

深圳大学

优秀毕业论文(设计)选

SELECTED GRADUATION THESES (PROJECTS) OF SZU 2000
(SCIENCES)

2000 届
理科卷

深圳大学教务处编

目 录

理学院

- | | | |
|-----|--------------------------------------|-----|
| 1/ | 智能牌照系统之车载电脑的研究 | 周宇龙 |
| 21/ | 金相显微镜的数字识别系统 | 杨国雄 |
| 44/ | Internet 化学信息资源导航系统开发 | 马 良 |
| 50/ | 几种大米新陈度测定方法的对照研究 | 冼 坚 |
| 57/ | 垂笑君子兰根尖组织培养育苗的研究 | 林奇龄 |
| 60/ | β —环糊精—吐温—双硫脲法测定
食品中的微量锌的研究 | 石胜光 |
| 73/ | 淋洗剂正相液相色谱法的研究 | 林 凯 |

建筑与土木工程学院

- | | | |
|------|-------------------|-----|
| 86/ | 网上房屋测量初探 | 曹 嘉 |
| 120/ | 基于互联网的地价图绘制研究 | 竺琳琳 |
| 161/ | 项目控制的思想、理论、方法与实务 | 潘海飞 |
| 184/ | 深圳大学科技楼设计 | 黄晓江 |
| 191/ | 深圳彩田村住宅小区中小学设计 | 何仲健 |
| 203/ | 深圳大学建筑与土木工程学院院馆设计 | 陈 蕴 |
| 209/ | 德国波茨坦大学图书馆设计 | 邓秉铸 |

信息工程学院

- | | | |
|------|-------------------------------|-----|
| 216/ | 基于模拟退火的学习矢量量化图象编码算法 | 何 东 |
| 223/ | 可编程语音芯片的设计及用 FPGA 硬件实现
的研究 | 王卓彦 |

231/	基于小波域的数字图像水印算法	潘宏嘉
237/	高精度椭偏光谱仪数据采集和分析系统	袁治国
244/	基于前向纠错码分多址的 HFC 网上行信道 噪声抑制	陈伟良

师范学院

252/	广义儒歇定理	陈晓华
258/	非中心差商收敛性与可导	陈云华
264/	Béziér 曲线、曲面的图形设计	陈国斌
273/	现代教育技术 CAI 系列软件之 Authorware 篇	郭宇辉
278/	视频展示台信号处理电路	刘昌满
286/	中学物理资料元集成系统	钟国洪
292/	推进剂燃速预估技术在配方设计中的应用 研究 CMDB	曹 铮
313/	化合物性能数据库智能管理系统研究	廖 矗
332/	Dirichlet 型空间上 Hankel 算子和乘法算子	耿莺歌

智能牌照系统之车载电脑的研究

周宇龙

(理学院应用物理系应用物理学专业)

