

曲阜师范大学

硕士学位论文

基于GPRS的GPS车载终端

姓名：程兵

申请学位级别：硕士

专业：控制理论与控制工程

指导教师：李洪亮

20090301

摘要

近几年来,随着计算机技术、电子技术的飞速发展,汽车的智能化逐渐成为汽车工业发展的必然要求。随着嵌入式系统的网络化、智能化趋势和半导体工艺的进步,嵌入式处理器的性能大幅提高,其应用领域也越来越广泛。嵌入式技术为汽车电子提供了极佳的解决方案,使开发出高性能、低成本的智能车载终端成为可能。无线数据传输技术的快速发展给相关产业和领域的发展带来了极大的机遇,尤其是对数据采集与传输系统方面的应用。无线传输技术极大的扩展了数据采集与传输系统的应用范围,降低了制造和维护成本,提高了系统的设计与运行效率。

车辆监控技术是方便、快速、正确引导汽车抵达目的地的一种技术,它在提高汽车运行效率、节约能源方面具有重要意义。目前,最新的 GPS 技术、GSM 技术和 GIS 技术在汽车工业领域得到应用。

本文对基于 GPS、GPRS 的车载监控系统进行了研究,并设计和实现了以 MEGA128 作为中央微处理器的车载监控终端,以远程监控中心作为服务平台的综合化系统解决方案。车载终端利用 GPS 模块对目标车辆进行定位,经过微处理器处理后使用 GPRS/GSM 模块通过移动通信网将位置、状态信息传送到远程监控中心。中心服务人员还可依据一定操作流程对车辆的位置信息设置为自主回传和呼叫回传。

最后对所做工作作了总结,对系统今后的研究方向做了展望。该车载监控终端能够胜任车辆定位、跟踪等基本工作要求,且系统功能扩展空间大,易于升级。

关键词: GPS 定位系统,通用分组无线业务(GPRS),MEGA128,车载监控

Abstract

Recently, great advance has been achieved both in computer science and electronic technology, which makes the modern automobiles much more intelligent and much more comfortable. With more and more network technique applied in embedded system and the great improvement in semiconductor manufacturing, the EMPU (Embedded Micro-Processing Unit) has extended its application to the automobile industry. The embedded technique gives the best solution for automobile electronics and makes it possible to produce high performance terminal for cars at low cost. Nowadays, wireless data transition technology has brought great fortune for business especially for the information acquisition system, which could extend the data acquisition range and cut down the maintenance cost.

The technique of vehicle supervision can guide the motor to its destination conveniently, quickly and rightly, which is very important in increasing the transportation efficiency and energy saving. Today, the latest GPS, GSM, GIS techniques have been applied in motor fields.

This paper discusses the Vehicles Managing Navigation System of GSPRS, GPS based on MEGA128, taking MEGA128 as the car of central microprocessor in vehicle monitoring terminal and using long range supervision center as comprehensive system resovement. The vehicle monitoring terminal positions the vehicles through GPS module, delivering the position, appearance information to the long range supervising and controlling center through GPRS, GSM module controlled by microprocessor, and sending out the controlling information analyzed after combine the GIS information the system. Center assistant can setup Calling Response or Self-determination Response for the car according to the operational Process.

At the end of the dissertation, a summary was made. Some possible promotions were also given at the same time. This intelligent vehicle terminal has compact structure and powerful function such as car location and tracking. The system was economical and easily for upgrading.

Key words: GPS (Global Position System), GPRS (General Packet Radio Service), MEGA128, Vehicle terminal

曲阜师范大学博士/硕士学位论文原创性说明

(在□划“√”)

本人郑重声明：此处所提交的博士□ 硕士 论文《基于 GPRS 的 GPS 车载终端》，是本人在导师指导下，在曲阜师范大学攻读博士□ 硕士学位期间独立进行研究工作所取得的成果。论文中除注明部分外不包含他人已经发表或撰写的研究成果。对本文的研究工作做出重要贡献的个人和集体，均已在文中已明确的方式注明。本声明的法律结果将完全由本人承担。

作者签名： 

日期： 2009.6.6

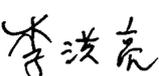
曲阜师范大学博士/硕士学位论文使用授权书

(在□划“√”)

《基于 GPRS 的 GPS 车载终端》系本人在曲阜师范大学攻读博士□ 硕士学位期间，在导师指导下完成的博士□ 硕士学位论文。本论文的研究成果归曲阜师范大学所有，本论文的研究内容不得以其他单位的名义发表。本人完全了解曲阜师范大学关于保存、使用学位论文的规定，同意学校保留并向有关部门送交论文的复印件和电子版本，允许论文被查阅和借阅。本人授权曲阜师范大学，可以采用影印或其他复制手段保存论文，可以公开发表论文的全部或部分内容。

作者签名： 

日期： 2009.6.6

导师签名： 

日期： 2009.6.6

第一章 绪论

1.1 引言

伴随着全球经济一体化,世界经济的迅速发展,交通的紧张不断加剧,世界各国都存在着不同程度的交通拥挤与堵塞现象,交通事故与日俱增,这些都严重影响着社会、经济发展和人民生活。在研究这一问题的过程中智能交通系统(ITS)随之出现,它作为一种解决交通问题的全新方法,在国内外迅速发展起来。^[1]ITS是一个综合性的系统,它的主要功能体现在优化交通控制、调度管理、快速反应和救援等方面。车辆监控系统则是ITS的主要组成部分,也是我国交通运输管理领域需求最为迫切、极具市场潜力和经济效益的应用项目之一。^[2]

现代车辆监控系统是一种将地理信息技术(GIS)、全球卫星定位技术(GPS)和现代通信技术结合起来的智能系统。系统将移动目标的时间、动态位置(经度、纬度)、状态等信息,实时地通过无线通信链路传送至监控中心,而后在具有地理信息查询功能的电子地图上进行移动目标运动轨迹的显示,并对目标的位置、速度、运动方向、车辆状态等用户所需要的参数进行监控和查询。^[3]

1.2 课题研究的意义与背景

2002年到2006年,我国的汽车工业以超过10%的年增长率持续稳定的发展,2006年汽车产量已达728.0万部。汽车工业的发展直接推动着我国车载GPS终端产品的发展,加之各行业和个人对车载信息终端的需求日渐增加等因素的推动,起步较晚的我国车载GPS终端市场已驶入高速发展轨道。2002至2006年间,中国车载GPS销售量以大于90%的年复合增长率迅速上升,2006年达到81.5万台,5年总销售量高达183.2万台。其销售额也以大于100%的年增长率从2002年的1.8亿元人民币冲刺到2006年的41.9亿元人民币,五年累计共达到77.8亿元。随着需求的进一步增长,预计2007至2011年间,我国车载GPS终端销售额还将以超过20%的速度增长,2011年将达到157.6亿元。

车载GPS终端产品市场的繁荣以及行业车辆和个人汽车用户对基于GPS定位功能的深层次需求的产生,催生出车载GPS市场的另一片天地——基于GPS终端的GPS服务的提供。截至2006年底,我国GPS服务提供商超过了千家,入网GPS车辆超过了百万,由服务提供产生的利润也逐年递增,车载GPS服务提供市场逐渐进入佳境。^[4]

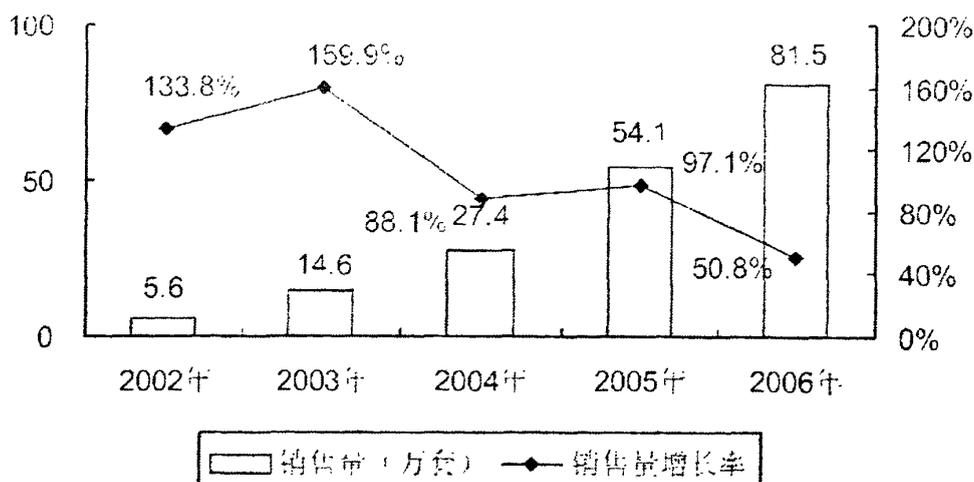


图 1 2002-2006 年中国车载 GPS 市场规模与增长 (按销售量)

1.3 本课题研究的国内外现状

1.3.1 国外车载监控系统研究现状

国外对车辆导航与监控系统的研究比较早,大约开始于 20 世纪 60 年代,最初主要应用于军事上。近十年来车辆导航与监控系统的研究已经达到相当高的水平,开发了各种智能化车载导航与监控电子装置,并被广泛应用于汽车上。就目前情况来看,日本是当今车辆导航定位监控系统发展最为成功的国家之一。在日本的一些豪华汽车上,电子导航设备已经成为标准配置。在日本和欧美,装载有导航设备的汽车数量比例非常高。2005 年日本汽车车载导航设备安装率高达 59%,欧美约在 25%左右。同时,在北美、欧洲、日本等地,汽车导航销售额居各类 GPS 市场之首。车辆导航监控系统可以很方便的使驾驶员得到车辆的具体位置、速度和方向等信息;还可以提供路线查询、路径查询,以使驾驶员可以更快的到达目的地;以及遭遇人身安全问题时,获得紧急求助。

1.3.2 国内车载监控系统研究现状

相对于国外智能交通系统的发展情况,我国智能车辆导航与监控系统应用经历了极其起伏而缓慢的发展道路^[5]。虽然我国早在 20 世纪 70 年代末就已经开始尝试把电子信息技术应用于交通运输和管理,但当时智能交通系统的概念尚未形成。20 世纪 80 年代末 GPS 卫星导航定位技术引入我国,主要应用在大地测量(测绘、勘探)、海上渔业和车辆定位监控等领域。1996~1997 年间国内逐步开始智能交通系统的研究和开发,这一阶段车载终端利用集群网络进行数据通信,是 GPS 车辆监控系统市场的调整和充实时期,主要应用于公安交警车、金融运钞车等特种车辆上。1998~2000 年间是我国车载监控系统的快速发

展时期，技术上得到极大提高，这一阶段采用 GSM 公众网的短信息服务进行数据通信，市场逐渐趋于成熟。2000 年以来，随着 GPRS 无线通信技术的发展，利用 GPRS 网络进行数据传输成为车载监控终端的理想选择方式，随着汽车的迅速增长，国内 GPS 监控终端市场需求潜力巨大。截止到 2006 年，中国汽车数量已经超过 3300 万辆，但车载 GPS 监控终端的安装率却仅为 2%，远远低于日本 59%、韩国 40%、欧美 25% 的水平，专家预计到 2009 年中国汽车 GPS 导航系统终端的销售额将接近 100 亿元，并以每年 200% 到 300% 的速度增长。

目前，国内的车载导航监控系统存在着很多问题。首先从卫星导航的整个应用而言，我国在卫星导航领域基本上都是依靠进口的 GPS 芯片，国内厂商在这些芯片产品上进行二次开发，生产车载终端、自导航和手持定位仪等产品。2007 年国内自主研发的首款应用于车载的卫星导航接收芯片“航芯二号”的设计成功，表明国内已经完全具备 GPS、伽利略、北斗等全球卫星导航接收芯片的开发能力，可以替代进口产品，成为中国集成电路设计领域的重要技术突破。另外，国内 GPS 应用尚未实现普及，并非是技术和硬件问题，而主要在于满足用户需求的服务比较少。国内在智能车载装置的研发还要进一步加大投入，增加系统的功能、提高系统性能。

随着嵌入式微处理器的不断发展和我国自主研发卫星导航定位技术和卫星导航接收芯片的不断完善，相信我国的车载监控终端会朝着功能越来越完善、性能越来越稳定、设计越来越人性化的方向快速发展。

综上所述可以看出，智能交通系统正处于初期向发展期的过渡，远没有到达成熟期。目前 ITS 的理论研究仍不完善，其标准化有待进一步研究确定。另外 ITS 仍以信息技术为主，真正技术含量少。但随着汽车的普及，ITS 仍不失为解决道路交通问题的必由之路。

1.4 选题的背景和意义

我国的智能交通系统的建设现在还处于初级阶段。导航终端作为智能交通系统的神经中枢，很多家公司都在进行研发生产，推出的 GPS 导航终端具有高端和低端两大类。高端产品除导航功能外，其它如无线通讯、DVD、上网等功能都一应俱全。它们的市场销售价格大多比较高，这类产品适合于一些高端客户。相比之下，低端产品只提供作为导航终端的必要的功能，系统复杂度低，成本低，面向的是普通客户以及行业客户，市场前景更为广阔。作为发展方向的高端导航系统在运用上却存在着诸多的不足：其一，在整个系统的售价上偏高，阻碍了它的应用前景；其二，整个系统偏大，在车载空间上存在限制；其三，电子地图的更新频率不够，存在着导航的准确性不足。^[6]

1.5 论文主要研究内容

基于以上课题背景、现状及科学和生产意义的论述，本文各章节中主要讨论的内容如

下:

(1) 绪论部分介绍课题背景、GPS 终端产品在国内外发展现状和趋势,最后提出本选题的意义和研究内容(第一章)。

(2) 提出两种基于 GPRS 的 GPS 车载终端产品的设计目标和思想,一种是其他人一 ARM 为核心的产品,一种是我们自己以 ATmega128 为核心的终端产品,介绍系统实现所涉及的关键技术,然后分析系统总体设计方案,两种方案比较确定方案(第二章)。

(3) 详细介绍终端硬件设计与具体实现(第三章)。

(4) 详细介绍终端软件设计与具体实现(第四章)。

(5) 对本终端已实现的工程应用以及本课题研究内容做出总结和评价,并对基于 GPRS 的 GPS 车载终端产品今后的改进和发展做出进一步的展望(第五章)。

第二章 终端总体设计、关键技术和方案比较

2.1 终端的设计目标和思想

我们的设计目标是构建一个车辆监控系统，通过这个系统可以准确、及时的掌握车辆的位置信息，以便于对车辆进行调度管理。由于车辆地理上的分散性和不确定性，这种监控需要通过无线网络来实现。GPS 技术已经成熟，本终端决定采用它来实现车辆位置的确定。GPRS 作为第二代和第三代无线通讯技术的过渡，相对于其他无线数据传输方式有非常大的优势，因而选用它来实现无线数据传输。车载监控系统主要由四大部分组成：车载监控终端、卫星导航定位系统、无线通信网络、控制中心服务器。系统整体构成如图 2 所示：

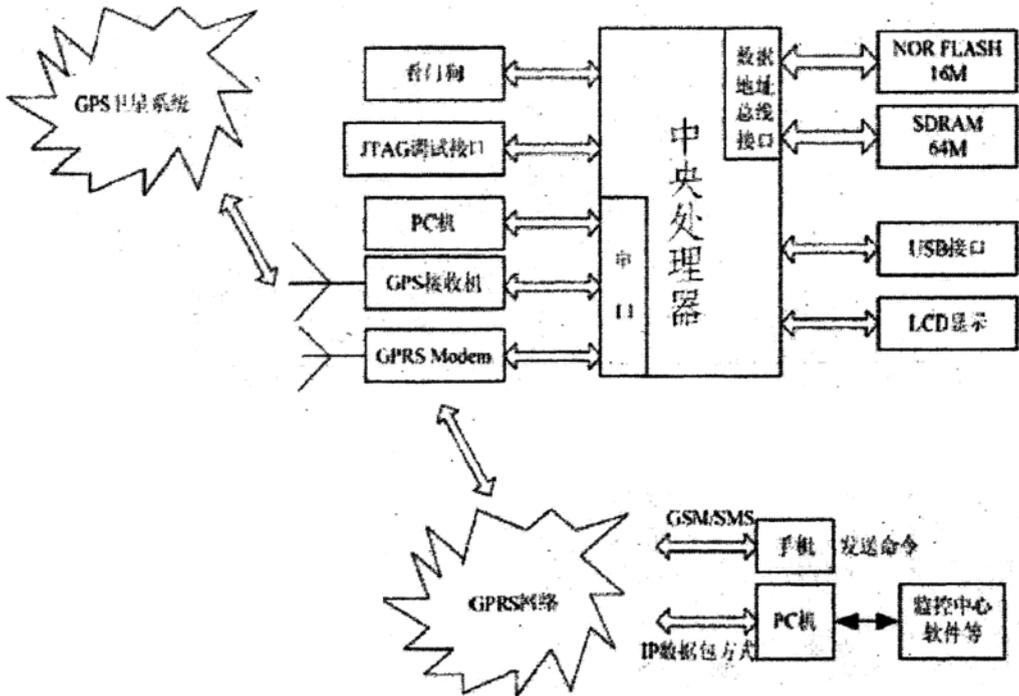


图 2 系统组成结构图

1. 车载监控终端

车载监控终端是整个系统的核心，安装在各个待监控的车辆上。本课题所设计的车载监控终端的主要功能如下：

1) GPS 定位信息的接收。通过串口接收来自 GPS 接收机的定位信息（如经纬度、速度、时间和状态等），并解析出车辆的位置、速度、方向等信息。

2) GPS 定位信息的显示。将解析后的经、纬度、时间等信息显示在服务器上。

3) 按规定格式和设定的时间间隔通过 GPRS 模块将 GPS 定位信息通过无线传输网络发送给监控中心。

4) 通过 GPRS 模块实现同监控中心的通信, 可接收来自监控中心的短信息等功能。

2. 卫星导航定位系统

鉴于 GPS 卫星导航定位技术的成熟, 采用它来实现车辆位置的确定。

3. 无线通信网络

无线通信网络是连接车载终端与监控中心的纽带, 是组成整个系统的关键。考虑到现阶段 GPRS 相对其他无线数据传输方式的巨大优势, 本系统采用 GPRS 无线网络传输数据, 保证系统数据传输的准确性和实时性。

4. 监控中心

监控中心由监控计算机和相应的监控软件组成。监控计算机必须是有固定 IP 地址的主机, 可接入 Internet 网络。监控中心接收车载终端上传的车辆位置、状态等信息, 进行相应数据转换处理后, 实时显示车辆位置、状态等信息。监控中心安装有数据库, 可以同时对这些数据进行记录和存储, 以供事后查询和进行轨迹回放。除接收车载终端上传数据外, 监控中心还具有发送调度命令、控制命令及其它信息的实时发布等功能, 实现对车辆的调度、控制等功能。

2.1.1 以 ARM 为处理器的解决方案

现在市面上有多开发人员都在用 ARM 在做车载终端, 现在介绍一种处理器为 ARM 的终端系统解决方案。车载终端主要分为硬件与软件两大部分, 其结构如图 3 所示:

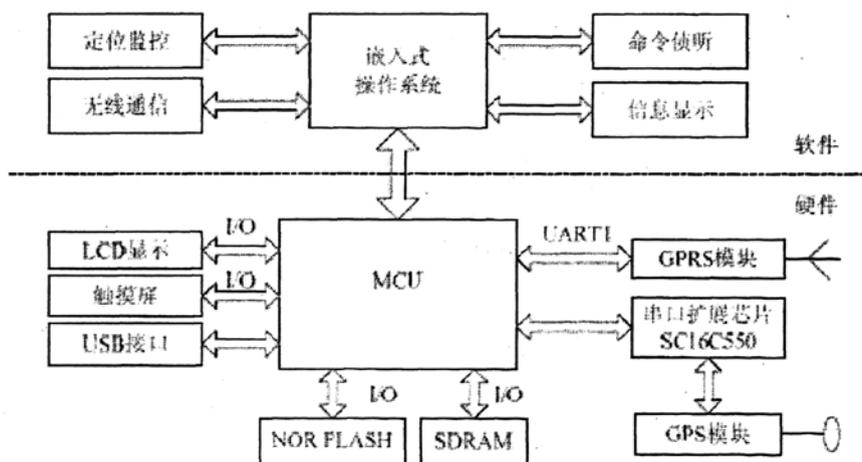


图 3 终端组成结构图

2.1.1.1 硬件部分功能设计

硬件部分主要完成核心电路、外围电路设计,并实现 GPS 模块和 GPRS 模块同 ARM9 处理器的连接。

1) 车载主控机:是车载终端的核心部件,主要负责与 GPS 模块、GPRS 模块的通信,提供人机交互的接口 LCD 显示屏及触摸屏,同时支撑嵌入式操作系统所需的底层硬件资源和相关的硬件部件。选用 ATMEL 公司的 ARM9200 来构建车载监控终端,可以提升终端性能且可扩展空间大。

2) GPS 模块:主要是接收 GPS 定位导航信息,实时提供车辆的位置、速度、时间等信息。设计中 GPS 信号的接收选用了 GPS 接收机 HOLUX GR-87。

3) GPRS 模块:主要负责与监控中心的通信。系统的无线通信模块采用的是西门子的新一代低功耗双频 GSM/GPRS 模块 MC39i,它是很多应用系统中无线高速数据传输的理想解决方案。

4) LCD 显示屏及触摸屏:主要是提供人机交互的平台,通过 LCD 显示屏提供给用户一个简单的友好的界面;用户可通过触摸屏输入控制命令等。

2.1.1.2 软件部分功能设计

选用 Linux 作为终端的操作系统,由其负责底层硬件的初始化和管理及系统的整体调度和控制,并向上层提供应用接口,从成本、资料来源、开发难度等方面上考虑都非常合适。软件部分主要完成基于 ARM 处理器的 Boot Loader、内核、文件系统的移植并在其基础上进行应用程序的编程开发,实现车载监控终端的上述功能。

2.1.1.3 核心技术

1) ARM 微处理器

1991 年 ARM 公司成立于英国剑桥,主要业务是设计 16 位和 32 位微处理器。它本身并不生产和销售芯片,而是采用技术授权的方式,由合作公司生产各具特色的芯片。世界各大半导体生产商从 ARM 公司购买其设计的 ARM 微处理器核,根据各自不同的应用领域,加入适当的外围电路,形成自己的 ARM 微处理器芯片进入市场。ARM (Advanced RISC Machines) 系列处理器是专门针对嵌入式设备设计的,是目前构造嵌入式系统硬件平台的首选。

1. 采用 RISC (精简指令集计算机) 架构的 ARM 微处理器一般具有如下特点^[7]:

- 1) 体积小、低功耗、低成本、高性能;
- 2) 支持 Thumb (16 位) /ARM (32 位) 双指令集,可兼容 8 位/16 位器件;
- 3) 大量使用寄存器,指令执行速度快;
- 4) 大多数数据操作都在寄存器中完成,而不直接对存储器进行操作;

- 5) 寻址方式灵活简单, 执行效率高;
- 6) 采用固定长度的指令格式, 简化了指令译码。

2. ARM 体系结构^[7]:

ARM 微处理器一般有两种工作状态, 并可在两种状态之间切换:

- 1) ARM 状态: 此状态下, 处理器执行 32 位的字对齐的 ARM 指令。
- 2) Thumb 状态: 此状态下, 处理器执行 16 位的、半字对齐的 Thumb 指令。

ARM 体系结构的存储格式:

- 1) 大端格式: 此格式下, 字数据的高字节存储在低地址中, 而字数据的低字节则存放在高地址中。
- 2) 小端格式: 此格式与大端格式相反, 高字节存储在高地址中, 低字节存放在低地址中。

ARM 处理器运行模式, 共支持 6 种运行模式如下:

- 1) 用户模式 (usr): ARM 处理器正常的程序执行状态。
- 2) 快速中断模式 (fig): 用于高速数据传输或通道处理。
- 3) 外部中断模式 (irq): 用于通用的中断处理。
- 4) 管理模式 (svc): 操作系统使用的保护模式。
- 5) 数据访问中止模式 (abt): 当数据或指令预取终止时进入该模式, 可用于虚拟存储及存储保护。
- 6) 系统模式 (sys): 运行具有特权的操作系统任务。

3. ARM 系列

ARM 微处理器目前包括下面几个系列: ARM7, ARM9, ARM9E, ARM10E, ARM11, SecurCore 以及 Intel 的 Xscale 和 StrongARM。除了具有 ARM 体系结构的共同特点以外, 每一个系列的 ARM 微处理器都有各自的性能特点用来满足不同应用领域的需求。其中, ARM7、ARM9 系列适合于较通用的嵌入式产品要求。ARM7 系列微处理器为低功耗的 32 位 RISC 处理器, 有 ARM7TDMI、ARM7TDMI-S、ARM7EJ-S、ARM720T 四种类型, 最适合用于对价位和功耗要求较高的消费类应用。ARM9 系列微处理器在高性能和低功耗方面提供最佳的性能, 包含 ARM920T、ARM922T 和 ARM940T 三种类型。其它几款类型面向更高端的产品应用: ARM9E、ARM10E 系列微处理器为综合处理器, 使用单一的处理器内核提供了微控制器、DSP, Java 应用系统的解决方案; ARM11 增加了面向视频音频数据处理的 SIMD 指令集和浮点运算的支持; SecurCore 系列专门为安全要求较高的应用而设计。

本系统选用的 AT91RM9200 属于 ARM9 系列的 ARM920T。主要特点有以下几点^[7]:

- (1) 5级流水线。如图4所示:

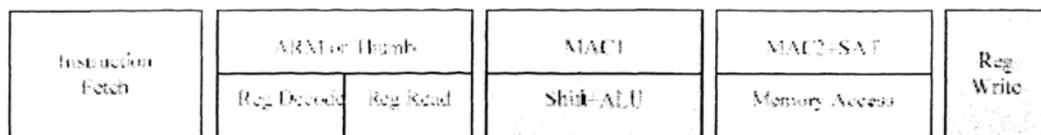


图4 ARM9E的5级流水线

(2) 采用哈佛结构。哈佛结构如图5所示：

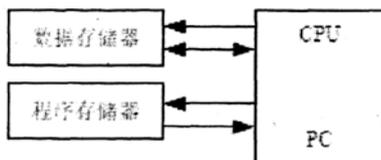


图5 哈佛结构

(3) 高速缓存和写缓存的引入。高速缓存 (Cache) 和写缓存 (Write Buffer) 很好的解决了存储器访问速度远低于处理器速度，有可能成为系统性能瓶颈的问题。它们存储最近常用的代码和数据，以供 CPU 快速存储，如图 6 所示：

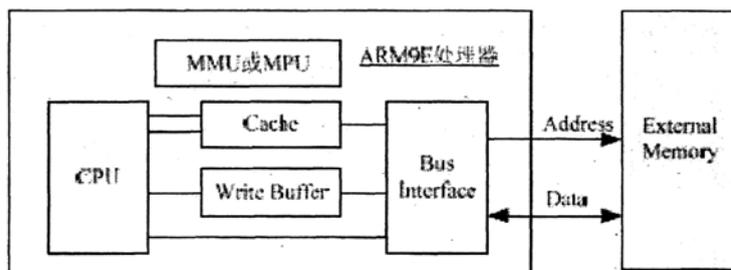


图6 ARM9的高速缓存和读缓存

(4) 支持 MMU。MMU 为内存管理单元，以“页”为单位来处理内存。页的固定大小一般为 4096B 或 8192B。当程序存取一块内存时，会把相应的虚拟地址 (virtual address) 传送给 MMU (MMU 是操作系统为每个正在运行的程序建立并维护的一张进程内存映射 (Process Memory Map) 的表，用来记录程序可以存取的所有内存页和它们的实际位置)。随后 MMU 会在 PMU 中查找此块内存的实际位置即物理地址 (physical address)，物理地址可以在内存中或磁盘上的如何位置。MMU 还可以实现内存保护，增强了系统的稳定性。MMU 的实现过程如图 7 所示：

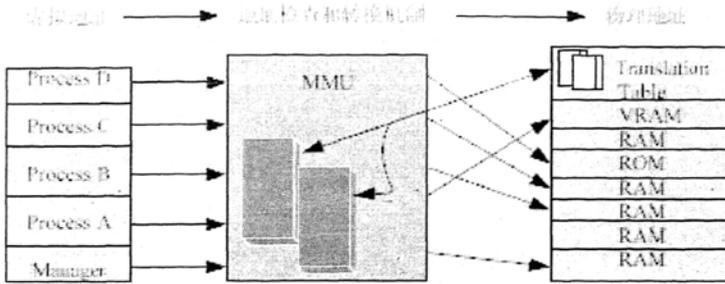


图7 MMU的实现过程

2) Linux 操作系统

目前，嵌入式操作系统在嵌入式应用中用得越来越广泛，尤其在功能复杂、系统庞大的应用中显得越来越重要。与一般的前后台系统相比，采用嵌入式操作系统具有如下优势：

(1) 提高系统的可靠性。操作系统管理的系统，程序出错引起的干扰可能只是破坏若干进程中的一个，并且可以通过系统运行的监控进程对其进行修复。

(2) 提高开发效率，缩短开发周期。在操作系统下开发应用程序使程序的设计和扩展变得容易，通过将应用程序分割成若干独立的任务模块，使应用程序的设计过程大为简化。

(3) 充分发挥 32 位 CPU 的多任务潜力。通过有效的系统服务，使得系统资源得到更好的利用。

表 1 常用嵌入式操作系统特性比较

OS特性	VxWorks	QNX	WinCE	uC/OS-II	Linux
任务调度机制	优先级剥夺式和优先级轮转	优先级FIFO调度，时间片轮转，自适应调度等	剥夺式的时间片轮转，支持优先级继承	优先级剥夺式（可在一定程度上避免优先级翻转）	优先级剥夺式FIFO调度
开发软件包	丰富	一般	丰富	较少	丰富
硬件支持	Power PC、X86、CPU32、SPARC、	X86，并可移植到Power PC、MIPS、	32位X86、Power PC、ARM、	可移植到几乎所有CPU上	X86、ARM、MIPS、ALPHA、SPARC等

	MIPS, 支持 RISC、DSP 技术	ARM等架构	MIPS等		
网络协议栈	有	有	有	无	有
开发费用	很高	较低	较高	很低	很低
源码公开	否	是	否	是	是
实时性	很高	很高	较高	很高	较高
稳定性	极高	高	高	高	高
内核大小	较大	小(不含设备驱动)	很大	小	较小
应用领域	对稳定性、可靠性和实时性要求极高的领域, 如大型电讯设备、航空航天、卫星通信、军事设备等	一般的工业系统、通信等领域	消费电子、手持设备等	适用性较广, 尤其适合小系统开发	广泛应用于网络及中小型系统, 如服务器、一般的工业系统、通信领域等

常见的嵌入式操作系统有: 嵌入式Linux, uCLinux, Windows CE, QNX, VxWorks, OSE, Nucleus, eCos, uC/OS-II 等。我们对常用的几款嵌入式操作系统进行几方面的比较(见表1所示)。

嵌入式Linux (Embedded Linux) 是指对标准Linux经过小型化剪裁处理之后, 能够固化在只有几KB或几MB的存储芯片或单片机中, 适合于特定嵌入式应用场合的专用Linux操作系统。嵌入式Linux的开发和研究是操作系统领域中的一个热点, 在目前已开发成功的嵌入式系统中, 大约有一半使用的是Linux。Linux之所以能够在嵌入式系统市场领域取得如此大的成果, 与其自身优良特性是分不开的^[8-11];

► 开发源码, 软件资源丰富。Linux遵循GPL (GNU通用许可证), 用法律保障了用户免费获得内核源代码的权利。由于嵌入式系统千差万别, 往往需要针对某一具体应用去修改和优化系统, 这样, 能否获得源代码就至关重要。Linux是自由的、免费的操作系统, 它的开放源代码使用户获得了最大的自由度。Linux上的软件资源十分丰富, 几乎每一种通用程序在Linux上都可以找到, 并且每天都在增加。在Linux上往往不需要从头做起, 而是先选择一个类似的自由软件, 进行二次开发。这就大大节省了开发工作量, 缩短了开发时

间，提高了开发效率。

➤ 支持大量的周边硬件设备，驱动丰富。Linux上的驱动已经非常丰富了，支持各种主流硬件设备和最新硬件技术，而且随着Linux的广泛应用，许多芯片厂家也已经开始提供Linux上的驱动，这进一步促进了Linux在各种硬件平台上的应用。

➤ 功能强大的内核，性能高效、稳定、多任务。Linux的内核非常的稳定，它的高效性和稳定性已经在各个领域尤其在网络服务器领域得到了事实的验证，而且Linux内核小巧灵活，易于裁剪。这使得Linux能很适合嵌入式系统的应用。

➤ 完善的网络通信、图形和文件管理机制。Linux自产生之日起就与网络密不可分，网络是Linux的强项。另外，它支持ext2, fat16, fat32, romfs等多种文件操作系统。在图形系统方面，Linux上既有成熟的X Window，也有embedded QT、MiniGUI等嵌入式图形用户界面，还有svgalib、framebuffer等优秀工具，可以适合众多不同的用途。

➤ 大小功能都可定制。Linux继承了Unix的优秀设计思想，内核与用户界面是完全独立的。它非常灵活，各部分的可定制性都很强，能适合多种需求。

➤ 具有优秀的开发工具。嵌入式Linux为开发者提供一套完整的工具链(tool chain)，它利用GNU的gcc做编译器，用gdb、kgdb、xgdb做调试工具，能很方便地实现从操作系统内核到用户态应用软件各个级别的调试。

➤ 支持多种体系结构。Linux能支持X86、ARM、MIPS、POWERPC、ALPHA、SPARC等多种体系结构。目前，Linux已被移植到数十种硬件平台上，Linux几乎支持所有流行的CPU。

2.1.2 以 Atmel Mega128 为处理器的解决方案

AVR 单片机的出现，使得许多以前以 ARM 为核心的产品开发有了一个新的思路，对于功能要求比较少，不需要多线程大工作量要求的终端产品，我们可以选用 AVR 系列单片机进行设计，下面介绍一下这种方案，框图如图 8：

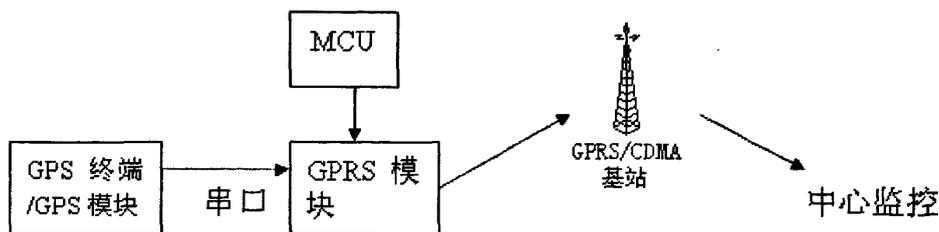


图 8 组成结构图

2.1.2.1 硬件部分功能设计

硬件部分主要完成核心电路、外围电路设计，并实现 GPS 模块和 GPRS 模块同 Mega 128 处理器的连接。

1) 车载终端：核心部件，主要负责与 GPS 模块、GPRS 模块的通信，并支撑嵌入式操作系统所需的底层硬件资源和相关的硬件部件。选用 Atmel 公司的 ATmega128 来构建车载监控终端，可以提升终端性能且节省空间。

2) GPS 模块：主要是接收 GPS 定位导航信息，实时提供车辆的位置、速度、时间等信息。设计中 GPS 信号的接收选用了 Leadtek 公司的 LR9548S GPS 模块。

3) GPRS 模块：主要负责与监控中心的通信。系统的无线通信模块采用的是 Wavecom 的新一代小尺寸和低功耗四频 GSM/GPRS 模块 SIM300，能够实现语音、SMS、数据和传真信息的高速传输，SIM300 的优良性能让它应用于许多方面，例如 WLL、M2M、手持设备等等。

2.1.2.2 软件部分功能设计

程序语言最终选用 C 作为开发语言，采用 ICC 编译器编译源码，AVR Studio 烧写程序。在编程上实现自主回传、中心呼叫回传、停止回传三种模式，开机后能实现自动拨号，终端便工作于 GPRS 在线状态，只要设置的 GPRS 中心 IP 地址有效且处于 GSM 信号范围，即可实现与中心进行数据通信。自主回传模式下采用固定时间间隔回传，间隔可由中心远程设定，自主回传模式下，车辆移动距离必须大于设定值才回传；终端需要有数据缓存功能，GPRS 离线或阻塞时，自动保存固定间隔的轨迹，GPRS 恢复通信联通时再自动补发，以确保 GPRS 阻塞时不会引起数据中断。

2.1.2.3 核心技术

AVR 单片机

AVR 单片机是 1997 年由 ATMEL 公司研发出的增强型内置 Flash 的 RISC (Reduced Instruction Set CPU) 精简指令集高速 8 位单片机。AVR 的单片机可以广泛应用于计算机外部设备、工业实时控制、仪器仪表、通讯设备、家用电器等各个领域。

AVR 单片机的优势特征：

高可靠性、功能强、高速度、低功耗和低价位，一直是衡量单片机性能的重要指标，也是单片机占领市场、赖以生存的必要条件。

早期单片机主要由于工艺及设计水平不高、功耗高和抗干扰性能差等原因，所以采取稳妥方案：即采用较高的分频系数对时钟分频，使得指令周期长，执行速度慢。以后的 CMOS 单片机虽然采用提高时钟频率和缩小分频系数等措施，但这种状态并未被彻底改观（51 以及 51 兼容）。此间虽有某些精简指令集单片机（RISC）问世，但依然沿袭对时

钟分频的作法。

AVR 单片机的推出，彻底打破这种旧设计格局，废除了机器周期，抛弃复杂指令计算机（CISC）追求指令完备的做法；采用精简指令集，以字作为指令长度单位，将内容丰富的操作数与操作码安排在一字之中（指令集中占大多数的单周期指令都是如此），取指周期短，又可预取指令，实现流水作业，故可高速执行指令。当然这种速度上的飞跃，是以高可靠性为其后盾的。

