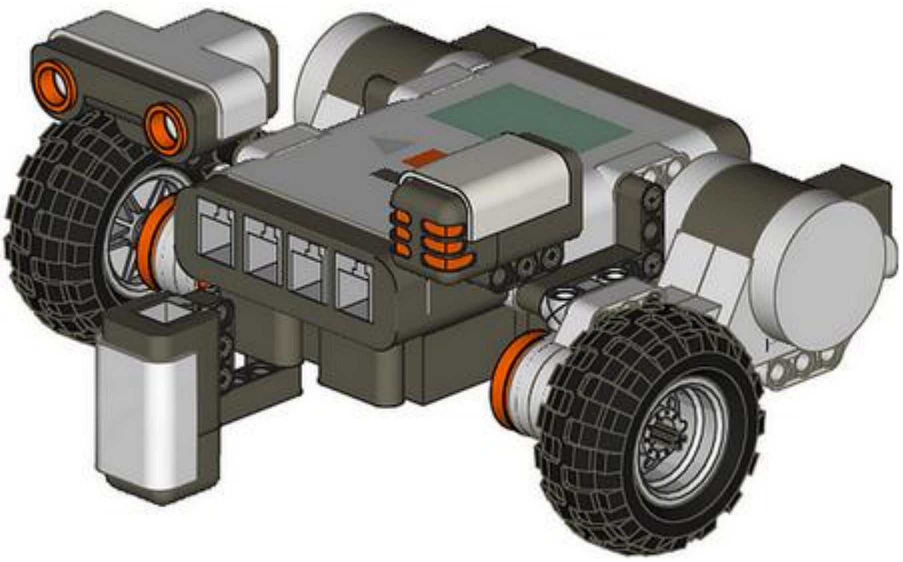
Robo Jam



School of Computer Science, University of Lincoln

University of Lincoln Computing Society

Contents

Welcome

Dear Participants,

I would like to welcome you to the Lincoln School of Computer Science Robo-Jam where you will have the chance to program your own robots.

The aim of the Robo-Jam is to improve student engagement and to enrich student experience. The Robo-Jam is intended to be a regular activity in the future. It is also envisaged that the designed tasks will be adopted for future use in outreach activities in high school, open days, etc.

This will be the first time the Robo-Jam has been run since the project started at the beginning of June. With this in mind we are targeting computing students from different

The event is split up into two consecutive days, one day will be used for training and the other will be used for a small competition. During this event we will be making use of the Lego Mindstroms NXT 1.0 kit (NXTs), allowing you to make use of a range of different sensors and motors. The days will start around 9am till 5pm with a lunch break in-between, you are welcome to schedule your day as you see fit.

Day 1 will consist of a variety of tutorials that will help you understand how to program the NXT. You will start of using the NXT-G software and finish up using C# to control the NXT.

Day 2 will consist of a variety of challenges, for each challenge you will gain points depending on how well you do. At the end of the day the group with the most points will win a prize.

This event has been funded by the Lincoln School of Computer Science, LNCD (an educational research group), and the University of Lincoln Computing Society.

Meet the NXT

Figures 1 and 2 below show the layout of both the Lego NXT Brick and the Lego NXT-G programming environment.

