

基于 Lora 的智慧农业移动端系统设计

周怡文¹, 张 伟¹, 沈琼霞², 王雪聰¹, 程中启¹

(1. 湖北大学 计算机与信息工程学院, 武汉 430062; 2. 烽火通信科技股份有限公司, 武汉 430073)

摘要: 针对现存农业物联网的各项问题, 如前期开发和后期维护成本高、系统复杂农业生产人员难以在实际生产中使用, 设计了基于 Lora 的智慧农业移动端系统; 依托于 Android 系统, 农业生产人员或研究人员只需在 Andorid 手机或其他 Android 终端安装软件, 即可实现远程对农业物联网的查看与控制; 移动终端以功能进行模块划分, 分别为用户管理、数据监控、设备控制和系统设置 4 个模块; 软件使用 MVP 架构, 将数据与界面交互进行分离; 使用 WebSocket 协议建立云服务与移动终端的全双工通信, 实现云服务对移动终端的消息推送, 使得用户可以实时取得环境信息以及底层设备的故障状态; 通过 Retrofit2 发送 HTTP 请求实现用户管理和对底层设备的运行控制; 使用 HLS 协议传输监控视频; 测试联调表明移动终端运行稳定可靠, 易于扩展。

关键词: 安卓系统; 智慧农业; 移动终端; Lora

Design of Intelligent Agricultural Mobile Terminal System Based on Lora

Zhou Yiwen¹, Zhang Wei¹, Shen Qiongxia², Wang Xuecong¹, Cheng Zhongqi¹

(1. School of Computer Science and Information Engineering, Hubei University, Wuhan 430062, China;

2. Service and CPE Business Unit, Fiberhome Telecommunication Technologies Co., Ltd., Wuhan 430073, China)

Abstract: Aiming at the existing problems of agricultural internet of things, such as high cost in the prophase development and maintenance, and difficult for the complex agricultural production personnel in practical production, the intelligent agriculture mobile terminal based on Lora is designed. With Android, agricultural producers or researchers can view and control the Internet of Things remotely by installing software on their Andorid phones or other Android terminals. The function of mobile terminal is divided into user management module, data monitoring module, device control module and system setting module. The software uses MVP architecture to separate data from interface interaction. The full duplex communication between cloud service and mobile terminal is established by WebSocket protocol. The message of the cloud service to mobile terminal is pushed. The environment information and fault status of the underlying equipment are acquired in real time by mobile terminal. The user management and operation control of the underlying device are realized by sending HTTP requests through Retrofit2. Use the HLS protocol to transmit surveillance video. The test results show that the mobile terminal is stable and reliable and easy to expand.

Keywords: Android system; intelligent agriculture; mobile termina; Lora

0 引言

传统农业生产主要依赖人工, 生产效率难以提高, 极易受环境气候的约束, 难以形成产业化、规模化^[1]。为了减小以上限制, 实现生产的现代化, 智能农业物联网应运而生。智能农业物联网依托物联网、大数据、云计算等技术^[2], 实现对农业生产、加工、销售等各个方面的优化。

现今, 农业物联网在我国的产业化仍处在进行中。当前农业物联网面临的主要问题是无论是前期开发还是后期维护都需要投入大量成本, 并且实际的农业生产人员难以准确使用农业物联网系统。基于 Android 平台的农业物联网系统, 通过 LoRa 无线网络结合以太网/4G 网络, 将感知层采集的传感器信息、实时监控视频等数据上行发送至农业

物联网智能云服务器^[3], 通过云服务器将数据分发至移动终端。Lora 网络成本低、可靠性高, 结合农业物联网智能云平台使得开发及维护成本降低。科研人员及农业生产人员通过 Android 手机、平板等移动设备远程查看并控制智能农业物联网, 避免了系统复杂难以操作等问题。

本文基于 Android 平台对农业物联网移动终端软件进行设计, 使用户能在具备网络的任何时间、任何地点对农业生产进行查看以及控制。

1 系统设计

1.1 系统整体设计

系统整体架构可分为感知层、传输层和应用层。智慧农业系统通过感知层收集农业大棚内的各项环境数据、监控视频, 感知层的数据使用 Lora 无线网络进行传输, 每个 LoRa 基站可以传输多个农业大棚内收集的数据。系统将感知层收集的数据经过有线、无线网络发往应用层。应用层包括云服务器、云数据库及移动终端。云服务器实现对上下行各项数据的处理, 云数据库实现对相应数据的存储管理, 移动端实现对数据的展示实现与用户的交互。系统整体架构如图 1 所示。系统支持对数据的上行下行双向传输。