AVR 单片机硬件结构采取 8 位机与 16 位机的折中策略，即采用局部寄存器存堆（32 个寄存器文件）和单体高速输入/输出的方案（即输入捕获寄存器、输出比较匹配寄存器及相应控制逻辑）。提高了指令执行速度（1Mips/MHz），克服了瓶颈现象，增强了功能；同时又减少了对外设管理的开销，相对简化了硬件结构，降低了成本。故 AVR 单片机在软/硬件开销、速度、性能和成本诸多方面取得了优化平衡，是性价比的单片机。

AVR 单片机内嵌高质量的 Flash 程序存储器，擦写方便，支持 ISP 和 IAP，便于产品的调试、开发、生产、更新。内嵌长寿命的 E²Prom 可长期保存关键数据，避免断电丢失。片内大容量的 RAM 不仅能满足一般场合的使用，同时也更有效的支持使用高级语言开发系统程序，并可像 MCS-51 单片机那样扩展外部 RAM。

AVR 单片机的 I/O 线全部带可设置的上拉电阻、可单独设定为输入/输出、可设定（初始）高阻输入、驱动能力强（可省去功率驱动器件）等特性，使得 I/O 口资源灵活、功能强大、可充分利用。

AVR 单片机片内具备多种独立的时钟分频器，分别供 URAT、I2C、SPI 使用。其中与 8/16 位定时器配合的具有多达 10 位的预分频器，可通过软件设定分频系数提供多种档次的定时时间。AVR 单片机独有的“以定时器/计数器（单）双向计数形成三角波，再与输出比较匹配寄存器配合，生成占空比可变、频率可变、相位可变方波的设计方法（即脉宽调制输出 PWM）”更是令人耳目一新。

增强性的高速同/异步串口，具有硬件产生校验码、硬件检测和校验侦错、两级接收缓冲、波特率自动调整定位（接收时）、屏蔽数据帧等功能，提高了通信的可靠性，方便程序编写，更便于组成分布式网络和实现多机通信系统的复杂应用，串口功能大大超过 MCS-51/96 单片机的串口，加之 AVR 单片机高速，中断服务时间短，故可实现高波特率通讯。

面向字节的高速硬件串行接口 TWI、SPI。TWI 与 I²C 接口兼容，具备 ACK 信号硬件发送与识别、地址识别、总线仲裁等功能，能实现主/从机的收/发全部 4 种组合的多机通信。SPI 支持主/从机等 4 种组合的多机通信。

AVR 单片机有自动上电复位电路、独立的看门狗电路、低电压检测电路 BOD，多个复位源（自动上下电复位、外部复位、看门狗复位、BOD 复位），可设置的启动后延时运行程序，增强了嵌入式系统的可靠性。

AVR 单片机具有多种省电休眠模式，且可宽电压运行（5-2.7V），抗干扰能力强，

可降低一般 8 位机中的软件抗干扰设计工作量和硬件的使用量。

2.2 方案比较与确定

这一章前面大体上介绍了一下现在比较流行的两种车载终端的设计方案,两种方案各有千秋,各有各的优势和缺点,下面我们具体分析一下两种方案的区别。

以 ARM 为核心处理器的方案是一个现在比较成熟的方案,在现在的 GPS 导航领域,很多人都是以 ARM 作为主流处理器来进行开发和设计的,使用 ARM 处理器使得系统硬件性能突出,工作主频达 203MHz,片内资源丰富,相应的外围电路设计方便,便于实现复杂功能,且可扩展空间巨大。软件上使用大量优秀的开源软件,如 Uboot, Linux。Uboot 不但能提供引导作用,其在开发中的作用也非常强大,如 tftp 下载功能, flash 烧写等。Linux 提供进程管理,内存管理,驱动接口等。该车载监控终端功能强、运行稳定可靠,能够胜任车辆定位、跟踪等基本工作要求。

而以 ATmega128 为核心处理器的方案是一种比较经济的方案,ATmega128 是 AVR 单片机,与 ARM 同属于 RISC (reduced instruction set computer, 精简指令集计算机),RISC 微处理器不仅精简了指令系统,采用超标量和超流水线结构;它们的指令数目只有几十条,却大大增强了并行处理能力。如:1987 年 Sun Microsystem 公司推出的 SPARC 芯片就是一种超标量结构的 RISC 处理器。而 SGI 公司推出的 MIPS 处理器则采用超流水线结构,这些 RISC 处理器在构建并行精简指令系统多处理机中起着核心的作用,所以 ATmega128 在处理能力和处理速度上完全能够满足需要。软件上我们使用 C 语言进行编写,通过 ICC 进行编译,AVR Studio 烧写程序,省去了嵌入系统和修改内核以及重新编写头文件的麻烦,可以缩短开发所需要的周期。

另外,在费用上,以 ARM 为核心的方案在价格上相对处于弱势,一块以 ARM 为核心处理器的核心板价格一般在 500~700 元之间,如果我们自己去开发核心板,会增大开发成本以及延长开发周期,另外体积也比较大,再加上外围的扩展接口的话,会使整个系统的体积稍显大一点,不利于让终端看起来小巧、精致和方便搁置;而使用 ATmega128 的话,我们只需要把自己需要的几个接口扩展出来,可以使终端变得小巧而精致,不会占用车内的空间,并且价格也不贵,一片工业级的 ATmega128 在 30~40 元之间,令开发成本容易被客户接受。

本系统开发之前,根据用户的需求,讨论了以上两种方案,经过草案分析,以及客户在时间和经费上的多种要求,最终确定了以 ATmega128 为核心处理器的方案。从下章开始,从硬件和软件两个方面详细介绍以 ATmega128 为核心处理器的 GPS 车载终端的具体设计方案。

第三章 终端硬件设计与实现

3.1 终端硬件总体设计

嵌入式硬件开发首先要根据终端所要实现的功能来确定系统硬件应该有的组成部分，这是进行开发的第一步。本终端根据所要实现的功能，在逻辑上划分为5个部分：供电电路模块，MCU模块，GPS信息采集模块，GPRS通信模块，RS232通信模块。

- 1) 供电电路模块：选用宽幅输入的电源模块 MIC29302BT 来设计 SIM300 的供电电路，其他的 5V 电源由电源模块 MC78M05 变压输出。
- 2) MCU 模块：这部分由我们的核心处理器 ATMEGA128L 以及 ST16C550 即带 16 字节 FIFO 编码器/译码器的通用异步收发器组成。
- 3) GPS 信息采集模块：GPS 模块选用丽台的 LR9548S，并根据接口设计外围电路，使其能够准确的接收到卫星传来的 GPS 信息。
- 4) GPRS 通信模块：GPRS 模块选用的是 Wavecom 公司自带协议的 sim300 GPRS OEM 模块，针对所选用的 GPRS 模块设计相应的电路，以完成终端和控制中心服务器的无线通信。
- 5) RS232 通信模块：这部分我们选用了 MAX3243，即 3V~5.5V 多通道 RS232 线驱动器/接收器，以及 SP3238，即智能+3.0V~+5.5V RS-232 收发器来组成，设计本部分硬件电路完成各个硬件之间的通讯。

终端大体的结构框图（如图 9）：

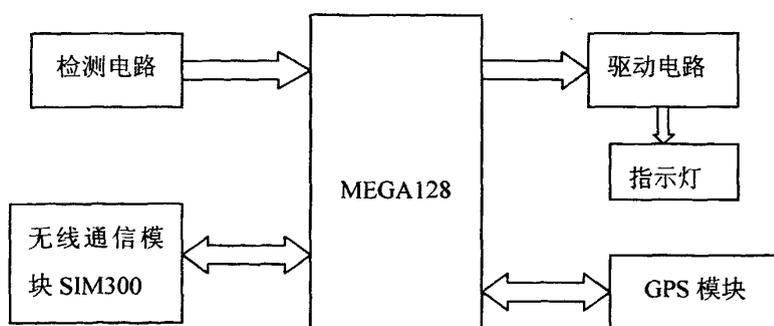


图 9 终端结构框图

3.2 终端硬件具体实现

下面将按照 MCU 模块、GPS 模块、GPRS 模块、RS232 模块和供电电路模块的顺序详细介绍一下终端硬件各个部分。

3.2.1 MCU 模块

MMU 模块是终端正常运行所必须的核心模块，包括主控制器，存储器以及时钟等等，所以在设计上需要仔细考虑。

3.2.1.1 主控制器

主控制器我们选用的是 ATmega128，RISC 单片机，它的主要特性如下^[12]：

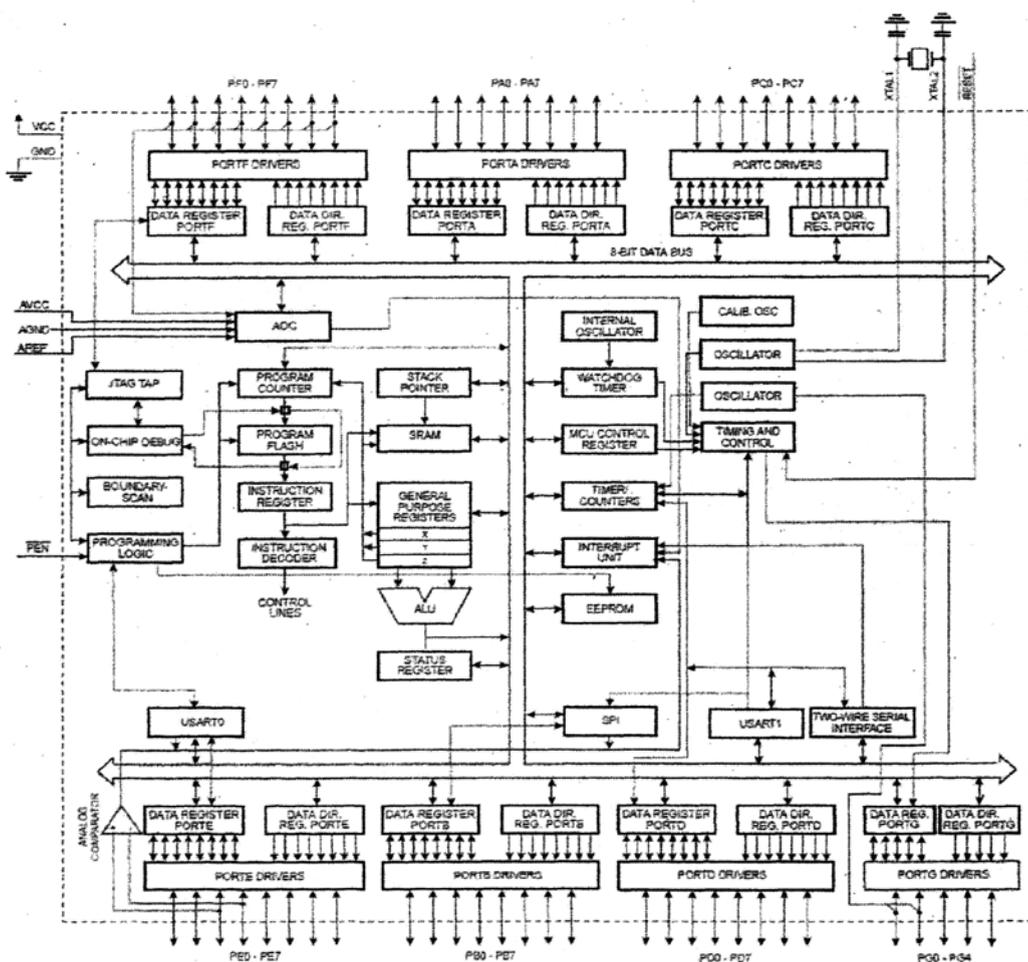


图 10 ATmega128 结构框图

- 高性能、低功耗的 AVR® 8 位微处理器
- 先进的 RISC 结构
 - 133 条指令
 - 大多数可以在一个时钟周期内完成
 - 32 x 8 通用工作寄存器 + 外设控制寄存器
 - 工作于 16 MHz 时性能高达 16 MIPS
- 非易失性的程序和数据存储器
 - 128K 字节的系统内可编程 Flash
- 寿命: 10,000 次擦写周期
 - 具有独立锁定位、可选择的启动代码区
- 通过片内的启动程序实现系统内编程
- 真正的读—修改—写操作
 - 4K 字节的 EEPROM
 - 4K 字节的内部 SRAM
 - 多达 64K 字节的优化的外部存储器空间
 - 可以对锁定位进行编程以实现软件加密
 - 可以通过 SPI 实现系统内编程
- JTAG 接口 (与 IEEE 1149.1 标准兼容)
 - 遵循 JTAG 标准的边界扫描功能
 - 支持扩展的片内调试
 - 通过 JTAG 接口实现对 Flash, EEPROM, 熔丝位和锁定位的编程
- 外设特点
 - 两个具有独立的预分频器和比较器功能的 8 位定时器/计数器
 - 两个具有预分频器、比较功能和捕捉功能的 16 位定时器/计数器
 - 具有独立预分频器的实时时钟计数器
 - 两路 8 位 PWM
 - 6 路分辨率可编程 (2 到 16 位) 的 PWM
 - 8 路 10 位 ADC
 - 8 个单端通道
 - 7 个差分通道
 - 2 个具有可编程增益 (1x, 10x, 或 200x) 的差分通道
 - 面向字节的两线接口
 - 两个可编程的串行 USART
 - 可工作于主机/从机模式的 SPI 串行接口
 - 具有独立片内振荡器的可编程看门狗定时器

- 片内模拟比较器
- 特殊的处理器特点
 - 上电复位以及可编程的掉电检测
 - 片内经过标定的 RC 振荡器
 - 片内/ 片外中断源
 - 6 种睡眠模式：空闲模式、ADC 噪声抑制模式、省电模式、掉电模式、Standby 模式以及扩展的 Standby 模式
 - 可以通过软件进行选择时钟频率
 - 全局上拉禁止功能
- I/O 和封装
 - 53 个可编程 I/O 口线
 - 64 引脚 TQFP 与 64 引脚 MLF 封装

ATmega128 为基于 AVR RISC 结构的 8 位低功耗 CMOS 微处理器。由于其先进的指令集以及单周期指令执行时间，ATmega128 的数据吞吐率高达 1 MIPS/MHz，从而可以缓减系统在功耗和处理速度之间的矛盾。

AVR 内核具有丰富的指令集和 32 个通用工作寄存器。所有的寄存器都直接与逻辑单元 (ALU) 相连接，使得一条指令可以在一个时钟周期内同时访问两个独立的寄存器。这种结构大大提高了代码效率，并且具有比普通的复杂指令集微处理器高 10 倍的数据吞吐率。

ATmega128 具有如下特点：128K 字节的系统内可编程 Flash（具有在写的过程中还可以读的能力，即 RWW）、4K 字节的 EEPROM、4K 字节的 SRAM、53 个通用 I/O 口线、32 个通用工作寄存器、实时时钟 RTC、4 个灵活的具有比较模式和 PWM 功能的定时器/计数器 (T/C)、两个 USART、面向字节的两线接口 TWI、8 通道 10 位 ADC（具有可选的可编程增益）、具有片内振荡器的可编程看门狗定时器、SPI 串行端口、与 IEEE 1149.1 规范兼容的 JTAG 测试接口（此接口同时还可以用于片上调试），以及六种可以通过软件选择的省电模式。空闲模式时 CPU 停止工作，而 SRAM、T/C、SPI 端口以及中断系统继续工作；掉电模式时晶体振荡器停止振荡，所有功能除了中断和硬件复位之外都停止工作，寄存器的内容则一直保持；省电模式时异步定时器继续运行，以允许用户维持时间基准，器件的其他部分则处于睡眠状态；ADC 噪声抑制模式时 CPU 和所有的 I/O 模块停止运行，而异步定时器和 ADC 继续工作，以减少 ADC 转换时的开关噪声；Standby 模式时振荡器工作而其他部分睡眠，使得器件只消耗极少的电流，同时具有快速启动能力；扩展 Standby 模式则允许振荡器和异步定时器继续工作。

器件是以 ATMEL 的高密度非易失性内存技术生产的。片内 ISP Flash 可以通过 SPI 接口、通用编程器，或引导程序多次编程。引导程序可以使用任何接口来下载应用程序到应用 Flash 存储器。在更新应用 Flash 存储器时引导 Flash 区的程序继续运行，实现 RWW

(Read-While-Write) 操作。通过将 8 位 RISC CPU 与系统内可编程的 Flash 集成在一个芯片内, ATmega128 为许多嵌入式控制应用提供了灵活而低成本方案。

ATmega128 AVR 有整套的开发工具, 包括 C 编译器, 宏汇编, 程序调试器/ 仿真器和评估板。

综上所述, ATmega128 具有较宽的工作温湿度范围、较好的工业参数、丰富的外部设备和数据传输特性, 因此选用其作为主控制器。

ATmega128 的外围引脚原理图 (如图 11) :

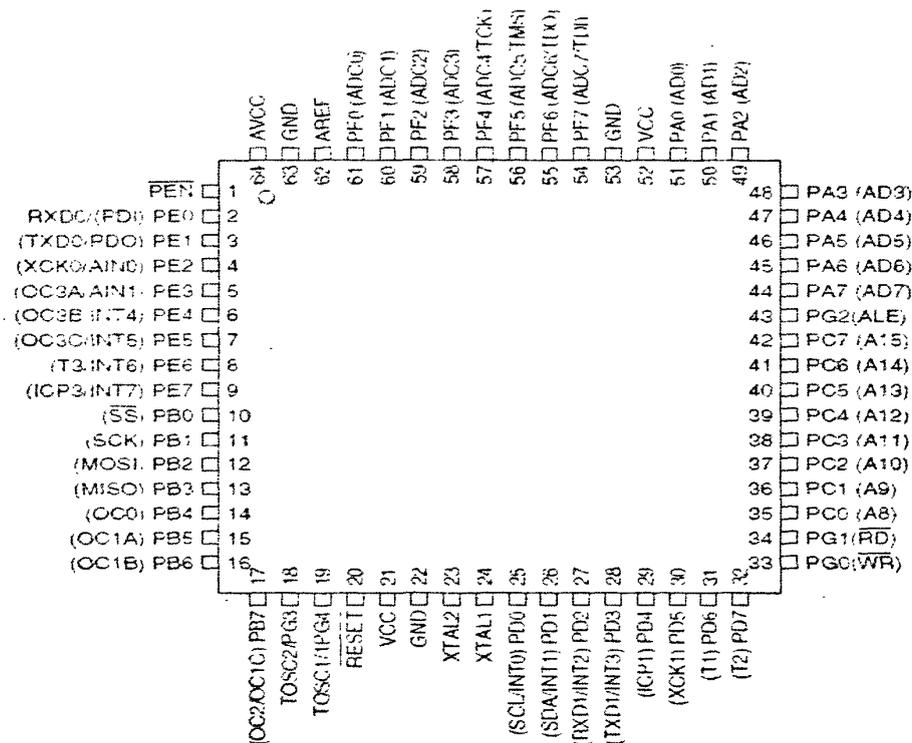


图 11 ATmega128 的引脚

➤ AVR ATmega128 存储器

AVR 结构具有两个主存储器空间: 数据寄存器和程序寄存器。此外, ATmega128 还有 EEPROM 存储器以保存数据。这三个存储器空间都是线性的。

ATmega128 具有 128K 字节的在线编程 Flash。因为所有的 AVR 指令为 16 位或 32 位, 故尔 FLASH 组织成 64K x 16 的形式。考虑到软件安全性, Flash 程序存储器分为两个区: 引导程序区和应用程序区。

