

原创性声明



本人声明：所呈交的学位论文是本人在导师的指导下进行的研究工作及取得的研究成果。除本文已经注明引用的内容外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得内蒙古大学及其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名： 赵鑫 指导教师签名： 李席
日 期： 2010.5.19 日 期： 2010.5.20

在学期间研究成果使用承诺书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：内蒙古大学有权将学位论文的全部内容或部分保留并向国家有关机构、部门送交学位论文的复印件和磁盘，允许编入有关数据库进行检索，也可以采用影印、缩印或其他复制手段保存、汇编学位论文。为保护学院和导师的知识产权，作者在学期间取得的研究成果属于内蒙古大学。作者今后使用涉及在学期间主要研究内容或研究成果，须征得内蒙古大学就读期间导师的同意；若用于发表论文，版权单位必须署名为内蒙古大学方可投稿或公开发表。

学位论文作者签名： 赵鑫 指导教师签名： 李席
日 期： 2010.5.19 日 期： 2010.5.20

奶源运输安全远程监控系统监控中心的设计与实现

摘要

在“阜阳奶粉”、“三鹿奶粉”事件之后，人们越来越关注乳品企业奶制品食品安全。食品安全的主要环节是食品原材料，为了有效的监控乳品企业奶站奶罐车运输过程，防止奶站因注重经济效益，途中非法添加非食品添加剂，本文研究了奶源运输安全远程监控系统。

本文首先介绍了课题的背景及意义，其次详细介绍了远程无线数据传输 GPRS 原理、全球定位 GPS 原理、MapX 控件技术的基本理论，然后详细阐述了奶源运输安全远程监控系统监控终端、监控中心的设计。

本文对监控中心数据库的建立、监控终端采集数据的导入、应用服务器中部分重要函数的构建、客户端各部分的设计作了重点探讨，主要工作有：采用基于 MIDAS 技术的三层体系结构开发模式构建了监控中心软件；根据本系统的应用目的及监控终端采集的数据，详细设计了监控中心数据库；利用 Object Pascal 语言编写了应用服务器函数，实现了应用服务器对客户端的各项业务服务以及数据请求服务进行封装，以函数接口调用的方式，向客户端提供业务处理；利用面向对象的 Delphi 开发环境，建立了客户端可视化监控界面，实现了对监控终端信息和奶站信息的管理；应用具有强大地图分析功能的 MapX 控件技术，实现了最基本的地图操作工具，奶罐车电子地图定位功能；完成接收图像的恢复及显示。

关键词：GPS 定位，GPRS，远程监控，MapX，图像采集

DESIGN AND IMPLEMENTATION OF MONITORING CENTER
OF THE MILK SOURCE TRANSPORT SAFETY
REMOTE MONITORING SYSTEM

ABSTRACT

After the events “Fuyang milk powder” and “Sanlu milk powder”, the people more and more attention to food safety of milk-maded products. The raw material is the key link of food safety, in order to effectively monitor milk tankers transportation process of enterprises’ milk stand to prevent milk stands by the way to economic adding illegal source of food ingredients, this article study the milk source transport safety remote monitoring system.

This article firstly introduced the background and significance of the subject, secondly detailed introduced a wireless remote data transmission GPRS principle, the global positioning GPS principle, Mapx activex technology's basic theory, and elaborated on the design for the terminal and monitoring centre of the milk source transport safety remote monitoring system.

This article mainly discussed the database for monitoring centre , the importing of the monitoring terminal’s collected data, the building of an application server’s important functions, the various parts design of the client. the main works contain: builded the monitoring centre by using three-tier architecture development patterns

based on the midas technology; according to the application purpose of this system software and the collected data of monitoring terminals, detailed designed the database of monitoring centre ; using Object Pascal language programming the functions of the application server , carried out packing the client's business services and request services to the application server, provided business deals to client by the way of calling functions ; using the object-oriented delphi development environment, builded the client visual interface to implement the management of monitoring information and milk stand information ; using the powerful map-analysis-function activex technology mapx, realized the basic operation of the map and the tankers' electronic maps location; recovered pictures and displayed.

KEYWORDS: GPS, GPRS, Remote monitor, MapX, Image acquisition

目 录

第一章 绪论	1
1.1 课题研究背景	1
1.2 远程监控系统的发展现状	2
1.3 本课题的主要任务及本论文的主要内容	4
第二章 奶源运输安全远程监控系统的总体设计方案	6
2.1 远程监控系统的功能框架	6
2.2 系统软硬件实现方案概述	7
2.2.1 系统的硬件实现方案	7
2.2.2 系统的软件实现方案	7
第三章 GPRS 技术简介	8
3.1 GPRS 技术原理	8
3.1.1 GPRS 概述	8
3.1.2 GPRS 数据传输原理	9
3.1.3 GPRS 提供的业务服务	12
3.2 GPS 技术原理	12
3.2.1 GPS 的概念	12
3.2.2 GPS 系统的组成	12
3.2.3 GPS 系统的定位原理	13
3.2.4 GPS 系统的特点	16
第四章 MAPX 控件技术介绍	17
4.1 MapX 概述	17
4.2 MapX 的数据文件和组件模型结构	17
4.2.1 MapX 的数据文件	17
4.2.2 MapX 的组件模型结构	18
4.3 MapX 的基本概念介绍	18
4.4 MapX 的特点	19
第五章 监控终端的硬件设计与实现	21

5.1 监控终端的组成	21
5.2 监控终端硬件资源介绍	22
5.2.1 微处理器 TMS320LF2407A DSP	22
5.2.2 GPRS 模块	22
5.2.3 C328 摄像头模块	22
5.2.4 GPS OEM SL-SDT11 板	22
5.3 监控终端主要模块的设计与实现介绍	23
5.3.1 GPS 模块功能的实现	23
5.3.2 C328 模块功能的实现	25
5.3.3 GPRS 模块功能的实现	26
第六章 奶源运输安全远程监控系统监控中心的设计与实现	29
6.1 监控中心的模式选择	29
6.2 监控中心的开发方案	29
6.2.1 软件实现功能详解	29
6.2.2 开发方案的选择	30
6.3 系统开发工具介绍	34
6.3.1 Delphi 介绍	34
6.3.2 Object Pascal 软件开发语言介绍	35
6.4 数据库服务器的设计与实现	36
6.4.1 数据库的建立	36
6.4.2 远程监控数据表格的设计	38
6.4.3 监控终端采集数据的导入	43
6.4.4 ADO 数据库访问技术	45
6.4.5 ADO 数据库连接参数配置	46
6.5 应用服务器功能的实现	48
6.5.1 应用服务器的建立	49
6.5.2 应用服务器的逻辑功能	51
6.5.3 应用服务器部分函数接口的实现	51
6.6 客户端的设计实现	55

6.6.1 客户端登录功能的实现	55
6.6.2 客户端主界面设计实现	57
6.6.3 客户端“终端管理”功能的实现.....	58
6.6.4 客户端“奶站管理”功能的实现.....	60
6.6.5 客户端“地图工具”的实现.....	61
6.6.6 客户端“地图导航”功能的实现.....	63
6.6.7 客户端界面节点树的设计实现	64
6.6.8 客户端“远程监控”功能的设计实现.....	66
6.6.9 客户端电子地图显示功能的实现	67
6.6.10 客户端图像显示及管理功能的实现	68
6.6.11 连接状态查看功能	70
6.6.12 Borland Socket Server 程序配置方法.....	71
第七章 总结与展望	72
致 谢	73
参考文献	74
附录 1 应用服务器部分函数接口代码	77
附录 2 客户端定位显示程序代码	87

第一章 绪论

1.1 课题研究背景

随着人们生活水平的不断提高，食品安全日益得到重视，饮食方面人们越来越注重食品的安全性、可靠性。

在“阜阳奶粉”、“三鹿奶粉”事件之后，国产乳业陷入“三聚氰胺”危机。尤其乳业是我区的特色产业，对于有“乳都”之称的呼和浩特，保证奶制品食品安全的问题迫在眉睫，亟待解决。

食品安全的主要环节是食品原材料。我国关于食品安全的监管分多个环节，包括农业、质检、工商、卫生等部门。监管资源不集中，连接环节出现脱节，直接导致监管的不全面。加之，监管资源匮乏，监管手段单一，致使乳品企业的牛奶提供者单纯注重经济效益，忽略控制牛奶质量，甚至非法添加非食品添加剂。要解决这个问题就必须对乳品企业奶源获取过程进行严厉监管。这些危及食品安全的监管问题如果得不到解决，必定会危及我区乳业经济的发展、自治区人民的根本利益和饮食健康。这也是本系统研究的原因及出发点。

本系统的初始设计思路是充分依托现有的移动通信传输技术及 GPS 技术，充分利用现有的通信资源和网络资源，实现对奶源运输过程的远程图像监控和远程目标定位。

近年来，远程图像监控和远程目标精确定位，以其直观、准确和信息量丰富，被广泛应用于各种场合，成为安全监控的主要手段之一。随着网络技术迅速发展，它的应用领域也在不断扩大和深化。GPRS 是技术成熟、网络覆盖广的无线数据传输技术；GPS 全球定位系统，目标定位准确。这些技术的成熟使得基于 GPRS 和 GPS 技术实现远程实时图像监控和精确定位成为可能。

针对于乳制品企业奶罐车运输过程的监控，本系统实现了监控系统的可视性、动态性、网络智能性，提高监测数据采集和传输的实时性、可靠性，设计了 GPS 技术、GPRS 技术和网络技术相结合的奶源运输安全监控系统。该系统是以奶罐车空间位置信息，停顿点拍摄图像信息为现场采集数据，利用 GPRS 传输技术将信息传至监控中心，在监控中心将奶罐车空间位置信息以电子地图形式还原显示，并将停顿点图像信息恢复显示。

1.2 远程监控系统的发展现状

远程监控系统是在传统监测监控系统的基础上,结合当前无线通信技术和信息处理技术而发展起来的测控系统^{[1][2]}。

远程监控系统一般包括三个部分:监测站、无线网络和控制中心软件^[2]。控制中心是整个系统运作的核心,负责收集各监测站上传的监测信息,发送各种操作命令以控制监测站的行为。监测站被布放于远离控制中心的各监测点处,负责完成信息的采集和响应控制中心发出的控制命令。控制中心可用普通微机、工作站或工控机实现,软件开发可靠基于现有的 Windows 或 Unix 操作系统。监测站的设计实现可根据不同的应用目的和应用环境,采用特定的技术形式,比如单片机、DSP 或者 Intel X86 系列的微处理器等。无线远程监控系统的组网方式也很灵活,可利用现有的无线通信网,如 GSM/GPRS 网络,CDMA 移动网络等,也可单独搭建专门的无线局域网。

以下为现有的远程监控系统各部分的实现方案。

监测站的设计与实现是整个无线远程监控系统研制开发的重点,监测站对信息数据处理的能力和精度将影响整个系统的最终性能。在整个开发过程中,监测站的设计是工作量最大、所需时间最长的一部分。监测站处于工作现场,只完成数据的采集、处理和传输,任务相对单一、固定,无须用较大的台式机来完成;考虑到节能和布放方便,监测站多为嵌入式系统。根据整个无线远程监控系统所要实现的功能,和对数据处理与对传感器控制能力的要求,监测站设计的复杂程度和采用的具体技术是不一样的。

采用单片机是大多数嵌入式系统设计时的首选方案。由于在片上集成有丰富的外设,具有良好的控制能力,单片机天生就是为嵌入式系统度身定做的,在嵌入式市场上占据了最大的份额。

基于单片机的设计方案一般适用于对数据处理要求不高,运算量不大的远程监控系统。根据需要,单片机可以选用较为低端的 4 位机或 8 位机,如 8051 等,也可选用功能较强的专用芯片,如 MSP430FE42X 系列。单片机主要用于监测站端的系统控制。片外存储器一般为 RAM、EEPROM 和 Flash 等存储器;I/O 设备一般为键盘、LCD 等供设计调试用的人机交互接口;传感器一般为话筒、摄像头、扬声器和伺服马达一类的设备。无线通信接口实现相对较为复杂。编解码器是可取舍的,对于低速率数据一般没有必要。根据系统的处理任务和信息的类别,编解码器可选用不同的芯片,如 CMX639 (用于音

频)或 LD9320 等,也可用编程逻辑器件实现。监测站软件可直接通过 C 或汇编语言实现,也可在实时操作系统上开发应用软件。对于低档的 4 位或 8 位单片机,控制能力较低,系统简单,一般采用直接编写控制程序的方法。对于功能较强大,各设备间交互复杂的系统而言,大多数是利用操作系统来进行任务管理、设备交互,应用软件只是完成上层的数据处理等工作。

随着 DSP 技术的日益成熟,现在越来越多地监控终端采用 DSP 的设计实现。DSP 的数字处理方面能力较强,技术已经很成熟,能处理各种运算的通用、专用芯片也很多。以 DSP 为核心设计开发的监测站,可以完成高速率数据处理,保证系统实时性方面的要求。这类设计方案一般适用于数据处理运算量比较大,实时性要求高而对控制能力要求相对较低的监控系统。与以单片机为基础的监控系统不同的是,DSP 除了作控制器以外,还可兼作数据计算、编/解码之用。对于较复杂的编/解码以及压缩解压运算(比如对图像视频数据的处理等)是否仍由 DSP 完成,须综合考虑。若 DSP 在系统控制和实现传输协议方面负担太重,则这部分运算需要由专门的处理芯片完成;若系统控制和传输协议较简单,或根本没有到上层协议栈,则这部分复杂的运算可由 DSP 完成。

在远程数据传输部分,目前使用最多的是利用现有网络实现监测站与监控中心的通信。现有的通信网络较多,按业务建网是 3G 以前通信网络的特点,无线网络也不例外。设计无线远程监控系统可以借用的无线网络主要有:全球数字移动电话系统(GSM)、通用分组无线业务(GPRS)、采用码分多址(CDMA)技术的移动网、蜂窝式数字分组数据(CDPD)系统。

GSM (Globem System for Mobile)是全球最主要的 2G 标准,能够在低服务成本、低终端成本条件下提供较高的通信质量。就其业务而言,GSM 是一个能够提供多种业务的移动 ISDN (Integrated Services Digital Network,综合业务数字网络)。

GPRS (General Packet Packet Radio Service)在现有的 GSM 网络基础上增加一些硬件设备和软件升级,形成一个新的网络逻辑实体。它以分组交换技术为基础,采用 IP 数据网络协议,提高了现有的 GSM 网的数据业务传输速率,最高可达 170kb/s。GPRS 把分组交换技术引入现有 GSM 系统,使得移动通信和数据网络合二为一,具有“极速传送”、“永远在线”、“价格实惠”等特点。

CDMA (Code Division Multiple Access)网络采用扩展频谱技术,使用多种分集接收方式,使其具有容量大、通信质量好、保密性高和抗干扰能力强等特点。

CDPD (Cellular Digital Data) 无线移动数据通信基于数字分组数据通信技术, 以蜂窝移动通信为组网形式, 是数据网与移动通信的结合物。这种通信方式基于 TCP/IP, 系统结构为开放式, 提供同层网络无缝连接和多协议网络服务。CDPD 网络具有速度快、数据安全性高等特点, 可与公用有线数据网络互联互通, 非常适合传输实时、突发性和在线数据。

控制中心的设计主要集中在应用软件的开发上, 一般是基于 Windows 和 Unix 等常用的操作系统。当前用于此类软件开发、调试的工具较多, 且功能强大, 给控制中心软件的设计带来便利。

就软件的实现形式而言, 一般除了界面模块外, 其余各个功能模块均可设计成动态连接库文件 (.dll)。人机接口界面模块可以为该无线远程监控系统的实际应用进行定制, 以满足用户在界面美观、操作方便等方面的特殊要求。

采用 C/C++ 语言在 VC 开发环境下设计这样的系统软件涉及到的技术较多, 包括内存管理、网络通信、多线程管理和数据库编程, 甚至 ActiveX 等。

1.3 本课题的主要任务及本论文的主要内容

本课题的设计任务是: 设计一套可以在呼和浩特市区内实时监控乳品企业奶罐车行车路线及沿途情况的监控系统。

系统设计要求:

- 硬件方面:
- 1、设计监控终端, 实现对车辆空间位置数据的采集并将数据传送至监控中心。
 - 2、可以实现当车辆停车时对沿途情况进行拍照, 并将拍摄图像传送至监控中心。
- 软件方面:
- 1、编写监控中心软件, 实现软件与终端系统的通信功能。
 - 2、恢复监控终端所采集的数据, 显示图像, 显示奶罐车空间位置。
 - 3、实现控制中心软件对采集信息的管理功能。

本论文的主要内容: 本文主要阐述了依托 GPRS 技术、GPS 技术进行远程监控的实现过程。重点是完成了远程监控系统控制中心软件的设计与实现。

在此系统的设计与实现中主要做了以下几方面的工作:

1. 掌握 GPRS 系统的结构及其传输原理;

2、掌握 GPS 技术的原理；

3. 使用面向对象的 Delphi 开发环境，及 Object Pascal 语言完成应用服务器和客户端的构建，实现管理人员使用客户端软件即可方便快捷的在局域网内实现对远程终端的监控；

4、对 SQL Server 数据库及数据库访问技术 ADO 技术进行研究，并运用这些技术完成控制中心数据库服务器的建设；

5、通过对 C328 摄像头的采集图像数据格式的分析，完成接收图像的重现；

6、掌握 MapX 开发地理信息应用系统的技术途径和解决方案，实现了在客户端对奶罐车空间位置的地图显示。

本论文各章内容安排如下：

第 1 章概述了本课题的研究背景、意义及远程监控系统的发展现状，并且介绍了本论文的主要内容。

第 2 章阐述了奶源运输安全远程监控系统的总体设计方案。

第 3 章介绍了 GPRS 原理和 GPS 原理，包括 GPRS 系统的概念、数据传输过程、GPRS 的业务、GPS 系统的概念、GPS 定位原理及 GPS 系统的特点。

第 4 章介绍 MapX 控件技术。

第 5 章介绍监控终端的硬件平台的构建，硬件设备的性能、特点及硬件主要模块功能的设计实现。

第 6 章具体阐述奶源运输安全远程监控系统监控中心的设计与实现，对实现监控中心各个功能模块所用到的开发技术，各模块的实现过程及功能进行了介绍，并给出了各功能模块的实现结果图和代码。

第 7 章对论文的工作进行了总结，并且提出了系统的不足之处和进一步深入研究的方向。

第二章 奶源运输安全远程监控系统的总体设计方案

2.1 远程监控系统的功能框架

远程监控系统一般由前端系统、数据传输系统、控制中心三个部分组成。远程监控系统结构如图 2.1 所示。

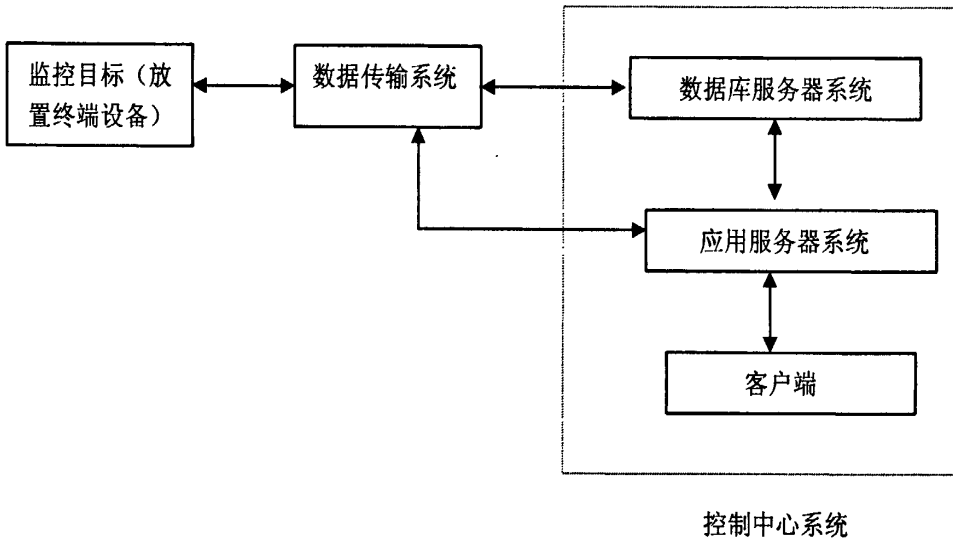


图 2.1 远程监控系统结构图

Fig.2.1 Remote monitoring system chart

1. 前端系统

前端设备，包括数据采集部分、前端处理部分。采集部分包括摄像头、GPS 模块等，完成监控现场的图像和监控目标的空间位置数据的采集。前端处理部分主要是完成对采集数据的筛选，重组。

2. 远程传输系统

传输系统是远程监控系统的关键组成部分，完成各种数据和监控指令的上传下达。远程传输系统可以采用有线和无线两种通信方式完成数据传输，实现数据和指令在监测现场和监测中心之间交互传递。

3. 控制中心系统

控制中心系统是整个系统的“大脑”，它用来存储来自各监控终端的图像、数据和终端的

状态,对图像数据进行拆包还原并且显示。对监控终端发送指令控制终端设备的运行,显示给用户所需要的监控目标位置信息和图像。

2.2 系统软硬件实施方案概述

本文所设计开发的奶源运输安全远程监控系统的工作模式是:奶罐车载有设计开发的监控终端,实时采集奶罐车取奶沿途的空间位置属性信息及停顿点图像信息,并将采集到的数据由 GPRS DTU 通过 GPRS 网络传输到 Internet 上,在监控部门的控制中心系统上将接收下来的数据储存在数据库服务器中并恢复显示,以电子地图的形式显示 GPS 经纬度数据,并恢复显示采集的图像,最终达到远程定位监控。

2.2.1 系统的硬件实施方案

监控终端硬件系统主要由微处理器 TMS320LF2407A DSP、摄像头模块 C328、GPS 模块、GPRS 模块和电源等组成。系统将在完成 DSP 最小系统的基础上,设计串行通信口的扩展,串口扩展完成了 1 扩 3 以及 SCI 和 RS232 的电平转换。同时还要设计 GPS、C328 和 GPRS 的接口电路。为提高实时性,软件采用 C 语言和汇编语言的混合编程,完成系统的初始化程序和串口扩展程序。设计 GPS 数据实时采集程序,利用高斯分布算法对 GPS 的数据进行分析处理。在此基础上,设计采集相应图像数据的程序,并根据 GPRS 带宽的情况将图像数据分包后转发给监控中心。

2.2.2 系统的软件实施方案

采用基于 MIDAS 技术的三层体系结构开发模式,三层分别是:客户端、应用服务器、数据库服务器。应用服务器负责将各项业务服务以及地图数据请求服务进行封装,以接口调用的方式,提供给客户端,进行业务请求处理。它可以被看作一个中间件,以接口的形式输出业务处理,供客户端调用,同时负责客户端与数据库之间的数据访问。数据库服务器负责终端数据、定位数据、图像数据和其他业务数据的存储。客户端则承担着向管理人员提供可视化远程监控界面及采集信息管理,实现的功能有:电子定位地图显示功能,最基本的地图工具功能;监控终端信息管理和奶站信息的管理功能;信息采集点与其对应线路的对比显示功能,通过 Mapx 常规技术实现奶罐车定位功能。

第三章 GPRS 技术简介

3.1 GPRS 技术原理

3.1.1 GPRS 概述^{[2][3][4]}

GPRS, 通用分组无线服务技术 (General Packet Radio Service)的简称, 它是 GSM 移动电话用户可用的一种移动数据业务。GPRS 可说是 GSM 的延续。GPRS 和以往连续在频道传输的方式不同, 是以封包式来传输, 其传输速率可提升至 56 甚至 114Kbps。

GPRS 经常被描述成“2.5G”, 也就是说这项技术位于第二代和第三代移动通讯技术之间。它通过利用 GSM 网络中未使用的 TDMA 信道, 提供中速的数据传递。GPRS 突破了 GSM 网只能提供电路交换的思维方式, 只通过增加相应的功能实体和对现有的基站系统进行部分改造来实现分组交换。

GPRS 网络是基于现有的 GSM 网络来实现的。GPRS 网络在原有的 GSM 网络的基础上增加了 SGSN (Serving GSN, GPRS 服务支持节点)、GGSN (网 Gateway GPRS Supporting Node, GPRS 网关支持节点) 等功能实体, GSN 是 GPRS 网络中最重要的网络节点。GSN 具有移动路由管理功能, 它可以连接各种类型的数据网络, 并可以连到 GPRS 寄存器。GSN 可以完成移动终端和各种数据网络之间的数据传送和格式转换。GSN 可以是一种类似于路由器的独立设备, 也可以与 GSM 中的 MSC 集成在一起。GSN 有两种类型: 一种为 SGSN (Serving GSN, 服务 GSN), 另一种为 GGSN (Gateway GSN, 网关 GSN), SGSN 的主要作用是记录移动终端的当前位置信息, 并且在移动终端和 GGSN 之间完成移动分组数据的发送和接收。GGSN 主要是起网关作用, 它可以和多种不同的数据网络连接, 如 ISDN、PSPDN 和 LAN 等。国外有些资料甚至将 GGSN 称为 GPRS 路由器。GGSN 可以把 GSM 网中的 GPRS 分组数据包进行协议转换, 从而可以把这些分组数据包传送到远端的 TCP / IP 或 X.25 网络。图 3.1 为 GPRS 网络的组成示意图。

GPRS 工作时, 通过路由管理来进行寻址和建立数据连接, 而 GPRS 的路由管理表现在以下三方面: 移动终端发送数据的路由建立; 移动终端接收数据的路由建立; 以及移动终端处于漫游时数据路由的建立。

对于第一种情况，当移动终端产生了一个 PDU 分组数据单元，这个 PDU 经过 SNDC 层处理，称为 SNDC 数据单元。然后经过 LLC 层处理为 LLC 帧通过空中接口送到 GSM 网络中移动终端所处的 SGSN。SGSN 把数据送到 GGSN。GGSN 把收到的消息进行解装处理，转换为可在公用数据网中传送的格式（如 PSPDN 的 PDU），最终送给公用数据网的用户。为了提高传输效率，并保证数据传输的安全，可以对空中接口上的数据做压缩和加密处理。

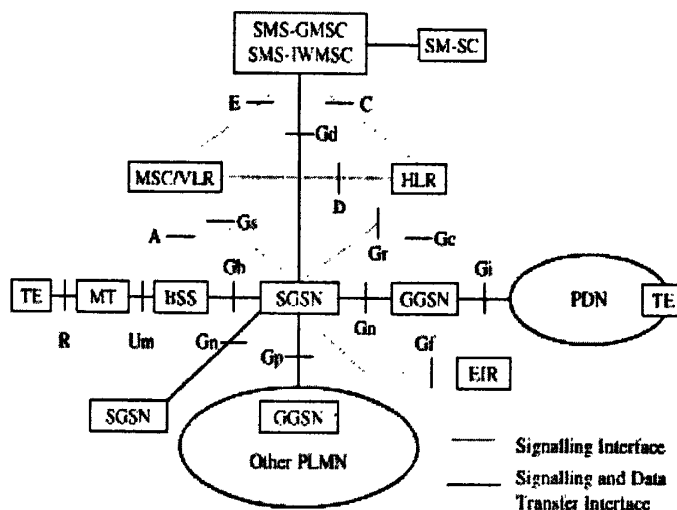


图 3.1 GPRS 网络组成
Fig.3.1 GPRS network structure

在第二种情况中，一个公用数据网用户传送数据到移动终端时，首先通过数据网的标准协议建立数据网和 GGSN 之间的路由。数据网用户发出的数据单元（如 PSPDN 中的 PDU），通过建立好的路由把数据单元 PDU 送给 GGSN。

而 GGSN 再把 PDU 送给移动终端所在的 SGSN 上，GGSN 把 PDU 封装成 SNDC 数据单元，再经过 LLC 层处理为 LLC 帧单元，最终通过空中接口送给移动终端。

