

多点式温度测量数字钟

摘要: 工、农业生产和日常生活中,对温度的测量及控制占据着极其重要地位。在实践中为了更好的分析数据能精确的记录相应温度的时间同样重要。为得到广泛应用,本系统以简单,易行,可靠为出发点从硬件和软件两方面介绍了 89C51 单片机温度测量数字钟系统的设计思路。

DS18B20 是一种可组网的高精度数字式温度传感器,由于其具有单总线的独特优点,可以使用户轻松地组建起传感器网络,并可使多点温度测量电路变得简单、可靠。数字钟部分采用实时时钟芯片 DS12C887。该系统实现容易,可用于仓库测温,楼宇空调控制及生产过程监控等领域。

关键词: AT89C51 单片机; 多点式; DS12C887; DS18B20; 硬件原理图; 设计。

温度检测系统有则共同的特点:测量点多、环境复杂、布线分散、现场离监控室远等。若采用一般温度传感器采集温度信号,则需要设计信号调理电路、A/D 转换及相应的接口电路,才能把传感器输出的模拟信号转换成数字信号送到计算机去处理。这样,由于各种因素会造成检测系统较大的偏差;又因为检测环境复杂、测量点多、信号传输距离远及各种干扰的影响,会使检测系统的稳定性和可靠性下降。所以多点温度检测系统的设计的关键在于两部分:温度传感器的选择和主控单元的设计。

采用数字温度芯片 DS18B20 测量温度,输出信号全数字化。便于单片机处理及控制,省去传统的测温方法的很多外围电路。且该芯片的物理化学性很稳定,它能用做工业测温元件,此元件线性较好。这样,测温系统的结构就比较简单,体积也不大,且由于 AT89C51 可以带多个 DS18B20,因此可以非常容易实现多点测量。轻松的组建传感器网络。

数字钟采用 DS12C887 实时时钟芯片功能丰富,由于 DS12C887 能够自动产生世纪、年、月、日、时、分、秒等时间信息且 DS12C887 中自带有锂电池,外部掉电时,其内部时间信息还能够保持 10 年之久。

(一) 方案设计

1, 主体方案框图 图 1

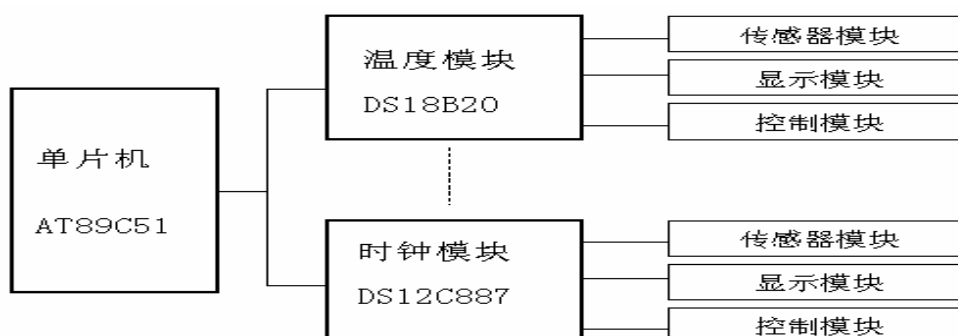


图 1 主体方案

2, 硬件电路设计

A, 温度模块

DS18B20 是 DALLAS 公司生产的一线式数字温度传感器,具有 3 引脚 TO-92

小体积封装形式。测温分辨率可达 0.0625°C ，被测温度用符号扩展的 16 位数字量方式串行输出 CPU 只需一根端口线就能与诸多 DS18B20 通信，占用微处理器的端口较少，可节省大量的引线和逻辑电路。测量温度范围为 $-55^{\circ}\text{C}\sim+125^{\circ}\text{C}$ ，在 $-10\sim+85^{\circ}\text{C}$ 范围内，精度为 $\pm 0.5^{\circ}\text{C}$ 。现场温度直接以“一线总线”的数字方式传输，大大提高了系统的抗干扰性。

DS18B20 有 4 个主要的数据部件：

① 64 位激光 ROM。64 位激光 ROM 从高位到低位依次为 8 位 CRC、48 位序列号和 8 位家族代码(28H)组成。

② 温度灵敏元件。

③ 非易失性温度报警触发器 TH 和 TL。可通过软件写入用户报警上下限值。

④ 配置寄存器。配置寄存器为高速暂存存储器中的第五个字节。DS18B20 在 0 工作时按此寄存器中的分辨率将温度转换成相应精度的数值，其各位定义如表 1 所示。

