



布袋风管



管道规范



射频功率放大



摄像机教程



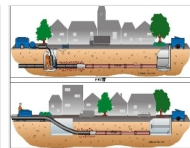
模拟检测



家用投影机



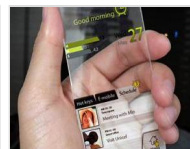
架构师



管道非开挖修



拍照手机排行



透明手机大小

等级: **BLU74**

排名: 第18832名

原创: 146篇

转载: 19篇

译文: 4篇

评论: 9条

文章分类

CSDN日报20170707——《稀缺：百分之二的选择》

征文 | 你会为 AI 转型么？

每周荐书 | Android、Keras、ES6（评论送书）

Android模拟实现检测心率变化

标签: android android应用

2016-11-15 19:13

1122人阅读

评论(0)

分类: android (20) j2se (5)

目录(?)

[+]

当今，市面上有了一些可以通过**Android**应用来检测病人心率，血压，体温，等等，一系列方便人们日常生活的

Android手机应用。那么，这些实用的手机应用程序是怎么做出来的呢？那么

的应用，那就是**android**上模拟实现检测心率的变化。我利用Android模拟实现

血管的变化来检测心率的功能。哇哦，听起来是不是很高大上呢？瞬间对这个

木有呢？！哈哈，那就让我们一起来实现这些功能吧。

一、原理

首先我们还是要讲讲这个应用的原理吧，在下认为，要做一个Android应用程序，不然，看了半天各位都不知道这个应用是基于什么原理做的呢。是吧，那就让

通过摄像头来获得心率，搜了一下这个技术真不是噱头，据说在iPhone早有实现的，连手机的闪光灯也会被自动打开，用户将手指放在摄像头上时，指尖皮下血管由于有



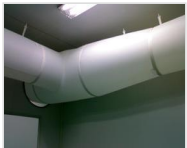
目录视图

摘要视图

RSS 订阅

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



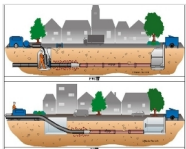
模拟检测



家用投影机



架构师



管道非开挖修



拍照手机排行



透明手机多小

图 (7)

spring (19)

springmvc (20)

动态规划 (6)

设计模式 (22)

中间件 (1)

网络 (1)

友情链接

(红色的深度)会有轻微的变化。这个过程可以凭借感光元件捕捉到。这样毛细血管的搏动就能通过画面明度的周期性变化反映出来。

好了，原理说完了，大家有木有看懂呢？

二、实现

1、创建图像处理类ImageProcessing

这个类主要提供处理图像本身的方法。

具体实现如下：

```
[java]
01. package com.lyz.monitor.utils;
02.
03. /**
04.  * 图像处理类
05.  * @author liuyazhuang
06.  *
07.  */
08. public abstract class ImageProcessing {
09.
10.     /**
11.      * 内部调用的处理图片的方法
12.      * @param yuv420sp
13.      * @param width
14.      * @param height
15.      * @return
16.      */
17.     private static int decodeYUV420SPtoRedSum(byte[] yuv4
18.         if (yuv420sp == null)
19.             return 0;
20.         final int frameSize = width * height;
21.         int sum = 0;
22.         for (int j = 0, yp = 0; j < height; j++) {
23.             int uvp = frameSize + (j >> 1) * width, u = 0
```

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



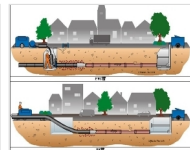
模拟检测



家用投影机



架构师



管道非开挖修



拍照手机排行



透明手机

小程序员在android移动端 (1711)

先验概率与后验概率的区 (1400)

Android手机在开发调试 (1293)

AbstractTableModel的介 (1168)

Android模拟实现检测心 (1121)

Swing组件布局setLayout (1118)

Bad XML block :header (987)

Android模拟实现检测心率变化 - sjyttkl的专栏 - CSDN博客

```

24.
25.     for (int i = 0; i < width; i++, yp++) {
26.         int y = (0xff & ((int) yuv420sp[yp])) - 16;
27.         if (y < 0)
28.             y = 0;
29.         if ((i & 1) == 0) {
30.             v = (0xff & yuv420sp[uvp++]) - 128;
31.             u = (0xff & yuv420sp[uvp++]) - 128;
32.         }
33.         int y1192 = 1192 * y;
34.         int r = (y1192 + 1634 * v);
35.         int g = (y1192 - 833 * v - 400 * u);
36.         int b = (y1192 + 2066 * u);
37.
38.         if (r < 0)
39.             r = 0;
40.         else if (r > 262143)
41.             r = 262143;
42.         if (g < 0)
43.             g = 0;
44.         else if (g > 262143)
45.             g = 262143;
46.         if (b < 0)
47.             b = 0;
48.         else if (b > 262143)
49.             b = 262143;
50.
51.         int pixel = 0xff000000 | ((r << 6) & 0xff
52.             | ((g >> 2) & 0xff00) | ((b >> 10)
53.             & 0xff0000);
54.         sum += red;
55.     }
56.     return sum;
57. }
58.
59. /**
60.  * 对外开放的图像处理方法
61.  * @param yuv420sp
62.  * @param width

```

关闭

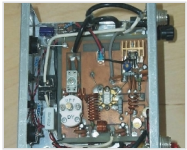




布袋风管



管道规范



射频功率放大



摄像机教程



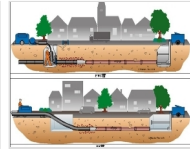
模拟检测



家用投影机



架构师



管道非开挖修



拍照手机排行



透明手机多小

应的VVC技术

* 细说反射, Java 和 Android 开发者必须跨越的坎

* 深度学习 | 反向传播与它的直观理解

* ArcGIS 水文分析实战教程——雨量计算与流量统计

* 每周荐书: Android、Keras、ES6 (评论送书)

```
63.     * @param height
64.     * @return
65.     */
66.     public static int decodeYUV420SPtoRedAvg(byte[] yuv420sp, int width,
67.         int height) {
68.         if (yuv420sp == null)
69.             return 0;
70.         final int frameSize = width * height;
71.         int sum = decodeYUV420SPtoRedSum(yuv420sp, width, height);
72.         return (sum / frameSize);
73.     }
74. }
```

2、MainActivity实现

为了简单,我没有单独新建别的类来分解这些功能,我直接在MainActivity中实现了这些功能,那么我们是如何一步步实现的吧。

(1)程序中用到的属性

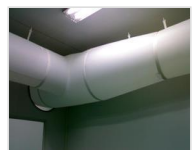
首先,我们来看看程序中定义了哪些属性字段,来实现这些功能吧。

具体属性字段如下所示:

```
[java]
01. //曲线
02. private Timer timer = new Timer();
03. //Timer任务,与Timer配套使用
04. private TimerTask task;
05. private static int gx;
06. private static int j;
07.
08. private static double flag=1;
09. private Handler handler;
10. private String title = "pulse";
11. private XYSeries series;
```

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



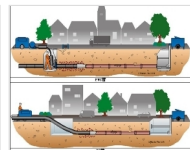
模拟检测



家用投影机



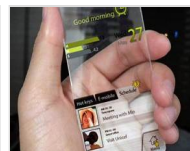
架构师



管道非开挖修



拍照手机排行



透明手机

```

12. private XYMultipleSeriesDataset mDataset;
13. private GraphicalView chart;
14. private XYMultipleSeriesRenderer renderer;
15. private Context context;
16. private int addX = -1;
17. double addY;
18. int[] xv = new int[300];
19. int[] yv = new int[300];
20. int[] hua=new int[]{9,10,11,12,13,14,13,12,11,10,9,8,7,6,7,8,9,10,11,10,10};
21.
22. // private static final String TAG = "HeartRateMonitor";
23. private static final AtomicBoolean processing = new AtomicBoolean(false);
24. //Android手机预览控件
25. private static SurfaceView preview = null;
26. //预览设置信息
27. private static SurfaceHolder previewHolder = null;
28. //Android手机相机句柄
29. private static Camera camera = null;
30. //private static View image = null;
31. private static TextView text = null;
32. private static TextView text1 = null;
33. private static TextView text2 = null;
34. private static WakeLock wakeLock = null;
35. private static int averageIndex = 0;
36. private static final int averageArraySize = 4;
37. private static final int[] averageArray = new int[averageArraySize];
38. //设置默认类型
39. private static TYPE currentType = TYPE.GREEN;
40. //获取当前类型
41. public static TYPE getCurrent() {
42.     return currentType;
43. }
44. //心跳下标值
45. private static int beatsIndex = 0;
46. //心跳数组的大小
47. private static final int beatsArraySize = 3;
48. //心跳数组
49. private static final int[] beatsArray = new int[beatsArraySize];
50. //心跳脉冲