第三种情况是一个数据网用户传送数据给一个正在漫游的移动用户。这种情况下的数据传送必须要经过归属地的 GGSN，然后送到用户 A 端。

3.1.2 GPRS 数据传输原理^[5]

GPRS 移动台像是一个内置了数据网络适配器的数据终端，有自己的标识和地址。GPRS 移动台要想在 GPRS 网络与外部数据网之间进行数据的发送和接收，必须先实现 GPRS 附着过程和 PDP 上下文激活过程。

1. GPRS 附着过程

移动台通过 GPRS 附着过程可以建立与 GPRS 网络的连接，具体过程如下：

(1) 移动台在系统分配的无线信道上向 SGSN 发送附着请求;

(2) SGSN 向移动台发送移动台标识请求, 移动台将自己的 IMSI 发送给 SGSN, SGSN 用取得的 IMSI 到 HLR 中请求认证并进行鉴权;

(3) 移动台认证通过后, SGSN 向 HLR 发送位置更新请求, HLR 响应该请求并向 SGSN 发送用户的相关信息, SGSN 收到此消息后向 HLR 发送确认消息。HLR 返回位置更新成功响应;

(4) SGSN 至此已接受了 GPRS 连接, 并向移动台发送 TLLI(临时逻辑链路标识), TLLI 用于标识移动台和 SGSN 之间的连接。SGSN 中的一个数据库完成分配给移动台的 TLLI 到移动台标识的映射。

2. PDP 上下文激活过程

移动台附着成功后, 就建立了移动台与 GPRS 网络之间的通信通道, 要想与外部数据网进行数据交换, 还必须发起 PDP 上下文激活过程。激活过程如下:

(1) 移动台向 SGSN 发送激活 PDP 上下文的请求(请求中带有接入点名称 APN 和 IP 地址参数);

(2) SGSN 进行移动台身份鉴别(IMSI)和设备检查(IMEI);

(3) 若身份鉴别和设备检查都通过了, SGSN 通过 DNS 得到 GGSN 的 IP 地址, SGSN 向其发送“建立 PDP 上下文请求”, GGSN 返回“建立 PDP 上下文响应”消息, 该消息中包括最后确认使用的 TID(隧道标识符)以及计费标识。GGSN 更新自己的表, 在表中完成 TID 和 SGSN IP 地址和与之相关的特定移动台的映射。

(4) SGSN 向移动台返回“PDP 上下文激活完成”消息, 同时, SGSN 也更新包含 TID 和 GGSN IP 地址的表, 并且用这些 TID 和 GGSN IP 地址为移动台建立隧道。至此, 移动台与 GPRS 网络之间完全建立了数据传输通路。

3. 移动台发起的数据传输过程

移动台附着成功并且成功激活了 PDP 上下文, 移动台和外部数据网络之间就可以进行数据传输了。从移动台发起的数据传输过程如图 3.2 所示:

具体实现步骤如下:

(1) 移动台有一张表, 保存着移动台到 TLLI 和网络层接入点 ID(NSAPI)的映射信息, 子网汇聚协议(SNDCP)把原始的 IP 分组加上包含 TLLI 和 NSAPI 的头信息, 通过移动台和 SGSN 之间的由移动台专用的 TLLI 将分组发送至 SGSN;

(2) SGSN 有一张表, 保存着 TLLI 和 NSAPI 到对应的 TID 和 GGSN IP 地址的映射信

息。SGSN用包含 TID 和 GGSN IP 地址的 GTP (GPRS 隧道协议) 取代包含 TLLI 和 NSAPI 的头信息;

(3) 数据分组以 SGSN 的 IP 地址为源地址, 以 GGSN 的 IP 地址为目的地址, 并且 TID 也是数据报分组的一部分;

(4) 在 GGSN 端, 去掉数据分组头信息, 就得到原始的 IP 或 X.25 分组。

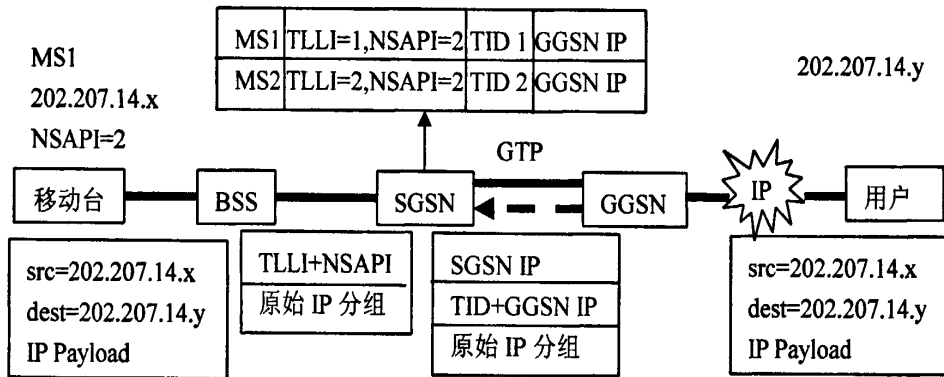


图 3.2 移动台发起的数据传输过程

Fig.3.2 Mobile originating data transmission process

4. 移动台终止的数据传输过程

在移动台终止的数据传输过程如图 3.3 所示, 具体实现步骤如下:

(1) 外部网络的分组数据到达 GGSN 后, GGSN 查表得到接收数据的移动台的特定 SGSN 地址和 TID;

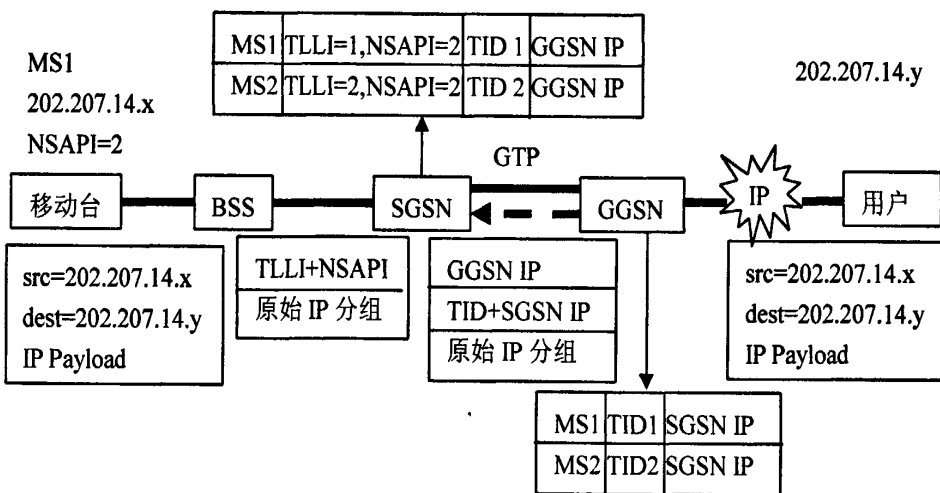


图 3.3 移动台终止的数据传输过程

Fig.3.2 Mobile terminating data transmission process

(2) GGSN 产生 IP 数据报, 数据报以 GGSN 的 IP 地址为源地址, 以 SGSN 的 IP 地址为目的地址, 并且还包括原始的 IP 分组和 TID 分组;

(3) SGSN 根据数据报中 TID 和 GGSN 查表得到相应的 TLLI 和 NSAPI, SGSN 就知道数据分组要送往移动台的位置;

(4) SGSN 取得原始 IP 分组, 加上包含 TLLI 和 NSAPI 的分组头信息, 并把它转发到移动台。

3.1.3 GPRS 提供的业务服务

GPRS 是在现有 GSM 网络上开通的一种新型的分组数据传输业务, 在有 GPRS 承载业务支持的标准化网络协议的基础上, GPRS 可以提供系列交互式业务服务^{[5][6]}:

1、点对点面向连接的数据业务。为两个用户或者多个用户之间发送多分组的业务, 该业务要求有建立连接、数据传送以及连接释放等工作程序。

2、单点对多点业务。根据某个业务请求者的要求, 把单一信息传送给多个用户。该业务又可以分为点对多点多信道广播业务、点对多点群呼业务和 IP 多点传播业务。

3、点对点无连接型网络业务。各个数据分组彼此互相独立, 用户之间的信息传输不需要端到端的呼叫建立程序, 分组的传送没有逻辑连接, 分组的交付没有确认保护, 是由 IP 协议支持的业务。

GPRS 除了提供点对点、点对多点的业务外, 还能支持用户终端业务、补充业务、GSM 短消息业务和各种 GPRS 电信业务。

3.2 GPS 技术原理

3.2.1 GPS 的概念

GPS 是英文 Global Positioning System, 全球定位系统的简称。GPS 是 20 世纪 70 年代由美国陆海空三军联合研制的新一代空间卫星导航定位系统。其主要目的是为陆、海、空三大领域提供实时、全天候和全球性的导航服务, 并用于情报收集、核爆监测和应急通讯等一些军事目的, 是美国独霸全球战略的重要组成。经过 20 余年的研究实验, 耗资 300 亿美元, 到 1994 年 3 月, 全球覆盖率达 98% 的 24 颗 GPS 卫星星座已布设完成。在机械领域 GPS 则有另外一种含义: 产品几何技术规范(Geometrical Product Specifications)-简称 GPS。

3.2.2 GPS 系统的组成^{[7][8]}

GPS 系统包括三大部分: 空间部分—GPS 卫星星座; 地面控制部分—地面监控系统; 用户设备部分—GPS 信号接收机

GPS 工作卫星及其星座由 21 颗工作卫星和 3 颗在轨备用卫星组成 GPS 卫星星座，记作 (21+3) GPS 星座。24 颗卫星均匀分布在 6 个轨道平面内，轨道倾角为 55 度，各个轨道平面之间相距 60 度，即轨道的升交点赤经各相差 60 度。每个轨道平面内各颗卫星之间的升交角距相差 90 度，一轨道平面上的卫星比西边相邻轨道平面上的相应卫星超前 30 度。

在两万公里高空的 GPS 卫星，当地球对恒星来说自转一周时，它们绕地球运行二周，即绕地球一周的时间为 12 恒星时。这样，对于地面观测者来说，每天将提前 4 分钟见到同一颗 GPS 卫星。位于地平线以上的卫星颗数随着时间和地点的不同而不同，最少可见到 4 颗，最多可见到 11 颗。在用 GPS 信号导航定位时，为了结算测站的三维坐标，必须观测 4 颗 GPS 卫星，称为定位星座。

地面监控系统：对于导航定位来说，GPS 卫星是一动态已知点。星的位置是依据卫星发射的星历—描述卫星运动及其轨道的参数算得的。每颗 GPS 卫星所播发的星历，是由地面监控系统提供的。卫星上的各种设备是否正常工作，以及卫星是否一直沿着预定轨道运行，都要由地面设备进行监测和控制。地面监控系统另一重要作用是保持各颗卫星处于同一时间标准—GPS 时间系统。这就需要地面站监测各颗卫星的时间，求出钟差。然后由地面注入站发给卫星，卫星再由导航电文发给用户设备。GPS 工作卫星的地面监控系统包括一个主控站、三个注入站和五个监测站。

GPS 信号接收机的任务是：能够捕获到按一定卫星高度截止角所选择的待测卫星的信号，并跟踪这些卫星的运行，对所接收到的 GPS 信号进行变换、放大和处理，以便测量出 GPS 信号从卫星到接收机天线的传播时间，解译出 GPS 卫星所发送的导航电文，实时地计算出测站的三维位置，位置，甚至三维速度和时间。

静态定位中，GPS 接收机在捕获和跟踪 GPS 卫星的过程中固定不变，接收机高精度地测量 GPS 信号的传播时间，利用 GPS 卫星在轨的已知位置，解算出接收机天线所在位置的三维坐标。而动态定位则是用 GPS 接收机测定一个运动物体的运行轨迹。GPS 信号接收机所位于的运动物体叫做载体（如航行中的船舰，空中的飞机，行走的车辆等）。载体上的 GPS 接收机天线在跟踪 GPS 卫星的过程中相对地球而运动，接收机用 GPS 信号实时地测得运动载体的状态参数（瞬间三维位置和三维速度）。

3.2.3 GPS 系统的定位原理^{[9]~[14]}

GPS 导航系统的基本原理是测量出已知位置的卫星到用户接收机之间的距离，然后综合多颗卫星的数据就可知道接收机的具体位置。GPS 卫星发射测距信号和导航电文，导

航电文中含有卫星的位置信息。用户用 GPS 接收机在某一时刻同时接收三颗以上的 GPS 卫星信号,测量出测站点(接收机广大线中心)P 至三颗以上 GPS 卫星的距离并解算出该时刻 GPS 卫星的空间坐标, 据此利用距离交会法解算出测站点 P 的位置。

设在 t_i 时刻在测站点 P 用 GPS 接收机同时测得 P 点至三颗 GPS 卫星 S1, S2, S3 的距离为 ρ_1, ρ_2, ρ_3 , 通过 GPS 电文解译出该时刻三颗 GPS 卫星的三维坐标分别为 (X^j, Y^j, Z^j) , $j=1, 2, 3$ 。用距离交会方法求解 P 点的三维坐标 (X, Y, Z) 的观测方程如式(2.1)所示。

$$\left. \begin{aligned} \rho_1^2 &= (X - X^1)^2 + (Y - Y^1)^2 + (Z - Z^1)^2 \\ \rho_2^2 &= (X - X^2)^2 + (Y - Y^2)^2 + (Z - Z^2)^2 \\ \rho_3^2 &= (X - X^3)^2 + (Y - Y^3)^2 + (Z - Z^3)^2 \end{aligned} \right\} \quad (2.1)$$

根据测距的原理, 其定位原理与方法主要有伪距定位法, 差分 GPS 定位以及载波相位测量定位。而对于待定点来说, 根据其运动状态可以将 GPS 定位分为静态定位和动态定位。静态定位指的是对于固定不动的待定点, 将 GPS 接收机安置于其上, 观测数分钟乃至更长的时间, 以确定该点的三维坐标, 又叫绝对定位。若以两台 GPS 接收机分别置于两个固定不动的待定点上, 则通过一定时间的观测, 可以确定两个待定点之间的相对位置, 又叫相对定位。动态定位则至少有一台接收机处于运动状态, 测定的是各观测时刻运动中的接收机的点位(绝对点位或相对点位)。

(一) 伪距定位法

伪距定位法是由 GPS 接收机在某一时刻测出的到四颗以上 GPS 卫星的伪距以及已知的卫星位置, 采用距离交会的方法求出接收机天线所在点的三维坐标。所测得的伪距就是由卫星发射的测距所得出的量测距离。由于卫星钟、接收机钟的误差以及无线电信号经过电离层和对流层中的延迟, 实际测出的距离 ρ' 与卫星到接收机的几何距离 ρ 有一定差值, 因此一般称量测出的距离为伪距。用 C/A 码进行测量的伪距为 C/A 码伪距, 用 P 码测量的伪距为 P 码伪距。伪距法定位方法的一次定位精度小高(P 码定位误差约为 10m, C/A 码定位误差为 20 至 30m, 但因其具有定位速度快、无多值性问题等优点, 仍然是 GPS 定位系统中进行导航的最基本方法。

GPS 卫星依据自己的时钟发出某一结构的测距码, 该测距码经过 τ 时间的传播后到达接收机。接收机在自己的时钟控制下产生一组结构完全相同的测距码——复制码, 并通过延时器产生延迟时间 τ' , 将这两组测距码进行相关处理, 若自相关系数 $R(\tau') \neq 1$, 则继续调整延迟时间直至自相关系数 $R(\tau')=1$ 为止。使接收机所产生的复制码与接收到的 GPS 卫星测距码完全对齐, 那么其延迟时间 τ' 即为 GPS 卫星信号从卫星传播到接收机所用的时间 τ' 。GPS

卫星信号的传播是一种无线电信号的传播，其速度等于光速 c ，卫星到接收机的距离即为 τ' 和 c 的乘积。

(二) 差分定位法

差分 GPS(DGPS)是显著地改善 GPS 精度和完善性的一种技术，它要求有高质量的 GPS “基准接收机”放在已知的经过测绘的位置上。将一台 GPS 接收机放在基准站并进行观测。根据基准站已知的精密坐标，计算出基准站到卫星的距离改正数，并由基准站实时地将这一改正数发送出去。用户接收机在进行 GPS 观测的同时，也接收到基准站的改正数，并对其定位结果进行改正，从而提高定位精度。采用差分 GPS 可以完全消除多台接收机公有的误差，例如：卫星钟误差，星历误差等；也可消除大部分传播延迟误差，如电离层误差，对流层误差等。

差分 GPS 技术及其应用有很多种，主要的技术有以下几种：

1. 局域差分 GPS (LADGPS)

使用单一基准站，对每颗卫星的码——相位测量产生的标量校正值。

2. 广域差分 GPS (WADGPS) 广域差分 GPS 对 GPS 观测量的误差源加以区分，并单独对每一种误差源分别加以“模型化”，然后将计算出的每一个误差源的数值，通过数据链传输给用户，以对用户 GPS 定位的误差加以改正，达到削弱这些误差源，改善用户 GPS 定位精度的目的。它有如下特点：

- 1) WADGPS：用户的定位精度对空间距离的敏感程度比局部区域 GPS 差分系统低的多。
- 2) 大区域内建立 WADGPS 网，需要的监测站数量少，投资小，具有更大的经济效益。
- 3) ADGPS 系统是一个定位精度均匀的系统，覆盖范围内任意地区定位精度相当，定位精度较高。
- 4) GPS 覆盖区域广，例如沙漠、森林、海洋等。
- 5) ADGPS 使用的硬件设备和通信工具昂贵，软件技术复杂，运行和维护费用较高。

3. 载波相位差分 GPS

载波相位差分又称 RTK(Real Time Kinematic)技术，是实时处理两个测站载波相位观测量的差分方法。它一般分为两类：修正法和差分法。修正法是将基准站的载波位修正值发送给用户，改正用户接收到的载波相位，再求解坐标。差分法是将基准站采集的载波相位发送给用户，进行求差解算坐标。修正法属于准 RTK，而差分法属于真正 RTK。

RTK 技术可应用于海上精密定位，地形测图和地籍测绘。

(三) 载波相位法

伪距测量由于测距码的码元长度较大,对于一些高精度应用来讲其测距精度还显得过低,而如果把载波作为量测信号,由于载波的波长短, $\lambda_{L_1} = 19cm, \lambda_{L_2} = 24cm$ 所以可以达到很高的精度。

载波相位测量的观测量为 GPS 接收机所接收的卫星载波信号与接收机本振参考信号的相位差。以 $\phi_k^j(t_k)$ 表示 k 接收机在接收机钟而时刻 t_k 时所接收到的 j 卫星载波号的相位值, $\phi_k(t_k)$ 表示 k 接收机在钟而时刻 t_k 时所产生的本地参考信号的相位值, 则 k 接收机在接收机钟而时刻 t_k 时观测 j 卫星所取得的相位观测量可写为

$$\phi_k^j(t_k) = \phi_k^j(t_k) - \phi_k(t_k) \quad (2.2)$$

通常的相位或相位差测量只是测出一周以内的相位值,实际测量中,如果对整周进行计数,则自某一初始取样时刻(t_0)以后就可以取得连续的相位观测量。如果在跟踪卫星过程中,由于某种原因,如卫星信号被障碍物挡住而暂时中断,受无线电信号干扰造成失锁,这样,记数器无法连续记数,因此当信号从新被跟踪后,整周记数就小正确,但是不到一个整周的相位观测值仍是正确的。这种现象称为周跳。周跳的出现和处理是载波相位测量的重要问题,需要探测和修复。

载波相位测量的精度最高,是由载波跟踪环得到。由于载波环的测量精度比码环高两个数量级,可以达到毫米级,因此如果能获得初始整周模糊度,就可以得到厘米级的高精度测量。

3.2.4 GPS 系统的特点^{[14][15]}

GPS 全球定位系统具有精度高、全天候、多功能、操作方便等特点:

1. 定位精度高,观测时间短

GPS 能连续地为用户提供三维位置、三维速度和精确时间信息;利用 GPS 进行导航定位,1 秒即可完成一次定位,对高动态用户尤其重要。

2、被动式全天候导航定位

由 GPS 导航定位时,用户设备只需接收机 GPS 信号就可进行导航定位,而不需用户发射任何信号。这种方式隐蔽性好,而且可容纳无数多用户。

3、全球覆盖连续导航定位

由于 GPS 有 24 颗卫星,且分布合理,轨道高达 20200km,所以在地球上和近地空间上任何一点,均可连续同频地观测 4 颗以上卫星、实现全球、全天候连续导航定位。

第四章 MapX 控件技术介绍

4.1 MapX 概述

MapX 是一个基于 ActiveX(OCX)技术的可编程控件。它使用与 MapInfo Professional 一致的地图数据格式，并实现了大多数 MapInfo Professional 的功能。

MapX 作为应用于二次开发的 OCX 控件，封装了 Mapinfo 地图操作大部分的功能，MapX 与 Mapinfo 的数据格式是统一的。MapX 的内部提供了四十多个面向 Microsoft 基础类的 OLE 控件，这使得 MapX 可以被快速集成到使用 Visual Basic、PowerBuilder、Delphi、Visual C++ 或其他面向对象的语言的客户端应用程序以及使用 Lotus Script 的 Lotus Notes 中。

采用嵌入 MapX 控件的方式，方便快捷地开发 GIS 应用程序。Mapinfo 用工作空间 (WorksPace)、图层(layer)、地物类等概念组织数据，MapX 则利用自动化对象将这些概念封装起来，相应地提供了 Dataset 对象、Layer:对象、Feature 对象等，并且考虑到数据组织中大量出现集合类型的引用的情况，对这些对象的集合也进行了封装。

与 MapInfo professional 相比，MapX 可以实现 95%以上的 MapInfo Professional 的功能，但在图形编辑编程上的开发周期可能是 MapInfo Professional 的 3 倍以上，但对于查询和空间分析以及其它简单的图形编辑功能、地图注记功能 MapX 在使用时是很方便的，具有系统开销小、开发周期短、方便快捷等优点，很适合于各种中小型 GIS，MIS 软件的开发。

4.2 MapX 的数据文件和组件模型结构

4.2.1 MapX 的数据文件^[16]

MapX 遵循与 Mapinfo 相同的数据格式。下面简要说明与 Mapinfo 有关的数据文件：

- DAT 文件：以 Mapinfo 格式保存的表格数据文件；
- DBF 文件：DBASE/FoxBASE 格式的表格数据文件；
- ID 文件：MapInfo 图形对象文件(*.MAP)的索引文件；
- IND 文件：Mapinfo 表格文件(*.DAT)的索引文件；
- MAP 文件：包含描述地理对象的地理数据；

- MID 文件：用于表格数据的 MapInfo 转入转出格式；
- MIF 文件：用于图形对象的 MapInfo 转入转出格式；
- TAB 文件：MapInfo 标的主文件，与*.DAT(或者“*.DBF、*.XLS、*.WKS、*.TXT 等文件)、*.MAP、*.ID、*.IND 等文件关联。
- TXT 文件：用于存储 ASCII 格式的表格数据；
- WOR 文件：MapInfo 的工作空间文件，用于保存有关在某次 MapInfo 交互中所使用的表和窗口的信息。

4.2.2 MapX 的组件模型结构

MapX 组件的基本组成单元是 Object(单个对象)和 Collection(集合)。其中集合包括对象，是多个对象的组合。每种对象和集合负责处理地图某一方面的功能。位于顶层的是对象本身，其它均由 Map 对象继承。Layer、Dataset、Almotation 是 Map 对象下面的三个重要的分支。其中，Layer 主要用于操作地图的图层，Dataset 用于访问空间数据表，Almotation 用于在地图上增加文本或者符号。

4.3 MapX 的基本概念介绍

※ 地图图层^[17]

在 MapX 中，图层是电子地图的构筑块，计算机地图实际是多个图层的集合。我们可以把计算机地图看成是由层层叠加的透明层组成的，而该透明层就被称为图层，每个图层包含了整个地图的一个不同方面。

图层来自于含有图形对象的数据库表，每个含有图形对象的数据库表都可显示为一个图层。例如有 3 个图层，一个图层包含世界各国边界，一个图层包含世界各个主要城市，另一个图层由标注文本组成，这样把 3 个图层叠加在一起就形成了一幅完整的世界主要城市地图。如图所示的点线面涂层的叠加。

※ 地图对象

如果说图层是 MapX 地图的构筑块，那么地图对象就是图层的基本元素。假如对一幅传统的地图仔细分析，不难发现地图均由点、线、面组成，因此大致可以把图层对象抽象为点、线、面 3 种类型。

点是由平面位置决定的一个空间地理实体，也用来表示地理线的起点、终点和交点；线具有一定走向和长度，表示线状的物或点之间的地理联系；面具有其确定的范围和形态，表

示空间连续分布的地理景观或作用范围。点、线、面结合组成网络；线、面结合成为地带；面、点结合组成地域类型。

点对象：由一对平面坐标表示，至少具有一个属性。在一幅城市地图中，一个建筑物，如一座桥梁、一座建筑、一个公共汽车车站等，均可称为点对象。

线对象：具有相同属性点的轨迹，在图层中由坐标集合的数据链表示。城市中的道路中心线、铁路、道路边界等均可称为线对象。

区域对象：区域对象可以是各种形状，区域对象是具有相同属性点的轨迹：海域、湖泊、行政区划、城市中的绿地等，均可作为区域对象处理。

文本对象：描述地图或其他对象的文本，如地图标题或标注等。

4.4 MapX 的特点^[17]

- 可深入的地图

通过简单的点击方式可浏览与地图对象相连的数据信息；

- 数据绑定

地图可通过嵌入 OCX 的容器与数据库相连，并提供了几种不同的数据绑定方式包括 ZlpCode — level 地理编码法；

- 注解

可提供方向、加亮显示特殊数据，还可加入文本、符号、表格使地图信息更加丰富直观；

- 图层

显示和控制图层的缩放、使用或创建无缝地图、还可支持一些特殊的应用，比如用于实时跟踪的活动图层和可绘制特殊图形的用户自定义图层；

- 栅格图象

采用栅格图象作为地图的基础图层可使其它图层有一个更细致的背景；

- 自动标记

自动在地图上加入标记，同时标记属性和显示；

- 选择

可在地图上拖动鼠标以在圆、矩形或特定的点上选择一个或多个对象或记录以供分析；

- 对象库

可以使用 FeatureFactory 对象，创建、联接或删除点、线、区域图形对象；

- 工具

可以使用标准工具或根据需要自己创建的自定义工具，用户可通过点击或拖拽对地图直接进行操作；

- 地图编辑

允许用户添加、修改、删除地图上的文本、编辑区域、点等特殊对象；

- 投影与坐标系

MapX 允许用户调整地图的显示，用本地坐标系处理 X—Y 坐标数据。

第五章 监控终端的硬件设计与实现^[18]

5.1 监控终端的组成

监控系统的监控终端单元由 TMS320LF2407A DSP 芯片、GPS 模块电路、C328 模块电路、GPRS DTU 模块电路、SRAM 数据存储器组成。系统原理框图如图 5.1 所示。

