

# 重庆大学

## 电子综合设计与实践

综合实验教学楼的设计



学生：张银悦            苏宇鸿

          聂志远            王 浩

学号：20114933, 20114939

          20114945, 20114574

专业：自动化 电子信息工程

年级： 2011 级

指导老师：谢礼莹

重庆大学自动化学院

重庆大学通信工程学院

2014 年 3 月

**Electronic Integrated Design And Practice Of  
Chongqing University  
Design of General Teaching Building**



**Undergraduate: Zhang Yinyue Su Yuhong**

**Nie Zhiyuan Wang Hao**

**Supervisor: EE. Xie Liying**

**Major: Automation**

**Electrical & Information Engineering**

**College Of Automation**

**College Of Communication Engineering**

**Chongqing University**

**March 2014**

重庆大学本科学生课程设计指导教师评定成绩表

学号	20114933, 20114939 20114945, 20114574	指导教师	谢礼莹
学院	自动化 通信工程	专业	自动化 电子信息工程
学生姓名	张银悦, 苏宇鸿, 聂志远, 王浩		
课程设计 题目	用单片机控制的乒乓球游戏机		
指 导 教 师 评 语			
课程设计 成绩	<p>指导教师签名： 年 月 日</p>		

## 重庆大学本科学生课程设计任务书

课程设计题目		用单片机控制的乒乓球游戏机			
学院	自动化 通信工程	专业	自动化 电子信息工程	年级、班	2011 自动化 1 班 2011 电子 1 班
<p>设计要求：</p> <ol style="list-style-type: none"><li>基本要求：必须实现 2 人单打游戏，要求实现乒乓球的来回运动，能显示比分。</li><li>提高要求：要求实现 4 人双打游戏，乒乓球的速度分级可调，能显示双方的比分和局数，可开关复位，有击球，接球的声音效果等。</li><li>扩展要求：除 1, 2 项功能必须达到之外，还要求有简单的语音功能(音乐伴奏)或者开机画面。</li></ol> <p>学生应完成的工作：</p> <p>电路原理图设计，扩展板布线，硬件焊接，软件编写，程序调试，至少能够完成基本要求。</p> <p>参考资料：</p> <ol style="list-style-type: none"><li>《51 单片机 C 语言开发详解》</li><li>《基于 Proteus 的单片机可视化软硬件仿真》</li><li>《单片机原理，应用与 Proteus 仿真》</li></ol> <p>课程设计工作计划：</p> <p>第一周，完成电路图设计及用软件画图，制定好整个工作计划； 第二周，小组分两部分，2 个同学负责程序编写，另 2 个同学负责 PCB 板设计，并完成电路的焊接，完成硬件工作； 第三周，开始调试，并找出一些问题，逐步解决。进行联调，并脱机。检查，推优。</p>					
<p style="color: red;">任务下达日期 <u>2014 年 2 月 17 日</u>      完成日期 <u>2014 年 3 月 7 日</u></p> <p>指导教师（签名）                          学生（签名）</p>					

说明：1、学院、专业、年级均填全称。

2、本表除签名外均可采用计算机打印。本表不够，可另附页，但应在页脚添加页码。

## 摘要

单片机又称单片微控制器，它不是完成某一个逻辑功能的芯片，而是把一个计算机系统集成到一个芯片上。单片机具有编程简单、易学、可靠性高、体积小等优点，广泛地应用于自动控制系统中。用单片机控制的乒乓球游戏机是为了实现模拟乒乓球比赛的过程和规则。本次我组设计采用 STC89C516 单片机来控制模拟，用 LM386 芯片构成功放电路配合游戏中所需音效，采用 12864-12 型 LCD 液晶显示屏来代表乒乓球桌面显示乒乓球路径，液晶屏两边为参赛双方的位置，可实现单打、双打等内容，比赛双方用按钮开关来模拟启动球拍击球，在显示上用数码管来显示比赛分数和局数之比。开机具有开机音乐和开机动画，接发球过程中有音效。在乒乓球游戏基础上我组对游戏机功能进行了扩展，除乒乓球游戏外使之具有俄罗斯方块，音乐播放器及数字温度计的功能，令游戏机功能多元化，丰富化。本报告将详细地介绍系统硬件过程和软件代码。

**关键词：** STC89C516, LM386, 12864-12

小组成员介绍及分工：

张银悦（组长）——负责硬件连线与设计及实验设计报告的撰写；

王浩——负责软件编程；

聂志远——负责原理图制作及软件编程；

苏宇鸿——负责硬件连线；

## ——基于单片机的乒乓球游戏系统的设计

### 1. 引言

#### 1.1 设计背景

随着科技的发展，智能化已逐步渗透我们日常生活的方方面面。在我们的娱乐生活中，各种智能化游戏机层出不穷，不仅丰富了我们的生活，带来了便捷，同时也给我们许多启发。基于单片机功能强、可靠性高、使用方便、体积小及重量轻等优点，其已被国外广泛应用于自动化控制的各个领域。而在国内，随着研究不断深入，单片机技术与产品开发应用方面的发展也很快，各种电子产品已越来越多地采用单片机控制系统且性价比高。本次课程设计我们从实用的角度以 STC89C516 单片机为核心设计了一种以乒乓球游戏为基础的多功能游戏机。

乒乓球是中国的国球，在中国拥有广泛的群众基础。此次设计融入了现实比赛的规则，使之不失乒乓球本身的娱乐性，更结合了传统的电子产品的小巧和便携性，缩小了乒乓球游戏实地体验受空间要求的局限性，使乒乓球爱好者更大程度地体验其娱乐性。本系统由 STC89C516 单片机，LCD 显示屏，LM386 功放模块，按键，温度测试模块等组成系统。以 51 系列单片机为中心控制器件，用 LED 发光二极管点阵或者液晶显示器来模拟乒乓球的运动，实现 2 人游戏，乒乓球的运动速度分级可调，能显示游戏比分，有击、接球的声音效果等，还可以扩展开、关机音乐，语音对话功能等。用 LCD 显示屏显示乒乓球的运动路径. 再用两个按键模拟两个球拍，按下键代表击球. 另设置四个键来控制其他附加功能的实现，增加游戏的趣味性.

本游戏供两人游戏，实现乒乓球的双打，以一方按键（发球）开始，此时“乒乓球”以默认的路径运动，在路径一半前可以按方向键改变球的运动路径. 接球方应在“球”到达终点时刻击球，提前或滞后击球则视为失误丢失“一分”，若在合适的时间内将球击回则继续游戏，直至本局分出胜负. 游戏过程中，LCD 显示器显示双方比分.

## 1.2 功能说明

本次设计的乒乓球游戏机是以 STC89C516 单片机为核心的，选用 12864-12 型 LCD 显示屏，通过 LCD 显示点阵来模拟乒乓球的运动。本实验板具体能实现以下四个功能：

1. 乒乓球游戏。乒乓球游戏分为单打和双打两种玩法，四键按下为发球，八键全部按下退出至主菜单。单打时，K1、K2、K5、K6 分别代表拍子的上下移动，K3、K7 为弧线球按钮，K4、K8 为墙（大招）按钮。（大招功能：不论拍子位置是否能接住球，按下大招按钮球都能以对角球击出）双打时，K1、K2、K5、K6 分别代表甲方乙方的上下拍子击球按钮，K3、K7 为弧线球按钮，K4、K8 为对角球按钮，按下击球按钮同时按下弧线球或对角球按钮，即可打出不同效果的球。
2. 俄罗斯方块。玩法与传统俄罗斯方块相同，K3、K4 为左右移动按钮，K1 为变形按钮，K2 为加速下落按钮，同时按住 K7、K8 按钮退出至主菜单界面。
3. 音乐播放器。播放器可以播放单片机中预存的几首音乐，其中 K1、K3、K4 都为暂停按钮，K2 为切歌按钮，同时按住 K7、K8 按钮退出至主菜单界面。
4. 数字温度计。温度计可以检测电路板所处位置的温度，同时按住 K7、K8 按钮退出至主菜单界面。

## 2. 设计要求

### 2.1 基本要求

1. **设计内容：**采用 LCD 液晶显示，51 系列单片机为中心控制器件来模拟乒乓球的运动。
2. **基本要求：**必须实现 2 人单打游戏，要求实现乒乓球的来回运动，能显示比分。
3. **提高要求：**要求实现 4 人双打游戏，乒乓球的速度分级可调，能显示双方的比分和局数，可开关复位，有击球，接球的声音效果等。
4. **扩展要求：**除 1, 2 项功能必须达到之外，还要求有简单的语音功能（音乐伴奏）或者开机画面。

## 2.2 技术指标

本课题的技术指标如下：

1. “乒乓球”移位一次的过程中，分别进行 50 次，37 次，25 次，12 次测试（既  $20\mu s$  一次）查询是否有击球动作。
2. 接球方的击球动作应发生在“乒乓球”到达本方最后三格内，如接球方在规定的地点内按键，则接球成功，然后乒乓球向对方运动。
3. 接球方提前或滞后击球，则接球失误，由对方得分。这时“乒乓球”熄灭，LED 显示双方比分，然后等待发球方发球。
4. 每一局中，哪方的得球个数首先满 11 个，判断哪方的在当局中获胜，打满三局后，系统将显示整体局分比表示哪方获得胜利。
5. 用左右按键分别模拟左拍击球和右拍击球，双方比分由 LED 显示。
6. 用喇叭模拟击球声音，发球声音频率为 1kHz，接球失误提示声音频率为 2kHz。每局结束蜂鸣器报警频率为 1.5kHz.

## 3. 硬件设计

### 3.1 硬件电路框图

本次电子综合设计的乒乓球游戏机的硬件电路方面包括：

单片机主电路，包括复位电路、振荡电路、电源（直流供电，USB 供电）；

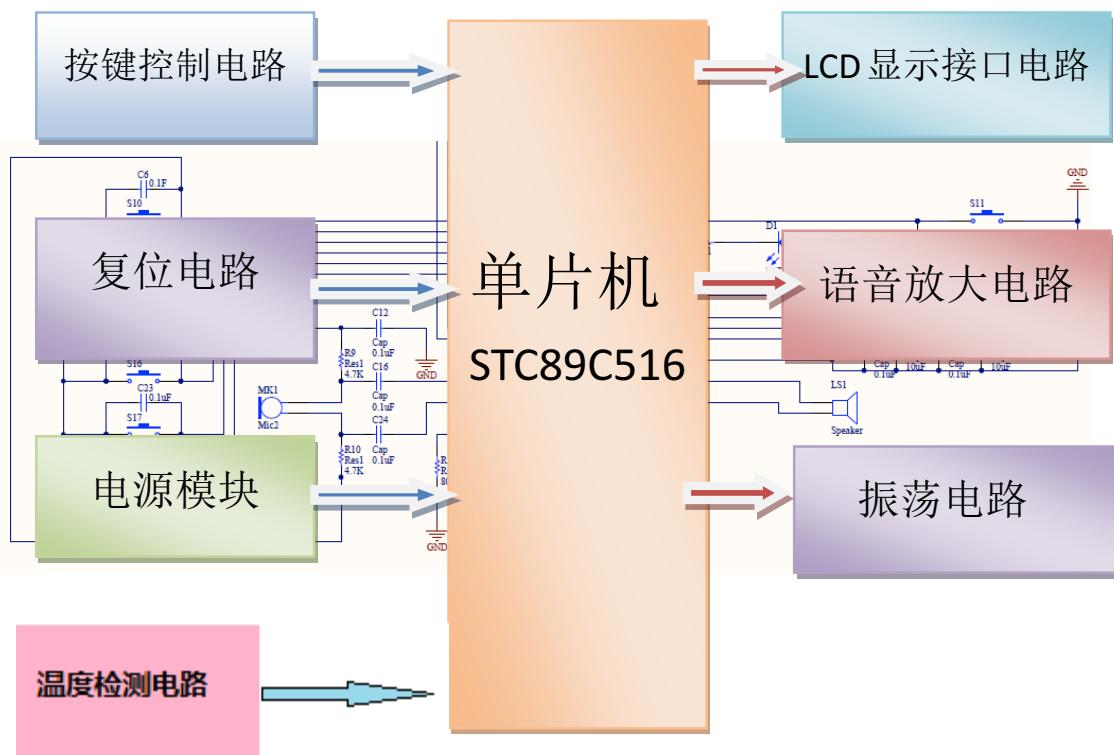
LCD 显示接口电路（模拟乒乓球，游戏比分显示）；

语音放大电路；

按键控制电路（模拟球拍）

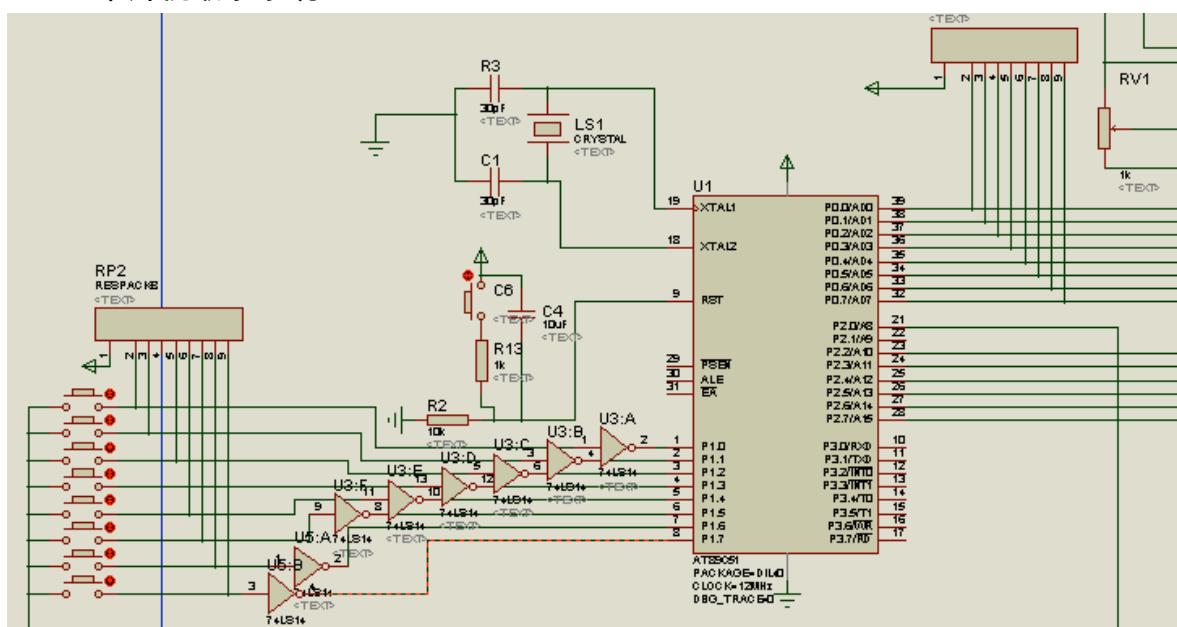
温度检测电路

其原理结构框图如下：

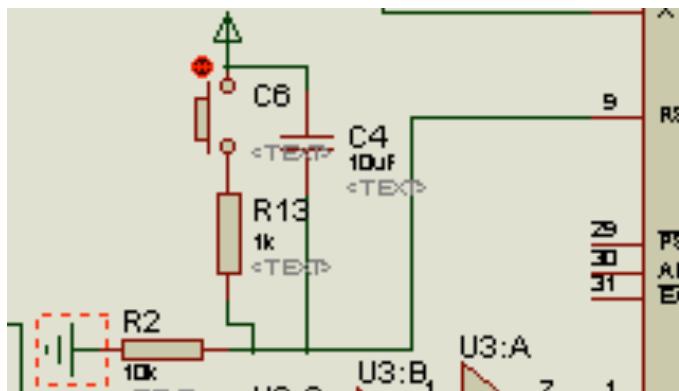


### 3.2 各部分电路图

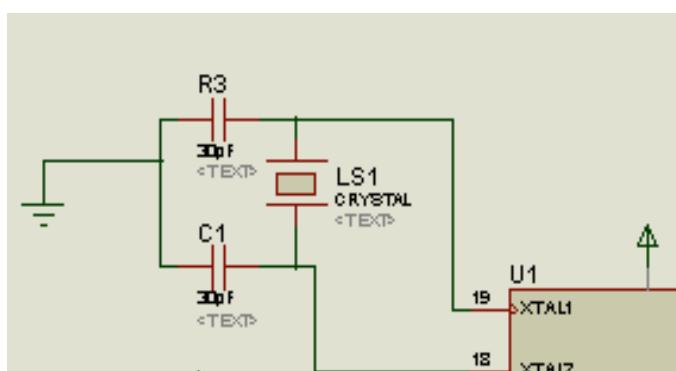
### 1) 51 单片机最小系统:



