

# 基于单片机实现的俄罗斯方块游戏

## MCU based Russia block game

### 摘 要

本设计是通过 AT89S52 单片机来实现俄罗斯方块游戏的设计，使用 C 语言进行编程，并通过 Proteus 来进行仿真。

本设计要实现的基本功能是：应用按键来控制方块的变换与移动；消除一行并计分；消除多行额外奖励记分，方块堆满时结束游戏等俄罗斯方块的基本功能。单片机在手持娱乐设备上的应用具有非常大的潜力，它能将其带入到一个新的阶段。为了解决外部电路图，必须充分了解所用液晶的显示方法和单片机的外部接口功能，和所需要处理的逻辑关系；还要理解 LCD 液晶的控制原理，来通过数据端口和控制端口来实现画面在液晶上面的显示。

程序则要求对外部电路了解的情况下完成自己所需要的功能，并将所要完成的功能用编程语言的形式来实现。然后通过 Protues 仿真实现通过外部按键来控制各种不同图形的方块来玩游戏，并且实现记录分数和通过过关数来改变方块下降的速度。最后将程序下载至 AT89S52 单片机并进行实际运行，实际运行表明，本设计可以实现基本的游戏功能，达到预期的目的。

**关键词：** AT89S52； C 语言； Proteus 仿真； LCD 液晶

## ABSTRACT

This design is achieved through the AT89S52 microcontroller Tetris game design, programming using C language, and carried out by proteus simulation.

The design to achieve the basic functions are: application buttons to control the box, change and movement; eliminate line and scoring; eliminate multi-line incentive points, the game ends when the box filled with basic functions such as Tetris. MCU in the application of handheld entertainment devices have great potential, it can be brought to a new stage. In order to solve the external circuit, we must fully understand the liquid crystal display method and external interface MCU features, and the need to address the logical relationship; also understand that the principle of LCD liquid crystal control to a data port and control port to achieve the above the LCD screen display.

Program requires knowledge of the external circuit where they need to complete the function and the function to be completed in the form of programming language used to achieve. Simulation then protues buttons to control an external box to play a variety of graphical games, and achieve record number of scores and the relationship had to change the speed of the module down. Finally, the program downloaded to AT89S52 MCU and the actual operation, the actual operation indicate that this design can achieve basic game functions, to achieve the desired purpose.

**Key Words:** AT89S52; C language; Proteus simulation; LCD liquid cryst

# 目 录

1	绪 论	1
1.1	课题背景	1
1.2	电子游戏及其分类	1
1.3	电子游戏发展现状	2
1.3.1	电子游戏国外发展现状	2
1.3.2	电子游戏国内发展现状	3
2	整体设计方案	5
2.1	系统的核心部分单片机	5
2.1.1	单片机的选择	5
2.1.2	AT89S52 引脚功能描述	7
2.1.3	晶振特性	9
2.2	液晶显示模块	10
2.2.1	液晶显示控制驱动器 HD61202 的特点	10
2.2.2	液晶显示控制驱动器 HD61202 的引脚功能	10
2.2.3	液晶显示控制驱动器 HD61202 的指令系统	11
2.2.4	HD61202 的软件设计	12
2.2.5	JM12864J 的电路结构特点	15
2.2.6	JM12864J 的应用	16
2.2.7	键盘电路	17
3	系统程序的设计	18
3.1	俄罗斯方块驱动流程图	18
3.2	俄罗斯方块所有绘图工作流程图	19
3.3	俄罗斯方块中按下键的流程图	20
3.4	俄罗斯方块所有绘图工作流程图	21
4	系统仿真	22
4.1	PROTUES 软件介绍	22
4.2	Keil 软件介绍	22

4.3 俄罗斯方块系统 PROTUES 仿真.....	24
5 实物的制作与调试.....	26
5.1 电路的焊接.....	26
5.2 系统性能测试与功能说明.....	27
结 论.....	28
参考文献.....	29
附录 1: C 语言程序.....	30
附录 2: 电路原理图.....	59
附录 3: 英文资料及中文翻译.....	60
致 谢.....	76

---

# 1 绪 论

计算机系统的发展已明显地朝三个方向发展；这三个方向就是：巨型化，单片化，网络化。以解决复杂系统计算和高速数据处理的仍然是巨型机在起作用，故而，巨型机在目前在朝高速及处理能力的方向努力。单片机在出现时，Intel 公司就为其单片机取名为嵌入式微控制器（embedded microcontroller）。单片机的最明显的优势，就是可以嵌入到各种仪器、设备中。这一点是巨型机和网络不可能做到的。单片机在内部已集成了越来越多的部件，这些部件包括一般常用的电路，例如：定时器，比较器，A/D 转换器，D /A 转换器，串行通信接口，Watchdog 电路，LCD 控制器等。

## 1.1 课题背景

俄罗斯方块是一款风靡全球的电视游戏机和掌上游戏机游戏，作为最经典的游戏之一，它曾造成的轰动与经济价值可以说是游戏史上的一件大事。这款游戏最初是由苏联的游戏制作人 Alex Pajitnov 制作的，它看似简单但却变化无穷，令人上瘾。相信大多数用户都还记得为它痴迷得茶不思饭不想的那个俄罗斯方块时代。

虽然用单片机来设计一个简单的俄罗斯方块游戏程序似乎有点大材小用了，但这仅仅是一个单片机在嵌入式游戏方面的简单应用，正因为他的前景无可预计，所以才有这个设计，此次设计仅仅是为了举一个单片机在游戏上应用的一个简单例子，他可以很好的说明单片机功能的强大，更高的可控性和高集成度的好处，因此它可以在电子游戏方面成为一个不可计量的明日之星。

2005 年，以计算机技术、通信技术和软件技术为核心的信息技术取得了更加迅猛的发展，加上 3C（计算机、通讯、消费电子）产业的加速融合及 3G 移动通信时代的逐步到来，嵌入式软件在国民经济各领域和日常生活中发挥了更加重要的作用。嵌入式软件的发展为几乎所有的电子设备注入了新的活力，各种装备与设备上嵌入式系统软件的广泛应用也大大地推动了其行业渗透性应用。嵌入式软件不仅提高了传统产品的技术含量，更成为产品增值的关键因素，在整个软件产业中占据了重要地位，并受到世界各国的广泛关注，如今已成为信息产业中最为耀眼的“明星”之一。

## 1.2 电子游戏及其分类

电子游戏，也就是运行在家用电脑、家用电子游戏机或是掌中宝游戏机及街机上的电子游戏程序。电子游戏是一种结合剧情故事、美术、音乐、动画、程序等技术于一身的互动型娱乐软件，涉及到多个行业。

从电子游戏的分类来看，有着多种分类方式。传统的游戏分类是按照游戏类型，

---

将其分为即时战略游戏、第一人称射击游戏、角色扮演游戏、策略型游戏等类别。根据游戏运行平台的不同，可将电子游戏分为游戏机游戏、单机版 PC 游戏、互动电视游戏、在线游戏和手机游戏。此外，韩国政府将电子游戏业分为家用游戏机游戏、单机版 PC 游戏、网络游戏和街机游戏四大类。考虑到中韩游戏业比较接近，论文采用韩国的分类标准。“网络游戏”也就是人们一般所指的“在线游戏”，是指通过互联网进行的电脑游戏，通过人与人之间的互动达到交流、娱乐和休闲的目的。在互联网（局域网）技术出现之前，电子游戏都可以归于单机版游戏，即“人机对战”游戏，或最多是双人对战游戏。而随着互联网技术的出现，通过连接游戏服务器，上百、上千、乃至上万的游戏玩家同时连线娱乐成为了现实，这就大大增加了游戏的互动性、真实性，丰富了电子游戏的内涵。从网络游戏的分类来看，目前业界还没有一个比较统一规范的分類。随着计算机设备和网络技术的发展，电子游戏在全球得到了迅猛的发展，取得了巨大的成功，已成为当今世界上不可缺少的精神粮食。

### 1.3 电子游戏发展现状

#### 1.3.1 电子游戏国外发展现状

电子游戏是最初产生于美国，1971 年，一个还在 MIT 就读的学生设计了世界上第一个业务用游戏机，这个街机游戏的名字叫《电脑空间》。可以说时至今日，电子游戏仍然是外国的天下。

美国至今仍是世界第一大游戏市场，它遵循的是以发行商为主的产业开发链模式，根据美国市场调查公司 NPD Group 调查统计，美国 2004 年游戏总销售金额（包括家用、掌上型主机及软件）为 99 亿美元，占了全球游戏市场的 45%。其中游戏软件销售的具体数字是：视频游戏软件的销售额达 52 亿美元(1.6 亿套)，PC 游戏软件销售额达 11 亿美元(4500 万套)，掌机游戏销售额达 10 亿美元(4230 万套)。2004 年一共大约有 2.48 亿套游戏售出。据估计，美国每个家庭平均购买过 2 部游戏。

日本近 20 年来疯狂赚电子游戏的钱。从上世纪 60 年代初的街机，到六七十年代之间的家用游戏机，再到八九十年代的掌上游戏机，日本经过 30 多年的耕耘，终于把电子游戏这棵“摇钱树”，培育成第一时尚娱乐产业，垄断全球业界长达 10 余年。对于日本来说，电玩业已是国家经济的重要支柱之一，在 GDP 中占有 1/5 的举足轻重地位。日本游戏业在最辉煌的 1998 年，曾经占领全球电子游戏市场硬件 90% 以上，软件 50% 以上。

韩国号称世界上网络游戏产业最发达国家之一。迄今为止，国内网络游戏市场有超过 60% 的产品来自韩国游戏软件开发商，韩国网络游戏使国内很多游戏运营公司在游戏营销市场或股票资本市场上获得了巨大的成功。韩国游戏市场的发展不是偶然，韩国政府不仅给游戏产业以正确的定位和引导，更是将其和本国经济发展紧

---

密地联系到一起，做了很多市场培育方面的工作。韩国政府认识到，靠重型工业来支撑全国经济的做法是片面的，于是将 IT、娱乐产业视为新的经济增长点，开始大力扶持。从那时起，科技含量极高而能源消耗极低的游戏制作及相关产业担负起了“富国兴邦”的重担。由文化观光部出面组建韩国游戏支援中心，向韩国游戏产业提供从资金到技术上的多方面支援；成立游戏投资联盟，政府每年向游戏产业投入的资金多达 500 亿韩元，并为游戏企业提供长期的低息贷款；设立信息化基金和文化产业基金，为游戏产业服务；对指定的风险企业实行各种税制优惠政策，减少甚至免除游戏企业的税务负担；建设游戏产业基地以扶持中小游戏企业的发展；对从事游戏产业的高科技人才免除两年的兵役。在韩国，从制作到运营，一个网络游戏公司起步门槛较低，主要得益于政府实行的援助计划。此外，在有利的政策形势下，韩国出现了众多的游戏院校，一些原本不涉足此领域的大学也开设了游戏相关专业课程，以大力培养游戏专业人才，从而为游戏生产提供了基础保障。

