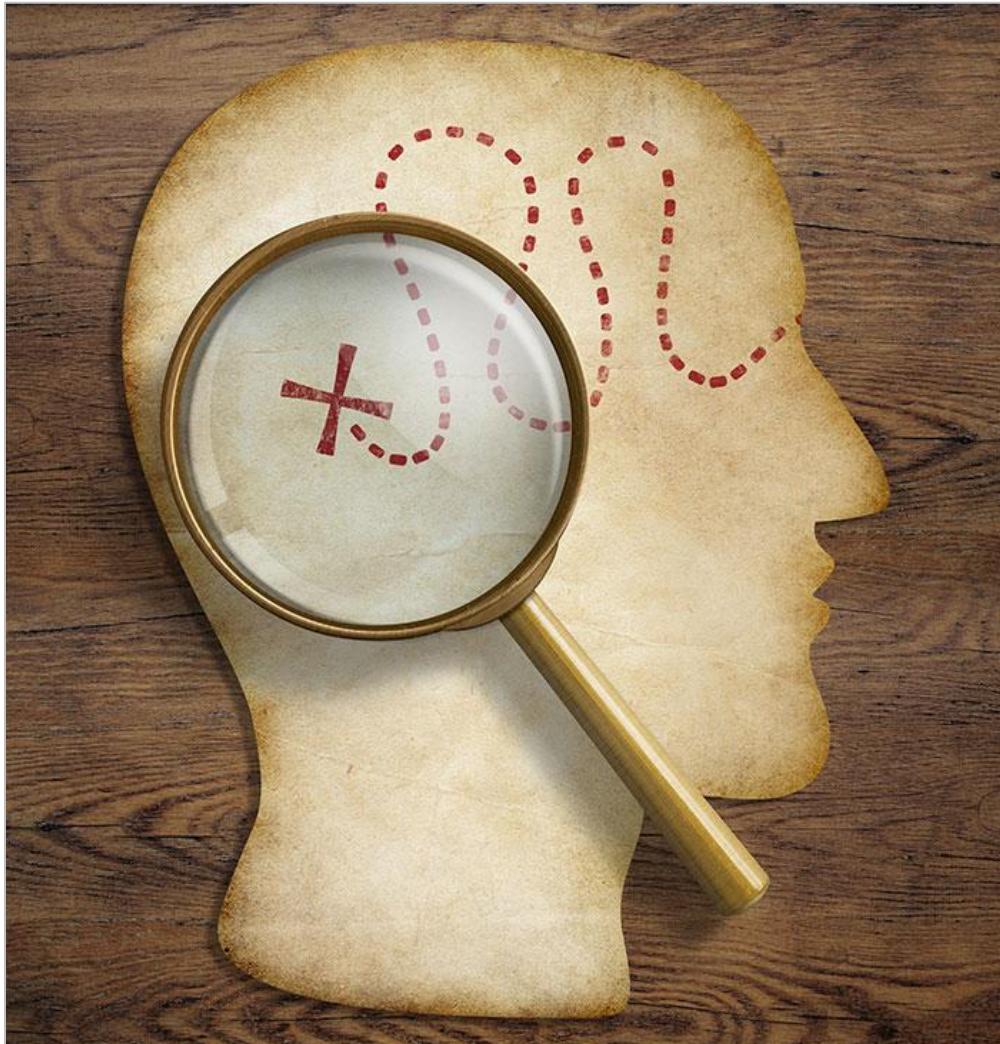
Now that the build is finished, explore and see what it can do. Then answer these questions in your engineering notebook.

- How would you use Autopilot in your daily life? What tasks could the robot accomplish?
- Think about the different sensors on the Autopilot and explain how these sensors could help solve a problem in your daily life. (Example: The distance sensor could help find walls in my apartment when it's vacuuming).
- Think back to constructing Autopilot, what advice would you give to someone that is starting the build?
- Think about a place you go in our school everyday. Write down 3-5 steps to get you to that place from here, as if you were explaining it to a friend.



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

Recycle Robot Run



Programming Specificity—Putting a design to work

Humans are able to read their environment using a combination of sensory information, decisions, inferences, and memories, all of which allow us to create mental maps which are largely qualitative (based on description, rather than measurement). However, robots can only do exactly what you tell them to do—and can only do it correctly—if they are given specific, quantitative (measurable) instructions. These instructions create robot behaviors: the ways that robots act, that range from basic to complex, depending on how the robot is built or programmed.

As expert direction followers, the more detailed your directions, the better the robot will be

able to accomplish your goal. In order to accomplish this, you, as the programmer, must design a plan which includes directions with measurement. These measurement-based directions create a blueprint for your robot to follow.

Imagine a Robot Helper Scenario...

Robots can help humans to do jobs more efficiently and easily. For instance, imagine a school where every afternoon someone has to come around to all of the classrooms to collect the recycling. While a student or a teacher could do this, that would take away from time they could be doing something else. We are going to design a plan and create a project so that a “Recycling Robot” could take on this task for us. The robot will travel to multiple classrooms, then take the recycling to a specific location, and come back to the beginning.



Where to begin? How the design process gets its start...

When architects are asked to build a new building, they don't pick up a hammer and start banging away. Before a tool is ever picked up by a contractor or construction worker, the architects spend a lot of time and energy on creating the plans for the building. They have to think about spaces and how they connect and relate to one another. This kind of thinking is called spatial reasoning.

First, architects need to know what the purpose of the building is, and how it is going to be used—the functionality. Then they think about the many, many different ways they could achieve that functionality, and what it could look like. Architects create sketches, lists, and design plans that they think might work. They work with others involved in the building, and those plans get revised, and eventually turn into blueprints—the specific, measured directions that will be used to build that particular building.

When you begin to think about a design plan for a robot, you will begin in much the same way. First, you have to know the functionality—what do you want the robot to do? In this case, we want the robot to help us collect recycling. It has to travel around the school, and pick things up, so you need to think about the spaces it is going to travel in and how those hallways and rooms connect to each other. Next, we can think about the many ways to map a path around your school—using sketches or lists to show your ideas helps others that you work with to see and better understand your thinking. Your group might take ideas from several plans and put them together into one. Then, when your group decides on a design plan, you can create the blueprint—the specific instructions that you want the robot to follow.

Open Ended STEM Lab

Exploration: Design

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	Autopilot Robot
1	Charged Robot Battery
1	VEXcode IQ Blocks
1	USB Cable (if using a computer)
1	Engineering Notebook
1	Large Paper for Mapping
1	Markers
1 per group	Block to use as obstacle (optional)

What will be in your project?

You will be creating a plan, or blueprint, for your Recycle Robot and then create a project to execute it. While your robot will not really be going around your school, creating a plan at a smaller scale still requires precision. In order to do this, each plan and project should include the following:

- Your classroom is “home base”, where it will leave and return to
- The robot must enter and exit 3 classrooms
- The project must include the four commands: drive forward, drive reverse, turn left, and turn right. (A successful project will include several of each of these commands.)

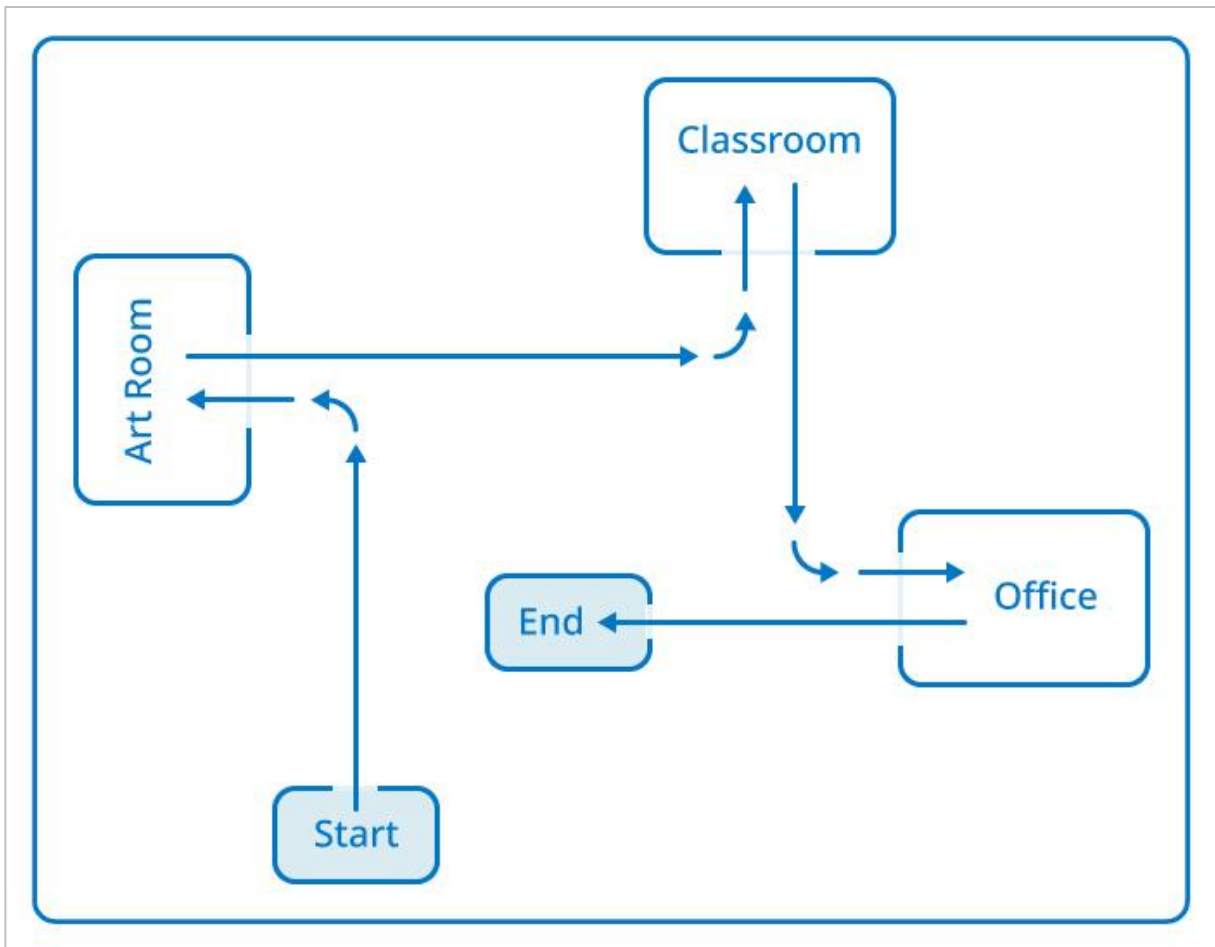
- An indicator that recycling has been picked up (i.e. Touch LED turns on; wait 3 seconds; etc.)

When your group is working on your design plan together, the Builder and Recorder should make sure that all of these requirements are being met by the path you design.

1. Map out a Solution—Spatial Thinking in Action

Now that you know the requirements and parameters, sketch out a map in your engineering notebook to show the three stops your robot will make to pick up recycling in the school.

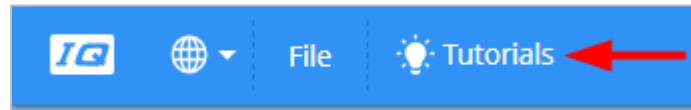
- Be sure to mark the start and end points, and label the classrooms or places that are important.
- Use arrows to show the direction and order that the robot will travel.
- Use the example sketch as a guide. Remember, this is a plan, so it doesn't need to be exact, but you may want to add notes or reminders for yourself to help when you begin coding.
- Check in with your teacher when you have a completed map in your notebook. The Recorder should have the final version to share with the teacher, and to base your large map off of.
- When your teacher approves your sketched map, use the large paper, markers, and rulers to transfer that map to a larger scale that the robot can drive on. Remember to measure the distances that you draw to match the parameters and units discussed. That way, your code will match your map, to help your robot drive successfully.



Open Ended STEM Lab

Exploration: Code

1. Preparing to Code



Tutorials and Help can found at the top of the VEXcode IQ Blocks interface

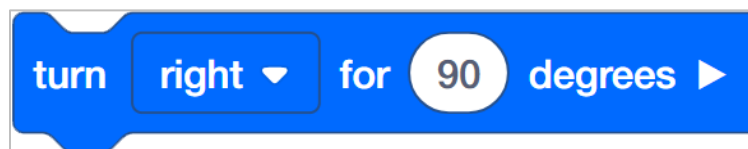
Remember, you can always look to the tutorials for more information about blocks, sequences, and troubleshooting. Use the Help section when you want to know about what a specific block does or how to use it.

Using Blocks with Specificity

To move a specific distance, use the `drive for` block, and change the direction and distance travelled by adjusting the parameters in the block.



To turn either right or left in any degree increment, use the `turn for` block. Again, you can adjust the parameters to suit your design needs.



Open a New Project

The Programmer should start a new project with the Autopilot (Drivetrain) template, name, and save it. Since we will be creating a path for a recycle robot, rename your project "Recycle Run."

For reminders on how to get started see the tutorials or refer to the example programs for reference.

2. Create a List of Steps Using the Comment Blocks

What Are Comments?