表 1

TM	R1	R0	1	1	1	1	1
MSB		DS18B20 配置寄存器结构图				LSB	

构成多点式硬件连接电路如下图 2：

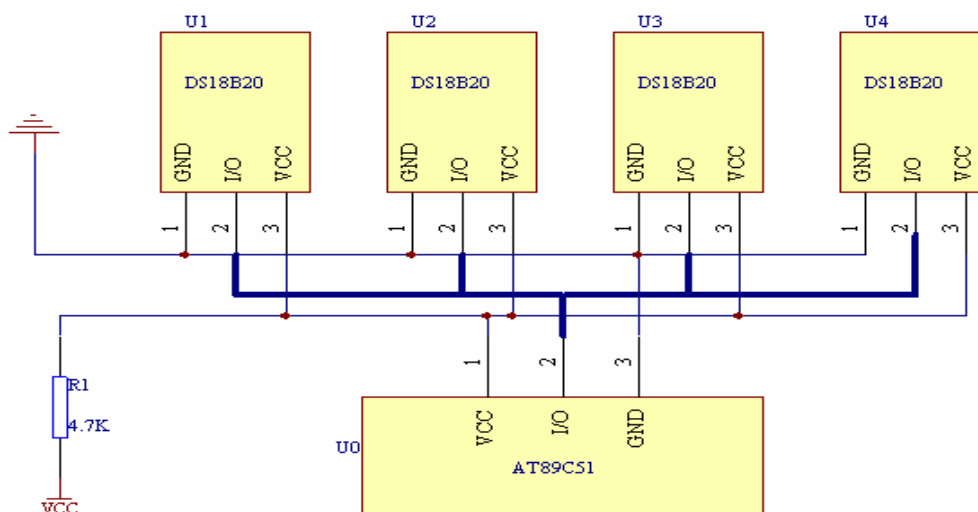


图 2 硬件连接电路

本系统为多点温度测试。DS18B20 采用外部供电方式，理论上可以在一根数据总线上挂 256 个 DS18B20，但时间应用中发现，如果挂接 25 个以上的 DS18B20 仍旧有可能产生功耗问题。另外单总线长度也不宜超过 80M，否则也会影响到数据的传输。在这种情况下我们可以采用分组的方式，用单片机的多个 I/O 来驱动多路 DS18B20。在实际应用中还可以使用一个 MOSFET 将 I/O 口线直接和电源相连，起到上拉的作用。

B. 时钟模块

时钟电路中我们采用 DALLAS 公司生产的 DS12C887。可以通过 CPU 读取 DS12C887 的内部时标寄存器得到当前的时间和日历，也可通过选择二进制码或 BCD 码初始化芯片的 10 个时标寄存器。其 114bit 非易失性静态 RAM 可供用户使用，可在主机掉电时来保存一些重要的数据。DS12C887 的 4 个状态寄存器用来控制和指出 DS12C887 模块的当前工作状态，除数据更新周期外，程序可随时

读写这 4 个寄存器。DS12C887 处于正常工作状态时，每秒钟将产生一个更新周期，芯片处于更新周期的标志是寄存器 A 中的 UIP 位为“1”。

在更新周期内，芯片内部时标寄存器数据处于更新阶段，故在该周期内，微处理器不能读芯片时标寄存器的内容，否则将得到不确定数据。更新周期的基本功能主要是刷新各个时标寄存器中的内容，同时秒时标寄存器内容加 1，并检查其他时标寄存器内容是否有溢出。如果有溢出则相应进位日、月、年。

DS12C887 通过 CPU 读取 DS12C887 的内部时标寄存器得到当前的时间和日历，其管脚排列见下，R/W：读/写输入端，CS：片选输入，低电平有效，AD0~AD7：复用地地址数据总线，GND、VCC：直流电源，单片机 P0 口做为 I/O 口与时钟芯片并行接口，读写口相互连接，如表 2。