```

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



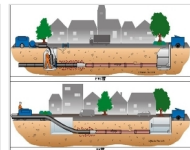
模拟检测



家用投影机



架构师



管道非开挖修



拍照手机排行



透明手机

```
51. private static double beats = 0;
52. //开始时间
53. private static long startTime = 0;
```

咋一看，是不是很多？有木有一种头晕乎乎的赶脚呢？没关系，通过后面具体的功能实现，相信大家能弄明白每个属性字段的作用与含义的。不要掉队，继续认真向下看哦。

(2)定义枚举类型来标识当前颜色

颜色类型，我在这里用一个枚举类型来定义，这个枚举类型很简单，只有两种颜色，一种是绿色，一种是红色。默认颜色为绿色。

具体实现的代码如下：

```
[java]
01. /**
02.  * 类型枚举
03.  * @author liuyazhuang
04.  *
05.  */
06. public static enum TYPE {
07.     GREEN, RED
08. };
09. //设置默认类型
10. private static TYPE currentType = TYPE.GREEN;
11. //获取当前类型
12. public static TYPE getCurrent() {
13.     return currentType;
14. }
```

(3)初始化配置方法initConfig

这个方法总体上的功能是初始化程序的各个配置，包括调用其他方法，例如应用程序启动后显示的样式，调用相机，通过Handler接收其他方法传递过来的

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



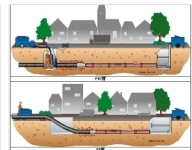
模拟检测



家用投影机



架构师



管道非开挖修



拍照手机排行



透明手机多小

现应用的配置功能，同时这个方法相当于一个应用程序的管家，它来直接或间接的调用其他方法，来使整个应用程序顺利运行起来。

具体代码实现如下:

```
[java]
01.  /**
02.     * 初始化配置
03.     */
04.     private void initConfig() {
05.         //曲线
06.         context = getApplicationContext();
07.
08.         //这里获得main界面上的布局，下面会把图表画在这个布局里面
09.         LinearLayout layout = (LinearLayout)findViewById(R.id.linearLayout1);
10.
11.         //这个类用来放置曲线上的所有点，是一个点的集合，根据这些点画出曲线
12.         series = new XYSeries(title);
13.
14.         //创建一个数据集的实例，这个数据集将被用来创建图表
15.         mDataset = new XYMultipleSeriesDataset();
16.
17.         //将点集添加到这个数据集中
18.         mDataset.addSeries(series);
19.
20.         //以下都是曲线的样式和属性等等的设置，renderer相当于一个片
21.         int color = Color.GREEN;
22.         PointStyle style = PointStyle.CIRCLE;
23.         renderer = buildRenderer(color, style, true);
24.
25.         //设置好图表的样式
26.         setChartSettings(renderer, "X", "Y", 0, 300, 4, 1);
27.
28.         //生成图表
29.         chart = ChartFactory.getLineChartView(context, mDataset, renderer);
30.
31.         //将图表添加到布局中去
```

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



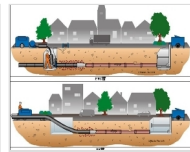
模拟检测



家用投影机



架构师



管道非开挖修



拍照手机排行



透明手机多小

```

32.         layout.addView(chart, new LayoutParams(LayoutParams.FILL_PARENT, LayoutParams.FILL
33.
34.         //这里的Handler实例将配合下面的Timer实例，完成定时更新图表的功能
35.         handler = new Handler() {
36.             @Override
37.             public void handleMessage(Message msg) {
38.                 //          刷新图表
39.                 updateChart();
40.                 super.handleMessage(msg);
41.             }
42.         };
43.
44.         task = new TimerTask() {
45.             @Override
46.             public void run() {
47.                 Message message = new Message();
48.                 message.what = 1;
49.                 handler.sendMessage(message);
50.             }
51.         };
52.
53.         timer.schedule(task, 1, 20);           //曲线
54.         //获取SurfaceView控件
55.         preview = (SurfaceView) findViewById(R.id.preview);
56.         previewHolder = preview.getHolder();
57.         previewHolder.addCallback(surfaceCallback);
58.         previewHolder.setType(SurfaceHolder.SURFACE_TYPE_
59.         //          image = findViewById(R.id.image);
60.         text = (TextView) findViewById(R.id.text);
61.         text1 = (TextView) findViewById(R.id.text1);
62.         text2 = (TextView) findViewById(R.id.text2);
63.         PowerManager pm = (PowerManager) getSystemService
64.         wakeLock = pm.newWakeLock(PowerManager.FULL_WAKE_
65.     }

```

(4)创建图表的方法buildRenderer

关闭

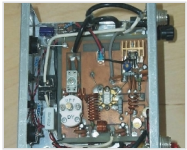




布袋风管



管道规范



射频功率放大



摄像机教程



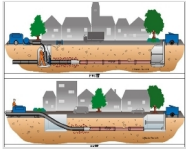
模拟检测



家用投影机



架构师



管道非开挖修



拍照手机排行



透明手机

这个方法主要是利用了第三方的类库类实现创建图标的操作。

具体代码实现如下

```
[java]
01.  /**
02.      * 创建图表
03.      * @param color
04.      * @param style
05.      * @param fill
06.      * @return
07.      */
08.  protected XYMultipleSeriesRenderer buildRenderer(int color, PointStyle style
09.      XYMultipleSeriesRenderer renderer = new XYMultipleSeriesRenderer();
10.
11.      //设置图表中曲线本身的样式，包括颜色、点的大小以及线的粗细等
12.      XYSeriesRenderer r = new XYSeriesRenderer();
13.      r.setColor(Color.RED);
14.      // r.setPointStyle(null);
15.      // r.setFillPoints(fill);
16.      r.setLineWidth(1);
17.      renderer.addSeriesRenderer(r);
18.      return renderer;
19.  }
```

(5)设置图表的样式方法setChartSettings

这个方法主要是对(4)中创建的图表，进行样式的设置。

具体代码如下：

```
[java]
01.  /**
02.      * 设置图标的样式
03.      * @param renderer
```

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



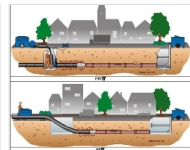
模拟检测



家用投影机



架构师



管道非开挖修



拍照手机排行



透明手机多小

```

04.      * @param xTitle : x标题
05.      * @param yTitle : y标题
06.      * @param xMin : x最小长度
07.      * @param xMax : x最大长度
08.      * @param yMin : y最小长度
09.      * @param yMax : y最大长度
10.      * @param axesColor : 颜色
11.      * @param labelsColor : 标签
12.      */
13.      protected void setChartSettings(XYMultipleSeriesRenderer renderer, String xTitle, Str:
14.          double xMin, double xMax, double yMin, double yMax, int axesColor, int labelsC
15.          //有关对图表的渲染可参看api文档
16.          renderer.setChartTitle(title);
17.          renderer.setXTitle(xTitle);
18.          renderer.setYTitle(yTitle);
19.          renderer.setXAxisMin(xMin);
20.          renderer.setXAxisMax(xMax);
21.          renderer.setYAxisMin(yMin);
22.          renderer.setYAxisMax(yMax);
23.          renderer.setAxesColor(axesColor);
24.          renderer.setLabelsColor(labelsColor);
25.          renderer.setShowGrid(true);
26.          renderer.setGridColor(Color.GREEN);
27.          renderer.setXLabels(20);
28.          renderer.setYLabels(10);
29.          renderer.setXTitle("Time");
30.          renderer.setYTitle("mmHg");
31.          renderer.setYLabelsAlign(Align.RIGHT);
32.          renderer.setPointSize((float) 3 );
33.          renderer.setShowLegend(false);
34.      }

```

(6)更新图表updateChart

这个方法主要实现了对图表中曲线图的更新绘制，同时检测手机摄像头感应的提示“请用您的指尖盖住摄像头镜头”的信息来提示用户。动态的更新绘制曲线

关闭

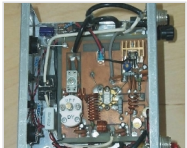




布袋风管



管道规范



射频功率放大



摄像机教程



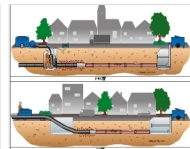
模拟检测



家用投影机



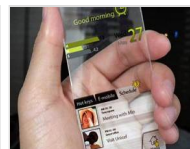
架构师



管道非开挖修



拍照手机排行



透明手机

具体代码实现如下：

```
[java]
01.  /**
02.     * 更新图标信息
03.     */
04.     private void updateChart() {
05.
06.         //设置好下一个需要增加的节点
07.         if(flag==1)
08.             addY=10;
09.         else{
10.             //
11.             addY=250;
12.             flag=1;
13.             if(gx<200){
14.                 if(hua[20]>1){
15.                     Toast.makeText(MainActivity.this, "请用您的指尖盖住摄像头
16. 头!", Toast.LENGTH_SHORT).show();
17.                     hua[20]=0;}
18.                     hua[20]++;
19.                     return;}
20.                 else
21.                     hua[20]=10;
22.                     j=0;
23.             }
24.             if(j<20){
25.                 addY=hua[j];
26.                 j++;
27.             }
28.             //移除数据集中旧的点集
29.             mDataset.removeSeries(series);
30.
31.             //判断当前点集中到底有多少点，因为屏幕总共只能容纳100个，所
32.             int length = series.getItemCount();
33.             int bz=0;
34.             //addX = length;
```

关闭

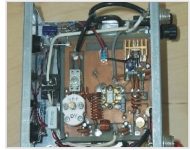




布袋风管



管道规范



射频功率放大



摄像机教程



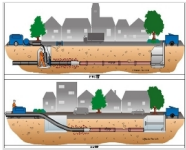
模拟检测



家用投影机



架构师



管道非开挖修



拍照手机排行



透明手机

```
35.         if (length > 300) {
36.             length = 300;
37.             bz=1;
38.         }
39.         addX = length;
40.         //将旧的点集中x和y的数值取出来放入backup中，并且将x的值加1，造成曲线向右平移的效果
41.         for (int i = 0; i < length; i++) {
42.             xv[i] = (int) series.getX(i) -bz;
43.             yv[i] = (int) series.getY(i);
44.         }
45.
46.         //点集先清空，为了做成新的点集而准备
47.         series.clear();
48.         mDataset.addSeries(series);
49.         //将新产生的点首先加入到点集中，然后在循环体中将坐标变换后的一系列点都重新加入到点集中
50.         //这里可以试验一下把顺序颠倒过来是什么效果，即先运行循环体，再添加新产生的点
51.         series.add(addX, addY);
52.         for (int k = 0; k < length; k++) {
53.             series.add(xv[k], yv[k]);
54.         }
55.         //在数据集中添加新的点集
56.         //mDataset.addSeries(series);
57.
58.         //视图更新，没有这一步，曲线不会呈现动态
59.         //如果在非UI主线程中，需要调用postInvalidate()，具体参考...
60.         chart.invalidate();
61.     } //曲线
```

(7)相机预览回调方法previewCallback

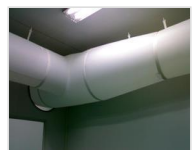
这个方法中实现动态更新界面UI的功能，通过获取手机摄像头的参数来实时动态计算心率值。

具体代码实现如下：

```
[java]
01.  /**
```

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



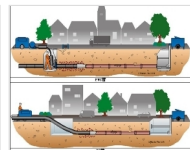
模拟检测



家用投影机



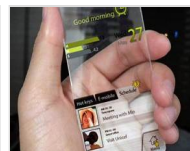
架构师



管道非开挖修



拍照手机排行



透明手机多小

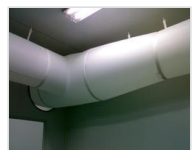
```

02.
03.
04.
05.
06.
07.
08.
09.
10.
11.
12.
13.
14.
15.
16.
17.
18.
19.
20.
21.
22.
23.
24.
25.
26.
27.
28.
29.
30.
31.
32.
33.
34.
35.
36.
37.
38.
39.
    * 相机预览方法
    * 这个方法中实现动态更新界面UI的功能,
    * 通过获取手机摄像头的参数来实时动态计算平均像素值、脉冲数,从而实时动态计算心率值。
    */
    private static PreviewCallback previewCallback = new PreviewCallback() {
        public void onPreviewFrame(byte[] data, Camera cam) {
            if (data == null)
                throw new NullPointerException();
            Camera.Size size = cam.getParameters().getPreviewSize();
            if (size == null)
                throw new NullPointerException();
            if (!processing.compareAndSet(false, true))
                return;
            int width = size.width;
            int height = size.height;
            //图像处理
            int imgAvg = ImageProcessing.decodeYUV420SPtoRedAvg(data.clone(), height, width);
            text1.setText("平均像素值是" + String.valueOf(imgAvg));
            //像素平均值imgAvg,日志
            //Log.i(TAG, "imgAvg=" + imgAvg);
            if (imgAvg == 0 || imgAvg == 255) {
                processing.set(false);
                return;
            }
            //计算平均值
            int averageArrayAvg = 0;
            int averageArrayCnt = 0;
            for (int i = 0; i < averageArray.length; i++)
                if (averageArray[i] > 0) {
                    averageArrayAvg += averageArray[i];
                    averageArrayCnt++;
                }
            //计算平均值
            int rollingAverage = (averageArrayCnt > 0)?
                (averageArrayAvg/averageArrayCnt):0;
            TYPE newType = currentType;
            if (imgAvg < rollingAverage) {

```

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



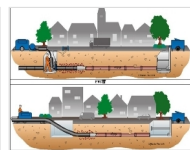
模拟检测



家用投影机



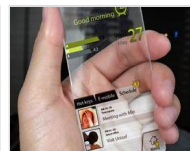
架构师



管道非开挖修



拍照手机排行



透明手机

```

40.         newType = TYPE.RED;
41.         if (newType != currentType) {
42.             beats++;
43.             flag=0;
44.             text2.setText("脉冲数是"+String.valueOf(beats));
45.             //Log.e(TAG, "BEAT!! beats=" + beats);
46.         }
47.     } else if (imgAvg > rollingAverage) {
48.         newType = TYPE.GREEN;
49.     }
50.
51.     if (averageIndex == averageArraySize)
52.         averageIndex = 0;
53.     averageArray[averageIndex] = imgAvg;
54.     averageIndex++;
55.
56.     // Transitioned from one state to another to the same
57.     if (newType != currentType) {
58.         currentType = newType;
59.         //image.postInvalidate();
60.     }
61.     //获取系统结束时间 (ms)
62.     long endTime = System.currentTimeMillis();
63.     double totalTimeInSecs = (endTime - startTime) / 1000d;
64.     if (totalTimeInSecs >= 2) {
65.         double bps = (beats / totalTimeInSecs);
66.         int dpm = (int) (bps * 60d);
67.         if (dpm < 30 || dpm > 180 || imgAvg < 200) {
68.             //获取系统开始时间 (ms)
69.             startTime = System.currentTimeMillis();
70.             //beats心跳总数
71.             beats = 0;
72.             processing.set(false);
73.             return;
74.         }
75.         //Log.e(TAG, "totalTimeInSecs=" + totalTimeInSecs);
76.         if (beatsIndex == beatsArraySize)
77.             beatsIndex = 0;
78.         beatsArray[beatsIndex] = dpm;

```

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



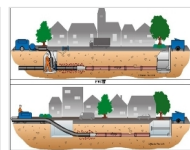
模拟检测



家用投影机



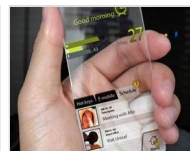
架构师



管道非开挖修



拍照手机排行



透明手机

```

79.         beatsIndex++;
80.         int beatsArrayAvg = 0;
81.         int beatsArrayCnt = 0;
82.         for (int i = 0; i < beatsArray.length; i++) {
83.             if (beatsArray[i] > 0) {
84.                 beatsArrayAvg += beatsArray[i];
85.                 beatsArrayCnt++;
86.             }
87.         }
88.         int beatsAvg = (beatsArrayAvg / beatsArrayCnt);
89.         text.setText("您的心率
是"+String.valueOf(beatsAvg)+" zhi:"+String.valueOf(beatsArray.length)
+" "+String.valueOf(beatsIndex)+" "+String.valueOf
90.         //获取系统时间 (ms)
91.         startTime = System.currentTimeMillis();
92.         beats = 0;
93.     }
94.     processing.set(false);
95. }
96. }
97. };

```

(8)SurfaceHolder.Callback

这个方法主要是相机摄像头，捕捉信息改变时调用。

具体代码实现如下：

```

[java]
01. /**
02.     * 预览回调接口
03.     */
04.     private static SurfaceHolder.Callback surfaceCallback
05.         //创建时调用
06.         @Override
07.         public void surfaceCreated(SurfaceHolder holder)
08.             try {

```

关闭

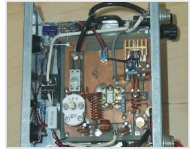




布袋风管



管道规范



射频功率放大



摄像机教程



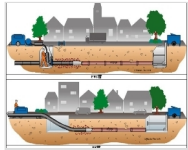
模拟检测



家用投影机



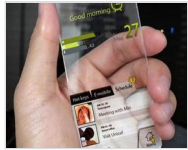
架构师



管道非开挖修



拍照手机排行



透明手机

```
09.         camera.setPreviewDisplay(previewHolder);
10.         camera.setPreviewCallback(previewCallback);
11.     } catch (Throwable t) {
12.         Log.e("PreviewDemo-
surfaceCallback", "Exception in setPreviewDisplay()", t);
13.     }
14. }
15. //当预览改变的时候回调此方法
16. @Override
17. public void surfaceChanged(SurfaceHolder holder, int format, int width, int height) {
18.     Camera.Parameters parameters = camera.getParameters();
19.     parameters.setFlashMode(Camera.Parameters.FLASH_MODE_TORCH);
20.     Camera.Size size = getSmallestPreviewSize(width, height, parameters)
21.     if (size != null) {
22.         parameters.setPreviewSize(size.width, size.height);
23.         // Log.d(TAG, "Using width=" + size.width + " height
24.     }
25.     camera.setParameters(parameters);
26.     camera.startPreview();
27. }
28. //销毁的时候调用
29. @Override
30. public void surfaceDestroyed(SurfaceHolder holder) {
31.     // Ignore
32. }
33. };
```

