
摘要

本设计是一 16×16 点阵 LED 电子显示屏的设计。

整机以美国 ATMEL 公司生产的 40 脚单片机 AT89C51 为核心，介绍了以它为控制系统的 LED 点阵电子显示屏的动态设计和开发过程。通过该芯片控制一个行驱动器 74HC154 和两个列驱动器 74HC595 来驱动显示屏显示。该电子显示屏可以显示各种文字或单色图像，全屏能显示 1 个汉字，采用 4 块 8×8 点阵 LED 显示模块来组成 16×16 点阵显示模式。显示采用动态显示，使得图形或文字能够实现静止、移入移出等多种显示方式。文中详细介绍了 LED 点阵显示的硬件设计思路、硬件电路各个部分的功能及原理、相应软件的程序设计，以及使用说明等。

单片机控制系统程序采用单片机汇编语言进行编辑，通过编程控制各显示点对应 LED 阳极和阴极端电平，就可以有效的控制各显示点的亮灭。所显示字符的点阵数据可以自行编写（即直接点阵画图），也可从标准字库中提取。

LED 显示以其组构方式灵活、显示稳定、功耗低、寿命长、技术成熟、成本低廉等特点在车站、证券所、运动场馆、交通干道及各种室内/外显示场合的信息发布，公益宣传，环境参数实时，重大活动倒计时等等得到广泛的应用。

经实践证明，该系统显示误差小，性能稳定，结构合理，扩展能力强。

关键词：AT89C51 单片机； LED； 点阵显示； 动态显示； 汇编语言。

Abstract

This design is a 16×16 lattice LED electron display monitor design.

The whole equipment is with the 40-pin AT89C51 MCU (Micro Controller Unit) produced by the American ATMEL company at the core, introduced take it as the control system LED lattice electron display monitor dynamic design and the development process. Controls good driver 74HC154 and two row driver 74HC595 through this chip actuates the display monitor demonstration. The electronic screen can show all kinds of written or monochrome images, one full screen display Chinese characters, four pieces of 8×8 dot-matrix LED display modules to form the 16×16 dot matrix display mode. Show dynamic show that makes static graphic or text can be achieved, shifted out of various formats. This paper describes the hardware design of the LED dot matrix display, and the principle function of the various parts of the circuit, the corresponding software program design and the use of some such.

SCM process control system used for editing MCU assembly language, Programming control points indicated by the corresponding LED anode and overcast extreme level. We can effectively control the defense showed bright spots. The lattice data shows characters can prepare themselves (that is, direct lattice Painting), which can also be extracted from the standard font.

LED display with fabric means flexibility, stability, low power consumption, long life, mature technology, low-cost features at the station, securities, sports venues, transportation corridors and various indoor / dissemination of information on foreign shows occasions, good publicity, real-time environmental parameters, etc. countdown major activities are widely used.

As the practice proves, the system possesses advantages in low shows errors, stable, rational structure and strong extensible abilities.

Key words: AT89C51 Micro Controller Unit; LED; Lattice display; Dynamic display; Assembly language.

目录

摘要.....	I
ABSTRACT.....	II
第一章 前言.....	1
第二章 系统整体设计方案.....	2
2.1 需要实现的功能.....	2
2.2 LED 显示特点.....	2
2.3 设计方案论证.....	3
2.3.1 显示模式方案.....	3
2.3.2 数据传输方案.....	4
第三章 系统硬件部分设计.....	5
3.1 电源设计.....	5
3.2 单片机系统及外围电路.....	6
3.2.1 单片机的选择.....	6
3.2.2 AT89C51 芯片介绍.....	6
3.2.3 单片机系统外围电路.....	9
3.3 列驱动电路.....	10
3.4 行驱动电路.....	12
3.4.1 行驱动芯片 74HC154 介绍.....	12
3.4.2 行驱动电路.....	13
3.5 LED 显示屏电路.....	15
第四章 系统软件部分设计.....	17
4.1 系统主程序.....	17

4. 2 显示驱动程序.....	18
第五章 调试及性能分析.....	20
5. 1 软件调试.....	20
5. 2 硬件调试.....	20
5. 3 性能分析.....	21
结束语.....	22
致谢.....	23
附录一 系统综合电路原理图.....	24
附录二 系统程序清单.....	25
主要参考文献.....	32

第一章 前言

单片机自 20 世纪 70 年代问世以来，以极其高的性能价格比受到人们的重视和关注，所以应用很广，发展很快。单片机的特点是体积小、集成度高、重量轻、抗干扰能力强，对环境要求不高，价格低廉，可靠性高，灵活性好，开发较为容易。正因为单片机有如此多的优点，因此其应用领域之广，几乎到了无孔不入的地步。在我国，单片机已被广泛地应用在工业自动化控制、自动检测、智能仪表、智能化家用电器、航空航天系统和国防军事、尖端武器等各个方面。我们可以开发利用单片机系统以获得很高的经济效益。更重要的意义是单片机的应用改变了控制系统传统的设计思想和方法。以前采用硬件电路实现的大部分控制功能，正在用单片机通过软件方法来实现。这种以软件结合硬件或取代硬件并能提高系统性能的控制技术称为微控制技术。例如，本文所要论述的通过单片机来控制 LED 点阵显示。

LED 是发光二极管英文 Light Emitting Diode 的简称，是六十年代末发展起来的一种半导体显示器件，七十年代，随着半导体材料合成技术、单晶制造技术和 P-N 结形成技术的研究进展，发光二极管在发光颜色、亮度等性能得以提高并迅速进入批量化和实用化。进入八十年代后，LED 在发光波长范围和性能方面大大提高，并开始形成平板显示产品即 LED 显示屏。

LED 电子显示屏是利用发光二极管点阵模块或像素单元组成的平面式显示屏幕。它是集微电子技术、光电子技术、计算机技术、信息处理技术于一体的显示系统，是目前国际上极为先进的显示媒体。由于它具有发光效率高、使用寿命长、组态灵活、色彩丰富、工作性能稳定以及对室内室外环境适应能力强等优点而日渐成为显示媒体中的佼佼者。在我国改革开放之后，特别是进入 90 年代国民经济高速增长，对公众场合发布信息的需求日益强烈，LED 显示屏的出现正好适应了这一市场形势，因而在 LED 显示屏的设计制造技术与应用水平上都得到了迅速的提高，生产也得到了迅速的发展，并逐步形成产业，成为光电子行业的新兴产业领域。LED 显示屏经历了从单色、双色图文显示屏，到图像显示屏的发展过程。

随着信息产业的高速发展，LED 显示屏作为信息传播的一种重要手段成为现代信息化社会的一个闪亮标志。近年 LED 显示屏已广泛应用于室内、外需要进行服务内容和服务宗旨宣传的公众场所如银行、营业部、车站、机场、港口、体育场馆等信息的发布，政府机关政策、政令，各类市场行情信息的发部和宣传等。目前，对于那些需要显示的信息量不是很大，分辨率不是很高，又需要制造成本相对较低的场合，使用大、小屏幕 LED 点阵显示器是比较经济适用的，它可以用单片机控制实现显示字符、数字、汉字和简单图形，可以根据需要使用不同字号、字型。

汉字显示方式是先根据所需要的汉字提取汉字点阵（如 16×16 点阵），将点阵文件存入 ROM，形成新的汉字编码。而在使用时则需要先根据新的汉字编码组成语言，再由 MCU 根据新编码提取相应的点阵进行汉字显示。具体显示技术和原理将会在正文中得到详细论述。

杜春雷
2007 年 4 月

第二章 系统整体设计方案

2.1 需要实现的功能

设计一个室内用 16×16 点阵 LED 图文显示屏，要求在目测条件下 LED 显示屏各点亮度均匀、充足，可显示图形和文字，显示图形或文字应稳定、清晰无串扰。图形或文字显示有静止、移入移出等显示方式。

2.2 LED 显示特点

LED 发光灯可以分为单色发光灯、双色发光灯、三色发光灯、面发光灯、闪烁发光灯、电压型发光灯等多种类型。按照发光灯强度又可以分为普通亮度发光灯、高亮度发光灯、超高亮度发光灯等。这种单个的发光灯适宜用做指示灯，如电源指示、电路状态指示灯，进而对能够转变成电信号的各种物理量进行指示。也可以用多个 LED 发光灯组成固定的字符或图形进行显示，如大型剧场会堂的出入口及洗手间的显示。和很多应用术语一样，LED 图文显示屏并没有一个公认的严格的定义，一般把显示图形和/或文字的 LED 显示屏称为图文屏。这里所说的图形，是指由单一亮度线条组成的任意图形，以便于不同亮度（灰度）点阵组成的图像相区别。图文显示屏的主要特征是只控制 LED 点阵中各发光器件的通断（发光或熄灭），而不控制 LED 的发光强弱。LED 图文显示屏的外观可以做成条形，叫做条形图文显示屏（简称条屏），也可以按一定高度比例做成矩形的平面图文显示屏。其实条屏只不过是其宽度远大于高度的平面显示屏，在显示与控制的原理上并无区别。

从理论上说，不论显示图形还是文字，都是控制与组成这些图形或文字的各个点所在位置相对应的 LED 器件发光。通常事先把需要显示的图形文字转换成点阵图形，再按照显示控制的要求以一定的格式形成显示数据。对于只控制通断的图文显示屏来说，每个 LED 发光器件占据数据中的 1 位（1bit），在需要该 LED 器件发光的数据中相应的位填 1，否则填 0。当然，根据控制电路的安排，相反的定义同样是可行的。这样依照所需显示的图形文字，按显示屏的各行各列逐点填写显示数据，就可以构成一个显示数据文件。显示图形的数据文件，其格式相对自由，只要能够满足显示控制的要求即可。文字的点阵格式比较规范，可以采用现行计算机通用的字库字模。组成一个字的点阵，其大小也可以有 16×16 、 24×24 、 32×32 、 48×48 等不同规格。汉字的点阵结构相应的显示数据是用 16 进制格式以字节为单位表示的。

用点阵方式构成图形或文字，是非常灵活的，可以根据需要任意组合和变化，只要设计好合适的数据文件，就可以得到满意的显示效果。因而采用点阵式图文显示屏显示经常需要变化的信息，是非常有效的。

点阵显示方式适应信息变化的优点，是以点阵显示器的价格和其复杂的控制电路为代价的。点阵显示器在整个显示单元的所有位置上都布置了 LED 器件，而像数码管一类的显示器件只在需要发光的七段位置上布置 LED 器件，其它位置是空白的。因此，点阵显示器在相同面积情况下，价格要贵些。但是，数码管可显示的信息有限，只有 0~9 或单个的英文字符，由于组合形成的字符不多，所以其显示数据和控

制电路都比较简单。点阵显示器则不然，它要对点阵上全部 LED 进行控制，并能生成所有可能显示的图形文字，其显示数据和控制电路自然要复杂得多。因此，根据不同的场合，不同的显示特点和格式可以分别采用数码管器件或点阵显示方式，有的情况可以采用两者搭配的混合屏显示方式。

图文显示屏的颜色，有单色、双色、和多色几种。最常用的是单色图文屏。单色屏多使用红色、橘红色或橙色 LED 点阵单元。双色图文屏和多色图文屏，在 LED 点阵的每一个“点”上布置有两个或多个不同颜色的 LED 发光器件。换句话说，对应于每种颜色都有自己的显示矩阵。显示的时候，各颜色的显示点阵是分开控制的。事先设计好各种颜色的显示数据，显示时分别送到各自的显示点阵，即可实现预期效果。每一种颜色的控制方法和单色的完全相同，因此掌握了单色图文显示屏的原理，双色屏和多色屏就不难理解了。

