Momentum Alley

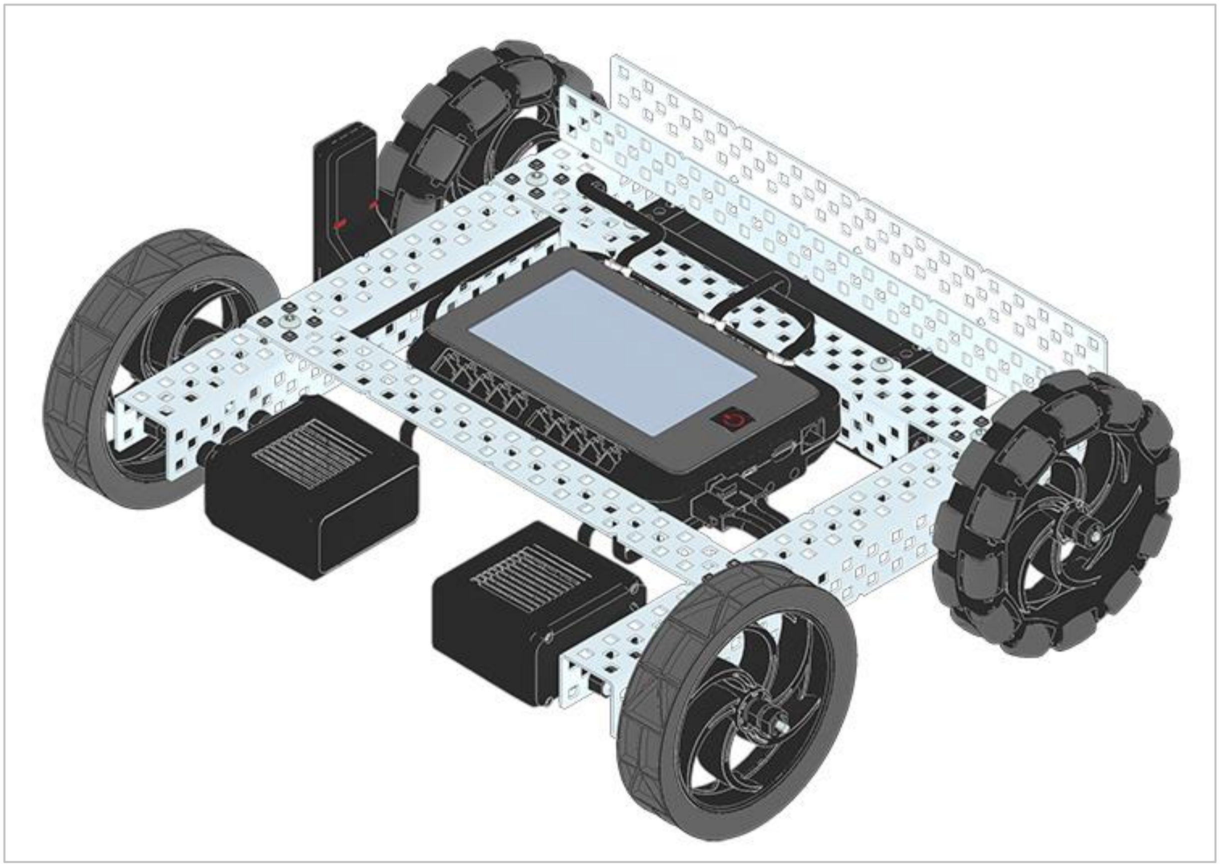
Explore momentum and bowl with the VEX V5 Speedbot!

SPARK

SEEK

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

The Completed Look of the Build



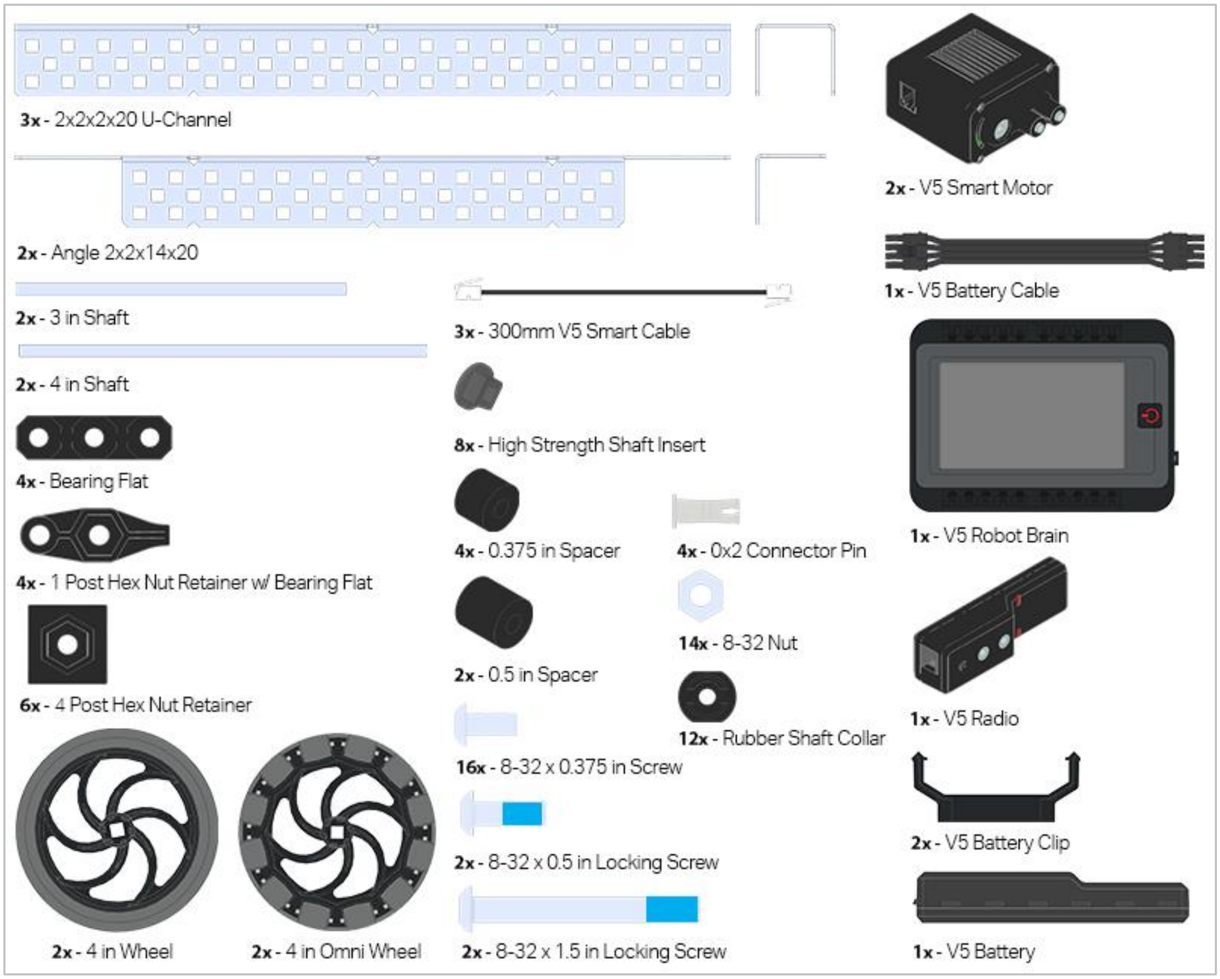
Completed VEX V5 Speedbot

This robot is designed so that it can be built quickly and driven around either autonomously or with the V5 Controller.

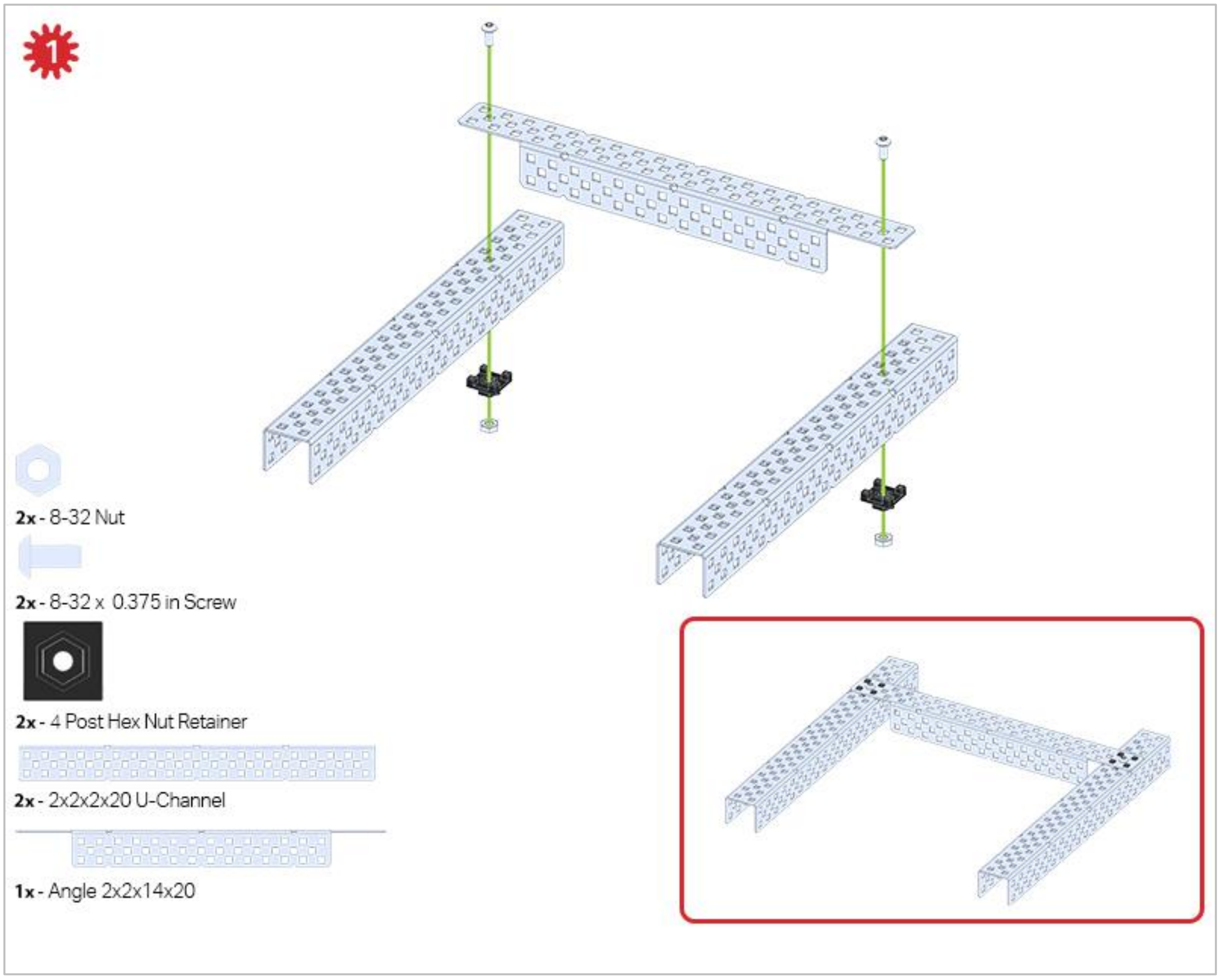
Parts Needed

Can be built with:

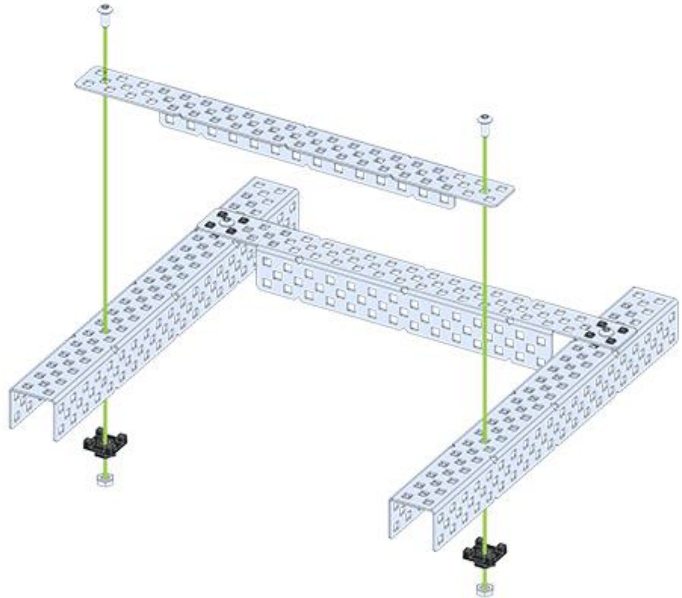
- VEX V5 Classroom Starter Kit



Build Instructions



2

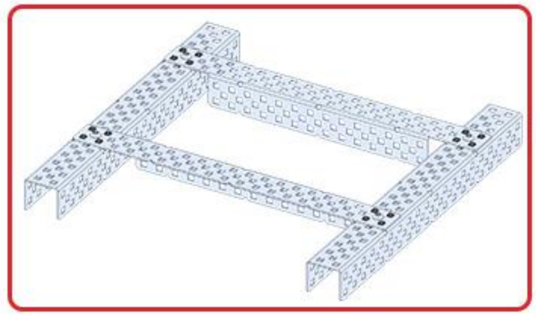


2x - 8-32 Nut

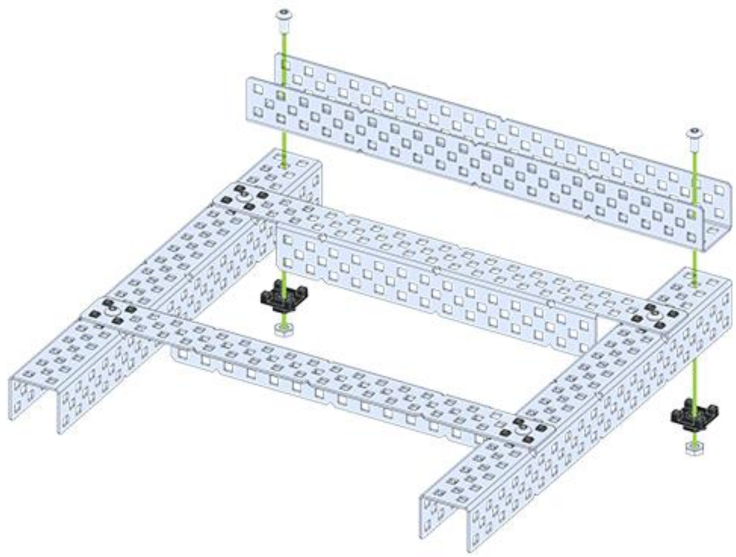
2x - 8-32 x 0.375 in Screw

2x - 4 Post Hex Nut Retainer

1x - Angle 2x2x14x20



3

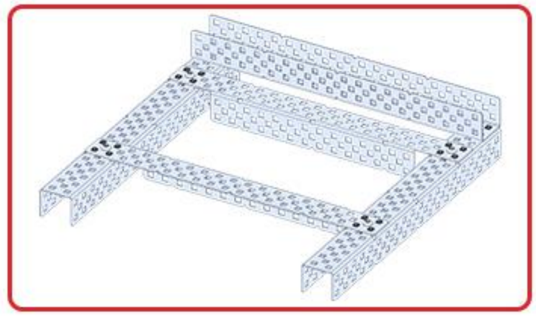


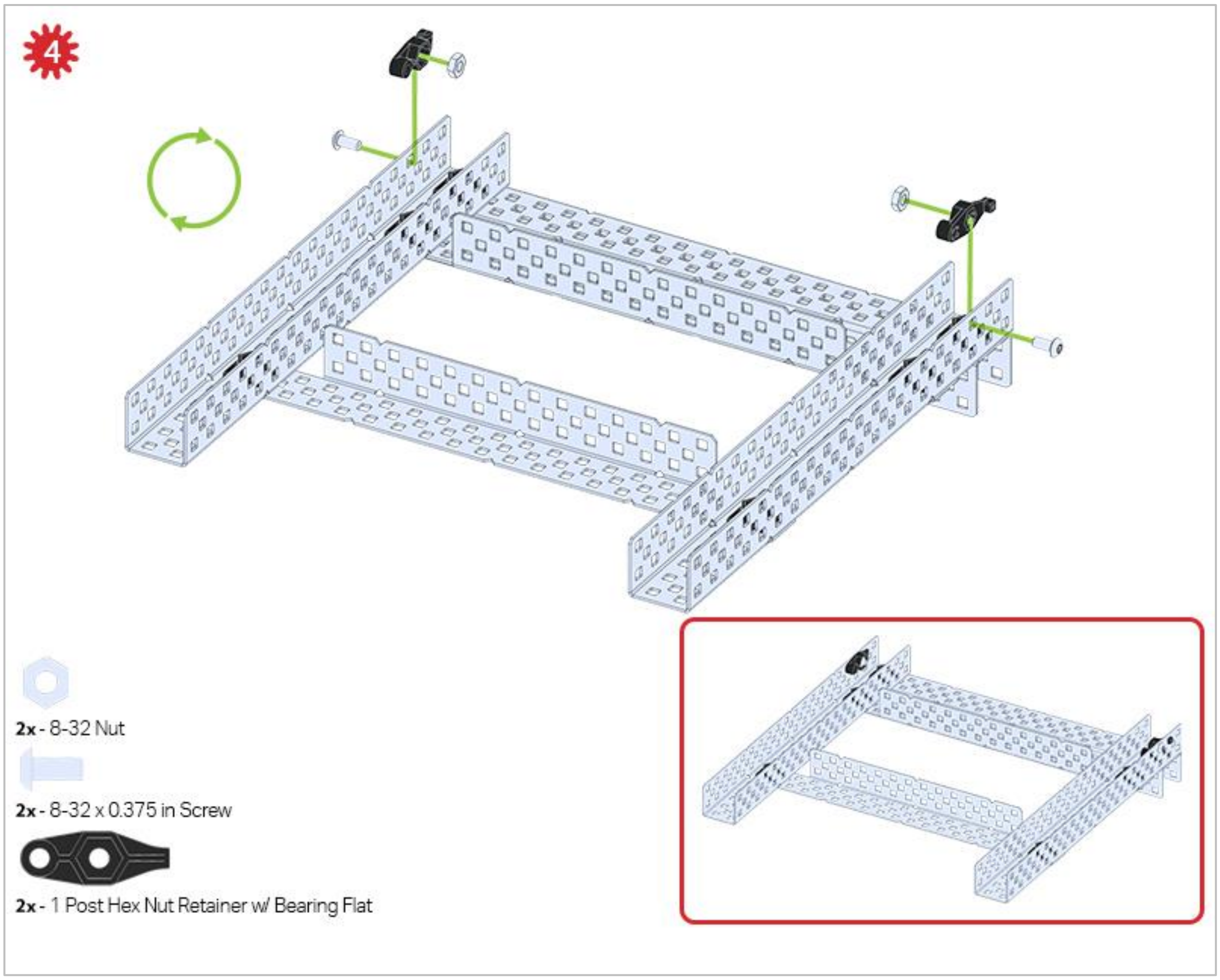
2x - 8-32 Nut

2x - 8-32 x 0.375 in Screw

2x - 4 Post Hex Nut Retainer

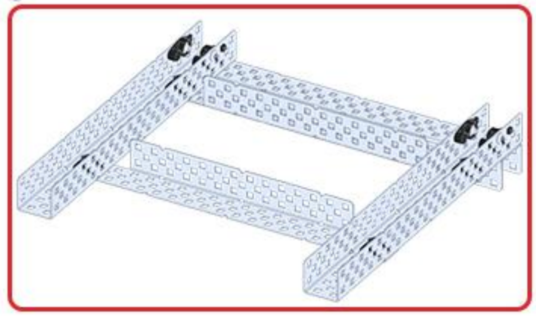
1x - 2x2x2x20 U-Channel



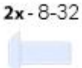


The green icon indicates that the build needs to be flipped over (upside down).

5

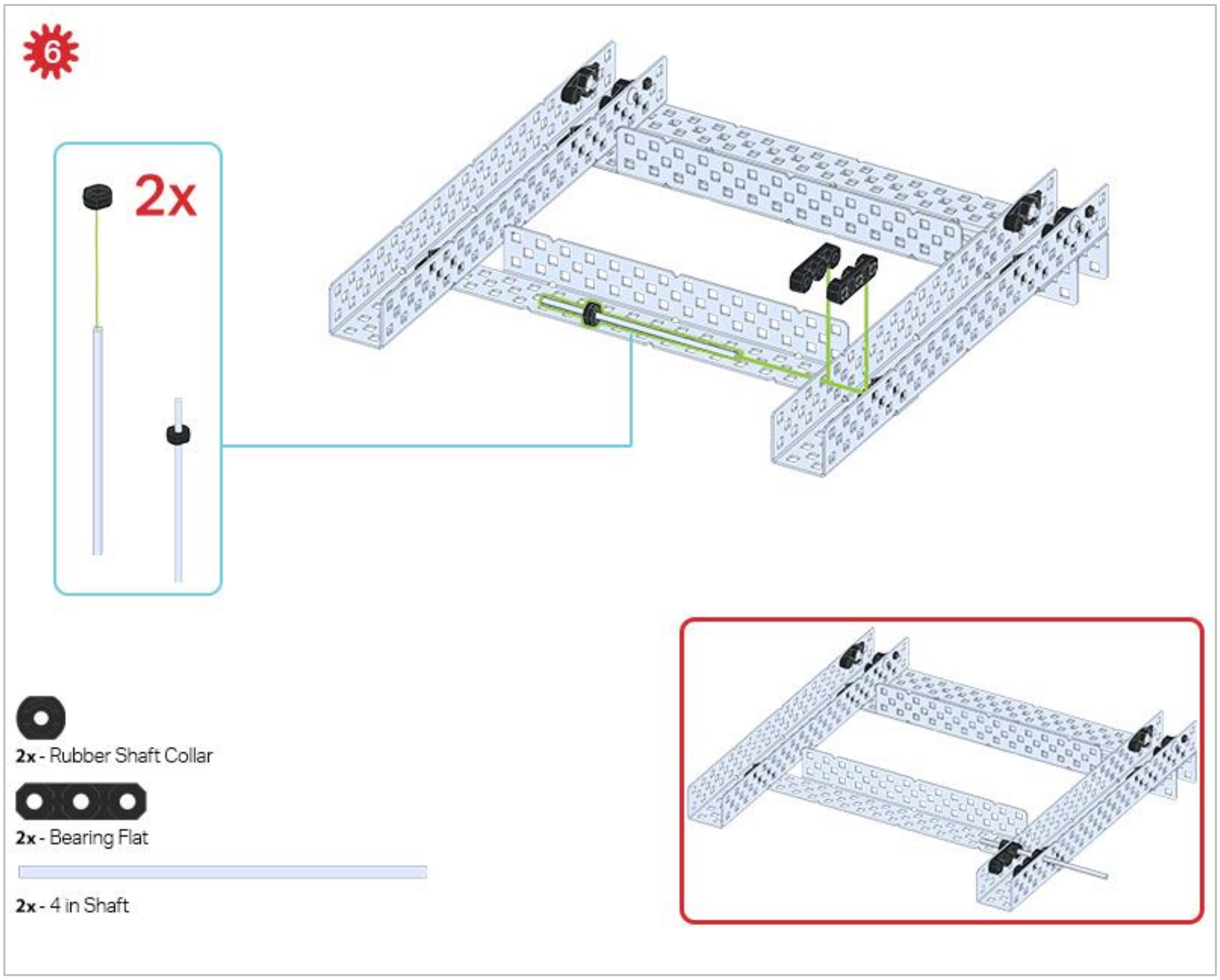


2x - 8-32 Nut

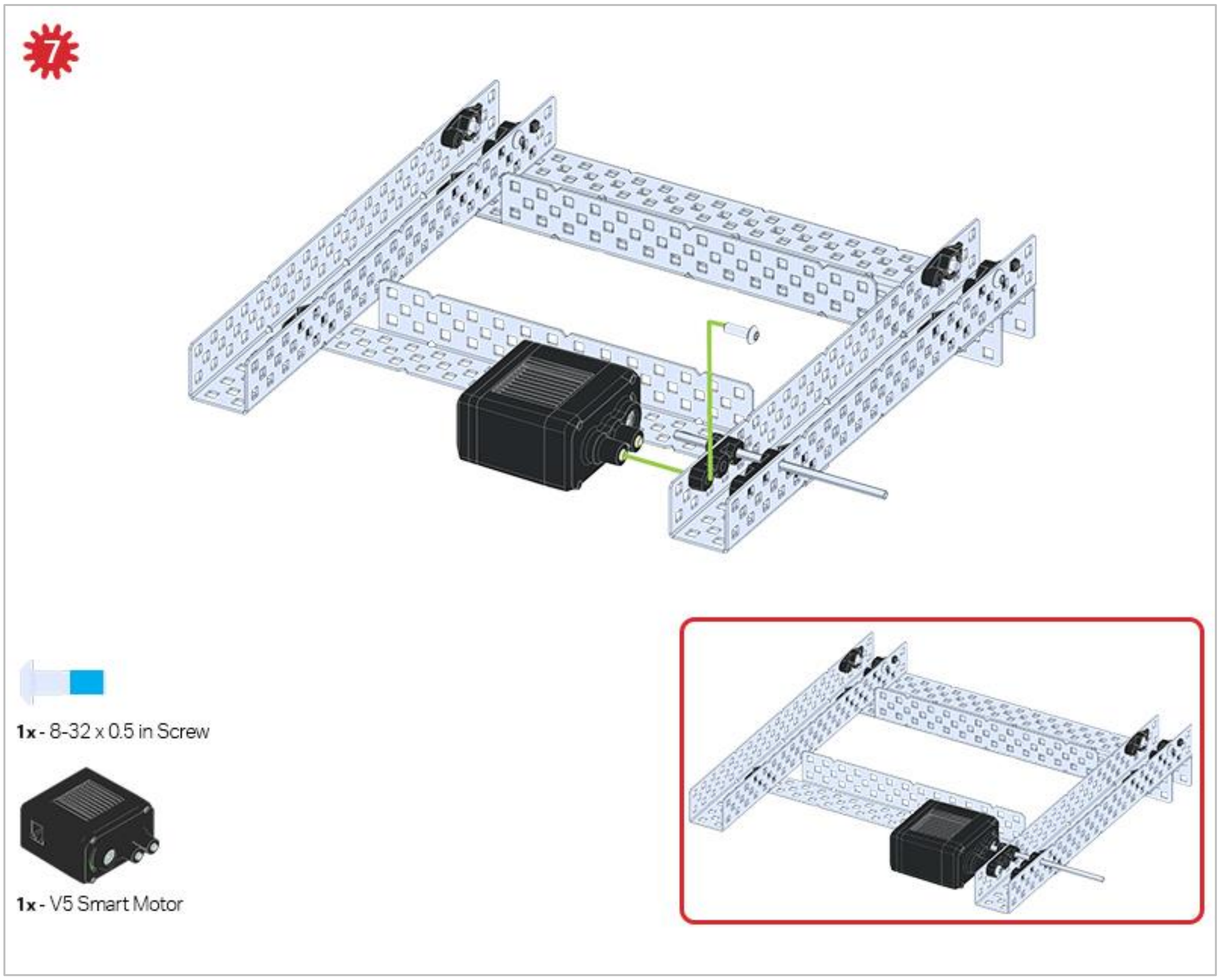


2x - 8-32 x 0.375 in Screw

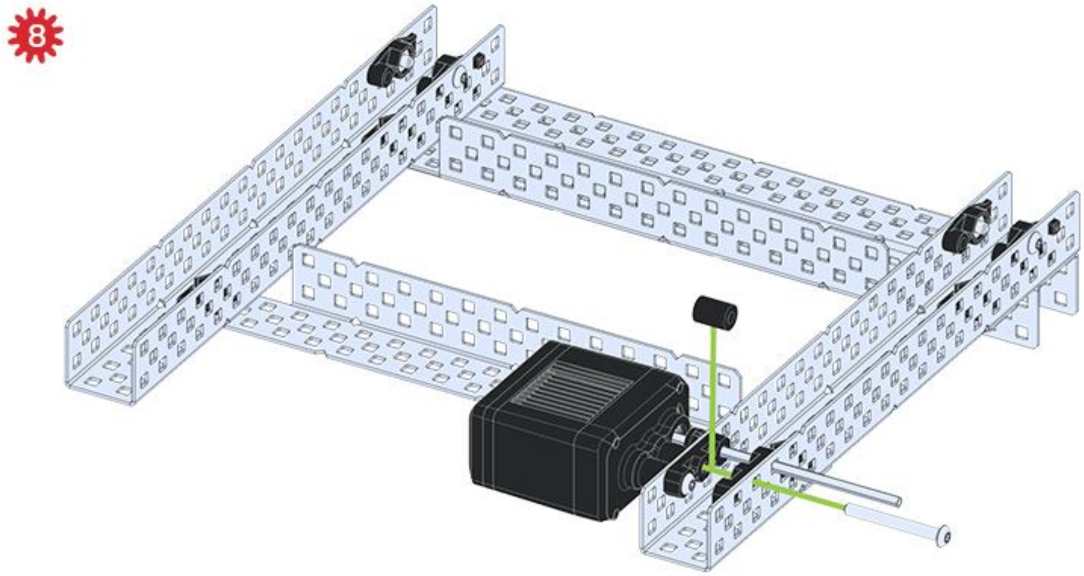
2x - 1 Post Hex Nut Retainer w/ Bearing Flat

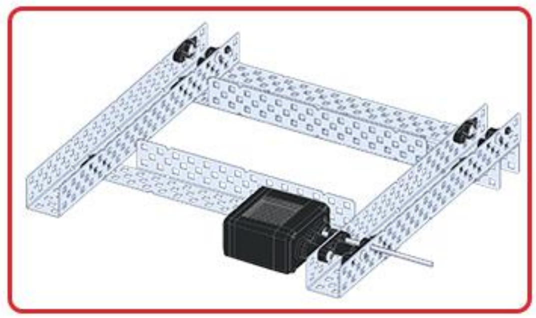


Only one of the two sub-assemblies made in this step is used right now. The other will be used later in step 9.



