

東北大學

NORTHEASTERN
UNIVERSITY

硕士学位论文

THESIS FOR MASTER'S DEGREE

论文题目 数字式称重仪及 CANopen 协议
的研究与实现

作者 李茂涛
学院 信息科学与工程学院
专业 检测技术与自动化装置
指导教师 李新光 副教授

备 注

二〇〇六年 二 月 日

数字式称重仪及 CANopen 协议的研究与实现

摘 要

随着现代工业和科学技术的发展,智能测量仪表在工业生产和科学研究等领域都获得了巨大的发展。本文结合自动化仪表实验室的热工综合实验系统的建设,自主开发了带 CAN 总线接口的智能式称重仪。

本文首先介绍了本课题的研究背景,设计过程中所用到的原理及理论,然后详细描述了其软硬件的实现过程,同时对控制网路中的现场总线进行了系统的描述,最后进行了误差分析及对本论文进行了总结。现场总线部分主要集中在第二章和第五章,第二章中介绍了 OSI 参考模型及各种网络的概念及术语,第五章中则对引出的一种现场总线(CAN 总线)进行了研究与实现,该章首先从其网络模型上入手,介绍了各层的功能及层次结构,然后重点转到其中的一层——应用层进行研究。应用层选取了在欧洲非常流行的 CANopen 协议,最后给出了数字式称重仪成为 CANopen 网络下最小能力节点的实现过程。本文的主要内容有:

(1)智能称重仪硬件电路的设计,包括高精密参考电压源、小信号放大电路及其非线性补偿电路、串行口通讯及总线通讯电路等的设计;

(2)软件编程及调试,实现了人机接口的参数设定、量程变换、去皮功能,数字滤波、分段线性化输出及串口通讯等功能;

(3)CANopen 协议研究,给出了 CANopen 协议的详细描述,同时按其标准完成了 CANopen 协议下最小能力设备的实现过程。

关键词: 在线测量称重仪; Cygnal 单片机应用; CANopen 协议; 对象字典; 最小能力设备实现

Research on Digital Weight Measurement and CANopen Protocol

Abstract

With the development of Modern Industry and Technology, intellectualized measurements have got a great improvement in industrial production fields and science research fields. Aiming at the upgrade of the lab's process parameters experiment synthesis system, this paper gives a project of Digital Weight Measurement and CANopen Protocol as an example to study intellectualized measurements.

This article firstly introduces the research background, the principle and the theory used in the design process; Then detailedly describes its software and hardware realization process, and the Field Bus in control network field. Finally carries on the error analysis and the summary of this paper. The Field Bus is mainly focused in second chapter and the fifth chapter, the second chapter introduces the OSI reference model, the network conception and terminology, the fifth chapter gives the detailed research of one kind of Field Bus – CAN. This chapter firstly shows its network model, introduces the layer structure and each layer function, then jumps to Application layer, starts to study the selected CANopen protocol which has been extremely popular in Europe, at last gives the realization process of the digital weight measurement to be one minimal function node in CANopen network.

(1)Hardware design of the digital weight measurement, includes high precise reference voltage supply, small signal amplifier circuit and its nonlinear compensation, the serial interface communication and Field Bus communication.

(2)Software programming and debugging, finished parameter demarcation, measuring range transformation, realization of dynamic zero function, software filtering, output partition linearization and serial communication program.

(3)Research of CANopen protocol, gives the detailed description and the project processing of one minimal function device in CANopen network.

Key words:On-line digital weight measurement; application for Cygnal Integrated Products; CANopen protocol; object dictionary; minimal function device

独创性声明

本人声明，所呈交的学位论文是在导师的指导下完成的。论文中取得的研究成果除加以标注和致谢的地方外，不包含其他人已经发表或撰写过的研究成果，也不包括本人已获得其他学位而使用过的材料。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谢意。

学位论文作者签名：李毅清

日期：2006.2.25

学位论文版权使用授权书

本学位论文作者和指导教师完全了解东北大学有关保留、使用学位论文的规定：即学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人同意东北大学可以将学位论文的全部或部分内容编入有关数据库进行检索、交流。

（如作者和导师不同意网上交流，请在下方签名；否则视为同意。）

学位论文作者签名：

导师签名：

签字日期：

签字日期：

第一章 绪 论

物料计量是工业生产和贸易流通中的重要环节,称重装置或仪器是不可缺少的计量工具,它不仅是提供重量数据的单体仪表,而且在工业生产管理现代化的过程中,起到了缩短作业时间,改善操作条件,降低能源和材料的消耗,提高产品质量以及加强企业管理,改善经管等多方面的作用。随着工农业生产的发展和商品流通的扩大,快速、准确、操作方便、消除人为误差、功能多样化^[1]的智能化称重仪表已获得越来越多的需求。

1.1 称重仪表的现状与发展方向

上世纪 50~60 年代,称重仪表表现为在传统机械式称重装置或机械衡器上发展起来的机电衡器,它的特点是借助于机电元件(继电器,键控电磁铁,光栅等),使机械衡器具备平衡状态识别功能及启动功能。其称量工作是机械式的,但与称量有关的功能则以电气方式显示,记录,远传或控制。进入 70~80 年代,鉴于机电元件受环境影响较大且有磨损现象,于是性能更可靠的电子元件取代了机电元件而形成了电子衡器。而从 90 年代后,随着微电子技术的发展以及现场总线技术的出现,称重仪表发展成为目前的智能称重仪表。它的测量精度高、自动补偿性能好、便于调校、具有控制和计算功能,能独立承担控制、通信任务,这使得上位机的控制功能分散到了底层网络,降低了系统故障的概率,为自动控制系统打下了坚实的基础,这体现了当今称重技术的发展趋势^[2]。

称重仪表一般由三部分组成:量值传感器是一只或多只称重传感器,用来直接把重量(或缩小的重量)变换成比例的电量,在称重仪表的结构中,它既可直接与秤台相连(纯电子秤或全电子秤),也可以通过杠杆机构与秤台相连(混和型电子秤或机电结合型电子秤);量值求取部分是处理称重传感器信号的电子线路,如放大器,模数转换器,电流源或电压源,调节器,补偿元件及保护线路等;数据输出部分包括所有把称量结果清晰的表现出来的部件,如显示,打印,数据传输和存储器件。目前常用的传感器有电力补偿式、振弦式、陀螺式、压磁式、电容式、音叉式、光纤应力式^[3]、光干涉式及辐射吸收式等,而最普遍使用的仍然是电阻应变式的。在称重仪表的设计思想上,主要有两种模式。美国和西欧等大多数采取终端机模式,即仪表本身主要解决专用功能,而把累计,编号,打印,统计报表等一般性的数据处理功能,通过接口传输到终端计算机去解决;日本等国家则采取小而全的模式,即集所有功能于一身,这种称重仪表比较适用于管理手段薄弱的小型企业。而无论哪种模式,智能仪表对网络的支持都已不可或缺。本系统的设计倾向于终端机模式,通过对现场总线的支持来完成其网络功能。

1.2 现场总线

自从 1946 年世界第一台电子计算机问世以来,计算机网络系统得到了迅速的发展,

而自上世纪 90 年代后,在生产过程领域中的现场总线也得到了迅猛的发展,它已成为当今自动化领域技术发展的热点之一^[4]。它应用在生产现场,又称为开放式、数字化、多点通信的底层控制网络。现场总线技术将专用的微处理器置于传统的测量控制仪表^[5],使它们各自都具有了数字计算和数字通信能力,采用可进行简单连接的双绞线等作为总线,把多个测量控制仪表连接成网络系统,并按公开、规范的通信协议,在位于现场的多个微机化测量控制系统设备之间以及现场仪表与远程监控计算机之间,实现数据传输与信息交换,形成各种适应实际需要的自动控制设备变成网络节点。简而言之,它把单个分散的测量控制设备变成控制节点,以现场总线为纽带,把它们连接成可以相互沟通信息,共同完成自控任务的网络系统与控制系统。而为把不同厂商之间的设备进行兼容通信,就要求形成统一的标准,组成开放互连网络。从这个意义上说,现场总线就是工厂自动化领域的开放互连系统。它具有如下技术特点:开放性;互可操作性与互用性;设备的智能化与功能自治性;系统结构的高度分散性;对现场环境的适应性。其优点是:节省硬件数量与投资;节省安装费用及维护开销;同时用户具有高度的系统集成主动权,也提高了系统的准确性与可靠性。以下是几种有影响的现场总线技术^[6]:

(1)基金会现场总线

基金会现场总线前身是以美国 Fisher-Rosemount 公司为首,联合 80 家公司制定的 ISP 协议和以 Honeywell 公司为首,联合 150 家公司指定的 World FIP 协议。它以 ISO/OSI 开放系统互连模型为基础,定义对应于其中的物理层、数据链路层、应用层的内容,在此基础上增加了用户层。

(2)LonWorks 总线

LonWorks 一种具有强劲实力的现场总线。是由 Echelon 公司推出,与 Motorola、Toshiba 公司共同倡导,于 1990 年正式公布而行成的。它是唯一一种采用全部 ISO/OSI 模型全部七层通讯协议的现场总线。

(3)PROFIBUS 总线

PROFIBUS 是德国国家标准 DIN19245 和欧洲标准 EN50170 的现场总线标准。由 PROFIBUS-DP, PROFIBUS-FMS, PROFIBUS-PA 组成了 PROFIBUS 系列。DP 型用于分散外设间的高速数据传输,适合于加工自动化领域的应用,而 PA 型是用于过程自动化的总线类型,采用了 OSI 模型的物理层、数据链路层, FMS 还采用了应用层。

(4)CAN 总线

CAN 全称为“Controller Area Network”,即控制器局域网,它是由 ISO 定义的,在国际上应用最广泛的现场总线之一。它最早由德国 BOSCH 公司推出,用于汽车内部测量与执行部件的数据通信,形成汽车电子控制网络。该总线是一种多主总线的串行数据通信协议,其通信介质可以是双绞线、同轴电缆或光导纤维,速率可达 1Mbps,传输距离可长达 10km,当信号传输距离达到 10Km 时,仍可提供高达 50Kbit/s 的数据传输速率。CAN 总线通信接口集成了 CAN 协议的物理层和数据链路层功能,可完成对通信

数据的成帧处理,包括位填充、数据块编码、循环冗余检验、优先级判别等工作。它的一个最大特点是废除了传统的站地址编码,而代之以对通信块进行编码。这种方法可使网络内的节点个数在理论上不受限制,数据块的标识码可由 11 位或 29 位二进制数据组成,因此可定义 2^{11} 位或 2^{29} 个不同的数据块,这种编码方式,可使不同的节点同时接收到相同的数据,这一点在分布式控制系统中尤其有用。数据长度最多为 8 个字节,可满足通常工业领域中控制命令、工作状态及测试数据的一般要求,同时 8 个字节占用时间不长,从而保证了通信的实时性。它采用了 CRC 检验并可提供相应的错误处理功能,保证了数据通信的可靠性。因此, CAN 已经在汽车工业、航空工业、工业控制、安全防护等领域中得到了广泛应用。其主要特性有:多主站依据优先权进行总线访问;无破坏性的基于优先权的仲裁;借助接受滤波的多地址帧传送;远程数据请求;配置灵活性;全系统数据相容性;错误检测和出错信令;发送期间若丢失仲裁或由于出错而遭破坏可自动重发送。

1.3 选题背景及研究内容

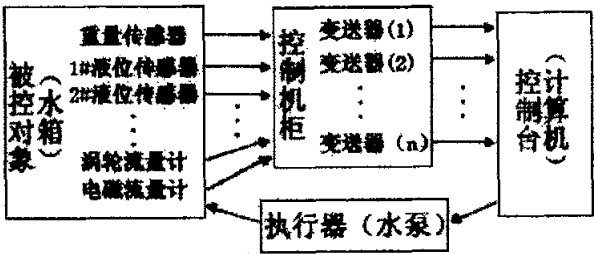


图 1.1 热工综合实验系统

Fig. 1.1 Many measurements integrated experiment system

本课题主要针对自动化仪表实验室的“热工综合实验系统”的升级,图 1.1 给出了该实验系统的控制原理框图。该实验系统由三大部分组成:水箱设备系统、控制柜和监控计算机。检测仪表(一次仪表)和水泵安装在水箱系统的管路上,分别完成各自的检测和控制功能(图 1.2);二次显示仪表、信号处理单元、变频器以及其它电气元件安装在控制柜内;最后经由二次仪表输出的标准信号和变频器的控制信号通过信号输入输出卡连入计算机,进行显示、运算与控制。由于一次仪表众多,造成与控制柜内的二次仪表连接复杂,而且其相互的干扰也对二次仪表的精度造成负面影响,因此简化系统的物理连接及保持未来的可扩展性就显得非常重要了。本课题以其中的一路称重信号做为切入点,重新研制现场仪表,提高检测性能的同时,增加对网络的支持,以保持整个系统未来的可扩展性。

本论文在充分吸收原有 Pacific PA8110A 称重变送器优点的基础上,利用压阻传感器,将重量值转换为弱电信号,然后通过小信号的放大电路进行高倍放大,送至单片机进行处理,转化成数字信号,再根据 CAN 协议的通信机制,做成现场总线智能测量仪

表。

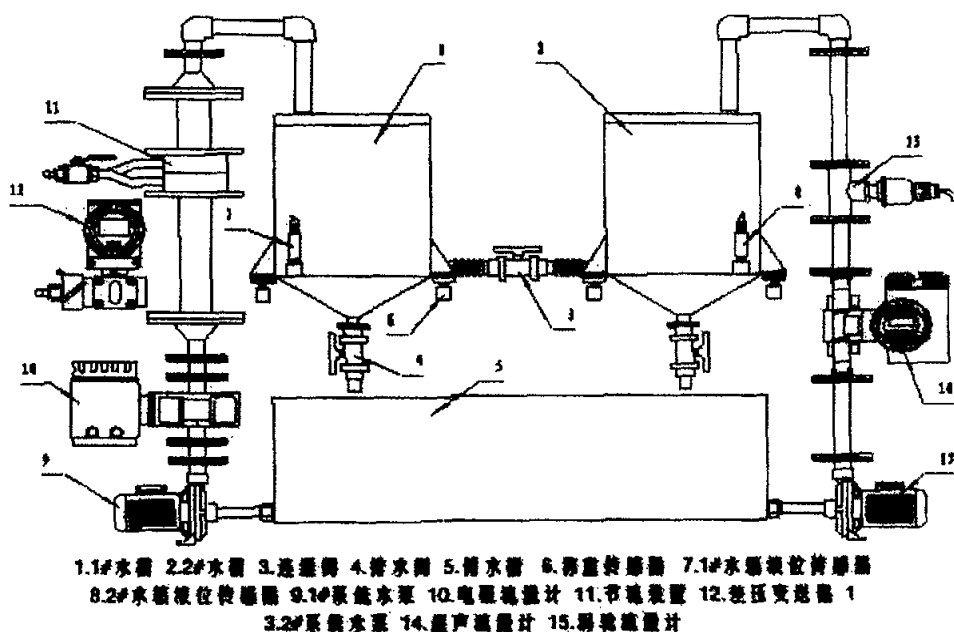


图 1.2 水箱设备系统

Fig. 1.2 Water tank subsystem

熟悉压电式传感器的工作原理及部分面向对象编程的原理，掌握 MSC-51 微控制器的应用及 C 语言编程，掌握 CAN 现场总线网络及 CANopen 通信协议的传输机制。

设计小信号放大电路，对小信号进行有效放大；设计放大电路的非线性补偿电路，消除放大倍数非线性对测量的误差；设计单片机的 RS-232 通讯接口及 CAN 总线接口电路，以完成仪表的通讯功能；然后对仪表整体原理结构进行设计，画出电路原理图，印制电路板，然后进行软件编程实现。

第二章 理论基础

2.1 压阻效应

金属或半导体材料的电阻值随它承受的机械形变大小而发生变化的现象称为压阻效应^[7]，又称应变效应。设电阻丝长度为 l ，截面积为 s ，电阻率为 ρ ，则电阻值 R 为：

$$R = \rho \frac{l}{s} \quad (2.1)$$

对上式进行微分，得

$$\frac{dR}{R} = \frac{d\rho}{\rho} + \frac{dl}{l} - \frac{ds}{s} \quad (2.2)$$

一般电阻丝为圆形截面， $s = \pi r^2$ ， r 为电阻丝半径，所以：

$$\frac{ds}{s} = 2 \frac{dr}{r} \quad (2.3)$$

由材料力学得知，轴向应变与横向应变的关系为：

$$\frac{dr}{r} = -\mu \frac{dl}{l} \quad (2.4)$$

式中 μ 为泊松系数，负号表示应变方向相反，所以有：

$$\frac{ds}{s} = -2\mu \frac{dl}{l} \quad (2.5)$$

将式 (2.5) 代入 (2.2) 中，得，

$$\frac{dR}{R} = (1 + 2\mu) \frac{dl}{l} + \frac{d\rho}{\rho} \quad (2.6)$$

$\frac{dl}{l}$ 通常称为应变，记作 ε ，则

$$\frac{dR}{R} = (1 + 2\mu + \frac{d\rho/\rho}{\varepsilon}) \varepsilon = k \varepsilon \quad (2.7)$$

式中 k 为电阻丝的应变灵敏度系数，即：

$$k = 1 + 2\mu + \frac{d\rho/\rho}{\varepsilon} \quad (2.8)$$

显然 k 受两个因数的影响， $(1 + 2\mu)$ 由电阻丝形变引起， $(d\rho/\rho)/\varepsilon$ 由电阻丝受力后材料的电阻率变化引起。若以 σ 表示应力，则

$$\frac{d\rho}{\rho} = \pi_e \sigma \quad (2.9)$$

式中 π_e 表示压阻系数, 根据虎克定律, 应力 σ , 应变 ε 与材料的弹性模量 E 有关系:

$$\sigma = E\varepsilon \quad (2.10)$$

$$\frac{d\rho}{\rho} = \pi_e E\varepsilon \quad (2.11)$$

将 (2.11) 代入 (2.8) 式中, 得

$$k = (1 + 2\mu) + \pi_e E \quad (2.12)$$

最后得到: $dR/R = k\varepsilon$, 一般金属材料的压阻效应很小, $k \approx 1 + 2\mu$, 其值在 2 左右, 半导体材料得压阻效应特别明显, π_e 约为 $(1 + 2\mu)$ 得 50~100 倍, $k \approx \pi_e E\varepsilon = \pi_e \sigma$ 。

2.2 称重传感器

把被称重物重量转化成电阻值的变化可直接用基于压阻效应的称重传感器来完成, 它主要由弹性体、应变计、检测电路三部分组成。弹性体是一个有特殊形状的结构件, 它的主要作用是将力转换为形变; 应变计是一种传感元件, 它的作用是将形变转变成电阻变化; 检测电路的主要部件是直流电桥, 它可以比较方便地解决称重传感器的补偿问题, 其功能是把电阻应变片的电阻变化转变为相应的电信号输出。

2.2.1 弹性体

弹性体是一个有特殊形状的结构件。它的功能有两个, 首先是它承受称重传感器所受的外力, 对外力产生反作用力, 达到相对静平衡; 其次, 它要产生一个高品质的应变场(区), 使粘贴在此区的电阻应变片比较理想的完成应变电信号的转换任务。

2.2.2 电阻应变片

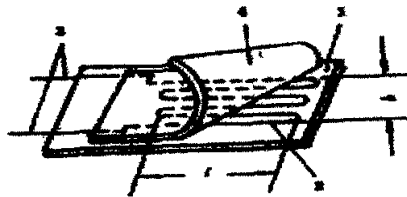


图 2.1 电阻应变片的结构

Fig. 2.1 Structure of resistor strain gage

电阻应变片的作用是把应变转化成电阻。其典型结构如图 2.1: 合金电阻丝 2 曲折形状(栅形), 用粘胶剂粘贴在绝缘基片 1, 电阻丝两端通过引线 3 引出, 丝栅上再粘贴在绝缘保护膜 4。应变丝栅的基长为 l , 宽度为 b , 把应变片粘贴于所需测量变形物体

表面,敏感栅随着被测物体表面形变而使电阻值改变,测量电阻的变化量便可得知形变的大小。电阻应变片作为应变测试手段已有几十年的历史了,其最大特点是:体积小,使用简便,测量灵敏度高,可以进行静态测量,动态测量,精度合于要求。而且随着材料和工艺技术的发展,超小型、高灵敏度、高精度的电阻应变片和传感器不断的出现,测量范围的不断扩大,使它已成为非电量电测技术中十分重要的手段。

2.2.3 直流电桥

在测量系统中使用的电桥分为平衡电桥和非平衡电桥两种。前者多采用直流供电,常用于物理实验中,调整调节臂电阻,使其达到平衡,并用其刻度表示被测量的值。后者应用比较广泛,它是利用电桥输出电流或电压与电桥各参数间的关系进行工作的(即输出值与其桥路电阻有关系,但其关系只存在线性关系,而没有对应的零点)。根据桥压源的不同,它又分为直流电桥^[8]和交流电桥。直流电桥又根据供电方式的不同分为恒压供电和恒流供电,下面是对恒压供电桥路的分析。

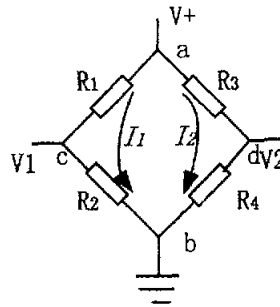


图 2.2 电桥原理图

Fig. 2.2 Bridge block circuit

直流供电电桥的电路如图 2.2, 其中 R_1 R_2 R_3 R_4 为桥臂电阻(电桥的桥臂也可以是其它元件,如电感等)。当 c 和 d 间开路或阻抗很大时

$$V_1 = I_1 R_2 \quad V_2 = I_2 R_4 \quad I_1 = \frac{V^+}{R_1 + R_2} \quad I_2 = \frac{V^+}{R_3 + R_4}$$

后两式代入前两式得

$$V_1 - V_2 = \frac{R_2 R_3 - R_1 R_4}{(R_1 + R_2)(R_3 + R_4)} V^+ \quad (2.13)$$

在电桥处于平衡时, $V_1 = V_2$, 因此有 $R_2 R_3 - R_1 R_4 = 0$ 即 $\frac{R_2}{R_1} = \frac{R_3}{R_4}$

该式即为直流电桥平衡条件,它说明欲使电桥达到平衡,其相邻桥臂的电阻比应该相等。在传感器中,根据压阻传感器的接法可分为全等臂电桥(全桥差动电路,相对两个受拉,相邻的两个受压)、双臂桥(半桥差动电路,相邻桥臂内一个受拉一个受压)和单臂桥(单桥)。

对半桥差动电路, 设 R_1, R_2 为应变电阻, $R_1 = R + \Delta R$, $R_2 = R - \Delta R$, 则由式 (2.13) 可得:

$$V_1 - V_2 = \frac{\Delta R}{R} \times \frac{1}{2} V^+ \quad (2.14)$$

分析该式可发现, $U_o = V_1 - V_2$, 主要依赖于 $\Delta R/R$, 对于输入桥路电压 $V^+/2$ 为常值时, U_o 的电压值线性反应了 ΔR 的变化量, 即电阻变化量与输出电压是线性关系, $V^+/2$ 为其灵敏度系数。同时从上式可发现该桥路的电压输出, 是全桥电压值的 $1/2$, 即半桥差动电路灵敏度是全桥差动电路的一半 (注: 差动方式, 可消除偶次方的影响, 从而有效改善其线性度)。

2.3 差动放大器

差动放大器实际为减法电路, 实现差值运算, 所以又称为差动运算放大器。

如图 2.3 所示, 同相输入端和反相输入端都接输入信号, 理想情况下, $I_1 = 0$, $U_- = U_+$, 所以有:

$$I_1 = I_2 \quad \frac{U_{i1} - U_-}{R_1} = \frac{U_- - U_o}{R_2} \quad U_+ = U_{i2} \times \frac{R_4}{R_3 + R_4}$$

解得: $U_o = U_{i2} \times \frac{R_4}{R_3 + R_4} \times \frac{R_1 + R_2}{R_1} - U_{i1} \frac{R_2}{R_1}$ (2.15)

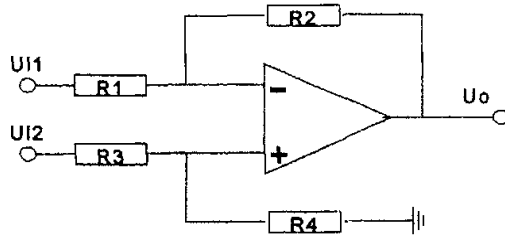


图 2.3 差值运放电路

Fig. 2.3 Differential operation diagram

当外电路电阻满足平衡对称条件 $R_1 = R_3$, $R_2 = R_4$ 时, 式 (2.15) 可写为:

$$U_o = -\frac{R_2}{R_1} (U_{i1} - U_{i2}) \quad (2.16)$$

$$U_- = U_+ = U_{i2} \times \frac{R_4}{R_3 + R_4} \quad (2.17)$$

上式表明输出电压与两个输入电压的差值 $U_{id} = U_{i1} - U_{i2}$ 成正比, 电路实现了差值运算, U_o 输出与 U_+ 同相, 所以要输出正电压, 应使 U_+ 大于 U_- 。 $A_f = U_o/U_{id} = -R_2/R_1$ 称为电路的差模放大倍数。而对共模信号而言, 输出电压为 0。理想情况下, 集成运放两个输入端上存在共模电压由式 (2.17) 确定。