(9)获取相机最小的预览尺寸方法getSmallestPreviewSize

这个方法的功能是获取当前手机相机最小的预览尺寸。

具体代码实现如下:

```
[java]
01. /**
02.     * 获取相机最小的预览尺寸
```

关闭

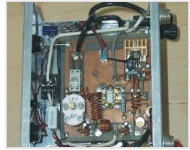




布袋风管



管道规范



射频功率放大



摄像机教程



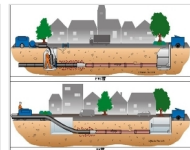
模拟检测



家用投影机



架构师



管道非开挖修



拍照手机排行



透明手机多小

```

03.      * @param width
04.      * @param height
05.      * @param parameters
06.      * @return
07.      */
08.      private static Camera.Size getSmallestPreviewSize(int width, int height,
09.          Camera.Parameters parameters) {
10.          Camera.Size result = null;
11.          for (Camera.Size size : parameters.getSupportedPreviewSizes()) {
12.              if (size.width <= width && size.height <= height) {
13.                  if (result == null) {
14.                      result = size;
15.                  } else {
16.                      int resultArea = result.width * result.height;
17.                      int newArea = size.width * size.height;
18.                      if (newArea < resultArea)
19.                          result = size;
20.                  }
21.              }
22.          }
23.          return result;
24.      }

```

(10)onCreate方法

这个方法是Android原生自带的方法，通常在这个方法中我们会实现页面控件。我们这个项目中，主要是设置要显示的UI和调用initConfig方法来启动应用程序运行。

具体实现代码如下：

```

[java]
01.      @Override
02.      public void onCreate(Bundle savedInstanceState) {
03.          super.onCreate(savedInstanceState);
04.          setContentView(R.layout.activity_main);

```

关闭

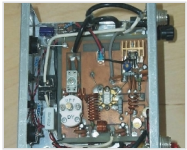




布袋风管



管道规范



射频功率放大



摄像机教程



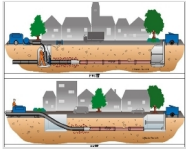
模拟检测



家用投影机



架构师



管道非开挖修



拍照手机排行



透明手机

```
05.         initConfig();
06.     }
```

(11)其他一些Android原生自带方法

```
[java]

01.  @Override
02.  public void onDestroy() {
03.      //当结束程序时关掉Timer
04.      timer.cancel();
05.      super.onDestroy();
06.  };
07.  @Override
08.  public void onConfigurationChanged(Configuration newConfig) {
09.      super.onConfigurationChanged(newConfig);
10.  }
11.
12.  @Override
13.  public void onResume() {
14.      super.onResume();
15.      wakeLock.acquire();
16.      camera = Camera.open();
17.      startTime = System.currentTimeMillis();
18.  }
19.
20.  @Override
21.  public void onPause() {
22.      super.onPause();
23.      wakeLock.release();
24.      camera.setPreviewCallback(null);
25.      camera.stopPreview();
26.      camera.release();
27.      camera = null;
28.  }
```

(12)MainActivity完整代码

关闭





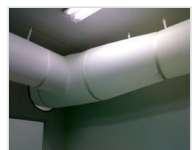
最后我还是给出MainActivity的完整代码吧，大家根据上面的分析仔细阅读几遍，便会体会这其中的奥妙了。嘿嘿，加油哦！

[java]

```
01. package com.lyz.xinlv.activity;
02. import java.util.Timer;
03. import java.util.TimerTask;
04. import java.util.concurrent.atomic.AtomicBoolean;
05.
06. import org.achartengine.ChartFactory;
07. import org.achartengine.GraphicalView;
08. import org.achartengine.chart.PointStyle;
09. import org.achartengine.model.XYMultipleSeriesDataset;
10. import org.achartengine.model.XYSeries;
11. import org.achartengine.renderer.XYMultipleSeriesRenderer;
12. import org.achartengine.renderer.XYSeriesRenderer;
13.
14. import android.app.Activity;
15. import android.content.Context;
16. import android.content.res.Configuration;
17. import android.graphics.Color;
18. import android.graphics.Paint.Align;
19. import android.hardware.Camera;
20. import android.hardware.Camera.PreviewCallback;
21. import android.os.Bundle;
22. import android.os.Handler;
23. import android.os.Message;
24. import android.os.PowerManager;
25. import android.os.PowerManager.WakeLock;
26. import android.util.Log;
27. import android.view.SurfaceHolder;
28. import android.view.SurfaceView;
29. import android.view.ViewGroup.LayoutParams;
30. import android.widget.LinearLayout;
31. import android.widget.TextView;
32. import android.widget.Toast;
33.
34. import com.lyz.monitor.utils.ImageProcessing;
```

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



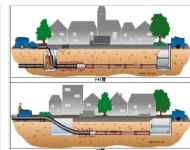
模拟检测



家用投影机



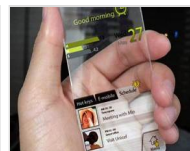
架构师



管道非开挖修



拍照手机排行



透明手机多小

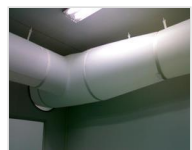
```

35.
36.  /**
37.  * 程序的主入口
38.  * @author liuyazhuang
39.  *
40.  */
41. public class MainActivity extends Activity {
42.     //曲线
43.     private Timer timer = new Timer();
44.     //Timer任务,与Timer配套使用
45.     private TimerTask task;
46.     private static int gx;
47.     private static int j;
48.
49.     private static double flag=1;
50.     private Handler handler;
51.     private String title = "pulse";
52.     private XYSeries series;
53.     private XYMultipleSeriesDataset mDataset;
54.     private GraphicalView chart;
55.     private XYMultipleSeriesRenderer renderer;
56.     private Context context;
57.     private int addX = -1;
58.     double addY;
59.     int[] xv = new int[300];
60.     int[] yv = new int[300];
61.     int[] hua=new int[] {9,10,11,12,13,14,13,12,11,10,9,8,
62.
63.     // private static final String TAG = "HeartRateMonitor";
64.     private static final AtomicBoolean processing = new AtomicBoolean();
65.     //Android手机预览控件
66.     private static SurfaceView preview = null;
67.     //预览设置信息
68.     private static SurfaceHolder previewHolder = null;
69.     //Android手机相机句柄
70.     private static Camera camera = null;
71.     //private static View image = null;
72.     private static TextView text = null;
73.     private static TextView text1 = null;

```

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



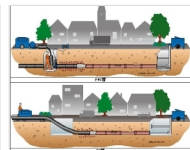
模拟检测



家用投影机



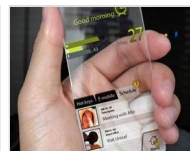
架构师



管道非开挖修



拍照手机排行



透明手机

```

74. private static TextView text2 = null;
75. private static WakeLock wakeLock = null;
76. private static int averageIndex = 0;
77. private static final int averageArraySize = 4;
78. private static final int[] averageArray = new int[averageArraySize];
79.
80. /**
81.  * 类型枚举
82.  * @author liuyazhuang
83.  *
84.  */
85. public static enum TYPE {
86.     GREEN, RED
87. };
88. //设置默认类型
89. private static TYPE currentType = TYPE.GREEN;
90. //获取当前类型
91. public static TYPE getCurrent() {
92.     return currentType;
93. }
94. //心跳下标值
95. private static int beatsIndex = 0;
96. //心跳数组的大小
97. private static final int beatsArraySize = 3;
98. //心跳数组
99. private static final int[] beatsArray = new int[beatsArraySize];
100. //心跳脉冲
101. private static double beats = 0;
102. //开始时间
103. private static long startTime = 0;
104.
105. @Override
106. public void onCreate(Bundle savedInstanceState) {
107.     super.onCreate(savedInstanceState);
108.     setContentView(R.layout.activity_main);
109.     initConfig();
110. }
111.
112. /**

