



Drive Forward and Reverse



Students will explore robot behaviors and program the Autopilot robot to drive forward and in reverse.



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

The Completed Look of the Build



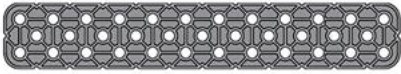
The completed Autopilot robot build.


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.


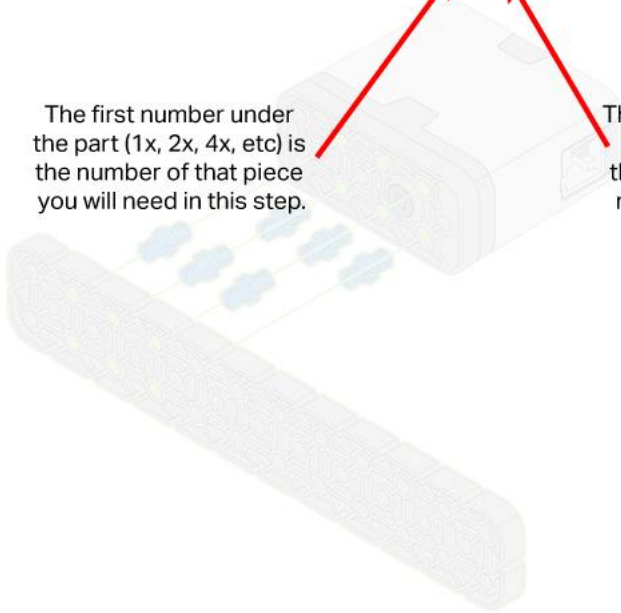
 **1x** - 2x12 Beam

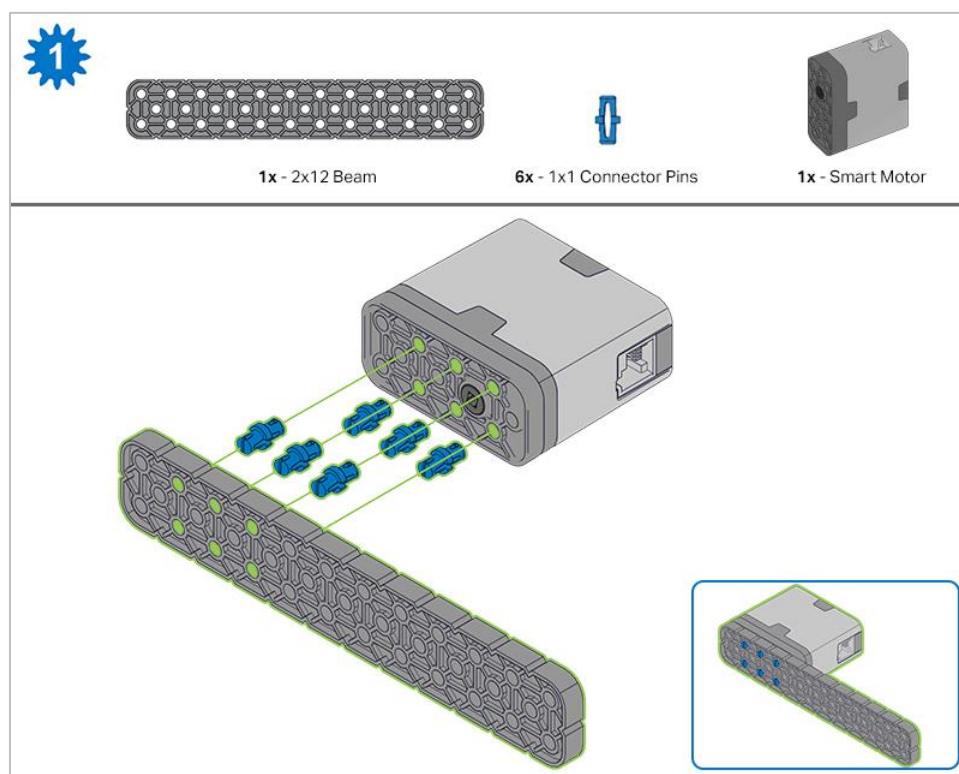
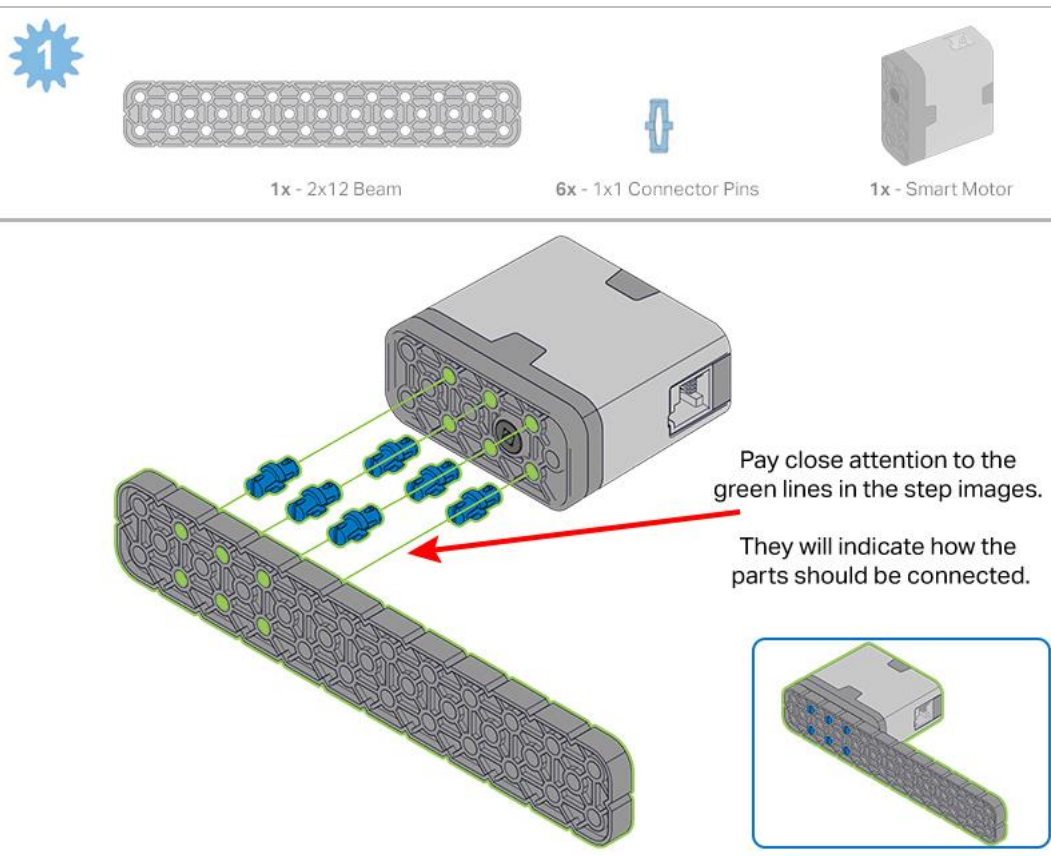
 **6x** - **1x1** Connector Pins

 **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



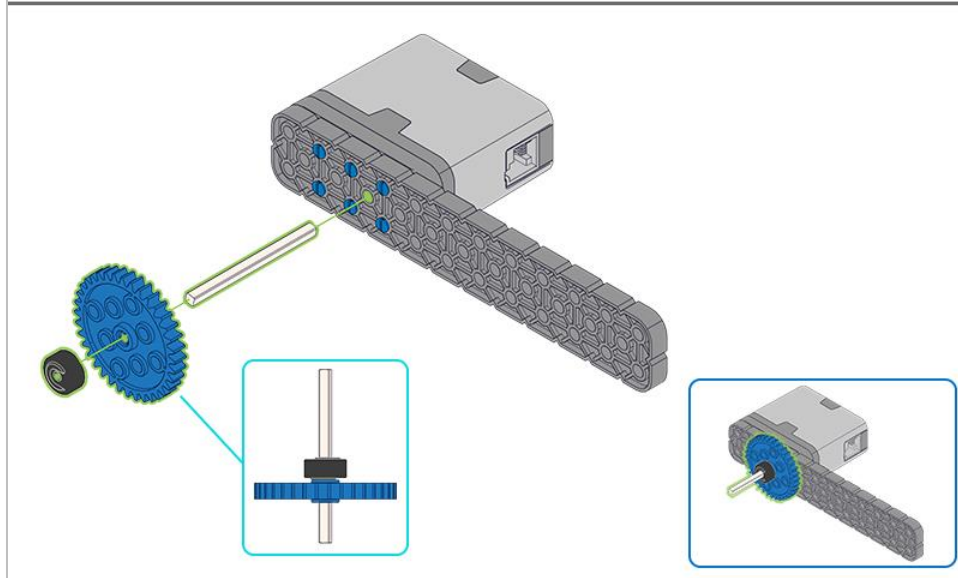
1x - 36 Tooth Gear



1x - Rubber Shaft Collar



1x - 4x Pitch Shaft

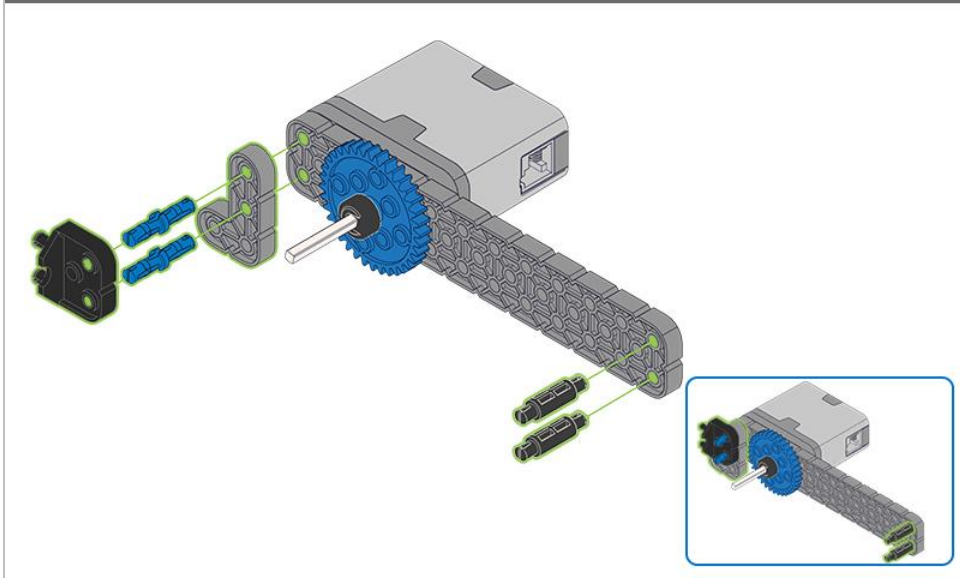


When adding the 4x Pitch Shaft, twist the pitch shaft to check for tension while turning. If it spins freely, it is not properly inserted into the motor.

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

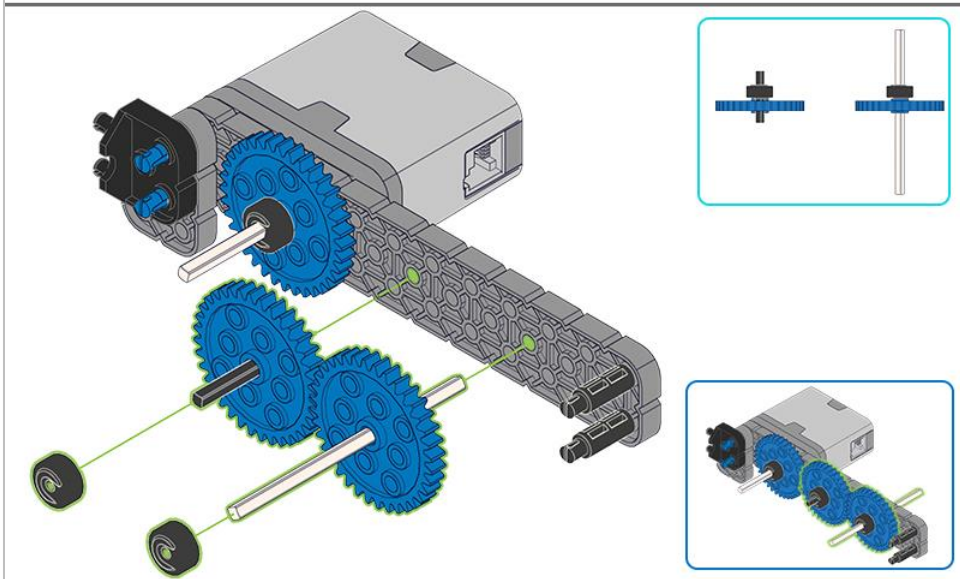


2x - Rubber Shaft Collars

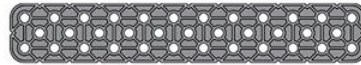
1x - 6x Pitch Shaft



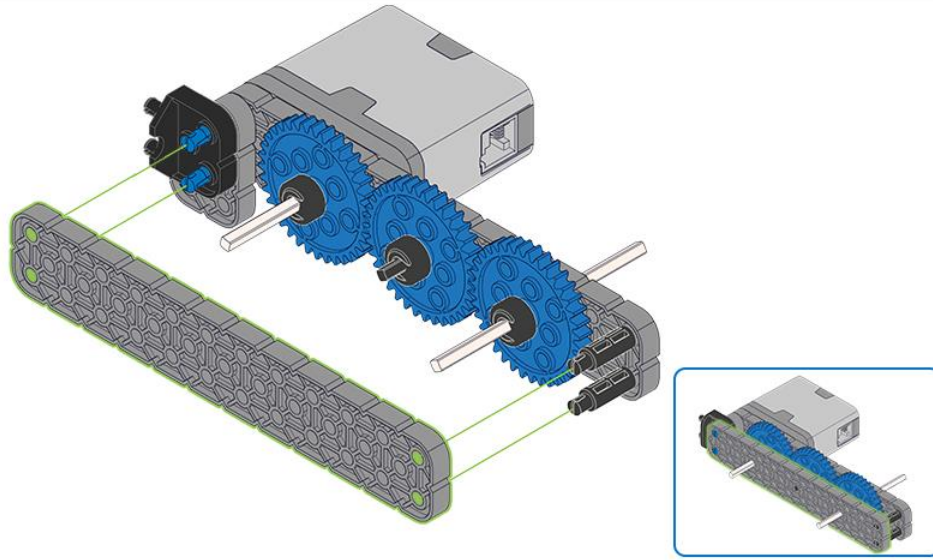
1x - 2x Plastic Pitch Shaft



5



1x - 2x12 Beam



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

6



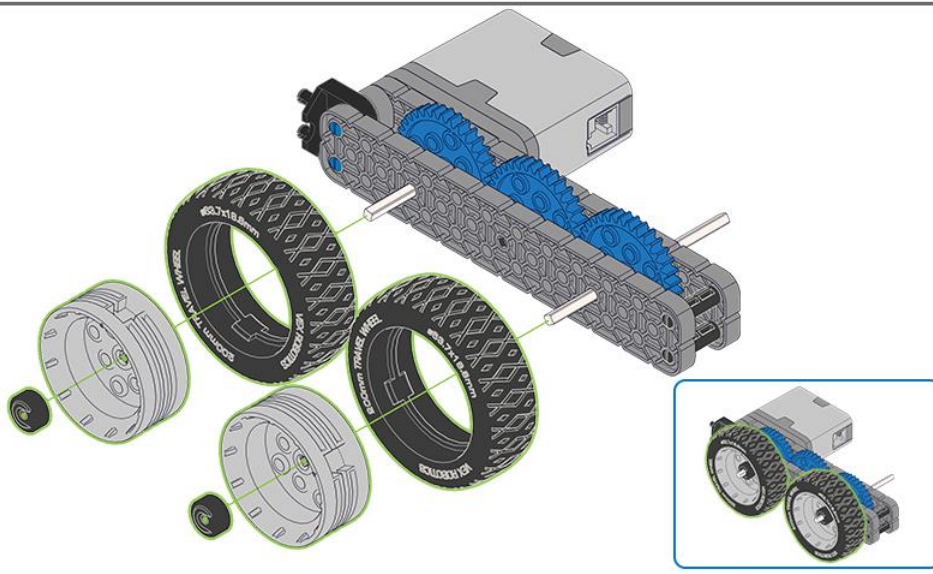
2x - Rubber Shaft Collars



2x - Wheel Hubs

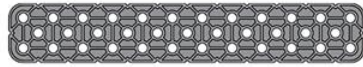


2x - Tires (200mm)



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.

