

中国传媒大学

电子设计竞赛设计报告

题 目：右脑锻炼小游戏设计
姓 名：马慧（201010013245）
院 系：信息工程学院
专 业：数字媒体技术
指导教师：杜伟韬

目录

- 一、方案简介.....2
- 二、游戏介绍.....3
- 三、硬件平台介绍.....3
- 四、模块设计.....5
 - 1、LCD 显示模块5
 - 2、软件设计流程.....6
 - 3、图形的随机显示.....6
 - 4、图形的随机消失.....7
 - 5、游戏继续与失败的判断.....7
 - 6、键值的获取与处理.....7
 - 7、图形个数增加的显示.....7
 - 8、游戏结束后的重新开始及数据初始化.....8
- 五、测试效果图.....8
- 六、设计总结.....11
- 七、参考文献.....12

一、方案简介

电路基于 P89V51RD2 单片机和 LM16032 LCD 阵列驱动芯片的游戏电路。通过控制单片

机程序来处理游戏中图形的产生与消失，内置 ST7920 芯片的 LM16032 LCD 阵列显示图形。电路复位，依次显示“燃烧吧，右脑运动会”和“马慧制作”字样，然后提示“按下 OK 键开始游戏”。游戏失败后再按下“OK”键可以重新开始。本设计使用 Keil C51 进行开发，编程语言为 C51。

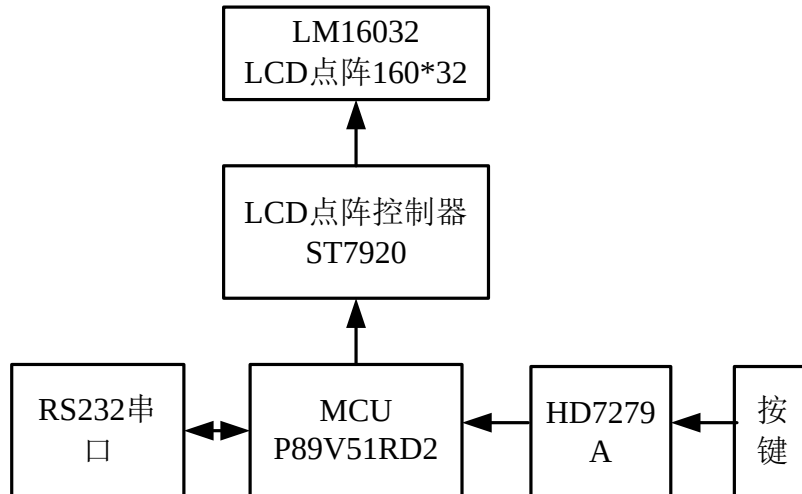
二、游戏介绍

游戏开始后，先随机出现 4 个表情字符，5 秒钟之后屏幕闪烁示意时间到，然后随机消失一个图形，玩家选出消失的图形是哪一个，在键盘上按下响应的键。如果选择正确，游戏继续进行，当答对的总题数大于 3 时，开始增加每次出现的图形数量，由于屏幕大小的限制，最多一次性出现六个图形。图形的出现具有随机性，可能在一次显示中有重复的图样，但是消失的时候，程序保证只会消失一个图形。如果玩家选择错误，屏幕上出现“再接再厉，锻炼右脑人人有责”的字样，按下“OK”键可以重新开始游戏。

