# Introduction to Android Magnetic Sensor

CS 436 Software Development on Mobile

**Dr.Paween Khoenkaw**

Department of Computer Science
Maejo University


# Magnetic field sensor

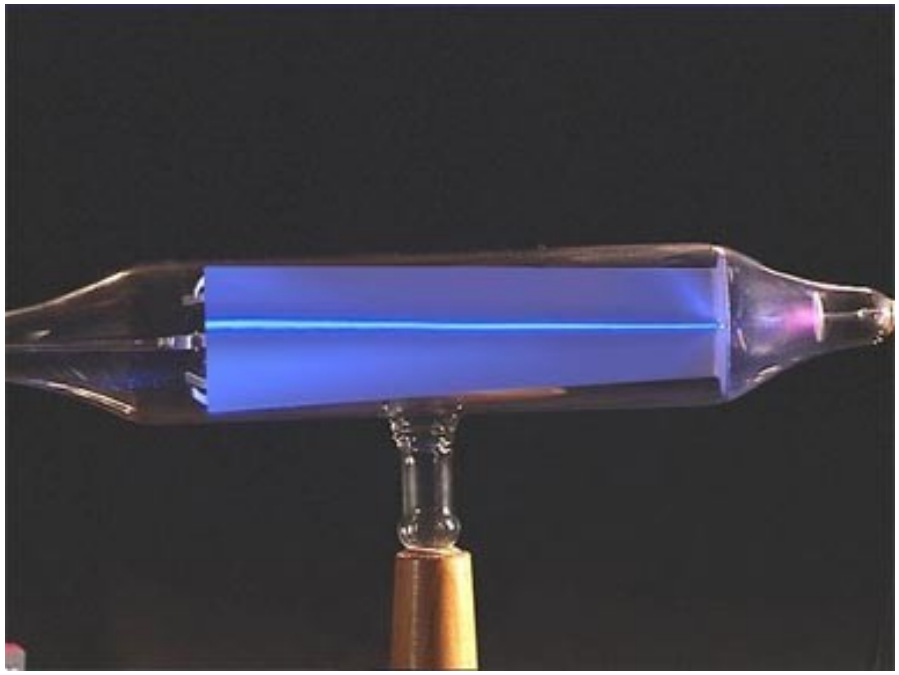
-Measure the magnetic field
-Detect all rotations
-Return data in SI uT

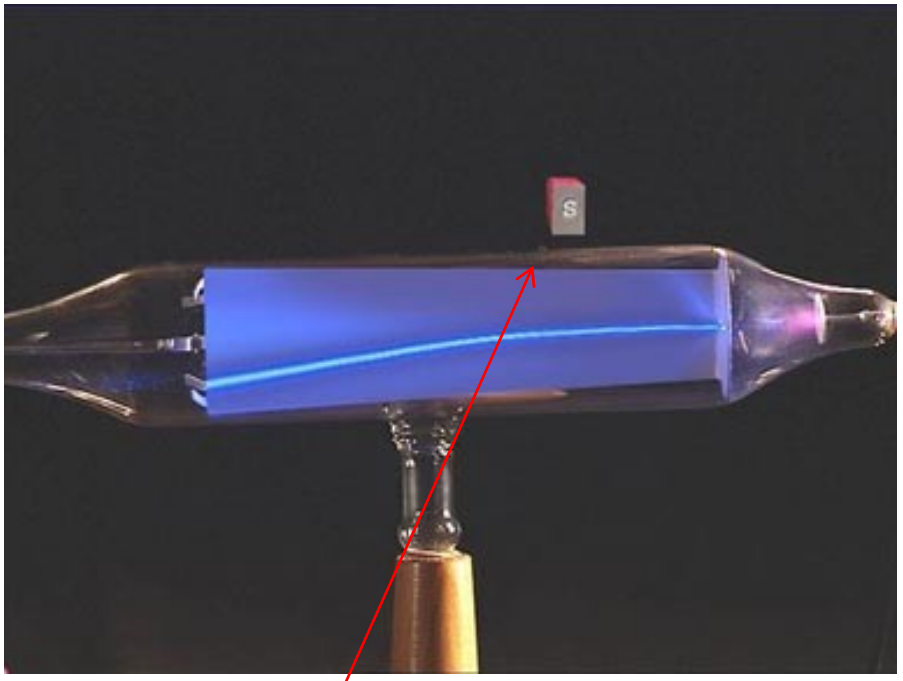
# Magnetic field sensor

## How magnetic field sensor works

# Magnetic field sensor

## Lorentz force






Magnet

http://lrrpublic.cli.det.nsw.edu.au/lrrSecure/Sites/Web/Forces_and_fields_creative_commons/7305/applets/viewer_magnets/viewer_magnets_text.htm