2. 3 设计方案论证

2. 3. 1 显示模式方案

为了吸引观众增强显示效果，可以有多种显示模式，最简单的显示模式是静态显示。这里所说的“静态显示模式”不同于静态驱动方式。与静态显示模式相对应，就有各种动态显示模式，它们所显示的图文都是能够动的。按照图文运动的特点又可以分为闪烁、平移、旋转、缩放等多种显示模式。产生不同显示模式的方法，并不意味着一定要重新编写显示数据，可以通过一定的算法从原来的显示数据直接生成。例如，按顺序调整行号，可以使显示图文产生上下平移；而顺序调整列显示数据的位置，就可以达到左右平移的目的；同时调整行列顺序，就能得到对角线平移的效果。其它模式的数据刷新，也可找到相应的算法。不过当算法太复杂，太浪费时间的话，也可以考虑预先生成刷新数据，存储备用。刷新的时间控制，要考虑运动图形文字的显示效果。刷新太慢，动感不显著；刷新太快了，中间过程看不清。一般刷新周期可控制在几十毫秒范围之内。

同时控制 LED 显示屏的各个发光点亮灭的方法称为静态驱动显示方式。16×16 的点阵共有 256 个发光二极管，显然单片机没有这么多端口，如果我们采用锁存器来扩展端口，按 8 位的锁存器来计算，16×16 的点阵需要 $256/8=32$ 个锁存器。这个数字很庞大，因为我们仅仅是 16×16 的点阵，在实际应用中的显示屏往往要大得多，这样在锁存器上花的成本将是一个很庞大的数字。而且静态显示功耗大，显示一个 16×16 的字的功耗为： $16 \times 16 \times 10 \times 5 = 12800 \text{mW} = 12.8 \text{W}$ ！普通电池根本无法驱动，要是组成大显示屏那功耗将更惊人。因此虽然静态显示效果好，但是在实际应用中的显示屏几乎都不采用这种设计，而采用另一种称为动态扫描的显示方法。

动态扫描的意思简单地说就是逐行轮流点亮，这样扫描驱动电路就可以实现多行（比如 16 行）的同名列共用一套列驱动器，每行有一个行驱动器，具体就 16×16 的点阵来说，把所有同一行的发光管的阳极连在一起，把所有同一列的发光管的阴极连在一起（共阳的接法），由行译码器给出的行选通信号，从第一行开始，按顺序依次对各行进行扫描（把该行与电源的一端接通）。另一方面，根据各列锁存的数据，确定相应的列驱动器是否将该列与电源的另一端接通。接通的列，就在该行该列燃

亮相应的 LED；未接通的列所对应的 LED 熄灭。当一行的扫描持续时间结束后，该行燃亮的 LED 也就熄灭；下一行又以同样的方法进行显示。全部各行都扫过一遍之后(一个扫描周期)，又从第一行开始下一个周期的扫描。只要一个扫描轮回的速度足够快（每秒 24 次以上），由于人眼的视觉暂留现象，就不容易感觉出闪烁现象，就能看到显示屏上稳定的图形了。而且动态扫描方式功耗低，硬件成本低，每个 LED 都不是连续工作，因而还有利于延长 LED 的使用寿命。

2. 3. 2 数据传输方案

采用扫描方式进行显示时，显示数据通常存储在单片机的存储器中，按 8 位一个字节的顺序排放。显示时要把一行中各列的数据都传送到相应的列驱动器上去，这就存在一个显示数据传输方式的问题。从控制电路到列驱动器的数据传输可以采用并行方式或串行方式。显然，采用并行方式时，从控制电路到列驱动器的线路数量大，相应的硬件数目多。当列数很多时，并行传输的方案是不可取的。

采用串行传输的方法，控制电路可以只用一根信号线，将列数据一位一位传往列驱动器，在硬件方面无疑是十分经济的。但是，串行传输过程较长，数据要经过并行到串行和串行到并行两次变换。首先，单片机从存储器中读出的 8 位并行数据要通过并串变换，按顺序一位一位地输出给列驱动器。与此同时，列驱动器中每一列都把当前数据传向下一列，并从前列接收新数据，一直到全部列数据都传输完为止。只有当一行的各列数据都已传输到位之后，这一行的各列才能并行地进行显示。这样，对于一行的显示过程就可以分解列数据准备（传输）和列数据显示两个部分。对于串行传输方式来说，列数据准备时间可能相当长，在行扫描周期确定的情况下，留给行显示的时间就太少了，以至影响到 LED 的亮度。

解决串行传输中列数据准备和列数据显示的时间矛盾问题，可以采用重叠处理的方法。即在显示本行各列数据的同时，准备下一行的列数据。为了达到重叠处理的目的，列数据的显示就需要具有锁存功能。

经过上述分析，可以归纳出列驱动器电路应具备的主要功能。对于列数据准备来说，它应能实现串入并出的移位功能；对于列数据显示来说，应具有并行锁存的功能。这样，本行已准备好的数据打入并行锁存器进行显示时，串并移位寄存器就可以准备下一行的列数据，而不会影响本行的显示。图 2-1 为显示屏电路实现的结构框图。

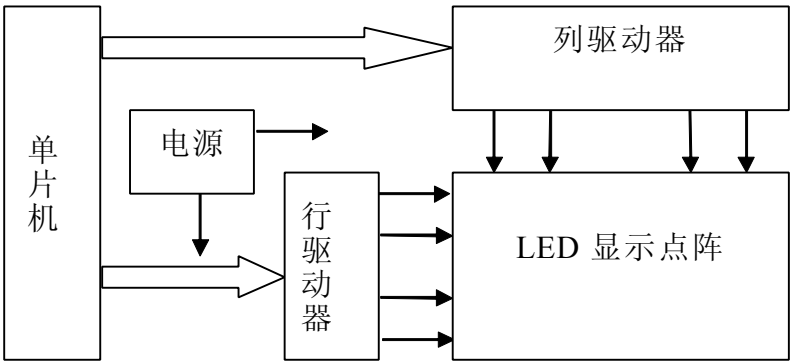


图 2-1 显示屏电路框图

第三章 系统硬件部分设计

由于图文屏的控制电路采用单片机方案，控制功能的实现应在硬件和软件两方面进行折中。单片机及相应软件，主要负责存储（或生成）显示数据、安排控制信号的定时与顺序等。但是单片机的接口数量少，驱动能力不强，必须扩展一定的硬件电路，才能满足显示屏的需要。系统硬件部分电路大致上可以分成稳压电源、单片机系统及外围电路、列驱动器电路、行驱动器电路和 LED 显示屏电路五部分。

3. 1 电源设计

稳压电源的功能是把来自电网的 220V 交流电压转变为所需的、稳定的直流电压。它由电源变压器、整流电路、滤波电路和稳压电路四个部分组成，如图 3-1 所示：

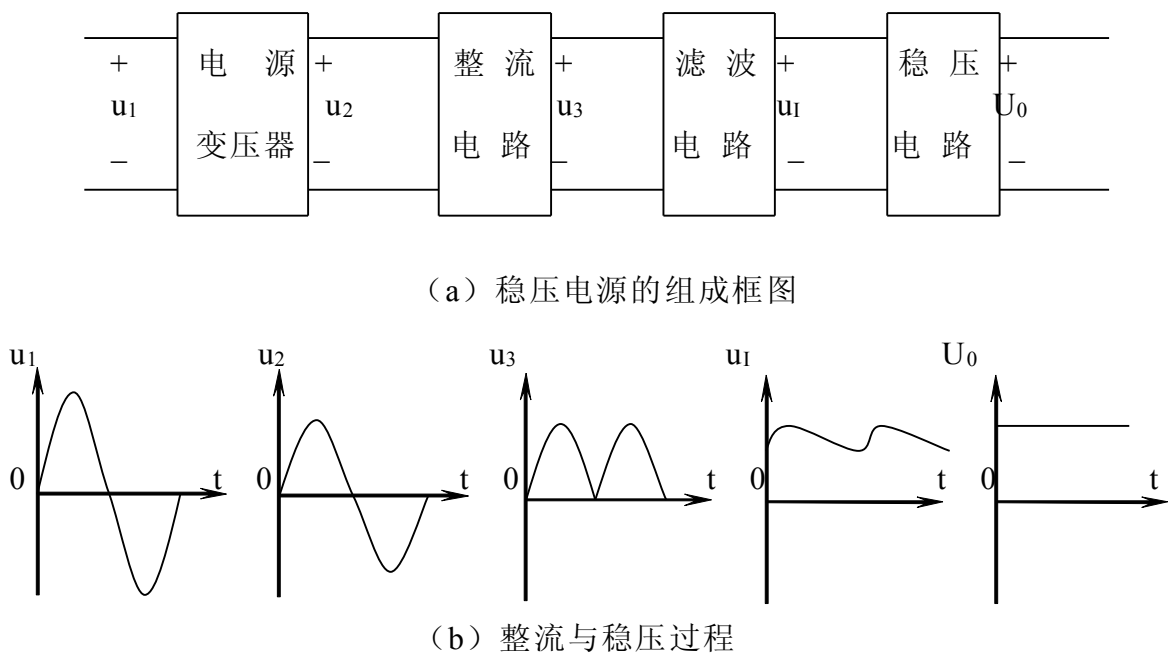


图 3-1 稳压电源的组成框图及整流与稳压过程

根据要求所确定的稳压电源的电路形式如图 3-2 所示。图中变压器 T 将来自电网的 220V 交流电压变换为整流电路所需要的 7.5V 交流电压，再经整流电桥（4 个二极管）D1 将交流电压变换成脉动的直流电压，随后电解电容器 C1 将脉动直流电压中的大部分纹波加以滤除，以得到较为平滑的直流电压。为了得到改善的纹波电压，再将直流电压通过电容 C2，然后经集成稳压器 IC1 稳压，在输出端得到稳定的 5V 直流电压。这时，在输出端接上电容 C3，用以滤除输出端的高频信号，改善负载的瞬态响应，最后即可得到所需的、稳定的直流电压。电路最后接入的发光二极管用做电源指示灯。

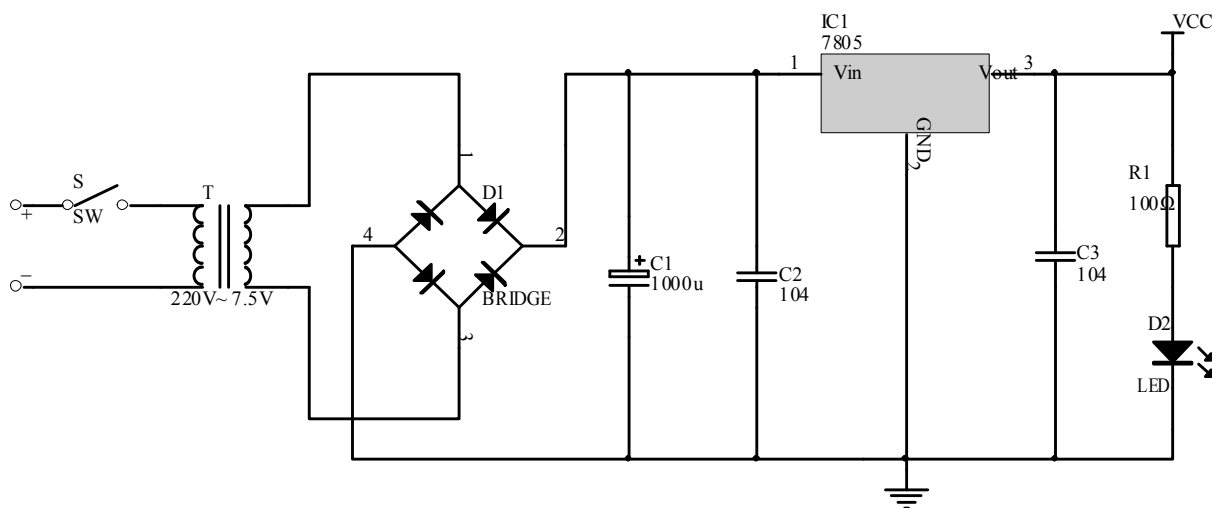


图 3-2 电源电路图

3. 2 单片机系统及外围电路

3. 2. 1 单片机的选择

单片机采用 89C51 或其兼容系列的芯片进行控制，它负责控制整个电路以及相应的程序的运行、以及给屏体电路部分发送命令。这里我们选择了内含 4K 字节 Flash 的 AT89C51，因为我们只需要显示特定的图形和文字，无需庞大的字库，因此 4K Flash 已经可以满足字库储存的需求，不需要扩展外存储器。