收稿日期: 2019-10-11; 修回日期: 2019-10-26。

基金项目: 国家自然科学基金(61301144, 61601175)。

作者简介: 周怡文(1996-), 女, 湖北荆州人, 研究生, 主要从事物联网、移动互联网方向的研究。

通讯作者: 张 伟(1979-), 男, 湖北武汉人, 博士, 副教授, 硕士研究生导师, 主要从事无线通信、物联网、嵌入式系统方向的研究。

从下自上时,传感器收集各个农业大棚的环境数据,经过 Lora 自组网络发往 Lora 基站, Lora 基站经过有线、无线、3G/4G 网络传输将数据发送至云服务器,云服务器对数据进行处理,用户通过移动客户端实现对数据的访问。自上而下时,用户通过移动端发出指令,云服务接收指令并处理发放,云服务器通过有线、无线、3G/4G 网络将指令发往 Lora 基站, Lora 基站通过 Lora 网络将指令发往对应控制设备。

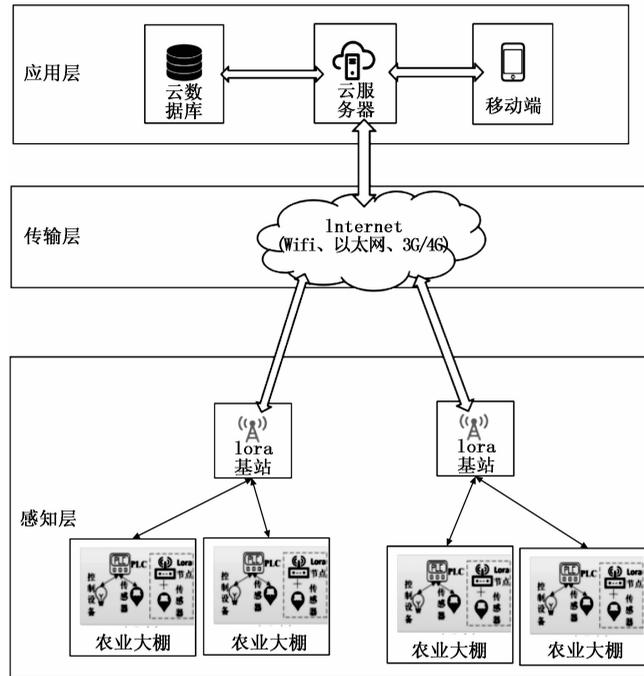


图 1 系统整体架构

1.2 通信协议设计

系统上下级以不同的协议传输数据。Lora 节点与 Lora 基站以 Lorawan 协议传输数据, Lora 基站与云服务器以 tcp/ip 协议传输数据, 云服务与移动端以 http, websocket 等协议传输数据。系统设计 5 种通用帧结构用于系统数据的上行下行传达。分别为上行数据帧、上行到达应答帧、上行执行应答帧、下行控制指令帧和下行设备扫描帧。其中上行数据帧携带感知层采集的各项数据; 上行数据帧发出后, 上行到达应答帧向下层反应数据到达情况; 下行控制指令帧携带从移动端发出的指令; 指令到达下层, 上行执行应答帧向上层反应指令到达情况; 下行设备扫描帧用于启动感知层各个各个设备。

Lora 基站、云服务器和移动端等各层的通信协议上荷载通用数据帧, 每层对数据帧进行拆包、数据处理、装包然后发出。通用数据帧格式如图 2 所示, 包括开始、类型、站点 id、大棚 id、设备类型、数据内容、校验位和结束字节。

2 移动端设计

2.1 移动端系统设计

Android 平台发展迅速, 2019 年安卓系统的份额已经

| 开始 | 类型 | 站点ID | 大棚ID | 设备类型 | 有效数据 | 校验位 | 结束 |
|-------|-------|-------|-------|-------|--------|-------|-------|
| 1byte | 1byte | 2byte | 2byte | 2byte | 不定byte | 2byte | 1byte |

