

中国传媒大学

电子设计竞赛设计报告

题 目：LED 阵列贪吃蛇游戏设计

姓 名：郭天楚（200910013276）

院 系：信息工程学院

专 业：广播电视工程

指导教师：卢起斌

目 录

一、方案简介.....	3
二、游戏介绍.....	3
三、硬件介绍.....	3
四、软件介绍.....	4
五、模块设计.....	5
1、HT1632C 的驱动.....	5
2、蛇数据的存储与更新.....	6
3、食物的放置.....	6
4、游戏完成与失败的判断.....	6
5、蛇吃到食物的判断.....	7
6、键值的获取与处理.....	7
7、蛇与食物的显示.....	7
8、游戏结束后的重新开始及数据初始化.....	7
9、按键的防抖.....	8
六、测试效果图.....	8
七、设计总结.....	11
八、参考文献.....	12

一、方案简介

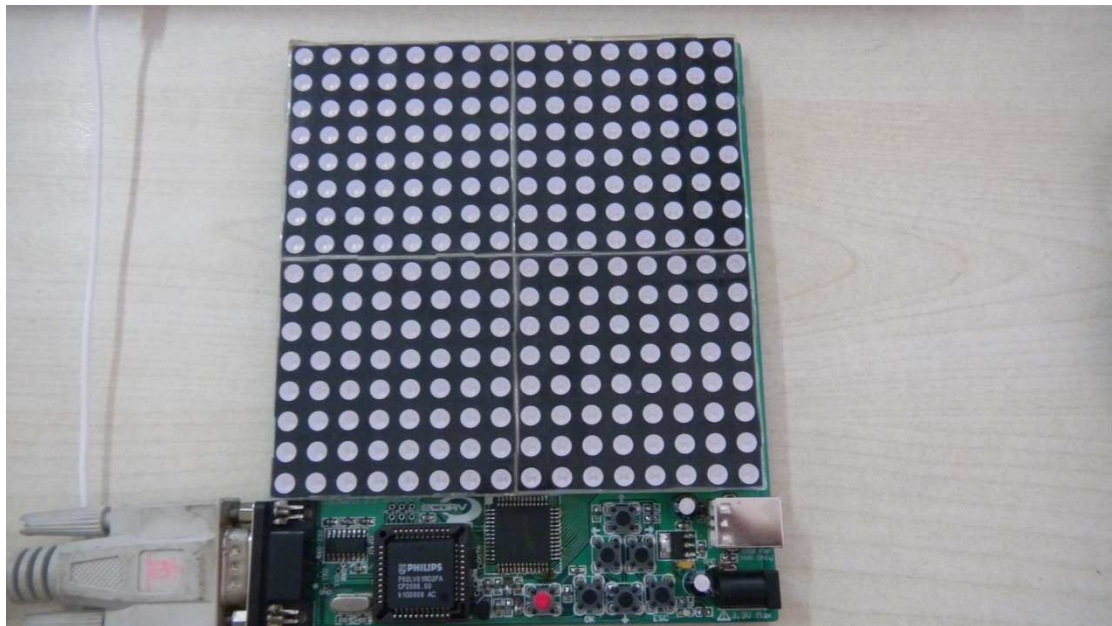
电路基于 P89V51RD2 单片机和 HT1632C LED 阵列驱动芯片的游戏电路。通过控制单片机程序来处理贪吃蛇游戏，控制 HT1632C 来驱动 16*16 的 LED 阵列显示。电路复位，依次显示“贪吃蛇”三个字，然后显示一个字符表情，按下“OK”键进入游戏。游戏失败或完成后按下“OK”键可以重新开始。本设计使用 Keil C51 进行开发，编程语言为 C51。

