Lego NXT

Lego NXT robots are ideal for delivering Control Technology in KS2 and KS3. They allow students to create, test, and improve a sequence of instructions. Robots can be programmed to perform simple tasks, such as navigating a maze, and include environmental sensors for more complex operations.

Beginners can find using the robots difficult at first. This help sheet will provide a basic introduction and will show you where to go for more detailed advice.

The activities in this help sheet are designed for KS2 pupils working in groups of 3 or 4 KS2. We have found that large groups (more than 4) prevent some students gaining hands-on experience.

The Robot

The Lego NXT Robots come pre-assembled in a standard design. Although other designs are possible we ask schools not to disassemble robots as we only have a limited time to prepare and check equipment as it travels in between schools.

Cables and Connections

For the robot to work properly, the motors and the sensors need to be attached to the control unit correctly.

The motors must be plugged into the top of the control unit, with the wheel motors going into plugs 'B' and 'C'.

Note that the USB plug to connect your robot to your computer is also in the top of the control unit.

Sensors must be plugged into the bottom of the control unit. Sound sensor into '2', light sensor into '3', and Ultrasonic (distance) sensor into '4'.
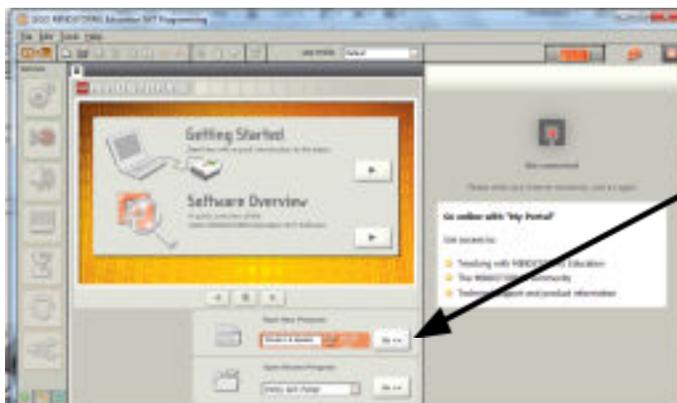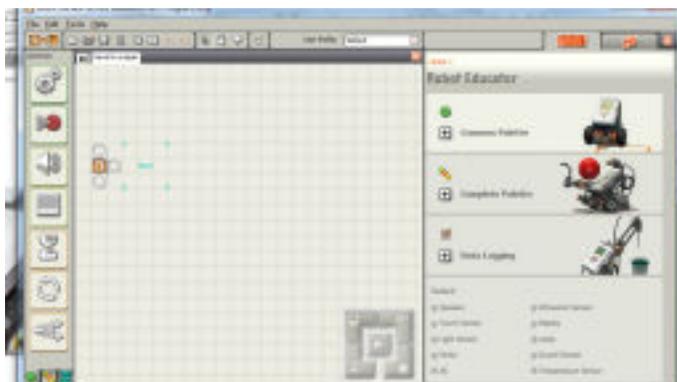
Building and Running Programs

To use the robot you must first design a set of instructions with the Lego Software on the laptop provided.

To open the software click

**'Start' > 'All Programs' > 'LEGO MINDSTROM Edu NXT' > 'NXT Programming'**

Begin by clicking 'File' > 'New' as you would in any other program, or type in a file name here and 'Press Go'.

This will open a new workspace where you can design instructions for your robot.

Travelling in a square

This is an excellent introductory activity. It is simple and introduces all the key concepts of programming, especially trial and improvement. Begin by marking out a 1 metre square on the floor. You can use masking tape, post-it notes, plastic cups, or anything else you have to hand.
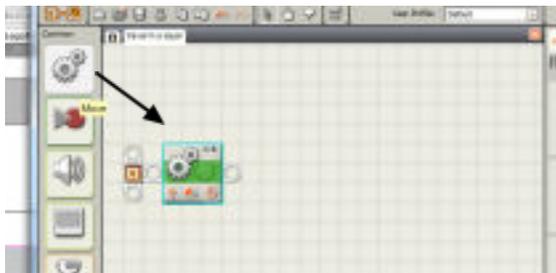
Begin by dragging a 'Move' block and connecting it to the programming line. This block will instruct the robot to go in a straight line along the edge of your square.