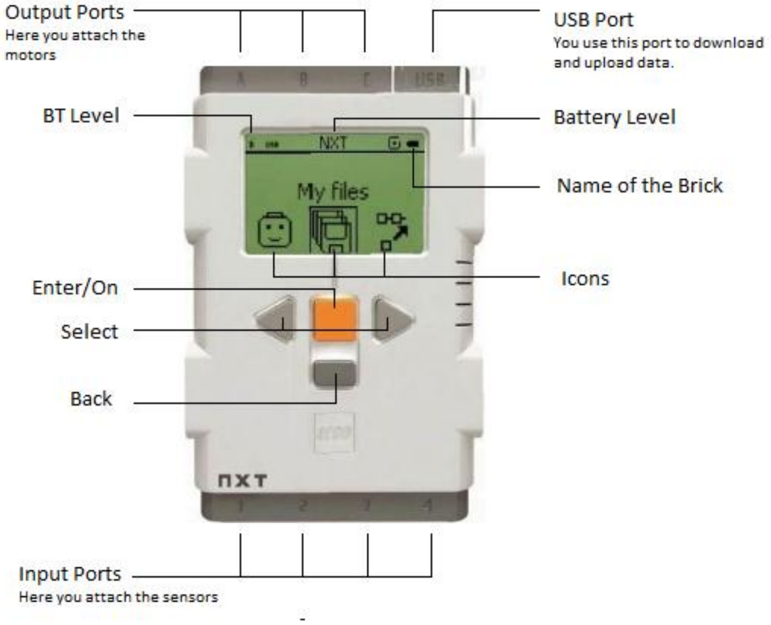


Figure 2: NXT Brick

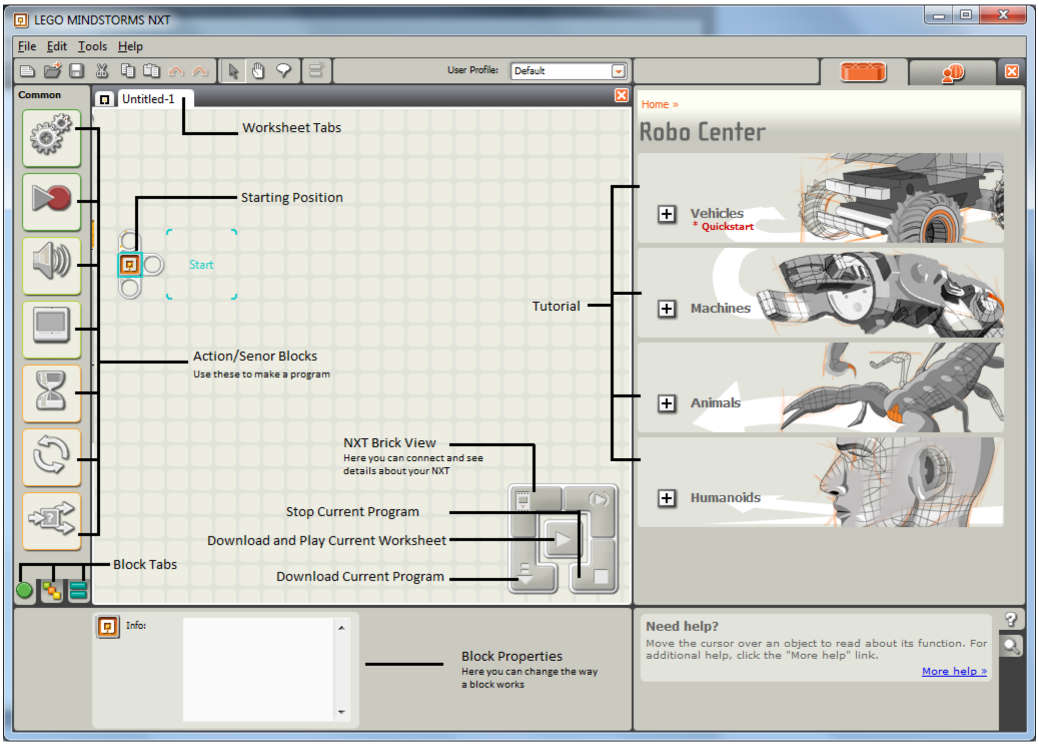


Figure 1: NXT-G Environment

Day 1 Tutorials

Tutorial 1 - Movement

The NXT comes with a set of motors (Figure 3) that plug into ports **A**, **B** and **C**. These motors can be used to make your robot move around the room or to give it moving parts e.g. an arm. The motors allow you to have precision control over their speed, rotation and direction.

Figure 3 – NXT Motor

Task 1 - Straight Line Movement

Here we will make our NXT move in a straight line.

1. Drag the 'Loop' block into the work area and connect it to the 'Starting Point' (Figure 4).

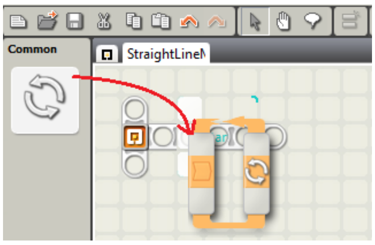
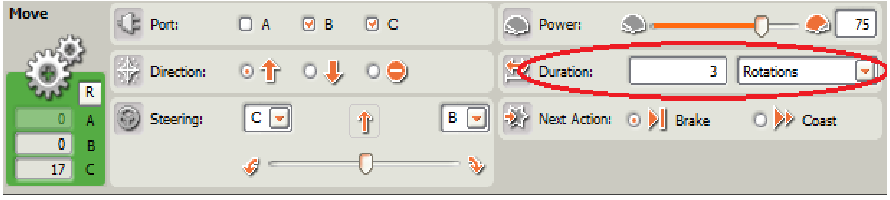


Figure 4 – Adding Loop Block

Drag the 'Move' block into the 'Loop' block (Figure 5). Make sure the 'Move' block settings match Figure 6.

If everything went well your NXT should move forward three rotations or the wheels, stop, and then repeat the task continuously.



Port: □ A ☑ B ☑ C

Direction: ● ↑ ○ ↓ ○ ⊖

Steering: C B

Power: 75

Duration: 3 Rotations

Next Action: ● Brake ○ Coast

Figure 5 – Move Block Properties

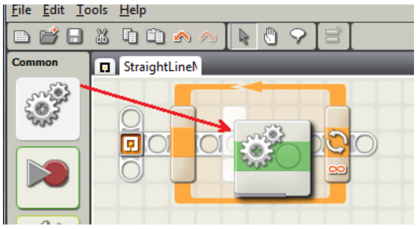


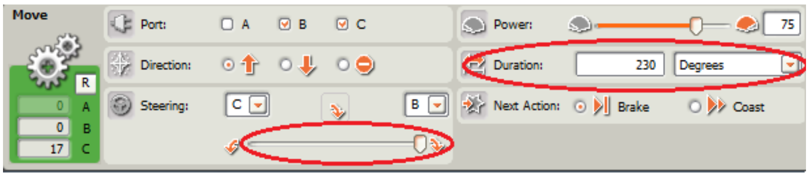
Figure 5 – Adding Move Block

Task 2 – Move in a Square

Now that we can make our NXT move forward we can look at how to make it turn.

Following on from 'Task 1':

2. Add another 'Move' block into the loop (Figure 7).
3. You will need to change the settings of one of the 'Move' blocks to match Figure 8.



