

توسعه برنامه های

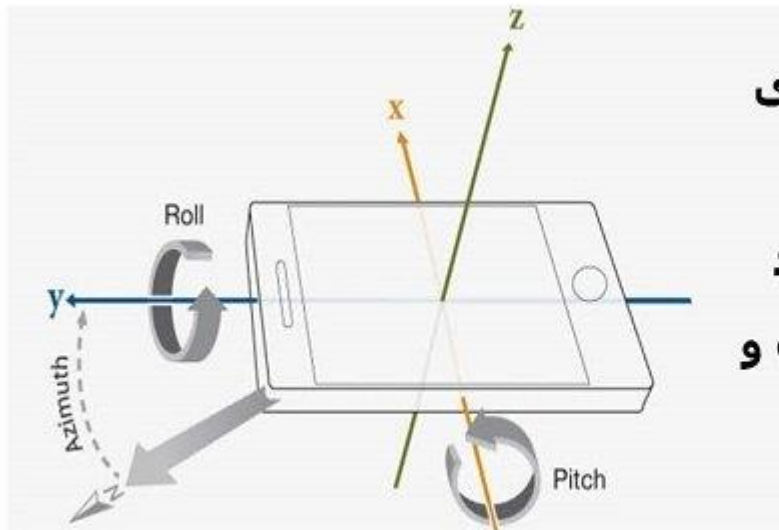
موبایل

جلسه دهم مجازی

بخش سوم

سحر صادقی

با سنسورهای گوشی اندروید کارهای منحصر به فرد انجام دهید



**با سنسورهای
گوشی
اندرویدی خود
کارهای جالب و
منحصر به فرد
انجام دهید**

بیشتر دستگاه‌های اندرویدی به‌ویژه آخرین مدل‌های معرفی شده دارای سنسورهای مختلف مانند سنسور روشنایی، سنسور جاذبه، سنسور مجاورت و ... هستند. اگر به‌طور مداوم از دستگاه اندرویدی خود استفاده می‌کنید حتماً با خیلی از این سنسورها و نوع کاری که انجام می‌دهند آشنایی دارید. البته احتمالاً از این سنسورها فقط برای یک‌سری از کارهای مشخص استفاده می‌کنید اما، نکته جالب این سنسورها آن است که می‌توانید برای انجام کارهای مختلف از آن‌ها استفاده کنید .

این‌که از این سنسورها برای خیلی کارهای دیگر هم می‌توان استفاده کرد، اما به‌عنوان کاربر اندروید باید بتوانید با این ابزار برای کاربردهای پیشرفته‌تر هم کار کنید. در این مطلب ده روش بسیار منحصر به‌فرد برای کار با سنسورهای به‌کار رفته روی گوشی اندرویدی معرفی شده که به‌کمک آن‌ها می‌توانید کارهای متنوعی انجام دهید. برای آشنایی با این راه‌ها مطلب زیر را مطالعه کنید.

روش هایی خاص برای استفاده منحصر به فرد از سنسورهای دستگاه اندروید

برای سنسور مجاورت:



(۱) از حرکات دست و سر خود برای کنترل پخش یک فایل صوتی یا فیلم بر روی دستگاه استفاده کنید. برای این منظور به سنسور مجاورت و اپلیکیشن Wave Control یا کنترل موج احتیاج دارید. از حالا به بعد، پخش موسیقی روی گوشی را با این روش جالب کنترل کنید. دست خود را روی سنسور مجاورت حرکت دهید تا قطعه موسیقی را عوض کنید یا پخش آن را متوقف کنید.

(۲) با سنسور مجاورت و به کمک یک اپلیکیشن به نام SpeakerPhone Ex به طور خودکار تماس های خود را پاسخ دهید. کافی است گوشی را نزدیک گوش تان ببرید تا تماس ها آماده پاسخ دهی شوند. علاوه بر آن، اگر کمی در این اپلیکیشن جستجو کنید لیست بلند بالا از تمام کارهایی که با حرکت دست و سر در ارتباط با تماس ها می توانید انجام دهید را مشاهده می کنید.

(۳) با تکان دادن دست روی سنسور مجاورت دستگاه تان را قفل یا قفل آن را باز کنید. برای این کار نیاز به اپلیکیشن Kin Screen دارید و با انجام تنظیمات داخل آن می توانید با حرکت دست کارهای زیادی انجام دهید.

