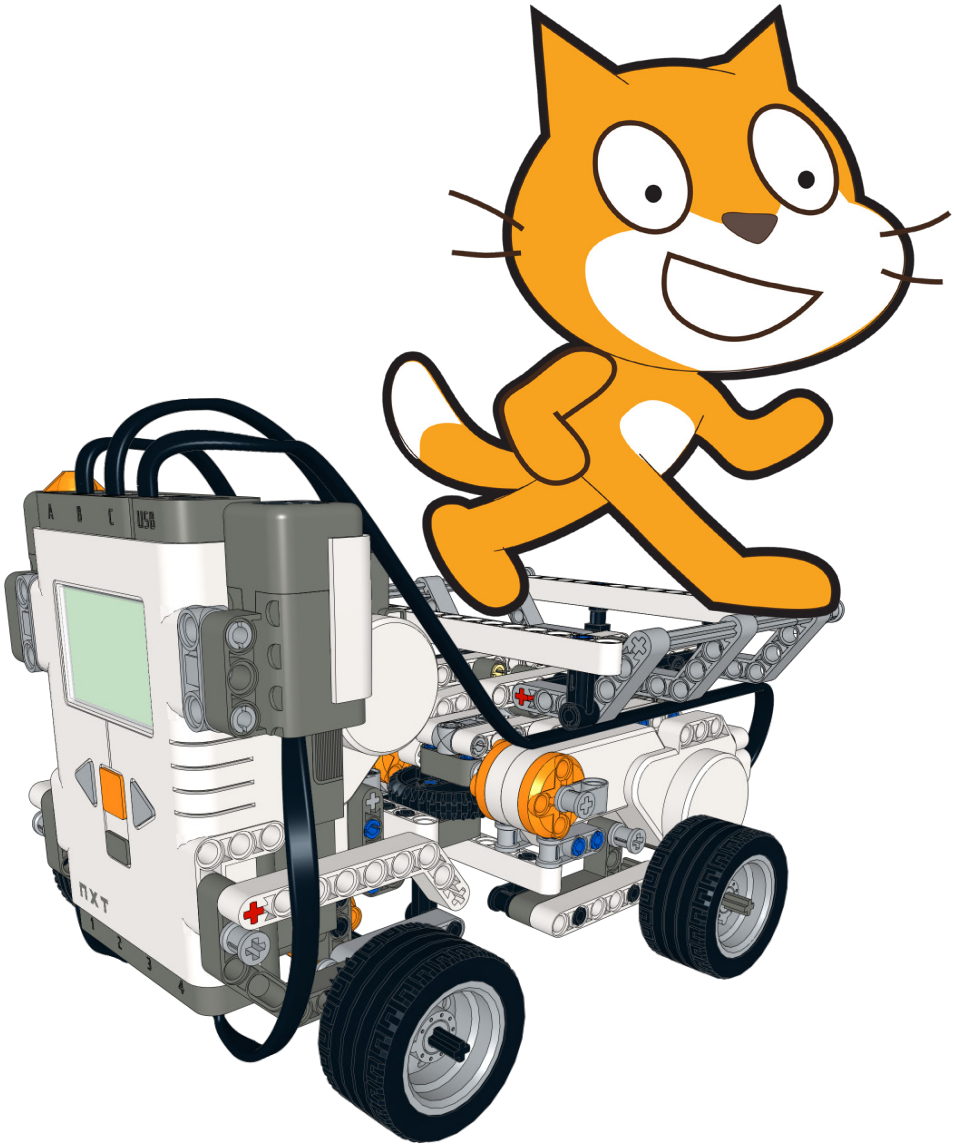
SCRATCH 2

Robot Revolution

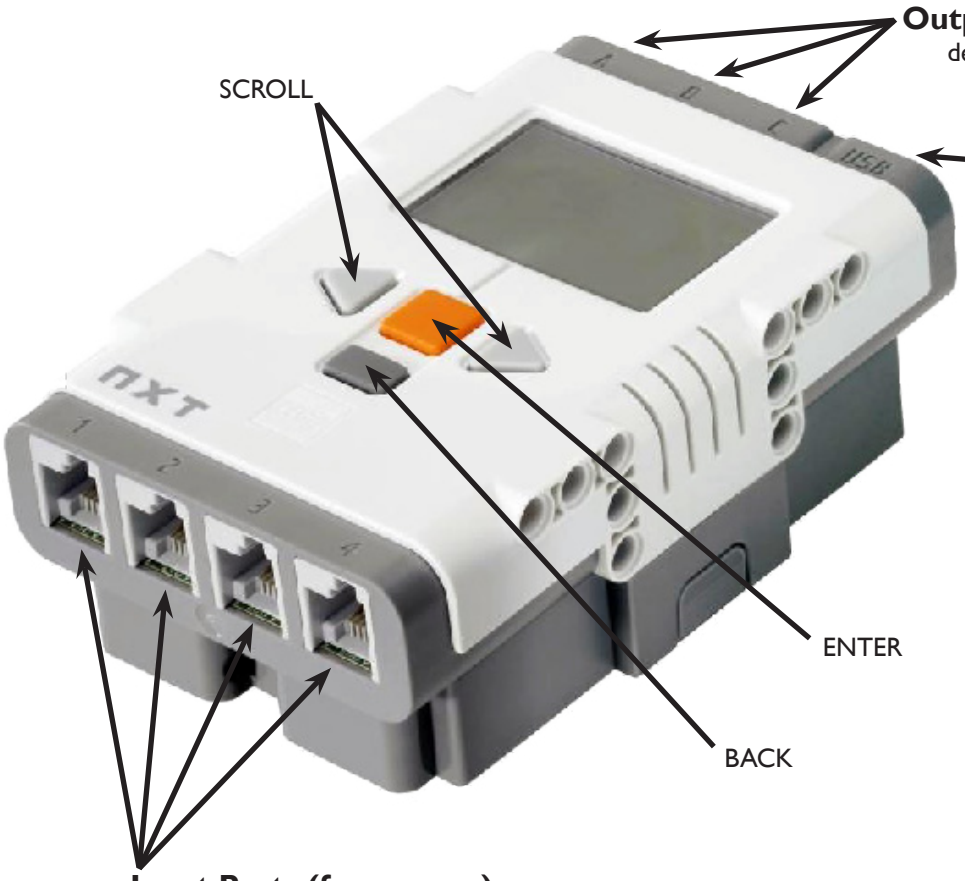


Summer Camp 2016

Wichita State University

smithlearning.com/robotics

NXT Key Parts



Output Ports (for motors)
defaults are _____ and _____

USB Port
connect to computer

Input Ports (for sensors)
defaults are
"**T**he **S**ensor **L**ine **U**p"

_____, _____, _____, _____


Touch Sensor


Sound Sensor


Light Sensor


Ultrasonic Sensor

Color Sensor


Motor

Configuring a Sensor

(tell your NXT what is plugged into it, and where)

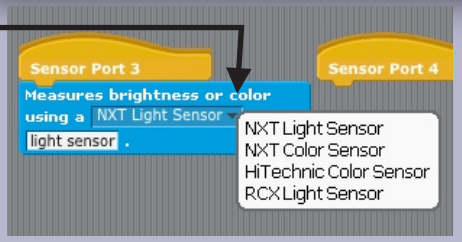