Port: □ A ☑ B ☑ C

Direction: ● ↑ ○ ↓ ○ ⊖

Steering: C B

Power: 75

Duration: 230 Degrees

Next Action: ● Brake ○ Coast

Figure 8 – Move Block Properties

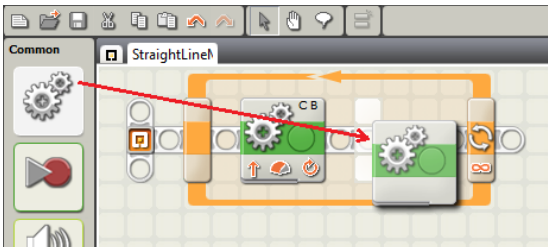
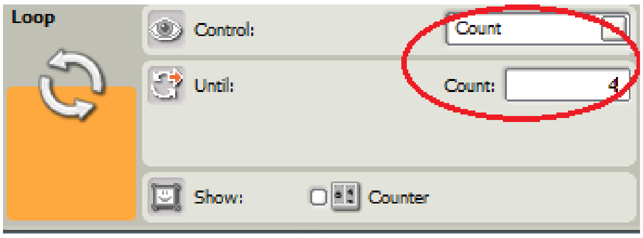


Figure 7 – Adding Move Block

4. You will need to make this loop repeat four times, this can be done by changing the Control type to ‘Count’ and the setting the Until Count to ‘4’ (Figure 9).



Control: Count

Until: Count: 4

Show: □ Counter

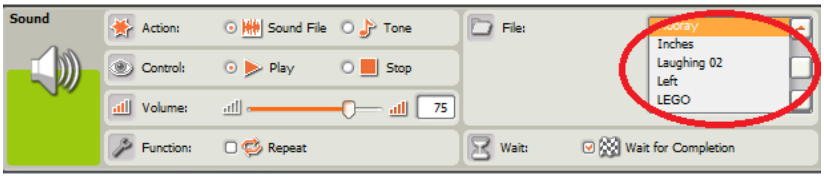
Figure 6 – Loop Block Properties

Task 2 – Adding Sound

The NXT has an inbuilt speaker that allows you to play tones, songs and sounds.

Following on from ‘Task 2’

5. Drag the ‘Sound’ block into the work area after the loop. The program will play the sound when is has finished (Figure 11). You can change what sound is played in the properties (Figure 10).



Action: ● Sound File ○ Tone

Control: ● Play ○ Stop

Volume: 75

Function: □ Repeat

File: Hooray
Inches
Laughing 02
Left
LEGO

Wait: ☑ Wait for Completion

Figure 8 – Sound Block Properties

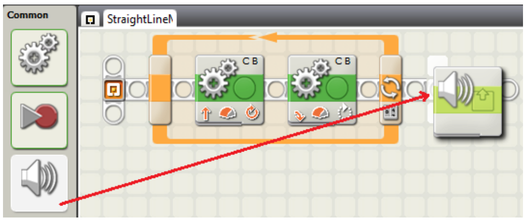


Figure 7 – Adding Sound Block

Task 3 - Figure of 8

Using the skills you have just learnt see how many of the following figure of 8s you can make the NXT move in.

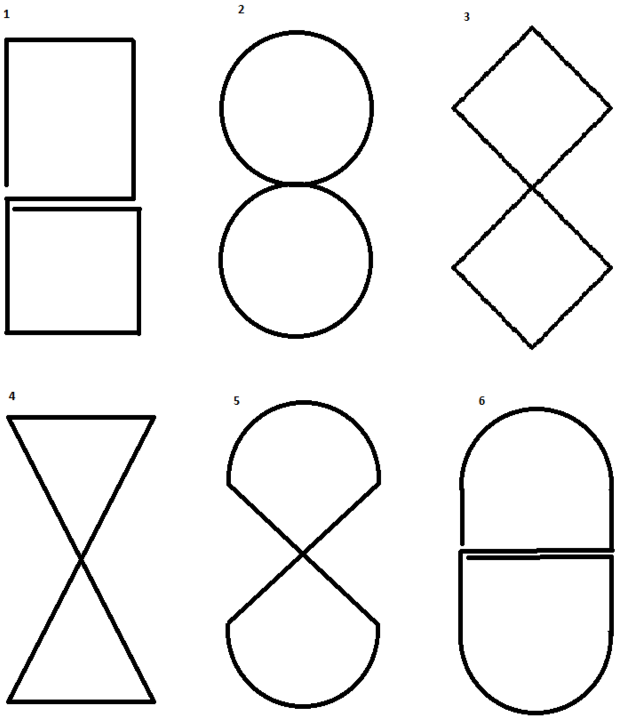


Figure 9 – Figure of 8s

Tutorial 2 - Sound Sensor

The NXT comes with a ‘Sound’ sensor (Figure 13), this is basically a microphone. This sensor is capable of hearing all sounds the human ear can hear but also some sounds outside of the audible spectrum.

The sensor returns a value between 0 – 100%, 0% being silent and 100% being the loudest.

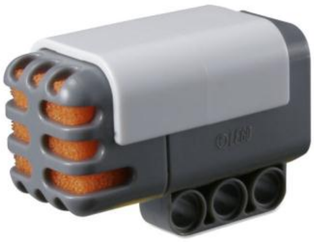


Figure 10 – Sound Sensor

Build Instructions

Note: Make sure you connect the sensor to Port 2!