3. 2. 2 AT89C51 芯片介绍

AT89C51

AT89C51 是美国 ATMEL 公司生产的 8 位 Flash ROM 单片机。其最突出的优点是片内 ROM 为 Flash ROM，可擦写 1000 次以上，应用并不复杂的通用 ROM 写入器就能方便的擦写，读取也很方便，价格低廉，具有在片程序 ROM 二级保密系统。因此可灵活应用于各种控制领域。AT89C51 包含以下一些功能部件：

- (1) 一个 8 位 CPU ；
- (2) 一个片内振荡器和时钟电路；
- (3) 4KB Flash ROM ；
- (4) 128B 内 RAM；
- (5) 可寻址 64KB 的外 ROM 和外 RAM 控制电路；
- (6) 两个 16 位定时/计数器；
- (7) 21 个特殊功能寄存器 ；
- (8) 4 个 8 位并行 I/O 口；
- (9) 一个可编程全双工串行口 ；
- (10) 5 个中断源，可设置成 2 个优先级。

AT89C51 单片机一般采用双列直插 DIP 封装，共 40 个引脚，图 3-3 为其引脚排列图。40 个引脚大致可分为 4 类：电源、时钟、控制各 I/O 引脚。

一. 电源

Vcc——芯片电源，接+5V；GND——接地端。

二. 时钟

XTAL1、XTAL2——晶体振荡电路反相输入端和输出端。

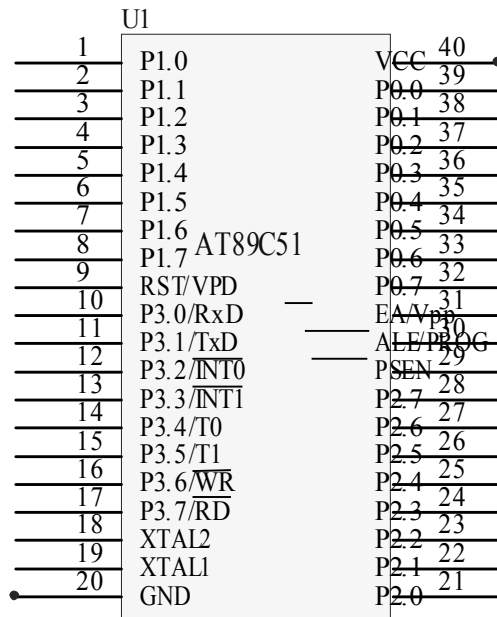


图 3-3 AT89C51 引脚图

三. 控制线

控制线共有 4 根，其中 3 根是复用线。所谓复用线是指具有两种功能，正常使用时是一种功能，在某种条件下是另一种功能。

1. ALE/ \overline{PROG} ——地址锁存允许/片内 EPROM 编程脉冲。

(1) ALE 功能：用来锁存 P0 口送出的低 8 位地址。

AT89C51 在并行扩展外存储器时，P0 口用于分时传送低 8 位地址和数据信号，且均为二进制数。当 ALE 信号有效时，P0 口传送的是低 8 位地址信号；ALE 信号无效时，P0 口传送的是低 8 位数据信号。在 ALE 信号的下降沿，锁定 P0 口传送的内容，即低 8 位地址信号。

需要指出的是，当 CPU 不执行访问外 RAM 指令，ALE 以时钟振荡频率 1/6 的固定频率输出，因此 ALE 信号也可作为外部芯片 CLK 时钟或其他需要。但是，当 CPU 执行 MOVX 指令时，ALE 将跳过一个 ALE 脉冲。

(2) \overline{PROG} 功能：片内 EPROM 的芯片，在 EPROM 编程期间，此引脚输入编程脉冲。

2. \overline{PSEN} ——外 ROM 读选通信号。

89C51 读外 ROM 时，每个机器周期内 \overline{PSEN} 两次有效输出。 \overline{PSEN} 可作为外 ROM

芯片输出允许 \overline{OE} 的选通信号。在读内 ROM 或读外 RAM 时， \overline{PSEN} 无效。

3. RST/V_{PD}——复位/备用电源。

(1) 正常工作时，RST 端为复位信号输入端，只要在该引脚上连续保持两个机器周期以上高电平，AT89C51 芯片即实现复位操作，复位后一切从头开始，CPU 从 0000H 开始执行指令。

(2) V_{PD} 功能：在 V_{CC} 掉电情况下，该引脚可接上备用电源，由 V_{PD} 向片内 RAM 供电，以保持片内 RAM 中的数据不丢失。

4. \overline{EA} /V_{PP}——内外 ROM 选择/片内 EPROM 编程电源。

(1) \overline{EA} 功能：正常工作时， \overline{EA} 为内外 ROM 选择端。AT89C51 单片机 ROM 寻址范围为 64KB，其中 4KB 在片内，60KB 在片外。当 \overline{EA} 保持高电平时，先访问内 ROM，但当 PC 值超过 4KB 时，将自动转向执行外 ROM 中的程序。当 \overline{EA} 保持低电平时，则只访问外 ROM，不管芯片内有无内 ROM。

(2) V_{PP} 功能：片内有 EPROM 的芯片，在 EPROM 编程期间，此引脚用于施加编程电源。

四. I/O 引脚

AT89C51 有 P0、P1、P2、P3 4 个 8 位并行 I/O 端口，共 32 个引脚。

P0 口是一组 8 位漏级开路型双向 I/O 口，也即地址/数据总线复用口。作为输出口用时，每位能以吸收电流的方式驱动 8 个 TTL 逻辑门电路，对端口写 1 时，又可作高阻抗输入端用。在访问外部程序和数据存储器时，它是分时多路转换的地址（低 8 位）/数据总线，在访问期间激活了内部的上拉电阻。在 Flash 编程时，P0 端口接收指令字节；而在验证程序时，则输出指令字节。验证时，要求外接上拉电阻。

P1 口是带内部上拉电阻的双向 I/O 口，向 P1 口写入 1 时 P1 口被内部上拉为高电平，可用作输入口。当作为输入脚时被外部信号拉低的 P1 口会因为内部上拉而输出一个电流。Flash 编程和程序校验期间，P1 接收低 8 位地址。

P2 口是带内部上拉电阻的双向 I/O 口，向 P2 口写入 1 时 P2 口被内部上拉为高电平可用作输入口，当作为输入脚时被外部拉低的 P2 口会因为内部上拉而输出电流。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 MOVX @DPTR 指令）时，P2 口送出高 8 位地址数据，当使用 8 位寻址方式 (MOVX @RI) 访问外部数据存储器时，P2 口发送 P2 特殊功能寄存器的内容，在整个访问期间不改变。Flash 编程和程序校验时，P2 也接收高位地址和一些控制信号。

P3 口是带内部上拉电阻的双向 I/O 口，向 P3 口写入 1 时 P3 口被内部上拉为高电平可用作输入口，当作为输入脚时被外部拉低的 P3 口会因为内部上拉而输出电流。P3 口除了作为一般的 I/O 口线外，更重要的是它的第二功能，如表 3.1 所示：

2. 3
系统外
单
统外围
式如图
示。单
荡器的反
器的输

端口引脚	第二功能
P3.0——RXD	串行口输入端
P3.1——TXD	串行口输出端
P3.2—— $\overline{INT0}$	外部中断0请求输入端
P3.3—— $\overline{INT1}$	外部中断1请求输入端
P3.4——T0	定时/计数器0外部信号输入端
P3.5——T1	定时/计数器1外部信号输入端
P3.6—— \overline{WR}	外RAM写选通信号输出端
P3.7—— \overline{RD}	外RAM读选通信号输出端

单片机
围电路
片机系
电路形
3-4 所
片机振
相放大
入端

(XTAL1) 和输出端 (XTAL2) 之间接上 12MHz 或更高频率的晶振, 以获得较高的刷新频率, 使显示更稳定。电容 C4、C5 是晶振的负载电容, 主要起频率微调 and 稳定的作用。单片机的串行口工作在方式 0 下, 作为同步移位寄存器使用, 端口 RXD (P3.0) 作为数据移位的输入/输出端, 而由 TXD (P3.1) 端输出移位时钟脉冲。移位数据的发送和接收均以 8 位为一帧, 不设起始位和停止位, 无论输入/输出, 均低位在前高位在后。89C51 的通用 I/O 口 P1 作为显示数据和二进制行号的公用输出口。两种数据的输出在时间上是错开的。P1 口的低 4 位与行驱动器相连, 送出二进制的行选信号; P1.5~P1.7 口则用来发送控制信号。P0 和 P2 口空着, 在有必要的时候可以扩展系统的 ROM 和 RAM。

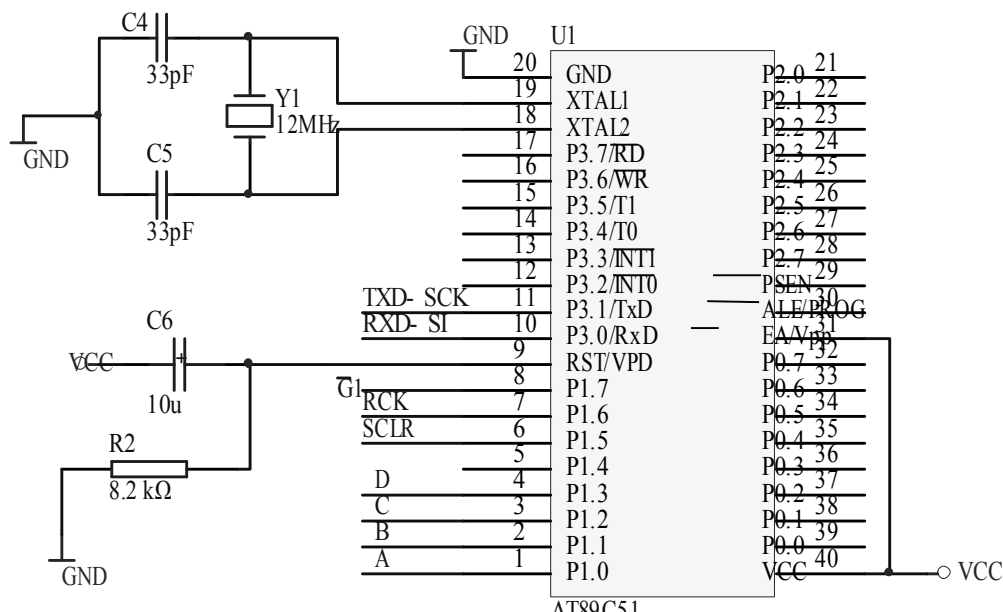


图 3-4 单片机系统外围电路

3. 3 列驱动电路

列驱动电路由集成电路 74HC595 构成，它具有一个 8 位串行输入/输出或者并行输出的移位寄存器和一个 8 位输出锁存器的结构，而且移位寄存器和输出锁存器的控制是各自独立的，可以实现在显示本行各列数据的同时，传送下一行的列数据，即达到重叠处理的目的。