```

关闭





布袋风管

管道规范



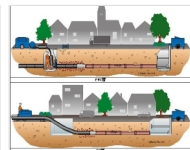
射频功率放大

摄像机教程



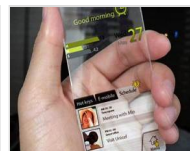
模拟检测

家用投影机



架构师

管道非开挖修



拍照手机排行

透明手机多小

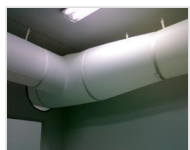
```

113. * 初始化配置
114. */
115. private void initConfig() {
116.     //曲线
117.     context = getApplicationContext();
118.
119.     //这里获得main界面上的布局，下面会把图表画在这个布局里面
120.     LinearLayout layout = (LinearLayout)findViewById(R.id.linearLayout1);
121.
122.     //这个类用来放置曲线上的所有点，是一个点的集合，根据这些点画出曲线
123.     series = new XYSeries(title);
124.
125.     //创建一个数据集的实例，这个数据集将被用来创建图表
126.     mDataset = new XYMultipleSeriesDataset();
127.
128.     //将点集添加到这个数据集中
129.     mDataset.addSeries(series);
130.
131.     //以下都是曲线的样式和属性等等的设置，renderer相当于一个用来给图表做渲染的句柄
132.     int color = Color.GREEN;
133.     PointStyle style = PointStyle.CIRCLE;
134.     renderer = buildRenderer(color, style, true);
135.
136.     //设置好图表的样式
137.     setChartSettings(renderer, "X", "Y", 0, 300, 4, 100, 0, 100, 0, 100);
138.
139.     //生成图表
140.     chart = ChartFactory.getLineChartView(context, mDataset, renderer);
141.
142.     //将图表添加到布局中去
143.     layout.addView(chart, new LayoutParams(LayoutParams.MATCH_PARENT, LayoutParams.MATCH_PARENT));
144.
145.     //这里的Handler实例将配合下面的Timer实例，完成定时更新图表
146.     handler = new Handler() {
147.         @Override
148.         public void handleMessage(Message msg) {
149.             //刷新图表
150.             updateChart();
151.             super.handleMessage(msg);

```

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



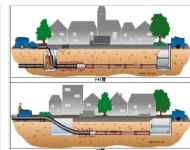
模拟检测



家用投影机



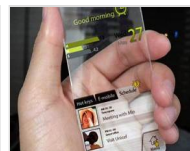
架构师



管道非开挖修



拍照手机排行



透明手机多小

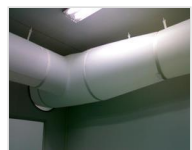
```

152.     }
153. };
154.
155. task = new TimerTask() {
156.     @Override
157.     public void run() {
158.         Message message = new Message();
159.         message.what = 1;
160.         handler.sendMessage(message);
161.     }
162. };
163.
164. timer.schedule(task, 1, 20);           //曲线
165. //获取SurfaceView控件
166. preview = (SurfaceView) findViewById(R.id.preview);
167. previewHolder = preview.getHolder();
168. previewHolder.addCallback(surfaceCallback);
169. previewHolder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);
170. // image = findViewById(R.id.image);
171. text = (TextView) findViewById(R.id.text);
172. text1 = (TextView) findViewById(R.id.text1);
173. text2 = (TextView) findViewById(R.id.text2);
174. PowerManager pm = (PowerManager) getSystemService(Context.POWER_SERVICE);
175. wakeLock = pm.newWakeLock(PowerManager.FULL_WAKE_LOCK, "DoNotDimScreen");
176. }
177.
178. // 曲线
179. @Override
180. public void onDestroy() {
181.     //当结束程序时关掉Timer
182.     timer.cancel();
183.     super.onDestroy();
184. };
185.
186. /**
187.  * 创建图表
188.  * @param color
189.  * @param style
190.  * @param fill

```

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



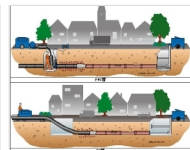
模拟检测



家用投影机



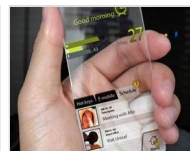
架构师



管道非开挖修



拍照手机排行



透明手机多小

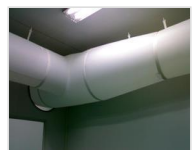
```

191.      * @return
192.      */
193.      protected XYMultipleSeriesRenderer buildRenderer(int color, PointStyle style, boolean
194.          XYMultipleSeriesRenderer renderer = new XYMultipleSeriesRenderer();
195.
196.          //设置图表中曲线本身的样式, 包括颜色、点的大小以及线的粗细等
197.          XYSeriesRenderer r = new XYSeriesRenderer();
198.          r.setColor(Color.RED);
199.          // r.setPointStyle(null);
200.          // r.setFillPoints(fill);
201.          r.setLineWidth(1);
202.          renderer.addSeriesRenderer(r);
203.          return renderer;
204.      }
205.
206.      /**
207.       * 设置图标样式
208.       * @param renderer
209.       * @param xTitle : x标题
210.       * @param yTitle : y标题
211.       * @param xMin : x最小长度
212.       * @param xMax : x最大长度
213.       * @param yMin : y最小长度
214.       * @param yMax : y最大长度
215.       * @param axesColor : 颜色
216.       * @param labelsColor : 标签
217.       */
218.      protected void setChartSettings(XYMultipleSeriesRende
219.          double xMin, double xMax, double yMin, double
220.          //有关对图表的渲染可参看api文档
221.          renderer.setChartTitle(title);
222.          renderer.setXTitle(xTitle);
223.          renderer.setYTitle(yTitle);
224.          renderer.setXAxisMin(xMin);
225.          renderer.setXAxisMax(xMax);
226.          renderer.setYAxisMin(yMin);
227.          renderer.setYAxisMax(yMax);
228.          renderer.setAxesColor(axesColor);
229.          renderer.setLabelsColor(labelsColor);

```

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



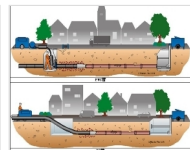
模拟检测



家用投影机



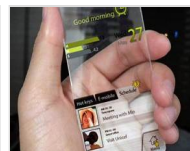
架构师



管道非开挖修



拍照手机排行



透明手机多小

```

230.         renderer.setShowGrid(true);
231.         renderer.setGridColor(Color.GREEN);
232.         renderer.setXLabels(20);
233.         renderer.setYLabels(10);
234.         renderer.setXTitle("Time");
235.         renderer.setYTitle("mmHg");
236.         renderer.setYLabelsAlign(Align.RIGHT);
237.         renderer.setPointSize((float) 3 );
238.         renderer.setShowLegend(false);
239.     }

240.
241.     /**
242.      * 更新图标信息
243.      */
244.     private void updateChart() {
245.
246.         //设置好下一个需要增加的节点
247.         if(flag==1)
248.             addY=10;
249.         else{
250.             //
251.             addY=250;
252.             flag=1;
253.             if(gx<200){
254.                 if(hua[20]>1){
255.                     Toast.makeText(MainActivity.this, "请屏住你的指尖并缓慢深呼吸!", Toast.LENGTH_SHORT).show();
256.                     hua[20]=0;
257.                     hua[20]++;
258.                     return;}
259.                 else
260.                     hua[20]=10;
261.                 j=0;
262.             }
263.             if(j<20){
264.                 addY=hua[j];
265.                 j++;
266.             }
267.

```

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



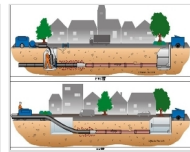
模拟检测



家用投影机



架构师



管道非开挖修



拍照手机排行



透明手机

```

268. //移除数据集中旧的点集
269. mDataset.removeSeries(series);
270.
271. //判断当前点集中到底有多少点，因为屏幕总共只能容纳100个，所以当点数超过100时，长度永远是100
272. int length = series.getItemCount();
273. int bz=0;
274. //addX = length;
275. if (length > 300) {
276.     length = 300;
277.     bz=1;
278. }
279. addX = length;
280. //将旧的点集中x和y的数值取出来放入backup中，并且将x的值加1，造成曲线向右平移的效果
281. for (int i = 0; i < length; i++) {
282.     xv[i] = (int) series.getX(i) -bz;
283.     yv[i] = (int) series.getY(i);
284. }
285.
286. //点集先清空，为了做成新的点集而准备
287. series.clear();
288. mDataset.addSeries(series);
289. //将新产生的点首先加入到点集中，然后在循环体中将坐标变换后的一系列点都重新加入到点集中
290. //这里可以试验一下把顺序颠倒过来是什么效果，即先运行循环体，再添加新产生的点
291. series.add(addX, addY);
292. for (int k = 0; k < length; k++) {
293.     series.add(xv[k], yv[k]);
294. }
295. //在数据集中添加新的点集
296. //mDataset.addSeries(series);
297.
298. //视图更新，没有这一步，曲线不会呈现动态
299. //如果在非UI主线程中，需要调用postInvalidate()，具体参考
300. chart.invalidate();
301. } //曲线
302.
303.
304. @Override
305. public void onConfigurationChanged(Configuration newC
306.     super.onConfigurationChanged(newConfig);