Suppose we have an NXT with a light sensor connected on port 3. How do we tell Enchanting this?

1. Click on the "Sensing" palette.

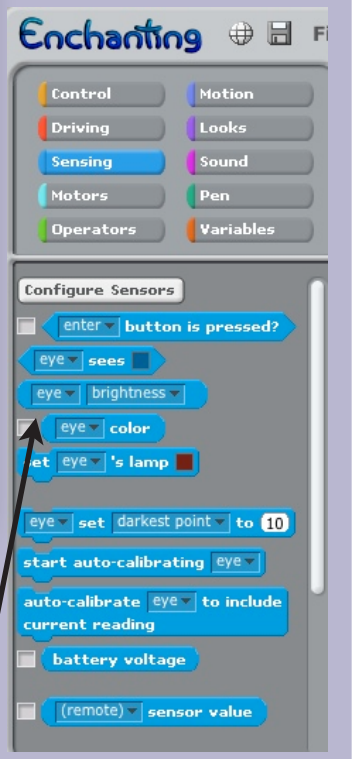
2. Click on the "Configure Sensors" button. A dialog comes up.

3. Find the light sensor configuration block on the left-hand side of the dialog. Drag it to "Sensor Port 3" and drop it there.

4. Choose what sort of light sensor you have.



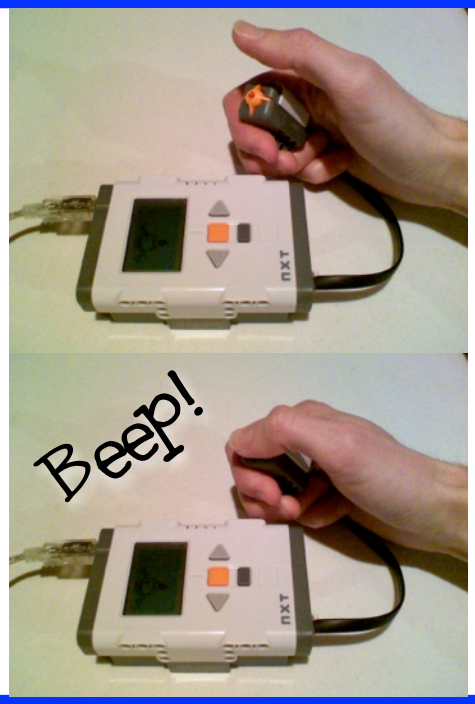
5. Optionally, give it a descriptive name.



7. You now have new blocks to use!

Morse Key

Press a button to make a sound.



YOU NEED

An NXT with a touch sensor connected to port 1.

