目 录

1. 设计内容及要求

1. 1、设计内容：

以 AT89C51单片机为核心，制作一个 LCD显示的智能电子钟。

1.2、设计要求：

(1) 计时：秒、分、时、天、周、月、年。

(2) 闰年自动判别。

(3) 时间、月、日交替显示。

(5) 自定任意时刻自动开/关屏。

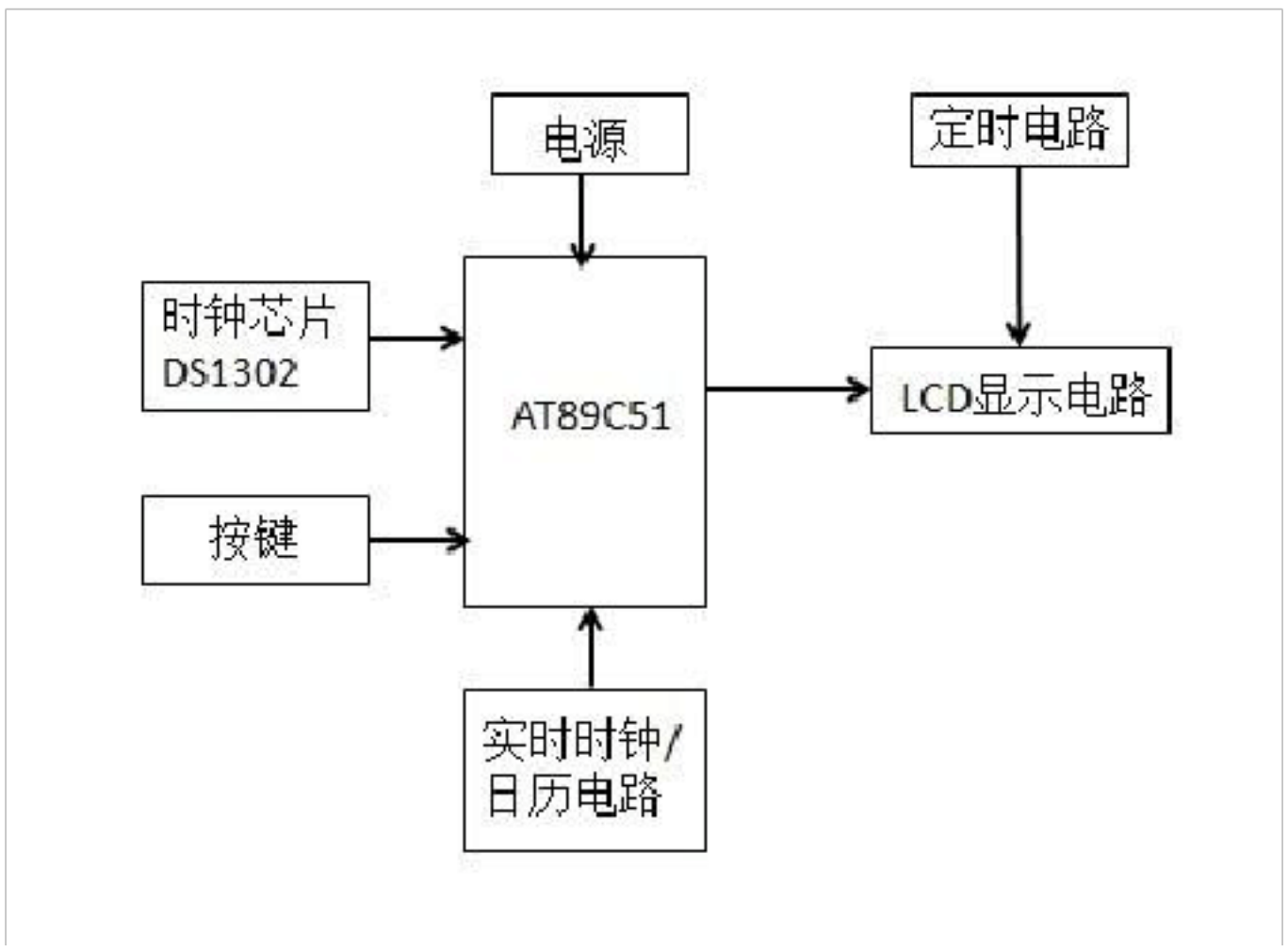
(6) 计时精度：误差≤1 秒/月（具有微调设置）。

1.3、 撰写设计报告

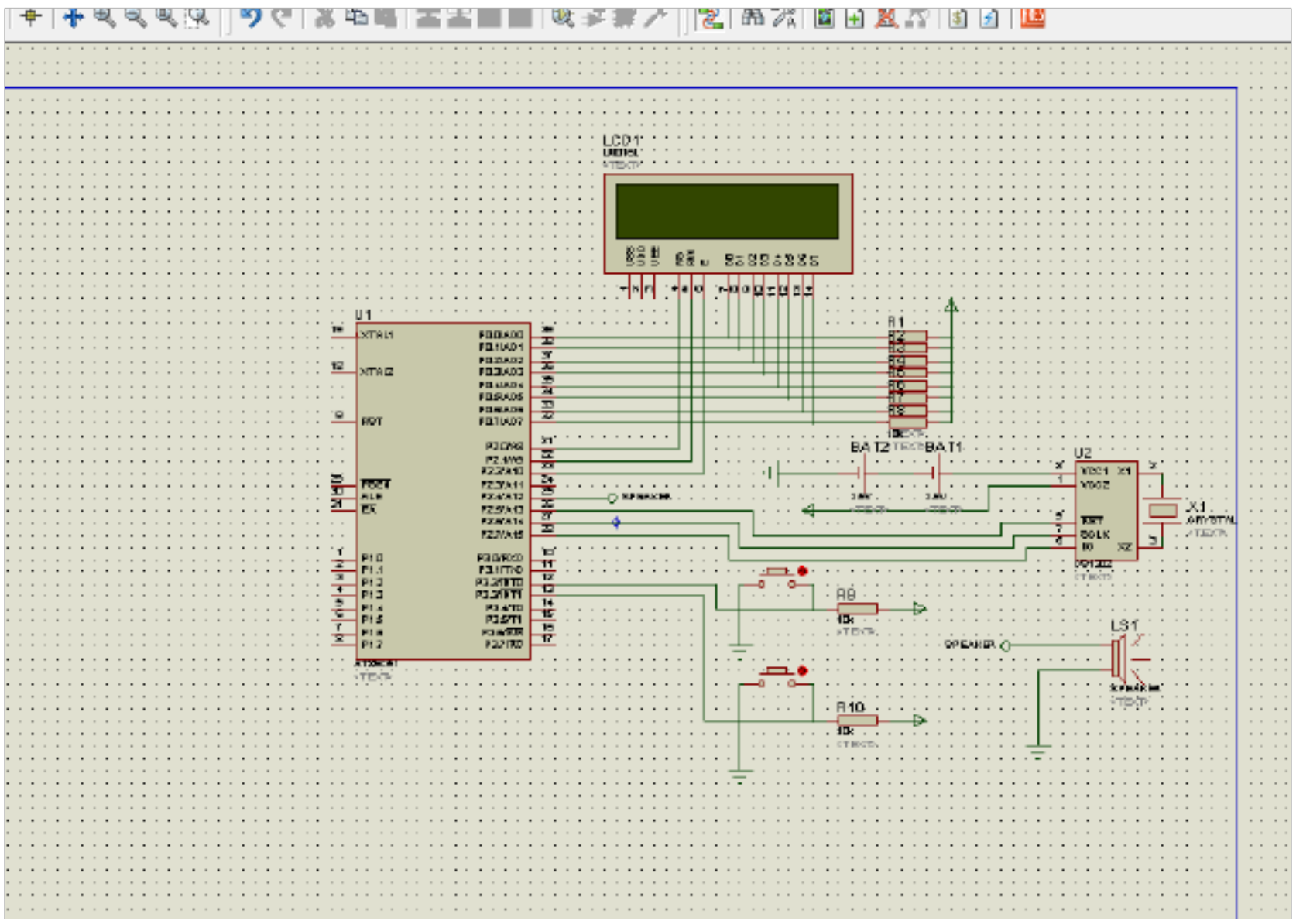
单片机课程设计是以课题或项目设计方式开展的一门课程,具有较强的综合性、实践性，是工科、工程类院校或职业类院校电类专业在校生的必修课，是将单片机原理与应用课程的理论知识转变为应用技术的重要教学环节。这一环节不但能加深对单片机原理的理解，而且还能培养学生的实践动手能力，开发学生的分析、解决问题的能力。单片机课程设计环节的训练能够让学生知道单片机工程项目的制作过程，使学生尽早了解单片机系统的开发流程。

2. 总体方案设计

2.1、方案图



2.2、面板布置图



2.3、方案讨论

方案一:采用实时时钟芯片

实时时钟芯片具备年、月、日、时、分、秒计时功能和多点计时功能，计时数据的更新每秒自动进行一次，不需程序干预。计算机可通过中断或查询方式读取计时数据进行显示，因此计时功能的实现无需占用 CPU的时间，程序简单。此外，实时时钟芯片多数带有锂电池做后备电源，具备永不停止的计时功能；具有可编程方波输出功能，可用做实时测控系统的采样信号等；有的实时时钟芯片内部还带有非易失性 RAM 可用来存放需长期保存但有时也需变更的数据，由于功能完善，精度高，软件程序设计相对简单，且计时不占用 CPU时间，因此，在工业实时测控系统中多采用这一类专用芯片来实现实时时钟功能。