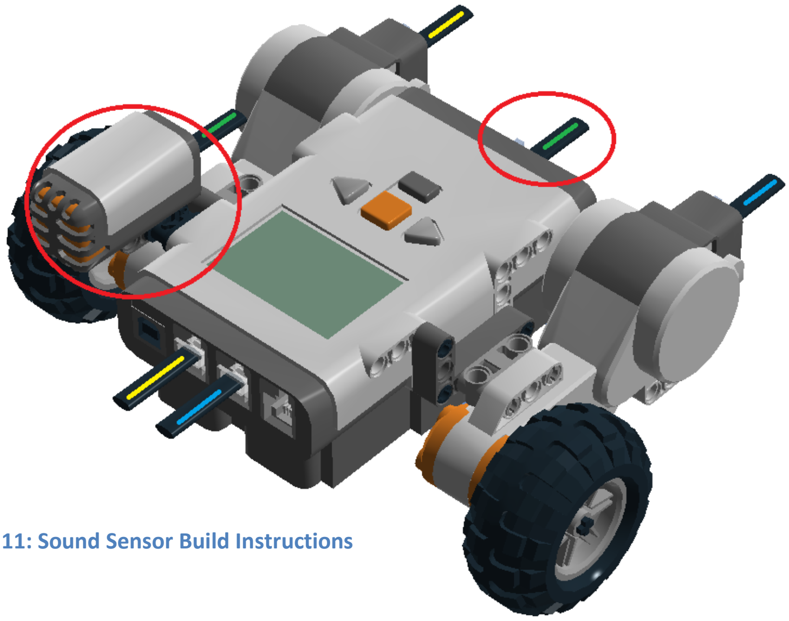


Figure 11: Sound Sensor Build Instructions

Task 1 – Hearing Sounds

Here we are going to give the NXT the ability to hear sounds.

1. Start a new project. You will need to add a wait cue; this will halt the program until the condition is fulfilled (in this case when a sound is heard). Open the ‘Wait’ menu and drag the ‘Sound’ block into the work sheet (Figure 15).

2. Add a ‘Loop’ block after the Sound block from ‘Step 1’ (Figure 16).

Make sure the settings match Figure 17.

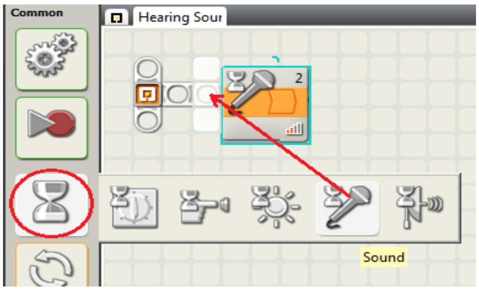
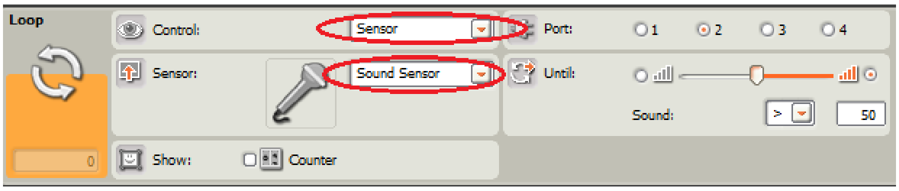


Figure 13 – Adding Sound Block



Loop

Control: Sensor

Port: ○ 1 ● 2 ○ 3 ○ 4

Sensor: Sound Sensor

Until:

Sound: > 50

Show: □ Counter

Figure 14 – Sound Block Properties

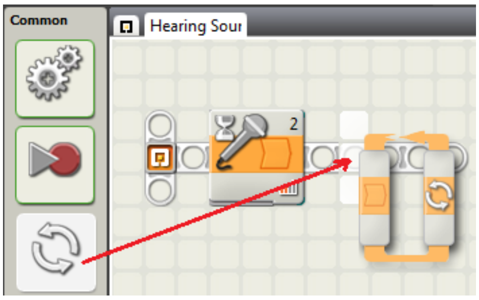


Figure 136 – Adding Loop Block

3. Add a sound wait cue inside the ‘Loop’ block. This will make sure the motor only moves when there is no sound (Figure 18).

4. Add a ‘Move’ block after the second ‘Sound’ block (Figure 19). Make sure the settings match Figure 20.

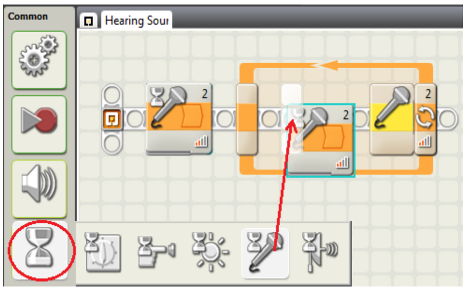
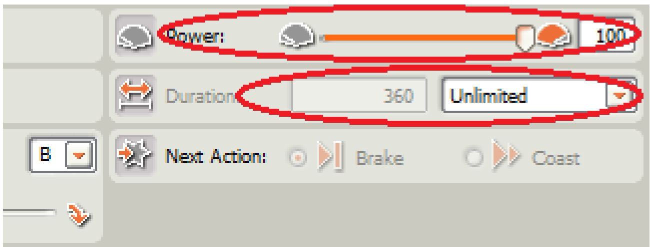


Figure 15 – Wait -> Sound Block



Power: 100

Duration: 360 Unlimited

Next Action: ○ Brake ○ Coast

B

Figure 16 – Move Block Properties

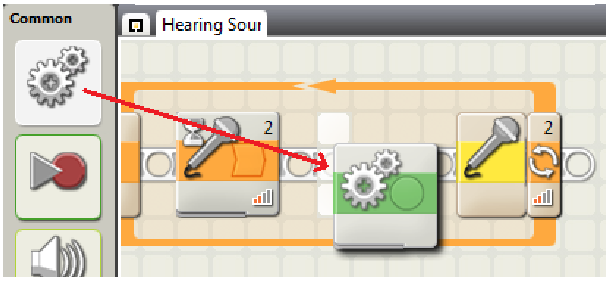


Figure 19 - Adding Move Block

If everything went well your NXT should wait to hear a sound, and then continuously move forward until it hears another sound.

