

# LME Software Block Quick Reference

## Common Palette



### Move Block

Use this block to set your robot to go forwards or backwards in a straight line or to turn by following a curve. Define how far your robot will go by using the Duration property.

#### Block Variables

1. The letters at the top right corner of the block show which of your NXT's ports will be controlled.
2. The icon at the bottom left shows which direction your robot will go.
3. The icon at bottom center shows the power level. Your robot's speed may also be affected by other conditions, like the surface it is moving over or whether it is moving up or down a slope.
4. The icon at bottom right shows whether you have set the Duration property to unlimited, degrees, rotations, or seconds.



### Record/Play Block

This block lets you record an action that you act out with your robot. It does this by remembering how many times the motors turn on your robot and then replaying the correct number of turns when you select "run" on your Record/Play block and run your program.

For example, you can have the Record/Play block record the motion as you roll a wheeled robot forward and then turn it sharply to the left. Or you might build a robotic arm and record the motion as it shakes hands with you. When you run these recorded motions using a Record/Play block, the wheeled robot will go forward and turn, and the robotic arm will shake hands, in both cases without your help.

To record an action, first choose a name for the action and an estimated time for how long it will take to complete. In the first example above of the wheeled robot, you might call the file "forward and turn" and set the time to 10 seconds.

You would then download the block. When you're ready, run the program and act out the motion you want to record. When the time is up (10 seconds in this case), the motion you acted out will be saved.

Finally, change the block's action property from "record" to "play," type the name you gave to the recorded sequence, download the Record/Play block again and run the program. It will repeat (as closely as possible) the acted out motion without your help.

Note: Your recorded movement may not be exactly the same as the movement you acted out.

#### Block Variables

The icon at bottom right shows whether the block is set to record or play.



## Sound Block

You can use this block to play a sound file or a single tone. To compose a melody of tones, arrange several sound blocks in a row with each set to play different tones.

If you select “Wait for completion” in the configuration panel, the sound file or tone will finish playing before the program moves on to the next block. With this item unchecked, the sound file or tone will continue to play while the next block of your program proceeds.

Choosing “Repeat” will cause a sound file to play again and again.

### Block Variables

1. The icon at bottom left shows whether the block will play a sound file or a tone.
2. The icon at bottom center shows whether the block will start or stop playing a sound.
3. The icon at bottom right shows the block’s volume. An icon with four orange bars is set to the loudest volume.
4. You can drag data wires from other blocks to this block’s data hub that will affect the Lamp block’s properties.



## Display Block

Use this block to display an image, write some text, or draw a shape on the NXT’s display screen.

By positioning several Display blocks in a row you can create more complex drawings by adding additional images, text and shapes with each new Display block.

If you want to erase the contents of the screen before beginning, check the “Clear” checkbox.

If you would like to return to the default LEGO MINDSTORMS icon, select Reset in the pull-down menu.

The screen measures 100 pixels wide by 64 pixels high.

### Block Variables

1. The icon at bottom right shows whether the block is set to display an image, some text, or a drawing; or whether it will just reset the display to the default icon.
2. You can change values dynamically by connecting data wires to this block’s data hub.



## Wait Block

This block lets your robot sense its environment for a certain condition before it continues. Use the slider or type in a value to set a trigger point so that the program continues when sensor values are below or above it.

1. The number or letter at the top right corner of the Wait block shows the port the block is monitoring. The configuration panel, which is described below, will let you change ports if necessary.
2. If you have chosen to wait for a light, sound, or ultrasonic sensor, this icon indicates at what level the trigger point is set; the more colored bars displayed, the higher the trigger point. If you have chosen to wait for a touch sensor, the icon will display the touch sensor condition (bumped, pressed, or released) that will trigger the block and allow the program to move on.

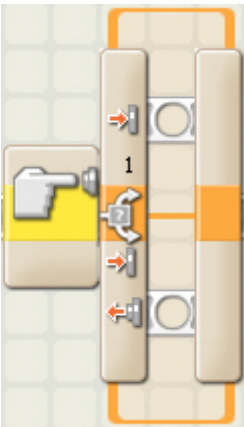


## Loop Block

Use this block to repeat sequences of code. Set the condition that will end the loop: elapsed time, the number of repetitions, a logic signal or a sensor. You can also set a loop to go on forever.

### Block Variables

1. When a Loop block's action property is set to "Forever," an infinity symbol ( $\infty$ ) is displayed at the bottom of the trailing portion of the block.
2. If the "Show Counter" checkbox is selected in the configuration panel, a plug will appear that will allow you to use the number of completed loops as an input elsewhere in your program (if you connect a data wire from the plug to another block's data hub). You can also use current count to control the loop itself. (See the Count setting in the Configuring the Loop Block section below).
3. If you choose for a sensor to control the loop, the trailing portion of the Loop block will expand in size revealing an icon for the chosen sensor. For example, if you choose a Light Sensor to control the loop, a Light Sensor icon will be displayed in the expanded portion of the block. Also, any relevant information about the chosen control property will be displayed at the bottom of the block.