(۴) اگر می خواهید برای کارهای شخصی حرکات جدیدی برای سنسور مجاورت تعریف کنید از اپلیکیشن Proximity Actions استفاده کنید. برای هر نوع کاری که می خواهید یک حرکت جدید تعریف کنید.

برای دوربین/فلش:

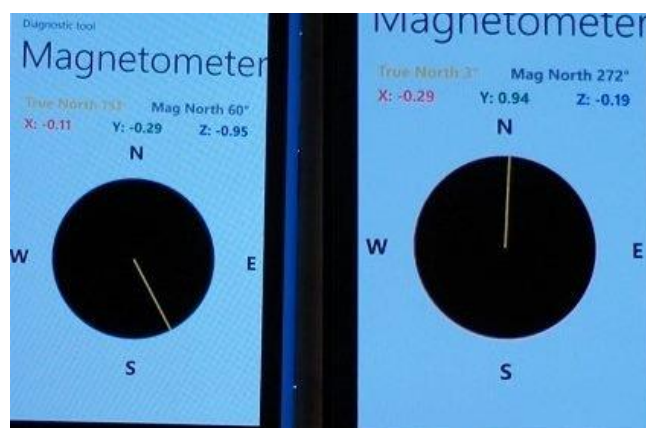


(۱) برای بررسی سریع و دقیق ضربان قلب از اپلیکیشن "کاردیوگرافی" استفاده کنید. برای این کار، انگشتان را روی دوربین اصلی و فلش گوشی قرار دهید. ممکن است در ابتدا دقت آن پایین باشد اما هرچه بیشتر با آن کار کنید مطمئناً نتایج جالب‌تری دست‌گیرتان می‌شود.

(۲) از طریق اپی به نام "فاصله هوشمند" (Smart Distance) "فاصله خود با اشیاء پیرامون‌تان را محاسبه کنید. این اپلیکیشن پیشرفته از داده‌های دوربین گوشی استفاده می‌کند تا فاصله دقیق تا اشیاء را محاسبه کند. ارتفاع تقریبی جسم موردنظر را در اپ وارد کنید و دوربین گوشی را به سمت آن بگیرید. این اپ ظرف تنها چند ثانیه فاصله شما با جسم موردنظر را اعلام می‌کند.

(۳) یکی از بهترین استفاده‌هایی که از یک گوشی اندروید قدیمی می‌توانید داشته باشید. به کمک اپلیکیشن Motion Detector، از دوربین گوشی اندروید قدیمی به عنوان شناسایی کننده حرکات یا برای زیرنظر داشتن استفاده کنید. این اپلیکیشن تصویر فرد مزاحم را به ایمیلی که برای آن تعریف شده ارسال می‌کند.

برای مغناطیس سنج:

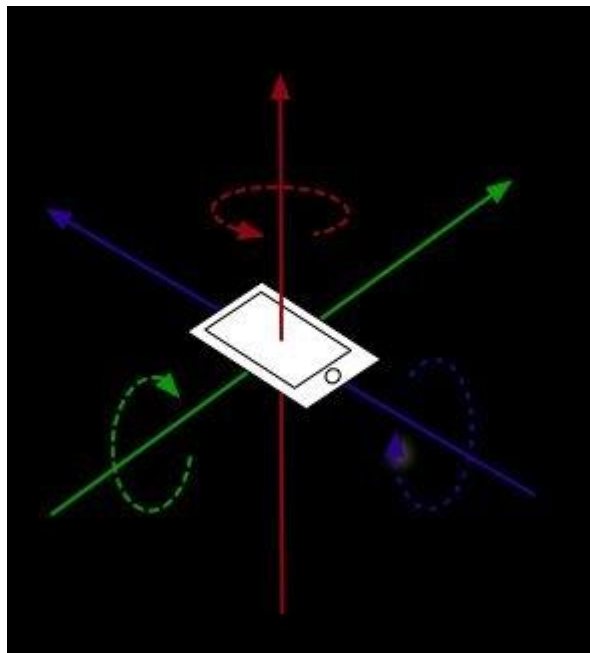


(۱) قطب‌نما از آن دسته وسایلی است که همیشه همراه‌تان ندارید اما دستگاه اندرویدی از آن دسته وسایلی است که همیشه همراه‌تان است. اگر زمانی نیاز به قطب‌نما داشتید دنبال خرید آن نباشید. زیرا گوشی اندروید به دلیل برخورداری از سنسور مغناطیس‌سنج و با نصب اپلیکیشنی به نام Smart Compass (قطب‌نمای هوشمند) نقش این وسیله را ایفا می‌کند.

برای شتاب‌سنج:

(۱) از دستگاه اندرویدی خود برای بررسی یا اندازه‌گیری سرعت دستگاه یا شتاب آن استفاده کنید. اپلیکیشن Accelerometer Gauge می‌تواند کاربردهای گوناگونی در این خصوص داشته باشد. به راحتی می‌توانید گوشی اندروید را به یک سرعت‌سنج تبدیل کنید.

برای ژيروسکوپ:



(۱) اپ Pucket Bubble Level به همراه سنسور ژيروسکوپ دستگاه شما به عنوان یک تراز‌سنج استفاده می‌شود. برای همین می‌توانید از دستگاه‌تان برای تعیین تراز پروژه‌ها یا اشیاء مختلف استفاده کنید. البته، ابتدا باید آن را کالیبره کنید و بعد برای هدفی که دارید از آن استفاده کنید.

آموزش خواندن بارکد در برنامه نویسی اندروید

در این سری از آموزش برنامه نویسی اندروید به آموزش خواندن بارکد در برنامه نویسی اندروید می‌پردازیم. برای هر بارکد یک result درون خود دارد یعنی هر بارکد می‌تواند یک url نیز در خود جای دهد و با اسکن آن یک سایت باز شود در ادامه با ما همراه باشید.

ابتدا وارد فایل Build.gradle شده در بخش dependencies خط زیر را اضافه کرده و پروژه را sync کنید.

```
1 compile 'com.google.android.gms:play-services-vision:9.6.1'
```

سپس خط زیر را به AndroidManifest.xml

```
1 <meta-data
2     android:name="com.google.android.gms.vision.DEPENDENCIES"
3     android:value="barcode"/>
```

حالا باید دسترسی و Feature را اضافه کنیم.

```
1 <uses-feature
2     android:name="android.hardware.camera"
3     android:required="true"/>
4 <uses-permission
5     android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

و کل کد AndroidManifest.xml همانند زیر می شود.

```
<?xml version="1.0" encoding="utf-8"?>
1 <manifest package="com.truiton.mobile.vision.qrcode"
2     xmlns:android="http://schemas.android.com/apk/res/android">
3
4     <uses-feature
5         android:name="android.hardware.camera"
6         android:required="true"/>
7     <uses-permission
8         android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
9     <application
10        android:allowBackup="true"
11        android:icon="@mipmap/ic_launcher"
12        android:label="@string/app_name"
13        android:supportsRtl="true"
14        android:theme="@style/AppTheme">
15        <meta-data
16            android:name="com.google.android.gms.vision.DEPENDENCIES"
17            android:value="barcode"/>
18        <activity android:name=".MainActivity">
19            <intent-filter>
20                <action android:name="android.intent.action.MAIN"/>
21
22                <category android:name="android.intent.category.LAUNCHER"/>
23            </intent-filter>
24        </activity>
25    </application>
26 </manifest>
```

```

        </intent-filter>
    </activity>
</application>

</manifest>

```

حالا یک فایل به نام activity_main.xml ایجاد کرده و کد زیر را در آن قرار دهید.

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
1   android:layout_width="match_parent"
2   android:layout_height="match_parent">
3
4
5
6
7   <TextView
8       android:id="@+id/scan_header"
9       android:layout_width="wrap_content"
10      android:layout_height="wrap_content"
11      android:layout_centerHorizontal="true"
12      android:text="Scan Results:"
13      android:textStyle="bold"/>
14
15
16
17
18   <TextView
19       android:id="@+id/scan_results"
20       android:layout_width="wrap_content"
21       android:layout_height="wrap_content"
22       android:layout_below="@+id/scan_header"
23       android:layout_centerHorizontal="true"
24       android:layout_marginTop="10dp"/>
25
26
27
28   <Button
29       android:id="@+id/button"
30       android:layout_width="wrap_content"
31       android:layout_height="wrap_content"
32       android:layout_alignParentBottom="true"
33       android:layout_centerHorizontal="true"
34       android:layout_marginBottom="10dp"
35       android:layout_marginTop="20dp"
36       android:text="Take Picture"/>

</RelativeLayout>

```

حالا یک فایل به نام MainActivity.java ایجاد کرده و کد زیر را در آن قرار دهید.

```

1 package ir.programchi;
2
3
4 import android.Manifest;
5 import android.content.Context;
6 import android.content.Intent;
7 import android.content.pm.PackageManager;
8 import android.graphics.Bitmap;
9 import android.graphics.BitmapFactory;
10 import android.net.Uri;
11 import android.os.Bundle;
12 import android.os.Environment;
13 import android.provider.MediaStore;
14 import android.support.annotation.NonNull;
15 import android.support.v4.app.ActivityCompat;
16 import android.support.v7.app.AppCompatActivity;
17 import android.util.Log;
18 import android.util.SparseArray;
19 import android.view.View;
20 import android.widget.Button;
21 import android.widget.TextView;
22 import android.widget.Toast;
23
24
25 import com.google.android.gms.vision.Frame;
26 import com.google.android.gms.vision.barcode.Barcode;
27 import com.google.android.gms.vision.barcode.BarcodeDetector;
28
29
30 import java.io.File;
31 import java.io.FileNotFoundException;
32
33 public class MainActivity extends AppCompatActivity {
34     private static final String LOG_TAG = "Barcode Scanner API";
35     private static final int PHOTO_REQUEST = 10;
36     private TextView scanResults;
37     private BarcodeDetector detector;
38     private Uri imageUri;
39     private static final int REQUEST_WRITE_PERMISSION = 20;
40     private static final String SAVED_INSTANCE_URI = "uri";
41     private static final String SAVED_INSTANCE_RESULT = "result";
42
43     @Override
44     protected void onCreate(Bundle savedInstanceState) {
45         super.onCreate(savedInstanceState);
46         setContentView(R.layout.activity_main);
47         Button button = (Button) findViewById(R.id.button);
48         scanResults = (TextView) findViewById(R.id.scan_results);
49         if (savedInstanceState != null) {
50             imageUri = Uri.parse(savedInstanceState.getString(SAVED_INSTANCE_URI));
51             scanResults.setText(savedInstanceState.getString(SAVED_INSTANCE_RESULT));
52         }
53     }
54 }

```



```

66     }
67     button.setOnClickListener(new View.OnClickListener() {
68         @Override
69         public void onClick(View view) {
70             ActivityCompat.requestPermissions(MainActivity.this, new
71                 String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE},
72                 REQUEST_WRITE_PERMISSION);
73         }
74     });
75
76     detector = new BarcodeDetector.Builder(getApplicationContext())
77         .setBarcodeFormats(Barcode.DATA_MATRIX | Barcode.QR_CODE)
78         .build();
79     if (!detector.isOperational()) {
80         scanResults.setText("Could not set up the detector!");
81         return;
82     }
83 }
84
85 @Override
86 public void onRequestPermissionsResult(int requestCode, @NonNull String[]
87 permissions, @NonNull int[] grantResults) {
88     super.onRequestPermissionsResult(requestCode, permissions, grantResults);
89     switch (requestCode) {
90         case REQUEST_WRITE_PERMISSION:
91             if (grantResults.length > 0 && grantResults[0] ==
92                 PackageManager.PERMISSION_GRANTED) {
93                 takePicture();
94             } else {
95                 Toast.makeText(MainActivity.this, "Permission Denied!",
96                     Toast.LENGTH_SHORT).show();
97             }
98         }
99     }
100
101     @Override
102     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
103         if (requestCode == PHOTO_REQUEST && resultCode == RESULT_OK) {
104             launchMediaScanIntent();
105             try {
106                 Bitmap bitmap = decodeBitmapUri(this, imageUri);
107                 if (detector.isOperational() && bitmap != null) {
108                     Frame frame = new Frame.Builder().setBitmap(bitmap).build();
109                     SparseArray<Barcode> barcodes = detector.detect(frame);
110                     for (int index = 0; index < barcodes.size(); index++) {
111                         Barcode code = barcodes.valueAt(index);
112                         scanResults.setText(scanResults.getText() + code.displayValue + "\n");
113                     }
114
115                     //Required only if you need to extract the type of barcode