### 1.3.2 电子游戏国内发展现状

与电子游戏发达的国家相比，我国在电子游戏方面还存在比较大的差距。目前，我国游戏市场正处于发展阶段，但市场上的游戏软件主要来自日本、美国、韩国等地，但由本土游戏制作人开发创作的游戏正在高速增加，国产原创的游戏即将成为游戏的主流。

中国的游戏产业经过 80 年代初期台湾厂商的探索至 90 年代，开始了自己的研发、发行的历程，直至网络游戏的风行，让中国的网络游戏开始有了飞速的发展。1983 年，智冠科技有限公司在中国台湾成立，为全球第一家签订授权重制中文版产品代理销售合约的公司，2000 年智冠在台湾股票上市，2002 年其控股的中华网龙在台湾上市。

1996 年底，UBI SOFT 上海分公司暨上海育碧电脑软件有限公司成立。同年，全球游戏软件领军企业 EA 在中国上海成立办事处。1997 年，由尚洋公司制作的《血狮》正式上市。1997 年，北京新天地互动多媒体技术有限公司成立，该公司引进了《古墓丽影 III》(TombRaider III)、《盟军敢死队》等著名欧美游戏，并在 1999 年在中国率先掀起“游戏软件价格革命”，全面推行 50 元的价格体系，从此正版游戏价格开始贴近大众消费者。1999 年，业内估算中国电脑游戏市场(正版)约 1.5 亿元人民币。2000 年，大陆华彩软件代理发行第一款中文 MMORPG《万王之王》正式推出，该游戏于 1999 年在台湾发行，由台湾雷爵资讯(Lager)开发。也在 2000 年，继《万王之王》、《石器时代》、《网络三国》在台湾上市之后，游戏橘子推出了《天堂》，游戏橘子成立于 1995 年，1999 年以《便利店》在台湾奠定了地位。《天堂》这款由韩国著名游戏公司 Ncsoft 研发的网络游戏不但在本国大获成功，也在台湾再

---

次夺冠。2001年初，北京华义在中国大陆地区正式推出了《石器时代》，华义国际股份有限公司于1993年在台北成立，2001年在台湾股票上市。《石器时代》成为大陆当时最流行的网络游戏。2001年，天人互动软件技术有限公司在北京成立。当年与SEGA结成PC游戏业务的战略合作关系，发行了《樱花大战II》《文明III》《无冬之夜》等经典游戏。2001年5月，“联众世界”经过3年多的迅速成长，以同时在线17万人、注册用户约1800万的规模，成为当时世界用户数量第一的在线游戏网站。2001年11月，上海盛大代理的《传奇》正式上市，这代表着中国网络游戏的第一个巅峰，同时也引发了众多的非议。2003年9月25日，《魔兽世界》被九城、新加坡电信、搜狐和英特尔联手拿下，并凭借其优秀的品质赢得了广大玩家的青睐。



## 2 整体设计方案

### 2.1 系统的核心部分单片机

#### 2.1.1 单片机的选择

单片机微型计算机是微型计算机的一个重要分支，也是颇具生命力的机种。单片机微型计算机简称单片机，特别适用于控制领域，故又称为微控制器。

通常，单片机由单块集成电路芯片构成，内部包含有计算机的基本功能部件：中央处理器、存储器和 I/O 接口电路等。因此，单片机只需要和适当的软件及外部设备相结合，便可成为一个单片机控制系统。

20 世纪末，电子技术获得了飞速的发展，在其推动下，现代电子产品几乎渗透了社会的各个领域，有力地推动了社会生产力的发展和社会信息化程度的提高，同时也使现代电子产品性能进一步提高，产品更新换代的节奏也越来越快。时间对人们来说总是那么宝贵，工作的忙碌性和繁杂性容易使人忘记当前的时间。忘记了要做的事情，当事情不是很重要的时候，这种遗忘无伤大雅。但是，一旦重要的事情，一时的耽误可能酿成大祸。

目前，单片机正朝着高性能和多品种方向发展趋势将是进一步向着 CMOS 化、低功耗、小体积、大容量、高性能、低价格和外围电路内装化等几个方面发展。下面是单片机的主要发展趋势。单片机应用的重要意义还在于，它从根本上改变了传统的控制系统设计思想和设计方法。从前必须由模拟电路或数字电路实现的大部分功能，现在已能用单片机通过软件方法来实现了。这种软件代替硬件的控制技术也称为微控制技术，是传统控制技术的一次革命。

单片机经过 3 代的发展，正朝着多功能、高性能、低电压、低功耗、低价格、大存储容量、强 I/O 功能及较好的结构兼容性方向发展。其发展趋势不外乎以下几个方面：

##### 1. 多功能

单片机中尽可能地把所需要的存储器和 I/O 口都集成在一块芯片上，使得单片机可以实现更多的功能。比如 A/D、PWM、PCA（可编程计数器阵列）、WDT（监视定时器——看家狗）、高速 I/O 口及计数器的捕获/比较逻辑等。

有的单片机针对某一个应用领域，集成了相关的控制设备，以减少应用系统的芯片数量。例如，有的芯片以 51 内核为核心，集成了 USB 控制器、SMART CARD 接口、MP3 解码器、CAN 或者 I<sup>2</sup>C 总线控制器等，LED、LCD 或 VFD 显示驱动器也开始集成在 8 位单片机中。

## 2. 高效率和高性能

为了提高执行速度和执行效率，单片机开始使用 RISC、流水线和 DSP 的设计技术，使单片机的性能有了明显的提高，表现为：单片机的时钟频率得到提高；同样频率的单片机运行效率也有了很大的提升；由于集成度的提高，单片机的寻址能力、片内 ROM (FLASH) 和 RAM 的容量都突破了以往的数量和限制。

由于系统资源和系统复杂程度的增加，开始使用高级语言（如 C 语言）来开发单片机的程序。使用高级语言可以降低开发难度，缩短开发周期，增强软件的可读性和可移植性，便于改进和扩充功能。

## 3. 低电压和低功耗

单片机的嵌入式应用决定了低电压和低功耗的特性十分重要。由于 CMOS 等工艺的大量采用，很多单片机可以在更低的电压下工作（1.2V 或 0.9V），功耗已经降低到  $\mu\text{A}$  级。这些特性使得单片机系统可以在更小电源的支持下工作更长的时间。

## 4. 低价格

单片机应用面广，使用数量大，带来的直接好处就是成本的降低。目前世界各大公司为了提高竞争力，在提高单片机性能的同时，十分注意降低其产品的价格。

在此，我们采用型号为 AT89S52 的单片机。因为：AT89S52 是一种低功耗、高性能 CMOS 8 位微控制器，具有 8K 在系统可编程 Flash 存储器。使用 Atmel 公司高密度非易失性存储器技术制造，与工业 80C51 产品指令和引脚完全兼容。片上 Flash 允许程序存储器在系统可编程，亦适于常规编程器。在单芯片上，拥有灵巧的 8 位 CPU 和在系统可编程 Flash，使得 AT89S52 为众多嵌入式控制应用系统提供高灵活、超有效的解决方案。

与 MCS-51 单片机产品兼容、8K 字节在系统可编程 Flash 存储器、1000 次擦写周期、全静态操作：0Hz~33MHz、三级加密程序存储器、32 个可编程 I/O 口线、三个 16 位定时器/计数器八个中断源、全双工 UART 串行通道、低功耗空闲和掉电模式、掉电后中断可唤醒、看门狗定时器、双数据指针、掉电标识符。

AT89S52 具有以下标准功能：8K 字节 Flash，256 字节 RAM，32 位 I/O 口线，看门狗定时器，2 个数据指针，三个 16 位定时器/计数器，一个 6 向量 2 级中断结构，全双工串行口，片内晶振及时钟电路。另外，AT89S52 可降至 0Hz 静态逻辑操作，支持 2 种软件可选择节电模式。空闲模式下，CPU 停止工作，允许 RAM、定时器/计数器、串口、中断继续工作。掉电保护方式下，RAM 内容被保存，振荡器被冻结，单片机一切工作停止，直到下一个中断或硬件复位为止。

AT89S52 单片机有 PDIP、PLCC、TQFP 三种封装方式，在此我选用 PDIP 也就是双列直插封装的单片机此种封装有如下特点：

### 1. 适合 PCB 的穿孔安装；

2. 易于对PCB布线;
3. 操作方便;

### 2.1.2 AT89S52 引脚功能描述

AT89S52 单片机的管脚说明如图 2-1 所示:

1	P1.0	VCC	40
2	P1.1	P0.0 (AD0)	39
3	P1.2	P0.1 (AD1)	38
4	P1.3	P0.2 (AD2)	37
5	P1.4	P0.3 (AD3)	36
6	P1.5	P0.4 (AD4)	35
7	P1.6	P0.5 (AD5)	34
8	P1.7	P0.6 (AD6)	33
9	RST	P0.7 (AD7)	32
10	P3.0 (RXD)	EA/VPP	31
11	P3.1 (TXD)	ALE/PROG	30
12	P3.2 ( $\overline{\text{INT0}}$ )	PSEN	29
13	P3.3 ( $\overline{\text{INT1}}$ )	P2.7 (A15)	28
14	P3.4 (T0)	P2.6 (A14)	27
15	P3.5 (T1)	P2.5 (A13)	26
16	P3.6 ( $\overline{\text{WR}}$ )	P2.4 (A12)	25
17	P3.7 ( $\overline{\text{RD}}$ )	P2.3 (A11)	24
18	XTAL2	P2.2 (A10)	23
19	XTAL1	P2.1 (A9)	22
20	GND	P2.0 (A8)	21

图 2-1 DIP 封装 AT89S52 单片机引脚结构图

1. VCC: 电源。
2. GND: 接地端。
3. P0口 (P0.0~P0.7): P0口是一个8位漏极开路的双向I/O口。作为输出口, 每位能驱动8个TTL逻辑电平。对P0端口写“1”时, 引脚用作高阻抗输入。当访问外部程序和数据存储器时, P0口也被作为低8位地址/数据复用。在这种模式下, P0具有内部上拉电阻。在Flash编程时, P0口也用来接收指令字节; 在程序校验时, 输出指令字节。程序校验时, 需要外部上拉电阻。
4. P1口 (P1.0~P1.7): P1口是一个具有内部上拉电阻的8位双向I/O口, P1输出缓冲器能驱动4个TTL逻辑电平。对P1端口写“1”时, 内部上拉电阻把端口拉高, 此时可以作为输入口使用。作为输入使用时, 被外部拉低的引脚由于内部电阻的原因, 将输出电流(IIL)。此外, P1.0和 P1.2分别作定时器/计数器2的外部计数输入 (P1.0/T2) 和定时器/计数器2的触发输入 (P1.1/T2EX), 具体如表2-1所示:

在 Flash编程和校验时，P1口接收低8位地址字节。

表2-1 P1口第二功能

引脚号	第二功能
P1.0	T2 (定时器/计数器 T2 的外部计数输入)，时钟输出
P1.1	T2EX (定时器/计数器 T2 的捕捉/重载触发信号和方向控制)
P1.5	MOSI (在系统编程用)
P1.6	MISO (在系统编程用)
P1.7	SCK (在系统编程用)

5. P2口 (P2.0~P2.7)：P2口是一个具有内部上拉电阻的8位双向I/O口，P2输出缓冲器能驱动4个TTL逻辑电平。对P2端口写“1”时，内部上拉电阻把端口拉高，此时可以作为输入口使用。作为输入使用时，被外部拉低的引脚由于内部电阻的原因，将输出电流 (IIL)。在访问外部程序存储器或用16位地址读取外部数据存储器 (例如执行MOVX @DPTR) 时，P2口送出高8位地址。在这种应用中，P2口使用很强的内部上拉发送1。在使用8位地址 (如 MOVX @RI) 访问外部数据存储器时，P2口输出P2 锁存器的内容。在flash编程和校验时，P2口也接收高8位地址字节和一些控制信号。

6. P3口 (P3.0~P3.7)：P3 口是一个具有内部上拉电阻的8位双向I/O口，P2输出缓冲器能驱动4个TTL逻辑电平。对P3端口写“1”时，内部上拉电阻把端口拉高，此时可以作为输入口使用。作为输入使用时，被外部拉低的引脚由于内部电阻的原因，将输出电流 (IIL)。P3口亦作为 AT89S52第二功能使用，如表2-2所示：

表2-2 P3口第二功能

引脚号	第二功能
P3.0	RXD (串行输入)
P3.1	TXD (串行输出)
P3.2	$\overline{INT0}$ (外部中断 0)
P3.3	$\overline{INT0}$ (外部中断 0)
P3.4	T0 (定时器 0 外部输入)
P3.5	T1 (定时器 1 外部输入)
P3.6	$\overline{WR}$ (外部数据存储器写选通)
P3.7	$\overline{RD}$ (外部数据存储器写选通)

在 flash 编程和校验时, P3 口也接收一些控制信号。RST:复位输入。晶振工作时, RST脚持续2个机器周期高电平将使单片机复位。看门狗计时完成后, RST脚输出96个晶振周期的高电平。特殊寄存器 AUXR (地址8EH) 上的DISRTO位可以使此功能无效。DISRTO默认状态下, 复位高电平有效。

$\overline{\text{ALE}}/\overline{\text{PROG}}$ : 地址锁存控制信号 (ALE) 是访问外部程序存储器时, 锁存低8位地址的输出脉冲。在flash编程时, 此引脚 (PROG) 也用作编程输入脉冲。在一般情况下, ALE以晶振六分之一的固定频率输出脉冲, 可用来作为外部定时器或时钟使用。然而, 特别强调, 在每次访问外部数据存储器时, ALE 脉冲将会跳过。如果需要, 通过将地址为8EH的SFR的第0位置“1”, ALE操作将无效。这一位置“1”, ALE 仅在执行MOVX或MOVC指令时有效。否则, ALE将被微弱拉高。这个ALE使能标志位 (地址为8EH的SFR的第0位) 的设置对微控制器处于外部执行模式下无效。

$\overline{\text{PSEN}}$ : 外部程序存储器选通信号 (PSEN) 是外部程序存储器选通信号。当AT89S52从外部程序存储器执行外部代码时, PSEN在每个机器周期被激活两次, 而在访问外部数据存储器时, PSEN将不被激活。

$\overline{\text{EA}}/\text{VPP}$ : 访问外部程序存储器控制信号。为使能从0000H到FFFFH的外部程序存储器读取指令, EA 必须接 GND。为了执行内部程序指令, EA应该接VCC。在flash编程期间, EA也接收12伏VPP电压。

XTAL1: 振荡器反相放大器和内部时钟发生电路的输入端。

XTAL2: 振荡器反相放大器的输出端。

### 2.1.3 晶振特性

AT89S52单片机有一个用于构成内部振荡器的反相放大器, XTAL1和XTAL2分别是放大器的输入、输出端。石英晶体和陶瓷谐振器都可以用来一起构成自激振荡器。从外部时钟源驱动器件的话, XTAL2可以不接, 而从XTAL1接入, 如图2-2所示。由于外部时钟信号经过二分频触发后作为外部时钟电路输入的, 所以对外部时钟信号的占空比没有其它要求, 最长低电平持续时间和最少高电平持续时间等还是要符合要求的。如图 2-2 所示:

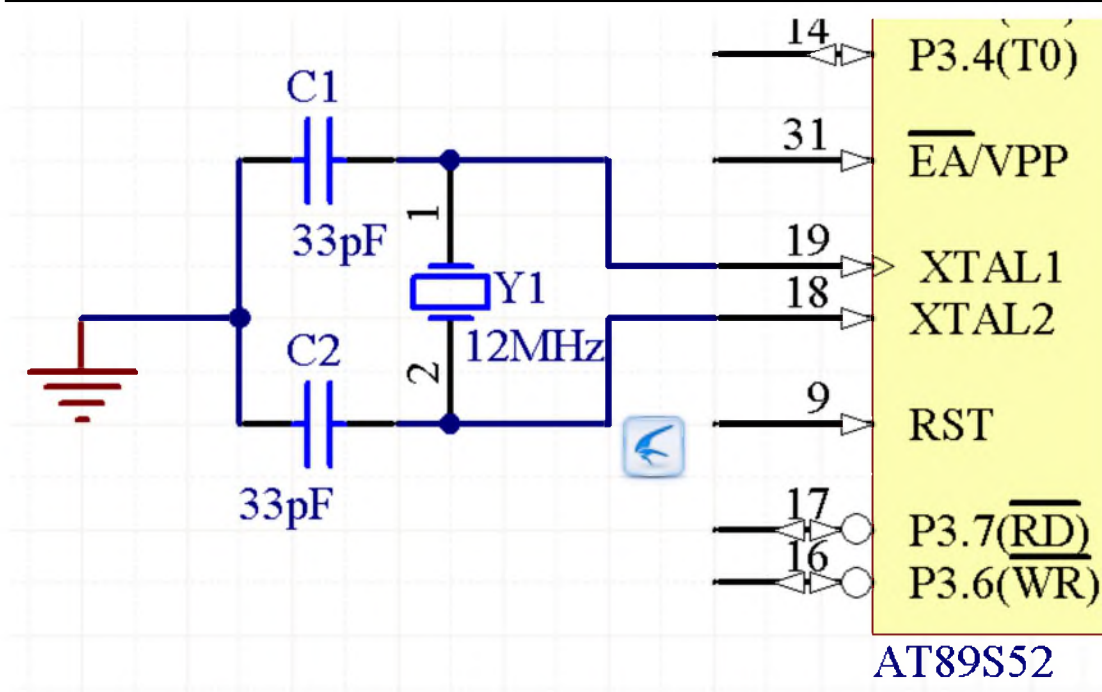


图 2-2 内部振荡电路连接图

## 2.2 液晶显示模块

显示器部分是由 HD61202 液晶显示控制驱动器和 JM12864J 液晶显示器组成的，下面我们对其分别进行介绍。

### 2.2.1 液晶显示控制驱动器 HD61202 的特点

HD61202 液晶显示控制驱动器是一种带有驱动输出的图形液晶显示控制器，它可直接与 8 位微处理器相连，它可与 HD61203 配合对液晶屏进行行、列驱动。HD61202 是一种带有列驱动输出的液晶显示控制器，它可与行驱动器 HD61203 配合使用，组成液晶显示驱动控制系统。

1. 内藏  $64 \times 64 = 4096$  位显示 RAM，RAM 中每位数据对应 LCD 屏上一个点的亮、暗状态；
2. HD61202 是列驱动器，具有 64 路列驱动输出；
3. HD61202 读、写操作时序与 68 系列微处理器相符，因此它可直接与 68 系列微处理器接口相连；
4. HD61202 的占空比为  $1/32 \sim 1/64$ 。

### 2.2.2 液晶显示控制驱动器 HD61202 的引脚功能

引脚 CS1, CS2, CS3 为芯片的片选端；引脚 E 为读写使能信号，它是在下降沿时数据被锁存入 HD61202 的；在 E 高电平期间，数据被读出；R/W 为读写选择信号，

当它为 1 时为读选通，为 0 时为写选通；DB0-DB7 为数据总线 RST 为复位信号复位信号有效时，关闭液晶显示，使显示起始行为 0，RST 可跟 MCU 相连，由 MCU 控制；也可直接接 VDD，使之不起作用。HD61202 的引脚功能如下表 2-3 所示：

表 2-3 HD61202 的引脚功能

引脚符号	状态	引脚名称	功能
CS1, CS2, CS3	输入	芯片片选端	CS1 和 CS2 低电平选通，CS3 高电平选通。
E	输入	读写使能信号	在 E 下降沿，数据被锁存（写）入 HD61202； 在 E 高电平期间，数据被读出
R/W	输入	读写选择信号	R/W=1 为读选通，R/W=0 为写选通
RS	输入	数据、指令选择信号	RS=1 为数据操作 RS=0 为写指令或读状态
DB0-DB7	三态	数据总线	
RST	输入	复位信号	复位信号有效时，关闭液晶显示，使显示起始行为 0，RST 可跟 MCU 相连，由 MCU 控制；也可直接接 VDD，使之不起作用。

### 2.2.3 液晶显示控制驱动器 HD61202 的指令系统

HD61202 的指令系统比较简单，总共只有七种。现分别介绍如下：

#### 1. 显示开/关指令

R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	1	1	1	1/0

当 DB0=1 时，LCD 显示 RAM 中的内容；DB0=0 时，关闭显示。

#### 2. 显示起始行（ROW）设置指令

该指令设置了对应液晶屏最上一行的显示 RAM 的行号，有规律的改变显示起始行，可以使 LCD 实现显示滚屏的效果。

R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	1	显示起始行（0-63）					

#### 3. 页（PAGE）置指令

R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	0	1	1	1	页号（0-7）		

显示 RAM 共 64 行，分 8 页，每页 8 行。

#### 4. 列地址（Y Address）设置指令

R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	显示列地址 (0-63)					