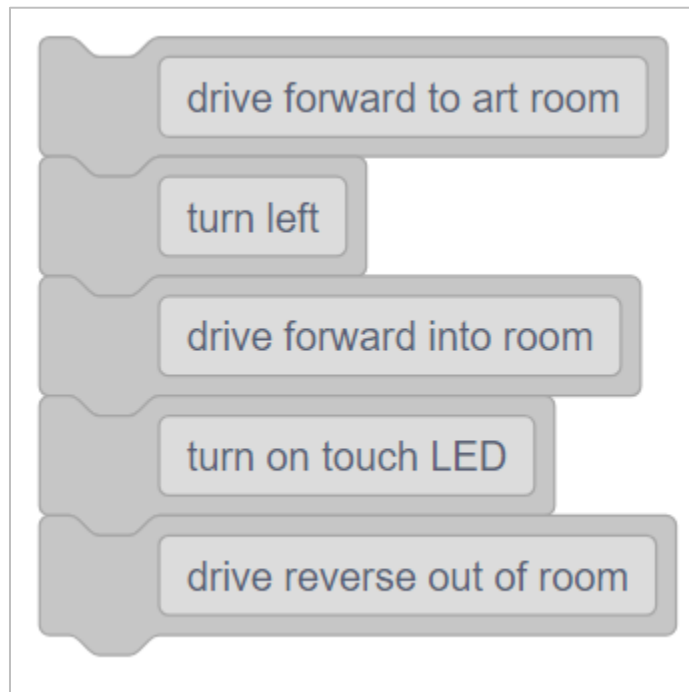
Comments are usually added to projects to explain what a programmer wanted parts of the project to do. So they are helpful when collaborating and troubleshooting.

In this case, we are going to use them to help us make a general list of the steps the robot needs to do to complete its task.

What is the First Thing the Robot is Going To Do?

In the project, use the `comment` blocks to create a basic sequence in your group. The Recorder or Programmer should list the first 10 steps in VEXcode IQ Blocks, with a `comment` block for each step. See the example here.

The text in the block does not have to be as specific as the blocks themselves, as this is just a guide to follow when you begin to create the actual coding sequence.



3. Create the Coding Sequence

Now, it's time to translate those steps into a block sequence.

- Your robot should begin the path facing forward, so drag and drop the `drive for` block and set the parameters to match the first step in your path.
- Look for the next step, and drop the `turn for` block and set the parameters to match the next step in your path.

When you get into a room, add an indicator block (like `set Touch LED color` or `play`

sound). Because the Autopilot cannot actually pick things up, the indicator represents the robot picking up the recycling before moving on.

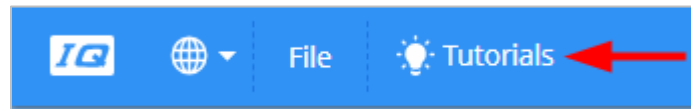
- Keep dragging and dropping blocks and setting parameters until you have finished the path for your Recycle Robot.

4. Preparing to Run Your Project

Before you run the project, 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?

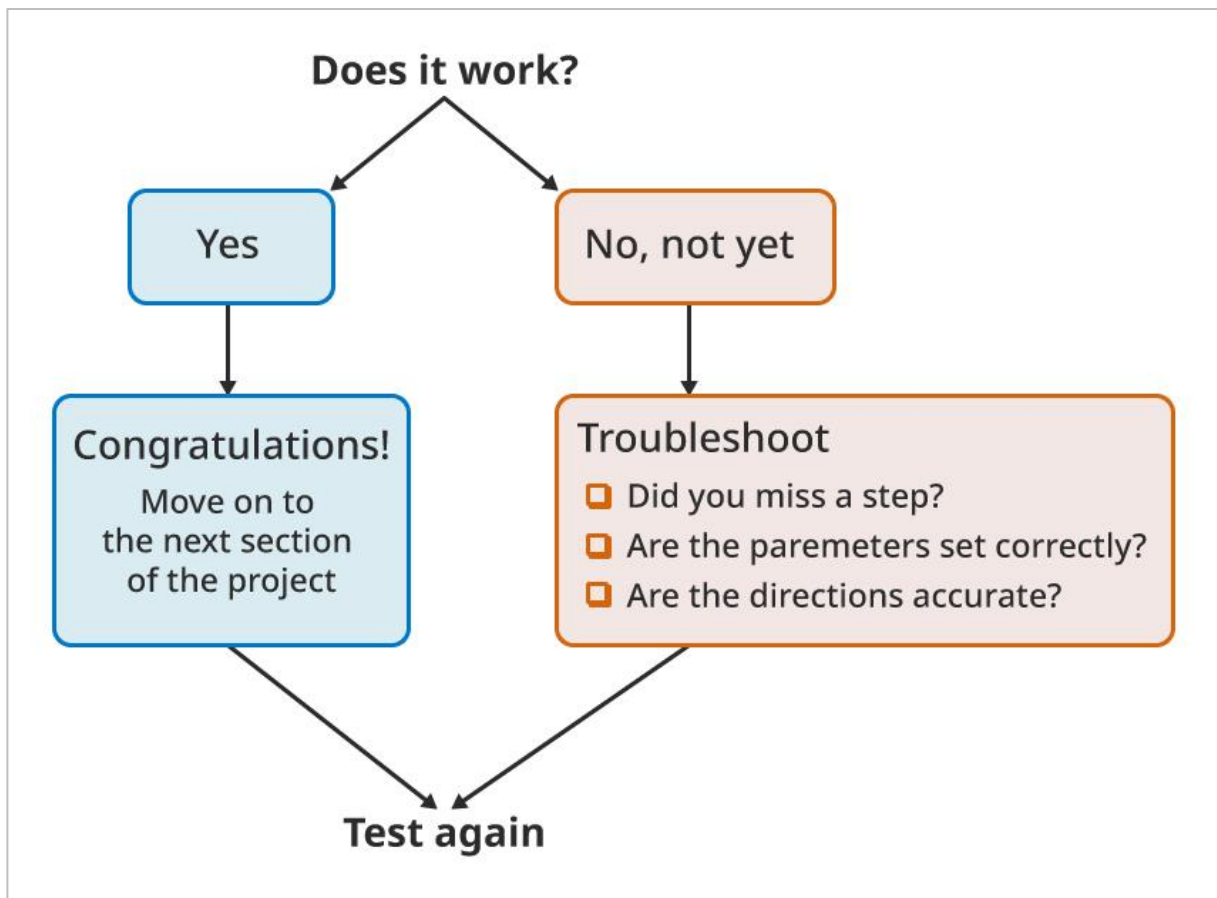
For a review of downloading and running your project, see the tutorials.



5. Test Run your project

Now it is time to test out your project! Make sure your Autopilot robot is set up with enough space to be able to run the complete program without falling or bumping into another robot or team.

- The Operator should now run the first part of the project on the Autopilot robot by making sure the project is highlighted and then pressing the Check button.
- How did your test run go? Did your robot drive the path you thought it would? Follow the troubleshooting chart, and continue to create and check sections of code until you have a completed path.



Open Ended STEM Lab

Exploration: Share

Reflection Questions

Congratulations on your project! Now that you have completed a design and coding experience, take a moment to think about the process you just engaged in. In your engineering notebooks, answer the following questions to help you reflect on your choices and your roles in the group:

- What did you consider when planning the path for the robot?
- What indicator did you choose and why?
- What part of the process was your or your group's biggest success?
- What part of the process was your or your group's biggest challenge?



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

Real World Connection



Italy's Trash Robot

Today's robots have the ability to do tasks that humans both do not want to do. In Italy's case, the politics of picking up trash landed cities with trash pile ups and ultimately a less cleaner area to live. The Trash Robot works by arriving a home, asking for a personal ID number that identifies both the user and then asking if the trash is organic, recyclable or waste. Using this robot both helps make trash clean up efficient but also tracks how much trash is being collected through Italian cities. The Trash Robot takes the trash in its belly to a dump site, where no human would want to travel to. Using technology like the Trash Robot, gives humans the ability to use robots to complete tasks that humans do not want to do and cannot do as efficiently.

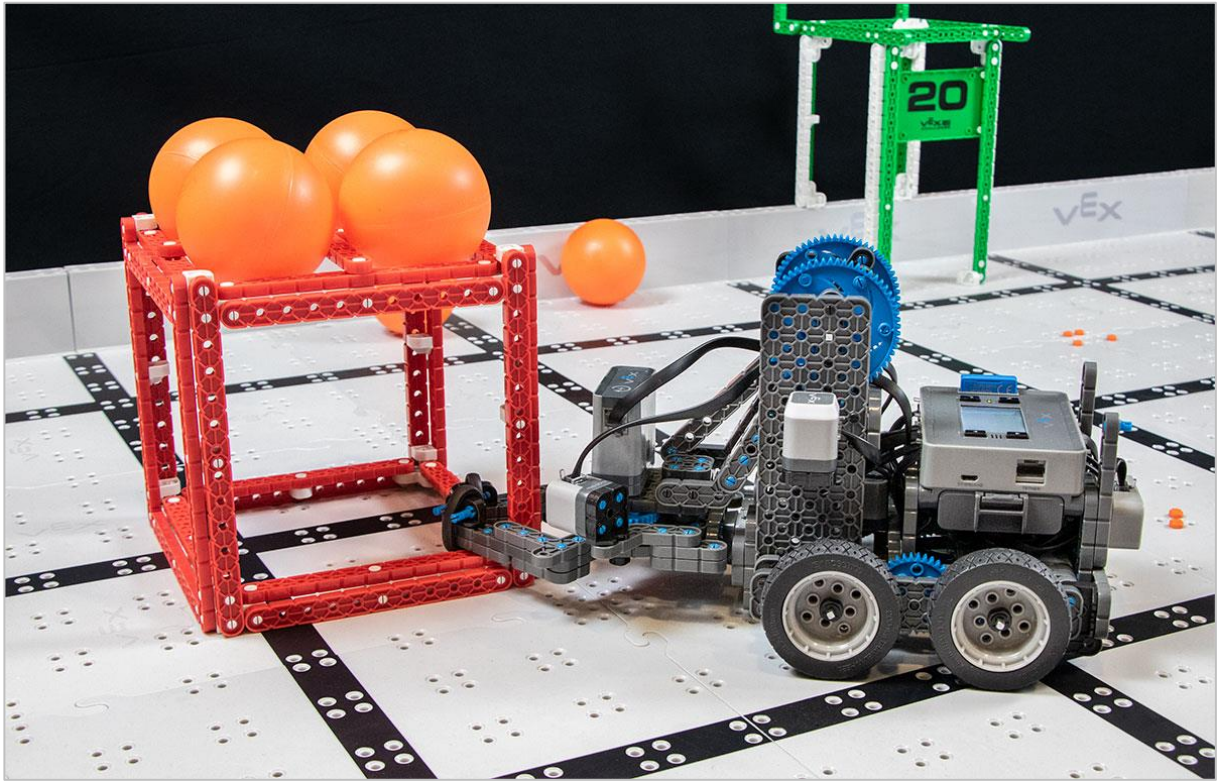
In the 21st century, engineers are continuing to develop robots to complete tasks that benefit humans in many ways. These advancements have led robots to become widely used in our communities.

There are many benefits for using robots:

- They are capable of lifting heavy loads without injury.
- They are more precise in their movements.
- They can produce more products in a shorter amount of time.

- They can work at a constant speed with no breaks.
- They can perform tasks that are dangerous to humans.
- They can work in hazardous conditions, such as poor lighting, toxic chemicals, or tight spaces.

Competition with a Purpose!



A VEX IQ robot playing the Squared Away Challenge

Speed and Control

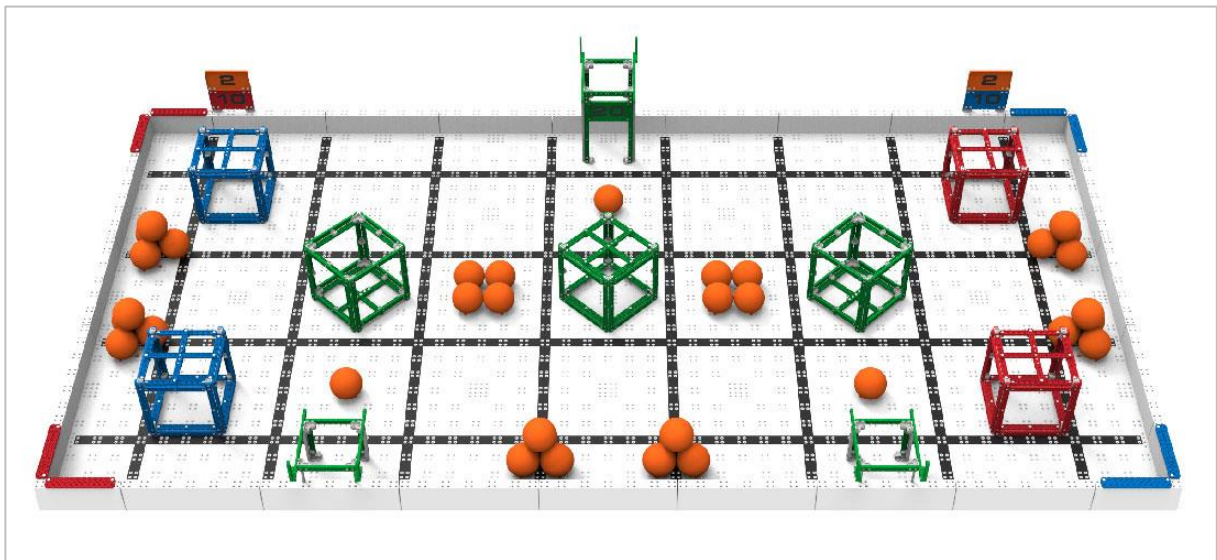
Each year, students can design and build a robot to play against other teams in a game-based engineering challenge presented by the Robotic Education and Competition (REC) Foundation. Tournaments are held year-round at the regional, state, and national levels leading to the VEX Robotics World Championship each April.

The VEX IQ Challenge is played on a 4'x8' rectangular field. Teams program their robots to move around the field grabbing, tossing, and placing game pieces in scoring zones in order to earn the most points.

In the 2019-2020 challenge entitled Squared Away, teams have to move balls in the squares as well as on top of the squares. Like Italy's Trash Robot, the driver will purposefully move and teams will work together to collect and move the squares into the correct color space at the corners of the board.