## Switch Block

Use this block to choose between two sequences of code. For example, when configured with a touch sensor, the switch block might run one series of blocks when the sensor is pressed and another when the touch sensor is not.

### Block Variables

1. The icon at the beginning of the switch block indicates the sensor or other condition that will cause the block to choose between the two rows of programming blocks. In this case the current status of a touch sensor will cause the program to switch.

- The upper blocks will run if the touch sensor is pressed.
- The lower block will run if the touch sensor is not being pressed.

# Complete Palette

## Action Blocks



### Motor Block

This block allows for precise control of one motor's speed. You can “ramp up” to a set speed or “ramp down” to a stop. By deselecting “Wait for Completion,” your program can move on to the next block once the Motor block has started a motor. # The letter at the top right corner of the block shows which NXT port is set to control a motor.

#### Block Variables

1. This icon shows which direction the motor will go.
2. This icon shows the power level.
3. This icon shows whether you have set the Duration property to unlimited, degrees, rotations, or seconds.
4. You can change values dynamically by connecting data wires to this block's data hub. (See the Data Hub section below for more information.)



### Sound Block

You can use this block to play a sound file or a single tone. To compose a melody of tones, arrange several sound blocks in a row with each set to play different tones.

If you select “Wait for completion” in the configuration panel, the sound file or tone will finish playing before the program moves on to the next block. With this item unchecked, the sound file or tone will continue to play while the next block of your program proceeds.

Choosing “Repeat” will cause a sound file to play again and again.

#### Block Variables

1. The icon at bottom left shows whether the block will play a sound file or a tone.
2. The icon at bottom center shows whether the block will start or stop playing a sound.
3. The icon at bottom right shows the block's volume. An icon with four orange bars is set to the loudest volume.
4. You can drag data wires from other blocks to this block's data hub that will affect the Lamp block's properties.



### Display Block

Use this block to display an image, write some text, or draw a shape on the NXT's display screen.

By positioning several Display blocks in a row you can create more complex drawings by adding additional images, text and shapes with each new Display block.

If you want to erase the contents of the screen before beginning, check the “Clear” checkbox.

If you would like to return to the default LEGO MINDSTORMS icon, select Reset in the pull-down menu.

The screen measures 100 pixels wide by 64 pixels high.

### Block Variables

1. The icon at bottom right shows whether the block is set to display an image, some text, or a drawing; or whether it will just reset the display to the default icon.
2. You can change values dynamically by connecting data wires to this block's data hub.



### Send Message Block

This block allows you to send a wireless message to another NXT.

Before wireless messages can be sent or received, the participating NXTs must be set up for wireless communication.

To send a wireless message (after all NXTs have been configured with connection numbers), first select the connection number of the target NXT. Then choose the message format (Text, Number, or Logic). Input the message by typing in some text or a number, or choosing a logical condition (true or false). Finally, choose the mailbox number (where the message will be stored) at the target NXT. The message format and the message itself can also be specified by using input data wires.

Note: see the Receive Message block for information on receiving a wireless message.



### Motor\* Block

This block allows you to control one motor's speed.

\*This motor requires an adaptor cable to connect to the NXT.

### Block Variables

1. The letter at the top right corner of the block shows which NXT port is set to control a motor.
2. The icon at bottom left shows which direction the motor will go.
3. The icon at bottom center shows the power level.
4. You can change values dynamically by connecting data wires to this block's data hub.



### Lamp\* Block

Use the Lamp\* block to control a lamp. You will need one Lamp\* block to turn a lamp on and a second Lamp\* block to turn it off.

\*This lamp requires an adaptor cable to connect to the NXT.

### Block Variables

1. The letters at the top right corner of the Lamp\* block show which NXT ports will be controlled by the block. If your lamp is plugged into port A, one of the letters displayed must be A. The configuration panel, which is described below, will let you change ports if necessary.
2. The icon at bottom center shows whether the block is set to turn a lamp on or off.
3. The icon at bottom right shows the lamp's intensity. An icon with four white bars is set to full intensity.
4. You can drag data wires from other blocks to this block's data hub that will affect the Lamp block's properties.

# Sensor Blocks



## Touch Sensor Block

This block checks a touch sensor's condition at a specific point in the program. It sends out its finding as a logic signal (true or false) through a data wire. If the sensor has been triggered, the block will send out a "true" signal; if it has not been triggered, the block will send a "false" signal.

