

## 摘要

汽车仪表是驾驶员与汽车进行交流的重要窗口，也是汽车高新技术的重要组成部分。传统汽车仪表多使用指针型显示器件为主，如步进电机、十字线圈，辅以液晶显示，显示的信息量相对较少，且结构复杂。一方面随着汽车电子化程度的不断提高，进行技术创新，研制开发新一代汽车仪表产品；另一方面，由于能源和环保问题，汽车也将从内燃机汽车发展到包括纯电动汽车（BEV）、混合动力汽车（HEV）以及燃料电池汽车（FCV）的新能源汽车时代，因此结合新能源汽车信息量多、电子化程度高的特点，开发新一代汽车智能仪表具有重要的现实和长远意义。

本文正是在这样的背景下，以同济大学汽车学院自主研发的 ROVER 燃料电池轿车为研究对象，进行了汽车智能仪表的一些功能研究与开发。所做的主要工作有：

- (1) 根据要实现的功能确定所需的硬件资源，选择合适的嵌入式硬件系统。
- (2) 嵌入式操作系统的选择和二次开发。在选择操作系统时要考虑到系统的硬件可移植性、实时性、对内存的需求以及提供哪些开发工具等。
- (3) 应用软件的开发。主要是仪表界面设计，包括数字图形显示，动画显示，数据库开发等。
- (4) 基于无线数据传输模块下的 GPRS 无线通讯实验。包括客户端和服务端系统配置，动态域名解析等。

该仪表已应用于 ROVER 燃料电池轿车，实践表明，在嵌入式平台上显示车载信息，同传统仪表相比具有较大的优势。可满足小型化、轻量化的要求；造型美观，可动画显示、可读性、可视性强；可实现一表多用。从软件方面来讲，引入了操作系统的概念，增强了代码的可读性、可维护性、可扩展性以及灵活性；信息显示自由度高，显示界面人性化，可定制；即使更换硬件平台，也只需对操作系统和底层驱动程序进行少量的移植工作，而无需修改与硬件无关的应用代码。

关键词：汽车仪表，嵌入式系统，显示系统，远程监控

## ABSTRACT

Drivers and cars are linked by the motor instruments, and the motor instruments incarnate the profundity of motor technology. The motor instruments are far behind the development of foreign car industry. The traditional instruments are often use step-motor, hands, LED. On the one hand, the mass movement for technical innovation and develop the new generation of motor instrument are imperative under the situation. On the other hand, The traditional engine vehicle will be replaced by battery electric vehicle (BEV), hybrid electric vehicle (HEV) and fuel cell vehicle (FCV). So it is of great importance to develop new intelligent car meters.

Taking "ROVER FCV" as research object, this paper is about the research and development of intelligent meter for auto based on ARM system.

Firstly, confirm the necessary resources through functions to be achieved. Choose right embedded system.

Secondly, choose and develop the embedded OS.

Thirdly, write and compile application program for information display of vehicle, including animation and GUI.

Finally, plan design of Long-distance monitoring for Automobile condition

This instrument is used in ROVER-FCV. The practice indicates compared with the traditional measuring appliance this instrument has big superiorities. It is more readable, more accurate, more updatable and more extensible when the embedded instrument is applied.

**Key Words:** Vehicle Instrument, Embedded System, Display System, Long-distance Monitoring

# 学位论文版权使用授权书

本人完全了解同济大学关于收集、保存、使用学位论文的规定，同意如下各项内容：按照学校要求提交学位论文的印刷本和电子版；学校有权保存学位论文的印刷本和电子版，并采用影印、缩印、扫描、数字化或其它手段保存论文；学校有权提供目录检索以及提供本学位论文全文或者部分的阅览服务；学校有权按有关规定向国家有关部门或者机构送交论文的复印件和电子版；在不以赢利为目的的前提下，学校可以适当复制论文的部分或全部内容用于学术活动。

学位论文作者签名：

年 月 日

-----

经指导教师同意，本学位论文属于保密，在 年解密后适用本授权书。

指导教师签名：		学位论文作者签名：	
	年 月 日		年 月 日

## 同济大学学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：

年    月    日

## 第1章 绪论

### 1.1 本课题来源、研究的意义和背景

本课题依托于新能源汽车 ROVER 燃料电池轿车仪表子项目，重点在于汽车智能仪表的功能研究与设计。

汽车仪表的发展基本上经历了机械式仪表、机电式模拟仪表和数字化仪表三个阶段，现在汽车仪表正朝着数字化智能化方向发展。

车辆仪表是驾驶员与汽车进行信息交流的重要接口和界面，是车辆安全行驶的重要保证；也是汽车电子技术的重要部分，各国一直在努力开发汽车仪表技术，并不断取得新的发展。传统汽车上较常用的一般仪表有电流表、电压表、机油压力表、水温表、燃油表、车速里程表等指示仪表以及方向指示灯等。

随着现代汽车新技术的不断出现，尤其是新能源环保电动汽车（主要有燃料电池、纯电动、混合动力汽车）的快速发展以及汽车电子化和信息化水平的不断提高，使得只能为驾驶员提供汽车运行中必要而又少量的数据信息的传统仪表，已远远不能满足行车的要求。以本课题的燃料电池汽车（FCV）为例，虽然没有了传统汽车发动机转速、水温、发动机报警等信号，但是却增加了更多的关于燃料电池动力系统参数的信号量以及其他用于车辆控制和状态显示的量。如燃料电池堆冷却水温度、储氢瓶压力、充氢和放氢指示、各电池电机参数等。

研发 ROVER 智能仪表，一是原车仪表已不能满足燃料电池动力系统众多参数的显示需求，二是开发先进的智能化、平面显示仪表与先进的燃料电池汽车技术相匹配，三是自行研发，可节约成本，此外该仪表不仅可用于燃料电池汽车，其先进的理念和技术也可用于纯电动汽车、混合动力汽车和传统汽车。

汽车发展到今天，不再是传统意义上的简单的机械装置和代步工具。汽车上的电子化、信息化和智能化装备在技术水平和应用数量上的不断提高已成为现代汽车发展的重要标志之一；这些新技术也是用来改善和提高汽车性能的有效技术手段<sup>[1]</sup>。行车安全、城市交通拥堵一直是人们关注的热门问题，将先进的电子、网络与通信技术应用于汽车，使汽车具有信息处理、通讯、导航、防

盗、语言识别、图像显示和娱乐等功能。不仅能大大提高汽车的综合性能，在提高车辆安全，对行车事故实施紧急救援、追踪被盗车辆、远程监控汽车状态和减轻驾驶员负担、丰富乘客旅途的文化生活、提高人们的出行效率和降低能源与社会资源浪费等方面，具有广阔的市场前景和深远的实际意义。在有线和无线数据通讯网的支持下，一方面通过链接各种信息交换中心和因特网获取和应用各种信息，使驾驶员与乘客始终与外界保持密切的联系，另一方面构筑汽车的信息应用平台，也有利于今后开展移动电子商务、移动网络办公和移动网上冲浪等增值服务。在此过程中汽车仪表承担重要角色，一方面是为仪表多用，另一方面也是电子信息技术等集成化发展的必然趋势，使得汽车仪表的发展突现出智能化、网络化、信息化、集成和多功能的特点，芯片、总线、接口技术成为仪器仪表的内核，嵌入式系统成为技术的代表，并由此形成了汽车仪表行业发展的趋势<sup>[2][3]</sup>。

本课题研究的另一个重要意义在于，设计汽车远程状态监控方案。将远程监控用于新车试验阶段，能大大缩短新车开发周期，及早地发现问题并改进，使新车快速走向成熟并开发下一代汽车。

## 1.2 汽车仪表的国内外研究现状和发展趋势

### 1.2.1 国内外研究现状

在国外，汽车仪表作为汽车驾驶室的主要行驶信息显示装置，其技术性能、外观造型、视觉效果以及功能等均受到高度重视，汽车仪表平均配套价格约占整车售价的 3%。以德国和日本的汽车仪表制造企业为技术领先的产品，已由传统的组合仪表向汽车集成信息系统方向发展，开发了代表国外同行业先进水平的多功能全电子显示仪表、平视显示仪表、汽车导航系统、行车记录仪等高技术产品。近二十年来，微电子技术、计算机技术、精密机械技术、集成技术、网络技术等高新技术获得了迅速发展，实现了现代汽车仪表众多的、全新的功能，已经完全突破了传统的框架，向着计算机化、网络化、智能化、多功能的方向发展，高科技化已经成为现代仪器仪表的主要特征。国外现在已应用于汽车仪表的先进高科技技术主要包括：方便驾驶员用声音指令设定、控制和打电话的声控技术；声控投影图象显示技术；为用户电子邮件收发、实时路况信息、

天气预报和股市跟踪的技术；GPS 卫星定位和精确电子导航技术；远程故障诊断和车载免提电话等<sup>[4]</sup>。

早在 1998 年 11 月的 Comdex 大展上，微软就展示了一台支持 Windows CE 的汽车 PC。汽车 PC 的基本组成系统包括音响系统、无线服务设备、地址簿、驾驶指南等。随后 Alpine Japan, Clarion, Nissan USA, Toyota 等十几家公司与微软合作生产汽车 PC。这些汽车 PC 的基础设计是以多媒体为中心，它们被安装在汽车仪表板上。因为在开车时应当注意力集中，去观察你周围的道路和其它状况，埋头看屏幕或键盘是不安全的，它们主要依靠语音控制，但也准备了足够的键让人们访问系统<sup>[5]</sup>。

目前我国汽车电子设备的研究开发和生产能力还相当薄弱，只相当于美国七八十年代的水平。国内汽车仪表生产企业 100 多家，全行业 2005 年生产成套汽车仪表约 500 万套，总产值约 25 亿元，产量排在前 10 位的企业占总产量的 90%以上。现在国内汽车仪表生产企业生产技术水平参差不齐，从工艺水平来看，主要分为两类：一类是与国外合资的企业，如上海德科电子仪表有限公司和广州马瑞利仪表有限公司等。这些企业的生产工艺和设备由国外直接引进，相对具有一定的优势，但国产化率不高，且不具备适应国内汽车工业产品市场的技术开发能力；另一类是原机械部或集团支持发展的企业，如芜湖仪表厂、四平仪表厂、东风汽车电子仪表股份有限公司、上海长江仪表厂及后来发展的绍兴怡东仪表有限公司等，这些企业拥有国内市场的销售渠道和适合国内汽车使用的低附加值产品，在产量和质量上能满足现有国内车型的配套要求，但工艺装备和产品技术水平与国际先进水平相比还有不小的差距<sup>[6]</sup>。

### 1.2.2 汽车仪表发展趋势

随着计算机技术、电子技术、网络技术以及液晶现实技术的发展，汽车仪表的发展趋势将更加体现这些高新技术的结合。如仪表的功能由软件和硬件共同来完成仪表的功能由软件和硬件共同实现，而且主要是通过软件实现。这对于量大且对成本极为敏感的汽车仪表有特殊意义，因为软件的开发费用分摊到每个仪表上是非常少的。与仅由电子线路硬件组成的汽车仪表相比，带有嵌入式系统的汽车仪表，其功能的实现手段更加灵活多样。产品的“柔性”更好，即在推出新款产品时，能最大限度地利用以前产品的硬、软件设计成果仅做少

量修改便可，这在产品更新换代很快的今天和未来特别重要未来的汽车仪表不再是单纯的车速、里程、油压等显示，汽车仪表将演变为车载综合信息平台。它将包括车载计算机及网络系统、综合显示系统、信息提示系统、组合式全球定位系统（Compound Global Position System, C-GPS）、地理信息系统（Geographical Information System, GIS）、综合信息管理系统（Integrated Information Manage System, IIMS）以及无线数据通讯系统等。汽车配置这些装备的目的是使汽车在智能交通系统的框架下帮助汽车驾驶员实现安全、快速、高效的运行，当然还可能具有节能和大幅提高社会公共资源综合利用率等优势<sup>[1]</sup>。其主要特点有：

(1) 仪表能提供大量复杂的信息。为了提高汽车的使用性能，汽车的电子控制程度越来越高。电子控制装置必须能迅速、准确处理各种复杂的信息，并通过汽车仪表盘以数字、文字或图形等方式显示出来，使驾驶员了解与掌握汽车的当前状态，以便及时处理各种复杂的情况。目前，汽车的故障诊断、地形图显示、导航及各种信息服务装置都已开始装备汽车，汽车仪表盘作为信息显示终端已经是大势所趋。

(2) 满足小型、轻量化的要求。为了使有限的驾驶空间尽可能宽敞些，用于汽车的各种仪表及附件都必须小型轻量化。电子化仪表不仅能适应各种传感器或控制系统的电子化，而且可实现小、轻、薄，既能节省汽车仪表盘附近的宝贵空间，增加一定的仪表数量，还可处理日益增多的信息量。

(3) 一表多用。汽车电子化仪表可采用数字显示，可用一组数字进行分时显示，也可以同时显示多个参数，这样，汽车就不必为每个参数设置一个指示表，故使仪表盘得以简化，实现一表多用。

(4) 外形设计自由度高。汽车仪表盘造型美观，这对一辆汽车来说非常重要，推出最流行的仪表新款式，选用外形设计自由度特别高的电子显示器件则是实现汽车现代化、提高产品竞争力的有力措施。

(5) 仪表指示形式将演变成计算机终端显示器，指示形式将演变成一个高清晰度的计算机显示器。

(6) 具备完善的通讯系统，在电子信息技术，网络通讯技术日趋成熟的今天，汽车仪表正在不断融入当今各学科、各领域的高新技术、新成果，向功能多元化、机电一体化、系统工程化、高度集成化方向发展



## 1.3 本课题的主要研究工作、方法与技术路线

### 1.3.1 本课题主要研究内容

#### 1. 硬件系统选择

确定了该仪表要解决的问题和要实现的目标以后，首先需要选择合适硬件系统，包括处理器及其外围设备的选择。

#### 2. 软件开发平台的选择

综合考虑各因素，选择合适的嵌入式操作系统和软件开发工具。

#### 3. 嵌入式操作系统二次开发

嵌入式操作系统一般都是根据功能需求可定制的，即操作系统的二次开发。

#### 4. 应用软件的设计

按照仪表需要实现的功能，软件开发包括仪表显示程序、数据库设计和远程通讯程序三部分。

### 1.3.2 研究方法和技术路线

研究方法和技术路线如图 1.1 所示。我做的主要工作是软件开发那一块。

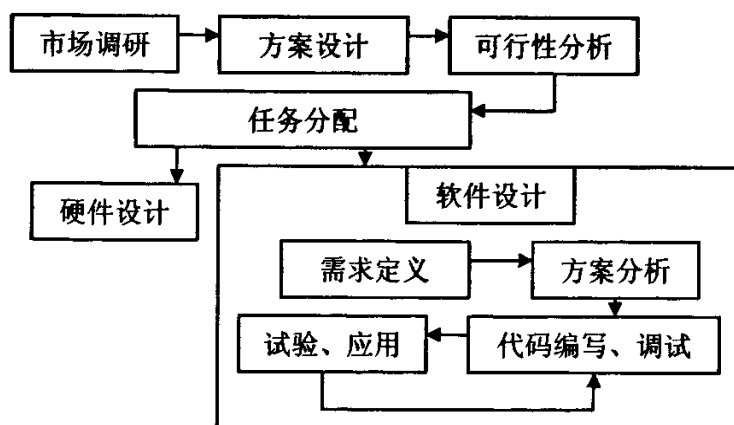


图 1.1 研究方法和技术路线

## 1.4 本章小结

本章介绍了本课题的来源、意义和背景，介绍了汽车仪表的发展现状和趋势。结合燃料电池汽车信息量多、技术先进的特点提出了嵌入式智能仪表的概念。最后还介绍了本课题的主要研究内容和技术路线。

## 第二章 ROVER-FCV 仪表总体设计

ROVER—FCV 组合仪表开发,一方面是由于 ROVER—FCV 汽车的燃料电池动力系统结构复杂,信息量多,原车组合仪表不再能够完全满足新的动力系统数据显示的要求;另一方面也是进行技术创新,研制开发新一代汽车仪表产品。

### 2.1 ROVER 仪表面板设计开发要点

ROVER 仪表面板设计满足以下要求。

- (1)能够实现原车保留部件的状态指示功能,增加新的动力系统数据显示功能。
- (2)造型美观,采用大面积彩色 LCD 作为主要显示器件。
- (3)和原车仪表电气接口兼容。
- (4)和原车仪表机械尺寸。

### 2.2 ROVER 仪表功能设计

#### 2.2.1 ROVER 仪表信号显示形式

ROVER 仪表显示的信息从显示形式上来划分,可分为四类,它们是 LED 指示灯显示、步进电机驱动指针指示、液晶显示屏显示和声音警告。

##### 1. LED 指示灯显示

- (1) H<sub>2</sub> 泄露报警。
- (2) 加 H<sub>2</sub> 状态显示。
- (3) 辅助蓄电池充电。
- (4) 发动机故障报警。
- (5) 停车制动显示。
- (6) 巡航控制。

- (7) 安全气囊。
- (8) 安全带。
- (9) ABS 故障报警。
- (10) 大灯。

其中氢气泄露报警有 4 个状态，无泄露、轻度泄露、泄露、严重泄露。4 个状态由 LED 的闪烁来区分：

- (1) 无泄露—LED 长灭。
- (2) 轻度泄露—LED 闪烁，频率为 640mS。
- (3) 泄露—LED 闪烁，频率为 320mS。
- (4) 严重泄露—LED 长亮。

H2 压力低报警灯，当储氢瓶压力低于 4MP 时报警灯亮。

## **2. 仪表指针指示：**

- (1) 燃料电池出水温度。
- (2) H2 贮备量。
- (3) 电动机功率。

## **3. 中间液晶显示屏显示：**

主要显示燃料电池动力系统工作状态，如车速、里程、温度等。

## **4. 声音警告：**

### **(1) 超速警告声**

设定两个超速阈值，一个是法规超速阈值=120km/h（中国高速公路限速值），一个是燃料电池车最高车速阈值=150km/h（受电机最高转速限制）。

如果当前车速达到法规超速阈值，组合仪表发出单次警告声。在同一次驾驶过程中，当车速降低到比法规超速阈值低后，再次达到法规超速阈值，组合仪表才会再次发出单次警告声。

如果车速达到最高车速阈值，组合仪表发出单次警告声，重复频率是 5s。直到车速降低到比最高车速阈值低后，警告声才停止。

### **(2) 安全带警告声**

安全带警告声的触发条件是 KL.15=ON 且安全带扣没有连接（Seat belt warning=高）。安全带警告声采用连续低频警告声，持续时间最长 10s，如果在

10s 内检测到安全带扣已连接，则解除警告声

### (3) 点火钥匙警告声

防盗模块 GIM 检查钥匙和车门状态，如果 KL.15=OFF，左前门打开且点火钥匙插在钥匙座内，则 GIM 发出点火钥匙警告信号 (via K bus, message ID =74h)。如果钥匙从钥匙座拔出，左前门关闭或者 KL.15=ON，则 GIM 取消点火钥匙警告信号。组合仪表接到信号，发出单次高频警告声，警告声重复的频率是 2.2Hz，持续周期 180s。如果组合仪表在 180s 内断电，或者 K bus message 取消了点火钥匙警告信号，则停止警告声。

## 2.2.2 ROVER 仪表参数选择及显示方案

ROVER 仪表参数选择及显示方案如表 2.1 所示。

表 2.1 ROVER 仪表参数选择及显示方案

参数	显示方式
车速	LCD 虚拟指针显示
电机功率	指针
H <sub>2</sub> 量	指针
FC 出水温度	指针
H <sub>2</sub> 泄露报警	LED 指示灯
ABS 故障报警	LED 指示灯
大灯	LED 指示灯
动力系统故障	LED 指示灯, (颜色-故障级别)
加 H <sub>2</sub> 状态	LED 指示灯
驻车制动	LED 指示灯
辅助电池充电	LED 指示灯
巡航控制	LED 指示灯
安全气囊报警	LED 指示灯
安全带报警	LED 指示灯
左转向灯	LED 指示灯
右转向灯	LED 指示灯
蓄电池剩余电量	LCD 图形显示
档位 PRND123	LCD 字符显示
故障码	LCD 字符显示
动力系统状态	LCD 图形显示

