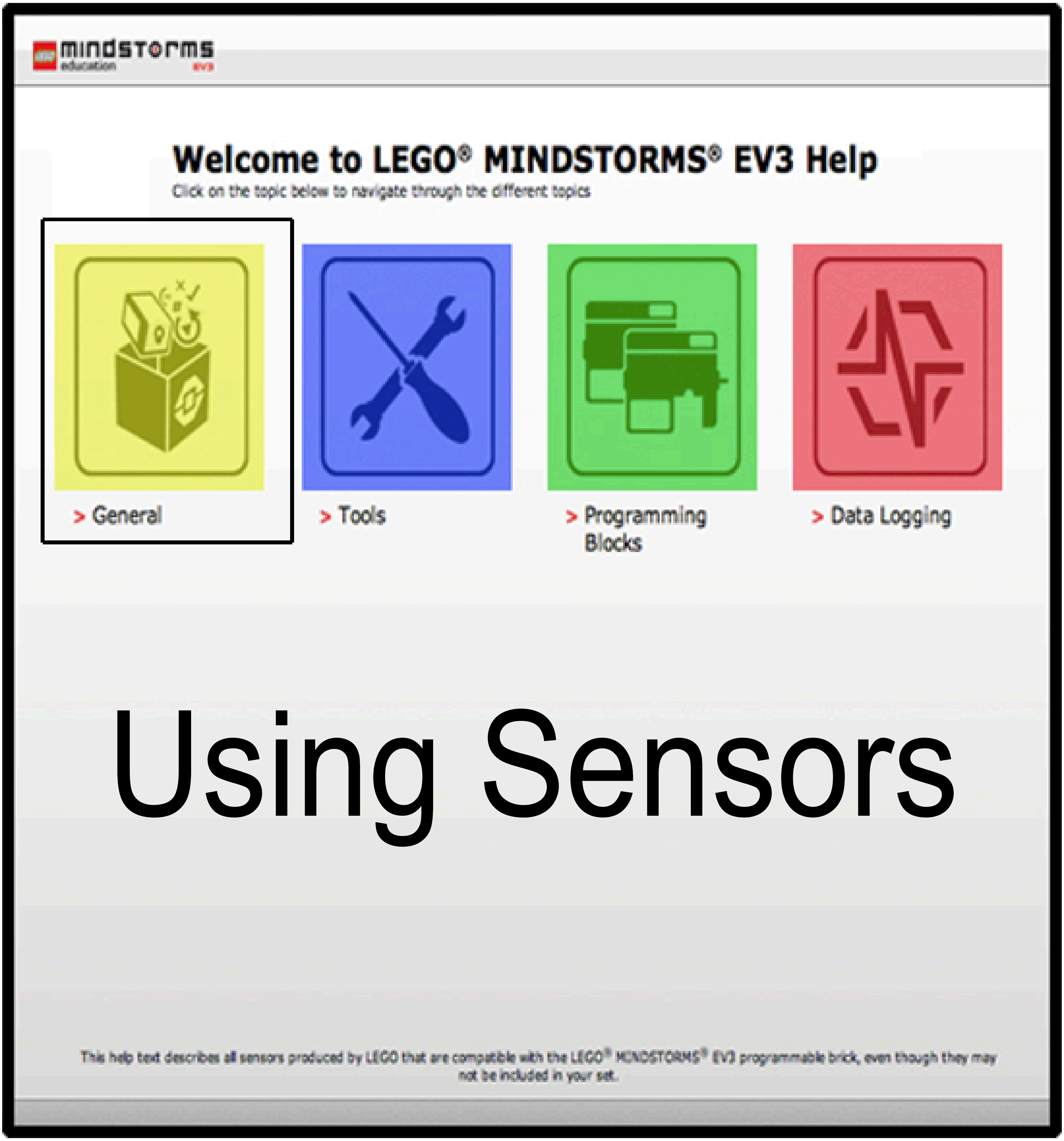
Using Sensors



Welcome to LEGO® MINDSTORMS® EV3 Help

Click on the topic below to navigate through the different topics

This help text describes all sensors produced by LEGO that are compatible with the LEGO® MINDSTORMS® EV3 programmable brick, even though they may not be included in your set.

Using the Infrared Sensor

The Infrared Sensor can detect infrared light signals that are sent from the Remote Infrared Beacon (IR Beacon). The Infrared Sensor can also send its own infrared light signal, and detect the reflection of this light by other objects.

The Infrared Sensor can be used in three different modes: Proximity, Beacon, and Remote.

PROXIMITY MODE

In Proximity mode, the Infrared Sensor sends its own infrared signal and can detect the reflection of this signal by an object in front of the sensor. The strength of the reflected signal can be used to estimate the proximity of (distance to) the object.

See Using the Infrared Sensor Proximity Mode.

BEACON MODE

In Beacon mode, the IR Beacon transmits a special beacon signal continuously, and the Infrared Sensor can detect the approximate position of the beacon in front of the sensor.

See Using the Infrared Sensor Beacon Mode.

REMOTE MODE

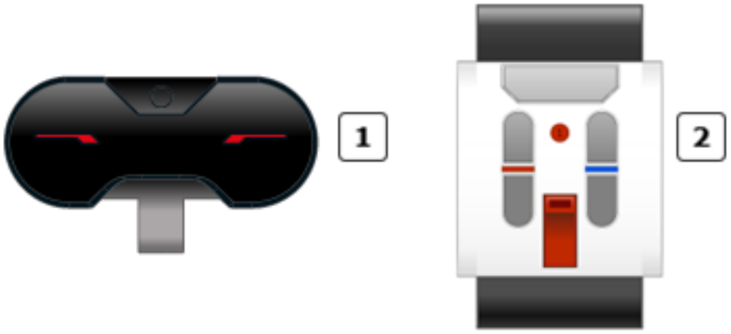
In Remote mode, the Infrared Sensor can detect button presses on the IR Beacon. You can use the Remote mode to make a remote control for your robot, for example.

See Using the Infrared Sensor Remote Mode.

Tips and Tricks

Infrared light is the same kind of signal used by most TV remote controls. You cannot see infrared light, but, like visible light, it is blocked if objects are in the way. The IR Beacon must have a “line of sight” to the Infrared Sensor to be seen. Sunlight can also interfere with the infrared signals, although normal room light should not affect it.

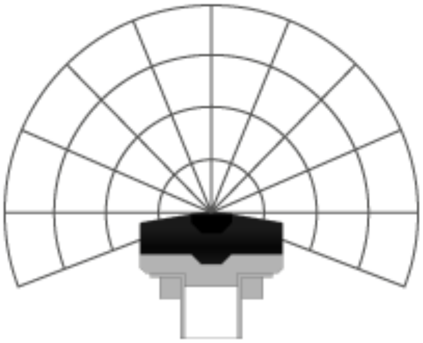
Using the Infrared Sensor Beacon Mode



1 Infrared Sensor

2 Remote Infrared Beacon

In Beacon mode, the Infrared Sensor can detect the approximate position of the Remote Infrared Beacon (IR Beacon) in front of the sensor. The sensor can give you the beacon’s Proximity (relative distance from the sensor) and its Heading (angle from the direction the sensor is pointing). You could use the Beacon mode, for example, to make your robot seek out and drive toward the IR Beacon.



TURN ON THE BEACON AND CHOOSE A CHANNEL

Turn the beacon on by pressing the Beacon Mode button on the top of the IR Beacon. The LED Indicator will turn on and stay on. The beacon will stay on and transmit continuously until you press the Beacon Mode button again to turn it off.

Choose one of the four channels from the Channel Selector. The Infrared Sensor will only detect a beacon on the channel that you specify in your program.

INFRARED SENSOR BEACON MODE DATA

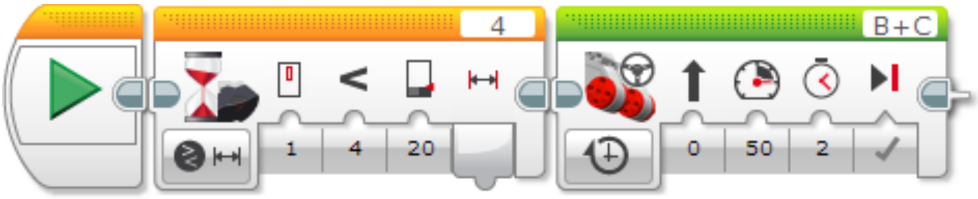
In Beacon mode, the Infrared Sensor gives the following data:

Data	Type	Values	Notes
Detected	Logic	True/False	True if an IR Beacon is detected on the specified channel, otherwise False.
Proximity	Numeric	0 to 100	The relative distance to the beacon. 0 means very close, and 100 means far away. The Proximity will be 100 if the beacon is not detected at all.
Heading	Numeric	-25 to 25	0 means the beacon is directly in front of the sensor, negative values are to the left, and positive values are to the right.

Tips and Tricks

The values for Proximity and Heading do not correspond directly to specific distances and angles. The values will depend on the strength of the signal and other factors.

Example



This program will make a robot start driving when the IR Beacon is on and gets close enough to the Infrared Sensor. It uses the Wait block in the Infrared Sensor – Compare – Beacon Proximity mode to wait for the Proximity to be less than 20, then it drives forward for 2 seconds.

INFRARED SENSOR BEACON MODE BLOCKS AND MODES

The table below shows all of the programming blocks and modes that you can use with the Infrared Sensor in Beacon mode.

Block	Mode	Use
Wait	Infrared Sensor – Compare – Beacon Heading	Wait for the beacon to be detected and for the Heading to reach a specified value.
Wait	Infrared Sensor – Compare – Beacon Proximity	Wait for the beacon to be detected and for the Proximity to reach a specified value.
Wait	Infrared Sensor – Change - Beacon Heading	Wait for the beacon Heading to change by a specified amount.
Wait	Infrared Sensor – Change - Beacon Proximity	Wait for the beacon Proximity to change by a specified amount.
Loop	Infrared Sensor –	Repeat a sequence of blocks until the beacon

Loop	Infrared Sensor – Beacon Heading	Repeat a sequence of blocks until the beacon Heading reaches a specified value.
Loop	Infrared Sensor – Beacon Proximity	Repeat a sequence of blocks until the beacon Proximity reaches a specified value.
Switch	Infrared Sensor – Beacon Heading	Choose between two sequences of blocks depending on the beacon Heading.
Switch	Infrared Sensor – Beacon Proximity	Choose between two sequences of blocks depending on the beacon Proximity.
Infrared Sensor	Measure - Beacon	Get the beacon Heading and Proximity on Numeric data wires, and the Detected state on a Logic data wire.
Infrared Sensor	Compare – Beacon Heading	Compare the beacon Heading to a threshold, and get the result on a Logic data wire.
Infrared Sensor	Compare – Beacon Proximity	Compare the beacon Proximity to a threshold, and get the result on a Logic data wire.

Using the Infrared Sensor Proximity Mode

In Proximity mode, the Infrared Sensor sends out an infrared signal, and it can detect the reflection of this signal by an object in front of the sensor. The strength of the reflected signal can be used to estimate the proximity of (distance to) the object. You could use the Proximity mode, for example, to detect when your robot gets close to a wall.

INFRARED SENSOR PROXIMITY MODE DATA

In Proximity mode, the Infrared Sensor gives the following data:

Data	Numeric	Values	Notes
Proximity	Numeric	0 to 100	The relative distance to an object in front of the sensor. 0 means very close, and 100 means far away.

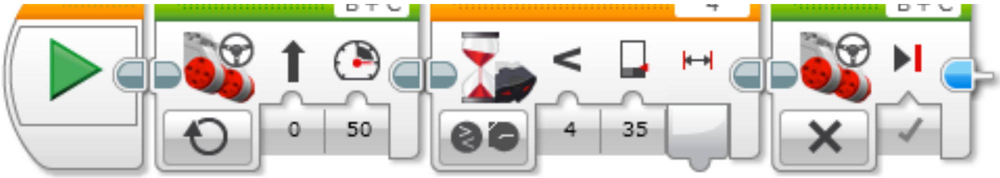
Tips and Tricks

- The Proximity value does not correspond directly to a specific distance. The value will depend on the color and material of the object in front of the sensor, and other factors.
- The Infrared Sensor cannot detect Proximity to an object that is very close to the sensor (closer than about 1 cm or half an inch).
- The Beacon mode of the Infrared Sensor also provides Proximity data, but only for detecting proximity to the IR Beacon. See Using the Infrared Sensor Beacon Mode for more information.

EXAMPLES USING THE INFRARED SENSOR IN PROXIMITY MODE

Some examples of how you can use the Infrared Sensor in Proximity mode are below.

Example 1: Stop Driving Before Reaching a Wall



This program will make a robot drive forward until the Infrared Sensor detects that it is close to a wall or other object. After the driving starts, the program uses the Wait block in the Infrared Sensor – Compare – Proximity mode to wait for the Proximity to be less than 35 before stopping the robot.