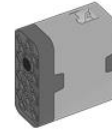
7



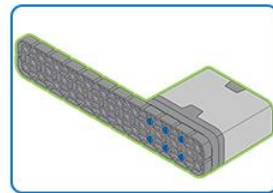
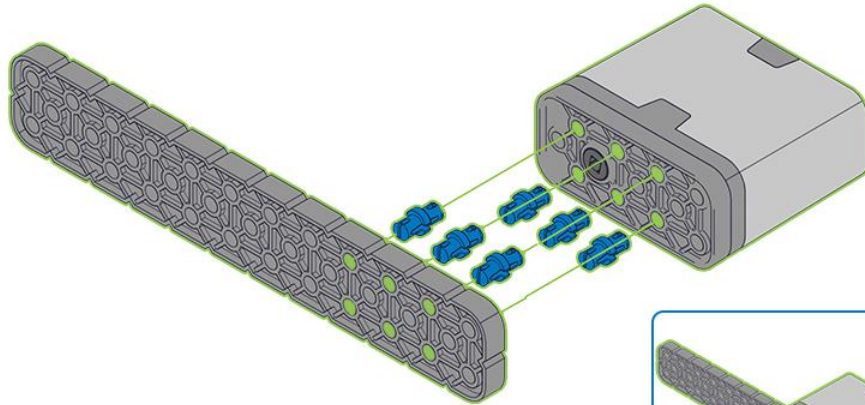
1x - 2x12 Beam



6x - 1x1 Connector Pins



1x - Smart Motor



8



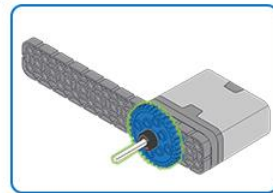
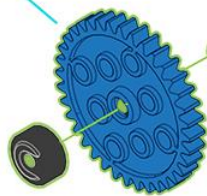
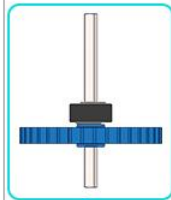
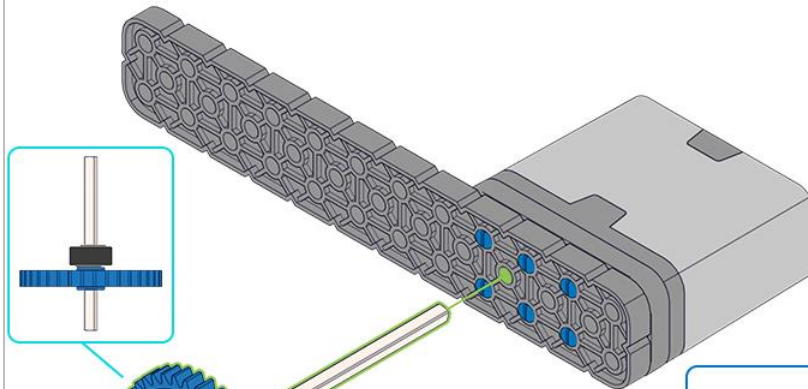
1x - 36 Tooth Gear



1x - Rubber Shaft Collar



1x - 4x Pitch Shaft

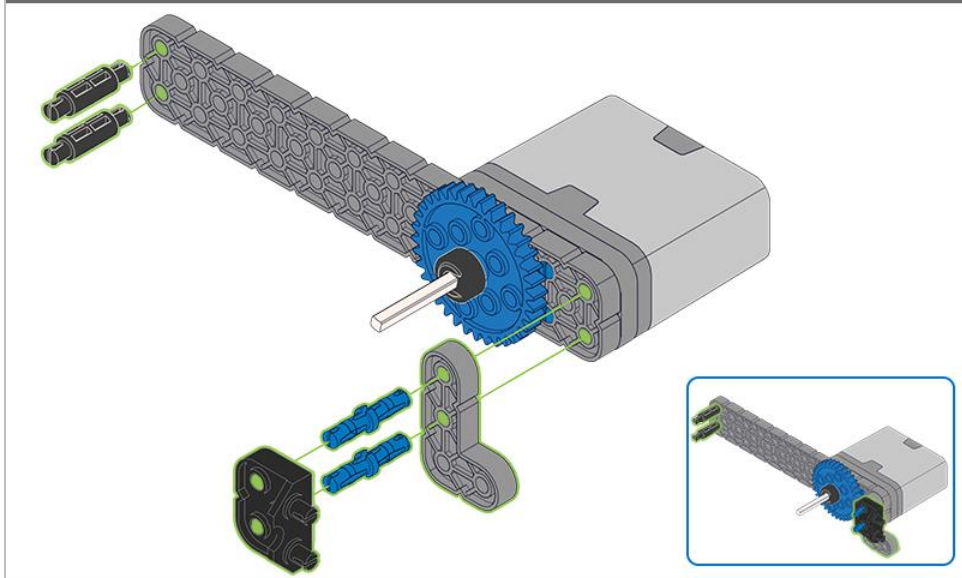


When adding the 4x Pitch Shaft, twist the pitch shaft to check for tension while turning. If it spins freely, it is not properly inserted into the motor.

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

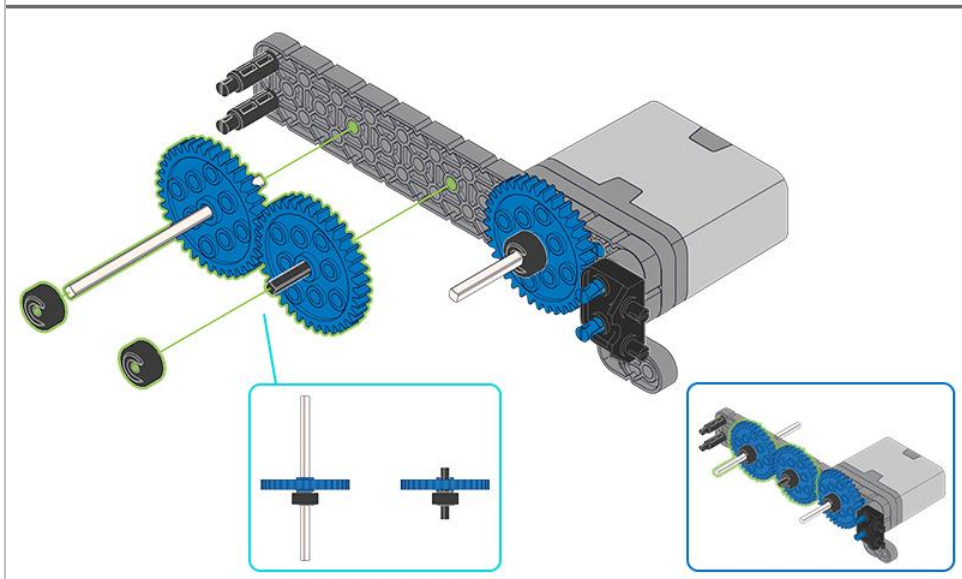


2x - Rubber Shaft Collars

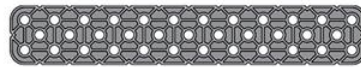
1x - 6x Pitch Shaft



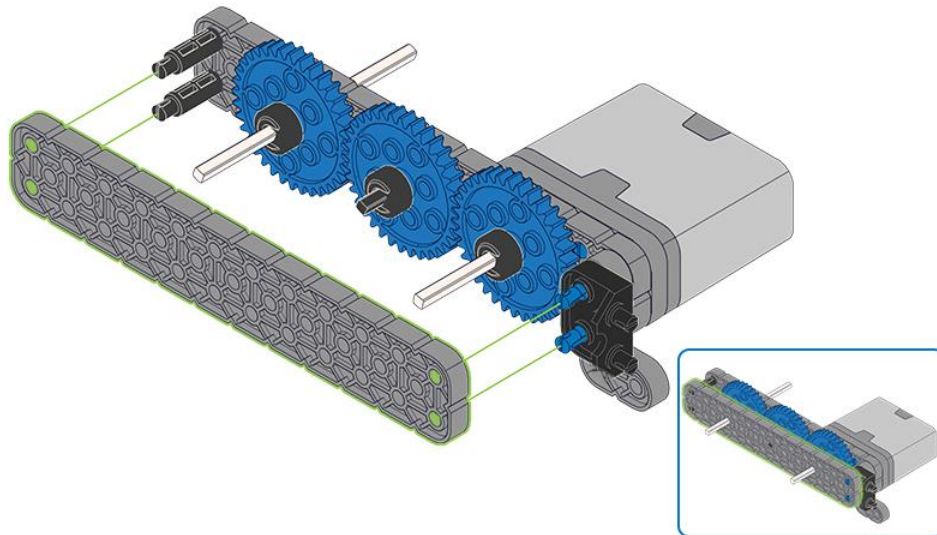
1x - 2x Plastic Pitch Shaft



11



1x - 2x12 Beam



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

12



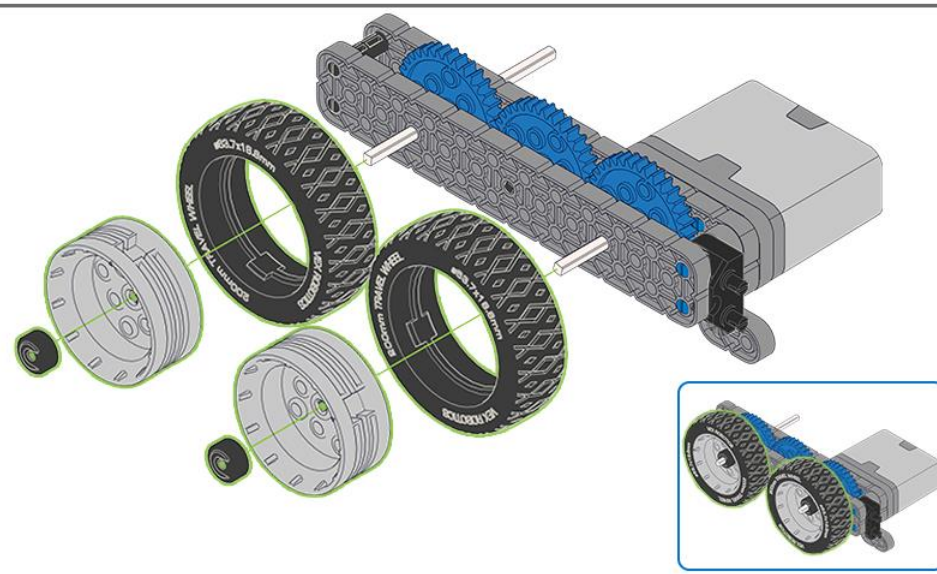
2x - Rubber Shaft Collars



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.

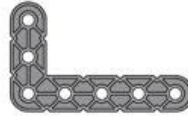
13



1x - 2x8 Beam



2x - 2x Wide, 1x2 Corner Connectors



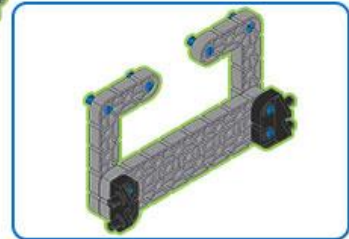
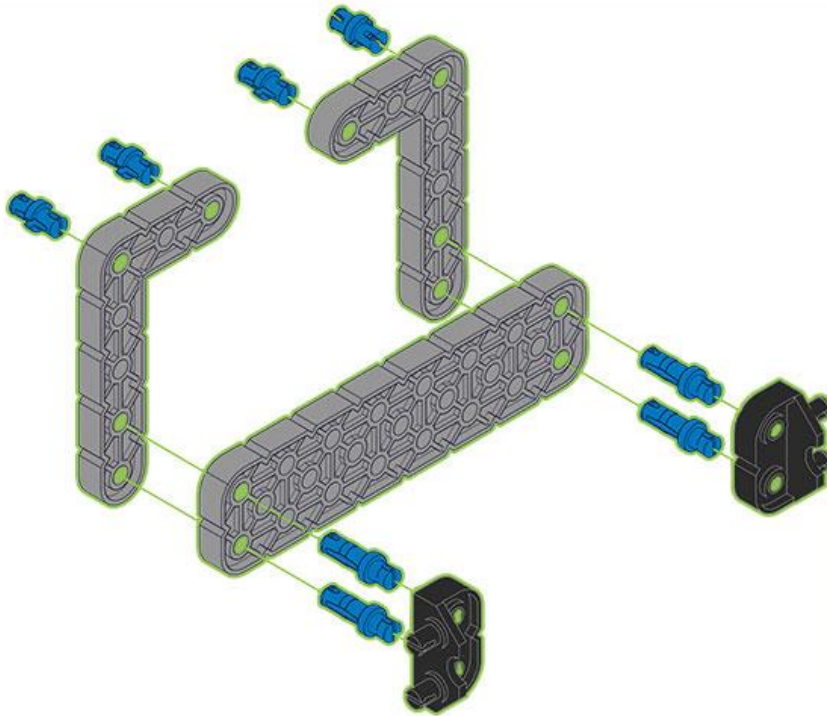
2x - 3x5 Right Angle Beams



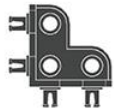
4x - 1x1 Connector Pins



4x - 1x2 Connector Pins



14

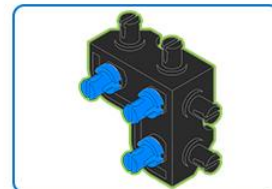
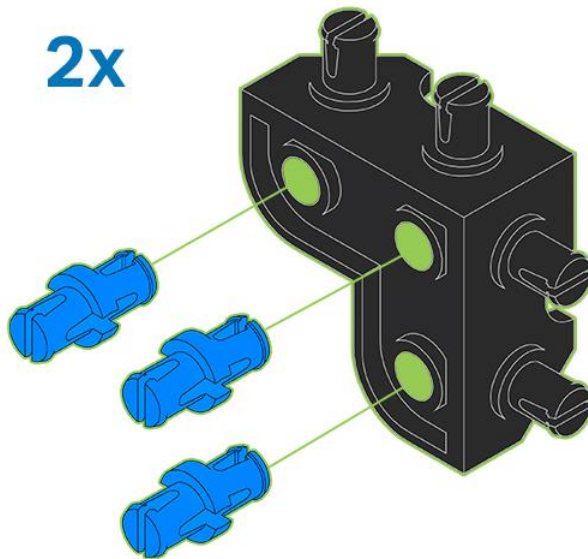