方案二：软件控制

利用单片机内部的定时/计数器进行中断定时，配合软件延时实现时、分、秒的计时及秒表计时。该方案节省硬件成本，且能使设计者对单片机的指令系统能有更深入的了解，从而掌握单片机应用技术 MCS-51汇编语言程序设计方法，因此，本系统设计采用此种软件控制方法来实现计时。而由于 Atmel 公司的 AT89C51是一种自带 4KB Flash 存储器的低电压、高性能的 CMOS 8 位微处理器。该器件采用 Atmel 高密度非易失存储器制造技术制造，与工业标准的 MCS-51指令集和输出引脚相兼容。AT89C51将多功能 8 位 CPU和闪存集成在单个芯片中，是一种高效的微控制器，使用也更方便，寿命更长，可以反复擦除 1000 次。形成了功能强大、使用灵活和具有较高性能价格比的微控制器。它的功能强大，而且也比较容易购买，故本设计中所选的单片机为 AT89C51单片机。

2.4、明晰任务

采用 AT89C51单片机作为系统的控制核心。时钟数据通过市场上流行的时钟芯片 DS1302来获取。DS1302是 DALLAS 公司推出的涓流充电时钟芯片，内含一个实时时钟/日历和 31 字节静态 RAM，可以通过串行接口与计算机进行通信，使得管脚数量减少。实时时钟/日历电路能够计算 2100 年之前的秒、分、时、日、星期、月、年的，具有闰年自动判断调整的能力。定时电路能够实现自定任意时刻自动开/关屏，采用 LCD LM016L 显示年、月、周、天、时、分、秒。通过按键开关实现微调，确保计时精度：误差≤1 秒/月。