Here are some typical behaviors for a VEX Robot:

- Moving forward and backward
- Turning left and right
- Grabbing a game object
- Precisely placing a game object
- Sorting between different game objects
- Throwing or launching a game object



The VEX IQ Squared Away Challenge field

There are two types of challenges the teams will tackle. In the Robotic Skills Challenge, teams try to score as many points as possible with their robotic build in two types of matches. Driving Skills Matches are entirely driver controlled and Programming Skills Matches are autonomous with limited student interaction. The second type of challenge is the Teamwork Challenge, in which two robots compete in the challenge as an alliance in 60 second long matches, working together to score the most points.

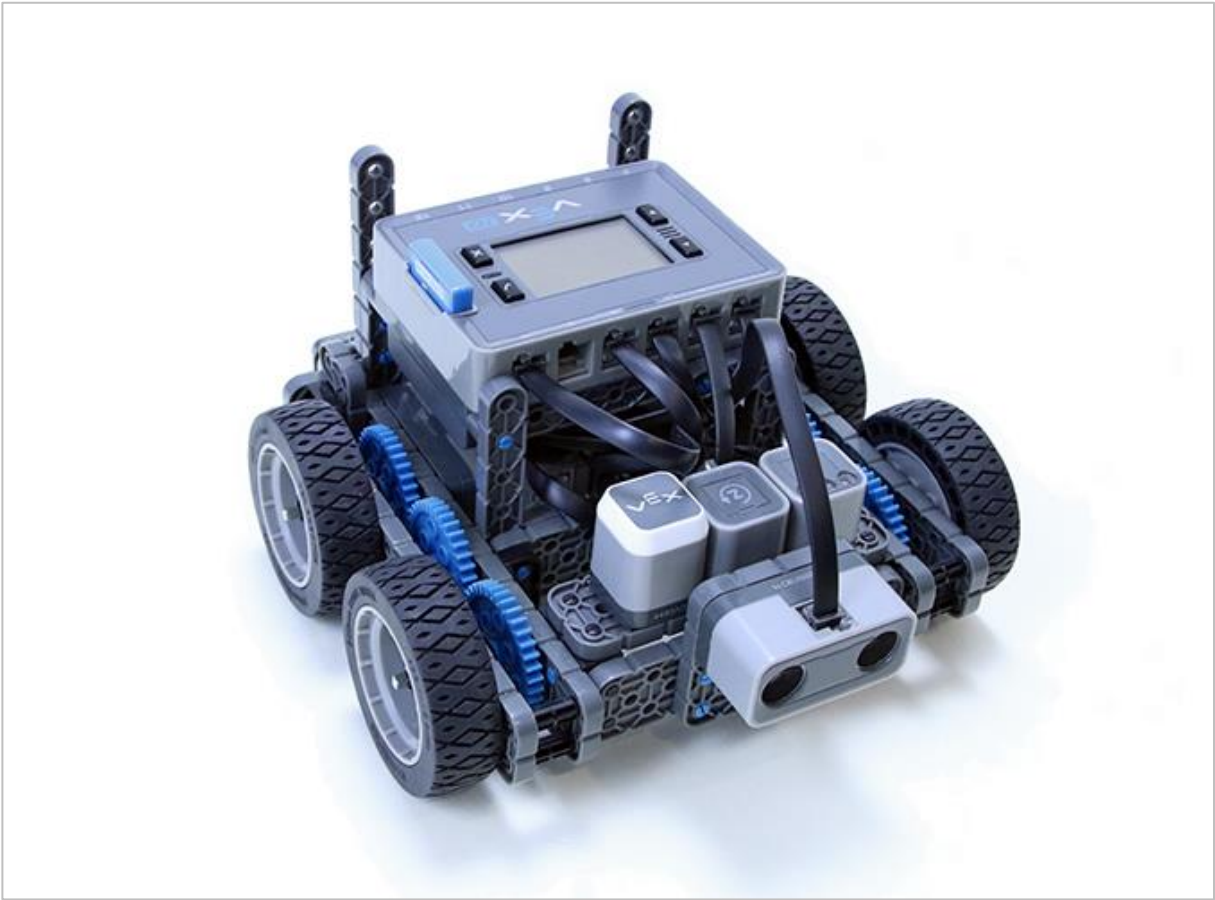
VEX Competitions give students the opportunity to:

- Demonstrate their driving and programming skills.
- Work together as a team to solve problems.
- Meet new people from their community, state, and even other countries.
- Have fun!



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

Improve Your Project



Refine Your Code

Remember your code from the Play section? We're going to play around with that code now, to help refine and revise it to better accomplish your goal. This Rethink activity has 3 options:

- A **Sequencing Challenge**—to help you think about the code you created, how it works, and why it might not work.
- A **Coding Challenge**—to help you practice creating coding sequences that work around additional obstacles.
- An **Efficiency Challenge**—to help you rethink your code and rewrite it to work with fewer blocks.

After your teacher tells your group which challenge to start with, proceed to the next page and follow the steps for that challenge.

For all challenges, 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	Charged Robot Battery
1	VEXcode IQ Blocks
1	Engineering Notebook
1	USB Cable (if using a computer)
1	1x1 Connector Pin

Before you begin 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?

Refine Your Code

Sequencing Challenge

- **Set up**—The Builder and Driver should set up your map and robot the way you had it when you finished the Play section. Place the robot at the Start point. The Programmer should open the Recycle Run project in VEXcode IQ Blocks. The Recorder should review the task to make sure that your project is meeting all the requirements.
- **Review**—The Programmer should run the project as is. While the robot is driving, the whole group should watch it carefully.
- **Troubleshoot**—Is the robot moving the way that you expected it to? If not, where is it different? Are there missed steps? The Recorder should review the list of steps you created aloud with the group. As the recorder reads, the Driver should pick up the robot and move it along the steps; and the Programmer and Builder should check the code sequence to make sure it matches the steps that are being read. When you notice a missed step or incorrect block, pause the process to fix it.
 - Is that the correct block?
 - Are the parameters accurate?
 - Do you need to add more blocks?
- **Refine**—Each time you fix a portion of the coding sequence, the Driver should return the robot to the start position, the Programmer should rerun the project, and the group should watch the robot to make sure the changes are doing what you expected them to. Repeat steps 3 and 4 until your project meets all of the requirements successfully. Check in with your teacher to share your revised code and robot path.

Coding Challenge

- **Set up**—The Builder and Driver should set up your map and robot the way you had it when you finished the Play section. Place the robot at the Start point. The Programmer should open the Recycle Run project in VEXcode IQ Blocks. The Recorder should add the obstacle block to the map in a central location. Check in with your teacher about your obstacle placement before moving on.
- **Review**—The Programmer should run the existing Recycle Run project. The whole group should watch the robot as it moves, and the Driver should stop the project when/if the robot hits the obstacle.
- **Revise**—Locate the point in the `comment` block list and the coding sequence when the robot met the obstacle. Use the `comment` blocks to amend the plan to navigate around the obstacle.

- Rewrite—The Programmer should then amend the code sequence accordingly—removing, adding, or editing parameters of blocks to match the new plan. When the code changes are complete, rerun the project and watch the robot to see if it moves the way that you intended it to. If it does not, try to troubleshoot within your group before asking your teacher for help.
- Troubleshooting—Is the robot moving the way that you expected it to? If not, where is it different? Are there missed steps? The Recorder should review the list of steps you created aloud with the group. For each step, as the recorder reads it, the Driver should pick up the robot and move it along the steps; and the Programmer and Builder should check the code sequence to make sure it matches the steps that are being read. When you notice a missed step or incorrect block, pause the process to fix it.
 - Is that the correct block?
 - Are the parameters accurate?
 - Do you need to add more blocks?

Check in with your teacher to share your new code and robot path.

Efficiency Challenge

- Set up—The Builder and Driver should set up your map and robot the way you had it when you finished the Play section. Place the robot at the Start point. The Programmer should open the Recycle Run project in VEXcode IQ Blocks. The Recorder should review the task to make sure that your project is meeting all the requirements.
- What is code efficiency?
If you needed to get 3 things from your kitchen, you would probably go get all 3 things in one trip, instead of making 3 separate trips. Why? Because it would be more efficient. Efficient means working something out without wasting any time, energy, or materials. So how does that connect to programming?

Code efficiency means that a project is written to work well and quickly, using the least amount of work, or steps, properly, while still doing its job. It is important because it reduces the risk of a program not functioning well, and in turn, causing problems for other parts of a program that are running around it. The efficiency of a code can be a measure of quality, and when we write projects in VEXcode IQ block, we are often looking for the most “efficient” project as the preferred solution.

Basically, code efficiency is a principle that means you want to write a code that accomplishes your goal using the fewest commands, or blocks, possible.

- Rethink—With your whole group, look at the path you designed for your robot to travel. Is there a shorter route the robot could take? Is there a different way to meet the requirements more quickly?
- Revise—Try to amend your code sequence to use 3 less blocks, while still meeting all the requirements. Use these questions to help you find places to revise:

- If we remove this block, what will the robot do?
- Is this block redundant? (Repeating something that doesn't need to be repeated.)
- Is there a single block that can do the work of two or three that are in your sequence?
- Are there any blocks or sequences repeated over and over which could be replaced with a loop/repeat block?

When the code changes are complete, the Programmer should rerun the project and the group should watch the robot to see if it moves the way that you intended it to. If it does not, try to troubleshoot within your group before asking your teacher for help.

Check in with your teacher to share your new code and robot path.

- Troubleshooting—Is the robot moving the way that you expected it to? If not, where is it different? Are there missed steps? The Recorder should review the list of steps you created aloud with the group. For each step, as the recorder reads it, the Driver should pick up the robot and move it along the steps; and the Programmer and Builder should check the code sequence to make sure it matches the steps that are being read. When you notice a missed step or incorrect block, pause the process to fix it.
 - Is that the correct block?
 - Are the parameters accurate?
 - Do you need to add more blocks?

Evaluate and Explain

Reading Code Sequences to Choose a Preferred Solution

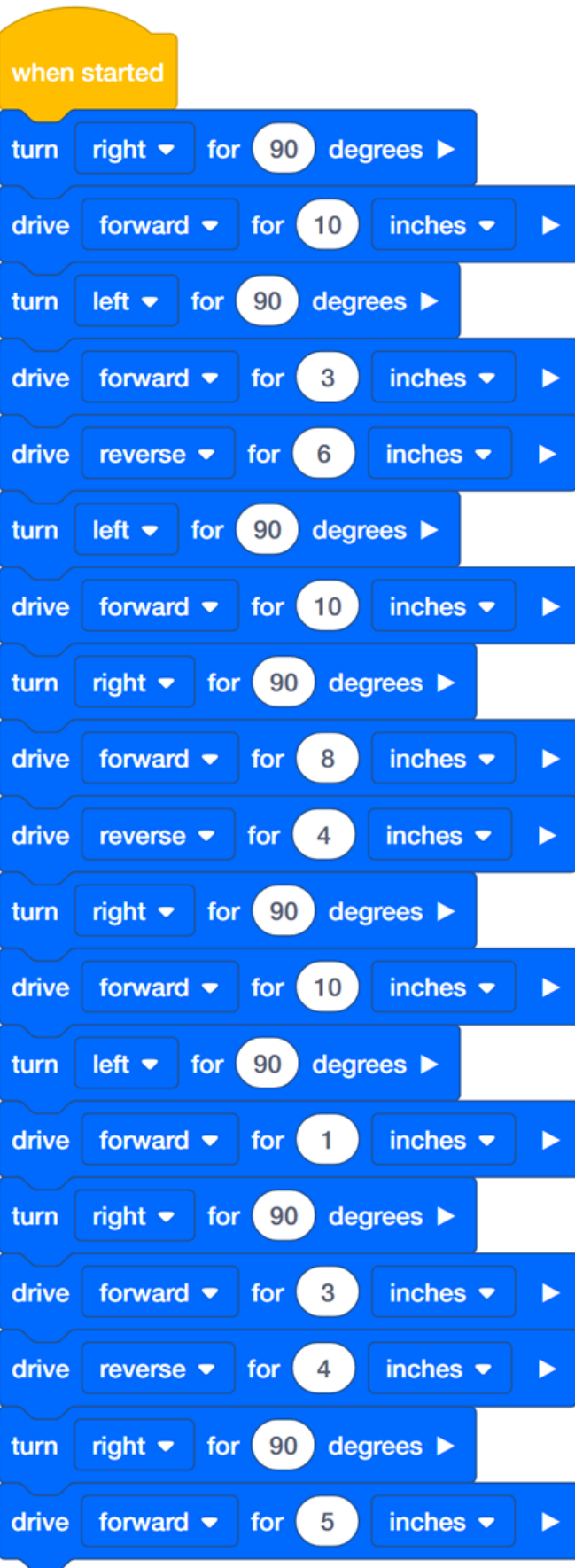
As your class worked on the Play section, you probably noticed that there were many different solutions to the same problem that was presented. That is something that happens often in the world of programming. While there can be different ways to do something, everyone will have a preferred method or solution, and that could be based on a number of factors. In this activity, your group will look at 3 possible code block solutions to a problem, figure out if they work in the intended fashion, then choose a preferred solution from the group and explain why you chose it.

The problem: Create a sequence that, when the robot begins at the Start, will travel into and out of all 3 classrooms (in any order) and return to the End point. The robot must navigate around the stairs, and can go through the doorways marked by the small rectangles on the map image.