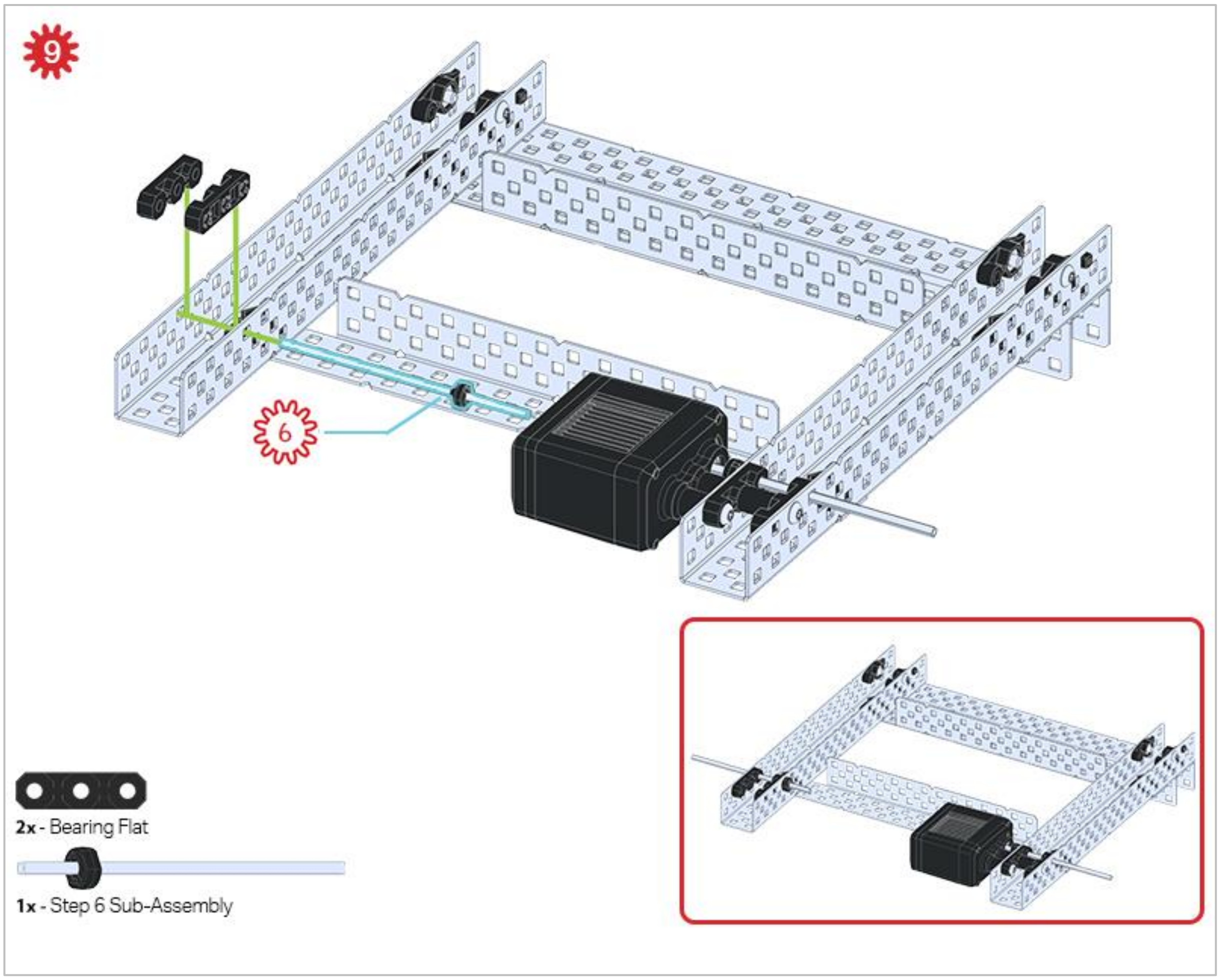
Make sure your Smart Motors are oriented in the correct direction (screw holes facing the outside of the build and the shaft hole towards the inside).



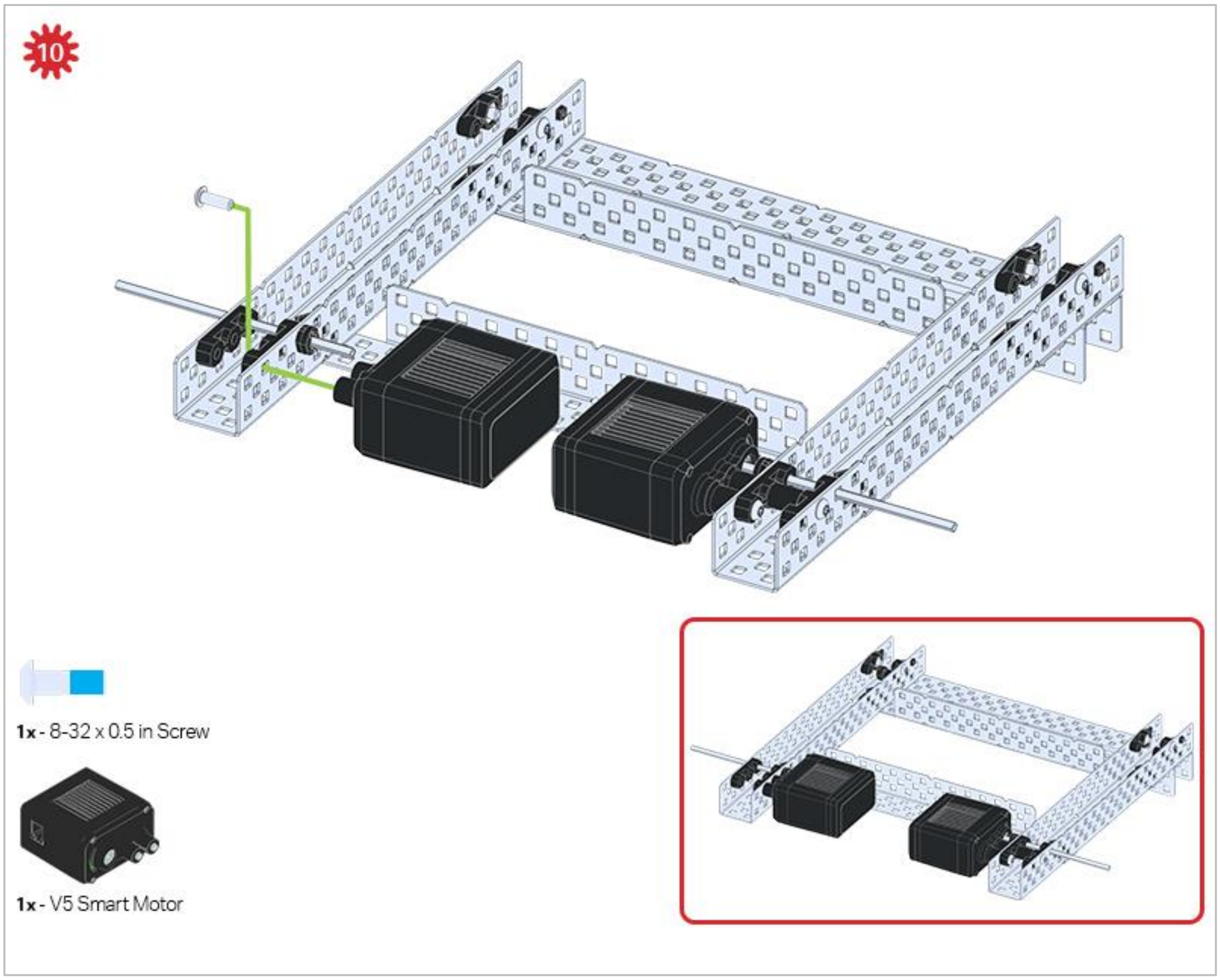


1x - 0.5 in Spacer

1x - 8-32 x 1.5 in Screw

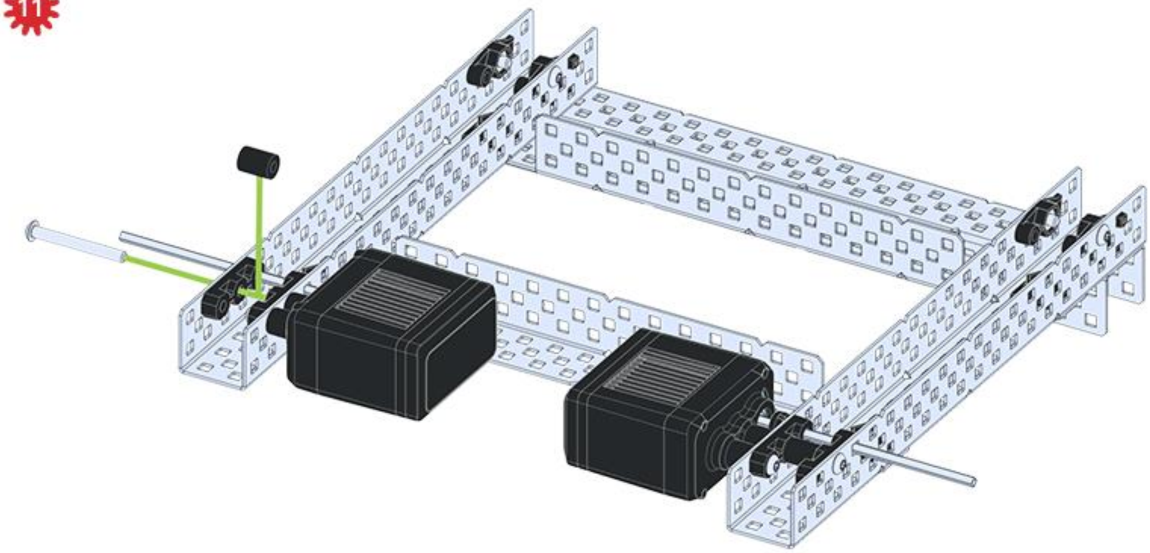


9
6
2x - Bearing Flat
1x - Step 6 Sub-Assembly



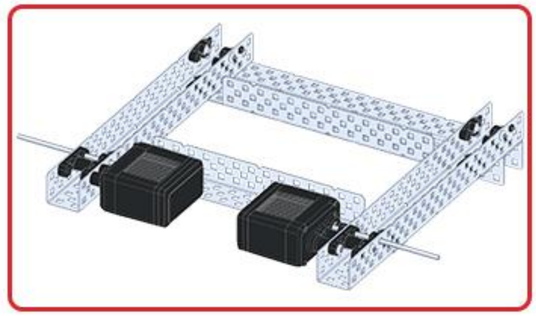
Make sure your Smart Motors are oriented in the correct direction (screw holes facing the outside of the build and the shaft hole towards the inside).

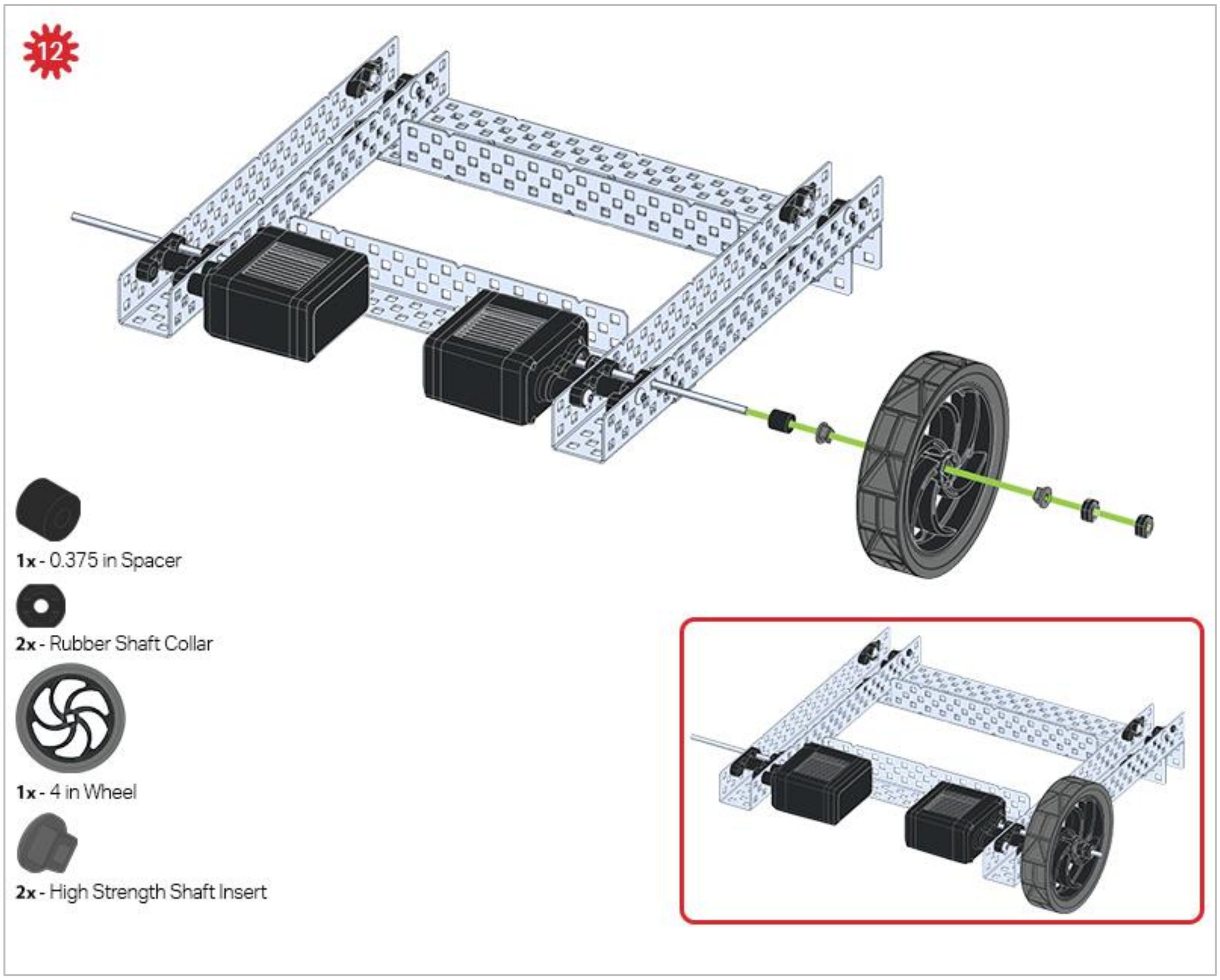
11



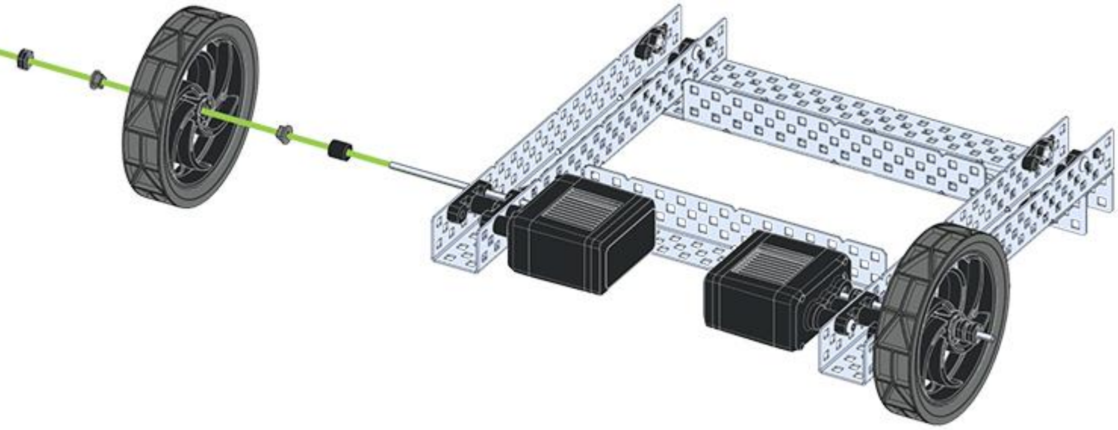
1x - 0.5 in Spacer

1x - 8-32 x 1.5 in Screw




12
1x - 0.375 in Spacer
2x - Rubber Shaft Collar
1x - 4 in Wheel
2x - High Strength Shaft Insert

