

# Android 系统下卫星天线 M&C 的实现

马曙晖

(南京邮电大学 通信与信息工程学院, 江苏 南京 210003)

**摘要:**卫星通信便携式地球站因其体积较小、组网下的高灵活性优点而被广泛应用。作为系统重要的组成部分,以往的监控平台难以满足流行的运行终端,故提出 Android 平台下的 M&C 系统。该系统通过与天线系统的连接,以友好的图形用户界面显示寻星效果,使通信地球站的对星精度基本满足要求,实现便携站与下位机的通信链路的监控。从 Android 开发的基本应用知识开始,加以必要的需求分析,在软件方面,给出包括代码的主要模块化设计与编写,提出了主要子模块模型和最终软件结构设计思路,最终实现简约图形用户界面,功能主要有参数设置,包括方位定位、自动和手动对星等。软件在 Eclipse 配置的 SDK 环境下,使用当下流行的 Java 语言开发。

**关键词:**卫星通信;M&C 系统;Android;图形用户界面;Java

**中图分类号:**TP39

**文献标识码:**A

**文章编号:**1673-629X(2016)10-0133-04

**doi:**10.3969/j.issn.1673-629X.2016.10.029

## Implementation of M&C for Satellite Antenna on Android System

MA Shu-hui

(College of Telecommunications and Information Engineering, Nanjing University of  
Posts and Telecommunications, Nanjing 210003, China)

**Abstract:**Man-pack satellite communication earth station is widely used owing to its compact size and flexibility on networking. As an important part of this system, the previous monitoring system is no longer satisfied with the popular running platform. In view of this, puts forward the thought of M&C system in satellite on the Android system. Through a connection with the antenna system, it displays satellite search effect with a friendly GUI (Graphical User Interface) and basically satisfies requirements of accuracy on search. Meanwhile it achieves the monitoring between portable station and slave computer. Beginning with the basic Android application of knowledge, plus necessary analysis and requirements, in the perspective of the software, the modularized program design and compiling is given. The main submodules and the design idea of final software structure are presented. Finally, it provides a concise GUI which makes users be able to complete the parameter settings, including pointing satellite location by automatic or manual pattern. The software development mainly uses the current Java under SDK environment with Eclipse.

**Key words:**satellite communication; M&C system; Android; GUI; Java

## 0 引言

一个完整的地球通信站<sup>[1]</sup>主要包括终端设备、伺服收发跟踪系统和监控系统。其中监控系统是整个通信系统的大脑,负责对天线系统主要参数进行控制、监管,使操作人员可以通过监控端监视和控制卫星通信系统设备的运行状况,同时当设备存在故障时将故障内容(error.log)记录在事件中,从而给检修人员提供帮助。监控系统为用户提供了友好便捷的可视化界面,极大方便了非专业用户的操作。

近些年大多数软件运行在 PC 端。由于 PC 相对

手机来说不便携带,所以 PC 端虽然拥有比较丰富的开发资源,但是为了软件运行所耗费的精力和时间也较多,成本也相对较高,而且准备工作又比较繁琐。

由于近年来 Android 系统<sup>[2-3]</sup>在终端领域的井喷式发展,使得基于 Android 平台的监控系统具有巨大的应用前景,因此设计一款能运行在 Android 操作系统下的控制软件,这样既轻便,又简单易操作,更有其重要的实际价值,对于企业和客户来说均能节约不少成本。

收稿日期:2015-12-24

修回日期:2016-04-20

网络出版时间:2016-09-19

基金项目:国家自然科学基金资助项目(61271234)

作者简介:马曙晖(1991-),男,硕士研究生,研究方向为计算机网络、卫星通信技术。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20160919.0839.016.html>

## 1 Android 系统的体系架构

Android 的系统架构如图 1 所示<sup>[4]</sup>, 与其操作系统一样, 采用了分层架构。从架构图看, Android 分为四层, 从高到低分别是应用程序层、应用程序框架层、系统运行库层和 Linux 核心层。