## 2) 复位电路和振荡电路

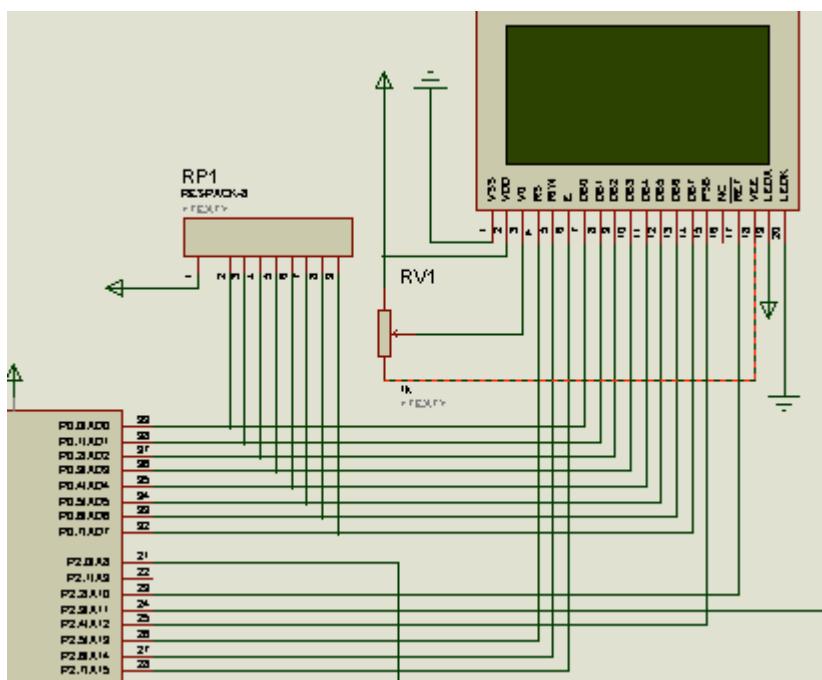


复位电路

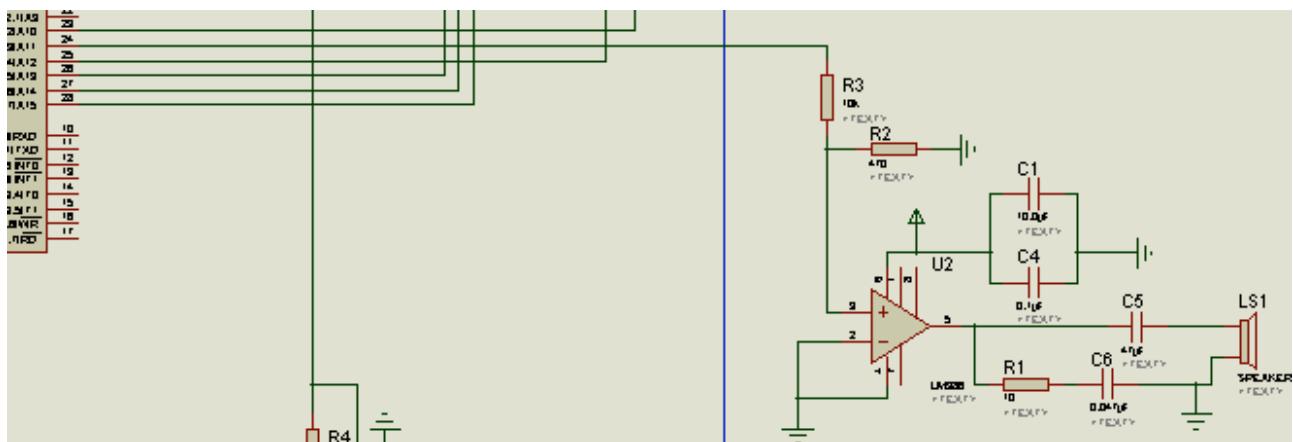


振荡电路

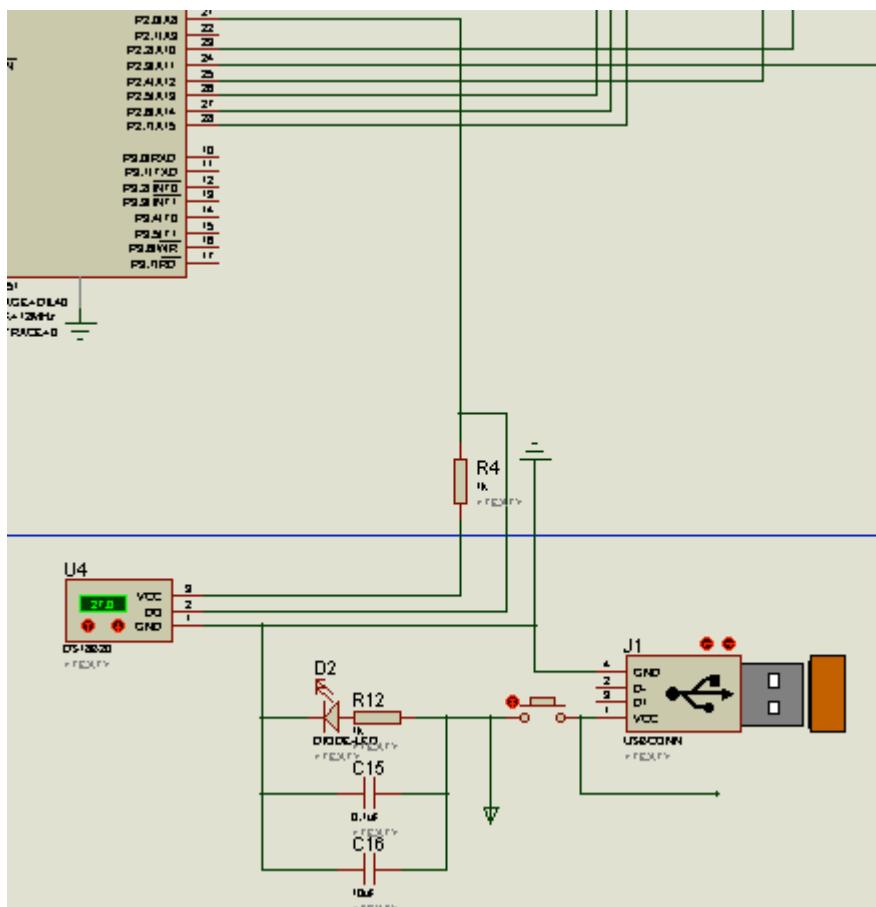
## 3) LCD 显示屏接口电路



#### 4) 语音放大电路



#### 5) 电源及温度模块



### 3.3 设计方案相关元件说明

由于本项目所需 I/O 端口比较少，建议采用一块 89C516 主控芯片。声音脉冲经三极管放大后驱动喇叭发声。

#### 1. 89C516 管脚功能分配：

其中 p1. 0 代表玩家1的左球拍；

其中 p1. 1 代表玩家1的大招墙；

其中 p1. 2 代表玩家1的弧线球；

其中 p1. 3 代表玩家1的右球拍；

其中 p1. 4 代表玩家2的左球拍；

其中 p1. 5 代表玩家2的右球拍；

其中 p1. 6 代表玩家2的弧线球；

其中 p1. 7 代表玩家2的大招墙；

当检测到相关按键时，89C516 执行对应操作。P2.3 输出脉冲作为喇叭的输入信号。P1.0——P1.7 口经 74LS14 与按键开关相接，电源及温度模块与 P2.0 口相接。

软件部分由 5 个模块组成，分别时主程序，乒乓球移动程序，延时程序，发音程序和显示程序。主程序完成系统初始化，并控制个模块工作；发音程序由三个子程序对应 1KHZ, 1.5KHZ, 2KHZ 三个声音频率。

复位电路和晶振部分参见芯片功能介绍部分的 89C516。

本方案电路结构简单，布局合理，软硬件可靠性高，升级潜力大！

#### 3.3.1 STC89C516 单片机

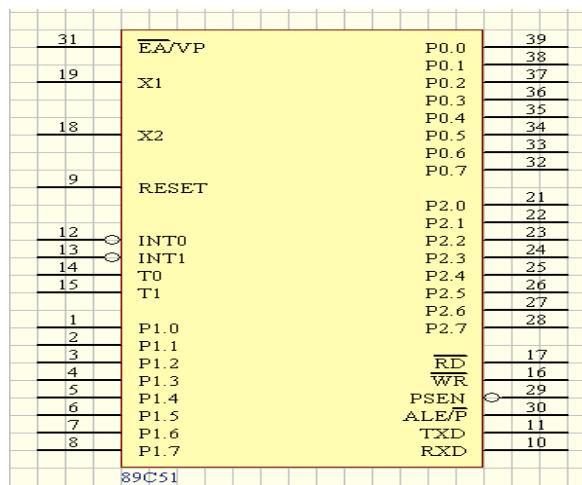
STC89C516 是 51 系列单片机的一个型号，它是 ATMEL 公司生产的。STC89C516 是一个低电压，高性能 CMOS 8 位单片机，片内含 61k bytes 的可反复擦写的 Flash 只读程序存储器和 1280 bytes 的随机存取数据存储器 (RAM)，器件采用 ATMEL 公司的高密度、非易失性存储技术生产，兼容标准 MCS-51 指令系统，片内置通用 8 位中央处理器和 Flash 存储单元。

STC89C516 有 40 个引脚，32 个外部双向输入/输出 (I/O) 端口，同时内含 2 个外中断口，3 个 16 位可编程定时计数器，2 个全双工串行通信口，2 个读写口线，STC89C516 可以按照常规方法进行编程，但不可以在线编程 (S 系列的才支持在线编程)。其将通用的微处理器和 Flash 存储器结合在一起，特别是可反复擦写的 Flash 存储器可有效地降低开发成本。

兼容 MCS51 指令系统

61k 可反复擦写(>1000 次) Flash ROM	可编程 UART 串行通道
32 个双向 I/O 口	2 个外部中断源
1280x8bit 内部 RAM	共 5 个中断源
3 个 16 位可编程定时/计数器中断	2 个读写中断口线
时钟频率 0~24MHz	3 级加密位
2 个串行中断	低功耗空闲和掉电模式
	软件设置睡眠和唤醒功能

STC89C516 为 40 脚双列直插封装的 8 位通用微处理器，采用工业标准的 C51 内核，在内部功能及管脚排布上与通用的 8xc52 相同，其主要用于会聚调整时的功能控制。功能包括对会聚主 IC 内部寄存器、数据 RAM 及外部接口等功能部件的初始化，会聚调整控制，会聚测试控制，红外遥控信号 IR 的接收解码及与主板 CPU 通信等。主要管脚有：XTAL1 (19 脚) 和 XTAL2 (18 脚) 为振荡器输入输出端口，外接 12MHz 晶振。RST/Vpd (9 脚) 为复位输入端口，外接电阻电容组成的复位电路。VCC (40 脚) 和 VSS (20 脚) 为供电端口，分别接+5V 电源的正负端。P0~P3 为可编程通用 I/O 脚，其功能用途由软件定义，在本设计中，P0 端口 (32~39 脚) 被定义为 N1 功能控制端口，分别与 N1 的相应功能管脚相连接，13 脚定义为 IR 输入端，10 脚和 11 脚定义为 I2C 总线控制端口，分别连接 N1 的 SDAS (18 脚) 和 SCLS (19 脚) 端口，12 脚、27 脚及 28 脚定义为握手信号功能端口，连接主板 CPU 的相应功能端，用于当前制式的检测及会聚调整状态进入的控制功能。



## 1. 电源和晶振：

Vcc——运行和程序校验时加+5V。

Vss——接地。

X1——输入到震荡器的反相放大器。

X2——反相放大器的输出，输入到内部时钟发生器。

## 2. I/O: 4 个端口，32 各 I/O:

P0——8 位、漏极开路的双向 I/O 口。

当使用片外存储器时，作地址和数据分时复用。在程序校验期间，输出指令字节（这时需加外部上拉电路）。P0 口（作为总线时）能驱动 8 个 LSTTL 负载。

P1——8 位、准双向 I/O 口。

在编程/校验期间，用做输入低字节地址。P1 口可以驱动 4 个 LSTTL 负载。

P2——8 位、准双向 I/O 口。

当使用片外存储器时，输出高 8 位地址。P2 可以驱动 4 个 LDTTL 负载。

P3——8 位、准双向 I/O 口，具有内部上拉电路。

P3 提供各种替代功能。在提供功能时，其输出锁存器应由程序置 1。

P3 口可以输入/输出 4 个 LSTTL 负载。

串行口：

P3.0——RXD（串行输入口），输入。

P3.1——TXD（串行输出口），输出。

中断：

P3.2——INT0 外部中断 0，输入。

P3.3——INT2 外部中断 1，输入。

定时器/计数器：

P3.4——T0 定时器/计数器 0 的外部输入，输入。

P3.5——T1 定时器/计数器 1 的外部输入，输入。

数据存储器选通：

P3.6——WR 低电平有效，输出，片外存储器写选通。

P3.7——RD 低电平有效，输出，片外存储器写选通。

## 3. 控制线：共四根。

### (1) . 输入：

RST——复位输入信号，高电平有效。在震荡器工作时，在 RST 上作用两个机器周期以上的高电平，将器件复位。EA/Vpp——片外程序存储器访问允许信号，低电平有效。在编程时，其上加 21V 的编程电压。

### (2) . 输入、输出：

ALE/PROG——地址锁存允许信号，输出。用做片外存储器访问时，低字节锁存。ALE 以 1/6 的震荡频率稳定速率输出，可用做对外输出的时钟或用于定时。在 EPROM 编程期间，作输入。输入编程脉冲 (PROG)。ALE 以驱动 8 个 LSTTL 负载。

### (3) . 输出：

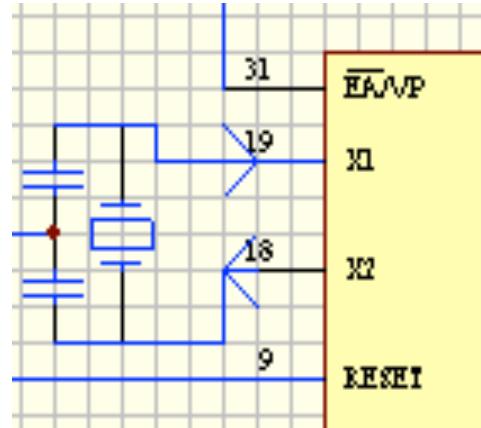
PSEN——片外程序存储选通信号，低电平有效。在从片外程序程序存储器取指期间，在每个机器周期中，当 PSEN 有效时，程序存储器的内容被送上 P0 口（数据总线）。PSEN 可

以驱动 8 个 LSTTL 负载。

#### 4. 振荡电路

在 89C51 芯片内部有一个高增益反相放大器，其输入端为芯片引脚 X1，其输出端为引脚 X2。只需要在片外通过 X1 和 X2 引脚跨接晶体振荡器和微调电容，形成反馈电路，振荡器即可工作。振荡电路原理如图所示：

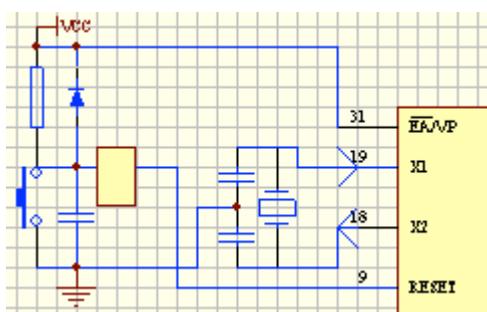
振荡器的工作可以由 PD 位（特殊功能寄存器 PCON 中的一位）控制。当 P/D 置 1 时，振荡器停止工作，系统进入低功耗工作状态。如图 3 所示，用晶振和电容构成谐振电路时，电容的大小影响振荡器振荡的稳定性和起振的快速性，通常选择 10—30PF 左右。在设计电路板时，晶振、电容等均应尽可能靠近芯片，以减小分布电容，保证振荡器振荡的稳定性。



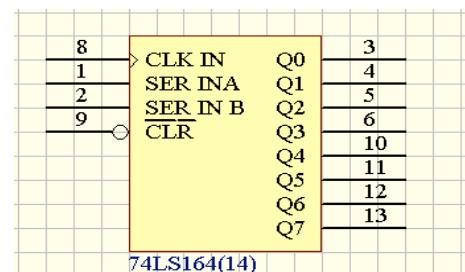
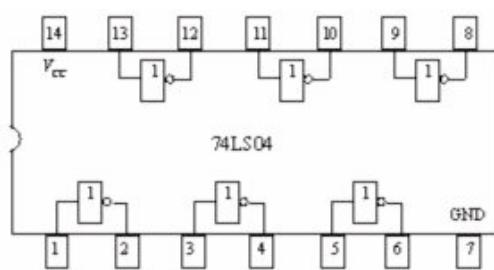
#### 5. 复位电路

##### (1)、复位信号

RST 引脚复位信号的输入端。复位信号是高电平有效，其有效时间应持续 24 个振荡个脉冲周期（即两个机器周期）以上。整个复位电路包括芯片内、外两部分。外部电路产生的复位信号（RST）送斯密特触发器，再由片内复位电路在每个机器周期的 S5P2 时刻对斯密特触发器的输出进行采样，然后才得到内部复位操作所需要的信号。产生复位信号的电路逻辑图如图所示：

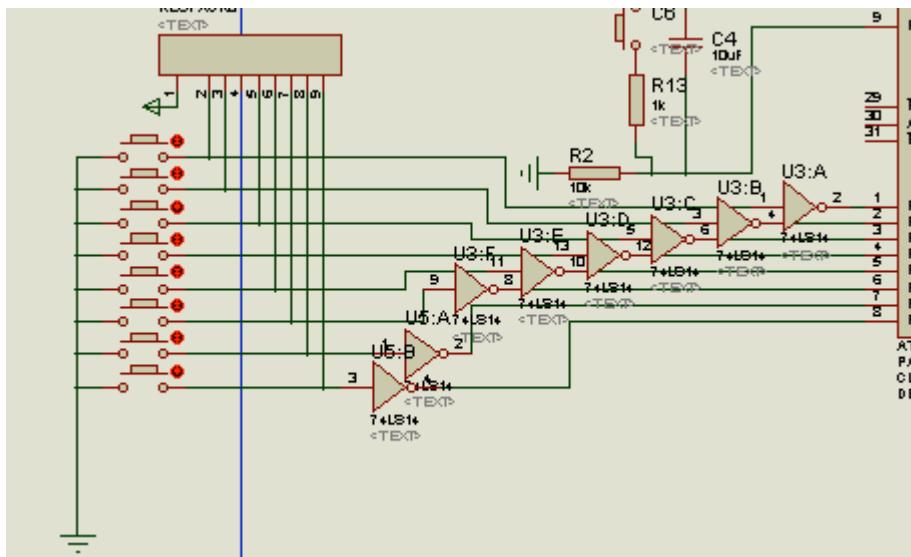


ST 引脚是复位信号的输入端，复位信号是高电平有效，其有效时候持续 24 个震荡脉冲周期以上。整个复位电路包括芯片内外两部分。本设计采用按键复位方式。按键电平复位是通过使复位端经电阻与 Vcc 电源接通而实现的。复位电路中用到了 74LS14，其结构如下：

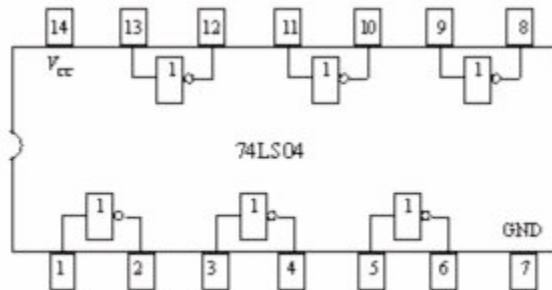


## 5、按键电路

由于按键实在一定时刻内按下的，程序要时时检测各个 P1 口。为了防止错误的产生，我们运用了去抖动电路。如下图所示：



上拉电阻为各个按键的信号输入提供驱动电流, 扳键开关由于是非即时的状态故不需要去抖动电路. 图中三角块装元件代表 74LS14 的一个非门, 其内部结构和 74LS04 相似:



### 3.3.2 中文字库液晶显示模块 12864-12

## 1、液晶显示模块概述

液晶显示模块是 $128 \times 64$ -12点阵的汉字图形型液晶显示模块，可显示汉字及图形，内置8192个中文汉字（16X16点阵）、128个字符（8X16点阵）及64X256点阵显示RAM（GDRAM）。可与CPU直接接口，提供两种界面来连接微处理机：8位并行及串行两种连接方式。具有多种功能：光标显示、画面移位、睡眠模式等。

## 2、模块引脚说明

引脚	名称	方向	说明	引脚	名称	方向	说明
1	VSS	-	GND (0V)	11	DB4	I	数据 4
2	VDD	-	Supply Voltage For Logic (+5v)	12	DB5	I	数据 5
3	V0	-	(+5v)	13	DB6	I	数据 6
4	RS (CS)	0	H: Data L: Instruction Code	14	DB7	I	数据 7
5	R/W (SID)	0	H: Read L: Write	15	PSB	0	H: Parallel Mode L: Serial Mode 串口低电屏
6	E (SCLK)	0	Enable Signal	16	NC	-	空脚
7	DB0	I	数据 0	17	/RST	0	Reset Signal 低电平有效
8	DB1	I	数据 1	18	NC	-	空脚
9	DB2	I	数据 2	19	LEDA	-	背光源正极 (LED+5V)
10	DB3	I	数据 3	20	LEDK	-	背光源负极 (LED-0V)

## 3、具体指令介绍：

### (1)、清除显示

CODE: RW RS DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

L	L	L	L	L	L	L	L	L	H
---	---	---	---	---	---	---	---	---	---

功能：清除显示屏幕，把 DDRAM 位址计数器调整为“00H”

### (2)、位址归位

CODE: RW RS DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

L	L	L	L	L	L	L	L	H	X
---	---	---	---	---	---	---	---	---	---

功能：把 DDRAM 位址计数器调整为“00H”，游标回原点，该功能不影响显示 DDRAM

### (3)、位址归位

CODE: RW RS DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

L	L	L	L	L	L	L	H	I/D	S
---	---	---	---	---	---	---	---	-----	---

功能：执行该命令后，所设置的行将显示在屏幕的第一行。显示起始行是由 Z 地址计数器控制的，该命令自动将 A0-A5 位地址送入 Z 地址计数器，起始地址可以是 0-63 范围内任意一行。Z 地址计数器具有循环计数功能，用于显示行扫描同步，当扫描完一行后自动加一。

(4)、显示状态 开/关

CODE: RW RS DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

L	L	L	L	L	L	H	D	C	B
---	---	---	---	---	---	---	---	---	---

功能: D=1; 整体显示 ON C=1; 游标 ON B=1; 游标位置 ON

(5)、游标或显示移位控制

CODE: RW RS DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

L	L	L	L	L	H	S/C	R/L	X	X
---	---	---	---	---	---	-----	-----	---	---

功能: 设定游标的移动与显示的移位控制位: 这指令并不改变 DDRAM 的内容

(6)、功能设定

CODE: RW RS DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

L	L	L	L	H	DL	X	0 RE	X	X
---	---	---	---	---	----	---	------	---	---

功能: DL=1 (必须设为 1) RE=1; 扩充指令集动作 RE=0: 基本指令集动作

(7)、设定 CGRAM 位址

CODE: RW RS DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

L	L	L	H	AC5	AC4	AC3	AC2	AC1	AC0
---	---	---	---	-----	-----	-----	-----	-----	-----

功能: 设定 CGRAM 位址到位址计数器 (AC)

(8)、设定 DDRAM 位址

CODE: RW RS DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

L	L	H	AC6	AC5	AC4	AC3	AC2	AC1	AC0
---	---	---	-----	-----	-----	-----	-----	-----	-----

功能: 设定 DDRAM 位址到位址计数器 (AC)

(9)、读取忙碌状态 (BF) 和位址

CODE: RW RS DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

L	H	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0
---	---	----	-----	-----	-----	-----	-----	-----	-----

功能: 读取忙碌状态 (BF) 可以确认内部动作是否完成, 同时可以读出位址计数器 (AC) 的值

(10)、写资料到 RAM

CODE: RW RS DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

H	L	D7	D6	D5	D4	D3	D2	D1	D0
---	---	----	----	----	----	----	----	----	----

功能: 写入资料到内部的 RAM (DDRAM/CGRAM/TRAM/GDRAM)

(11)、读出 RAM 的值

CODE: RW RS DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

H	H	D7	D6	D5	D4	D3	D2	D1	D0
---	---	----	----	----	----	----	----	----	----

功能: 从内部 RAM 读取资料 (DDRAM/CGRAM/TRAM/GDRAM)

(12)、待命模式 (12H)

CODE: RW RS DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

L	L	L	L	L	L	L	L	L	H
---	---	---	---	---	---	---	---	---	---

功能：进入待命模式，执行其他命令都可终止待命模式

(13)、卷动位址或 IRAM 位址选择 (13H)

CODE: RW RS DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

L	L	L	L	L	L	L	L	H	SR
---	---	---	---	---	---	---	---	---	----

功能：SR=1；允许输入卷动位址      SR=0；允许输入 IRAM 位址

(14)、反白选择 (14H)

CODE: RW RS DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

L	L	L	L	L	L	L	H	R1	R0
---	---	---	---	---	---	---	---	----	----

功能：选择 4 行中的任一行作反白显示，并可决定反白的与否

(15)、睡眠模式 (015H)

CODE: RW RS DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

L	L	L	L	L	L	H	SL	X	X
---	---	---	---	---	---	---	----	---	---

功能：SL=1；脱离睡眠模式    SL=0；进入睡眠模式

(16)、扩充功能设定 (016H)

CODE: RW RS DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

L	L	L	L	H	H	X	1 RE	G	L
---	---	---	---	---	---	---	------	---	---

功能：RE=1；扩充指令集动作    RE=0；基本指令集动作    G=1；绘图显示 ON  
G=0；绘图显示 OFF

(17)、设定 IRAM 位址或卷动位址 (017H)

CODE: RW RS DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

L	L	L	H	AC5	AC4	AC3	AC2	AC1	AC0
---	---	---	---	-----	-----	-----	-----	-----	-----

功能：SR=1；AC5~AC0 为垂直卷动位址    SR=0；AC3~AC0 写 ICONRAM 位址

(18)、设定绘图 RAM 位址 (018H)

CODE: RW RS DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

L	L	H	AC6	AC5	AC4	AC3	AC2	AC1	AC0
---	---	---	-----	-----	-----	-----	-----	-----	-----

功能：设定 GDRAM 位址到位址计数器 (AC)

#### 4、显示步骤

(1)、显示资料 RAM (DDRAM)

显示资料 RAM 提供  $64 \times 2$  个位元组的空间，最多可以控制 4 行 16 字 (64 个字) 的中文字型显示，当写入显示资料 RAM 时，可以分别显示 CGROM、HCGROM 与 CGRAM 的字型；ST7920A 可以显示三种字型，分别是半宽的 HCGROM 字型、CGRAM 字型及中文 CGROM 字型，三种字型的选择，由在 DDRAM 中写入的编码选择，在 0000H—0006H 的编码中将自动的结合下一个位元组，组成两个位元组的编码达成中文字型的编码 (A140—D75F)，各种字型详细编码如下：

显示半宽字型：将 8 位元资料写入 DDRAM 中，范围为 02H—7FH 的编码。

显示 CGRAM 字型：将 16 位元资料写入 DDRAM 中，总共有 0000H, 0002H, 0004H, 0006H 四种编码。

显示中文字形：将 16 位元资料写入 DDRAMK，范围为 A1A1H—F7FEH 的编码。

## (2)、绘图 RAM (GDRAM)

绘图显示 RAM 提供  $64 \times 32$  个位元组的记忆空间，最多可以控制  $256 \times 64$  点的二维也纳绘图缓冲空间，在更改绘图 RAM 时，先连续写入水平与垂直的坐标值，再写入两个 8 位元的资料到绘图 RAM，而地址计数器 (AC) 会自动加一；在写入绘图 RAM 的期间，绘图显示必须关闭，整个写入绘图 RAM 的步骤如下：

关闭绘图显示功能。

先将水平的位元组坐标 (X) 写入绘图 RAM 地址；

再将垂直的坐标 (Y) 写入绘图 RAM 地址；

将 D15—D8 写入到 RAM 中；

将 D7—D0 写入到 RAM 中；

打开绘图显示功能。

### 3. 4. 3 LM386 芯片

LM386 是一种音频集成功放，具有自身功耗低、电压增益可调整、电源电压范围大、外接元件少和总谐波失真小等优点，广泛应用于录音机和收音机之中。LM386 是美国国家半导体公司生产的音频功率放大器，主要应用于低电压消费类产品。为使外围元件最少，电压增益内置为 20。但在 1 脚和 8 脚之间增加一只外接电阻和电容，便可将电压增益调为任意值，直至 200。输入端以地位参考，同时输出端被自动偏置到电源电压的一半，在 6V 电源电压下，它的静态功耗仅为 24mW，使得 LM386 特别适用于电池供电的场合。LM386 的封装形式有塑封 8 引线双列直插式和贴片式。与通用型集成运放相类似，它是一个三级放大电路。

第一级为差分放大电路，T1 和 T3、T2 和 T4 分别构成复合管，作为差分放大电路的放大管；T5 和 T6 组成镜像电流源作为 T1 和 T2 的有源负载；T3 和 T4 信号从管的基极输入，从 T2 管的集电极输出，为双端输入单端输出差分电路。使用镜像电流源作为差分放大电路有源负载，可使单端输出电路的增益近似等于双端输出电容的增益。

第二级为共射放大电路，T7 为放大管，恒流源作有源负载，以增大放大倍数。

第三级中的 T8 和 T9 管复合成 PNP 型管，与 NPN 型管 T10 构成准互补输出级。二极管 D1 和 D2 为输出级提供合适的偏置电压，可以消除交越失真。

引脚 2 为反相输入端，引脚 3 为同相输入端。电路由单电源供电，故为 OTL 电路。输出端（引脚 5）应外接输出电容后再接负载。

---

电阻 R7 从输出端连接到 T2 的发射极，形成反馈通路，并与 R5 和 R6 构成反馈网络，从而引入了深度电压串联负反馈，使整个电路具有稳定的电压增益。

### 3. 4. 5 DS18B20

DS18B20 为数字温度传感器，接线方便，封装成后可应用于多种场合。封装后的 DS18B20 可用于电缆沟测温，高炉水循环测温，锅炉测温，机房测温，农业大棚测温，洁净室测温，弹药库测温等各种非极限温度场合。耐磨耐碰，体积小，使用方便，封装形式多样，适用于各种狭小空间设备数字测温和控制领域。

#### 1: 技术性能描述

- ①、独特的单线接口方式，DS18B20 在与微处理器连接时仅需要一条口线即可实现微处理器与 DS18B20 的双向通讯。
- ②、测温范围  $-55^{\circ}\text{C} \sim +125^{\circ}\text{C}$ ，固有测温误差（注意，不是分辨率，这里之前是错误的） $0.5^{\circ}\text{C}$ 。
- ③、支持多点组网功能，多个 DS18B20 可以并联在唯一的三线上，最多只能并联 8 个，实现多点测温，如果数量过多，会使供电电源电压过低，从而造成信号传输的不稳定。
- ④、工作电源: 3~5V/DC （可以数据线寄生电源）
- ⑤、在使用中不需要任何外围元件
- ⑥、测量结果以 9~12 位数字量方式串行传送
- ⑦、不锈钢保护管直径  $\Phi 6$
- ⑧、适用于 DN15~25, DN40~DN250 各种介质工业管道和狭小空间设备测温
- ⑨、标准安装螺纹 M10X1, M12X1.5, G1/2" 任选
- ⑩、PVC 电缆直接出线或德式球型接线盒出线，便于与其它电器设备连接

## 4 系统调试

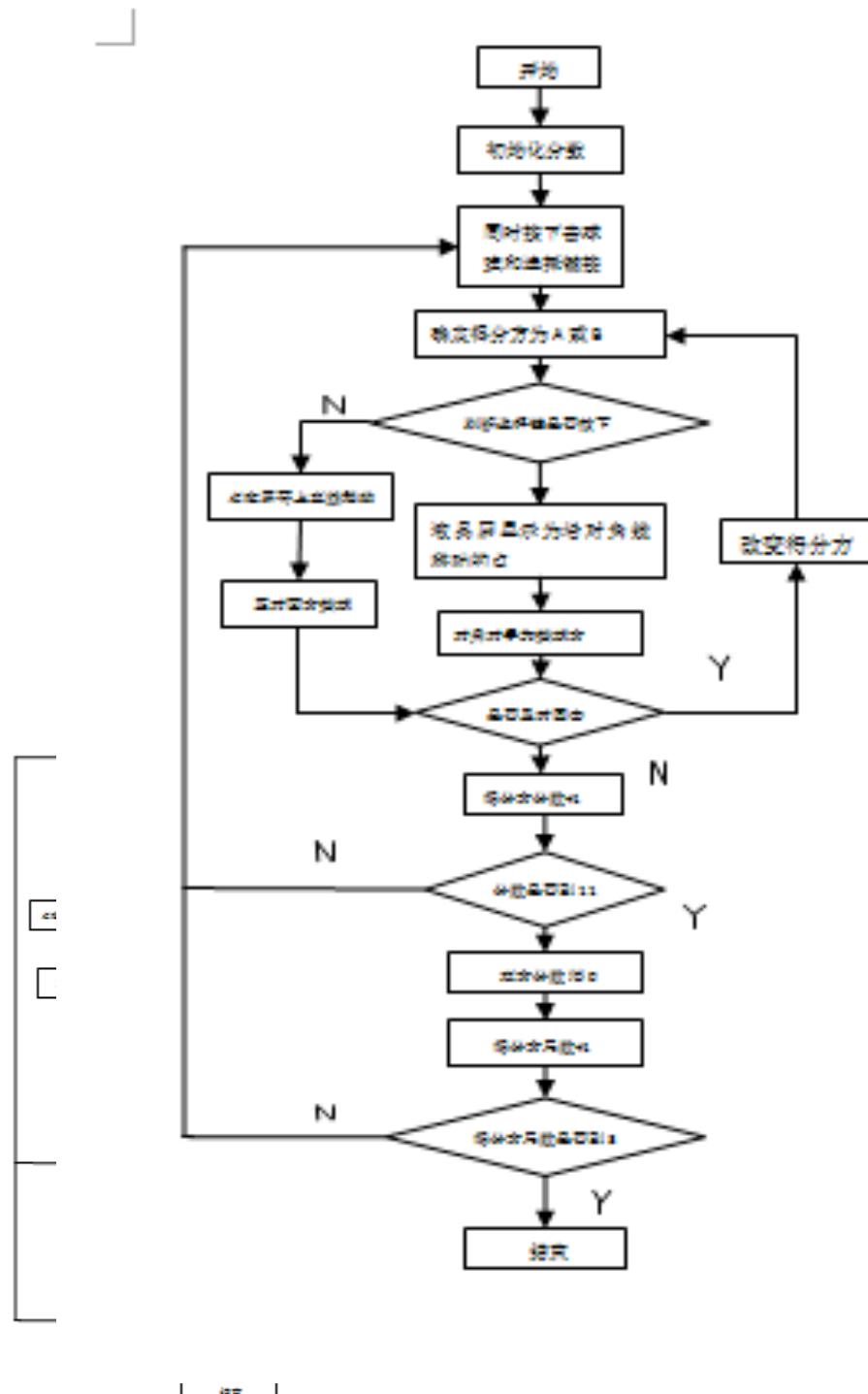
硬件布线焊接后需要通过万用表完成对硬件焊接的测试以及用仿真器对整机的仿真。从中我们可以发现了一些焊接，程序上的错误，便于我们及时发现错误并加以改正，提高效率，因此排错过程要足够重视，必须细心耐心。

- 1) 用万用表测试硬件电路，测定是否有断路，短路等故障，如导线断开，开关等原件虚焊，同时注意地线和火线不能相接。
- 2) 对照自己设计的电路图，认真核实硬件电路的正确性，避免应该接地或者接电源的线路漏接，或者粗心把元件焊接到错误的地方。对电容，电阻数值进行核对，最好用万用表对各电阻阻值进行检测，避免因此造成故障。
- 3) 编写好程序后利用仿真器进行仿真，发现不足和错误。对没有头绪的问题可以进行单步调试，逐步找到问题所在。
- 4) 用 WAVE6000 仿真器进行调试时，发现程序和硬件上都有多个功能不能正常实现，通过对软件和硬件两方面对问题进行分析，并尝试解决，通过不断努力，使之能正常实现各功能。

5 软件设计

5.1 软件设计原理

为了便于调试，本系统的软件主要采用模块化结构设计，对于基础的乒乓球游戏具有以下简要环节。流程图如下：



## 6 电子综合设计总结

为期三周的电子综合设计就这么结束了，三周的时间说短也不短，说长却也紧迫。这三周，我们从电路原理图的确定，硬件设计与连线再到软件编写，直至最后的调试过程，一天天都过得忙碌而充实。三周的时间我们学到了很多东西。

由于想要实现的功能颇多，编程量也大幅增多，因此舍弃了 STC89C52 单片机，选择了 FLASH 较大的 STC89C516 以 STC89C54 单片机为核心，充分利用了 STC89C54 有限的外围引脚资源，实现了对“乒乓球”游戏机的左右操作及比分显示的控制。整个系统通过按键可以很方便地操作，且运行稳定，抗干扰能力强。真正实现了娱乐产品的智能化。

本次课程设计是理论与实践相结合的过程，做出满意的成品必须将软硬件结合在一起，不能顾此失彼。这次设计不仅是对以往所学的回顾和应用，也是对未知知识探求的过程。不仅加强了我们的动手能力。而且更让我们在独立思考中加深了对单片机技术的理解。具体收获可简要概括为以下几点：

(1) 增强了自己搜索有用知识的能力。这次的设计中涉及到很多知识的应用，面对自己有限的知识储存，查找资料现学现用无疑是必须的。上网查找便捷又迅速，资源也相对丰富，对整个设计过程有很大的帮助。但是面对五花八门的资料，我们并不能盲目地照搬网上所有的，而应该根据所得资料学会如何分析问题，发现问题和解决问题的能力，将网上资料真正转化为自己的知识。学会有目的，高效率地学习。

