

Android_加速度センサー

概要^{1.0版}

システムセンス(株)
遠藤

Ihr Logo

PresentationPoint



変更履歴

日付	変更箇所	版数	改版者
2010/02/19	初版	1.0	遠藤





目次

- Androidとセンサー
- 加速度とは何か？
- 3方向で得られる加速度
- 重力加速度とは何か？
- SensorManagerクラスと実装方法
- ローパスフィルタとハイパスフィルタ



Androidのセンサー

Android SDKは加速度、ジャイロ(角速度)、磁界、圧力、温度など各種センサーに対応しており、2009年4月時点で発売されている実機では以下の4種類の動作が確認されています。

- ・TYPE_ACCELEROMETER(加速度)
- ・TYPE_ORIENTATION(傾き)
- ・TYPE_MAGNETIC_FIELD(磁界)
- ・TYPE_TEMPERATURE(温度)

本ドキュメントでは加速度センサーの使い方について解説します。



加速度とは

加速度とは単位時間あたりの速度の変化率のことであり、通常 m/s^2 (メートル毎秒毎秒)の単位で表されます。

加速度 a は以下の式のように速度の時間微分であり

$$a = dv/dt$$

逆に加速度を時間で積分をすることで速度が得られ、端末の運動状態を知ることができます。

また加速度 a は質量 M 、力 F から次のようにも表され

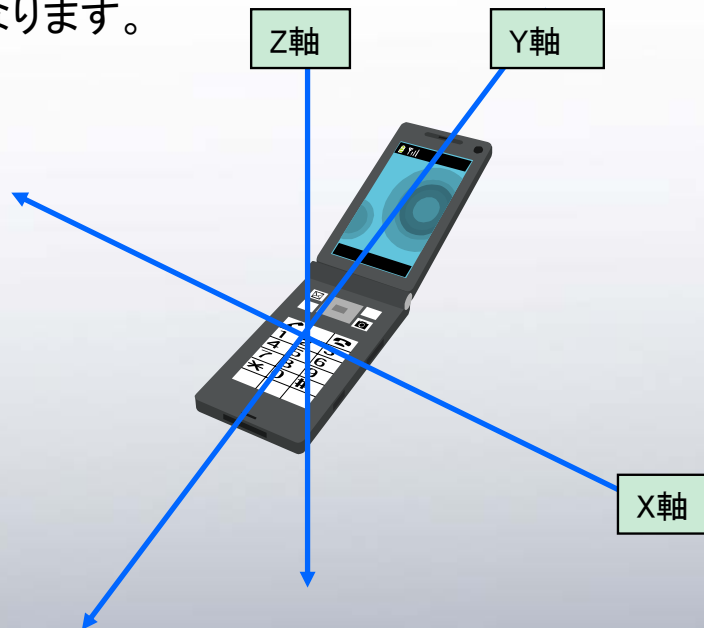
$$F = Ma$$

物体に加速度がかかることと力が加わることは物理学的に全く等価であることが証明されます(運動の第2法則)。

例えば重力と加速度により発生する慣性力を区別することは不可能です。

加速度は3軸のベクトルで得られる

加速度センサーで得られる加速度はX軸、Y軸、Z軸の3軸のベクトルに分解されそれぞれ個別に得ることができます。X軸が端末の左右方向、Y軸が上下方向、Z軸が裏表方向になります。





重力加速度

地球など重力のかかる場所では加速度センサーは重力の影響を受けます。

例えば地球上で端末を平らな場所で静止させると、X軸=0.0、Y軸=0.0、Z軸=9.80665(地球の重力加速度)が得られます。

端末が自由落下している時、あるいは端末が無重力下にあるとき3軸全ての値が0になります。

この3軸の配分を見ることで端末の傾きを得ることもできますが、端末が加速度運動している場合などはその影響も現れるので注意が必要です。傾きのみを取得したい場合は角度を返してくれる傾きセンサーを使う方が便利です。

また後ほど説明するSensorManagerクラスには様々な星の重力が定義されていて、地球の重力加速度はGRAVITY_EARTHでいつでも使用することができます。

他に火星や木星、太陽、デススターなどの重力加速度も定数として定義されています。



SensorManagerクラスと実装の仕方

SensorManagerは端末がセンサーにアクセスするためのクラスです。このクラスのインスタンスを使用することで、センサーイベントのリスナの登録やセンサー情報の取得などを行います。