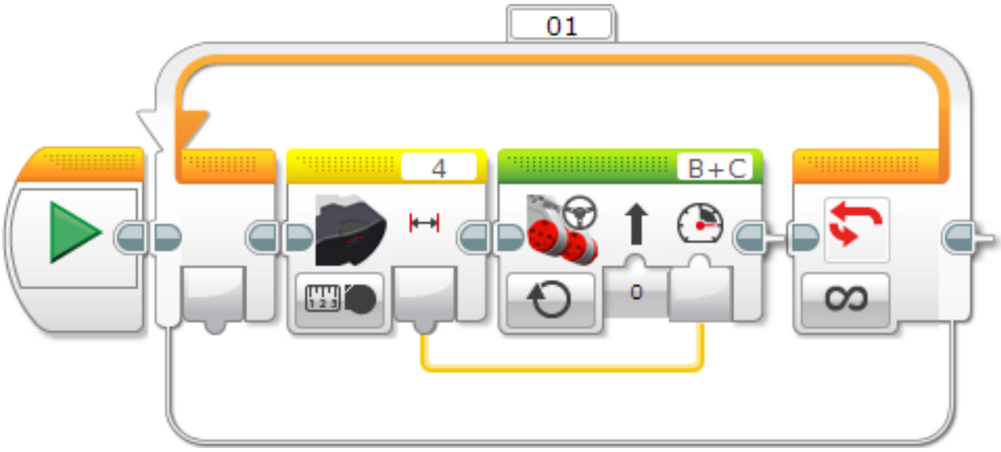
Tips and Tricks

The distance that the robot stops before reaching an object will depend a lot on the color of the object. This is because light colored objects reflect (infrared) light better than dark objects.

Tips and Tricks

Remember to use the On mode of the Move Steering block when you want to drive while waiting for a sensor.

Example 2: Slow Down when Approaching a Wall



This program makes a robot gradually slow down as it approaches a wall or other object. It uses the Infrared Sensor block in the Measure – Proximity mode to get the Proximity on a Data Wire. This value is used for the Power input of the Move Steering block, and the process is repeated in a Loop so that the speed is continuously adjusted based on the Proximity.

INFRARED SENSOR PROXIMITY MODE BLOCKS AND MODES

The table below shows all of the programming blocks and modes that you can use with the Infrared Sensor in Proximity mode.

Block	Mode	Use
Wait	Infrared Sensor – Compare – Proximity	Wait for the Proximity to reach a specified value.
Wait	Infrared Sensor – Change - Proximity	Wait for the Proximity to change by a specified amount.

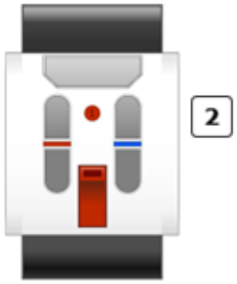
Loop	Infrared Sensor – Proximity	Repeat a sequence of blocks until the Proximity reaches a specified value.
Switch	Infrared Sensor – Beacon Proximity	Choose between two sequences of blocks depending on the Proximity.
Infrared Sensor	Measure - Proximity	Get the Proximity value on a Numeric data wire.
Infrared Sensor	Compare – Proximity	Compare the Proximity to a threshold, and get the result on a Logic data wire.

IR Remote Mode

Quick links

- Infrared Sensor Remote Mode Data
- Examples Using the Infrared Sensor in Remote Mode
- Infrared Sensor Remote Mode Blocks and Modes

U d Sensor Remote Mode



1 Infrared Sensor

2 Remote Infrared Beacon

In Remote mode, the Infrared Sensor can detect which button on the Remote Infrared Beacon (IR Beacon) is pressed. You can also detect when certain combinations of two buttons are pressed at the same time. You can use the Remote mode, for example, to make a remote control for your robot.

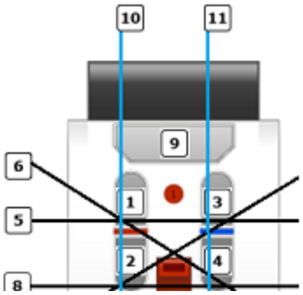
The IR Beacon has a Channel Selector that lets you choose one of four different channels for the signals. The Infrared Sensor will only detect signals from the channel you specify.

Tips and Tricks

If two robots are being controlled by two different IR Beacons, they should use different channels. Otherwise, one beacon will control all of the robots on its channel.

INFRARED SENSOR REMOTE MODE DATA

In Remote mode, the Infrared Sensor gives the following data:

Data	Type	Range	Notes
Button ID	Numeric	0 - 11	Identifies which button, or combination of buttons, is pressed on the IR Beacon. 0 = No button (and Beacon Mode is off) 1 = Button 1 2 = Button 2 3 = Button 3 4 = Button 4 5 = Both Button 1 and Button 3 6 = Both Button 1 and Button 4 7 = Both Button 2 and Button 3 8 = Both Button 2 and Button 4 9 = Beacon Mode is on 10 = Both Button 1 and Button 2 11 = Both Button 3 and Button 4 

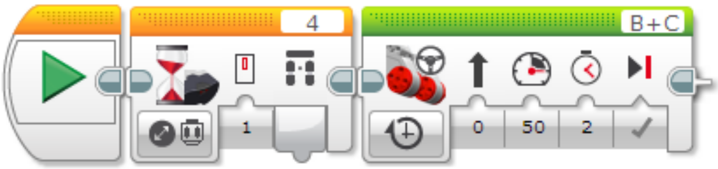
Tips and Tricks

The Beacon Mode button (Button ID = 9) acts different than the other four buttons. When you press the Beacon Mode button, the beacon starts transmitting continuously until you press the Beacon Mode button again to turn it off. The other four buttons only transmit when they are held down and stop transmitting when you release the button.

EXAMPLES USING THE INFRARED SENSOR IN REMOTE MODE

Some examples of how you can use the Infrared Sensor in Remote mode are below.

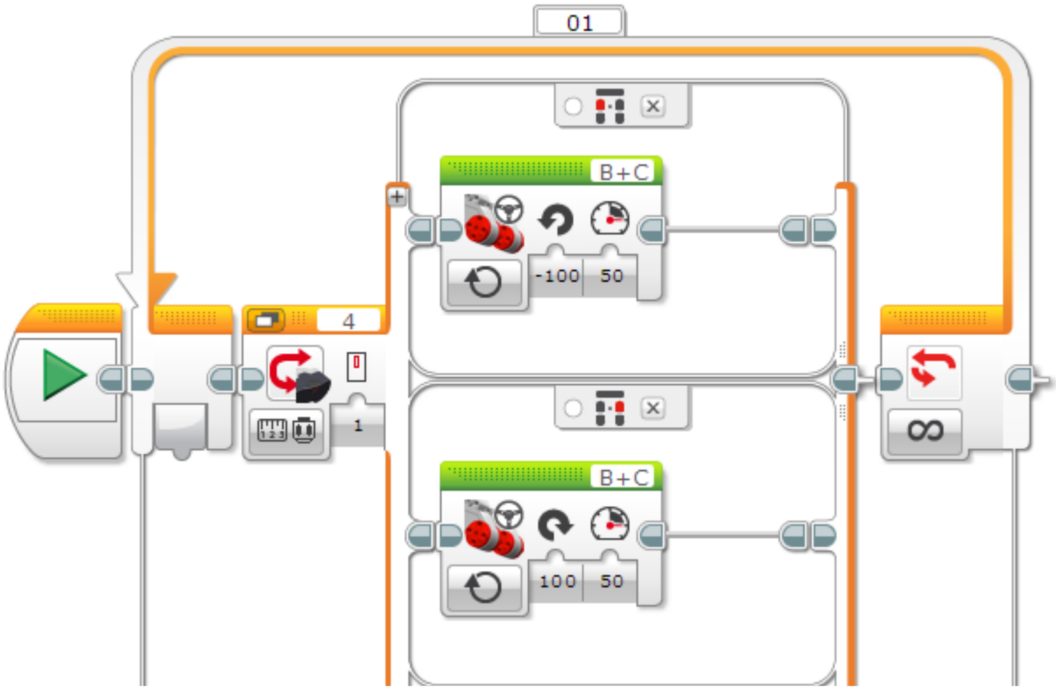
Example 1: Remote Start Button

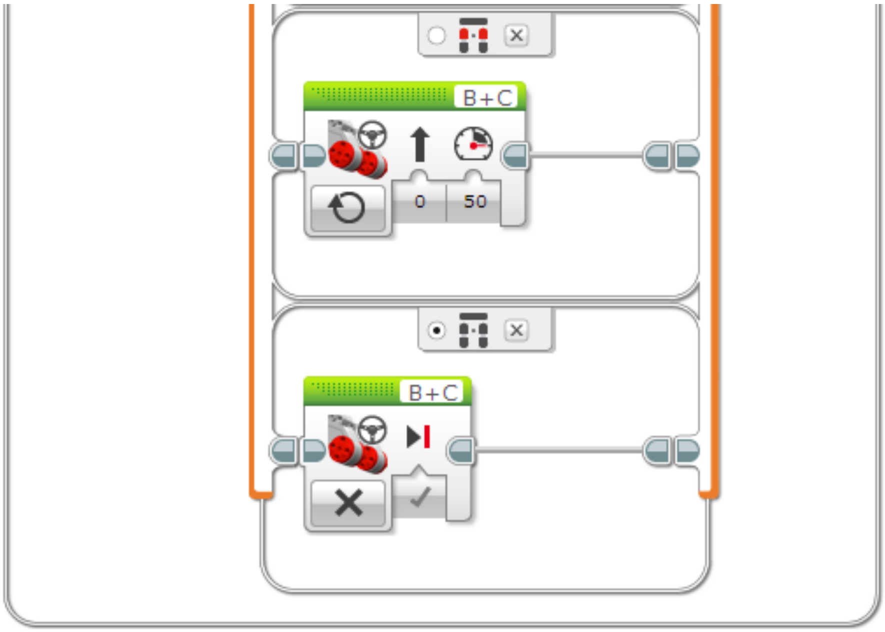


This program makes a robot wait until a button on the IR Beacon is pressed (using channel 1), then it drives forward for 2 seconds. It uses the Wait block in the Infrared Sensor – Change – Remote mode. If you start with no button pressed, this will wait until any button on the IR Beacon is pressed.

Example 2: Remote Control Driving

The program below can drive a robot by remote control from the IR Beacon. It uses a Switch in Infrared Sensor – Measure – Remote Buttons mode to choose from four different driving motions depending on which button(s) on the IR Beacon are pressed. You can turn left and right by pressing the top left and top right direction buttons, and go straight by pressing both of these buttons at the same time. The robot is stopped when all buttons are released.



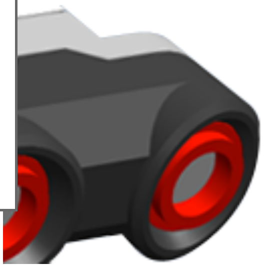


INFRARED SENSOR REMOTE MODE BLOCKS AND MODES

The table below shows all of the programming blocks and modes that you can use with the Infrared Sensor in Remote mode.

Block	Mode	Use
Wait	Infrared Sensor – Compare - Remote	Wait for a specified button on the IR Beacon to be pressed. You can also wait for one of a set of specified buttons to be pressed.
Wait	Infrared Sensor – Change - Remote	Wait for any button on the IR Beacon to be pressed, or for the button state to change.
Loop	Infrared Sensor – Remote	Repeat a sequence of blocks until a specified button on the IR Beacon is pressed (or until one of a set of specified buttons is pressed).
Switch	Infrared Sensor – Measure – Remote	Choose from two or more sequences of blocks depending on which button(s) are pressed on the IR Beacon.
Switch	Infrared Sensor – Compare - Remote	Choose between two sequences of blocks depending on whether a specified button on the IR Beacon is pressed (or whether one of a set of specified buttons is pressed).
Infrared Sensor	Measure - Remote	Get the Button ID of the currently pressed button on the IR Beacon on a Numeric data wire.
Infrared Sensor	Compare - Remote	Test if a specified button on the IR Beacon is pressed (or one of a set of specified buttons), and get the result on a Logic data wire.

Using the Ultrasonic Sensor



The Ultrasonic Sensor can measure the distance to an object in front of it. It does this by sending out sound waves and measuring how long it takes the sound to reflect back to the sensor. The sound frequency is too high for you to hear ("ultrasonic").

You can measure the distance to an object in either inches or centimeters. You could use this to, for example, make your robot stop a certain distance from a wall.

You can also use the Ultrasonic Sensor to detect whether another ultrasonic sensor nearby is operating. For example, you could use this to detect the presence of another robot that is using an ultrasonic sensor nearby. In this "listen only" mode, the sensor listens for sound signals but does not send them.

ULTRASONIC SENSOR DATA

The Ultrasonic Sensor can give the following data:

Data	Type	Range	Notes
Distance in Centimeters	Numeric	0 to 255	Distance to object in centimeters.
Distance in Inches	Numeric	0 to 100	Distance to object in inches.
Ultrasound Detected	Logic	True/False	True if another ultrasonic sensor is detected.

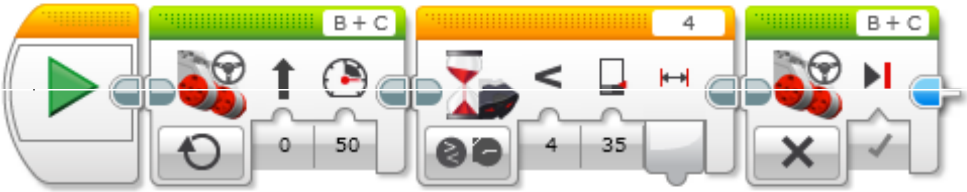
Tips and Tricks

- The Ultrasonic Sensor works best to detect objects with hard surfaces that reflect sound well. Soft objects, such as cloth, may absorb the sound waves and not be detected. Objects with rounded or angled surfaces are also harder to detect.
- The sensor cannot detect objects that are very close to the sensor (closer than about 3 cm or 1.5 inches).
- The sensor has a wide "field of view" and may detect a closer object off to the side instead of a farther object straight ahead.