表 2 DS12887

引脚符号	INTEL (MOT= V _{ss})	MOTOROLA (MOT= V _{dd})
ALE	ALE	AS
DS	RD	DS
R/W	WR	RD/WR

硬件连接如图 3 所示：

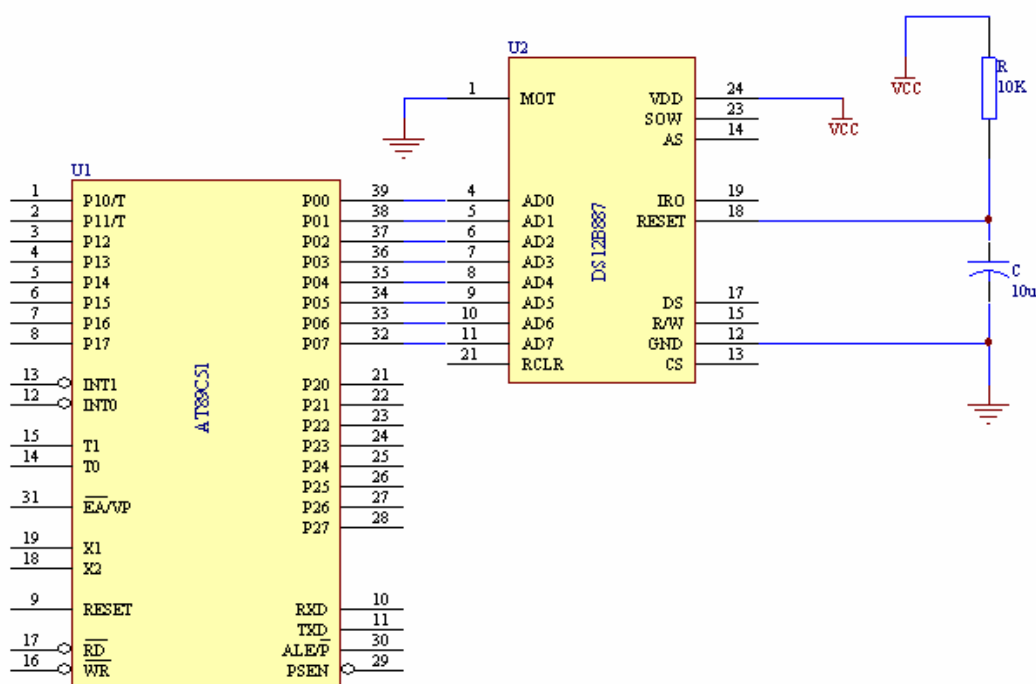


图 3 AT89C51 与 DS12C887 接口电路

C, 显示模块

显示模块采用八段共阳极 LED 数码管静态显示，由两片四位 LED 构成，其中 a, b, c, d, e, f, g, dp, 为公用管脚，图中以总线相连，LED 与单片机并行接口，P1 口为 I/O 口，P2 口对八位 LED

构成片选，其中四位用于显示温度，另四位用于显示时钟，硬件图如图 4 所示：

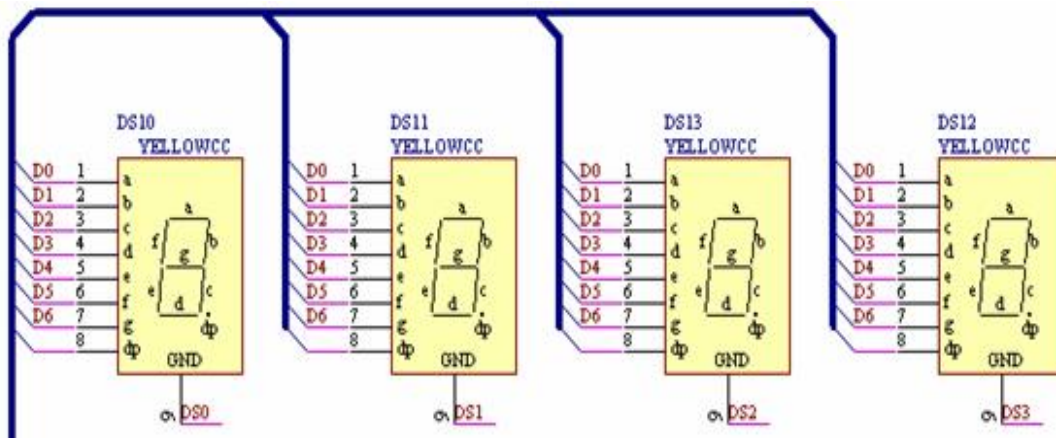


图 4 LED 硬件图

D, 硬件总图如附图 1

3, 软件设计

整个系统的功能是由硬件电路配合软件来实现的，当硬件基本定型后，软件的功能也就基本定下来了。从软件的功能不同可分为两大类：一是监控软件（主程序），它是整个控制系统的核心，专门用来协调各执行模块和操作者的关系。二是执行软件（子程序），它是用来完成各种实质性的功能如测量、计算、显示、通讯等。每一个执行软件也就是一个小的功能执行模块。这里将各执行模块一一列出，并为每一个执行模块进行功能定义和接口定义。各执行模块规划好后，就可以规划监控程序了。主程序流程图如图 5 所示。