13

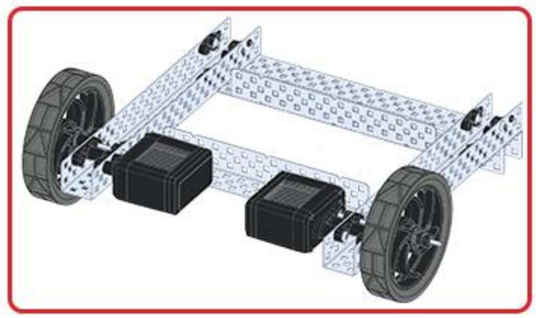


1x - 0.375 in Spacer

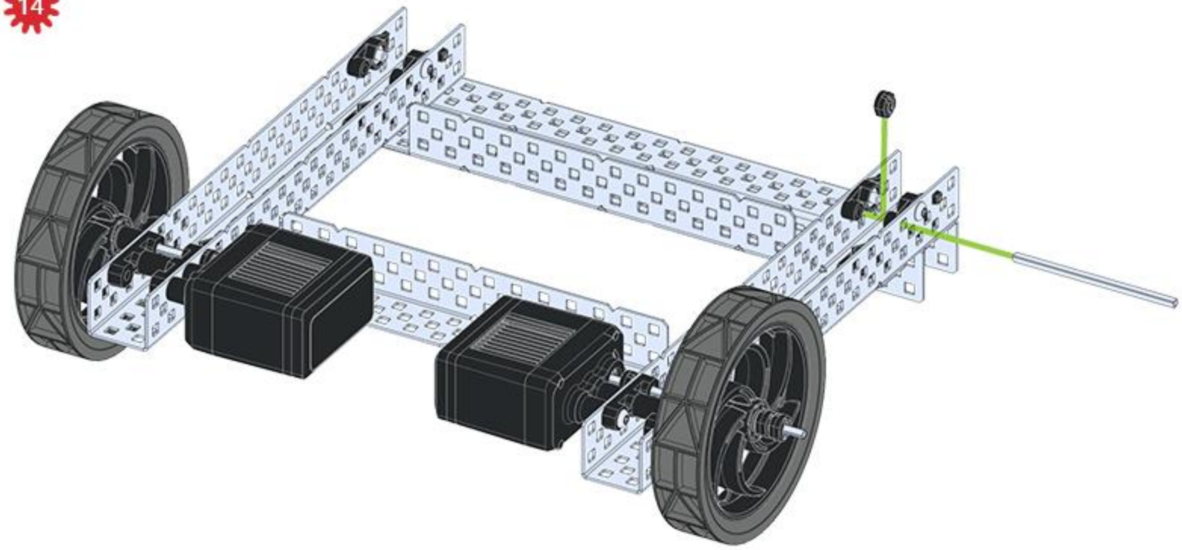
2x - Rubber Shaft Collar

1x - 4 in Wheel

2x - High Strength Shaft Insert

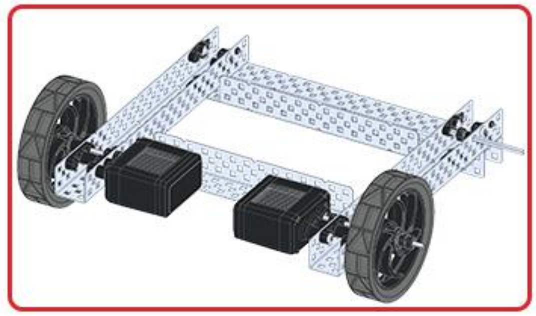


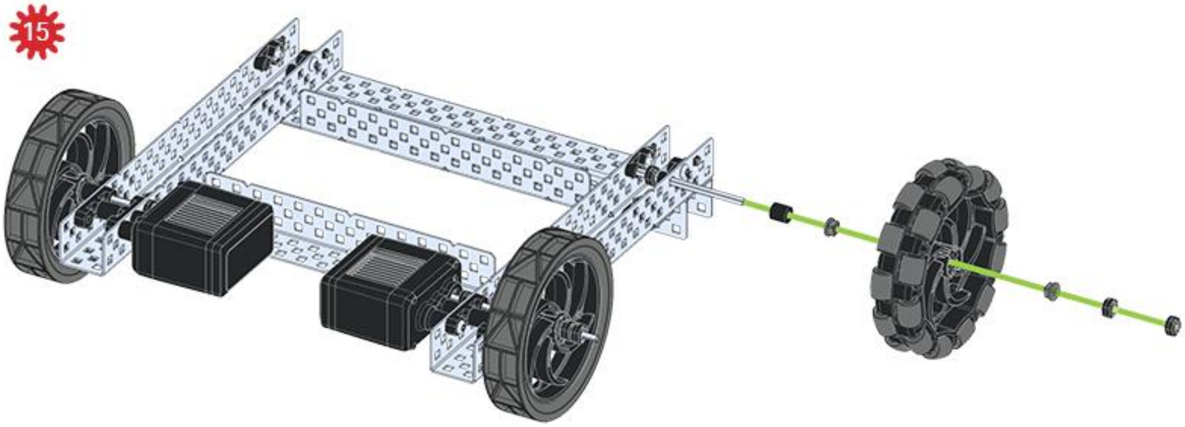
14



1x - Rubber Shaft Collar

1x - 3 in Shaft



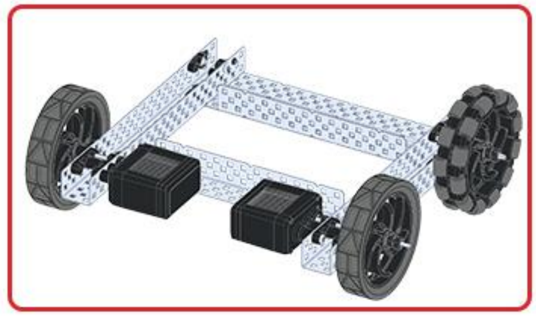


1x - 0.375 in Spacer

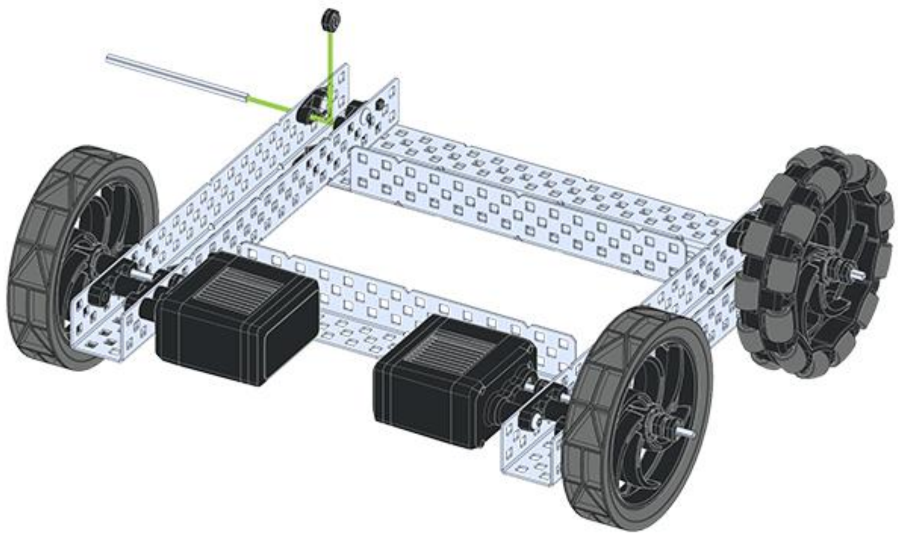
2x - Rubber Shaft Collar

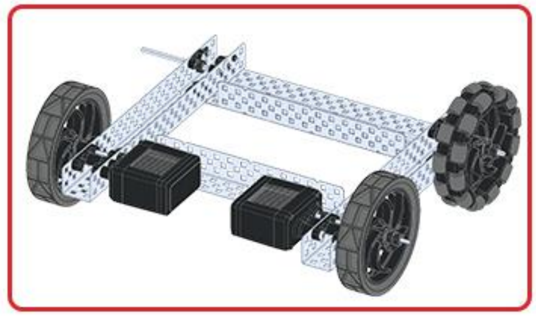
1x - 4 in Omni Wheel

2x - High Strength Shaft Insert



16

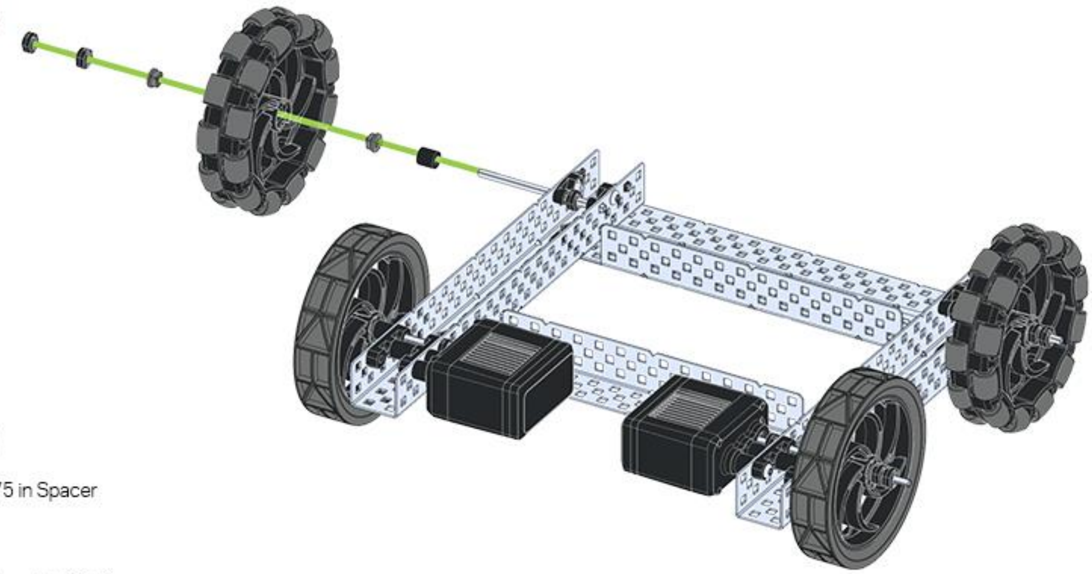




1x - Rubber Shaft Collar

1x - 3 in Shaft

17

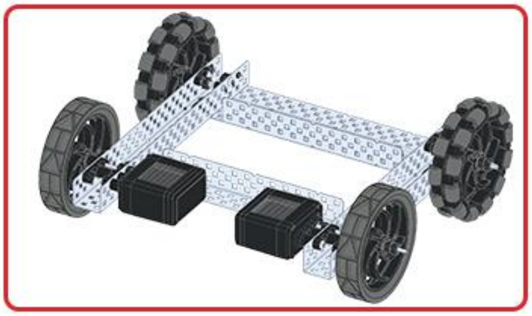


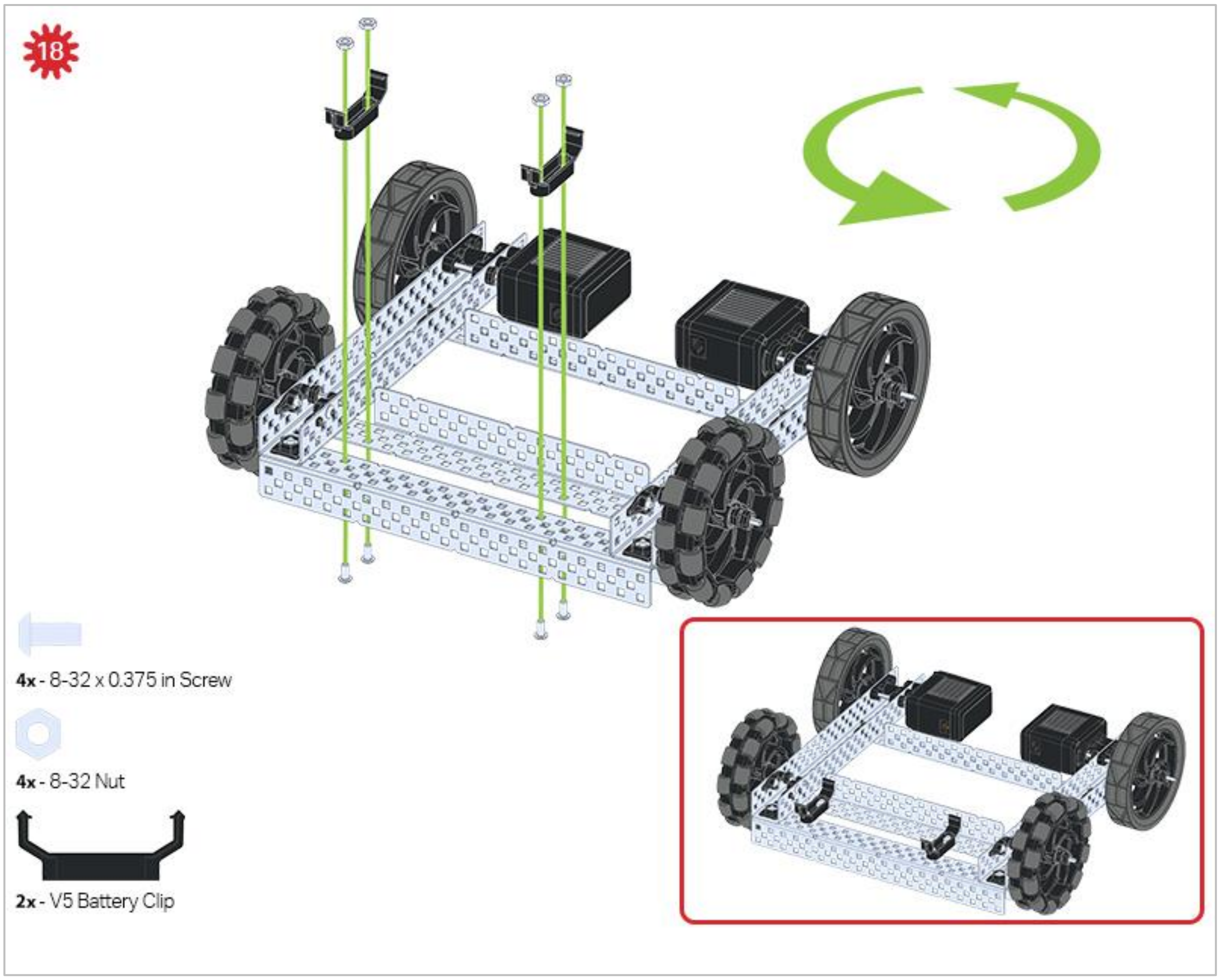
1x - 0.375 in Spacer

2x - Rubber Shaft Collar

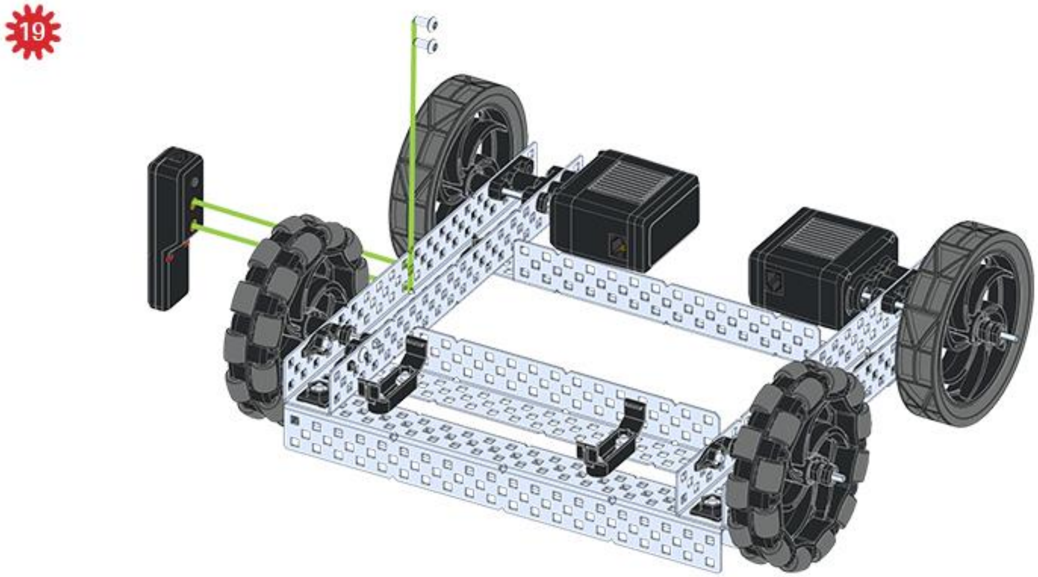
1x - 4 in Omni Wheel

2x - High Strength Shaft Insert



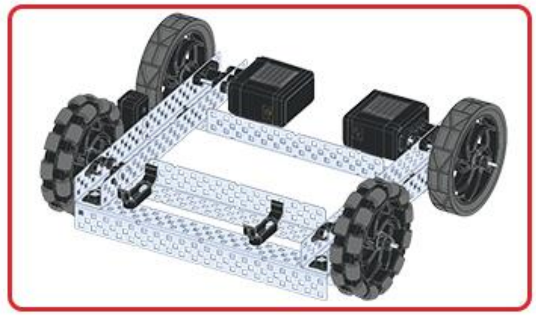


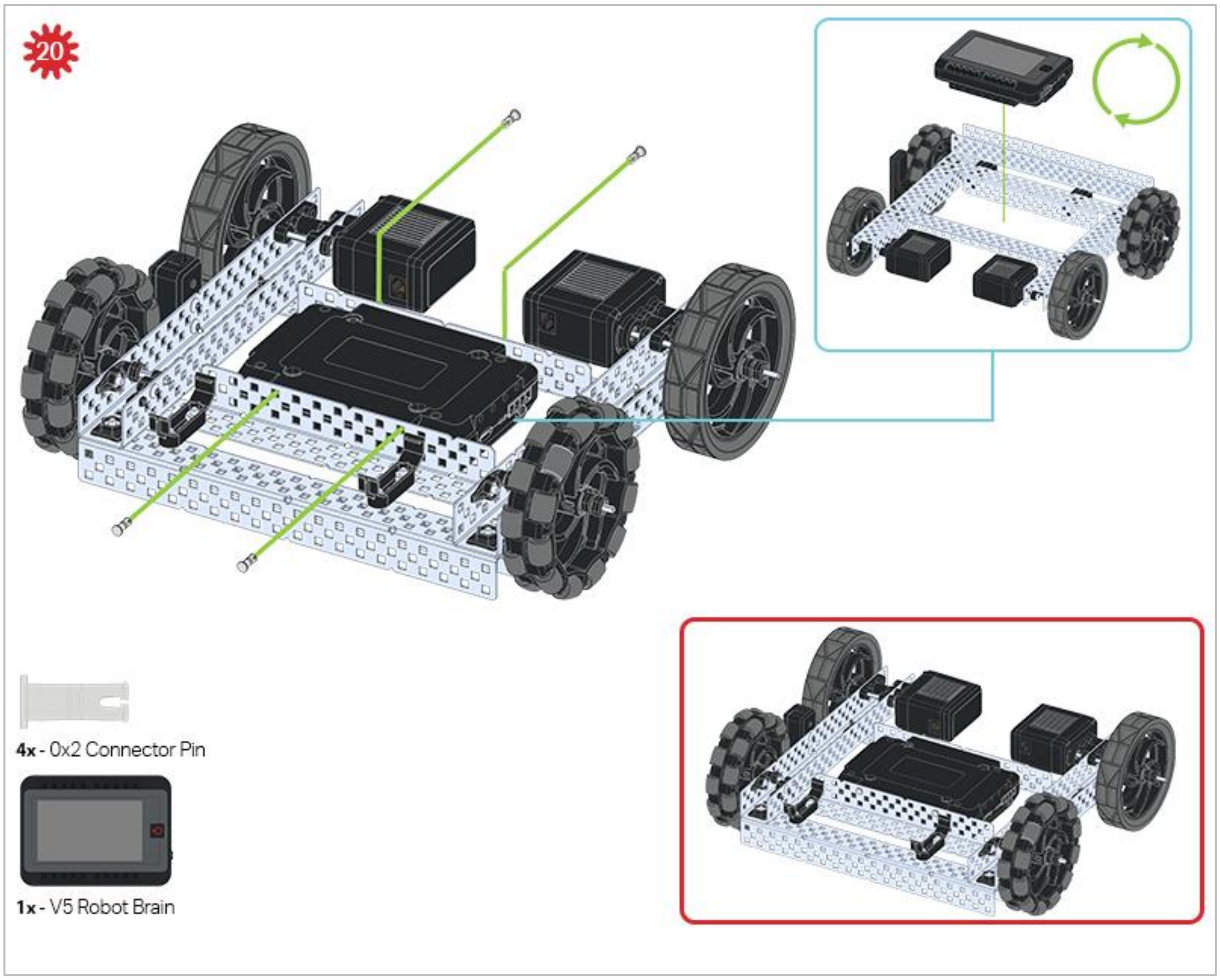
The green icon indicates that the build needs to be rotated (180 degrees).



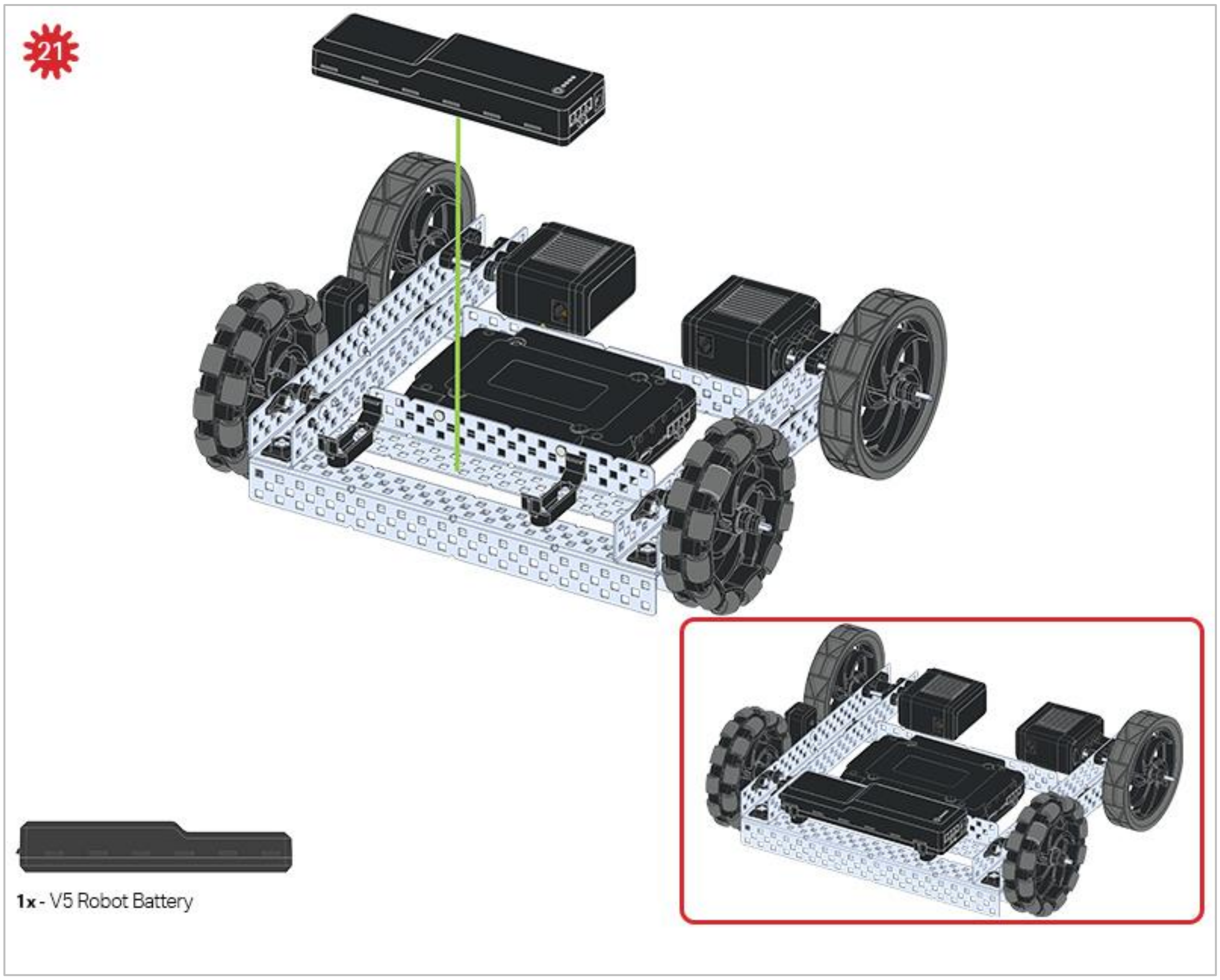
1x - V5 Radio

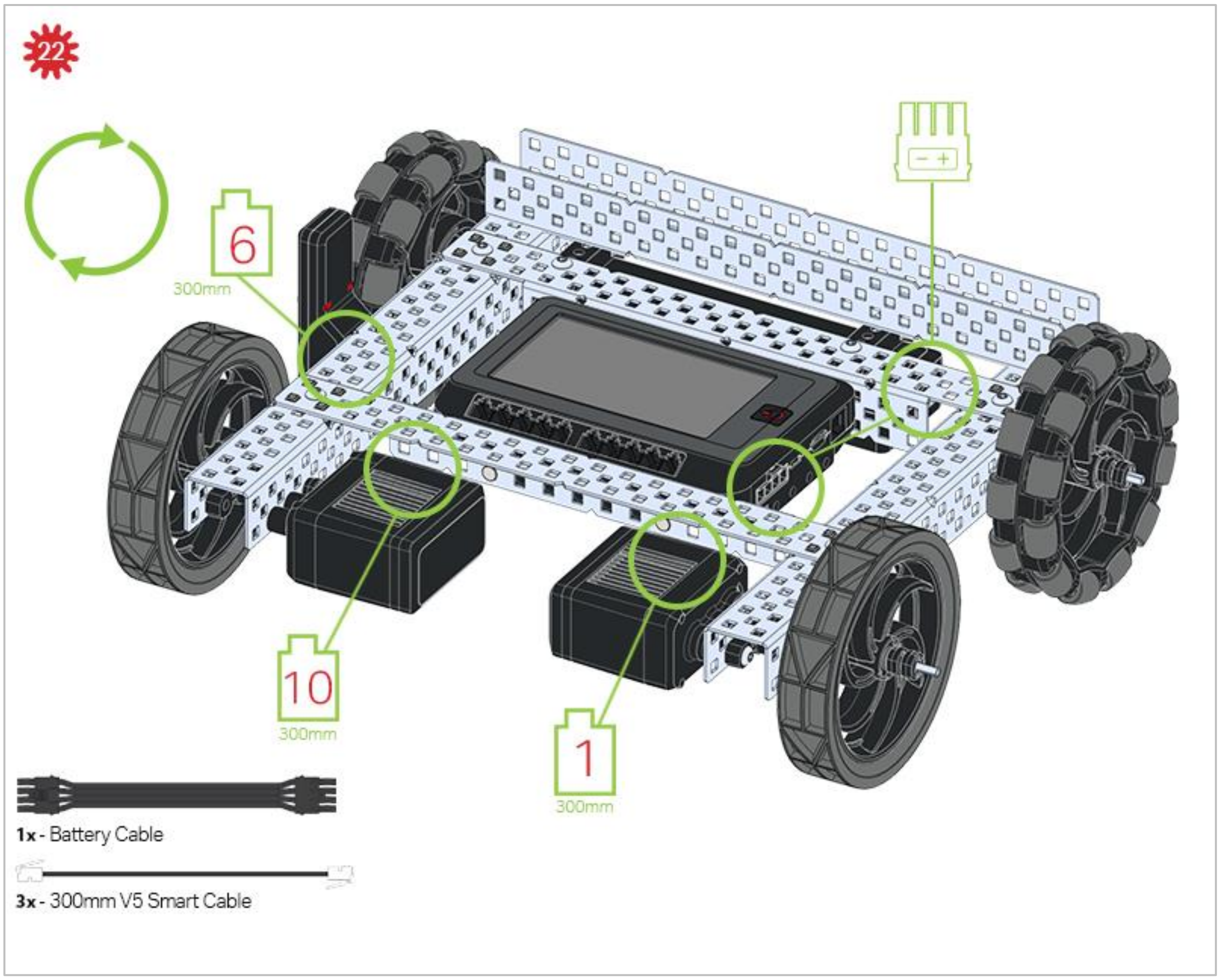
2x - 8-32 x 0.375 in Screw





The blue call out shows what the orientation of the Robot Brain should be if the build were flipped right side up. Make sure the 3 wire ports on the Robot Brain are facing the V5 Radio!


21
1x - V5 Robot Battery



The green call outs indicate which port on the Robot Brain to plug each device into using their respective cable.

Build Instruction Tips

Check the Appendix for info on how to use the new Hex Nut Retainers.

Exploration

The VEX V5 Speedbot is meant to get you comfortable working with the system as quickly as possible. Now that the build is finished, explore and see what it can do. Then answer these questions in your engineering notebook.

- How can this robot build be more beneficial in some scenarios compared to a bigger and more complex build?
- If this robot had the task of pushing a ball forward, very similar to a basketball or bowling ball, what features of the robot build would be important? What characteristics of the ball would have to be considered?
- If you were tasked with designing a new version of this build, what would you add to or remove from the build to improve its ability to push objects? Explain with details and sketches.

SPARK
PLAY

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

Momentum

Bowling ball strikes pins

The Science of Momentum

Momentum can be thought of as the amount of motion that an object has. Momentum is determined by the object's mass (how much is moving) and its velocity (how fast it is moving). In physics, momentum is represented by "p" so the equation for determining momentum is $p = mv$ (momentum equals mass times velocity). A heavier object, like a bowling ball, traveling at the same speed as a lighter object, like a kickball, would have greater momentum according to the laws of physics.

For this STEM Lab, you need to understand momentum and how it affects collisions because later, you will compete in a bowling game called the Strike Challenge. You will program the Speedbot to run into a ball in order to knock down pins and earn the highest score. Let's think about what happens when two objects collide. They transfer kinetic energy, yes. But, the momentum of each of the two colliding objects before the collision can predict how the two objects will react after the collision. We will return to this idea later.

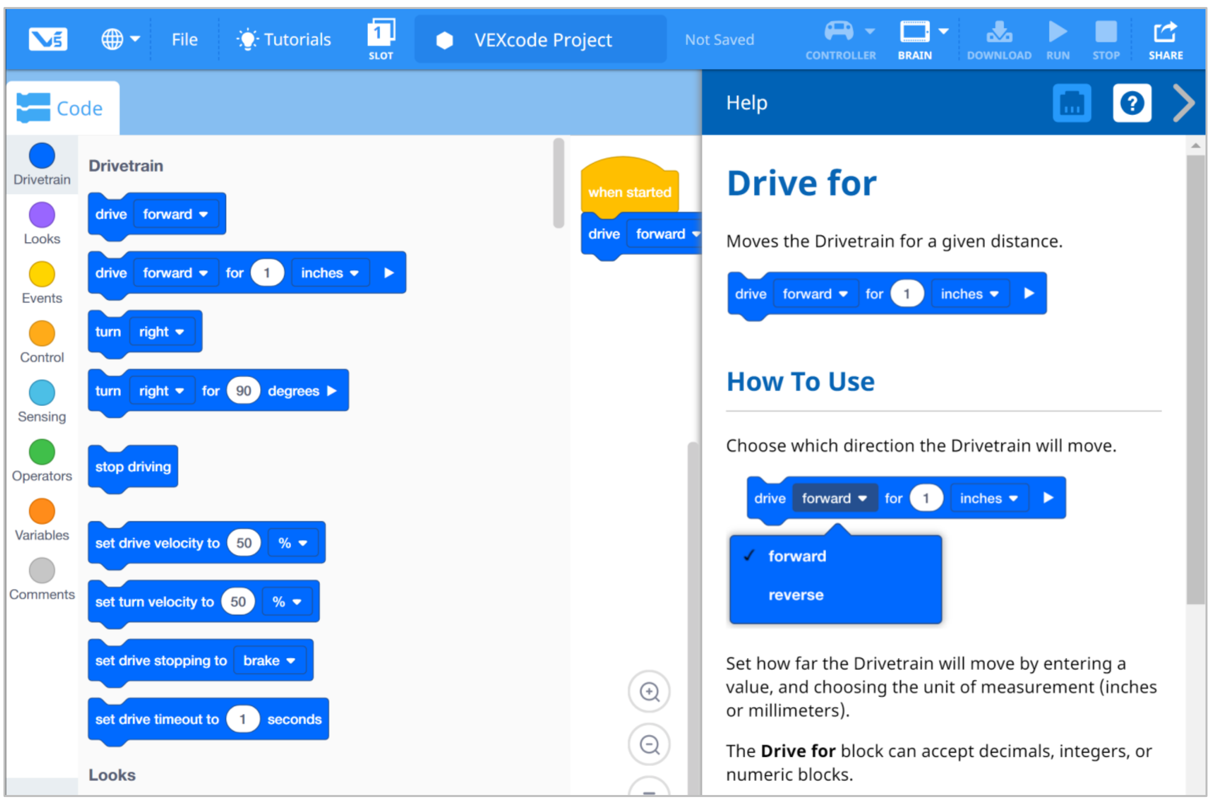
Programming Drive Forward and Reverse - VEXcode V5 Blocks

The Speedbot is ready to move!