二、游戏介绍

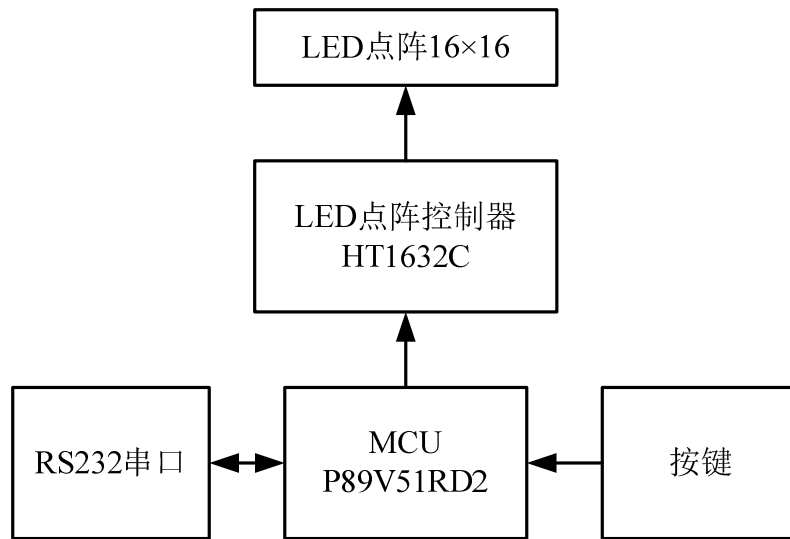
游戏开始后，蛇长为 4，在屏幕左上角横向排列，运动方向向下。通过按键“←”“→”“↑”“↓”来控制蛇的运动方向，使蛇吃到随即出现在屏幕上的食物。每当蛇吃到食物后，蛇的长度增加 1，当蛇长为 20 时，游戏顺利结束，显示成功字符表情。当蛇碰壁或者碰到自己身体时，游戏失败，显示失败字符表情。当游戏结束时按下“OK”游戏会重新开始，恢复初值状态。

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：



实验平台实物外观

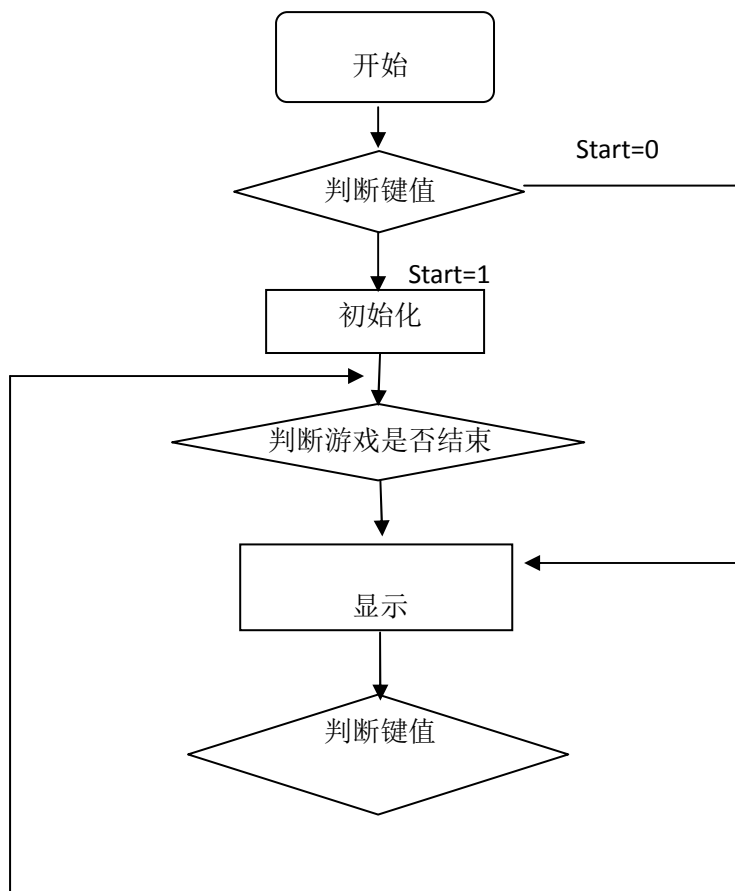


实验平台结构框架

本设计单片机采用 NXP（原 Philips）公司的 P89LV51RD2，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）。HT1632C 是内存映射的 LED 显示控制驱动芯片，包含 16 个级别脉宽调制控制亮度，集成了 256KHz RC 振荡器。有数据模式和命令模式。可选的 NMOS 输出渠道和 PMOS 输出渠道。可以选择多重的 ROW/COM 模式，即 32ROW/8COM 和 24ROW/16COM。本设计采用的是 24ROW/16COM 模式驱动 16*16LED 点阵。