Tutorial 3 - Touch Sensor

The NXT comes with a Touch sensor (Figure 21) for you to make use of. The sensor has two different states, pressed and released. This could be used to detect when the NXT has bumped into an obstacle.

Figure 17 – Touch Sensor

Build Instructions

Note: Make sure you connect the sensor to Port 1!

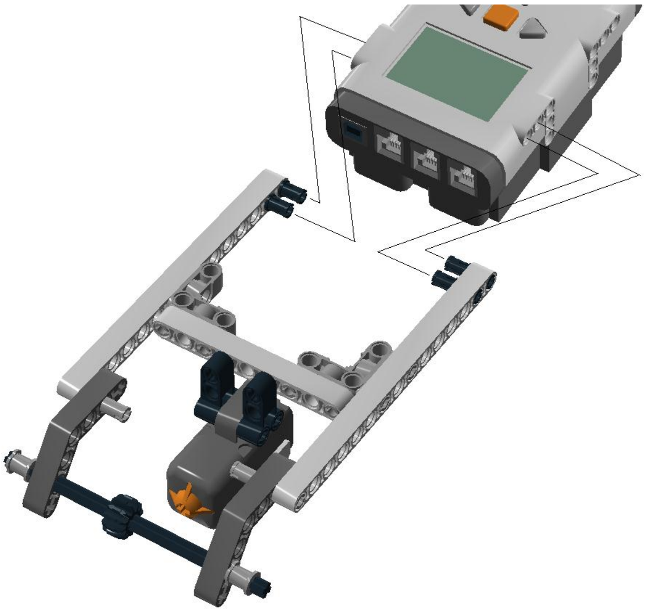


Figure 18: Touch Sensor Build instructions

Task 1 - Change Direction When You Hit Obstacle

1. Start a new project. Add the 'Loop' block to the start position in the work sheet (Figure 23).

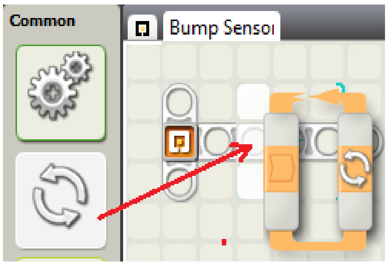


Figure 19 – Adding Loop Block

2. Drag a 'Switch' block into the 'Loop' block placed in Step 1 (Figure 24).

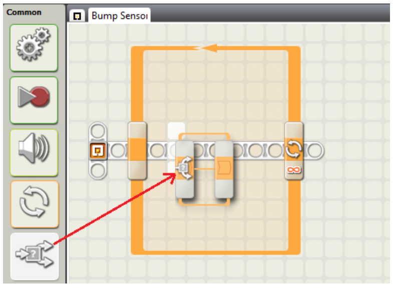
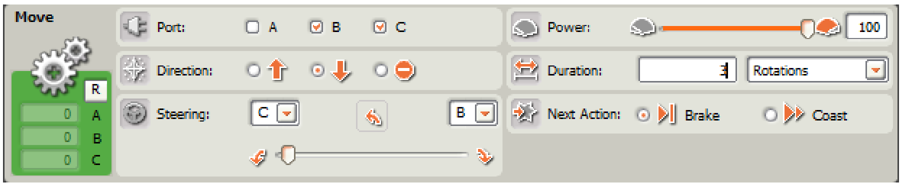


Figure 24 – Adding Switch Block

3. Add a 'Move' block to the upper section of the 'Switch' block (Figure 26), this instruction will only be performed when the switch is pressed. Make sure the setting match Figure 25.



Move

Port: ☐ A ☑ B ☑ C

Direction: ○ ↑ ● ↓ ○ ⊖

Steering: C B

Power: 100

Duration: 3 Rotations

Next Action: ● Brake ○ Coast

R

0 A

0 B

0 C

Figure 205 – Move Block Properties

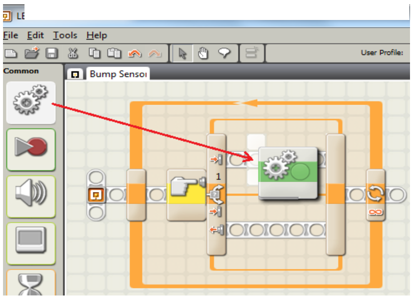


Figure 21 - Adding Move Block

4. Add a 'Move block to the lower section of the 'Switch' block (Figure 28), this instruction will run when the button is not pressed. Make sure the settings match Figure 29.

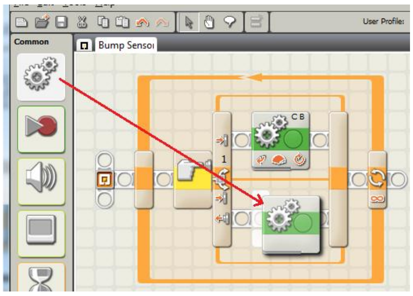
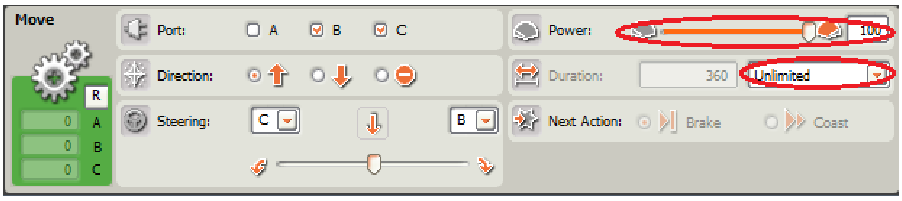


Figure 23 - Adding Move Block



Move

Port: ☐ A ☑ B ☑ C

Direction: ● ↑ ○ ↓ ○ ⊖

Steering: C B

Power: 100

Duration: 360 Unlimited

Next Action: ● Brake ○ Coast

R

0 A

0 B

0 C

Figure 239 – Move Block Properties

If everything when well, your NXT should reverse and turn when it hits and obstacle, then repeat the task continuously.