DS1302时钟芯片的主要功能特性：

(1) 能计算2100年之前的年、月、日、星期、时、分、秒的信息；每月的天数和闰年的天数可自动调整；时钟可设置为24或12小时格式。

(2) 31B 的 8 位暂存数据存储RAM

(3) 串行I/O 口方式使得引脚数量最少。

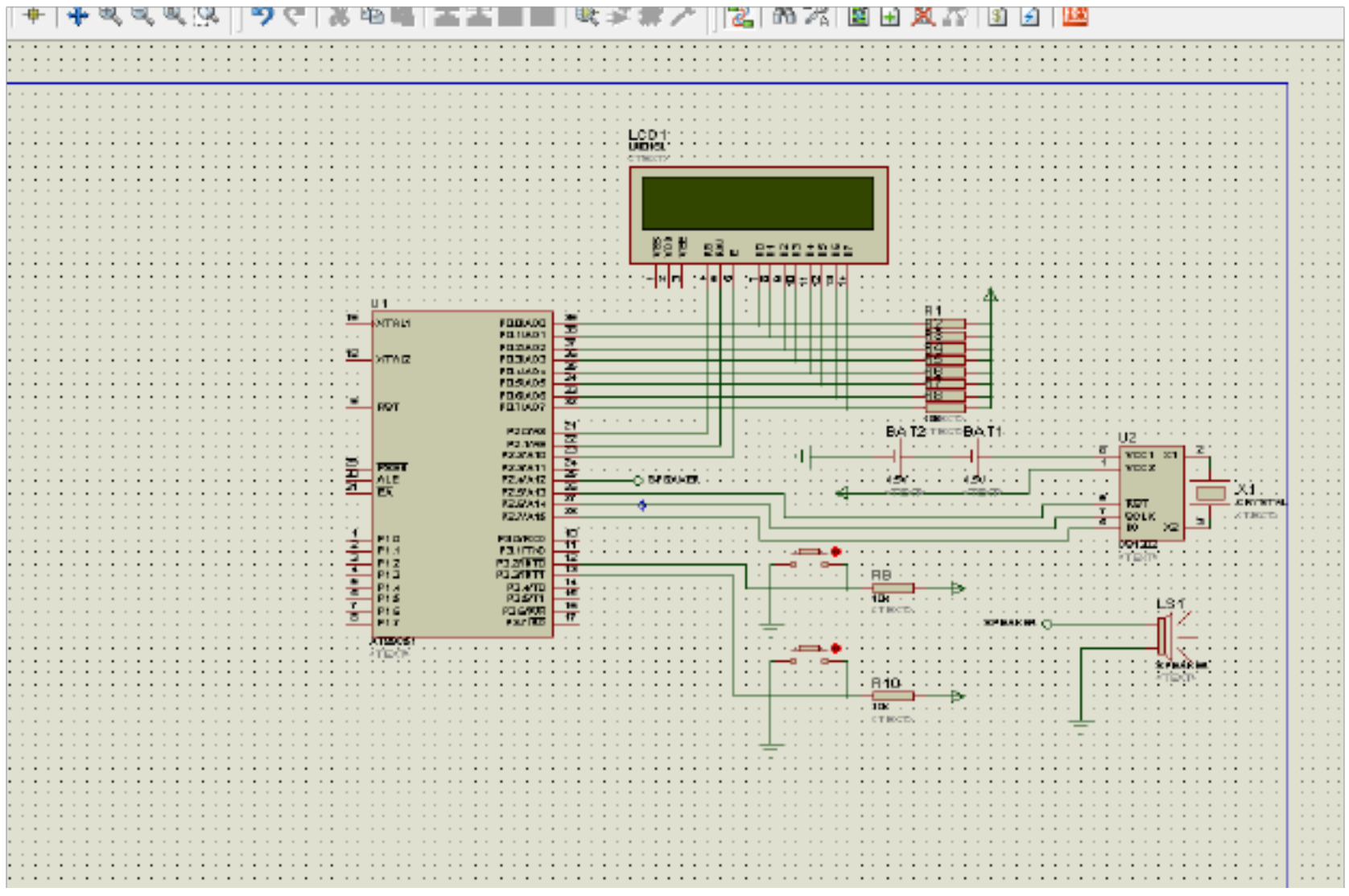
(4) DS1302与单片机之间能简单地采用同步串行的方式进行通信，仅需3根线。