§ 1. 序言

§ 1.1 智能牌照系统简介

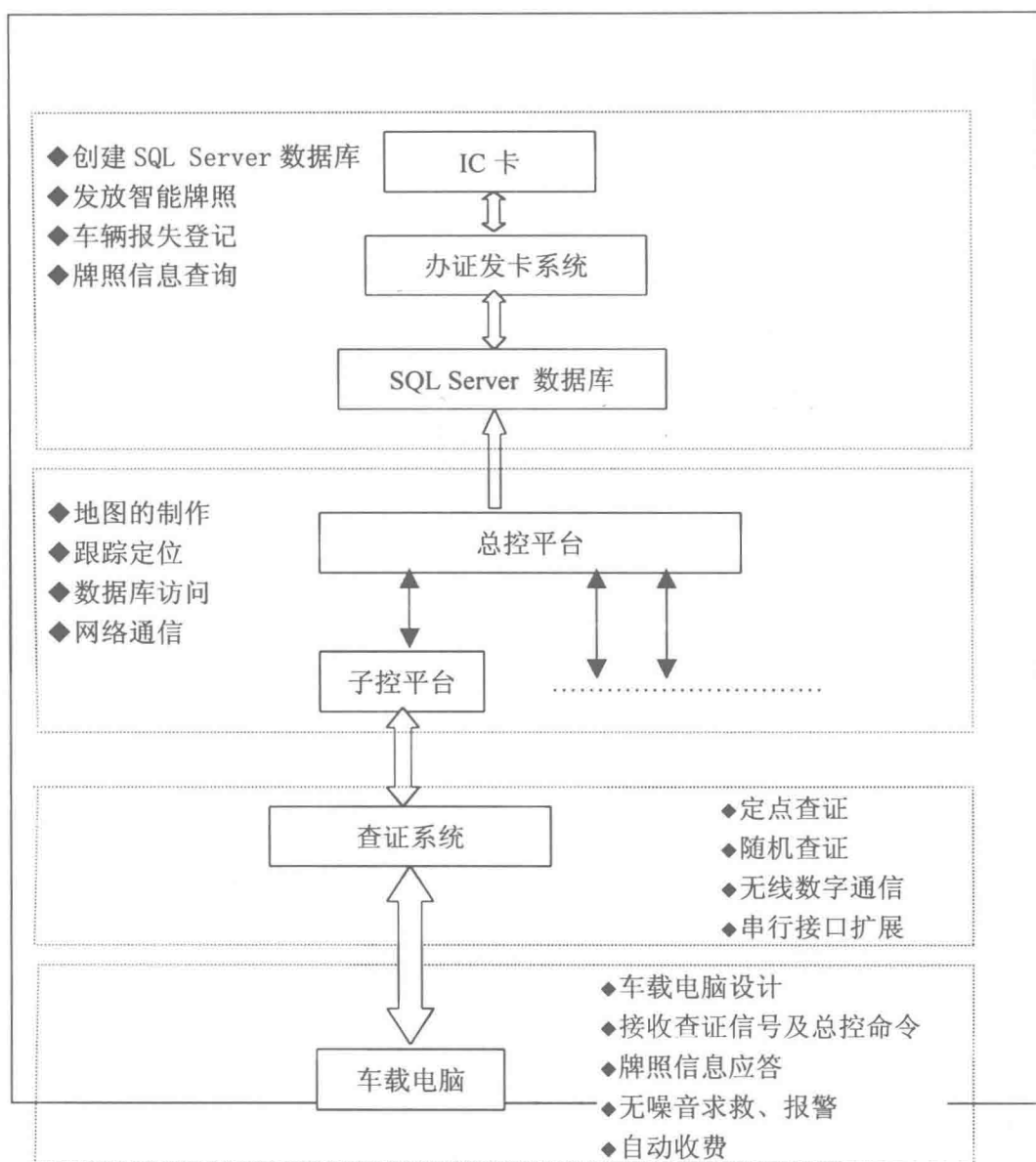
一、问题的提出

机动车走私、丢失已成为一项严重的社会问题，我们经常从各种媒体上看到各地花费大量的人力财力打击盗车犯罪、非法行驶车辆的报道。可以说这已经是全国甚至全球的普遍性问题。解决这个问题的关键在于有效地制止无照行驶、无证驾驶，使脏车不能上路。从而使走私和盗窃车辆的犯罪行为失去利益支点，从根本上解决问题。现行的停车验证办法很难奏效：一是车流量日益增大，从交通警力上不可能全面实施；其次停车检查容易造成交通堵塞，不可能经常进行。这样就给脏车避检、逃检、拒检造成机会；再说伪证、伪牌也难杜绝，有些伪证、伪牌甚至可以达到以假乱真的程度。

随着高新科技的迅猛发展，解决上述问题日趋可行。我们研究的《智能牌照和车辆管理智能化系统》项目充分应用现今快速发展的计算机信息管理技术、网络技术、无线数字通信技术、智能卡技术等来实现对车辆信息智能管理，从根本上制止无证驾驶、脏车上路。

二、系统总体设计

本系统由智能牌照办证发卡系统、车载电脑、查证系统及车辆智能网络系统（简称总控系统）四个部分组成，具体结构如图所示：



◆ 办证系统把车牌信息与驾照信息写入 IC 卡，发放智能牌照，同时建立相关信息的网络数据库，供总控系统使用。

◆ 在每辆车上装载具有接收查证信号和总控命令、应答牌照信息、发送无噪声求救信号和报警信号等功能的专用电脑。车载电脑在检查到有合法牌照后才允许车辆启动，当接收到查证信号时，将以智能牌照的相关信息应答。

◆ 查证系统（分为设置在主要交通路口的定点查证和由交警手持遥控的随机查证）以无线通信方式接收这些信息并传送到总控系统。

◆ 总控系统具有地理信息处理、对车辆定位跟踪的功能，并即时提供车辆与驾驶员的详细信息查询功能。

对车辆采用智能牌照的管理，无证、伪证车辆的运行就完全暴露在总控系统的视野中，所以可以有效的制止无证驾驶。同时还方便于扩展包括电子防盗、自动收费、车辆年审验证、违章登记、处罚认定、交通事故、事实报警与跟踪等功能。

§ 1.2 车载电脑简介

在整个智能牌照系统中，车上的车载电脑作为系统的一个终端，无疑有着举足轻重的作用。从总的来说，它应该具备读写智能卡、跟外部进行数据交换、处理数据并根据结果对车辆实行控制等功能，其中对智能卡的读写部分是难点。

本车载电脑由一片 MCS-51 单片机和少数几个外围器件构成，最大特点是器件少，成本低，可靠性高。由于车上的环境比较恶劣，温度波动范围大，车载电脑容易受到环境的干扰，因此抗干扰能力是一个不可忽视的问题。为了提高车载电脑的可靠性，本方案同时从硬件和软件两个方面着手，力求提高其抗干扰能力。在硬件上，问题主要出在由于温度变化而引起的电压漂移以及电路分布参数的干扰，因此尽量减少器件数目，降低电路的复杂度是最有效的办法；此外，使用稳压器件稳定电源的电压，并把智能卡的电源从其它电路独中立出来也是很好的策略。在软件上，为了纠正数据在传输期间出现错误，可以采用奇偶校验技术，解决部分问题。

我们这个智能牌照系统是要实现以卡代证，一卡多用，这样才能使得管理更简单有效，因此，车载电脑必须能够识别和处理不同的命令，可用于查证、路桥收费、泊车收费、加油收费等。当收到系统发来的停车命令时能关掉油路，当司机在车上遇到紧急情况时可以发出求救信号。

由于整个系统的应用跟 IC 卡密切相关，而车载电脑的核心也在于跟 IC 卡交换数据，因此要开发车载电脑必须对 IC 卡非常了解，为此，我在本文的第二章简单介绍了一些智能卡的相关知识。第三章介绍车载电脑的软硬件开发以及在开发过程中常遇到的问题和解决的办法。第四章总结全文，对本车载电脑进行可行性分析和提出改进方向。

§2. 智能卡相关知识

§2.1 智能卡概况

智能卡的名称起源于英文“Smart Card”，又称为集成电路卡（Integrated Circuit Card），即 IC 卡。IC 卡是 70 年代由法国人提出来的，Bull 公司首先制成产品，并将这项技术应用到金融、交通、医疗、身份证明等领域。

根据集成电路的不同，IC 卡可以分为以下三类：

- (1) 存储卡：卡中的集成电路为 EEPROM，这种 IC 卡只有数据存储能力，没有数据处理能力，也没有数据加密功能，一般用于安全要求不高的领域。
- (2) 逻辑加密卡：卡中的集成电路具有逻辑加密功能，在对卡片进行写操作之前必须验证密码。卡内有一个错误计数器，当连续三次密码错误时自动锁卡；如果三次错误前有一次正确则自动恢复错误计数器。这种卡型应用非常广泛，最典型的应用是电话 IC 卡。
- (3) CPU 卡：准确地说，只有这种带 CPU 的卡型才算得上智能卡。这种 IC 卡内的集成电路中含有 CPU、EEPROM、RAM 以及固化在 ROM 中的片内操作系统 COS

(Chip Operating System)。CPU 卡有数据处理能力，数据以文件的形式储存，可以实现一卡多用，并且具有高保密性，一般的金卡都是 CPU 卡。考虑到我们这个系统也要求很高的安全性，并且要实现一卡多用，所以选择了 CPU 卡。

根据卡的触点电极是否与读写器接触，又可以分为接触式 IC 卡和非接触式 IC 卡（也叫射频卡）两种。非接触式 IC 卡通过无线电方式跟读写器交换数据，因为卡和读写器之间没有接触摩擦，所以寿命比接触式长，使用也方便，将是以后 IC 卡发展的主流。由于非接触式读写器的成本比较高，自己制作的难度又非常大，所以我们这个系统暂时没有采用它。

随着科学技术的迅速发展，全球的市场正在趋于一体化，不同企业之间的产品要互相兼容才有利于长远的发展，这就需要国际化标准组织（ISO）对其进行一些规范。ISO 7816 标准对 IC 卡的尺寸大小、材料化学性质、触点大小、各种电信号和传输协议等都作了详细的规定。下面简单介绍几个标准：

(1) 触点功能如图(1)所示，

- C1 Vcc(电源电压)
- C2 RST(复位信号)
- C3 CLK(时钟信号)
- C4 保留未用
- C5 GND(信号地)
- C6 Vpp(编程电压)
- C7 I/O(输入/输出)
- C8 保留未用

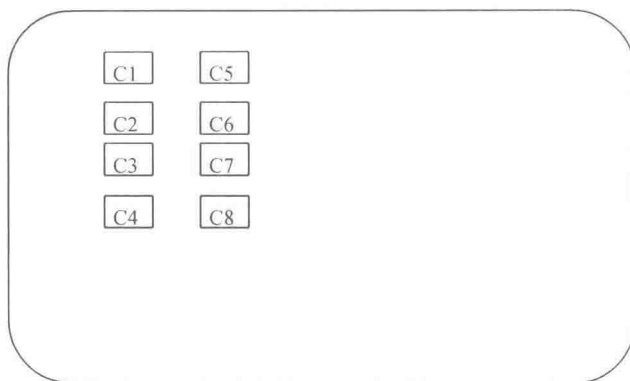


图 (1) IC 卡的触点功能

在 ISO7816 标准中对每个触点的中心位置都有规定，虽然对触点的形状没有规定，但规定了触点的最小尺寸为 2.0mm x 1.7mm。

(2) 标准还对 Vcc、RST、CLK、Vpp、I/O 等信号的电气参数都作了具体的规定。

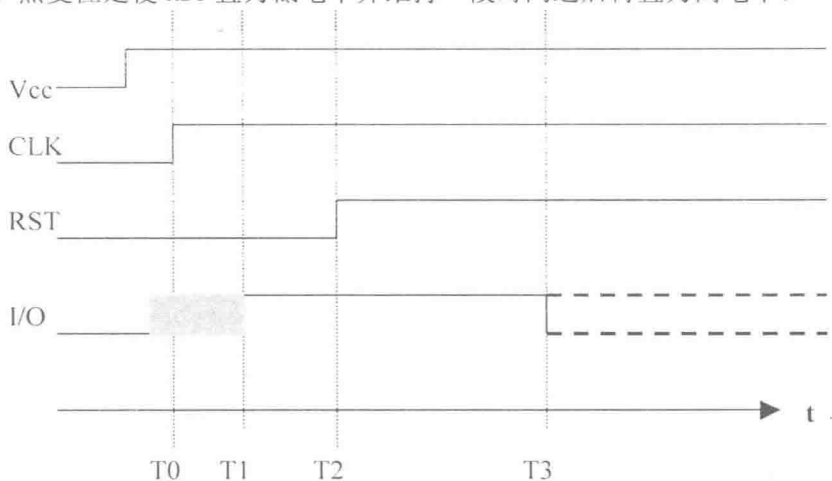
Vcc=5.0±0.25V；对 Vpp 也有详细规定，但一般都悬空不使用，因为 IC 卡内有升压泵，当要写卡时会启动升压泵将电压提高到 12V；IC 卡对时钟 CLK 要求很高：占空比在 45%至 55%之间，频率 $f > 1\text{MHz}$ ， $t_R, t_F < 9\%T$ ，其中 t_R 为信号幅度从 10%上升至 90%的时间， t_F 为信号幅度从 90%下降至 10%的时间， T 为周期。

(3) 数据的传输有同步和异步两种方式，同步方式通过 CLK 进行同步，这种传输格式符合 I²C 总线协议，可以实现多个器件通信，非 CPU 卡都采用这种传输方式。CPU 卡采用异步传输方式，跟通用的串行通信很相似，但 IC 卡只有一根数据线，输入输出都要通过它进行，因此正确地切换输入输出状态很重要，空闲时使之处于接收状态。多数的 IC 卡内部都没有时钟，就把输入时钟 CLK 作为卡的内部工作时钟。卡与外部通信传输一位的时间称为一个基本时间 (etu)，显然一个基本时间就等于比特率的倒数 (即 $1\text{etu} = 1/\text{bps}$)。传输数据时 $\text{etu} = D * F / f_s$ ，其中 F 为时钟频率变换因子，默认值为 372； D 为比特率调整因子，默认值为 1， F 和 D 可以通过在 IC 卡建立 ATR 文件来设定； f_s 为 CLK 的频率。

(4) IC 卡跟读写设备的操作流程如下：

- 1) 插卡, 读写设备连接和激活触点
- 2) 使卡复位
- 3) 卡复位应答
- 4) 卡和读写设备进行信息交换 (主体部分)
- 5) 释放触点, 拔卡

为了保护集成块, 在插卡的时候必须保证所有触点不带电, 卡到位之后再给卡上电。上电的次序也有讲究, 如图 (2) 首先加电源 V_{cc} , 并使其它信号处于低电平。待电源稳定后就可以给卡片加 CLK (设此时时刻为 T_0), I/O 线必须在 T_0 之后 200 个周期内置为接收状态 (高电平), 也可以在 CLK 之前就置为高电平。RST 在 T_0 之后至少维持 40000 个周期的低电平, 然后置为高电平 (T_2 时刻), IC 卡应该在 T_2 之后的 400 至 40000 个周期之间返回应答信号 (T_3 时刻), 如果没有返回应答信号, 则读写设备可以启动热复位。热复位是使 RST 置为低电平并维持一段时间之后再置为高电平。



图(2) 卡的复位时序

(5) 复位应答信号

1) 异步传输的字符帧格式。IC 卡传输字符的格式和一般的异步传输格式相同, 空闲时处于高电平, 传输时先传一个起始位, 然后先低位后高位传送数据, 最后传送奇偶校验位。一般的串行异步通信的奇偶校验位是可选的, 而对 IC 卡而言该位的则是必须, 如果接收端发现传输有误则必须在收到停止位之后的 $0.3et_u$ 至 $0.7et_u$ 之间将 I/O 线置为低电平, 发送方在送完停止位之后 $0.9et_u$ 至 $1.2et_u$ 之间检测 I/O 线, 如果测得结果为低电平则说明传输有误, 要求重新发送上一个字符。正确传输的字符帧如图 (3)。

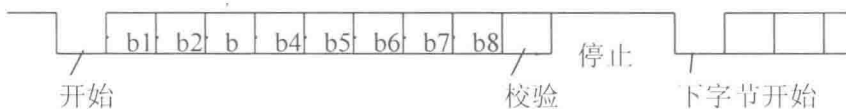


图 (3) 字符帧

2) 复位应答信号 (ATR) 的结构和内容。如果复位成功, IC 卡会返回应答信号, 它的结构如图 (4)。

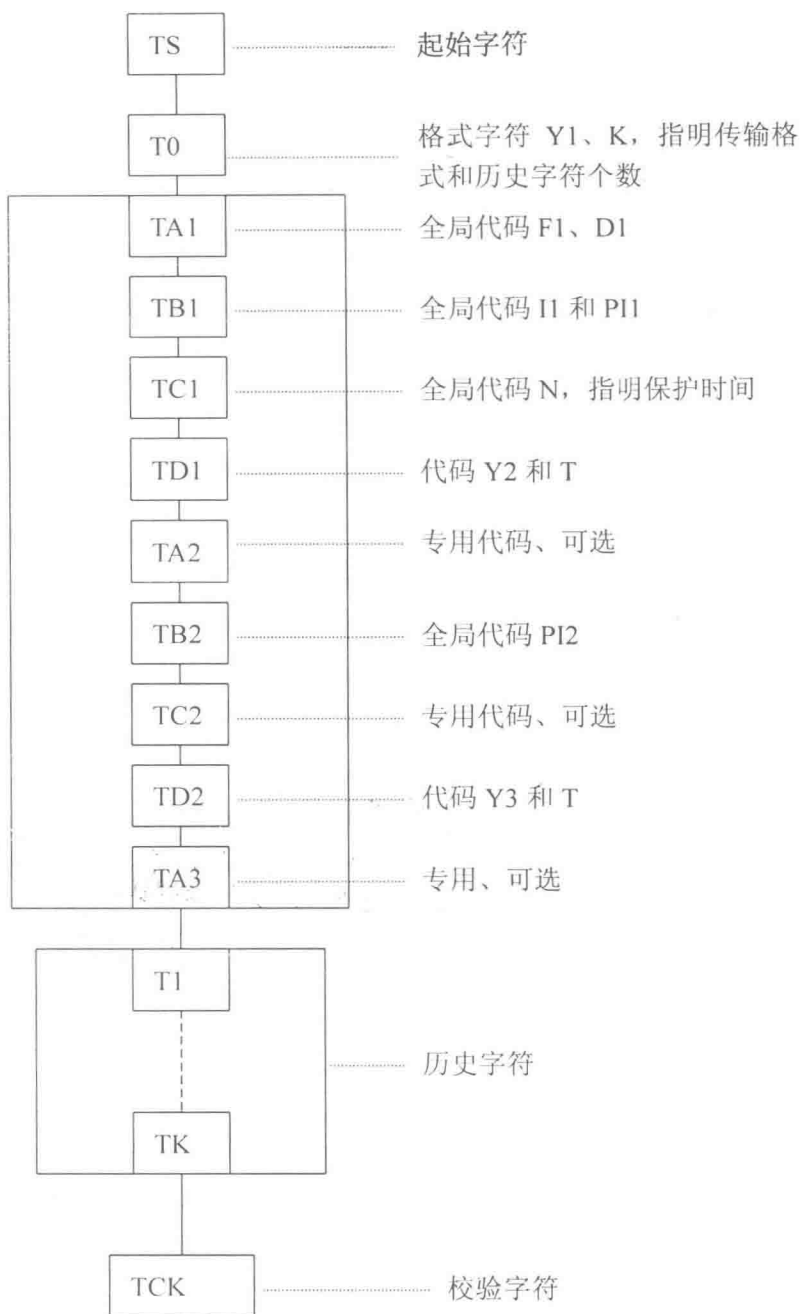


图 (4) 复位信号的组成

§ 2.2 智能卡的操作系统 SmartCOS

和 PC 机的 DOS 操作系统类似, CPU 卡内部也有文件操作系统。操作系统是为了屏蔽底层的硬件, 为用户提供面向应用的界面, 通过这个界面, 用户不必了解卡的内部结构,

只须使用操作系统提供的面向文件操作的命令，就可以对 IC 卡进行各种操作。目前，据我了解，国内有两个公司开发出自己的 COS，一个是深圳明华公司的 SmartCOS，另一个是北京握奇公司的 TimeCOS，我们用的是 SmartCOS，所以下面我介绍的是 SmartCOS。

§ 2.2.1 SmartCOS 的文件系统

CPU 卡内的数据是以文件的形式储存的，这样用户就可以方便地使用命令直接对数据进行读、写、加密、认证等操作。SmartCOS 操作系统的文件管理跟 DOS 很相似，有一个主控文件 MF (Master File)，MF 之下可以有基本文件 EF (Elementary File) 和专用文件 DF (Dedicated File)。MF 即相当于跟目录，DF 相当于目录文件的文件夹，EF 才是储存数据的文件。一个卡片内有且仅有一个 MF，MF 的储存空间即是整个卡片的文件。一般在规划文件结构时使一个 DF 为一个应用，这样就可以实现一卡多用，一个文件夹一种应用，方便管理。ISO7816 规定 DF 之下可以有 DF，但 SmartCOS 的 DF 之下不能建 DF，只能建 EF。

EF 是真正储存数据的文件。EF 可以分为安全基本文件 (SEF) 和工作基本文件 (WEF) 两类。安全基本文件是用于储存个人密码和各种密钥的文件，其内容是不可以读出的，但在满足特定权限时可以修改，在每个 MF 和 DF 之下只能建立一个安全基本文件。工作基本文件是储存用户的应用数据的文件，其内容不被卡解释。在满足一定的条件时工作基本文件的内容可进行读、写和加密操作。

SmartCOS 支持如下四种基本文件结构：

- 1) 二进制文件结构。有的书也称之为透明文件，因为二进制文件的数据对 IC 卡是透明的，IC 卡不解释其中的内容。二进制文件跟记录文件中只有一个记录时的特殊情况有点相似，但其读写的指令是不同的，二进制文件通过指定偏移地址和读写的字节数来访问；而记录文件则通过记录号来实现。在文件建立时要指明文件的大小。
- 2) 线性定长记录文件结构。这种文件结构以固定的长度来储存数据，在建立文件时指明记录长度和记录数，处理时以记录为单位进行。各种 COS 的最大记录长度和最大记录数可能不同，SmartCOS 的最大记录长度为 110 字节，最大记录数为 254。
- 3) 线性变长记录文件结构。这种文件结构的每条记录的长度可以不同，仍然以记录号来访问数据。文件的空间在建立时指定，以后在文件未满时可以用 Append Record 命令追加记录；如果已有记录的总空间已经达到文件在建立时指定的大小，则可以用 Update Record 更新记录，但更新的内容的长度不能超过原记录的长度。
- 4) 循环记录文件结构。这种文件结构也是以定长记录的形式储存数据，在文件建立时指定记录长度和记录数。与定长记录文件不同的是循环记录文件只能追加记录 (Append Record)，不能用更新命令 (Update Record)。追加记录时新追加的记录号为第一号，上次加的为第二号，依次类推，越早加入的记录号越大，当记录号大于建立时指定的最大记录号时自动删掉，这样就可以保存最新的一些记录。

§2.2.2 SmartCOS 的安全系统

IC 卡从设计、生产、运输、应用规划、发行、使用到销毁的整个生命期内都要考虑到安全性。卡片的设计和生要保证其保密性，从商业的利益出发，制造商都会做到这点。为了保证卡片安全地运送到发行商，卡片制造商在制造卡片时写入一些运输密码，运输密码通过安全可靠的途径交给应用发行商，发行商在进行应用规划时首先要输入运输密码，如果正确则可以进行应用规划，否则不可用，这样可以保证用户的合法性。

为了保证卡在使用期间的安全性，发行商在对卡进行应用规划时必须格外小心谨慎，因为很可能因为规划时的一个漏洞而导致不可估量的损失，尤其是金融卡。下面主要讲述如何从文件规划方面来保证卡在使用过程的安全性。

文件在建立时都要指定其读权限和写（更新）权限，读权限和写权限是互相独立的，各用一个字节表示，文件的读写权限成为文件的属性之一。而用户在某个文件下都具有一个确定的安全状态，安全状态也用一个字节表示，取值在 00h~0fh 之间，只有用户的安全状态满足要求时才可以访问卡内数据。设文件的写权限为十六进制 XY，用户当前的安全状态为 M，则只有当 $X \leq M \leq Y$ 时才具有写权限。只有通过校验个人密码和外部认证两种方法可以改变当前的安全状态。复位后用户的状态为 0，如果没有经过个人密码校验或外部认证则一直保持该状态。在规划文件时如果不想把某种权限给最终用户，可以把它的权限设置为 $X > Y$ ，因为用户永远无法达到这样的状态；如果想把某个文件权限可以给最终用户自然获得，可以把它的权限设置为 0Y，因为 0 是用户自然获得的状态。在实际应用文件中，一般不允许用户随意修改卡内的数据，即使可以修改也要经过必要的认证措施，以保证访问者的合法性。如果用户在某个 DF 下通过认证而获得了某个状态，但当离开这个 DF 后将回到状态 0。

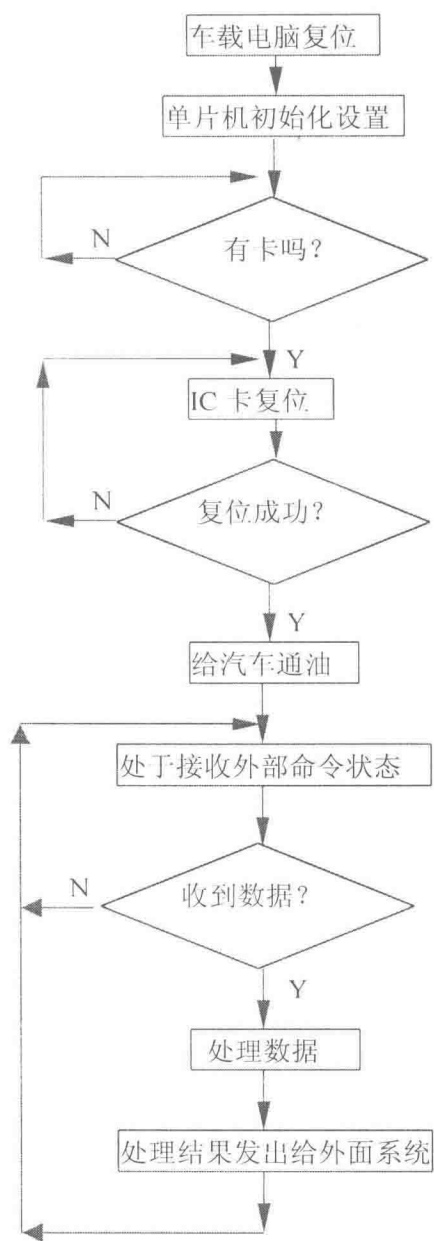
在建立密钥文件时给个人密码和外部认证密钥设定了错误允许计数初值，进行个人密码校验或外部认证时错一次计数器减一，当计数值等于 0 时自动把卡的该目录锁定。在 IC 卡和外部进行数据交换时也存在安全性问题，因为数据有可能被中途截取甚至修改，因此 IC 卡提供了密文传输的功能，数据可以加密之后传输，并且通过在信息尾部加上起校验作用的信息认证码（MAC），以保证数据的完整性。

§3 车载电脑的设计

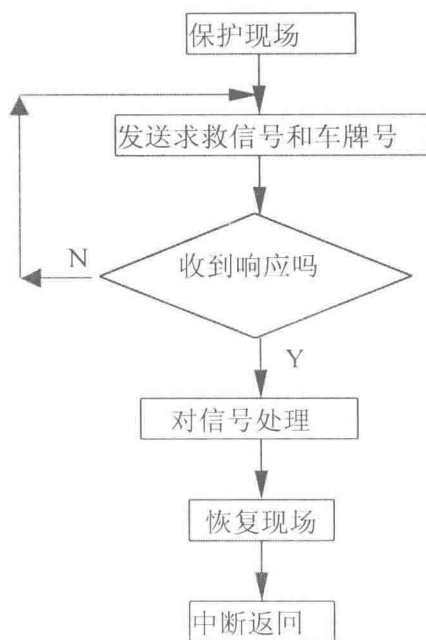
本章详细讲述车载电脑的设计方法，分四个部分进行。第一部分讲述整个设计方案的框架；第二部分讲述硬件的设计方法以及器件的选择；第三部分讲述软件方面的设计，这部分是本文的核心；第四部分讲述设计过程中遇到的问题及其解决方法。

§3.1 整体设计方案

本车载电脑要实现如下功能：无卡者不能开车、接收检测站发来的查证命令、实现路桥不停车收费、泊车收费、紧急情况呼救等功能。可以总结为读写卡、跟外部通信和数据处理三大功能。整体框架如图（5），求救中断图（6）：



图（5）主流程图



图（6）呼救中断处理流程

图（5）中收到数据后对数据进行处理是程序的关键，跟 IC 卡交换数据也是在这部分。车载电脑在复位后初始化时把外部中断 EX0 设置为高优先级，这样就可以保证在遇到紧急情况时能够及时得到处理，处理流程见图（6）。

§3.2 硬件设计

硬件设计主要是解决产生时钟、多个控制信号之间的切换、保护 IC 卡、稳压等问题，

见附录 3 的硬件结构总图。下面就各个问题给予剖析。

- (1) IC 卡的时钟问题。按照要求，时钟的频率范围为 1M Hz ~ 5M Hz，占空比在 45% 至 55% 之间，上升时间 t_R 和下降 t_F 时间均小于周期的 9%，也就是要求波形是比较陡的准方波。开始时我考虑使用单片机的地址锁存信号 (ALE) 作时钟，用示波器测得其频率为 2M Hz，但占空比和 t_R 、 t_F 都达不到要求，用触发器二分频后变成 1M Hz 的方波，占空比没有问题，但 t_R 、 t_F 还是达不到要求。经过权衡之后决定使用晶振，因为晶振的频率稳定，占空为 50%，而且可以选择需要的频率，这一点尤为重要，因为 IC 卡 I/O 线上的比特率是输入时钟的 372 分频，如果选定通信比特率为 9600 bps，则时钟频率 $f=9600*372=3.5712\text{M Hz}$ ，我选择频率为 3.579M Hz 的晶振。如图 (7)，非门要用 74HC04，不能用 CD4069，否则波形不理想。

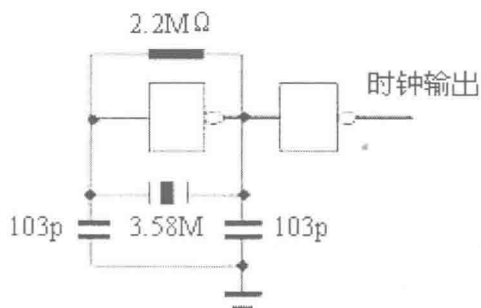
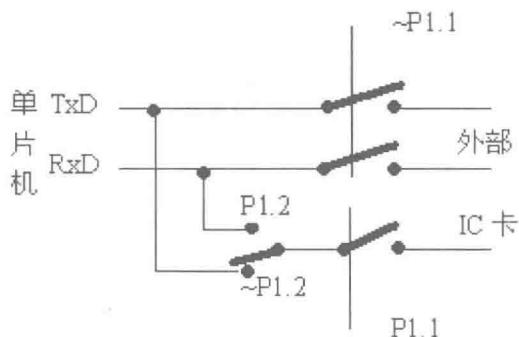


图 (7) 晶振电路



注：~ 表示取反，开关用 74HC4066
图 (8) 读、写、内、外控制信号

- (2) 因为单片机的串口既要跟 IC 卡交换数据，又要跟外面交换数据，这就需要一对互斥的选通开关来控制单片机是跟卡或是跟外面通信。我用 P1.1 来控制，当 P1.1=1 时跟 IC 卡通信，等于 0 时跟外面通信。见图 (8)。图中的开关用的是 74HC4066。
- (3) 注意到单片机串口的发送端 (Tx/D) 和接收端 (Rx/D) 是分开的，而 IC 卡只有一个 I/O 端口，收发合用，因此也要一对互斥的控制信号作选通。我用 P1.2 控制，当 P1.2=1 时读卡；P1.2=0 时写卡。见图 (8)。
- (4) 为了保护 IC 卡，在插卡的时候卡座应该不带电，否则很容易烧坏 IC 卡。这样就需要一个用于判断卡是否到位的信号，如果到位了就给卡片供电并且使单片机复位，否则撤掉通向 IC 卡的电源及各条信号线。见图 (9)。注意：开关要选用频率响应较好的 74HC4066，不可选用 CMOS 系列，否则时钟不能有效地通过。如图所示，把双向传输门 74HC4066 的四个通道的控制端 (5、6、12、13) 连在一起，受卡片是否到位而控制。当 IC 卡没有插到位时，控制信号通过金属片跟地相连，控制信号为低电平，传输门不导通；当 IC 卡插到位后，卡片顶起金属片，使控制信号跟地隔离，通过 2.20MΩ 电阻和电源相连，为高电平，使传输门导通，并且使单片机复位。这样就可以做到插卡时不带电，保护 IC 卡的安全。

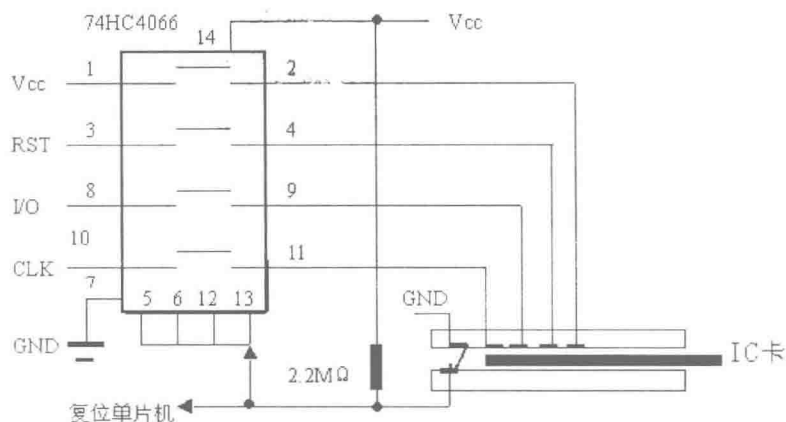


图 (9) IC卡保护电路

- (5) 稳压电路。稳压电路只简单地使用三端稳压器件 7805，该稳压器件效果还可以，能够减少电压的波动，基本能满足要求。电路如图 (10)。

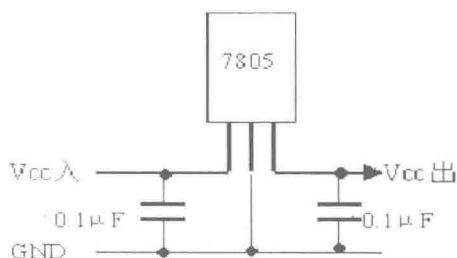


图 (10) 稳压电路

§ 3.3 软件设计

硬件设计主要是电路方面的工作，协调各种控制信号，保证系统正确运作。而软件部分则是车载电脑的单片机程序设计，各种控制信号的信号源都是从单片机发出的，所以软件设计是最重要部分，下面分三部分内容进行讲述。

§ 3.3.1 单片机程序设计

单片机程序要实现如下功能：读写 IC 卡、收发数据、处理数据、发出控制信号。程序的主流程见图（5），单片机复位后进行如下初始化设置工作：串口模式 3（8 位数据，一位奇偶校验，比特率可调，中断接收方式）；定时器 1 工作于模式 2（用于设置比特率，不允许定时器中断，定时器溢出时自动装入计数初值）；外部中断 0（用于发出求救信号，中断优先级最高）。

初始化定时器和串口后通过检测 P1.4 判断是否已经插卡到位（低电平表明卡到位），如果没有插卡则等待插卡，检测到卡后才继续往下执行，这样可以防止无证驾驶。检测到插卡后通过置 P1.0 为低电平向 IC 卡发出复位信号，由于受环境的干扰，第一次复位不一定成功，如果在 400 etu 内没有收到 IC 卡的复位应答信号，则说明复位失败，可再启动一次复位，直到复位正确为止，否则不给车通油，这样可以防止使用假卡。

IC 卡复位成功后就进入等待外部命令状态（有条件等待，如果标志位 B.0=1 则退出），准备进入主循环。当车载电脑收到外部数据后即进入串口中断处理程序，置等待标志 B.0 为 1，把数据暂存到 R0 所指的 RAM 内（R0 的初值设置为 50h）。中断返回后退出条件等待状态，进入延时状态。为了能把同一批数据收下来，但又不至于无限期等待下去，可以通过检测字符之间的时间间隔的办法来解决。如果从最近收到字符后至今已经 100 etu 没有收到新的字符，则认为这个数据包已经接收完毕；如果在 100 etu 内收到新的数据，则恢复延时初值，以保证相临两个字符间隔不超过 100 etu 的所有数据都能收到。

整个数据包接收下来之后就暂时关掉跟外面通信的通道，要对接收到的数据进行分析，如果是查证命令，则读 IC 卡，将卡内的驾驶证号读出放到 RAM，然后将车牌号（保存在单片机程序内）和驾驶证号一起发出。车牌号和证号发出后又打开跟外面的通信通道，处于延时状态，准备接收外部响应，如果收到停车命令（停车命令之后是车牌号码），则比较收到的车牌号码是否自己的号码，是则关掉油路停车，否则继续前进；如果没有收到停车命令则继续前进。这是查证情况的例子，路桥收费和泊车收费等其它应用情况也大同小异，在此不再一一列举。

当车上遇到紧急情况时，可以按下安装在比较隐蔽地方的触发按钮，该按钮引发一个外部中断，单片机立即进入这个中断处理程序并发出求救信号。求救信号包含该车的车牌号，检测站收到该信号后通过计算机网络传送到交管局总控中心，总控中心根据信号来源可以知道出事地点和车辆，根据情况作出处理。

下面补充说一下车载电脑如何向 IC 卡发指令。在 IC 卡和读写设备之间交换的数据是命令—响应对（Command—Response Pair），即读写器发送一个命令给 IC 卡，IC 卡执行命令后返回响应信息给读写器。命令信息结构包括四个字节的命令头和长度可变的命令体，如图（11）。其中命令头是必需的，命令体则根据命令的不同而不同，但数据长度参数 Lc 或 Le 最少有一个。下面解释各个参数的含义（若无声明所有数字均为十六进制，X 表示任意数字）。

命令头				命令体		
CLA	INS	P1	P2	Lc	Data	Le

图（11）

CLA：指令的类别。为 8X 表示外部命令，0X 表示内部命令；

X4 表示采用安全报文传送，X0 表示不采用安全报文。

INS：指令代码。各种不同的指令就是通过该字节来识别。

P1、P2：这两个字节用于指出命令的参数，不同的指令有不同的具体意义。

Lc：命令体中数据域的字节数，如果没有数据域则 Lc 也不存在，例如取随机数。

Data：传给 IC 卡的数据，它的长度为 Lc，有些命令没有这部分。

Le：IC 卡返回数据的字节数，有些命令可能没有这部分，例如追加记录命令。

响应信息结构包括数据体和状态信息两部分，见图（12）。其中状态信息是必备的，用于指明指令是否正确执行，如果出错则指出错误类型代码，而响应数据体也是根据不同的指令可能存在或不存在的，如果存在则应等于命令体中 Le 的大小。

