

# 智能双显张力仪的研制

## 中文提要

在纺织工业中，纱线的张力是一项重要的工艺参数。张力仪则是检测这一参数的常用手段。然而目前生产中大多应用的传统张力仪存在着诸多不足之处。针对这一情况设计了智能型的双显张力仪，以取代一些传统以及高价进口张力仪。

此仪表的最大特色在于能够同时显示张力值以及反映张力变化情况的模拟光指针，同时具备了数字化的直观性和指针式的可视性。

本文对该仪表的原理、软硬件结构等部分作了较为详尽的阐述，同时与国外先进的同类型仪表进行了测试对比，试验结果证明了本仪表的合理性和可靠性。

**关键词：**张力仪 光指针 单片机

作 者：柏松乔

指导教师：陈庆官

## The Research Of Dual-display Tensionmeter

### Abstract

In textile , the tension is an important parameter ,which the main means to measure is the tensionmeter .While , much more insufficients existing in the traditional tensionmeters. Therefore, the new intelligent dual-display tensionmeter is researched .

Displaying the data with light-indicator together is the super specialty, which possesses both the ocular of the data and the visuality of the indicator .

In this paper , the theories of tensionmeter and the structure are talked particalarly. Simultaneitily ,contrasted with the other instrument ,which is excellence in the fremdness, and indicated by the result, the tensionmeter is reliability and exact .

**Keywords:** tension light-indicated siglechip

Written by: Bai Songqiao

Surpervised by: Chen Qingguan

# 苏州大学学位论文独创性声明及使用授权的声明

## 学位论文独创性声明

本人郑重声明：所提交的学位论文是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含其他个人或集体已经发表或撰写过的研究成果，也不含为获得苏州大学或其它教育机构的学位证书而使用过的材料。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人承担本声明的法律责任。

研究生签名： 相松乔 日期： 05.4.30

## 学位论文使用授权声明

苏州大学、中国科学技术信息研究所、国家图书馆、清华大学论文合作部、中国社科院文献信息情报中心有权保留本人所送交学位论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内容。论文的公布（包括刊登）授权苏州大学学位办办理。

研究生签名： 相松乔 日期： 05.4.30  
导师签名： 陈文富 日期： 05.4.30

# 第一章 绪 论

在纺织工业生产中, 纱线的张力是一项重要工艺参数。从织造准备到织造完成的各道工序都要涉及到张力, 其大小与稳定将直接关系到产品的外观效应及内在质量。

以丝织为例<sup>[1]</sup>, 络丝工序中的张力波动大小对络筒质量和筒子成形均有很大影响, 如果张力过大, 将使丝线受到过度的拉伸, 从而损伤丝的物理机械性能, 增加断头和回丝, 在织物中可能造成亮丝疵点; 如果张力过小, 则会导致络筒成形不良, 产生松软的卷装, 下道工序退解时发生困难等不良后果。在织造过程中, 张力控制显得更为重要, 经丝上机张力的大小将直接影响织造生产的进行和产品质量: 上机张力过大, 经丝易发毛, 毛丝和断头增加, 生产效率下降, 织物经向的强力伸长下降; 上机张力过小则会造成织口不清晰, 织疵增加, 织物不平挺, 打纬区增大, 纬丝打不紧, 织口不稳定以及打纬时经丝在综箱中过多地往复移动, 摩擦增多而造成断头。同样, 纬向张力控制也很重要, 纬向张力过大, 易产生亮丝、急纬、断纬以及纬向织物强力和伸长下降等严重后果; 纬向张力过小则易产生缩纬、松边。

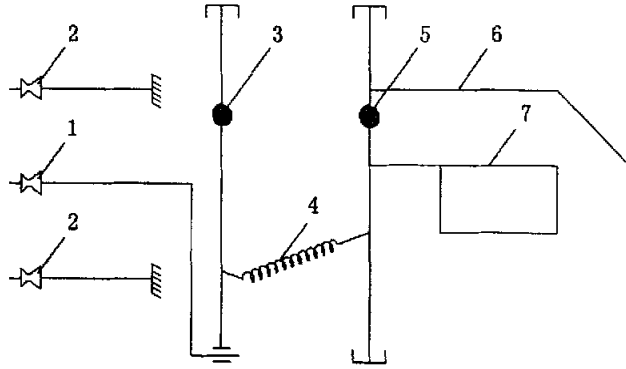
这些是丝织工程中的几个典型的张力控制, 另外还有许多方面需要对张力进行监控和检测。对于丝织工程中的张力检测可以采用定期定时的全面检测, 如检测络丝张力, 可每天一次, 对全厂络丝机逐锭加以检测, 观察测定值是否在规定的工艺参数值范围内, 以便及时加以纠正, 调整工艺, 避免不良后果。

由此可见, 张力值是影响纺织行业产品质量的一项重要指标。

## 1.1 张力仪的发展历程及工作原理

纱线张力仪的发展经历了从机械式到电子式的发展历程。机械式张力仪又称为指针式张力仪, 应用较为广泛的主要有 SFY13 型和 YG302 型。以 SFY13 型张力仪为例<sup>[2]</sup>, 其主要由测试罗拉 1、固定罗拉 2、扭簧 3、拉簧 4、游丝 5、指针 6、阻尼板 7、刻度盘及按钮等几部分组成, 如图 1-1 所示。在无外力作用下, 测试罗拉 1、扭簧 3、拉簧 4 与游丝 5 均处于静止平衡状态, 此时指针指向零位。在测试单丝张力时, 测试罗拉 1 受外力作用, 促使扭簧 3 沿受力方向扭转, 同时带动拉簧 4 向扭转方向伸张。拉簧 4 的伸张导致了游丝 5 产生扭力, 为维持其平衡状态, 游丝就向着

拉簧伸张的反方向扭转，同轴的指针随之发生偏转，指针稳定后所指示的值即为被测张力的实际值。



1——测试罗拉 2——固定罗拉 3——扭簧 4——拉簧 5——游丝  
6——指针 7——阻尼板

图 1-1 SFY13 型张力仪内部结构示意图

常见的机械式纱线张力仪还有日本 NAKASA 公司的 AN 型以及 TK 型。详细规格和技术特征见表 1-1<sup>[3]</sup>。

表 1-1 几种机械式纱线张力仪的规格特征

型 号	形 式	量 程 (CN)
SFY12	三罗拉	0~50      0~100
TK (日)	二罗拉	0~25      0~50      0~100
AN (日)	三罗拉	AN1 型 0~20   AN2 型 0~50   AN3 型 0~100 AN4 型 0~200   AN5 型 0~300   AN6 型 0~500
YG302	三罗拉	0~100

其它型号的张力仪，比如英国锡莱发展有限公司的机械式张力仪，意大利的某品牌机械式张力仪，以及国内一些厂家生产的机械式纱线张力仪等等，这里不作详尽列举。

机械式的纱线张力仪虽然能大致读出张力值，但存在一些固有的缺点，如操作

麻烦，修理不便，尤其是动态测定中的响应较慢，以致读数随着张力频率的变化而有较大误差。目前环锭纺的锭速已经超过 18000r/min，张力频率约 300Hz 左右，其值以及其变化是机械表无法读出的。且随着生产的发展，对张力的控制逐步量化、精确，这使得传统的指针式单纱张力仪愈发不能适应时代发展的要求。电子纱线张力仪由此应运而生。

电子纱线张力仪的主要原理是利用一个弹性传感器元件(通常为电阻式传感器，采用悬臂梁式弹性元件，其上下对称粘贴两片电阻应变片，组成全桥)对张力进行测量。工作流程为当弹性元件受力后，测量电桥阻值发生变化，将纱线的张力瞬时变化值转换成电信号，经过放大、滤波然后输出。按其输出方式，电子纱线张力仪可分为普通型和智能型两种。

普通电子纱线张力仪又可以分为两种：其一是将滤波后的结果经过 A/D 转换器后直接驱动液晶显示器，显示出即时张力，以满足张力精确量化的要求，但这种显示方式在张力变化频率较高时，难以读出张力值，使用不太方便；其二是在外部接记录仪，将所测得的张力值打印出来，以满足对张力波动进行分析的需求，但由于这样一套设备体积较为庞大，不便于在车间进行便携式测量，因此应用较少。

智能型电子纱线张力仪在内部使用了单片机，以单片机为核心，对 A/D 转换后的数据进行数据处理，并对各种参数(最大值、最小值、均方差、变异系数以及平均值)进行数字显示。而且，可以根据需要对数据进行保存，以便传输到上位机进行进一步的分析。因为这种智能型张力仪具有数据处理能力，并可显示多个参数值，因此应用效果较好，但同时价格也很昂贵。

电子纱线张力仪的主要型号有英国锡莱发展公司的电子张力仪、德国 SCHMIDT 公司的 ETM 系列电子张力仪以及国产的 YG301 型、SFYG14 型、YG984 型等等，具体规格和技术特征见表 1—2<sup>[9]</sup>。

表 1-2 几种电子纱线张力仪的规格特征

项 目	主要规格及技术特征		
	YG301 型	SFYG14 型	ETM 系列
形 式	三罗拉式	三罗拉式	三罗拉式
测量范围 /cN	0~100	0~200	0.5~100.01~500
测量精度 /1%	±5	±1	±1
电 源 /v	9	9	9
功 耗 /mW	约 200	≤150	不详
外形尺寸(长×宽×高)/mm	160×100× 35	250×65×31	269×62×46
重 量 /g	500	320	400

## 1.2 传统张力仪存在的问题

随着科技发展的日新月异, 纺织机械转速越来越高, 纱线速度越来越快, 虽然张力仪的各方面性能都做了很大的改进, 但传统张力仪还是存在着一些问题。

### 1.2.1 传统张力仪的导丝器材料不好

传统张力仪的导丝器采用的是耐磨的陶瓷材料, 高速运动的丝条与导丝器摩擦产生静电, 因此也就带来了静电干扰的问题。由于张力仪内部放大器是一种弱电系统, 具有很高的灵敏度, 很容易受到外界或内部一些无规则电信号的影响, 如果干扰信号和有用信号幅度相当时, 就会导致放大器在输出端的有用信号很难分辨, 甚至被干扰信号所淹没, 从而使仪表产生难以估计的误差。为了消除静电干扰, 有些厂家采用金属材料来做张力仪的导丝器, 导丝器另一端接地, 以便将产生的静电荷传导出去。这样一来, 静电干扰的问题就得到了解决, 但是随之带来的是金属材料的耐磨性问题。金属材料不耐磨, 在高速运动的丝条的摩擦下, 导丝器极易被磨损, 影响张力的测量, 使其寿命大大缩短。

### 1.2.2 张力仪的信号放大问题较难解决

张力仪传感器的输出电压为微伏级, 故张力仪对放大器的要求比较高, 选择一款合适的放大器就成为解决张力仪放大问题的关键。过去采用的是普通运算放大器,

其体积大、功耗大，各方面性能很难达到张力仪放大要求。如今广泛采用集成运算放大器，为了满足不同客户的需求，销售商提供了各种各样的集成运算放大器，每种都有各自不同的性能、特点和价格。以美国TI公司的INA128为例，它是一种低电压通用型仪表放大器，其特性优良、体积小，但是价格昂贵。

### 1.2.3 测量过程中张力值动态变化的可视性问题

传统指针式的张力仪在测量过程中，可以动态地显示出张力的波动范围，具有可视性的特点，但其响应较慢，以致读数随着张力频率的变化而有较大误差，并且缺乏精确性和数据收集、处理功能。电子纱线张力仪通过传感器将张力值以电信号的形式传到内部，然后经过一系列处理，由显示器完成数显功能，以满足张力精确量化的要求，其克服了指针式张力仪的缺点，却又失去了指针式张力仪的直观性、可视性。

## 1.3 本课题研究的主要内容和意义

当前我国的纺织行业正处在一个新的发展时期，经过“砸锭”转型、一系列的优化整合后，国内纺织生产力水平得到了释放，生产力水平大幅上升，进一步缩小了与国外同行业间的差距，提高了国际竞争力。但我们也要认识到我们所面临的困境，入世以来，虽然我国的纺织出口规模加大，但美国及欧盟对我国的纺织品出口限制也在增加，各种壁垒接踵而来。面对这种机遇与挑战并存的局面，把好质量关永远都是必要的，生产领域的质量控制尤其重要。国外发达国家的生产都有严格的量化标准，而在我国许多工厂还停留在经验判断上，许多重要工艺设置都是凭经验大致而为，经不起严格检验。随着我国纺织工业的发展，传统的张力检测仪器已经不能够完全适应现在的需求。此外，国外的某些同类型的仪器虽然可以满足要求，但价格却十分昂贵。因此本课题主要的研究内容就是研制一个智能型的电子张力仪，同时可以满足低成本以及当前高速和更加准确的张力量化控制的要求。

(1) 在导丝杆前端加上金属导丝头，变滑动摩擦为滚动摩擦，既减少导丝器的磨损，又可以解决静电积聚的问题。

(2) 本课题采用MAXIM公司的MAX4194仪表放大器，体积小、功耗微，与INA128相比，其优越之处在于性能相近，但体积较小、价格便宜（约为INA128的十分之一）。

(3) 本课题采用软件模拟形成光指针，同步显示数字量的显示方式，来解决测



量过程中张力值动态变化的可视性问题。

近年来，市面上出现了一种光指针/数字LCD双显仪表（如图1-2所示），该仪表在数显功能的基础上增加了光指针功能，是一种集模拟显示和数字显示优点于一身的新型显示技术。

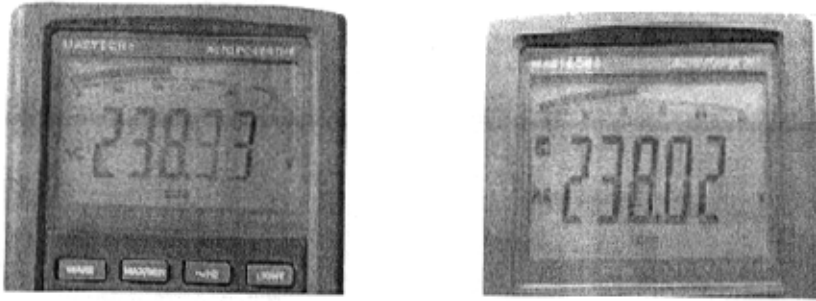


图 1-2 双显仪表效果图

这种双显系统(同时显示模拟量和数字量)既可以直观地显示精确的数字量，又可以一目了然地动态显示出张力的变化趋势和范围，提高了对被测信号波动变化倾向的判断能力，可以为控制张力，配置合适的工艺以及开发产品提供可靠的依据。但这种仪表要专用模具开发，对研究中的仪表则成本太高。如何发扬优点，克服成本高的缺点将在本课题中加以讨论。

## 第二章 张力仪的硬件设计

智能仪表的硬件电路设计包括微处理器的选择、电路设计、外部设备的选择与电路设计等。在仪表的硬件中，往往外设占有很大的比重，主机部分是通过输入、输出通道经测量元件和执行器件直接连结至被测或被控对象，并通过各种接口与显示器、打印机等人机部件相连。本课题中张力仪的硬件组成主要包括导丝器、传感器、微处理器、放大电路、LCD 显示电路、电源电路以及一些控制按键等。

### 2.1 导丝器的结构设计

本课题在导丝杆前端加上金属导丝头，变滑动摩擦为滚动摩擦，既减少导丝器的磨损，又可以解决静电积聚的问题。

#### 2.1.1 导丝器的测量原理

张力仪金属导丝器是由三个带滚动轴承的导轮构成，其中两个为固定导轮，起导丝和托丝作用，一个为张力轮，起测量作用。如图2-1，导轮的安装精度有一定的要求：三个导轮应在同一平面上，这样阻力最小，从而减少测量误差。图中， $T_1$ 、 $T_2$ 为经纱张力， $\alpha$ 、 $\beta$ 为经纱经过张力轮2 时和竖直方向的夹角， $F$  为经纱张力对张力轮2 的压力，根据力的合成原理有：

$$F = T_1 \times \cos \alpha + T_2 \times \cos \beta$$

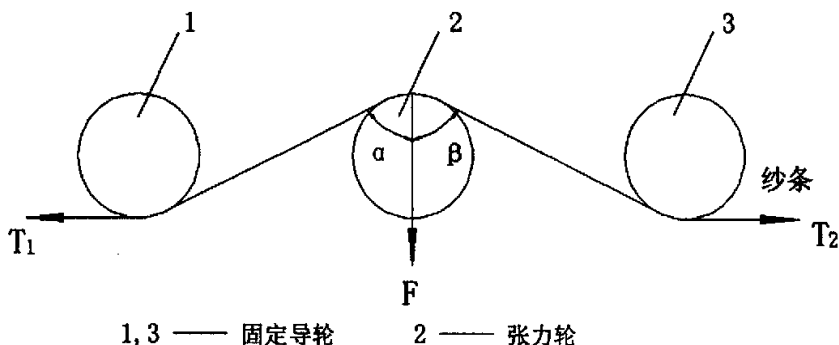


图2-1 张力仪导丝器结构示意图

在实际测量过程中，测量杆在 $F$ 的作用下会产生微小的偏移， $F$ 通过测量杆作用

在双孔梁上，使之产生变形<sup>[6]</sup>，如图2-2所示。

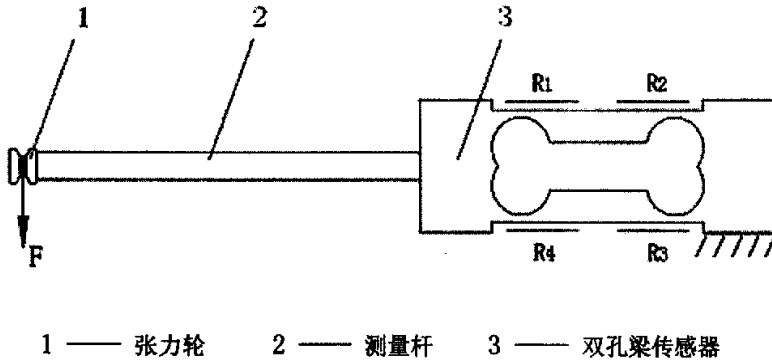


图2-2 张力仪测量结构示意图

### 2.1.2 导丝器的选择

传统张力仪的导丝器采用的是陶瓷材料，耐磨性很好，但其与高速运动的丝条摩擦易产生静电，对张力仪放大器产生静电感应而带来干扰。由于张力仪内部放大器是一种弱电系统，具有很高的灵敏度，很容易受到外界或内部一些无规则电信号的影响，如果干扰信号和有用信号可相比较时，那么放大器在输出端的有用信号将很难分辨，甚至被干扰信号所淹没，从而使仪表产生难以估计的误差。因此导丝器的选择应本着尽量消除静电的原则，将静电的干扰作用降至最低。

