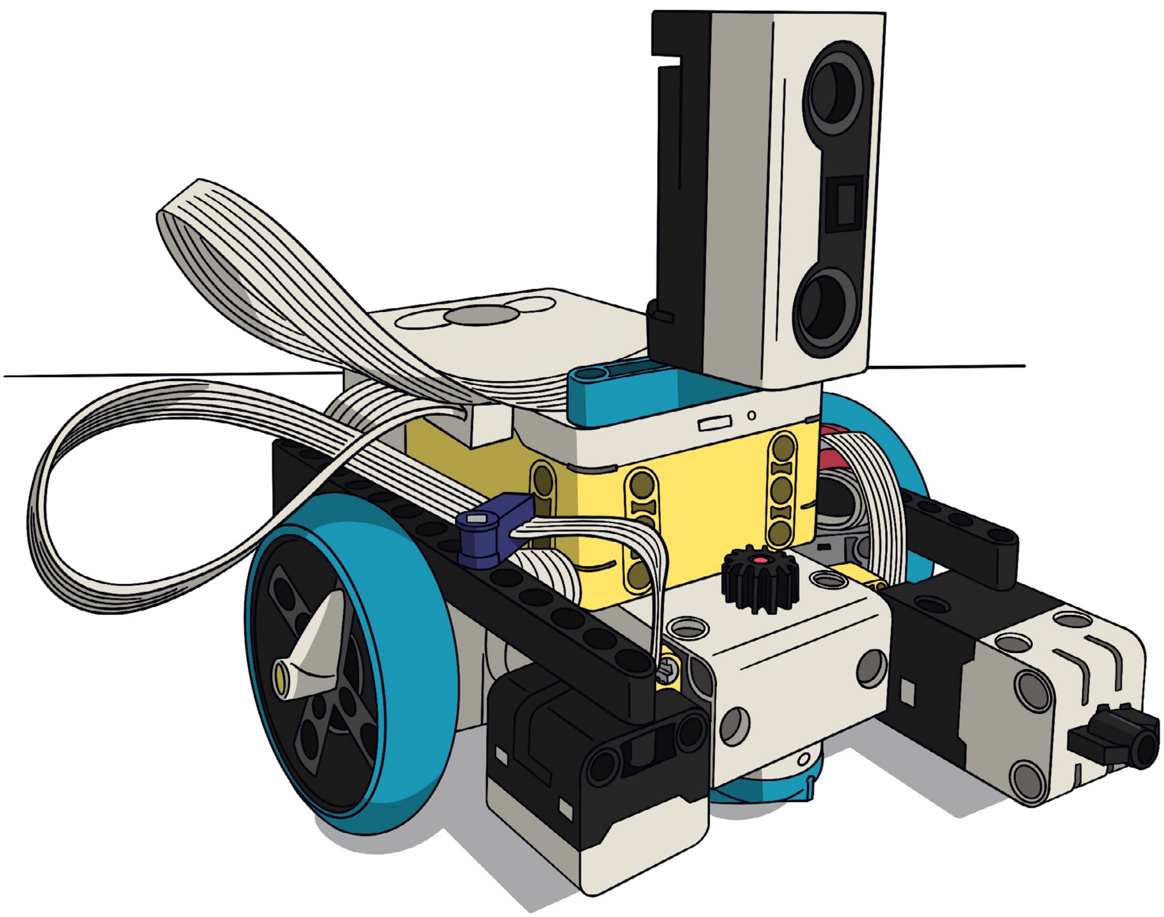
Bio-Coding Club Curriculum

MICROBIOME EXTENSION



BROAD INSTITUTE
STEM ENGAGEMENT AND INCLUSION

Acknowledgement

The Microbiome Extension Unit was made possible through a generous sponsorship from Intel.

BROAD INSTITUTE | STEM ENGAGEMENT AND INCLUSION

Table of Contents

WELCOME & PURPOSE

Welcome to the Bio-Coding Club curriculum. The goal of this curriculum is to give educators all the tools needed to run successful student-driven coding projects related to biology – and to boost student interest in STEM by making science fun. The Broad Institute created the Bio-Coding Club, in collaboration with Putnam Avenue Upper School in Cambridge, Massachusetts, as an after-school program to teach students in grades 6-8 about both biology and coding. The club met weekly for 75 minutes. A team of professional scientists, all volunteers from the Broad Institute, served as mentors. Club meetings generally started with a snack and a discussion of a biology topic. Students then took part in a hands-on activity before moving on to practice coding with the coding language Scratch (scratch.mit.edu).

This module is an extension of the Microbiome Unit curriculum found at broad.io/biocodingcurriculum. It utilizes the SPIKE Prime robot kit by LEGO to add additional elements of robotics and coding in Python. It consists of five more advanced computing lessons that can be completed in place of the beginner-level Scratch lessons outlined in the original curriculum. It may also be used as an addendum to the Microbiome Unit, after students have gained familiarity with using coding in Scratch in the original curriculum. Here, we move the lesson outside the computer. Instead of moving a digital bacterium through a maze, the students program a robot to do so. The student can then move beyond block-based coding by using the SPIKE Prime's Python coding interface. The lessons you'll find here are suggestions; they should be modified to fit your program's resources and students.

MEET THE MICROBIOME

All the microorganisms (bacteria, viruses, fungi, and protozoa) that live in and on a person are known as their "microbiome." Interest in the human microbiome has exploded in recent years as advances in genetic sequencing have allowed scientists to begin to catalogue and analyze all the microbes people carry. It has become clear that a person's microbiome can have profound effects on their health.

The microbiome is a nice vehicle for teaching biology because it pairs well with discussion of microorganisms, anatomy, human health, genetics and DNA sequencing, and even ecology. While the science is complex, our work with middle school students through the Bio-Coding Club shows that they enjoy learning about this trendy topic and have no trouble understanding the fundamentals.

The Microbiome Unit introduces students to genetics and the microbiome through at least six meetings' worth of hands-on activities. Each is paired with a Scratch project that builds upon itself over the weeks: students create a bacterium and then code it to move through a maze to get to a food source. Students can add their personal touches to the project as they go along, from drawing their own bacterium to designing their own maze obstacles. Each new component to their project comes with a Scratch card, inspired by the ones developed by the MIT Media Lab. Scratch cards provide the students with sample code so they can work independently from the teacher and volunteers. By the end of the original Microbiome Unit, students will have learned about microbiomes and created their own Scratch maze project, giving them the foundational coding skills needed to create more Scratch projects in the future.

Some middle school students already have experience with Scratch from summer camps, after-school programs, or from following their own interests. We developed this extension for those students who are ready for the next step of coding. Students can practice using LEGO's Scratch-based block coding to program a robot. They can literally see their code come to life with the SPIKE Prime robots. They can also practice coding their robots in Python. Python is used in diverse fields, from AI to video game development. Notably, Python is one of the coding languages used most frequently by scientists.[1] Python is easy to learn, making it a natural second programming language after learning a block-based code like Scratch.

[1] TIOBE. (2023, October). TIOBE Index. www.tiobe.com/tiobe-index/

MEET THE MICROBIOME

Here is one sample schedule for completing this unit, but of course it can be modified to suit your needs:

Meeting 1 (optional)	**Introduction to the SPIKE Prime Robot:** Do the tutorial activities provided by LEGO
Meeting 2	**Build the bot:** Students build the robot they will be utilizing throughout the unit
Meeting 3	**Learning about the bot:** Students challenge the robot to make some basic moves
Meeting 4	**Using sensors:** Students explore the force, distance, and color sensors
Meeting 5	**Challenge:** Students code the robot to find its way through a maze
Meeting 6	**Python:** Students re-write their block code to use Python

NEXT GENERATION SCIENCE STANDARDS

- **MS-ETS1-1:** Define the criteria and constraints of a design problem with sufficient precision to ensure a successful solution, taking into account relevant scientific principles and potential impacts on people and the natural environment that may limit possible solutions.
- **MS-ETS1-2**: Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.
- **MS-ETS1-3:** Analyze data from tests to determine similarities and differences among several design solutions to identify the best characteristics of each that can be combined into a new solution to better meet the criteria for success.
- **MS-ETS1-4:** Develop a model to generate data for iterative testing and modification of a proposed object, tool, or process such that an optimal design can be achieved.

Setting Up LEGO SPIKE Prime

Using the LEGO SPIKE Prime requires significant time for initial setup. LEGO SPIKE Prime kits can be purchased at LEGO's website: https://education.lego.com. After buying, plan on spending at least 20 minutes per kit unboxing and updating the hub.

This is the setup procedure:

1. Add the LEGO SPIKE software to student computers: https://education.lego.com/en-us/downloads/spike-app/software. LEGO also offers a web-based platform if downloading software onto students' computers is difficult: https://spike.legoeducation.com.
2. Unbox the SPIKE Prime robot. The kit comes with sticker labels and trays to organize and store the various parts.
3. Connect the yellow SPIKE Prime hub to the computer and click on the SPIKE Prime App to update the hub. You will be prompted to name your hub. If you have more than one hub in a class/group, give each a unique name and use the included labeling stickers to write your chosen name on all the pieces.
4. After updating the hub, you can click on "New Project" to start a project. This will give you the option of choosing "WORD BLOCKS," which will start a project using a block-based code based on Scratch. Alternately, you can click on "PYTHON" to begin a project using Python.
5. Make sure your hub is connected to your computer via the included cord or by Bluetooth. You can run through the tutorials on the home page or get started on your own projects!

Hub Center Button Color Key:

Red: Program error

Orange: Low battery

Green: Lost connectivity − hold down the Center button for 3 seconds

Blue: Update hub with cable and SPIKE App

MICROBIOME EXTENSION 1:
Introduction to the SPIKE Prime Robot and Block-Based Coding (Optional)

1

Objective

Students will be able to (SWBAT) connect their computers to the SPIKE Prime hub, write and send simple block-based code to the hub, and become familiar with how the components of the SPIKE Prime kit (light matrix, motor, color sensor, distance sensor, force sensor, and gyro sensor) operate.

Vocabulary

- **Hub:** The brick-shaped device that receives code from the SPIKE Prime App. It features a customizable light matrix, speaker, Bluetooth connectivity, and six ports for connecting sensors and motors.

Framing Question

What is LEGO SPIKE Prime, and what can I do with it?

Materials

- Bluetooth-capable computers
- LEGO SPIKE Prime App installed on computers, or an internet connection to access the web app (https://spike.legoeducation.com)
- LEGO SPIKE Prime kit for each group of 2-4 students

Plan

1. Organize students into groups based on the number of robot kits and computers available.
2. Launch the SPIKE Prime App in each group. The students should select SPIKE Prime as their solution.
3. LEGO has created a very nice set of tutorial activities that introduce students to the SPIKE Prime robot and their Scratch-like block-based coding language. On the home page, students click "Get started with SPIKE prime" to access six tutorials that demonstrate how to connect the hub and show some of the capabilities of the robot components (light matrix, motor, color sensor, distance sensor, force sensor, and gyro sensor).
4. As students run through these tutorials, they'll learn about the different components of the robot they will be coding. Allow students to proceed at their own pace, or step through the tutorials as a class if you feel that would be more helpful.
5. Optional challenge: End the meeting by having the students write code to spell out a message using the hub's light matrix.

How will you determine if students met the objective?

Students should be able to demonstrate some of the SPIKE Prime robot's capabilities with code they have written.

Facilitator Tips

- Assign students to a single robot kit they will re-use each week.
- Remind the students that they will only be using the hub, motors, and sensors for this day's lesson. Therefore, all other materials should stay in their bins.
- When the Center button on the hub flashes orange, the battery is low. You can recharge the hub by connecting it via the provided USB cord to the computer while not directly in use. Charging a battery from empty will take about two hours, and the battery will last for several hours with typical use.

Hub Center Button Color Key:

Red: Program error

Orange: Low battery

Green: Lost connectivity − hold down the Center button for 3 seconds

Blue: Update hub with cable and SPIKE App

MICROBIOME EXTENSION 2: Building a Robot

2

Objective

SWBAT use the directions provided by LEGO to build a robot that incorporates the distance, force, and color sensors.

Vocabulary

- **Robot:** A machine, especially one programmable by a computer, that can sense its environment, carry out computations to make decisions, and perform actions in the real world

Framing Question

Can I follow a set of visual directions to build a robot with LEGO pieces?

Materials

- Bluetooth-capable computers
- LEGO SPIKE Prime App installed on computers, or an internet connection to access the web app (https://spike.legoeducation.com)
- LEGO SPIKE Prime kit for each group of 2-4 students

MICROBIOME EXTENSION 2: Building a Robot

2

Plan

1. Explain to students that in the next several meetings, they will be coding a robot to move through a maze. Before they can send code to their robot, they must build it! This unit uses a modified version of LEGO's "Driving Base 2" for the projects. This robot form incorporates motors, the color sensor, the distance sensor, and the force sensor, so code for all of those sensors can be sent to their robot.

2. Allow students time to build the robot:

   - On the home page of the app, click "Building Instructions."
   - Select "Driving Base 2."
   - Select "Driving Base" and complete the 34 steps.
   - Click on "Tools and Accessories." The grabbing arm will inhibit use of the distance sensor, so do not add the grabbing arm. Instead, start from step 18.
   - Finally, add the Color Sensor by returning to the "Building Instructions" page and selecting Driving Base 3.
   - Click on the Color Sensor Module for the steps to add that module.
   - The final build should look like this:

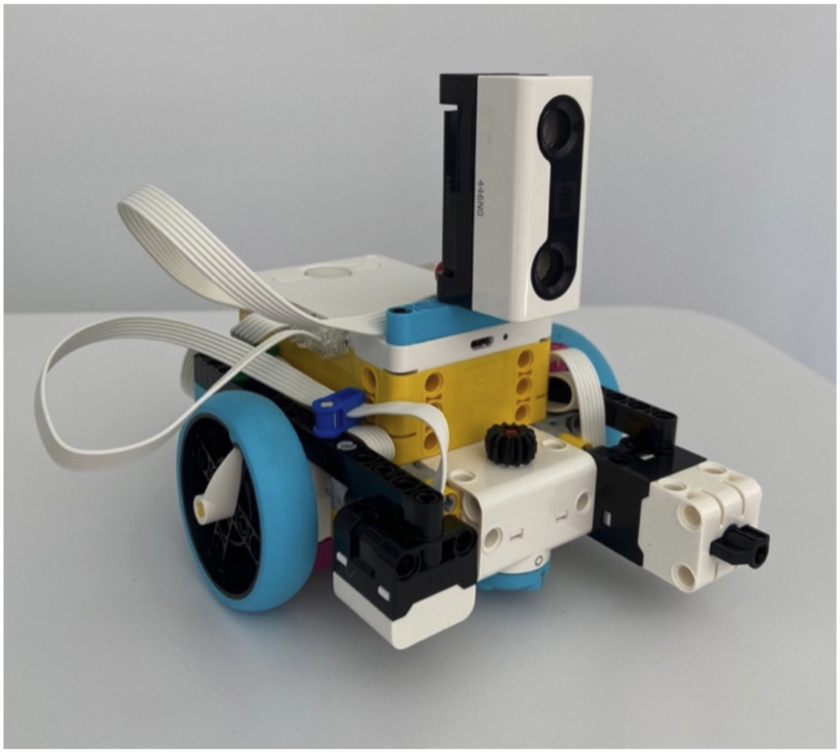


Figure 1. Completed build of Driving Base 2 with an added color sensor.

3. If students finish early, they can experiment with sending code to their newly-built robot.

How will you determine if students met the objective?

Do students have a completed robot with distance, force, and color sensors attached?

Facilitator Tips

- Building the robot will take at least 20 minutes for a speedy builder with experience. It may take much more time for those who are new to LEGO directions. Allot the amount of time you feel your students need.
- Keep in mind that a completed robot will not fit in the provided storage bin until you detach the distance sensor.
- A final sweep of the room after the students have left can reveal several stray LEGO bricks.

MICROBIOME EXTENSION 3: Learning to Program the SPIKE Prime Robot

3

Objective

SWBAT code their robot to move and turn.

Vocabulary

- **Degree:** In circular motion, “degrees” are a unit of measure for the angles in a circle. One complete circular rotation is 360 degrees. Halfway around the circle is 180 degrees.
- **Angular Motor:** A machine that transfers energy into motion. The LEGO SPIKE Prime motors use battery energy to turn rotors in a circular motion.
- **Yaw:** twisting left or right

Framing Question

What code do we need to get our robot to move and turn?

Materials

- Bluetooth-capable computers
- LEGO SPIKE Prime App installed on computers, or an internet connection to access the web app (https://spike.legoeducation.com)
- LEGO SPIKE prime kit for each group of 2-4 students
- Flat, unobstructed floor to run robots on

MICROBIOME EXTENSION 3: Learning to Program the SPIKE Prime Robot

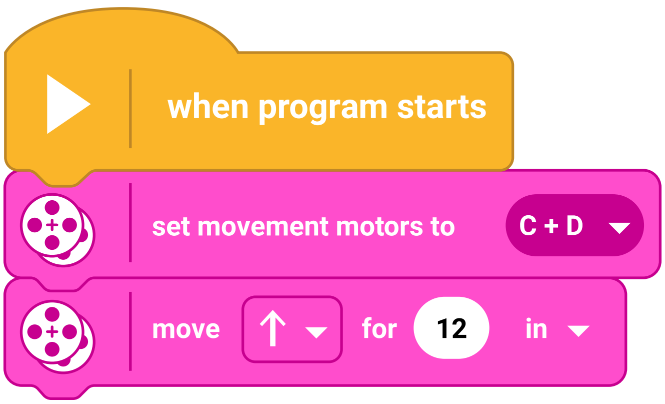
3

Plan

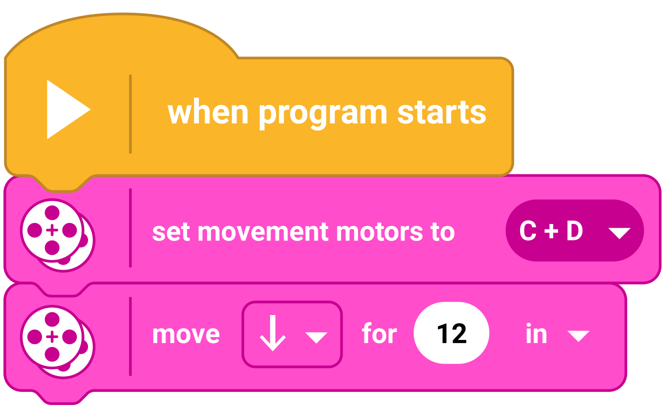
1. Explain to the students that today they will be figuring out how to code their robots to move. The block-based code of the SPIKE Prime robot looks very similar to Scratch. However, there are many new blocks that the students will have to become familiar with.

2. Start the lesson by showing the students the general format of the LEGO's block-based code. It is especially useful to point out how paired motors work together to keep the robot on course.

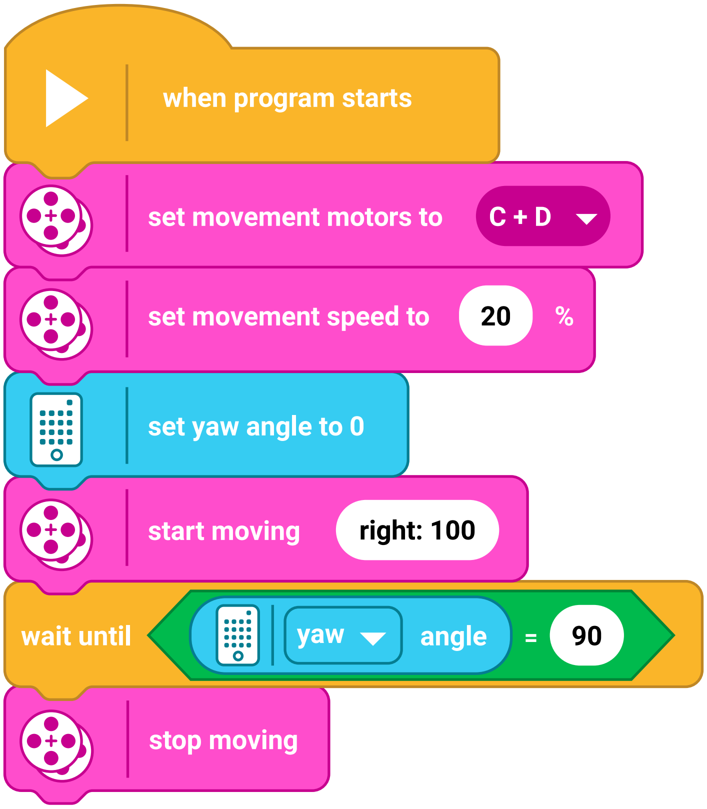
    - Give the students a series of challenges (utilizing Microbiome Extension 3 LEGO SPIKE Prime card)
        - Move the robot forward about a foot
            - Example code here:



        - Move the robot backward about a foot
            - Example code here:



- Turn the robot completely to the right.
    - Example code here:



MICROBIOME EXTENSION 3:
Learning to Program the SPIKE Prime Robot

3

- Make the robot move in an “S” shape.
  - Two different possible examples of code here:

```
when program starts
set movement motors to [C + D]
set movement speed to (20) %
move [↑] for (1) [rotations]
move (left: -100) for (180) [degrees]
move [↑] for (1) [rotations]
move (left: -100) for (180) [degrees]
move [↑] for (1) [rotations]
move (right: 100) for (180) [degrees]
move [↑] for (1) [rotations]
move (right: 100) for (180) [degrees]
move [↑] for (1) [rotations]
```

```
when program starts
set movement motors to [C + D]
set movement speed to (20) %
set yaw angle to 0
start moving (left: -30)
wait until <([yaw] angle) < (-175)>
set yaw angle to 0
start moving (right: 30)
wait until <([yaw] angle) > (175)>
stop moving
```

How will you determine if students met the objective?

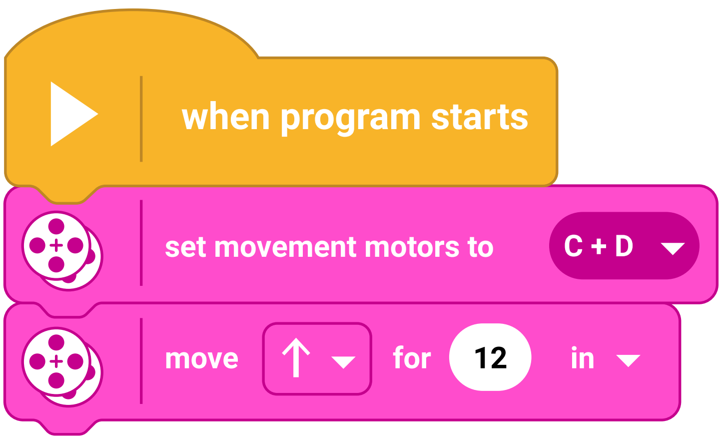
Students are able to rely on their own exploration and the hints provided on the extension card to move their robot as desired.

Facilitator Tips

- When a robot doesn't respond to a line of code, students may have trouble figuring out if their code has an error or if they didn't successfully send the code to the robot. To avoid this problem, students can start each program with a sound block (e.g., "play sound Cat Meow 1 until done") that will always play before the rest of their code is read, indicating that the code was successfully sent and initiated.
- Turning an exact degree amount is beyond the capabilities of the SPIKE Prime robots. Programming the robot to turn 90° will almost always result in a slight under- or over-rotation depending on the robot's traveling surface. To making an accurate turn, slow the robot's movement speed and use trial and error.

LEGO SPIKE Prime
Intro to movement

1 Code your robot to move forward

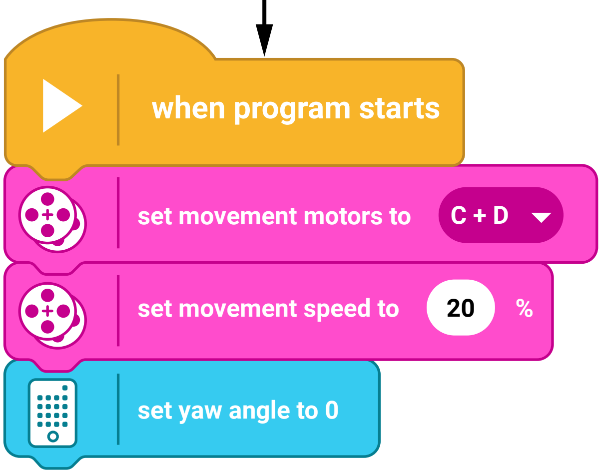


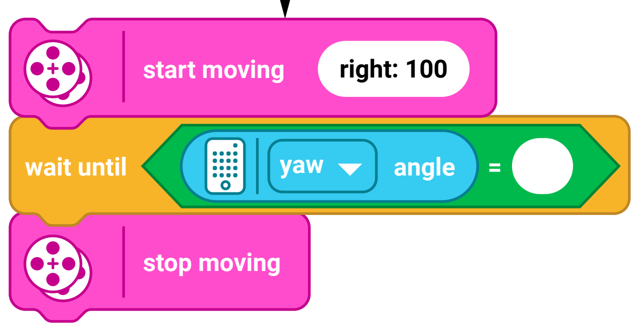
How would you code backward movement?

2 Code your robot to turn to the right

Set your starting conditions

Tell the robot how to turn





Challenge!!!
Make your robot move in an *S* shape

You can use these blocks

move ↑ for 1 rotations

move left: -100 for 180 rotations

move right: 100 for 180 rotations

Or you can use these blocks

set yaw angle to 0

wait until yaw angle < -175

start moving left: -30

wait until yaw angle > 175

start moving right: 30

MICROBIOME EXTENSION 4: Learning About the SPIKE Prime Robot Sensors

4

Objective

SWBAT code their robot to respond to information gathered from the force, distance, and color sensors.

Vocabulary

- **Robot sensor:** A device that is added to a robotic system in order to gather information from the robot's environment
- **LEGO Force sensor:** Detects and measures pressure or touch
- **LEGO Distance sensor:** Detects objects between 1-200 cm directly in front of it
- **LEGO Color sensor:** Detects colors, and measures light from darkness to bright sunlight

Framing Question

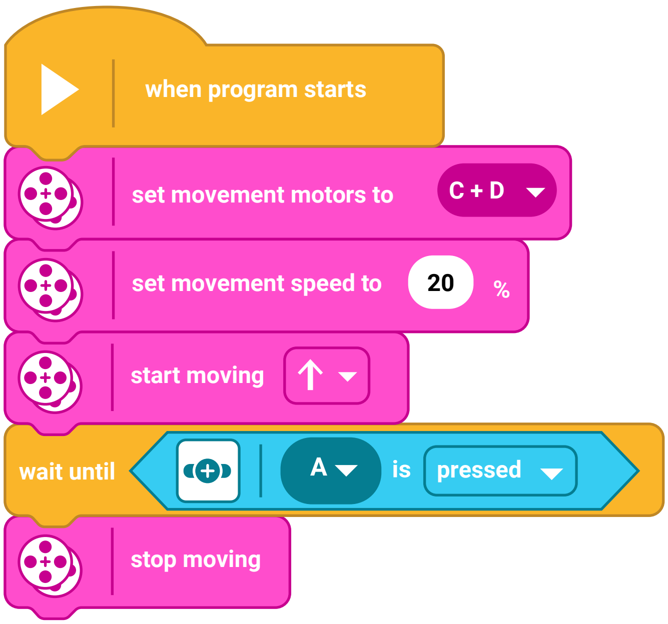
How can I incorporate the force, distance, and color sensors into my robot's code?

Materials

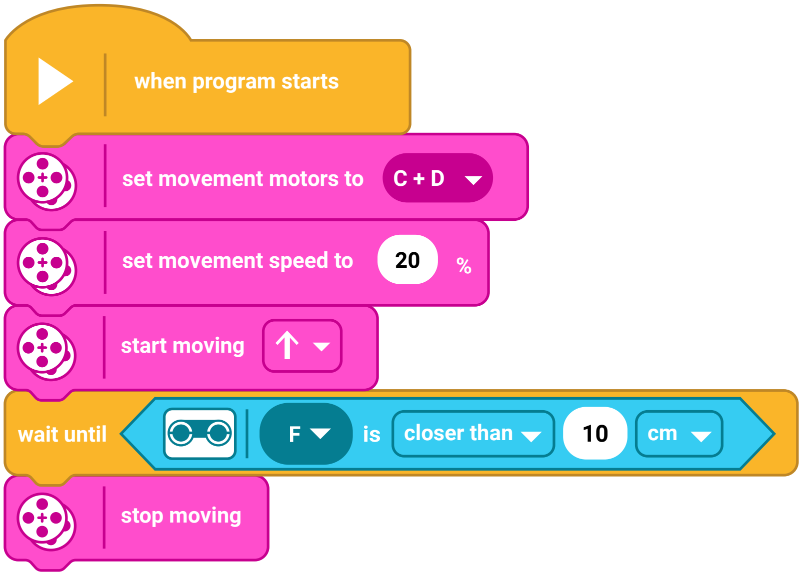
- Bluetooth-capable computers
- LEGO SPIKE Prime App installed on computers or an internet connection to access the web app (https://spike.legoeducation.com)
- LEGO SPIKE Prime kit for each group of 2-4 students
- Flat, unobstructed floor to run robots on
- Paper with a red line drawn on it. Line should be about 0.5+ cm thick.

Plan

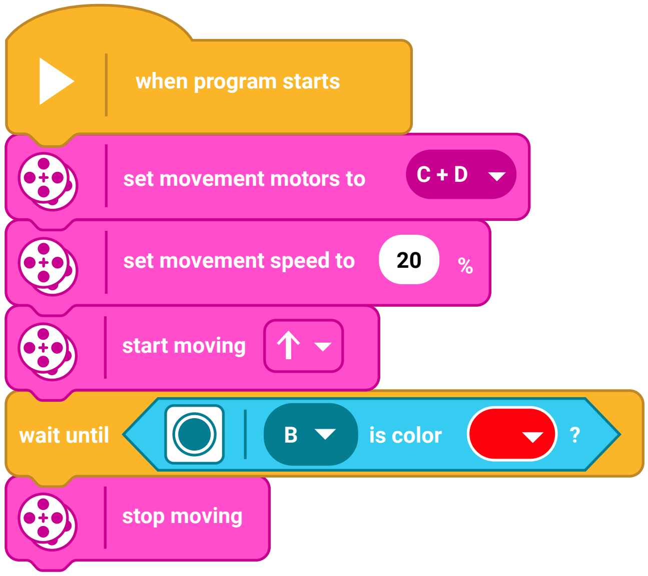
1. Explain to students that today they'll be challenged to become more familiar with the force, distance, and color sensors.
2. Demonstrate the following with one of the robots:
    - Moving forward until the force sensor is triggered and then stopping.
        - Example code:



    - Moving forward until the distance sensor is triggered and then stopping.
        - Example code:

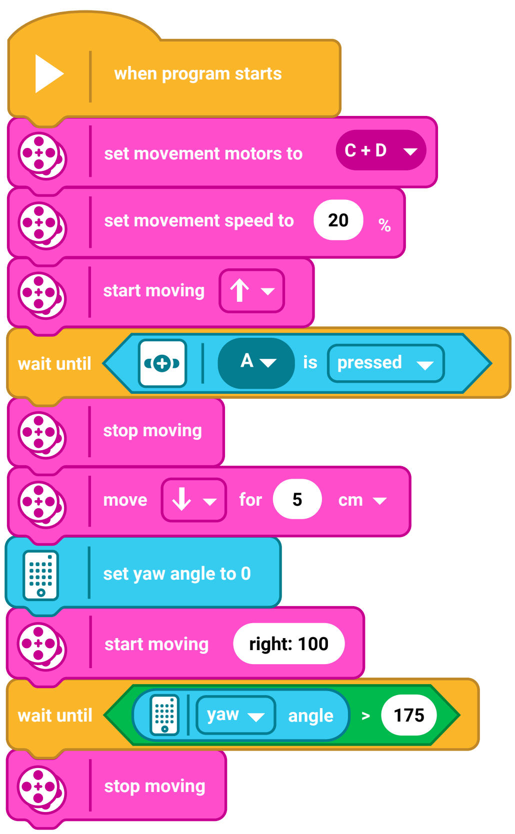


- Moving forward until the color sensor is triggered and then stopping.
  - Example code:



3. Challenge the students to do the same with their robot. They can use the hints on the Microbiome Extension 4 LEGO SPIKE Prime card.

4. If students complete the challenge, add on the task of turning their robot around after it comes to a stop. Here's some sample code that uses the force sensor.



MICROBIOME EXTENSION 4: Learning About the SPIKE Prime Robot Sensors

How will you determine if students met the objective?

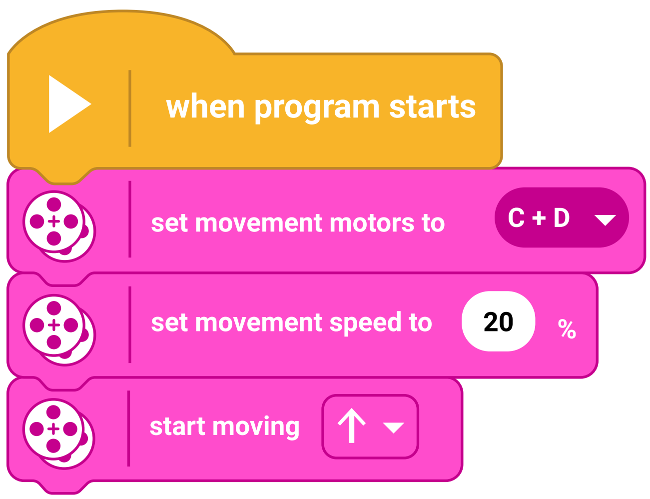
Students can code their robot to stop using each of the three sensors.

Facilitator Tips

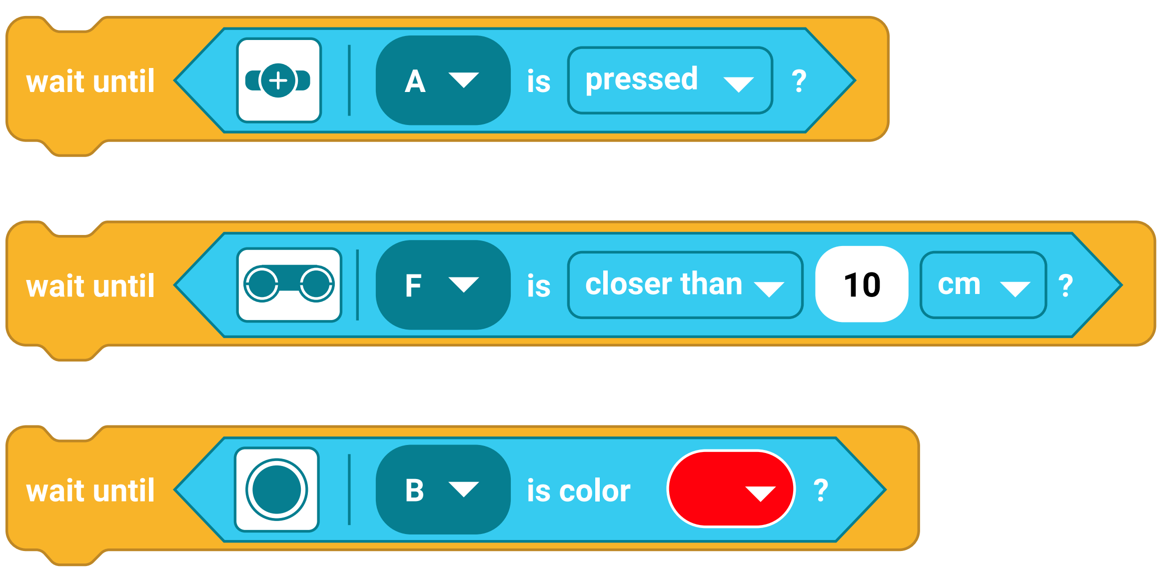
- The color sensor may have trouble detecting the color black. Use red or another color when coding the color sensor.

LEGO SPIKE Prime
Intro to sensors

1 Set your robot moving



2 Stop it with one of the sensors



Challenge!!!

Turn your robot around after it stops

You can use these blocks

stop moving

set yaw angle to 0

start moving right: 100

move ↓ for 5 cm

wait until yaw angle > 175

MICROBIOME EXTENSION 5: Block-Based Maze Challenge

5

Objective

SWBAT build a maze/obstacle course and program their SPIKE Prime robot to navigate through it.

Vocabulary

- **Debugging:** The process of identifying and fixing errors, bugs, or unexpected issues in code.

Framing Question

Can you use your coding skills to navigate your robot through a maze?

Materials

- Bluetooth-capable computers
- LEGO SPIKE Prime App installed on computers, or an internet connection to access the web app (https://spike.legoeducation.com)
- LEGO SPIKE Prime kit for each group of 2-4 students
- A maze or obstacle course for the students to run their robot through
- Colored tape (optional)

MICROBIOME EXTENSION 5: Block-Based Maze Challenge

Plan

1. Preparation

Build a simple maze for the students to run their robot through (Figure 2). You can use cardboard boxes and make it a formal maze. Or, you can create an obstacle course of found objects like trash cans. Just make sure the walls/obstacles are at least 6 inches high, so that the distance sensor (if being used by students) can detect them.

Figure 2. Example of a simple maze the robot must navigate.

2. Class time

Explain to the students that they will use their skills to code their robot to move successfully through the maze to the reward at the end, just like they coded their bacterium in Scratch. In Scratch, they were able to use the arrow keys to move their bacterium through the maze. Here, they can't communicate with their robot via the arrow keys. How will they help their robot navigate the maze? They can use the force sensor, distance sensor, color sensor, none of these, or all three.

As the students work, ask them which sensors they're using. How will the robot know it has reached the end?

MICROBIOME EXTENSION 5: Block-Based Maze Challenge

How will you determine if students met the objective?

Students have coded a robot to move through the maze given the rules which the class has decided upon.

Facilitator Tips

- Establish a set of rules that each group must follow. If their robot over-rotates on a turn, are its programmers allowed to straighten it? Can the robot accept input from its programmer? (For example, can students press the left and right buttons while in the maze?)
- There are several ways to move the robot through the maze. Students can program their robots to:
  - Make turns at specific distances.
  - Turn randomly until it reaches the reward.
  - Turn left or right based on the left and right buttons.
  - Turn left or right based on signals given to the color sensors.
  - Follow a line drawn for it to the reward.

MICROBIOME EXTENSION 6: Python Maze Challenge

6

Objective

SWBAT use the Python programming language to code their robot to navigate a maze.

Vocabulary

- **Python:** A popular text-based programming language
- **Modules:** Related code that is organized together. One exists for each SPIKE Prime component
- **Library:** A collection of modules that provide pre-written code to perform various tasks
- **Script:** A sequence of Python statements written to perform a specific task or solve a problem

Framing Question

Can I convert my block-based code into Python and program my robot to navigate a maze?

Materials

- Bluetooth-capable computers
- LEGO SPIKE Prime App installed on computers, or an internet connection to access the web app (https://spike.legoeducation.com)
- LEGO SPIKE Prime kit for each group of 2-4 students
- A maze or obstacle course for the students to run their robot through
- Colored tape (optional)

MICROBIOME EXTENSION 6: Python Maze Challenge

6

Plan

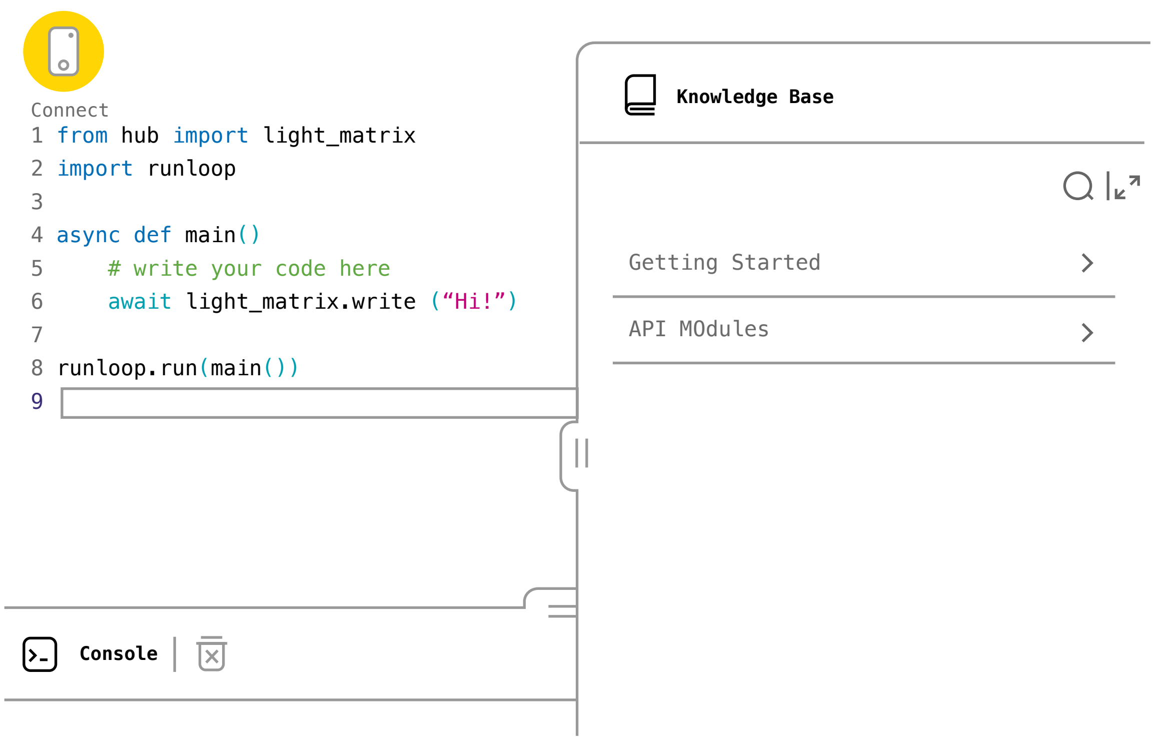
1. Introducing Python

Block-based coding is a lot of fun. Its “drag and drop” format makes syntax easy to remember and limits typos. However, professional programmers obviously need to use a use a text-based language to do more complex tasks. We are going to start using the Python language to code our SPIKE Prime robots. Python is one of the most-used coding languages in the world. According to Stack Overflow’s 2020 survey[2] of professional software developers, 41.6 % use Python. In the biomedical research world, most scientists use Python or R, with Python rapidly gaining in popularity.

2. Getting Familiar with Python

On the home page of the SPIKE Prime App, click “New Project” ⟶ “PYTHON” ⟶ “CREATE.”

The app will load the Python interface like this:



[2] Stack Overflow 2020 Developer Survey. https://insights.stackoverflow.com/survey/2020

3. Python code overview

Connect the hub to the computer by clicking the yellow “Connect” icon in the upper left. Some lines of code are already written. Clicking the yellow “Play” button will run the code, which will write “Hi!” on the light matrix.

Line 1 imports libraries into Python. The libraries teach Python how to interact with the hub. Lines 4-6 define the code, and then line 8 runs the code.

The “Knowledge Base” found in the right-side pull-out menu contains a library of information on how to code the SPIKE Prime in Python. Click “Getting Started” to run through some introductory modules.

Show students the basics of how to start a Python project in the SPIKE Prime App. Note that they will have to import the modules and pair their motors to start out with. Point out that they can use the Knowledge Base and the Microbiome Extension 6 LEGO SPIKE Prime Card to help them as they convert their block-based code. What differences do they notice? Is it harder or easier to code in Python?

4. Challenge

Challenge the students to code their robot in Python and move it through their maze.

MICROBIOME EXTENSION 6: Python Maze Challenge

6

How will you determine if students met the objective?

Students will successfully code in Python. They can use the Microbiome Extension 6 LEGO SPIKE Prime Card and the Knowledge Base to discover the syntax they will need.

Facilitator Tips

- Students can feel much more intimidated by text-based code. Encourage students to take their time and remind them that making errors and learning from them is a good thing.
- Strongly encourage the students to make use of the Knowledge Base to look up the syntax of the code they are working on.

Take notes for yourself!

```
#Hashtags are great for leaving one-line comments.
'''
Quotes before and after code keep it from running.
'''
```

Set Up

```
#Import your code libraries
import motor_pair                    #Imports code to synchronize the motors.
from hub import port                 #Imports info on what's in each port.
from hub import motion_sensor        #This senses the motion of the hub.
from hub import sound                #You need this for sound.
import force_sensor                  #Import your sensors.
import distance_sensor
import color_sensor
import color


async def main():
    motion_sensor.set_yaw_face(motion_sensor.FRONT) #Sets the default yaw angle.
    motion_sensor.reset_yaw(0)                      #Sets hub's current position as angle 0.
    motor_pair.pair(motor_pairPAIR_1,port.C,port.D) #Sets PAIR_1 as the motors in port C & D.
```

Move

```
await motor_pair.move_for_degrees(motor_pair.PAIR_1,360,0,velocity=200)

                                     #Moves the wheels one 360° turn in direction 0
                                     at 200 degrees/second. You can also
                                     move_for_time().
```

Turn

```
#turning left
while motion_sensor.tilt_angles()[0]<900:    #While a turn is less than 900 decidegrees
                                             (90°)...
    motor_pair.move(motor_pair.PAIR_1,-100)  #...Move left.(Steering goes from left -100 to
                                             right +100.)
motor_pair.stop(motor_pair.PAIR_1)           #Then it's going to stop when it hits 90°.

'''
How do you go right? Hint: Turning right is making a negative turn. Also, don't forget
to reset your yaw angle.
'''
```

Sensors

Distance Sensor

```
while True:                                                  #This is a loop.
    if distance_sensor.distance(port.F)>100 or distance_sensor.distance(port.F)==-1:
                                                             #If no object is closer than than 100mm,
                                                             or if there is an error signal...

        motor_pair.move(motor_pair.PAIR_1,0)                 #Move straight forward.
    else:
        motor_pair.stop(motor_pair.PAIR1)                    #Stop.
        break                                                #Break out of while loop.
motor_pair.stop(motor_pair.PAIR_1)                           #Stop moving.
```

Force Sensor

```
motor_pair.move(motor_pair.PAIR_1,0,velocity=200) #Move.
while True:                                       #Stop moving if the force sensor is
    if force_sensor.pressed(port.A):              pressed.
         motor_pair.stop(motor_pair.PAIR_1)
```

Color Sensor

```
while True:                                       #Stop moving if the color sensor sees red.
    if color_sensor.color(port.B)==color.RED:
          motor_pair.stop(motor_pair.PAIR_1)
```

Challenge: Put it all together to move your robot through the maze!