

[文章编号] 1003—4684(2022)01-0001-05

基于 Android 的联合收割机远程故障监测终端系统

杨光友^{1,2}, 谢 旺^{1,2}, 陈学海^{1,2}

(1 湖北工业大学农业机械工程研究设计院, 湖北 武汉 430068;

2 湖北省农机装备智能化工程技术研究中心, 湖北 武汉 430068)

[摘 要] 针对传统的 PC 端故障查询系统便携性差, 难以满足操作员田间作业要求的问题, 借助 Android 移动平台适应性广、交互性能强等优点, 提出一种基于 Android 平台的联合收割机远程故障监测和查询系统。该 Android 移动端和远程云服务器端建立 socket 通信获取预警故障, 同时结合操作员的经验判断故障位置, 实现远程故障诊断。利用 Mysql 数据库实现故障查询数据库的搭建, 可进行故障原因和解决方案的查询。经测试, 该移动端具有良好的人机交互性和便携性, 能有效的提高故障诊断的准确性和解决故障的效率。

[关键词] 安卓操作系统; Mysql 数据库; 故障查询数据库; 远程故障诊断; 联合收割机

[中图分类号] S126 [文献标识码] A

由于联合收割机^[1-3]结构复杂, 工作环境恶劣, 其工作过程中的故障多样化。常见的故障监测和诊断系统大多运行在 PC 端, 虽然实现了远程故障监测, 减轻了故障诊断的工作量和难度, 但不易携带, 难以满足实际需要^[4]。随着移动互联网和移动设备的不断发展^[5], 如能将联合收割机的故障通过移动端显示或发出预警信息, 则对联合收割机的运维具有重要意义。目前, 有关移动端故障监测和诊断系统的研究已经在飞机航电系统^[6]、汽车^[7-9]和工业设备^[10]等领域展开。本文以联合收割机故障监测为背景, 开展基于 Android 移动设备的联合收割机远程故障监测终端系统研究, 为联合收割机故障监测和诊断提供解决方法, 提高运维效率。

1 系统总体结构

基于 Android 的联合收割机远程监测系统主要由车载数据采集端、云服务器端和 Android 移动端组成(图 1)。

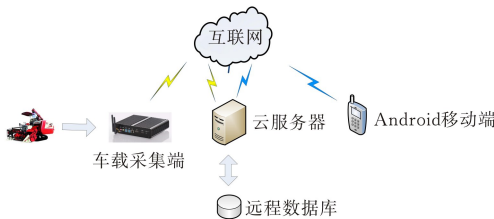


图 1 总体框架

1.1 车载数据采集端

以雷沃 RG50 型联合收割机为试验样机, 根据收割机工作情况, 在收割机工作时易发生故障的部位安装速度传感器, 以监测风机转速、脱粒滚筒转速、输送链耙转速、喂入搅龙转速、杂余搅龙转速、籽粒搅龙转速、逐稿器和割刀往复频率等。车载数据采集端通过数据采集模块将这些传感器数据利用 4G 通信模块上传至云服务器端。

1.2 云服务器端

作为整个系统的核心枢纽, 云服务器端负责调用故障模型进行故障预警和故障模型的更新: 云服务器端接受车载端上传的数据并解析保存到数据库—调用这些数据进行故障预警—得到预警结果实时发送到 Android 移动端。同时, 云服务器端获取 Android 移动端人工反馈的实际故障结果, 并与远程数据库中模型预警的故障状态进行一致性判断, 以此结果作为模型是否需要更新的重要条件之一。

1.3 Android 移动端

Android 移动端软件采用 TCP/IP 协议与云服务器端建立 socket 通信, 实时接收云服务器端发送的监测模型预警结果。当其结果是故障状态时, 即断开移动端与云服务器的连接, 由操作员判断是否与实际故障相符, 同时上传同一时刻操作员判断给出的故障状态, 以便服务器端计算故障预警模型的准确率。通过连接访问远程数据库, 操作员根据不

[收稿日期] 2021-01-24

[基金项目] 国家重点研发项目(2017YFD0700603)