静电的消除有两种主要方式<sup>[6]</sup>：一是中和，二是泄漏。中和主要是通过空气进行，由于空气中含有带电粒子，带电体在与空气接触中，其所带有的电荷逐渐得到中和。但因空气中带电粒子的数量有限，且受空气湿度的影响很大，所以这种中和极为缓慢。泄漏主要是通过带电体本身以及与其相连接的其它物体进行。在张力测量过程中，由于摩擦而使张力仪导丝器出现静电，如果只靠中和来使静电消失所需时间是很长的，是测量所不允许的。为此考虑采取泄漏的方式，加速静电消失。

为了较好地消除静电干扰，本课题采用不锈钢材料来做张力仪的导丝器，导丝器另一端接地，以便将产生的静电荷及时传导出去，同时加盖铝质仪表壳，用以静电屏蔽。这样一来，静电干扰的问题就可以得到较好的解决。

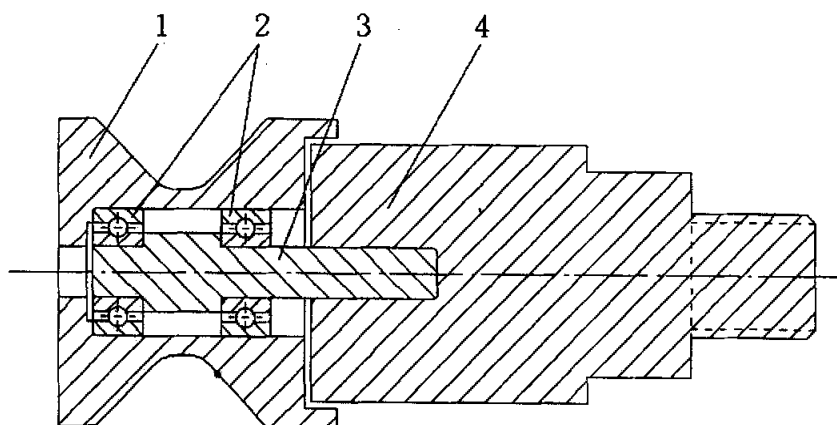
由于采用了金属材料做张力仪的导丝器，随之带来了金属材料的耐磨性问题。金属材料不耐磨，在高速运动的丝条的摩擦下，导丝器极易被磨损，给张力的测量带来影响，并使其寿命大大缩短。于是考虑在导丝器内嵌入微型轴承，将滑动摩擦

变为滚动摩擦，以减少导丝器的磨损。

### 2.1.3 导丝器的内部构造

本课题的导丝器设计参考了德国SCHMIDT公司生产的DTMX系列张力仪，但是由于其结构精密，无法拆卸以进行研究，所以在陈庆官教授的指导下，根据所学知识对其结构进行了猜想、设计。

张力仪导丝器的固定导轮和张轮内部都嵌入了两个微型滚动轴承，因而转动灵活，摩擦阻力很小，使张力仪能准确反映张力大小。具体结构如图2-3所示。



1——导轮壳 2——微型轴承 3——导轮轴 4——导轮座

图2-3 张力仪导丝器内部构造

导轮轴3一端与导轮座4联接，另一端通过两个微型滚动轴承2与导轮壳1联接，导轮壳在微型轴承的作用下绕着导轮轴转动。

为了防止微型轴承与导轮壳、导轮轴之间，导轮轴与导轮座之间在导轮运转时产生不应有的相对滑动，必须选择正确的配合。我们采用过盈配合来实现联接，过盈联接的结构简单、定心性好，承载能力较大并能承受振动和冲击<sup>[7]</sup>，工作时由于两配合件的弹性变形和装配过盈量，从而在配合面间产生很大的正压力，靠此正压力产生的静摩擦力来防止两配合件的相对滑动。但由于联接的承载能力直接取决于配合过盈量的大小，故对配合面加工精度要求较高。

由于微型轴承是标准件，因此在微型轴承的配合中，应通过改变导轮轴和导轮壳的尺寸公差来获得过盈配合。这样，微型轴承外圈与导轮壳间采用基轴制，即微

型轴承外圈是标准轴，导轮壳按圆柱体公差的一般基轴制的孔来制造；微型轴承内圈与导轮轴的配合采用基孔制，微型轴承内圈的孔是基准孔，导轮轴则按圆柱体公差的一般基孔制的轴来制造。

在导丝座的右端设计了螺纹，使导丝器与测量杆拆卸比较容易，因此可以方便地更换受损的导丝器。

## 2.2 传感器的选择

在几何量和机械量的测量中，最常用的传感器是某些金属和半导体材料制成的电阻应变片式传感器，它的工作原理是基于电阻的应变效应，即在导体产生机械变形时，它的电阻值相应的发生变化。

应变片式电阻传感器是以应变片为传感元件的传感器，它具有以下优点：

- (1) 精度高，测量范围广。
- (2) 使用寿命长，性能稳定可靠。
- (3) 结构简单、尺寸小、重量轻，因此在测试时，对工件工作状态及应力分布影响小。
- (4) 频率响应特性好。应变片响应时间为  $10^{-7}$  s。
- (5) 可在高低温、高速、高压、强烈振动、强磁场、核辐射和化学腐蚀等恶劣环境条件下工作。
- (6) 应变片种类繁多、价格便宜<sup>[18]</sup>。

但同时，它也存在着一一些缺点：

- (1) 在大应变状态下具有较大的非线性。
- (2) 输出信号微弱。
- (3) 不适用于高温环境中（1000℃以上）。
- (4) 应变片测出的只是某一面积上的平均应变，不能完全显示应力场中应力梯度的情况。

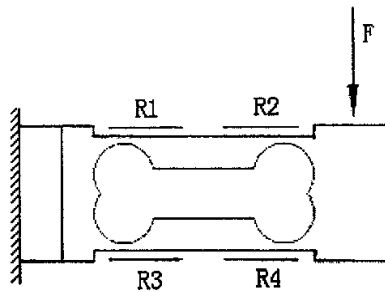


图 2-4 双孔梁式力传感器

本课题采用的正是基于这种应变片式的梁式力电阻传感器。如图 2-4 所示，为双孔梁式力传感器，多用于小量程工业电子称和商业电子秤。

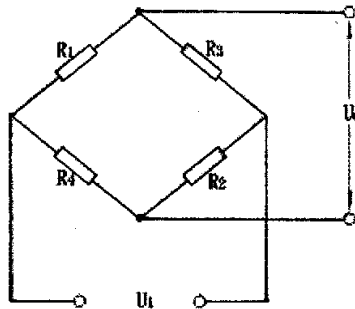


图 2-5 惠斯登全桥电路

在双孔梁上下表面有四个应变电阻  $R_1 \sim R_4$ ，组成图 2-5 所示的惠斯登全桥电路。当  $F$  作用在双孔梁时， $R_1$ 、 $R_2$  受到拉伸，电阻值增大； $R_3$ 、 $R_4$  受到压缩，其阻值减小。此时电桥失去平衡，产生  $U_o$  电压输出，即

$$U_o = U_i \cdot \Delta R/R = k \varepsilon U_i$$

其中， $k$  为灵敏度系数， $\varepsilon$  为应变， $U_i$  为输入电压。

则双孔梁的应变为：

$$\varepsilon = 3F(d - a/2 - \delta)/(Ebh^2)$$

其中  $d$  为梁端到梁中心的距离， $\delta$  为梁端到应变电阻的距离， $h$  为梁厚度， $b$  为梁宽度， $E$  为材料的弹性模量， $a$  为应变电阻基长。则双孔梁传感器的输出电压为：

$$U_o = 3kFU_i(d - a/2 - \delta)/(Ebh^2)$$

可见，输出电压  $U_o$  与外力  $F$  成正比。

由于双孔梁力传感器的 4 个应变电阻组成全桥差动电路，故输出电压的灵敏度

高, 线性度好, 但其输出电压为 $10^{-4}\sim 10^{-6}$  V 级, 信号十分微弱, 须经一定倍数放大, 才能被计算机数据采集系统所识别。此外, 当受力点的位置变化时, 一孔的弯矩增加, 另一孔的弯矩减少, 这可在桥内自动补偿, 从而提高测量的精度。

## 2.3 放大电路的解析

### 2.3.1 仪表放大器的发展与特点

随着电子技术的飞速发展, 运算放大电路也得到广泛的应用。仪表放大器为精密差分电压放大器。仪表放大器把关键元件集成在放大器内部, 其独特的结构使它具有高共模抑制比、高输入阻抗、低噪声、低线性误差、低失调漂移、增益设置灵活和使用方便等特点, 使其在数据采集、传感器信号放大、高速信号调节、医疗仪器和高档音响设备等方面倍受青睐。仪表放大器是一种具有差分输入和相对参考端单端输出的闭环增益组件, 具有差分输出和相对参考端的单端输出。与运算放大器不同之处是运算放大器的闭环增益是由反相输出端与输出端之间连接的外部电阻决定, 而仪表放大器则使用与输入端隔离的内部反馈电阻网络。仪表放大器的两个差分输入端施加输入信号, 其增益即可由内部预置, 也可由用户通过引脚内部设置或者通过与输入信号隔离的外部增益电阻设置。

**高共模抑制比**<sup>[8]</sup> 共模抑制比 (CMRR) 是差模增益 ( $A_d$ ) 与共模增益 ( $A_c$ ) 之比, 即  $CMRR = 20\lg_{10}(A_d/A_c)$  dB; 仪表放大器具有很高的共模抑制比, CMRR 典型值为 70~100 dB 以上。

**高输入阻抗** 要求仪表放大器必须具有极高的输入阻抗, 仪表放大器的同相和反相输入端的阻抗都很高而且相互十分平衡, 其典型值为 10<sup>9</sup>~10<sup>12</sup> K $\Omega$ 。

**低噪声** 由于仪表放大器必须能够处理非常低的输入电压, 因此仪表放大器不能把自身的噪声加到信号上, 在 1kHz 条件下, 折合到输入端的输入噪声要求小于 10nV/Hz。

**低线性误差** 输入失调和比例系数误差能通过外部的调整来修正, 但是线性误差是器件的固有缺陷, 它不能由外部调整来消除。一个高质量的仪表放大器典型的线性误差为 0.01%, 有的甚至低于 0.0001%。

**低失调电压和失调电压漂移** 仪表放大器的失调漂移也由输入和输出两部分组成, 输入和输出失调电压典型值分别为 100  $\mu$ V 和 2mV。

**低输入偏置电流和失调电流误差** 双极型输入运算放大器的基极电流, FET型输

入运算放大器的栅极电流，这个偏置电流流过不平衡的信号源电阻将产生一个失调误差。双极型输入仪表放大器的偏置电流典型值为 $1\text{nA}\sim 50\text{pA}$ ；而FET输入的仪表放大器在常温下的偏置电流典型值为 $50\text{pA}$ 。

**充裕的带宽** 仪表放大器为特定的应用提供了足够的带宽，典型的单位增益小信号带宽在 $500\text{kHz}\sim 4\text{MHz}$ 之间。

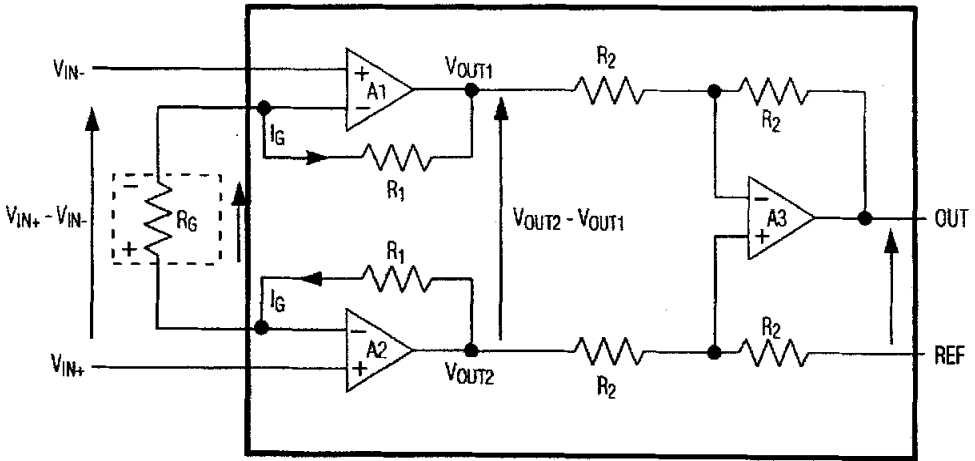
**具有“检测”端和“参考”端** 仪表放大器的独特之处还在于带有“检测”端和“参考”端，允许远距离检测输出电压而内部电阻压降和地线压降（IR）的影响可减至最小。

### 2.3.2 MAX4194内部电路及性能

本课题的张力仪采用MAXIM公司的仪表放大器MAX4194。MAX4194是一款增益可调的高精度仪表放大器，它具有满摆幅输出（Rail-to-Rail）、单电源工作、高精度以及高增益带宽等特点。这种固定增益仪表用放大器具有一种关断模式功能，它可以将截止电流降低到 $8\mu\text{A}$ 。采用传统的三运放结构，可以获得最大的直流精度。MAX4194具有满摆幅的输出，并且输入端可以在 $-200\text{mV}\sim +1.1\text{V}$ 范围内波动。其允许电流只有 $93\mu\text{A}$ ，单电源工作电压为 $+2.7\text{V}\sim +7.5\text{V}$ ，双电源为 $\pm 1.35\text{V}\sim \pm 3.75\text{V}$ 。该放大器都采用了8引脚的SO封装，且具有宽工作温度范围（ $-40^\circ\text{C}\sim +85^\circ\text{C}$ ）<sup>[9]</sup>。

MAX4194属于低功耗仪表放大器，采用三级放大布局。输入端由两个工作放大器组成，他们提供了一个固定增益的差分和一个共模增益。输出端是一个普通的微分放大器，它提供了大小为 $115\text{dB}$ 的共模抑制（ $G=10\text{V/V}$ ）。故MAX4194的增益可以完全控制在 $1\sim 10000$ 之间，其内部电路如图2-6所示。以往一些应用中，为了达到高放大倍数的目的而不得不采用外部多级放大布局，导致了放大后的信号混杂大量的干扰信号，有时甚至淹没了有用的信号。MAX4194由于内部集成了三级放大布局，在很大程度上削弱了干扰信号的影响。





注： $R_1=R_2=25k\Omega$

图2-6 MAX4194内部放大电路示意图

对该放大器而言，共模输入范围都为  $V_{EE}-0.2V \sim V_{CC}-1.1V$ 。理想状态下，放大器只需在  $IN-$ 、 $IN+$  两个输入端输入一对差动电压。如果两个输入端保持相同电压，输出的就是  $V_{REF}$ 。在  $IN-$ 、 $IN+$  两个输入端输入的一对差动电压，通过增益电阻，从而产生电流。该电流同时也经过两个输入放大器  $A_1$  和  $A_2$  的反馈电阻，从而产生一个差动电压<sup>[9]</sup>：

$$V_{OUT2} - V_{OUT1} = I_G \cdot (R_1 + R_G + R_1) \quad (2-1)$$

其中， $V_{OUT1}$  和  $V_{OUT2}$  分别为  $A_1$  和  $A_2$  的输出电压， $R_G$  为增益电阻， $R_1$  为输入放大器的反馈电阻， $I_G$  由下面的方程式确定：

$$I_G = (V_{IN+} - V_{IN-}) / R_G \quad (2-2)$$

(2-2) 式代入 (2-1) 得

$$V_{OUT2} - V_{OUT1} = (V_{IN+} - V_{IN-}) \cdot [(2 \cdot R_1) / R_G + 1]$$

因为

$$V_{OUT} = V_{OUT2} - V_{OUT1}$$

所以放大器的输出电压表达如下：

$$V_{OUT} = (V_{IN+} - V_{IN-}) \cdot [(2 \cdot R_1) / R_G + 1]$$

$R_1$  的阻值为  $25k\Omega$ ，故

$$G = (V_{IN+} - V_{IN-}) = 1 + 50k\Omega / R_G$$

MAX4194 放大器在  $R_G$  引脚（1，8 引脚）间连接一个外部增益电阻  $R_G$ ，可以通过

改变 $R_G$ 的阻值来方便地改变放大器的增益,以获得所需的放大倍数。 $R_G$ 的选择与增益的对应关系如表2-1所示。

表2-1 MAX4194外部增益电阻选择表

增益G (V/V)	$R_G$ (1%) ( $\Omega$ )	$R_G$ (5%) ( $\Omega$ )
+1	$\infty^*$	$\infty^*$
+2	49.9k	51k
+5	12.4k	12k
+10	5.62k	5.6k
+20	2.61k	2.7k
+50	1.02k	1.0k
+100	511	510
+200	249	240
+500	100	100
+1,000	49.9	51
+2,000	24.9	24
+5,000	10	10
+10,000	4.99	5.1

注: 引脚1、8开路可获得 $G=1V/V$

下面通过实验来验证MAX4194的增益情况,实验电路如图2-7。

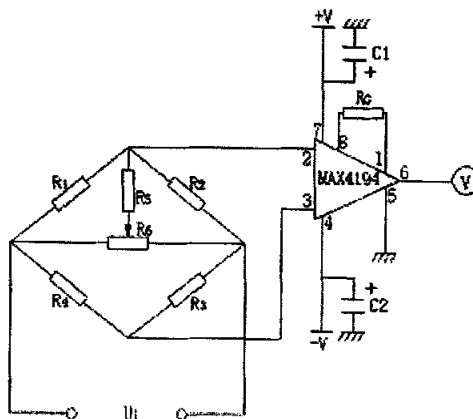


图2-7 增益验证实验电路图

$R_0$  为调零电位器， $C_1$  为滤波电容，用以保持电源电压的稳定，防止因电源电压的不稳定导致的信号的浮动。输入的电压信号经放大器放大后由电压表测得。

外界条件：

电桥与放大器部分合用电源  $V_{总}=5.854V$

在无负载的情况下，调节调零电位器，使得输出值为  $2.000V$ ，因为不够稳定的缘故，最后测量时输出值为  $1.946V$ ，测得  $V_{23}=1.95mV$

$R_G=50\Omega$ ，由公式  $G=(1+50K\Omega/R_G)$  得到

$G=1001$

$V_{OUT}=V_{23}\times G$

$=1.95\times 1001$

$=1951.95mV$

$=1.95195V$

因为  $1.95195/1.946=1.003\approx 1$

所以可以认为在此情况下的放大是合乎要求的。

### 2.3.3 MAX4194 在张力仪中的应用电路

本课题所研究张力仪的工作流程图如图 2-8。通过传感器将纱线张力的瞬时变化值转换成电信号，经过 MAX4194 放大器放大后，进入单片机进行 A/D 转换并经过一系列的数据处理，最后送达显示装置。经过数据处理后可显示多个张力参数值，还可以将张力值输出到电脑进行进一步的分析，使纱线的张力测量更具科学性和合理性。MAX4194 在其内部的应用电路如图 2-9 所示：