You must drag an output data wire from this block's data hub to another block for any information to be sent.

### Block Variables

Use the radio buttons to decide which action will produce the "true" signal.

1. The number shows which of your NXT's ports are connected to the touch sensor. You can change this number in the configuration panel if you need to.
2. This icon will show which action will send a "true" signal (Bumped, Pressed, or Released).
3. The block's data hub will open automatically when the block is placed in the work area. A data wire must be dragged from the block's output plug to another block's data hub.



## Sound Sensor Block

This block is a sound detector. Using data wires, it can send out the current sound value and a logic signal (true/false) based on whether the current sound value falls above or below a trigger point.

Specify the trigger point by using the slider or by typing a value into the input box. To specify the range (above or below the trigger point) that will generate the "true" signal, use the radio buttons or the pull-down menu. The "true" portion of the range will be in color; the "false" portion will be gray.

The default setting for the sound sensor block is for sound levels above 50% to generate a "true" signal. The radio button to the right of the range is selected and the slider is set at 50. To switch the "true" portions of the range (setting values below 50% as "true"), you would select the left radio button.

The feedback box in the configuration panel shows the current sound value. (To receive feedback, make sure that the sensor is connected to the chosen port and that communication has been established with the NXT.)

You must drag at least one output data wire from this block's data hub to another block for any information to be sent.

### Block Variables

1. The number shows which of your NXT's ports is connected to the sound sensor. You can change this number in the configuration panel if you need to.
2. The icon at bottom right indicates at what level the trigger point is set. The more colored bars displayed, the higher the trigger point.
3. The block's data hub will open automatically when the block is attached to a sequence beam. At least one data wire must be dragged from the block's output plug to another block's data hub.



## Light Sensor Block

This sensor detects ambient (surrounding) light. Using data wires, it can send out the current light value and a logic signal (true/false) based on whether the current light value falls above or below a trigger point.

Specify the trigger point by using the slider or by typing a number into the input box. To specify the range (above or below the trigger point) that will generate the “true” signal, use the radio buttons or the pull-down menu. The “true” portion of the range will be in color; the “false” portion will be gray.

The default setting for the light sensor block is for light levels above 50% to generate a “true” signal. The radio button to the right of the range is selected and the slider is set at 50. To switch the “true” portions of the range (setting values below 50% as “true”), you would select the left radio button.

The feedback box in the configuration panel shows the current light value. (To receive feedback, make sure that the light sensor is connected to the chosen port and that communication has been established with the NXT.)

You must drag at least one output data wire from this block’s data hub to another block for any information to be sent.

### Block Variables

1. The number shows which of your NXT’s ports are connected to the light sensor. You can change this number in the configuration panel if you need to.
2. This icon indicates at what level the trigger point is set. The more colored bars displayed, the higher the trigger point.
3. This icon is displayed if “Generated Light” is turned on.
4. The block’s data hub will open automatically when the block is placed in the work area. At least one data wire must be dragged from the block’s output plug to another block’s data hub.



## Ultrasonic Sensor Block

This block can detect objects at a maximum range of about 250 cm (or 100 inches). Using data wires, it can send out the current ultrasonic reading and a logic signal (true/false) based on whether the current ultrasonic reading falls above or below a trigger point.

Specify the trigger point by using the slider or by typing a value into the input box. To specify the portion of the ultrasonic sensor’s range (above or below the trigger point) that will generate the “true” signal, use the radio buttons or the pull-down menu. The “true” portion of the range will be in color; the “false” portion will be gray.

The default setting for the ultrasonic sensor block is for objects detected at less than 50 cm to generate a “true” signal. The radio button to the left of the range is selected and the slider is set at 50. To switch the “true” portion of the range (setting readings at more than 50 cm as “true”), you would select the right radio button.

The feedback box in the configuration panel shows the current ultrasonic reading. (To receive feedback, make sure that the sensor is connected to the chosen port and that communication has been established with the NXT.)

You must drag at least one output data wire from this block's data hub to another block for any information to be sent.

### Block Variables

1. The number at top right shows which of your NXT's ports are connected to the ultrasonic sensor. You can change this number in the configuration panel if you need to.
2. The icon at bottom right shows whether the ultrasonic sensor is set to detect objects nearby or farther away. The more colored bars, the farther away the detection.
3. The block's data hub will open automatically when the block is placed in the work area. At least one data wire must be dragged from the block's output plug to another block's data hub. The trigger point can be set dynamically by connecting an input data wire.



### NXT Buttons Block

This block outputs a "true" signal through a data wire when one of the NXT buttons is activated. You will have to select the button and the action that will send the "true" signal.

### Block Variables

1. The icon at the bottom of the block indicates which NXT button has been selected.
2. The block's data hub will automatically open when the block is placed on the sequence beam. You will need to drag a data wire from the NXT Buttons block to another block that can accept a logic (true/false) signal.



### Rotation Sensor Block

This block counts the number of degrees (one full rotation is 360 degrees) or full rotations that your motor turns. Through data wires, this block can send out the current number of degrees or rotations, and a logic signal (true/false) based on whether the number of degrees or rotations falls above or below a trigger point.

Specify the trigger point by typing a number into the input box. To specify the range (above or below the trigger point) that will generate the "true" signal, use the pull-down menu to choose greater than or less than [the trigger point].

The default setting for the Rotation Sensor block is for degrees more than 180 to generate a "true" signal.

The feedback box will count how many degrees or rotations your motors turn. As a default, the box is set to display degrees. With the Duration property set to Rotation, the current count will be displayed in full rotations. Press the reset button to return it to zero. (To receive feedback, make sure that the motor is connected to the chosen port and that communication has been established with the NXT.)

### Block Variables

1. The letter at top right shows which port the Rotation Sensor block will be monitoring. You can change to another port in the configuration panel if you need to.
2. The icon at bottom right will show whether the block is counting the number of rotations (the solid circular arrow) or degrees (the dashed circular arrow).
3. # The block's data hub will open automatically allowing you to drag out a logic (true/false) data wire or a number (#) data wire that will transmit the Rotation Sensor's current count.





## Timer Block

When your program starts, the three built-in timers in the NXT will automatically start counting. With this block you can choose to either read a timer's current value or cause a timer to start counting again from zero.

Through data wires, this block can send out the current timer value and a logic signal (true/false) based on whether the timer value falls above or below a trigger point.

Specify the trigger point by typing a number into the input box. To specify the range (above or below the trigger point) that will generate the "true" signal, use the pull-down menu to choose greater than or less than [the trigger point].

For the Timer block to function, you must drag a data wire from the block's data hub and attach it to another block's data hub. You can drag two kinds of output data wires from the Timer block's data hub: a logic data wire that will send out a true/false signal, and a number (#) data wire that will send out the timer's current value.

You can also supply the trigger value dynamically by connecting an input data wire to the Timer block's data hub.

### Block Variables

1. This number shows which of the three timers in the NXT you will be monitoring. You can change to another timer in the configuration panel if you need to.
2. The block's data hub will open automatically when the block is attached to the sequence beam, allowing you to drag out a logic (true/false) data wire or a number (#) data wire that will transmit the timer's current time value. You can also supply the trigger value dynamically by connecting an input data wire to the Timer block's data hub.

Note: you must drag at least one output data wire from the Timer block's data hub to another block for any information to be sent.



## Receive Message Block

To receive a wireless message, set the message type and mailbox number to match those of the sending NXT. This block's output can be the incoming message itself or a "true/false" logic value (if you are comparing the incoming message to a test message).

Before wireless messages can be sent or received, the participating NXTs must be set up for wireless communication.

To receive a wireless message (after all NXTs have been configured with connection numbers), first indicate the format of the incoming message (Text, Number, or Logic).

If you want to test that a certain message has been received (and have the Receive Message block send out a "true" signal), type the test text or number in the input box, or check either true or false. If the incoming message, matches the test message, a "true" signal will be sent from the block through the "v/x" plug.

To finish configuring the block, choose the mailbox number where the incoming message will be stored.

Note: See the Send Message block for information on sending a wireless message.



## Touch\* Sensor Block

This block checks a touch sensor's condition at a specific point in the program. It sends out its finding as a logic signal (true or false) through a data wire. If the sensor has been triggered, the block will send out a "true" signal; if it has not been triggered, the block will send a "false" signal.

You must drag an output data wire from this block's data hub to another block for any information to be sent.

Use the radio buttons to decide which action will produce the "true" signal.

\*This sensor requires an adaptor cable to connect to the NXT.

### Block Variables

1. The number in the upper right shows which of your NXT's ports are connected to the touch sensor. You can change this number in the configuration panel if you need to.
2. The icon at the bottom right will show which action will send a "true" signal (Bumped, Pressed, or Released).
3. The block's data hub will open automatically when the block is attached to the sequence beam. A data wire must be dragged from the block's output plug to another block's data hub.



## Rotation\* Sensor Block

This block counts the number of ticks (16 to a rotation) that your motor turns. Through data wires, this block can send out the current number of ticks and a logic signal (true/false) based on whether the number of ticks falls above or below a trigger point

\* This sensor requires an adaptor cable to connect to the NXT.

Specify the trigger point by typing a value into the input box. To specify the range (above or below the trigger point) that will generate the "true" signal, use the pull-down menu to choose greater than or less than [the trigger point].

The default setting for the Rotation\* Sensor block is for ticks above 16 to generate a "true" signal. To switch the "true" portions of the range (setting values below 16 as "true"), you choose "less than" in the pull-down menu.

The feedback box in the configuration panel shows the current number of ticks. (To receive feedback, make sure that the sensor is connected to the chosen port and that communication has been established with the NXT.)

You must drag at least one output data wire from this block's data hub to another block for any information to be sent.

### Block Variables

1. The number at upper right shows which of your NXT's ports are connected to the Rotation Sensor. You can change this number in the configuration panel if you need to.

2. The icon at bottom center shows which direction the sensor will be monitoring: forwards or backwards.
3. The icon at bottom right shows whether the block is set to read ticks or to reset the sensor.
4. The block's data hub will open automatically when the block is placed in the work area. At least one data wire must be dragged from the block's output plug to another block's data hub.



## Light\* Sensor Block

Through data wires, this block can send out the current light value and a logic signal (true/false) based on whether the current light value falls above or below a trigger point.

\* This sensor requires an adaptor cable to connect to the NXT.

Specify the trigger point by using the slider or by typing a value into the input box. To specify the range (above or below the trigger point) that will generate the “true” signal, use the radio buttons or the pull-down menu. The “true” portion of the range will be in color; the “false” portion will be gray.

The default setting for the Light\* sensor block is for light levels above 50% to generate a “true” signal. The radio button to the right of the range is selected and the slider is set at 50. To switch the “true” portions of the range (setting values below 50% as “true”), you would select the left radio button.

The feedback box in the configuration panel shows the current light value. (To receive feedback, make sure that the sensor is connected to the chosen port and that communication has been established with the NXT.)

You must drag at least one output data wire from this block's data hub to another block for any information to be sent.

### Block Variables

1. The number at upper right shows which of your NXT's ports are connected to the light sensor. You can change this number in the configuration panel if you need to.
2. The icon at bottom right indicates at what level the trigger point is set. The more colored bars displayed, the higher the trigger point.
3. The block's data hub will open automatically when the block is placed in the work area. At least one data wire must be dragged from the block's output plug to another block's data hub.



## Temperature\* Sensor Block

Through data wires, this block can send out the current temperature and a logic signal (true/false) based on whether the current temperature is above or below a trigger point.

You can change the default temperature units from degrees Celsius to Fahrenheit (or vice versa) in the configuration panel.

\* This sensor requires an adaptor cable to connect to the NXT.

Specify the trigger point by using the slider or by typing a value into the input box. To specify the range (above or below the trigger point) that will generate the “true” signal, use the radio buttons or the pull-down menu. The “true” portion of the range will be in color; the “false” portion will be gray.

The default setting for the Temperature\* Sensor block is for temperature values below 25° C to generate a

“true” signal. The radio button to the left of the range is selected and the slider is set at 25° C. To switch the “true” portions of the range (setting values above 25° C as “true”), you would select the right radio button.

The feedback box in the configuration panel shows the current temperature in degrees Celsius. (To receive feedback, make sure that the sensor is connected to the chosen port and that communication has been established with the NXT.)

You must drag at least one output data wire from this block’s data hub to another block for any information to be sent. (See the Data Hub section below for more information.)

### Block Variables

1. The number at upper right shows which of your NXT’s ports are connected to the temperature sensor. You can change this number in the configuration panel if you need to.
2. The icon at bottom right indicates at what level the trigger point is set. The more colored bars displayed, the higher the trigger point.
3. The block’s data hub will open automatically when the block is attached to the sequence beam. At least one data wire must be dragged from the block’s output plug to another block’s data hub.

## Flow Blocks



### Wait Block

This block lets your robot sense its environment for a certain condition before it continues. Use the slider or type in a value to set a trigger point so that the program continues when sensor values are below or above it.

1. The number or letter at the top right corner of the Wait block shows the port the block is monitoring. The configuration panel, which is described below, will let you change ports if necessary.
2. If you have chosen to wait for a light, sound, or ultrasonic sensor, this icon indicates at what level the trigger point is set; the more colored bars displayed, the higher the trigger point. If you have chosen to wait for a touch sensor, the icon will display the touch sensor condition (bumped, pressed, or released) that will trigger the block and allow the program to move on.

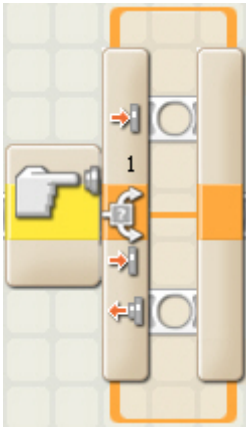


### Loop Block

Use this block to repeat sequences of code. Set the condition that will end the loop: elapsed time, the number of repetitions, a logic signal or a sensor. You can also set a loop to go on forever.

### Block Variables

1. When a Loop block’s action property is set to “Forever,” an infinity symbol ( $\infty$ ) is displayed at the bottom of the trailing portion of the block.
2. If the “Show Counter” checkbox is selected in the configuration panel, a plug will appear that will allow you to use the number of completed loops as an input elsewhere in your program (if you connect a data wire from the plug to another block’s data hub). You can also use current count to control the loop itself. (See the Count setting in the Configuring the Loop Block section below).
3. If you choose for a sensor to control the loop, the trailing portion of the Loop block will expand in size revealing an icon for the chosen sensor. For example, if you choose a Light Sensor to control the loop, a Light Sensor icon will be displayed in the expanded portion of the block. Also, any relevant information about the chosen control property will be displayed at the bottom of the block.



## Switch Block

Use this block to choose between two sequences of code. For example, when configured with a touch sensor, the switch block might run one series of blocks when the sensor is pressed and another when the touch sensor is not.

### Block Variables

1. The icon at the beginning of the switch block indicates the sensor or other condition that will cause the block to choose between the two rows of programming blocks. In this case the current status of a touch sensor will cause the program to switch.

- The upper blocks will run if the touch sensor is pressed.
- The lower block will run if the touch sensor is not being pressed.



## Stop Block

This block will stop your program and any running motors, lamps or sounds. Stopped motors will start to coast.

You can control the Stop block dynamically by connecting data wires (from other blocks' data hubs) to the Stop block's data hub.

# Data Blocks



## Logic Block

This block performs a logical operation on its inputs and sends out the true/false answer by a data wire. The inputs (which must also be "true" or "false") can be set using the radio buttons or supplied dynamically from data wires.

The Logic block uses only two possible values, "true" and "false," for both input and output. Often these values are written as the numbers "1" and "0" where any true statement is written as a 1 and any false statement is written as a 0.

This system is very simple for computers to understand because computer memory has an easy time recording "binary" values, values that come in only two states like the 1 and 0. Computers handle more complicated numbers and equations by building up from 1 and 0.

Note: you can use the values 1 and 0 as input to this block but they must be supplied by logic data wires.

The four logical operations that this block can perform (And, Or, Xor, and Not) allow you to perform a series of comparisons.

### Block Variables

A Logic block dropped into the work area will open with two input ports and three output ports. The input ports will have to be connected to other blocks using data wires (except when one of them is supplied by a constant value that you set with radio buttons).

Output from the logical operation will be delivered from the uppermost output plug; connect this plug using a data wire to another block's data hub. The two output plugs opposite the input plugs allow you to pass the input values on to other blocks if this is necessary.



## Math Block

This block performs simple arithmetic operations like addition, subtraction, multiplication, and division. The input numbers can be typed in or supplied dynamically by data wires.

A Math block dropped into the work area will have an open data hub with two input plugs (on the left) and three output plugs (on the right). The input numbers for the arithmetic equation can be typed in or supplied dynamically by data wires.

Output from the equation will be delivered from the uppermost output plug; connect this plug using a data wire to another block's data hub. The two output plugs opposite the input plugs allow you to pass the input numbers on to other blocks if this is necessary.

### Block Variables

1. Input values can be typed in or supplied dynamically by data wires. The input boxes will be grayed out when data wires are connected.
2. The pull-down menu will let you choose from four arithmetic operations you can perform:
  - Addition (+) [default]
  - Subtraction (-)
  - Multiplication (x)
  - Division (/)



## Compare Block

This block can determine if a number is greater than (>), less than (<), or equal to (=) another number. The input numbers can be typed in or supplied dynamically by data wires.

A Compare block dropped into the work area will have an open data hub with two input ports (on the left) and three output ports (on the right). Both input ports will have to be connected to other blocks using data wires (except when one of the input ports is supplied by a constant value that you type in).

Output from the comparison will be delivered from the uppermost output plug; connect this plug using a data wire to another block's data hub. The two output plugs opposite the input plugs allow you to pass the input values on to other blocks if this is necessary. (See the Data Hub section below for more information.)

### Block Variables

1. Input numbers can be typed in or supplied dynamically by data wires. The input boxes will be grayed out when data wires are connected.
2. The pull-down menu will let you choose from five comparisons you can perform:
  - Greater Than (>)
  - Greater Than or Equal to (>=)
  - Less Than (<)
  - Less Than or Equal to (<=)
  - Equal To (=)



## Range Block

This block can determine if a number is either inside or outside of a range of numbers. The input numbers can be typed in, set using the sliders, or supplied dynamically by data wires. The output logic (true/false) signal will be sent by a data wire.

A Range block dropped on the sequence beam will have an open data hub with three input ports (on the left) and four output ports (on the right). The input ports can be connected to other blocks using data wires (except when one or more of the input ports is supplied by a number that you type in).

Logic output from the Range block will be delivered from the uppermost output plug; connect this plug using a data wire to another block's data hub. The three output plugs opposite the input plugs allow you to pass the input numbers on to other blocks if this is necessary. (See the Data Hub section below for more information.)

### Block Variables

1. The test number can be typed in or supplied dynamically using a data wire. The test box will be grayed out if a data wire is connected.
2. The upper and lower bounds of the range can be set using the slider, typed in, or supplied dynamically by data wires. The input boxes will be grayed out when data wires are connected.
3. The pull-down menu will let you choose from two operations:
  - Inside Range ([])  
In this case, if the test number is between the upper and lower boundary numbers, or is equal to either one of boundary numbers, the Range block will return a value of "true." Otherwise, it will return a value of "false."
  - Outside Range ([])  
In this case, if the test number is outside the range of the upper and lower boundary numbers (and does not include either one of the boundary numbers), the Range block will return a value of "true." Otherwise, it will return a value of "false."



## Random Block

This block generates a random number. You can use random numbers to create unpredictable behavior by your robot. For instance, if you use a data wire to connect a Random block to the Duration plug of a Move block's data hub, your robot might go forward for four seconds one time but then eight seconds the next time you run the program.

The random number will be generated between (and including) minimum and maximum limits that you set. If you want to generate random numbers between 5 and 10, and including 5 and 10 as the possible numbers, set the minimum value to 5 and the maximum value to 10. If you don't want 5 or 10 to be within the set of possible random numbers, choose a minimum value of 6 and a maximum value of 9.

Set the minimum and maximum limits either by using the slider or by typing values into the input box. To get a dynamic number, attach input data wires to the block's data hub.

Output for the block (i.e., the random number) can only be sent through a data wire from the output value (#) plug.

This block's data hub will open automatically when the block is attached to a sequence beam.



## Variable Block

Think of a variable as a place to store a value in the NXT's built-in memory. Other programming blocks can read the variable's current value (and even change it) by connecting to the block with data wires.

A variable's value can change over time. You might create a variable called "Recorded Light Level" and give it an initial value of 50. Then as your robot (fitted with a light sensor) drives to a certain location, you can use a Variable block (set to "write") to record the current light level at the location. To do this, you would connect an output data wire from the Light Sensor block to the input plug of the Variable block.

Later in your program you can use a Variable block (set to "read") to read the light value recorded at the location. Other blocks connected to the block's output plug by data wire would be able to access the information saved in the variable (i.e., the recorded light level).

The block's data hub will open automatically when the block is placed in the work area. A data wire must be connected to this block's output plug for other blocks to read the variable's current value. To write new information to the variable, you must also connect an input data wire to the block's data hub.

# Advanced Blocks



## Text Block

This block can add together groups of characters called text. These are all pieces of text: abcd23, Dog, john!. Text can contain numbers and even special characters like symbols and punctuation marks. Text is important because it can be displayed on the NXT's screen.

A sentence, like this one, is a long piece of text, while each of its individual words is a smaller piece of text. The Text block can take three smaller pieces of text and put them together to make a large piece of text like a sentence.

If input A is the piece of text "This" and inputs B and C are the pieces of text " program is " and "great!", the resulting, combined piece of text would be "This program is great!" Notice that text B (" program is ") includes two blank spaces that are important if the final sentence is to come out right.

Numbers generated by the NXT are not ordinarily text; they can't be displayed directly on the NXT's screen. If you use another programming block called Number To Text, you can convert a number into text. This would allow you to use a Text block to create larger pieces of text that include numbers read from sensors like: "The motor turned 6 times" or "The light sensor reading is 35%." Both of these pieces of text can be displayed on the NXT's screen.

The three pieces of input text can be typed in or supplied dynamically by data wires. The output text will be sent out by data wire.

### Block Variables

1. The data hub will open automatically when the block is attached to a sequence beam. The three pieces of input text can be supplied dynamically by data wires connected to the plugs on the left side of the data hub. The output text will be sent from the plug on the right at the top. The three other output plugs on



the right will pass any input text information through to other blocks, if necessary.

2. The three pieces of input text (A, B, & C) can be typed in or supplied dynamically by data wires. Remember that spaces are important and will be added to the output text only if you include them in the input text.



## Number to Text Block

This block will take a number (like a reading from a sensor) and turn it into text that can be displayed on the NXT's screen. The input number can be typed in or supplied dynamically by a data wire.

Text is a group of characters of any length that you type in using your keyboard. These are all examples of text: abcd23, Dog, John!. Text is important because it can be displayed on the NXT's screen.