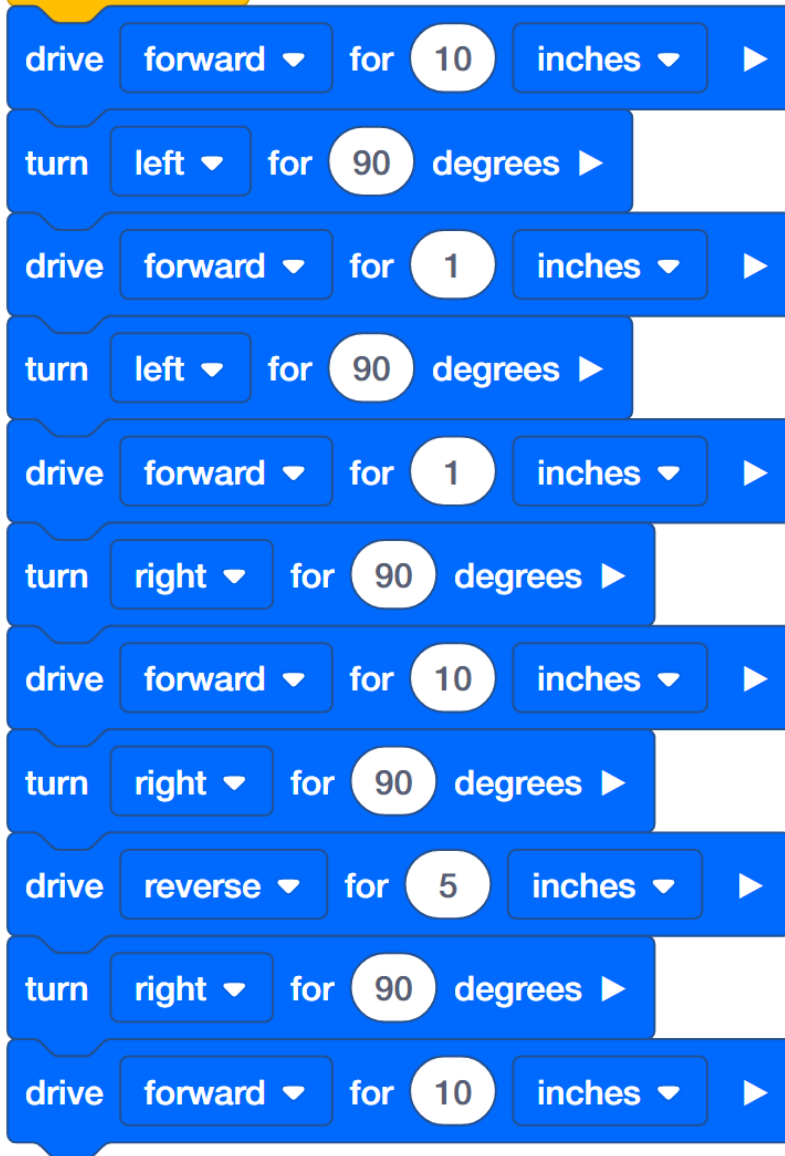
- For each solution below, answer these 3 questions in your engineering notebook:
 - Does this code accomplish the task? Why or why not?
 - If it does work, what are the advantages of this code over the others?
 - What are the disadvantages of this code over the others?
- Most of the time, programming is evaluated on its efficiency, and the preferred solution is one that accomplishes the goal in the most efficient way possible.

Click on the solutions below to see a larger image.



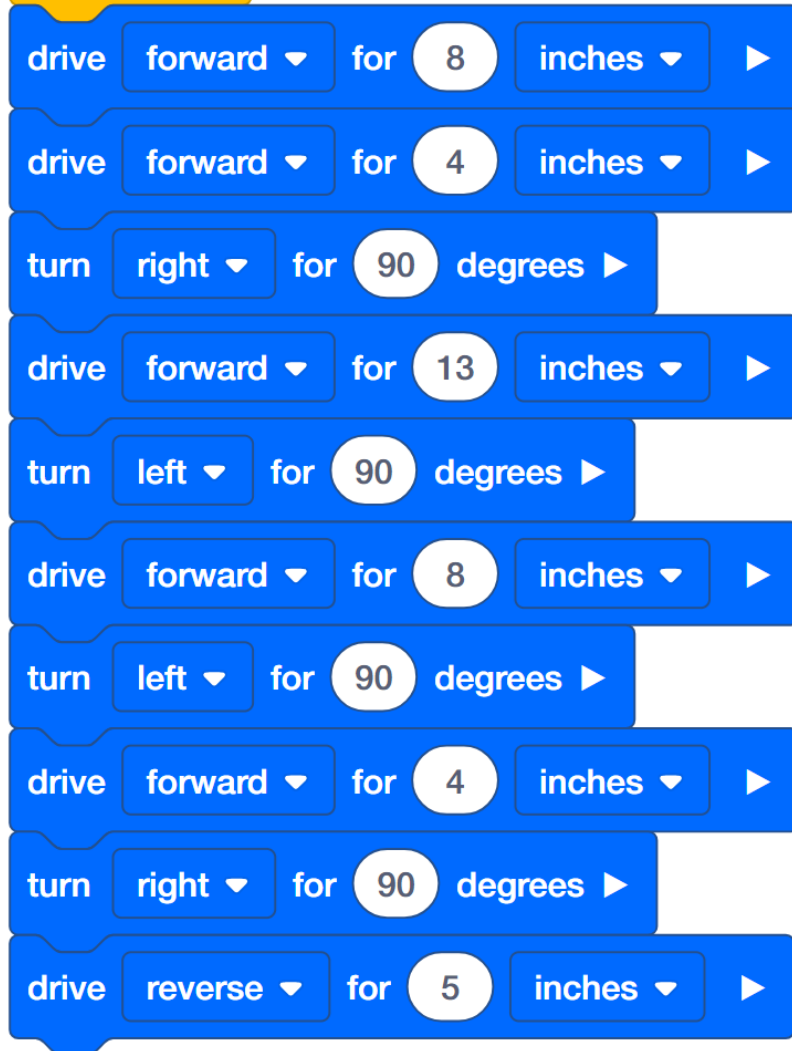
Solution A (inches)

when started



Solution B (inches)

when started

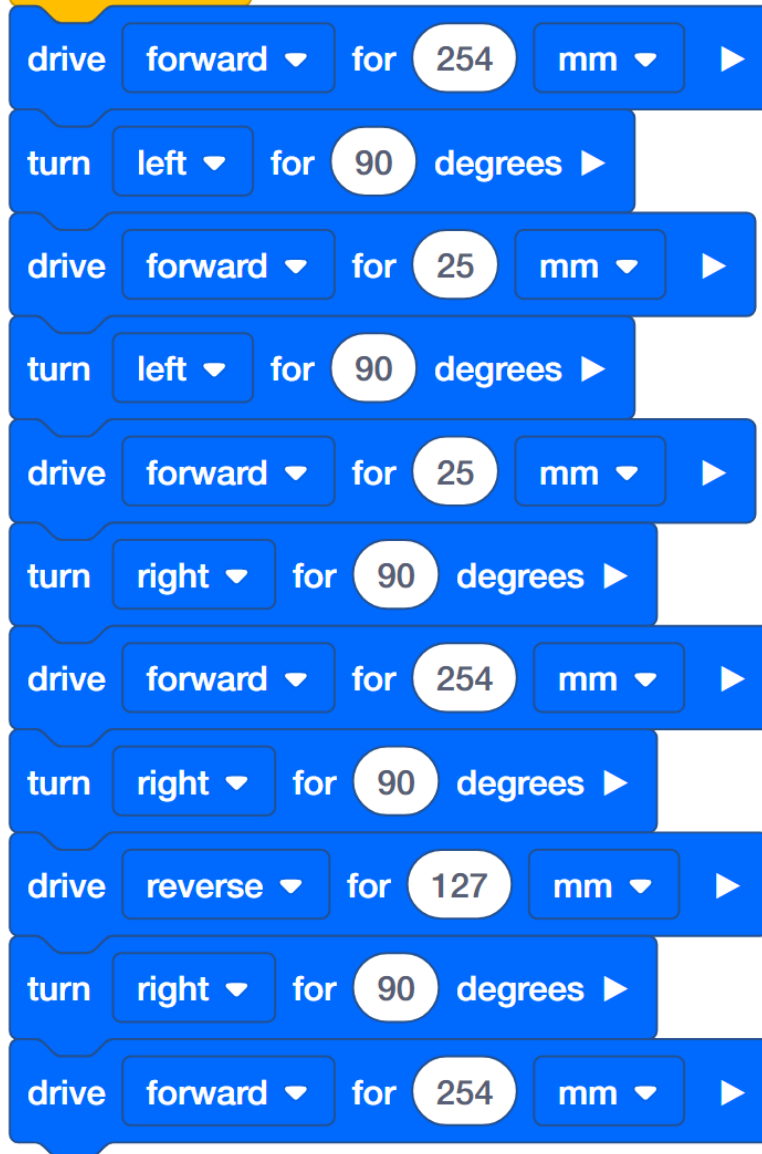


Solution C (inches)



Solution A (metric)

when started



Solution B (metric)



Solution C (metric)

What is Code Efficiency?

If you needed to get 3 things from your kitchen, you would probably go get all 3 things in one trip, instead of making 3 separate trips. Why? Because it would be more efficient. Efficient means “achieving maximum productivity with minimum wasted effort or expense.” So how does that connect to programming?

Code efficiency means that a program is written to work well and quickly, using the least amount of work, or steps, properly, while still doing its job. It is important because it reduces the risk of a program not functioning well, and in turn, causing problems for other parts of a program that are running around it. The efficiency of a code can be a measure of quality, and when we write projects in VEXcode IQ block, we are often looking for the most “efficient” project as the preferred solution.

Basically, code efficiency is a principle that means you want to write a code that accomplishes your goal using the fewest commands, or blocks, possible.

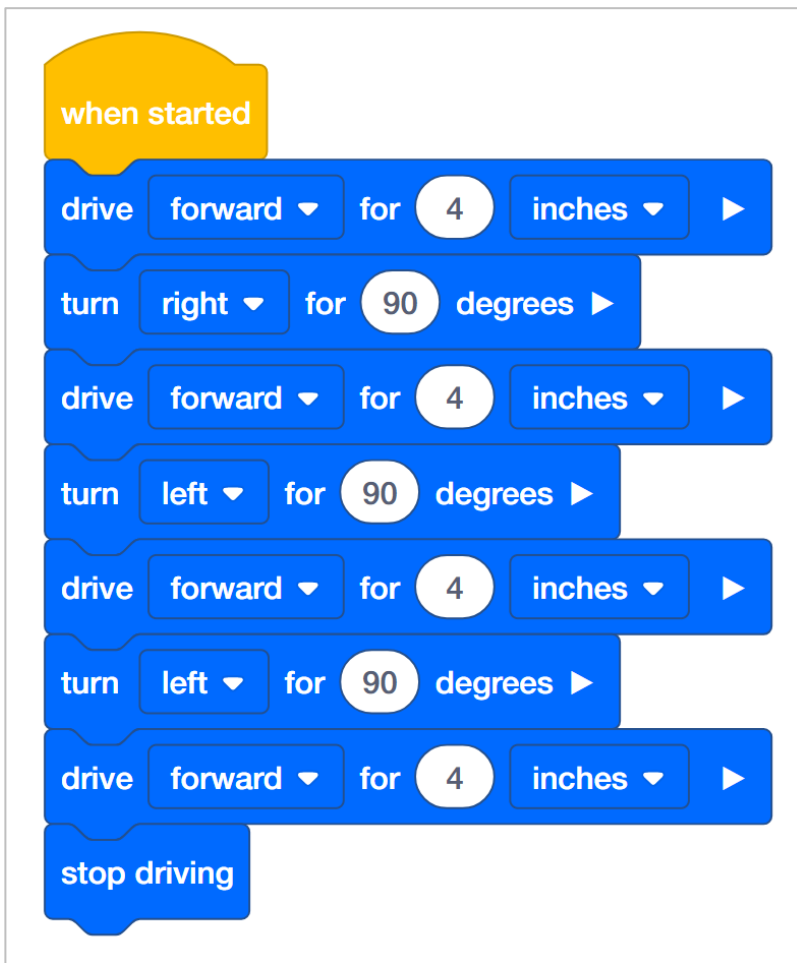
In your Engineering Notebook, answer the following questions:

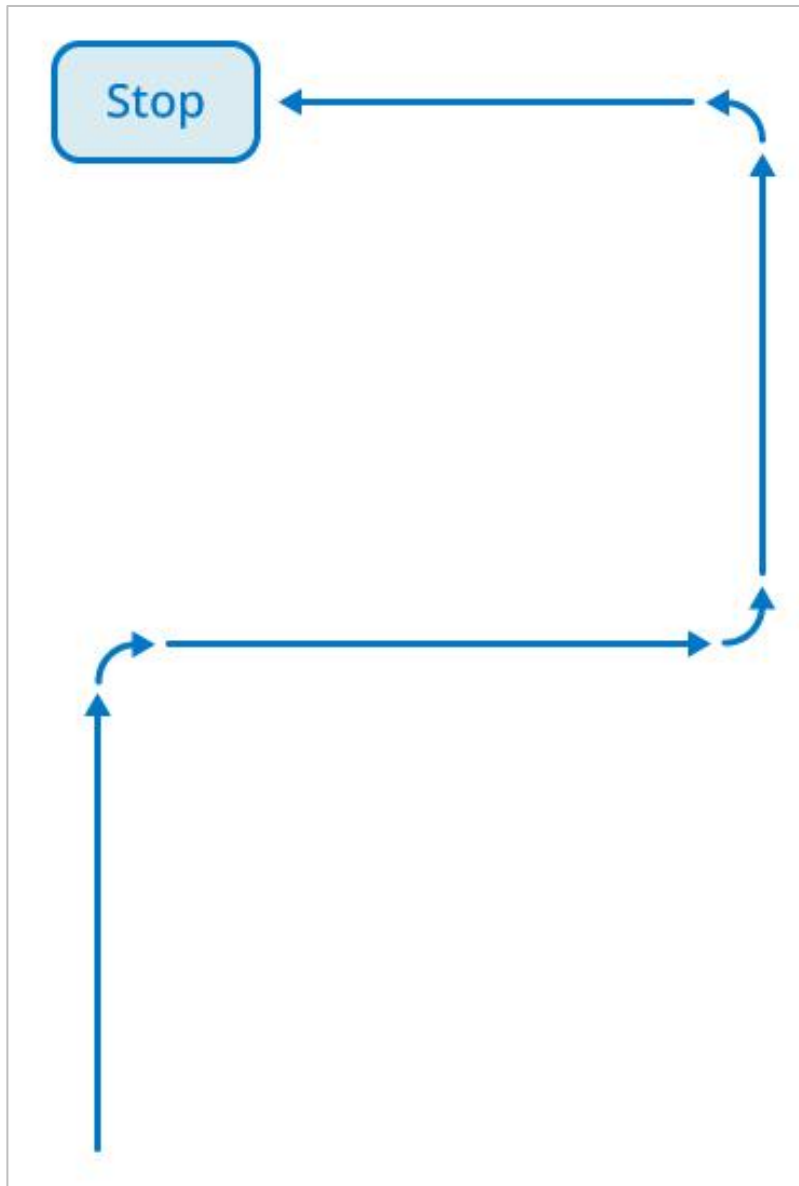
- With code efficiency in mind, which is your preferred solution and why?
- Do you think that is the best solution, why or why not?