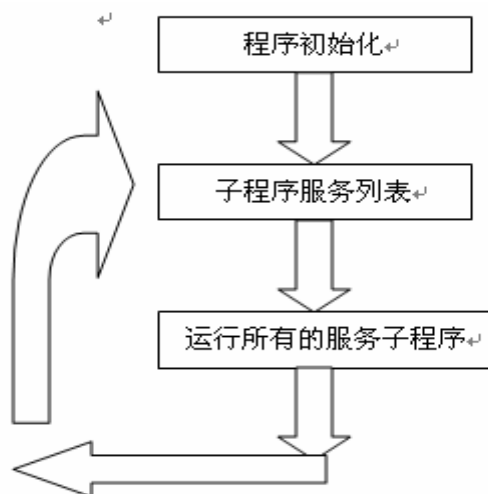


图 5 主程序流程图

主程序调用了三个大块子程序，分别为温度测试程序，时钟读写程序，数码管显示程序。

温度测试程序：对温度芯片送过来的数据进行处理，进行判断和显示。时钟读写

初始程序：向时钟芯片写初始数据，将时钟数据送入单片机加以显示。

数码管显示程序：向数码管的显示送数，控制系统的显示部分。

A, 温度测试子程序

温度传感器 DS18B20 程序框图如图 6 所示：

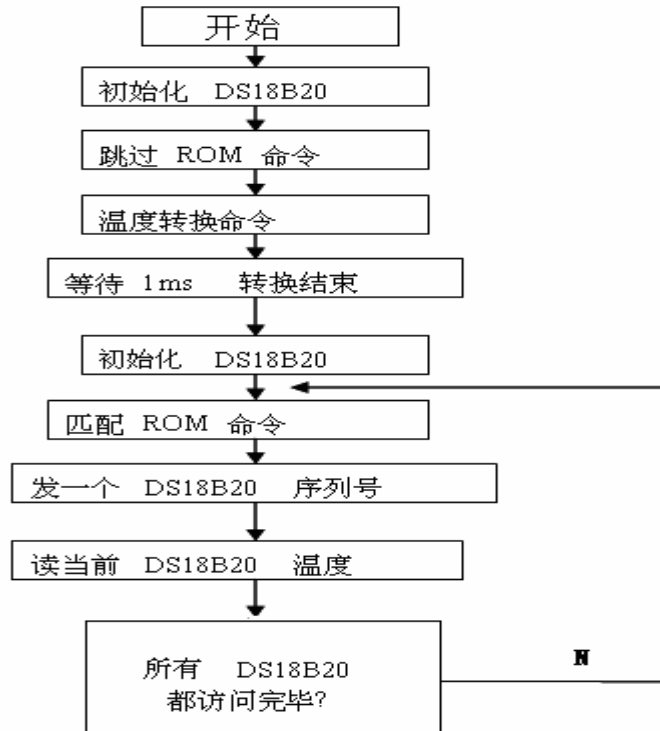


图 6 DS18B20 程序框图

其初始化程序见下:

INIT_1820:

```

SETB  P3.2
NOP
CLR   P3.2
MOV   RO,#06BH
MOV   R1,#03H

```

B、数字钟模块

DS12887 的地址/数据总线分时复用，程序按时序图描述。

时序图如图 6 所示:

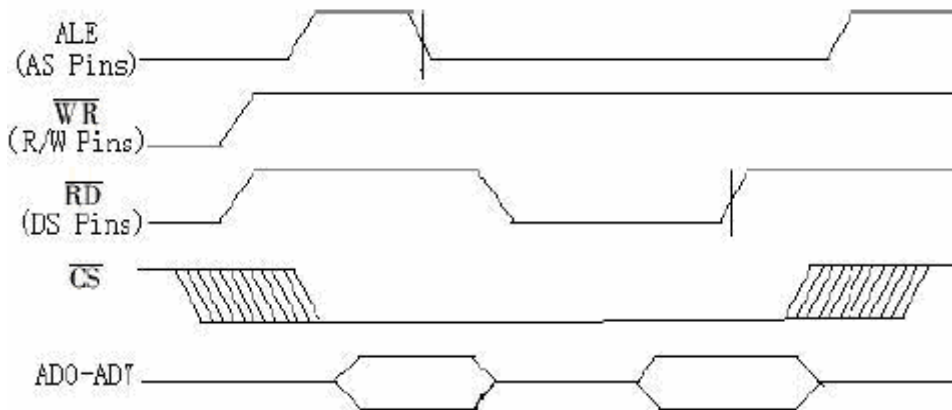


图 6 DS12887 INTEL 模式读时序

DS12887 初始化程序见下:

```
*****
DS12887:
MOV DPTR, #0F0BH
MOV A, #82H ; ET 位=1(停止时钟更新)
MOV @DPTR, A ; D1 位=1 设时钟为 24 小时制
MOV DPTR, #0F00H ; 秒地址
MOV A, 36H ; 秒缓冲器内数据
MOVX @DPTR, A ; 置“秒”时间
INC DPTR
INC DPTR ; 分寄存器地址
MOV A, 37H
MOVX @DPTR, A ; 置“分”时间
INC DPTR INC DPTR ; 时寄存器地址
MOV A, 38H
MOVX @DPTR, A ; 置“时”时间
INC DPTR ; “A”寄存器地址
MOV A, #2FH
MOVX @DPTR, A ; 开启时钟, 设方波输出为 500MS
INC DPTR ; “B”寄存器地址
MOV A, #5AH
MOVX @DPTR, A ; 允许周期中断, 时间更新中断
RET
*****
```

C、数码管显示程序

由于数码管并行接口较为明了，其程序见后总程序。

(二) 结论

众所周知，用单片机构成数字钟及温度测量仪均是一个比较传统的话题，而本课题是可望在传统的基础上能将其加以融合，并能构成简易可行的多点式测量。

在多点测温系统中，传统的测温方法是将模拟信号远距离采样进行 AD 转换，而为了获得较高的测温精度，就必须采取措施解决由长线传输，多点测量切换及放大电路零点漂移等造成的误差补偿问题。采用数字温度芯片 DS18B20 测量温度，输出信号全数字化。便于单片机处理及控制，省去传统的测温方法的很多外围电路。这便是本课题的创新点所在。