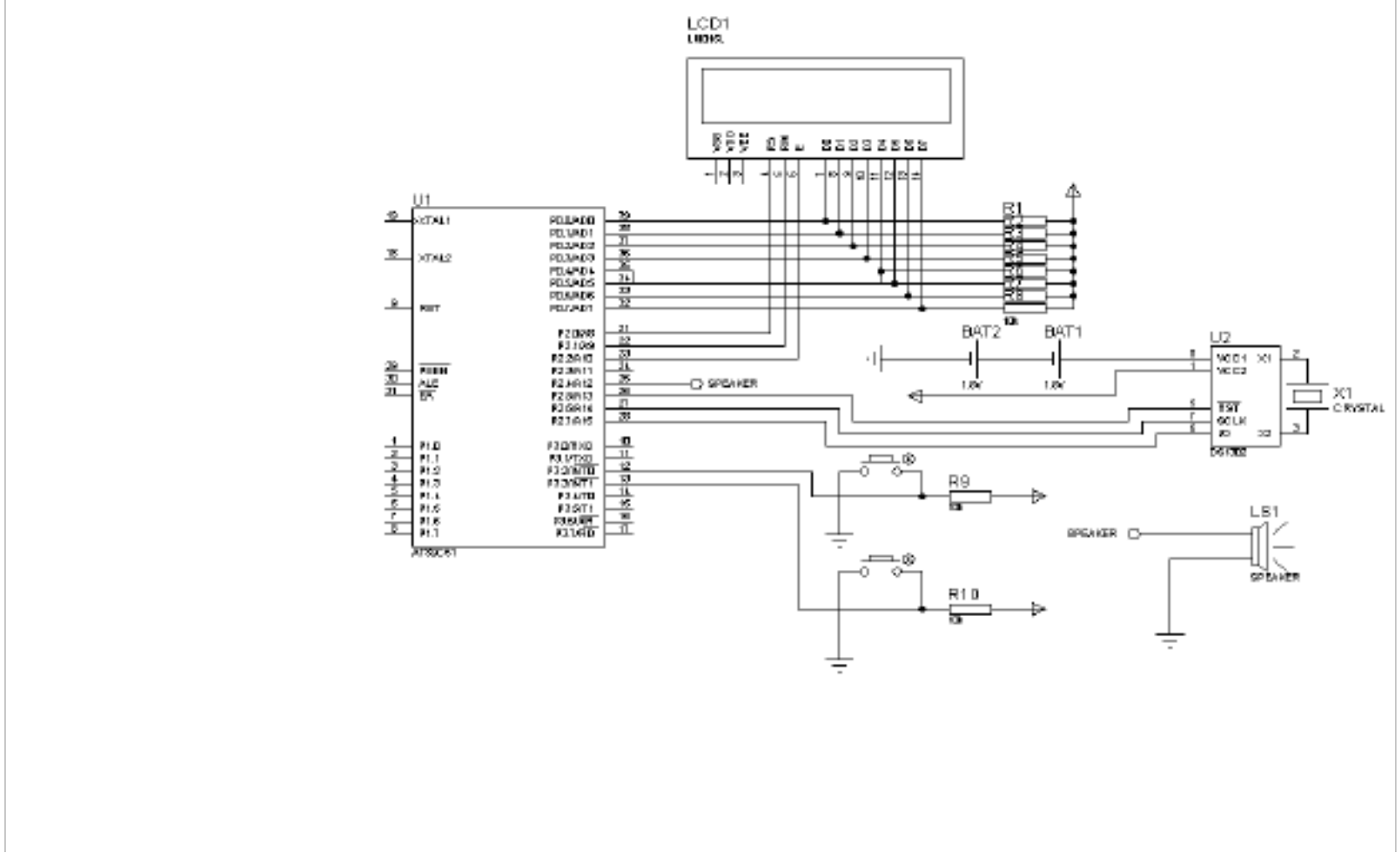
(5) 宽范围工作电压2.0-5.5V。

(6) 工作电流为2.0A时，小于300nA

(7) 功耗很低，保持数据和时钟信息时功率小于1mW

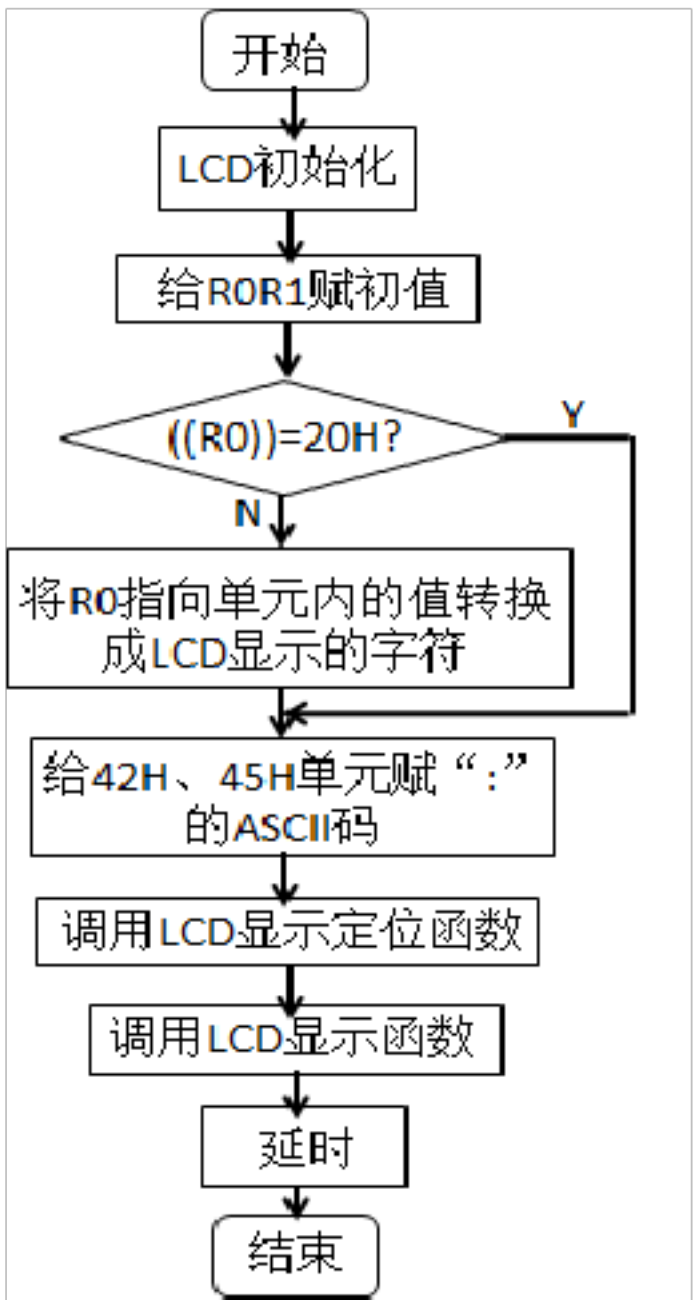
3. 电路原理图



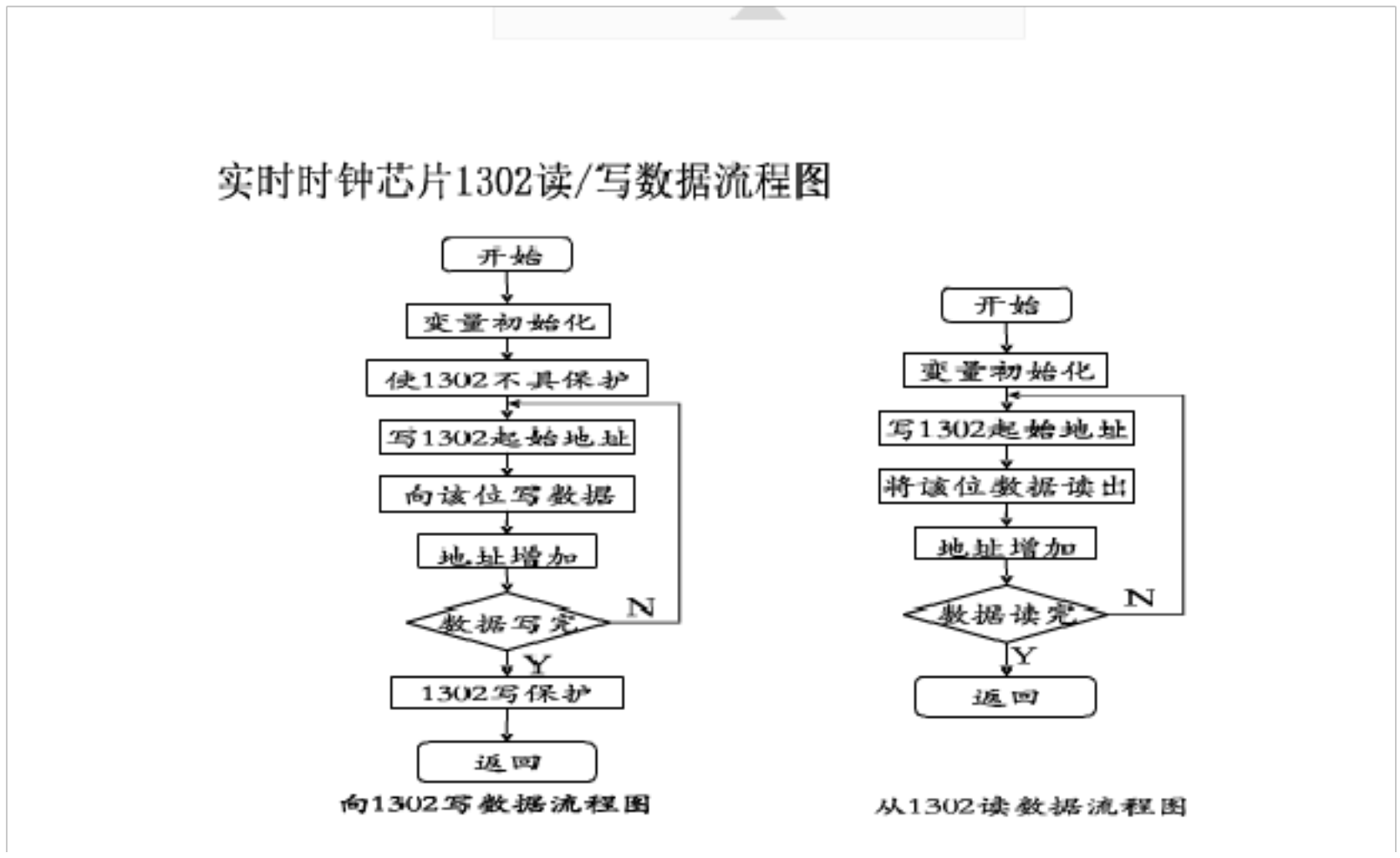


4．程序框图

4.1、显示子程序流程图