# Magnetic field sensor

## Hall Effect

V=0

Metal sheet

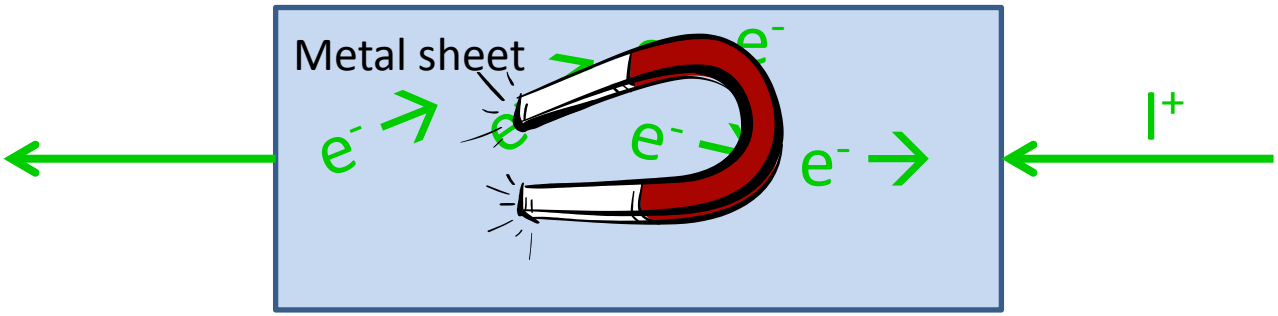
$e^-$ → $e^-$ → $e^-$ → $e^-$ → $e^-$ → $e^-$ →

$I^+$

V=IR (Ohm's law)

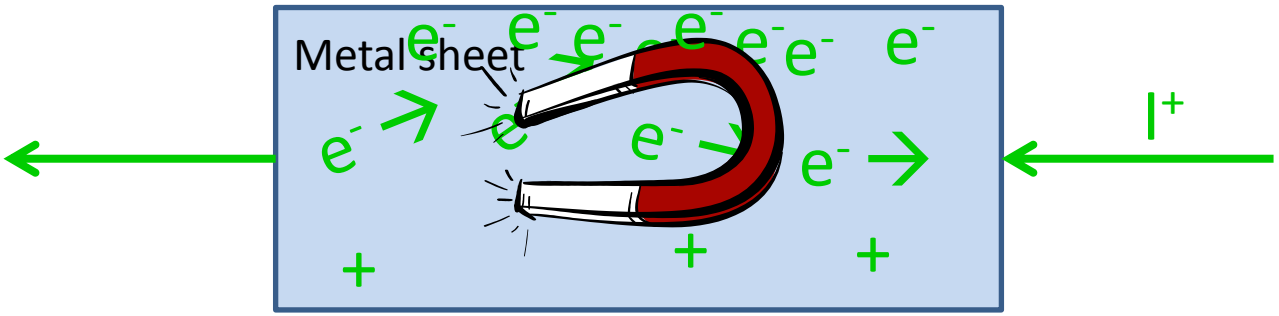
# Magnetic field sensor

## Hall Effect



# Magnetic field sensor

## Hall Effect



# Magnetic field sensor

## Hall Effect

$V=V_H$

Steady state (balance)