EXAMPLES USING THE ULTRASONIC SENSOR

Some examples of how you can use the Ultrasonic Sensor in your program are shown below.

Example 1: Stop a Certain Distance before a Wall

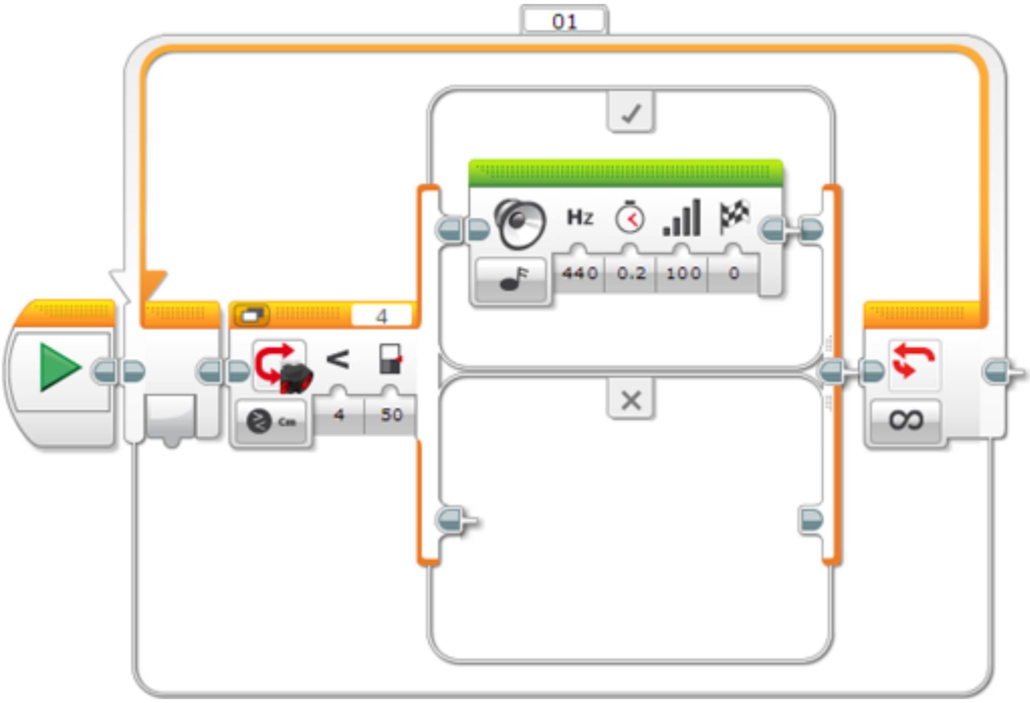


This program makes a robot drive forward until the Ultrasonic Sensor detects something closer than 10 inches, then the robot is stopped. The program uses the Wait block in the Ultrasonic Sensor - Compare – Distance Inches mode to wait for the detected distance to become less than 10 inches. If the Ultrasonic Sensor is facing forward, the robot will stop about 10 inches before a wall.

Tips and Tricks

Remember to use the On mode of the Move Steering block when you want to drive while waiting for a sensor.

Example 2: Sound an Alarm when an Object is Detected Nearby

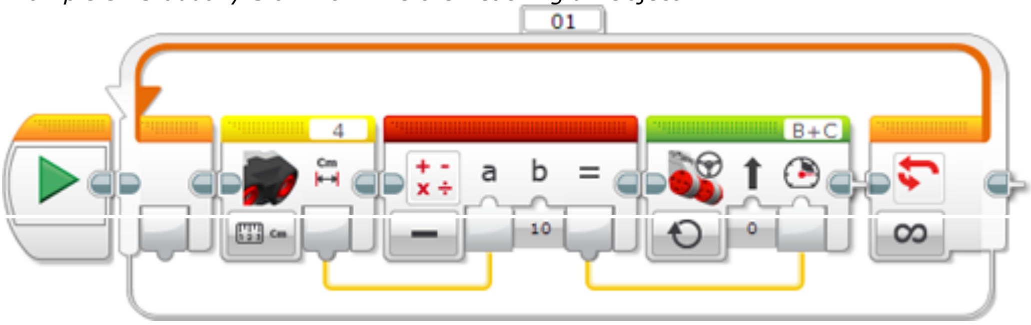


This program has the robot make a sound whenever the Ultrasonic Sensor detects an object closer than 50 centimeters away. The program uses a Switch with the Ultrasonic Sensor - Compare – Distance Centimeters mode to test whether the distance detected is less than 50 centimeters. If so, the Switch plays a tone. The Switch is repeated in a loop so that the test is repeated continuously.

Tips and Tricks

While running this program, try moving objects around in front of the sensor to experiment with how wide the sensor's "field of view" is.

Example 3: Gradually Slow Down Before Reaching an Object



This program makes a robot gradually slow down and then stop about 10 cm away from anything it detects in front of it. The closer it gets to the object, the slower it will drive.

The program uses the Ultrasonic Sensor block in the Measure – Distance Centimeters mode to get a distance measurement and get the resulting number on a data wire. A Math block then subtracts 10 from the distance, and the result is wired to the Power input of a Move Steering block. Shorter distances result in lower power, and when the distance reaches 10 cm, the power will be zero, and the robot will stop. The process is repeated in a loop so that the motor power is adjusted continuously based on new distance measurements.

Tips and Tricks

You can also try moving the object while this program is running. The robot will continuously adjust its speed.

ULTRASONIC SENSOR BLOCKS AND MODES

The table below shows all of the programming blocks and modes that you can use with the Ultrasonic Sensor. The Distance modes have sub-modes that let you choose between centimeters and inches.

Block	**Mode**	**Use**
Wait	Ultrasonic Sensor - Compare – Distance	Wait for the distance to reach a certain value.
Wait	Ultrasonic Sensor - Compare – Presence	Wait, in "listen only" mode, for an ultrasonic signal to be detected.
Wait	Ultrasonic Sensor - Change – Distance	Wait for the distance to change by a certain amount.
Loop	Ultrasonic Sensor - Compare - Distance	Repeat a sequence of blocks until the distance reaches a certain value.
Loop	Ultrasonic Sensor – Compare - Presence	Repeat a sequence of blocks until an ultrasonic signal is detected, in "listen only" mode.
Loop	Ultrasonic Sensor – Change - Distance	Repeat a sequence of blocks until the distance changes by a certain amount.
Switch	Ultrasonic	Choose between two sequences of blocks based

	Sensor – Compare - Distance	on the distance.
Switch	Ultrasonic Sensor – Compare - Presence	Choose between two sequences of blocks based on whether an ultrasonic signal is detected in “listen only” mode.
Ultrasonic Sensor	Measure – Distance	Measure the distance and get the result on a Numeric data wire.
Ultrasonic Sensor	Measure – Presence	Listen for other ultrasonic signals in “listen only” mode, and get the result on a Logic data wire.
Ultrasonic Sensor	Compare – Distance	Compare the distance to a threshold, and get the result on a Logic data wire.
Ultrasonic Sensor	Compare – Presence	Listen for other ultrasonic signals in “listen only” mode, and get the result on a Logic data wire.
Ultrasonic Sensor	Advanced	Similar to Measure – Distance, but with the option to make only a single sound ping.
Data Logging		See Data Logging.

Jsing the Color Sensor

he Color Sensor can detect the color or intensity of light that enters the small window n the face of the sensor. The Color Sensor can be used in three different modes: olor Mode, Reflected Intensity Mode, and Ambient Intensity Mode.

COLOR MODE

In Color mode, the Color Sensor can detect the color of a nearby object, or the color of a surface near the sensor. You can use the Color mode to detect, for example, the color of a LEGO part held close to the sensor, or the color of different markings on a piece of paper.

Tips and Tricks

When the Color Sensor is in Color mode, red, green, and blue LED lights on the front of the sensor will turn on.

The sensor can detect seven different colors: black, blue, green, yellow, red, white, and brown. An object that is not one of these colors may be detected as “No Color”, or it may be detected as a similar color. For example, an orange object might be detected as red or yellow, depending on how much red the orange has in it, or as brown or black if the orange is very dark or too far away from the sensor.

Tips and Tricks

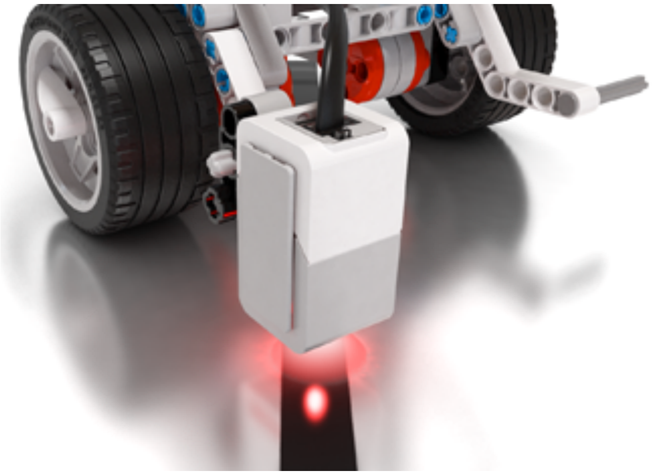
The object or surface should be very close to the sensor (but not touching it) to be detected accurately.

REFLECTED LIGHT INTENSITY MODE

In Reflected Light Intensity mode, the Color Sensor detects the intensity of light that enters the sensor. The intensity of the light is measured as a percentage from 0 to 100, with 0 being very dark, and 100 being very bright.

When the Color Sensor is in Reflected Light Intensity mode, a red LED light on the front of the sensor will turn on. If the sensor is close to an object or surface, this red light will reflect off of the object and then enter the sensor to be detected. You can use this to measure shades of color on a surface or object, because darker shades of color will reflect less of the red light back to the sensor.

You can use this mode to, for example, make your robot follow a black line on a white surface. As the sensor passes over the black line, the light measurement will gradually decrease as the sensor gets closer to the black line. This can be used to tell how close the robot is to the line.



Tips and Tricks

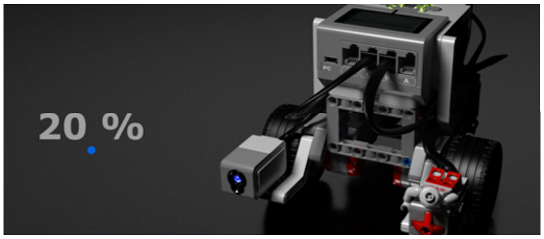
The Reflected Light Intensity mode measures the total amount of light entering the sensor. This includes the reflection of the red LED, plus any lights in the room. The sensor should be positioned close to (but not touching) the surface being measured, to reduce the effect of outside light sources.

AMBIENT LIGHT INTENSITY MODE

In Ambient Light Intensity mode, like the Reflected Light Intensity mode, the Color Sensor detects the intensity of light that enters the sensor. The intensity of the light is measured as a percentage from 0 to 100, with 0 being very dark, and 100 being very bright.

In Ambient Light Intensity mode, a blue LED light on the front of the sensor will turn on dimly. This blue light helps you identify that the sensor is in Ambient Light Intensity mode, but it does not affect the light measurement unless an object is very close to the sensor.

You can use this mode to detect the brightness of the room lights, or when other light sources shine on the sensor. You could use this also to detect when the lights to a room are turned on, or when a flashlight is shined on your robot.

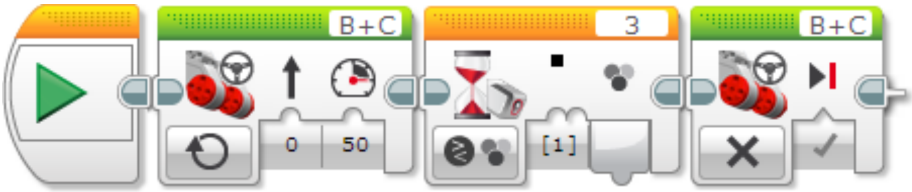


COLOR SENSOR DATA

The Color Sensor can give the following data:

Data	Type	Range	Notes
Color	Numeric	0-7	Used in Color mode. 0 = No Color 1 = Black 2 = Blue 3 = Green 4 = Yellow 5 = Red 6 = White 7 = Brown
Light	Numeric	0-100	Used in Reflected Light Intensity and Ambient Light Intensity modes. Measures light intensity as a percentage, 0 = darkest, 100 = brightest.

Example 1: Drive until a Black Line is Reached (Method 1)

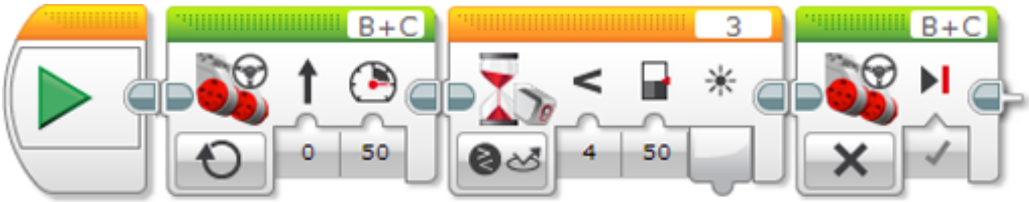


This program makes a robot drive until the Color Sensor detects a black color, then it stops. The program uses the Wait block in the Color Sensor - Compare – Color mode to test for the black color.