4.2、实时时钟芯片 1302 读/写数据流程图



5. 编程序

源程序见附录部分

6. 调试

6.1、软件调试

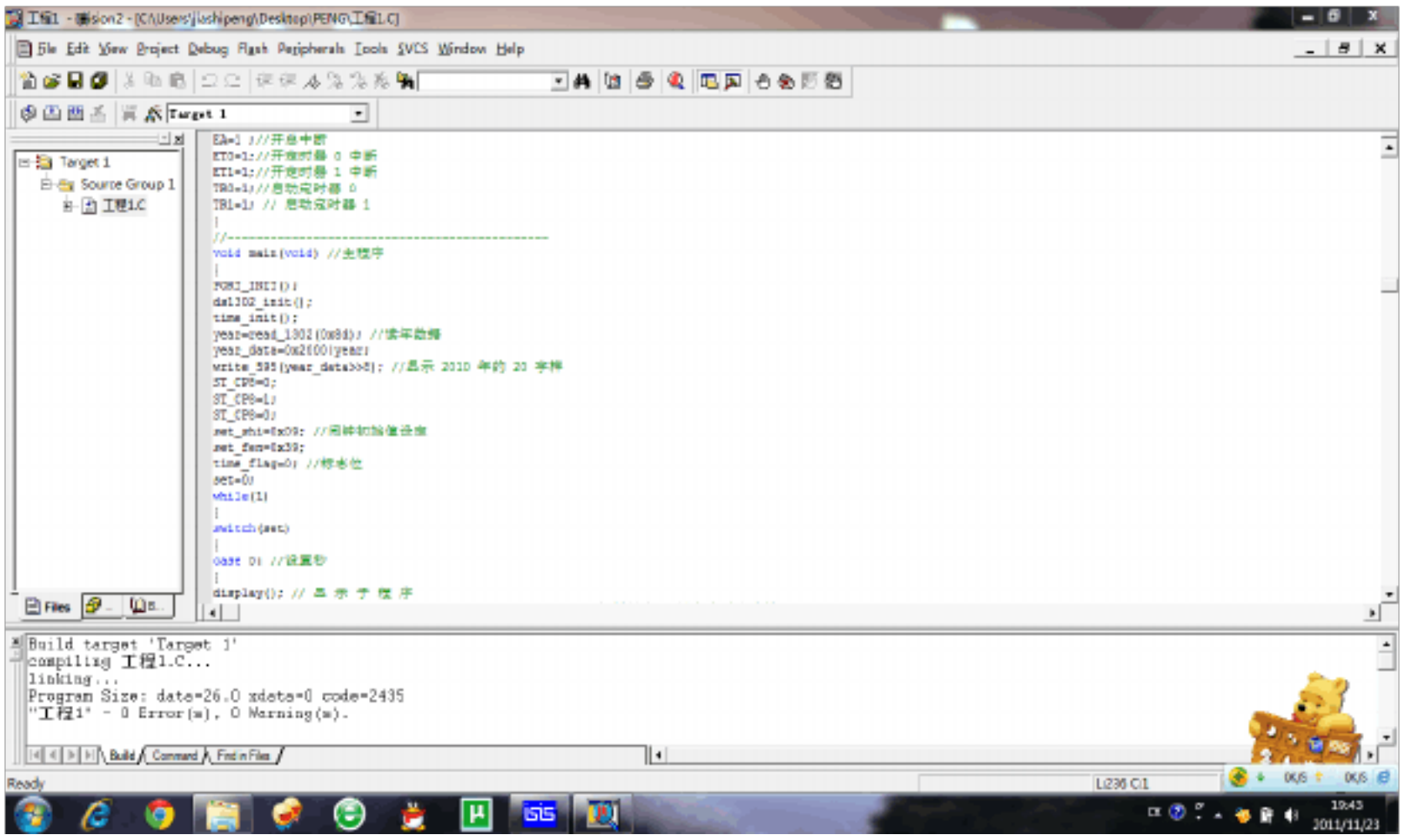
目前设计过程中容易造成元件和仪器仪表的损坏，而借助 Keil 和 Proteus 进行单片机系统的开发，可以节省设计成本，提高设计速度。

Keil 软件包是一个功能强大的开发平台,它包括项目管理器、CX51编译器、AX51宏汇编器、BL51/LX51连接定位器、RTX51实时操作系统、Simulator 软件模拟器及 Monitor51 硬件目标调试器。它是一种集成化程度高的文件管理编译环境，主要功能为编译 C语言源程序，汇编程序或混合语言源程序，连接和定位目标文件和库，创建 HEX文件，调试目标程序等。Keil 是目前最好的 51 单片机开发工具之一。Keil 支持软件模拟仿真（Simulator ）和用户目标调试（Monitor51 ）两种工作模式。前者不需要任何单片机硬件即可完成用户程序仿真、调试，后者利用硬件目标板中的监控程序可以直接调试目标硬件系统。

