Scratch: Lessons for Hummingbird Duo

The future of Scratch 2.0 offline and ScratchX is uncertain.
We encourage you to use Snap! with your Hummingbird Kit to ensure long-term support.
Before you can build an awesome robot with your Hummingbird, you need to know how to use all the parts.
This lesson will first show you how to use the Hummingbird outputs, the lights and motors. These components are called outputs because programs in Scratch send commands to them to make something happen. Then you will learn how to use the Hummingbird inputs, the sensors that provide information to the robot about its environment.

Keys Available

Keys are available to educators.
GET ACCESS

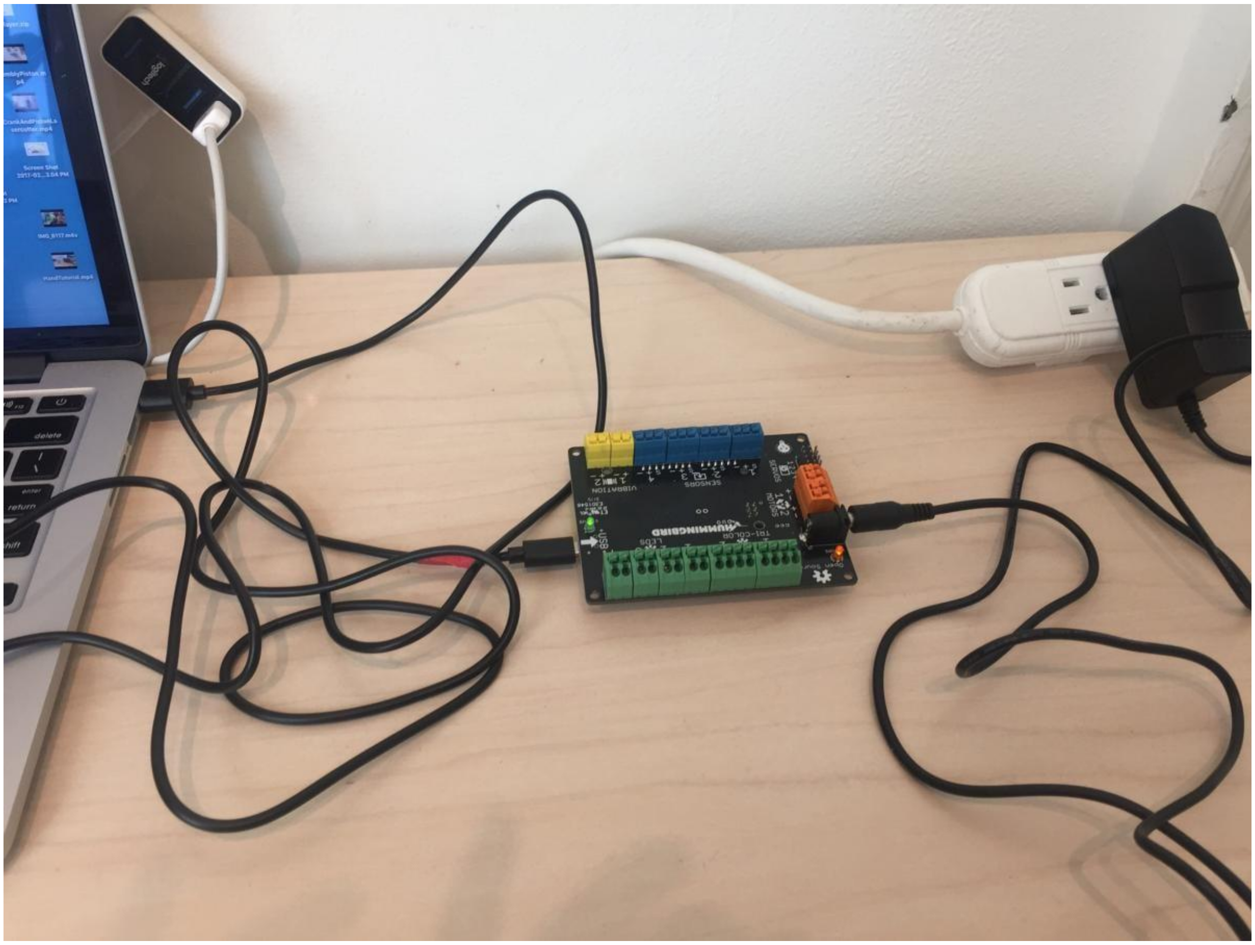
Getting Started

Using the Outputs

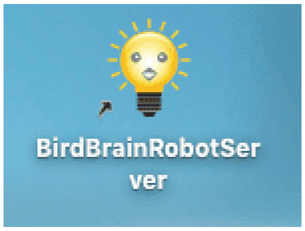
Using the Inputs

Getting Started

Start by watching the videos shown below.
The first will show you how to attach components to the Hummingbird board, and the second will show you how to connect the Hummingbird to the computer and how to open Scratch when using the Hummingbird.

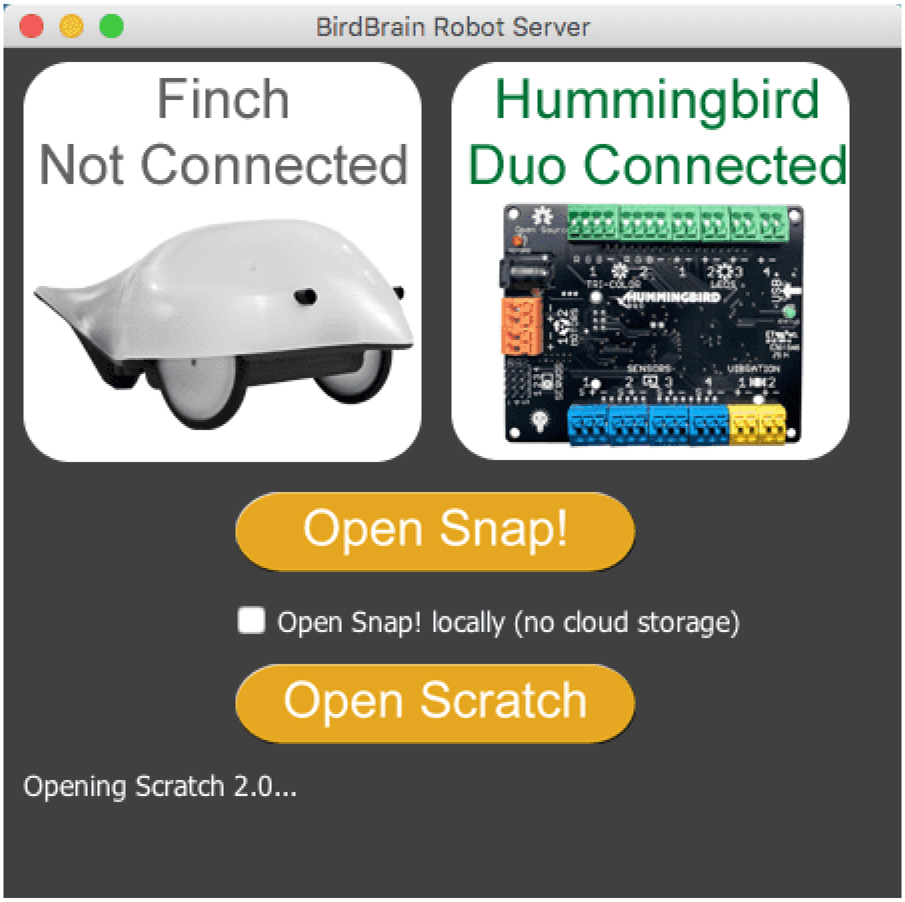


Use the USB cord to connect the Hummingbird to your computer, and connect the AC power adapter.



Then open the BirdBrain Robot Server (or the Hummingbird Connection App, if you are using a Chromebook).

This program must remain open if you are programming in Scratch. It is like a translator between Scratch and your Hummingbird board.



Finch
Not Connected

Hummingbird
Duo Connected

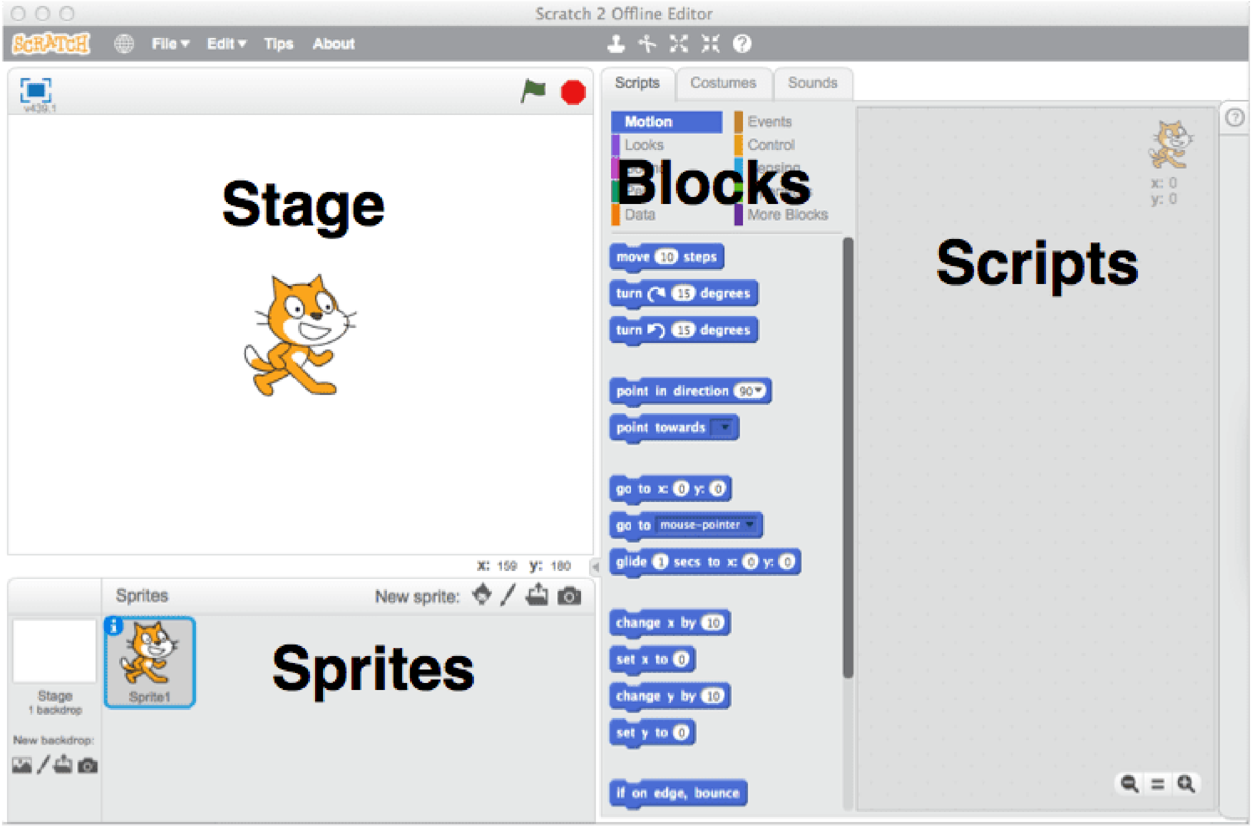
□ Open Snap! locally (no cloud storage)

Opening Scratch 2.0...

The Birdbrain Robot Server should tell you that the Hummingbird is connected. Click Open Scratch.

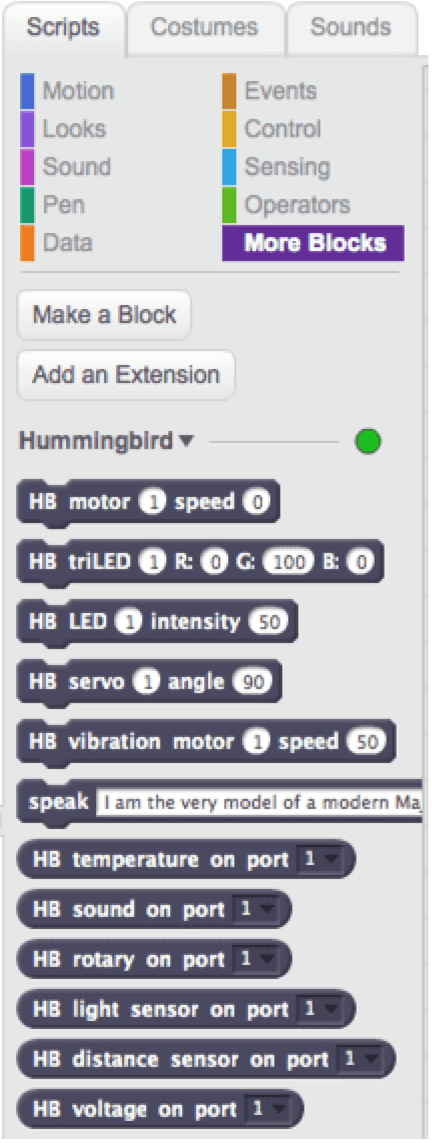
NOTE: On a Mac, the USB cord for the Hummingbird must be plugged in before you open the Birdbrain Robot Server.

Programming in Scratch



This is the Scratch window. It has a few different parts.

To write a program, you will drag blocks from the Blocks area to the Scripts area. In Scratch, programs are also called scripts. You can use Scratch to write scripts that use the Hummingbird or scripts that use cartoon characters called sprites. These lessons will focus on the Hummingbird. To learn more about using sprites to create an animation that runs in the Stage area, please see the Resources section at the end of this lesson.

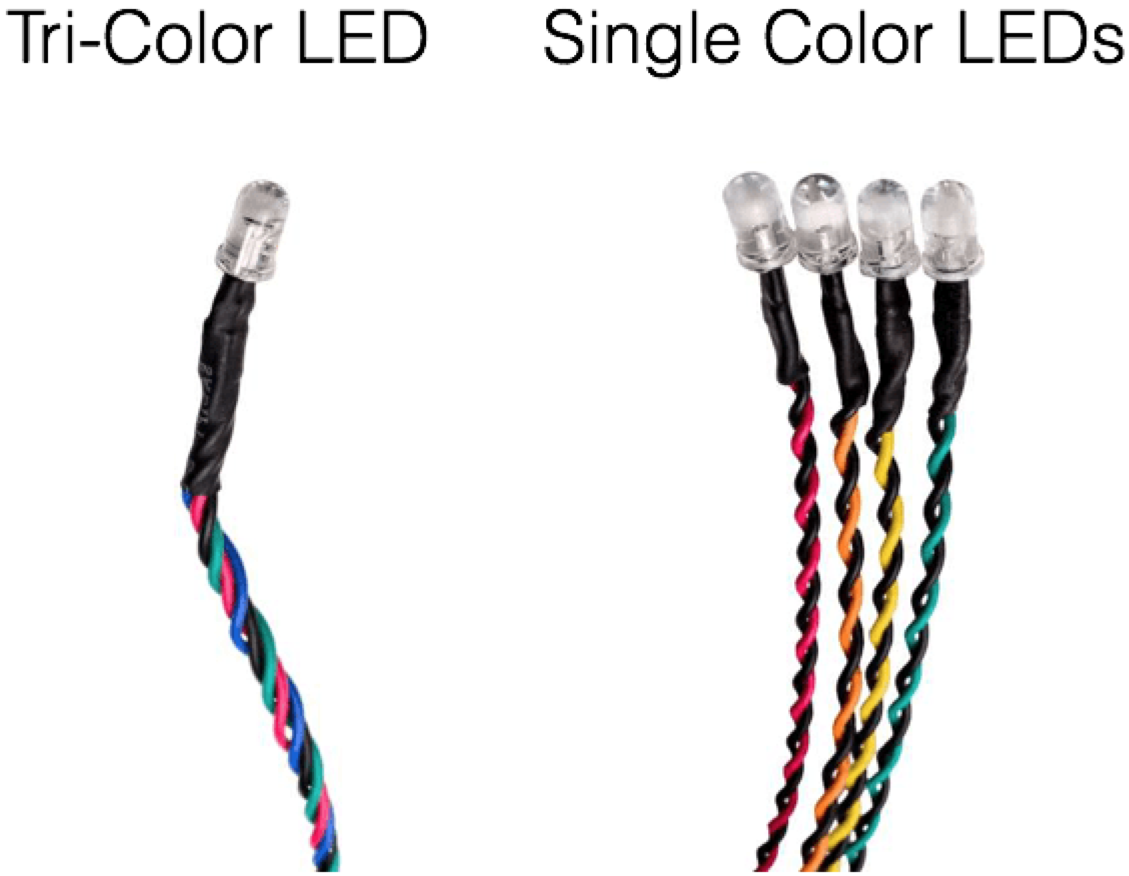


The Blocks area in Scratch contains 10 different menus – Motion, Looks, Sound, etc. Click on the different menus to get an idea of the blocks that they contain. All the Hummingbird blocks are located in the More Blocks menu in Scratch.

Saving Your Work

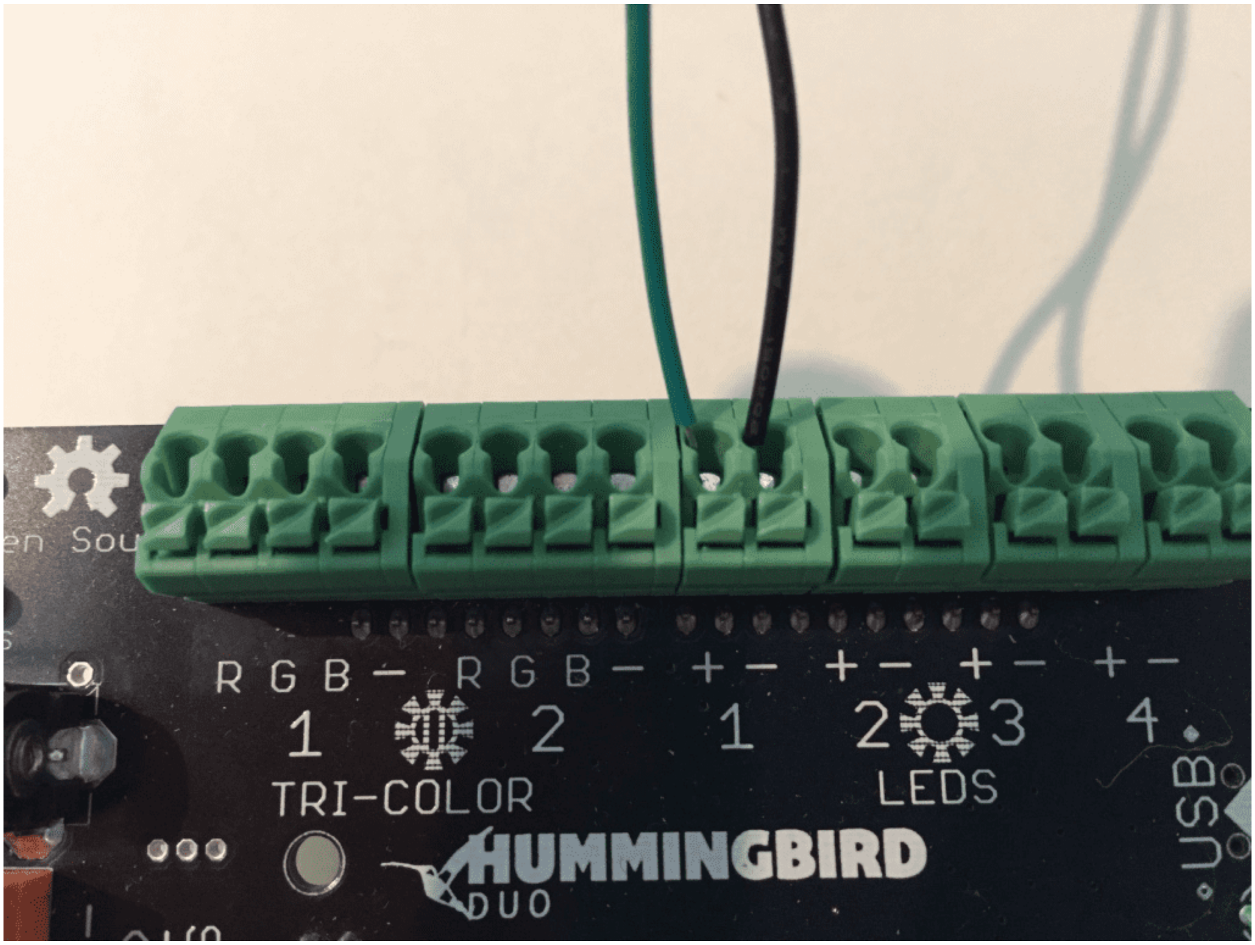
It is very important to save your work often! Otherwise, you might lose something important.
To save a new project in Scratch, click on File and then Save as (Save Project for Chromebooks using ScratchX). Give your project a name and then click Save. Once your project has a name, you can save it by going to File and then Save.
DO NOT name your Scratch file HummingbirdStart. This will overwrite the blank file that should appear when you open Scratch from the Birdbrain Robot Server.

Using the Outputs



First, you will learn to use the lights, which are also called LEDs. The Hummingbird kit contains two types of LEDs, single color LEDs and tri-color LEDs. Single color LEDs have two wires, while tri-color LEDs have four wires.

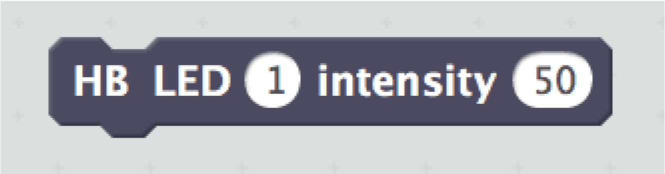
Module 1: Single Color LEDs



A single color LED can be connected to one of four ports labeled “LEDS” on the Hummingbird board. Since you have four ports, you can connect up to four single color LEDs. The black LED wire should be connected to the ‘-’ terminal, while the colored wire should be connected to the ‘+’ terminal.

The HB LED block is used to control a single color LED. This block requires two numbers.The number on the left is the number of the Hummingbird port to which the LED is attached. This will be a number between 1 and 4. In the picture above, the LED is connected to LED port 1.

The second number in the HB LED block is the intensity of the LED. This is a number between 0 and 100. 0 means that the LED is off, and 100 means that the LED is at maximum brightness.

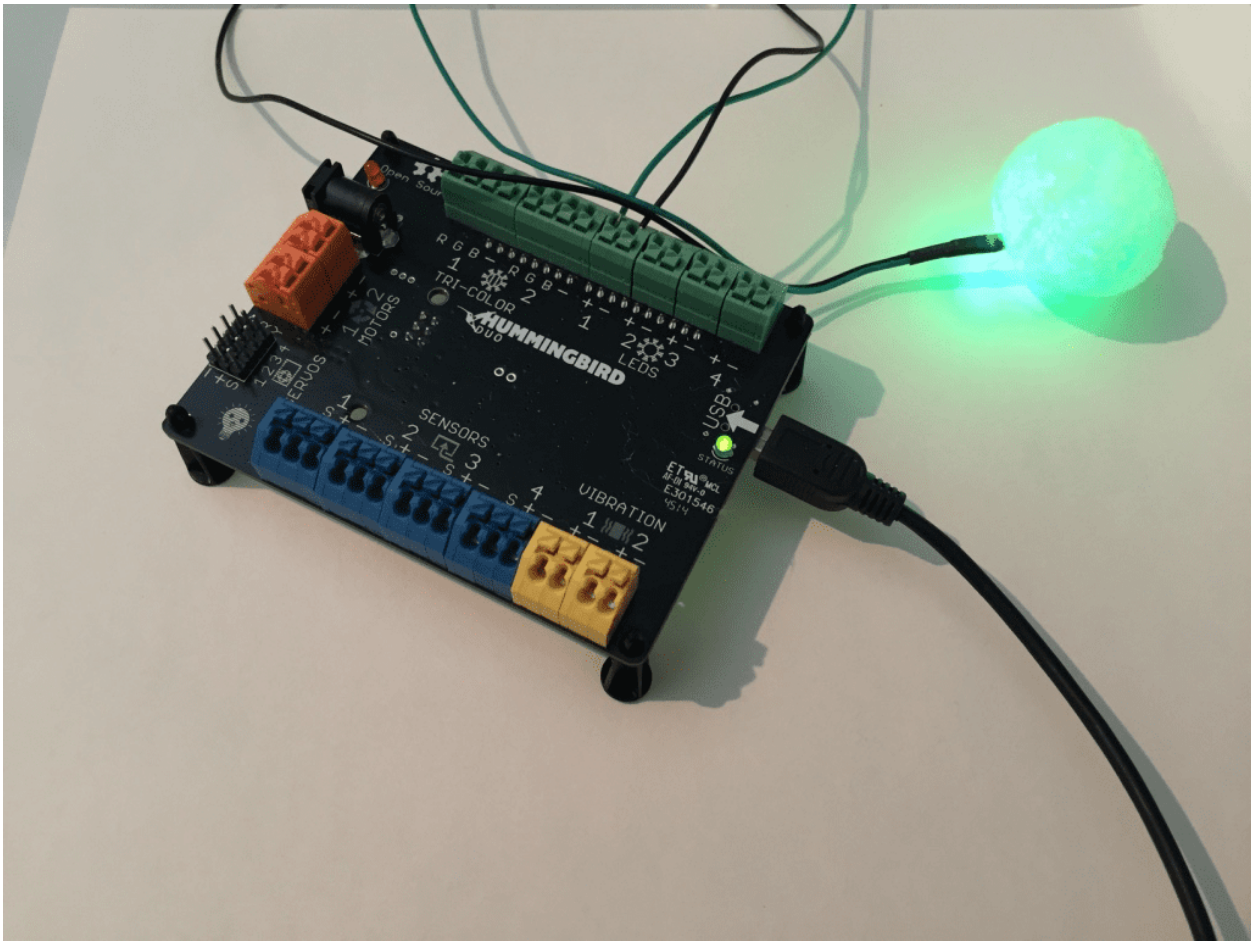


HB LED 1 intensity 50

Attach a single color LED to LED port 1. Then drag the HB LED block out into the Scripts area. This is your first program! Click on this block to run your program. The LED should turn on.

Exercise 1.1

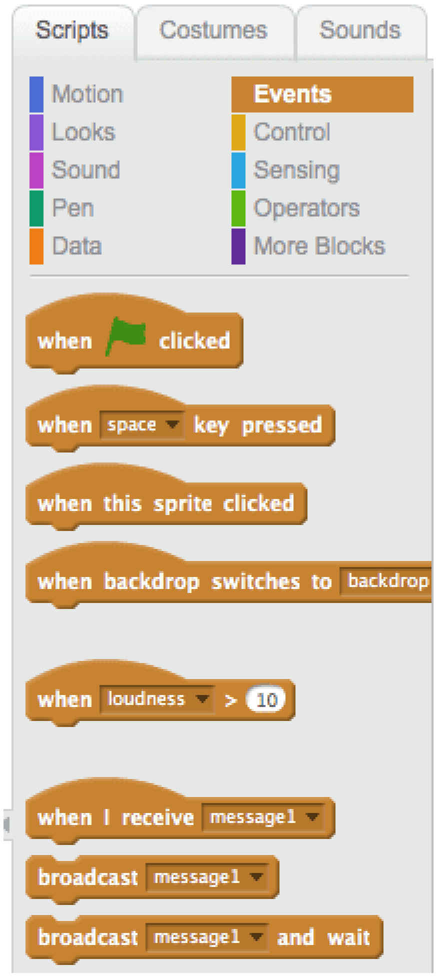
Change the 50 to 0 and click on the block again.
This should make the LED turn off. Change the 0 to 100 and click on the block. This is the maximum brightness of the LED. Try out other numbers between 0 and 100 to explore the different levels of brightness that are possible.



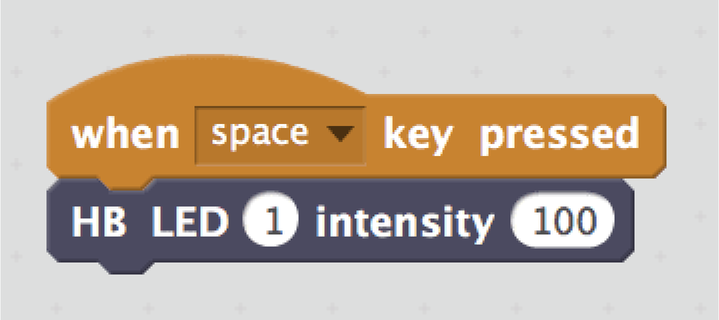
Building Tip

LEDs are quite small, but you can use a styrofoam ball to diffuse the light. This will make the light appear larger.

Events



So far, you have been running your program by clicking on it. You can also use an event to start a program. An event is an action that the computer can recognize. For example, you might press a key on the keyboard. Click on the Events menu.



The second block on this menu is when key pressed. Drag this block into your program until it connects to the top of your script.

Notice that the shape of this block shows you that it must be at the top of a script. It can't be connected below another block.

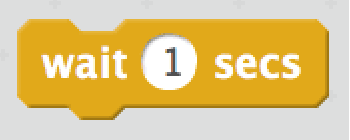
Now you can run your program by pressing the space bar. Try using the space bar to turn on the LED.

Click on the black triangle in the when key pressed block.
A menu will pop up that will allow you to select other keys. Modify your program so that the LED turns on when you press ‘o.’

Exercise 1.2

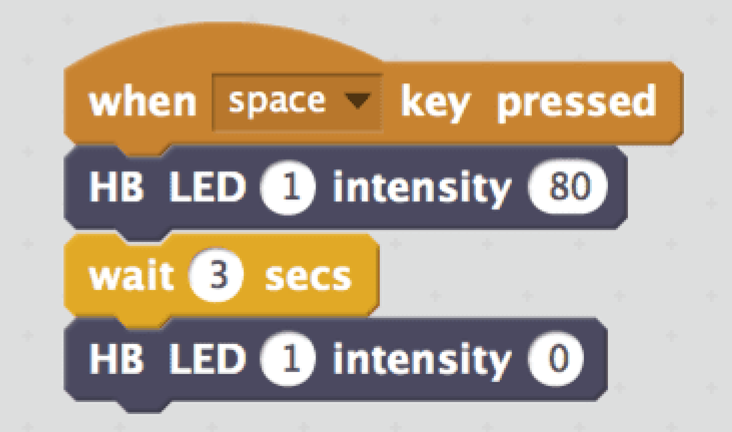
You may have already discovered that you can write multiple scripts in the Scripts area. Add a second script that turns the LED off when you press ‘x.’

Creating Longer Scripts



```
wait 1 secs
```

The wait block can be found on Control menu. This block pauses the program for the number of seconds shown in the block. This number can be a whole number or a decimal number.

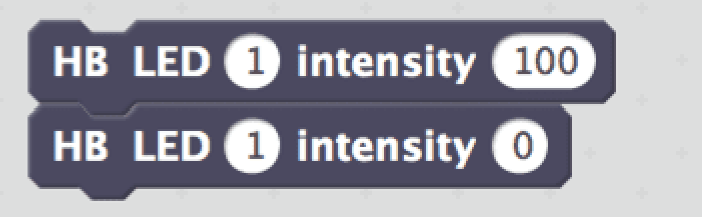


```
when space key pressed
HB LED 1 intensity 80
wait 3 secs
HB LED 1 intensity 0
```

You can use the wait block to turn the LED on for a certain period of time. For example, this program will turn the LED on for three seconds and then turn it off.

Exercise 1.3

Connect a second LED to LED port 2. Write a script that turns the first LED on and waits 2 seconds. Then the script should turn the second LED on for 2 seconds. Then the program should turn both LEDs off. Remember to save your work!



```
HB LED 1 intensity 100
HB LED 1 intensity 0
```

Programming Tip

When Scratch reaches an HB LED block, it sets the LED and moves immediately to the next block. This means that if you do not have a wait block between two commands for the same

LED, you may not see the effects of the first block. For example, you may not see the LED turn on when you run this script. Place a wait block between the HB LED blocks in order to see the LED turn on and then off.

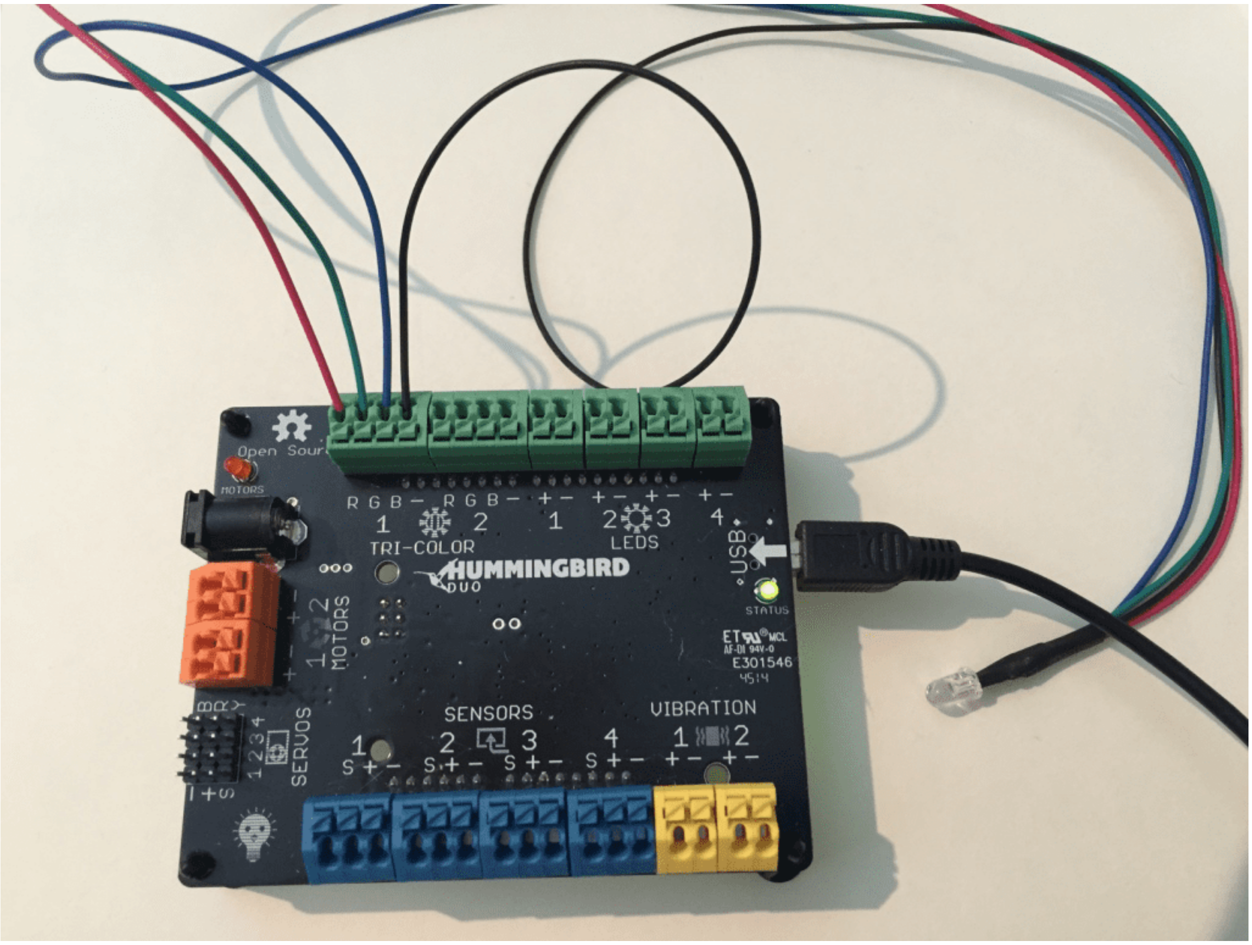
Exercise 1.4

Write a program that turns two LEDs on at different intensities, waits five seconds, and then turns both LEDs off.

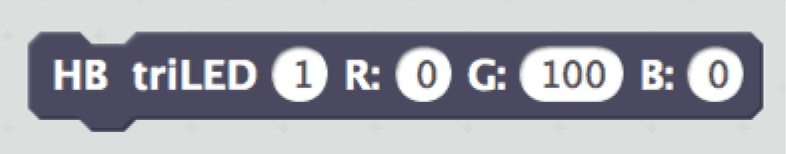
Programming Tip

It is good programming practice to always turn off your LEDs at the end of the program. They may go off automatically after a few seconds (particularly on Chromebooks), but this can be unpredictable.

Module 2: Tri-Color LEDs

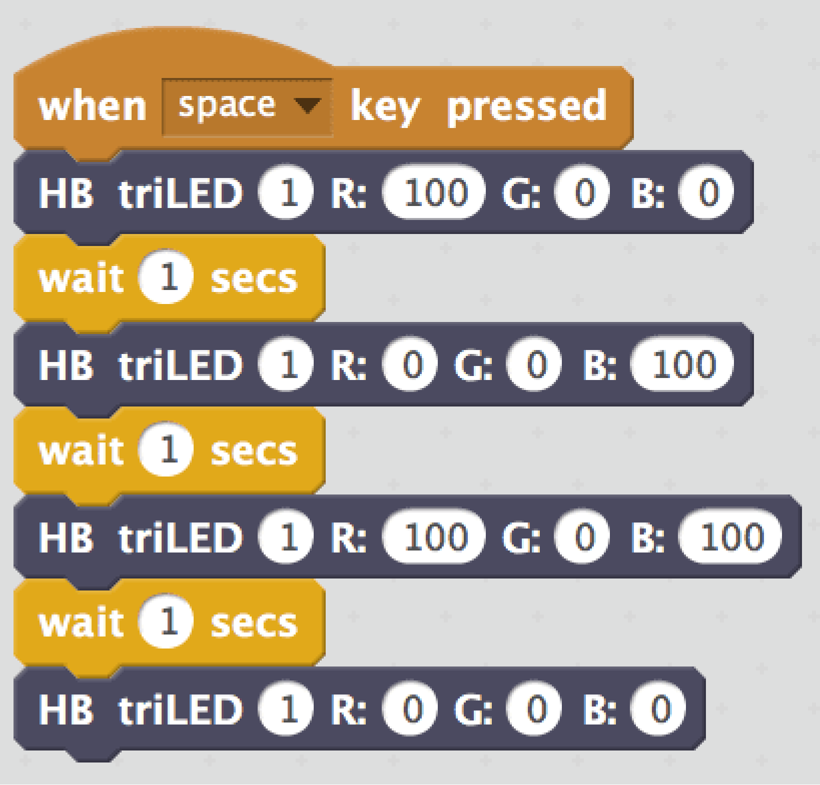


A tri-color LED can be attached to one of the two ports labeled “TRI-COLOR” on the Hummingbird board. The four wires of the tri-color LED are red, green, blue, and black. The black wire should be connected to the ‘-’ terminal. The red wire should be connected to the ‘R’ terminal, the green to the ‘G’ terminal, and the blue to the ‘B’ terminal.



```
HB triLED 1 R: 0 G: 100 B: 0
```

The tri-color LED actually has three tiny lights inside it. One is red, one is green, and one is blue. This is important for programming the tri-color LED. The HB triLED block is used to control a tri-color LED. This block requires four numbers. The number on the left is the number of the port to which the tri-color LED is attached (1 or 2). The other numbers are labelled R, G, and B. R controls the amount of red light from 0 (none) to 100 (maximum brightness). G and B control the amount of green and blue light, respectively, from 0 to 100. As an example, this block makes a tri-color LED in port 1 turn green.



```
when space key pressed
HB triLED 1 R: 100 G: 0 B: 0
wait 1 secs
HB triLED 1 R: 0 G: 0 B: 100
wait 1 secs
HB triLED 1 R: 100 G: 0 B: 100
wait 1 secs
HB triLED 1 R: 0 G: 0 B: 0
```

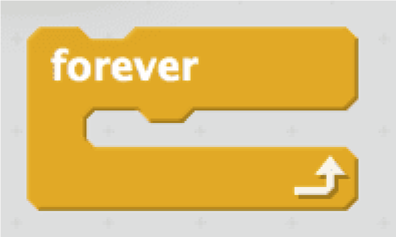
Exercise 2.1

What do you think this script will do? Make a prediction and then try it out. How do you turn the tri-color LED off?

Exercise 2.2

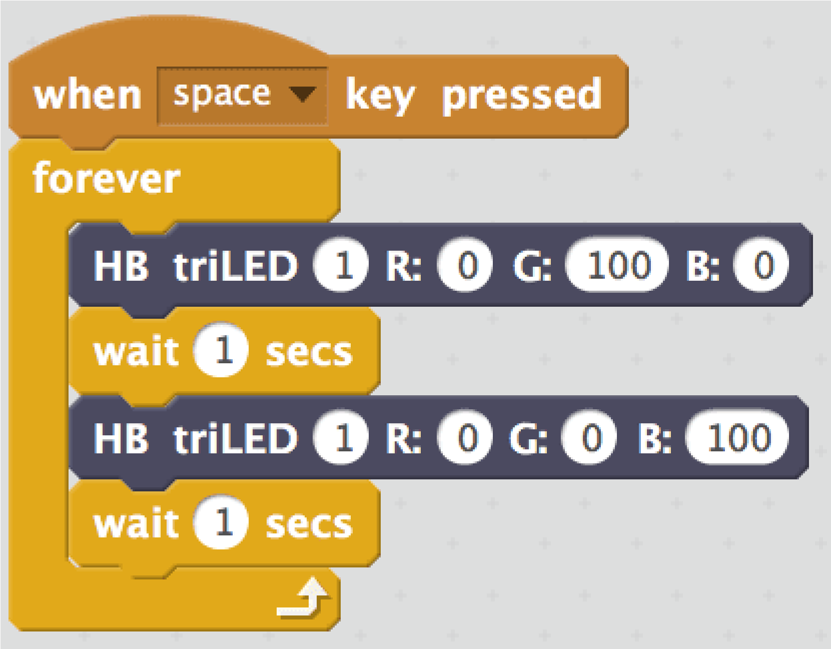
Write four scripts to turn the tri-color LED four different colors. Each script should use a when key pressed block and a HB triLED block. For example, the ‘a’ key might make the LED aqua.

Loops



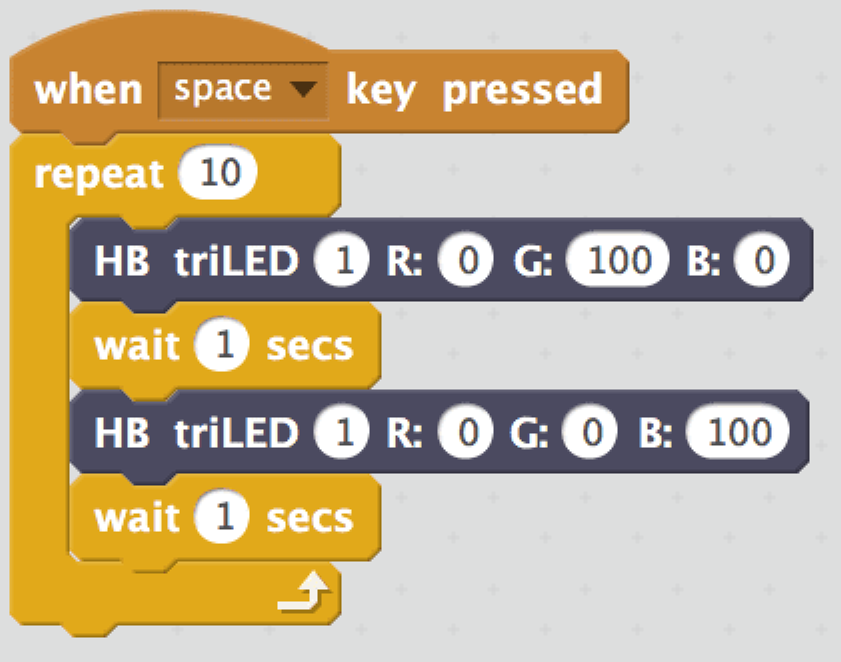
```
forever
```

Think about how you might turn an LED on and off repeatedly. One way to do this would be to use a long sequence of commands, but it is much simpler to use the forever block in the Control menu. This block is called a loop. A loop is a programming structure that repeats a portion of a program. Look at the Control menu. Which of the other blocks do you think might be loops?



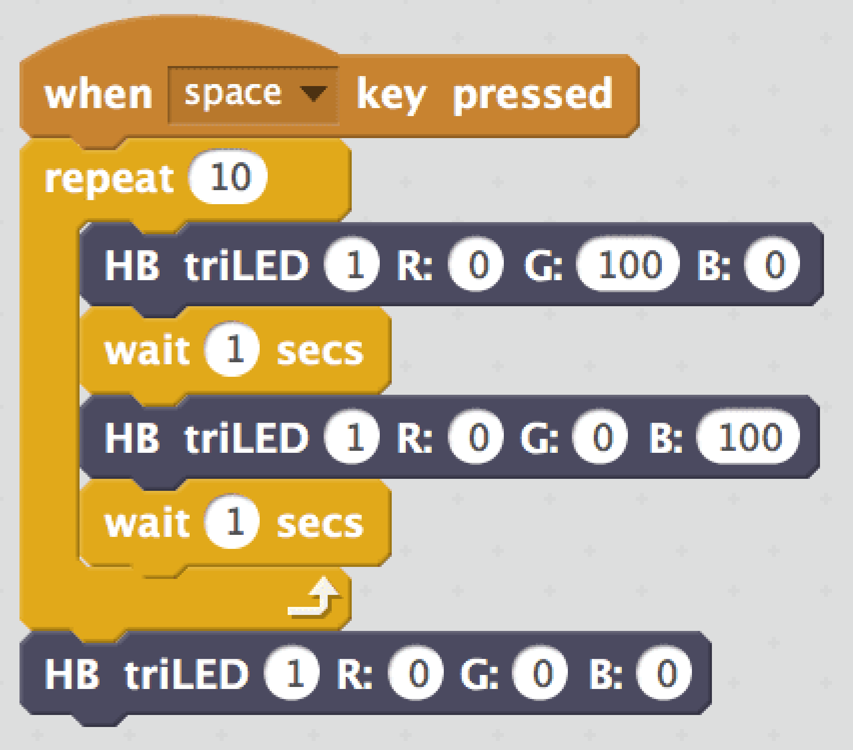
```
when space key pressed
forever
    HB triLED 1 R: 0 G: 100 B: 0
    wait 1 secs
    HB triLED 1 R: 0 G: 0 B: 100
    wait 1 secs
```

Other blocks can be placed inside the forever block. The forever block repeats the blocks inside it until you press the stop sign to stop the program. The loop shown blinks the tri-color LED from green to blue. The loop repeats the four blocks inside it over and over. After the second wait block, the program immediately goes back to the first block inside the loop, the block that turns the LED green. What happens if you remove the second wait block?



```
when space key pressed
repeat 10
    HB triLED 1 R: 0 G: 100 B: 0
    wait 1 secs
    HB triLED 1 R: 0 G: 0 B: 100
    wait 1 secs
```

The repeat block is a loop that repeats the blocks inside it a certain number of times. For example, this program makes the tri-color LED change from green to blue ten times.



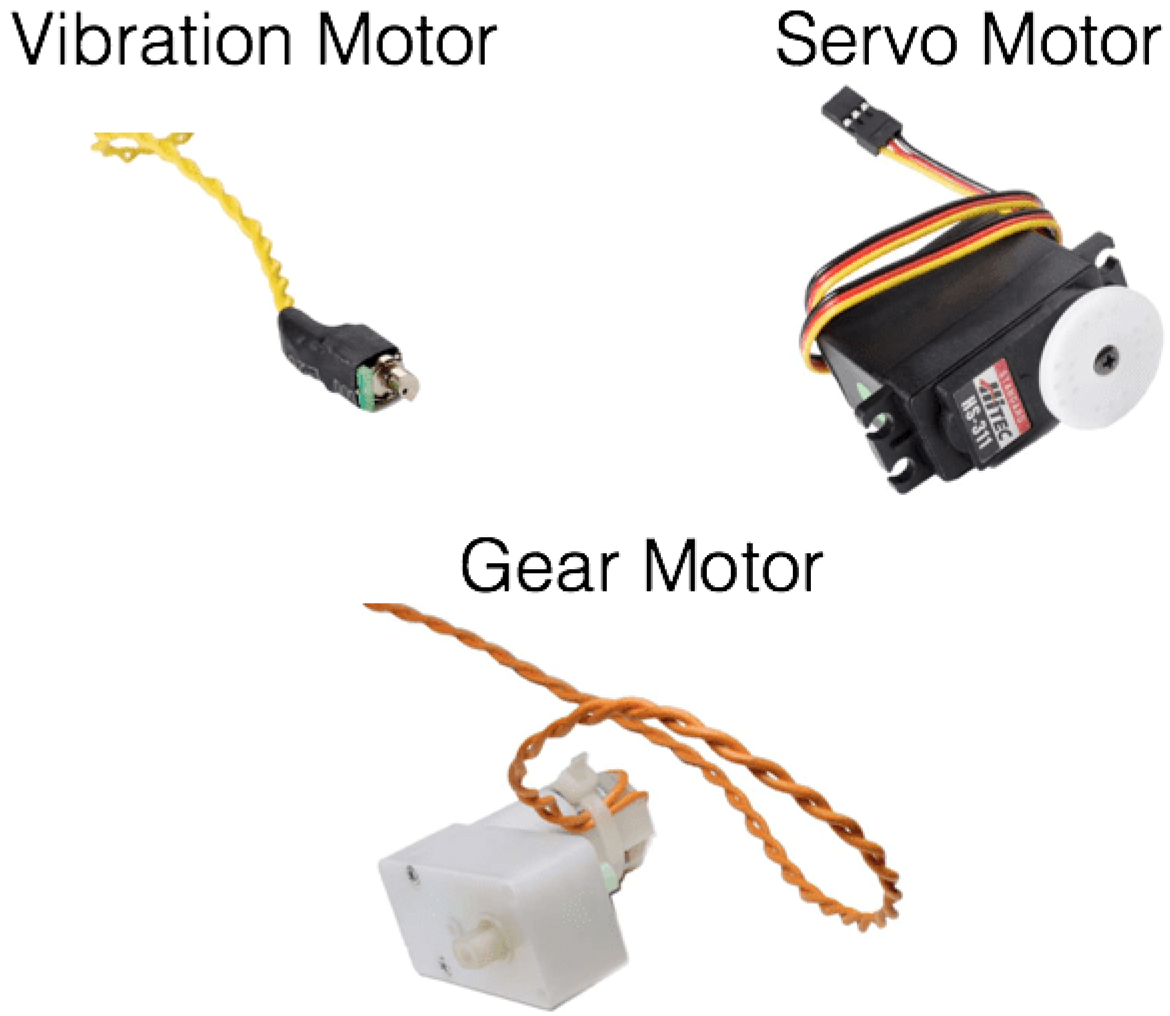
```
when space key pressed
repeat 10
    HB triLED 1 R: 0 G: 100 B: 0
    wait 1 secs
    HB triLED 1 R: 0 G: 0 B: 100
    wait 1 secs
HB triLED 1 R: 0 G: 0 B: 0
```

After the loop shown above ends, the light just stays blue. If you want to turn the LED off at the end of the script, you need to add a command below the repeat block.

Exercise 2.3

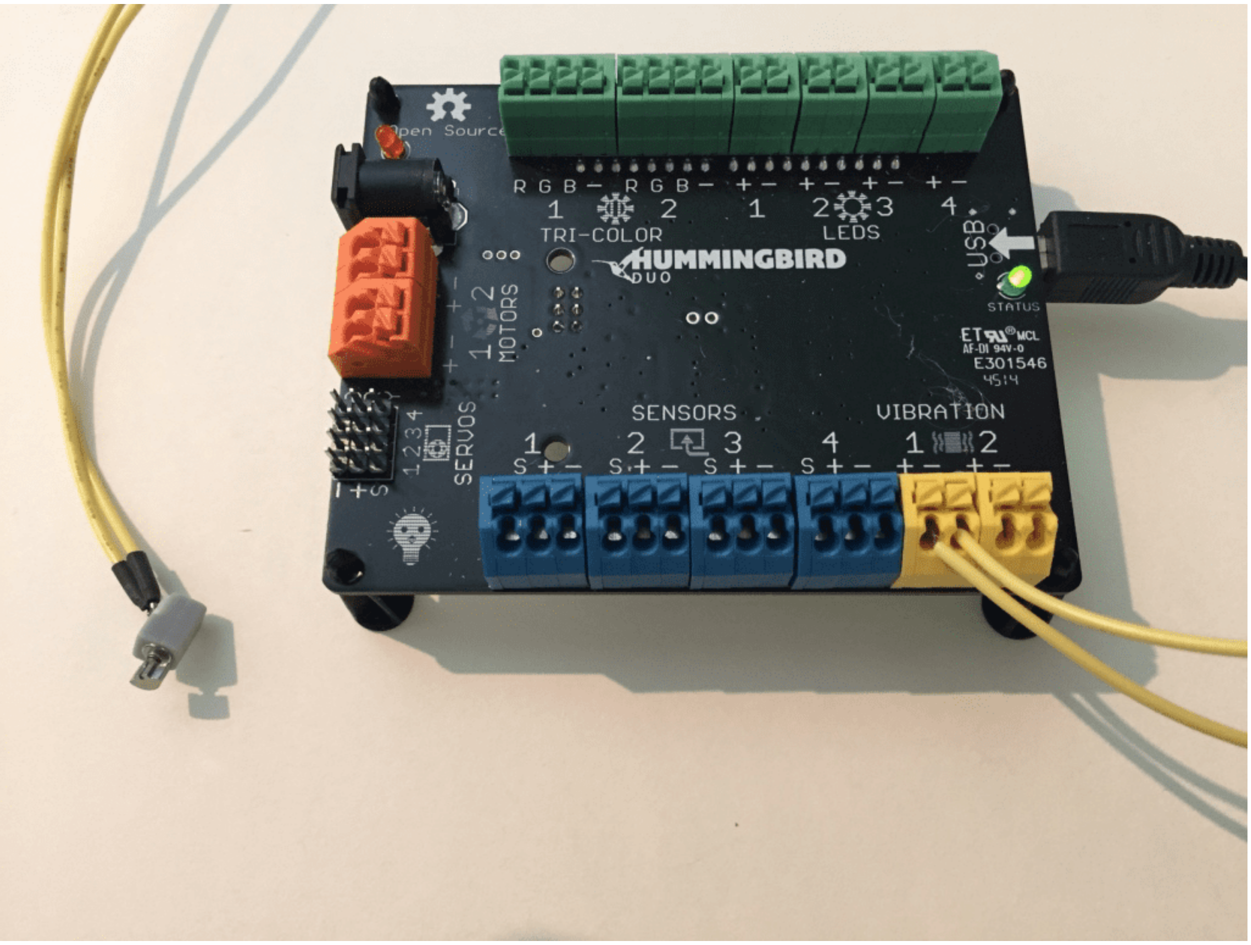
Make a tri-color LED blink on and off in your favorite color at least ten times in five seconds. The tri-color LED should blink evenly – in other words, it should be off for the same amount of time that it is on. Next, add a single color LED. The single color LED should be off when the tri-color is on and on when the tri-color is off.

Module 3: Motors

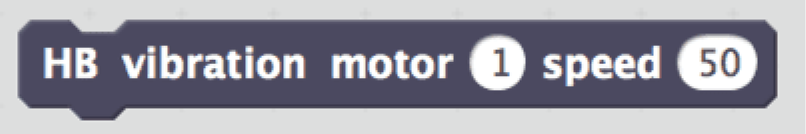


The Hummingbird kit works with three different kinds of motors: vibration motors, servo motors, and gear motors.

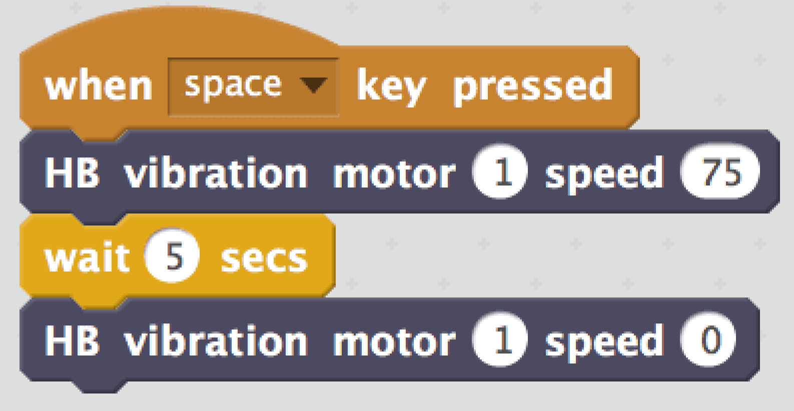
Vibration Motors



A vibration motor can be attached to one of the two ports labeled “VIBRATION” on the Hummingbird board. It does not matter which wire is attached to ‘+’ and which is attached to ‘-.’


HB vibration motor 1 speed 50

The HB vibration motor block is used to control the vibration motor. This block requires two numbers. The number on the left is the port attached to the vibration motor (1 or 2) and the second number is the speed of vibration from 0 to 100. 0 means no vibration, and 100 means maximum vibration.

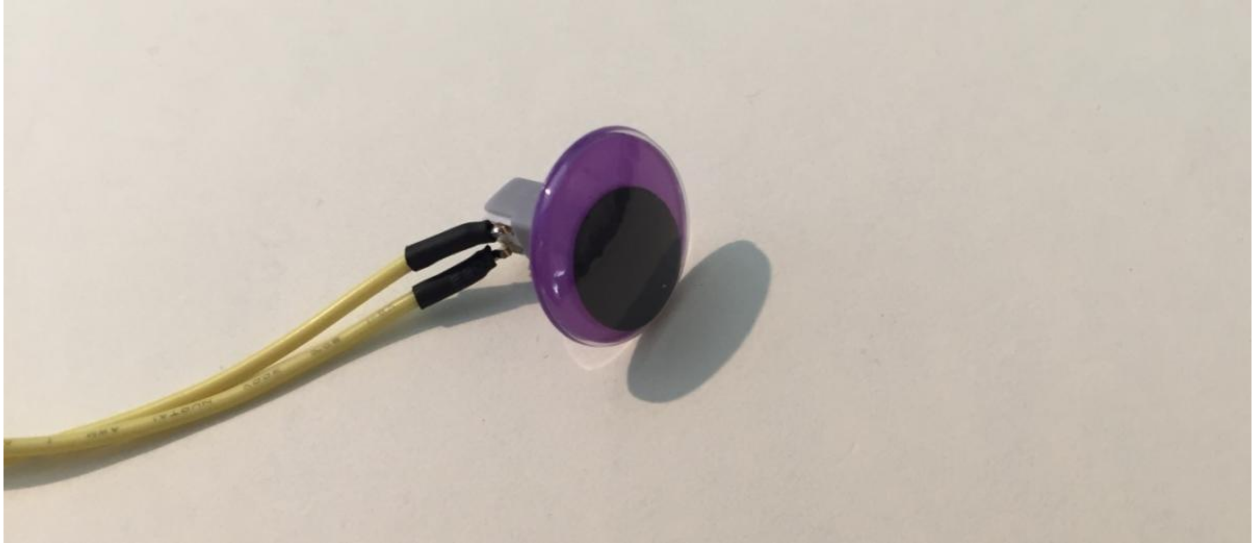


```
when space key pressed
HB vibration motor 1 speed 75
wait 5 secs
HB vibration motor 1 speed 0
```

As an example, this script turns the vibration motor on for five seconds and then turns it off.

Exercise 3.1

Write a program that turns both a vibration motor and a single color LED on for four seconds and then off for four seconds. Then place these commands within a loop to repeat them five times.



Building Tip

Small, very light items such as googly eyes can be attached to a vibration motor so that they move when the motor vibrates.

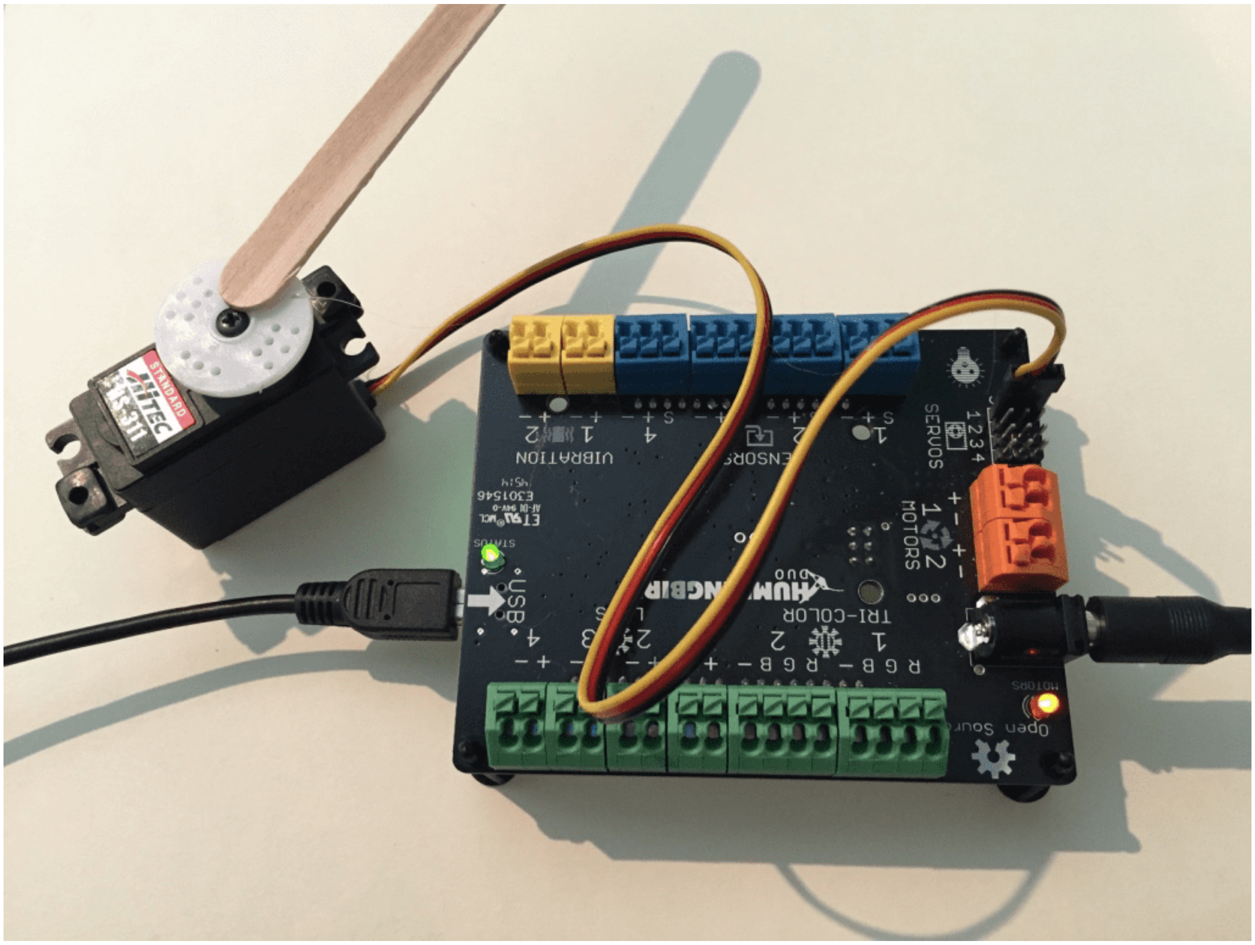
Servo Motors

A servo motor is a motor that moves to a particular angle. The Hummingbird servo motor can rotate to any angle from 0° to 180°.

Important Note: When using the servo motor, you must also use the AC power adapter (or a battery pack). Otherwise, the Hummingbird board will not have enough power to run the motor.

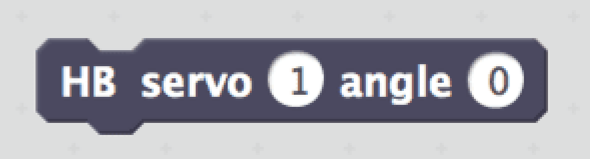
The servo motor has a small plug. This plug should be connected to one of the four sets of pins in the “SERVOS” section of the Hummingbird board. Each set of three pins is one servo port.

The black wire should be connected to the '-' pin, the red wire to the '+,' and the yellow wire to the 'S.'



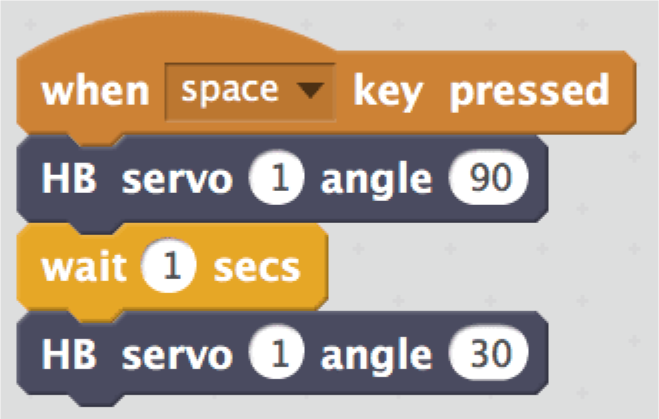
Use hot glue to attach a popsicle stick to the white plastic circle on the servo motor.

This will help you to see the movement of the servo motor more easily. If you don't want to use hot glue, you can tape a feather to the plastic circle.



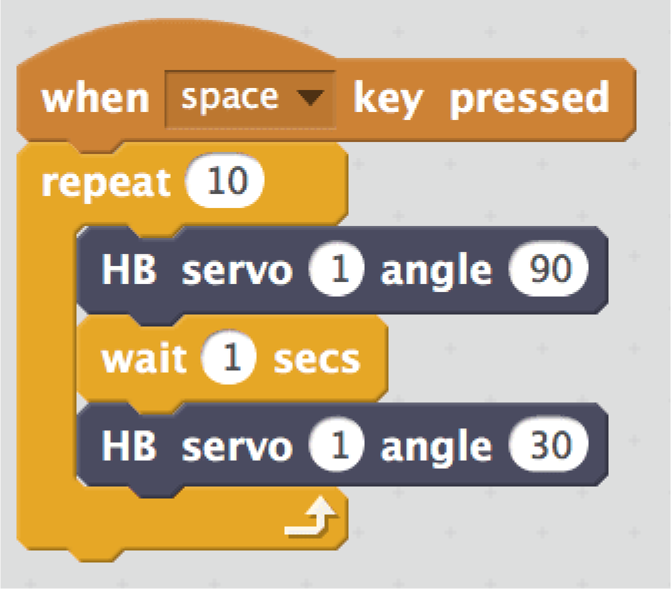
HB servo 1 angle 0

The HB servo block is used to move the servo motor. It requires two numbers. The number on the left is the port attached to the servo (1-4) and the second number is the angle (0° to 180°). The command below should move a servo motor attached to port 1 to 0°.



```
when space key pressed
HB servo 1 angle 90
wait 1 secs
HB servo 1 angle 30
```

If you write a script with more than one angle, make sure to put a wait block between the HB servo blocks. If you omit the wait block from the script below, the servo will not have time to move to 90° before the command to move to 30°.



```
when space key pressed
repeat 10
    HB servo 1 angle 90
    wait 1 secs
    HB servo 1 angle 30
```

Exercise 3.2

This script is supposed to move the servo motor repeatedly to two different angles, but there is a problem. Try out this script, and figure out how to correct it.

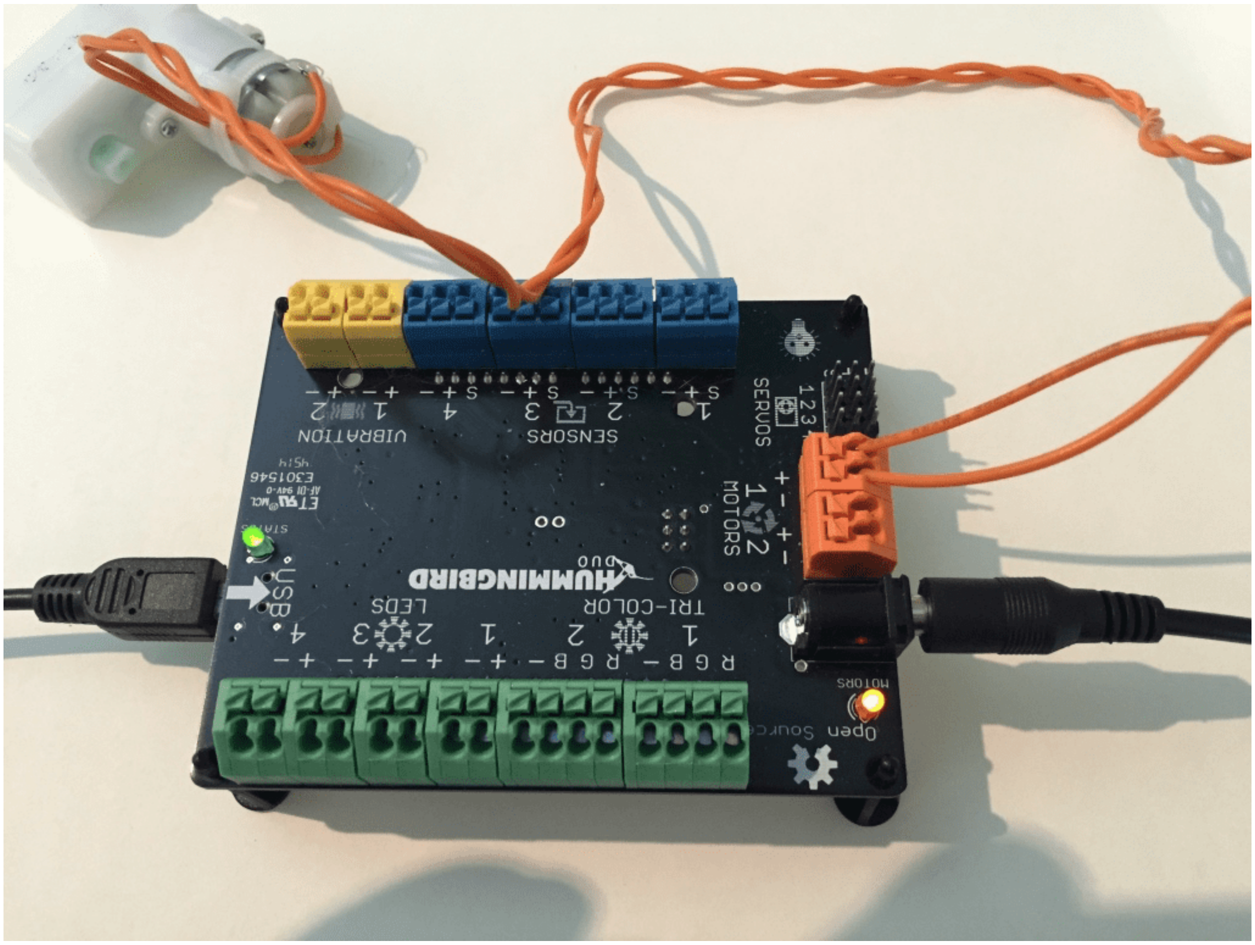
Exercise 3.3

Write a script that repeatedly moves the servo to five different angles. Make the tri-color LED a different color at each angle.

Building Tip

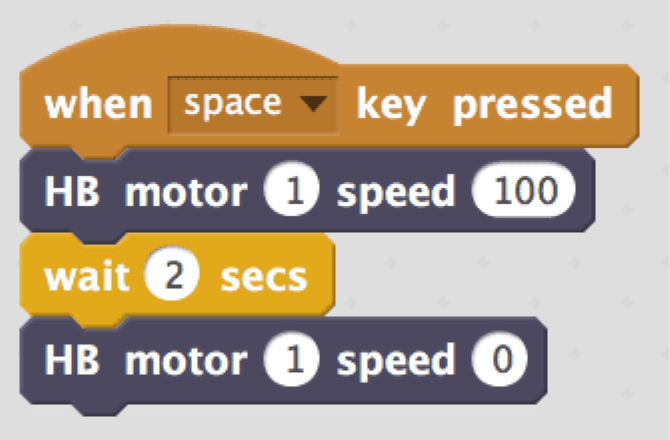
Before you start building with a servo, always set it to 90°. This will make sure that the servo can move both left and right once it is attached to your robot.

Gear Motors



The gear motor can be attached to either of the ports labeled "MOTORS" on the Hummingbird board. It does not matter which wire is attached to '+' and which is attached to '-.' When using the gear motor, you must also use the AC power adapter (or a battery pack).

Important Note: Do not cut the zip ties off your motors. Without the zip tie, the wires may break off the motor.



```
when space key pressed
HB motor 1 speed 100
wait 2 secs
HB motor 1 speed 0
```

The HB motor block is used to control the gear motor. This block requires two numbers. The number on the left is the port attached to the motor. The number on the right is labeled "speed."

This number can be any whole number from -100 to 100. As an example, this script turns the motor on for two seconds and then off.

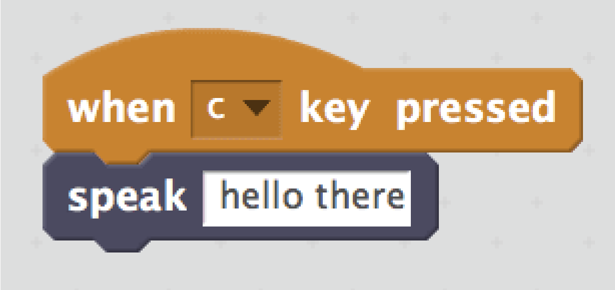
Exercise 3.4

Use a small screwdriver to attach a wheel to the motor so that you can see it move more easily (or just place the wheel on the motor without using a screw). Try several speeds between 0 and 100. Describe how the speed affects the motor. Try several negative speeds. What does it mean for the speed to be negative?
Important Note: You may notice that your motor rotates clockwise for positive numbers, while your neighbor’s rotates counterclockwise for the same number. This just means that you and your neighbor chose ‘+’ and ‘-‘ differently when you connected your motors. Either way is fine!

Exercise 3.5

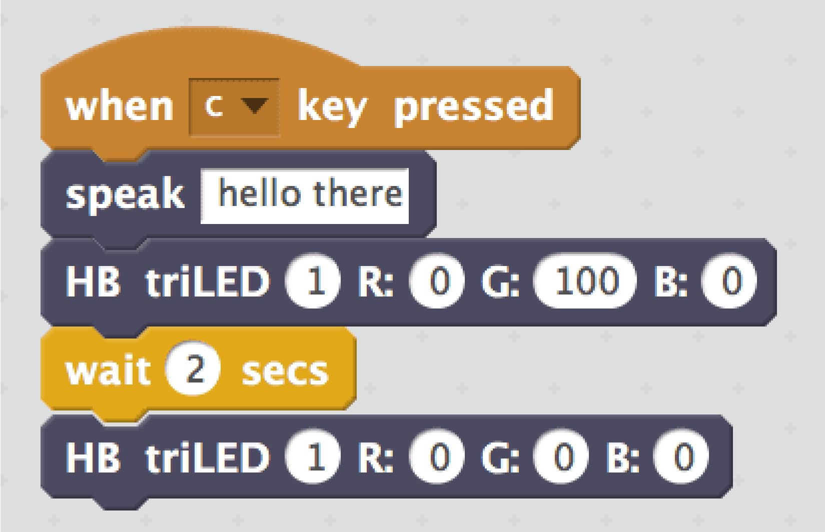
Write a program that rotates the motor first in one direction, then in the other. Next, use a loop to repeat these actions.

Module 4: Speak Block and Using All the Outputs



```
when c key pressed
speak hello there
```

The speak block can be used to make the computer say words or phrases. You can place words in this block, and the computer will say them over its speakers.



```
when c key pressed
speak hello there
HB triLED 1 R: 0 G: 100 B: 0
wait 2 secs
HB triLED 1 R: 0 G: 0 B: 0
```

When it runs the speak block, the computer starts to say the words and then moves on immediately to the next block. This means that in this script, the tri-color LED will turn on as soon as the computer starts talking.

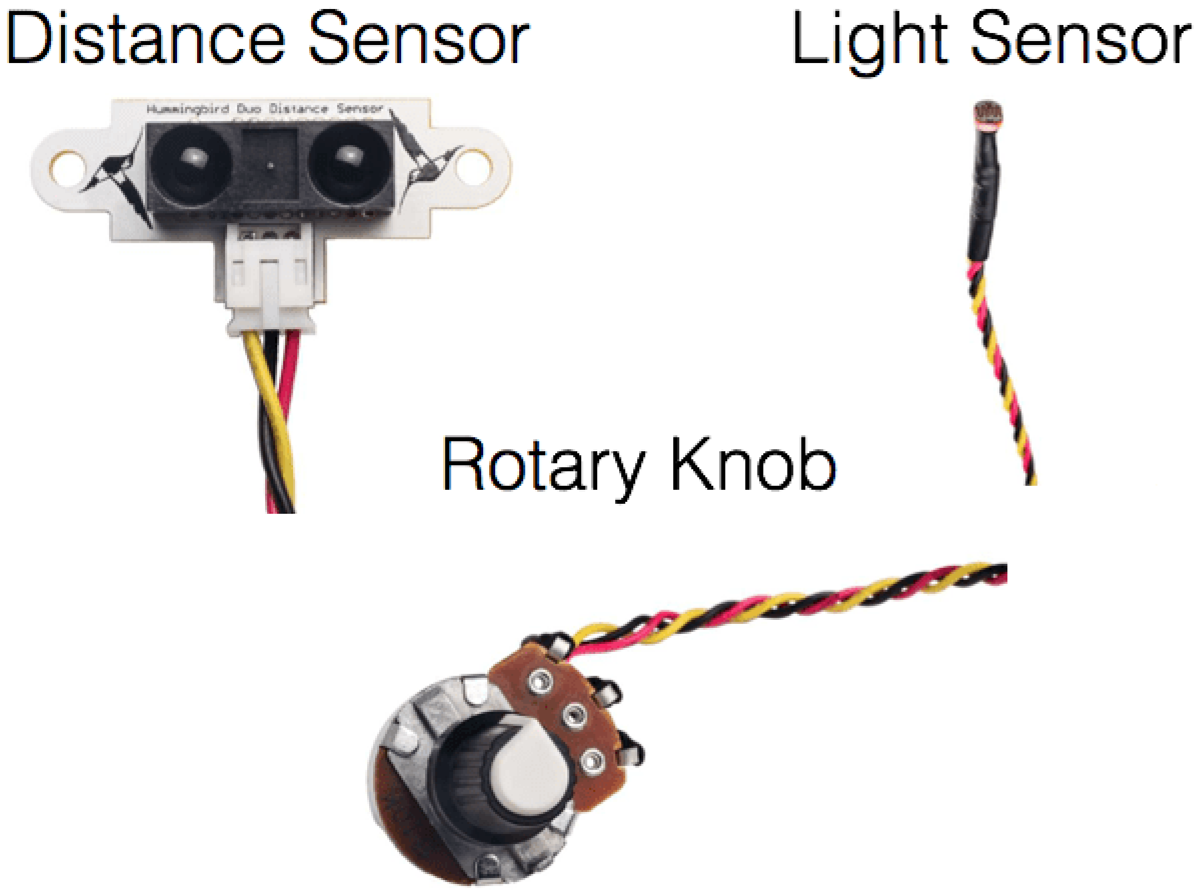
Exercise 4.1

Write a program that uses two speak blocks. What block do you need to place between your speak blocks?

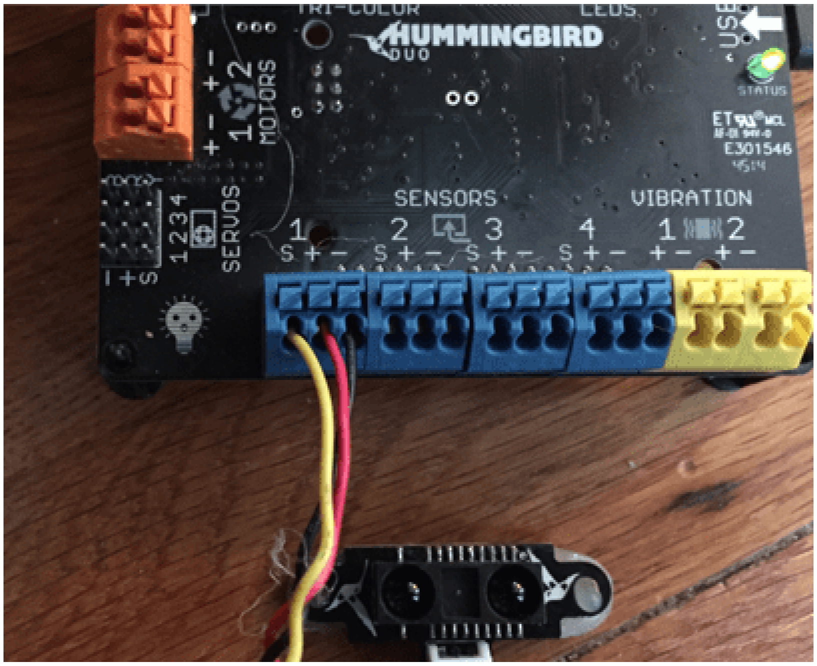
Exercise 4.2

Are you ready to use all the Hummingbird outputs? Write a program that includes at least two scripts that run when you press different keys, at least three LEDs, at least two different kinds of motors, and at least one loop.

Using the Hummingbird Inputs

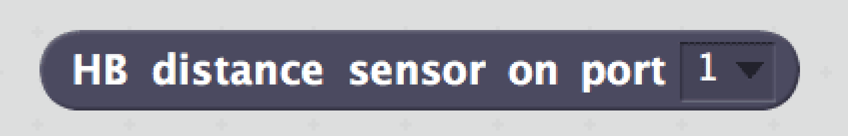


The Hummingbird LEDs and motors are output devices. Scripts in Scratch send commands to these devices to make something happen. The Hummingbird sensors, on the other hand, are input devices. They send information to the Scratch program. The Scratch program can use this information to make a decision or control a light or a motor.



All of the sensors are attached to the Hummingbird in the same way. A sensor can be attached to any one of the four ports labeled “SENSORS.” Each sensor has a red wire, a black wire, and a yellow wire. The red wire should be connected to ‘+,’ the black to ‘-,’ and the yellow to ‘S.’

Module 5: Distance Sensor

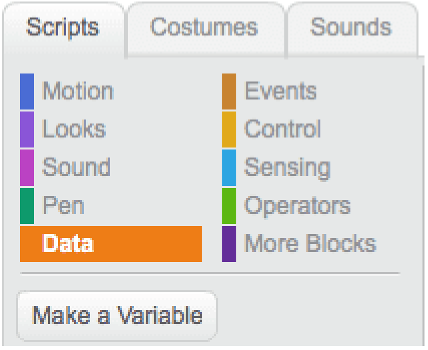


HB distance sensor on port 1

The distance sensor measures the distance to the closest object. Attach a distance sensor to the Hummingbird. Drag the HB distance sensor block into the Scripts area away from the other blocks. Make sure that the port number is set to the port attached to the distance sensor. Click on this block to see the current value of the sensor. Then hold something in front of the sensor and click the block again. Is the value of the distance sensor different?

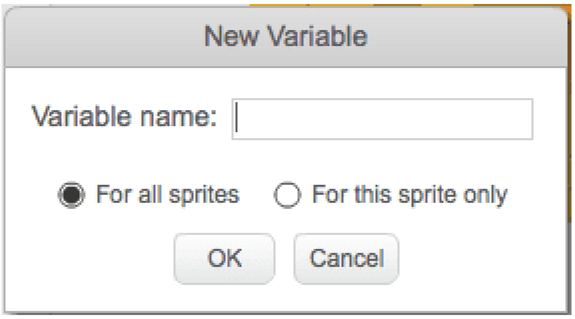
The distance sensor measures the distance to an object in centimeters. The value of the distance sensor is between about 8 cm and 100 cm. The sensor cannot detect objects very close to the sensor or very far from it. It works best in the range of 20 cm to 60 cm.

Creating a Variable



To see the value of the distance sensor change, it is helpful to create a variable. A variable is just a name that represents a value. You will now create a variable named “distance” that will hold the value of the distance sensor and display that value on the screen.

First, go to the Data menu. Then click Make a Variable.

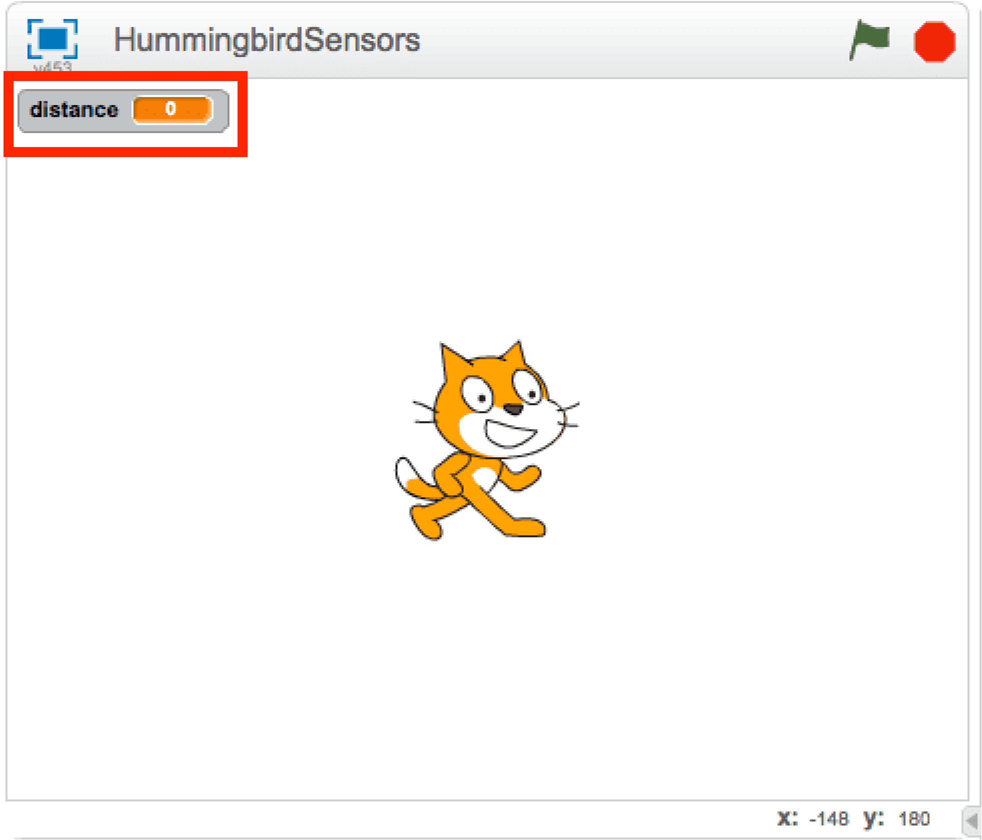


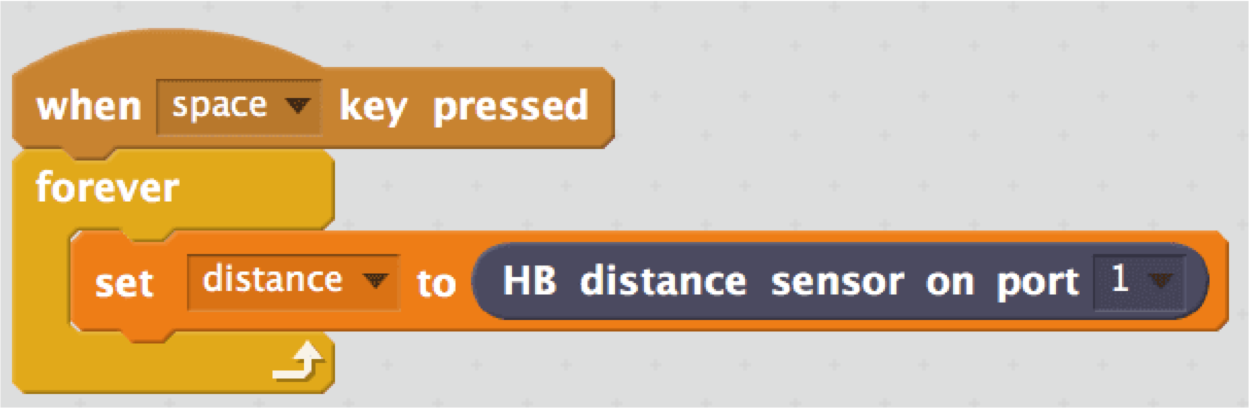
Variable name:

● For all sprites ○ For this sprite only

The New Variable window will appear. Call the variable “distance” and click OK.

You will notice that the Data menu looks different now. It contains some blocks that you can use to change the value of your variable. Also, the value of the variable is shown in the top-left corner of the stage.





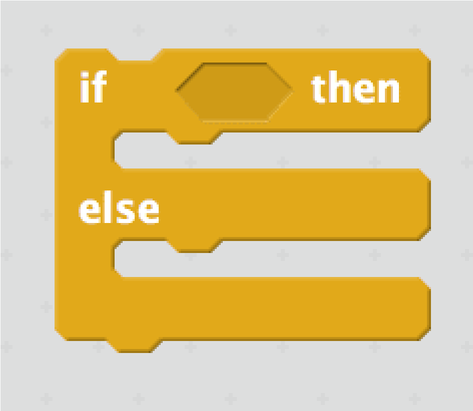
```
when space key pressed
forever
    set distance to HB distance sensor on port 1
```

We want to set the value of our variable equal to the value of the distance sensor. To do this, we can use the set to block. We will place this block inside a forever loop to continually set the variable to the value of the sensor.

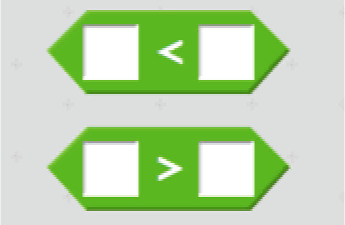
Exercise 5.1

Run the script above and watch the value of the distance variable change as you move an object back and forth in front of the sensor.

Making a Decision

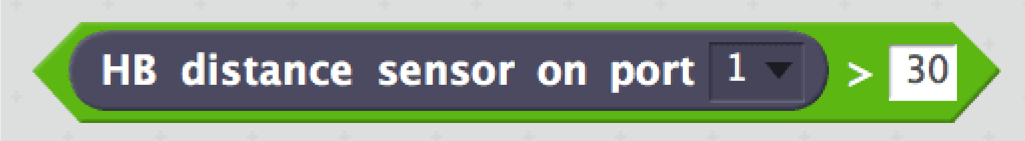


You can use the distance sensor to enable a robot to make a decision. The if then else block in Scratch is a decision block. This block is found on the Control menu. What do you notice about it?



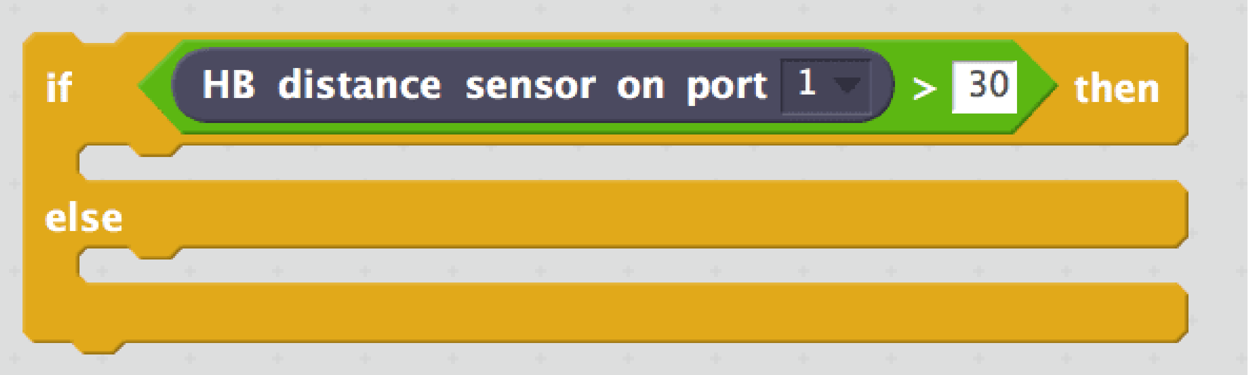
The if then else block has spaces in the middle that can hold other Scratch blocks. In addition, the if then else block contains a hexagonal space. This space requires a Boolean block. A Boolean block is a block that can be either true or false.

You can use the distance sensor to create a Boolean block using these two blocks, which are found on the Operators menu. Notice that these blocks match the shape of the space in the if then else block.



Create the Boolean block shown.

This block is true if the value of the distance sensor is greater than 30 cm, and false otherwise. The value that the Boolean block uses to make a decision is called the threshold. In this case, the threshold is 30 cm. Click on the green part of the block with an object in front of the sensor. Is the block false?



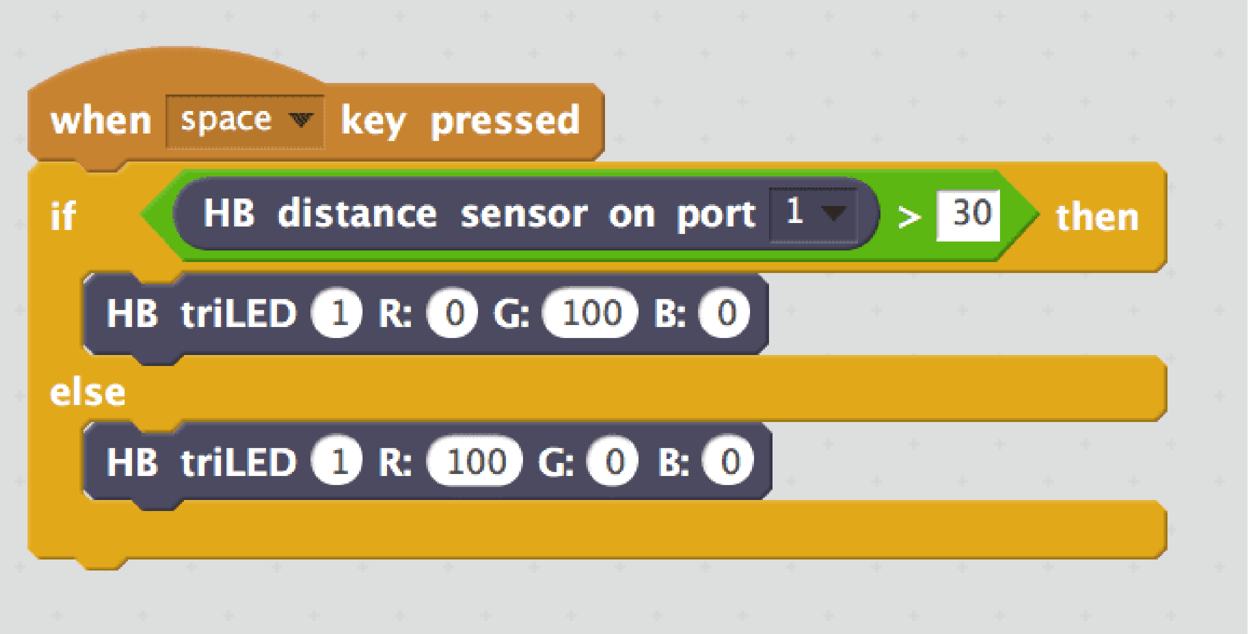
```
if < (HB distance sensor on port 1) > 30 > then

else

```

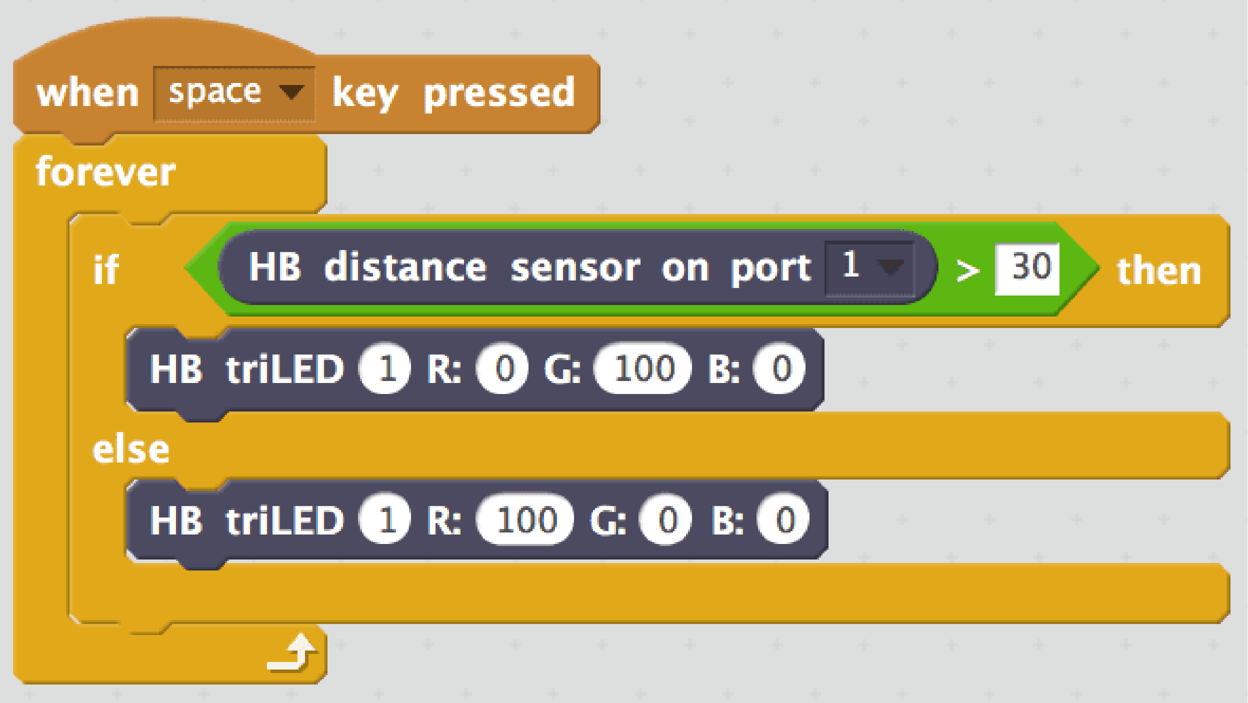
Place the Boolean block into the hexagonal space in the if then else block.

When a script reaches an if then else block, it checks the value of the Boolean block inside it. If the value is true, then the script runs the blocks inside the top part of the if then else block. If the value of the Boolean block is false, then the script runs the blocks in the bottom part of the if then else.



```
when space key pressed
if < (HB distance sensor on port 1) > 30 > then
    HB triLED 1 R: 0 G: 100 B: 0
else
    HB triLED 1 R: 100 G: 0 B: 0
```

As an example, consider this script. When you press the space bar, it will check the distance sensor. If the distance to an object is greater than 30 cm, the tri-color LED will turn green. Otherwise, it will turn red. Try it out!