Metal sheet

$e^-$ $e^-$ $e^-$ $e^-$ $e^-$ $e^-$ $e^-$

$e^-$ $e^-$ $e^-$ $e^-$

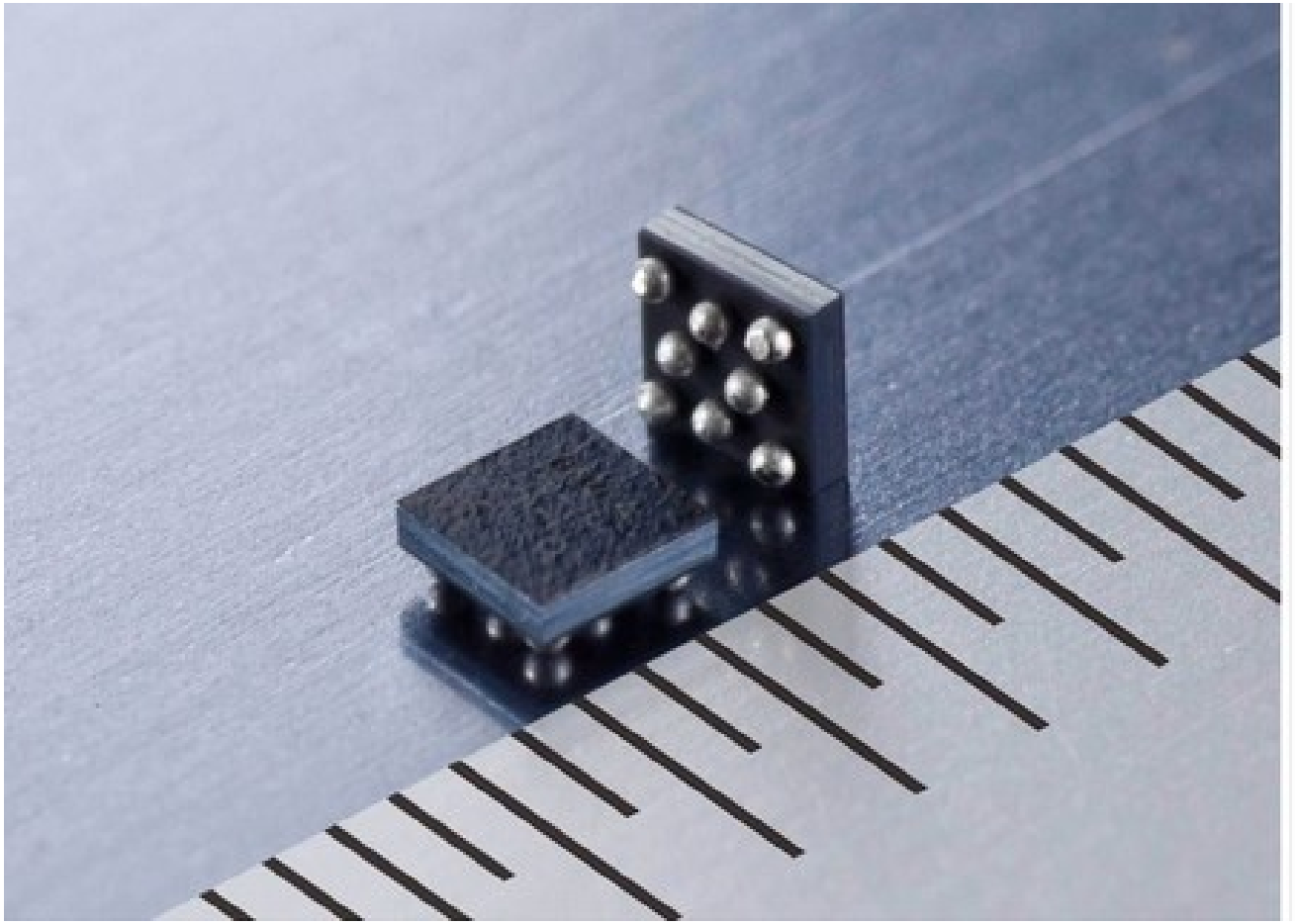
$I^+$

+ + + + + + +

$V=IR$ (Ohm's law)