2x - Large Chassis Corner Connectors



6x - 1x1 Connector Pins

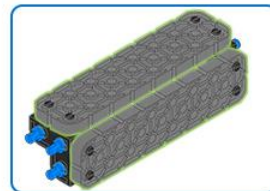
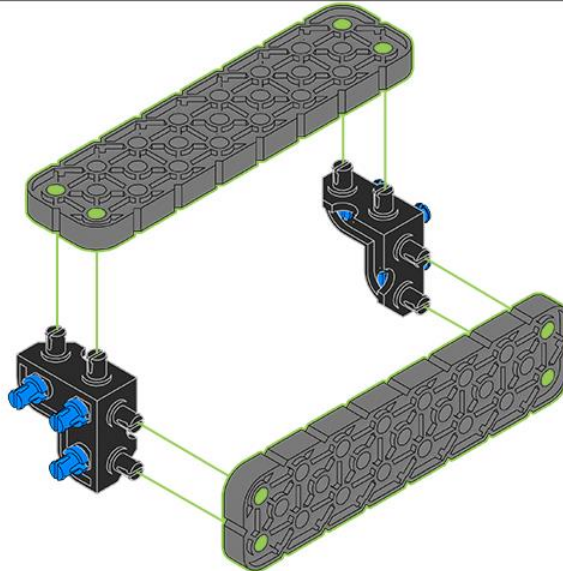
2x



15



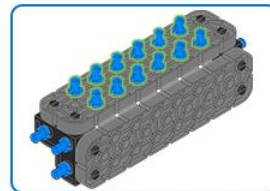
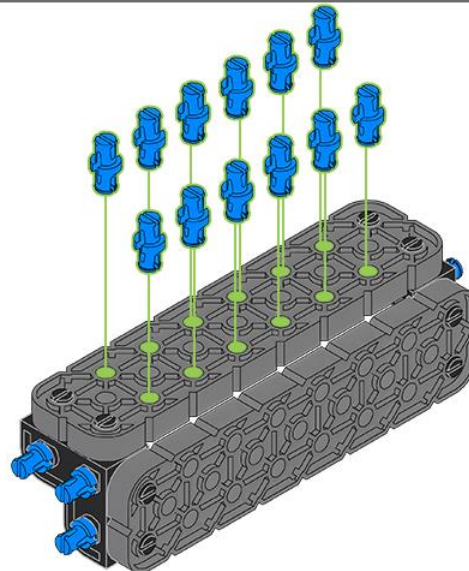
2x - 2x8 Beams



16



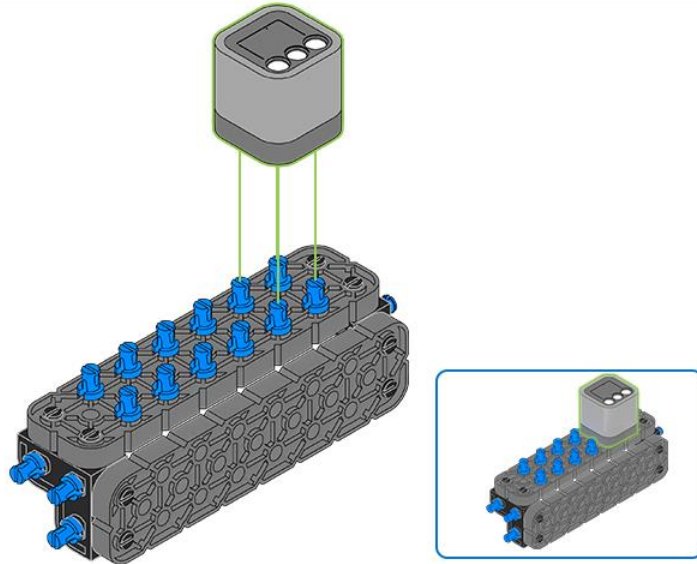
12x - 1x1 Connector Pins



17



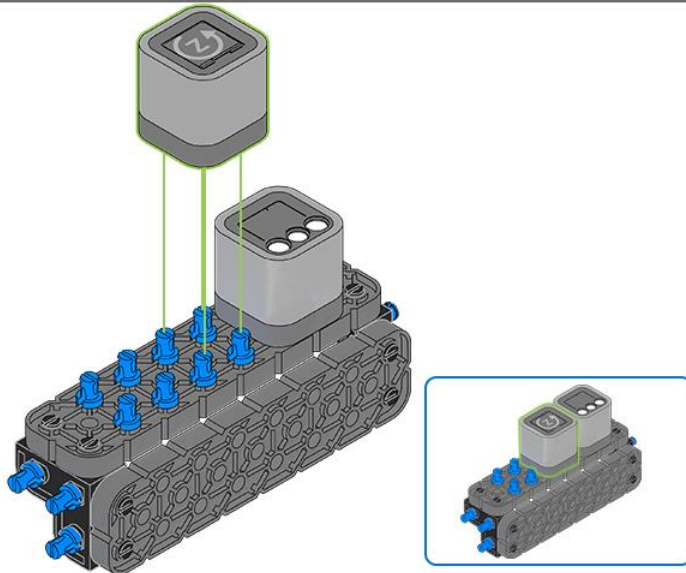
1x - Color Sensor



18



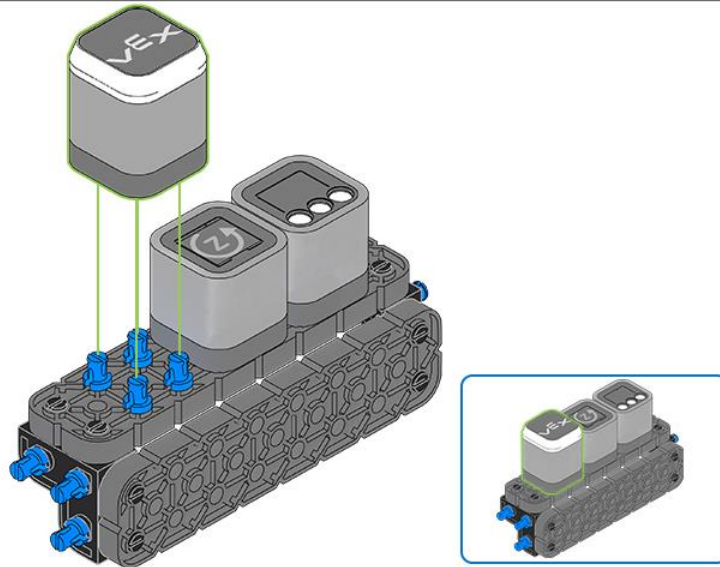
1x - Gyro Sensor



19



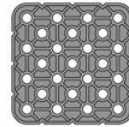
1x - Touch LED



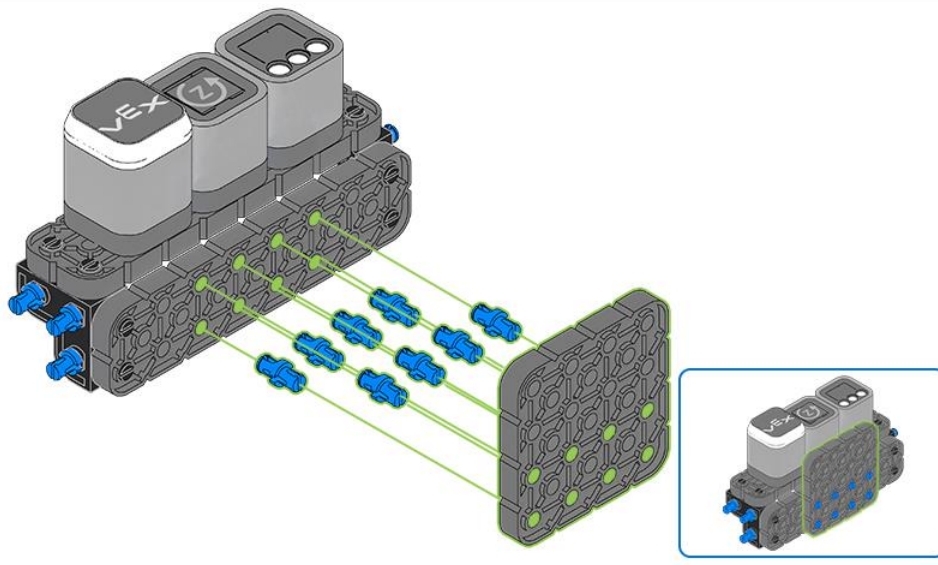
20



8x - 1x1 Connector Pins



1x - 4x4 Plate



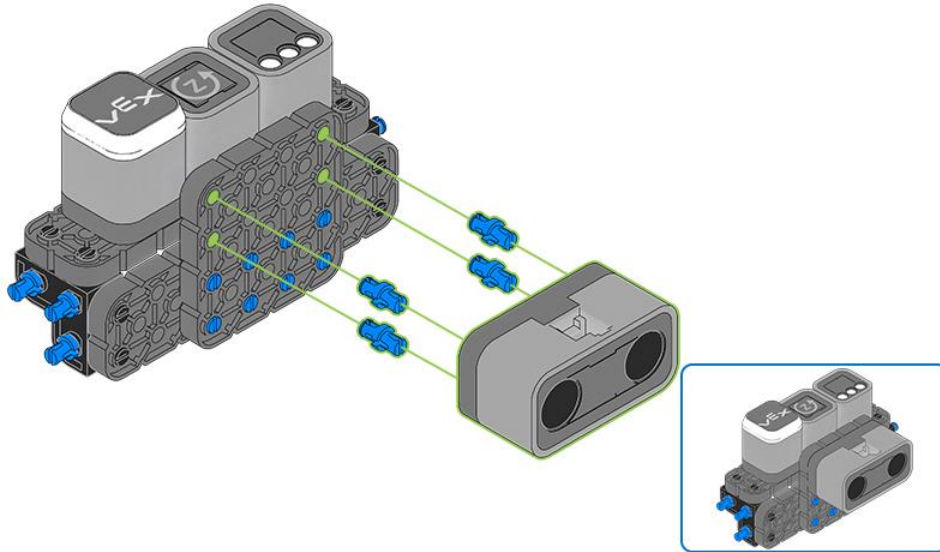
21



1x - Distance Sensor



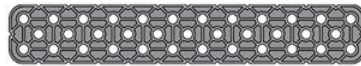
4x - 1x1 Connector Pins



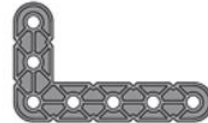
22



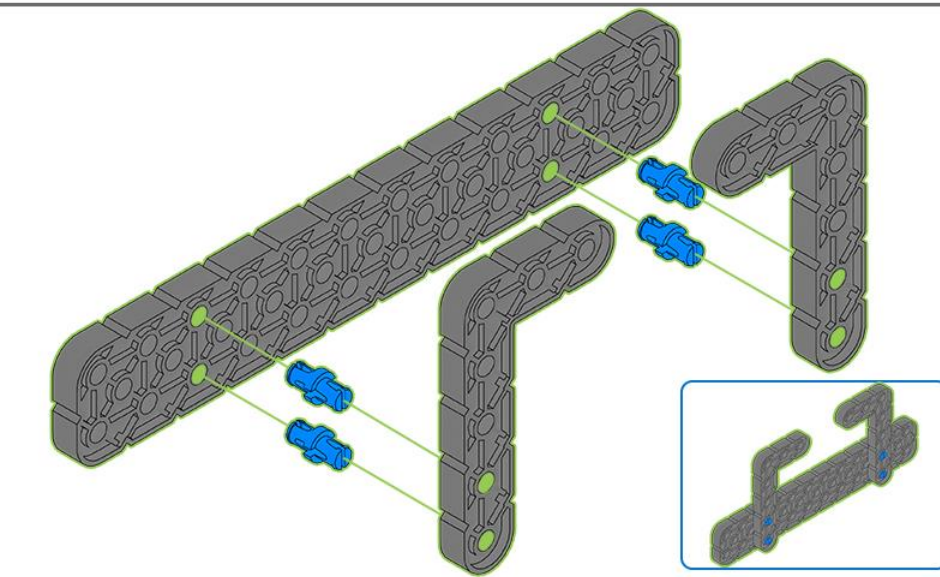
4x - 1x1 Connector Pins



1x - 2x12 Beam



2x - 3x5 Right Angle Beams



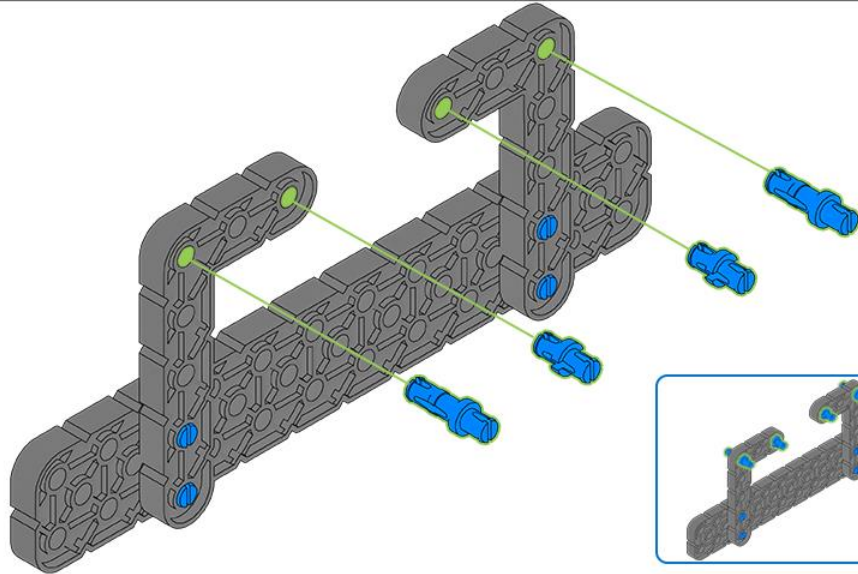
23



2x - 1x2 Connector Pins



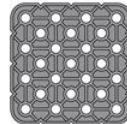
2x - 1x1 Connector Pins



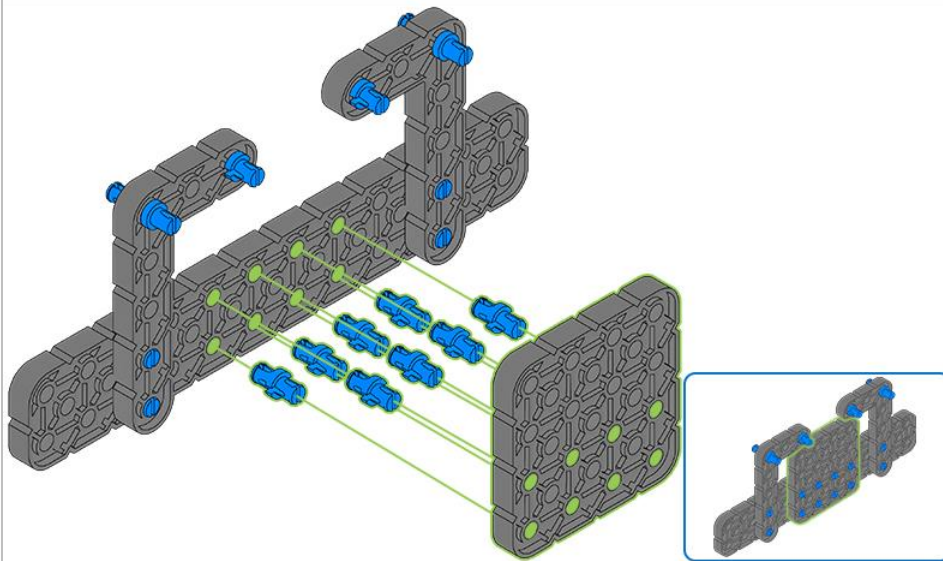
24



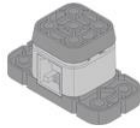
8x - 1x1 Connector Pins



1x - 4x4 Plate



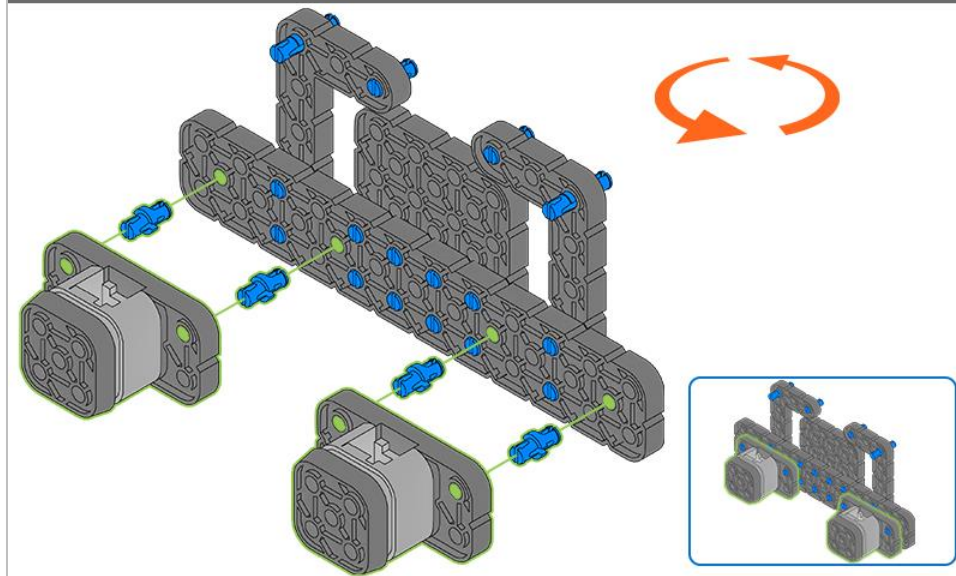
25



2x - Bumper Switch



4x - 1x1 Connector Pins



The orange arrows indicate to turn the assembly around.