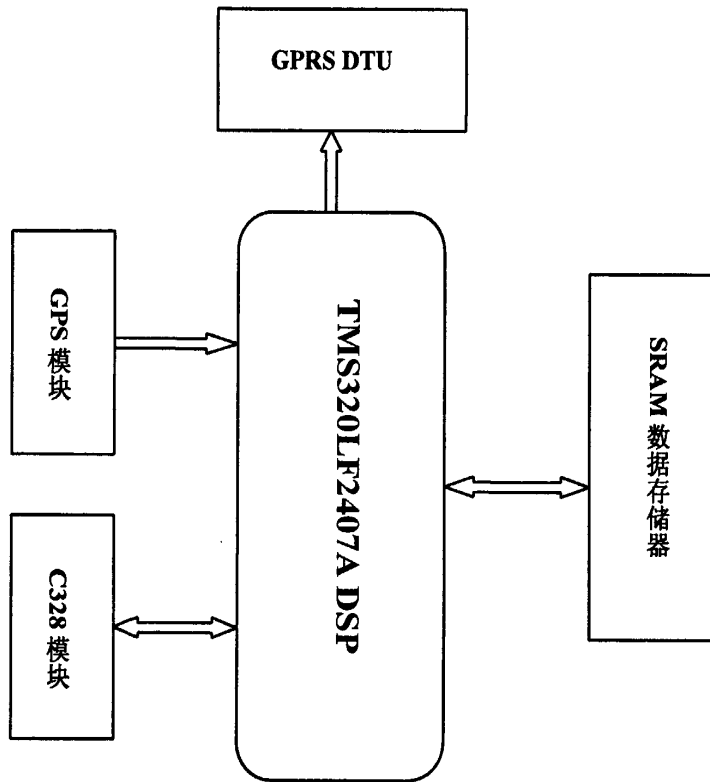


图 5.1 监控终端系统框图

Fig.5.1 System diagram of the monitoring end

TMS320LF2407A 作为整个系统的核心，主要用来完成地理信息和图像数据的采集、存储、处理和转发在内的所有软件功能。GPS 完成实时地理信息的跟踪和采集。C328 完成图像数据的采集和压缩。GPRS DTU 负责建立透明传输信道。SRAM 用来存储、交换图像数据。

5.2 监控终端硬件资源介绍

5.2.1 微处理器 TMS320LF2407A DSP

TMS320LF2407A 是由美国 TI 公司推出, 针对控制应用而设计的、高性能的 16 位定点 DSP 芯片。TMS320LF2407A DSP 采用改进的哈佛结构, 支持分离的程序总线和数据总线。该芯片将实时信号处理能力和控制器外设功能集于一体, 为控制应用提供了一个理想的平台。具有灵活的指令集, 同时可分离的程序总线和数据总线使取指令、执行指令、数据传送和外设控制可以并行进行, 极大地提高了工作速度和处理能力。

5.2.2 GPRS 模块

本系统采用驿唐科技公司的 MD—600 GPRS 模块, 它提供单独的 RS232 配置/本地调试串口及标准 RS232/485 数据接口, 与用户设备即插即用; +5V—36V 宽范围电压输入, 多种电源接口, 适用于多种场合; 包含 TCP/IP 协议栈及 ETCPTM 协议, 保证传输数据不丢失; 硬件设计符合电工电子产品低温 GB/T2423.1、高温 GB/T2423.2 的要求, 适用于在气候条件恶劣的地区及户外使用。

5.2.3 C328 摄像头模块

图像数据的采集主要是由摄像头模块完成。本系统采用康大科技有限公司生产的 C328 摄像头模块, 它是一款高分辨率、外型小巧、低功耗的串行总线系统; 内设程序存储器, 可以为外部控制器提供便捷的命令接口; 采用 OV7640/8 VGA 图像传感器。

5.2.4 GPS OEM SL-SDT11 板

本设计的 GPS 模块采用了嵌入式 GPS OEM SL-SDT11 板。以实现接受卫星信号并将实时定位信息输出给 DSP, 从而经过 DSP 处理后, 转发给监控中心决策。SL-SDT11 具有以下特点:

- 1.源于由 u-blox 与 Atmel 共同开发的 GPS 芯片组 ATR0601+ATR0625。
- 2.具有 16 路卫星接收通道, 接收灵敏度达-158 dBm, 精度误差 2.5 米。
- 3.支持 DGPS、WAAS、EGNOS 和 MSAS。
- 4.提供 USB 和 UART 各一个, 通讯波特率 9600。
- 5.工作电压 3.3V, 工作电流 39mA。

- 6.内置 RTC, 可由后备电池支持。
- 7.工业级产品, -40--+85 度。
- 8.尺寸 24×20×2.8 mm。
- 9.天线的中心频率为: 1575.42MHZ, 电压为 3-5V。

5.3 监控终端主要模块的设计与实现介绍

基于 DSP 的监控终端系统主要实现 GPS 定位数据的实时采集、存储、分析、处理和转发, 并根据相应需求算法对图像监控数据进行采集、存储和传输的功能。下面简单介绍硬件系统的主要模块功能的实现。

5.3.1 GPS 模块功能的实现

系统的 GPS 采集模块会不断采集卫星定位数据, 并按 NMEA-0183 数据格式输出。DSP 对 GPS 定位数据实时接收并进行相应数据的提取、存储、处理和转发。

1. GPS 输出数据格式

GPS 采集模块输出的数据格式为 NMEA-0183, 它是 RTCM (海事无线电委员会) 的标准, 由美国国家海洋电子协会 NMEA (The National Marine Electronics Association) 制定的一套通讯协议。其信息是 ASCII 码字符, 定义如表 5.1 所示:

表 5.1 NMEA-0183 定义表
Tab.5.2 Definition table of NMEA-0183

信息字段	描述
\$	信息起始标志。
ID	2Bytes 的信息源起始标志。接收 GPS= "GP", GLONASS=" GL", 几个卫星系统= "GN"。
MSG	3Bytes 的信息标志, 数据的内容、数量、序号
,	信息字段的分隔符号
D1, ..., Dn	数据字段
*	校验和标志
CS	2Bytes 校验和
<CR><LF>	结束和换行标志

NMEA-0183输出信息包括以下语句, 如表5.2所示:

表 5.2 NMEA-0183 输出信息表
Tab.5.2 NMEA-0183 Output Messages

NMEA语句	语句描述
GGA (Default)	Global positioning system fixed data
GLL (Default)	Geographic position - latitude/longitude
GSA (Default)	GNSS DOP and active satellites
GSV (Default)	GNSS satellites in view
RMC (Default)	Recommended minimum specific GNSS data
VTG (Default)	Course over ground and ground speed

以上 NMEA-0183 的语句中，设计选用了 RMC 语句，通过 DSP 软件处理提取所需要的定位和速度等信息。下面就 RMC 语句进行举例说明。

RMC—Recommended Minimum Specific GNSS Data

\$GPRMC,161229.487,A,3723.2475,N,12158.3416,W,0.13,309.62,120598, *10<CR> <LF>

表 5.3 RMC 数据格式
Tab.5.3 RMC Data Format

name	example	Unit	Description
Message ID	\$GPRMC		RMC protocol header
UTC Position	161229.487		hhmmss.sss
Status	A		A=data valid or V=data not valid
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		dddmm.mmmm
E/W Indicator	W		E=east or W=west
Speed Over Ground	0.13	knots	
Course Over Ground	309.62	degrees	True
Date	120598		ddmmyy
Magnetic Variation		degrees	E=east or W=west (Not shown)
Checksum	*10		
<CR> <LF>			End of message termination

2. GPS 数据接收

GPS 数据接收流程图如图 5.2 所示，主要步骤包括：1、串口切换到子串口 1。2、筛选 \$GPRMC 的 GPS 输出信息。3、数据接收并以顺序表存储结构做缓冲区，等待处理。4、对缓冲区数据进行处理，提取出定位信息和速度信息，并以动态申请存储空间的链表结构存储。5、以此循环 100 次结束。

由于 \$GPRMC 的输出语句的字段是已知的 (\$GPRMC 为 70Bytes)，而且数据结构是线性表结构，所以其存储结构采用向量的一维数组。GPS 数据经过提取后，为了节约内存空间，

算法中采用链式存储结构。另外，在提取算法中，仅提取 GPS 的经度、纬度和速度信息。

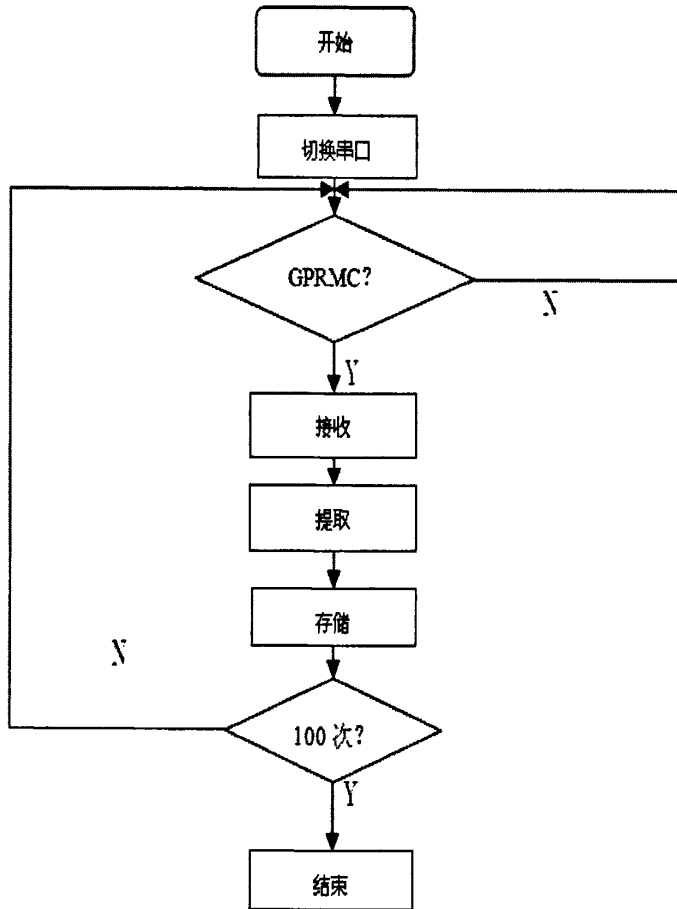


图 5.2 GPS 数据接收流程图

Fig.5.2 Flow graph of GPS receiving

3. GPS 数据处理算法设计

通过 GPS 数据采集模块接收到数据较多，为了比较准确判断出速度是否为零，设计上采用了随机样本分析的方法，利用高斯分布来检测速度为零。速度不为零的情况下，将提取后的 GPS 数据转发监控中心。如果速度为零，先将此时的 GPS 数据利用堆栈保存，之后调用图像采集模块，图像数据处理结束后，再将堆栈中的 GPS 数据发送至监控中心。软件流程图如图 5.3 所示。

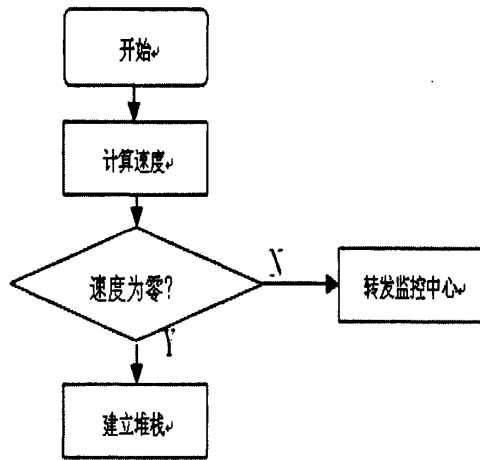


图 5.3 计算速度流程图

Fig.5.3 Calculating speed flow

5.3.2 C328 模块功能的实现

C328 模块输出的数据格式为 JPEG^[27]。JPEG 文件的格式是以段的形式来存储的，段的大小由段头来标识。JPEG 文件的每个段的标识的构成是：第一个字节是十六进制 0xFF，第二个字节对于不同的段，这个值是不同的。紧接着的两个字节存放的是这个段的长度。长度的表示方法是按照高位在前，低位在后的。DSP 的串口切换到图像采集模块后，连续向 C328 发送同步命令 SYNC，直到 C328 返回应答确认信号 ACK 实现同步。

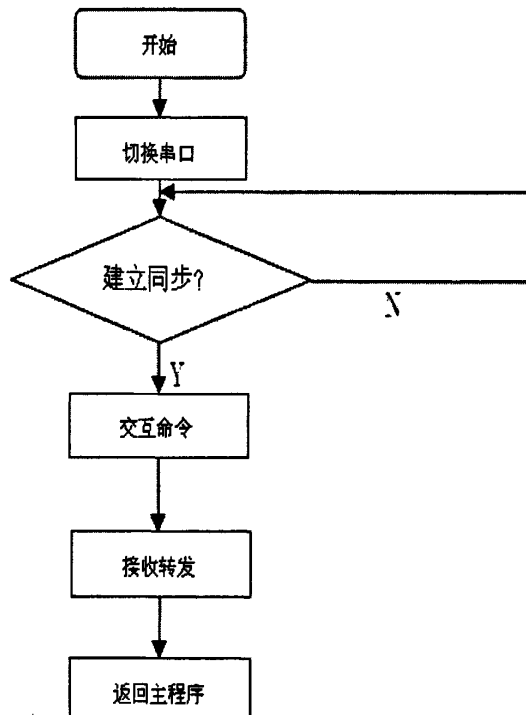


图 5.4 接收图像数据流程图

Fig.5.4 Receiving image data flow

接收 C328 采集到的图像之前，需要 DSP 和 C328 进行一系列的交互命令后才能接收图像数据。交互命令有 C328 模块图片模式初始化、设置包大小、拍摄、采图等几个步骤。接收图像数据流程图如图 5.4 所示。

相关算法的思想是，DSP 按顺序发送相关命令，发送一条命令之后，等待 C328 的确认，为了保证收到确认，设计连续发送 3 次命令。待确认后，再发送下一条命令。

图像数据的接收和转发，首先要甄别一帧图像数据的大小，然后据此可以判断需要收发多少数据包。考虑到图像数据比较大，不宜在监控终端保留，所以接收一包（512Bytes）转发一包。

5.3.3 GPRS 模块功能的实现

在传输数据之前，首先要通过 GPRS 模块建立与监控中心的端到端的信道。信道的建立和传输的流程如下图 5.5 所示。

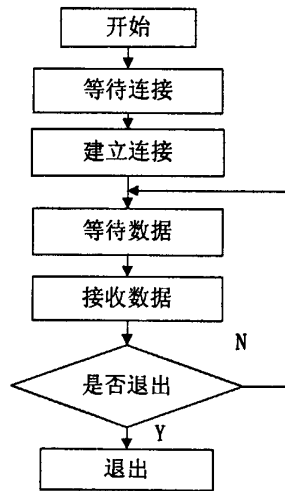


图 5.5 数据传输流程图

Fig.5.5 Flow Chart of Data Transmission

端到端的信道建立后，监控终端和监控中心就可以实现数据的通信。需要传送的数据，在发送前根据 GPRS 无线通信协议将所发送的数据打包成如表 5.4 的数据格式，再封装成 IP 包传输，封装过程如图 5.6 所示。

表 5.4 GPRS 模块通信数据格式

Tab.5.4 Format of GPRS Module Communication Data

名称	大小	意义
ID	4 bytes	表示一个消息包的开始, 为四个字节的 0x7e
IMEI	16bytes	是 GPRS 模块的唯一标识, 是最后一个字节为结束符 “\0” 的, 实际长度为 15 字节的字符串
Name	16bytes	是 GPRS 模块的别名, 用来便于人工识别 GPRS 模块, 是最后一个字节为结束符 “\0” 的, 实际最大长度为 15 字节的字符串
Msg Type	1bytes	消息类型
Reserved	1bytes	保留字段
Msg Len	2bytes	指明了 Msg Body 的实际长度, 两个字节长, 以网络字节序存放
Msg Body	待定	消息体, 长度由 Msg Len 字段指明, 不同的消息类型有不同的消息体。最大消息体长度为 1492byte。

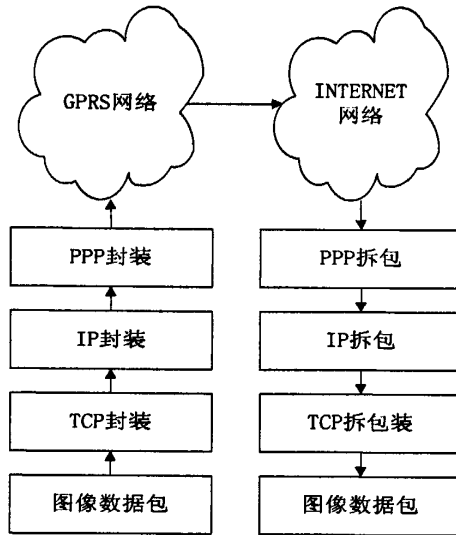


图 5.6 数据包封装过程

Fig.5.6 Encapsulating Process of Datagram

第六章 奶源运输安全远程监控系统监控中心的设计与实现

6.1 监控中心的模式选择

C/S (Client/Server, 客户机/服务器) 模式是一种两层结构的系统,第一层在客户端上安装了客户端应用程序,第二层在服务器上安装服务器管理程序。在 C/S 模式的工作过程中,客户机程序发出请求,服务器程序接收并且处理客户机程序提出的请求,然后返回结果。

B/S 模式,即浏览器/服务器模式,是一种从传统的两层 C/S 模式发展起来的新的网络结构模式,其本质是三层结构的 C/S 模式。在用户的计算机上安装浏览器软件,在服务器上存放数据并且安装服务应用程序,服务器有 WWW 服务器和文件服务器等。用户通过浏览器访问服务器,进行信息浏览、文件传输和电子邮件等服务。

C/S 模式有以下特点:

- 1.C/S 模式将应用与服务分离,系统具有稳定性和灵活性;
- 2.C/S 模式配备的是点对点的结构模式,适用于局域网,有可靠的安全性;
- 3.由于客户端实现与服务器端的直接连接,没有中间环节,因此响应速度快;
- 4.在 C/S 模式中,作为客户机的计算机都要安装客户机程序,一旦软件系统升级,每台客户机都要安装客户机程序,系统升级和维护较为复杂。

与 B/S (Browser/Server, 浏览器/服务器) 模式相比, C/S 模式的应用系统最大的好处是不依赖企业外网环境,即无论企业是否能够上网,都不影响应用。基于 C/S 模式的这个优点,我们选择 C/S 模式开发控制中心软件。

6.2 监控中心的开发方案

6.2.1 软件实现功能详解

在选择好开发模式后,接下来就要对每个业务环节进行详细的分解。分析系统的业务需求,要实现的主要功能如下。

1. 登录操作

系统可以设置的服务器 IP 地址、用户名和密码登录到应用服务器。

2. 数据管理

系统应能够实现对基础数据的管理，基础数据包括奶站信息、监控终端信息、管理员信息、终端 GPS 采集数据、图像数据、终端状态信息等。

3. 奶站线路录入功能

系统建立初期，需要将所有奶站到企业的线路，进行 GPS 数据采集并导入数据库，存储后数据后可在电子地图上生成线路图。

4. 奶站信息管理

该模块应具有：1、奶站信息数据（包括奶站名称、奶站经度纬度、距离企业路程等）增加、删除处理功能。

5. 管理人员信息管理

系统应实现对管理人员信息的增加、修改和删除功能。

6. 监控终端信息管理

系统应实现对监控终端信息（包括名称、所属奶站以及所绑定的奶罐车编码等属性）的增加、修改和删除功能。

7. 远程定位

远程定位作为整个系统的核心业务，也是远程监控功能实现的主体模块，该模块的主要内容是：1、能够以电子地图的形式显示 GPS 经纬度数据，实现远程定位监控；2、选择定位对象，可以选择单一定位对象，也可同时选择多个对象；3、定位开启、关闭，系统设置定位触发模块，以便于管理人员自动启动、结束定时跟踪。

8. 图像显示及管理

系统能够恢复显示监控终端在奶罐车停顿时现场采集的图像；

6.2.2 开发方案的选择^{[19][20][21]}

目前的开发工具所提供的技术功能和开发模式日益丰富，所以针对某一个功能，可供我们使用的实现方式有很多种。但针对本课题的设计要求，要注意以下几点：(1)可维护性：系统的设备平台、应用软件都应提供方便、灵活的维护手段，方便管理人员的维护和管理；(2)安全性：要保证系统安全、可靠地运行。(3)可扩展性：系统的设计和建设要充分考虑网络扩展需要、系统二次开发需要以及支持未来可能出现的新业务的需要。系统采用开放的可扩充模块结构，保证以后可以方便地升级和不断增加新业务。(4)经济性：在满足业务需求的前提下，控制网络流量、保证数据传输的速度。

基于以上要求,我们决定将所有的业务处理放在局端的应用服务器中,将业务处理封装成函数,以接口的形式提供给客户端调用。因此,在开发技术上选择使用基于 COM 组件技术的 Midas 结构的自动化服务器模式作为应用服务器的实现。

Midas 是 Multi-tier Distributed Application Services Suite(多层分布式应用程序服务包)的缩写,这也诠释了 Midas 技术的实质。多层计算(Multi-tiered Computing)是业界对此类技术通用的术语,Multi-tier 是 Borland 公司采用的技术术语。Borland 典型的三层结构如下:第一层是数据库服务器,第二层是应用服务器,第三层是瘦客户机。分布式结构允许在客户端的应用中使用所有标准的 Delphi 组件,包括数据库工具,但是客户端无须装载 BDE、ODBC 或者任何数据库类库(例如 Oracle SQL*NET、Sybase CT-Lib 等),只需要一套服务器端的数据库工具,每一个客户端就可以使用它。同时分布式结构也是缩减网络通信量的一种方法。从服务器端下载数据后,在客户端操作数据而无须初始化任何网络事务,除非需要更改服务器端。这意味着在不启动网络事务的情况下,可以编辑、插入、删除多个记录。更改服务器数据时,可以在预先选定的时间段内把多个数据包传送到网络上。

三层分布式系统比两层的多了一个应用服务器层。其优点是可以重复某多层分布式系统比两层的多了一个应用服务器层。其优点是可以重复某些代码,使整个系统的开发和维护成本大为降低。数据模块 DM(DataModule, DM)包含了与用户会话的数据感知组件以及和应用服务器连接的 RemoteServer 组件。远程数据模块 RDM 提供的 IAppServer 接口是一个容器,可以容纳访问远程数据库服务器的数据集控件和 DataSetProvider 等控件。客户端应用程序提供友好的人机交互界面,通过标准数据控制组件与用户交互。它是通过应用程序服务器提供的 IAppServer 接口获得,更新数据,并不直接存储或维护数据。RemoteServer 控件通过 Socket 协议与应用服务器建立连接;ClientDataSet 控件负责产生远程数据模块中数据集组件在客户端数据模块的映像;数据感知组件用于向用户显示数据并感知用户存储或变更数据的请求。应用服务器是整个系统的核心部分,主要由远程数据模块(RemoteDataModule,RDM)组成。其主要功能是让客户端能够通过应用服务器存取远程数据。DataSet-Provider 控件输出 IAppServer 接口,并通过 DataSet 属性与数据集组件(DataAccess)相连,数据集组件通过 BDE 引擎接口访问远程数据库,这样客户端通过 IAppServer 接口就可以存取远程数据。数据库服务器是独立的,用于对数据进行维护和管理。即使在整个程序已经开发完成的情况下更换数据库也不需要重新编写客户端或是应用服务器,只需要对应用服务器稍做修改即可。Delphi 支持多种数据库,如 Oracle、SQLserver 等。Delphi 还支持在一个数据库应用程序中使用多个数据库系统。

多层分布式结构是顺应网络发展需要的产物，它分离了客户界面和企业逻辑，克服了以往两层结构难于维护、客户端臃肿等弊端，还具有良好的扩展性。Delphi 提供的 MIDAS 技术是多层分布式系统中的佼佼者，功能强大，并提供了丰富的连接方式。随着 MIDAS 技术的进一步完善，Delphi 将成为开发多层分布式应用系统的首选平台。

通过对设计要求的解析，本文中监控中心所采用的开发方案是：

采用基于 MIDAS 技术的三层体系结构开发模式，三层分别是：客户端、应用服务器、数据库服务器。应用服务器负责将各项业务服务以及地图数据请求服务进行封装，以接口调用的方式，提供给客户端，进行业务请求处理。它可以被看作一个中间件，以接口的形式输出

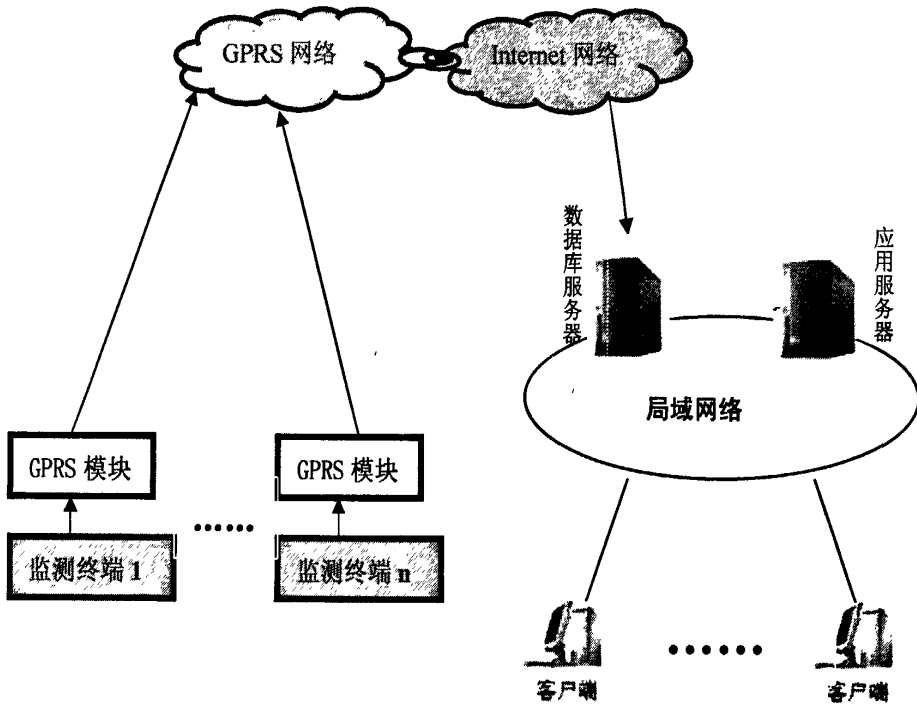


图 6.1 系统控制中心软件结构实况图

Fig. 6.1 System control center software's structure figure

业务处理，供客户端调用，同时负责客户端与数据库之间的数据访问。数据库服务器负责人员数据、定位数据、地理信息数据和其他业务数据的存储。客户端则承担着向管理人员提供可视化远程监控界面及采集信息管理功能，包括：电子定位地图显示功能，最基本的地图工具功能；监控终端信息管理和线路信息的管理功能；信息采集点与其对应线路的对比显示功能，通过 Mapx 常规技术实现奶罐车定位功能。

系统控制中心软件结构实况图如图 6.1 所示。

基于以上因素，我们在数据库选择上，需要从性能及使用规模上做一个综合性的分析，进行全面的比较。结合项目的实施规模，我们最终选取 SQL Server 数据库，本系统设计中采

用了 SQL Server 2000 版本。

SQL Server 2000 全面扩展了 SQL Server 7.0 的性能及可靠性和易用性,使它成为一个杰出的数据库平台,可用于大型联机事务处理、数据仓库、以及电子商务等。SQL Server 2000 对 XML 和 HTTP 提供充分的支持,提供分割装载和确保时间的特性,提供先进的管理和协调机制,可以自动管理线程任务,降低总体开销。SQL Server 2000 有四个系统数据库: Master、Model、Msdb、Tempdb。这些系统数据库的文件存储在 Microsoft SQL Server 默认安装目录的 MSSQL 子目录的 Data 文件夹中。

在 SQL Server 中数据库是由数据库文件和事务日志文件组成的。一个数据库至少应包含一个数据库文件和一个事务日志文件。

(1) 数据库文件

数据库文件是存放数据库数据和数据库对象的文件。一个数据库可以有一个或多个数据库文件,一个数据库文件只属于一个数据库。当有多个数据库文件时,有一个文件被定义为主数据库文件,扩展名为.mdf,它用来存储数据库的启动信息,主文件还用于存储部分或全部数据,每个数据库只能有一个主数据库文件。其它数据库文件被称为次数据库文件,扩展名为.ndf,用来存储不能至于主数据库文件中的所有数据。如果主数据库文件足够大,能够容纳数据库中的所有数据,则该数据库不需要次数据库文件。但有些数据库可能非常大,因此需要多个次数据库文件,或可在不同的硬盘中存储次数据库文件,这样可以同时对几个硬盘做数据存取,提高了数据处理的效率。对于服务器型的计算机尤为有用。

(2) 事务日志文件

事务日志文件包含用于恢复数据库的日志信息。一个数据库可以有一个或多个事务日志文件。

SQL Server 中采用提前写方式的事务,即对数据库的修改先写入事务日志中,再写入数据库。其具体操作是,系统先将更改操作写入事务日志中,再更改存储在计算机缓存中的数据,为了提高执行效率,此更改不会立即写到硬盘中的数据库,而是由系统以固定的时间间隔将更改过的数据批量写入硬盘。SQL Server 有个特点它在执行数据更改时会设置一个开始点和一个结束点,如果尚未到达结束点就因某种原因使操作中断,则在 SQL Server 重新启动时会自动恢复已修改的数据,使其返回未被修改的状态。由此可见,当数据库破坏时可以用事务日志恢复数据库内容。

6.3 系统开发工具介绍

6.3.1 Delphi 介绍^{[22][23]}

Delphi, 是 Windows 平台下著名的快速应用程序开发工具。它的前身, 即是 DOS 时代盛行一时的“Borland Turbo Pascal”, 最早的版本由美国 Borland 公司于 1995 年开发。主创者为 Anders Hejlsberg。经过数年的发展, 此产品也转移至 Embarcadero 公司旗下。Delphi 是一个集成开发环境 (IDE), 使用的核心是由传统 Pascal 语言发展而来的 Object Pascal, 以图形用户界面为开发环境, 透过 IDE、VCL 工具与编译器, 配合连结数据库的功能, 构成一个以面向对象程序设计为中心的应用程序开发工具。

Delphi 拥有一个可视化的集成开发环境 (IDE), 采用面向对象的编程语言 ObjectPascal 和基于部件的开发结构框架。Delphi 它提供了 500 多个可供使用的构件, 利用这些部件, 开发人员可以快速地构造出应用系统。开发人员也可以根据自己的需要修改部件或用 Delphi 本身编写自己的部件。主要特点如下:

(1)直接编译生成可执行代码, 编译速度快。由于 Delphi 编译器采用了条件编译和选择链接技术, 使用它生成的执行文件更加精炼, 运行速度更快。在处理速度和存取服务器方面, Delphi 的性能远远高于其他同类产品。

(2)支持将存取规则分别交给客户机或服务器处理的两种方案, 而且允许开发人员建立一个简单的部件或部件集合, 封装起所有的规则, 并独立于服务器和客户机, 所有的数据转移通过这些部件来完成。这样, 大大减少了对服务器的请求和网络上的数据传输量, 提高了应用处理的速度。

(3)提供了许多快速方便的开发方法, 使开发人员能用尽可能少的重复性工作完成各种不同的应用。利用项目模板和专家生成器可以很快建立项目的构架, 然后根据用户的实际需要逐步完善。

(4)具有可重用性和可扩展性。开发人员不必再对诸如标签、按钮及对话框等 Windows 的常见部件进行编程。Delphi 包含许多可以重复使用的部件, 允许用户控制 Windows 的开发效果。

(5)具有强大的数据存取功能。它的数据处理工具 BDE(Borland Database Engine)是一个标准的中介软件层, 可以用来处理当前流行的数据格式, 如 xBase、Paradox 等, 也可以通过 BDE 的 SQLLink 直接与 Sybase、SQLServer、Informix、Oracle 等大型数据库连接。Delphi 既可用于开发系统软件, 也适合于应用软件的开发。

(6)拥有强大的网络开发能力，能够快速开发 B/S 应用，它内置的 IntraWeb 和 ExpressWeb 使得对于网络的开发效率超过了其他任何的开发工具。

(7)Delphi 使用独特的 VCL 类库，使得编写出的程序显得条理清晰，VCL 是现在最优秀的类库，它使得 Delphi 在软件开发行业处于一个绝对领先的地位。用户可以按自己的需要，任意的构建、扩充、甚至是删减 VCL，以满足不同的需要。

6.3.2 Object Pascal 软件开发语言介绍

Object Pascal 是一种高级编译语言，具有强类型特性，支持结构化和面向对象编程。它的优点包括代码的易读性、快速编译，以及支持多个单元文件从而实现模块化编程。

Object Pascal 的基本语句简介^[24]:

- 声明语句

声明语句主要有类型定义、常量定义、变量定义、标号定义、函数或过程的说明等。如标号定义的语法格式为：Label 标号名 1, 标号名 2...

- 赋值语句

赋值语句就是给变量、对象成员等赋值的语句，一般赋值语句的语法格式为：
变量名 | 函数名 | 对象.字段名: =表达式;

- 过程或函数调用语句

有些过程或函数调用语句可作为独立的语句，如 Inc (...), Dec(...), Halt, exit.

- Goto 语句

Goto 语句用于实现无条件的转换，其语法格式为：Goto 标号;。其中标号必须事先用 Label 语句定义，并在转移的目标位置插入相应的标号。

- 条件语句 (If, Case)

条件语句也称为选择语句、分支语句，其作用是根据条件从多个语句中选择一个语句进行执行，按可选择语句的数量的不同可分为双分支语句 (If 语句) 和多分支语句 (Case 语句)。

- 循环语句 (While, Repeat, for)

当某个语句要重复执行多次时可以使用循环语句，循环语句有三种：While 语句，Repeat 语句，for 语句。

- 转向语句 (break, continue)

break 语句：循环遇到 break 立即跳出循环。

continue 语句：循环遇到 continue 结束本次循环。

- 其他语句 (Exit, Halt)

Exit: 退出过程或函数。

Halt: 终止程序执行, 返回操作系统。

6.4 数据库服务器的设计与实现

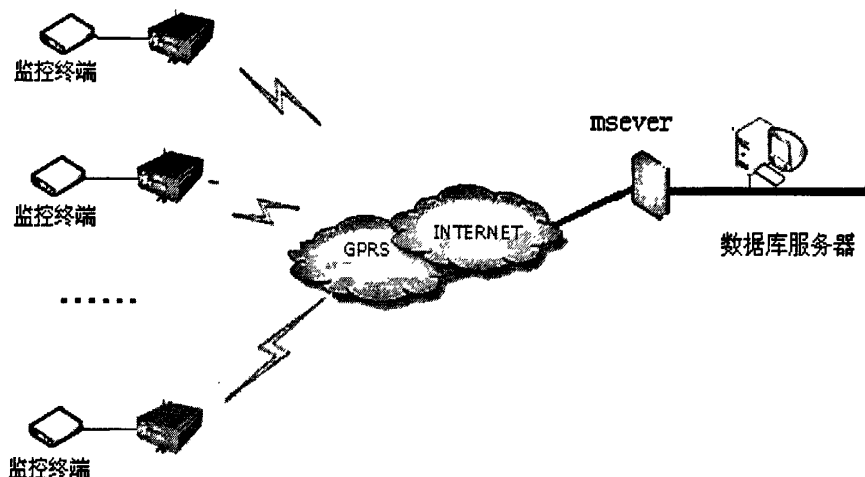


图 6.2 数据库服务器接收终端采集数据原理图

Fig. 6.2 Database server receive data diagram

数据库服务器作为监控中心的数据存储层, 要实现管理人员数据、定位数据、地理信息数据和其他业务数据的存储。数据库服务器接收终端采集数据原理图如图 6.2 所示。

6.4.1 数据库的建立

启动 SQL Server Management Studio, 选择连接的服务器、通过身份验证后进入对象资源管理器界面, 展开目录, 右击“数据库”选项, 选择“新建数据库”, 如图 6.3 所示。

进入“新建数据库”窗口后, 在“常规”选项卡中输入数据库名称“ycjkxt”后, 系统会默认产生数据文件“ycjkxt.mdf”和日志文件“ycjkxt_log.ldf”并显示文件的默认属性 (如数据库文件和日志文件的自动增长属性和路径属性等), 用户可以自行修改这些默认属性。在该窗口中设置完要创建的数据库后, 单击“确定”按钮, 完成数据库的创建, 创建好的奶源运输安全远程监控后台数据库如图 6.4 所示。

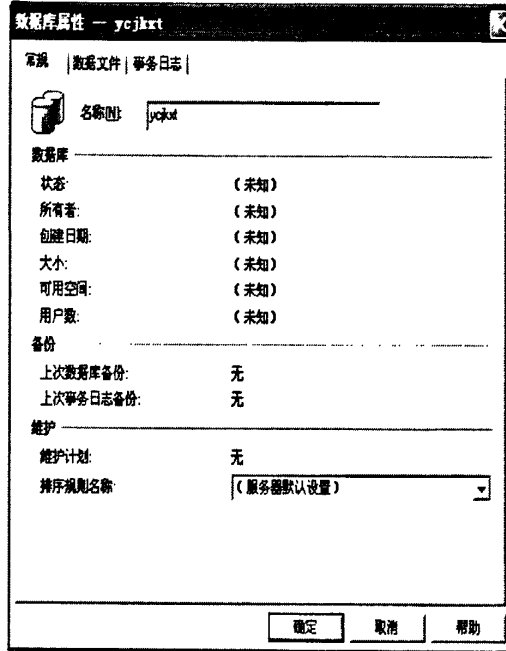


图 6.3 新建数据库对话框
Fig. 6.3 New database dialog

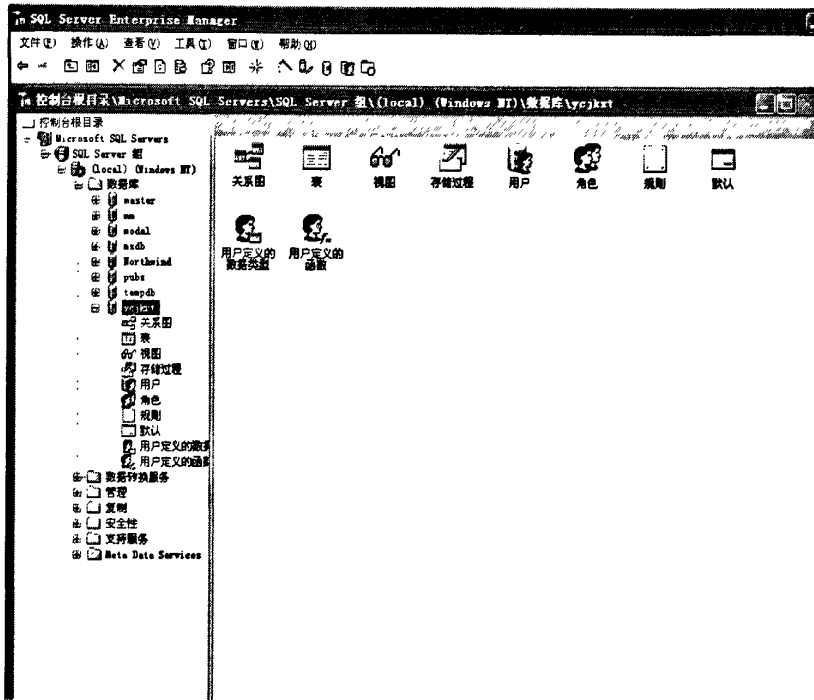


图 6.4 新建好的数据库界面
Fig. 6.4 Newing database interface

6.4.2 远程监控数据表格的设计

创建了“ycjkxt”数据库后，就要创建表来存储各种信息。在对象资源管理界面中，选择“ycjkxt”数据库，展开目录，右击“表”，选择“新建表”，进入表设计窗口，如图 6.5

所示。在该窗口中主要是完成数据表列的设计，包括列名、列的数据类型和列的属性等，列具有的属性包括允许空属性、默认值属性和标识属性等。对表设置完以后将其保存为相应功能的名字。

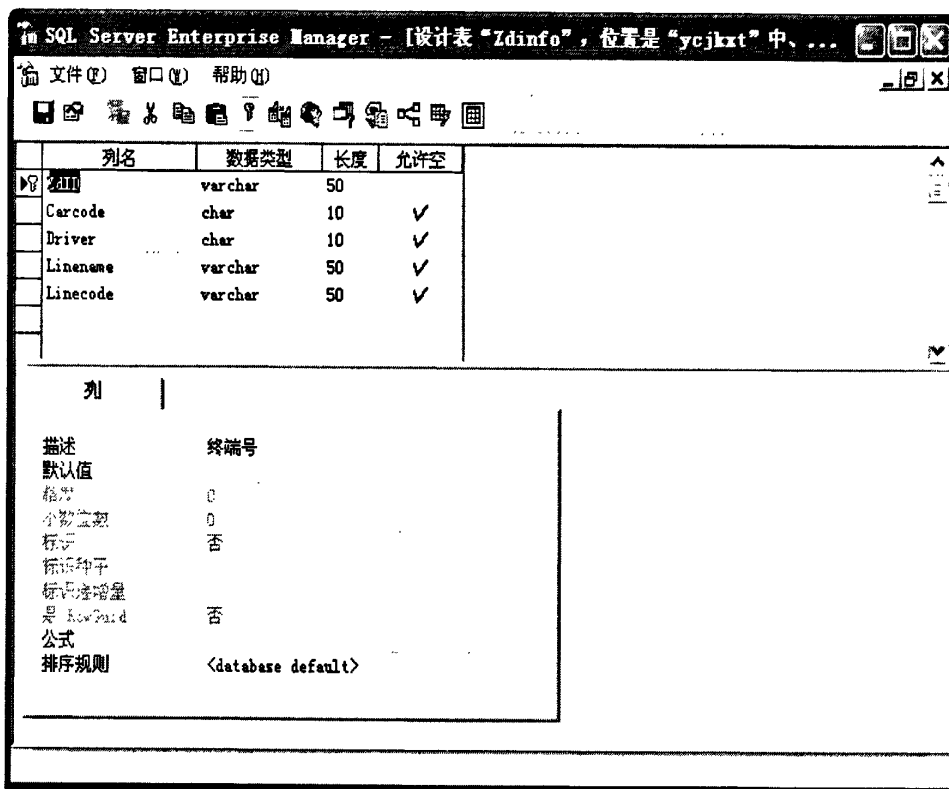


图 6.5 表设计窗口

Fig. 6.5 The table design window

本项目的基础数据表格包括：监控终端信息表、线路信息表、线路路由表、监控人员信息表、定位数据采集表、定位现状表、图像信息表。

1、监控终端信息表 Zdinfo

监控终端管理及定位列队的建立：线路数据建立完成后，我们还需要对线路分配定位终端并对监控终端建立定位队列，通过定位队列，维持位置数据的接收，从而实现对监控终端定位的功能。

监控终端信息表设计如图 6.6 所示。

图 表“Zdinfo”中的数据, 位置是“ycjkxt”中、“(local)”上

ZdID	Carcode	Driver	NZname	NZcode
240305001023019	01001	王立	奶站1	#1
240305001023020	01002	李洪	奶站1	#1
240305001023021	01003	王玉	奶站1	#1
240305001023031	01004	吴青	奶站2	#2
240305001023032	01003	李丰	奶站2	#2
240305001023033	01006	王玉洁	奶站2	#2
240305001023041	01007	苏林	奶站3	#3
240305001023042	01008	南丁	奶站3	#3
240305001023043	01009	葛志	奶站3	#3
240305001023051	02001	马晓	奶站4	#4
240305001023052	02002	马鸿	奶站4	#4
240305001023053	02003	李健	奶站4	#4

图 6.6 监控终端信息数据表 Zdinfo

Fig. 6.6 The monitoring terminal information table Zdinfo

其中各字段表示的含意如下:

ZdID——终端号; Driver——奶罐车驾驶员

Carcode——车辆代码; NZname——所属奶站名称

NZcode——奶站代码

2、奶站信息表 NZinfo

奶站信息表存储了奶罐车所属奶站的基本信息。表格的设计如图 6.7 所示。

图 表“NZinfo”中的数据, 位置是“ycjkxt”中、“(local)”上

NZcode	NZname	Jdu	Wdu	length
#1	奶站1	111.649311	40.795678	1056
#2	奶站2	111.653003	40.795248	10200
#3	奶站3	111.659667	40.792761	9900
#4	奶站4	111.658495	40.790511	1865

图 6.7 奶站信息数据表 NZinfo

Fig. 6.7 The milk stands' information table NZinfo

其中各字段表示的含意如下:

NZcode——奶站代码; NZname——奶站名称;

Jdu——奶站经度; Wdu——奶站纬度;

Length——奶站距企业线路长度;

3、奶罐车运输线路路由表 LinePoint

前期通过 GPS 采集仪采集各线路路由点 (沿线路没隔 50 米采集一个坐标点) 的经纬度

数据，然后将线路的坐标数据存放在数据库 LinePoint 表中。当客户端进行显示线路的请求时，服务器响应客户端请求，对 LinePoint 表进行检索，将获得的相关奶站的路由点数据按照约定的格式编码发送到请求的客户端，客户端按照约定格式解析后便可在电子地图上生成定位线路路由图。

奶罐车运输线路路由表 LinePoint，它存储了生成路线图的路由点经纬度信息，如图 6.8 所示：

表 "LinePoint" 中的数据, 位置是 "ycjkr" 中、"(local)" 上

PointID	NZname	NZcode	Jdu	Wdu
1000100	奶站3	#3	111.659609	40.792857
1000101	奶站3	#3	111.659505	40.792849
1000102	奶站3	#3	111.659365	40.792831
1000103	奶站3	#3	111.659191	40.792796
1000104	奶站3	#3	111.659040	40.792752
1000105	奶站3	#3	111.658889	40.792708
1000106	奶站3	#3	111.658715	40.792664
1000107	奶站3	#3	111.658553	40.792611
1000108	奶站3	#3	111.658379	40.792559
1000109	奶站3	#3	111.658216	40.792515
1000110	奶站3	#3	111.658054	40.792471
1000111	奶站3	#3	111.657891	40.792427
1000112	奶站3	#3	111.657705	40.792374
1000113	奶站3	#3	111.657519	40.792321
1000114	奶站3	#3	111.657334	40.792260
1000115	奶站3	#3	111.657125	40.792190
1000116	奶站3	#3	111.656927	40.792146
1000117	奶站3	#3	111.656695	40.792093
1000118	奶站3	#3	111.656486	40.792093
1000119	奶站3	#3	111.656277	40.792093
1000120	奶站3	#3	111.656068	40.791900
1000121	奶站3	#3	111.655848	40.791829
1000122	奶站3	#3	111.655615	40.791768
1000123	奶站3	#3	111.655383	40.791697
1000124	奶站3	#3	111.655174	40.791654
1000125	奶站3	#3	111.654907	40.791592
1000126	奶站3	#3	111.654698	40.791522
1000127	奶站3	#3	111.654431	40.791390
1000128	奶站3	#3	111.654268	40.791390
1000129	奶站3	#3	111.654083	40.791337
1000130	奶站3	#3	111.653885	40.791284

图 6.8 线路路由表 LinePoint

Fig. 6.8 The routing table LinePoint

其中各字段表示的含意如下：

PointID——路由点序号；Jdu——经度；

Wdu——纬度；NZname ——所属奶站名称

NZcode——所属奶站代码

4、监控人员信息表 Managerinfo

数据库中监控人员信息表的设计如图 6.9 所示。它储存了系统监控人员的基本信息，使得系统只对系统内部登记的管理人员开放，其他用户无法登陆监控系统。

Managername	Scode	Managercode	Departmentcode	Sex	BirthData
赵鑫	30712001	0001	908	女	1983-5-1
王宇	30712003	0002	908	女	1984-2-9
李娜	85693211	0003	908	女	1978-9-14
李新	61586235	0004	908	男	1977-5-8
小红	24636652	0005	908	女	1980-12-12
南丁	1646aa3131	0006	908	男	1982-2-11

图 6.9 管理人员基本信息表 Managerinfo

Fig. 6.9 Managers' basic information table Managerinfo

其中各字段表示的含意如下：

Managername——管理员姓名； Scode ——密码；

Managercode——管理员编码； Departmentcode ——部门代码；

Sex——性别； BirthData——出生日期

5、定位数据采集表

GPS 数据表 GPScjponit 作为整个系统的核心数据之一，它负责存储了所有监控终端的采集点坐标，并且负责将线路采集点的所属奶站、所属监控终端等属性数据进行规划，如图 6.10 所示。

其中各字段表示的含意如下：

cjpointID——采集点序号； Jdu ——经度；

Wdu——纬度； ZdID——终端号

NZcode——奶站代码； cjtime——采集时间；

表 "GPScjpoint" 中的数据, 位置是 "ycjbrt" 中、"(local)" 上

cjpointID	NZcode	zdID	cjtime	Jdu	Wdu
100000001	#1	240305001023019	2010-3-11 9:11::	111.614512	40.816667
100000002	#1	240305001023019	2010-3-11 9:12::	111.614512	40.816667
100000003	#1	240305001023019	2010-3-11 9:13::	111.614512	40.816667
100000004	#1	240305001023019	2010-3-11 9:14::	111.614512	40.816667
100000005	#1	240305001023019	2010-3-11 9:15::	111.614512	40.816667
100000006	#1	240305001023019	2010-3-11 9:16::	111.614512	40.816667
100000007	#1	240305001023019	2010-3-11 9:17::	111.614512	40.816667
100000008	#1	240305001023019	2010-3-11 9:18::	111.614512	40.816667
100000009	#1	240305001023019	2010-3-11 9:19::	111.614512	40.816667
100000010	#1	240305001023019	2010-3-11 9:20::	111.614512	40.816667
100000011	#1	240305001023019	2010-3-11 9:21::	111.614512	40.816667
100000012	#1	240305001023019	2010-3-11 9:22::	111.614512	40.816667
100000013	#1	240305001023019	2010-3-11 9:23::	111.614512	40.816667
100000014	#1	240305001023019	2010-3-11 9:24::	111.614512	40.816667
100000015	#1	240305001023019	2010-3-11 9:25::	111.614512	40.816667
100000016	#1	240305001023019	2010-3-11 9:26::	111.614512	40.816667
100000017	#1	240305001023019	2010-3-11 9:27::	111.614512	40.816667
100000018	#1	240305001023019	2010-3-11 9:28::	111.614512	40.816667
100000019	#1	240305001023019	2010-3-11 9:29::	111.614512	40.816667
100000020	#1	240305001023019	2010-3-11 9:30::	111.614512	40.816667
100000021	#1	240305001023019	2010-3-11 9:31::	111.614512	40.816667
100000022	#1	240305001023019	2010-3-11 9:32::	111.614512	40.816667
100000023	#1	240305001023019	2010-3-11 9:33::	111.614512	40.816667
100000024	#1	240305001023019	2010-3-11 9:34::	111.614512	40.816667

图 6.10 GPS 定位数据采集表 GPScjpoint
Fig. 6.10 GPS location data collection form GPScjpoint

6、终端位置现状表 Zdstate

表 "Zdstate" 中的数据, 位置是 "ycjbrt" 中、"(local)" 上

ZdID	NZcode	Carcode	Jdu	Wdu	Imagename
240305001023019	#1	01001	111.645265	40.819965	2010-3-11 14:20
240305001023020	#1	01002	111.645265	40.819965	
240305001023021	#1	01003	111.645265	40.819965	
240305001023053	#4	04003	111.645265	40.819965	
240305001023033	#2	02001	111.645265	40.819965	
240305001023032	#2	02002	111.645265	40.819965	
240305001023033	#2	02003	111.645265	40.819965	
240305001023041	#3	03001	111.645265	40.819965	
240305001023042	#3	03002	111.645265	40.819965	
240305001023043	#3	03003	111.645265	40.819965	
240305001023051	#4	04001	111.645265	40.819965	
240305001023052	#4	04002	111.645265	40.819965	
240305001023052	#4	04003	111.645265	40.819965	

图 6.11 终端位置现状表 Zdstate
Fig. 6.11 Terminal position status sheet Zdstate

其中各字段表示的含意如下:

ZdID----终端号; Jdu----经度; Wdu----纬度; NZcode----奶站代码;

Carcode——车辆代码; Imagename——终端最近一次拍摄的图像

7、图像信息存储表 ZdImage

图像信息存储表 ZdImage 作为整个系统的另一个核心数据，它负责存储了所有监控终端采集的图像，图像接收下来之后以采集图像的时间命名存储，如图 6.12 所示：

Imagecode	imagename	NZcode	ZdID
1000000045	2010-3-12 9:21: #1	#1	240305001023019
1000000046	2010-3-12 9:56: #1	#1	240305001023019
1000000047	2010-3-12 10:21 #1	#1	240305001023019
1000000048	2010-3-13 14:21 #1	#1	240305001023019
1000000049	2010-3-12 14:26 #1	#1	240305001023019
1000000050	2010-3-12 14:27 #1	#1	240305001023019
1000000051	2010-3-12 14:29 #1	#1	240305001023019
1000000052	2010-3-12 14:31 #1	#1	240305001023019
1000000053	2010-3-12 14:33 #1	#1	240305001023019
1000000054	2010-3-12 14:35 #1	#1	240305001023019
1000000055	2010-3-12 14:38 #1	#1	240305001023019
1000000056	2010-3-12 14:39 #1	#1	240305001023019
1000000057	2010-3-12 14:41 #1	#1	240305001023019
1000000058	2010-3-12 14:43 #1	#1	240305001023019

图 6.12 图像信息存储表 ZdImage

Fig. 6.12 Images-stored form ZdImage

其中各字段表示的含意如下：

ZdID——终端号； Imagename ——图像名称；

Imagecode ——图像序号； NZcode——奶站代码；

8、数据表格的关联

基础数据表之间的关联关系如图 6.13 所示。

6.4.3 监控终端采集数据的导入

首先，在建立系统数据库时，在应用服务器端设计数据加载函数接口，程序实现功能如下：自动完成与 mserver 的连接，完成对 GPRS DTU 传输过来的采集数据的存储功能，即将采集数据分类放入各自的基础数据表中。程序代码见附录 1。

然后，在应用服务器上安装 mServer 无线通信服务软件^[26]，它为用户和远程监控终端之间的通信服务。它是以 Windows 服务的形式运行于操作系统后台，不需要管理员干预。远程监控终端与 mServer 进行通信，然后应用服务器再和 mServer 通信获得监控终端的采集的数据信息。这个功能的实现是利用了 DCC.DLL，它是驿唐科技开发的动态链接库 (Dynamic-link

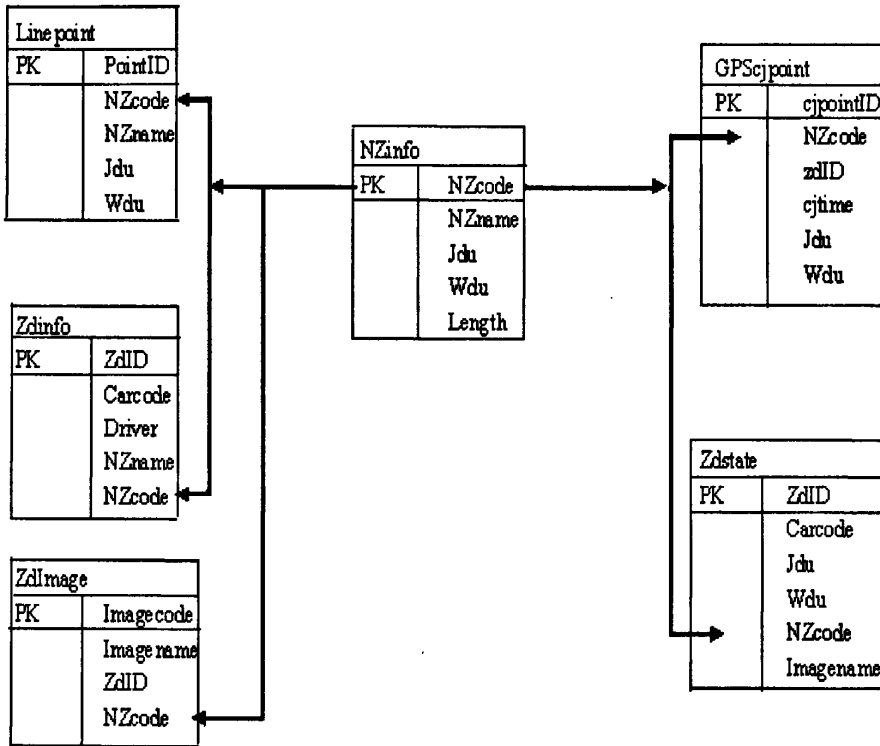


图 6.13 数据表之间的关联关系

Fig. 6.13 The relationship between data tables

Library, DLL)。应用服务器与 mServer 之间通信的消息格式如表 6.1 所示： [28]