Tips and Tricks

If you use this program with the Color Sensor on your robot pointing downwards and close to a light-colored surface with a thick black line on it, the robot can drive until it reaches the line.

Example 2: Drive until a Black Line is Reached (Method 2)

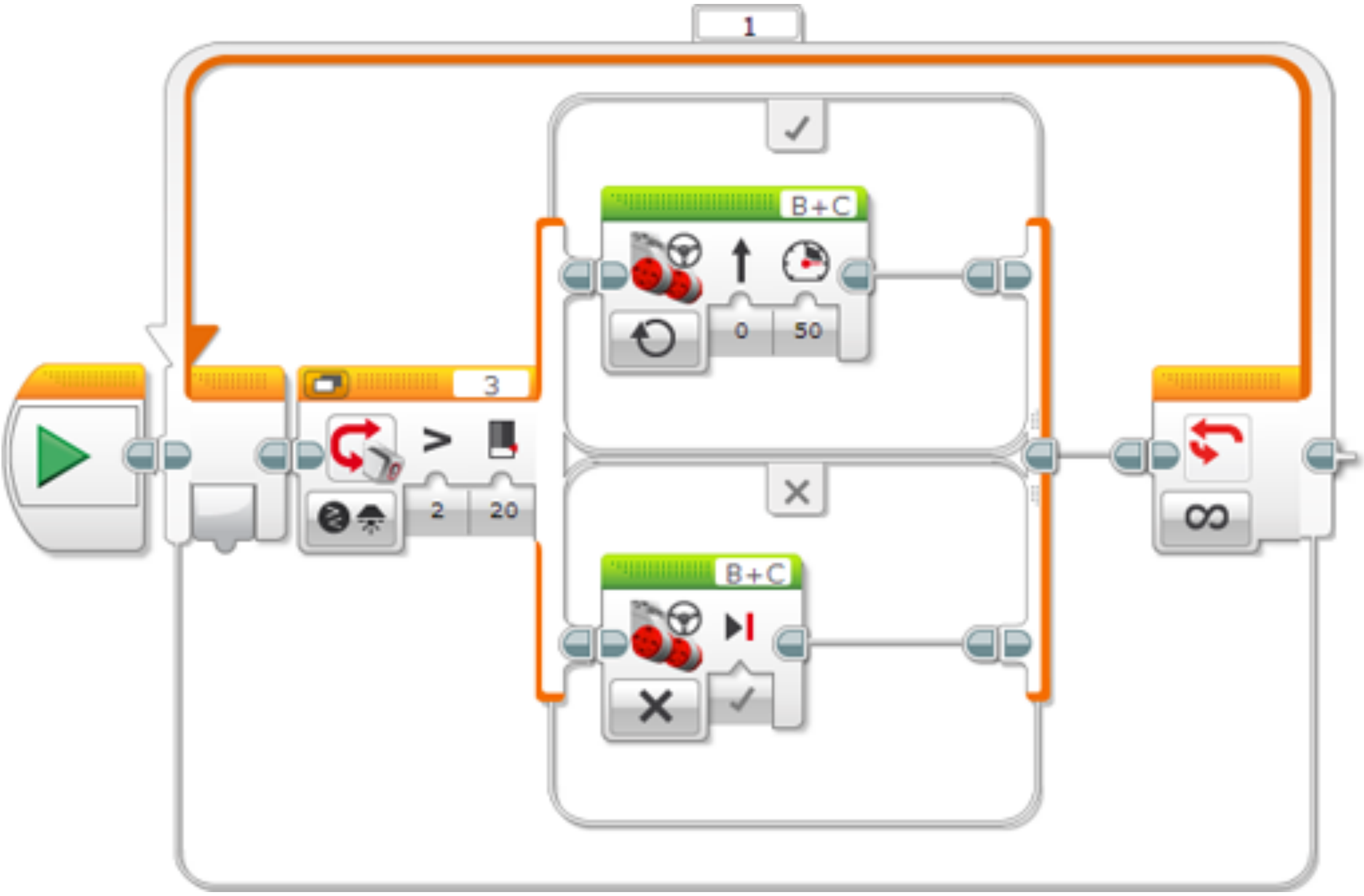


This program makes a robot drive until the Color Sensor detects a dark color, then it stops. The program uses the Wait block in the Color Sensor - Compare – Reflected Light Intensity mode to wait until the light intensify becomes less than 50%.

Tips and Tricks

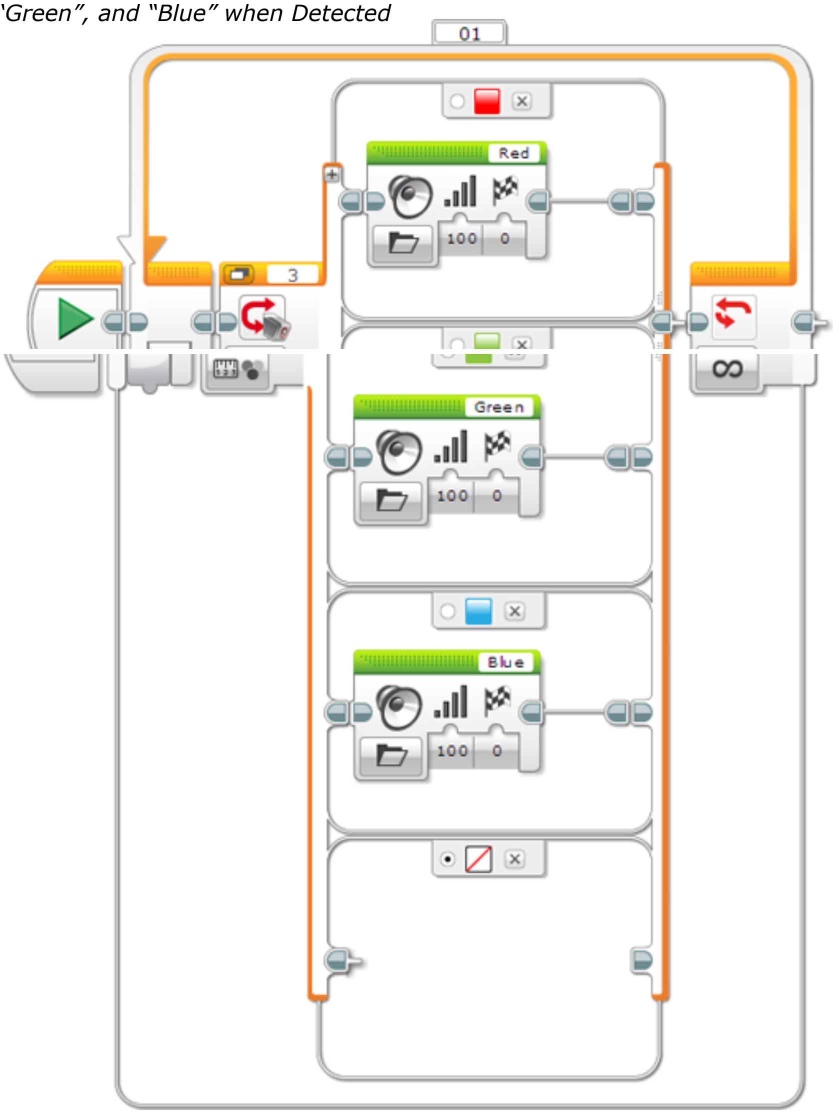
Compared to the method in Example 1 above, this program allows you to adjust how dark the line needs to be, by changing the Threshold Value (here 50%). Also, the robot will stop on any dark color, not just black.

Example 3: Drive Only When the Room Lights Are On

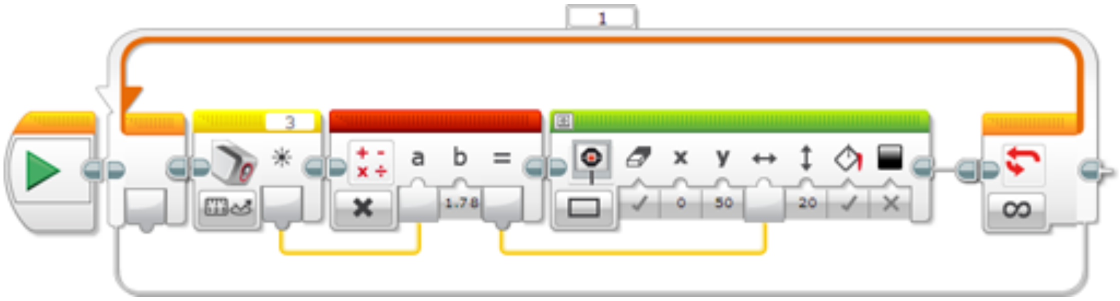


This program makes a robot drive when the room lights are on and stop when you turn off the lights. The program uses a Switch with the Color Sensor - Compare – Ambient Light Intensity mode to test whether the light is greater than 20%. The Switch chooses whether to turn the motors on or off. The Switch is repeated in a loop so that the robot will keep reacting to changes in light.

Example 4: Say "Red", "Green", and "Blue" when Detected



This program makes the EV3 say "Red", "Green", and "Blue" when the Color Sensor detects these colors. The program uses a Switch in the Color Sensor – Measure – Color mode to choose between different Sound blocks based on the color that is detected. A "No Color" case is added and selected as the default so that the EV3 won't say anything when one of the three colors is not seen.



Example 5: Display a Reflected Light Meter

This program puts a graphical light meter on the EV3 Display. The program uses a Color Sensor block in the Measure – Reflected Light Intensity mode to measure the reflected light (0-100) and get the result on a data wire. The result is then multiplied by 1.78 to scale it to the EV3 screen width (178 pixels) and then used as the width of a filled rectangle shape. The process is repeated in a loop so that the display is continuously updated.

Tips and Tricks

While running this program, try moving the Color Sensor over surfaces of different colors and shades. You will be able to see which colors reflect more light.

COLOR SENSOR BLOCKS AND MODES

The table below shows all of the programming blocks and modes that you can use with the Color Sensor.

Block	Mode	Use
Wait	Color Sensor - Compare - Color	Wait for the sensor to detect one of the selected colors.
Wait	Color Sensor - Compare - Reflected Light Intensity	Wait for the reflected light intensity to reach a certain value.
Wait	Color Sensor - Compare - Ambient Light Intensity	Wait for the ambient light intensity to reach a certain value.
Wait	Color Sensor - Change - Color	Wait for the detected color to change.
Wait	Color Sensor - Change - Reflected Light Intensity	Wait for the reflected light intensity to change by a certain amount.
Wait	Color Sensor - Change - Ambient Light Intensity	Wait for the ambient light intensity to change by a certain amount.
Loop	Color Sensor - Color	Repeat a sequence of blocks until one of the selected colors is detected.
Loop	Color Sensor - Reflected Light Intensity	Repeat a sequence of blocks until the reflected light intensity reaches a certain value.
Loop	Color Sensor - Ambient Light Intensity	Repeat a sequence of blocks until the ambient light intensity reaches a certain value.
Switch	Color Sensor - Measure - Color	Choose between two or more different sequences of blocks depending on which color is detected.
Switch	Color Sensor - Compare - Color	Choose between two sequences of blocks depending on whether or not one of the selected colors is detected.
Switch	Color Sensor - Compare - Reflected Light Intensity	Choose between two sequences of blocks depending on the reflected light intensity.
Switch	Color Sensor - Compare -	Choose between two sequences of blocks depending on the ambient light intensity.

	Compare - Ambient Light Intensity	on the ambient light intensity.
Color Sensor	Measure - Color	Measure the detected color (0-7) and get the result on a Numeric data wire.
Color Sensor	Measure - Reflected Light Intensity	Measure the reflected light intensity (0-100) and get the result on a Numeric data wire.
Color Sensor	Measure - Ambient Light Intensity	Measure the ambient light intensity (0-100) and get the result on a Numeric data wire.
Color Sensor	Compare - Color	Compare the detected color to one or more selected colors, and get the result on a Logic data wire (True if it matches any of the selected colors).
Color Sensor	Compare - Reflected Light Intensity	Compare the reflected light intensity to a threshold, and get the result on a Logic data wire.
Color Sensor	Compare - Ambient Light Intensity	Compare the ambient light intensity to a threshold, and get the result on a Logic data wire.
Data Logging		See Data Logging.

Timer

Quick links

- Timer Data
- Resetting a Timer
- Examples Using the Timer
- Timer Blocks and Modes

Jsing the Timer

he Timer can be used to measure time intervals. The Timer is used like a sensor, but is internal to the EV3 Brick and does not require a sensor port. You could use the imer to measure, for example, how long it takes your robot to move a certain istance.

The EV3 has eight timers, so you can time up to eight different things together. You can reset a timer to zero at any point in your program, and it will start timing from that point.

Tips and Tricks

If you simply want to wait for a certain amount of time in your program, you can use the Wait block in the Wait Time mode. Using the Timer lets you reset the timer and test the timer at different places in your program.

TIMER DATA

The Timer gives the following data:

Data	Type	Notes
Elapsed Time	Numeric	Elapsed time since the timer was last reset, in seconds.

Tips and Tricks

Tip: Time is measured in seconds using a decimal number. An interval of one tenth of a second would result in an Elapsed Time of 0.1 seconds.

RESETTING A TIMER

You can reset a timer to zero (0.0 seconds) at any point in your program by using the Timer block in Reset mode. After a timer is reset, it starts timing again immediately from zero. All eight timers are automatically reset at the beginning of a program and are always running.