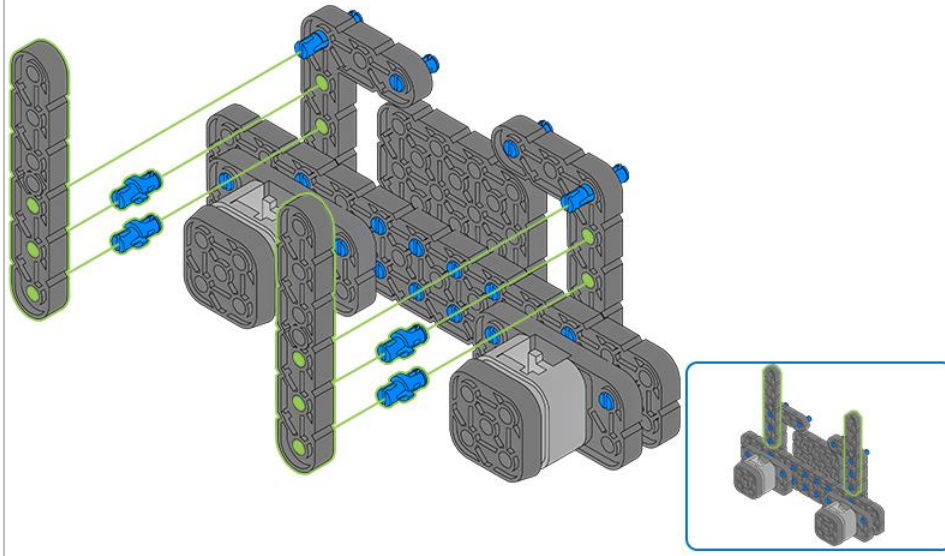
26



2x - 1x6 Beams



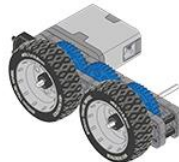
4x - 1x1 Connector Pins



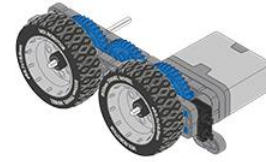
27



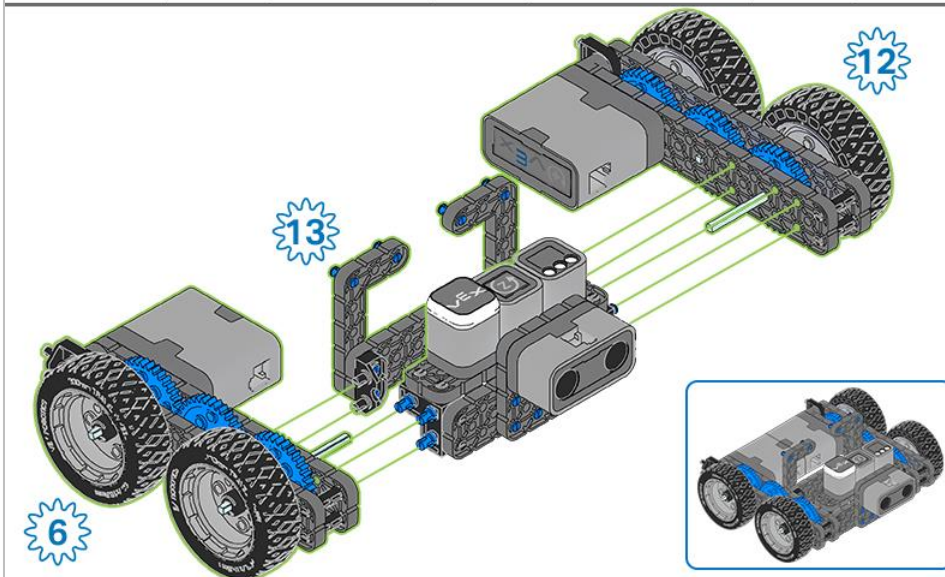
1x - Step 13 Assembly



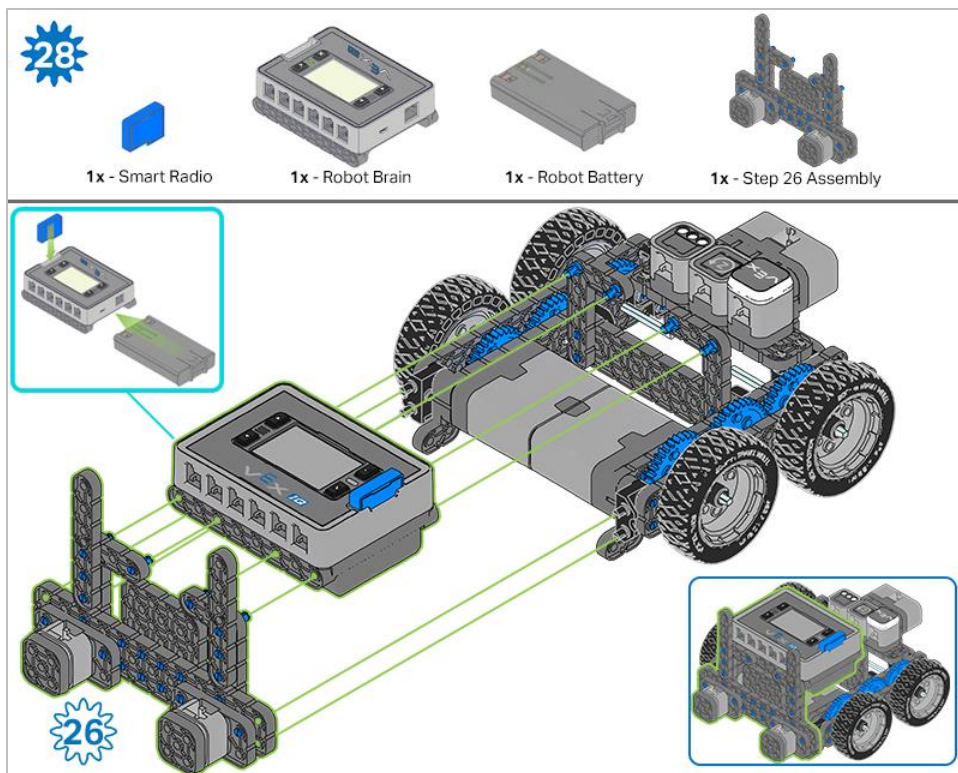
1x - Step 6 Assembly



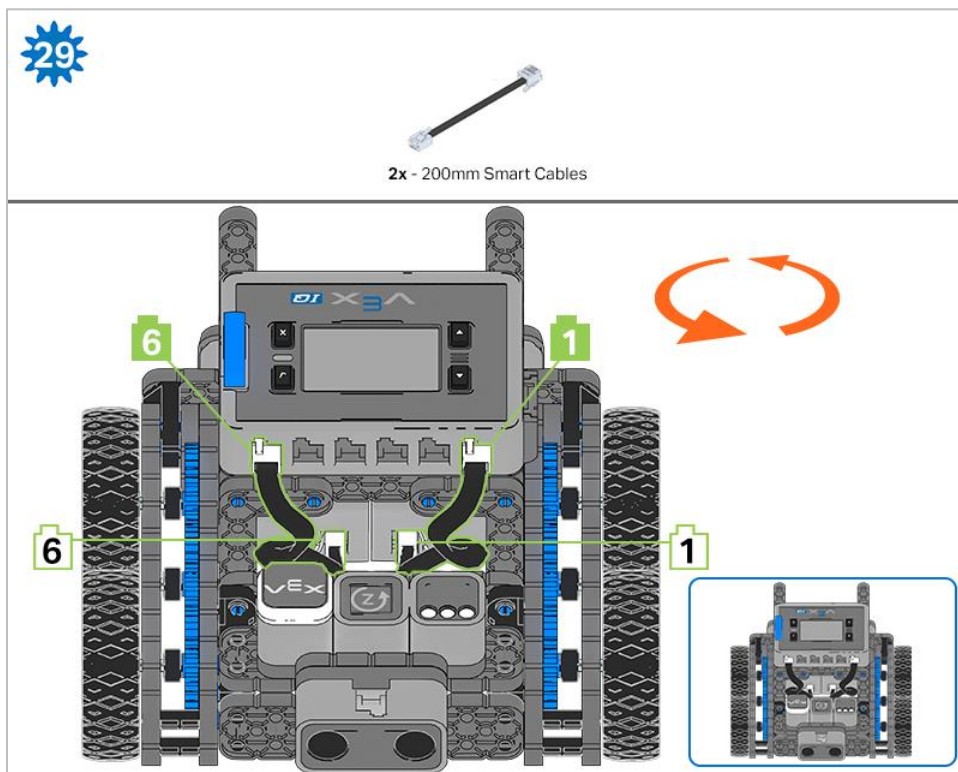
1x - Step 12 Assembly



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.



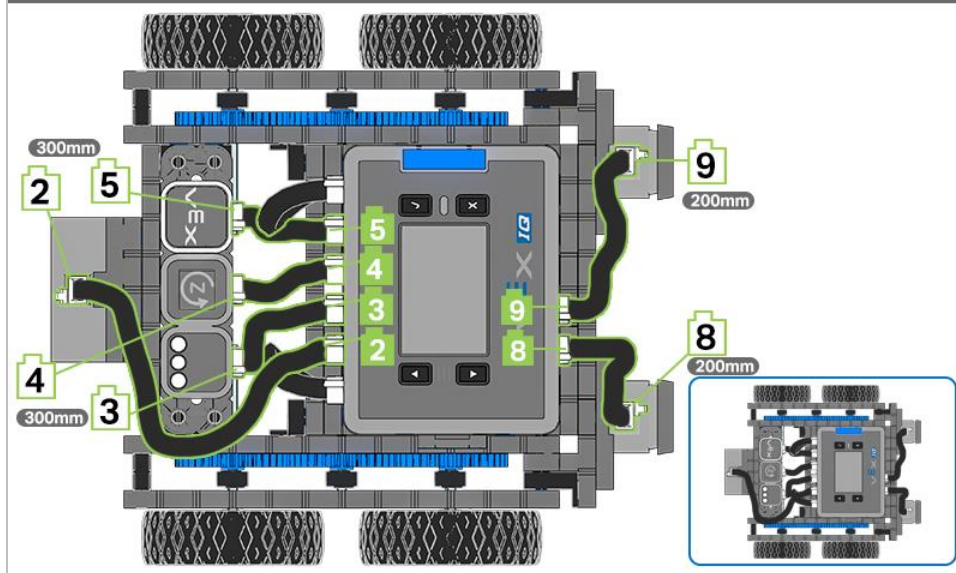
Steps 29-30, when you attach the Smart Cables, make sure they are tucked away so as to not block the Smart Sensors. The orange arrows indicate to turn the robot around.



4x - 300mm Smart Cables



2x - 200mm Smart Cables



Exploration

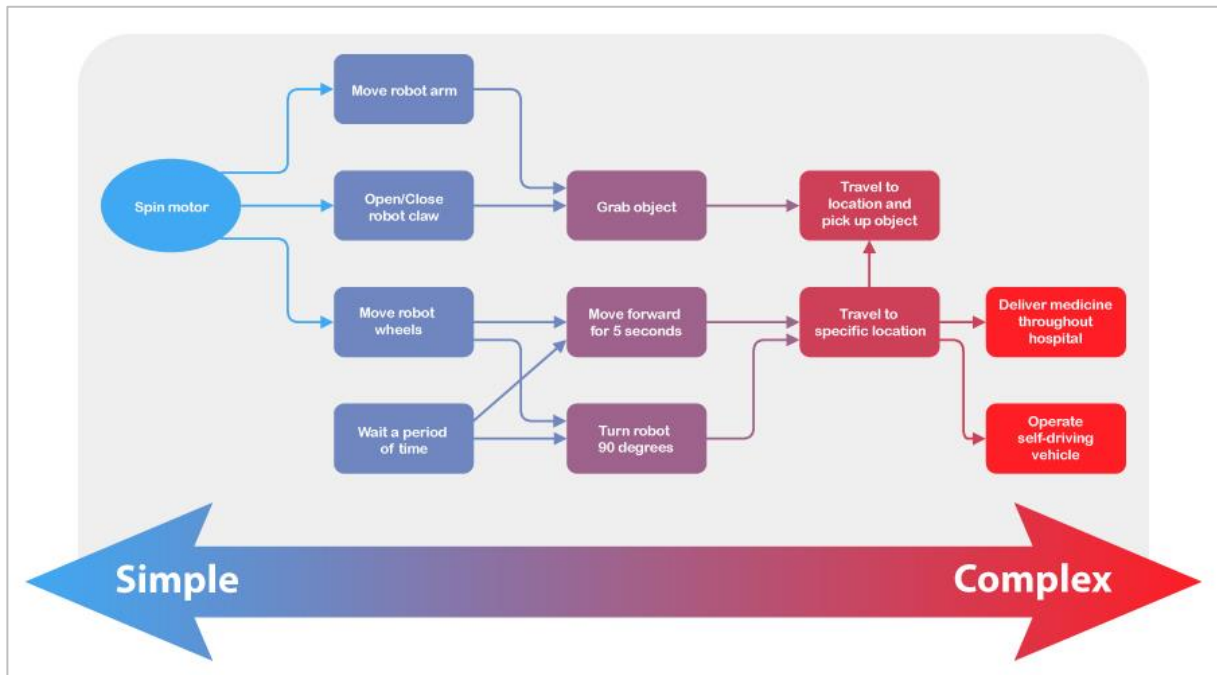
Now that the build is finished, explore and see what it can do. Then answer these questions in your engineering notebook.