SensorManagerは直接インスタンスを生成することができないのでその方法を解説していきます。

メンバ変数として宣言します。

```
private SensorManager sensorManager;
```

onCreateメソッドでgetSystemServiceメソッドを使用します。

```
sensorManager = (SensorManager)getSystemService(Context.SENSOR_SERVICE);
```


onCreateやonClickなどセンサーを動かしたいタイミングでリスナーを登録します

○

```
sensorManager.addListener(this,  
sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),  
SensorManager.SENSOR_DELAY_NORMAL);
```

2つ目の引数で加速度センサー(TYPE_ACCELEROMETER)を指定し、3つ目の引数でセンサーの反応速度を指定します。センサーの反応速度は4段階で指定できます

定数名	説明	実際の値	最小取得周期※
SENSOR_DELAY_FASTEST	最速	0	約20ms(1秒間に50回程度)
SENSOR_DELAY_GAME	ゲーム用	1	約40ms(1秒間に25回程度)
SENSOR_DELAY_UI	ユーザーインターフェース用	2	約80ms(1秒間に12.5回程度)
SENSOR_DELAY_NORMAL	通常	3	約220ms(1秒間に5回弱)



センサーの情報を受け取るためアクティビティにリスナーをインプリメントします

```
public class sensorTest extends Activity implements SensorEventListener {
```

```
@Override
```


```
public void onSensorChanged(SensorEvent event) {  
}
```

```
@Override
```

```
public void onAccuracyChanged(Sensor sensor, int accuracy) {  
}
```

onSensorChangedがセンサーの加速度が更新された時、onAccuracyChangedが精度を変更した時のイベントです。

加速度が取得されるたびにonSensorChangedメソッドが走るようになります。



onSensorChangedメソッド内で3軸の加速度を取得します。必ず3軸全てを取得しなくてはならないわけではないので、必要に応じて取捨選択して下さい。加速度はfloat型で取得されます。

```
float sensorValueX = event.values[SensorManager.DATA_X];  
float sensorValueY = event.values[SensorManager.DATA_Y];  
float sensorValueZ = event.values[SensorManager.DATA_Z];
```

また、加速度の取得を停止したい場合は

```
sensorManager.unregisterListener(this);
```

でリスナーを解除します。

あとは取得した加速度情報をどのように使うかは貴方次第です。



ローパスフィルタとハイパスフィルタ

取得した加速度データは使用するアプリの目的に合わせて加工してあげる必要がある場合があります。

通常、加速度センサーから得られる値は端末を持つ手の震えや微妙な周りの振動などから極端に大きな値が得られることがあります。重力加速度や緩やかな加速度運動の影響を見たい場合、これらの高周波を除去し連続的な値に均す必要があります。

これをローパスフィルタといい実装にはいくつかの方法があり、次のページで代表的なものを解説します。

逆に瞬間的に加わる大きな加速度を得たい場合は(撃力など)、ハイパスフィルタを利用します。こちらは重力や慣性力など瞬間的に加わる大きな力に対して小さな力を除去します。

ハイパスフィルタはローパスフィルタとの組合せで実装されます。



ローパスフィルタの実装

- ・中央値を取る方法

特定のサンプリング回数を取得後、その中央値を加速度の代表値として採用します。例えば30回のサンプリングならば15番目に大きい値と16番目に大きい値の算術平均を代表値とします。なぜ平均値を使わないかということ、サンプルの中に極端に大きな値が含まれる場合は平均値の値に大きく影響を及ぼし代表値としての信頼度が失われるからです。

- ・直近に取得した加速度の値を少しずつ反映させる方法

例えば次の式の $LowX = x \times 10\% + lowX \times 90\%$

ように前回使用した値の90%と新しく得られた加速度の10%を加算して新しい加速度の値として採用します。このようにして極端に大きな値が得られた時の変化を緩和して連続的な加速度の値が得られるようにします。ここでは90%としましたが、直近の値を何%残すかはその時々状況によって最適な値を探する必要があります。



ハイパスフィルタの実装

ハイパスフィルタの実装はローパスフィルタの値を使います。

次の式のようにセンサーが感知した加速度からローパスフィルタをかけた値を引くことでハイパスフィルタの値を算出することができます。

$$\text{HighX} = x - \text{LowX};$$

これにより重力の影響などは除去され、瞬間的に作用した大きな加速度の変化だけが検出できます。