

自动加料机控制系统

福星电子网

提供单片机学习板，开发板，最小系统板；超声波测距实验应用板，各类器件仪表，详情请访问网站 <http://www.fxdzw.com>

目 录

第一章	绪 论	1
1.1	题目来源及课题意义	1
1.2	自动加料机控制系统的工作原理及技术要求	1
1.3	系统的主要技术参数:	1
第二章	方案论证	2
2.1	单片机的选择	2
2.2	物位传感器的选择	3
2.3	存储器扩展电路的选择	5
2.4	LED 显示电路选择	7
2.5	键盘输入电路	8
第三章	自动加料机主电路	10
3.1	系统结构原理图	10
3.2	主机电路核心器件介绍	10
3.3	显示电路	18
3.4	继电器控制电路	22
3.5	键盘及显示电路	23
3.6	外部存储器扩展电路	27
3.7	看门狗 MAX813L 电路	30
3.8	料位开关	32
第四章	系统的抗干扰及可靠性	35
第五章	软件设计	36
第六章	结论语	38
	参考文献:	39
	附录 A: 系统框图	40
	附录 B: 系统硬件框图	40
	程序清单:	42

第一章 绪论

1.1 题目来源及课题意义

在现代科学技术的许多领域中，自动控制技术起这愈来愈重要的作用，并且，随着生产和科学技术的发展，自动化水平也越来越高。自动控制利用控制装置使被控对象的某个参数自动的按照预定的规律运行。本设计的自动加料机控制系统就是采用自动控制技术来实现功能的，这样就大大提高了工作的效率，整个过程又快又稳。

1.2 自动加料机控制系统的工作原理及技术要求

本设计的由单片机控制的自动加料系统是与料斗式干燥机配套的加料系统。根据加料工艺要求，其工作原理是：先将真空管关闭，启动电机，用低真空气流将塑料树脂粒子送入真空管，电机停转，再将粒子排入料斗，如此循环。

在设计控制系统中，可用一个电机控制两个加料生产线，由方向阀切换。两个生产线既可单独运行，也可同时运行。假如两者同时运行，当一生产线输送结束后，判断到另一个生产线排料已经结束，那么，电机不停转而方向阀换向，从而为另一个生产线送料。这样可以发挥控制系统和电机的效率，从而实现供料自动化。

控制系统的控制器有单片机 89C51 和扩展电路组成，单片机控制继电器，继电器控制交流接触器，又由接触器控制电机等执行机构的运动。本控制系统可以根据送料工艺的需要，设置两条生产线的输送、排料、满料、空料等参数值，也可装载系统前次工艺参数值。

1.3 系统的主要技术参数：

- (1) 用一台电机控制两条生产线
- (2) 要能检测到满料状态，并显示出输送、排料、满料时间
- (3) 时间误差：0.1 秒
- (4) 具有抗干扰能力

第二章 方案论证

2.1 单片机的选择

20 世纪 80 年代以来，单片机的发展非常迅速，就通用单片机而言，世界上一些著名的计算机厂家已投放市场的产品就有 50 多个系列，数百个品种。目前世界上较为著名的 8 位单片机的生产厂家和主要机型如下：

美国 Intel 公司：MCS—51 系列及其增强型系列

美国 Motorola 公司：6801 系列和 6805 系列

美国 Atmel 公司：89C51 等单片机

美国 Zilog 公司：Z8 系列及 SUPER8

美国 Fairchild 公司：F8 系列和 3870 系列

美国 Rockwell 公司：6500/1 系列

美国 TI（德克萨司仪器仪表）公司：TMS7000 系列

NS（美国国家半导体）公司：NS8070 系列 等等。

尽管单片机的品种很多，但是在我国使用最多的还是 Intel 公司的 MCS—51 系列单片机和美国 Atmel 公司的 89C51 单片机

MCS—51 系列单片机包括三个基本型 8031、8051、8751

8031 内部包括一个 8 位 CPU、128 个字节 RAM，21 个特殊功能寄存器（SFR）、4 个 8 位并行 I/O 口、1 个全双工串行口、2 个 16 位定时器/计数器，但片内无程序存储器，需外扩 EPROM 芯片。比较麻烦，不予采用

8051 是在 8031 的基础上，片内集成有 4K ROM，作为程序存储器，是一个程序不超过 4K 字节的小系统。ROM 内的程序是公司制作芯片时，代为用户烧制的，出厂的 8051 都是含有特殊用途的单片机。所以 8051 适合与应用在程序已定，且批量大的单片机产品中。也不予采用。

8751 是在 8031 基础上，增加了 4K 字节的 EPROM，它构成了一个程序小于 4KB 的小系统。用户可以将程序固化在 EPROM 中，可以反复修改程序。但其价格相对 8031 较贵。8031 外扩一片 4KB EPROM 的就相当与 8751，它的最大优点是价格低。随着大规模集成电路技术的不断发展，能装入片内的外围接口电路也可以是大规模的。也不予采用。

AT89C51 是美国 ATMEL 公司生产的低电压，高性能 CMOS 8 位单片机，片内含 4K bytes 的可反复擦写的只读程序存储器（PEROM）和 128bytes 的随机存取数据存储器

(ROM), 器件采用 ATME1 公司的高密度、非易失性存储技术生产, 兼容标准 MCS-51 指令系统, 片内置通用 8 位中央处理器 (CPU) 和 Flash 存储单元。功能强大 AT89C51 单片机可提供许多高性价比的应用场合, 可灵活应用于各种控制领域。此设计就采用 AT89C51。

2.2 物位传感器的选择

物位是指贮存容器或工业生产设备里的液体、粉粒状固体、气体之间的分界面位置, 也可以是互不相溶的两种液体间由于密度不等而形成的界面位置。根据具体用途分为液位、料位、界位传感器或变送器。物位不仅是物料耗量或产量计量的参数, 也是保证连续生产和设备安全的重要参数。特别是在现代工业中, 生产规模大, 速度高, 且常有高温、高压、强腐蚀性或易燃易爆物料, 对于物位的监视和自动控制更是至关重要。

物位测量可用于计算物料储量。对于粉粒体, 必须考虑到颗粒间有空隙, 应区分密度和容重。密度是指不含空隙的物料每单位体积的质量, 即通常的质量密度 ρ , 如果乘以重力加速度 g , 就成为重力密度 ρ_v , 简称为重度。容重是包含空隙在内的每单位体积的重量 ρ_v 。也就是视在重度或宏观重度, 它总要比颗粒物质本身的重度小, 其差额决定于空隙率。而空隙率又取决与许多因素。例如颗粒形状、尺寸的一致程度、是否受外力压实、是否经受过振动、有无黏结性等, 所以粉粒体物料的体积储量和质量储量之间不易精确换算, 这是需要注意的。

2.2.1 电容式物位传感器

利用物料介电常数恒定时极间电容正比与物位的原理, 可构成电容式物位传感器。

根据电机的结构可将容式物位传感器分为三中: (1) 适用与导电容器中的绝缘性物料, 且容器为立式圆筒形, 器壁为一极, 沿轴线插入金属棒为另一极, 其间构成的电容 C 与物位成比例。也可悬挂带重锤的软导线作为电机。(2) 适用与非金属容器, 或虽为金属容器但非立式圆筒形, 物料为绝缘性的。这时在棒状电极周围用绝缘支架套装金属筒, 筒上下开口, 或整体上均匀分布多个孔, 使内外物位相同。中央圆棒和与之同轴的套筒构成两个电极, 其间电容和容器形状无关, 只取决于物位。所以这种电极只用于液位, 粉粒体容易滞留在极间。(3) 用于导电性物料, 起外形和 (1) 一样, 但中央圆棒电极上包有绝缘材料, 电容是由绝缘材料的介电常数和物位决定的, 与物料的介电常数无关, 导电物料使筒壁与中央电极间的距离缩短为绝缘层的厚度, 物位升降相当于电极面积改变。

电容式物位传感器无可动部件, 与物料密度无关, 但应注意物料中含水分时将

测量结果影响很大,并且要求物料的介电常数与空气介电常数差别大,需用高频电路。所以不予采用。

2.2.2 阻力式料位传感器

阻力式料位传感器是指物料对机械运动所呈现的阻挡力。粉末颗粒状物料比液态物质流动性差,对运动物体有明显的阻力,利用这一特点可构成各种料位传感器。

(1) 重锤探索法:在容器顶部安装由脉冲分配器控制的步进电机,此电机正转时缓缓释放悬有重锤的钢索。重锤下降到与料面接触后,钢索受到的合力突然减小,促使力传感器发出脉冲。此脉冲改变门电路的状态,使步进电机改变转向重锤提升,同时开始脉冲计数。待重锤升至顶部触及行程开关,步进电机停止转动,同时计数器也停止计数并显示料位(料位值即容器全高减去重锤行程之差)。显示值一直保持到下次探索后刷新为另一值。开始探索的触发信号可由定时电路周期性地供给,也可以人为地启动。不进行探索时,重锤保持在容器顶部,以免物料将重锤淹埋。万一重锤被物位埋没,排放物料时产生的强大拉力就可能拉断钢索报警措施及出料过滤栅。

但这种方法运用了逻辑电路和数字技术,可连续测量料位值并输出数字量,是数字传感器,但其采样是周期性的,对时间而言不连续,此设计不予采用。

(2) 旋桨或推板法:这是一种位式传感器,或称料位开关。在容器壁的某一高度处装小功率电动机,其轴伸入容器内,末端带有桨状叶片。叶片不接触物料时,自由旋转的空载状态下电动机的电流很小,一旦料位上升到与叶片接触,转动阻力增加,甚至成堵转状态,电流显著加大。根据电流的大小使继电器的接点动作,发出料位报警或位式控制信号。如电机轴经过曲柄连杆机构变为往复运动,则可带动活塞或平板在容器中做推拉动作,即成推板法。旋桨法或推板法不一定是靠电机电流的大小时继电器接点动作,也可以利用离合器或连杆上的传动机构,在叶片或推板负载增大时改变电接点的通断状态。所用电动机应能在长时间堵转状态下,或离合器打滑状态下,不致过热而损坏。

这类原理构成的料位开关,只能安装在容器壁上,安装高度取决于动作所对应的料位值。应用不那么广泛,所以次设计也不予采用。

(3) 音叉法:根据物料对振动中的音叉有无阻力探知料位是否到达或超过某高度,并发出通断信号,这种原理不需要大幅度的机械运动,驱动功率小,机械结构简单、灵敏而可靠。

音叉由弹性良好的金属制成,本身具有确定的固有频率,如外加交变力的频率与其固有频率一致,则叉体处于共振状态。由于周围空气对振动的阻尼微弱,金属内部的能量损耗又很少,所以只需微小的驱动功率就能维持较强的振动。当粉粒体物料触及叉体之后,能量消耗在物料颗粒间的摩擦上,迫使振幅急剧衰减,音叉停振。

为了给音叉提供交变的驱动力,利用放大电路对压电元件施加交变电场,靠逆压电效应产生机械力作用在叉体上。用另外一组压电元件的正压电效应检测振动,它把

振动力 为微弱的交变电信号。再由电子放大器和移相电路，把检振元件的信号放大。经过移相，施加到驱动元件上去，构成闭环振荡器。在这个闭环中，既有机械能也有电能，叉体是其中的一个环节，倘若受到物料阻尼难以振动，正反馈的幅值和相位都将明显的改变，破坏了振荡条件，就会停振。只要在放大电路的输出端接以适当的器件，不难得到开关信号。

为了保护压电元件免受物料损坏和粉尘污染，将驱动和检振元件装在叉体内部，经过金属膜片传递振动。如果在容器的上下方都装叉体，可以实现自动进料或自动出料的逻辑控制，或者把料位超限信号远传到控制室。在控制室里的控制电路判断料位是否超限，并按要求使被控的进出料设备启停。

并且叉体的制造和装配良好时，音叉也可用于液体测量和控制。在测量时不需要大幅度的机械运动，驱动功率小，机械结构简单、灵敏而可靠。此设计选择音叉法阻力式料位传感器。

2.3 存储器扩展电路的选择

2.3.1 24C01 扩展:

串行总线上的各单片机或集成电路模块，通过一条数据线(SDA)和一条时钟线(SCL)，按照通信规约进行寻址和信息传输。每个集成电路模块都有唯一伪地址，既可以是主控机(能控制总线，并能完成一次传输过程的初始化和产生时钟信号及传输终止信号的器件)或被控机(被主控器寻址的器件)，可以是发送器(在总线 1: 发送信息的器件)或接收器(从总线上接收信息的器件) I²C 总线上的器件，根据它的不同工作状态，可分为主控发送器、主控接收器、被控发送器、被控接收器。当多个主控器同时企图控制总线而不丢失信，这叫多主竞争。这时就要进行仲裁，仲裁就是针对这种情况进行裁决的过程。只允许其下一个主控器继续占用总线，其它退出丰换器状态。仲裁过程中还要保证总线的信息不丢失。多主竞争时必须对所有参与竞争的主控器的时钟信号进行同步处理。信息传输时，SCL 为高电平期间，SDA 上的信息必须保持稳定不变，只有 SCL 为低电平期间，SDA 上的信息才允许变化。同时 SDA 上信息每一位都和 SCL 的时钟脉冲相对应。SCL 没有时钟信号，SDA 信息将停止传输处于等待状态。这因为线“与”逻辑，使 SCL 在低电平时钳住总线。实现线“与”逻辑功能各 I²C 总线接口的输出端必须是漏极开路或集电极开路结构。SCL 保持高电平期间，SDA 由高电平向低电平变化这种状态定义为起始信号。SCL 保持高电平期间，SDA 由低电平向高电平变化，这种状态定义为终止信号。SDA 传输的每个字节必须 8 位(最高有效位首先传送)，每个传送字节必须跟随一位应答位。与应答信号相应的时钟信号由主控器产生，发送器在这个时钟信号释放 SDA，使它处于高电平状态，

以便接收由接收器在这位发出的应答信号。这时接收器还必须 SCL 在这位高电平期间，在 SDA 上输出一个恒定低电平信号以完成应答信号的输出。整个传输过程中，传输的字节数目是没有限制的。数据传输一段时间后，接收器无法继续接收更多的数据，主控器同样可以终止数据的传送^[4]。