实验板的四个方向以及“ESC”“OK”按键分别和单片机的 P2.0-P2.4 以及 P3.7 相连。

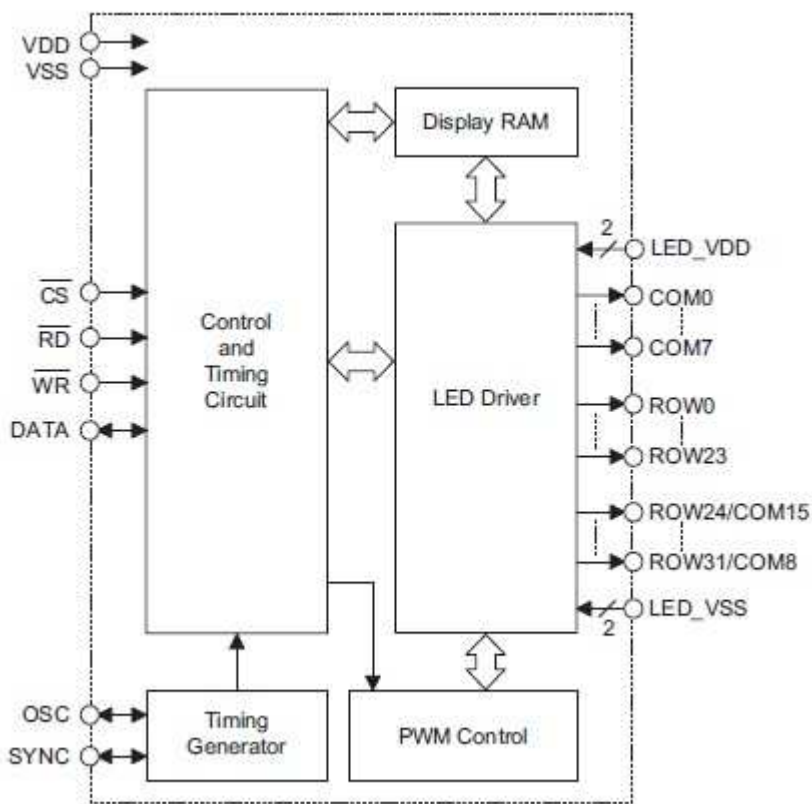
四、软件设计介绍





五、模块设计

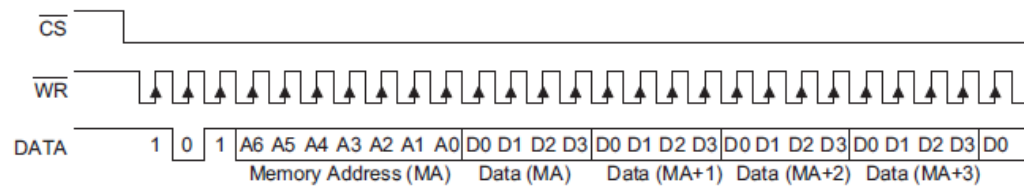
1、HT1632C 的读写控制



HT1632C 接口概述

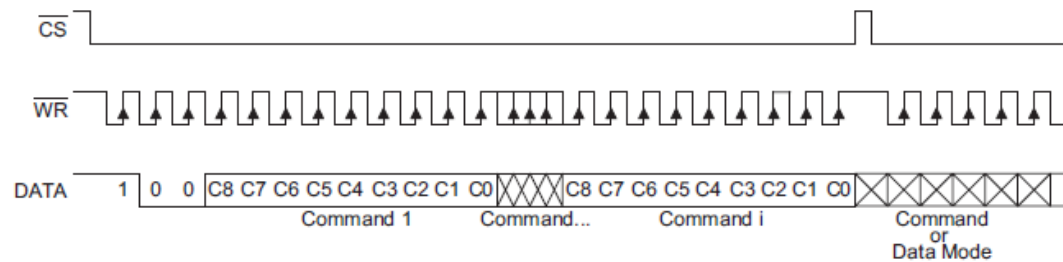
在 led 显示的程序当中，HT1632C 只采用写入数据模式，不需要从 RAM 中读出数据做处理。HT1632C 写入数据时，需要用到的引脚有 \overline{CS} , \overline{WR} , DATA，数据由 DATA 写入，由 \overline{WR} 做时钟脉冲（其脉冲输入需要软件控制单片机引脚输出）， \overline{CS} 为低电平时允许写入，高电平时禁止写入。HT1632C 不仅数据使用该方式写入，驱动其命令也按此方式写入。驱动命令控制着 HT1632C 的执行方式，模式选择等。所命令 ID 由 101 及 100 开头。输入命令以 100 开头，输入数据以 101 开头，当数据输入完毕， \overline{CS} 置 1，led 点亮。

WRITE Mode – Successive Address Writing



本设计主要采用的数据传送方式为连续地址写入方式，时序图如上。

Command Mode – Command Code = 1 0 0



本设计主要采用的指令传送方式为连续传送，时序图如上。

2、蛇形数据的存储与更新

蛇形在屏幕上显示时是靠坐标控制，只有横纵坐标，故定义二维数组 `Snake_point[snakemax][2]` 保存整个蛇每个部位的坐标，其数组最大为 `snakemax=20`。左上角一点坐标为 (0,0)，横向向右坐标为正，纵向向下坐标为正。

蛇数据更新时，调用 `for` 循环，限制循环小于蛇的目前长度，从蛇尾开始，坐标一次为前一个的数值，直到蛇头，蛇头坐标等于蛇头下一坐标 `snakenext_x,snakenext_y`。