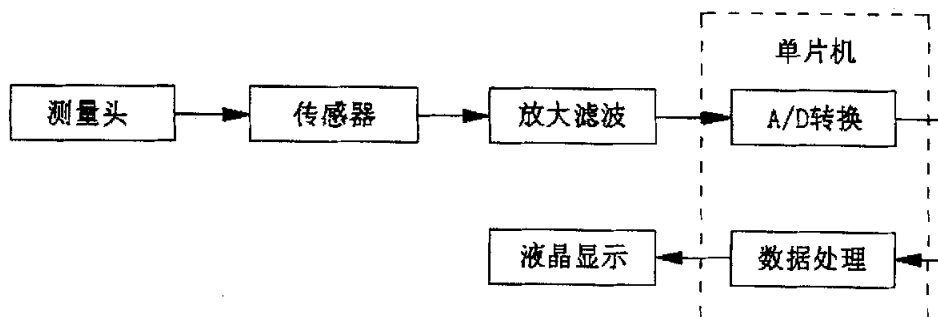


图2-8 张力测量流程图

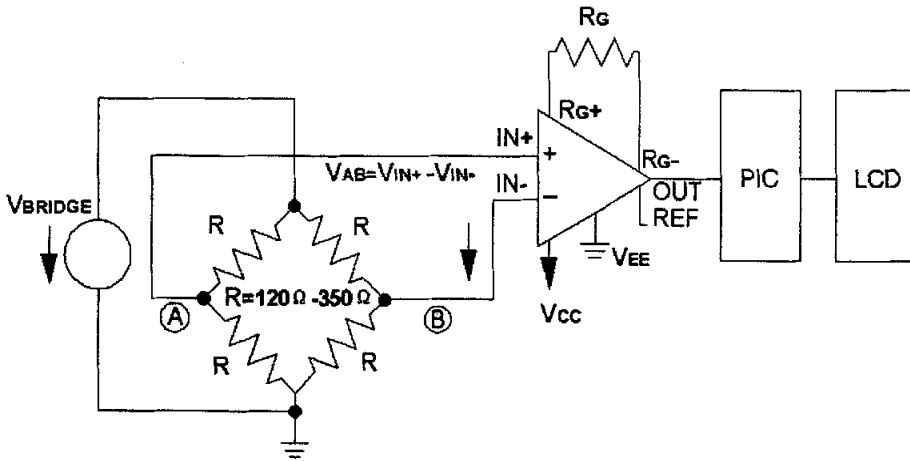


图2-9 MAX4194在张力仪中的应用电路

### 2.3.4 MAX4194以及悬臂梁传感器的线性测试

为检测4194和悬臂梁传感器组成的系统线性度，构建了一个小系统进行了测试。装置的原理框图如图2-10所示：

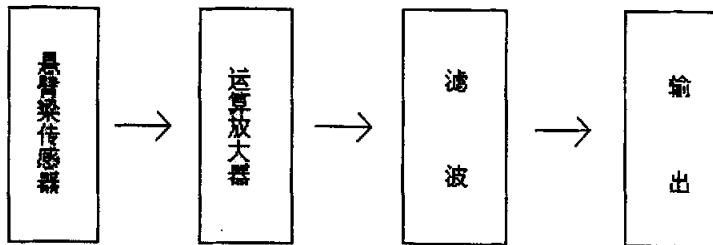


图2-10传感器的线性测试原理图

通过原理框图可以看该仪器的工作流程。将悬臂梁传感器一端固定，纱线通过另一端时，纱线张力的大小变化，可以导致传感器输出电压的大小变化。电压经由传感器进入运算放大器，因为纱线的张力变化很小，所以电压的变化量若不经放大根本无法观察到，通过运算放大器约 1000 倍的放大，电压进入到滤波电路，经过滤波除掉一些较强的干扰，最后输出到终端——电压表。经过这一系列的处理后，显示在电压表上的电压变化就非常明显了，足以达到观察张力的动态变化的要求。硬件设计电路图如图 2-11 所示：

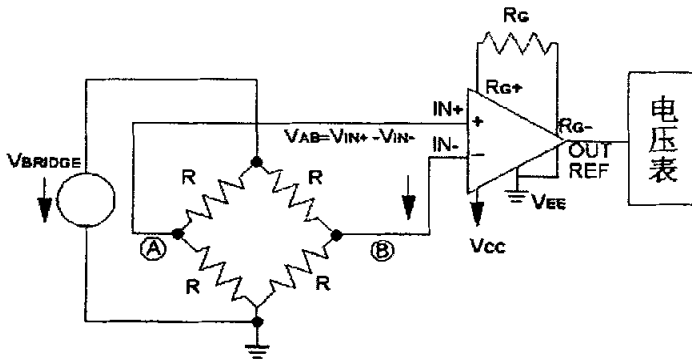


图 2-11 传感器的线性测试硬件电路图

在室温 21℃，无风的状况下对该装置的性能作了一系列的测试和分析。

在以上的环境下，每隔 30 秒记录一次数据，建立表 2-2:

表 2-2 实验数据 1

负载 (g)	0	1	2	5	10	20	50	100
读数 (mv)	160.08	165.20	169.65	183.44	206.50	252.7	391.60	622.70

(注：以上读数均为在同等条件下三次读数的平均值)

对上表数据进行一元线性回归分析:

令 y 为读数, x 为负载

设  $y = a + bx$

为便于计算, 令  $y_1 = y - 160$

建立如表 2-3 所示的实验数据 2

2-3 实验数据 2

$y_1$	x	$y_1^2$	$x^2$	$xy_1$	
0.08	0	0.0064	0	0	
5.20	1	27.04	1	5.20	
9.65	2	93.1225	4	18.5	
23.44	5	549.43	25	117.2	
46.50	10	2162.25	100	465	
92.70	20	8593.29	400	1854	
231.60	50	53638.56	2500	11580	
462.70	100	214091.29	10000	46270	
$\Sigma$	871.87	188	279155	13030	60310.7

进行回归分析:

$$y_1 = a + bx$$

$$a = \bar{y}_1 - b\bar{x}$$

$$b = \frac{\sum_{i=1}^n x_i y_i - n\bar{x}\bar{y}_1}{\sum_{i=1}^n x_i^2 - n\bar{x}^2}$$

$$n=8 \quad \bar{x} = 23.5 \quad \bar{y}_1 = 108.98$$

$$b = \frac{60310.7 - 8 \times 23.5 \times 108.98}{13030 - 8 \times 23.5^2}$$

$$= \frac{39822.46}{8612}$$

$$= 4.62$$

$$a = \bar{y}_1 - b\bar{x}$$

$$= 108.98 - 4.62 \times 23.5$$

$$= 0.4$$

$$\text{解得 } y = 160 + 0.4 + 4.62x = 160.4 + 4.62x$$

相关性检验:

$$\text{相关系数为 } r = \frac{L_{xy}}{\sqrt{L_{xx} \cdot L_{yy}}}$$

$$L_{xx} = 13030 - \frac{188^2}{8} = 8612$$

$$L_{yy} = 279155 - \frac{871.87^2}{8} = 184135.34$$

$$L_{xy} = 60310.7 - \frac{188 \times 871.4}{8} = 39812.8$$

$$r = \frac{39812.8}{\sqrt{8612 \times 184135.34}} = \frac{39812.8}{39821.77} = 0.99$$

查相关系数表知极显著。由此可知, 由该传感器和运放组成的系统测量结果线性度很好, 可以应用到测量张力上。

## 2.4 微处理器的选择

### 2.4.1 微处理器的选择原则<sup>[15]</sup>

微处理器（或微处理机）是智能仪表的核心部件，它的结构、特性对所研制仪表的性能有很大影响。所以要成功地研制一台智能仪表，首先应选择合适的微处理器。在选择微处理器时，一般应考虑如下的主要特性：

#### (1) 用途

一般而言，数据处理和控制是它的两个主要用途。

#### (2) 字长

微处理器的字长取决于并行数据总线的数目。通常使用4位、8位或16位的微处理器来研制智能仪表及其应用系统。一般8位的微处理器应用较多。

#### (3) 寻址范围和寻址方式

微处理器的地址反映了微机可寻址的范围，它表示系统中可存放的程序和数据量，微处理器有多种寻址方式，如直接、间接、相对寻址、变址寻址等，选择恰当的寻址方式，能使程序量大为减少，从而可节省存储空间和加快程序的执行速度。

#### (4) 指令的功能

一般说来，指令条数多的微处理器，其操作功能要强些，但一个微处理器的功能究竟丰富与否，不能单由指令的数量确定，还要看每一条指令的具体内容。

#### (5) 执行速度

执行速度的选择也依对象的不同而有区别。对于采样周期较短而有大量实时计算的处理或过程控制系统，应选择速度高的微处理器。

#### (6) 中断能力

在实际应用中，外部设备常要求微处理器暂时停止目前的工作，转去执行一个为中断服务的程序。为了满足这一要求，微处理器必须具有较强的中断能力。对于快速、多通道实时处理的对象，应选择中断功能丰富的微处理器。

#### (7) 硬件与软件支持

选择微处理器时，应考虑该器件有足够的硬软件支持。从硬件来说，构成一个智能仪表要有足够的外围芯片，例如串行口、并行口、定时计数器、A/D和D/A转换器等。

#### (8) 功耗

应按器件所允许的温度范围和仪表使用环境等条件来选择不同功耗的微处理

器。

#### (9) 成本

智能仪表的成本是优先考虑的指标之一，特别是成批生产仪表时更是如此。但是，是否能正确选择微处理器，又将直接影响到整个仪表的成本。因此必须仔细权衡、全盘考虑。

### 2.4.2 PIC单片机的机构及功能

PIC是美国Microchip公司所生产的单片机系列产品型号的前缀。PIC系列单片机的硬件系统设计简洁，指令系统设计精炼。它与以往单片机相比，有其自身的优点<sup>[11]</sup>：

(1) 哈佛总线结构 PIC系列单片机在框架上采用了与众不同的设计手法，它既不象MOTOROLA公司开发生产的MC68HC05/08系列的单片机那样，其程序存储器和数据存储器统一编址，也不象早期在国内市场上最流行的MCS-51系列单片机产品那样，其程序存储器和数据存储器虽然独立编址但是它们与CPU之间必须共用一条总线，仍然摆脱不了瓶颈效应的制约，于是影响到CPU运行速度的进一步提高。而在PIC单片机芯片内部将数据总线和指令总线分离，并且采用不同的宽度。这样做的好处是，便于实现指令提取的“流水作业”，也就是在执行一条指令的同时对下一条指令进行取指操作，便于实现全部指令的单字节化、单周期化，有利于提高CPU执行指令的速度。

(2) 指令单字节化 因为数据总线和指令总线是分离的，并且采用了不同的宽度，所以程序存储器ROM和数据存储器RAM的寻址空间是相互独立的，而且两种存储器宽度也不同。这样设计不仅确保了数据的安全性，还能提高运行速度和实现全部指令的单字节化。

(3) 精简指令集(RISC)技术 PIC系列单片机的指令系统只有35条指令，而MCS-51单片机的指令系统共有111条指令，这对于指令的学习、记忆，程序的编写、调试带来了极大的便利。另外，PIC系列单片机不仅全部指令均为单字节指令，而且绝大多数都为单周期指令，利于提高执行速度。

(4) 寻址方式简单 PIC系列单片机只有4种寻址方式(寄存器间接寻址、立即数寻址、直接寻址和位寻址)，容易掌握。而MCS-51单片机有7种寻址方式。

(5) 代码压缩率高 1KB的存储器空间，对于象MCS-51这样的单片机，大约只能



存放600条指令，而对于PIC系列单片机则可以存放的指令条数可达1024条。从图2-12中可以看出，与几种典型的单片机相比，PIC16C5X是最节省程序存储器的单片机，也就是说，对于完成相同功能的一段程序，所占用地址空间MC68HC05是PIC16C5X的2.24倍。

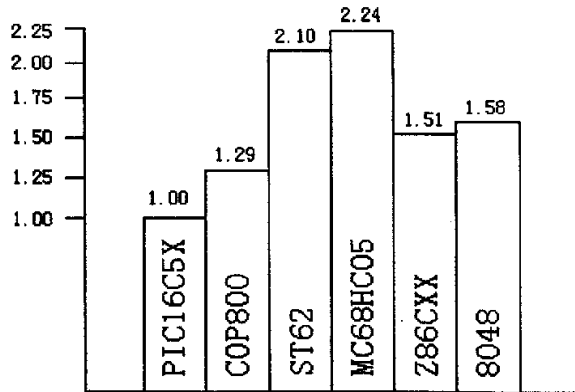


图2-12 典型单片机代码紧凑性比较

(6) 运行速度快 由于采用了哈佛总线结构，以及指令的读取和执行采用了流水线作业方式，使得运行速度大大提高，远远高于同一档次的其他单片机。

(7) 功耗低、驱动能力强 PIC单片机的功率消耗极低，是目前世界上最低的单片机品种之一。I/O端口驱动负载的能力较强，每个I/O引脚输入和输出电流的最大值分别达到25mA和20mA，能够直接驱动发光二极管LED、光电耦合器或者微型继电器等。

(8) 外围电路简洁 PIC系列单片机内集成了上电复位电路、I/O引脚上拉电路、看门狗定时器等，可以最大程度地减少或免用外接器件，以便实现“纯单片”应用。这样，不仅便于开发，而且还可以节省用户的电路板空间和制造成本。

(9) C语言编程 对于掌握了C语言的用户，Microchip公司还为其提供了“C语言编译程序”，这样的用户如果使用C语言这种高级语言进行程序设计的话，可以大大提高工作效率。

(10) 程序保密性强 目前，尚无办法对其直接进行解密拷贝，可以最大限度地保护用户的程序版权。

在此我们选用的是Microchip公司于1998年底推出的一款特色明显的新产品——PIC16F877，它隶属于PIC16F87X子系类，是一款在8位单片机中功能最为全面的

型号。它具有以下一些主要性能：8K×14 容量的 FLASH 存储器、368 个字节的 RAM 存储器、256 字节 EEPROM、40/44 引脚封装、3 个定时器、8×10 位的 A/D 转换、RA—RE 端口总数为 33 的 I/O 端口、14 个中断源、WDT 看门狗，具有通用串口 UASRT 和同步串口 SSP，还具有上电延时复位、掉电延时复位、在线串行编程等功能。

另外，Microchip 公司还专门为 PIC 系列单片机设计了一套功能强大的集成开发环境 MPLAB-ICD，可以通过网上下载和光盘发行 2 种方式为用户免费提供，借助于这套在线调试工具，我们可以在自己的微机系统上对 PIC 系列单片机进行程序的创建、录入、编辑以及汇编，甚至还能实现程序的模拟运行和动态调试之类的虚拟实战演练，并且调试的方式可以采用连续运行、单步运行。自动单步运行、设置断点运行等多种运行方式。用户既可以实现硬件仿真，又能够实现程序烧写，可以边学边练，学用结合，而不需要在经济上和时间上付出太大的投入<sup>[11]</sup>。

## 2.5 显示器件

本课题中采用 LCD 作为显示器件，选用的 LCD 模块是三星电子公司的 KS0066U 驱动芯片，KS0066U 是点阵液晶显示器的驱动控制器大规模集成电路，它采用低功耗 CMOS 技术，能显示 2 行 5×8 或 5×11 点阵格式，具有如下的主要特征：

- (1) 字符液晶显示器驱动控制器。
- (2) 内部驱动器：16 共极和 40 段信号输出。
- (3) 显示字符类型：5×8 点阵格式（204 种），5×11 点阵格式（32 种）。
- (4) 特殊字符可通过字符发生器 RAM 直接编程。
- (5) 多种指令功能。
- (6) 自动上电复位。
- (7) 内部存储器

ROM (CGRAM)：10080 位（208 字符×5×8 点）和（32 字符×5×11 点）；

RAM (CGROM)：8×8 位（8 字符）；

RAM (DDRAM)：80×8 位（最大 80 字符）。

与单片机的连接电路如图 2—13 所示<sup>[13]</sup>所示。DB4~DB7 依次与单片机的 RB2~RB5 连接，为数据线，DB0~DB3 未使用；RS 为显示、指令选择线，当 RS 为高电平时单片机向 LCD 模块进行读写操作；E 为数据接收、发送标志，每当 E 线向 LCD 模块发送一个脉冲，LCD 模块与单片机之间将进行一次数据交换；R/W 为读写选择线，本论文应用

中R/W线接地，只做对LCD模块的写操作。

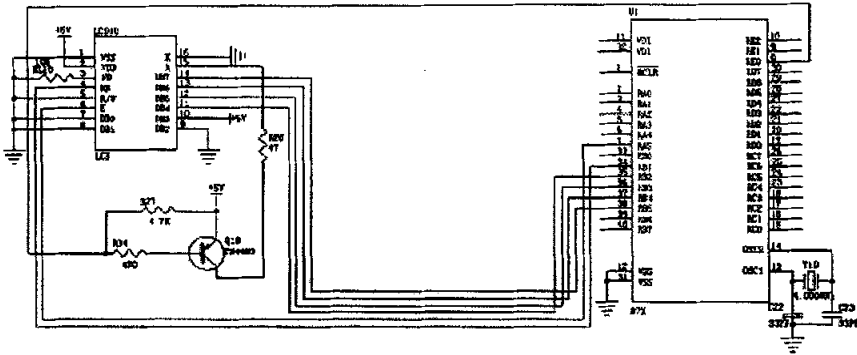


图 2-13 单片机与显示模块连接图

## 2.6 电源模块

系统中采用了 MAXIM 公司一款新型电压模块 MAX6165，它是精密的、低压差、低功耗电压基准，工作于 ( $V_{out} + 200\text{mV}$ ) 至 12.6V 输入电压范围，具有 1.25V、2.048V、2.5V、3V、4.096V、4.5V 和 5V 输出电压选项。它们具有专用的曲率校正电路和光刻薄膜电阻，温度系数低至  $5\text{ppm}/^\circ\text{C}$ （最大值）、初始精度  $\pm 2\text{mV}$ （最大值）。这些器件可工作于扩展级温度范围 ( $-40^\circ\text{C}$  至  $+85^\circ\text{C}$ )。MAX6165 消耗电源电流  $90\mu\text{A}$ （典型值），并可输出  $5\text{mA}$  (MAX6161 为  $4\text{mA}$ ) 或灌入  $2\text{mA}$  负载电流。传统的并联模式(2端)基准源消耗较大的电源电流、且需外加电阻。这些器件则不同，其电源电流与电源电压基本无关（变化量为  $8\mu\text{A}/\text{V}$ ），并且无需外加电阻。此外，这些内部补偿器件无需外部补偿电容，在高达  $1\mu\text{F}$  的电容性负载下仍可稳定工作。消除外部补偿电容可有效节省空间受限应用中的印刷板的尺寸。低压差电压和与电源电压无关的超低电源电流使得这些器件非常适合高性能、低电压的电池供电系统<sup>[10]</sup>。

采用了 MAX6165 使得仪表的功耗大大降低。而且，由于它出色的稳定性，可以用来作为 A/D 转换的电压基准，降低了由于电压基准的飘动而导致的转换结果的漂移现象。

## 2.7 张力仪电路原理图

本张力仪电路采用专业的电路绘图工具 protel99se 进行了原理图的设计, 如图 2-15 所示, 图中按功能划分可以分为五部分:

(1). 单片机主体部分

这一部分以单片机为核心, 包括单片机的各个外围输入输出接口、晶振、电源滤波电路等。

(2). LCD 部分

LCD 部分主要是 KS0066 与单片机的连接图。

(3). 传感器—放大器部分

包括传感器原理图、MAXIM4194 运放。

(4). 功能按键部分

HOLD、ZERO 等功能键的电路原理图。

(5). 电源部分

电源模块 MAXIM6165 及其外围电路。

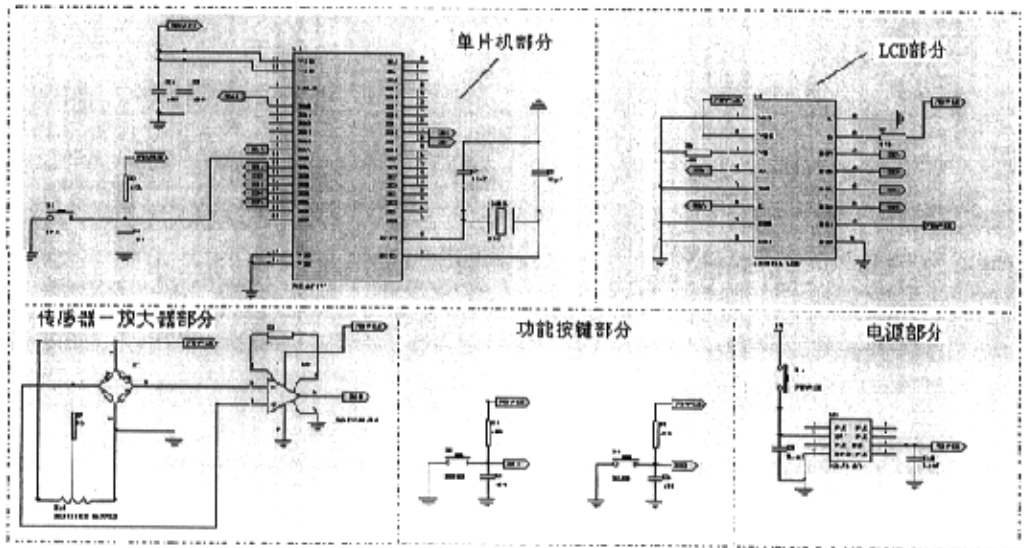


图 2-15 张力仪电路原理图

因为电路中主要采用了集成模块, 所以省去了大量的外围器件, 不但节省了宝贵的电路板空间, 而且可以提高系统的稳定性。

## 第三章 张力仪的软件设计

智能仪表的软件设计是仪表设计的重要任务之一。硬件电路确定之后,仪表的主要功能将由软件来实现。对同一个硬件电路,配以不同的软件,所实现的功能也就不同,而且有些硬件电路功能可以用软件来实现,可以说智能仪表的设计很大程度上是软件的设计。

本课题中张力仪的软件设计主要分为以下几个方面:

1. 监控程序设计
2. 功能程序设计
3. 软件的标定设计

### 3.1 监控程序设计

监控程序的主要作用是能及时地响应来自系统或者仪表内部的各种服务请求,有效地管理仪表的自身软硬件及人机联系设备,与系统中其它仪器设备交换信息,在系统一旦出现故障时,及时进行相应处理。监控程序一般包括初始化管理、中断管理、人机界面管理等。

#### 3.1.1 初始化管理

初始化管理主要由以下几个部分组成<sup>[15]</sup>:

(1) 对微处理器内部特殊功能寄存器以及其它各种可编程器件进行初始化状态设计。

(2) 对监控使用的 RAM 进行分配并赋予初值。通常包括确定各种标志位,设置显示缓冲区以及用户特殊要求的专用单元。如在 A/D 转换时,用户要先设置转换速率、采集次数、采集数据存放的首地址等,这些参数都应有确定的存储单元存储。

本课题中,也对内部的特殊功能寄存器以及对监控使用的 RAM 进行了初始化状态的设计。如程序 3-1 所示。

#### 程序 3-1

```
list          p=16f877
include      "p16f877.inc"
```

ACCALO	EQU	20h
ACCAHI	EQU	21h
ACCBLO	EQU	23h
ACCBHI	EQU	24h
ACCCLO	EQU	26h
ACCCHI	EQU	27h
ACCDLO	EQU	28h
ACCDHI	EQU	29h
TEMP	EQU	2Ah
SIGN	EQU	2Bh
Temp	EQU	2cH
TempH	EQU	2DH
TempM	EQU	2EH
TempL	EQU	2FH
TempMM	EQU	30h
COUNT3	EQU	31h
TEMP	EQU	45h
S0	EQU	32h
S1	EQU	33h
S2	EQU	34h
R0	EQU	35h
R1	EQU	36h
R2	EQU	37h
R0LO	EQU	38h
R0HI	EQU	39h
R1LO	EQU	3ah
R1HI	EQU	3bh
R2LO	EQU	3ch
R2HI	EQU	3dh
R3	EQU	4bh
R3HI	EQU	4eh
R3LO	EQU	4fh
R4	EQU	50h
R4HI	EQU	51h
R4LO	EQU	52h
ACCA	EQU	46H
DIVRES	EQU	47H
DIVRES1	EQU	48H
DIVRES2	EQU	49H
MSB	EQU	7

```

TRUE      EQU      1
YES       EQU      1
FALSE     EQU      0
SIGNED    EQU      TRUE
CARRY     EQU      0
Z_BIT     EQU      2
DCARRY    EQU      1
; 定义 LCD 数据& 控制 I/O 口
LCD_DATA  EQU      PORTB
LCD_CNTL1 EQU      PORTB
LCD_CNTL2 EQU      PORTA
; 定义 LCD 控制相应的 I/O 引脚
RS        EQU      1
E         EQU      5
; LCD 模块命令
DISP_ON   EQU      0x00C ; 开显示
DISP_ON_C EQU      0x00E ; 开显示, 开指针
DISP_ON_B EQU      0x00F ; 开显示, 开指针, 开闪烁
DISP_OFF  EQU      0x008 ; 关显示
CLR_DISP  EQU      0x001 ; 清显示的操作数
ENTRY_INC EQU      0x006 ; 操作数
ENTRY_INC_S EQU      0x007 ; 操作数
ENTRY_DEC EQU      0x004 ; 操作数
ENTRY_DEC_S EQU      0x005 ; 操作数
DD_RAM_ADDR EQU      0x080 ; 初始化第一行显示的操作数
DD_RAM_UL  EQU      0x0c0 ; 初始化第二行显示的操作数

; 设置显示相关寄存器
Byte      EQU      3fH
Count     EQU      40H
Count1    EQU      41H
Count2    EQU      42H

```

LCD_DATA1	EQU	43H
PORTDB	EQU	44H

在程序 3-1 中,对软件中的内部功能寄存器以及要用到的 RAM 寄存器进行了分配。此外,还对显示器件进行了初始化。

在许多应用中,常常要把模拟量转换成数字量,再送入单片机处理。本系统中采用单片机内部集成的 A/D 功能模块,它是一种逐次逼近型的 ADC。ADC 的操作过程要求占用较多的时间,并且其占用的时间主要包括两个部分:采样/操持电容的充电时间和 A/D 转换电路的转换时间。对于采样时间可以采用下式计算最短采用时间(即采用所需的最短时间)<sup>[12]</sup>:

$$\begin{aligned}
 T_{ACQ} &= \text{放大器延迟时间} + \text{保持电容充电时间} + \text{温度补偿时间} \\
 &= T_{AMP} + T_C + T_{COFF} = 2 \mu s + T_c + [( \text{温度} - 25^\circ\text{C} ) \times 0.05 \mu s / ^\circ\text{C}] \\
 &= 2 \mu s + 16.47 \mu s + [(50^\circ\text{C} - 25^\circ\text{C}) \times 0.05 \mu s / ^\circ\text{C}] \\
 &= 19.72 \mu s
 \end{aligned}$$

( $V_{DD} = 5V$  信号源内阻  $R_s$  为最大值  $10K\Omega$  工作温度为  $50^\circ\text{C}$ )

一般在转换完成之后,下一次采样重新开始之前,必须加入  $2T_{AD}$  (每一位数据的转换时间,  $T_{AD} \geq 1.6 \mu s$ ) 的等待时间,由于本文在 A/D 转换结束后还有数据处理以及显示程序,经过时间计算需要耗费  $8.18ms$ ,远大于  $2T_{AD}$ ,所以没有再另加  $2T_{AD}$  的延时。

下面是一段本系统中的 A/D 程序片断(程序 3-2),其实现的功能是:

- (1) 应用 RA0/AN0 作为 A/D 转换输入通道;
- (2) 参考电压源  $V_{REF}$  选择内部的  $V_{DD}$  和  $V_{SS}$ ;
- (3) A/D 转换时钟选用外部振荡器;
- (4) 禁止 A/D 模块的中断功能

#### 程序 3-2

```

bcf      status, rp0
clrf     portd
movlw   b'01000001'
movw    adcon0
bsf     status, rp0
clrf    trisd

```



```

        movlw    b'00001110'
        movwf   adcon1
        bcf     status, rp0

START
        bsf     adcon0, go
wait    btfss   pir1, adif
        goto    wait
        movf    adresh, w
        movwf   portd
        movwf   accalo
        movwf   acca

```

### 3.1.2 中断管理

智能仪表在很大程度上是一个实时操作系统，因而要具有及时处理各种可能事件的能力，这主要靠中断功能来实现。能够发出中断请求信息的外设或事件称为中断源。中断管理主要是接受各类中断请求信号，根据优先级，转入相应的一段程序，按下什么键就可以调用该键所对应的程序段。通常智能仪表开机时，处于禁止中断状态，初始化结束后，监控主程序执行一条“开放中断”命令，使仪表在接到中断请求信号后，能立即进入中断工作方式。

中断管理程序应具有以下功能：

#### (1) 实现中断及返回

当某一中断源发出中断申请时，决定是否响应这个中断请求（当仪表在执行更紧急、更重要的工作时，可以暂不响应中断），若允许响应这个中断请求，CPU 必须在现行的指令执行完毕后，把断点处的 PC 值，各个寄存器的内容和标志位的状态保存下来，称为保护断点和现场。然后，转到需要处理的中断源的服务程序入口，同时清除中断请求触发器。当中断处理完毕后，再恢复被保留的各个寄存器和标志位的状态（称为恢复现场），再恢复 PC 值（恢复断点），使 CPU 返回断点，继续执行程序。

#### (2) 能实现优先权排队

通常，在系统中有多个中断源，不可避免会出现两个或多个中断源同时提出中

断请求的情况，这样，需要设计者事先根据轻重缓急，给每个中断源确定一个中断级别——优先权。当多个中断源同时发出中断申请时，CPU 能找到优先权级别最高的中断源，响应它的中断请求；在优先权级别最高的中断源处理完毕以后，再响应级别较低的中断源。

### (3) 高级中断源能中断低级别的中断处理

当 CPU 响应某一中断源的请求，在进行中断处理时，若有优先权级别更高的中断源发出中断申请，则 CPU 要能中断正在进行的中断服务程序，保留这个程序的断点和现场（类似于子程序嵌套），响应高级中断，在高级中断处理完毕后，再进行被中断的中断服务程序。而当发出新的中断申请的中断优先级与正在处理的中断源同级或更低时，则 CPU 就先不响应这个中断申请，直至正在处理的中断服务程序执行完以后才去处理新的中断申请<sup>[15]</sup>。

由于本课题中只开放了一个中断源，所以上面的第二和第三功能没有体现出来，具体的实现语句如程序 3-3 所示。

#### 程序 3-3

开放中断语句：

```
BSF      INTCON, GIE      ; 开放全局总中断
```

```
BSF      INTCON, INTE     ; 开放 INT 中断
```

现场保护语句：

```
SERV                                           ; 保护现场
```

```
MOVWF   W_TEMP
```

```
SWAPF   STATUS, W
```

```
CLRF    STATUS
```

```
MOVWF   STATUS_TEMP
```

```
MOVF    PCLATH, W
```

```
MOVWF   PCLATH_TEMP
```

```
CLRF    PCLATH
```

恢复现场语句：

```
MOVF    PCLATH_TEMP, W
```

```
MOVWF   PCLATH
```

```
SWAPF   STATUS_TEMP, W
```

```
MOVWF STATUS
SWAPF W_TEMP, F
SWAPF W_TEMP, W
```

### 3.1.3 人机界面管理

智能仪表的管理界面是人机联系的重要通道，是操作者管理仪表的主要方式。目前智能仪表的按键定义一般都比较简单，属于一键一义，即一个按键代表一个确切的命令或一个数字。编程时只要根据当前按键的编码直接分支到相应的处理模块入口，而无需知道在此之前的按键情况。

本文定义了四个按键：POWER、HOLD、INT、ZERO，分别为电源、保持、中断显示、清零键。各个按键的功能描述如下：

- POWER 电源开关，负责整个系统的电源供给
  - HOLD 保持当前显示界面，以供操作者查看结果
  - INT 定义为进入平均值和均方差的计算以及显示
  - ZERO 清零键，按下该键后可使仪表输出零
- 详细程序实现过程可参见下一节。

## 3.2 功能程序设计

### 3.2.1 光指针的显示

近年来，市面上出现了一种光指针/数字LCD双显仪表，该仪表在数显功能的基础上增加了光指针功能，是一种集模拟显示和数字显示优点于一身的新型显示技术，既具有指针式表头的一目了然，又具有数字式表头的直观、精确。但这种双显仪表的主要显示器件——LCD面板都是开模特制的，而本系统所要讨论的是如何利用软件模拟形成光指针的方法。

本系统中利用PIC单片机的程序控制实现在LCD面板上同步显示光指针和数字量。LCD面板的第一行作为光指针显示行，利用16个“■”字符来表示满量程，在此采用测量电压值作为测试方法，则当满量程时表示电压值为5V，即每一个“■”字符表示的电压值为312.5mv。显示效果如图3-1所示，图中为所测电压从小到大变化而得到的三张双显效果图。

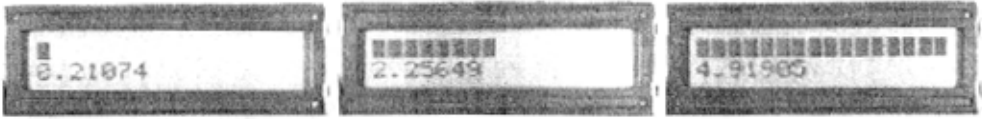


图3-1 双显效果图

程序设计说明：初始化LCD，将8位二进制采样数据除以16（因为液晶显示器每行有16个5×7点阵，对于一个8位二进制数，每一个点阵代表十进制数16），得到的结果存入DIVRES和DIVRES1，将DIVRES1（当前值）与DIVRES2（前一值）比较，相等则显示不变，直接进行数据处理，否则调用光指针显示子程序，然后再进行数据处理，如图3-2。调用光指针显示子程序时，先显示DIVRES个“■”字符，然后将前一值DIVRES2与当前值DIVRES1相减，结果存入DIVRES2，若DIVRES1小于DIVRES2，则显示DIVRES2个“ ”字符，否则子程序返回，如图3-3所示。

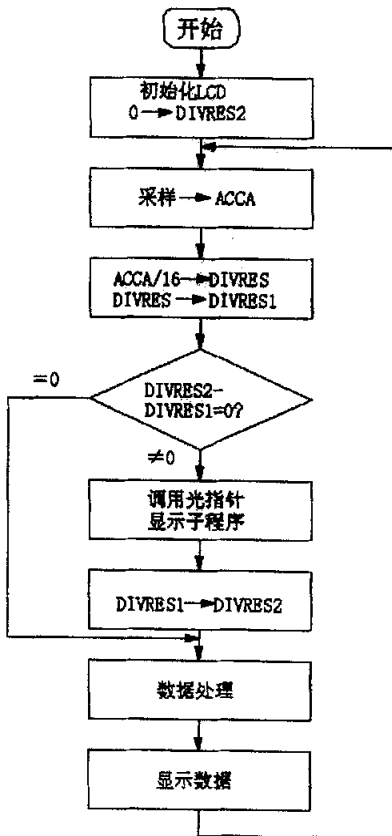


图3-2 光指针主程序框图

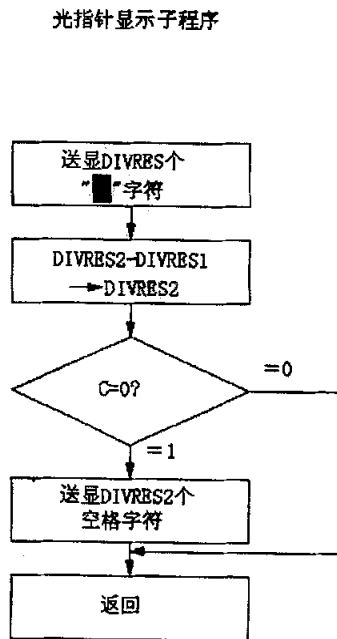


图3-3 光指针显示子程序框图

部分程序如下:

```

.....
DIVAB   CLRF           DIVRES           ;DIVRES/16结果放入DIVRES中
        MOVLW        0F0H
        ANDWF        ACCA,F
        SWAPF        ACCA,W
        MOVWF        DIVRES
        INCF          DIVRES,F
        MOVF          DIVRES,W
        MOVWF        DIVRES1
        SUBWF        DIVRES2,W         ;DIVRES1与DIVRES2比较
        BTFS        STATUS,Z
        RETURN        ;相等,返回进行数据处理
        CALL         DISPLAY           ;不等,调用光指针显示子程序
OVER    MOVF          DIVRES1,W         ;DIVRES1的值赋给DIVRES2,
        MOVWF        DIVRES2         ;用做比较
        RETURN        ;返回进行数据处理
DISPLAY MOVLW        080H             ;光指针显示子程序
        CALL         SendCmd          ;调用LCD命令子程序
DISLOOP1 MOVLW       B'11111111'
        CALL         putcLCD          ;调用LCD数据子程序
        DECFSZ       DIVRES,F
        GOTO         DISLOOP1
        MOVF          DIVRES1,W
        SUBWF        DIVRES2,F         ;DIVRES2-DIVRES1结果存
        BTFS        STATUS,C         ;入DIVRES2
        GOTO         OVER             ;DIVRES2小于DIVRES1
DISLOOP2 MOVLW       ' '              ;DIVRES2大于DIVRES1
        CALL         putcLCD
        DECFSZ       DIVRES2,F

```

```
GOTO      DISLOOP2
RETURN
.....
```

### 3.2.2 光指针的改进

在上一节中的光指针技术虽然具有先进性，但是其在具体显示方面还不够精细，显示比较的粗糙，它是将 $5 \times 7$ 点阵作为一个单元，显示器一行可以显示16个字符，对于一个八位二进制数而言，一个字符代表二进制数为“10000”，因此显示比较粗略，所以，在后来的研究中作者考虑将 $5 \times 7$ 点阵分成5个 $1 \times 7$ 单元，也即，将显示精度提高了五倍，使得光指针显示变得更加细腻、精确。

要想将 $5 \times 7$ 点阵分成5个 $1 \times 7$ 单元，就必须要求LCD能够显示出“|”、“||”、“|||”、“||||”“|||||”这样的五个字符。一般而言LCD模块的显示内容是由用户决定的，对字符型而言，有些显示内容是可以直接从内部CGROM中调用的，而有些CGROM中没有的专用字符或符号，则只能由用户自定义。本文的光指针符号就是属于这类自定义符号。

#### 3.2.2.1 字符代码（DDRAM DATA）及 CGRAM 首地址关系

KS0066UP-00的字符发生器RAM（CGRAM）可存储8个自行编辑的任意 $5 \times 7$ 点阵字符图形。自编程字符图形代码为00H—07H（或08H—0FH），共8个字符图形代码，为了在液晶显示屏上显示存储在CGRAM中的自行编程的字符图形，必须把它们的代码写入DDRAM中。字符代码、CGRAM地址、字符图形之间的关系如表3-1所示。表中的“×”表示既可以为“0”，也可以为“1”。表中的字符图形为“中”，它是由8行0与1的组合数据表示出来的，此8行数据存储在CGRAM的8个单元中，这8个单元的CGRAM地址表示每个数据存放在CGRAM中的位置，它们的0位—2位与字符图形各行的序号相对应，而这8个单元的地址的3位—5位与DDRAM中的字符代码0位—2位相同，表示这8个CGRAM单元是用来存放同一个字符代码所表示的字符图形数据的。一个DDRAM中的字符代码00H或08H就确定了一个执行编程字符“中”的图形。换言之，要想取出CGRAM中的图形数据，必须向DDRAM中送入4位—7位的字符代码，例如对表3-1来说，要取出CGRAM中的字符图形“中”的数据，就必须向DDRAM中送入代码“00H”或“08H”。

表 3-1 字符代码、CGRAM 地址、字符图形之间的关系

字符代码 (DDRAM数据)								CGRAM地址								字符图形 (CGRAM数据)							
7	6	5	4	3	2	1	0	5	4	3	2	1	0	7	6	5	4	3	2	1	0		
高位				低位				高位				低位				高位				低位			
												0 0 0				x x x 0 0 1 0 0							
												0 0 1				x x x 0 0 1 0 0							
												0 1 0				x x x 1 1 1 1 1							
0 0 0 0 x 0 0 0								0 0 0 0 1 1				x x x 1 0 1 0 1											
								1 0 0				x x x 1 0 1 0 1											
								1 0 1				x x x 1 1 1 1 1											
								1 1 0				x x x 0 0 1 0 0											
								1 1 1				x x x 0 0 1 0 0											

### 3.2.2.2 确定光指针图形数据及其字符代码

要精细地模拟光指针,就要先定义五个图形“|”、“||”、“|||”、“||||”“|||||”,但其中“|||||”图形可以用 CGROM 中代码为“FF”的字符代替。这样,只需要定义四个字符“| || ||| ||||”其中,“|”图形的 CGRAM 数据为 8 个“10H”,“||”图形为 8 个“18H”,“|||”图形为 8 个“1CH”,“||||”则为 8 个“1EH”,共 32 个字节,由于 KS0066UP-00 模块只给出 CGRAM 8 个字符代码 00H-07H,所以四个图形的字符代码自定义为:

“|” : 00H ; “||” : 01H ; “|||” : 02H; “||||” : 03H

根据 CGRAM 地址确定法,6 位地址中的高 3 位为字符代码的低 3 位,即 00H-07H,地址的低 3 位是行计数器的内容,变化范围从 00H 到 07H。由于图形数据是从第一行依次写入的,所以在设置 CGRAM 地址时,把这三位置“0”,这样用户定义的 8 个字符的字符数据在 CGRAM 的首地址固定为表 3-2 所示

表 3-2 字符代码与 CGRAM 地址对应表

字符代码	CGRAM地址	字符代码	CGRAM地址
00H	00H	01H	08H
02H	10H	03H	18H
04H	20H	05H	28H
06H	30H	07H	38H

### 3.2.2.3 CGRAM 地址设置指令

KS0066UP-00 模块的 CGRAM 地址设置指令格式如表 3-3 所示:

表 3-3 设定 CGRAM 地址

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0

表中 AC0-AC5 为自定义字符在 CGRAM 中的首地址, 因此设置某一字符字模数据首地址的指令代码为:

指令代码=40H + 首地址

因为光指针显示图形的字符代码为 00H-03H, 所以 CGRAM 地址的起始地址为 00H, 则指令代码为 40H。

### 3.2.2.4 把图形数据写入 CGRAM 单元内

本文所采用单片机的是 PIC 16F877 型, 其中要用到的寄存器为:

PONITER1-数据量 PONITER2-查表偏移量

数据写入及调用的子程序如程序 3-4 所示。

#### 程序 3-4

```

...
初始化
...
MOVLW    40H           ; 设置 CGRAM 地址为 0, 这样四个图形的
                        显示代码就为 00H-03H
CALL     SendCmd       ; 调用指令发送子程序
CALL     Delay         ; 延时, 以等待忙指令
MOVLW    20H
MOVWF   PONITER1
MOVLW    00H
MOVWF   PONITER2
LOOP MOVF   PONITER2, 0
CALL     CONVERT1     ; 调用查表子程序
CALL     putcLCD      ; 调用数据发送子程序

```



```

INCF    PONITER2, 1      ; 查表偏移量加一
CALL    Delay           ; 调用延时子程序
DECFSZ  PONITER1, 1
GOTO    LOOP           ; 循环发送数据
RETURN

```

#### CONVERT1

```

ADDWF   PCL, 1          ; 偏移量加当前 PCL 值

```

#### TABLE1

```

DB:  10H, 10H, 10H, 10H, 10H, 10H, 10H, 10H,
DB:  18H, 18H, 18H, 18H, 18H, 18H, 18H, 18H
DB:  1CH, 1CH, 1CH, 1CH, 1CH, 1CH, 1CH, 1CH
DB:  1EH, 1EH, 1EH, 1EH, 1EH, 1EH, 1EH, 1EH
MOVLW   80H            ; 初始化第一行 (设置 DDRAM地址)
CALL    SendCmd
MOVLW   00H            ; 显示第一个光指针图形
CALL    putcLCD
MOVLW   01H            ; 显示第二个光指针图形
CALL    putcLCD
MOVLW   02H            ; 显示第三个光指针图形
CALL    putcLCD
MOVLW   03H            ; 显示第四个光指针图形
CALL    putcLCD
...

```

### 3.2.2.5 光指针的形成

光指针的形成是通过软件控制来实现的，其程序框图如图 3-1 所示。程序框图中的 ACCBLO、ACCCLO 分别为存放商和余数的寄存器，“FF”字符为液晶模块 CGROM 中字符代码为“FF”的字符“||||”，它是由 8 个 FFH 字节组成的，与自定义的四个图形共同构成形成光指针所需的五种图形。本程序的光指针是通过外部的数据变化来体现的，涉及到数据的采集、A/D 转换以及液晶输出等一系列过程。

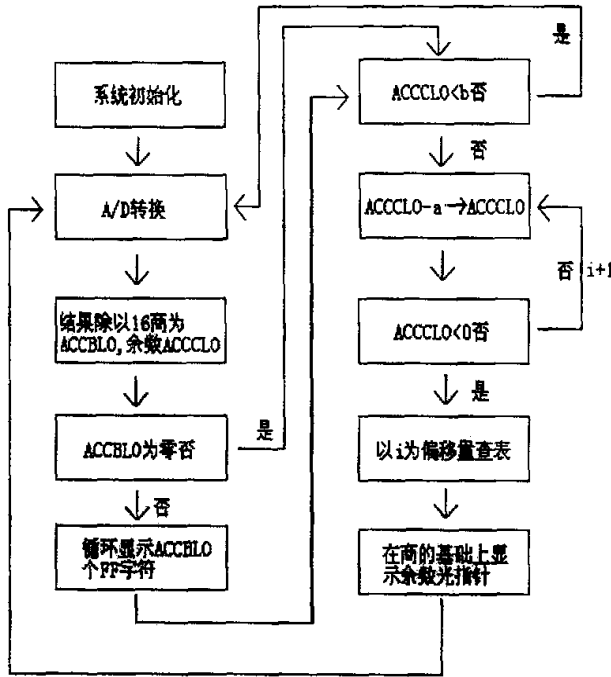


图 3-4 光指针形成程序框图

程序实现过程中，首先要进行 LCD 的初始化，包括自定义字符的写入程序。一次 A/D 转换后得到的数据经过数学处理（在本文中除以 16，因为所用的 LCD 面板一行可以显示 16 个字符），得到商以及余数，商就是要显示的主指针（图形即为上面所提到的“FF”字符—“||||”），而余数则为尾数指针（即为上面定义的四个图形之一），用来提高精度，本文中由于尾数指针的存在，使得精度提高了五倍。显示流程为先显示主指针，显示的个数为商的大小，接着显示尾数指针。框图中的常数 a 为最低精度，用来确定尾数，当尾数小于 a 时就不显示，认为余数为零；当大于 a 时就循环减 a，并将循环次数保存在 I 中。因为最小精度 a 为“||||”字符所代表的数字大小的五分之一，因此 I 不会大于五，然后以 I 为查表偏移量进行查表，从而确定尾数的图形。具体的对应关系如表 3-4 所示：

表 3-4 尾数图形与偏移量对应关系

尾数图形					
I	1	2	3	4	5

### 3.2.2.6 显示效果

显示效果如图 3-5 所示, 照片中为所测电压从小到大变化而得到的三张双显效果图。图中光指针下方的数字为对应的电压值大小, 可以验证光指针的正确性。通过测试和计算, 满屏时输入电压为 4.918V, LCD 面板的第一行用来显示光指针, 可以显示 16 个 5×7 字符, 即共可以显示 80 列 (16×5) 光指针, 最小分辨率可达到 61mv, 也就是说精度达到了 1.25%, 可以非常细腻地反映出电压值的变化情况。

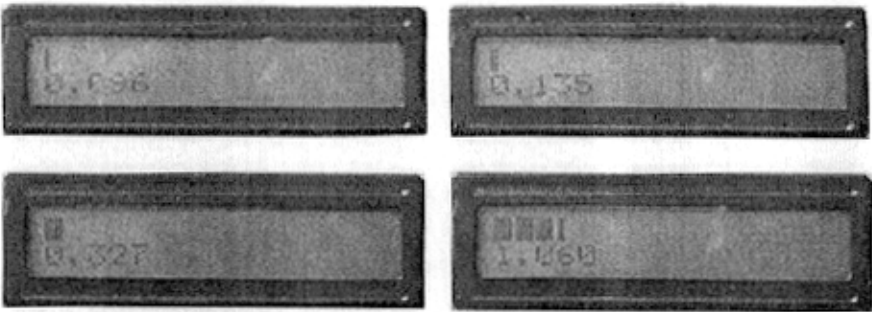


图 3-5 效果显示图

### 3.2.3 张力平均值、均方差的计算及其软件实现

由于在实时检测中张力的波动情况非常复杂, 单单一个即时张力值在某种程度上不能完全反映出当时的变化情况, 因此, 本课题中对张力的测量增加了两个参数: 张力平均值和均方差。

平均值可以用来度量张力的分布位置, 即表示张力平均水平的高低, 用数学公式表示为:  $\bar{x} = \frac{1}{N} \sum_{i=1}^n x_i$ ; 均方差可以用来度量张力分布的离散程度, 即反映张力波

动的绝对值, 数学表达式为  $\delta = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$ 。

通过这两个参数的设置, 可以更加方便地对张力进行测量和控制。

在本课题中, 采用 RB0 脚的外部中断功能进行切换显示均方差和平均值。当按下 RB0 脚对应的按键时, 进入中断处理: 开始对自此以后的 255 个点进行采样求均方差以及平均值, 并通过 LCD 显示出来。

另外, 为了使算法简单化, 在程序中对算法进行了优化, 其数学过程如下:

求平均值  $\bar{x}$  和均方差  $\delta^2$

$$\begin{aligned}\bar{x} &= \frac{1}{N} \sum_{i=1}^n x_i^2 \quad (N=255) \\ \delta^2 &= \frac{1}{N} \sum (x_i - \bar{x})^2 \\ &= \frac{1}{N} \sum (x_i^2 - 2x_i\bar{x} + \bar{x}^2) \\ &= \frac{1}{N} (\sum x_i^2 - 2\bar{x} \sum x_i + \sum \bar{x}^2) \\ &= \frac{1}{N} \sum x_i^2 - 2\bar{x} \cdot \frac{1}{N} \sum x_i + \frac{1}{N} \sum \bar{x}^2 \\ &= \frac{1}{N} \sum x_i^2 - 2\bar{x} \cdot \bar{x} + \bar{x}^2 \\ &= \frac{1}{N} \sum x_i^2 - \bar{x}^2\end{aligned}$$

具体的程序流程图如图 3-6 所示，在主程序中，循环对当前的张力进行测量和显示，同步显示出数字量和光指针。当要显示均方差和平均值时，按动对应按键，进入中断，进行均方差和平均值的计算和显示。

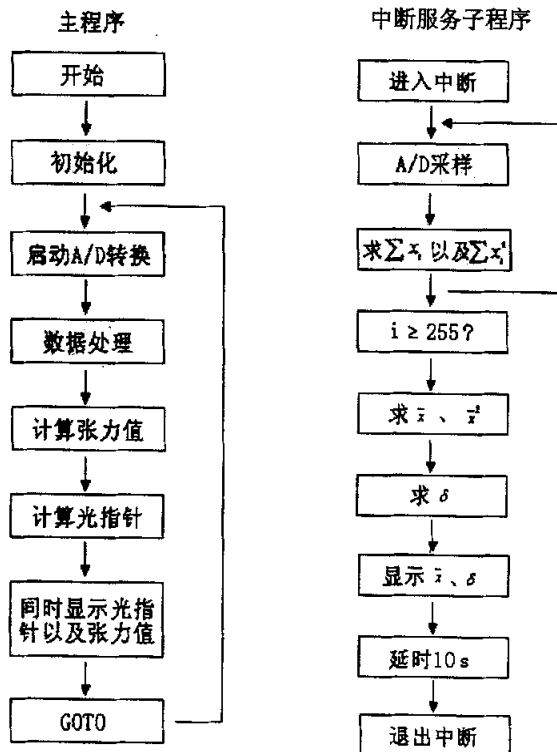


图 3-6 均方差、平均值计算及显示程序流程图

### 3.2.4 保持功能的实现

由于动态张力的易变性,操作者有时很难及时读出张力值,所以仪表的读数保持功能就显得非常必要。在本课题中,保持功能对应的外部按键为“HOLD”键,通过对该键的控制可以实现显示的保持功能。具体实现方法是在显示步骤之后加上一条引脚电平判断语句:当“HOLD”键按下时,对应的单片机引脚由高电平变化为低电平,从而触发保持功能,使程序停留在显示语句上,始终显示当前的张力值,不进行显示更新,直到释放“HOLD”键,程序才继续进行下一个张力的显示。

具体的实现语句如下所示:

```
HOLD
      BTFSS   PORTD,2
      GOTO   HOLD
      GOTO   Adstart
```

### 3.2.5 清零功能的实现

测量仪表最常见的故障是长期工作过程中,测量值会发生温漂及测量特性不稳定,这是仪表的零点漂移和增益变化造成的。零漂即是指当输入为零时,仍有输出值的现象。仪表的零点漂移既可能来自传感器,也可能来自放大器。就零点漂移的性质而言,它可能是系统误差,也可能是随机误差。零点漂移现象的产生严重影响了系统的精度,因此要对测量仪表的输出零点进行调整<sup>[16]</sup>。

目前,普遍的调零方法主要有两种:一种是将机械式多圈电位器放在仪表的控制面板上,通过旋转电位器来实现手动调整零点;另一种方法是通过仪表内部的单片机控制 A/D 转换器,采集当前的偏移数值作为零点,存入内部存储器中,以后每次采样减去零点。上述方法中前者是传统的方法,实现起来也比较简单,且成本较低,但是由于机械式电位器本身的特点,使得零点的调节有时难免不够准确。本文将这两种调零方法进行了结合,既采用了外部的电位器调零,又采用了内部的软件调零的方式,从而确保零点输出的准确性。

#### 3.2.5.1 外部手动调整零点

电路原理图如图 3-7 所示,在传感器的输出端,添加一个机械式可调多圈电位器,通过对其调节,可使输入到放大器中的信号实现初步的零点。

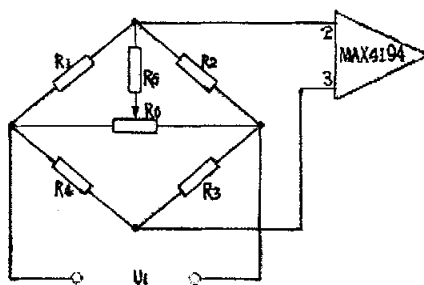


图 3-7 外部调零原理图

### 3.2.5.2 内部软件调零

内部软件调零主要是采用单片机控制 A/D 转换器，采集当前的偏移数值作为零点，存入内部存储器中，以后每次采样减去零点。具体实现步骤如下：

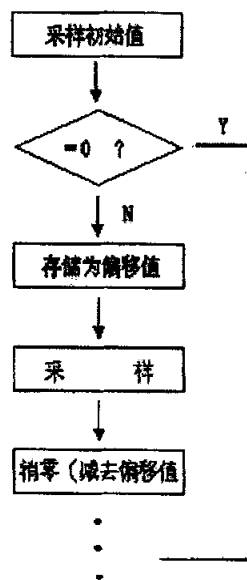


图 3-8 软件调零原理图

### 3.3 软件的标定设计

在测量过程中，一般把各种非电量信号先变为电量的模拟信号，而模拟电信号

则被转换成数字信号。通常我们希望显示器能够直接显示出该测量对象的单位，如压力 Pa、温度℃等等。所以要将 A/D 转换后的代码（它们一般不等于原有参数的量纲值，只表示出相对大小）变换成带有量纲的值。这种转换叫做标定或称为工程量的转换。利用程序来进行转换叫做软件标定。

软件标定只需要用程序进行测量值与工程量之间的转换，较之用硬件调节各种电位器来说要显得更简单、方便、灵活。本文的软件标定采用的是线性值公式法。这种标定的前提是被测参数值与 A/D 转换的结果代码之间在数量上是呈线性的。若输入信号为 0，但 A/D 输出不是 0，则变换公式为

$$Y = (Y_{\max} - Y_{\min})g(X - N_{\min}) / (N_{\max} - N_{\min}) + Y_{\min}$$

式中  $Y$  — 参数测量值（有量纲）；

$Y_{\max}$  — 参数测量最大值；

$Y_{\min}$  — 参数测量最小值；

$N_{\max}$  — 量程终点对应 A/D 转换后的值；

$N_{\min}$  — 量程起点对应 A/D 转换后的值；

$X$  — 参数测量值对应 A/D 输出的测量值或数字滤波后的真值。

本课题中被测对象为转换后的电压值，单位  $mv$ ，测量范围为  $0 \sim 5000mv$ ，对应的 A/D 输出代码为  $0 \sim 255$ ，则公式可以转换为： $y = (5000/255) \cdot x_i$ ，其中  $y$  为转换后的电压值， $x_i$  为每个即时转换的 A/D 代码（转换为十进制后进行计算）。

此外，还进行了电压值到张力值的标定，从而在 LCD 上直接显示出张力值。具体的标定过程是用极细的丝线材料悬挂已知重量，示意图如图 3-9 所示，最终确定形成的最小有效精度为 1CN。

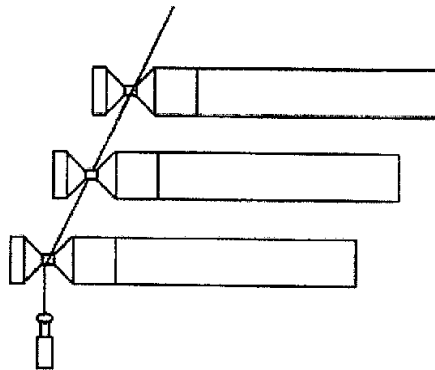


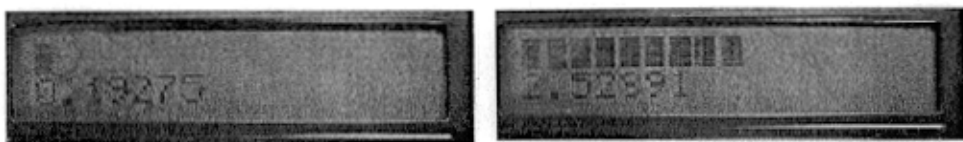
图 3-9 标定示意图

## 第四章 测试与分析

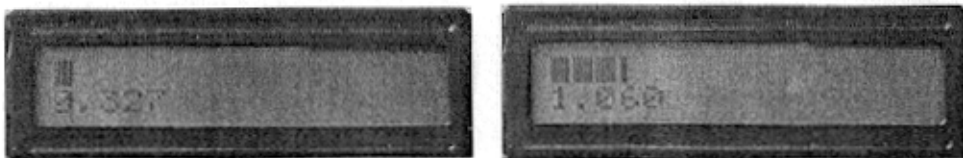
### 4.1 光指针显示对比

本文光指针的形成先后经历了两个过程，现在所采用的是第二种设计。

第一种设计同样是利用PIC单片机的程序控制实现在LCD面板上同步显示光指针和数字量。LCD面板的第一行作为光指针显示行，利用16个“■”字符来表示满量程，在此采用测量电压值作为测试方法，则当满量程时表示电压值为5V，即每一个“■”字符表示的电压值为312.5mv。显示效果如照片4-1所示，照片中为所测电压从小到大变化而得到的双显效果图。



照片 4-1 双显效果图 1



照片 4-2 双显效果图 2

照片 4-2 为第二种设计所得到的四张显示效果图，通过与照片 4-1 的对比可以很明显的看出第二种的显示精度比第一种提高了 4 倍，显示更为细腻。

### 4.2 张力测试对比

德国施密特 (SCHMIDT) 公司是世界生产张力仪规格最多最全的专业厂家，有着 50 多年的生产研发经验，以杰出的质量和精度闻名于世，目前已有超过 140000



只施密特张力仪在全世界范围内广为使用。

将自行研发的智能双显张力仪与德国施密特 (SCHMIDT) 公司的 ETM-200 型张力仪进行了对比测试, 其方法和结果如下。

方法: 采用悬挂砝码的方法对张力仪进行对比测试

条件: 室温、无风、静态

数据如表 4-1 所示。

表 4-1 静态张力读数对比

序号	砝码质量 w (单位: 克)	本张力仪读数 (单位: CN)	ETM-200 读数 (单位: CN)
1	0	0	0
2	1	1.1	1
3	2	2.4	2
4	5	4.9	4
5	10	11.2	10
6	25	26.1	26
7	50	49.8	52
8	100	98.2	102

由于砝码采用如图 3-6 所示的方式悬挂, 根据公式:

$$\begin{aligned}
 F &= G = mg \\
 &= wg \\
 &= w \times 10 \times 100 \text{cN} / 1000 \text{g} \\
 &= w
 \end{aligned}$$

(注: 为了方便计算取重力加速度  $g=10\text{N/kg}$ )

因此, 丝线上的张力值理论上应该近似等于  $w$ 。

分析:

建立如图 4-3 所示的线性度对比图, 图中蓝色线条为自制双显张力仪, 紫红色线条为 ETM-200 所表现出的线性线条,  $x$  轴是悬挂的已知砝码重量,  $y$  轴为两种张力仪的测得值。

从表 4-1、图 4-3 可以看出, 在静态测试下, 本课题研制的张力仪所表现的线性度和准确度都较好, 但在张力较小的情况时 (1-2cN) 偏差较大。

原因分析:

本张力仪采用的传感器虽然经过软件和硬件的调零，但在零点附近还是会存在一定的漂移。由于这些漂移的存在，靠近零点的张力值会生漂移。另外，随着张力值的增大，由于计算的误差，会产生一定的累积效应，从而导致当张力增加到一定的程度时偏差会增大。因此，将量程暂定为 100cN，以免发生太大的测量误差。

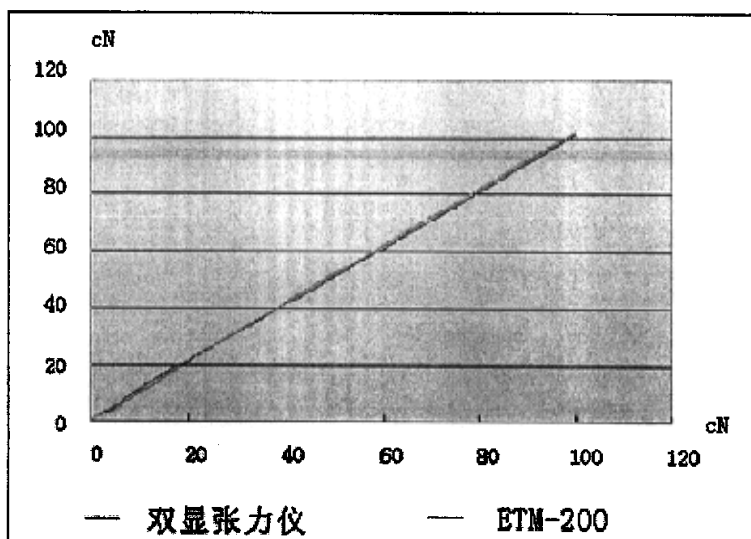


图 4-3 线性对比图

### 4.3 主要性能指标

根据传感器的参数说明手册，所选用的应变片的固定频率为 1200Hz，因此双显张力仪理论工作频率为 120Hz。

如表 4-2 为目前无梭织机的最高车速，由于张力的波动很大程度上是由于织机转动引起的，所以可以推算出织机上的张力变化频率最高在 25Hz 左右，因此我们研究的双显张力仪在理论上是可以满足当前的测试频率要求的。

表 4-2 无梭织机车速对比<sup>[19]</sup>

转速	片梭织机	剑杆织机	喷气织机	喷水织机
r/min	420	700	1300	1500

模拟量最大输出 4000mV

有效张力测定范围 100cN

线性度  $\pm 1\%$

最高工作频率 120Hz

## 结 论

本课题采用了美国 MAXIM 公司先进的仪表运放以及美国 Microchip 公司的单片机, 制作了智能型的双显张力仪, 克服了现有张力仪一些不足之处, 经过测试基本达到了预期的效果。具体有以下几点:

### 1. 系统具有创新性

本系统首次将模拟显示和数字显示有机结合起来, 利用自行开发的在普通 LCD 面板上显示的光指针, 有效结合了数字显示和指针显示的优点。既可以直观地显示精确的数字量, 又可以一目了然动态地显示出张力的变化趋势, 提高了对被测信号波动变化倾向的判断能力, 可以为控制张力、配置合适的工艺以及开发产品提供可靠的依据。

### 2. 仪表具有稳定性和可靠性

通过实验可知, 不论是稳定性还是可靠性, 都与国外先进仪表的测试数据具有很高的相关性。

### 3. 低功耗

由于采用了美国 MAXIM 公司的性价比很高的仪表运放 MAX4194, 以及电压模块 MAX6165, 再加上美国 Microchip 公司功耗极低的 PIC16F877, 整个系统的功耗非常之低( $<25\text{mW}$ ), 体积小, 而且系统也更加的稳定。

### 4. 磨损减少

采用了金属导丝头, 变滑动摩擦为滚动摩擦, 既减少了导丝器的磨损, 又解决了静电积聚的问题。

但是, 由于本人的知识范围及研究能力有限, 再加上试验条件和时间的限制, 目前本课题尚存在一些不足之处, 有待进一步完善, 具体有:

(1) 传感器难于购买, 因而没有大量的传感器样品可供选择, 所选用的传感器依然存在漂移现象, 虽然在硬件和软件方面做了大量的处理, 但难免对系统的性能造成了一定影响。

(2) 条件和时间的限制, 没有进行在线的动态测试。如有条件宜进行大量的在

线测试。

总之，课题取得了一定成绩，同时也存在着不足。成绩凝聚了导师耐心细致的教导，各位老师和同学的关心和帮助；不足说明了各种因素的限制。真心希望能对“智能双显张力仪”做进一步完善，使张力的检测能够更方便、更准确同时也能得到更加广泛的应用。

## 参 考 文 献

- [1] 浙江丝绸工学院 苏州丝绸工学院编. 丝织学(上册)第二版, 北京: 纺织工业出版社, 1993年12月. 77
- [2] 胡三贵, 卢竹萍. 单丝张力仪测量范围的改进. 计量技术, 1994, 4: 35
- [3] 王进岑. 丝织手册(上册)第二版. 北京: 中国纺织出版社, 2000. 709
- [4] 王绍斌, 张茂林, 陈硕. 经纱张力的动态测试. 北京纺织, 20(6): 27
- [5] 本德萍, 李景社, 谷炎武. 络纱张力动态测试系统的研制. 西北纺织工学院学报 2000, 14(3): 397
- [6] 张利春. 电测量指示仪表防静电影响的技术措施. 电子工艺技术, 2000, 21(5): 216
- [7] 汪信远. 机械设计基础(第三版). 北京: 高等教育出版社, 2002. 241
- [8] 李江华, 谢红, 于蕾. 仪表放大器技术初探与应用. 应用科技, 2001, 28(9): 10
- [9] MAX4194 外文使用说明
- [10] MAX6165 外文使用说明
- [11] 李学海. PIC 单片机实用教程——基础篇. 北京: 北京航空航天大学出版社, 2002. 12
- [12] 李学海. PIC 单片机实用教程——提高篇. 北京: 北京航空航天大学出版社 2002. 172
- [13] PIC 单片机在线调试器使用说明书, 福州高奇晶圆电子科技有限公司
- [14] 李宏, 张家田等. 液晶显示器件应用技术(第一版). 机械工业出版社. 2004. 105
- [15] 李志全, 张淑清等. 智能仪表设计原理及其应用. 国防工业出版社, 1998. 152 - 153
- [16] 李汉江, 肖江淑. 智能仪表的自动调零分析. 湖南工程学院学报, 2001, 11(2): 21-23
- [17] 刘曙光等. 智能光指针/数字双显仪表的研究. 仪表技术., 1997, 6: 8
- [18] 陈杰, 黄鸿. 传感器与检测技术. 高等教育出版社. 2002. 30
- [19] 高卫东, 荣瑞萍等. 现代织造工艺与设备. 北京: 中国纺织出版社, 2003. 245

## 攻读学位期间本人出版或公开发表的论著、论文

- [1] 柏松乔, 林峰, 陈庆官. 利用普通液晶显示屏显示光指针的一种方法. 苏州大学学报(工科版), 2004, 12: 35-38
- [2] 柏松乔, 林峰, 陈庆官. 利用 CGRAM 形成 LCD 光指针的一种方法. 苏州大学学报(工科版), 拟录用.

## 附录

```

list          p=16f877
include      "p16f877.inc"

ACCALO      EQU      20h
ACCAHI      EQU      21h
ACCBLO      EQU      23h
ACCBHI      EQU      24h
ACCCLO      EQU      26h
ACCCHI      EQU      27h
ACCDLO      EQU      28h
ACCDHI      EQU      29h
ACCEHI      EQU      57H
ACCELO      EQU      58H
ACCFHI      EQU      59H
ACCFLO      EQU      5AH
TEMP        EQU      2Ah
SIGN        EQU      2Bh
Temp        EQU      2cH
TempH       EQU      2DH
TempM       EQU      2EH
TempL       EQU      2FH
TempMM      EQU      30h
A_TEMPLO    EQU      47H
A_TEMPHI    EQU      48H
BXHI        EQU      49H
BXLO        EQU      4AH

count3      equ      31h
temp        equ      45h
s0          equ      32h
s1          equ      33h
s2          equ      34h
r0          equ      35h
r1          equ      36h
r2          equ      37h
r0lo        equ      38h
r0hi        equ      39h
r1lo        equ      3ah
r1hi        equ      3bh

```

```

r2lo      equ    3ch
r2hi      equ    3dh
r3        equ    4bh
r3hi      equ    4eh
r3lo      equ    4fh
r4        equ    50h
r4hi      equ    51h
r4lo      equ    52h
DACCAHI   equ    53h
DACCALO   EQU     54H
DACCDHI   EQU     55H
DACCDLO   EQU     56H

tempr0    equ    3eh

carry      equ    0
fsr        equ    04h
status     equ    03h
portd      equ    08h
adcon0     equ    1fh
option_reg equ    81h
trisd      equ    88h
adcon1     equ    9fh
intcon     equ    0bh
t0if       equ    2
go         equ    2
adif       equ    6
adresh     equ    1eh
pir1       equ    0ch
tmr0       equ    01h
LBS        EQU     0
MSB        EQU     7
TRUE       EQU     1
YES        EQU     1
FALSE      EQU     0
SIGNED     EQU     TRUE
CARRY      EQU     0
Z_BIT      EQU     2
DCARRY     EQU     1
rp0        equ    5
; 定义 LCD 数据& 控制 I/O 口
LCD_DATA   equ    PORTB

```



```

LCD_CNTL1    equ PORTB
LCD_CNTL2    equ PORTA

; 定义 LCD 控制相应的 I/O 引脚
RS            equ 1
E            equ 5
; LCD 模块命令
DISP_ON      EQU    0x00C ; 开显示
DISP_ON_C    EQU    0x00E ; 开显示, 开指针
DISP_ON_B    EQU    0x00F ; 开显示, 开指针, 开闪烁
DISP_OFF     EQU    0x008 ; 关显示
CLR_DISP     EQU    0x001 ; 清显示的操作数
ENTRY_INC    EQU    0x006 ; 操作数
ENTRY_INC_S  EQU    0x007 ; 操作数
ENTRY_DEC    EQU    0x004 ; 操作数
ENTRY_DEC_S  EQU    0x005 ; 操作数
DD_RAM_ADDR  EQU    0x080 ; 初始化第一行显示的操作数
DD_RAM_UL    EQU    0x0c0 ; 初始化第二行显示的操作数

; 设置显示相关寄存器
Byte         EQU    3FH
Count        EQU    40H
Count1       EQU    41H
Count2       EQU    42H
LCD_DATA1    EQU    43H
PORTDB       EQU    44H
DISCOUNT1  EQU    46H
DISCOUNT    EQU    5BH
BE_TEMPHI    EQU    5CH
BE_TEMPLO    EQU    5DH
BF_TEMPHI    EQU    5EH
BF_TEMPLO    EQU    5FH
PINGJUNHI    EQU    60H
PINGJUNLO    EQU    61H
s3           EQU    62H
ACCA         EQU    63H
DIVRES       EQU    64H
DIVRES1      EQU    65H
DIVRES2      EQU    66H
FIG          EQU    67H
TEMPB        EQU    68H
TEMPC        EQU    69H

```

```

WTEMP EQU 6AH
TEMPAB EQU 6BH
PONITER1 EQU 6CH
PONITER2 EQU 6DH
ZERO EQU 6EH
INTE equ 4
INTF equ 1
W_TEMP equ 7fh
STATUS_TEMP equ 6fh
GIE EQU 7
ZERO_TEMPEQU 70H

org 000h
nop
GOTO STAMAINS
ORG 0004H
GOTO SERV
STAMAINS
bcf status,rp0
clrf portd
BSF INTCON,GIE
BSF INTCON,INTE ;开放 INT 中断
bsf status,rp0
movlw b'11111111'
movwf trisd
call InitLCD
CALL SETT

goto START

;.....A/D 转换.....
START
bcf status,rp0
movlw b'01000001'
movwf adcon0
bsf status,rp0
movlw b'00001110'
movwf adcon1
bcf status,rp0
AD1 bsf adcon0,go
wait
btfs p1r1,adif
goto wait

```

```

        movf      adresh,w
        movwf    ZERO
        MOVWF    ZERO_TEMP

ADstart
        bsf      adcon0,go
wait1
        btfss   pir1,adif
        goto    wait1
        movf    adresh,0
        movwf   ACCALO
        MOVWF   ACCA

        MOVF   ZERO,0
        BCF   STATUS,C
        SUBWF  ACCALO,0
        BTFSC STATUS,0
        GOTO  SUB1
        GOTO  SUB2

SUB2    MOVF   ACCALO,0
        MOVWF  ZERO_TEMP
        SUBWF  ZERO,0
        MOVWF  ACCA
        MOVWF  ACCALO

SUB1
        MOVWF  ACCA
        MOVWF  ACCALO
;        BTFSS PORTD,1
;        GOTO  AD1
        goto  MAP

;;.....
;设定 CGRAM 的地址以及内容
;起始地址为 00H,指令代码为 01,所以总的指令代码为 40H
;共 4 个自定义字符, 每个字符由 8 个字节构成,共 32 个字节
;利用查表进行设置内容
SETT
        MOVLW  40H    ;设定 CGRAM 的地址
        CALL   SendCmd
SEND1  MOVLW  20H    ;设定总字节数 (32 字节)

```

```

MOVWF  PONITER1
MOVLW  00H    ;初始化查表偏移量
MOVWF  PONITER2

SENDLOOP MOVF  PONITER2,0
        CALL   CONVERT1 ;调用查表子程序
        CALL   putcLCD
        INCF   PONITER2,1 ;查表偏移量加一
        call   Delay
        DECFSZ PONITER1,1 ;总字节数减一
        GOTO   SENDLOOP
        RETURN

CONVERT1
        ADDWF  PCL,1      ;将查表偏移量与当前PCL值相加，转到相应的程序
TABLE1 RETLW  10H
        RETLW  10H
        RETLW  10H
        RETLW  10H
        RETLW  10H
        RETLW  10H
        RETLW  10H
        RETLW  10H ;定义第一个字符
        RETLW  18H
        RETLW  18H
        RETLW  18H
        RETLW  18H
        RETLW  18H
        RETLW  18H
        RETLW  18H
        RETLW  18H ;定义第二个字符
        RETLW  1cH
        RETLW  1cH
        RETLW  1cH
        RETLW  1cH
        RETLW  1cH
        RETLW  1cH
        RETLW  1cH
        RETLW  1cH ;定义第三个字符
        RETLW  1EH
        RETLW  1EH
        RETLW  1EH

```

```

        RETLW    1EH
        RETLW    1EH
        RETLW    1EH
        RETLW    1EH
        RETLW    1EH    ;定义第四个字符
;.....
;.....乘法.....
;A/D 结果放到 ACCALO 中
;结果存在 ACCBHI,ACCBLO,ACCCHI,ACCCLO 中

D_ADD MOVF  ACCALO,W ;ACCB+ACCA-->ACCB
      ADDWF  ACCBLO,1
      BTFSC  STATUS,C
      INCF   ACCBHI,1
      MOVF   ACCAHI,W
      ADDWF  ACCBHI,1
      RETLW  0

D_MPYS
      CALL  SETUP
MLOOP BCFSTATUS,C
      RRF   ACCDHI,1
      RRF   ACCDLO,1
      BTFSC STATUS,C
      CALL  D_ADD
      RRF   ACCBHI,1
      RRF   ACCBLO,1
      RRF   ACCCHI,1
      RRF   ACCCLO,1
      DECF  TEMP,1
      GOTO  MLOOP

SETUP MOVLW.16
      MOVWF TEMP
      MOVF  ACCBHI,W ;ACCB-->ACCD
      MOVWF ACCDHI
      MOVF  ACCBLO,W
      MOVWF ACCDLO
      CLRF  ACCBHI   ;?ACCB
      CLRF  ACCBLO
      RETLW 0

```

```

LOADAB  MOVLW00H
        MOVWFACCAHI
        MOVF  ACCA,0
        MOVWFACCALO
        MOVLW04H
        MOVWFACCBHI
        MOVLW0DDH ;ACCB=04DDH
        MOVWFACCBLO
        RETLW 0

MAP
CALL  DIVAB      ;显示光指针,求商以及余数,确定最后尾数
CALL  LOADAB
CALL  D_MPYS
MOVF  ACCDHI,W
MOVWFACCBHI
MOVF  ACCDLO,W
MOVWFACCBLO
goto BIN2BCDT

D_div  CALL  S_SIGN ;确定商的符号,并将负数取补
        CALL  SETUP1 ;初始化 TEMP, 将被除数移至 ACCD,
        ;(SETUP 子程序请参见 16×16 位定点数
        ;乘法子程序 SETUP)
        INCF  TEMP,1
        CLRF  ACCCHI ;清余数寄存器
        CLRF  ACCCLO
DLOOP  BCFSTATUS,C ;清进位位
        RLF  ACCDLO,1 ;被除数、余数左移 1 位
        RLF  ACCDHI,1
        RLF  ACCCLO,1
        RLF  ACCCHI,1
        MOVF  ACCAHI,0 ;ACCCHI-ACCAHI
        SUBWF ACCCHI,0
        BTFSS STATUS,Z ;ACCCHI=ACCAHI?
        GOTO  NOCHK
        MOVF  ACCALO,0 ;是, ACCCLO-ACCALO
        SUBWF ACCCLO,0
NOCHK  BTFSS STATUS,C ;ACCC>ACCA?
        GOTO  NOGO
        MOVF  ACCALO,0 ;是, 余数减除数
        SUBWF ACCCLO,1

```

```

    BTFSS STATUS,C
    DECF ACCCHI,1
    MOVF ACCAHI,0
    SUBWF ACCCHI,1
    BSF STATUS,C ;置进位位
NOGO RLFACCBLO,1 ;商左移 1 位
    RLF ACCBHI,1
    DECFSZ TEMP,1 ;循环完毕?
    GOTO DLOOP
    BTFSS SIGN,7 ;是, 确定商的符号
    GOTO DIVOVER ;为正, 除法结束, 跳转到结束行
    COMF ACCCLO,1 ;为负, 商和余数分别取补
    INCF ACCCLO,1
    BTFSC STATUS,Z
    DECF ACCCHI,1
    COMF ACCCHI,1
    CALL NEG_B ;见乘法程序中间 NEG_B
DIVOVER RETURN ;子程序返回
;*****除法运算确定结果符号子程序*****
S_SIGN MOVF ACCAHI,0 ;ACCAHI 异或 ACCBHI, 结果送 SIGN 单元
    XORWF ACCBHI,0
    MOVWF SIGN
    BTFSS ACCBHI,7 ;ACCB 为负?
    GOTO CHEK_A ;否, 检查 ACCA
    COMF ACCBLO,1 ;是, ACCB 取补
    INCF ACCBLO,1
    BTFSC STATUS,Z
    DECF ACCBHI,1
    COMF ACCBHI,1
CHEK_A BTFSC ACCAHI,7 ;ACCA 为负?
    CALL NEG_A ;ACCA 为负, 取补 (NEG_A 子程序请参见
;16×16 位定点数乘法子程序 NEG_A)
    RETURN ;ACCA 和 ACCB 均为负, 返回
SETUP1 MOVLW .15 ;初始化 TEMP 寄存器
    MOVWF TEMP
    MOVF ACCBHI,0 ;ACCB 送 ACCD
    MOVWF ACCDHI
    MOVF ACCBLO,0
    MOVWF ACCDLO
    CLRF ACCBHI ;清 ACCB
    CLRF ACCBLO
    RETURN ;子程序返回

```

```

;***** ACCA 取补子程序*****
NEG_A COMF ACCALO,1 ;ACCALO 取反加 1
      INCF  ACCALO,1
      BTFSC STATUS,Z      ;低 8 位有进位吗?
      DECF  ACCAHI,1      ;有, ACCAHI 减 1, 再取反
      COMF  ACCAHI,1      ;否则 ACCAHI 直接取反
      RETURN              ;子程序返回

NEG_B DECF  ACCBLO,1
      COMF  ACCBLO,1
      BTFSC STATUS,Z
      DECF  ACCBHI,1
      COMF  ACCBHI,1
OVER  RETURN              ;子程序返回
;.....
;显示余数的光指针
;.....
QIUYUSHU
      BCFSTATUS,C      ;如果采样数据小于 3 则直接显示 0
      MOVF  ACCCLO,0
      SUBLW 03H
      BTFSC STATUS,C
      RETURN
      CLRF  FIG
      MOVLW03H
SUBLOOP  BCFSTATUS,C
          SUBWF ACCCLO,1
          BTFSS STATUS,C
          GOTO OVER1
          INCF  FIG,1
          GOTO SUBLOOP
OVER1
      CALL  CONVERT
      CALL  PUTOUT
      RETURN
;.....
CONVERT
      MOVF  FIG,0
      ADDWF PCL,1
      NOP

```



```

TABLE RETLW    00H
      RETLW    01H
      RETLW    02H
      RETLW    03H
      RETLW    0ffH
;.....
PUTOUT  MOVWF   WTEMP      ;保存 W 寄存器
      BCF     STATUS,Z     ;判断商是否为零
      MOVF    TEMPAB,1    ;判断商是否为零
      BTFSC   STATUS,Z
      GOTO    JIANC        ;为零,则需要加上第一行的初始化程序
      GOTO    ZHIJIE       ;不为零,则直接接着商显示余数
JIANC   MOVLW   80H
      CALL    SendCmd
ZHIJIE  MOVF    WTEMP,0    ;恢复 W 寄存器
      CALL    putcLCD

      INCF    TEMPB,0     ;显示空格,以抵消不能返回的现象
      SUBLW   10H
      MOVWF   TEMPB
      BCF     STATUS,Z
      MOVF    TEMPB,1
      BTFSC   STATUS,Z
      RETURN

SPLOOP  MOVLW   20H      ;显示空格
      CALL    putcLCD
      DECFSZ  TEMPB,1
      GOTO    SPLOOP
      RETURN
;.....
;.....
XIANS
      BCF     STATUS,Z     ;判断商是否为零
      MOVF    ACCBLO,1
      BTFSC   STATUS,Z
      RETURN              ;为零则直接返回,不需要显示
      MOVLW   080H
      CALL    SendCmd     ;不为零则显示“商”个ASC码为“FF”的字符
SLOOP  MOVLW   B'11111111'
      CALL    putcLCD
      DECFSZ  ACCBLO,1
      GOTO    SLOOP      ;循环显示“商”个ASC码为“FF”的字符
      RETURN

```

```

;.....
DIVAB MOVF ACCALO,0          ;将 A/D 结果赋给 ACCBLO
    MOVWF ACCBLO
    MOVLW 0X00
    MOVWF ACCBHI
    MOVLW 0X10              ;除数 10H 送 ACCA
    MOVWF ACCALO
    MOVLW 0X00
    MOVWF ACCAHI
    CALL D_div              ;调用双字节除法子程序, 求商,ACCB/ACCA→ACCB,商在 ACCC 中
    MOVF ACCBLO,0          ;商保留在 ACCBLO 中
    MOVWF TEMPB            ;商复制到 TEMPB 中
    MOVWF TEMPAB           ;复制到 TEMPAB,以便判断商是否为零
    CALL XIANS              ;调用显示商的程序,即显示主光指针
    CALL QIUYUSHU          ;调用显示余数的程序,即显示尾数光指针
    RETURN

;* LCD 模块子程序 *
;*****
;InitLCD——初始化 LCD 模块 *
;*****
InitLCD
    bcf STATUS,RP0          ;Bank 0
    bcf STATUS,RP1
    clrf LCD_DATA           ;清 LCD 数据 & 控制位
    bsf STATUS,RP0          ;Bank 1
    movlw 0xc1              ;设置 RB1~RB5 为输出
    movwf LCD_DATA
    bcf TRISA,5
    bcf STATUS,RP0          ;
    movlw 0x00              ;
    movwf LCD_DATA
    bsf LCD_CNTL2, E
    bcf LCD_CNTL2, E
    movlw 0x08              ;
    movwf LCD_DATA
    bsf LCD_CNTL2, E
    bcf LCD_CNTL2, E
    call LongDelay
    call LongDelay
    movlw b'00101000'
    call SendCmd
    movlw DISP_ON          ;开显示
    call SendCmd

```

```

    movlw ENTRY_INC      ; 设置指针移动方式
    call  SendCmd
    movlw DD_RAM_ADDR    ; 设置起始地址
    call  SendCmd
InitLCDEnd      ;
    bcf   STATUS,RP0     ;
    movlw CLR_DISP      ;
    call  SendCmd
    call  LongDelay
    return
;*putcLCD—— 将字符数据送入 LCD *
;*          本程序将字符数据分成了高半字节和低半字节分别输入 *
;*          先送入的是高半字节 *
;*****
putcLCD
    bcf   STATUS,RP0
    movwf Byte          ; 存 WREG 到 Byte 寄存器
    swapf Byte,W        ; 写入高半字节
    movwf LCD_DATA1
    rlf   LCD_DATA1,1
    rlf   LCD_DATA1,1
    movlw 0x3c
    andwf LCD_DATA1,1
    movf  LCD_DATA1,w
    movwf LCD_DATA
    bsf   LCD_CNTRL1,RS ; 设置为送数据模式
    bsf   LCD_CNTRL2,E  ; 发送数据信号
    bcf   LCD_CNTRL2,E

    movf  Byte,W        ; 接着送低半字节
    movwf LCD_DATA1
    rlf   LCD_DATA1,1
    rlf   LCD_DATA1,1
    movlw 0x3c
    andwf LCD_DATA1,1
    movf  LCD_DATA1,w
    movwf LCD_DATA
    bsf   LCD_CNTRL1,RS ; 设置为送数据模式
    bsf   LCD_CNTRL2,E  ; 发送数据信号
    bcf   LCD_CNTRL2,E
    call  Delay
    return
;*****

```

```

;* SendCmd - 送命令到 LCD *
;*          本程序将命令数据分成了高半字节和低半字节分别输入 *
;*          先送入的是高半字节 *
;*****
SendCmd
    bcf     STATUS,RP0
    movwf  Byte          ; 存 WREG 到 Byte 寄存器
    swapf  Byte,W        ; 送入命令的高字节
    movwf  LCD_DATA1
    rlf   LCD_DATA1,1
    rlf   LCD_DATA1,1
    movlw  0x3c
    andwf  LCD_DATA1,1
    movf   LCD_DATA1,w
    movwf  LCD_DATA
    bcf   LCD_CNTRL1,RS ; 设置为送命令模式
    bsf   LCD_CNTRL2,E  ; 发送数据信号
    bcf   LCD_CNTRL2,E
    movf  Byte,W        ; 接着送命令的低字节

    movwf  LCD_DATA1
    rlf   LCD_DATA1,1
    rlf   LCD_DATA1,1
    movlw  0x3c
    andwf  LCD_DATA1,1
    movf   LCD_DATA1,w
    movwf  LCD_DATA
    bcf   LCD_CNTRL1,RS ; 设置为送命令模式
    bsf   LCD_CNTRL2,E  ; 发送数据信号
    bcf   LCD_CNTRL2,E
    call  Delay
    return
;* Delay - 延时子程序 *
;*****
Delay          ; call 指令占用 2 个指令周期
    bcf  STATUS,RP0
    clrf Count          ; 清 Count 占用 1 个指令周期
Dloop
    decfsz Count,f      ; 这两行指令将延时
    goto Dloop          ; (256 * 3) - 1 个指令周期
    return              ; return 占用 2 个指令周期
;*****

```

```

;* LongDelay - 长延时子程序 *
;*****
LongDelay
    bcf    STATUS,RP0
    clrf   Count
    clrf   Count1
    movlw  0x03
    movwf  Count2
LDloop
    decfsz Count,f
    goto  LDloop
    decfsz Count1,f
    goto  LDloop
    decfsz Count2,f
    goto  LDloop
    return
;-----
;Bin2BCDT - 将 Temp 寄存器内的二进制数转 ASC 码 *
BIN2BCDT
    movf  ACCBLO,W
    movwf s0
    movF  ACCCHI,W
    movwf s1
    movF  ACCCLO,w
    movwf s2
    call  B2_BCD
    call  Over0
    goto  LCDstart

B2_BCD
    bcf  status,carry
    movlw .24
    movwf count3
    clrf r0
    clrf r1
    clrf r2
loop16
    rlf  s2,1
    rlf  s1,1
    rlf  s0,1
    rlf  r2,1
    rlf  r1,1
    rlf  r0,1

```

```

    decfsz    count3,1
    goto adjdec
    retlw    0
adjdec
    movlw    r2
    movwf    fsr
    call adjbcd
    movlw    r1
    movwf    fsr
    call adjbcd
    movlw    r0
    movwf    fsr
    call adjbcd
    goto loop16
adjbcd
    movlw    3
    addwf    0,w
    movwf    temp
    btfsc temp,3
    movwf    0
    movlw    30h
    addwf    0,w
    movwf    temp
    btfsc temp,7
    movwf    0
    retlw    0
Over0
    ;将 BCD 码转成 ASC 码,
    ;结果存在 rohi,r0lo,r1hi,r1lo,r2hi,r2lo (r0 为高位)
    movf    r0,w
    movwf    tempr0
    andlw    0fh
    movwf    r0lo
    swapf    tempr0,w
    andlw    0fh
    movwf    r0hi
;.....
    movf    r1,w
    movwf    tempr0
    andlw    0fh
    movwf    r1lo
    swapf    tempr0,w
    andlw    0fh
    movwf    r1hi

```

```

;.....
movf    r2,w
movwf   tempr0
andlw   0fh
movwf   r2lo
swapf   tempr0,w
andlw   0fh
movwf   r2hi
;.....
movlw   30h
addwf   r0hi,1
addwf   r0lo,1
addwf   r1hi,1
addwf   r1lo,1
addwf   r2hi,1
addwf   r2lo,1
retlw   0
;.....乘法.....
;A/D 结果放到 ACCALO 中
;结果存在 ACCBHI,ACCBLO,ACCCHI,ACCCLO 中
D_ADDs
MOVWF  ACCALO,W ;ACCB+ACCA-->ACCB
ADDWF  ACCBLO,1
BTFSZ  STATUS,C
INCF   ACCBHI,1
MOVWF  ACCAHI,W
ADDWF  ACCBHI,1
RETLW  0
D_MPYs
CALL   SETUPs
MLOOPs
BCF    STATUS,C
RRF    ACCDHI,1
RRF    ACCDLO,1
BTFSZ  STATUS,C
CALL   D_ADDs
RRF    ACCBHI,1
RRF    ACCBLO,1
RRF    ACCCHI,1
RRF    ACCCLO,1
DECFSZ TEMP,1
GOTO   MLOOPs
RETLW  0

```

## SETUPs

```

    MOVLW.16    ;
    MOVWF TEMP
    MOVF  ACCBHI,W ;ACCB-->ACCD
    MOVWF ACCDHI
    MOVF  ACCBLO,W
    MOVWF ACCDLO
    CLRF  ACCBHI   ;ACCB
    CLRF  ACCBLO
    RETLW 0

```

## LOADABs

```

    MOVLW 04H
    MOVWF ACCBHI
    MOVLW 0ddH ;ACCB=4ddH
    MOVWF ACCBLO
    RETLW 0

```

## MAPs

```

    CALL  D_MPYs
    RETLW 0

```

```

F_SWAPs  MOVF      ACCAHI,W
          MOVWF    TEMP
          MOVF     ACCBHI,W ;ACCAHI<-->ACCBHI
          MOVWF   ACCAHI
          MOVF     TEMP,W
          MOVWF   ACCBHI
          MOVF     ACCALO,W
          MOVWF   TEMP
          MOVF     ACCBLO,W ;ACCALO<-->ACCBLO
          MOVWF   ACCALO
          MOVF     TEMP,W
          MOVWF   ACCBLO
          MOVF     ACCDHI,W
          MOVWF   TEMP
          MOVF     ACCCHI,W ;ACCDHI<-->ACCCHI
          MOVWF   ACCDHI
          MOVF     TEMP,W
          MOVWF   ACCCHI
          MOVF     ACCDLO,W
          MOVWF   TEMP
          MOVF     ACCCLO,W ;ACCDLO<-->ACCCLO
          MOVWF   ACCDLO
          MOVF     TEMP,W

```



```

                MOVWF    ACCCLO
                RETLW 0
;,,,,,,,,,,,,,32 位的二进制到 BCD 码的转换
;Bin2BCDT - 将 Temp 寄存器内的二进制数转 ASC 码
BIN2BCDT32s
    movf    ACCBHI,W
    movwf   s0
    movF    ACCBLO,W
    movwf   s1
    movF    ACCCHI,w
    movwf   s2
    movF    ACCCLO,w
    movwf   s3
    call    B2_BCD32s
    call    Over032s
    RETLW 0
B2_BCD32s
    bcf    status,carry
    movlw  .32
    movwf  count3
    clrf  r0
    clrf  r1
    clrf  r2
    clrf  r3
    clrf  r4
loop1632s
    rlf  s3,1
    rlf  s2,1
    rlf  s1,1
    rlf  s0,1
    rlf  r4,1
    rlf  r3,1
    rlf  r2,1
    rlf  r1,1
    rlf  r0,1
    decfsz count3,1
    goto  adjdec32s
    retlw 0
adjdec32s
    movlw  r4
    movwf  fsr
    call  adjbcd32s
    movlw  r3

```

```

movwf  fsr
call   adjbcd32s
movlw  r2
movwf  fsr
call   adjbcd32s
movlw  r1
movwf  fsr
call   adjbcd32s
movlw  r0
movwf  fsr
call   adjbcd32s
goto   loop1632s
adjbcd32s
movlw  3
addwf  0,w
movwf  temp
btfsc temp,3
movwf  0
movlw  30h
addwf  0,w
movwf  temp
btfsc temp,7
movwf  0
retlw  0
Over032s      ;将 BCD 码转成 ASC 码,
               ;结果存在 rohi,r0lo,r1hi,r1lo,r2hi,r2lo (r0 为高位)
movf    r0,w
movwf   tempr0
andlw   0fh
movwf   r0lo
swapf   tempr0,w
andlw   0fh
movwf   r0hi
;.....
movf    r1,w
movwf   tempr0
andlw   0fh
movwf   r1lo
swapf   tempr0,w
andlw   0fh
movwf   r1hi
;.....
movf    r2,w

```

```

movwf tempr0
andlw 0fh
movwf r2lo
swapf tempr0,w
andlw 0fh
movwf r2hi
;.....
movf r3,w
movwf tempr0
andlw 0fh
movwf r3lo
swapf tempr0,w
andlw 0fh
movwf r3hi
;.....
movf r4,w
movwf tempr0
andlw 0fh
movwf r4lo
swapf tempr0,w
andlw 0fh
movwf r4hi
;.....
movlw 30h
addwf r0hi,1
addwf r0lo,1
addwf r1hi,1
addwf r1lo,1
addwf r2hi,1
addwf r2lo,1
addwf r3hi,1
addwf r3lo,1
addwf r4hi,1
addwf r4lo,1
RETLW 0
;.....;开平方;.....;
;32位减法 (ACCB-ACCA-->ACCB)
DD_SUBs          CALL          NEG_As          ;ACCA??
;32位加法 (ACCB+ACCA-->ACCB)
DD_ADDs
                BCF          STATUS,C
                MOVF         ACCDLO,W
                ADDWF        ACCCLO,1

```

	BTFSC	STATUS,C
	GOTO	ADD1s
	GOTO	ADDA <sub>s</sub>
ADD1s	BCF	STATUS,Z
	INCF	ACCCHI,1
	BTFSC	STATUS,Z
	GOTO	ADD2s
	GOTO	ADDA <sub>s</sub>
ADD2s	BCF	STATUS,Z
	INCF	ACCBLO,1
	BTFSC	STATUS,Z
	GOTO	ADD3s
	GOTO	ADDA <sub>s</sub>
ADD3s	BCF	STATUS,Z
	INCF	ACCBHI,1
	BTFSC	STATUS,Z
	GOTO	ADD4s
	GOTO	ADDA <sub>s</sub>
ADD4s	BSF	STATUS,C
	GOTO	ADD0s
ADDA <sub>s</sub>	BCF	STATUS,C
	MOVF	ACCDHI,W
	ADDWF	ACCCHI,1
	BTFSC	STATUS,C
	GOTO	ADD5s
	GOTO	ADDB <sub>s</sub>
ADD5s	BCF	STATUS,Z
	INCF	ACCBLO,1
	BTFSC	STATUS,Z
	GOTO	ADD6s
	GOTO	ADDB <sub>s</sub>
ADD6s	BCF	STATUS,Z
	INCF	ACCBHI,1
	BTFSC	STATUS,Z
	GOTO	ADD4s
	GOTO	ADDB <sub>s</sub>
ADDB <sub>s</sub>	BCF	STATUS,C
	MOVF	ACCALO,W
	ADDWF	ACCBLO,1
	BTFSC	STATUS,C
	GOTO	ADD7s
	GOTO	ADDC <sub>s</sub>

ADD7s	BCF	STATUS,Z	
	INCF	ACCBHI,1	
	BTFSC	STATUS,Z	
	GOTO	ADD4s	
	GOTO	ADDCs	
ADDCs	BCF	STATUS,C	
	MOVF	ACCAHI,W	
	ADDWF	ACCBHI,1	
	BTFSC	STATUS,C	
	GOTO	ADD4s	
	GOTO	ADD0s	
ADD0s	RETLW	0	
NEG_As			
	COMF	ACCDLO,1	;ACCA??
	COMF	ACCDHI,1	
	COMF	ACCALO,1	
	COMF	ACCAHI,1	
	BCF	STATUS,Z	
	INCF	ACCDLO,1	
	BTFSC	STATUS,Z	
	GOTO	NEG1s	
	GOTO	NEG0s	
NEG1s	BCF	STATUS,Z	
	INCF	ACCDHI,1	
	BTFSC	STATUS,Z	
	GOTO	NEG2s	
	GOTO	NEG0s	
NEG2s	BCF	STATUS,Z	
	INCF	ACCALO,1	
	BTFSC	STATUS,Z	
	GOTO	NEG3s	
	GOTO	NEG0s	
NEG3s	INCF	ACCAHI,1	
	GOTO	NEG0s	
NEG0s	RETLW	0	
	:.....		
PINGFANG			
	clrf	ACCAHI	
	CLRF	ACCALO	
	CLRF	ACCDHI	
	CLRF	ACCDLO	
	clrf	DACCAHI	
	CLRF	DACCALO	

	CLRF	DACCDHI	
	CLRF	DACCDLO	
	INCF	DACCDLO,1	
STARTs			
	BCF	STATUS,C	
	MOVF	DACCDLO,W	
	MOVWF	ACCDLO	
	MOVF	DACCDHI,W	
	MOVWF	ACCDHI	
	MOVF	DACCALO,W	
	MOVWF	ACCALO	
	MOVF	DACCAHI,W	
	MOVWF	ACCAHI	
	RLF	ACCDLO,1	
	RLF	ACCDHI,1	
	RLF	ACCALO,1	
	RLF	ACCAHI,1	
	BCF	STATUS,CARRY	
	MOVLW	01H	
	SUBWF	ACCDLO,1	
	BTFS	STATUS,CARRY	
	GOTO	MNs	
	DECF	ACCDHI,1	
	BCF	STATUS,CARRY	
MNs	CALL	DD_SUBs	
	BTFS	STATUS,CARRY	;为零跳, 不够减
	GOTO	MN1s	;够减, ACCA 加一
	GOTO	MN2s	;不够减, DACCA 减一返回
MN1s	INCF	DACCDLO,1	
	BTFS	STATUS,Z	
	INCF	DACCDHI,1	
	GOTO	STARTs	
MN2s			
	BCF	STATUS,Z	
	MOVF	DACCDLO,1	;判断 DACCALO 是否为零
	BTFS	STATUS,Z	
	GOTO	MN3s	;DACCALO 为零
	GOTO	MN4s	
MN3s	DECF	DACCDLO,1	
	DECF	DACCDHI,1	
	GOTO	NH5s	
MN4s	DECF	DACCDLO,1	
NH5s	RETLW	0	

;.....16 位加法

```

F_ADDs    MOVF      ACCALO,W ;ACCB+ACCA-->ACCB
          ADDWF    ACCBLO,1
          BTFSC    STATUS,C
          INCF     ACCBHI,1
          MOVF     ACCAHI,W
          ADDWF    ACCBHI,1
          RETLW   0

```

;.....液晶显示数据.....

;.....中断子程序

;,,,实现 AD 结果实行累加求和, 求 XI 平方的以及平方的总和

```

STAR1s    CLRF     ACCALO
          CLRF     ACCAHI
          CLRF     ACCBLO
          CLRF     ACCBHI
          CLRF     ACCCLO
          CLRF     ACCCHI
          CLRF     ACCDLO
          CLRF     ACCDHI
          CLRF     ACCEHI
          CLRF     ACCELO
          CLRF     ACCFHI
          CLRF     ACCFLO
          CLRF     BXHI
          CLRF     BXLO
          MOVLW   0ffH
          MOVWF   DISCOUNT1
          bcf     status,rp0
          clrf   portd
          movlw  b'01000001'
          movwf  adcon0
          bsf     status,rp0
          clrf   trisd
          movlw  b'00001110'
          movwf  adcon1
          bcf     status,rp0
INTSTARTs
          bsf     adcon0,go
INTwait
          btfs   pir1,adif
          goto   INTwait
          movf   adresh,w
          movwf  ACCALO ;将 A/D 结果存于 ACCA,并同时

```

```

MOVWF    A_TEMPLO
MOVLW    00H
MOVWF    ACCAHI
MOVWF    A_TEMPHI
MOVF     BXHI,W
MOVWF    ACCBHI
MOVF     BXLO,W
MOVWF    ACCBLO
CALL     F_ADDs
MOVF     ACCBHI,W ;存储累加结果到 BX, 腾出 ACCB
MOVWF    BXHI
MOVF     ACCBLO,W
MOVWF    BXLO ;累加结果始终在 BX 中
MOVF     A_TEMPHI,W ;恢复 ACCA, 并使
MOVWF    ACCAHI ;ACCA=ACCB 以进行平方运算
MOVWF    ACCBHI ;同时放到 ACCB 中, 以进行平方运算
MOVF     A_TEMPLO,W
MOVWF    ACCALO
MOVWF    ACCBLO
CALL     MAPs ;执行 X*X
;.....求 Xi 平方的累加.....
;XI 平方的结果放在 ACCB,ACCC 中, 累加时先将平方的结果放到 ACCA 和 ACCD 中, 用于累加
;ACCE 和 ACCF 为累加缓冲区, 用于存放累加结果
MOVF     ACCBHI,W ;将 x2(B,C)→A,D 以求 x2 的累加
MOVWF    ACCAHI
MOVF     ACCBLO,W
MOVWF    ACCALO
MOVF     ACCCHI,W ;将 x2(B,C)→A,D 以求 x2 的累加
MOVWF    ACCDHI
MOVF     ACCCLO,W
MOVWF    ACCDLO
MOVF     ACCEHI,W ;将 E, F→B,C
MOVWF    ACCBHI
MOVF     ACCELO,W
MOVWF    ACCBLO
MOVF     ACCFHI,W
MOVWF    ACCCHI
MOVF     ACCFLO,W
MOVWF    ACCCLO
CALL     DD_ADDs ;执行 xi 平方的累加
MOVF     ACCBHI,W ;将结果高位放到 E 当中
MOVWF    ACCEHI ;平方累加结果在 ACCE 中

```



```

MOVF      ACCBLO,W
MOVWF     ACCELO
MOVF      ACCCHI,W ;将结果低位放到 F 当中
MOVWF     ACCFHI   ;平方累加结果在 ACCF 中
MOVF      ACCCLO,W
MOVWF     ACCFLO
DECFSZ    DISCOUNT1,1
GOTO      INTSTARTs
GOTO      DISPLAYs
;.....求两个平均数: Xi 平方和平均 以及 Xi 和的平均(共 8 个数据, 所以为 2 的 3
次方).....
DISPLAYs
        MOV LW      .8
        MOV WF     DISCOUNT      ;Xi 平方和平均
NEXTs   BCF        STATUS,C
        RRF        ACCEHI,1
        RRF        ACCELO,1
        RRF        ACCFHI,1
        RRF        ACCFLO,1
        DECFSZ    DISCOUNT,1
        GOTO      NEXTs

        MOV LW      .8
        MOV WF     DISCOUNT
NEXT1s  BCF        STATUS,C ;xi 总和的平均
        RRF        BXHI,1
        RRF        BXLO,1 ;循环结束后 BX 中存放的是 XI 的平均值
        DECFSZ    DISCOUNT,1
        GOTO      NEXT1s
        MOVF      ACCEHI,W ;XI 平方的平均
        MOVWF     BE_TEMPHI
        MOVF      ACCELO,W
        MOVWF     BE_TEMPLO
        MOVF      ACCFHI,W
        MOVWF     BF_TEMPHI
        MOVF      ACCFLO,W
        MOVWF     BF_TEMPLO
        MOVF      BXHI,W
        MOVWF     ACCAHI ;用于 x 平均的平方计算
        MOVWF     ACCBHI
        MOVWF     PINGJUNHI ;输出显示 x 的平均值
        MOVF      BXLO,W
        MOVWF     ACCALO

```

```

MOVWF ACCBLO
MOVWF PINGJUNLO ;输出显示 x 的平均值
CALL MAPs ;求 x 平均的平方 (结果在 ACCB,ACCC 中)
; 将 放到 ACCA 中

MOVF BE_TEMPHI,W
MOVWF ACCAHI
MOVF BE_TEMPLO,W
MOVWF ACCALO
MOVF BF_TEMPHI,W
MOVWF ACCDHI
MOVF BF_TEMPLO,W
MOVWF ACCDLO
CALL F_SWAPs ;将 ACCA,ACCD 和 ACCB,ACCC 交换
CALL DD_SUBs ;以进行减法运算
;减法运算结果存放在 ACCB,ACCC 中

CALL PINGFANG ;调用 32 位开平方子程序求均方差
MOVF DACCDHI,W ;将均方差转存到 ACCC 中
MOVWF ACCAHI
MOVF DACCDLO,W
MOVWF ACCALO
CALL LOADABs
CALL MAPs
CALL BIN2BCDT32s
movlw CLR_DISP
call SendCmd
CALL LongDelay
movlw 80h ;初始化第一行
call SendCmd
CALL LongDelay
movlw B'11100101'
call putcLCD
movlw B'0111010'
call putcLCD
CALL LCDstart1 ;在液晶上显示均方差
GOTO SHOWPs

LCDstart1
movf r2hi,w
call putcLCD
movf r2lo,w
call putcLCD
movf r3hi,w
call putcLCD
movlw b'00101110'

```



```

        movlw      B'0111010'
        call      putcLCD
        CALL      LCDstart2
        RETLW     0

LCDstart2
    movf      r2hi,w
    call      putcLCD
    movf      r2lo,w
    call      putcLCD
    movf      r3hi,w
    call      putcLCD
    movlw     b'00101110'
    call      putcLCD
    movf      r3lo,w
    call      putcLCD
    movlw     ''
    call      putcLCD
    movlw     ''
    call      putcLCD
    movlw     ''
    call      putcLCD
    movlw     ''
    call      putcLCD
    movlw     ''
    call      putcLCD
    movlw     ''
    call      putcLCD
    movlw     ''
    call      putcLCD
    movlw     ''
    call      putcLCD
    movlw     ''
    call      putcLCD
    movlw     ''
    call      putcLCD
    RETLW     0

.....*****
*****
SERV
    MOVWFW_TEMP      ;保护现场
    SWAPF STATUS,W
    CLRF STATUS
    MOVWFSTATUS_TEMP
;*****
    CALL STAR1s
    CALL LongDelay ;*****保持键*****

```

```

CALL LongDelay
CALL LongDelay
CALL LongDelay
CALL LongDelay
CALL LongDelay
CALL LongDelay
CALL LongDelay
CALL LongDelay
CALL LongDelay
movlw CLR_DISP
call SendCmd
bcf intcon,INTF
goto retfie0
retfie0
    swapf STATUS_TEMP,w
    movwf status
    swapf W_TEMP,F
    swapf W_TEMP,W
    RETFIE
LCDstart
    movlw 0C0h ;初始化第 2 行
    call SendCmd
    movf r0hi,w
    call putcLCD
    movf r0lo,w
    call putcLCD
    movf r1hi,w
    call putcLCD
    movlw b'00101110'
    call putcLCD
    movf r1lo,w
    call putcLCD
    movlw 'C'
    call putcLCD
    movlw 'N'
    call putcLCD
;*****保持键*****
HOLD
; BTFSS PORTD,2
; GOTO HOLD
goto ADstart
End

```

## 致 谢

本课题是在导师陈庆官教授的悉心指导下完成的。整个课题从题目的选择、设计方案的确定到具体试验的进行以及论文的撰写，无不倾注了导师大量的心血；在此过程中，不仅学到了许多书本中无法学到的知识，而且导师那种严谨求实，一丝不苟、精益求精的治学态度；大胆假设、小心求证、勇于创新的学术作风；以及平易近人、谦虚好学、胸襟宽广的高尚情操，更让学生受益匪浅。

此外，在本课题中，得到了本研究室的博士研究生孙鸿文的大力帮助，同时也得到了硕士研究生戴新兰、林枫、陈艳华、罗斯杰等人的大力支持。

在此，谨向为本课题付出了辛勤劳动的导师和同学致以最诚挚的谢意！