

嵌入式手机终端开发技术

大作业报告

（ 2020 / 2021学年 第 一 学期）

题 目： 二维码实现

|  |  |
| --- | --- |
| **专 业** | **计算机科学与技术** |
| **学 生 姓 名** | **吴 雯** |
| **班 级 学 号** | **B18030322** |
| **指 导 教 师** | **易云山** |
| **指 导 单 位** | **计算机科学与技术系** |
| **日 期** | **2020.10.20** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **评 分 细 则** | **评分项和课程目标** | | | |
| 实验报告和程序完成情况能够反映出该生有兢兢业业的学习态度和责任感 | | **课程目标1（10分）** | |
|  | |
| 实验报告的撰写认真程度、规范性和完整性能够真实的反映出该生独立完成本次大作业，并且工作量饱满 | | **课程目标1（10分）** | |
|  | |
| 实验报告能够反映出该生掌握移动互联网的基本概念，对移动互联网最新应用开发技术有一定的了解 | | **课程目标1（10分）** | |
|  | |
| 从实验报告和程序完成情况反映出该生对课题中存在的复杂工程问题进行了分析并给出合理的解决方案 | | **课程目标1（30分）** | |
|  | |
| 从程序完成情况能够反映出该生熟练掌握最新移动开发应用技术的开发环境搭建、调试方法和接口的使用 | | **课程目标2（20分）** | |
|  | |
| 程序能够正确运行，满足功能要求，用户体验较好，并具备一定的创新性 | | **课程目标2（20分）** | |
|  | |
| **最终得分** |  | | |
| **评阅教师** | | **易云山** | **评阅日期** | **2020.XX.XX** |

**二维码实现**

1. **课题内容和要求**

课题内容：二维码使用黑白矩形图案表示二进制数据符号信息，用若干个与二进制相对应的几何形体来表示文字数值信息，通过图象输入设备或光电扫描设备自动识读以实现信息自动处理。结合生活实际，实现一个简单的二维码功能。

课题要求：

1. 要求能够通过文本（注：文本可以自己定义，比如：文字或网页地址等）生成二维码图片，该生成的二维码图片中央需包含一个小的南邮图徽（图片可以自己搜索获取），类似QQ自动生成的二维码中央有企鹅图标；
2. 能够扫描1生成的二维码或其他二维码图片，获取文本信息；
3. 通过上述2获取的文本信息进行自动判断是否是文本信息还是网页浏览；
4. 如果上述3中是文本，则调用Android文本阅读器显示；如果是网页链接，需启动浏览器进行访问；
5. 上述功能需要在Android智能手机上实现演示；
6. 可以通过二维码支持扩展至一维码支持（可选）

**二、课题需求分析**

本课题目标系统“二维码实现”的功能框架图如图1所示。

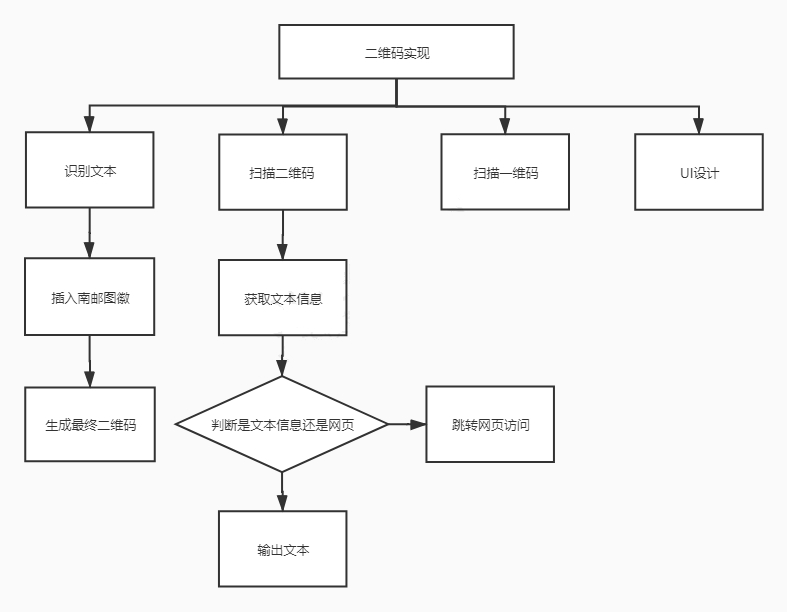


图1 功能框架图

（1）支持自定义输入文本或网页网址；

（2）利用zxing将自定义内容转换成带有南邮logo的二维码；

（3）支持扫描解析（2）中生成的二维码；

（4）支持分析判断解析出的二维码内容是文本内容还是网页网址，如果是文本则调用Android文本阅读器显示，若是网页链接则启动浏览器进行访问；

（5）支持将二维码的功能扩展至一维码；

**三、程序设计**

1 主要数据结构

类的继承

1. **public** **class** MainActivity **extends** AppCompatActivity {
2. ...
3. ...
4. ...
5. }