在应用服务器启动时，自动调用接口函数 `dcc_init(mode, port, dc_ip, dc_port, block)`，它根据指定的参数去初始化与 mServer 的连接，当返回的连接句柄 `dcc_hdl` 为正数则表示连接成功，并存入“连接状态记录”文件中。

表 6.1 消息格式

Tab6.1. Message format

名称	字节	说明
ID	4	一个消息包的开始，为四个字节的 0x7e
IMEI	16	远程监控终端 GPRS 模块的唯一标识
Name	16	远程监控终端 GPRS 模块的别名
Msg Type	1	消息类型
Reserved	1	保留字段
Msg Len	2	Msg Body 的实际长度，两个字节长，以网络字节序存放
Msg Body	Msg Len 指明	消息体，不同的消息类型有不同的消息体

与 mServer 连接成功后，数据加载函数调用函数 `dcc_msg_receive(int dcc_hdl, ref dc_msg`

msg)实现对监控终端进行数据收发。

数据加载函数调用的DCC.DLL中的接口函数有：

```
[DllImport("dcc.dll", EntryPoint = "dcc_init")]
public static extern int dcc_init(int mode, UInt16 port, string dc_ip, UInt16 dc_port, int block);
[DllImport("dcc.dll", EntryPoint = "dcc_msg_receive")]
public static extern int dcc_msg_receive(int dcc_hdl, ref dc_msg msg);
[DllImport("dcc.dll", EntryPoint = "dcc_close")]
public static extern int dcc_close(int dcc_hdl);
```

以下是在数据加载函数中定义的用于存放接收数据的结构体：

```
type
    Pdc_msg = ^dc_msg;
    dc_msg=record
        ImeiDTU:array[1..IMEI_LEN+1] of char;      //终端的IMEI号
        NameDTU:array[1..DTU_NAME+1] of char;    //终端的别名
        msg_type:byte;                            //消息类型
        reserved:byte;                            //保留
        msg_len:word;                              //消息长度
        msg_body:array[1..1942] of byte;         //消息体
    end;
```

最后，编程将接收下来的数据根据不同的终端号、消息类型，剔除无用信息后，分别存入对应分类的基础数据表中。

6.4.4 ADO 数据库访问技术^[25]

在本系统中采用 ADO 技术对数据库进行访问，它是应用程序和数据库之间的一座重要桥梁。动态数据对象（ADO）用来为 Visual InterDev 等应用程序提供灵活的和可升级的数据库连接功能。特别对于设计基于网络的数据库访问，ADO 提供了一种基于对象的网络上的访问方法，为开发者提供了一系列强大的数据库命令来操作数据。

ADO 把绝大部分的数据库操作封装在 7 个对象中，在 ASP 页面中编程调用这些对象执行相应的数据库操作。ADO 建立了基于 Web 方式访问数据库的脚本编写模式，不仅支持任何大型数据库的核心功能，而且支持许多数据库所特有的特性。ADO 使用本机数据源，通过

ODBC 访问数据库。ADO 主要优点是易用、高速、占用内存和磁盘空间少，所以非常适合于作为服务器端的数据库访问技术。

目前 ADO 包括 Command、Connection、Recordset 等 7 个对象和一个动态的 Properties 集合，绝大部分的数据库访问任务都可以通过他们的组合来完成。一般使用 ADO 访问数据库的 ASP 脚本程序应该使用 Connection 对象建立并管理与远程数据库的连接；使用 Command 对象提供灵活的查询；而使用 Recordset 对象访问数据库所返回的结果。这三者是 ADO 中最基本也是最核心的对象。ADO 组件又称为数据库访问组件，用于访问存储在数据库或其他表格式数据结构中的信息。ADO 组件一般提供了以下 7 种对象：

- ①Connection 对象：提供对数据库的连接服务。
- ②Recordset 对象：由数据库服务器所返回的记录集。
- ③Fields 对象：由数据库服务器所返回的单一数据字段。
- ④Command 对象：定义对数据源操作的命令。
- ⑤Parameters 对象：表示 Command 对象的参数。
- ⑥Property 对象：单独一个 Property 对象，提供属性功能。
- ⑦error 对象：提供处理错误的功能。

使用 ADO 对象可以建立和管理数据库的连接，按数据库服务器的要求获取数据，执行更新、删除、添加数据，获取 ODBC 的错误信息等。

6.4.5 ADO 数据库连接参数配置

在应用服务器运行过程中，所有数据业务的处理，都通过 ADO 连接与数据库进行通信。因为采用分布式结构，数据库服务器可以放置在网络中的不同位置上，所以，需要对数据库的连接进行动态的配置，配置过程实现如下。

首先调用 Windows 系统提供的 ADO 连接配置 API，如图 6.14 所示。然后，单击 Build 按钮，出现数据库连接模式选择界面，如图 6.15 所示。在连接模式选择界面中，我们根据实际设计中使用的 SQL Sever 数据库，选择 Microsoft OLE DB Provider for SQLSever 模式，这是微软提供的针对 SQLSever 数据库的 ADO 连接。双击选项或单击“下一步”按钮，接下来就是 SQLSever 的 ADO 配置界面，如图 6.16 所示。

在服务器名称栏中输入我们已建好的数据库名称“ycjkxt”，在用户名称中输入建立数据库时的用户名及密码。单击“测试连接”按钮，跳出“测试连接成功”窗体后就表示成功完成了 ADO 连接。

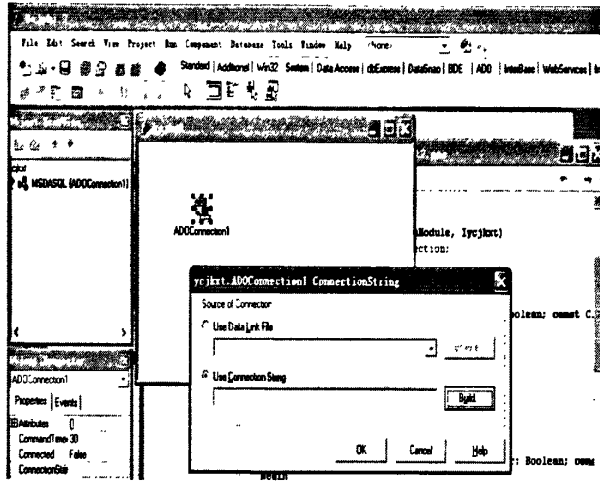


图 6.14 ADO 连接字符串配置

Fig. 6.14 ADO config

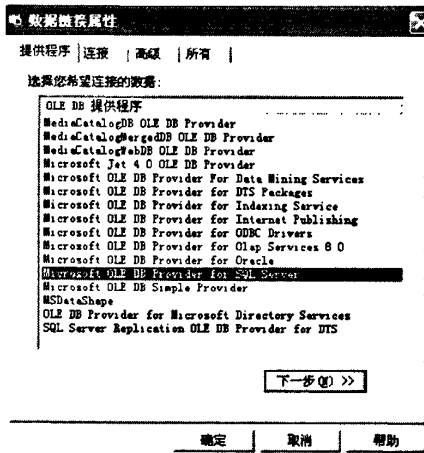


图 6.15 数据库连接模式选择界面

Fig. 6.15 Database schema choose interface

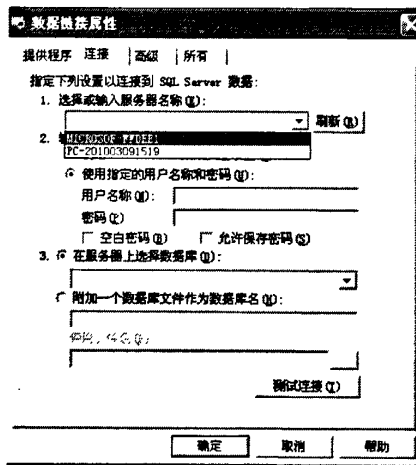


图 6.16 SQLSever 的 ADO 配置界面

Fig. 6.16 SQLSever's ADO config

测试连接成功后，数据库的 ADO 连接字符串就保存到配置文件中。本项目中实现 ADO

配置、保存的实现代码如下：

```
Procedure TycjkxtMainForm.ActionSetADOCntExecute(Sender: TObject);  
var  
    Sini:string;  
    Ini_file: TIniFile;  
    ADOCnt:TADOConnection;  
begin  
    Sini:= S_M_AppPath + 'PUcRtsrvr.ini';  
    Ini_file := TIniFile.Create(Sini);  
    ADOCnt:=TADOConnection.Create(application);  
    ADOCnt.Close;  
    try  
        if EditConnectionString(ADOCnt) then  
            begin  
                Ini_file.WriteString('ADOCnt','Connectstring',ADOCnt.ConnectionString);  
                ws_okMsg(handle,'请重新启动程序');  
            end  
        finally  
            Ini_file.Free;  
            ADOCnt.Free;  
        end;  
    end;  
end;
```

6.5 应用服务器功能的实现

应用服务器定义：是指通过各种协议把商业逻辑暴露给客户端的程序。它提供了访问商业逻辑的途径以供客户端应用程序使用。应用服务器使用此商业逻辑就像调用对象的一个方法一样。

在该系统结构中，应用服务器作为独立的核心层，封装了客户端业务处理和对数据库的访问。它可以被看作一个中间件，以接口的形式输出业务处理，供客户端调用，同时负责客

户端与数据库之间的数据访问。这种 3 层分布式工作机制的优势是非常明显的：

1、简化客户端程序的开发，把业务处理封装在通用的中间件服务器中，有利于监控系统的开发升级。

2、实现了分布式数据处理，在多层体系结构中数据库、应用服务器、客户端可以独立分布在网络的不同位置上，实现灵活配置。

3、应用服务器中间层封装了企业逻辑，统一进行业务处理和数据库访问，从而有效节省网络数据流量，是事务处理能力更强，运行效率更高。

6.5.1 应用服务器的建立

在 Delphi 中建立应用服务器的具体实现的流程如下：

在 Delphi 的 File 菜单中的 New 子菜单中选择 Other 选项，在弹出的对话框中切换到 Multitier 选项卡，双击 Remote Data Module 图标，弹出对话框，如图 6.17 所示。

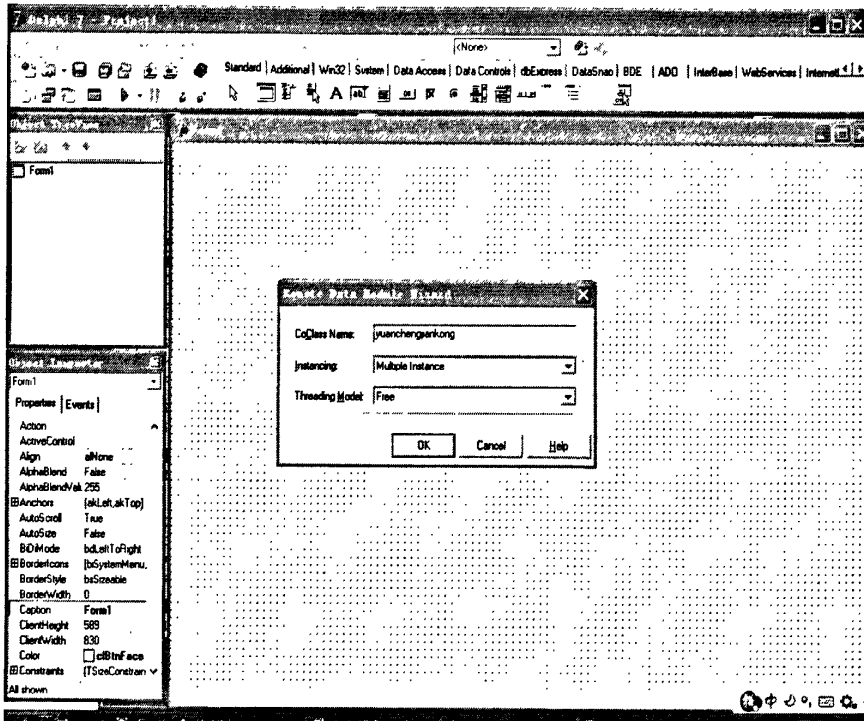


图 6.17 新建远程监控系统工程

Fig.6.17 new remote control system engineering

在第一个文本框中输入远程数据模块的名字“ycjkxt”。系统会在这个名字前加“T”，作为远程数据模块的自动化接口，供客户端调用。同时生成一个基于同样名字前面加“T”的 (TRemoteDataModule) 的继承类，该类负责解释这个接口。要实现客户端调用，还需要在客户端的连接控件的 ServerName 属性中指定刚才输入的名字。

对应用服务器运行实例化类型和线程模型，要根据设计要求和系统的性能配置进行合理

的选择^[27]。

1、应用服务器的实例化模型选择

在 Instancing 下拉列表框中有 3 个选项可供选择, 分别是 Internal Instance、Single Instance、Multiple Instance, 它们是 Delphi 中 midas 应用程序 COM 服务器实例化模型种类。

Internal Instance 选项是指创建一个 In-Process 的 COM 程序, 即以 DLL 模式存在和运行的服务器。使用 Single Instance 模型, 服务器为每个请求的客户端都生成一个运行实例。使用 Multiple Instance 模型, 客户端程序共享服务器程序。也就是说服务器运行后, 只开辟一个内存空间, 只生成一个运行实例, 不会根据并发客户数的改变而调整实例数量。该模型可以让服务器同时处理多个客户端程序的链接, 换句话说就是所有客户端程序共享单一进程的应用程序服务器, 客户端的请求需要排队。该模式的最大优点是, 激活及运行的效率较快, 不会耗用服务器太多系统资源, 而且管理比较方便。

因为服务器性能配置不是很高, 查询数据量较小, 出于后期维护方面的考虑, 保证系统足够健壮的情况下, 本系统选择使用 Multiple Instance 模式。

2、应用服务器的线程模型选择

Threading Model 选项里选择的是应用服务器运行的线程模型, 在下拉列表框中共有 5 种模式供选择。

Single 模型是针对 Internal Instance 模式的, 它一次只能接收一个客户端发出来的请求。由于是单一线程模式, 所以不会有多任务同时进行的问题。Neutral 这种模式允许多个客户端在同一个时间以不同的线程来调用服务器程序, 不过 COM 对象必须确保不会有前台程序调用的冲突。Apartment 线程模式一次只能处理一个客户端发出的请求, 特别对于 DLL 形式的应用服务器来说, 它可以通过生成多个 COM 组件的方式, 以多线程来分别处理用户并发请求。Free 线程模式可以让应用服务器产生的每个实例通过各自的线程处理每个客户端发出的请求, 所以服务器程序本身必须采用多线程的程序写法, 必须保护每个实例所处理的数据区域不会互相冲突。这种模式适用于使用 ADO 数据连接。Both 这种模式与 Free 基本相同, 唯一的区别在于它对客户端接口的回调是序列化的。基于各线程模式的特点, 本系统用服务器选择采用 Free 线程模式。

各部分选择合适的选项后, 点击“确定”按钮, 就建立好了远程数据模块实例 TRemoteDataModule, 并生成类型库 PUcRtSrvr_TLB, 用作 COM 服务器的接口, 远程数据模块是应用服务器的核心部件。

6.5.2 应用服务器的逻辑功能

应用服务器的功能是将业务实现封装成函数,通过输出接口的方式,供客户端调用,应用服务器作为基于 COM 组件的服务,其核心逻辑功能如图 6.18 所示。

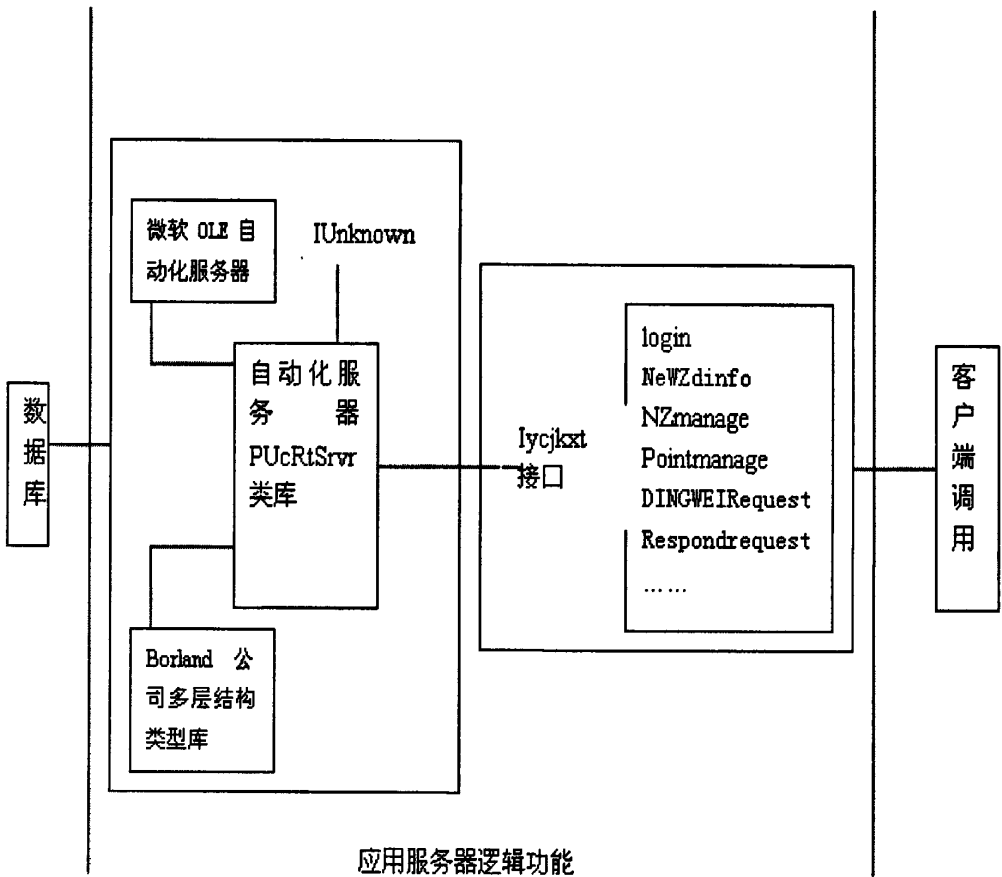


图 6.18 应用服务器的逻辑功能

Fig.6.18 The logic functions of the application sever

根据课题研究中的设计要求,我们将服务器端的逻辑功能集成于接口中,客户端通过接口来调用服务器内的函数,实现需要的业务。服务器通过 ADO 等连接方式,负责实现对数据库的访问,并且所有数据库访问控件都集中放在 Remote Data Module 数据控件容器里进行管理和调用。

6.5.3 应用服务器部分函数接口的实现

1. 登录函数接口的设计

将 ADO 数据连接控件 ADOConnection 放入 Remote Data Module 的数据控件容器里,命名为 ADOConnection1,如图 6.19 所示。

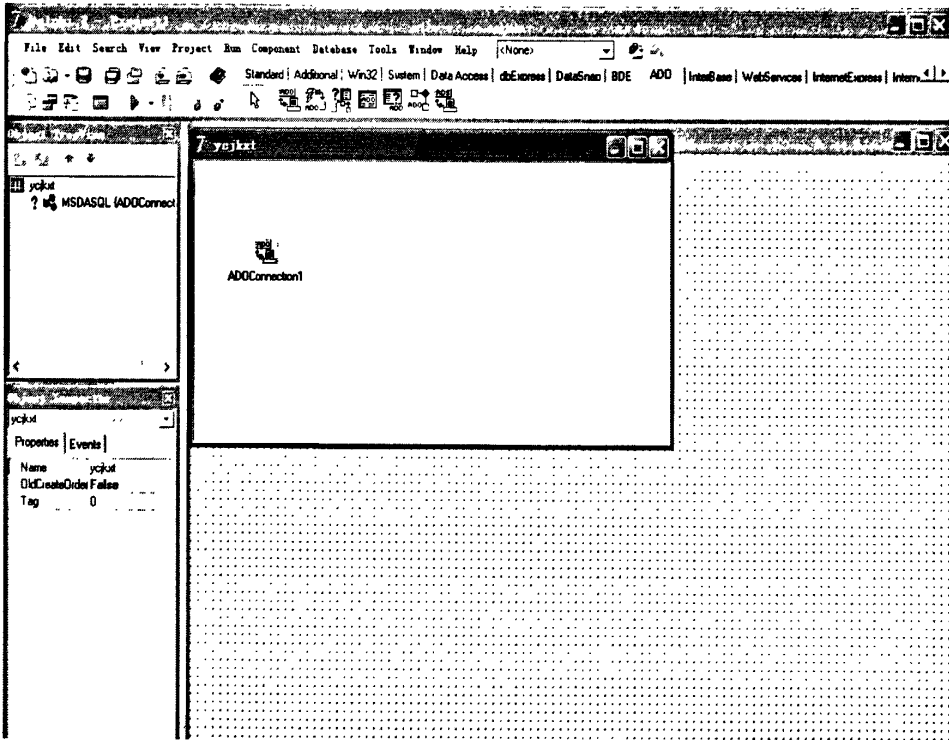


图 6.19 Remote Data Module 数据控件容器
Fig.6.19 Remote Data Module datagrid

ADOConnection 是负责服务器与数据库进行 ADO 连接的控件。ADOConnection 提供了与数据库的连接途径，其他的数据库控件通过 ADO 连接完成数据访问的功能。

首先在服务器的 Remote Data Module 容器中再加入一个 ADOQuery，命名为 ADOQryLogin。ADOQuery 是 ADO 的数据查询控件，这里我们可以将其 Connection 属性设置为“ADOConnection1”。

然后，在应用服务器端编写认证登录的函数，通过接口方式，供客户端调用，实现登录功能。在 Delphi 的 View 下拉菜单中选择 Type Library，弹出该应用服务器程序的类型库窗口。在类型库中建立客户端登录的接口函数。

这样我们就在类型库中构建完成登录函数 Login。登录函数 Login 的实现流程图如图 6.20 所示。

在类型库中，编写的函数 function Login 被分别声明在 Iycjkxt、IycjkxtDisp 接口中。Iycjkxt 接口在服务器和客户端都要实现，只有通过 Iycjkxt 接口，客户端才能调用服务器端的应用服务。IycjkxtDisp 是一种纯粹的自动化接口，这种接口就只能通过自动化的方式来调用，COM 组件的事件一般都用的是这种形式的接口。所以通过这种接口的封装，函数就可以在 COM 结构下被调用。同时，在类型库中将远程数据模块实例化，以 Iycjkxt 接口方式输出服务，这样客户端就可以调用了。

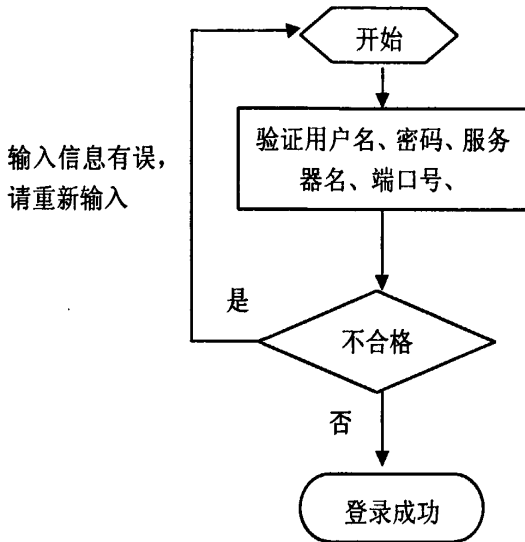


图 6.20 Login 登录函数实现流程图
Fig. 6.20 Login function flow chart

Login 函数可以实现客户端的管理员姓名、密码验证登录，代码见附录 1。

2. 服务器端定位终端信息录入函数的设计

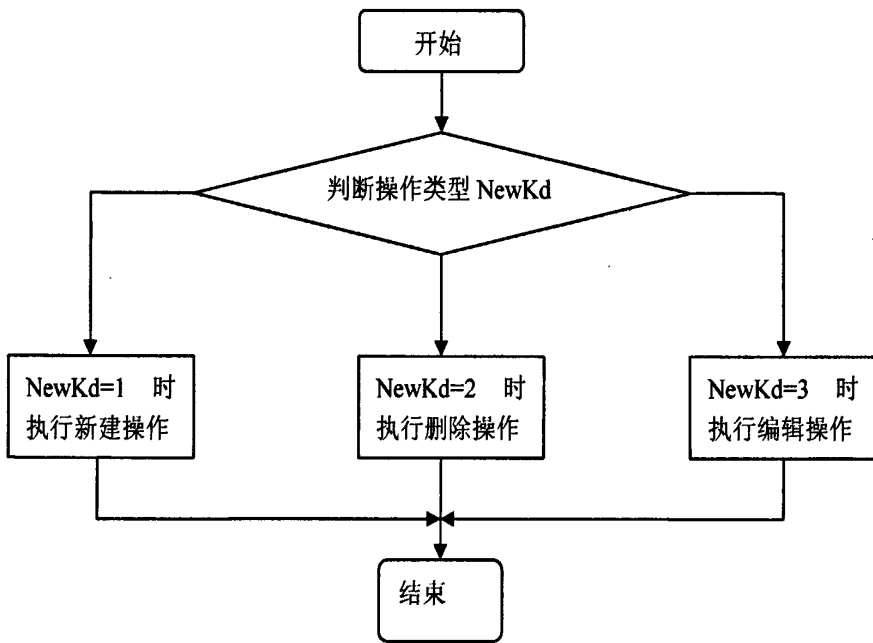


图 6.21 NeWZdinfo 函数流程图
Fig. 6.21 NeWZdinfo function flow chart

监控终端信息管理函数 NeWZdinfo 的建立过程与登陆函数一样。首先，在 Remote Data Module 中添加 ADOCommand 控件，命名为“ADOCmdNeWZd”，其 Connection 属性设置为“ADOCConnection1”。

函数的流程图如下如图 6.21 所示。实现代码见附录 1。

应用服务器端 NeWZdinfo 函数实现控制终端信息数据在数据库中的写入、删除、修改功能。

3.奶站管理函数 NZManage 函数

在 Remote Data Module 中添加 ADOCommand 控件, 命名为 ADOCmdNZManage。然后在应用服务器的类型库中建立线路管理函数 NZManage 函数。这里将 ADOCommand 控件的 Connection 属性设置为 ADOConnection1。然后在 Remote Data Module 中实现该函数。函数的流程图如下如图 6.22 所示。实现代码见附录 1。

