Self-Driving Code Base Activity Series- Teacher Notes

Setting the Stage:
This Activity Series is designed to introduce students to coding an Eye Sensor in an interactive and engaging way.

The Activity content is based on the following context setting. Students can gain familiarity with coding the Code Base 2.0 - Eye Down build because each Activity builds on the previous. They can be used individually, or as a sequence, to enable students to build and explore the Code Base and VEX GO Eye Sensor and how they work. The context provided here is an example setting, and could be adjusted or changed to best meet the needs and interests of your classroom and topics of study.

Before your students begin working on the Activities, share this context with them.

Self-driving cars are becoming more and more common. You may have seen some of the different prototypes in pictures, or even driving around a city while people test the cars. There are many things that go into making a self-driving car able to navigate safely. One of the key pieces are sensors – different sensors collect data, and that data is used in the car's code so that it can make decisions, like stopping at a stop sign.

We are going to explore this idea of sensors, and coding sensors for self-driving cars, by building and coding the Code Base 2.0 - Eye Down! The Code Base 2.0 - Eye Down is a VEX GO robot that has an Eye Sensor pointed down, facing the ground. It uses the Eye Sensor to report data back to the robot to help the Code Base interact with its environment. Like self-driving cars in a city, our Code Base will need to drive, stop, and slow down, based on the data the Eye Sensor reports.

First, we are going to see how we can use the Eye Sensor data to make our Code Base stop when it sees a line on the road. Then, we will add colors to our code, so that our Code Base only stops when the Eye Sensor sees a red line. Finally, we'll build on our coding practice and use it to create a project where the Code Base has to complete different actions depending on what color the Eye Sensor detects.

Setting up the Activities:

Each Activity fits into a portion of this context setting. **Move Until Line** uses the Code Base 2.0 - Eye Down build and asks students to test and iterate on an example project in VEXcode GO to become familiar with coding the Eye Sensor. **Stop Sign** has students create their own VEXcode GO project to stop the robot when the color red is detected. **Construction Zone** combines the first two activities to have students code their robot to drive through a construction zone with different colors indicating what the Code Base should do. All Activities can use the same materials, which are listed below.

Making the Activities Your Own:

These Activities are editable Google docs, so you can easily change them to fit the needs of your classroom. Add in the use of **If then else** or **If then** blocks, or remove other elements, for differentiation purposes, or to best match the coding concepts your students are learning in the classroom. See the following articles for more information about editing Google docs for your own classroom use with Google drive or with Microsoft Office.

Setting up the “Self-Driving Scenario”:

Prior to beginning the Activities, make sure students have the following materials ready:

Materials Needed:

- VEX GO Kit
- VEX GO Field
- Code Base 2.0 - Eye Down Build Instructions (PDF)
- Code Base 2.0 -Eye Down Build Instructions (3D)
    - To build the Code Base 2.0 - Eye Down, students will need a completed Code Base 2.0.
        - Code Base 2.0 Build Instructions (PDF)
        - Code Base 2.0 Build Instructions (3D)
- Tablet or Computer
- VEXcode GO
- Red, green, and blue paper OR whiteboard markers of the same colors
- Tape (to attach colored paper to the Field)

Setting up VEXcode GO:

- Have VEXcode GO downloaded on all tablets or computers students will be using before starting the Activities. For information about setting up VEXcode GO, see this article.
- During the first Activity, show students how to connect the Brain on their Code Base to their device in VEXcode GO. Because connection steps vary between devices, see the Connecting articles of the VEX Library for specific steps to connect the VEX GO Brain to your computer or tablet.

Teacher Tips:

- **Make sure students are aware of the concept of a self-driving car –** If your students are unfamiliar with the idea of a self-driving car, give them that prior knowledge to make sure that everyone is on the same page with the context of the Activities. You can share images, or use videos, to help illustrate what a self-driving car is, and how it works.

- **If students get done building at different times,** there are a number of meaningful learning activities early finishers can participate in as the rest of the group finishes building. View this article for several suggestions about how to plan for engaging students who finish building earlier than others.

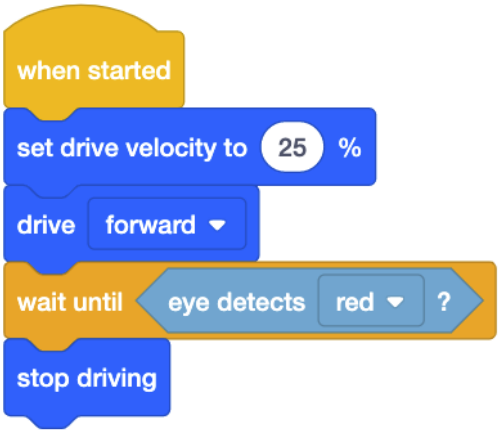
- **For a simpler setup** -
    - Have one central testing station with individual Tiles for the **Move Until Line** and **Stop Sign** Activities. To create a simpler setup for the **Construction Zone** Activity, set up the Fields before class begins.
    - Use whiteboard markers rather than paper and tape to create the colored lines on the Field. If you are unsure if the markers you have will work, test with an Eye Sensor and

the Monitor Console in VEXcode GO before beginning. See this article for information about using the Monitor Console in VEXcode GO.

- **For extra engagement -** Add an additional coding challenge! The second and third Activities ask students to create their projects using **Wait until** blocks. Introduce students to **If then** blocks and **Forever** loops to create a code that will continuously repeat.

Sample Solutions:
While there are many ways to solve the coding challenges in **Stop Sign** and **Construction Zone**, here are two possible solutions.