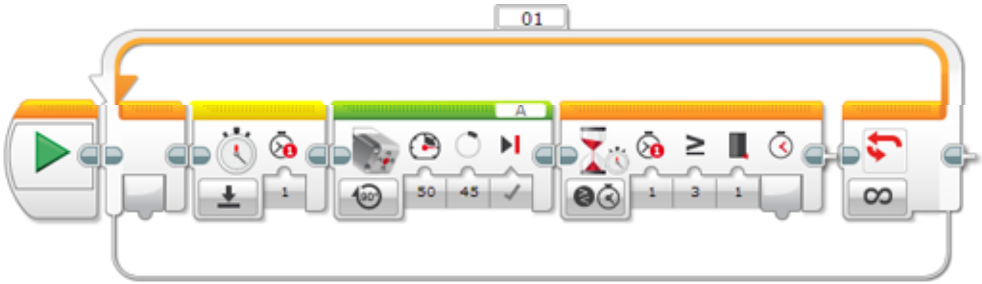
Tips and Tricks

If you measure a timer that has never been reset, you will get the elapsed time since the program started.

EXAMPLES USING THE TIMER

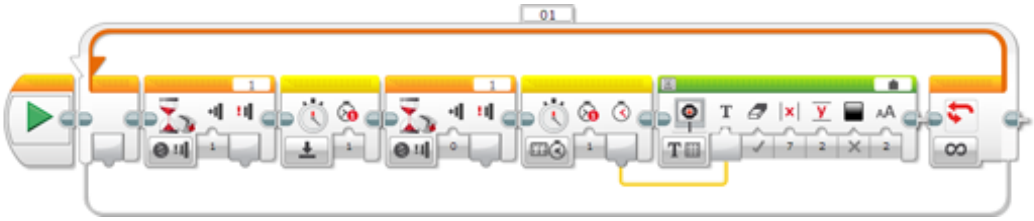
Examples of how you can use the Timer in your program are shown below.

Example 1: Make a Motor Move Once every Second



This program makes a motor turn 45 degrees exactly once every second, like a ticking clock. The program uses the Medium Motor block to turn the motor by 45 degrees, which will take a bit of time, but less than 1 second. Then the program needs to wait for the remainder of the 1-second interval to end before moving the motor again. To do this, the program starts timer 1 before starting the motor by using the Timer block in Reset mode. Then after the motor stops, a Wait block in Timer – Compare - Time mode waits for timer 1 to reach 1 second. This will make the total interval 1 second, including both the time the motor is moving and the time it is stopped.

Example 2:
Measure How Long a Touch Sensor is Held in



This program measures how long the touch sensor is held in each time it is pressed, and the result in seconds is displayed on the EV3 Display. The program uses the Wait block to wait for a touch sensor press and then again to wait for the release. After the press, timer 1 is reset using the Timer block in Reset mode. After the release, the elapsed time for timer 1 is measured using the Timer block in Measure – Time mode. The resulting number is wired to a Display block to display the number in seconds.

TIMER BLOCKS AND MODES

The table below shows all of the programming blocks and modes that you can use with Timer.

Block	Mode	Use
Wait	Timer - Compare – Time	Wait for a timer to reach a certain value.
Wait	Timer – Change - Time	Wait for a timer to change by a certain amount.
Loop	Timer	Repeat a sequence of blocks until a timer reaches a certain value.
Switch	Timer	Choose between two sequences of blocks based on a timer.
Timer	Measure	Read a timer, and get the result in seconds on a Numeric data wire.
Timer	Compare	Compare a timer to a threshold, and get the result on a Logic data wire.
Timer	Reset	Reset a timer to zero. The timer starts timing again immediately.

Tips and Tricks

Simple uses of timers may also be able to use the following blocks and modes:

Block	Mode	Use
Wait	Time	Wait for a certain amount of time.
Loop	Time	Repeat a sequence of blocks for a certain amount of time.

Using the Touch Sensor

The Touch Sensor detects whether the button on the face of the sensor is pressed in. You can use a Touch Sensor to detect, for example, when your robot drives into something. You could also use a finger press on a Touch Sensor to trigger an action.

The Touch Sensor can indicate that it is either pressed in, or not. It cannot measure how far or how hard the button is pressed in. The Touch Sensor gives Logic data (True or False). The position of the Touch Sensor button is called its State, and is True for pressed in and False for not pressed in (released).

The Touch Sensor can also keep track of whether the button has been pressed and then released in the past. This is called Bumped and is useful to detect, for example, finger presses. See Understanding Bumped for more information.

TOUCH SENSOR DATA

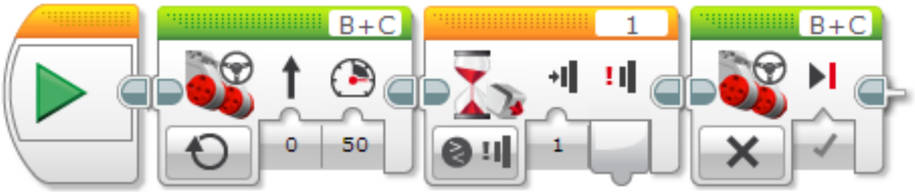
The Touch Sensor can give the following data:

Data	Type	Notes
State	Logic	True if the button is pressed in, False if not.
Pressed	Logic	True if pressed, False if not (same as State).
Released	Logic	False if pressed, True if not (opposite of State).
Bumped	Logic	True if the button has been pressed and released in the past. The next Bumped occurrence will then require a new press and release.

EXAMPLES USING PRESSED

Some examples of how you can use the Pressed data of the Touch Sensor are below.

Example 1: Drive until a Touch Sensor is Pressed

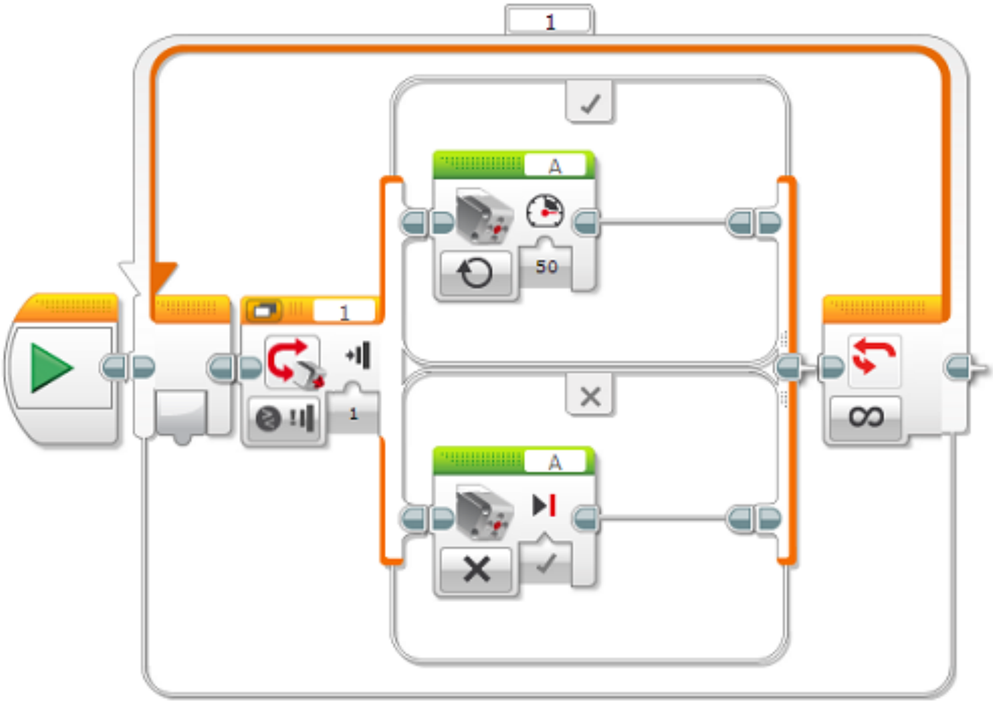


This program makes a robot drive straight forward until a Touch Sensor is pressed, then the robot is stopped. It uses the Wait block with the Touch Sensor - Compare – Touch mode to test for Pressed.

Tips and Tricks

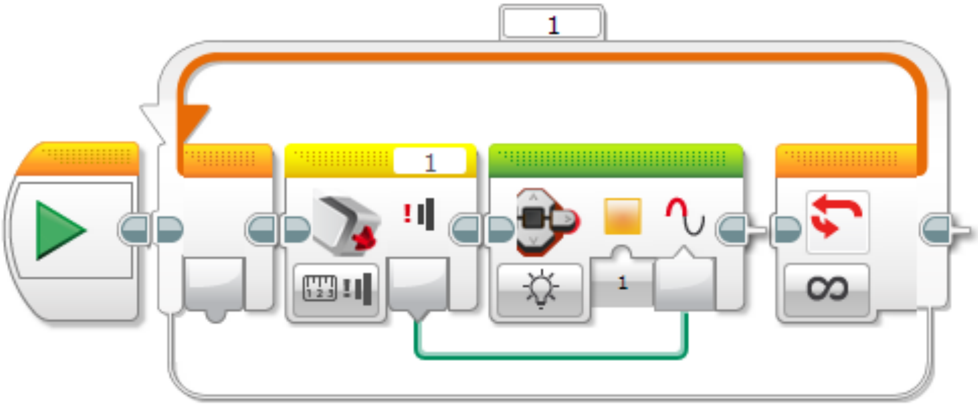
Remember to use the On mode of the Move Steering block when you want to drive while waiting for a sensor.

Example 2: Run a Motor whenever a Touch Sensor is Held in (Method 1)



This program makes a motor run whenever the Touch Sensor is pressed and held in. The motor is stopped whenever the Touch Sensor is released. The program uses a Switch block with the Touch Sensor - Compare – Touch mode to test for Pressed. The result of the test is used to choose between turning the motor on or off. The test is repeated continuously in a Loop.

Example 3: Make the Brick Status Light Pulse whenever a Touch Sensor is Held in

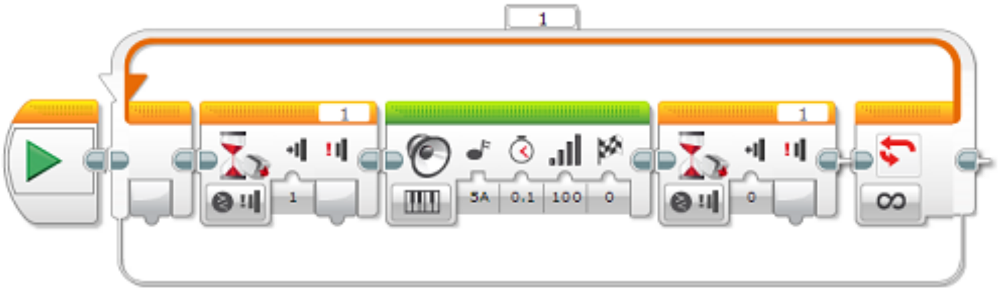


This program turns the Brick Status Light on in orange, and makes it pulse whenever the Touch Sensor is held in. It uses the Touch Sensor block in Measure mode to get the state of the Touch Sensor. The result is wired to the Pulse input of the Brick Buttons block with a Logic data wire.

EXAMPLES USING RELEASED

Some examples of how you can use the Released data of the Touch Sensor are below.

Example 4: Beep on each Touch Sensor Press

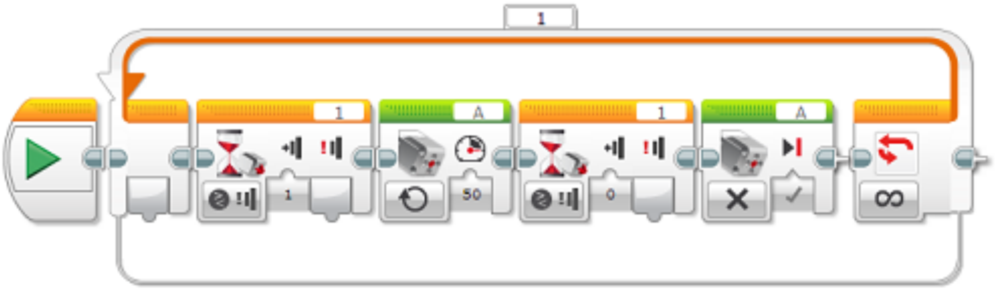


This program sounds a short tone each time the Touch Sensor is pressed. Only one tone is sounded for each press. The program uses a Wait block in the Touch Sensor - Compare – Touch mode to test for Pressed and then another Wait block to wait for Released before letting the loop continue.

Tips and Tricks

If you delete the Wait for Release from this program, you will find that the tone will repeat as long as the Touch Sensor is held in. This is because the Wait for Pressed will immediately continue to the next block if the Touch Sensor is already pressed. Try it!

Example 5: Run a Motor whenever a Touch Sensor is Held in (Method 2)



This program makes a motor run whenever the Touch Sensor is pressed and held in. The motor is stopped whenever the Touch Sensor is released. The program uses a Wait block in the Touch Sensor - Compare – Touch mode to wait for Pressed to start the motor, then another Wait to wait for Released before stopping the motor. The process is repeated in a Loop.

Tips and Tricks

This program does the same thing as Example 2 above, using a different method.

