

机车入库无线采集 系统

机车入库无线采集系统是在机车即将入库时对列车重要信息进行无线采集，取代以往机车入库后，利用存储卡到机务段的微机系统中读出列车信息，这样费时、费力，而且不能及时对机车信息进行采集。本设计实现一无线数据采集系统，能在机车入库的途中完成数据的自动采集。

本设计在功能实现上可分为三个模块，依次是无线数据传输模块、串口通信模块、数据终端处理模块。

数据无线传输模块是利用桑锐 SRWF-501 芯片实现的，它与主机 RS-232 串口相连，将 RS-232 串口的数据无线发射，或接收无线数据再传给 RS-232 串口，从而实现两台主机数据的无线传输。

串口通信部分的功能则是架设一座数据从程序到外界的桥梁，主机的数据都是经过这个桥梁传输出去，本设计采用 RS-232 串口完成这一任务。

终端数据库主要完成数据的存储、提取以及查询等数据操作，服务于列车调度，本设计采用微软 ACCESS 数据库实现这部分的功能。

通过这三个模块的设计，实现了机车入库时数据无线采集的过程，极大的提高了列车管理的效率。

关键词 桑瑞 SRWF-501, RS-232, 串口通信, 无线采集

ABSTRACT

Train information's Wireless Collection System is designed to collect the important information wirelessly when a train is going to enter the station. The new wireless way can replace the old way in which railway station staffers have to get the important information from the computer system of computer center with a memory card. The old system can't collect the important information efficiently and even isn't able to gain the important information sometimes. In this design, I complete a wireless collection system which can be used to collect information in an automatic way while a train is going to enter the station.

This design is mainly made up of three parts: Shanghai Sangrui SRWF-501 module, RS-232 serial port, and terminal database part.

Shanghai Sangrui SRWF-501 module takes charge the wireless communication. It connects with the computer by RS-232 serial port and beams the data of RS-232 serial port or transmits the data it has received into RS-232 serial port.

To the data, RS-232 serial port is a bridge between the program and Shanghai Sangrui SRWF-501 module. All the data in the program is transmitted through this port.

Microsoft Access is used in this design as terminal database to accomplish the storage and management of data.

Through the design of the three parts, the important message of a train can be gained in wireless way when the train hasn't entered the station, which will improve the management efficiency largely.

KEY WORDS SRWF-501, RS-232, serial port communication, wireless collection

目录

摘 要.....	I
ABSTRACT.....	II
第一章 绪论.....	1
1. 1 研究背景及意义.....	1
1. 1. 1 列车调度简介.....	1
1. 1. 2 列车信息采集方式.....	1
1. 1. 3 该设计的意义.....	2
1. 2 列车信息采集及其意义.....	2
1. 2. 1 列车号.....	2
1. 2. 2 司机代号.....	3
1. 2. 3 最高速度.....	4
1. 2. 4 最高轴温.....	4
1. 2. 5 耗电量.....	5
1. 2. 6 故障.....	5
1. 3 国内外发展现状及本设计的定位.....	6
1. 3. 1 国内外发展现状.....	6
1. 3. 2 本设计的功能及定位.....	7
1. 4 论文研究内容.....	7
1. 4. 1 SRWF-501 型模块.....	8
1. 4. 2 关于 RS-232 串口.....	8
1. 4. 3 ACCESS 数据库.....	8
1. 5 论文组织结构.....	9
第二章 基于上海桑锐 SRWF-501 无线模块的设计.....	10
2. 1 SRWF-501 无线模块环境介绍.....	10
2. 1. 1 SRWF-501 无线模块结构.....	10
2. 1. 2 SRWF-501 无线模块集成环境.....	10
2. 1. 3 SRWF-501 无线模块图片.....	11
2. 1. 4 SRWF-501-50 型无线模块接口的定义.....	12
2. 2 本设计的硬件设计.....	14
2. 2. 1 本设计的硬件参数.....	14
2. 2. 2 开发环境.....	14
2. 2. 3 硬件连接.....	14
2. 2. 4 硬件设计过程.....	14
2. 3 小结.....	16
第三章 基于 RS-232 串口进行的通信.....	17
3. 1 串行通信.....	17
3. 1. 1 串行通信与并行通信.....	17
3. 1. 2 串行通信接口.....	18
3. 1. 3 串行通信实现方式.....	18

3. 2 RS-232 串行通信接口	18
3. 2. 1 RS-232 在计算机的位置.....	18
3. 2. 2 RS-232 针脚图.....	18
3. 2. 3 RS-232 针脚介绍.....	19
3. 2. 4 RS-232 针脚连接.....	19
3. 3 利用 RS-232 进行通信.....	19
3. 3. 1 MSCComm 控件简介.....	19
3. 3. 2 MSCComm 控件的引用.....	20
3. 3. 3 MSCComm 的工作方式.....	20
3. 3. 4 MSCComm 的操控原则.....	20
3. 3. 5 MSCComm 控件属性介绍.....	20
3. 3. 6 MSCComm 控件属性的应用.....	21
3. 4 本设计中的 RS-232 串口应用.....	21
3. 5 小结.....	22
第四章 基于 ACCESS 数据库的数据存储与提取.....	23
4. 1 数据的存储与提取.....	23
4. 1. 1 文件	23
4. 1. 2 数据库	23
4. 1. 3 数据库与文件的对比	24
4. 2 本程序中 ACCESS 数据库的应用	24
4. 2. 1 程序中数据库的主要功能	24
4. 2. 2 程序中数据库的设计与实现	24
4. 3 本设计中的数据库应用.....	27
4. 3. 1 发送端数据库	27
4. 3. 2 接收端数据库	27
4. 4 小结.....	28
第五章 功能的整合与程序的实现	29
5. 1 功能的整合.....	29
5. 2 程序流程图.....	29
5. 3 程序界面.....	30
5. 3 本设计主要模块的程序实现.....	32
5. 4 本设计的软件调试过程.....	35
第六章 结束语	36
6. 1 结论总结.....	36
6. 2 未来展望.....	36
致谢	37
参考文献	38
附录.....	39

第一章 绪论

目前, 机车入库后, 机车的的信息基本上是利用存储卡再到机务段的微机系统中读出, 这样费时、费力, 而且不能及时对机车信息进行采集。这样低效率的采集方式在现在这个列车提速的大背景中成为了瓶颈, 急待解决。本设计中设计一无线数据采集系统, 能在机车入库的途中实现数据的自动采集, 这样极大的提高了对列车的调度效率, 有着非常重要的现实意义。

1. 1 研究背景及意义

随着国民经济的高速发展和人流的日益加大, 铁路部门的承载能力正面临着严峻的挑战, 而对于此种情况最行之有效的方法是提高列车调度效率, 据现在铁路部门的现实状况的调查, 不能及时采集到列车运行的信息是一个非常突出的问题。

1. 1. 1 列车调度简介^[1]

很多火车在运行, 但是铁道是固定的, 火车怎么在道轨之间选择自己的路线而不会发生撞车事故呢? 其实, 火车站有个调度室, 有像黑板一样的台子 (调度台), 上面有密密麻麻的很多线路, 还有很多的小灯, 火车要到站的时候前面的站会发信号或者打内线过来的, 像对讲机一样的, 这边的操作调度台上是有一个调度指挥管理系统的, 可以监视列车运行及信号设备的工作状态, 实时传送各种报警信息, 实时传送阶段计划和调度命令, 值班的人只要操作就可以了。另外, 有缓冲轨的, 在几条轨道上完成对车厢的调度。

1. 1. 2 列车信息采集方式

就信息采集方式而言, 可分为现场采集和无线采集。

现场采集是目前应用比较多的一种方式, 利用存储卡再到机务段的微机系统中读出, 这样费时、费力, 而且不能及时对机车信息进行采集。

无线采集方式是在列车即将进站入库时向车库发送无线信息, 车库进行接收, 然后进行处理, 以这样的方式进行信息采集, 极大的提高了信息的采集效率, 有利于快速得到最有价值的列车信息, 将在未来成为主流。

1. 1. 3 该设计的意义

此次设计完成的是信息无线采集的课题,基本方法是应用上海桑锐 SRWF-501 无线模块,在通过 RS-232 串口与计算机相连,这样可以使计算机中的信息通过 RS-232 串口传往 SRWF-501 无线模块,SRWF-501 无线模块收到信息后即可向外传送,当车库收到列车发送的信息后就开始接受,这样列车即将进站入库时就可以得到列车的信息了,这样显然极大地提高了信息采集的效率,在现实中一旦投入使用,必将为提高列车的信息采集作出巨大贡献,进而又必然提高-列车调度效率地提高,创造出更高地国民经济效益。

1. 2 列车信息采集及其意义

前一小节介绍了此次设计的大背景是列车的全国范围内的提速,以及列车承载量的空前增加,这除了要求着高效率的调度,而且对列车的安全运行提出了更高的要求,如果我们能在车库及时了解列车的具体状况,对于工作人员能更好的控制机车有着非常重要的意义。

1. 2. 1 列车号

列车号是列车最直接的指代,我们习惯上也是用列车号来描述列车的,所以列车即将进站入库时最先传回车库的信息就是列车号。下面简单介绍一下列车号码。

T 开头的列车:特别快速旅客列车,简称特快,字母 T 是“特”字汉语拼音的简写。这样的列车在行程中一般只经停省会城市或当地的大型城市。这类列车的车底一般都是 25K(蓝皮车)。

K 开头的列车:快速旅客列车,简称快速,字母 K 是“快”字汉语拼音的简写。这样的列车在行程中一般只经停地级行政中心或重要的县级行政中心。这类列车的车底一般是 25G(红皮车)。

N 开头的列车:管内快速旅客列车,简称管内快速,字母 N 是“内”字汉语拼音的简写。这样的列车一般经停一些重要车站。这类列车的车底有 25G 25B(红皮绿皮的都有) 22 型绿皮车,空调列车较多。

四位数的车也有车次区分

1 开头的四位数车次列车:跨三个或以上铁路局的直通普通快速旅客列车,简称普快或直快。这样的列车一般经停一些主要车站。

2 开头的四位数车次列车:跨两个铁路局的直通普通快速旅客列车,简称普快

或直快。这样的列车和 1 开头的四位数经停车站和车底配置差不多,但空调列车的数量明显减少。这类列车运营的距离一般属于中等,经停的车站也较多,速度一般也不会太快。属于面向中途旅行的普快列车。

3 开头的四位数车次列车到目前为止还没有。

4 和 5 开头的四位数车次列车:管内普通快速列车,简称普快,曾经简称快客,但这个简称现在已经不用了。这样的列车经停一些主要车站,同时也经停一些小型车站。这类列车的车底主要是 22 型和 25B 型绿皮车,少有 25G 和 25B 的红皮空调车,而空调车多是一些长途列车套跑的。这类列车运营的里程一般不长,一般属于短途列车。

6/7/8/9 开头的四位数车次列车:普通旅客列车,简称普客,曾经有直通(跨铁路局运营)的普客列车,但是现在已经没有了。这样的列车一般经停所有能停的车站,部分普客列车虽然站距较远,但是也明显多于普快列车。这类列车的车底主要是 22 型绿皮车,其他车底均少见,根本没有空调车。这类列车运营里程一般不长,一般属于短途多站的列车。

L 开头的列车:临时旅客列车,简称临客,字母 L 是“临”字汉语拼音的简写。这类列车只在需要的时候才运营的,车种也是最杂的列车类别,L 系列中有少部分列车相当于快速,大多的相当于普快,也有的相当于普客。车底编制也是杂乱,L 系列列车一般没有自己专用的车底,通常是随便拉来些车底就编组。

A 开头的列车:按需临时旅客列车,好象没什么简称,如果有,那就叫按需临客吧。字母 A 是“按”字汉语拼音的简写。这类列车比 L 系列的地位更低,但是大多特征与 L 系列的相似。同样 A 系列的列车在<全国铁路旅客列车时刻表>上是查不到的。

Z 开头的列车:直达特别快速旅客列车,简称直 特,字母 Z 是“直”字的汉语拼音简写。这样的列车在行程中一站不停或者经停必须站但不办理客运业务,这类列车的车底都是 25T,全部都是空调列车。所有的 直特列车都是跨局(不是在一个铁路局内)运营列车。

1. 2. 2 司机代号

在实际运营中,列车司机要对列车运行路线非常熟悉,只有这样才能保证在列车运行过程中可以正确处理随时遇到的问题,这一点和汽车司机不太一样,所以列车司机通常是一直负责某条路线,也就是司机通常在一条列车上工作,这样以来,就有必要对司机编号,一边对司机进行管理。

司机代号是一位司机师父所唯一对应的,我们对这些信息进行存储,在调度管理时,可以对其进行查询,这样查询必查询司机名字方便的多,而且这样一来,

还避免了重名的司机无法辨认的麻烦，所有在实际应用中通常对司机进行编号。

1. 2. 3 最高速度

列车速度时我们都非常关心的一个问题，其实，列车时速不但但时旅客朋友们关心，铁路部门更是关心，旅客朋友们关心速度是关心旅行时间，而我们铁路运营部门关心速度则是关心安全问题，因为速度是与安全问题挂钩最紧的一个参数，所有这个参数对于在车库中管理和维护列车运行的工作人员就更是显得尤为重要了。

最新的列车速度调查表明：法国高速列车 3 日在行驶试验中达到 574.8 公里的时速，打破了 1990 年由法国高速列车创下的时速 515.3 公里的有轨铁路行驶世界纪录^[2]。

也正因为如此，在列车信息采集的设计中自然少不了这个参数的设计，尤其是最大速度，这个参数几乎是所有采集列车信息的系统都不可回避的一个问题。

1. 2. 4 最高轴温

轴温可分为机车轴箱轴承、牵引电机轴承、抱轴承（空心轮对轴承）以及传动轴承的温度。我们可以用机车轴温监测仪获取这方面的信息，机车轴温监测仪能在机车运行中，自动监测各轴承传递到轴箱、电机轴承端盖以及电机抱轴承罩的温度。当轴温超过设定报警温度时，能及时准确的发出声、光报警信号，并跟踪记录。正常情况下，轴温的温度在 20~25℃ 的范围内。

铁路客车集中式轴温报警器是由传输线路、轴温传感器及控制显示器组成，是监测铁路客车轴温，预报热轴，防止切轴，保证旅客列车运行安全的重要设备。轴温报警器使用以来，在预报客车热轴，防止切轴方面发挥了重要作用，准确预报了大量热轴故障。因为调度所里边有轴温预警装置，列车运行过程中，每个列车轴温多少度？哪个轴温高了，如何可能发生摩擦、燃油了，都在调度室马上能反应出来。旅客列车运行中自动监测客车轴温变化的装置，是防止客车热轴引起的切轴事故，确保行车安全的科学手段。

随着列车的提速，轴温检测的作用就愈加明显了，否则列车事故必然会随着速度的增加而增加。在现实机车运行中，这个参数也必须及时的送回车库，使得车库的管理维护人员了解机车的状况。这时列车安全运行的重要保证。

1. 2. 5 耗电量

众所周知，列车上是要耗电的，旅客列车上除了照明，还有很多用电的设备。尤其是近几年研制的新型快速列车、准高速列车更是增加了空调装置、车门集中遥控、粪便集存密封处理、燃油及电热两用取暖装置等电气设备，最大用电量高达 400kW。耗电量这个参数已日益成为一个非常重要的参数，当然也就有了传给车库总部的需要。

那么，这些电是从那里来的呢？原来，在客车上有个小小的发电装置，就是客车的底架或转向架上吊挂的发电机。发电机通过皮带与安装在车轴上的皮带轮连接，当列车运行时，车轮滚动，由皮带带动发电机转动而发电，供应车上各种电器具使用。当列车停靠时，车轮不动了，发电机就不再发电。为了保证客车无论开停都能正常供电，就在客车上装备蓄电池供电系统。开车时，它把发电机发出的电能转变成化学能储蓄起来，这个过程叫蓄电池充电；停车时，它能把化学能转变成电能，向各用电器具送电，这个过程叫蓄电池放电。铁路上把安装有发电机和蓄电池的车厢称为“母车”，没有安装的车厢称为“子车”，母车与子车的比例一般为 1: 1^[3]。这种靠车轮转动，通过皮带带动发电机发电的供电方式，称为车轴发电机式供电。中国铁路列车广泛采用这种供电方式。这种供电方式的发电装置，运用数量最多的是 J 型三相交流感应子发电机，其发电量只有 5kW，显然不适于用电量几百倍的新型空调客车。

新型空调客车采取的供电方式是集中式供电，就是在列车中的某一节车厢内设置发电站或在列车上设立变电站，向整个列车供电。供电方式主要有两种：一种是在专门的发电车（图 3. 5. 5_02KD 型发电车）或行李发电车内，安装柴油发电机组，构成列车发电站。列车发电站的工作由专门的配电盘控制。发电站发出的电，通过贯穿全列车的输电干线和专门的车端连接器，送到列车各节车厢。另一种是在电气化铁道的列车牵引区段，电力机车升起受电弓，将接触网供给的 25kV，50Hz 的单相交流电引入列车变电站。然后，经过列车变电站中的变压器、整流器、变流机等电器设备变换后，给整个列车供电。这种供电系统，配线经济、不用蓄电池、车辆构造成本较低、发电量不受列车速度的影响。但是，它也有缺点，一旦发电站、变电站出现故障或发电车从列车上摘挂下来，就会影响列车供电。为了弥补这种缺陷，列车可以同时装备轴驱式供电装置作为备用，以保证列车用电。

1. 2. 6 故障

在诸多交通工具中，火车算是一种非常稳定的工具，但是这只是在乘客严重

看到的，其实在列车运行时会出现这样那样的问题，只是列车的处理故障工作做的非常好，即使产生了故障，也能即使得到排除，而不影响乘客的旅行。

但是这并不是意味着可以疏忽大意，相反，为了保持这种高稳定性，我们必须进行更加周密的设计，其中之一就是将列车检验出来的故障即使的报给车库，以便进行及时的处理。

列车车载微机在司机台显示屏幕上用汉字显示各种机车运行数据及设备的工作状态和数据。当机车设备出现异常和故障时，车载微机装置将根据机车设备运行中发生的异常和故障表现，在车载屏显示幕上显示故障种类、故障数据、故障发生原因的判断及处理措施。故障发生时，在屏幕显示的同时，还将发出报警声提示。

车载装置按状态变化和定时两种方式记录机车运行数据和设备工作数据。机车设备发生故障、故障自动消失、故障由司机确认时，将记录故障发生时间（故障自动消失时间、故障由司机确认时间）、公里标、故障类型、故障期间典型数据等参数。

信息传给车库计算机后，车库计算机根据车载微机记录的数据，进行分析判断，建立各机车设备的运行档案数据库，打印各种报表，为机车设备实现状态修提供可靠数据。为机务管理人员了解各机车运行情况和设备状态提供帮助。

1.3 国内外发展现状及本设计的定位

1.3.1 国内外发展现状

高速铁路的信号与控制系统，是高速列车安全、高密度运行的基本保证。因此，世界各国发展高速铁路，都十分重视行车安全及其相关支持系统的研究和开发。高速铁路的信号与控制系统是集微机控制与数据传输于一体的综合控制与管理系统，是当代铁路适应高速运营、控制与管理而采用的最新综合性高技术，一般通称为先进列车控制系统（Advanced Train Control Systems）。如北美的先进列车控制系统（ATCS）和先进铁路电子系统（ARES），欧洲列车控制系统（ETCS），法国的实时追踪自动化系统（ASTREE），日本的计算机和无线列车控制系统（CARAT），等等。

先进列车控制系统是铁路在技术上的一次突破，它将使铁路和整个国民经济取得巨大的经济效益。

从80年代初开始研究的先进列车控制系统，现仍处于研究、试验与完善之中。近年来，许多国家为先进列车控制系统研制了多种基础技术设备，如列车自动防护系统、卫星定位系统、车载智能控制系统、列车调度决策支持系统、分散式微

机联锁安全系统、列车微机自动监测与诊断系统等。世界上许多国家如美国、加拿大、日本和西欧各国都将在 20 世纪末到 21 世纪初,逐步推广应用这些新技术。目前一些国家已经开始分层次的实施。

1. 3. 2 本设计的功能及定位

本设计主要是从信息采集方面进行深入研究,并借以大幅度提高车辆调度效率,为整个铁路系统的效率提高提供保障。

本设计实现了这样的功能:

- 1: 列车即将进站入库时,由列车端主机通过桑锐 SRWF-501 芯片无线发送前面介绍的列车信号。
- 2: 信号以微波形式进行无线传输,传至车库主机。
- 3: 车库主机利用桑锐 SRWF-501 芯片进行自动信号接收,并显示列车信息,然后将这些信息存入数据库。

本设计信号流程图如下:

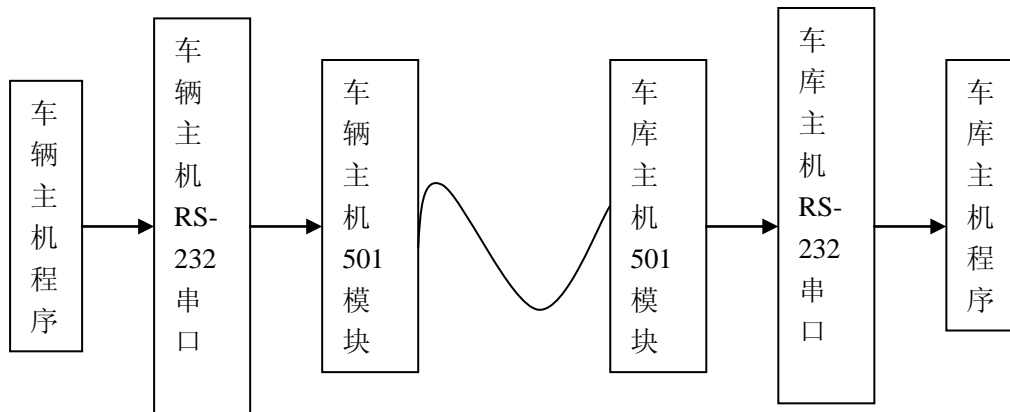


图 1.1 本设计的信号流程图

其最大特点就是使用上海桑锐 SRWF-501 无线模块实现无线传输数据,因而在列车还没进站就进行数据采集,从而节约了大量时间,这些时间在日益繁忙的铁路系统中显得非常宝贵,因此,本次设计投入使用以后,必将为铁路部门列车调度带来实质性的进展,有力的支持铁路部门营运效率的提高。

1. 4 论文研究内容

本设计主要有三个设计模块,首先是上海桑锐 SRWF-501 无线模块,这一模块主要是完成无线传输信息的功能,它并不管具体在其上面传送的是什么东西,

什么内容，它只管传送，并进行简单的差错控制，为整个设计提供底层支持；其次是 RS-232 串口，这是一个大家都非常熟悉的串行通信接口，应用起来也非常方便，对其编程也非常方便，很多语言都提供了关于控制 RS-232 串口通信的方法，这里我是用了 VB 环境；最后一个模块是数据的存储和提取，常规来讲，可以应用文件或数据库两种方法，但由于在车库终端需要对已存的列车信息数据进行查询，例如按车号进行查询等，应用文件很难实现，而如果用数据库来实现就容易多了，所以本次设计的最后一个模块是数据库。

1. 4. 1 SRWF-501 型模块

SRWF-501 型无线通信模块是上海桑锐电子科技有限公司采用高效 FEC 前向纠错技术结合高性能的无线射频 IC，以及高速微处理器相结合开发出的一款无线通信模块，可与 SRWF-505、SRWF-506、SRWF-507、SRWF-508、进行无线通信。该无线通信模块具有很强的抗干扰能力，全透明传输，体积小，功耗低传输距离远的特点，客户使用时不需要任何编码技术。

1. 4. 2 关于 RS-232 串口^[4]

以前 RS-232 广泛用于数字终端设备如计算机与调制解调器之间的接口，以实现通过电话线路进行远距离通信，现在这方面的应用已经不如以前广泛，但是 RS-232 串口通信却没有因此退出历史舞台，而是因其方便而为很多串行通信程序员所热衷。RS-232 使用 25 个引脚，但在绝大多数情况下，仅使用其 9 个引脚，所以我们常称其为 25 口或者 9 口，在台式计算机后面通常含有两个 9 口的 RS-232 串口，分别称为 COM1 口和 COM2 口，笔记本电脑上的应用就不如台式机广泛了。应该注意的是所有的串行通信接口都是公头的，并行接口都是母头的，所有台式机后面通常还有一个 25 针的母头的接口，那是 LPT 并口，通常用来接并口打印机，切不可把它认为是 RS-232 串口。

1. 4. 3 ACCESS 数据库

Access 是微软公司推出的基于 Windows 的桌面关系数据库管理系统 (RDBMS)，是 Office 系列应用软件之一。它提供了表、查询、窗体、报表、页、宏、模块 7 种用来建立数据库系统的对象；提供了多种向导、生成器、模板，把数据存储、数据查询、界面设计、报表生成等操作规范化；为建立功能完善的数据库管理系统提供了方便，也使得普通用户不必编写代码，就可以完成大部分数

据管理的任务。

1.5 论文组织结构

本论文详细阐述了本次毕业设计的过程和实现的功能,构建了一个实现机车进站入库时进行机车信息无线采集的整体模型,即可以实现机车上的主机向外发送机车信息,通过上海桑锐 SRWF-501 模块进行无线传输,然后车库进行无线接收,最后还可以在车库进行入库车辆的信息查询。

该论文有以下五章组成:

第一章为绪论,主要论述了论文的研究背景和意义,分析了目前机车入库信息采集在国内外的发展现状,并提出了本文的研究内容和论文的组织结构。

第二章着重介绍了关于上海桑锐 SRWF-501 模块的有关问题,模块的使用,模块的主要接口,模块的电器特性,以及模块的基本控制等。

第三章则着重介绍了基于 RS-232 串口编程,串行通信的特点,与并行通信的不同,串行通信的种类,串行通信的控制方式,以及 RS-232 串行通信接口的电气特性,接口原理,传输控制等。

第四章着重介绍了有关数据存储以及提取方面的内容,结合了用文件实现和用数据库实现的不同,及它们的优缺点,因为本次设计首先我是用文件实现数据存储和提取的,然后又改为用数据库,所以对其二者的比较印象非常深。

第五章总结了论文所做的工作,对程序所实现的结果与预计目的进行了比较,得出不足之处,用以在以后的工作中对这些不足进行改进。

第二章 基于上海桑锐 SRWF-501 无线模块的设计

该设计的无线部分是由上海桑锐 SRWF-501 模块来承担的，并且进行底层的简单传输控制。在列车上和车库里各有一个 SRWF-501 模块，分别担负着信息无线发送和信息无线接收的任务。它们分别跟列车上和车库里的计算机通过 RS-232 串口相连，有了它们去完成无线传输任务，我们就只需要和 RS-232 串口进行联络就可以了。

2. 1 SRWF-501 无线模块环境介绍

上海桑锐 SRWF-501 模块在程序中负责的是无线传输的任务^[4]，他可以从 RS-232 中获得数据，也可以向 RS-232 传送数据，我们需要人为干预的是正确的连接 RS-232 与 SRWF-501 模块的接线，这样 RS-232 串口一旦有了数据就会传给 SRWF-501 模块，而 SRWF-501 模块一旦接收到了数据，也会马上传给 RS-232 串口。

2. 1. 1 SRWF-501 无线模块结构

SRWF-501 无线模块可以分为三个部分：接口部分，集成电路部分，天线部分。其中接口部分提供 RS-232 的接口，进行与计算机的数据传输工作；集成电路部分负责对计算机传来的数据进行处理然后送往天线发射的功能；天线部分要把集成电路提供的信息数据发射出去，或者接收无线数据传给集成电路，为了保持天线部分的灵敏，不用的时候需要将其拿下来，这样有利于保证数据正确传输。

2. 1. 2 SRWF-501 无线模块集成环境^[5]

型号 SRWF-501

中心频率 433MHZ/868MHZ/915MHZ

频率范围 429~438MHZ

信道数 8/16/32 信道可选

调制方式 GFSK

通信距离(开放环境) >800 米

功能 双向半双工通信

波特率 1200/2400/4800/9600/19200bps/38400bps/76800bps

接口方式 TTL 串口/RS-232/RS485

发射功率 +10mW/10dbm

接收灵敏度 -118dbm

通信数据格式 8N1/8E1/8O1 可调

工作电压 +3.0V~+5.0V

2. 1. 3 SRWF-501 无线模块图片



图 2.1 无线传输芯片 SRWF-501



图 2.2 天线

2. 1. 4 SRWF-501-50 型无线模块接口的定义

SRWF-501-50 提供 1 个 9 针的连接器 (CON1), 其定义基于终端的连接方法见下表:

表 2.1 引脚定义

序号	SRWF-501 端	说明	电平	连接到终端
1	GND	电源地		电源地
2	VCC	电源 DC	+3.6~5.0V	
3	RXD/TTL	串行数据接收	TTL	TXD
4	TXD/TTL	串行数据发	TTL	RXD
5	SGND	信号地		模拟地
6	A (TX)	TX		A (RX)
7	B (RX)	RX		B (TX)
8	SLEEP	休眠控制	TTL	休眠信号
9	RESET	复位控制	TTL	复位信号

2. 1. 5 与 RS-232 接口连接

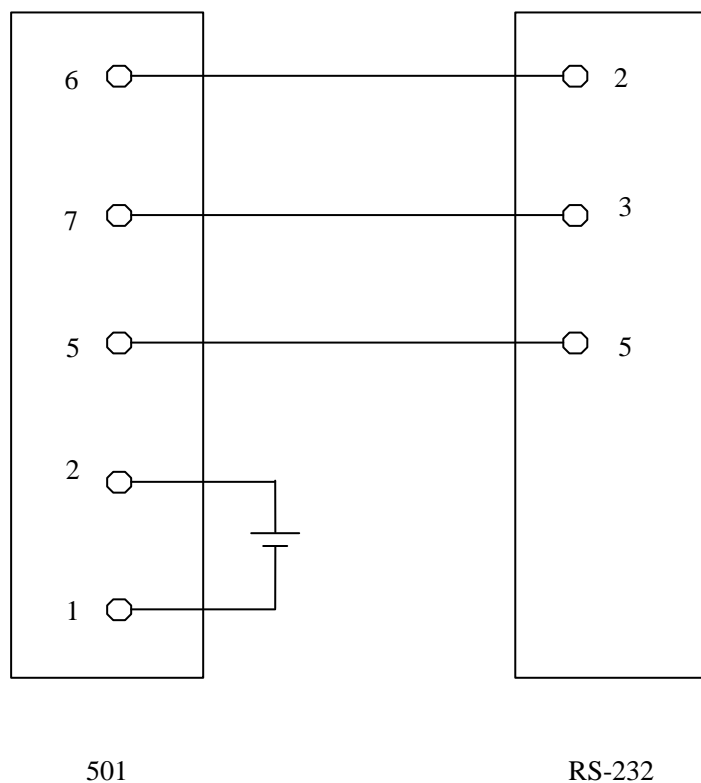


图 2.3 硬件连接

2. 2 本设计的硬件设计

2. 2. 1 本设计的硬件参数

本设计中所使用的上海桑锐 SRWF-501 芯片工作在 429~438MHZ 的频率范围内, 很明显, 属于微波范围。

由于本设计中涉及了检错重传的功能, 所有需要有确认等通信信号, 所以两块桑锐 501 模块都必须工作在半双工状态。

本设计综合了稳定性、高效性、高可靠性等因素, 最后选择了 9600 波特的波特率。

本设计与 RS-232 串口相连, 采用+10mW/10dbm 功率发射, 工作电压+4. 5V。

2. 2. 2 开发环境

本设计使用 VB 开发环境, 将数据通过 RS-232 串口将数据传送至 501 模块, 然后由 501 模块将数据发射出去, 接收端首先有 501 模块进行数据接收, 然后传给 RS-232 串口, 再有 VB 开发环境由程序将数据写入数据库进行存储, 以待日后查询使用。

2. 2. 3 硬件连接

连接图见图 2. 3 所示, 芯片本身并没有提供与 RS-232 的接口, 故需要我们自己设计接口电路, 将芯片连接到 RS-232 串口上。

2. 2. 4 硬件设计过程

1: 电源部分设计: 本设计中的无线传输芯片桑锐 SRWF-501 模块电源电压范围为 3~5V, 为方便起见, 我才用了 4. 5V 的电源, 由三节 1. 5V 的干电池组成。

2: 接口部分设计: 本设计中使用的上海桑锐 SRWF-501 模块为了适用于其它接口, 没有直接提供与 RS-232 串口的连接口, 所以需要自己设计, 将上海桑锐 SRWF-501 模块的引脚与 RS-232 的引脚按图 2. 1 用导线进行焊接即可。

3: 501 模块测试: 将硬件按图 2. 1 连接, 编写最简单的检验程序如下:

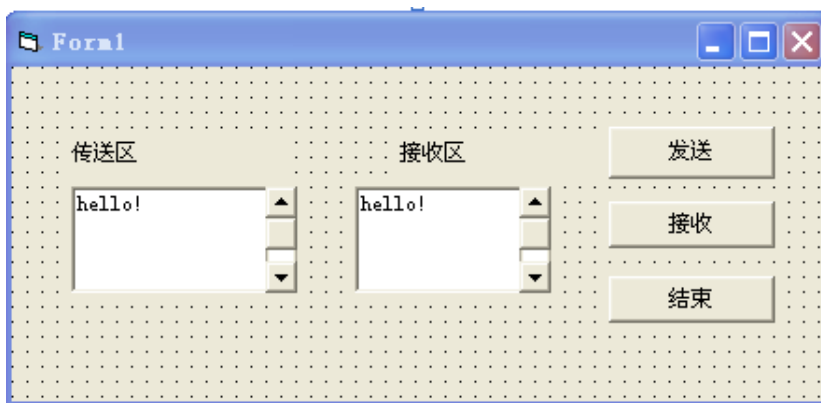


图 2.4 测试程序界面

测试程序:

```
Private Sub command1_Click()  
Comm1.Output = Text1.Text  
End Sub
```

```
Private Sub Command2_Click()  
Text2.Text = Comm1.Input  
End Sub
```

```
Private Sub Command3_Click()  
Comm1.PortOpen = False  
End Sub
```

```
Private Sub Form_Load()  
Comm1.PortOpen = True  
End Sub
```

该测试程序完成了最简单的数据传输，在输入端输入一个字符串，在接收端就可以收到这串字符串，这可以表明芯片的工作是完全正常，可以进行使用了。

4: 501 模块调试: 使用过程中多次出现数据发送不正常的情况, 经过检查, 大部分情况是因为供电系统不稳定造成的, 这又包括电压不足和接触不良两种情况。另外, 传输的稳定性和所处的环境也有很大关系。

2.3 小结

本小节给出了有关上海桑锐 SRWF-501 模块的产品简介, 运行环境, 使用方法以及在程序中的作用等内容, 通过该章节的论述, 明确了整个设计过程中无线传输部分任务的完成, 也明确了怎样将无线设备与计算机终端相连, 完成了整个设计中非常重要的一部分。

第三章 基于 RS-232 串口进行的通信

随着信息技术的发展以及计算机网络的广泛使用,计算机通信技术越来越成熟。在众多通信实现方案中,串行通信技术作为一种灵活、方便、可靠的通信手段,有着广泛的应用。它是计算机应用中一种常用的通信方式,在计算机控制系统中的应用是非常广泛和重要的。此外,在工业控制领域,微机与微机、微机与外设、微机与分布式下位机等之间都可以通过 RS-232C 串行端口互连通信,以实现控制和传输数据等目的。VB 提供了完备的程序接口函数,使得在 Windows 环境下通过中断方式实现串行通信变得很简单,它具有界面开放、易于使用、易于理解等优点,因此受到众多用户的欢迎。

3.1 串行通信

串行通信由于接线少、成本低,在数据通信、计算机网络以及分布式工业控制系统中,经常采用它来交换数据和信息。1969 年,美国电子工业协会(EIA)公布的 RS-232C 作为串行通信接口的电气标准,它定义了数据终端设备(DTE)和数据通信设备(DCE)间按位串行传输的接口信息,合理安排了接口的电气信号和机械要求,在世界范围内得到了广泛的应用。

3.1.1 串行通信与并行通信^[6]

微型计算机主机与外部设备的连接,基本上使用了两类接口:串行接口与并行接口。并行接口是指数据的各位同时进行传送,其特点是传输速度快,但当传输距离较远、位数又多时,导致了通信线路复杂且成本提高。串行通信是指数据一位位地顺序传送,其特点是通信线路简单,只要一对传输线就可以实现双向通信,并可以利用电话线,从而大大降低了成本,特别适用于远距离通信,但传送速度较慢。串行通信本身又分为异步通信与同步通信两种。串行通信线路上传送的是数字信号,表示传送数字信号能力的指标为数据速率(Data Rate),其单位为 bps(bit persecond),即每秒钟传送的二进制位数。采用并行传送方式在微型计算机与外部设备之间进行数据传送的接口叫并行接口,它有 2 个主要特点:一是同时并行传送的二进制数就是数据宽度;二是在计算机与外设之间采用应答式的联络信号来协调双方的数据传送操作,这种联络信号又称为握手信号。

3. 1. 2 串行通信接口

目前普遍采用的一种串行接口标准是 RS-232-C 标准。RS-232-C 接口标准采用 25 个引脚的连接器 (D 型插座)。RS-232-C 规定有 25 根连线。

3. 1. 3 串行通信实现方式^[7]

1: 异步通信 (Asynchronous Data Communication) 以字符为传输单位, 通信中两个字符间的时间间隔是不固定的, 然而在同一字符中的两个相邻位代码间的时间间隔是固定的。传送一个字符的信息规定有起始位、数据位、奇偶校验位、停止位等。

2: 同步通信 (Synchronous Data Communication) 以帧为传输单位, 每个帧中包含有多个字符。在通信过程中, 每个字符间的时间间隔是相等的, 而且每个字符中各相邻位代码间的时间间隔也是固定的。

3. 2 RS-232 串行通信接口

3. 2. 1 RS-232 在计算机的位置

一般的计算机, 尤其是台式计算机, 都会有一到两个 RS-232 串行通信接口, 分别成为 COM1 口和 COM2 口^[8], 它们一般为蓝颜色, 9 针的公口 (为 25 针的简化), 特别要注意的是, 一般机器后面还会由一个 25 针的母口, 一般为红颜色, 这个是 LPT 并口, 通常用来接并口打印机的, 不要把它错认为是 25 针的串口了。

3. 2. 2 RS-232 针脚图

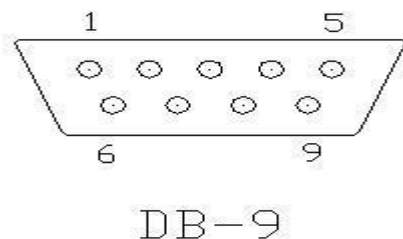


图 2.4 9 针 RS-232

3. 2. 3 RS-232 引脚介绍

表 3.1 RS-232C 的引脚

引脚号	名称	引脚号	名称
1	载波检测 CD	2	接收数据 RxD
3	发送数据 TxD	4	数据终端准备好 DTS
5	信号地 GND	6	数据装置准备好 DSR
7	发送器时钟 TxC	8	清除发送 CTS
9	振铃提示 RI		

3. 2. 4 RS-232 引脚连接

见图 2.3。

3. 3 利用 RS-232 进行通信

前面一章提到，当我们把上海桑锐 501 模块与 RS-232 串口相连后就只需对 RS-232 串口编程^[9]就可以了，当我们把数据传递到 RS-232 串口后，经过上海桑锐 501 模块就可以进行无线传输了，所以我们现在要研究的就是怎样将数据传递到 RS-232 上。

而要通过 RS-232 串口传输数据，就不得不提到 Microsoft 在 VB 中提供的一个控件：MSComm，这个控件可以方便的对 RS-232 串口进行控制，这样正好满足我们对串行通信的需要，下面介绍这个控件。

3. 3. 1 MSComm 控件简介^[10]

MSComm 控件首先提供了一个与 RS-232 通信的通道，建立了这个通道之后，就可以进行通信了，这就像是高速公路，只有先将高速公路先铺设好，然后在放上汽车，就可以进行正常的通信了。RS-232 串口通信与这个过程非常类似，用 MSComm 控件的打开命令（MSComm.open）打开与 RS-232 的连接后，可以用 MSComm 的其他属性进行通信编程，就可以实现与 RS-232 串口的通信了，具体的属性及其控制将在 MSComm 的属性一节详细讲解。

3.3.2 MSComm 控件的引用

当我们应用 VB 开始一个项目时, Visual Basic 的工具栏中会有许多默认的控件供程序设计者使用, 这些原本就出现在工具栏中的控件是内置控件, 他提供一些基本的程序控制组件给程序设计者使用, 但是, 一些比较特殊的控件不会出现在工具栏中, 这里提到的 MSComm 控件就不默认在工具栏中, 所以我们需要自己添加, 它存在于“Microsoft Comm Control 6.0”组建中, 选择即可。

3.3.3 MSComm 的工作方式

MSComm 控件的目的是为了让用户设计一个系统可以和串口进行通信, 因此信息会在硬件电路上流动, 此控件提供了两种方式来处理这种流动: 一种是事件驱动, 这种方法是一种有效的处理事件的方法。这种方式中, 只要有通信时间发生, 不管是错误还是由新的信息发来, 都会出发事件, 程序会得知, 根据程序的设计可以对其作出反应。

另一种方式是通过检查 ComEvent 属性的值来轮询时间和错误。如果应用程序不大, 这种方法会比较好。

3.3.4 MSComm 的操控原则

Windows 的操作系统与 DOS 是完全不同的, Windows 采用的是全新的对象化观念, 把所有的程序都对象化, 在设计 VB 串行通信的相关时, 要遵循下四个四个步骤: 1: 对象, 2: 属性, 3: 事件, 4: 方法。Visual Basic 的串行通信对象乃是将对 RS-232 的初级操作予以封装, 用户以高级语言的语法就可以实现利用 RS-232 与外界进行通信, 并不需要了解其它初级操作。

3.3.5 MSComm 控件属性介绍

Comport: 设计和返回通信连接端口代号。程序必须指定所使用的串行端口的号码, Windows 会使用所指定的端口与外界进行通信, 程序也可以此属性返回所使用的串行口的号码。

Settings: 初始化参数。以字符串的形式设计或返回连接速度、校验码、数据位、停止位等四个参数。其格式为: “BBBB, P, D, S”, 其中 BBBB 为连接速度, P 为字符校验方式, D 为数据位数, S 为停止位数。

PortOpen: 设置或返回连接口的连接状态。使用串行口前必须先将它打开,

使用完毕后必须将它关闭。

Input: 从输入寄存器返回并删除字符。这是靠着这个属性,VB 可以经 RS-232 串口中的数据出入到程序中来,这是我们进行 RS-232 串口通信设计中最基本的一条属性。

Output: 将数据发送到 RS-232 串口,这个属性正好和上一个属性的功能相反,靠着这个属性,我们可以将想输出去的数据传输给外界。

InputLen: 指定读入字符的长度或字符数。

HandShaking: 指定双方的通信交握协议。所谓“交握协议”是指通常意义上所称的流量控制。

CommEvent: 只有有事件发生或出现错误就会产生 OnComm 事件,CommEvent 属性存有该错误或事件的数值码,这个属性也会返回值供程序员读取,这样可以确定是什么错误或事件。

InputMode: 设计或返回属性取回的数据的类型。

3.3.6 MSComm 控件属性的应用

按上述连接方法将 RS-232 串口的引脚的硬件连接好,然后应用上面所述控件就可以实现通信了。具体步骤如下:

1: 打开串口:

```
MSComm. PortOpen=True
```

2: 参数设置(包括传输速率,差错校验,码位控制等)

```
MSComm. settings= "9600, e, 8, 1"
```

3: 数据传给串口:

```
MSComm. Output=DataOutPut (DataOutPut 为含有输出数据的变量)
```

4: 串口的数据输入到程序

```
DataInPut=MSComm. InPut (DataInPut 为接受输入数据的变量)
```

5: 闭串口:

```
MSComm. PortOpen=False
```

更深一步的属性设置详见程序部分,这里只是简单的列出了最基础的实现对 RS-232 串口传输的部分。

3.4 本设计中的 RS-232 串口应用

本设计中 RS-232 串口充当了程序与上海桑锐 SRWF-501 模块的桥梁,数据先

到的串口，然后在传出去，它与 501 模块的硬件连接如图 2.3。

1: 串口参数设置:

本设计中，串口的码速率设置为 9600 波特，采用奇偶校验方式，每次传输 8 位，停止位一位。

2: 开发环境:

本设计中使用 VB 开发环境，正如前一小节中所介绍的，应用 VB 中的 MSCOMM 控件进行设计，这个控件可以控制整个过程的收、发、参数更改等过程，具体的实现如下:

发送:

```
MSComm1.Output = strinsertsql 'strinsertsql 为发送的字符串
```

接收:

```
strinsertsql = MSComm1.Input 'strinsertsql 为发送的字符串
```

参数设置:

```
MSComm1.settings= "9600, e, 8, 1"
```

3: 硬件连接:

请见图 2.3。

4: 硬件调试:

1: 硬件打开: 有很多机器，为了减少自检时间，将串口关闭，我们试验时必须将串口打开，更改方法是在主机的 BIOS 中将 InBoard Serial 的值设置为 True。

2: 硬件测试: 将 RS-232 串口的接收端引脚（2 号角）和发送端引脚（3 号引脚）相连，即让机器自发自收，这样可以检测串口的好坏，串口的有效是本次设计的前提。

3.5 小结

RS-232 串口通信这一章详细讲述了有关串口通信以及最长用的串口 RS-232 的有关问题，依次讲述了 Windows 为 VB 所提供的串口编程接口——MSComm 控件。

MSComm 控件的引用，MSComm 控件的使用方法，MSComm 控件的常用属性等，以及用这个控件完成最基本的数据通信方法等。

第四章 基于 ACCESS 数据库的数据存储与提取

数据对一个完整的程序的作用不言而喻，没有一个强大的数据处理作为支持，就不可能有一个性能优良的软件。那么通常数据都是怎么处理的呢？有那些处理方式呢？怎样才能保证为程序提供最高可靠性的数据保障呢？这一章将结合本次设计阐述这些问题。

4.1 数据的存储与提取

习惯了用面向过程的语言编写一些小程序的朋友可能会觉得数据和程序完全可以混在一起，只要在程序运行时将数据输入就可以了，顶多在数据比较多的时候，输入会比较麻烦，但也只需用一个文件就足解决问题了。但是对于经常用面向对象的编程的朋友可能有不同的体会，因为面向对象的编程已经不是模拟一个简单的过程，而是建立了很多对象，等着用户输入事件，然后对其反应，这时数据已经是一个非常复杂的部分，他要包含提取，处理，查询等功能，而且要求安全性能非常高，这就不得不引入数据库。

4.1.1 文件

文件是最简单的数据存储和提取方式，当程序的数据非常大，人工输入起来会非常麻烦，这时我们可以考虑到文件方式，几乎每一种语言都有对文件访问的命令，我们可以用这些命令进行非常简单的文件访问，免去复杂的操作，同时，也可以通过程序对文件进行写数据，这样又避免了复杂生成数据的过程，熟练的应用文件存储工具，可以为我们提供很大的便利。

4.1.2 数据库

数据库也是一种数据的存储和提取的方式，现在应用非常广，可谓已经到了家喻户晓的地步了，最简单的数据库是由 Microsoft 提供的 ACCESS 数据库，本程序就是应用这个数据库进行数据存储和提取，以及查询的。数据可以提供稳定性非常高的数据存储，越来越受到大家的青睐。

4. 1. 3 数据库与文件的对比

数据库和文件是两种主要的数据存储和提取方式，它们各具优点和缺陷。以文件存储和提取数据的最大好处是简单，方便，容易上手，只需了解几条命令就可轻易上手。相对文件来说，数据库的优势就在于其强大的性能，例如其建表，查询等功能都是文件所没有的。

4. 2 本程序中 ACCESS 数据库的应用

上一小节介绍了文件和数据库在存储和提取数据时的各自优势，由于功能上的需要，在此次程序设计中，只能用数据库，这样才可以实现数据随时输入列表，进行查询处理，输出等操作。

4. 2. 1 程序中数据库的主要功能

在程序中建立了一个 Microsoft Access 数据库，在其中建立了一张列表，包含了列车的信息，然后向其中添加多次列车的信息，之后进行查询工作，可以按照不同方式查询，并同时将其读取，最后将数据显示出来。

4. 2. 2 程序中数据库的设计与实现

上述功能将在程序中依次实现，具体方法如下：

1: 数据库的建立

这一步可以不与 VB 开发环境直接关联^[1]，只需在预定的路径出建立一个新的数据库即可。

2: 连接至数据库

对数据库的操作之前必须先建立与数据库的联接，这也非常像高速公路，必须先铺设好路线，然后才能谈到运输，建立与数据库的连接非常简单，使用数据库的连接对象 ADO.Connect，具体语法如下：

1: 实例化一个连接对象：

```
Public g_dbcon As New ADODB.Connection
```

2: 用连接对象属性设置连接参数：

```
g_dbcon.ConnectionString =  
"Provider=Microsoft.Jet.OLEDB.4.0;_ & "Data  
Source=E:\graduate\emp.mdb;" & "Mode=ReadWrite"
```

```
g_dbcon.ConnectionTimeout = 30
```

3: 建立连接

```
g_dbcon.Open
```

3: 数据库的初始化

这一步就最好要在程序中完成了,当然在数据库中操作也可以,但是在开发环境中用 SQL 语言完成更为方便,关键字: Create, 具体语法如下:

```
g_dbcon.Execute ("create table ManageList (Mname char(32) ,MPWD_
CHAR(16))")
```

```
g_dbcon.Execute("create table EMPLIST (ENumber char(32)primary
key, Ename_ char(16), Eage int, Edate char(32), Eadress
char(128))")
```

4: 数据库的数据添加

数据添加到数据库是程序功能实现非常重要的一环,它使用了 SQL 关键字: Insert Into, 具体语句如下:

1: 整合要添加的语句:

```
strinsertsql= "INSERT INTO EMPLIST
(ENumber, Ename, Eage, Edate, Eadress) values ("
strinsertsql = strinsertsql & "" & Trim(Textcode.Text) & ", "
strinsertsql = strinsertsql & "" & Trim(Textname.Text) & ", "
strinsertsql = strinsertsql & Val(Textage.Text) & ", "
strinsertsql = strinsertsql & "" & Trim(Textdate.Text) & ", "
strinsertsql = strinsertsql & "" & Trim(Textaddress.Text) & ")")
```

2: 执行添加命令:

```
ExecuteSQL(strinsertsql)
```

5: 数据的查询

数据查询功能是要在火车站车库的主机上完成的功能,因为调度人员要完成调度就必须随时了解各个机车的情况,这个功能是通过使用 SQL 语句中的 Select 语句来实现的,具体语法如下:

1: 定义参数字符串:

```
strSQL = "select*from EMPLIST where (Eage>=" & Text1.Text & ") and
(Eage<=" & Text2.Text & ")")
```

2: 执行查询命令:

```
QueryEmpInfo(strSQL)
```

6: 数据的读取及显示

这个功能主要也是在火车站车库的主机上实现, 因为查询完了之后必须用效果非常好的显示软件显示出来才可以起到效果。本设计使用 VB 开发环境中的 MSFlexgrid 控件进行数据显示, 具体语法如下:

1: 建表

```
grdTBL.Cols = 7
grdTBL.Row = 0
grdTBL.Col = 0
grdTBL.Text = "Enamber"
grdTBL.Col = 1
grdTBL.Text = "Ename"
grdTBL.Col = 2
grdTBL.Text = "Eage"
grdTBL.Col = 3
grdTBL.Text = "Edate"
grdTBL.Col = 4
grdTBL.Text = "adress"
```

2: 输出数据:

```
Do While Not g_dbrct.EOF
grdTBL.AddItem (Empty)
grdTBL.Row = grdTBL.Rows - 2
grdTBL.Col = 0
grdTBL.Text = CStr(g_dbrct.Fields("ENumber").Value)
grdTBL.Col = 1
grdTBL.Text = CStr(g_dbrct.Fields("Ename").Value)
grdTBL.Col = 2
grdTBL.Text = CStr(g_dbrct.Fields("Eage").Value)
grdTBL.Col = 3
grdTBL.Text = CStr(g_dbrct.Fields("Edate").Value)
grdTBL.Col = 4
grdTBL.Text = CStr(g_dbrct.Fields("Eadress").Value)
g_dbrct.MoveNext
Loop
```

4.3 本设计中的数据库应用

上节介绍了数据库在程序中的主要功能,本结讲述数据库在发送端和接收端的各自功能。

4.3.1 发送端数据库

发端数据库,即列车上主机的数据库。主要完成数据的提取,将列车数据从数据库中提取,然后通过 RS-232 串口传送给上海桑锐 501 模块,进行无线传输。具体实现如下:

```
strSQL = "select*from Rail where (司机代号=1)"
```

```
If QueryEmpInfo(strSQL) = True Then
```

```
  'Unload Me
```

```
  frmRctlist.Show
```

```
End If
```

```
g_dbrct.MoveFirst
```

```
strinsertsql = "INSERT INTO Station (列车号,司机代号,司机姓名,均速,轴温,车  
管压力,车厢温度) values ("
```

```
strinsertsql = strinsertsql & "" & CStr(g_dbrct.Fields("列车号").Value) & ","
```

```
strinsertsql = strinsertsql & "" & CStr(g_dbrct.Fields("司机代号").Value) & ","
```

```
strinsertsql=strinsertsql & "" & CStr(g_dbrct.Fields("司机姓名").Value) & ","
```

```
strinsertsql = strinsertsql & "" & CStr(g_dbrct.Fields("均速").Value) & ","
```

```
strinsertsql = strinsertsql & "" & CStr(g_dbrct.Fields("轴温").Value) & ","
```

```
strinsertsql = strinsertsql & "" & CStr(g_dbrct.Fields("车管压力").Value) & ","
```

```
strinsertsql = strinsertsql & "" & CStr(g_dbrct.Fields("车厢温度").Value) & ")"
```

```
MSComm1.Output = strinsertsql
```

4.3.2 接收端数据库

接收端数据库,即火车站车库上主机的数据库,当它接收到数据时进行数据向数据库的添加,之后还要进行查询,显示,用以完成列车的调度。具体实现如下:(以下都是用子函数完成,具体代码请见附录代码部分)


```
strinsertsql = MSComm1.Input    ‘接收数据  
Call ExecuteSQL(strinsertsql)   ‘入库  
frmrcodelist.Show              ‘显示
```

4.4 小结

本章主要介绍了程序的数据管理与处理的部分，这部分首先比较了文件和数据库两种数据管理方式的不同点，介绍了各自的优势，进而说明了为什么选择了数据库而放弃了使用文件。然后又对程序对数据库的操作方法进行了简单的介绍，最后明确了接收端和发送端的各自不同的数据库操作，至此，数据库的使用就基本介绍完了。

第五章 功能的整合与程序的实现

5.1 功能的整合

前面几章分别介绍了各个模块，现在将前面几章的内容的功能整合起来。首先我们将上海桑锐 501 模块与主机的 RS-232 串口相连，也就是完成无线模块和串口部分的连接，然后在 VB 开发环境中用 MSComm 控件将程序与主机 RS-232 串口相连，这样三大模块就连接到一起了。

5.2 程序流程图

前面已经已经讲述了各个模块的作用，现在着重讨论具体程序的实现，下面是程序流程图^[12]：

列车端：

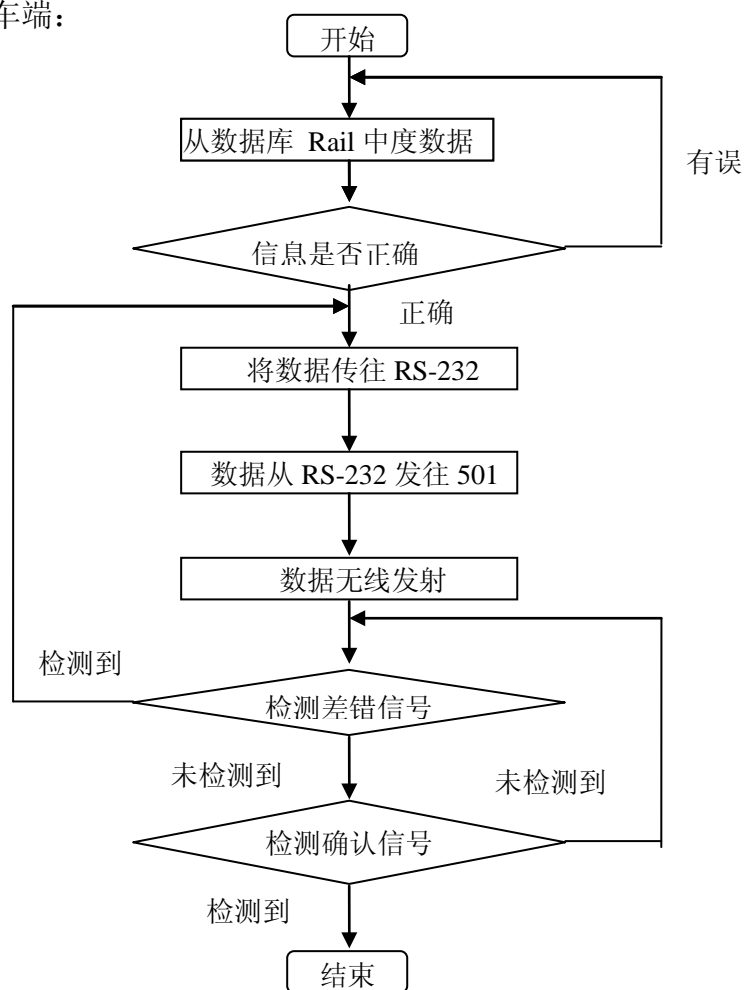


图 5.1 列车端程序流程图

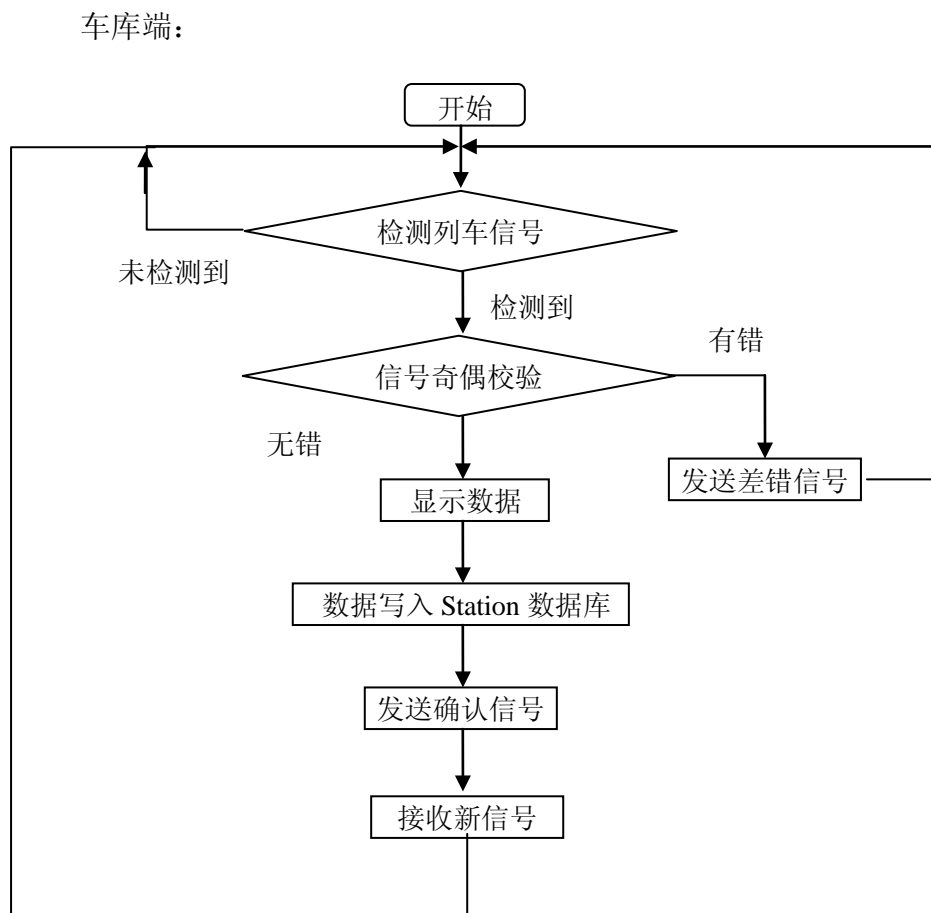


图 5.2 车库端程序流程图

5.3 程序界面

1: 发送端主界面:



图 5.3 列车端发送主界面

2: 发送端查看发送信息界面（点击主界面中的查看按钮即出现此界面）:

列车号	司机代号	司机姓名	最高速度	最高轴温	耗电量	故障
T124	1	时观	150千米每小时	30摄氏度	87%	无

记录: [Navigation icons] 2 [Navigation icons] * 共有记录数: 2

图 5.4 列车端查看界面

3: 车库接收端主界面（自动接收，故无接收按钮）:



图 5.5 车库接收端主界面

4: 车库接收显示界面:



图 5.6 车库端接收显示界面

5: 车库端查询显示界面:



图 5.7 车库端查询显示界面

5. 3 本设计主要模块的程序实现

以上是本设计两端的整体设计思路,下面介绍一下上述图中主要功能的具体代码实现:

1: 列车端发送—检错重传—确认的代码实现:

```

MSComm1.PortOpen = False           '打开串口
If ConnectToServer = False Then End '打开数据库

Line1:
MSComm1.Output = strinsertsql      '发送数据

Line2:
SuccessOrWrong = MSComm1.Input     '检测反馈信号
Text2.Text = SuccessOrWrong

If SuccessOrWrong = Empty Then     '继续检验
    GoTo Line2
End If

If SuccessOrWrong = ErrorToRepeat Then '有错重传
    
```

```

        GoTo Line1
    End If

    If SuccessOrWrong <> Success Then          ' 无反馈信号报错
        MsgBox "错误"
    End If

```

```

    MSComm1.PortOpen = False                ' 关闭串口
    If g_dbcon.State = adStateOpen Then     ' 关闭数据库
        g_dbcon.Close
    End If

```

2: 车库端接收-检错重传-确认过程的代码实现:

```

    If ConnectToServer = False Then End     ' 打开数据库
    MSComm1.PortOpen = True                ' 打开串口

```

```

line1:
BB = 1                                     ' 出错标记
strinsertsql = MSComm1.Input              ' 接收数据
If strinsertsql = Empty Then               ' 检测数据
    GoTo line1
End If
If BB = 2 Then                             ' 检测差错
    GoTo line1
End If

```

```

CC = "success"
MSComm1.Output = CC                       ' 发送确认信号

```

```

strSQL = "select*from Station where (司机代号>0)"

```

```

If ExecuteSQL(strinsertsql) = True Then    ' 信息入库
    ' MsgBox "员工信息已保存", vbInformation + vbOKOnly, "
    保存成功"
    ' Unload Me

```

```
End If

If QueryEmpInfo(strSQL) = True Then      ‘查询数据
    ’Unload Me
    frmrcclist1.Show
End If

GoTo line1                                ‘继续检测信号
```

3: 车库但查询功能的代码实现:

```
strSQL = "select*from Station where (司机代号>0)"

If Text1.Text <> Empty Then
    strSQL = strSQL & "and (列车号=" & Text1.Text & ")"
End If

‘按列车号查询

If Text2.Text <> Empty Then
    strSQL = strSQL & "and (司机代号=" & Text2.Text & ")"
End If

‘按司机代号查询

If Text3.Text <> Empty Then
    strSQL = strSQL & "and (司机姓名=" & Text3.Text & ")"
End If

‘按司机姓名查询

If QueryEmpInfo(strSQL) = True Then
    ’Unload Me
    frmrcclist1.Show
End If

‘开始查询
```

5.4 本设计的软件调试过程

众所周知，软件调试对于一个软件程序的设计过程来说是非常重要的。这次设计当然也不例外，我大致总结了软件调试过程如下：

1、程序中的符号：这部分错误主要是由于没有及时转换输入法造成的，例如将英文的等号写成了汉字的等号，造成程序无法识别，从而无法运行，这种错误有很难检测的出，所以一定要注意。

2、变量类型的区别：主要是字符串变量和整形变量，这两种变量在程序中一些语句中的语法格式是不一样的，例如查找字符型变量的语法与查找整型变量的语法就有所不同。

3、变量的作用范围：变量有不同的作用范围，如果不注意这一点就会造成变量无法正常使用，最终导致程序无法运行。

4、循环检测：在设置循环检测程序的时候，必须指定多长时间检测一次，时间可以设置的很短，但是不能没有，否则会造成死循环。

以上是此次设计中比较突出的几个问题，这些问题以及遇到的其它问题其实都可以用设置断点的方法解决，这样一步一步的排除，最终肯定能够排除错误。

第六章 结束语

经过这段时间的毕业设计，对用 vb 开发环境开发 RS-232 串口通信程序过程有了一定的了解，并且了解了上海桑锐 SRWF-501 无线模块，对无线通信有了深入了解，并完成了整个无线通信功能的实现。本次设计以 WINDOWS XP 作为开发环境，采用 VB 进行程序开发，用非常友好的界面实现了整个设计的预设功能。

6.1 结论总结

经过一个学期的努力，此次毕业设计终于实现了预计的功能，下面将所实现的功能列出如下：

- 1: 列车上浏览确认所发送的信息。
- 2: 列车主机将确认后的数据发往 RS-232 串口。
- 3: 数据从 RS-232 串口上经桑锐 501 模块无线发射出去。
- 4: 车库的桑锐 501 模块无线接收列车发来的数据，并作奇偶校验。
- 5: 桑锐 501 模块接收的数据传给 RS-232 串口。
- 6: 接收 RS-232 串口数据，显示，并存入数据库。
- 7: 在车库主机上可以查询所有相关记录。

6.2 未来展望

虽然经过了认真的设计，程序已经实现了预计的要求，但是就实际来说还有很大的改进空间，大致总结如下：

1: 供电部分，本实验中的无线部分桑锐 501 模块是要外接电源的，本次设计的电源是本人用三节 1.5V 电池连接而成，价格虽然便宜，但是有时候供电不稳，造成收发受阻的后果，供电部分是的一个要改进的地方。

2: 桑锐 501 无线模块虽然在实验中完成了预定的任务，但是会受环境影响，所以随着技术的发展，这一弊端将得到克服。

3: 本设计所传输的只是最基本的信息，要想在实际中得到广泛应用，则必须添加更多的数据参数。

致谢

在毕业设计完成之即，首先感谢指导老师彭春华老师。大学期间彭老师渊博的知识和优秀的教学方法给我留下了深刻的印象，使我受益匪浅。在毕业设计的一个多学期以来，彭老师不仅给我提供了很好的学习环境，还给予我很多的指导，在论文修改过程中，给我提出了许多宝贵的意见。彭老师严谨的治学态度、勇于开拓创新的胆识和忘我的工作精神，为我们树立了很好的榜样，尤其是彭老师积极进取的精神，让我深受启发。在此，对彭老师表示衷心的感谢！

最后，感谢所有的同学一个多学期以来对我的帮助，以及所有关心和帮助过我的朋友。

参考文献

- [1] 陈炳祥, 施雨农, 高速铁路信号与控制系统, file:///E:/graduate/高速铁路信号与控制系统(之一)%20---%20中华铁道网.htm 2007-3-22
- [2] <http://www.chinataiwan.org/web/webportal/W5269819/Uzhuj/A456718.html>
- [3] <http://baike.baidu.com/view/260711.htm>
- [4] 李群芳, 黄建. 单片微型计算机与接口技术. 北京: 电子工业出版社, 2001
- [5] 上海桑锐电子科技有限公司. SRWF-501 模块使用说明书, 2005
- [6] 李旭, 数据通信技术教程, 北京: 机械工业出版社, 2001
- [7] 钱晓捷, 陈涛, 16/32 位微机原理、汇编语言及接口技术, 机械工业出版社, 2005
- [8] 沙占友等, 单片机外围电路设计, 北京: 电子工业出版社, 2003
- [9] 求是科技, 单片机通信技术与工程实践, 北京: 人民邮电出版社, 2005
- [10] 范逸之, 陈立元, Visual Basic 与 RS-232 串行通信控制, 中国青年出版社, 2002
- [11] 雄桂喜, Visual Basic 简明教程, 机械工业出版社, 1998
- [12] Sarah E. Hutchinson, Computers, Communications, And Information A user's introduction, Higher Education Press, 2001

附录

VB 程序代码:

1: 列车端工程模块代码

```
Option Explicit
Public g_dbcon As New ADODB.Connection
Public strinsertsql As String
Public g_dbrct As New ADODB.Recordset

Public Function ConnectToServer() As Boolean
On Error GoTo connecterr
g_dbcon.ConnectionString =
"Provider=Microsoft.Jet.OLEDB.4.0;" & "Data
Source=E:\graduate\vb_zhong\数据库建立\列车数据库\Rail.mdb;"
& "Mode=ReadWrite"
g_dbcon.ConnectionTimeout = 30
g_dbcon.Open
ConnectToServer = True
Exit Function
connecterr:
ConnectToServer = False
MsgBox "错误代码: " & Err.Number & vbCrLf & "错误描述: " &
Err.Description, vbCritical + vbOKOnly, "连接错误"
End Function

Public Function ExecuteSQL(ByVal strSQL As String) As Boolean
On Error Resume Next
g_dbcon.Execute (strSQL)
If Err.Number > 0 Then
MsgBox "错误代码: " & Err.Number & vbCrLf & "错误代码: " &
Err.Description, vbCritical + vbOKOnly, "连接错误"
Err.Clear
ExecuteSQL = False
Else
```

```
        ExecuteSQL = True
    End If
End Function

Public Function Disconnect() As Boolean
On Error Resume Next
    If g_dbcon.State = adStateOpen Then
        g_dbcon.Close
    End If
    Disconnect = True
End Function

Public Function QueryEmpInfo(ByVal strSQL As String) As Boolean
On Error GoTo on_error
    Set g_dbrct = Nothing
    Call g_dbrct.Open(strSQL, g_dbcon, adOpenDynamic,
adLockBatchOptimistic, -1)
    QueryEmpInfo = True
Exit Function
on_error:
    MsgBox "错误代码: " & Err.Number & vbCrLf & "错误描述: " &
Err.Description, vbCritical + vbOKOnly, "错误"
    QueryEmpInfo = False
End Function
```

2: 列车端工程操作控件代码:

```
SuccessOrWrong = MSComm1.Input
Text2.Text = SuccessOrWrong

If SuccessOrWrong = Empty Then
    Timer1.Enabled = True
    Timer2.Enabled = True
Else
    Timer2.Enabled = False
    Command7_Click
End If
```

```
End Sub

Private Sub Command7_Click()
If SuccessOrWrong = ErrorToRepeat Then
    Command5_Click
End If

MSComm1.PortOpen = False
If g_dbcon.State = adStateOpen Then
    g_dbcon.Close
End If
End Sub

Private Sub Form_Load()
If ConnectToServer = False Then End
MSComm1.PortOpen = True
End Sub

Private Sub Timer1_Timer()
Timer1.Enabled = False
Command6_Click
End Sub

Private Sub Timer2_Timer()
Timer2.Enabled = False
Timer1.Enabled = False
Command5_Click
End Sub
```

3: 列车端工程显示窗体代码:

```
Private Sub Form_Load()
grdTBL.Cols = 7
grdTBL.Row = 0
grdTBL.Col = 0

grdTBL.Text = "列车号"

grdTBL.Col = 1

grdTBL.Text = "司机代号"

grdTBL.Col = 2

grdTBL.Text = "司机姓名"
```

```
grdTBL.Col = 3
grdTBL.Text = "最高速度"
grdTBL.Col = 4
grdTBL.Text = "最高轴温"
grdTBL.Col = 5
grdTBL.Text = "耗电量"
grdTBL.Col = 6
grdTBL.Text = "故障"
If Err.Number > 0 Then Exit Sub
Do While Not g_dbrct.EOF
    grdTBL.AddItem (Empty)
    grdTBL.Row = grdTBL.Rows - 2
    grdTBL.Col = 0

    grdTBL.Text = CStr(g_dbrct.Fields("列车号").Value)
    grdTBL.Col = 1

    grdTBL.Text = CStr(g_dbrct.Fields("司机代号").Value)
    grdTBL.Col = 2

    grdTBL.Text = CStr(g_dbrct.Fields("司机姓名").Value)
    grdTBL.Col = 3

    grdTBL.Text = CStr(g_dbrct.Fields("最高速度").Value)
    grdTBL.Col = 4

    grdTBL.Text = CStr(g_dbrct.Fields("最高轴温").Value)
    grdTBL.Col = 5

    grdTBL.Text = CStr(g_dbrct.Fields("耗电量").Value)
    grdTBL.Col = 6

    grdTBL.Text = CStr(g_dbrct.Fields("故障").Value)
    g_dbrct.MoveNext
Loop
End Sub
```

4: 车库端工程模块代码

```
Option Explicit
Public g_dbcon As New ADODB.Connection
Public g_dbrct As New ADODB.Recordset

Public Function ConnectToServer() As Boolean
On Error GoTo connecterr
g_dbcon.ConnectionString =
"Provider=Microsoft.Jet.OLEDB.4.0;" & "Data
Source=E:\graduate\vb_zhong\ 数据库建立 \ 车库数据库
\Station.mdb;" & "Mode=ReadWrite"
g_dbcon.ConnectionTimeout = 30
g_dbcon.Open
ConnectToServer = True
Exit Function
connecterr:
ConnectToServer = False
MsgBox "错误代码: " & Err.Number & vbCrLf & "错误描述: " &
Err.Description, vbCritical + vbOKOnly, "连接错误"
End Function

Public Function ExecuteSQL(ByVal strSQL As String) As Boolean
On Error Resume Next
g_dbcon.Execute (strSQL)
If Err.Number > 0 Then
MsgBox "错误代码: " & Err.Number & vbCrLf & "错误代码: " &
Err.Description, vbCritical + vbOKOnly, "连接错误"
Err.Clear
ExecuteSQL = False
Else
ExecuteSQL = True
End If
End Function

Public Function Disconnect() As Boolean
On Error Resume Next
If g_dbcon.State = adStateOpen Then
```



```
        g_dbcon.Close
    End If
    Disconnect = True
End Function

Public Function QueryEmpInfo(ByVal strSQL As String) As Boolean
On Error GoTo on_error
    Set g_dbrct = Nothing
    Call g_dbrct.Open(strSQL, g_dbcon, adOpenDynamic,
adLockBatchOptimistic, -1)
    QueryEmpInfo = True
    Exit Function
on_error:
    MsgBox "错误代码: " & Err.Number & vbCrLf & "错误描述: " &
Err.Description, vbCritical + vbOKOnly, "错误"
    QueryEmpInfo = False
End Function
```

5: 车库端工程操作代码:

```
. Private Sub Form_Load()
If ConnectToServer = False Then End
MSComm1.PortOpen = True
Command5_Click
End Sub
```

```
Private Sub MSComm1_OnComm()
Dim aa As String
aa = "errortorepeat"
Select Case MSComm1.CommEvent
    Case comEventRxParity
        Timer1.Enabled = False
        Timer2.Enabled = False
        MSComm1.Output = aa
        Command5_Click
End Select
```

```
End Sub
```

```
Private Sub Timer1_Timer()  
Timer1.Enabled = False  
Command5_Click  
End Sub
```

```
Private Sub Timer2_Timer()  
Timer2.Enabled = False  
Command6_Click  
End Sub
```

```
Private Sub Command1_Click()  
Dim strSQL As String  
  
strSQL = "select*from Station where (司机代号>0)"  
  
If Text1.Text <> Empty Then  
    strSQL = strSQL & "and (列车号=" & Text1.Text & ")"  
End If  
If Text2.Text <> Empty Then  
    strSQL = strSQL & "and (司机代号=" & Text2.Text & ")"  
End If  
If Text3.Text <> Empty Then  
    strSQL = strSQL & "and (司机姓名='" & Text3.Text & "')"  
End If  
If QueryEmpInfo(strSQL) = True Then  
    'Unload Me  
    frmrcodelist.Show  
End If  
End Sub
```

6: 车库端工程显示代码:

```
Private Sub Form_Load()
```

```
grdTBL.Cols = 7
grdTBL.Row = 0
grdTBL.Col = 0

grdTBL.Text = "列车号"
grdTBL.Col = 1

grdTBL.Text = "司机代号"
grdTBL.Col = 2

grdTBL.Text = "司机姓名"
grdTBL.Col = 3

grdTBL.Text = "最高速度"
grdTBL.Col = 4

grdTBL.Text = "最高轴温"
grdTBL.Col = 5

grdTBL.Text = "耗电量"
grdTBL.Col = 6

grdTBL.Text = "故障"

If Err.Number > 0 Then Exit Sub
Do While Not g_dbrct.EOF
    grdTBL.AddItem (Empty)
    grdTBL.Row = grdTBL.Rows - 2
    grdTBL.Col = 0

    grdTBL.Text = CStr(g_dbrct.Fields("列车号").Value)
    grdTBL.Col = 1

    grdTBL.Text = CStr(g_dbrct.Fields("司机代号").Value)
    grdTBL.Col = 2

    grdTBL.Text = CStr(g_dbrct.Fields("司机姓名").Value)
    grdTBL.Col = 3

    grdTBL.Text = CStr(g_dbrct.Fields("最高速度").Value)
    grdTBL.Col = 4

    grdTBL.Text = CStr(g_dbrct.Fields("最高轴温").Value)
    grdTBL.Col = 5

    grdTBL.Text = CStr(g_dbrct.Fields("耗电量").Value)
```

```
    grdTBL.Col = 6
    grdTBL.Text = CStr(g_dbrct.Fields("故障").Value)
    g_dbrct.MoveNext
Loop

End Sub
```