```

```

11         int type = barcodes.valueAt(index).valueFormat;
5
11         switch (type) {
6             case Barcode.CONTACT_INFO:
11                 Log.i(LOG_TAG, code.contactInfo.title);
7                 break;
11             case Barcode.EMAIL:
8                 Log.i(LOG_TAG, code.email.address);
9                 break;
12             case Barcode.ISBN:
0                 Log.i(LOG_TAG, code.rawValue);
1                 break;
12             case Barcode.PHONE:
2                 Log.i(LOG_TAG, code.phone.number);
3                 break;
12             case Barcode.PRODUCT:
4                 Log.i(LOG_TAG, code.rawValue);
5                 break;
12             case Barcode.SMS:
6                 Log.i(LOG_TAG, code.sms.message);
7                 break;
12             case Barcode.TEXT:
8                 Log.i(LOG_TAG, code.rawValue);
9                 break;
13             case Barcode.URL:
0                 Log.i(LOG_TAG, "url: " + code.url.url);
1                 break;
13             case Barcode.WIFI:
2                 Log.i(LOG_TAG, code.wifi.ssid);
3                 break;
13             case Barcode.GEO:
4                 Log.i(LOG_TAG, code.geoPoint.lat + ":" + code.geoPoint.lng);
5                 break;
13             case Barcode.CALENDAR_EVENT:
6                 Log.i(LOG_TAG, code.calendarEvent.description);
7                 break;
13             case Barcode.DRIVER_LICENSE:
8                 Log.i(LOG_TAG, code.driverLicense.licenseNumber);
9                 break;
14             default:
0                 Log.i(LOG_TAG, code.rawValue);
1                 break;
14         }
2     }
14 }
3
14 if (barcodes.size() == 0) {
4     scanResults.setText("Scan Failed: Found nothing to scan");
14 }
5 } else {
14     scanResults.setText("Could not set up the detector!");
6 }

```

```

14         } catch (Exception e) {
7             Toast.makeText(this, "Failed to load Image", Toast.LENGTH_SHORT)
14                 .show();
8             Log.e(LOG_TAG, e.toString());
9         }
15     }
0 }
15 }
1 private void takePicture() {
15     Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
3     File photo = new File(Environment.getExternalStorageDirectory(),
4 "picture.jpg");
15     imageUri = Uri.fromFile(photo);
5     intent.putExtra(MediaStore.EXTRA_OUTPUT, imageUri);
15     startActivityForResult(intent, PHOTO_REQUEST);
6 }
7
15 @Override
8 protected void onSaveInstanceState(Bundle outState) {
9     if (imageUri != null) {
0         outState.putString(SAVED_INSTANCE_URI, imageUri.toString());
16         outState.putString(SAVED_INSTANCE_RESULT,
1 scanResults.getText().toString());
16     }
2     super.onSaveInstanceState(outState);
3 }
16 }
4
16 private void launchMediaScanIntent() {
5     Intent mediaScanIntent = new
16 Intent(Intent.ACTION_MEDIA_SCANNER_SCAN_FILE);
6     mediaScanIntent.setData(imageUri);
7     this.sendBroadcast(mediaScanIntent);
16 }
8
16 private Bitmap decodeBitmapUri(Context ctx, Uri uri) throws
9 FileNotFoundException {
17     int targetW = 600;
1     int targetH = 600;
17     BitmapFactory.Options bmOptions = new BitmapFactory.Options();
2     bmOptions.inJustDecodeBounds = true;
17     BitmapFactory.decodeStream(ctx.getContentResolver().openInputStream(uri),
3 null, bmOptions);
17     int photoW = bmOptions.outWidth;
5     int photoH = bmOptions.outHeight;
17
6     int scaleFactor = Math.min(photoW / targetW, photoH / targetH);
17     bmOptions.inJustDecodeBounds = false;
7     bmOptions.inSampleSize = scaleFactor;
8

```

```
17     return BitmapFactory.decodeStream(ctx.getContentResolver()  
9         .openInputStream(uri), null, bmOptions);  
18 }  
18 }  
1  
18  
2  
18  
3  
18  
4  
18  
5  
18  
6  
18  
7  
18  
8  
18  
9  
19  
0
```