- How do you think the Autopilot could be used as a tool for measurement?
- If you didn't have a ruler to measure how far the Autopilot robot moved, which VEX piece would you choose to act as a measuring stick? Why?
- Looking at the Autopilot robot, how many pieces do you think you used to build it? Record your guess and why you chose that number in your engineering notebook.



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

Introduction to Robot Behaviors



Programming Complexity

Robots can be designed to perform many different types of tasks. Some of these tasks are very simple, like opening an automatic door. Others are not so simple, like a robot moving around a warehouse to collect items for an order. No matter how complicated the task is, it can be broken down into simpler ones. These tasks are known as behaviors and are the building blocks of robotics programming.

A behavior is a way that a robot acts, and can range from basic to complex, depending on how the robot is built or programmed. A simple mobile robot like the Autopilot only has two motors, so its behaviors will involve turning those motors to accomplish its goals. With more design and coding, you can start from this basic behavior and do more complex behaviors.

Drive Forward and Reverse

Exploration - Part 1

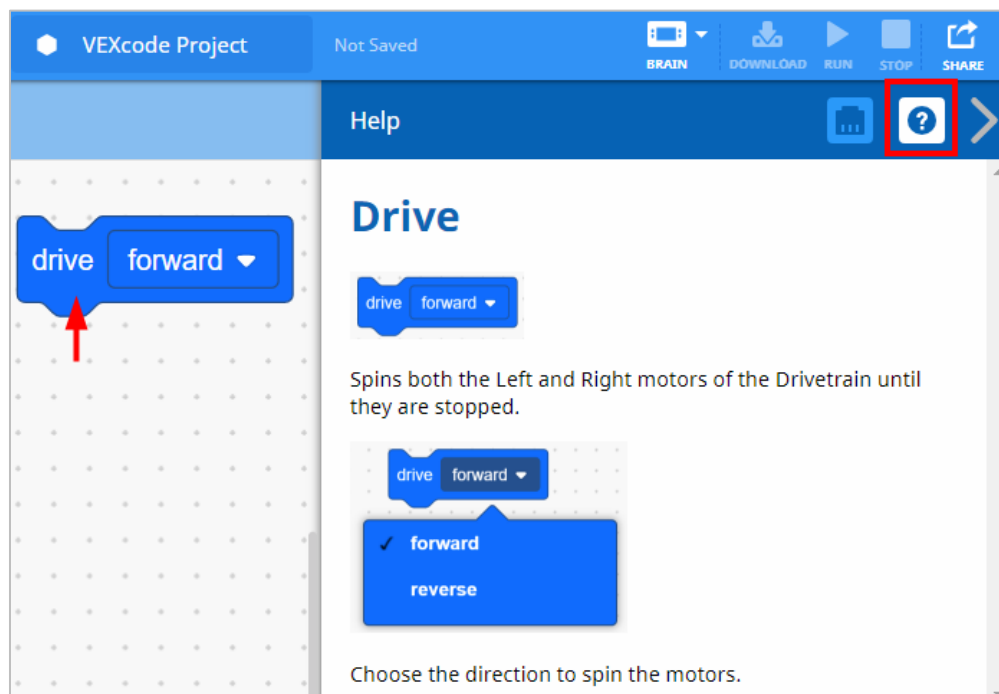
Autopilot is ready to move!

This exploration will give you the tools to be able to start creating some cool projects for your Autopilot to follow.

- VEXcode IQ Blocks that will be used in this exploration:



- To find out more information about the block, open the Help and then select the drive block.



- 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:

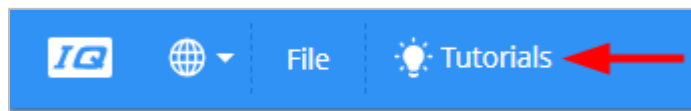
Quantity	Hardware/Other Items
1	Autopilot Robot
1	Charged Robot Battery
1	VEXCode IQ Blocks
1	Engineering Notebook
1	USB Cable (if using a computer)

1. Preparing for the Exploration

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

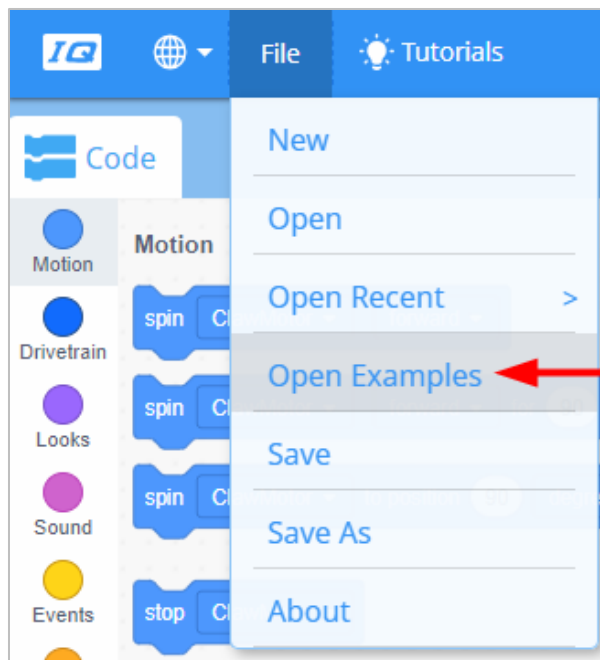
- Are all the motors and sensors plugged into the correct port?
- Are the smart cables fully inserted into all of the motors and sensors?
- Is the Brain turned on?
- Is the battery charged?

Check out the troubleshooting tutorial if Autopilot does not pass any of these steps.



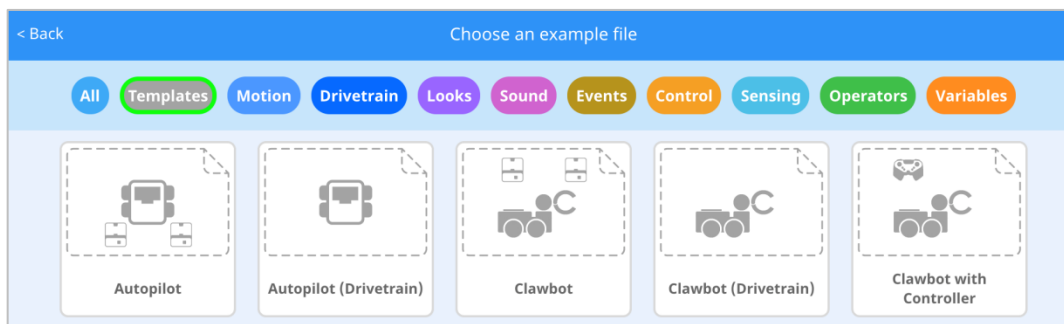
2. Start a New Project

Before you begin your project, the Programmer needs to select the correct template project. The Autopilot (Drivetrain) template contains the Autopilot motors and sensors configuration. If the template is not used, your robot will not run the project correctly.

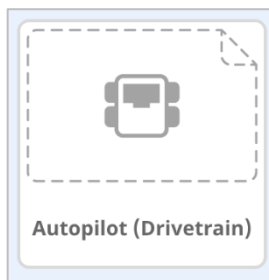


The Programmer should complete the following steps:

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



- Select and open the **Autopilot (Drivetrain) Template**.



- Since we will be using the drive block, rename your project Drive.
- **Save** your project.
- Check to make sure the project name Drive is now in the window in the center of the toolbar.



Drive

Saved

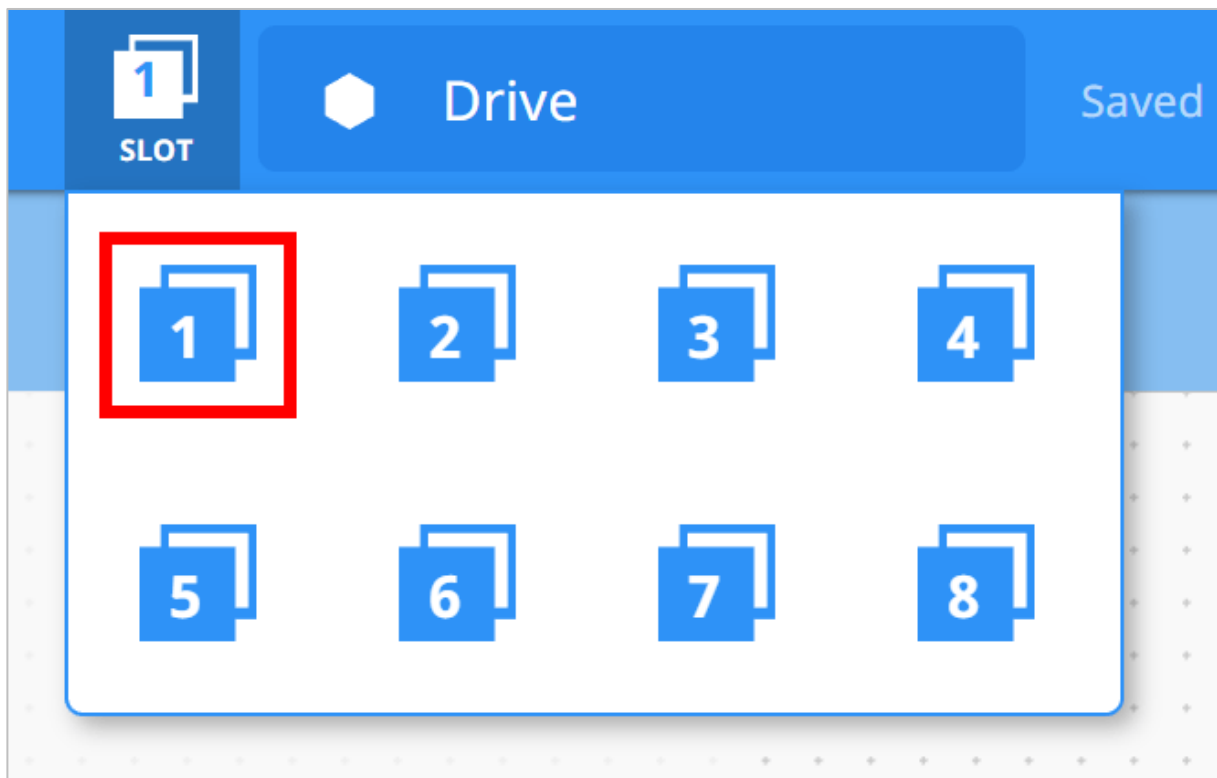
Drive Forward and Reverse

Exploration - Part 2

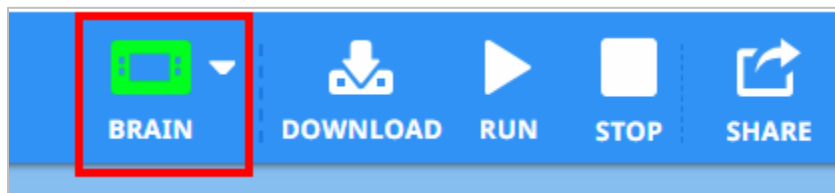
1. Drive Forward



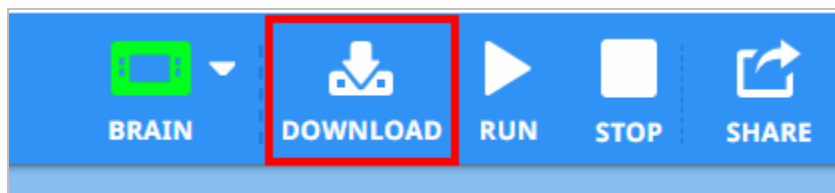
- Add the drive block to the when started block in the programming area.



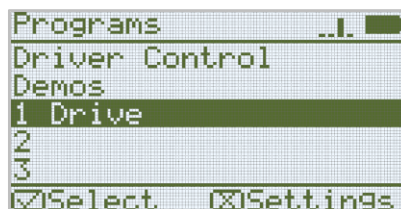
- Click on the **Slot** icon. You can download your project to one of the four available slots in the Robot Brain. Click on the number 1.



- The Programmer should connect the robot to your computer or tablet. The Brain icon in the toolbar turns green once a successful connection has been made.



- Click the Download button on the toolbar to download the Drive project to the Robot Brain.



- The Operator should check to make sure your project has downloaded to the Autopilot's Brain by looking at the Robot Brain's screen. The project name should be listed in slot 1.

- The Operator should now run the project on the Autopilot robot by making sure the project is highlighted and then press the Check button. Congratulations on creating your first program!

2. Drive Reverse



- Go back to VEXCode IQ Blocks programming area. The Programmer should change the drive block to display **reverse** instead of **forward**.
- The Programmer should download the project.
- The Operator should now run the project on the Autopilot robot by making sure the project is highlighted and then press the Check button.



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

Robotic Movements



A Dragon Runner Robot moves through the desert.

Photo: Steve Dock/MOD [OGL v1.0 (<http://NationalArchives.gov.uk/doc/open-government-licence/version/1/>)]

Robotic Movements

Today's robots have the ability to complete many complicated tasks, such as the Dragon Runner Robot. This robot uses many operations and movements to find and remove dangerous items to keep people safe.

Robots didn't start out being able to carry out complicated programs. The very first robot, the Unimate, was created by General Motors to work in a factory in 1961. It was programmed to complete very simple movements, like moving forward and backward, in order to move pieces of hot metal on an assembly line.

Over time, engineers have developed robots to be able 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.

Let's Compete!



Students competing in the Teamwork Challenge

The VEX IQ Challenge

Each year, teams can design and build a robot to play against other teams in a game-based VEX IQ Challenge. The game changes each year. [Click here](#) for this year's current game and rules. 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 and grab, toss, and place game pieces in scoring zones in order to earn the most points.

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.