74HC595 的管脚及内部结构形式如图 3-5 所示。它的输入侧有 8 个串行移位寄存器，每个移位寄存器的输出都连接一个输出锁存器。引脚 SER 是串行数据的输入端。引脚 SRCLK 输入移位寄存器的移位时钟脉冲，在其上升沿发生移位，并将 SER 的下一个数据打入最低位。移位后的各位信号出现在各移位寄存器的输出端，也就是输出锁存器的输入端。RCLK 是输出锁存器的打入信号，其上升沿将移位寄存器的输出打入到输出锁存器。引脚 \overline{OE} 是输出三态门的开放信号，只有当其为低时锁存

器的输出才开放，否则为高阻态。 \overline{SRCLR} 信号是移位寄存器的清零输入端，当其为低时移位寄存器的输出全部为 0，由于 SRCLK 和 RCLK 两个信号是互相独立的，所以能够做到输入串行移位与输出锁存互不干扰。芯片的输出端为 QA~QH，最高位 QH 可作为多片 74HC595 级连应用时，向上一级的级连输出。但因 QH 受输出锁存器打入控制，所以还从输出锁存器前引出了 QH'，作为与移位寄存器完全同步的级连输出。移位寄存和输出锁存的时序波形如图 3-6 所示：

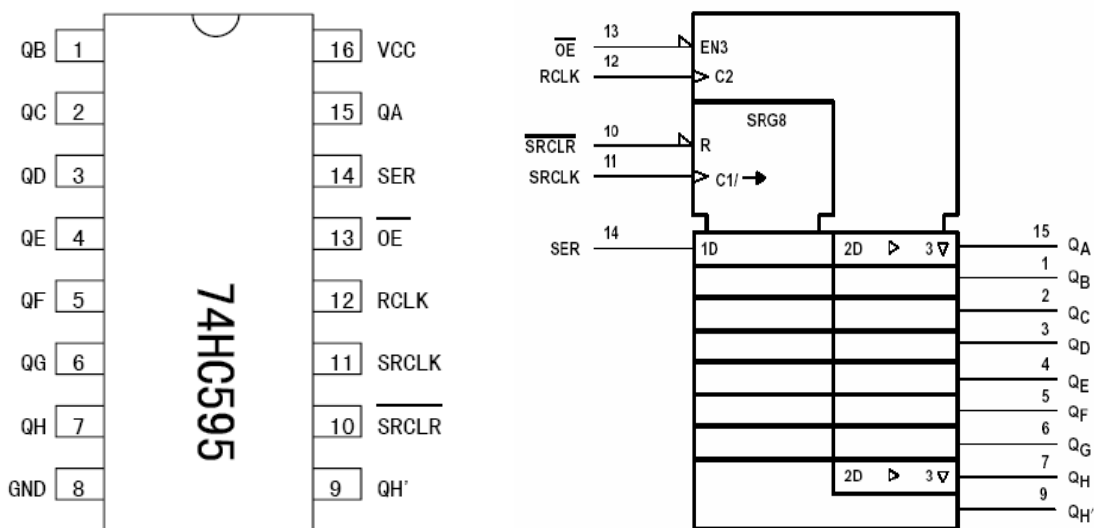


图 3-5 74HC595 的管脚及内部结构逻辑图

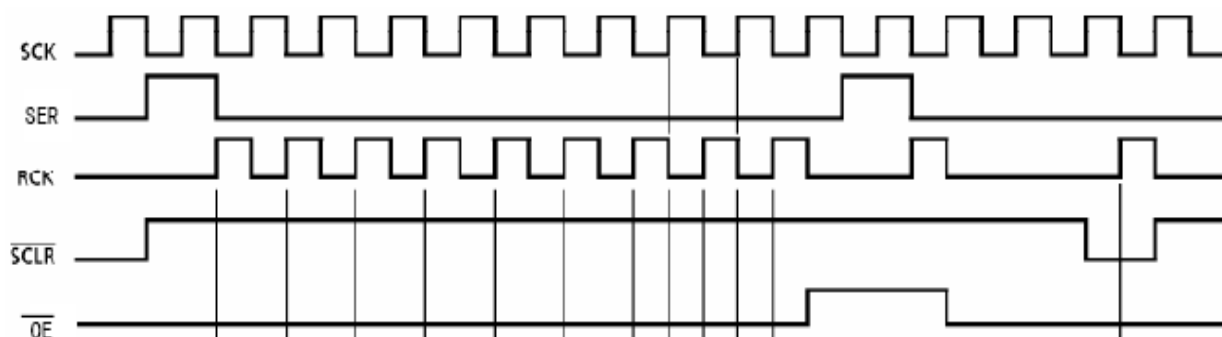


图 3-6 移位寄存和输出锁存的时序波形图

由 74HC595 组成的列驱动器示于图 3-7 中。该图由两片 74HC595 组成 16 列的驱动，由 16 个行驱动器驱动 16 行。第一片列驱动器的 SER 端接单片机的串行列显示数据，其 QH' 端连接第二片的 SER 端，采用这样的方法组成两片的级连。两片相应的 SRCLK、 \overline{SRCLR} 、RCLK 端分别并联，作为统一的串行数据移位信号、串行数据清除信号和输出锁存器打入信号。这样的结构，使得各片串行移位能把 16 列的显示数据依次输入到相应的移位寄存器输出端。移位过程结束之后，控制器输出 RCLK 打入信号，16 列显示数据一起打入相应的输出锁存器。然后选通相应的行，该行的各列就按照显示数据的要求进行显示。

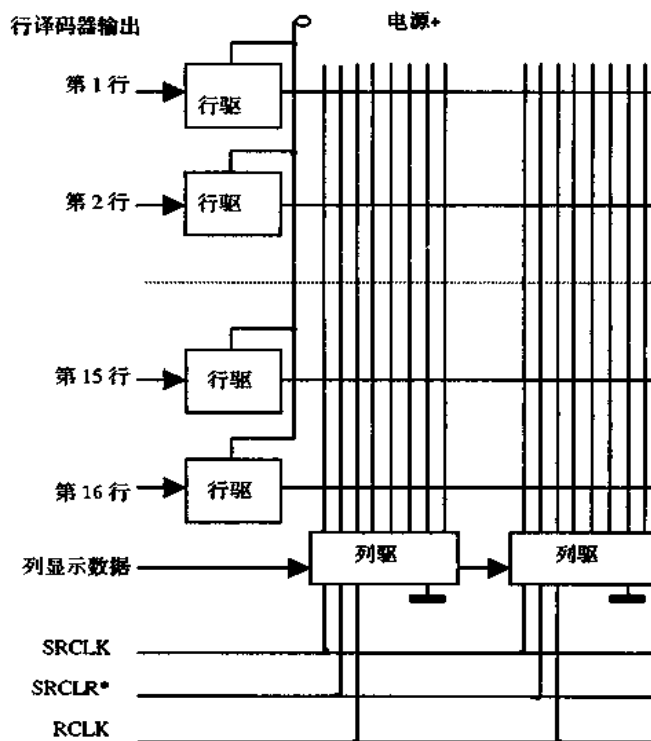


图 3-7 显示驱动电路

3. 4 行驱动电路

3. 4. 1 行驱动芯片 74HC154 介绍

译码器是一个多输入、多输出的组合逻辑电路。其功能是将给定的输入码组进行“翻译”，变换成对应的输入信号，对每一种可能的输入组合，使输出通道中相应的一路有信号输出，一个且仅一个输出信号为有效电位。74HC154 为变量译码器，也称二进制译码器，它是一种四线—十六线译码器，译码的输入端有 4 个，输出端有 $2^4=16$ 个，并有两个选通端（使能端），它的管脚形式如图 3-8 所示，当选通端 $\overline{E}1$ 、 $\overline{E}2$ 均为低电平时，译码器处于工作状态，可将地址输入端（A0~A3）的二进制编码在一个对应的输出端以低电平译出。否则，译码器被禁止，所有的输出端被封锁在高电平。 $\overline{Y}_0 \sim \overline{Y}_{15}$ 为译码输出端，输出是低电平有效，即在选通时，每输入一个二进制代码将使对应的一个输出端为低电平，而其它输出端均为高电平的无效信号，也可以说对应的输出端被“译中”。74HC154 译码器的真值表如表 3. 2 所示：

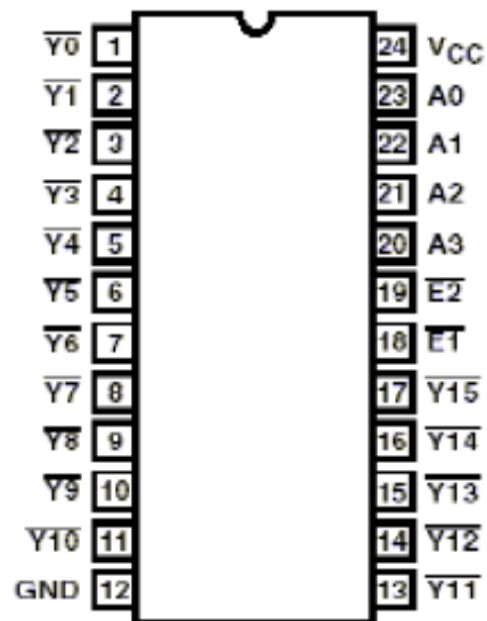


图 3-8 74HC154 管脚图

表 3. 2 74HC154 译码器的真值表

输 入						输 出															
\bar{S}_0	\bar{S}_1	A3	A2	A1	A'0	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12	Y13	Y14	Y15
L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	L	L	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
L	L	H	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H
L	L	H	H	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L
H	L	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H

从真值表可知，每组 4 个变量输入，在 16 个输出中只有一个引脚为“0”（且正好与输入代码是一一对应），其余 15 个全为“1”，这种译码输出称为低电平有效；四线—十六线译码器逻辑形式为：

$$\begin{aligned} \bar{Y}_0 &= \bar{A}_3 \bar{A}_2 \bar{A}_1 \bar{A}_0, \bar{Y}_1 = \bar{A}_3 \bar{A}_2 \bar{A}_1 A_0, \bar{Y}_2 = \bar{A}_3 \bar{A}_2 A_1 \bar{A}_0, \bar{Y}_3 = \bar{A}_3 \bar{A}_2 A_1 A_0, \bar{Y}_4 = \bar{A}_3 A_2 \bar{A}_1 \bar{A}_0, \\ \bar{Y}_5 &= \bar{A}_3 A_2 \bar{A}_1 A_0, \bar{Y}_6 = A_3 \bar{A}_2 \bar{A}_1 \bar{A}_0, \bar{Y}_7 = A_3 \bar{A}_2 \bar{A}_1 A_0, \bar{Y}_8 = A_3 \bar{A}_2 A_1 \bar{A}_0, \bar{Y}_9 = A_3 \bar{A}_2 A_1 A_0, \\ \bar{Y}_{10} &= \bar{A}_3 A_2 \bar{A}_1 \bar{A}_0, \bar{Y}_{11} = \bar{A}_3 A_2 \bar{A}_1 A_0, \bar{Y}_{12} = \bar{A}_3 A_2 A_1 \bar{A}_0, \bar{Y}_{13} = \bar{A}_3 A_2 A_1 A_0, \bar{Y}_{14} = A_3 \bar{A}_2 \bar{A}_1 \bar{A}_0, \\ \bar{Y}_{15} &= A_3 \bar{A}_2 \bar{A}_1 A_0。 \end{aligned}$$

3. 4. 2 行驱动电路

行驱动电路相对简单。行选信号来源于单片机按照时序要求所给出的二进制信号，每次更新行号（开始扫描新的一行）时，由单片机输出 4 位二进制行号，行号经 4/16 线译码器译码后，生成 16 条行选信号线，再经过驱动器驱动对应的行