2.4 网络的分层结构

为了减少网络设计的复杂性，大多数网络都按层（layer）或级（level）的方式来组织，每一层都建立在它的下层之上。不同的网络，其层的数量、各层的名字、内容和功能都不尽相同。然而在所有的网络中，每一层的目的都是向它的上一层提供一定的服务，而把如何实现这一服务的细节对上一层加以屏蔽。一台机器上的第 n 层与另一台机器上的第 n 层进行对话。通话的规则就是第 n 层协议（protocol）。协议基本上是通信双方关于通信如何进行达成一致的规则定义。层和协议的集合被称为网络体系结构^[9]。

2.4.1 接口和服务

接口与层的关系如图 2.4 所示，每一层的功能是为它的上层提供服务的，每一层中的活动元素常被称为实体（entity），不同机器上的同一层的实体叫做对等实体（peer entity）。 n 层实体实现的服务为 $n+1$ 层所利用， n 层被称为服务提供者（service provider）， $n+1$ 层为服务用户（service user）， n 层利用 $n-1$ 层的服务来提供它自己的服务。服务是在服务接入点 SAP（service access point）提供给上层使用的， n 层 SAP 就是 $n+1$ 层可以访问 n 层服务的地方。每个 SAP 都有一个能唯一标明它的地址。相邻层之间通过接口交换信息，在典型的接口上， $n+1$ 层通过 SAP（如图 2.4）把一个接口数据单元 IDU（interface data unit）传递给 n 层实体。IDU 由服务数据单元 SDU（service data unit）和一些控制信息组成。SDU 是即将跨过网络传递给对等实体，然后向上交给 $n+1$ 层的信息，控制信息用于帮助下一层完成任务（如 SDU 中的字节数），它本身并不是数据的一部分。为了传递 SDU， n 层实体可能将 SDU 分成几段，每一段加上一个报头后作为独立的协议数据单元 PDU（protocol data unit）送出。

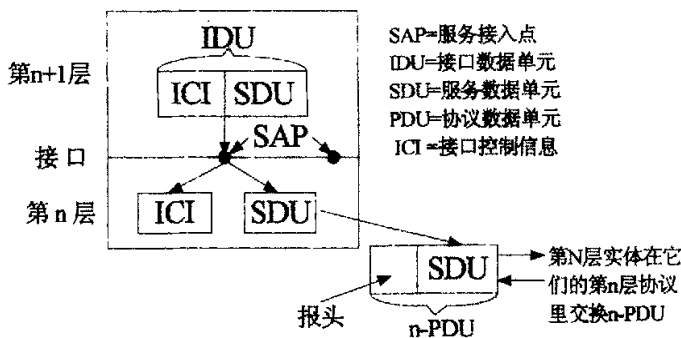


图 2.4 处于接口两边的两层之间的关系

Fig. 2.4 Relation of the two layers

2.4.2 面向连接的服务和无连接的服务

下层能向上层提供两种不同形式的服务，即面向连接的服务和无连接的服务。面向

连接的服务 (connection-oriented service) 以电话系统为模式, 在使用面向连接的服务时, 用户首先要建立连接, 使用连接, 然后释放连接。连接本质上像个管道: 发送者在管道的一端放入物体, 接收者在另一端按同样的次序取出物体。相反, 无连接服务 (connectionless service) 以邮政系统为模式。每个报文 (信件) 带有完整的目的地址, 并且每一个报文都独立于其他报文, 经由系统选定的路线传递。正常情况下, 两个报文同发往同一目的地时, 先发的先收到, 但是也有可能先发的报文在途中延误了, 后发的报文反而会先收到, 而这种情况在面向连接的服务中是绝对不可能发生的。不可靠的 (即无确认) 无连接的服务通常被称作数据报服务 (datagram service)。电报服务与此相似, 它也不向发送者发回确认信息; 在另一种情况下, 为了发一个短报文, 既希望免除建立连接的麻烦, 又要求确保信息可靠时, 可以选用有确认的数据报服务 (acknowledged datagram service), 这很像寄出的一封挂号信又要求回执一样, 当收到回执时, 寄信人有绝对的把握信件已达到目的地而没有丢失在途中; 还有一种服务, 即问答服务 (request-reply service), 使用这种服务时, 发送者传送一个询问数据报, 应答数据报则包含回答。问答服务通常被用于实现客户-服务器模式下的通信: 客户发出一个请求, 服务器做出响应。表 2.1 总结了上述服务的各种形式。

表 2.1 六种不同类型的服务

Table 2.1 Six different services

连接服务	服务类型	应用例子
面向连接的服务	可靠的消息流	页码序列
	可靠的字节流	远程登录
	不可靠的连接	数字化的声音
无连接的服务	不可靠的数据报	电子方式的函件
	有确认的数据报	挂号邮件
	问答	数据查询

2.4.3 服务原语

服务在形式上是由一组原语 (primitive) (或操作) 来描述的, 这些原语供用户和其他实体访问该服务, 这些原语通知服务提供者采取某些行动或报告某个对等实体的活动, 它的分类如表 2.2 所示:

表 2.2 四类服务原语

Table 2.2 Four different primitives

原语	含 义
请求 (request)	一个实体希望得到完成某些操作的服务
指示 (indication)	通知一个实体, 有某个事件发生
响应 (response)	一个实体希望响应一个事件
证实 (confirm)	返回对先前请求的响应

原语可以带参数, 并且大多数原语都带参数。连接请求的参数可能指明要与哪台机器连接、需要的服务类别和拟在该连接上使用的最大报文长度。连接指示原语的参数可

能包含呼叫者标志、需要的服务类别和建议的最大报文长度。如果被呼叫实体不同意呼叫实体所建议的最大报文长度，它可能在响应原语中作出一个反建议，呼叫方可从证实原语中获知它。这一协商（negotiation）的细节是协议的内容。例如：在两个建议的最大报文长度不一致的情况下，协议可能规定选择较小的值。

服务有“有证实（confirmed）”和“无证实（unconfirmed）”之分。有证实服务包括请求、指示、响应和证实 4 个原语。而无证实服务则只有请求和指示 2 个原语。CONNECT 服务总是有证实的服务，因为远程对等实体必须同意才能建立连接。在另一方面，数据传输要么是有证实的要么是无证实的，这取决于发送方是否要求确认。这 2 种服务都可以在网络中使用。

2.4.4 服务与协议的关系

服务和协议是完全不同的概念，但二者常常被混淆在一起。服务是各层向它上层提供的一组原语（操作），它定义了两层之间的接口，接口告诉上面的进程如何访问它，它定义需要什么参数以及预期结果是什么样的，如同程序设计语言中的抽象数据类型，它和该层如何工作无关它；与之相对比，协议是定义同层对等实体之间交换的帧、分组和报文的格式及意义的一组规则。实体利用协议来实现它们的服务定义。只要不改变提供给用户的服务，实体可以任意的改变它们的协议。这样，服务和协议就被完全分离开来。总之，服务定义该层做些什么，而不管上面的层如何访问它或该层如何工作；某一层中使用的对等协议是该层的内部事务，它可以使用任何协议，只要能完成工作（例如提供承诺的服务），也可以改变使用的协议而不会影响到它上面的层。

2.5 OSI 参考模型

OSI 参考模型如图 2.5 所示，该模型基于国际标准化组织 ISO(International Standard Organization)的建议，作为各种层上使用的协议国际化的第一步而发展起来的，它被称作 ISO OSI（Open System Interconnection）开放系统互联参考模型。因为它是关于如何把开放式系统（即为了与其他系统通信而相互开放的系统）连接起来的，所以常被简称为 OSI 模型。它提供的是概念性和功能性的七层结构，而不是互联结构的设施和协议细节的精确定义。它的低层协议包括三层：物理层、数据链路层、网络层，它们实现的是 OSI 系统中面向通信的功能。

(1)物理层

物理层（physical layer）主要定义了物理线路的机械、电气和电信号特征及功能。机械特征包括接口连接器件的尺寸、插针数目和每个插针功能的分配等内容；电气特征包括最大允许数据传输速率、最大传输距离、每种信号电平可处状态所代表的意义和连接器可承受的最大电流、电压等；电信号功能包括每种信号的逻辑信号、各种信号的传输时序、数据采样方式等。物理层的交互有一整套握手协议。它主要涉及到通信在信道

上传输的比特流。

(2)数据链路层

数据链路层（data link layer）处理两个由物理通道直接相连的邻接站之间的通信。数据链路层协议的目的在于提高数据传输的效率，为其上层提供透明的无差错的通道服务。把传输媒体的不可靠因素尽可能地屏蔽起来，让高层协议免于考虑物理介质的可靠性问题，而把通道看作无差错的理想通道。它的主要任务是加强物理层传输原始比特的功能，使之对网络层显现为一条无错线路，其任务具体表现在以下几个方面：

- ①数据链路的建立和拆除，包括同步、站地址确认、收发关系的确认、最终一次传输的表示等。
- ②信息传输，包括信息格式、数量、顺序编号、接收认可、信息流量调节方案等。
- ③传输差错控制，包括一套防止信息丢失、重复和失序的方法。
- ④异常情况处理，包括如何发现可能出现的异常情况及发现后的处理过程。协议中对异常情况处理主要用于发现和恢复永久性故障。

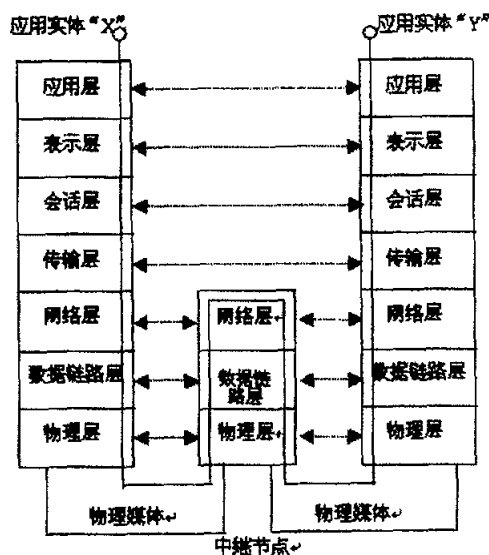


图 2.5 OSI 网络参考模型

Fig. 2.5 OSI network reference model

(3)网络层

网络层（network layer）关系到子网的运行控制，其中一个关键问题是确定分组从源端到目的端如何选择路由。该层主要完成信息传送时的路由选择、差错校验、流量控制等，在局域网中由于只有一个子网，所以此层很弱，甚至不存在。

(4)传输层

传输层（transport layer）的基本功能是从会话层接收数据，并在必要时把它分成较小的单元，传递给网络层，并确保到达对方的各段信息高效率且正确无误。该层是真正

的从源到目标“端到端”的层，而其以下的各层中，协议是每台机器和它相邻的机器间的协议。

(5)会话层

会话层（session layer）允许不同机器上的用户建立会话（session）关系。其服务之一是管理对话。一种与会话有关的服务是令牌管理（token management），另一种会话服务是同步（synchronization）。

(6)表示层

表示层（presentation layer）完成某些特定的功能，因为这些功能常被请求，从而不必让每个用户来实现。值得一提的是，表示层以下的各层只关心可靠地传输比特流，而表示层关心的是所传输的信息的语法和语义。如数据编码服务。

(7)应用层

应用层（application layer）包含大量人们普遍需要的协议。它实现的功能分为两大部分，即用户应用进程和系统应用管理进程。用户应用进程由用户要求决定，系统应用管理进程管理系统资源，如优化分配系统资源和控制资源的使用等。

表 2.3 OSI 模型各层功能

Table 2.3 Function of OSI layers

7	应用层	最高层。用户、软件、网络终端等之间用来进行信息交换
6	表示层	将两个应用不同数据格式的系统信息转化为能共同理解的格式
5	会话层	依靠低层的通信功能来进行数据的有效传递
4	传输层	两通讯节点之间数据传输控制。操作如：数据重发，数据错误修复
3	网络层	规定了网络连接的建立、维持和拆除的协议。如：路由和寻址
2	数据链路层	规定了在介质上传输的数据位的排列和组织。如：数据校验和帧结构
1	物理层	规定通讯介质的物理特性。如：电气特性和信号交换的解释

各层功能总结如表 2.3。OSI 模型的数据传输，信息从最上层开始，每层添加相应的数据报头，然后送至下一层，直至物理层，然后被实际传输到接收机。在接收机里，信息向上传递时，各种报头被一层一层的剥去，最后信息到达接收进程。整个过程虽然数据的实际传输方向是垂直的，但每一层在编程时却好象数据一直是水平传输的，每一层与另一设备上相同的那一层通讯，实际的通讯是发生在每一设备上相邻的两层，而设备只通过模型物理层的物理介质互连。

第三章 硬件设计

3.1 仪表总体结构

本系统主要由以下模块构成：荷重传感器；弱信号放大电路；测量仪；Com 通讯软件及 CAN 通讯软件。荷重传感器主要用于把压力（重力）线性转换成电阻的变化量，通过补偿运算电路供给激励电源输出线性的电压变化量值；弱信号放大电路对传感器输出的小电压信号进行放大滤波，使其从微小信号转换成至测量仪可检测的信号范围内；测量仪进行模数转换、键盘控制、数字显示及数值计算；Com 通讯及 CAN 通讯分别满足与计算机的串口通讯及总线式通讯功能。

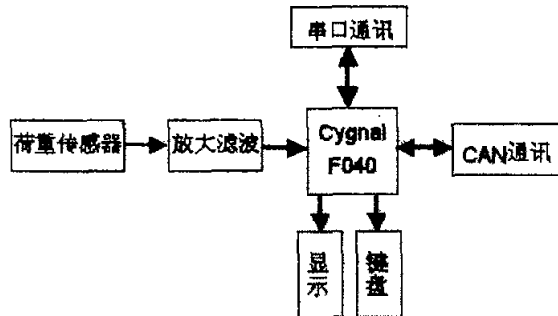


图 3.1 测量系统框图

Fig. 3.1 Measuring system chart

测量系统总体结构如图 3.1 所示，整个设计以 Cygnal 8051F040 为核心，荷重传感器及放大电路组成了重量测量线路，键盘、数码管显示，构成人机接口；串口通讯实现了查看测量数据及修改测量参数；CAN 通讯实现 CANopen 网络通信功能。

3.2 测量电路系统

3.2.1 Cygnal 8051F040 单片机介绍

Cygnal 8051F040 单片机^[10]是美国 Cygnal 公司的产品，使用 Cygnal 的专利 CIP-51 微控制器内核，采用 100 脚 TQFP 封装，是完全集成的混合信号系统级芯片 SOC(System On Chip) 单片机。与其它 8 位单片机相比具有更为优异的性能，更小的体积，其结构及封装如图 3.2 所示。其主要特点如下：

(1) CIP-51™ 微控制器核

当前市场上的主流单片机主要有两种，一种是发展的比较成熟的 MCS-51 系列的 CISC (Complex Instruction Set Computer) 复杂指令系统计算机，另一种就是基于 Arm 核心的 RISC (Reduced Instruction Set Computer) 精简指令系统计算机^[11]。Arm^[12]是

Advanced RISC Machine 的缩写，它实际上一家公司的名称，该公司只研发 RISC 系统的架构，而进行对其它公司授权生产。RISC 对应于 CISC，主要是对 CISC 做了一些优化，剔除了其中一些不常用的指令（主要是通过统计发现，有一些很复杂的指令很少用），而采用简单的指令，简化硬件的设计而提高其并发执行能力，从而在同频的情况下性能得到大幅提升。CIP-51 微控制器内核与 MCS-51TM 指令集完全兼容，可以使用标准 803x/805x 的汇编器和编译器进行软件开发。但由于其采用了 RISC 处理器中常用的流水线结构，所以其指令执行速度有很大的提高。在一个标准的 8051 中，除了 MUL 和 DIV 以外所有指令都需要 12 或 24 个系统时钟周期，最大系统时钟频率为 12~24MHz。而对于 CIP-51 内核，它 70% 的指令的执行时间为一个或两个系统时钟周期，只有 4 条指令的执行时间大于 4 个系统时钟周期。当它工作在最大系统时钟频率 25MHz 时，其峰值性能达到 25MIPS。

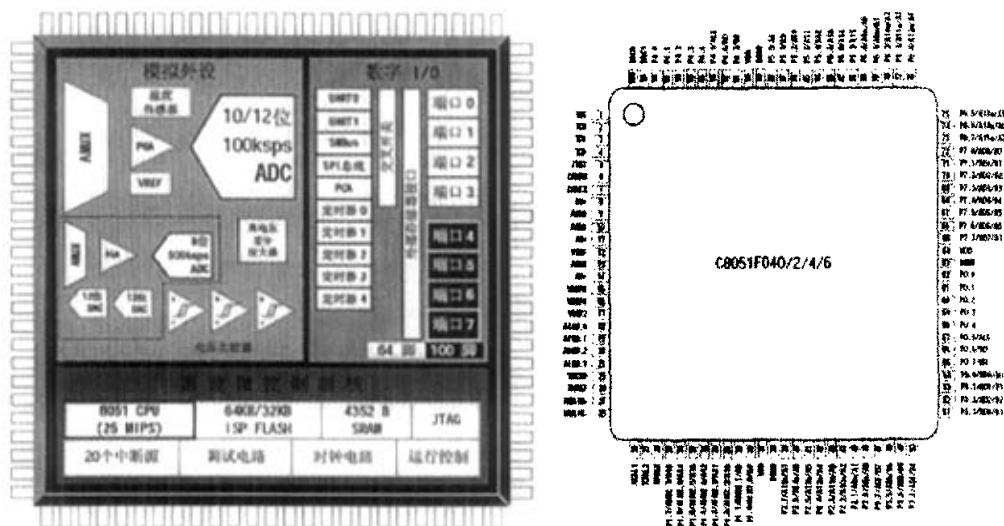


图 3.2 Cygnal 8051F040 单片机的结构图及 TQFP-100 引脚图封装形式

Fig. 3.2 Structure and encapsulation form TQFP-100 of Cygnal C8051F040

CIP-51 内核具有标准 8052 的所有外设部件，包括 5 个 16 位的计数器/定时器、两个全双工 UART、256 字节内部 RAM、128 字节特殊功能寄存器（SFR）地址空间及 8 个 8 位宽的 I/O 端口。除此外，它还集成了数据采集和控制系统中常用的模拟部件和其它数字外设及功能部件，以及支持 CAN 工业网的控制器。扩展的中断系统提供 20 个中断源（标准 8051 只有 7 个中断源），允许大量的模拟和数字外设中断微控制器。一个中断驱动的系统需要较少的 MCU 干预，因而有更高的执行效率。在设计一个多任务实时系统时，这些增加的中断源是非常有用的。

MCU 可有多达 7 个复位源：一个片内 VDD 监视器、一个看门狗定时器、一个时钟丢失检测器、一个由比较器 0 提供的电压检测器、一个软件强制复位、CNVSTR0 输入

引脚及/RST 引脚；内部有一个独立运行的时钟发生器，在复位后被默认为系统时钟。如果需要，时钟源可以在运行时切换到外部振荡器，外部振荡器可以使用晶体、陶瓷谐振器、电容、RC 或外部时钟源产生系统时钟。

(2)片内存储器

CIP-51 有标准的 8051 程序和数据地址配置。它包括 256 字节的数据 RAM，其中高 128 字节为双映射。用间接寻址访问通用 RAM 的高 128 字节，用直接寻址访问 128 字节的 SFR 地址空间。CIP-51 的 SFR 地址空间可包含多达 256 个 SFR 页。通过 SFR 分页，CIP-51 MCU 可以控制大量用于控制和配置片内外设所需要的 SFR。数据 RAM 的低 128 字节可用直接或间接寻址方式访问。前 32 个字节为 4 个通用寄存器区，接下来的 16 字节既可以按字节寻址也可以按位寻址。图 3.3 给出了 MCU 系统的存储器结构。

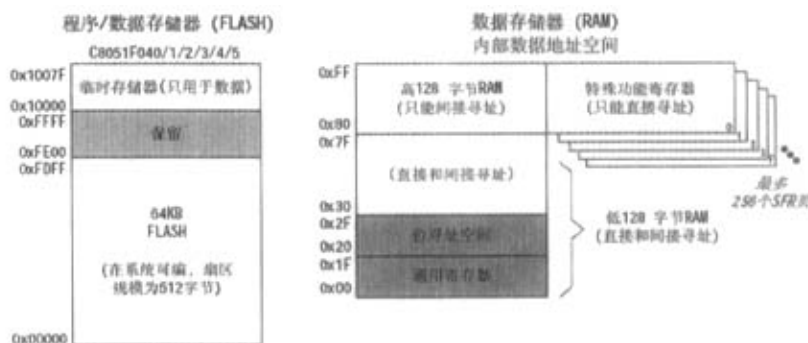


图 3.3 片内存储器组织

Fig. 3.3 On-Chip memory map

C8051F040 中的 CIP-51 还另有位于外部数据存储器地址空间的 4KB 的 RAM 块和一个可用于访问外部数据存储器的外部存储器接口 (EMIF)。这个片内的 4KB RAM 块可以在整个 64K 外部数据存储器地址空间中被寻址（以 4K 为边界重叠）。外部数据存储器地址空间可以只映射到片内存储器、只映射到片外存储器、或两者的组合（4KB 以下的地址指向片内，4KB 以上的地址指向 EMIF）。EMIF 可以被配置为地址/数据线复用方式或非复用方式。程序存储器包含 64KB 的分块 FLASH，该存储器以 512 字节为一个扇区，可以在系统编程，且不需特别的外部编程电压。从 0xFE00 到 0xFFFF 的 512 字节被保留。还有一个位于地址 0x10000—0x1007F 的 128 字节扇区，该扇区可用于存储小规模软件常数表。

(3)JTAG 调试和边界扫描

C8051F040 具有片内 JTAG 边界扫描和调试电路，通过 4 脚 JTAG 接口并使用安装在最终应用系统中的产品器件就可以进行非侵入式（不占用片内资源）、全速的在系统调试。该 JTAG 接口完全符合 IEEE1149.1 规范，为生产和测试提供完全的边界扫描功能。Silicon Lab 的调试系统支持观察和修改存储器和寄存器，支持断点、观察点、堆栈

指示器和单步执行。不需要额外的目标 RAM、程序存储器、定时器或通信通道。在调试时所有的模拟和数字外设都正常工作。当 MCU 单步执行或遇到断点而停止运行时，所有的外设（ADC 和 SMBus 除外）都停止运行，以保持同步。这使得开发更简单方便。

(4) 可编程数字 I/O 和交叉开关

C8051F040 具有标准 8051 的端口（0、1、2 和 3），同时有 4 个附加的 8 位端口（4、5、6 和 7），因此共有 64 个通用 I/O 端口。这些 I/O 端口的工作情况与标准 8051 相似，但有一些改进。每个端口引脚都可以被配置为推挽或漏极开路输出。在标准 8051 中固定的“弱上拉”可以被总体禁止，这为低功耗应用提供了进一步节电的能力。但最独特的改进是引入了数字交叉开关。这是一个大的数字开关网络，允许将内部数字系统资源映射到 P0、P1、P2 和 P3 的端口 I/O 引脚。与具有标准复用数字 I/O 的微控制器不同，这种结构可支持所有的功能组合。可通过设置交叉开关控制寄存器将片内的计数器/定时器、串行总线、硬件中断、ADC 转换启动输入、比较器输出以及微控制器内部的其它数字信号配置为出现在端口 I/O 引脚。这一特性允许用户根据自己的特定应用选择通用端口 I/O 和所需数字资源的组合。

(5) 串行端口和控制器局域网

C8051F040 内部有两个增强型全双工 UART、一个增强型 SPI 总线和 SMBus/I²C。每种串行总线都完全用硬件实现，都能向 CIP-51 产生中断，因此需要很少的 CPU 干预。这些串行总线不“共享”定时器、中断或端口 I/O 等资源，所以可以使用任何一个或同时使用多个。C8051F040 内部还有一个控制器局域网（CAN）控制器，它的集成简化了用 CAN 协议实现通信时外围电路的复杂度，其使用在第五章进行详细描述。

(6) 12 位模/数转换器

C8051F040 片内有一个 12 位 SAR ADC（ADC0），一个 9 通道输入多路选择开关和可编程增益放大器。该 ADC 工作在 100ksps 的最大采样速率时可提供真正的 12 位精度，INL 为 $\pm 1\text{LSB}$ 。ADC0 的电压基准有其专用的 VREF0 输入引脚，片内 15ppm/°C 的电压基准可通过 VREF 输出引脚为片内 ADC 或其它系统部件产生基准电压。

ADC 完全由 CIP-51 通过特殊功能寄存器控制。一个输入通道被连到内部温度传感器，其它 8 个通道接外部输入。8 个外部输入通道的每一对都可被配置为两个单端输入或一个差分输入。系统控制器可以将 ADC 置于关断状态以节省功耗。可编程增益放大器接在模拟多路选择器之后，增益可以用软件设置为 0.5、1、2、4、8 或 16。当不同 ADC 输入通道之间输入的电压信号范围差距较大或需要放大一个具有较大直流偏移的信号时（在差分方式，DAC 可用于提供直流偏移），这个放大环节是非常有用的。A/D 转换有 4 种启动方式：软件命令、定时器 2 溢出、定时器 3 溢出和外部信号输入。这种灵活性允许用软件事件、外部硬件信号或周期性的定时器溢出信号触发转换。转换结束由一个状态位指示，或者产生中断（如果中断被使能）。在转换完成后，12 位转换结果数据字被锁存到两个特殊功能寄存器中。这些数据字可以用软件控制为左对齐或右对