续表 2.1

功率	LCD 字符显示
进水温度	LCD 字符显示
加 H <sub>2</sub> 状态	LED 指示灯
氢瓶温度	LCD 字符显示
续驶里程	LCD 字符显示
平均消耗率	LCD 字符显示
功率	LCD 字符显示, 信号源自 CAN
温度	LCD 字符显示, 信号源自 CAN
电压	LCD 字符显示, 信号源自 CAN
SOC	LCD 字符显示, 信号源自 CAN
单电池最小电压	LCD 字符显示, 信号源自 CAN
功率	LCD 字符显示, 信号源自 CAN
水温	LCD 字符显示, 信号源自 CAN
温度	LCD 字符显示, 信号源自 CAN
水温	LCD 字符显示, 信号源自 CAN
电压	LCD 字符显示, 信号源自 CAN

## 2.3 ROVER 仪表面板设计

### 2.3.1 面板设计

本课题研究目的是开发出适合车载智能仪表的系统及应用软件。对比传统仪表就是充分利用电子技术、计算机技术、信息技术及网络技术,使仪表的功能将极大的拓宽,指示形式将演变为计算机终端显示器。经过各方面的调查研究,结合汽车仪表的主要功能、当前的主要技术和汽车仪表信息系统的发展趋势,考虑到驾驶员们的使用习惯,整个仪表界面仿制传统车辆仪表界面,车速使用仿机械指针式仪表。整个仪表面板如图 2.1 所示。左侧的 H<sub>2</sub>(氢气)温度、氢气量和右侧电动机功率仍然采用步进电机驱动物理指针的电子仪表形式显示;LED 指示灯显示的信号有 H<sub>2</sub> 泄露报警、加 H<sub>2</sub> 状态显示、辅助蓄电池充电、发动机故障报警、停车制动显示、大灯等;中间液晶屏则构成仪表盘的主画面,显示车速和燃料电池动力系统工作状态。本设计中通过 LCD 来代替车上的部分传统机械式仪表。其特点是风格变化多样,可分屏显示,可动画显示,信息量

增大，可以根据具体的需要来改变具体的仪表界面。

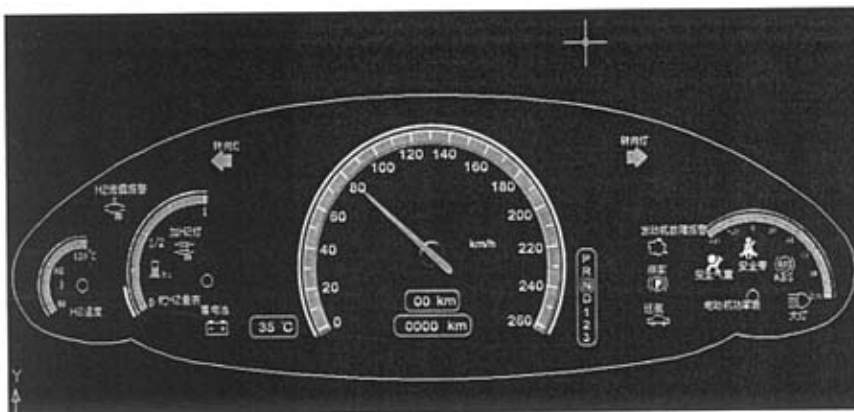


图 2.1 仪表面板设计图

### 2.3.2 液晶显示屏 (LCD) 选择

在本仪表方案中使用的是夏普 10.4 寸的 16 位色 TFT 液晶屏。液晶显示器显示原理与传统的 CRT 显示器迥然不同。相比之下，液晶显示器是“健康”型显示器，它低辐射，低耗能，散热小，纤薄轻巧，CRT 显示器则较厚重。采用数字接口的液晶显示器能精确还原图像，可以把画面完美的在屏幕上呈现出来，且可以做到无闪烁。TFT 型液晶显示器从技术上来说是最新一代液晶显示器，同 TN 和 STN 型的液晶显示器相比，其构造较为复杂，显示器先利用背光源发光，只要改变加在液晶上的电压值就可以控制最后出现的光线强度与色彩，这样就能在液晶面板上变化出有不同色调的颜色组合了，其显示效果也有了大幅度提高。并且 TFT 型液晶显示器的可视角度能达到 90-170 度，抗震动、抗电磁干扰性好，工作温度宽，一般为  $-30^{\circ} \sim 85^{\circ}$ ，这些都满足车用要求；本课题使用的 TFT 液晶显示器价格不到 100 美元，其背光灯的使用寿命可达 15,000 小时，假设汽车以平均 40 公里的时速每天行驶 8 小时，则 TFT 液晶显示器使用寿命折合为 600,000 公里。TFT 与 TN 和 STN 型的液晶显示器的其它一些参数比较见表 2.1。由于 TFT 的优越性能，我们选用了 TFT 型液晶屏。

表 2.1 TFT、TN 及 STN 型的液晶显示器之比较

类别	TN	STN	TFT
原理	液晶分子, 扭转 90 度	液晶分子, 扭转 180~270 度	液晶分子, 扭转 90 度
特性	黑白、单色、低对比	黑白、彩色、低对比	彩色 (1667 万色) 高对比
全色彩化	否	否	可媲美 CRT 之全彩色
动画显示	否	否	可媲美 CRT
视角	30 度以下	40 度以下	90-170 度
面板尺寸	1~3 寸	1~12 寸	37 寸以下

## 2.4 ROVER 仪表硬件总体方案

ROVER 仪表硬件采用双 CPU 的结构, 如图 2.2 所示。

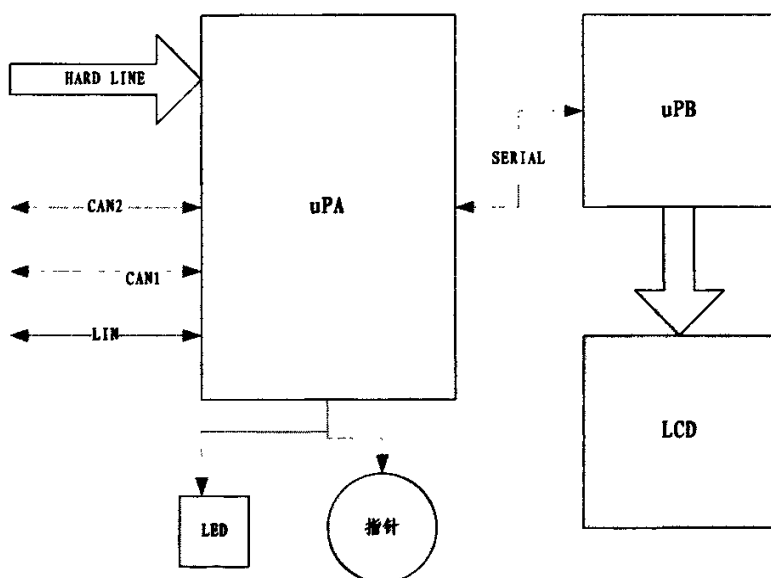


图 2.2 ROVER 仪表硬件总体方案示意

其中 uPA 使用的是英飞凌的 XC164，其功能是：指针驱动、LED 指示灯、CAN 网关及接口、K-BUS 硬件接口、蜂鸣器、串口连接 uPB、ARM 复位。

uPB 为 ARM 嵌入式系统，它是仪表的信息中心用于图形和数据显示、远程连接、语音等。



来自汽车各部分的信号有三种形式，硬件信号、CAN 信号和 LIN 信号。这些信号首先输入到 XC164，XC164 对信号进行处理，由 LED 指示灯指示和由指针指示的信号则直接由 XC164 控制 LED 灯和步进电机指示，其余的信号则通过串口送往 RAM 系统，由 LCD 输出。

我要做的主要工作就是硬件层面上的 ARM 系统的软件开发。

### 2.5 本章小节

本章讲述了 ROVER 仪表的总体设计方案，包括仪表功能说明、面板设计和硬件总体方案等，整个仪表界面仿制传统车辆仪表界面，中间液晶屏则构成仪表盘的主画面，造型美观、大方。

## 第3章 嵌入式平台选择

按照嵌入式系统的工程设计方法，嵌入式系统的设计可以粗略的分成三个阶段，即：分析，设计，实现。分析阶段的主要任务是确定要解决的问题及需要完成的目标。因此，分析阶段也常常被称为“需求阶段”，或“系统需求阶段”。设计阶段主要任务是解决如何在给定的约束条件下完成用户的要求。此阶段是在分析阶段的基础上研究“如何做”。实现阶段主要是解决如何在所选择的硬件和软件的基础上进行整个软、硬件系统的协调实现。按照这种设计思想，在分析阶段结束后，通常开发者面临的一个棘手的问题就是硬件和软件的选择，因为它的好坏直接影响着实现阶段的任务完成。通常硬件和软件的选择包括处理器选择，硬件部件选择，操作系统选择，语言选编程择，软件开发工具选择，硬件调试工具选择和软件组件选择。

### 3.1 嵌入式系统设计流程

嵌入式设计包括软、硬件协同设计，主要设计流程如图 3.1 所示。

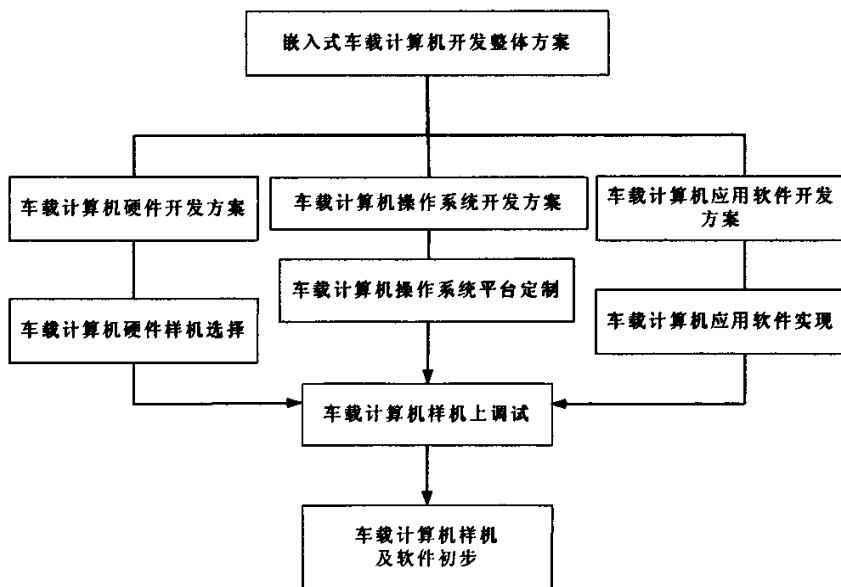


图 3.1 系统设计流程

## 3.2 硬件的选择

### 3.2.1 嵌入式处理器的选择

嵌入式系统的核心部件是各种类型的嵌入式处理器，目前据不完全统计，全世界嵌入式处理器的品种总量已经超过 1000 多种，流行体系结构有 30 几个系列。但与全球 PC 市场不同的是没有一种微处理器和微处理器公司可以主导嵌入式系统，仅以 32 位的 CPU 而言，就有 100 种以上嵌入式微处理器。由于嵌入式系统设计的差异性极大，因此选择是多样化的。设计者在选择处理器时要考虑的主要因素有：

目前市场上的 CPU 供应商。

有些公司如 Motorola、Intel、AMD 很有名气，而有一些小的公司如 QED (Santa Clara CA) 虽然名气很小，但也生产很优秀的微处理器。另外，有一些公司，如 ARM、MIPS 等，只设计但并不生产 CPU，他们把生产权授予世界各地的半导体制造商。ARM 是另外一种近年来在嵌入式系统有影响力的微处理器制造商，ARM 的设计非常适合于小的电源供电系统。Apple 在 Newton 手持计算机中使用 ARM，另外有几款数字无线电话也在使用 ARM<sup>[8]</sup>。

处理器的处理速度。

一个处理器的性能取决于多个方面的因素：时钟频率，内部寄存器的大小，指令是否对等处理所有的寄存器等。对于许多需用处理器的嵌入式系统设计来说，目标不是在于挑选速度最快的处理器，而是在于选取能够完成作业的处理器和 I/O 子系统。如果你的设计是面向高性能的应用，那么还是应该考虑某些新的处理器，其价格极为低廉，如 IBM 和 Motorola 的 PowerPC。以前 Intel 的 i960 是销售得极好的 RISC 高性能芯片，但是最近几年却遇到强劲的对手，让位于 MIPS、SH 以及后起之星 ARM<sup>[9]</sup>。

技术指标。

当前，许多嵌入式处理器都集成了外围设备的功能，从而减少了芯片的数量，进而降低了整个系统的开发费用。开发人员首先考虑的是，系统所要求的一些硬件能否无需过多的胶合逻辑(GL)就可以连接到处理器上。其次是考虑该处理器的一些支持芯片，如 DMA 控制器，内存管理器，中断控制器，串行设备、时钟等的配套<sup>[10]</sup>。

处理器的低工耗。

嵌入式微处理器最大并且增长最快的市场是手持设备、电子记事本、PDA、手机、GPS 导航器、智能家电等消费类电子产品，这些产品中选购的微处理器典型的特点是要求高性能、低工耗。许多 CPU 生产厂家已经进入了这个领域。今天，用户可以买到一颗嵌入式的微处理器，其速度像笔记本中的 Pentium 一样快，而它仅使用普通电池供电，并且价格不足 50 美元。典型的例子有 NEC、日立为 HPC、Palm PC 而设计的 VR4111 和 SH7707。Digital 的 Strong ARM1100，在一个极小的 200MHz 主频封装中集成了彩色 LCD 控制器、PCMCIA、触屏接口等 6 个接口(含 USB、IRDA)，而批量的价格也在 40 美元以内<sup>[1]</sup>。

处理器的软件支持工具。

仅有一个处理器，没有较好的软件开发工具的支持，也是不行的，因此选择合适的软件开发工具对系统的实现会起到很好的作用。

处理器是否内置调试工具。

处理器如果内置调试工具可以大大的缩小调试周期，降低调试的难度。

处理器供应商是否提供评估板。

许多处理器供应商可以提供评估板来验证你的理论是否正确，验证你的决策是否得当。

以上是嵌入式处理器的选择原则，我们在调查了现在国内较流行的处理器类型，同时考虑到汽车电子的应用特点后选择了 ARM 处理器。ARM 处理器三大特点：

- (1) 小体积、低功耗、低成本、高性能
- (2) 16/32 位双指令集
- (3) 全球众多的合作伙伴

随着集成电路的发展及信息时代的到来，在 2001 年，32 位 ARM 处理器市场占有率超过了 75%，广泛应用在信息电器如掌上电脑、个人数字助理 (PDA)、可视电话、移动电话、TV 机顶盒、数码相机等嵌入式设备。ARM 已成为移动通信、手持计算、多媒体数字消费等嵌入式解决方案的 RISC 标准。

ARM (Advanced RISC Machines) 是微处理器行业的一家知名企业，是知识产权 (IP) 供应商，本身不生产芯片，靠转让设计许可由合作伙伴来生产各具特色的芯片。ARM 公司设计了大量高性能、廉价、耗能低的 RISC 处理器、相关技术及软件。目前，有超过 30 家半导体公司与 ARM 签订了硬件技术使用许可协

议,其中包括 Intel、IBM、SAMSUNG、OKI、LG、NEC、SONY、PHILIPS 等大公司。至于软件系统的合伙人,则包括微软、SYMBIAN 和 MRI 等一系列知名公司。

ARM32 位体系结构被公认为业界领先的 32 位嵌入式 RISC 处理器结构,所有 ARM 处理器共享这一体系结构。这可确保开发者转向更高性能的 ARM 处理器时,由于所有产品均采用一个通用的软件体系,所以相同的软件可在所有产品中运行(理论上如此),从而使开发者在软件开发可获得最大回报<sup>[12]</sup>。当前 ARM 体系的扩充包括:

- (1) Thumb: 16 位指令集,用以改善代码的密度;
- (2) DSP: 用于 DSP 应用的算术运算指令集;
- (3) Jazeller: 允许直接执行 Java 字节码的扩充。

### 3.2.2 硬件平台的选择

在硬件设计中,采用 CPU 核心板加扩展板的设计结构,使调试和使用灵活方便,扩展方式得到延伸,可以根据设计需要更换或升级核心板有效保护前期资源。系统结构图如图 3.2 所示。

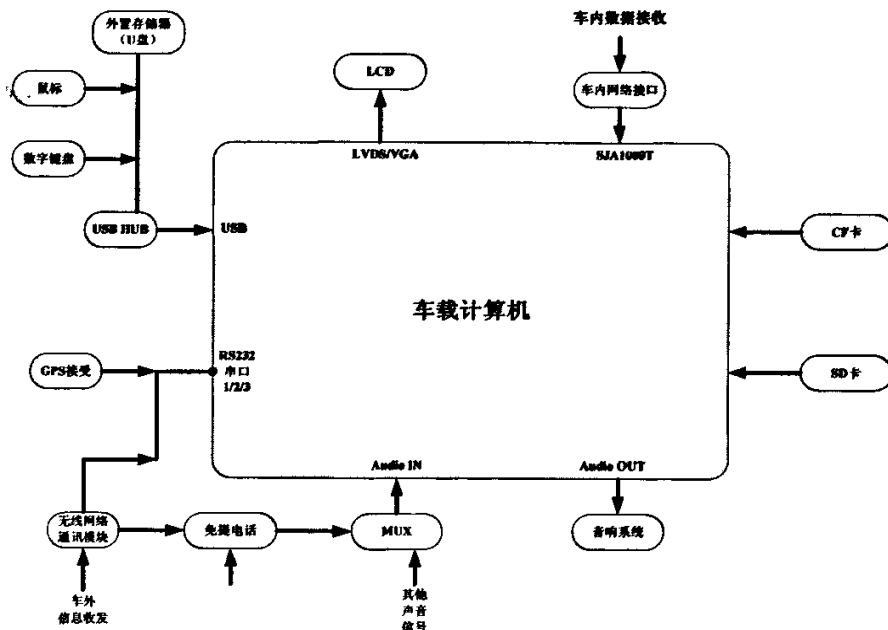


图 3.2 硬件系统架构

由于作为研究项目,规模方面不可能很大,为了缩短开发周期,降低开发

风险，在本课题中我们使用 YFDVK-2410-II 开发板，其结构如图 3.3 所示。

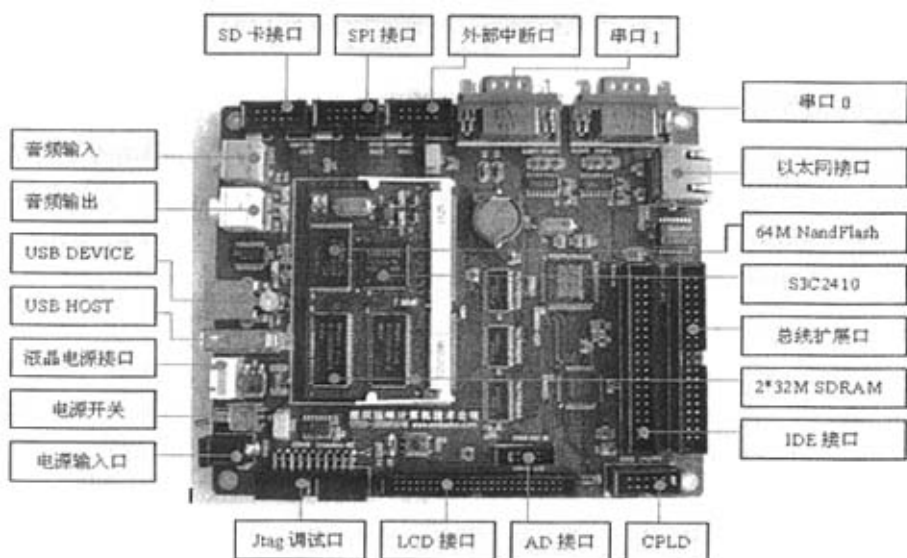


图 3.3 YFDVK-2410-II 开发板结构图

它是针对 S3C2410 的高性能开发平台，并适合用来作为开发高性能手持式以及便携式智能设备或终端。核心板使用三星公司的目前较先进的 32 位嵌入式 ARM 处理器 S3C2410 系列，主频最高达 400MHz。提供的接口包括 TFT 64K 色 LCD 控制器以及触摸屏控制器，2 通道 UART（已包含 IRDA 红外线数据通讯口），1 个 USB 主机接口，1 个 USB 设备接口，SD 卡/MMC 卡主机控制器接口，Embedded-ICE 调试接口，RTC 实时时钟（具备后备锂电池），IIC 总线接口（驱动 AT24C04-SC27），ADC 模数转换接口，SPI 接口，IIS 数字音频输入/输出接口，EINT 外部中断接口，10M 以太网接口，多功能总线扩展接口和 IDE（硬盘）接口。