UNDERSTANDING BUMPED

In addition to telling you whether the Touch Sensor button is currently pressed or released, the Touch Sensor also keeps track of whether it has been pressed and released in the past, which is called Bumped. This makes it easy to find out whether a Touch Sensor has been pressed like a pushbutton, without needing to check it constantly for a press and then wait for the release.

Once a Touch Sensor indicates that it has been Bumped, it will not indicate Bumped again until the Touch Sensor is pressed and then released a new time. This makes it easy to make sure that, for example, each press corresponds to an action happening only once.

The table below shows an example where a Touch Sensor is pressed and released twice as a series of steps. The table shows the result of a program testing for Pressed, Released, and Bumped after each action.

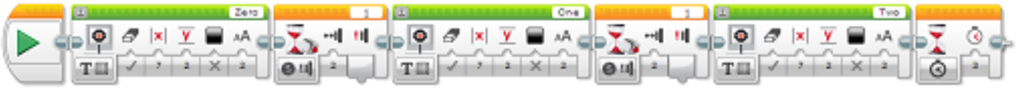
Step	Action	Pressed	Released	Bumped
1	Button starts released	False	True	False
2	Button is pressed in	True	False	False
3	Button is released	False	True	True
4	Button is still released, and the program tests the Touch Sensor again	False	True	False
5	Button is pressed a second time	True	False	False
6	Button is held in, and the program tests the Touch Sensor again	True	False	False
7	Button is released	False	True	True
8	Button is still released, and the program tests the Touch Sensor again	False	True	False

Note that when the button is held in, the Touch Sensor will continue to indicate Pressed each time the program tests it. However, once the button is released, the sensor will only indicate Bumped the first time the program tests it for Bumped. The sensor will not indicate Bumped again until it is pressed and released a new time.

EXAMPLES USING BUMPED

Some examples of how you can use the Bumped data of the Touch Sensor are shown below.

Example 6: Change the Display when the Touch Sensor is Pressed

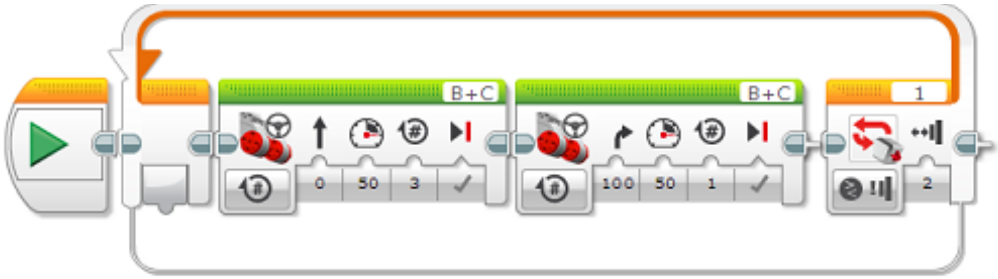


This program will make the EV3 Brick Display show “Zero”, then “One”, then “Two”, changing the Display each time the Touch Sensor is bumped (pressed and then released).

Tips and Tricks

If you change the Wait for blocks in this program to test for Pressed instead of Bumped, you will find that the display goes from “Zero” directly to “Two”, skipping “One”. Try it! This is because the Display blocks execute so quickly that when the second Wait for Pressed test happens, your finger is still holding the button in from the first press, so the second Wait for ends immediately. When testing for Bumped, only one test will succeed for each different press.

Example 7: Drive in a Pattern until the Touch Sensor is Pressed



This program makes a robot repeat a pattern of driving straight and then turning, until a Touch Sensor, acting as a "Stop" button on the robot, is pressed. After the Touch Sensor is pressed, the robot will stop after the next turn. The program uses a Loop in Touch Sensor mode to repeat the driving until the Touch Sensor is Bumped (pressed and then released).

Tips and Tricks

If you try this program using Pressed instead of Bumped, you will find that pressing the Touch Sensor usually does not make the robot stop. Try it! This is because the loop tests the sensor only briefly after the two Move blocks have completed. If you press and release the sensor while the Move blocks are running, the Pressed state will not be seen. Using Bumped, the Touch Sensor remembers that it was pressed and released in the past.

TOUCH SENSOR BLOCKS AND MODES

The table below shows all of the programming blocks and modes that you can use with the Touch Sensor.

Block	Mode	Use
Wait	Touch Sensor – Compare	Wait for the Touch Sensor to be Pressed, Released, or Bumped.
Wait	Touch Sensor - Change	Wait for the Touch Sensor state to change.
Loop	Touch Sensor	Repeat a sequence of blocks until the Touch Sensor is Pressed, Released, or Bumped.
Switch	Touch Sensor	Choose between two sequences of blocks depending on whether the Touch Sensor is Pressed or not, Released or not, or Bumped or not.
Touch Sensor	Measure	Get the current Touch Sensor state (Pressed or not) on a Logic data wire.
Touch Sensor	Compare	Test the Touch Sensor for Pressed, Released, or Bumped, and get the result on a Logic data wire.
Data Logging		See Data Logging.

INDEX

Using the Brick Buttons

The Brick Buttons are the five buttons (Left, Center, Right, Up, and Down) on the face of the EV3 Brick. You can use the Brick Buttons like a sensor to detect if a button is pressed, and to find out which button is pressed.

You can use the Brick Buttons to make your program respond to button presses. For example, you could make a robot arm lift up and down when the Up and Down buttons are pressed.

Tips and Tricks

The Back button on the EV3 is not included in the Brick Buttons. Pressing the Back button aborts a running program.

The Brick Buttons can also keep track of whether a button has been pressed and then released in the past. This is called Bumped, and it works the same as the Bumped state of the Touch Sensor. See Understanding Bumped Using the Touch Sensor – for more information.

BRICK BUTTONS DATA

The Brick Buttons can give the following data:

Data	Type	Notes
Button ID	Numeric	Indicates which button is currently pressed: 0 = None 1 = Left 2 = Center 3 = Right 4 = Up 5 = Down
Pressed	Logic	For a specified Button ID (1-5), True if the button is pressed, False if not.
Released	Logic	For a specified Button ID (1-5), False if the button is pressed, True if not.

		pressed, True if not.
Bumped	Logic	For a specified Button ID (1-5), True if the button has been pressed and released in the past. The next Bumped occurrence will then require a new press and release.

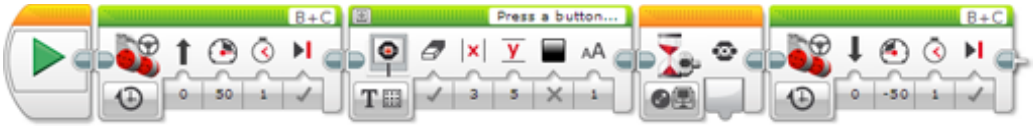
Tips and Tricks

The Brick Buttons cannot detect when more than one button is pressed at the same time. If you press two buttons at the same time, one button will override the other.

EXAMPLES USING THE BRICK BUTTONS

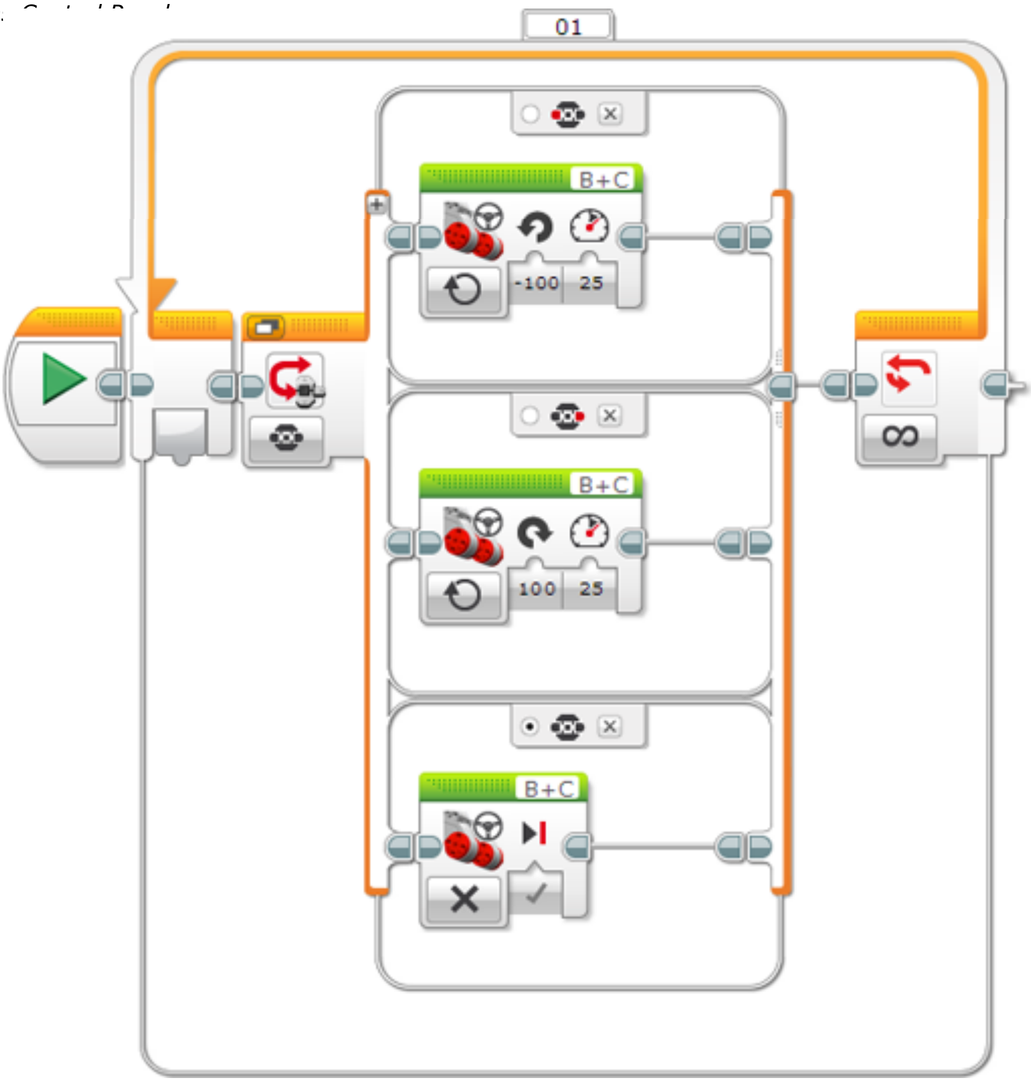
Some examples of how you can use the Brick Buttons in a program are below.

Example 1: Press a Button to Continue



This program makes a robot drive forward for one second, then it displays “Press a Button…”. It then waits for one of the Brick Buttons to be pressed before driving backward for one second. The program uses the Wait block with the Brick Buttons - Change mode to wait for any brick button to be pressed.

Example 2: A Brick Button

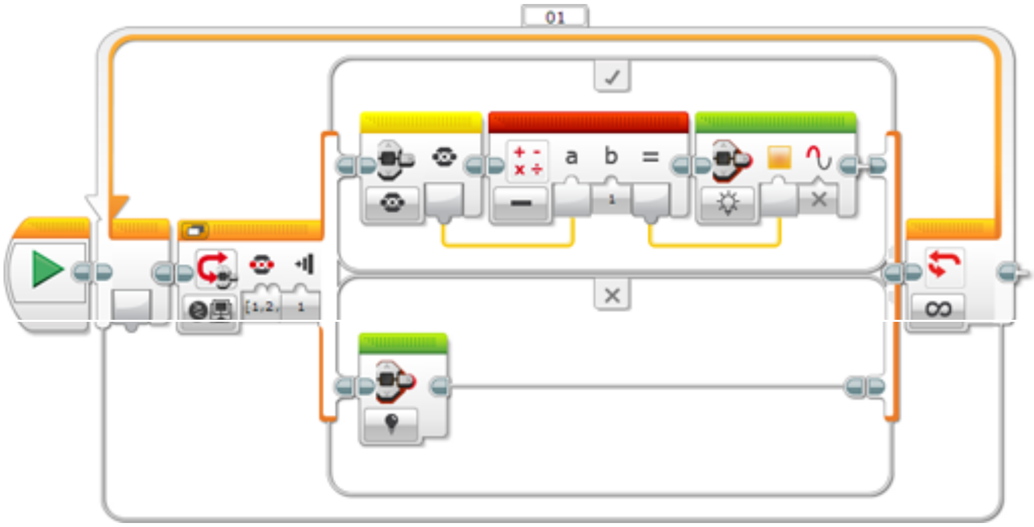


This program makes a robot turn left when the Left button is pressed, and turn right when the Right button is pressed. The program uses a Switch block with the Brick Buttons - Measure mode to find out which of the Brick Buttons is pressed. The three different cases in the Switch make the robot: turn left when the Left button is pressed, turn right when the Right button is pressed, and stop when no button is pressed. The test is repeated continuously in a Loop.

Tips and Tricks

Try using the "+" button on the Switch to add more cases. For example, you might make the robot drive forward and backward when you press the Up and Down buttons.

Example 3: Control the Brick Status Light Color with the Brick Buttons



This program turns the Brick Status Light on in green when you press the Left button, in orange when you press the Center button, and in red when you press the Right button. First, it uses a Switch block in the Brick Buttons - Compare mode to test whether the Left, Center, or Right button is pressed. If not, a Brick Status Light block turns the light off. If one of the three buttons is pressed, a Brick Buttons block in Measure mode gets the Button ID of the pressed button (1-3) on a Data Wire. A Math block then subtracts 1 from this value so that it can be used as the Color input (0-2) to the Brick Status Light block.