Morse Key

GET READY

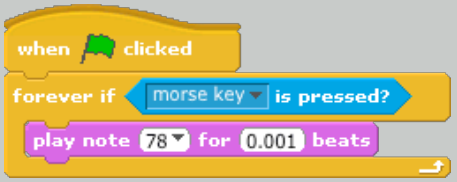
Select the Sensing palette, and click on the Configure Sensors button [it is above all the light blue blocks].

Configure a touch sensor, like this:

Name it.

Make sure that your touch sensor is on port 1 on the actual robot.

TRY THIS CODE



DO IT!

Click the and wait for the program to run on the robot. Press and release the touch sensor. What happens?

GO FARTHER

What would happen if you changed the note or the duration?

Where could you use pick random 50 to 90 or set instrument to 2?

SCRATCH Planning Sheet

What should the program do:

Variables:

Inputs:

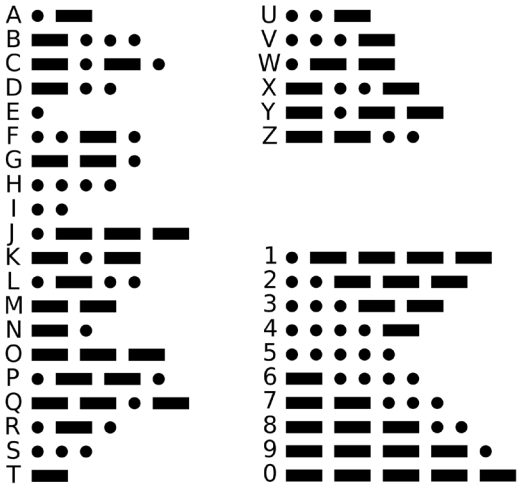
Outputs:

Sketch:

How the program should work:

International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.



Range Finder

See how far away something is.

YOU NEED

An NXT with an ultrasonic sensor connected to port 4.

Range Finder

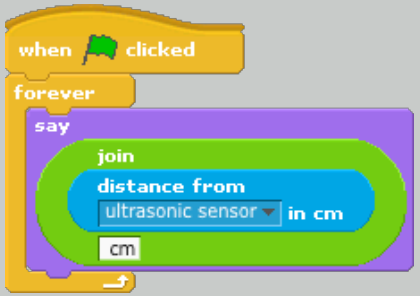
GET READY

From the Sensing palette, choose the Configure Sensors button.

Configure a ultrasonic sensor.

Make sure the port you configure it for matches the port the real ultrasonic sensor is plugged into!

TRY THIS CODE



DO IT!

Click the and wait for the program to run.

Point the ultrasonic sensor at different objects.
What do you see on the screen?

GO FARTHER

What are the largest and smallest values you see?
Are there any gaps where you get no numbers?

SCRATCH Planning Sheet

What should the program do:

Variables:

Inputs:

Outputs:

Sketch:

How the program should work:

Sound Meter

Create a sound-level meter.

YOU NEED

An NXT with a sound sensor* connected to port 2.

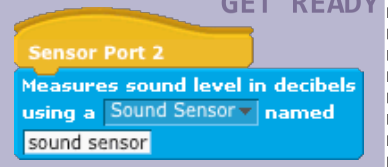
* Not included in all NXT sets.

Sound Meter

GET READY

From the Sensing palette, choose the Configure Sensors button.

Configure a sound sensor.



Paint a new costume for the sprite -- make it a dot!

TRY THIS CODE

DO IT!

Click the and wait for the program to run.

Whisper, talk, sing, shout, or blow on the sound sensor.

GO FARTHER

Draw a gauge on the background.
Adjust how far the meter moves back and forth.

SCRATCH Planning Sheet

What should the program do:

Variables:

Inputs:

Outputs:

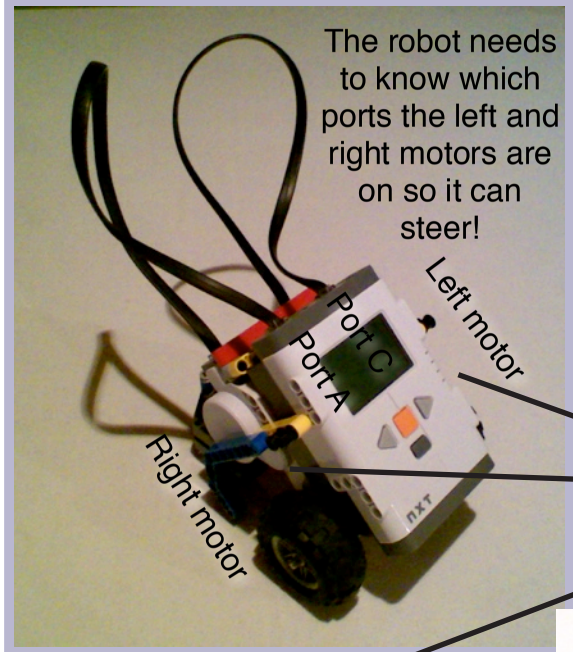
Sketch:

How the program should work:

Differential Drive

(lets your two-wheel robot drive around)