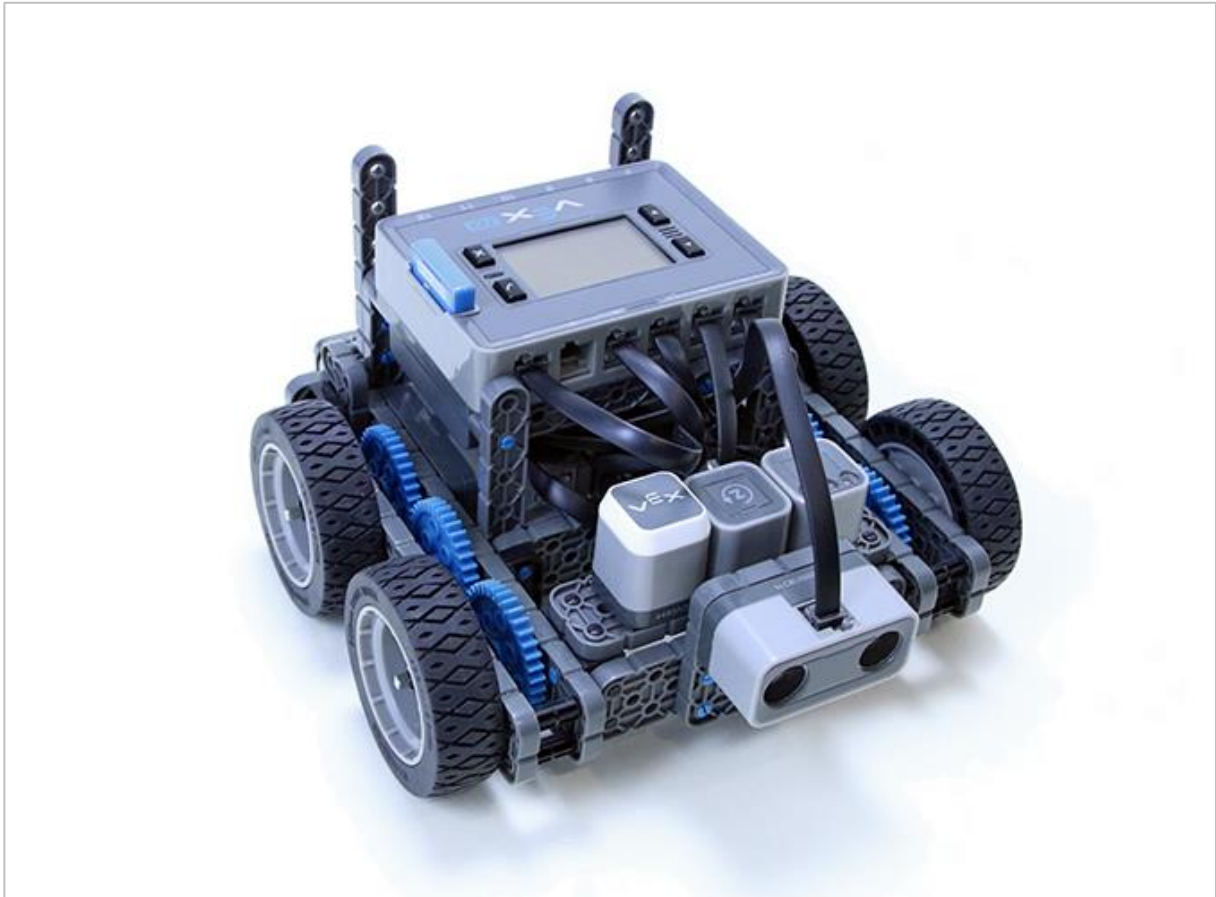
VEX Competitions give students the opportunity to:

- 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



Exploring Another Block

Hardware/Software Required:

Quantity	Hardware/Other Items
1	Autopilot Robot
1	Charged Robot Battery
1	VEXCode IQ Blocks
1	Engineering Notebook

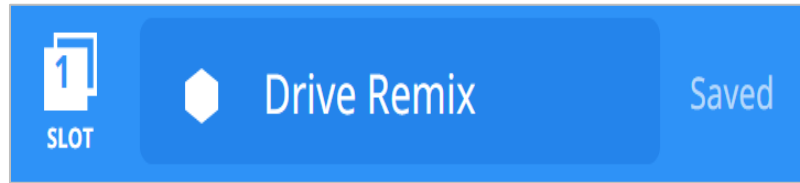
Quantity	Hardware/Other Items
1	USB Cable (if using a computer)

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 fully charged?

Remix Your Project



The Programmer should complete the following steps:

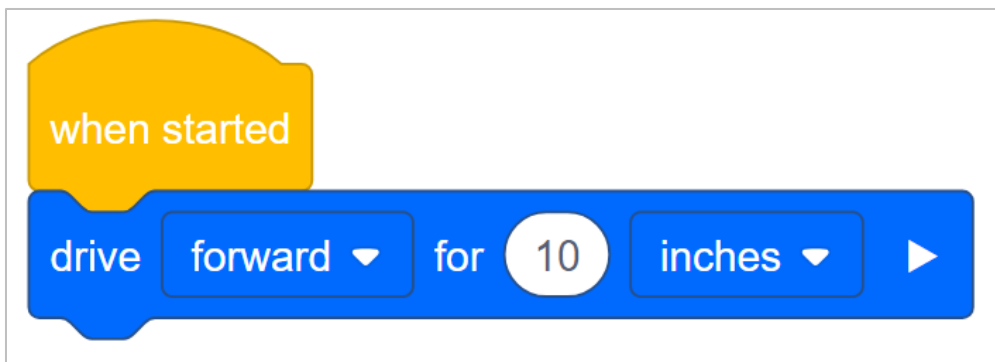
- Open the File menu.
- Select Open **Examples**.
- Select and open the **Autopilot (Drivetrain) Template**.
- Name your project **Drive Remix**.
- **Save** your project.



Let's get moving!

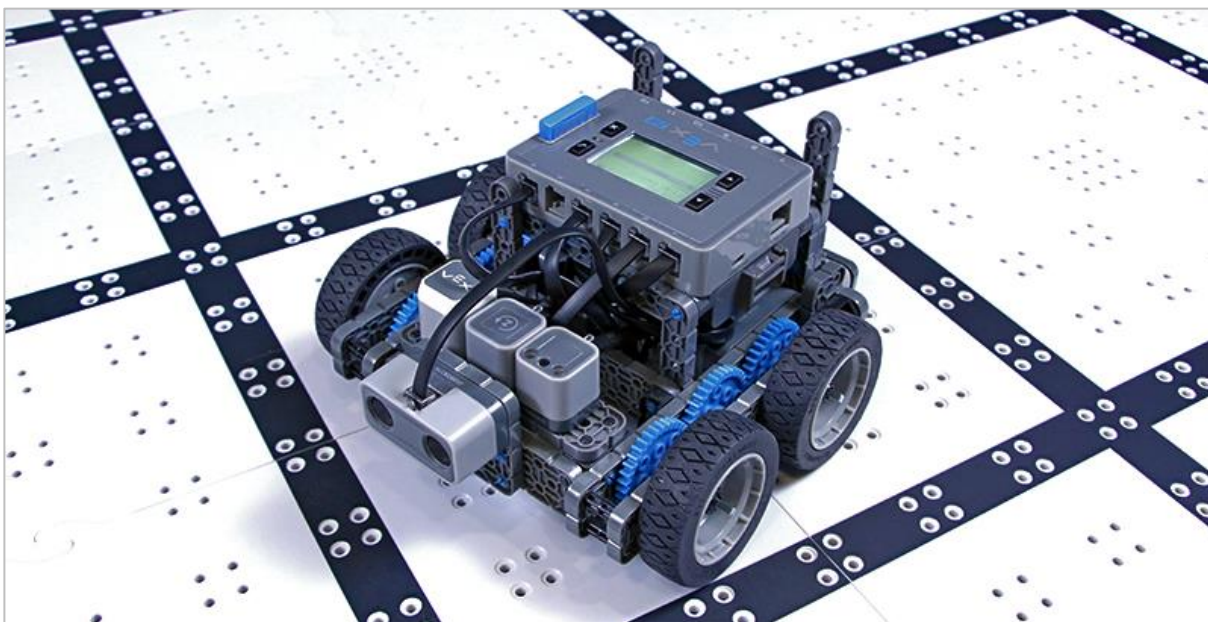
Activity A: Move 10 Inches!

- Programmer, use the `drive for` block to program Autopilot to move forward 10 inches.



- Download the project.
- Operator, run the project on Autopilot.
- Does the robot move forward 10 inches?

Bonus Challenge: Add a sound after Autopilot moves 10 inches! In the engineering notebook, the Recorder can write down what block the group thinks would allow the robot to perform this action.



Activity B: More fun with the drive for block!

Here are some more challenges:

- Drive in reverse 20 inches.
- Drive forward 100 millimeters.
- Drive in reverse 150 millimeters.

Bonus Challenge: Program Autopilot to move forward 10 inches and then backwards 5 inches.

Remix Questions

Answer the following questions in your engineering notebook after completing Activity A and B.

- Do you think that the Autopilot will always move the exact distance written in the project? Why or why not?
- If your teacher asks you to move the Autopilot 10 mm, but you can only program it to move the distance in inches, what could you do?



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

Review

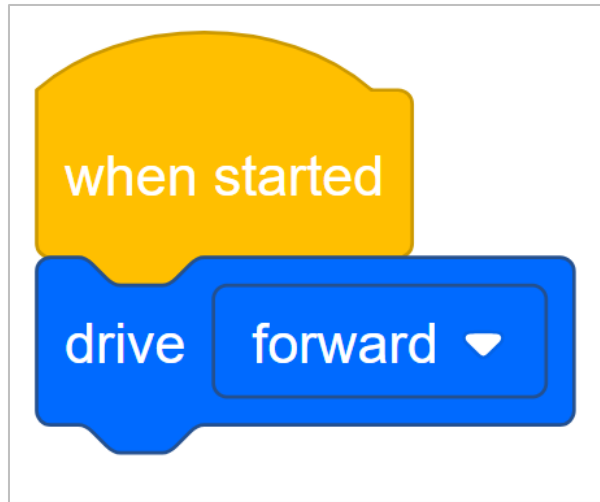
You have accomplished a lot in this STEM Lab! The following questions will help you think back over everything you have learned. You can only answer once, so think carefully before you submit!

- 1. Shawna estimated that she used 30 pieces to build her Autopilot robot. How would you respond to this estimation?**
 - ☐ This estimation is too high, and Shawna could count the pieces to confirm.
 - ☐ Shawna's estimation is fine, because estimations are only random guesses.
 - ☐ This estimation is too low, and Shawna could find a systematic way to estimate the pieces to revise her estimate.

- 2. Troubleshooting is a systematic way to solve a problem by investigating the error and trying possible solutions. Which of the following scenarios would require troubleshooting?**
 - ☐ The project won't download to the Robot Brain.
 - ☐ The project runs but does not move the robot.
 - ☐ The robot does not move the distance you expected it to.
 - ☐ All of these answers are correct.

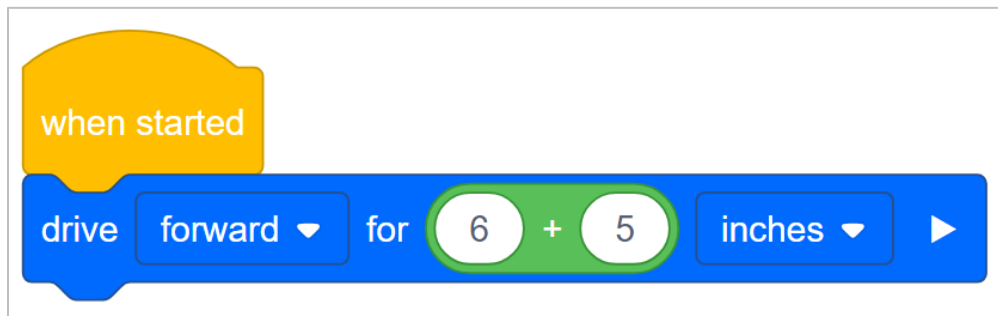
- 3. True or False: The robot can be programmed to travel in both inches and millimeters.**
 - ☐ True
 - ☐ False

- 4. In this example program, the robot will continuously move forward. How could you make the robot stop moving forward?**



- Add a command to only move forward a certain number of inches.
- Add a command to only move forward a certain amount of time.
- Press the X button on the robot brain while the program is running.
- All of these answers are correct.

5. In this example program, the robot will:



- Move forward 6 inches, wait and then move 5 inches.
- Move forward 11 inches.
- Move forward 1 inch.
- Move forward 11 mm.

6. Aaron wants to move his robot forward 40 inches. How could he write the distance for the block?

- $10 + 30$ inches
- $4 * 10$ inches
- 1016 mm
- All of these answers are correct.

7. The autopilot robot is programmed to move $(5+13)$ inches forward, and $(3+8)$ inches in reverse. How many inches will the robot be from the starting position once the entire program is complete?
- ☐ 7 inches
 - ☐ -7 inches
 - ☐ 0 inches
 - ☐ 11 inches
8. Elise wants to send her Autopilot robot to her friend. The edge of her friend's desk is 64 inches away. The program will move the Autopilot forward 12 inches. How many times would Elise need to run the program to move the Autopilot past the edge of her friend's desk?
- ☐ 4 times
 - ☐ 5 times
 - ☐ 6 times
 - ☐ 7 times

APPENDIX

Additional information, resources, and materials.

Knowledge Base Articles

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

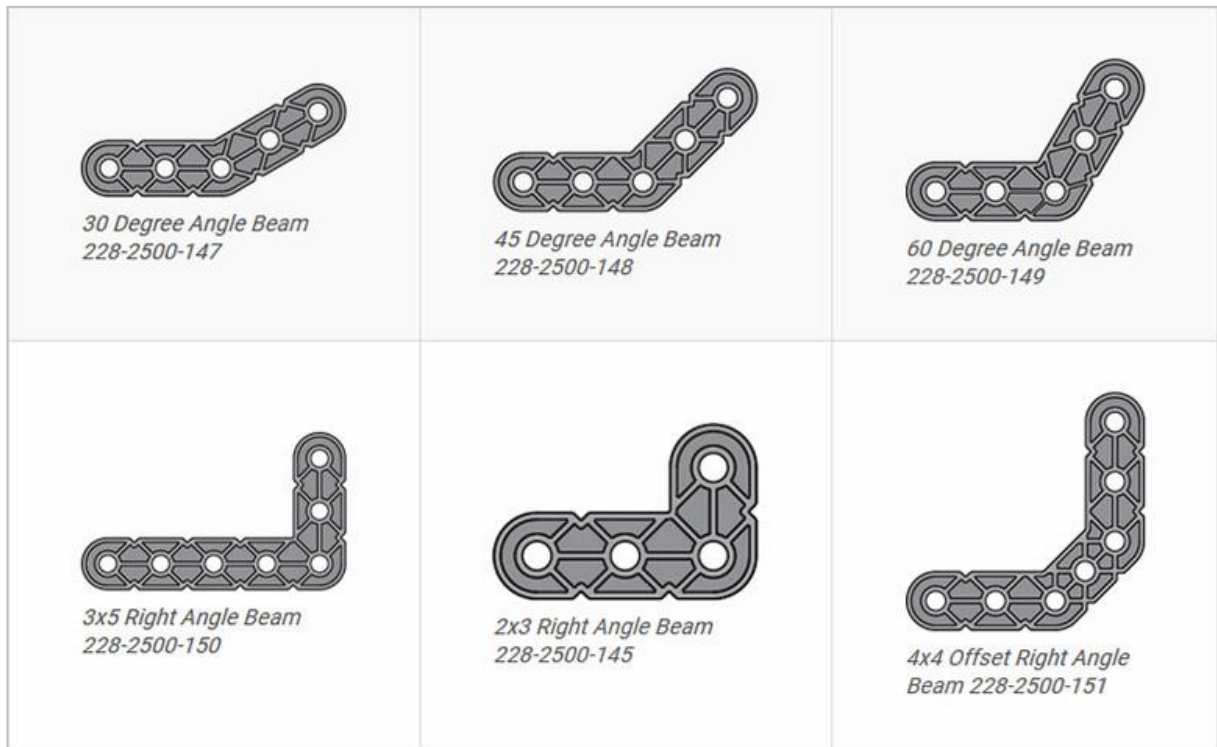
- How to Turn On/Off a VEX IQ Robot Brain
<https://kb.vex.com/hc/en-us/articles/360035952571-How-to-Turn-On-Off-a-VEX-IQ-Robot-Brain>
- How to Read Indicator Lights on the VEX IQ Robot Brain
<https://kb.vex.com/hc/en-us/articles/360035590672-How-to-Read-Indicator-Lights-on-the-VEX-IQ-Robot-Brain>
- How 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>