This exploration will give you the tools to be able to program your Speedbot for simple movements that you can later apply to compete in the Strike Challenge bowling game.

- VEXcode V5 Blocks that will be used in this exploration:

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



- Make sure you have the hardware required, your engineering notebook, and VEXcode V5 Blocks downloaded and ready.

Hardware/Software Required:

Amount	Hardware/Other Items
1	Speedbot Robot
1	Charged Robot Battery
1	VEXcode V5 Blocks
1	USB Cable (if using a computer)
1	Engineering Notebook

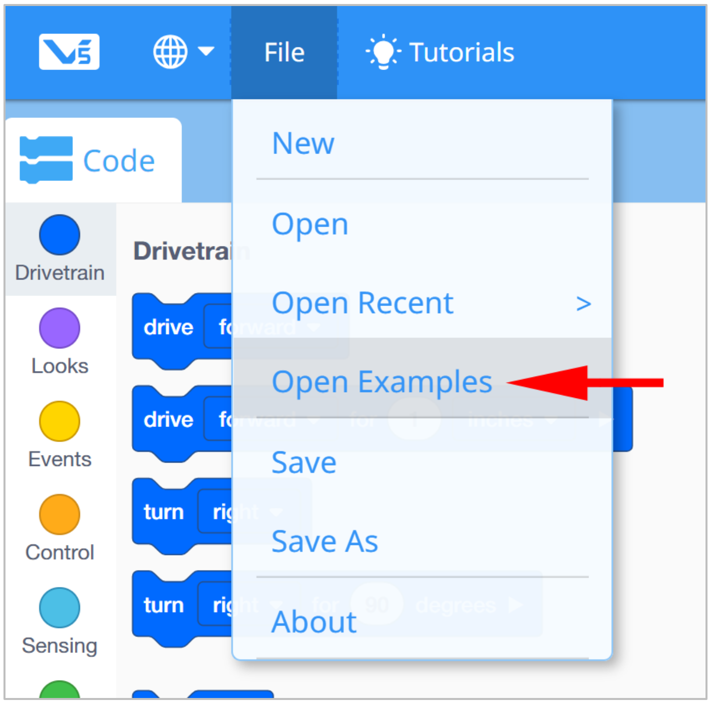
1. Preparing for the Exploration

Before you begin the activity, do you have each of these items ready? Check each of the following:

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

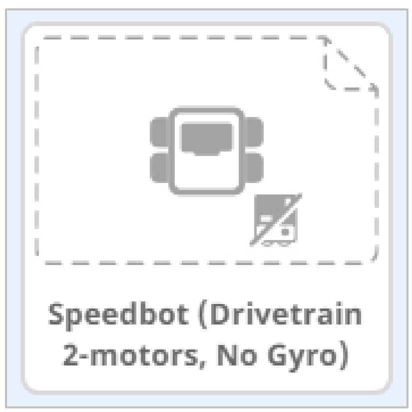
2. Start a New Project

Before you begin your project, select the Speedbot (Drivetrain 2-Motors, No Gyro) template project. The template project contains the Speedbot's motor configuration. If the template is not used, your robot will not run the project correctly.



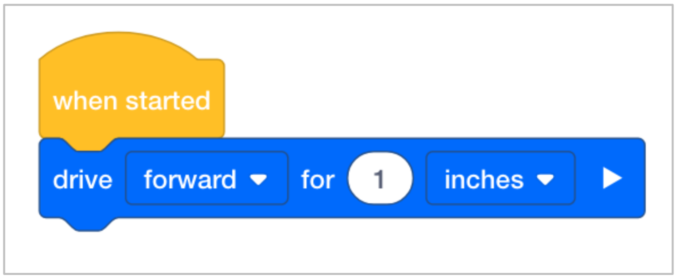
Complete the following steps:

- Open the File menu.
- Select **Open Examples**.

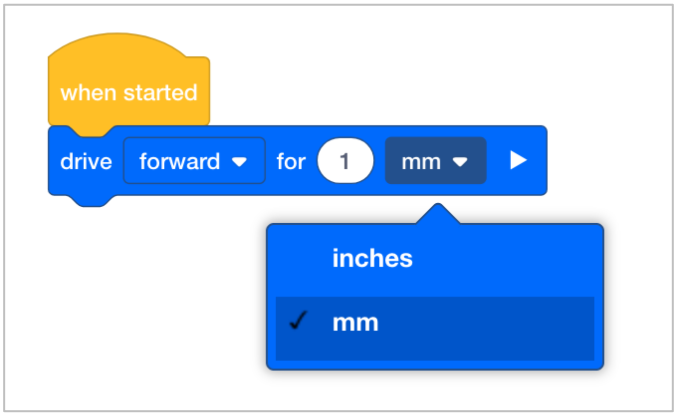


- Select and open the Speedbot (Drivetrain 2-motors, No Gyro) template project.
- Since we will be using the `drive for` block, rename your project Drive.
- **Save** your project.
- Check to make sure the project name Drive is now in the window in the center of the toolbar.

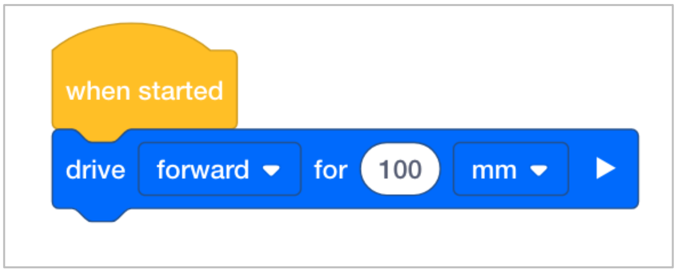
3. Drive Forward



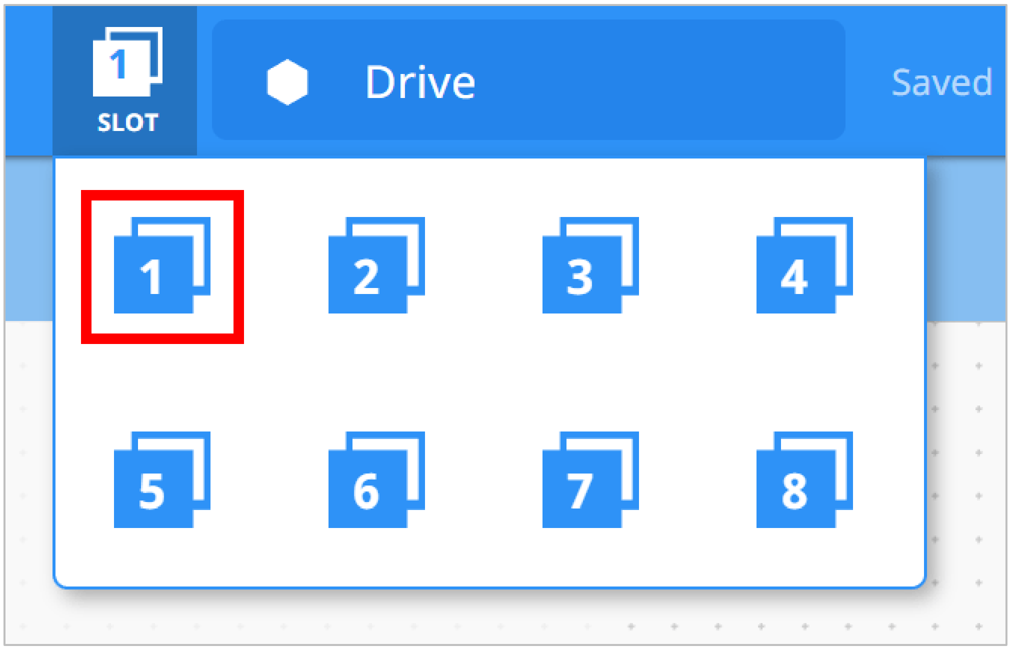
- Add the *drive for* block to the *when started* block in the programming area.



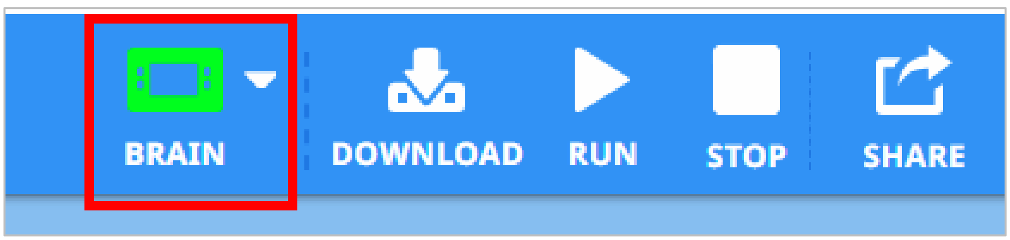
- Select the drop-down and change the units from inches to millimeters.



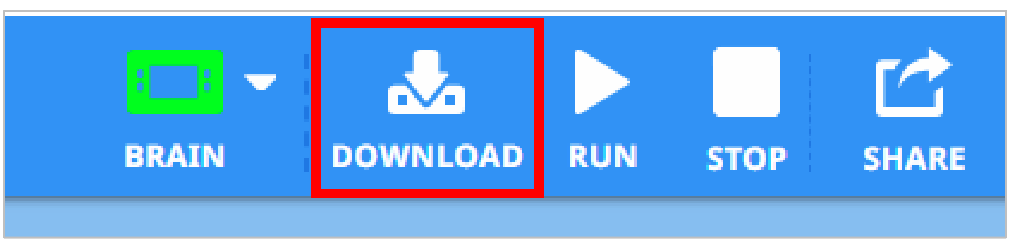
- Change the distance from 1 mm to 100 mm.



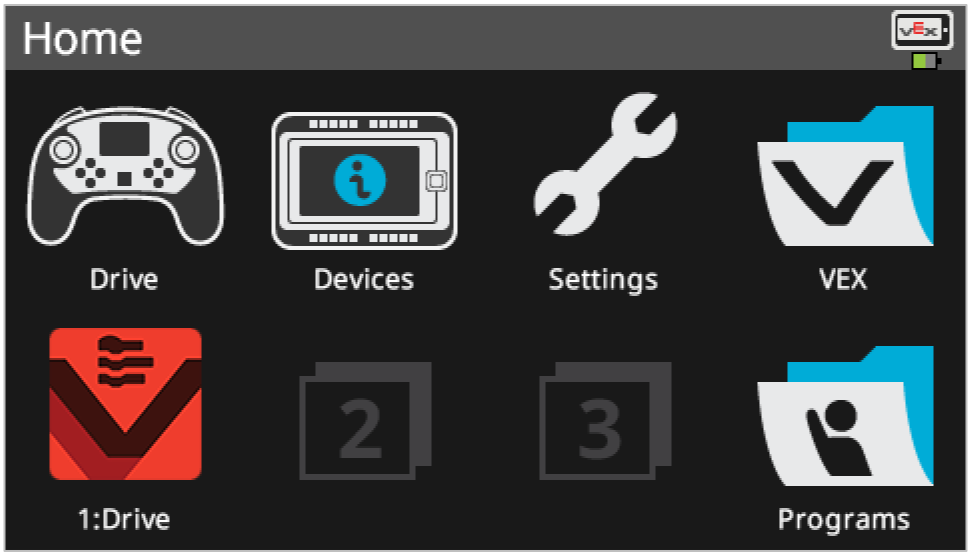
- Click on the **Slot** icon. You can download your project to one of the eight available slots in the Robot Brain. Click on the number **1**.



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



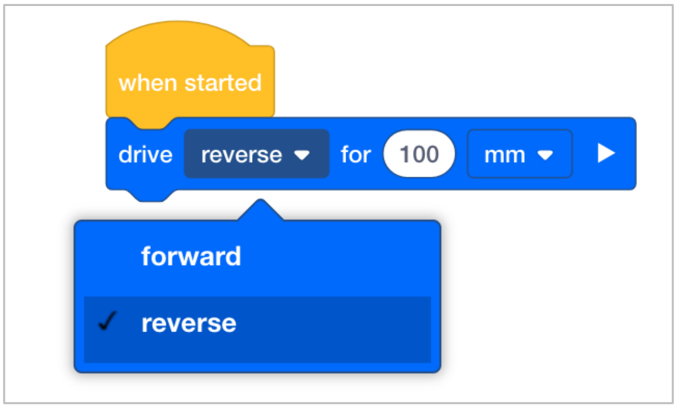
- Click the Download button on the toolbar to download the Drive project to the Robot Brain.



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

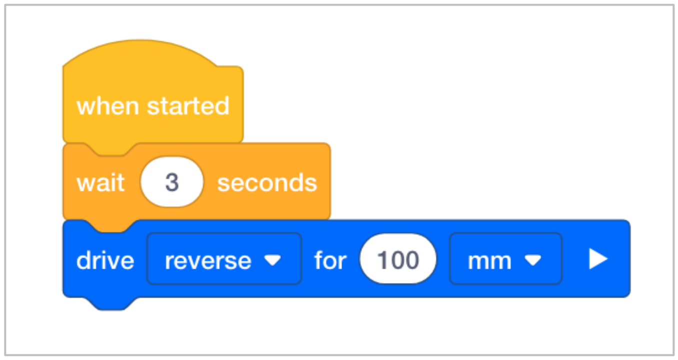
- Run the project on the Speedbot robot by making sure the project is selected and then press the Run button on the Robot Brain. Congratulations on creating your first project!

4. Drive Reverse



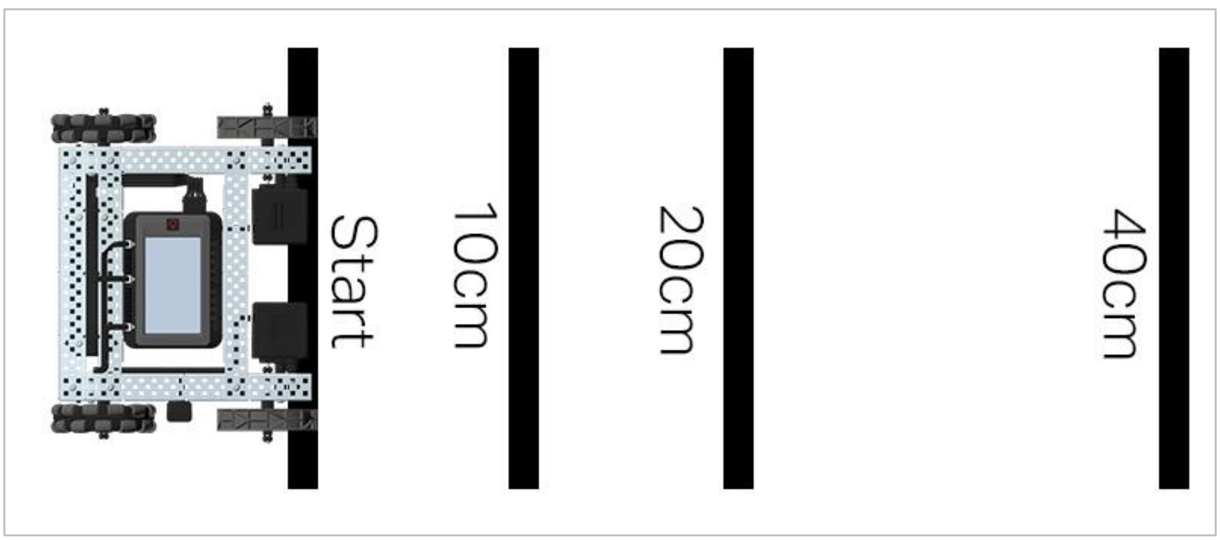
- Go back to the VEXCode V5 Blocks programming area. Change the `drive for block` to display **reverse** instead of **forward**.
- Download the project.
- Run the project on the Speedbot robot by making sure the project is selected and then press the Run button on the Robot Brain.

5. Wait then Drive in Reverse



- Go back to the VEXCode V5 Blocks programming area. Add a *wait* block before the *drive for* block. This will tell the robot to wait before driving in reverse.
- Insert three seconds into the *wait* block. This tells the robot to wait three before driving in reverse.
- Download the project.
- Run the project on the Speedbot robot by making sure the project is selected and then press the Run button.

6. Complete the Basketball Drills Challenge!



Basketball Drills Challenge layout

In the Basketball Drills Challenge, the robot must be able to navigate a series of lines at different distances. The robot will travel forward to the first line which is 10 cm from the starting position, wait 1 second, and then travel backwards returning to that same line. The robot will then repeat the action by driving forward 20 cm to the second line, wait 1 second, and then drive backward to the original starting line. The robot will drive forward to the third line with a distance of 40 cm, wait 1 second, and then finally return to the starting line to finish the challenge.

Before programming the robot, plan out the robot's path and behaviors in your engineering notebook.

After completing the Basketball Challenge, you will be able to combine forward and reverse movements with additional robot behaviors to complete even more advanced challenges.

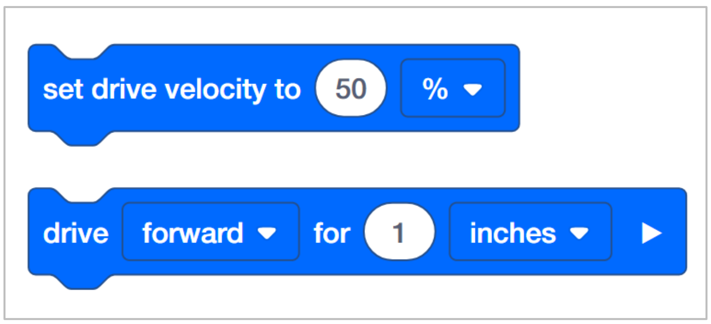
Keep in mind when programming that 1 cm = 10 mm

Exploring Velocity - VEXcode V5 Blocks

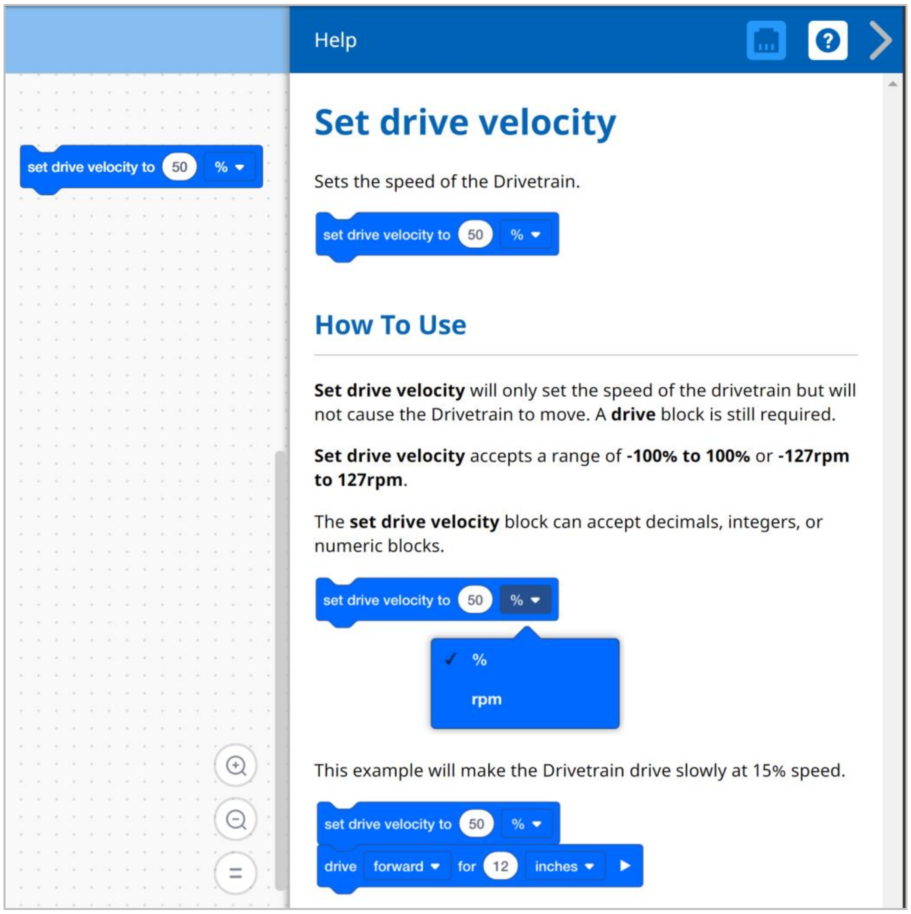
Speedbot is ready to drive at different velocities!

This investigation will help you to learn more about programming the Speedbot to drive at speeds that are best suited for the task. In the Strike Challenge at the end, you will need to find a velocity for the Speedbot that allows it to be fast and have great momentum but remain in control in order to hit the ball at a good angle and with great force.

VEXcode V5 Blocks that will be used in the first part of this investigation:



To find out more information about the block, open the Help and then select the block you want to read about.



Each group of students should get the hardware required and the group's engineering notebook. Then open VEXcode V5 Blocks.

Hardware/Software Required:

Quantity	Hardware/Other Items
1	Speedbot Robot
1	Charged Robot Battery
1	VEXcode V5 Blocks
1	USB Cable (if using a computer)
1	Engineering Notebook
1	Ball (the size and shape of a soccer ball)
1	3m x 3m clear space

Quantity	Hardware/Other Items
1	Meter stick or Ruler
1	Roll of tape
1	Data table

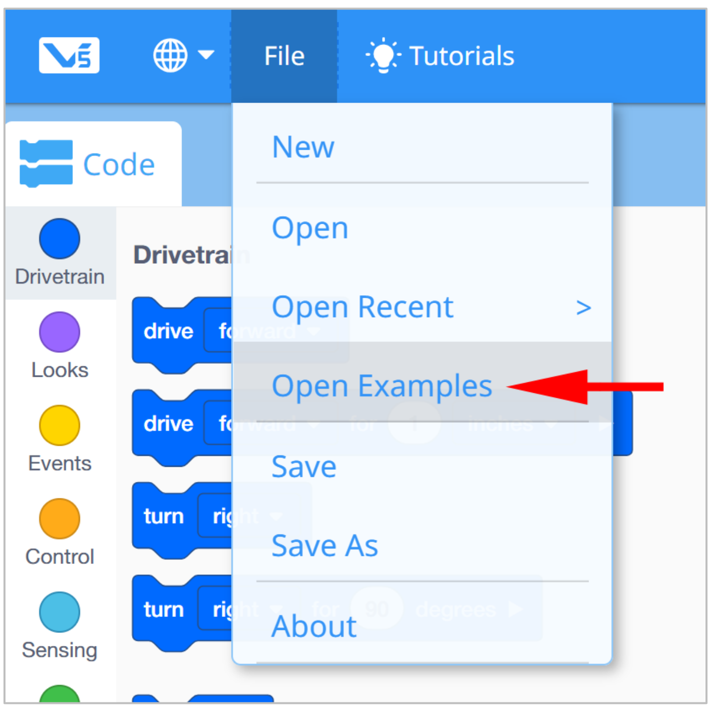
1. Preparing for the Exploration

Before you begin the activity, do you have each of these items ready?

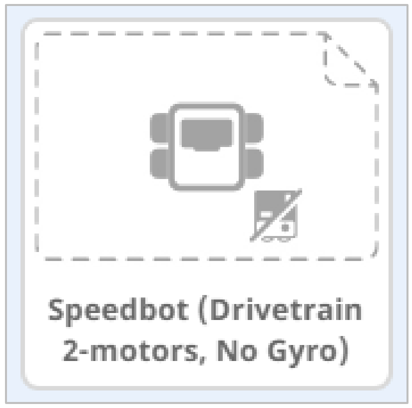
- Are all the motors plugged into the correct ports?
- Are the smart cables fully inserted into all of the motors?
- Is the Brain turned on?
- Is the battery charged?

2. Start a New Project

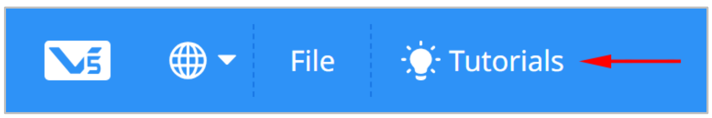
Before you begin your project, select the Speedbot (Drivetrain 2-motors, No Gyro) - Template project. The template project contains the Speedbot's motor configuration. If the template is not used, your robot will not run the project correctly.



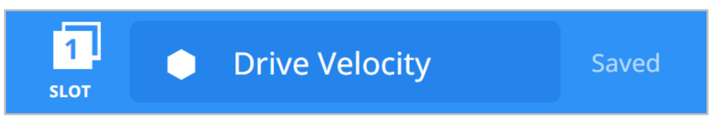
- Open the File menu.
- Select **Open Examples**.



- Select and open the **Speedbot (Drivetrain 2-motors, No Gyro) template project**.
- Rename your project *Drive Velocity* because we will be using the `set drive velocity` block.
- Save your project.
  - For help with saving a project, review the Tutorials inside VEXcode V5 Blocks.

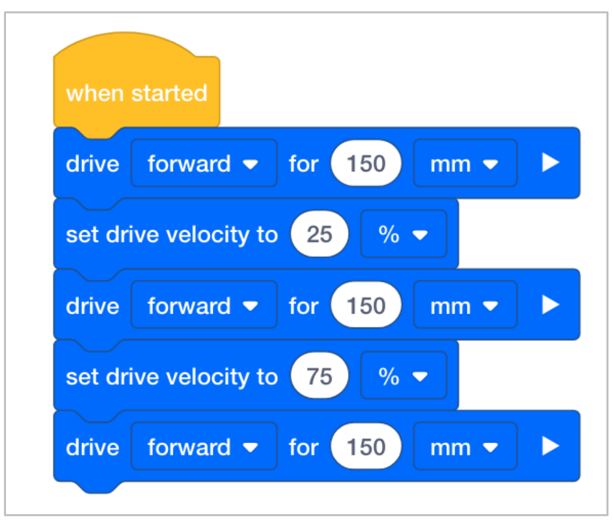


- Check to make sure the project name **Drive Velocity** is now in the window in the center of the toolbar.