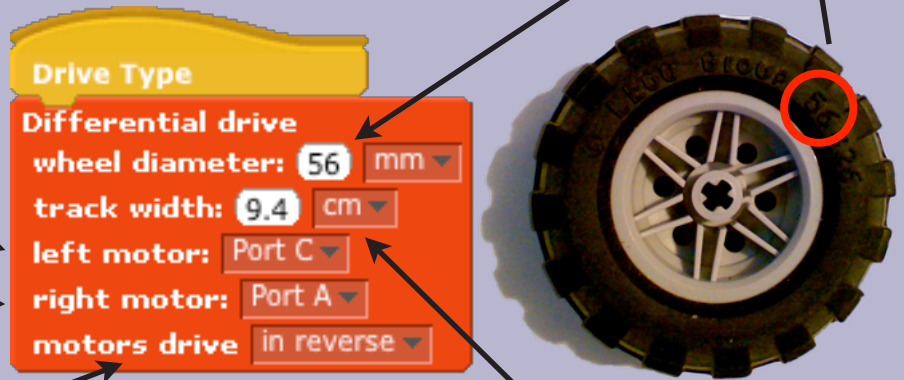
From the Driving palette, choose the Configure Drive Type button. Configure a differential drive.

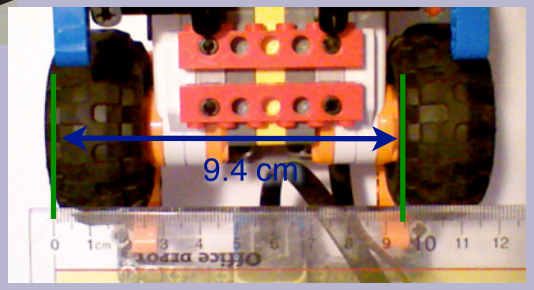


The robot needs to know which ports the left and right motors are on so it can steer!

If your robot has two wheels, it is using a differential drive!

The first number embossed on a LEGO tire is the tire's diameter in millimeters. This tire is 56 mm across.





If you tell the robot to drive forwards and it goes backwards, swap the direction that you've configured the motors to drive in the drive type.

The track width is how far apart the wheels are. Measure from a place on the left wheel to that same place on the right wheel. Here, measuring from the left-most point of the left wheel to the left-most point of the right wheel, the track width is about 9.4 cm.

Activities & Questions

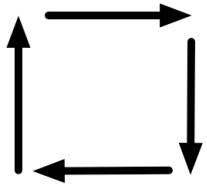
Activities

For each path below, program your robot to drive as indicated:

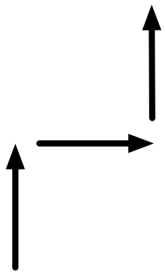
1 Forward, reverse

2 Forward, turn 180˚, forward

3 Right turns

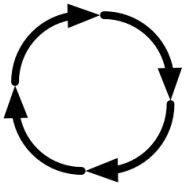


4 Right turn, left turn



5 Curve right

6 Circle

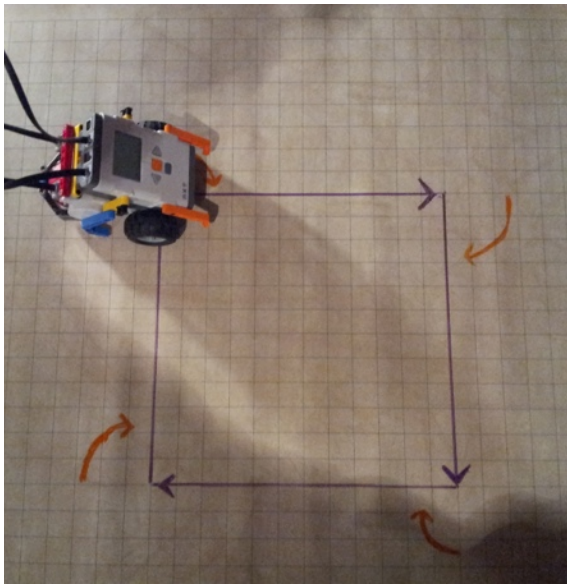


Question

For **part 2**, by how many degrees should each motor rotate in order to turn the robot 180°? (assuming "standard car" configuration)

Square Dance

Drive in a square.



YOU NEED

An NXT with two driving wheels.

Square Dance

GET READY

From the Driving palette, choose the Configure Drive Type button. Configure a differential drive, using the instructions on Card #4. Measure numbers to match your robot.

TRY THIS CODE

DO IT!

Click the and wait for the program to run. Unplug the USB cable and place the robot where you want it to start. Press the orange "enter" button on the NXT.

Does the robot "dance" in a square? (Is it configured properly?) Note that you'll never get 100% accurate motion.

Press the orange "enter" and dark grey "exit" buttons on the NXT to quit the program before plugging it back into the computer.

GO FARTHER

What other shapes and patterns could you make the robot "dance"? Could you have one robot "dance" a square inside a square being "danced" by another robot, at different speeds?