Once in place, click on the 'Move' block and look at the bottom of the screen. A new window will appear with lots of options.
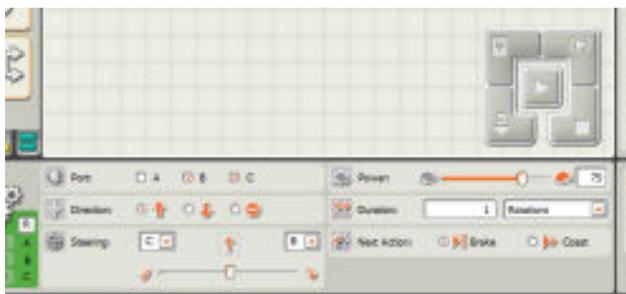
This window controls:

- The direction and steering
- The speed
- The duration (distance to travel)
- Finishing action ('break' to stop on the spot, or 'coast' until friction slows you down)

We want the robot to travel in a straight line so leave all the settings as they are, except for the duration.
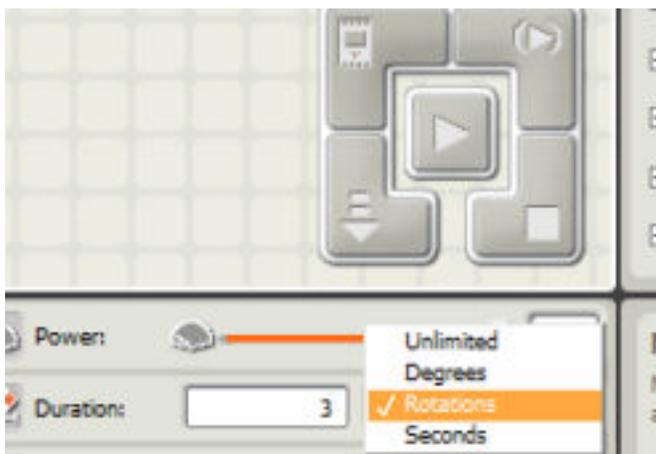

We do not have the option to enter an exact distance. We can only select:

- Unlimited (move forever)
- Degrees (move small distances: turn the wheels of the robot less than one full rotation. E.g. 90 represents a quarter turn of the wheels)
- Rotations (Move one full rotation of the wheels. The circumference of each wheel is c.15cm so 2 rotations = c.30cm)
- Seconds (move for an allotted period of time)

Select 'Rotations' or 'Seconds' and guesstimate the number required to travel the length of the square.

Now test your estimate. Save your program: click **'File' > 'Save'**.

Now plug the robot into the computer via the USB and press the 'Download' button.

Once the download is complete, unplug the USB cable and place the robot at the start of your square. Now you must run the downloaded file. Use the orange button on the front of the robot to select **'My Files' > 'Software files' > 'YourFileName' > 'Run'**. The robot should move straight forward. Check the distance it has travelled. If it is incorrect go back to your computer, adjust the duration (remember you can use decimals to improve the accuracy), save, and download the file once again. Keep testing and improving until it is accurate.
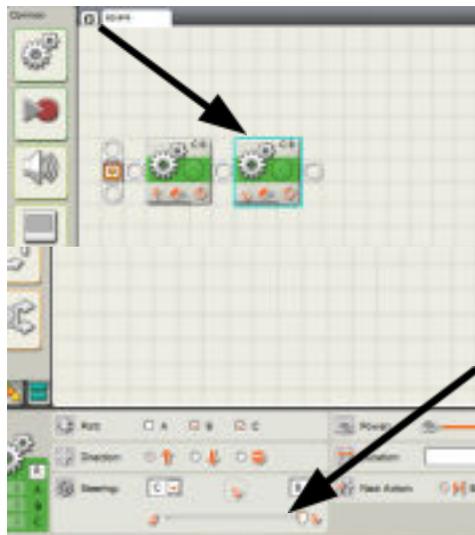
Once you have completed the straight, you need to add another 'Move' block to turn the corner. Add a second 'Move' block to your program line.