The numbers generated by the NXT as it runs your program are not text; they can't be displayed directly on the NXT's screen. You must first convert them to text using a Number To Text block.

When you convert a number into text, you can add it to a sentence that will be displayed on your NXT's screen. By using a different block called a Text block, you can create sentences like: "The motor turned 6 times" or "The light sensor reading is 35%" that can be displayed on the NXT's screen. In these sentences, the numbers 6 and 35 were generated by the NXT (and not typed in) and had to be processed by a Number To Text block before being added to the other pieces of text. (See the Text block entry.)

The Number To Text block's input number can be typed in or supplied dynamically by a data wire. The output text can only be sent out using a data wire.

## Block Variables

1. The data hub will open automatically when the block is attached to a sequence beam. The input number can be supplied dynamically by a data wire connected to an input plug on the left side of the data hub. The output text will be sent from the output plug at the lower right. (The number output plug on the right of the data hub can be used to pass any input number through to other blocks, if necessary.)
2. The input number can be typed in or supplied dynamically by a data wire.



## Keep Alive Block

This block will keep the NXT from entering sleep mode. Add this block if your program needs to wait for longer than the NXT's set "sleep time" (configured using the NXT's on-board menus).

You can control the Keep Alive block dynamically by connecting data wires (from other blocks' data hubs) to the Keep Alive block's data hub.



## File Access Block

With this block you can save data from your robot in files on your NXT. After writing data to a file, you must use another File Access block to close the file before you can read or delete the file using a third File Access block.

For example, to write to a file and then read from the same file, you must close the file between those two actions. You would need three File Access blocks to accomplish this task: the first File

Access block (with “Write” selected in its configuration panel) would write data to a file; a second File Access block positioned somewhere later in the program would have to be set to close the same file; a third File Access block (with “Read” selected in its configuration panel) would be able to read the file. These three blocks could be positioned right next to each other or could be spaced throughout your program.

Writing to an existing file will add data to the end of the file; it will not erase any pre-existing data. To re-write a file, first use a File Access block that deletes the file. Then use another File Access block to write to a new file of the same name.

### Block Variables

The icon at bottom right indicates whether the block is set to Write, Read, Close, or Delete a file.



### Calibration Block

Use this block to calibrate the minimum (0%) and maximum (100%) values detected by a sound or light sensor.

You will need two Calibration blocks to fully calibrate a sound or light sensor: one to set the minimum value for the sensor and another to set the maximum value. You don't need to calibrate both minimum and maximum values; if you only set one of these values, the default will be used for the other.

One way to fully calibrate a sound or light sensor is to set up two Calibration blocks at the beginning of your program with a Wait block (set to wait for a Touch sensor) positioned after each one. If you are calibrating a light sensor, for example, and you set the first Calibration block to calibrate the minimum value and the second Calibration block to calibrate the maximum value, the calibration procedure would happen like this:

1. Place your robot in the darkest place it will encounter during its program and press the touch sensor. This will calibrate the light sensor's minimum value.
2. Then place your robot in the lightest place it will encounter during its program and press the touch sensor. This will calibrate the light sensor's maximum value and your robot will continue on with its program.

You can also calibrate a sensor by placing Calibration blocks in certain positions in your program. For example, if your robot with a built-in light sensor is set to drive under a table (and this will be the darkest place the robot will encounter), you can place a Calibration block (set to calibrate the light sensor's minimum value) in your program in a position so that it is activated when the robot is passing under the table.

Note: Calibration is per sensor, not per port. For example, if you use this block to read the light sensor on port 2 and record this value as the maximum, then ALL light sensors will use that value as maximum (whether or not they use port 2).

Important: Be aware that you can also use the Calibrate Sensors function in the Tools menu to calibrate light and sound sensors.

### Block Variables

The data hub will open automatically when the block is placed in the work area. A number of parameters can be set dynamically using data wires.



## Reset Motor Block

The interactive servo motors have an automatic error correction mechanism that helps your robot move very precisely. However, there may be times when you want to turn this feature off -- the Reset Motor block will let you do this.

### Block Variables

The letter at the top right corner of the block shows which NXT port(s) will be affected by the Reset Motor block.

# Custom Blocks



## My Block

The My Block Builder lets you take a number of blocks you select in the work area and group them together into your own My Block with a customized icon. For example, a My Block you create called "Robot Motion" might group a Move block (to control your robot's wheels) together with a Sound block (that plays a sound file when the robot has traveled a certain distance).

When you create a new program using the same robot, you can just drag the "Robot Motion" block from the Custom palette and you will be done programming this part of the robot's motion because you already set all of the parameters in the My Block.

Over time you may develop a library of My Blocks that you can use in other programs and trade with other MINDSTORMS users.