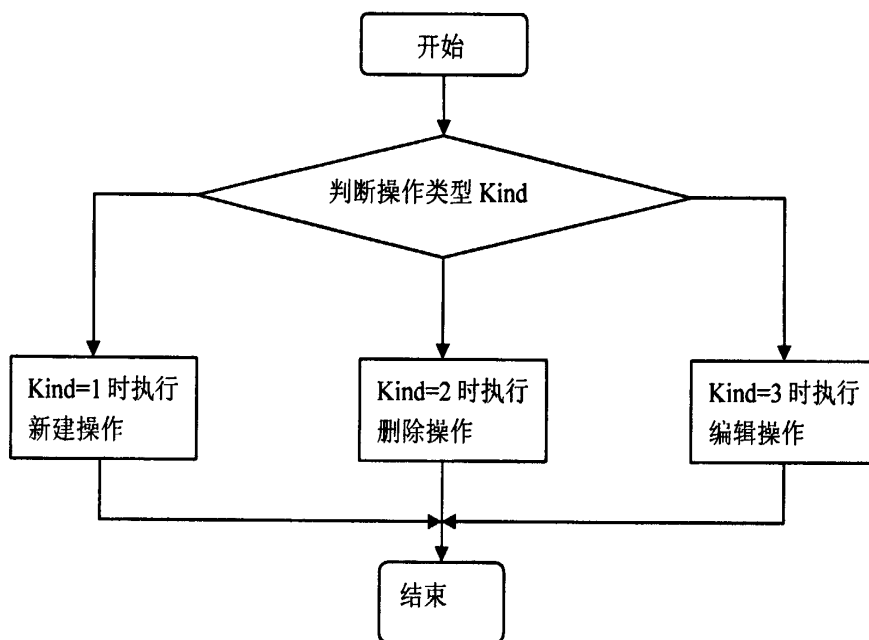


图 6.22 NZManage 函数流程图

Fig. 6.22 NZManage function flow chart

4.服务器端定位请求函数 DINGWEIRequest 函数

客户端定位的请求, 是由管理人员对监控终端主动发起的定位请求, 对于监控终端来说是被动的。主要目的就是让管理人员随时都可以掌握奶罐车的工作状态。这个操作只是发起一次定位请求, 不显示奶罐车当前的位置。DINGWEIRequest 函数流程图如图 6.23 所示。实现代码见附录 1。

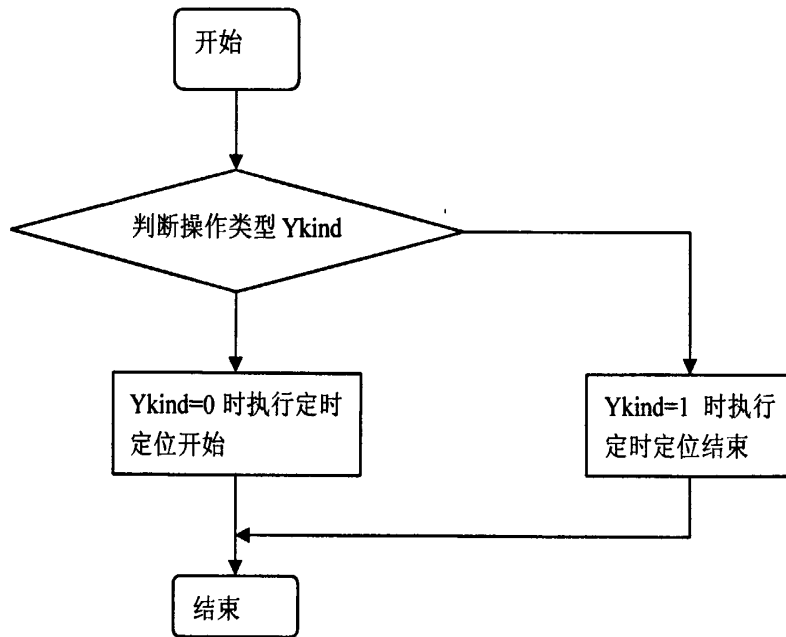


图 6.23 DINGWEIRequest 函数流程图

Fig. 6.23 DINGWEIRequest function flow chart

以上这些函数建立完成后，在客户端数据块中引入该函数的接口，就可以调用了。

6.6 客户端的设计实现

系统客户端承担着向管理人员提供可视化远程监控界面及信息管理功能，包括：电子定位地图显示功能，最基本的地图工具功能；监控终端信息管理和线路信息的管理功能；信息采集点与其对应线路的对比显示功能，通过 MapX 常规技术实现奶罐车定位功能。

客户端要实现的主要功能有：1. 登录操作；2. 线路信息管理；3. 奶站信息管理；4. 监控终端信息管理；5. 远程定位；6. 图像信息管理；7. 地图工具功能；8. 导航图功能；9. 电子地图定位显示功能；10. 图像显示功能。

6.6.1 客户端登录功能的实现

客户端到应用服务器采用 Socket 连接，其优点是：易于安装，可以跨网关布置。Socket 连接方式下，客户端只需要知道应用服务器的 IP 地址及所应用端口就可以直接连接。套接字是支持 TCP/IP 协议的网络通信的基本操作单元，它由通信双方的一组 IP 地址和应用程序的端口号组成。通过 IP 地址可以唯一地确定网络上的一台主机，而通过端口号又可以唯一地确定此主机上的一个应用进程，这样通过套接字就可以在 network 中唯一地确定通信双方的进程，进而可以唯一地确定 network 中的一条链路。

Windows 套接字(Sockets)是存在于通信域中,通信域是为了处理一般的线程通过套接字通信而引进的一种抽象概念。可以将套接字看作不同主机间的进程进行双向通信的端点。套接字应用程序一般仅在同一类的套接字间通信。不过只要底层的通信协议允许,不同类型的套接也可以通信。

开发 C/S 模式应用程序时,可以通过套接字来交换数据包,以完成应用程序之间的通信。C/S 模式使多台计算机有机的结合在一起,分别执行各自的功能,协同完成整个系统的应用,从而达到系统中软、硬件资源最大限度的利用。C/S 模式中首先在客户机和服务器之间定义一套通信协议,并创建一条链接,然后,客户机/服务器再在这条链接上传递数据。当用户需要访问服务器时通过客户机发出请求,服务器接受请求并做出响应,然后执行相应地服务,并把执行结果送回给客户机,由它处理后提交给用户。

为了实现客户端与应用服务器的连接及调用服务器端函数接口的功能,在客户端程序的 Remote Data Module 单元 khd 中放入 TSocketConnection 控件,我们将它命名为 Clcon,如图 6.24 所示.,它是客户端与服务器端进行通信连接的控件,通过我们先期开发的应用服务器在网络中注册的 COM 接口与应用服务器进行连接。然后通过 AppServer 接口引入要调用的服务器函数。

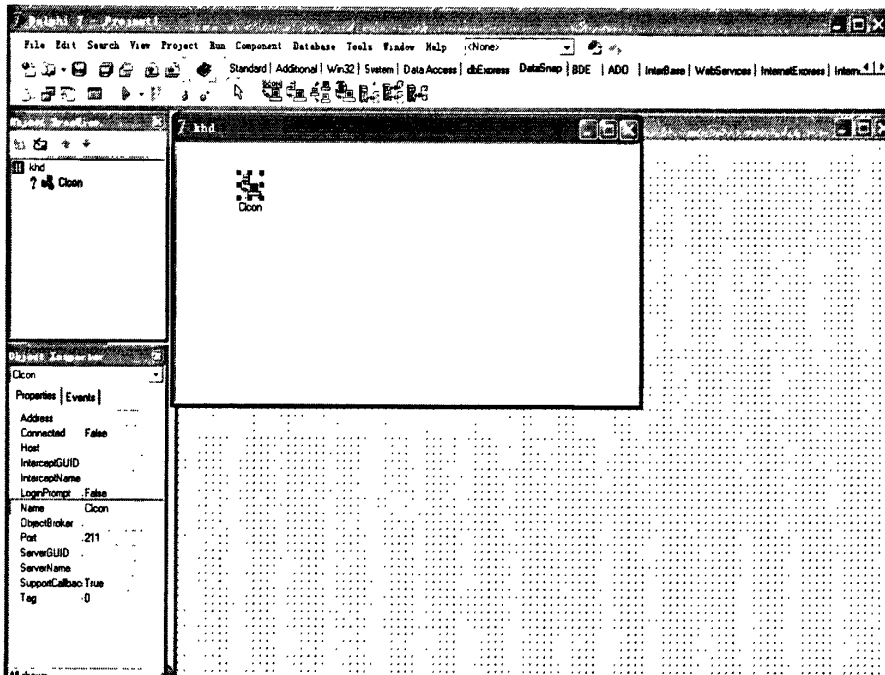


图 6.24 客户端 khd 放入 TSocketConnection 控件

Fig. 6.24 khd admit TsocketConnection activex

Clcon 的属性设置如下:Address 属性设置为应用服务器所在的 IP 地址,Port 属性设置为应用服务器所在的系统端口号,这里设置成 9001,其 ServerName 属性设置为 PUCRtSrvr.Clcon,

就是我们先前建立的自动化服务器运行一次后，在系统中注册的 COM 接口名称。通过该接口的连接，我们就可以调用应用服务器内的系统登录函数，即通过 TSocketConnection 控件在客户端引入 IAppServer 接口，实现了调用服务器的 Login 函数。

管理员登录界面的实现图如图 6.25 所示。

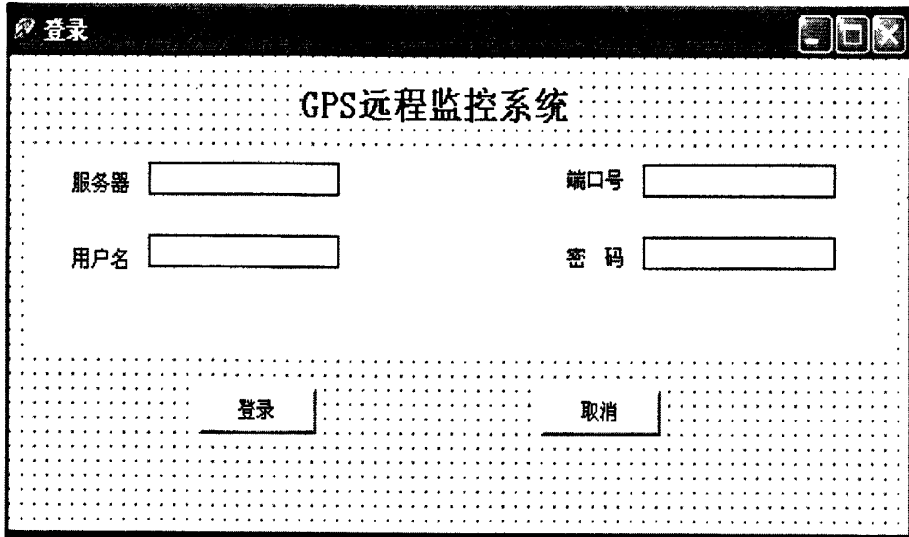


图 6.25 客户端登录界面

Fig.6.25 Client logining interface

管理员只要在服务器文本框中输入应用服务器所在的 IP 地址，在端口号文本框中输入通过 Borland Socket Server 程序设置的应用服务器的监听端口，这里设置监听端口为 9001；然后输入自己的用户名和密码，点击“登录”按钮，就可以进入系统了。

管理员在进入奶源运输安全监控系统之前必须通过登录页面的验证，只有系统的合法用户才能进入监控系统。登录页面的设置保证了监控信息和系统数据库的安全。

6.6.2 客户端主界面设计实现

客户端界面设计如图所示。主界面分为 4 个部分：1、功能菜单区；2、电子地图显示区；3、图像显示区；4、奶站—终端节点树。客户端软件功能分类框图如图 6.26 所示。

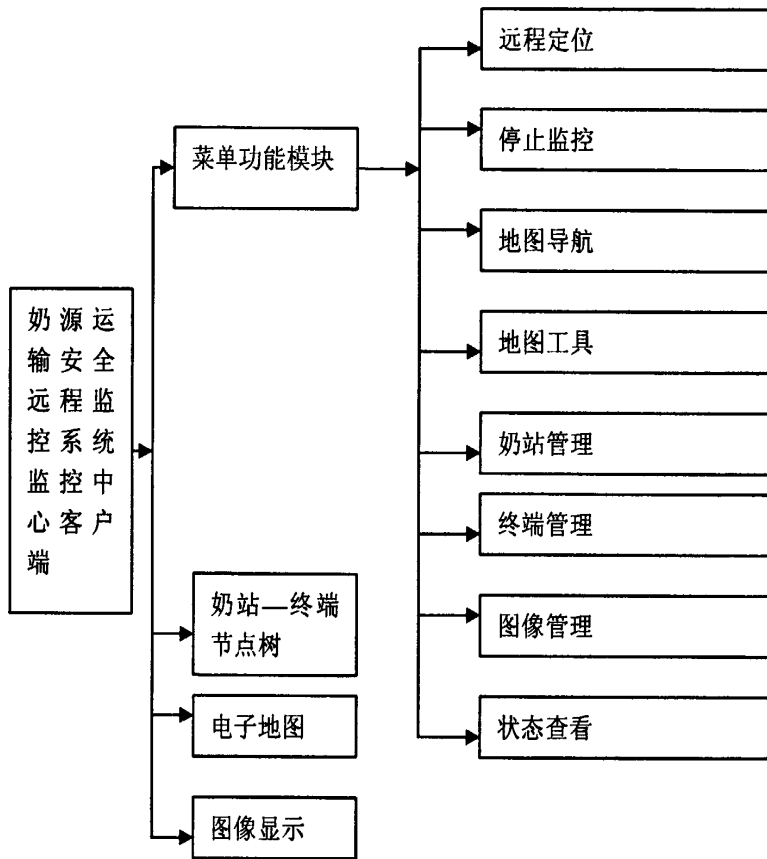


图 6.26 客户端软件功能分类框图

Fig. 6.26 Client software function Classified chart

6.6.3 客户端“终端管理”功能的实现

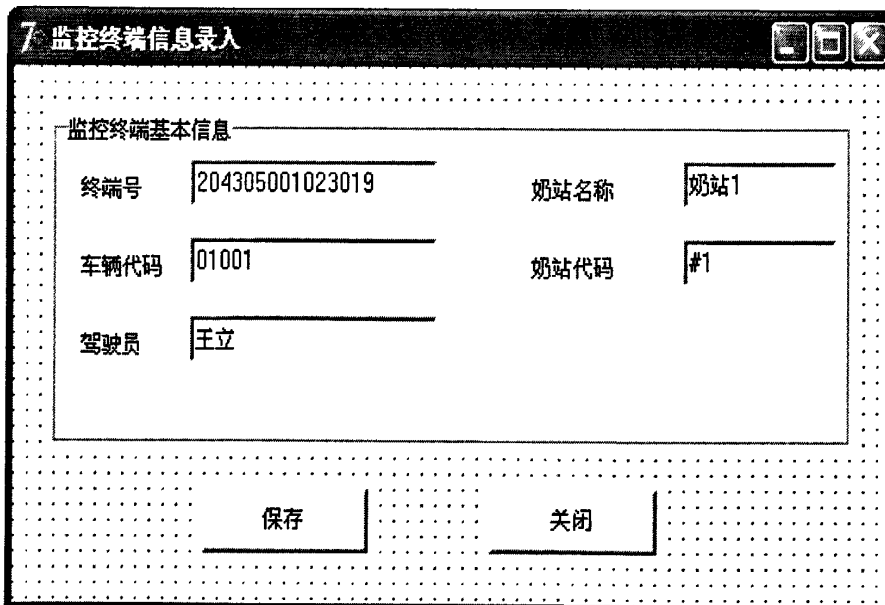


图 6.26 监控终端信息录入界面

Fig. 6.26 Monitor terminal information entering interface

客户端监控终端信息录入界面如图 6.26 所示。

当监控人员点击菜单工具栏中的终端管理菜单项的“新建终端”项时，就会进入这个界面。它的功能是新建一个终端并在数据库 Zdinfo 表中添加的新建监控终端的基本信息，点击“保存”按钮后即可把新添加的终端添加到数据库的 Zdinfo 表中。

其中“终端号”就是奶罐车所安装的定位终端的终端 IMEI 码，车辆代码即是奶罐车的车辆编号。

在客户端的数据模块 Khd 中引入服务器端的函数，这样就可以调用服务器端 NewZdinfo 函数接口，实现定位终端的数据新建了。

在客户端界面的“保存”按钮所对应的任务函数中。这样就在 Zdinfo 表中建立了新的定位终端数据。

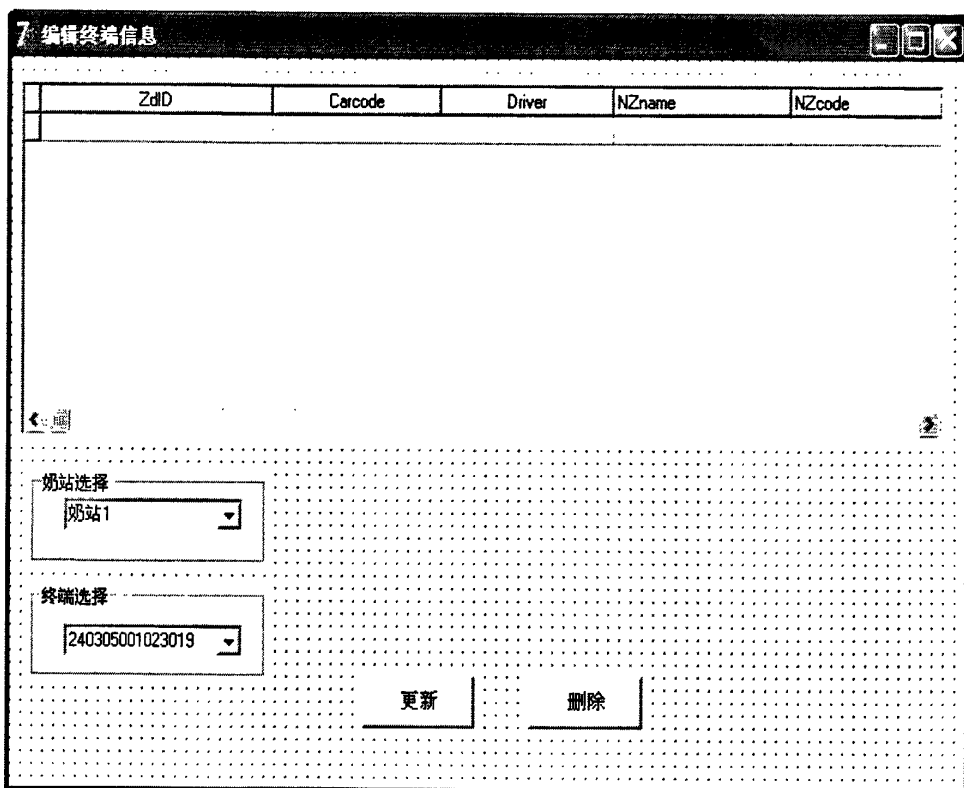


图 6.27 监控终端信息编辑界面

Fig. 6.27 Monitoring terminal information editing interface

客户端定位终端信息编辑界面如图 6.27 所示。

当监控人员点击菜单工具栏中的“终端管理”菜单项的“编辑奶站”项时，就会进入这个界面。它的功能有两个：1、更新数据库 Zdinfo 表中的终端信息，只要在列表框中选中要编辑的终端，并在其最后一行中更改相应的信息，点击“更新”按钮后即可更新的所选的终端信息；2、删除终端，只要在列表框中选中要删除的终端，点击“删除”按钮后即可在数据库 Zdinfo

表中删除的所选终端的信息。

同样点击“更新”“删除”按钮，调用服务器端 NewZdinfo 函数接口，就实现定位终端的更新、删除了。

6.6.4 客户端“奶站管理”功能的实现

奶站信息管理录入界面如图 6.28 所示。

The screenshot shows a software window titled "奶站信息录入" (Milk Station Information Entry). Inside the window, there is a form with a dotted background. The form is titled "奶站基本信息" (Basic Information of Milk Station). It contains four input fields: "奶站名称" (Milk Station Name), "奶站代码" (Milk Station Code), "起 点" (Start Point), and "终 点" (End Point). Below the form, there are two buttons: "保存" (Save) and "关闭" (Close). The window has standard Windows-style window controls (minimize, maximize, close) in the top right corner.

图 6.28 奶站信息管理录入界面

Fig. 6.28 Milking station information entering interface

当监控人员点击菜单工具栏中的“奶站管理”下拉菜单的“新建奶站”项时，就会进入这个界面。它的功能是在窗口中输入新添加的奶站的基本信息，点击“保存”按钮后即可把新添加的奶站添加到数据库的 NZinfo 表中。

其中“奶站名称”、“奶站代码”就是奶罐车行驶线路的名称及代码，“经度”、“纬度”即标明了奶站的地理位置。

编辑奶站信息界面如图 6.29 所示。

当监控人员点击菜单工具栏中的“奶站管理”下拉菜单的“编辑奶站”项时，就会进入这个界面。它的功能有两个：1、更新数据库 NZinfo 表中的终端信息，只要在列表框中选中要编辑的奶站，并在其后一行中更改相应的信息，点击“更新”按钮后即可更新的所选的奶站信息；2、删除终端，只要在列表框中选中要删除的终端，点击“删除”按钮后即可在数据库 NZinfo 表中删除的所选奶站的信息。

服务器端的 NZManage 函数设计完成后，提供外部引用接口，客户端通过接口调用，就可以完成新建，编辑，删除奶站的功能。

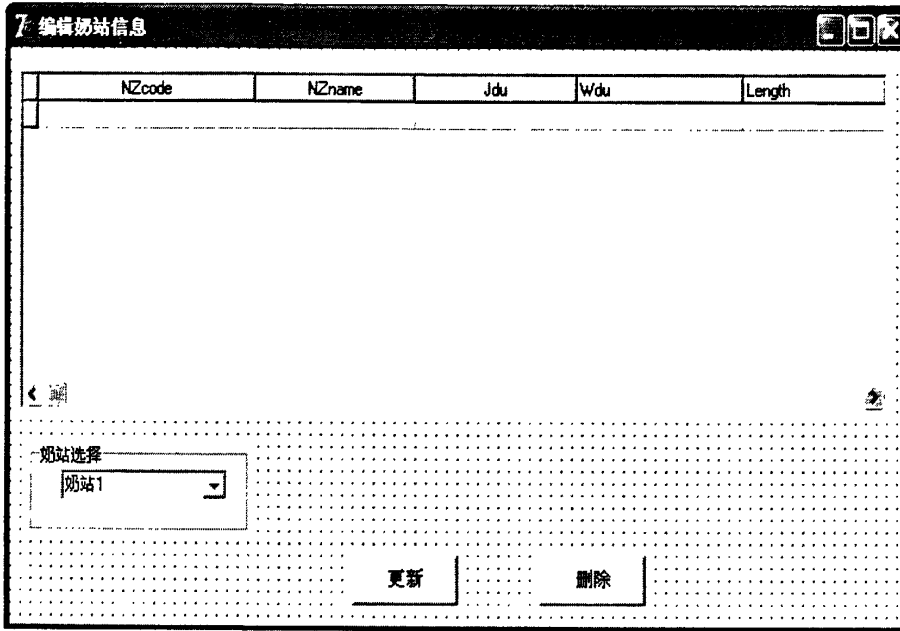


图 6.29 奶站信息编辑界面

Fig. 6.29 Milking station information editing interface

6.6.5 客户端“地图工具”的实现

1. MapX 在 Delphi 中的注册使用

MapX 安装后, 是以 ActiveX 控件的形式与开发环境进行集成的, 这就需要在开发环境中进行注册引入。在 Delphi 中的调用过程是这样的, 首先在 IDE 界面中选择 Component, 在下拉菜单中选择, 如图 6.30 所示。

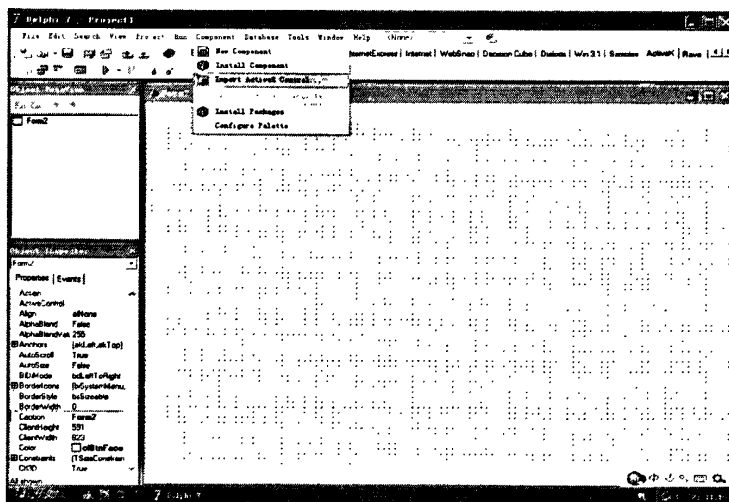


图 6.30 IDE 界面 Component 添加项

Fig. 6.30 IDE interface Component delphi listview

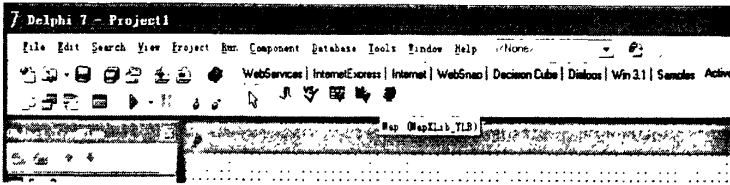


图 6.31 已添加的 MapX 控件
Fig. 6.31 Loaded MapX activex

引入 MapX 组件后，在 Delphi 的 IDE 界面 ActiveX 控件标签，如图 6.31 所示。在 IDE 环境中注册成功后，就可以与 Delphi 的其他控件一样，正常使用了，如图 6.32 所示。

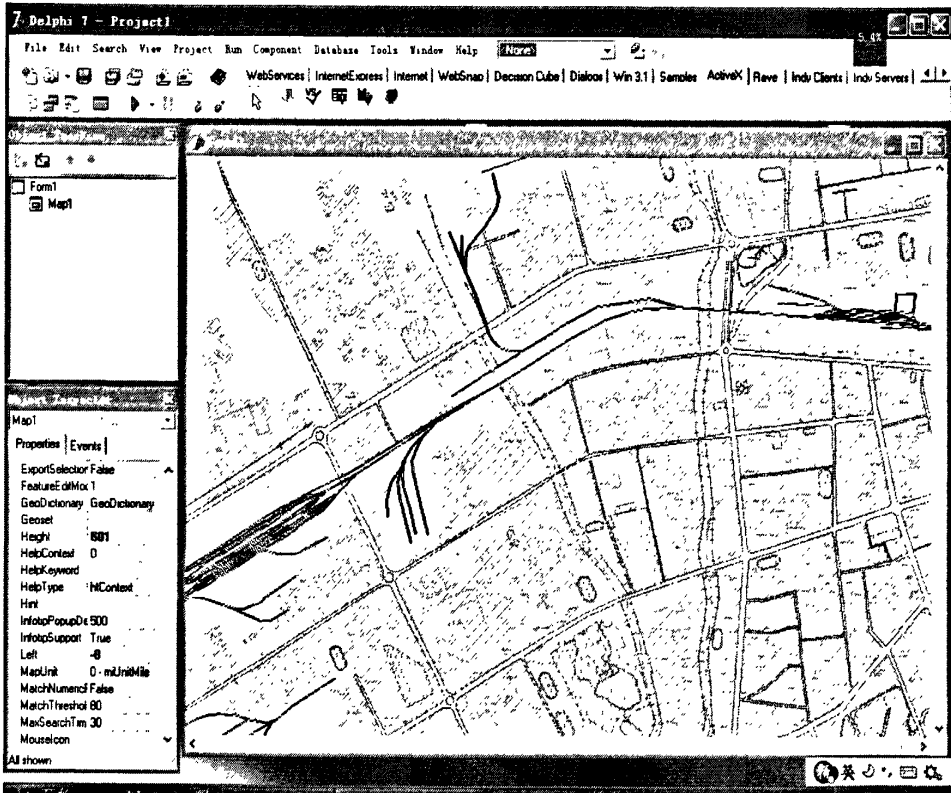


图 6.32 MapX 控件
Fig. 6.32 MapX activex

2. 地图操作的实现

地图的放大、缩小、平移、测距、图层管理、图元选择等操作是通过特定地图工具实现的，这些是地图操作最基本的功能。

首先在界面上放置可以浮动的 TToolBar 控件，在其上面放置代表地图工具的按钮，实现结果如图 6.33 所示。



图 6.33 地图工具
Fig. 6.33 Earthmap tools

- 地图平移功能

当鼠标单击地图工具面板中“平移”按钮时，触发相应任务，该过程实现地图当前工具为地图平移，当按下鼠标左键拖动鼠标时，自动调用 MapX 封装的 API 函数实现地图的平移。