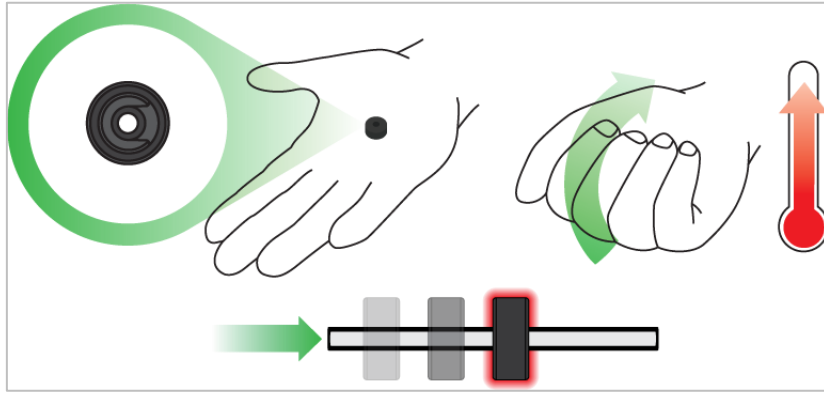
Identifying Angle Beams



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

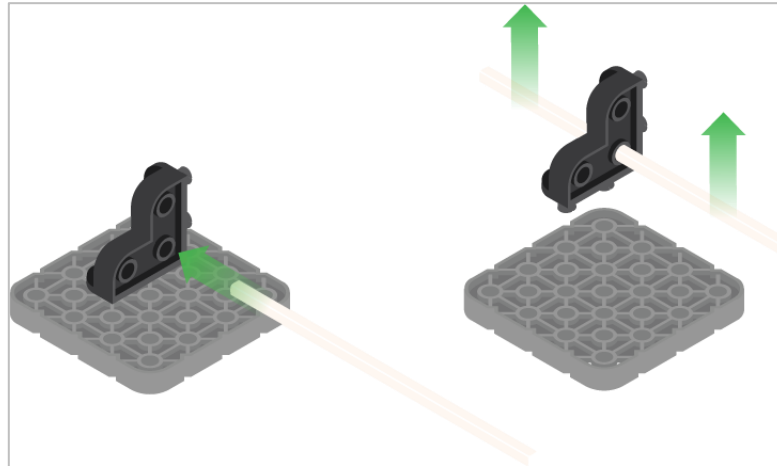


Using your hand to warm a Rubber Shaft Collar

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

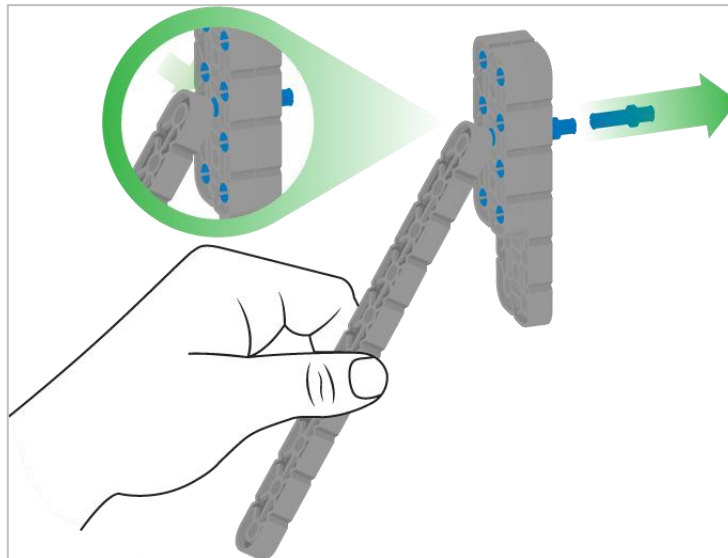


Using a pitch shaft to remove a corner connector

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

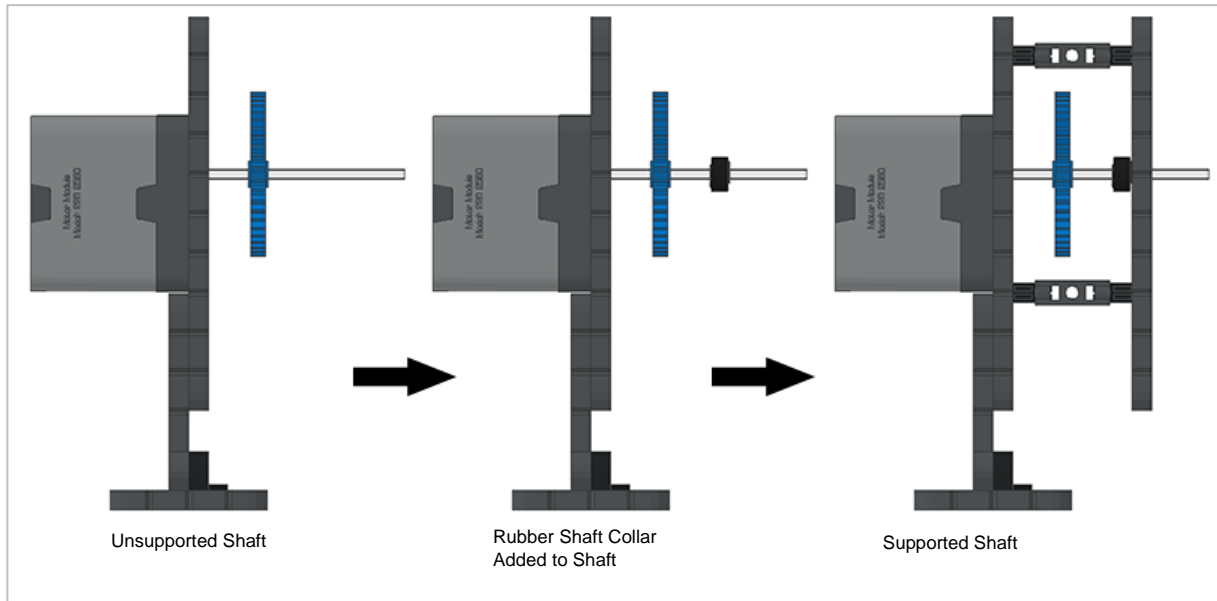


Removal of a standoff from a Mini Standoff Connector

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

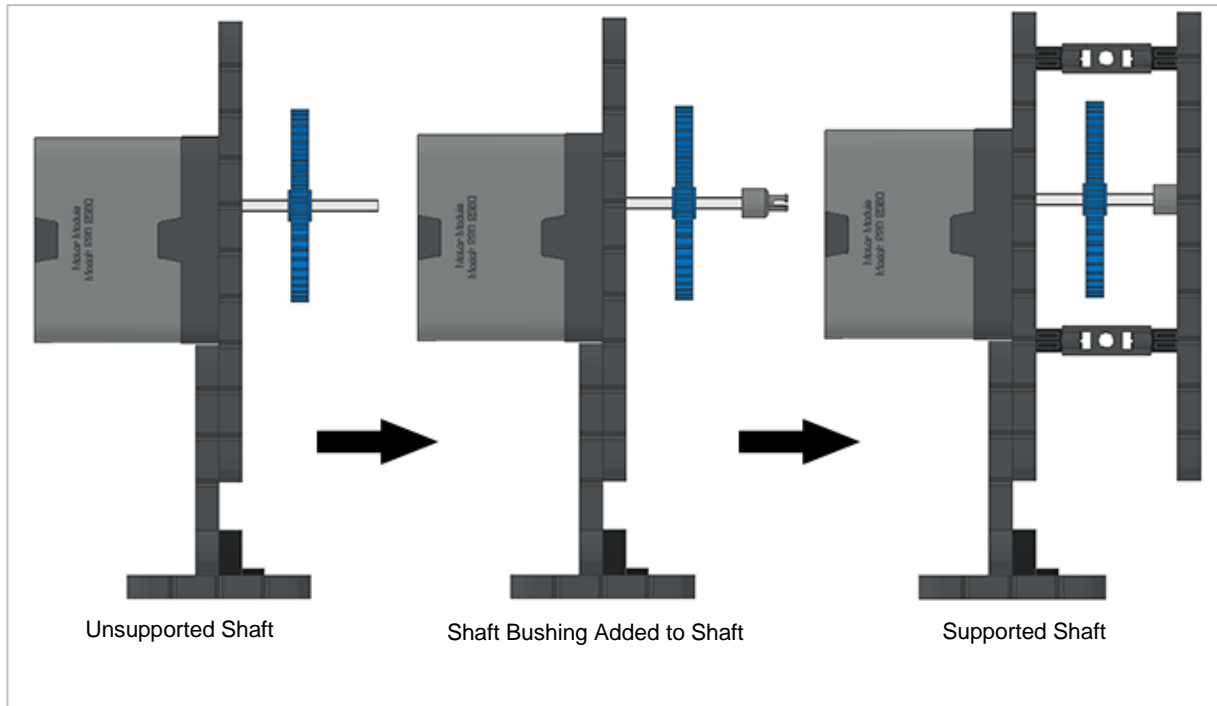


Supporting a shaft with a Rubber Shaft Collar

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

Stop and Discuss – Driving Forward and Reverse Exploration

These steps are very important because they will begin almost all programming explorations. For example you can say, “Let’s pause here for a moment. As a group summarize the steps we just completed. Record your summary in your engineering notebook.”

Remind the students that each group should have someone in the Recorder role. Allow students approximately 5 - 10 minutes to summarize the steps. If time allows, ask each group to share their summary. An example of what the summary could look like:

- Open the file menu
- Select examples
- Choose a template
- Name the project
- Save the project

Program Autopilot to Move Forward

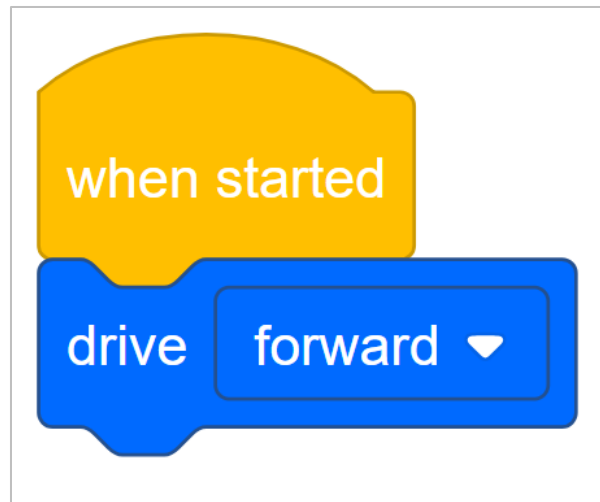
The `drive` block can be used to drive a robot forward or backwards.



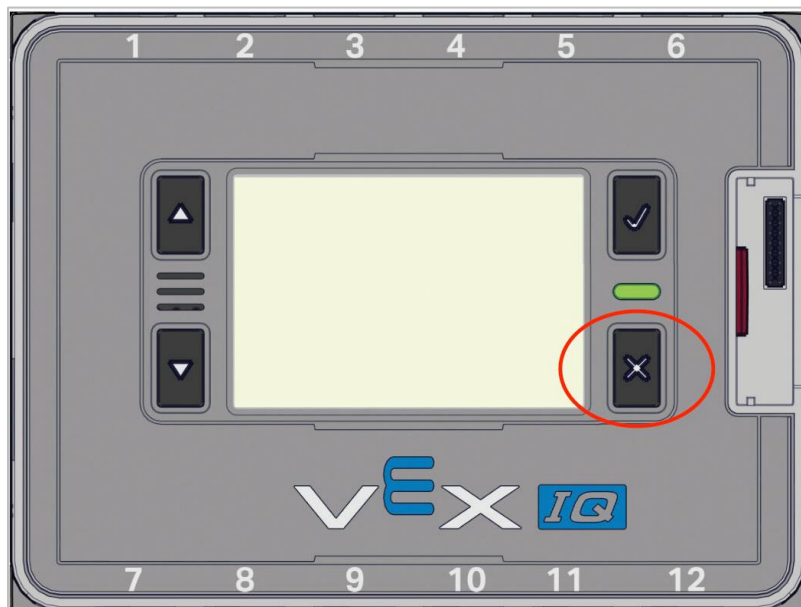
In order to move the Autopilot in reverse, the student will just need to change the `drive` block to “reverse.” Anything editable inside of the blocks is called a parameter.



When using the `drive` block, it is important to note that block used by itself does not cause the robot to stop moving. For example, if the project below was downloaded and run on a robot, the robot would drive forward without stopping.

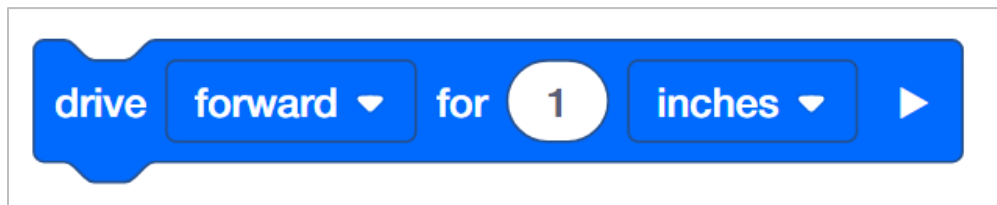


In order to get Autopilot to stop moving, the X would need to be pressed on Autopilot's Robot Brain. Pressing the X button on the Robot Brain ends the project and stops all of the motors.

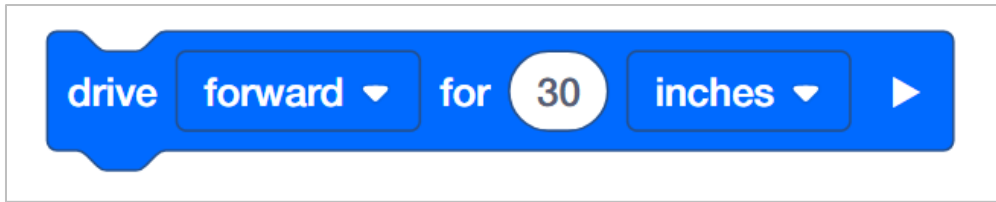


In the Forward and Reverse exploration, the students will just use the drive block. However, other blocks can be used to program basic movements.

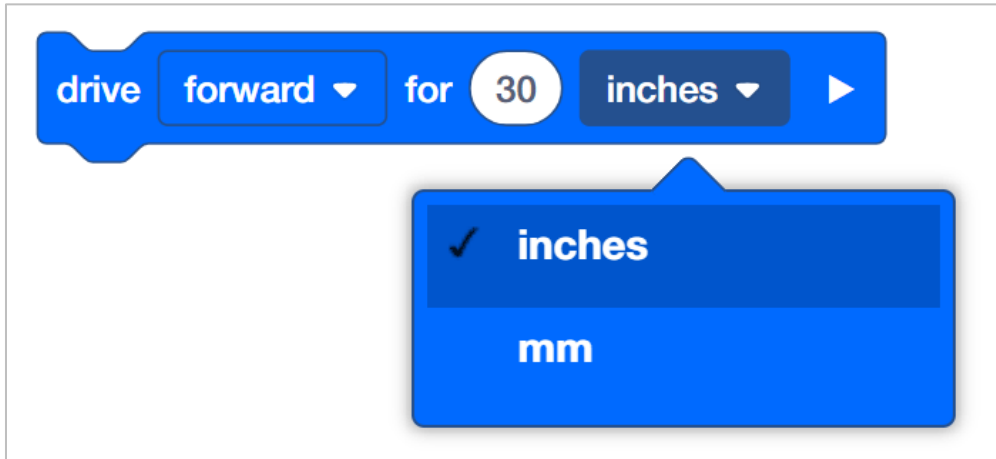
If the student wants to move a specific distance, they can use the drive for block.



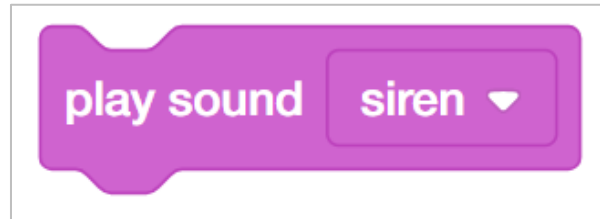
Students can change the distance traveled by adjusting the parameter in the block.