SCRATCH Planning Sheet

What should the program do:

Variables:

Inputs:

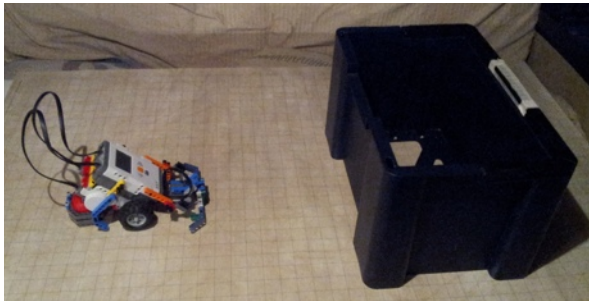
Outputs:

Sketch:

How the program should work:

Crash Test

Drive into a wall and then stop.



YOU NEED

An NXT with two driving wheels and a bumper (touch sensor) on port 1.

Crash Test

GET READY

Driving → Configure Drive Type as on Card #4.

Sensing → Configure Sensors.

Configure a touch sensor for use as a bumper.

TRY THIS CODE

DO IT!

Click the and wait for the program to run. Unplug the USB cable and place the robot where you want it to start. Press the orange “enter” button on the NXT.

The robot drives until it hits something. Pick it up and press the “enter” button again. It drives until it hits something.

GO FARTHER

Can you make the robot say something when it hits a wall, such as, “Ouch, my nose” or “Ooof”? Could you draw eyes on it and make it wince?

How would you make the robot explore a room? Try it out!

SCRATCH Planning Sheet

What should the program do:

Variables:

Inputs:

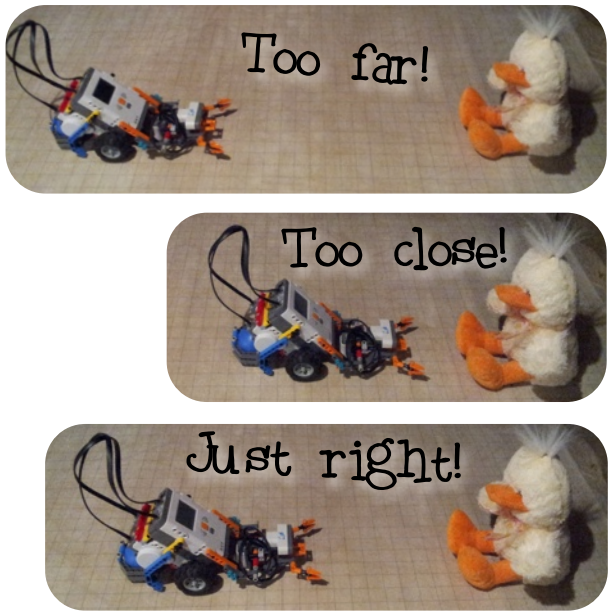
Outputs:

Sketch:

How the program should work:

Shy Puppy

Robot follows you around and wants to be close, but not too close!



YOU NEED

An NXT with two driving wheels and an ultrasonic sensor pointing forwards

Shy Puppy

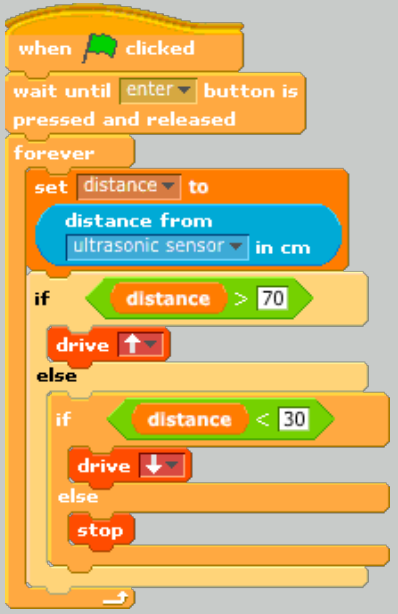
GET READY

Driving → Configure Drive Type as on Card #4.

Sensing → Configure Sensors.

Configure an ultrasonic sensor.

TRY THIS CODE



DO IT!

Click the flag and wait for the program to run. Unplug the USB cable and press the orange "enter" button on the NXT.

Try walking away from the robot. Does it follow? Try going towards it. Does it shy away? Does it ever stand still?

GO FARTHER

What would happen if you made the robot turn?

What do two or more robots together do?

Add a face to the robot, make it "speak", or dress it up!

SCRATCH Planning Sheet

What should the program do:

Variables:

Inputs:

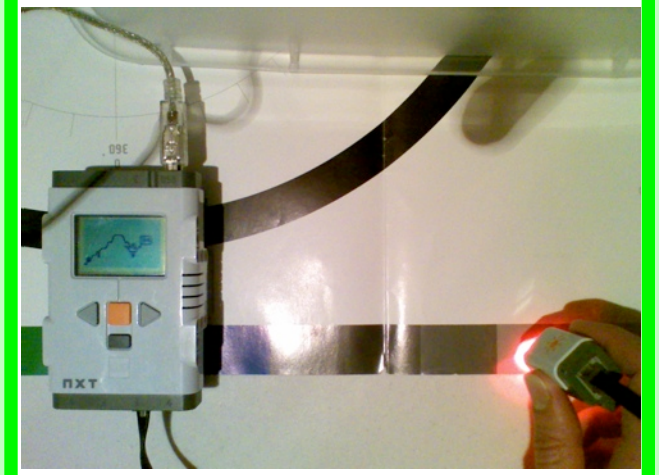
Outputs:

Sketch:

How the program should work:

Sparkline

Graph a changing value.