### 3.3 软件的选择

软件的选择包括嵌入式操作系统、编程语言及编程工具的选择。

#### 3.3.1 操作系统的选择

尽管嵌入式系统有着无比广阔的市场需求和发展前景，但嵌入式系统的发

展多年来却经历了一个曲折和痛苦的历程。随着微处理器的产生,价格低廉、结构小巧的 CPU 和外设连接提供了稳定可靠的硬件架构,那么限制嵌入式系统发展的瓶颈就突出表现在了软件方面。从八十年代末开始,陆续出现了一些嵌入式操作系统,比较著名的有 Tornado/VxWorkx、pSOSystem、Nucleus 和 Windowss CE、QNX、VRTX,以及沸沸扬扬的“女娲计划”,中国人自己的嵌入式操作系统 HOPEN 和现在谈论最多的嵌入式 Linux 操作系统等等。可用于嵌入式系统软件开发的操作系统很多,但关键是如何选择一个适合你所开发项目的操作系统,我们认为应该从以下几点进行考虑<sup>[13]</sup>:

(1) 操作系统提供哪些开发工具。

有些实时操作系统(RTOS)只支持该系统供应商的开发工具。也就是说,还必须向操作系统供应商获取编译器、调试器等。而有些操作系统使用广泛且有第三方工具可用,因此,选择的余地比较大。如 PSOSystem 支持的工具有: pRISM+, 全集成开发环境,含源代码调试器(SpOTLIGHT), C/C++ 编译器(e g Diab), 汇编器,连接器, C/C++ 开发环境(SNiFF+), 嵌入式系统监视工具(Esp), CORBA 等。而 Tornado/VxWorkx 支持的工具有: 远程源级调试器; 浏览器; WindSh 命令行接口; 模块载入器; 目标工具; WindConfig 板基支持包配置; 大约 90 个第三方嵌入式开发工具和扩展。

(2) 操作系统向硬件接口移植的难度。

操作系统到硬件的移植是一个重要的问题。它是关系到整个系统能否按期完工的一个关键因素。因此我们要选择那些可移植性程度高的操作系统。从而避免操作系统难以向硬件移植而带来的种种困难,加速系统的开发进度。

(3) 操作系统的内存要求。

均衡考虑是否需要额外花钱去购买 RAM 或 EEPROM 来迎合操作系统对内存的较大要求。有些操作系统对内存的要求是 target dependent。如 Tornado/VxWorkx, 开发人员能按照应用需求分配所需的资源,而不是为操作系统分配资源。从需要几 K 字节存储区的嵌入设计到需求更多的操作系统功能的复杂的高端实时应用,开发人员可任意选择多达 80 种不同的配置。

(4) 开发人员是否熟悉此操作系统及其提供的 API。

(5) 操作系统是否有提供硬件的驱动程序,如网卡等。

(6) 操作系统是否具有可剪裁性,即能否根据实际需要进行系统功能的剪裁。有些操作系统具有较强的可剪裁性,如嵌入式 Linux、Tornado/VxWorks 等

等。

#### (7) 操作系统的实时性<sup>[14]</sup>。

实时性分为：软实时和硬实时。有些嵌入式操作系统只能提供软实时，如 Windows CE。作为微软大名鼎鼎的“维那斯”，Microsoft Windows CE.NET 是 32 位，Windows 兼容，小内核，可伸缩实时操作系统，满足大部分嵌入式和非嵌入式应用的需要。

针对汽车电子对操作系统的要求以及可行性，我们主要比较了 Windows CE 和嵌入式 Linux 两种操作系统。在现在诸多的嵌入式操作系统中，Embedded Linux 和 Windows CE .NET 的应用都是十分广泛。他们有着各自的优缺点，适用于不同的需求环境中。

Embedded Linux 是多任务多进程，有一定实时性的操作系统<sup>[15]</sup>。他最主要的特点就是源代码开放，不存在黑箱技术。Embedded Linux 是由一个 Kernel 和一些根据需要进行定制的系统模块组成。Kernel 很小，一般只有几百 KB，加上其它的系统模块，所需的空间也很小。在开发工程中，技术细节和错误处理都可以完全掌握，但是付出的代价是开发的难度和复杂度增加。

Windows CE .NET 是由 Microsoft 公司开发的新一代的嵌入式操作系统。他是一个抢先式多任务并具有强大通信能力的 Win32 嵌入式操作系统。他有很丰富的系统模块用来定制各种应用环境下所需要的操作系统。并且它使用 Platform Builder 图形界面的开发工具实现系统模块的裁减，开发效率很高。他使用 EVC 等工具开发，和普通的 Windows 程序开发一脉相承，支持 SDK, MFC, ATL 等 Windows 类库。对于有 Windows 开发经验的开发者来说，能够很快地开发 Windows CE .NET 环境下的应用程序<sup>[16]</sup>。并且，Microsoft 公司能够提供强大的技术支持。

从操作系统内核的角度看，Windows CE 具有灵活的电源管理功能，包括睡眠/唤醒模式。在 Windows CE 中，还使用了对象存储(Object Store)技术，包括文件系统、注册表及数据库。它还具有很多高性能、高效率的操作系统特性，包括按需换页、共享存储、交叉处理同步、支持大容量堆(Heap)等<sup>[17]</sup>。

Windows CE 拥有良好的通信能力。它广泛支持各种通信硬件，亦支持直接的局域网连接以及拨号连接，并提供与 PC、内部网以及 Internet 的连接，包括用于应用级数据传输的设备至设备间的连接。

在提供各种基本的通信基础结构的同时，Windows CE 还提供与 Windows 9x



/ NT 的最佳集成和通信。Windows CE 的图形用户界面相当出色。它拥有基于 Microsoft Internet Explorer 的 Internet 浏览器, 此外, 还支持 TrueType 字体。开发人员可以利用丰富灵活的控件库在 Windows CE 环境下为嵌入式应用建立各种专门的图形用户界面。Windows CE 甚至还能支持诸如手写体和声音识别、动态影像、3D 图形等特殊应用<sup>[18]</sup>。

Embedded Linux 在通讯方面的支持远不如 Windows CE .NET 全面丰富。Linux 的 Web 浏览和多媒体相对比较弱。Linux 厂商常常为这类功能收取附加费用, 而 Windows CE .NET 则自带这些功能。在与 PC 机的互连方面, 由于 Windows 操作系统在 PC 机上的垄断地位, 所以 Windows CE .NET 在这方面有着得天独厚的优势。

所以在注重界面, 通讯, 和多媒体技术的应用环境中, Windows CE 有着 Embedded Linux 不可比拟的优势。

同时考虑到我们开发人员对 Windows 的 API 较熟悉, 在 Windows CE 平台上相对开发周期较短, 风险较低。

#### 3.3.2 编程语言的选择

在选择编程语言的时候需要考虑一下几个因素。

##### (1) 通用性。

随着微处理器技术的不断发展, 其功能越来越具体, 种类越来越多, 但不同种类的微处理器都有自己专用的汇编语言。这就为系统开发者设置了一个巨大的障碍, 使得系统编程更加困难, 软件重用无法实现, 而高级语言一般和具体机器的硬件结构联系较少, 比较流行的高级语言对多数微处理器都有良好的支持, 通用性较好。

##### (2) 可移植性程度。

由于汇编语言和具体的微处理器密切相关, 为某个微处理器设计的程序不能直接移植到另一个不同种类的微处理器上使用, 因此, 移植性差; 而高级语言对所有微处理器都是通用的, 因此, 程序可以在不同的微处理器上运行, 可移植性较好。这是实现软件重用的基础。

##### (3) 执行效率。

一般来说, 越是高级的语言, 其编译器和开销就越大, 应用程序也就越大

越慢。但单纯依靠低级语言，如汇编语言来进行应用程序的开发，带来的问题是编程复杂、开发周期长。因此存在一个开发时间和运行性能间的权衡。

#### (4)可维护性。

低级语言如汇编语言，可维护性不高。高级语言程序往往是模块化设计，各个模块之间的接口是固定的。因此，当系统出现问题时，可以很快地将问题定位到某个模块内，并尽快得到解决。另外，模块化设计也便于系统功能的扩充和升级。

#### (5)基本性能。

在嵌入式系统开发过程中使用的语言种类很多，比较广泛应用的高级语言有：Ada、C/C++、Modula-2 和 JAVA 等。Ada 语言定义严格，易读易懂，有较丰富的库程序支持，目前在国防、航空、航天等相关领域应用比较广泛，未来仍将在这些领域占有重要地位。C 语言具有广泛的库程序支持，目前在嵌入式系统中是应用最广泛的编程语言，在将来很长一段时期内仍将在嵌入式系统应用领域占有重要地位。C++ 是一种面向对象的编程语言，目前在嵌入式系统设计也得到了广泛的应用，如 GNU C++。Visual C++，是一种集成开发环境，支持可视化编程，广泛应用于 GUI 程序开发。但 C 与 C++ 相比，C++ 的目标代码往往比较庞大和复杂，在嵌入式系统应用中应充分考虑这一因素。Modula-2 定义清晰，支持丰富，具有较好的模块化结构，在教学科研方面有较广泛的应用。虽然该语言的开发应用一直比较平缓，但近两年在欧洲有所复苏。Java 语言相对年轻，但有很强的跨平台特性，目前发展势头较为强劲。它的“一次编程，到处可用”的特性使得它在很多领域倍受欢迎。随着网络技术和嵌入式技术的不断发展，Java 及嵌入式 Java 的应用也将越来越广泛。

综合各因素我们在开发时主要使用 C 和 C++。在 WindowsCE.net 下进行开发主要使用 EVC。

### 3.3.3 基本编程规范

编程过程中严格遵循 VC++ 编程规范，可大大加强代码的可读性、容错性、可重用性以及可扩展性。遵循的编程原则择其要者如下：

(1) 程序结构清晰，单个函数的程序行数不超过 100 行，使用子函数实现单个功能。

- (2) 各函数的功能，主要变量的定义，典型算法都有详细的注释。
- (3) 保持结构清晰，遵循面向对象的程序设计原则。
- (4) 变量定义具有一定的实际意义，能体现出变量类型和代表的意义。

### 3.4 本章小节

作为系统设计，应该考虑所选的硬件及操作系统能够完成所需要的实时处理任务，作为研究还要考虑到系统功能的扩展性。

本章主要介绍了硬件平台的选择原则、嵌入式操作系统和编程语言的选择原则，通过综合考虑应用领域、用户需求、成本、开发难易程度等因素，最后选择通用的 ARM9 内核处理器 S3C2410 作为主控芯片，选择 WindowsCE.NET 作为嵌入式实时操作系统，选用 EVC 作为软件开发工具。

## 第四章 WinCE 操作系统二次开发

引入操作系统是 ROVER—FCV 智能仪表的一个重要特征和标志。在该仪表中使用微软的嵌入式操作系统 Windows CE，它是一个可定制的、稳定的、实时的多任务操作系统。

### 4.1 Windows CE.NET 简介

#### 4.1.1 Windows CE 简介

微软 Windows CE 是一个开放且多样化的 32 位嵌入式操作系统。其设计目的是满足广泛的智能设备的需求，例如从诸如工业控制器、通信集线器、和收款机系统 (POS) 等企业工具到诸如摄影机、电话和家庭娱乐设备等电子消费产品，为自动控制、视听娱乐、移动计算、终端、数据收集、数据共享以及联网等各个应用领域提供一个稳定、实时及多任务的操作系统。一个典型的 Windows CE 嵌入式系统常被定制为一个能达到特定目的，一个轻量化及内置的能果断反应系统中断的操作系统。Windows CE.NET 目前支持四大系列的 CPU 架构 (ARM、MIPS、SHx、X86) 及超过两百种品牌的 CPU，同时因为轻量化及高度的模块化及客制化，Windows CE.NET 被广泛地应用于设计各种移动地、小型地以及各种工业控制器中。

Windows CE 提供给程序开发者标准 Win-32API，ActiveX 控件，MSMQ(message queuing)，COM(Component Object Model) 接口，ATL(Active Template Library) 和 MFC 链接库等开发环境。ActiveSync 提供台式计算机和嵌入式装置之间的网络连接。Windows CE 对多媒体、通信 (TCP/IP, SNMP, TAPI 等) 还有安全性提供内置的支持。种种地整合应用程序，包括了小型的 IE 浏览器，针对小型的 Outlook 收信信箱的客户程序，还有小型地允许用户扩展并自定义现行地系统地 Word expose 对象，和扩展应用程序的功能<sup>[19]</sup>。

#### 4.1.2 Windows CE 操作系统的基本体系结构<sup>[20]</sup>

Windows CE 是由许多离散模块构成的，每一模块都提供特定的功能。这些模块中的一部分被划分成组件。组件使 Windows CE 变得非常紧凑（只占不到 200KB 的 RAM），因此只占用了运行设备所需的最小的 ROM、RAM 以及其它硬件资源。

Windows CE 包含提供操作系统最关键功能的 4 个模块：内核模块；对象存储模块；图形、窗口和事件子系统 (GWES) 模块以及通讯模块。Windows CE 还包含一些附加的可选择模块。

内核。内核是操作系统的核心，通过 Coredll 模块表示。它提供在所有设备中都出现的基本操作系统功能。内核负责内存管理、进程管理以及特定文件管理等功能。它还管理虚拟内存、调度、多重任务处理以及例外处理等。内核从 ROM 执行，支持多线程，抢先式多任务，支持八个线程优先级、优先级继承和纠正优先级倒置。

对象存储。包括数据库、注册表和文件系统在内的对象存储方式可将用户数据和应用程序数据存入文件或注册表。

GWES (图形、窗口和事件子系统)。GWES 是用户、应用程序和操作系统之间的图形用户接口。GWES 通过处理键盘、笔针等动作来接受用户输入，并选择传送到应用程序和操作系统的信息。GWES 通过创建并管理在显示设备和打印机上显示的窗口、图形以及文本来处理输出。

通讯。通讯组件提供对下列通讯硬件和数据协议的支持：

- (1) 串行 I/O 支持。
- (2) 远程访问服务 (RAS)。
- (3) 传输控制协议/Internet 协议 (TCP/IP)。
- (4) 局域网。
- (5) 电话技术 API (TAPI)。
- (6) Windows CE 的无线服务。

除了上述主要模块之外，还有其它的可选组件和模块。这些组件和模块主要有：

- (1) 设备管理器和设备驱动程序。
- (2) 多媒体（声音）支持模块。
- (3) COM 支持模块。

### 4.1.3 Windows CE.NET 操作系统的实时性<sup>[20]</sup>

Windows CE.NET 是一个硬实时操作系统,是组成一个硬实时系统的重要元素之一,它提供了以下的功能和支持来帮助应用程序开发和系统整合工程师架构完整可靠的硬实时系统。

256 个等级的线程优先权(thread priority):提供嵌入式系统对线程排班的控制能力更弹性化。

套叠的中断(Nested interrupt):能暂停优先权低的中断服务,让优先权高的中断服务先执行。

每个线程的执行时间量(Pre-thread quantum):应用程序可以设置线程排班时所分配到的执行时间量。

优先权改变或倒置(Priority inversion):当系统资源例如 mutex、关键区和信号量等被优先权低的线程取得占用时,如果同时有其他的优先权高的线程也在等待取得同样的资源才能执行,这将会造成某种程度的竞争现象而严重影响系统整体的效率。因为优先权低的线程必须执行完毕才会释放这些资源,而此时的执行权是由优先权高的线程所取得,但是优先权高的线程却必须取得这些资源后才能执行,这就造成了互不相让而使系统迟滞甚至死机。为了解决这个问题,Windows CE 提供了优先权改变或倒置的功能,当系统核心发现这种现象时,会提高原本优先权低的线程的优先级,让它高于原本优先权高的线程,以取得执行权后赶快执行完并释放资源,好让其它也需要这些资源的线程接着执行。

### 4.1.4 Windows CE.NET 操作系统的开发套件

Microsoft Windows CE.NET 是 Windows CE 3.0 的下一个版本,从 2002 年推出以后,目前已有 Windows CE.NET4.0, Windows CE.NET4.1, Windows CE.NET4.2 以及 Windows CE.NET5.0 等版本。是一个功能强劲的实时嵌入式操作系统,提供了众多强大工具,允许用户利用它快速开发出下一代的智能化小体积连接设备。借助于完善的操作系统功能和开发工具,Windows CE.NET 为开发人员提供了构建、调试和部署基于 Windows CE 的定制设备所需的一切特性。WindowsCE.net 操作系统平台开发工具 Platform Builder 是一个完全集成的开发环境(IDE),并且包括一个软件开发工具包(SDK)导出工具,Platform Builder

主界面如图 4.1 所示。Windows CE .NET 支持 Microsoft embedded Visual C++ 和 Microsoft Visual Studio.NET, 为面向 Microsoft .NET Compact Framework (Microsoft .NET Framework 的一个子集) 的 Web 服务和应用程序开发提供了一个完整的开发环境。利用这些工具, 开发人员可以迅速开发出能够在最新硬件上运行各种应用程序的智能化设计<sup>[3, 3]</sup>。



图 4.1 Platform Builder 主界面

## 4.2 WindowsCE. net 的二次开发

### 4.2.1 Windows CE 二次开发的工具

Platform Builder 是为基于 Microsoft Windows CE .NET 操作系统构建定制嵌入式平台而提供的集成开发环境 (IDE)。Platform Builder 附带有设计、创建、构建、测试和调试一个基于 Windows CE 的平台所需的所有开发工具。

### 4.2.2 Windows CE 的二次开发流程

Windows CE 的开发流程大致可以分为：

**1. 内核裁减，平台 NK 的生成，通过以下步骤完成：**

- (1) OAL、Driver、shell 的定制；
- (2) 系统模块的定制；
- (3) Boot loader 的定制。

**2. 应用程序的开发，通过以下步骤完成：**

- (1) SDK 的发布、安装、联机调试环境的建立；
- (2) 中间模块的开发；
- (3) 应用界面的开发。

**3. 产品的生成，可以两种形式发布：**

- (1) 应用程序与 NK 分开发行；
- (2) 应用程序集成到 NK 中发行。

实际上，Windows CE 并不是一个通用的安装版操作系统，裁减 Windows CE 操作系统是 Platform Builder 的重要工作，只有经过合适的裁减，添加适当的组件，Windows CE 才能在车载计算机上正常高效的运行。整个 Windows CE.net 应用平台的开发过程如图 3.2 所示。在本课题中我们走的路线是：从硬件 OEM 厂商处得到 BSP 和 CE 镜像，然后使用 Platform Builder 和 BSP 定制 Windows CE 并导出自己的 SDK，再用 EVC++ 开发应用程序，如图 4.2 中虚线所示。