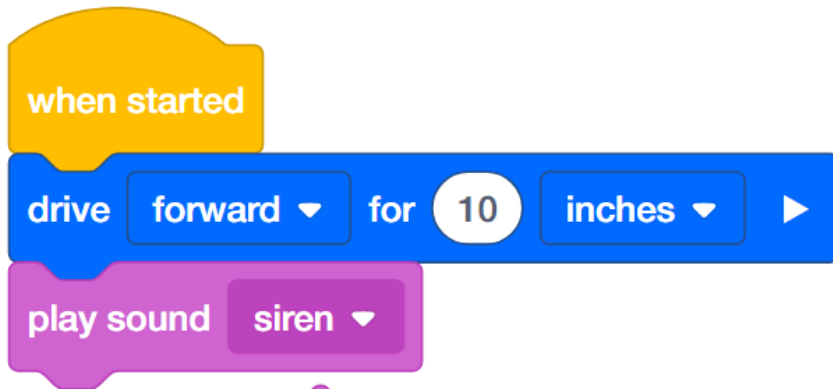
In the project above, the distance was changed from 1 inch to 30 inches. Students can program their robot to move in mm or inches.



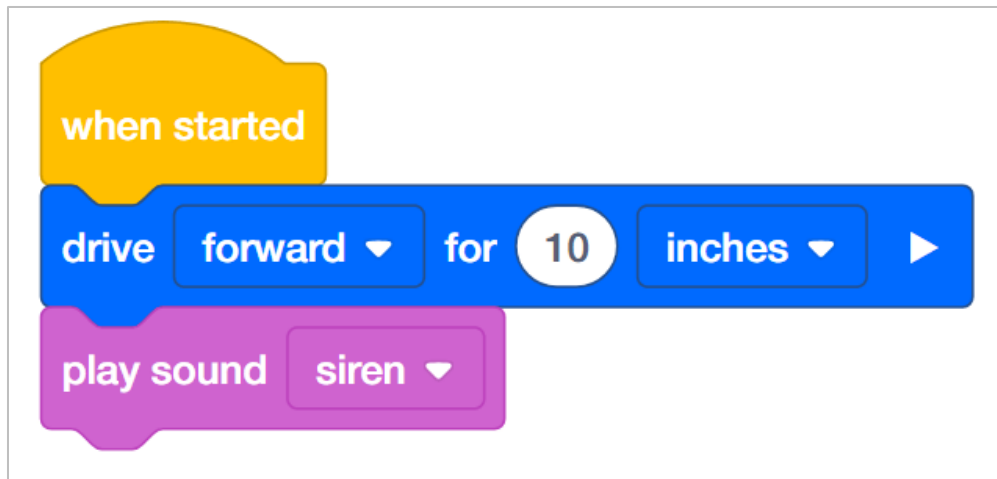
play sound Block



Students will be adding a play sound block to their program.

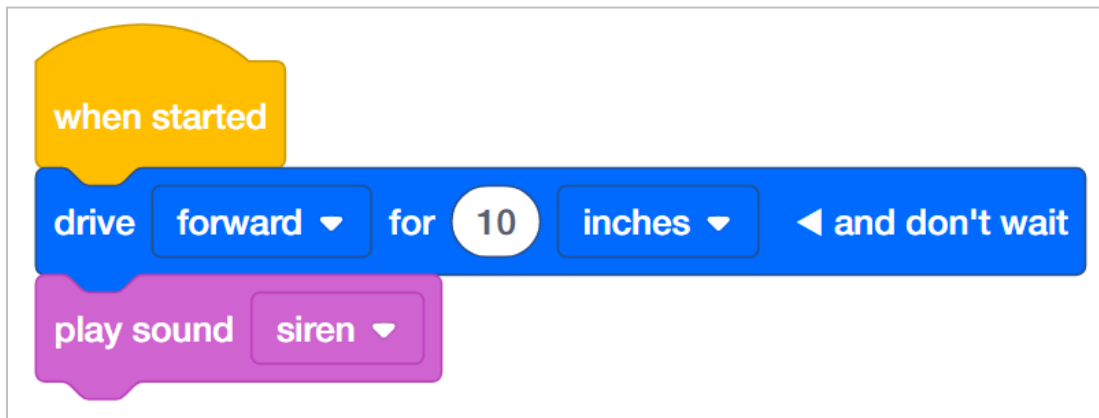


- ✓ siren
- wrong way
- wrong way (slow)
- fill up
- headlight on
- headlights off
- tollbooth
- alarm
- tada
- door close

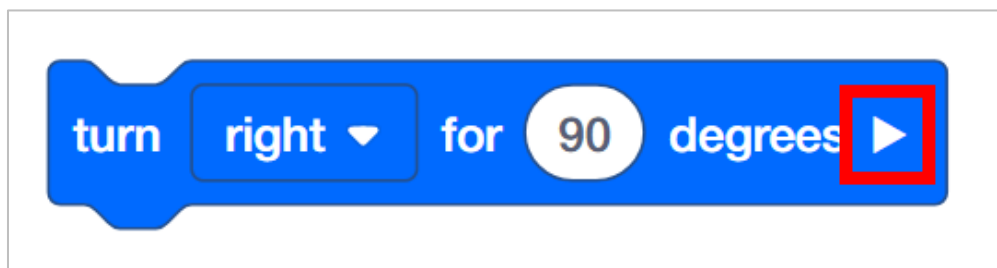


With this example, the robot will drive forward for 10 inches and then play the siren sound. The `drive for` block in this project is a waiting block. This means that the block will pause the rest of the stack until it has been completed. Therefore, the Autopilot will drive forward for 10 inches and then make the siren sound.

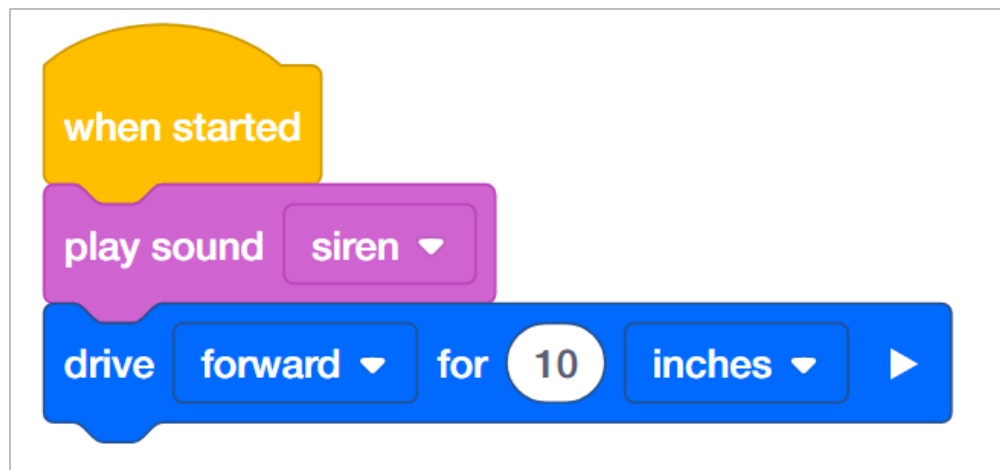
The `drive for` block can be expanded to become a non-waiting block, as in the example below.



With this program, as soon as Autopilot starts driving, it will immediately play the siren sound. This is because the `drive for` block is now set to non-waiting. It is NOT going to pause the stack until it has been completed. Some of the blocks in VEXcode IQ Blocks can be changed from waiting to non-waiting by selecting the arrow at the end of the block.



The `play sound` block is also a non-waiting block. Therefore, when the project below is run, Autopilot will play the siren and immediately begin to drive forward.



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 `drive` block 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.

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. Reminding students about who is supposed to be doing what can be a useful intervention.

Motivate Discussion – Driving Forward and Reverse Exploration

Motivate Discussion

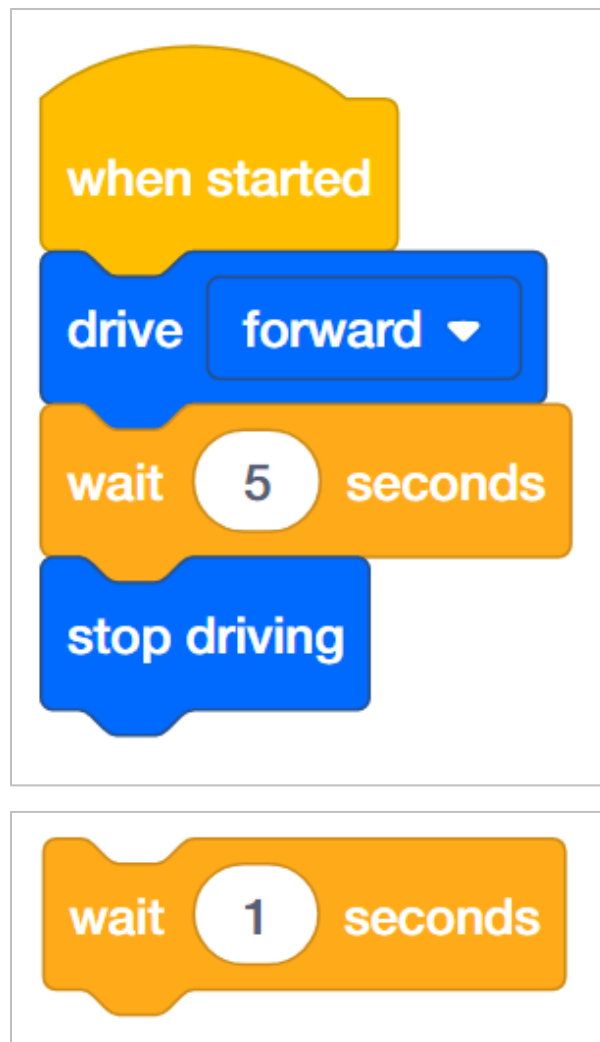
Q: Ask the students what they think would happen if they added a `stop driving` block to their project. (Have them record their answers in their engineering notebooks).



A: With this project, the robot would not move at all. This is because the project is telling the robot to drive forward, but then it is immediately telling it to stop. This happens so quickly that the robot does not move.

If students did answer that Autopilot would move forward and then stop, the students could then be asked how far the robot would move forward before it stopped. This could lead into the discussion that the project moves very quickly through the commands and eventually lead the students to realize that because the project does move so quickly through the commands that the robot does not move.

The discussion could then be a good introduction to the use of the `wait` block:



By using the wait block, a student can program a robot to drive forward for a certain amount of time. For example, in the project below, a robot would drive forward for 5 seconds and then stop.

Exploration Roles

Students can be organized 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 port? Is the Robot Brain turned on?) before a project is run.
- **Programmer** -- This person will use the `drive` block 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 has 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 makes 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.

Drive Forward and Reverse

Exploration Outline

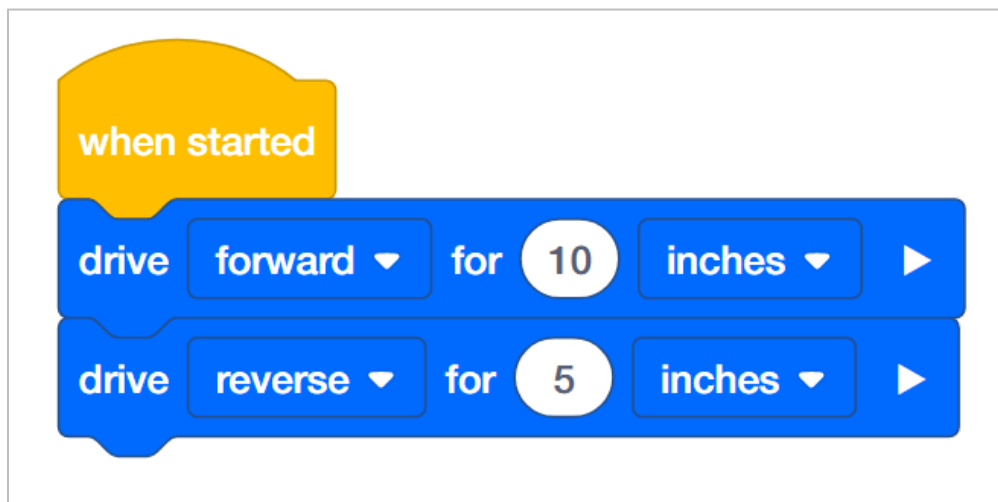
The outline for the Drive Forward and Reverse Exploration is as follows:

- Introduce the `drive` block
- Do a quick troubleshooting check that the VEX IQ Autopilot is ready
- Start a new project in VEXcode IQ Blocks
- Rename and save the project
- Create the Drive project that moves Autopilot forward
- Download and run the project
- Change the Drive project to move Autopilot in reverse
- Download and run the project
- Wrap up the activity with a discussion

drive for Activity B Program Answers

-
-
-

Bonus:



Demonstrating how the Drivetrain to Drive Blocks are Related

Have an Autopilot robot to demonstrate for the students. Introduce to students the drive block. Read the description of the drive block in the Help. Either have VEXcode IQ Blocks displayed in front of the classroom, or have each student group follow along at their workstation. When discussing the descriptor and purpose of the block, ask the students if they can identify what a Drivetrain is. Discuss with the students that a Drivetrain consists of:

- A rectangular Chassis (the structure of a mobile robot that holds wheels, motors, and/or any other hardware used to make up a Drivetrain)
- Two Motors
- Four Wheels
- Gears transmitting Power from the Motors to all Wheels

Use Autopilot robot to show the students the parts of the Drivetrain during the discussion. Next, gently turn one of the wheels that is connected to a motor. Show the students that because of the gears, even though the force is being applied to one wheel, both wheels are moving. Tell the students that instead of moving the wheels by hand, we'll use the `drive` block to program the motors to run and the wheels to turn.

Cultivating a Positive Learning Environment

Recognize and reinforce positive behaviors by creating a list of specific behaviors you want to encourage. Examples could include:

- Students self-organizing with the roles within a group
- Students performing each of their roles well within a group
- Students handling the robot and the computers/tablets with care
- Students praising and encouraging one another during the exploration

When students use these behaviors, praise them immediately. Be specific when offering praise. For example, instead of saying, “good job,” you could instead say, “good job carefully returning the Autopilot robot to the correct spot.”

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

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

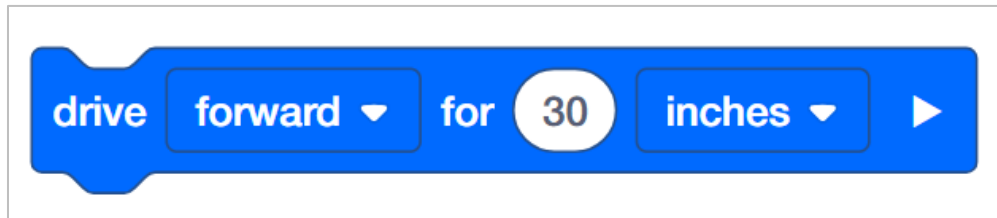
students in a group, each student can have one role.

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.

Adjusting Parameters

Students can change the distance traveled by adjusting the parameter in the block.



In the program above, the distance was changed from 1 inch to 30 inches. Students can program their robot to move in mm or inches.



How is the `drive for` block able to program a robot to move to a precise distance? In the Robot Configuration, you are able to specify the size of the wheels. For example, the typical wheel size for the Autopilot is 200 millimeters. This is the default setting in the Robot Configuration.

However, 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 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 programmed to move forward 2000 millimeters, that means that the wheels have to travel 10 complete rotations. The `drive for` block is able to track each degree the wheel moves via the encoders that are found within the motor.

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