- **Stop Sign:**



- **Construction Zone**

Additional Standards That Can Be Reached with this Activity Series:

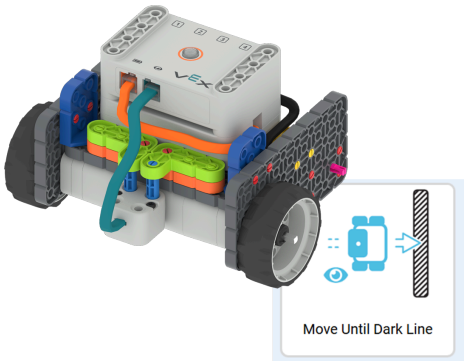
International Society for Technology in Education (ISTE)

- ISTE (5) Computational Thinker - 5c: Students break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.
- ISTE (6) Creative Communicator - 6b: Students create original works or responsibly repurpose or remix digital resources into new creations.

Computer Science Teachers Association (CSTA)

- CSTA 1A-AP-10: Develop programs with sequences and simple loops, to express ideas or address a problem.

VEX GO Activity



Move Until Line

Detect brightness with the Eye Sensor

Use the “Move Until Dark Line” example project to see how the Eye Sensor can detect lines to keep your car in its lane!

Step by Step

1. Build the Code Base 2.0 (PDF Build Instructions or 3D Build Instructions (3D)) and add on the Eye Sensor with the Code Base 2.0 - Eye Down build instructions (PDF or 3D).
2. Connect your Code Base - Eye Down to your device in VEXcode GO.
3. Open the “Move Until Dark Line” example project in VEXcode GO.
4. Place your Code Base - Eye Down on the GO Field and start the project!
5. Change the parameters of the two blocks marked in the image to the right: the **Set eye light power** and the **Less than** block. Start the project again. How do these changes affect the movement of the Code Base?

‘LEVEL UP’

- **Detour -** Code your Code Base to turn around 180 degrees when the dark line is detected and drive to another area.
- **Monitor Console -** Open the Monitor Console and check the box next to the **Eye brightness** block in the Toolbox. Start the project again. What are the highest and lowest values you see? Where was the Code Base on the Field when those numbers appeared on the console?

Pro Tips

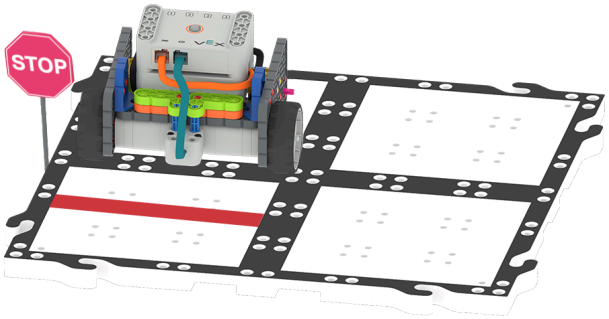
Set the Light

The light on the Eye Sensor is used to brighten the area being observed by the Eye Sensor.

If you are in a dark classroom, set the light brighter. If you are in a bright classroom, set the light power lower.

Standard: ISTE (1) Empowered Learner - 1c: Students use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways.

VEX GO Activity



Stop Sign

Detect color with the Eye Sensor

Create a VEXcode GO project to make your car stop at a stop sign!

Step by Step

1. Use your Code Base - Eye Down, and connect your Code Base - Eye Down to your device in VEXcode GO.
2. Set up your Field with a red line as shown in the image above, and configure your Code Base in VEXcode GO.
3. Create a project for the Code Base - Eye Down to stop when it sees the red line. You can use Drivetrain blocks and the **Wait until** block with the **Detects color** block to do this.

4. Set the **Detects color** block to ‘red,’ then test and iterate on your project until it works as intended!

‘LEVEL UP’

- **Stop Signs All Around -** Add to your code to have your car continue to drive forward to another stop sign after stopping at the first!
- **Build a Stop Sign -** Design and build a traditional stop sign using VEX GO pieces and art supplies. Place it on the Field next to your red line!

Pro Tips

Set drive velocity

Adjust the drive velocity of the Code Base to see the behaviors more clearly. 50% is the default speed. Try 20 or 25% and see how the Code Base moves differently.

Standard: ISTE (1) Empowered Learner - 1c: Students use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways.

VEX GO Activity

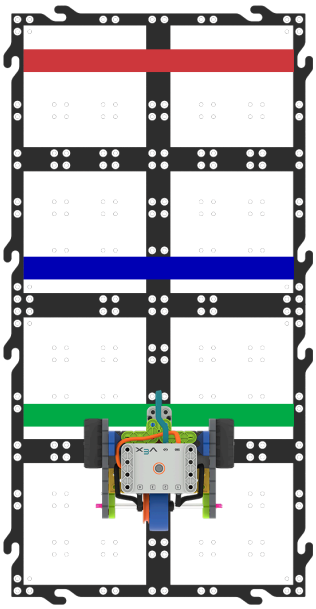
Construction Zone

Slow down in the construction zone

Code the Code Base to drive, slow down, and stop based on the colors detected by the Eye Sensor!

Step by Step

1. Use your Code Base - Eye Down and connect it to your device in VEXcode GO.
2. Set up your Field with three colored lines (green, red, and blue), like the example to the right.
3. Create a project where the Code Base - Eye Down will drive forward when it detects green, slow down when it detects blue, and stop driving when it detects red.
4. Test your project! Keep iterating on your project until you have successfully coded all three colors.



‘LEVEL UP’

- **Do it Again -** How can you edit your code so that the Code Base would know what to do if two blue lines were on the Field? Create and test your project!
- **Car Design -** Sketch your own self-driving car design! What does your car need on it to be safe? Label all the different sensors and other unique features.

Pro Tips

Start on Green!

Make sure the Eye Sensor is above the green line before you start your project.

If not, the Code Base may not start driving forward as you intended.

Standard: ISTE (1) Empowered Learner - 1c: Students use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways.