Appendix1: LegoMindstorms NXT (hardware and software)

Author: Michele Moro

The NXT brick is the brain of a MINDSTORMS® robot. It is a computer-controlled LEGO® brick that can read MINDSTORMS's sensors and command MINDSTORM's motors. The NXT brick is much powerful and flexible with respect to previous LEGO MINDSTORMS programmable bricks (see Table 1 – Technical specifications).

Now, the NXT brick has two microcontrollers one for running the main applications and one for controlling the motors. Both microcontrollers are fitted with FLASH memory so they will not loose your programs nor the firmware if one removes the batteries (differently from the previous RCX brick). The NXT brick can be programmed via USB or Bluetooth. In fact, the LEGO MINDSTORMS NXT includes a wireless Bluetooth node that enables the NXT brick to communicate with other Bluetooth devices. Actually, both of these connections are bidirectional so one can also upload data from the robot to the computer.

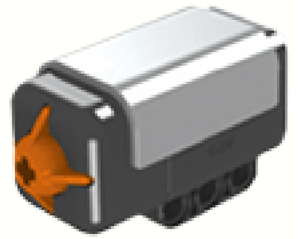
The NXT brick has a LCD display large enough to also play simple video-games. Also, the sound output has been improved, while the old RCX brick could utter only sine wave tones, the new NXT brick, in addition to generate these tones, can also play any pre-recorded audio file in PCM and ADPCM formats.

Table 1-Technical specifications

Main CPU:	32-bit ARM7 microcontroller, 256 Kbytes FLASH, 64 Kbytes RAM
Motor controller:	8-bit AVR microcontroller, 4 Kbytes FLASH, 512 Byte RAM
Input /Output:	Bluetooth wireless communication (Bluetooth Class II V2.0 compliant 460.8 Kbit/s)
	USB full speed port (12 Mbit/s)
	4 input ports for attaching sensors - Ports 1, 2, 3 and 4. The ports uses the 6-wire cable digital platform (One port includes a IEC 61158 Type 4/EN 50 170 compliant expansion port for future use)
	3 output ports for attaching motors - Ports A, B and C with 6-wire cable digital platform
Display:	100 x 64 pixels LCD graphical display
Sound:	Loudspeaker - 8 kHz sound quality. Sound channel with 8-bit resolution and 2-16 KHz sample rate.
Power source:	6 AA batteries

TOUCH SENSOR

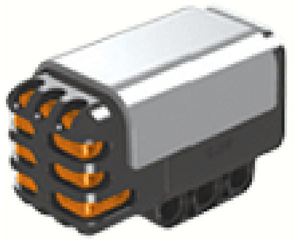
The Touch Sensor gives the robot the sense of touch. The Touch Sensor detects when it is being pressed by something and when it is released again. It has a cross-axle hole at the front of the sensor to attach cross-axle for expanding the sensor range or use it as animals use wiskers.



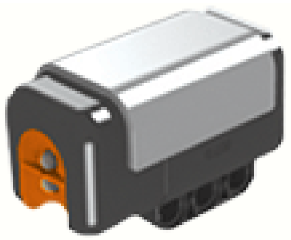
SOUND SENSOR

The Sound Sensor is a microphone that picks-up sounds from the environment. It can work both in decibels [dB] mode and in adjusted decibel [dBA] mode. In **dB mode** the sensor is detecting standard [unadjusted] decibels, all sounds are measured with equal sensitivity. Thus, these sounds may include some that are too high or too low for the human ear to hear. In **dBA mode** the sensor is detecting adjusted decibels, the sensitivity of the sensor is adapted to the sensitivity of the human ear and is limited at the range of frequencies that the humans can ear.

The Sound Sensor can measure sound pressure levels up to 90 dB – about the level of a lawnmower. Sound pressure levels are extremely complicated, so the Sound Sensor readings on the MINDSTORMS NXT are displayed in percent [%]. The lower the percent the quieter the sound. For example:



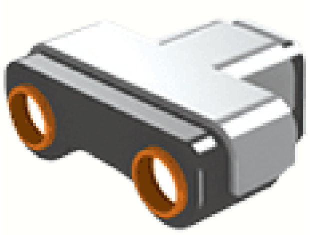
- 4-5% is like a silent living room
- 5-10% would be someone talking some distance away
- 10-30% is normal conversation close to the sensor or music played at a normal level
- 30-100% are people shouting or music being played at a high volume



LIGHT SENSOR

The Light Sensor is one of the two sensors that give your robot vision [The Ultrasonic Sensor is the other]. This sensor is made of a light emitter (a LED) and a light meter (a photocell). It has to be noticed that the new NXT sensor has a small plastic barrier between the emitter and the sensor solving the problem of emitter light going directly into the sensor. The Light Sensor can measure the light intensity reflected by colored surfaces at the short range. Alternatively, the emitter can be disabled and the sensor can measure the ambient light, so the robot can distinguish between light and dark or measure the light level in different locations of the room.

ULTRASONIC SENSOR



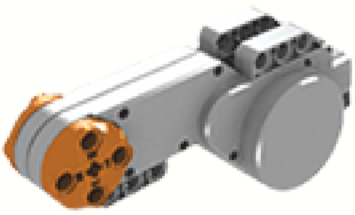
The Ultrasonic Sensor is one of the two sensors that give your robot vision [The Light Sensor is the other]. The Ultrasonic Sensor enables your robot to detect objects and to measure the distance to them. You can also use it to make your robot avoid obstacles and detect movement.

The Ultrasonic Sensor measures distance in centimeters and in inches. It is able to measure distances from 0 to 255 centimeters with a precision of +/- 3 cm.

The Ultrasonic Sensor uses the same scientific principle as bats: it measures distance by calculating the time it takes for an ultrasound wave to hit an object and return (this is called time of flight). Again like in the light sensor, there is an emitter (i.e., an ultrasound emitter) and a receiver (i.e., a microphone) picking up the echo waves reflected by the obstacles in front of the sensor.

Large sized objects with hard surfaces return the best readings. Objects made of soft fabric or that are curved [like a ball] or are very thin or small can be difficult for the sensor to detect. This sensor can give false reading also with hard surfaces oriented at small angles with respect to the sensor, because this might cause the ultrasound wave to be specularly reflected in a different direction and no echo is coming back to be picked up by the sensor receiver. Another situation that can disturb the sensor readings is the presence of other ultrasonic sensors (maybe mounted on other robots). One sensor could pick up the ultrasound wave emitted by a second sensor, instead of the echo of its own emitted waves and thus calculating an incorrect time of flight and then calculating an incorrect distance to the obstacles. Fortunately, the ultrasonic sensors have a special operation mode that help solving such a situation.

The three Servo Motors give the robot the ability to move. The NXT motors are larger and less efficient than the RXC motors, but have several improvements. They have a gear train that provides less speed and more torque, so most of the times complex gear-down trains are no longer needed. They provide an integrated



encoder or Rotation Sensor to read the exact motor position. The Rotation Sensor measures motor rotations in degrees [accuracy of +/- one degree]. One rotation is equal to 360 degrees. The Rotation Sensors combined with the new capability of the NXT firmware to control motors in pairs to synchronise them enables precise control of robot's movements. For instance, using the Move block in the LEGO MINDSTORMS NXT-G software to program the motors, the two motors will automatically synchronise, so that the robot will move in a straight line. Moreover, the new motors have two different connection possibilities either the central hole to fix cross-axles of different lengths or four off-center holes to fix the standard Technic connetors.

Third party sensors.

The four input ports for attaching sensors (namely Port 1, 2, 3, and 4) are compatible with the industrial standard I^2C bus. This allow everyone to develop a new sensor that comply with the I^2C standard and to read it through the NXT sensor ports. Companies like HiTechnic and Mindsensors produce several sensors that use the I^2C bus.

MINDSTORM NXT-G Software

LEGO® MINDSTORMS® NXT-G software enables to simply program the NXT even without knowing any programming language. NXT-G is a what is called a visual programming environment, this means one will use a graphical interface to develop code (the G in NXT-G stands for graphical). With NXT-G one can upload his/her programs to the NXT via USB or Bluetooth connectivity. The intuitive drag and drop software, powered by National Instruments LabVIEW, comes with building instructions and programming guides to easily begin constructing and programming with MINDSTORMS NXT. Users create a program by selecting and dragging functional blocks to a canvas area from the palettes. In the palettes, one can find blocks implementing all basic functionalities of the NXT kit. In other words, each block is a programming method. The user/programmer can set different values for the parameters of the method by setting the block's sliders, drop-dwon menu, radio button, text boxes, etc.

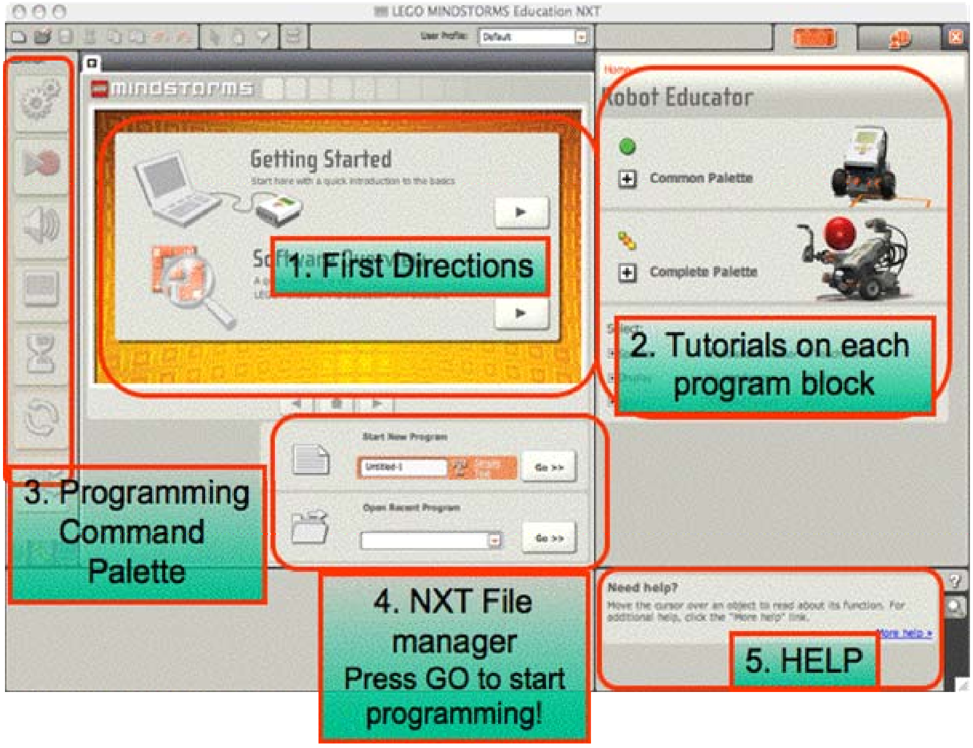

LEGO MINDSTORMS Education NXT
User Profile: Default
MINDSTORMS
Getting Started
Start here with a quick introduction to the basics
Robot Educator
Common Palette
Complete Palette
Start New Program
Untitled-1
Go >>
Open Recent Program
Go >>
Need help?
Move the cursor over an object to read about its function. For additional help, click the "More help" link.
More help »
1. First Directions
2. Tutorials on each program block
3. Programming Command Palette
4. NXT File manager Press GO to start programming!
5. HELP

Fig. 1 MINDSTORMS NXT-G Software

The Figure 1 depicts the first window appearing when opening NXT-G. One can notice five major areas, as highlighted in Figure 1. Area 1: the first directions for novices on how to get start with NXT-G. Area 2: a tutorial on each programming block. Area 3: the area of the programming palettes containing the programming blocks. Area 4: the file manager to open an existing file or to give a name of your new project. Area 5: the detailed help in which one can find all he/she needs. As soon as a new block is input for the new project a new window appear. Here, one can find again on the right the help and tutorials section.

In Fig. 2, we highlighted three areas. Area 1: the programming canvas where program blocks have to be dropped and ordered to obtain the desired robot behaviour. Area 2: the NXT brick controller, and Area 3: the palette selector used to switch between the three block palettes.

Once, a NXT-G program is created it has to be transferred to the robot and it has to be executed by the NXT brick. This process is governed using the controller (see Fig. 3). The controller has five buttons. Top left button connects NXT-G with the NXT brick and once connected information status of the NXT brick will be displayed (e.g., battery level, internal memory usage, etc.). Bottom left button downloads the open program from the computer to the NXT brick. The central button downloads and runs the open program on the NXT brick. The top right button downloads and runs a different program to be selected on the NXT brick. The bottom right button stops the current program running on the NXT brick.