三、硬件平台介绍

实验平台由电赛辅导组提供，外观如下图所示：





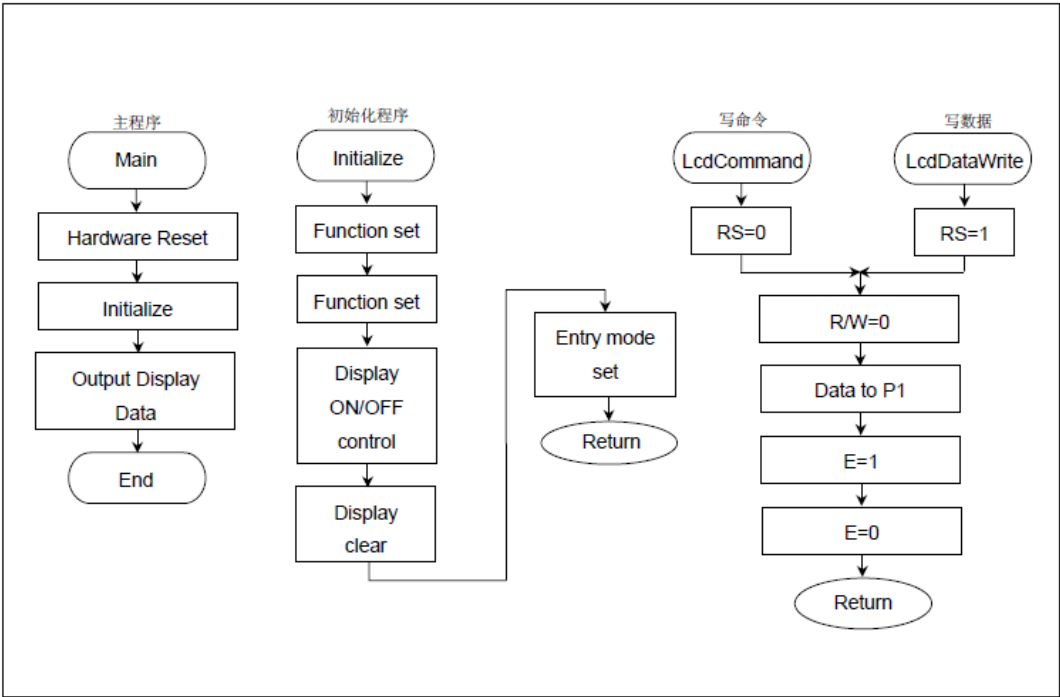
本设计单片机采用 NX(原 Philips)公司的 P89V51RD2，它是一款 80C51 微控制器，包含 64kB Flash 和 1024 字节的数据 RAM。Flash 程序存储器支持并行和串行在系统编程 (ISP)。可使用软件通过电脑串口对器件进行重复编程 (烧录)。

LCD 屏采用 LM16032 系列点阵图形式液晶显示模块，采用 ST7920 控制器，模块点阵数 160*32dot。有 4bit 和 8bit 两种规格的并口数据传输，内带 8192 个标准 GB 码的简体中文，126 个字母符号字型。实验中采用 8bit 并口数据传输，LCD 的数据\指令选择 RS、读\写 R\W、使能 E 端分别和单片机 P1.0、P1.2、P1.3 相连。

键盘由 HD7279A 串行接口智能控制芯片控制，单片就完成了键盘接口的全部功能，对于键盘驱动功能来说，还内含了去抖动电路。实验板中芯片的片选 CS、同步时钟输入端 CLK、串行数据输入输出端 DATA、按键有效输入端分别和单片机 P3.5、P3.4、P1.5、P1.4 相连。