齐。窗口比较寄存器可被配置为当 ADC 数据位于一个规定的范围之内或之外时向控制器申请中断。ADC 可以用后台方式连续监视一个关键电压，当转换数据位于规定的窗口之内时才中断控制器。

(7)比较器和 DAC

C8051F040MCU 内部有两个 12 位 DAC 和两个比较器。MCU 与每个比较器和 DAC 之间的数据和控制接口通过特殊功能寄存器实现。MCU 可以将任何一个 DAC 或比较器置于低功耗关断方式。比较器的回差电压可以用软件编程。可以通过调整比较器的响应时间使功耗最小或使速度最快。每个比较器都能在上升沿、下降沿或在两个边沿都产生中断。这些中断能将 MCU 从休眠方式唤醒。比较器的输出状态可以用软件查询。可通过设置交叉开关将比较器的输出接到端口 I/O 引脚。DAC 为电压输出方式，有灵活的输出更新机制。这一机制允许用软件写和定时器 2、定时器 3 及定时器 4 的溢出信号更新 DAC 输出，其电压基准由专用的 VREFD 输入引脚提供。DAC 在作为比较器的参考电压或为 ADC 差分输入提供偏移电压时非常有用。

3.2.2 荷重传感器

荷重传感器种类很多，在选型时可参考线性度、灵敏度、精确度、最小检测量和分辨率、零点漂移、温度漂移，以及迟滞，重复性等特性。本系统中的荷重传感器采用的是浙江余姚 Pacific 公司的 CZL-YB-6 型的电阻应变式称重传感器，CZL-YB-6 是采用国家 GB7666-87《传感器命名法及代号》的标准命名的，C 是主称：传感器；ZL 是被测量：重量（称重）；YB 指转换原理：应变式；最后一位数字序号：用阿拉伯数字表示设计序号。该传感器采用双孔悬臂梁结构形式，其偏心及侧向力影响小，动态性能好，精度高，同时外形低，安装方便。其精度等级为 0.05，输入/输出阻抗分别为 $380\ \Omega/350\ \Omega$ ，灵敏度为 2mV/V ，额定载荷为 100Kg。

传感器使用过程中为保证良好的效果，应注意：(1)设计加载装置及安装时，应保证使用加载力的作用线与传感器受力轴线重合，使倾斜负荷和偏心负荷的影响减到最小；(2)使用过程中建议载荷不超过额定载荷的 80%；(3)引线识别，输入(+)：红；输入(-)：黄；输出(+)：蓝；输出(-)：白。

整个传感器系统使用了三台 CZL-YB-6，安装各台传感器使其尽可能的共在一个水平面上，则其各个传感器向上的合力即是重物的重力（牛顿定律）。各台传感器再接至 BJH-1 补偿接线盒，通过 BJH-1 调节其信号灵敏度一致性，进行信号的和算。从整个过程来看，电阻线性反应载荷重量，输出电压信号线性反应电阻变化值，从而输出电压信号线性反应载荷重量。

3.2.3 BJH-1 补偿线盒

本系统在称重传感器和称重仪表之间使用 BJH-1 补偿接线盒^[13]连接，该补偿装置的

主要功能有：一是称重仪表通过该补偿线盒向系统中的各个称重传感器提供桥路激励电源；二是将系统中各个称重传感器的输出信号和算后，汇成一路信号提供给称重仪表；三是对系统中各个称重传感器的输出信号具有一定的调整作用，使各个传感器的输出灵敏度一致。BJH-1 补偿接线盒采用了 3mm 冷轧板制造，坚固耐用，特别适用于环境恶劣的工业现场，同时可连接多达 4 台传感器。其接线方法较为简单，桥压 (+) / (-) 端接传感器的输入 (+) / (-)，输出 (+) / (-) 即为进行并联和算后的信号。（注：也可用万用表 $2K\Omega$ 电阻档测量传感器的四根引线，其中电阻最大的一对为桥压输入端，剩下一对即为信号输出端。按以上的接法进行连接后对系统加电，然后在每个传感器上进行加载试验，若仪表示值变大，则说明接线正确，反之则说明极性反了，应将两根桥压线或信号线对调。）其工作原理如图 3.4 所示：

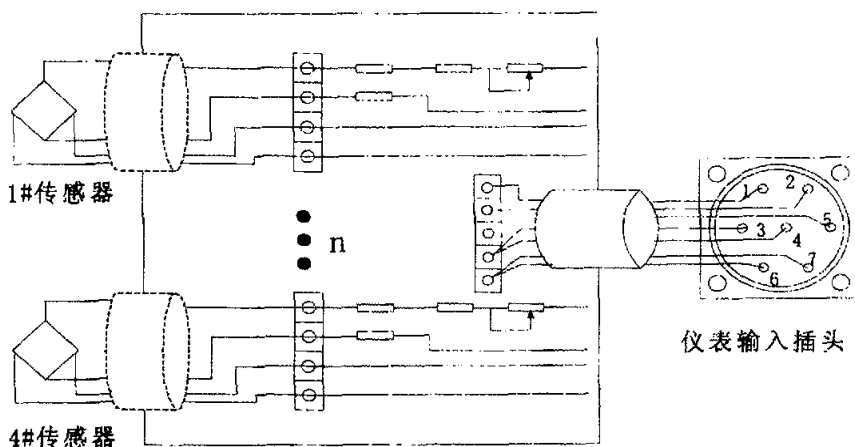


图 3.4 补偿线盒原理图

Fig. 3.4 Compensate block diagram

注：1 +SIG 传感器信号 (+)；2 -SIG 传感器信号 (-)；3 +EXC 激励电源 (+)；

4 -SEN 激励电源反馈 (-)；5 屏蔽 屏蔽地；6 +SEN 激励电源反馈 (+)；

7 -EXC 激励电源 (-)；另：使用 4 芯屏蔽电缆时，+EXC 与 +SEN、-SEN 与 -SEN 短接。

传感器的供桥电源在接线盒内并联。传感器输出信号两端之间跨接有各自独立的阻抗调节电阻和电位器，通过调节灵敏度系数和其阻抗之比实现传感器的输出信号灵敏度 $(mV/V)/R_{out}$ 一致性，从而每个传感器的输出信号接近完全相同，输出信号每端再通过隔离电阻器并联，实现传感器信号并联结构的和算，此时传感器的输出电压范围与单路输出没有差别，但桥路的输出电阻变小，输出电流增大，驱动负载能力增强，总体的线性度也得到改善。信号的和算也可采用串联结构，此时需要三路独立的桥压源供电，使仪器变的复杂，所以在本设计中采用传感器并联连接方式。

3.2.4 高精密参考电压源

从桥路的灵敏度分析可知，输入激励电源稳定性与灵敏度密切相关，本系统使用

AD584 来获得高精度的基准电压源。集成基准源按其工作原理大致可分为两大类，一类是齐纳型基准源^[14]，另一类是带隙型基准源。前者是以具有温度补偿的 Zener 二极管为基础，采用多片混合电路设计，因而较为复杂，同时由于 Zener 击穿发生在 6.3V 左右，使得在较低输入电压时，基准源不能工作。此外，它需要多个电源供电，给系统配置及电路布线带来了一定的困难。带隙型基准源是利用在正偏工作区硅三极管基射结电压中隐含的物理特征制成的。实验证明，所有的硅三极管都存在一个近似为 $-2\text{mV}/^\circ\text{C}$ 的温度系数，不适宜直接用来做低温漂的基准源。然而，只要延伸这些器件的温度特性曲线到绝对零度，可以看到它们均相交于一点且 $V_{BE}=1.205\text{V}$ 。因此，如果能够设计一种电路，使之能产生与上述三极管相反的温度系数且使二者之和为 1.205V 的电压，那就就可以产生零温度系数的基准电压。AD584 就是利用两个三极管的基射集能带差（即带隙）所产生的压差，并放大至 2.5V 提供 1 个稳定可靠的基准源。这两个三极管工作的电流密度不同，由于压差很低，因此基准源能在单一的低电压电源下工作。

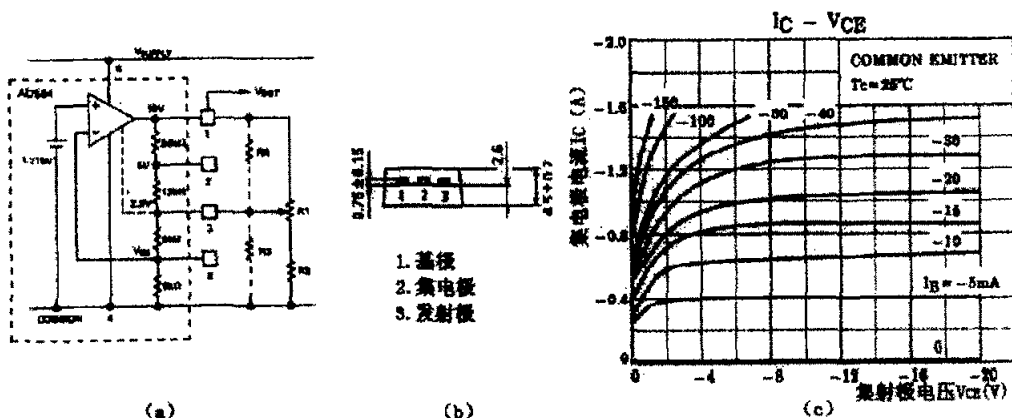


图 3.5 AD584 工作原理及 A940 引脚图和 I_C-V_{CE} 工作曲线

Fig. 3.5 AD584 schematic circuit and pins of AD949 and I_C-V_{CE} relationship

AD584 可被看作 1 个同相输出的运算放大器，如图 3.5 (a) 所示，其输入是芯片内高精度高稳定性的 1.215V 带隙基准电压，若外加反馈调整电阻，可扩大其输出电压范围，如获得其它通用基准电压 10.24V ， 5.12V ， 2.56V 及 6.3V 等。最常使用的办法是分别用 R_1 和 R_2 来调整最大范围和最小分辨率。当调整 R_1 到顶端时， 2.5V 和输出引脚相连，使输出电压变为 2.5V ；当调整 R_1 到底端时，且设 $R_1 \gg R_2$ ，则输出电压上升到由 R_2 限定的值，若 $R_2=6\text{K}\Omega$ ，则输出电压上限大约为 20V （此时总电流扩大两倍，从而电压增一倍）。当调整 R_1 在其中间时，可得到 2.5V 到上限值之间的任何值。在信号和电源调节电路中，AD584 还被用作过流过压保护，因为它有一选通端 Strobe，通过一定的连接，能使输出电压变为 0V 。

电压基准源常用的参数指标有，电源调节率：给定输入电压增量引起输入电压的变化量，单位为 mV/V ；负荷调节率：给定负荷电流增量引起输出电压的变化量，它包括

负荷电流较大使功耗增加时所产生的自热现象引起的输出电压的变化量，单位为 mV/mA ；输出电压误差：当温度为 25°C 时，标称输出电压与给定输入电压下实测输出电压之差，单位为 mV ；噪音：出现在基准源输出端的窄带（噪音 $0.1\text{--}10\text{Hz}$ ）随机噪声，单位为 $\mu\text{V}_{\text{p-p}}$ 。AD584 是具有温度补偿的基准源，当输入电压在 $12\text{--}30\text{V}$ 之间时，不需要任何输入电平和外加元件就能提供高精度 10.000V 基准电压。采用的激光薄片工艺使得 25°C 时的原始误差仅为 5mV ，而温度系数仅为 $5\text{ppm}/^{\circ}\text{C}$ ，优于混合电路设计和恒温调节模式所能达到的精度。

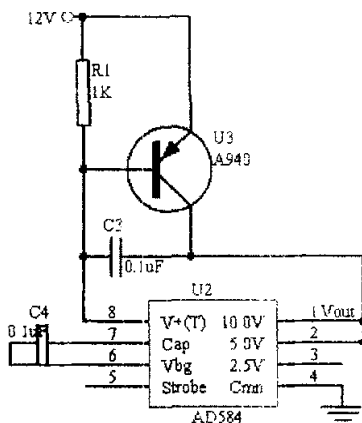


图 3.6 功率放大及精密电源电路

Fig. 3.6 Power amplifier application and high precisoin output circuit

AD584 的输出电流最大只有 10mA ，很难满足做为精密电源时的功率要求，但 AD584 容易和大功率 PNP 管或大功率达林顿 PNP 管连接，提供高精度大电流电压源。本系统使用 A940 与 AD584 连接，电路连接如图 3.6，以进行电流放大，增加带载能力，满足功率的需求。A940 集电极最大可提供 1.5A 的负载电流，消耗功率最大可达 25W 。在图 3.5(b) 为 A940 的正视图中，从左至右分别为其 P1：基极 (base)；P2：集电极 (collector)；P3：发射极 (emitter)。图 3.5 (c) 为其 I_{c} 与 V_{CE} 的工作曲线，从中可看出，当 V_{CE} 为 -4V 时，设计 $I_{\text{B}}=5\text{mA}$ 时，可获得 0.4A 的稳定 I_{c} 电流。本系统中要求电流输出 200mA 就已足够，分析图 3.6 电路模块，输入采用 12V 电压供电，输出为 7.5V 精密电源，减除 V_{BC} 大约 0.7V 的压差，则落在 $R1$ 上的压降约为 3.8V ，从而得 I_{B} 的电流约为 4mA ，其输出电流在 300mA 以上，可以满足本系统的要求。

3.2.5 ICL7650 集成运放

ICL7650 是 Intersil 公司利用动态校零技术和 CMOS 工艺制作的斩波稳零式高精度运放，它具有输入偏置电流小、失调小、增益高、共模抑制能力强、响应快、漂移低、性能稳定及价格低廉等优点，其主要特点^[15]是超低失调电压： $5\mu\text{V max}$ ；超低漂移： $0.02\mu\text{V}/^{\circ}\text{C}$ ，超低输入漂流： 20pA max ，高增益： 135dB min ，高共模抑制比： 120dB min ，

高输入阻抗: $10^{12} \Omega$ 。

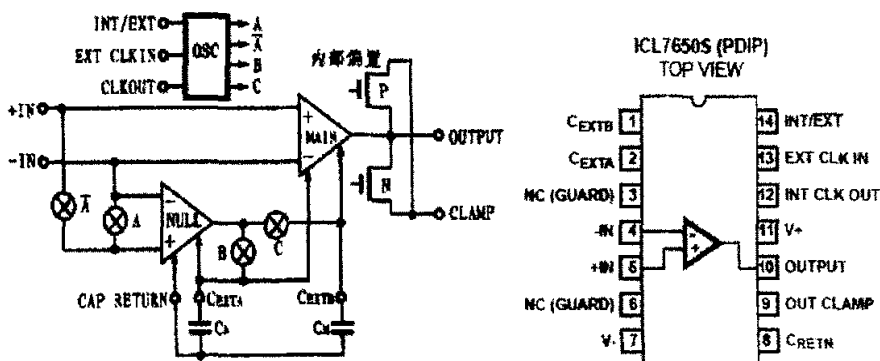


图 3.7 ICL7650 的工作原理及封装

Fig. 3.7 Schematic circuit and pin picture of ICL7650

ICL7650 的动态校零技术消除了 CMOS 器件固有的失调和漂移,从而摆脱了传统稳零电路的束缚,克服了传统稳零放大器的这些缺点。其工作原理^[16]如图 3.7 左侧图所示。图中,MAIN 是主放大器(CMOS 运算放大器),NULL 是调零放大器(CMOS 高增益运算放大器)。电路通过电子开关的转换来进行两个阶段工作,第一是在内部时钟(OSC)的上半周期,电子开关 A 和 B 导通, \hat{A} 和 C 断开,电路处于误差检测和寄存阶段;第二是在内部时钟的下半周期,电子开关 \hat{A} 和 C 导通, A 和 B 断开,电路处于动态校零和放大阶段。由于 ICL7650 中的 NULL 运算放大器的增益 A_{IN} 一般设计在 100dB 左右,因此,即使主运放 MAIN 的失调电压 V_{OSN} 达到 100mV,整个电路的失调电压也仅为 $1 \mu V$ 。由于以上两个阶段不断交替进行,电容 C_N 和 C_M 将各自所寄存的上一阶段结果送入运放 MAIN、NULL 的调零端,这使得图 3.7 左侧图所示电路几乎不存在失调和漂移。但 ICL7650 作为一种斩波稳零的运放,其斩波频率低(200Hz),输出信号中含有斩波尖峰噪声,因此反馈回路中要有滤波电路以滤除斩波噪声。

ICL7650 采用 14(8)脚双列直插式和 8 脚金属壳(圆形)两种封装形式,前者在市场上易于购到,其引脚排列如图 3.7 右侧图所示,各引脚功能说明如下:

-IN, +IN 分别为反相/同相输入端; V-, V+ 分别为负/正电源供电端; OUTPUT 输出端; INT CLKOUT, EXT CLKIN 分别为时钟输出端/时钟输入端; INT/EXT 为时钟控制端,可通过该端选择使用内部时钟或外部时钟,当选择外部时钟时,该端接负电源端 (V-),并在时钟输入端 (EXT CLKIN) 引入外部时钟信号,当该端开路或接 V+ 时,电路将使用内部时钟去控制其它电路的工作; Out clamp 箝位端,用以减少过载恢复时间,当连接同相端或反相端时,在输出端饱和之前,输出端与该端点形成电流回路。因此,与校正电容的充电作用一起避免了输入的不确定的微分信号,从而输出信号波动的减小很少; P1, P2 和 P8 电容连接 Cexta 和 Cextb 与 Cretn 相连,采用内部频率信号时,用 0.1uF 电容,作电路板时,应把 Cexta 和 Cextb 的贴片单独引至 Cretn 公共端; P3,

P6（14P）封装的 NC 引脚，与邻近的输入电压相连，自建闭环导点，用以去除邻近电源引脚等带来的输入噪声。

3.2.6 非线性补偿电路

ICL7650 常用差分式放大电路如图 3.8 所示，该电路存在着以下问题，其输入电阻不高，在 V_i 为 0—5mV 范围，放大器放大倍数线性较差^[17]（ $\Delta A_v \approx 40\%$ ），

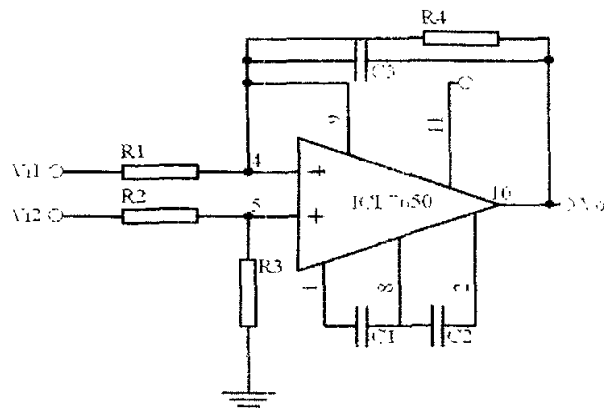


图 3.8 常用差动放大电路图

Fig. 3.8 Common differential operation circuit

而且稳定度也不高(约有±10mV 波动)。按上图电路进行实验,R1=R2=1K,R3=R4=100K,c1=c2=0.1uF。使用 200mV 量程，分辨率 100uV，得实验数据如表 3.1：

表 3.1 实验数据

Table 3.1 Experiment Data

输入 (mV)	35.1	68.1	74.3	23.4	13.1	5.3
输出 (V)	0.675	1.302	1.425	0.448	0.250	0.0916
放大倍数	19.2	19.119	19.179	19.145	19.084	17.283

分析：随着输入信号的减小，从表中读取稳定电压值也越来越困难，从而测量的误差越来越大，但从以上几组数字可以看出，放大器的放大倍数在小于 10mV 以下的时候，有明显的放大倍数非线性，其它几组数字可认为是可接受的。本系统采用补偿电路法来解决上述问题，其思想是：整个放大电路分两级，对前级 ICL7650 的输入信号进行叠加一个 10mV 左右的信号，使其工作在较大的信号输入区域，然后把叠加产生的信号在第二级差动电路中进行减去。本系统中采用电路如图 3.9 所示：前级放大电路采用一个电阻桥来产生可调的叠加输入信号，后级仍然采用差动放大电路实现减法运算，被减数值由精密电源通过一个 10K，一个 1K 精密电阻及一个 1k 变位器分压产生（ $U -$ 在 $U_j \times 1/12$ 与 $U_j \times 2/12$ 之间），把输入端短接，则叠加信号产生前级放大电路的输出，这样可以通过调整电阻桥桥臂电阻，使总输出为 0，从而实现了前级叠加输入信号，后级减除叠

加信号影响的功能。

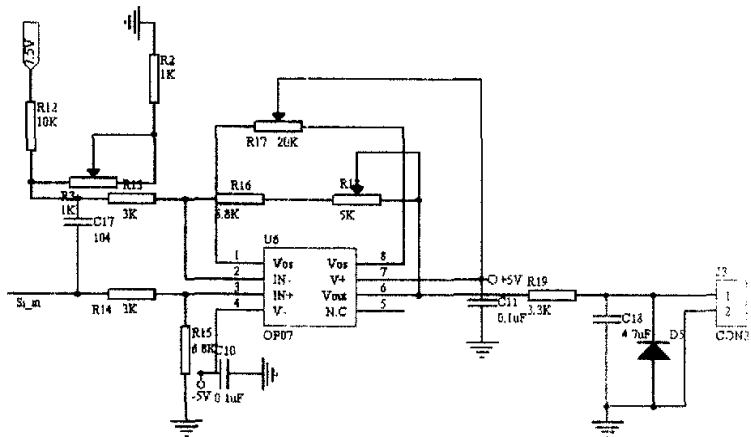


图 3.9 非线性补偿放大电路

Fig. 3.9 Nonlinear compensate amplifier circuit

3.2.7 键盘及显示电路

C8051F040 的 I/O 较多，而键盘数目又较少（6 个），所以直接使用 I/O 口外接按键组成键盘。本系统中使用 P20、P33—P37 作为键盘输入 I/O 口（通过配置端口为“漏极开路”，并向其相应端口口数据寄存器写‘1’，从而配置引脚为数字输入方式），通过读相应的端口为 0 确认对应按键按下。

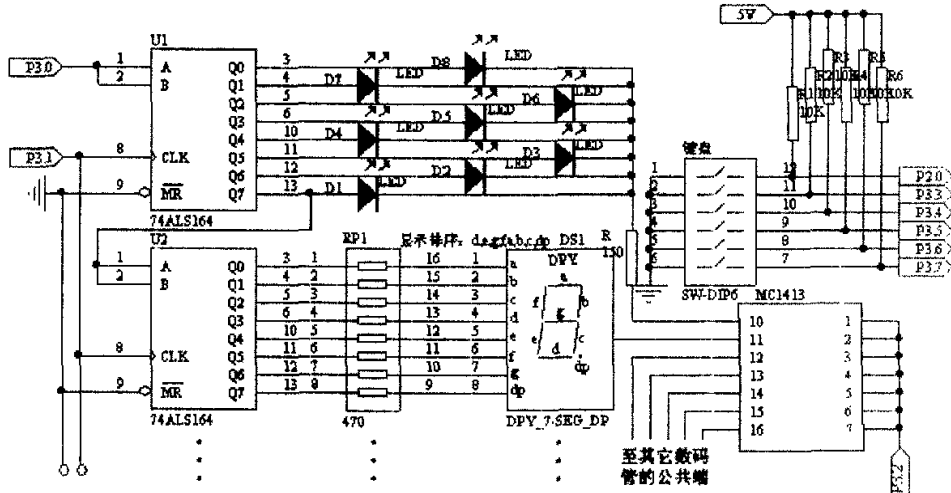


图 3.10 键盘显示（部分）电路图

Fig. 3.10 Keyboard and display circuit(part)

显示电路由 7 个 74LS164、8 个发光二极管、6 个数码管和一片 MC1413 组成。显示部分采用静态显示，占用三根 I/O 口线，其中 P30 作为数据输出，P31 作为时钟信号，P32 作为使能控制。第一片 74LS164 控制 8 个发光二极管，后面的 6 片 74LS164 驱动 6

个数码管。每个单元 74LS164 的 QH 与下一个单元 74LS164 的数据输入端相连。这样，每个单元对应的显示码可以从 P30 串行输出，至相应显示单元。LED 的位选线由 MC1413 的输出控制。该部分电路如图 3.10，其工作原理是：每一次改变显示内容之前，需要输入显示码至 74ALS164。显示码的输入为串行输入，即每一次输入一位显示码，6 位 LED 及 8 个发光二极管需要输入 56 位显示码。在输入显示码之前，先使 P32 为低电平，全部 LED 熄灭，防止输送显示码的过程中产生闪烁。74LS164 是下降沿锁存，在每送出一位显示码之后，P31 输出下降沿锁存数据。完成显示码输出之后，使 P32 为高电平，各数码管及 LED 处于工作显示状态。