图 2 数据帧格式

达到 87%。本文旨在实现移动终端与云平台之间的数据交互, 使得用户在远程状态下可以通过网络对农业生产数据进行访问和控制, 即实现一个 C/S 架构的应用系统。

基于 Android 平台的移动终端是用户对农业物联网进行操作的入口。依据主要功能可以划分为用户管理、数据监控块、设备控制和系统设置共 4 个模块。

用户管理模块实现用户登陆与用户权限管理。数据监控模块采集并实时地显示农业大棚外内的光照强度、风力大小、降水强度、温度、湿度、二氧化碳浓度等影响农业生产的环境数据, 各个设备运行状态以及监控视频。设备控制模块通过数据采集模块采集到的各项数据对农业大棚中生产设备^[5], 包括对内外遮阳、顶通风、水帘补水、苗床灌溉、肥料搅拌、施肥泵、增氧机、照明和热风机等, 进行手动或者自动控制, 并对各种设备进行添、删除等管理操作。系统设置主要是对农业大棚的各项参数如环境数据阈值、设备控制参数进行设置。功能模块如图 3 所示。

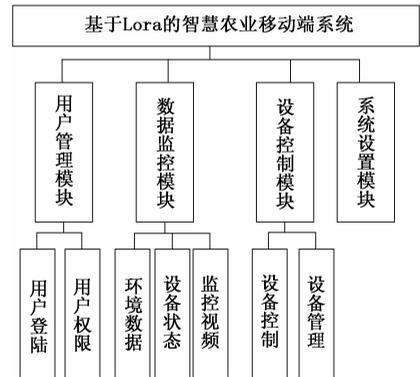


图 3 移动端功能模块

2.2 开发环境

云平台服务器使用 JAVA 语言进行开发, 使用 jdk8 在 IDEA 软件进行开发, 使用 mysql 数据库, 1 核 cup、2G 内存的阿里云服务器。

Android 移动终端使用 JAVA 语言进行开发, 使用 jdk8 在 Android studio 软件进行开发。调试环境为一台 Android8.1.0 系统版本的小米手机。

3 功能模块实现

3.1 用户管理模块

用户登陆模块需要实现用户登陆和用户注册功能^[6]。当用户登陆时, 移动终端发起请求时, 云服务器接收到请求, 并在云数据库中查找用户信息进行验证。第一次使用的用户需要进行注册。移动端使用 MVP 架构使得登陆注册的界面与数据的交互分类。MVP 架构可以分为以下三个层面:

View 层: Android 的 Activity 或 Fragment 通常用作 View 层, 负责对登陆注册页面进行数据展示, 便于实现界面交互功能。

Model 层: 即图 3 中的 Repository, Model 层将通过网络请求云服务器, 取得数据库信息, 验证用户登陆。

Presenter 层: 对 View 层和 Model 层进行逻辑连接与处理。

View 层和 Model 层不直接进行交互, 在 Presenter 层以面向接口编程方式实现功能。View 层和 Model 层以接口形式引入到 Presenter 层。Presenter 层将 View 层和 Model 层隔离, 并将具体实现代码交由 Presenter 接口实现类处理, 保证 View 层和 Model 层解耦合, 易于测试和维护^[7]。其实现过程如图 4 所示。

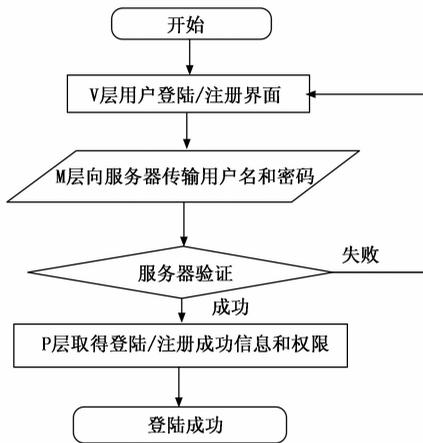


图 4 用户登陆/注册流程图

3.2 用户权限管理

智慧农业移动端系统拥有查看各项环境、设置环境参数阈值、查看监控视频、查看设备运行状态、控制设备运行状态及管理设备等功能。不同用户的使用需求不同, 并且出于对智慧农业移动端系统的安全保障, 需对用户进行权限分类管理。普通用户可以查看当前数据监控模块中的环境数据和监控视频。一级管理员可以查看数据监控模块的所有数据并拥有设备控制操作权限。二级管理员拥有设备控制管理权限和系统设置权限。三级管理员可对账户管理、站点管理等进行操作。每级管理员拥有当前及以下等级用户权限。权限管理流程如图 5 所示。