结果分析：

- 1、时钟显示较为准确。
- 2、各点温度显示正常，串口传输数据正确。
- 3、因为芯片是塑料封装，所以对温度的感应灵敏度不是相当高，需要一个很短的时间才能达到稳定。

致此本组设计基本完成了预期的目标，系统在硬件实时显示方面做的比较好。但是由于时间仓促、条件有限，设计成果并不是很完美，还存在下面问题：多点式不易实现，串口通讯不稳定，未对温度数值统计处理以及存储。我准备在今后的工作过程中进一步完善此设计。

前景：采用温度芯片 DS18B20 测量温度，可以体现系统芯片化这个趋势。部分功能电路的集成，使总体电路更简洁，搭建电路和焊接电路时更快。而且，集成块的使用，有效地避免外界的干扰，提高测量电路的精确度。所以集成芯片的使用将成为电路发展的一种趋势。本方案应用这一温度芯片，也是顺应这一趋势。

技术关键及难点：

(1) 对硬件结构简单的单线数字温度传感器 DS18B20 进行操作，需要用较为复杂的程序完成。编制程序时必须严格按芯片数据手册提供的有关操作顺序进行，读、写时间片程序要严格按照要求编写。

(2) 有多个测温点时，应考虑系统能实现传感器出错自动指示，进行自动 DS18B20 序列号和自动排序，以减少调试和维护工作量。

(3) DS18B20 在三线制应用时，应将其三线焊接牢固；在两线应用时，应将 VCC 与 GND 接在一起，焊接牢固。若 VCC 脱开未接，传感器只送 85.0 ℃ 的温度值。

(4) 实际应用时，要注意单线的驱动能力，不能挂接过多的 DS18B20，同时还应注意最远接线距离。另外还应根据实际情况选择其接线拓扑结构。

四、参考文献

- [1]. 余永权. 单片机原理及应用. 北京: 电子工业出版社, 1997
- [2]. 吉鹏, 马云峰等. 微机原理与接口技术 [M]. 北京: 高等教育出版社, 2001.
- [3]. 月霞, 孙传友. DS18B20 硬件连接及软件编程[J]. 传感器世界, 2001 (12): 25-29.
- [4]. 胡汉才. 单片机原理及系统设计[M]. 北京: 清华大学出版社, 2002
- [5]. 余永权. AT89C51 系列单片机应用技术[M]. 北京: 北京航空航天大学出版社 2002.

五、心得及体会:

这次创新，我们收获颇丰，此次实习是我们首次用单片机做一个完整的系统，在此之前，我们的知识储备不足是显而易见的，在完成的过程中走了很多弯路，比如由于对硬件方面的东西尚不熟，查阅资料以及准备材料时都没有考虑到实践的可行性，在这里要特别感谢实验室的老师给予了我们耐心的指导，并给我们提出了建设性的意见。可以说，正是通过这些弯路我们才真正学到了不少东西。在做系统的同时，和同学们之间的相互探讨也使我获益匪浅。在此，对他们表示由衷的感谢！

电子信息日新月异飞速发展，人们总是处在不断学习阶段，再加上我们水平有限，所以本设计肯定存在许多不尽如人意的地方，欢迎老师批评指正。

对于自主创新，我们认为这是一个很好的平台让我们真正学会了独立思考，用自己的能力去完成一件属于自己的作品。希望以后有更多的人能参与！

附录一：软件总程序

*****时钟部分*****

DS_DATA	BIT	P3. 3	
TEMPER_H	EQU	28H	;
TEMPER_L	EQU	29H	;
FLAG1	BIT	20H	;是否检测到 18B20 的标志位

```

TEMP            EQU        0CH    ;温度显示标志
SECOND_S       EQU        30H    ;
MINUTE_S       EQU        31H
HOUR_S         EQU        32H
SECOND         EQU        33H
MINUTE         EQU        34H
HOUR           EQU        35H

    ORG    0000H
    AJMP   MAIN
    ORG    0030H
MAIN:
    MOV    SP, #60H
    MOV    RO, #30H
    MOV    R2, #07H

CLC:
    MOV    A, @R0
    CLR    A
    INC    R0
    DJNZ   R2, CLC
    LCALL  WR_DS12887

LOOP_MAIN:
    LCALL  READ_DS12887
    LCALL  DISP
    AJMP   LOOP_MAIN

WR_DS12887:
    MOV    DPTR, #0FE0BH    ; THE ADDRESS OF DS12887' S B REGISTER
    MOV    A, #8BH        ; FANGBO ENABLE
    MOVX   @DPTR, A        ; SET B REGISTER
    MOV    DPTR, #0FE00H   ; SECONDE WRITE
    MOV    A, SECOND_S
    MOVX   @DPTR, A
    INC    DPTR
    INC    DPTR
    MOV    A, MINUTE_S
    MOVX   @DPTR, A
    INC    DPTR
    INC    DPTR
    MOV    A, HOUR_S
    MOVX   @DPTR, A
    MOV    DPTR, #0FE0AH   ; THE ADDRESS OF DS12887' S A REGISTER
    MOV    A, #2FH
    MOVX   @DPTR, A        ; FANGBO OUTPUT
    INC    DPTR
    MOV    A, #0BH

