摘 要

计轴设备用于实时检查所防护轨道区段占用还是空闲，其作用等效于轨道电路。它的工作原理是基于列车驶入或驶出轨道区段计数点时所记录轴数的比较结果，以此确定该轨道区段处于占用还是空闲状态。目前我国还在广泛使用的轨道电路，受外界因素影响大，存在道床泄漏电阻过低、轨道电路分路不良等情况，难以保证轨道电路正常运用的电气参数，使轨道电路无法正常工作，使电务信号联锁失效。这些状况难以适应列车速度和安全性能的进一步提高。在安装计轴设备的区段内，由于计轴设备不受道床条件影响，可以彻底解决以上出现的问题，从而有效保证行车安全。

论文从实际需求出发，分析站内多点计轴系统的特点及要求，得出系统需求，结合计算机、网络通信、控制技术，提出了一种采用嵌入式实时系统 Windows CE .NET 为核心的解决方案。

论文的主要内容包括绪论,多点计轴系统总体设计，多点计轴系统硬件设计，Windows CE 操作系统,多点计轴系统软件设计与实现等部分。绪论部分主要对该系统的研究背景及研究意义作了简要说明；第二章通过系统所需功能的分析，描述了系统总体设计；第三章分析描述了二取二方式的硬件架构方案的设计;第四章对 Windows CE 操作系统的定制开发、应用开发等作了介绍。第五章中应用面向对象技术对系统的软件部分进行了设计与实现；最后对文中的设计实现给出结论，并对其可进一步改进之处作了说明。

关键词：铁路信号；多点计轴；二取二；Windows CE

Abstract

Axle-counting Devices are used to perform real-time check of the status of railway sections, which are occupied or not. The functions are equal to that of track circuits. The operational principles are based on the computation result using the axle-counting information in the section. At present, track circuits are wildly used in our country. But they are easily influenced by circumstance such as low roadbed bleeder resistance, bad shunting, etc. These instances make wrong operations occur and lead to the malfunction of signaling interlock. These circumstances baffle the improvement of safety and the train speed. Axle-counting devices are not influenced by the circumstance has mentioned, and can ensure the transport safety.

In this paper, the requirements and specialties of the intra-station multipoint axle-counting system are studied. Then with the integration of computer technology, network communication technology and automatic control technology, a kind of solution is proposed, in which the embedded real-time operation system, Windows CE .NET is used.

The contents of the paper consist of introduction, system framework design, hardware design, introduction of Windows CE, software design and implementation, etc. In the introduction, the background and the significance of the study in the system are described. In chapter 2, the analysis for the system requirements is described, and the system framework is designed. Then the system hardware design using two voting method is described by analysis. In chapter 4, the procedure of developing and customizing Windows CE operating system, and of the application, is introduced. In chapter 5, the Object-Oriented technology is applied, and the software of the system is designed and implemented. At last, the design and the implementation are summarized, and the methods which can improve the system are described.

Key Words: Railway signaling; Multipoint axle-counting; 2 voting; Windows CE

第 1 章 绪论

1.1 研究意义与必要性

铁路信号设备是铁路运输安全生产的重要技术设备之一，对保障行车安全、提高运输效率和运营管理水平具有关键性作用。现阶段车站站内的轨道电路系统是重要的铁路信号设备之一。轨道电路的工作状况直接影响到铁路行车安全及行车效率。

轨道电路有其特定的优点，但是也有一些自身无法克服的缺点。如不经常行车的区段，出现钢轨生锈现象；码头地带由于潮湿，钢轨也易生锈；沙漠地带，风沙大，沙尘浮于钢轨表面；轨道区段跳线锈断；轨道绝缘破损等。由于站内轨道电路是信号联锁的重要组成部分，以上情况会造成轨道电路分路不良，成为安全生产的突出薄弱环节，由于轨道电路分路不良造成的事故给全路带来过重大损失和惨痛的教训。另外，在铁路隧道内由于潮湿等客观因素影响，道床电阻极低，现有的轨道电路不能正常工作，经常出现红光带，严重影响行车效率。

计轴技术是以计算机为核心，辅以外部设备，利用统计车辆轴数检测相应轨道区段是否有列车占用或列车已出清的技术。计轴设备是实时检查区间线路、站内股道、道岔、平面交叉及道口区段占用或空闲状态的安全设备。计轴设备的基本作用与轨道电路等效，因此适用于一切应用轨道电路的场合。

计轴系统不受道床条件影响，可以彻底解决出现红光带的问题。由于不用钢轨做电路通道，所以特别适用于长大区间、潮湿地段、钢枕地段、强风沙地段、山区、矿区、桥隧、各种专用线等区段。

由于受多种环境因素影响，在我国，许多轨道电路不能正常工作，存在相当数量的“压不死”轨道电路区段及许多长期红光带轨道电路区段。据 2006 年全路电务工作会议分析资料统计，全路约存在 27000 处“压不死”轨道区段，已严重威胁铁路行车安全和运输效率。因此，铁道部运输局要求：2006 年要把解决“压不死”区段作为保障行车安全的首项整治内容，并明确提出采用计轴技术作为解决该问题的主要方案和手段之一，并积极加以推广应用。

现阶段，我国在站间闭塞系统中使用计轴技术已经较为广泛。区间行车安全性也得到了较大提高。在站内的个别轨道电路区段，也已有较多应用。

多点计轴技术就是采用计轴技术替代较多轨道电路采用的设计方案。站内存在较多的不良轨道电路区段时，必须要使用多点计轴技术。另外，利用计轴技术对区间分割，实现列车追踪，也在大规模应用，也必须采用多点计轴技术。

如果是大型站场，其自然环境条件较差的情况下，存在较多的不良轨道电路区段，采用计轴方案解决该问题的应用方案还比较少，而且解决方案还不完善。

目前，国内外已有少量相关产品，但存在以下问题：

1、采用主机叠加处理的方式，容量较少，在较大站场和“压不死”区段或长期红光带轨道电路区段较多车站使用时，产生了较大困难。

2、成本较高，特别在大型站场和“压不死”区段较多车站使用时，设备投资无法令人接受。

所以，采用新技术，设计新方案，研究开发大容量、低成本的多点计轴处理设备对保障行车安全及提高车站作业效率是十分必要的。

1.2 国内外研究概况

发达国家的计轴技术起步比较早，并已开发出技术成熟的产品推广使用，如阿尔卡特公司的 ZP30CA 计轴器、德国西门子公司的 AZS350T 计轴器及德国 SEL 公司研制的 SIG-L90 系统。

以西门子 AzS(M)350M 型微机计轴系统为例，其核心是 ZP43V 型计轴点设备和 AzS(M)350M 运算单元。ZP43V 型计轴点设备安装于铁路轨道区段的各端点位置，每个端点位置安装一套，使得这几个 ZP43 V 计轴点共同检测这个封闭的轨道区段。ZP43V 型计轴点通过车轮传感器感应进出区段的车轮及其运行方向，并将感应信号预处理后，将预处理后的信号经连接电缆系统传输至室内运算单元。

AzS(M)350M 在多点计轴技术中应用时，如果两个相邻区段采用的都是计轴技术，则共用的计轴点信息可以进行复用。对于不相

邻的计轴区段无法复用。这时候使用主机堆叠的方式，即使用独立的一套微机计轴系统。当计轴区段数量越多时，堆叠量越大，成本非常高，且实施施工也更为困难。

我国铁路从 20 世纪 80 年代开始计轴技术的研究与应用，国内许多研究机构在学习发达国家的计轴技术基础上，借鉴多年来国外计轴设备开发应用的成功经验和存在问题，自主研发了多种计轴设备产品。上海铁路局于 1985 年从联邦德国引进 $A_2L70/SK30$型计轴器，于 1986 年应用在沪杭线。其他的产品还有：哈尔滨铁科所研制开发了 JWJ-C2 型计轴器，成都铁路通信设备厂研制开发了 JZ1-G 型计轴器，等等。

现阶段国内用于站内多点计轴应用的产品很少，主要就是成都铁路通信设备厂研制开发的 JZ.GD-1 型站内微机计轴设备。该计轴设备可对站内 8 个计轴点进行运算控制，在计轴点数量多于 8 个时需要使用主机堆叠方式。因此，在计轴点非常多时整个系统仍然会很复杂。

因此，设计计算机集中控制的多点计轴新方案，研究开发大容量、低成本的多点计轴处理设备具有较为迫切的需求。研究内容具有创新性。

1.3 研究内容

本论文以实际现场需求出发，分析站内多点计轴系统的特点及要求，得出系统实际需求，结合计算机，网络通信、控制技术提出了一种采用嵌入式实时系统 Windows CE .NET 为核心的解决方案。文中对系统整体的方案设计进行了详细的描述。包括硬件平台的结构设计、软件整体的结构设计及组成部分的功能设计及实现方法、以及使用的平台和开发环境都作了较为详细的阐述，并完成了实现工作。

论文中，采用的计轴设备是成都通信工厂的 JZ1-G 型计轴器，是已经通过铁道部鉴定的产品，具有高安全性、可靠性。以后的设计开发、测试都是以 JZ1-G 型计轴器为基础。

论文的安排如下：

第一章首先论述了站内多点计轴系统的发展与现状，从而给出本课题研究的必要性。提出了研究多点计轴解决方案的必要性以及

研究内容和安排。

第二章通过对系统总体需求的分析，对系统的各个方面进行了充分的考虑，设计了站内多点计轴系统的整体方案，并且对系统工作的流程进行了描述。

第三章对系统的硬件结构进行了设计，并对其中实现的原理进行了详细的描述。

第四章对软件方面的相关技术做了相应介绍。包括对实时操作系统 Windows CE 开发流程和定制过程、应用开发需要使用的软件的介绍。

第五章完成多点计轴系统的软件设计和实现，包括软件整体构架、各程序模块的设计、内部算法的设计及实现、界面调试显示部分的设计等。

论文的结论部分对所做工作进行了总结，并对论文的不足之处已及对后续工作进行了说明。

第 2 章 多点计轴系统总体设计

2.1 总体需求

站内多点计轴系统采用计算机集中控制方式，对站内所有计轴区段的计轴设备进行采集，运算得出各计轴区段的空闲或占用状态，驱动轨道继电器表示正确的区段状态。同时，还需要完成的功能有计轴复零信息的采集和继电器状态的采集。

多点计轴系统的需要完成的功能如图 2-1 中所示：

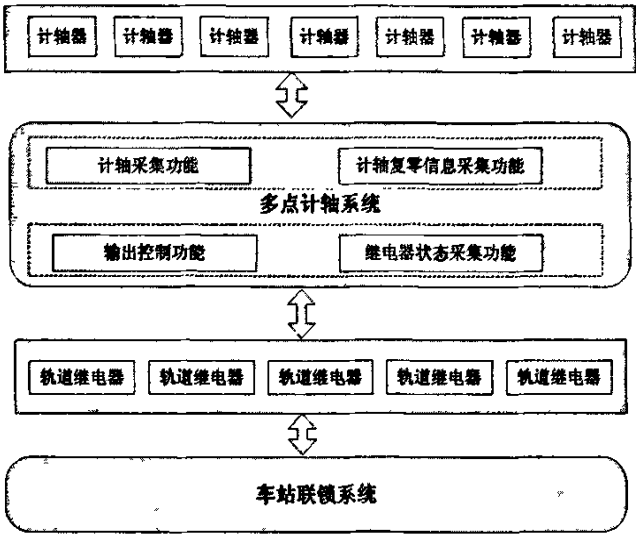

图 2-1 多点计轴系统功能框图

对系统总体技术需求进行分析，得出系统需要具备下面的功能：

2.1.1 采集正确的计轴器轴数信息

确保采集正确的计轴器轴数信息，是系统正常工作的基础。如果采集到错误的轴数信息，系统运算后必然得出错误的结果，这对行车安全将造成最为严重的后果。

对于计轴器而言，必须要保证传感器的采集完全正确，传感器与室内计轴相关设备的通信完全正确。

JZ1 型计轴器采用双传感器和双 CPU 采集一组两个轴数，在自动站间闭塞等应用中，均以两个轴数相同时才确认轴数有效。其

二取二方式保证了采集到的轴数正确无误。

本系统设计中，主计算机必须和计轴器通信，获取得到轴数信息。如果采用单台主机方式，在计算机因异常情况，内存中保存的轴数发生变化，结果是非常危险的。

在很短的时间间隔内，系统中两台主机同时出错并且呈现同一模式的概率几乎为 0。因此采用多机硬件冗余来实现故障-安全的系统的很好的解决方法。

因此，为了保障系统的安全，在本系统中，设计采用双主机方式，以二取二方式比较轴数信息，只有双机数据一致时才确定计轴器数据为正确的。

2.1.2 根据采集的轴数准确运算

在计轴轨道区段，各端点均安装计轴器，对所有计轴器采集得到正确的轴数信息后，需要进行区段运算以得出轨道区段是否占用的信息。

计轴轨道区段有多种类型：无岔区段、和道岔区段。其中道岔区段又分为一送双受、一送三受、交叉渡线方式。交叉渡线方式的计轴器设置是分解为一送三受方式完成的，故可归入一送三受的方式。因此一个计轴轨道区段可能需要安装 2-4 个计轴器。

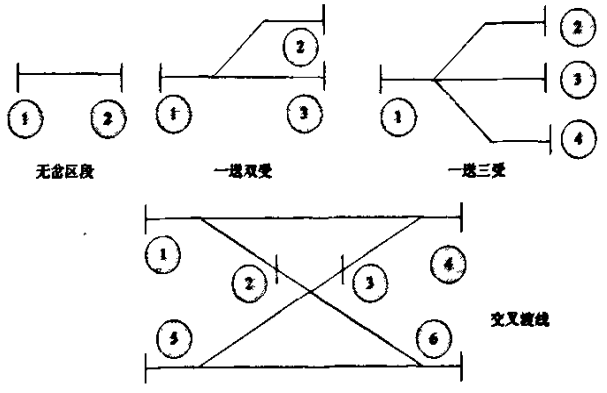

图 2-2 计轴区段计轴器设置方案

在相邻的两个区段都使用计轴技术，则共用的计轴器需要进行复用。例如图 2-2 中交叉渡线部分的计轴器 2、3，即需要复用。复用的方法是每一个计轴器 CPU 存储两个轴数，分别代表该端点的上、下行方向的计轴轴数信息。这两个轴数可以不同。

系统需判断计轴区段是否空闲。判断的原理如下：

1、在计轴区段处于空闲状态时，上行方向的计轴器轴数之和与下行方向的计轴器轴数之和相同。

例如，对于无岔区段，计轴器 1、2 的轴数相等；对于一送三受区段，计轴器 1 的轴数与计轴器 2、3、4 的轴数之和相等。

2、当上一条描述的上下行方向轴数和不等时，判断为区段占用。

区段运算结果依赖于所有轴数的正确性。单机的区段运算结果出错的可能性远远大于双机运算比较的结果。为保证区段运算结果正确无误，在本系统中需要采取措施比较区段运算结果。

2.1.3 可靠的输出控制

轨道继电器是计轴运算结果后的输出目标。区段占用时，需要控制轨道继电器处于落下状态；当计算结果为区段空闲时，确保轨道继电器的可靠吸起。

在区段内有计轴器发生故障时，无法判断区段的占用或者空闲，根据故障-安全原则，必须采取措施，保障信号故障安全。

实现铁路信号故障安全的重要方法之一是安全侧分配方法。该方法给信号器件或设备分配安全侧，然后，采用故障-安全技术使设备发生故障时导向安全侧。

常用的安全侧分配方法有：

- ✧ 基于能量的安全侧定义方法；
- ✧ 基于闭路法和串联法的安全侧定义方法；
- ✧ 基于时间安全侧定义方法；

另外，危险侧故障率最小化技术、故障弱化技术以及联锁方法都是提高系统故障安全度的有效方法。

在本系统中，考虑使用其中的一种方法或几种方法结合使用，以实现铁路信号安全。

2.1.4 可靠的继电器状态采集

对继电器状态采集可以使系统能够对继电器的输出控制结果进行检查，了解继电器的工作状态。如果系统对继电器输出控制是使继电器吸起，而采集的继电器状态是处于落下状态，说明系统中输出电路或者继电器采集电路、继电器本身工作不正常。

继电器的状态采集也需要高度可靠，否则采集到错误信息，对正常运作的系统无疑带来的是副作用。

要继电器状态采集结果正确无误，在本系统中同样采取以二取二方式进行采集、比较运算结果。

2.1.5 可靠的计轴复零状态采集

系统中设置了计轴复零按钮，其作用是在计轴区段实际空闲时，可能因其中的计轴器维修启用后，或者干扰的原因，造成该计轴区段运算结果为区段占用。这种情况下，需要操作人员确认区段空闲，手动按下该区段的计轴复零按钮，用于实现该区段内所有计轴器轴数重置为零。

计轴复零按钮不能在计轴区段内的计轴器运行正常时按下，否则如果此时有车占用区段，后果是极其严重的。需要采取措施，保证计轴复零按钮不会被随意按下，同时也要保证采集到的计轴复零状态是正确的。

本系统中，采取对计轴复零按钮采用带计数器加铅封的方式，在执行该操作前必须登记，得到相关部门人员认可，破铅封后，按压计轴复零按钮 13 秒以上，系统执行该区段内计轴器轴数置零操作。这里采取的是基于时间的安全侧定义方法。

2.2 系统的总体结构设计

根据总体需求分析，在系统总体结构中，拟采用的主要技术方案有：

- ✧ 二取二的双主机控制方案；
- ✧ 二取二的计轴轴数采集、软件比较方案；
- ✧ 二取二的区段状态软件比较方案；
- ✧ 二取二的继电器输出控制硬件方案；
- ✧ 二取二的继电器状态采集、软件比较方案；
- ✧ 二取二的计轴复零状态采集、软件比较方案。

设计的系统的总体结构如下：

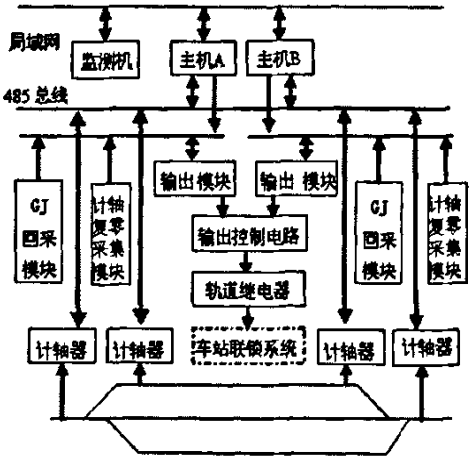

图 2-3 多点计轴系统总体结构图

下面对系统结构中各组成部分作简要描述：

1、主机 A 与主机 B 以及监测机通过 100Mbps 局域网连接进行通信，A、B 机之间高速传送待比较的数据。监测机与 A、B 机通信，记录各区段的轴数信息、区段运算信息、继电器实际状态信息、计轴复零状态信息以及各种错误信息。

2、主机 A 与主机 B 均连接到计轴 CPU 所在的 485 总线上，获取计轴器的轴数信息。轴数信息经过网络传输给对方进行比较。

3、主机 A、B 各自拥有一套继电器输出模块、继电器状态采集模块和计轴复零采集模块。它们连接到属于各主机的一条控制 I / O 的 485 总线上。

4、主机 A、B 驱动属于各自总线的继电器输出模块，共同作用于输出控制电路部分，完成对轨道继电器（简称 GJ）的控制，同时需要满足故障-安全原则。

5、主机 A、B 各自独立采集继电器状态和计轴复零按钮状态。两组状态数据也通过网络传输后进行比较。

2.3 系统的工作流程设计

根据系统的总体需求和设计的总体结构，可以设计得到系统的工作流程：

1、系统主机对输入/输出模块通信，置模块状态为初始值。

2、系统主机与计轴 CPU 之间通过计轴 485 总线通信，读取得到本机计轴器轴数。

3、系统主机与计轴复零状态采集模块通过 I/O 485 总线通信，读取得到本机计轴复零按钮信息。

4、系统主机之间通过网络通信，把本机计轴器轴数、计轴器故障信息、计轴复零按钮信息、继电器状态信息和另一主机相互传送。

5、系统主机以区段方式取得区段内的各计轴器轴数与另一主机相应计轴器轴数进行比较，如果均相同则进行运算得出区段状态值。

6、根据运算得出的区段状态值，系统主机驱动输出模块进行响应的继电器状态输出控制。

7、系统主机与继电器状态采集模块通过 I/O 485 总线通信，读取得到本机继电器状态信息。

8、系统主机进行区段状态值和本机继电器状态信息、另一主机继电器状态信息比较。

9、系统主机进行本机和另一主机计轴复零按钮信息比较。比较结果一致且为按下状态，在按下状态保持 13 秒后，对相应区段内各计轴器进行复零操作。

10、各项比较的结果如果不一致，需进行相应的故障-安全导向处理。

第 3 章 多点计轴系统硬件方案设计

3.1 硬件级的故障-安全设计

3.1.1 输出控制的故障-安全

在系统总体需求的分析过程中，对轨道继电器的输出控制硬件实现方案给了多种选择。现在在已有的系统总体结构设计基础上，我们考虑具体的选择。

系统中选择了双机控制方式，可以看出，对轨道继电器的输出控制，采取基于能量的安全侧定义并结合串联法是较为合适的。

首先定义轨道继电器落下状态为安全侧。双主机的控制电路以串联方式连接，只有两台主机的均有输出时，才构成回路，能使轨道继电器吸起，处于危险侧。如果一台主机未输出，闭环电路被断开，切断了控制能量，轨道继电器落下，处于安全侧。

3.1.2 输入采集的故障-安全

根据总体需求里的分析设计结果，采用双机各自采集的方式，具体的比较依靠软件来完成。

因此，输出控制和输入采集的功能逻辑框图可以如下表示：

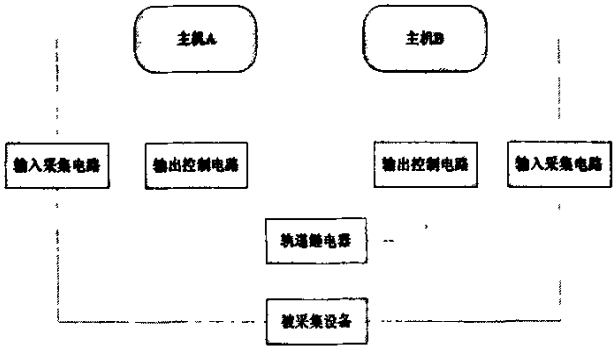

图 3-1 双机输入、输出功能逻辑框图

3.1.3 主机运行时的故障-安全考虑

为能在系统软件运行出现问题而发生死机后，能进行有效和及时的处理，可以设计为死机后系统自动进行重启，系统重新进入准

备状态。在系统主机的选择时，采用带看门狗的主机。

3.2 保障安全的系统硬件实时需求分析

多点计轴系统是有时限要求的硬实时系统。系统中的硬件需要完成的实时性任务有：

1、计轴器在列车通过时，必须在允许的时限内将变化的轴数信息传送到主机。

2、计轴区段由空闲转为被占用，区段轨道继电器由吸起转落下的时限必须保证。

3.2.1 计轴设备控制分析

1、计轴器的响应时间

系统采用的 JZ-1 型计轴设备，与主机之间的通信采用 485 总线方式，主机和计轴器之间的通信速率为 4800bps。

主机采用轮询的方式同 485 总线上的所有计轴器通信。其通信规约中，主机发往计轴器的字节数为 7，计轴器响应的字节数为 10。

通过理论计算可以得出主机取得一个计轴器轴数数据的响应时间约为 25ms，在计轴器硬件上实际测试得到的读取时间也基本是这个值。

2、总线上计轴器数量的确定

系统需要保证在轮询所有计轴器的时间范围内，当车辆通过某段轨道，该端点处计轴器轴数变化信息必须被采集得到，并且加上输出部分执行驱动轨道继电器的时间，对于铁路系统，要求轨道继电器由吸起转落下的响应时间应不超过 300ms，轨道继电器由吸起转落下的响应时间应不超过 500ms。暂时忽略驱动输出时间及其他时间损耗因素，可得总线上最大计轴器数目为：300/25，约 12 个计轴器。显然还需要为时间限留有余度。80%的余度是个较好的选择。假设输出控制和其他时间损耗之和比计轴通信至少低一个数量级，则可以被忽略掉。因此，在该假设成立情况下，一根总线上的计轴器数量确定为 10 个。

3、485 总线数量的确定

在系统中，如果计轴点的数量大于 10 个，可以在硬件上增加 485 总线的数量来满足要求。总线数量 N = 计轴点数量 / 10。

3.2.2 输出电路控制的时间分析

1、输出电路的响应时间

在系统中，计轴器的通信响应时间为 25ms。要尽快执行输出，应选择响应时间数量级低于计轴器通信响应时间的硬件设备。这样，尽量使其他设备工作消耗的时间可以忽略，以保证整个系统由计轴器通信到输出控制继电器动作满足实时需求。

2、输出电路硬件的可靠性

输出电路硬件的可靠性关系到整个系统的安全，应尽量满足严格的级别标准，例如工业级的硬件设备或军用级的硬件设备。

3.3 系统总体硬件方案

通过上述分析，设计了多点计轴系统的硬件方案。

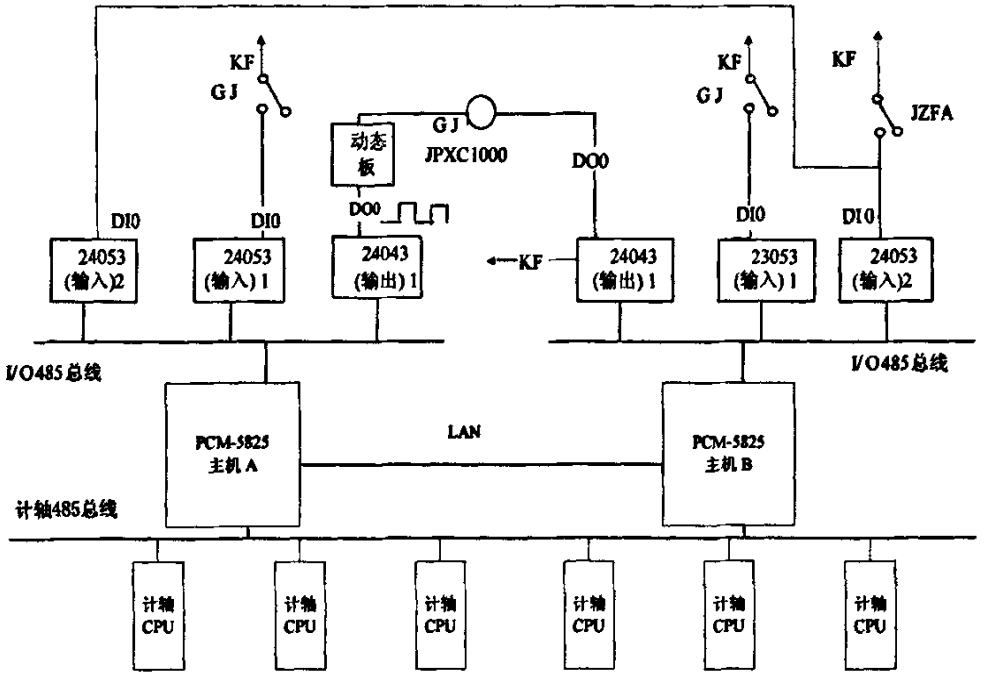

图 3-2 多点计轴系统硬件方案图

该硬件方案实现的功能有：

- 计轴器设备的双机采集。
- 二取二的硬件输出控制。
- 继电器状态的双机采集。
- 计轴复零状态的双机采集。

◆ 局域网方式的双机通信。

在计轴设备控制分析时，做了如下假设：输出控制和其他时间损耗之和比计轴通信至少低一个数量级，则可以被忽略掉。在此，对输入/输出控制的硬件响应时间进行说明：

◆ 系统采用了 485 总线方式连接输入/输出模块，通信速率设置为 115200bps。

◆ 主机轮询输入、输出模块的命令字节数分别为 5、8 个字节，模块响应字节数分别为 8、2 个字节。可计算得出响应时间不超过 1ms。

◆ 因此，输入/输出控制的硬件响应时间满足该项假设。系统实时性可以得到保证。

图 3-2 中的计轴 485 总线，根据计轴点的增多，需要相应进行增加。

以下对方案中的硬件设备作简要说明：

(1) 系统主机采用的是嵌入式计算机 PCM－5825,其主要规格为：

✧ 板载 AMD Geode GX1－300 CPU

✧ 集成 10 / 100M 以太网

✧ PC-104 总线

✧ 支持 VGA / LCD / LVDS

✧ 4 串口

✧ 支持 CF 电子盘。

✧ 具备 1-62 秒计时长度的看门狗定时器

主机 A 与主机 B 之间通过 802.3 局域网连接进行通信，交换数据信息。

(2) 主机与各计轴 CPU 通过一条 485 总线相连接，以下称计轴 485 总线；而主机与输入输出模块是通过另一条 485 总线相连，以下称 I / O 485 总线。

(3) 输入模块：采用数字量输入控制模块 ARK24053D,支持 485 总线并具备 16 路数字量输出。硬件连接及动作逻辑：

✧ 每个输入模块的 DIO－DI15 端子，均连接到相应的 GJ 采集接点或计轴复零按钮采集接点。

✧ GJ 采集接点或计轴复零按钮采集接点的另一端，均接至电源地端 KF。

- ✧ 在 GJ 采集接点或计轴复零按钮采集接点为常开状态时，ARK24053D 相应输入端子未拉低至与地电平相差 2V 以内，此时该端子为逻辑状态 1；在接点闭合状态时，该端子为逻辑状态 0。

(4) 输出模块：采用数字量输出控制模块 ARK24043D，支持 485 总线并具备 16 路数字量输入。硬件连接及动作逻辑：

- ✧ 每个输出模块的 DO0－DO15 端子，均连接到相应动态板的输入端。
- ✧ ARK24053D 在接收到 485 总线置相应输出端子为逻辑状态 1 时，将该输出端拉低至接近于地电平；相应输出端子为逻辑状态 1 时，该端子为高阻态。

(5) 动态板：采用的是成都铁路通信工厂的与计轴、联锁设备等配套使用的动态信号控制板，动态板的各输出端接相应 GJ 的 KZ 端。动作逻辑：

- ✧ 在输入端有动态信号时，动态板相应一路输出端产生高电平输出。
- ✧ 只要输入端没有动态信号，无任何输出。

(6) 轨道继电器：采用的是偏极继电器 JPXC1000，偏极继电器可鉴别电源的极性，如果将偏极继电器正极端子接地，这样当电路发生故障时，也不会使偏极继电器励磁吸起。动作逻辑：只有在控制端子连接极性正确时才能使之正确动作吸起，否则仍然处于落下状态。从而驱动条件更严格，安全性也更高。

3.4 加快区段运算的硬件连接设计

考虑对一个区段的计轴器通信，如果可以并行进行，则可以以最短时间取得整个区段所属计轴器轴数。也就可以更快的得到区段运算结果。

因此，对硬件连接设计作如下调整：

1、不采用以计轴点多少的方式设置计轴 485 总线数量，在计轴点未超出 40 个时，均采取设置 4 条 485 总线的设计；未超过 80 个时采取设置 8 条 485 总线的设计。对于主机 PCM-5825 采取在 PC-104 总线上堆叠 PC-104 多口 485 控制卡的方式。

2、一个计轴区段内的各个计轴器分配到不同 485 总线，尽量

均匀地将所有计轴器分配给各条 485 总线。

通过该设计，每个计轴区段的各计轴器轴数能更为快速地采集到，并供区段运算使用，提高了系统性能。

3.5 系统硬件工作流程说明

图 3-2 中，输入模块和输出模块仅示意了输入 1、输入 2 和输出 1 共 3 个模块。可以驱动 16 个轨道继电器、采集 16 路继电器状态和采集 16 路计轴复零按钮状态。根据计轴器的数量，相应增减模块的数量。

结合图示，简述系统硬件的工作流程如下：

1、主机通过计轴 485 总线采集得到各计轴器轴数。

2、主机 A、B 分别通过独立的 I/O 485 总线采集得到继电器状态和计轴复零按钮状态。

3、主机 A、B 各自采集的信息通过局域网进行传输。

4、主机 A、B 对双机数据进行比较，确认采集的轴数信息一致后，主机 A 对本机相应输出模块相应端子（如输出模块 1 的 DO0 端子）输出逻辑 1、0 交替的动态控制信息。主机 B 对该端子输出逻辑 1 的控制信息。

5、主机 A 驱动动态板，接通 JPXC1000 的正极，主机 B 将 JPXC1000 的负极与地相连通，JPXC1000 吸起。

可以看出，系统结合了软件上的二取二输入比较，在硬件上实现二取二的输出控制。

3.6 系统双机工作原理说明

3.6.1 二取二硬件输出控制

设计的多点计轴二取二硬件输出控制原理如图 3-3 所示：

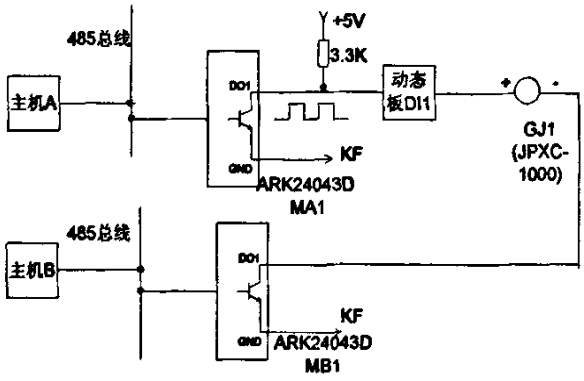

图 3-3 多点计轴二取二硬件输出控制原理图

动作逻辑说明：

- 主机接收到计轴 CPU 的轴数信息后，进行双机比较。
- 如果比较结果一致，则控制 ARK24043D 相应端子输出相应逻辑电平。图中示意的是数字控制模块 MA1 及 MB1 的 DO1。
- 如果计轴区段状态为空闲，则主机 A 产生动态输出至 MA1 的 DO1，主机 B 产生静态输出至 MB1 的 DO1。
- 动态驱动信号接入动态板的输入端 DI1，动态板输出端同轨道继电器 GJ1 的正极端子相连。
- 主机 B 的输出模块相应端子接至 GJ 的负极端子，与动态板输出端共同作用驱动相应轨道继电器吸起。
- 如果计轴区段状态为占用，则主机 A、B 均输出逻辑电平“0”，输出端子处于高阻态。轨道继电器处于落下状态。

通过动作逻辑，可以看到系统在硬件层完成了二取二输出。具有故障-安全导向。

3.6.2 双机继电器状态采集

设计的多点计轴双机继电器采集原理如图 3-4 所示：

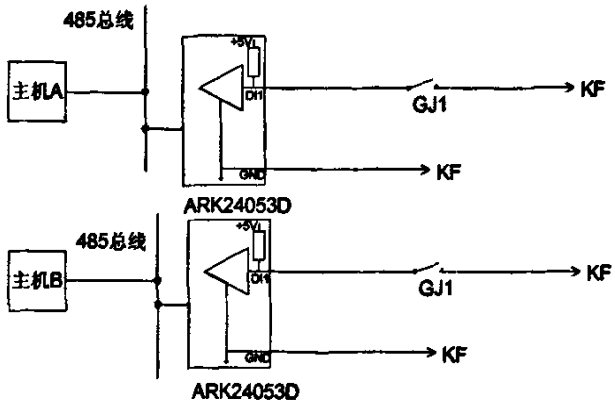

图 3-4 多点计轴双机继电器采集原理图

图 3-4 中示意的是主机 A、B 分别通过独立的 485 总线连接数字输入模块 ARK24053D，分别采集来自轨道继电器 GJ1 的两组接点的状态信息。

动作逻辑说明：

- GJ1 吸起时，两组接点均闭合，主机 A、B 通过采集模块信息得到 GJ1 状态为吸起，并通过局域网送对方主机进行比较。
- 只有采集得到的状态和输出控制状态双机均一致时为正常，否则进行相应的错误处理。

3.6.3 双机计轴复零状态采集

设计的多点计轴双机计轴复零按钮采集原理如图 3-5 所示：

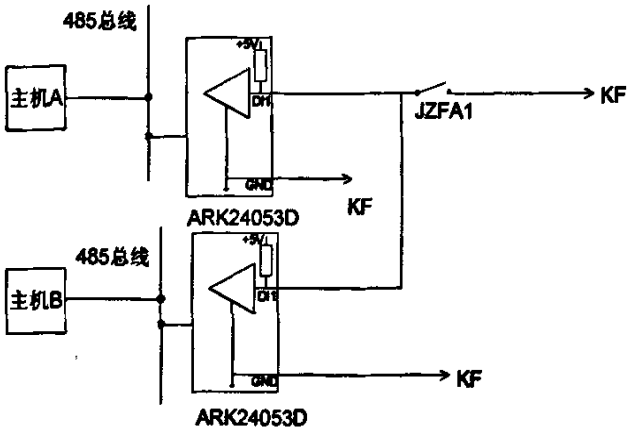

图 3-5 多点计轴双机计轴复零按钮采集原理图

图 3-5 中示意的是主机 A、B 分别通过独立的 485 总线连接数字输入模块 ARK24053D，分别采集来自 JZFA1 的两组接点的状态信息。

动作逻辑说明：

- JZFA1 按下时，两组接点均闭合，主机 A、B 通过采集模块信息得到 JZFA1 状态为吸起，并通过局域网送对方主机进行比较。
- 只有采集得到的状态和输出控制状态双机均一致时为正常，否则进行相应的错误处理。

多点计轴双机计轴复零按钮采集原理与多点计轴双机 GJ 采集原理一致，硬件上不同之处是计轴复零按钮每个区段只有一个按钮。

第 4 章 Windows CE .NET 操作系统

4.1 嵌入式操作系统 Windows CE .NET

Windows CE .NET 是微软公司在嵌入式操作系统市场上的一个重要产品。它的第一版于 1996 年发布。但是，直到 Windows CE 3.0 的发布，它才真正被人们所接受，并逐步取得了成功。2002 年 1 月，微软发布了 Windows CE .NET 4.1，这是一个非常成功的版本，它与 2004 年发布的 Windows CE .NET 4.2 版的差别很小，而且它的平台定制工具 Platform Builder 和应用软件开发工具 Embedded Visual C++都是非常实用的开发工具。

Windows CE .NET 是一个 32 位、多线程、多任务、完全抢占式的操作系统，模块化设计允许它对于从掌上电脑到专用工业控制器的电子设备进行定制。

Windows CE .NET 的体系结构采用独立于通常的程序设计语言并且和 Windows 兼容的 API 的方式，这样就可以保障 Windows CE .NET 的组件化和 ROM 化，充分适应有限的存储空间和各种不同芯片的要求。Windows CE .NET 是模块型的操作系统。这意味着可选择、组合和配置的模块和组件来创建用户版的操作系统。

针对于本实时控制系统需求而言，Windows CE .NET 所具有功能中满足的特点有：

一、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。

二、拥有良好的通信能力。广泛支持各种通信硬件，亦支持直接的局域连接以及拨号连接，并提供与 PC、内部网以及 Internet 的连接。

三、支持嵌套中断。允许更高优先级的中断首先得到响应，而不是等待低优先级的 ISR 完成。这使得该操作系统具有嵌入式操作系统所要求的实时性。

四、更好的线程响应能力。对高级别（中断服务线程）的响应时间上限的要求更加严格，在线程响应能力方面的改进帮助开发人员掌握线程转换的具体时间，并通过增强的监控能力和对硬件的控

制能力帮助他们创建新的嵌入式应用程序。同时，也提供了可靠的内核操作系统服务，用来支持设备所必不可少的要求最苛刻的实时嵌入式设计。例如，嵌入式开发人员现在可以通过严格的实时操作系统（RTOS）内核支持，实现低等待时间、有限的、决定性的系统性能。Windows CE .NET 平台的实时性能已经在许多工厂实现中得到了验证。

五、256 个优先级别。可以使开发人员在控制嵌入式系统的时序安排方面有更大的灵活性。

为了提高系统的可移植性，Windows CE .NET 采用硬件抽象层（Hardware Abstraction Layer，HAL）和开发板支持包（Board Support Package，BSP）的底层结构设计。HAL 提供了与设备无关的特性，屏蔽了不同平台硬件的差异，向操作系统的上层提供了一套统一的接口。HAL 隐藏了各种与硬件有关的细节，保证了整个系统的可移植性。而一般由硬件厂家提供的，按照给定的规范完成的 BSP，保证了嵌入式操作系统可以在新推出的微处理器硬件平台上运行。

4.2 基于 Windows CE .NET 的开发流程

基于 Windows CE .NET 的产品开发流程大致上可以分为 6 个阶段：硬件设计、选定 BSP、开发驱动程序、裁剪内核、生成 SDK、编写应用软件。

在系统开发中，首先需要为自己的产品选定相应的硬件平台，对于硬件的选择主要是根据产品所要实现的功能决定的。将选定相关的硬件把它们集成到你的产品中，也就是集成到我们常说的目标机（Target machine）里。目标机的基础平台可以自己研发也可以从硬件厂商直接购买，有了它我们就可以开始下面的工作了。在本课题里我们采用的是直接购买的方式。

对应用系统开发来说，系统迁移主要是实现 BSP，使得系统可以支持专门用于该系统的底板以及相关的设备。微软的 Platform Builder 自带了一些标准的 BSP。你也可以往 Platform Builder 里面导入其他的 BSP。部分 CPU 产品的供应商（如 SIS 公司）也提供相应的 BSP 开发包。

Platform Builder 虽然自带了一些标准的 BSP，里面也有一些

通用的驱动程序，例如，串口的驱动、USB 口的驱动，以及通用的总线驱动等，但是对于一些专用的设备，Platform Builder 并没有给出驱动程序，这个时候就需要单独的驱动程序。有些公司对于自己的产品定制了自己的 BSP 开发包，这时只需要导入他们提供的 BSP 开发包到 Platform Builder 中，然后添加相关的驱动程序和组件就可以了。如果厂家没有提供硬件的驱动程序，那么就需要自己开发相关的驱动程序。

Platform Builder 包含了 Windows CE .NET 的内核程序，并自带了很多 BSP。通过 Platform Builder 可以对 Windows CE .NET 的系统结构进行裁剪。对于一些不必要的系统组件可以不添加到定制的内核里面，而对于一些必须要添加到目标操作系统里面的组件，可以在这个阶段添加到内核里。

定制好自己的内核以后如果需要的话可以通过 Platform Builder 生成自己的 SDK，这样做的目的是为了方便编写应用程序。生成的 SDK 很容易安装到其他的编辑器上面。例如，可以把自己生成的 SDK 安装到 EVC 下面，这样当需要新建一个工程时，定制的 SDK 就会出现在选择画面中。

以上步骤已完成了 Windows CE .NET 操作系统平台的搭建，接下来的工作就是在这个舞台上进行应用软件的编写，并完成最终的产品开发。

4.2.1 Platform Builder 内核定制工具

把 Windows CE .NET 移植到新硬件平台由内核定制工具 Platform Builder 完成。Platform Builder 是为基于 Microsoft 的 Windows CE .NET 操作系统构建定制嵌入平台而提供的集成开发环境(IDE)。它包括了基本的文件编译能力、多目标编译调试工具包、特性编辑器、软件开发工具包 SDK(Software Development Kit)导出工具、向导工具以及用于做测试和性能分析等工作的多个工具包，开发人员可以在集成开发环境中完成操作系统镜像的定制、编译与调试工作，并可进行应用软件、驱动程序的开发。

Platform Builder 提供了示例的用于 OEM 适配层(OAL, OEM adoption layer)的源代码，OAL 是由 OEM 编写的用来使 Windows CE 适合特殊硬件的代码层。OAL 包括硬件抽象层(HAL)代码，它用来

支持 Windows CE 核心的需要以及支持内置硬件(如键盘、触摸屏和显示器)的驱动程序。它还提供了用于声音串行驱动程序以及 PCMCIA 控制器驱动程序的示例代码。Platform Builder 还包含底层调试工具。虽然这些工具的设计目标主要是用来帮助 Windows CE 转向新的硬件平台，但也能够在分析应用程序软件所带来的难题时使用它们。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 .NET 的定制设备所需的一切特性。

4.2.2 Platform Builder 内核定制过程

运行 Platform Builder 后在文件菜单中选择 New Platform，启动操作系统镜像的建立向导。在欢迎屏幕之后是选择开发板支持包。Windows CE .NET 支持目前流行的多款 CPU 系统，主要有 X86 系列、ARM 系列、MIPS 系列、SH3 系列、SH4 系列等，每一种系列除了相应的标准开发包外，还带有此系列中较为常用的具体 CPU 的开发包。如，X86 系列除了标准的 CEPC（对应所有的 X86 系列）开发包外，还带有我们经常使用的 NATIONAL GEODE:X86 开发包。我们可以同时选择 X86 的仿真方式（EMULATOR:X86）与 NATIONAL GEODE:X86 方式，进入下一步，选择基本配置结构。

在操作系统镜像的基本配置对话框中，有多个 Windows CE .NET 已经设定的基本配置结构供选择，如果设计人员设计的产品与列表中保存的基本配置一样，那么可以直接选择以完成基本的设定，节省时间。也可以选择自定义配置，完全由用户来完成配置工作。完成配置后，已经定义了一个平台，准备用于生成一个操作系统镜像。选择生成 Debug 版本或 Release 版本，分别用于内核调试和发布给用户使用。最后，用 Build Platform 即可生成操作系统镜像文件 NK.BIN。

4.3 Windows CE 下的编程开发工具

4.3.1 编程开发工具的选择

在多点计轴系统中，我们选择 Windows CE .NET 作为操作系统，编制在 Windows CE .NET 下运行的应用程序，可选择的编程

工具有：

- Embedded Visual C++
- Visual Studio .NET

其中，Embedded Visual C++是我们进行应用开发的首选。这是由于 Windows CE .NET 作为一种实时嵌入式系统应用于多点计轴系统中，嵌入式设备拥有的硬件资源较为有限，较慢的处理器和较少的存储器这样的硬件环境必然要求操作系统和应用软件尽可能地减少对系统资源的消耗，同时还需要保证很高的执行效率。在 Windows CE .NET 下调用 Windows CE API 函数，C 和 C++语言最为直接且高效。相比较用 Visual Studio .NET 开发工具软件在执行效率和资源节省程度都不如 Embedded Visual C++，而且 Windows CE .NET 内核中必须包含.NET 框架才能支持使用它编译的程序。我们在此使用的是 Embedded Visual C++ 4.0 sp4，支持 Windows CE .NET 4.2 内核下的应用开发。

4.3.2 Embedded Visual C++ 简介

1、 EVC 编程开发的特点

Windows CE .NET 下的 EVC 编程都是对特定目标硬件的编程，运行 Windows CE .NET 的机器通常比台式计算机的资源贫乏得多，所以编程时首先要明确目标硬件的特点和要求。这是 EVC 编程与 VC 编程的重要不同。

2、模拟器

EVC 编程环境提供了模拟器(Emulator)来模仿目标硬件进行调试。因为目标硬件的运行环境与台式机的运行环境绝大多数都是不同的，所以 EVC 编程无法象 VC 编程一样随时运行、调试。但 EVC 提供了与大多数硬件平台相似的模拟器，这样就方便了编程人员，可以直接在 EVC 下调试程序。一般 EVC 开发人员都是先在模拟器中将应用程序界面设计好，然后在加入对特定硬件操作的功能，到硬件平台上进行调试，这样就加快了开发速度。

3、WIN32 编程与 Windows CE .NET API

EVC 编程是 WIN32 编程。Windows CE .NET 支持 WIN32 API 中的绝大部分函数，但有些是不支持的，同时它又扩充了一些特定的 Windows CE .NET 函数，只在 Windows CE .NET 下可以使用。

比如命令条(Command Bar) API 等。

4、Unicode 环境

Windows CE .NET 象 Windows NT 一样，是 Unicode 环境。尽管 Windows CE .NET 支持 ASCII 功能来进行文件交换，但是 Windows CE .NET 的本地文本格式是 Unicode。

虽然 Embedded Visual C++与 Visual C++编程有许多不同之处，但它依旧是 Windows 编程，Windows 下 VC 开发人员所积累的对类、COM/ATL 的使用，程序的调试方法、开发流程等编程经验都可以用于 Embedded Visual C++编程。

第 5 章 多点计轴系统软件设计

5.1 软件系统需求分析

5.1.1 软件系统功能需求

通过分析，确定了多点计轴软件主要需实现以下功能：

1、主机通过计轴 485 总线向指定的计轴器下发读取轴数命令，对计轴器响应返回的数据进行校验、解包，获得轴数信息或计轴器错误信息。

2、主机通过 I / O 485 总线向指定的输入模块下发读取模块状态命令，对输入模块响应返回的数据进行校验；向指定的输出模块下发写模块状态命令，模块进行相应输出。

3、主机间通过局域网建立网络连接，将计轴复零数据、计轴器轴数、区段占用运算数据、轨道继电器回采数据等互传并比较，比较一致则作为有效数据。

4、根据各个区段的有效轴数信息进行区段运算，判断区间是否占用，并通过输出模块及电路控制相应区段轨道继电器的动作。

5、通过输入模块采集的控制台计轴复零信息，如为有效数据则对相应区段两端的计轴器轴数复位为零。

软件主要需要实现的功能如图 5-1 所示：

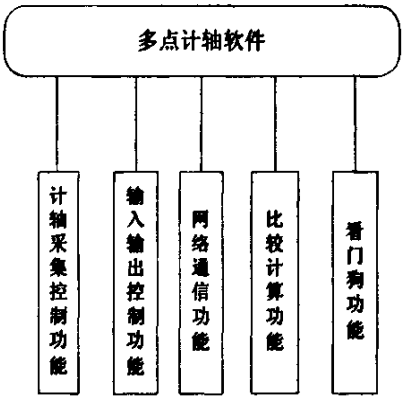

图 5-1 多点计轴软件功能框图

由于分析设计过程是个迭代的软件开发过程，所以需求也会在分析设计过程中不断补充、细化。上述需求在逐步的分析设计过程中，还有待不断的细化、完善。

5.1.2 使用用例描述需求

在需求分析阶段，可以使用 UML 中的用例来捕获用户需求，使用用例图来展示系统外部的各类执行者与系统提供的各种用例之间的关系。分析阶段主要关心问题域中的主要概念（如抽象、类和对象等）和机制，需要识别这些类以及它们相互间的关系，并用类图描述。

用例图显示了系统中的使用案例、参与者及其相互关系，它从用户的角度描述了系统所实现的功能以及每个功能的用例描述。每一个用例都是使用系统的一种方法，而且每一个用例都会产生不同的结果；而参与者是与系统交互的对象，它可以是一个使用系统的人，即用户，也可以是使用本系统的其他系统。

根据分析，可以识别出系统参与者有：Axle Counter（计轴器）、IO Module（输入/输出模块）、Network Device（网络通信设备）。

一、Axle Counter：计轴器负责采集通过的列车车辆的轴数信息、执行将计轴器内部记录的轴数置为零的操作。

二、IO Module：输入/输出模块负责采集轨道继电器的状态信息、计轴复零按钮的状态信息、输出轨道继电器的状态控制信息。

三、Network Device：网络设备负责在双机系统之间相互传输数据信息。

用例图从系统较高的层次描述了系统与外部执行者的交互，本系统中包括六个主要用例。

一、获取计轴器轴数信息（Get Counter Data）:计轴器响应系统主机的读取轴数命令，返回轴数值给主机，如果计轴器检测到本身有错误则返回错误信息；

二、计轴器复零（Clear Counter Data）：计轴器响应系统主机的计轴器复零命令，执行将计轴器内部记录的轴数置为零的操作，返回轴数值零给主机，如果计轴器检测到本身有错误则返回错误信息；

三、轨道继电器输出控制（GJ Output）：输出模块根据系统主机的输出控制命令，将输出通道置相应的数字量值。

四、获取轨道继电器状态（Get GJ Status）：输出模块响应系统主机的输入读取命令，将连接至输入通道相应的轨道继电器数字量状态返回给主机。

五、获取计轴复零按钮状态（Get JZFA Status）：输出模块响应系统主机的输入读取命令，将连接至输入通道相应的计轴复零按钮数字量状态返回给主机。

六、交换数据（Exchange Data）：系统主机通过网络将各自的轴数信息、计轴器复零状态信息、轨道继电器输出信息、轨道继电器状态信息等相互传送，用于相互比较。

包含系统主要用例的系统用例图如图 5-2 所示：

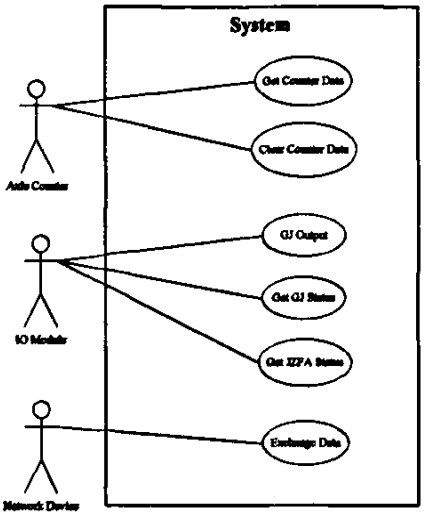

图 5-2 系统用例图

5.2 系统体系结构设计

5.2.1 系统对象类的识别

从需求分析和用例图中，可以识别出系统需要为计轴器、输入输出模块、网络设备的通信建立相应的类，实现与参与者的交互。在此定义类名称分别为计轴通信类 CCommCounter、模块通信类 CCommMdl、网络通信类 CCommNet。同时，我们还需要做进一步分析，得出系统中隐含的类。

一、计轴通信类和模块通信类都需要与主机串口通信交换数据，因此使用串口通信类封装使用串口资源、读写串口、关闭串口等功能。定义串口通信类名称为 CSerialComm。

二、由于区段占用的计算是使用该区段所包含计轴器的轴数作相应计算的，此处因此需要使用区段类来对区段的类型、所含计轴

器的信息、运算方法等进行封装。定义区段类名称为 CSec。

三、由计轴器用例的活动图分析，可以发现要对所有计轴器进行通信需要定义相应的主动对象，需要抽象出计轴控制类完成该功能。定义计轴处理类名称为 CCounterCtrl。

四、对所有输入输出模块的操作需要定义相应的主动对象，统一对模块通信类进行调用。定义 IO 控制类名称为 CIOCtrl。

五、对双机数据的校核需要使用主控运算类来完成各项数据的比较核对，对于计轴、输入输出、网络通信发生错误，也需要进行相应故障-安全处理。定义主控运算类名称为 CMainCtrl。

系统对象的类图如图 5-3 所示：

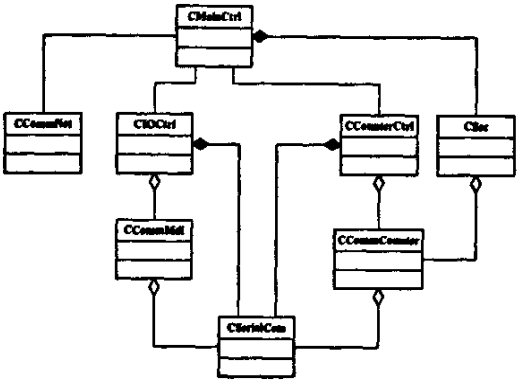

图 5-3 系统对象类图

其中，CMainCtrl 类要具备各区段信息，聚合了 CSec 类。CIOCtrl 类和 CCounterCtrl 类聚合了 CSerialComm 类。而 CCommMdl 类和 CCommCounter 类则包含对 CSerialComm 类的引用。

5.2.2 界面显示类的识别

多点计轴系统运行时无需界面显示，但显示界面可以给系统运行状况以直观的表示，也可以用于演示、调试等情况下使用。在本系统中加入了界面显示类来实现调试和演示，仅作简要描述。根据系统需求的分析，识别出本系统中所需的界面显示类：

1、CMyView 类：使用的是 MFC 类 CView 派生的视类，对应用程序提供文档--视图结构中的界面显示支持。

2、CDevice 类：作为界面显示的股道和道岔区段显示类的基类，定义其中共有的属性特征和动作。

3、CSection 类：股道显示类，用于在界面上显示股道及其状态，使用红色表示占用，白色表示空闲。

4、CSwitch 类：道岔显示类，用于在界面显示道岔及其状态。

5、CMyButton 类：派生自 MFC 类 CButton，用于在界面显示模拟的计轴复零按钮操作。

6、CMyEdit 类：派生自 MFC 类 CEdit，用于在界面模拟计轴器轴数变化操作。

7、COutputWnd 类：派生自 MFC 类 CWnd，用于在界面显示信息输出窗口。

界面显示对象类的类图如图 5-4 所示：

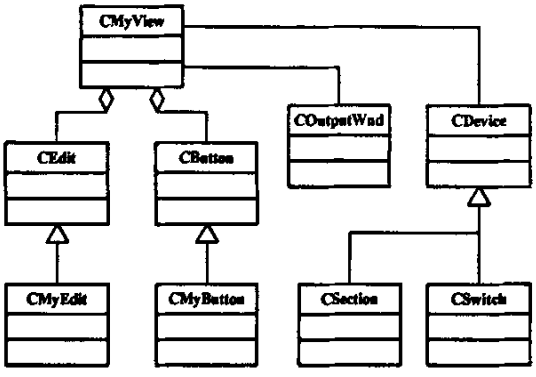

图 5-4 界面显示对象类图

整个系统的系统类图如图 5-5 所示：

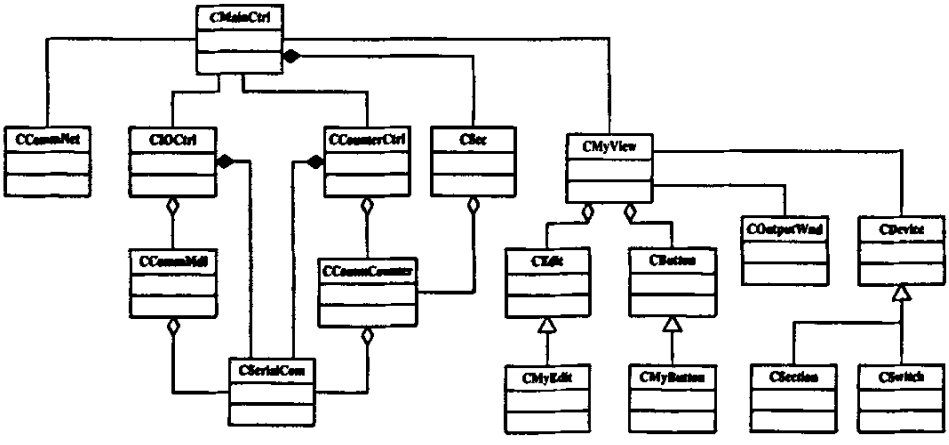

图 5-5 多点计轴系统类图

5.2.3 使用协作图描述对象

为了理解系统的动态行为，需要创建描述系统动态的图，如顺序图、协作图、状态图、活动图等。根据各系统的特性不同，应选用合适的图来表达系统动态行为。

所有的系统对象之间的动态行为，采用协作图来描述：

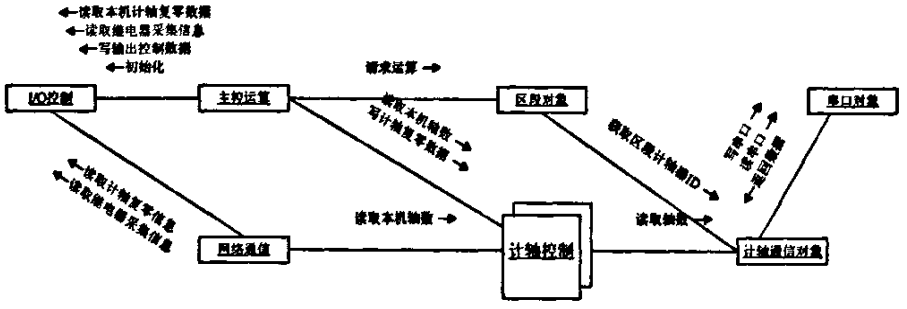

图 5-6 系统对象的协作图

各系统对象之间存在功能上的协作关系，共同完成系统各项任务。图中的主控运算、计轴控制、I/O 控制、网络通信对象均为主动对象，分别启动主控运算线程、计轴控制线程、I/O 控制线程、网络通信线程。另外，非系统对象产生的线程有界面显示对象产生的界面线程。多线程的设计在后续章节叙述。

系统包含的线程框图如图 5-7 所示：

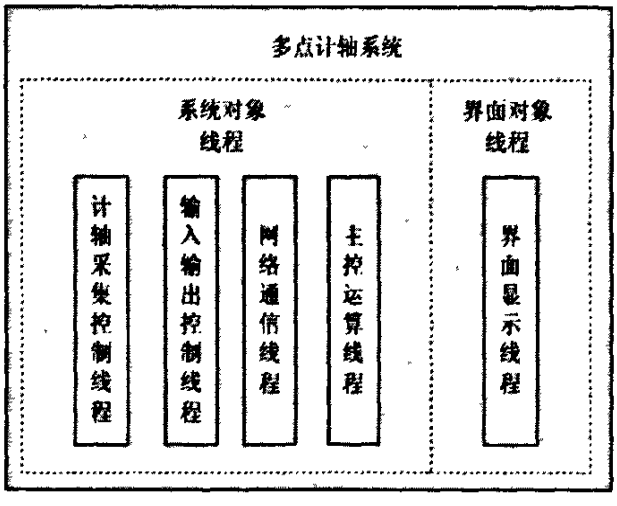

图 5-7 系统包含的线程框图

5.2.4 使用顺序图描述对象关系

顺序图显示对象之间的动态协作关系，它强调对象之间传递消息的时间顺序。顺序图可以用来进行一个场景的说明——即一个事务的处理过程。

本系统的计轴运算的动态行为，采用计轴运算顺序图来表示：

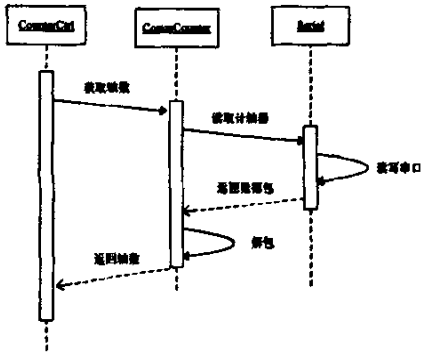

图 5-8 计轴运算顺序图

图 5-8 中，计轴通信控制类对象 CounterCtrl 向计轴通信类对象 Commcounter 发送获取轴数的消息，对象 Commcounter 随即向执行串口通信类对象 Serial 发送读取计轴器的消息，对象 Serial 进行实际的串口读写，获取得到的数据包返回给 CommCounter，CommCounter 校验检查得到轴数及错误信息，将轴数、信息返回给 CounterCtrl。

输入输出控制顺序图同计轴运算顺序图类似，由图中可清楚看出对象的执行顺序关系。在此不再说明：

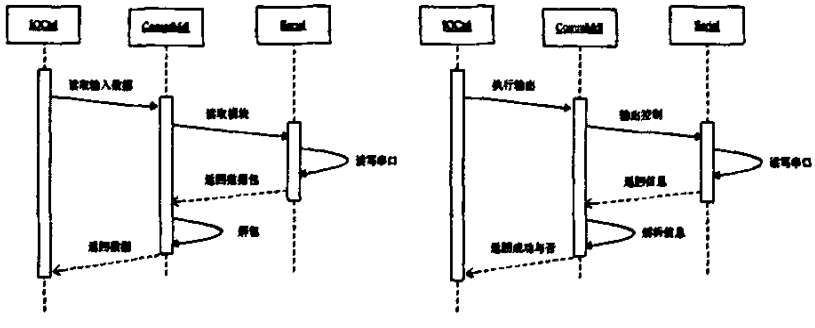

图 5-9 输入及输出控制顺序图

5.3 系统多线程设计

系统中，从各 I/O 设备、计轴设备获取得到的数据，需要网络通信交换后进行双机比较，而各 I/O 设备、计轴设备的速度远远低于网络通信速度以及主控运算速度。线程在一个封闭的地址空间内并存，通过这个共享的地址空间，线程能够实现更快的任务间通信。系统采用单进程多线程的设计，可以最大限度降低任务切换所带来的性能影响。系统中的需要运行的线程有：界面显示线程、计轴控制线程、I/O 控制线程、网络通信线程、主控运算线程。本系统中，除了界面显示线程外，其余线程均设计为工作线程，以后台方式完成各项任务。

5.3.1 多线程数据读写方案设计

多点计轴系统中，需要解决各线程共享的数据读写的安全性。

当计轴通信控制线程正在把获取到的轴数值的写入共享数据区时，如果主控运算线程正在从该位置读取数据进行相应的处理，在计轴通信控制线程写完数据之前，就把数据读出，那么它读到的数据必定是错误的。因此，就必须设计一种方法，使得在计轴通信控制线程写完数据后才允许其它线程读取该数据。

解决多线程同步的常用方法有 4 种：临界区、互斥体、信号量和事件。其中，除临界区外的三种均是 Win32 核心对象，可以在多个进程之间使用，临界区只存在一个进程的内存空间中，是运行效率最高的方法，不能在多个进程间使用。本系统要求有较高的运行速度，又因任何时刻都只需运行这一个应用进程，因此，临界区是最适合本系统的多线程同步解决方法。

根据以上分析，在系统设计时，把多线程共享数据结构设计为临界区变量，线程在读写共享数据前都要判断临界区是否可以进入，如果可以则进入临界区的所有权，在读写完共享数据后立即退出临界区，以避免其他读写线程长时间处于阻塞状态。

系统中对用于不同功能的各共享数据分别进行临界区变量设置。各共享数据主要有：

计轴轴数信息数据。计轴器的故障信息也包含在内。

计轴复零信息数据。包含计轴复零对应的模块故障信息。

继电器状态信息数据。包含输出模块的故障信息。

用于网络发送的本机信息数据和网络接收的邻机信息数据。本机信息数据是上述三种信息的集合。

5.3.2 线程优先级设计

Windows CE 从 3.0 版本开始，支持 0～255 共 256 个优先级，其中 0 优先级最高，255 优先级最低。在同一进程中可以使用所有 256 个优先级，每个线程都有自己的优先级。0～248 的优先级属于实时性优先级。

本系统中，IO 控制线程对输出模块进行控制需要最高优先级，然而输入模块所需优先级远不必如输出模块那么高。界面显示线程无需以实时优先级运行。计轴控制、网络通信、主控运算线程也需要使用实时优先级，它们之间需要共享资源，可能在临界区阻塞。如果设置的优先级不合适，可能会造成优先级的反转。由于 Windows CE 仅支持一级优先级的反转，因此，不合适的优先级别设置可能会导致死锁。