Rethink Questions

Answer the following questions in your engineering notebook after completing the activities.

- What are three things you can check to troubleshoot your project, before asking your teacher for help?
- Read the following stack of code blocks, and draw and write the path the robot will take when the project is run.





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

Review

You have accomplished a lot in this STEM Lab! The following questions will help you think back over everything you have learned. You can only answer once, so think carefully before you submit!

1. True or False: Someone with good spatial abilities might also be skilled at planning a route on a map.

- ☐ True
- ☐ False

2. Which of the following is the best definition for the term sequence?

- ☐ A condition, best understood by considering them as “if” statements – if this, then do that
- ☐ Loops, where conditions are checked repeatedly
- ☐ Identifying the steps needed to accomplish a task and the correct order of those steps

3. True or False: A sequence with fewer steps (or blocks) is less likely to contain errors than a sequence with many steps (or blocks).

- ☐ True
- ☐ False

4. All of the following considerations are important in planning your robot's movements EXCEPT _____.

- ☐ How to accomplish the task in the fewest number of steps and/or blocks
- ☐ Which blocks you'll need to program your robot to move, turn, etc.
- ☐ How to accomplish the task with the fewest number of blocks
- ☐ Whether your robot is on a tile or carpeted surface

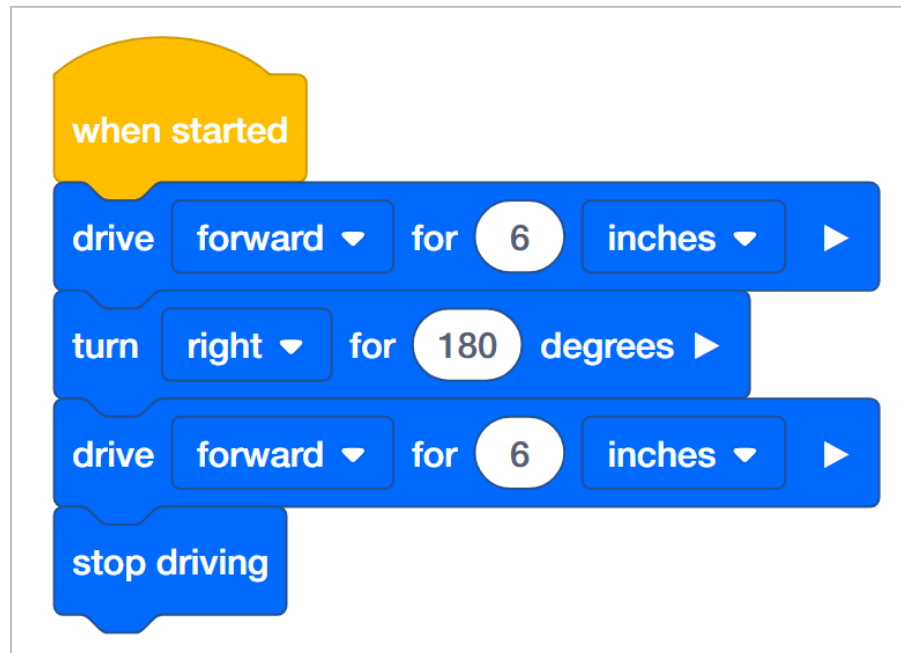
5. What is the purpose of the `comment` block?



- Text in the `comment` block appears on the Robot Brain.
 - Allows programmers to write information to help describe their program
 - Commands in the `comment` block tells the robot which behaviors to perform.
 - None of the above
6. If you click “Run” and your robot does nothing, all of the following are steps you might take to correct the problem EXCEPT _____.
- Make sure the program is displayed on the Robot Brain.
 - Make sure the program is connected to the `when started` block.
 - Make sure the Touch LED is properly connected to the Autopilot.
 - Connect the robot to the Robot Brain, and click download before attempting to Run.
7. If you expect your robot to turn left 90 degrees, and it turns right instead, how would you fix the problem?
- Pick up the robot, and rotate it manually in the direction you’d like it to turn.
 - Disconnect and reconnect the VEX IQ Gyro Sensor.
 - Check the robot configuration is properly configured for Autopilot.
 - Change the direction in the `turn for` block from right to left.
8. True or False: The `comment` block and the note feature can be used interchangeably in VEXcode IQ Blocks.
- True
 - False
9. What is the best order operations in problem solving using your robot?
- 1. Design the solution, 2. Code the steps, 3. Define the problem, 4. Is the problem solved?

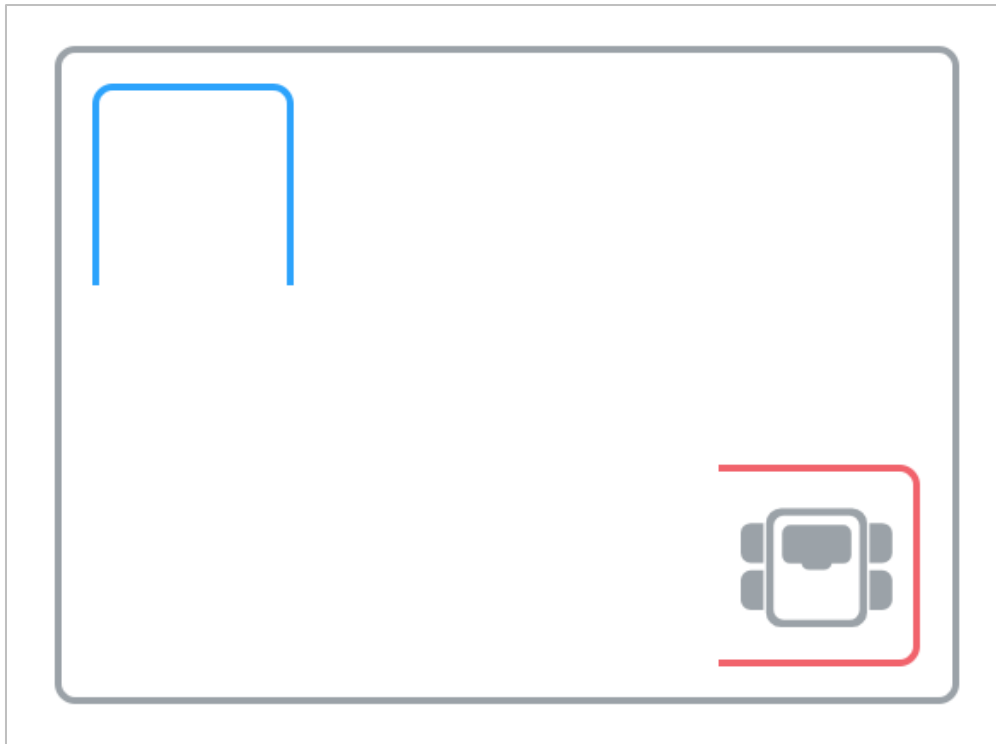
- 1. Code the steps, 2. Define the problem, 3. Design the solution, 4. Is the problem solved?
- 1. Define the problem, 2. Design the solution, 3. Code the steps, 4. Is the problem solved?
- 1. Is the problem solved?, 2. Define the problem, 3. Design the solution, 4. Code the steps

10. Susie was asked to program her robot to drive forward six inches, turn around, and return to its starting location. To successfully accomplish this task, is Susie's code (below) correct?



- Yes
- No

11. Which of the following sequences is the best possible sequence to plan the robot's movements from the red box to the blue box in the diagram below?



- 1) Turn left for 90 degrees
2) Drive forward for 6 inches
3) Turn right for 90 degrees
4) Drive forward for 6 inches
5) Turn left for 90 degrees
6) Drive forward 14 inches
7) Turn right for 90 degrees
8) Drive forward for 6 inches
- 1) Turn left for 90 degrees
2) Drive forward for 20 inches
3) Turn right for 90 degrees
4) Drive forward for 12 inches
- 1) Turn left for 45 degrees
2) Drive forward for 20 inches
3) Turn right for 45 degrees
4) Drive forward for 12 inches

APPENDIX

Additional information, resources, and materials.

Knowledge Base Articles

Links to the VEX Robotics Knowledge Base Articles for this STEM Lab:

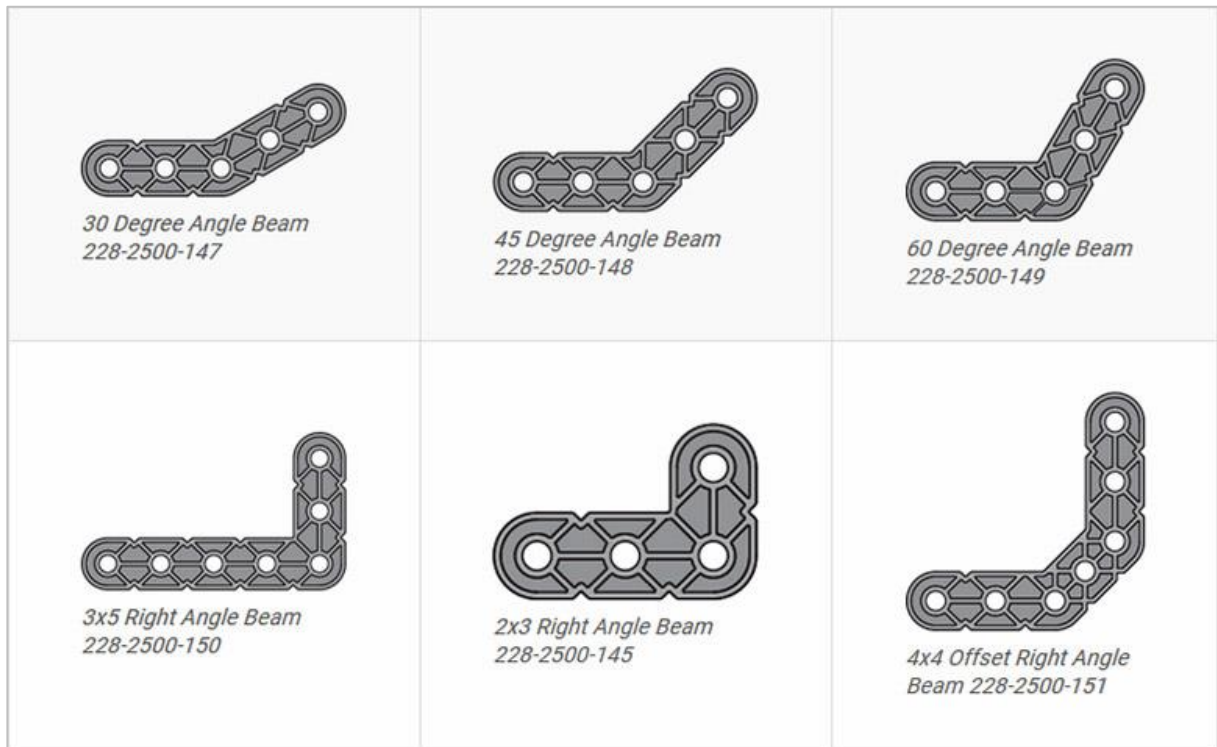
- How to Turn On/Off a VEX IQ Robot Brain
<https://kb.vex.com/hc/en-us/articles/360035952571-How-to-Turn-On-Off-a-VEX-IQ-Robot-Brain>
- How to Read Indicator Lights on the VEX IQ Robot Brain
<https://kb.vex.com/hc/en-us/articles/360035590672-How-to-Read-Indicator-Lights-on-the-VEX-IQ-Robot-Brain>
- How to Navigate the VEX IQ Robot Brain
<https://kb.vex.com/hc/en-us/articles/360035952331-How-to-Navigate-the-VEX-IQ-Robot-Brain>
- How to Connect VEX IQ Devices to Smart Ports
<https://kb.vex.com/hc/en-us/articles/360035952151-How-to-Connect-VEX-IQ-Devices-to-Smart-Ports>
- How to Install or Remove the VEX IQ Robot Battery
<https://kb.vex.com/hc/en-us/articles/360035951991-How-to-Install-or-Remove-the-VEX-IQ-Robot-Battery>
- How to Charge the VEX IQ Robot Battery
<https://kb.vex.com/hc/en-us/articles/360035955011-How-to-Charge-the-VEX-IQ-Robot-Battery>
- How to Use the Autopilot Program in the Demos Folder
<https://kb.vex.com/hc/en-us/articles/360035952031-How-to-Use-the-Autopilot-Program-in-the-Demos-Folder>
- Best Practices for Preserving the VEX IQ Robot Battery's Life
<https://kb.vex.com/hc/en-us/articles/360035953671-Best-Practices-for-Preserving-the-VEX-IQ-Robot-Battery-s-Life>
- Ideas for Organizing the VEX IQ Super Kit
<https://kb.vex.com/hc/en-us/articles/360035590332-Ideas-for-Organizing-the-VEX-IQ-Super-Kit>
- VEX IQ Brain Status (USB Cable)
<https://kb.vex.com/hc/en-us/articles/360035955411-How-to-Understand-the-VEX-IQ-Brain-Status-Icon-USB-VEXcode-IQ-Blocks>

