

单片机与嵌入式系统 程序设计技术

周 航 慈

内容介绍

- 引言：编程言语与编程环境。
- 功能设计：结构分析与模块设计。完成功能设计后，系统即达到“有用”的水平。
- 可靠性设计：抗干扰设计和容错设计。完成可靠性设计后，系统即达到“能用”的水平。
- 算法设计：数据结构的优化设计和算法的优化设计。完成优化设计后，系统即达到“好用”的水平。
- 实例简介：配料控制系统设计。

引言

编程言语与编程环境

编程语言

- 汇编言语:

采用汇编言语编程必须对单片机的内部结构和外围电路非常了解，尤其是对指令系统必须非常熟悉，故对程序开发者的要求是比较高的。用汇编言语开发软件是比较辛苦的，程序量通常比较大，方方面面均需要考虑，一切问题都需要由程序设计者安排。

- 高级言语:

采用高级言语（主要是C言语）编程时，只要对单片机的内部结构和外围电路基本了解，对指令系统则不必非常熟悉，其编程比汇编言语轻松，很多细节问题不需要考虑，编译软件会替设计者安排好。故C言语在单片机软件开发中的应用越来越广，使用者越来越多。

编程环境

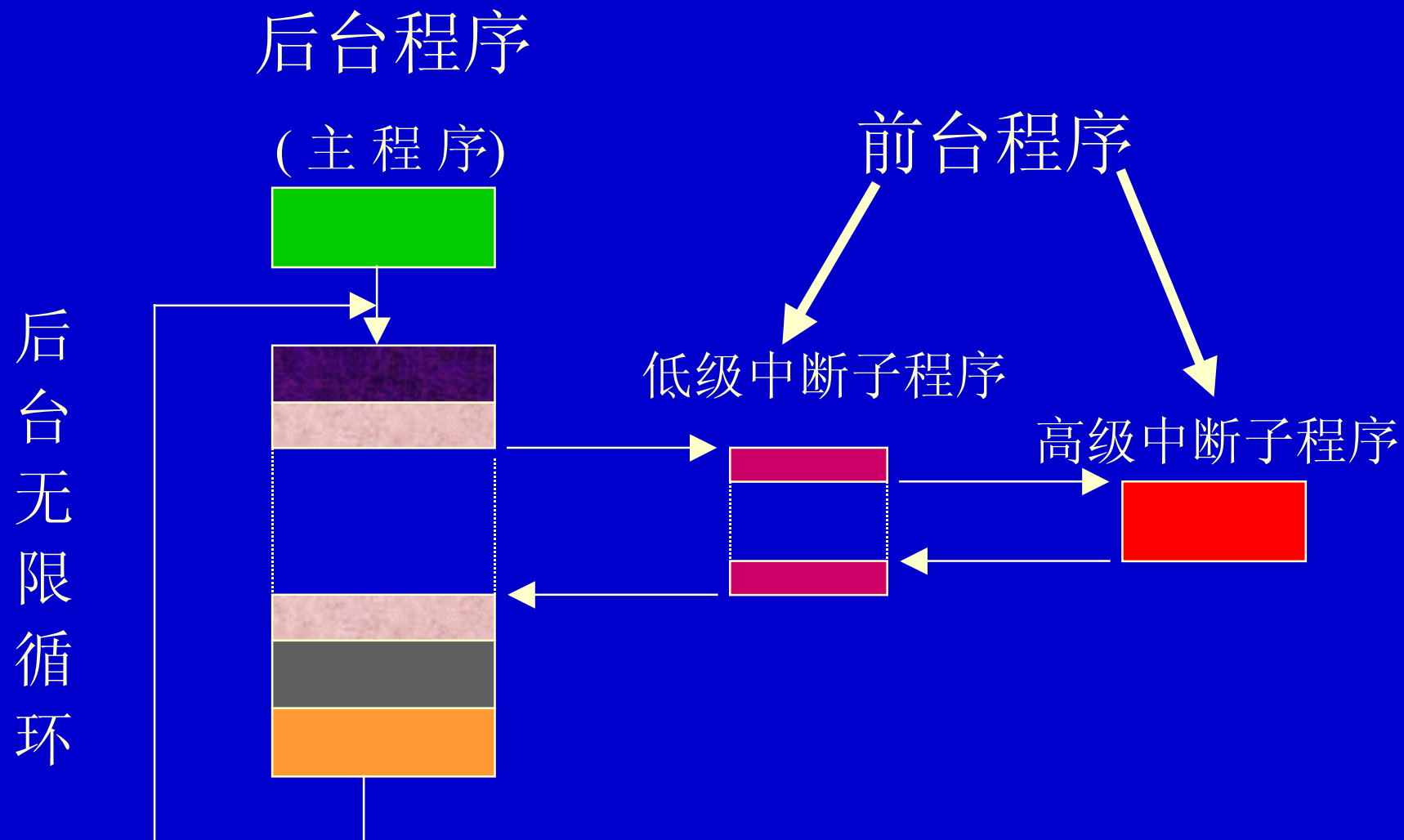
- 基于操作系统的编程：

以“实时多任务操作系统”内核为基础，只需完成相关任务的编程，其实时性和可靠性有保障，适用于功能较复杂的应用系统。

- 基于裸机的编程：

以空白的单片机芯片为基础，完成全部软件设计，其实时性和可靠性与设计人员的水平密切相关，适用于功能较简单的中小型应用系统，是电子工程师应该掌握的基本编程技术，本讲座将介绍其编程方法。

基于裸机的编程



功 能 设 计

结构分析与模块设计

软件系统结构分析

- 程序模块的自主性：

主程序和若干中断子程序是可以自主运行的程序模块，而普通子程序是不能自主运行的，可以将普通子程序看作主调程序模块的一部分。因此，整个系统程序设计就是完成主程序设计和若干个中断子程序设计的过程。

- 程序模块的功能性：

系统软件通常包含以下模块：自检模块、初始化模块、监控模块、显示输出模块、信息采集模块、数据处理模块、控制决策模块、信号输出模块、通讯模块、时钟模块等。

系统资源的安排

- 芯片硬件资源的安排：

端口引足、中断资源、定时器/计数器、其它功能部件（如串行口、A/D、PWM、比较器、看门狗等）。

- **RAM**资源的分配：

片内RAM用来存放各种变量、标志、堆栈、数据处理中的临时结果等等。片外RAM主要用来存放数据块。

程序模块的组织安排

- 自检和初始化模块： 安排在主程序中，系统上电后执行。
- 时钟模块： 固定安排在定时中断子程序中。
- 通讯模块： 通常安排在通讯中断子程序中。
- 监控、显示、信息采集、数据处理、控制决策、输出等模块： 可安排在主程序中，也可以安排在各种中断子程序中。建议全部安排在若干中断子程序（如定时中断子程序）中，使主程序的无限循环无事可做，进入节电睡眠模式，从而使低级中断子程序不需要保护现场，并可提高系统可靠性。

自检模块的设计

- **程序代码自检：** 执行校验算法，判断程序代码是否改变。
- **数据存储器自检：** 进行非破坏性读写校验，判断是否正常。
- **A/D通道自检：** 测试已知信号，判断是否正常。
- **D/A通道自检：** 输出已知数字量，用A/D通道来检测其转换结果是否正常。
- **显示自检：** 显示固定内容，判断是否正常。
- **蜂鸣器自检：** 响一声，判断是否正常。

初始化模块的设计

- 外部硬件初始化：对各种外部芯片设定明确的初始状态。
- 功能部件初始化：对片内功能部件设定明确的初始状态。
- 堆栈初始化：设置堆栈空间，初始化堆栈指针。
- 变量初始化：为各种变量和指针设置初始值（默认值）。
- 软件标志初始化：为所有软件标志设置初始状态。
- 系统时钟初始化：设置初始时间。
- 数据区初始化：通常是清零。

时钟模块的设计

- **时钟源**：硬件时钟芯片或者软件定时器。
- **定时周期的决定**：由系统最小时间分辨率来确定。
- **时钟单元的安排**：由系统时间动态变化范围来确定。
- **时钟的设置与校对**：通过键盘操作来完成。
- **系统时钟的使用**：触发与系统时间相关的任务。
- **计时器的使用**：测试某任务执行的时间。任务启动时清零，任务执行时与时钟一起运行，任务结束时停止，读取结果。
- **闹钟的使用**：控制某任务的时间间隔。任务启动时置初始值，任务执行时由时钟进行倒计时，计时结束则停止任务。

通讯模块的设计

- **波特率的设置：** 与信道质量有关，通讯双方共同约定。
- **通讯协议(帧结构)的设计：** 由通讯内容来决定。一般包含地址码、帧长（变长帧）、命令码、数据、校验码。
- **通讯缓冲区：** 其长度应该能够存放下最长帧，工作时和一个指针进行配合，完成一帧数据的收发。
- **通讯过程：** 如果采用查询模式，可一次接收（或发送）完一帧内容。为提高系统效率，最好采用中断模式，一次中断只接收（或发送）一个字节。
- **通讯命令的执行：** 最好在监控模块中执行。

信息采集模块的设计

- **采样周期的选择：** 由采样对象的频率特性决定。
- **数字信号的采集：** 光电隔离、重复采集。
- **模拟信号的采集：** 使用合适的数字滤波算法。
- **多路信号的采集：** 当定时间隔远小于采样周期时，可采用一路A/D器件对各路信号轮回进行采样。当定时间隔与采样周期相当时，必须采用多路A/D器件对各路信号同时进行采样。
- **随机信号的采集：** 由随机信号产生外部中断，在该中断子程序中进行采集。

数据处理模块的设计

- **数据格式的选择：** 用汇编语言编程时采用定点数格式还是采用浮点数格式，用C语言编程时采用整数格式还是采用实数格式，应该由数据的变化范围和分辨率来确定。
- **数据格式的转换：** 用汇编语言编程时应该进行人机交互格式与内部运算格式之间的转换，可调用相关子程序来完成。
- **数据处理过程：** 先编制若干相关标准运算子程序，然后将各种复杂运算分解为若干标准运算，通过调用这些标准运算子程序来实现数据处理的目标。为提高可靠性，尽可能使用子程序库中的子程序。为提高效率，尽可能使用迭代算法。

控制决策模块的设计

- **控制决策模块的位置：** 安排在信息采集模块和数据处理模块之后，信号输出模块之前。
- **控制决策算法的选择：** 根据控制对象的特性和系统控制指标的要求来选择，常用算法有PID算法及其变形算法、模糊控制算法，对于简单系统，也可以采用“乒乓”控制算法。
- **控制决策模块的输出：** 用来对对象进行逻辑控制（如通断控制、启停控制等）的决策结果一般用软件标志来表示，用来对对象进行程度控制（如温度控制、流量控制等）的决策结果为一个数据，必须转换成D/A器件对应的整形数据。

信号输出模块的设计

- **输出信号的缓冲：** 控制决策模块的运算结果不直接控制对象，而是存放在内存的输出缓冲区中，由本模块来执行输出。将“输出”从决策模块中独立出来以后，就可以实现一次决策，多次输出，提高可靠性。
- **异步决策、同步捆绑输出：** 不同的输出控制信号在不同的情况下决策产生，在不同时刻存入输出缓冲区。本模块执行时一并捆绑输出。
- **按状态输出：** 控制决策模块不产生各个独立的逻辑控制信号，而是产生系统状态信号，本模块按状态查表来输出一组逻辑控制信号，可保证输出的合理性，避免事故。

显示模块的设计

- **显示输出集中处理：** 将系统所有的显示输出全部集中到本模块中，可以避免分散编程时产生的冲突。
- **显示数据的获取：** 该模块通过查询系统的状态信息（状态编码和各种状态标志）可以判断出应该显示哪些数据，在预定的位置找到这些数据，并将其转换成显示所需要的格式。
- **显示内容的刷新：** 当某显示内容发生变化时，可置位“申请刷新”标志，由本模块来检测该标志并刷新显示，然后清除该标志。为保证显示内容正确，即使没有“申请”，也应该定时刷新。

监控模块的设计

- **监控模块的任务：** 获取键盘操作信息并解释之，调度执行相应模块，完成预定任务。遥控操作也可以合并到监控模块中进行解释执行。
- **监控模块的地位：** 整个软件系统的骨架。
- **监控模块的实质：** 保证系统在运行过程中的因果关系符合设计要求。其中“因”包括内因（系统当前状态）和外因（操作），“果”包括执行的操作和状态的变化。
- **监控模块的设计过程：** 系统状态分析、系统状态转移分析。状态编码，键盘编码，模块编码，生成监控框架。