四、模块设计

1、LCD 显示模块



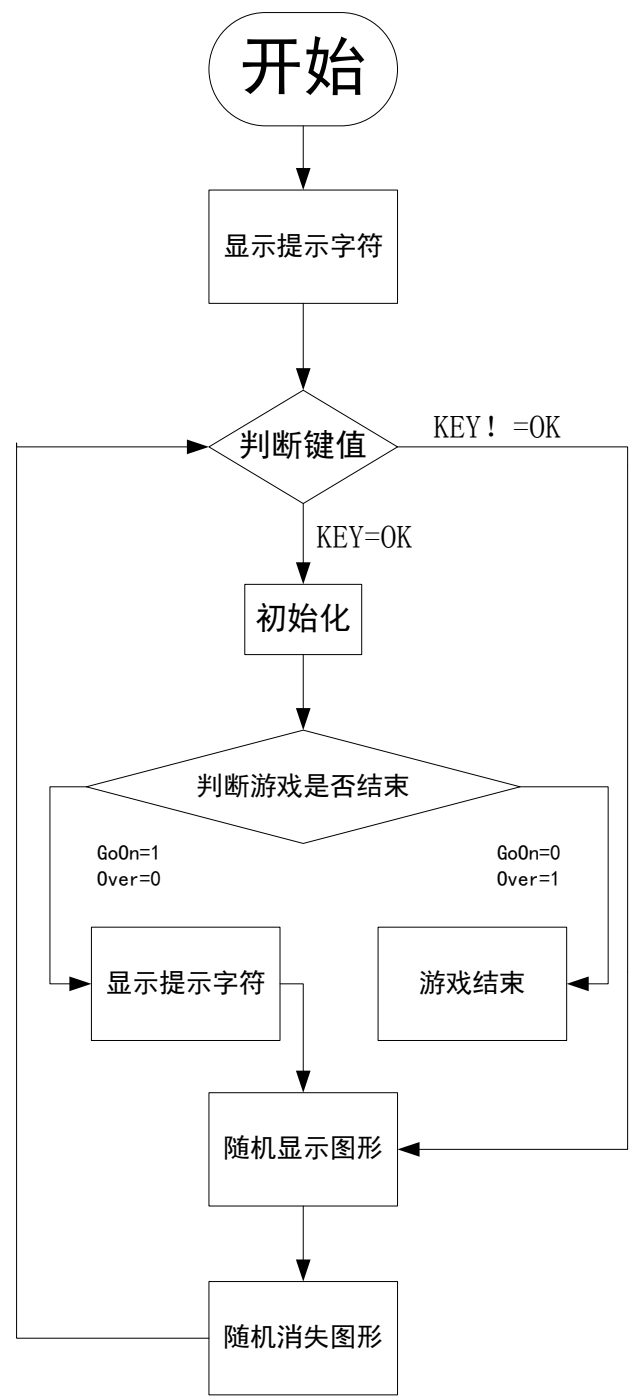
LM16032 软件编程流程图

在 LCD 显示的程序当中，只采用写入数据和写入寄存器命令的信号组合，不需要从 RAM 中读出数据做处理。写入数据或者命令时，需要用到的引脚有 E,R\W,RS。实验中用到的 8 位并行时序接口信号组合功能如下表：

操作	RS	R\W	E	功能说明
写寄存器命令	0	0		写指令到指令寄存器
写数据操作	1	0		写显示数据

在游戏的过程中，需要有大约 2.5s 的延时，然后屏幕闪动两次示意玩家时间思考时间快到了，屏幕的闪动设定就采用了写寄存器命令操作，将屏幕背光关闭后迅速开启。而文字的操作提示和游戏过程中表情符号的显示都采用了写数据操作，使用循环变量控制好表情或者文字的显示位置。

2、软件设计流程



3、图形的随机显示

在程序中所有对图形的操作，都是对图形索引 `ShapeIndex` 的操作。因为游戏的设计中一共有 6 个选择按键，也就是有 6 种样式的图形，但是在游戏初期只有 4 个图形随机出现，并且出现的图形由随机数产生函数 `rand()` 直接生成索引，可能会有重复，这些索引编号保存在缓存中开辟的一块专门存储本次要显示图形的索引集合 `ShapeIndex[6]` 中。其实每次显示

的时候都是在主函数中直接调用 LM16032 LCD 的写入数据命令，用两个循环变量设置好显示位置，switch 函数根据索引编号将相应图形的指针写入。

另外程序后期加上了一个显示图形的最大数量 max 设定，随着玩家答对的题目数量增多，一次性显示的图形数量也会增加，由于屏幕大小的限制，目前最多可以同时显示 6 个图形。

4、图形的随机消失

图形的随机消失有两个前提：1、消失的图形必须是之前显示的图形中的一个；2、当有重复的图形出现时，只会随机消失 1 个。

对于前一个条件，采用 `DisappearIndex=ShapeIndex[(rand()%4)]` 的方式，就保证了消失的一定是出现的图形。而后一个条件使用一个循环语句进行判断，当检测到有 `if(ShapeIndex[i]==DisappearIndex)` 时就用 `break` 跳出循环，对以后的图形不再继续进行判断。在程序中，采用把显示集合 `ShapeIndex[6]` 中要消失的图形索引置零的方式，在后期 `main` 函数中如果检测到 0，直接 `break` 和进行坐标循环变量的增加，来实现随机消失。

5、游戏继续与失败的判断

（a）游戏继续判断

当程序判定消失的图形索引号和玩家选择的图形一致时，继续标志 `GoOn=1`，并且回到判断键值的循环。

（b）游戏失败判断

相反，当程序判定消失的图形索引号和玩家选择的图形不一直时，结束标志 `Over=1`，按“OK”键可以重新开始游戏，但是跳过了最初的文字提示阶段。

6、键值的获取与处理

程序中使用 `ReadKey()` 函数将相应键转换成字符 ASCII 码，首先需要从 HD7279A 中读出键码，把十六进制的键码用一个 `switch` 函数转换成字符，保存在 `key[1]` 这个数组中。之后的所有比较都是将 `DisappearIndex` 这个随机数和 `key[0]` 的值进行比较来判断游戏的结束和继续。

7、图形个数增加的显示

图形的显示在主程序中进行，由于之前的随机显示 `void RandomAppear(void)` 会将 `ShapeIndex[6]` 这个数组中的元素赋值为 1-6 的随机数，只要设定好显示位置的循环变量，规定当显示的图像个数 max 为 4 时，显示为 2 行 2 列 4 个图像，max 为 6 时，显示为 2 行 3 列 6 个图像就行。经过一定时间的延时和 LCD 短时开关的提示后，`void RandomDisappear(void)` 函数会将 `ShapeIndex[6]` 这个数组中任意一个元素赋值为 0，并将消失的索引赋值给 `DisappearIndex`，清屏后再次进行上面过程的显示，只是 `switch` 检测到 0 时只进行位置变量的改变，不显示图形，就可以出现“消失”的效果，并且随着玩家答对的数量增多图形增多。