Links to VEXcode IQ Blocks Knowledge Base Articles for this STEM Lab:

- How to Begin a New Project in VEXcode IQ Blocks
<https://kb.vex.com/hc/en-us/articles/360035954551-How-to-Begin-a-New-Project-VEXcode-IQ-Blocks>

- How to Download and Run a Project
<https://kb.vex.com/hc/en-us/articles/360035591232-How-to-Download-and-Run-a-Project-VEXcode-IQ-Blocks>
- How to Save a Project on Windows
<https://kb.vex.com/hc/en-us/articles/360035954531-How-to-Save-a-Project-on-Windows-VEXcode-IQ-Blocks>
- How to Save a Project on macOS
<https://kb.vex.com/hc/en-us/articles/360035954511-How-to-Save-a-Project-on-macOS-VEXcode-IQ-Blocks>
- How to Save a Project on Chromebook
<https://kb.vex.com/hc/en-us/articles/360035955351-How-to-Save-on-a-Chromebook-VEXcode-IQ-Blocks>
- How to Download to a Selected Slot on the Brain
<https://kb.vex.com/hc/en-us/articles/360035591292-How-to-Download-to-a-Selected-Slot-on-the-Brain-VEXcode-IQ-Blocks>

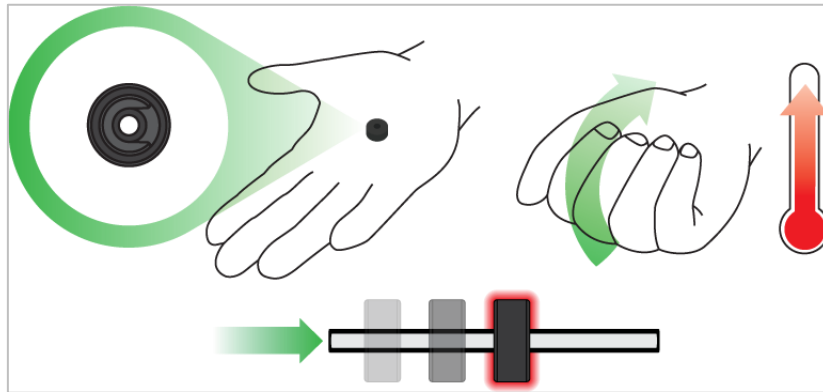
Identifying Angle Beams



How to Identify the Different Angles of the Angled Beams

There are four different types of beams that have a bend at an angle: 30° Angle Beams, 45° Angle Beams, 60° Angle Beams, and Right Angle (90°) Beams. There are also three types of Right Angle Beams: 3x5, 2x3, and Offset. The best way to tell which angles are which is to stack the beams on top of each other. Then you can compare how they look. You can also use a protractor to measure the angle of the beam.

Installing Rubber Shaft Collars

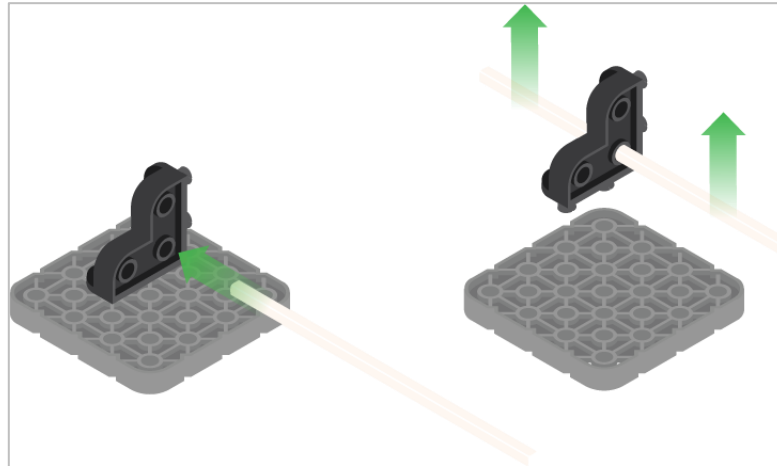


Using your hand to warm a Rubber Shaft Collar

Rubber Softens as it gets Warm

Hold the Rubber Shaft Collars in your hand for 15-30 seconds before you slide them onto a shaft. Holding the Rubber Shaft Collar in your hand will warm and soften the rubber to make it easier to slide onto a shaft.

Removing Connectors from Beams and Plates

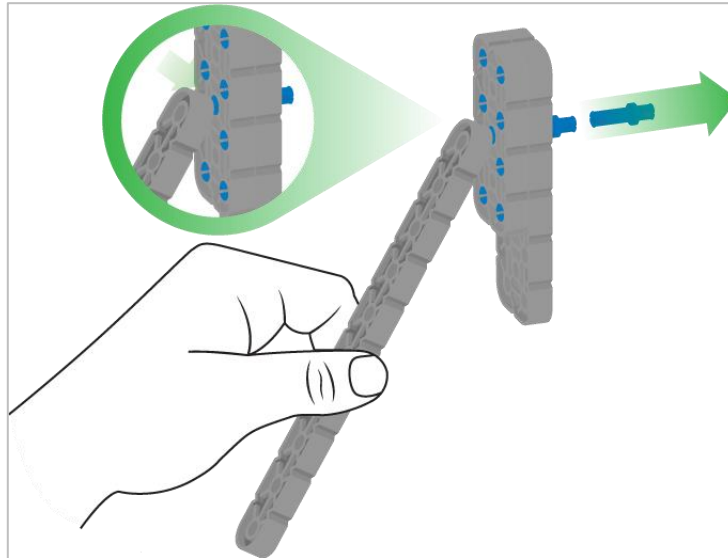


Using a pitch shaft to remove a corner connector

How to Easily Remove Connectors

You can easily remove corner connectors from beams or plates by placing a metal shaft through one of the holes of the corner connector and pulling outward while holding down the beam or plate.

Removing Pins from VEX IQ Beams and Plates



Removing a pin from a plate assembly using a beam

How to Easily Remove Pins from Beams and Plates

You can quickly remove connector pins from beams or plates by pressing a beam against the back of the pin, which partially pushes the pin out, so you can remove it with your fingers. You can use this technique to more easily remove pins from individual plates and beams, or from built structures.

Removing Standoffs from Mini Standoff Connectors

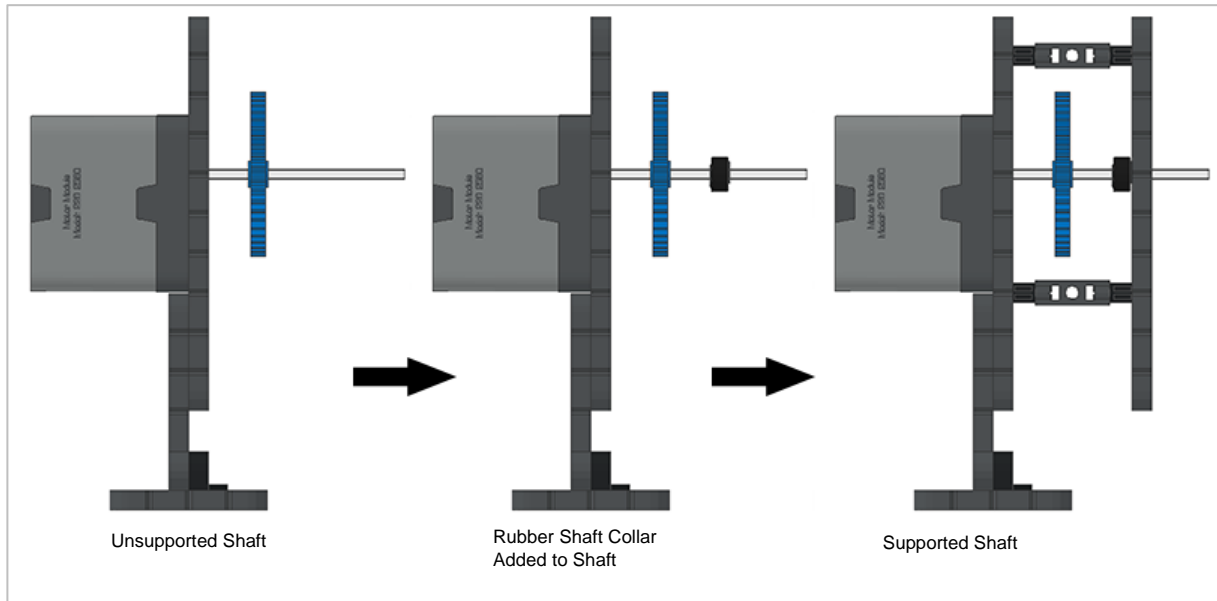


Removal of a standoff from a Mini Standoff Connector

How to Easily Remove Parts from Mini Standoff Connectors

Standoffs and Mini Standoff Connectors can be separated by pushing a shaft through the Mini Standoff Connector. The same technique can be used for parts with similar ends in Mini Standoff Connectors, such as pins.

Supporting Shafts using Rubber Shaft Collars

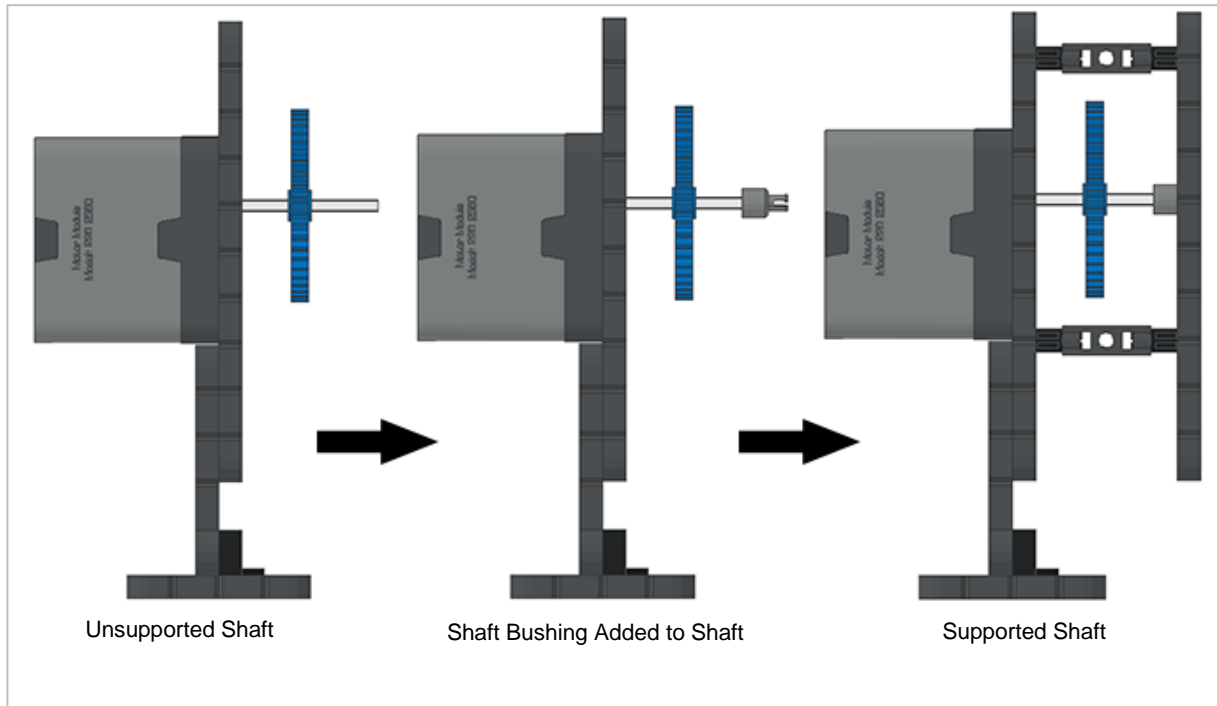


Supporting a shaft with a Rubber Shaft Collar

How to Support Shafts with Rubber Shaft Collars

Shafts can fall out of place or alignment very easily if they aren't supported properly. You can make a shaft more secure and prevent it from falling out of place by putting a Rubber Shaft Collar before the end of it. You can then connect the shaft to a support structure with the shaft collar positioned against it. That will allow the shaft to turn but will prevent it from wobbling or falling out.