Tutorial 4 - Ultrasonic Sensor

The NXT comes with an Ultrasonic sensor (Figure 29); this gives your NXT the ability to see. The sensor will be able to detect object in front of it for up to 255cm.

Figure 24 – Ultrasonic Sensor

Build Instructions

Note: Make sure you connect the sensor to Port 4!

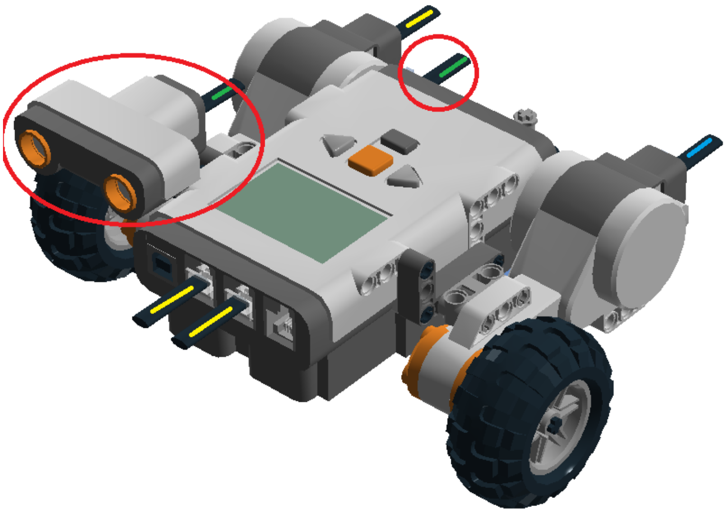


Figure 25: Sonar Sensor Build Instructions

Task 1 – Obstacle Avoidance

This program will make the robot stop when it detects an object in front of it. .

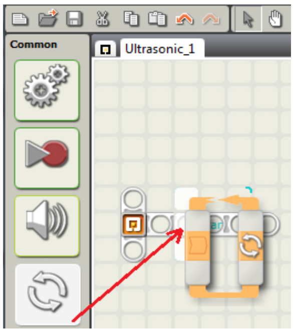


Figure 26 – Adding Loop Block

1. Start a new project. Add a ‘Loop’ block to the start position in the work sheet (Figure 31).
2. Add a ‘Switch’ block into the ‘Loop’ block placed in Step 1 (Figure 32). You will need to change the sensor type to ‘Ultrasonic Sensor’ (Figure 33).

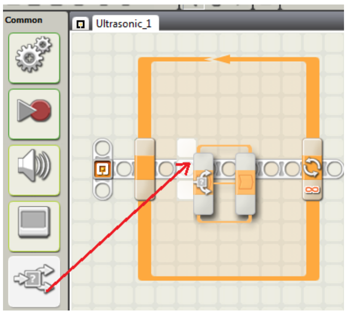
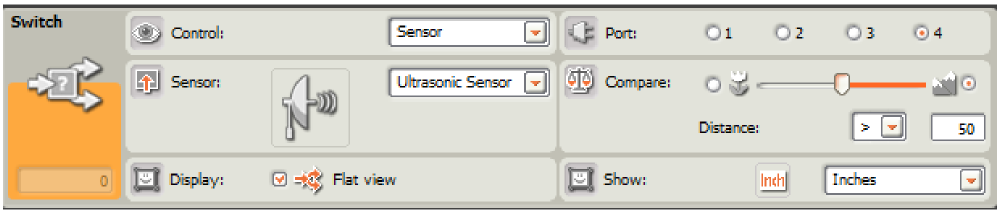