24C01 是一种 128 字节串行 CMOS EEPROM,它具有如下特点: 1. 存储容量为 128 字节。2. 串行接口可使用普通两根 I/O 接口。3. 具有页写模式: 每页 4 字节。4. 同步周期小于 10ms^[4]。它只使用一条数据线和一条时钟线,采用 ATMEL 公司的 24C01 串口存储器,应用简单方便,但是其编程较为复杂。

2.3.2 2864A 芯片扩展:

2864A 是一种并行 EEPROM,它的特点同上,但每页有 16 字节,2864A 与 8051 单片机的接口电路如下图所示,2864A 的片选端 \overline{CE} 与高地址线 P2. 7 连接, P2. 7=0 才能选中 2864A, 这种线选法决定了 2864A 对应多组地址空间, 即 0000H~1FFFH,2000H~3FFFH,4000H~5FFFH,6000H~7FFFH,这 8K 字节存储器可作为数据存储器使用,但掉电后数据不丢失^[1]。

2864A 的四种工作方式:

(1) 维持方式: 当 \overline{CE} 为高电平时, 2864A 进入低功耗维持状态。此时, 输出线呈高阻状态, 芯片的电流从 140mA 下降至维持电流 60mA。

(2) 读方式: 当 \overline{CE} 和 \overline{OE} 均为低电平而 \overline{WE} 为高电平时, 内部的数据缓冲器被打开, 数据送上总线, 此时, 可进行读操作。

(3) 写方式: 2864A 提供了两种数据写入方式: 页写入和字节写入。

页写入: 为了提高写入速度, 2864A 片内设置了 16 字节的“页缓冲器”, 并将整个存储器阵列划分成 512 页, 每页 16 个字节。页的区分可由地址的高 9 位(A4~A12)来确定, 地址线的低四位(A0~A3)用以选择页缓冲器中的 16 个地址单元之一。对 2864A 的写操作可分为两步来实现: 第一步, 在软件控制下把数据写入页缓冲器, 这部称为页装载, 与一般的静态 RAM 写操作是一样的。第二步, 在最后一个字节(即第 16 个字节)写入到页缓冲器后 20ns 自动开始, 把页缓冲器的内容写到 EEPROM 阵列中对应的地址单元中, 这一步成为页存储。

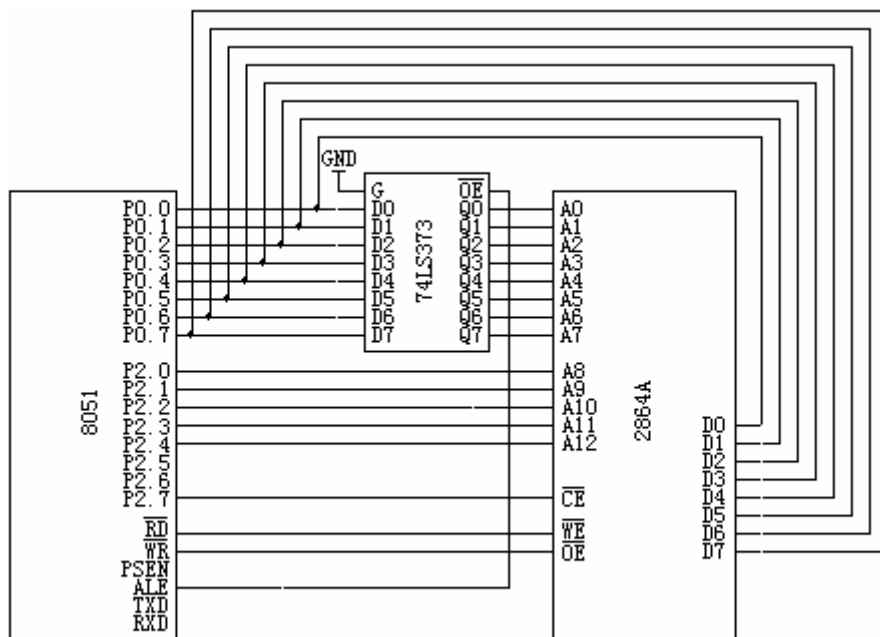


图2.1 2864A扩展电路

写方式时， \overline{CE} 为低电平，在 \overline{WE} 下降沿，地址码 A0~A12 被片内锁存器锁存，在上升沿时数据被锁存片内还有一个字节装载限时定时器，只要时间未到，数据可以随机地写入页缓冲器。在连续向页缓冲器写入数据的过程中，不用担心限时定时器会溢出，因为每当 \overline{WE} 下降沿时，限时定时器自动被复位并重新启动计时。限时定时器要求写入一个字节数据的操作时间 T_{BLW} 须满足： $3\mu S < T_{BLW} < 20\mu S$ ，这样是正确完成对 2864A 页面写入操作的关键。当一页装载完毕，不再有 \overline{WE} 信号时，限时定时器将溢出，于是页存储操作随即自动开始。首先把选中页的内容擦除，然后写入的数据由页缓冲器传递到 EEPROM 阵列中。

字节写入：字节写入的过程与页写入的过程类似，不同之处在于仅写入一个字节，限时定时器就溢出。

(4) 数据查询方式：数据查询是指用软件来检测写操作中的页存储周期是否完成。在页存储期间，如对 2864A 执行读操作，那么读出的是最后写入的字节，若芯片的转储工作未完成，则读出数据的最高位是原来写入字节最高位的反码。据此，CPU 可判断芯片的编程是否结束。如果读出的数据与写入的数据相同，表示芯片已完成编程，CPU 可继续向 2864A 装载下一页数据。并且编程起来比较简单，所以此设计采用此方案。

2.4 LED 显示电路选择

LED 显示器是由 N 个 LED 显示块拼接成 N 位 LED 显示器。N 个 LED 显示块有

N 跟位选线，根据显示方式的不同，位选线和段选线的连接方法也各不相同，段选线控制显示字符的字型，而位选线为各个 LED 显示块的公共端，它控制该 LED 显示位的亮、暗。LED 显示器有静态显示和动态显示两种显示方式。

2.4.1 LED 静态显示方式

LED 显示器工作于静态显示方式时，各位的共阴极（或共阳极）连接在一起并接地（或+5V）；每位的段选线（a~dp）分别与一个 8 位的锁存器输出相连。所以称为静态显示。各个 LED 的显示字符一经确定，相应锁存器的输出将维持不变，直到显示另一个字符为止。也正因此如此，静态显示器的亮度都较高。这种显示方式接口编程容易。付出的代价是占用口线较多，若用 I/O 接口，则要占用 4 个 8 位 I/O 口，若用锁存器接口，则要用 4 片 74LS373 芯片。如果显示器位数增多，则静态显示方式更是无法适应，因此在显示位数较多的情况下，一般都采用动态显示方式。

2.4.2 LED 动态显示方式

在多位 LED 显示时，为了简化硬件电路，通常将所有位的段选线相应的并联在一起，有一个 8 位 I/O 口控制，形成段选线的多路复用。而各位的共阳极或共阴极分别由相应的 I/O 线控制，实现各位的分时选通。其中段选线占用一个 8 位 I/O 口，而位选线占用一个 4 位 I/O 口。由于各位的段选线并联，段码的输出对各位来说都是相同的，因此，同一时刻，如果各位位选线都处于选通状态的话，4 位 LED 将显示相同的字符。若要各位 LED 能够显示出与本位相应的显示字符，就必须采用扫描显示方式，即在某一时刻，只让某一位的位选线状态，而其他各位的位选线处于关闭状态，同时，段选线上输出相应位要显示字节的段码。在确定 LED 不同位显示的时间间隔，不能太短，因为发光二极管从导通到发光有一定的延时，导通时间太短，发光太弱人眼无法看清。但也不能太长，因为毕竟要受限于临界闪烁频率，而且此时间越长，占用 CPU 时间也越多，另外，显示位增多，也将占用大量的 CPU 时间，因此动态显示实质是一牺牲 CPU 时间来换取元件的减少。

所以，由于本系统只涉及到 2 位显示输出，就采用了和 2 片 8 位移位寄存器串级使用的 LED 静态显示方式。

2.5. 键盘输入电路

2.5.1 矩阵式键盘接口：

矩阵式键盘（也称行列式键盘）适用于按键数目较多的场合，它由行线和列线组成，按键位于行列的交点上。一个 3×3 的行列结构可以构成一个有 9 个按键的键盘。同理，一个 4×4 的行列结构可以构成一个 16 键的键盘，很明显，在按键数量较多的场合，矩阵式键盘与独立式键盘相比，要节省很多的 I/O 口线。按键设置在行列线交点

上，行列线分别接到按键开关两端。行线通过上拉电阻接到+5V 上。平时无按键按下时，行线处于高电平状态，而当有按键按下时，行线电平状态将由于此行线相连的列线电平决定。列线电平如果为低电平，则行线电平为低电平，列线电平如果为高电平，则行线电平为高电平。这是识别矩阵键盘按键是否按下的关键所在^[1]。由于矩阵键盘中行列线为多键公用，各按键均影响该键所在行列的电平。因此各按键彼此将相互发生影响，所以必须将行列信号配合起来比做适当的处理，才能确定闭合键的位置。

2.5.2 独立式按键接口：

独立式按键就是各按键相互独立，每个按键各接入一根输入线，一根输入线上的按键工作状态不会影响其他输入线上的工作状态。因此，通过检测输入线的电平状态可以很容易判断哪个按键按下了。独立式按键电路配置灵活，软件简单。但每个按键需要占用一个输入口线，在按键数量较多时，需要较多的输入口线且电路结构复杂，故此种键盘适用于按键较少或操作速度较高的场合。

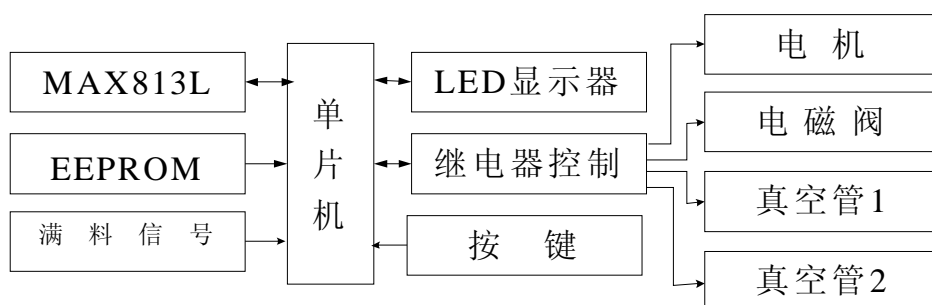
由于此系统中共有启动两条生产线的“启动 1”键和“启动 2”键、分秒选择键、时间设置加、时间设置减、显示生产线状态的切换键，时间设置键、时间切换键。只有这 8 个键，比较简单。所以就采用独立式按键接口电路。

第三章 自动加料机主电路

主电路采用 AT89C51，由于 AT89C51 内含 4KB 容量，因此在设计中不需要外扩 ROM。硬件电路主要有 LED 显示电路、键盘接受电路、继电器控制电路、EEPROM 外部存储器扩展电路，以及看门狗 MAX813L 等组成。

3.1 系统结构原理图

主电路采用 AT89C51，由于 AT89C51 内含 4KB 容量，因此在设计中不需要外扩 ROM。硬件电路主要有 LED 显示电路、键盘接受电路、继电器控制电路、EEPROM 外部存储器扩展电路，以及看门狗 MAX813L 等组成。电路原理框图如图所示：



自动加料机控制系统硬件框图

3.2 主机电路核心器件介绍

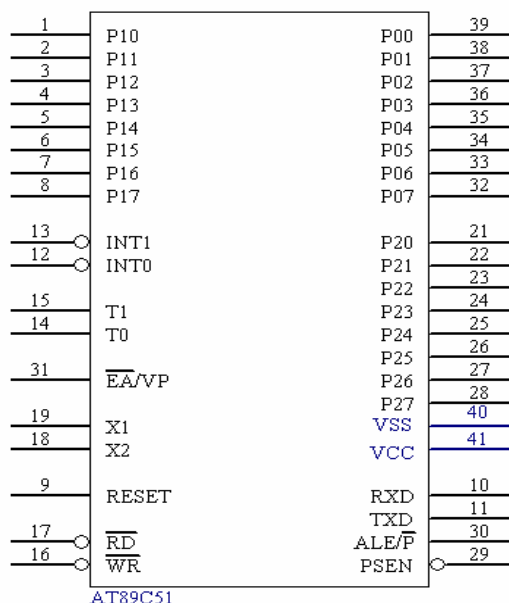
AT89C51 是美国 ATMEL 公司生产的低电压，高性能 CMOS 8 位单片机，片内含 4K bytes 的可反复擦写的只读程序存储器（PEROM）和 128bytes 的随机存取数据存储器（ROM），器件采用 ATMEL 公司的高密度、非易失性存储技术生产，兼容标准 MCS-51 指令系统，片内置通用 8 位中央处理器（CPU）和 Flash 存储单元。功能强大 AT89C51 单片机可提供许多高性价比的应用场合，可灵活应用于各种控制领域。

3.2.1 AT89C51 主要性能参数

- . 与 MCS-51 产品指令系统完全兼容
- . 4K 字节可重擦写 Flash 闪速存储器
- . 1000 次擦写周期
- . 全静态操作：0Hz---24MHz
- . 三级加密程序存储器
- . 128×8 字节内部 RAM
- . 32 个可编程 I/O 口线
- . 2 个 16 位定时/计数器
- . 6 个中断源
- . 可编程串行 UART 通道
- . 低功耗空闲和掉电模式

3.2.2 AT89C51 功能特性概述

AT89C51 提供以下标准功能：4K 字节 Flash 闪速存储器，128 字节内部 RAM，32 个 I/O 口线，两个 16 位定时/计数器，一个 5 向量两级中断结构，一个全双工串行通信口，片内振荡器及时钟电路。同时，AT89C51 可将至 0Hz 的静态逻辑操作，并支持两种软件可选的节电工作模式。空闲方式停止 CPU 的工作，但允许 RAM，定时/计数器，串行通信口及中断系统继续工作。掉电方式保存 RAM 中的内容，但振荡器停止工作并禁止其他所有部件工作直到下一个硬件复位。



. Vcc: 电源电压

. GND: 地

. P0 口: P0 口是一组 8 位漏极开路型双向 I/O, 也即地址/数据总线复用口。作为输出口用时, 每位能吸收电流的方式驱动 8 个 TTL 逻辑门电路, 对端口写“1”可作为高阻抗输入端用