Flash 存储器至少可以擦写 10,000 次。ATmega128 的程序计数器 PC 为 16 位, 因此可以寻址 64K 的程序存储器。

ATmega128 是一个复杂的微处理器, 其支持的外设要比预留的 64 个 I/O (通过 IN/OUT 指令访问) 所能支持的要多。对于扩展的 I/O 空间 \$60 - \$FF, 只能使用 ST/STS/STD 和

LD/LDS/LDD 指令。

ATmega128 包含 4K 字节的 EEPROM。它是作为一个独立的数据空间而存在的，可以按字节读写。EEPROM 的寿命至少为 100,000 次（擦除）。EEPROM 的访问由地址寄存器，数据寄存器和控制寄存器决定。

ATmega128 的所有 I/O 和外设都被放置在 I/O 空间。所有的 I/O 地址都可以通过 LD/LDS/LDD 和 ST/STS/STD 指令来访问，在 32 个通用工作寄存器和 I/O 之间传输数据。地址为 \$00 - \$1F 的 I/O 寄存器还可用 SBI 和 CBI 指令直接进行位寻址，而 SBIS 和 SBIC 则用来检查单个位置位与否。当使用 IN 和 OUT 指令时地址必须在 \$00 - \$3F 之间。如果要像 SRAM 一样通过 LD 和 ST 指令访问 I/O 寄存器，相应的地址要加上 \$20。ATmega128 是一个复杂的微处理器，其支持的外设要比预留的 64 个 I/O（通过 IN/OUT 指令访问）所能支持的要多。对于扩展的 I/O 空间 \$60 - \$FF，只能使用 ST/STS/STD 和 LD/LDS/LDD 指令。

由于外部存储器接口所提供的特性，此接口非常适合于与存储器器件互连，如外部 SRAM 和 Flash，LCD，A/D，D/A 等。其主要特点为：

- 四个不同的等待状态设置（包括无等待状态）。
- 不同的外部存储器可以设置不同的等待状态。
- 地址高字节的位数可以有选择地确定。
- 数据线具有总线保持功能以降低功耗。

连接方法（如图 12）：

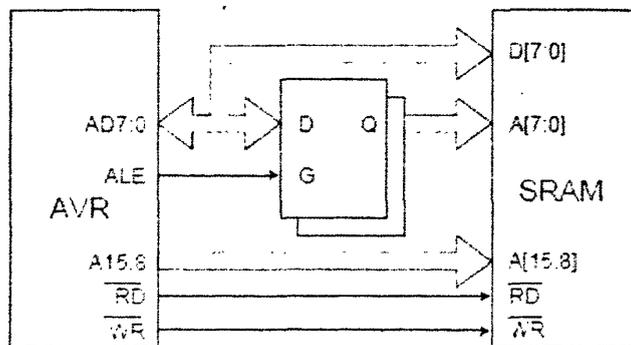


图 12 与 AVR 连接的外部 SRAM

3.2.1.2 串行通信接口

串行接口的实现技术已经非常成熟，本文选用比较常见的 ST16C55 芯片^[13]。ST16C550 是可编程的通用异步接收/发送器，工作在 3.3V 或 5V，内部收发均有 16 个字节 FIFO，能实现数据的串行与 8 位并行之间的转换，支持异步通信协议。片内有时钟产

生电路，最高工作频率 24MHz，波特率可以从 50bps~1.5Mbps。ST16C550 可以为应用于远程通信系统中的调制解调器提供控制信号，并接收和记录调制解调器的状态信息。

ST16C550 是 EXAR 公司推出的最稳定、最可靠的通用异步收发器 (UART) 电路，能够提供数据的串/并、并/串转换功能。其中，串行数据流的同步功能是通过在传输数据中加入起始位和结束位组成的数据字节来实现的，通过在数据字节中附加奇偶校验位确保数据的完整性。接收方检验奇偶校验位，确定是否出现传输错误。用普通电路实现这些功能非常复杂，尤其是将电路集成在单个芯片上时。ST16C550 采用 0.6 μ m CMOS 工艺，达到低功耗、高速度、高集成度。ST16C550 带有 16 字节收发 FIFO 存储器，高速存储器进行通信，要求具有很高的处理速度。ST16C550 的封装图 (如图 13)：

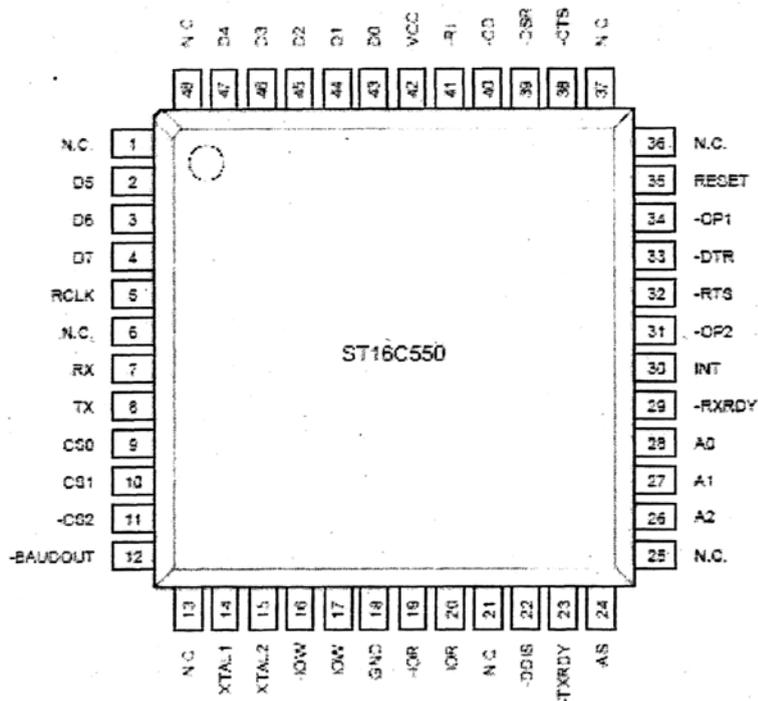


图 13 ST16C550 的引脚图

3.2.1.3 MCU 模块的原理图

MCU 模块具体的原理图 (如图 14) :

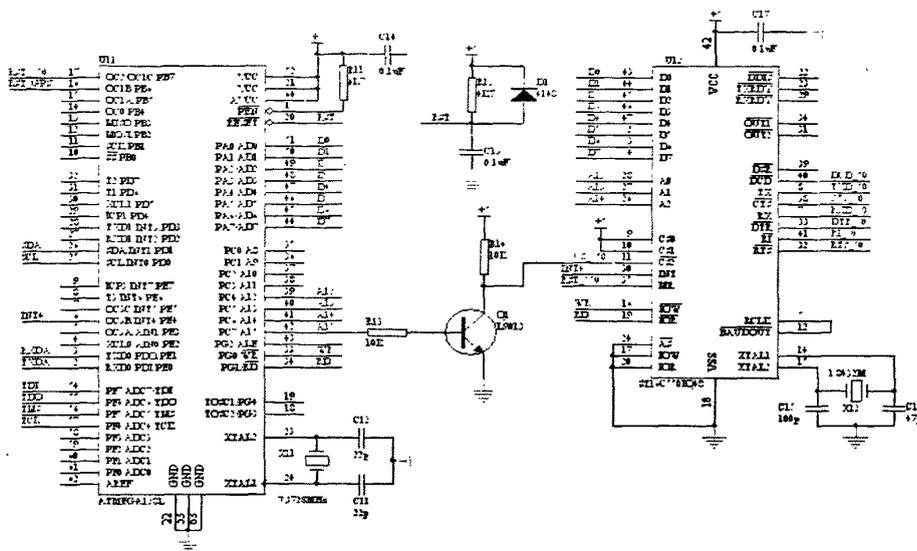


图 14 MCU 模块原理图

如图所示, 在硬件连接上, 三个片选信号中 CS0、CS1 接 5V 电源, CS2 接模拟地选中 ATmega128; \overline{IOR} 、 \overline{IOW} 直接与 ATmega128 的读写信号线 \overline{RD} 、 \overline{WR} 相连; 数据线 D0-D7 与 ATmega128 的低 8 位数据线相连; 片内寄存器选择线 A2-A0 接 ATmega128 的 PC4-PC6; \overline{AS} 直接接地, 确保 A0-A2 有效。XTAL1、XTAL2 与 1.8432MHz 晶振相连, 组成内部振荡电路。晶振与 ST16C550 之间的连线尽量短而粗。

3.2.2 GPS 信息采集模块

3.2.2.1 导航定位技术

卫星导航定位是指利用卫星导航定位系统提供的位置、速度及时间等信息来完成对各种目标的定位、导航、监测和管理。卫星导航定位系统是一种以卫星为基础的无线电导航系统, 可提供高精度、全天时、全天候的导航、定位和授时信息, 是一种可供海、陆、空军民用用户共享的信息资源。^[14]

世界上最早的卫星导航系统是美国的子午仪导航系统 (1964 年开始运行)。随后, 为满足日益增长的军事需要, 20 世纪 60 年代末 70 年代初, 美国和前苏联分别开始研制全天候、全天时、连续实时提供精确定位服务的新一代全球卫星导航系统, 到 90 年代中期全球卫星导航定位系统 GPS 和 GLONASS 均已建成并投入运行。我国也建设了具有自主知识产权的北斗卫星定位系统, “北斗一号” 于 2003 年底正式开通运行。欧盟筹建的 Galileo 全球卫星导航系统正在计划实施之中。

综合考虑各系统性能以及全球卫星导航接收机芯片技术等方面, 本系统中选用了全球定位系统 GPS, 获得高精度的定位信息。^[15]

3.2.2.2 GPS 卫星定位技术

全球定位系统^[16] (Global Position System--GPS) 是美国从本世纪 70 年代由美国国防部批准开始研制, 历时 20 年, 耗资 300 亿美元, 于 1994 年全面建成, 是具有在海、陆、空进行全方位实时三维导航与定位能力的新一代卫星导航与定位系统。全球定位系统是在子午仪卫星导航系统的基础上发展起来的, 采纳了子午仪系统的成功经验, 是美国第二代卫星导航系统。

GPS 系统包括三大部分^[17]: 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。

空间部分为 21 颗工作卫星和 3 颗在轨备用卫星组成的卫星星座。卫星分布在 6 个轨道平面上, 每个轨道平面上有 4 颗卫星, 在约 2 万千米高空的卫星, 从地平线升起至没落, 可以在用户视野持续 5 小时左右。每一个用户在任何地方都能够同时接收到来自 4~12 颗 GPS 卫星的定位信号, 实现全球性全天时的连续不断的导航定位。

控制部分主要由 1 个主控站、5 个监控站、3 个地面注入站组成, 形成一个分布在全世界的地面控制监视网, 监视着各个卫星的工作状态。主控站主要协调和管理地面监控系统的工作, 监控站是在主控站直接控制下的数据采集中心, 地面注入站的主要任务是在主控站的控制下, 将主控站推算和编制的卫星星历、卫星时钟偏差、导航电文和其他指令等注入到相应卫星的存储系统。

用户设备的主要任务是接收 GPS 卫星发射的信号, 以获得必要的导航和定位信息及参数, 经过数据处理, 实现导航和定位功能。

1. GPS 卫星定位原理

GPS 是利用测距交会原理确定点的位置来进行定位的^[18]。它采用多星高轨测距体制, 以接收机到 GPS 卫星之间的距离作为基本观测量。当地面用户的 GPS 接收机同时接收到 3 颗以上卫星的信号后, 通过使用伪距测量或载波相位测量, 测算出卫星信号到接收机所需要的时间、距离, 再结合各卫星所处的位置信息, 将卫星至用户的多个等距离球面相交后, 即可确定用户的三维(经度、纬度、高度)坐标位置以及速度、时间等相关参数。定位原理图(如图 15)所示, 假设 t 时刻在地面待测点上安置 GPS 接收机, 可以测定 GPS 信号到达接收机的时间 Δt , 再加上接收机所接收到的卫星星历等其它数据可以确定以下四个方程式:

$$\left[\begin{matrix} (X_1 - X) \\ (Y_1 - Y) \\ (Z_1 - Z) \end{matrix} \right]^2 - C(V_{t1} - V_{t0}) = d_1 \quad (1)$$

$$\left[\begin{matrix} (X_2 - X) \\ (Y_2 - Y) \\ (Z_2 - Z) \end{matrix} \right]^2 - C(V_{t2} - V_{t0}) = d_2 \quad (2)$$

$$\left[\begin{matrix} (X_3 - X) \\ (Y_3 - Y) \\ (Z_3 - Z) \end{matrix} \right]^2 - C(V_{t3} - V_{t0}) = d_3 \quad (3)$$

$$\left[\begin{matrix} (X_4 - X) \\ (Y_4 - Y) \\ (Z_4 - Z) \end{matrix} \right]^2 - C(V_{t4} - V_{t0}) = d_4 \quad (4)$$

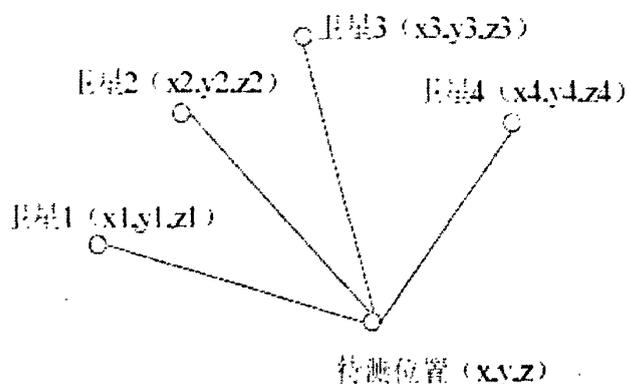


图 15 GPS 定位原理

上述四个方程式中待测点坐标 X , Y , Z 和 V_{t0} 为未知参数, 其中 $d_i = c\Delta t_i$ ($i=1, 2, 3, 4$). d_i ($i=1, 2, 3, 4$) 分别为卫星 1、卫星 2、卫星 3、卫星 4 到接收机之间的距离。 Δt_i ($i=1, 2, 3, 4$) 分别为卫星 1、卫星 2、卫星 3、卫星 4 的信号到达接收机所经历的时间。 c 为 GPS 信号的传播速度 (即光速)。

四个方程式中各个参数意义如下:

X, Y, Z 为待测点坐标的空间直角坐标。 X_i, Y_i, Z_i ($i=1, 2, 3, 4$) 分别为卫星 1、卫星 2、卫星 3、卫星 4 在 t 时刻的空间直角坐标, 可由卫星导航电文求得。 V_{ti} ($i=1, 2, 3, 4$) 分别为卫星 1、卫星 2、卫星 3、卫星 4 的卫星钟的钟差, 由卫星星历提供。 V_{t0} 为接收机的钟差。

由以上四个方程即可解算出待测点的坐标 X, Y, Z 和接收机的钟差 V_{t0} 。

2. GPS 的应用

全球定位系统 GPS 拥有全球性、全能性、全天候性的导航定位、定时、测速等优点。

在诸多领域中得到越来越广泛的应用，最早应用于军用定位和导航。随着技术的发展和完善，目前全球卫星定位系统 GPS 已逐步从军用扩展到民用，主要涉及海、陆、空的导航和定位，并在工业、农业、测绘、气象等领域得到广泛应用，使世界交通运输业发生了深刻变革，推动了航天事业的发展。下面介绍一下 GPS 在智能交通中的应用。

GPS 在 ITS 中主要应用于车辆定位、导航和交通管理，是 ITS 的重要组成部分。在任一时刻、任一目标能通过 GPS 系统得知汽车的经纬度、速度和准确时间，然后把这些信息通过无线通讯网络提供给监控中心，监控中心负责在电子地图上显示出车辆运行轨迹；同时，监控中心可根据路况信息，发出调度指令，来完成对车辆的集中监控。国外早已进行研究，并已取得了一定成果，已广泛地应用于公共医疗事业、公共服务事业、银行、消防、公安等行业。目前，国内在车辆监控系统的开发上投入了大量的精力，以灵图和东方泰坦为代表。GPS 技术在我国道路工程和交通管理中的应用还刚刚起步，相信随着我国经济的发展、高等级公路的快速修建和 GPS 技术应用研究的逐步深入，其在道路工程中的应用也会更加广泛和深入，并发挥更大的作用。

3. GPS 导航定位系统的特点

GPS 与其它导航系统相比，具有的优点概括起来主要有以下几个方面^[16]：

- 定位精度高：通过很多应用实践已经证明，GPS 相对定位精度在 50km 以内可达 10⁻⁶，100~500km 可达 10⁻⁷，1000km 以上可达 10⁻⁹，在 300~1500m 工程精密定位中，1 小时以上观测的解算，其平面位置误差小于 1mm。基线边长越长越能突显其定位精度高的优势。
- 观测时间短：采用 GPS 布设控制网时每个测站上的观测时间一般在 30~40min 左右，采用快速静态定位方法，观测时间更短。目前 20km 以内相对静态定位，仅需 15~20 分钟；快速静态相对定位测量时，当每个流动站与基准站相距在 15km 以内时，流动站只需观测 1~2 分钟；动态相对定位测量时，流动站出发时观测 1~2 分钟，然后可随时定位，每站观测仅需几秒钟。
- 测站间无需通视：GPS 测量不要求站点间相互通视，只需测站上空开阔即可。
- 可提供三维坐标：经典大地测量将平面与高程采用不同方法分别施测，而 GPS 可同时精确测定测站点的三维坐标，目前 GPS 水准可达到四等水准测量的精度。
- 操作简单：随着 GPS 的不断改进，自动化程度越来越高，体积也越来越小，重量越来越轻。
- 全天候作业：目前 GPS 观测可在一天 24 小时内的任何时间进行，不受阴天黑夜、起雾刮风、下雨下雪等天气状况的影响。
- 功能多、应用广：GPS 系统不仅可以用于定位测量，还可用于测速、测时。测速精度可达 0.1ms，测时精度可达几十毫秒，精度都比较高。随着 GPS 技术的不断提高，其应用领域正在不断的扩大。

3.2.2.3 Leadtek LR9548S GPS

我们的终端选用了台湾丽台（Leadtek）的LR9548S GPS模块，外观图（如图16）：

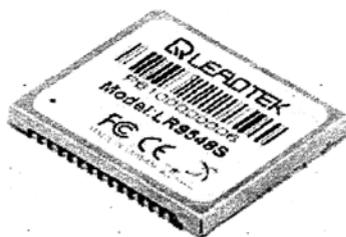


图 16 Leadtek LR9548S GPS

丽台科技（Leadtek）GPS 9548 是一款专为 OEM 应用设计，具有高灵敏度、低功耗、SMD 型、20 通道等特性的 GPS 接收模块。GPS 9548 基于丽台 GPS 产品以往的技术，为快速的整合进标准或定制应用提供了方便。

高灵敏度：GPS 9548 相比其它的独立的 GPS 解决方案，能使用户在很低的信号强度下获取并连续的追踪 GPS 信号。这意味着它可以使用在以前认为不可能使用的环境下，比如林立的都市大厦中，茂密的树林中，车库中，和许多室内环境下。

轻松的定制：丽台 GPS 9548 基于最新的 SiRFStar III 技术，为快速整合进标准或定制应用提供了方便。

3.2.2.4 GPS 信息采集模块原理图

GPS 信息采集模块的原理图（如图 17）：

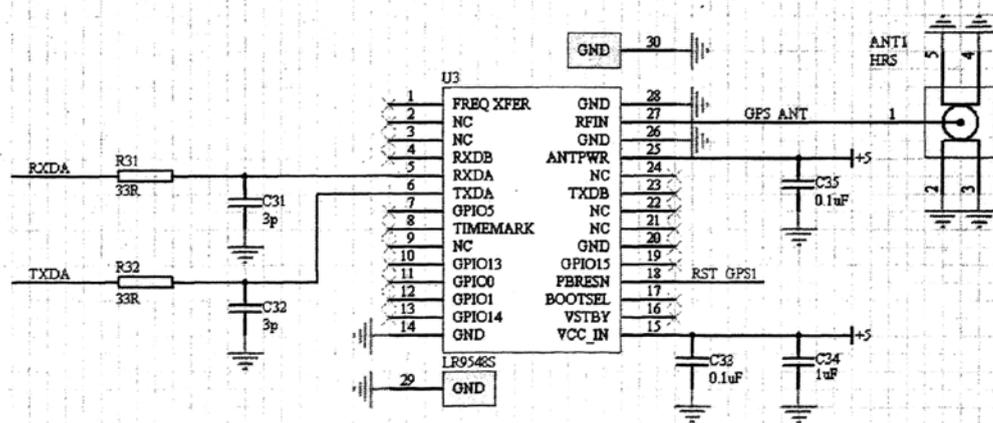


图 17 GPS 信息采集模块原理图

如图所示，GPS 的收发信号端直接连接到 ATmega128 的数据收发端口，除天线引脚、

电源引脚以及地线外，大部分引脚作悬空处理。

3.2.3 GPRS 通信模块

本设计中 GPRS 通信模块采用了 Wavecom 公司自带协议的 SIM300 GPRS OEM 模块，本章节详细描述分析了 GPRS 技术以及 SIM300 模块，并在此基础上完成了无线传输模块的电路接口设计。

3.2.3.1 GPRS 无线通讯技术

GPRS (General Packet Radio Service) 是通用分组无线业务的英文简称。GPRS 网络是在现有 GSM 网络上增加 GGSN (Gate GPRS Supporting Node, GPRS 网关支持节点) 和 SGSN (Serving GPRS Supporting Node, GPRS 服务支持节点) 来实现的，使得用户能够在端到端分组方式下发送和接收数据，同时兼容电路型数据和分组交换数据，属于 2.5 代移动通讯系统，它在技术上已非常成熟。^[19]目前中国移动通讯公司的 GPRS 业务已非常成熟。

1. GPRS 系统基本工作原理

GPRS 是在现有的 GSM 系统上发展出来的一种新的分组数据承载业务^[20]。与现有的 GSM 语音系统最根本的区别是，GSM 是一种电路交换系统，而 GPRS 是一种分组交换系统。它特别适用于间断的、突发性的或频繁的、少量的数据传输，也适用于偶尔的大数据量传输，这一特点正适合大多数移动互联的应用^[21]。

GPRS 因为引入了 SGSN 和 GGSN，其数据支持能力较之 GSM 网络增强很多，在理想情况下，最高数据速率可达 172Kbs，因此 GPRS 技术正不断受到青睐。GPRS 系统采用无线分组交换技术，提供端到端、广域的无线 IP 连接。通俗地讲，GPRS 是一项高速的数据处理技术，方法是以“分组”的形式将信息分发到客户手上。其系统结构如图 18 所示：

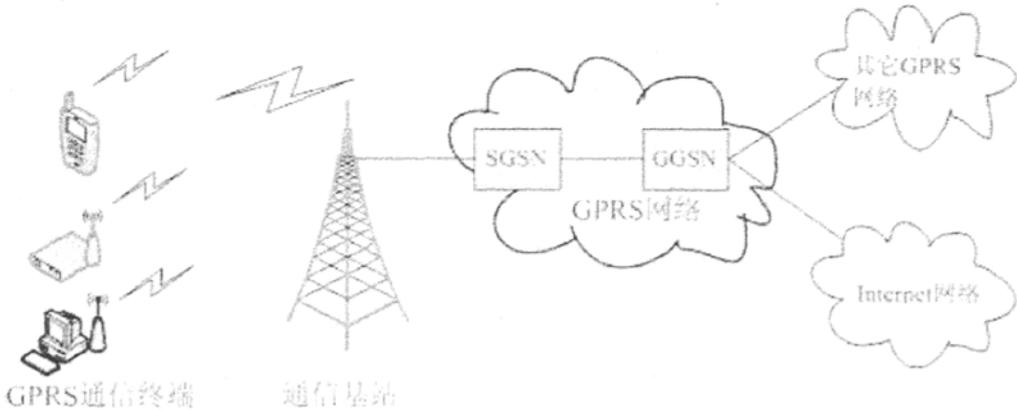


图 18 GPRS 系统结构图

网络终端利用 GPRS 通信模块通过 GPRS 网络实现与 Internet 的连接；GPRS 通讯模块

与 GSM 基站通信，但与电路交换式数据呼叫不同的是：GPRS 分组是从基站发送到 SGSN，而不是连接到语音网络上。SGSN 与 GGSN 进行通信；GGSN 对分组数据进行相应的处理，再发送到目的网络，比如因特网或 X.25 网络。来自因特网标识有移动台地址的 IP 包，由 GGSN 接收，再转发到 SGSN，继而传送到移动台上。SGSN 是 GSM 网络结构中的一个节点，它与 MSC 处于网络体系的同一层。SGSN 通过帧中继与 BTS 相连，是 GSM 网络结构与移动台之间的接口。SGSN 的主要作用是记录移动台的当前位置信息，并且在移动台和 GGSN 之间完成移动分组数据的发送和接收。

GGSN 通过基于 IP 协议的 GPRS 骨干网连接到 SGSN，是连接 GSM 网络和外部分组交换网（如因特网和局域网）的网关。GGSN 主要是起网关作用，也将 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 TCPIIP 或 X.25 网络。SGSN 和 GGSN 利用 GTP（GPRS 隧道协议）对正或 X.2.5 分组进行封装，实现二者之间的数据传输。^[22]

2. GPRS 的技术优势主要有以下几个方面^[23]：

(1) 资源利用率高。GPRS 引入了分组交换的传输模式，使得原来采用电路交换模式的 GSM 传输数据方式发生了根本性的变化，这在无线资源稀缺的情况下显得尤为重要。按电路交换模式来说，在整个连接期内，用户无论是否传送数据都将独自占有无线信道。而对于分组交换模式，用户只有在发送或接收数据期间才占用资源，这意味着多个用户可高效率地共享同一无线信道，从而提高了资源的利用率。GPRS 用户的计费以通信的数据量为主要依据，体现了“得到多少、支付多少”的原则。实际上，GPRS 用户的连接时间可能长达数小时，却只需支付相对低廉的连接费用。

(2) 传输速率高。GPRS 可提供高达 115kbit/s 的传输速率（最高值为 171.2kbit/s，不包括 FEC）。这意味着通过便携式电脑，GPRS 用户能和 ISDN 用户一样快速地上网浏览，同时也使一些对传输速率敏感的移动多媒体应用成为可能。

(3) 接入时间短。分组交换接入时间缩短为少于 1 秒，能提供快速即时的连接，可大幅度提高一些事务（如信用卡核对、远程监控等）的效率，并可使已有的 Internet 应用（如 E-mail、网页浏览等）操作更加便捷、流畅。

(4) 支持 IP 协议和 X.25 协议。GPRS 支持因特网上应用最广泛的 IP 协议和 X.25 协议。且由于 GSM 网络覆盖面广，使得 GPRS 能提供 Internet 和其它分组网络的全球性无线接入。

3. 系统中 GPRS 通信组网方式的选取

虽然 GPRS 网络使用 IP 协议，但是其 IP 地址是 GPRS 内网地址，并且这个内网 IP 地址大多是动态分配的。目前 GPRS 主要组网方式有五种：

(1) 公网固定 IP 方式。主机接入公网，即 Internet 网络，具有固定 IP 地址。在这种方式下，先由终端机发起连接，终端机登录 GPRS 后获得 IP 地址，将这个 IP 地址发送给主机。由于终端机的 IP 地址是 GPRS 内网的 IP 地址，所以要经过 GPRS 网络的 NAT

(Network Address Transfer) 服务器进行网络地址转换, 之后主机和终端机之间就可以通信。

(2) 公网动态 IP 方式, 主机接入公网, 但是其 IP 地址是动态的。这种情况下, 要借助于公网动态 DNS (域名解析服务器)。监控点先采用域名寻址方式连接 DNS 服务器, 再由 DNS 服务器找到主机公网动态 IP 建立连接。

(3) GPRS 专线方式, 主机使用专线接入 GPRS 网络。在这种方式下, 主机具有 GPRS 内网的固定 IP 地址, 其通信过程和方式一的公网固定 IP 的组网相似。区别在于通信过程发生在 GPRS 网内, 和 Internet 公网无关。

(4) GPRS 动态 IP 方式, 主机无线方式接入 GPRS, 连接 GPRS 内网的 DNS 服务器, 获得 GPRS 内网动态 IP 地址。远程监控终端采用域名寻址, 由 DNS 服务器找到主机动态 IP, 以建立连接。

(5) 绑定 IP 地址方式, 主机无线方式接入 GPRS 网络, 其 IP 地址和 SIM 卡号绑定, 此时的主机也具有 GPRS 内网的静态 IP 地址, 其通信过程和方式三 GPRS 专线组网相似。

在使用 GPRS 网络组建监控系统时, 如果监控中心服务器 (或称主机) 和 GPRS 终端机都使用动态 IP 地址, 它们是无法通信的。由于 GPRS 的无线接入动态 IP 方式比较容易, 建设费用和运行费用较低, 所以在监控系统中尽量多地使用这种方式。同时, 监控系统多是由一台服务器和多台终端机组成的星型结构, 所以最合理的办法是终端机使用无线接入动态 IP 方式而服务器使用静态 IP 地址。^[24]

组网结构 (详见图 19) 所示, 实验室监控中心数据库服务器接入通过宽带接入 Internet, 具有固定 IP 地址。在这种方式下, 由远程监控终端发起连接, 登录 GPRS 后获得 IP 地址, 同时主动向数据库服务器端建立面向连接 (TCP) 的通信机制, 主机和终端机之间实现通信。主要考虑依托实验室现有网络资源建立可靠、稳定的通信方式。

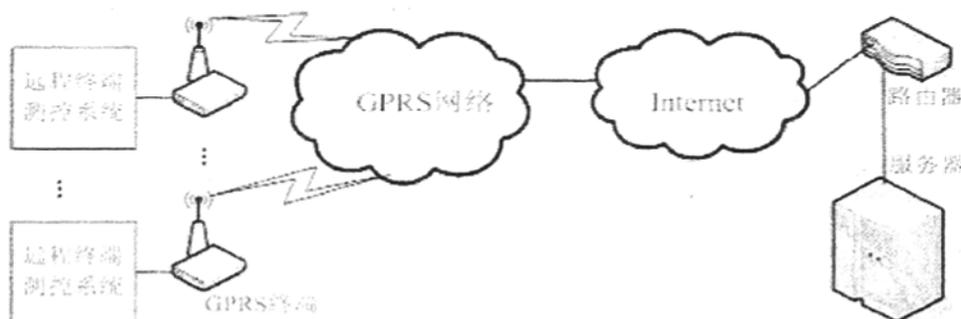


图 19 公网固定 IP 的 GPRS 组网结构图

3.2.3.2 SIM300 GPRS 模块

SIM300 是小体积即插即用模组中完善的四频 GSM/GPRS 解决方案, 使用工业标准界

来稳压，30 引脚接入网指示灯，34 引脚接工作指示灯，50、51 接模拟地，28、38、40、42、44、46、48 引脚接数据位，19、21、23、25 接 sim 卡，用来入网，其他大部分引脚作悬空。

3.2.4 RS232 通信模块

RS232 接口是 PC 机与通信工业中应用广泛的一种串行接口，被美国电子工业协会定义为“在数据终端设备和数据通讯设备之间使用串行二进制数据交换的接口”。作为一种硬件协议，RS232 被用于连接 DTE (Data terminal Equipment, 数据终端设备) 和 DCE (Data Communication Equipment, 数据通信设备) 两种设备。^[25] RS-232-C 标准接口有 9 芯或 25 芯的 D 型插头。要完成最基本的串行通信功能，实际上只需要 RXD、TXD 和 GND 即可，但由于 RS-232-C 标准所定义的高、低电平信号与 ATmega128 系统的 LVTTTL 电路所定义的高、低电平信号完全不同，LVTTTL 的标准逻辑“1”对应 2V~3.3V 电平，标准逻辑“0”对应 0V~0.4V 电平，而 RS-232-C 标准采用负逻辑方式，标准逻辑“0”对应 +5V~+15V 电平，标准逻辑“1”对应 -5V~-15V 电平，显然，两者间要进行通信必须经过信号电平的转换，本设计选用的智能收发器 SP3238 和 MAX3243 通用同步异步收发器 (USART) 提供一个全双工通用同步异步串行连接。

3.2.4.1 智能+3.0V~+5.5V RS-232 收发器 SP3238

在 GPS 信息采集的过程中，以 ATmega128 单片机为控制单元，单片机实用 TTL 电平，SP3238 是 3.0V-5.5V 的 RS232 转换芯片，它可以完成 TTL 电平与 RS232 电平之间的转换以及串口通信功能，具有低功耗、高数据速率、增强型 ESD 保护等特性。^[26]

SP3238 的内部逻辑图（如图 21）：

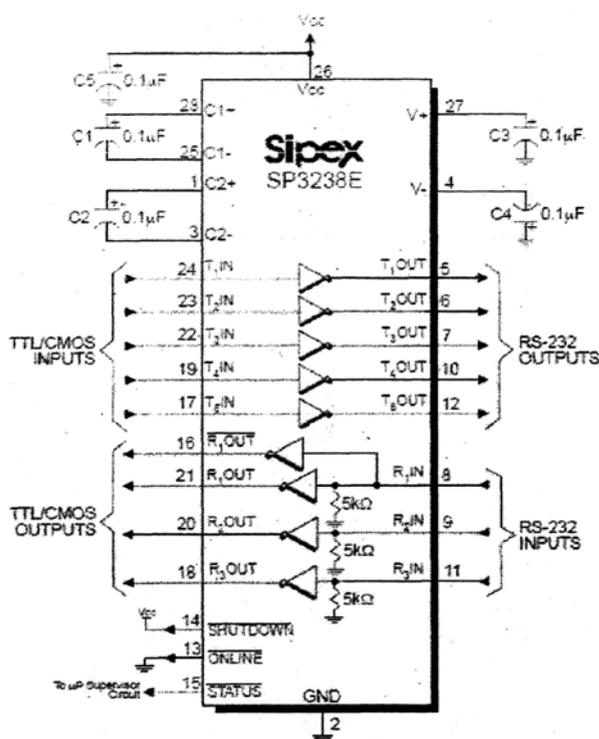


图 21 SP3238 内部逻辑图

3.2.4.2 RS232 线驱动器/接收器 MAX3243

MAX3243 产品是由德州仪器公司 (TI) 推出的一款兼容 RS232 标准的芯片。该器件包含 3 驱动器、5 接收器和双电荷泵电路，并带有管脚±15kV ESD 保护。

该器件符合 TIA/EIA-232-F 标准，它所包含的驱动器和接收器数目完全与标准的 IBM PC/AT 串口一致。电荷泵电路仅仅需要 4 个很小的外部电容就能工作在 3V~5V 的电路中。另外，该器件还提供一个永远激活状态的非反相输出 (ROUT2)，当芯片掉电时它能用作振铃检测。该器件的最大传输数据速率可达 250kbit/s 和最大 30V/us 的输出斜率。^[27]

MAX3243 的内部逻辑图（如图 22）：

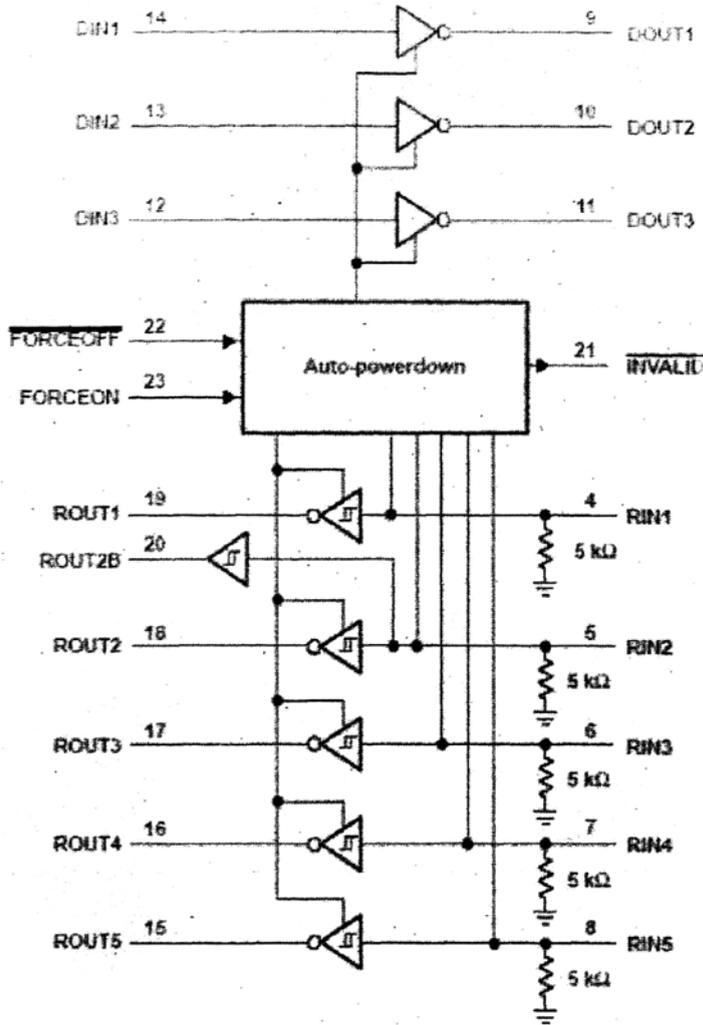


图 22 MAX3243 内部逻辑图

3.2.4.3 RS232 通信模块原理图

RS232 通信模块的原理图（如图 23）：

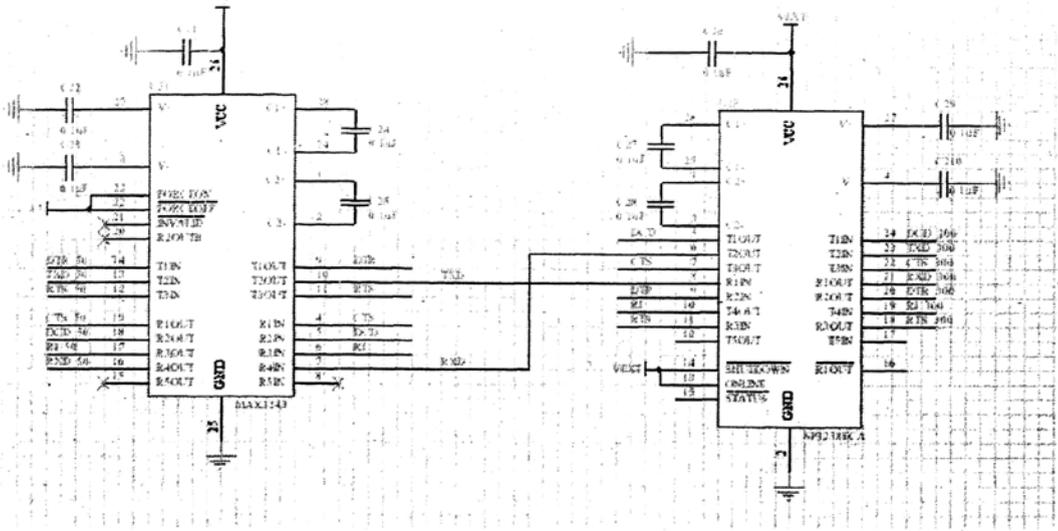


图 23 RS232 通信模块原理图

如图所示，信息从 MMU 模块过来以后，先进入 MAX3243，进行电平转换后进入 SP3238，然后再送到 SIM300。

3.2.5 供电电路模块

嵌入式车载监控终端共需要 5V、3.3V 和 4.2V 三种直流稳压电源，其中 GPRS 模块 SIM300 使用 4.2V 电源，SP3238 使用 3.3V 电源，ATmega128 和绝大多数芯片使用 5V 电源。电源模块设计时采用多个分离元件来实现具体设计。因为在本设计中 Atmega128 处理器应用在车载场合，而汽车上提供的电源主要有 12V 和 24V 两种直流电源，本系统中选用 12V 作为为车载电源，经稳压电路将电压降至 5V、4.2V 输出，3.3V 由低压差线性稳压器（LDO）通过+5V 调整输出得到。供电电路模块原理图（如图 24）：

