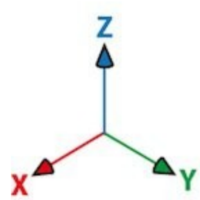
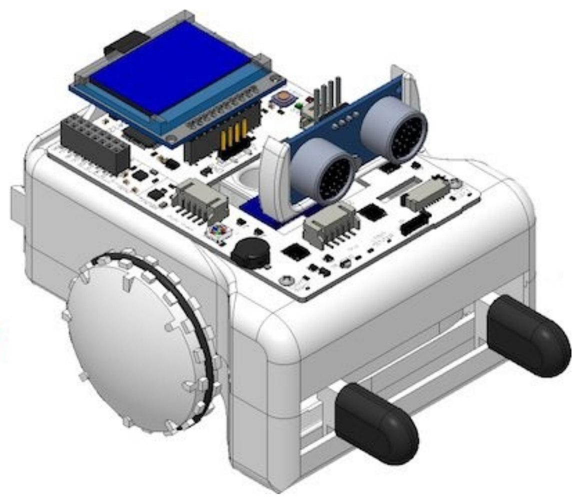
Sparki With Arduino, Guide #4 – Accelerometer

Sparki has many components, including the ultrasonic distance ranger, light detectors, line following sensors, and more. Lets take a look at the 3-Axis accelerometer. The accelerometer can be used for a lot of things, but in this lesson we will have a simple goal in mind: make music by moving Sparki with our hands.

How It Works

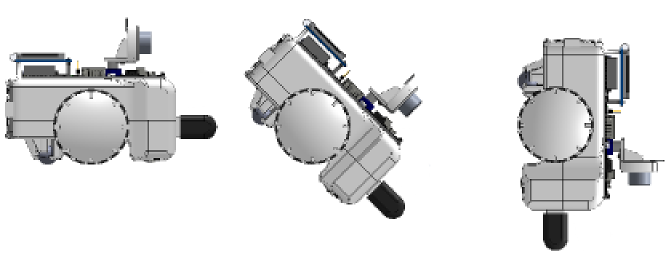
The accelerometer on the sparki is an MicroEletroMechanical device that's able to measure the acceleration in 3 separate axes. We call these the X, Y and Z axes. To the right is a map of those Axes. These are the basic instructions to read the accelerations:





```
sparki.accelX();
sparki.accelY();
sparki.accelZ();
```

Acceleration is the how fast something speeds up. For example, if something goes from a standstill to a speed of 10 meters second, and takes 10 seconds to reach that speed, then its acceleration is 10 meters per second per 10 seconds, which through algebra can be simplified to 1 m/s/s, or 1m/s^2. One thing you'll notice later is that with the Sparki still on a level surface the Z axis has a negative reading. This is because of gravity accelerating everything toward the ground at 9.8m/s^2, or -9.8m/s^2 from the Sparki's perspective, which is a good thing or else we'd all be in space! If you tilt the Sparki you change the perspective and the values change accordingly, as shown below.



Accel X: 0.00	Accel X: 0.00	Accel X: 0.00
Accel Y: 0.00	Accel Y: 6.00	Accel Y: -9.81
Accel Z: -9.81	Accel Z: -8.00	Accel Z: 0.00

To start, lets run a sketch to show the accelerometer values on the LCD. Upload the program from the next page and move the Sparki around to become familiar with how the values change. This will be helpful when trying to make the Sparki play a tune later.

```
/*******************************************
Basic Accelerometer Sensor test

Sparki has a 3-Axis accelerometer. It is
used to detect the acceleration Sparki is
experiencing. Usually, this mostly means the
force of gravity. It can do this in all
3 XYZ axis, which are left/right (X),
forward/backwards (Y), and up/down (Z). This
is the same part that smartphones use to tell
how you're tilting them.

This program shows how to read the sensor
and display the information on the LCD.
********************************************/
#include <Sparki.h> // include the sparki library

void setup()
{
}

void loop() {
  sparki.clearLCD(); // wipe the screen

  float x  = sparki.accelX();   // measure the accelerometer x-axis
  float y  = sparki.accelY();   // measure the accelerometer y-axis
  float z  = sparki.accelZ();   // measure the accelerometer z-axis

  // write the measurements to the screen
  sparki.print("Accel X: ");
  sparki.println(x);

  sparki.print("Accel Y: ");
  sparki.println(y);

  sparki.print("Accel Z: ");
  sparki.println(z);

  sparki.updateLCD(); // display all of the information written to
the screen
  delay(100);
}
```

You'll notice the values produced have some small variations even when not moving the Sparki. Accelerometers are very sensitive and can pick up quite small vibrations. You'd normally want to record multiple samples and smooth out this value, but since we're just experimenting we'll skip that and let them add some character to our music. A musical note will be defined by two parameters: its frequency and its duration. So let's first make different notes, which means modifying the frequency sent to Sparki's buzzer but with a fixed duration. Let's use the Y value to make the notes. The idea is simple:

- Read the Y acceleration value from the accelerometer and make that value to always be positive (since frequencies can not be negative!) using the **abs()** function.

•Multiply that value for a number. The number should be big enough to convert our Y reading in a frequency corresponding to an audible sound that can be sent to the buzzer. For example: a frequency of 440Hz is the A note. So, let's start multiplying the reading by 100. Thus if Sparki is fully vertical (in the Y axis), the frequency will be near 1000 Hz (or 1 KHz), since 9.8 x 100 = 980.

•Wait for some milliseconds (note's duration) using the delay function.

Here is the code. We are just modifying the previous example, since viewing the accelerometer values is always useful:

```
#include <Sparki.h> // include the sparki library

void setup()
{
}

void loop() {
  sparki.clearLCD(); // wipe the screen

  float x  = sparki.accelX();   // measure the accelerometer x-axis
  float y  = sparki.accelY();   // measure the accelerometer y-axis
  float z  = sparki.accelZ();   // measure the accelerometer z-axis

  // write the measurements to the screen
  sparki.print("Accel X: ");
  sparki.println(x);

  sparki.print("Accel Y: ");
  sparki.println(y);

  sparki.print("Accel Z: ");
  sparki.println(z);

  sparki.updateLCD(); // display all of the information written to the screen

  sparki.beep(abs(y)*100);

  delay(300); // note duration
}
```

See the key line? `sparki.beep(abs(y)*100);` You can experiment with that 100 value to create higher and lower notes. To complete our musical instrument, we can change the delay in a way that the note duration is defined by the X value from the accelerometer instead of being a fixed number. Let's try this line of code in place of the `delay(300);`:

```
delay(abs(x)*50 + 100); // note duration: the 100 number is there to prevent zero duration notes.
```

Of course, you can now play with the numbers in that line too! Did you ever think you'd use math and music at the same time? The two are actually quite related!