在访问外部数据存储器或程序存储器时, 这组口线分时转换地址(低 8 位)和数据总线复用, 在访问期间激活内部上拉电阻。

在 Flash 编程时, P0 接收指令字节, 而在程序校验时, 输出指令字节, 校验时, 要求外接上拉电阻。

. P1 口: P1 口是一个带内部上拉电阻的 8 位双向 I/O 口, P1 的输出缓冲级可驱动(吸收或输出电流) 4 个 TTL 逻辑门电路。对端口写“1”, 通过内部的上拉电阻把端口拉倒高电平, 此时可作输入口。作输入口使用时, 因为内部存在上拉电阻, 某个引脚被外部信号拉低时会输出一个电流 (I_{IL})。

Flash 编程和程序校验期间, P1 接收低 8 位地址。

. P2 口: P2 口是一个带有内部上拉电阻的 8 位双向 I/O 口, P2 的输出缓冲级可驱动(吸收或输出电流) 4 个 TTL 逻辑门电路。对端口写“1”, 通过内部的上拉电阻把端口拉到高电平, 此时可作输入口, 作输入口使用时, 因为内部存在上拉电阻, 某个引脚被外部信号拉低时会输出一个电流 (I_{IL})。

在访问外部程序存储器或 16 位地址的外部数据存储器(例如执行 MOVE @DPTR 指令)时, P2 口送出高 8 位地址数据。在访问 8 位地址的外部数据存储器(例如执行 MOVX @RI 指令)时, P2 口线上的内容(也即特殊功能寄存器(SFR)区总 R2 寄存器的内容), 在整个访问期间不改变。

Flash 编程或校验时, P2 亦接收高位地址和其他控制信号。

. P3 口: P3 口是一组带有内部上拉电阻的 8 位双向 I/O 口, P3 口输出缓冲级可驱动(吸收或输出电流) 4 个 TTL 逻辑门电路。对 P3 口写入“1”时, 它们被内部上拉电阻拉高并可作为输入端口。作输入端时, 被外部拉低的 P3 口将用上拉电阻输出电流 (I_{IL})。

P3 口除了作为一般的 I/O 口线外, 更重要的用途是它的第二功能, 如下表所示:

端口引脚	第二功能
P3.0	RXD (串行输入口)
P3.1	TXD (串行输出口)
P3.2	$\overline{INT0}$ (外中断 0)
P3.3	$\overline{INT1}$ (外中断 1)
P3.4	T0 (定时/计数器 0)
P3.5	T1 (定时/计数器 1)
P3.6	\overline{WR} (外部数据存储器写选通)
P3.7	\overline{RD} (外部数据存储器读选通)

P3 口还接收一些用于 Flash 闪存存储器编程和程序校验的控制信号。

. RST: 复位输入。当振荡器工作时, RST 引脚出现两个机器周期以上高电平将单片机复位。

. ALE/ \overline{PROG} : 当访问外部程序存储器或数据存储器时, ALE (地址锁存允许) 输出脉冲用于锁存地址的低 8 位字节。即使不访问外部存储器。ALE 仍一时钟振荡频率的 1/6 输出固定的正脉冲信号, 因此它可对外输出时钟或用于定时目的。但要注意的是: 每当访问外部数据存储器时将跳过一个 ALE 脉冲。

对 Flash 存储器编程期间, 该引脚还用于输入编程脉冲 (\overline{PROG})。

如有必要, 可通过对特殊功能寄存器 (SFR) 区中的 8EH 单元的 D0 位置位, 可禁止 ALE 操作。该位置位后, 只有一条 MOVX 和 MOVC 指令 ALE 才会被激活, 此外, 该引脚会被微弱拉高, 单片机执行外部程序时, 应设置 ALE 无效。

. \overline{PSEN} : 程序存储允许 (\overline{PSEN}) 输出是外部程序存储器的读选通信号, 当 AT89C51 由外部程序存储器取指令 (或数据) 时, 每个机器周期两次 \overline{PSEN} 有效, 即输出两个脉冲, 在此期间, 当访问外部数据存储器, 这两次有效的 \overline{PSEN} 信号不出现。

. EA/VPP: 外部访问允许, 欲使 CPU 仅访问外部程序存储器 (地址为 0000H--FFFFH), EA 端必须保持低电平 (接地)。需要注意的是: 如果加密位 LB1 被编程, 复位时内部会锁存 EA 端状态。

如 EA 端为高电平 (接 Vcc 端), CPU 则执行内部会锁存 EA 端状态。

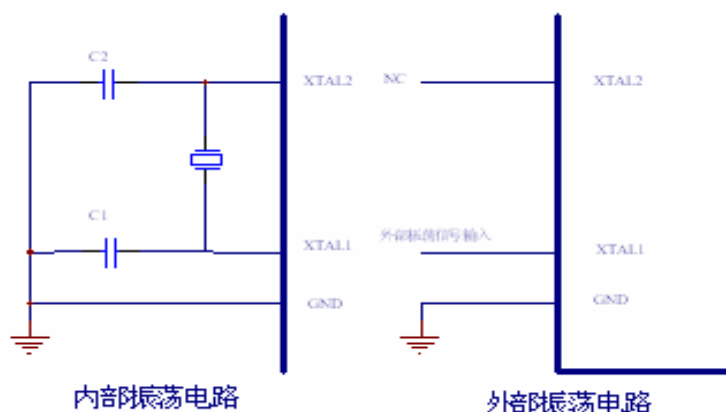
Flash 存储器编程时, 该引脚加上+12V 的编程允许电源 Vpp, 当然这必须是该器件是使用 12V 编程电压 Vpp。

. XTAL1: 振荡器反相放大器的及内部时钟发生器的输入端。

. XTAL2: 振荡器 3 放大器的输出端。

3.2.4 时钟振荡器

AT89C51 中有一个用于构成内部振荡器的高增益反相放大器，引脚 XTAL 和 XTAL2 分别是该放大器的输入端和输出端。这个放大器与作为反馈元件的片外石英晶体或陶瓷谐振器一起构成自激振荡器，振荡电路如图：



外接石英晶体（或陶瓷振荡器）及电容 C1、C2 接在放大器的反馈回路中构成并联振荡电路。对外接电容 C1、C2 虽然没有十分严格的要求，但电容容量的大小会轻微影响振荡频率的高低，振荡器工作的稳定性，起振的难易程序及温度稳定性，如果使用石英晶体，则推荐电容使用 $30\text{pF} \pm 10\text{pF}$ ，而如使用陶瓷振荡器建议选择 $40\text{pF} \pm 10\text{pF}$ 。

用户也可以采用外部时钟，采用时钟的电路如图。在这种情况下，外部时钟脉冲接到 XTAL1 端，即内部时钟发生器的输入端，XTAL2 则悬空。

由于外部时钟信号是通过一个 2 分钟触发器后作为内部时钟信号的，所以对外部时钟信号的占空比没有特殊要求，但最小高电平持续时间和最大的低电平持续时间应符合产品技术条件的要求。

3.2.5 空闲节电模式

AT89C51 有两种可用软件编程的省电模式，它们是空闲模式和掉电工作模式。这两种方式是控制专用寄存器 PCON（即电源控制寄存器）中的 PD（PCON.1）和 IDL（PCON.0）位来实现的。PD 是掉电模式，当 PD=1 时，激活掉电工作模式，单片机模式，即 PD 和 IOL 同时为 1，则先激活掉电模式。

在空闲工作模式状态，CPU 保持睡眠状态而所有片内的外设保持激活状态，这种方式由软件产生。此时，片内 RAM 和所有特殊功能寄存器的内容保持不变。空闲模式可由任何允许的中断请求或硬件复位终止。

终止空闲工作模式的方法有两种，其一是任何一条被允许中断的事件被激活，IDL (PCON.0) 被硬件清除，即刻终止空闲工作模式。程序会首先响应中断，进入中断服务程序，执行完中断服务程序并紧随 RETI (中断返回) 指令后，下一条要执行的指令就是使单片机进入空闲模式那条指令后面的一条指令。

其二是通过硬件复位也可将空闲工作模式终止。需要注意的是，当有硬件复位来终止空闲工作模式时，CPU 通常是从激活空闲模式那条指令的下一条指令开始继续执行程序，要完成内部复位操作，硬件复位脉冲要保持两个机器周期 (24 个时钟周期) 有效，在这种情况下，内部禁止 CPU 访问片内 RAM，而允许访问其它端口。为了避免可能对端口产生意外写入，激活空闲模式的那条指令后一条指令不应是一条对端口或外部存储器的写入指令。

3.2.6 掉电模式

在掉电模式下，振荡器停止工作，进入掉电模式的指令是最后一条被执行的指令，片内 RAM 和特殊功能寄存器的内容在终止掉电模式前被冻结。退出掉电模式的唯一方法是硬件复位，复位后将重新定义全部特殊功能寄存器但不改变 RAM 中的内容，在 V_{cc} 恢复到正常电平前，复位应无效，且必须保持一定时间以使振荡器重新启动并稳定工作。

空闲和掉电模式外部引脚状态

模式	程序存储器	ALE	/PSEN	P0	P1	P2	P3
空闲模式	内部	1	1	数据	数据	数据	数据
空闲模式	外部	1	1	浮空	数据	地址	数据
掉电模式	内部	0	0	数据	数据	数据	数据
掉电模式	外部	0	0	浮空	数据	数据	数据

3.2.7 程序存储器的加密

AT89C51 可使用对芯片上的 3 个加密位 LB1、LB2、LB3 进行编程 (P) 或不编程 (U) 来得到下表所示的功能:

加密位保护功能表

	程序加密位			保护类型
	LB1	LB2	LB3	
1	U	U	U	没有程序保护功能
2	P	U	U	禁止从外部程序存储器中执行 MOVC 指令读取内部程序存储器的内容
3	P	P	U	除上表功能外, 还禁止程序校验
4	P	P	P	除以上功能外, 同时禁止外部执行

当加密位 LB1 被编程时, 在复位期间, EA 端的逻辑电平被采样并锁存, 如果单片机上电后一直没有复位, 则锁存起的初始值是一个随机数, 且这个随机数会一直保存到真正复位为止, 为使单片机能正常工作, 被锁存的 EA 电平必须与该引脚当前的逻辑电平一致。此外, 加密位只能通过整片擦除的方法清除。

Flash 闪速存储器的编程

AT89C51 单片机内部有 4K 字节的 Flash PEROM, 这个 Flash 存储阵列出厂时已处于擦除状态 (即所有存储单元的内容均为 FFH), 用户随时可对其进行编程。编程接口可接收高电压 (+12V) 或低电压 (Vcc) 的允许编程信号。低电压编程模式适合于用户在线编程系统, 而高电压编程模式可与通用 EPROM 编程器兼容。

AT89C51 单片机中, 有些属于低电压编程方式, 而有些则是高电压编程方式。用户可从芯片上的型号和读取芯片内的签名字节获得该信息。如图

	Vpp=12V	Vpp=5V
芯片顶面标识	AT89C51 xxxx yyww	AT89C51 xxxx—5 yyww
签名字节	(030H) =1EH (031H) =51H (032H) =FFH	(030H) =1EH (031H) =51H (032H) =05H

AT89C51 的程序存储器阵列采用字节写入方式编程的，每次写入一个字节，要对整个芯片内的 PEROM 程序存储器写入一个非空字节，必须使用擦除的方式将整个存储器的内容清楚。

3.2.8 编程方法

编程前，先设置好地址，数据及控制信号，编程单元的地址加在 P1 口和 P2 口的 P2.0—P2.3(11 位地址范围为 0000H—0FFFH)，数据从 P0 口输入，引脚 P2.6、P2.7 和 P3.6、P3.7 的电平，PSEN 为低电平，RST 保持高电平，EA/Vpp 引脚是编程电源的输入端，按要求加上编程电压，ALE/PROG 引脚输入编程脉冲（负脉冲）。编程时，可采用 4—20MHz 的时钟振荡器，AT89C51 编程方法如下：

1. 在地址线上加上要编程单元的地址信号。
2. 在数据线上加上要写入的数据字节。
3. 激活相应的控制信号。
4. 在高电压编程方式时，将 /EA/Vpp 端加上 +12V 编程电压。
5. 每对 Flash 存储阵列写入一个字节或每写入一个程序加密位，加上一个 ALE/PROG 编程脉冲。

改变编程单元的地址和写入的数据，重复 1—5 步骤，直到全部文件编程结束。

每个字节写入周期是自身定时的，通常约为 1.5ms

3.2.9 数据查询

AT89C51 单片机用数据查询方式来检测一个写周期是否结束，在一个写周期中，如需读取最后写入的那个字节，则读出的数据的最高位（P0.7）是原来写入字节最高的反码，写周期完成后，有效的数据就会出现在所有输出端上，此时，可进入下一个字节的写周期，写周期开始后，可在任意时刻进行数据查询。

Ready/Busy：字节编程的进度可通过 RDY/BSY 输出信号监测，编程期间，ALE 变成高电平“H”后 P3.4 端电平被拉低，表示正在编程状态。编程完成后，P3.4 变为高电平表示准备就绪状态。

3.2.11 程序校验及芯片擦除

如果加密位 LB1、LB2 没有进行编程，则代码数据可通过地址和数据线读回原编写的数据。采用下图电路。程序存储器的地址由 P1 和 P2 口的 P2.0-P2.3 输入，数据

有 P0 口读出，P2.6、P2.7 和 P3.6、P3.7 的控制信号 \overline{PSEN} 保持低电平，ALE、 \overline{EA} 和 RST 保持高电平。校验时 P0 口须接上 10K 左右的上拉电阻。

加密位不可直接校验，加密位的校验可通过对存储器的校验和写入状态来验证。

利用控制信号的正确组合并保持 ALE/ \overline{PROG} 引脚 10ms 的低电平脉冲宽度即可将 PEROM 阵列（4k 字节）和三个加密位整片擦除，代码阵列在片擦除操作中将任何非空单元写入“1”，这步骤需再编程之前进行。

3.2.12 读片内签名字节及编程接口

AT89C51 单片机内有 3 个签名字节，地址为 030H、031H 和 032H。用于声明该器件的厂商、型号和编程电压。读签名字节的过程和单元 030H、031H 和 032H 的正常校验相仿，只需将 P3.6、P3.7 保持低电平，返回值意义如下：