设置了页地址和列地址, 就唯一确定了显示 RAM 中的一个单元, 这样 MCU 就可以用读、写指令读出该单元中的内容或向该单元写进一个字节数据。

#### 5. 读状态指令

R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	BUSY	0	ON/OFF	REST	0	0	0	0

该指令用来查询 HD61202 的状态, 各参量含义如下:

BUSY: 1-内部在工作 0-正常状态

ON/OFF: 1-显示关闭 0-显示打开

REST: 1-复位状态 0-正常状态

在 BUSY 和 REST 状态时, 除读状态指令外, 其它指令均不对 HD61202 产生作用。在对 HD61202 操作之前要查询 BUSY 状态, 以确定是否可以对 HD61202 进行操作。

#### 6. 写数据指令

R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	写 数 据							

读、写数据指令每执行完一次读、写操作, 列地址就自动增一, 必须注意的是, 进行读操作之前, 必须有一次空读操作, 紧接着再读才会读出所要读的单元中的数据。

#### 7. 读数据指令

R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	读 显 示 数 据							

### 2.2.4 HD61202 的软件设计

液晶控制器 HD61202 一共有七条指令, 从作用上可分为两类, 显示状态设置指令和数据读/写操作指令。详见指令系统可查看图形液晶显示器产品有关手册。显示起始行设置中 L5~L0 为显示起始行的地址, 取值在 0-3FH(1-64 行)范围内。页面地址设置中 P2-P0 为选择的页面地址, 取值范围为 0-7H, 代表 1-8 页。列地址设置中 C5-C0 为 Y 地址计数器的内容, 取值在 0-3FH(1-64 行)范围内。

显示器上 128 点×64 点, 每 8 点为一字节数据, 都对应着显示数据 RAM(在 HD61202 芯片内), 一一对应一个 bit, 计算机写入或读出显示存储器的数据代表显示屏上某一点列上的垂直 8 点行的数据。D0 代表最上一行的点数据, D1 为第二行的点数据, ……D7 为第八行的点数据。该 bit=1 时该点则显示黑点出来, 该 bit=0 时该



点则消失。另外 LCD 指令中有一条 display ON/OFF 指令, display ON 时显示 RAM 数据对应显示的画面; display OFF 则画面消失, RAM 中显示数据仍存在。

点阵字模文件的建立:

由于 MGLS12864 液晶显示器没有内部字符发生器,所以在屏幕上显示的任何字符、汉字等须自己建立点阵字模库,然后均按图形方式进行显示。由于 HD61202 显示存储器的特性,不能将计算机内的汉字库和其它字模库提出直接使用,需要将其旋转 90 度后再写入。点阵字模库建立包括以下几个方面:

1. 建立 8×16 点阵常用字符、数字、符号字模库。

可选用计算机 BIOS 中 ASCII 的 8×16 字模库,所有字符按照 ASCII 值从小到大升序排列。

```
asm{MOV ax, 1130h /*AH=11h—功能调用。装入字库至软字库 */
mov bh, 6 /*AL=30h 取点阵信息 */
int 10h /*BH=6 取 ROM8X16 点阵指针(VGA) */
mov ax, es /*出口: ES:BP 指向字库指针 */
mov ascii_es, ax
mov ax, bp
mov ascii_bp, ax };
ascii_offset=ascii_bp+16*asciicode;
for(j=0;j<16;j++) buf[j]=peekb(ascii_es,ascii_offset+j); /*读 16 字节
点阵数据*/
for(m=0;m<16;m++) /*点阵数据转换成 LCD 格式数据*/
{ if(m<8) { beginbyte= 7; shiftn="7";}
else { beginbyte="15"; shiftn="15";}
for(j=0;j<8;j++)
ascii8x16[m]=(ascii8x16[m]+ (buf[beginbyte-j]>>(shiftn-m))&0x01)<<1;
}
```

也可选用 UCIDOS 的 ASC16 文件做字模库。ASC16 文件的字符为 8×16 点阵。所有字符按照 ASCII 值从小到大升序排列。计算字符首地址的公式: 字符首地址=字符的 ASCII 码值×16 加字模库首地址。

2. 建立所用到的 16×16 点阵字模库。汉字字符可选用 UCIDOS 的 HZK16 文件做字模库。HZK16 文件的字符为 16×16 点阵。所有字符按照区位码从小到大升序排列。计算汉字字符首地址的公式如下: 汉字首地址=((区码-1)×94+位码-1)×32。作者用 C 语言编写的读取 UCIDOS 点阵字库字模程序,完成字模读取。

数据重新排列,并按 MCS-51 汇编程序的要求写成相应格式的文本文件。

```

p=((quma-1)*94+weima-1)*32;
cclibfile="fopen"("HZK16","rb");
fseek(cclibfile,(long)p,SEEK_SET);
fread(buf,sizeof(unsigned char),32,cclibfile); /*读 32 字节点阵数据*/
for(m=0;m<32;m++){ /*点阵数据转换成 LCD 格式数据*/
if(m<8) { beginbyte="14"; shiftn="7";}
else if( m>= 8 && m<16 ) { beginbyte="15"; shiftn="15";}
else if( m>=16 && m<24 ) { beginbyte="30"; shiftn="23";}
else { beginbyte="31"; shiftn="31";}
for(j=0;j<8;j++)
hzk16x16[m]=(hzk16x16[m]+ (buf[beginbyte-2*j] >>(shiftn-m))&0x01)<<1;
}

```

3. 常用图形（如产品商标等）的点阵图形的建立。对已有的图形可采用扫描仪进行扫描，然后用图形处理软件进行处理，再将 BMP 格式文件转换成 MCS-51 的汇编文件的格式。

以上所有的字模数据都存放在单片机 W78E58 的程序存储器中，如用到的汉字、图形较多，可选用较大容量的程序存储器。

通用子程序：通用子程序分左半屏、右半屏写指令代码子程序和写显示数据子程序。液晶显示驱动器 HD16202 内部有个忙标志寄存器，当 BF=1 时，表示内部操作正在运行，不能接受外部数据或指令。下面子程序中设指令代码寄存器为 COMM，数据寄存器为 DATA。

```
(COMM EQU 20H /*指令寄存器*/ DATA EQU 21H /*数据寄存器*/)
```

#### 1. 左半屏写指令子程序

```

WR_CMD1: MOV DPTR,#CRADR1 ; /*读状态字口地址*/
WAIT1:  MOVX A,@DPTR ; /* 读状态字 */
JB ACC.7,WAIT1 ; /*判忙标志 BF,如 BF=1 忙, 等待*/
MOV DPTR,#CWADR1 ; /*写指令字口地址*/
MOV A,COMM ; /*取指令代码*/
MOVX @DPTR,A ; /*写指令代码*/
RET

```

#### 2. 左半屏写数据子程序

```

WR_DATA1: MOV DPTR,#CRADR1 ; /*读状态字口地址*/
WAIT11:  MOVX A,@DPTR ; /* 读状态字 */
JB ACC.7,WAIT11 ; /*判忙标志 BF,如 BF=1 忙, 等待*/

```

```

MOV DPTR, #DWADR1 ; /*写数据字口地址*/
MOV A, DATA ; /*取数据/
MOVX @DPTR, A ; /*写数据*/
RET

```

3. 右半屏写指令子程序 WR\_CMD2 和右半屏写数据子程序 WR\_DATA2 的编制同左半屏子程序相同，只是对应口地址不同。

### 2.2.5 JM12864J 的电路结构特点

JM12864J 是使用 HD61202 及其兼容控制驱动器作为列驱动器，同时使用 HD61203 作为行驱动器的液晶模块。由于 HD6120 不与 MPU 发生联系，只要提供电源就能产生行驱动信号和各种同步信号，比较简单，因此这里就不作介绍了。下面主要介绍一下 JM12864J 的逻辑电路图。

JM12864J 共有两片 HD61202 及其兼容控制驱动器和一片 HD61203，如图 2-3 所示：

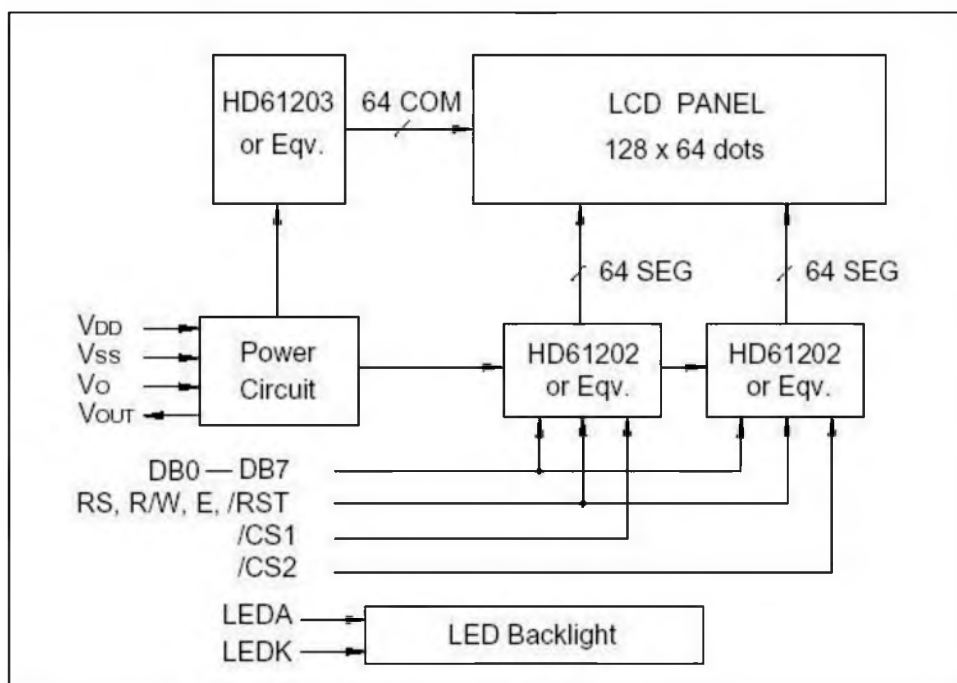


图 2-3 JM12864J 的逻辑电路图

在 JM12864J 中，两片 HD61202 的 ADC 均接高电平，RST 也接高电平，这样在使用 JM12864J 时就不必再考虑这两个引脚的作用。 $\overline{CSA}$  跟 HD61202 (1) 的  $\overline{CS1}$  相连； $\overline{CSB}$  跟 HD61202 (2) 的  $\overline{CS1}$  相连，因此  $\overline{CSA}$ 、 $\overline{CSB}$  选通组合信号为  $\overline{CSA}$ ， $\overline{CSB}=01$  选通 (1)， $\overline{CSA}$ ， $\overline{CSB}=10$  选通 (2)。对于 JM12864J，只要供给 VDD、VSS 和 V0 即可，HD61202 和 HD61203 所需的电源将由模块内部电路在 VDD 和 V0、VSS 的作用下产生。