3.3 数据监控模块

数据监控模块包括环境数据、监控视频和设备运行状态监控, 图 6 为数据监控模块流程图。

视频监控使用 HLS 协议, 延时在可接受范围内。HSL 协议的特性使得客户端在不同的网络情况下, 可以依据网速对视频流在不同的码率中进行切换, 使用户能更方便在远程查看农业大棚中情况。监控视频在移动终端上通过 HTML5 播放, HTML5 的 VIDEO 标签支持 HLS 协议^[8]。

环境数据监控和设备运行状态监控使用 WebSocket 协议。感知层将底层的各项数据通过网关/DTU 发送到云服

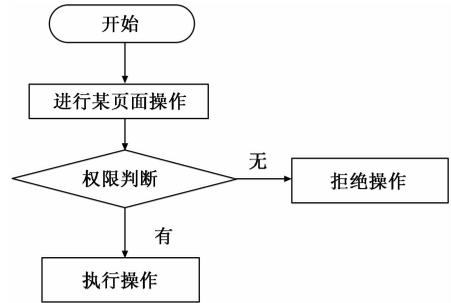


图 5 权限管理流程图

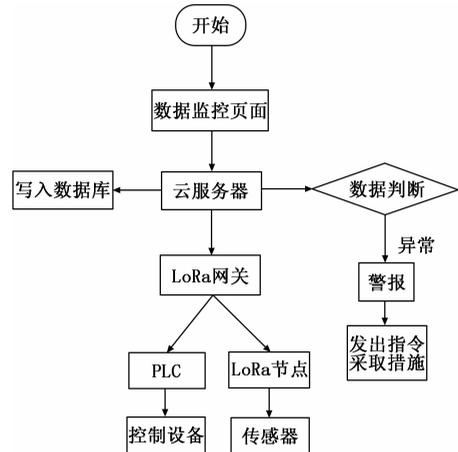


图 6 数据监控流程图

务器。云服务器将收到的数据解析为不同的环境信息, 并持久化到数据库, 同时通过 WebSocket 协议与移动终端传输信息。

WebSocket 协议是 HTTP/1.1 协议的升级, 可以实现基于服务器推送并保持长连接^[9]。在移动终端对服务器发起请求时, 服务器读取 HTTP 请求首部, 读取相应的 WebSocket 请求字段, 服务器将请求识别为 WebSocket 请求并与移动终端建立连接。

云服务器端实现 WebSocket 的 onopen, onmessage, onclose 和 onerror 消息。当成功接收到一个 WebSocket 请求后, 建立一个 WebSocket 连接接收 onopen 消息并将这个 WebSocket 连接加入队列。当 WebSocket 连接接收 onclose 和 onerro 消息^[10], 表明连接出现错误或已经关闭。服务器将当前 WebSocket 连接从连接队列中移除。当收到传感器信息时发送当前队列所有 WebSocket 连接的 onmessage 消息, 将传感器信息发送至移动终端。

移动终端在用户登陆成功后, 立刻发送一个 Websocket 请求建立与云平台建立连接, 等待推送消息。为了保证 WebSocket 连接能持续运行在后台, 移动终端使用 Android 四大组件之一的 Service 组件。通过 Service 组件保持 WebSocket 连接的使用方法如下:

1) Activity 组件在收到服务器返回的登陆成功信息后, 通过 startService () 启动服务, Service 服务在生命周期的 onStartCommand, 服务开始运行。

2) Service 组件在 Android 系统中优先级比较低, 当内存不足时会被系统回收。通过 startForeground () 开启前台服务, 将 Service 通过通知的方式运行在后台。

3) 在 onStartCommand () 中对 WebSocket 连接进行初始化并发起连接。当 WebSocket 接收到服务器发送的 onMessage 消息, Service 组件发送广播, 向展示环境信息的 Activity 和 Fragment 发送消息。

4) 当用户退出登陆, Service 进入 onDestroy 状态, WebSocket 发起 onClose 消息, 断开 WebSocket 连接。

3.4 设备控制模块

设备控制模块包括设备控制和设备管理, 设备控制模块流程如图 7 所示。

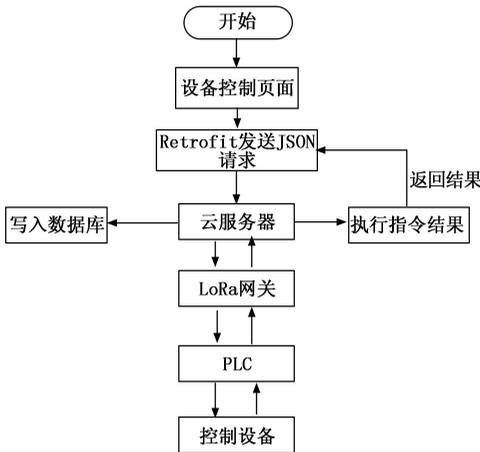


图 7 设备控制流程图

设备控制主要实现移动端控制用于农业生产的设备, 如灯光、通风、水帘、肥料搅拌等。当用户通过移动终端对任意一个控制设备进行操作时, 移动终端会对云服务发送一个 POST 请求, 并带有操作类型 id 的 JSON 请求体。

设备管理主要实现对设备的添加、删除等操作。在实际农业生产的不同时期, 对各类设备的需求有所不同, 因此涉及到设备的添加、删除管理。移动终端会对云端发送 POST 请求, 发送带有操作设备 id 和操作类型的 JSON 请求体。

移动终端的网络请求架构使用 Square 公司的 Retrofit2。Retrofit2 是常用于移动端开发的 OkHttp 网络请求框架的升级。通过 Retrofit2 构建网请求的方法如下:

- 1) 创建一个实体类, 用于接收服务器端发送的数据。
- 2) 创建一个用于 Http 请求的 java 接口, 接口标注表示 Http 请求方法和参数的注解。方法注解与请求方法同名, 用户登陆使用 POST 请求方法, 用于登陆的接口使用 @POST 标注。接口使用 @Body 标注为非表单请求体, 请求体为 JSON 类型。
- 3) 构建网络请求实例。创建 Retrofit 事件, 设置网络请求的 URL (统一资源定位符) 设置 OkHttp 请求的连接超时等参数, 设置 Gson 数据解析器用于处理服务器返回的数据, 通过 retrofit.create (ApiService.class) 创建网络

请求。

4) 转换线程, 发送网络请求, 并对服务器返回数据进行处理。

3.5 系统设置模块

系统设置模块主要实现对底层设备包括数据监控设备和控制设备进行初始化设置和各项参数调整。对环境数据可以进行采集频率设置, 采集频率包括秒、分钟、小时等单位; 对数据精度进行设置, 如对光照强度单位设置; 对环境数据阈值进行区间设定, 当高于或低于阈值区间时, 云服务器向移动终端推送警报信息。对设备控制可以设置设备的运行情况, 如运行速度、运行方向等。根据生产人员的经验, 在系统运行初期进行初始化系统设置。系统进入稳定运行后, 分析已产生的数据对系统参数进行相应的调整以根据满足生产。系统设置流程如图 8 所示。

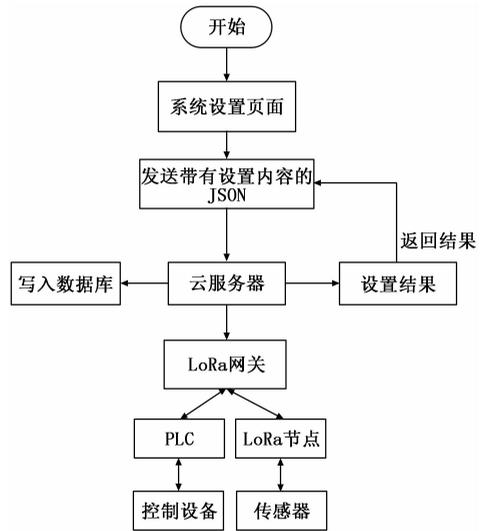


图 8 系统设置流程图

移动终端会依据系统设置内容通过 Retrofit2 向服务器发送 http 请求, 请求体为包括设置设备和设置内容的 JSON 字符串。

4 系统测试

4.1 系统功能测试

移动终端主要测试对服务器的请求能否正常提交, 云服务器是正常接收传感器数据并通过 WebSocket 与移动终端进行连接各项传输数据, 视频监控监控能否实时传输到移动终端, 移动端控制指令能否正确发送。