(030H) =1EH 声明产品由 ATMEL 公司制造。

(031H) =51H 声明为 AT89C51 单片机。

(032H) =FFH 声明为 12V 编程电压。

(032H) =05H 声明为 5V 编程电压。

编程接口：采用控制信号的正确组合可对 Flash 闪速存储阵列中的每一代码字节进行写入和存储器的整片擦除，写操作周期是自身定时的，初始化后它将自动定时到操作完成。

3.2.13 AT89C51 的极限参数：

极限参数：工作温度.....-55°C to +125°C

储藏温度.....-65°C to +150°C

任一引脚对地电压.....-1.0V to +7.0V

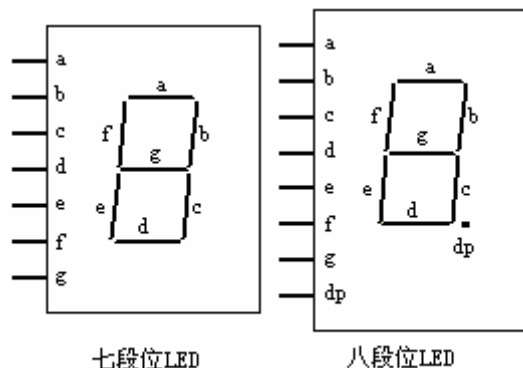
最高工作电压.....6.6V

直流输出电流.....15.0mA

3.3 显示电路

在单片机应用系统中，如果需要显示的内容只有数码和某些字母，使用 LED 数码管是一种较好的选择。LED 数码管显示清晰、成本低廉、配置灵活，与单片机接

口简单易行。LED 数码管是由发光二极管作为显示字段的数码型显示器件，其中七只发光二极管分别对应 a~g 笔端构成“日”字形，另一只发光二极管 Dp 作为小数点。因此这种 LED 显示器称为七段数码管或八段数码管。如图所示：

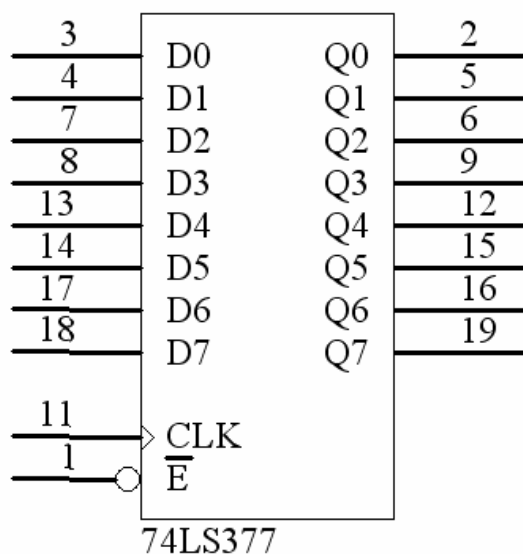


LED 数码管按电路中的连接方式可分为共阴型和共阳型两大类，共阳型是将各段发光二极管的正极连在一起，作为公共端 COM，公共端 COM 接高电平，a~g、Dp 各笔段通过限流电阻接控制端。某笔段控制端低电平时，该笔段发光，高电平时不发光。控制某几段笔端发光，就能显示出某个数码或字符。共阴型是将各段发光二极管的负极连在一起，作为公共端 COM 接地，某笔段通过限流电阻接高电平时发光。

在自动加料机控制系统中运行是要显示输送、排料、满料、空料时间，有时间切换键和标志哪条生产线的发光二极管表示，显示的位数少，所以就采用静态显示的方式。LED 显示器工作于静态显示方式时，各位的共阴极（或共阳极）连接在一起并接地（或+5V）；每位的段选线（a~dp）分别与一个 8 位的锁存器输出相连。所以称为静态显示。各个 LED 的显示字符一经确定，相应锁存器的输出将维持不变，直到显示另一个字符为止。也正因此如此，静态显示器的亮度都较高。这种显示方式接口编程容易。若用 I/O 接口，则要占用 4 个 8 位 I/O 口，若用锁存器接口，则要用 4 片 74LS373 芯片。如果显示器位数增多，则静态显示方式便无法适应。

在设计中，LED 显示电路采用 74LS377 驱动器和 MC14511B 译码器控制 LED 数码管。

3.3.1 74LS377 芯片介绍

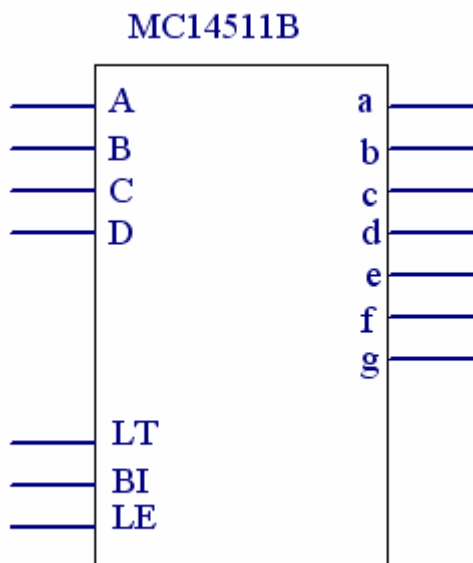


- . $D_0 \sim D_7$: 8 个信号输入端。
- . $Q_0 \sim Q_7$: 8 个信号输出端。
- . CLK: 时钟信号输入端。
- . \bar{E} : 锁存允许信号。当 $\bar{E}=0$ 时, CLK 端的上跳变将把 8 位 D 输入端的数据打入 8 位锁存器。

74LS377 真值表

\bar{G}	CLK	D	Q
1	X	X	Q0
0	↑	1	1
0	↑	0	0
X	0	X	Q0

3.3.2 MC14511B 芯片介绍



. A~D: 四个信号输入端。

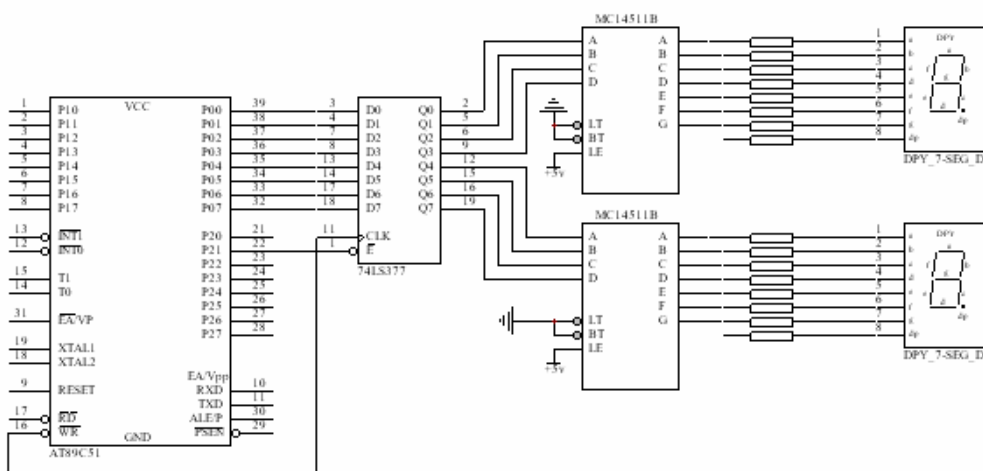
. a~g: 七个信号输出端。

. LT、BI: 接地。

. LE: 接电源。

3.3.3 LED 接口电路

LED 显示电路采用 74LS377 驱动器和 MC14511B 译码器控制 LED 数码管。两个 LED 显示一条生产线一个工作过程的秒数。两片 MC14511B 把 P0 口的高四位和低四位译码成十进制控制 LED 显示。电路图如下：

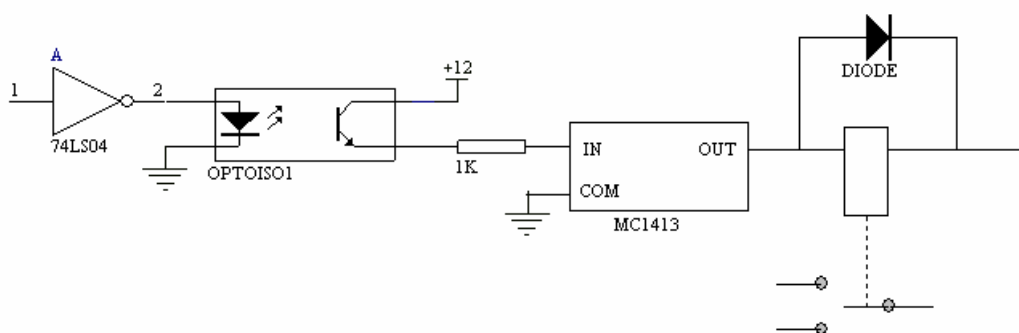


3.4 继电器控制电路

在电气控制领域或产品中，凡是需要逻辑控制的场合，几乎都需要使用继电器，从家用电器到工农业应用，甚至国民经济各个部门，可谓无所不见。继电器是一种利用各种物理量的变化，将电量或非电量信号转化为电磁力（有触头式）或使输出状态发生阶跃变化（无触头式），从而通过其触头或突变量促使在同一电路或另一电路中的其它器件或装置动作的一种控制元件。根据转化的物理量的不同，可以构成各种各样的不同功能的继电器，以用于各种控制电路中进行信号传递、放大、转换、联锁等，从而控制主电路和辅助电路中的器件或设备按预定的动作程序进行工作，实现自动控制和保护的目的。被转化或施加于继电器的电量或非电量称为继电器的激励量，当继电器被激励，从一个起始位置达到预定的工作位置，并完成电路的切换动作，称为继电器的工作特性，包括吸合、不吸合、保持与释放状态。当输入量变化到高于它的吸合值或低于它的释放值时，继电器动作，对于有触头式继电器其触头闭合或断开，对于无触头式继电器起输出发生阶跃变化，以此提供一定的逻辑变量。

自动加料机是把塑料粒子送到一个真空管，在输送时真空管关闭合，排料时真空管需要打开，将粒子送到排料漏斗。本设计共需 2 个继电器控制交流接触器，一个接带动生产的电动机。另一个为控制工作方向的方向阀。经考虑采用 4123 无极 12V 直流控制 24V 的交流继电器，并利用光耦合器件 P521 和 MCU 隔开。89C51 的 P1 初始值为 OFFH，所以加一个 74LS04 反相器使得继电器初始不产生闭合，使用 MC1413 为无源驱动器，IN4007 组成继电器的续流二极管。

电路图如下：



3.5 键盘及显示电路

键盘在单片机应用系统中能实现向单片机输入数据、传送命令等功能，是人工干扰单片机的主要手段。本设计采用 8255A 为 I/O 扩展。

3.5.1 键盘接口

非编码键盘与单片机的接口单片机系统所用的键盘有编码键盘和非编码键盘两种。

编码键盘本身除了按键之外，还包括产生键码的硬件电路，只要按下某一个键，就能产生这个键的代码，一般称为键码，同时，还能产生一个脉冲信号，以通知 CPU 接收（输入）键码。这种键盘的使用比较方便，亦不需要编写很多程序，但使用的硬件较复杂，在微型计算机控制系统中使用还不多。

非编码键盘是由一些按键排列成的一个行列矩阵。按键的作用，只是简单地实现接点的接通和断开，但必须有一套相应的程序与之配合，才能产生出相应的键码。非编码键盘几乎不需要附加什么硬件电路，目前，在微型计算机控制系统中使用比较普遍。

使用非编码键需要用软件来解决按键的识别，防止抖动以及键码的产生等工作。

设有一个 6 行×5 列的非编码键盘，其中有 16 个为数字键 0~F，其余的为控制键，用以发布各种控制命令。键盘的行线接 8155C 口的六条线 PC5~PC0，键盘的列线则接 8155B 口的 5 条线。在没有任何键按下时，所有键盘列线上的信号都是高电平。当有按键按下时，就会出现键的识别、防止抖动以及确定键码等一系列问题。

按键识别有各种方法，此系统只“行扫描”法：

(1) 确定是否有按键按下。CPU 通过并行口输出 000000 到键盘的行线，然后检测键盘的列线信号。若没有键按下，则为 11111。若有任一个按键按下，则有某一条列线为 0，也就是当 PB4~PB0 不为 11111 时，就表示有键按下。

(2) 通过“行扫描”确定已按键的行、列位置。所谓行扫描就是依次给每条行线输入 0 信号，而其余各行都输入 1，并检测每次扫描时所对应的列信号。在图 2 中就是在 C 口先输出 111110 (PC5~PC0)，然后是 111101，直到最后是 011111，并检测每次所对应的 B 口输入。

只有在某行上有键按下时，在这一行上输入 0 (其他行为 1)，在列输出上才能检测到 0 信号。若是输入为 0 的这一行上没有按键按下，则收到的列信号仍然全是 1。因此，只要记下列信号不全为 1 时的 C 口输出及 B 口输入，就能确定以按键的位置。设图 2 中处于第 3 行第 1 列的键已按下，则必须是行输出信号为 110111，检测到的列信号为 11101。对应于其他的行信号，列信号都是 11111。这样，通过行扫描，就

可以确定按键的行、列坐标。

(3) 确定是否有多键同时按下。有时一次按下的键不止一个，这在一般情况下是由于误操作引起的，是不应该出现的通常称为窜键。出现这种情况时，就可能不止一次会得到列信号不为全 1，这时就不容易判断哪个键是真正需要按下的。为了处理这种情况可采取两种办法：一是行扫描一定是扫到最后一行才结束，而不是检测到列信号不为全 1 时就结束，以便发现窜键；二是如果出现了窜键，最简单的处理办法就是这次行扫描不算，再来一遍，即以最后放开的那个键为准。实际上，由于扫描的速度很快，真正找到两个键同时按下的情况是很少的。

(4) 消除键抖动。一般按键在按下的时候有抖动的问题，即键的簧片在按下时会有轻微的弹跳，需经过一个短暂的时间才会可靠地接触。若在簧片抖动时进行扫描就可能得出不正确的结果。因此，在程序中要考虑防抖动的问题。最简单的办法是在检测到有键按下时，等待（延迟）一段时间再进行“行扫描”，延迟时间为 10~20ms。这可通过调用子程序来解决，当系统中有显示子程序时，调用几次显示子程序也能同时达到消除抖动的目的。

3.5.2 8255A 芯片介绍

自动加料系统有 8 个按键：启动两条生产线的“启动 1”键和“启动 2”键、分秒选择键、时间设置加/减键、显示生产线状态切换键，时间设置键，时间切换键，由于单片机的并行口有限。本系统采用 8255A 扩展并行口。

(1). 8255A 的内部结构

8255A 的内部结构有以下几部分组成：

a. 并行 I/O 端口 A、B、C 8255A 的内部有 3 个 8 位并行 I/O 口：A 口、B 口、C 口。3 个 I/O 口都可以通过编程选择为输入或输出，但在结构和功能上有所不同。

A 口：含有一个 8 位数据输出锁存/缓冲器和一个 8 位输入锁存器。

B 口：含有一个 8 位数据输出锁存/缓冲器和一个 8 位输入锁存器（不锁存）。

C 口：含有一个 8 位数据输出锁存/缓冲器和一个 8 位输入锁存器（不锁存）。

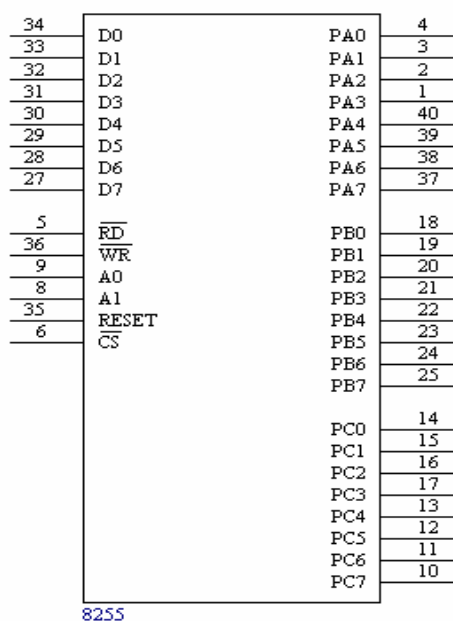
当数据传送不需要联络信号时，这 3 个端口都可以用作输入或输出。当 A 口 B 口需要有联络信号时，C 口可以作为 A 口和 B 口的联络信号线。

b. 工作方式控制电路：8255A 的三个端口在使用使可分为 A、B 两组。A 组包括 A 口 8 位和 C 口高 4 位；B 组包括 B 口 8 位和 C 口低 4 位。两组的控制电路中分别有控制寄存器，根据写入的控制字决定两组的工作方式，也可对 C 口每一位置“1”或清“0”。

c. 数据总线缓冲器：数据总线缓冲器是三态双向的 8 位缓冲器，是 8255A 与单片机数据总线的接口，8255A 的 D₀~D₇可以和 AT89C51 单片机的 P0.0~P0.7 直接相连。数据的输入输出、控制字和状态信息的传递，均可通过数据总线缓冲器进行。

d. 读/写控制逻辑：8255A 读/写控制逻辑的作用是从 CPU 的地址和控制总线上接受有关信号，转变成各种控制命令送到数据缓冲器及 A 组和 B 组的控制电路，控制 A、B、C3 个端口的操作。

3.5.3 8255A 引脚功能



8255A 共有 40 个引脚，一般为双列直插 DIP 封装，40 个引脚可分为与 CPU 连接的数据线、地址和控制信号以及与外围设备连接的三个端口线。

- . D0~D7：双向三态数据总线。
- . RESET：复位信号，输入，高电平有效。复位后，控制寄存器清 0，A 口、B 口、C 口被置为输入方式。
- . \overline{CS} ：片选信号，输入，低电平有效。
- . \overline{RD} ：读信号，输入，低电平有效。 \overline{RD} 有效时，允许 CPU 通过 8255A D0~D7 读取数据或状态信息。
- . \overline{WR} ：写信号，输入，低电平有效。有效时，允许
- . A₁A₀：端口控制信号，输入。2 位可构成四种状态，分别寻址 A 口、B 口、C 口和控制寄存器
- . PA₀~PA₇：A 口数据线，双向。
- . PB₀~PB₇：B 口数据线，双向。

. PC₀~PC₇: C 口数据/信号线, 双向。当 8255A 工作于方式 0 时, PC₀~PC₇分为两组(每组 4 位)并行 I/O 数据线; 当 8255A 工作于方式 1 或方式 2 时, PC₀~PC₇为 A 口、B 口提供联络信号。

A₁A₀与 \overline{RD} 、 \overline{WR} 、 \overline{CS} 信号一起, 可确定 8255A 的操作状态, 如图所示:

8255A 功能操作

A1	A0	\overline{RD}	\overline{WR}	\overline{CS}	操 作	
0	0	0	1	0	A 口→数据总线	输入操作
0	1	0	1	0	B 口→数据总线	
1	0	0	1	0	C 口→数据总线	
0	0	1	0	0	数据总线→A 口	输出操作
0	1	1	0	0	数据总线→B 口	
1	0	1	0	0	数据总线→C 口	
1	1	1	0	0	数据总线→控制口	
x	x	X	x	1	数据总线为高阻态	禁止操作
1	1	0	1	0	非法状态	
x	x	1	1	0	数据总线为高阻态	

8255A 与 AT89C51 接口电路。

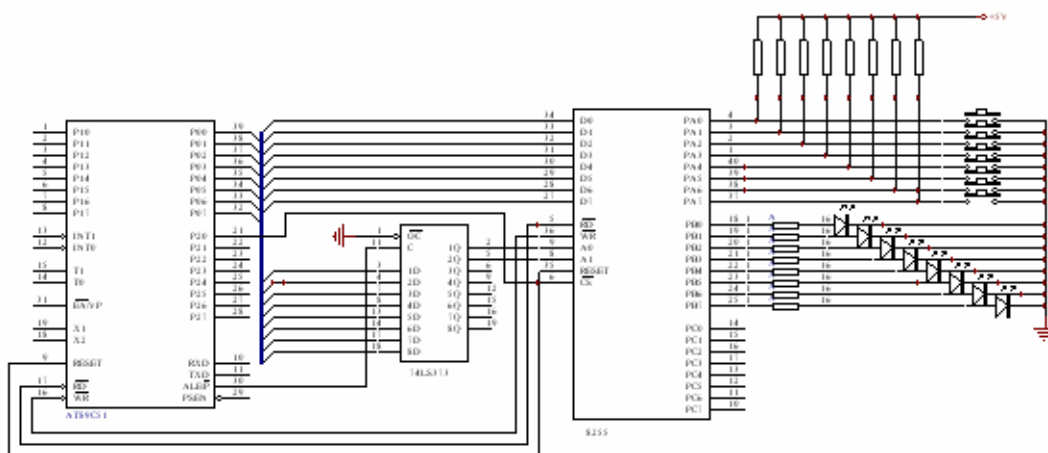
在连接键盘电路采用 8255A 为扩展 I/O 口时, 要在中间加入一个 74LS373, 74LS373 是一个三态门的 8D 锁存器, 它可以作为 AT89C51 外部的一个扩展输入口, 借口电路的工作原理是当外设把数据准备好后, 发出一个控制信号加到 373 的 G 端, 即锁存端, 使输入数据在 373 中锁存, 同时信号加到 AT89C51 单片机的中断请求 $\overline{INT0}$ 端, 单片机响应中断, 在中断服务程序中执行下面程序:

```
MOV DPTR, #0BFFFH
```

```
MOVX A, @DPTR
```

在执行上面的第二条指令时, P2.6=0, \overline{RD} 有效, 通过或门后加到 373 的 \overline{OE} 端, 即 373 的三态门控制端, 使三态门畅通, 锁存的数据读入到累加器 A 中。

电路如图所示:



3.6 外部存储器扩展电路

自动加料机存在两条生产线的输送、排料、满料、空料，共 8 个设置的重要数据需要读/写，并且能够提供掉点保护，所以需要外扩程序存储器。程序存储器一般采用自读存储器，因为这种存储器在电源关断后，仍能保存程序，在系统上电后，CPU 可取出这些指令予以重新执行。只读存储器简称 ROM。ROM 中的信息一旦写入之后，就不能随意更改，特别是不能在程序运行的过程中写入新的内容，故称之为只读存储器。

向 ROM 中写入信息叫做 ROM 编程。根据编程的方式不同，ROM 分为以下几种：

(1) 掩膜 ROM 是在制造过程中编程。因编程是以掩膜工艺实现的，因此称为掩膜 ROM。这种芯片存储结构简单，集成度高，但由于掩膜工艺由于成本较高，因此只适合于大批量生产。

(2) 可编程 ROM (PROM)

PROM 芯片出厂是并没有任何程序信息，是由用户用独立的编程器写入的，但 PROM 只能写入一次，写入内容后，就不能在进行修改。

(3) EPROM

EPROM 是用电信号编程，用紫外线擦除的只读存储器芯片。在芯片外壳上的中间位置有一个圆形窗口，通过这个窗口照射紫外线射就可擦除原有的信息。

(4) E²PROM

这是一种用电信号编程，也用电信号擦除的 ROM 芯片，对 E²PROM 的读写操作与 RAM 存储器几乎没有什么差别，只是写入的速度慢一些，但断电后能够保存信息。

此设计采用外扩 E²PROM 的扩展方式。E²PROM 是电擦除可编程自读存储器，其突

出优点是能够在线擦除和改写，无须像 EPROM 那样必须用紫外线照射才能擦彻，较新的 E²PROM 产品在写入时能自动完成擦除，且不再需要专用的编程电源，可以直接使用单片机系统的+5V 电源。

常用的 E²PROM 芯片有 2816/1816A, 2817/2817A, 2864A。本设计采用的是用 2864A 的扩展方式。

2864A 有四种工作方式：

(1) 维持方式

当 \overline{CE} 为高电平时，2864A 进入低耗维持方式。此时，输出线呈高阻态，芯片的电流从 140mA 降至维持电流 60mA。

(2) 读方式

当 \overline{CE} 和 \overline{OE} 均为低电平而 \overline{WE} 为高电平时，内部的数据缓冲器被打开，数据送上总线，此时，可进行读操作。

(3) 写方式

2864A 提供了两种数据方式：字节写入和页写入

页写入：2864A 片内设置了 16 字节的“页缓冲器”，并将整个存储器阵列划分成 512 页，每页 16 个字节。页的区分可由地址的高 9 位来确定，地址线的低 4 位用以选择页缓冲器中的 16 个地址单元之一。写方式时， \overline{CE} 为低电平，在 \overline{WE} 下降沿，地址码 A0~A12 被片内锁存器锁存，在上升时数据被锁存。片内还有一个字节装载限时定时器，只要时间未到，数据可以随机地写入页缓冲器。在连续向页缓冲器写入数据的过程中，不用担心限时定时器会益处，因为每当 \overline{WE} 下降沿时，限时定时器自动被复位并重新启动计时。

字节写入：字节写入的过程与页写入的过程类似，不同之处是仅写入一个字节，限时定时器就溢出。

(4) 数据查询方式

数据查询是指用软件来检测写操作中的页存储周期是否完成。

在页存储期间，如对 2864A 执行读操作，那么读出的是最后写入的字节，若芯片的转储工作未完成，则读出数据的最高位是原来写入字节最高位的反码。据此，CPU 可判断的编程是否结束。如果读出的数据与写入的数据相同，表示芯片已完成编程，CPU 可继续向 2864A 装载下一页数据。

2864A 与单片机接口时，2864A 的片选端 \overline{CE} 与高地址线 P2.7 连接，P2.7=0 才能选中 2864A，这种线选法决定了 2864A 对应多组地址空间，即：0000H~1FFFFH，2000H~3FFFFH，4000H~5FFFFH，6000H~7FFFFH。这 8K 字节存储器可作为数据存储使用，但掉电后数据不丢失。2864A 与 AT89C51 接口电路如图：单片机由于受引脚数的限制，数据线和地址线是复用的，由 P0 口兼用。为了将它们分离出来，以便同单片机片外的扩展芯片正确的连接，需要在单片机外部增加地址锁存器。我们这本设计

中使用 74LS373。74LS373 是一种带有三态门的 8D 锁存器，其引脚如图所示

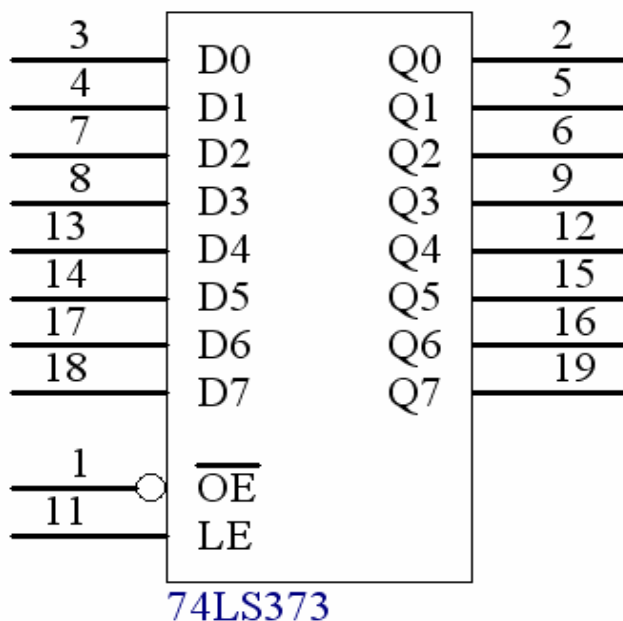


图 3.6.1 锁存器 74LS373 的引脚

其引脚的功能如下：

D7~D0：8 位数据输入线

Q7~Q0：8 位数据输出线

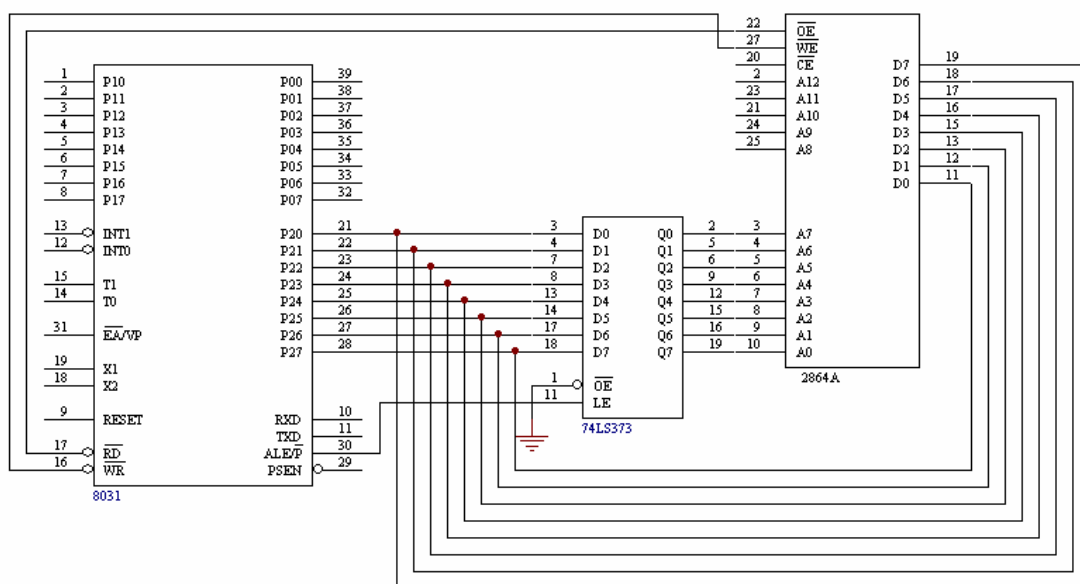
G：数据输入锁存选通信号，高电平有效。当该信号为高电平的时候，外部数据选通到内部锁存器，负跳变时，数据锁存到锁存器中。

\overline{OE} ：数据输出允许信号，低电平有效。当该信号为低电平的时候，三态门打开，锁存器中数据输出到数据输出线。当该信号为高电平的时候，输出线为高阻态。

74LS373 的功能表见表

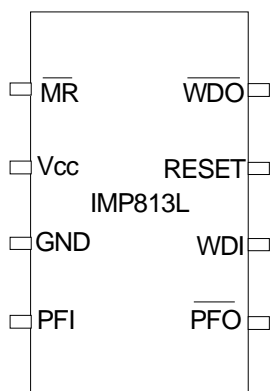
\overline{OE}	G	D	Q
0	1	1	1
0	1	0	0
0	0	×	不变
1	×	×	高阻态

连接电路图：



3.7 看门狗 MAX813L 电路

几乎所有的单片机都需要复位电路，对复位电路的基本要求是：在单片机上电时能可靠复位，在下电时能防止程序乱飞导致 EPROM 中的数据被修改；另外，单片机系统在工作时，由于干扰等各种因素的影响，有可能出现死机现象导致单片机系统无法正常工作，为了克服这一现象，除了充分利用单片机本身的看门狗定时器(有些单片机无看门狗定时器)外，还需外加看门狗电路；除此以外，有些单片机系统还要求在掉电瞬间单片机能将重要数据保存下来，因掉电的发生往往是根随机的，因而此类单片机系统需要电源监控电路，在掉电刚发生时能告知单片机。MAXIM 公司推出的 MAX813L 刚好能满足这些要求，下面具体介绍该芯片的性能特点及使用方法。



MAX813L 有双列直插和贴片两种封装形式，其双列直插如图所示，引脚功能如下：

第①脚为手动复位输入，低电平有效；第②、②脚分别为电源和地；第④脚为电源故障输入；第⑤脚为电源故障输出；第⑥脚为看门狗输入，第⑦脚为复位输出，第

⑧脚为看门狗输出。

MAX813L 的性能特点:

MAX813L 的内部结构框图如图所示, 具有以下主要性能特点: 由图可知该芯片具有以下主要性能特点:

(1) 复位输出。系统上电、掉电以及供电电压降低时, 第⑦脚产生复位输出, 复位脉冲宽度的典型值为 200ms, 高电平有效, 复位门限的典型值为 4.65V。

(2) 看门狗电路输出。如果在 1.6s 内没有触发该电路(即第⑥脚无脉冲输入), 则第⑧脚输出一个低电平信号。

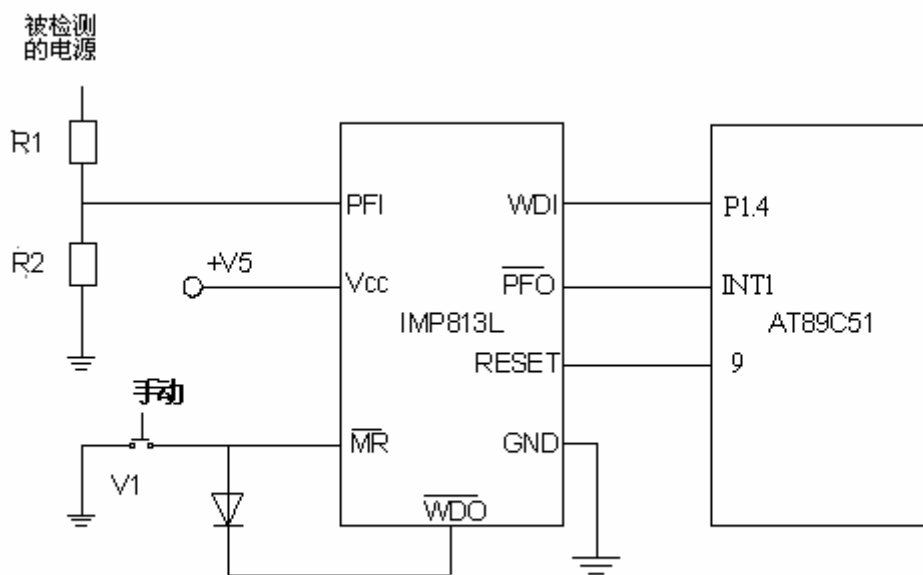
(3) 手动复位输入, 低电平有效, 即第①脚输入一个低电平, 则第⑦脚产生复位输出。

(4) 1.25V 门限值检测器, 第④脚为输入, 第⑤脚为输出。当第④脚电压低于 1.25V 时, 第⑤脚输出一个低电平信号。

MAX813L 的典型应用电路:

MAX813L 的典型应用电路如图所示。图中单片机以 AT89C51 为例, MAX813L 的第①脚与第⑧脚相连。第⑦脚接单片机的复位脚(AT89C51 的第⑨脚); 第⑥脚与单片机的 P1.4 相连。在软件设计中, P1.4 不断输出脉冲信号, 如果因某种原因单片机进入死循环, 则 P1.4 无脉冲输出。于是 1.6s 后在 MAX813L 的第⑧脚输出低电平, 该低电平加到第①脚, 使 MAX813L 产生复位输出, 使单片机有效复位, 摆脱死循环的困境。另外, 当电源电压低于门限值 4.65V 时, MAX813L 也产生复位输出, 使单片机处于复位状态, 不执行任何指令, 直至电源电压恢复正常, 可有效防止因电源电压较低时单片机产生错误的动作。

电源故障输入 PFI 通过一个电阻分压器监测未稳压的直流电源。当 PFI 低于 1.25V 时, 电源故障输出脚第⑤脚 PFO 变低, 可引起 AT89C51 中断, 进行电源故障处理, 或将重要数据保存下来。把分压器接到未稳压的直流电源是为了更早地对电源故障告警。



MAX813L 的典型应用电路

MAX813L 是一体积小、功耗低、性价比高的带看门狗和电源监控功能的复位芯片；它使用简单、方便，它所提供的复位信号为高电平，因而是应用于复位信号为高电平场合的单片机系统的理想芯片。

3.8 料位开关

在料体进入料斗式干燥机后，要能检测到什么时候到达满料状态，所以就选用简单的阻力式传感器，阻力式料位传感器是指物料对机械运动所呈现的阻挡力。粉末颗粒状物料比液态物质流动性差，对运动物体有明显的阻力，利用这一特点可构成各种料位传感器。

采用音叉法，音叉法：根据物料对振动中的音叉有无阻力探知料位是否到达或超过某高度，并发出通断信号，这种原理不需要大幅度的机械运动，驱动功率小，机械结构简单、灵敏而可靠。

音叉由弹性良好的金属制成，本身具有确定的固有频率，如外加交变力的频率与其固有频率一致，则叉体处于共振状态。由于周围空气对振动的阻尼微弱，金属内部的能量损耗又很少，所以只需微小的驱动功率就能维持较强的振动。当粉粒体物料触及叉体之后，能量消耗在物料颗粒间的摩擦上，迫使振幅急剧衰减，音叉停振。

为了给音叉提供交变的驱动力，利用放大电路对压电元件施加交变电场，靠逆压电效应产生机械力作用在叉体上。用另外一组压电元件的正压电效应检测振动，它把振动力为微弱的交电信号。再由电子放大器和移相电路，把检振元件的信号放大。经过移相，施加到驱动元件上去，构成闭环振荡器。在这个闭环中，既有机械能也有电能，叉体是其中的一个环节，倘若受到物料阻尼难以振动，正反馈的幅值和相位都

将明显的改变，破坏了振荡条件，就会停振。只要在放大电路的输出端接以适当的器件，不难得到开关信号。

为了保护压电元件免受物料损坏和粉尘污染，将驱动和检振元件装在叉体内部，经过金属膜片传递振动。如果在容器的上下方都装叉体，可以实现自动进料或自动出料的逻辑控制，或者把料位超限信号远传到控制室。在控制室里的控制电路判断料位是否超限，并按要求使被控的进出料设备启停。

并且叉体的制造和装配良好时，音叉也可用于液体测量和控制。在测量时不需要大幅度的机械运动，驱动功率小，机械结构简单、灵敏而可靠。

系统采用LD-YC/YG 型 音叉式物位限位开关



LD-YC/YG型音叉式物位限位开关作为一种新型的物位传感器，具有安装简单，不用标定，搅动、气泡对测量无影响等优点。在未来工业过程中将全面取代如浮球液位开关等落后的检测方式广泛应用于各行各业。

原理

传感器安装于仓的顶部或仓的侧壁上。音叉由压电晶体驱动产生振动，当叉体被液体浸没或被物料埋没时振动频率发生变化。这个变化由电子电路检出并输出一个开关量用于报警或控制。

适用范围

中等粘度、腐蚀性液体。流动性好的粉状物料和颗粒状物料。

禁用范围

高温、高粘度液体。高温、易结垢固体、块状物料。

主要技术指标

[YC 部分]

供电电压：直流 24V 或 10-55V (防爆型用 DC 24V)

交流 25-250V 50Hz

工作温度：叉体 -40-80℃ 仪表 -20-45℃

输出方式：

直流24V 供电时为继电器输出(AC 220V 1A)

直流10-55V 供电时为集电极开路输出 (350mA

防爆型产品≤50mA)

交流25-250V 供电时为两线制输出(220V

4.4W-77W)

功耗：直流供电时 0.25W(DC 24V)

交流供电时 1.5W(AC 220V)

安装方式：G1 管螺纹

防爆标志：本安 ia II BT4 隔爆型 d II BT4

[YG 部分]

供电电压：直流 24V 或 18-55V

交流50-250V 50Hz

工作温度：叉体 -40-150℃ 仪表 -20-70℃

输出方式：

直流24V 供电时为继电器输出(AC 220V 1A)

直流18-55V 供电时为集电极开路输出

(350mA)

交流50-250V 供电时为两线输出(220V

350mA 防爆型产品≤50mA)

功耗：直流供电时 0.25W(DC 24V)

交流供电时 1.5W(AC 220V)

安装方式：G1.5 管螺纹

第四章 系统的抗干扰及可靠性

单片机系统供电线路是干扰的主要来源，电源采用隔离变压器接入电网，隔离变压器在初级和次级之间加了一层屏蔽层。交流接触器是继电器控制的，在紧靠交流接触器的输入端口并上一个电阻和电容串联的电路滤去高次谐波。每个集成电路芯片的电源和地之间要并上 $0.1\ \mu\text{F}$ 的陶瓷电容。在输入输出通道上要加光藕 P521。除了在继电器通道上外，两个满料信号也要经过 P521 光藕才送至 INT0、INT1。另外，数字地和模拟地要分开，数字地线要粗而短。

软件抗干扰方面，本系统采用指令冗余和软件陷阱。当 CPU 受到干扰后，会把一些操作数当作指令码来执行，引起程序混乱。由于当 PC 指针飞到单字节指令上时，会自动步入正轨。因此我们在程序中人为地在 RET、RET1、LCALL、LJMP、AJMP、DJZE、JZ、CJNE 等处插入一些单字节指令（NOP）。并在继电器开闭指令 SET、CLR 前插入两条 NOP 指令。同时，当程序飞到 ROM 中未使用的非程序区时，非程序区要能捕获 PC 指针并强行拉到处理故障程序中。所以非程序区要每隔一段设置一个软件陷阱。软件陷阱

由三条指令构成：

NOP

NOP

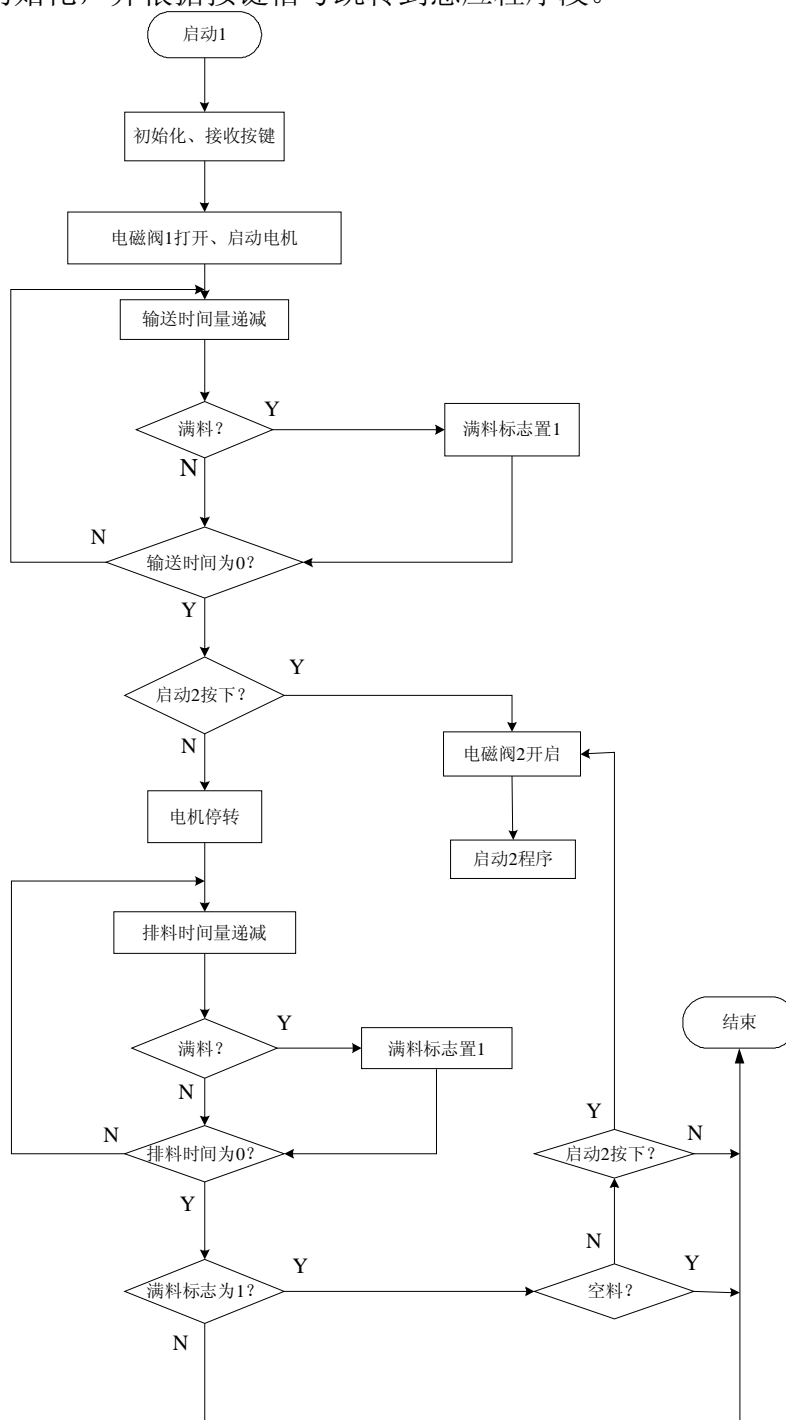
LJMP ERROR

ERROR 处理过程放在 0030H

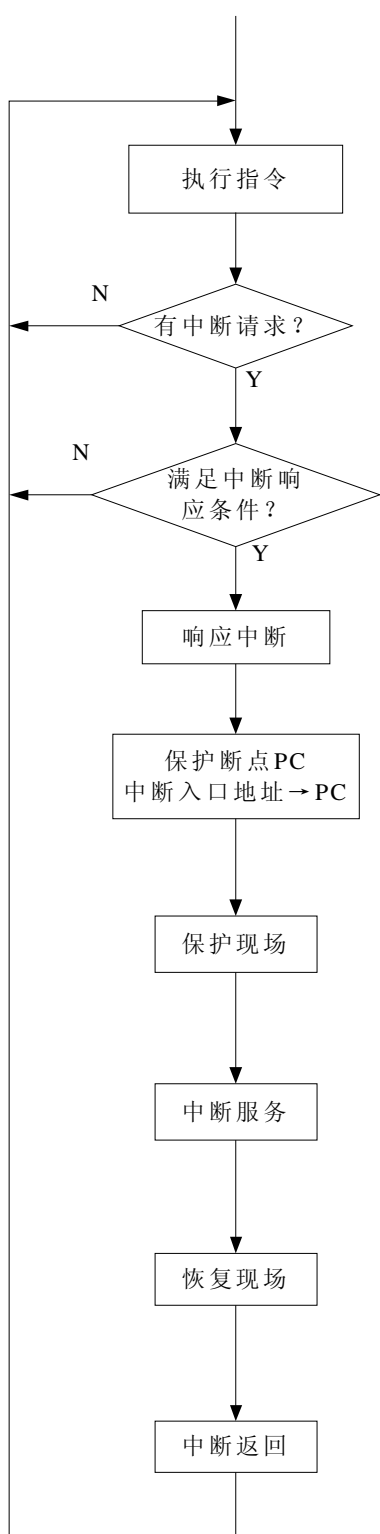
开始处

第五章 软件设计

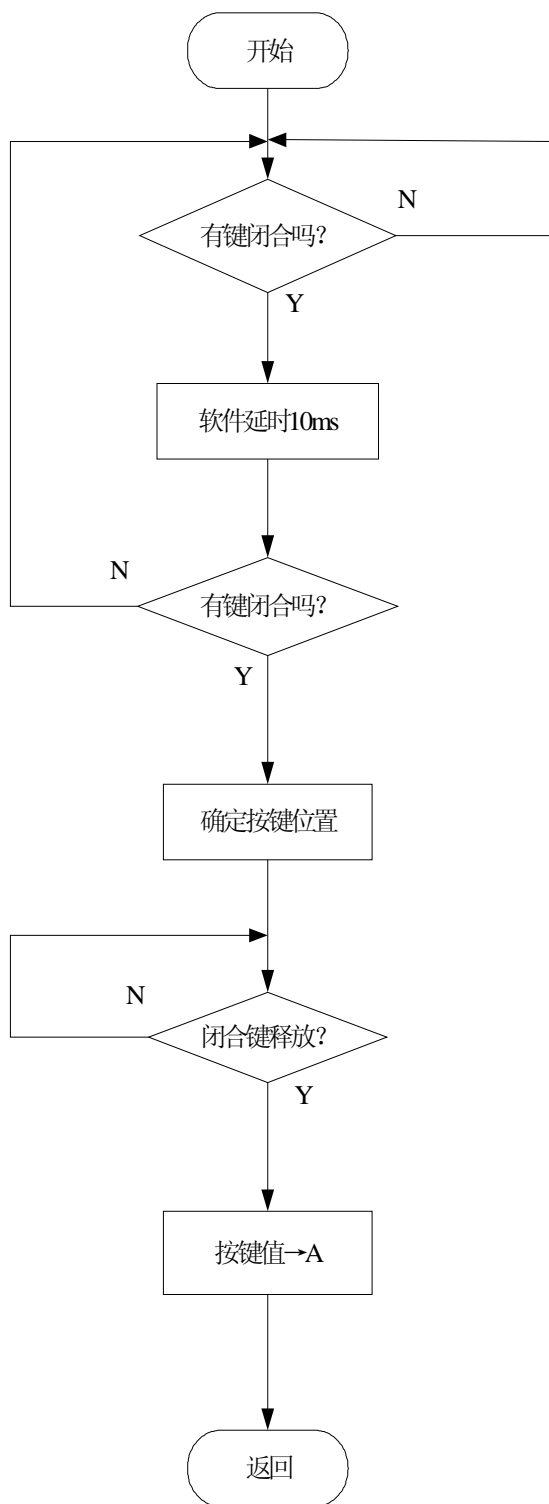
程序设计包括主程序、中断程序、故障处理程序，系统中用到了两个定时器中断：定时器 0 每隔 1.2s 使 P1.7 产生脉冲信号；定时器 1 产生 1s 的定时，使 LED 显示递减。故障处理程序使 PC 指针跳回到原来起动的相应程序段；主程序完成内部 RAM 数据区初始化、8255 初始化，并根据按键信号跳转到想应程序段。



主程序流程图



中断处理流程图



编程扫描
程序框图

第六章 结论语

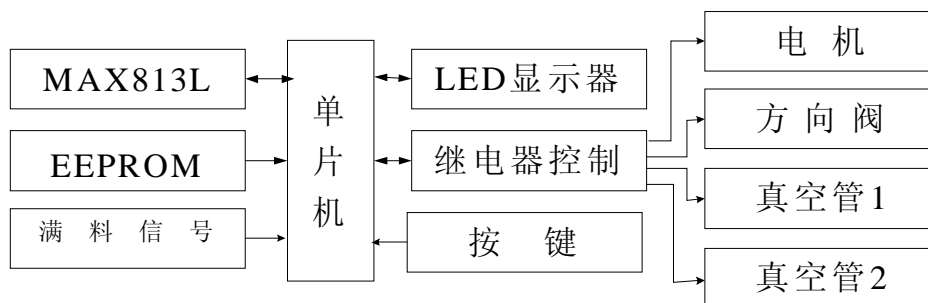
本设计是基于 AT89C51 的自动加料系统没有外扩 ROM 和 RAM，程序直接放在 AT89C51 内部闪存中。设定数据通过串行口存入 2864A 中，另外，系统从硬件和软件方面采取了抗干扰措施。

但是在物体送到料斗式干燥机后没有设计如何把物体从干燥机排除来，如果考虑到这一点，那设计就更加完美了。

参考文献:

1. 张毅刚、彭喜元,《MCS—51 单片机应用技术》,哈尔滨工业大学出版社,2003.7
2. 何立民,《MCS-51 单片机应用系统设计》,北京航空航天大学出版社,1998.7
3. 徐爱军,《单片机高级语言 C51 应用程序设计》,电子工业出版社,1997.7
4. 李华,《MCS—51 系列单片机实用接口技术》,北京航空航天大学出版社,1993.3
5. 王福瑞,《单片微机测控系统设计大全》,北京航空航天大学出版社,2000.5
6. 于微波、林晓梅、刘俊萍,《微型计算机控制系统》,吉林人民出版社,2002.5
7. 陈汝全、林水生、夏利,《实用微机与单片机控制技术》,电子科技大学出版社.1998.11
8. 武庆生等,《单片机原理与应用》,电子科技大学出版社,1998.2
9. 刘灿军,《实用传感器》,国防工业出版社,2004.6
10. 方佩敏,《新编传感器原理应用与电路详解》,电子工业出版社,1998.6
11. 赵负图,《传感器集成电路手册》,化学工业出版社 2002.4
12. 薛文达、谢文和、张呈详,《传感器应用技术》,东南大学出版社,1998.12
13. 张勇德,《过程控制装置》,化工出版社,2002.2
14. 杨绪东,《实用电子电路精选》,化工出版社,2001.5
15. 徐科军,《传感器与检测技术》,电子工业出版社,2004.5
16. 丁树模,《机械工业学》,机械工业出版社,2003.3
17. 陈兆宽,《计算机过程控制软件设计》,电子工业出版社,1993.12
18. 高锋编,《单片微机应用系统设计及使用技术》,机械工业出版社,2004..9
19. 付晓光,《单片机原理与实用技术》,清华大学出版社,2004.8
20. 刘勇等,《数字电路》,电子工业出版社,2003.1
21. 李仁,《电器控制》,机械工业出版社,2002.1
22. 马明建等,《数据采集与处理技术》,西安交通大学出版社,1998.12
23. 孙传友、孙晓斌,《测控系统原理与设计》,北京航空航天大学出版社,2002.9
24. 王仁详,《常用低压电器原理及其控制技术》,机械工业出版社,2001.8
25. 杨帮文,《新型继电器使用手册》,人民邮电出版社,2004.5
26. 周秉功,《继电器选型手册》,机械工业出版社,1998.8
27. 李刚,《现代测控电路》,高等教育出版社,2004.11
28. 谢宛清、李宗豪、朱金明,《信号处理电路》,电子工业出版社,1994.5
29. 浦昭邦等,《测控仪器设计》,机械工业出版社,2001.11
30. 贾瑞皋,薛庆忠,《电磁学》,高等教育出版社,2003.9

附录 A: 系统框图



自动加料机控制系统硬件框图

程序清单:

```

ORG
LJMP I1NTO
ORG 0080H
MAIN: MOV SP, #60H
MOV PSW, #00H
MOV 71H, #14H
MOV 72H, #
MOV 73H, #00H
MOV 74H, #00H
MOV 75H, #00H
MOV 76H, #00H
MOV TMOD, #01H
MOV TLO, #0B0H
MOV TH0, #3CH
SETB TR0                ; , 开中断
SETB EA
SETB ETO
JMCGS: JMB P1.2, KK
NOP
NOP
JB P1.2, KK
NOP
MOV 70H, #01H
KK: LJMP JMCGS          ; 等待中断
      ORG 0100H        ; 中断服务子程序
I1NTO: PUSH A          ; 入栈保护
PUSH PSW
CLR EA                  ; 关中断
CLR TR0
DJNZ 71H, ZDFH        ; 不到 2 秒直接中断返回
MOV A, 70H
MOV B, 72H
MVL AB
    
```

```

MOV R6, B
MOV R5, A
MOV R2, #02H
LCALL CHUFA
MOV 73H, R5
LCALL KZCX
MOV 74H, 73H
ZDFH: SETB EA                ; 开中断
SETB ET0
MOV TMOD, #01H              0 重新赋初值
MOV TLO, #0B0H
MOV TH0, #3CH
SETB TRO
POP PSW                      ; 出栈
POP A
RETI                          ; 中断返回
CHUFA: MOV R7, #08H
CHU1: CLR C
MOV A, R5
RLC A
MOV R5, A
MOV A, R6
RLC A
MOV 07H, C
CLR C
SUBB A, R2
JB 07H, CHU2
JNC CHU2
ADD A, R2
AJMP CHU3
CHU2: INC R5
CHU3: MOV R6, A
JNE R7, CHU1
RET
KZCX: MAIN A, 73H           ; 控制子程序

```

```

CJNE A, 74H, KZ2
ZJFH: RET ; 直接返回
KZ2: JC PFKZ
CFKZ: MOV B, #03H
DIV AB
MOV 75H, A
CJNE A, #04H, ZJFH
JC ZJFH
MOV A, 75H
CJNE A, #0AH, CFKZ1
JNC CFKZ2
CFKZ1: MOV TMOD, #10H
MOV TLO, #0B0H
MOV TH0, #3CH
SETB TR1
SETB P1.0
DELAY: JNB TF1, DELAY
CLR TR1
CLR TF1
CLR P1.0
LJMP ZJFH
CFKZ2: MOV R4, #02H
CFKZ3: MOV TMOD, #10H
MOV TH0, #3CH
SETB TR1
SETB P1.0
DELAY: JNB TF1, DELAY1
CLR TR1
CLR TF1
CLR P1.0
DJNZ R4, CFKZ3
LJMP ZJFH
PFKZ: MOV B, #03H
DIV AB
MOV 76H, A
    
```

```

CJNE A,#08H,ZJFH
JC ZJFH
MOV A,76H
CJNE A,#0AH,PFKZ1
JNC PFKZ2
PFKZ1:MOV TMOD,#10H
MOV TL0,#3CH
SETB TR1
SETB P1.1
DELAY:JNB TF1,8
CLR TR1
CLR TF1
CLR P1.1
LJMP ZJFH
PFKZ2:MOV R4,#02H
PFKZ3:MOV TMOD,#10H
MOV TL0,#0B0H
MOV TH0,#3CH
SETB TR1
SETB P1.1
DELAY4:JNB TF1,DELAY4
        CLR TR1
        CLR TF1
        CLR P1.1
        DJNZ R4,PFKZ3
        LJMP ZJFH
    
```

存储器扩展子程序

```

STORE:  SETB VSDA;
        SETB VSCL;
        NOP;
        NOP;
        NOP;
    
```



```
NOP;
CLR  VSDA;
NOP;
CLR  VSDA;
NOP;
NOP;
NOP;
NOP;
CLR  VSCL;
RET;
STOP: CLR  VSDA;
SETB VSCL;
NOP;
NOP;
NOP;
NOP;
SETB VSDA;
NOP;
NOP;
NOP;
NOP;
CLR   VSDA;
CLR   VSCL;
RET;
MACK: CLR  VSDA;
SETB  VSCL;
NOP;
NOP;
NOP;
NOP;
CLR   VSCL;
SETB  VSDA;
RET;
MNACK: SETB VSDA;
SETB  VSCL;
```

```

NOP;
NOP;
NOP;
NOP;
CLR VSCL;
CLR VSDA;
RET;
CACK:  SETB VSDA;
SETB VSCL;
CLR FO;
MOV C, VSDA;
JNC CEND;
SETB FO;
CEND: CLR VSCL;
RET;
WRBYT: MOV RO, #08H;
WLP:  RLC A;
JC WR1;
AJMP WRO;
WLP1: DJNZ RO, WLP;
RET;
WR1: SETB VSDA;
SETB VSCL;
NOP;
NOP;
NOP;
NOP;
CLR VSCL;
CLR VSDA;
AJMP WLP1;
WRO: CLR VSDA;
SETB VSCL;
NOP;
NOP;
NOP;

```

```

NOP;
CLR VSCL;
AJMP WLP1;
RDBYT: MOV RO, #08H;
RLP: SETB VSDA;
SETB VSCL;
MOV C, VSDA;
MOV A, R2;
RLC A;
MOV R2, A;
CLR VSCL;
DJNZ RO, RLP;
RET;
WRNBYT: MOV R3, NUMBYT;
LCALL STA;
MOV A, SLA;
LCALL WRBYT;
LCALL CACK;
JB FO, WRNBYT;
MOV R1, #MTD;
WRDA: MOV A, @R1;
LCALL WRBYT;
LCALL CACK;
JB FO, WRNBYT;
INC R1;
DJNZ R3, WRDA;
LCALL STOP;
RET;
RDNBYT: MOV R3, NUMBYT;
LCALL STA;
MOV A, SLA;
LCALL WRBYT;
LCALL CACK;
JB FO, RDNBYT;
RDN: MOV R1, #MRD;
    
```

```

RDN1: LCALL RDBYT;
MOV @R1,A;
DJNZ R3,ACK;
LCALL MNACK;
LCALLSTOP;
RET;
ACK: LCALL MACK;
INC R1;
SJMP RDN1;
    
```

中断服务子程序:

```

XOS:   MOV DPTR,#2000H
        MOVX @DPTR,A
        SETB TRO
        MOV 3AH,#20H
        MOV 3BH,#00H
        MOV R7,#00H
        MOV R6,#04H
        CLR 70H
XOS0:  CLR01H
XOS1:  JNB 01H,XOS1
        CJNE R6,#00H,XOS0
        CJNE R7,#00H,XOS0
        CLR TRO
        CLR EX0
        SETB 00H
        RETI
    
```

键盘程序:

```

        KEYI: ACALL  KS1           ; 调用判有无键闭合子程序
        JNZ     LK1             ; 有键闭合,跳LK!
        NI:    ACALL  DIR         ; 无键闭合,调用显示子程序,延时
        AJMP   KEYI
    
```

```

LK1:  ACALL  DIR      ; 可能有键闭合, 延时 12MS 软件区
      ACALL  DIR
      ACALL  KS1      ; 调用判有无子程序
      JNZ   LK2      ; 经去消抖, 判断确实有键按下跳
      ACALL  DIR      ; 调用显示子程序延时 6MS
      AJMP  KEY1     ; 抖动引起跳 KEY1 去处理
LK2:  MOV   R2, #0FEH ; 列选码—> R2
      MOV   R4, #00H  ; R4 为列号计数器
LK4:  MOV   DPTR, #7F01H ; 列选码—> 8155H
      MOV   A, R2
      MOVX  @DPTR, A
      INC   DPTR      ; 数据指针增 2, 指向 PC 口
      INC   DPTR
      MOVX  A, @DPTR ; 读 8155H 的 PC 口
      JB   Acc. 0, LONE ; 第 0 行线为高, 无键闭合, 跳 LONE
      ; 转判第 1 行
      MOV   A, #00H   ; 第 0 行有键闭合, 首键号—> A
      AJMP  LKP       ; 跳 LKP, 计算键号
LONE: JB   Acc. 1, LTOW ; 1 行线为高, 无键闭合, 跳 LTWO
      ; 转判断第 2 行
      MOV   A, #08H   ; 1 行有键闭合, 首键号 8—> A
      AJMP  LKP       ; 跳 LKP , 计算键号
LTWO: JB   Acc. 2, LTHR ; 2 行线为高, 无键闭合跳 LTHR
      ; 转判; 第 3 行
      MOV   A, #10H   ; 2 行有键闭合, 首键号 10H—> A
      AJMP  LKP       ; 跳 LKP, 计算键号
LTHR: JB   Acc. 3, NEXT ; 3 行线为高, 无键闭合跳 NEXT,
      ; 准备下一列的扫描
      MOV   A, #18H   ; 3 行有键按下, 首键号 18H—>A
      LKP:  ADD  A, R4 ; 计算键号
      PUSH  A         ; 键号进栈保护
LK3:  ACALL  DIR      ; 调用显示子程序, 延时 6MS
      ACALL  DIR      ; 调用判有无键闭合子程序, 延时 6MS
      JNZ  LK3      ; 判键释放否, 未释放, 则循环
      POP  A         ; 键已释放, 键号出栈->A
    
```

```

RET
NEXT: INC R4 ;列计数器加 1, 为下一列扫描做准备
MOV A, R2 ;判是否已扫到最后一列(最右一列)
NB Acc. 7, KND ;键扫描已到最后一列, 跳 KND 重新
; 进行整个
; 键盘的扫描
RL A ;键扫描未扫到最后一列, 位选码左
; 移一位
MOV R2, A ;位选码->R2
AJMP LK4
KND: AJMP KEY1
KS1: MOV DPTR, #7F01H ;判有无键闭合子程序, 全 0->扫描(PA 口)
MOV A, #00H ;列线全为低电平
MOVX @DPTR, A ;
INC DPTR ; DPTR 增 2, 指向 PC 口
INC DPTR
MOVX A, @DPTR ; 从 PC 口读行线的状态
CPL A ;行线的状态取反, 如无键按下, 则 A
; 中内容为 0
ANL A, #0FH ; 屏蔽无用的高 4 位
RET

PID: MOV R5, 31H ;取 W
MOV R4, 32H
MOV R3, #00H ;取 Yi
MOV R2, 2AH
ACALL CPL1 ;取 Yi 的补码
ACALL DSUM ;计算 ei =W-Yi
MOV 39H, R7 ;存 ei
MOV 3AH, R6
MOV R5, 35H ;取 I
MOV R4, 36H
MOV R0, #4AH ;R0 放乘积高位字节地址指针
ACALL MULT1 ;计算 Pi =I *ei
MOV R5, 39H ;取 ei
    
```

```

MOV    R4, 3AH
MOV    R3, 3BH                ;取 ei-1
MOV    R2, 3CH
ACALL  CPL1                  ;求 ei-1: 的补码
ACALL  DSUB                   ;求 pp=Δei =ei -ei-1:
MOV    A, R7
MOV    R5, A                  :存Δei
MOV    A, R6
MOV    R4, A
MOV    R3, 4BH                ;取 Pi
MOV    R2, 4AH
ACALL  DSUM                   ;求 Pi +Pp
MOV    R3, 4BH, R7
MOV    R4, 4AH, R6           ;存和(Pi +即)
MOV    R5, 39H
MOV    R4, 3AH                ;取 ei
MOV    R3, 3DH
MOV    R2, 3EH                ;取 ei-2
ACALL  DSUM                   :计算 ei=ei-2
MOV    A, R7
MOV    R5, A
MOV    A, R6
MOV    R4, A
MOV    R3, 3BH
MOV    R2, 3CH                ;取 ei-1
ACALL  CPL1                  :求 ei-1, 补码
ACALL  DSUM                   ;计算 ei=ei-2-ei-1
MOV    A, R7
MOV    R5, A                  ;存和
MOV    A, R6
MOV    R4, A
MOV    R3, 3BH
MOV    R3, 3CH                ;取 ei-1
ACALL  CPL1                  :求 ej-1 补码
ACALL  DSUM                   ;计算 ei=ei-2-2ei-1、

```



```

MOV    R5, 37H           ;取 D
MOV    R4, 38H
MOV    R0, #46H
ACALL  MULT1             ; 求出  $Pd=D*(e_i=e_{i-2}-2e_{i-1})$ 
MOV    R5, 47H
MOV    R4, 46H           : 存 Pd
MOV    R3, 4BH
MOV    R2, 4AH           ; 取(pi +PP)
ACALL  DSUM             ; 计算 pi +pp+pd
MOV    R5, 33H           ; 取 Kp
MOV    R4, 34H
MOV    R0, #46H
ACALL  MULTI            ; 计算  $Kp*(Pi +Pp+Pd)$ 
MOV    R3, 47H
MOV    R2, 46H
MOV    R5, 2FH           ; 取  $u_{i-1}$ 
MOV    R4, 30H
ACALL  DSUM             ; 求出  $u_i = u_{i-1} + Kp*(Pi +Pp +pd)$ 
MOV    2FH, R7           ; 存  $u_i \sim u_{i-1}$ 
MOV    30H, R6
MOV    3DH, 3BH         ;  $e_{i-1} - e_{i-2}$ 
MOV    3EH, 3CH
MOV    3BH, 39H         :  $e_i - e_{i-1}$ 
MOV    3CH, 3AH
RET
DSUM:  MOV    A, R4
ADD    A, R2
MOV    R6, A
MOV    A, R5
ADDC   A, R3
MOV    R7, A
RET
CPLI:  MOV    A, R2
CPL    A
ADD    A, #01H
    
```

```

MOV    R2, A
MOV    A, R3
CPL    A
ADDC   A, #00H
MOV    R3, A
RET
MULT1: MOV    A, R7
RLC    A
MOV    SIGN1, C           ; 存被乘数符号位
JNC    POS1              ; 被乘数为正转
MOV    A, R6              ; 求补
CPL    A
ADD    A, #01H
MOV    R6, A
MOV    A, R7
CPL    A
ADDC   A, #00H
MOV    R7, A
POS1:  MOV    A, R5
RLC    A
MOV    SIGN2, C           ; 存乘数符号位
JNC    POS2              ; 乘数为正转
MOV    A, R4
CPL    A
ADD    A, #01H
MOV    R4, A
MOV    A, R5
CPL    A
ADDC   A, #00H
MOV    R5, A
POSZ:  ACALL  MUL1,
MOV    C, SIGN1
ANL    C, SIGN2
JC     TPL                ; C=1 是两个负数相乘转
MOV    C, SIGN1
    
```

```

ORL    C, SIGN2
JNC    TPL                    ;C=0 是两个正数相乘转
DEC    R0
DEC    R0
DEC    R0
MOV    A, @R0
CPL    A
ADD    A, #01H
MOV    @R0, A
INC    R0
MOV    A, @R0
CPL    A
ADDC   A, #00H
TPL:   RET
MULT:  MOV    A, R6            ;取 b
MOV    B, R4                  ;取 d
MUL    AB                    ;求 bd
MOV    @R0, A                 ;求 bdL
MOV    R3, B                  ;求 bdL~R3
MOV    A, R4                  ;取 d
MOV    B, R7                  ;取 a
MUL    AB                    ;求 ad
ADD    A, R3                  ;求 adL+bdH
MOV    R3, A                  ;暂存 adL+bdH
MOV    A, B
ADDC   A, #00H                ;adH+进位
MOV    R2, A                  ;暂存
MOV    A, R6                  ;取 b
MOV    B, R5                  ;取 c
MUL    AB                    ;求 bc
ADD    A, R3                  ;求 adL+bcL+bdH
INC    R0                    ;修改指针
MOV    @R0, A                 ;乘积第三字节存入
CLR    55BH                   ;清标志
MOV    A, R2

```

```

ADDC  A, B                ; 求 adH+bCH
MOV   R2, A
JNC   LAST                ; 无进位数
SETB  5BH                ; 置进位标志
MOV   A, R7               ; 取 a
MOV   B, R5               ; 取 c
MUL   AB                  ; 求 ac
ADD   A, R2               ; 求 acL+adH+bCH
INC   R0
MOV   @R0, A              ; 存积的第二个字节
MOV   A, B
ADDC  A, #00H
MOV   C, 5BH
ADDC  A, #00H
INC   R0                  : 修改指针
MOV   @R0, A              ; 存积的最高字节
RET
    
```

显示子程序:

```

ORG  055CH
MOV  R0, #79H
MOV  @R0, #01H
INC  R0
MOV  @R0, #01H
INC  R0
MOV  R0, #05H
INC  R0
MOV  @R0, #16H
INC  R0
MOV  @R0, #1BH
INC  R0
MOV  @R0, #1EH
MOV  DPTR, #4F00H
MOV  A, #0C3H
MOVX @DPTR, A
    
```

```

AJMP  DISUP
ORG  0772H
MOV  R0, #79H
MOV  R3, #01H
MOV  A, R3
MOV  DPTR, #4F01H
MOVX @DPTR, A
INC  DPTR
MOV  A, @R0
ADD  A, #12H
MOVC A, @A+PC
JNB  F0, DISUP2
SETB a, 7
MOVX @DPTR, A
ACALL D2MS
INC  R0
MOV  A, R3
JB  A, 5, DISUP3
RL  A
MOV  R3, A
AJMP DISUP1
ORG  0792H
TABLE: DB  3FH
DB  06H
DB  5BH
DB  4FH
DB  66H
DB  6DH
DB  7DH
DB  07H
DB  7FH
DB  6FH
DB  77H
DB  7CH
DB  39H
    
```

```
DB 5EH
DB 79H
DB 71H
DB 73H
DB 3EH
DB 3EH
DB 31H
DB 48H
DB 1CH
DB 23H
DB 40H
DB 03H
DB 18H
DB 80H
DB 00H
DB 3DH
DB 76H
DB 38H
DB 1EH
ORG 07B1H
D2MS :MOV R7,#02H
D2MS1:MOV R6,#OFFH
DJNZ R6,$
DJNZ R7,DSMS1
```