线。采用译码器的方案，还可以保证同一时刻只选通一条行线，从而达到显示的稳定性。行驱动电路原理如图 3-9 所示：

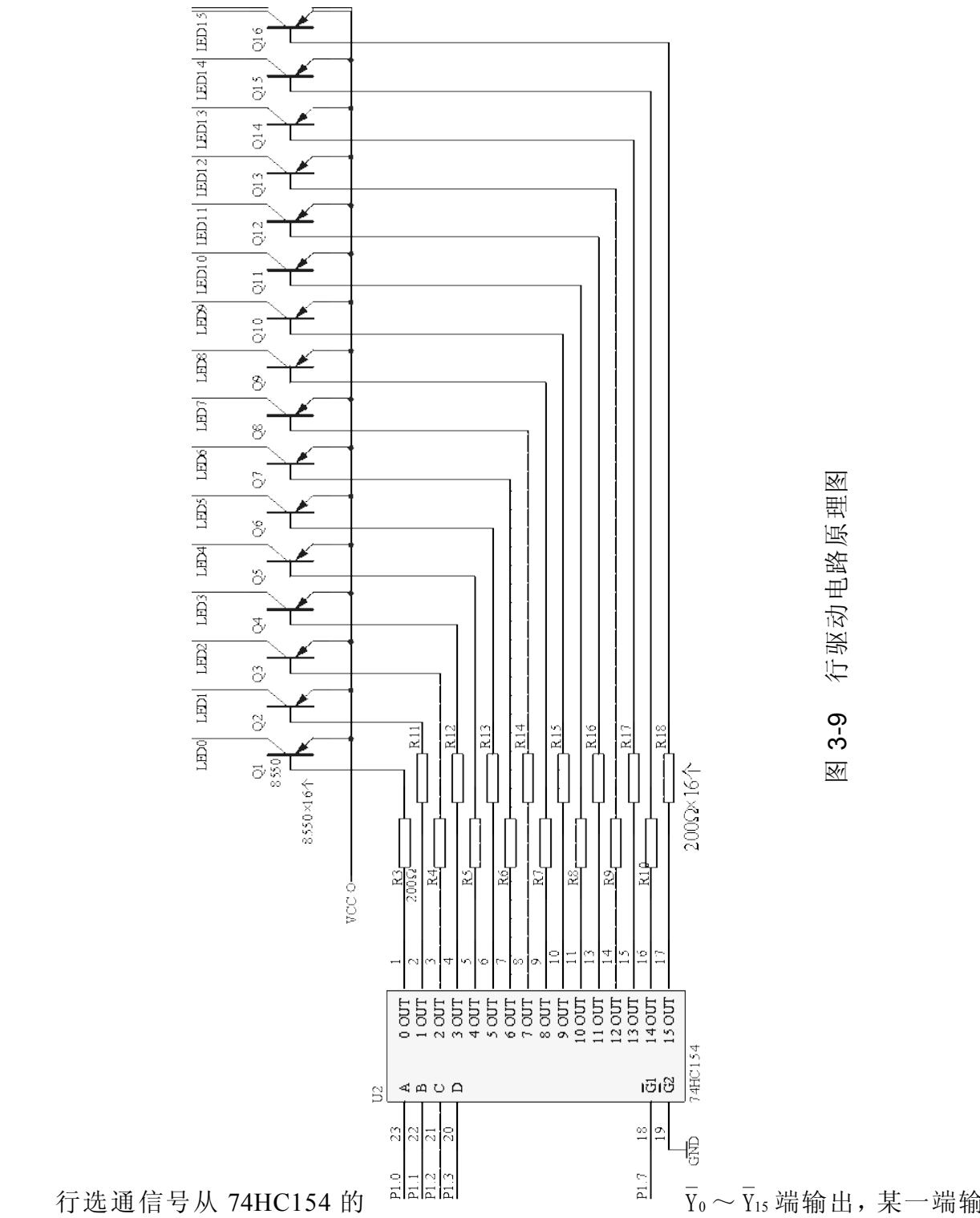


图 3-9 行驱动电路原理图

出低电平即为有效，而其它端输出均为高电平的信号无效。如 \bar{Y}_0 端输出低电平，此信号传至相应的 PNP 型三极管 Q1，此时，三极管的基极为低电平，因此，发射结正向偏置，集电结反向偏置，三极管导通，其集电极再将高电平加于 LED 阵列的对应的行上，即将此行选通；而同时 74HC154 的其它端输出高电平而致使该行对应的三极管截止，从而所对应 LED 行线不被选通。行选通按顺序从 $\bar{Y}_0 \sim \bar{Y}_{15}$ ，全部各行都选通一遍之后又重新开始，这就是行驱动电路的逐行扫描过程。行信号 A、B、C、D 的顺序变化范围从 0000、0001、0010、至 1111，来一个选通信号，行信号顺序就变化一次，其频率由扫描电路决定。

由于行驱动电路一条行线上要带动 16 列的 LED 进行显示，按每一 LED 器件 20mA 电流计算，16 个 LED 同时发光时，就需要 $16 \times 20 = 320\text{mA}$ 的驱动电流，选用三极管 8550 作为驱动管可以在逻辑功能和驱动能力上符合 LED 的驱动要求。

3. 5 LED 显示屏电路

LED 显示屏是将发光二极管按行按列布置的，驱动时也就按行按列驱动。在扫描驱动方式下可以按行扫描按列控制，当然也可以按列扫描按行控制。LED 显示屏现多采用多块 8×8 点阵显示单元拼接而成。本文就是使用 4 块 SBM1388 型号的实验模块组成 16×16 点阵，以满足汉字显示的要求。 8×8 LED 点阵是最基本的点阵显示模块，理解了 8×8 LED 点阵的工作原理就可以基本掌握 LED 点阵显示技术。 8×8 点阵 LED 结构如图 3-10 所示，其等效电路如图 3-11 所示：

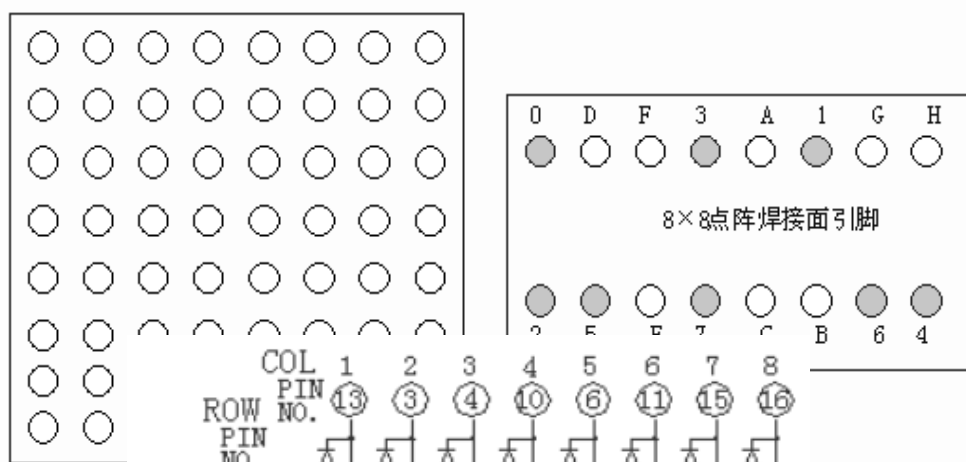


图 3-10 8×8 点阵 LED 结构图

图 3-11 8×8 点阵 LED 等效电路图

从图中（本图的 LED 阵列采用共阳的接法）可以看出，8×8 点阵共需要 64 个发光二极管组成，且每个发光二极管是放置在行线和列线的交叉点上。要实现显示图形或字体，只需考虑其显示方式，通过编程控制各显示点对应 LED 阳极和阴极端电平，就可以有效的控制各显示点的亮灭。当采用按行扫描按列控制的驱动方式时，LED 显示屏 8 行的同名列共用一套列驱动器。行驱动器一行的行线连接到电源的一端，列驱动器一系列的列线连接到电源的另一端。应用时还应在各条行线或列线上接上限流电阻。扫描中控制电路将行线的 1 到 8 轮流接通高电位，使连接到各该行的全部 LED 器件接通正电源，但具体那一个 LED 导通，还要看它的负电源是否接通，这就是列控制的任务了。当对应的某一系列置 0 电平，则相应的二极管就亮；反之则不亮。例如：如果想使屏幕左上角 LED 点亮，左下角 LED 熄灭的话，在扫描到第一行时，第一列的电位就应该为低，而扫描到第八行时第一列的电位就应该为高。这样行线上只管一行一行的轮流导通，列线上进行通断控制，实现了行扫描列控制的驱动方式。

以上就是 16×16 点阵 LED 电子显示屏系统硬件部分的各功能模块分述，经过合理的设计论证后和就可以整体结合在一起，再根据设计配备好各元器件实物，通过组装后等在单片机的程序存储器里放入编制好的程序即可成为一个完整的应用系统。系统综合电路原理图见附录一。

第四章 系统软件部分设计

在单片机系统中，硬件是系统的基础，软件则是在硬件的基础上对其合理的调配和使用，从而完成应用系统所要完成的任务。软件的设计是设计控制系统的应用程序。其任务是在总体设计和硬件设计的基础上，确定程序结构，分配内 RAM 资源，划分功能模块，然后进行主程序和各模块程序的设计，最后连接起来成为一个完整的应用程序。

在进行系统总体设计时，曾经规划过软件结构，但由于硬件系统尚未仔细确定，软件结构框图十分粗糙，当硬件设计接口扩展及各功能模块与 CPU 连接关系确定后，就能够具体明确对软件设计的要求。本设计的 LED 显示屏软件的主要功能是向屏体提供显示数据，并产生各种控制信号，使屏幕按设计的要求显示。根据软件分层次设计的原理，可把显示屏的软件系统分成两大层：第一层是底层的显示驱动程序，第二层是上层的系统应用程序。显示驱动程序负责向屏体送显示数据，并负责产生行扫描信号和其它控制信号，配合完成 LED 显示屏的扫描显示工作。显示驱动程序由定时器 T0 中断程序实现。系统应用程序完成系统环境设置（初始化）、显示效果处理等工作，由主程序来实现。

从有利于实现语言的直观，易于记忆和检查，可读性较好和使语言程序占用较少的单片机存储空间，实时处理系统可以快速的执行指令的角度考虑，本设计中显示屏程序采用汇编语言编写。

4.1 系统主程序

系统主程序的总体结构如图 4-1 所示。系统的主程序开始以后，首先是对系统环境初始化，包括设置串口、定时器、中断和端口；然后 LED 显示屏进入开机状态，转入正常的显示。首先是向上滚动显示“单片机是工业中最基本的应用方式”15 个汉字，每个字停留时间约 1.6s；接着再向上滚动显示“我爱单片机”这 5 个汉字。由于单片机没有停机指令，所以可以设置系统程序不断地循环执行上述显示效果。

系统程序结构属中断方式，绝大多数功能在中断服务子程序中完成。根据总体结构，可将程序划分为几个功能化模块：串行口中断服务程序、多字滚动显示子程序、单字显示子程序、扫描程序。各个模块可进行独立设计、调试和查错，最终再连接成一个整体。这样可方便程序调用，程序整体层次清晰，结构一目了然，方便阅读。