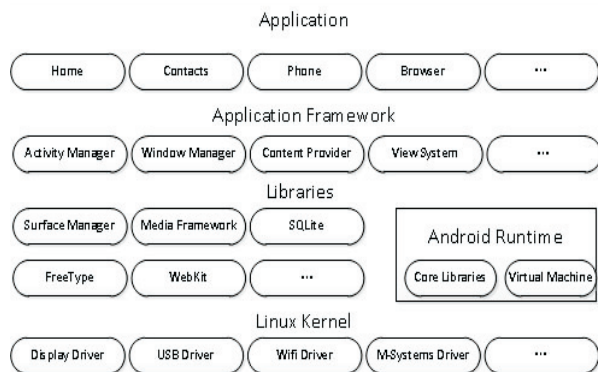


图 1 Android 架构

(1) 应用程序: Android 系统一般内置有 Email、短信收发程序、浏览器、联系人等功能的应用程序。除了内置的应用程序外, 开发者还可以编写更多的应用程序, 让用户能够使用更多便利的功能。

(2) 应用程序框架: 其实就是 Android 的 API (Application Programming Interface), 开发者只要善用此 API 即可开发出 Android 应用程序。应用程序就是依赖框架层次 API, 所有应用都是一组服务和系统, 一般包含: 一套丰富且可扩展的视图组件, Activity Manager (活动管理器), Content Providers (内容提供器), Resource Manager (资源管理器), Notification Manager (信息管理器)。

(3) 系统运行库: Android 有一个内部函数库, 此函数库主要用 C/C++ 编写。Android 应用程序开发人员并非直接使用此函数库, 而是通过更上层的应用程序框架来使用此函数库功能, 所以有人称此类函数库为原生函数库 (Native Libraries)。此函数库依照功能也可细分成各种类型的函数库。

(4) Linux 内核: Android 以 Linux 2.6 版作为整个系统的核心, Linux 提供 Android 主要的系统服务, 如: 安全性管理 (Security)、内存管理 (Memory Management)、进程管理 (Process Management)、网络栈 (Network Stack)、驱动模型 (Driver Model)、电源管理 (Power Management) 等。

## 2 总体设计

### 2.1 系统顶层建模

建模过程<sup>[5]</sup>如图 2 所示, 像多数管理软件一样, user 可以设置成管理员和普通用户不同的登录模式, 均可实现基本操作, 不同的是管理员对系统参数设置

有更高的权限, 主要可以管理数据库<sup>[6]</sup>。

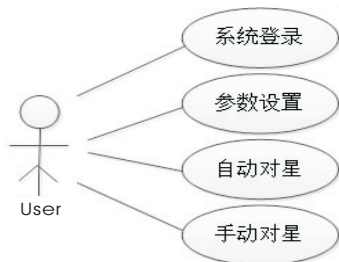


图 2 顶端模型图

### 2.2 自动与手动对星建模

如图 3 所示, 自动模式即采用自动对星方式寻星, 并且在界面实时显示参数, 而手动是在事先知道卫星参数情况下手动调整, 或是卫星偏移较大以至于长时间难以自动对准而手动调到参数附近位置, 然后再采用自动模式, 手动调整的是俯仰、方位以及极化角度。