3.2.8 串行口通讯电路

串行通讯接口标准目前已有多种，不过都是在 RS-232C 的基础上改进而形成的。RS-232C^[18]标准是美国 EIA（电子工业联合会）与 BELL 公司一起开发并于 1969 年公布的通讯协议，是目前最常用的串行接口标准，用于计算机与计算机之间，计算机与外设之间的数据通信。该标准的目的是定义数据终端设备（DTE）之间的电气特性，提供了单片机与单片机、单片机与 PC 机之间串行数据通信的标准接口，通信距离可达 15m。RS232 标准是按负逻辑定义的，它的‘1’电平在 -5~-15V 之间，‘0’电平在 +5~+15V 之间，与单片机的 TTL 逻辑电平是不一致的，因此在应用中，必须把微处理器的信号电平（TTL 电平）转换为 RS-232C 电平，或者对二者进行逆转换。C8051F040 最多可配置两路全双工串行通讯口，本系统中选用专用电平转换芯片 SP3223E 来实现一路，同时保留了另一路串行口的工作电路的接口于 SBUF2 端子上。

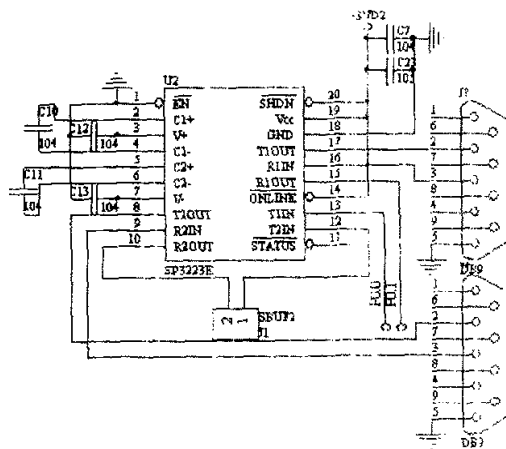


图 3.11 串口电路连接图

Fig. 3.11 Serial bus circuit

SP3223E 是美国 Sipex 公司生产的，包含两路驱动器的低功耗 IC 芯片，主要用于便携设备及手持设备如笔记本电脑及掌上电脑中的应用^[19]。它内部的高效电泵电压变换器在 3.3V 供电电源工作时只需外接 0.1uF 的电容器件，同时电泵电压变换器及 Sipex 的驱

动架构使得 SP3223E 只需要 3.3V—5V 的单一供电电源。没有外设相连时，Sp3223 会自动进入关断状态，此时其消耗电流小于 1 μ A，而当外设连上并进入工作状态后，它会自动唤醒进入工作态，这是它的自动在线技术的特点。在 3.3V 工作电压下，其最小输出驱动电压为 ± 5.0 V，典型值为 ± 5.4 V，满足规定的最恶劣条件下的 ± 5 V 的最小输出电平及 3K 负载。输入门限值为 1.2V 及 2.4V，这比规定的 ± 3 V 更严格。电路接线如图 3.11 所示，按三线制连接串口，也就是说和电脑的 9 针串口只连接其中的 3 根线：第 5 脚的 GND、第 2 脚的 RXD、第 3 脚的 TXD。

3.2.9 CAN 通讯电路

CAN 通讯电路是由单片机的 CANRX 及 CANTX 引脚直接引至 CAN 的收发器来完成的，详细的描述见 5.4.1 节。

第四章 软件设计

4.1 开发平台

开发平台采用 Cygnal 的 Silicon IDE 集成开发环境，该平台内置了 keil C 编译器，提供了开发和测试一个设计所必需的所有工具。IDE 与 C8051f040 的片内 JTAG 和调试逻辑接口，提供一个完整的开发系统，可以对安装在最终应用系统中的 MCU 进行在系统编程和非侵入式的全速、在系统调试。在系统调试不需要额外的目标 RAM、程序存储器或通信通道。IDE 支持 C 语言和汇编语言的源码级调试，支持单步执行（包括单步通过中断服务程序）、运行至断点、条件存储器观察点（数据断点）以及修改和检查寄存器、数据存储器和程序存储器。它可以在 Windows 95/98/ME 和 Windows NT/2000/XP 下运行，与目标 MCU 之间通过 PC 的 USB 口和 MCU 的 JTAG 接口相连，其转换协议由适配器 EC2 来完成。主要特点：源码编辑器；项目管理器；集成的 8051 宏汇编器；FLASH 编程器；支持 Cygnal 的全速、非侵入式在系统调试逻辑；源码级调试；支持第三方开发工具；MCU 配置向导^[20]。

4.2 软件系统流程图及模块构成

软件系统采用前后台系统结构。前后台系统又称为超循环系统，它是基于芯片开发中应用程序常用的结构，整个应用程序是一个无限的循环。循环中调用相应的函数完成相应的操作，这部分可以看成后台行为；中断服务程序处理异步事件，这部分可以看成前台行为。后台又叫任务级，前台又叫中断级。本系统中开启了串口接收的中断及 CAN 总线接口的中断，使通讯作为前台行为，以及时响应上位机的指令响应，其它数据处理等均工作在后台，进行循环执行。其总体程序结构如图 4.1

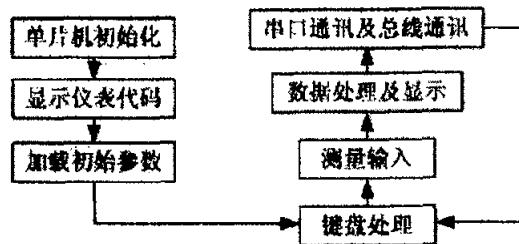


图 4.1 软件流程图

Fig. 4.1 Soft flow diagram

软件模块主要包括：(1)单片机初始化模块；(2)模数转换；(3)数据处理；(4)参数设定和修改；(5)数码显示及指示灯；(6)通讯处理。

4.3 单片机初始化

在进行单片机初始化时应该禁止片内看门狗定时器 (WDT)，防止其在系统初始化时使 CPU 进入复位状态。禁止看门狗则向 WDTCN 寄存器写入 0xDE 后再写入 0xAD 将禁止 WDT。

(1) 可编程数字 I/O 和交叉开关的配置

MCU 中有大量的数字资源需要通过数字 I/O 端口 P0、P1、P2 和 P3 才能使用。端口 0、1 和 2 中的每个引脚即可定义为对应的端口 I/O，又可以分配给一个内部数字资源，这种资源分配的灵活性是通过使用优先权交叉开关译码器实现的^[21]。

交叉开关是一个多路选择器，用于为器件内部的硬件外设分配 I/O 端口。交叉开关负责 SMBus、SPI、UART、定时器捕捉模块、外部 PCA 输入、比较器输出、定时器外部输入、/SYSCLK 以及 A/D 转换启动输入的引脚分配。必须在访问这些外设的 I/O 之前配置和允许交叉开关。未被指定的端口引脚作为通用 I/O 引脚。为了使用端口 0、端口 1 或端口 2 的任何一个引脚作为输出，交叉开关必须被允许。交叉开关通常在复位时被配置一次（在复位处理过程的最开始处），以后不再进行配置。每一种交叉开关设置导致唯一的器件引脚分配，如果在交叉开关中允许或禁止外设，则引脚将发生变化。对于输出端口引脚，其输出方式（漏极开路或推挽）必须被明确设置，即使那些由交叉开关分配的端口引脚也是如此。例外的是 SMBus 上的 SDA 和 SCL 及 UART 的 RXD 引脚，这些引脚被自动配置为漏极开路。

为了将一个通用 I/O 引脚配置为输入，与这一引脚的相关的端口配置寄存器位必须被清‘0’。这样即可选择该引脚为漏极开路输出方式。另外，与该引脚相关的端口位必须被置‘1’，这样使该引脚处于高阻态，或在 XBR2 中的 WEAKPUD 被清‘0’时弱上拉为高电平，这是端口引脚的复位配置。在任意时刻可以通过读取相应端口 SFR 得到端口引脚的电平值，而不管交叉开关寄存器的设置如何或引脚被配置为输入还是输出。与标准 8051 不同的是，CF040 提供了真正的推挽输出。如果需要 8051 的上拉功能，则可以通过将相应的端口输出配置为“推挽”方式，然后再配置为“漏极开路”来进行软件仿真。本系统中采用的如下程序配置：

```
void PORT_Init(void)
{SFRPAGE_SAVE = SFRPAGE;
// Save Current SFR page
SFRPAGE = CONFIG_PAGE; // set SFR page
XBR0    = 0x04;          //配置使能串口一
XBR1    = 0x00;
XBR2    = 0x40;          // 使能交叉开关
XBR3     = 0x80;          //使能 CAN 端口并配置端口为数字输出
P0MDOUT |= 0x01;         //配置串口发送端为数字输出
P1MDOUT |= 0x6A;         //输出端口配置为数字输出
P3MDOUT = 0x07;          //配置输出至显示的端口
P3MDIN  = 0xFF;
SFRPAGE = SFRPAGE_SAVE; //Restore SFR page}
```

(2) 振荡器的配置

每个 MCU 都有一个内部振荡器和一个外部振荡器驱动电路，每个驱动电路都能产

生系统时钟。系统时钟可以自由地在内部振荡器和外部振荡器之间进行切换，也可以在内部振荡器时让外部振荡器保持在允许状态，这样可以避免在系统时钟被切换到外部振荡器时的启动延迟。

外部振荡器具有很高的可配置性，为系统设置者提供了多种选择。时基信号可以从外部 CMOS 电平时钟源、晶体或陶瓷振荡器、RC 组合电路或外部电容获得。在系统复位后，内部振荡器被选为系统时钟并工作在 1.9MHz。使用内部振荡器作为系统时钟的情况下，允许改变内部振荡器的频率。内部振荡器的频率可以在运行中切换，频率改变只需要几个系统周期。

本课题中采用外部晶体振荡器方式。由于晶体振荡器的启动需要几个 ms 的时间，所以需要在将 CLKSL 置‘1’之前等待 XTLVLD 变为高电平（表示外部振荡器已经稳定）。主时钟可以通过将一个晶体并接到 XTAL1 和 XTAL2 引脚而得到。需要将 XOSCMD 设置为‘110’以直接使用晶体频率。其配置程序如下：

```
void SYSCLK_Init(void)
{
    unsigned char n;
    SFRPAGE_SAVE = SFRPAGE;
    // Save Current SFR page
    SFRPAGE = CONFIG_PAGE; // set SFR page
    CLKSEL=0x00;
    //采用内部晶振，可省略，复位时的默认值
    OSCXCN=0x67; //配置为外部晶振，驱动电流 111
    while((OSCXCN&0x80)==0);
    //查询 XTLVLD 是否为 1，若为 1，表明外部晶振稳定，可切换到外部时钟
    CLKSEL=0x01; //选择外部晶振
    OSCICN=0x00; //将内部晶振关闭，也可不关闭，但不关闭是 MCU 功耗能高
    SFRPAGE = SFRPAGE_SAVE; //Restore SFR page
}
for(n=0;n<255;n++); //等街 1ms 以上
```

4.4 模数转换

C8051F040 的 ADC 子系统包括一个 9 通道的可配置模拟多路开关（AMUX），一个可编程增益放大器（PGA）和一个 100kps，12 位分辨率的逐次逼近寄存器型 ADC，ADC 中集成了跟踪保持电路和可编程窗口检测器。AMUX、PGA、数据转换方式及窗口检测器都可用软件通过特殊功能寄存器来配置。只有当 ADC 控制寄存器中的 ADCEN 位被置‘1’时，ADC 子系统才被允许工作。当 ADCEN 位为‘0’时，ADC 子系统处于低功耗关断方式。如果要给 ADC 提供偏置，必须将 REF0CN 寄存器中的偏置允许位置‘1’。AMUX 中的 8 个通道用于外部测量，而第九通道在内部被接到片内温度传感器。PGA 对 AMUX 输出信号的放大倍数由 ADC 配置寄存器确定。PGA 增益可以用软件编程为 0.5、1、2、4、8 或 16，复位时的默认增益为 1。

ADC 使用 VREF 来确定它的满度电压，因此在进行一次转换之前必须正确设置这个参考电压。本系统中采用内部参考电压，其值为 2.43V。ADC 的最高转化速度为 100kps，转换时钟来源于系统时钟。可以通过设置 ADC0CF 寄存器将转换时钟的速度最高降为系统时钟的 1/32，这一功能用于根据不同的系统时钟速度调整转换速度，系统默认 ADC 的转换时钟为系统时钟的 32 分频。ADC0 的最大 SAR 时钟为 2.5MHz，分辨

率为 12bits, 偏移误差为 $\pm 1\text{LSB}$, 转换占用 16 个 SAR 时钟, 输入电压范围 $0-V_{\text{REF}}V$, 信号建立时间约为 $0.5\mu\text{s}$, 转换前均有 3SARS 时钟跟踪时间。本系统中 $\text{SAR0} = \text{SYSCLK}/9 = 22118400/9 \approx 2.5\text{M}$, 转换的速度也接近于 100K 。

ADC 采样的过程: ①设置电压参考标准; ②设置 ADC 转换的速率和 PGA; ③配置模拟多路开关 (AMUX); ④向 ADBUSY 写‘1’开始跟踪, 持续 3 个 SAR; ⑤等待转换结束 $\text{ADCINT}=1$; ⑥读取 ADC 的采样结果。

本系统中使用了过采样和求均值的方法^[22]以提高 ADC 分辨率, 该方法通过连续 ADC256 次并求和, 然后除以 16 (通过右移四位得到), 从而得到 16 位的模数转换值, 从而把原 ADC 的 12 位分辨率提高至 16 位。尽管这一过程会增加 CPU 的时间, 但由于采样的速率足够快, 对采样 256 次而言, 所需时间只有 $256 \times 1/100\text{K} \approx 2.56\text{ms}$, 加上指令执行的时间 μs 级的, 每次 ADC 转换时间对于系统来言仍然是足够快的, 所以过采样 256 次求平均和并没有降低太多的系统性能。其程序如下, 输入采样通道及放大增益, 返回 16 位的 ADC 结果:

```
long GET_ADC(uchar subindex, uchar pganum)
//subindex 为 amx0 寄存器的 3-0 位, 0000-0111
//选择 0-7 通道, 1XXX 选第 8 通道
{
    unsigned long adc_result;
    unsigned int int_dec;
    //对于 256 来说, uchar 型超限, 所以用 int 型
    SFRPAGE_SAVE=SFRPAGE;//Save SFRpage
    SFRPAGE = ADC0_PAGE;
    if (pganum!=7)    ADC0CF |= pganum;
    //增益 (000-1,01-2,010-4, 011-8,10x-16,11x-0.5)
    if (subindex!=15) AMX0SL = subindex;
    //选择通道, 通道数为 15 时, 保持寄存器值不变
    adc_result=0L;
    for(int_dec=0;int_dec<INT_DEC;int_dec++)
    {
        AD0BUSY=1;//启动一次转换
        while(1)    //等待一次完成转换
        {
            if(AD0INT==1)
            {
                AD0INT = 0;
                break;
            }
        }
        adc_result=adc_result+ADC0;
    }
    adc_result=adc_result>>4;
    //得到 16 位有效值 (256 个值求和然后除以
    //16,12 位的 256 个值的和是 20 位, 右移 4 位之后
    //是 16 位)
    return adc_result;
}
```

4.5 采样值的数据处理

模数转换过程中采用的过采样和求均值法等价于一个降采样低通滤波器, 这个数字信号处理过程通常被称为插补, 求均值的样本数量越大, 则低通滤波器的选择性也越强, 插值的效果会越好, 但相应实现难度也越大, 动态性能也会降低。这种方法虽然可以改善信噪比, 但仅限于对白噪声 (频谱分布为从零到无穷大的一种理论上信号) 有效, 它的意义更多的在于减少了量化噪声, 而对不能用白噪声建模的噪声不起作用, 而且量化噪声的功率大于自然白噪声 (例如热噪声), 这种技术也不会奏效。为了减少测量系统中常见的工频干扰等, 本系统采用了算术平均滤波法^[23], 而对该方法的参数值的确定采用了试验分析的方法, 其过程如下: 首先获取多次连续的模数转换值, 然后采用不同的算术平均滤波法的参数值对这些数据进行滤波, 最后对试验结果进行比较, 得出一个对

本系统最好的算术平均滤波法的参数值。以下给出了连续进行 1000 次模数转换及通过串口发送给上位机的数据采集试验的部分代码：

```

unsigned int xdata testdata[1000];
//声明长度为 1000 的 INT 型数组
for(tdataindex=0;tdataindex<1000;tdataindex++)
//连续采样 1000 次
{   adc_result=0L;
    for(int_dec=0;int_dec<INT_DEC;int_dec++)
    {   AD0BUSY=1;//启动 A/D 转换
        while(1)
        {   if(AD0INT==1)
            {   AD0INT = 0;
                break;
            }   }
        adc_result=adc_result+ADC0;
    }
    adc_result=adc_result>>4;
    //256 次采样和求平均值
    testdata[tdataindex]=adc_result;}
//串口连续发送 1000 个 INT 型数据程序,通过主
//机主动发送命令字应答完成
unsigned char ii, iii;
unsigned char *s_testp;//声明字符型指针
s0_mode=0xff;
s_tbuff[0]=s_rbuff[0];//发送命令字为数字字符 7
for(ii=0;ii<10;ii++)//总发 10 次
{   s_testp=&testdata[ii*100];
    for(iii=0;iii<50;iii++)
    //发 100 个, 一次循环发两个数
    {   s_tbuff[1]=*s_testp;
        s_tbuff[2]=*(s_testp+1);
        s_tbuff[3]=*(s_testp+2);
        s_tbuff[4]=*(s_testp+3);
        s_testp=s_testp+4;
        s_send();
    }
}
注: 连续发 1000 个数据, 采用一次循环不成功,
应该是指针型变量出界, 采用一次循环体只收到
256 个有效数据
    
```

实验一：1.5V 干电池，10K 电阻及 100K 电位器各一只，组成简单的串联电路，对两电阻之间的点进行模数转换的数据采集。图 4.2 给出了 Matlab 下计算机对接收到的 1000 个数据的前 100 个数据的 plot 函数图。

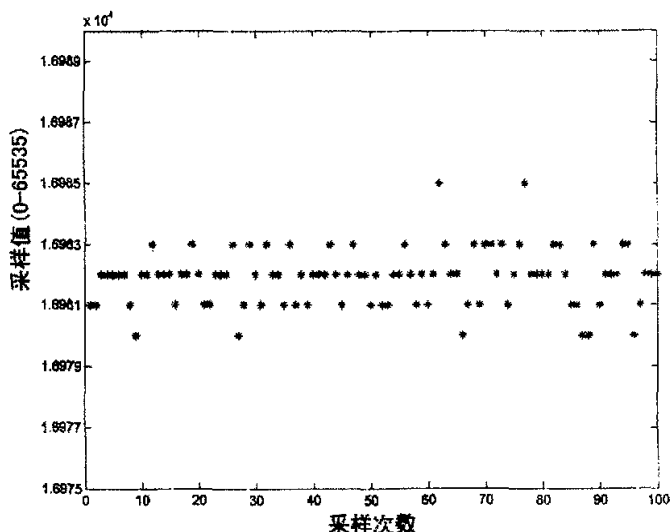


图 4.2 干电池电路试验的采样值

Fig. 4.2 AD sample with dry battery

从图中可看出，采样值集中在 16981-16983 之间，最大采样值为 16985，最小采样

值也只有 16979，差值仅为 6，其采样精度为 6/65535，要高于 12bit A/D 的分辨率值 1/4095，这说明过采样和求均值法能有效抑制噪声，提高 ADC 的转换精度，但同时说明 ADC 对采样信号的要求极高，任何小的波动，都会反应到 ADC 的输出值里面。

实验二：对本系统中的小信号放大电路的输出信号进行数据采集实验，上位机把获取的数据存入一文本文件，仍然采用 plot 函数得到图 4.3。

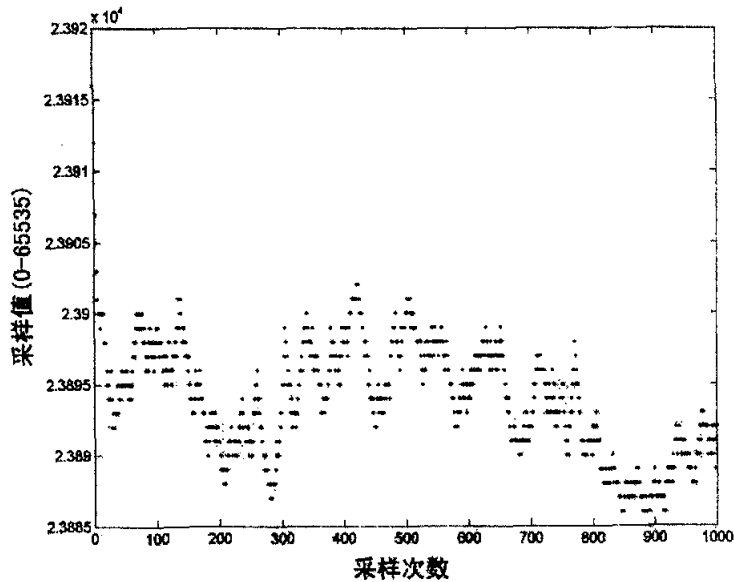


图 4.3 测量电路的采样值

Fig. 4.3 AD sample with measurement circuit

从图中可看出，前三次采样值与后续的采样值之间的差距非常明显，考虑 A/D 转换建立的瞬时电路的影响，所以在实际数据处理过程中，舍弃前三次的采样值；常见的工

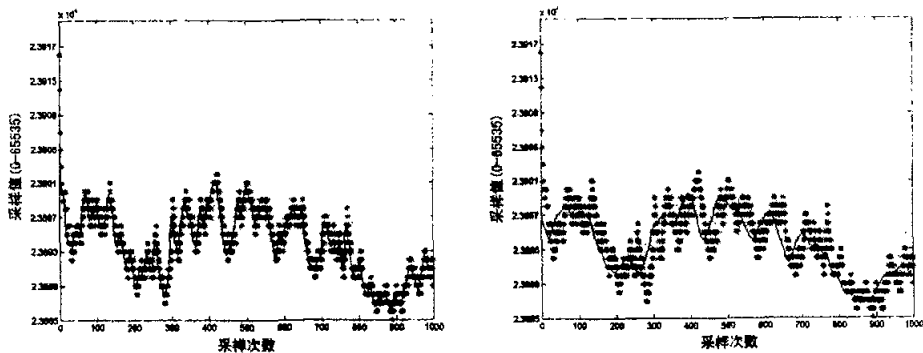


图 4.4 8 个或 40 个连续采样值取平均

Fig. 4.4 8 or 40 times ADC data to average

频干扰，其频率为 50Hz，即周期为 20ms，从理论上讲，连续取 20ms 内的采样值求平均可消除这一干扰，而一次模数转换的完成时间大约在 2.5ms，所以算术平均滤波方

法下取参数为 8 的整数倍时可消除这一干扰。从波形同时可看出，两个波谷之间的距离均在 100 次左右，所以取平均值的次数为 80 次，这仅仅是一个理论上的分析数据，而该数据最后又是进行了多次实验才确定的。图 4.4 给出了参数为 8 和 40 时的算术平均滤波方法处理效果图，图 4.5 给出了参数为 80 时的算术平均滤波法的效果图，图中*号线为原始数据的 plot 函数图，而连续的线则是经过算术平均滤波处理后得到的 plot 函数图。

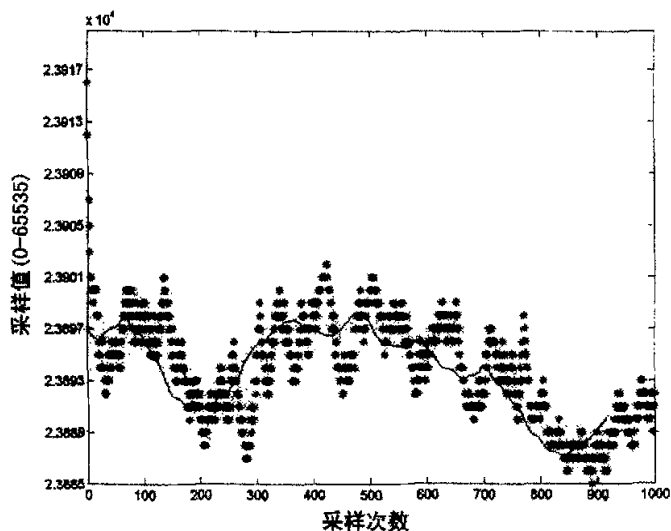


图 4.5 连续 80 个采样值取平均

Fig. 4.5 80 times ADC data to average

生成图例的 Matlab 程序为：

```
function []=datashow(columns)%定义函数，其中
columns 为算术平均滤波法的参数
fid=fopen('d:\data4.txt','r');%打开文件
T=fscanf(fid,'%g',[1 inf]);%导入矩阵
b=zeros(1,1000-columns);%定义新矩阵
for i=1:1:1000-columns
    s=0;
    for j=i:1:i+columns-1
        s=s+T(j);%对 T 中连续个 columns 数据求和
    end
    s=s/columns;%取平均
    b(i)=s;%给矩阵 b
end
plot(b);%绘图
%下面为在 matlab 下命令行的程序
>> fid=fopen('d:\data4.txt','r');
>> T=fscanf(fid,'%g',[1 inf]);
>> plot(T,'*');
>> xlabel('采样次数',FontSize,28);%设横轴标识
>> ylabel('采样值 (0-65535)',FontSize,28);%设纵轴标识
>> set(gca,'ytick',[23885:4:23918]);%设纵轴间隔
>> hold on;%绘新图时不消除原图
>> data(80);
```

比较图 4.4 和图 4.5 可发现，当算术平均滤波法的参数为 80 时，其性能最好。而又进行的参数超过 80 的算术平均滤波法的其它实验，可发现其性能提高不明显，同时考虑单片机的处理能力及对信号的动态跟随上，由于取样次数越大，其去干扰能力越强，但处理的难度加大，而且动态测量能力上会越差，所以本系统中算术平均滤波法的参数最终确定为 80，此时其精度可确保在 1/4095 之上。而对于键盘标定时采用了参数为 120

的算术平均滤波法,这主要考虑标定值的精度对测量结果的精度影响很大,且标定过程对时间的跟随要求不高,所以选用了更大参数的算术平均滤波方法。

4.6 分段线性化输出

A/D 采样值只是中间结果,对用户来讲,要看到的是实际的称重显示值。本系统中采用了动态分段线性化的处理技术。用户在使用时,首先须对本系统做标定,标定数值从 2-9 次不等,从而提供一段直线或者八段直线的标定。实现过程如下:

```
struct pairedata {unsigned int adcddata;          array_data[i].inputdata=0;
    unsigned char inputdata;
} xdata array_data[9];
//定义结构体,utdata 为输入的标定值,adcddata
//为系统当前自动获得的采样值
void array_sort(void)//排序
{
    unsigned char i,j;
    par_read(0x31,array_data,sizeof(array_data));
    //读取标定值
    for(i=0;i<9;i++)
    {
        if(array_data[i].adcddata==0xffff)
        //如果没有输入标定值全部赋0
        {array_data[i].adcddata=0;
        }
    }
    for(i=0;i<8;i++) //起泡法排序
    {
        for(j=i+1;j<9;j++)
        {if(array_data[j].adcddata<array_data[i].adcddata)
        {
            array_temp.adcddata=array_data[j].adcddata;
            array_temp.inputdata=array_data[j].inputdata;
            array_data[j].adcddata=array_data[i].adcddata;
            array_data[j].inputdata=array_data[i].inputdata;
            array_data[i].adcddata=array_temp.adcddata;
            array_data[i].inputdata=array_temp.inputdata;
        }
    }
}
```

每次修改标定值时,系统自动调用排序程序,从而产生新的分段标定值。从 A/D 转换得来的值首先进行定位,以确定处于哪一段,然后采用就近两个标定值确定的直线来运算确定相应的输出值。最小值以下与最大值以上则分别采用最左端及最右端的线段的延长线来进行计算。

4.7 参数设定及修改

C8051F040 闪存含 64KB 的程序存储单元和 128 字节的快速存储单元(存储数据用)。快速存储单元的地址为 0x00-0x7F,只能作为数据存储用,不能作为程序代码存储。闪存支持在线编程,可通过 JTAG 口或 MOVX 指令写入数据。在读写闪存时,MCU 处于挂起状态,此时若发生中断,则中断入栈,等闪存读写完毕再处理。

写 FLASH 存储器可以清除数据位,但不能使数据位置‘1’。只有擦除操作能将 FLASH 中的数据位置‘1’。所以在写入新值之前,必须先擦除待写入的地址。64K 字节的 FLASH 存储器是以 512 字节的扇区为单位组织的。一次擦除操作将擦除整个扇区(将扇区内的所有字节设置为 0xFF)。对快速存储单元也是如此,扇区结构要求数据应先读入一个缓冲区,然后修改数据后回写。

擦除一个 FLASH 页的步骤如下:①禁止中断,将 FLWE (FLSCL.0)置 1,使能 FLASH 擦/写;置位 PSEE (PSCTL.1)和 PSWE (PSCTL.0)允许 FLASH 扇区写和擦除;②向待擦除扇区内的任何一页写入一个数据字节,此时该页被擦除;③清除 PSEE

(PSCTL.1) 以禁止 FLASH 扇区擦除。

本系统中采用了 128 字节的快速存储单元来存储设置的参数，参数设置通过键盘的组合来实现，表 4.1 指明了各键功能及使用说明，具体被设置参数在数据存储器中的分配如下：

表 4.1 键盘功能说明
Table 4.1 Function of keys

Key	按键	按键功能
K1	回零	在称量工作状态时（“运行”指示灯熄灭），使显示回零。在称量为动态、锁零时此按键无效。在参数设置时使设置退到根菜单；值设置时取消当前输入，退回到根菜单
K2	启动→	称量工作状态时可使仪表进入配料工作状态；参数设置时进入上一级菜单；置数时可使显示位右移一位
K3	停机←	配料工作状态时可使仪表进入称量工作状态；参数设置时可使仪表进入下一级菜单；置数时左移一位数
K4	切换	重新启动 CAN；参数设置时可用于同级菜单项之间进行切换；置数时使最低位在 0——9 之间循环
K5	设置	称量工作状态时，显示温度值；参数设置时无效；置数时确认输入的数据并使菜单退回到对应的值设置菜单
K6	菜单	在参数设置状态与运行状态之间进行切换

临时存储器的分配使用如表 4.2：

表 4.2 临时存储器的分配
Table 4.2 Temp flash use

存储地址	存储参数说明
0x01	can_baud (F42) 设置 CAN 的波特率
0x02	baud 的索引值 (F21)
0x03	sub_mod 的索引值 (F22)
0x04	canmacid (F41) 刷新 CAN 标识值
0x05	alarm_hvalue, 2 个字节 (F11), 最高上限值
0x07	alarm_lvalue, 2 个字节 (F12), 最低限值
0x09	Zerodata, 2 个字节, 零点的 A/D 转换值
0x31-0x4c	27 个值, 存贮分段的标定值 (F31—F39)

每次修改参数的时候，新的参数值会被写入，采用了回写机制，即先读取整个扇区至内存区（开辟一段内存做缓冲区），修改指定参数，然后再整体回写回去，程序如下：

```
void par_write(unsigned char xdata *pwrite, unsigned char *buffrom, unsigned char len)
{
    /*pwrite 为 Flash 地址指针，采用 xdata 强制指明指针类型，以保证采用 MOVX 操作；buffrom 为数据指针；len 为数据长度。*/
    unsigned char ii,tmpdatax;
    unsigned char xdata tmp_buff[128]; //0-127 下标
    unsigned char code *pread;
    SFRPAGE_SAVE=SFRPAGE;
    SFRPAGE_SAVE=0x00;
    pread=0;

    PSCTL=0x04;
    //选中对临时存储器操作，读取数值
    for(ii=0;ii<128;ii++)
    {
        tmp_buff[ii]=*pread;
        pread++;
    }
    PSCTL=0; //去掉对临时操作器的选取
    tmpdatax=(char)pwrite; //修改指定参数值
    for(ii=0;ii<len;ii++)
        tmp_buff[tmpdatax+ii]=*buffrom++;
    WDTCN=0xde; //标止看门狗
    WDTCN=0xad; //禁止中断
}
```

```

EA=0;
FLSCL=0x81;
PSCtl=0x07;
*pwrite=0;//擦除操作
PSCtl=0x05;
pwrite=00;
for(ii=0;ii<128;ii++)//写回参数新值
{
    *pwrite++=tmp_buff[ii];
    PSCtl=0;//去掉使能闪存写
    FLSCL=0x80;
    SFRPAGE=SFRPAGE_SAVE;
    EA=1;
    SFRPAGE=SFRPAGE_SAVE;
}

```

4.8 键盘及显示

键盘部分的程序实现,采用查询方式,通过间隔 20ms 重读一次端口值与原先值进行比较去除抖动。其控制流程主要是采用了使用全局变量做标志位与按键进行一一对应的方法,当检测到键盘动作时,更新标志位的值,然后调用键盘子程序。

显示采用了 6 个数码管及 8 个状态指示灯,所以共需要 7 字节的显示缓冲区,在本系统的设计中,使用了 7 个字节的全局变量数组作为显示缓冲区,dispbuff[0-5]为数码管显示值,dispbuff[6]为发光二极管控制值,分别控制 L1 表停止状态, L2 表显示当前温度值, L3 表单次测量, L4 表跟随测量, L5 表菜单功能设置状态, L6 表 CAN 发送故障。把欲显示的值放入到缓冲区中,缓冲区再直接送入到数码显示电路,显示程序如下:

```

void disp()
{
    unsigned short n,k,kpp;
    bit flagt;
    //先清除当前显示的值
    for(n=0;n<=8;n++)
    {
        for(k=0;k<=7;k++)
        {
            dispclk=0;//产生方波时钟
            dispdata=0;
            dispclk=1;
        }
    }
    dispstb=0;//熄灭所有指示灯及 LED
    for(n=0;n<=6;n++)
    {
        kpp=1;//显示值送入显示芯片控制器
        for(k=0;k<=7;k++)
        {
            if((dispbuff[n]&kpp)==0) flagt=0;
            else flagt=1;
            kpp=kpp*2;
            dispclk=0;
            dispdata=flagt;
            dispclk=1;
        }
        dispstb=1;
    }
}

```

4.9 通讯处理

4.9.1 单片机串口程序

串行口主要用来与 PC 机进行通讯,从而一些比较复杂的操作可以用计算机的图形化界面来做,比如参数的设定。串行口的波特率可以进行选择,本系统中提供了六种(4800,9600,14400,19200,38400,115200)波特率,均是采用 8 位数据位和 1 位开始停止位,无奇偶校验的配置。参考 CAN 的 8 字节固长数据机制,考虑本系统输出数据为两个整型 4 字节长数据,并使用 1 字节来做命令字,所以设置通讯的缓冲区为 5 字节,即 5 字节帧结构式通讯。单片机的数据输出可配置为数据广播模式或问答模式,在广播模式下,单片机连续的定时向 PC 输出数据,而在问答模式下,PC 机向单片机发送指令,

单片机回答一数据。而参数配置都采用有确认的数据报的服务模式，PC 机发送配置信息，单片机接收到后进行一个确认回复。PC 发送至单片机的帧意义定义如表 4.3。对于修改参数值的命令，单片机修改成功后均返回原命令字至 PC 以通知 PC 操作成功。串口接收时采用中断模式，每接收到五个字节后进行数据处理；发送采用查询模式。

表 4.3 单片机接收到命令字规定

Table 4.3 CIP received command specification

BYTE1	BYTE2	BYTE3	BYTE4	BYTE5	说 明
0x01	-	-	-	-	启动数据发送允许位
0x31	0x30	0x31	0x30	0x31	测试与串行口的连接
0x32	-	-	-	-	读串口的配置信息
0x33	0x00	-	-	-	读当前的配置值
	0x01	地址	字节 1	字节 2	写新的上限值下限值至单片机
	<0x4d >0x30	标定 地址	字节 1	字节 2	写新标定值至单片机
0x34	0x30	-	-	-	读当前串行口数据的发送模式
	0x00	-	-	-	修改串行口发送模式
0x35	-	-	-	-	跟随读取 10 分钟内的实验数据值
0x37	-	-	-	-	读取 1000 个试验数据

中断服务程序及发送程序代码如下：

```
void UART0_ISR (void) interrupt 4
//中断服务子程序
{
    if (RI0) //接收到一字节数据
    {
        s_rbuff[s_rnum]=SBUF0;
        if (s_rnum==4)//检查是否到五字节
        {
            s_rnum=0;
            dealdata();//命令处理函数
        }else
            s_rnum=s_rnum+1;
        RI0=0;
    }
}

void s_send(void) //发送子程序
{
    unsigned char xxx;
    unsigned char counter_time,jj;
    unsigned char IEtemp;
    IEtemp=IE;
    IE=0x00; //禁止发送中断
    TR1 = 1; //启动 T1
```

```
TI0=0; //开始发送数据
for(xxx=0;xxx<5;xxx++)
{
    SBUF0 = s_tbuff[xxx];
    counter_time=0;
    while(TI0!=1)
    {
        my_delay(1); //等待发送完
        counter_time++;
        if (counter_time>200)//发送超时
        {
            for(jj=0;jj<6;jj++)
            {
                dispbuff[jj]=num[0];
                //显示全 0 提示发送失败
                return; //error send ;
            }
        }
        TI0=0;
    }
    IE=IEtemp;//打开中断
```

4.9.2 PC 机程序

PC 端采用 Visual Basic 来进行实现设计，Visual Basic 6.0（以下简称 VB）^[24]是一种功能强大、简单易学的程序设计语言。它不但保留了原先 Basic 语言的全部功能，而且还增加了面向对象程序设计功能。它不仅可以方便快捷地编写适用于数据处理、多媒体等方面的程序，而且利用 ActiveX 控件 MSComm 还能十分方便地开发出使用计算机串口的计算机通信程序，本系统就是使用了 MSComm 来完成与单片机的通讯。

4.9.2.1 设计简介

MSComm 控件提供了两种处理通信的方式：一种为事件驱动方式，该方式相当于一般程序设计中的中断方式。当串口发生事件或错误时，MSComm 控件会产生 OnComm 事件，用户程序可以捕获该事件进行相应处理。另一种为查询方式，在用户程序中设计定时或不定时查询 MSComm 控件的某些属性是否发生变化，从而确定相应处理。在程序空闲时间较多时可以采用该方式。本系统中使用的为事件驱动方式。利用 MSComm 控件实现计算机通信的关键是理解并正确设置 MSComm 控件众多属性和方法。以下是 MSComm 控件的常用属性和方法^[25]：

(1)Commport：设置或返回串口号；Settings：以字符串的形式设置或返回串口通信参数。

(2)Portopen：设置或返回串口状态；InputMode：设置或返回接收数据的类型。

(3)Inputlen：设置或返回一次从接收缓冲区中读取字节数；InBufferSize：设置或返回接收缓冲区的大小，缺省值为 1024 字节；InBufferCount：设置或返回接收缓冲区中等待计算机接收的字符数；Input：从接收缓冲区中读取数据并清空该缓冲区，该属性设计时无效，运行时只读。

(4)OutBufferSize：设置或返回发送缓冲区的大小，缺省值为 512 字节；OutBufferCount：设置或返回发送缓冲区中等待计算机发送的字符数；Output：向发送缓冲区发送数据，该属性设计时无效，运行时只读。

(5)Rthreshold：该属性为一阈值。当接收缓冲区中字符数达到该值时，MSComm 控件设置 Commevent 属性为 ComEvReceive，并产生 OnComm 事件。若 Rthreshold 属性设置为 0，则不产生 OnComm 事件。

(6)Sthreshold：该属性亦为一阈值。当发送缓冲区中字符数小于该值时，MSComm 控件设置 Commevent 属性为 ComEvSend，并产生 OnComm 事件。要特别注意的是仅当发送缓冲区中字符数小于该值的瞬间才产生 OnComm 事件，其后就不再产生 OnComm 事件。若 Sthreshold 属性设置为 0，则不产生 OnComm 事件。

表 4.4 CommEvent 通信事件

Table 4.4 CommEvent specification

代码	常数	含义
1	ComEvReceive	接收到 Rthreshold 个字符。该事件将持续产生，直到用 Input 属性从接收缓冲区中读取并删除字符。
2	ComEvSend	发送缓冲区中数据少于 Sthreshold 个，说明串口已经发送了一些数据，程序可以用 Output 属性继续发送数据。
3	ComEvCTS	Clear To Send 信号线状态发生变化。
4	ComEvDSR	Data Set Ready 信号线状态从 1 变到 0。
5	ComEvCD	Carrier Detect 信号线状态发生变化。
6	ComEvRing	检测到振铃信号。
7	ComEvEOF	接收到文件结束符。

(7)CommEvent: 这是一个非常重要的属性。该属性设计时无效，运行时只读。一旦串口发生通信事件或产生错误，依据产生的事件和错误，MSComm 控件为 CommEvent 属性赋不同的代码，同时产生 OnComm 事件。用户程序就可在 OnComm 事件处理程序中针对不同的代码，进行相应的处理。CommEvent 属性的代码、常数及含义参见表 4.4。

本系统中 MSCommok 的属性设置如下：

```
MSCommok.CommPort = com_index
    '配置端口号及选择配置信息
MSCommok.Settings = Getsetting()
MSCommok.InBufferSize = 40 '读取缓冲区大小
MSCommok.InputLen = 5
    '一次从缓冲区中读取的字节数
MSCommok.RThreshold = 5 '读取阈值
MSCommok.InputMode = comInputModeBinary
    '设置为二进制通讯模式
MSCommok.InBufferCount = 0
MSCommok.OutBufferSize = 40
MSCommok.SThreshold = 5 '发送的阈值
```

主要的事件响应程序如下：

```
Private Sub MSCommok_OnComm()
    '接收数据的部分代码
```

```
With MSCommok
    '读接收的数据
    Select Case MSCommok.CommEvent
        '判断通信事件
        Case comEvReceive: '收到 threshold 字节数据
            var = Null 'var 为变体型全局变量
            var = MSCommok.Input '读取五字节的数据
            Outi = var '赋给 byte 型数组
            Dealdata '调用数据处理函数
        Case Else 'PortOpen = False
    End Select
End With
End Sub
```

而对发送的过程主要如下：

```
Outstr = "40000" 'Outstr 为字符串变量
Outbuffer = Outstr '把变量赋值给变体型变量
MSCommok.Output = Outbuffer '把变体型变量
值送入到输出缓冲区
```

4.9.2.2 软件简单说明

启动窗口如图 4.6 左侧图所示，首先应先对串行口参数进行配置，然后点测试按钮，此时软件后台会发一测试帧至单片机，如果得到正确的数据响应则其它选择项激活，从而可进行配置及数据读取窗口。



图 4.6 软件启动窗口和试验二窗口

Fig. 4.6 Start form of VB soft and Test2 window

数据窗口显示当前的采样值，设置一窗口主要设置标定的值、零点以及上限下限值，设置二设置串行口的通讯模式，试验一用来跟随 10 分钟的 A/D 采样值，试验二用来进行数据采集实验，每次可获得连续的 1000 次采样值如图 4.6 右侧图。

第五章 CAN 总线及 CANopen 协议

5.1 CAN 网络参考模型

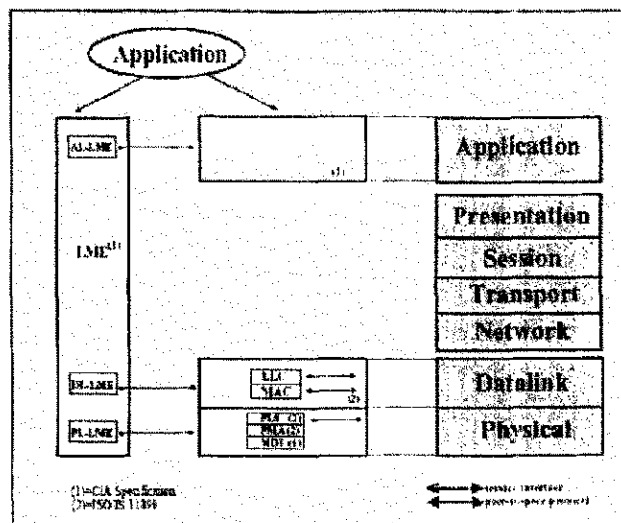


图 5.1 CAN 参考模型

Fig. 5.1 CAN reference model

CAN 总线系统结构^[26]对应计算机网络结构统一为四层（图 5.1），即物理层、数据链路层、应用层和用户层。对应 OSI 的第三到五层在 CAN 网络中没有的原因：CAN 网络为局域网络，没有网络间的相互访问及路由功能的需要，每个 CAN 报文会到达所有 CAN 节点，所以没有网络层；CAN 网络中没有网络层，而且数据链路层被认为是足够可靠的，所以不需要传输层来提供可靠的数据传输服务，若是确实需要，而由应用层里来实现相应的服务；在实时分布式控制系统中，通常不支持会话、同步点及回退机制，所以没有会话层，但 CiA 在将来会选择性增加会话层。

CAN 的规范定义了模型的最下面两层：数据链路层和物理层。而应用层协议可以由 CAN 用户定义成适合特别工业领域的任何方案。已在工业控制和制造业领域得到广泛应用的标准是 DeviceNet，这是为 PLC 和智能传感器设计的。国际上广泛应用的 CAN 协议工业标准还有 CANopen 和 SDS。在汽车工业，许多制造商都应用他们自己的标准。

5.1.1 CAN 物理层

CAN 物理层在不同节点之间根据所有的电气属性进行位信息的实际传输。CAN 能
CAN 物理层在不同节点之间根据所有的电气属性进行位信息的实际传输。CAN 能

够使用多种物理介质,例如双绞线、光纤等,最常用的就是双绞线。信号使用差分电压传送,能有效地抵偿噪音,其差分的电压令 CAN 网络即使在一条信号线断开或者在噪声极大的环境中也能够工作。CAN 收发器提供便宜的接口,把 5V 的逻辑电平转换成 CAN 要求的对称线电平,两条信号线被称为“CAN_H”和“CAN_L”,静态时均是 2.5V 左右,此时状态表示为逻辑‘1’,也可以叫做“隐性”。用 CAN_H 比 CAN_L 高表示逻辑‘0’,称为“显性”,此时,通常电压值为:CAN_H=3.5V 和 CAN_L=1.5V。

5.1.2 CAN 数据链路层

CAN 数据链路层又称为 CAN 传输层,该层的作用主要是传送规则,也就是控制帧结构、执行仲裁、错误检测、出错标定、故障界定等,以及总线上什么时候开始发送新报文及什么时候开始接收报文,均在该层确定。位定时的一些普通功能也可以看作是传输层的一部分。该层的定义规范有 CAN1.2 和更新的 CAN2.0,分别对应标准格式和扩展格式报文^[27]。这两种格式的报文其显著不同是其标识位的位数,标准格式报文其地址范围由 11 位识别位定义,而扩展格式报文地址范围同 29 位识别位定义。这两种报文可共存于同一个网络中,通过报文中的 PTR 位进行区别。

CAN 网络使用载波侦听多路访问/冲突检测 CSMA/CD 机制(Carrier Sense Multiple Access with Collision Detect)来提供其仲裁机制。利用 CSMA 访问总线,可对总线上信号进行检测,只有当总线处于空闲状态时,才允许发送。利用这种方法,可以允许多个节点挂接到同一网络上。当检测到一个冲突位时,所有节点重新回到“监听”总线状态,直到该冲突时间过后,才开始发送。在总线超载的情况下,这种技术可能会造成发送信号经过许多延迟。为了避免发送时延,可利用 CSMA/CD 方式访问总线。当总线上有两个节点同时进行发送时,必须通过“无损的逐位仲裁”方法来使有最高优先权的的报文优先发送。在 CAN 总线上发送的每一条报文都具有唯一的一个 11 位或 29 位数字的 ID。CAN 总线状态取决于二进制数‘0’而不是‘1’,所以 ID 号越小,则该报文拥有越高的优先权。因此一个为全‘0’标志符的报文具有总线上的最高级优先权。可用另外的方法来解释:在消息冲突的位置,第一个节点发送 0 而另外的节点发送 1,那么发送 0 的节点将取得总线的控制权,并且能够成功的发送出它的信息。

CAN 网络中链路层的报文传输由 4 种不同的帧类型表示和控制:数据帧,将数据从发送器传输到接收器;远程帧,总线单元发出远程帧,请求发送具有同一识别符的数据帧;错误帧,任何单元检测到总线错误就发出错误帧;过载帧,过载帧用以在先行的和后续的数据帧(或远程帧)之间提供一附加的延时。数据帧和远程帧可以使用标准帧及扩展帧两种格式,它们用一个帧间空间与前面的帧分隔。

物理层与数据链路层完全由硬件实现,这些硬件部件有带有集成 CAN 接口的微控制器(如:与 8051 兼容的 Philips P97C591 处理器、Cygnal F040 以及基于 Arm7 核心的 Philips LPC2194 等),也有独立的 CAN 控制器(如:Philips SJA1000),其它 SIEMENS,

NEC 和 INTEL 也提供独立的 CAN 控制器。

5.1.3 CAN 应用层

由于没有规定应用层, CAN 本身并不完整, 因此需要一个高层协议来定义报文中的 11/29 位标识符、8 字节数据的使用。要成为一个开放的、标准化的高层协议: 这个协议应支持各种 CAN 厂商设备的互用性、互换性, 能够实现在 CAN 网络中提供标准的、统一的系统通讯模式, 提供设备功能描述方式, 执行网络管理功能。应用层 (Application layer), 为网络中每一个有效设备都能够提供一组有用的服务与协议; 通讯描述 (Communication profile), 提供配置设备、通讯数据的含义, 定义数据通讯方式; 设备描述 (Device profile), 为设备 (类) 增加符合规范的行为。许多系统 (像汽车工业) 中, 可以特别制定一个合适的应用层, 但对于许多的行业来说, 这种方法是不经济的。一些组织已经研究并开放了应用层标准, 以使系统的综合应用变得十分容易。一些可使用的 CAN 高层协议^[28] (制定组织及主要高层协议) 有: CiA CAL 协议; CiA CANOpen 协议; ODVA DeviceNet 协议; Honeywell SDS 协议; Kvaser CANKingdom 协议。

5.1.3.1 DeviceNet

在 1990 的早些时候, 美国俄亥俄州的机械工程公司的工程师们与 Allen-Bradley 公司、Honeywell 微型开关公司开始了一个合资项目, 内容是基于 CAN 的通讯与控制。但是, 不久之后, 项目组的重要成员离开合资项目终止。但 Allen-Bradley 公司和 Honeywell 公司各自继续从事这项工作。这导致产生了两个高层协议: “DeviceNet” 和 “Smart Distributed System (SDS)”, 而且这 2 个协议在较低层的通讯层上非常相似。在 1994 年早些时候, Allen-Bradley 将 DeviceNet 规范移交给专职推广 DeviceNet 的组织 “Open DeviceNet Vendor Association (ODVA)”。而 Honeywell 则放弃了在 SDS 方面的努力, 使得 SDS 更象 Honeywell 公司的内部解决方案。DeviceNet 特别为工厂自动控制而定制, 因此, 使其成为类似 Profibus-DP 和 Interbus 协议的有力竞争者。倘若仅从即插即用的功能考虑, DeviceNet 已经成为美国特定应用领域中的领导者。

5.1.3.2 CAL 及 CANopen

CAL (CAN Application Layer) 协议是目前基于 CAN 的高层通讯协议中的一种, 最早由 Philips 医疗设备部门制定。现在 CAL 由独立的 CAN 用户和制造商集团 CiA 协会负责管理、发展和推广。尽管 CAL 在理论上正确, 并在工业上可以投入应用, 但每个用户都必须设计一个新的子协议, 因为 CAL 是一个真正的应用层, 所以 CAL 常被看作应用 CAN 方案的必要理论步骤, 它却不会被推广。从 1993 年起, 在 Esprit project ASPIC 范围内, 由 Bosch 领导的欧洲协会研究出一个原型, 由此发展成为 CANopen。它是一个基于 CAL 的子协议, 用于产品部件的内部网络控制。在理论方面, 来自德国 Reutlingen 的 Applied Science 大学教授 Gerhard Gruhler 博士和来自 Newcastle (UK) 大学的 Mohammed Farsi 均是最成功的活跃分子之一。在项目完成之后, CANopen 规范移

交给 CiA 组织进行维护与发展。在 1995 年, CiA 发表了完整版的 CANopen 通讯子协议; 仅仅用了 5 年的时间, 它已成为全欧洲最重要的嵌入式网络标准。CANopen 不仅定义了应用层和通讯子协议, 也为可编程系统、不同器件、接口、应用子协议定义了页状态, 这也就是工业领域(比如: 打印机、海事应用、医疗系统)决定使用 CANopen 的一个重要原因。

5.2 CAL 协议

CAL (CAN Application Layer) 是一个描述 CAN 网络的基本术语, CANopen 网络协议概念就是建立在它和串行总线系统基础之上的。CAL 提供对用户来说非常有用的通讯服务集合, 它使用户可以在他们的应用中随意配置, CAL 提供了 4 种应用层服务对象: CMS 服务对象^[29]、NMT 服务对象^[30]、DBT 服务对象^[31]及 LMT 服务对象^[32](图 5.2)。

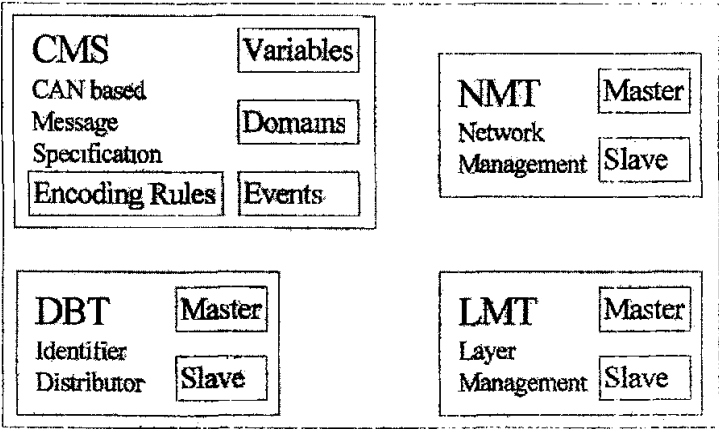


图 5.2 CAL 服务对象

Fig. 5.2 Service element of the CAN application layer

(1)CMS (CAN-based Message Specification)

CMS 提供了一个开放的、面向对象的环境, 用于实现用户的应用。CMS 提供基于变量、事件、域类型的对象, 以设计和规定一个设备(节点)的功能如何被访问(例如, 如何上载下载超过 8 字节的一组数据(域), 并且有终止传输的功能)。CMS 从 MMS (Manufacturing Message Specification) 继承而来。MMS 是 OSI 为工业设备的远程控制和监控而制定的应用层规范。

CMS 为它的消息定义了 8 个优先级, 每个优先级拥有 220 个 COB-ID, 范围从 1 到 1760。剩余的标志(0, 1761-2031)保留给 NMT, DBT 和 LMT, 如表 5.1。

(2)NMT (Network Management)

NMT 提供网络管理(如初始化、启动和停止节点, 侦测失效节点)服务。这种服务是采用主从通讯模式(所以只有一个 NMT 主节点)来实现的。

(3)DBT (DistriBuTor)

DBT 提供动态分配 CAN ID (正式名称为 COB-ID, Communication Object Identifier) 服务。这种服务是采用主从通讯模式 (所以只有一个 DBT 主节点) 来实现的。

表 5.1 映射到 CAL 服务和对象的 COB-ID (11 位 CAN 标识符)

Table 5.1 COB-ID distribution of CAL service object

COB-ID	服务或对象
0	NMT 启动/停止服务
1 — 220	CMS 对象 优先级 0
221 — 440	CMS 对象 优先级 1
441 — 660	CMS 对象 优先级 2
661 — 880	CMS 对象 优先级 3
881 — 1100	CMS 对象 优先级 4
1101 — 1320	CMS 对象 优先级 5
1321 — 1540	CMS 对象 优先级 6
1541 — 1760	CMS 对象 优先级 7
1761 — 2015	NMT 节点保护
2016 — 2031	NMT, LMT, DBT 服务

(4)LMT (Layer Management)

LMT 提供修改层参数的服务: 一个节点(LMT Master)可以设置另外一个节点(LMT Slave)的某层参数(如改变一个节点的 NMT 地址或改变 CAN 接口的位定时和波特率)。

CAL 提供了所有的网络管理服务和报文传送协议, 但并没有定义 CMS 对象的内容或者正在通讯的对象的类型, 而这正是 CANopen 切入点。

5.3 CANopen 协议

CANopen 是在 CAL 基础上开发的, 使用了 CAL 通讯和服务协议子集, 提供了分布式控制系统的一种实现方案。CAN 标准与 CANopen 协议在 OSI 模型中的关系如图 5.3。CANopen 由一系列称为子协议的文档组成:

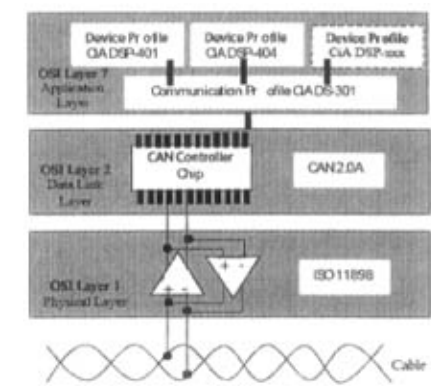


图 5.3 CAN 与 CANopen 的关系

Fig. 5.3 Relation between CAN and CANopen

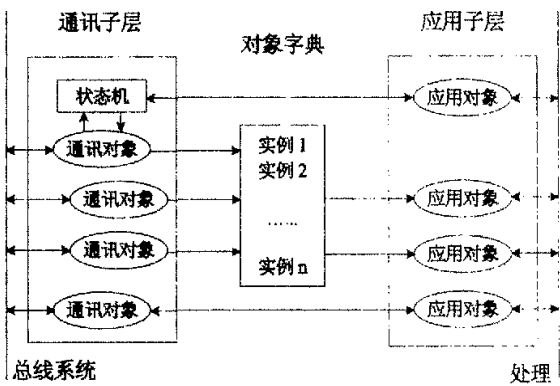


图 5.4 CANopen 设备模型

Fig. 5.4 CANopen device model

通讯子协议 (communication profile): 描述对象字典的主要形式和对象字典中的通

讯子协议区域中的对象，通讯参数。同时描述 CANopen 通讯对象。这个子协议适用于所有的 CANopen 设备。

设备子协议（device profile）：为各种不同类型设备定义对象字典中的对象。目前已有 5 种不同的设备子协议，并有几种正在发展。设备子协议为对象字典中的每个对象描述了它的功能、名字、索引和子索引、数据类型，以及这个对象是必需的还是可选的（必需的对象应该保持最少数目以减小实现的工作量），这个对象是只读、只写或者可读写等等。可选项（在通讯部分和与设备相关部分）可以根据需要增加以扩展 CANopen 设备的功能。如果需要的项超过了设备子协议中可以提供的，在设备子协议中已预留足够空间提供给厂商的特定功能使用。

CANopen 设备的组织结构如图 5.4：通讯子层接口提供了通讯对象及功能函数用以在下层传输数据；对象字典包括了所有能影响通讯接口、应用、及状态机行为的数据项；应用子层集成了与应用环境相关的设备功能描述。

CANopen 在保证网络节点互用性的同时允许节点的功能随意扩展：或简单或复杂。它的核心概念是设备对象字典（OD：Object Dictionary），通过它可以得知各节点的状态；确定网络的通讯模式；与输入输出接口相接、存储具体应用中的输入输出数据等。

5.3.1 对象字典 OD

对象字典^[33]（OD：Object Dictionary）是一个有序的对象组，每个对象采用一个 16 位的索引和一个 8 位的子索引寻址。其结构参照表 5.2，其中索引值低于 0x0FFF 的项仅仅是一些数据类型定义，一个节点的对象字典的有关范围在 0x1000 到 0x9FFF 之间。

表 5.2 CANopen 对象字典通用结构

Table 5.2 Structure of Object Dictionary

索引	对象
0000	没有使用
0001-001F	静态数据类型（标准数据类型，如 Boolean,Integer 16）
0020-003F	复杂数据类型（预定义由简单类型合成的结构如 PDOCommPar）
0040-005F	制造商规定的复杂数据类型
0060-007F	设备子协议规定的静态数据类型
0080-009F	设备子协议规定的复杂数据类型
00A0-0FFF	保留
1000-1FFF	通讯子协议区域（如设备类型，错误寄存器，支持的 PDO 数量）
2000-5FFF	制造商特定子协议区域
6000-9FFF	标准的设备子协议区域（例如：DS-401 I/O 模块设备子协议等）主要是对数据的预定义的意义，通过该协议，数据发送接收方从而知道数据的确切含义。
A000-FFFF	保留

CANopen 网络中每个节点都有一个对象字典，它包含了描述这个设备和它的网络行为的所有参数。节点本身只需要能够提供对象字典中必需的对象，以及其它可选择的、构成节点部分可配置功能的对象。对象字典中描述通讯参数部分对所有 CANopen 设备

(例如在 OD 中的对象是相同的, 对象值不必一定相同)都是一样的, 而设备相关部分对于不同类的设备是不同的。

一个设备的通讯功能、通讯对象、与设备相关的对象以及对象的缺省值由电子数据文档 (EDS: Electronic Data Sheet) 提供, 但没必要也不需要通过 CAN-bus“审问”一个节点的对象字典中的所有参数。单个设备的对象配置的描述文件称作设备配置文件 (DCF: Device Configuration File), 它和 EDS 有相同的结构^[34]。

EDS(DCF)文件必须使用 ASCII 码, 使用 ISO646 定义的字符集, 每一行以 LF(010) 字符或者 CR(013)/LF 组合字符结束, 每行的长度不得多于 255 个字符。EDS 包含数个部分, 每个部分包括相关的实体。每个部分和实体应该按以下格式:

(1)[部分名称] //[]是必须的, 且必须在顶头开始, 名字不区分大小写;

(2)keyname=value //定义每个实体的值, 它可以包含字母和数字, 后须跟数字。名称也不区分大写, 如果只包含数字, 则认为是字符串, 而不是数字。值也为字符串, 对应实体类型。

(3)注释以 (;) 在行首开始, 允许空行存在;

各个部分不区分次序, 每个部分中的实体也不区分次序。若不做详细说明, 则 DCS 中的部分和实体均是必须的。为支持将来应用, 可以采用[additional section]和[additional entries]。一致性测试工具使用警告消息来识别。

5.3.2 CANopen 通讯

CANopen 协议定义了 3 种通讯的关系模型及 4 大类通讯对象^[35], 通讯关系模型给出了通讯过程中节点的关系, 而通讯对象则说明了 CANopen 网络中通讯消息的内容和功能。三种通讯关系模型为:

(1)主/从关系模式

网络中任何时候只存在一个主节点, 其他节点为从节点。根据协议提供数据报文的应答和无应答两种类型的服务。

(2)客户/服务器关系模式

它是点对点的一种关系模型, 由客户端发送请求 (upload/download) 触发服务器端执行一定的任务。任务完成后, 服务器端发送应答进行确认。

(3)生产者/消费者 (拉/推) 关系模型

这是一对多的关系模式, 一个生产者可以对应 0 个或多个消费者, 它又对应两种服务模式: 生产者发起的 Push (推) 无应答数据报服务类型, 生产者发送数据至消费者; 消费者发起的 pull (拉) 应答数据报类型, 消费者首先发送远程帧, 生产者收到消息后, 以发送数据作为应答。

CANopen 通讯模式定义的 4 种报文 (通讯对象), 使用 COB-ID 来标识, 这与对象字典中的对象使用对象字典索引和子索引来标识不同。

(1)管理报文

层管理，网络管理和 ID 分配服务。如初始化，配置和网络管理（包括节点保护）。服务和协议符合 CAL 中的 LMT，NMT 和 DBT 服务部分。这些服务都是基于主从通讯模式，在 CAN 网络中，只能有一个 LMT，NMT 或 DBT 主节点以及一个或多个从节点。

(2)数据对象 SDO（Service Data Object）

通过使用索引和子索引（在 CAN 报文的前几个字节），SDO 使客户机能够访问设备（服务器）对象字典中的项（对象）。它通过 CAL 中多元域的 CMS 对象来实现，允许传送任何长度的数据（当数据超过 4 个字节时分拆成几个报文）。它使用的协议是确认服务类型：为每个消息生成一个应答（一个 SDO 需要两个 ID）。SDO 请求和应答报文总是包含 8 个字节（没有意义的数据长度在第一个字节中表示，第一个字节携带协议信息）。SDO 通讯有较多的协议规定。这些服务是基于客户/服务器模式。

(3)过程数据对象 PDO（Process Data Object）

用来传输实时数据，数据从一个创建者传到一个或多个接收者，即基于生产者/消费者模式。数据传送限制在 1 到 8 个字节（例如，一个 PDO 可以传输最多 64 个数字 I/O 值，或者 4 个 16 位的 AD 值）。PDO 通讯没有协议规定，PDO 数据内容只由它的 CAN-ID 定义，假定创建者和接收者知道这个 PDO 的数据内容。每个 PDO 在对象字典中用 2 个对象描述：PDO 通讯参数，包含哪个 COB-ID 将被 PDO 使用，传输类型，禁止时间和定时器周期；PDO 映射参数：包含一个对象字典中对象的列表，这些对象映射到 PDO 里，包括它们的数据长度。创建者和接收者必须知道这个映射，以解释 PDO 内容。

表 5.3 PDO 传输类型定义

Table 5.3 Transition type definition

传输类型	触发PDO的条件(B=both needed O=one or both)			PDO 传输
	SYNC	RTR	Event	
0	B	—	B	同步，非循环
1-240	O	—	—	同步，循环
241-251	—	—	—	Reserved
252	B	B	—	同步，在 RTR 之后
253	—	O	—	异步，在 RTR 之后
254	—	O	O	异步，制造商特定事件
255	—	O	O	异步设备子协议特定事件

注：SYNC 接收到 SYNC-object，RTR 接收到远程帧，Even 例如数值改变或者定时器中断，传输类型为 1 到 240 时，该数字代表两个 PDO 之间的 SYNC 对象的数目。

PDO 消息的内容是预定义的（或者在网络启动时配置的），映射应用对象到 PDO 中是在设备对象字典中描述的。如果设备（创建者和接收者）支持可变 PDO 映射，那么使用 SDO 报文可以配置 PDO 映射参数。PDO 可以有多种传送方式：同步（通过接收 SYNC 对象实现同步），包括周期和非周期同步方式。非周期：由远程帧预触发传送，或者由设备子协议中规定的对象特定事件预触发传送；周期：传送在每 1 到 240 个 SYNC 消息后触发。异步则由远程帧触发传送；由设备子协议中规定的对象特定事件触发传送。表 5.3 给出了由传输类型定义的不同 PDO 传输模式，传输类型为 PDO 通讯参数对象的

一部分，由 8 位无符号整数定义。

PDO 可以指定禁止时间，即定义两个连续 PDO 传输的最小间隔时间，避免由于高优先级信息的数据量太大，始终占据总线，而使其它优先级较低的数据无力竞争总线的问题。禁止时间由 16 位无符号整数定义，单位 100us；也可以指定一个事件定时周期，当超过定时时间后，一个 PDO 传输可以被触发（不需要触发位），事件定时周期由 16 位无符号整数定义，单位 1ms；PDO 通过 CAL 中存储事件类型的 CMS 对象实现。PDO 数据传送没有上层协议，而且 PDO 报文没有确认（一个 PDO 需要一个 CAN-ID）。每个 PDO 报文传送最多 8 个字节（64 位）数据。

(4) 预定义报文或者特殊功能对象

CANopen 还定义了三个特定用于同步，应急指示和时间标记的预定义对象，用于配合各种数据的通讯和网络管理。

① 时间标记对象（Time Stamp）

为应用设备提供公共的时间帧参考，使对时间要求苛刻的大型网络能够精确同步。它用 CAL 中存储事件类型的 CMS 对象实现。

② 同步（SYNC）对象

SYNC 由同步发生器定时广播，用于 PDO 的同步通讯，它的优先级较高，CANopen 建议用优先级较高的标识符，可以不传送数据以使报文尽可能短。

主从模式：SYNC 主节点定时发送 SYNC 对象，SYNC 从节点收到后同步执行任务；用 CAL 中基本变量类型的 CMS 对象实现。

③ 应急对象（Emergency）

应急指示对象由设备内部出现的致命错误触发，由相关应用设备以最高优先级发送到其它节点，它适用于中断型的错误报警，用 CAL 中存储事件类型的 CMS 对象实现。

除了预定义的对象之外，还有一些特殊功能对象：

① 节点/寿命保护（Node/Life guarding）

采用主从通讯模式，NMT 主节点监控节点状态，称作节点保护（Node guarding），节点也可以（可选择）监控 NMT 主节点的状态，称作寿命保护（Life guarding），当 NMT 从节点接收到 NMT 主节点发送的第一个 Node Guard 报文后启动寿命保护，其中设备的网络接口错误（不是设备自身的错误）通过应急指示报告；节点的状态通过 NMT 主节点发送远程请求到一个特定节点，根据 NMT 节点保护协议，节点给出应答来实现。

② Boot-UP

采用主从通讯模式；NMT 从节点通过发送这个报文，向 NMT 主节点说明该节点已经由初始化状态进入预操作状态。

本节小结：本节提到的 4 类通讯对象中有 2 类对象（SDO/PDO）用于数据传输，它们采用两种不同的数据传输机制实现：(1)SDO 用来在设备之间传输大的低优先级数据，典型的是用来配置 CANopen 网络上的设备；(2)PDO 用来传输 8 字节或更少数据，没有

其它协议预设，即意味着数据内容已预先定义。

5.3.3 CANopen 预定义连接集及标识符分配

为了减小简单网络的组态工作量，CANopen 定义了强制性的缺省报文标识符（CAN-ID）分配表^[36]。这些标志符在预操作状态下可用，通过动态分配还可修改他们。

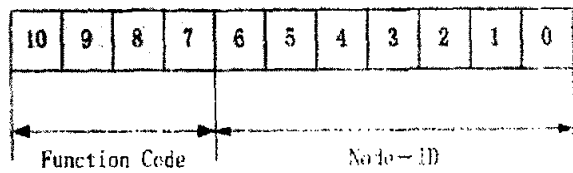


图 5.5 预定义连接集 ID 分配图

Fig. 5.5 Predefine connector ID set picture

CANopen 设备必须向它所支持的通讯对象的提供相应的标识符。缺省 ID 分配表是基于 11 位 CAN-ID 的，包含一个 4 位的功能码部分和一个 7 位的节点 ID(Node-ID)部分，如图 5.5 所示，Node-ID 由系统集成商定义，例如通过设备上的拨码开关设置。Node-ID 范围是 1~127（0 不允许被使用）。

表 5.4 CANopen 预定义主/从连接集 CAN 标识符分配表

Table 5.4 Predefine Master/Slave connector CAN-ID distribution

CANopen 预定义主/从连接集的广播对象			
对 象	功能码（ID-bits 10-7）	COB-ID	通讯参数在 OD 中的索引
NMT Module Control	0000	000H	—
SYNC	0001	080H	1005H, 1006H, 1007H
TIME SSTAMP	0010	100H	1012H, 1013H
CANopen 预定义主/从连接集的对等对象			
紧急	0001	081H—0FFH	1024H, 1015H
PDO1（发送）	0011	181H-1FFH	1800H
PDO1（接收）	0100	201H-27FH	1400H
PDO2（发送）	0101	281H-2FFH	1801H
PDO2（接收）	0110	301H-37FH	1401H
PDO3（发送）	0111	381H-3FFH	1802H
PDO3（接收）	1000	401H-47FH	1402H
PDO4（发送）	1010	481H-4FFH	1803H
PDO4（接收）	1011	501H-57FH	1403H
SDO（发送/服务器）	1100	581H-5FFH	1200H
SDO（接收/客户）	1100	601H-67FH	1200H
NMT Error Control	1110	701H-77FH	1016H-1017H

注：①PDO/SDO 发送/接收是由（slave）CAN 节点方观察的；②NMT 错误控制包括节点保护（Node Guarding），心跳报文（Heartbeat）和 Boot-up 协议报文。

预定义的连接集定义了缺省 ID 分配表如表 5.4。它包括 4 个接收 PDO（Receive—PDO），4 个发送 PDO（Transmit—PDO），1 个 SDO（占用 2 个 CAN-ID），1 个紧急

对象和 1 个节点错误控制（Node-Error-Control）ID。也支持不需确认的 NMT-Module-Control 服务，SYNC 和 Time Stamp 对象的广播。

CANopen 标识符分配与预定义的主从连接集相对应，它可使用三种不同的方法进行分配：① 使用预定义的主从连接集。ID 是缺省的，不需要配置。如果节点支持，PDO 数据内容也可以配置。② 在预操作状态下 PDO 的 ID，这可使用预定义的 SDO 在节点的对象字典中适当位置进行修改来完成。③ 使用 CAL DBT 服务：节点或从节点最初出它们的配置 ID 指称，这可由设备上的拨码开关配置，或使用 CAL LMT 服务进行配置。当网络初始化完毕，并且启动后，主节点首先通过“Connect_Remote_Node”报文（是一个 CAL NMT 服务）和每个连接的从设备建立一个对话。COB-ID 就可以用 CAL DBT 服务分配，这需要节点支持扩展的 boot-up。

5.3.4 CANopen boot-up 过程

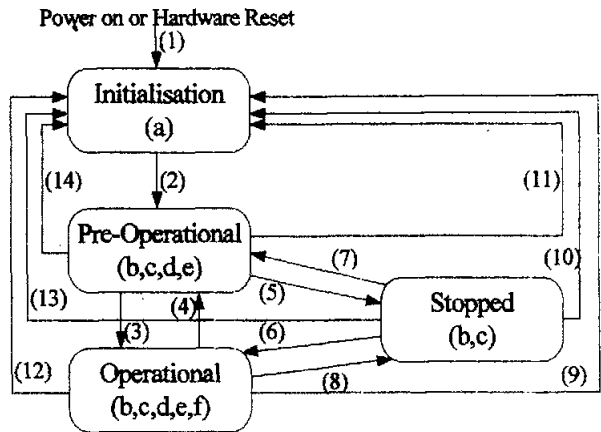


图 5.6 节点启动状态转换图

Fig. 5.6 Triger for state transition

注：(1)图中括号内的字母表示处于不同状态下可用的通讯对象：

a. Boot-up; b. NMT; c. Node Guard; d. SDO; e. Emergency; f. PDO;

(2)状态转移（1—14）由 NMT 服务发起，括号内为 NMT 命令字，如表 5.5 所示。

表 5.5 状态转移表

Table 5.5 State transfer

(1)	上电或者重启，该状态自动进入
(2)	初始化完成——自动进入预操作状态，发送 Boot-up 消息
(3)(6)	Start Remote node (0x01)
(4)(7)	Enter Pre-Operational State (0x80);
(5)(8)	Stop Remote Node (0x02)
(9)(10)(11)	Reset Node (0x81);
(12)(13)(14)	Reset Communication (0x82)

在网络初始化过程中，CANopen 支持扩展的 boot-up^[26]，也支持最小化 boot-up 过

程。扩展 boot-up 是可选的, 最小 boot-up 则必须被每个节点支持。两类节点可以在同一个网络中同时存在。如果使用 CAL 的 DBT 服务进行 ID 分配, 则节点必须支持扩展 boot-up 过程。可以用节点状态转换图表示这两种初始化过程, 如图 5.6。扩展 boot-up 的状态图在预操作和操作状态之间比最小化 boot-up 多了一些状态。

在任何时候 NMT 服务都可使所有或者部分节点进入不同的工作状态。NMT 服务的 CAN 报文由 CAN 头(COB-ID=0)和两字节数据组成; 第一个字节表示请求的服务类型(NMT command specifier), 第二个字节是节点 ID, 或者 0 (此时寻址所有节点)。仅支持最小化 boot-up 的设备叫最小能力设备, 它在设备初始化结束后自动进入预操作状态。在这个状态, 可以通过 SDO 进行参数配置和进行 COB-ID 分配。设备进入准备状态后, 除了 NMT 服务和节点保护服务(如果支持并且激活的话)外, 将停止通讯。

5.3.5 本节小结

基于 CAN 总线的 CANopen 网络通讯具有以下特点: 使用对象字典 OD (Object Dictionary) 对设备功能进行标准化的描述; 使用 ASCII 文档; 电子数据文档(EDS)和设备配置文件(DCF)对设备及其配置进行标准化的描述; CANopen 网络的数据交换和系统管理基于 CAL 中 CMS 服务; 系统 boot-up 和节点保护(Node Guarding)的标准基于 CAL 中 NMT 服务; 定义了整个系统的同步操作; 定义了节点特定的应急报文。

一个 CANopen 设备必须支持一定数量的网络管理服务(管理报文, administrative messages), 需要至少一个 SDO。每个生产或消费过程数据的设备需要至少一个 PDO。所有其它的通讯对象是可选的。一个 CANopen 设备至少应该具有(最小能力设备):

- (1)一个节点 ID;
- (2)一个对象字典(内容由设备功能决定);
- (3)一个 SDO, 能够访问对象字典中必需的对象(只读);
- (4)支持下列 NMT 从设备服务: Reset_Node, Enter_Preoperational_State, Start_Remote_Node, Stop_Remote_Node, Reset_Communication;
- (5)缺省的标识符分配。

5.4 CANopen 最小能力设备的实现

CANopen 做为一种应用层的协议, 它的功能就是屏蔽掉底层硬件对高层的可见性, 同时对更高层提供标准的服务, 所以它在不同的硬件上的实现方法是不同的, 但对上层提供的服务却是相同的。

5.4.1 物理层实现

由于本系统中 CAN 的控制器 Silicon Labs CAN 只是一个协议控制器, 不提供物理层驱动器, 所以须外接收发器来完成其信号的物理传输。本系统采用外接 VD230 芯片

作为外接收发器,进行电气转换,将逻辑信号转换为平衡差分码,以满足 CAN 物理层信号的标准,其电路连接如图 5.7。图中 VD230 的引脚 8 (Rs) 进行工作方式的选定:高速、斜率控制和待机。该端接地时,工作于高速工作方式下,发送器输出晶体管以尽可能快的速度启闭,不采取任何措施限制上升和下降斜率,建议使用屏蔽电缆以避免射频干扰问题;该端外接电阻至地时工作于斜率控制模式,此时斜率正比于引脚 8 上的电流输出,所以使用不同的电阻值可获得不同的斜率。这种模式主要应用于较低速度或较短总线长度情形下,使用限制上升和下降斜率用以降低射频干扰,从而可使用非屏蔽双绞线或平行线做总线;当引脚 8 加有高电平时,电路进入低电流待机模式,此时发送器被关掉,接收器转至低电流。若检测到显性位,通过 CAN 的控制器可将模式转为正常工作模式。引脚 5 (V_{REF}) 为参考电压,其值为 V_{CC} 的 1/2。

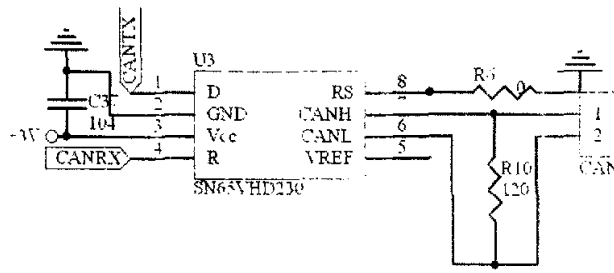


图 5.7 CAN 通讯电路连接图

Fig. 5.7 CAN field bus circuit

5.4.2 数据链路层的实现

数据链路层通过 C8051 F040 内置且符合 Bosch 规范 2.0A (基本 CAN) 和 2.0B (扩展 CAN) 的片内 CAN 控制器 Silicon Labs CAN 来实现,它包含一个 CAN 核、消息 RAM、消息处理状态机和控制寄存器。它的工作位速率可达 1M 位/秒,实际速率可能受 CAN 总线上所选择的传输数据的物理层的限制。它有 32 个消息对象,可以被配置为发送或接收数据。输入数据、消息对象及其标识掩码存储在 CAN 消息 RAM 中,它能完成所有的数据发送和接收过滤的协议处理而不用 CIP-51 干预,这就使得用于 CAN 通信的 CPU 带宽最小。CIP-51 通过特殊功能寄存器配置 CAN 控制器,读取接收到的数据和写入待发送的数据。CAN 控制器的原理框图如图 5.8 所示,CAN 核提供移位 (CANTX 和 CANRX)、消息的串/并转换及其它与协议相关的任务(如数据发送和接收过滤)。消息 RAM 可存储 32 个可以在 CAN 网络上接收和发送消息对象。CAN 寄存器和消息处理器为 CAN 控制器和 CIP-51 之间的数据传送和状态通知提供接口。初始化 CAN 控制器的一般步骤如下:(1)将 SFRPAGE 寄存器设置为 CAN0_PAGE;(2)将 CAN0CN 寄存器中的 INIT 和 CCE 位设置为 1;(3)设置位定时寄存器和 BRP 扩展寄存器中的时序参数。(设置过程见下小节);(4)初始化每个消息对象或将其 MsgVal 位设置为 NOT VALID (无效);(5)将 INIT 位清‘0’。

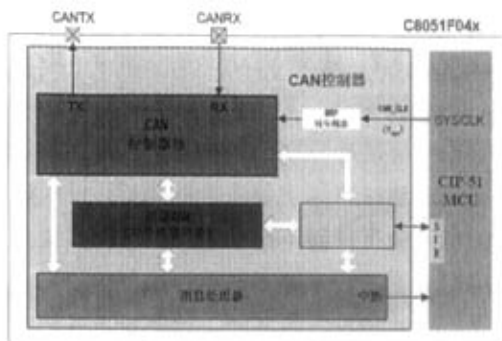


图 5.8 CAN 控制器原理图

Fig. 5.8 CAN control core schematic

5.4.2.1 Silicon Labs CAN 位率计算

要完成 CAN 网络的通讯，应首先配置 CAN 网络的通讯速率^[37]。

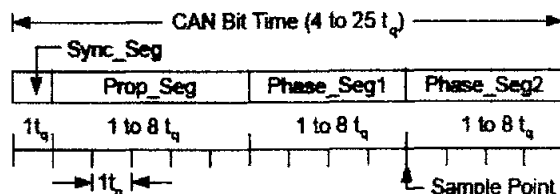


图 5.9 CAN 位时间的 4 个时段

Fig. 5.9 Four Segments of CAN Bit Time

CAN 网络物理层上每个发送位一般被分为 4 个时段，如图 5.9 所示，分别称为 *Sync_Seg*、*Prop_Seg*、*Phase_Seg1* 和 *Phase_Seg2*，这些时段之和决定 CAN 的位时间（1/位速率）。以下通过本系统中默认位速率为 1M 位/秒（位时间为 1000ns）的计算过程来说明 BRP 寄存器设置值的计算过程。这 4 个位段的和应调整至与所期望的位时间最接近，而且每个时段必须是时间量子（ t_q ）的整数倍，所以可得到最接近的位时间为 $22t_q$ （994.642ns）。*Sync_Seg* 固定为 $1t_q$ ，*Prop_Seg* 必须大于或等于 400ns 的传输延迟时间，这里选 $9t_q$ （406.899ns）。位时间中剩余的时间量子数（ t_q ）按方程(5.1)分配给 *Phase_Seg1* 和 *Phase_Seg2*，同时要遵从以下规则：若 *Phase_Seg1* + *Phase_Seg2* 为偶数，则 *Phase_Seg2* = *Phase_Seg1*，否则 *Phase_Seg2* = *Phase_Seg1* + 1；同时 *Phase_Seg2* 至少应为 $2t_q$ 。

$$Phase_Seg1 + Phase_Seg2 = \text{位时间} - (Sync_Seg + Prop_Seg2) \quad (5.1)$$

由此选 $Phase_Seg1 = 6t_q$ ， $Phase_Seg2 = 6t_q$ 。

$$SJW = \min(4, Phase_Seg1) \quad (5.2)$$

同步跳转宽度（SJW）这一时序参数由方程（5.2）定义，该参数用于确定写入到定位定时寄存器中的数值和确定所期望的振荡器误差。由于本系统使用石英晶体作为系统时钟源，所以不需要计算振荡器误差容限。所以 $SJWp = SJW - 1 = \min(4, 6) - 1 = 3$

BRP 扩展寄存器 = BRP - 1，保持其复位值 0x0000 不变，即 BRPE = 0x0000。

$TSEG1=(Prop_Seg + Phase_Seg1 - 1) = 9 + 6 - 1 = 14$

$TSEG2=(Phase_Seg2 - 1) = 5$

位时间寄存器 = $TSEG2 \times 0x1000 + TSEG1 \times 0x0100 + SJW_p \times 0x0040 + BRPE$ (5.3)

位时间寄存器值由式(5.3)计算，代入以上计算的各值得：位时间寄存器=0x5EC0

通过以上计算此时得到的位速率为 1.00539M 位/秒，这也是实际通讯过程中的位速率，与要求的 1M 之间的误差为 0.539%。通过公式 (5.6) 可计算时钟的容差：

$A = \min(Phase_Seg1, Phase_Seg2) / (2 \times (13 \times bit_time - Phase_Seg2))$ (5.4)

$B = SJW / (20 \times bit_time)$ (5.5)

容差 = $|A - B|$ (5.6)

本系统中 $A=6/(2*(13*22-6))=3/(286-6)=1.07146\%$ ， $B=2/(20*11)=1/110=0.9091\%$ ，从而得时钟的容差为 0.1623%，这对于普通晶振最低为 $10^{-5} \sim 10^{-4}$ 量级的频率精度来讲是很容易满足的。

5.4.2.2 Silicon Labs CAN 寄存器结构

CAN 寄存器地址空间为 128 个字（256 字节），消息访问器分为两组 IF1 和 IF2，其索引号和复位值如表 5.6。

表 5.6 CAN 寄存器索引号和复位值

Table 5.6 CAN register index and reset value

寄存器索引号	名称	复位值	注 释
0x00	CAN控制寄存器	0x0001	可以用CIP-51的SFR访问
0x01	状态寄存器	0x0000	可以用CIP-51的SFR访问
0x02	错误寄存器	0x0000	只读
0x03	位定时寄存器	0x2301	CAN0CN中的CCE位控制其写使能
0x04	中断寄存器	0x0000	只读
0x05	测试寄存器	0x0000	位7（RX）由CAN总线确定
0x06	BRP扩展寄存器	0x0000	CAN0CN中的TEST位控制其写使能
0x08（0x20）	IF1（IF2）命令请求	0x0001	在IF1索引号范围内（0x08～0x12），写CAN0DATL时CAN0ADR自动加1。括号内的为对应的IF2，索引号范围为0x20～0x2A，在这段地址范围内，CAN0ADR也会自动加1
0x09（0x21）	IF1（IF2）命令掩码	0x0000	
0x0A（0x22）	IF1（IF2）掩码1	0xFFFF	
0x0B（0x23）	IF1（IF2）掩码2	0xFFFF	
0x0C（0x24）	IF1（IF2）仲裁1	0x0000	
0x0D（0x25）	IF1（IF2）仲裁2	0x0000	
0x0E（0x26）	IF1（IF2）消息控制	0x0000	
0x0F（0x27）	IF1（IF2）数据A1	0x0000	
0x10（0x28）	IF1（IF2）数据A2	0x0000	
0x11（0x29）	IF1（IF2）数据B1	0x0000	
0x12（0x2A）	IF1（IF2）数据B2	0x0000	
0x40（0x41）	发送请求1（请求2）	0x0000	消息对象发送请求标志（只读）
0x48（0x49）	新数据1（新数据2）	0x0000	消息对象新数据标志（只读）
0x50（0x51）	中断标志1（标志2）	0x0000	消息对象中断请求标志（只读）
0x58（0x59）	消息有效1（有效2）	0x0000	消息对象消息有效标志（只读）

CIP-51 可以通过特殊功能寄存器直接或间接访问 CAN 控制器中的 CAN 控制寄存

器 (CAN0CN)、CAN 测试寄存器 (CAN0TST) 和 CAN 状态寄存器 (CAN0STA), 所有其它 CAN 寄存器必须通过间接索引法访问。间接访问时使用 CAN 数据寄存器 CAN0DATH 和 CAN0DATL 及 CAN 地址寄存器 CAN0ADR, 此时地址寄存器 (CAN0ADR) 的值为欲操作寄存器的索引号, 数据寄存器 (CAN0DATH 和 CAN0DATL) 为对应寄存器的值。为便于对消息对象编程, CAN0ADR 在索引范围 0x08 ~ 0x12 (接口寄存器 1) 和 0x20~0x2A (接口寄存器 2) 内有自动加 1 功能。当 CAN0ADR 中的索引号位于这两个范围之内时, CAN0ADR 在每次读/写 CAN0DATL 时自动加 1, 指向下一个 CAN 寄存器 16 位字。当配置消息对象时, 这一特性可以加快对频繁访问的接口寄存器的访问速度。

5.4.2.3 与上层的接口设计

Silicon Labs CAN 控制器有 32 个 MSG 对象, 其对象的编号越低, 优先级越高。上层通过这些 MSG 对象发送接收数据, 所以这里定义了 MSG 对象初始化方法 Initialising、数据发送 Send 及数据 Receive 接收方法, 由于 Keil C 不支持面向对象编程的结构, 本系统中采用函数的方式来完成方法的设计, 下面其讨论了部分接口的实现过程。

对于初始化方法, 发送 MSG 和接收 MSG 是不同的, 所以须分别定义, 下面给出了发送对象的初始化程序和数据发送方法的程序:

```
void init_msg_object_TX (char MsgNum, uint id)
{
    uint temp;
    SFRPAGE = CAN0_PAGE;
    CAN0ADR = IF1CMDMSK;
    //指向 IF1 Command Mask Registers
    CAN0DAT = 0x00b3;
    /*IF1 Command Mask Registers =0x00b3
    WR/RD=1, Mask=0, Arb=1, Control=1, ClrIntPnd=0,
    TxRqst=0, DataA=1, DataB=1, 一次发送 8 字节数据*/
    CAN0ADR = IF1ARB1;
    CAN0DAT = 0x0000;
    /*IF1 Arbitration Registers1 =0x0000, 即
    ID15-0=0*/
    temp=id<<2;
    //标准 id 为 ID28-ID18, 所以要左移两位
    temp&=0x1fff;
    //高三位清 0, 这是 bostchcan 控制器的要求
    temp|=0xa000; //第 15 位 msgval 置 1, 第 13
    位 direction 置 1, 定义发送
    CAN0DAT = temp;
    //地址自增, 指向 IF1 Arbitration Registers2
    /*IF1 Arbitration Registers2=101,(id)00b,
    MsgVal=1, Xtd=0, 标准模式, 扩展 ID 无效*/
    CAN0DAT = 0x0088;
    /*IF1 Message Control Registers=0x0088,
    NewDat=0, MsgLst=0, IntPnd=0, UMask=0, TxIE=0
    , RxIE=0, RmtEn=0, TxRqst=0, EoB=1, DLC3-0=100
    0, 数据长度为 8*/
    CAN0ADR = IF1CMDRQST;
    CAN0DAT = MsgNum;
    /*IF1 Command Request Registers=MsgNum
    配置信息写入指定的 MsgNum 对象*/
}
//发送方法
void transmit (char MsgNum)
{
    uchar num;
    SFRPAGE = CAN0_PAGE;
    CAN0ADR = IF1CMDMSK;
    CAN0DAT = 0x0087;
    CAN0ADR = IF1DATA1;
    for(num=0; num<8; num++){
        /*将 8 字节数据写入 IF1*/
        CAN0DATH=sdata[num];
        num++;
        CAN0DATL=sdata[num];
    }
    CAN0ADR = IF1CMDRQST;
```

```
CAN0DATL = MsgNum; //写入对应对象    }
```

而对数据接收对象来说，配置过程是相同的，只是对应的寄存器值不同而已，不再单独列出程序。

5.4.3 CANopen 协议实现

CANopen 是一个完整的应用层协议栈，对于 CANopen 网络中的节点，并不需要完全实现其所有的功能，考虑到本系统做为网络控制中的前端仪表，在 CANopen 网络中只做为数据提供节点，不须具备网络管理功能，所以只需完成其最小能力设备节点要求的功能即可实现与其他设备兼容。考虑到本系统的功能，应能发送称重的测量值，所以除 5.3.5 节所总结的最小设备应具有的功能外，应同时添加对 PDO 服务对象的支持。

(1)节点 ID

本系统中采用 P2.1—P2.7 外接跳线来确定节点的物理 ID(MAC_ID)，由于 CANopen 网络节点最多支持 127 个从设备，所以需要七位二进制码来表示其物理 ID，在 CAN 初始化的同时，创建一个字符型变量做为节点 ID 对象，该对象的属性值即是变量存储的 ID 值，与此属性相关而定义方法 ReadID() 和 UpdataID()。updataID() 用于从 FLASH 中读取最近保存的值，它可以由 SDO 进行配置，保存地址为临时存储器区的 0x04。而每次通过跳线设置完新的 MAC_ID 后，由键盘操作菜单 F41 触发 readID() 方法进行读取新属性值写入临时存储器。

(2)对象字典

对象字典其实是一个节点所用到的全部参数及其值的集合，本系统中通讯子协议设备字典的一部分如表 5.7 所示。

表 5.7 设备字典

Table 5.7 Device Object Dictionary

索引	子索引	对象类别	名称	数据类型	配置值	访问
1000		VAR	Device type	Unsigned32	0x01040191	只读
1001		VAR	Error register	Unsigned8		只读
1018		RECORD	Identity Object	Identity(23h)		只读
Sub (23h)	0x00	VAR	Number of entries	Unsigned8	1	只读
	0x01	VAR	Vendor ID	Unsigned32	0x02	只读
1800		RECORD	1 st TPDOparameter	PDO CommPar(20h)		只读
Sub (20h)	0x00	VAR	Numberofentries	Unsigned8	3	只读
	0x01	VAR	COB-ID	Unsigned32	0x181+macid	只读
	0x02	VAR	Transmission type	Unsigned8	253	只读
	0x03	VAR	Inhibit time	Unsigned16	2	只读
1A00		RECORD	1 st TPDOMapping	PDO Mapping(21h)		只读
Sub (21h)	0x00	VAR	Numberofentries	Unsigned8	1	只读
	0x01	VAR	1 st objecttobemapped	Unsigned32	0x64000120	只读

注：①devicetype=0x01040191;对于该值为对设备的一个标识，确定后就是不变的，所以把直接保存在代码区，其低 32 位值表示设备配置的版本号 0x0191(401)，高 32 位为附加的信息 0x0104(仅有

模拟信号输入的设备); ②errorregister 错误寄存器结构如下: 第 0 位, Generic error; 第 1 位, Current; 第 2 位, Voltage; 第 3 位, Temperature; 第 4 位, Communication error(overflow, error state); 第 5 位, Device profile specific; 第 6 位, Reserved(always 0); 第 7 位, Manufacturer specific; ③Vendor ID 值为制造商的对该设备的描述编号; ④COB-ID 的设置采用预定义连接集的定义方法, Transmission type 设置为远程帧触发模式; ⑤1stobjectto bemapped=0x64000120 表示映射对象为第一组 32 位模拟量输入

表中对象类型为 VAR(code 为 7)的其数据类型可用基本变量类型实现, 对 RECORD 记录类型(code 为 9), 可以数据类型中括号内的项做索引查找对象字典 0x0001—0x0FFF 区得到变量类型的精确定义。由于本系统中只是为了实现 CANopen 协议下的最小能力设备, 因此对象字典采用了结构体 OBJSTU{ulong index;union udatatype {uchar:uint;ulong} datavalue;} objdictionary[]数组的方式, 来对任一属性值进行保存和引用。由于本系统中所用到的属性值的类型只有三种, 所以结构体中使用了一个联合体结构 udatatype 创建的 udatatype 类型, 同时为该数组结构创建了 objdic_find(ulong objindex)方法, 该方法通过对象字典中的索引值来返回对象字典数组中的索引值, 从而把对对象字典根据对象字典索引的引用方式转化成对其数组按数组下标进行引用的方式, 这样其属性值的访问就可面向对象编程中常用的点访问方式了。

(3)NMT 从设备服务的支持及 Boot-up

对于 CANopen 网络中的节点来说, 加电后其任一时刻的状态只能为 Initialising, Pre-operation, Operational 及 Stopped 四种状态之中的一种, 所以可采用定义一个枚举型的公有变量来表示节点的当前状态值。对于最小能力设备, 节点 Initialising 完成后自动

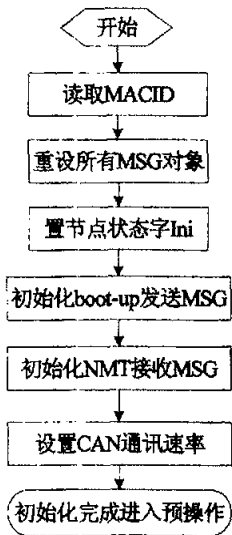


图 5.10 CAN 初始化框图

Fig. 5.10 CAN initialise diagram

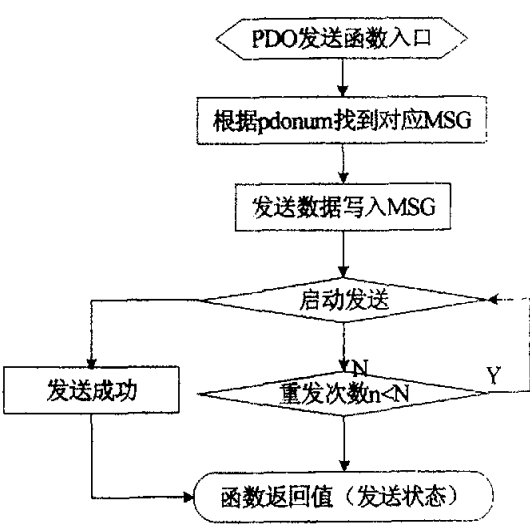


图 5.11 PDO 发送框图

Fig. 5.11 PDO transmit diagram

进入 Pre-operation, 此时节点须发送 boot-up 报文以通知网络上的 NMT 服务节点这一行为, 预定义连接集中这一报文所占的优先级最低, 所以选用控制器的 MSG32 来发送此消息。而对于 NMT 模块控制服务, 由于此服务消息不需要应答, 考虑其高优先级, 所

以采用 MSG1 接收 NMT 控制信息。根据 NMT 报文消息的协议规则, MSG1 应配置对所有 COB-ID 为 0x000 的数据进行接收, 然后根据 Node-ID 值及对应的命令字通知节点做出状态调整, 图 5.10 给出了 CAN 初始化程序的流程图。

节点初始化完成后直接进入 preoperational 状态, 进入此状态后, 更改状态变量的值, 打开中断, 并发送 boot-up 消息。发送 boot-up 消息只需调用 MSG32 的发送方法即可, 报文内容为 1 字节长的 0 值。中断开启后, 单片机就可以处理 MSG1 接收到的消息, 调用 nmtmessage_process() 函数, 判断 byte1 是否为本节点的 Node-ID 值, 然后根据协议规定的 byte0 的值调用相应的函数来进行状态切换。

(4)PDO 服务对象

PDO 务对象根据预定义连接集的 ID 分配规则, 结合本系统实际, 选 MSG5 来进行完成 PDO 服务功能, 它的初始化应结合对象字典中与 PDO 相关的参数设置值。定义 PDO 的结构体如下, 通过声明新变量做为该对象的实例:

```
struct pdo
{
    unsigned int pdonum;//该属性值范围为{1—512}, 按预定义连接集规定应为 181H+MACID
    unsigned char pdotype;//{生产者, 消费者}之一, 应该为生产者
    unsigned char datatype;//为 32b, float 型, 由上节对象字典相关值确定
    unsigned inhibittm;//5, 由上节对象字典确定
    unsigned char msgnum;
    bit isnewdata;//判断是否新值
    unsigned char pdodata[8];
    unsigned char datalen;//有效数据字节长
} xdata pdoarray[MAX]//这里 MAX 为 1
```

pdoarray 的赋值应在 Pre-operational 子函数内完成, 对于可支持远程配置的, 可在此状态内通过 SDO 服务对象来进行参数更改, 参数更改完成后, 可调用 MSG5 的初始化方法, 这里应注意初始化完成后, 应先暂停该 MSG 对象的激活(设置 MsgVal 位为 0), 因为在此节点状态内规定 PDO 服务是不可用的, 或者只修改参数值, 到 Operational 状态下再进行 MSG5 的初始化。这里状态切换不是自动完成的, 而是通过上一小节提到的 MSG1 接收到的命令调用函数来完成的。由于本系统只是做为 PDO 生产者存在于网络中的, 所以应定义 PDO 的发送方法, 即提供 WritePDO() 方法, 该方法按要求有两个参数 pdonum 及要发送的数据, 返回结果为字符型值, 用以判断是否发送成功。图 5.11 给出了 PDO 发送方法的程序流程图。

(5)SDO 服务对象

SDO 通讯模型为客户/服务器的应答模式, 既要求从节点能接收客户端消息的同时, 也能发送节点的包含本地字典参数值的应答消息, 所以它与一个 PDO 只占用一个 MSG 对象不同, 它必需用两个 MSG 对象来完成服务功能。根据预定义连接集, 可确定从节点接收消息的 cob-id 为 580h+MACID, 而发送消息所占用的 cob-id 为 600h+MACID。按优先级的分配, 本系统中分别分配 MSG29 和 MSG30 作为接收/发送的消息对象。由于 SDO 在节点 Pre-operation 状态下就应该是可用的, 所以初始化这两个 MSG 对象的子

函数应在 Enter_Preoperational_State() 中调用, 从而从节点发送 boot-up 完消息后, 就可以做为服务器端接受客户端的连接要求, 对消息请求按照 SDO 的协议规则做出应答。

由于本系统的数据存储器较少, 所以对对象字典中的值均保存入程序存储器, 从而 SDO 客户节点不能完成对从节点的写 (download) 操作, 而只回答读 (upload) 操作的申请。MSG29 的接收中断服务程序把符合过滤条件 (cob-id) 的消息接收到的接收缓冲区, 并转入协议处理函数 sdo_upload_response, 本系统中 SDO 的传送机制采用的是加速传送机制, 即每个 SDO 消息报文中仅包含 4 字节长的有效数据。程序部分代码如下:

```
void sdo_upload_response()
{
    unsigned long objindex;
    unsigned char objarrindex,i,*p;
    if ((rdata[0]<=0x5F)&(rdata[0]>=0x40))
    objindex=(rdata[1]*256+rdata[2])*256+rdata[3];
    //根据客户端的启动域上传请求计算索引值
    objarrindex=objdic_find(objindex);
    //查找数组中索引值对应对象字典索引的元素
    tdata[0]=0x43+(sizeof(objdictionary[objarrindex].datavalue)+4)<<2;//协议规定
    tdata[1]=rdata[1];      tdata[2]=rdata[2];
    tdata[3]=rdata[3];//byte1-3 按接收报文返回
    p=&objdictionary[objarrindex].datavalue;
    //指向属性值地址
    for(i=0;i<sizeof(objdictionary[objarrindex].datavalue);i++) tdata[i+4]=*p++;
    //把属性值移入到发送缓冲区的后四个字节
    data_transmit(30);
    //用 MSG30 发送发送缓冲区的值
    end if
    ... //继续其它处理
}
```

本节以前面所做的称重仪表为对象, 详细讨论了如何实现 CANopen 网络协议规定下的最小能力设备, 尽管这一设备未进行兼容性测试, 但是在自己组建的两个节点的网络中测试正常, 其中另一节点扮演了主节点的角色, 它向仪表节点发送模拟的协议报文。

第六章 误差分析

6.1 误差来源及处理

称重系统工作过程中,各个环节都会受到很多种干扰,而各种都会影响最终的测量精度。下面给出了本系统设计过程中考虑到的误差来源^[38]及相应的处理技术。

(1) 电源部分

电源的稳定性能不好,精度不高,整流滤波性能不好都会引起电源纹波系数增加,纹波电压幅度升高,从而引入测量系统中常见的工频干扰。而且在既有模拟电路又有数字电路的系统中,若这两者的电压隔离性能不好,而引起的模数电压之间的相互串扰,也会降低测量系统的精度。本系统设计过程中,对模拟电路及数字电路部分采用了分离的电源整流模块,同时使用 AD584 产生高精密度参考电压对传感器提供激励电源,并在软件上使用了算术滤波方法以减小工频干扰对测量结果的影响,提高测量精度。

(2) 外界杂波信号

在热工综合实验系统中,变频器及水泵通过电源等的耦合会给称重仪表带来干扰,同时其产生的电磁辐射干扰也较为明显,前一部分的干扰主要通过电源部分的抗干扰措施来抑制,对后一部分的干扰,本系统在设计过程中采用了静电屏蔽的方法,通过在主要的信号线两侧加接地的屏蔽线,同时在电路板的空余部分使用覆铜接地来实现屏蔽。

(3) 系统的非线性

失调电压、温电效应以及传感器本身的非线性等都会引起整个系统的非线性畸变。针对这一误差来源,系统设计过程中除了选取线性度好的优良器件外,同时采用了放大电路的非线性补偿,以及输出结果的分段线性化技术来提高线性度。

以上给出了设计过程中主要考虑到的误差来源及相应的处理技术,而对于象算术平均滤波法对其它干扰也会产生一定的抑制作用,而不是单纯的仅仅局限于对所提到的干扰进行抑制。

6.2 精度分析

测量的误差有很多的来源,主要有:测量装置误差,环境误差,方法误差,人员误差。按照误差的特点和性质,误差分为系统误差、随机误差和粗大误差^[39]。在同一条件下,多次测量同一量值时,绝对值和符号保持不变,或在条件改变时,按一定规律变化的误差为系统误差,而对绝对值和符号以不可预定方式变化着的误差称为随机误差。超出在规定条件下预期的误差称为粗大误差,或称“寄生误差”,如过失性误差等。考虑到本系统的实际情况,无法对各个环节进行测量并进行误差分析,所以总的系统误差分析只是对各个环节按器件提供的误差进行综合。影响测量精度的主要有:传感器的精度,

激励电源的稳定度 (AD584)，放大电路的线性度，ADC 的转换精度。

(1) 传感器的精度

传感器的精度为 0.05%；

(2) 电源的稳定度误差

AD584，在本系统中工作时，其输出电流一定，因为负载是确定的，所以可以不考虑负载变化对其输出电压稳定度的影响，而其温度系数为 5ppm/°C (百万分率，简称 ppm, Parts Per Million, 定义为百万分之一，1ppm 即是一百万分之一)，即 $5\mu\text{V}/^\circ\text{C}$ ，按 100°C 的变化范围，其温漂也只有 0.5mV，其稳定度误差则为 $0.5 \times 10^{-3} / 7.5 = 0.007\%$ 。

(3) 放大器件的非线性误差

放大器件的非线性，大约在 $0.97\% < 1\%$ ^[17]，这里主要考虑了前级放大电路，由于前级放大电路放大倍数在 50 倍左右，其非线性不可忽视，而对后级放大电路其放大倍数只有 2.5 倍左右，可忽略其非线性。对前级放大电路，非线性造成的放大倍数误差为 $50 \times 1\% = 0.5$ 倍，则由非线性带来的最大输出误码差电压为 (满量程时输入为 15mV) $15\text{mV} \times 0.5 = 7.5\text{mV}$ ，由此带来的整个放大电路的相对误差为： $7.5 \times 10^{-3} / 2 = 0.375\%$ 。而考虑到输出过程中采用的分段线性化处理对它的补偿，理论上若采用整个量程内平均间隔标定时，此误差可缩减为原来的 1/段数，当段数为 9 时，则其误差降为 $0.375\% / 9 = 0.042\%$ 。

(4) ADC 的转换精度

ADC 的转换精度：C8051F040 的 ADC 转换器是 12 位，而且本系统中采用的过采样和求均值方法可以确保在该精度之上，12 位的 ADC 转换器的量化误差为 $1/4095 \approx 0.025\%$ 。

总的误差估计

以上各环节的误差可看成是互不相关即相互独立的，所以用方和根方法进行对以上各环节误差进行合成^[41]，以得到总的系统误差：

$$\delta = \sqrt{\sum_{i=1}^n \delta_i^2} = \sqrt{\delta_1^2 + \delta_2^2 + \delta_3^2 + \delta_4^2} \quad (6.1)$$

式中 δ_1 —传感器的精度； δ_2 —电源的稳定度误差； δ_3 —放大器件的非线性误差； δ_4 —ADC 的转换精度；

把以上各项的估计误差带入式 (6.1) 计算得总的合成误差：

$$\delta = \sqrt{\delta_1^2 + \delta_2^2 + \delta_3^2 + \delta_4^2} = \sqrt{(0.05\%)^2 + (0.007\%)^2 + (0.042\%)^2 + (0.025\%)^2} \quad (6.2)$$

$$\delta = \sqrt{49.38 \times 10^{-4}} = 0.0703\% \quad (6.3)$$

式 (6.3) 所得值 0.063% 即为测量系统的总理论误差，由于受时间及条件所限，这次开发的数字称重仪没有进行标定，所以其精度也只是进行了估计。

第七章 结论与展望

本文对设计中所采用的压阻应变式传感器及网络模型进行了系统阐述,同时给出了小信号放大电路及非线性补偿,软件滤波,分段线性化输出,串口通讯以及 CANopen 协议的实现过程。本文所开发的在线数字称重仪,与传统的变送器相比,该仪表准确度高,功耗低,电路更简捷,功能更加完善,而且其模块化的结构更加易于检修及更换,而增加的串行口通讯,高速的总线通信接口,及对 CANopen 协议的支持,使得它的应用前景非常广阔。其主要特点有:

(1)模块化的系统结构。本文所开发的在线数字称重仪其软硬件上均采用了模块化的系统结构。硬件系统的模块化设计,不但易于检修及更换,同是也为系统将来的部分模块升级提供了方便,如显示模块、放大电路模块等。模块化的思想同样贯穿在了软件的设计过程中,整个软件结构分成几个部分,每个部分的程序完成一定的功能,而主程序通过调用相应的函数即完成系统整合,这使得主程序的控制流程更加简洁明了,相应的编程实现也更加容易,这一结构同时也为以后的软件更新及再开发提供了便利。

(2)多种提高精度的方法综合运用。由于称重仪表内部采用了微处理器,因此设计过程中除了采用硬件电路滤波外,在软件系统中同时使用了算术平均的数字滤波方法来进行提高测量精度;而针对非线性的处理,则是在硬件电路中采用了放大电路的非线性补偿,软件上使用分段线性化的处理方法以提高实际值与测量值的拟合程度。

(3)丰富的网络接口。本系统除了提供可与计算机直接相连的 RS-232 通信模式外,还提供了高速的 CAN 总线接口,并同时提供了对 CANopen 协议的支持,从而可使得本系除了可直接与计算机相连实现在线测量外,还可做为 CAN 网络中的一个节点完成测量功能。

现场总线的出现,使自动化仪表在产品体系结构、功能结构等方面产生了较大的变革,采用数字式传输技术的现场总线智能仪表正成为自动化领域发展的热点。本文选取了一种已获得巨大发展应用的 CAN 现场总线技术,对现场总线智能设备的开发进行了有益的尝试和探索。本文的结合 OSI 参考模型对 CAN 网络模型进行系统描述,然后切入到其中的一层应用层、具体的一种协议 CANopen 协议进行研究,并进行对 CANopen 协议下最小能力设备进行设计实现的这一细致的过程对于开发其它总线协议产品同样具有借鉴意义。同时作者认为仍存在以下问题待今后进一步完善:

分段线性化输出功能对于能进行整个量程内完整标定的条件下较为合适,而对于只能小量程内进行标定的条件下,应采取最小二乘法或其它更好的方法进行直线的拟合,以使得单次标定的误差对测量结果产生最小的影响,这可通过在未来的设计中加置相应的模块功能支持来实现。串行通讯口程序完成了可视化的参数标定功能及在线测量功能,而其采用的通讯规则主要借鉴了 CAN 的固定帧长的结构,与其简单的功能相比,

该规则下做数据通讯时的数据传输效率不是很高，在未来的设计中，可以针对此局部做出相应的调整。另外由于受条件所限，CANopen 设备的兼容性测试以及整个系统的标定还没有进行。

现场总线智能仪表的优点是不容置疑的，研制国产的现场总线仪表不但可以节省大量的外汇，充分展现现场总线控制系统的优越性，而且对现场总线技术在国内的推广及应用也将起到巨大的推动作用。

参考文献

1. 田武. 处理称重仪器的新技术[J], 现代计量测试, 1994, 6: 60-61
2. 许福国. 现状与发展——观 96 中国衡器展览会[J], 衡器, 1996, 6: 35-36
3. 魏恒, 盛秋琴, 张婷. 光纤应力传感技术研究[J], 光电子技术, 2005, 25 (1): 28-34
4. 张广川, 陈小枫, 马小永. 现场总线智能仪表的整体解决方案[J], 现代制造, 2005, 27: 56-57
5. 叶煌. 基于 WorldFIP 现场总线的智能差压变送器研发 (D), 保定: 华北电力大学, 2005
6. 夏德海. 现场总线的现状及其应用[J], 现场总线与 PLC 应用论文选, 1999, 4 (现场总线部分), 1999: 12-13
7. 卜云峰. 检测技术[M], 北京: 机械工业出版社, 2005, 14-17
8. 苏铁力, 关振海, 孙继红. 传感器及其接口技术[M], 北京: 中国石化出版社, 1998, 18-21
9. Andrew S. Tanenbaum 著, 熊桂喜, 王小虎译. 计算机网络[M], 北京: 清华大学出版社, 2003, 13-28, 368-400
10. 新华龙公司. C8051 F040 User's Guide[J], 深圳新华龙公司网站 <http://www.xhl.com.cn/xhl/downfile/e-c8051f04x.pdf>: 20-30
11. 康宝祥, 栾贵兴. RISC 体系结构及其实现技术[M], 北京: 科学出版社, 1996, 12-17, 172-186
12. 周立功等. ARM 嵌入式系统基础教程[M], 北京: 北京航空航天大学出版社, 2005, 5-9
13. 浙江余姚太平洋自控有限公司. BJH-1 补偿线盒数据手册[J], 浙江余姚: 浙江余姚公司, 2004: 1-6
14. 黄晨, 汪贵平. 带隙型集成电压基准源的原理及应用[J], 西安公路学院院报, 1994, 14 (2): 148-151
15. MAXIM. MAXIM Chopper-Stabilized Op Amps ICL7650/ICL7650B[J], 21IC 网站 <http://www.21icsearch.com/searchpdf/maxim/ICL7650-ICL7650B.pdf>: 1-12
16. 吴祖国. ICL7650 斩波稳零运算放大器的原理及应用[J], 国外电子元器件, 2003, 4: 41-42
17. 任志山, 黄春耀. 高精度、高稳定度微弱信号放大器的设计[J], 漳州师范学院学报 (自然科学版), 2003, 16 (3): 42-45
18. 朱立忠, 冯丹. PC 机与 MSC51 单片机串行通信接口电路的设计[J], 沈阳工业学院

- 学报, 2003, 22 (2) : 22-23
19. Sipex Corporation. SO3223E/3243E Datasheet[J], <http://www.21icsearch.com/searchpdf/other/SP3223.pdf>: 1-19
20. 潘琢金, 施国君. C8051FXXX 高带 SOC 单片机原理及应用[M], 北京: 北京航空航天大学出版社, 2002, 222-225
21. 童长飞. C8051F 系列单片机开发与 C 语言编程[M], 北京: 北京航空航天大学出版社, 2005, 173-175
22. Cygnal Integrated Products Inc (潘琢金, 孙德龙, 夏秀峰译). C8051F 单片机应用解析[M], 北京: 北京航空航天大学出版社, 2002, 260-277
23. 郭凌. 称重仪表常用数字滤波技术[J], 衡器, 1998, 27 (4) : 20-22
24. 微软(中国)有限公司/Microsoft Corporation. Visual Basic 6.0 中文版程序员指南 (Microsoft Visual Basic 6.0 Programmer's Guide), 北京: 北京希望电脑公司, 1998, 10-14, 667-669
25. 李怀明, 骆原, 王育新. Visual Basic 6.0 中文版参考详解[M], 北京: 清华大学出版社, 1999.11, 495-496
26. CiA Draft Standard 301. CANopen Application Layer and communication Profile[J], CiA: 2002, V4.0, 14: 23-86, 74-77
27. 郭宽明. CAN 总线原理和应用系统设计[M], 北京: 北京航空航天大学出版社, 1996, 241-256, 38-56
28. 王桂荣, 钱剑敏. CAN 总线和基于 CAN 总线的高层协议[J], 计算机测量与控制, 2003, 11 (5) : 391-395
29. CiA DS202. CMS Service Specification[J], CAN Application Layer for Industrial Applications, 1996, Version1.1: 18-24
30. CiA DS203. NMT Service Specification[J], CAN Application Layer for Industrial Applications, 1996, Version1.1: 3-20
31. CiA DS204. DBT Service Specification[J], CAN Application Layer for Industrial Applications, 1996, Version1.1: 2-13
32. CiA DS202. LMT Service Specification[J], CAN Application Layer for Industrial Applications, 1996, Version1.1: 2-9
33. 吴爱国, 刘莉. CAN 总线控制系统的应用层协议 CANopen 剖析[J], 微计算机信息, 2003, 19 (3) : 27-29
34. CiA Draft Standard 306. Electrocnic data sheet specification for CANopen[M], CiA: 2005, V1.3, 8-9
35. 何光宇, 胡正, 秦霆镐, 王健. 针对工业控制的 CANopen 系统[M], 微计算机信息, 2003, 19 (12) : 5-6

36. 周立功单片机有限公司. CANopen 协议中文简介[J], 广州周立功公司网站转自荷兰 NIKHEF 公司网站 H.boterebrood 的 CANopen: high-level protocol for CAN-bus, <http://www.zlgmcu.com/philips/can/about/CANopenxieyi.pdf>: 6-8, 9-16
37. Robert Bosch GmbH. C_CAN User's Manual[J], http://www.xhl.com.cn/xhl/download/Bosch_CAN_User's_Guide.pdf: 13-14
38. 金篆芷, 王明时. 现代传感器技术[M], 北京: 电子工业出版社, 1995, 370-373
39. 费业泰. 误差理论与数据处理[M], 北京: 机械工业出版社, 2003, 4-5, 67
40. 瞿明, 刘文彦, 邓新中. 测试系统中干扰及形成机理[J], 自动化仪表, 2003, 24 (6): 20-21
41. 王立吉. 计量学基础[M], 北京: 中国计量出版社, 1997, 90-91

致 谢

值此硕士论文完成之际，首先感谢我的导师李新光副教授，我的硕士论文能够顺利完成离不开导师的悉心指导和严格要求。导师广博的知识、严谨的治学态度和兢兢业业的工作精神，给我留下了深刻的印象，并将使我终身受益。攻读硕士学位期间参加的科研活动，使我学到了不少新的知识，同时提高了自身的科研能力，在此，再向导师表示衷心的感谢。

同时要感谢邵富群教授及仪表研究所的其他老师们给我的很多指导以及律德才博士，杜丽娜师姐、杨梅影、魏薇、王丽娜、郭圆圆、许明英、王淮昆及缠彦名等同实验室的不同届的硕士同学们以及天津大学的赵艳硕士等给予我的帮助！此外还要感谢检测专业的高德才、丁贺平、李小兵等其他所有的同学，他们良好的求学作风让我受益匪浅，生活中融洽的相处给我提供了一个很好的研究环境。

感谢控制理论与控制工程的汤健硕士等舍友们，以及同届的强军计划的同学们。

感谢在我求学期间给予我莫大的帮助和支持的家人们。

再次向所有关心和帮助我的老师、同学、朋友及家人致以最诚挚的谢意！

李 茂 涛

2006年2月于东北大学

作者: [李茂涛](#)
学位授予单位: [东北大学](#)

参考文献(42条)

1. [田武](#) [处理称重仪器的新技术](#) 1994(06)
2. [许福国](#) [现状与发展—观96中国衡器展览会](#) 1996(06)
3. [魏崐](#), [盛秋琴](#), [张婷](#) [光纤应力传感技术研究](#)[期刊论文]-[光电子技术](#) 2005(1)
4. [张广川](#), [陈小枫](#), [马小永](#) [现场总线智能仪表的整体解决方案](#) 2005(27)
5. [叶煌](#) [基于WorldFIP现场总线的智能差压变送器研发](#) 2005
6. [夏德海](#) [现场总线的现状及其应用](#) 1999
7. [卜云峰](#) [检测技术](#) 2005
8. [苏铁力](#), [关振海](#), [孙继红](#), [孙彦卿](#) [传感器及其接口技术](#) 1998
9. [Andrew S Tanenbaum](#), [熊桂喜](#), [王小虎](#), [李学农](#) [计算机网络](#) 2003
10. [新华龙公司](#) [C8051 F040 User's Guide](#)
11. [康宝祥](#), [栾贵兴](#) [RISC体系结构及其实现技术](#) 1996
12. [周立功](#) [ARM嵌入式系统基础教程](#) 2005
13. [浙江余姚太平洋自控有限公司](#) [BJH-1补偿线盒数据手册](#) 2004
14. [黄晨](#), [汪贵平](#) [带隙型集成电压基准源的原理及应用](#) 1994(02)
15. [MAXIM](#) [MAXIM Chopper-Stabilized Op Amps ICL7650/ICL7650B](#)
16. [吴祖国](#) [ICL7650斩波稳零运算放大器的原理及应用](#)[期刊论文]-[国外电子元器件](#) 2003(4)
17. [任志山](#), [黄春耀](#) [高精度、高稳定度微弱信号放大器的设计](#)[期刊论文]-[漳州师范学院学报\(自然科学版\)](#) 2003(3)
18. [朱立忠](#), [冯丹](#) [PC机与MCS51单片机串行通信接口电路的设计](#)[期刊论文]-[沈阳工业学院学报](#) 2003(2)
19. [Sipex Corporation](#) [S03223E/3243E Datasheet](#)
20. [潘琢金](#), [施国君](#) [C8051FXXX高带SOC单片机原理及应用](#) 2002
21. [童长飞](#) [C8051F系列单片机开发与C语言编程](#) 2005
22. [Cygnal Integrated Products Inc.](#), [潘琢金](#), [孙德龙](#), [夏秀峰](#) [C8051F单片机应用解析](#) 2002
23. [郭凌](#) [称重仪表常用数字滤波技术](#) 1998(04)
24. [微软\(中国\)有限公司](#) [Visual Basic 6.0中文版程序员指南](#) 1998
25. [李怀明](#), [骆原](#), [王育新](#) [Visual Basic 6.0中文版参考详解](#) 1999
26. [CiA Draft Standard 301](#).CANopen Application Layer and communication Profile 2002(16)
27. [邬宽明](#) [CAN总线原理和应用系统设计](#) 1996
28. [王桂荣](#), [钱剑敏](#) [CAN总线和基于CAN总线的高层协议](#)[期刊论文]-[计算机测量与控制](#) 2003(5)
29. [CiA DS202](#).CMS Service Specification 1996
30. [CiA DS203](#).NMT Service Specification 1996
31. [CiA DS204](#).DBT Service Specification 1996
32. [CiA DS202](#).LMT Service Specification 1996
33. [吴爱国](#), [刘莉](#) [CAN总线控制系统的应用层协议CANopen剖析](#)[期刊论文]-[微计算机信息\(测控仪表自动化\)](#) 2003(3)
34. [CiA Draft Standard 306](#).Electrocnic data sheet specification for CANopen 2005
35. [何光宇](#), [胡正](#), [秦霆镐](#), [王健](#) [针对工业控制的Canopen系统](#)[期刊论文]-[微计算机信息\(测控仪表自动化\)](#) 2003(12)

36. [周立功单片机有限公司 CANopen协议中文简介](#)
37. [H boterebrood CANopen:high-level protocol for CAN-bus](#)
38. [Robert Bosch, Gmb H C_CAN User's Manual](#)
39. [金篆芷, 王明时 现代传感器技术](#) 1995
40. [费业泰 误差理论与数据处理](#) 2003
41. [瞿翌, 刘文彦, 邓新中 测试系统中干扰及其形成机理](#)[期刊论文]-[自动化仪表](#) 2003(6)
42. [王立吉 计量学基础](#) 1997

相似文献(2条)

1. 期刊论文 [白生威, 何全茂 高密度介质自动补给装置磁铁粉称重仪的设计](#) -[工矿自动化](#)2009, 35(11)
文章介绍了高密度介质自动补给装置磁铁粉称重仪的硬件结构及工作原理, 详细阐述了称重仪压力传感器和速度传感器的选型, 给出了流量在线测量的实现方法. 该称重仪采用称重传感器测量胶带输送机上的物料重量, 并以AT89C52单片机为计算和控制的核心元件, 对传感器送来的信号进行分析、控制、处理和驱动显示, 实现了重量的在线动态测量和控制.
2. 会议论文 [施昌彦 称重测力仪表的发展动态](#) 1995
该文从动态测量、自动在线测重及模型化测量三个方面, 阐述了称重测力仪表和装置的近期发展动态. 简要介绍了动态称重测力的特征及研究方法, 叙述了自动在线测量的必要性及常见的自动秤, 并指出模型化测量的意义及开发软件技术的重要性.

本文链接: http://d.g.wanfangdata.com.cn/Thesis_Y852686.aspx

下载时间: 2010年5月29日