8、游戏结束后的重新开始及数据初始化

设置游戏初始化按键 OK，无论在刚刚复位后显示字符表情的过程中还是在游戏结束后，按下“OK”键，游戏都会重新开始。

游戏数据初始化后玩家答对的题数 count 归零，键盘进行初始化，游戏继续标志 GoOn=1，游戏结束表示 Over=0。游戏不设置答对题数的上限，count>3 的时候会逐渐增加一次出现的图像数量 max 的大小，max 最大为 6。

五、测试效果图



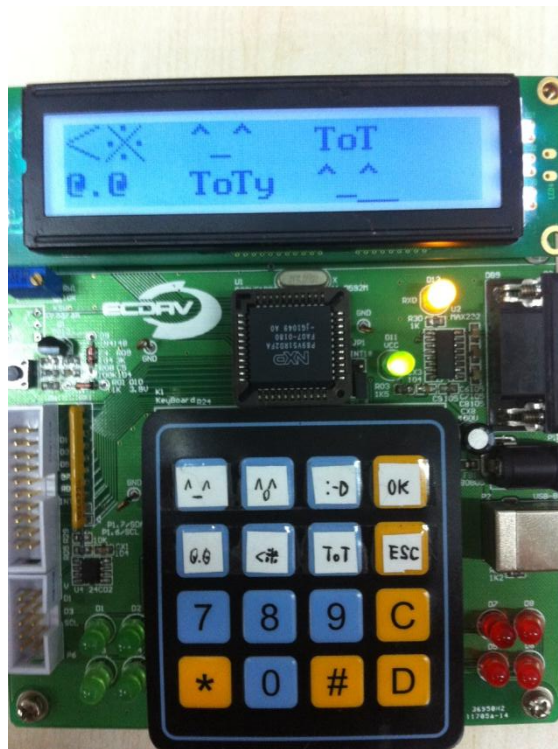
开机显示文字



文字提示开始游戏



按下“OK”键后游戏开始



游戏进行中 (1)



游戏进行中 (2)



游戏失败

六、设计总结

在设计和调试、测试过程中，遇到了一些问题，记录和总结如下：

1、在随机产生索引编号的过程中，首先想到随机数的产生用定时器 0，工作方式 1 来产生。设定中断方式为开启定时器中断，这样中断函数中还能将定时器重新赋值，继续定时。但是后来发现因为实验中的随机数是个一维变量，晶振频率又很高，只用 TL%6 或者 TH%6 的方式，每次产生的索引都是同一个数，达不到随机的效果。后来查阅了 C51 的随机数产生函数，直接加入<stdlib.h>的系统头文件，使用 void rand(void)函数来产生随机性更大的数。

2、在程序中设定了一个玩家答对题目数的计数变量 count，在最后游戏失败想显示：“您一共答对了 x 题”，但是变量 count 始终显示不正确，因为 LM16032 规定当 16*16GB 中文字符和 8*16 半宽字符混合时，GB 中文字符的高位只可出现在每一地址计数器 AC 的高位 H 上，半宽字符为每 8 个点为十六进制一个数表示，设置 GDRAM 时先设置垂直地址范围，再设定水平地址范围，垂直地址和水平地址要进行异或运算。

3、在 debug 过程中最不知道为什么但是调对的 error 是变量 char DisappearIndex 和用户按键 unsigned char key 的比较，需要判定当两个数一样的时候，游戏继续标志 GoOn=1，游戏结束标志 Over=0，但是比较的过程始终不正确，最初的试验中随便按键都会显示游戏结束，后来把判断条件改为 if((key[0]- '0') !=DisappearIndex)，判断过程正确，游戏达到预期目的。

4、这个游戏如果把随机消失一个图形改成随机产生一个图形，就是“打地鼠”游戏了。

游戏的构思和软件设计还有很多的提升空间。由于开始得比较晚，从看资料学习 C51 单片机编程到程序主体完成只用了 3 天时间，后期分模块调试的时间也很有限，因此没有实现最初想的用图形显示 GDRAM 映射到 DDRAM 达到显示真正的“图形”而不是现在 char* 结构显示的“字符串图形”的目标。另外虽然实现了随着玩家答题数目增多，一次性随机显示图形增加的想法，但是图形消失后没有一个计时模块，设定好答题所用的最多时间，超过

时间限制自动判别为游戏失败。

七、参考文献

LM16032 器件手册

HD7279A 器件手册

单片机 C 语言入门，龙脉工作室 刘鲲 孙春亮，人民邮电出版社，2008 年 4 月 1 日第一版

单片机 C 语言程序设计实训 100 例——基于 8051+Proteus 仿真（第 2 版）[彭伟][案例及素材]，电子工业出版社，2012