测试时以测试用户 test 登陆, 如图 9 所示。当 test 用户登陆成功时, 移动终端开启与云服务器的 WebSocket 连接。如图 10 所示, 建立 WebSocket 连接成功, 移动终端接收的消息可以在数据监控模块查看, 表明数据上行成功。

移动终端可以对底层多个控制设备进行多项操作。如图 11 所示, 测试用户控制外遮阳 1 开启正转, 说明数据下行成功。

如图 12 所示, 移动终端可以在任意时间查看监控视频。

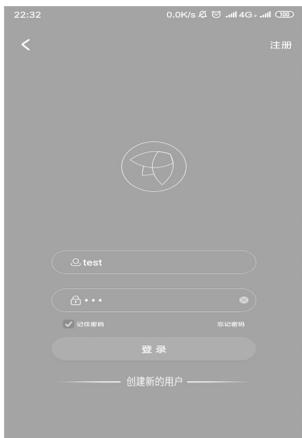


图 9 test 用户登陆

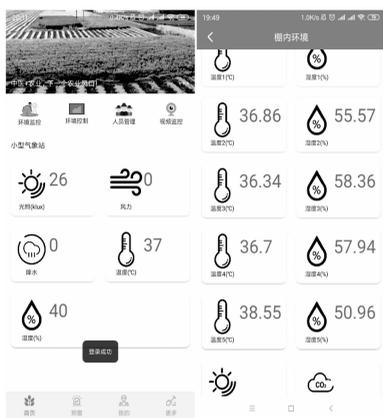


图 10 部分环境数据

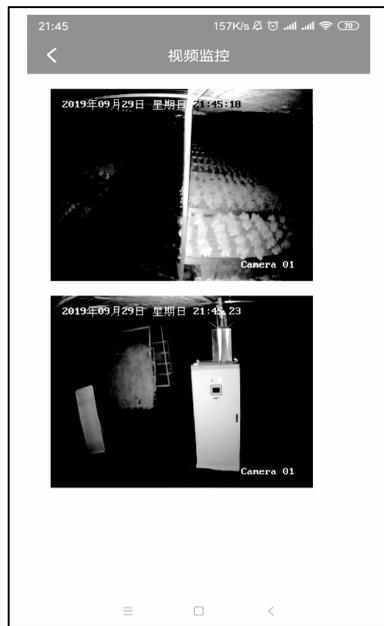


图 12 视频监控夜间情况

表 1 采集频率测试结果

| 组数 | 时间间隔/s | 误码率/% |
|----|--------|-------|
| 1 | 182 | 0 |
| 2 | 184 | 0 |
| 3 | 179 | 0 |
| 4 | 181 | 0 |

从移动端发起对控制设备的操作请求，记录每组请求从发起请求到完成操作的平均时间，测试结果如表 2，用户从移动端发起指令到底层设备响应延时在合理范围以内，用户使用感流畅。

表 2 设备控制延时

| 组数 | 响应时间/ms | 请求次数 | 误码率/% |
|----|---------|------|-------|
| 1 | 1 292 | 100 | 0 |
| 2 | 1 273 | 100 | 0 |
| 3 | 1 156 | 100 | 0 |
| 4 | 965 | 100 | 0 |
| 5 | 933 | 100 | 0 |

测试结果表明，移动终端对服务器的请求，与服务器的 WebSocket 连接以及视频监控的传输等功能都正常运行，能够满足农业物联网的各项需求。

5 结束语

本文旨在设计基于 Lora 的智慧农业 Android 移动终端系统。移动终端实现功能帮助农业生产人员或研究人员，在任何时间、任何地点可通过数据采集模块查看农业大棚的各项环境数据，通过视频监控模块掌握作物的生长情况。通过对各项环境数据实时情况的分析，农业生产人员或研究

(下转第 248 页)

4.2 系统性能测试

系统功能测试主要测试 websocket 连接传输数据的频率和移动端发起 http 请求控制底层设备的延时。

在系统设置页面将数据采集设备的设采集频率设置为 3 分钟。以一天中四个时间段测试，记录每个时间段平均时间间隔，测试结果如表 1 所示。



图 11 外遮阳正转指令