3、食物的放置

当食物放置标志位 `setfood` 为 1 时，可以放置食物。贪吃蛇游戏食物的位置是随机产生，故食物坐标的需要获取两个随机数。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，中断函数中将定时器重新赋值，继续定时。由于整个屏幕的坐标在 0-15 内，故对于随机数要除以 16 取余数，才能使所有随机数在 0-15 范围内。程序为：`Food_x=TH0%16,Food_y=TL0%16`。但，产生食物的坐标与蛇的坐标不能相同，否则会出现食物与蛇重叠的现象，故需要将食物坐标与蛇坐标一一比较，当食物坐标不与蛇坐标相同，且不与蛇头的下一坐标相同时，食物放置成功，设置食物放置标志位为 0，使得蛇在未吃到食物时只在屏幕上固定显示一个食物。

4、游戏完成与失败的判断

(a) 游戏完成判断

当蛇长等于 20 时，游戏成功完成，设置成功完成标志位 `success=1`。

(b) 游戏失败判断

当蛇头的横纵坐标大于 15 或小于 0，即蛇碰壁，游戏失败。当蛇头坐标与蛇身坐标相同，即蛇撞到自己身体，游戏失败。游戏失败标志位 `over=1`。

5、蛇吃到食物的判断

蛇头的下一个坐标 `snakenext_x,snakenext_y`与食物坐标相同时，蛇吃到食物，此时设置 `setfood` 为 1，可以放置下一个食物，蛇长增加 1。

6、键值的获取与处理

定义运动状态 `up, down, right, left`。检测当有按键按下时，当蛇的运动方向与按键不相反时，按键有效，获取新的运动状态。例如：当蛇的运动状态为向右，那么按下向左的按键无效，按下其它按键有效，改变新的运动状态。

检测运动状态，当运动状态为向上时，蛇头下一纵坐标减 1，运动状态向下时，蛇头下一纵坐标加 1，运动状态向左时，蛇头下一横坐标减 1，运动状态向右时，蛇头下一横坐标加 1。

7、蛇与食物的显示

蛇与食物的存储方式都为坐标形式，需要将坐标转换成 32 个数据才能正确在屏幕上显示。先将显示数组的数据全部清零，然后计算数值。在横向显示蛇时，由于 16 进制的一个数字代表屏幕上横向 4 个点，所以为了不清除前面的数据，需将显示数组的数据按位或。其具体算法如下：

```
if((snake_x<0)||snake_y<0)
    return;
else if(snake_x<8)
    FT1632_FrameBuffer[snake_y*2]=FT1632_FrameBuffer[snake_y*2]|(0x80>>(7-(snake_x%8)));
else if(snake_x>7)
    FT1632_FrameBuffer[snake_y*2+1]=FT1632_FrameBuffer[snake_y*2+1]|(0x80>>(7-(snake_x%8)));
```

当检测到标志位 `success` 或 `over` 为 1 时，游戏结束，显示游戏结束图像。

8、游戏结束后的重新开始及数据初始化

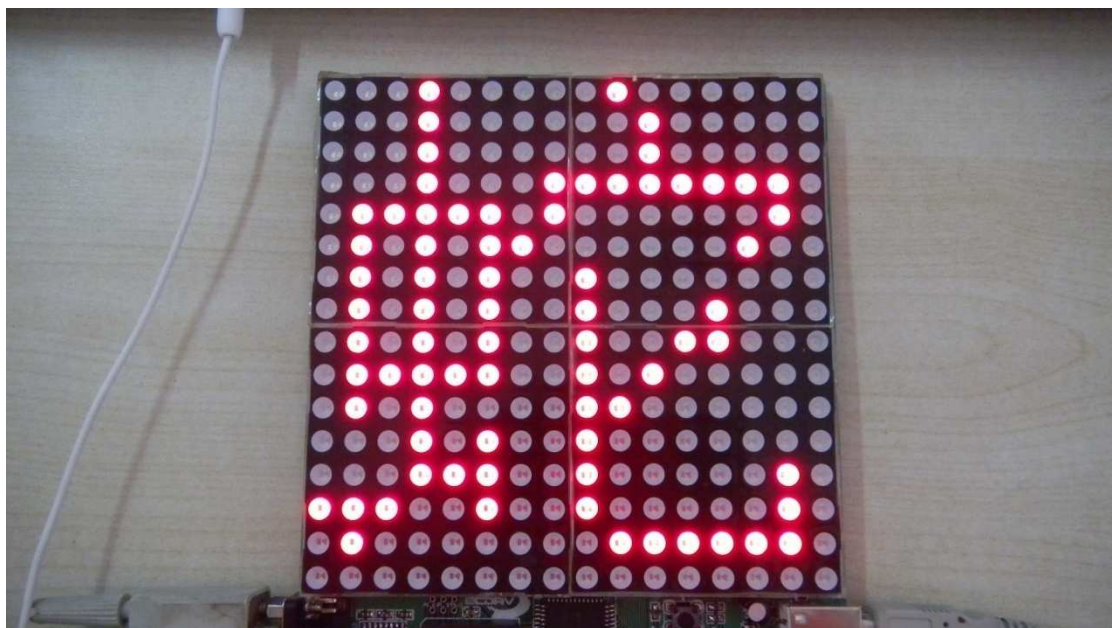
设置游戏开始标志位 `start`，无论在刚刚复位后显示的字符表情还是在游戏结束后，按下“OK”键后，`start` 标志位置 1。游戏程序检测到 `start` 位为 1 时才正常运行。