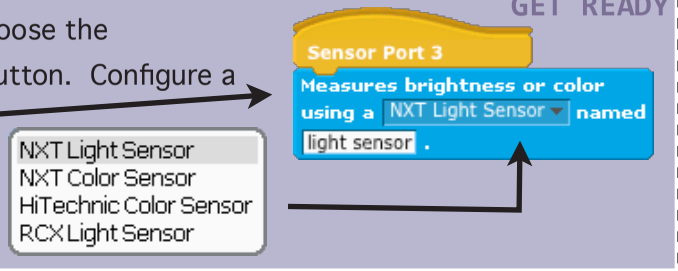
YOU NEED

An NXT with a light or colour sensor on port 3.

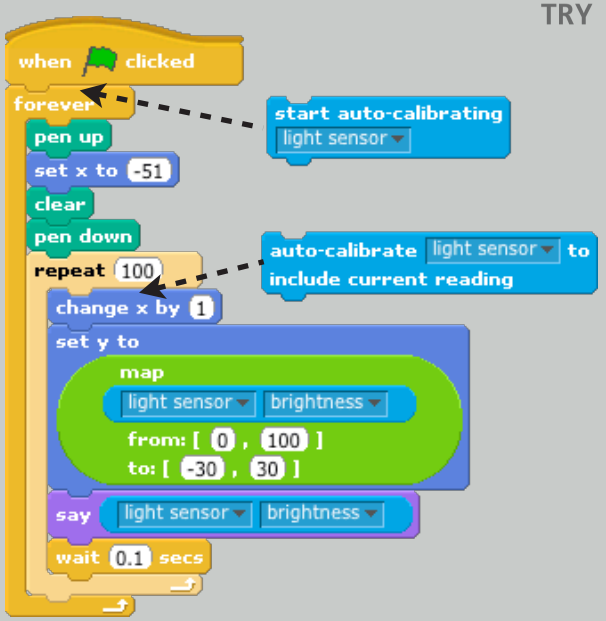
Sparkline

GET READY

In Sensing, choose the Configure Sensors button. Configure a light or colour sensor, choosing the one you have.



TRY THIS CODE



DO IT!

Click the flag to start.
Move the light sensor over different colours and shades. Move it closer and farther from a surface. Point it at a light. Change the angle. What happens?

GO FARTHER

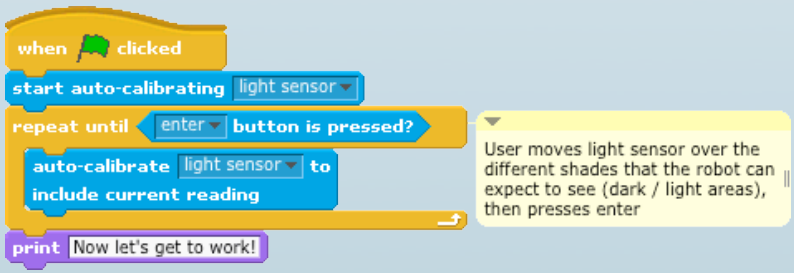
What happens if you add in the auto-calibration blocks?
Try drawing a white line on a black background.
Change the lamp colour.

Activities & Questions

Auto-Calibration

Here's an example of an auto-calibration technique that can be used at the start of a program:

Figure 2.1 Auto-Calibration Example



The "raw" brightness level from the LEGO® Light Sensor varies depending on the lighting conditions you are working with. This can make it difficult to use your programs in different lighting conditions, for example moving from inside to outside.

Auto-calibration allows you to show the robot the different brightness levels it can expect to see in its current conditions. It will then translate the raw brightness values such that the darkest area it was shown produces 0, and the brightest produces 100. All the shades in between will fall somewhere between 0 and 100.

Line following is an important part of your programming toolbox when using the LEGO® NXT. It allows you to create paths through the environment for your robot to follow, meaning that your robot can perform tasks that require basic navigation. There are many different ways of creating a line follower depending on how many light sensors you have available, however in this activity it's assumed that you only have one.

Figure 2.2 Following a Black Line

Let's say we want to follow a black line, like the one pictured above. You might think that the way to achieve this would be to ensure that the robot's light sensor is always directly over a black surface.