(2) 能够很好的运用所学的数字电子技术、模拟电子技术及单片机编程等知识解决了一些问题，体会到了将知识用于实际的快乐感。

(3) 通过这次设计，我们还提高了自己的课程设计报告撰写水平，增强了运用计算机及互联网丰富的知识来设计和分析问题、解决问题独立完成工作任务的能力，为以后的学习工作中解决更多的、更困难的问题打好了很好的基础。

为期三周的课程设计得以顺利完成，首先要感谢我们组的指导老师谢礼莹老师，在她的耐心指导下我们才能够顺利完成此次基于单片机的乒乓球游戏机的设计。同时也要感谢校图书馆和电子综合实验室为我们提供了如此多的资料和工具，为我们节约了很多时间和精力。虽然本次设计还存在着很多不足，程序还可以更精简，硬件排线也可以更美观，但是作为一次收获良多的学习，还是为以后解决此类实际问题提供了一个模式，最后老师能在批阅中给予更多意见和建议，指导我们更上一层楼。

## 附录 1---参考文献

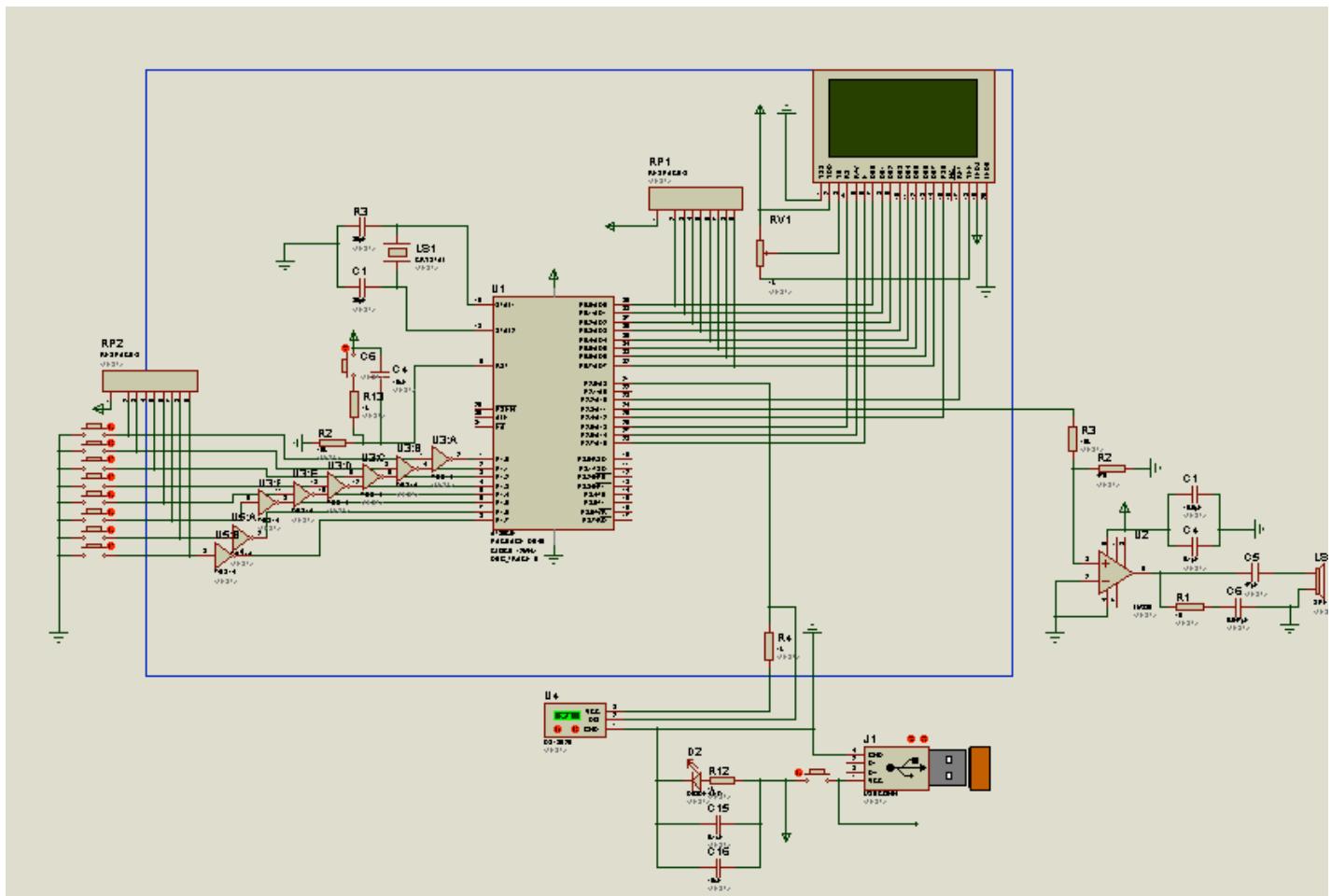
- [1]. 《51 单片机 C 语言开发详解》
- [2]. 《基于 Proteus 的单片机可视化软硬件仿真》
- [3]. 《单片机原理，应用与 Proteus 仿真》

## 附录 2---元器件的选择

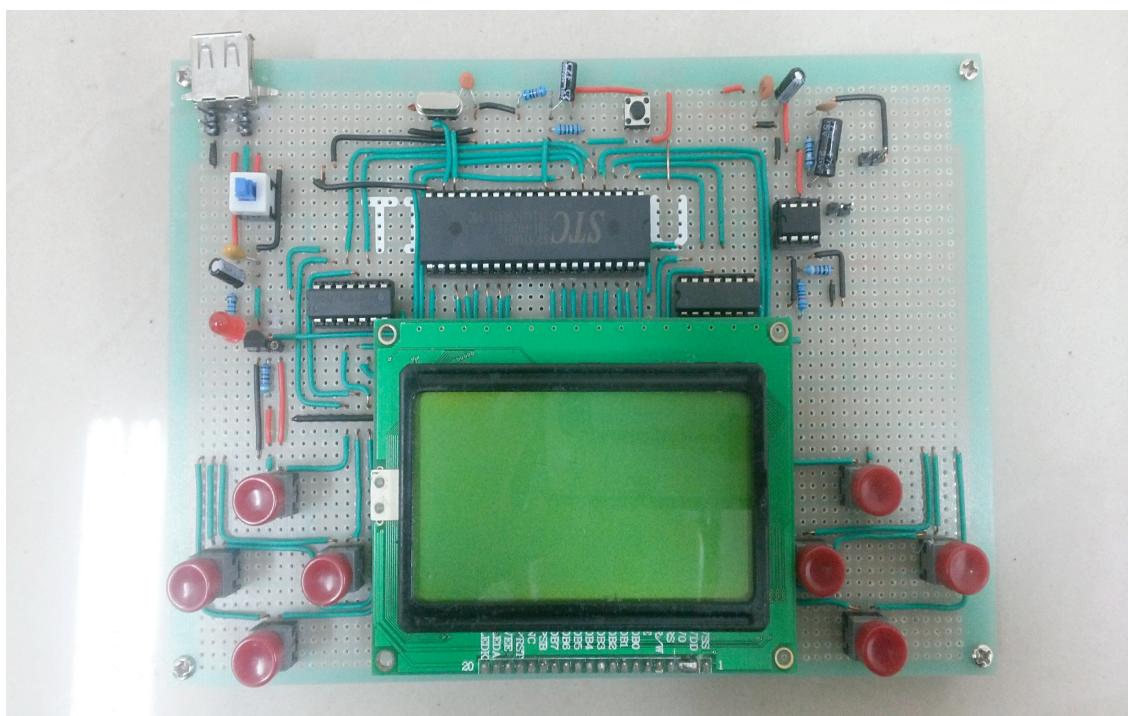
### 自动化学院《电子综合设计》元器件领用清单

元件名称	需要数量	元件名称	需要数量
STC89C516 单片机	1	电容 22pf	2
EDM12864-12	1	电容 0.047	2
LM386	1	电容 10uf (有极性的)	3
74ls14	2	电容 0.1uf	10
DS18B20	1	电容 47uf	2
电阻 1kΩ	5	矩阵按键开关	1
电阻 100Ω	1	按键开关(带锁)	1
电阻 960Ω	2	扬声器	1
电阻 470Ω	1	LED 发光二极管	1
电阻 100kΩ	5	按键开关 (带帽)	2
电阻 4.7kΩ	2		
电阻 10kΩ	5		
排阻 10kΩ	3		
12MHZ 晶振	1		
电源 5v	1		
滑动变阻器 20k	2		
方孔插针	2		
插针	2		
USB 接口	1		
		导线	若干
		跳线	若干
		底座	若干
		实验板：	1
		工具：	
		烙铁	1
		镊子	1
		斜口钳	1
		小起子	1

### 附录 3---电路图



附录 4---实物效果图：



## 附录 4---部分程序

```
#include "LCD12864_drv.h"
#include <reg52.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <intrins.h>

#define NN 113           //总路径点数

uint8 ball[2]={7, 26}; //球的实时路径
uint8 ball0[4]={7, 26, 119, 26}; //球的开始路径
//四个拍子的实时位置
    uint8 bat1[2]={3, 26};
    uint8 bat2[2]={123, 26};
    uint8 bat3[2]={3, 51};
    uint8 bat4[2]={123, 51};

//四个拍子的初始位置
    uint8 bat10[2]={3, 26};
    uint8 bat20[2]={123, 26};
    uint8 bat30[2]={3, 51};
    uint8 bat40[2]={123, 51};

    uint8 sel[4][2]={26, 8, 26, 23, 26, 39, 26, 54};//界面中的乒乓球位置
    uint8 speed[]={250, 200, 170, 140, 100, 60, 20, 9, 3, 1};//界面中选项的
标记(最终为游戏速度(延时))

    uint8 pos;
    uint8 select;
    uint8 xian;
    uint8 sit=0;           //记录球的位置
    uint8 ctrl=0;          //记录游戏的运行程度
    uint8 grade[4]={0}; //记录游戏比分的数组 前两个是小分 后两个总分
    uint8 dly=0;           //显示速度分级
    uint8 jjj=0;           //用于标记是否有变动
    uint8 players; //用于标记游戏者人数(0 为单打, 1 为双打)
    uint8 ff=4;           //检测按键在球不同运行状态下的状态
//    uint8 mm=0;
```







```
uint8 Russia();
uint8 temperature();
void screen_protect();
void danda_ctrl();
void wall_2(uint8 z);
void wall_1(uint8 z);
void wall1();
void wall2();
uint8 die_scan2();
void huxian_23();
void huxian_32();
void huxian_14();
void huxian_41();
void xiexian_23();
void xiexian_32();
void xiexian_14();
void xiexian_41();
void zhixian_12();
void zhixian_21();
void zhixian_34();
void zhixian_43();
void shuangda_ctrl();
void music_faqiu();
void music_shiwu();
void music_jujieshu();
static unsigned long Seed = 1;
#define A 48271L
#define M 2147483647L
#define Q (M / A)
#define R (M % A)
sbit K1=P1^3;
sbit K2=P1^0;
sbit K3=P1^2;
sbit K4=P1^1;
```

```
sbit K5=P1^4;
sbit K6=P1^7;
sbit K7=P1^6;
sbit K8=P1^5;
unsigned int idata num[19+2]={
    0xffff, //第 1 行，最下面
    0x801, 0x801, 0x801, 0x801, 0x801, 0x801, 0x801, 0x801, 0x801,
    0x801, 0x801, 0x801, 0x801, 0x801, 0x801, 0x801, 0x801, //第 2 行到第 20
    行共 19 行
    0xffff //第 21 行，最上面
}; //定义共 21 行，其中 num[0] 为下墙壁行，num[20] 为上墙壁行，每行 12 格，
//最左一格为左墙壁列，最右一格为右墙壁列
unsigned char code Block[28][2]={
/*
 *   口     口口口   口口
 *   口     口         口     口
 *   口口        口   口口口
 */
{0x88, 0xc0}, {0xe8, 0x00}, {0x62, 0x20}, {0x02, 0xe0},
/*
 *   口       口口 口口口
 *   口 口     口       口
 *   口口 口口口 口
 */
{0x22, 0x60}, {0x08, 0xe0}, {0xc8, 0x80}, {0xe2, 0x00},
/*
 *   口
 *   口口      口口
 *   口       口口
 */
{0x8c, 0x40}, {0x6c, 0x00}, {0x8c, 0x40}, {0x6c, 0x00},
/*
 *   口       口口
 *   口口      口口
 */
}
```

```
* 口
*/
{0x4c, 0x80}, {0xc6, 0x00}, {0x4c, 0x80}, {0xc6, 0x00},
/*
*      口          口
*  口    口口  口口口  口口
* 口口口  口      口      口
*/
{0x04, 0xe0}, {0x8c, 0x80}, {0xe4, 0x00}, {0x26, 0x20},
/*口
* 口
* 口      口口口口
* 口
*/
{0x44, 0x44}, {0x0f, 0x00}, {0x44, 0x44}, {0x0f, 0x00},
/*
* 口口
* 口口
*/
{0x06, 0x60}, {0x06, 0x60}, {0x06, 0x60}, {0x06, 0x60}};

#define PASSSCORE 20
struct Jimu
{
    unsigned int dat;
    char x;
    unsigned char y;
    unsigned char type;
    unsigned char change}Sign[3];//积木结构体
unsigned char SysFlag=0;
#define NEWSIGNFLAG 0
#define DEADFLAG 1
#define PAUSEFLAG 2
unsigned char Score=0;
unsigned char Level=1;
unsigned char DelayCnt=5;
```

```
*****  
#define N 25  
*****  
*****  
*****  
*****  
*****  
功能：延时函数  
输入参数：n：延时时长  
输出参数：无  
返回值：  
*****  
static void Delay(uint16 n)  
{    uint16 i;  
    while(n--)  
    {        i=200;  
        while(i--) ;}  
*****  
显示拍子函数      输入左列的为中心坐标      长为 9， 宽为 2  
*****  
void paizi(unsigned int x, unsigned int y, unsigned int z)  
{    Lcd_VertLine(x, y-5, 11, z);  
    Lcd_VertLine(x+1, y-5, 11, z)}  
*****  
显示乒乓球函数      输入左列的为中心坐标      x 在 2-125 之间  
y 在 2-61 之间      直径为 5  
*****  
void qiu(unsigned int x, unsigned int y, unsigned int z)  
{    Lcd_HoriLine(x-1, y-2, 3, z);  
    Lcd_HoriLine(x-2, y-1, 5, z);  
    Lcd_HoriLine(x-2, y, 5, z);  
    Lcd_HoriLine(x-2, y+1, 5, z);  
    Lcd_HoriLine(x-1, y+2, 3, z);}  
*****  
文字、乒乓球桌、(乒乓球、)球拍布局
```

```
*****  
void configuration1()  
{    // 画乒乓球案子  
    Lcd_WriteStr(0, 0, " 00:00  0:0 ");  
    Lcd_VertLine(0, 14, 50, 1);  
    Lcd_VertLine(127, 14, 50, 1);  
    Lcd_HoriLine(0, 14, 127, 1);  
    Lcd_HoriLine(0, 63, 127, 1);  
//    Lcd_HoriLine(7, 38, 114, 1);  
/*****  
清除原乒乓球函数  
*****  
void clear_ball(unsigned int x, unsigned int y)  
{    Lcd_PutPixel(x-1, y-2, 0);  
    Lcd_PutPixel(x, y-2, 0);  
    Lcd_PutPixel(x+1, y-2, 0);  
    Lcd_PutPixel(x-2, y-1, 0);  
    Lcd_PutPixel(x-1, y, 0);  
    Lcd_PutPixel(x, y-1, 0);  
    Lcd_PutPixel(x+1, y-1, 0);  
    Lcd_PutPixel(x+2, y-1, 0);  
    Lcd_PutPixel(x-2, y, 0);  
    Lcd_PutPixel(x-1, y, 0);  
    Lcd_PutPixel(x, y, 0);  
    Lcd_PutPixel(x+1, y, 0);  
    Lcd_PutPixel(x+2, y, 0);  
    Lcd_PutPixel(x-2, y+1, 0);  
    Lcd_PutPixel(x-1, y+1, 0);  
    Lcd_PutPixel(x, y+1, 0);  
    Lcd_PutPixel(x+1, y+1, 0);  
    Lcd_PutPixel(x+2, y+1, 0);  
    Lcd_PutPixel(x-1, y+2, 0);  
    Lcd_PutPixel(x, y+2, 0);  
    Lcd_PutPixel(x+1, y+2, 0);}
```

```
/****************************************
单双打选择
****************************************/
void tabletennis_1or2()
{
    jjj=0;
    select=0;
    PictureDisplay12864(pic_danda);
    while(1)
    {
        if((K1==1) || (K2==1))
        {
            if(select==0)
            {
                select++; //下一项}
            else if(select==1)
            {
                select=0; }
            jjj=1;}
            if(K8==1)
            {
                Reset12864(); //液晶 12864 初始化
                Delay(20); //延时, 保证显示更稳定
                ScreenClear12864(); //清屏
                Lcd_Clear(0);
                Delay(20);
                if(select==0)
                {
                    Reset12864(); //液晶 12864 初始化
                    Delay(20); //延时, 保证显示更稳定
                    ScreenClear12864(); //清屏
                    Lcd_Clear(0);
                    Delay(20);
                    players=0;
                    danda_ctrl(); }
            }
            else
            {
                Reset12864(); //液晶 12864 初始化
                Delay(20); //延时, 保证显示更稳定
                ScreenClear12864(); //清屏
                Lcd_Clear(0); }
```

```
        Delay(20);
        players=1;
        shuangda_ct; } }

if (K7==1)
{
    Reset12864();           //液晶 12864 初始化
    Delay(20);             //延时,保证显示更稳定
    ScreenClear12864();    //清屏
    Lcd_Clear(0);
    Delay(20);
    tabletennis_speed();   }

// 选择单打 or 双打
if (jjj==1)
{
    if (select==0)
    {
        Lcd_Clear(0);
        Delay(20);
        PictureDisplay12864(pic_danda);

    else if (select==1)
    {
        Lcd_Clear(0);
        Delay(20);
        PictureDisplay12864(pic_shuangda);

    jjj=0;
}
Delay(100); }
```