游戏数据初始化后蛇显示位置在屏幕左上角，蛇长为 4，故设置蛇坐标为 (0,0), (1,0), (2,0), (4,0), `snakelenth=4`。标志位 `over, success` 为 0，游戏正常运行，`setfood` 为 1，允许放置食物。蛇头下一坐标为蛇头坐标。运动方向为 `down` 向下运动。

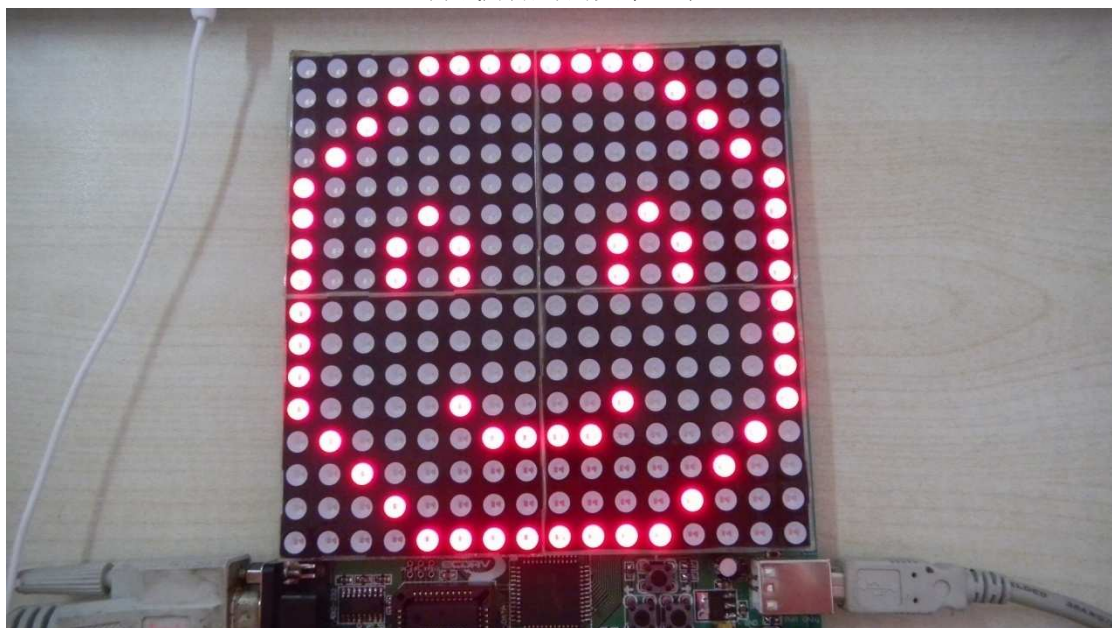
9、按键的防抖

在用单片机对键盘处理的时候需要涉及到键盘的去抖动。抖动是机械按键按下和弹起的瞬间，由于触点的弹性而产生的一连串的抖动，一般抖动时间是 5-10ms。为了取得正确的键值，必须设法消除抖动的影响。采用软件去抖动是避开抖动部分的时间，等按键触电稳定后再对其进行处理。实现方法是多次查询，即先查询按键，当查询到电平时立即延时 10ms-30ms，以避开抖动，延时结束后再读取一次 I/O 口的值，判断是干扰信号还是按键按下。