### 2.2.6 JM12864J 的应用

以下内容以单片机 89S52 为例机的接口电路，控制电路为直接访问方式的接口电路。电路原理图如图 2-4 所示：

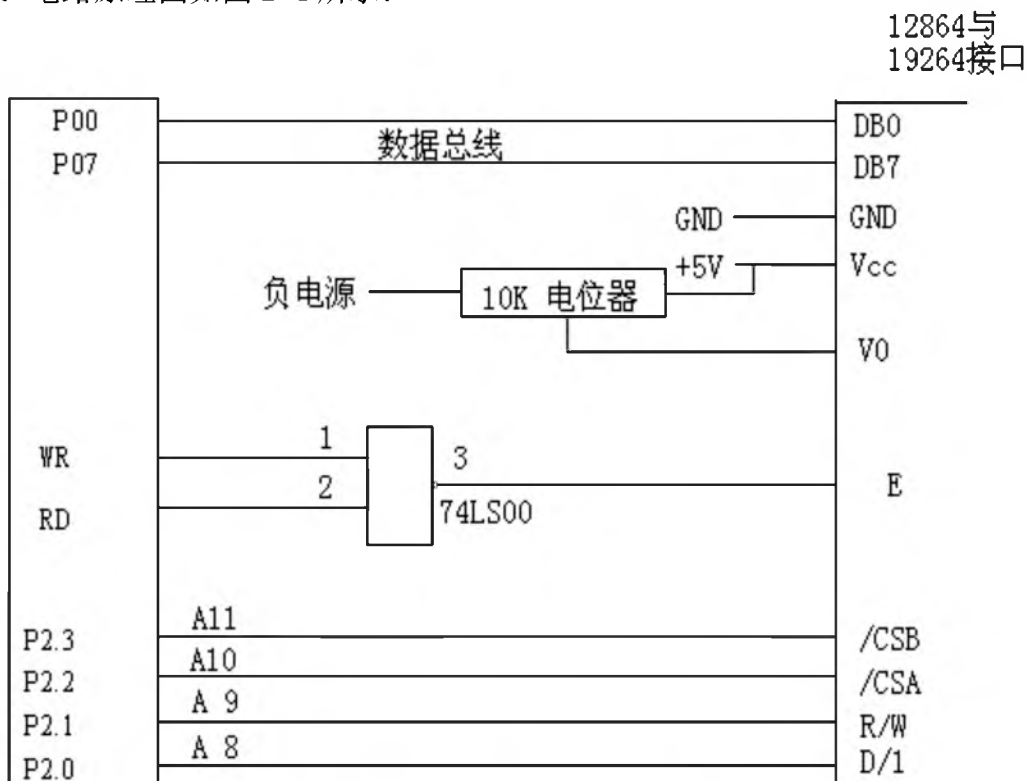


图 2-4 AT89S52 与液晶的接口电路

根据以上电路原理图中液晶的各引脚与单片机的接法，可得本设计的液晶模块电路如图 2-5 所示：

由图 2-5 可以看出：VSS 接地；数字电源 VDD 接+5V；数据、指令选择信号 RS 接单片机 P3.2 口；读写选择信号 R/W 接单片机 P3.1 口；DB0~DB7 分别接单片机的 P2.0~P2.7 口；芯片 1、芯片 2 的片选分别接单片机的 P3.4、P3.3 口；复位端 RST、背光正电源 LEDA 接+5V；液晶驱动电压 VEE、背光负电源 LEDK 接地。