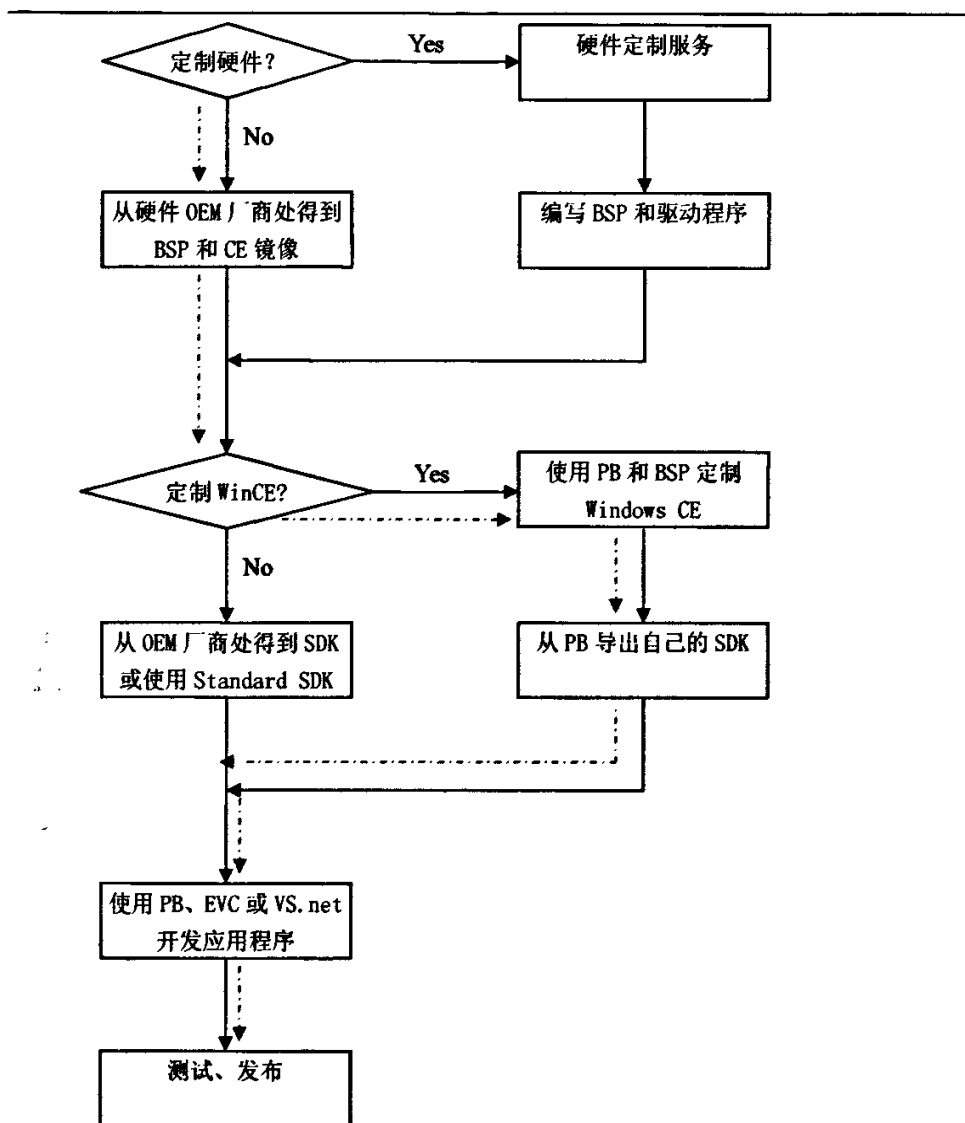


图 4.2 Windows CE.net 应用平台的开发流程图

### 4.2.3 使用 Platform Builder 架构 WindowsCE.net 平台

首先在 windows 操作系统中安装 Platform Builder4.2, 然后在 Platform Builder4.2 中定制 WinCE 操作系统。

#### 1. 加入 BSP

想要定制一个完整的 WindowsCE.net 系统，BSP（板级开发包）是必须要使用的。BSP 对各种板子的硬件功能提供了统一的接口，它包括硬件初始化、中断的产生和处理、硬件时钟和计时器管理、局域和总线内存地址映射、内存分配等等。每个板级支持包包括一个 ROM 启动（Boot ROM）或其他启动机制。

BSP 主要包括 Boot Loader，设备驱动程序和各种配置文件，其通常的结构如图 4.3。

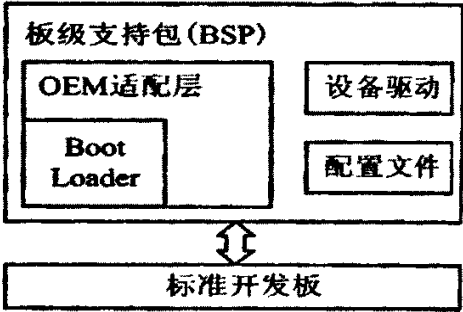


图 4.3 板级支持包 (BSP) 的结构图

使用 BSP 也很简单，只要把从硬件 OEM 厂商处得到 BSP 文件复制到 Platform Builder4.2 安装文件夹下的 PLATFORM 文件夹下，再使用 Platform Builder4.2 中的 BSP 管理器把 BSP 导入就可以了。

## 2. 使用 Platform Builder4.2 创建一个基于新 BSP 的 Windows CE 平台

(1) 创建 Mobile Handheld 类型的 Windows CE 平台，打开 Platform Builder，从 File 菜单，选择 New Platform，打开 New Platform Wizard，如图 4.4 所示。



图 4.4 New Platform Wizard 界面

(2) 点击界面下方的 Next 按钮, 进入 New Platform Wizard-step2 界面  
选择“选择 BSP”选项, 在“Available BSPs”列表中选择刚才加入的 BSP (Available BSP 列表包含所有已经安装的板级支持包 BSP, 比如 Emulator x86, 这是模拟器), 然后选择 Next。

(3) 在“Available Configurations”列表中选择“Mobile Handheld”。  
Available Configurations 是微软已经预先定制好的几个类型的平台, 包括网关, IP 电话, 机顶盒等, 我们可以选择其中一个, 然后稍作修改。如果 Available Configurations 里面没有我们与我们想近的平台, 我们可以在这一步选择“Custom Configuration”从 0 开始完全自定义。这里我们选择“Mobile Handheld”, 如图 4.5 所示。然后在“Platform name”文本框内为平台起个名字, 也可以在“Location”文本框中更改平台的目标路径。完成后选择 Next。

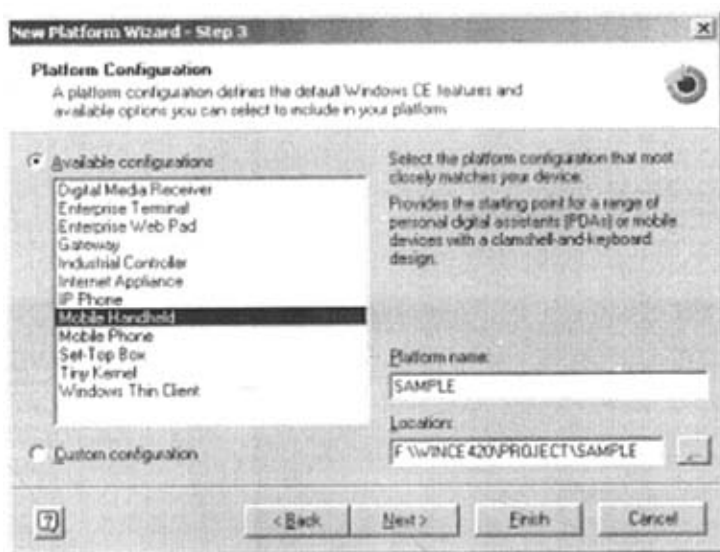


图 4.5 Platform Configuration 选择界面

(4) 在“Application & Media”和“Networking & Communications”两步选择需要包含的 feature。然后选择 Next, 逐步设置, 直到完成。

(5) 选择: Build -> Build Platform 来构建平台。整个编译过程大概分为四个阶段, 根据机器配置和选择的内容不同, 大概需要 20-30 分钟。如果编译

出错，可以到安装 Platform Builder 的目录（默认为 C:\WINCE420）下查看 build.log 和 build.err。如果编译成功，没有 Error。就可以在模拟器下运行 Windows CE 了。选择 Target -> Download\Initialize，第一次，有可能会弹出对话框（如下图），让我们配置信息，这个时候，在两个对话框里面都选择 Emulator。然后确认。就可以在模拟器中运行我们构建的 Windows CE 操作系统。

### 3. 特性裁剪

本部分介绍如何对已经建立的 WindowsCE.net 平台进行特性修改和自定义裁减。Platform Builder 的工作区界面如图 4.6 所示，左面的一栏是 Feature 视图，显示的是当前的平台中所选择的特性（feature）。右面的一栏是 Catalog 视图，是所有可以选择的特性列表。当我们使用新建平台向导建立一个新的平台之后，就可以在此基础上根据自己的需求，对平台进行裁减。

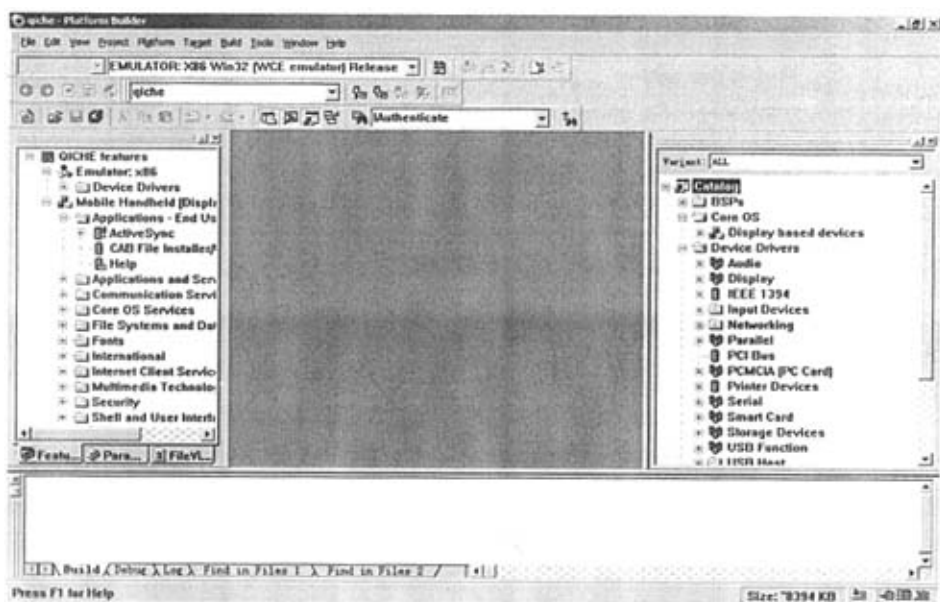


图 4.6 WindowsCE.net 平台的 feature 和 Catalog 视图

值得注意的是有些功能之间有依赖关系，也就是某项功能只有在另外一项功能可用时才可以正常运行。在以前的 PB 版本中，要人工解决依赖问题。但是在 Platform Builder 4.2 中，系统会帮我们解决功能之间的依赖。当我们添加

一项功能时，系统会自动把其依赖于的功能自动添加进去。要查看某个 feature 的依赖项，如图 4.7 所示，可以在 Feature 视图中选中该 feature，鼠标右键单击，选择 Feature Dependencies。

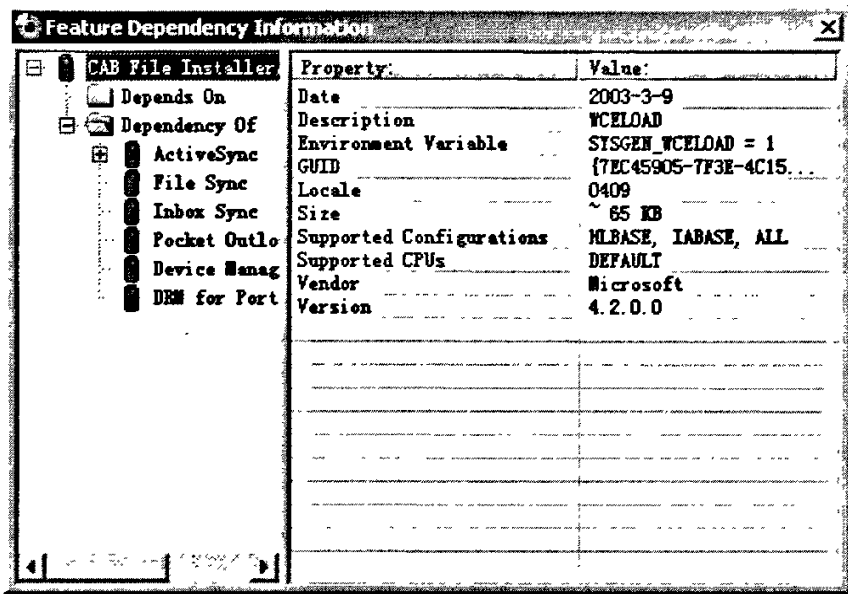


图 4.7 特征相关信息界面

如果要删除一个功能，可以在左面的 Feature 视图中选中一项，然后按 Del 键，或者鼠标右键单击，然后选择 Delete。如果有别的项依赖与要删除的项，PB 会进行提示。

如果要添加一个功能，可以在右面的 Catalog 视图中选中某一项，然后鼠标右键单击，选择 Add to Platform。

裁减结束后，必须要重新编译操作系统。选择 Build -> Build Platform。

WINDOWS CE 的 PLATFORM BUILDER 生成 OS IMAGE 的步骤共 4 个阶段：Sysgen 阶段，Feature Build 阶段，Release Copy 阶段，Make Image 阶段。

(1) Sysgen 阶段：PB 负责将所有选定的 Feature 对应的 VARIABLE 搜集起来。根据创建 PLATFORM 后对应的 Feature。生成针对 PLATFORM 的头文件。导入一堆库文件。在 Sysgen 阶段最后一步是把 BSP BUILD 好。

(2) Feature Build 阶段：PB 将其他不属于 OS 组件的 Feature BUILD 好。

(3)Release Copy 阶段: PB 将所有的编译好的文件,无需编译但是属于指定 Feature 的文件(例如帮助文件等)复制到 Release 目录下。

(4)Make Image 阶段: PB 生成最终的 WINDOWS CE IMAGE 文件。

这样,我们就可以在 ReleaseDir 里面得到相应的 NK 文件。按照硬件提供商的说明,把它烧到 flash ROM 或 download 到 ROM 里面就可以运行 Windows CE 了。图 4.8 是模拟器下的 windows CE 界面。

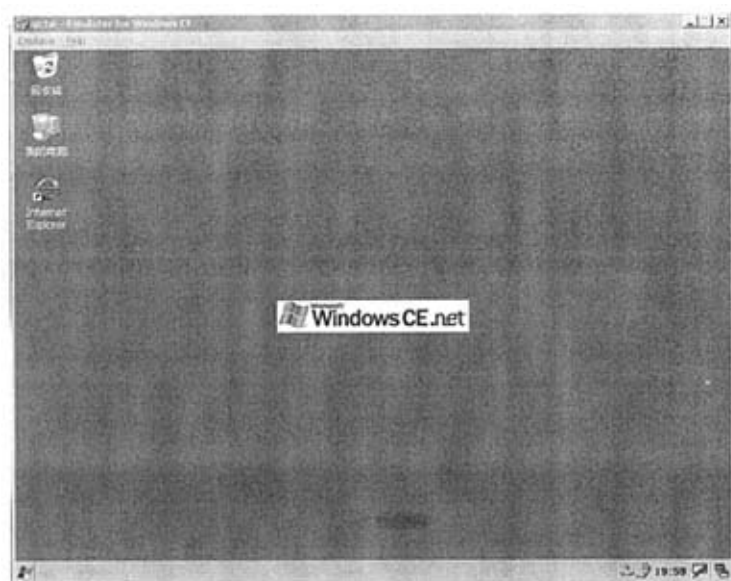


图 4.8 模拟器下的 windows CE 界面

#### 4.2.4 定制 ARM 的 WindowsCE..net 平台

##### 1. 设计目标

仪表的目标功能有包括用虚拟数字指针显示车速、动画显示功率流图和其他参数以及故障代码的翻页显示;此外还需进行 TCP/IP 网络通讯和 RAS 远程拨号实验。

## 2. WindowsCE..net 包含的组件

根据设计目标 Windows CE.net 除了核心操作系统服务组件, 还应有: c 和 c++库, 组件服务(Component Object Model (COM, 组件对象模型)), Microsoft Message Queuing (MSMQ, Microsoft 消息队列) Microsoft Foundation Classes (MFC, Microsoft 基础类库), ActiveSync, 远程桌面连接, TCP/IP 网络连接, 广域网(WAN) 拨号连接、点对点 and 电话 API, 文件系统和数据存储。此外还加入了基本的多媒体服务, 包括 WMA 和 MP3 本地播放、流媒体、WMV 以及 MPEG-4, 和多媒体组件, 包括音频、数字版权管理 (DRM)、DirectX 8、Windows Media 9 Series 编解码器。

## 3. 构建 WindowsCE..net

(1) RAM 系统硬件供应商提供的配套资料光盘中有 BSP 文件, 将其中的内容拷贝到 Platform Builder 4.2 安装目录的 PLATFROM\...目录下, 再打开 Platform Builder 4.2, 用 BSP 管理器把刚才拷贝的 BSP 导入。

(2) 按照上一节讲述的逐步完成 WindowsCE..net 的裁剪并生成镜像下载到 YFDVK-2410 的 ROM 中。需要注意的是, 在 New Platform Wizard—Step2 对话框中, 选择新安装的 BSP—SMDK2410, 如图 4.9 所示。然后分别选择需要包含的特真。这样就定义好了我们需要的操作系统平台。



图 4.9 New Platform Wizard—Step2 对话框界面

(3) 将定义好的平台编译生成操作系统镜像文件，通过以太网接口下载到 ARM 系统中。通过了调试就可以并其上进行应用程序的调试和运行。

### 4.2.5 建立当前平台的 SDK

已经建立好了正常启动并已经调试成功的平台，接下来就是使用 Platform Builder 建立这个平台的软件开发包(SDK)，这样就可以使用这个 SDK 专门为该平台开发应用程序。SDK 支持的开发语言特性一共有两个选择，是 EVC 和 .NET，我们要在 EVC 下开发应用软件，所以就选择 EVC。生成 SDK 后，进行安装，就可以开发应用软件了。

## 4.3 本章小节

本章讲述了嵌入式实时操作系统 WinCE 的特点、结构及应用等。并针对 ARM9 内核处理器 S3C2410，结合需要实现的功能定制了自己的 WinCE 操作系统，操作系统镜像大小适中，运行效率高。用于应用软件的样机调试和运行，并在 ROVER 燃料电池轿车上应用。



## 第五章 液晶仪表界面软件设计

ROVER—FCV 智能仪表的主画面是中央液晶显示屏，它显示汽车的包括车速在内的大部分信息，本章设计的软件就是用于在液晶屏上显示这些信息。

### 5.1 系统需求分析

为燃料电池汽车配备相应的仪表，需要满足用户实时了解汽车运行状态、查询相关参数，要求系统具有以下功能。

- (1)随时能让驾驶员方便快捷的了解车速信息。
- (2)记录并显示里程；显示续驶、里程档位信息、环境温度。
- (3)察看动力系统参数，包括燃料电池、蓄电池、电机等。
- (4)察看燃料电池、蓄电池、DC/DC 转换器和轮毂电机之间的能量供给状况。
- (5)记录并察看汽车故障的故障代码和简单描述。
- (6)显示蓄电池 LOC 低、风窗洗涤液面低、刹车片磨损、刹车油面低警告信号和雾灯开启信号。

### 5.2 系统设计

#### 5.2.1 系统总体设计

根据需求分析，设计系统框架。仪表信息系统由数据接收、车速转换和指针模拟、功率流图、动力系统参数显示、故障代码存储和显示、里程计算存储与显示、档位和警告信号指示等部分组成。主要模块具体功能如下。

