

Sensors

Android sensors give applications access to a mobile device's underlying physical sensors. They are data-providing virtual devices defined by [sensors.h](https://android.googlesource.com/platform/hardware/libhardware/+master/include/hardware/sensors.h)



(<https://android.googlesource.com/platform/hardware/libhardware/+master/include/hardware/sensors.h>)

, the sensor Hardware Abstraction Layer (HAL).

What are Android sensors?

Android sensors are virtual devices that provide data coming from a set of physical sensors: accelerometers, gyroscopes, magnetometers, barometer, humidity, pressure, light, proximity and heart rate sensors.

Not included in the list of physical devices providing data are camera, fingerprint sensor, microphone, and touch screen. These devices have their own reporting mechanism; the separation is arbitrary, but in general, Android sensors provide lower bandwidth data. For example, “100hz x 3 channels” for an accelerometer versus “25hz x 8 MP x 3 channels” for a camera or “44kHz x 1 channel” for a microphone.

Android does not define how the different physical sensors are connected to the system on chip (SoC).

- Often, sensor chips are connected to the SoC through a [sensor hub](#) (/devices/sensors/sensor-stack#sensor_hub), allowing some low-power monitoring and processing of the data.
- Often, Inter-Integrated Circuit (I2C) or Serial Peripheral Interface (SPI) is used as the transport mechanism.
- To reduce power consumption, some architectures are hierarchical, with some minimal processing being done in the application-specific integrated circuit (ASIC - like motion detection on the accelerometer chip), and more is done in a microcontroller (like step detection in a sensor hub).
- It is up to the device manufacturer to choose an architecture based on accuracy, power, price and package-size characteristics. See [Sensor stack](#)

(/devices/sensors/sensor-stack) for more information.

- Batching capabilities are an important consideration for power optimization. See Batching (/devices/sensors/batching) for more information.

Each Android sensor has a “type” representing how the sensor behaves and what data it provides.

- The official Android Sensor types (/devices/sensors/sensor-types) are defined in sensors.h
(<https://android.googlesource.com/platform/hardware/libhardware/+/-/master/include/hardware/sensors.h>)
under the names SENSOR_TYPE_...
 - The vast majority of sensors have an official sensor type.
 - Those types are documented in the Android SDK.
 - Behavior of sensors with those types are tested in the Android Compatibility Test Suite (CTS).
- If a manufacturer integrates a new kind of sensor on an Android device, the manufacturer can define its own temporary type to refer to it.
 - Those types are undocumented, so application developers are unlikely to use them, either because they don't know about them, or know that they are rarely present (only on some devices from this specific manufacturer).
 - They are not tested by CTS.
 - Once Android defines an official sensor type for this kind of sensor, manufacturers must stop using their own temporary type and use the official type instead. This way, the sensor will be used by more application developers.
- The list of all sensors present on the device is reported by the HAL implementation.
 - There can be several sensors of the same type. For example, two proximity sensors or two accelerometers.
 - The vast majority of applications request only a single sensor of a given type. For example, an application requesting the default accelerometer will get the first accelerometer in the list.
 - Sensors are often defined by wake-up
(/devices/sensors/suspend-mode#wake-up_sensors) and non-wake-up
(/devices/sensors/suspend-mode#non-wake-up_sensors) pairs, both sensors sharing

the same type, but differing by their wake-up characteristic.

Android sensors provide data as a series of sensor events.

Each event (/devices/sensors/hal-interface#sensors_event_t) contains:

- a handle to the sensor that generated it
- the timestamp at which the event was detected or measured
- and some data

The interpretation of the reported data depends on the sensor type. See the sensor type (/devices/sensors/sensor-types) definitions for details on what data is reported for each sensor type.

Existing documentation

Targeted at developers

- Overview
 - https://developer.android.com/guide/topics/sensors/sensors_overview.html
(https://developer.android.com/guide/topics/sensors/sensors_overview.html)
- SDK reference
 - <https://developer.android.com/reference/android/hardware/SensorManager>
(https://developer.android.com/reference/android/hardware/SensorManager)
 - <https://developer.android.com/reference/android/hardware/SensorEventListener>
(https://developer.android.com/reference/android/hardware/SensorEventListener)
 - <https://developer.android.com/reference/android/hardware/SensorEvent>
(https://developer.android.com/reference/android/hardware/SensorEvent)
 - <https://developer.android.com/reference/android/hardware/Sensor>
(https://developer.android.com/reference/android/hardware/Sensor)
- Stack Overflow and tutorial websites
 - Because sensors documentation was sometimes lacking, developers resorted to Q&A websites like Stack Overflow to find answers.
 - Some tutorial websites exist as well, but do not cover the latest features like batching, significant motion and game rotation vectors.

- The answers over there are not always right, and show where more documentation is needed.

Targeted at manufacturers

- Overview
 - This [Sensors \(/devices/sensors/\)](/devices/sensors/) page and its sub-pages.
- Hardware abstraction layer (HAL)
 - </platform/hardware/libhardware/+master/include/hardware/sensors.h>
(<https://android.googlesource.com/platform/hardware/libhardware/+master/include/hardware/sensors.h>)
 - Also known as “sensors.h”
 - The source of truth. First document to be updated when new features are developed.
- Android CDD (Compatibility Definition Document)
 - <https://source.android.com/compatibility/9/android-9-cdd>
(/compatibility/9/android-9-cdd)
 - See sections relative to sensors.
 - The CDD is lenient, so satisfying the CDD requirements is not enough to ensure high quality sensors.

Content and code samples on this page are subject to the licenses described in the [Content License \(/license\)](/license).
Java is a registered trademark of Oracle and/or its affiliates.