Figure 32 – Adding Switch Block



Switch

Control: Sensor

Sensor: Ultrasonic Sensor

Display: ☑ Flat view

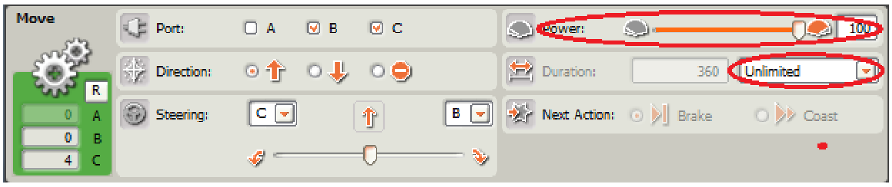
Port: ○ 1 ○ 2 ○ 3 ● 4

Compare: Distance: > 50

Show: Inches

Figure 27 – Switch Block Properties

3. Add a ‘Move’ block to the upper section of the ‘Switch’ block (Figure 34). Make sure the settings match Figure 25.



Move

Port: □ A ☑ B ☑ C

Direction: ● ↑ ○ ↓ ○

Steering: C B

Power: 100

Duration: 360 Unlimited

Next Action: ○ Brake ○ Coast

R A 0 B 0 C 4

Figure 29 – Move Block Properties

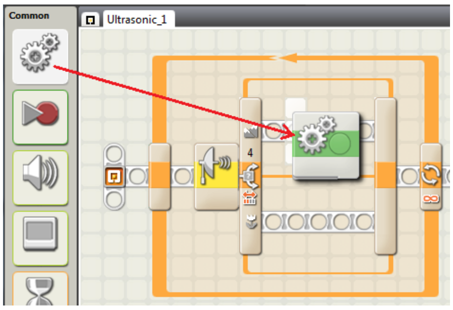
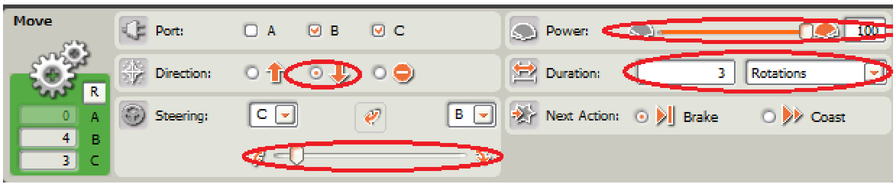


Figure 294 - Adding Move Block

4. Add a ‘Move’ block to the lower section of the ‘Switch’ block (Figure 36). Make sure the setting match Figure 37.



Move

Port: □ A ☑ B ☑ C

Direction: ○ ↑ ● ↓ ○

Steering: C B

Power: 100

Duration: 3 Rotations

Next Action: ● Brake ○ Coast

R A 0 B 4 C 3

Figure 31 - Move Block Properties

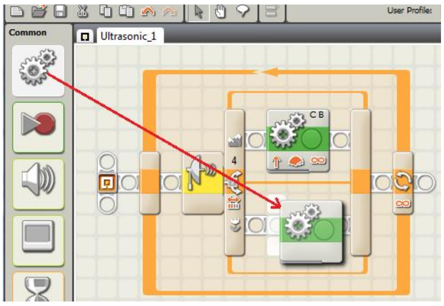


Figure 316 - Adding Move Block

Task 2 – Random Direction

This program will be built on Task 1, however, this time your NXT will pick a random direction when it detects an object, then move.

1. Repeat steps 1-3 from Task 1

4. Drag a ‘Random’ block into the upper section of the Ultrasonic Switch block placed in Task 1 – Step 2 (Figure 38).

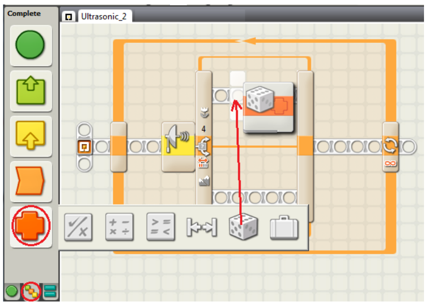
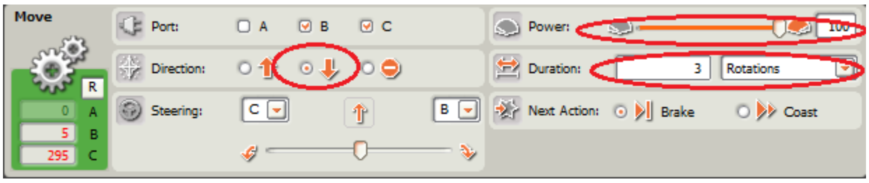


Figure 32 – Adding Random Block

Drag a ‘Move’ block into the upper section of the ‘Switch’ block, after the ‘Random’ block (Figure 39). Make sure the settings match Figure 40.



Move

Port: □ A ☑ B ☑ C

Direction: ○ ↑ ● ↓ ○

Steering: C B

Power: 100

Duration: 3 Rotations

Next Action: ● Brake ○ Coast

R A 0 B 5 C 295

Figure 33 – Move Block Properties

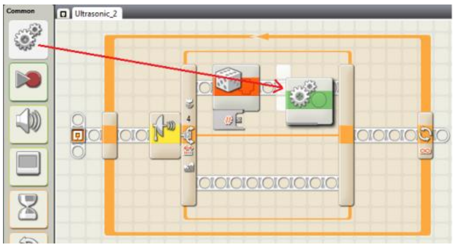


Figure 39 - Adding Move Block

5. Connect the ‘Number’ of the ‘Random’ block to the ‘Steering’ plug of the ‘Move’ block. To connect plugs just simply click one, and then click another. To disconnect a Wire from a plug, click on the plug (Figure 41).

6. Drag a ‘Move’ block to the lower section of the ‘Switch’ block (Figure 42). Make sure the settings match Figure 43.

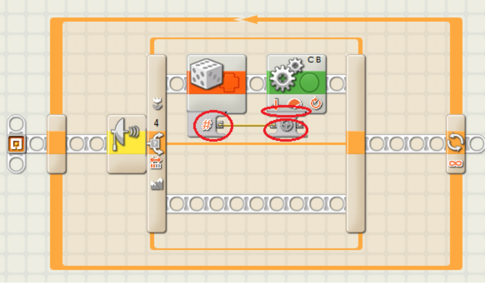
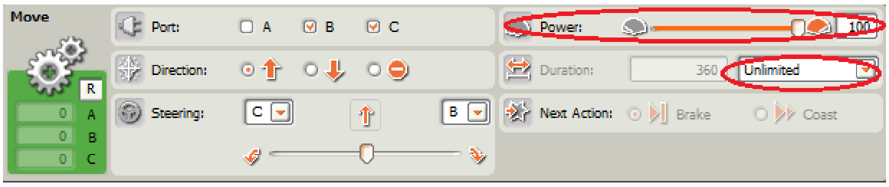