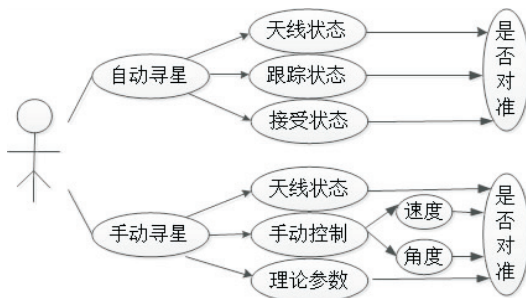


图 3 自动与手动对星模型图

### 2.3 系统软件结构图

软件结构图如图 4 所示。

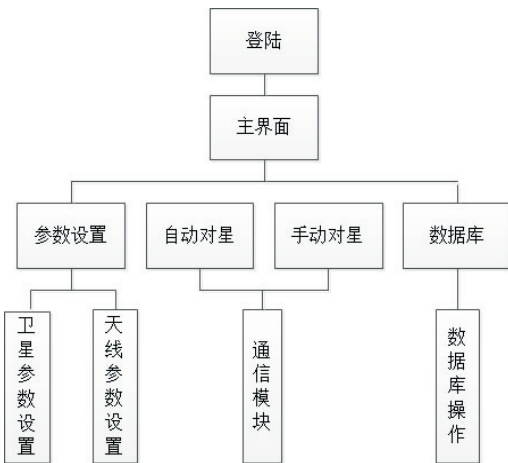


图 4 软件结构图

## 3 系统的具体实现

### 3.1 Socket 通信模式

Socket 套接字机制<sup>[7]</sup>广泛运用于网络通信, 采用 TCP 协议<sup>[8]</sup>, 通过面向连接的可靠性服务实现 C/S 模式的通信。操作 Socket 过程大致如下:

(1) 调用 Socket 类的构造函数, 以服务器的指定 IP 地址或指定的主机名和指定的端口号为参数, 创建

一个 Socket 流,在创建 Socket 流的过程中包含了向服务器请求建立通信连接的过程实现。

(2)建立了 client 端通信 Socket 后,就可以使用 Socket 的方法 `getInputStream()` 和 `getOutputStream()` 来创建输入/输出流。这样,使用 Socket 类后,网络输入输出也转化为使用流对象的过程。

(3)使用输入输出流对象的相应方法读写字节流,因为流连接着通信所用的 Socket,Socket 又是和服务端建立连接的一个接口,因此数据将通过连接从服务器得到或发向服务器。这时就可以对字节流数据按客户端和服务端之间的协议进行处理,完成双方的通信任务。

(4)通信结束时,先关闭输入输出流,再关闭 Socket,分别采用对应的 `close()` 方法。

### 3.2 登录界面的实现

经过登录界面进入时,会有三个空白窗,分别为 user、password 和 IP 地址。IP 地址作为唯一识别对应天线的参数,一般情况下,天线型号与 IP 地址相关联能够解析出对应的 IP 地址,而该 IP 是由后台 Web 服务器端通过 WiFi 设置的,要想修改 IP,需要先修改 Web 端对应的 IP。

登录界面关键实现代码如下:

```
UserImpl impl=new UserImpl(Login.this);
Cursor cursor=impl.login(name.getText().
toString(),password.getText().toString(),
ipadd.getText().toString());
//管理消息
if(cursor.moveToNext()){
Toast.makeText(Login.this,"登陆成功",Toast.LENGTH_
SHORT).show();
}else{Toast.makeText(Login.this,"输入错误",Toast.
LENGTH_SHORT).show();
}
```

### 3.3 参数设置界面的实现

鉴于卫星主要的参数均设置在界面当中,若用户要对其进行修改,直接修改即可。卫星信息已经关联了数据库,重新选择通信卫星,直接在下拉菜单选中,比如中星 6A,对应显示其经纬度,用户若想对卫星数据进行修改,直接操作数据库中数据即可。值得一提的是,卫星通信保证有 500 MHz 的频带传输,其中上行频段为 14.0~14.5 GHz,下行频段为 12.25~12.75 GHz,变频之后的信号必须在此范围内,且保证传码率以及接收发送的频率都必须一致才能互通。最后选择接收机,只能选 DVB 接收机与信标机之一。

布局与数据读入关键代码:

```
public classSetActivity extends Activity{
private Cursor cursor;
```

```
private DataBase db=new DataBase(this);
/*省略部分代码*/
MytabActivity mytabs=(MytabActivity)this.getParent();
public void onCreate(Bundle savedInstanceState){...}
cursor=db.getAllTitles();
SimpleCursorAdapteradapter=new SimpleCursorAdapter(this,
android.R.layout.simple_spinner_item,mycursor,new String[]{"
卫星名称"},new int[]{android.R.id.text});
/*省略部分代码*/
satname.setOnItemClickListener(new OnItemSelectedListener(){...}

public voidonItemSelected(AdapterView<?> arg0,View arg1,
int arg2,long arg3){...}
```

### 3.4 位置定位

与监控系统连接的是天线系统<sup>[9-11]</sup>,上面安装有 GPS 接收机,监控具有两种工作模式—自动模式和人工模式。随着技术的发展,诸如蓝牙、WiFi 等多种无线通信模块早已做成集成芯片安装在手机上,充分利用 Android 智能手机<sup>[12-14]</sup>内置 GPS 定位模块,基于 Eclipse 开发平台通过程序的调用,实现定位功能<sup>[15]</sup>,给出当前天线地理位置的经度和纬度。

利用手机 GPS 获取地理方位关键代码如下:

```
private void setLocation(Location location){
if(location!=null){
String latitudeStr=Double.toString(location.getLatitude());
//经度
StringlongitudeStr=Double.toString(location.getLongitude());
//纬度
if(latitudeStr!=null&&!latitudeStr.equals("")){latitude.
setText(latitudeStr);}
if(longitudeStr!=null&&!longitudeStr.equals("")){longi-
tude.setText(longitudeStr);}
if(altitudeStr!=null&&!altitudeStr.equals("")){altitude.
setText(altitudeStr);}
}else{Log.v("location","location 为空");}
LocationManager locationManager=null;
updateToNewLocation(locat);//监听器监听
Locationlocat=locationMgr.getLastKnownLocation(provider);
//获取位置
locationMgr.requestLocationUpdates(provider,100*1000,
500,locationListener);
}
```

### 3.5 理论计算与代码实现

理论计算是根据所选择参数以及方位信息等通过公式计算天线的俯仰方位和极化。在界面中单击菜单,选择对星按钮后,如果给出所有参数格式且不为空时,系统就会根据读取的参数并参照开发时规定的公式计算出卫星天线的极化角、俯仰角和方位角,并将理论值显示在监控界面上。理论上计算公式如下:

俯仰角:

$$F_y =$$

$$\arctan\left(\frac{\cos(\text{Local}_w * \text{PI}/180) * (\cos(-t * \text{PI}/180) - 0.1513)}{\sqrt{1 - \cos^2(\text{Local}_w * \text{PI}/180) * \cos^2(-t * \text{PI}/180)}}\right)$$

方位角:

$$F_w = \text{PI} - \arctan\left(\frac{\tan(t * \text{PI}/180)}{\sin(\text{Local}_w * \text{PI}/180)}\right)$$

极化角:

$$J_h = \arctan\left(\frac{\sin(t * \text{PI}/180)}{\tan(\text{Local}_w * \text{PI}/180)}\right)$$

实现计算关键代码:

```
private void computer() {
double local_j=0;//本地经度
double local_w=0;//本地纬度
// * * * 省略若干初始化计算代码 * * *
double fw_angle=0,fy_angle=0,fy=0,pf=0,jh_angle=0;
double zz=PI/180,float tt=sate_j-local_j;
//方位角
fw_angle=Math.atan(Math.tan(tt*zz)/Math.sin(local_w*zz));
fw_angle=PI-fw_angle;//中国所在为东经
//俯仰角
pf=1-(Math.cos(local_w*zz)*Math.cos(-tt*zz))*
(Math.cos(local_w*zz)*Math.cos(-tt*zz));
fy=((Math.cos(local_w*zz))*(double)Math.cos(-tt*zz))-0.1513/Math.sqrt(pf);
fy_angle=Math.atan(fy); //zz 不用再乘
//极化角
if(jComboBox.getSelectedItem()=="垂直"){
jh_angle=Math.atan(Math.sin(tt*zz)/Math.tan(local_w*zz)); //垂直方式
v_flag=0;
}
if(jComboBox.getSelectedItem()=="水平"){
jh_angle=Math.atan(Math.sin(tt*zz)/Math.tan(local_w*zz)); //水平方式
h_flag=0;
if(jh_angle>0){jh_angle=jh_angle-PI/2;}
else{jh_angle=jh_angle+PI/2;}
}
fw_angle=fw_angle*180/PI;
fy_angle=fy_angle*180/PI;
jh_angle=jh_angle*180/PI;
}
```

#### 4 实现结果

系统采用 MVC 设计模式,运用 Socket 编程实现互通,实现了基本操作功能,包括登录、参数设置、寻星的自动与手动模式,如图 5 所示。



图 5 多层次图形界面

#### 5 结束语

该系统是在 Android 的虚拟环境下开发实现,虽测试能完成相应基本功能,但毕竟是模拟器,在测试响应过程容易受到外部环境(如天气等)的影响,还要加以改进确保无差错响应。对于其他更高层次的功能,如无人值守,气象监控,早已出现在现如今许多地球站,由于能力有限以及设备的局限等主客观原因暂未能实现,但这不失为今后研究的大方向。

#### 参考文献:

- [1] 汤 鹏. 一种新型便携式卫星通信地球站的 M&C 系统 [D]. 南京:南京邮电大学,2014.
- [2] Meier R. Android 4 高级编程 [M]. 余建伟,赵 凯,译. 第 3 版. 北京:清华大学出版社,2013.
- [3] 农丽萍,王力虎,黄一平. Android 在嵌入式车载导航系统的应用研究[J]. 计算机工程与设计,2010,31(11):2473-2476.
- [4] 徐 玲,蒋欣志,张 杰. 手机二维码识别系统的设计与实现[J]. 计算机应用,2012,32(5):1474-1476.
- [5] Blaha M, Rumbaugh J. UML 面向对象建模与设计 [M]. 车皓阳,杨 眉,译. 北京:人民邮电出版社,2011.
- [6] 吴 中,郑晓华. 基于 Java 的 Web 数据库访问系统[J]. 科学之友:上旬,2007(2):158-159.

( $R_1, R_2, R_3, R_4, R_5$ ) 上,若其处理能力为 (120, 200, 150, 340, 500) MFLOPS,实验过程中分别记录 5 个资源节点上的资源负载情况,其结果如图 5 所示。

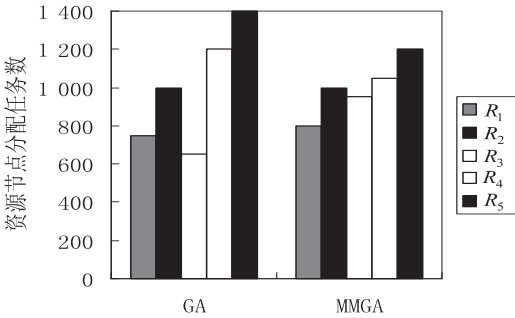


图 5 多任务情况下节点的资源负载情况

由图 5 可知,在任务量大、资源节点运算能力差异较大的情况下,MMGA 的资源负载均衡程度明显好于 GA。

综上所述,文中提出的 MMGA 比 GA 收敛速度更快,且可以使得任务执行时间较短,资源负载较均衡,能较好地应用在云计算资源环境中。

4 结束语

文中在充分考虑大规模任务处理特性和云计算环境动态异构性的基础上,提出了一种基于传统遗传算法的改进任务调度算法-MMGA。该算法既可以保证种群具有较高的平均适应度,又可以维护种群个体的多样性;同时,算法中采用任务执行时间和负载均衡作为双适应度函数,使得在提高收敛速度的同时,兼顾资源均衡。仿真结果表明,该改进算法收敛性能较好、资源负载较均衡,具有良好的效率,能更有效地解决云计算环境下的任务调度问题。

参考文献:

[1] Chien A, Calder B, Elbert S, et al. Entropia: architecture and performance of an enterprise desktop grid system[J]. Journal of Parallel and Distributed Computing, 2003, 63(5): 597-

610.  
[2] Kim J S, Nam B, Marsh M, et al. Creating a robust desktop grid using peer-to-peerservices [EB/OL]. [2009-10-16]. ftp://ftp.cs.umd.edu/pub/hpsl/papers/papers-pdf/ngs07.pdf.  
[3] Armbrust M, Fox A, Griffith R, et al. A view of cloud computing[J]. Communications of the ACM, 2009, 53(4): 50-58.  
[4] Carretero J, Xhafa F. Use genetic algorithms for scheduling jobs in large scale grid applications[J]. Technologies and Economic Development of Economy, 2006, 12(1): 11-17.  
[5] Buyya R, Ranjan R, Calheiros R N. Modeling and simulation of scalable cloud computing environments and the CloudSim Toolkit: challenges and opportunities[C]//Proceedings of the seventh high performance computing and simulation conference. New York, USA: IEEE Press, 2009: 21-24.  
[6] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters[C]//Proceedings of the 6th symposium on operating system design and implementation. New York: ACM, 2004: 137-150.  
[7] 王小平,曹立明. 遗传算法[M]. 西安: 西安交通大学出版社, 2002.  
[8] 李建锋,彭 舰. 云计算环境下基于改进遗传算法的任务调度算法[J]. 计算机应用, 2011, 31(1): 184-186.  
[9] 朱宗斌,杜中军. 基于改进 GA 的云计算任务调度算法[J]. 计算机工程与应用, 2013, 49(5): 77-80.  
[10] 叶春晓,陆 杰. 基于改进遗传算法的网格任务调度研究[J]. 计算机科学, 2010, 37(7): 233-235.  
[11] 王春莲. 基于改进遗传算法的网格任务调度算法[D]. 济南: 山东大学, 2009.  
[12] 刘 愉,赵志文,李小兰,等. 云计算环境中优化遗传算法的资源调度策略[J]. 北京师范大学学报: 自然科学版, 2012, 48(4): 378-384.  
[13] 段卫军,付学良,王 芳,等. 云计算环境下融合遗传算法和蚁群算法 QoS 约束任务调度[J]. 计算机应用, 2014, 34(S2): 66-69.  
[14] 邹伟明,于 炯. 云计算环境下基于用户满意度的遗传算法[J]. 计算机应用研究, 2014, 31(1): 85-88.

(上接第 136 页)

[7] Eckel B. Java 编程思想[M]. 陈昊鹏,译. 北京: 机械工业出版社, 2007.  
[8] 张海燕. Java 多线程技术在手机联网中的应用[J]. 农业网络信息, 2008(3): 97-98.  
[9] Kraus J D, Marhefka R J. Antennas: for all applications[M]. New York: New McGraw-Hill, 2001.  
[10] Bridge W M. Cross coupling in a five horn monopulse tracking system[J]. IEEE Transactions on Antennas and Propagation, 1972, 20(4): 436-442.  
[11] Weaver A C, Luo Jihao, Zhang Xinyuan. Monitoring and con-

trol using the Internet and Java industrial electronics society [C]//Proc of IECON'99. [s.l.]: IEEE, 1999: 1154-1158.  
[12] 代 敏. 基于 Android 平台下手机定位程序的设计及实现[J]. 计算机与数字工程, 2012, 40(4): 143-145.  
[13] 耿东久,索 岳,陈 渝,等. 基于 Android 手机的远程访问和控制系统[J]. 计算机应用, 2011, 31(2): 559-561.  
[14] 王朝华,陈德艳,黄国宏,等. 基于 Android 的智能家居系统的研究与实现[J]. 计算机技术与发展, 2012, 22(6): 225-228.  
[15] 艾国祥,施浒立,吴海涛,等. 基于通信卫星的定位系统原理[J]. 中国科学: G 辑, 2008, 38(12): 1615-1633.