

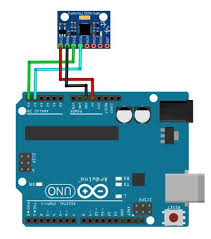
MPU-6050 & Unity

a Data Acquisition cross-over project

Mohamed Essam Soliman | Alieldin Ayman | 5/10/2018

# MPU-6050

The InvenSense MPU-6050 sensor contains a MEMS accelerometer and a MEMS gyro in a single chip. It is very accurate, as it contains 16-bits analog to digital conversion hardware for each channel. Therefor it captures the x, y, and z channel at the same time. The sensor uses the I2C-bus to interface with the Arduino.



Gyroscope Features

• Digital-output X-, Y-, and Z-Axis angular rate sensors (gyroscopes) with a user- programmable full scale range of ±250, ±500, ±1000, and ±2000°/sec

• External sync signal connected to the FSYNC pin supports image, video and GPS synchronization

• Integrated 16-bit ADCs enable simultaneous sampling of gyros • Enhanced bias and sensitivity temperature stability reduces the need for user calibration

• Improved low-frequency noise performance

• Digitally-programmable low-pass filter

• Gyroscope operating current: 3.6mA • Standby current: 5µA

• Factory calibrated sensitivity scale factor • User self-test

Accelerometer Features

• Digital-output triple-axis accelerometer with a programmable full scale range of ±2g, ±4g, ±8g and ±16g

• Integrated 16-bit ADCs enable simultaneous sampling of accelerometers while requiring no external multiplexer

• Accelerometer normal operating current: 500µA

• Low power accelerometer mode current: 10µA at 1.25Hz, 20µA at 5Hz, 60µA at 20Hz, 110µA at 40Hz

• Orientation detection and signaling

• Tap detection

• User-programmable interrupts

• High-G interrupt

• User self-test

Motion Processing

• Internal Digital Motion Processing™ (DMP™) engine supports 3D Motion Processing and gesture recognition algorithms

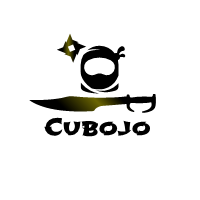
• The MPU-60X0 collects gyroscope and accelerometer data while synchronizing data sampling at a user defined rate. The total dataset obtained by the MPU-60X0 includes 3-Axis gyroscope data, 3- Axis accelerometer data, and temperature data. The MPU’s calculated output to the system processor can also include heading data from a digital 3-axis third party magnetometer.

• The FIFO buffers the complete data set, reducing timing requirements on the system processor by allowing the processor burst read the FIFO data. After burst reading the FIFO data, the system processor can save power by entering a low-power sleep mode while the MPU collects more data.

• Programmable interrupt supports features such as gesture recognition, panning, zooming, scrolling, tap detection, and shake detection

• Digitally-programmable low-pass filters

• Low-power pedometer functionality allows the host processor to sleep while the DMP maintains the step count.



We managed to integrate the functionality of this sensor into Unity3D game-engine, by telling the Arduino to send its readings in a string, that we can parse in Unity. This allowed us to set-up an environment where we can control a game object by applying the integration of the acceleration from the accelerometer, and the angles on the axes from the gyrometer on any object in the scene of the game.

This resulted in the ability to remotely control an object, forming a motion sensor game controller. That we then built a scene, that puts you in a dojo, where you’d have to train on slashing blocks thrown at you in a random fashion. And so **Cubojo** was born, a cross-over between MPU-6050 and Unity.

**Cubojo** is still in its initial state, we’re planning to add physics, levels, more scenes and functionality. Also, we’ll be using the MPU6050 to make our own game controller that can be used and optimized for a variety of games.

\*The Unity cross-over code, Arduino code, and GIF are attached with the report.