```



```

MOVX @DPTR, A
LCALL DL1MS
RET

```

READ_DS12887:

```

MOV DPTR, #0FE00H
MOVX A, @DPTR
MOV SECOND, A
MOV DPTR, #0FE02H
MOVX A, @DPTR
MOV MINUTE, A
MOV DPTR, #0FE04H
MOVX A, @DPTR
MOV HOUR, A
RET

```

```

DL1MS:      MOV     R6, #14H
DL1:        MOV     R7, #19H
DL2:        DJNZ   R7, DL2
            DJNZ   R6, DL1
            RET

```

```

DISP:      MOV     A, 34H
            LCALL  BCD_CNV
            MOV     40H, R4
            MOV     41H, R3
            MOV     A, 34H
            LCALL  BCD_CNV
            MOV     42H, R4
            MOV     43H, R3
            MOV     R1, #40H
            MOV     R5, #07FH
PLAY:      MOV     P1, #0FFH
            MOV     A, R5
            MOV     P2, A
            MOV     A, @R1
            MOV     DPTR, #TAB
            MOVC   A, @A+DPTR
            ORL    A, #01H
            MOV     P1, A

            MOV     A, R5
            NOP
            NOP

```

```

        JB      ACC. 5, LOOP5
        NOP
        NOP
        CLR    P1.0
LOOP5:  LCALL   DL1MS
        INC    R1
        MOV    A, R5
        JNB   ACC. 4, ENDOUT
        RR     A
        MOV    R5, A

        AJMP   PLAY
ENDOUT: MOV    P1, #0FFH
        MOV    P2, #0FFH
        RET

TAB:    DB     03h, 09fh, 025H, 0DH, 99H, 49H, 41H, 1FH, 01H, 09H

        RET

BCD_CNV:
        MOV    R3, A
        ANL   A, #0FH
        MOV    R4, A
        MOV    A, R3
        SWAP  A
        ANL   A, #0FH
        MOV    R3, A
        RET

```

*****温度部分*****

```

TIMEL   EQU    0EOH
TIMEH   EQU    0B1H
TEMPHEAD EQU 36H
BITST   DATA 20H
TIME1SOK BIT   BITST.1
TEMPONEOK BIT  BITST.2
TEMPL   DATA 26H
TEMPH   DATA 27H
TEMPHC  DATA 28H
TEMPLC  DATA 29H
TEMPDIN BIT   P3.3    we伪指琳
ORG     0000H
LJMP    START
ORG     00BH
LJMP    TOIT
ORG     100H

```

```

START:      MOV SP, #60H
CLSMEM:     MOV R0, #20H
            MOV R1, #60H
CLSMEM1:    MOV @R0, #00H      清寄存器
            INC R0
            DJNZ R1, CLSMEM1
            MOV TMOD, #00100001B
            MOV TH0, #TIMEL
            MOV TLO, #TIMEH    定时期初始化
            SJMP INIT
ERROR:      NOP
            LJMP START
            NOP
INIT:       NOP
            SETB ETO
            SETB TRO
            SETB EA
            MOV PSW, #00H
            CLR TEMPONEOK
            LJMP MAIN
TOIT:      PUSH PSW
            MOV PSW, #10H
            MOV TH0, #TIMEH
            MOV TLO, #TIMEL
            INC R7              定时初直从装
            CJNE R7, #32H, TOIT1
            MOV R7, #00H
            SETB TIME1SOK
TOIT1:     POP PSW              初始化
            RETI
MAIN:      LCALL DISP1
            JNB TIME1SOK, MAIN    检测
            CLR TIME1SOK
            JNB TEMPONEOK, MAIN2
            LCALL READTEMP1
            LCALL CONVTEMP
            LCALL DISPCD
            LCALL DISP1
MAIN2:     LCALL READTEMP
            SETB TEMPONEOK
            LJMP MAIN
INI TDS1820: SETB TEMPDI N
            NOP
            NOP

```

```

                CLR    TEMPDI N
                MOV    R6, #0A0H
                DJNZ   R6, $
                SETB   TEMPDI N
                MOV    R6, #32H
                DJNZ   R6, $
                MOV    R6, #3CH
LOOP1820:      MOV    C, TEMPDI N
                JC     I NI TDS18200UT
                DJNZ   R6, LOOP1820
                MOV    R6, #064H
                DJNZ   R6, $
                SJMP  I NI TDS1820
                RET

I NI TDS18200UT: SETB   TEMPDI N
                RET

READDS1820:    MOV    R7, #08H
                SETB   TEMPDI N
                NOP
                NOP

READDS1820LOOP: CLR    TEMPDI N
                NOP
                NOP
                NOP
                SETB   TEMPDI N
                MOV    R6, #07H
                DJNZ   R6, $
                MOV    C, TEMPDI N
                MOV    R6, #3CH
                DJNZ   R6, $
                RRC    A
                SETB   TEMPDI N
                DJNZ   R7, READDS1820LOOP
                MOV    R6, #3CH
                DJNZ   R6, $
                RET

WRI TEDS1820:  MOV    R7, #08H
                SETB   TEMPDI N
                NOP
                NOP

WRI TEDS1820LOP: CLR    TEMPDI N
                MOV    R6, #07H
                DJNZ   R6, $
                RRC    A

