**How to read Sensor Info IOS:**

<https://developer.apple.com/documentation/coremotion>

<https://navoshta.com/motion-sensor-data/>

# **Motion Sensors in iOS**

Apple mobile devices have so many capabilities nowadays, that it is not always obvious where this or that functionality is coming from. Have you ever thought of how the Google Cardboard VR apps work? The answer is — they all use device motion sensors, be it an Android or iOS device.

Most of the contemporary mobile devices have a set of sensors that let you track device’s position in space in some directions and axes to some degree of accuracy. And the accuracy is reasonably good, allowing the app to track device’s movements so precisely that it can even create some sort of a VR experience for the user, when paired with a special Google Cardboard hardware.

Now, most of iOS devices are equipped with three sensors:

* **Accelerometer**. Measures non-gravitational acceleration along the X, Y or Z axes.
* **Gyroscope**. Measures device rotation with respect to Earth gravity.
* **Magnetometer**. Measures the strength of the magnetic field surrounding the device.

In terms of iOS SDK all three sensors are handled through CMMotionManager class. It lets you check if specific sensor is available and subscribe to sensor updates. There is a forth “sensor” available in CMMotionManager which is labeled as Device Motion — although in fact this is not a sensor, but rather a conjunction of other sensors’ output processed by a specific algorithm.

The API is really easy and straightforward. In order to start receiving accelerometer updates, for instance, you simply need to do the following:

let motionManager = CMMotionManager()

if motionManager.accelerometerAvailable {

motionManager.accelerometerUpdateInterval = 0.1

motionManager.startAccelerometerUpdatesToQueue(NSOperationQueue.mainQueue()) { (accelerometerData, error) in

guard let accelerometerData = accelerometerData where error == nil else {

/// Report an error.

return

}

/// Handle accelerometer data.

}}

You check if the accelerometer is available, configure update interval and provide a closure to be triggered on accelerometer updates.