Figure 34 – Connecting Plugs



Move

Port: □ A ☑ B ☑ C

Direction: ● ○ ○

Steering: C B

Power: 100

Duration: 360 Unlimited

Next Action: ● Brake ○ Coast

Figure 36 – Move Block Properties

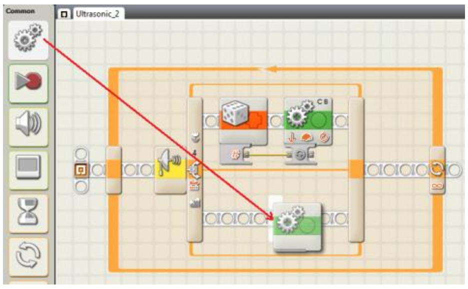


Figure 362 - Adding Move Block

If everything went well, your NXT should drive round the room until it detects an obstacle. Once and obstacle is detected the NXT will reverse and turn in a random direction then repeat the task continuously.

Figure 37 – Light Sensor

Tutorial 5 - Light Sensor

The NXT comes with a Light sensor (Figure 44). This sensor is can distinguish between light and dark. The sensor is also capable of generating light; this could be used to detect colour shade in poor lighting conditions.

Build Instructions

Note: Make sure you connect the sensor to Port 3!

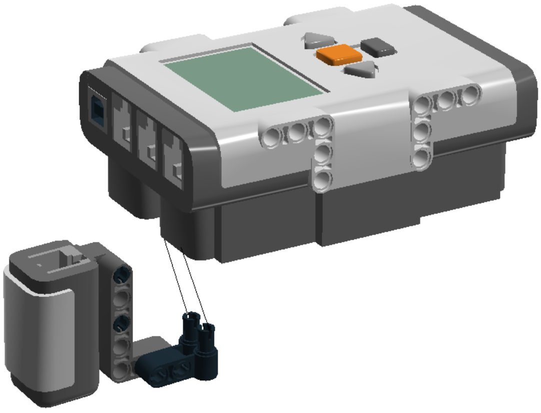


Figure 38: Light Sensor Build Instructions

Task 1 – Line Following

1. Start a new project. Add a 'Loop' block to the start of your program (Figure 46).

2. Place a 'Switch' block inside the 'Loop' placed in Step 1 (Figure 47). Make sure the settings match Figure 48.

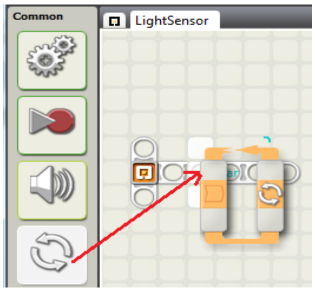
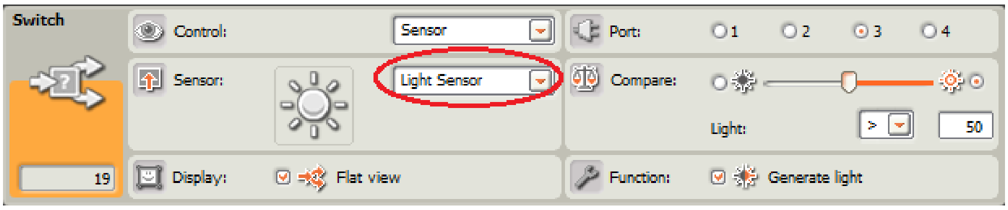


Figure 39 – Adding Loop Block



Switch

19

Control: Sensor

Sensor: Light Sensor

Display: ☑ Flat view

Port: ○ 1 ○ 2 ● 3 ○ 4

Compare:

Light: > 50

Function: ☑ Generate light

Figure 408 – Switch Block Properties

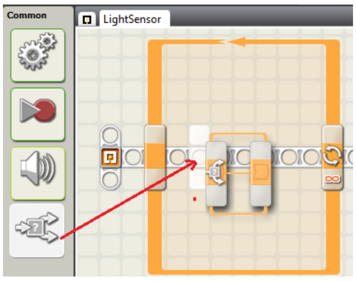


Figure 47 – Adding Switch Block

3. Place two 'Move' blocks in upper section of the 'Switch' block. One 'Move' block will need to stop Motor B, and the other to make Motor C go forward (Figure 49).

4. Place two 'Move' blocks in lower section of the 'Switch' block. One 'Move' block will need to stop Motor C, and the other to make Motor B go forward (Figure 50).

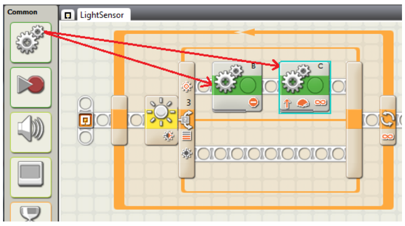


Figure 49 – Adding Two Move Blocks

If everything went well your NXT will be able to follow the line supplied in your pack.

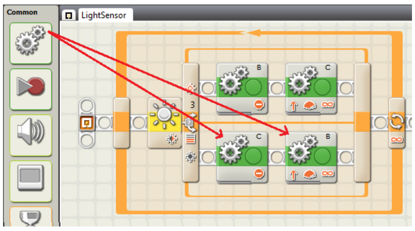


Figure 50 – Adding Two Move Blocks

Congratulations!

You have finished all the tutorials on how to the use the NXT-G software to program your NXT.

You are now ready to move onto programming the NXT in C#.