remoteproject Documentation

Release 1.0

dumiao

June 05, 2015

CONTENTS

Contents:

CHAPTER

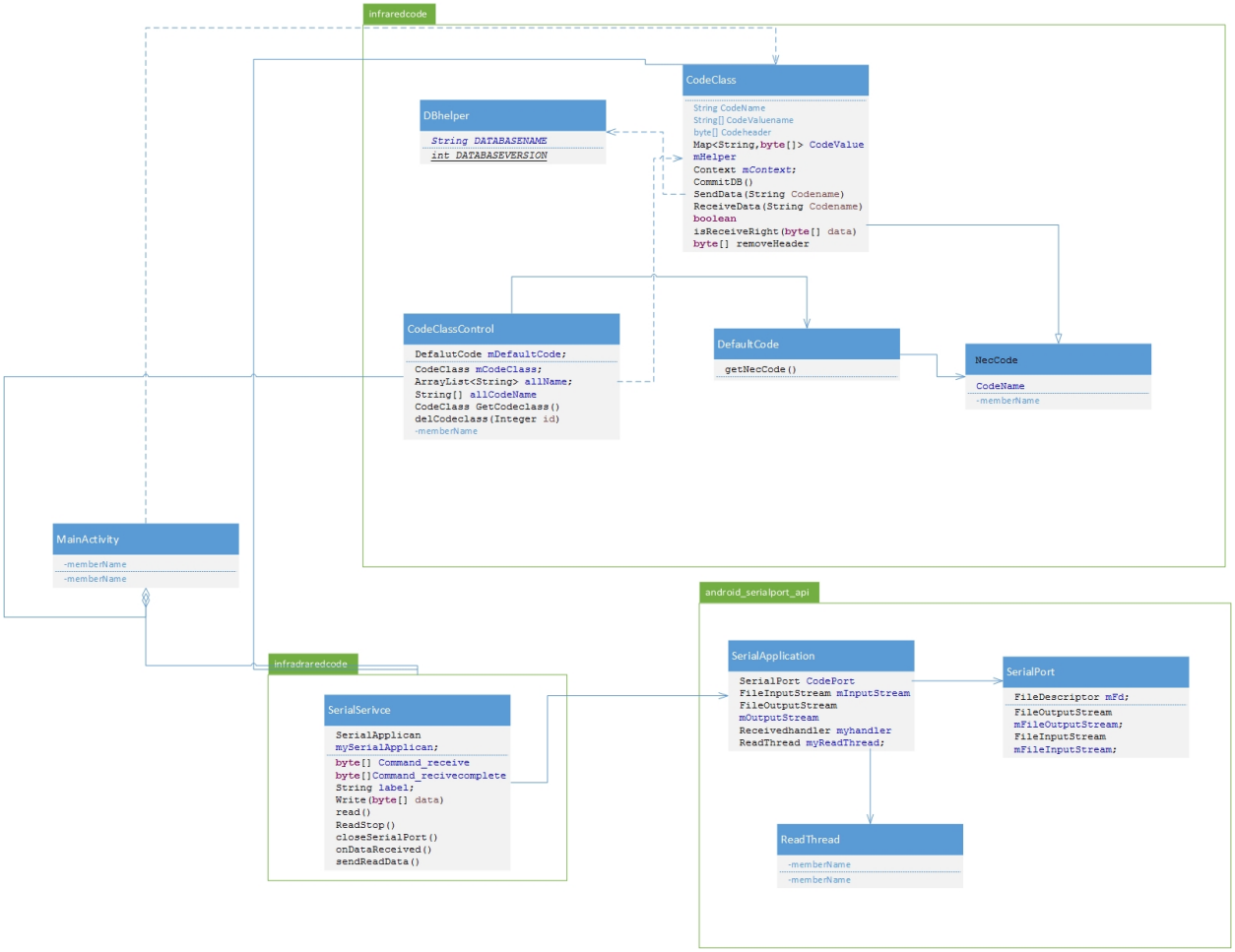
ONE

THIS IS A BASIC INTRODUCTION

The remote Android App is a complement demo of Cms remote IR Remote Controller.It's extensible and convenient. User can directly use it for most scenes,but we also provide APIs for user to develope some special remote or just use Android device's UART. this page will show the App's class diagram and work flow.