\*\*\*\*\*

速度选择

\*\*\*\*\*

```
void tabletennis_speed()
{
    dly=0;
    PictureDisplay12864(pic_youxisudu);
    while(1)
    {
        if (K1==1)
        {
            if (dly<9)
            {
                dly++;           //下一项
            else if (dly==9)
            {
                dly=0;
```

```

        }

    }

    if(K2==1)
    {
        if(dly==0)
        {
            dly=9;           //下一项}
        else if(dly>=1)
        {
            dly--;         //下一项  }
        if(K8==1)
        {
            Reset12864();           //液晶 12864 初始化
            Delay(20);             //延时, 保证显示更稳定
            ScreenClear12864();     //清屏
            Lcd_Clear(0);
            Delay(20);

        tabletennis_1or2();      }
    }

    if(K7==1)
    {
        Reset12864();           //液晶 12864 初始化
        Delay(20);             //延时, 保证显示更稳定
        ScreenClear12864();     //清屏
        Lcd_Clear(0);
        Delay(20);
        fun_select();          }

// 显示速度分级
if(dly==0)
{
    StringDisplay12864(0, 2, "0");
    StringDisplay12864(0, 2, "0");
}

else if(dly==1)
{
    StringDisplay12864(0, 2, "1");
    StringDisplay12864(0, 2, "1");
}

else if(dly==2)
{
    StringDisplay12864(0, 2, "2");
    StringDisplay12864(0, 2, "2");
}

else if(dly==3)
{
    StringDisplay12864(0, 2, "3");
}

```

```

        StringDisplay12864(0, 2, "         3      ");
    }
else if(dly==4)
{
    StringDisplay12864(0, 2, "         ");
    StringDisplay12864(0, 2, "         4      ");
}
else if(dly==5)
{
    StringDisplay12864(0, 2, "         ");
    StringDisplay12864(0, 2, "         5      ");
}
else if(dly==6)
{
    StringDisplay12864(0, 2, "         ");
    StringDisplay12864(0, 2, "         6      ");
}
else if(dly==7)
{
    StringDisplay12864(0, 2, "         ");
    StringDisplay12864(0, 2, "         7      ");
}
else if(dly==8)
{
    StringDisplay12864(0, 2, "         ");
    StringDisplay12864(0, 2, "         8      ");
}
else if(dly==9)
{
    StringDisplay12864(0, 2, "         ");
    StringDisplay12864(0, 2, "         9      ");
}

Delay(100);      }

/*****************
音乐播放
*****************/
void music_play()
{
    music_wing();
    music_kanongxuqu();
    music_tiankongzhicheng();
    music_longmao();

    Reset12864();           //液晶 12864 初始化
    Delay(20);              //延时, 保证显示更稳定
    ScreenClear12864();     //清屏
    Lcd_Clear(0);
    Delay(20);
}

```

```
PictureDisplay12864(pic_jiemian1);
fun_select();
/***********************/
功能选择
********************/
void fun_select()
{
    pos=0;
    PictureDisplay12864(pic_jiemian1);
    while(1)
    {
        if(K2==1)
        {
            qiu(sel[pos][0], sel[pos][1], 0); //有变化则清除原乒乓球
            if(pos<3)
            {
                pos++;                                //下一项
            }
            else if(pos==3)
            {
                pos=0;                                }
        }
        if(K1==1)
        {
            qiu(sel[pos][0], sel[pos][1], 0); //有变化则清除原乒乓球
            if(pos==0)
            {
                pos=3;                                //下一项}
            else if(pos>=1)
            {
                pos--;                                //下一项
            }
        }
        if(K7==1)                                //屏幕保护程序
        {
            Reset12864();                         //液晶 12864 初始化
            Delay(20);                            //延时，保证显示更稳定
            ScreenClear12864();                  //清屏
            Lcd_Clear(0);
            Delay(20);
            while(1)
            {
                screen_protect(); //屏幕保护程序
            }
        }
    }
}
```

```
    }

    if(K8==1)
    {
        Reset12864();           //液晶 12864 初始化
        Delay(20);             //延时, 保证显示更稳定

        ScreenClear12864();     //清屏
        Lcd_Clear(0);
        Delay(20);

        if(pos==0)              //乒乓球游戏
        {
            Reset12864();       //液晶 12864 初始化
            Delay(20);           //延时, 保证显示更稳定
            ScreenClear12864();   //清屏
            Lcd_Clear(0);
            Delay(20);

            PictureDisplay12864(pic_tabletennis);
            Delay(500);

            Reset12864();           //液晶 12864 初始化
            Delay(20);             //延时, 保证显示更稳定
            ScreenClear12864();     //清屏
            Lcd_Clear(0);
            Delay(20);
        }
    }

    while(1)
    {
        tabletennis_speed();
    }

    if(pos==1)                //俄罗斯方块
    {
        Reset12864();           //液晶 12864 初始化
        Delay(20);             //延时, 保证显示更稳定
        ScreenClear12864();     //清屏
        Lcd_Clear(0);
        Delay(20);

        BeepI0=0;
        Russia();
        BeepI0=0;               }
    }

    if(pos==2)                //音乐播放器
    {
        Reset12864();           //液晶 12864 初始化
```

```

        Delay(20);           //延时,保证显示更稳定
        ScreenClear12864();   //清屏
        Lcd_Clear(0);
        Delay(20);
        music_play();          }

    if(pos==3)           //数字温度计
    {
        Reset12864();      //液晶 12864 初始化
        Delay(20);           //延时,保证显示更稳定
        ScreenClear12864();   //清屏
        Lcd_Clear(0);
        Delay(20);
        PictureDisplay12864(pic_temperature);
        temperature();          }
    }

// 显示乒乓球位置
    qiu(sel[pos][0], sel[pos][1], 1);
    Delay(100); }           //球向右飞行时,左边玩家的墙大招
    void wall2()
{
    wall_2(1);
    Delay(200);
    wall_2(0);
    paizi(bat2[0], bat2[1], 1);      // 画乒乓球球拍 }

//球向左飞行时,右边玩家的墙大招
void wall1()
{
    wall_1(1);
    Delay(200);
    wall_1(0);
    paizi(bat1[0], bat1[1], 1); // 画乒乓球球拍
}

} /*****乒乓球拍移动的函数*****/
*****paizi_move()*****/
void paizi_move()
{
    if(K1==1)
    {
        bat2[1]--;
        Lcd_PutPixel(bat2[0], (bat2[1]+6), 0);
    }
}

```

```
Lcd_PutPixel(bat2[0], (bat2[1]-4), 1);  
Lcd_PutPixel((bat2[0]+1), (bat2[1]+6), 0);  
Lcd_PutPixel((bat2[0]+1), (bat2[1]-4), 1);  
Delay(10);  
  
if(K2==1)  
{      bat2[1]++;  
    Lcd_PutPixel(bat2[0], (bat2[1]-6), 0);  
    Lcd_PutPixel(bat2[0], (bat2[1]+4), 1);  
    Lcd_PutPixel((bat2[0]+1), (bat2[1]-6), 0);  
    Lcd_PutPixel((bat2[0]+1), (bat2[1]+4), 1);  
    Delay(10);  
  
if(K5==1)  
{      bat1[1]--;  
    Lcd_PutPixel(bat1[0], (bat1[1]+6), 0);  
    Lcd_PutPixel(bat1[0], (bat1[1]-4), 1);  
    Lcd_PutPixel((bat1[0]+1), (bat1[1]+6), 0);  
    Lcd_PutPixel((bat1[0]+1), (bat1[1]-4), 1);  
    Delay(10);  
  
if(K6==1)  
{      bat1[1]++;  
    Lcd_PutPixel(bat1[0], (bat1[1]-6), 0);  
    Lcd_PutPixel(bat1[0], (bat1[1]+4), 1);  
    Lcd_PutPixel((bat1[0]+1), (bat1[1]-6), 0);  
    Lcd_PutPixel((bat1[0]+1), (bat1[1]+4), 1);  
    Delay(10); } }  
  
/*********************************************  
乒乓球游戏结束后的函数  
*****  
/*  
void tabletennis_end()  
{      Reset12864();          //液晶 12864 初始化  
    Delay(20);           //延时, 保证显示更稳定  
    ScreenClear12864();   //清屏  
    Lcd_Clear(0);
```

```
Delay(20);
PictureDisplay12864(pic_gameover);
Delay(400);
    Reset12864();           //液晶 12864 初始化
Delay(20);           //延时, 保证显示更稳定
ScreenClear12864(); //清屏
Lcd_Clear(0);
Delay(20);

fun_select();}

*/
/*****这个球结束后的函数
*****
void tabletennis_gameover()
{
    qiu(ball[0], ball[1], 0);           //清除球的位置
    music_shiwu();                     //这个球结束后的声音
//    ff=die_scan2();
/*    if(((K1)&&(K2)&&(K3)&&(K4)&&(K5)&&(K6)&&(K7)&&(K8))==1)           //八
    {
        Reset12864();           //液晶 12864 初始化
        Delay(20);           //延时, 保证显示更稳定
        ScreenClear12864(); //清屏
        Lcd_Clear(0);
        Delay(150);
        fun_select();
    }*/
    if(xian<=5)
    {
        if(grade[0]<10)           //小局分改变
        {
            grade[0]++;
            switch(grade[0])
            {
                case 0: Lcd_WriteStr(1, 0, "00");break;
                case 1: Lcd_WriteStr(1, 0, "01");break;
                case 2: Lcd_WriteStr(1, 0, "02");break;
                case 3: Lcd_WriteStr(1, 0, "03");break;
                case 4: Lcd_WriteStr(1, 0, "04");break;
                case 5: Lcd_WriteStr(1, 0, "05");break;
            }
        }
    }
}
```

```
        case 6: Lcd_WriteStr(1, 0, "06");break;
        case 7: Lcd_WriteStr(1, 0, "07");break;
        case 8: Lcd_WriteStr(1, 0, "08");break;
        case 9: Lcd_WriteStr(1, 0, "09");break;
        case 10: Lcd_WriteStr(1, 0, "10");break;
        case 11: Lcd_WriteStr(1, 0, "11");
/*
if((grade[0]+grade[1]+grade[2]+grade[3])%2==0
{
    ball[0]=ball0[0];
    ball[1]=ball0[1];
}
else
{
    ball[0]=ball0[3];
    ball[1]=ball0[4];
}
*/
else
{
    Lcd_WriteStr(1, 0, "11");
    Delay(400);
    grade[2]++;
music_jujieshu()          //这个球结束后的声音
grade[0]=0;
switch(grade[2])
{
    case 0:
Lcd_WriteStr(5, 0, "0:");Lcd_WriteStr(1, 0, "00");break;
    case 1:
Lcd_WriteStr(5, 0, "1:");Lcd_WriteStr(1, 0, "00");break;
    case 2:
Lcd_WriteStr(5, 0, "2:");Lcd_WriteStr(1, 0, "00");}
if(grade[2]==2)
{
    Reset12864();      //液晶 12864 初始化
    Delay(20);         //延时,保证显示更稳定
    ScreenClear12864(); //清屏
    Lcd_Clear(0);
    Delay(20); //图标 甲方胜
    PictureDisplay12864(pic_jiafangsheng);
```

```
while(1)

    {if(((K1) || (K2) || (K3) || (K4) || (K5) || (K6) || (K7) || (K8))==1
    { Reset12864();           //液晶 12864 初始化
    Delay(20);              //延时, 保证显示更稳定
    ScreenClear12864();     //清屏
    Lcd_Clear(0);
    Delay(20);
    fun_select();}

/*                         if((grade[0]+grade[1]+grade[2]+grade[3])%2==0
{               ball[0]=ball0[0];
               ball[1]=ball0[1];
}
else
{               ball[0]=ball0[3];
               ball[1]=ball0[4];
}

*/
    }

    else if(xian>5)          //左边玩家
    {
        if(grade[1]<10)
        {
            grade[1]++;
            switch(grade[1])
            {
                case 0:
Lcd_WriteStr(2, 0, ":0");Lcd_WriteStr(3, 0, "0 ");break;
                case 1:
Lcd_WriteStr(2, 0, ":0");Lcd_WriteStr(3, 0, "1 ");break;
                case 2:
Lcd_WriteStr(2, 0, ":0");Lcd_WriteStr(3, 0, "2 ");break;
                case 3:
Lcd_WriteStr(2, 0, ":0");Lcd_WriteStr(3, 0, "3 ");break;
                case 4:
Lcd_WriteStr(2, 0, ":0");Lcd_WriteStr(3, 0, "4 ");break;
            }
        }
    }
}
```

```
        case 5:  
Lcd_WriteStr(2, 0, ":0");Lcd_WriteStr(3, 0, "5 ");break;  
        case 6:  
Lcd_WriteStr(2, 0, ":0");Lcd_WriteStr(3, 0, "6 ");break;  
        case 7:  
Lcd_WriteStr(2, 0, ":0");Lcd_WriteStr(3, 0, "7 ");break;  
        case 8:  
Lcd_WriteStr(2, 0, ":0");Lcd_WriteStr(3, 0, "8 ");break;  
        case 9:  
Lcd_WriteStr(2, 0, ":0");Lcd_WriteStr(3, 0, "9 ");break;  
        case 10:  
Lcd_WriteStr(2, 0, ":1");Lcd_WriteStr(3, 0, "0 ");break;  
        case 11:  
Lcd_WriteStr(2, 0, ":1");Lcd_WriteStr(3, 0, "1 ");  
    }  
/*  
     if((grade[0]+grade[1]+grade[2]+grade[3])%2==0)  
     {  
         ball[0]=ball0[0];  
         ball[1]=ball0[1];  
     }  
     else  
     {  
         ball[0]=ball0[3];  
         ball[1]=ball0[4];  
     }  
*/  
*/  
    }else  
    {  
        Lcd_WriteStr(2, 0, ":1");Lcd_WriteStr(3, 0, "1 ");  
        Delay(400);  
        grade[1]=0;  
        grade[3]++;  
        music_jujieshu(); //这个球结束后的声音  
/*  
     if((grade[0]+grade[1]+grade[2]+grade[3])%2==0)  
     {  
         ball[0]=ball0[0];  
         ball[1]=ball0[1];  
     }  
     else
```

```

    {
        ball[0]=ball0[3];
        ball[1]=ball0[4];
    }

*/     switch(grade[3])
{
    case 0: Lcd_WriteStr(6, 0, "0
");Lcd_WriteStr(2, 0, ":0");Lcd_WriteStr(3, 0, "0 ");break;
    case 1: Lcd_WriteStr(6, 0, "1
");Lcd_WriteStr(2, 0, ":0");Lcd_WriteStr(3, 0, "0 ");break;
    case 2: Lcd_WriteStr(6, 0, "2
");Lcd_WriteStr(2, 0, ":0");Lcd_WriteStr(3, 0, "0 ");}

    if (grade[3]==2)
    {
        Reset12864();           //液晶 12864 初始化
        Delay(20);             //延时, 保证显示更稳定
        ScreenClear12864();     //清屏
        Lcd_Clear(0);
        Delay(20); //图标 乙方胜
        PictureDisplay12864(pic_yifangsheng);
        while(1)
        {

if(((K1) || (K2) || (K3) || (K4) || (K5) || (K6) || (K7) || (K8))==1)
{
    Reset12864();           //液晶 12864 初始化
    Delay(20);             //延时, 保证显示更稳定
    ScreenClear12864();     //清屏
    Lcd_Clear(0);
    Delay(20);

    fun_select();
}
}
}
}

Delay(150);
}
/*****

```

左到右 单打控制

```
*****  
void danda2_1()  
{  
    // 检测  
    if((K3==1) || (K4==1))  
    {  
        ff=0;  
        {  
            tabletennis_gameover();  
        }  
    }  
}  
  
void danda2_2()  
{  
    // 检测  
    if(K3==1)  
    {  
        if(((bat2[1]+5)>=ball1[1])&&(ball1[1]>=(bat2[1]-5)))  
        {  
            ff=1; //1 代表击球成功  
            if((xian==0) || (xian==4) || (xian==5)) //位置 2  
            {  
                huxian_23();  
            }  
            if((xian==1) || (xian==2) || (xian==3)) //位置 4  
            {  
                huxian_41();  
            }  
        }  
        else  
        {  
            ff=2; //2 代表死亡  
            tabletennis_gameover();  
            zhixian_12();  
        }  
    }  
    if(K4==1)  
    {  
        wall2();  
        ff=3; //3 代表击球成功  
        if((xian==0) || (xian==4) || (xian==5)) //位置 2  
        {  
            xiexian_23();  
        }  
    }  
    弧线  
    //斜线  
}
```

```
if((xian==1) || (xian==2) || (xian==3)) //位置 4
{
    xiexian_41();
}
}

void danda2_3()
{
// 检测
if(K3==1)
{
    if(((bat2[1]+5)>=ball[1])&&(ball[1]>=(bat2[1]-5)))
    {
        ff=1; //1 代表击球成功 弧线
        if((xian==0) || (xian==4) || (xian==5)) //位置 2
        {
            huxian_23();
        }
        if((xian==1) || (xian==2) || (xian==3)) //位置 4
        {
            huxian_41();
        }
    }
    else
    {
        ff=2; //2 代表死亡
        tabletennis_gameover();
        zhixian_12();
    }
}
if(K4==1)
{
    wall2();
    ff=3; //3 代表击球成功 斜线
    if((xian==0) || (xian==4) || (xian==5)) //位置 2
    {
        xiexian_23();
    }
    if((xian==1) || (xian==2) || (xian==3)) //位置 4
    {
        xiexian_41();
    }
}
```

```

if(((K3==0)&&(K4==0))&&((bat2[1]+5)>=ball[1])&&(ball[1]>=(bat
2[1]-5)))
{
    ff=0;                                //0 代表未按键
    if((xian==0) || (xian==4) || (xian==5)) //位置 2
    {
        zhixian_21();
    }
    if((xian==1) || (xian==2) || (xian==3)) //位置 4
    {
        zhixian_43();
    }
}
/*****右到左 单打控制*****
void danda1_1()
{// 检测
    if((K7==1) || (K4==8))
    {
        ff=0;
        {          tabletennis_gameover();
        }
    }
}
void danda1_2()
{// 检测
    if(K7==1)
    {
        if(((bat1[1]+5)>=ball[1])&&(ball[1]>=(bat1[1]-5)))
        {
            ff=1;                                //1 代表击球成功
            if((xian==6) || (xian==10) || (xian==11)) //位置 1
            {
                huxian_14();
            }
            if((xian==7) || (xian==8) || (xian==9)) //位置 3
            {
                huxian_32();
            }
        }
    }
}
弧线