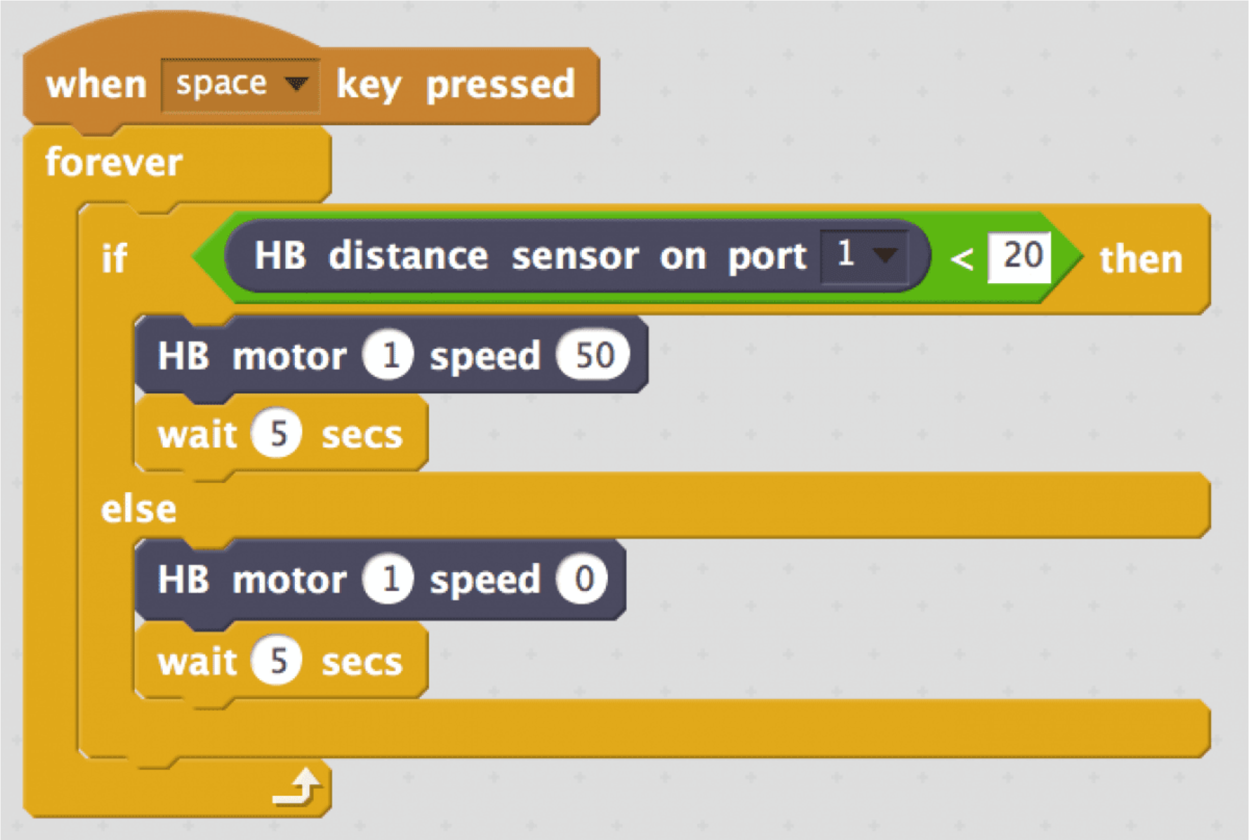


```
when space key pressed
forever
    if HB distance sensor on port 1 > 30 then
        HB triLED 1 R: 0 G: 100 B: 0
    else
        HB triLED 1 R: 100 G: 0 B: 0
```

The script above makes a decision each time you press the space bar. You may also want to repeat this decision over and over. Then the light will turn red whenever something gets close to the sensor. To do this, place the if then else block inside a forever loop.

Exercise 5.2

Write a program that blinks a red light and moves a servo motor when an object is close to the distance sensor. When nothing is near the distance sensor, the light should be off and the servo should not move.

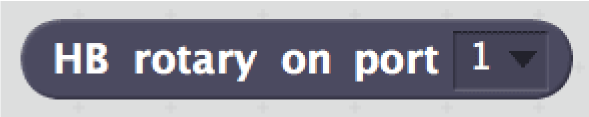


```
when space key pressed
forever
    if HB distance sensor on port 1 < 20 then
        HB motor 1 speed 50
        wait 5 secs
    else
        HB motor 1 speed 0
        wait 5 secs
```

Exercise 5.3

The Boolean block for the if then else is checked each time the program repeats the decision. However, the Boolean block is not checked during the statements inside the top or the bottom of the if then else block. Try out this script. In this script, the motor should be on when something is close to the distance sensor. When you run this program, you will notice that the program does not respond quickly when you move an object toward or away from the distance sensor. How can you modify this program to make it respond more quickly to a change in the value of the distance sensor?

Module 6: Rotary Knob



```
HB rotary on port 1
```

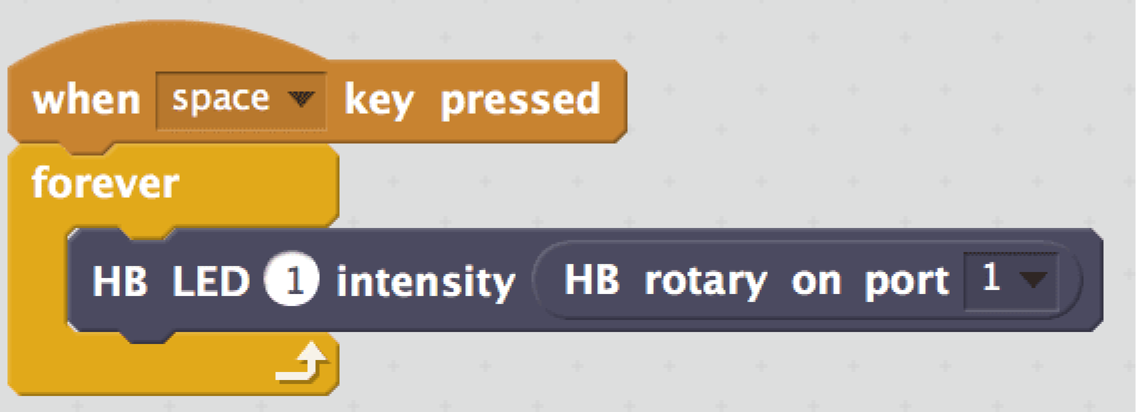
The rotary knob is another sensor; this sensor measures how much you have turned the knob. You can use the HB rotary block to find the value of this sensor. This value will be between 0 and 100. Always remember to make sure that the HB rotary block is set to the port attached to the rotary knob.

Exercise 6.1

Attach the knob to the Hummingbird board. Declare a variable named “knob” and write a script that uses the set toblock to continually set this variable to the value of the HB rotary block. Then turn the knob all the way to the left. What is the value of the HB rotary block? What is the value of the HB rotary block when the knob is turned all the way to the right?

Exercise 6.2

Write a program that turns on a single color LED when the knob is turned far enough to the left. When the knob is turned far enough to the right, the LED should be off. What will your threshold be for this program? Remember, the threshold is the value that the Boolean block uses to make a decision.

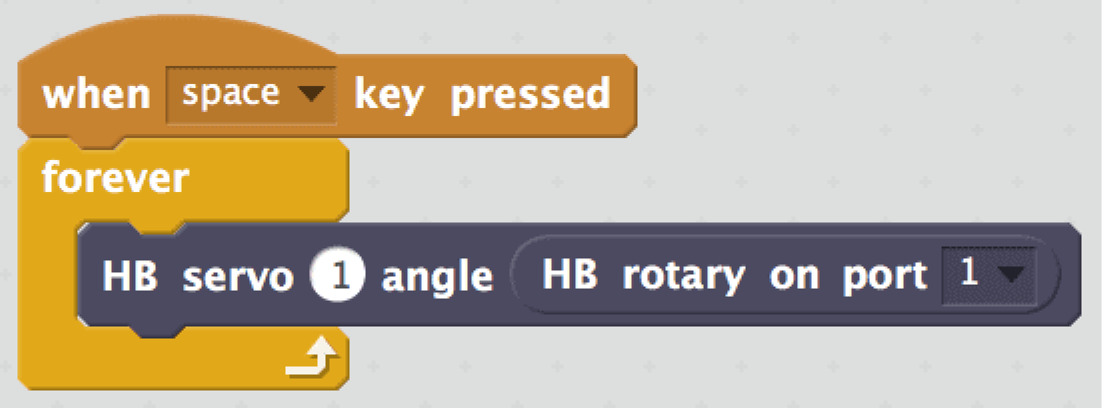


```
when space key pressed
forever
    HB LED 1 intensity HB rotary on port 1
```

In the previous exercise, you used the rotary knob to decide whether the LED should be on or off. Instead, you could use the knob to determine the brightness of the LED. Because the HB rotaryblock is a number between 0 and 100, you can place it inside the HB LED block to set the intensity of the light. Try out the script shown. The forever loop repeatedly changes the brightness of the LED to match the value of the knob.

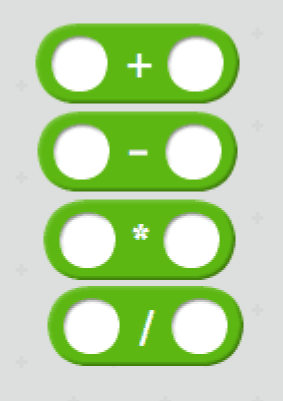
Exercise 6.3

Use the knob to control a gear motor and the amount of red in a tri-color LED (the amount of green and blue can be whatever you want). These components should change at the same time.

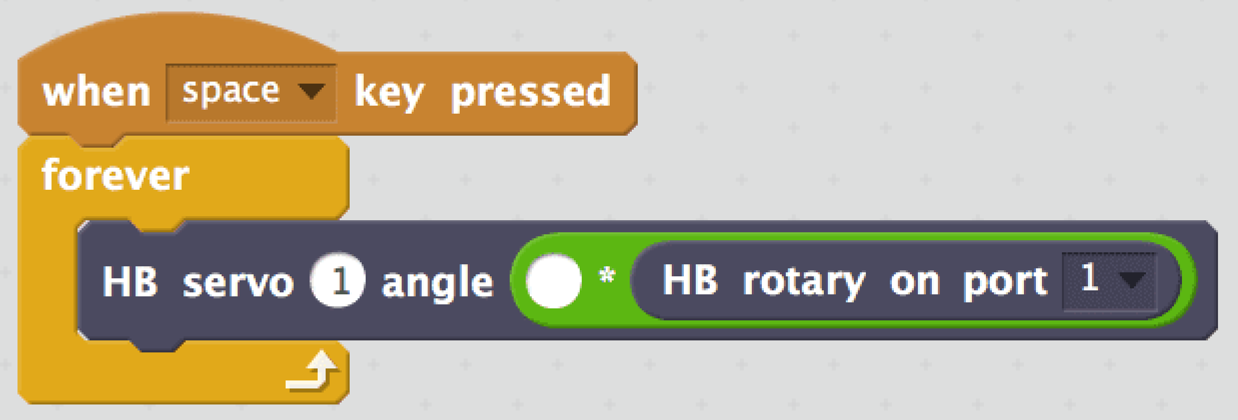


```
when space key pressed
forever
    HB servo 1 angle HB rotary on port 1
```

The rotary knob is useful for controlling the LEDs because its range (0-100) is the same as the input range for these components. If you place the knob block inside the HB servo block as shown, the servo will only move from 0° to 100°.



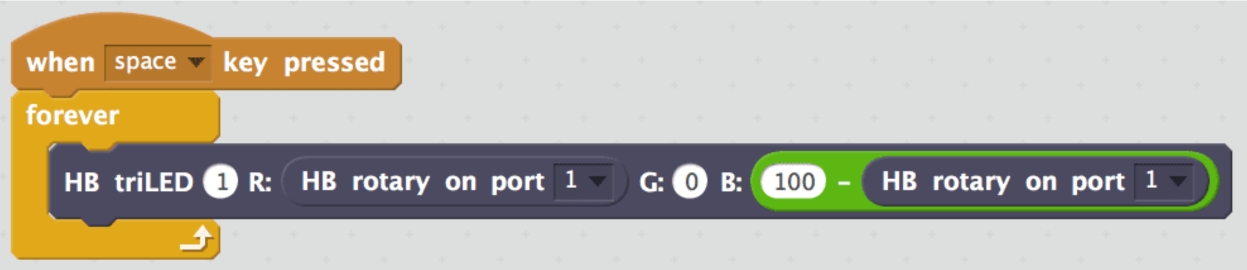
You can make the servo move from 0° to 180° using an arithmetic block from the Operators menu. These blocks enable you to add, subtract, multiply, and divide numbers.



```
when space key pressed
forever
    HB servo 1 angle ( * HB rotary on port 1 )
```

Exercise 6.4

Place the HB rotary block inside a multiplication block as shown. What number should you put in the blank so that the knob moves the servo from 0° to 180°? When the value of the knob is 0, the servo should be at 0°. When the value of the knob is 100, the servo should be at 180°.



```
when space key pressed
forever
    HB triLED 1 R: HB rotary on port 1 G: 0 B: 100 - HB rotary on port 1
```

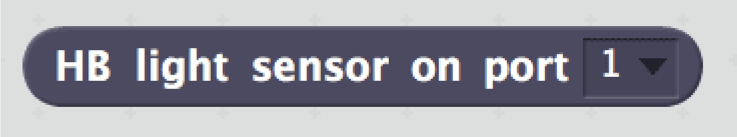
Exercise 6.5

What do you think the script shown will do? Make a prediction before you try it out.

Programming Tip

Don’t forget to set the port in the sensor block to the port on the Hummingbird board that you are actually using! If the value of the sensor is not changing as expected, be sure to check that the sensor block is reading the correct port.

Module 7: Light Sensor and Using Multiple Sensors



```
HB light sensor on port 1
```

The light sensor measures the amount of light around it. This sensor is very similar to the other sensors that you have learned about. You can use the HB light sensor block to find the value of this sensor. This value will be between 0 and 100. Always remember to make sure that the HB light sensor block is set to the appropriate port.

Exercise 7.1

Attach the light sensor to the Hummingbird board. Use the HB light sensor block to measure the amount of light in your room. Then measure the value of the light sensor when you cover it with your hand. The average of these two values is a good threshold for the light sensor.

Exercise 7.2

Write a script that turns on a vibration motor when the light sensor detects that it is dark. Otherwise, the vibration motor should be off.

You have now practiced using all the Hummingbird motors and lights, as well as three of the sensors. There are other sensors, including a temperature sensor and a sound sensor. These work very similarly to the sensors you have already used. The HB temperature block tells you the temperature in Celsius, and the HB sound block tells you the amount of noise around the Hummingbird on a scale of 0-100.

The last two exercises below will give you some practice using multiple sensors. After you have completed these, you will be ready to create many exciting, unique Hummingbird robots!

Exercise 7.3

Write a program that uses the knob sensor to control the color of a tri-color LED. However, the LED should only be on when an object is less than 30 cm from the distance sensor. Otherwise, the LED should be off.

Exercise 7.4

Write a program that sends the servo to four different angles based on the values of the distance and light sensors. All of the values are shown in the table below.

Distance	Light	Servo Angle

Greater than 20 cm	Above the threshold	0°
Greater than 20 cm	Below the threshold	60°
Less than 20 cm	Above the threshold	120°
Less than 20 cm	Below the threshold	180°

Congratulations! You have completed Hummingbird Scratch training.
Have fun making robots!

Resources

- Hummingbird Video Tutorials
- Getting Started with Scratch
- Scratch Cards
- Scratch Video Tutorials
- Scratch Wiki