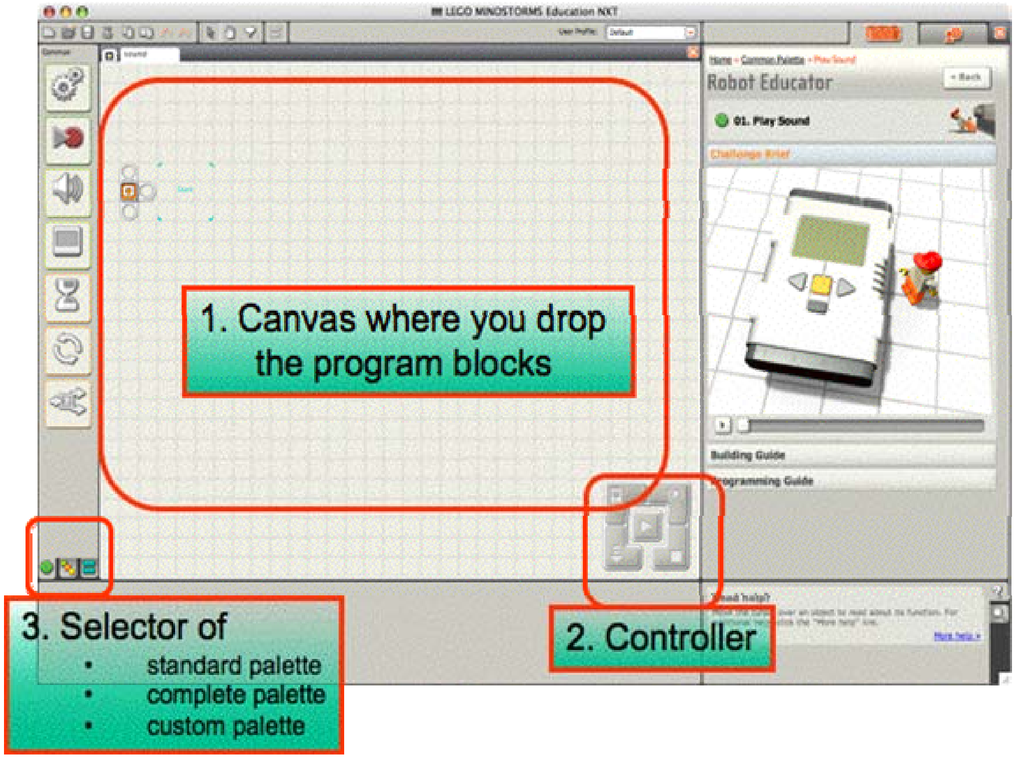


Fig. 2 MINDSTORMS NXT-G Software

The programming environment is very simple for novice, but is more powerful that one could expect. You can create custom blocks that contains lot of basic blocks

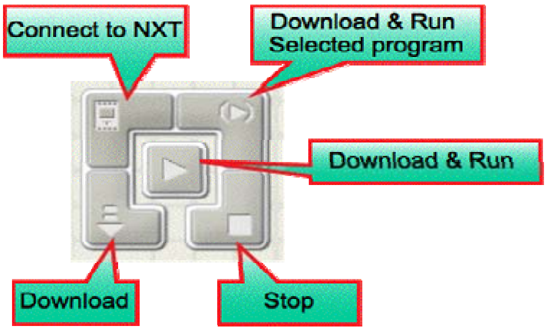
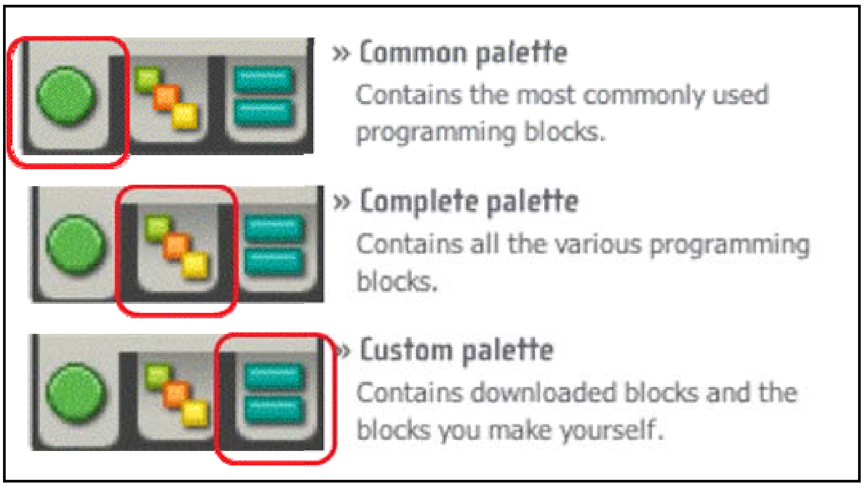


Fig. 3 MINDSTORMS NXT-G Software

(i.e. lot of code) that connected together realise a specific task. This custom block can be saved to the custom palette and used later as a part of a larger program. This makes easy to fit a lot of code into the limited graphic canvas, There are three palettes (see Fig. 4): the common palette, the complete palette, and the custom palette. In the custom palette one can import blocks created bu other users or for new devices as they become available for the NXT.



» Common palette

Contains the most commonly used programming blocks.

» Complete palette

Contains all the various programming blocks.

» Custom palette

Contains downloaded blocks and the blocks you make yourself.

Fig. 4 MINDSTORMS NXT-G Software

Although the iconic programming environment of the graphical programming interface is in general very easy to learn for users with no programming experience, sometimes it might be misleading. First of all, it is not always clear what a block or a diagram of block is doing. Second, sometimes the iconic language primitives might be much more ambiguous than a textual programming language primitive. In the end, when the experience of the user grows over a certain threshold and thus his/her programs grow over a certain complexity and size it is hard to manage the code just with the graphical interface both because of space and because of NXT-G software instability. Once one reach a certain degree of confidence and skill in controlling the NXT kit, he/she should move to a textual programming language like Java or C. For using Java, LeJOS NXJ is the choice. For using C, NXC is the choice.