```

```
else
{
    ff=2; //2 代表死亡
    tabletennis_gameover();
    zhixian_21();
}

if(K8==1)
{
    wall1();
    ff=3; //3 代表击球成功 斜线
    if((xian==6) || (xian==10) || (xian==11)) //位置 1
    {
        xiexian_14();
    }
    if((xian==7) || (xian==8) || (xian==9)) //位置 3
    {
        xiexian_32();
    }
}

void danda1_3()
{// 检测
if(K7==1)
{
    if(((bat1[1]+5)>=ball[1])&&(ball[1]>=(bat1[1]-5)))
    {
        ff=1; //1 代表击球成功 弧线
        if((xian==6) || (xian==10) || (xian==11)) //位置 1
        {
            huxian_14();
        }
        if((xian==7) || (xian==8) || (xian==9)) //位置 3
        {
            huxian_32();
        }
    }
    else
    {
        ff=2; //2 代表死亡
        tabletennis_gameover();
        zhixian_21();
    }
}
```

```

        }

    }

    if(K8==1)
    {
        wall1();
        ff=3;           //3 代表击球成功           斜线
        if((xian==6) || (xian==10) || (xian==11)) //位置 1
        {
            xiexian_14();
        }
        if((xian==7) || (xian==8) || (xian==9))           //位置 3
        {
            xiexian_32();
        }
    }

    if(((K7==0)&&(K8==0))&&((bat1[1]+5)>=ball1[1])&&(ball1[1]>=(bat
1[1]-5)))
    {
        ff=0;           //0 代表未按键
        if((xian==6) || (xian==10) || (xian==11)) //位置 1
        {
            zhixian_12();
        }
        if((xian==7) || (xian==8) || (xian==9))           //位置 3
        {
            zhixian_34();
        }
    }
}

/*****************************************
单打 *** 总控制
*****************************************/
void danda_ctrl()
{
    grade[0]=0;           //分数清零
    grade[1]=0;           //分数清零
    grade[2]=0;           //分数清零
    grade[3]=0;           //分数清零
    configuration1();      //整体布局
    paizi(bat10[0], bat10[1], 1); // 画乒乓球球拍
    paizi(bat20[0], bat20[1], 1);
    qiu(ball10[0], ball10[1], 1); //显示球的位置
}

```

```
while(1)
{
    if((K5==1)&&(K6==1)&&(K7==1)&&(K8==1)) //发球
    {
        music_faqiu(); //发球的声音
        zhixian_12();
        Delay(100);
    }
}

/*********************************************
左到右 双打控制
*****/



void shuangda2_1()
{//
    检测
    if((K1==1) || (K2==1) || (K3==1) || (K4==1))
    {
        ff=0;
        {
            tabletennis_gameover();
        }
    }
}

void shuangda2_2()
{//
    检测
    if((K3==1)&&(K1==1))
    {
        ff=1; //1 代表击球成功 弧线
        if((xian==0) || (xian==4) || (xian==5)) //位置 2
        {
            huxian_23();
        }
        else
        {
            ff=2; //2 代表死亡
            tabletennis_gameover();
            if((xian==0) || (xian==4) || (xian==5)) //位置 2
            {
                zhixian_21();
            }
            else if((xian==1) || (xian==2) || (xian==3)) //位置 4
            {
                zhixian_43();
            }
        }
    }
}
```

```
    }
}

else if((K3==1)&&(K2==1))
{
    ff=1; //1 代表击球成功 弧线
    if((xian==1) || (xian==2) || (xian==3)) //位置 4
    {
        huxian_41();
    }
    else
    {
        ff=2; //2 代表死亡
        tabletennis_gameover();
        if((xian==0) || (xian==4) || (xian==5)) //位置 2
        {
            zhixian_21();
        }
        else if((xian==1) || (xian==2) || (xian==3)) //位置 4
        {
            zhixian_43();
        }
    }
}

else if((K4==1)&&(K1==1))
{
    ff=1; //1 代表击球成功 弧线
    if((xian==0) || (xian==4) || (xian==5)) //位置 2
    {
        xiexian_23();
    }
    else
    {
        ff=2; //2 代表死亡
        tabletennis_gameover();
        if((xian==0) || (xian==4) || (xian==5)) //位置 2
        {
            zhixian_21();
        }
        else if((xian==1) || (xian==2) || (xian==3)) //位置 4
        {
            zhixian_43();
        }
    }
}
```

```

else if((K4==1)&&(K2==1))
{
    ff=1; //1 代表击球成功 弧线
    if((xian==1) || (xian==2) || (xian==3)) //位置 4
    {
        xiexian_41();
    }
    else
    {
        ff=2; //2 代表死亡
        tabletennis_gameover();
        if((xian==0) || (xian==4) || (xian==5)) //位置 2
        {
            zhixian_21();
        }
        else if((xian==1) || (xian==2) || (xian==3)) //位置 4
        {
            zhixian_43();
        }
    }
}
else if((K1==1)&&(K2==0)&&(K3==0)&&(K4==0))
{
    ff=1; //1 代表击球成功 弧线
    if((xian==0) || (xian==4) || (xian==5)) //位置 2
    {
        zhixian_21();
    }
    else
    {
        ff=2; //2 代表死亡
        tabletennis_gameover();
        if((xian==0) || (xian==4) || (xian==5)) //位置 2
        {
            zhixian_21();
        }
        else if((xian==1) || (xian==2) || (xian==3)) //位置 4
        {
            zhixian_43();
        }
    }
}
else if((K1==0)&&(K2==1)&&(K3==0)&&(K4==0))
{
ff=1; //1 代表击球成功 弧线

```

```
if((xian==1) || (xian==2) || (xian==3)) //位置 4
{
    zhixian_43();
}
else
{
    ff=2; //2 代表死亡
    tabletennis_gameover();
    if((xian==0) || (xian==4) || (xian==5)) //位置 2
    {
        zhixian_21();
    }
}
else if((xian==1) || (xian==2) || (xian==3)) //位置 4
{
    zhixian_43();
}
}

//****************************************************************************
右到左 双打控制
*****
void shuangda1_1()
{// 检测
if((K5==1) || (K6==1) || (K7==1) || (K8==1))
{
    ff=0;
    {
        tabletennis_gameover();
    }
}
}

void shuangda1_2()
{// 检测
if((K5==1)&&(K7==1))
{
    ff=1; //1 代表击球成功 弧线
    if((xian==6) || (xian==10) || (xian==11)) //位置 1
    {
        huxian_14();
    }
}
}
```

```

    {
        ff=2;                                //2 代表死亡
        tabletennis_gameover();
        if((xian==6) || (xian==10) || (xian==11)) //位置 1
        {
            zhixian_12();
        }
        else if((xian==7) || (xian==8) || (xian==9))      //位置 3
        {
            zhixian_34();
        }
    }

    else if((K6==1)&&(K7==1))
    {
        ff=1;                                //1 代表击球成功      弧线
        if((xian==7) || (xian==8) || (xian==9))      //位置 3
        {
            huxian_32();
        }
        else
        {
            ff=2;                                //2 代表死亡
            tabletennis_gameover();
            if((xian==6) || (xian==10) || (xian==11)) //位置 1
            {
                zhixian_12();
            }
            else if((xian==7) || (xian==8) || (xian==9))      //位置 3
            {
                zhixian_34();
            }
        }
    }

    else if((K5==1)&&(K8==1))
    {
        ff=1;                                //1 代表击球成功      弧线
        if((xian==6) || (xian==10) || (xian==11)) //位置 1
        {
            xiexian_14();
        }
        else
        {
            ff=2;                                //2 代表死亡
            tabletennis_gameover();
        }
    }
}

```

```

        if((xian==6) || (xian==10) || (xian==11)) //位置 1
        {
            zhixian_12();
        }

        else if((xian==7) || (xian==8) || (xian==9))           //位置 3
        {
            zhixian_34();
        }

    }

else if((K6==1)&&(K8==1))
{
    ff=1;                      //1 代表击球成功          弧线
    if((xian==7) || (xian==8) || (xian==9))           //位置 3
    {
        xiexian_32();
    }
    else
    {
        ff=2;                      //2 代表死亡
        tabletennis_gameover();
        if((xian==6) || (xian==10) || (xian==11)) //位置 1
        {
            zhixian_12();
        }

        else if((xian==7) || (xian==8) || (xian==9))           //位置 3
        {
            zhixian_34();
        }
    }
}

else if((K5==1)&&(K6==0)&&(K7==0)&&(K8==0))
{
    ff=1;                      //1 代表击球成功          弧线
    if((xian==6) || (xian==10) || (xian==11)) //位置 1
    {
        zhixian_12();
    }
    else
    {
        ff=2;                      //2 代表死亡
        tabletennis_gameover();
        if((xian==6) || (xian==10) || (xian==11)) //位置 1
        {
            zhixian_12();
        }

        else if((xian==7) || (xian==8) || (xian==9))           //位置 3
    }
}

```

```

        {
            zhixian_34();
        }
    }

else if((K5==0)&&(K6==1)&&(K7==0)&&(K8==0))
{
    ff=1;                                //1 代表击球成功      弧线
    if((xian==7) || (xian==8) || (xian==9)) //位置 3
    {
        zhixian_34();
    }
    else
    {
        ff=2;                                //2 代表死亡
        tabletennis_gameover();
        if((xian==6) || (xian==10) || (xian==11)) //位置 1
        {
            zhixian_12();
        }
        else if((xian==7) || (xian==8) || (xian==9)) //位置
        {
            zhixian_34();
        }
    }
}
}

*****双打 *** 总控制*****
void shuangda_ctrl()
{
    grade[0]=0;                            //分数清零
    grade[1]=0;                            //分数清零
    grade[2]=0;                            //分数清零
    grade[3]=0;                            //分数清零
    configuration1();                     //整体布局
    paizi(bat10[0], bat10[1], 1);        // 画乒乓球球拍
    paizi(bat20[0], bat20[1], 1);        // 画乒乓球球拍
    paizi(bat30[0], bat30[1], 1);        // 画乒乓球球拍
    paizi(bat40[0], bat40[1], 1);
}

```

```
    qiu(ball0[0], ball0[1], 1); //显示球的位置
    while(1)
    {
        if((K5==1)&&(K6==1)&&(K7==1)&&(K8==1)) //发球
        {
            music_faqiu(); //发球的声音
            zhixian_12();
            Delay(100);
        }
    }
    /*****从左到右画直线函数*****
void zhixian_34()
{
    xian=3;
    for(sit=0;sit<NN;sit++)
    { if(players==0) //检测是否为单打，只有单打才会有拍子的移动
        {
            //判断游戏是否结束
            if(sit<(NN-3))
            {
                danda2_1();
            }
            else if((NN-1)>=sit)&&(sit>=NN-3))
            {
                danda2_2();
            }
        }
    else if(sit==NN)
        {
            danda2_3();
            paizi_move();
        }
        else //双打
    {
        //判断游戏是否结束
        if(sit<(NN-3))
        {
            shuangda2_1();
        }
        else if(sit>=NN-3)
        {
            shuangda2_2();
        }
    }
}
```

```

        }

        clear_ball(ball[0], ball[1]);
        road[sit]=zhi[sit];
        if(road[sit]==0)
        {
            ball[0]=ball[0]+1;                                //直行
        }
        else if(road[sit]==1)
        {
            ball[0]=ball[0]+1;                                //斜下
            ball[1]=ball[1]+1;
        }
        else
        {
            ball[0]=ball[0]+1;                                //斜上
            ball[1]=ball[1]-1;
        }
        qiu(ball[0], ball[1], 1);
        Delay(speed[dly]);
    }
}

void zhixian_12()
{
    xian=0;
    for(sit=0;sit<NN;sit++)
    {
        if(players==0)          //检测是否为单打，只有单打才会有拍子的移动
        {
            //判断游戏是否结束
            if(sit<(NN-3))
            {
                danda2_1();
            }
            else if((NN-1)>=sit)&&(sit>=NN-3))
            {
                danda2_2();
            }
            else if(sit==NN)
            {
                danda2_3();
            }
            paizi_move();
        }
    }
}

```

```
    }
    else//双打
{
    //判断游戏是否结束
    if(sit<(NN-3))
    {
        shuangda2_1();
    }
    else if(sit>=NN-3)
    {
        shuangda2_2();
    }
}
clear_ball(ball[0],ball[1]);
road[sit]=zhi[sit];
if(road[sit]==0)
{
    ball[0]=ball[0]+1; //直行
}
else if(road[sit]==1)
{
    ball[0]=ball[0]+1; //斜下
    ball[1]=ball[1]+1;
}
else
{
    ball[0]=ball[0]+1; //斜上
    ball[1]=ball[1]-1;
}
qiu(ball[0],ball[1],1);
Delay(speed[dly]);
}

/*
从左到右弧线 14 函数
*/
void huxian_14()
{
    xian=2;
```

```
for(sit=0;sit<NN;sit++)
{
    if(players==0) //检测是否为单打，只有单打才会有拍子的移动
    {
        //判断游戏是否结束
        if(sit<(NN-3))
        {
            danda2_1();
        }
        else if((NN-1)>=sit)&&(sit>=NN-3))
        {
            danda2_2();
        }
        else if(sit==NN)
        {
            danda2_3();
        }
        paizi_move();
    }

    else //双打
    {
        //判断游戏是否结束
        if(sit<(NN-3))
        {
            shuangda2_1();
        }
        else if(sit>=NN-3)
        {
            shuangda2_2();
        }
    }

    clear_ball(ball[0],ball[1]);
    road[sit]=huxian14[sit];
    if((ball[1]>=57)|| (ball[1]<=20))
    {
        ball[0]=ball[0]+1;
    }
    else if(road[sit]==0)
    {
        ball[0]=ball[0]+1; //直行
    }
    else if(road[sit]==1)
    {
        ball[0]=ball[0]+1; //斜下
        ball[1]=ball[1]+1;
    }
}
```

```
    else
    {
        ball[0]=ball[0]+1;                      //斜上
        ball[1]=ball[1]-1;
    }
    qiu(ball[0],ball[1],1);
    Delay(speed[dly]);
}
}

/*****
```

从左到右弧线 32 函数

```
*****
void huxian_32()
{
    xian=5;
    for(sit=0;sit<NN;sit++)
    {
        if(players==0)//检测是否为单打，只有单打才会有拍子的移动
        {
            //判断游戏是否结束
            if(sit<(NN-3))
            {
                danda2_1();
            }
            else if((NN-1)>=sit)&&(sit>=NN-3))
            {
                danda2_2();
            }
            else if(sit==NN)
            {
                danda2_3();
            }
            paizi_move();
        }
        else          //双打
        {
            //判断游戏是否结束
            if(sit<(NN-3))
            {
                shuangda2_1();
            }
            else if(sit>=NN-3)
            {
                shuangda2_2();
            }
        }
    }
}
```

```
    }
    clear_ball(ball[0], ball[1]);
    road[sit]=huxian32[sit];
    if((ball[1]>=57) || (ball[1]<=20))
    {
        ball[0]=ball[0]+1;
    }
    else if(road[sit]==0)
    {
        ball[0]=ball[0]+1; //直行
    }
    else if(road[sit]==1)
    {
        ball[0]=ball[0]+1; //斜下
        ball[1]=ball[1]+1;
    }
    else
    {
        ball[0]=ball[0]+1; //斜上
        ball[1]=ball[1]-1;
    }
    qiu(ball[0], ball[1], 1);
    Delay(speed[dly]);
}

}
/****************
从右到左画直线函数
****************/
void zhixian_21()
{
    xian=6;
    for(sit=NN;sit>0;sit--)
    {
        if(players==0) //检测是否为单打，只有单打才会有拍子的移动
        {
            //判断游戏是否结束
            if(sit>3)
            {
                danda1_1();
            }
            else if((3>=sit)&&(sit>=1))
            {
                danda1_2();
            }
        }
    }
}
```

```
        }
    else if(sit==0)
    {
        danda1_3();
    }
    paizi_move();
}

else //双打
{
    //判断游戏是否结束
    if(sit>3)
    {
        shuangda1_1();
    }
    else if(sit<=3)
    {
        shuangda1_2();
    }
}

clear_ball(ball[0], ball[1]);
road[sit]=zhi[sit];
if(road[sit]==0)
{
    ball[0]=ball[0]-1; //直行
}
else if(road[sit]==1)
{
    ball[0]=ball[0]-1; //斜上
    ball[1]=ball[1]-1;
}
else
{
    ball[0]=ball[0]-1; //斜下
    ball[1]=ball[1]+1;
}
qiu(ball[0], ball[1], 1);
Delay(speed[dly]);
}

}
/***********************
从右到左画直线函数
```

```
*****  
void zhixian_43()  
{      xian=9;  
    for(sit>NN;sit>0;sit--)  
    { if(players==0) //检测是否为单打，只有单打才会有拍子的移动  
        {      //判断游戏是否结束  
            if(sit>3)  
            {      danda1_1();  
            }  
            else if((3>=sit)&&(sit>=1))  
            {      danda1_2();  
            }  
            else if(sit==0)  
            {      danda1_3();  
            }  
            paizi_move();  
        }  
        Else //双打  
        {      //判断游戏是否结束  
            if(sit>3)  
            {      shuangda1_1();  
            }  
            else if(sit<=3)  
            {      shuangda1_2();  
            }  
        }  
        clear_ball(ball[0],ball[1]);  
        road[sit]=zhi[sit];  
        if(road[sit]==0)  
        {      ball[0]=ball[0]-1; //直行  
        }  
        else if(road[sit]==1)  
        {      ball[0]=ball[0]-1; //斜上  
              ball[1]=ball[1]-1;  
        }  
    }  
}
```

```
    }
else
{
    ball[0]=ball[0]-1;                      //斜下
    ball[1]=ball[1]+1;
}
qiu(ball[0],ball[1],1);
Delay(speed[dly]);
}

}

/*****
```