BRICK BUTTONS BLOCKS AND MODES

The table below shows all of the programming blocks and modes that you can use with the Brick Buttons.

Block	**Mode**	**Use**
Wait	Brick Buttons – Compare	Wait for one of the selected Brick Buttons to be Pressed, Released, or Bumped.
Wait	Brick Buttons – Change	Wait for the pressed Brick Button (Button ID) to change. If no Brick Button is pressed at the start of the block, this will wait for any Brick Button to be pressed.
Loop	Brick Buttons	Repeat a sequence of blocks until one of the selected Brick Buttons is Pressed, Released, or Bumped.
Switch	Brick Buttons - Measure	Choose between two or more sequences of blocks depending on which Brick Button is pressed.
Switch	Brick Buttons - Compare	Choose between two sequences of blocks depending on whether one of the selected Brick Buttons is Pressed, Released, or Bumped.
Brick Buttons	Measure	Get the Button ID of the currently pressed Brick Button (0 if none is pressed) on a Numeric data wire.
Brick Buttons	Compare	Test whether one of the selected Brick Buttons is Pressed, Released, or Bumped, and get the result on a Logic data wire.

Jsing the Motor Rotation Sensor

he Motor Rotation sensor is used to measure how far a motor has turned. A rotation ensor is built into the Medium Motor, the Large Motor, and the NXT Motor. The sensors in these motors can detect an amount of rotation in degrees. A full turn of a motor is 360 degrees of rotation.

You can also use the Motor Rotation sensor to find out what power level a motor is currently running at.

Tips and Tricks

A Motor Rotation sensor is used with a motor that is connected to a motor port on the EV3 Brick (A, B, C, or D). Motor Rotation sensors cannot be used with the EV3 sensor ports (1, 2, 3, and 4).

MOTOR ROTATION DATA

A Motor Rotation sensor can give the following data:

Data	**Type**	**Notes**
Degrees	Numeric	Amount of rotation in degrees
Rotations	Numeric	Amount of rotation expressed in rotations (degrees/360, as a decimal number)
Current Power	Numeric	Current motor power level if the motor is running (1-100), or 0 if the motor is stopped

RESETTING A MOTOR ROTATION SENSOR

A Motor Rotation sensor can be reset to zero at any point in a program. The sensor will then measure the total amount of rotation relative to the reset point. To reset a Motor Rotation sensor, use the Motor Rotation block in the Reset mode.

Tips and Tricks

If you measure a Motor Rotation sensor that has never been reset, you will get the total amount of rotation that the motor has turned since the program started.

MOTOR ROTATION DIRECTION AND TOTAL ROTATION

Forward rotation of a motor results in a positive number of degrees or rotations, and backward rotation results in a negative number. Rotation is always measured as the total amount of forward rotation since the sensor was last reset. Backward rotation is subtracted from any accumulated forward rotation.

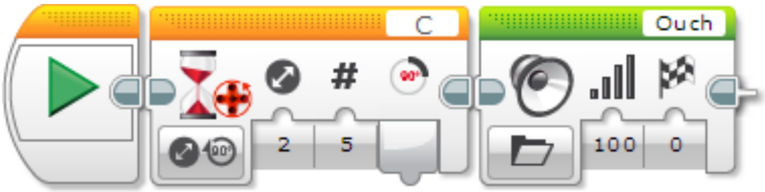
The table below shows an example of motor actions happening in several steps and the result of measuring the motor rotation after each step.

Step	Action	Motor Rotation is then:
1	Program starts, motor has not turned yet	0 degrees
2	Motor turns forward one full turn (360 degrees)	360 degrees
3	Motor turns forward one full turn again	720 degrees
4	Motor turns forward 60 degrees	780 degrees
5	Motor turns backward for 30 degrees	750 degrees
6	Motor Rotation is reset	0 degrees
7	Motor turns backward for 100 degrees	-100 degrees
8	Motor turns backward for 60 degrees	-160 degrees
9	Motor turns forward for 360 degrees	200 degrees

EXAMPLES USING THE MOTOR ROTATION SENSOR

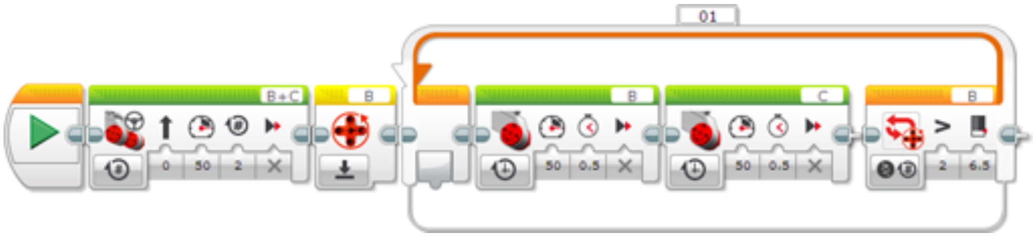
Some examples of how you can use the Motor Rotation sensor in your program are shown below.

Example 1: Make a Sound when your Robot is Pushed



This program makes a robot make a sound when it is pushed by hand so that the wheels move a little bit. The program uses the Wait block in the Motor Rotations – Change – Degrees mode to wait for the rotation sensor for motor C to change by 5 degrees in either direction. Then a Sound block makes a sound.

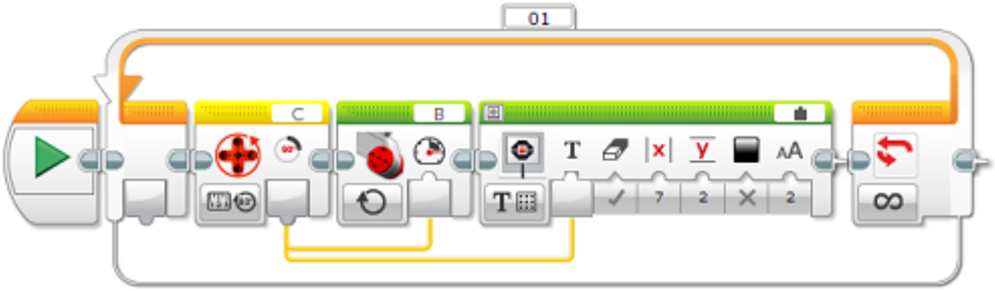
Example 2: Drive in a Pattern for a Certain Distance



This program makes a robot drive straight for 2 rotations and then drive in a zigzag pattern for 6.5 total rotations. It drives in a zigzag pattern by moving first one wheel, then the other. The zigzag pattern is repeated in a Loop until motor B has driven a total of 6.5 rotations. The program uses the loop in the Motor Rotation – Rotations mode to stop the loop when the rotation sensor for motor B measures a total of 6.5 rotations. To make the 6.5 rotations measure only the zigzag driving, not including the 2 straight rotations in the beginning, the rotation sensor for motor B is reset to zero before the zigzag driving using the Motor Rotation block in the

Reset mode.

Example 3: A Speed Control Dial



This program makes the motor connected to port C act like a speed control dial for the motor connected to port B. Turning the C motor forward and backward by hand will control the speed of the B motor. The program uses the Motor Rotation block in the Measure – Degrees mode to measure the degrees turned by motor C. This result is used for the Power input of a Large Motor block and also displayed using a Display block. The process is repeated in a Loop so that the speed is continuously updated.

Tips and Tricks

If the Power input to the Large Motor block is greater than 100, it will use 100% power.

MOTOR ROTATION BLOCKS AND MODES

The table below shows the programming blocks and modes that you can use with the Motor Rotation Sensor.

Block	Mode	Use
Wait	Motor Rotation - Compare	Wait for a rotation sensor to reach a certain value (Degrees, Rotations, or Current Power).
Wait	Motor Rotation - Change	Wait for a rotation sensor to change by a certain amount (Degrees, Rotations, or Current Power).
Loop	Motor Rotation	Repeat a sequence of blocks until a rotation sensor reaches a certain value (Degrees, Rotations, or Current Power).
Switch	Motor Rotation	Choose between two sequences of blocks based on a rotation sensor (Degrees, Rotations, or Current Power).
Motor Rotation	Measure	Read a rotation sensor (Degrees, Rotations, or Current Power), and get the result on a Numeric data wire.
Motor Rotation	Compare	Compare a rotation sensor (Degrees, Rotations, or Current Power) to a threshold, and get the result on a Logic data wire.
Motor Rotation	Reset	Reset a rotation sensor to zero.
Data Logging		See Data Logging.

Tips and Tricks

Motor Rotation sensors are also used internally in the following Action blocks and modes:

Blocks	Modes	Use
Medium Motor, Large Motor	On for Degrees, On for Rotations	Turn a motor by a certain number of degrees or rotations.
Move Steering, Move Tank	On for Degrees, On for Rotations	Drive using two Large Motors for a certain number of degrees or rotations.

Using the Gyro Sensor

The Gyro Sensor detects rotational motion. If you rotate the Gyro Sensor in the direction of the arrows on the case of the sensor, the sensor can detect the rate of rotation in degrees per second. You can use the rotation rate to detect, for example, when a part of your robot is turning, or when your robot is falling over.

In addition, the Gyro Sensor keeps track of the total rotation angle in degrees. You can use this rotation angle to detect, for example, how far your robot has turned.

GYRO SENSOR DATA

The Gyro Sensor can give the following data:

Data	Type	Notes
Angle	Numeric	Rotation angle in degrees. Measured from the last reset. Reset with the Reset mode of the Gyro Sensor block.
Rate	Numeric	Rotation rate in degrees per second.

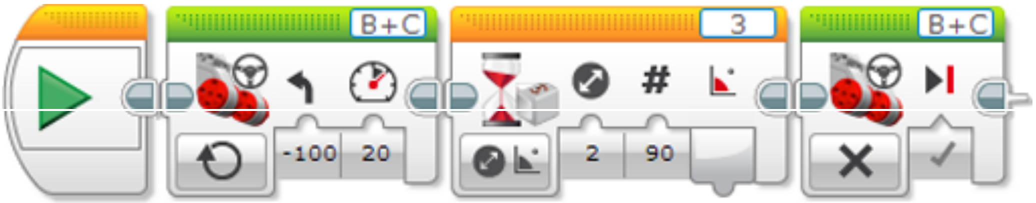
Tips and Tricks

- The Gyro Sensor can only detect motion around a single axis of rotation. This direction is indicated by the arrows on the case of the sensor. Make sure you attach the sensor to your robot in the correct orientation to measure rotation in the desired direction.
- The Angle and Rate can both be either positive or negative. Clockwise rotation is positive and counter-clockwise is negative.
- When connecting the Gyro Sensor to your EV3 Brick, you must hold it completely still in order to minimize “drifting”
- The Angle may “drift” over time and become less accurate. For best results, reset the angle using the Reset mode of the Gyro Sensor block before every motion that you want to measure the angle of.
- The Motor Rotation sensor can also measure rotation in degrees, but only for the rotating part of a motor.

EXAMPLES USING THE GYRO SENSOR

Examples of how you can use the Gyro Sensor in your program are shown below.

Example 1: Turn by a Specified Angle

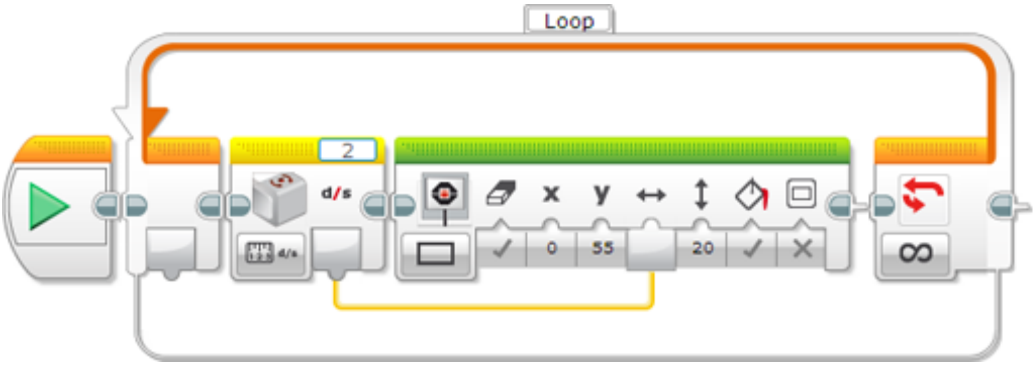


This program makes a robot turn 90 degrees to the left. It uses the Wait block with the Gyro Sensor - Change – Angle mode to wait for the rotation angle to change by 90 degrees.

Tips and Tricks

Tip: Because the program above uses the Change mode of the Wait block, it measures the change in angle relative to the start of the Wait Block. Therefore, it is not necessary to reset the Gyro Sensor before the movement.

Example 2: Display a Rotation Rate Meter



This program has the robot display the rotation rate graphically. The Gyro Sensor block measures the numeric value of the rotation rate, and this value is used to vary the width of a rectangle on the EV3 Display.

GYRO SENSOR BLOCKS AND MODES

The table below shows the programming blocks and modes that you can use with the Gyro Sensor.

