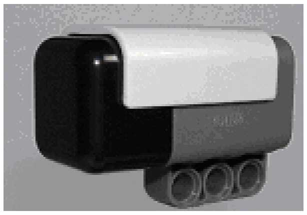
HiTechnic NXT Acceleration Sensor for LEGO Mindstorms NXT

Introduction



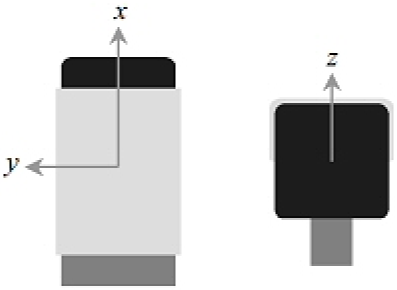
The NXT Acceleration Sensor contains a three axis accelerometer that measures acceleration in three axes, *x, y* and *z*. Acceleration is measured in the range of –2g to +2g with scaling of approximately 200 counts per g.

The Acceleration Sensor can also be used to measure tilt in three axes.

The Acceleration Sensor connects to an NXT sensor port using a standard NXT wire and uses the digital I2C communications protocol. The acceleration measurement for each axis is refreshed approximately 100 times per second.

The Acceleration Sensor is housed in a standard Mindstorms sensor housing to match the other Mindstorms elements.

The three axes of measurement are labeled *x, y*, and *z* as shown.



To test your new sensor, plug it into port 2 of your NXT and select View > Ultrasonic cm > Port 2. Hold the sensor level in your hand and slowly tilt it forward and back. The value displayed on the NXT will represent the acceleration or tilt value for the *x* axis and will be in the range of 0 - 254. (0 will display as ?????? while in View mode.) Note that only the *x* axis can be displayed using the view function.

Programming

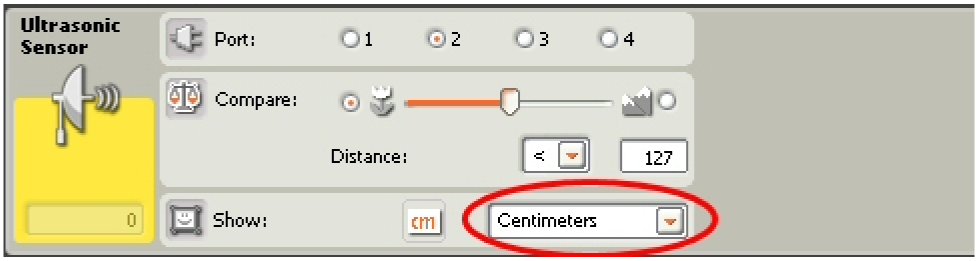
Mindstorms NXT-G

The Acceleration Sensor can be programmed using LEGO Mindstorms NXT Software Acceleration Sensor Block. If the Acceleration Sensor Block is not available you may also use the standard Ultrasonic Block to access the *x* axis output.

The Acceleration Sensor Block will be available for free download at the time the Acceleration sensor is available.

Ultrasonic Block

If using the Ultrasonic Block to program the Acceleration Sensor, configure the block to centimeter mode as shown.



Ultrasonic Sensor

Port: ○ 1 ● 2 ○ 3 ○ 4

Compare: ● ○

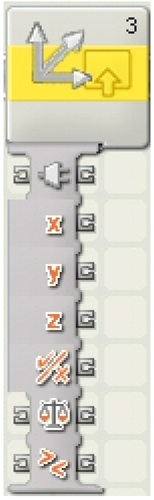
Distance: < 127

Show: cm Centimeters

0

The Ultrasonic Block will return the X axis value which will be in the range 0 - 254.

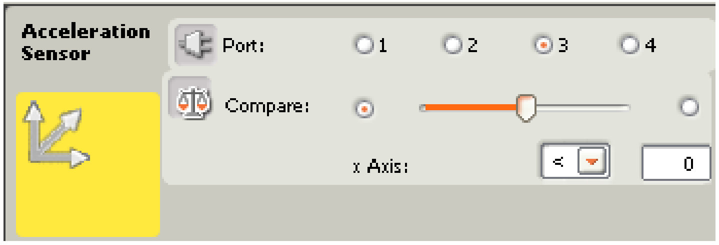
Acceleration Sensor Block



The Acceleration Block will return the value for the positive or negative acceleration for each axis, *x*, *y* and *z* as shown.

In additon, a trigger point may be set for the x axis. The conditions for the trigger will be met whenever the *x* axis value exceeds the trigger.

Configuring the Acceleration Sensor Block



Acceleration Sensor

Port: ○ 1 ○ 2 ● 3 ○ 4

Compare: ● ○

x Axis: < 0

1. Choose the port where your Acceleration sensor is plugged in. By default, the block will be set to port 3 for an acceleration sensor. You can change this selection if you need to.
2. If you choose the Compare function, the block will be triggered when the *x* axis reading is above or below the selected value; select < "less than" to trigger the block when the reading is below the trigger value or > "greater than" to trigger the block when the reading is above the trigger value. Use the slider to set the trigger value or type it directly into the input box.

Sensor Register Layout

Address	Type	Contents
42H	byte	*X* axis upper 8 bits
43H	byte	*Y* axis upper 8 bits
44H	byte	*Z* axis upper 8 bits
45H	byte	*X* axis lower 2 bits
46H	byte	*Y* axis lower 2 bits
47H	byte	*Z* axis lower 2 bits

Other Programming Environments

RobotC

All features of the HiTechnic Acceleration Sensor can be accessed using RobotC.
For more information go to http://www-education.rec.ri.cmu.edu/robotc/.

NXC

NXC is a C programming language that can access all Acceleration Sensor features.
For more information go to http://bricxcc.sourceforge.net/nbc/.

Example NXC Code

```
#include "NXCDefs.h"
task main()
{
  SetSensorLowspeed(IN_1);
  int count;

  int xval;
  int yval;
  int zval;

  byte inI2Ccmd[];
  byte outbuf[];
```

```
while (TRUE)
 {
 ArrayInit(inI2Ccmd, 0, 2); // set the buffer to hold 10 values (initially all are zero)
 inI2Ccmd[0] = 0x02; // set values in the array
 inI2Ccmd[1] = 0x42;
 count=8;                      //read count set to 8 bytes
 I2CBytes(IN_1, inI2Ccmd, count, outbuf);  //read the acceleration sensor on port 1
 xval=outbuf[0];                 //load x axis upper 8 bits
 yval=outbuf[1];                 //load Y axis upper 8 bits
 zval=outbuf[2];                 //load z axis upper 8 bits
 if (xval > 127) xval-=256;         //convert x to 10 bit value
 xval=xval*4 + outbuf[3];
 if (yval > 127) yval-=256;         //convert y to 10 bit value
 yval=yval*4 + outbuf[4];
 if (zval > 127) zval-=256;         //convert z to 10 bit value
 zval=zval*4 + outbuf[5];
 ...

 }
```

Notes:

NXT Firmware version 1.03 must be loaded in the NXT for the compass to operate correctly. You can check the firmware version by displaying the NXT Window in the Mindstorms software.