nxtIDE reference manual

XLC Team

Float(*motor*)

Kill power for the motor.

motor (*int*) motor we want to stop.

ClearScreen()

Clear the screen.

OnRev(*motor, speed*)

Set motor to reverse direction and turn it on.

motor (*int*) motor we want to run.

speed (*int*) speed we want to run the motor at from 0 to 100. Negative value reverses direction.

Sensor(*sensor*)

Read value from given sensor.

sensor (*int*) sensor we want to read from

RectOut(*x, y, width, height*)

Draw a rectangle from [x, y] with specified width and height.

x (*int*) X coordinate of the start point of the rectangle.

y (*int*) Y coordinate of the start point of the rectangle.

width (*int*) The width of the rectangle.

height (*int*) The height of the rectangle.

SetSensor(*sensor, type*)

TextOut(*x, y, text*)

Print text on the screen.

x (*int*) X coordinate of the text

y (*int*) Y coordinate or the text

text (*str*) The text to print

SetSensorLowspeed(*sensor*)

ClearLine(*line*)

Clear one line on the screen.

line (*int*) line we want to clear.

SensorUS(*sensor*)

Read value from given lowspeed sensor (e.g. Ultrasonic). The input port has to be configured as a Lowspeed before using this function.

sensor (*int*) sensor we want to read from

Wait(*milisec*)

Waits for given number of miliseconds.

milisec (*int*) number of miliseconds

Off(*motor*)

Turn the motor off (with break).

motor (*int*) motor we want to stop.

PlayTone(*freq, duration*)

Play a tone.

freq (*int*) Frequency of the tone in Hz.

duration (*int*) For how long should the brick play this tone.

NumOut(*x, y, num*)

Print number on the screen.

x (*int*) X coordinate of the text

y (*int*) Y coordinate or the text

num (*int*) The number to print

LineOut(*x0, y0, x1, y1*)

Draw a line from [x0, y0] to [x1, y1].

x0 (*int*) X coordinate of the start point of the line

y0 (*int*) Y coordinate of the start point of the line

x1 (*int*) X coordinate of the end point of the line

y1 (*int*) Y coordinate of the start point of the line

CircleOut(*x, y, radius*)

Draw a circle with center at [x, y] and specified radius.

x (*int*) X coordinate of the center of the circle.

x (*int*) Y coordinate of the center of the circle.

radius (*int*) The radius of the circle.

MotorTachoCount(*motor*)

Get motor tachometer counter value.

motor (*int*) motor we want to get tachometer count from.

PointOut(*x, y*)

Draw a point on the screen at (x, y)

x (*int*) The x coordinate of the point

y (*int*) The y coordinate of the point

Random(*n = 0*)

Returns a random number

n (*int*) the maximal value this function should return

OnFwd(*motor, speed*)

Set motor to forward direction and turn it on.

motor (*int*) motor we want to run.

speed (*int*) speed we want to run the motor at from 0 to 100. Negative value reverses direction.

ResetTachoCount(*motor*)

Reset tachometer counter.

motor (*int*) desired motor output.

RotateMotor(*motor, speed, angle*)

Rotate motor in specified direction at specified speed for the specified number of degrees.

motor (*int*) motor we want to rotate

speed (*int*) speed we want to run the motor at, from 0 to 100. Negative value reverses direction.

angle (*int*) number of degrees we want to rotate the motor. Negative value reverses direction.