从右到左弧线 23 函数

```
*****
void huxian_23()
{
    xian=8;
    for(sit>NN;sit>0;sit--)
    { if(players==0) //检测是否为单打，只有单打才会有拍子的移动
        {
            //判断游戏是否结束
            if(sit>3)
            {
                danda1_1();
            }
            else if((3>=sit)&&(sit>=1))
            {
                danda1_2();
            }
            else if(sit==0)
            {
                danda1_3();
            }
            paizi_move();
        }
        else //双打
        {
            //判断游戏是否结束
            if(sit>3)
            {
                shuangda1_1();
            }
            else if(sit<=3)
            {
                shuangda1_2();
            }
        }
        clear_ball(ball[0],ball[1]);
        road[sit]=huxian23[sit];
    }
}
```

```
if((ball[1]>=57) || (ball[1]<=20))
{
    ball[0]=ball[0]-1;
}
else if(road[sit]==0)
{
    ball[0]=ball[0]-1; //直行
}
else if(road[sit]==1)
{
    ball[0]=ball[0]-1; //斜上
    ball[1]=ball[1]-1;
}
else
{
    ball[0]=ball[0]-1; //斜下
    ball[1]=ball[1]+1;
}
qiu(ball[0], ball[1], 1);
Delay(speed[dly]);
}

/***********************/

从右到左弧线 41 函数

*****
```

```
void huxian_41()
{
    xian=11;
    for(sit>NN;sit>0;sit--)
    {
        if(players==0) //检测是否为单打，只有单打才会有拍子的移动
        {
            //判断游戏是否结束
            if(sit>3)
            {
                danda1_1();
            }
            else if((3>=sit)&&(sit>=1))
            {
                danda1_2();
            }
            else if(sit==0)
            {
                danda1_3();
                paizi_move();
            }
            else //双打
            {
                //判断游戏是否结束
                if(sit>3)
                {
                    shuangda1_1();
                }
            }
        }
    }
}
```

```

        else if(sit<=3)
        {
            shuangda1_2();
        }

        clear_ball(ball[0], ball[1]);
        road[sit]=huxian41[sit];
        if((ball[1]>=57) || (ball[1]<=20))
        {
            ball[0]=ball[0]-1;
        }
        else if(road[sit]==0)
        {
            ball[0]=ball[0]-1; //直行
        }

        else if(road[sit]==1)
        {
            ball[0]=ball[0]-1; //斜上
            ball[1]=ball[1]-1;
        }

        else
        {
            ball[0]=ball[0]-1; //斜下
            ball[1]=ball[1]+1;
        }

        qiu(ball[0], ball[1], 1);
        Delay(speed[d1y]);
    }

} */

从左到右斜线 14 函数
*****
```

```

void xiexian_14()
{
    xian=1;
    for(sit=0;sit<NN;sit++)
    { if(players==0) //检测是否为单打，只有单打才会有拍子的移动
        {
            //判断游戏是否结束
            if(sit<(NN-3))
            {
                danda2_1(); }
            else if((NN-1)>=sit)&&(sit>=NN-3))
            {
                danda2_2(); }
            else if(sit==NN)
            {
                danda2_3(); }
                paizi_move(); }
    }
}
```

```

        else                                //双打
{
    //判断游戏是否结束
    if(sit<(NN-3))
    {
        shuangda2_1();
    }
    else if(sit>=NN-3)
    {
        shuangda2_2();                }
    clear_ball(ball[0],ball[1]);
    road[sit]=xiexian14[sit];
    if((ball[1]>=57) || (ball[1]<=20))
    {
        ball[0]=ball[0]+1;           }
    else if(road[sit]==0)
    {
        ball[0]=ball[0]+1;           //直行
    }
    else if(road[sit]==1)
    {
        ball[0]=ball[0]+1;           //斜下
        ball[1]=ball[1]+1;           }
    else
    {
        ball[0]=ball[0]+1;           //斜上
        ball[1]=ball[1]-1;           }
    qiu(ball[0],ball[1],1);
    Delay(speed[dly]);          }
}
*****
```

从左到右斜线 32 函数

```
*****
void xiexian_32()
{
    xian=4;
    for(sit=0;sit<NN;sit++)
    { if(players==0)    //检测是否为单打，只有单打才会有拍子的移动
        {
            //判断游戏是否结束
            if(sit<(NN-3))
            {
                danda2_1();                  }
            else if((NN-1)>=sit)&&(sit>=NN-3))
            {
                danda2_2();                  }
        }
    }
}
```

```

        else if(sit==NN)
        {
            danda2_3();
        }
        paizi_move();                }

        else //双打
    {
        //判断游戏是否结束
        if(sit<(NN-3))
        {
            shuangda2_1();           }
        else if(sit>=NN-3)
        {
            shuangda2_2();           }
    }

    clear_ball(ball[0],ball[1]);
    road[sit]=xiexian32[sit];
    if((ball[1]>=57) || (ball[1]<=20))
    {
        ball[0]=ball[0]+1;          }
    else if(road[sit]==0)
    {
        ball[0]=ball[0]+1;          //直行
    }

    else if(road[sit]==1)
    {
        ball[0]=ball[0]+1;          //斜下
        ball[1]=ball[1]+1;           }
    else
    {
        ball[0]=ball[0]+1;          //斜上
        ball[1]=ball[1]-1;           }
    qiu(ball[0],ball[1],1);
    Delay(speed[dly]);           }

/*****

```

从右到左画斜线 23 函数

```
*****
void xiexian_23()
{
    xian=7;
    for(sit=NN;sit>0;sit--)
    { if(players==0) //检测是否为单打，只有单打才会有拍子的移动
        { //判断游戏是否结束

```

```

        if(sit>3)
        {
            danda1_1();
        }
        else if((3>=sit)&&(sit>=1))
        {
            danda1_2();
        }
        else if(sit==0)
        {
            danda1_3();
            paizi_move();
        }
        else //双打
        {
            //判断游戏是否结束
            if(sit>3)
            {
                shuangda1_1();
            }
            else if(sit<=3)
            {
                shuangda1_2();
            }
            clear_ball(ball[0], ball[1]);
            road[sit]=xiexian32[sit];
            if((ball[1]>=57) || (ball[1]<=20))
            {
                ball[0]=ball[0]-1;
            }
            else if(road[sit]==0)
            {
                ball[0]=ball[0]-1; //直行
            }
            else if(road[sit]==1)
            {
                ball[0]=ball[0]-1; //斜上
                ball[1]=ball[1]-1;
            }
            else
            {
                ball[0]=ball[0]-1; //斜下
                ball[1]=ball[1]+1;
            }
            qiu(ball[0], ball[1], 1);
            Delay(speed[dly]);
        }
    ****

```

从右到左画斜线 41 函数

```
*****  
void xiexian_41()  
{      xian=10;  
    for(sit>NN;sit>0;sit--)  
    {if(players==0) //检测是否为单打，只有单打才会有拍子的移动  
     {      //判断游戏是否结束  
       if(sit>3)  
       {      danda1_1();  
       }  
       else if((3>=sit)&&(sit>=1))  
       {      danda1_2();  
       }  
       else if(sit==0)  
       {      danda1_3();  
       }  
       paizi_move();  
     }  
     else //双打  
     { //判断游戏是否结束  
       if(sit>3)  
       {      shuangda1_1();  
       }  
       else if(sit<=3)  
       {shuangda1_2();  
     }  
   }  
   clear_ball(ball[0],ball[1]);  
   road[sit]=xiexian14[sit];  
   if((ball[1]>=57) || (ball[1]<=20))  
   {      ball[0]=ball[0]-1;  
   }  
   else if(road[sit]==0)  
   {      ball[0]=ball[0]-1;          //直行  
   }  
   else if(road[sit]==1)  
   {      ball[0]=ball[0]-1;          //斜上
```

```
ball[1]=ball[1]-1;  
}  
else  
{      ball[0]=ball[0]-1;          //斜下  
      ball[1]=ball[1]+1;  
}  
qiu(ball[0], ball[1], 1);  
Delay(speed[dly]);    }  
}  
*****  
左边玩家的墙 显示/消失 墙在拍子的两个像素宽度范围内显示，使  
用后需要重新生成拍子  
*****  
void wall_1(uint8 z)  
{      Lcd_VertLine(3, 15, 48, z);  
      Lcd_VertLine(4, 15, 48, z); }  
*****  
右边玩家的墙 显示/消失 墙在拍子的两个像素宽度范围内显示，使  
用后需要重新生成拍子  
*****  
void wall_2(uint8 z)  
{      Lcd_VertLine(123, 15, 48, z);  
      Lcd_VertLine(124, 15, 48, z); }  
*****  
播放开机音乐的函数  
*****  
uint8 start_music()  
{      BeepIO=0;  
      InitialSound();  
      Play(kj, 0, 3, 360);  
      if((K7==1)&&(K8==1))  
      {          BeepIO = 0;  
          Reset12864();          //液晶 12864 初始化  
          Delay(20);            //延时，保证显示更稳定  
          ScreenClear12864();   //清屏
```

```
Lcd_Clear(0);
Delay(20);
fun_select();
return 0;
}

BeepIO=0; }

/***********************/

调用屏幕保护程序的函数

*****void screen_protect()
{
    uint8 X0, Y0, X1, Y1, i;
    { X0=32-1;
    Y0=32-1;
    X1=128-32;
    Y1=64-32;
    for(i=0;i<32;i++)
    { //有按键按下则返
    if((K1==1) || (K2==1) || (K3==1) || (K4==1) || (K5==1) || (K6==1) || (K7==1) || (K8
    ==1))
        {
            fun_select();
            Lcd_Rectangle(X0, Y0, X1, Y1, 1);

            X0--;
            Y0--;
            X1++;
            Y1++;

            Delay(50);
        }
    for(i=0;i<32;i++)
    { //有按键按下则返回
    if((K1==1) || (K2==1) || (K3==1) || (K4==1) || (K5==1) || (K6==1) || (K7==1) || (K8
    ==1))
        {
            fun_select();
            Lcd_Rectangle(X0, Y0, X1, Y1, 0);

            X0++;
        }
    }
}
```

```
        Y0++;
        X1--;
        Y1--;
        Delay(50);
    }

    Lcd_Clear(0);
        Lcd_WriteStr(0, 0, "          ");
        Lcd_WriteStr(1, 0, "          ");
        Lcd_WriteStr(2, 0, "          ");
        Lcd_WriteStr(3, 0, "          ");
    Delay(50);    }}

/*****************/
温度          18B20      开始
/*****************/
/* 名称 : delay1()
 * 功能 : 延时函数
 * 输入 : 无
 * 输出 : 无
/*****************/
*/
void delay1(uint nnn)
{
    int i;
    for(i=0; i<nnn; i++)
}

/*****************/
* 名称 : Reset()
* 功能 : 复位 DS18B20
* 输入 : 无
* 输出 : 无
/*****************/
*/
uchar Reset(void)
{
    uchar receive_ready;
    DQ = 0;
```

```
delay1(29);  
DQ = 1;  
delay1(3);  
deceive_ready = DQ;  
delay1(25);  
return(deceive_ready);}  
/*************************************************************  
* 名称 : read_bit()  
* 功能 : 从 DS18B20 读一个位值  
* 输入 : 无  
* 输出 : 从 DS18B20 读出的一个位值  
*****  
**/  
uchar read_bit(void)  
{      uchar i;  
    DQ = 0;  
    DQ = 1;  
    for(i=0; i<3; i++);  
    return(DQ);}  
/*************************************************************  
* 名称 : write_bit()  
* 功能 : 向 DS18B20 写一位  
* 输入 : bitval (要对 DS18B20 写入的位值)  
* 输出 : 无  
*****  
**/  
void write_bit(uchar bitval)  
{  
DQ=0;if(bitval==1)  
DQ=1;  
delay1(5);  
DQ=1;}  
/*************************************************************  
* 名称 : read_byte()
```

```
* 功能：从 DS18B20 读一个字节
* 输入：无
* 输出：从 DS18B20 读到的值
*****
*/
uchar read_byte(void)
{
    uchar i, m, receive_data;
    m = 1;
    receive_data = 0;
    for(i=0; i<8; i++)
    {
        if(read_bit())
        {
            receive_data = receive_data + (m << i);
        }
        delay1(6);
    }
    return(receive_data);
}
*****
* 名称：write_byte()
* 功能：向 DS18B20 写一个字节
* 输入：val (要对 DS18B20 写入的命令值)
* 输出：无
*****
*/
void write_byte(uchar val)
{
    uchar i, temp;
    for(i=0; i<8; i++)
    {
        temp = val >> i;
        temp = temp & 0x01;
        write_bit(temp);
        delay1(5);
    }
}
*****
得到和显示温度的函数
*****
*/
void disp_num(uint8 x, uint8 y, uint8 num)
{
    if(num==0)
```

```
Lcd_WriteStr(x, y, "0");
else if(num==1)
Lcd_WriteStr(x, y, "1");
else if(num==2)
Lcd_WriteStr(x, y, "2");
else if(num==3)
Lcd_WriteStr(x, y, "3");
else if(num==4)
Lcd_WriteStr(x, y, "4");
else if(num==5)
Lcd_WriteStr(x, y, "5");
else if(num==6)
Lcd_WriteStr(x, y, "6");
else if(num==7)
Lcd_WriteStr(x, y, "7");
else if(num==8)
Lcd_WriteStr(x, y, "8");
else if(num==9)
Lcd_WriteStr(x, y, "9");
}

/*****************/
清楚温度条
*****************/
*/
void clear_wendu()
{
    Lcd_HoriLine(21, 40, 87, 0);
    Lcd_HoriLine(21, 41, 87, 0);
}

/*****************/
温度计变化的函数
*****************/
*/
void disp_wendu(uint8 shi, uint8 ge)
{
    if(shi==0&&ge==1)
    {
        Lcd_HoriLine(21, 40, 2, 1);
        Lcd_HoriLine(21, 41, 2, 1);
    }
}
```

```
else if(shi==0&&ge==2)
{
    Lcd_HoriLine(21, 40, 4, 1);
    Lcd_HoriLine(21, 41, 4, 1);}
else if(shi==0&&ge==3)
{
    Lcd_HoriLine(21, 40, 6, 1);
    Lcd_HoriLine(21, 41, 6, 1);}
else if(shi==0&&ge==4)
{
    Lcd_HoriLine(21, 40, 8, 1);
    Lcd_HoriLine(21, 41, 8, 1);}
else if(shi==0&&ge==5)
{
    Lcd_HoriLine(21, 40, 10, 1);
    Lcd_HoriLine(21, 41, 10, 1);}
else if(shi==0&&ge==6)
{
    Lcd_HoriLine(21, 40, 12, 1);
    Lcd_HoriLine(21, 41, 12, 1);}
else if(shi==0&&ge==7)
{
    Lcd_HoriLine(21, 40, 14, 1);
    Lcd_HoriLine(21, 41, 14, 1);}
else if(shi==0&&ge==8)
{
    Lcd_HoriLine(21, 40, 16, 1);
    Lcd_HoriLine(21, 41, 16, 1);}
else if(shi==0&&ge==9)
{
    Lcd_HoriLine(21, 40, 18, 1);
    Lcd_HoriLine(21, 41, 18, 1);}
else if(shi==1&&ge==0)
{
    Lcd_HoriLine(21, 40, 20, 1);
    Lcd_HoriLine(21, 41, 20, 1);}
else if(shi==1&&ge==1)
{
    Lcd_HoriLine(21, 40, 22, 1);
    Lcd_HoriLine(21, 41, 22, 1);}
else if(shi==1&&ge==2)
{
    Lcd_HoriLine(21, 40, 24, 1);
    Lcd_HoriLine(21, 41, 24, 1);}
else if(shi==1&&ge==3)
```

```
{      Lcd_HoriLine(21, 40, 26, 1);  
      Lcd_HoriLine(21, 41, 26, 1);}  
else if(shi==1&&ge==4)  
{      Lcd_HoriLine(21, 40, 28, 1);  
      Lcd_HoriLine(21, 41, 28, 1);}  
else if(shi==1&&ge==5)  
{      Lcd_HoriLine(21, 40, 30, 1);  
      Lcd_HoriLine(21, 41, 30, 1);}  
else if(shi==1&&ge==6)  
{      Lcd_HoriLine(21, 40, 32, 1);  
      Lcd_HoriLine(21, 41, 32, 1);}  
else if(shi==1&&ge==7)  
{      Lcd_HoriLine(21, 40, 34, 1);  
      Lcd_HoriLine(21, 41, 34, 1);}  
else if(shi==1&&ge==8)  
{      Lcd_HoriLine(21, 40, 36, 1);  
      Lcd_HoriLine(21, 41, 36, 1);}  
else if(shi==1&&ge==9)  
{      Lcd_HoriLine(21, 40, 38, 1);  
      Lcd_HoriLine(21, 41, 38, 1);}  
else if(shi==2&&ge==0)  
{      Lcd_HoriLine(21, 40, 40, 1);  
      Lcd_HoriLine(21, 41, 40, 1);}  
else if(shi==2&&ge==1)  
{      Lcd_HoriLine(21, 40, 42, 1);  
      Lcd_HoriLine(21, 41, 42, 1);}  
else if(shi==2&&ge==2)  
{      Lcd_HoriLine(21, 40, 44, 1);  
      Lcd_HoriLine(21, 41, 44, 1);}  
else if(shi==2&&ge==3)  
{      Lcd_HoriLine(21, 40, 46, 1);  
      Lcd_HoriLine(21, 41, 46, 1);}  
else if(shi==2&&ge==4)  
{      Lcd_HoriLine(21, 40, 48, 1);
```

```
        Lcd_HoriLine(21, 41, 48, 1);}

else if(shi==2&&ge==5)
{
    Lcd_HoriLine(21, 40, 50, 1);
    Lcd_HoriLine(21, 41, 50, 1);}

else if(shi==2&&ge==6)
{
    Lcd_HoriLine(21, 40, 52, 1);
    Lcd_HoriLine(21, 41, 52, 1);}

else if(shi==2&&ge==7)
{
    Lcd_HoriLine(21, 40, 54, 1);
    Lcd_HoriLine(21, 41, 54, 1);}

else if(shi==2&&ge==8)
{
    Lcd_HoriLine(21, 40, 56, 1);
    Lcd_HoriLine(21, 41, 56, 1);}

else if(shi==2&&ge==9)
{
    Lcd_HoriLine(21, 40, 58, 1);
    Lcd_HoriLine(21, 41, 58, 1);}

else if(shi==3&&ge==0)
{
    Lcd_HoriLine(21, 40, 60, 1);
    Lcd_HoriLine(21, 41, 60, 1);}

else if(shi==3&&ge==1)
{
    Lcd_HoriLine(21, 40, 62, 1);
    Lcd_HoriLine(21, 41, 62, 1);}

else if(shi==3&&ge==2)
{
    Lcd_HoriLine(21, 40, 64, 1);
    Lcd_HoriLine(21, 41, 64, 1);}

else if(shi==3&&ge==3)
{
    Lcd_HoriLine(21, 40, 66, 1);
    Lcd_HoriLine(21, 41, 66, 1);}

else if(shi==3&&ge==4)
{
    Lcd_HoriLine(21, 40, 68, 1);
    Lcd_HoriLine(21, 41, 68, 1);}

else if(shi==3&&ge==5)
{
    Lcd_HoriLine(21, 40, 70, 1);
    Lcd_HoriLine(21, 41, 70, 1);}
```

```
else if(shi==3&&ge==6)
{
    Lcd_HoriLine(21, 40, 72, 1);
    Lcd_HoriLine(21, 41, 72, 1);
}
else if(shi==3&&ge==7)
{
    Lcd_HoriLine(21, 40, 74, 1);
    Lcd_HoriLine(21, 41, 74, 1);
}
else if(shi==3&&ge==8)
{
    Lcd_HoriLine(21, 40, 76, 1);
    Lcd_HoriLine(21, 41, 76, 1);
}
else if(shi==3&&ge==9)
{
    Lcd_HoriLine(21, 40, 78, 1);
    Lcd_HoriLine(21, 41, 78, 1);
}
else if(shi==4&&ge==0)
{
    Lcd_HoriLine(21, 40, 80, 1);
    Lcd_HoriLine(21, 41, 80, 1);
}
else if(shi==4&&ge==1)
{
    Lcd_HoriLine(21, 40, 82, 1);
    Lcd_HoriLine(21, 41, 82, 1);
}
else if(shi==4&&ge==2)
{
    Lcd_HoriLine(21, 40, 84, 1);
    Lcd_HoriLine(21, 41, 84, 1);
}
else if(shi==4&&ge==3)
{
    Lcd_HoriLine(21, 40, 86, 1);
    Lcd_HoriLine(21, 41, 86, 1);
}
else //44 度及以上
{
    Lcd_HoriLine(21, 40, 88, 1);
    Lcd_HoriLine(21, 41, 88, 1);
}

} ****
得到和显示温度的函数
***** */
uint8 temperature()
{
    while(1) {
        uint8 temp;
```

```
uint8 wendu[2]={0} ;  
uint8 wendu_shi, wendu_ge;  
uint8 shi1, ge1;  
Reset();  
shi1=0;           //判断温度有没有变化  
ge1=0;  
//    Lcd_WriteStr(0, 0, "温度: ");  
write_byte(jump_ROM);  
write_byte(start);  
Reset();  
write_byte(jump_ROM);  
write_byte(read_EEROM);  
TMPL = read_byte();  
TMPH = read_byte();  
temp = TMPL / 16 + TMPH * 16;  
wendu_shi=temp%10;  
wendu_ge=temp/10%10;  
if((shi1!=wendu_shi)&&(ge1!=wendu_ge))  
{      disp_num(4, 0, wendu_ge);  
      disp_num(5, 0, wendu_shi);  
      clear_wendu();  
      disp_wendu(wendu_ge, wendu_shi);  
      Delay(500);  
      shi1=wendu_shi;  
      ge1=wendu_ge;    }  
if((K7==1)&&(K8==1))  
{      Lcd_Clear(0);  
      Lcd_WriteStr(0, 0, "");  
      Lcd_WriteStr(1, 0, "");  
      Lcd_WriteStr(2, 0, "");  
      Lcd_WriteStr(3, 0, "");  
      Delay(20);  
      PictureDisplay12864(pic_jiemian1);  
      return 0; } }
```

```
/****************************************
温度          18B20      结束
****************************************/
/****************************************
俄罗斯方块 函数  开始
****************************************/
/****************************************
伪随机数发生器
****************************************/
double Random(void)
{
    long TmpSeed;
    TmpSeed=A*(Seed%Q)-R*(Seed/Q);
    if(TmpSeed>=0)
        Seed=TmpSeed;
    else
        Seed=TmpSeed+M;
    return (double)Seed/M; }

/****************************************
为伪随机数发生器播种
****************************************/
void InitRandom(unsigned long InitVal)
{
    Seed=InitVal; }

//延时子程序
void Delay11(unsigned int t)
{
    unsigned int i, j;
    for(i=0;i<t;i++)
        for(j=0;j<10;j++) }

/****************************************
LCD 初始化
****************************************/
void Lcd_Reset()
{
    LCD12864_PSB=1;
    CommandWrite12864(0x30);      //选择基本指令集
    CommandWrite12864(0x0c);      //开显示(无游标、不反白)
```

```
CommandWrite12864(0x01);      //清除显示，并且设定地址指针为00H
CommandWrite12864(0x06);      //指定在资料的读取及写入时，设定游标的移动方向及指定显示的移位    }
/*****************************************
初始化 MPU
****************************************/
void InitCpu(void)
{
    TMOD=0x0;
    TH0=0;
    TL0=0;
    TR0=1;
    ET0=1;
    EX1=1;
    EA=1;
    TCON|=0x04; }

/*****************************************
welcome 游戏选择界面
****************************************/
void welcome()
{
    Lcd_WriteStr(0, 0, "欢迎来玩");
    Lcd_WriteStr(0, 1, "俄罗斯方块 ");
    Lcd_WriteStr(0, 2, "设置按 K1");
    Lcd_WriteStr(0, 2, "开玩按 K2"); }

/*****************************************俄罗斯方块部分
*****************************************
画墙壁， 初始化界面
****************************************/
void DrawBoard(void)
{
    unsigned char n;
    for(n=0;n<12;n++)
    {
        Lcd_Rectangle(3*n, 0, 3*n+2, 2, 1);
        Lcd_Rectangle(3*n, 60, 3*n+2, 62, 1);    }
    for(n=0;n<20;n++)
    {
        Lcd_Rectangle(0, 3*n, 2, 3*n+2, 1);
```

```
Lcd_Rectangle(33, 3*n, 35, 3*n+2, 1); }  
Lcd_WriteStr(4, 0, "经典游戏");  
Lcd_WriteStr(3, 2, "Score:");  
Lcd_WriteStr(3, 3, "Level:"); }  
*****  
游戏结束处理  
*****  
void GameOver(void)  
{ if((SysFlag&(1<<DEADFLAG))!=0)  
    Lcd_WriteStr(3, 1, "You Fail");  
else  
    Lcd_WriteStr(3, 1, "You Pass");}  
unsigned int code MaskTab[16]={  
0x0001, 0x0002, 0x0004, 0x0008, 0x0010, 0x0020, 0x0040, 0x0080,  
0x0100, 0x0200, 0x0400, 0x0800, 0x1000, 0x2000, 0x4000, 0x8000};  
*****  
根据积木图标左下坐标 X, Y 来画出积木图标  
*****  
void DrawSign(struct Jimu Temp, unsigned char DrawMode)  
{ unsigned char m, n;  
for(m=0; m<4; m++)  
    for(n=0; n<4; n++)  
    { if((Temp.dat&MaskTab[4*m+n])!=0  
        Lcd.Rectangle(Temp.x+n*3, Temp.y-2-3*m, Temp.x+n*3+2, Temp.y-3*m, DrawMode); } }  
*****  
将积木图标值融入 num 数据中  
也即把积木图标固定，无法再下降  
*****  
FixSign(void)  
{ unsigned char m, n;  
for(m=0; m<4; m++) //行循环  
    for(n=0; n<4; n++) //列循环
```

```
{           if((Sign[0].dat&MaskTab[4*m+n])!=0)
{      num[20-(Sign[0].y-2)/3+m]|=MaskTab[11-Sign[0].x/3-n]; }
}

//*****判断积木图标中方块是否与障碍方块重合*****
//*****判断积木图标是否可以继续下降一格*****
//*****判断积木图标是否可以向左移动*****
//*****判断积木图标是否可以向右移动*****
```