[第一作者] 杨光友(1962—), 男, 湖北武汉人, 湖北工业大学教授, 研究方向为农业装备智能化与信息化

[通信作者] 谢 旺(1995—), 男, 湖北天门人, 湖北工业大学硕士研究生, 研究方向为农业装备智能化与信息化

同故障类型查询相应故障原因以及解决方案,实现故障快速定位和排除。

2 Android 移动端开发平台

2.1 硬件系统结构

考虑到系统开发的软件运行在 Android 移动端,选用 ARMv8-A 架构的 Cortex-A53 的四核处理器,以满足 Android6.0 系统的运行需求。Android 移动端需要与服务器通信,连接访问远程数据库。开发平台选用广泛应用于各种工业监控场合的 4G 无线传输模块 ME3630^[11]。4G 无线传输模块 ME3630 可以提供 50 Mbps 上行速率,以及 150 Mbps 的下行速率,并支持回退到 3G 或 2G 网络。该模组也支持 GPS 定位功能,完全符合系统功能要求。Android 移动端开发平台带有 OTG 接口,便于软件开发的调试。

Android 移动端开发平台硬件系统(图 2)主要由 CPU 处理器、触控显示屏、OTG 接口,4G 模块,12V/2A 主电源输入接口、蓝牙模块和 WIFI 模块等组成。

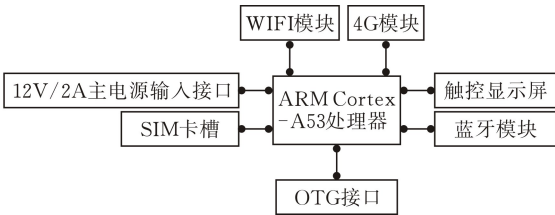


图 2 移动端开发平台硬件系统结构

开发平台硬件系统实物如图 3 所示。



图 3 开发平台硬件系统实物

2.2 软件开发工具

Android Studio 是谷歌专门为 AndroidAPP 开发设计的 IDE 工具。该工具基于 IDEA 设计,引进了新的项目构建工具 Gradle,优化了模拟器运行慢、内存大的问题,提供了更好的开发和调试环境。本文选用 Android Studio 3.4.2 作为 Android 移动端的软件开发工具。

3 Android 移动端软件开发

Android 作为一个以 Linux 为基础完全开源的移动设备操作系统,提供了丰富的开发资源。An-

droid 移动端软件基于 Android 平台开发,该平台支持 GUI 开发,能够提供良好的交互体验,同时还支持轻量级的 SQLite 数据库,可用作结构化的数据存储。Android 移动端软件的功能模块如图 4 所示。

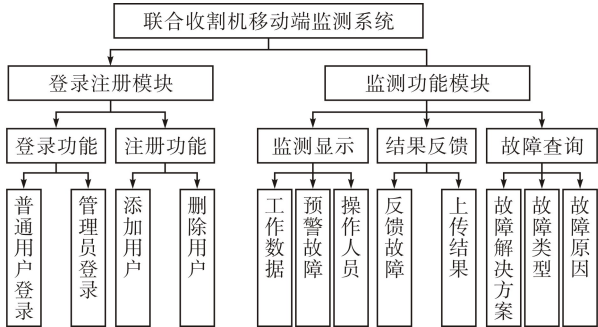


图 4 Android 移动端软件功能模块

3.1 Android 移动端软件开发关键技术

3.1.1 远程数据库连接 JDBC (Java Database Connectivity)是 Java 语言中用来访问远程数据库的应用程序接口,提供一种与平台无关的用于执行 SQL 语句的标准 javaAPI,方便实现多种关系型数据库的统一操作^[12]。Android 平台基于 Java 语言开发,可以通过添加第三方依赖库调用 JDBC 驱动,实现对远程数据的操作。本文选用 Mysql 数据库测试。

导入连接 Mysql 数据库的依赖包 mysql-connector-java-5.1.48-bin.jar 后,调用 Class.forName (com.mysql.jdbc.Driver) 程序语句初始化驱动类,调用 DriverManager 类的 getConnection() 与远程数据库建立连接。最后,通过调用 SQL 语句对数据库进行操作。该功能的具体流程如图 5 所示。