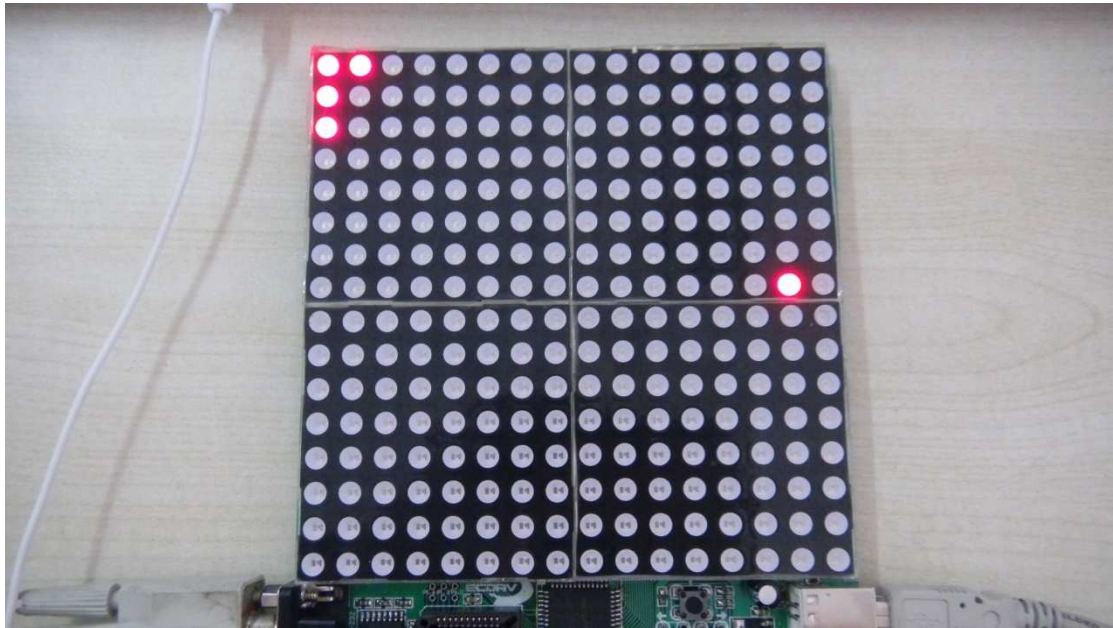
六、测试效果图



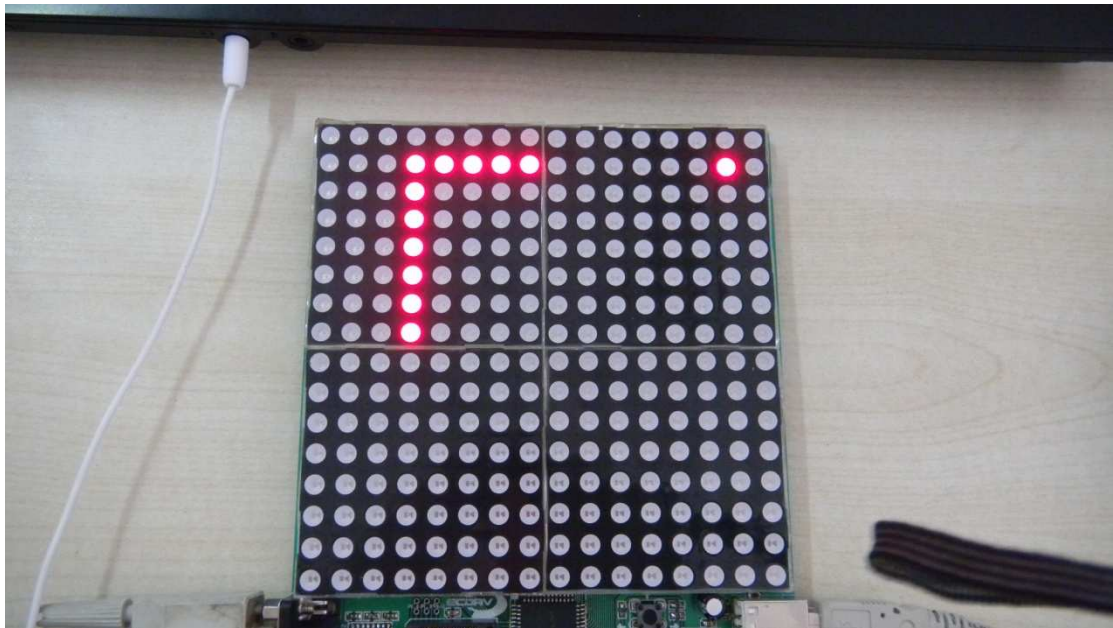
游戏开始时的文字显示



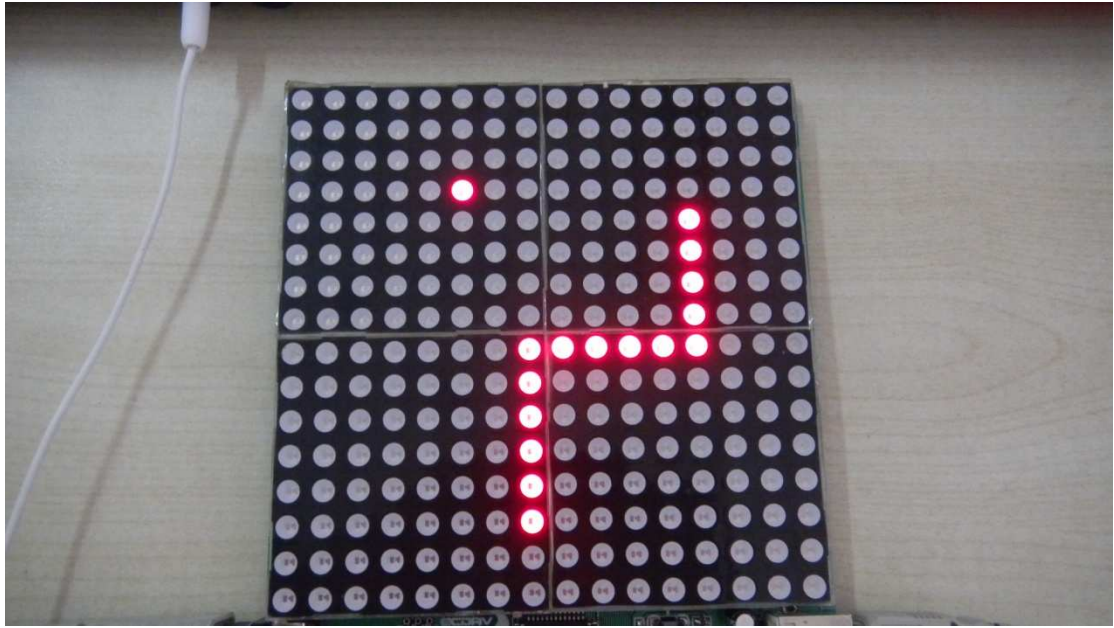
显示文字后的字符表情



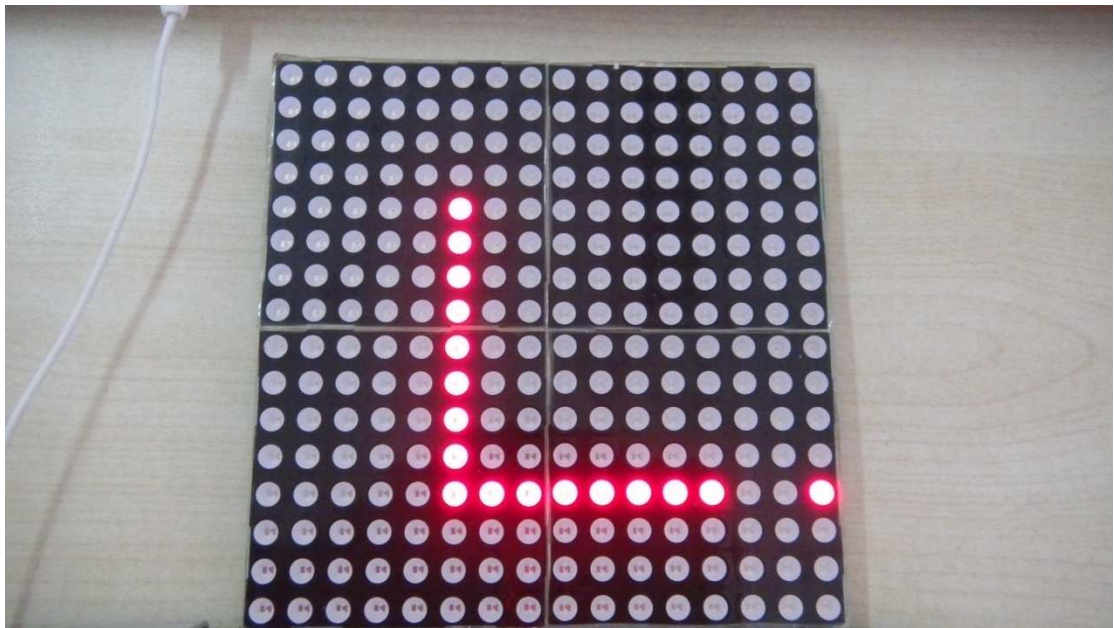
按下“OK”键后游戏开始



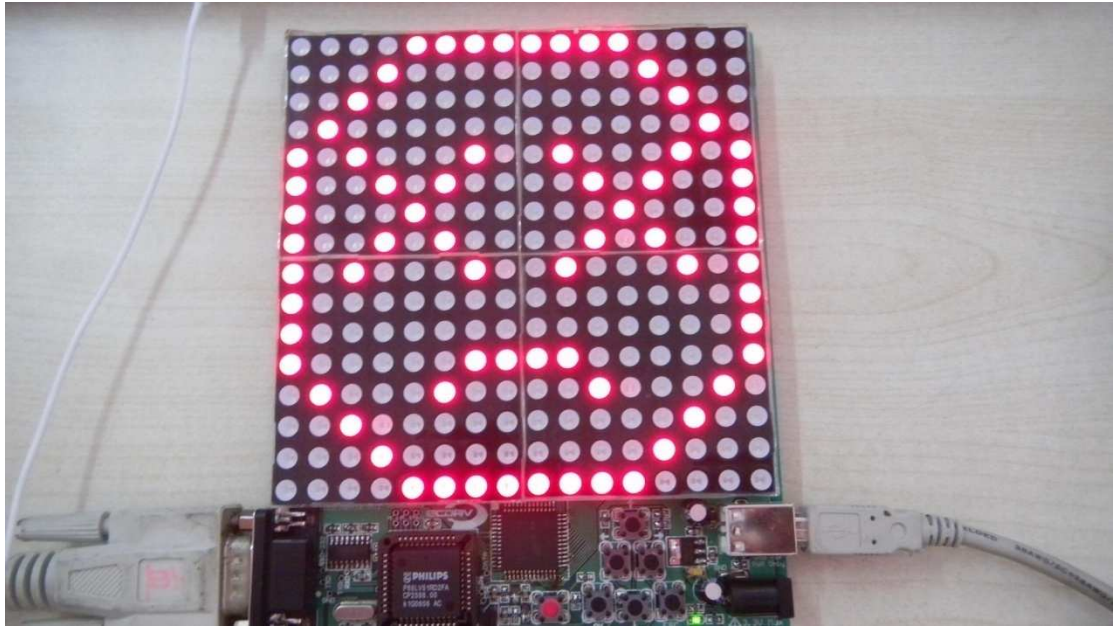
游戏进行中 (1)



游戏进行中 (2)



游戏进行中 (3)



游戏失败

七、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、在程序设计的开始，没有考虑到如何存储蛇与食物的数据。它们的存储方式与字符表情显示不同，字符表情显示可以一次性存储 32 个数据来表示一个图案，但是蛇与食物是时刻在变的，无法用特定的某 32 个数据保存。后来在网上搜索参考资料，看到有人采用坐标存储方式，这样不仅能很好的记录蛇与食物的数据，连蛇的移动判断也变的十分简单。之后的问题就是如何将坐标转换成数据存入显示缓存数组 `buffer[32]`中。

