



NXT control

The NXT object is used to manage the connection between your Android phone and the NXT brick.

Description

Create an NXT object using the **CreateNxt** function of the **app** object:

```
nxt = app.CreateNxt();
```

Methods

Method	Description
Beep(frequency,duration)	Commands the NXT brick to play a tone of a given frequency for a given duration in milliseconds.
Brake(motors)	Commands the NXT brick to Brake one or more of it's motors. The motors parameter can be a single motor, for example just "A" or a combination of motors such as "ABC". Unlike the Stop function, this function actively brakes the motors using power from the battery.
CheckConnection()	Returns true if the NXT brick is currently connected to the phone/tablet but it will show a popup warning message and return false if no brick is currently connected. This function is useful if you want to warn the user that they have no connection when they start pressing buttons etc in your program.
Disconnect()	Disconnects the Bluetooth link between your phone and the NXT brick.
Drive(motors,power,rotation)	Commands the NXT brick to drive one or more of it's motors with a given power and number of rotations. The motors parameter can be a single motor, for example just "A" or a combination of motors such as "ABC". Power is given as a percentage value and negative values make the motor reverse. If the number of rotations is set to zero, then the motor will continue until the Stop function is called.
GetCurrentProgram()	Gets the name of the NXT-G program currently running on the NXT brick.
IsBluetoothEnabled()	Checks if Bluetooth is enabled on the phone/tablet. Returns true if Bluetooth is enabled and false if it is disabled. This is alias for function IsEnabled.

Method	Description
IsConnected()	Returns true if the NXT brick is currently connected to the phone/tablet. Unlike CheckConnected method, it will not show a popup warning message and return false if no brick is currently connected.
IsEnabled()	Checks if Bluetooth is enabled on the phone/tablet. Returns true if Bluetooth is enabled and false if it is disabled. This is the same function as IsBluetoothEnabled.
IsMotorIdle(motor)	Checks if the specified motor ("A", "B", "C") is currently powered. This can be useful in order to check if a previously sent motor command has completed.
PlaySoundFile(file,repeat)	Commands the NXT brick to play a sound file (.rso) file which is available on the brick. The file parameter should be the name of the sound file and the repeat parameter should be number of times you want the sound to repeat.
ReadColorSensor(input,mode)	Reads the color currently being 'seen' by the NXT color sensor. (Note: brick firmware 1.28 or greater is required for this function) The input parameter should be an NXT input port number between 1 and 4 (which the color sensor is plugged into). The mode parameter should be one of the following values:- "ColorDetect", "LightSense", "RedSense", "GreenSense", "BlueSense". If the mode ColorDetect is chosen, then the returned values will be a number between 1 and 6 which represent the following colors: black, blue, green, yellow, red, white. You can use the ToColorName function to convert from these six values to a color name. If any of the other modes are chosen, then the result will be a color intensity value between 0 and 1023.
ReadDistanceSensor(input)	Reads the distance measured by the ultrasonic sensor in centimeters. The input parameter should be an NXT input port number between 1 and 4 (which the ultrasonic sensor is plugged into).
ReadMail(mailbox,type,remove)	Reads a message from the NXT brick's mail box. This message can be written using a normal NXT-G program running on the brick. This allows you to read values from NXT-G programs with your phone or tablet. The mailbox parameter is the target mailbox number and can be a value between 1 and 10. The type parameter should be one of the following values:- "Text", "Number" or "Logic" depending on what type of value you wish to read from the brick's mailbox. The remove parameter should be true if you wish to remove the message from the NXT's mailbox after reading it or false if you wish to leave the message in the mailbox.
ReadLightSensor(input,active)	Reads the intensity of the light currently being 'seen' by the NXT light sensor. The input parameter should be an NXT input port number between 1 and 4 (which the light sensor is plugged into). The active parameter should be set to true if you want the white light to be turned on during sensing and false if the light is not required. The returned value will be a light level value between 0 and 100.
ReadTouchSensor(input)	Reads the state of the NXT touch sensor, which will be true if the switch is currently pushed in and false otherwise. The input parameter should be an NXT input port number between 1 and 4 (which the touch sensor is plugged into).

Method	Description
ReadSoundSensor(input,mode)	<p>Reads the sound pressure level (loudness) of the sound currently being 'heard' by the sound sensor (Note: Sound sensors are not included with the standard NXT kit)</p> <p>The input parameter should be an NXT input port number between 1 and 4 (which the sound sensor is plugged into).</p> <p>The mode parameter should be one of the following values:- "DB" or "DbA" depending if you want the returned value in standard Decibels or A-weighted Decibels.</p>
RequestEnable()	<p>Invokes system dialog box which could enable Bluetooth on tablet/phone when Bluetooth is disabled. If Bluetooth is enabled, this function doesn't invoke any visible dialog box.</p> <p>Be careful when checking Bluetooth state immediately after this function by calling IsEnabled or IsBluetoothEnabled, while RequestEnable doesn't stop program execution, and there is no SetOnRequestEnabled method called after this function invoke. This function is internally invoked by ShowDevices method when Bluetooth is disabled.</p>
SendMail(mailbox,type,message)	<p>Sends a message to the NXT brick's mail box. This message can be read using a normal NXT-G program running on the brick. This allows you to communicate with NXT-G programs from your phone or tablet.</p> <p>The mailbox parameter is the target mailbox number and can be a value between 1 and 10.</p> <p>The type parameter should be one of the following values:- "Text", "Number" or "Logic" depending on what type of value you wish to send to the brick's mailbox.</p>
SetInvert(boolean)	<p>Provides a convenient way to invert the direction of the motor commands.</p>
SetLampColor(input,color)	<p>Commands the NXT brick to set the color sensor lamp to one of the following: 'White', 'Red', 'Green', 'Blue', 'Off'.</p> <p>The input parameter should be an NXT input port number between 1 and 4 (which the color sensor is plugged into).</p>
SetOnConnected(myfunc)	<p>Allows you to set the name of a function that you would like to be called when the NXT has been successfully connected via Bluetooth.</p>
ShowDevices()	<p>Shows the user a dialog box that contains a list of NXT bricks which are paired with the phone. The user can then select the brick to connect to via Bluetooth.</p>
StartProgram(program)	<p>Starts an NXT-G program (.rx file) on the NXT brick (if the program is available on the brick).</p> <p>Note: This function will also launch .rso sound files.</p>
Stop(motors)	<p>Commands the NXT brick to Stop powering one or more of it's motors and allow them to coast to a halt.</p> <p>The motors parameter can be a single motor, for example just "A" or a combination of motors such as "ABC".</p>
StopProgram()	<p>Stops the NXT-G program (.rx file) which is currently running on the NXT brick.</p>
ToColorName(colorNum)	<p>Function returns string with one of the six colors: black, blue, green, yellow, red, white converted from colorNum parameter. You can use the ToColorName function to convert value obtained from color sensor by method ReadColorSensor with mode set to ColorDetect.</p>