由于各工作线程需要紧密合作，它们的优先级别基本相同。因此，设置 I/O 控制、计轴控制、网络通信、主控运算线程优先级为 248，显示界面采用默认线程优先级 251。

Windows CE 支持线程的优先级设置，因此，可以在需要快速响应轨道继电器由吸起控制其落下的情况下，升高 I/O 控制线程优先级别。

5.3.3 各工作线程的流程设计

系统工作线程的设计思路依据的重点有：

1、尽量减少线程在临界区内运行的时间，即尽量短的时间锁住共享资源。

在设计中，实现这一原则的方法有：

对提供各项数据的共享数据块分别建立临界区，避免了对不必使用的共享数据的锁定带来无谓的阻塞延迟。

对各共享数据块的读取采用整体读出的方式，对于网络通信传送的邻机数据等数据块，也采用整体写入的方式，不必在每次读写少量数据时即需要进入临界区。最大程度减少阻塞发生的可能性。

2、对于需要进行 485 总线通信的线程，为减少对关键数据的

变化的响应时间。采取相应的算法来进行处理。

为能设计相应的算法来提高响应速度，需要对各实时任务进行进一步分析。在下面的章节中，将对这项内容进行描述。

根据以上思想，结合功能需求，分析得出系统中的各工作线程的流程如下所示：

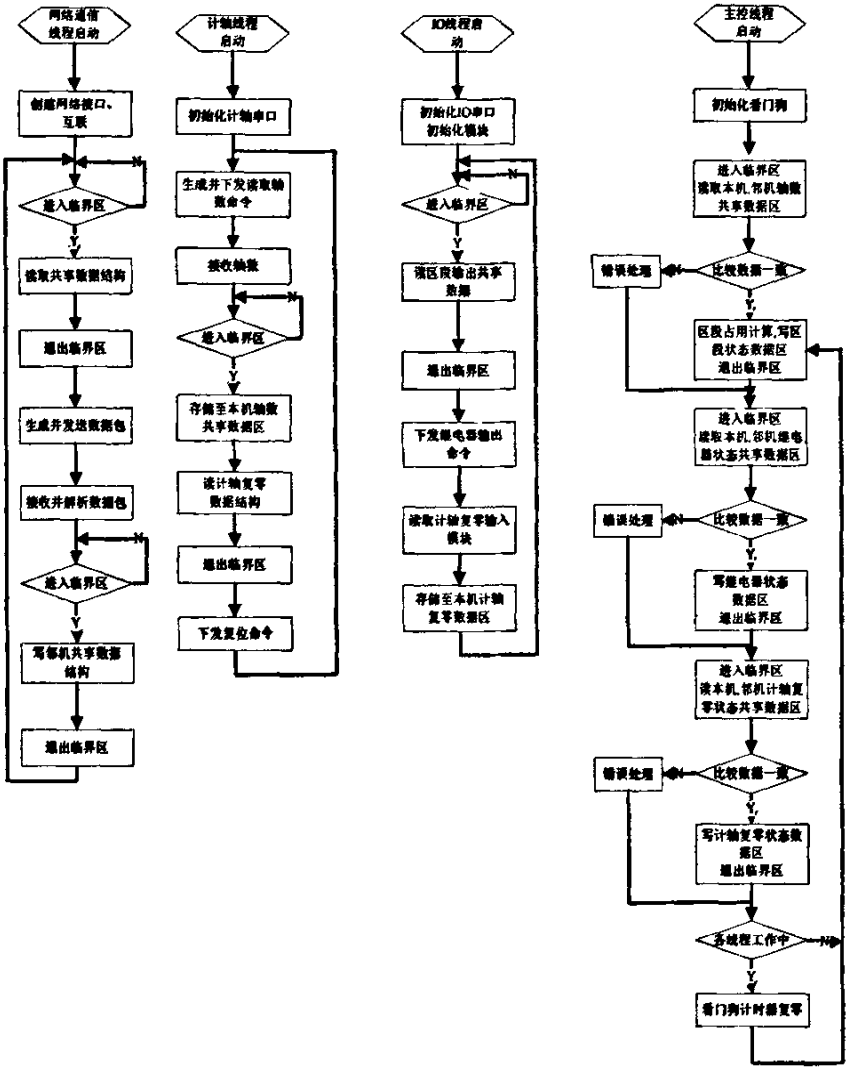

图 5-10 工作线程流程图

网络通信线程在收发本机和邻机数据之前，需要进入临界区对本机的共享数据结构进行读操作，此时主控运算线程无法对该资源进行写入；读取完毕立即退出该临界区。然后将收到的邻机数据包解包后，进入另一临界区，对各邻机共享数据结构进行写操作，此时避免了主控运算线程错误读数据的可能。

计轴控制线程需要操作的共享资源是本机轴数共享区和计轴复零数据区。主控运算线程分别对这两资源进行读取和写入。由于取得一次轴数的时间耗费较长，故每次取得轴数后即进入临界区写本机轴数共享区。

I/O 控制线程对区段输出共享数据和计轴复零数据进行操作，两资源也被主控计算线程访问。

主控线程对网络、计轴、I/O 线程进行数据交换，完成系统的各项功能。

5.3.4 改善实时性的软件算法

计轴区段的空闲状态转占用状态，最终引起区段轨道继电器由吸起转落下状态，属于最需改善实时性的问题。

通过对计轴区段的动作分析，我们能得出这样一个事实依据：

1、计轴区段由空闲转占用，必定是区段内开始无车，此时车辆经过区段内某一个计轴器，而且仅该计轴器轴数发生变化。

2、推理可以得到：区段如果开始是空闲状态，则其内仅一个计轴器轴数变化即可判定为有车占用。

3、算法流程思路：

设置一优先输出队列共享资源，I/O 线程中读取到有优先输出请求（非空），优先执行该输出。

主控线程保留上一次的轴数值，用于比较轴数的变化。如检测到空闲区段的一个计轴器轴数发生变化，即写优先输出队列共享资源，并提高 I/O 线程的优先级。

由流程可以看出，当主控中任一个空闲的区段内，计轴器轴数变化，主控线程提高 I/O 线程优先级后，I/O 线程迅速抢先，取得该区段相应信息，在很短时间内就执行了区段输出。

为简化流程描述，下图中访问的共享资源都是进入临界区访问然后退出的，但没有对每个操作对进行绘制。

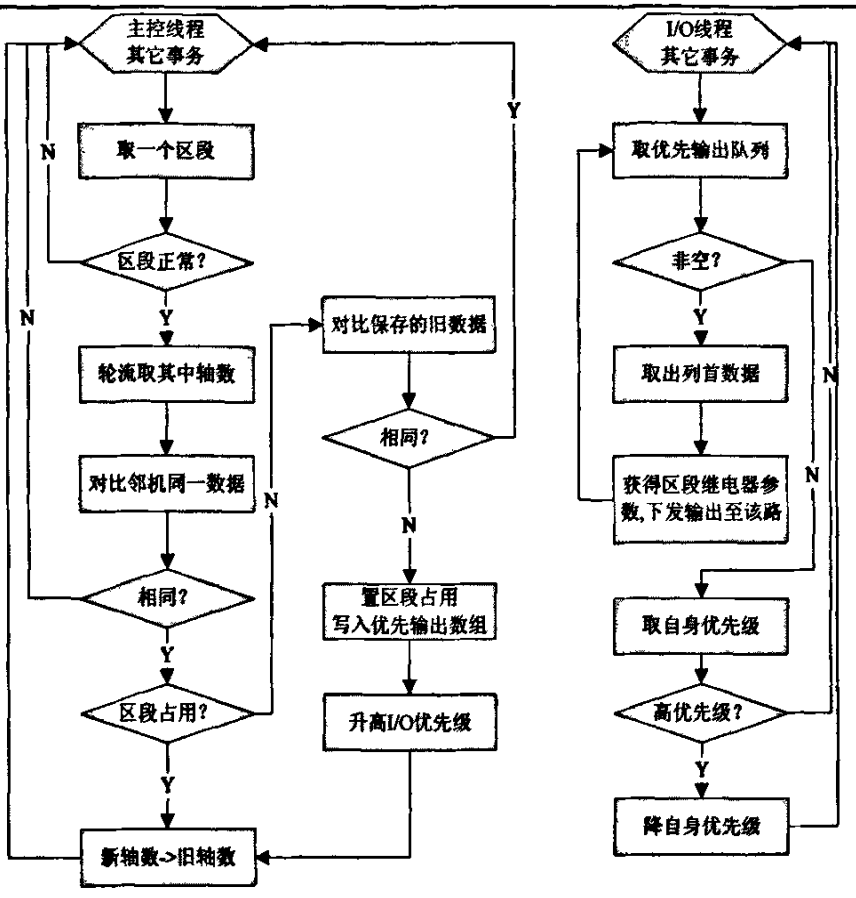

图 5-11 主控、I/O 实时性算法设计流程图

5.4 系统故障-安全特性设计

5.4.1 二取二计轴设备故障-安全特性设计

1、计轴设备硬件故障

当某一区段各个终端的计轴器其中之一发生故障时，便无法判别该区段是否出清或者占用。此时主机将判断得出该计轴器故障。

按照故障安全导向原则，应将轨道继电器状态导向安全侧，即轨道继电器落下状态。

系统中设计两主机均立即停止对该区段轨道继电器控制电路的输出。双主机的控制电路以串联方式连接，只有两台主机的均有输出时，才构成回路，能使轨道继电器吸起。此时只要一台主机无输出，闭环电路被断开，该轨道继电器即落下，保障系统安全。同

时进行相应报警。

2、主机取得计轴器轴数信息不一致

当两台主机取得计轴器轴数信息不一致时，进行通信后比较双机数据不同，在超出容许限度后仍是如此，将其所属区段轨道继电器状态导向安全侧。

系统中设计双机均对该计轴器所属区段置区段占用状态。I/O控制部分以该状态对该路进行控制，即停止输出。轨道继电器落下，保障系统安全。同时进行相应报警。

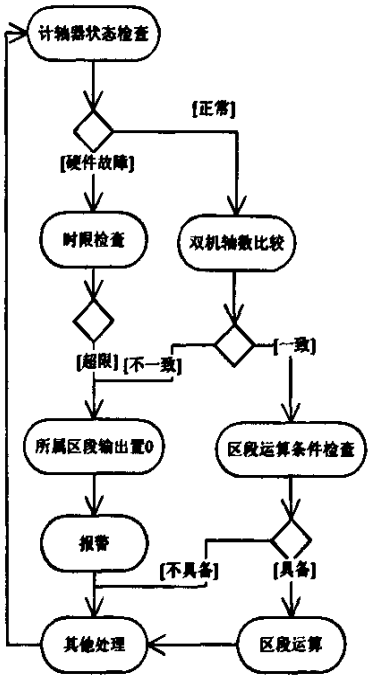

图 5-12 计轴设备故障处理活动图

5.4.2 二取二输入输出故障-安全特性设计

1、输入模块故障

输入模块获取 GJ 的回采状态或者获取计轴复零按钮的状态，如果本主机某一路输入模块发生故障，两台主机采集到的状态信息将不一致。

主机之间通信并比较采集的状态信息结果不一致，在超出容许限度后仍是如此，由于采集状态无法改变，因此采取保持现有状态作为安全侧。同时主机进行相应报警。

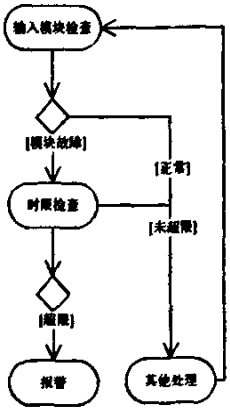

图 5-13 输入模块故障处理活动图

2、输出模块故障

输出模块部分硬件电路设计上，按照安全侧对应原则设计，要求必须双机同时输出逻辑状态 1，才能使驱动该路的 GJ 的回路接通，否则该路 GJ 保持落下状态。

如果本主机某一路输出模块发生故障，本主机无法对该模块进行控制，模块的输出状态是未知的。此时需将属于该模块控制的轨道继电器状态导向安全侧。

通过网络通信，另一主机可获知该模块状态，并进行导向安全侧的操作。

系统中设计由另一主机对其控制的该路模块停止输出，使模块所属轨道继电器均落下，保障系统安全。同时主机进行相应报警。

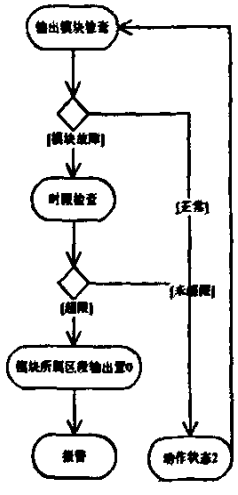

图 5-14 输出模块故障处理活动图

5.4.3 二取二网络通信故障-安全特性设计

网络通信的故障可能是由于网络通道损坏造成主机之间无法正常通信，也可能是由于某一台主机发生故障，系统未正常运行所致。

这两种情况下，正常运行中的主机无法进行双机数据相互比较，系统无法继续正常运行。

系统中设计在超出容许限度之后，对所有输出模块执行停止输出操作，使全站所有使用计轴器的区段的 GDJ 落下，确保导向故障-安全。同时，报警处理。

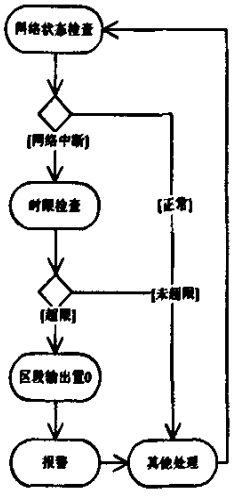

图 5-15 网络通信故障处理活动图

5.5 系统具体开发实现

5.5.1 系统中的主要数据结构

1、计轴器的数据信息结构体：

```
typedef struct STRUCTCNTSTATE_tag {
  int nCpu1Num;
  int nCpu2Num;
  int nCntSt;
  int JZ_CPU1_STATUS, JZ_CPU2_STATUS;
  int nCntFailed;
  int nCntReset;
```

```
}STRUCTCNTSTATE;
```

说明：nCpu1Num、nCpu2Num 表示计轴器的两个 CPU 所保存的轴数；nCntSt 表示计轴器的状态，有正常状态、复零状态和故障状态；JZ_CPU1_STATUS、JZ_CPU2_STATUS 表示计轴器内部 CPU 的状态，有 CPU 故障和正常状态；nCntFailed 记录与计轴器通信发生错误的类型，有 CRC 校验错、信息不完整等；nCntReset 表示计轴器如需复零时的对复用的哪个部分复零。

2、计轴器的参数信息结构体：

```
typedef struct STRUCTCNTPARAMS_tag {
unsigned short nCntID,  nLeftSec,  nRightSec;
BYTE byteType;
BYTE byteAddr;   //计轴器的地址号
int  x, y;
}STRUCTCNTPARAMS;
```

说明：nCntID、nLeftSec 、nRightSec 表示计轴器的编号、左侧区段编号、右侧区段编号；byteType 计轴器的使用类别，分为复用还是单用；byteAddr; 表示计轴器的地址号；x、y 是界面模拟显示计轴轴数输入框的坐标。

3、计轴区段的状态信息结构体：

```
typedef struct STRUCTSECSTATE_tag{
BOOL GJ;
unsigned short nCounter[4];
unsigned short nInAdd;//reset button address
unsigned short nOutAdd;//GDJ address
unsigned short nBackAdd;//feed back from GDJ
} STRUCTSECSTATE;
```

说明：GJ 表示轨道继电器的本机运算结果，有吸起和落下两种状态；nCounter[4]表示本区段内的所有计轴器的 ID 号，最多有 4 个；nInAdd、nOutAdd、nBackAdd 分别表示本区段的计轴复零按钮所在模块地址、轨道继电器输出所在模块地址及继电器状态采集所在模块地址。

输出模块的状态信息结构体：

```
typedef struct MODELOUTPUT_tag{
BYTE head;
```

```
BYTE addr[2];
BYTE chanelNo[2];
BYTE outData[4];
BYTE cr;
}STRUCTMODELOUTPUT;
```

说明：结构体内各变量分别表示输出模块命令的字节起始字节、2 字节的地址表示、2 字节的通道表示、4 字节的输出数据、结束字节。

4、输入模块的状态信息结构体：

```
typedef struct MODELINPUT_tag{
BYTE head;
BYTE addr[2];
BYTE command;
BYTE cr;
}STRUCTMODELINPUT;
```

说明：结构体内各变量分别表示输入模块命令的字节起始字节、2 字节的地址表示、命令字节、结束字节。

5.5.2 系统中的主要类定义

1、类 CSerialComm：

(1) 属性：

```
private:
HANDLE m_hComm;  //串口资源句柄
```

(2)操作：

BOOL Init485CommPort（int port，int type）；//打开指定串口资源，并按照实际需要配置串口参数

DWORD ReadComm（char *buf，DWORD dwLength）；//从串口缓冲区读取指定数量的字符，并返回实际读取的字符数量

BOOL WriteComm（char *buf，DWORD dwLength）；//将指定数量的字符从串口输出，并返回是否输出成功

BOOL CloseComm（）；　　//释放串口资源，并返回是否成功

2、类 CCommCounter：

(1) 属性：

private:

STRUCTCNTSTATE m_cntState;//该数据结构包含计轴器硬件相关信息

STRUCTCNTPARAMS m_cntParams; //该数据结构包含计轴器软件相关参数

BYTE m_cmdToCnt[5]，m_recvFromCnt[10]; //发往计轴器命令字和接收自计轴器的数据

BYTE recvFromCpu[2][10];

public:

int m_cntNum[2];//记录计轴器上下行区段轴数

CSerialComm *m_pCommCnt;//指向计轴设备通信使用的串口类实例的指针

COutputWnd* m_pOutWnd; //界面输出窗口类指针

(2) 操作:

int InitCntComm（int port）; //包装成员类变量 m_comCnt 的初始化计轴串口操作

BOOL SetCounterParams(STRUCTCNTPARAMS param);

BYTE GetRand（）; //生成随机数，用于参与计算生成 CRC 校验码

BYTE ComputeCRC（BYTE DATA[5]）; //计算 CRC 校验码，发送计轴命令的校验

int GetAxels（）; //从指定的 ID 的计轴器类实例取得轴数信息

int JudgeCPUAxes（int cpu，BYTE data[10]，unsigned short cntAdd）;//计轴器信息帧校验

void ErrCounterInfo（BYTE err）; //计轴器故障类型显示到界面的输出窗口

void ErrHandle（CString str）; //输出自定义字符串至界面的输出窗口

3、类 CCounterCtrl:

(1)属性:

public:

int m_nAxes[MAXCOUNTER+1][2]; //记录当前的轴数

int m_nCntCommFailed[100];//计轴器通信失败次数

CCommCounter * m_pCommCounter;// 指向计轴通信模块功能封装的类变量实例指针

CWinThread *m_pCounterThread;//指向计轴控制线程的指针

(2)操作:

int CntComm（unsigned short cntID）； //与指定 ID 的计轴器通信收发信息；包括按要求做复零或读轴数

4、类 CSec

(1) 属性:

private:

unsigned short m_nType;//区段类型，无岔、一送双受、一送三受

unsigned short m_nSecID;

CString m_strName;

STRUCTSECSTATE m_SecState;//该数据结构包含区段所需的各状态信息

(2) 操作:

unsigned short GetType();

unsigned short GetID();

int SetStatus（int st）;//写区段状态到本机共享数据区

int Compare（）；

5、类 CCommMdl:

(1)属性:

CSerialComm *m_pCommIO;//与 I/O 设备通信使用的串口类实例

STRUCTMODELOUTPUT ModelOutput;// 结构体成员变量数组，用于存放模块输出命令

STRUCTMODELINPUT ModelInput;// 结构体成员变量数组，用于存放模块输入命令

BYTE m_cmdOutToModel[8]，m_OutRecvFromModel[2];// 发往输出模块命令字和返回自输出模块的数据

BYTE m_cmdInToModel[5]，m_InRecvFromModel[8];// 发往输入模块命令字和返回自输入模块的数据

(2) 操作:

UINT InitIOComm（int port）；//该操作封装串口类的初始化相

应串口的功能，打开指定串口资源，并按照通信模块所需配置串口参数

int InitInCommMdl（int mdlID）； //以输入模块地址为参数，生成读数据命令

int InitOutCommMdl（int mdlID）； //以输出模块地址为参数，生成写数据命令

void MakeOutData（）； //生成输出控制数据的命令字

int CheckRecvFromMdl(BYTE *data, int nBytes, unsigned short mdlID）；//检查读取的数据格式，如正确则保存数据

6、类 CIOCtrl：

(1) 属性：

CCommMdl * m_pCommMdl; //输入输出模块硬件功能封装的类变量实例指针

CWinThread *m_pIOThread;//指向 IO 控制线程的指针

STRUCTMODELOUTPUT ModelOutput[MAXOUTMODEL];//结构体成员变量数组，用于存放模块输出命令

STRUCTMODELINPUT ModelInput[MAXINMODEL];// 结构体成员变量数组，用于存放模块输入命令

BOOL m_bFlash; //用以表示是否动态输出的标志

(2) 操作：

int SInComm（unsigned short mdlID，unsigned short firstMdl）；//对输入模块进行通信，获取各输入模块的信息

int DOutComm（unsigned short ModelID，bool bDynaOut）；//对输出模块进行通信，指定各输出模块的输出状态信息，以及是否是动态输出

7、类 CMainCtrl：

(1) 属性：

private：

CSec m_Sec[MAXSEC];//区段类

public：

CWinThread *m_pMainThread;//指向主控运算线程的指针

(2)操作：

InitWDT();//启用看门狗

ResetWDT();//看门狗计时器复零

CompareCounterNum();//比较双机轴数
CompareRelayStatus();//比较双机继电器状态数据
CompareCounterReseter();//比较双机计轴复零数据

5.5.3 Platform Builder 中的主要定制工作

在本系统的 Windows CE 操作系统定制中，使用的硬件功能在 PB 工具中已经具备。因此主要的工作就是对内核进行裁剪定制。

PCM-5825 采用的是 Geode CPU。因此，在 Platform Builder 的新建工程向导中，在 Step 2，选择常用开发包中的 NATIONAL GEODE:X86 和 EMULATOR:X86 两个开发包。分别由于生成适用于本系统的内核镜像和模拟器的内核镜像。在 Step 3，可以选择与需要定制的配置相近的可用配置（Available configurations），然后在工程向导完成后进行可选特性的添加、删除。以下对本系统中必须添加的特性予以简要说明：

1、加入硬件设备支持。

以系统中使用的显示驱动为例，在 IDE 的 Catalog 面板中选择 Device Drivers 下的 Display，展开后选择 Geode/MediaGX 后，加入平台特性中。

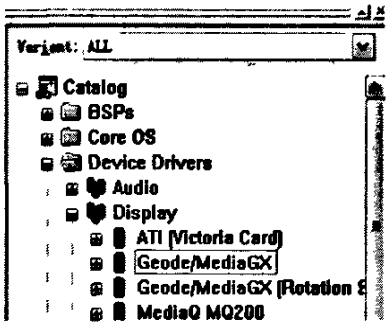

图 5-16 IDE 的 Catalog 面板

在系统中需要加入的驱动支持有：

(1) 加入显示驱动支持。
(2) 加入网络设备驱动支持。
(3) 加入串口驱动支持。
(4) 加入存储设备驱动支持。

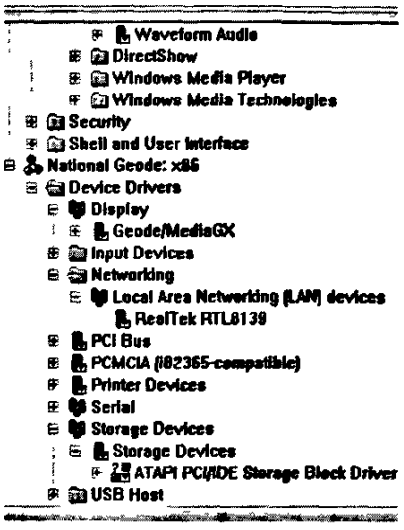

图 5-17 IDE 的 Workspace 面板中已加入的特性

2、加入系统功能支持

(1) 加入 802.3 网络功能支持。

(2) 加入 TFAT 文件系统支持。

TFAT 文件系统是 Windows CE 4.0 版本以后提供的一种新的 FAT 文件系统，称为交易安全的 FAT 文件系统（Transcation-Safe FAT File System，简称 TFATFS 或 TFAT）。交易的概念来自数据库，它是一种保证操作一定完成的一种机制。而 FAT 文件系统不是一个交易安全的文件系统。如果在一个交易中间写操作被中断，FAT 文件系统可能崩溃。

TFAT 文件系统驱动是 FAT 文件系统驱动的超集。TFAT 既支持 TFAT 卷又支持 FAT 卷。因此系统中只需包括 TFAT 驱动即可。

(3) 加入 wav 音频格式回放支持。

3、加入多点计轴软件开机自启动功能。

(1) 要自动加载自己的应用程序作为设备启动界面需要在操作系统启动时 加载该文件。一般放在注册表的键[HKEY_LOCAL_MACHINE\init]下面。例如:

```
[HKEY_LOCAL_MACHINE\init]
"Launch10"="shell.exe"
"Launch20"="device.exe"
"Launch30"="gwes.exe"
"Depend30"="hex:14,00"
```

上述注册表入口设置规定内核在启动时必须自动运行 shell.EXE 和 device.EXE 模块，gwes.EXE 模块必须在 device.EXE 正常启动以后才能运行。添加：

"Launch60"="\\硬盘\\MyProject.exe"

"Depend60"="hex:1E,00"

即可实现应用程序在 Windows CE 系统启动后的自启动。

(2)实现自启动，首先 CE 系统必须能自启动。将 CF 电子盘格式化并加入 DOS 操作系统，将生成的内核镜像文件 NK.BIN 和 DOS 系统自带的文件 Himem.sys 拷贝到电子盘上，建立 AUTOEXEC.BAT 文件，加入如下一行：

LoadCEPC NK.BIN /L:800x600x16

建立 CONFIG.SYS，加入：

dos=high,umb

device=himem.sys /testmem:OFF

即实现了 Windows CE 操作系统的自启动。

5.5.4 系统运行中的界面显示

系统界面显示的是示意的站场图形，股道和道岔在被占用时，显示为红色，空闲时为白色。

在各区段下方均有两个 CMyEdit 控件，供无计轴器时模拟轴数变化使用。如某个计轴点复用，上方的 CMyEdit 控件用于表示计轴点左侧的计轴轴数值。下方的控件则表示右侧的计轴轴数值。

界面上方的按钮用于模拟计轴复零按钮动作，当按钮被按下，相应的本机该区段计轴复零状态值被置为 1。

界面下方的两个文本显示框分别用于显示计轴器故障信息和运算比较错误信息等。

在双机启动后，界面显示如下：

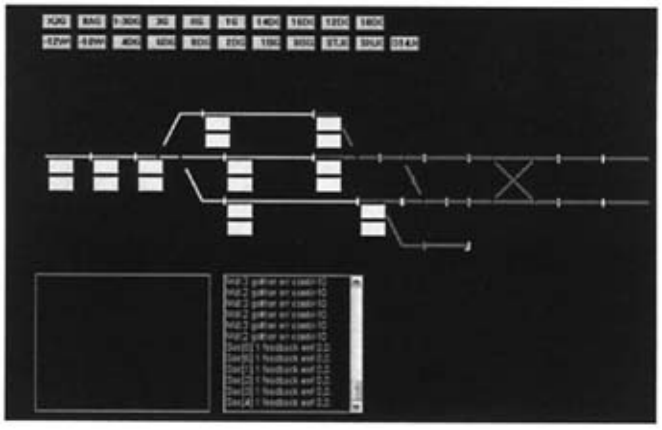

图 5-18 启动界面

在模拟条件下，设置了六个区段数据，分别为 XJG、IIAG、1-3DG、3G、IIG、1G。由于这时候取自轴数编辑框的轴数值均为0，这六个区段经过运算并且双机比较一致，输出状态为区段空闲。

模拟一台主机死机的情况：在关闭一台主机后，由于网络通信中断，根据故障-安全导向原则，所有区段的轨道继电器均落下，各区段均以红色显示。

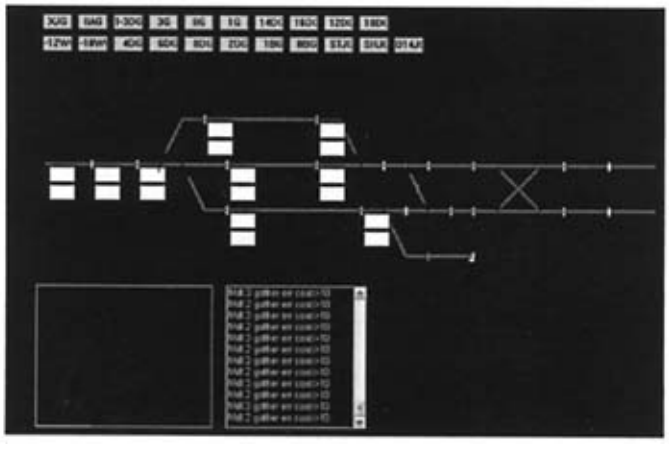

图 5-19 单机运行界面

模拟计轴器轴数变化的情况：在一个计轴区段的左下输入框、右上输入框输入轴数，当运算结果为区段占用时，该区段以红色显示；运算结果为区段空闲时该区段以白色显示。

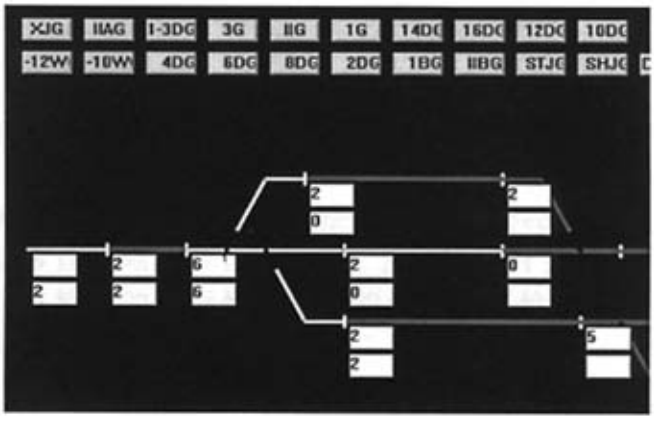
XJG
IIAG
I-3DG
3G
IIG
1G
14DG
16DG
12DG
10DG
-12W
-10W
4DG
6DG
8DG
2DG
1BG
IIBG
STJG
SHJG

图 5-20 模拟轴数界面

结 论

本文通过对多点计轴系统的分析，设计了多点计轴系统总体结构，二取二的硬件方案，并在以 Windows CE .NET 为操作系统平台的基础上，使用面向对象的方法，采用 Embedded Visual C++编程开发实现多点计轴软件。本文实现了一种站内多点计轴的总体解决方案，具有较强的可行性、较高的安全性、价格成本相对较低，对实际应用具有积极的意义。

在整个系统的设计开发过程中，我主要完成了以下几方面的工作：

- ✧ 进行系统的需求和关键问题分析；
- ✧ 进行系统总体结构方案设计和硬件结构方案设计；
- ✧ 进行系统软件总体设计，各功能模块的详细设计，并代码实现；
- ✧ 构建了实验室环境下的最小化多点计轴系统，通过计轴器模拟的方式完成了系统单元测试和整体测试。

目前，系统在成都铁路通信工厂的计轴实验测试车间进行了连接 4 个计轴点的测试和调试工作，系统运行正常。大量计轴点和实际站场的测试环境不容易建立，因此下一步需要使用更多的仿真手段对多点计轴系统进行调试和测试工作。

本文的工作还存在一些不足的地方需要完善和改进：

1、系统在的实时性算法方面还有很多可以研究的地方，如采用动态的队列轮询计轴设备的方式进行调度、进一步研究通过复用计轴区段周边的轨道电路信息进行计轴轴数变化预测等。

2、研究过程中限于现有条件，采用的计轴设备是以 485 总线进行通信。但是由于 485 总线的单主机多从机结构，计轴设备不能在轴数发生变化后主动发起通信，存在通信速度相对较低、响应速度慢、灵活性不够等不足之处。在今后可以根据条件采取如 CAN 总线之类的现场总线方式对设计进行更新，必然能进一步提高系统的性能。

3、系统的实现是以 2 取 2 方式实现的，如果能够实现 2 乘 2 取 2 方式，则系统的可靠性将能够进一步提高，在今后的研究中，可以考虑 2 乘 2 取 2 方式的系统设计和实现。

致 谢

首先感谢我的导师郭进教授，在学校期间，郭老师在生活上给予了我关怀和帮助，在学习上给我提供良好的学习环境，不仅教给我科研工作方法，而且教导我勤奋严谨和实事求是的工作作风，使我终生受益。

感谢成都通信设备厂谢玉琼高工等人给予的协助，正是由于他们的协助，使得系统与实际计轴设备的联调得以顺利实施。

我还要感谢杨扬老师、刘利芳师姐、及同实验室的同学、学弟学妹们，感谢你们给予我的帮助，在研究生的求学过程中，有你们才有了更多的精彩。

特别感谢生我养我的父母，感谢他们为我的成长所做的一切。

参考文献

[1] 付军等. 计轴技术的发展及容错型计轴设备的应用. 铁道通信信号,2003 年第 12 期

[2] 成都通信设备厂. JZ1-G 型计轴设备使用说明书. 2000 年 8 月

[3] 阎学文 王世军. 用计轴设备消灭“站内轨道电路分路不良”现象. 京铁科技通讯：太原刊, 2003 年 1 期

[4] 钟小伟. 西门子 AzSM350M 型微机计轴设备系统. 都市快轨交通, 2004 年增刊 第 17 卷

[5] 冯启建 林震. 实时计轴控制系统的设计与分析. 微计算机信息, 2006 年 第 22 卷 第 6-1 期

[6] 杨世武 王晓冰. 适于电气化铁道的微机计轴通信子系统的研究. 北方交通大学学报, 1994 年 第 18 卷 第 2 期

[7] 方志刚等. 基于局域网的微机联锁系统研究. 铁道学报 1996 年 12 月

[8] 郑丽英 董显等. 车站电气集中计算机联锁控制系统研究与开发 自动化与仪器仪表, 2005 年第 1 期

[9] 刘名元 郭进. 铁路站间自动闭塞信息传输和热备的研究. 铁路计算机, 2005 年第 14 卷第 1 期

[10] 陈文赛. 一种高可靠、高安全性系统-三取二计算机系统. 现代雷达, 2004 年 第 26 卷 第 6 期

[11] 渡边 郁夫 等. 可靠的铁道信号系统. 变流技术与电力牵引, 2005 年第 1 期

[12] 赵志熙. 车站信号控制系统. 中国铁道出版社, 1995

[13] 徐洪泽 岳强等. 车站信号计算机联锁控制系统—原理与应用 中国铁道出版社, 2005

[14] 阳宪惠. 现场总线技术及其应用. 清华大学出版社, 2002

[15] Lubomir F.Bic Alan C.Shaw 著. 操作系统原理. 梁洪亮等译. 清华大学出版社, 2005

[16] 李成忠 张新有等. 计算机网络原理与设计(第二版). 西南交通大学出版社, 1995

[17] 郑宗汉. 实时系统软件基础. 清华大学出版社, 2003

[18] 桑楠. 嵌入式系统原理及应用开发技术. 北京航空航天大学

出版社, 2002
[19] 舒忠海. 实时系统中任务可调度性研究. 计算机应用, 2001 年第 1 期
[20] Windows CE.NET 的新增特性. http://www.microsoft.com/china/windows/ embedded/ce.net/evaluation/whatnew, 2003
[21] 姜山 程君实. Windows CE 的实时性分析. 测控技术, 2000 年第 19 期第 1 卷
[22] 刘大鹏 马孝江. 基于 Windows CE 的嵌入式操作系统实时性分析. 计算机应用, 2002 年第 1 期
[23] 周毓林 宁杨等. Windows CE .net 内核定制及应用开发. 电子工业出版社, 2004
[24] 张东泉 谭南林等. Windows CE 实用开发技术. 电子工业出版社, 2006
[25] Douglas Boling. Windows CE 程序设计. 北京大学出版社, 1999
[26] 叶宏材, 陈峙榄. Windows CE.NET 嵌入式工业用控制器及自动控制系统设计. 清华大学出版社, 2005
[27] 金华标 常勇. 基于 Windows CE .NET 的嵌入式系统软件开发的研究. 武汉理工大学学报, 2003 年 第 27 卷 第 2 期
[28] 陈正茂 杨维忠 胡波. 基于 Windows CE. net 平台的嵌入式系统的定制和裁剪. 微型机与应用, 2004 年第 4 期
[29] 张新房等. 使用 Platform Builder 配置 Windows CE 操作系统. 单片机与嵌入式系统应用, 2002 年第 10 期
[30] Micbael Barr. C/C++嵌入式系统编程. 中国电力出版社, 2001
[31] 汪兵 李存斌等. EVC 高级编程及其应用开发. 中国水利水电出版社, 2005
[32] 傅曦 齐宇. 嵌入式系统 Windows CE 开发技巧与实例. 化学工业出版社, 2004
[33] 微软公司. Windows CE 程序员指南. 北京希望电子出版社, 1999
[34] Jim Beveridge, Robert Wiener. Win32 多线程程序设计. 侯俊杰译. 华中科技大学出版社, 2002
[35] Jeff Prosise. MFC Windows 程序设计. 清华大学出版社, 2001
[36] 郑人杰 殷人昆. 实用软件工程. 清华大学出版社, 1997

[37] Grady Booch, James Rumbaugh, Ivar Jacobson. The Unified Modeling Language User Guide. Addison Wesley Longman, 1999

[38] Ivar Jacobson, Grady Booch, James Rumbaugh. The Unified Software Development Process. Addison Wesley Longman, 1999

[39] B. P. Douglass. 嵌入式与实时系统开发—使用 UML、对象技术、框架与模式. 柳翔译. 机械工业出版社, 2005

[40] Perdita Stevens, Rob Pooley. 使用 UML—关于对象和组件软件工程. 包晓露等译. 人民邮电出版社, 2003

[41] Martin Fowler. UML distilled: a brief guide to the standard object modeling language(2nd Edition), Addison Wesley, 2000

[42] Grady Booch. Object-Oriented Analysis and Design, Benjamin/ Cummings, 1994

[43] Babak Dehbonei, Fernando Mejia. Formal Development of Safety-critical Software System in Railway Signaling. Application of Formal Methods, Prentice Hall.1995, 227-252

[44] Neil Storey. Safe-Critical Computer Systems. Addison Wesley, 1995

[45] Bjarne Stroustrup. The C++ Programming Language (3rd Edition), Addison Wesley, 1997

[46] Stanley B. Lippman, Josee Lajoie. C++ Primer (3rd Edition). Addison Wesley,1998

[47] Robert B., Murray. C++ Strategies and Tactics, Addison Wesley, 1993

[48] Cameron Hughes, Tracey Hughes. Parallel and Distributed Programming Using C++. Addison Wesley, 2003

[49] Charles Petzold. Progrmming Windows. Microsoft Press, 1998

[50] Robert Burdick, Essential Windows CE Application Programming, Wiley Computer Publishing, 1999

[51] Douglas Boling. Programming Microsoft Windows CE .NET (3rd Edition), Microsoft Press, 2003

[52] JosephA.Fisher, PaoloFaraboschi. Embedded Computing. Morgan Kaufmann, 2004

攻读硕士学位期间发表的学术论文及科学实践

[1] 黄重子 郭进. 微机化站内多点计轴系统的研究. IECT'2006

[2] Jin Guo, Zhongzi Huang, and Mingyuan Liu. Research on the Railway Safety Critical System with Petri nets. ITST' 2006

[3] 研究微机化站内多点计轴系统项目，主要完成硬件平台选型构建、Windows CE 内核定制、软件开发、调试及系统故障-安全逻辑分析与实现

[4] 参与微机化自动站间闭塞一体化系统项目的研发，负责 Windows CE 下软件开发、调试

[5] 参与成都乐创公司空间管理信息系统项目的嵌入式手持设备子系统的研发