- 地图放大功能

当鼠标单击地图工具面板中“放大”按钮时，触发相应任务，该过程实现地图当前工具为地图放大，当鼠标单击地图时，自动调用 MapX 封装的 API 函数实现地图局部的放大。

- 地图缩小功能

当鼠标单击地图工具面板中“缩小”按钮时，触发相应任务，该过程实现地图当前工具为地图缩小，当鼠标单击地图时，自动调用 MapX 封装的 API 函数实现地图视野的扩大。

- 图元选择功能

用图元选择功能可以实现显示图元的信息或者进行编辑修改功能。要实现图元选择，用鼠标单击地图工具面板中“图元选择”按钮，触发相应任务，该过程实现地图当前工具为图元选择，然后在地图的 OnSelectionChanged 事件中实现图元选择功能。

- 距离测算功能

当鼠标单击地图工具面板中“测距”按钮时，触发相应任务，该过程实现测距功能，使当前地图状态为测距状态，可以测出任意两点间的距离。

- 图层管理功能

当鼠标单击地图工具面板中“图层管理”按钮时，触发相应任务，该过程实现地图图层查看功能，管理员可以查看地图的任意一个图层。

- 矩形区域选择功能

鼠标单击地图工具面板中“矩形区域选择”按钮，触发相应任务，该过程实现地图当前工具为地图矩形区域选择，当鼠标单击地图并拖动时，自动调用 MapX 封装的 API 函数，实现地图以鼠标按下点为起点，拖动距离为对角线的矩形区域内图元选择功能。

6.6.6 客户端“地图导航”功能的实现

所谓地图导航，就是在小地图上小范围地动实现大地图上大范围的随动，这样更有利于地图浏览的效率。

首先在主界面上再放入一个小的 MapX 地图控件，命名为 MapMaster1，如图 6.34 所示。

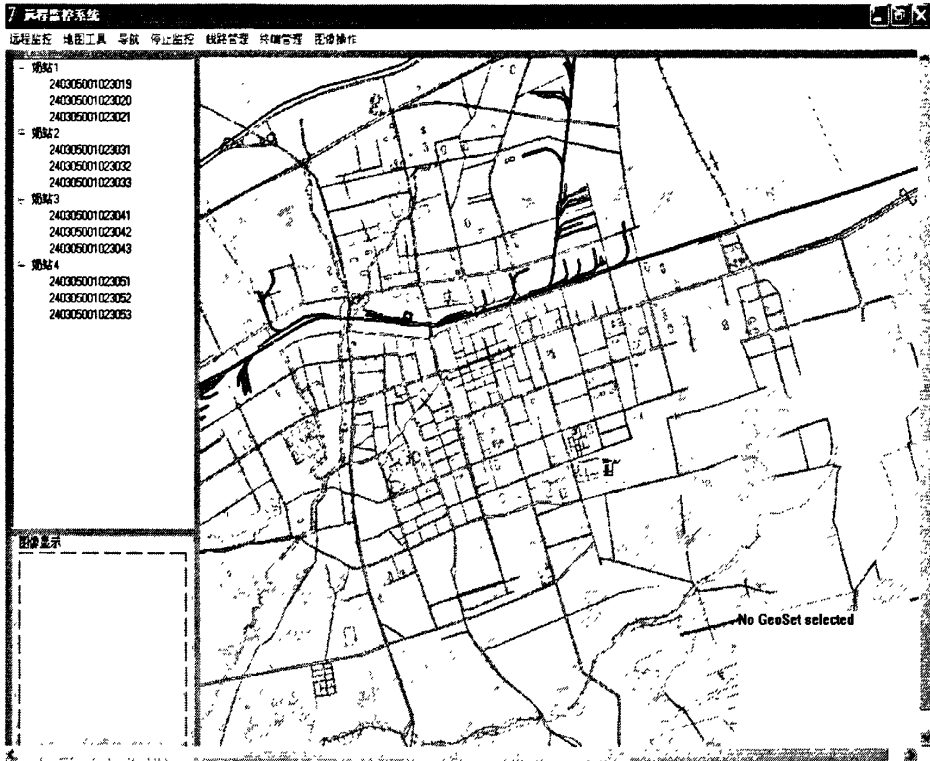


图 6.34 地图导航功能设计界面

Fig. 6.34 Maps navigation designing interface

要实现地图导航，需实现 MapMaster1 以下事件。OnMapInitalized 事件，实现导航地图的初始化，该过程实现导航地图上的矩形工具和鼠标箭头工具，用于区域移动选择；OnMouseDown 事件，在按下鼠标左键时响应；OnMouseMove 事件，在拖动鼠标时响应；OnMouseUp 事件，在鼠标按键弹起时响应；该过程判断如果是左键弹起，则地图就不响应鼠标拖动，结束导航。在此同时，还需要同时实现主地图的相应事件 OnMouseMove 事件，使两个地图之间进行双向互动。

6.6.7 客户端界面节点树的设计实现

客户端数据组织的设计方案是：

在监控界面中，按照“奶站”-“监控终端号”的格式进行数据组织，将数据库中的奶站及其所属监控终端号的数据以节点的方式组织，在 TTreeView 控件 TV1 中实现显示；通过单击鼠标右键弹出快捷菜单实现对树形结构中不同监控终端的操作。

节点树的构造过程，实际上就是一个数据检索和加载的过程，这个过程的实现需要在客户端的数据模块中放置 CDSDataSet，它是客户端的数据集合控件，其通过 TSocketCorulection 控件 Clcon 连接到应用服务器中的 DataSetProvider。DataSetProvider 作为数据集的提供者，它的作用是为客户端提供数据，其 DataSet 属性设置为 ADODataset，ADODataset 是服务器端

的数据集合控件，ADODataset 通过 ADOConnection 对数据库进行访问。

首先，需要从数据库服务器中获得数据，为此，在客户端程序数据模块 Data Module 中加入 CDSGetLineKd、CDSGetLine、CDSGetPatrolman 3 个 CDSDataSet 控件。它们共同通过应用服务器端的 DataSetProvider1 的桥接获得 ADODataset1 的数据集合，ADODataset1 通过 ADOConnection1 对数据库进行访问。

节点树的显示效果如图 6.35 所示。

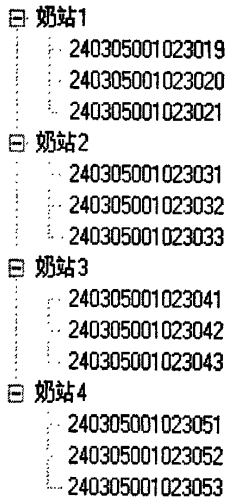


图 6.35 “奶站—监控终端”节点树显示图

Fig. 6.35 “Milking station-- monitor terminal”treeview display image

针对节点树的组织形式，采取单击鼠标右键弹出快捷菜单的操作方式，对树形结构的每一个终端进行操作。为实现这一功能，需要借助 TPopupMenu 控件。对应于监控终端的操作的设计如图 6.36 所示。

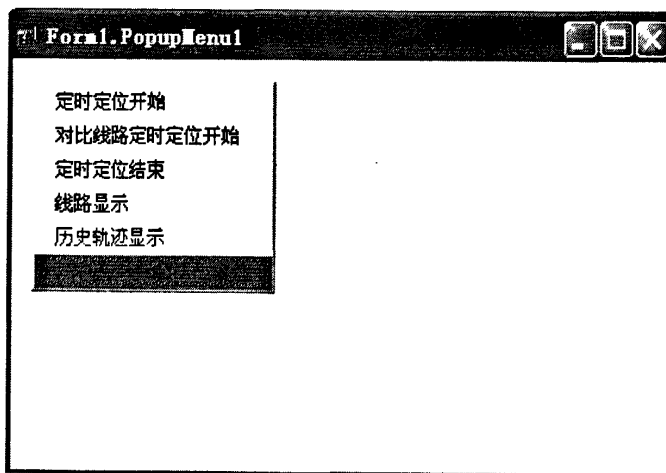


图 6.36 对应于监控终端的操作菜单设计图

Fig. 6.36 Options designing picture for monitor terminal

单个终端历史轨迹显示实现图如图 6.37 所示。

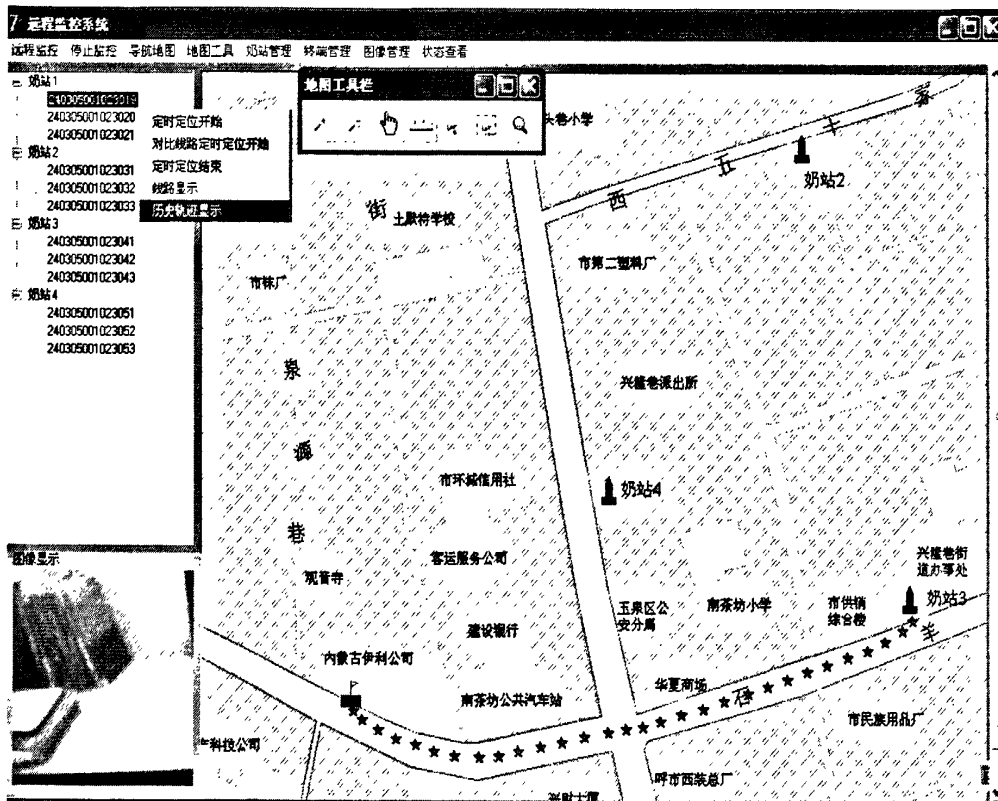


图 6.37 历史轨迹功能实现图

Fig. 6.37 History navigation function implementation diagram

6.6.8 客户端“远程监控”功能的设计实现

远程监控功能是奶源运输安全监控系统的核心功能。它实现了对监控终端的选择性定位，不仅实现了单终端定位，也可以对多个终端同时定位，提供了“定时定位”和“对比线路定时定位”两种电子地图定位显示模式。“远程监控”功能界面图如图 6.38 所示。

首先在界面的显示过程中检索奶站名称，然后在客户端数据模块中加入 ClientDataSet 控件，用于检索奶站信息数据，在奶站选择下拉列表中选择某个奶站时，动态生成该奶站所有的终端列表。获得奶站终端号是在 ComboBox1Change 的过程中调用服务器端函数实现的，这样就完成了远程定位界面的建立。

进入远程定位模块界面后，管理人员需要先选择想要定位的终端所在的线路，此操作在奶站选择下拉列表中完成。选择好线路后，软件会自动生成此奶站所有的终端列表。然后在终端列表中选择想要定位的终端，可以选择一个，也可以同时选择多个。在客户端应用程序中可以完成对监控终端的定位操作，包括定时定位开始、立即定位、定时定位结束，这些请求反馈到应用服务器，对其中的定位队列信息进行动态刷新。

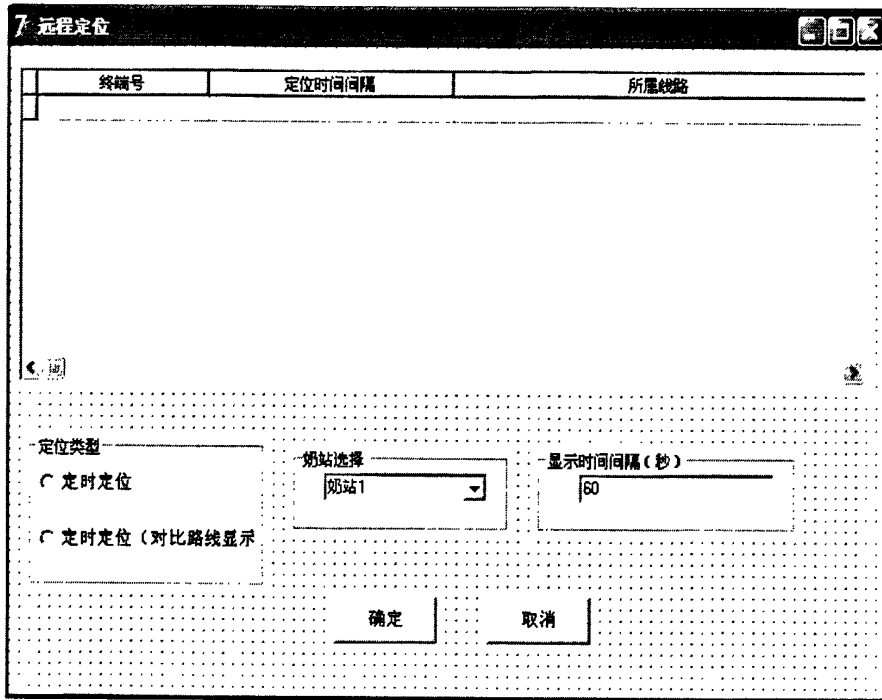


图 6.38 “远程监控”功能界面图

Fig. 6.38 “Remote monitoring” function interface

对于“奶站—监控终端”节点树中针对单个终端的定时定位操作，或是远程定位模块中人员选择后的“确定”事件，都在应用服务器端 DINGWEIRequest 函数接口中实现。

6.6.9 客户端电子地图显示功能的实现

定位显示就是根据条件检索数据库，获取监控终端当前位置的经纬度数据，实时显示在电子地图上，从而直观地观察奶罐车的运输情况。

监控终端的位置数据保存在数据库服务器中，通过检索 Zdstate 表，取得当前监控终端的经纬度数据在地图上显示，实现定位。

在服务器端已经建立了响应客户端位置数据请求函数。

在 Respondrequest 函数中，实现将定位终端号码按照与服务器约定的格式组合，发送位置数据请求，获取定位终端的位置数据，在电子地图中进行定位显示。函数代码见附录 1。该函数根据获得的位置数据在地图上生成相应的点图元，实现地理定位。

为了进行定时监控，还需要定时控件 Timer，在其定时过程中调用该函数，实现动态刷新。

针对“远程监控”功能模块中的多人同时定时定位显示功能，在奶站选择后，在 Listview 中选择的监控终端号码，点击“确定”按钮，启动主界面的 TmOrient 定时事件，完成对所选监控终端的定位显示。

这样，就实现了多人同时定位显示，实现图如图 6.39 所示。

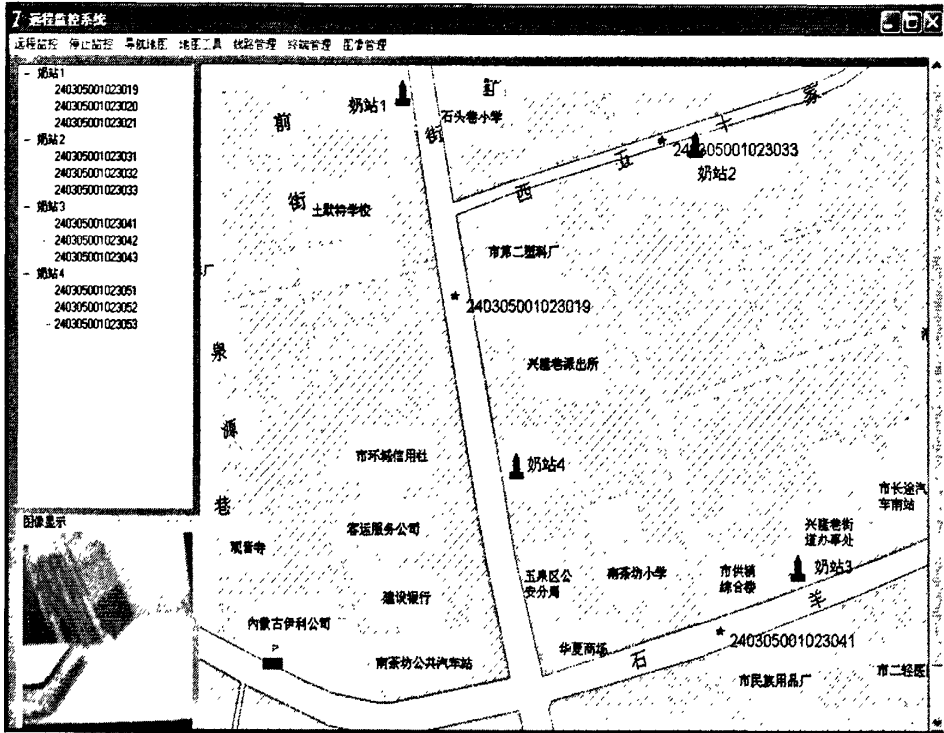


图 6.39 多人同时定位的实现图

Fig. 6.39 Multiplayer simultaneously allocation implementation diagram

6.6.10 客户端图像显示及管理功能的实现

1. 图像显示功能实现

(1) 图像数据格式分析^{[5][29]}

在远程监测终端是通过 C328 摄像头来完成图像的采集，采集的图像是 JPEG 格式，经过传输系统，在监测中心收到的图像数据是十六进制的字符串。因此，要想将收到的数据恢复成 JPEG 图像形式，必须了解图像采集和发送时图像数据的格式。

C328 摄像头采集的图像是 JPEG 格式的。JPEG 格式的图像是以“FFD8”为开始标识，以“FFD9”为结束标识，而一幅图像是以“包”来组成的，每包 512Byte。表 6.2 说明了接收到数据的组成：

表 6.2

Tab.6.2

命令应答	AA 0E 0D 09 00 00 AA 0E 04 0D 00 00
数据长度	AA 0A 01 ~~~(~~~是 C328 返回图像大小)
图像数据	01 DD FA 01 CE D8.....XX XX FA B1 CC A9

命令应答和数据长度对于图像恢复来说是没有用的信息，在恢复的过程中将这些数据删掉。而图像数据又是由一包一包的数据组成，每一包数据的结构如表 6.3 所示：

表 6.3 图像数据包的结构

Tab.6.3 Image data packet structure

ID	Data Size	Image Data	Verify Code
(2Bytes)	(2Bytes)	(Packet Size-6Bytes)	(2Bytes)

ID: 数据包的编号, 从零开始。

Data Size: 数据包中图像数据的大小。

Image Data: 该包中的图像数据。

Verify Code: 数据包的校验码。

经过对图像数据包的分析, 对于图像恢复来说只有 Image Data 中的数据是有用的息, 是组成图像的数据, 将其他的信息都删除就完成了图像的恢复。图像恢复以后以恢复的时间为图像名, 并将图像名存储在相应监控终端的数据表 Zdimage 中。

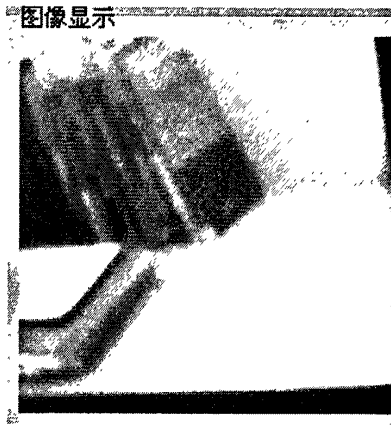


图 6.40 图像显示的实现

Fig. 6.40 Picture display

(2) 图像显示编码

现在客户端主界面上放置 Image 控件, 因为 Image 控件不是数据控制组件, 因此需要编程绑定到数据库的 Zdimage 的图像字段上, 显示当前终端中图像状态表中存储的图像。实现代码如下:

```

Procedure TForm.ADODataSet1AfterScroll(DataSet:TDataSet);
begin
    if j=nil then j:=TJpegImage.Creat;
    if ADODataSet1.Fieldbyname('TP').Value=""then
        begin Image1.Picture.Graphic:=nil;exit;end;
    j.Assign(ADODataSet1.Fieldbyname('TP'));
    Image1.Picture.Graphic:=j;
end;
    
```

end;

2. 图像数据管理功能

图像数据管理功能界面如图 6.41 所示。

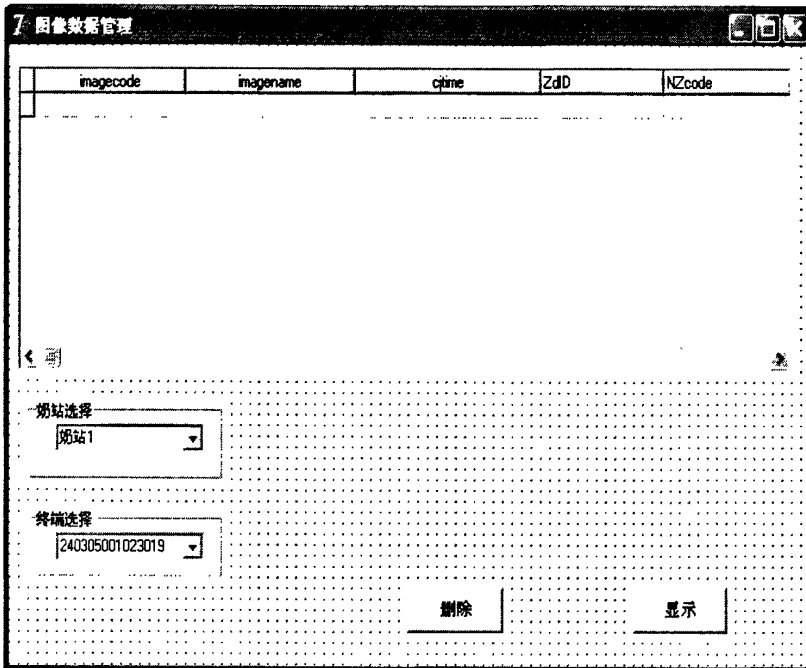


图 6.41 图像数据管理功能界面

Fig. 6.41 Image data manage function interface

当监控人员点击菜单工具栏中的“图像管理”项时，就会进入这个界面。它的功能有两个：

- 1、显示数据库 Imageinfo 表中的终端信息，只要在列表框中选中要显示的图像，点击“显示”按钮后即可在主界面左下角图像显示区中显示的所选的图像；
- 2、删除图像，只要在列表框中选中要删除的图像，点击“删除”按钮后即可在数据库 Imageinfo 表中删除的所选的图像。

服务器端的 ImageManage 函数设计完成后，提供外部引用接口，客户端通过接口调用，就可以完成图像的删除和显示功能。

6.6.11 连接状态查看功能

有时管理人员需要查看监控中心系统是否与 msever 保持连接状态，这时只需点击菜单项“查看状态”，即可在在连接状态查看功能界面中，点击“获取状态”按钮调用相应的接口函数查看对应终端的连接状态。连接状态查看功能如图 6.42 所示。

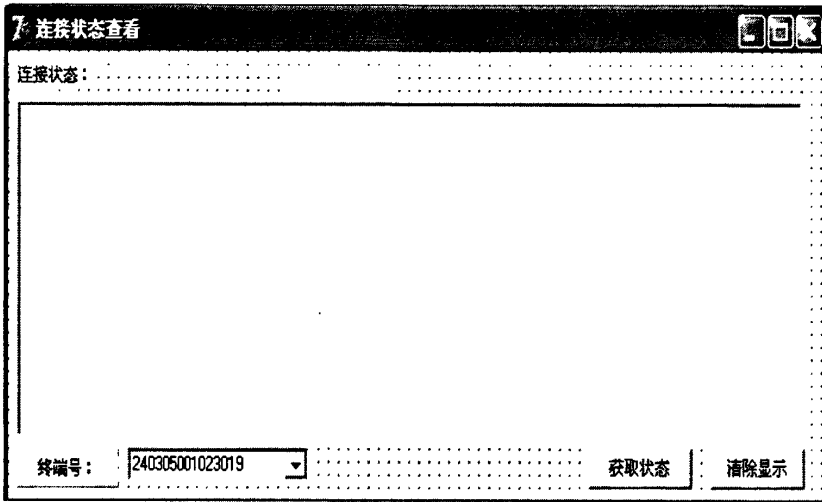


图 6.42 连接状态查看功能界面

Fig. 6.42 Connect status view function interface

6.6.12 Borland Socket Server 程序配置方法

在客户端与服务器连接之前，必须开启 Borland Socket Server，并且设置连接端口号，这里设置端口号为 9001，可以根据需要修改。Borland Socket Server 程序为 Delphi 的安装路径下 Bin 子目录的 scktsrvr.exe 文件。

Borland Socket Server 程序配置界面如图 6.43 所示。

在 Borland Socket Server 程序配置界面，最主要的配置项就是 Port，即服务器的监听端口号，客户端将通过这个端口号与服务器连接，获得服务器注册的 COM 服务，通过接口调用服务器内部的函数，完成业务功能。

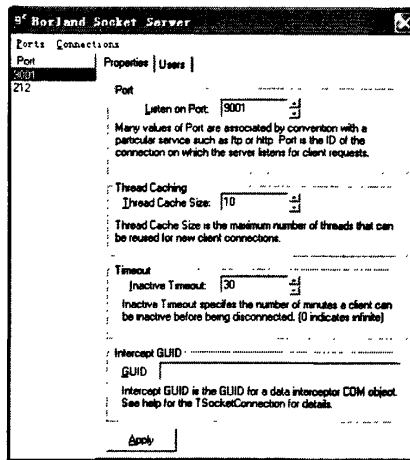


图 6.43 Borland

Fig. 6.43 Borland

第七章 总结与展望

本文在参考国内已有的远程监控系统监控中心开发方案的基础上,结合本课题的实际应用环境,针对奶源运输安全监控设计了一套 C/S 模式的远程监控系统。

一年来,通过对软件各功能模块的编程制作,基本上实现了课题的设计构想,论文对监控中心数据库的建立、监控终端采集数据的导入、应用服务器中部分重要函数的构建、客户端各部分的设计工作作了重点探讨,主要工作及创新点有:

- 1、采用基于 MIDAS 技术的三层体系结构开发模式构建了监控中心软件;
- 2、根据本系统的应用目的及监控终端采集的数据,详细设计了监控中心数据库;
- 3、利用 Object Pascal 语言编写了应用服务器函数,实现了应用服务器对客户端的各项业务服务以及数据请求服务进行封装,以函数接口调用的方式,向客户端提供业务处理;
- 4、利用面向对象的 Delphi 开发环境,建立了客户端可视化监控界面,实现了对监控终端信息和奶站信息的管理;
- 5、应用具有强大地图分析功能的 MapX 控件技术,实现了最基本的地图操作工具,奶罐车电子地图定位功能;
- 6、通过对 C328 摄像头的采集图像数据格式的分析,完成接收图像的重现。

奶源运输安全远程监控系统是集成通信技术、计算机技术和电子技术的综合系统,涉及的技术比较多。本文所设计开发的奶源运输安全远程监控系统,要开发一个成熟,实用的监控系统还有许多工作要做。这主要包括:

- 1.更加完善的无线通讯。做为实时监控的无线通讯方式 GPRS 仍有许多不足,包括速度不够,以及在部分地区缺乏信号。随着 3G 的发展,在远距离无线通讯上的这些问题将得到突破,为开发提供更好的方式。

- 2.更加美观友好的界面。在项目中,采用了不太美观的用户交互界面。

- 3.更强的应用功能。在项目中,只实现了实时定位,多人同时定位,历史轨迹显示,奶站信息管理,终端信息管理,地图导航,图像管理和显示等功能。实时视频采集显示是未来技术的发展方向。

- 4.更加优化的程序结构。系统中对应用函数的编程由于时间紧,个别函数并没有调试成功,有待继续调试,调试成功的函数,其结构也有待继续优化,已达到程序结构的合理化。

致 谢

论文是在我的导师李全虎老师的悉心指导下完成的，导师渊博宽广的专业知识、严谨认真的治学态度和谦虚质朴的待人风格使我受益匪浅，是我终生学习的楷模。一年来，李老师对我的论文从选题、研究、实验直到论文的最后完成都进行了精心的指导，在此，我向他致以深深地敬意和表示衷心的感谢。

在论文的完成期间，我还得到了刘荣林、云鹏、邸晓峰等同学的帮助和支持，他们在论文的完成过程中提供了许多宝贵的资料和有用的建议，在此向他们表示深深的谢意。

此外本论文的顺利完成，还离不开电子信息工程学院的各位老师和同学的关心与帮助，在此我表示衷心的感谢。

最后，向多年求学过程中，在物质和精神上无私支持我、关怀我的父母致以深深的谢意！

参考文献

- [1]张培仁等. 自动控制技术和应用: 监控网络设计[M]. 合肥: 中国科学技术大学出版社, 2001
- [2] <http://baike.baidu.com>
- [3]孟丽, 吕英杰. GPRS 系统简介. 邮电设计技术第 7 期[J], 2001 年 7 月
- [4]韦惠民, 李国民, 暴宇. 移动通信技术[M]. 北京: 人民邮电出版社, 2006.10
- [5]王瑞. 基于 GPRS 的草原生态远程监测系统监测中心的设计与实现[D]. 内蒙古: 内蒙古大学, 硕士学位论文, 2009 年 5 月 1 日
- [6]孙少陵. GPRS 技术特点及应用[D]. 电信技术 2002 年 3 月
- [7]Animesh Chakravarthy, Kyung Yeol Song, Eric Feron. A GPS-based slowdown warningsystem for automotive safety.[C]IEEE Intelligent Vehicles Symposium, 2004
- [8]B.W.Parkins and J.J.Spilker Jr., Eds., Global Positioning System: Theory and Applications. Washington, DC: Amer. Inst. Aeronautics Astronautics, 1996, vol. II.
- [9]徐绍铨, 张华海等. GPS 测量原理及应用[D]. 武汉大学出版社(武汉), 1998.
- [10]川王忠等. GPS 相位测量中的几个重要问题[D]. 导航, Vol.12, No.4, 2000
- [11]赵健康. 利用 GPS 载波相位测量快速性确定动态载体姿态及精度分析[D]. 长沙: 国防科技大学学报, Vol 21, No.4, 1999
- [12]GPS 理论与应用(第 1 卷). 西安导航技术研究所, 1999.6
- [13]GPS 理论与应用(第 2 卷). 西安导航技术研究所, 1999.10
- [14]李翠. 车载 GPS 道路测量与修正系统的研究与开发. 西安: 长安大学, 硕士学位论文 2006 年 4 月 21 日
- [15]王惠南. GPS 导航原理与应用[M]. 北京: 科学出版社, 2003.8
- [16]李连营, 李清泉, 李汉武. 基于 MapX 的 GIS 应用开发[M]. 武汉: 武汉大学出版社, 2003.6
- [17]杜巧玲, 吴秀芹, 张森. Mapinfo7 中文版 入门与提高[M]. 北京: 清华大学出版社, 2005.10
- [18]张增平. 奶源运输安全远程监控系统的设计与实现[M]. 内蒙古: 内蒙古大学, 硕士学位论文, 2010 年 5 月 24 日
- [19]侯纪业. 基于 GPS 的遥测跟踪技术研究实现[D]. 北京: 电子科技大学, 硕士学位论文, 2006 年 3 月 14 日

- [20]柏宝华. 基于 MapX 技术的地理信息系统开发实例[M]. 北京: 清华大学出版社, 2009 年 9 月
- [21]孙毅中等. 城市规划管理信息系统[M]. 北京: 科学出版社, 2004
- [22]杨盛泉, 刘白林. Delphi 2007 程序设计教程[M]. 北京: 清华大学出版社, 2010.9
- [23]田民格, 卢昌荆. Delphi 程序设计[M]. 北京: 清华大学出版社, 2007.9
- [24]吴文虎. Delphi 程序设计教程[M]. 北京: 中国铁道出版社, 2004.11
- [25] Gary Cornell, Troy Strain. 周关边译. Delphi 程序员必读[M]. 北京: 科学出版社, 1995
- [26]赛奎春, 陈紫红等. Delphi 数据库开发实例解析[M]. 北京: 机械工业出版社, 2006
- [27]王业龙, 范朝辉 李育龙. Delphi 软件工程案例精解[M]. 北京: 电子工业出版社, 2005.1
- [28] mServer 接口手册
- [29] C328 User Manual. 康大科技有限公司. 2004 年 8 月 19 日
- [30] Doug Adomatis. The commercialization of exchanging GPS data[J]. 2003 Tumer Endeavors, 2003.
- [31] Hinter J.C., 1996, GIS and Remote Sensing Integration for Environmental Applications, Int. J. Geographical Information Systems, 10(7)
- [32]牛立军. 基于 Mapinfo 的 WebGIS 图形系统应用研究[D]. 交通运输学院, 硕士学位论文, 2008 年 12 月 12 日
- [33]牛立军. 基于 WebGIS 的勘探信息系统的设计与实现[D]. 交通运输学院, 硕士学位论文, 2009 年 6 月 17 日
- [34]黄杏元 马劲松. 地理信息系统概论[M]. 北京: 高等教育出版社, 2008
- [35]刘明德, 林杰斌. 地理信息系统 GIS 理论与实务[M]. 北京: 清华大学出版社, 2006
- [36]崔铁军. 地理空间数据库原理[M]. 北京: 科学出版社, 2007
- [37]王占全, 赵思斯, 徐慧. 地理信息系统 (GIS) 开发工程案例精选[M]. 北京: 北京邮电出版社, 2005
- [38]刘学军, 徐鹏. 交通地理信息系统[M]. 北京: 科学出版社, 2006
- [39]刘光. 地理信息系统二次开发教程. 组件篇[M]. 北京: 清华大学出版社, 2003
- [40]成春望. 监控系统中基于 GPRS 的无线数据传输系统的研究与实现[D]. 北京: 北京邮电大学 硕士学位论文 2006 年 3 月 1 日
- [41]施新刚. Delphi 实效编程百例[M]. 北京: 人民邮电出版社, 2004.8
- [42]萧秋水, 文娟. Windows 网络编程之 Delphi 篇[M]. 北京: 清华大学出版社, 2001.5

- [43]张岭, 宋坤, 梁冰. Delphi 程序开发范例宝典[M]. 北京: 人民邮电出版社, 2006
- [44]段兴. Delphi7 实用程序设计 100 例[M]. 北京: 人民邮电出版社, 2003
- [45]Laganathan, G. GPS and GIS technology trends[J]. Electronics & Communication Engineering Journal, 2002, 4(6):292~294
- [46]邓丽. 车载电子中 GPS_GPRS 导航系统的研究[D]. 成都: 电子科技大学, 硕士学位论文, 2006 年 5 月 16 日
- [47]陶诗涌. 基于 DSP 的 GPS 卫星接收机的研究与开发[D]. 成都: 电子科技大学, 硕士学位论文, 2006 年 1 月 2 日

附录 1 应用服务器部分函数接口代码

1、数据连接初始化代码:

```

procedure Tycjkxt.RDMChushihua(Sender: TObject);
var
  path:string;
  ini:TIniFile;
begin
  AppPath:= ExtractFilePath(Application.ExeName);
  path:= AppPath + 'PUcRtsrvr.ini';
  ini := TIniFile.Create(path);
  ADOConnectionString:=ini.ReadString('ADOConnection', 'ConnectionString', '192.168.1.102');
  ADOConnection1.Close;
  try
    ADOConnection1.Open;                                //连接数据库服务器
  except
    ws_ErrorMsg(application.handle,'连接失败');        //连接失败时显示'连接失败'
    UcRtSrvrMainForm.ActionSetADOConnect.Execute;
  end;
  exit;
end;
ini.Free;
end;

```

```

procedure Tycjkxt.ADOConnection1BeforeConnect(Sender: TObject);
begin
  ADOConnection1.ConnectionString :=ADOConnectionString;
end;

```

2、用户登录验证接口函数代码:

```

function Tycjkxt.Login(const Ymanagername, YScode: WideString;
  Key: Integer; out Rt: Integer; out Mesg: WideString): WordBool;
var
  i: integer;
  inputmima,realmima:string;
  convertmima,codemima:string;
  Logintime: TDatetime;          //登陆时间
begin
  result:=false;
  codemima:="";
  with ADOQryLogin do
  begin

```

```

close;
sql.Clear;
sql.Add('select * from Managerinfo where Managename=:aa');
ADOQryLogin.Parameters.ParamByName('aa').Value:= Ymanagename;
Open;
first;
if eof then
begin
    Mesg:='该用户不存在';
    Rt:=-1;
end else
begin
    realpswd:= fieldbyname('mima').asString;
    if Ymima=realmiam then
    begin
        Mesg:='验证通过';
        result:=true;
    end else
    begin
        PSMesg:='密码错误';
    end;
end;
end;
UcRtSvrMainForm.LVClient.Repaint;
end;

```

3、刷新数据库中终端号定位状态过程代码:

```

procedure Tycjkxt.RefrZdstate(YJdu:string;YWdu:integer);
begin
    with ADOCmdRefrZdstate do
    begin
        commandtext:='update Zdstate set Jdu=:aa where Wdu=:bb';
        Parameters.ParamByName('aa').Value:= YJdu;
        Parameters.ParamByName('bb').Value:= YWdu;
        execute;
    end;
end;

```

4、实现客户端定位请求接口函数代码

```

function Tycjkxt.DINGWEIRequest (const YZdID: WideString; YKind: Integer): WordBool;
var
    i,j,k:integer;
    Mesg:string;
    DWcode:string;
    DWcodes:string;

```



```

end;
result:=true;
end;
end;
end;

```

5、奶站信息管理

```

function Tycjkxt.NZManage(const YNZCode, YNZName, YJdu, YWdu, Ylength: WideString;
                           NZKind, Kind: Integer): WordBool;

```

```

begin
result:=false;
if NZKind =1 then
begin
try
with ADOCmdNZManage do
begin
commandtext:='delete from point where NZcode=:aa ';
Parameters.ParamByName('aa').Value:= YNZCode;
execute;
end;
except
exit;
end;
end;

try
with ADOCmdNZManage do
begin
case Kind of
1: begin
commandtext:='insert into line (NZCode, NZName, Jdu, Wdu, length, '+
'values (:aa,:bb,:cc,:dd,:ee) ';
Parameters.ParamByName('aa').Value:= YNZCode;
Parameters.ParamByName('bb').Value:= YNZ Name;
Parameters.ParamByName('cc').Value:= Jdu;
Parameters.ParamByName('dd').Value:= Wdu;
Parameters.ParamByName('ee').Value:= length;
end;
2: begin
commandtext:='delete from line where NZcode=:aa ';
Parameters.ParamByName('aa').Value:= YNZCode;
end;
3: begin
commandtext:='update line set NZname=:aa where NZcode=:bb ';
Parameters.ParamByName('bb').Value:= YNZCode;

```

```

Parameters.ParamByName('aa').Value:= YNZName;
end;
end;
Execute;
result:=true;
end;
except
exit;
end;
end;

```

6、在数据库新建监控终端并更新定时定位队列接口函数代码:

```

function Tycjkxt.NewZdinfo (const YZdID, YCarcode, YDriver, YNZname, YNZcode: WideString;
                            NewKd: Integer; out Mesg: WideString): WordBool;
var
  ij:integer;
begin
  result:=false;
  case NewKd of
    1: begin
      try
        with ADOCmd NewZdinfo do
          begin
            commandtext:='insert into patrolman (ZdID, Carcode, Driver, NZname,NZcode: WideString,
                                                '+values (:aa,:bb,:cc,:dd,:ee) ');
            Parameters.ParamByName('aa').Value:= YZdID;
            Parameters.ParamByName('bb').Value:= YCarcode;
            Parameters.ParamByName('cc').Value:= YDriver;
            Parameters.ParamByName('dd').Value:= YNZname;
            Parameters.ParamByName('ee').Value:= YNZcode;
            execute;
          end;
        except
          on E:EOleException do
            begin
              case E.ErrorCode of
                AdoKeyViol: begin
                  Mesg:='该终端号已存在';
                end;
              end;
            end;
          exit;
        end;
      else
        begin
          Mesg:='新建未成功,请重新输入';
        end;
      end;
    end;
  end;
end;

```



```

        exit;
    end;
end;
end;
i:=length(A_Line);
setlength(A_Line,i+1);
A_Line [i].GtHandle:= YZdID;
A_Line [i].GtInterv:=trunc(strtfloat(ZdIntval));
A_Line [i].Gtstatus:=0;
end;
2: begin
    try
        with ADOCmdNewZdinfo do
            begin
                commandtext:='delete from Zdinfo where ZdID=:aa ';
                Parameters.ParamByName('aa').Value:= YZdID;
                execute;
            end;
            with ADOCmdNewZdinfo do
                begin
                    commandtext:='update point set ZdID="" where ZdID=:aa ';
                    Parameters.ParamByName('aa').Value:= YZdID;
                    execute;
                end;
            except
                Mesg:='删除失败,请重新操作';
                exit;
            end;
        for j:=0 to length(A_Line)-1 do
            begin
                if A_Line [j].GtHandle= YZdID then
                    begin
                        A_Line [j].GtHandle:="";
                        A_Line [j].Gtstatus:=0;
                        break;
                    end;
                end;
            end;
        end;
    3: begin
        try
            with ADOCmdNewZdinfo do
                begin
                    commandtext:='update Zdinfo set ZdID '+
                        'ZdID =:aa,Carcode =:bb,sex=:dd,Driver =:ee, '+
                        'NZcode=:ff,NZname =:gg, '+
                        'where ZdID =:cc ';
                end;
            end;
        end;
    end;
end;

```

```

Parameters.ParamByName('aa').Value:= YZdID;
Parameters.ParamByName('bb').Value:= YCarcode;
Parameters.ParamByName('cc').Value:= YDriver;
Parameters.ParamByName('dd').Value:= YNZname;
Parameters.ParamByName('ee').Value:= YNZcode;
execute;
end;
with ADOCmdNewZdinfo do
begin
commandtext:='update point set ZdID=:aa where ZdID=:bb ';
Parameters.ParamByName('aa').Value:= YNZname;
Parameters.ParamByName('bb').Value:= YNZcode;
execute;
end;
except
Msg:='更新失败,请重新操作';
exit;
end;
for j:=0 to length(A_Line)-1 do
begin
if A_Line [j].GtHandle= YZdID then
begin
A_Line [j].GtHandle:= YZdID;
A_Line [j].GtInterv:=trunc(strtfloat(ZdIntval));
break;
end;
end;
end;
result:=true;
end;

```

7、mserver 连接程序及终端采集数据倒入数据库程序

```

Procedure lianjie ();
var
cPort,mPort,cMode:Longint;
cIP:string;
begin
cMode:=1;
mPort:=strtoint(Form2.Edit1.text);
if BitBtn1.Caption='连接 mServer' then
begin
cIP:=Form2.Edit2.Text;
cPort:=strtoint(Form2.Edit1.text);
dc_hwnd:=dcc_init(cMode,mPort,cIP,cPort,MODE_NONBLOCK);

```

```

If dc_hwnd > 0 Then
    begin
        Label3.Caption:='已连接';
        BitBtn1.Caption:='断开连接';
    end
Else
    BitBtn1.Caption:='连接失败';
end
else
    if BitBtn1.Caption='断开连接' then
        begin
            dcc_close(dc_hwnd);
            dc_hwnd:=0;
            Label3.Caption:='未连接';
            BitBtn1.Caption:='连接 mServer';
        end;
end;
end;

```

```

procedure getstate(); //获取 DTU 状态
var
    msgsend:dc_msg;
    mm:Array[0..15] of Char;
    SIMEI:string;
    i:integer;
begin
    SIMEI:= ''+ Edit4.Text;
    StrPCopy(mm,SIMEI);

    for i:=1 to 15 do
        msgsend.ImeiDTU[i]:= mm[i];

    msgsend.msg_len := 0;
    msgsend.msg_type := 3;
    if dc_hwnd>0 then
        dcc_msg_send(dc_hwnd,@msgsend);
end;

```

```

procedure storedata ();
var
    retreceive:integer;
    msgreceive:dc_msg;
    i:integer;
    s1,s2,s3,sdata,data,sdt,Status:string;
begin

```

```

msgreceive.msg_type :=9;
retreive := -1;
if dc_hwnd>0 then
    retreive := dcc_msg_receive(dc_hwnd,@msgreceive);

if retreive>0 then
begin
    status := '终端状态: ';
    s1:="";//表示 IMEI
    for i:=1 to 15 do
        s1:=s1+msgreceive.ImeiDTU[i];

    s2:="";//表示别名
    for i:=1 to 15 do
        s2:=s2+msgreceive.NamedDTU[i];

    sdt:= '('+DateTimeToStr(Now)+)';

case msgreceive.msg_type of
0: begin
    s3:="";
    sdata:="";
    s3 := '收到'+ Inttostr(msgreceive.msg_len) +'字节数据';
    for i:=1 to msgreceive.msg_len do
        sdata:=sdata+IntToHex(msgreceive.msg_body[i],2)+' ';

    if Length(sdata)=20 then
begin//1
        sdata:=MidStr(sdata,5,10);

        if midstr(sdata,0,2)='0A' then
begin //2
            sdata:=midstr(sdata,3,8);
            if midstr(sdata,0,2)='0A' then
begin//3
                sdata:=midstr(sdata,3,6);
                if midstr(sdata,0,2)='0A' then
                    sdata:=midstr(sdata,3,4);
                end; //end 3
            end; //end 2

        data:="";
        for i:=1 to (Length(sdata) div 2) do
            data:=data+midstr(sdata,i*2,1);
        end; //end 1

```

```
    Memo1.Lines.Add(s3 + sdt);
    Memo1.Lines.Add('IMEI:'+ trim(s1)+ ', Name: '+ trim(s2)+ ':');
    Memo1.Lines.Add(sdata);
end;//case 0 end

1: begin//DTU offline
    Status := Status + '离线';
    Memo1.Lines.Add(Status + sdt);
    Memo1.Lines.Add('IMEI:'+ trim(s1)+ ', Name: '+ trim(s2)+ ':');
end;

2: begin//DTU online
    Status := Status + '在线';
    Memo1.Lines.Add(Status + sdt);
    Memo1.Lines.Add('IMEI:'+ trim(s1)+ ', Name: '+ trim(s2)+ ':');
end;

5: begin
    if msgreceive.msg_body[1]=0 then
        Status := Status+'终端不存在';
    if msgreceive.msg_body[1]=1 then
        Status := Status + '在线';
    if msgreceive.msg_body[1]=2 then
        Status := Status + '离线';

    if msgreceive.msg_body[1] in [0..2] then
        begin
            Memo1.Lines.Add(Status + sdt);
            Memo1.Lines.Add('IMEI:'+ trim(s1)+ ', Name: '+ trim(s2)+ ':');
        end;
    end;
end;
else
end;
end;
end;
```

附录 2 客户端定位显示程序代码

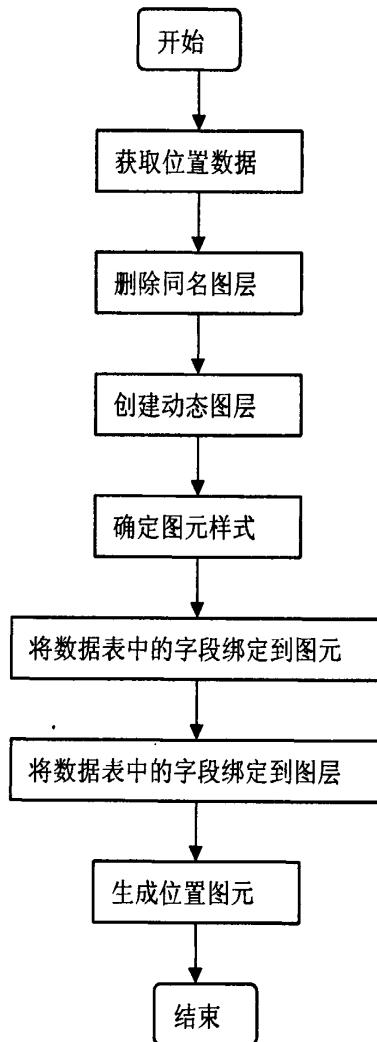


图 1 客户端电子地图定位显示函数流程图

Fig.1 Client side electronic map localization demonstration function flow chart

Procedure Dingweixianshi (var YCode:string;Y Layer:string;Y Path:

string;YMap:TMap;Ykind:integer);

var

cx:CMapXdataset; // MapX 图层数据集类型

ftr:Feature; // MapX 图元类型

flds :fields; // MapX 数据表格的字段类型

info:CMapXlayerinfo; // MapX 图层信息类型

Style :CMapXStyle; // MapX 样式类型

YLayer:Layer; // MapX 图层类型

```

x,y:Double;
YColor:TColor;
MPts :integer;
point:MapXLib_TLB.point;
v1,v2,v3,v4,v5,v6:CMapxRowvalue;
Sqlstr:string;
code:string;
m:string;
i,j:integer;
S:string;
O:integer;
Cur:Feature;
Sym:Feature;
Edkey:string;
s1,s2,s3,s4,s5:string;
is:integer;
p:pointer;
m3:Tmemorystream;
m1:OleVariant;
m2 :Tstringlist;
begin
  ZuHe:=Ykind;
  sqlstr:=PSTIcode+'&'+datetostr(date)+'&';//按照以下格式从数据库中取出数据
  m1:=dm.ZHiding(sqlstr,Ykind,MPts);
  try
    sm := TMemoryStream.Create;
    m3.Clear;
    m2:= TStringlist.Create;
    m2.Clear;
    try
      m3.Position:=0;
      isize:=VarArrayHighBound(m1,1)-VarArrayLowBound(m1,1)+1;
      p := VarArrayLock(m1);
      m3.Write(p^, isize);
      VarArrayUnlock(m1);
      m3.Seek(O,soFromBeginning);
      m2.LoadFromStream(m3);
    except
      exit;
    end;
    if m2.Count>=1 then
      begin
        info:= CoLayerInfo.Create; //创建新图层
        flds := cofields.Create; //创建点、点集合
      end;
    end;
  end;
end;

```

```

point:=CoPoint.Create();//确定点式样
with Y Map.DefaultStyle do
begin
    SymbolType := miSymbolTypeBitmap;
    SymbolBitmapOverrideColor := False;
    SymbolBitmapName := 'CAMP1-32.BMP';//GetStringValue(s, 2);
    SymbolBitmapSize := 16;//StrToInt(GetStringValue(s, 3));
    SymbolBitmapTransparent := true;//GetStringValue(s, 4) = '1';
end;
v1:= CoRowvalue.Create; //将数据库中 Zdstate 数据表的字段数据绑定到图元
v2:= CoRowvalue.Create;
v3:= CoRowvalue.Create;
v4:= CoRowvalue.Create;
v5:= CoRowvalue.Create;
vs:= CoRowvalues.Create;

flds.AddStringField('RvTime',20,true); //将数据库中 Zdstate 数据表的字段信息绑定
到图层
    ,
flds.AddStringField('Rvkind',16,false);
flds.AddStringField('Tlcode',16,false);
flds.AddFloatField('ptlong',false);
flds.AddFloatField('ptlatd',false);

info.Type_:=miLayerInfoTypeNewTable;
info.AddParameter('FileSpec', YPath+"+YLayer"+"+'.tab'); //tab 文件输出路径
info.AddParameter('Name', YLayer);
info.AddParameter('Fields',flds);
info.AddParameter('OverwriteFile','1'); //覆盖已经存在的 tab 文件
YMap.Layers.Add(info,1);
YLayer:=YMap.Layers.Item(1);

cx:=YMap.Datasets.Add(miDataSetLayer,CELayer,Emptyparam,Emptyparam, //完成
数据绑定
    Emptyparam,Emptyparam,emptyparam,Emptyparam);
CELayer.AutoLabel:=true;
CELayer.labelproperties.overlap:=true;
CELayer.labelproperties.duplicate:=true;
CELayer.LabelProperties.Offset:=5;
CELayer.LabelProperties.Position:=miPositionTC;
CELayer.LabelProperties.Style.TextFontColor:=clRed;
CELayer.LabelProperties.Dataset:=cx;
CELayer.LabelProperties.DataField:=ds.Fields.Item('RvTime');

i:=0;

```



```
j:=0;
for i:=0 tom2.Count-1 do
begin
  s:=m2.Strings[i];

  j:=pos('*',s);
  s1:=copy(s,1,j-1);

  s:=copy(s,j+1,length(s)-j);
  j:=pos('*',s);
  s2:=copy(s,1,j-1);

  s:=copy(s,j+1,length(s)-j);
  j:=pos('*',s);
  s2:=copy(s,1,j-1);

  s:=copy(s,j+1,length(s)-j);
  j:=pos('*',s);
  s4:=copy(s,1,j-1);

  s:=copy(s,j+1,length(s)-j);
  j:=pos('*',s);
  s5:=copy(s,1,j-1);

  x:=strtofloat(s4);
  y:=strtofloat(s5);
  point.Set_(x, y);
  YMap.Layers.Item(YLayer).LabelAtPoint(x,y);
  v6.RemoveAll;
  v1.dataset:=cx;
  v1.field:=cx.Fields.Item(1);
  v1.value:=s3;
  v2.dataset:=cx;
  v2.field:=cx.Fields.Item(2);
  v2.value:=s2;
  v3.dataset:=cx;
  v3.field:=cx.Fields.Item(3);
  v3.value:=s1;
  v4.dataset:=cx;
  v4.field:=cx.Fields.Item(4);
  v4.value:=x;
  v5.dataset:=cx;
  v5.field:=cx.Fields.Item(5);
  v5.value:=my;
  v6.add(v1);
```

```
v6.add(v2);
v6.add(v3);
v6.add(v4);
v6.add(v5);
if Ortflag = 0 Then
begin
    ftr:=YMap.FeatureFactory.CreateSymbol(piont,YMap.DefaultStyle);
    Sym:=YMap.Layers.Item(YLayer).AddFeature(ftr,v6);
    Edkey:= fMapSymbol.FeatureKey;
    O:= 1;
end
else
begin
    YLayer.DeleteFeature(Edkey);
    ftr:=PMap.FeatureFactory.CreateSymbol(point,YMap.DefaultStyle);
    Sym:=YMap.Layers.Item(YLayer).AddFeature(ftr,v6);
    Edkey:= Sym.FeatureKey;
end;
FindFeatureByKeyValue(YMap,YLayer,Edkey, [stClearAll, stSelect]);
YMap.CenterX := x;
YMap.CenterY := y;
with MainForm.LVRecv.Items.Add do
begin
    Caption:=s1;
    SubItems.append(copy(s2,length(s2)-7,8));
    SubItems.append('操作成功');
end;
end;
end;
finally
    m3.Free;
    m2.Free;
end;
end;
```