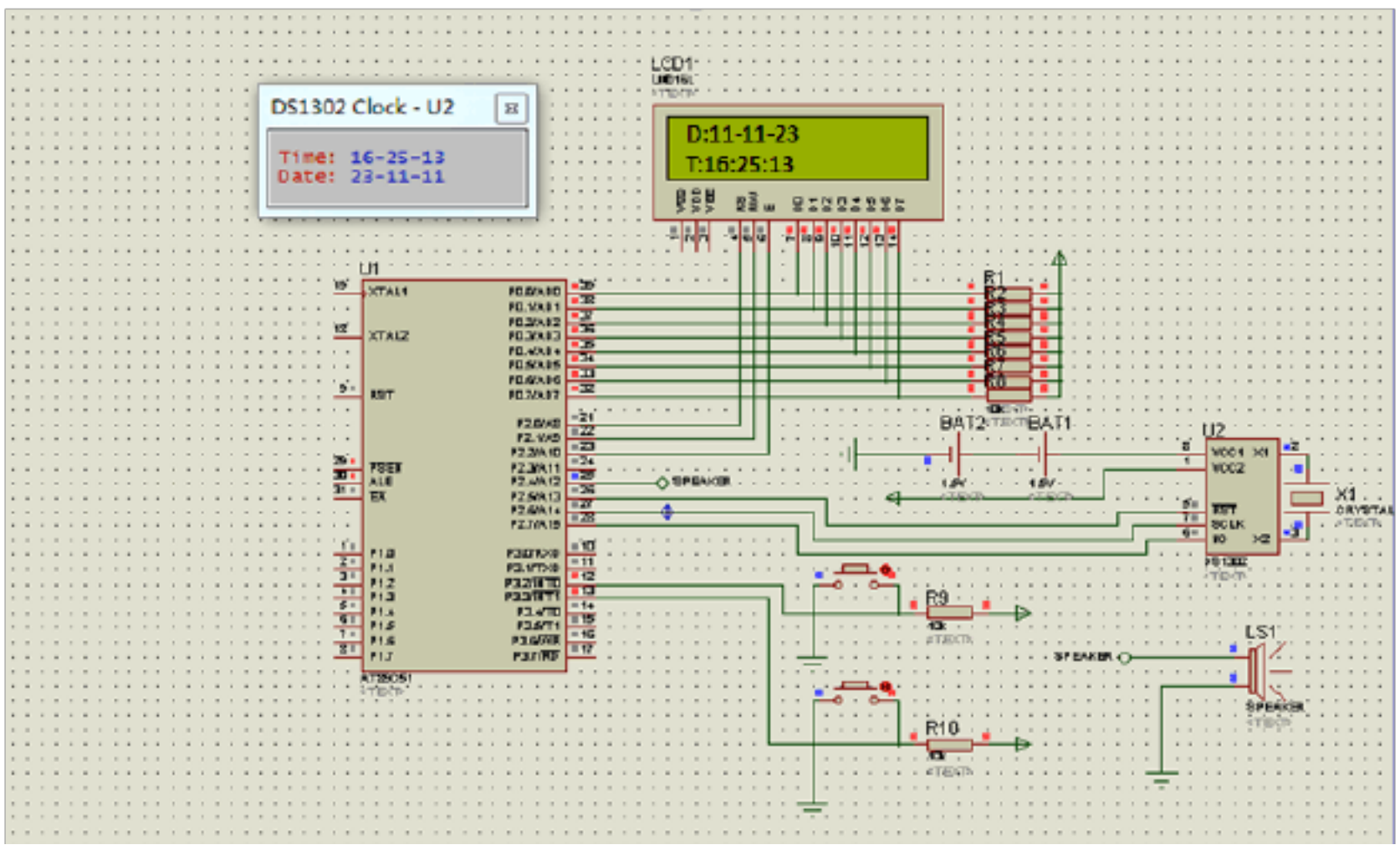
Proteus 是一个完整的嵌入式系统软件、硬件设计仿真平台，它包括原理图输入系统 ISIS 、带扩展的 Prospice 混合模型仿真器、动态元件库、高级图形分析模块和处理器虚拟系统仿真模型 VSM.ISIS 是 Proteus 系统的中心，具有超强的控制原理设计环境。ProteusVSM 最重要的特点是能把微处理器软件作用在处理器上，并和该处理器的任何模拟和数字元件协同仿真，仿真执行目标码就像在真正的单片机系统上运行一样，VSMCPU模型能完整仿真 I/O 接口、中断、定时器、通用外部设备口及其他与 CPU有关的外部设备，甚至能仿真多个处理器。

6.2、仿真调试

Keil 仿真：



Proteus 仿真：



7. 自我感想

经历过这么多天不间断的课程设计，我们有挺多感触的，从最基本上说我们看到了，也意识到了自己的不足，对于不断克服的各种阻碍也让我们体会到了课程设计的意义所在。

对于只接触课本只动笔杆的我们，面临实际的设计尺寸，让我们很是尴尬，

都说理论联系实际，真正到联系的时候才发现挺困难的，不过正是理论知识的各种补充才让我们能最终完成任务，然后深深地体会到理论对现实的指导作用。我们现在最缺乏的就是实际工作经验，而理论联系实践并不像我们想象的那么简单，他需要坚实的理论基础和实际工作经验。坚实的理论基础决定了我必须坚持学习新的知识新的理论，完善了自己的知识结构，才能在以后的实际中轻松面对，才能设计出更好的更有益于人们生活与工作的机械，才能跟上时代的步伐，不被淘汰。

在这个一边忙着复习忙着考试又要准备课程设计的日子里，真真正正的体会到了时间的宝贵，有点像高中忙忙碌碌的生活，不过能按时完成课程设计对我们来说也是一个莫大的安慰。

严谨和细心是做机械设计的必要态度，要想做好一件事，就必须一丝不苟、态度认真。俗话说:“失之毫厘，谬之千里。”在机械设计上尤其应该注意。在以后的工作中，你的很小的一个疏忽将会造成一个公司很大的损失，甚至给用户带去生命危险，而自己也会为自己的不负责任行为付出代价。

再者就是设计中要严谨和细心，对于机械是不能出差错的，任何的微小误差都可能产生不可预计的后果，当然对于我们来说就是设计中要走一些弯路，而且在这个严重缺少时间又惦记回家问题的我们来说也是一个很严重的后果。