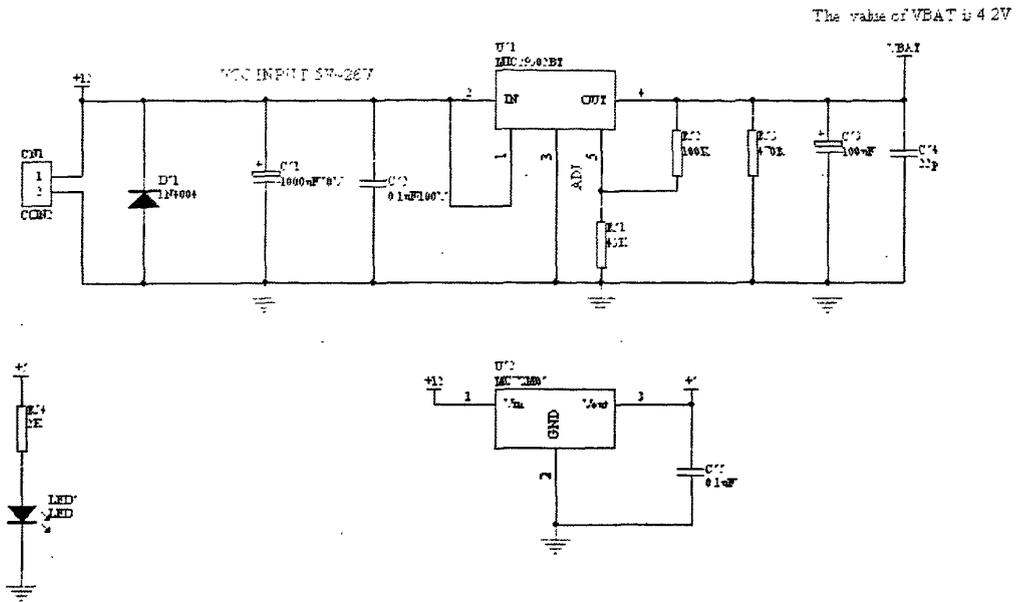


图 24 供电电路模块原理图

如图所示，采用了 MICREL 公司的电源转换芯片 MIC29032 来实现 12V 到 4.2V 电源的转换。MIC29032 是一种支持宽幅输入电压和输出可调电压的电源芯片，在稳压器的外围配置一定阻值的电阻，即可使输出可调，根据公式 $V_{OUT} = 1.240V \times [1 + (R1 / R2)]$ 就可以算出所需的电阻。

12V 转 5V 我们直接采用 MC78M05 进行转换，输出与地之间添加一个退藕电容即可。

3.2.6 PCB 设计原则

要使电子电路获得最佳性能，元器件的布局及导线的布设是很重要的。为了设计质量好、造价低的 PCB，应遵循以下一般原则^[28,29,30]：

(1) 布局 首先，要考虑 PCB 尺寸大小。PCB 尺寸过大时，印制线条长，阻抗增加，抗噪声能力下降，成本也增加；过小，则散热不好，且邻近线条易受干扰。其次，在确定 PCB 尺寸后，再确定特殊元件的位置。最后，根据电路的功能单元，对电路的全部元器件进行布局。尽可能缩短高频元器件之间的连线，设法减少它们的分布参数和相互间的电磁干扰。易受干扰的元器件不能相互挨得太近，输入和输出元件应尽量远离。某些元器件或导线之间可能有较高的电位差，应加大它们之间的距离，以免放电引发意外短路。带高电压的元器件应尽量布置在调试时手不易触及的地方。那些又大又重、发热量多的元器件，不宜装在印刷板上，而应装在整机的机箱底板上，且应考虑散热问题。热敏元件应远离发热元件。

位于电路板边缘的元器件，离电路板边缘一般不小于 2mm。电路板的最佳形状为矩

形，长宽比为 3:2 或 4:3。电路板面尺寸大于 200×150mm 时，应考虑电路板所受的机械强度。

(2) 布线 印制导线的最小宽度主要由导线与绝缘基板间的粘附强度和流过它们的电流值决定。当铜箔厚度为 0.05mm，宽度为 1~15mm 时，通过 2A 的电流，温度不会高于 300℃，因此导线宽度为 1.5mm 可满足要求。对于集成电路，尤其是数字电路，通常选 0.02~0.3mm 导线宽度。当然，只要允许，还是尽可能用宽线，尤其是电源线和地线。印制导线的拐弯处一般取圆弧形，直角或夹角在高频电路中会影响电气性能。此外，尽量避免使用大面积铜箔。如要大面积铜箔时，最好用栅格状，这样有利于排除铜箔与基板间粘合剂受热产生的挥发性气体。

(3) 焊盘 要比器件引线直径大一些，焊盘太大易形成虚焊。焊盘外径 D 一般不小于 $(d+1.2)$ mm，其中 d 为引线孔径。对高密度的数字电路，焊盘最小直径可取 $(d+1.0)$ mm。

(4) PCB 及电路抗干扰措施 电源线设计：根据印刷线路板电流的大小，尽量加粗电源线宽度，减少环路电阻。同时使电源线、地线的走向和数据传输的方向一致，这样有助于增强抗噪声能力。地线设计：若线路板上既有逻辑电路又有线性电路，应使它们尽量分开。低频电路的地应尽量采用单点并联接地，实际布线有困难时可部分串联后再并联接地。高频电路宜采用多点串联接地，地线应短而粗，高频元件周围尽量用栅格状的大面积地箔。接地线应尽量加粗，如有可能，接地线应在 2~3mm 以上。接地线构成闭环路，其接地电路布成闭环路大多能提高抗噪声能力。去藕电容配置：去藕电容的一般配置原则是：

- a) 电源输入端跨接 10~100 uF 的电解电容器。如有可能，接 100 uF 以上的更好。
- b) 原则上每个集成电路芯片都应布置一个 0.01 pF 的瓷片电容，如遇印刷板空隙不够，可每 4~8 个芯片布置一个 1~10 pF 的钽电容。
- c) 对于抗噪能力弱、关断时电源变化大的器件，如 RAM，ROM 存储器件，应在芯片的电源线和地线之间直接接入去藕电容。
- d) 电容引线不能太长，尤其是高频旁路电容不能有引线。

第四章 软件设计

4.1 软件总体设计

- 1、通过 GPRS 回传终端 ID 号和位置信息，要求终端能设置唯一的 ID 号；
- 2、开机后能实现自动拨号，终端便工作于 GPRS 在线状态，只要设置的 GPRS 中心 IP 地址有效且处于 GSM 信号范围，即可实现与中心进行数据通信；
- 3、掉线后能实现自动重拨，保证终端的永远在线；
- 4、终端需要有自主回传、中心呼叫回传、停止回传三种模式，中心可以远程控制切换；终端默认为自主回传模式，中心向终端发送呼叫回传模式的指令，该终端即进入了中心呼叫回传模式，此模式下，中心发出一个指令，终端收到后才回传自己的位置信息；
- 5、自主回传模式下采用固定时间间隔回传，间隔可由中心远程设定，自主回传模式下，车辆移动距离必须大于设定值才回传；
- 6、中心可以远程设定回传 IP，可远程控制终端自动重启；
- 7、终端需要对数据进行压缩和精简，最大限度压缩传输数据大小，以节省开支，可以采用自定义协议；
- 8、终端需要有数据缓存功能，GPRS 离线或阻塞时，自动保存固定间隔的轨迹，GPRS 恢复通信联通时再自动补发，以确保 GPRS 阻塞时不会引起数据中断；
- 9、GPRS 终端需要支持 TCP/UDP/IP/PPP 多种协议，并能完成协议处理；
- 10、终端有 GPRS 和 GPS 天线接口，有相应的网络状态指示灯，数据输入输出指示灯。

4.2 GPS 信息解析

GPS 0183 协议 GGA 和 RMC 的解释

GPRMC

例: \$GPRMC, 085223.136, A, 3957.6286, N, 11619.2078, E, 0.06, 36.81, 180908...A*57

字段 0: \$GPRMC, 语句 ID, 表明该语句为 Global Positioning Recommended Minimum

Specific GPS/TRANSIT Data (RMC) 推荐最小定位信息

字段 1: UTC 时间, hhmmss.sss 格式

字段 2: 状态, A=定位, V=未定位

字段 3: 纬度 ddmn.mmmmm, 度分格式 (前导位数不足则补 0)

字段 4: 纬度 N (北纬) 或 S (南纬)

字段 5: 经度 dddmm.mmmmm, 度分格式 (前导位数不足则补 0)

字段 6: 经度 E (东经) 或 W (西经)

字段 7: 速度, 节, Knots

字段 8: 方位角, 度

字段 9: UTC 日期, DDMMYY 格式

字段 10: 磁偏角, (000 - 180) 度 (前导位数不足则补 0)

字段 11: 磁偏角方向, E=东 W=西

字段 16: 校验值

\$GPGGA

例: \$GPGGA, 085223.136,3957.6286,N,11619.2078,E,1,03,4.0,6.5,M,-6.5,M,0.0,0000*63

字段 0: \$GPGGA, 语句 ID, 表明该语句为 Global Positioning System Fix Data (GGA)

GPS 定位信息

字段 1: UTC 时间, hhmmss.sss, 时分秒格式

字段 2: 纬度 ddmm.mmmm, 度分格式 (前导位数不足则补 0)

字段 3: 纬度 N (北纬) 或 S (南纬)

字段 4: 经度 dddmm.mmmm, 度分格式 (前导位数不足则补 0)

字段 5: 经度 E (东经) 或 W (西经)

字段 6: GPS 状态, 0=未定位, 1=非差分定位, 2=差分定位, 3=无效 PPS, 6=正在

估算

字段 7: 正在使用的卫星数量 (00 - 12) (前导位数不足则补 0)

字段 8: HDOP 水平精度因子 (0.5 - 99.9)

字段 9: 海拔高度 (-9999.9 - 99999.9)

字段 10: 地球椭球面相对大地水准面的高度

字段 11: 差分时间 (从最近一次接收到差分信号开始的秒数, 如果不是差分定位将为空)

字段 12: 差分站 ID 号 0000 - 1023 (前导位数不足则补 0, 如果不是差分定位将为空)

字段 13: 校验值

4.3 GPRS/GPS 终端通信协议

1: 终端启动后根据当前设置的服务器 IP 和端口, 使用 Tcp 协议连接服务, 连接后发送登录包, 协议中统一发送二进制数, 所有的数据比如 2 字节的回传时间间隔, 或者 4 字节的经纬度, 均采用 Little Endian 格式, 也就是低字节存放在内存的低位, 在发送数据包以前, 先做判断, 判断该包数据中是否有 0x1a, 如果没有, 则直接发送。如果有 0x1a, 则找该包数据中不包含的字符代替, 并在包尾增加这个字符, 说明是由该字符代替了 0x1a。该包数据中的命令字也发生相应变化, 在原来基础上加上 0x20。

登录包格式 (共 10 个字节), 终端->中心:

用数据包长度作包头	命令字	终端 ID	是否是自主回传模式	自主回传时间间隔 (单位秒)	心跳包发送间隔 (单位秒)	校验和
9 (包长度, 不包含自己, 1 个字节)	0x11 (1 个字节)	(2 个字节)	(1 个字节, 0:不是, 1:是)	(2 个字节)	(2 个字节)	前面的所有信息值进行异或, 包括包头 (1 个字节)

例如: 09 ^ 1a ^ 01 ^ 00 ^ 01 ^ 08 ^ 07 ^ 58 ^ 02=46, 1 号终端 30 分钟自主回传一次, 心跳包发送间隔为 10 分钟, 所发送的登录包校验和为 0x46。

如果校验和出错, 终端则认为该数据包无效, 终端复位, 重新连接。

2: 中心收到终端的登录请求, 记录下终端 ID, 并回传登录确认包给终端, 然后终端进入正常工作模式,

如果当前开启自主回传功能则按照设置的间隔时间自动回传位置包。

登录确认包 (共 5 个字节), 中心->终端:

包长度	命令字	终端 ID	校验和
4 (包长度, 1 个字节)	0x21 (1 个字节命令字)	(2 个字节)	前面的所有信息值进行异或, 包括包头 (1 个字节)

3: 回传位置 (不区分是自主回传还是呼叫回传, 默认自主回传的时间间隔为 30 分钟), 信息包格式 (共 16 个字节)

终端->中心:

包长度	命令字	终端 ID	位置报	校验和
15 (包长度, 1 个字节)	0x1b (1 个字节命令字)	(2 个字节)	(纬度[32bit], 经度[32bit], 航速[12bit], 航向[12bit], 共 11 个字节)	前面的所有信息值进行异或, 包括包头 (1 个字节)

位置报：11 个字节

内容	所占位数	取值范围	说明
纬度	32	-90 - +90	用 1/10000 分表示的纬度（北=+，南=-）负数用其补码表示
经度	32	-180 - +180	用 1/10000 分表示的经度（东=+，西=-）负数用其补码表示
对地航速	12	0-1022	以 1/10 节距为单位，1023=无；1022= 102.2 节。
对地航向	12	0-3599	以 1/10 度表示的航向

4: 自主回传模式开启或关闭命令，该命令用于开启或关闭自主回传（共计 6 个字节），中心->终端

包长度	命令字	终端 ID	开启或关闭自主回传	校验和
5 (包长度, 1 个字节)	0x2c (1 个字节命令字)	(2 个字节)	(1 个字节, 1-开启, 0-关闭)	前面的所有信息值进行异或, 包括包头 (1 个字节)

终端收到中心的设置自主回传模式开启或关闭命令后，将给中心一个回复，用于中心判断此命令是否设置成功。

(共计 6 个字节)，终端->中心

包长度	命令字	终端 ID	当前是否自主回传	校验和
5 (包长度, 1 个字节)	0x1c (1 个字节命令字)	(2 个字节)	(1 个字节, 1-开启了自主回传, 0-关闭了自主回传)	前面的所有信息值进行异或, 包括包头 (1 个字节)

5: 呼叫回传命令，中心呼叫终端，立刻回传位置包（共计 5 个字节），中心->终端

包长度	命令字	终端 ID	校验和
4 (包长度, 1 个字节)	0x2b(1 个字节命令字)	(2 个字节)	前面的所有信息值进行异或, 包括包头 (1 个字节)

位置包格式和自主回传的信息包格式一样 终端->中心:

包长度	命令字	终端 ID	位置报	校验和
15 (包长度, 1 个字节)	0x1b(1 个字节命令字)	(2 个字节)	(纬度 [32bit], 经度 [32bit], 航速 [12bit], 航向 [12bit], 共 11 个字节)	前面的所有信息值进行异或, 包括包头 (1 个字节)

6: 中心设置自动回传间隔命令
(共计 7 个字节), 中心->终端

包长度	命令字	终端 ID	时间间隔 (单位 s)	校验和
6 (包长度, 1 个字节)	0x2d (1 个字节命令字)	(2 个字节)	设置的回传间隔 (2 个字节)	前面的所有信息值进行异或, 包括包头 (1 个字节)

设置回传间隔时间成功后, 终端回复确认命令给中心
(共计 7 个字节), 终端->中心

包长度	命令字	终端 ID	当前的回传间隔(单位 s)	校验和
6 (包长度, 1 个字节)	0x1d(1 个字节命令字)	(2 个字节)	当前的回传间隔 (2 个字节)	前面的所有信息值进行异或, 包括包头 (1 个字节)

7: 中心设置心跳包发送间隔
(共计 7 个字节), 中心->终端

包长度	命令字	终端 ID	心跳包发送间隔(单位 s)	校验和
6 (包长度, 1 个字节)	0x2e (1 个字节命令字)	(2 个字节)	设置的心跳包发送间隔 (2 个字节)	前面的所有信息值进行异或, 包括包头 (1 个字节)

终端回复设置心跳包发送间隔命令

(共计 7 个字节), 终端->中心

包长度	命令字	终端 ID	当前间隔(单位 s)	校验和
6 (包长度, 1 个字节)	0x1e (1 个字节命令字)	(2 个字节)	当前的心跳包发送间隔 (2 个字节)	前面的所有信息值进行异或, 包括包头 (1 个字节)

8: 中心发送重启指定给终端

(共计 5 个字节), 中心->终端

包长度	命令字	终端 ID	校验和
4 (包长度, 1 个字节)	0x2f (1 个字节命令字)	(2 个字节)	前面的所有信息值进行异或, 包括包头 (1 个字节)

终端收到中心发送的重启命令后, 首先发送一个确认包给中心, 然后会在 10s 后重启终端

(共计 5 个字节), 终端->中心

包长度	命令字	终端 ID	校验和
4 (包长度, 1 个字节)	0x1f (1 个字节命令字)	(2 个字节)	前面的所有信息值进行异或, 包括包头 (1 个字节)

9: 中心发送设置 IP 和端口的指令给终端

(共计 11 个字节)，中心->终端

包长度	命令字	终端 ID	新的 IP	新的端口	校验和
10 (包长度, 1 个字节)	0x20 (1 个字节命 令字)	(2 个 字节)	新的 IP (4 字节)	新的端口 (2 字节)	前面的所有信息值 进行异或, 包括包头 (1 个字 节)