Click on the new 'Move' block and check the settings at the bottom of the screen.

Drag the steering control to the left or the right (depending on which way you want to turn).

Now guesstimate the duration of the turn, then download and test using the trial and improvement method until you have a precise 90 degree turn.

You should now have the two basic blocks for your square: one forward and one turn. To travel in a full square, you must replicate these blocks 4 times. You can do this is two ways:

1. Copy and Paste:
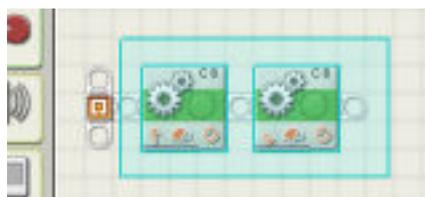
Select both blocks by clicking and dragging a box around them.

Click **'Edit' > 'Copy' and then 'Edit' > 'Paste'** (as you would in any

other program). Repeat and then connect the blocks to your programming line to make a list of 8 block, 4 straights and for 4 turns, making a complete square.
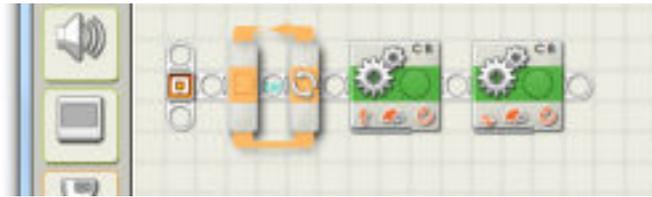
2. Use a 'Loop'

Drag and drop a 'Loop' block to the beginning of the programming line.

Drag and drop the two move blocks so they are inside the Loop.
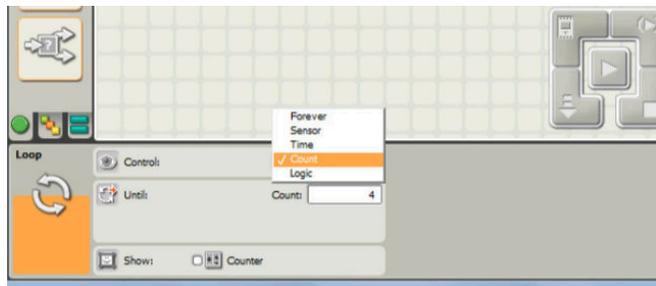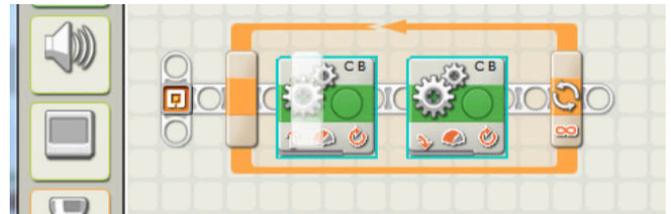
Click on the 'Loop' block and adjust the settings at the bottom of the screen. In the 'Control' drop down menu select 'Count'

(this defines the number of times the loop repeats). In the 'Until' box enter '4'. This will repeat the forward - turn sequence 4 times producing a square.

In your classroom

Once you have completed the square you can use the Lego Help to investigate other progamming blocks. This can be particularly useful when learning about sensors. You may also want to try some more activities. Here are a few suggestions:

- Once you have completed a square, place just two markers on the floor and try to navigate 'figure-of-8'. This can be tricky so think carefully before you begin programming. It can be helpful to walk out the path and imagine how you might program each stage (e.g. straight for 30cm, turn 45 degrees left, etc...).
- Ask each group to create an obstacle course and then swap groups around to see which course is the most challenging. - Use the sensors to create an alarm system to stop people stealing your robot. Use the 'Wait' block and distance and light to do this.