```

```

MOV     TEMPDI N, C
MOV     R6, #34H
DJNZ   R6, $
SETB   TEMPDI N
DJNZ   R7, WRI TEDS1820LOP
RET
READTEMP:  LCALL  I NI TDS1820
MOV     A, #0CCH
LCALL  WRI TEDS1820
MOV     R6, #34H
DJNZ   R6, $
MOV     A, #44H
LCALL  WRI TEDS1820
MOV     R6, #34H
DJNZ   R6, $
RET
READTEMP1: LCALL  I NI TDS1820
MOV     A, #0CCH
LCALL  WRI TEDS1820
MOV     R6, #34H
DJNZ   R6, $
MOV     A, #0BEH
LCALL  WRI TEDS1820
MOV     R6, #34H
DJNZ   R6, $
MOV     R5, #09H
MOV     R0, #TEMPHEAD
MOV     B, #00H
READTEMP2: LCALL  READDS1820
MOV     @R0, A
INC     R0
READTEMP21: LCALL  CRC8CAL
DJNZ   R5, READTEMP2
MOV     A, B
JNZ    READTEMPOUT
MOV     A, TEMPHEAD+0
MOV     TEMPL, A
MOV     A, TEMPHEAD+1
MOV     TEMPH, A
READTEMPOUT: RET
CONVTEMP:  MOV     A, TEMPH
ANL    A, #80H
JZ     TEMPC1
CLR    C

```

```

MOV     A, TEMPL
CPL     A
ADD     A, #01H
MOV     TEMPL, A
MOV     A, TEMPH
CPL     A
ADDC    A, #00H
MOV     TEMPH, A
MOV     TEMPHC, #0BH
SJMP    TEMPC11
TEMPC1: MOV     TEMPHC, #0AH
TEMPC11: MOV     A, TEMPHC
        SWAP    A
        MOV     TEMPHC, A
        MOV     A, TEMPL
        ANL     A, #0FH
        MOV     DPTR, #TEMPDOTTAB
        MOVC   A, @A+DPTR
        MOV     TEMPLC, A
        MOV     A, TEMPL
        ANL     A, #0FOH
        SWAP    A
        MOV     TEMPL, A
        MOV     A, TEMPH
        ANL     A, #0FH
        SWAP    A
        ORL     A, TEMPL
        LCALL   HEX2BCD1
        MOV     TEMPL, A
        ANL     A, #0FOH
        SWAP    A
        ORL     A, TEMPHC
        MOV     TEMPHC, A
        MOV     A, TEMPL
        ANL     A, #0FH
        SWAP    A
        ORL     A, TEMPLC
        MOV     TEMPLC, A
        MOV     A, R7
        JZ      TEMPC12
        ANL     A, #0FH
        SWAP    A
        MOV     R7, A
        MOV     A, TEMPHC

```

```

                ANL    A, #0FH
                ORL    A, R7
                MOV    TEMPHC, A
TEMPC12:       RET
TEMPDOTTAB:   DB  0H, 01H, 01H, 02H, 03H, 04H, 04H, 05H, 06H
                DB  06H, 07H, 08H, 08H, 09H, 09H
DISPBCD:      MOV    A, TEMPLC
                ANL    A, #0FH
                MOV    70H, A
                MOV    A, TEMPLC
                SWAP   A
                ANL    A, #0FH
                MOV    71H, A
                MOV    A, TEMPHC
                ANL    A, #0FH
                MOV    72H, A
                MOV    A, TEMPHC
                SWAP   A
                ANL    A, #0FH
                MOV    73H, A
                MOV    A, TEMPHC
                ANL    A, #0FH
                CJNE   A, #010H, DISPBCD0
                SJMP   DISPBCD2
DISPBCD0:     MOV    A, TEMPHC
                ANL    A, #0FH
                JNZ    DISPBCD2
                MOV    A, TEMPHC
                SWAP   A
                ANL    A, #0FH
                MOV    73H, #0AH
                MOV    72H, A
DISPBCD2:     RET
DISP1:        MOV    R1, #70H
                MOV    R5, #0F7H
PLAY:         MOV    P1, #0FFH
                MOV    A, R5
                MOV    P2, A
                MOV    A, @R1
                MOV    DPTR, #TAB
                MOVC   A, @A+DPTR
                ORL    A, #01H
                MOV    P1, A