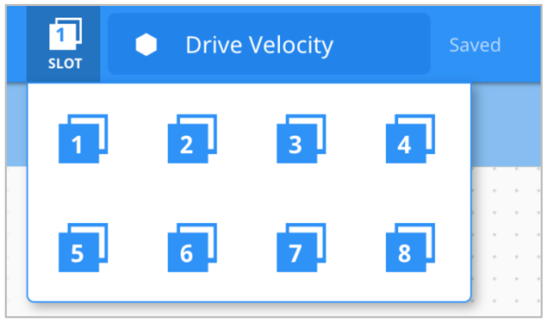


3. Drive Forward for 450 mm at Different Velocities

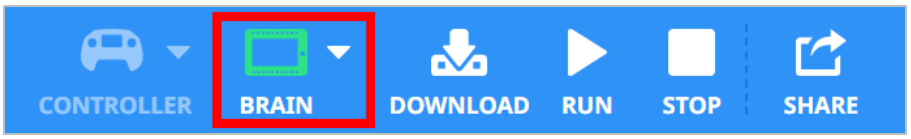
- Build this project in VEXcode V5 Blocks.



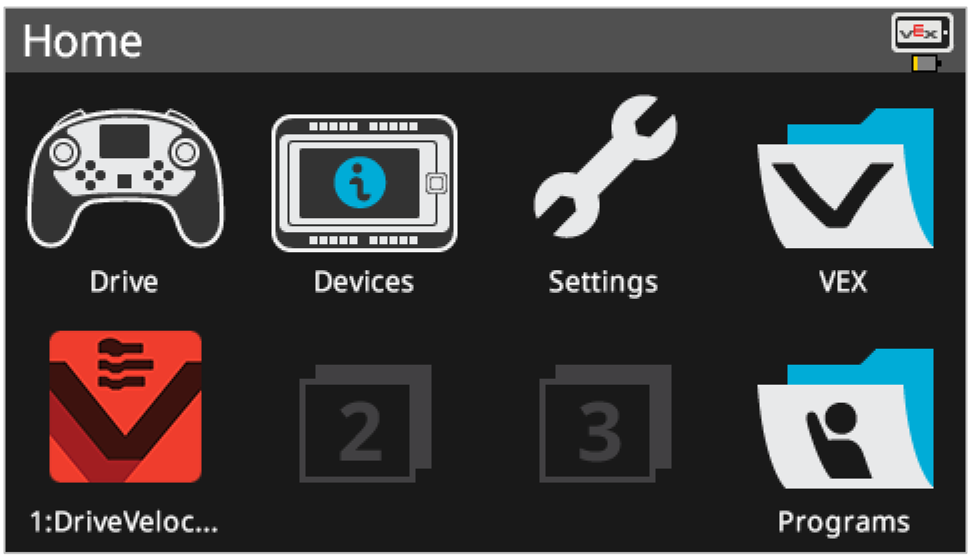
- Select on the Slot icon. You can download your project to one of the eight available slots in the Robot Brain. Select slot 1.



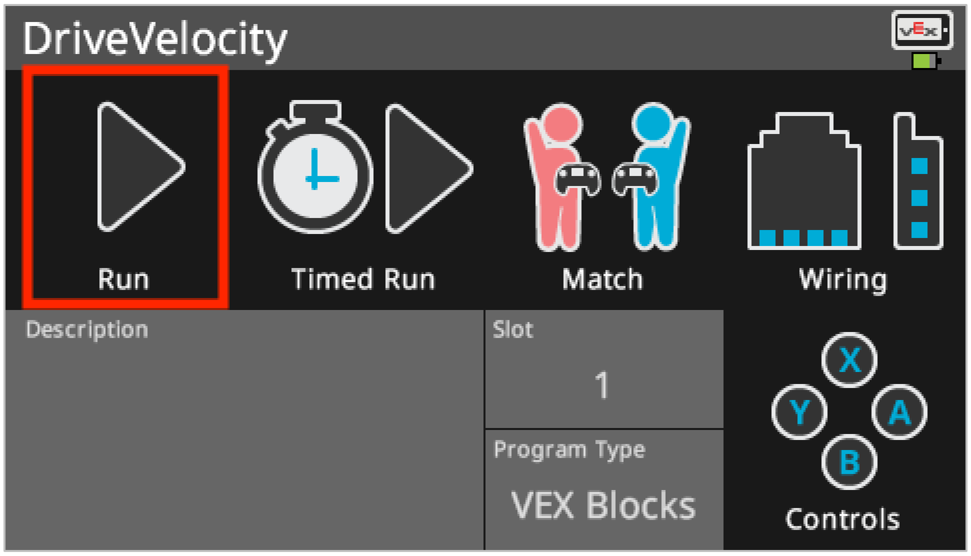
- Connect the robot to your computer or tablet. The Brain icon in the toolbar turns green once a successful connection has been made.



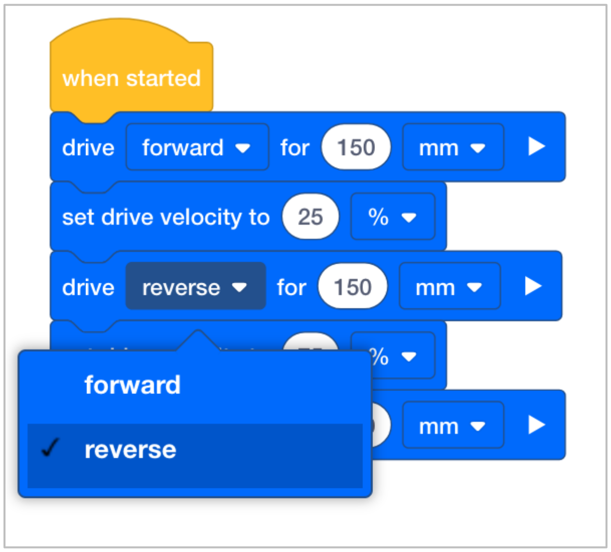
- Then, click the Download button on the toolbar to download the Drive Velocity project to the Robot Brain.



- Check to make sure your project has downloaded to the Speedbot's Brain by looking at the Robot Brain's screen. The project name should be listed in Slot 1.
- Run the project on the Speedbot robot by selecting the project and then pressing Run.

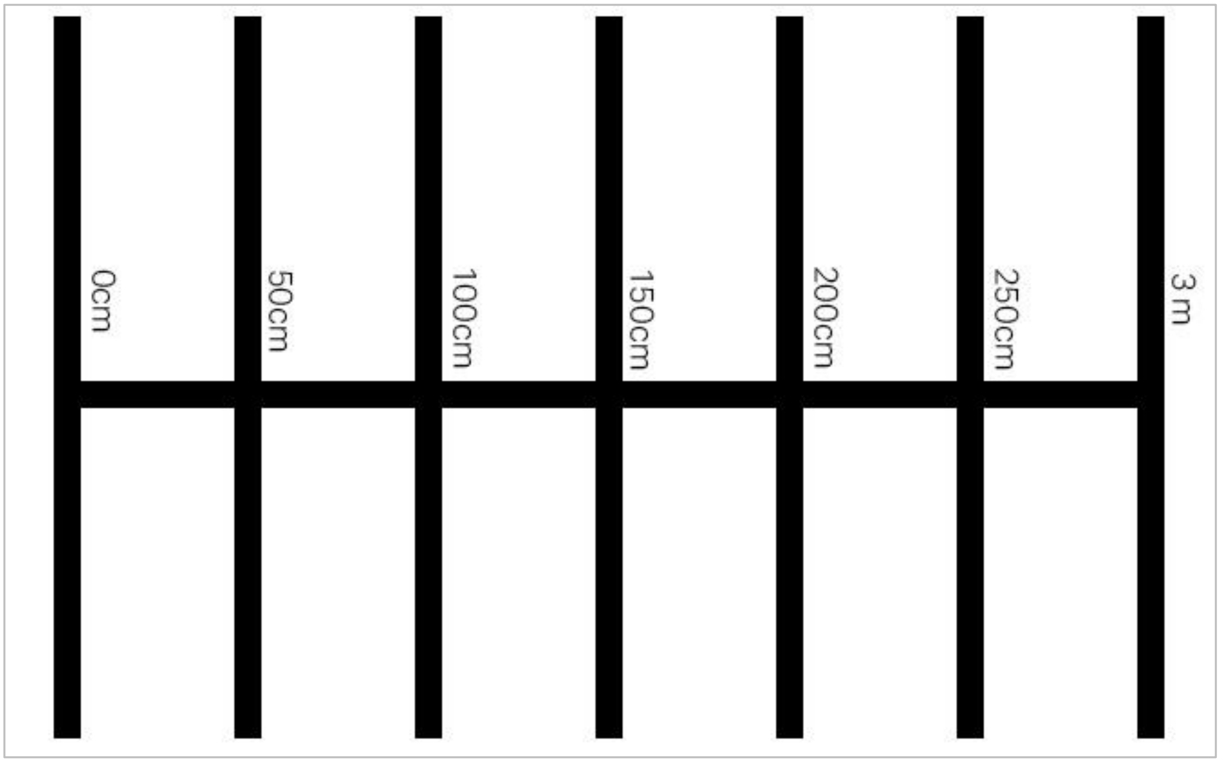


4. Drive Forward and in Reverse at Different Velocities



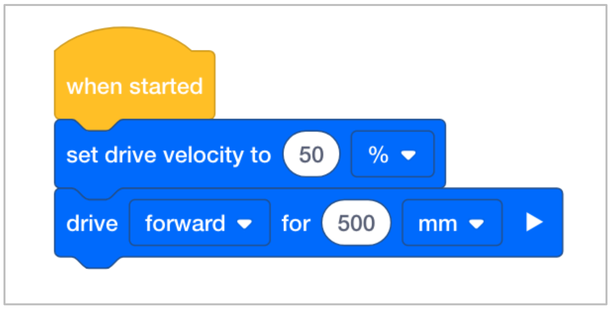
- Change the second `drive for` block to drive in reverse instead of forward.
- Then download the project.
- Run the project on the Speedbot robot.

5. Setting Up Your Testing Area

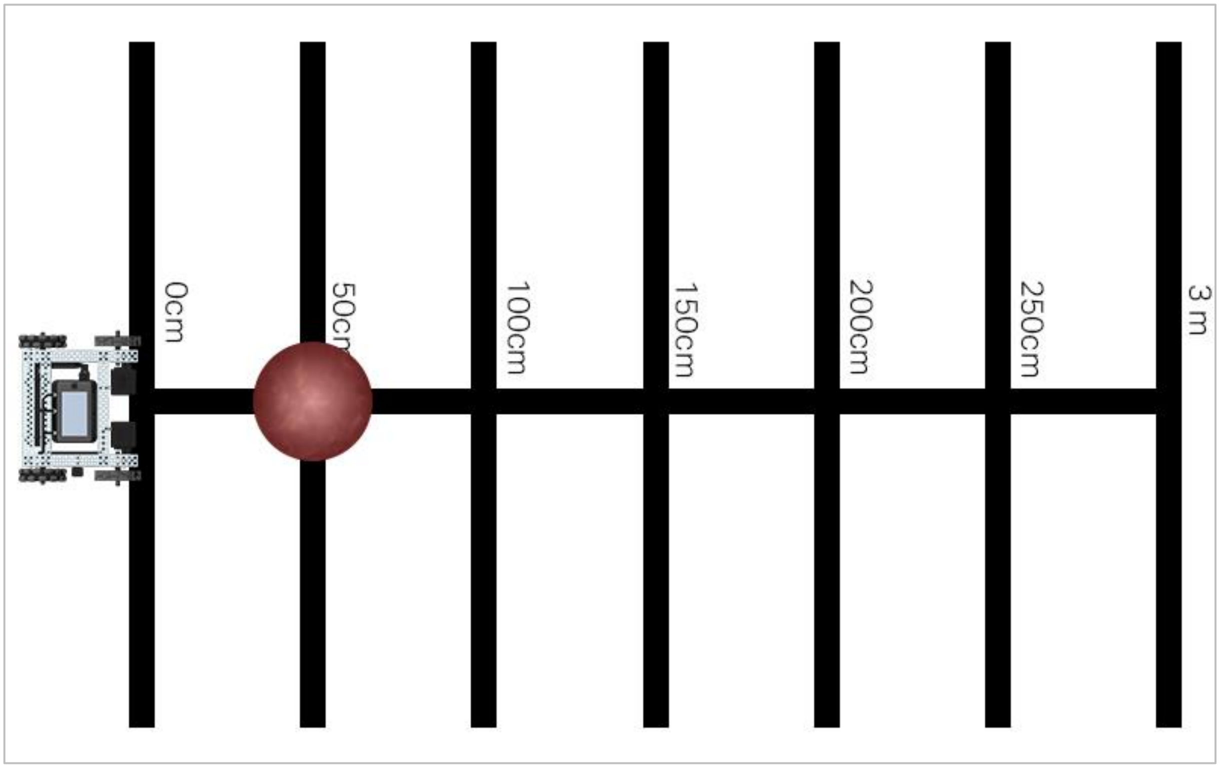


Example testing area layout

- Use tape and a meter stick to create a 3m line on the floor like the horizontal line shown in the image above.
  - After the line is created, use tape and your meter stick once more to create 1m lines across the 3m line like the vertical lines in the image above. Tape a 1m line at every 50cm mark on the vertical line by starting at 0cm.
  - The shorter horizontal lines should be centered on the longer vertical line.
- While the area is being set up, one or two members of your team should create a new project named Momentum. Set the velocity at 50% and have the Speedbot drive forward to the first line at 50 cm. Keep in mind 1 cm = 10 mm, so the robot will travel forward for 50 cm or 500 millimeters.



6. Testing the Transfer of Energy during Collisions



Bowling challenge test area with robot and ball

Center the ball on the horizontal line at 50cm and place your robot so that the front of it is centered on the horizontal line at 0cm. Make sure the front of the robot is facing the direction of the ball. Run your first Momentum project that has the velocity set to 50% and pay close attention as the robot collides with the ball.

Record the set velocity, the distance driven, and the distance the ball traveled in this table. The first row of the table has been started for you based on the Momentum project you worked on in the previous step. Continue to add data to this table as you try setting different velocities. You can then add other teams' data as you discuss your findings as a class.

Distance Driven by Speedbot	Set Velocity of Speedbot	Distance Traveled by the Ball
500mm	50%	
500mm		
500mm		

Think about and respond to the questions below in your engineering notebook as you collect your data:

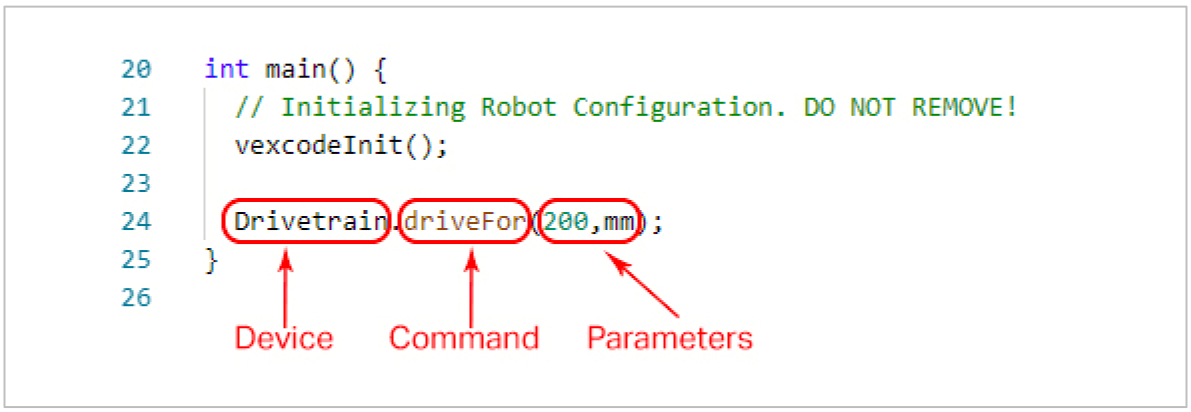
- How can you tell that the momentum of the robot transferred energy to the ball during the collision? Explain with details.
- Repeat the test at least twice more. Try a velocity less than 50%. Reset the ball in its position and record in the table how far the ball travels. Also, try a velocity more than 50%. Reset the ball in its position and record in the table how far the ball travels.
- When all groups have completed their three tests, discuss the velocities that the other groups chose and how far the ball traveled in their tests. As teams share their data, add their findings to your table.
- Look for pattern(s) in the data. Does the distance traveled by the ball increase or decrease as the set velocity increases?

Introduction to Text-Based Programming

What is Text-Based Programming?

In order to control your robot, you will be creating projects in a text-based programming language called VEXcode V5 Text. This is a language based on C++ and it uses text and special syntax to write the instructions which ultimately tell the robot what to do. If you worked with block-based programming before, such as VEXcode V5 Blocks, these text instructions are replacing the blocks.

An instruction is an entire line within the project. The instruction can contain portions of information for the device, the command, the parameters. The image below shows an instruction with each of these portions outlined.



It's important to remember specific rules for writing instructions in VEXcode V5 Text. For example, capitalization has specific rules within instructions. Devices are capitalized, but the commands must be in camel case.

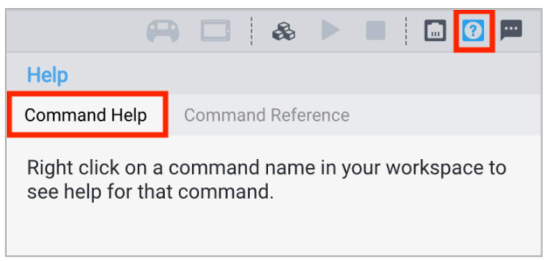
For more information on how to format the grammar and syntax, click here. It's also helpful to remember that there is help available within VEXcode V5 Text. Review how to access this help by clicking here.

Programming Drive Forward and Reverse - VEXcode V5 Text

The Speedbot is ready to move!

This exploration will give you the tools to be able to program your Speedbot for simple movements. At the end of this activity, you will engage in the Basetball Drills Challenge using forward, reverse, and waiting behaviors.

- VEXcode V5 Text instructions that will be used in this exploration:
  - **Drivetrain.driveFor(1, inches);**
  - **wait(1, seconds);**
- To find out more information about the instruction, select Help and then select Command Help. For more information on the help feature of VEXcode V5 Text click here.



- Make sure you have the hardware required, your engineering notebook, and VEXcode V5 Text downloaded and ready.

Hardware/Software Required:

Amount	**Hardware/Other Items**
1	Speedbot Robot
1	Charged Robot Battery
1	VEXcode V5 Text
1	USB Cable (if using a computer)
1	Engineering Notebook

1. Preparing for the Exploration

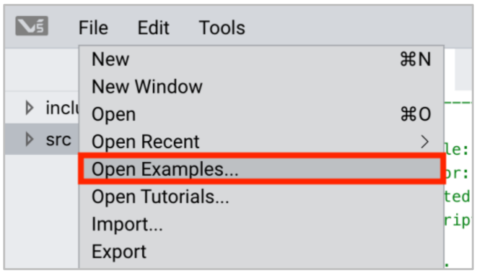
Before you begin the activity, do you have each of these items ready? Check each of the following:

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

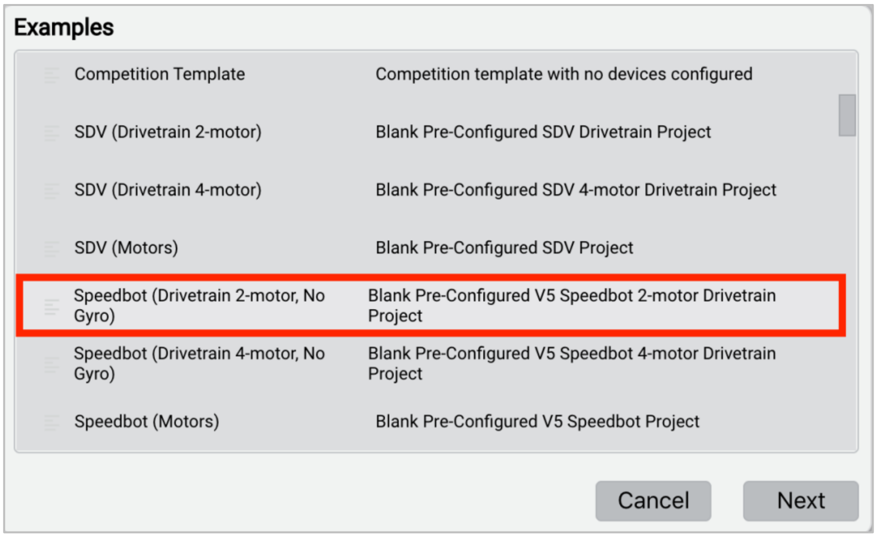
2. Start a New Project

Complete the following steps to begin the project:

- Open the File menu and select Open Examples.



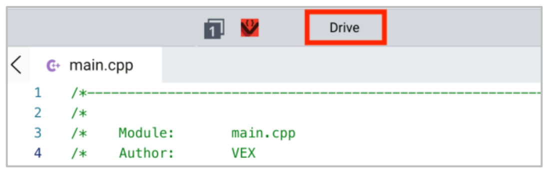
- Select and open the Speedbot (Drivetrain 2-motor, No Gyro) template project. The template project contains the Speedbot's motor configuration. If the template is not used, your robot will not run the project correctly.



- Since you will be working on moving the Speedbot forward and reverse, you will name your project **Drive**.

- When finished, select **Create**.

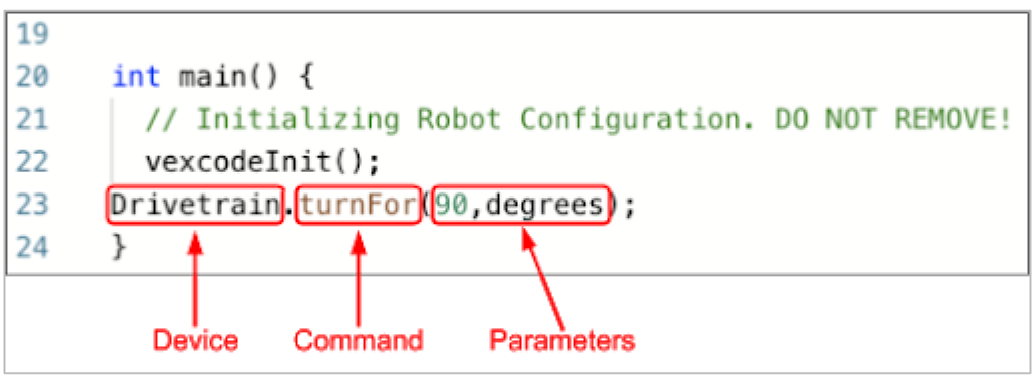
- Check to make sure the project name **Drive** is now in the window in the center of the toolbar.



3. Drive Forward

You are now ready to begin programming the robot to drive forward!

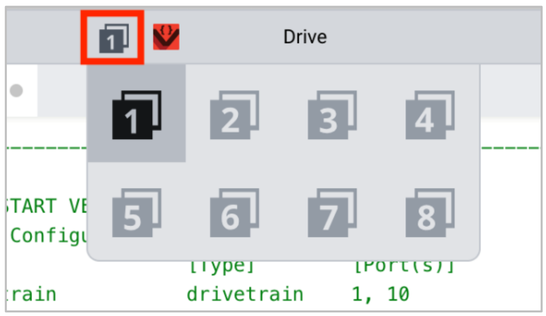
- Before we begin programming, we need to understand what an instruction is. There are three parts to an instruction. For more information on what an instruction is, view the article, "Syntax Rules and Guidelines - VEXcode V5 Text."



- Add the instruction to the project:

```
#include "vex.h"

using namespace vex;

int main() {
  // Initializing Robot Configuration. DO NOT REMOVE!
  vexcodeInit();

  Drivetrain.driveFor(100,mm);

}
```

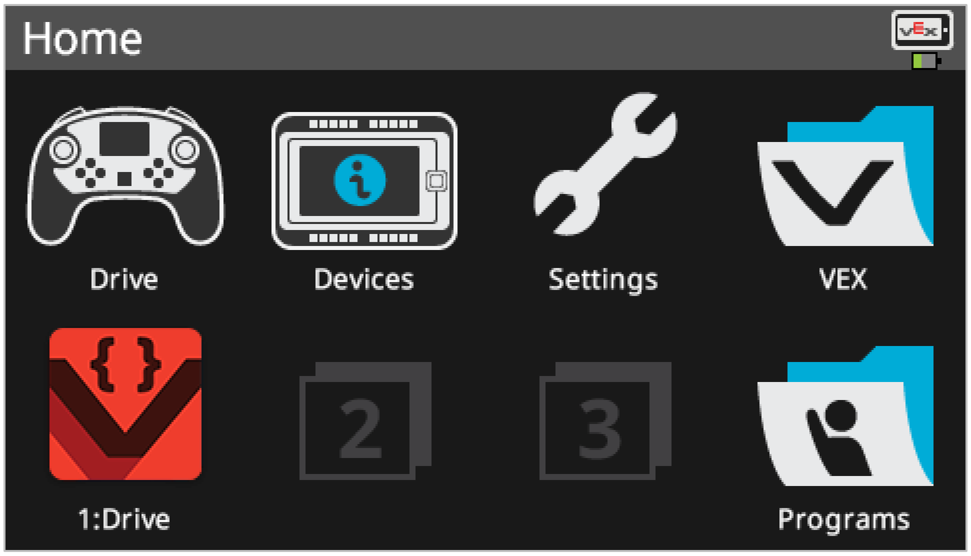
- Select the Slot icon to choose one of the eight available slots on the Robot Brain and select slot 1.



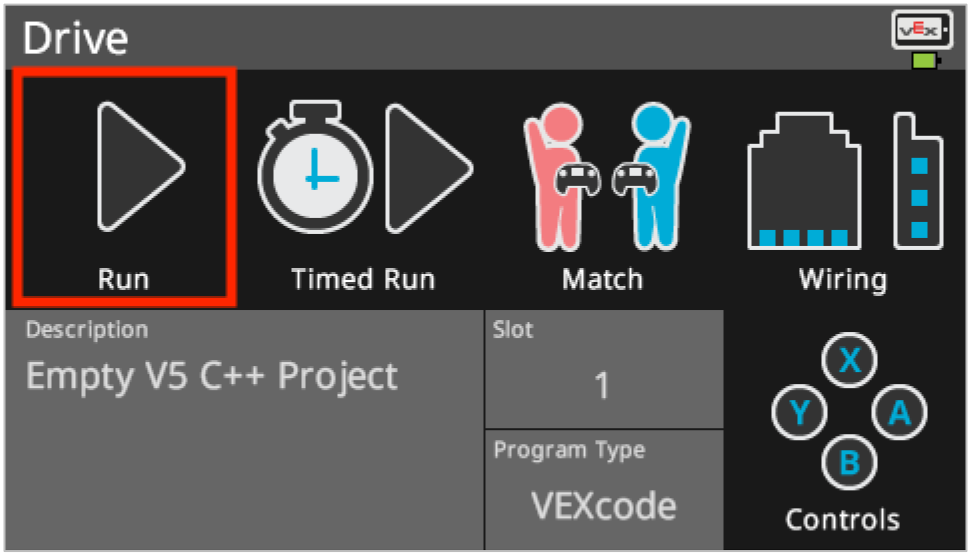
- Connect the V5 Robot Brain to the computer using a micro USB cable and power on the V5 Robot Brain. The Brain icon in the toolbar turns green once a successful connection has been made.