However, there's a problem with this strategy:

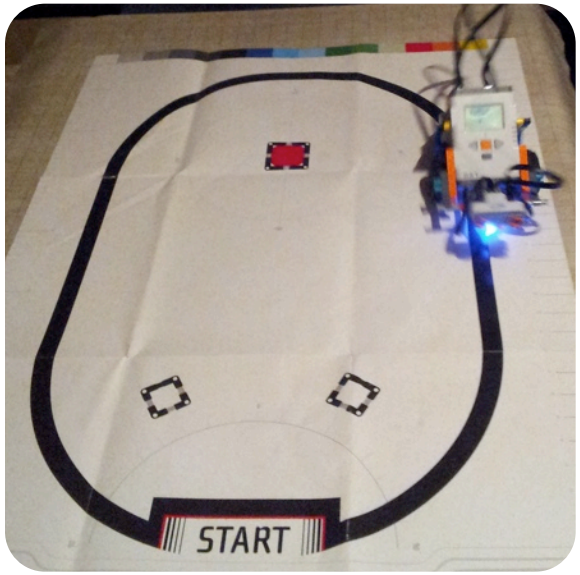
Question A
What happens when we detect the white surface?
How do we know if we've veered off to the left or the right?

The solution is to instead follow the *edge* of the black line, rather than the line itself. Either edge is fine, we'll use the lower edge in this example.

Figure 2.3 Following the *Edge* of a Line

Follow That Line!

Follow a line around.



YOU NEED

An NXT with two driving wheels and a light/colour sensor pointing downwards.

Follow That Line!

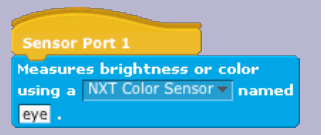
GET READY

Driving → Configure Drive Type as on Card #4.

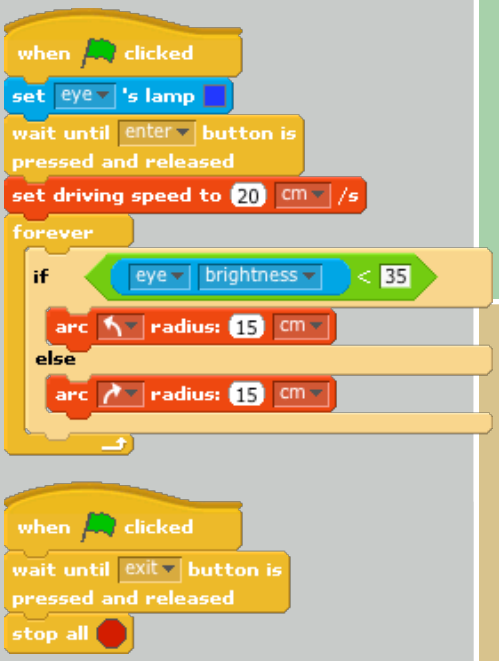
Sensing → Configure Sensors.

NXT Light Sensor
NXT Color Sensor
HiTechnic Color Sensor
RCX Light Sensor

Configure and name a light or colour sensor, choosing the type that matches your sensor.



TRY THIS CODE



DO IT!

Click the ⚑ and wait for the program to run. Unplug the USB cable, place the robot on or beside a line, and press the orange “enter” button on the NXT.

Press the dark grey “exit” button to stop the robot.

GO FARTHER

What happens if you adjust the numbers? Does the line-following course make a difference? Can you follow the edge of a table? Where is the best place to put your light sensor? What would you do with a second or third sensor? Draw eyes and make them look left and right!

SCRATCH Planning Sheet

What should the program do:

Variables:

Inputs:

Outputs:

Sketch:

How the program should work:

Sumo

(card 1 of 2)

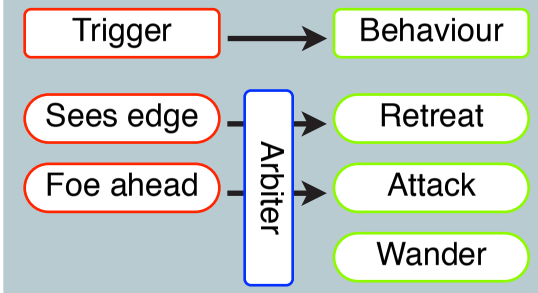
Push your opponent out of a ring.

YOU NEED

Each competitor needs an NXT with two driving wheels, an ultrasonic sensor, and a light or colour sensor.

Sumo

THE PLAN



If it has nothing better to do, the robot wanders around. If it sees a foe ahead of it, the robot attacks, trying to push the enemy out of the ring. Most importantly, if it sees the edge, the robot retreats from it, in an effort to stay in the ring.

GET READY

Driving → Configure Drive Type as on Card #5.

Sensing → Configure Sensors. Configure an ultrasonic sensor and a light/colour sensor.