监控模块的风格

- **顺序编码的监控模块：** 查表风格。将系统的“因果”关系变成表格，用“因”作为表格的索引，用“果”作为表格的内容。
- **特征编码的监控模块：** 判断分支风格。用“因”作为判断条件，即根据系统的当前状态和操作者的按键来进行分支，形成一棵“判断树”，将“果”作为“树叶”，执行对应的模块并更新系统的状态。
- **菜单驱动的监控模块：** 在使用CRT或大屏幕点阵液晶屏的设备中，采用菜单工作方式，这时系统的状态信息由“菜单树”中的节点位置来表示。监控结构可以采用上面两者之一。

可靠性设计

抗干扰设计和容错设计

硬件抗干扰设计

- **抗串模干扰的措施**：光电隔离、硬件滤波、过压保护、调制解调技术、高品质稳压电源、数字信号采用负逻辑传输。
- **抗共模干扰的措施**：平衡对称输入、选用高质量的运算放大器、良好的接地系统、系统接地点的正确联接方式、屏蔽。

软件抗干扰设计

- 数字信号的输入：多次采样，符合判断。
- 数字信号的输出：状态关联、重复输出。
- 模拟信号的输入：数字滤波。
- 模拟信号的输出：功率驱动（低阻输出）。
- 系统抗干扰技术：看门狗、后台睡眠、软件陷阱、重要信息的备份、掉电保护、热启动时的系统恢复技术。

硬件容错设计

- **硬件冗余**：系统级冗余（双机系统）、部件级冗余。
- **故障测试系统**：在原系统的电路中增加若干检测电路，配合检测软件，可随时对系统的运行状态进行监测，使系统具有故障自诊断功能。

软件容错设计

- 输入界面的容错设计：提示、合理性检查。
- 输出界面的容错设计：精度合理、单位信息。
- 堆栈空间：留有余地。
- 资源冲突的预防：完整独占地访问一个资源。
- 其它：软件标志的生命周期、子程序的资源占用、数据处理算法的优化。

算法设计

数据结构的优化处理

线性结构的处理

- **表格**：短表与长表的处理、动态表格的生成、数据的插入和删除。
- **数据缓冲区的管理**：初始化、数据填入、数据取出。应用场合：“整存整取”、“零存整取”、“整存零取”。
- **队列的管理**：初始化、数据入队、数据出队。应用场合：“零存零取”，先进先出。
- **堆栈的管理**：初始化、数据入栈、数据出栈。应用场合：“零存零取”，先进后出。
- **线性表格的排序**：插入排序、选择排序、冒泡排序、归并排序、快速排序。
- **线性表格的查找**：顺序查找、折半查找、分块查找、串的匹配。

非线性结构的处理

- **非线性结构的应用**：不能简单地用线性表格来处理的情况，如多层次的数据、相互关系复杂的数据。
- **树的存储及使用**：常规链接存储的低效及其“线性化”存储后的遍历算法。
- **图的存储及使用**：邻接矩阵和邻接表存储的用法、图的深度优先搜索和广度优先搜索算法、网络的最小生成树和最短路径问题。

常用数据处理算法

- 线性方程组的求解：主元消去法、三元线性方程组的行列式法。
- 常用插值算法：线性插值算法、抛物线插值算法。
- 常用统计算法：均值和标准方差的估算、用数理统计方法剔除坏数据样本。

常用特殊算法

- **递归算法**：从终点出发的算法，对堆栈空间的需求很大，要小心使用。
- **递推算法**：从起点出发的算法，对堆栈空间的需求很小，只要有可能，应尽力采用。
- **回溯算法**：应用在探索性的问题解决方案求解过程中。需要保存一系列求解过程的局部方案，当问题规模较大时，时间和空间的需求均较大，需要配合其它智能搜索算法提高效率，才能使问题在可以接受的时间之内求解成功。

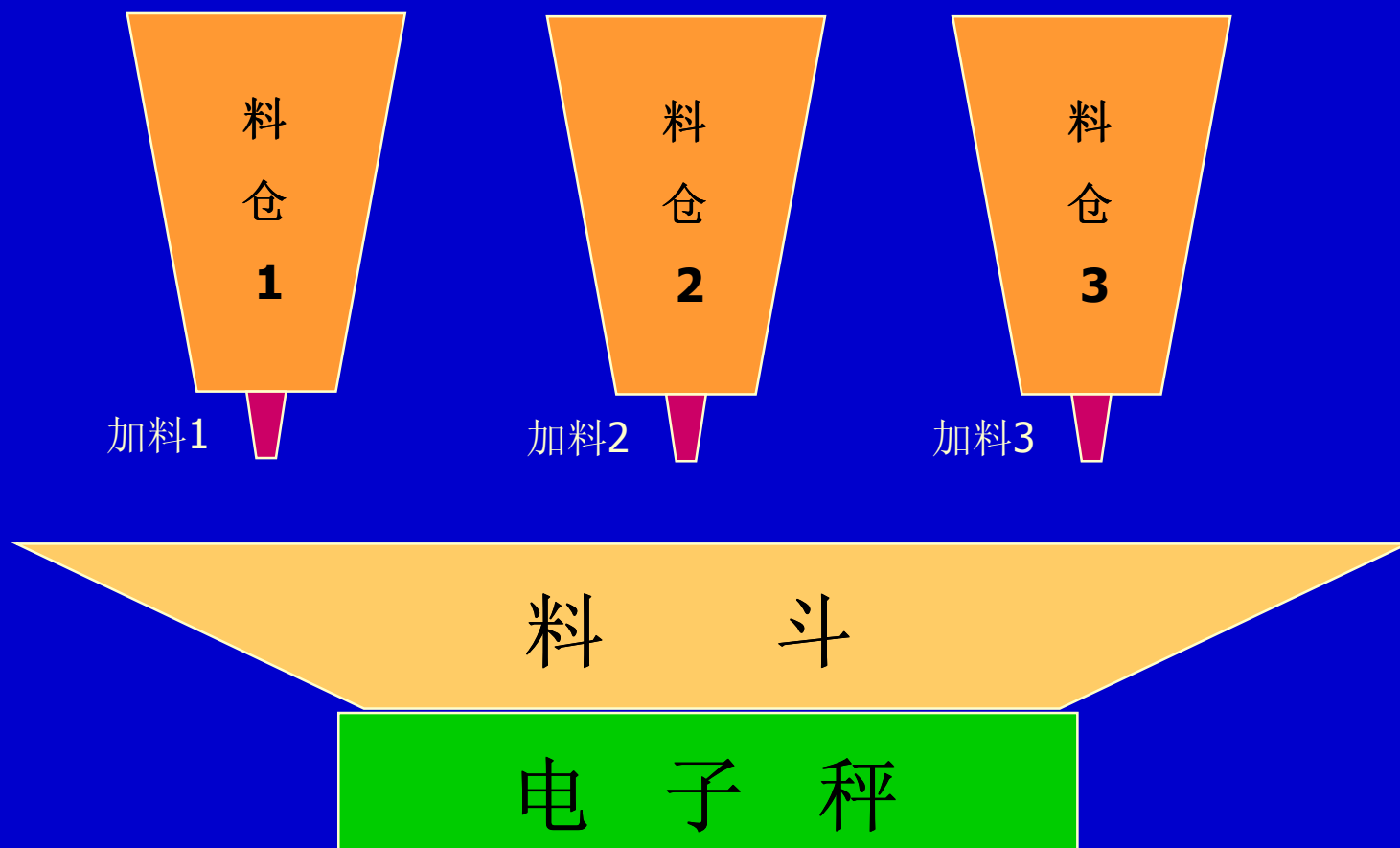
信息论与编码算法

- **系统的信息流量估算：** 通过估算，优化系统设计。
- **检错码的使用：** 能够发现数据在传送或储存后产生的错误，常用检错码有：奇偶校验码、CRC校验码。
- **纠错码的使用：** 能够发现并纠正数据在传送或储存后产生的错误，常用纠错码有：汉明码、BCH码。
- **随机数的产生：** 人机交互式随机数产生方法、用线性移位寄存器构成伪随机数发生器。

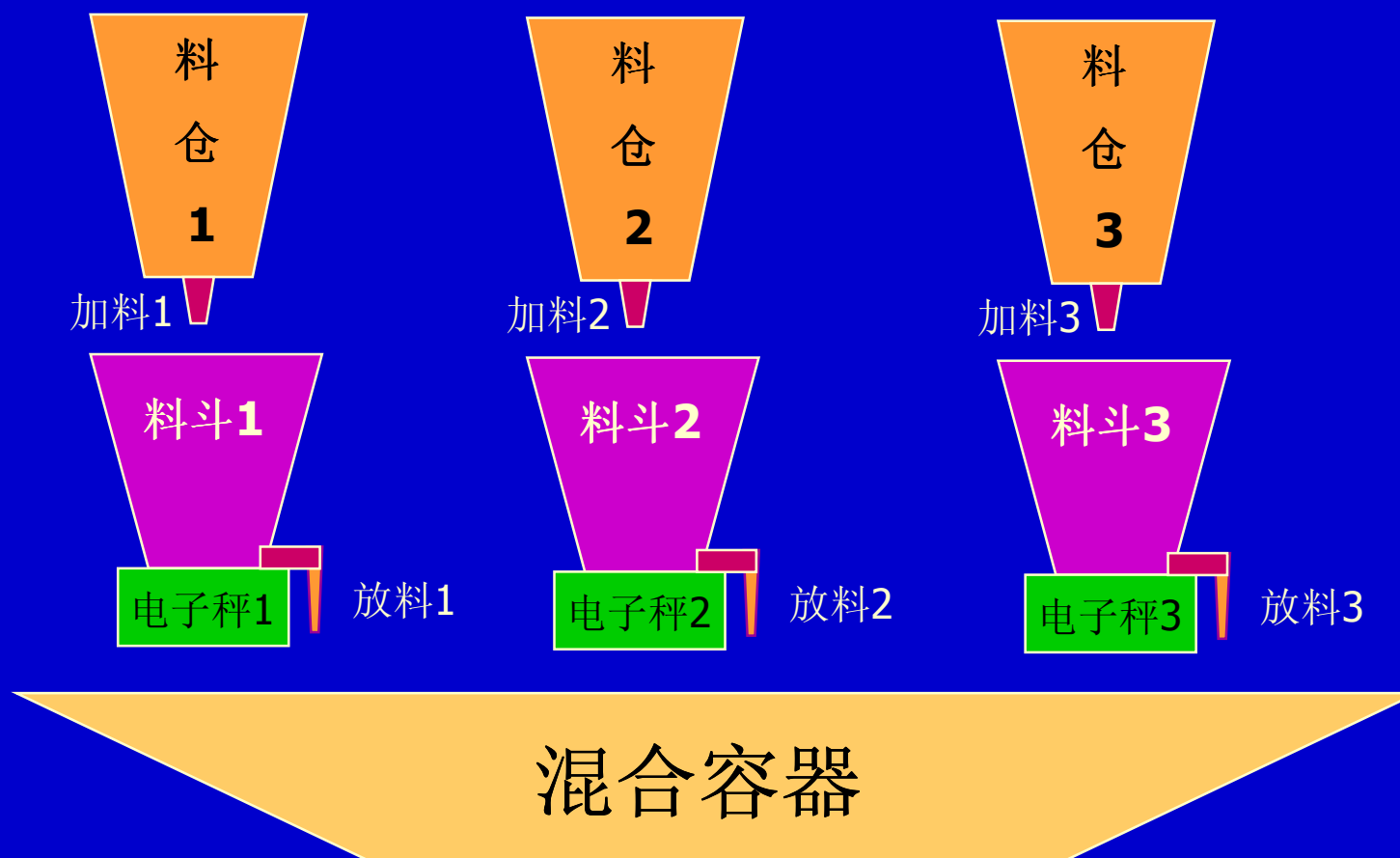
实例简介

配料控制系统设计

配料控制系统示意图一



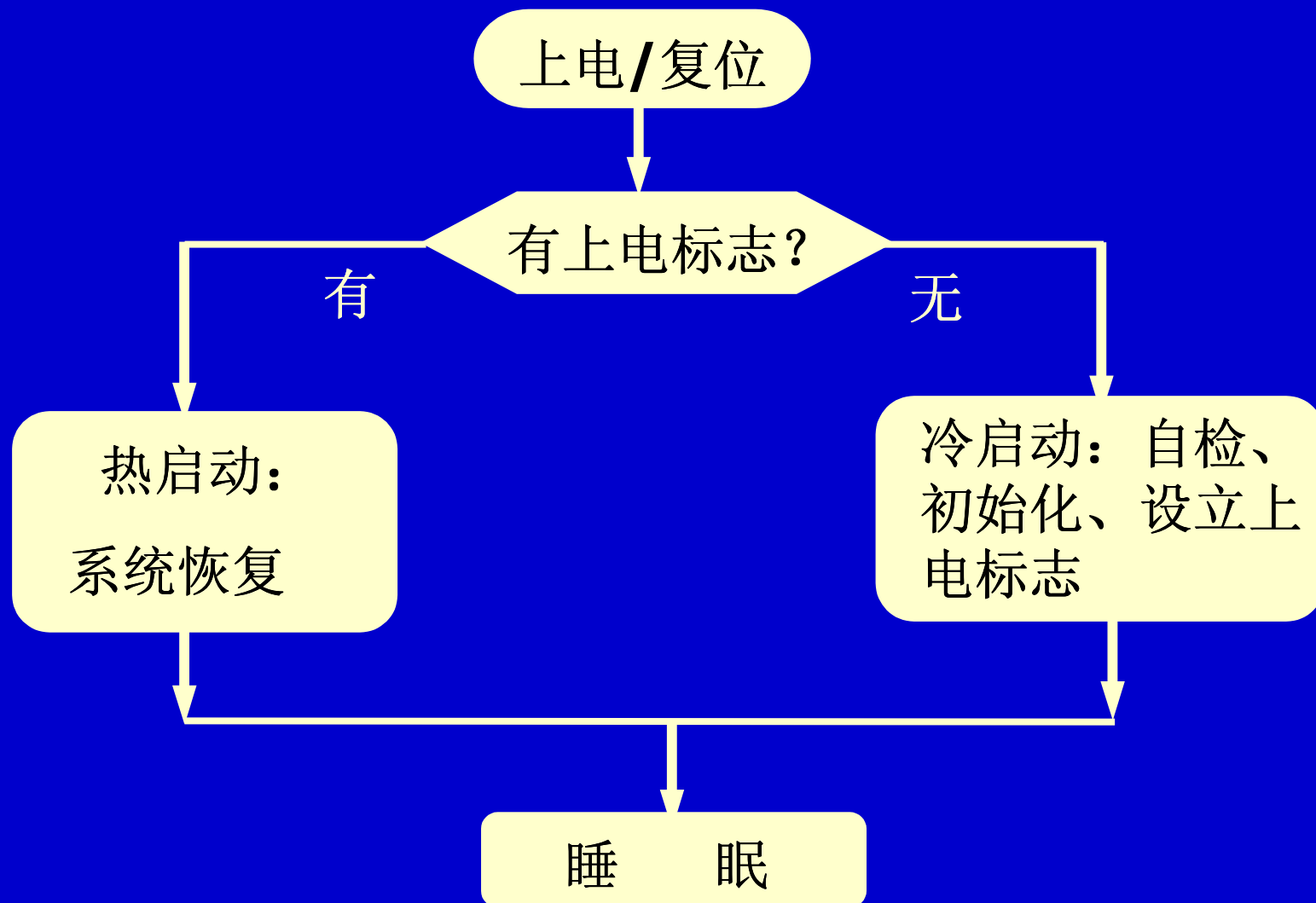
配料控制系统示意图二



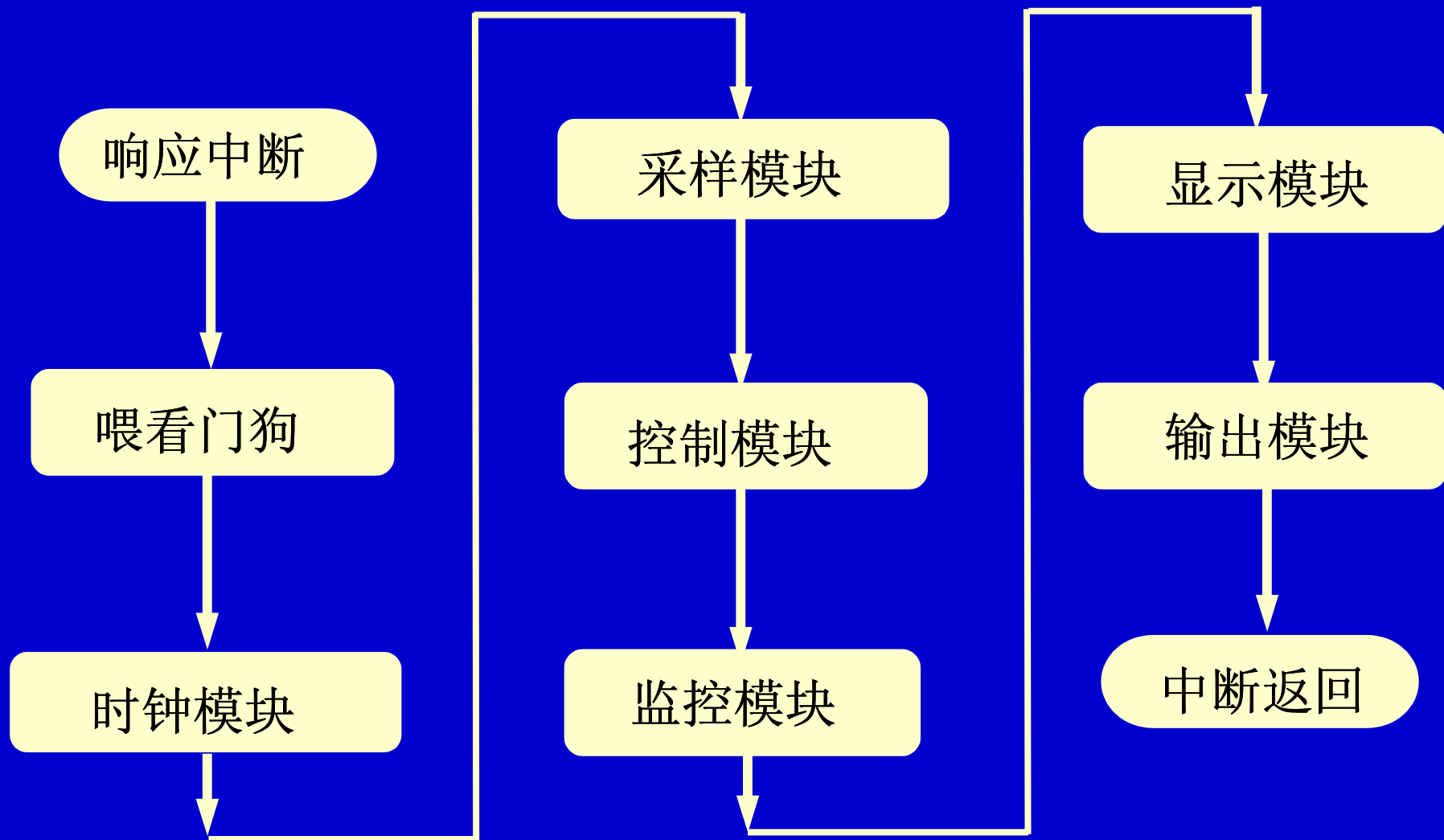
配料控制系统功能

- 电子秤的标定：按不同量程标定各个电子秤。
- 系统的手动控制：检查设备是否正常。
- 配方输入：输入各种成分的数据。
- 任务输入：输入总共配料次数。
- 自动配料：启动自动配料过程。

主程序流程图



定时中断子程序流程图



参考书介绍

- 《单片机应用程序设计技术》（修订版）
周航慈 著 北京航空航天大学出版社 2002.11
内容：功能设计与可靠性设计
- 《单片机程序设计基础》（修订版）
周航慈 著 北京航空航天大学出版社 2003.8
内容：算法设计（附上机学习光盘）
- 《信息技术基础》
周航慈 等编著 北京航空航天大学出版社 2002.8
内容：信息论基础知识、信源编码（信息压缩）、信道编码（信息容错）、加密处理（信息安全）。

欢 迎 报 考

东 华 理 工 学 院

《电路与系统》硕士点

www.ecgi.jx.cn

谢谢各位