- When the V5 Robot Brain is connected to the computer, the **Build** icon changes to the **Download** icon. Select **Download** to download the project to the Brain.

- Check to make sure your project has downloaded by looking at the Robot Brain's screen. The project name Drive should be listed in Slot 1.



- Run the project on the robot by making sure the project is selected and then press the **Run** button on the Robot Brain. Congratulations on creating your first project!



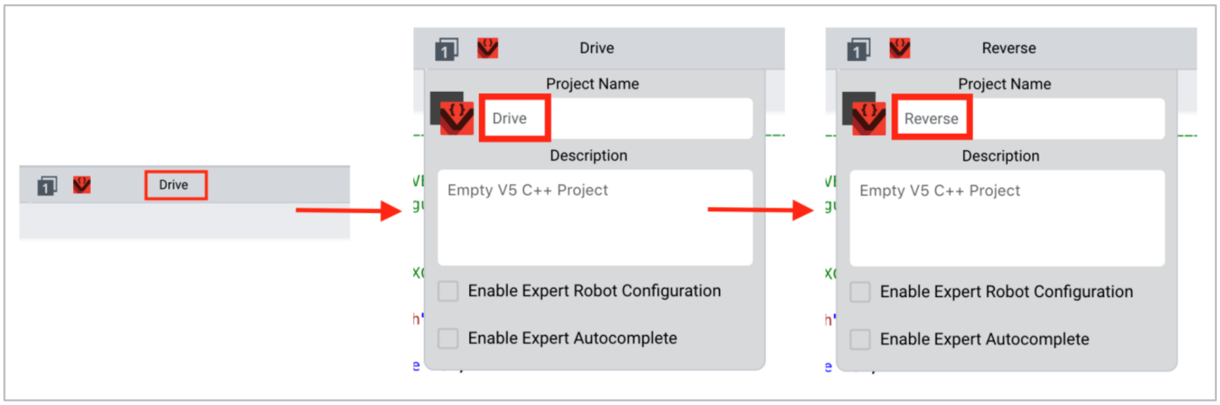
4. Drive Reverse

Now that you have programmed your robot to drive forward, let us program it to now drive in reverse.

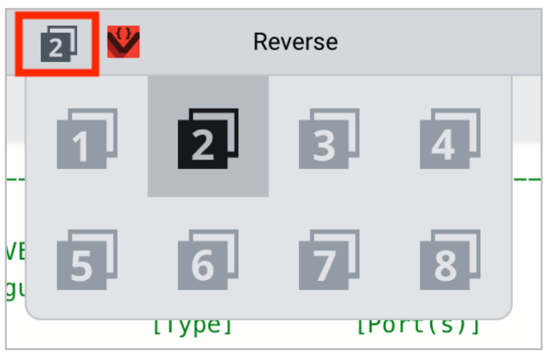
- Change the parameter in the drive instruction to display -100.

```
#include "vex.h"

using namespace vex;

int main() {
  // Initializing Robot Configuration. DO NOT REMOVE!
  vexcodeInit();

  Drivetrain.driveFor(-100,mm);

}
```

- Select the Project Name to change it from Drive to Reverse.

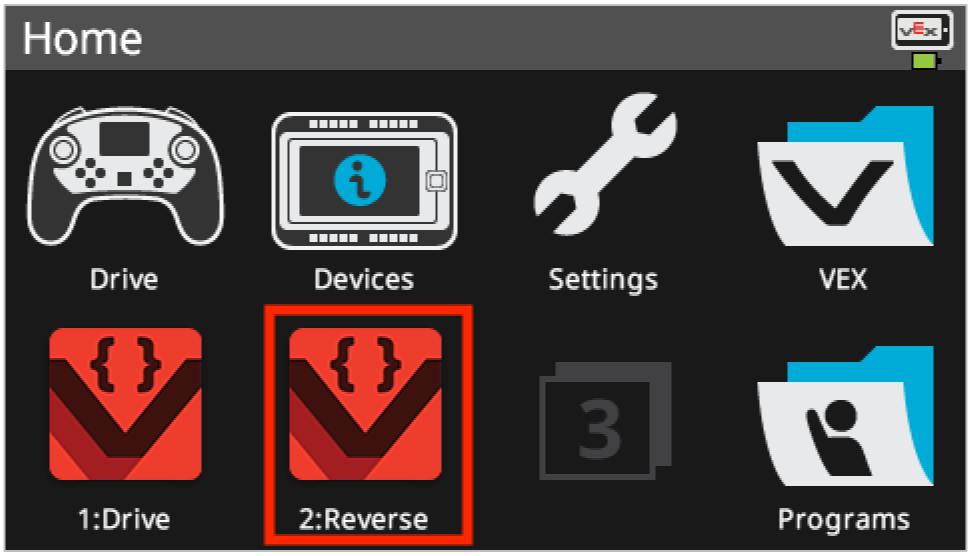


- Select the Slot icon to choose a new slot. Select slot 2.

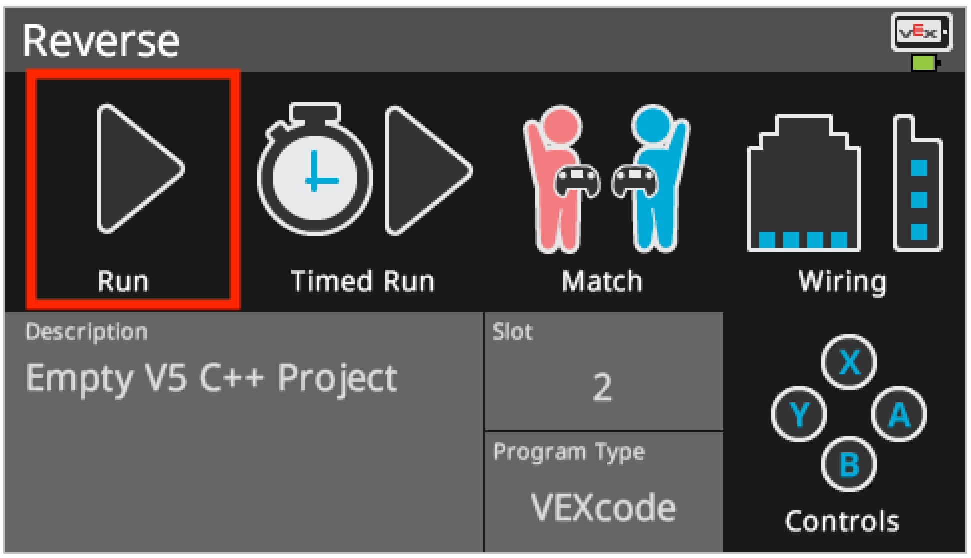


- Download the project.

- Check to make sure your project has downloaded by looking at the Robot Brain’s screen. The project name Reverse should be listed in Slot 2.



- Run the project on the robot by making sure the project is selected and then press the **Run** button on the Robot Brain.



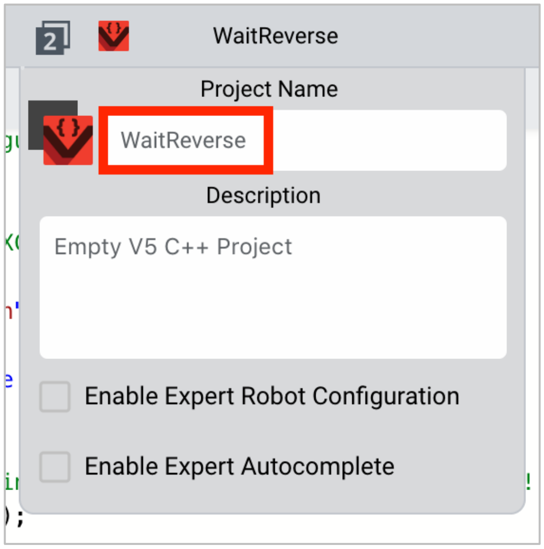
5. Wait then Drive in Reverse

Now that we have programmed the robot to drive forward and then in reverse, we can now add a wait instruction so that the robot waits a certain amount of time before driving in reverse.

- Add a wait instruction as shown. This tells the robot to wait three seconds before driving in reverse.

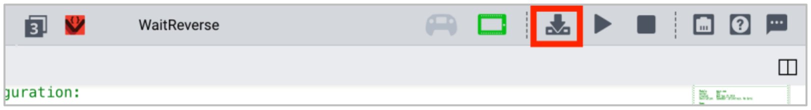
```
int main() {
  // Initializing Robot Configuration. DO NOT REMOVE!
  vexcodeInit();

  wait(3,seconds);

  Drivetrain.driveFor(-100,mm);

}
```

- Select the Project Name to change it from Reverse to WaitReverse.

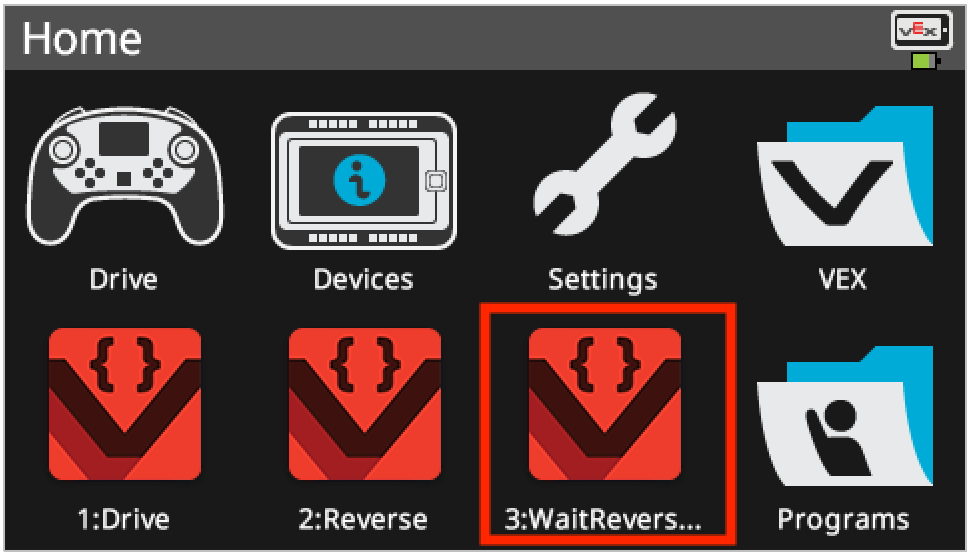


- Select the Slot icon to choose a new slot. Select slot 3.

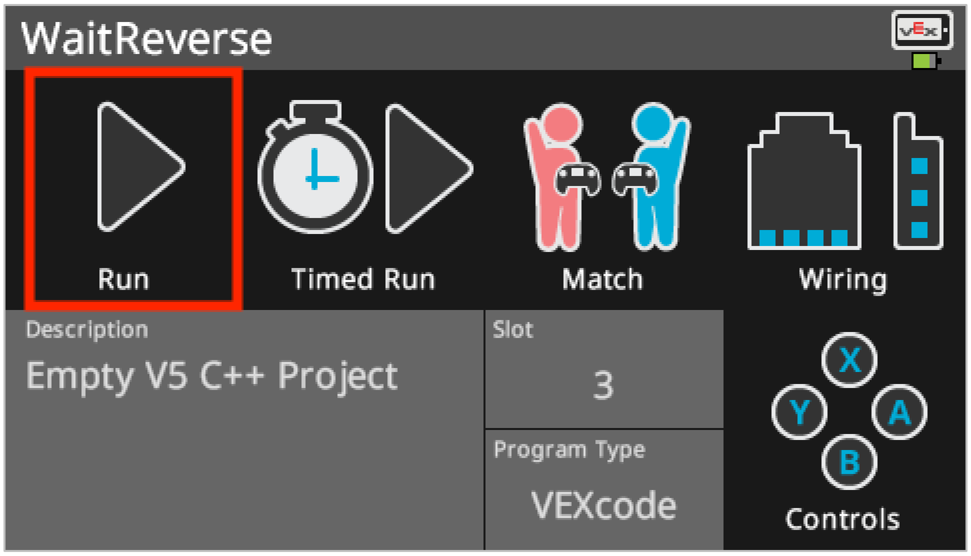
- Download the project.



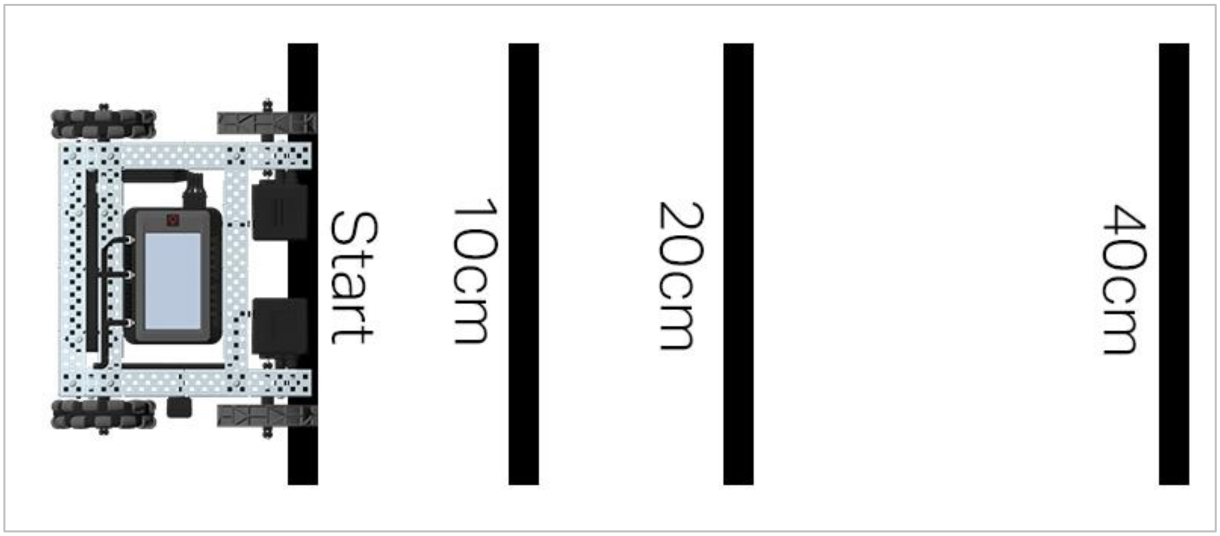
- Check to make sure your project has downloaded by looking at the Robot Brain's screen. The project name WaitReverse should be listed in Slot 3.



- Run the project on the robot by making sure the project is selected and then press the **Run** button.



6. Complete the Basketball Drills Challenge!



Basketball Drills Challenge layout

In the Basketball Drills Challenge, the robot must be able to navigate a series of lines at different distances. The robot will travel forward to the first line which is 10 cm from the starting position, wait 1 second, and then travel backwards returning to that same line. The robot will then repeat the action by driving forward 20 cm to the second line, waiting 1 second, and then driving backward to the original starting line. The robot will drive forward to the third line with a distance of 40 cm, wait 1 second, and then finally return to the starting line to finish the challenge.

Before programming the robot, plan out the robot's path and behaviors in your engineering notebook.

After completing the Basketball Challenge, you will be able to combine forward and reverse movements with additional robot behaviors to complete even more advanced challenges. Keep in mind when programming that 1 cm = 10 mm.

Exploring Velocity - VEXcode V5 Text

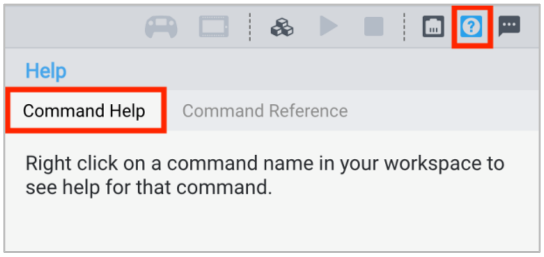
Speedbot is ready to drive at different velocities!

This investigation will help you to learn more about programming the Speedbot to drive at speeds that are best suited for the task. In the Strike Challenge at the end, you will need to find a velocity for the Speedbot that allows it to be fast and have great momentum but remain in control in order to hit the ball at a good angle and with great force.

VEXcode V5 Text instructions that will be used in the first part of this investigation:

- **Drivetrain.setDriveVelocity(50, percent);**
- **Drivetrain.driveFor(1, inches);**

- To find out more information about the instruction, select Help and then select Command Help. For more information on the help feature of VEXcode V5 Text click here.



- Make sure you have the hardware required, your engineering notebook, and VEXcode V5 Text downloaded and ready.

Each group of students should get the hardware required and the group's engineering notebook. Then open VEXcode V5 Text.

Hardware/Software Required:

Quantity	Hardware/Other Items
1	Speedbot Robot
1	Charged Robot Battery
1	VEXcode V5 Text
1	USB Cable (if using a computer)

Quantity	Hardware/Other Items
1	Engineering Notebook
1	Ball (the size and shape of a soccer ball)
1	3m x 3m clear space
1	Meter stick or Ruler
1	Roll of tape
1	Data table

1. Preparing for the Exploration

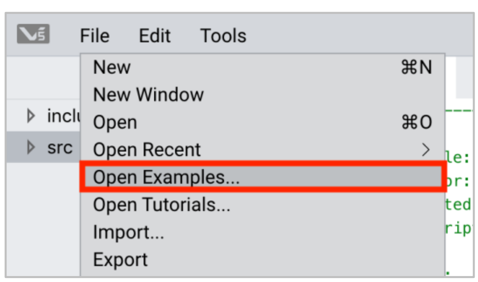
Before you begin the activity, do you have each of these items ready?

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

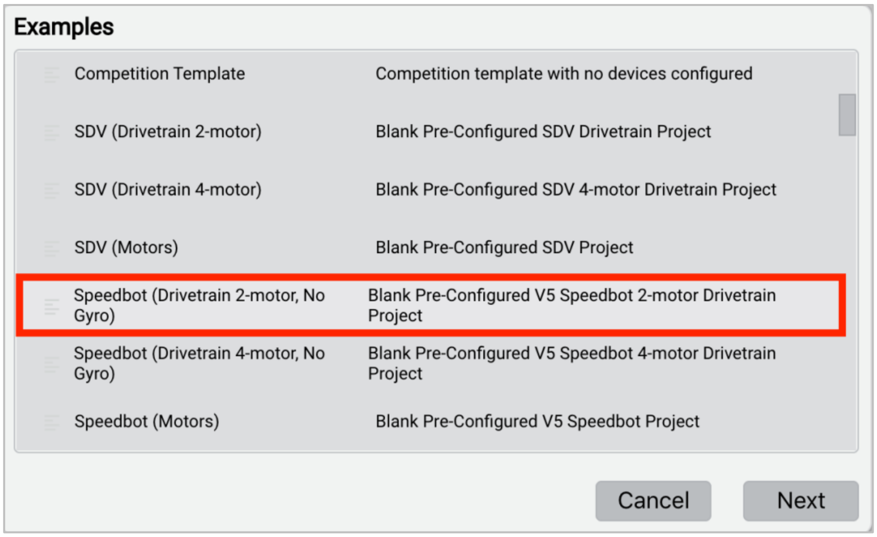
2. Start a New Project

Complete the following steps to begin the project:

- Open the File menu and select Open Examples.



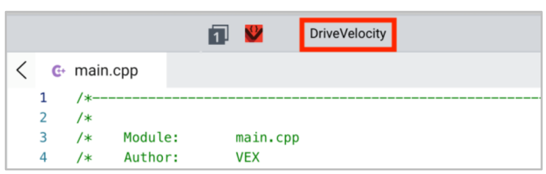
- Select and open the Speedbot (Drivetrain 2-motor, No Gyro) template project. The template project contains the Speedbot's motor configuration. If the template is not used, your robot will not run the project correctly.



- Since you will be working on exploring velocity, you will name your project **DriveVelocity**.

- When finished, select **Create**.

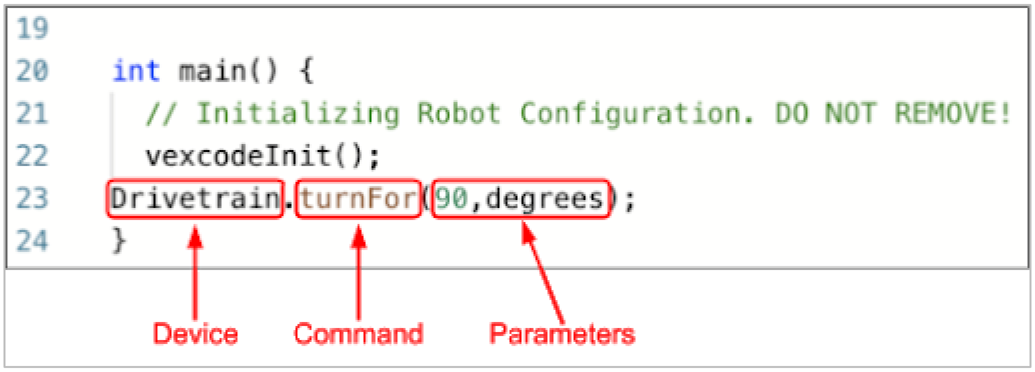
- Check to make sure the project name **Drive** is now in the window in the center of the toolbar.



3. Drive Forward for 150 mm at Different Velocities

You are now ready to begin programming the robot to drive forward at different velocities!

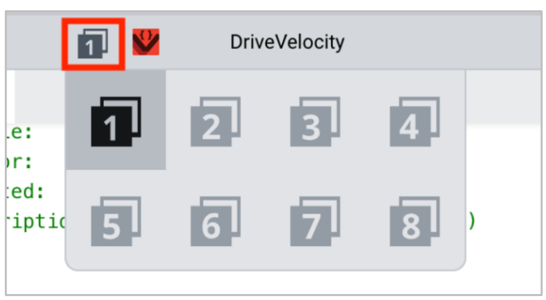
- Before we begin programming, we need to understand what an instruction is. There are three parts to an instruction. For more information on what an instruction is, view the article, "Syntax Rules and Guidelines - VEXcode V5 Text."



- Add the instructions to the project:

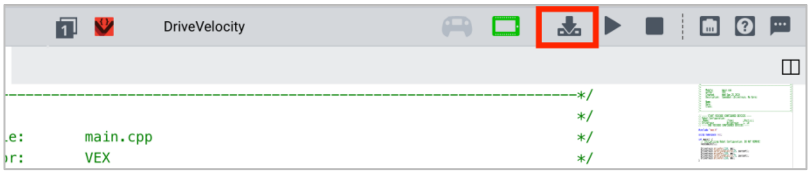
```
#include "vex.h"

using namespace vex;

int main() {
  // Initializing Robot Configuration. DO NOT REMOVE!
  vexcodeInit();

  Drivetrain.driveFor(150, mm);
  Drivetrain.setDriveVelocity(25, percent);
  Drivetrain.driveFor(150, mm);
  Drivetrain.setDriveVelocity(75, percent);
  Drivetrain.driveFor(150, mm);
}
```

- Select the Slot icon to choose one of the eight available slots on the Robot Brain and select slot 1.

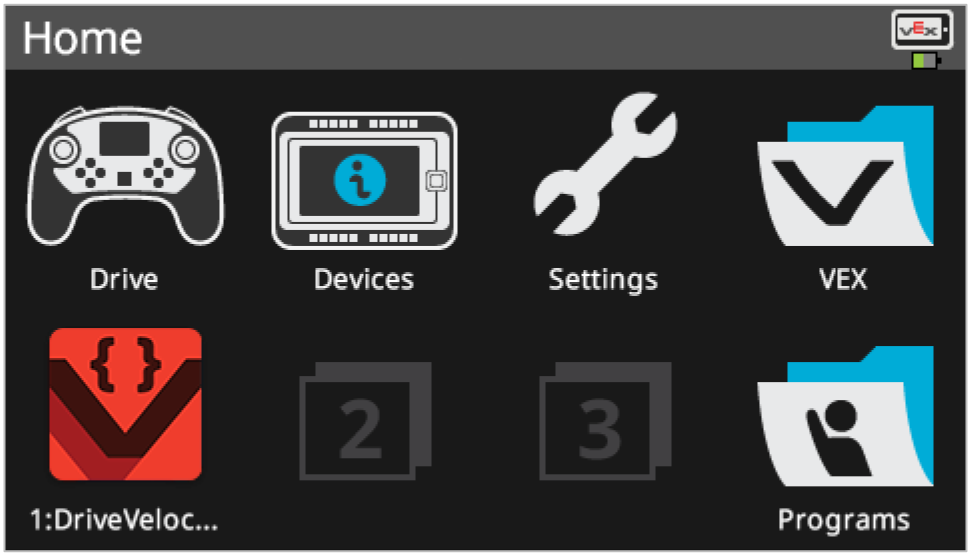


- Connect the V5 Robot Brain to the computer using a micro USB cable and power on the V5 Robot Brain. The Brain icon in the toolbar turns green once a successful connection has been made.

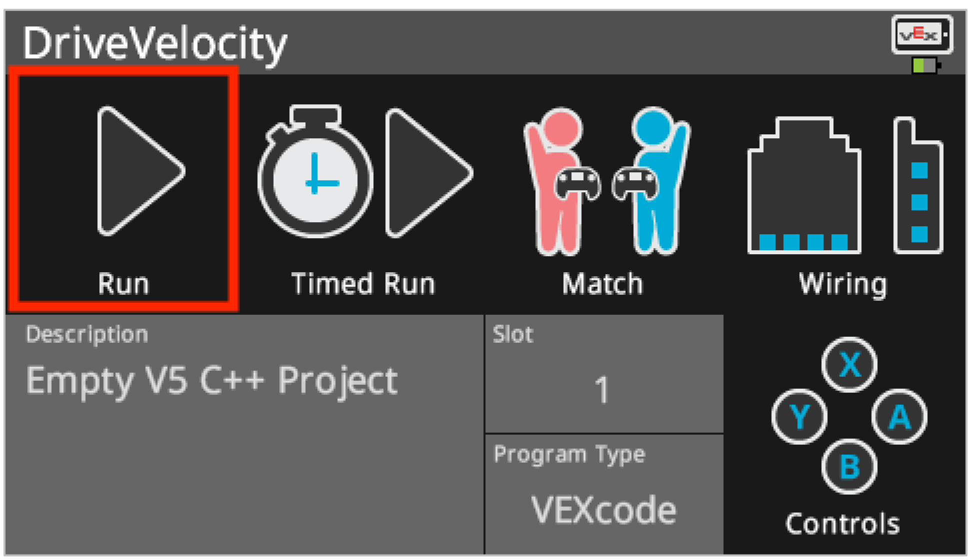
- Select **Download** to download the project to the Brain.