# Magnetic field sensor

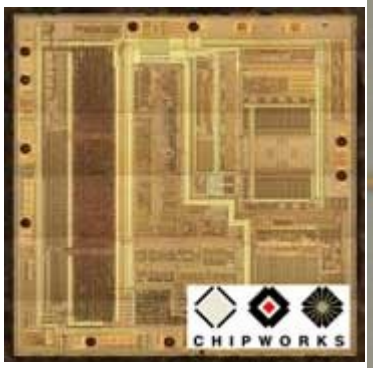
## Hall sensor

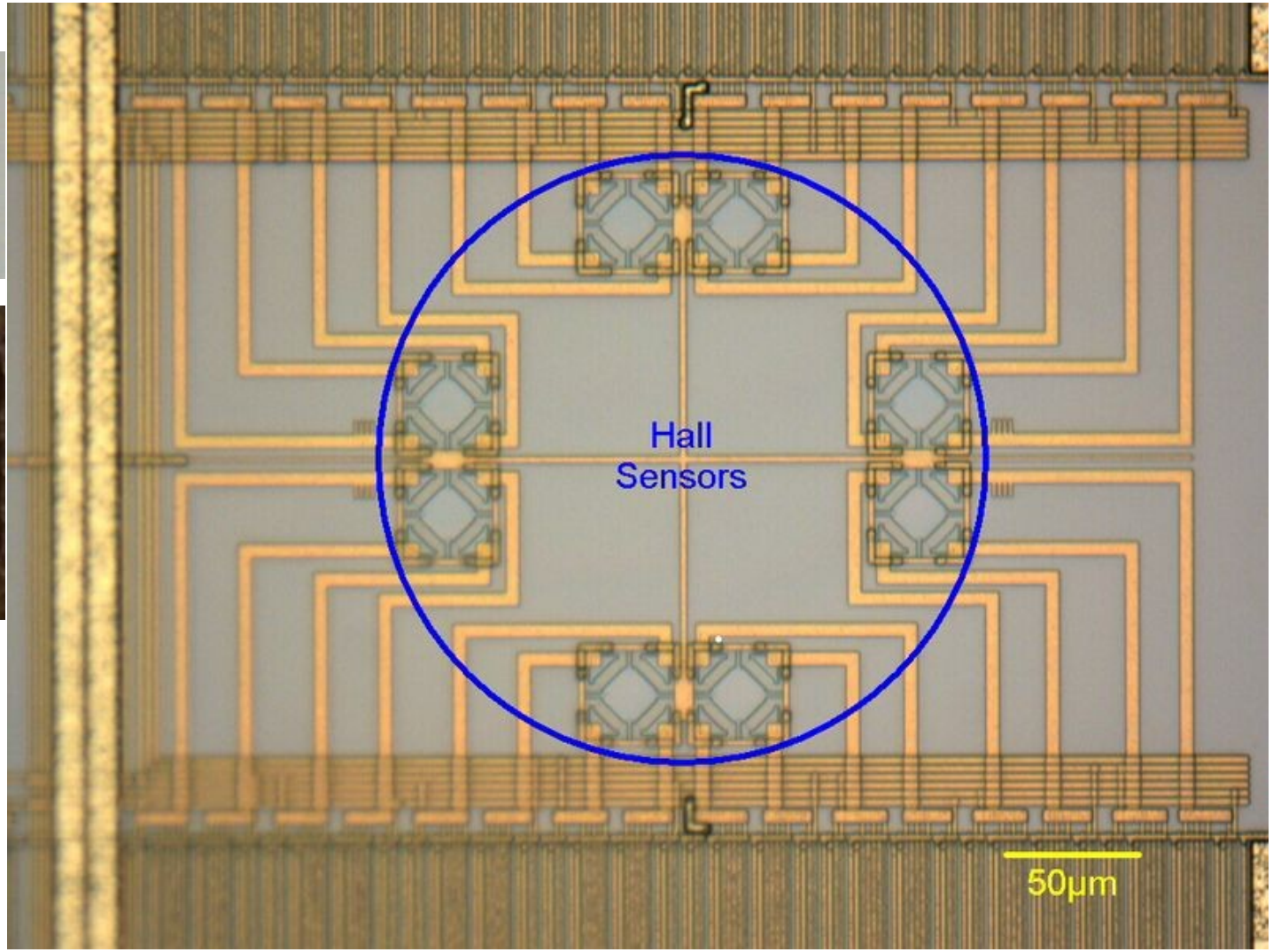


*YAS532 Tri-Axial Geomagnetic Sensor IC*

# Magnetic field sensor

## Hall sensor

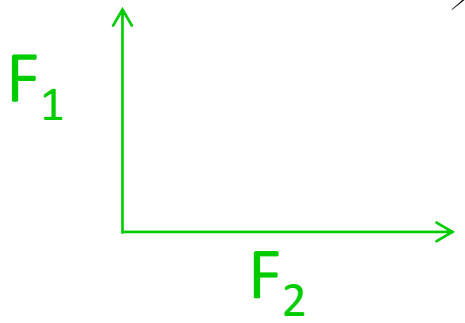
# Magnetic field sensor

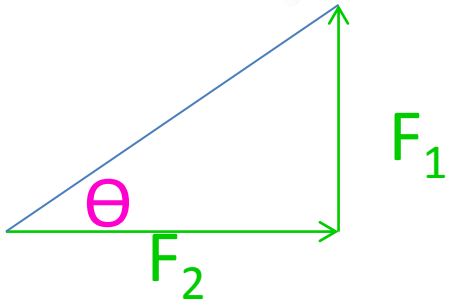
```
mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
mMagnetic = mSensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
```