```

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



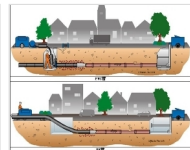
模拟检测



家用投影机



架构师



管道非开挖修



拍照手机排行

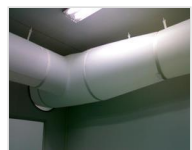


透明手机

```
307.     }
308.
309.     @Override
310.     public void onResume() {
311.         super.onResume();
312.         wakeLock.acquire();
313.         camera = Camera.open();
314.         startTime = System.currentTimeMillis();
315.     }
316.
317.     @Override
318.     public void onPause() {
319.         super.onPause();
320.         wakeLock.release();
321.         camera.setPreviewCallback(null);
322.         camera.stopPreview();
323.         camera.release();
324.         camera = null;
325.     }
326.
327.     /**
328.      * 相机预览方法
329.      * 这个方法中实现动态更新界面UI的功能，
330.      * 通过获取手机摄像头的参数来实时动态计算平均像素值、脉冲数，从而计算心率变化
331.      */
332.
333.     private static PreviewCallback previewCallback = new
334.         public void onPreviewFrame(byte[] data, Camera camera) {
335.             if (data == null)
336.                 throw new NullPointerException();
337.             Camera.Size size = camera.getParameters().getPreviewSize();
338.             if (size == null)
339.                 throw new NullPointerException();
340.             if (!processing.compareAndSet(false, true))
341.                 return;
342.             int width = size.width;
343.             int height = size.height;
344.             //图像处理
345.             int imgAvg = ImageProcessing.decodeYUV420SPto
```

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



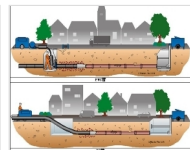
模拟检测



家用投影机



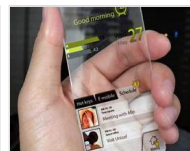
架构师



管道非开挖修



拍照手机排行



透明手机

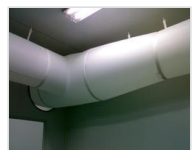
```

346.         gx=imgAvg;
347.         text1.setText("平均像素值是"+String.valueOf(imgAvg));
348.         //像素平均值imgAvg,日志
349.         //Log.i(TAG, "imgAvg=" + imgAvg);
350.         if (imgAvg == 0 || imgAvg == 255) {
351.             processing.set(false);
352.             return;
353.         }
354.         //计算平均值
355.         int averageArrayAvg = 0;
356.         int averageArrayCnt = 0;
357.         for (int i = 0; i < averageArray.length; i++) {
358.             if (averageArray[i] > 0) {
359.                 averageArrayAvg += averageArray[i];
360.                 averageArrayCnt++;
361.             }
362.         }
363.         //计算平均值
364.         int rollingAverage = (averageArrayCnt > 0)?
365.             (averageArrayAvg/averageArrayCnt):0;
366.         TYPE newType = currentType;
367.         if (imgAvg < rollingAverage) {
368.             newType = TYPE.RED;
369.             if (newType != currentType) {
370.                 beats++;
371.                 flag=0;
372.                 text2.setText("脉冲数是"+String.valueOf(beats));
373.                 //Log.e(TAG, "BEAT!! beats=" + beats)
374.             }
375.         } else if (imgAvg > rollingAverage) {
376.             newType = TYPE.GREEN;
377.         }
378.         if (averageIndex == averageArraySize)
379.             averageIndex = 0;
380.         averageArray[averageIndex] = imgAvg;
381.         averageIndex++;
382.
383.         // Transitioned from one state to another to

```

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



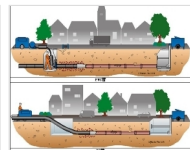
模拟检测



家用投影机



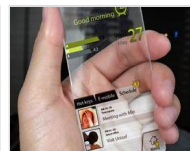
架构师



管道非开挖修



拍照手机排行



透明手机

```

384.         if (newType != currentType) {
385.             currentType = newType;
386.             //image.postInvalidate();
387.         }
388.         //获取系统结束时间 (ms)
389.         long endTime = System.currentTimeMillis();
390.         double totalTimeInSecs = (endTime - startTime) / 1000d;
391.         if (totalTimeInSecs >= 2) {
392.             double bps = (beats / totalTimeInSecs);
393.             int dpm = (int) (bps * 60d);
394.             if (dpm < 30 || dpm > 180 || imgAvg < 200) {
395.                 //获取系统开始时间 (ms)
396.                 startTime = System.currentTimeMillis();
397.                 //beats心跳总数
398.                 beats = 0;
399.                 processing.set(false);
400.                 return;
401.             }
402.             //Log.e(TAG, "totalTimeInSecs=" + totalTimeInSecs + " beats=" + beats);
403.             if (beatsIndex == beatsArraySize)
404.                 beatsIndex = 0;
405.             beatsArray[beatsIndex] = dpm;
406.             beatsIndex++;
407.             int beatsArrayAvg = 0;
408.             int beatsArrayCnt = 0;
409.             for (int i = 0; i < beatsArray.length; i++) {
410.                 if (beatsArray[i] > 0) {
411.                     beatsArrayAvg += beatsArray[i];
412.                     beatsArrayCnt++;
413.                 }
414.             }
415.             int beatsAvg = (beatsArrayAvg / beatsArrayCnt);
416.             text.setText("您的的心率"
417.                 + String.valueOf(beatsAvg) + " zhi:" + String.valueOf(beatsIndex)
418.                 + String.valueOf(beatsArrayCnt));
419.             //获取系统时间 (ms)
420.             startTime = System.currentTimeMillis();
421.             beats = 0;

```

关闭





布袋风管



管道规范



射频功率放大器



摄像机教程



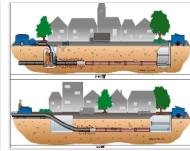
模拟检测



家用投影机



架构师



管道非开挖修



拍照手机排行



透明手机

```
422.         processing.set(false);
423.     }
424. };
425.
426. /**
427.  * 预览回调接口
428.  */
429. private static SurfaceHolder.Callback surfaceCallback = new SurfaceHolder.Callback() {
430.     //创建时调用
431.     @Override
432.     public void surfaceCreated(SurfaceHolder holder) {
433.         try {
434.             camera.setPreviewDisplay(previewHolder);
435.             camera.setPreviewCallback(previewCallback);
436.         } catch (Throwable t) {
437.             Log.e("PreviewDemo-
surfaceCallback", "Exception in setPreviewDisplay()", t);
438.         }
439.     }
440.     //当预览改变的时候回调此方法
441.     @Override
442.     public void surfaceChanged(SurfaceHolder holder, int format, int width, int
443.         Camera.Parameters parameters = camera.getParameters();
444.         parameters.setFlashMode(Camera.Parameters.FLASH_MODE_TORCH);
445.         Camera.Size size = getSmallestPreviewSize(width, height);
446.         if (size != null) {
447.             parameters.setPreviewSize(size.width, size.height);
448.             // Log.d(TAG, "Using width="
449.         }
450.         camera.setParameters(parameters);
451.         camera.startPreview();
452.     }
453.     //销毁的时候调用
454.     @Override
455.     public void surfaceDestroyed(SurfaceHolder holder) {
456.         // Ignore
457.     }
458. };
459.
```

关闭

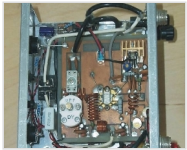




布袋风管



管道规范



射频功率放大



摄像机教程



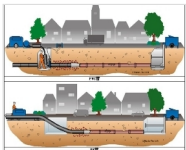
模拟检测



家用投影机



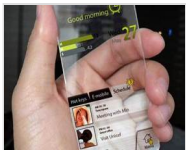
架构师



管道非开挖修



拍照手机排行



透明手机多小

```
460. /**
461.  * 获取相机最小的预览尺寸
462.  * @param width
463.  * @param height
464.  * @param parameters
465.  * @return
466.  */
467. private static Camera.Size getSmallestPreviewSize(int width, int height,
468.     Camera.Parameters parameters) {
469.     Camera.Size result = null;
470.     for (Camera.Size size : parameters.getSupportedPreviewSizes()) {
471.         if (size.width <= width && size.height <= height) {
472.             if (result == null) {
473.                 result = size;
474.             } else {
475.                 int resultArea = result.width * result.height;
476.                 int newArea = size.width * size.height;
477.                 if (newArea < resultArea)
478.                     result = size;
479.             }
480.         }
481.     }
482.     return result;
483. }
484. }
```

3、UI布局

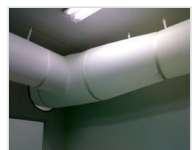
这里就不多说了，对于UI布局的实现，相信大家看了源码都懂得。

具体代码实现如下：

```
[html]
01. <LinearLayout xmlns:android="http://schemas.android.com/a
02.     xmlns:tools="http://schemas.android.com/tools"
03.     android:layout_width="match_parent"
04.     android:layout_height="match_parent"
```