- Check to make sure your project has downloaded by looking at the Robot Brain's screen. The project name DriveVelocity should be listed in Slot 1.



- Run the project on the robot by making sure the project is selected and then press the **Run** button on the Robot Brain. Congratulations on creating your first project!



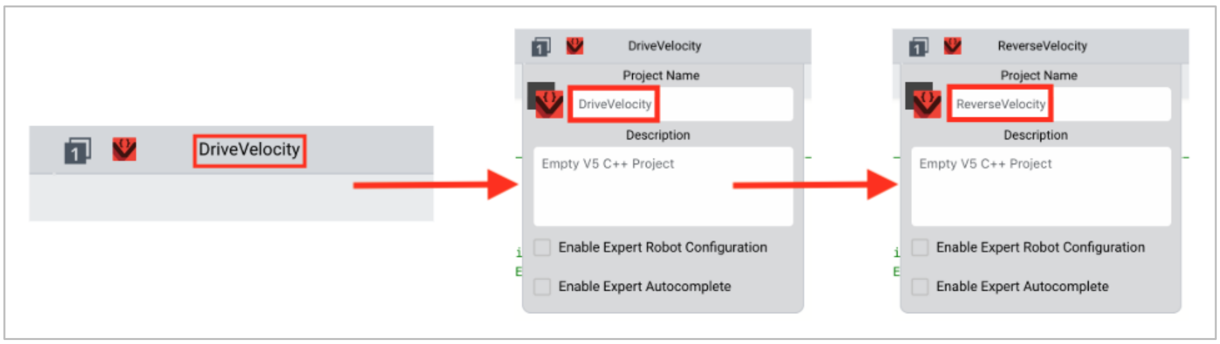
4. Drive Forward and Reverse for 150 mm at Different Velocities

Now that you have programmed your robot to drive forward at different velocities, program it to now drive forward and in reverse at different velocities.

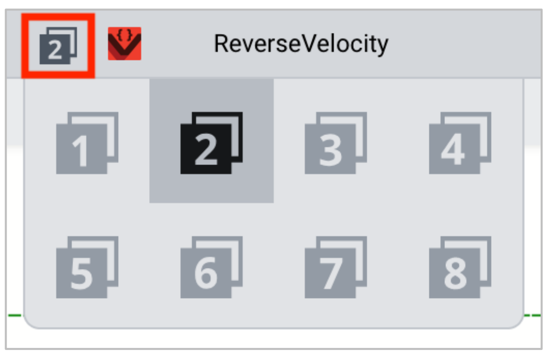
- Change the parameter in the second *driveFor* instruction to display -150.

```cpp
#include "vex.h"

using namespace vex;

int main() {
  // Initializing Robot Configuration. DO NOT REMOVE!
  vexcodeInit();

  Drivetrain.driveFor(150, mm);
  Drivetrain.setDriveVelocity(25, percent);
  Drivetrain.driveFor(-150, mm);
  Drivetrain.setDriveVelocity(75, percent);
  Drivetrain.driveFor(150, mm);
}
```

- Select the Project Name to change it from DriveVelocity to ReverseVelocity.

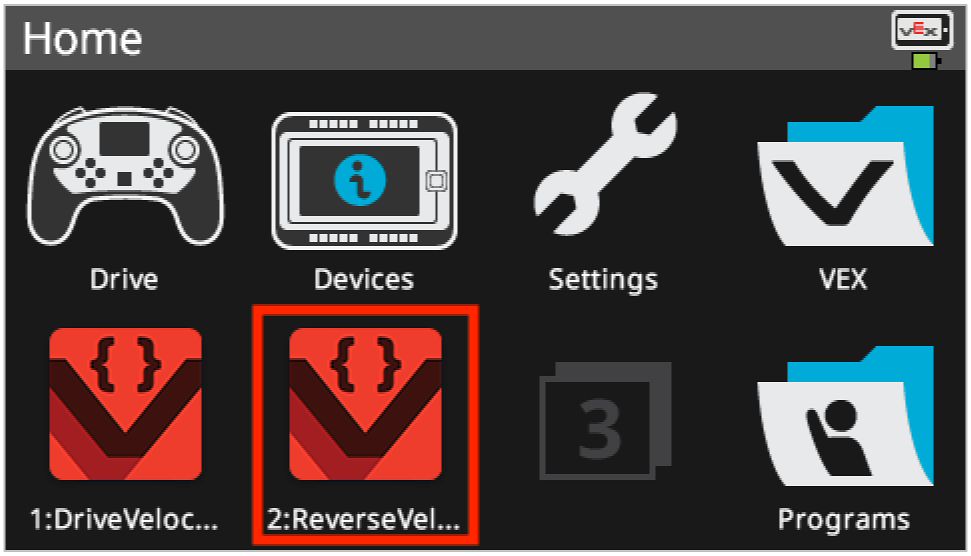


- Select the Slot icon to choose a new slot. Select slot 2.

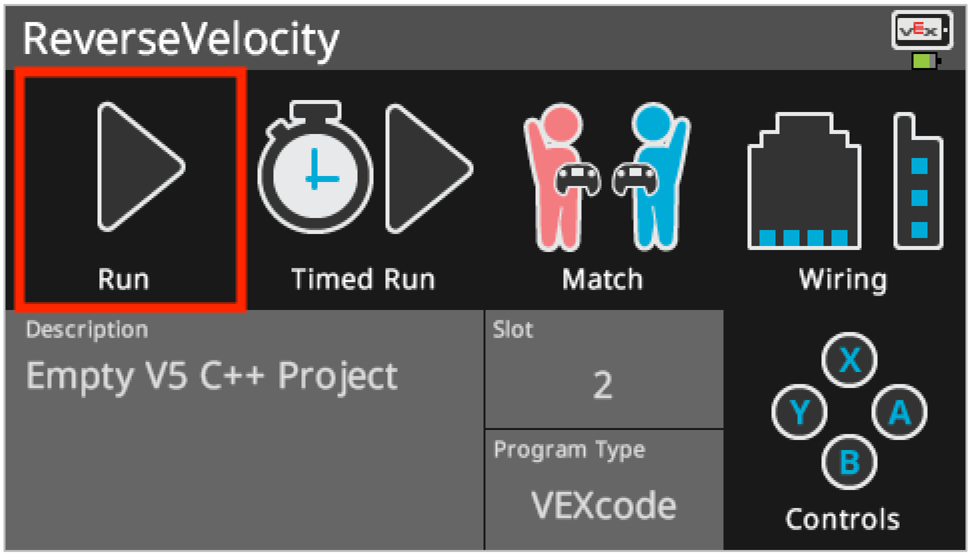


- Download the project.

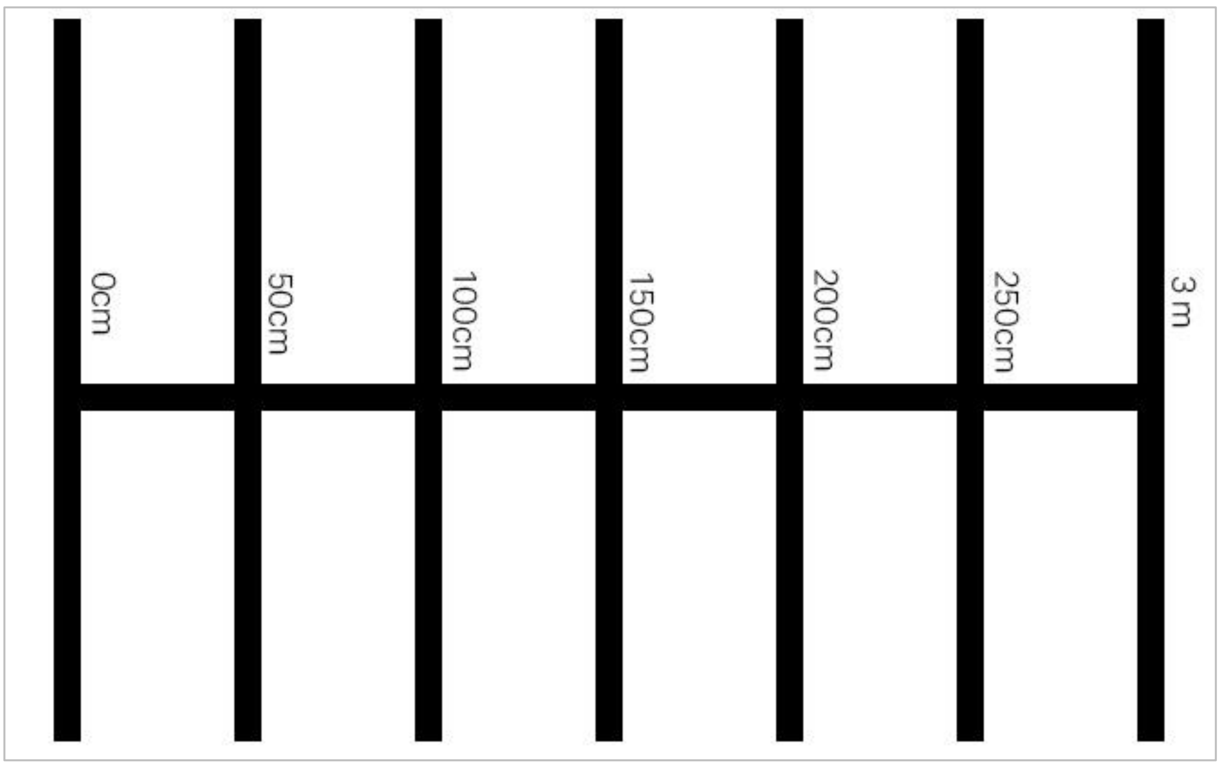
- Check to make sure your project has downloaded by looking at the Robot Brain's screen. The project name ReverseVelocity should be listed in Slot 2.



- Run the project on the robot by making sure the project is selected and then press the **Run** button on the Robot Brain.



5. Setting Up Your Testing Area



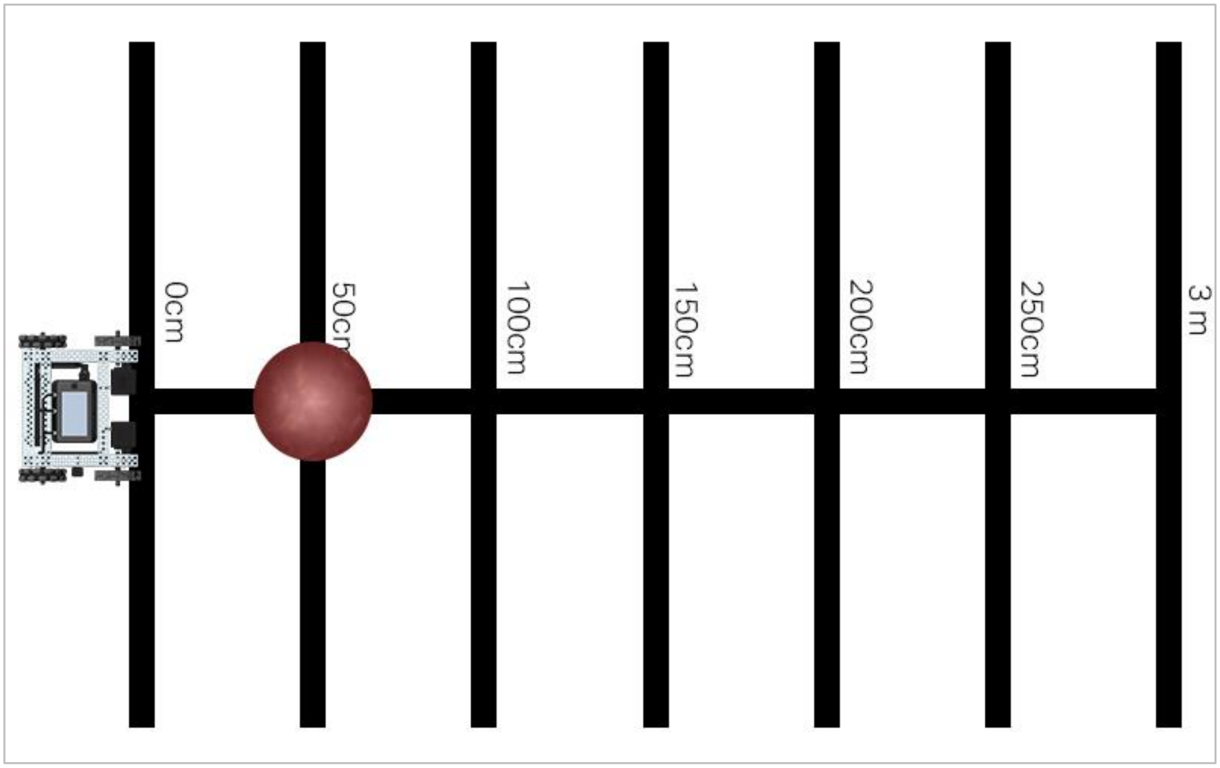
Example testing area layout

- Use tape and a meter stick, to create a 3m line on the floor like the horizontal line shown in the image above.

- After the line is created, use tape and your meter stick once more to create 1m lines across the 3m line like the vertical lines in the image above. Tape a 1m line at every 50cm mark on the vertical line by starting at 0cm.
- The shorter horizontal lines should be centered on the longer vertical line.

- While the area is being set up, one or two members of your team should create a new project named Momentum. Set the velocity at 50% and have the Speedbot drive forward to the first line at 50 cm. Keep in mind 1 cm = 10 mm, so the robot will travel forward for 50 cm or 500 millimeters.

6. Testing the Transfer of Energy during Collisions



Bowling challenge test area with robot and ball

Center the ball on the horizontal line at 50cm and place your robot so that the front of it is centered on the horizontal line at 0cm. Make sure the front of the robot is facing the direction of the ball. Run your first Momentum project that has the velocity set to 50% and pay close attention as the robot collides with the ball.

Record the set velocity, the distance driven, and the distance the ball traveled in this table. The first row of the table has been started for you based on the Momentum project you worked on in the previous step. Continue to add data to this table as you try setting different velocities. You can then add other teams' data as you discuss your findings as a class.

Distance Driven by Speedbot	Set Velocity of Speedbot	Distance Traveled by the Ball
500mm	50%	
500mm		
500mm		

Think about and respond to the questions below in your engineering notebook as you collect your data:

- How can you tell that the momentum of the robot transferred energy to the ball during the collision? Explain with details.
- Repeat the test at least twice more. Try a velocity less than 50%. Reset the ball in its position and record in the table how far the ball travels. Also, try a velocity more than 50%. Reset the ball in its position and record in the table how far the ball travels.
- When all groups have completed their three tests, discuss the velocities that the other groups chose and how far the ball traveled in their tests. As teams share their data, add their findings to your table.
- Look for pattern(s) in the data. Does the distance traveled by the ball increase or decrease as the set velocity increases?

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

Car Safety

Driving a vehicle

Slowing Down the Effects of Momentum

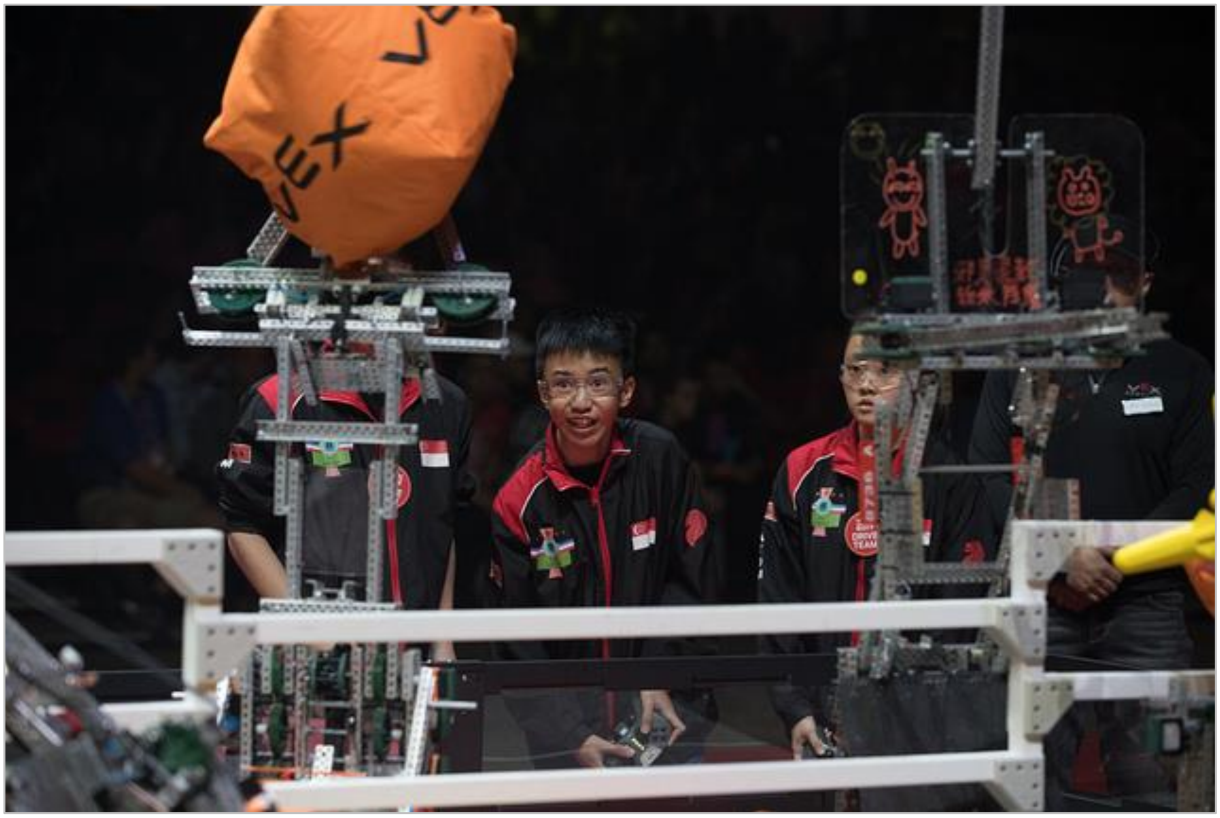
When a vehicle is traveling at any speed, momentum is at work. When that vehicle has a collision or brakes quickly, the sudden change in momentum of the car, its passengers, and any cargo can cause injuries and damage because of the force. That is why car manufacturers have developed several safety features, such as seat belts, airbags, padded dashboards, and crumple zones to protect any riders who may be involved in an accident.

Seat belts and airbags are important devices as they are designed to slow the body down more gradually. Slowing down more gradually reduces the forces on the body during a collision. Seat belts and airbags are legally required to be installed in vehicles and there are many states that enforce seat belt laws to ensure riders are protected.

Padded dashboards are safety features that give riders a way to protect themselves in case an airbag does not deploy. Hitting the padding instead of the dashboard reduces the forces acting on the body during impact. This feature has a huge effect on the severity of head injuries from car collisions.

Crumple Zones were first developed in 1952 by Béla Barényi, who worked for Daimler-Benz. He designed a car with designated areas that could collapse and absorb the kinetic energy released in an impact. These zones are still designed and utilized by auto engineers. Vehicles are designed to collapse in a controlled way during a collision, absorbing and redirecting the force of the impact. Crumple zones are usually located in the front and back of the vehicles.

Designing with Momentum in Mind



Optimizing forces is a challenge for competition robots

How Physics can Affect Competition Robots

When designing a robot for VEX Robotics Competitions, you must remember that any motor will be fighting against the robot's inertia whenever the robot is running. Inertia is an object's resistance to changes in its velocity. Inertia is increased as the object's mass and therefore its momentum are increased. This means if you add mass to your robot and make it heavier than it has to be, the motors will not be as effective in changing the robot's velocity! Therefore, you should try to use as light and as few materials as possible if you want to maximize the motors' efficiencies.

On the other hand, running a light robot very quickly can also lead to difficulties. If you are trying to make precise and accurate movements in the course of a competition, you may need to ease off of the power by reducing the velocity during your movements.

Let's explore the idea that the momentum of two colliding objects predicts what will happen after they collide. This is an important factor to consider when developing competition projects because you want your robot to move as fast as possible. You also want to have as many components as possible built into the robot that will give it an advantage for manipulating and collecting during the game.

Momentum is the amount of motion that an object has and is determined by the moving object's mass and velocity. So, a competition robot with all of its components can be heavy and moving as fast as possible. Therefore, its momentum is very high. This is when you need to consider what happens when it comes into contact with parts of the field or other robots.

Look back at your table from the Exploring Velocity activity. You tested the transfer of energy during collisions by setting different velocities for the robot and driving it forward until it hit the ball. You should have noticed that higher velocities set for the robot pushed the ball farther after a collision than lower velocities did. This is an obvious effect of the robot's momentum because the mass of the robot remained the same but the velocity increased and therefore, its momentum increased.

Something important to consider about that test is that the ball was not moving. It had a velocity, a momentum, and an acceleration of all zeros before the robot collided with it. Importantly, its mass was likely far less than the robot's mass. After the collision, its acceleration and therefore its velocity and its momentum all increased. How fast the ball's velocity was after the collision depended in part on the mass of the ball. Lighter balls accelerate and move faster. If your class used a ball with more mass, imagine a bowling ball, the ball might have moved slowly and not very far after the collision.

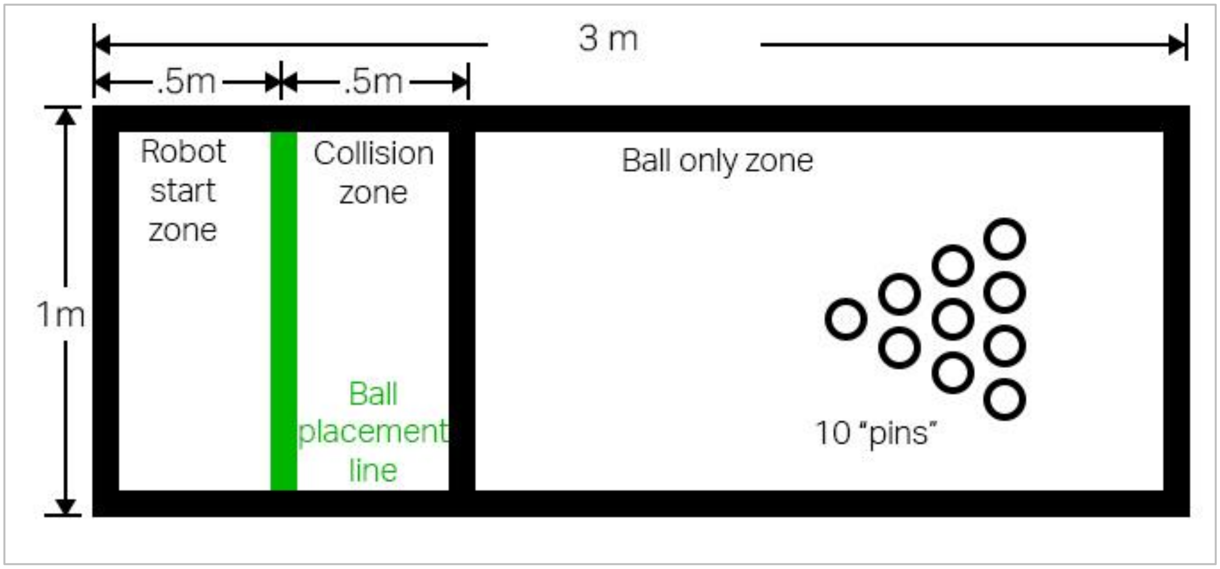
Again, this is important to consider when planning for a competition because you can break parts of the field, parts of your robot, or parts of other robots if the robot's momentum is too high. Imagine if your robot had a high velocity and crashed into an object that couldn't roll away like the ball in the previous activity. That object could have been broken by the impact forces (energy) of the collision.

SPARK

RETHINK

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

Prepare for the Strike Challenge



Layout of the Strike Challenge field

Prepare for the Strike Challenge

In this challenge, you will program your robot to collide with a ball to compete in bowling! To successfully complete this challenge, you need to create a project that transfers energy from your moving robot to a ball in order to knock down multiple pins. The project is only allowed to drive the robot forward using one instruction for setting the drive velocity and one instruction for driving for a distance. The project should stop the robot just after it comes in contact with the ball.

You can create a new project or use the Momentum project you created in the Exploring Velocity activity. While testing different velocities, record your data in the new table on the next page or in the table from the Exploring Velocity activity.

Consider what you've learned about the momentum of objects and the transfer of energy during collisions. Plan not only for the collision of the robot with the ball but also with the collision of the ball with the pins.

To complete the challenge you will need:

- 3m x 1m open area
- Tape (to create your bowling lane)
- Ball (the size and shape of a soccer ball)

- Paper (to roll up for creating bowling pins)

Design, Develop, and Iterate on your Project - VEXcode V5 Blocks

Answer the following questions in your engineering notebook as you design your project.

- What do you want the project have the robot do? Explain with details.
- What steps will you follow to test the project? Explain with details.
- How can your robot be programmed to complete the task with the most accuracy? Explain with details.

Follow the steps below as you create your project:

- Plan out the path you want to program your robot to follow using drawings and pseudocode when working with Blocks.
- Use the pseudocode you created to develop your project.
- Test your project often and iterate on it using what you learned from your testing. After each trial, record how far the robot drove, what the velocity was set to, about how far the ball traveled, and how many pins were moved. Here is an updated table for data collection and analysis.