Block	**Mode**	**Use**
Wait	Gyro Sensor - Compare	Wait for the rotation angle or rate to reach a certain value.
Wait	Gyro Sensor - Change	Wait for the rotation angle or rate to change by a certain amount.
Loop	Gyro Sensor	Repeat a sequence of blocks until the rotation angle or rate reaches a certain value.
Switch	Gyro Sensor	Choose between two sequences of blocks based on the rotation angle or rate.
Gyro Sensor	Measure	Measure the rotation angle and/or rate, and get the result on a Numeric data wire.
Gyro Sensor	Compare	Compare the rotation angle or rate to a threshold, and get the result on a Logic data wire.
Gyro Sensor	Reset	Reset the rotation angle to zero.
Data Logging	Gyro Angle Gyro Rate	See Data Logging.

Using the NXT Sound Sensor

The NXT Sound Sensor measures the intensity (volume) of sound using the microphone on the face of the sensor. For example, you can use the sound sensor to make your robot react to a loud sound such a hand clap.

SOUND SENSOR DATA

The Sound Sensor can give the following data:

Data	Type	Range	Notes
Sound Level (dB)	Numeric	0 to 100	Sound level, scaled to a percentage (0-100%)
Sound Level (dBA)	Numeric	0 to 100	Sound level, adjusted to approximate human ear sensitivity, and then scaled to a percentage (0-100%)

The Sound Level (dBA) value is adjusted to approximate the sensitivity of the human ear to different frequencies. This means that sound frequencies that are heard by the sensor but are hard for you to year will not result in a high sound level value.

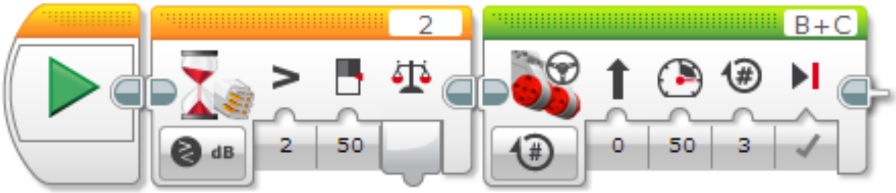
Tips and Tricks

Quiet sounds and normal talking usually result in sound levels less than 50%. A hand clap or loud voice will usually produce a level greater than 50%.

EXAMPLES USING THE SOUND SENSOR

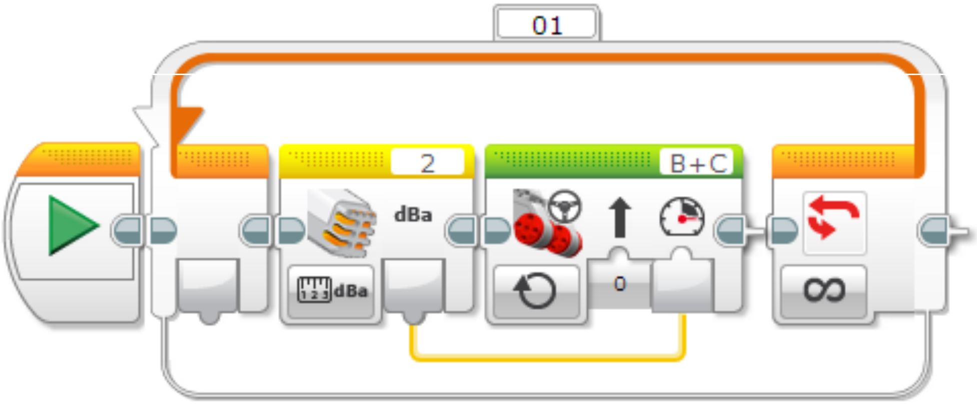
Some examples of how you can use the NXT Sound Sensor in your program are shown below.

Example 1: Start Your Robot with a Clap



This program makes your robot start driving when you clap your hands. It uses the Wait block in the Sound Sensor – Compare – dB mode to wait until the sound level rises above 50%.

Example 2: Sound Controlled Speed



This program makes your robot drive forward, with the speed of the robot controlled by the sound level. The louder you yell at the robot, the faster it will drive! The program uses the NXT Sound Sensor block in the Measure – dBA mode to get the sound level on a Numeric data wire. The result is wired to the Power input of a Move Steering block to make the sound level control the motor power. The process is repeated in a Loop so that the motor power is continuously adjusted based on new sound readings.

BLOCKS THAT CAN USE THE SOUND SENSOR

The table below lists the different programming blocks that can be used with the NXT Sound Sensor. Each block will have different modes for the dB and dBA data provided by the sensor.

Block	**Mode**	**Use**
Wait	Sound Sensor - Compare	Wait for the sound level to reach a certain value.
Wait	Sound Sensor - Change	Wait for the sound level to change by a certain amount.
Loop	Sound Sensor	Repeat a sequence of blocks until the sound level reaches a certain value.
Switch	Sound Sensor	Choose between two sequences of blocks based on the sound level.
NXT Sound Sensor	Measure	Measure the sound level and get the result on a Numeric data wire.
NXT Sound Sensor	Compare	Compare the sound level to a threshold and get the result on a Logic data wire.
Data Logging		See Data Logging.

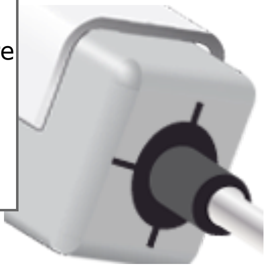
INDEX

Temperature

Quick links

- Temperature Sensor Data
- Examples Using the Temperature Sensor
- Temperature Sensor Blocks and Modes

Using the Temperature Sensor



The Temperature Sensor measures the temperature at the tip of its metal probe. You can measure the temperature in either degrees Celsius (°C) or degrees Fahrenheit (°F).

You can use the temperature sensor, for example, to measure the air temperature around your robot, or to track changes in water temperature.

TEMPERATURE SENSOR DATA

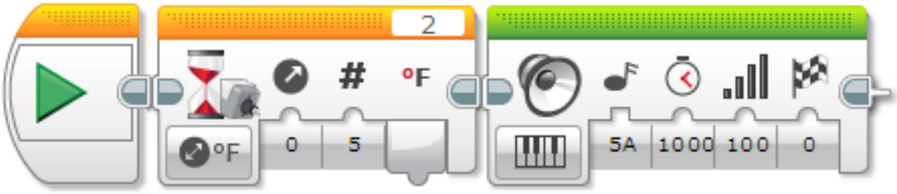
The Temperature Sensor can give the following data:

Data	Type	Range	Notes
Degrees Celsius	Numeric	-20 to 120	Temperature in degrees Celsius (°C)
Degrees Fahrenheit	Numeric	-4 to 248	Temperature in degrees Fahrenheit (°F)

EXAMPLES USING THE TEMPERATURE SENSOR

Some examples of how you can use the Temperature Sensor in your program are shown below.

Example 1: Beep when the Sensor Gets Warmer

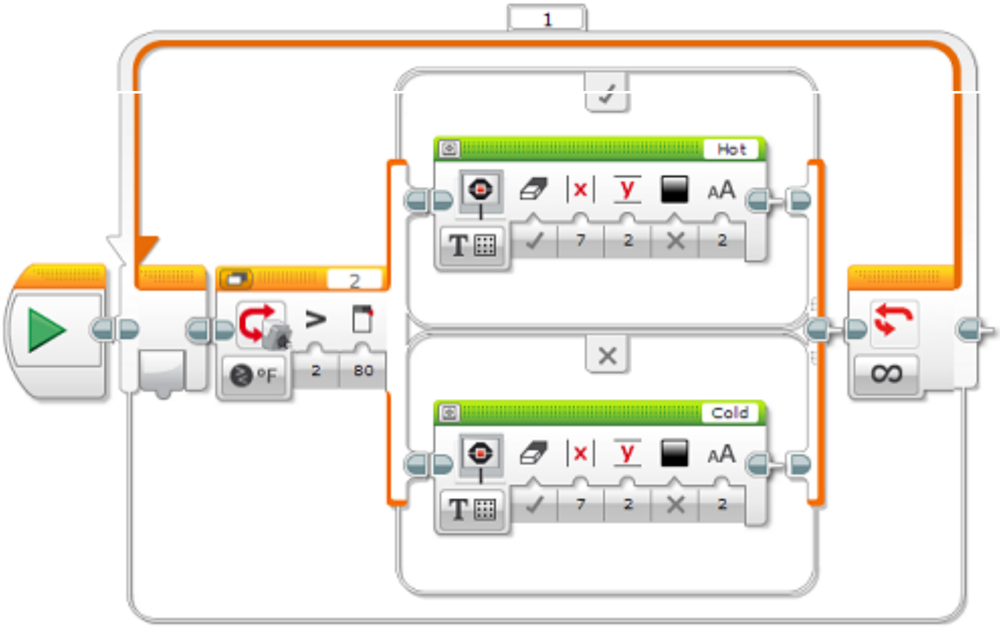


This program has the robot make a beep sound once the temperature at the end of the probe rises by 5°F. The Program uses the Wait block in the Temperature Sensor – Change – Fahrenheit mode to wait for the temperature of the probe to increase by 5°F.

Tips and Tricks

Try warming the probe with your hands while running this program.

Example 2: Display "Hot" or "Cold" depending on the Temperature

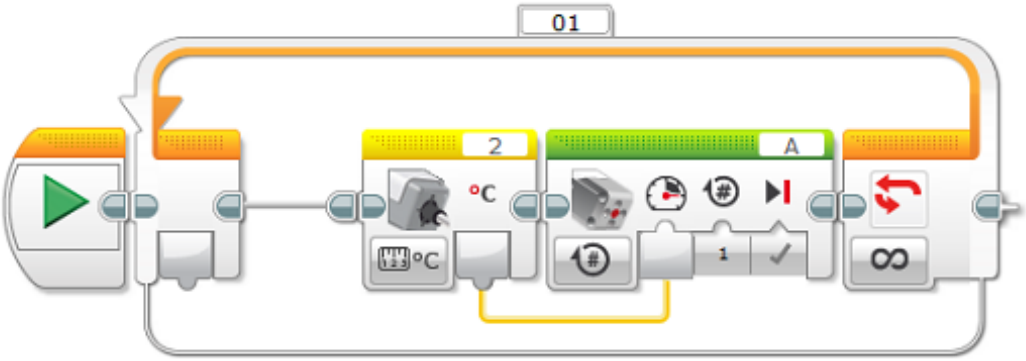


This program changes the display to show "Hot" or "Cold" depending on whether the temperature is greater than 80°F. The program uses a Switch block in the Temperature Sensor – Compare – Fahrenheit mode to choose between two different Display blocks.

Tips and Tricks

While running this program, try moving the temperature sensor probe between glasses of hot and cold water.

Example 3: A Temperature Controlled Motor



This program varies the speed of a motor based on the temperature of the temperature sensor. It uses the Temperature Sensor block in Measure – Celsius mode to measure the temperature. The result is connected to the Power input of a Medium Motor block using a data wire.

Tips and Tricks

Try sampling the temperature of different objects while this program is running.

TEMPERATURE SENSOR BLOCKS AND MODES

The table below shows all of the programming blocks and modes that you can use with the Temperature Sensor.

Block	Mode	Use
Wait	Temperature Sensor - Compare	Wait for the temperature to reach a certain value.
Wait	Temperature Sensor - Change	Wait for the temperature to change by a certain amount.
Loop	Temperature Sensor	Repeat a sequence of blocks until the temperature reaches a certain value.
Switch	Temperature Sensor	Choose between two sequences of blocks based on the temperature.
Temperature Sensor	Measure	Measure the temperature, and get the result on a Numeric data wire.
Temperature Sensor	Compare	Compare the temperature to a threshold, and get the result on a Logic data wire.
Data Logging		See Data Logging.

Using the Energy Meter

The Energy Meter is part of the Renewable Energy Add-On Set. If you connect the Energy Meter to a sensor port on the EV3 Brick, the meter can provide data about the electrical energy storage, input, and consumption of the electrical components connected to it.

You could use the Energy Meter to program your own automated energy experiments. For example, you might program a motor to run only when there is enough input power from a solar panel or enough stored energy in the battery.

ENERGY METER DATA

The Energy Meter provides seven kinds of Numeric data:

Data	Type	Range	Units	Notes
In Voltage (V)	Numeric	0.0 to 10.0	Volts (V)	Input voltage
In Current (A)	Numeric	0.0 to 0.3	Amps (A)	Input current
In Wattage (W)	Numeric	0.0 to 3.0	Watts (W)	Input power
Out Voltage (V)	Numeric	0.0 to 10.0	Volts (V)	Output voltage
Out Current (A)	Numeric	0.0 to 0.5	Amps (A)	Output current
Out Wattage (W)	Numeric	0.0 to 5.0	Watts (W)	Output power
Joule (J)	Numeric	0 to 100	Joules (J)	Stored energy

See the documentation for the Renewable Energy Add-On Set for more information.

Using the data from the Energy Meter in a program is very similar to using data from other sensors that give Numeric data. See, for example, Using the Color Sensor.

BLOCKS THAT CAN USE THE ENERGY METER

The table below lists the different programming blocks that can be used with the Energy Meter. Each block will have different modes for each of the seven kinds of data provided by the meter.

Block	Mode	Use
Wait	Energy Meter - Compare	Wait for a data reading to reach a certain value.
Wait	Energy Meter - Change	Wait for a data reading to change by a certain amount.
Loop	Energy Meter	Repeat a sequence of blocks until a data reading reaches a certain value.
Switch	Energy Meter	Choose between two sequences of blocks based on a data reading.
Energy Meter	Measure	Get a data reading on a Numeric data wire.
Energy Meter	Compare	Compare a data reading to a threshold and get the result on a Logic data wire.
Data Logging		See Data Logging