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



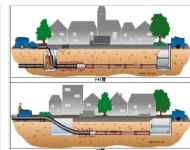
模拟检测



家用投影机



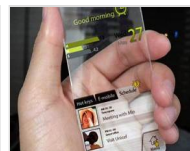
架构师



管道非开挖修



拍照手机排行



透明手机

```

05.         android:orientation="vertical" >
06.
07.     <SurfaceView
08.         android:id="@+id/preview"
09.         android:layout_width="fill_parent"
10.         android:layout_height="200dp"
11.         android:layout_marginLeft="50dip"
12.         android:layout_marginRight="50dip" />
13.
14.     <LinearLayout
15.         android:id="@+id/linearLayout1"
16.         android:layout_width="match_parent"
17.         android:layout_height="200dp"
18.         android:orientation="vertical" >
19.     </LinearLayout>
20.
21.     <TextView
22.         android:id="@+id/text"
23.         android:layout_width="wrap_content"
24.         android:layout_height="wrap_content"
25.         android:layout_marginLeft="50dip"
26.         android:layout_weight="1"
27.         android:text="@string/show" >
28.     </TextView>
29.
30.     <TextView
31.         android:id="@+id/text1"
32.         android:layout_width="wrap_content"
33.         android:layout_height="wrap_content"
34.         android:layout_marginLeft="50dip"
35.         android:layout_weight="1"
36.         android:text="@string/show" >
37.     </TextView>
38.
39.     <TextView
40.         android:id="@+id/text2"
41.         android:layout_width="wrap_content"
42.         android:layout_height="wrap_content"
43.         android:layout_marginLeft="50dip"

```

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



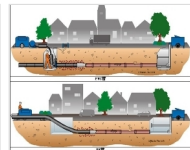
模拟检测



家用投影机



架构师



管道非开挖修



拍照手机排行



透明手机

4、strings.xml

```
[html]
01. <?xml version="1.0" encoding="utf-8"?>
02. <resources>
03.
04.     <string name="app_name">心率检测</string>
05.     <string name="action_settings">Settings</string>
06.     <string name="hello_world">Hello world!</string>
07.     <string name="show">显示</string>
08. </resources>
```

5、授权AndroidManifest.xml

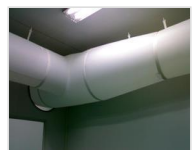
最后，我把授权文件贴出来吧。

具体代码实现如下：

```
[html]
01. <?xml version="1.0" encoding="utf-8"?>
02. <manifest xmlns:android="http://schemas.android.com/apk/r
03.     package="com.lyz.xinlv.activity"
04.     android:versionCode="1"
05.     android:versionName="1.0" >
06.
07.     <uses-sdk
08.         android:minSdkVersion="15"
09.         android:targetSdkVersion="15" />
10.     <uses-permission android:name="android.permission.WAK
11.     <uses-permission android:name="android.permission.CAM
```

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



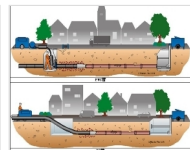
模拟检测



家用投影机



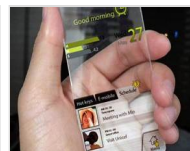
架构师



管道非开挖修



拍照手机排行



透明手机

```

12. <uses-feature android:name="android.hardware.camera" />
13. <uses-feature android:name="android.hardware.camera.autofocus" />
14.
15. <application
16.     android:allowBackup="true"
17.     android:icon="@drawable/ic_launcher"
18.     android:label="@string/app_name"
19.     android:theme="@style/AppTheme" >
20.     <activity
21.         android:name="com.lyz.xinlv.activity.MainActivity"
22.         android:label="@string/app_name" >
23.         <intent-filter>
24.             <action android:name="android.intent.action.MAIN" />
25.
26.             <category android:name="android.intent.category.LAUNCHER" />
27.         </intent-filter>
28.     </activity>
29. </application>
30.
31. </manifest>