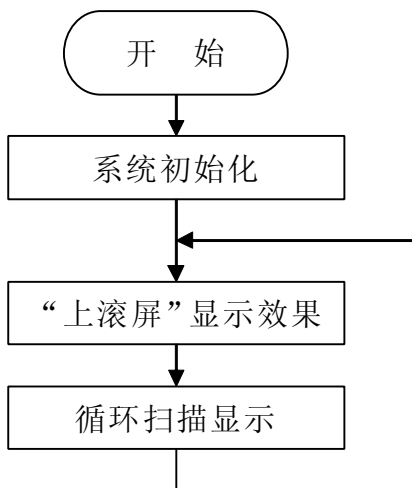


图 4-1 系统主程序的总体结构

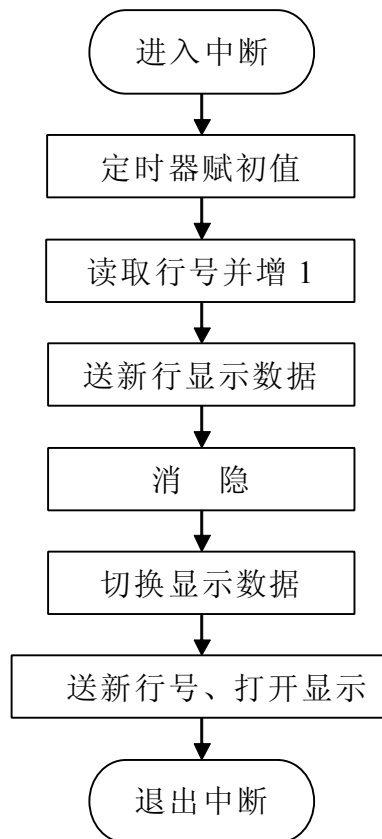


图 4-2 显示驱动程序流程图

流
显
重

4. 2 显示驱动程序

图 4-2 为显示驱动程序（显示屏扫描函数）程图。根据流程图可以编写出相应的显示程序。示驱动程序在进入中断后首先要对定时器 T0 新赋初值，以保证显示屏刷新率的稳定，1/16 扫描显示屏的刷新率（帧频）计算公式如下：

$$\begin{aligned} \text{刷新率（帧频）} &= \frac{1}{16} \times T0 \text{ 溢出率} \\ &= \frac{1}{16} \times \frac{f_{osc}}{12(65536 - t_0)} \end{aligned}$$

其中 f_{osc} 为晶振频率， t_0 为定时器 T0 初值（工作在 16 位定时器模式）。

$$T0 \text{ 初值} = 2^N - \frac{\text{定时时间}}{\text{机周时间}}$$

其中 N 与工作方式有关，本设计定时/计数器工作于方式 1，如图 4-3 所示，方式 1 构成一个 16 位定时计数器，即 N=16。

方式 1 最大计数值为 $2^{16}=65536$ ，若 $f_{osc}=12\text{MHz}$ ，则定时时间范围为 1~65536 μs 。

机周时间与主振频率有关，机器周期是时钟周期的 12 倍，因此，机周时间=12/ f_{osc} 。当 $f_{osc}=12\text{MHz}$ 时，1 机周=1 μs 。

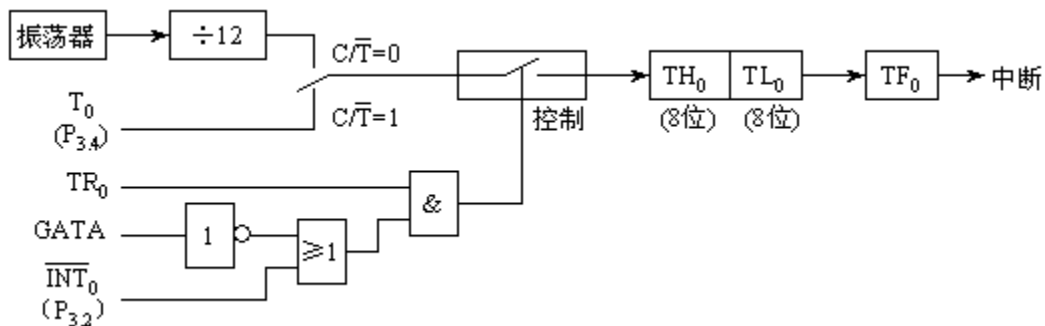


图 4-3 定时/计数器 T0 工作方式 1 逻辑电路结构图

然后显示驱动程序查询当前燃亮的行号，从显示缓存区内读取下一行的显示数据，并通过串口发送给移位寄存器。为消除在切换行显示数据的时候产生拖尾现象，驱动程序先要关闭显示屏，即消隐，等显示数据打入输出锁存器并锁存，然后再输出新的行号，重新打开显示。显示程序给出了各控制信号和显示数据，其作用及相互配合关系均已在硬件部分作过介绍，相应的程序部分在清单的注释区内分别进行了说明，在此不再一一详述。系统程序清单见附录二。

第五章 调试及性能分析

一个单片机系统经过总体设计，完成了硬件和软件设计开发。元器件安装后，在系统的程序存储器中写入编制好的应用程序，系统即可运行。但编制好的程序或焊接好的线路不能按预计的那样正常工作是常见的事，多少会出现一些硬件、软件上的错误。这就需要通过调试来发现错误并加以改正。调试可分为硬件调试和软件调试。在允许的条件下，根据本设计系统的需求性首先采用在 PC 机上用模拟开发软件进行检测和调试，然后进行硬件的组装与调试。

5.1 软件调试

软件调试采用脱机调试的方法，即完全用仿真器软件在 PC 机上对目标电路原理图和程序进行检测和调试。调试过程中单片机相应输入端由通用键盘和鼠标设定，运行状态、各寄存器状态、端口状态等都可以在 CRT 指定的窗口区域显示出来，以确定程序运行有无错误。调试可按下列步骤进行：

一. 目标程序纠错

该阶段工作通常在目标程序编辑时就完成。一般来说，仿真器软件能给用户输入的程序指令纠错，包括书写格式、标号未定义或多重定义、转移地址溢出等错误。

二. 子程序功能调试

程序设计通常采用模块程序设计，调试时可对一个个子程序分别进行调试，设置好入口条件，然后采用单步运行或断点运行方式，检查系统 CPU 现场、RAM 和 I/O 口状态，看程序执行结果是否符合设计要求。

三. 整体程序综合调试

即把各子程序整体连起来进入到综合电路调试，看是否能实现预计的功能显示。在这阶段若发生故障，可以考虑各子程序在运行时是否破坏现场，数据缓冲单元是否发生冲突，标志位的建立和清除在设计上是否失误，堆栈是否溢出，输入输出状态是否正常等。

用软件模拟器调试不需任何在线仿真器，也不需要用户样机，直接就可以在 PC 机上开发和调试。调试和修改完毕后，对于实时性要求不高的应用系统可以直接使用编程器将软件固化在目标系统 ROM 中，然后独立投入运行。

5.2 硬件调试

单片机应用系统的硬件调试和软件调试是分不开的，许多硬件故障在软件调试时才能发现，但通常要先排除系统中明显的硬件故障。调试工作可以分为四步：

一. 目测检查

根据硬件逻辑设计图，仔细检查样机线路是否连接正确，并核对元器件的型号、规格和安装是否符合要求，必要时可用万用表检测线路通断情况。

二. 电源调试

样机的第一次通电测试很重要，若样机中存在电源故障，则加电后将造成器件损坏。调试的方法有两种：一种是断开样机稳压电源的输出端检查空载时电源工作

情况；另一种是拔下样机上的主要集成芯片，检查电源的负载能力（用假负载）。确保电源无故障并性能符合设计要求。

三. 通电检查

在确保电源良好前提下，接通电源。最好在电源与其余电路之间串接一个电流表。若接通后电流很大，必须立即切断电源。电源大得超出正常范围，说明电路中有短路或故障。通电检查的主要目的是看系统是否存在短路或由元器件损坏、装配错误引起的电流异常。

四. 检查相应芯片的逻辑关系

加电后检查各芯片插座上相关引脚的电位，仔细测量相应的输入输出电平是否正常。单片机系统大都是数字逻辑电路，使用电平检查法可首先查出逻辑设计是否正确，选用器件和连接关系是否符合要求等。

5. 3 性能分析

系统通过调试组装后一般即可正常工作。LED 显示屏的刷新频率及显示显示效果是主要的性能指标。显示屏刷新率由定时器 T0 的溢出率和单片机的晶振频率决定，表 5-1 给出了显示屏采用的频率及其对应的定时器 T0 初值。

表 5. 1 显示屏刷新率（帧率）与 T0 初值关系表

刷新率/Hz	25	50	62.5	75	85	100	120
T0 初值	OxEC78	OxF63C	OxF830	OxF97E	OxFA42	OxFB1E	OxFBEE

从理论上来说，24Hz 以上的刷新率就能看到连续稳定的显示，刷新率越高，显示越稳定，同时刷新率越高，显示驱动程序占用的 CPU 时间也越多。实验证明，在目测条件下刷新率 40Hz 以下的画面看起来闪烁较严重，刷新率 50Hz 以上的已基本觉察不出画面闪烁，刷新率达到 85Hz 以上时再增加画面闪烁没有明显改善。

显示效果处理的内容和方法非常广泛，根据需求可做变动，在这里不做具体讨论。

根据实践证明，这个方案设计的 16×16 点阵 LED 图文显示屏，结构合理，成本较低，且较容易扩展成更大的显示屏；显示屏各点亮度均匀、充足；显示图形或文字稳定、清晰无串扰；可用静止、移入移出等多种显示方式显示图形或文字。

结束语

以上仅对LED显示屏的结构和驱动、显示电路原理作一详细介绍和分析。LED点阵的应用很广，对于不同的应用环境和应用要求，可以有各种各样的应用方式，在这里就不做详尽叙述。由于自己知识水平的局限和时间的仓促，设计中或还存在着一些不足，我真诚的接受老师们的批评和指正。

经过了几周的努力学习和工作，我终于完成了论文。从课题选择、方案论证到具体设计，每一步对我来说无疑是巨大的尝试和挑战，我不断地给自己提出新的问题，然后去论证、推翻，不懂就请教老师或同学，再接着提出新的问题，在这个往复的过程中，我这篇稚嫩的设计日趋完善。每一次改进我都收获良多，虽然我的设计作品不是很成熟，而且借鉴了前人的很多资料，但我仍然心里有一种莫大的幸福感，因为我实实在在地走过了一个完整的设计所应该走的每一个过程，并且享受了每一个过程，更重要的是这个设计中我加入了自己鲜活的思想。

在做这次毕业设计过程中使我学到了很多，加深了对数字电路的理解，验证了所学理论知识，提高了基本的解决实际问题的能力，并增加了对电子设计方面的兴趣。更重要的是我体会到不论做什么事都要真真正正用心去做，才会使自己更好的成长，没有学习就不可能有实践的能力，实践才是最终的目的，没有实践就不会有自己的突破和创新，希望这次的经历能让我在以后的工作和生活中不断成长与进步。

就此作结！

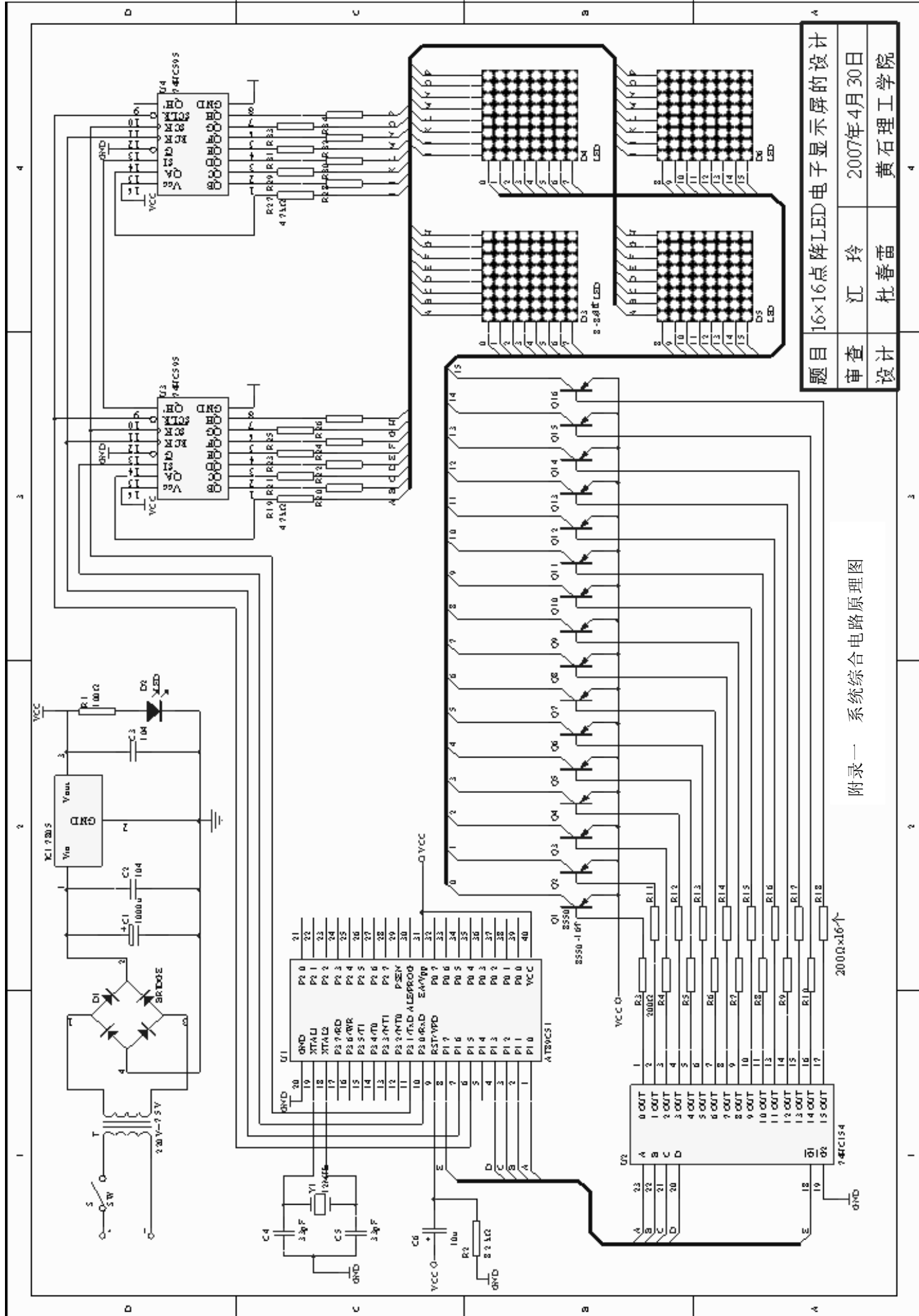
致谢

在论文即将完成之际，我的心情不是抵达终点的欣喜，虽然这里面有我的辛勤付出，但是，由于经验的匮乏，难免有许多考虑不周全的地方。在课题设计的整个过程，以及毕业论文的写作、修改和定稿过程中，得到了老师和同学们的热心帮助。

首先我要衷心的感谢江玲老师对我整个毕业设计由始至终地悉心督促指导，并提供了大量而又详实的资料给我，使我从困惑迷茫的初期走向论文的顺利完成，在完成论文的同时也深受启发和教育，使我受益终生；然后还要感谢进大学以来所有的老师们，为我打下电气专业知识的基础；同时还要感谢我的同学和朋友们，正是因为有了你们的热心帮助和鼓励，使我有一个愉快而又严肃认真的学习和生活环境，这段日子成为我一生中美好的回忆。

最后感谢电气学院和我的母校——黄石理工学院三年来对我的大力栽培。

总之，如果没有你们我想圆满完成这个设计是难以想象的，在这里请最后接受我最诚挚的谢意！



附录一 系统综合电路原理图

题目	16×16点阵LED电子显示屏的设计		
审查	江 玲	2007年4月30日	
设计	杜春雷	黄石理工学院	

附录二 系统程序清单

以下为用汇编语言编写的字符显示控制程序：

```

; *****
; *
; *   单个 16×16 点阵电子屏字符显示器   *
; *       AT89C51       12 MHz 晶振       *
; *       2007. 4. 30       LRM           *
; *****

; 显示字用查表法，不占内存，字符用 16 × 16 共阳 LED 点阵，
; 效果：向上滚动显示 20 个字，再重复循环。
; R1：查表偏址寄存器，B：查表首址，R2：扫描地址（从 00~0FH）。
; R3：滚动显示时控制移动速度，单字显示可控制静止显示的时间。
; *****;
;   中断入口程序   ;
; *****;
;
ORG    0000H        ; 复位地址
LJMP   START        ; 跳到标号 START 执行
ORG    0003H        ; 外中断 0 中断程序入口
RETI                   ; 外中断 0 中断返回
ORG    000BH        ; 定时器 T0 中断程序入口
LJMP   INTT0        ; 跳至 INTT0 执行
ORG    0013H        ; 外中断 1 中断程序入口
RETI
ORG    001BH        ; 定时器 T1 中断程序入口
RETI
ORG    0023H        ; 串行中断程序入口地址
RETI
ORG    002BH        ; 定时器 T2 中断入口地址
RETI
;
; *****;
;   初始化程序   ;
; *****;
;
```

```

;
; *****;
;      主程序      ;
; *****;
;
START:      MOV      20H, #00H      ; 清标志, 00H 为第 16 行开始扫描标志,
                                         01 为 1 帧
                                         ; 扫描结束标志
                                         ; 端口初始化

          MOV      A, #0FFH
          MOV      P1, A
          MOV      P2, A
          MOV      P3, A
          MOV      P0, A
          CLR      P1.6      ; 串行寄存器输入打入输出控制位
          MOV      TMOD, #01H      ; 使用 T0 作 16 位定时器, 行扫描用。
          MOV      TH0, 0FCH      ; 1 ms 初值 (12 MHz)
          MOV      TL0, #18H
          MOV      SCON, #00H      ; 串口 0 方式传送显示字节
          MOV      IE, #82H      ; T0 中断允许, 总中断允许
          MOV      SP, #70H

MAIN:      LCALL   DIS1      ; 显示准备, 黑屏, 1.5s
          MOV      DPTR, #TAB
          LCALL   MOVDISP      ; 向上滚动显示一页 (8 个字)
          INC      DPH
          LCALL   MOVDISP      ; 向上滚动显示一页 (8 个字)
          INC      DPH
          LCALL   MOVDISP      ; 向上滚动显示一页 (8 个字)
          AJMP    MAIN

;
;
; *****;
; 多字滚动显示子程序 ;
; *****;
; 每次 8 个字, 入口时定义好 DPTR 值
;
MOVDISP:   MOV      B, #00H      ; 向上移动显示, 查表偏址暂存 (从 00
                                         开始)
DISLOOP:   MOV      R3, #07H      ; 移动速度
DISMOV:    MOV      R2, #00H      ; 第 0 行开始
          MOV      R1, B

```

```

                SETB    TR0                ; 开扫描（每次一帧）
WAITMOV:        JBC     01H, DISMOV1      ; 标志为 1 扫描一帧结束（16 ms 为 1 帧，
                                         每行 1 ms）

                AJMP    WAITMOV
DISMOV1:        DJNZ    R3, DISMOV        ; 1 帧重复显示（控制移动速度）
                INC     B                  ; 显示字的下一行（每行 2 字节）
                INC     B
                MOV     A, R1              ; R1 为 0, 8 个字显示完
                JZ       MOVOUT
                AJMP    DISLOOP
MOVOUT:         RET                       ; 移动显示结束
;
;
; *****;
;   单字显示子程序   ;
; *****;
; 显示表中某个字
DIS1:           MOV     R3, #5AH           ; 静止显示时间控制（16 ms * # = 1.6 s）
DIS11:          MOV     R2, #00H           ; 一帧扫描初始值（行地址从 00~0FH）
                MOV     DPTR, #TAB         ; 取表首址
                MOV     R1, #00H           ; 查表偏址（显示第一个字）
                SETB    TR0                ; 开扫描（每次一帧）
WAIT11:         JBC     01H, DIS111        ; 为 1，扫描一帧结束
                AJMP    WAIT11
DIS111:         DJNZ    R3, DIS11
                RET
;
;
; *****;
;   扫描程序   ;
; *****;
; 1ms 刷新一次，每行显示 1s

INTT0:          PUSH    ACC
                MOV     TH0, #0FCH         ; 1 ms 初值重装
                MOV     TL0, #18H
                JBC     00H, GOEND          ; 16 行扫描标志为 1，结束
                INC     R1                 ; 取行右边字节偏址
                MOV     A, R1
                MOVC    A, @A+DPTR         ; 查表

```

```

        MOV     SBUF, A           ; 串口 0 方式发送
WAIT:    JBC     T1, GO           ; 等待发送完毕
        AJMP    WAIT
GO:      DEC     R1               ; 取行左边字节偏址
        MOV     A, R1
        MOVC    A, @A+DPTR
        MOV     SBUF, A
WAIT1:   JBC     T1, GO1
        AJMP    WAIT1
GO1:     SETB    P1.7             ; 关行显示, 准备刷新
        NOP                     ; 串口寄存器数据稳定
        SETB    P1.6             ; 产生上升沿, 行数据打入输出端
        NOP
        NOP
        CLR     P1.6             ; 恢复低电平
        MOV     A, R2             ; 修改显示行地址
        ORL     A, #0F0H          ; 修改显示行地址
        MOV     R2, A            ; 修改显示行地址
        MOV     A, P1             ; 修改显示行地址
        ORL     A, #0FH          ; 修改显示行地址
        ANL     A, R2             ; 修改显示行地址
        MOV     P1, A            ; 修改完成
        CLR     P1.7             ; 开行显示
        INC     R2                ; 下一行扫描地址值
        INC     R1
        INC     R1                ; 下一行数据地址
        MOV     A, R2
        ANL     A, #0FH
        JNZ     GO2
        SETB    00H              ; R2 为 10H, 现为末行扫描, 置标志
GO2:     POP     ACC
        RETI
GOEND:   CLR     TR0              ; 一帧扫描完, 关扫描
        SETB    01H              ; 一帧扫描完, 置结束标志
        POP     ACC
        RETI                      ; 退出
;
; *****;
;   扫描文字表   ;
; *****;

```

;

TAB: DB OFFH, OFFH, OFFH, OFFH, OFFH, OFFH, OFFH, OFFH, OFFH, OFFH, OFFH, OFFH, OFFH, OFFH ; 黑屏

DB OFFH, OFFH, OFFH, OFFH, OFFH, OFFH, OFFH, OFFH, OFFH, OFFH, OFFH, OFFH, OFFH, OFFH

DB 0F7H, 0DFH, 0F9H, 0CFH, 0FBH, 0BFH, 0C0H, 007H, 0DEH, 0F7H, 0C0H, 007H, 0DEH, 0F7H, 0DEH, 0F7H ; 单

DB 0C0H, 007H, 0DEH, 0F7H, 0FEH, 0FFH, 000H, 001H, 0FEH, 0FFH, 0FEH, 0FFH, 0FEH, 0FFH, 0FEH, 0FFH

DB OFFH, 0BFH, 0EFH, 0BFH, 0EFH, 0BFH, 0EFH, 0BBH, 0E0H, 001H, 0EFH, 0FFH, 0EFH, 0FFH, 0EFH, 0FFH ; 片

DB 0E0H, 00FH, 0EFH, 0EFH, 0EFH, 0EFH, 0EFH, 0EFH, 0DFH, 0EFH, 0DFH, 0EFH, 0BFH, 0EFH, 07FH, 0EFH

DB 0EFH, 0FFH, 0EFH, 007H, 0EFH, 077H, 001H, 077H, 0EFH, 077H, 0EFH, 077H, 0C7H, 077H, 0CBH, 077H ; 机

DB 0ABH, 077H, 0AFH, 077H, 06EH, 0F7H, 0EEH, 0F5H, 0EDH, 0F5H, 0EDH, 0F5H, 0EBH, 0F9H, 0EFH, 0FFH

DB 0FFH, 0FFH, 0F0H, 00FH, 0F7H, 0EFH, 0F0H, 00FH, 0F7H, 0EFH, 0F0H, 00FH, 0FFH, 0FFH, 000H, 001H ; 是

DB 0FEH, 0FFH, 0F6H, 0FFH, 0F6H, 007H, 0F6H, 0FFH, 0EAH, 0FFH, 0DCH, 0FFH, 0BFH, 001H, 0FFH, 0FFH

DB 0FFH, 0FFH, 0C0H, 003H, 0FEH, 0FFH, 0FEH, 0FFH, 0FEH, 0FFH, 0FEH, 0FFH, 0FEH, 0FFH, 0FEH, 0FFH ; 工

DB 0FEH, 0FFH, 0FEH, 0FFH, 0FEH, 0FFH, 0FEH, 0FFH, 0FEH, 0FFH, 000H, 001H, 0FFH, 0FFH, 0FFH, 0FFH

DB 0FBH, 0BFH, 0FBH, 0BFH, 0FBH, 0BFH, 0FBH, 0BBH, 0BBH, 0B9H, 0DBH, 0B3H, 0DBH, 0B7H, 0EBH, 0AFH ; 业

DB 0E3H, 0AFH, 0EBH, 09FH, 0FBH, 0BFH, 0FBH, 0BFH, 0FBH, 0BBH, 000H, 001H, 0FFH, 0FFH, 0FFH, 0FFH

DB 0FEH, 0FFH, 0FEH, 0FFH, 0DEH, 0F7H, 0C0H, 003H, 0DEH, 0F7H, 0DEH, 0F7H, 0DEH, 0F7H, 0DEH, 0F7H ; 中

DB 0DEH, 0F7H, 0C0H, 007H, 0DEH, 0F7H, 0FEH, 0FFH, 0FEH, 0FFH, 0FEH, 0FFH, 0FEH, 0FFH, 0FEH, 0FFH

DB 0FEH, 0FFH, 0FEH, 0FFH, 0DEH, 0F7H, 0C0H, 003H, 0DEH, 0F7H, 0DEH, 0F7H, 0DEH, 0F7H, 0DEH, 0F7H ; 中

DB 0DEH, 0F7H, 0C0H, 007H, 0DEH, 0F7H, 0FEH, 0FFH, 0FEH, 0FFH, 0FEH, 0FFH, 0FEH, 0FFH, 0FEH, 0FFH

DB 0E0H, 00FH, 0EFH, 0EFH, 0E0H, 00FH, 0EFH, 0EFH, 0E0H, 00FH, 0FFH, 0FFH, 000H, 001H, 0DDH, 0FFH ; 最

DB 0C1H, 003H, 0DDH, 077H, 0C1H, 0AFH, 0DCH, 0DFH,

	0C1H, 0AFH, 01DH, 071H, 0FCH, 0FBH, 0FDH, 0FFH	
DB	0F7H, 0DFH, 0F1H, 0DFH, 080H, 003H, 0F7H, 0DFH,	
	0F0H, 01FH, 0F7H, 0DFH, 0F0H, 01FH, 0F7H, 0DFH	; 基
DB	000H, 001H, 0F7H, 0DFH, 0EEH, 0E7H, 0C0H, 011H,	
	03EH, 0FBH, 0FEH, 0FFH, 080H, 003H, 0FFH, 0FFH	
DB	0FEH, 0FFH, 0FEH, 0FFH, 0FEH, 0FFH, 000H, 001H,	
	0FCH, 07FH, 0FCH, 0BFH, 0FAH, 0BFH, 0FAH, 0DFH	; 本
DB	0F6H, 0EFH, 0EEH, 0E7H, 0D0H, 011H, 03EH, 0FBH,	
	0FEH, 0FFH, 0FEH, 0FFH, 0FEH, 0FFH, 0FFH, 0FFH	
DB	0EFH, 07FH, 0EFH, 07FH, 0DFH, 07FH, 083H, 003H,	
	0BAH, 0FBH, 0BAH, 0FBH, 0B9H, 0FBH, 083H, 07BH	; 的
DB	0BBH, 0BBH, 0BBH, 09BH, 0BBH, 0DBH, 0BBH, 0FBH,	
	083H, 0FBH, 0BBH, 0D7H, 0BFH, 0EFH, 0FFH, 0FFH	
DB	0FEH, 0FFH, 0FFH, 07FH, 0C0H, 003H, 0DFH, 0FFH,	
	0DDH, 0FFH, 0DEH, 0F7H, 0CFH, 073H, 0D7H, 037H	; 应
DB	0DBH, 06FH, 0DBH, 06FH, 0D9H, 0DFH, 0BBH, 0DFH,	
	0BFH, 0BFH, 0A0H, 001H, 07FH, 0FFH, 0FFH, 0FFH	
DB	0FFH, 0FFH, 0E0H, 003H, 0EFH, 07BH, 0EFH, 07BH,	
	0EFH, 07BH, 0E0H, 003H, 0EFH, 07BH, 0EFH, 07BH	; 用
DB	0EFH, 07BH, 0E0H, 003H, 0EFH, 07BH, 0EFH, 07BH,	
	0DFH, 07BH, 0DFH, 07BH, 0BFH, 06BH, 07FH, 077H	
DB	0FDH, 0FFH, 0FEH, 0FFH, 0FFH, 07FH, 000H, 001H,	
	0FDH, 0FFH, 0FDH, 0FFH, 0FCH, 00FH, 0FDH, 0EFH	; 方
DB	0FBH, 0EFH, 0FBH, 0EFH, 0F7H, 0EFH, 0F7H, 0EFH,	
	0EFH, 0EFH, 0DFH, 06FH, 03FH, 09FH, 0FFH, 0FFH	
DB	0FDH, 0FFH, 0FEH, 0FFH, 0FFH, 07FH, 000H, 001H,	
	0FDH, 0FFH, 0FDH, 0FFH, 0FCH, 00FH, 0FDH, 0EFH	; 方
DB	0FBH, 0EFH, 0FBH, 0EFH, 0F7H, 0EFH, 0F7H, 0EFH,	
	0EFH, 0EFH, 0DFH, 06FH, 03FH, 09FH, 0FFH, 0FFH	
DB	0FFH, 05FH, 0FFH, 067H, 0FFH, 06FH, 080H, 003H,	
	0FFH, 07FH, 0FFH, 07FH, 0FFH, 07FH, 0C1H, 07FH	; 式
DB	0F7H, 0BFH, 0F7H, 0BFH, 0F7H, 0BFH, 0F4H, 0DFH,	
	0E3H, 0DDH, 08FH, 0EDH, 0DFH, 0F5H, 0FFH, 0FBH	
DB	0F9H, 0BFH, 0C7H, 0AFH, 0F7H, 0B7H, 0F7H, 0B7H,	
	0F7H, 0BFH, 000H, 001H, 0F7H, 0BFH, 0F7H, 0B7H	; 我
DB	0F1H, 0D7H, 0C7H, 0CFH, 037H, 0DFH, 0F7H, 0AFH,	
	0F6H, 06DH, 0F7H, 0F5H, 0D7H, 0F9H, 0EFH, 0FDH	
DB	0FFH, 007H, 0C0H, 06FH, 0EDH, 0EFH, 0F6H, 0DFH,	
	0C0H, 001H, 0DDH, 0FDH, 0BDH, 0FFH, 000H, 003H	; 爱
DB	0FBH, 0FFH, 0F8H, 00FH, 0F3H, 0DFH, 0F4H, 0BFH,	

	0EFH, 03FH, 09CH, 0CFH, 073H, 0F1H, 0CFH, 0FBH	
DB	0F7H, 0DFH, 0F9H, 0CFH, 0FBH, 0BFH, 0C0H, 007H,	
	0DEH, 0F7H, 000H, 007H, 0DEH, 0F7H, 0DEH, 0F7H	; 单
DB	0C0H, 007H, 0DEH, 0F7H, 0FEH, 0FFH, 000H, 001H,	
	0FEH, 0FFH, 0FEH, 0FFH, 0FEH, 0FFH, 0FEH, 0FFH	
DB	0FFH, 0BFH, 0EFH, 0BFH, 0EFH, 0BFH, 0EFH, 0BBH,	
	0E0H, 001H, 0EFH, 0FFH, 0EFH, 0FFH, 0EFH, 0FFH	; 片
DB	0E0H, 00FH, 0EFH, 0EFH, 0EFH, 0EFH, 0EFH, 0EFH,	
	0DFH, 0EFH, 0DFH, 0EFH, 0BFH, 0EFH, 07FH, 0EFH	
DB	0EFH, 0FFH, 0EFH, 007H, 0EFH, 077H, 001H, 077H,	
	0EFH, 077H, 0EFH, 077H, 0C7H, 077H, 0CBH, 077H	; 机
DB	0ABH, 077H, 0AFH, 0FFH, 06EH, 0F7H, 0EEH, 0F5H,	
	0EDH, 0F5H, 0EDH, 0F5H, 0EBH, 0F9H, 0EFH, 0FFH	
DB	0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH,	
	0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH	; 黑屏
DB	0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH,	
	0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH	
END		

主要参考文献

- [1] 李全利 编著：《单片机原理及接口技术》（第2版）。北京：高等教育出版社，2004
- [2] 张志良 主编：《单片机原理及控制技术》（第2版）。北京：机械工业出版社，2005
- [3] 张毅刚 主编：《单片机原理及应用》。北京：高等教育出版社，2006
- [4] 李光飞 编著：《单片机课程设计实例指导》（第2版）。北京：北京航空航天大学出版社，2004
- [5] 李 华 主编：《MCS-51 单片机实用接口技术》。北京：北京航空航天大学出版社，1997
- [6] 徐仁贵 主编：《微型计算机接口技术及应用》。北京：机械工业出版社，1998
- [7] 诸昌铃 编著：《LED 显示屏系统原理及工程技术》。成都：电子科技大学出版社，2000
- [8] 梅开乡 主编：《数字逻辑电路》（第2版）。北京：电子工业出版社，2005
- [9] 杨素行 主编：《模拟电子技术基础简明教程》（第2版）。北京：高等教育出版社，2004
- [10] 吉 雷 主编：《Protel99 从入门到精通》。西安：西安电子科技大学出版社，2004

课题名称	16×16 点阵 LED 电子显示屏的设计				
学生姓名	杜春雷	学号	043023133	指导教师	江玲

答辩中提出的主要问题及回答的简要情况：

一．电容 104 所代表的电容值是多少？电容单位 1pF（皮法）等于多少 F（法）？

答：电容 104 的电容值是 $10 \times 10^4 \text{ pF}$ （皮法）= $0.1 \mu \text{ F}$ （微法）， 1 pF （皮法）= 10^{-12} F （法）。

二．在电路中为什么要用到三极管 8550？

答：因为 16 个行选通信号是其中一个低电平的为有效，其余为高电平的信号无效，8550 是 PNP 型三极管，低电平信号传送至三极管的基极，则发射结正向偏置，集电结反向偏置，三极管导通，其集电极再将高电平加于 LED 阵列的对应的行上，即将此行选通；而同时其它高电平端所对应三极管截止，从而所对应 LED 行线不被选通。由于行驱动电路一条行线上要带动 16 列的 LED 进行显示，按每一 LED 器件 20mA 电流计算，16 个 LED 同时发光时，就需要 $16 \times 20 = 320 \text{ mA}$ 的驱动电流，由于三极管有电流放大作用，选用三极管 8550 作为驱动管可以在逻辑功能和驱动能力上符合 LED 的驱动要求。

三．芯片 74HC595 的端口 \overline{SRCLR} 起什么作用？

答：端口 \overline{SRCLR} 是 74HC595 芯片中移位寄存器的清零信号输入端，当其为低时移位寄存器的输出全部为 0，也就没有数据打入到输出锁存器，即 74HC595 没有列选信号输出。