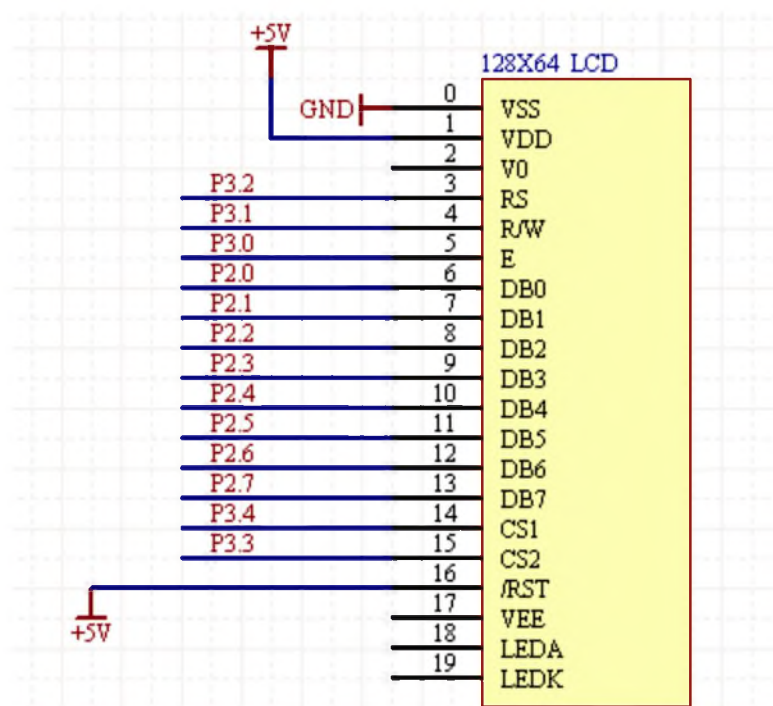


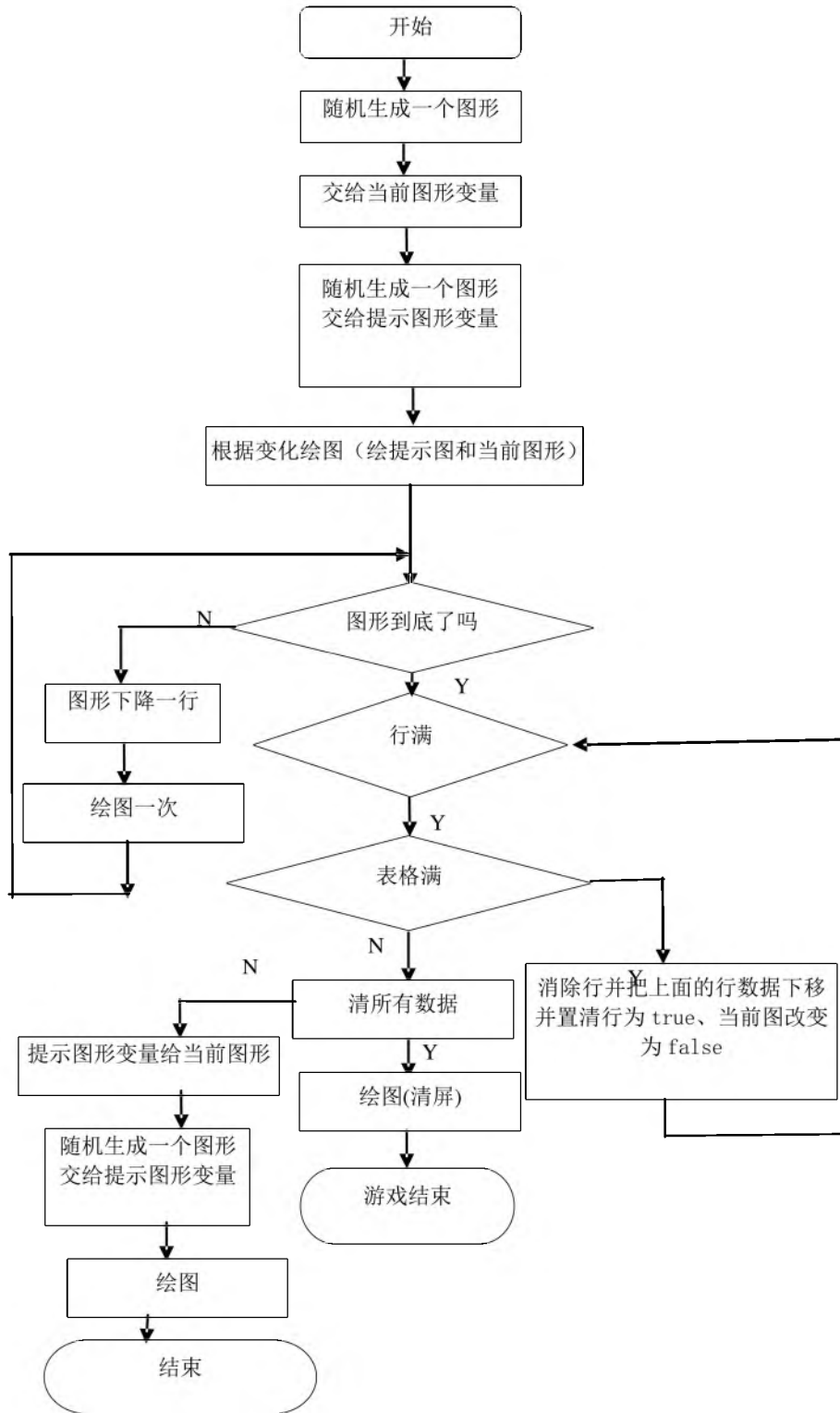
图 2-5 液晶模块电路

### 2.2.7 键盘电路

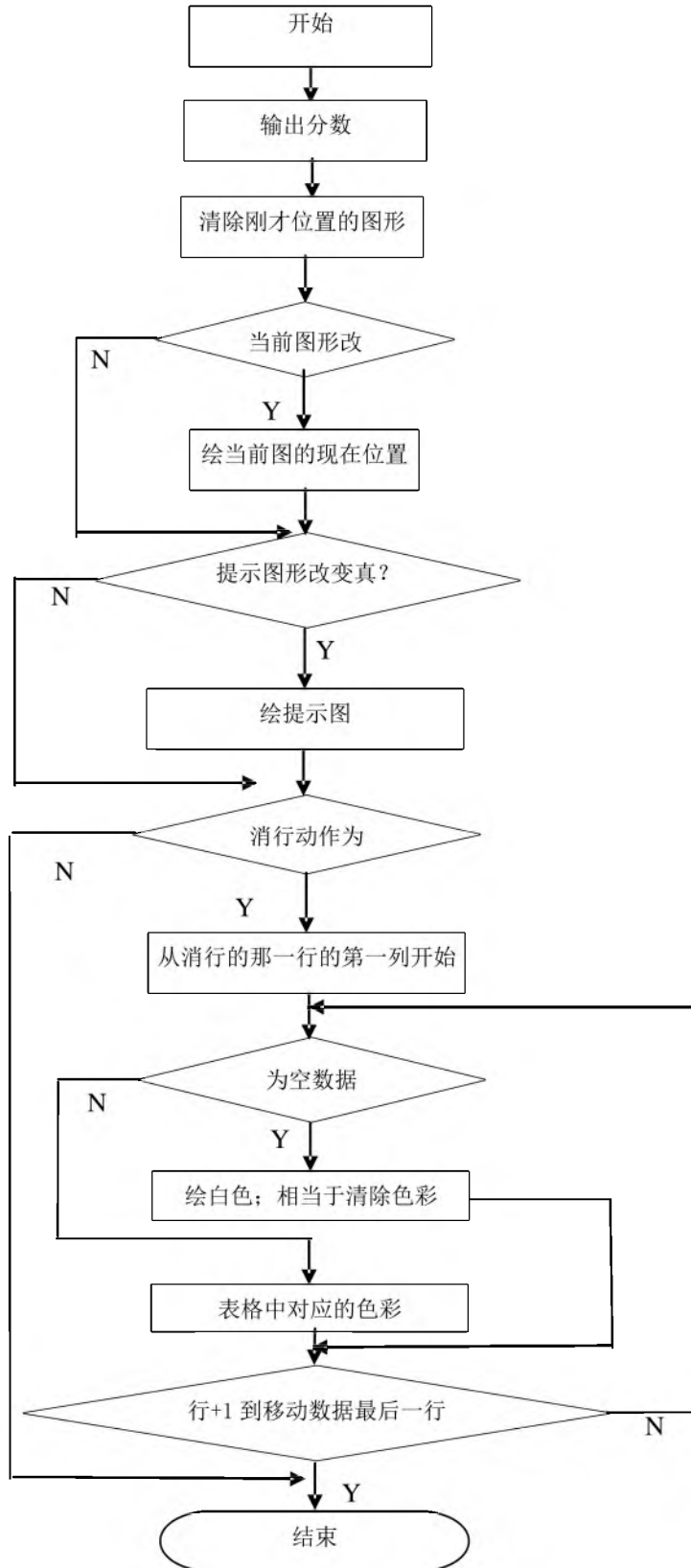
键盘电路本设计共采用按键 3 个，分别与单片机的 P1.0、P3.5、P3.6、P3.7 口相连，分别对应旋转、左移、下、右移。

### 3 系统程序的设计

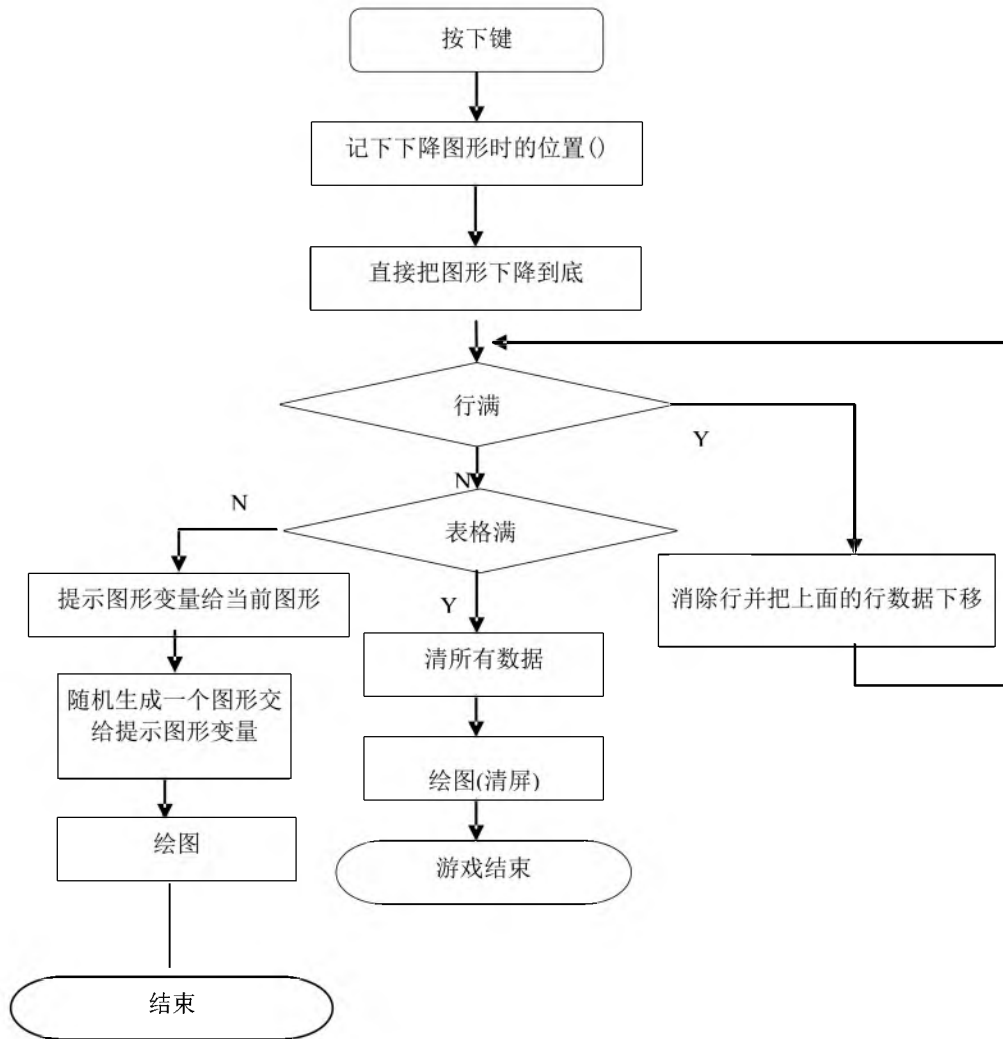
#### 3.1 俄罗斯方块驱动流程图



### 3.2 俄罗斯方块所有绘图工作流程图

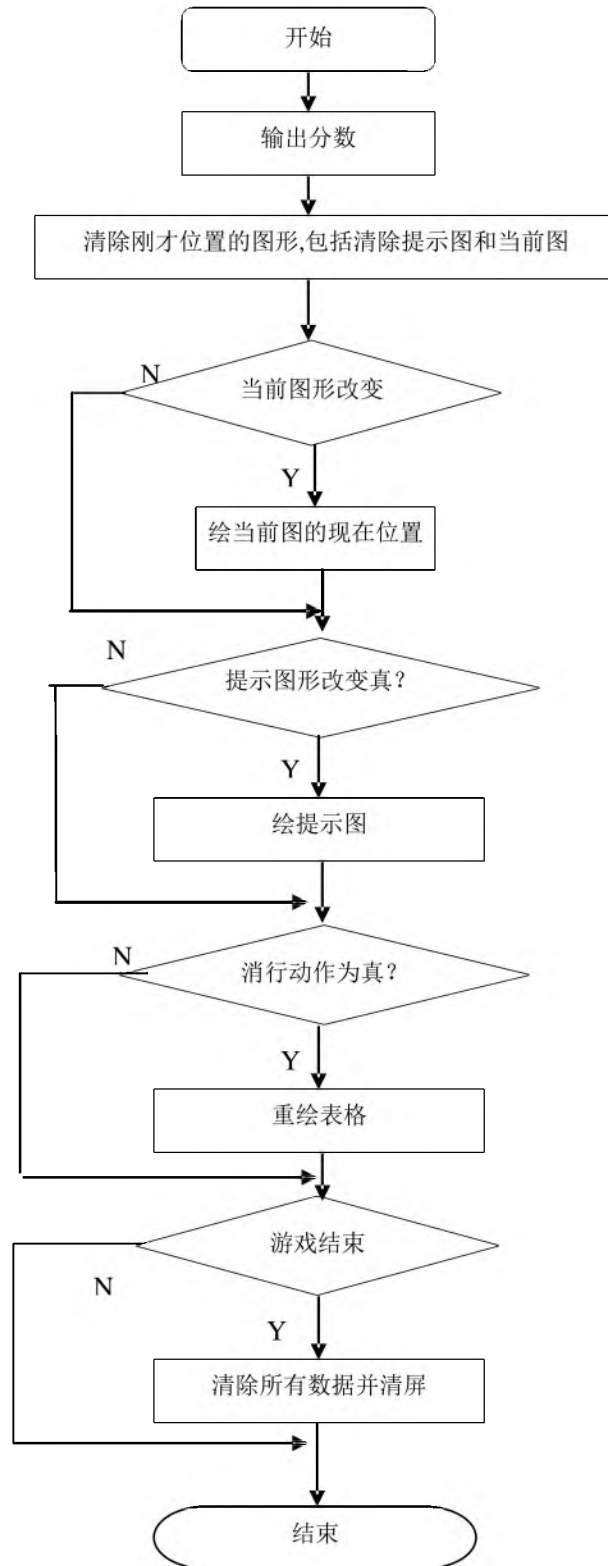


### 3.3 俄罗斯方块中按下键的流程图





### 3.4 俄罗斯方块所有绘图工作流程图



## 4 系统仿真

### 4.1 Proteus 软件介绍

Proteus 软件是来自英国 Labcenter electronics 公司的 EDA 工具软件。Proteus 软件有十多年的历史,在全球广泛使用,除了其具有和其它 EDA 工具一样的原理布图、PCB 自动或人工布线及电路仿真的功能外,其革命性的功能是,他的电路仿真是互动的,针对微处理器的应用,还可以直接在基于原理图的虚拟原型上编程,并实现软件源码级的实时调试,如有显示及输出,还能看到运行后输入输出的效果,配合系统配置的虚拟仪器如示波器、逻辑分析仪等,不需要别的,Proteus 建立了完备的电子设计开发环境!

Proteus 组合了高级原理布图、混合模式 SPICE 仿真,PCB 设计以及自动布线来实现一个完整的电子设计系统。此系统受益于 15 年来的持续开发,被《电子世界》在其对 PCB 设计系统的比较文章中评为最好产品—“The Route to PCB CAD”。Proteus 产品系列也包含了革命性的 VSM 技术,用户可以对基于微控制器的设计连同所有的周围电子器件一起仿真。用户甚至可以实时采用诸如 LED/LCD、键盘、RS232 终端等动态外设模型来对设计进行交互仿真。

其功能模块:一个易用而又功能强大的 ISIS 原理布图工具;Proteus 混合模型 SPICE 仿真;ARES PCB 设计。Proteus 仿真器的一个扩展 Proteus VSM:便于包括所有相关的器件的基于微处理器设计的协同仿真。此外,还可以结合微控制器软件使用动态的键盘,开关,按钮,LEDs 甚至 LCD 显示 CPU 模型。

1. 支持许多通用的微控制器,如 PIC, AVR, HC11 以及 8051;
2. 交互的装置模型包括:LED 和 LCD 显示,RS232 终端,通用键盘;
3. 强大的调试工具,包括寄存器和存储器,断点和单步模式;
4. IAR C-SPY 和 Keil uVision3 等开发工具的源层调试;
5. 应用特殊模型的 DLL 界面-提供有关元件库的全部文件。

### 4.2 Keil 软件介绍

Keil C51  $\mu$ Vision2 集成开发环境是 Keil Software, Inc/Keil Elektronik GmbH 开发的基于 80C51 内核的微处理器软件开发平台,内嵌多种符合当前工业标准的开发工具,可以完成从工程建立到管理、编译、链接、目标代码的生成、软件仿真、硬件仿真等完整的开发流程尤其是 C 编译工具在产生代码的准确性和效率方面达到了较高的水平,而且可以附加灵活的控制选项,在开发大型项目时非常理想。Keil C51

集成开发环境的主要功能有以下几点：

1.  $\mu$ Vision2 for Windows：是一个集成开发环境，它将项目管理、源代码编辑和程序调试等组合在一个功能强大的环境中；

2. C51 国际标准化 C 交叉编译器：从 C 源代码产生可重定位的目标模块；

3. A51 宏汇编器：从 80C51 汇编源代码产生可重定位的目标模块；

4. BL51 链接器/定位器：组合由 C51 和 A51 产生的可重定位的目标模块，生成绝对目标模块；

5. LIB51 库管理器：从目标模块生成连接器可以使用的库文件；

6. OH51 目标文件至 HEX 格式的转换器，从绝对目标模块生成 Intel Hex 文件；

7. RTX-51 实时操作系统：简化了复杂的实时应用软件项目的设计。

这个工具套件是为专业软件开发人员设计的，但任何层次的编程人员都可以使用，并获得 80C51 单片机的绝大部分应用。Keil Software 提供了一流的 80C51 系列开发工具软件，下面描述每个套件及其内容：

(1) PK51 专业开发套件：PK51 专业开发套件提供了所有工具，适合专业开发人员建立和调试 80C51 系列微控制器的复杂嵌入式应用程序。专业开发套件可针对 80C51 及其所有派生系列进行配置使用；

(2) DK51 开发套件：DK51 开发套件是 PK51 的精简版，它不包括 RTX51 Tiny 实时操作系统。开发套件可针对 80C51 及其所有派生系列进行配置使用；

(3) 编译器套件：如果开发者只需要一个 C 编译器而不需要调试系统，则 CA51 编译器套件就是最好的选择。CA51 编译器套件只包含  $\mu$ Vision2 IDE 集成开发环境，CA51 不提供  $\mu$ Vision2 调试器的功能。这个套件包括了要建立嵌入式应用的所有工具软件，可针对 80C51 及其所有派生系列进行配置使用；

表 4-1 开发套件和工具对照表

Components	PK51	DK51	CA51	A51	FR51
$\mu$ Vision2 Project Management&Editor	√	√	√	√	
A51 Assembler	√	√	√	√	
C51 Compiler	√	√	√		
BL51 Liker/locator	√	√	√	√	
LIB51 Library Magaer	√	√	√	√	
$\mu$ Vision2 Debugger/Simulator	√	√			
RTX51 Tiny	√				
RTX51 Full					√

(4) A51 汇编器套件：A51 汇编器套件包括一个汇编器和创建嵌入式应用所需要的所有工具。它可针对 80C51 及其所有派生系列进行配置使用；

(5) RTX51 实时操作系统 (FR51)：RTX51 实时操作系统是 80C51 系列微控制器的一个实时内核。RTX51 Full 提供 RTX51 Tiny 的所以功能和一些扩展功能，并且包括 CAN 通信协议接口子程序；

(6) 比较表：表 4-1 列出了每个套件的功能，表的顶上一栏为工具套件名称，表的左边一列为软件组成部分，使用这个对照表可以选择符合需要的套件。

### 4.3 俄罗斯方块系统 PROTUES 仿真

用 Proteus 软件，根据俄罗斯方块电路原理图，画出仿真图。

电路图画完后使用 Proteus 与 Keil 联调，这里使用的软件是 Keil  $\mu$ Vision3.8 与 Proteus7.4。

1. 打开 PROTEUS 的安装目录,进入 help 目录,打开 ARM.HLP 帮助文档,点击联接:Remote Debugger Drivers/Download and Install remote debugger driver for Keil uVision3, 下载 VDMAGDI.EXE, 运行安装到 KEIL 安装目录 VDM51.dll 文档会自动安装到 KEIL\C51\BIN 目录。

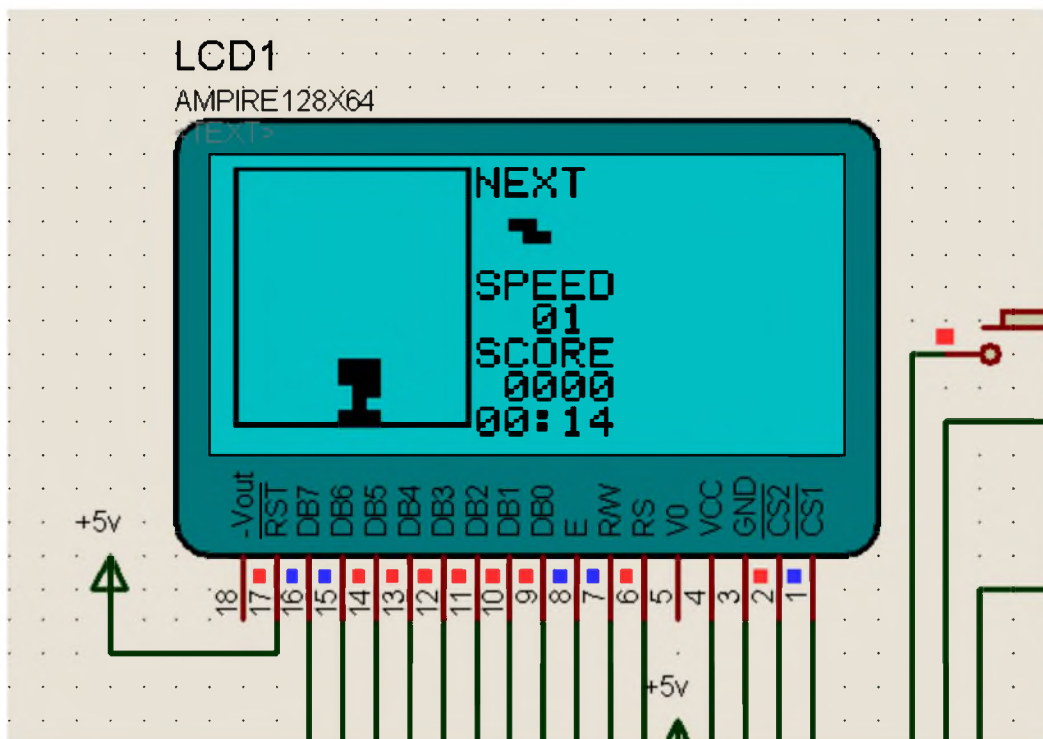


图 4-1 Protues 仿真

2. 打开 KEIL 中打开要联调的项目，在 project workspace 的“target1”上点右键/options for target'target1'。

在打开的对话框中点击到 Debug 选项卡，在右上角上先中 USE 选项，并在下拉菜单中选中 proteus VSM Simulator，在旁边的 Settings 点一下，如果是 proteus 在同一台电脑，host:127.0.0.1, port:8000, 点确定/OK 保存。

3. 打开 proteus 并打开要仿真的图纸，点击 Debug 菜单/选中 Use Remote Debug Monitor。

4. 完成，如果程序和电路图没问题，在 KEIL 中 build all 并且仿真运行后，可以在 proteus 看到实时效果了如图 4-1 所示：

经 Proteus 仿真，证明此俄罗斯方块游戏能正常运行，可以进入实物焊接阶段。

## 5 实物的制作与调试

### 5.1 电路的焊接

#### 1. 焊前准备

首先要熟悉所焊印制电路板的装配图，并按图纸配料，检查元器件型号、规格及数量是否符合图纸要求，并做好装配前元器件引线成型等准备工作，由于制作的是便携式的俄罗斯方块游戏机，所以在这里我选用了电池来做电源。

#### 2. 焊接顺序

元器件装焊顺序依次为：电阻器、电容器、二极管、三极管、集成电路、大功率管，其它元器件为先小后大。

#### 3. 对元器件焊接要求

##### (1) 电阻器焊接

按图将电阻器准确装入规定位置。要求标记向上，字向一致。装完同一种规格后再装另一种规格，尽量使电阻器的高低一致。焊完后将露在印制电路板表面多余引脚齐根剪去。

##### (2) 电容器焊接

将电容器按图装入规定位置，并注意有极性电容器其“+”与“-”极不能接错，电容器上的标记方向要易看可见。先装玻璃釉电容器、有机介质电容器、瓷介电容器，最后装电解电容器。

##### (3) 二极管的焊接

二极管焊接要注意以下几点：第一，注意阳极阴极的极性，不能装错；第二，型号标记要易看可见；第三，焊接立式二极管时，对最短引线焊接时间不能超过 2S。

##### (4) 三极管焊接

注意 e、b、c 三引线位置插接正确；焊接时间尽可能短，焊接时用镊子夹住引线脚，以利散热。焊接大功率三极管时，若需加装散热片，应将接触面平整、打磨光滑后再紧固，若要求加垫绝缘薄膜时，切勿忘记加薄膜。管脚与电路板上需连接时，要用塑料导线。

##### (5) 集成电路焊接

首先按图纸要求，检查型号、引脚位置是否符合要求。焊接时先焊边沿的二只引脚，以使其定位，然后再从左到右自上而下逐个焊接。

对于电容器、二极管、三极管露在印制电路板面上多余引脚均需齐根剪去。

## 5.2 系统性能测试与功能说明

焊接完成后，将电源接通，游戏画面出现后，分别按下左、右、下、旋转等按钮，均能正常在屏幕上做出反应。当方块堆满一行时，方块正确消除，并将分数累计在屏幕右侧，同时消除多行方块时，分数有额外奖励。达到一定分数时，方块下落速度加快，难度提高。当屏幕中方块有触及顶端的时候，游戏结束。