```

```

MOV    A, R5
NOP
NOP
JB     ACC. 2, LOOP5
NOP
NOP
CLR    P1. 0
LOOP5: LCALL  DL1MS
INC    R1
MOV    A, R5
JNB    ACC. 1, ENDOUT
RR     A
MOV    R5, A

AJMP   PLAY
ENDOUT: MOV    P1, #0FFH
MOV    P2, #0FFH
RET

TAB:   DB     03h, 09fh, 025H, 0DH, 99H, 49H, 41H, 1FH, 01H, 09H
DL1MS: MOV    R6, #14H
DL1:   MOV    R7, #19H
DL2:   DJNZ   R7, DL2
        DJNZ   R6, DL1
RET

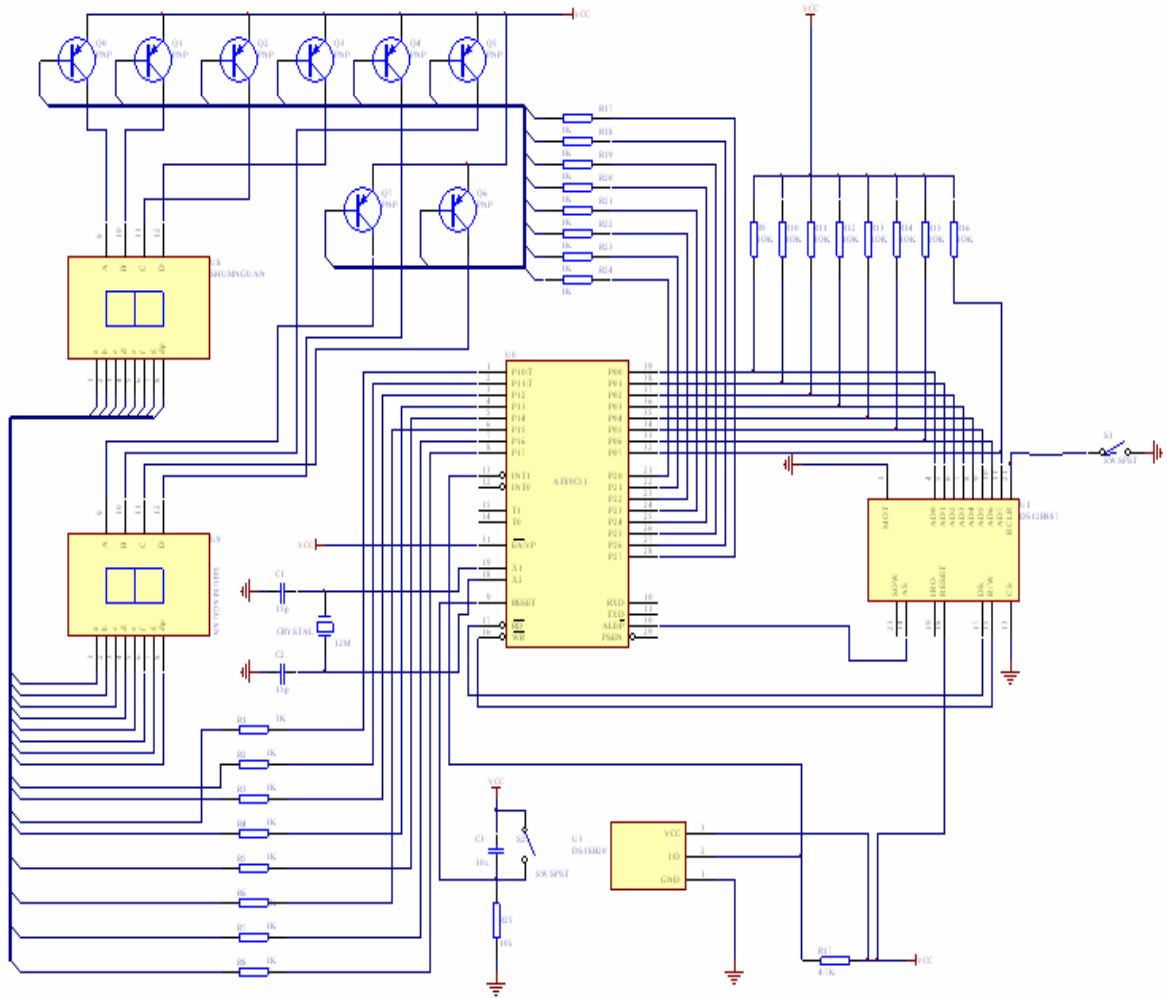
HEX2BCD1: MOV    B, #064H
        DIV   AB
        MOV   R7, A
        MOV   A, #0AH
        XCH  A, B
        DIV  AB
        SWAP A
        ORL  A, B
RET

CRC8CAL: PUSH  ACC
        MOV  R7, #08H
CRC8LOOP1: XRL  A, B
        RRC  A
        MOV  A, B
        JNC  CRC8LOOP2
        XRL  A, #18H
CRC8LOOP2: RRC  A
        MOV  B, A
        POP  ACC
        RR   A

```


PUSH ACC
 DJNZ R7, CRC8LOOP1
 POP ACC
 RET
 END

附录二：硬件电路总图



附图（1） 硬件电路总图

项目	选题	功能	方案	设计	结论	文献	格式	字数	总计
得分									
评阅人					日期				