1.1 Class diagram

There are three java packages in the demo:*android_serialport_api*, *infraredcode*, *infraredCodeSerivce*. *android_serialport_api* is a set of classes about serialport, *infraredcode* is a set of classes about infraredcode,and *infraredCodeSerivce* include a serivce to operate Communication between CodeClass and Serialport. In these Classes, the **CodeClass** is the core class, It's a abstraction of reality remote which includes member variable *CodeName*s ,*CodeValue*s.It's Member functions *SendData(Codename)* will trigger SerialSerivce to Send byte[] to Fpga through serial port,and *CommitDB()* will wirte CodeClass to Sqltite. *CodeClassControl* is a Class which can create *CodeClass*,delete *CodeClass* and get All CodeClasses name in the Sqltie.

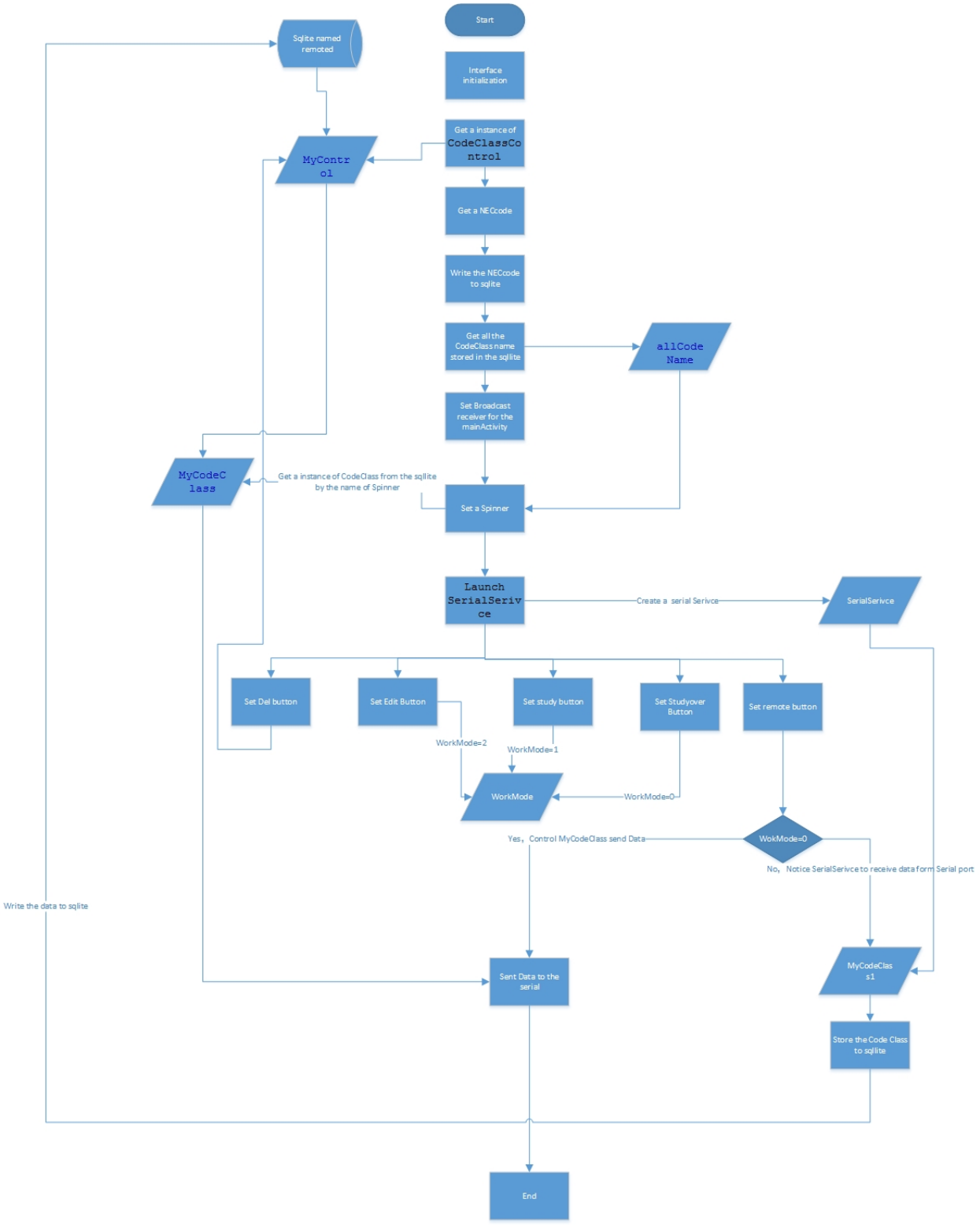


There are a few things to note

1. This app runs successfully in the android Development Board which mode is Intel Sharks Cove. So theoretically It can be compatible with all x86 and arm cpu.
2. This app need root permission.

1.2 Work flow

this is the APP‘s MainActivity initialization process.If you want to modify this App deeply, you can refer to this process , if you just want to add some special button, or change the layout ,We recommend that you refer to the second **tutorial_**.


Start
Sqlite named remoted
Interface initialization
Get a instance of CodeClassControl
MyControl
Get a NECcode
Write the NECcode to sqlite
Get all the CodeClass name stored in the sqlite
allCode Name
Set Broadcast receiver for the mainActivity
MyCodeClass
Get a instance of CodeClass from the sqllite by the name of Spinner
Set a Spinner
Launch SerialSerivce
Create a serial Serivce
SerialSerivce
Set Del button
Set Edit Button
Set study button
Set Studyover Button
Set remote button
WorkMode=2
WorkMode=1
WorkMode
WorkMode=0
WokMode=0
Yes, Control MyCodeClass send Data
No, Notice SerialSerivce to receive data form Serial port
Write the data to sqlite
Sent Data to the serial
MyCodeClass1
Store the Code Class to sqlite
End

CHAPTER

TWO

THIS IS A INTRODUCTION OF HOW TO EXTEND THE APP (ADD BUTTON)

The remote Android App is a complement demo of cms' FPGA remote IR Remote Controller.It's extensible and convenient. User can directly use it for most scenes, but we also provide APIs for user to develope some special remote or just use Android device's UART. this page will show you the Processes and considerations when you extend this app.

2.1 client.xml

1. This is the *remote button code* part,you can add your remote button among them. we recommand adding *image-button* which will auto include click effects.

```
<RelativeLayout
   android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:id="@+id/buttonfield">"
   <ImageButton
       android:id="@+id/power"
       android:layout_width="40dip"
       android:layout_height="40dip"
       android:layout_alignParentLeft="true"
       android:layout_alignParentTop="true"
       android:layout_marginLeft="40dip"
       android:layout_marginTop="20dip"
       android:background="@drawable/power" />
   <ImageButton
       android:id="@+id/cli1"
       android:layout_width="40dip"
       android:layout_height="40dip"
       android:layout_alignParentLeft="true"
       android:layout_below="@+id/power"
       android:layout_marginLeft="40dip"
       android:layout_marginTop="10dip"
       android:background="@drawable/cli1" />
   <ImageButton
       android:id="@+id/cli2"
       android:layout_width="40dip"
       android:layout_height="40dip"
       android:layout_alignParentRight="true"
       android:layout_alignTop="@+id/power"
       android:layout_marginRight="40dip"
       android:background="@drawable/cli2" />

   <ImageButton
       android:id="@+id/volume_add"
       android:layout_width="65dip"
       android:layout_height="65dip"
       android:layout_below="@+id/cli1"
       android:layout_centerHorizontal="true"
       android:background="@drawable/volume_add" />

   <ImageButton
       android:id="@+id/play"
       android:layout_width="65dip"
       android:layout_height="65dip"
```