Create a bunch of costumes for your sprite, using the text tool to display the following words in large letters, “Countdown”, “Wander”, “Attack”, “Retreat” and “Press Button to Start”. →

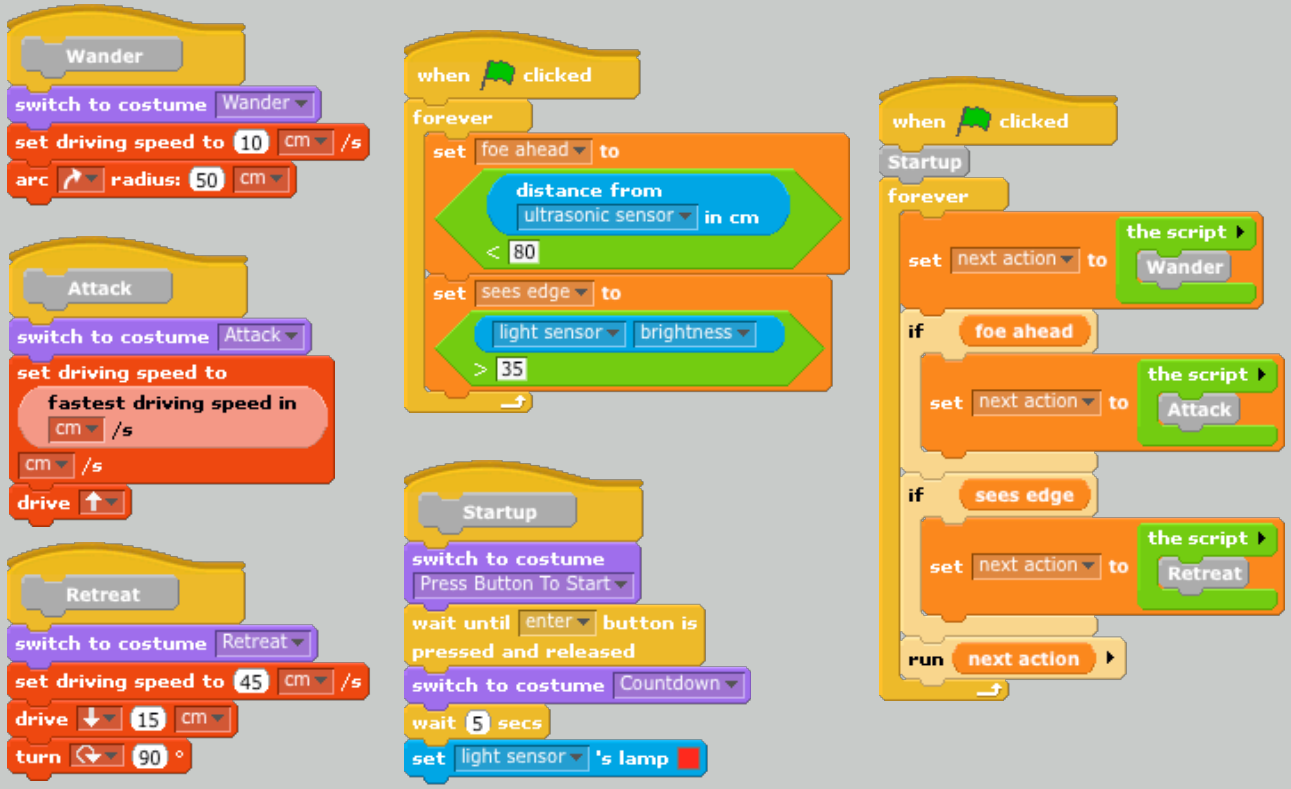
Variables → Make a variable. Create variables named “next action”, “foe ahead”, and “sees edge”.

Sumo

(card 2 of 2)

TRY THIS CODE

Variables → Make a block. You'll make blocks called "Attack", "Retreat", "Wander", and "Startup".



DO IT!

Click the and wait for the program to run.

Unplug the USB cable and place the robot on the sumo board. When you are told to start your robot, press the orange "enter" button on the NXT.

GO FARTHER

Can you add other behaviours to the robot, or change some of the existing ones? What could you do to make your sumo robot better?

Would this method of programming work for other robots?

SCRATCH Planning Sheet

What should the program do:

Variables:

Inputs:

Outputs:

Sketch:

How the program should work:

SCRATCH Planning Sheet

What should the program do:

Variables:

Inputs:

Outputs:

Sketch:

How the program should work:

SCRATCH Planning Sheet

What should the program do:

Variables:

Inputs:

Outputs:

Sketch:

How the program should work:

SCRATCH Planning Sheet

What should the program do:

Variables:

Inputs:

Outputs:

Sketch:

How the program should work:

MONDAY REFLECTIONS

Something I learned today:

My favorite "Discovery" (can be an "ah-ha", or something that went right or wrong):

Tomorrow I would like to try:

TUESDAY REFLECTIONS

Something I learned today:

My favorite "Discovery" (can be an "ah-ha", or something that went right or wrong):

Tomorrow I would like to try:

WEDNESDAY REFLECTIONS

Something I learned today:

My favorite "Discovery" (can be an "ah-ha", or something that went right or wrong):

Tomorrow I would like to try:

THURSDAY REFLECTIONS

Something I learned today:

My favorite "Discovery" (can be an "ah-ha", or something that went right or wrong):

Tomorrow I would like to try:

FRIDAY REFLECTIONS

Something I learned today:

My favorite "Discovery" (can be an "ah-ha", or something that went right or wrong):

Tomorrow I would like to try: