

摘要

针对人工点滴输液方式的缺陷和当前高校实践教学的迫切需要,作者研究、设计和实现了基于 MCGS 组态软件的液体点滴监控系统模型。该系统模型综合运用了先进的组态软件、单片机控制、信号检测与处理、语音等技术对液体点滴情况进行有效的监视和控制,并可对各种数据进行方便的统计和查询。系统设计目标是将上位机和下位机结合起来实现远程监控、现场监控和医疗管理等功能。

本文首先分析了液体点滴监控系统的总体方案设计,接着详细论述了凌阳单片机 SPCE061A 的 MCGS 设备驱动程序的开发方法和过程、下位机监控系统的软硬件设计和实现原理以及上位机监控系统程序的构建方法,最后给出了整个监控系统的测试结果。

关键词: 液体点滴, SPCE061A 单片机, MCGS·组态软件, 监控

ABSTRACT

Aiming at the defect of manual dropping transfusion mode and the demand for the practice teaching in advanced school, author research , design and implement the liquid dropping supervisory and control system model based on MCGS .Author synthetically use the technique of advanced configuration soft, control of MCU, detection and processing of signals,speech for supervisory and control of the circs of liquid dropping,we can make statistics and enquiry of various datas . The target of the design is realizing long-distance supervisory and control, locale supervisory and control , medical management etc.

First of all, the total scheme of the design is analyzed, after that author particularly discuss the design of driver for the SPCE061A based on MCGS, the design and realizing of soft and hardware of secondary controller, and the construct way of MCGS supervisory and control program, at last, author present the test result of whole supervisory and control system.

KEY WORDS: liquid dropping, SPCE061A, MCGS Configuration, Supervisory and Control

Li Qin(information engineering)

Directed by prof. Ling Guang Sheng

摘要

针对人工点滴输液方式的缺陷和当前高校实践教学的迫切需要,作者研究、设计和实现了基于 MCGS 组态软件的液体点滴监控系统模型。该系统模型综合运用了先进的组态软件、单片机控制、信号检测与处理、语音等技术对液体点滴情况进行有效的监视和控制,并可对各种数据进行方便的统计和查询。系统设计目标是将上位机和下位机结合起来实现远程监控、现场监控和医疗管理等功能。

本文首先分析了液体点滴监控系统的总体方案设计,接着详细论述了凌阳单片机SPCE061A的MCGS设备驱动程序的开发方法和过程、下位机监控系统的软硬件设计和实现原理以及上位机监控系统程序的构建方法,最后给出了整个监控系统的测试结果。

关键词：液体点滴，SPCE061A 单片机，MCGS·组态软件，监控

ABSTRACT

Aiming at the defect of manual dropping transfusion mode and the demand for the practice teaching in advanced school, author research, design and implement the liquid dropping supervisory and control system model based on MCGS. Author synthetically use the technique of advanced configuration soft, control of MCU, detection and processing of signals, speech for supervisory and control of the circs of liquid dropping, we can make statistics and enquiry of various datas. The target of the design is realizing long-distance supervisory and control, locale supervisory and control, medical management etc.

First of all, the total scheme of the design is analyzed, after that author particularly discuss the design of driver for the SPCE061A based on MCGS, the design and realizing of soft and hardware of secondary controller, and the construct way of MCGS supervisory and control program, at last, author present the test result of whole supervisory and control system.

KEY WORDS: liquid dropping, SPCE061A, MCGS Configuration, Supervisory and Control

Li Qin(information engineering)

Directed by prof. Ling Guang Sheng

声 明

本人郑重声明：此处所提交的硕士学位论文《基于 MCGS 的液体点滴监控系统模型的设计与实现》，是本人在华北电力大学攻读硕士学位期间，在导师指导下进行的研究工作和取得的研究成果。据本人所知，除了文中特别加以标注和致谢之外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得华北电力大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名: 李琴 日期: 2006.3.10

关于学位论文使用授权的说明

本人完全了解华北电力大学有关保留、使用学位论文的规定,即:①学校有权保管、并向有关部门送交学位论文的原件与复印件;②学校可以采用影印、缩印或其它复制手段复制并保存学位论文;③学校可允许学位论文被查阅或借阅;④学校可以学术交流为目的,复制赠送和交换学位论文;⑤同意学校可以用不同方式在不同媒体上发表、传播学位论文的全部或部分内容。

(涉密的学位论文在解密后遵守此规定)

作者签名： 李琴

导师签名: 梁志伟

日期: 2006.3.10

日期: 2006.3.10

第一章 绪论

1.1 课题的背景和研究意义

1.1.1 人工点滴输液现状

目前,医院对点滴输液装置的控制方法是将液体容器挂在架子上由护士通过软管夹对胶管口径的压紧或放松来控制滴速。这样做有几个弊端:第一,控制不方便。有经验的医护人员可以根据药剂的特性对滴速进行控制,而一般的病人却无法做到,做的不好会有一定的危险性。第二,医护人员必须到病床前去操作控制,增加了医护人员临床的次数,增加了医护人员受感染的几率。第三,不能对各种数据进行自动和直观的统计、查询和分析,缺乏一个功能强大的人机交互界面。

1.1.2 国内外智能点滴输液控制系统的研究现状

国外对智能型输液装置的研制较早,如日本、美国和德国等国家上世纪 80 年代末就进行了智能型输液装置的研制,现在市场上流行的大多是国外产品,类型多样,性能较好,如日本 JMS 株式会社的 OT-601 型输液泵(控制精度为 10%)和 SP-500 型注射泵,美国、德国、以色列等国家也有性能较好的产品。

国内对输液装置的研制起步较晚,大都在 90 年代中期开始研究,市场上也有一些国产输液装置,如北京科力丰高科发展有限责任公司的 ZNB 系列产品,深圳康福特公司也有输液装置产品。不过总体来说,种类较少,性能也需改进。

1.1.3 高校人才培养对实践教学的迫切需要

随着数字化、信息化技术对各行各业的渗透,旧岗位结构的变化以及新岗位的出现,对实践性教学环节应具有技术的先进性、综合性、应用性的要求日趋加强。教育部《关于进一步加强高等学校本科教学工作的若干意见》指出,高等学校应着眼于国家发展和人的全面发展需要,坚持知识、能力、素质协调发展,注重能力培养,着力提高大学生的学习能力、实践能力和创新能力。要坚持以社会需求为导向,深化教学改革,构建主动适应经济社会发展需要的人才培养体系。

实践教学是高等教育的重要教学环节,对指导学生理论联系实际,培养学生创新精神与综合素质具有不可替代的作用。开展先进性、系统性、综合性实践教学环

节的重要性还有以下几方面：1.有利于培养学生对问题的敏感性。2.有利于形成既合作又竞争的气氛。对于一些关键部件、关键技术、关键原理需要深层次的独立思考，以便找到解决问题的突破口。与此同时，对于一些复杂、综合的问题则往往需要互相讨论、互相合作，以便一步步地找到解决问题的方法。3.有利于形成激励机制下的愉悦教育。学生接触工程实践之初，充满了疑惑，并倍感工程项目的复杂性，而每当完成一小步，学生会豁然开朗，认识到原来复杂的问题就是由几门简单的技术组合而成，内心充满了解决问题的喜悦。因此，为了缩短学校与社会、课堂与企业的距离，突出培养学生的综合应用和设计能力，师生必须共同重视实践教学。

现阶段实践性教学环节有它的薄弱之处，大都处于滞后、分散、针对性不强、系统性不够、综合性不强等低层次循环状态，师生重视程度不够，造成一方面学生找不到工作，另一方面某些岗位的高级应用技术人才紧缺。为此必须充分重视实践教学环节，充分发挥出实践性教学的最大功效。其有效措施之一就是配合一些优秀的实验教学模型开展设计性、系统性、综合性的实验和实习教学。对实验教学模型有几点主要要求：1.系统性；2.新技术的综合性；3.设计性；4.直观性。

综合对上述问题的分析和思考，提出了“基于 MCGS 的液体点滴监控系统模型的设计与实现”的论文课题。

1.2 论文所完成的主要工作.

作者针对人工点滴输液方式的缺陷和当前高校实践教学的迫切需要，研究、设计和实现了基于 MCGS 组态软件的液体点滴监控系统模型。该系统模型综合运用了先进的组态软件、单片机控制、信号检测与处理、语音等技术对液体点滴速度进行方便的控制、对各种数据进行自动、直观的统计。

基于 MCGS 组态软件的液体点滴监控系统模型的设计目标是将上位机（PC 机）和下位机结合起来实现远程监控、现场监控和医疗管理等功能。围绕这一目标开展了以下几个方面工作：

1. 设计开发了凌阳单片机 SPCE061A 的 MCGS 设备驱动程序

液体点滴速度监控系统要求上、下位机都能对点滴速度、点滴是否有异常等情况进行监控，下位机采用凌阳 SPCE061A 作为控制系统的中心，用于对点滴速度、点滴是否有异常等进行监控，然后传送至上位机供 MCGS 进行处理、存储和管理，MCGS 没有提供凌阳 SPCE061A 的设备驱动程序，因此需要用 VB 来开发 SPCE061A 的 MCGS 设备驱动程序。

2. 以凌阳单片机 SPCE061A 为核心设计和实现了液体点滴下位机监控系统

系统采用主站控制从站的有线监控系统方式实现医疗输液过程的控制，单片机是监控系统下位机的核心，由它完成对监控对象的选择、点滴速度的设定、液位警

戒线的报警、数据采样及处理、监控内容显示以及与上位机数据通信等功能。

3. 在 MCGS 环境下设计和实现了液体点滴上位机监控系统

要实现远程监控，必须设计上位机监控程序，来集中监视和控制输液室或病房的点滴输液监控装置，方便护士在护士室监控若干人的输液情况，从而监视和控制病人的输液过程，减轻护士的工作量，实现医院护理自动化。

第二章 液体点滴监控系统的总体方案设计

基于 MCGS 组态软件的液体点滴监控系统模型采用上、下位机的工作方式，利用 MCGS 组态软件和 SPCE061A 单片机联合设计开发，实现了对液体点滴的速度和液位的全面监控。本章分析讨论了该系统的总体结构设计、上下位机软硬开发平台的选择以及监控系统模型的教学功能设计。

2.1 液体点滴监控系统的总体结构设计

液体点滴监控系统的总体结构如图 2-1 所示，整个系统主要由以下几部分组成：以 PC 机为载体的 MCGS 监控系统程序、挂接到 MCGS 环境下的设备驱动程序、下位机与上位机的串行通信模块、SPCE061A 单片机、储液检测模块、点滴速度检测模块、点滴速度的电机控制模块、键盘和 LED 数码显示模块、声光报警模块、乐曲演奏模块、LCD 显示模块等。

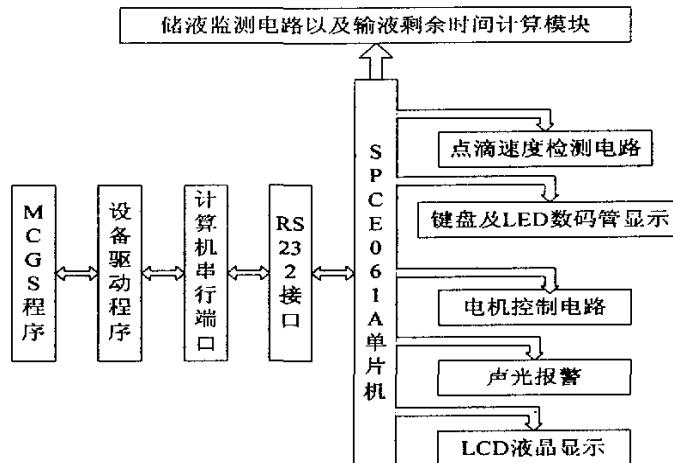


图 2-1 系统整体结构

储液检测电路用来检测剩余的储液量是否到达报警液位，若到达，系统就通过报警电路对危险情况进行报警。键盘和 LED 数码显示模块用来设置并显示液体点滴速度值。点滴速度检测电路不停地检测液体点滴的速度值。系统将设置的点滴速度值与当前检测到点滴速度值进行比较，然后通过电机控制系统来调节点滴速度达到设置值并显示。PC 机上的 MCGS 程序通过设备驱动程序实时采集现场数据后，进行

数据分析,完成现场模拟和监控、数据存储分析、重要参数设定、显示和打印各种数据报表及实时或历史曲线图。

2.2 液体点滴监控系统软硬件开发平台的选择

液体点滴监控系统由上位机和下位机组成。上位机系统采用组态软件构建,因为组态监控技术为实施数据采集和过程监控提供了基础平台,组态软件具有功能强大、二次开发简便、开发周期短、通用性强、可靠性高等优点。下位机系统由凌阳单片机 SPCE061A 模块电路、信号检测与处理电路、键盘与显示电路、控制电路等构成,不仅完成对液体点滴情况的全面监控,而且还起着与上位机通信的作用。

2.2.1 上位机系统软件开发平台的选择

上位机监控程序可以采用组态软件开发,也可以采用 Visual Basic 等软件来开发, Visual Basic 是一种功能非常强大的编程语言,它是基于窗口和面向对象的程序设计,包含有数百条语句、函数和关键词,具有结构化的事件驱动编程模式, ActiveX 技术可以使用其它应用程序提供的功能,具有快速开发、易学易用、效率高、界面制作美观方便等优点,一直成为软件开发人员的首选工具。但是如果采用组态软件来开发,其优点更是突出。

一、采用组态软件来开发监控系统的优点

1、采用组态软件扩展性更强。采用组态软件来开发监控系统具有很好的扩展性,只要开发出其设备驱动程序,扩展应用系统很容易,不需要改动其设备驱动程序就可以开发出很多的应用系统,如果采用 Visual Basic 开发新的应用系统需要做大量的工作。

2、采用组态软件来开发可以缩短开发周期、提高运行效率。这里需要指出采用组态软件作为开发平台不仅仅是采用组态软件,还需要结合 VB 来开发组态软件设备的驱动程序, MCGS 为用户提供了 OLE Automation 技术及网络支持,允许用 Visual Basic 操作 MCGS 中的对象,同时还提供了一套开放的可扩充接口,用户可根据自己的需要用 VB 编制特定的功能构件来扩充系统,最终实现 MCGS 与 VB 的交互链接,从而缩短了开发周期、大大提高了运行效率。

可以看出采用组态软件来开发监控系统有很大的优势,因此本系统采用组态软件来开发。

二、MCGS 组态软件的应用

大多数组态软件都提供了很多灵活的技术手段,基本上可以满足用户的要求。由于用户要求的多样化,直接用户对监控系统的需求不可能固定为单一模式,因此

直接用户的监控系统是始终需要“选择”和“定制”的。目前国内的监控组态软件以组态王、力控、MCGS (Monitor and Control Generated System: 监视与控制通用系统) 等为主要代表, 它们是在借鉴国际同类产品的基础上发展起来的, 具有较高的起点, 发展速度很快, 在可靠性和稳定性方面, 已接近于国际同类产品。

MCGS 是一套 32 位工控组态软件, 可稳定运行于 Windows98/2000/XP/NT 操作系统之上, 集动画显示、流程控制、数据采集、设备控制与输出、网络数据传输、双机热备、工程报表、数据与曲线等诸多强大功能于一身, 并支持国内外众多数据采集与输出设备, 广泛应用于石油、电力、化工、钢铁、矿山、冶金、机械、纺织、航天、建筑、材料、制冷、交通、通讯、食品、制造与加工业、水处理、环保、智能楼宇、实验室等多种工程领域。MCGS 提供以 OLE 自动化技术为基础的开放式扩充接口, 允许用户使用 VB 来快速编制各种设备驱动构件、动画构件和各种策略构件, 通过 OLE 接口用户可以方便地定制自己特定的系统。因此, 笔者在设计系统时, 选用了北京昆仑通态自动化软件科技有限公司开发的全中文工控组态软件 MCGS。

2.2.2 下位机系统硬件开发平台的选择

下位机系统的核心是微控制器, 微控制器可以采用单片机也可以采用 PLC (可编程序逻辑控制器), 该系统选用单片机。单片机的选择决定于以下多方面的因素:

1. 单片机的速度、程序存储器容量、I/O 引脚数量。
2. 单片机的增强功能, 例如看门狗、指针数量、串口数、RTC (实时时钟)、EEPROM 和 RAM 的扩展接口、CAN 接口、I2C 接口、SPI 接口、USB 接口等。
3. 在线可编程还是使用编程器编程。
4. 封装形式: DIP、PLCC、贴片等。
5. 工作电压和功耗。
6. 价格。
7. 编程语言: 汇编语言、C 语言。
8. 需要的外围电路的多少。

随着单片机功能集成化的发展, 其应用领域也逐渐地由传统的控制, 扩展为控制处理、数字处理以及数字信号处理等领域。单片机有 51、AVR、PIC、凌阳 SPCE061A 等很多种可供选择。SPCE061A 是台湾凌阳科技公司推出的 16 位微控制器, 因为它具有以下优良的性能特点, 所以选择 SPCE061A 单片机作为下位机系统的核心微控制器。

- 内置 2K 字 SRAM, 32K 字 FLASH;
- 可编程音频处理;
- 2 个 16 位可编程定时器/计数器 (可自动预置初始计数值);
- 2 个 10 位 DAC 输出通道;
- 7 通道 10 位电压模数转换器 (ADC) 和单通道声音模数转换器;
- 32 位通用可编程输入/输出端口;
- 14 个中断源可来自定时器 A/B、时基、2 个外部时钟源输入、键唤醒;

- 具备串行设备接口；
- 具有低电压复位(LVR)和低电压监测(LVD)功能。

2.3 液体点滴监控系统的教学功能设计

在教学过程中，如果教师能为学生营造一个多感官共同参与的学习氛围，充分让学生动眼、动耳、动脑、动手来感知事物、领悟概念、掌握原理，这样就能使学生的学习由被动变为主动，大大提高学生学习的积极性和思维的能动性。

基于 MCGS 组态软件和凌阳单片机构建完成的液体点滴监控系统就是专门为此次设计开发的一个实验教学模型，它具有以下多个方面教学功能，以满足当前实践教学的紧迫需要。

1. 便于系统地学习组态软件的使用与应用开发方法

MCGS 组态软件应用面向窗口的设计方法，以窗口为单位，构造用户运行系统的图形界面，增加了可视性和可操作性，使得 MCGS 的组态工作既简单直观又灵活多变。用户可以使用系统的缺省构架，也可以根据需要自己组态配置，生成各种类型和风格的图形界面。

2. 便于学习 MCGS 设备驱动程序的设计方法

不论是对软、硬件开发工程人员还是对于第三方的专业、业余开发者或用户来说，能够开发 MCGS 设备驱动程序越来越成为必须的技能要求。本文第三章详细介绍了基于 MCGS 组态软件的凌阳单片机设备驱动程序的设计方法和测试方法，为用户（包括学生）开发设备驱动程序提供了设计思路。

3. 便于学习信号的检测、处理和控制方法

信号的检测、处理和控制技术是系统自动控制的基础。论文详细分析和讨论了如何用光电传感器对液体点滴速度和液位进行检测、对检测信号进行有效处理的方法、如何用步进电机去控制液体点滴速度等方面的问题。

4. 便于开展上下位机联动实验的教学

该液体点滴监控系统既可以以下位机形式独立工作，也可以与上位机联机工作。这样就可以做到上下位机互动或联动，既可用单片机指挥 PC 屏幕上控件的动作，也可以用 PC 屏幕上的控件指挥单片机动作，以实现现场或远端对液体点滴输液系统情况的全面监控。学生既可在 PC 屏幕上自由组态设计，也可在单片机上自由开发、下载和实验。这样可以使实验教学生动、形象、直观。

第三章 凌阳单片机 SPCE061A 的 MCGS 设备驱动程序的开发

MCGS 通过设备驱动程序与外部设备进行数据交换, 包括数据采集和发送设备指令。MCGS 负责在运行环境中调用相应的设备驱动程序, 将数据传送到工程中各个部分, 完成整个系统的通讯过程。

MCGS 作为一种方便有效的通用工控软件, 它提供了国内外各种常用的工控设备的驱动程序。但在实际应用中, 因为所用设备的特殊性或实际工程的需要, 允许用户根据需要来定制自己的设备驱动程序。液体点滴监控系统下位机的核心控制器是凌阳单片机 SPCE061A, MCGS 中没有它的设备驱动, 因此需要利用 MCGS 的高级开发包所提供的一套接口规范来设计编写 SPCE061A 的设备驱动程序, 其中又以 MCGS 上下位机的串口通信最为简单和基本。

鉴于 VB6.0 的通用性、简单性和以二进制码来编译执行程序, 它完全能满足功能构件所需的速度要求, 而且 MCGS 提供了利用 VB 来开发设备驱动程序的方法, 因此, 作者选用 VB6.0 来开发 MCGS 的设备驱动程序。

本章首先简单介绍了 MCGS 组态软件和 VB6.0 开发环境, 然后详细论述了 VB6.0 环境下利用 OLE(对象链接与嵌入)技术设计和编写 MCGS 上下位机的串口驱动程序的方法和过程。

3.1 MCGS 组态软件的功能和特点

3.1.1 组态软件的发展

组态的概念最早来自于英文 Configuration, 其含义是使用软件工具对计算机及软件的各种资源进行配置, 按照预先的设置来自动执行特定任务, 以满足使用者的要求。监控组态软件是面向监控和数据采集 (supervisory control data acquisition, SCADA) 的软件平台工具, 它包含两层含义: 一是分布式的智能数据采集系统, 即通常所说的下位机实时检测与控制软件系统; 二是数据处理和显示系统, 即上位机 (中央控制机) HMI (human machine interface) 人机界面系统。随着它的快速发展, 实时数据库、实时控制、SCADA、通信及联网、对 I/O 接口的广泛支持等已成为它的主要内容, 它具有丰富的设置项目, 使用方式灵活, 功能强大。

监控组态软件是工业应用软件的一个组成部分, 它是伴随着分布式控制系统 (distributed Control system) 及计算机控制技术的日趋成熟而发展起来的, 是 DCS 的商品化应用促进了监控组态软件概念的普及和发展。随着微处理器及个人计

计算机技术的飞速发展，自动化监控设备的价格得以大幅度降低，体积也逐渐缩小，另外计算机网络技术的发展使得监控设备之间的互联互通变得简便易行，这一切都促进了监控组态软件的普及与推广。同时监控组态软件也促使自动化技术走出工业应用的狭小范围，向楼宇自动化、农业自动化、环境（监测、保护）自动化等领域渗透，加快了整个社会的信息化步伐，提高了自动化工程的工作效率，减少了系统的维护和升级费用。

在一个自动监控系统中，投入运行的监控组态软件是系统的数据搜集处理中心、远程监视中心和数据转发中心，它与各种控制和检测设备（如 PLC、智能设备、DCS 等）共同构成快速响应/控制中心。

3.1.2 MCGS 工控组态软件的功能和特点

MCGS 工控组态软件是目前国内监控组态软件的杰出代表之一，它具有以下功能和特点：

- 简单灵活的可视化操作界面

MCGS 采用了“面向窗口”的设计方法，以窗口为单位，构造用户运行系统的图形界面，增加了可视性和可操作性，从而使 MCGS 的组态工作既简单直观，又灵活多变。利用丰富的“动画组态”功能，快速构造各种复杂生动的动态画面，同时，MCGS 为用户提供了丰富的动画构件，模拟工程控制与实时监测作业中常用的物理器件的动作和功能。用户可以使用系统的缺省构架，也可以根据需要自己组态配置，生成各种类型和风格的图形界面，包括 DOS 风格的图形界面、标准 Windows 风格的图形界面以及带有动画效果的工具条和状态条。

- MCGS 为解决工程监控问题提供了丰富多样的手段

从设备驱动（数据采集）到数据处理、报警处理、流程控制、动画显示、报表输出、曲线显示等各个环节，均有丰富的功能组件和常用图形库可供选用，用户只需根据工程作业的需要和特点，进行方案设计和组态配置，即可生成用户应用软件系统。

- 简单方便地编写复杂的控制程序

首先设立设备工具箱，针对外部设备的特征，用户从中选择某种构件，设置于设备窗口内，赋予相关的属性，建立系统与外部设备的连接关系，即可实现对该种设备的驱动和控制。其次设立对象元件库，解决了组态结果的积累和重新利用问题。组态时，可把制作完好的对象（包括图形对象、窗口对象、策略对象以至位图文件等等）以元件的形式存入图库中，也可把元件库中的各种对象取出，直接为当前的工程所用。有的控制系统用户甚至可以不写一行程序就能够构成自己的控制系统。

- 实时数据库为用户分部组态提供了极大方便

MCGS 由五部分组成,其中的“实时数据库”是整个系统的核心。在生成用户应用系统时,每一部分均可分别进行组态配置,独立建造,互不相干;而在系统运行过程中,各个部分都通过实时数据库交换数据,形成互相关联的整体。实时数据库是一个数据处理中心,是系统各个部分及其各种功能性构件的公用数据区。各个部件独立地向实时数据库输入和输出数据,并完成自己的差错控制。MCGS 中数据的存储不再使用普通的文件,而是用数据库来管理一切。组态时,系统生成的组态结果是一个数据库;运行时,数据对象和报警信息的存储也是一个数据库。利用数据库来保存数据和处理数据,提高了系统的可靠性和运行效率,同时,也使其它应用软件系统能直接处理数据库中的存盘数据。

- 强大的网络功能

如果需要通过互联网来使群控更加有效和直观,就需要应用 MCGS 网络版组态软件来设计监控程序,利用网络版组态软件有以下几个优点:先进的瘦客户机机制;系统开发、维护工作量小;版本升级简单易行;充分利用现有资源;支持多种网络结构。

3.2 MCGS 设备驱动构件的实现方法和运行机制

3.2.1 MCGS 设备驱动构件的实现方法

MCGS 提供了开放性的可扩充接口,用 Active DLL 构件来实现设备驱动程序,通过规范的 OLE (Object Linking and embedding) 接口挂接到 MCGS 中,使其构成一个整体,所以 MCGS 把设备驱动程序叫做设备驱动构件。OLE 是一个可扩充的开放标准,即是一个结构可扩展、基于对象集成的、应用程序共享对象的工业标准,其实质是在应用程序中嵌入其它程序提供的对象和数据,从而获得特定功能的程序设计方法,主要用于实现不同软件之间的相互操作。MCGS 把和操作系统底层交换的内容全部包装起来,以函数调用的方式提供,同时, MCGS 的高级开发向导能自动生成设备驱动程序的框架。鉴于 Visual Basic 语言的通用性和简单性,使用 VB 来开发单片机驱动。MCGS 的实现方法和原理与标准的 Active DLL 完全一致,但 MCGS 规定了一套接口规范,只有遵守这些接口规范的 Active DLL 才能用作 MCGS 的设备驱动构件。

在 MCGS 中实现设备驱动的基本方法是:在设备窗口内配置不同类型的设备构件,并根据外部设备的类型和特征,设置相关的属性,将设备的操作方法如硬件参数配置、数据转换、设备调试等都封装在构件之中,以对象的形式与外部设备建立数据的传输通道连接。系统运行过程中,设备构件由设备窗口统一调度管理,通过通道连接,向实时数据库提供从外部设备采集到的数据,从实时数据库查询控制参

数, 发送给系统其它部分, 进行控制运算和流程调度, 实现对设备工作状态的实时检测和过程的自动控制。

3.2.2 MCGS 可扩充的设备驱动构件

可扩充的设备驱动构件能实现多种设备驱动的方式, 可以使用独立的设备驱动构件形式, 也可以使用父子设备驱动结构形式; 在设备驱动构件中, 可以使用查询的工作方式, 也可以使用中断的方式, 来高效完成设备驱动的任务; 还可以实现设备热更换的功能; 设备驱动构件的概念还可以扩充到所有的高可靠性要求、数据输入输出、执行命令等任务方面。

3.2.3 MCGS 设备驱动构件的运行机制

MCGS 把设备驱动程序实现成 Active DLL 构件, 通过调用构件的不同接口来完成设备驱动所需的功能, 下面详细讲解 MCGS 是如何来操作构件各接口以及按什么样的顺序来操作。

①在 MCGS 中要操作一个设备, 必须首先确保对应设备已被选入设备工具箱。在设备构件管理工具中, 登记选择需要使用或新编的设备。MCGS 将首先检查该构件是否满足接口要求, 同时调用设备构件的 GetDevName 接口, 返回设备构件的类型名称, 该类型名称最后显示在设备工具箱中, 用于标识对应的设备构件。

②从工具箱中把需要使用的设备构件选入到 MCGS 的设备窗口中, 此时, MCGS 依次调用 InitNewDev 接口, 对设备的属性进行初始化设置; 读取 DevType 属性, 确定设备的类型 (=0 父设备, =1 子设备); 读取 DevStyle 属性, 确定设备的类别 (对独立的设备构件可以任意指定, 但不能与 MCGS 已有的设备类别相同; 对子设备, 应设置成其父设备构件的类别)。

③打开设备属性窗口, 设置设备的属性。在属性窗口的基本属性页中, MCGS 把构件的所有读写属性列在该页中, 供用户修改。当用户按设备在线帮助对应的按钮时, MCGS 调用 GetDevHelp 接口, 以显示设备构件的在线帮助; 按设备内部属性对应的按钮时, MCGS 调用 SetDevPage 接口, 以显示设备构件的特定属性页。(如对应的接口为空, 则不执行任何功能)。MCGS 调用 GetChlType 接口, 读取设备通道的信息, 显示在通道连接属性页中。

④当切换到设备调试属性页中, MCGS 定时调用 CollectDevDat 接口, 把采集进来的数据显示在窗口中。

⑤在 MCGS 运行环境中, 首先调用一次 InitDevRun 接口, 以便对设备进行一些必须的初始化工作。

⑥在运行的过程中, MCGS 按设定的采集周期, 调用 CollectDevDat 接口, 从设备采集或向设备输出数据。

运行时, 当用户在脚本程序中调用!SetDevice 函数或在运行策略中调用设备操作功能构件时, MCGS 调用 RunDevCommand 接口, 以执行设备的特定功能。

3.3 MCGS 设备驱动程序的开发环境 Visual Basic 6.0

Visual Basic 是 Microsoft 公司推出的强有力的系列开发软件之一, 以其实用、方便、快捷、开发周期短、广泛而强大的功能越来越被广大编程人员所亲赖。尽管 Visual C++ 在灵活性、代码紧凑、运行速度快及底层开发等方面 Visual Basic 无法与之比拟, 但编写 C++ 应用程序过长的调试和开发周期确实有点不适应当今的应用程序开发, 因此有些专业的编程人员放弃了 Visual C++ 而改用了 Visual Basic 之类的编程工具。

要创建 Active DLL 构件, 首先要清楚 Visual Basic 程序的基本框架。在 Visual Basic 中, 基本模块是以文件形式体现的, 有三种文件级模块: 标准模块、窗体模块和类模块, 文件模块中包含过程、函数、属性、方法和事件等内容, 可以把它们看成单元级模块。在 Visual Basic 中单元级模块中不能再进行模块细化, 因此单元模块是 Visual Basic 最小的模块单位。

标准模块 (文件扩展名为.BAS) 是具有结构化特征的模块。它是一个包含许多过程、函数等单元模块的仓库, 被用来为应用程序中的其它模块提供能够访问的过程或函数。标准模块并不一定属于某个专门程序, 因此, 可以将一些经常使用的过程、函数保存在标准模块中, 使这些经常使用的基本程序单元能够被许多不同的应用程序重用。

类模块 (文件扩展名为.CLS) 是面向对象编程的基础。类模块为用户创建新的对象提供了条件。类模块中可以包含用户定义的属性、方法和事件等单元成分。

窗体模块 (文件扩展名为.FRМ) 是用来构造前端用户界面, 通过在窗体上放置空间, 可以实现对用户界面的可视化设计。窗体是一种特殊的类模块, 窗体上的控件是窗体对有关控件类的引用, 而窗体和控件的事件过程, 则是窗体程序中最重要的单元模块成分。

通过 Visual Basic 创建应用程序时, 首先建立标准模块、窗体模块和类模块等文件级模块, 然后在其中接着建立过程、函数、属性、方法和事件等单元级模块。

3.4 MCGS 设备驱动程序的具体设计

计算机与计算机或计算机与终端之间的数据传送可以采用串行通讯和并行通

讯两种方式,由于串行通讯方式具有使用线路少、成本低等优点,所以在本系统中采用串行通讯方式。串行通讯要求通讯双方都采用一个标准接口,使不同的设备可以方便地连接起来。

SPCE061A 单片机的 UART 模块提供了一个全双工标准接口,通过标准 232 接口与 PC 机相连接,按照字节传输数据,数据帧格式为:1 位启动位、8 位数据位、1 位奇偶校验位、1 位停止位,设计中没有利用校验位。利用 MCGS 高级开发向导生成设备驱动程序的框架,启动组态环境,打开一个 MCGS 工程,运行工程组,设置串口父设备属性里的参数分别为:通信波特率为 9600,数据位数为 8 位,串行端口号为 COM1,数据校验方式为无校验,停止位为 1 位,数据采集方式为同步采集,通讯方式为本地串口通信,如图 3-1 和图 3-2 所示。

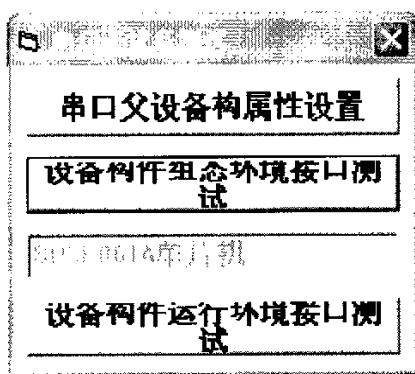


图 3-1 串口子设备测试窗口

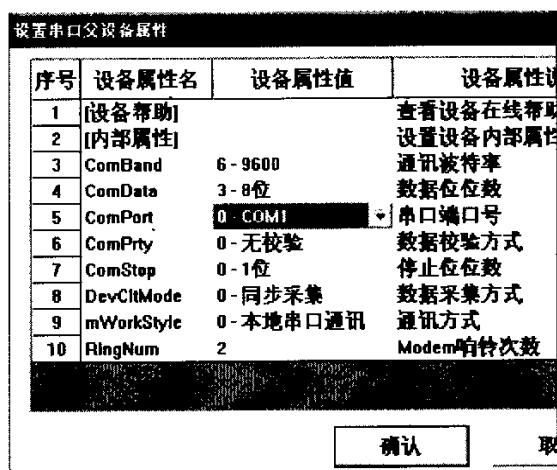


图 3-2 串口父设备属性设置窗口

3.4.1 MCGS 设备驱动程序的总体结构

本系统的设备驱动程序需要建立一个 ActiveX DLL 部件,创建 ActiveX DLL 需要创建一个工程组,包括部件工程和测试工程,具体来说,就是实现部件中类的接口,实现部件所需要的窗体。设备驱动程序的总体结构图如图 3-3 所示。

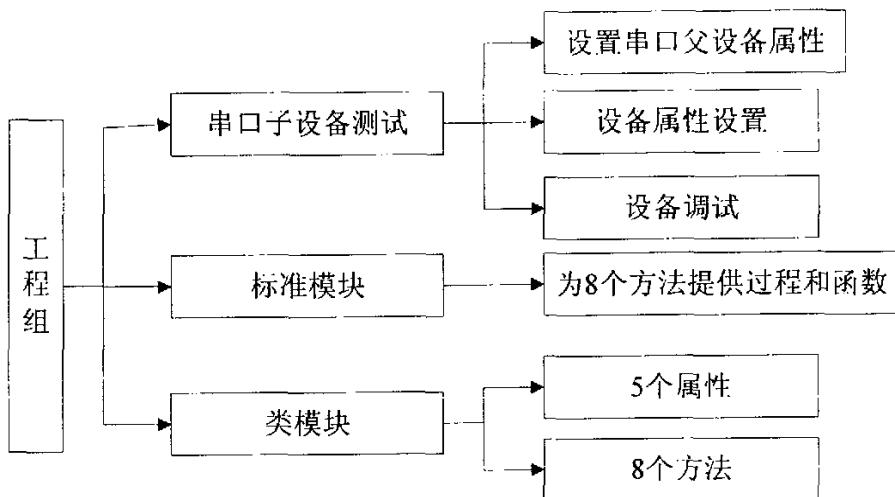


图 3-3 设备驱动程序总体结构图

3.4.2 设备驱动程序各个模块的详细设计

整个工程组包括串口子设备测试模块、标准模块和类模块三部分，其中标准模块是为类模块提供过程和函数的，因此把标准模块放在 5 个属性和 8 个方法中介绍。

一、串口子设备测试模块

串口子设备测试模块 frmTest 是为测试工程建立的窗体模块，如图 3-1 所示，它是总的测试模块，包括设置串口父设备属性、设备属性设置和设备调试模块。可以点击它的串口父设备属性设置按钮、设备构件组态环境接口测试按钮和设备构件运行环境接口测试按钮来测试部件工程。

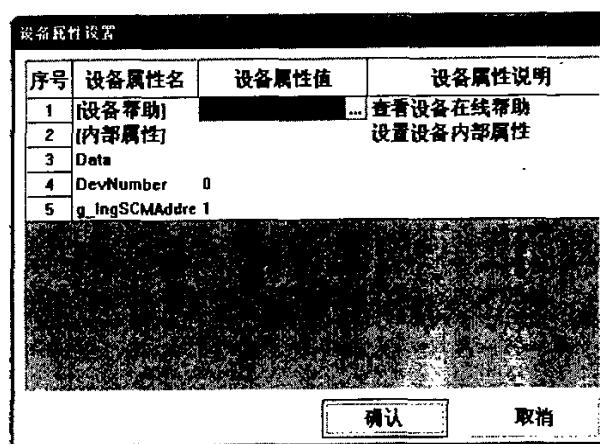


图 3-4 设备属性设置窗口

1. 串口父设备属性设置模块

串口父设备属性设置构件是用来完成对串口的基本操作和参数设置，如图 3-2

所示。

2. 设备属性设置模块

点击设备构件组态环境接口测试按钮进入设备属性设置窗口，如图 3-4 所示。设备属性设置窗口用来设置设备帮助、内部属性以及设备号等，其中设备号为以后扩展设备作准备。

3. 设备调试模块

点击设备构件运行环境接口测试进入设备调试窗口，如图 3-5 所示。设备调试模块用来调试设备驱动程序，从调试窗口可以看到最终的调试结果。图中第一列的数据为通道号，在设备驱动构件设计中经常要用到通道。我们把输入输出装置读取数据和输出数据的通道称为设备通道。建立设备通道和实时数据库中数据对象的对应关系称为通道连接。建立通道连接的目的是通过设备构件确定采集进来的数据送入实时数据库的什么地方，或从什么地方取用数据。建立通道连接要进入“通道连接”属性页，该属性是将设备的数据传输通道与实时数据库中对应的数据变量之间，建立连接关系。应当注意，设备通道与数据变量在类型上必须一致，否则连接无效。窗口中的“快速连接”按钮，用来对同一系列的数据变量与一组设备通道建立连接。

对通讯状态通道调试时，通道值一列显示设备通讯状态，0 表示通讯成功，1 表示通讯失败。对模拟量输入通道调试时，通道值一列显示模拟量输入通道的实际测量值。对数字量通道调试时，通道值一列显示数字量通道的开关状态，为 0 表示关，为 1 表示开。

通道	数据类型	通道类型	对应值
01	1	通讯状态标志	
02	1	IOA0	
03	1	IOA1	
04	1	IOA2	
05	1	IOA3	
06	1	IOA4	
07	1	IOA5	
08	1	IOA6	
09	1	IOA7	
10	1	IOB1	
11	1	IOB2	
12	1	IOB3	

图 3-5 设备调试窗口

二、5 个属性和 8 个方法的应用设计

MCGS 提供的接口规范中有 5 个属性和 8 个方法，下面分别介绍如下：

1. 五个属性

- Public Property Get DevType() As Long, 此函数目的是标明设备的类型：0 为独立的设备，1 为通信子设备。

MCGS 把实际外部设备分为独立的设备、父设备和子设备三类，独立的设备是可独立工作、完成特定数据输入输出功能的设备，如用于模拟量输入输出、数字量输入输出的 PC 接口卡等；父设备是本身不直接处理数据输入输出，但起到管理其它设备工作的设备，如串行口设备、IDCN-893 的通讯卡等；子设备是本身具有处理数据输入输出的功能，但只有和对应的父设备一起才构成完整工作系统的设备：如 PLC 设备、带智能通讯功能的仪器仪表等；不同类型的设备的工作方式有所不同，其对应的设备驱动程序的处理方法也有所不同。

此单片机设备属于子设备，故：

```
Public Property Get DevType() As Long
  DevType = 1
End Property
```

- Public Property Get DevStyle() As Long, 此函数目的是标明设备的种类：0 为独立设备，1 为和串口父设备对应的子设备，本设备属于子设备，故：

```
Public Property Get DevStyle() As Long
  DevStyle = 1
End Property
```

- Public Property Get DevChannel() As Long, 此函数目的是标明设备的通道个数，因下位机的不同和工程的需要，所需的通道数不同。

```
Public Property Get DevChannel() As Long
  DevChannel = chlnumber      扩充属性用于设置通道个数
End Property
```

- Public Property Get DevIOPumber() As Long, 此函数目的是标明设备所占用系统 I/O 的个数，该单片机为子设备，不需要占用系统 I/O, DevIOPumber = 0, 如下所示：

```
Public Property Get DevIOPumber() As Long
  DevIOPumber = 0
End Property
```

- Public Property Get DevBaseI0() As Long, 此函数目的是标明设备所用的 I/O 基地址，由于单片机不占用 I/O 地址，DevBaseI0 = 0。

在 MCGS 设备接口中，属性 DevType、DevStyle、DevIOPumber 为只读属性，用户在组态时不能修改。属性 DevBaseI0 可设置为读写属性，也可设置为只读属性，对不需要 I/O 端口地址的设备，如通讯子设备，设为只读，在本系统中即可设为只

读属性。

定义一个读写属性方法有两种，一是在类模块 DevObj.cls 的定义部分，用 Public 定义属性，如 Public DevBaseIO as Long；一是在 VB 的“工具”菜单中，选取“添加过程…”菜单项，在弹出的对话框中输入属性名，并把选取“属性”选项和“公有的”选项，完成后则在类模块 DevObj.cls 中生成如下代码：

```
Public Property Get DevBaseIO() As Long
End Property
Public Property Let DevBaseIO(ByVal vNewValue As Long)
End Property
```

在上述代码中，如只保留前一部分，则表示该属性为只读。即：

```
Public Property Get DevBaseIO() As Long
  DevBaseIO = 0
End Property
```

用 Property Get 和 Property Let 的方法实现读写属性的目的，是为了在属性值的改变过程中，能对其进行必要的处理；如不需对属性进行处理，则可直接用 Public 来定义属性。

2. 8 个方法的程序设计

- Public Function GetDevName() As String, 此函数是在 MCGS 运行环境中调用，供 MCGS 读取设备的类型名称，在设备工具箱中显示，函数返回设备的类型名称。

```
Const gc_DeviceName = "SPCE061A 单片机"
Public Function GetDevName() As String
  GetDevName = gc_DeviceName
End Function
```

- Public Function GetDevHelp() As Long, 此函数是在 MCGS 组态环境中调用，在设备属性设置窗口的基本属性页中，按“在线帮助”行，调用本接口显示设备构件在线帮助。

```
Public Function GetDevHelp() As Long
  MsgBox "显示设备构件的在线帮助!"
End Function
```

- Public Function SetDevPage(objMcgsSave As Object) As Long, 此函数是在 MCGS 组态环境中调用，用来设置设备构件的内部属性。在设备属性设置窗口的基本属性页中，按“内部属性”行，调用本接口。

```
Public Function SetDevPage(objMcgsSave As Object) As Long
  MsgBox "MCGS 组态环境中，显示设备构件的内部属性!"
End Function
```

● Public Function InitNewDev(objMcgsSave As Object) As Long, 此函数是在 MCGS 组态环境中调用, 当把设备从设备工具箱中选取到设备窗口中时, 调用本接口, 设置设备的缺省属性值。

● Public Function GetCh1Type(alngCh1DataType As Variant, astrCh1Explain As Variant) As Long, 此函数是在 MCGS 运行环境中调用, MCGS 调用本接口读取设备各通道的数据类型和通道内容的文字描述, 在设备属性设置窗口的通道连接属性页中所显示的内容来自本接口。返回值为 0 表示设备通道有效, 返回值为 1 表示设备通道无效。alngCh1DataType 是长整型数组, 标识设备各通道的数据类型和输入输出类型: 值为 1 表示开关型, 值为 2 表示数值型, 值为 3 表示字符型, 大于 0 表示把设备通道的数据采集到 MCGS 中, 小于 0 表示把 MCGS 中的数据输出到设备构件中。在程序中设置了 19 个通道. 其中第一个通道是通信状态标志, 为 0 表示通信正常, 第 2 到 9 通道是采集 IOA0- IOA7 口的状态值, 第 10 到第 17 是采集 IOB0- IOB7 口的状态值, 第 18 个通道是采集到的下位机点滴速度, 第 19 通道是上位机向下位机发送的点滴速度。程序如下所示:

```
Public Function GetCh1Type(alngCh1DataType As Variant, astrCh1Explain
As Variant) As Long
    Dim i As Integer
    For i = 1 To 8
        alngCh1DataType(i) = 1
        astrCh1Explain(i) = "IOA" + Format(i - 1)
    Next i
    For i = 9 To 16
        alngCh1DataType(i) = 1
        astrCh1Explain(i) = "IOB" + Format(i - 8)
    Next i
    alngCh1DataType(17) = 2
    astrCh1Explain(17) = "点滴速度"
    alngCh1DataType(18) = -2
    astrCh1Explain(18) = "开关控制速度"
    alngCh1DataType(0) = 1
    astrCh1Explain(0) = "通讯状态标志"
End Function
```

● 在 MCGS 运行环境启动时调用的函数是 Public Function InitDevRun(objMcgsSave As Object, objMcgsData As Object) As Long, 从而对设备构件进行初始化工作, 具体程序如下所示:

```

Public Function InitDevRun(objMcgsSave As Object, objMcgsData As Object)
As Long
On Error Resume Next
Set m_objCommParent = objMcgsSave.ParentDev
If Err Then
Set m_objCommParent = objMcgsData.ParentDev
End If
m_lngSCMChannelTotal = DevChannel    通过 MCGS 存盘数据对象获取设备的
各种信息
End Function

```

- 在 MCGS 中, 频繁读写的参数都是在通道连接中实现的, 不常用的参数的访问才通过设备命令完成, 这样可以充分提高串口的有效利用率。因此设计中没有采用设备命令, 如果有需要也可以在函数 Public Function RunDevCommand(objMcgsData As Object, ByVal strCommandString As String) As Long 中添加设备命令。
- 在运行环境中调用一个重要的函数是: Public Function CollectDevDat(alngDataFlag As Variant, asngDataValue As Variant, astrDataString As Variant) As Long, 调用此函数可以根据指定的采集周期, 对单片机进行数据采集工作。

```

Public Function CollectDevDat(alngDataFlag As Variant, asngDataValue
As Variant, astrDataString As Variant) As Long
Dim i As Integer
Dim strReturnData As String
Dim ar_compliment(0 To 7) As Integer      ' 余数
Dim quotient(0 To 7) As Integer          ' 商
On Error Resume Next
Dim flag As Long
Data = Chr$(asngDataValue(18))
flag = CollectOneComm0(m_objCommParent, DevNumber, Data)
CollectDevDat = 0
If flag <> 0 Then CollectDevDat = -1
flag = CollectOneComm(m_objCommParent, DevNumber, strReturnData)
CollectDevDat = 0
If flag <> 0 Then CollectDevDat = -1
strReturnData1 = Split(strReturnData, "|", 2)
asngDataValue(17) = strReturnData1(0)

```

如果读取单片机 IOA 口的状态值,就把 IOA0-IOA7 口作为输入。假如 IOA0-IOA7 口置为 01000111,用 PC 机接收到的数据是十进制 64,需要将十进制转化为二进制,得到 IOA0-IOA7 口的状态值。

十进制转化为二进制的程序如下所示:

```
asngDataValue(0) = 0
ar_compliment(0) = strReturnData Mod 2
quotient(0) = strReturnData \ 2
For i = 0 To 6
    quotient(i + 1) = quotient(i) \ 2
    ar_compliment(i + 1) = quotient(i) Mod 2
Next i
For i = 0 To 7
    asngDataValue(i + 1) = ar_compliment(i)
Next i
```

对 Public Function CollectOneComm0(objCommParent As Object, ByVal num As Long, dat As String) As Long 函数的编写要向往单片机写数据去考虑:

```
即 Public Function CollectOneComm0(objCommParent As Object, ByVal num As Long, dat As String) As Long
    Dim Cmd As String, i As Long, m As Long, n As String
    m = WriteData (dat, objCommParent)
    If m = -1 Then CollectOneComm0 = -1: Exit Function
End Function
```

向单片机写数据,调用函数: Public Function WriteData (ByVal Cmd As String, objCommParent As Object) As Long,把想要发的数据帧转换成字节数组的形式,对串行端口进行的操作只需调用父设备提供的标准串口读写函数。MCGS 根据驱动程序的实现特点,充分运用 32 位操作系统多任务多线程的优势,把复杂的对串行端口的设置、初始化和读写操作全部封装在 MCGS 提供的串行通讯父设备中,而且由于在 MCGS 中每个设备构件都独立地运行在自己的线程内,当设备构件等待下位机的返回信息时,它会把自己“挂”起来,处于不占用 CPU 资源的睡眠状态,不用担心在等待过程中会影响系统其他部分的运行。所以只要通过串口父设备提供的写串行端口函数 objCommParent.ComOutDat Len(Cmd), OutPutByte(), 把数据帧发送出去,数据帧是要传送的数据。程序如下所示:

```
Public Function WriteData (ByVal Cmd As String, objCommParent As Object) As Long
On Error Resume Next
```

```

Dim i As Long, iLen As Long
For i = 1 To Len(Cmd)
  OutPutByte(i - 1) = Asc(Mid$(Cmd, i, 1))
Next i
Sleep 1000  '延时 1000 毫秒
iLen3 = objCommParent.ComOutDat(Len(Cmd), OutPutByte2())
End Function

```

对 Public Function CollectOneComm(objCommParent As Object, ByVal num As Long, dat As String) As Long 函数的编写要从单片机读数据去考虑：

从单片机读数据，调用函数：Public Function ReadData (ByVal Cmd As String, objCommParent As Object) As Long，把想要发的数据帧转换成字节数组的形式，然后通过串口父设备提供的写串行端口函数 objCommParent.ComOutDat Len(Cmd), OutPutByte(), 把数据帧发送出去，数据帧是要传送的数据，发送出去后等待下位机的返回信息，下位机收到后返回需要返回的信息，另外加上结束字符一\$，通过串口父设备提供的读串口端口函数 iLen = objCommParent.ComAllInDat(InputByte()) 返回的收到的字节数或者结束符来判断返回结果是否正确，如果返回字节数小于等于 0，则说明没有数据或者通信出错。ReadData 函数如下所示：

```

Public Function ReadData(ByVal Cmd As String, objCommParent As Object)
As Long
  On Error Resume Next
  Dim i As Long, iLen As Long
  For i = 1 To Len(Cmd)
    OutPutByte(i - 1) = Asc(Mid$(Cmd, i, 1))
  Next i
  Sleep 1000  '延时 1000 毫秒
  '通过串行口父设备，给模块发指令
  iLen = objCommParent.ComOutDat(Len(Cmd), OutPutByte())
  Sleep 3000  '延时 3000 毫秒
  iLen = objCommParent.ComoutInDat(InputByte(), 0, 3, 3, 36, 4000)
  If iLen <= 0 Then ReadData = -1: Exit Function
  ReadData = 0
End Function

```

不管是向单片机写数据还是从单片机读数据，为了保证通讯的正确性，在数据发送时适当的增加延时是必要的，一般来说，当速度较慢的 CPU 向速度较快的 CPU 发送数据时应增加适当的延时。

3.5 MCGS 设备驱动构件的测试和挂接

测试设备驱动程序可以在 VB 环境下，也可以在组态环境下，其效果是一样的。

1. 在 VB 环境下测试的结果

图 3-6 为在 VB 组态环境下测试组态环境接口和运行环境接口的数据界面，例如把 A 口的 8 位全部置为低电平，同时实时采集下位机点滴速度，由上位机向下位机发送控制速度等。经过实际运行检验，该驱动程序效果良好，数据采集及时可靠。

通道	数据类型	通道类型	对应值
08	1	IOA5	0
09	1	IOA7	0
10	1	IOB1	0
11	1	IOB2	0
12	1	IOB3	0
13	1	IOB4	0
14	1	IOB5	0
15	1	IOB6	0
16	1	IOB7	0
17	1	IOB8	0
18	2	占满速度	96
19	-2	开关控制速度	60

图 3-6 驱动程序测试图

2. 在组态环境下测试和挂接的方法

在组态环境下测试的结果与在 VB 环境下测试的结果是一样的，下面重点介绍一下测试和挂接的方法。

在组态环境下进行测试的方法如下：SPCE061A 驱动程序在 VB 环境下测试成功后，启动组态环境，新建一个组态工程，把 VB 环境下的工程生成动态链接库文件，拷贝到 D:\MCGS\program\Drivers\用户定制设备目录下（MCGS 的默认安装目录为 D 盘），当开发上位机监控程序时，打开组态环境设备窗口，然后在设备管理窗口中将 SPCE061A 单片机驱动构件（即生成的动态链接库）增加在在设备管理中，串口通讯父设备作为设备 0，SPCE061A 驱动构件作为设备 1 添加到设备窗口中，设置串口通讯父设备的属性，然后打开设备 1，进入“通道连接”属性页，把 17、18 通道的对应数据对象分别设置为点滴速度和开关控制速度，点击设备调试按钮进行调试。

3.6 MCGS 设备驱动程序的应用推广

MCGS 设备驱动程序具有良好的可扩展性，可以应用在基于凌阳单片机开发的实验或应用系统中，因此 MCGS 设备驱动程序的设计方法容易推广到教学实验中。例如，可以开发 SPCE061A 系列组态实验，进行形象直观并富于乐趣的实验教学。

推广 MCGS 设备驱动程序很方便，可以在不改变设备驱动的情况下，只改变上位机监控程序，其中包括用户界面、数据对象和控制流程的设计。具体在扩展时注意需要建立设备通道与实时数据库的连接，指明每个设备通道所对应的数据对象，以便通过设备构件，把采集到的外部设备的数据送入实时数据库。

第四章 液体点滴监控系统下位机的硬件设计与实现

下位机通常指各种数据采集和监控设备，它们与其它必要设备或仪表相结合，感知设备各种参数如过程量和状态量，并将其转换成数字信号传递到上位机系统。上位机系统接收到这些信号后，以适当的形式如声音、图像、图形等方式显示给用户，同时数据经过处理后，告知用户设备各种参数的状态（报警、正常或报警恢复），这些处理后的数据保存到数据库中，也可以通过网络系统传输到不同的监控平台，还可以与别的系统结合形成功能更加强大的系统，上位机还可以接受操作人员的指示，将控制信号发送到下位机，以达到控制的目的。

本系统下位机的核心控制器是凌阳单片机，因此本章首先给出监控系统的功能和总体设计，然后介绍凌阳单片机，最后详细论述各部分硬件电路的设计与实现。

4.1 液体点滴监控系统下位机的功能和结构设计

4.1.1 下位机系统的功能设计

点滴监控系统的下位机是通过凌阳 16 位单片机 SPCE061A 及其它外围电路来实现的，具体功能如下：①检测点滴速度并动态显示；②调整点滴速度到目标值，即点滴速度可用键盘设定并显示，设定范围为 20~150 滴/分，控制误差范围为 12 滴/分钟，调整时间≤5min；③可随时改变设定的点滴速度；④液位低于 2~3cm（大概 20~30ml）时报警；⑤预测输液剩余时间。

4.1.2 下位机系统的结构设计

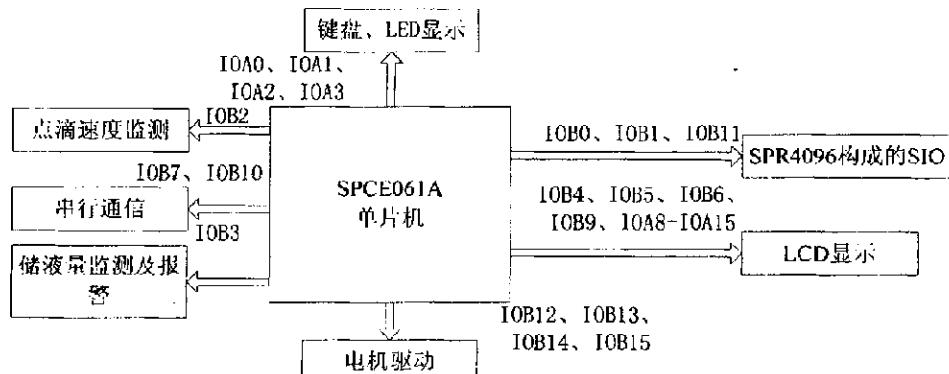


图 4-1 下位机系统的整体结构和 SPCE061A I/O 口的分配方案

图 4-1 下位机系统的整体结构和 SPCE061A I/O 口的分配方案

图 4-1 给出了下位机系统的整体结构以及 SPCE061A 单片机的 I/O 口的分配方案。主控制器采用 SPCE061A 单片机，键盘显示部分采用 HD7279A，数据存储模块以 SPR4096 构成，其它功能模块的设计与实现后面再详细分析。

4.2 SPCE061A 单片机的结构、特点及主要功能模块

4.2.1 SPCE061A 单片机的硬件结构

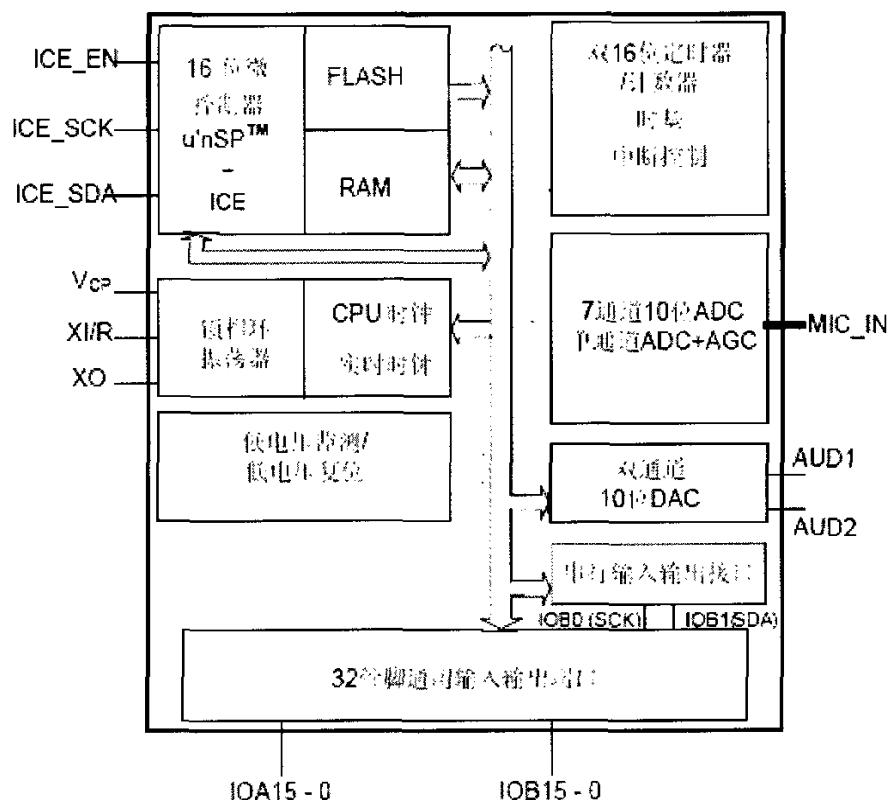


图 4-2 SPCE061A 的结构图

4.2.2 SPCE061A 单片机的特点

SPCE061A 的 CPU 内核采用凌阳 u' nsp16 位微处理器（以下简称 u' nsp™），它具有以下特点：

- 体积小、集成度高、可靠性好且易于扩展
- 具有较强的中断处理能力，其中断系统支持 10 个中断向量及 10 余个中断源，适合实时应用领域。

- 高性能价格比

u' nspTM 片内带有高寻址能力的 ROM、静态 RAM 和多功能的 I/O 口。另外，还提供具有较高运算速度的 16*16 位的乘法运算指令和内积运算指令，为其应用增添了 DSP 功能，使得 u' nspTM 运用在复杂的数字信号处理方面既很便利又比专用的 DSP 芯片廉价。

- 功能强、效率高的指令系统
- 低功耗、低电压

u' nspTM 采用 CMOS 制造工艺，同时增加了软件激发的弱振方式、空闲方式和掉电方式，极大地降低了其功耗。另外 u' nspTM 的工作电压范围大，能在低电压供电时正常工作，且能用电池供电。

4.2.3 SPCE061A 的可编程输入输出口

输入输出接口（I/O 口）是单片机与外设交换信息的通道，SPCE061A 有并行和串行两种方式的 I/O 口。SPCE061A 有两个 16 位通用的并行 I/O 口：A 口（IOA0—IOA15）和 B 口（IOB0—IOB15）。这两个口的每一位都可通过编程单独定义成输入或输出接口。

SPCE061A 提供了位控制结构的 I/O 端口，每一位都可以被单独定义用于输入或输出数据，通常，对每一位的设定包括 3 个基本项：数据向量 Data、属性向量 Attribution 和方向控制向量 Direction。3 个端口内每个对应的位组合在一起，形成一个控制字，用来定义相应 I/O 口位的输入输出状态和方式。

A 口和 B 口的 Data、Attribution 和 Direction 的设定值均在不同的寄存器里，I/O 端口的组合控制设置如表 4-1 所示：

表 4-1 I/O 端口的组合控制设置

Direction	Attribution	Data	功能	是否带唤醒功能	功能描述
0	0	0	下拉	是	带下拉电阻的输入管脚
0	0	1	上拉	是	带上拉电阻的输入管脚
0	1	0	悬浮	是	悬浮式输入管脚
0	1	1	悬浮	否	悬浮式输入管

					脚
1	0	0	高电平输出（带数据反相器）	否	带数据反相器的高电平输出
1	0	1	低电平输出（带数据反相器）	否	带数据反相器的低电平输出
1	1	0	低电平输出	否	带数据缓存器的低电平输出
1	1	1	高电平输出	否	带数据缓存器的高电平输出

4.2.4 SPCE061A 的中断系统

SPCE061A 可提供 14 个中断源，即两个定时器中断、两个外部中断、一个串行口中断、一个触键唤醒中断、7 个时基信号中断、PWM 音频输出中断，如表 4-2。

表 4-2 中断源列表

中断源	中断优先级	中断向量	保留字
Fosc/1024 溢出信号 PWM INT	FIQ/IRQ0	FFF8H/FFF6H	_FIQ/_IRQ0
TimerA 溢出信号	FIQ/IRQ1	FFF9H/FFF6H	_FIQ/_IRQ1
TimerB 溢出信号	FIQ/IRQ2	FFFAH/FFF6H	_FIQ/_IRQ2
外部时钟源输入信号 EXT2	IRQ3	FFFFBH	_IRQ3
外部时钟源输入信号 EXT1			
触键唤醒信号	IRQ4	FFFFCH	_IRQ4
4096Hz 时基信号			
2048Hz 时基信号			
1024Hz 时基信号	IRQ5	FFFFDH	_IRQ5
4Hz 时基信号			
2Hz 时基信号			
频选信号 TMB1	IRQ6	FFFFEH	_IRQ6
频选信号 TMB2			
UART 传输中断	IRQ7	FFFFFH	_IRQ7
BREAK	软中断		

4.3 SPCE061A 单片机外围各模块电路的设计

4.3.1 液体点滴装置外形图

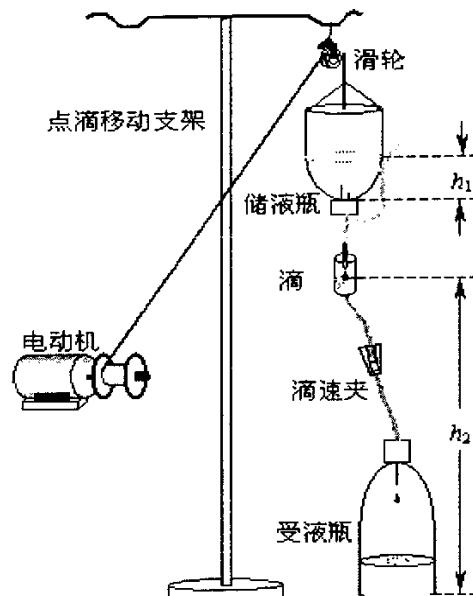


图 4-3 液体点滴装置外形图

图 4-3 是液体点滴装置的外形图, 包括点滴移动支架、步进电机、滑轮、红外发射接收管、储液瓶等。

4.3.2 液体点滴速度检测及信号处理电路的设计

方案 1: 采用电感式传感器测量点滴速度。在输液器的漏斗外围绕线圈作为敏感元件, 当液滴滴下时电感量发生变化, 通过 LC 振荡电路输出变化的频率值, 经过 F/V 转换电路及电压比较后输出 TTL 电平信号来检测点滴速度。此方案测量精度高, 但外围电路复杂。

方案 2: 可见光发光二极管与光敏三极管组成的传感电路。由于系统外界光源对光敏三极管的工作会有很大干扰, 一旦外界光亮度改变, 就会影响对液滴的判断。如果采用超强亮度发光管可以减小干扰, 但功率损耗大。

方案 3: 采用红外光电传感器测量点滴速度。当液滴滴下时, 红外光电传感器发射的光透过液滴后强度发生变化, 光电接收管接收强度变化的光信号后输出变化的电压信号, 此电压信号经放大、整形后被转化为 TTL 电平信号, 送给单片机计数来测

量点滴速度。由于红外光波长比可见光长,受可见光的影响较小,且该传感器具有体积小、灵敏度高、线性好等特点,其外围电路简单,性能稳定可靠。

因此,对点滴速度的检测采用方案3。

采用红外传感器测量点滴速度的电路原理图如图4-4所示,光电检测器用的是普通的红外发射接收对管,需要做的工作是把红外发射和接收管正对着固定在滴斗两侧。红外发射管在恒定电源驱动下发射一定频率的恒定红外线,红外线经过对管间的间隙时,由接收管接收。当有液滴通过对管间隙时,红外发射管发出的红外线透过液滴时接收端光功率发生变化,光电接收管将变化的光信号转化为变化的电信号,由于电信号非常微弱,应放大到一定幅度且通过积分电路消除干扰,再经过比较器整形得到与点滴同频方波,把变化的电压信号输出到单片机上,通过单片机来计算其变化的次数。

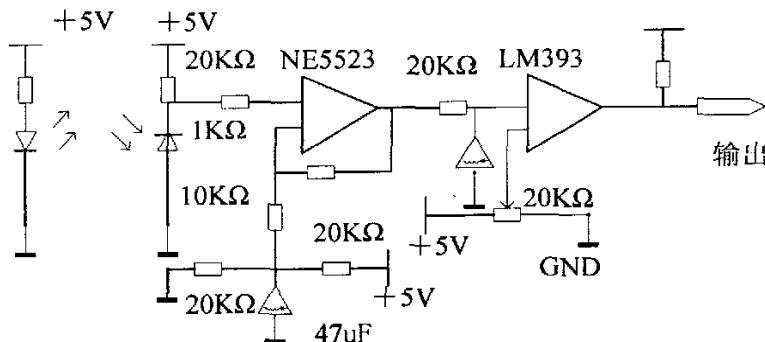


图4-4 点滴速度检测、放大电路图

4.3.3 储液检测电路的设计

在输液快要结束时,必须将药液截断,防止空气进入患者体内造成危险,为此设计了储液检测电路。

方案一:采用红外对管,根据接收管接收到的光强大小来判断液位是否到达警戒线。

方案二:利用超声波在不同密度物质内传播速度的不同;通过检测超声波发射后回波的时间来检测超声波穿过物质的结构。利用MCU定时控制超声波的发射,以中断方式接收检测到的回波,经MCU处理后获得所需要的数据。此系统将预先测定的液位到达警戒线时的回波时间作为标准,系统将每次测量的结果与此标准比较,便可得知是否到达警戒液位。

方案一的电路结构简单,软硬件都比较容易实现;方案二理论成熟,但超声波探测存在一个盲区,盲区的大小与MCU的处理速度有关,在对精度要求较高的场合还需加入温度补偿模块及其软件算法,以改善环境温度对超声探测的影响。综合权

衡，采用了方案一作为液位的检测方案。

4.3.4 键盘和数码显示电路的设计

点滴监控系统的点滴速度需通过键盘来设置，同时还需要将设置的点滴速度值和当前点滴速度值显示出来。为了节省单片机 I/O 口，键盘和数码显示模块电路采用了芯片 HD7279A。

一、HD7279A 的原理与应用

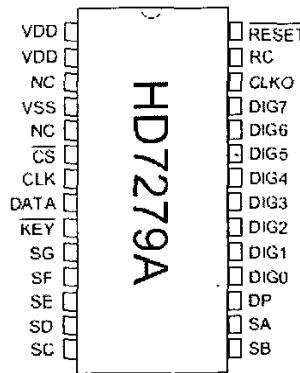


图 4-5 HD7279A 引脚图

如图 4-5 所示的 HD7279A 是一种管理键盘和 LED 显示器的专用智能控制芯片。它能对 8 位共阴极 LED 显示器或 64 个 LED 发光管进行管理和驱动，同时能对 8*8 键盘矩阵的按键情况进行监视，具有自动消除键抖动并识别按键代码的功能，从而提高 CPU 的工作效率，其内部含有译码器，可直接接收 BCD 码或 16 进制码，具有两种译码方式，此外还具有多种控制指令。

1、HD7279A 引脚说明

表 4-3 HD7279 引脚说明

引脚	名称	说明
1, 2	VDD	正电源
3, 5	NC	无连接，必须悬空
4	VSS	接地
6	CS	片选输入端，此引脚为低电平时，可向芯片发送指令及读取键盘数据。
7	CLK	同步时钟输入端，向芯片发送数据及读取键盘数据时，此引脚电平上升沿表示数据有效。
		串行数据输入/输出端，当芯片接收指令时，此引脚

8	DATA	为输入端;当读取键盘数据时,此引脚在‘读’指令最后一个时钟的下降沿变为输出端。
9	KEY	按键有效输出端,平时为高电平,当监测到有效按键时,此引脚变为低电平。
10-16	SG-SA	段 g-段 a 驱动输出
17	DP	小数点驱动输出
18-25	DIG0-DIG7	数字 0-数字 7 驱动输出
26	CLK0	振荡输出端
27	RC	RC 振荡器连接端
28	RESET	复位端

2、串行接口

HD7279A 采用串行方式与微处理器通讯,串行数据从 DATA 引脚送入芯片,并由 CLK 端同步。当片选信号变为低电平后,DATA 引脚上的数据在 CLK 引脚的上升沿被写入 HD7279A 的缓冲寄存器。

HD7279A 的指令有三种类型:①不带数据的纯指令,指令的宽度为 8 个 BIT,即微处理器需发送 8 个 CLK 脉冲。②带有数据的指令,宽度为 16 个 BIT,即微处理器需发送 16 个 CLK 脉冲。③读取键盘数据指令,宽度为 16 个 BIT,前 8 个 BIT 是微处理器发送到 HD7279A 的指令,后 8 个 BIT 为 HD7279A 返回的键盘代码。执行此指令时,HD7279A 的 DATA 端在第九个 CLK 脉冲的上升沿变为输出状态,并于第 16 个脉冲的下降沿恢复为输入状态,等待接收下一个指令。

二、HD7279A 应用电路原理图

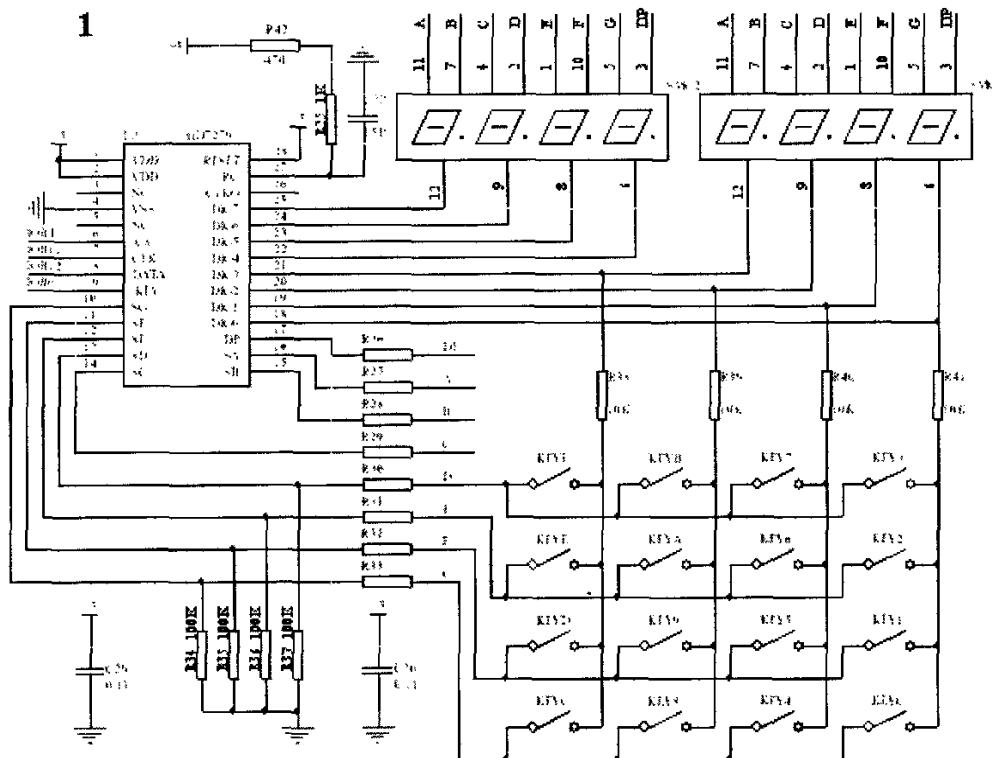


图 4-6 HD7279A 应用电路原理图

如图 4-6 所示, HD7279A 应连接共阴式数码管, 4 只下拉电阻和 4 只键盘连接位选线 DIG0-DIG3 的 4 只电阻(简称位选电阻), 应遵从一定的比例关系, 下拉电阻应大于位选电阻的 5 倍而小于其 50 倍, 典型值为 10 倍, 下拉电阻的取值范围是 10K-100K, 位选电阻的取值范围是 1K-10K。在不影响显示的前提下, 下拉电阻应尽可能地取较小的值, 这样可以提高键盘部分的抗干扰能力。

4.3.5 LCD 显示电路的设计

在液体点滴监控系统中, 当上位机发送点滴速度值或其它命令到下位机时, 通过液晶 LCD 来显示速度值及相关命令, 其特点是可以显示汉字。此处采用 SPG12063YS2 液晶, 它是自带 SPLC501 的 128×64 点阵式 LCD, SPLC501 是凌阳 LCD 驱动电路。SPLC501 液晶显示控制驱动器集行、列驱动器和控制器于一体, 广泛应用于小规模液晶显示模块, 其功能原理图如图 4-7 所示。

在使用 LCD 显示板的时候, 用户只要通过其接口, 微处理器与 LCD 及其驱动的 RAM、命令寄存器、状态寄存器交换状态、数据以及命令, 就可以控制 LCD 的显示内容、显示特色(闪烁、背光等)和颜色。

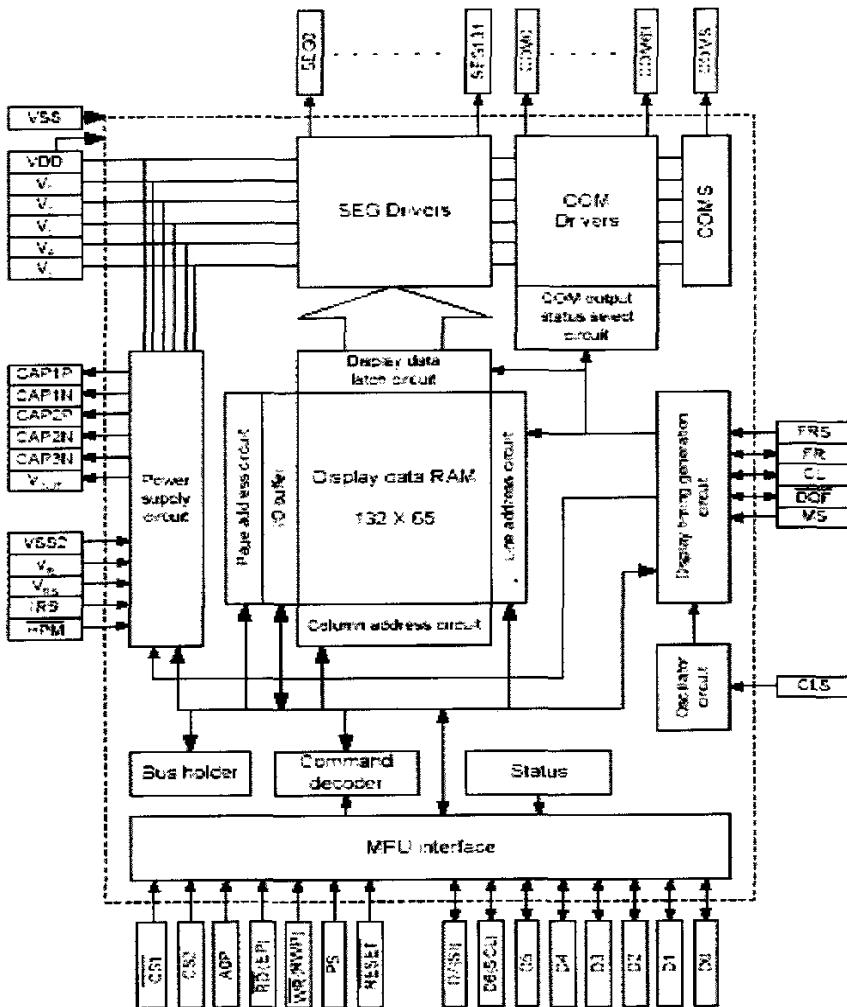


图 4-7 SPLC501 原理框图

4.3.6 液体点滴速度控制电路的设计

方案 1: 对滴速夹松紧的控制

对滴速夹松紧的控制就是通过对滴速夹松紧的调节,改变塑料点滴管的形状以控制液体的流速。此方法虽然直观,但存在很多的缺点。首先对管壁施加的压力与流量改变的关系呈非线性,这给流量控制带来了难度;其次滴管塑料的,存在变形问题,长时间受压后松开滴速夹并不能使塑料滴液管完全恢复原形,控制装置无法保证理想的控制效果。

方案 2: 通过改变滴斗到受液瓶的高度 h_2 来调节点滴的速度

由步进电机带动储液瓶使储液瓶上升或下降, 以改变滴斗到受液瓶的高度 h_2 。当检测到的滴速值低于键盘设定值时, 电机反向转动以增加 h_2 的高度, 从而增加点滴速率。

滴速度；反之，电机正向转动降低 h_2 的高度，从而降低点滴速度。此方案思路简单，易于对点滴速度进行控制。因此，选择了此方案。

1、电机的选择

方案 1:采用直流电机

方案 2:采用步进电机

方案 3:采用伺服电机

由于直流电机上电即转动，掉电后惯性较大，停机时转动一定角度后才可停下来，如果要求准确停在一个位置，其闭环算法较复杂。伺服电机价格较高。所以在达到基本使用要求的前提下采用了步进电机。

步进电动机有如下优点：①步进电动机的角位移与输入脉冲严格成正比，因此，当它转一圈后，没有累计误差，具有良好的跟随性。②由步进电机与驱动电路组成的开环数控系统，不仅简单、可靠、廉价，而且它也可以与角度反馈环节组成高性能的闭环数控系统。③步进电动机的动态响应快，易于起停、正反转及变速。④速度可在相当宽的范围内平滑调节，低速下仍能保证获得大转矩，因此，一般可以不用减速器而直接驱动负载。

不过步进电机存在振荡和失步等缺点，其失步原因有两种：第一种是转子的转速慢于旋转磁场的速度，或者说慢于换相速度。例如，步进电动机在启动时，若脉冲的频率较高，则电动机来不及获得足够的能量，无法令转子跟上旋转磁场的速度，从而引起失步。步进电机有一个启动频率，超过启动频率启动时，肯定会产生失步。第二种是转子的平均速度大于旋转磁场的速度。这主要是发生在制动和突然换向时，转子获得过多的能量，产生严重的过冲，引起失步。为了减少失步的现象，最好不使用整步状态，整步状态时振动大，所以设计时选择步距角为 0.9 度。

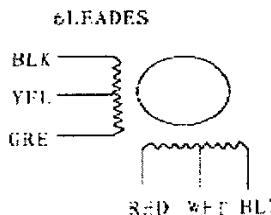


图 4-8 电机线路图

系统选用的电机型号是 42BYGH404，电机线路图如图 4-8 所示。

图中 BLK 对应 A 相、YEL 对应 O 相、GRE 对应 C 相、RED 对应 B 相、BLU 对应 D 相、WHT 对应 O' 相，其所加脉冲是四相八拍的，其通电顺序为：红—绿—蓝—黑。

2、电机驱动模块

电机驱动电路采用 L293D 芯片以放大信号。L293D 芯片输出具有高电压、大电

流、四通道驱动、使用简单和温度过高保护等特点,设计用来接收 DTL 或者 TTL 逻辑电平,驱动感性负载(比如继电器、直流和步进电动机)。

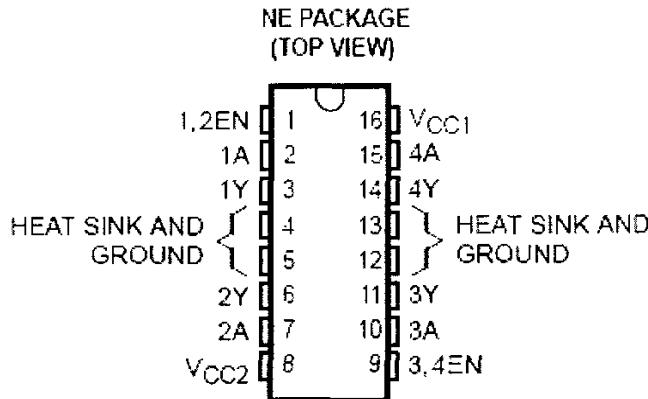


图 4-9 L293D 引脚图

L293D 的引脚图如图 4-9 所示,其中引脚 2、7、10、15 作为 L293D 的输入端,分别接 IOB12、IOB13、IOB14、IOB15,引脚 3、6、11、14 作为 L293D 的输出去控制步进电机,分别接步进电机的 A、B、C、D 四相,连接图如图 4-10 所示。

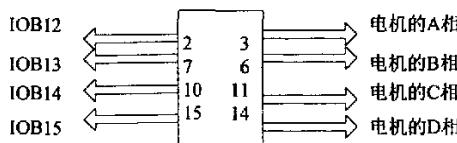


图 4-10 L293D 与单片机、电机接线图

4.3.7 异步串口通讯电路的设计

一、异步串口通讯电路

异步串行传输采用 RS232 标准,其最大传输速率为 20kb/s。通讯接口采用标准的 232 接口电平,串口 9 芯插头与 PC 机 RS232 串口相连。PC 机的 RS232 串行口标准逻辑电平“0”为-3V 至-12V,“1”为+3V 至+12V。芯片 HIN232CP 既能把±12V 电平转换为 0~5V,同时也能把单片机的 0~5V 信号电平转换成 PC 机 RS232 所要求的±12V 信号电平。PC 机 RS232 口和单片机串行通讯只用了 RXD、TXD、GND 三根线,其原理图如图 4-11 所示。图中滤波电容 C5,给 HIN232CP 提供稳定的供电电压;另外 4 个电容是升压电容,芯片内部经过倍压把+5V 转换成±10V。HIN232CP 第 2 引脚上的电压为+10V,第 6 引脚上的电压为-10V。

UART

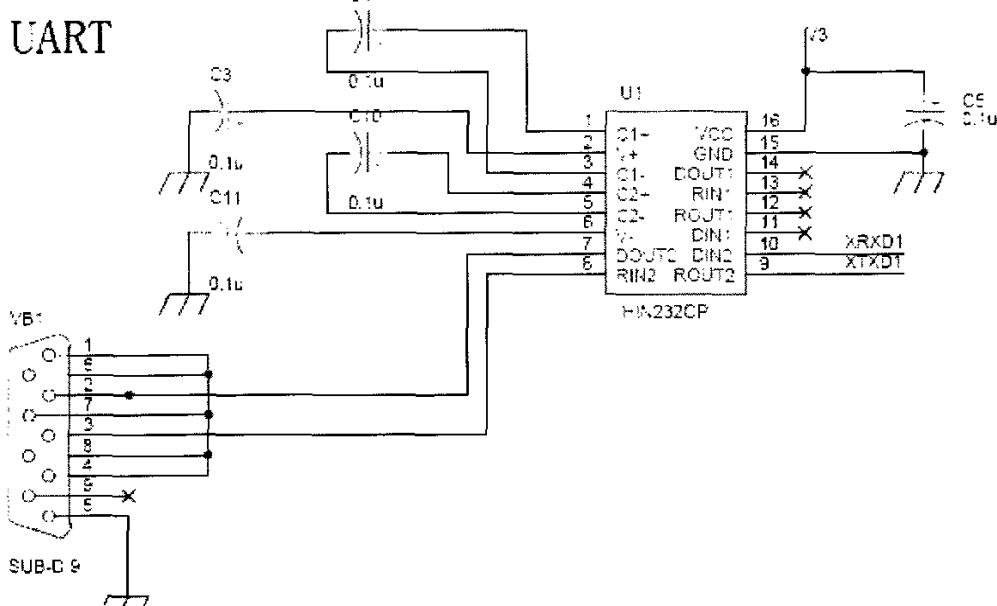


图 4-11 串口通信电路原理图

二、通用异步串行接口 UART 的使用

使用 UART 模块进行通讯时,首先将 IOB7 和 IOB10 分别设置为输入和输出状态,然后通过 P_UART_BaudScalarLow 和 P_UART_BaudScalarHigh 单元指定所需波特率(如 9600bps),同时设置 P_UART_Command1 和 P_UART_Command2 单元以激活 UART 通讯功能。设置 P_UART_Command1 的 6、7 位可以激活 UART IRQ 中断,并决定中断是由 TxRDY 或 RxRDY 信号触发还是由二者共同触发。设置 P_UART_Command2 单元的 6、7 位可以激活 UART 的 Tx、Rx 引脚功能。当 u' nspTM 接收或发送 1 个字节数据时, P_UART_Command2 单元的第 6、7 位被置为 1 且同时触发 UART IRQ。无论 UART 中断是否被激活, UART 接收/发送功能都可以由 P_UART_Command2 单元的第 6、7 位控制。任意时刻读 P_UART_Command2 单元都会清除 UART IRQ 中断标志。

表 4-4 UART 数据帧的格式

start	D0	D1	D2	D3	D4	D5	D6	D7	parity	stop
-------	----	----	----	----	----	----	----	----	--------	------

使用 UART 串行接口时需要设置以下寄存器:

1、P_UART_Command1 (写) (7021H)

设置该寄存器的第 2、3 位可以控制数据奇偶校验功能,第 6、7 位控制着 UART IRQ 中断。

表 4-5 P_UART_Command1 单元

b7	b6	b5	b4	b3	b2	b1	b0	功能
RxIntEn	TxIntEn	I_Rest	-	Parity	P_Check	-	-	
1	-	-	-	-	-	-	-	允许 UART 中断(由 RxRDY 信号触发)
0	-	-	-	-	-	-	-	禁止 UART 中断
-	1	-	-	-	-	-	-	允许 UART 中断(由 TxRDY 信号触发)
-	0	-	-	-	-	-	-	禁止 UART 中断
-	-	1	-	-	-	-	-	内部复位信号复位
-	-	0	-	-	-	-	-	内部复位信号置位
-	-	-	-	1	-	-	-	激活偶校验功能
-	-	-	-	0	-	-	-	激活奇校验功能
-	-	-	-	-	1	-	-	激活奇偶校验功能
-	-	-	-	-	0	-	-	屏蔽奇偶校验功能

2、P_UART_Command2 (写) (7022H)

该单元写入时为 UART 数据发送/接收控制寄存器, 第 6、7 位分别控制着数据发送和接收管脚的允许/禁止, 此时 IOB7 和 IOB10 必须分别被置为 Rx 和 Tx 管脚, 当发送管脚被允通时, IOB10 输出管脚将自动被置为高电平。

表 4-6 P_UART_Command2 单元

b7	b6	b5	b4	b3	b2	b1	b0
RxPinEn	TxPinEn	-	-	-	-	-	-
1: 允许接收管脚	1: 允许发送管脚						
0: 禁止接收管脚	0: 禁止发送管脚						

3、P_UART_Command2 (读) (7022H)

P_UART_Command2 单元读出时为 UART 状态信息 (如表 4—7 所示)。第 7 位是 RxRDY 标志位, 当接收到数据时该标志位被置为“1”, 读 P_UART_Data 单元将清除该标志位; 第 6 位是 TxRDY 标志位, 当通过写入本单元第 6 位为‘1’来允通发送管脚后, 该标志位被置为“1”, 表示发送器的数据缓存器为空, 已准备好可以发送

写入 P_UART_Data 单元数据。

表 4-7 P_UART_Command2 单元

b7	b6	b5	b4	b3	b2	b1	b0
RxRDY	TxRDY	FE	OE	PE	-	-	-
1: 数据已接收完毕 0: 未接收到数据	1: 数据发送已准备 0: 数据发送未准备	1: 存在帧错误 0: 无帧错误	1: 存在溢出错误 0: 无溢出错误	1: 存在奇偶校验错误 0: 无奇偶校验错误			

4、P_UART_Data (读/写) (7023H)

表 4-8 P_UART_Data 单元

b7	b6	b5	b4	b3	b2	b1	b0
数据							

5、P_UART_BaudScalarLow (读/写) (7024H) 和 P_UART_BaudScalarHigh (读/写) (7025H)

两个单元的组合控制数据的传输速率 (波特率), UART 波特率的计算公式如下:
当 $F_{osc}=49.152MHz$ 、 $40.960MHz$ 或 $32.768MHz$ 时, 波特率 = $(F_{osc}/4) / Scale$;
当 $F_{osc}=24.576MHz$ 或 $20.480MHz$ 时, 波特率 = $(F_{osc}/2) / Scale$ 。

4.3.8 SPR4096 存储电路的设计

如图 4-12 所示的 SPR4096 存储电路用来存储上位机传来的数据及命令。

SPR4096 & SIO

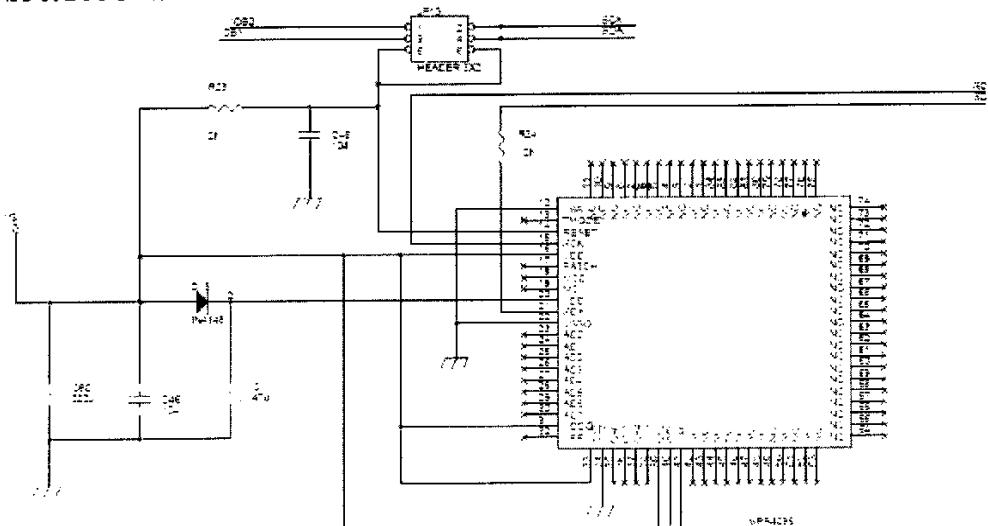


图 4-12 SPR4096 构成的 SIO 存储电路

一、SPR4096 的结构

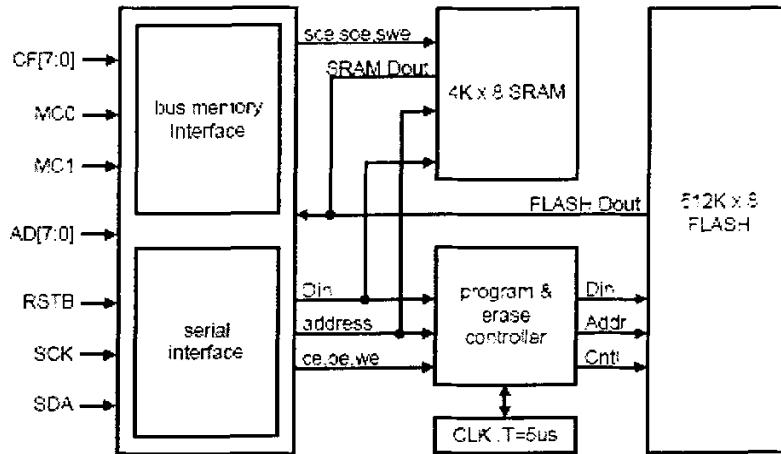


图 4-13 SPR4096A 体系结构

SPR4096 的内部结构如图 4-13 所示, 它有 $512K \times 8$ bitsFLASH 和 $4K \times 8$ bitsSRAM, 有两种接口方式: BMI 接口模式 (Bus Memory Interface) 和 SIF 接口模式 (Serial Interface), 还包括定时器和写入擦除控制器 (PECON)。CF0~CF7 用来定义存储配置模式。当选择 BMI 接口模式时, MC0 和 MC1 作为读写控制信号, AD[7:0] 是双向地址/数据总线。BMI 处理这些信号并产生控制信号和 FLASH 或 SRAM 读写的地址/数据总线信号。当 BMI 接收到写入、扇区擦除或全部擦除 FLASH 命令时, BMI 把这些命令送给 PECON 来执行。当选择 SIF 接口模式时, SCK 作为串行时钟信号, SDA 作为 1-bit 串行数据线。如果 SIF 接收到读写 FLASH 或 SRAM 命令时, SIF 则可以直接从相应的存储器中读写数据。如果 SIF 接收到写入、扇区擦除或者全擦除命令时, SIF 也把这些命令送给 PECON 来执行。需要 200KHz 时钟启动 PECON (此时钟由体系结构图中的定时器 CLK 来提供)。本系统采用 SIF 方式, 即串行同步传送数据, 系统默认 IOB11 为 SPR4096 的片选信号。

二、串行输入输出端口 S10 的应用

串行输入输出端口 S10 提供了一个一位的串行接口, 用于与其它设备进行数据通讯。在 SPCE061A 内通过 IOB0 和 IOB1 这两个端口实现与设备进行串行数据交换功能, IOB0 作为时钟端口, IOB1 作为数据端口。

使用 S10 时需要设置以下寄存器:

1、S10 控制寄存器 P_S10_Ctrl (701EH)

用户必须通过设置 P_S10_Ctrl 单元的第 7 位, 将 IOB0、IOB1 分别设置为 SCK 管脚和 SDA 管脚。

表 4-9 P_SIO_Ctrl 单元

b7	b6	b5	b4	b3	b2	b1	b0	设置功能说明
SIO_Config	R/W	R/W_EN	Clock_Set	-	Addr_Select			
x	x	x	x	x	-	0	0	串行设备地址 (缺省)设置为16位
x	x	x	x	x	-	0	1	无地址设置
x	x	x	x	x	-	1	0	串行设备地址设置为8位
x	x	x	x	x	-	1	1	串行设备地址设置为24位
x	x	x	0	0	-	x	x	数据传输速率为 CPUClk/16(缺省)
x	x	x	0	1	-	x	x	无用
x	x	x	1	0	-	x	x	数据传输速率为 CPUClk/8
x	x	x	1	1	-	x	x	数据传输速率为 CPUClk/32
1	x	x	x	x	-	x	x	设置IOB0=SCK, IOB1=SDA。用户 不必设置IOB0和 IOB1的输入输出 状态

2、SIO 数据寄存器 P_SIO_Data(701AH)

P_SIO_Data 单元格式如表 4-10 所示, 该单元为接收/发送串行数据的缓冲单元, 向它写入或读出数据, 可按串行方式发送或接收数据字节。通过写入 P_SIO_Start 单元来启动 P_SIO_Data 单元与串行设备数据交换的过程。传输从串行设备的起始地址(由 P_SIO_Addr_Low、P_SIO_Addr_Mid 和 P_SIO_Addr_High3 个单元指定)开始, 然后是数据。

表 4-10 P_SIO_Data

b7	b6	b5	b4	b3	b2	b1	b0
D7	D6	D5	D4	D3	D2	D1	D0

表 4-11 P_SIO_Addr_Low

b7	b6	b5	b4	b3	b2	b1	b0
A7	A6	A5	A4	A3	A2	A1	A0

表 4-12 P_SIO_Addr_Mid

b7	b6	b5	b4	b3	b2	b1	b0
A15	A14	A13	A12	A11	A10	A9	A8

表 4-13 P_SIO_Addr_High

b7	b6	b5	b4	b3	b2	b1	b0
A23	A22	A21	A420	A19	A18	A17	A16

3、SIO 启动控制寄存器 P_SIO_Start (70FAH)

P_SIO_Start 单元格式如表 4-14 所示, 向它写入任意一个数值, 就可以启动数据传输。当对 P_SIO_Start 单元读写操作时会使 SIO 根据 P_SIO_Addr_Low、P_SIO_Addr_Mid 和 P_SIO_Addr_High 的内容传输读/写操作的起始地址, 之后再读写 P_SIO_Data 单元时将不再传输此起始地址。如果需要重新指定一个起始地址进行数据传输, 就向 P_SIO_STOP 单元写入任意一个数值以停止 SIO 操作, 然后向 P_SIO_Addr_Low、P_SIO_Addr_Mid 和 P_SIO_Addr_High 写入新的地址, 最后, 向 P_SIO_Start 单元写入任意一个数值重新启动 SIO 操作。

读 P_SIO_Start 单元可获取 SIO 的数据传输状态, 其第 7 位 Busy 为占用标志位, Busy=‘1’表示正在传输数据, 传输操作完成后, 该位将被清‘0’, 可以开始传输新的数据字节。

表 4-14 P_SIO_Start 单元

b7	b6	b5	b4	b3	b2	b1	b0
Busy	-	-	-	-	-	-	-

4、SIO 停止控制寄存器 P_SIO_Stop (7020H)

向 P_SIO_Stop 单元写入任一数值, 可停止数据传输。

三、FLASH 扇区擦除

如表 4-15 所示, FLASH 扇区擦除命令为: 一个起始位、一个 4-bit 操作码 (A[24:21]=0100)、一个 2 位存储器选择和一个 19 位地址 (A[18:0])。其中 A18~A11 的值表明擦除哪个扇区, A10 到 A0 为任意值。当接收到 FLASH 扇区擦除命令后, 必须有一定的时延 (t_{ERASE}) 来擦除指定 FLASH 扇区中的数据; 然后由停止位结束命令。

表 4-15 FLASH 扇区擦除命令

S	0100	CF4	CF3	A18~A0	P	- wait 24ms
---	------	-----	-----	--------	---	-------------

4.3.9 语音报警电路

当点滴液位降到警戒值时驱使语音报警。系统利用 SPR4096 存储录制的语音数据，播放时从 SPR4096 中读出并送语音输出通道即可。

凌阳 SPCE061A 单片机自带双通道 DAC 音频输出，DAC1、DAC2 转换输出的模拟量电流信号分别通过 AUD1 和 AUD2 管脚输出并转换为电压，然后经过 SPY0030 音频放大，以驱动喇叭放音，音频放大电路如图 4-14 所示。这为单片机的音频设计提供了极大方便。

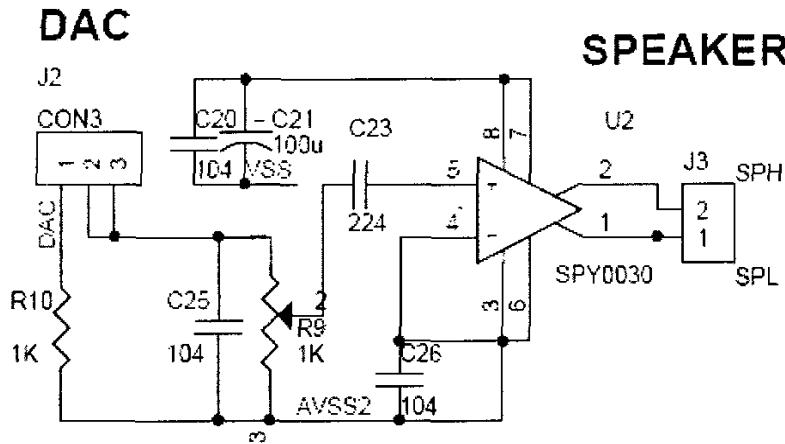


图 4-14 音频放大电路

第五章 液体点滴监控系统下位机的软件设计与实现

液体点滴监控系统下位机的核心控制器是凌阳单片机，其软件设计平台采用了凌阳 u'spTMIDE 集成开发环境，它支持汇编语言和 C 语言混合编程。

5.1 下位机软件系统的总体设计

液体点滴监控系统下位机的核心是单片机，由它完成监控对象选择、点滴速度设定、液位到达警戒线时报警、数据采样及处理、监控内容显示以及与上位机数据通信等功能，其系统结构框图和程序流程图如图 5-1 和 5-2 所示。

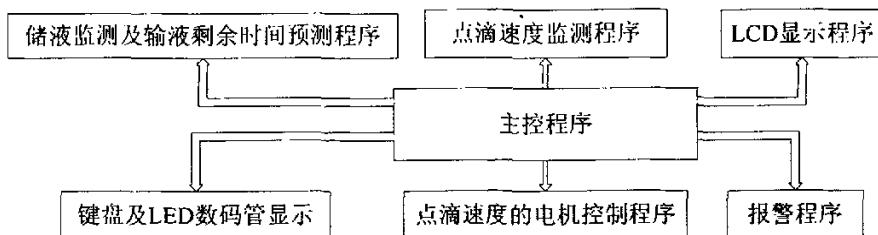


图 5-1 程序总体结构图

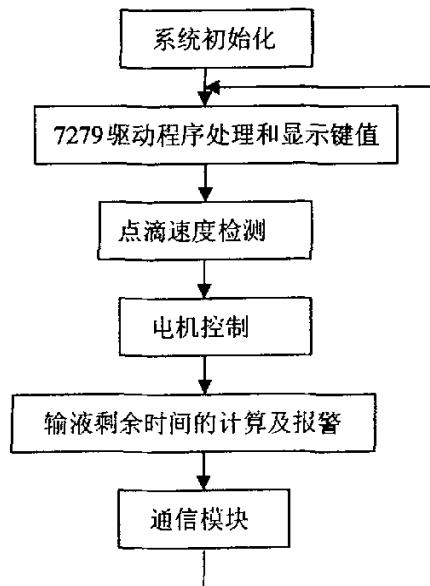


图 5-2 下位机软件设计流程图

5.2 下位机系统各软件模块的详细设计

整个下位机控制系统程序由六个子程序模块和一个主控程序构成。u'spTM IDE 开发环境集程序的编辑、编译、链接、调试以及仿真等功能为一体，具有友好的交互界面、下拉菜单、快捷键和快速访问列表等，使得编程、调试工作更加方便和高效。此外，其软件仿真功能可以在不连接仿真板的情况下模拟硬件的各项功能来调试程序。程序开发过程中有几个常用的库需要加载，包括音频库 sacmv26e.lib、宏定义库 Cmacro.lib、标准 C 函数库 clib100.Lib 等。

5.2.1 点滴速度检测模块

传感器信号接至外部中断引脚，当有液滴滴下时电压信号发生变化，从而产生一个中断，以每 10s 为单位计算中断的次数，从而算出每分钟中断的次数，即每分钟的点滴数，其程序流程图如图 5-3 所示。

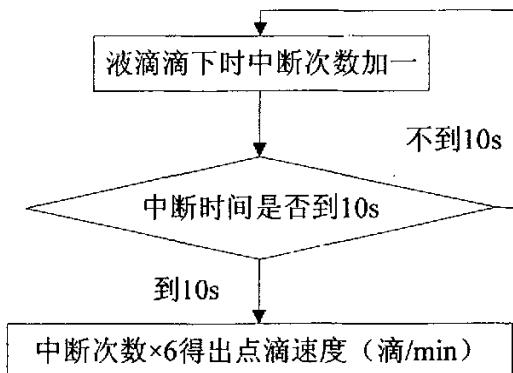


图 5-3 点滴速度检测模块

这里用到的中断是 Irq3_ext1，SPCE061A 有 EXT1 和 EXT2 两个外部中断。IOB2 (EXT1) 和 IOB3 (EXT2) 两条外部中断请求输入线，以负跳沿触发方式输入两个外部中断源的中断请求信号。

10s 的定时利用 IRQ5_4Hz 实现，它属于时基信号中断源，SPCE061A 单片机具有时基中断，减少了软硬件关于实时时钟处理过程，时基信号频率丰富，有 2Hz、4Hz、8Hz、16Hz、32Hz、64Hz、128Hz、256Hz、512Hz、1024Hz、2048Hz、4096Hz 等多种频率，为用户实时处理应用提供了各种时钟选择。

SPCE061A 的时间基准信号来自于 32768Hz 实时时钟。当开启时基信号中断后有时基信号到来，发出时基信号中断请求，CPU 查询到有中断请求后，允许中断并置位 P_INT_Ctrl 相应的中断请求位，在中断服务程序中通过测试 P_INT_Ctrl 来确定

是哪个频率时基信号产生的中断。可以采用不同频率的时基信号来做长时间或短时间的定时控制。

1. SPCE061A 的时钟和时间基准信号

时基信号发生器的选频逻辑 TMB1 为 TimerA 的时钟源 B 提供各种频率选择信号并为中断系统提供中断源 (IRQ6) 信号。此外，时基信号发生器还可以通过分频产生 2Hz、4Hz、1024Hz、2048Hz 以及 4096Hz 的时基信号，为中断系统提供各种实时中断源 (IRQ4 和 IRQ5) 信号。时基信号发生器的结构如图 5-4 所示。

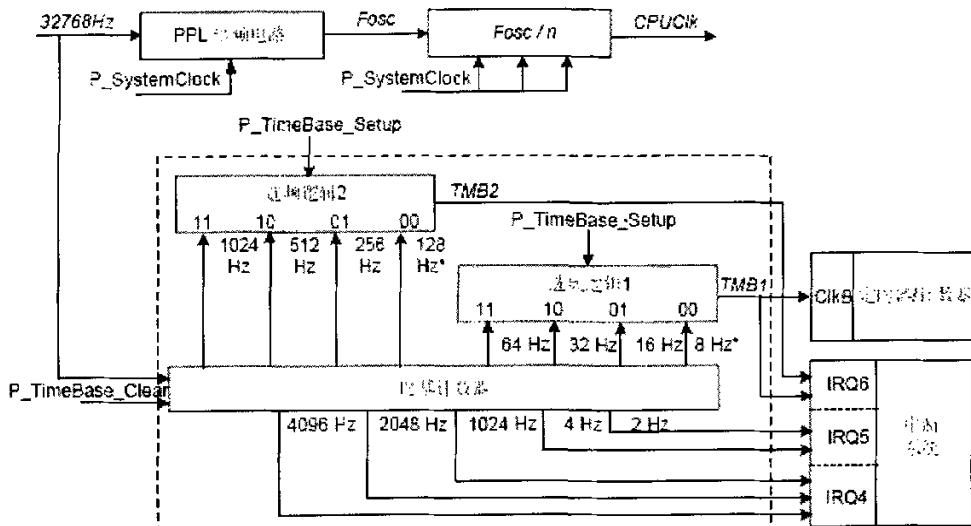


图 5-4 时基信号发生器的结构

时基信号发生器通过对 P_Timebase_Setup (写) (700EH) 单元的编程写入来进行选频操作，如如表 5-1 和 5-2 所示。

表 5-1 P_Timebase_Setup 单元

b15-b4	b3	b2	b1	b0
...	TMB2 选频逻辑		TMB1 选频逻辑	

表 5-2 选频逻辑

b3	b2	TMB2	b1	b0	TMB1
0	0	128Hz	0	0	8Hz
0	1	256Hz	0	1	16Hz
1	0	512Hz	1	0	32Hz
1	1	1024Hz	1	1	64Hz

2. 时基中断使用的控制寄存器

表 5-3 时基中断使用的控制寄存器

配置单元	读写属性	存储地址	配置单元功能说明
P_TimeBase_Setup	写	700EH	时基信号发生器输出的TMB1、TMB2频率设定。
P_TimeBase_Clear	写	700FH	时基信号发生器复位并进行时间的精确校准。
P_SystemClock	写	7013H	系统时钟的选择控制单元
P_INT_Ctrl	读/写	7010H	写可控制各中断源允许或禁止中断，读可判断产生中断请求的中断源。
P_INT_Clear	写	7011H	用来清除中断源的中断请求

SPCE061A 对中断源的开放和屏蔽，以及每个中断源是否被允许中断，都受中断允许寄存器 P_INT_Ctrl、P_INT_Clear 及 P_INT_New 控制。下面重点介绍一下 P_INT_Ctrl 单元和 P_INT_Clear 单元。

① 中断控制单元 P_INT_Ctrl (读/写) (7010H)

P_INT_Ctrl 控制单元具有可读和可写属性，但读写时的意义是不同的。

表 5-4 中断控制单元 P_INT_Ctrl

b7	b6	b5	b4	b3	b2	b1	b0
IRQ3_K EY	IRQ4_4 KHz	IRQ4_2 KHz	IRQ4_1 KHz	IRQ5_4 KHz	IRQ4_2 KHz	IRQ6_T MB1	IRQ6_T MB2

b15	b14	b13	b12	b11	b10	b9	b8
FIQ_Fos c/1024	IRQ0_Fo sc/1024	FIQ_TMA	IRQ1_TM A	FIQ_TMB	IRQ2_TM B	IRQ3_EX T2	IRQ3_EX T1

② 清除中断标志控制单元 P_INT_Clear (写) (7011H)

清除中断标志控制单元主要用于清除中断控制标志位，当 CPU 响应中断后，会将中断标志位置为“1”，进入中断服务程序后，要将其控制标志清零，否则 CPU 总是执行该中断。

表 5-5 清除中断标志控制单元 P_INT_Clear

b7	b6	b5	b4	b3	b2	b1	B0
IRQ3_K EY	IRQ4_4 KHz	IRQ4_2 KHz	IRQ4_1 KHz	IRQ5_4 Hz	IRQ5_2 Hz	IRQ6_T MB1	IRQ6_T MB2

b15	b14	b13	b12	b11	b10	b9	b8

FIQ_Fo sc/102 4	IRQ0_ Fosc/1 024	FIQ_TM A	IRQ1_T MA	FIQ_TM B	IRQ2_T MB	IRQ3_E XT2	IRQ3_E XT1
-----------------------	------------------------	-------------	--------------	-------------	--------------	---------------	---------------

软件设计中一定要按照上述时基中断的使用方法正确设置相应寄存器。

5.2.2 储液检测以及输液剩余时间计算模块

储液检测程序的设计类似点滴速度检测，在储液检测中用到 `Irq3_ext2` (I0B3) 中断，其用法同 `Irq3_ext1`。当液位下降到 2~3 厘米时，产生一个外部中断，进行声光报警。

输液剩余时间的确定是根据储液剩余量以及当前点滴的速度来计算的，具体的方法是：由于总的储液量为 250ml，1 滴储液大概相当于 0.17ml，因此剩余储液量 `Numpos0` 的计算公式如 5-1 所示，剩余输液时间的计算公式如 5-2 所示。

$$\text{Numpos0} = 250 / 0.17 - \text{Tdrop} \quad (5-1)$$

$$\text{剩余储液量时间 } t = \text{Numpos0} / \text{Dropspd} \quad (5-2)$$

其中 `Tdrop` 表示一瓶液体点滴已经注射出去的点滴总数，`Dropspd` 表示当前点滴速度。

5.2.3 键盘和数码显示模块

对点滴速度的设置和显示采用由 HD7279A 芯片驱动的 4×4 键盘和 LED 组合来完成，当设置完毕后便会在 LED 数码管上显示并将所设置的速度送到设置的 Buffer 中，在 10s 检测的模块中会将设置的值和当前点滴的速度进行对比，这里误差设置为 12，如果误差大于 12，则通过电机转动来调节点滴速度，反之，就停止电机转动，认为点滴速度是相同的。

HD7279A 键盘和显示驱动程序的设计流程图如图 5-5 所示，当键盘输入数据后，首先将所得到的键盘码进行处理，然后送回 HD7279A 显示。设计时具体包括定义 RAM、定义 I/O、发送命令和数据字节到 HD7279、从 HD7279 接收数据字节、延时子程序等模块。

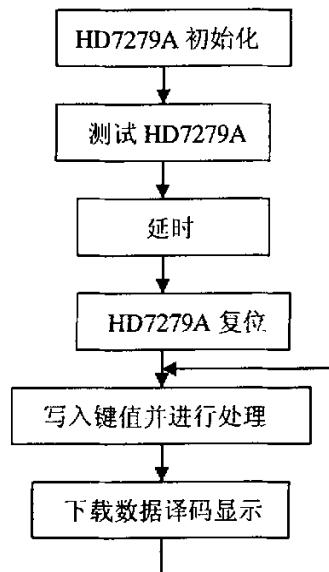


图 5-5 HD7279 键盘和显示驱动程序框图

5.2.4 LCD 显示模块

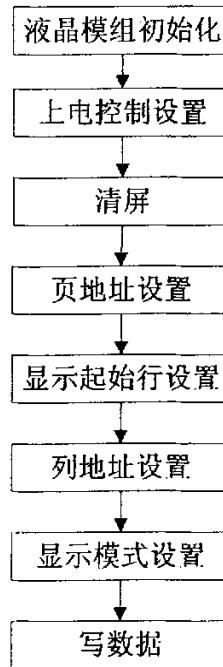


图 5-6 液晶模块驱动程序流程图

下位机接收上位机发送的液体点滴速度设置值或者其它数据（包括一些相关汉字命令），并立即在 LCD 屏上显示出来，当然也可以实时显示下位机检测到的点滴

速度值。

液晶模组的驱动芯片是 SPLC501, 它共有 23 种显示指令, 分别是显示开关指令、显示起始行设置、页地址设置、列地址设置、读状态、写显示数据、读显示数据、ADC 选择 (Segment 方向选择)、正向/反向选择、全屏点亮/变暗、LCD 偏压设置、读/改/写、结束、复位、正常输出选择模式、上电控制设置、电压内部电阻调整设置、电量设置模式、静态指示器、页闪动、驱动模式设置、节电模式、空命令等。液晶模块驱动程序流程图如图 5-6 所示, 主要包括液晶模组初始化、上电控制设置、清屏、页地址设置、显示起始行设置、列地址设置、显示模式设置、写数据等八个步骤, 每一步都按照上述对应的指令来设计。

5.2.5 点滴速度的电机控制模块

点滴速度控制部分采用步进电机控制输液瓶的高度来控制点滴下落的速度。系统采用步进电机, 以单八拍脉冲控制方式工作, 即 A、AB、B、BC、C、CD、D、DA。具体方法是根据设定的点滴速度值和当前测到的点滴速度值之差来控制步进电机的转动, 如果速度差大于零且大于 12 滴, 这时需要电机逆时针转动, 以带动储液瓶上移; 相反, 如果速度差小于零且绝对值大于 12 滴, 这时需要电机顺时针转动, 以带动储液瓶下移; 否则电机停止转动。电机控制程序流程图如图 5-7 所示。

需要注意的是: 本系统在采集点滴速度时, 是每 10 秒钟累计一次数据和(即点滴速度总数), 再把这个点滴数乘以 6 得出每分钟的点滴速度。因此得出每分钟的点滴数始终是 6 的倍数, 当我们需要设置一个不是 6 的倍数时, 应该怎样来处理这个数呢? 方法是: 用设置速度 (SV) 减去当前速度 (PV), 把速度之差的绝对值除以 6 求余, 然后判断设置速度与当前速度之差是否大于 0, 再判断该余数大于 3 还是小于 3。若设置速度大于当前速度且余数大于 3, 则用设置速度加上 6 与该余数之差, 例如: 当 $SV=70, PV=54$ 时, 处理后的 $SV1$ 为 72; 反之, 若该余数小于 3, 则用设置速度减去该余数, 例如: 当 $SV=68, PV=54$ 时, 处理后的 $SV1$ 为 66。若设置速度小于当前速度且余数大于 3, 则用设置速度减去 6 与该余数之差, 例如当 $SV=38, PV=54$ 时, 处理后的 $SV1$ 为 36; 反之, 若该余数小于 3, 则用设置速度加上余数。

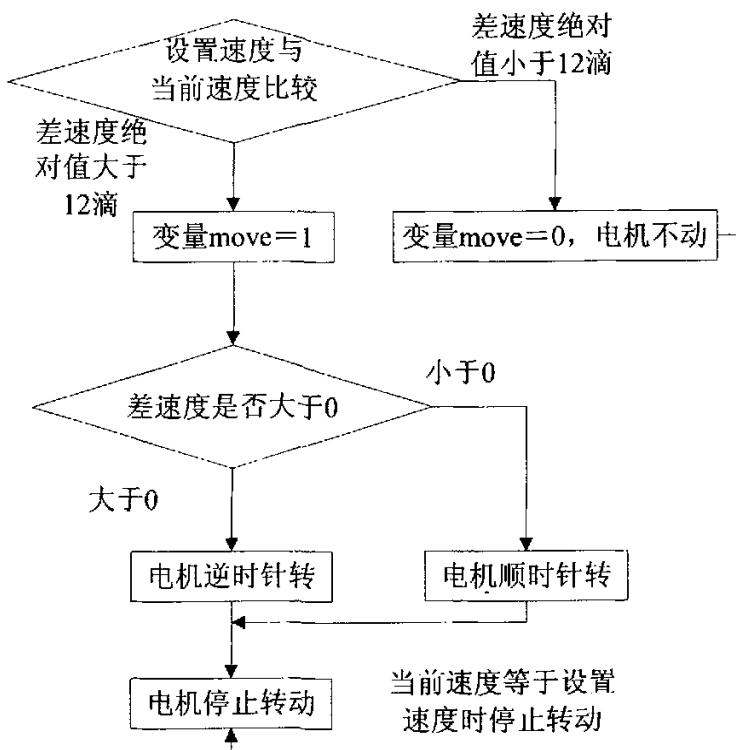


图 5-7 电机控制流程图

5.2.6 串口通讯模块和 SPR4096 数据存储模块

要实现上下位机通信, 下位机必须有通信模块和存储模块。SPCE061A 单片机的 UART 模块, 借助于 IOB 口的特殊功能和 UART IRQ 中断, 可以同时完成 UART 串行口接收和发送数据的过程, 并可以缓冲地接收数据。

SPR4096 芯片是一个高性能 4M-bit FLASH, 分为 256 个扇区, 用来存储数据, 对它编程操作之前, 必须先进行擦除操作。具体设计过程是:首先分别对 S10、UART、LCD 进行初始化, 开 2Hz 中断用来定时清看门狗, 然后置 IOB11 口为低电平, 选择 SPR4096 的 FLASH, 并对 SPR4096 的所有扇区进行擦除, 开始等待接收数据, 接收完后在 LCD 上显示, 这时再让 LCD 清屏, 同时根据上位机的命令发出相应的数据, 如图 5-8 所示。

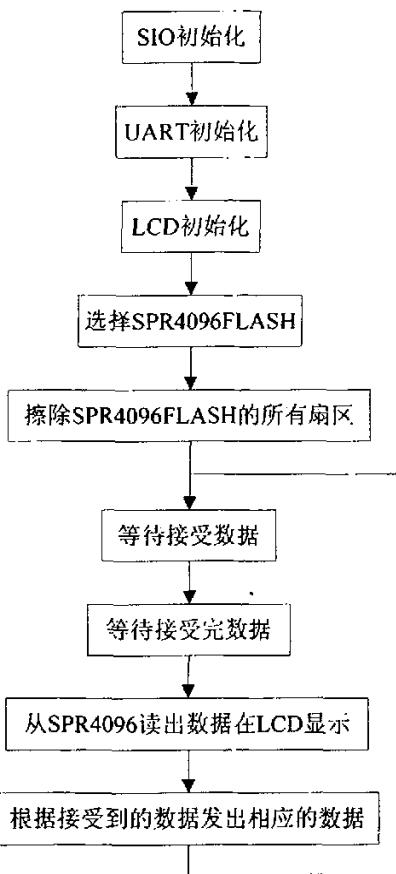


图 5-8 下位机程序设计流程图

下位机系统等待接收的数据是上位机发出的开始命令以及所设定的点滴速度值。当接收到上位机发出的开始命令后，继续等待接收数据，接收到设定点滴速度后在液晶上显示，并且及时改变下位机点滴速度值，使得点滴速度值尽快趋近于设定的点滴速度值。当上位机没有向下位机发送命令时，下位机一直在不断的向上位机发送当前点滴速度值。

5.2.7 语音播报模块

语音播报模块有两个功能，第一，当液位达到危险警戒值时报警，第二，用来播报点滴速度值。

一、常用应用程序接口 API 的功能介绍

麦克风输入所生成的 WAVE 文件占用的存储空间很大，对于单片机来说想要存储大量的信息显然是不可能的。凌阳 SPCE061A 提出了解决办法，即 SACM-LIB，该库将 A/D、编码、解码、存储及 D/A 做成相应的模块（如图 5-9 所示），对于每个模块都有其应用程序接口 API，调用 API 函数就可实现某些功能。表 5-6 列出了凌

阳音频的几种算法。

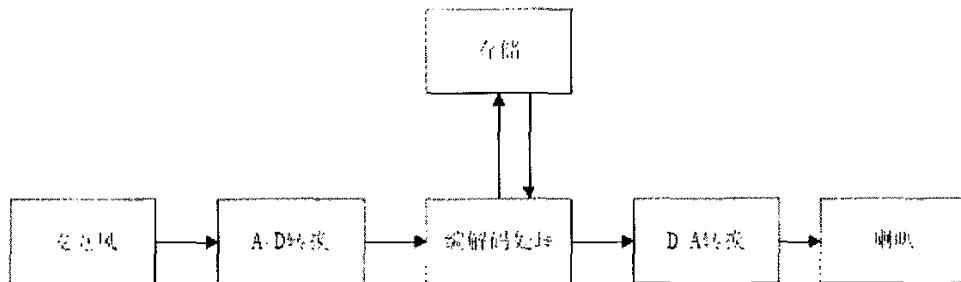


图 5-9 单片机对语音处理过程

表 5-6 SACM-lib 库中模块及算法类型

模块名称 (Model-index)	语音压缩编码率类型	资料采样率
SACM_A2000	16Kbit/s, 20 Kbit/s, 24 Kbit/s	16KHz
SACM_S480/S720	4.8 Kbit/s, 7.2Kbit/s	16KHz
SACM_S240	2.4 Kbit/s	24KHz
SACM_MS01	音乐合成 (16 Kbit/s, 20 Kbit/s, 24 Kbit/s)	16KHz
SACM_DVR(A2000)	16 Kbit/s 的资料率, 8K 的采样率, 用于 ADC 通道录音功能	16KHz

本系统采用凌阳音频编码 SACM_S240 方式 (2.4 位/秒), SPCE061A 能容纳 210 秒的语音数据, 该压缩算法的压缩比较大, 为 80: 1.5。其相关 API 函数及作用如下所示。

```

int SACM_S240_Initial (int Init_Index) — 初始化
void SACM_S240_ServiceLoop (void) — 获取语音资料, 填入译码队列
void SACM_S240_Play (int Speech_index, int Channel, int Ramp_Set) — 播放
void SACM_S240_Stop (void) — 停止播放
void SACM_S240_Pause (void) — 暂停播放
void SACM_S240_Resume (void) — 暂停后恢复
void SACM_S240_Volume (Volume_index) — 音量控制
unsigned int SACM_S240_Status (void) — 获取模块状态
Call F_FIQ_Service_SACM_S240 — 中断服务函数
  
```

二、语音播报模块的具体设计

压缩算法 SACM_S240 的编码率为 2.4Kbps, S240 只有自动播放方式。在中断 FIQ_TMA 中, 通过 SACM_S240_ServiceLoop () 对语音进行解码, 然后将其送入 DAC

通道播放，其软件流程图如图 5-10 所示。

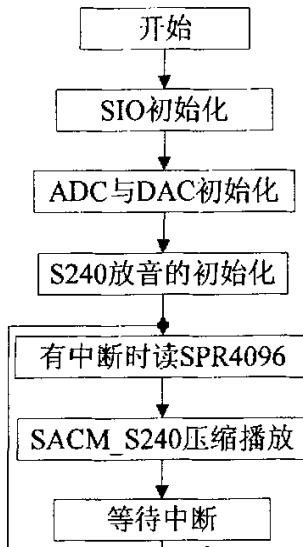


图 5-10 语音播报模块

5.3 下位机系统的软件调试

在确保下位机系统硬件设备连接好的前提下，利用在线调试器 probe，将程序下载至 SPCE061A，然后运行程序，同时打开液体点滴开关，调节输液滑轮使得速度达到允许范围内的某个值，然后用键盘输入一个合理的值，观察系统的运行调整过程。

1. 点滴速度检测模块检测到的速度基本正确。方法是用一个秒表来定时 30 秒，然后观察点滴数，计算出 1 分钟的点滴数，把这个数和用软件检测到的点滴数进行比较，发现基本相等。

2. LED 上能正确显示当前点滴速度值和键盘输入值。
3. 在 5min 内点滴速度值达到键盘设置的值，调整值和调整时间如表 5-7 所示。
4. LCD 能正确显示来自上位机传来的数值。
5. 储液量到达危险值时能及时报警。

表 5-7 滴速调整表

设定滴速 (滴/min)	实际滴速 (滴/min)	调整时间 (s)
116	114	100
97	96	60
60	60	80
40	36	40

第六章 液体点滴监控系统上位机的设计与实现

采用 MCGS 组态软件开发的液体点滴上位机监控系统，能对液体点滴的滴速和储液液位进行全面监控。本章首先分析了 MCGS 组态软件的系统构成和应用系统的结构，然后论述了上位机监控系统程序的具体设计。

6.1 MCGS 组态软件的系统构成

MCGS 组态软件由组态环境和运行环境组成，其系统构成如图 6-1 所示。

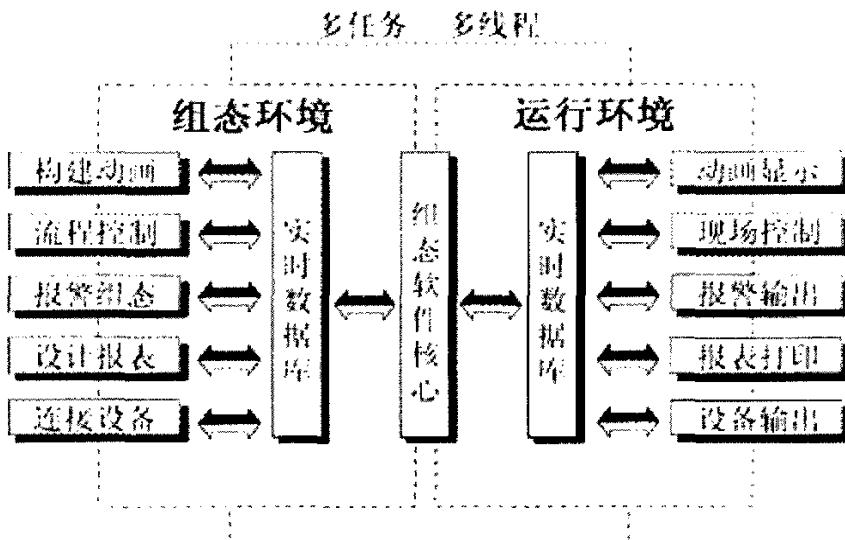


图 6-1 MCGS 组态软件系统构成图

组态环境是生成用户应用系统的工作环境，它相当于一套完整的工具软件，所有的组态配置过程都在组态环境中完成，具体来说用户在组态环境中完成动画设计、设备连接、构造数据库、编写用户策略、绘制曲线、报表等工作。运行环境是用户应用系统的运行环境，运行环境是一个独立的运行系统，它按照组态结果数据库中用户指定的方式进行各种处理，完成用户组态设计的目标和功能，运行环境本身没有任何意义，必须与组态结果数据库一起作为一个整体，才能构成用户应用系统，一旦组态工作完成，运行环境和组态结果数据库就可以离开组态环境而独立运行在监控机上。

6.2 MCGS 组态软件应用系统的结构

由 MCGS 生成的用户应用系统，其结构由主控窗口、设备窗口、用户窗口、实时

数据库和运行策略五个部分构成, 如图 6-2 所示。

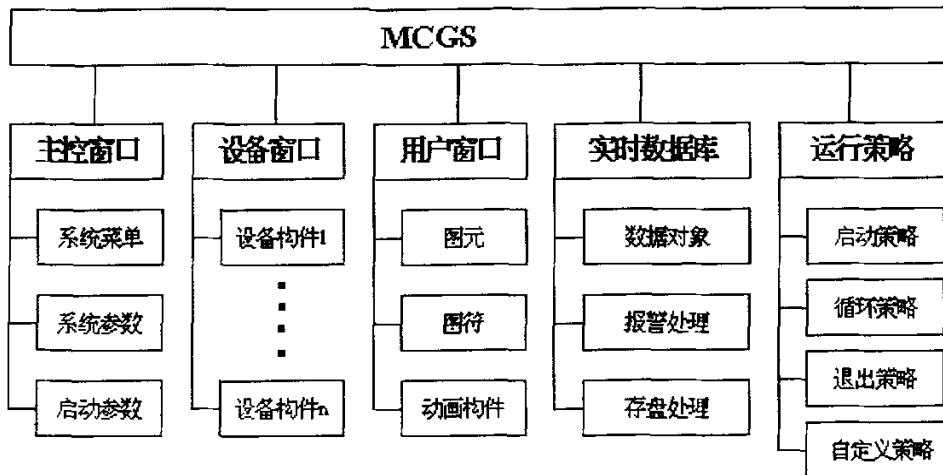


图 6-2 应用系统结构图

主控窗口是应用系统的主窗口或主框架。它确定了工程作业的总体轮廓、运行流程、菜单命令、特性参数、启动特性等内容。

设备窗口是 MCGS 系统与外部设备联系的媒体。它专门用来放置不同类型和功能的设备构件, 实现对外部设备的操作和控制。设备窗口通过设备构件把外部设备的数据采集进来, 送入实时数据库, 或把实时数据库中的数据输出到外部设备。

用户窗口是构成应用系统图形界面的基本单位。其中可以放置三种不同类型的图形对象: 图元、图符和动画构件。不同的动画构件对应不同的动画功能, 它们是根据工程实践经验, 总结出来的常用的动画显示和操作模块, 用户可以直接使用这些动画构件。图元、图符对象则为用户提供了一套完善的设计制作图形画面和定义动画的方法。通过在用户窗口内放置不同的图形对象, 用户就可以用不同的形式实现实时数据的“可视化”。各种复杂的图形界面, 可以由多个用户窗口搭制而成。组态过程中, 根据需要可以创建多个用户窗口, 最多可定义 512 个。所有的用户窗口均位于主控窗口内, 其打开时, 窗口可见; 关闭时, 窗口不可见。允许多个用户窗口同时处于打开状态。用户窗口的位置、大小、边界等属性能随意改变和设置, 如: 可以让一个用户窗口在顶部作为工具条, 也可以放在底部作为状态条, 还可以使其成为一个普通的最大化显示窗口。多个用户窗口的灵活组态配置, 就构成了丰富多彩的图形界面。

实时数据库是 MCGS 的核心, 相当于一个数据处理中心, 同时也起到公用数据交换区的作用。MCGS 用实时数据库来管理所有实时数据。从外部设备采集来的实时数据送入实时数据库, 系统其它部分操作的数据也来自于实时数据库, 实时数据库自动完成对实时数据的报警处理和存盘处理, 同时, 还根据需要把有关信息以事件的方式发送给系统的其它部分, 以便触发相关事件, 进行实时处理。因此, 实时数

数据库所存储的单元，不单单是变量的数值，还包括变量的特征参数（属性）及对该变量的操作方法（报警属性、报警处理、存盘处理等）。这种将数值、属性、方法封装在一起的数据称为“数据对象”。实时数据库采用面向对象的技术，为其它部分提供服务，提供了系统各个功能部件的数据共享。

运行策略是指系统的运行流程受条件控制，系统执行某项操作和实现某种功能是有条件约束的。运行策略由多个复杂的功能模块组成，称为策略块，用来完成对系统运行流程的自由控制，使系统能按照设定的顺序和条件，操作实时数据库，控制用户窗口的打开、关闭以及设备构件的工作状态，从而达到对系统工作过程精确控制及有序调度管理的目的。一个应用系统有三个固定的运行策略：启动策略、循环策略和退出策略。系统允许用户创建或定义最多512个用户策略。启动策略在应用系统开始运行时调用，退出策略在应用系统退出运行时调用，循环策略由系统在运行过程中定时循环调用，用户策略供系统中的其它部件调用。

6.3 液体点滴上位机监控管理系统的总体设计

进行上位机监控系统设计时，要考虑多用户的监控需求，以便扩展成多用户监控系统。上位机监控管理系统的总体设计框图如图6-3所示。

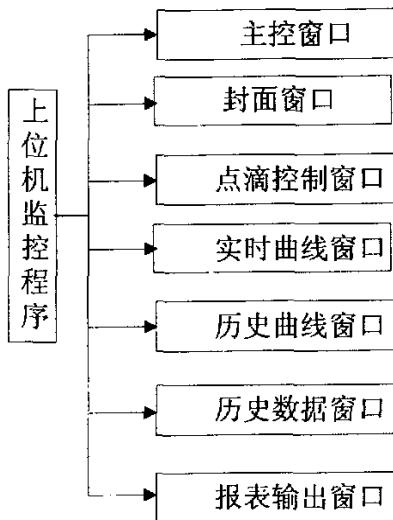


图 6-3 上位机监控管理系统的总体设计

系统要求对液体点滴的滴速和储液瓶液位等参数能够实时采集，同时用户可以自己设定点滴速度值，并在尽可能短的时间内让点滴速度值达到设定值。由于下位机是一套相对完整的监控系统，因此在上位机监控系统中不需要编写各种复杂的控制程序，只需要把点滴速度设定值送到下位机，由下位机来控制电机的转动即可。

6.4 液体点滴上位机监控管理系统的详细设计

6.4.1 主控窗口的组态

主控窗口是工程的主窗口或主框架，其窗口设计如图6-4所示。用户窗口组态完成之后，在主控窗口中，通过对系统菜单和参数的定义和设置来调度和管理这些用户窗口的打开或关闭。

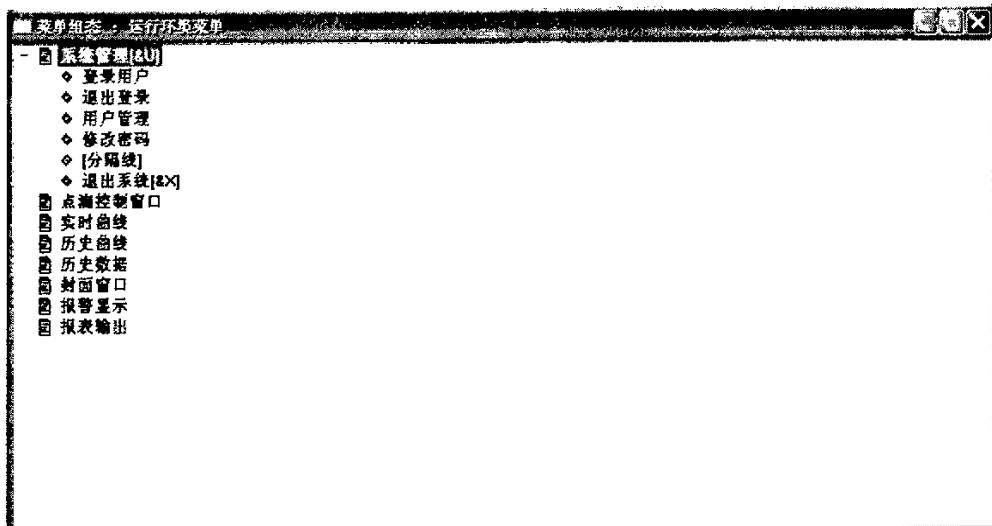


图 6-4 主控窗口

主控窗口包括系统管理窗口、点滴控制窗口、实时曲线窗口、历史曲线窗口、历史数据窗口、封面窗口、报警显示和报表输出等。系统管理是对监控系统进行管理和维护的功能菜单项，为保障系统安全运行，采用分级管理模式，设定不同的管理权限。系统管理包括登录用户、退出登录、用户管理、修改密码、退出系统等功能。

6.4.2 数据部分

MCGS把数据对象的属性封装在对象内部，作为一个整体，由实时数据库统一管理。在组态环境工作窗口中选择“实时数据库”标签钮，进入实时数据库窗口页。

“新增对象”按钮，用于在数据对象列表中增加新的成员。若要在对象列表的某一位置增加一个新对象，就可在该处选定某一数据对象，按“新增对象”按钮即可。若不指定位置，则在对象表的最后增加一个新的数据对象。新增对象的名称以前一对象名称为基准，按字符顺序递增的约定由系统缺省确定。

新增对象的属性设置，包括基本属性、存盘属性和报警属性。基本属性包括定

义数据对象的名称、数据类型和相关属性，数据对象有开关型、数值型、字符型、事件型、组对象五种类型。在实际应用中，数字量的输入输出对应于开关型数据；模拟量的输入输出对应于数值型数据对象；字符型数据对象是记录文字信息的字符串；事件型数据对象用来表示某种特定事件的产生及相应时刻，如报警事件、开关量状态跳变事件；组对象用来表示一组特定数据对象的集合，以便于系统对该组数据统一处理。在MCGS中报警被作为数据对象的属性，封装在数据对象内部，由实时数据库统一处理，用户只需按照报警属性窗口页中所列的项目正确设置，如数值量的报警界限值、开关量的报警状态等。运行时由实时数据库自动判断有没有报警信息产生、何时产生、何时结束、何时应答，并通知系统的其它部分；也可根据用户的需要，实时存储和打印这些报警信息。MCGS把数据的存盘处理作为一种属性或是一种操作方法，封装在数据内部以整体处理。运行过程中实时数据库自动完成数据存盘工作，用户不必考虑这些数据如何存储以及存储在什么地方。用户的存盘要求在存盘属性窗口页中设置，存盘方式有两种：按数值变化量存盘和定时存盘。组对象以定时的方式来保存相关的一组数据，而非组对象则按变化量来记录对象值的变化情况。

按此方法新增四个对象：Numpos1用来表示输液瓶的液位高度；Dropspd用来表示点滴速度；Swspd用来表示开关控制速度；Tdrop用来表示一瓶液体点滴已经注射出去的点滴总数。其中Numpos1对象初值为250，工程单位为ml；Dropspd对象初值为0，工程单位为滴/分钟；Swspd对象初值为0，工程单位为滴/分钟；Tdrop对象初值为0，工程单位为滴。

6.4.3 用户界面

打开组态环境下的用户窗口进入用户界面设计窗口，包括封面窗口、液体点滴监控系统操作流程界面、状态条、实时曲线、历史曲线等界面。

一、封面窗口

如果要扩展系统为多用户系统，需要在封面窗口中设计进入各个用户系统的界面。通过该界面，操作人员可以点击相应按钮进入相应用户界面来监控该用户的液体点滴情况，并且操作人员可以在封面窗口中浏览各个用户系统的点滴速度值，如图6-5所示。

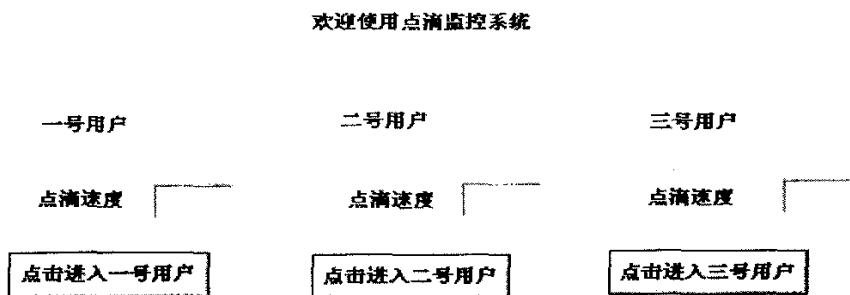


图 6-5 系统封面窗口

二、液体点滴监控系统操作流程界面

如图6-6所示的液体点滴监控系统操作流程界面是一个用户系统的主界面，为操作人员和学习人员提供了一个易于操作的环境。使用者可以从此界面切换到系统任意一个其它界面。

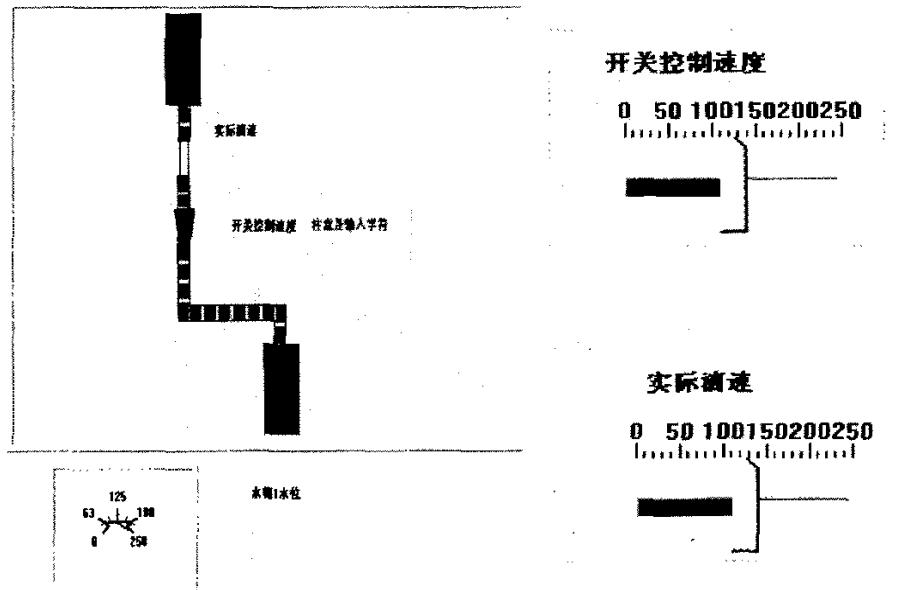


图 6-6 用户界面主图

主界面左边模拟了液体瓶、一次性输液器、控制开关；右边有开关用来控制液滴速度，形象直观；左下方旋转式开关用来模拟滴瓶的液位；可以在如图6-7所示速度控制图框中输入想要达到的液体点滴速度值。

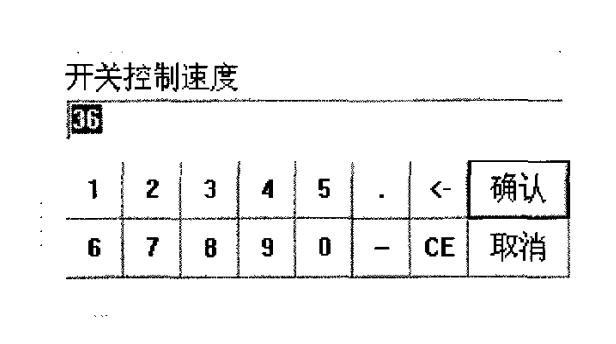


图 6-7 控制速度的图框

三、实时曲线和历史曲线

利用MCGS的实时数据、实时曲线、历史数据、历史曲线等功能，可以查看各个量的实时数据值、历史数据值，同时可以将各个数据、曲线打印等。系统中记录的数据为液体点滴速度的当前值和设定值，所保存的数据可以根据需要随时查询或随时删除。由于增加历史曲线的方法与实时曲线的方法相同，因此下面只介绍增加实时曲线的方法。

MCGS动画构件工具箱内，专门为用户提供了生成实时曲线图形的构件，称为实时曲线构件。该构件由坐标网格、刻度标注和曲线三部分组成，一个构件可同时显示6条曲线。横坐标可以代表绝对时间，此时构件所显示的是数据对象的值随时间变化的函数关系。也可以是另一个数据对象的值，曲线则表示两个数据对象之间相关性的变化关系。

用工具箱中的“实时曲线”构件新增实时曲线后，打开实时曲线构件的基本属性设置窗口。其设置要点如下：设置坐标网格的线型、颜色和刻度线数目。在“背景网格”栏目中，设置X、Y方向的刻度线数目均为12，线型和颜色默认即可；在“曲线类型”栏目中，选择“绝对时钟实时趋势曲线”选项钮；在“标注属性”一栏，设置X轴的长度为120，在“Y轴标注”栏目中，设定刻度值的最大和最小值分别为150、0（比例关系是10：1，因为250ml的液体大概是1500滴）；在“画笔属性”一栏，设置曲线的线型和颜色，并与曲线所表示的数据对象建立连接关系。在“画笔对应的表达式和属性”栏目中，曲线1、曲线2和曲线3输入框内分别设定Dropspd、Swspd和Numpos0（Numpos0的单位为滴，与Numpos1之间可以互相换算得来，即 $Numpos0 = Numpos1 / 0.17$ ），同时指定相应的线型和颜色；鼠标点击“可见度”标签，弹出曲线的可见度属性窗口，在“表达式”一栏内，空白，或输入非0数字，表示构件始终可见；在“当表达式非零时”栏目中，选择“实时曲线构件可见”选项钮。最终运行效果如图7-1所示。

6.4.4 运行策略

运行策略是对系统运行流程实现有效控制的手段。它用来完成和实现对系统运行流程的自由控制，使系统能按照设定的顺序和条件操作实时数据库、控制用户窗口的打开或关闭、确定设备构件的工作状态等，从而实现对外部设备工作过程的精确控制。根据系统控制算法及要完成的特定流程和操作处理，在MCGS的“用户策略”、“循环策略”、“存盘策略”、“事件策略”、“启动策略”、“退出策略”等部分分别进行组态和设置。这几种策略除策略的启动方式各自不同之外，其功能本质上没有差别，用户策略自己并不启动，需要其他策略、按钮、菜单等调用，循环策略是按设定的循环时间自动循环运行，事件策略是等待某事件发生后启动运行。本系统中没有用到用户策略组态和事件策略组态，启动策略和退出策略为系统固有策略。启动策略在MCGS系统开始运行时自动被调用一次，启动策略属性设置如图6-8所示。退出策略在退出MCGS系统时自动被调用一次，退出策略属性设置如图6-9所示。下面重点介绍一下循环策略组态。

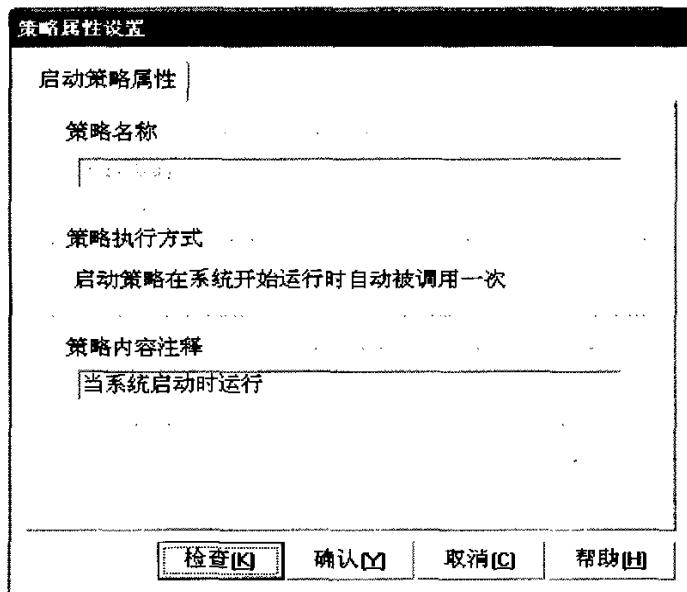


图 6-8 启动策略属性设置

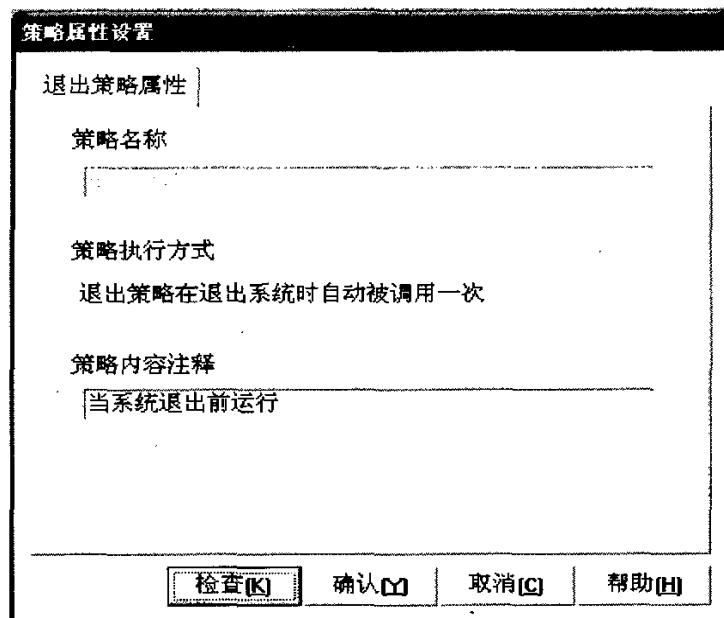


图 6-9 退出策略属性设置

对循环策略的组态是利用策略工具箱的脚本程序构件，用类似于Basic语言的命令语言编写该监控系统的控制算法、控制流程和操作处理程序，同时设置策略行条件属性。一个系统必须有一个循环策略，其组态方法如下：打开策略构件工具箱，选中“脚本程序”构件，如图6-10所示。

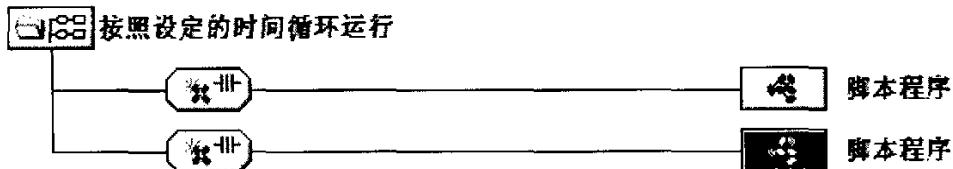


图 6-10 循环策略

编辑的脚本程序如下所示：

NumPos1=250-tdrop (总点滴数) × 0.17

IF NumPos1<= 20 THEN swspd = 0

当液量剩下20ml时，将驱使报警装置发出报警。

第七章 液体点滴监控系统的系统联合调试

按照系统设计将上、下位机用标准 R232 线连接，首先运行下位机，然后打开上位机监控程序，设置设备驱动程序中的通信波特率为 9600kps，进入运行环境，观察上位机监控程序的实时曲线，在这个过程中还可以在上位机监控程序中设置想要的点滴速度值。测试到的实时曲线如图 7-1 所示。

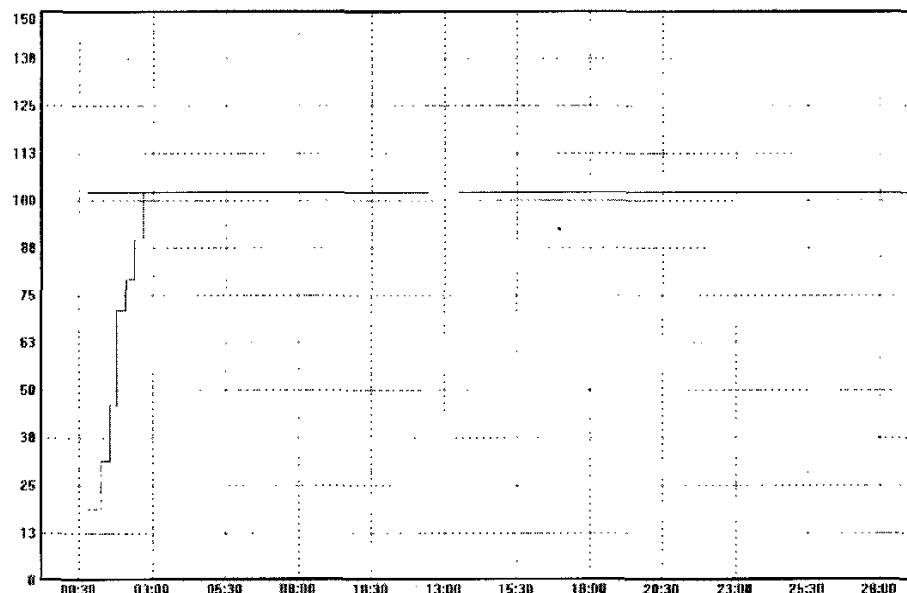


图 7-1 实时曲线

图中红色曲线表示当前值，蓝色直线表示设置值，绿色曲线表示输液瓶的液位。可以看出，在 1min 后，点滴速度值达到了设置的速度值。

刚开始调试时，其结果可能会不尽人意，但可以从以下几个方面分析原因，寻求解决办法。

1. 上下位机之间串行通信是否顺畅；
2. 上位机数据库与设备驱动程序通道连接是否正确；
3. 下位机能否采集到相应传感器输出信号；
4. 下位机能否控制继电器、电机等外围设备的运行。

结束语

课堂教学要适应素质教育的要求，不仅要让学生掌握必要的科学知识，还要为他们的素质发展奠定必要的科学方法基础。本文设计的液体点滴监控系统模型正是基于这种思想。

以单片机、监控软件为基本工具，运用科学的思维方法，构建监控系统模型，是一种培养学生动手能力、创造思维和探索能力的重要途径。在监控系统中，监控组态软件是上位机系统的数据收集处理中心、远程监视中心和数据转发中心，处于运行状态的监控组态软件与以单片机为核心的下位机控制系统构成一个整体。由于组态软件具有良好的人机操作界面，所以大大加快了应用系统的开发速度。如果单纯依靠计算机程序语言去编制一个大型监控系统的软件，不仅费时、费力、而且通用性差，不便于系统的扩充和升级，也不便于推广和应用。

本文只完成了对单个用户的液体点滴情况进行监控，由于系统有良好的可扩展性，今后可以把对单个液体点滴系统的监控扩展成为对多个液体点滴系统的群控。另外，本系统采用的是单机版组态软件，今后可以采用网络版组态软件，通过网络对现场的各种事务进行浏览和控制。

参考文献

- [1] 于仲安. 组态软件 MCGS 在微机监控加药系统中的应用. 电子技术, 2003 年, 12
- [2] MCGS 快速入门. 北京昆仑通态自动化软件科技有限公司, 2000
- [3] MCGS 用户指南. 北京昆仑通态自动化软件科技有限公司, 2000
- [4] 王国辉. 智能型医用输液泵及其应用. 世界医疗器械, 2002, 3
- [5] 雷思孝, 李伯成, 雷向莉等. 单片机原理及实用技术. 西安电子科技大学出版社, 2004
- [6] 曾强聪. Visual Basic 程序设计与应用开发案例教程. 清华大学出版社, 2004
- [7] 李善茂, 杜大鹏, 刘国宏等. Visual Basic 6.0 高级编程技巧. 电子工业出版社, 1999
- [8] 张培仁, 张志坚等. 十六位单片微处理器原理及应用. 清华大学出版社, 2005
- [9] 王晓明. 电动机的单片机控制. 北京航空航天大学出版社, 2002
- [10] 白雷, 沈安文, 郑定富等. 在 MCGS 环境下开发 ST2000 设备驱动. 自动化博览, 2003
- [11] 李训杰. MCGS 组态软件在供水自动化监控系统中的应用. 工业控制计算机, 2005
- [12] 孟凡德, 张颖, 罗铃等. MCGS 组态软件的应用及驱动程序开发. 化工自动化及仪表, 2004
- [13] 闵华松, 张颖, 陈奎生等. 基于 SPCE061A 的串行数据采集及控制系统的设
计. 武汉科技大学学报, 2003
- [14] 硬道理电子技术工作室. 步进电机原理及使用说明. 2005
- [15] 先锋工作室. 单片机程序设计实例. 清华大学出版社, 2004
- [16] Dave Harrold. Next generation control system technologies promote
solutions. Control Engineering, 2000
- [17] Amy J. Murtha. The development of a Configuration control Tool.
Maryland 21076
- [18] Wheater, S. M; Little, M. C . The design and implementation of a
framework for configurable software. 1996

致 谢

两年多的研究生学习生活即将结束了，在这里我要感谢我的导师梁光胜老师，感谢他对我的悉心指导和关心，从老师这里，学到的不仅仅是知识，更有严谨的治学态度、勤勉认真的工作作风。这些品格永远值得我学习，并定将使我在日后的工
作、学习中受益匪浅。在离开校园之际，再次感谢两年来导师对我的辛勤培养。

同时还要感谢研究生期间，实验室的老师们、同学们对我的热心帮助和支持，以及所有任课老师传授我丰富的专业知识。在这里表示真挚的感谢！

在学期间发表的学术论文和参加科研情况

[1] 李琴、梁光胜. 基于 MCGS 的凌阳单片机驱动程序设计. 微计算机信息, 2006, 11