捕获异常

1. **try** {
2. // 打开Camera硬件设备
3. cameraManager.openDriver(surfaceHolder);
4. // 创建一个handler来打开预览，并抛出一个运行时异常
5. **if** (handler == **null**) {
6. handler = **new** CaptureActivityHandler(**this**, decodeFormats,
7. decodeHints, characterSet, cameraManager);
8. }
9. } **catch** (IOException ioe) {
10. Log.w(TAG, ioe);
11. displayFrameworkBugMessageAndExit();
12. } **catch** (RuntimeException e) {
13. Log.w(TAG, "Unexpected error initializing camera", e);
14. displayFrameworkBugMessageAndExit();
15. }

2 主要流程

1. 生成简单二维码：首先判断字符串内容是否为空，其次设置二维码相关配置，然后将参数传入到QRCodeWriter的encode方法生成BitMatrix(位矩阵)对象中，接着创建像素数组，并根据BitMatrix(位矩阵)对象为数组元素赋颜色值，最后创建Bitmap对象，根据像素数组设置Bitmap每个像素点的颜色值，并返回Bitmap对象，即二维码；其实现流程图如图2所示。

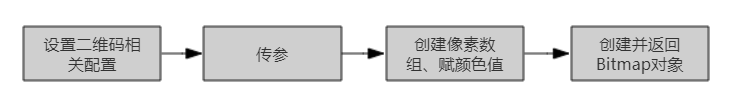


图2 生成简单二维码

1. 生成带logo的二维码：在（1）的基础上添加logo图片，在上述已实现的简单二维码程序的最后加上为二维码添加logo图标的程序段；
2. 生成一维码：首先判断字符串内容是否为空，其次设置一维码相关配置，接着然后将文本内容进行编码，最后创建Bitmap对象，根据像素数组设置Bitmap每个像素点的颜色值，并返回Bitmap对象，即一维码；其实现流程图如图3所示。

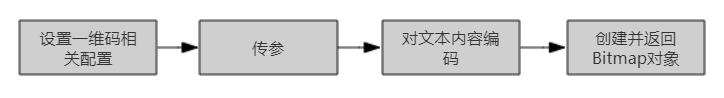


图3 生成一维码

1. 扫描一维码：首先申请打开摄像头权限，接着解码，识别出图片二维码/条码，内容为content，然后通过Intent返回最终结果，最后将文本通过Android文本阅读器显示出来。其实现流程图如图4所示。

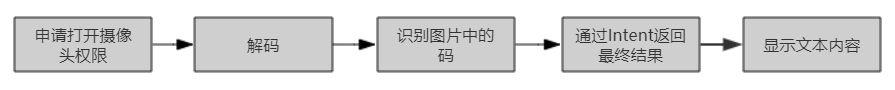


图4 扫描一维码

**四、源程序代码**

MainActivity.java

1. **public** **class** MainActivity **extends** AppCompatActivity {
3. **private** **static** **final** String DECODED\_CONTENT\_KEY = "codedContent";
4. **private** **static** **final** String DECODED\_BITMAP\_KEY = "codedBitmap";
5. **private** **static** **final** **int** REQUEST\_CODE\_SCAN = 0x0000;
7. QRCodeBitmapService qrCodeBitmapService;
8. Resources resources;
9. EditText editText;
10. Button buttonTrans;
11. Button buttonScan;
12. ImageView image;
13. TextView tv\_scanResult;
14. Button buttonOneCodeTrans;
15. Context context;
16. Button about;
17. @Override
18. **protected** **void** onCreate(Bundle savedInstanceState) {
19. **super**.onCreate(savedInstanceState);
20. setContentView(R.layout.activity\_main);
21. init();
22. }
24. **private** **void** init() {
25. qrCodeBitmapService = **new** QRCodeBitmapServiceImpl();
26. resources = getResources();
27. editText = findViewById(R.id.edit);
28. buttonTrans = findViewById(R.id.trans);
29. buttonScan = findViewById(R.id.scan);
30. image = findViewById(R.id.imageQR);
31. tv\_scanResult=findViewById(R.id.text\_view);
32. buttonOneCodeTrans = findViewById(R.id.oneCodeTrans);
33. context = **this**;
34. about = findViewById(R.id.about);
35. image.setImageBitmap(BitmapFactory.decodeResource(resources,resources.getIdentifier(
36. "njupt","drawable","com.pawncs.qrcode"
37. )));
38. addButtonListener();
39. //        changeButtonColor();
40. }

43. @Override
44. **public** **void** onRequestPermissionsResult(**int** requestCode, @NonNull String[] permissions, @NonNull **int**[] grantResults) {
45. **switch** (requestCode) {
46. **case** 1:
47. **if** (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION\_GRANTED) {
48. //goScan();
49. Intent intent = **new** Intent(MainActivity.**this**, CaptureActivity.**class**);
50. startActivityForResult(intent, REQUEST\_CODE\_SCAN);
51. } **else** {
52. Toast.makeText(**this**, "你拒绝了权限申请，可能无法打开相机扫码哟！", Toast.LENGTH\_SHORT).show();
53. }
54. **break**;
55. **default**:
56. }
57. }

60. @Override
61. **protected** **void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {
62. **super**.onActivityResult(requestCode, resultCode, data);
63. // 扫描二维码/条码回传
64. **if** (requestCode == REQUEST\_CODE\_SCAN && resultCode == RESULT\_OK) {
65. **if** (data != **null**) {
66. //返回的文本内容
67. String content = data.getStringExtra(DECODED\_CONTENT\_KEY);
68. //返回的BitMap图像
69. Bitmap bitmap = data.getParcelableExtra(DECODED\_BITMAP\_KEY);
71. //tv\_scanResult.setText("你扫描到的内容是：" + content);
72. }
73. }
74. }
76. **public** **void** addButtonListener(){
77. //单击转换，将内容转换为二维码。
78. buttonTrans.setOnClickListener(**new** View.OnClickListener() {
79. @Override
80. **public** **void** onClick(View view) {
81. String text = editText.getText().toString();
82. Log.i("TEST", text);
83. //生成二维码
84. Bitmap qrCode = qrCodeBitmapService.createQRCode(text);
85. Bitmap logo =  BitmapFactory.decodeResource(resources,resources.getIdentifier(
86. "njupt","drawable","com.pawncs.qrcode"
87. ));
88. **if**(logo == **null**){
89. Log.w("??","no logo");
90. }
91. **if**(qrCode != **null**){
92. image.setImageBitmap(qrCodeBitmapService.
93. AddLogo(qrCode, logo,(**float**) 0.15));
94. }**else**{
95. image.setImageBitmap(BitmapFactory.decodeResource(resources,resources.getIdentifier(
96. "njupt","drawable","com.pawncs.qrcode"
97. )));
98. Toast.makeText(context,"请输入合法内容哦",Toast.LENGTH\_SHORT).show();
99. }
100. }
101. });
103. buttonOneCodeTrans.setOnClickListener(**new** View.OnClickListener() {
104. @Override
105. **public** **void** onClick(View view) {
106. String text = editText.getText().toString();
107. Log.i("TEST", text);
108. //生成一维码
109. Bitmap qrCode = qrCodeBitmapService.createOneCode(text);
110. **if**(qrCode!=**null**){
111. image.setImageBitmap(qrCode);
112. }**else**{
113. image.setImageBitmap(BitmapFactory.decodeResource(resources,resources.getIdentifier(
114. "njupt","drawable","com.pawncs.qrcode"
115. )));
116. Toast.makeText(context,"请输入合法内容哦",Toast.LENGTH\_SHORT).show();
117. }
118. }
119. });
121. //单击扫描，将二维码转换为内容。如果是网页，则打开页面。
122. buttonScan.setOnClickListener(**new** View.OnClickListener(){
123. @Override
124. **public** **void** onClick(View view) {
125. //TODO
127. //动态权限申请
128. **if** (ContextCompat.checkSelfPermission(MainActivity.**this**, Manifest.permission.CAMERA) != PackageManager.PERMISSION\_GRANTED) {
129. ActivityCompat.requestPermissions(MainActivity.**this**, **new** String[]{Manifest.permission.CAMERA}, 1);
130. }
131. Intent intent = **new** Intent(MainActivity.**this**, CaptureActivity.**class**);
132. startActivityForResult(intent, REQUEST\_CODE\_SCAN);
133. }
134. });
135. //页面跳转
136. about.setOnClickListener(**new** View.OnClickListener() {
137. @Override
138. **public** **void** onClick(View view) {
139. Intent intent = **new** Intent(MainActivity.**this**,AboutActivity.**class**);
140. startActivity(intent);
141. }
142. });
143. }
144. }