##### 1. 数据接收模块

该模块负责从串口接收整车控制器发送过来的数据，并按照数据通讯协议进行处理。

##### 2. 车速转换和指针模拟模块

该模块负责将车速转换为指针角度并模拟显示在仪表盘上。

### 3. 功率流图模块

根据相关数据，以动画的形式显示动力系统各部分之间的能量流动方向。

### 4. 动力系统参数显示

该模块分页分类显示动力系统运行状态的重要数据，供驾驶员查看。

### 5. 故障代码存储和显示

该模块负责汽车故障代码的存储，新故障代码的添加，已修复故障的故障代码删除，故障说明的查询和显示。

### 6. 里程计算存储与显示

该模块负责根据车速计算里程并以文件的形式存储在 ROM 中，每次开机时读出最近一次存储的历程数，显示在仪表上。

### 7. 档位和警告信号指示模块

根据接收到的档位信号和各警告信号用醒目的方式将其显示在仪表上。各模块层次结构如图 5.1 所示。

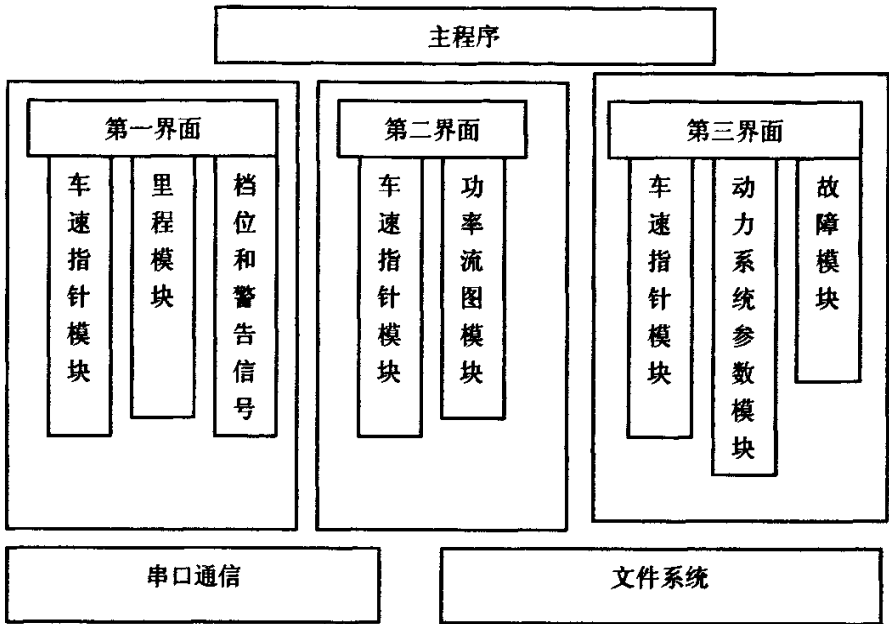
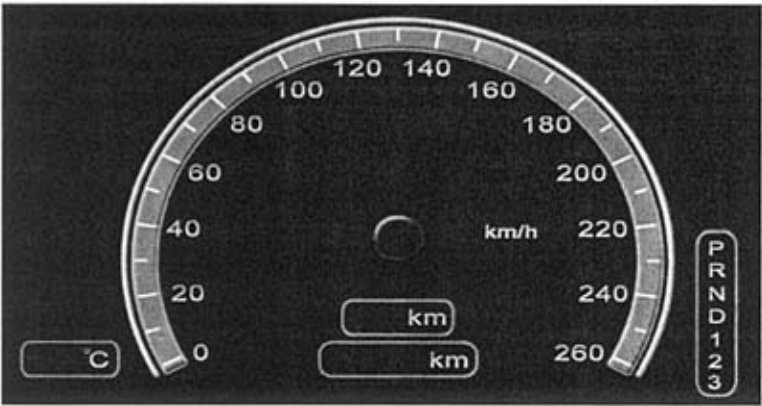


图 5.1 各主要模块层次结构图

仪表显示系统的显示屏为 10.4 英寸的 TFT 液晶屏，和传统机械式指针仪表上车速表的面积相当。为了能显示众多的信息，依据信息的重要程度和类别使用分屏显示。总共分为三屏，第一屏是车速界面，第二屏是功率流图界面，第三屏是参数界面，如图 5.2 到图 5.7 所示。开机时默认显示车速界面，通过按键可在三个界面之间依次切换。从途中可以看到，在第二和第三个界面上一部分指针的形式指示车速，突出仪表指示车速的主体地位，方便用户在查看其他参数时也能醒目的看到车速。



5. 2 车速界面背景图

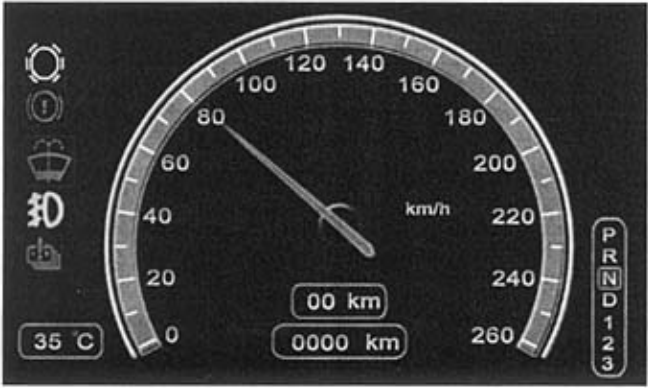


图 5.3 车速界面效果图



图 5. 4 功率流图界面

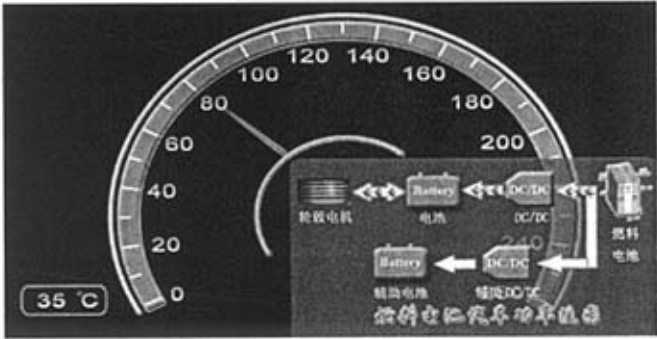


图 5. 5 功率流图效果图

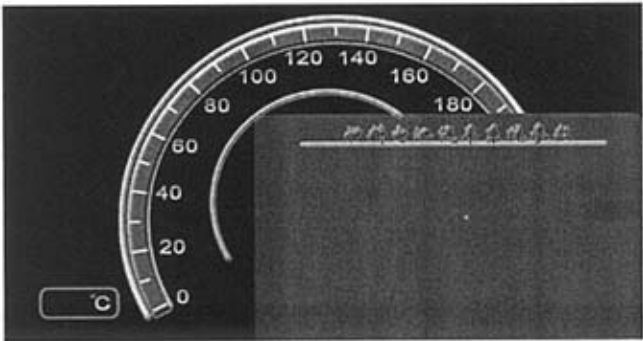


图 5. 6 系统参数背景图

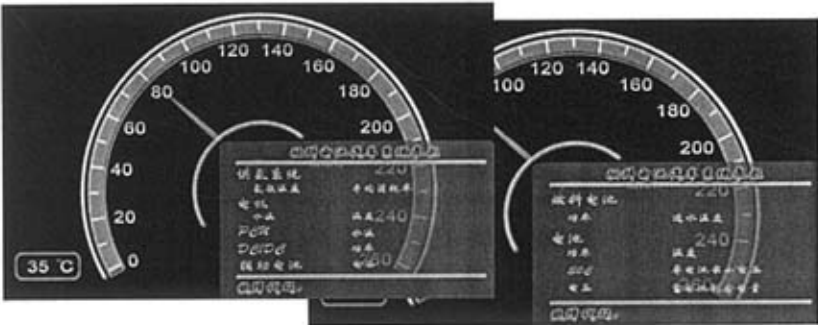


图 5. 7 系统参数效果图

### 5.2.2 系统设计目标

汽车仪表是用户与汽车交流的重要窗口，它给用户重要信息，帮助用户决策，保障行车安全。通过本系统可以达到以下目标。

- (1) 系统运行稳定，安全可靠。
- (2) 界面设计美观大方。友好实用。
- (3) 数据查看灵活、方便、快捷、准确。
- (4) 数据接收和存储安全可靠。

### 5.2.3 系统运行环境

- (1) 系统开发平台：Embedded Visual C++4.0。
- (2) 运行平台：Windows CE.net。

### 5.2.4 串口数据通讯协议

串口数据通讯协议由头协议、中间协议和尾协议三部分组成。各协议定义如下。

#### 1. 头协议

头协议有三字节，它们是起始标志符(0x33)、源地址(0x0A)和数据类型(0x00/0x10/0x11)。数据类型中 0x10 表示该数据帧为汽车运行参数数据帧；0x11 为故障代码帧；0x00 为控制帧。

#### 2. 中间协议

中间协议分三种情况，数据帧包含 29 字节，故障代码帧 18 字节，控制帧 4 字节。

#### 3. 尾协议

尾协议一字节，数据为 0xEE, 代表帧结尾。协议详细信息请参看附录 A。

### 5.3 主要功能模块设计

仪表显示系统采用对话框<sup>[21]</sup>结构，各界面效果上文已有描述。下面介绍主要功能模块的程序源代码设计。

#### 5.3.1 串口通讯模块

RAM 系统通过串口从 XC164 获得数据，串行通信是指外设和计算机之间使用一根数据线（另外还需要地线，可能还需要控制线），数据在一根数据线上位一位地进行传输，每一位数据都占据一个固定的时间长度<sup>[22]</sup>。串口通讯流程如图 5.8 所示。在 WINCE 中可直接使用串口编程 API 函数来编写串口通信程序<sup>[23]</sup>。

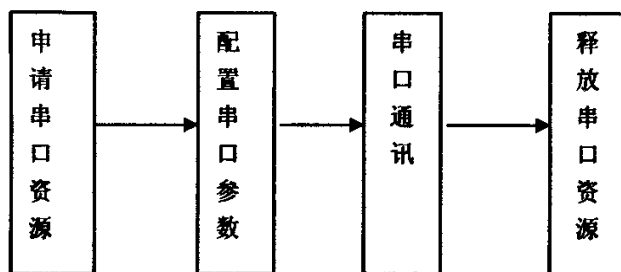


图 5.8 串口通信流程图

##### 1. 打开串口

应用程序要使用串口进行通信，必须在使用之前向操作系统提出资源申请要求（打开串口），通信完成后必须释放资源（关闭串口）。在 WINCE 中，通过 CreateFile 函数打开串口。需要注意的是，在串口名之后必须加一个冒号(:)。如果打开串口成功，就返回打开串口的句柄，否则就返回 INVALID\_HANDLE\_VALUE<sup>[24]</sup>。打开串口代码如下：

```

HANDLE MyOpenPort()
{
    HANDLE hCom;
    hCom= CreateFile (TEXT("COM1:"),           // 打开 COM1
        GENERIC_READ | GENERIC_WRITE,        // 允许对串口读和写
        0,                                     // 共享模式为独占方式
        NULL,

```

---

```
OPEN_EXISTING,           // 打开而不是创建
0,
NULL); }

```

## 2. 初始化串口

在使用串口时，还必须配置好串口的波特率、奇偶校验和数据位等参数。WINCE 中使用函数 SetCommState 函数设置串口，该函数有两个参数，分别是已打开串口的句柄和一个指向 DCB 结构的指针<sup>[26]</sup>。以下代码将串口初始化为数据位 8，一位停止位，无奇偶校验，波特率 38400。

```
BOOL Initialize_Port(HANDLE hCom)
{
    DCB PortDCB;
    PortDCB.DCBlength = sizeof (DCB);
    PortDCB.BaudRate = 38400;           //设定波特率
    PortDCB.fBinary = TRUE;             //二进制通讯
    PortDCB.fParity = TRUE;             //进行奇偶校验
    PortDCB.fOutxCtsFlow = FALSE;       // No CTS output flow control
    PortDCB.fOutxDsrFlow = FALSE;       // No DSR output flow control
    PortDCB.fDtrControl = DTR_CONTROL_ENABLE; // DTR flow control
    PortDCB.fDsrSensitivity = FALSE;     // DSR sensitivity
    PortDCB.fTXContinueOnXoff = TRUE;     // XOFF continues Tx
    PortDCB.fOutX = FALSE;               // No XON/XOFF out flow
    PortDCB.fInX = FALSE;                // No XON/XOFF in flow
    PortDCB.fErrorChar = FALSE;          // Disable error replacement
    PortDCB.fNull = FALSE;               // Disable null stripping
    PortDCB.fRtsControl = RTS_CONTROL_ENABLE; // RTS flow control
    PortDCB.fAbortOnError = FALSE;       // Do not abort reads/writes on
    PortDCB.ByteSize = 8;                //字符为 8 位
    PortDCB.Parity = NOPARITY;           //无奇偶校验
    PortDCB.StopBits = ONESTOPBIT;       // 一位停止位
}

```



### 3. 从串口读数据

使用 ReadFile 函数读取串口数据。由于从串口读写数据的速度表较慢，因此一般不再主线程中读写大量数据，这样会阻塞主线程中处理主窗体的消息队列，所以建立一个单独的线程[26]来读写数据。读串口数据之前需要用 SetCommMask () 函数设置串口事件来监视指定通信端口上的事件，该函数包含两个参数，一个是由 CreateFile 函数返回的标志通讯串口的句柄，另一个是指定的通讯事件。接下来用 WaitCommEvent() 函数来等待串口上我们利用 SetCommMask () 函数设置的事件。当检测到 EV\_RXCHAR 时间发生时，用 ReadFile 函数读取串口数据。

数据从串口读出后，为了准确无误的按照通讯协议辨别数据帧类型和丢弃错误的帧，建立了“CComBuf”类对数据进行识别。在该类中有一个数据缓冲区，对数据采取先进先出的操作。用定时器定时，每一百毫秒从缓冲区中读一次数据，从当前位置开始逐个比较缓冲区里的数据，符合帧协议，则读出并赋给已定义好的数据结构类，然后在缓冲区中定义新的数据位位置。

### 5.3.2 车速转换和指针模拟模块

车速指示采用虚拟指针模仿传统仪表物理指针的方式，背景和车速刻度采用位图，WINCE 提供了功能强大的图形设备接口 (GDI)。从串口得到车速后，还需将其转换位刻度盘上对应值处指针的角度。对于圆弧形、刻度均匀的仪表盘刻度，知道车速后可根据 8 个参数来确定指针的位置和长度，它们定义如下：

```
POINT m_ptMeterCenter;    //指针转动中心点坐标 (x, y)
int m_dMinValue;          //刻度盘车速最小值，一般为 0
int m_dMaxValue;          //刻度盘车速最大值，罗孚车为 180km/h
int m_nStartAngleGrad;    //刻度起始角度，即最小车速对应的角度，
int m_nEndAngleGrad;      //刻度终止角度，此处为 324
double m_dCurrentValue;   //当前车速
int m_nRadiusFrame;       //指针顶端至指针中心半径
int m_nCenterRadius;      //指针末端至指针中心半径
```

如图 5.9 所示，以刻度盘中心即指针转动中心为坐标原点，x 正半轴向左，y 正半轴向上，刻度起始角度的计算是以 x 正半轴为起始为止，逆时针方向旋转。在该仪表中，起始角为 216°，终止角为 324°。

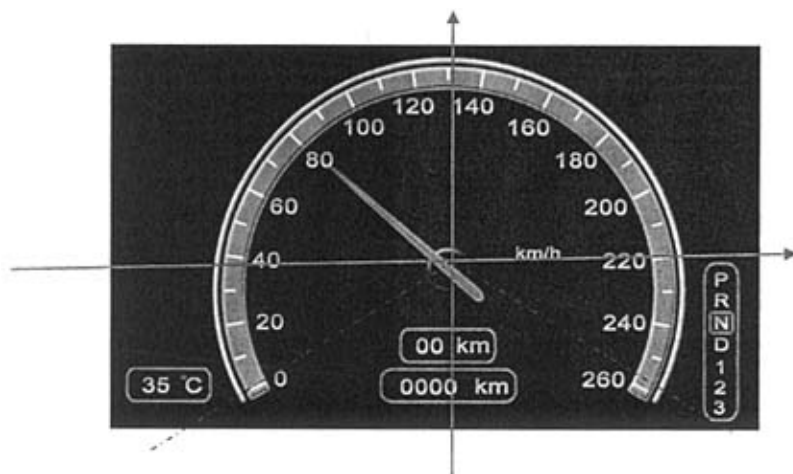


图 5.9 指针角度计算示意图

在粘贴背景和绘制指针时需要用到“设备环境 (Device Context, DC)”类、“图形对象 (GDI)”类等<sup>[25]</sup>。“设备环境”是 Windows 用来管理访问显示和打印设备的工具，是 Windows 应用程序、设备驱动程序和输出设备（如显示器、打印机、绘图仪等）之间的桥梁。设备环境类定义了逻辑显示画面，提供了绘图方法，而 GDI 对象则提供了 CDC 类绘图时的工具。图 5.10 显示了 Windows 应用程序经过设备环境到设备驱动程序并最终到输出设备的信息流。

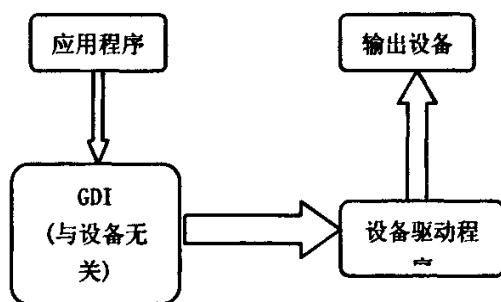


图 5.10 应用程序信息流图

指针绘制逻辑流程如图 5.11 所示。

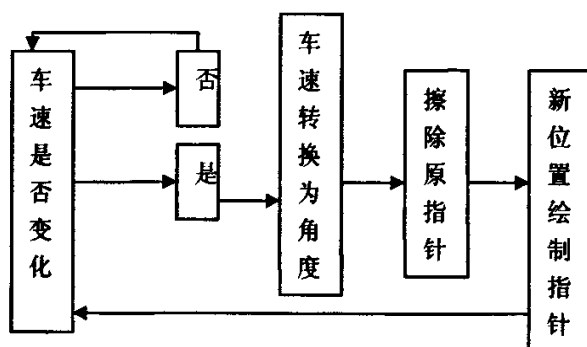


图 5.11 指针绘制流程图

### 5.3.3 里程计算存储与显示模块

在串口通讯数据协议中没有里程, 里程需要通过车速来计算, 因此可通过文件来存储累计的里程。在 WINCE 中文件的创建、读写等操作可使用 WINCE 提供的 API 函数或使用 MFC 类库中的 CFile 类。这两种操作的实质是相同的, 因为 CFile 类只是对原始文件操作的 API 函数进行了封装<sup>[27]</sup>。

建立一个 Mile.dat 文件用于存储里程, 每次开机时首先从文件中读出最近一次保存的里程, 其流程逻辑如图 5.12 所示。汽车行驶中使用定时器定时每一百毫秒累加一次里程, 计算方法为当前车速乘以时间间隔 (此处为 100 毫秒)。当里程累加增长超过 0.5 公里便更新显示并写入文件保存。

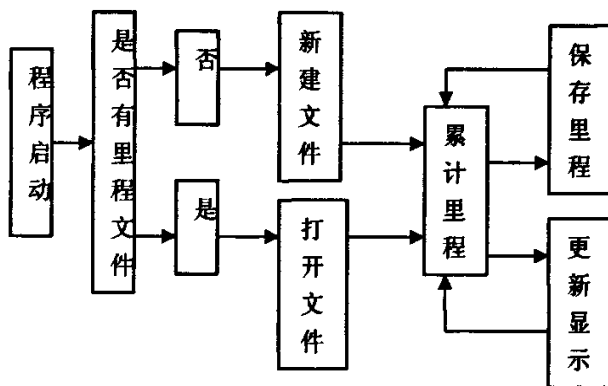


图 5.12 里程计算存储与显示逻辑图

### 5.3.4 功率流图模块

功率流图实时的显示车辆动力系统中燃料电池、蓄电池、轮毂电机和 DC/DC 转换器的动力分布情况。如图 5.13 所示。界面采用车辆动力系统的抽象图为背景，直观的显示了动力系统各个主要部分大致的分布情况，以及各个部分之间的相互联系。如，左右轮毂电机与蓄电池相连，由其直接提供动力，而蓄电池则通过 DC/DC 转换器由燃料电池充电，以得到动力。功力流动画是在在定时器<sup>[28]</sup>的驱动下不断刷新界面来实现的。

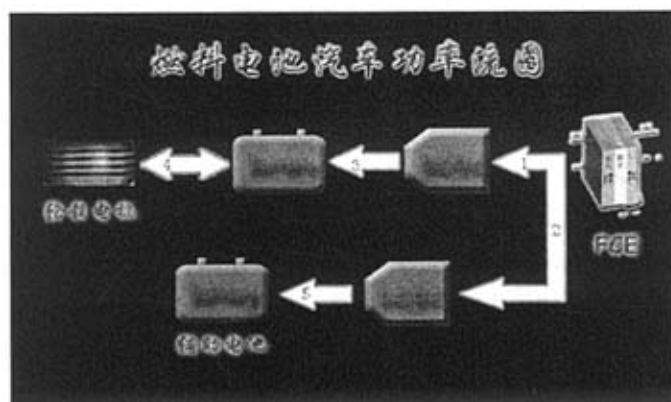


图 5.13 功率流图

如图所示，箭头指示动态功率流的方向。流向表示如下：

- (1) 燃料电池 (FCE) 功率 $>0$  时，1 向左，当 DC/DC1 功率 $>0$  时，2 向下，向左；5 向左。
- (2) 当 DC/DC1 功率 $>0$  时，3 向左。
- (3) 当 Battery1 功率 $>0$  时，电机功率 $>0$  时，4 向左。
- (4) 当电机功率 $<0$  时，4 向右。

### 5.3.5 动力系统参数显示模块

动力系统参数显示由第三个对话框来实现，和功率流图对话框一样，开机时默认是隐藏的，只有接收到界面切换命令才显示。动力系统参数分两屏显示，第一屏显示燃料电池的功率、进水温度，蓄电池的功率、温度、SOC 值、单电池最小电压、总电压和剩余电量；第二屏显示供氢系统的氢瓶温度、平均消耗率，

电机的水温、温度，DC/DC 的功率、温度，辅助电池的电压。

在定时器的驱动下，动力系统参数以五秒的时间间隔轮流显示，其流程逻辑如图 5.14 所示。

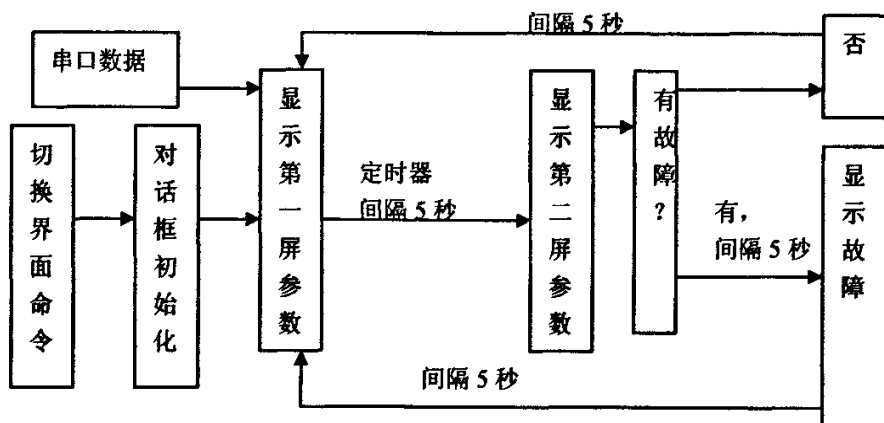


图 5.14 动力系统参数显示流程图

### 5.3.6 故障代码存储和显示

该模块主要用于普通的故障记录与故障简单描述，供用户查看，便于故障的快速排查与检修。从串口收到故障信息后，首先将其存储在数据类型为 long 的数组中，然后按照故障数据帧将其转换为标准的故障代码存储在 CString 类型的数组中。

收到的故障代码分三种情况：

- (1) 新出现的故障，故障代码数组中无记录，则将其加入故障代码数组中。
- (2) 故障代码在故障数组中已有记录，则不予处理。
- (3) 该故障已被排除，则将其从故障代码数组中删除。

按照故障代码字符数组从故障代码数据库中查询，得到故障的简单描述，然后紧接动力系统参数分屏显示出来。故障处理流程如图 5.15 所示。

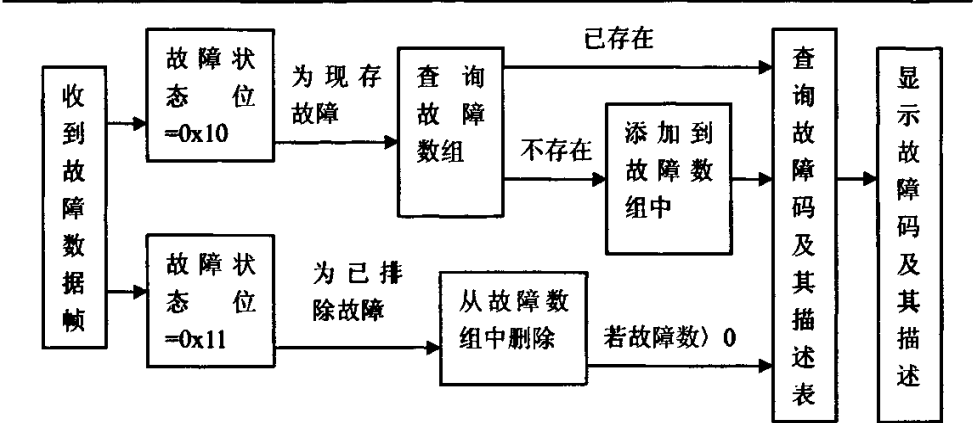


图 5.15 故障处理流程图

5.3.7 警告信号指示模块

该模块的功能是当出现蓄电池 LOC 低、风窗洗涤液面低、刹车片磨损、刹车油面低警告信号和雾灯开启时，在仪表上显示出相关警示信息。该模块的内部程序处理流程如图 5.16 所示。程序内部实现主要是通过逻辑判断而进行贴图、插图操作。

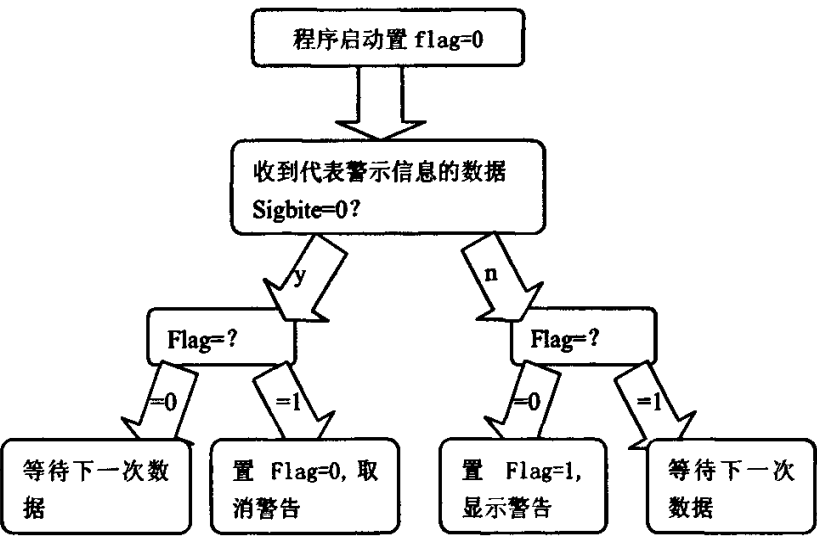


图 5.16 警告指示信息流

## 5.4 本章小结

软件是本课题的核心，本章针对 ROVER—FCV 仪表开发的仪表界面软件也是本课题的重点部分。主要实现了串口协议的制定和串口数据接收，虚拟指针显示车速，动画显示动力系统功率流图，数字图形显示动力系统参数及警示信号等。

## 第6章 汽车状态参数远程监控系统方案设计

汽车远程监控系统可以将汽车行驶中的状态参数上传到远程服务器，用于燃料电池汽车实验阶段，对汽车进行远程状态监控，及时发现潜在的问题、保障行车安全以及为新车的改进和技术成熟提供实时数据依据。

### 6.1 系统设计

#### 6.1.1 系统总体规划

远程监控系统一方面对汽车运行过程的状态数据进行记录、存储和管理，为及时发现潜在的问题和对汽车进行改进积累重要的实际数据；另一方面进行动力系统运行过程中故障发生情况的监测、记录、存储和管理，为系统故障诊断、故障排除，以及后续提高系统的可靠性研究建立基础。该系统主要由车辆—数据中心通信系统，数据中心数据管理、WEB 信息查询与显示等几个功能模块组成，规划系统功能模块如下。

##### 1. 车辆—服务器通信服务器模块

该模块负责车辆与数据中心之间的通信将汽车的状态信息和故障信息通过无线网络上传到数据中心。

##### 2. 数据中心数据管理模块

该模块监听来自远程车载 CE 客户端的连接，记录、存储和管理由客户端上传的汽车状态信息和故障信息；对数据库进行维护、管理等。

##### 3. 信息查询

通过该模块查询汽车的基本信息、状态和故障信息。

#### 6.1.2 系统架构



整个系统由车载客户端、无线网络通讯和服务端组成，其架构如图 6.1 所示。

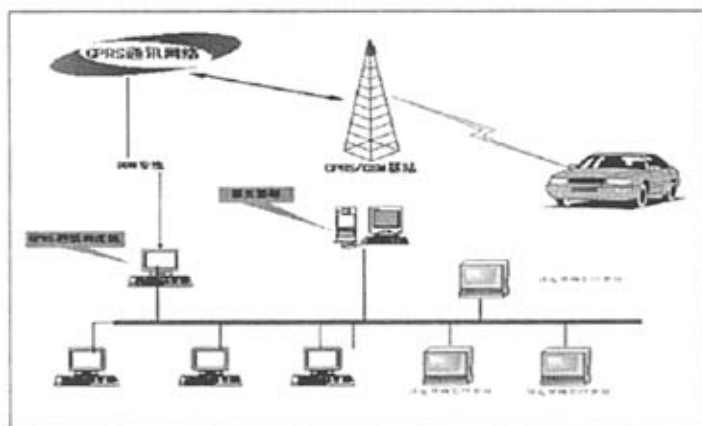
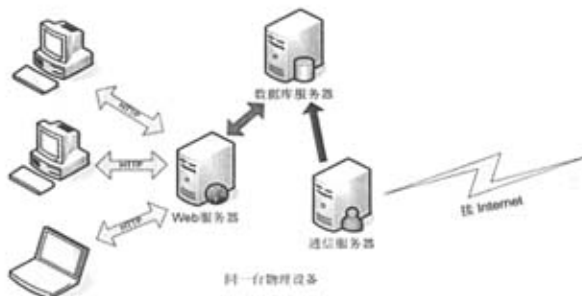


图 6.1 远程信息服务系统架构

### 6. 1. 3 服务器方案

中心服务器包括数据通信服务器、web 服务器和数据库服务器，数据通信服务器、web 服务器和数据库服务器运行在同一个物理设备上，数据通信服务器负责接收 DTU 发送过来的数据，处理后写入数据中心上的数据库；可以通过 web 服务器远程访问数据中心上的数据库；数据库服务器为 SQL Server2000，用来管理建立的数据库，如图 6.2 所示。



### 图 6.2 服务器方案

#### 6.1.4 系统设计目标

本系统属于客户机/服务器（C/S）和浏览器/服务器（B/S）[29]的混合结构，是针对汽车用户、汽车远程监控开发设计的。通过本系统可达到以下目标。

(1) 数据传递与存储安全可靠

(2) 界面设计美观大方，人机对话模式友好，WEB 服务器提灵活、方便、快捷、准确的供信息查询与浏览。

(3) 系统运行稳定、安全可靠。

(4) 系统功能易于扩展。

#### 6.1.5 开发及运行环境

(1) WEB 服务器系统开发平台：Eclipse+TomcatVisual C++6.0

(1) 通信服务器系统开发平台： Visual C++6.0

(2) 数据库管理系统软件：SQL Server2000。

(3) 服务器运行平台：Windows XP/ Windows 2000。

(4) 客户端信息浏览：WEB 浏览器，如微软的 IE。

(5) 车载端开发平台：Embedded Visual C++4.0。

(6) 车载端运行平台：Windows CE.net。

#### 6.1.6 车载信息网间通信协议

通讯协议由三个部分组成。

##### 1. 起始段

头协议=起始标志符（3）+目的地址+源地址+数据长度。

其中，目的地址——标志发送去的目的（预留6个字节）。

源地址——即，车辆编号（VIN 码后6位）。

L——从目的标志到中间协议结束所有的字节数。

##### 2. 数据段

数据段构成：标志位+时间（BCD 码表示日、时、分、秒，共4字节）+

车辆数据 0+车辆数据 1+车辆数据 2+车辆数据 3+……+车辆数据 n（车速+电量+水量+里程+电池功率+DC/DC+电机功率+FCE 功率+电池 SOC+电

池电压+氢瓶压力+氮瓶压力+汽车开关量标志+电池温度+DC/DC 温度+电机温度+逆变器温度+发动机水温+ 电压+电流)。

### 3. 结束段

一帧中头协议和中间协议所有字节的异或 (xor) 校验所得到的结果作为一个字节。协议详情请参看附录路 C。

## 6. 1. 7 数据库设计

本系统数据库采用 SQL Server2000 数据库[30], 数据库名为 AUTODATA, 主要包括车辆基本信息表、车辆状态和故障信息表等如图 6.2 所示。

车辆基本信息表

VEHICLE		
列名	简洁数据类型	
VIN	varchar (20)	
CID	varchar (50)	
[TIME]	datetime	
OTHER	nvarchar (100)	

车辆状态与故障记录

V DATA		
列名	简洁数据类型	
VIN	varchar (20)	
[TIME]	datetime	
IP	varchar (15)	
SPEED	smallint	
DISTANCE	float (53)	
BETPOW	smallint	
DCPOW	smallint	
ENGPOW	smallint	
FCPOW	smallint	
MOTORPOW	smallint	
LHCPOW	smallint	
RHCPOW	smallint	
SGPOW	smallint	
BETTEMP	smallint	
DCTEMP	smallint	
INVERTERT	smallint	
MOTORTEMP	smallint	
WATERTEMP	smallint	
M2	smallint	
M2	smallint	
MOTORSPEE	smallint	

车辆故障检索表, 用于具体故障信息的检索

DTC		
列名	简洁数据类型	
ERRID	varchar (50)	
Explain	nvarchar (100)	
MapID	nvarchar (100)	
Sug	nvarchar (500)	
Sort	varchar (10)	
[Level]	int	

图 6.2 主要数据表设计

车辆基本信息表用来存储注册车辆的基本信息, 包括车辆 vin 码, 客户编号, 注册时间和备注。车辆状态与故障记录表用于记录车辆的状态数据和发生故障的故障码, 以车辆 vin 码作为外键与车辆基本信息表相关联。故障检索表用来存储详细的故障信息, 包括故障码、故障解释、故障类别、故障等级、故

障对应部位图纸号和建议操作。

## 6.2 GPRS 无线通讯系统方案

车辆—数据中心间通信使用移动运营商的 GPRS 无线网络进行通信，其网络覆盖范围广，数据传输率根据不同情况为 19.2kbps~115kbps。

### 6.2.1 无线数传模块 ZWG-22A 简介

ZWG-22A 是由广州致远电子有限公司开发的工业级外置式 GPRS 无线数传模块 (DTU)，它是基于移动运营商现有的 GPRS 网络，可以方便的实现远程、无线、网络化的通信方式。具有覆盖范围广、组网方便快捷等优点。该产品的主要特性有：透明数据传输与协议转换，支持多数据中心，支持数据中心动态域名或 IP 地址访问，数据终端永远在线等多种工作方式可选，RS232 DB9 接口，单 5V~35V 宽范围供电，工作电流最大 300mA、休眠时 $\leq 10\text{mA}$ ，支持本地和远程图形界面配置与维护，支持短信配置与维护。典型应用模式如图 6.3 所示。

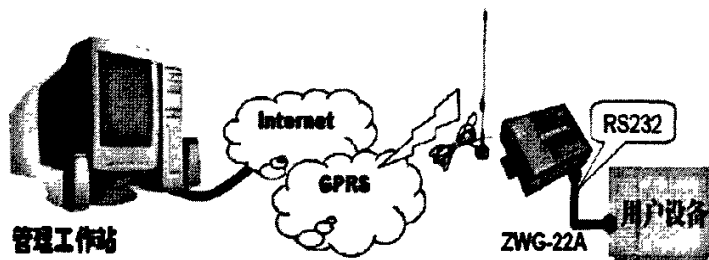


图 6.3 ZWG-22A 模块的典型应用模式

透明数据通信指的是用户设备之间没有通信协议，DTU 将用户设备发送过来的数据不作修改的传送到目标 PC 上，运行于 PC 上的软件可以完整的接收到 DTU 发来的数据包。

### 6.2.2 系统网络架构

实验目的是开发汽车远程通信服务和远程监控系统。在这次通信实验中，使用的是数据中心动态域名解析。此方案中，数据中心通过 ADSL 连接上

Internet。但是每次连上 Internet 后数据中心获得的公网 IP 地址都不一样，所以在 DTU 中无法直接设置目标 IP 来连接数据中心。这时可以使用动态域名解析系统（DDNS）来解决这个问题。DDNS 的原理是在数据中心上运行一个 DDNS 的客户端软件（如花生壳），它会定期的把该 PC 的公网 IP 号发送给 DDNS 服务器。在 DDNS 服务器上有一张经常被刷新的表单，该表单把数据中心的域名和它当前的动态 IP 对应上。所以 DTU 在使用域名访问数据中心时，首先要到 DDNS 服务器上查询该域名对应的 IP 是什么，DDNS 大致的工作流程如图 6.4 所示。

DTU 与数据中心服务器通信时需配置目标服务器参数，包括 IP 地址和端口号等，如果服务器没有固定的 IP 地址，可以使用域名。目前方案中采用免费域名方式，免费域名可以通过花生壳网站申请，如 sfcv.eicp.net，然后在数据中心服务器上运行 DDNS 的客户端软件（如花生壳，如图 6.5 所示），以动态更新域名 sfcv.eicp.net 对应的 IP 地址，DTU 则可通过此域名动态的访问数据中心服务器。

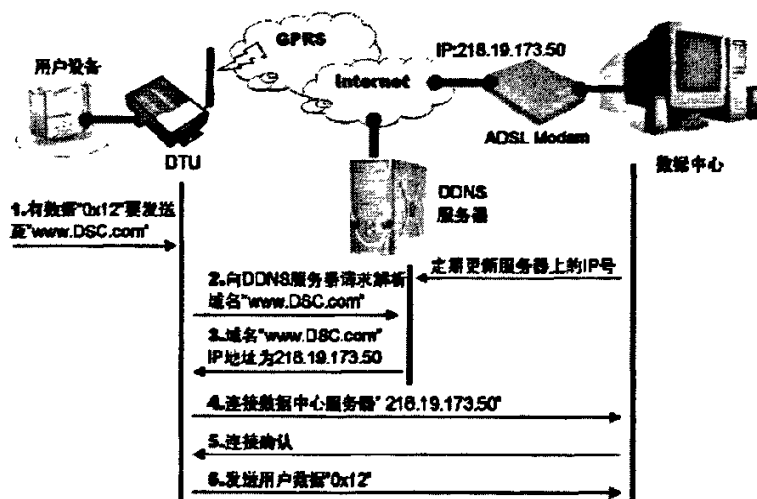


图 6.4 DDNS 大致的工作流程图



图 6.5 花生壳客户端程序运行界面

目前是在公司局域网上 IP 地址为 192.168.0.20（私有 IP 地址）的 PC 上架设数据服务中心，而整个局域网通过一台路由器网关上网（路由器的公网 IP 地址为 166.111.8.238，内网 IP 地址为 192.168.0.1），如果内网 IP 要与公网 IP 进行通信，则需将公网 IP 地址 166.111.8.238（路由器 WAN 口地址）和内网 IP 地址 192.168.0.20 之间建立一个端口映射（如端口 51234），这样访问公网 IP 地址为 166.111.8.238，端口为 51234 的数据就会被转发到私有 IP 地址 192.168.0.20 的数据中心服务器上，整个网络的拓扑结构如图 6.6 所示。

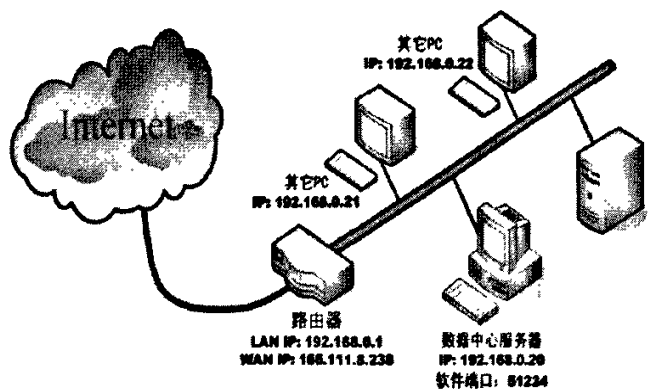


图 6.6 网络拓扑结构

6.2.3 通信实验

确定好网络架构和设置好服务器之后，将无线数传模块(DTU)，通过串口连接到一台 pc 上，放入 SIM 卡，打开本地配置软件（如图 6.7 所示）。

进入配置状态后，便可以进行参数设置，包括基本配置、高级配置、和附加配。基本配置是保证 DTU 正常工作的起码配置，包含串口参数设置、数据中心域名、IP 地址和端口号；合理的高级配置可节约数据流量、保证 DTU 与数据中心的可靠连接。数据中心主站域名一栏里填上 SFCV.EICP.NET,端口号为 9001,便可提交配置了。

本地配置成功后，将自动连接到服务器。连接成功后，服务器显示连接上的设备信息,如图 6.8 所示。点击获取信息便可获得无线数传模块的当前配置。还可以使用它对无线数传模块的参数进行远程修改和配置。



图 6.7 无线数传模块本地配置界面

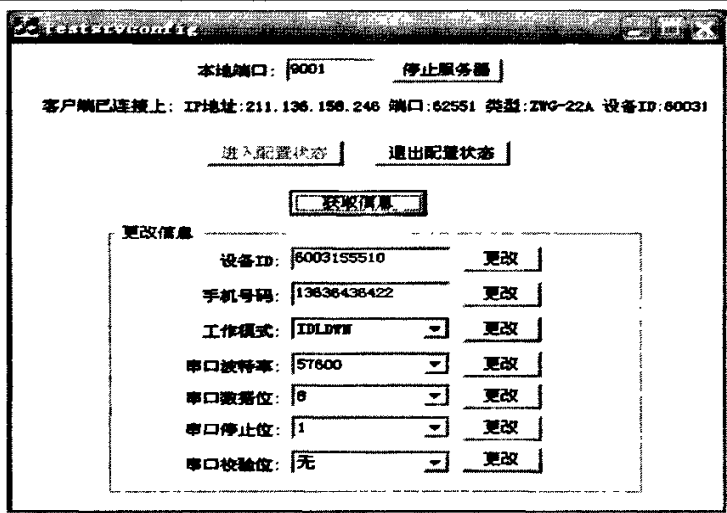


图 6.8 无线数传模块远程配置界面

接下来便是数据发送实验,在连接 DTU 的 pc 上使用串口调试工具发送数据。

要打开的串口号为无线数传模块所使用的串口,这里为 com1。串口参数要与配置好的无线数传模块的参数相一致。因为设置的工作模式是空闲下线方式,空闲时间可以设定,范围为 30—65536 秒,步进一秒。空闲下线后,当设备检测到有数据要发送,DTU 将先与数据中心重新建立连接,连接成功后将数据发出。

6.2.4 无线数传模块下的应用软件开发

产品供应商提供了函数库和动态连接库,包括串口控制函数、远程控制函数、通用配置函数等,使用这些函数在 VC、VB、CB 和 Delphi 编程环境下调用这些函数进行客户端和服务端的应用软件开发。在本系统同中使用 VC 开发平台。

6.3 通信服务器设计

通信服务器负责接受车载 CE 端发送过来的数据,将数据存入数据库,同时管理员可通过图形化界面维护和管理数据库。

6.3.1 服务器程序框架



服务器程序采用对话框结构，其界面如图 6.10 所示。

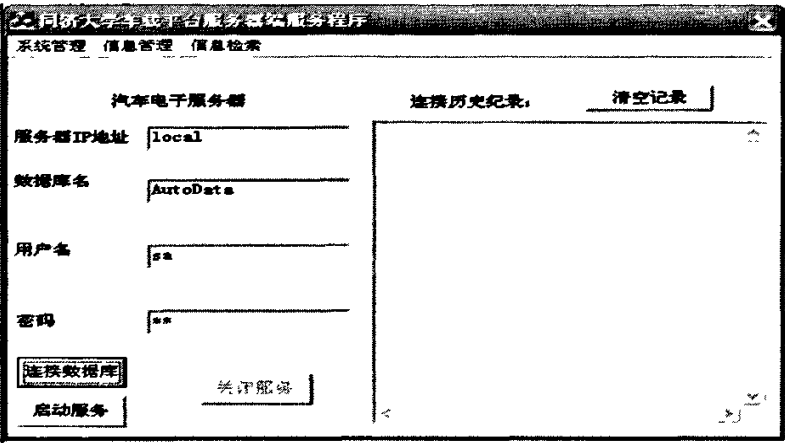


图 6.10 服务器程序运行界面

该程序一方面负责接受车载 CE 端发送过来的数据，将数据存入数据库，同时和远程服务模块进行通讯，传送实时的数据；另一方面管理员可通过菜单维护与管理本地数据库，如上图所示，正确填写好数据库所在 IP 地址，以及数据库名、用户名、密码后，成功连接数据库后，启动服务即可。通过菜单可进行本地数据库管理。其流程逻辑如图 6.11 所示。

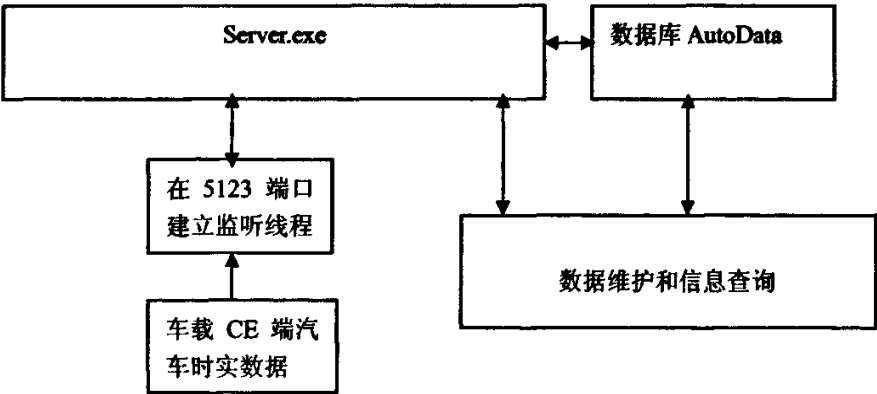


图 6.11 服务器程序逻辑图

整个服务器程序采用分层的模块结构设计，模块划分如图 6.12。本系统

逻辑方面的操作较少，主要是界面和数据库的操作，该架构也体现了这一点。

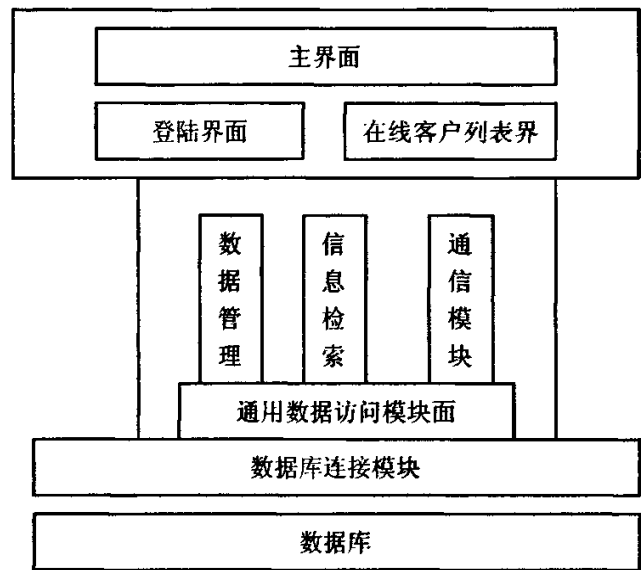


图 6.12 模块划分图

6.3.2 服务器程序初始化

在设计程序时，考虑到程序在不同的应用环境下需要不同的参数配置，为了提高程序的灵活性，服务器程序在启动时，一些参数是通过读取配置文件来设置的。当前用到的 PID 参数表(表 6.1)如下。

表 6.1 PID 参数表

编号	PID	name	K	Z	max	min	precision	unit	sort	测试值
[0]	100	车速	*	0	220	0	0. 1	KM/H	P	50
[1]	101	里程		0	—	0	0. 1	KM	C	100
[2]	102	电池功率		0	100	0	0. 1	KW		50
[3]	103	电池 SOC		0	100	0	0. 1	%		80
[4]	104	氢瓶压力		0	255	0	0. 1	KPa		200
[5]	105	DC/DC 功率		0	100	0	0. 1	KW		50
[6]	106	电机功率		0	100	0	0. 1	KW		50
[7]	107	FCE 功率		0	100	0	0. 1	KW		80
[8]	108	电池电压		0	120	0	0. 1	V		100

第 6 章 汽车状态参数远程监控系统方案设计

[9]	109	氮瓶压力		0	255	0	0. 1	KPa		200
[10]	110	电池温度		0	100	0	0. 1	℃		80
[11]	111	DC/DC 温度		0	200	0	0. 1	℃		80
[12]	112	电机温度		0	200	0	0. 1	℃		50
[13]	113	逆变器温度		0	200	0	0. 1	℃		50
[14]	114	发动机水温		0	100	0	0. 1	℃		80
[15]	115	发动机转速	**	0	6000	0	1	Rps		2000
[16]	116	电机扭矩	*	0	100	0	0. 1	NM		50
[17]	117	电机转速	**	0	2000	0	1	Rps		1000

对应的 PID 参数配置文件，格式如下：

```
[0]                                /* 标识号
PID=                              /* 车辆参数 PID 代号
name =                            /* 车辆 PID 具体名称
K =                               /* 数据格式转换方法
Z=                                /* 数据偏移量
max =                             /* 该参数最大值
min =                             /* 该参数最小值
precision =                       /* 该参数精度
unit =                            /* 单位
sort =                            /* 分类

[1]
PID=
Name=
...
```

在程序中定义一个结构体数组，如下所示：

```
Struct
{
int PID;
char name[50];
```

```
}

```

程序启动时，读取配置文件信息将其存储在结构体数组中，以供程序使用。

### 6.3.3 封装数据库

#### 1. ADO 对象基本模型

服务器程序采用 ADO 技术操作数据库，ADO (Active Data Object) 是 Microsoft 的 Universal Data Access 发展策略的一个重要组成部分，它将取代 DAO (Data Access Object) 和 RDO (Remote Data Object) 两种技术实现对不同类型数据源的访问<sup>[30]</sup>。

ADO 模型包含了连接对象、命令对象、域对象、参数对象、记录集对象、错误对象等。对象之间又具有层次关系，其关系如图 6.13 所示。

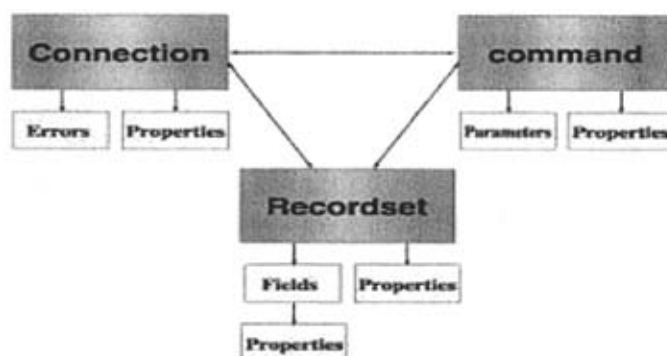


图 6.13 ADO 模型图

在使用 ADO 技术时，需要导入一个 ADO 动态链接库 msado15.dll，方法是在封装 ADO 数据库操作类的头文件中添加“#import “lib\\msado25.tlb” no\_namespace rename(“EOF”, “adoEOF”)”语句。程序中定义三个类来封装对数据库的操作，它们是 CADOConnection, CADODataset 和 CADOStorage。

#### 2. 连接数据库

使用数据库前，先要创建 Connection 数据源连接对象。在使用其他对象之前必须先建立数据源连接，然后在连接对象上创建和使用其他对象<sup>[31]</sup>。创建 Connection 对象的部分代码如下：

---

```

CADOConnection::CADOConnection()
{
    ::CoInitialize(NULL); //初始化 com 对象
    hr=this->m_pConnection.CreateInstance(__uuidof(Connection));
    //创建 Connection 连接对象
}

```

Connection 对象代表者与数据源进行的唯一会话,使用 Connection 对象的 Open 方法可建立到指定数据源的物理连接,管理着应用程序和数据库之间的通讯。对于大多数的数据库系统,每一个物理连接都需要耗费大量系统内存,是宝贵的系统资源,不可滥用。考虑到应用程序的效率,每一个数据库客户端都只用一个物理连接,对数据库操作结束后应该调用 Connection 对象的 Close 方法释放。该程序中使用 CADOConnection 对 Connection 对象进行了封装,包括连接、关闭连接、判断、获取连接等方法。

## 2. 数据库操作

成功打开数据库后,就可以建立 Recordset 对象,本程序中封装 Recordset 对象的类为 CADODataset,定义了基本的数据库添加、查询、删除、排序等函数<sup>[32]</sup>。下面为 CADODataset 类部分成员函数。

```

bool SetFieldValue(int nIndex, CString strValue);
bool SetFieldValue(LPCTSTR lpFieldName, CString strValue);
bool SetFieldValue(int nIndex, int nValue);
bool SetFieldValue(LPCTSTR lpFieldName, int nValue);
bool SetFieldValue(int nIndex, long lValue);
bool SetFieldValue(LPCTSTR lpFieldName, long lValue);
bool SetFieldValue(int nIndex, double dblValue);

```

以上为获得、设置字段值成员函数,根据不同的数据类型,调用不同的成员函数,

```

//编辑功能函数
void CancelUpdate();
bool Update();
void Edit();

```

---

```

        bool AddNew();
        //关键字查找函数, 无返回值
        Bool Find(LPCTSTR lpFind, int nSearchDirection=
CADODataset::searchForward);
        //执行 SQL 语句, 并返回 Recordset 对象
        _RecordsetPtr OpenSql(LPCTSTR TblName, LPCTSTR
FldList="*", LPCTSTR Constr="", LPCSTR OrderBy="");。

```

上述就是 ADO 访问数据库的基本过程, 并描述了封装 ADO 访问数据库类的方法。使用好 Recordset 记录集对象时, 也需调用 Close 方法关闭, 但要注意不要多次关闭, 否则会出错。

### 6.3.4 Windows Sockets 下的 TCP/IP 通信模块

Windows Sockets 规范定义了一套 Microsoft Windows 下的网络编程接口, Windows Sockets 规范并记录了如何使用 API 与 Internet 协议族 (IPS, 通常指的是 TCP/IP) 连接, 所有的 Windows Sockets 实现都支持流套接字和数据报套接字接口<sup>[33]</sup>。为了保证数据传输的可靠性, 在该系统中使用流套接字, 即 TCP/IP 协议来进行通信。使用流套接字通信的过程如图 6.14 所示<sup>[34]</sup>。

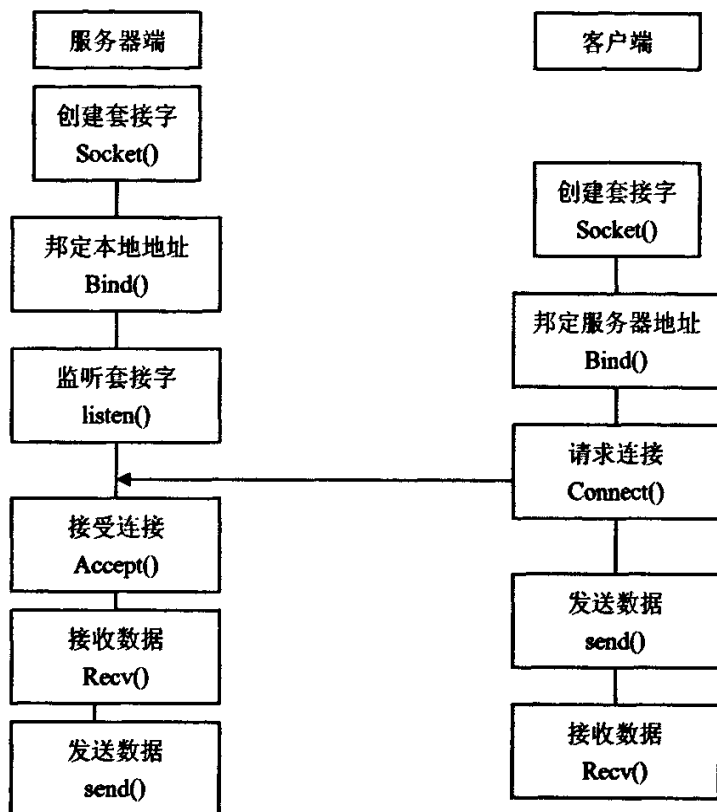


图 6.14 Winsock 流套接字编程流程图

WinSock 以 DLL 的形式提供，在调用任何 WinSock API 之前，必须调用函数 `WSAStartup` 进行初始化，最后，调用函数 `WSACleanup` 作清理工作。Socket 是网络通信过程中端点的抽象表示。Socket 在实现中以句柄的形式被创建，包含了进行网络通信必须的五种信息：连接使用的协议，本地主机的 IP 地址，本地进程的协议端口，远地主机的 IP 地址，远地进程的协议端口<sup>[35]</sup>。

为了创建 socket，使用 `socket` 函数得到一个 socket 句柄。当一个 socket 被创建时，WinSock 将为一个内部结构分配内存，在此结构中保存此 socket 的信息，到此，socket 连接使用的协议已经确定。

创建了 socket 之后，需要配置 socket。对于面向连接的客户，WinSock 自动保存本地 IP 地址和选择协议端口，但是必须使用 `connect` 函数配置远地 IP 地址和远地协议端口。

面向连接的服务器则使用 bind 指定本地信息, 使用 listen 和 accept 获取远地信息。使用数据报的客户或者服务器使用 bind 给 socket 指定本地信息, 在发送或者接收数据时指定远地信息。Bind 函数给 socket 指定一个本地 IP 地址和协议端口<sup>【36】</sup>。

在 socket 配置好之后, 使用 socket 发送或者接收数据。面向连接的 socket 使用 send 发送数据, recv 接收数据; 使用数据报的 socket 使用 sendto 发送数据, recvfrom 接收数据。