不过，困难虽是难免的，但我们有信心就能并且已经战胜了困难，完成了这个无比揪心的课程设计。

因为时间等各种关系设计中难免有些不足还请老师助教给予批评和帮助。

8．参考书目

《单片机原理及其应用教程》　张元良 主编

《MCS-51系列单片机原理及应用》 孙涵芳　主编

《新概念 51 单片机 C语言教程》 郭天祥　主编

《51 单片机课程设计》 周向红　主编

9．附录：C语言编程源程序

```
#include<reg51.H>
#include<intrins.h>
//-----------------------------------
#define uint unsigned int
#define uchar unsigned char
//-----------------------------------
/*uchar              code
table[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf 8,
0x80,0x98,0x88,0x83,0xc6,0xa1,0x86,0x8e};//          共 阳极数码管代码 */
uchar              code
```

```
xingqi[8]={0x00,0x07,0x01,0x02,0x03,0x04,0x05, 0x06};
//星期显示代码
uchar
miao,shi,fen,date,month,day,year,year10,set,mun,set_shi,set_fen,time_
flag;
 // 全 局 定 义
 uint year_data,t;
 //-----------------------------------
sbit SCLK=P3^5; //DS1302     通讯线定义
sbit DIO=P3^6;
sbit RST=P3^7;
sbit speak=P0^0;
sbit DS=P2^0; //595       通讯线定义
sbit SH_CP=P2^1;
sbit ST_CP1=P2^2;
sbit ST_CP2=P2^3;
sbit ST_CP3=P2^4;
sbit ST_CP4=P2^5;
sbit ST_CP5=P2^6;
sbit ST_CP6=P2^7;
sbit ST_CP7=P3^0;
sbit ST_CP8=P3^1;
sbit OE1=P1^0;
sbit OE2=P1^1;
sbit OE3=P1^2;
sbit OE4=P1^3;
sbit OE5=P1^4;
sbit OE6=P1^5;
sbit OE7=P1^6;
sbit OE8=P1^7;
sbit K1=P3^2;//    按键接口定义
sbit K2=P3^3;
sbit K3=P3^4;
sbit K4=P0^1;
sbit K5=P0^2;
//-----------------------------------
void write_595(uchar temp) //         写 74HC595 一个字节
{
uchar temp_595,i;
temp_595=temp;
for(i=0;i<8;i++)
  {
 SH_CP=0;
 _nop_();_nop_();_nop_();
```

```
if(temp_595&0x80)
{
DS=1;
}
else
{
DS=0;
}
_nop_();_nop_();_nop_();
SH_CP=1;
temp_595<<=1;
}
}
//-----------------------------------------
void delay(uint z) //Nms        延时
{
uint x,y;
for(x=z;x>0;x--)
for(y=112;y>0;y--);
}
//-----------------------------------------
void delaynus(uint z) //ums        延时
{
uint x,y;
for(x=z;x>0;x--)
for(y=10;y>0;y--);
}
//-----------------------------------------
void write(uchar date) //        写入 DS1302 一个字节
{
uchar temp,i;
RST=1;
SCLK=0;
temp=date;
for(i=0;i<8;i++)
{ SCLK=0;
if(temp&0x01)
DIO=1;
else
DIO=0;
SCLK=1;
temp>>=1;
}
}
```

```
//-----------------------------------
uchar read() //      读出 DS1302 一个字节
{
uchar a,temp;
RST=1;
for(a=8;a>0;a--)
{
temp>>=1;
SCLK=1;
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
SCLK=0;
if(DIO)
{
temp=temp|0x80;
}
else
{
temp=temp|0x00;
}
}
return (temp);
}
//-----------------------------------
void write_1302(uchar add,uchar dat) //          写 DS1302数据
{
RST=0;
SCLK=0;
RST=1;
write(add);
write(dat);
SCLK=1;
RST=0;
}
//-----------------------------------------
uchar read_1302(uchar add) //          读 DS1302数据
{
uchar temp;
RST=0;
```

```
SCLK=0;
RST=1;
write(add);
temp=read();
SCLK=1;
RST=0;
return(temp);
}
//-----------------------------------------
void display() //       显示子程序
{
miao=read_1302(0x81); //     读秒
fen=read_1302(0x83); //      读分
shi=read_1302(0x85)&0x3f;//     读时
date=read_1302(0x87); //      读日
month=read_1302(0x89); //     读月
year=read_1302(0x8d); //      读年
day=read_1302(0x8B); //     读星期
write_595(miao); //      显示秒
ST_CP1=0;
ST_CP1=1;
ST_CP1=0;
delaynus(10);
write_595(fen); //       显示分
ST_CP2=0;
ST_CP2=1;
ST_CP2=0;
delaynus(10);
write_595(shi); //       显示时
ST_CP3=0;
ST_CP3=1;
ST_CP3=0;
delaynus(10);
write_595(date); //       显示日
ST_CP4=0;
ST_CP4=1;
ST_CP4=0;
delaynus(10);
write_595(month); //      显示月  读
ST_CP5=0;
ST_CP5=1;
ST_CP5=0;
delaynus(10);
write_595(year); //       显示年
```

```
ST_CP6=0;
ST_CP6=1;
ST_CP6=0;
delaynus(10);
write_595(xingqi[day]); //        显示星期
ST_CP7=0;
ST_CP7=1;
ST_CP7=0;
delaynus(10);
}
//--------------------------------------
void ds1302_init() //1302          初始化
{
RST=0;
SCLK=0;
/*
write_1302(0x80,0x00);//      设置初始值 SEC
write_1302(0x82,0x00);//      设置初始值 MIN
write_1302(0x84,0x00);//      设置初始值 HR
write_1302(0x86,0x00);//      设置初始值 DATE
write_1302(0x88,0x00);//      设置初始值 MONTH
write_1302(0x8A,0x00);//      设置初始值 DAY
*/
write_1302 (0x8C,0x10);//      设置初始值 YEAR
}
//----------------------------------------
void PORT_INIT() //   端口初始化
{
P0=0XFE;
P1=0X00;
P2=0X00;
P3=0XFC;
}
void time_init() //        定时器初始化
{
TMOD=0x11; //设置定时 器        都为工作方式 1
TH0=(65536-50000)/256; //    装入初值
TL0=(65536-50000)%256;
TH1=(65536-10000)/256; //    装入初值
TL1=(65536-10000)%256;
PT0=1; //T0   定时器优先级最高
EA=1 ;// 开总中断
ET0=1;// 开定时器 0 中断
ET1=1;// 开定时器 1 中断
```