**BHQ Sensors**

The "raw data" sensors may be divided into three categories:

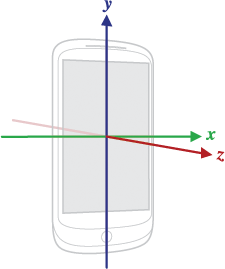
* Movement sensors
* Environment sensors
* Communication sensors

1. **Movement sensors (i.e motion and position):**

This category includes Motion Sensors - measuring acceleration forces and rotational forces along three axes, and Position Sensors, that measure a device's physical position, by using a specific coordinate system:

Sensor Coordinate System

In general, the sensor framework uses a standard 3-axis coordinate system to express data values. For most sensors, the coordinate system is defined relative to the device's screen when the device is held in its default orientation (Figure 1). When a device is held in its default orientation, the X axis is horizontal and points to the right, the Y axis is vertical and points up, and the Z axis points toward the outside of the screen face. In this system, coordinates behind the screen have negative Z values. This coordinate system is used by the following sensors:



**Figure 1.** Coordinate system (relative to a device) that's used by the Sensor API.

It is important to understand that in this coordinate system the axes are not swapped when the device's screen orientation changes—that is, the sensor's coordinate system never changes as the device moves. This behavior is the same as the behavior of the OpenGL coordinate system. Also, the sensor coordinate system is always based on the natural (default) orientation of a device (Portrait/Landscape orientations are not assumed).

The sensors in this category are:

* 1. Accelerometer

Measures the acceleration force that is applied to a device on all three physical axes *(x, y, z),* including the force of gravity.

Measurement Units:.

Accelerometers use the standard sensor coordinate system. In practice, this means that the following conditions apply when a device is laying flat on a table in its natural orientation:

* If you push the device on the left side (so it moves to the right), the x acceleration value is positive.
* If you push the device on the bottom (so it moves away from you), the y acceleration value is positive.
* If you push the device toward the sky with an acceleration of A m/s2, the z acceleration value is equal to A + 9.81, which corresponds to the acceleration of the device (+A m/s2) minus the force of gravity (-9.81 m/s2).
* The stationary device will have an acceleration value of +9.81, which corresponds to the acceleration of the device (0 m/s2 minus the force of gravity, which is -9.81 m/s2).

In general, the accelerometer is a good sensor to use if you are monitoring device motion. Almost every Android-powered handset and tablet has an accelerometer, and it uses about 10 times less power than the other motion sensors.

* 1. Gyroscope

Measures a device's rate of rotation around each of the three physical axes*(x, y, z).* Measurement Units:

* 1. Magnetic field

Measures the ambient geomagnetic field for all three physical axes *(x, y, z)*.

Measurement Units:

All of the motion sensors (1.1., 1.2., 1.3) return multi-dimensional arrays of sensor values for each sensor event. For example, during a single sensor event the accelerometer returns acceleration force data for the three coordinate axes, and the gyroscope returns rate of rotation data for the three coordinate axes. These data values are returned in a float array along with other sensor event parameters.

In order to use the raw data values from these sensors effectively, you need to filter out factors from the environment, such as gravity. You might also need to apply a smoothing algorithm to the trend of values to reduce noise.

To measure the real acceleration of the device, the contribution of the force of gravity must be removed. This can be achieved by applying a high-pass filter. Conversely, a low-pass filter can be used to isolate the force of gravity.

* 1. Activity

The significant motion sensor triggers an event each time significant motion is detected and then it disables itself. A significant motion is a motion that might lead to a change in the user's location; for example walking, biking, or sitting in a moving car.

* 1. Step Count

The step counter sensor provides the number of steps taken by the user since the last reboot while the sensor was activated.

\* **Note**: You must declare the [ACTIVITY\_RECOGNITION](https://developer.android.com/reference/android/Manifest.permission#ACTIVITY_RECOGNITION) permission in order for your app to use this sensor on devices running Android 10 (API level 29) or higher.

1. **Environment sensors**

Sensors in this category provide information regarding the user's surroundings.

* 1. Location

The locations sensor provides the best available location estimate for the users’ current location at specified intervals (our default is 15 minutes). This is based on the best currently available location providers such as WiFi and GPS, or if not available, cellular network location through proximity to cellular cell towers. The provider determines the accuracy of the location, the best resolution achieved by GPS is approximately 100 m.

Note that the information regarding the specific geographic location of the user is encrypted (hashed) prior to it's registration in our data base, to ensure the users' privacy.

The encryption is constant for each specific location so that "unique locations" of each user may be acknowledged but not identified. The distances between each two locations remains genuine.

* 1. Light:

Measures the ambient light level (illumination).

Measurement Units: .

You can use this sensor to monitor relative ambient illuminance near an Android-powered device. The raw data you acquire from the light sensor usually requires no calibration, filtering, or modification. This sensor is hardware-base, but since most device manufacturers use it to control screen brightness, it is usually available in all devices. The sensor returns a single value for each data event (rather than a multi-dimensional array of sensor values returned by the motion sensors). Unlike motion sensors and position sensors, which often require high-pass or low-pass filtering, environment sensors such as the Light sensor do not typically require any data filtering or processing.

* 1. Screen – Provides information regarding screen on/off times.
  2. Charge – Provides information regarding charging/Discharging status and the battery percentage.

1. **Communication sensors**
   1. Bluetooth

The Bluetooth sensor logs the mobile device’s Bluetooth sensor and detects surrounding Bluetooth-enabled and visible devices with respective RSSI dB values at specified intervals (our default is 15 minutes). A scan session assigns the same timestamp for all the found Bluetooth devices.

* 1. Wi-Fi

Logs the mobile device’s Wi-Fi sensor, current AP and surrounding Wi-Fi visible devices with respective RSSI dB values at specified intervals (our default is 15 minutes), while the router's IP is hashed to ensure privacy. Additional provided information is the distance between the devices and the router.

As the encryption is constant, this sensor can help identifying unique locations (home/work).

* 1. Calls

Provides the call logs at specified time intervals (our default is 30 minutes).

Note that the contact's identity remains encrypted prior to registration into our data base to ensure users' privacy. Each contact gets a specific identity so that common contact may be acknowledged but not identified.

* 1. Usage Events

The BHQ platform provides information regarding the user's app usage. This information is also encrypted before registration to our data base to ensure user privacy. The applications are divided into 7 categories (Music and Audio, Images, Communication and Social, Maps Navigation, Productivity), and only the category number is logged. The information includes the time duration of which the application was visible (used by the user) and the time it was running in the background.