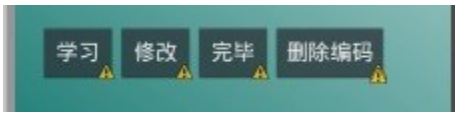
```
        android:layout_alignLeft="@+id/volup"
        android:layout_below="@+id/volume_add"
        android:layout_centerInParent="true"
        android:layout_marginTop="10dip"
        android:background="@drawable/center" />

    <ImageButton
        android:id="@+id/prev"
        android:layout_width="65dip"
        android:layout_height="65dip"
        android:layout_alignLeft="@+id/cli1"
        android:layout_alignTop="@+id/play"
        android:layout_below="@+id/volume_add"
        android:background="@drawable/prev" />

    <ImageButton
        android:id="@+id/next"
        android:layout_width="65dip"
        android:layout_height="65dip"
        android:layout_alignRight="@+id/cli2"
        android:layout_alignTop="@+id/play"
        android:background="@drawable/next" />

    <ImageButton
        android:id="@+id/volume_down"
        android:layout_width="65dip"
        android:layout_height="65dip"
        android:layout_alignLeft="@+id/play"
        android:layout_below="@+id/play"
        android:layout_marginTop="10dip"
        android:background="@drawable/volume_down" />
</RelativeLayout>
```

2. This is the *command button code* part. We do not recommend changing this part.




```
  <RelativeLayout
    android:id="@+id/relate_level1"
    android:layout_width="wrap_content"
    android:layout_height="50dp"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="40dp">
 <Button
     android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginLeft="20dp"
      android:id="@+id/study"
     android:text="学习"/>
 <Button
     android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_toRightOf="@id/study"
      android:id="@+id/edit"
     android:text="修改"/>
```

```
  <Button
     android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_toRightOf="@id/edit"
      android:id="@+id/studyover"
     android:text="完毕"/>

     <Button
         android:id="@+id/delete"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:layout_toRightOf="@id/studyover"
         android:text="删除编码" />

</RelativeLayout>
```

2.2 the process of add button

there are only six remote button in the demo,you can add your button conveniently, Here's an example of adding remote button *MUTE*

1. Add a Imagebutton in the xml, Attention: you need a button named mute in the drawer folder

```
<ImageButton
              android:id="@+id/mute"
              android:layout_width="40dip"
              android:layout_height="40dip"
              android:layout_alignParentLeft="true"
              android:layout_below="@+id/power"
              android:layout_marginLeft="40dip"
              android:layout_marginTop="10dip"
              android:background="@drawable/mute" />
```

2. Add a Imagebutton declaration in the Mainactivity

```
private ImageButton mute;
volume_up=(ImageButton)findViewById(R.id.mute);
```

3. Set a cliklistener for the Imagebutton

```
mute.setOnClickListener(new OnClickListener(){
                    @Override
                public void onClick(View v) {
                    // TODO Auto-generated method stub
                    if(WorkMode==0){

                            MyCodeClass.SendData("MUTE");
                    }else
                    {
                    CodeClass.ReceiveData("MUTE");



                    }
                }
 });
```

There are a few things to Attention

1. You can redesign the interface following your favorite style.
2. wo don't recommended that you change the ID of the views,becasue most of them are bound in the Mainactivity.

2.3 Waring:

Currently the keywords supported by the sqllite are listed below:

"NUM1"（数字键）,"NUM2","NUM3","NUM4","NUM5","NUM6","NUM7","NUM9","NUM0","PAUSE"(暂停),"LIKE"（喜爱）,"SINALSOURCE"（切换信号源）,"SLEEP"（睡眠）,"MENU","MUTE（静音）", "OK","POWER"（电源键）,"PLAY","PRE","NEXT","VOLUME_DOWN","VOLUME_UP" };

There are a few things to Attention:

1. Keywords there has no capacity constraints, eg.you can learn your reality remote mute button on the "POWER" or other keywordes.
2. wo don't recommended that you change the ID of the views,becasue most of them are bound in the Mainactivity.