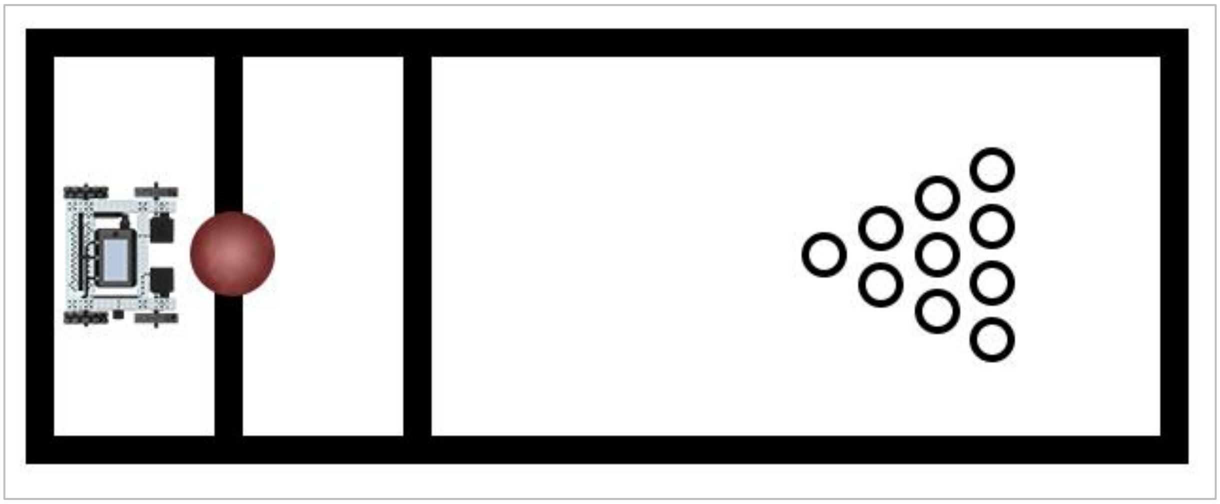
- Use the pseudocode you created to develop your project in VEXcode V5 Blocks.
- Test your project often and iterate on it using what you learned from your testing. After each trial, record how far the robot drove, what the velocity was set to, about how far the ball traveled, and how many pins were moved. Here is an updated table for data collection and analysis.

Distance Driven by Speedbot	Set Velocity of Speedbot	Distance Traveled by the Ball	Number of Pins Moved/Knocked Over
500mm	50%		
500mm			
500mm			

If you're having trouble getting started and you're working with Blocks, review one of the following Tutorials in VEXcode V5 Blocks:

- Drivetrain Movements
- Getting Started
- Download and Run a Project
- Moving and Removing Blocks
- Naming and Saving Projects
- Using Examples and Templates

Strike Challenge - VEXcode V5 Blocks



Strike Challenge setup

Strike Challenge

In this challenge, you will program your robot to transfer energy to a ball as you compete in bowling!

Challenge rules:

- The robot must begin in the Robot Start Zone.
- The ball must begin anywhere on the Ball Placement Line.
- The robot can only be touching the ball when the ball is:
  - On the ball placement line
  - In the collision zone
- Each game consists of 10 frames and follows standard bowling rules.
- The person or team with the most points at the end of the 10 frames wins!
- Have fun!

Design, Develop, and Iterate on your Project - VEXcode V5 Text

Answer the following questions in your engineering notebook as you design your project.

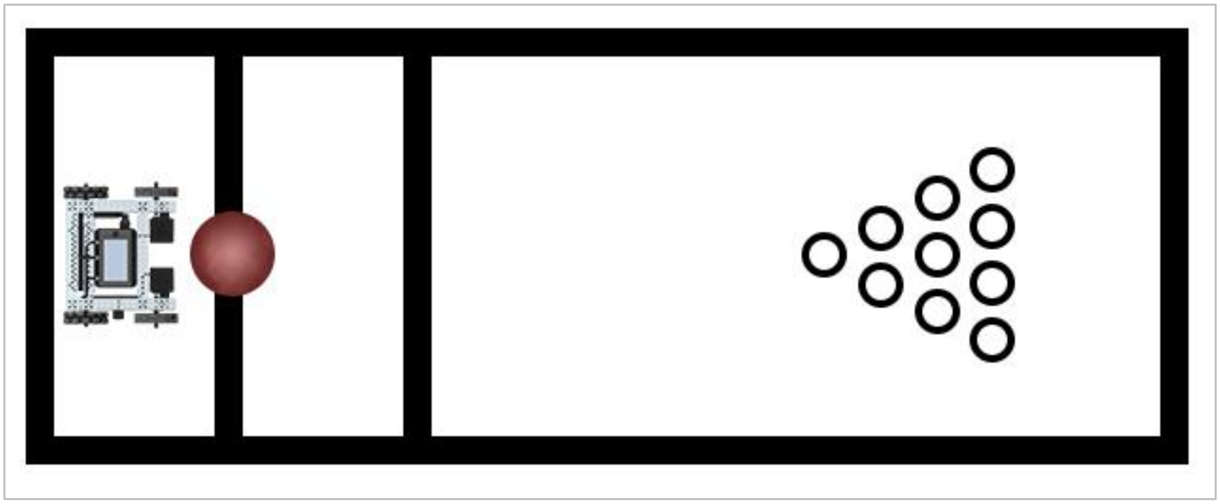
- What do you want the project have the robot do? Explain with details.
- What steps will you follow to test the project? Explain with details.
- How can your robot be programmed to complete the task with the most accuracy? Explain with details.

Follow the steps below as you create your project:

- Plan out the path you want to program your robot to follow using drawings and pseudocode when working with Text.
- Use the pseudocode you created to develop your project.
- Test your project often and iterate on it using what you learned from your testing. After each trial, record how far the robot drove, what the velocity was set to, about how far the ball traveled, and how many pins were moved. Here is an updated table for data collection and analysis.

Distance Driven by Speedbot	Set Velocity of Speedbot	Distance Traveled by the Ball	Number of Pins Moved/Knocked Over
500mm	50%		
500mm			
500mm			

Strike Challenge - VEXcode V5 Text



Strike Challenge setup

Strike Challenge

In this challenge, you will program your robot to transfer energy to a ball as you compete in bowling!

Challenge rules:

- The robot must begin in the Robot Start Zone.
- The ball must begin anywhere on the Ball Placement Line.
- The robot can only be touching the ball when the ball is:
  - On the ball placement line
  - In the collision zone
- Each game consists of 10 frames and follows standard bowling rules.
- The person or team with the most points at the end of the 10 frames wins!
- Have fun!

SPARK

KNOW

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

Review

1. **True or False: Before downloading a project onto the Speedbot, it is important to make sure the Brain is connected to the computer. If there is a successful connection, the Brain will be white.**
    - True
    - False

2. **True or False: Momentum is determined by the object's mass (how much is moving) and its velocity (how fast it is moving).**
    - True
    - False

3. **True or False: The momentum of the robot influences the acceleration and velocity of the ball after the collision. The mass of the ball did <u>not</u> make a difference.**
    - True
    - False

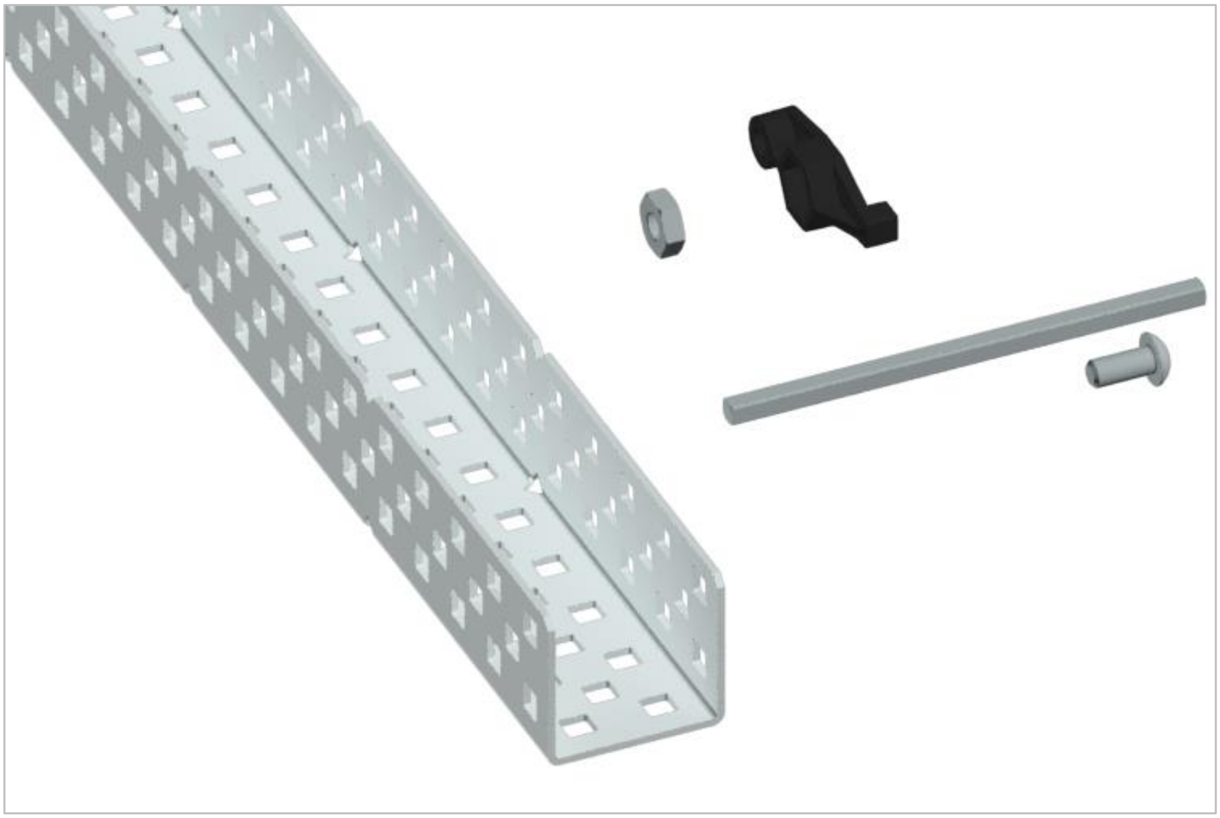
4. **Which of the following would have the greatest amount of momentum?**
    - A heavy or large object traveling at a slow speed.
    - A light or small object traveling at a high speed.
    - A heavy or large object traveling at a high speed.

5. **True or False: Acceleration is the rate at which an object's velocity is increasing. If an object is accelerating, its momentum is also increasing.**
    - True
    - False

APPENDIX

Additional information, resources, and materials.

Using the 1 Post Hex Nut Retainer w/ Bearing Flat



1 Post Hex Nut Retainer w/ Bearing Flat

Using the 1 Post Hex Nut Retainer w/ Bearing Flat

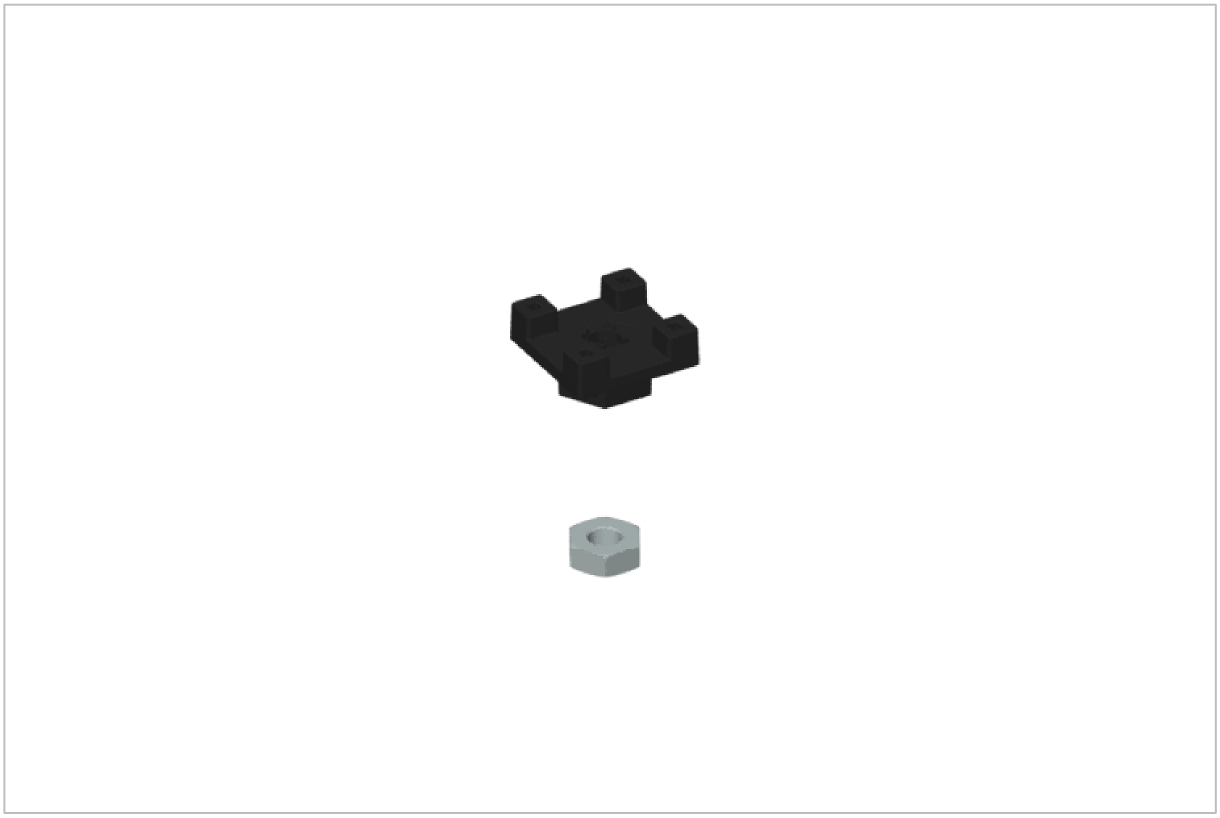
The 1 Post Hex Nut Retainer w/ Bearing Flat allows shafts to spin smoothly through holes in structural components. When mounted, it provides two points of contact on structural components for stability. One end of the retainer contains a post sized to securely fit in the square hole of a structural component.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

To make use of the retainer:

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

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

Using the 4 Post Hex Nut Retainer



4 Post Hex Nut Retainer

Using the 4 Post Hex Nut Retainer

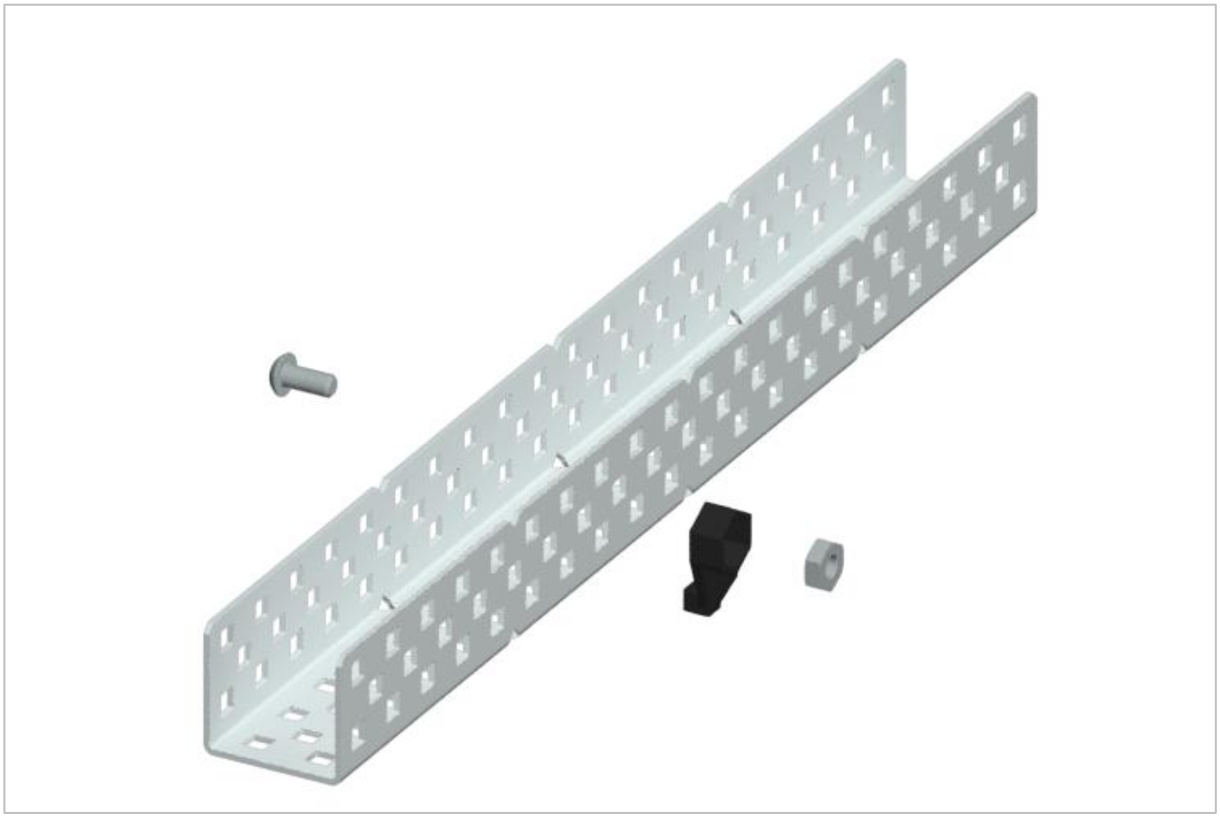
The 4 Post Hex Nut Retainer provides five points of contact for creating a strong connection between two structural components using one screw and nut. Each corner of the retainer contains a post sized to securely fit in a square hole within a structural component. The center of the retainer is sized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

To make use of the retainer:

- Align it on a VEX structural component such that the center hole is in the desired location, and each corner is also backed by the structural component.
- Insert the square posts extruding from the retainer into the structural component to help keep it in place.
- Insert a hex nut into the center section of the retainer so that it is flush with the rest of the component.

- Align any additional structural components to the back of the main structural component, if applicable.
- Use an 8-32 screw of appropriate length to secure the structural component(s) to the retainer through the center hole and hex nut.

Using the 1 Post Hex Nut Retainer



1 Post Hex Nut Retainer

Using the 1 Post Hex Nut Retainer

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

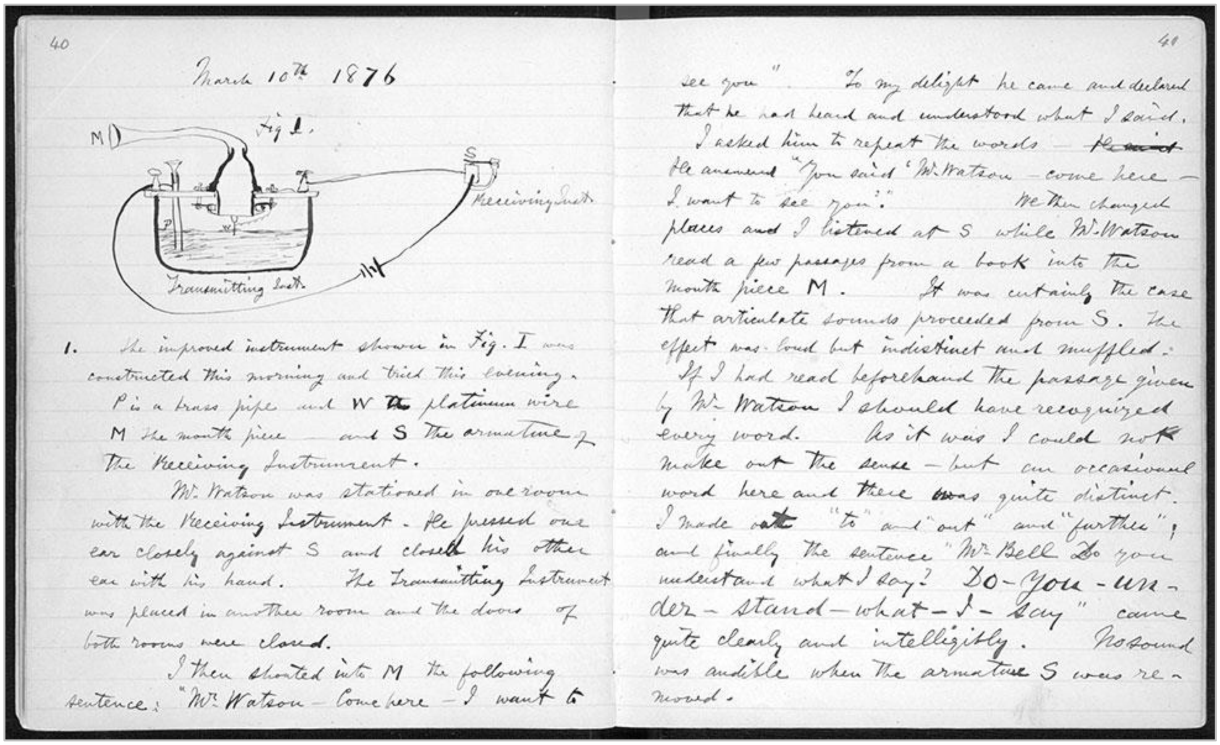
To make use of the retainer:

- Align it on a VEX structural component such that both ends are backed by the structural component and positioned to secure the second piece.
- Insert the square post extruding from the retainer into the structural component to help keep it in place.
- If the retainer is being used to secure two structural components, insert a hex nut into the other end of the retainer so that it is flush with the rest of the component. If used to secure

a different type of component, such as a standoff, it may be appropriate to insert the screw through this side.

- Align any additional components to the back of the main structural component, if applicable.
- If the retainer is being used to connect two structural components, use an 8-32 screw of appropriate length to secure the structural components through the hole and hex nut. If used to connect a different type of component, such as a standoff, secure it directly or with a hex nut.

Engineering Notebooks


40

March 10th 1876

Fig. 1.

M, S, Receiving Inst., Transmitting Inst.

1. The improved instrument shown in Fig. I was constructed this morning and tried this evening.
P is a brass pipe and W the platinum wire M the mouth piece and S the armature of the Receiving Instrument.

Mr. Watson was stationed in one room with the Receiving Instrument. He pressed one ear closely against S and closed his other ear with his hand. The Transmitting Instrument was placed in another room and the doors of both rooms were closed.

I then shouted into M the following sentence: "Mr. Watson — Come here — I want to

41

see you." To my delight he came and declared that he had heard and understood what I said.

I asked him to repeat the words — ~~He said~~ He answered "You said 'Mr. Watson — come here — I want to see you.'" We then changed places and I listened at S while Mr. Watson read a few passages from a book into the mouth piece M. It was certainly the case that articulate sounds proceeded from S. The effect was loud but indistinct and muffled.

If I had read beforehand the passage given by Mr. Watson I should have recognized every word. As it was I could not make out the sense — but an occasional word here and there was quite distinct. I made out "to" and "out" and "further"; and finally the sentence "Mr. Bell Do you understand what I say? Do - you - un-der - stand - what - I - say" came quite clearly and intelligibly. No sound was audible when the armature S was removed.

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

An Engineering Notebook Documents your Work

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

Your engineering notebook should have the following:

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

An entry might include:

- Brainstorming ideas
- Sketches or pictures of prototypes

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

Popups within this STEM Lab

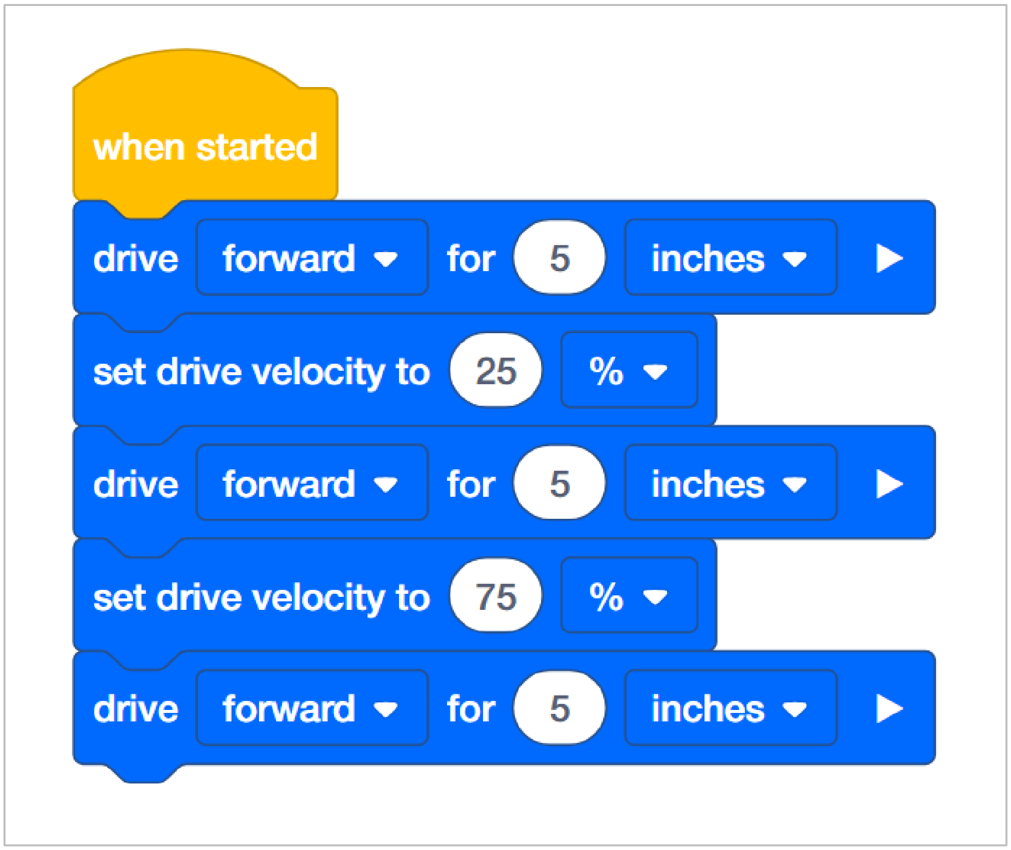
Exploring Velocity Outline

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 V5 Blocks.
- Check that the VEX V5 Speedbot is ready for the exploration.
- Start a new project in VEXcode V5 Blocks.
- Rename and save the project (Windows, MacOS, Chromebook).
- Create the Drive Velocity project that moves the Speedbot at different velocities.
- Download and run the project.
- Change the Drive Velocity project to move the Speedbot in reverse.
- Download and run the project.
- Test how different velocity settings affect the robot's momentum and transfer of energy to a ball by collecting data in a table and recognizing patterns.

Changing the Speed Build's Velocity

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



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 this or other blocks used in this project.

Students might ask about the arrow at the end of the drive 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.

Demonstrating how the Drivetrain to Drive Blocks are Related

Have a Speedbot to demonstrate for the students. Introduce to students the drive block. Read the description of the drive block in the Help. Either have VEXcode V5 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 the Speedbot 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.

Whenever starting a new project...

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.”

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