AboutActivity.java

1. **package** com.pawncs.qrcode;
3. **import** androidx.appcompat.app.AppCompatActivity;
5. **import** android.content.Intent;
6. **import** android.os.Bundle;
7. **import** android.view.View;
9. **public** **class** AboutActivity **extends** AppCompatActivity {
11. @Override
12. **protected** **void** onCreate(Bundle savedInstanceState) {
13. **super**.onCreate(savedInstanceState);
14. setContentView(R.layout.activity\_about);
15. findViewById(R.id.back).setOnClickListener(**new** View.OnClickListener() {
16. @Override
17. **public** **void** onClick(View view) {
18. Intent intent = **new** Intent(AboutActivity.**this**,MainActivity.**class**);
19. startActivity(intent);
20. }
21. });
22. }
23. }

handleDecode()

1. /\*\*
2. \* 扫描成功，处理反馈信息
3. \*
4. \* @param rawResult
5. \* @param barcode
6. \* @param scaleFactor
7. \*/
8. **public** **void** handleDecode(Result result, Bitmap barcode, **float** scaleFactor) {
9. inactivityTimer.onActivity();
10. String resultString = result.getText();
11. **if** (resultString.equals("")) {
12. Toast.makeText(CaptureActivity.**this**, "Scan failed!", Toast.LENGTH\_SHORT).show();
13. }**else** {
14. Intent resultIntent = **new** Intent();
15. Bundle bundle = **new** Bundle();
16. bundle.putString("result", resultString);
17. resultIntent.putExtras(bundle);
18. **this**.setResult(RESULT\_OK, resultIntent);
19. Pattern pattern1=Pattern.compile("http://(([a-zA-z0-9]|-){1,}\\.)[a-zA-z0-9]{1,}-\*");
20. Matcher matcher1=pattern1.matcher(resultString);
21. Pattern pattern2=Pattern.compile("https://(([a-zA-z0-9]|-){1,}\\.)[a-zA-z0-9]-\*");
22. Matcher matcher2=pattern2.matcher(resultString);
23. **if**(matcher1.find()||matcher2.find()){ //判断扫描的结果是否为url、是url直接跳转到网页中
24. Intent intent = **new** Intent();
25. intent.setAction("android.intent.action.VIEW");
26. Uri content\_url = Uri.parse(resultString);
27. intent.setData(content\_url);
28. startActivity(intent);
29. }**else**{//不是直接显示
30. //创建dialog对象
31. AlertDialog alertDialog = **new** AlertDialog.Builder(CaptureActivity.**this**).create();
32. alertDialog.setTitle("扫描结果");
33. //显示扫描的信息
34. alertDialog.setMessage(resultString);
35. //设置按钮
36. alertDialog.setButton(AlertDialog.BUTTON\_NEGATIVE, "取消", **new** DialogInterface.OnClickListener() {
38. @Override
39. **public** **void** onClick(DialogInterface dialog, **int** which) {
40. // TODO Auto-generated method stub
41. dialog.dismiss();
42. }
43. });
44. alertDialog.show();
45. }
46. }
47. }

activity\_main.xml

1. <?xml version="1.0" encoding="utf-8"?>
2. <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3. xmlns:app="http://schemas.android.com/apk/res-auto"
4. xmlns:tools="http://schemas.android.com/tools"
5. android:layout\_width="match\_parent"
6. android:layout\_height="match\_parent"
7. tools:context=".MainActivity">
9. <RelativeLayout
10. android:id="@+id/layout"
11. android:layout\_width="match\_parent"
12. android:layout\_height="match\_parent"
13. tools:layout\_editor\_absoluteX="0dp"
14. tools:layout\_editor\_absoluteY="0dp">
16. <EditText
17. android:id="@+id/edit"
18. android:layout\_width="500dp"
19. android:layout\_height="wrap\_content"
20. android:hint="输入要转换为一维码/二维码的文本"
21. android:layout\_marginBottom="10dp"
22. android:inputType="text" />
24. <Button
25. android:id="@+id/oneCodeTrans"
26. android:layout\_width="wrap\_content"
27. android:layout\_height="wrap\_content"
28. android:layout\_below="@id/edit"
29. android:layout\_marginLeft="30dp"
30. android:background="@drawable/button\_selector"
31. android:text="转换一维码" />
33. <Button
34. android:id="@+id/trans"
35. android:layout\_width="wrap\_content"
36. android:layout\_height="wrap\_content"
37. android:layout\_below="@id/edit"
38. android:layout\_alignParentRight="true"
39. android:layout\_marginRight="30dp"
40. android:background="@drawable/button\_selector"
41. android:text="转换二维码"/>
43. <ImageView
44. android:id="@+id/imageQR"
45. android:layout\_width="340dp"
46. android:layout\_height="340dp"
47. android:layout\_below="@id/trans"
48. android:layout\_marginTop="30dp"
49. android:layout\_centerHorizontal="true"
50. android:src="@mipmap/ic\_launcher"
51. />
52. <Button
53. android:id="@+id/scan"
54. android:layout\_width="wrap\_content"
55. android:layout\_height="wrap\_content"
56. android:layout\_below="@id/imageQR"
57. android:layout\_centerInParent="true"
58. android:layout\_marginTop="30dp"
59. android:background="@drawable/button\_selector"
60. android:text="扫码" />
61. <Button
62. android:id="@+id/about"
63. android:layout\_width="wrap\_content"
64. android:layout\_height="wrap\_content"
65. android:layout\_below="@id/scan"
66. android:layout\_centerHorizontal="true"
67. android:layout\_marginTop="60dp"
68. android:background="@drawable/button\_selector"
69. android:text="关于" />
70. <TextView
71. android:layout\_width="match\_parent"
72. android:layout\_height="wrap\_content"
73. android:text=""
74. android:layout\_below="@id/imageQR"
75. android:id="@+id/text\_view"/>
77. </RelativeLayout>

80. </androidx.constraintlayout.widget.ConstraintLayout>

**五、测试数据及其结果分析**

运行QRCode.apk，可以得到如图5所示的界面，可以看出此apk能够实现的基础功能：转一维码、转二维码、扫描二/一维码以及关于四个部分，初始时二维码用南邮校徽来显示，彰显学校特色；

在输入栏中输入“哈哈哈哈哈，hello njupt!”（如图6），其包含了中英文字符，点击“转换二维码”可以成功转换成二维码显示出来；

图5 首页 图6 生成二维码 图7 申请权限

第一次点击“扫码”按钮，会弹出申请权限的对话框（如图7），点击“始终允许”后可以进入扫码界面，我们扫描之前生成的二维码可以得到显示在Android文本阅读器中的“哈哈哈哈哈，hello njupt!”（如图8），由此可以证明我们的二维码生成与扫码对于中英文都适用；

下面我们继续输入<https://www.baidu.com/>，并且转换成二维码（如图9），利用扫码功能扫描此二维码，成功跳转到百度首页（如图10），成功实现了判断是文本文档还是网页链接并打开的功能；

图8 扫码调用文本阅读器 图9 转网址为二维码 图10 扫码跳转网址

然后我们在输入栏输入“hello,njupt”，成功转换成一维码（如图11），利用扫码功能扫码生成的一维码成功解码得到原字符串（如图12），二维码功能扩展到一维码实现完成；