```

至此，这个应用开发完成，亲们，是不是比想象的简单呢？

三、运行效果

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



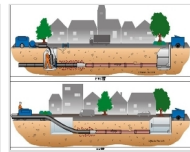
模拟检测



家用投影机



架构师



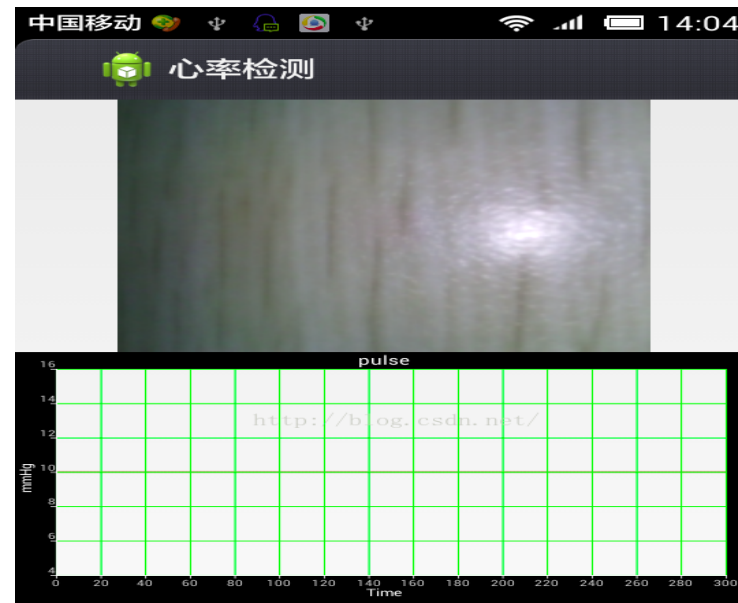
管道非开挖修



拍照手机排行



透明手机

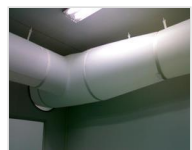


关闭

可穿戴技术逆袭帕金森

英特尔与迈克尔 J. 福克斯帕金森氏症基金会 (MJFF) 携手开发数据分析方案, 识别疾病模式并进行归纳, 加快实现治疗方案的突破

英特尔、Intel 是英特尔公司在美国和其他国家的商标。*其他的名称和品牌可能是其他所有者的财产。



布袋风管



管道规范



射频功率放大



摄像机教程



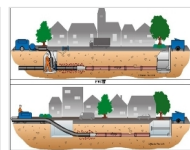
模拟检测



家用投影机



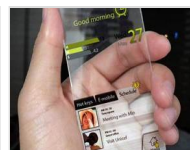
架构师



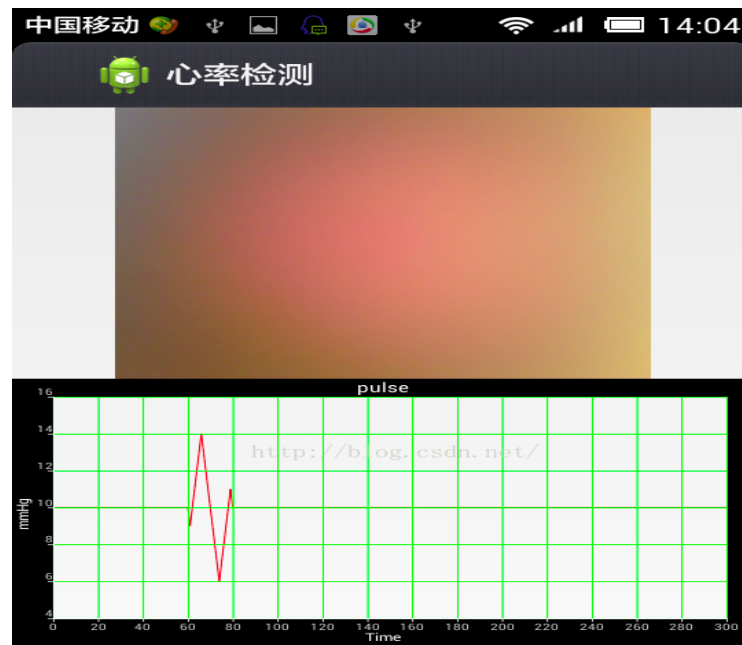
管道非开挖修



拍照手机排行



透明手机



显示

平均像素值是187

脉冲数是2.0

关闭

可穿戴技术逆袭帕金森

英特尔与迈克尔·J. 福克斯帕金森氏症基金会 (MJFF) 携手开发数据分析方案, 识别疾病模式并进行归纳, 加快实现治疗方案的突破

英特尔、Intel 是英特尔公司在美国和其他国家的商标。*其他的名称和品牌可能是其他所有者的商标。



布袋风管



管道规范



射频功率放大



摄像机教程



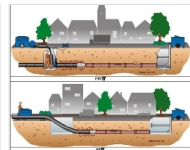
模拟检测



家用投影机



架构师



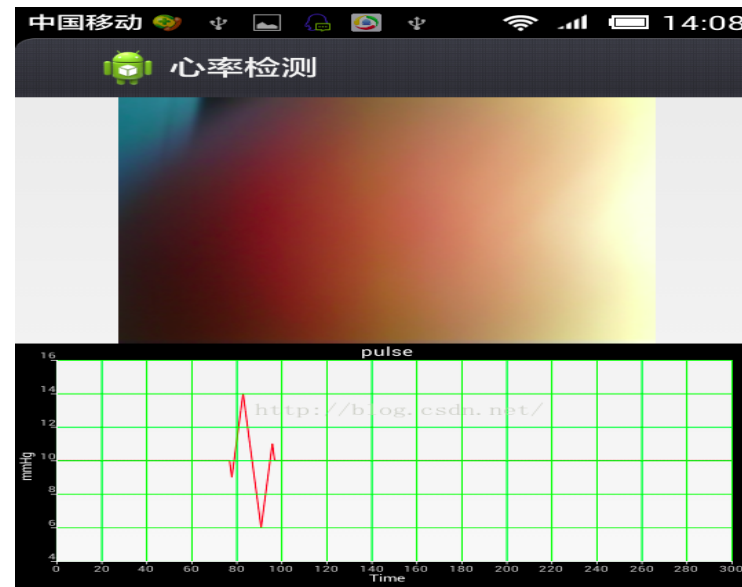
管道非开挖修



拍照手机排行



透明手机



您的心率是54 zhi:3 1 54 1

平均像素值是239

脉冲数是1.0

关闭

可穿戴技术逆袭帕金森

英特尔与迈克尔 J. 福克斯帕金森氏症基金会 (MJFF) 携手开发数据分析方案, 识别疾病模式并进行归纳, 加快实现治疗方案的突破

英特尔、Intel 是英特尔公司在美国和其他国家的商标。*其他的名称和品牌可能是其他所有者的商标。



布袋风管



管道规范



射频功率放大



摄像机教程



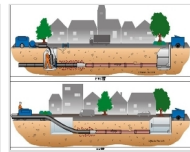
模拟检测



家用投影机



架构师



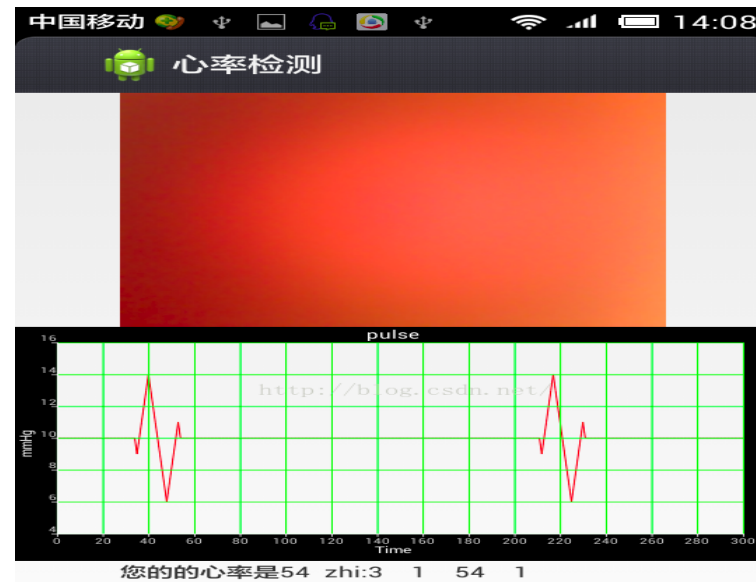
管道非开挖修



拍照手机排行



透明手机多小



平均像素值是224

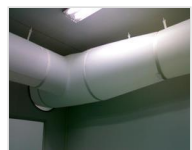
脉冲数是1.0

关闭

可穿戴技术逆袭帕金森

英特尔与迈克尔 J. 福克斯帕金森氏症基金会 (MJFF) 携手开发数据分析方案, 识别疾病模式并进行归纳, 加快实现治疗方案的突破

英特尔、Intel 是英特尔公司在美国和其他国家的商标。*其他的名称和品牌可能是其他所有者的财产。



布袋风管



管道规范



射频功率放大



摄像机教程



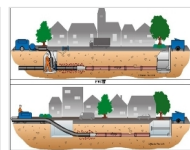
模拟检测



家用投影机



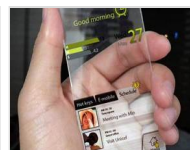
架构师



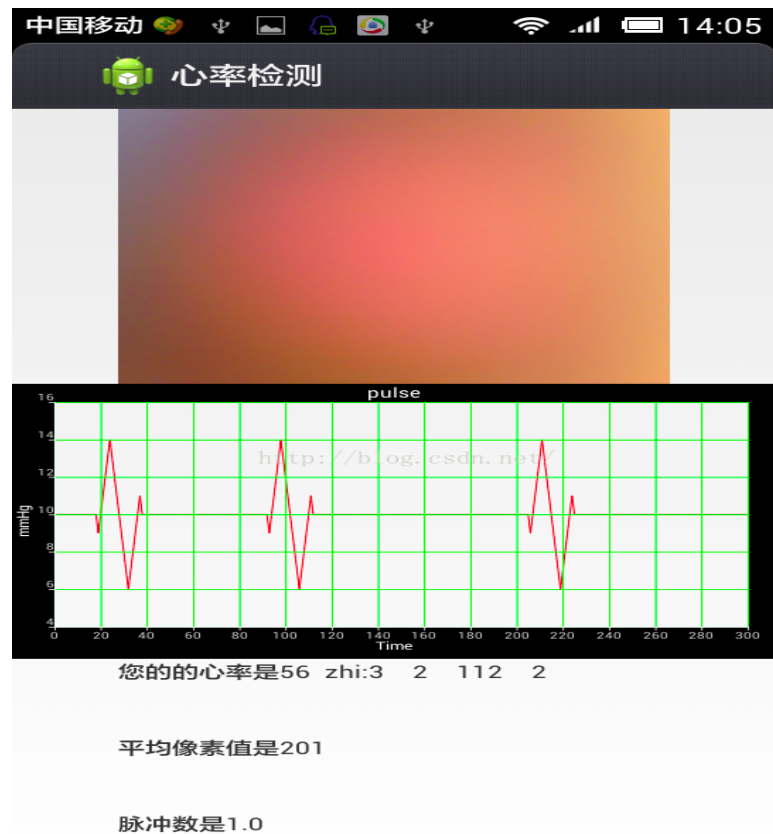
管道非开挖修



拍照手机排行



透明手机



四、温馨提示

本实例中，为了方面，我把一些文字直接写在了布局文件中。项目中要把这些文字写在string.xml文件中，在外部引用这些资源。Android程序员最基本的开发常识和规范，我在这里只是为了中。

大家可以到链接<http://download.csdn.net/detail/I102838680>测心率变化的应用源代码

关闭





布袋风管



管道规范



射频功率放大



摄像机教程



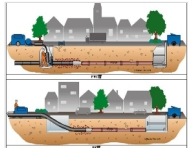
模拟检测



家用投影机



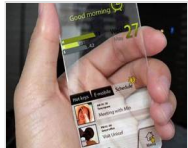
架构师



管道非开挖修



拍照手机排行



透明手机

顶

0

踩

0

上一篇 [一、Android调用WebServices原理](#)

下一篇 [Android 之窗口小部件详解--App Widget](#)

相关文章推荐

- [光学心率测量原理](#)
- [Android 那些事- 小米手环 测量心率 动画实现](#)
- [iOS手机摄像头测心率](#)
- [IOS 测心跳的实现原理](#)
- [keepalived\(监测心跳\)](#)
- [Android4.0新增API](#)
- [Android4.0 SDK功能详解](#)
- [基于qualcomm平台实现心率变化的检测功能](#)
- [Android4.0新的SDK新特性（汉化）](#)
- [android测心率、测睡眠质量、测心率数据、心率图](#)

关闭



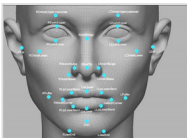
什么国产手机



人工智能机器



app开发报价单



人脸识别



三防手机

猜你在找

[机器学习之概率与统计推断](#)

[机器学习之凸优化](#)

[响应式布局全新探索](#)

[机器学习之数学基础](#)

[机器学习之矩阵](#)

[探究Linux的总线、设备](#)



[深度学习基础与TensorFlow实践](#)[前端开发在线峰会](#)[深度学习之神经网络原理与实战技巧](#)[TensorFlow实战进阶：手把手教你做图像识别应用](#)[查看评论](#)

暂无评论

[发表评论](#)用户名：[haijzun](#)

评论内容：

[提交](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

[关闭](#)[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)[网站客服](#)[杂志客服](#)[微博客服](#)[webmaster@csdn.net](#)

400-660-0108

| 北京创新乐知信息技术有限公司 版权所有 | 江苏知

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved