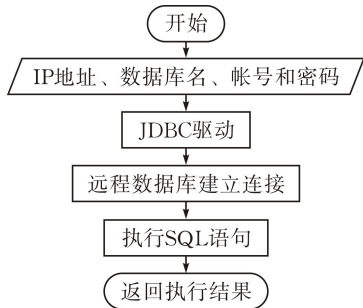


图 5 远程数据库连接流程

3.1.2 故障查询 远程服务端利用 Navicat Premium 数据库可视化管理工具,根据故障信息特点搭建故障数据库。创建的数据库包括有以下几列:编号 id、故障位置 error_location、故障类型 error_type、故障原因 error_reason、解决方案 error_solution。其中分为发动机、电气系统、割台装置、进料装置、脱粒装置、清选装置、集粮装置等 7 个部位。故障查询数据表结构如图 6 所示。

Android 移动端通过 JDBC 技术连接远程数据

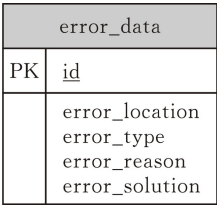


图 6 故障查询数据库

库,根据预警故障或反馈结果确定故障类型,通过该故障类型即可查询相应的故障原因以及解决方案。故障查询流程如图 7 所示。

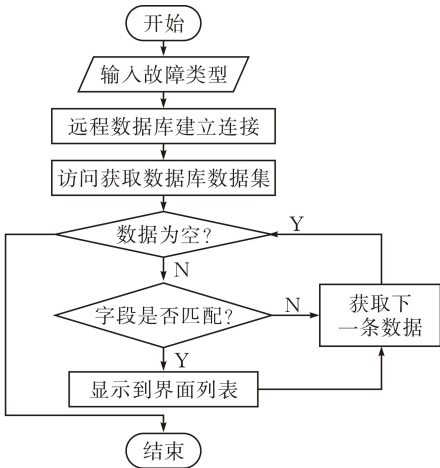


图 7 故障查询流程

3.1.3 故障反馈 接收到云服务端的预警结果为故障后,Android 移动端会提示操作员进行反馈。操作员需要根据实际情况进行判断,确定实际故障之后会将反馈结果和完成反馈的时刻一起记录保存。然后将故障 Id、故障数据的上传时间、故障数据的预警码、反馈结果和完成反馈的时刻一起转化成 JSON 字符串。通过连接 Redis 数据库,调用 rpush()方法存入 Redis 列表。云服务端通过该 Redis 列表获取 Android 移动端的反馈值。该功能实现流程如图 8 所示。

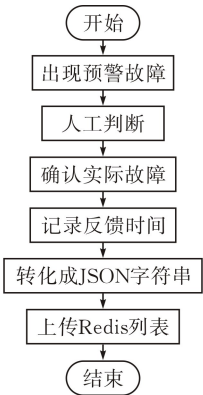


图 8 故障反馈流程

3.1.4 LitePal 开源数据库框架 LitePal 是一款开源的 Android 数据库框架。它采用了对象关系映射(ORM)的模式,并将我们平时开发最常用到的一

些数据库功能进行了封装,使得不用编写一行 SQL 语句就可以完成各种建表和增删改查的操作^[13]。LitePal 很“轻”,jar 包不到 100 kb,而且基本上零配置。使用 LitePal 开源框架可以让操作数据库更简单。LitePal 的使用方法如下。

1)引入 jar 包:打开 app/build.gradle 文件,在 dependencies 闭包中添加如下代码:

```
implementation 'org.litepal.android:core:2.0.0'
```

2)配置 litepal.xml 文件:在 app/src/main 目录下新建文件夹 assets,然后在 assets 文件夹中新建 litepal.xml 文件。配置文件内容如下:

```
<? xml version="1.0" encoding="utf-8" ? >
<litepal>
<dbname value = "leiwo"></dbname>
<version value = "1"></version>
<list>
<mapping class = "com.example.User"></mapping>
</list>
</litepal>
```

<dbname>标签用于指定数据库名称;<version>用于指定数据库版本号,当新添加了表要更新数据库时,将 version 中的 value 值加 1 即可;<list>标签用于指定所有的映射模型;每个映射模型用<mapping>标签声明,class 指定映射模型类名(完整类名),User 为一个模型类的名字,类中定义的变量对应着数据库表中的列。

3)配置 LitePalApplication:操作数据库需要用到 Context,LitePal 通过在 AndroidManifest.xml 中配置 LitePalApplication,简化了 Context 参数,使用数据库操作时就无需再传 Context。如以下代码所示:

```
<application
    android:name="org.litepal.LitePalApplication"
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".ResgiterActivity"></activity>
    <activity android:name=".MainActivity" />
```

4)初始化和操作数据库:每个 Activity 需要应用 LitePal 时,调用一次 LitePal.getDatabase()即可进行初始化。根据对象关系映射模式的理论,新建一个 User 模型类,类中元素对应数据库的数据表

User 中的列数据编号 id、账号 user、密码 password 和用户姓名 username。操作 User 数据表只需要分别调动模型类 User 对象的 user.save()和 user.delete()方法即可实现数据表的保存和删除操作。这样就可以应用 LitePal 框架对 Android 数据库进行账号信息的添加和删除,实现用户注册和用户注销功能。SQLite 数据表的结构模型如图 9 所示。

User	
PK	id
	user password user_name

图 9 数据表结构示意图

3.2 软件功能界面程序设计

Android 移动端的软件基于 Android 平台开发,利用 Android 平台强大的 UI 开发功能为软件各功能模块设计界面,实现良好的人机交互。

3.2.1 Intent 技术 Intent 技术作为 Android 开发的核心技术之一,用来协助各应用的交互和通讯。Intent 负责对应用中一次操作的动作、动作涉及数据、附加数据进行描述。根据 Intent 的描述,Android 负责找到对应的组件,将 Intent 传递给调用的组件,并完成组件的调用。Intent 不仅可用于应用程序之间,也可以开启一个 activity、发送广播消息和开启 Service 服务。调用 startActivity()方法跳转到指定 Activity,可以实现登录界面到注册界面和功能界面的跳转。还可以调用 putExtra()方法实现 Activity 间的数据传递,将 LoginActivity 获取的操作员信息传递到 MainActivity。

3.2.2 Button 控件 Button 是进行 UI 开发时用于与用户交互的重要控件。其中 Button 的点击事件功能是实现交互的关键,设置点击相应事件有实现 OnClickListener 接口、匿名内部类和定义内部类三种方法,用户根据需求点击相应的 Button。登录界面设置登录和注册两个 Button 触发 Activity 跳转事件。

3.2.3 Spinner 控件 Spinner 控件是一个列表下拉框,在用户点击之后弹出一个数据列表供用户选择。对于 Spinner 列表显示的数据源,有两种方法进行设定:一是通过 Adapter 设置,调用 xml 资源文件获取数据源通过 Adapter 绑定到 Spinner 控件显示,xml 资源文件结构如图 10 所示;二是通过布局文件在 Spinner 控件的设置中添加 android:entries="@array/error_data" 代码直接调用 xml 文件显示到界面。但是通过布局文件设置的 Spinner 不能对选择事件响应,不利于用户交互。软件界面

利用 Adapter 设置 Spinner,调用 AdapterView.OnItemSelectedListener()方法实现列表项被选择时触发响应事件。

```
<resources>
<string-array name="error_data">
<item>正常</item>
<item>脱粒滚筒轻微堵塞</item>
<item>脱粒滚筒严重堵塞</item>
<item>喂入搅龙轻微堵塞</item>
<item>喂入搅龙严重堵塞</item>
<item>输送链耙轻微堵塞</item>
<item>输送链耙严重堵塞</item>
<item>逐稿器皮带打滑</item>
</string-array>
</resources>
```

图 10 xml 文件结构

3.2.4 ScrollView 控件 ScrollView 是一种特殊类型的 FrameLayout。它可以使用户滚动显示一个占据的空间大于物理显示的视图列表。ScrollView 只能包含一个子视图或视图组,通常包含的是一个垂直的 LinearLayout。在 Activity 中添加 ScrollView 控件将整个页面转化成滚动视图,以便于浏览数据。

利用 Android 平台中的相关关键技术和 UI 控件进行软件界面设计,相关界面如图 11 所示。



图 11 软件界面

4 结束语

通过 Android 技术实现了移动端与远程服务器的交互,实时获取云服务器端的故障预警结果,并由操作员结合实际判断预警结果是否准确,实时将反

馈结果上传至远程数据库,为服务器端的模型更新提供准确的数据标签。利用云服务器端的故障数据库,终端系统可根据故障类型查询相应的故障原因和故障解决方案。该终端系统的应用可以移植于多种 Android 移动设备,比如 Android 智能手机和 Android 智能平板等,减少了传统 PC 端远程监测的局限性,提高了户外操作员故障监测和排除故障的实时性和便利性,具有很好的应用前景。

[参 考 文 献]

[1] 李哲,杨光友,陈学海.基于 B/S 架构的联合收割机远程监测平台研究[J].中国农机化学报,2019,40(4):151-157.

[2] 霍强.水稻联合收割机的现状及发展趋势分析[J].南方农机,2020,51(4):31

[3] 郭兵,方锡顺,周印富,等.国内外联合收割机的发展趋势分析[J].南方农机,2020,51(16):1-3.

[4] 高芬莉,耿朝阳,刘白林.Android 平台武器故障诊断专家系统设计与实现[J].西安工业大学学报,2019,39

(4):470-474.

[5] 范会敏,陈棚,张萌.基于 Android 的故障诊断专家系统设计与实现[J].石油管材与仪器,2015,1(4):21-24.

[6] 庄绪岩.飞机航电系统故障分析方法与故障诊断系统研究[D]. 广汉:中国民用航空飞行学院,2015.

[7] 乔美昀.基于 Android 平台的汽车故障诊断系统设计[J].装备制造技术,2018(8):60-65.

[8] 徐丽娜. 基于 Android 客户端的车辆实时监测和诊断系统的研究与实现[D].南京:东南大学,2018.

[9] 罗冰洋,陈苗,陈雅,等.基于 Android 平台汽车监测与诊断终端[J].计算机系统应用,2016,25(2):263-267.

[10] 刘洋. 基于 Android 平台的工业设备远程监测系统设计与实现[D].武汉:武汉理工大学,2016.

[11] 唐娜娜. 基于 4G-LTE 的脱机 LED 控制系统的设计[D].长春:中国科学院大学(中国科学院长春光学精密机械与物理研究所),2018.

[12] 王诚.研究 JDBC 连接 MySQL 数据库技术[J].网络安全和信息化,2018(12):68-69.

[13] 申圣兵.Android 开源数据库框架 LitePal 的研究与应用[J].中国新通信,2016,18(10):49-50.

Remote Fault Monitoring Terminal System for
Combined Harvester Based on Android

YANG Guangyou^{1, 2}, XIE Wang^{1, 2}, CHEN Xuehai^{1, 2}

(1 Agricultural Machinery Engineering Research and Design Institute,
Hubei Univ. of Tech., Wuhan 430068, China;

2 Hubei Province Agricultural Machinery Equipment Intelligent
Engineering Technology Research Center, Wuhan 430068, China)

Abstract: Aiming at the problem that the traditional PC-side fault query system is poor in portability and difficult to meet the needs of field operators, the advantages of the Android mobile platform, such as wide adaptability and strong interaction performance, are analyzed, and a remote fault of the monitoring and query system of the combined harvester based on the Android platform is proposed. The Android mobile terminal and the remote cloud server establish socket communication to obtain early warning of faults and determine the location of the fault based on the experience of the operator to realize remote fault diagnosis. Mysql database is used to realize the establishment of a fault query database, which can query the cause and solution of the fault. After testing, the mobile terminal has good human-computer interaction and portability, which can effectively improve the accuracy of fault diagnosis and the efficiency of troubleshooting, and has a good application prospect.

Keywords: Android; Mysql; fault query database; remote fault diagnosis; combined harvester

[责任编辑: 张 众]