最后，我们看看“关于”按钮，点击后进入图13所示界面，其中简单描述了此apk的功能以及开发者信息。

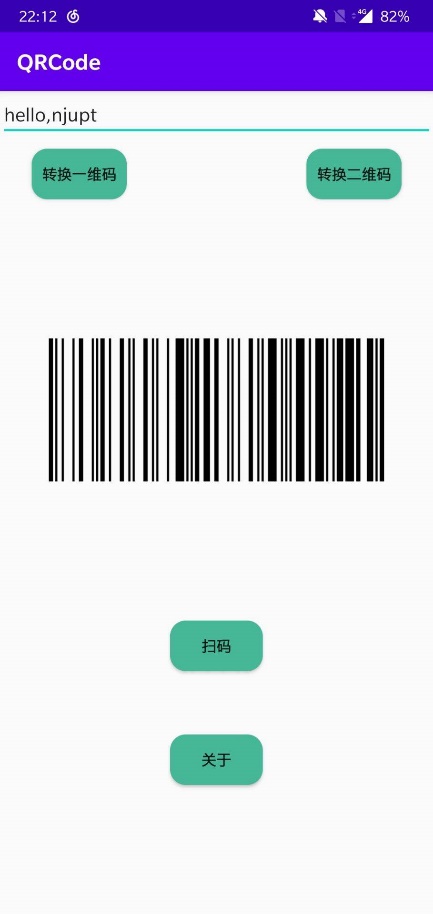
  

图11 转一维码 图12 扫码一维码 图13 关于

根据以上的数据测试与分析可以看出此apk能够正常无错的执行5个必选课题要求和1个可选要求，程序健壮性良好。

**六、课题完成过程中遇到的问题及解决方法**

问题1：环境问题，在初次使用Android studio时需要安装SDK，需要自己手动安装；在新建项目后，编译器要根据自己的版本下载对应的gradle版本，下载过慢无法开始执行程序；

解决方法：停止在编译器中的自动下载，通过查询项目主文件夹下面的gradle文件夹下面的wrapper下的gradle-wrapper.properties文件中的gradle版本，然后利用迅雷下载其官网压缩包放置到对应的文件夹中重新启动该项目，项目就开始配置运行环境了；

问题2：导入其他项目中的包时由于代码中import的包名不一致发生很多报错，修改包名以后还是很多bug；

解决方法：修改导入的zxing包中的包名，根据报错信息修改相应的函数和参数；

问题3：刚开始写一维码时，手机无法打开摄像头来识别一维码；

解决方法：经查询，在Android8以后，打开摄像头需要申请打开摄像头的权限，需要在清单文件中加上扫描需要的权限：

1. @Override
2. **public** **void** onRequestPermissionsResult(**int** requestCode, @NonNull String[] permissions, @NonNull **int**[] grantResults) {
3. **switch** (requestCode) {
4. **case** 1:
5. **if** (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION\_GRANTED) {
6. //goScan();
7. Intent intent = **new** Intent(MainActivity.**this**, CaptureActivity.**class**);
8. startActivityForResult(intent, REQUEST\_CODE\_SCAN);
9. } **else** {
10. Toast.makeText(**this**, "你拒绝了权限申请，可能无法打开相机扫码哟！", Toast.LENGTH\_SHORT).show();
11. }
12. **break**;
13. **default**:
14. }
15. }

问题4：写好相关代码后发现编译器中无报错，但是无法运行；

解决方法：经过向队友请教，得知该执行程序段没有注册，无法执行，然后将CaptureActivity.java类的方法在配置文件中进行注册，问题解决。

**七、总结**

通过此次的课题程序设计，我们对这学期所学的嵌入式手机终端开发有一个更深刻的了解，将所学知识应用于实践。为了使apk能够实现预先的构想，我们也查阅了网上相关程序，比较好的流程图及功能模块，然后再着手实现程序并不断修改以达到预先的构想，完成之后感觉对Java程序以及Android开发又有了新的认识。

本次课程设计我们使用的是Google的zxing框架，利用其提供的接口实现课题需要实现的功能要求。了解到课程要求后，我们通过百度等途径得到相关信息：ZXing是一个开源的，用Java实现的多种格式的1D/2D条码图像处理库，它包含了联系到其他语言的端口。zxing可以实现使用手机的内置的摄像头完成条形码的扫描及解码。因此我们选则了zxing来实现功能。在利用zxing实现功能的过程中，我们也是遇到了很多很多问题，各种依赖各种包问题，我们通过一步一步解决，最终解决了问题，也提高了我们debug的能力，收获颇大。

在本课题之前也做过其他语言的课程设计，但模块之间联系没有这么紧因为各个模块之间的联系十分紧密，代码重复执行率高，经过调试、修改后，终于看到成品。

在这个过程中，我们了解到了团队合作的必要性和对团队成员的提高功能，在编写程序的过程中，互相学习彼此的代码，发现了自己的错漏之处并予以补强。同时在讨论过程中对要做怎样的功能逐渐完善，也跟让我们明白了讨论的重要性。在完成整个程序的进程里，我们采用了“先完成主体功能，再逐渐细化其他要求”的方式。

纸上得来终觉浅，绝知此事要躬行。通过本次的课题设计，我们愈发感受到了实践在学习过程中起到的重要作用，我平时学到的理论知识得到了很好的应用，理论与实际相结合，受益匪浅。