2、在最初写的将坐标转换成数据存储在缓存数组的算法中，横向显示始终不完整。后发现横向 4 个点为十六进制一个数表示，为了使数据不被覆盖，应采用按位或的方式，这样在屏幕上显示就正常了。该存储方式也可以应用在其它类似游戏中，如俄罗斯方块。

3、最开始设计的程序中，是“贪吃蛇”文字显示之后直接进入游戏，无任何手动控制。这样产生的后果就是采用定时器获取的食物随即坐标每次都是一样的。故加入手动按键，这样游戏开始的时间就会每次都不同，获取的食物随即坐标也不同。

游戏的构思和软件设计还有很多的提升空间。例如游戏目前蛇的移动速度是相同的，可以再加入控制速度的变量 `speed`，作为传入延时函数的参数，控制显示延时，这样蛇的移动速度就会随着吃的食物的增多而加快。还可以可以多设置几关，并逐渐增加难度，使游戏变成通关模式，并记录分数。

该设计仍存在一些问题。在游戏的显示过程中，延时函数占用了大部分时间。也就是说在延时函数调用时，按下方向键并不能立刻被响应，所以造成的结果是，在蛇改变方向的前一个坐标长按方向键，蛇才会改变走向。造成的延时还是不可忽略的。其解决方案有两种方式：

(1) 改进硬件设计。使用专用的按键控制器或其它措施，将按键动作与单片机外部中断相关联，这样当任意一个按键被按下时，软件均能立刻响应外部中断并检测键值。

(2) 改进软件结构。在延时程序中加入按键检测函数。在每次执行延时过程中都会先检测一次按键，当有按键按下时，跳出延时程序，对按键做处理。

八、参考文献

HT1632C 器件应用手册，网址：

<http://www.holtek.com.cn/china/tech/appnote/consumer/pdf/HA0127sv110.pdf>

Proteus 显示控制系统设计与实例，朱清慧，清华大学出版社，2011-05

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，2008-04

网络示例程序，网址：<http://ishare.iask.sina.com.cn/f/14060056.html>，作者：秦时明月