## 结 论

通过半个学期地全心投入和辛苦努力,终于完成了基于单片机的俄罗斯方块游戏的设计,设计的过程就是综合运用所学知识和学习新知识的过程。由于经验不足,难免出现一些小的失误,但无论怎样,本次毕业设计对于我来说都是一次难得的锻炼机会,使我积累了许多宝贵的经验。

本文对单片机制作的俄罗斯方块游戏的运行原理进行了分析,全面、详细地阐述了俄罗斯方块游戏的设计过程。本游戏机已经达到了掌上娱乐的要求,但在设计上还留有进一步改善的余地,如可以美化游戏界面、加入背景和动作音乐、力回馈系统等。

毕业设计的整个开发过程是曲折的,首先在硬件设计上,由于开始对 LCD 液晶屏不熟悉,加之其程序量较多,所以,在设计的过程中,不免有些困难,通过我多方面的查找资料,并不断的向老师和同学请教,结合网络资源,最终找出比较完善的方案,在理论上能够实现功能要求。经过几个月的前期学习,我积累了很多的关于液晶显示方面和 C 语言设计方面的资料,并对电路的整合有了一个基本的概念,最后设计出总的电路图,经过不断的测试与修改,最终完善了硬件电路的设计。

对于软件设计,因为以前的编程经验不够,再加上对 AT89S52 这一芯片的了解不是很彻底,因此,在这方面花费了很多的精力和时间。当然这个过程不乏有同学的帮助,我要非常感谢实验室同学的帮助在我的软件调试过程中给了我至关重要的帮助,并给了我很多非常好的建议。

当然由于我在理论和实践方面存在一定的不足,所以在设计思路和实现功能上难免有不足和没有想到的地方,还请各位老师给予指正。



## 参考文献

- [1] 张齐, 杜群贵. 单片机应用系统设计技术——基于 C 语言编程[M]. 北京: 电子工业出版社, 2004.
- [2] Apress. Introduction to Visual Basic Programming [M]. Professional Computing and Web Design, 2006.
- [3] 李华. 单片机原理与接口技术, 北京: 清华大学出版社, 33—80.
- [4] 周润景著. Proteus 在 MCS-51&ARM7 系统中的应用百例[M]. 北京: 电子工业出版社, 2006.
- [5] 游安弼, 李玉岐. 基于 JavaBean 组件模型的俄罗斯方块[R]. 北京: 全国第四届 Java 技术及应用学术会议, 2001.
- [6] Jason Lam. J2ME&Gaming[M]. Addison Wesley, 2004.
- [7] 高凌琴, 陈青华. 俄罗斯方块游戏关键技术探讨[J]. 信息技术与信息化. 2008.
- [8] 荣钦科技编著, 《Java 2 游戏设计》[M]. 北京: 清华大学出版社, 2004.
- [9] 陈立伟, 张克非, 黎秀红著. 精通 JAVA 手机游戏与应用程序设计[M]. 北京: 中国青年出版社, 2005-5.
- [10] 谭浩强. C 程序设计. [M] 北京: 清华大学出版社, 2003.
- [11] 吴金戌, 沈庆阳. 8051 单片机实践与应用, 北京: 清华大学出版社, 2003.
- [13] 李宏, 张家田. 液晶显示器件应用技术, 北京: 机械工业出版社, 2004.
- [13] 公茂法. 单片机人机接口实例集, 北京: 北京航空航天大学出版社, 1998.
- [14] 何立明. 单片机高级教程——应用与设计, 北京: 北京航空航天大学出版社, 2000.
- [15] 李广弟, 朱月秀. 单片机基础, 北京: 北京航空航天大学出版社, 2001.
- [16] 楼然苗, 李光飞. MCS-51 系列单片机设计实例, 北京: 北京航空航天大学出版社, 2001.
- [17] 杨元强. 酷比杯获奖作品选登《LEGO 俄罗斯》[J]. “酷比杯”游戏策划大赛, 2007.

## 附录 1：C 语言程序

```
#include <AT89X51.H>
#define uchar unsigned char
#define uint unsigned int
#define DOWNTIME 30
#define MAXHANG 20
#define MAXLIE 16
#define MAXPIX 3
#define PUSHON 50

#define LCD P2
#define EN P3_0
#define RW P3_1
#define RS P3_2
#define CS1 P3_3
#define CS2 P3_4

#define KEYLEFT P3_5
#define KEYDOWN P3_6
#define KEYRIGH P3_7
#define KEYROTATION P1_0

uchar gkey=0xff,keystate=0,t0ms1=0,t0ms=0,downtimegap=0;
uchar miao=0,fen=0;
uchar downok;
bit keyflag,timeupdate,fashionupdate;
uchar idata cubeMap[MAXHANG][2];
typedef struct{
    uchar code * box;
    uchar cube : 4;
    uchar state : 4;
    char row;
```

```

char column;
} block;

block this;

uint score=0;
uchar speed=1;
uchar code bittable[8]={1,2,4,8,0x10,0x20,0x40,0x80};
uchar code cube[]=
{
/* ■
   ■ ■ ■
*/
0,4,0xe,0, 0,2,6,2, 0,7,2,0, 4,6,4,0,

/* ■
   ■ ■ ■
*/
0,8,0xe,0, 0,4,4,0xc, 0,0,0xe,2, 0,6,4,4,
/* ■ ■ ■
   ■
*/
0,0xe,8,0, 0,4,4,6, 0,1,7,0, 6,2,2,0,
/* ■ ■
   ■ ■
*/
0,0xc,6,0, 0,2,6,4, 0,6,3,0, 2,6,4,0,
/* ■ ■
   ■ ■
*/
0,6,0xc,0, 0,4,6,2, 0,3,6,0, 4,6,2,0,
/* ■ ■ ■ ■
*/
0,0xf,0,0, 4,4,4,4, 0,0,0xf,0, 2,2,2,2,
/* ■ ■
   ■ ■

```

```
*/
0,6,6,0,    0,6,6,0,    0,6,6,0,    0,6,6,0
};
uchar code asii[]=
{
    0x3E,0x51,0x49,0x45,0x3E, // -0-
    0x00,0x42,0x7F,0x40,0x00, // -1-
    0x62,0x51,0x49,0x49,0x46, // -2-
    0x21,0x41,0x49,0x4D,0x33, // -3-
    0x18,0x14,0x12,0x7F,0x10, // -4-
    0x27,0x45,0x45,0x45,0x39, // -5-
    0x3C,0x4A,0x49,0x49,0x31, // -6-
    0x01,0x71,0x09,0x05,0x03, // -7-
    0x36,0x49,0x49,0x49,0x36, // -8-
    0x46,0x49,0x49,0x29,0x1E, // -9-
    0x00,0x36,0x36,0x00,0x00, // -:-10
//next
    0x7F,0x04,0x08,0x10,0x7F, // -N-11
    0x7F,0x49,0x49,0x49,0x41, // -E-12
    0x63,0x14,0x08,0x14,0x63, // -X-13
    0x01,0x01,0x7F,0x01,0x01, // -T-14
//speed
    0x26,0x49,0x49,0x49,0x32, // -S-15
    0x7F,0x09,0x09,0x09,0x06, // -P-16
    0x7F,0x49,0x49,0x49,0x41, // -E-17
    0x7F,0x41,0x41,0x41,0x3E, // -D-18
//score
    0x3E,0x41,0x41,0x41,0x22, // -C-19
    0x3E,0x41,0x41,0x41,0x3E, // -O-20
    0x7F,0x09,0x19,0x29,0x46, // -R-21
    0x00,0x00,0x00,0x00,0x00, // - -22
//GAME OVER
    0x3E,0x41,0x51,0x51,0x72, // -G-23
    0x7C,0x12,0x11,0x12,0x7C, // -A-24
```

```
    0x7F,0x02,0x0C,0x02,0x7F, // -M-25
    0x1F,0x20,0x40,0x20,0x1F, // -V-26
//TIME
// 0x00,0x41,0x7F,0x41,0x00 // -I-27
};
/////////////////////////////////////////////////////////////////
void lcdCmd(uchar cmd)
{
    bit ea;
    ea=EA;
    EA=0;
    EN=0;
    RW=0;
    RS=0;
    LCD=cmd;
    EN=1;
    EN=1;
    EN=0;
    EA=ea;
}
//-----
void lcdWriteByte(uchar ch)
{
    EN=0;
    RS=1;
    RW=0;
    LCD=ch;
    EN=1;
    EN=1;
    EN=0;
}
//-----
void lcdSetPage(uchar page)
{
```

```
    page &=0x7;
    page +=0xb8;
    lcdCmd(page);
}
//-----
void lcdSetColumn(uchar column)
{
    column &=0x3f;
    column +=0x40;
    lcdCmd(column);
}
//-----
//character fron=5*8
void lcdPlayChar(uchar index,uchar page,uchar colume)
{
    uchar i,temp;
    uint p;
    p=5*index;
    for(i=colume;i<colume+5;i++)
    {
        if(i<64)
        {
            CS1=1;
            CS2=0;
            temp=i;
        }
        else
        {
            CS1=0;
            CS2=1;
            temp=i-64;
        }
        lcdSetPage(page);
        lcdSetColumn(temp);
    }
}
```

```
        lcdWriteByte(asii[p++]);
    }
}
//-----
//rectangle(3,0,50,60)
void rectangle(void)
{
    uchar i,page;
    CS1=1;
    CS2=0;
    lcdSetPage(0);
    lcdSetColumn(2);
    EN=0;
    RS=1;
    RW=0;
    LCD=0xff;
    EN=1;
    EN=1;
    EN=0;
    for(i=3;i<51;i++)
    {
        EN=0;
        RS=1;
        RW=0;
        LCD=0x1;
        EN=1;
        EN=1;
        EN=0;
    }
    EN=0;
    RS=1;
    RW=0;
    LCD=0xff;
    EN=1;
```

```
    EN=1;
    EN=0;
//-----
for(page=1;page<7;page++)
{
    lcdSetPage(page);
    lcdSetColumn(2);
    EN=0;
    RS=1;
    RW=0;
    LCD=0xff;
    EN=1;
    EN=1;
    EN=0;
    for(i=3;i<51;i++)
    {
        EN=0;
        RS=1;
        RW=0;
        LCD=0x0;
        EN=1;
        EN=1;
        EN=0;
    }
    EN=0;
    RS=1;
    RW=0;
    LCD=0xff;
    EN=1;
    EN=1;
    EN=0;
}
//-----
    lcdSetPage(7);
```



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/337053156042006056>