终端在收到中心设置的新 IP 和端口后，会发送一个确认包给中心，然后 30s 内关闭当前连接并尝试使用新的 IP 和端口进行连接，如果在 5 分钟内没有连接成功，则重新使用老的 IP 和端口进行连接，也就是说只有当新的 IP 和端口真正连接上了，才能替换老的，并保存。

如果在尝试连接的过程中断电，则重启后还是使用老的 IP 和端口进行连接。

另外，IP 的 4 个字节使用这种方式存储。

比如 IP:192.168.0.1 存储为

pStart[0] = 192;

pStart[1] = 168;

pStart[2] = 0;

pStart[3] = 1;

也就是 IP 中前面的字节放在低位

(共计 11 个字节)，终端->中心

包长度	命令字	终端 ID	设置的新 IP	设置的新端口	校验和
10 (包长度, 1 个字节)	0x10 (1 个 字节命令字)	(2 个 字节)	设置的新 IP (4 字节)	设置的新 端口 (2 字 节)	前面的所有信息值 进行异 或, 包括 包头 (1 个字 节)

10: 心跳包，终端在登录并收到中心发送的登录确认后，按照设定的心跳包发送间隔发送心跳包，中心收到心跳包后会给终端一个回复。

终端如果在 2 个心跳包发送时间间隔内都没有收到一次中心的确认包，则认为该连接中断，要重新连接。

心跳包的格式（共计 1 个字节）

心跳检测间隔（以分钟为单位，缺省为 10 分钟），终端=> 中心

00000000
(1 字节)

中心接收到心跳包以后，也向终端回传确认包，中心=> 终端

00000000
(1 字节)

附表：各命令字列表

命令字	内容	数据流向	备注
0x11	登录包	终端=> 中心	终端启动后发送登录包
0x21	登录包确认	中心=> 终端	中心收到终端的登录请求，记录下终端 ID,并回传登录确认包给终端
0x1b	回传位置	终端=> 中心	呼叫回传命令，中心呼叫终端，立刻回传位置包
0x2b	呼叫回传	中心=> 终端	呼叫回传命令，中心呼叫终端，立刻回传位置包
0x2c	自主回传模式开启或关闭	中心=> 终端	该命令用于开启或关闭自主回传
0x1c	终端回复设置自主回传的时间间隔	终端=> 中心	终端收到中心的设置自主回传模式命令后将给中心一个回复
0x2d	中心设置自动回传时间间隔	中心=> 终端	中心设置自动回传时间间隔（以秒为单位，范围：0~65535s）
0x1d	终端回复设置自动回传间隔	终端=> 中心	终端回复设置自动回传间隔命令

0x2e	中心设置心跳包发送时间间隔	中心=> 终端	中心设置心跳包发送时间间隔
0x1e	终端回复设置心跳包发送间隔	终端=> 中心	终端回复设置心跳包发送间隔命令
0x2f	中心发送重启指定给终端	中心=> 终端	中心发送重启指定给终端命令字
0x1f	终端回复中心发送的重启命令	终端=> 中心	终端收到中心发送的重启命令后, 首先发送一个确认包给中心, 然后会在 10s 后重启终端
0x20	中心发送设置 IP 和端口的指令给终端	中心=> 终端	中心发送设置 IP 和端口的指令给终端
0x10	终端回复中心设置的新 IP 和新端口	终端=> 中心	终端在收到中心设置的新 IP 和端口后, 会发送一个确认包给中心, 然后 30s 内关闭当前连接并尝试使用新的 IP 和端口进行连接
0x00	心跳包	终端《=》中心	维持 GPRS 链路 心跳包, 终端在登录并收到中心发送的登录确认包后, 按照设定的心跳包发送间隔发送心跳包, 中心收到心跳包后会给终端一个回复 终端如果在 2 个心跳包发送时间间隔内都没有收到一次中心的确认包, 则认为该连接中断, 要重新连接

数据包中出现 0x1a 的解决方法:

在发送数据包以前, 先做判断, 判断该包数据中是否有 0x1a, 如果没有, 则直接发送。如果有 0x1a, 则找该包数据中不包含的字符代替, 并在包尾增加这个字符, 说明是由该字符代替了 0x1a。该包数据中的命令字也发生相应变化, 在原来基础上加上 0x20。即在原来命令字的基础上扩展出一些命令字, 这些扩展的命令字是在原来的基础上加上 0x20, 代表该包数据中含有 0x1a, 其他意义不变。

例如: 终端回传的位置包信息中包含 0x1a, 则格式如下:

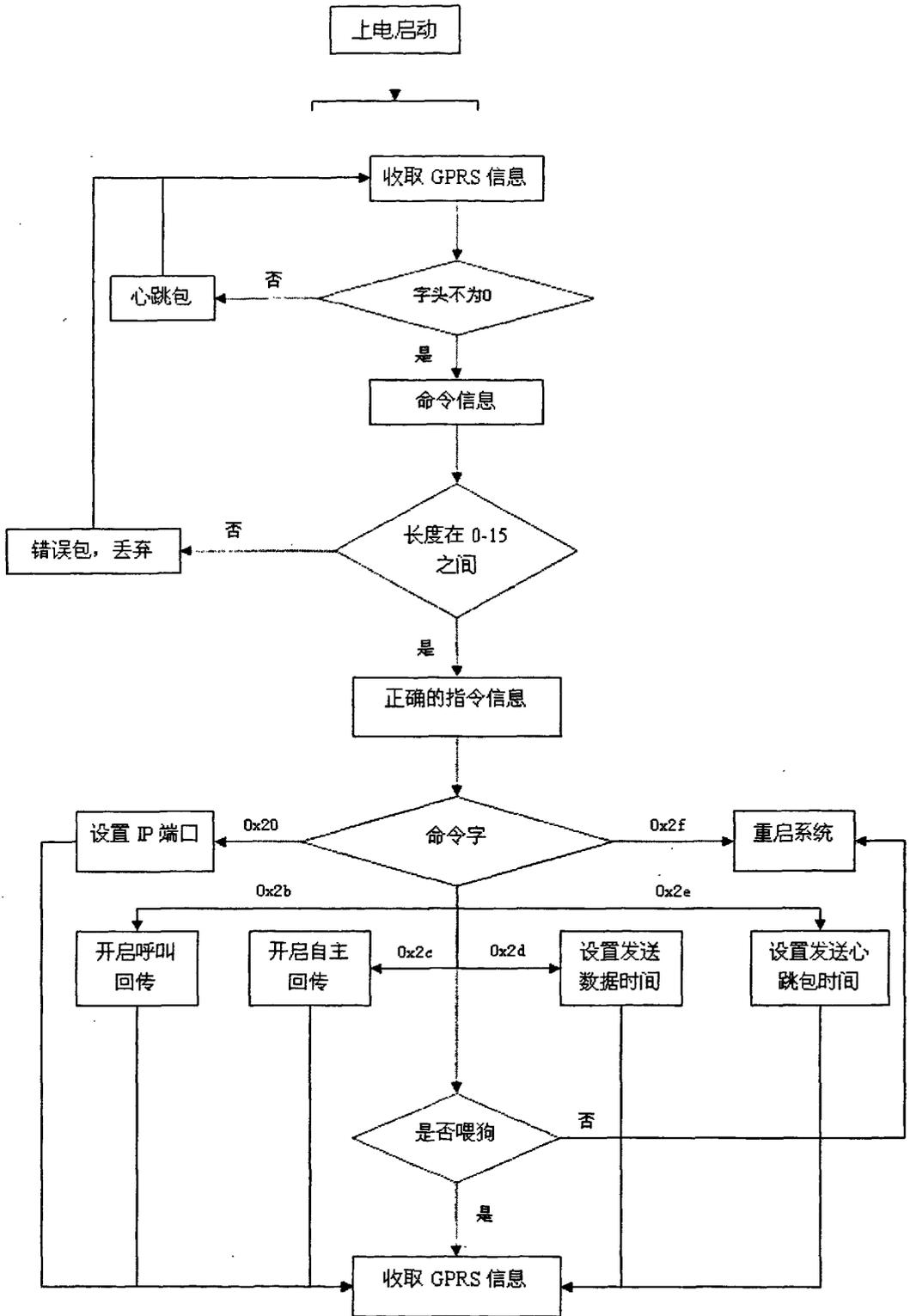
包长度	命令字	终端 ID	位置报	替 换 0x1a 的 字节	校验和
16 (包长度, 1 个字节)	0x3b (1 个字节 命令字)	(2 个字 节)	(纬度 [32bit], 经度 [32bit], 航速[12bit], 航 向[12bit], 共 11 个字 节)	(1 个 字节)	前面的所有 信息值进行 异或, 包括包 头 (1 个字节)

中心发送设置 IP 和端口的指令中含有 0x1a, 则格式如下:

包长度	命令字	终端 ID	新的 IP	新的端口	替 换 0x1a 的 字节	校验和
10 (包长 度, 1 个 字节)	0x40 (1 个字节命 令字)	(2 个字 节)	新的 IP(4 字节)	新的端口 (2 字节)	(1 个 字节)	前面的所有 信息值进行 异或, 包括包 头 (1 个字节)

4.4 软件程序说明

系统的简要工作流程是这样的, 系统加电以后, 先进行硬件的初始化, 同时加载硬件的启动参数, 然后就进行 GPRS 连接, 在经过测试连接证明链接到网络以后, 就进入主程序进行主循环, 不通就重新连接, 直到连接成功再进入主循环程序, 流程图如图 25 所示:



的字
的丢
代表
行了

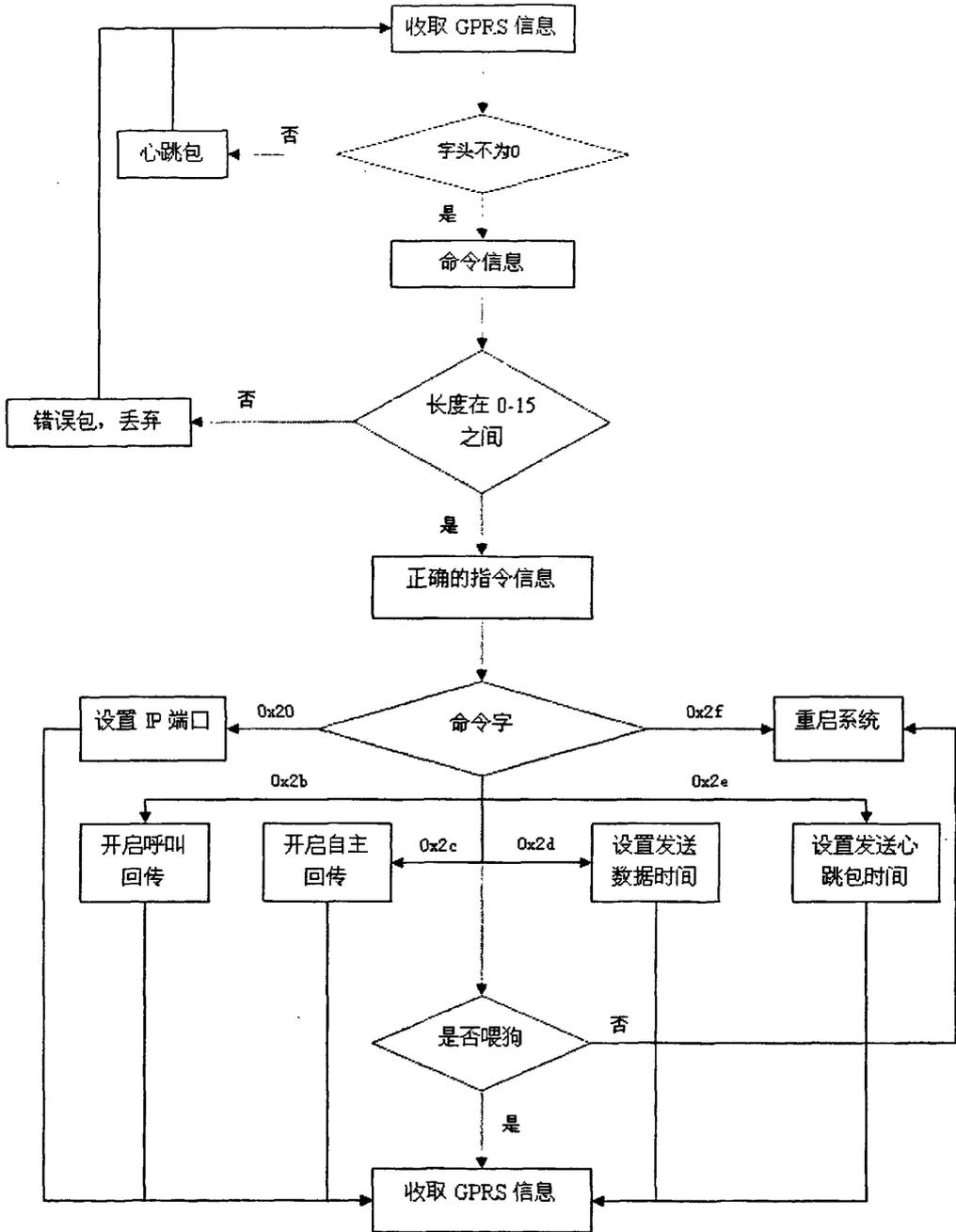


图 26 主循环程序流程图

下面是主程序源代码:

```

void main_loop_process (void)
{
    unsigned char n ;

```

```
unsigned char len;
while (1)
{
    memset (recv_buf, 0, 32) ;
    n = 0;
    n = GPRS_Receive (recv_buf,10) ;

    if (n)
    {
        len = recv_buf[0];
        if (len == 0)
        {
            //收到心跳包
            connect_ok = 1;
            heart_count = 0;
            continue;
        }
        else if ( (len < 0) || (len > 15) )
        {
            //watchdog_start ();
            continue;
        }
        if (CalcChecksum (recv_buf,len) != recv_buf[len])
        {
            continue;
        }
        switch (recv_buf[1])
        {
            case 0x2b: CallSendData () ;break;
            case 0x2c: set_atuo_send (recv_buf) ;break;
            case 0x2d: SetSelfSendDataTime (recv_buf) ;break;
            case 0x2e: SetHeartPkgTime (recv_buf) ;break;
            case 0x2f: RestartSystem () ;break;
            case 0x20: SetIpAndPort (recv_buf) ;break;
            default:break;
        }
    }
}
```

```

} //end switch
continue;
} //end if (n)

if (heart_count)
watchdog_start ();
SelfSendData ();
SendHeartPkg ();

} //end while
}

```

软件程序部分另外一个比较重要的地方就是GPRS部分代码的编写,共有4个子程序,分别为GPRS连接、断开GPRS连接、GPRS数据发送以及GPRS数据接收,在这部分子程序编写的时候需要了解一些关于SIM300 AT指令的知识,下面简要介绍一下AT指令。

下面先列表介绍一下常用命令:^[31]

命令	功能描述
AT+CIPSTART	连接到服务器
AT+CIPSEND	发送数据到服务器
AT+CIPCLOSE	关闭TCP/UDP连接
AT+CIPSHUT	关闭移动场景
AT+CLPORT	设置本地端口号
AT+CSTT	启动任务并设置APN, user name, password
AT+CIICR	激活移动场景
AT+CIFSR	获得本地IP地址
AT+CIPSTATUS	查询目前状态
AT+CDNSCFG	配置DNS的IP地址
AT+CDNSGIP	域名解析, 获取IP地址
AT+CDNSORIP	设置连接域名或IP地址
AT+CIPHEAD	设置接收的数据头
AT+CIPAT\$	设置自动发送时间
AT+CIPSPRT	设置在AT+CIPSEND后是

	否跟有发送提示符>
AT+CIPSERVER	配置为服务器
AT+CIPCSGP	设置为CSD或GPRS连接以及相关参数
AT+CIPCCON	设置当前的连接
AT+CIPFLP	是否固定本地端口号
AT+CIPSRIP	设置是否显示发送方的IP地址和端口号

下面详细介绍一下几个 AT 指令：

1、AT+CIPSTART :

AT+CIPSTART 建立TCP连接或注册UDP端口号	
执行命令 AT+CIPSTART = (“TCP”, “UDP”) , (“IP ADDRESS”, “DOMAIN NAME”) , “PORT”	响应 命令正确返回OK 错误返回ERROR
参数:	对TCP连接: AT+CIPSTART=“TCP”, “IP ADDRESS/DOMAIN NAME”, “PORT” SERVER可以是IP地址或者已在 INTERNET上注册过的域名 命令正确返回OK, 错误返回ERROR 连接成功返回CONNECT OK 失败先返回STATE 状态码, 后再返 回 CONNECT FAIL 对UDP连接: AT+CIPSTART=“UDP”, “IP ADDRESS/DOMAIN NAME”, “PORT” SERVER可以是IP地址或者已在 INTERNET上注册过的域名

	<p>命令正确返回OK，错误返回ERROR 连接成功返回CONNECT OK 失败先返回STATE 状态码，后再返回 CONNECT FAIL 在ATV0设置下，只返回状态码0-8</p>
<p>参考</p>	<p>STATE 状态说明</p> <p>0 IP INITIAL 初始化</p> <p>1 IP START 启动任务</p> <p>2 IP CONFIG 配置场景</p> <p>3 IP IND 接受场景配置</p> <p>4 IP GPRSACT 场景已激活</p> <p>5 IP STATUS 获得本地IP地址</p> <p>6 TCP/UDP CONNECTING 与 SERVER建立连接阶段</p> <p>7 IP CLOSE 连接已关闭</p> <p>8 CONNECT OK 连接建立成功</p> <p>AT+CIPSTART只有在IP_INITIAL、IP_CLOSE、IP_STATUS状态下才可以进行连接的建立，连接建立成功后的状态为CONNECT OK；连接失败，则处在失败的状态上。</p>

2、AT+CIPSEND

<p>AT+CIPSEND 发送数据</p>	
<p>执行命令</p> <p>AT+CIPSEND</p> <p>返回"> "后写入需发送的数据，按CTRL+Z开始发送</p>	<p>响应</p> <p>此命令用来发送可变长度的数据</p> <p>连接未建立或已被断开，返回ERROR</p>

	<p>如果数据发送成功， 返回SEND OK</p> <p>如果数据发送失败， 返回 SEND FAIL:<error no></p>
<p>设置指令</p> <p>AT+CIPSEND=<data length></p>	<p>响应</p> <p>此命令用来发送固定长度的数据 连接未建立或已被断开， 返回 ERROR</p> <p>如果数据发送成功， 返回SEND OK 如果数据发送失败， 返回 SEND FAIL:<error no></p> <p>参数</p> <p><data length> 需要一次发送的数据长度</p>
<p>参考</p>	<p>Note</p> <p>注： 1.每次传送的数据字节数应 ≤1024 bytes。</p> <p>2.通过AT+CIPATS可以在设定的时间内自动发送数据。</p> <p>3.只有在连接已建立的状态下才可发送数据， 否则返回ERROR。</p>

3、AT+CIPCLOSE

AT+CIPCLOSE 关闭TCP或UDP连接	
<p>执行命令</p> <p>AT+CIPCLOSE</p>	<p>响应</p> <p>关闭成功返回CLOSE OK</p> <p>关闭失败返回ERROR</p>
<p>参考</p>	<p>AT+CIPCLOSE 只有在 TCP/UDP CONNECTING或CONNECT OK 状态下才关闭连接， 否则认为关闭失败返回 ERROR， 关闭后的状态为 IP CLOSE。</p>

