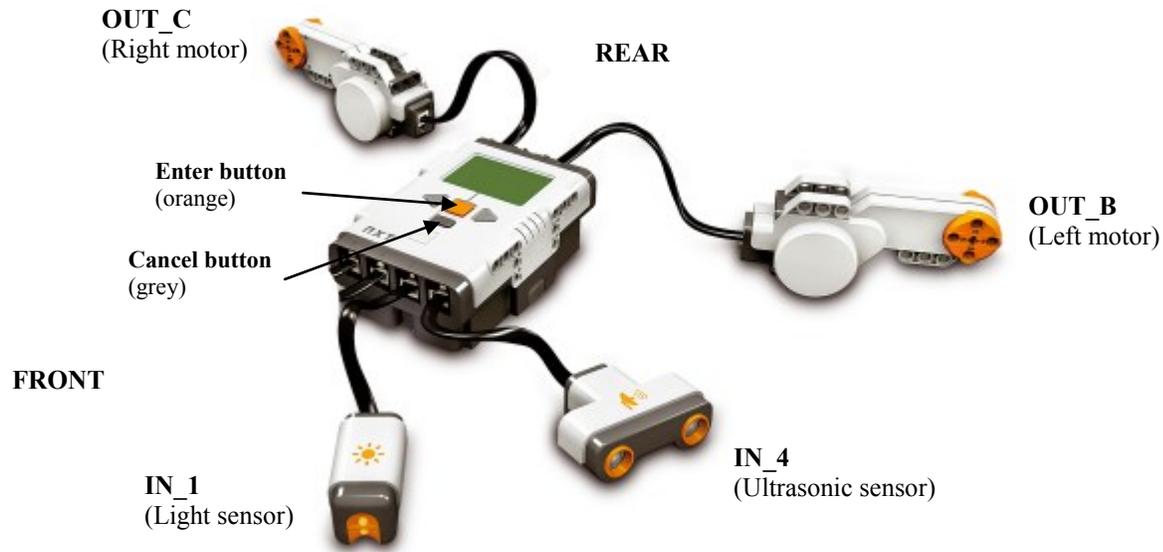


Lego NXT Reference Chart



Common Commands

- OnFwd(OUT_A, 80);** Switches on motor A at 80 percent power, forwards. The motor will run until we stop it (using **Off**).
- OnRev(OUT_A, 75);** Switches on motor A at 75 percent power, backwards. The motor will run until we stop it (using **Off**).
- Wait(1000);** Waits for 1000 milliseconds (1 second). During this time, the robot will do nothing – no more instructions will be processed. However, if the motors are running already they will continue to run.
- Off(OUT_A);** Switches off motor A.
- // Anything....** Use two slashes to write a comment. The computer will ignore this but it can help you understand your own program when you come back to it later!
- repeat (5) { }** Repeats the instructions between the curly brackets 5 exactly times.
- while (true) { }** Repeats the instructions between the curly brackets over and over again, forever.

Notice: every line ends with a semi-colon; this is essential; Also, the command are case-sensitive!

Reading Sensors

The **ultrasonic sensor** measures distance in centimetres. To check how far away an object is, use the **SensorUS** command and an **if** command, like this:

```
if (SensorUS(IN_4) < 30) {  
    Off(OUT_B);  
    Off(OUT_C);  
}
```

This checks if the ultrasonic sensor connected to IN_4 (input 4) is less than 30 centimetres away an object. If it is the two Off commands will run and stop the vehicle. If it is more than 30 centimetres from an obstacle, nothing will be done. An alternative is to use **if-else**, like this:

```
if (SensorUS(IN_4) < 10) {  
    Off(OUT_B);  
    Off(OUT_C);  
} else {  
    OnFwd(OUT_B, 100);  
    OnFwd(OUT_C, 100);  
}
```

Now, if the sensor is less than 10cm from an object, the vehicle will stop. Otherwise, motors A and B will go on at full speed (100%)

Getting started

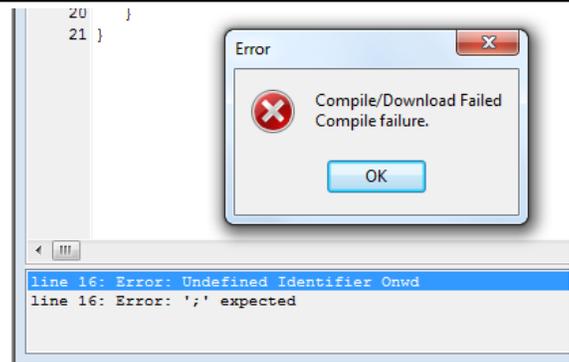
This code below is the basis for most programs. You should type it into the Brixcc software and then add your own program in the indicated space.

```
task main() {  
  
    // Set up the sound sensor attached to input 4  
    SetSensorLowspeed(IN_4);  
  
    // We are going to repeat this forever....  
    while (true) {  
  
        ← Type your instructions here!  
  
    }  
}
```

Ready?

Before you can use your program you need to compile it.

1. Go to the **Compile** menu and select **Compile**.
2. If you see an error message like the one here, it means you have made a typing mistake somewhere. The computer might tell you which line to look at. Look at your program closely!
3. If you don't receive a message, your program compiled fine. Now you can download it to the robot.



Downloading the program to the Robot

[Make sure you have compiled your program first!](#)

1. Save your program with your name as the filename
2. Connect the robot using the USB cable
3. Go to **Tools** -> **Find Brick**.
4. Make sure **Port** says **usb** and **Brick type** says **NXT**.
5. Click OK
6. Go to **Compile** -> **Download**.
7. If successful, the robot will give you a nice little beep
8. On the Lego robot, press the **Run button** to go into **My Files**, then again to go into **Software files**. Find your program name.
9. Put the robot down and press the **Run button** two more times to run the program
10. Press the **Cancel button** to stop the robot

Herein there be dragons...

Detecting colours....

```
SetSensorLight(IN_1);           // Set up the colour sensor  
SetSensorColorFull(IN_1);  
ResetSensor(IN_1);  
  
if (Sensor(IN_1) == INPUT_BLUECOLOR) {  
    // Do something here  
}
```

The names you can use to represent colours are:

INPUT_BLACKCOLOR	INPUT_BLUECOLOR
INPUT_GREENCOLOR	INPUT_YELLOWCOLOR
INPUT_REDCOLOR	INPUT_WHITECOLOR