# Magnetic field sensor

## Digital compass

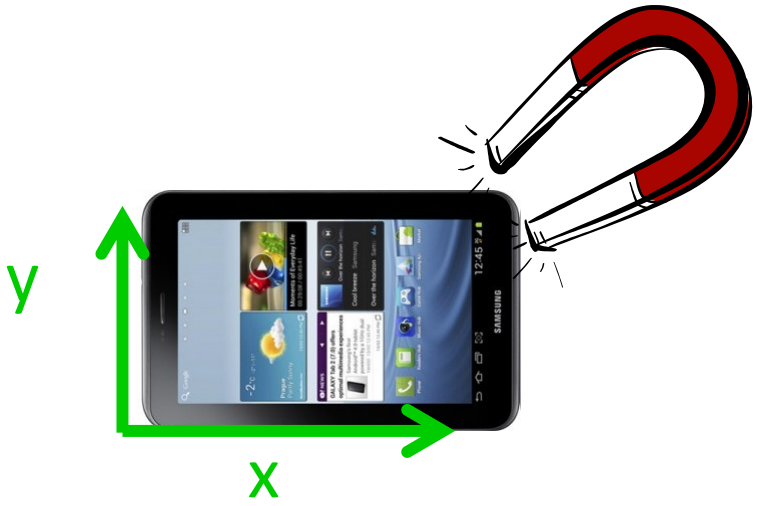
# Magnetic field sensor
# Digital compass