Supporting Shafts using Shaft Bushings



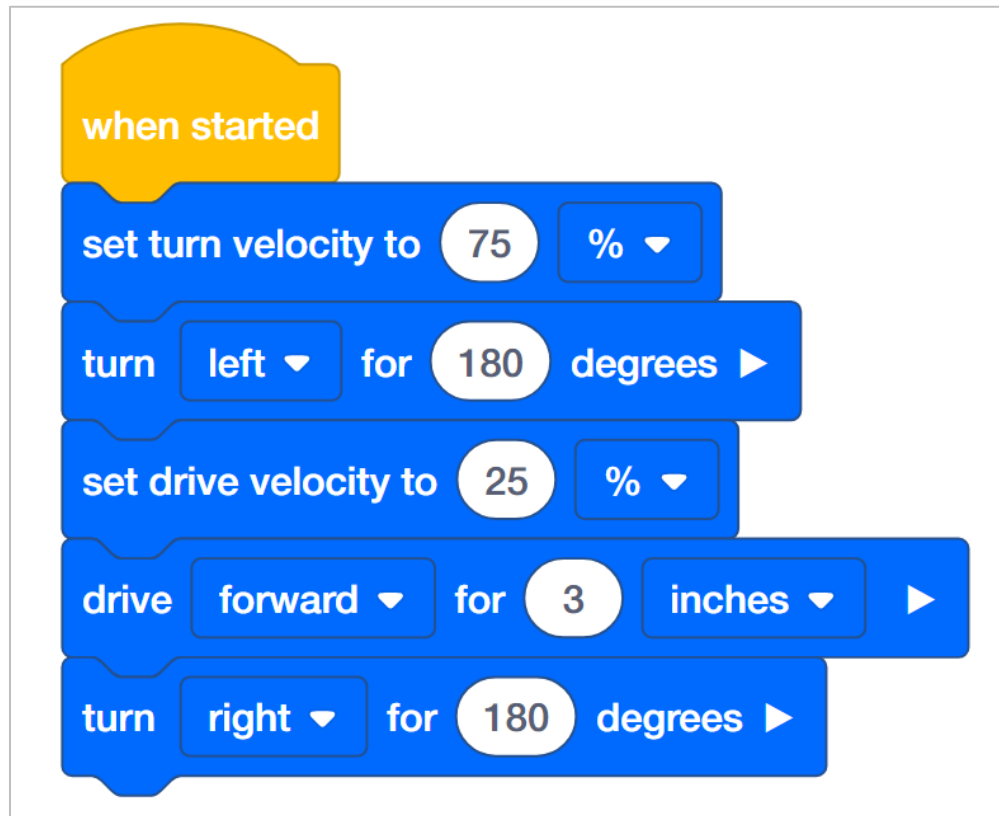
Supporting a shaft with a Shaft Bushing

How to Support Shafts Using Shaft Bushings

Shafts can fall out of place or alignment very easily if they aren't supported properly. You can make a shaft more secure and prevent it from falling out of place by putting a bushing at the end of it. You can then connect that bushing into another beam or additional part. That will allow the shaft to turn but will prevent it from wobbling or falling out.

Popups within this STEM Lab

Tag You're It! Solution

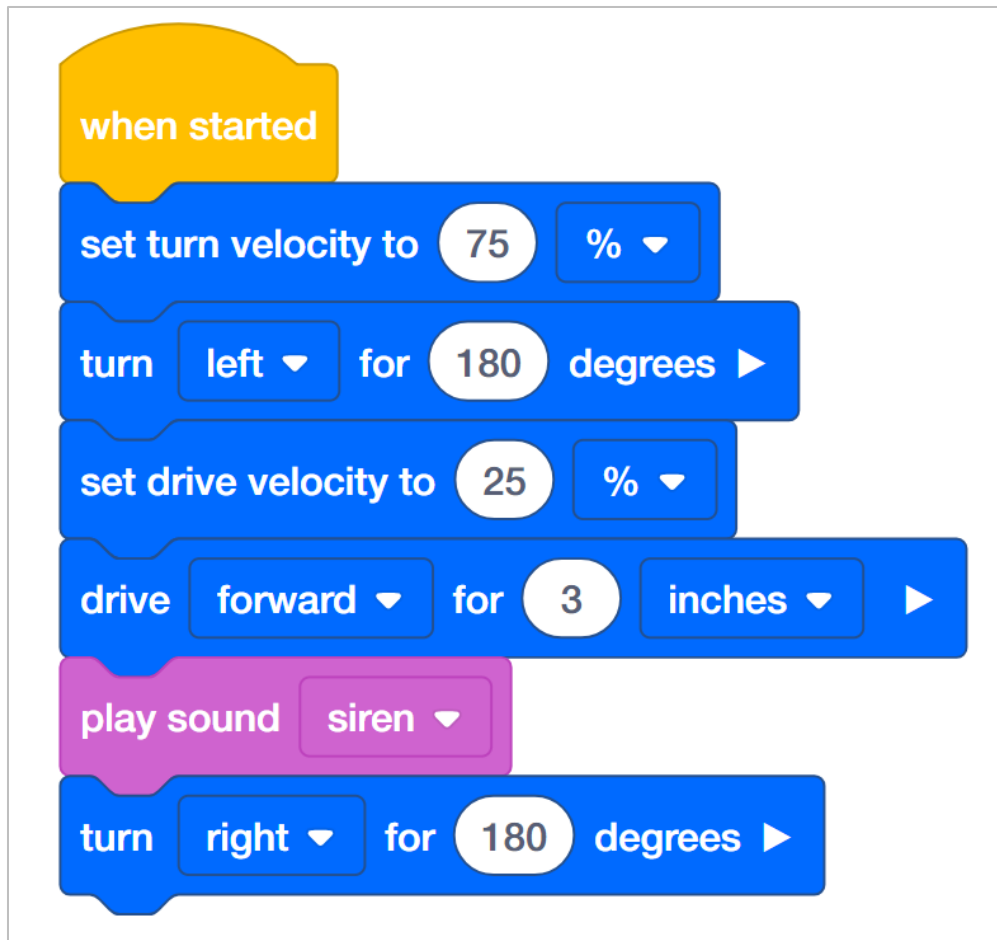


Students were prompted with a very open-ended challenge. The Autopilot needs to turn quickly, drive slowly, and then turn quickly again. The definitions of quickly and slowly were not provided and so, they could be any velocities that are comparably faster or slower than each other. Any project that does all of the following is correct:

- Turns the robot quickly in any direction for any distance.
- Drives the robot slowly in any direction for any distance.
- Turns the robot quickly in any direction for any distance.
- Students might have a block to set the velocity each time the robot turns. This is a teachable moment when you can point out that setting the velocity for driving or turning can be done once unless they want to change the velocity.

Note: Velocity settings close to 100% are unnecessary and not recommended.

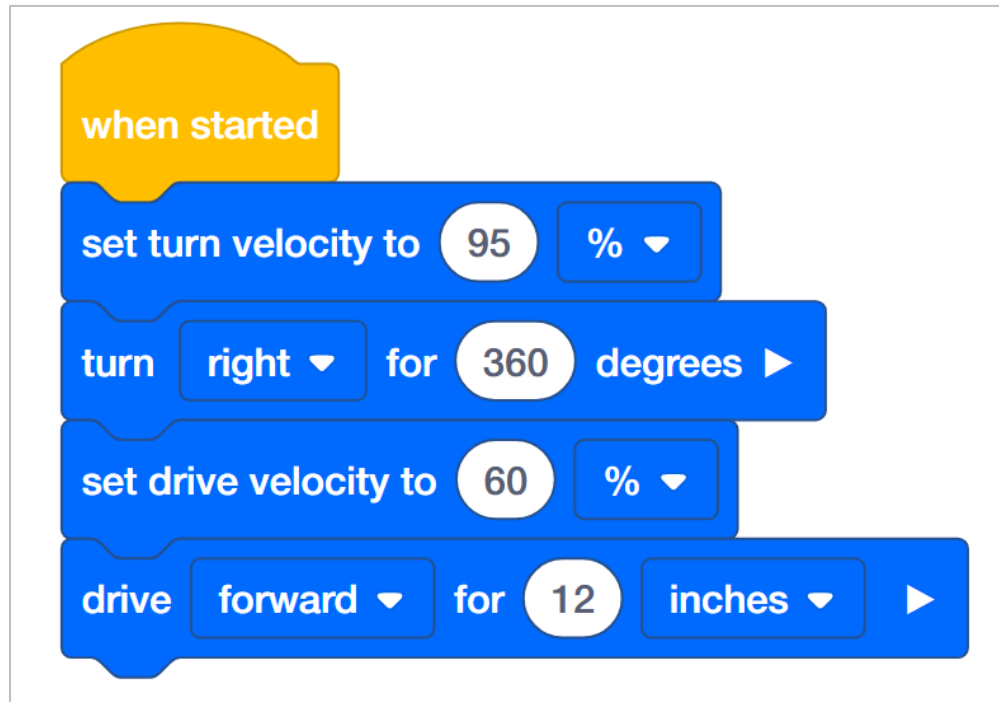
Bonus:



The Bonus Challenge requires the addition of only one `play sound` block. The students should have the `play sound` block after the robot has moved forward to indicate that the robot has moved forward to tag someone.

Robot Waiter Solution

An example solution to Robot Waiter:

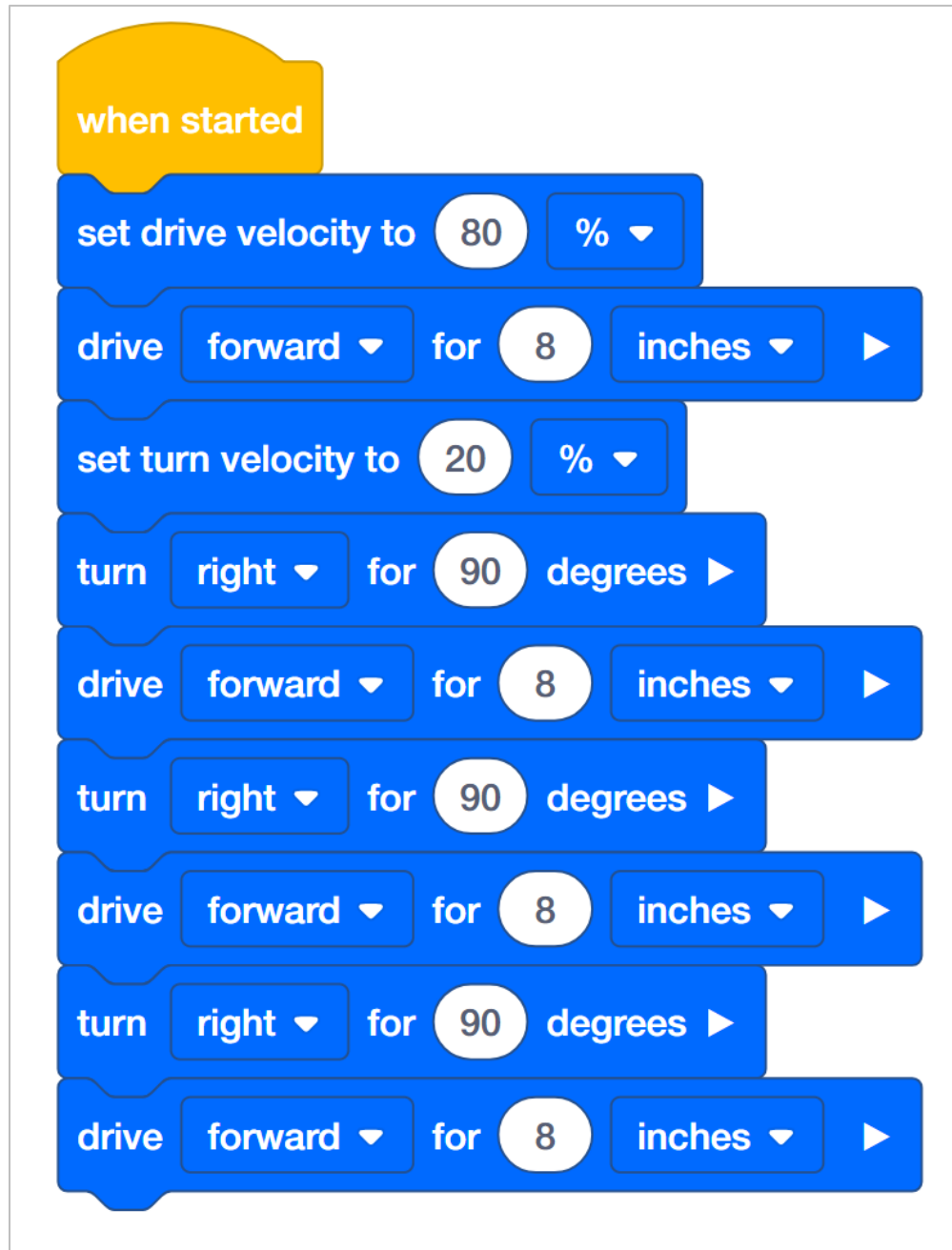


Students were prompted with an open-ended challenge: to see how fast the Autopilot can turn 360 degrees and drive 12 inches while keeping a 6x Pitch Standoff balanced on the brain. Because of the shape of the standoff, the velocity for turning can be set higher than the velocity for driving.

Note: Velocity settings close to 100% are unnecessary and not recommended.

Protect the Castle! Solution

An example solution to Protect the Castle!:



Students were challenged to have the Autopilot move in a square. They were told to have the robot drive quickly along the sides and turn slowly at the corners. Again, the definitions of quickly and slowly were not provided and so, they could be any velocities that are

comparably faster or slower than each other. Any project that does all of the following is correct:

- Sets the velocity for driving higher than the velocity for turning.
- Drives the robot quickly forward at least four times.
- Turns the robot slowly for 90 degrees at least three times.
- Students might have additional blocks to set the velocity each time the robot drives or turns. This is a teachable moment when you can point out that setting the velocity for driving or turning can be done once unless they want to change the velocity.

Note: Velocity settings close to 100% are unnecessary and not recommended.

Bonus:

The Bonus Challenge requires the addition of at least four set Touch LED color blocks positioned in the stack before each drive for block.

when started

set drive velocity to 80 %

set TouchLED5 color to green

drive forward for 8 inches

set turn velocity to 20 %

turn right for 90 degrees

set TouchLED5 color to yellow green

drive forward for 8 inches

turn right for 90 degrees

set TouchLED5 color to purple

drive forward for 8 inches

turn right for 90 degrees

set TouchLED5 color to red orange

drive forward for 8 inches

Programming Precise Distances and Velocities

How are blocks like `drive for`, `turn for`, `set drive velocity`, and `set turn velocity` able to program a robot to move precise distances at precise velocities? In the Robot Configuration, you are able to specify the size of the wheels and the gear ratio of the drivetrain. For example, the typical wheel size for the Autopilot is 200 millimeters and the typical gear ratio is 1:1. These are the default settings in the Robot Configuration.