4、AT+CIFSR

AT+CIFSR 获得本地IP地址	
执行命令 AT+CIFSR	响应 命令正确返回本地 IP 地址 错误返回 ERROR
参考	只有在移动场景已激活的状态：IP GPRSACT、TCP/UDP CONNECTING、CONNECT OK、IP CLOSE下才可以AT+CIFSR查询得到本地IP地址，否则返回ERROR。 只有在状态为 IP GPRSACT 下才将状态改变为 IP STATUS，既已获得本地 IP 地址。

4、AT+CDNSCFG

AT+CDNSCFG 配置DNS	
执行命令 AT+CDNSCFG=<pri_dns>,<sec_dns>]	响应 命令正确返回 OK 参数 <pri_dns> Primary DNS IP Address <sec_dns> Secondary DNS IP Address

6、AT+CIPHEAD

AT+CIPHEAD 设置接收数据的IP头	
查询命令 AT+CIPHEAD?	响应 此命令用来查询是否设置了 IP头 <mode>
测试命令	响应

AT+CIPHEAD=?	+CIPHEAD: (0-NO HEADER,1-ADD HEADER)
设置命令 AT+ CIPHEAD =<mode>	响应 此命令用来设置是否在接收数据时增加IP头提示。 命令正确返回 OK 命令错误返回 ERROR <mode> 0 未设置 IP 头 1 设置 IP 头 IP 头格式为: ”+IPD (data length) :”

7、AT+ CIPCSGP

AT+CIPCSGP 设置为CSD或GPRS连接	
查询命令 AT+CIPCSGP?	响应 <mode>
测试命令 AT+CIPCSGP=?	响应 +CIPCSGP: 0-CSD,DIAL NUMBER,USER NAME,PASSWORD,RATE (0,3) +CIPCSGP: 1-GPRS,APN,USER NAME,PASSWORD
设置命令 AT+ CIPCSGP=<mode>,[<apn>,< user ID>,<pwd>,<dial num>,<user ID>,<pwd>,<rate>]	响应 此命令配置为CSD或GPRS连接。 <mode> 0 CSD 连接 1 GPRS 连接 CSD 连接下要设置的参数:

	<p><dial num> 拨打号码默认为 17201</p> <p><user ID> 用户名 默认为 172</p> <p><pwd> 密码默认为 172</p> <p><rate> 连接速率 0 2400bps 1 4800bps 2 9600bps 3 14400bps 默认为 9600bps</p> <p>GPRS 连接下要设置的参数:</p> <p><apn> 访问接入点 默认为 CMNET</p> <p><user ID> 用户名 默认为空</p> <p><pwd> 密码 默认为空</p>
--	--

以上是 SIM300 一些常用的 AT 指令介绍，熟悉了 AT 指令之后，对 GPRS 部分的编程才能做到比较熟练。下面给出 GPRS 部分的源代码，代码如下：

```

/*****
//SIM300 硬件复位
void Init_GPRS_Reset (void)
{
    PORTB &= ~BIT ( PB4);          // 低电平复位
    DelayXms (5000);
    DelayXms (5000);
    PORTB |= BIT ( PB4);
}
*****/

/*****GPRS 连接*****/
unsigned char GPRS_Connect_Tcp ( void )
{
    unsigned char *n,i;

```

```

unsigned char con_cmd0[] = "AT\r";
unsigned char con_cmd1[42] = "AT+CIPSTART=\"TCP\",\"";

unsigned char con_cmd2[] = "AT+CIPCLOSE\r";

unsigned char ans0[] = "OK\r\n";
unsigned char ans1[] = "CONNECT OK";
unsigned char ans2[] = "ALREADY CONNECT";
unsigned char ans3[] = "CLOSE OK";
strcat ( con_cmd1,serv_ip_str ) ;
strcat ( con_cmd1,"\"," ) ;
strcat ( con_cmd1,serv_port_str ) ;
strcat ( con_cmd1,"\r" ) ;
GPRS_TXD ( con_cmd0, 3000 ) ;           //发 AT 回车变更波特率 9600
n = search_reply ( Recive_buff,ans0 ) ; //
if ( n == 0 ) return 1;                 //与 SIM300 模块通信失败, 错误返回
GPRS_TXD ( con_cmd1, 8000 ) ;           //向远端服务器发起连接
n = search_reply ( Recive_buff,ans1 ) ; //找 CONNECT OK
if ( n == 0 ) {                          //n=0,无 CONNECT OK
    n = search_reply ( Recive_buff,ans2 ) ; //再找 ALREADY CONNECT
    if ( n == 0 ) {                        //均未找到
        GPRS_TXD ( con_cmd2, 5000 ) ;     //发断开命令
        n = search_reply ( Recive_buff,ans3 ) ; //
        return 2;                          //与远端服务器连接失败, 错误返回
    }
}
return 0; //返回码=0, 连接成功
}

```

//*****断开GPRS连接*****//

```

unsigned char GPRS_Disconnect ( void )
{
    unsigned char *n;
    unsigned char con_cmd0[] = "AT\r";
    unsigned char con_cmd1[] = "AT+CIPCLOSE\r";

```

```

unsigned char con_cmd2[] = "AT+CIPSHUT\r";

unsigned char ans0[] = "\r\nOK";
unsigned char ans1[] = "CLOSE OK";
unsigned char ans2[] = "SHUT OK";

GPRS_TXD ( con_cmd0, 3000 );          //发 AT 回车变更波特率 9600
n = search_reply ( Recive_buff,ans0 ); //
if ( n == 0 ) return 1;              //与 SIM300 模块通信失败, 错误返回

GPRS_TXD ( con_cmd1, 500 );          //发断开命令
n = search_reply ( Recive_buff,ans1 ); //
if ( n != 0 ) return 0;              //返回码=0, 断开成功
else
{
    GPRS_TXD ( con_cmd2, 500 );      //发断开命令
    n = search_reply ( Recive_buff,ans2 );
    if ( n != 0 ) return 0;
    else return 2;
}
}

/*****GPRS 发送*****/
unsigned char GPRS_Transmit ( unsigned char *str, unsigned char len )
{
    unsigned char *n;
    unsigned char i;
    unsigned char con_cmd3[] = "AT+CIPSEND\r";

    unsigned char ans3[] = ">";
    unsigned char ans4[] = "SEND OK";
    unsigned char tail[] = "\x1a";
    unsigned char transmit_str[32];
    memset ( transmit_str, 0, 32 );

```

```

memcpy ( transmit_str, str, len ) ;
memcpy ( transmit_str + len, tail, 2 ) ;
GPRS_TXD ( con_cmd3, 500 ) ;
n = search_reply ( Recive_buff,ans3 ) ;
if ( n == 0 ) return 3;    //没有发送提示符， 错误返回
gprs_send ( transmit_str, len+1, 1000 ) ;
n = search_reply ( Recive_buff,ans4 ) ;
if ( n == 0 )
{
    DelayXms ( 5000 ) ;
    n = search_reply ( Recive_buff,ans4 ) ;
    memset ( Recive_buff, 0, RECEIVE_buf_len ) ;
    Rec_buf_outp = Rec_buf_inp = 0;
    if ( n == 0 ) return 5;
}
else
{
    memset ( Recive_buff, 0, RECEIVE_buf_len ) ;
    Rec_buf_outp = Rec_buf_inp = 0;
    return 0;//发送成功返回 0
}
return 0;
}

//*****GPRS 接收*****//
unsigned char GPRS_Receive ( unsigned char *receive, unsigned char timeout )
{
    unsigned char n = 0;
    unsigned char i;

    for ( i=0; i<10; i++ )
    {
        DelayXms ( 100 * timeout ) ;

        while ( Rec_buf_outp != Rec_buf_inp )

```

```

{
    *receive++ = Recive_buff[ Rec_buf_outp ];
    n++;
    Rec_buf_outp = ++ Rec_buf_outp & (RECEIVE_buf_len-1) ;
    if (n >= 30)
    {
        //watchdog_start ( ) ;
        break;
    }
}
if (n) break;
}
memset (Recive_buff, 0, RECEIVE_buf_len) ;
Rec_buf_outp = Rec_buf_inp = 0;
return n;
}

```

第五章 总结与展望

5.1 工作总结

本文首先论述了车载监控系统的重要性和我国车载监控系统的研究现状,阐明了开发适合中国国情的车载监控系统的必要性,提出了两种以嵌入式系统、GPS 定位系统、GPRS 无线传输网络为核心的车载监控系统的总体技术方案,并进行比较,从实际出发确定了开发方案。继而对嵌入式操作系统、GPS 定位系统、GPRS 无线传输进行了细致的研究学习。在此基础上,设计开发了以 AVR 单片机 ATmega128 为核心的嵌入式车载终端的硬件平台,然后针对硬件设计进行了嵌入式 C 的软件编写,进行了软件应用程序的开发,实现了监控系统的基本功能。实现的终端主要特点如下:

(1) 系统硬件性能突出:硬件上使用 8 位高端 RISC AVR 单片机 ATmega128,其功能强大,ATmega128 的数据吞吐率高达 1 MIPS/MHz,片内资源丰富,相应的外围电路设计方便,便于实现复杂功能,且可扩展空间具大。

(2) 系统采用 GPRS 进行无线通讯,取代传统的 GSM 通讯方式。GPRS 数据传输的优点:

通讯费用低:车载 GPS 监控系统的通信特点是具有突发性,但数据量小,对采用按流量计费的 GPRS 非常有优势。费用降低了 80%以上,非常具有成本优势。

数据传输效率高:GPRS 是一种新型移动数据通信业务,给移动用户提供高速无线 IP 服务。GPS 设备采集的位置信息经过分析处理后,封装在 IP 报文中进行传输,其最大数据传输效率大于 90%。

强有力的安全措施:GPRS 网络采用 GSM 的多种物理信道加密方式,同时在应用层,还提供中心专线接入、专用 APN 等安全措施,可完全满足运输管理系统对安全的扩展性要求。

可扩展性强:可根据将来业务的需要在 GPRS 上增加新的监控内容,如增加图像传送等等。

可选短信备份:在 GPRS 不通的情况下,可以通过传统的短信备份方式来保证关键信息的传送。

(3) 与新兴的手机定位技术相比,该车载定位终端采用 GPS 定位精度高、定位精度与城乡位置无关。

(4) 系统接口划分清楚,应用程序和系统平台可以分别研发,应用灵活。

(5) 直接在 ICC 下编写和编译 C 语言程序,并通过 AVR Studio 进行仿真,方便、快捷,提高了开发效率。

5.1.1 终端性能测试

(1) 车载定位方面：冷启动时，系统能够在 100 秒内获取车辆的位置、速度、方向等定位信息；热启动时系统可在 15 秒内获取信息；定位误差 3-10 米。

(2) 远程数据传输方面：丢包率极低（小于 5%）；当利用网络协议栈同监控中心通信时，系统能在 60 秒内完成 PPP 链路的建立和自身 IP 地址的获取，实现同监控中心的通信。

(3) 该车载监控终端功能强、运行稳定可靠，能够胜任车辆定位、跟踪等基本工作要求。

5.1.2 系统存在的不足

由于个人力量有限和时间、条件的限制，系统在设计 and 功能上存在诸多不足之处，很多地方需要加以改进，有待于进一步的深入研究。主要有如下几个方面：

(1) 采用了 GPS 单点定位技术。受外界环境如高建筑群、树木等的干扰时，数据会发生较大的跳跃而失效，从而使 GPS 无法连续定位，定位精度下降。可以采用 GPS 技术与航位推算 DR 等组合导航定位技术来解决 GPS 盲区定位问题。

(2) 终端功能目前比较简单，只完成了 GPS 功能的开发和 GPRS 通讯的开发。虽然硬件上预留了许多功能的接口，但还需要进一步的软件开发。

(3) GPS 芯片与电子地图配合使用，更能丰富 GPS 的定位功能。由于时间和资源关系，论文只详细设计了 GPS 定位，而关于 GIS 这一块没有进行深入讨论。

(4) 在服务器端应用软件的开发上，没有投入精力，只是设计了一个小程序进行测试时使用，也没有建立服务器端数据库。

5.2 工作展望

目前我国的汽车电子行业相对国外发达国家还比较落后，存在一定的差距，需要进一步的研究和投入。下一步我们要做的是，在现有硬件平台上做足软件功夫，充分发掘硬件的能力，实现更多应用功能；并在系统中采用组合导航定位技术，诸如 GPS/DR/MM 组合导航定位技术，实现更加精确的导航定位，使之真正成为低成本、高性能、功能丰富的车载终端产品。

【参考文献】

- [1]杨冰. 智能运输系统 [M]. 北京: 中国铁道出版社, 2000.
- [2]赵会平. 汽车导航系统与交通信息通信系统 [M]. 公路交通科技, 1998, 15(3).
- [3]倪建军, 范跃祖, 李松林. 利用 GSM 短消息业务实现 GPS 车辆监控 [M]. 电子技术应用, 2000, (7).
- [4]方圆. 从产品、后装、行业应用向服务、前装、个人应用迈进—探索中国车载 GPS 终端及服务行业发展之路 [M]. 中国集成电路, 2007.11.
- [5]曹冲. 我国 GPS 车辆应用系统的产业化前景分析和市场展望[J]. 全球定位系统网络, 2001.26, (4).
- [6]徐俊仕. 基于嵌入式 Linux 的车载 GPS 导航系统设计与实现[D]. 西北工业大学, 2007.3.
- [7] 孙琼. 嵌入式 Linux 应用程序开发详解[M]. 北京: 人民邮电出版社, 2006.
- [8]毛德操, 胡希明. Linux 内核源代码情景分析(上、下册)[M]. 杭州: 浙江大学出版社, 2001.
- [9] Jonathan Corbet, Alessandro Rubini. Linux 设备驱动程序[M]: (第 2 版). 魏永明等译. 北京: 中国电力出版社, 2004.
- [10]Daniel P.Bovet, Marco Cesati. 深入理解 Linux 内核[M]. 陈莉君等译. 北京: 中国电力出版社, 2001.
- [11] Michael K.Johnson, Erik W.Troan. Linux 应用程序开发[M]: (第 2 版). 武延军, 郭松柳译. 北京: 电子工业出版社, 2005.
- [12] ATmega128 Datasheet, [Z]. ATMEL Co. Ltd, 2005.
- [13] ST16C550 Data Sheet[Z], EXAR Corp , 2000.
- [14] 倪金生, 董宝青, 官小平编著. 导航定位技术理论与实践[M]. 北京: 电子工业出版社, 2007.
- [15]张兰云. 基于 GPS 与 GPRS 的嵌入式车载监控终端的研究与设计[D]. 中国科学技术大学, 2008.
- [16] 黄丁发, 熊永良, 袁林果著. 全球定位系统 (GPS) —理论与实践[M]. 成都: 西南交通出版社, 2006.
- [17] 刘基余. GPS 卫星导航定位原理与方法[M]. 北京: 科学出版社, 2003.
- [18] 边少锋, 李文魁编著. 卫星导航系统概论[M]. 北京: 电子工业出版社, 2005.
- [19] 曹洪太. 基于 Internet 和 GPRS 技术的温室远程监控系统[D]:[硕士]. 曲阜: 曲阜师范大学, 2006.
- [20] Communication Magazine, [J], 1997,(10).

- [21] CaiJian, David J. Goodman. General Packet Radio Service in GSM[J]. IEEE Communications Magazine, 1997.
- [22] 郭梯云, 鄢国扬, 李建东. 移动通信[M]. 西安: 西安电子科技大学出版社, 2005.
- [23] 文志成. 通用分组无线业务—GPRS[M]. 北京: 电子工业出版社, 2004.
- [24] Xavier Lagrange, Philippe Godlewski, Sami Tabbane 著. GSM网络与GPRS[M]. 顾肇基译. 北京: 电子工业出版社, 2002.
- [25] 刘焱. 嵌入式系统接口设计与Linux驱动程序设计[M]. 北京: 北京航空航天大学出版社, 2006.5.
- [26] 甘金明, 邓明建. 基于ARM7的路灯监控通信终端的设计[M]. 微计算机信息. 2008.
- [27] MAX3243 Datasheet, [Z]. MAXIM Co. Ltd, 2004.
- [28] Mark I. Montrose. 电磁兼容和印刷电路板理论、设计和布线. [M] 人民邮电出版社, 2002.
- [29] 顾海洲, 马双武. PCB电磁兼容技术—设计实践[M]. 清华大学出版社, 2004.
- [30] 甘亚光. 嵌入式无线远程监测终端的设计与实现 [D]:[硕士]. 曲阜:曲阜师范大学, 2008.
- [31] SIM100TCP AT命令使用说明,[Z], SIMCOM Ltd,2003.

致 谢

三年的研究生生活就要过去了。回首这三年，无疑它是我生命中美好的三年。我要向关心和支持我学习的所有领导、老师和朋友们表示真挚的谢意!感谢他们对我的关心和支持!

首先衷心感谢我的导师李洪亮教授在学习与生活中的谆谆教导和悉心照料。导师治学严谨，学识渊博，平易近人，生性随和，实事求是的作风，敏锐开阔的科研思路，永远是我学习的榜样。借此机会，谨向导师李洪亮教授致以衷心的感谢和崇高的敬意!

衷心感谢中国农业科学院农业环境与可持续发展研究所可控环境农业实验室的孙忠富教授。感谢孙老师在学习和生活上对我的指导与帮助，感谢孙老师给我提供了良好的实验条件。借此机会，谨向孙老师致以衷心的感谢和崇高的敬意!

衷心感谢曲阜师范大学自动化研究所各位老师，谢谢他们这三年来给我的关心与帮助。衷心感谢曲阜师范大学自动化研究所05级师兄师姐们给我的支持与帮助，特别是师兄甘亚光和吕永田，从两位师兄生身上我学到了很多宝贵的东西。感谢06级全体同学，感谢他们对我的帮助和支持。感谢同实验室的张斌、陈冉、方燕同学，同窗之谊和手足之情，我将终生难忘!另外感谢中文系07级董倩倩同学帮我最后修改论文中的用词。

衷心感谢可控环境农业实验室的各位同学给我的帮助和支持。尤其感谢杜克明老师、赵伟、韩华峰和贾倩等同学，是他们的帮助我才能顺利完成毕业论文等相关工作。

衷心感谢教育过我的所有老师，感谢他们在我学业上对我的支持与帮助，感谢他们默默无闻做出的奉献。

最后，衷心的感谢我的父母和其他所有关心我的人。他们一直默默的在我的背后支持我，鼓励我，他们是我不断前进的最大动力!衷心感谢我这么多年来一直支持与关心我的所有亲人和朋友。

程兵

2009年3月

基于GPRS的GPS车载终端

作者：[程兵](#)
学位授予单位：[曲阜师范大学](#)

本文链接：http://d.g.wanfangdata.com.cn/Thesis_Y1459728.aspx

授权使用：湖南大学(hunandx)，授权号：ef8bc8f1-e018-434b-8138-9dad0101f9d0

下载时间：2010年7月8日