响应数据	状态信息	
Data	SW1	SW2

图（12）

发指令时并不是整个命令所有字节一下发完的，而是先发命令头和 Lc 或 Le 之一（如果有数据体则发 Lc，否则发 Le），共五个字节。读写器发出这五个字节后就等待 IC 卡返回一个字节的过程字节，过程字节用于指示读写器下一步该怎么做，主要有如下几种情况：

- 1) 是否激活编程电压 V_{pp} 。一般 IC 卡都不用外部提供 V_{pp} ，只有极少数 IC 卡内部没有升压泵，当要写卡时需由读写器提供 12V 的编程电压。
- 2) 发送下一个字节。
- 3) 发送所有后续字节。
- 4) 返回 SW1，继续等待 SW2（这种情况一般由命令头出错引起）。

读写器接收到过程字节后对其分析进行并作出相应的处理。如果读写器发出的指令头没有错误，一般会收到“不用激活 V_{pp} 、发送所有后续字节”的指示。读写器收到这样的指示后把所有后续数据送给 IC 卡，之后等待 IC 卡的响应信息。读写 IC 卡的流程见图（13）。

值得注意的是：IC 卡收发的数据都是含奇偶校验位的，用的是偶校验，否则将出错：



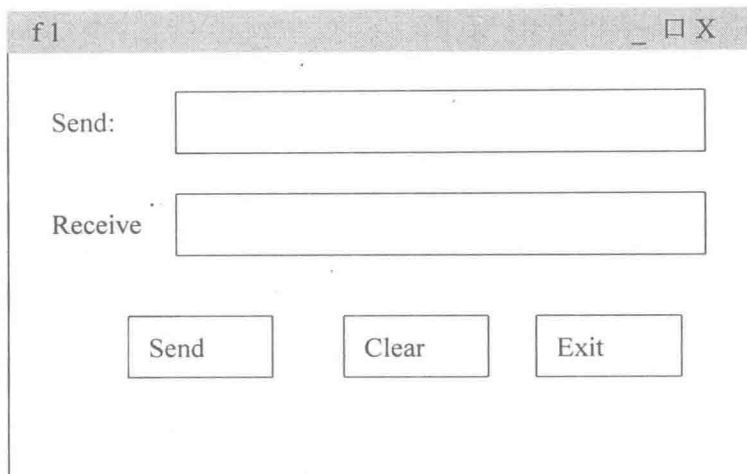
图（13）读写 IC 卡流程

还有一点是：IC 接收数据的速度比较慢，相临两个字符之间需要间隔一段时间，这段时间称为保护时间，SmartCOS 的 IC 卡需要 2 个 etu 的保护时间，也就是说单片机发送数据给 IC 卡时发完一个字符后要延时 2 个 etu 之后才能发送下一个字符，否则 IC 卡可能会出现接收错误。单片机的程序见附录 1。

§ 3.3.2 PC 机模拟程序的设计

在开发车载电脑过程中必须有外部设备跟它通信才能检验是否达到要求，并有助于发现问题和不断改进功能。为此，我在 Visual C++ 环境下设计了一个串口通信程序，用于发命令和检验车载电脑的功能。程序的界面如图（14），当按下 Send 命令时将上面那个编辑框的内容发送给单片机，发完之后处于接收状态，等待单片机的响应数据，当收到单片机的返回数据时就将其显示在 Receive 编辑框。按下 Clear 清除两个编辑框的内容，并把发送缓冲区和接收缓冲区清空。按下 Eixt 退出应用程序。

在 Visual C++的环境下，串口被当作文件来处理，通信之前要先进行初始化工作，打开串口，设置波特率、数据位、校验方式、停止位、字符时间间隔、收发缓冲区大小等。要注意的是：从键盘输入并显示在屏幕上字符串，变量 `m_Send` 得到的是各个字符的 ASCII 值，而不是各数字的十六进制数值。例如，从键盘输入“0084000008”并显示在 Send 编辑框内，我们的意愿是把这个字符串作为十六进制指令发出，从卡片取一个 8 字节的随机数，但实际上变量 `m_Send` 得到的数值是{30h,30h,38h,34h,30h,30h,30h,30h,30h,38h}，因此不能把这个字符串直接发出去，需要自己编写一个函数，把输入的 ASCII 字符转换成相应的十六进制数值，然后才发出去。同样，PC 机从单片机收到的是十六进制数，不便于显示，也要自己编一个函数将十六进制数转换成 ASCII 字符，然后再显示出来。



图（14）PC 程序界面

下面说说这个程序怎么编写。

- （1） 打开 Visual C++编辑环境，新建一个基于对话框的工程，设工程名为 f1。
- （2） 在对话框编辑器做一个如图（14）的界面，用 Class Wizard 给两个编辑框添加变量，名称分别为 `m_Send` 和 `m_Rec`，类型为 `CString`，最大长度为 60 个字符；给三个命令按钮添加单击消息处理函数，函数名分别为 `OnSend`、`OnClear` 和 `OnExit`。
- （3） 编写代码。以下代码都在文件 `f1Dlg.cpp` 之下。

1) 在 `f1Dlg.cpp` 文件的前面声明一个句柄，用于标识串口，语句如下：

HANDLE hCom;

2) 重载主窗口的 `ON_CREATE` 消息的处理函数 `OnCreate()`，在该函数内进行串口的初始化工作，代码如下：

```
int CF1Dlg::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if (CDialog::OnCreate(lpCreateStruct) == -1)
        return -1;

    DCB dcb;
    DWORD dwError;
    BOOL fSuccess;

    //打开串口
    hCom=CreateFile("COM1",GENERIC_READ|GENERIC_WRITE,
        0,NULL,OPEN_EXISTING,0,NULL);
    if(hCom==INVALID_HANDLE_VALUE)
    {
```