按照上述方法在程序中专门建立一个 AcceptThread 线程在 5123 端口等待 CE 客户端连接, 连接后启动另一个线程 SaveDataThread, 专门用来服务该连接, 接收从 CE 客户端上传数据分开存放到 V\_DATA, V\_ERROR, V\_STATE 3 个表中, AcceptThread 线程继续等待连接, 详细源代码请参考源程序。

程序中定义了一个数组结构, 将接收到的数据写入该结构中, 然后调用数据库操作对象将数据插入到相应的表中。

### 6.3.5 DTU 无线数传模块下的通信。

使用 DTU 无线数传模块时的通信过程与上述原理一致。他们提供的通信函数实际上是对 Windows Sockets 的 API 的封装, 简化了编程过程。

## 6.4 WEB 服务器方案设计

### 6.4.1 设计目标

对汽车的状态和故障数据进行远程监控。根据通信服务器收集到的数据定时更新显示的信息, 供用户浏览, 还可查询历史信息, 利用这些信息可对汽车的状态和故障进行分析。

### 6.4.2 总体设计

WEB 服务器总体结构图如图 6.15 所示。



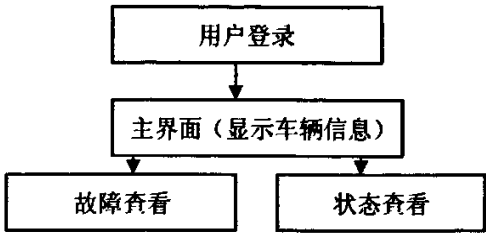


图 6.15 web 服务器结构图

1. 用户登陆界面

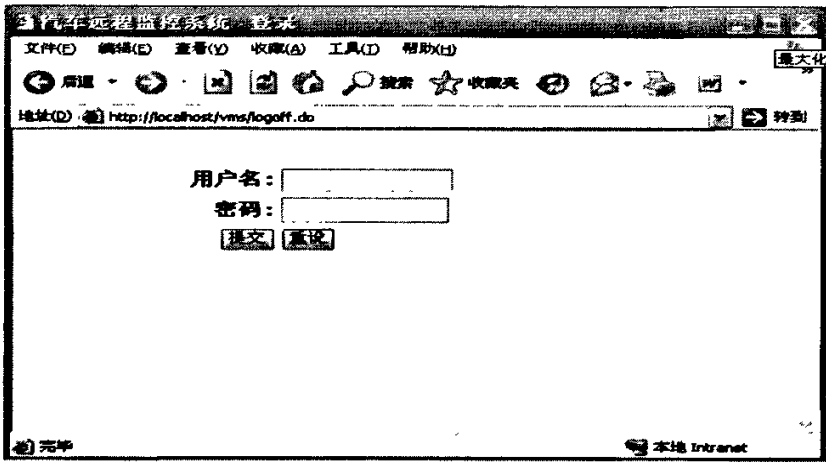


图 6.15 用户登录界面

2. 车辆列表主界面

用户通过身份验证登录后，显示车辆列表界面，如图 6.16 所示。



图 6.16 车辆列表界面

该界面中，上面灰色框中为主菜单，左侧显示当前选中的主菜单的子菜单项，用户可以通过这些菜单进入车辆状态和车辆故障查看页面。

6.5 本章小结

本章讲述了燃料电池汽车远程监控系统的网络架构，设计目标和初步方案。从服务器端来看，包括通信服务器，数据管理服务器和 WEB 服务器，它们位于同一台无力设备商。该系统的一个重要特征是，服务器访问采用免费的动态域名解析，既节省成本，又使得服务器的维护更灵活方便。此外，本文在 Visual C++ 环境下开发了该系统的通信和数据管理服务器程序；对 WEB 服务器进行了总体方案设计。

## 第7章 总结与展望

### 7.1 总结

本人在研究汽车仪表功能的基础上，对嵌入式系统编程及其应用作了认真研究，在系统的开发过程中，积累了一定的经验，完成了研究任务并取得了成果，可概括如下。

(1)用嵌入式液晶显示智能仪表大大拓宽了仪表的显示功能，显示的信息量大，信息可视性、可读性强，能实现一表多用。

(2)引入了操作系统的概念，增强了代码的可读性、可维护性、可扩展性以及灵活性；信息显示自由度高，显示界面人性化，可定制；即使更换硬件平台，也只需对操作系统和底层驱动程序进行少量的移植工作，而无需修改与硬件无关的应用代码。

(3)汽车状态参数远程监控系统的主要工作已经完成，服务器使用动态域名来访问，完成了服务器端数据库开发，通信和数据库管理程序也已初步完成。

### 7.2 进一步工作的方向

本文的研究虽然取得了初步的成功，但依然任重道远，尚有许多有待进一步深入进行的研究工作，这里择其要者简要讨论如下：

对汽车信息显示软件进行升级，实现灵活的，对用户可定制的车载信息显示系统。

远程通信服务只是在有线连接的情况下进行了初步测试，更好更多的功能和服务设计需要在实践中逐步更新和完善。

修改远程通信服务程序中的通信代码，以便能够在 DTU 下实现无线网络通信。

利用嵌入式硬、软件的可移植性与扩展性，逐步实现汽车“黑匣子”、GPS、自动导航、后视摄像头、收音机、影音娱乐模块等在车载信息平台上的显示。

## 致谢

本文是在导师沈勇教授的精心指导下完成的，他严谨的治学态度、渊博的学识、丰富的项目经验、诲人不倦的精神以及对于理想的不懈追求是我学习的榜样。攻读硕士学位期间，在学习和生活中都得到了沈老师的悉心教诲，使我在项目的设计和开发过程中不仅顺利的解决了实际的技术难题，而且学到了很多新的专业知识和实践经验，获益匪浅，谨此向沈老师表示最真挚的感谢。

感谢“四轮独立驱动微型电动车”项目组的以及课题组的所有同学，对论文的撰写和试验给予了大力协助。感谢上海燃料电池汽车动力系统有限公司的陆勤波、李恒宇、毛振涛和魏臻在项目研究、论文的撰写和试验过程中给予的热心协助和大力支持。

感谢同济大学汽车学院领导和其他老师的帮助和热情鼓励，使我顺利完成学业以及论文的撰写。

感谢同窗硕士研究生宋吉伟、吕成浩、顾剑杰和汤若斌在论文写作期间给予了本人无私的帮助。

另外论文中引用了一些作者的论文及其研究成果，这对本论文的完成提供了很大的帮助，在此也表达我的谢意！

承蒙各位专家、教授的审阅和评议，你们提出的宝贵意见将使我受益匪浅。所以向为审阅本论文付出辛勤劳动的各位专家教授表示最衷心的感谢！

最后，感谢我的家人这么多年来对我的大力支持！

2007年6月

## 参考文献

- [1] 关惠平, 谢飞鸿, 刘有录. 车载综合信息系统终端构筑方案的研究与实践. 计算机工程与设计, 2006, Vol. 28 (2): 278-282
- [2] Harvey J. Miller and Shih-Lung Shaw. Geographic Information System for Transportation-Principles and Application[Z]. Oxford University Press, Inc, 2001
- [3] 朱昱琼. 汽车电子信息系统设计. 计算机与现代化, 2006(3):55-57
- [4] 李伟. 车载信息显示平台的研究: [硕士学位论文]. 武汉: 武汉理工大学控制理论与控制工程专业, 2006
- [5] 王涛, 黄海波, 黄维真等. Windows CE 傻瓜书. 北京: 清华大学出版社, 2000
- [6] 肖永清, 杨忠敏. 汽车的发展与未来. 化学工业出版社, 2004
- [7] 穆斌. 智能化车载信息终端的设计与实现. 安徽工程科技学院学报, 2004(3):46-49
- [8] 周洁, 杨心怀. 32 位 RISC CPU ARM 芯片的应用和选型. 电子技术应用, 2002, Vol. 27
- [9] Xu, X. H., Clarke, C. T., Jones, S. R. High performance code compression for the Embedded ARM/THUMB processor. 2004 Computing Frontiers Conference, 2004 Computing
- [10] 林永仁. 嵌入式系统项目分析入门与实践. 北京: 中国铁道出版社, 2004
- [11] 吴明晖. 基于 ARM 的嵌入式系统开发与应用. 北京: 人民邮电出版社, 2004
- [12] 胥静. 嵌入式系统设计与开发实例详解. 北京: 北京航空航天大学出版社, 2005
- [13] (美) David E. Simon. 嵌入式系统软件教程. 北京: 机械工业出版社, 2005
- [14] (美) Qing Li. 嵌入式系统的实时概念. 北京: 北京航空航天大学出版社, 2004
- [15] 傅曦, 齐宇. 嵌入式系统 Windows CE 开发技巧与实例. 北京: 化学工业出版社工业装备与信息工程出版中心, 2004
- [16] 蒋静, 徐志伟. 操作系统, 原理·技术与编程. 北京: 机械工业出版社, 2004
- [17] 何宗键. Windows CE 嵌入式系统. 北京: 北京航空航天大学出版社, 2006
- [18] 叶宏材, 陈峙翥. Windows CE.NET 嵌入式工业用控制器及自动控制系统设计. 北京: 清华大学出版社, 2005: 2
- [19] 姜波. Windows CE.Net 程序设计. 机械工业出版社, 2007: 6
- [20] 陈向群. Windows CE.NET 系统分析及实验教程. 北京: 机械工业出版社, 2003
- [21] 汪兵, 李存斌, 陈鹏. EVC 高级编程及其应用开发. 北京: 中国水利水电出版社, 2005:17-18
- [22] 王世同, 李强. Visual C++6.0 编程基础. 北京: 清华大学出版社, 1999
- [23] 李长林, 高洁编. Visual C++串口通信技术与典型实例. 北京: 清华大学出版社, 2006
- [24] Microsoft Corporation. Microsoft Windows CE Programmers Guide. Microsoft Press, 1999

## 参考文献

- 
- [25] Microsoft Corporation. Microsoft Windows CE Communications Guide. Microsoft Press, 1999
  - [26] 齐舒创作室. Visual C++6.0 用户界面制作技术与应用实例. 北京: 中国水利水电出版社, 1999
  - [27] 刘小石. 精通 Visual C++6.0. 北京: 清华大学出版社, 2000
  - [28] 汪兵, 李存斌, 陈鹏. EVC 高级编程及其应用开发. 北京: 中国水利水电出版社, 2005
  - [29] 东风汽车公司技术中心, 陈延寿, 王建萍. 信息技术在汽车上的应用. 汽车工业研究, 2001(10): 26-28
  - [30] 夏邦贵, 刘凡馨等. Visual C++数据库开发经典实例精解. 北京: 机械工业出版社, 2006
  - [31] Microsoft Corporation. Microsoft SQL Server. Microsoft Press, 1998
  - [32] 高春艳、李艳. Visual basic 数据库开发关键技术与实例应用. 北京: 人民邮电出版, 2005
  - [33] 王端、于速、张雨. Visual C++ 数据库系统开发完全手册. 北京: 人民邮电出版社, 2006
  - [34] Microsoft Corporation. Microsoft Visual C++6.0 MFC Library Ref2. Microsoft Press, 1999
  - [35] 《电脑编程技巧与维护》杂志社. Visual C++编程技巧典型案例解析 网络与通信及计算机安全与维护篇. 北京: 中国电力出版社, 2005
  - [36] 王罡, 林立志. 基于 Windows 的 TCP/IP 编程. 北京: 清华大学出版社, 2002

附录 A ROVER 仪表串口通讯数据协议

一、 数据通讯协议：

- ✓ 串口通信数据帧定义：  
1 帧 = (5+30+1=36 字节) -/- (5+18+1=24 字节) -/- (5+6+1=12 字节)  
1 帧分为三个部分：
- ✓ 头协议： 5 字节  
头协议 = 起始标志符(0x33) + 源地址(0x0A) + 数据类型(0x00/0x01)  
+ 数据长度 + 保留位
- 备注： 数据类型： 0x10 为数据帧； 0x11 为故障代码帧； 0x00 为控制帧；  
数据长度——从目的标志到中间协议结束所有的字节数。

协议	名称	数据长度	数据位置	数据说明
起始段 共 5byte	起始标志	1	0	0x33
	源地址	1	1	0x0A
	数据类型	1	2	0x00/0x10/0x11
	数据长度	1	3	
	保留字节	1	4	

➤ 中间协议： 30 字节 （数据帧 0x10）

协议	名称	数据长度	数据位置	数据说明
中间段	标志位	1	5	0x10
	Data0			燃料电池功率
	Data1			燃料电池出水温度
	Data2			燃料电池进水温度
	Data3			保留位
	Data4			电池功率
	Data5			电池温度
	Data6			电池电压
	Data7			电池 soc
	Data8			单电池最小电压

## 附录

	Data9			蓄电池剩余电量
	Data10			辅助电池电压
	Data11			保留位
	Data12			供氢压力（燃料量）
	Data13			氢瓶温度
	Data14			续航里程
	Data15			平均消耗率
	Data16			保留位
	Data17			电机水温
	Data18			电机功率
	Data19			电机温度
	Data20			保留位
	Data21			PCU 水温
	Data22			DC/DC 功率
	Data23			档位
	Data24			车速
	Data25			保留位
	Data26			保留位
	Data27			保留位
	Data28			保留位

► 中间协议： 18 字节（故障代码帧 0x11）

协议	名称	数据长度	数据位置	数据说明
中间段	标志位	1	5	0x11
	故障状态位	1	6	0x10/0x11
	Data0			DTC_1_1
	Data1			DTC_1_2
	Data2			DTC_1_3
	Data3			保留位
	Data4			DTC_2_1
	Data5			DTC_2_2
	Data6			DTC_2_3
	Data7			保留位
	Data8			DTC_3_1
	Data9			DTC_3_2
	Data10			DTC_3_3



附录

	Data11			保留位
	Data12			DTC_4_1
	Data13			DTC_4_2
	Data14			DTC_4_3
	Data15			保留位

备注：

- 1. 故障状态位说明： 0x10 表示发送当前故障为现存故障；  
0x11 表示发送当前故障为取消的故障
- 2. 故障代码发送说明：故障代码分为三个字节，  
故障代码的字节 1，位 8~6，故障等级（二进制）；  
故障代码的字节 1，位 5~1，系统分类代码（二进制）；  
故障代码的字节 2，故障代码编号低字节；  
故障代码的字节 3，故障代码编号高字节；
- 3. 无故障数据填充： 0x00；

➤ 中间协议： 6 字节（控制帧 0x00）

协议	名称	数据长度	数据位置	数据说明
中间段	标志位	1	5	0x00
	界面切换标志位	1	6	0x01、0x02、0x03
	保留位		7	
	保留位		8	
	保留位		9	
	保留位		10	

➤ 尾协议： 1 字节

协议	名称	数据长度	数据位置	数据说明
结束段	结束位	1		异或数据

## 附录 B 数据库主要数据表设计

表 V\_DATA 与 DATA\_EX

说明：记录由 C E 端上传的数据，每车每 1 秒一次；

编号	字段名	Name	max	min	unit	类型	大小
[1]	VIN	车辆识别号	17 位字母数字的组合				
[2]	TIME	上传时间	年月日时分秒				
[3]	IP	CE 端 IP	标准 I P V 4				
[4]	SPEED	车速	220	0	KM/H		
[5]	DISTANCE	里程	—	0	KM		
[6]	BETPOW	电池功率	100	0	KW		
[7]	DCPOW	DC/DC 功率	100	0	KW		
[8]	ENGPOW	发动机输出功率	200	0	KW		
[9]	FCEPOW	FCE 功率	100	0	KW		
[10]	MOTORPOW	电机功率	100	0	KW		
[9]	LMCPOW	左轮毂电机功率	100	-40	KW		
[12]	RMCPow	右轮毂电机功率	100	-40	KW		
[13]	SGPOW	SSG/ISG 电机功率	100	-40	KW		
[14]	BETTEMP	电池温度	100	0	℃		
[15]	DCTEMP	DC/DC 温度	200	0	℃		
[16]	INVERTERTEMP	逆变器温度	200	0	℃		
[17]	MOTORTEMP	电机温度	200	0	℃		
[18]	WATERTEMP	发动机水温	100	0	℃		
[19]	H2	氢瓶压力	255	0	KPa		
[20]	N2	氮瓶压力	255	0	KPa		
[21]	MOTORSPEED	电机转速	2000	0	Rps		
[22]	ROTATESPEED	发动机转速	6000	0	Rps		
[23]	BMIbIs	蓄电池充/放电电流	-200	300	A		
[24]	BETVOLT	蓄电池端电压	400	0	V		
[25]	MOTORTORQU E	电机扭矩	100	0	NM		
[26]	OIL	油量	100	0	%		
[27]	SOC	蓄电池 SOC(电余量)	120	0	%		

## 附录

表 V\_STATE

说明：车辆状态记录，当车辆状态改变时记录一次。

字段名	类型	大小	说明
VIN			车辆识别号
SID			状态号
TIME			状态改变时间

表 V\_ERROE

说明：车辆故障记录，每车每一故障号，每秒只记一次

字段名	类型	大小	说明
VIN			车辆识别号
ERRID			故障代码
TIME			故障发生时间

表 VEHICLE

说明：车辆基本信息

字段名	类型	大小	说明
VIN			车辆识别号
CID			客户 I D
Other			备注
TIME			注册日期，即开始记录日期

表 DTC

说明：车辆故障检索表,用于具体故障号的检索

字段名	类型	大小	说明
ERRID			故障代码
Explain			故障解释
MapID			故障部位图纸号(同一部位可有多个故障)
Sug			维修建议
Sort			(作用尚不明,使用原来的设定)
Level			故障等级

表 STATE

说明：车辆状态检索表，用于车辆具体状态号的检索。

字段名	类型	大小	说明
SID			状态号
STATE			状态说明

## 附录

表 MAP

说明：用于存储车辆各部分的图纸信息。

(注：这部分的图片以后提供,现在有几个象征性的 MAPID 即可)

字段名	类型	大小	说明
MAPID			图纸号
PICTURE			图纸图片(使用位图格式 BMP)

表 feedback

说明：用于存储用户反馈信息表

字段名	类型	大小	说明
Key	Int	4	索引号
item			反馈项名称
score			用户反馈分数

表 info

说明：用于存储常用问题及其解答表

字段名	类型	大小	说明
key			索引号
question			用户提问问题
answer			问题答案

## 附录 C 车载信息网间通信协议

车辆状态定义							
0	1	2	3	4	5	6	7
整车状态				部件状态			
0001	启动			4=FCE	0=Off	1=On	
0010	加速			5=DC/DC	0=Off	1=On	
0011	低负荷			6=Battary	0=充电	1=放电	
0100	全负荷			7=Motor	0=Off	1=On	
0101	减速（制动或刹车）						
0110	滑行						
0111	驻车						

通讯协议	名称	信号编号	数据顺序位置	备注
起始段 共 15byte	起始标志	0	0	‘3’
	目的地址	0	1-6	预留 6 字节
	源地址	0	7-12	VIN 码后 6 位
	保留字节	0	13	
	传输长度	0	14	
数据段	标定文件	0	15-16	数据对照标定文件
	保留字节	0	17-18	
	时间	1	19-22	BCD 码表示日、时、分、秒
	Data0	2	23-24	以 2 字节表示，其中数据段 排列顺序为：状态、数据、 主要故障代码
	Data1	3	25-26	
	Data2	4	27-28	
	Data3	5	29-30	
	Data4	6	31-32	
	Data5	7	33-34	
	Data6	8	35-36	
	Data7	9	37-38	
结束段	结束位	0	128	

## 个人简历 在读期间发表的学术论文与研究成果

个人简历:

周逸峰, 男, 1980 年 12 月 26 日生。

2004 年 7 月毕业于同济大学机械制造及其自动化(汽车)专业 获学士学位。

2004 年 9 月入同济大学读硕士研究生。

已发表论文:

[1] 周逸峰.基于 ARM 的新能源汽车显示系统.农用装备与车辆工程, 2007, 5。