What does wheel size actually mean? The listed wheel size is the actual wheel circumference. This means that every time the wheel completes one full rotation, it travels 200 millimeters. When the `drive for` block is programmed to move forward for a specific amount of inches or millimeters, the programming logic inside of the `drive for` block performs math calculations. For example, if the `drive for` block is set to move forward for 2000 millimeters, that means that the wheels will travel for 10 complete rotations. The `drive for` block converts the distance into the number of degrees the motor needs to rotate for. The number of degrees is then sent to the Smart Motor, which uses sensors inside of the motor to travel for the set distance.

All of this math is completed with the programming logic built into the `drive for` block.

The programming logic for the `turn for` block uses the Autopilot's Gyro Sensor to detect when the robot has turned for the set distance. The Gyro Sensor measures the amount of change in the robot's position or heading. So the Autopilot turns until the Gyro Sensor's data indicates the robot has turned for the set number of degrees and then the Drivetrain stops.

The `set drive velocity` or `set turn velocity` block works because there is a microprocessor within the Smart Motor that sets how often the internal motor turns on. The internal motor does not actually have different "velocities" as it can only be turned "on" or "off" - like a light switch. A velocity of 50% velocity does not mean that the motor is half turned on - instead, it means that the motor is turned on at full speed, but only half of the time.

What this means is that the microprocessor inside of the motor turns the motor off and on at different intervals - or duty cycles - to provide the illusion of different velocities. The microprocessor can turn the motor on and off over a thousand times a second to give a wide

range of different speeds.

Imagine if you were able to turn a light on and off incredibly fast - you would effectively be able to change the brightness in the room depending upon how long the light was on versus how long the light was off each second. But, the brightness of the light's bulbs would not be changing. The only factor changing would be the amount of time the light was on.

That is how the Smart Motors set a velocity. They turn on and off faster than is perceivable so that they seem to be powered at specific speeds. The blocks for setting velocity let the programmer enter a velocity that is sent to the microprocessor which sets the rate at which the motor turns on and off.

Percentages, Fractions, & Decimals

A percent is a ratio that compares a number to 100 using division. You can express percents as fractions and decimals by dividing by 100.

For example: $65\% = 65 \div 100$ or $65/100$

Provide the following example for the students before they begin trying to convert percents to fractions on their own: If I was working with a velocity of 65%, I would express this percent as a fraction by writing 65/100.

Then, ask your students to choose a velocity that they have been working with and take a few moments to write that velocity as a fraction instead of a percent in their engineering notebooks. The students should write the velocity that they have been using over 100. Students may also reduce fractions as they see fit. For example, 60% written as a fraction is 60/100. This is the same as 30/50 or $\frac{3}{5}$.

Percents can also be written as decimals. Decimals are written using the scale of 10 (tenths, hundredths, thousandths). Decimals are commonly used to represent a whole number plus a fraction.

Decimal Place Value Chart													
Millions	Hundred Thousands	Ten Thousands	Thousands	Hundreds	Tens	Ones	Decimal point	Tenths	Hundredths	Thousandth	Ten - Thousandth	Hundred - Thousandth	Millionths
Whole part							•	Decimal part					

Since percentages are out of 100, decimals will also follow the same pattern extending two units to the right of the decimal point to the hundredths place.

For example: $65\% = 65 \div 100$ or $65/100 = 0.65$

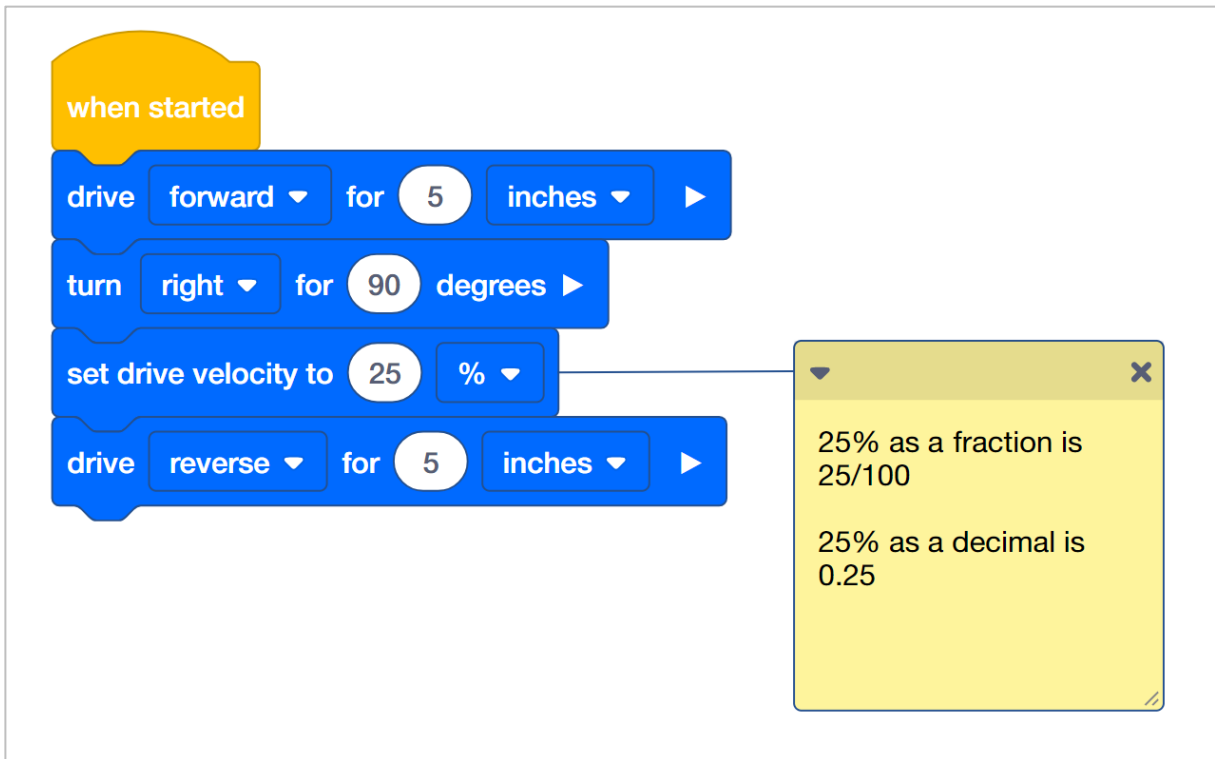
Provide the following example for the students before they begin converting percents to decimals on their own: if I was working with a velocity of 65%. I would express this percent as a decimal by writing 0.65.

Then, ask your students to take a few moments to rewrite the velocity that they had converted to a fraction earlier. Now they should write it as a decimal in their engineering notebooks. The students should write the velocity out to two decimal places.

Once the students have all three representations of the same amount (percent, fraction, and decimal) ask them to take a few moments to share ideas with a partner as to how these three representations are connected. Ask the students to write both of their explanations in their engineering notebook.

Connect the concept of conversions to the projects the students have created. Ask the students to read the project they have just worked on in the groups that they were in previously (Builder, Programmer, Driver and Recorder). Ask the group to convert the percentages of the blocks used to set velocities to both fractions and decimals. Ask the Recorder to note all of the work used for the conversions in the engineering notebook. Once the group has done all of the necessary converting, ask the Programmer to insert the

conversion into a note attached to the `set drive velocity` block.



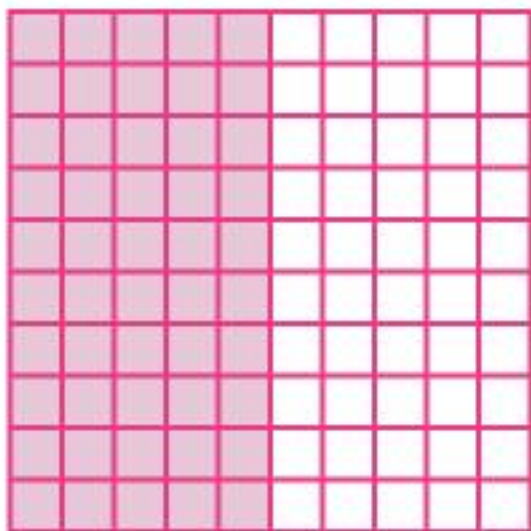
If the group need assistance attaching a note to a block in the project, direct them here.

Begin a discussion with the students about how all three of the representations are connected. Responses will vary, but students should notice how each representation is out of 100. Students who may have reduced their fractions may not notice that fractions, percents and two-digit decimals are being represented with 100 being the whole. Even if a student reduces $50/100$ to $\frac{1}{2}$, the student can still make a connection as to how this relates to some part out of the whole of 100.

Students can practice writing percents as fractions by filling in or recognizing parts of a fraction grid to help students visually understand that a percent is comparative to 100 by using graph paper. Below is an example of how students can place all three representations together as well as a filled-in fraction grid:

50%

50 out of 100



$$\frac{50}{100} = \frac{1}{2} = 0.5$$

Organizing Students into Groups for Exploration

Organize the students into groups before beginning the exploration. Students can be organized into groups of two to four students when participating in the exploration. The following roles can be utilized during the exploration:

- **Builder** — This person checks that the robot is properly built and ready (e.g., Are all the motors and sensors plugged into the correct ports? Is the Robot Brain turned on?) before a project is run.
- **Programmer** — This person will use the set drive velocity block and others to create a project on the computer or tablet. This person will also download the project to the robot.
- **Driver** — This person selects the project and then runs it on the robot. This person will also be the one to retrieve the robot after it is run.
- **Recorder** — This person writes down all of the group answers/reflections in the engineering notebook.

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 definitions 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.

Exploration of Velocity Outline

The outline for the Exploration of Velocity is as follows:

- Recognize the set drive velocity block.
- Review how to find Help information about blocks in VEX IQ Blocks.
- Check that the VEX IQ Autopilot is ready for the exploration.
- Start a new project in VEXcode IQ Blocks.
- Rename and save the project (Windows, MacOS, Chromebook).
- Create the Drive Velocity project that moves the Autopilot at different velocities.
- Download and run the project.
- Change the Drive Velocity project to move the Autopilot in reverse.
- Download and run the project.
- Add `set turn velocity` and `turn for` blocks to the Drive Velocity project.
- Download and run the project.
- Wrap up the activity with a discussion.

Changing the Autopilot's Velocity

These are the three versions of the project that students are asked to build in the Play section:

when started

drive forward ▾ for 5 inches ▾ ▶

set drive velocity to 25 % ▾

drive forward ▾ for 5 inches ▾ ▶

set drive velocity to 75 % ▾

drive forward ▾ for 5 inches ▾ ▶

when started

drive forward ▾ for 5 inches ▾ ▶

set drive velocity to 25 % ▾

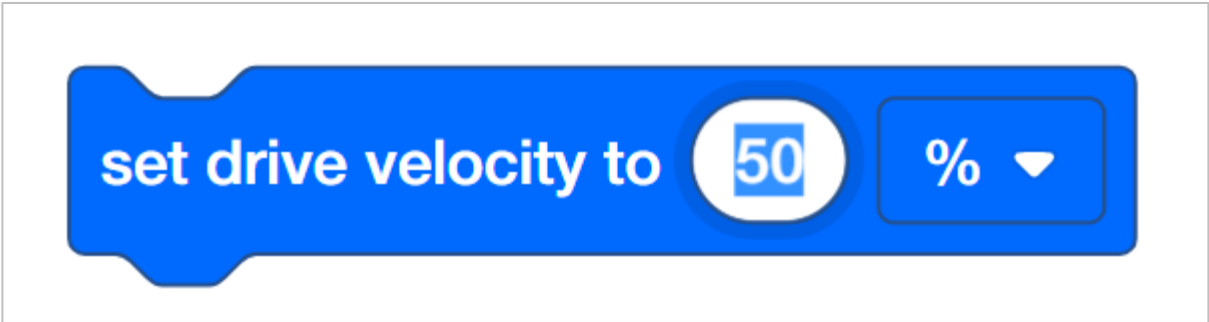
drive reverse ▾ for 5 inches ▾ ▶

forward

✓ reverse

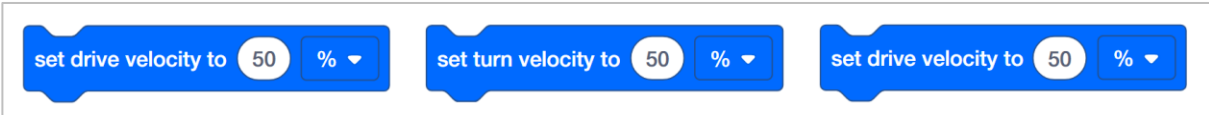


In order to set the Autopilot's velocity, the student will need to change the default 50 to a value of their choosing. Anything editable inside of a block is called a parameter. For this STEM Lab, the student does not need to change the second parameter because all activities will only use the percentage (%) of the maximum velocity.



See the Help feature within VEXcode IQ Blocks for further information about these blocks or other blocks used in this project.

Students might ask about the arrow at the end of the drive for and turn for blocks. Please see this page from the VEX Robotics Knowledge Base for an explanation of setting blocks that can be waiting or non-waiting.



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.

Configuration for Autopilot's Motors and Sensors

The configuration for Autopilot's motors and sensors are:

- Port 1: Left Motor
- Port 2: Distance Sensor
- Port 3: Color Sensor
- Port 4: Gyro Sensor
- Port 5: Touch LED
- Port 6: Right Motor
- Port 8: Bumper Switch
- Port 9: Bumper Switch