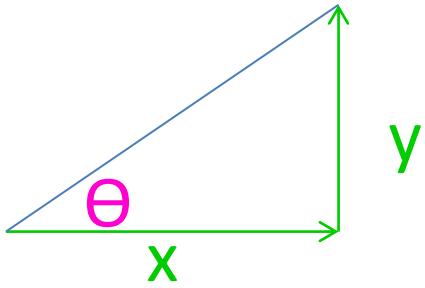



$$\theta = \tan^{-1}\left(\frac{F_1}{F_2}\right)$$

# Magnetic field sensor
# Digital compass

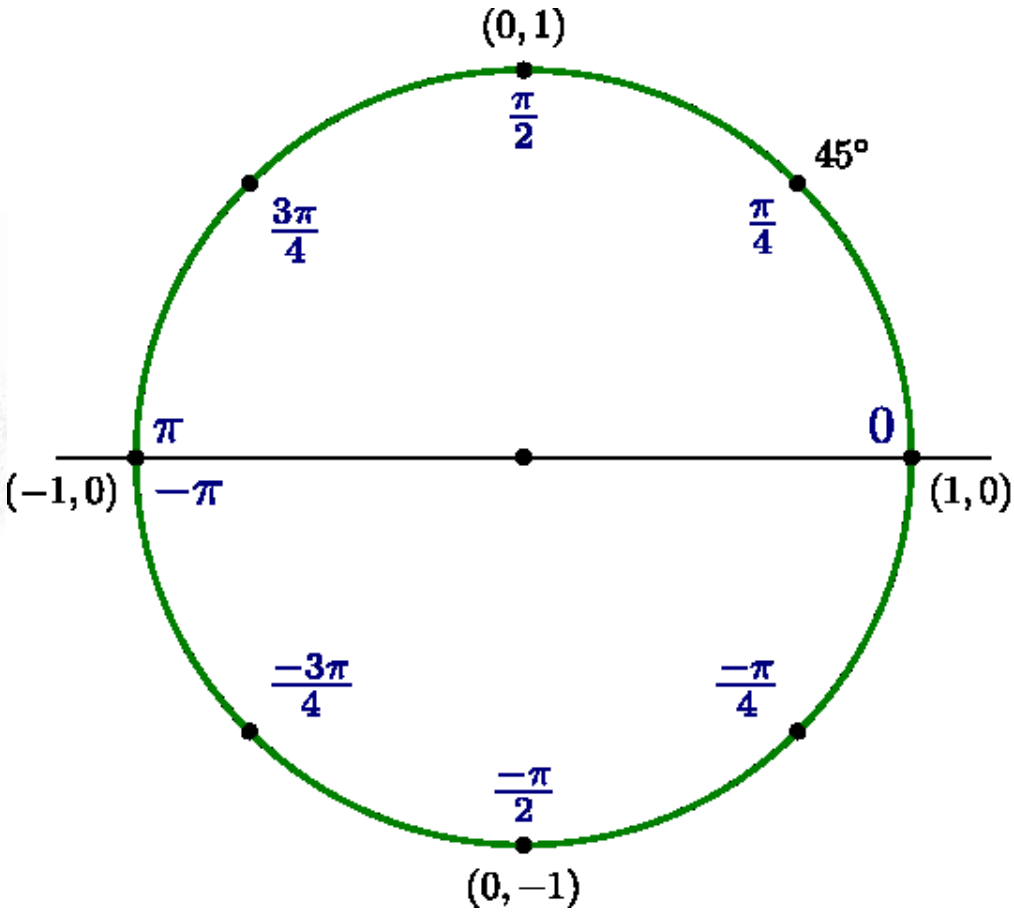



$$\theta = \tan^{-1}\left(\frac{y}{x}\right)$$

# Magnetic field sensor
# Digital compass


Compass




$\tan^{-1}(a/b)$

# Magnetic field sensor
# Digital compass


$$\theta = \tan^{-1}\left(\frac{y}{x}\right) - \frac{\pi}{2}$$


$$\theta = \tan^{-1}\left(\frac{x}{y}\right)$$

Lenovo K1 sensor configuration

# Magnetic field sensor
# Digital compass


$$\theta = \tan^{-1}\left(\frac{y}{x}\right) - \frac{\pi}{2}$$


$$\theta = \tan^{-1}\left(\frac{x}{y}\right) - \pi$$

Galaxy tab (ICS) sensor configuration

# Magnetic field sensor
# Digital compass

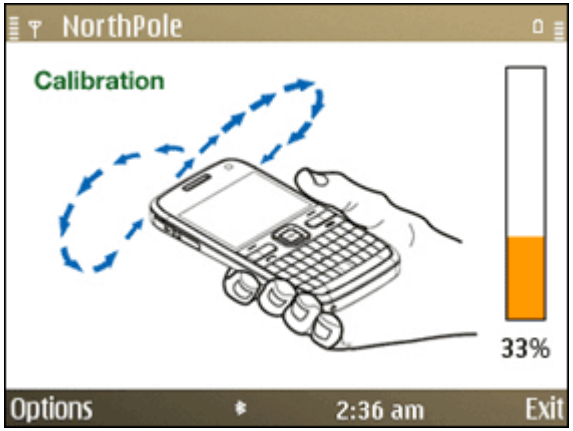

??????????


??????????

# Magnetic field sensor

## Calibration





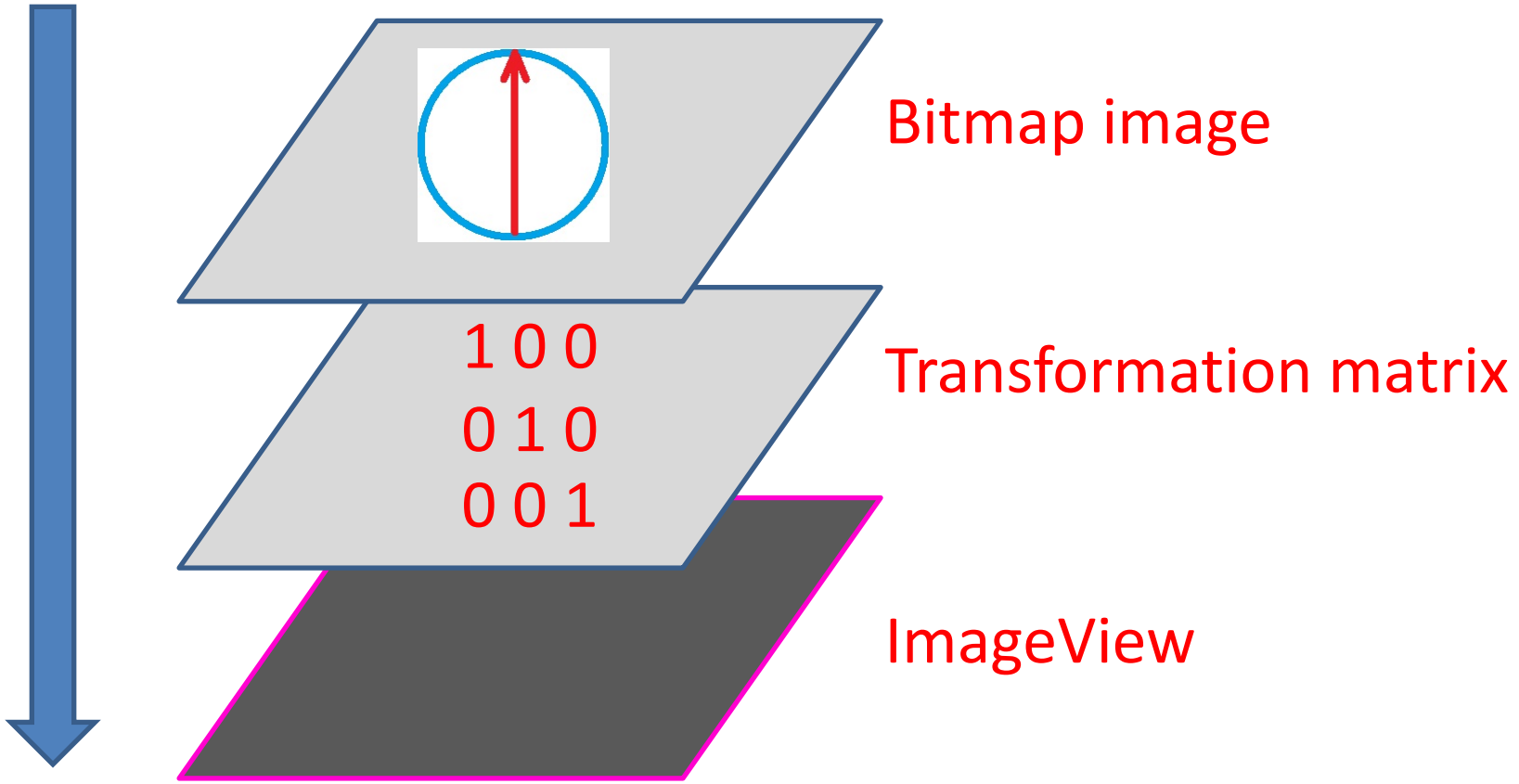

Why we use 8-pattern ?

# Magnetic field sensor
# Digital compass

How to rotate image in image view

# Magnetic field sensor

## Digital compass



# Magnetic field sensor

## Digital compass

$$\theta_{i+1} = \left(\theta_{i-1} - \theta_i\right) + k$$

Where k is an angle compensation

# Magnetic field sensor

## Layout part

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity"
   android:id="@+id/layout1"
    >
   <Button
      android:id="@+id/button1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentBottom="true"
      android:layout_alignParentLeft="true"
      android:layout_alignParentRight="true"
      android:text="Button" />
   <TextView
      android:id="@+id/textView1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_above="@+id/button1"
      android:layout_alignParentLeft="true"
      android:layout_alignParentRight="true"
      android:text="Large Text"

      android:textAppearance="?android:attr/text
AppearanceLarge" />
      <ImageView
         android:id="@+id/imageView1"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:layout_alignParentTop="true"
         android:layout_centerHorizontal="true"
         android:src="@drawable/compass" />
</RelativeLayout>
```

# Magnetic field sensor

## Initial imageview part

```
ImageView imageview1;
TextView textview1;
Button button1;
Matrix compass;
Bitmap bitmap1;
bitmap1 = BitmapFactory.decodeResource(getResources(),
                        R.drawable.compass);
                         compass=new Matrix();
imageview1.setScaleType(ScaleType.MATRIX);
layout1.setBackgroundColor(0xffffffff);
compass.postRotate((float) 0,bitmap1.getWidth()/2,bitmap1.getHeight()/2 );
imageview1.setImageMatrix(compass);
```

# Magnetic field sensor

## Apply roation part

```
double fx=event.values[0];
double fy=event.values[1];
double degree=0;
int orientation = getResources().getConfiguration().orientation;
 if(orientation==2)
 { // landscape
  degree=Math.toDegrees(Math.atan2(fy, fx))-90; // k1
 }else
 { // portrait
 // degree=Math.toDegrees(Math.atan2(fy, fx))-180; // samsung
   degree=Math.toDegrees(Math.atan2(fy, fx)); // k1
              }
compass.postRotate(old_degree-
(float)degree,bitmap1.getWidth()/2,bitmap1.getHeight()/2 );
imageview1.setImageMatrix(compass);
old_degree=(float)degree;
```

Thank you ☺