unsigned char CheckIf(void)

```
{      unsigned char m,n;
    for(m=0;m<4;m++)//行循环
        for(n=0;n<4;n++)//列循环
    {           if((Sign[1].dat&MaskTab[4*m+n])!=0)
{if((num[20-(Sign[1].y-2)/3+m]&MaskTab[11-Sign[1].x/3-n])!=0)
            return 0;         }
    return 1; }
```

//\*\*\*\*\*判断积木图标是否可以向左移动\*\*\*\*\*

```
unsigned char CheckIfDown(void)
```

```
{      Sign[1]=Sign[0];//  
    Sign[1].y+=3;//假设下降一格  
    return CheckIf();}
```

//\*\*\*\*\*判断积木图标是否可以向右移动\*\*\*\*\*

```
unsigned char CheckIfLeft(void)
```

```
{      Sign[1]=Sign[0];
    Sign[1].x-=3;
    return CheckIf();}
```

//\*\*\*\*\*判断积木图标是否可以向右移动\*\*\*\*\*

```
unsigned char CheckIfRight(void)
```

```
{      Sign[1]=Sign[0];
    Sign[1].x+=3;
```

```
        return CheckIf();  
/*****  
判断是否可以旋转  
*****/  
unsigned char CheckIfRoll(void)  
{      unsigned char i;  
      unsigned int Temp;  
      Sign[1]=Sign[0];  
      if (++Sign[1].change>3)  
          Sign[1].change=0;  
      i=Sign[1].type*4+Sign[1].change;  
      Temp=(unsigned int)Block[i][0]<<8;  
      Temp=Temp|Block[i][1];  
      Sign[1].dat=Temp;  
      return CheckIf();  
/*****  
寻找满格的行并做消除处理  
最多寻找 4 个满行并做消除  
*****/  
void DelFull(void)  
{      unsigned char m, n;  
      unsigned char Temp;  
      unsigned char Flag=0;  
      Temp=(Sign[0].y-2)/3;  
      if (Temp>=20)//防止越过了下边界  
          Temp=1;  
      else  
          Temp=20-Temp;  
      for (n=Temp+3; n>=Temp; n--) //积木图标的最顶行开始寻找满行比较有利于运算  
      {  
          if (num[n]==0xffff)  
          {      Flag=1;  
              for (m=n+1; m<=19; m++)  
              {      num[m-1]=num[m]; }  
          }  
      }
```

```
num[m]=0x801;
Score++; //每找到一个满行，则分数加 1}
}

if(Flag) //为加速而设置并判断的标志，有已固定的积木有满格消行变化则重画
积木界面
{
    for(m=Temp;m<=19;m++) //为加速，不必要重第一行重画起，
只需要从积木图标最下行开始往上的重画
    {
        for(n=1;n<=10;n++)
        {
            if((num[m]&MaskTab[n])==0)
{if(Lcd_ReadPixel(30-(n-1)*3, 57-(m-1)*3)!=0) //为加速而做的读象素操作
{
Lcd_Rectangle(30-(n-1)*3, 57-(m-1)*3, 30-(n-1)*3+2, 57-(m-1)*3+2, 0);
}
else
{
if(Lcd_ReadPixel(30-(n-1)*3, 57-(m-1)*3)==0) //为加速而做的读象素操作
{
Lcd_Rectangle(30-(n-1)*3, 57-(m-1)*3, 30-(n-1)*3+2, 57-(m-1)*3+2
, 1); } } }

/*****随机产生一个积木图标放到预产生区域并显示出来*****
void CreatSign(void)
{
    unsigned char n;
    unsigned int Temp;

    DrawSign(Sign[2], 0); //先清除
    n=Random()*28;
    Temp=(unsigned int)Block[n][0]<<8;
    Temp=Temp|Block[n][1];
    Sign[2].dat=Temp;
    Sign[2].x=45;
    Sign[2].y=4*3+2;
    Sign[2].type=n/4;
```

```
Sign[2].change=n%4;
DrawSign(Sign[2], 1); //后画出}

void PrintScore(void)
{
    unsigned char Str[3];
    Str[0]=(Score/10) | 0x30;
    Str[1]=(Score%10) | 0x30;
    Str[2]=0;
    Lcd_WriteStr(6, 2, Str); }

void PrintLevel(void)
{
    unsigned char Str[3];
    Str[0]=(Level/10) | 0x30;
    Str[1]=(Level%10) | 0x30;
    Str[2]=0;
    Lcd_WriteStr(6, 3, Str); }

/******************

游戏的具体过程，也是俄罗斯方块算法的关键部分
****

uint8 GamePlay(void)
{
    unsigned char m, n;
    unsigned int Temp;
    SysFlag|=1<<NEWSIGNFLAG; //刚开始初始化为需要产生新的积木图标
    InitRandom(TL0);
    Lcd_WriteStr(3, 1, "Playing");
    PrintScore();
    PrintLevel();
    CreatSign();
    while(1)
    {
        BeepI0=0;
        if((K7==1)&&(K8==1))
        {
            return 0;
        }
        if((SysFlag&(1<<NEWSIGNFLAG))==1) //判是否需要产生新的积木图标
        {
            SysFlag&=~(1<<NEWSIGNFLAG);
            Sign[0]=Sign[2];
            CreatSign();
        }
    }
}
```

```
Sign[0].x=12;
Sign[0].y=14;
for(m=0;m<4;m++) //行循环
{
    for(n=0;n<4;n++) //列循环
{
if((Sign[0].dat&MaskTab[15-m*4-n])==0)
{
break;
}
if(n==4)
Sign[0].y=3;} //将积木图标出现置顶
for(m=0;m<4;m++) //行循环
for(n=0;n<4;n++) //列循环
{
if((Sign[0].dat&MaskTab[4*m+n])!=0)
{
if((num[20-(Sign[0].y-2)/3+m]&MaskTab[11-Sign[0].x/3-n])!=0)
{
}
}
if((SysFlag&(1<<DEADFLAG))!=0)
break;//如果产生新的积木图标中的方块与已固定好的方块重合，则死亡。游戏
结束
DrawSign(Sign[0], 1);
if((CheckIfLeft())&&(K3==1)) //左
{DrawSign(Sign[0], 0);
Sign[0].x-=3;
if((CheckIfRight())&&(K4==1))
{DrawSign(Sign[0], 0);
Sign[0].x+=3;
DrawSign(Sign[0], 1);
if((CheckIfDown())&&(K2==1)) //下
{
    DrawSign(Sign[0], 0);
    Sign[0].y+=3;
    DrawSign(Sign[0], 1);
}
}
if((CheckIfRoll())&&(K1==1)) //翻转
{
    Sign[0].change=0;
    m=Sign[0].type*4+Sign[0].change;
    Temp=(unsigned int)Block[m][0]<<8;
    Temp=Temp|Block[m][1];
}
```

```
Sign[0].dat=Temp;
DrawSign(Sign[0], 1);
if((SysFlag&(1<<PAUSEFLAG))!=0)
    continue;
Delay11(500);
if(++DelayCnt>=2*(11-Level))
{
    DelayCnt=0;
    if(CheckIfDown())//判断是否能继续下降一格
    {
        DrawSign(Sign[0], 0);
        Sign[0].y+=3;
        DrawSign(Sign[0], 1);
    }
    else
    {
        FixSign();
        DelFull();
        PrintScore();
        if(Score>=PASSTSCORE)
        {
            SysFlag&=^(1<<DEADFLAG);
            break;//跳出玩游戏过程
        }
        SysFlag|=1<<NEWSIGNFLAG;//新的积木图标产生标志置 1}
    }
}
uint8 russia()
{
    InitCpu();//初始化 CPU
    Lcd_Reset(); //初始化 LCD 屏
    Lcd_Clear(0);//清屏
    Lcd_Reset(); //初始化 LCD 屏
    Lcd_Clear(0);//清屏
    BeepI0=0;
    DrawBoard();//画界面
    BeepI0=0;
    GamePlay();//玩游戏
    if((K7==1)&&(K8==1))
    {
        Reset12864();           //液晶 12864 初始化
    }
}
```

```
Delay(20); //延时，保证显示更稳定
ScreenClear12864(); //清屏
Lcd_Clear(0);
Delay(20);
PictureDisplay12864(pic_jiemian1);
return 0; }

// GameOver() //游戏结束
// Lcd_Clear(0);
// PictureDisplay12864(pic_gameover);
// Delay(500);
Lcd_Reset() //初始化 LCD 屏
Lcd_Clear(0); //清屏 }

/***********************/
俄罗斯方块 函数 结束
*****void kajidonghua()
{
    uint8 ss; //相邻图片间延时时间
    ss=50;
    PictureDisplay12864(pic1);
    Delay(ss);
    PictureDisplay12864(pic2);
    Delay(ss);
    PictureDisplay12864(pic3);
    Delay(ss);
    PictureDisplay12864(pic4);
    Delay(ss);
    PictureDisplay12864(pic5);
    Delay(ss);
    PictureDisplay12864(pic6);
    Delay(ss);
    PictureDisplay12864(pic7);
    Delay(ss);
    PictureDisplay12864(pic8);
    Delay(ss);
}
```

```
PictureDisplay12864(pic9);  
Delay(ss);  
PictureDisplay12864(pic10);  
Delay(ss);  
PictureDisplay12864(pic11);  
Delay(ss);  
PictureDisplay12864(pic12);  
Delay(ss);  
PictureDisplay12864(pic13);  
Delay(ss);  
PictureDisplay12864(pic15);  
Delay(ss);  
PictureDisplay12864(pic16);  
Delay(ss);  
PictureDisplay12864(pic17);  
Delay(ss);  
PictureDisplay12864(pic18);  
Delay(ss);  
PictureDisplay12864(pic19);  
Delay(ss);  
PictureDisplay12864(pic20);  
Delay(ss);  
PictureDisplay12864(pic22);  
Delay(ss);  
PictureDisplay12864(pic23);  
Delay(ss);  
PictureDisplay12864(pic24);  
Delay(ss);  
PictureDisplay12864(pic25);  
Delay(ss);  
PictureDisplay12864(pic26);  
Delay(ss);  
PictureDisplay12864(pic28);  
Delay(ss);
```

```
PictureDisplay12864(pic29);
Delay(ss); }

/*****主函数*****/
void main()
{
    BeepIO=0;
    Reset12864();           //液晶 12864 初始化
    Delay(20);              //延时, 保证显示更稳定
    ScreenClear12864();     //清屏
    Lcd_Clear(0);
    BeepIO=0;
    kaijidonghua();         //开机动画
    PictureDisplay12864(label); //名字标签
    start_music();           //      开机音乐
    fun_select();}
```