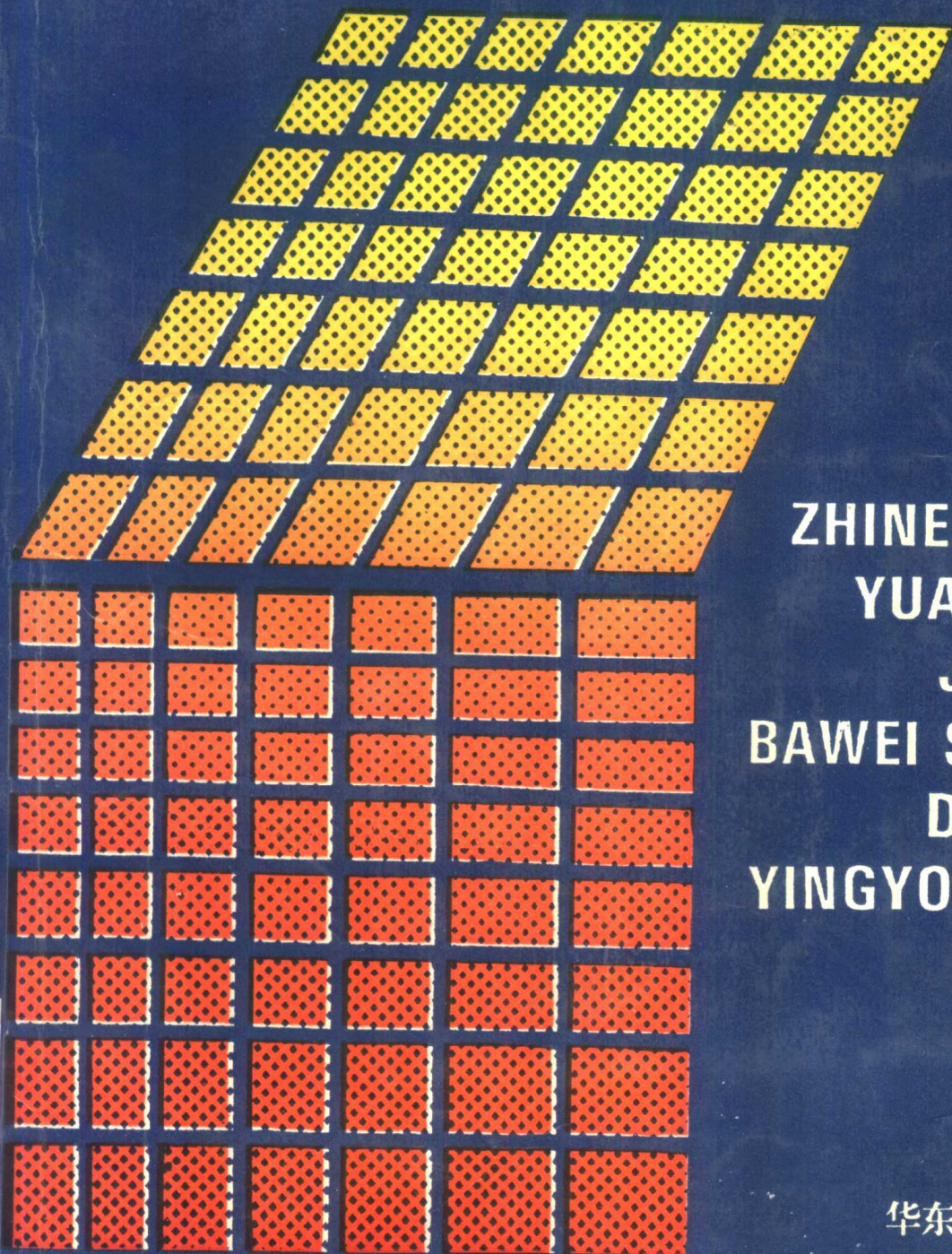


• 季建华 • 都志杰 • 吴勤勤 • 编著

智能仪表原理、设计及调试

—— 8 位、16 位单片机应用技术



ZHINENG YIBIAO
YUANLI SHEJI
JI TIAOSHI
BAWEI SHILIUWEI
DANPIANJI
YINGYONG JISHU

华东理工大学出版社

智能仪表原理、设计及调试

—— 8 位、16 位单片机应用技术

季建华 都志杰 吴勤勤 编著

华东理工大学出版社

(沪)新登字 208 号

智能仪表原理、设计及调试

—— 8 位、16 位单片机应用技术

季建华 都志杰 吴勤勤 编著

华东理工大学出版社出版发行

上海市梅陇路 130 号

邮政编码 200237

新华书店上海发行所发行经销

上海展望印刷厂印刷

开本 787×1092 1/16 印张 18 字数 443 千字

1995 年 4 月第 1 版 1997 年 6 月第 2 次印刷

印数 3001—6000

ISBN 7-5628-0562-8/TP·78 定价: 17.00 元

前 言

随着微电子技术和计算机技术的迅速进展,特别是单片微机的出现和发展,引起了仪器仪表设计的巨大变革。自动化仪表正日趋智能化、系统化、小型化和多功能化,变革的关键是将微机应用于仪表之中。

近年来,各仪表研究所、高等院校、仪表制造厂均在开发带微机的新一代仪器仪表,并将其应用于生产过程自动化系统中。为了研制性能优越的新型智能仪表,广大自动化仪表设计、生产及使用人员迫切希望系统了解和掌握智能仪表的基本原理和设计方法。本教材即是根据这一要求,以目前普遍使用的8位、16位单片机为例,详细阐述智能仪表的设计原理和调试方法。

本书共分十章。第一章扼要介绍智能仪表的功能、组成,仪表的基本设计思想和研制步骤。第2章至第6章分别介绍仪表的硬件电路设计,包括主机电路、过程输入输出通道、人机接口电路、通信接口以及硬件抗干扰技术。第7章至第9章分别阐述仪表的软件设计,包括软件设计的一般方法、监控程序、测量算法和控制算法。第10章以典型产品为例,具体介绍智能仪表的设计与调试。

本书在内容安排上注重过程检测控制中的实际问题,叙述上力求深入浅出,强调实用性和先进性,所涉及的内容尽可能给出实际线路、程序或软件流程。

本书由季建华、都志杰、吴勤勤编著。吴勤勤负责编写第1、2、10章,季建华负责编写第3~6章,都志杰负责编写7~9章。全书由吴勤勤统编,并和季建华一起共同整理定稿。参加编写本书第一版的还有朱康余、单青。

此外,吴斌昌为编写本书提供了技术资料,并提出了宝贵的意见。在编写过程中还得到了华东理工大学工业自动化仪表教研组等10位同志的帮助,谨在此表示谢意。

本书可作为大专院校自动化仪表或计算机应用专业的教材,也可供从事智能仪表设计、制造、使用的工程技术人员参考。

由于编者水平有限,错误和不妥之处在所难免,恳请读者批评指正。

编 者

1994年10月于上海

目 录

1 智能仪表设计概述	(1)
1.1 智能仪表的功能和组成	(1)
1.1.1 智能仪表的主要功能	(1)
1.1.2 智能仪表的基本组成	(3)
1.2 智能仪表的设计思想和研制步骤	(4)
1.2.1 智能仪表的基本设计思想	(4)
1.2.1.1 模块化设计	(4)
1.2.1.2 模块的连接	(4)
1.2.2 智能仪表的设计研制步骤	(5)
1.2.2.1 确定任务、拟制设计方案	(5)
1.2.2.2 硬件、软件研制及仪表结构设计	(6)
1.2.2.3 仪表总调、性能测定	(7)
1.3 智能仪表的开发工具	(7)
1.3.1 开发系统的功能	(7)
1.3.2 几种开发系统	(8)
1.3.2.1 简易型开发系统	(8)
1.3.2.2 软件模拟开发系统	(8)
1.3.2.3 通用型开发系统	(8)
1.3.2.4 专用开发系统	(9)
2 主机电路——仪表硬件电路设计之一	(10)
2.1 由 MCS-51 单片微机构成的主机电路	(10)
2.1.1 MCS-51/52 单片微型计算机	(10)
2.1.1.1 概述	(10)
2.1.1.2 MCS-51 引脚	(10)
2.1.1.3 微处理器	(11)
2.1.1.4 存贮器	(11)
2.1.1.5 输入输出口	(13)
2.1.1.6 定时/计数器	(13)
2.1.1.7 串行口	(19)
2.1.1.8 中断系统	(22)
2.1.2 主机电路	(25)
2.1.2.1 外接存贮器	(25)
2.1.2.2 外接 I/O 接口	(26)
2.1.2.3 主机电路举例	(29)
2.2 由 MCS-96 单片微机构成的主机电路	(30)

2.2.1	MCS-96 单片微型计算机	(30)
2.2.1.1	概述	(30)
2.2.1.2	8098 引脚	(30)
2.2.1.3	微处理器	(31)
2.2.1.4	存储器	(31)
2.2.1.5	基本输入输出	(34)
2.2.1.6	中断系统	(35)
2.2.1.7	控制和状态寄存器	(37)
2.2.1.8	定时/计数器	(39)
2.2.1.9	高速输入部件 HSI	(40)
2.2.1.10	高速输出部件 HSO	(42)
2.2.1.11	A/D 转换器	(44)
2.2.1.12	脉宽调制输出部件(D/A)	(46)
2.2.1.13	串行口	(46)
2.2.2	主机电路	(49)
3	过程输入输出通道——仪表硬件电路设计之二	(51)
3.1	模拟量输入通道	(51)
3.1.1	模拟量输入通道的结构	(51)
3.1.2	A/D 转换芯片及其与单片机的接口	(52)
3.1.2.1	A/D 转换芯片的一般描述	(52)
3.1.2.2	几种 A/D 芯片及接口电路	(53)
3.1.3	模拟量输入通道的其它器件	(69)
3.1.3.1	多路模拟开关	(69)
3.1.3.2	采样保持电路	(70)
3.1.3.3	前置放大电路	(71)
3.1.4	模拟量输入通道设计举例	(74)
3.2	模拟量输出通道	(77)
3.2.1	模拟量输出通道的结构	(77)
3.2.2	D/A 转换芯片及其与单片机的接口	(78)
3.2.2.1	D/A 转换芯片的一般描述	(78)
3.2.2.2	几种 D/A 芯片及其接口电路	(79)
3.2.2.3	电压/电流转换电路	(82)
3.2.3	模拟量输出通道设计举例	(83)
3.3	开关量输入输出通道	(84)
3.3.1	开关量输入输出通道的结构	(85)
3.3.2	开关量输入输出通道设计举例	(85)
4	人一机接口电路——仪表硬件电路设计之三	(89)

4.1 显示器接口	(89)
4.1.1 LED 显示器	(89)
4.1.1.1 LED 显示器结构	(89)
4.1.1.2 LED 显示器工作原理	(89)
4.1.1.3 LED 显示器接口实例	(90)
4.1.2 LCD 显示器	(92)
4.1.2.1 LCD 显示器结构	(92)
4.1.2.2 LCD 显示器工作原理	(93)
4.1.2.3 LCD 显示器接口实例	(94)
4.1.3 点阵式 LED 显示器	(95)
4.2 键盘接口	(95)
4.2.1 键盘结构和类型	(95)
4.2.2 抖动与串键	(96)
4.2.2.1 抖动	(96)
4.2.2.2 串键	(96)
4.2.3 非编码式键盘接口电路	(96)
4.2.4 编码式键盘接口电路	(99)
4.2.4.1 键盘、显示接口芯片 8279	(99)
4.2.4.2 键盘、显示器与 8279 的接口	(105)
4.3 打印机接口	(108)
4.3.1 打印原理	(108)
4.3.2 GP-16 微型打印机接口	(108)
4.3.2.1 GP-16 接口信号	(108)
4.3.2.2 命令字	(108)
4.3.2.3 状态字	(110)
4.3.2.4 GP-16 与单片机的接口	(111)
4.3.3 PP-40 彩色描绘器接口	(111)
4.3.3.1 PP-40 接口信号	(111)
4.3.3.2 PP-40 操作方式	(112)
4.3.3.3 PP-40 与单片机的接口	(115)
5 通信接口电路——仪表硬件电路设计之四	(117)
5.1 串行通信接口	(117)
5.1.1 RS-232C 标准	(117)
5.1.1.1 数据传送格式和电气特性	(117)
5.1.1.2 接口信号	(118)
5.1.1.3 通信系统结构	(120)
5.1.2 串行通信接口电路	(121)
5.1.2.1 串行通信方法	(121)

5.1.2.2 串行通信的接口器件	(123)
5.1.2.3 8251 与单片机的接口	(126)
5.1.3 单片机与 IBM-PC 计算机的数据通信	(128)
5.1.4 RS-422、RS-423、RS-485 标准	(132)
5.2 并行通信接口	(133)
5.2.1 IEEE-488 标准	(133)
5.2.1.1 性能	(133)
5.2.1.2 总线结构和接口信号	(134)
5.2.1.3 总线工作过程	(137)
5.2.2 并行通信接口器件	(139)
6 智能仪表硬件电路的抗干扰	(142)
6.1 干扰与噪声源	(142)
6.1.1 外部干扰	(142)
6.1.1.1 串模干扰	(142)
6.1.1.2 共模干扰	(143)
6.1.1.3 电源干扰	(144)
6.1.2 数字电路的干扰	(144)
6.2 抗干扰措施	(146)
6.2.1 串模干扰的抑制	(146)
6.2.2 共模干扰的抑制	(148)
6.2.3 过程通道的抗干扰	(149)
6.2.3.1 开关量输入输出通道的抗干扰	(149)
6.2.3.2 模拟量输入输出通道的抗干扰	(152)
6.2.4 电源与电网干扰的抑制	(156)
6.2.4.1 抑制电网干扰的措施	(156)
6.2.4.2 印刷电路板电源开关噪声的抑制	(156)
6.2.5 地线系统干扰的抑制	(157)
7 监控程序——仪表功能的软件实现之一	(158)
7.1 软件设计方法	(158)
7.1.1 结构化设计与编程	(158)
7.1.1.1 “自顶向下”的设计方法	(158)
7.1.1.2 模块化编程	(159)
7.1.2 软件功能测试	(159)
7.1.2.1 测试目的	(159)
7.1.2.2 测试方法	(160)
7.1.2.3 测试的基本原则	(160)
7.1.3 软件的运用、维护和改进	(160)

7.2 监控程序设计	(161)
7.2.1 概述	(161)
7.2.2 监控主程序	(162)
7.2.3 初始化管理	(163)
7.2.4 键盘管理	(164)
7.2.4.1 一键一义的键盘管理	(164)
7.2.4.2 一键多义的键盘管理	(166)
7.2.5 显示管理	(169)
7.2.6 中断管理	(170)
7.2.7 时钟管理	(172)
7.2.8 手—自动控制	(173)
7.2.9 自诊断处理	(173)
8 测量算法——仪表功能的软件实现之二	(175)
8.1 克服随机误差的软件算法(数字滤波)	(175)
8.1.1 程序判断法	(175)
8.1.2 中位值滤波法	(176)
8.1.3 算术平均滤波法	(178)
8.1.4 递推平均滤波法	(179)
8.1.5 加权递推平均滤波法	(180)
8.1.6 一阶惯性滤波法	(181)
8.1.7 复合滤波法	(182)
8.2 克服系统误差的软件算法	(183)
8.2.1 系统误差的模型校正法(非线性校正)	(183)
8.2.1.1 代数插值法	(184)
8.2.1.2 最小二乘法	(188)
8.2.2 系统误差的标准数据校正法	(190)
8.2.3 仪表零位误差和增益误差的校正方法	(191)
8.2.3.1 零位误差的校正方法	(191)
8.2.3.2 增益误差的校正方法	(191)
8.2.4 传感器温度误差的校正方法	(192)
8.3 量程自动切换与工程量变换	(193)
8.3.1 量程自动切换	(193)
8.3.1.1 采用程控放大器	(193)
8.3.1.2 自动切换不同量程的传感器	(194)
8.3.2 工程量变换(标度变换)	(195)
9 控制算法——仪表功能的软件实现之三	(197)
9.1 PID 控制算法	(197)

9.1.1	概述	(197)
9.1.1.1	P 控制算法	(198)
9.1.1.2	PI 控制算法	(198)
9.1.1.3	PD 控制算法	(199)
9.1.2	完全微分型 PID 控制算法	(199)
9.1.3	不完全微分型 PID 控制算法	(200)
9.1.4	PID 算法的改进	(202)
9.1.4.1	抗积分饱和	(202)
9.1.4.2	防止积分极限环的产生	(203)
9.1.4.3	微分先行和输入滤波	(203)
9.1.5	PID 调节器参数的整定及采样周期的选择	(204)
9.1.5.1	控制度	(204)
9.1.5.2	采样周期	(204)
9.1.5.3	参数整定	(205)
9.2	智能控制算法	(207)
9.2.1	PID 参数自整定	(208)
9.2.2	模糊控制算法	(209)
9.2.2.1	概述	(209)
9.2.2.2	控制算法结构的设计	(210)
9.2.2.3	模糊控制实时算法	(214)
9.2.3	人工神经网络技术	(217)
9.2.3.1	概述	(217)
9.2.3.2	神经网络基本概念	(217)
9.2.3.3	学习算法举例	(219)
10	智能仪表设计实例	(221)
10.1	设计准则	(221)
10.2	设计实例	(223)
10.2.1	温度程序控制仪	(223)
10.2.1.1	设计要求	(223)
10.2.1.2	系统组成和工作原理	(224)
10.2.1.3	硬件结构和电路设计	(224)
10.2.1.4	软件结构和程序框图	(228)
10.2.2	变频调速控制器	(234)
10.2.2.1	变频调速系统	(234)
10.2.2.2	控制器功能和硬件配置	(235)
10.2.2.3	变频算法和软件设计	(236)
10.2.3	多通道 pH 控制器	(237)
10.2.3.1	设计要求	(238)

10.2.3.2	硬件组成	(238)
10.2.3.3	控制特性	(238)
10.2.3.4	软件设计	(238)
10.2.4	色谱分析数据处理仪	(241)
10.2.4.1	设计要求	(241)
10.2.4.2	硬件结构和电路设计	(242)
10.2.4.3	软件算法和程序框图	(243)
10.3	仪表调试	(250)
10.3.1	常见故障	(250)
10.3.2	调试方法	(251)
10.3.2.1	静态调试	(252)
10.3.2.2	动态调试	(253)
附录 1	思考题与习题	(255)
附录 2	MCS-51 指令表	(259)
附录 3	MCS-96 指令表	(264)
附录 4	常用集成芯片引脚图	(267)

1 智能仪表设计概述

微型计算机技术的迅速发展,使传统的仪器仪表发生了根本性变革。即以微型计算机(简称微机)为主体,代替传统仪表的常规电子线路,成为新一代的具有某种智能的灵巧仪表。将微机引入测量、控制仪表中,不仅能解决传统仪表不能解决或不易解决的问题,而且能简化电路、增加功能、提高精度和可靠性。由于这类仪表已能实现人脑的一部分功能,例如四则运算、逻辑判断、命令识别等,有的还能够进行自校正、自诊断,并具有自适应、自学习的能力,因此,人们习惯上称它们为智能仪表(或微机化仪表)。但智能仪表的水平高低不一,目前所见的大部分产品智能化的程度还不高。当然,随着科学技术的进一步发展,这类仪表所具有的智能水平将会越来越高。

单片微型计算机(简称单片微机或单片机)的问世和性能的不断改善,大大加快了仪器仪表微机化和智能化的进程。与多芯片组成的微机相比,单片微机的体积小、功耗低、价格也较便宜,用单片机开发各类微机化产品,周期短、成本低,在计算机和仪表一体化设计中,有着一般微机无法比拟的优势。因此,它在微机系统和仪表中的应用已日趋广泛。本教材顺应这一趋势,采用当前流行的 MCS-51 和 MCS-96 系列的单片机作为主机电路,介绍智能仪表的设计和调试。

1.1 智能仪表的功能和组成

1.1.1 智能仪表的主要功能

将单片微机引入仪表中,能解决的问题是多方面的,大体上能实现如下一些功能:

(1) 自动校正零点、满度和切换量程

智能仪表的自校正功能大大降低了因仪表零漂和特性变化造成的误差,而量程的自动切换又给使用带来了方便,并可提高读数的分辨率。

(2) 多点快速检测

能对多个参数(模拟量或开关量信号)进行快速、实时检测,以便及时了解生产过程的瞬变工况。

(3) 自动修正各类测量误差

许多传感器的特性是非线性的,且受环境温度、压力等参数的影响,从而给仪表带来误差。在智能仪表中,只要掌握这些误差的规律,就可依靠软件进行修正。常见的有测温元件的非线性校正,热电偶冷端温度补偿,气体流量的温度压力补偿等等。

(4) 数字滤波

通过对主要干扰信号特性的分析,采用适当的数字滤波算法,可抑制各种干扰(例如低频

干扰、脉冲干扰)的影响。

(5) 数据处理

能实现各种复杂运算,对测量数据进行整理和加工处理,例如统计分析、查找排序、标度变换、函数逼近和频谱分析等等。

(6) 各种控制规律

能实现 PID 及各种复杂控制规律,例如可进行串级、前馈、解耦、非线性、纯滞后、自适应、模糊等控制,以满足不同控制系统的需要。

(7) 多种输出形式

输出形式有数字(或指针)显示、打印记录、声光报警,也可以输出多点模拟量(或开关量)信号。

(8) 数据通信

能与其它仪表和计算机进行数据通信,以便构成不同规模的计算机测量控制系统。

(9) 自诊断

在运行过程中,可对仪表本身各组成部分进行一系列测试。一旦发现故障即能告警,并显示出故障部位,以便及时处理。

(10) 掉电保护

仪表内装有后备电池和电源自动切换电路。掉电时,能自动将电池接至 RAM,使数据不致丢失。也可采用电可改写只读存储器 EEPROM,来代替 RAM,存储重要数据,以实现掉电保护的功能。

在一些不带微机的常规仪表中,通过增加器件和变换线路,亦能或多或少地具有上述的某种功能,但往往要付出较大的代价。性能上的些许提高,会使仪表的成本大大增加。而在智能仪表中,性能的提高,功能的扩大,是比较容易实现的。低廉的单片微机芯片使这类仪表具有较高的性能价格比。

为对传统仪表更新换代,近年来,国内各仪表研制与使用单位正致力于智能仪表的开发和应用研究工作。例如开发出能自动进行温压补偿的节流式流量计,能对测量元件、检测装置或执行机构进行快速测试和校核的各种校验设备,能对各种谱图进行分析和数据处理的色谱数据处理仪,能输出正弦波的变频调速控制器,能进行程序控温的多段温度控制仪,以及能实现 PID 和复杂控制规律的数字式调节器、智能式控制器等等。

与此同时,一些厂家也从国外引进了新的产品。例如美国 Honeywell 公司的 DSTJ-3000 系列智能式变送器,它在半导体硅单晶片上配置了差压、静压和温度三种传感元件,进行差压值状态的复合测量,可对温度、静压实现自动补偿,从而获得较高的测量精度($\pm 0.1\%FS$)。该变送器还可用遥控操作器远距离地进行零位校正、阻尼调整,测量范围的变更,以及线性或平方根的选择,使用和维护十分方便。

日本横河(Yokogawa)公司的模拟数字混合式记录仪,采用开环扫描测量方法,没有伺服放大器、平衡电机、滑线电阻等部件。测量信号经多路开关扫描输入后,进行前置放大和 A/D 转换,再在微机控制下,发出相应的脉冲数驱动步进电机,带动打印头作横向移动而画出模拟曲线,也可打出数据和表格,其测量精度比传统记录仪高。这种记录仪除能进行模拟或数字显示和记录外,还具有自诊断、自校正、求差、报警等功能,并带有通信接口。

美国 Foxboro 公司的数字化自整定调节器,能自行计算 PID 参数,并使过程的恢复时间减

到最小值。该调节器具有人工智能式的控制方法,采用“专家系统”技术,像有经验的控制工程师那样,能运用操作经验来整定调节器,工作迅速、正确。自整定调节器组态灵活,操作方便,节省了控制系统的投入时间,特别当对象特性变化频繁或在非线性系统中,由于它能自动改变参数,并始终保持系统品质最佳,因此大大提高了系统运行的经济效率。

1.1.2 智能仪表的基本组成

智能仪表由硬件和软件两大部分组成。

硬件部分包括主机电路、过程输入输出通道(模拟量输入输出通道和开关量输入输出通道)、人机联系部件和接口电路以及串行或并行数据通信接口等,如图 1.1.1 所示。主机电路用来存贮数据、程序,并进行一系列运算处理,它由单片微机(其中包括中央处理器 CPU、程序存储器 ROM、数据存储器 RAM、输入输出接口 I/O 和定时/计数电路 CTC 等)及扩展电路组成。模拟量输入输出通道(分别由模/数(A/D)转换器、数/模(D/A)转换器等构成)用来输入输出模拟量信号;而开关量输入输出通道则用来输入输出开关量信号。人机联系部件的作用是沟通操作者与仪表之间的联系。通信接口则用来实现仪表与外界交换数据。

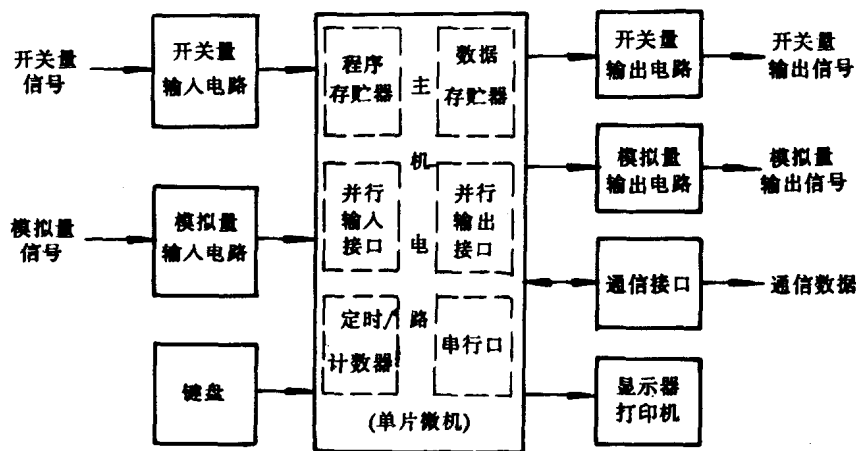


图 1.1.1 智能仪表硬件组成框图

智能仪表的工作过程如下:输入信号先在过程输入通道的预处理电路中进行变换、放大、整形、补偿等项处理。对于模拟量信号,需经模拟量通道的 A/D 转换器转换成数字信号,再通过接口送入主机。由 CPU 对输入数据进行加工处理、计算分析等一系列工作,并将运算结果存贮在 RAM 中。同时可通过接口送至显示器或打印机,也可输出开关量信号和经模拟量通道的 D/A 转换器转换成模拟量的信号。还可通过串行标准接口(例如 RS-232C 等)或并行标准接口(例如 IEEE-488 等)实现数据通信,完成更复杂的测量、控制任务。仪表的整个工作是在软件控制之下进行的,工作程序应预先编制好,写入 EPROM 中。必要的参数、命令可由键盘输入,存于 RAM 中。

智能仪表的软件包括监控程序、中断处理(服务)程序以及实现各种算法的功能模块。监控程序是仪表软件的中心环节,它接受和分析各种命令,并管理和协调整个程序的执行;中断处

理程序是在人机联系部件或其它外围设备提出中断申请,并为主机响应后直接转去执行,以便及时完成实时处理任务;功能模块用来实现仪表的数据处理和控制在功能,包括各种测量算法(例如数字滤波、标度变换、非线性校正等)和控制算法(例如 PID 控制、前馈控制、纯滞后控制、模糊控制等)。

以上只是智能仪表的大致组成。至于仪表内部的具体线路和硬软件的设计方法,将在以后各章节中详细阐述。

1.2 智能仪表的设计思想和研制步骤

研制一台智能仪表是一个复杂的过程,这一过程包括分析仪表的功能要求和拟制总体设计方案,确定硬件结构和软件算法,研制逻辑电路和编制程序,以及仪表的调试和性能的测试等等。为保证仪表质量和提高研制效率,设计人员应在正确的设计思想指导下进行仪表研制的各项工作。

1.2.1 智能仪表的基本设计思想

1.2.1.1 模块化设计

依据仪表的功能要求和技术经济指标,自顶向下(由大到小)地按仪表功能层次把硬件和软件分成若干个模块,分别进行设计和调试,然后把它们连接起来,进行总调,这就是设计仪表的基本思想。

如前所述,通常把硬件分成主机、过程通道、人机联系部件、通信接口和电源等几个模块;而把软件分成监控程序(包括初始化、键盘、显示管理、中断管理、时钟管理、自诊断等)、中断处理程序以及各种测量和控制算法等功能模块。这些硬件、软件模块还可继续细分,由下一层次的更为具体的模块来支持和实现。模块化设计的优点是:无论是硬件还是软件,每一个模块都相对独立,故能独立地进行设计、研制、调试和修改,从而使复杂的工作得以简化。模块间的相对独立也有助于研制任务的分解和设计人员之间的分工合作,这样可提高工作效率和仪表的研制速度。

1.2.1.2 模块的连接

上述各种软、硬件模块的研制调试完成之后,还需要将它们按一定的方法连接起来,才能构成完整的仪表,以实现数据采集、传输、处理和输出等各项功能。软件模块的连接,一般是通过监控主程序调用各种功能模块,或采用中断的方法实时地执行相应的服务模块来实现,并且按功能层次继续调用下一级模块。模块之间的联系是由数据接口(数据缓冲器和标志状态)来完成的。

硬件模块(模板)的连接有两种方法:一种是以主机模块为核心,通过设计者自行定义的内部总线(数据总线、地址总线和数据总线)连接其它模块;另一种是采用标准总线(例如 STD 总线)来连接所有模块。第一种方法由设计人员自行研制模板,电路结构简单,硬件成本低;第二种方法是设计人员可选购商品化的模板(也可自行研制),配接灵活方便,研制周期短,但硬件成本高。

1.2.2 智能仪表的设计研制步骤

设计、研制一台智能仪表大致上可以分为三个阶段：确定任务、拟制设计方案阶段；硬件、软件研制及仪表结构设计阶段；仪表总调、性能测定阶段（见图 1.2.1）。以下对各阶段的工作内容和设计原则作一简要的叙述。

1.2.2.1 确定任务、拟制设计方案

A 确定设计任务和仪表功能

首先确定仪表所要完成的任务和应具备的功能。例如仪表是用于过程控制呢还是数据处理，其功能和精度如何；仪表输入信号的类型、范围和处理方法如何；过程通道为何种结构形式，通道数是多少，是否需要隔离；仪表的显示格式如何，是否需要打印输出；仪表是否具有通信功能，并行还是串行；仪表的成本应控制在多少范围之内等等。以此作为仪表软、硬件的设计依据。另外，对仪表的使用环境情况及制造维修的方便性给予充分的注意。设计人员在对仪表的功能、可维护性、可靠性及性能价格比综合考虑的基础上，提出仪表设计的初步方案，并将其写成“仪表功能说明书（或设计任务书）”的书面形式。功能说明书主要有以下三个作用：

- (1) 作为用户和研制单位之间的合约，或研制单位设计开发仪表的依据。
- (2) 规定仪表的功能和结构，作为研制人员设计硬件、编制软件的基础。
- (3) 作为将来验收时的依据。

B 完成总体设计，选择确定硬件类型和数量

通过调查研究对方案进行论证，以完成智能仪表的总体设计工作。在此期间应绘制仪表系统总图和软件总框图，拟订详细的工作计划。完成了总体设计之后，便可将仪表的研制任务按功能模块分解成若干课题（子任务），去做具体的设计。

主机电路是智能仪表的核心，为确保仪表的性能指标，在选择单片机（或其它微处理器）时，需考虑字长和指令功能，寻址范围与寻址方式，位处理和中断处理能力，定时计数和通信功

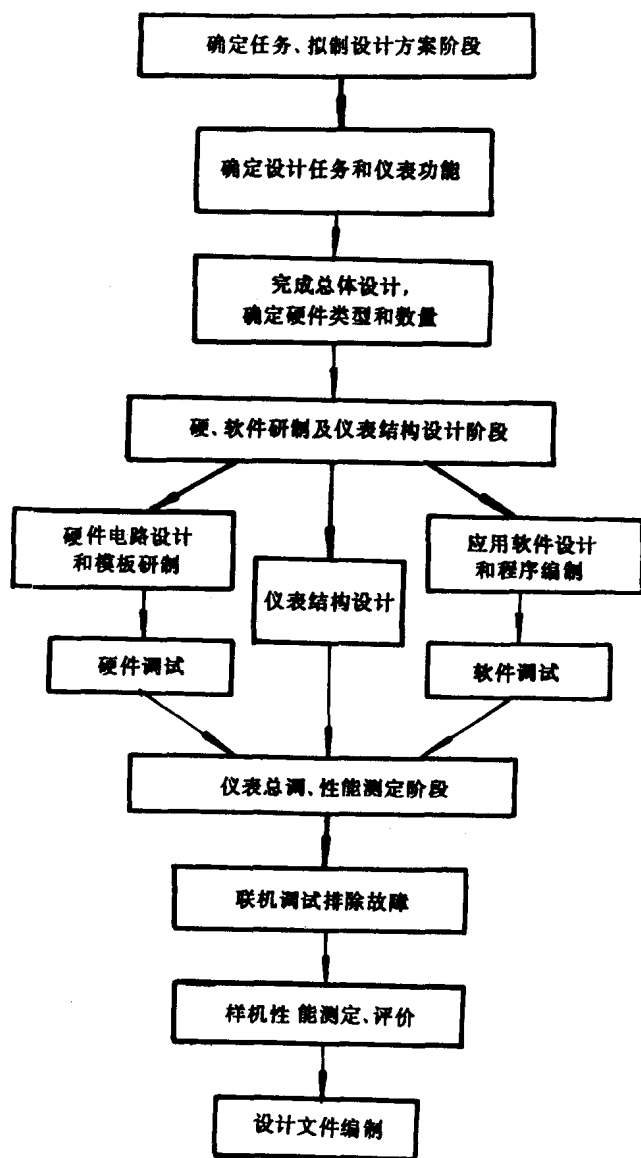


图 1.2.1 智能仪表开发过程

能,内部存储器容量的大小,硬件配套是否齐全,以及芯片的价格等等。典型的单片机有 Intel 公司的 MCS-48、MCS-51/52、MCS-96 等系列, Motorola 公司的 MC6805、MC68HC05、MC68HC11、MC68HC16 等系列的芯片。在内存容量要求不大,外部设备要求不多的智能仪表中,采用 8 位单片机较为适宜。若要求仪表运算功能强、处理精度高、运行速度快,则可选用 16 位单片机。

在智能仪表所需的硬件中,输入输出通道往往占有很大的比重,因此在估计和选择输入输出所需的硬件时,应考虑输入输出通道数,串行操作还是并行操作,数据的字长、传输率及传输方式等等。

由于硬件和软件具有互换性,设计人员要反复权衡仪表硬件和软件的比例。多用硬件可以简化软件设计工作,并使装置的性能得到改善。然而,这样会增加元器件数,成本相应提高。若采用软件来代替一部分硬件功能,虽可减少元器件数,但将增加编程的复杂性,并使系统的速度相应降低。所以,应当从仪表性能、器件成本、软件费用、研制周期等各方面考虑,对硬、软件比例作出合理的安排,从而确定硬件的类型和数量。

1.2.2.2 硬件、软件研制及仪表结构设计

A 硬件电路设计、研制和调试

硬件电路的设计包括主机电路,过程输入输出通道,人机接口电路和通信接口电路等功能模块。为提高设计质量,加快研制速度,通常采用计算机辅助设计(CAD)方法绘制电路逻辑图和布线图。设计电路时,尽可能采用典型的线路,力求标准化;电路中的相关器件性能须匹配;扩展器件较多时须设置线路驱动器;为确保仪表能长期可靠运行,还须采取相应的抗干扰措施,包括去耦滤波、合理的走线、通道隔离等。

完成电路设计、绘制好布线图后,应反复核对,确认线路无差错,才可加工印刷电路板。制成电路板后仍须仔细校核,以免发生差错,损坏器件。

由于主机部分是通过各种接口与键盘、显示器、打印机等部件相连接的,并通过输入输出通道,经测量元件和执行器直接连至被测和被控对象。因此,人机接口电路和输入输出通道的设计是研制仪表的重要环节,力求可靠实用。

如果逻辑电路设计正确无误,印刷电路板加工完好,那么功能模板的调试一般来说是比较方便的。模板运行是否正常,可通过测定一些重要的波形来确定。例如可检查单片机及扩展器件几个控制信号的波形与硬件手册所规定的指标是否相符,由此断定其工作正常与否。

通常采用开发工具来调试硬件,将其与功能模板相连,再编制一些调试程序,即可迅速排除故障,较方便地完成硬件部分的查错和调试任务。

B 应用软件设计、程序编制和调试

将软件总框图中的各个功能模块具体化,逐级画出详细的框图,作为编制程序的依据。编写程序可用机器语言、汇编语言或各种高级语言。究竟采用何种语言则由程序长度、仪表的实时性要求及所具备的研制工具而定。对于规模不大的应用软件,大多采用汇编语言来编写,因这可减少存储容量,降低器件成本,节省机器时间。研制较复杂的软件且运算任务较重时,可考虑使用高级语言来编程。C51、C96 交叉编译软件是近年来较为流行的一种软件开发工具。它采用 C 语言编写源程序,编程方便,软件可读性强,易于修改和扩充。该软件功能强,编译效率高,有助于开发规模大、性能更完善的应用软件。编完程序,经汇编或编译生成目标码,再经调试通过后,可直接写入 EPROM 中。