数字温度传感器 DS1820(DS18B20)的应用

一、单线数字温度计 DSl820 介绍

DSl820 数字温度计提供 9 位(二进制)温度读数，指示器件的温度。信息经过单线接口送入 DSl820 或从 DSl820 送出，因此从主机 CPU 到 DSl820 仅需一条线(和地线)。DSl820 的电源可以由数据线本身提供而不需要外部电源。因为每一个 DSl820 在出厂时已经给定了唯一的序号，因此任意多个 DSl820 可以存放在同一条单线总线上。这允许在许多不同的地方放置温度敏感器件。DSl820 的测量范围从-55℃到+125℃，增量值为 0.5℃，可在 l s(典型值)内把温度变换成数字。

每一个 DSl820 包括一个唯一的 64 位长的序号，该序号值存放在 DSl820 内部的 ROM(只读存贮器)中。开始 8 位是产品类型编码(DSl820 编码均为 10H)。接着的 48 位是每个器件唯一的序号。最后 8 位是前面 56 位的 CRC(循环冗余校验)码。DSl820 中还有用于贮存测得的温度值的两个 8 位存贮器 RAM，编号为 0 号和 1 号。1 号存贮器存放温度值的符号，如果温度为负(℃)，则 1 号存贮器 8 位全为 1，否则全为 0。0 号存贮器用于存放温度值的补码，LSB(最低位)的“1”表示 0.5℃。**将存贮器中的二进制数求补再转换成十进制数并除以 2 就得到被测温度值(-550℃—125℃)**。DSl820 的引脚如图 2．26—l 所示。每只 D51820 都可以设置成两种供电方式，即数据总线供电方式和外部供电方式。采取数据总线供电方式可以节省一根导线，但完成温度测量的时间较长；采取外部供电方式则多用一根导线，但测量速度较快。

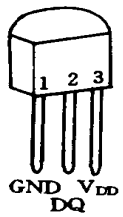

图 2．26-1　DS1820 的引脚
1．GND：地；
2．DQ：数字输入/输出
3．V_{DD}：可选的＋5V 电源

温度计算

1、 Ds1820 用 9 位存贮温值度，最高位为符号位，下图为 18b20 的温度存储方式，负温度 S=1，正温度 S=0。如：

00AAH 为+85℃,0032H 为 25℃，FF92H 为－55℃

TEMPERATURE REGISTER FORMAT Figure 2

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
LS Byte	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}
	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
MS Byte	S	S	S	S	S	S	S	S

2、Ds18b20 用 12 位存贮温值度，最高位为符号位，下图为 18b20 的温度存储方式，负温度 S=1，正温度 S=0。如：

0550H 为+85℃，0191H 为＋25.0625℃,FC90H 为-55℃

TEMPERATURE REGISTER FORMAT Figure 2

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
LS Byte	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
MS Byte	S	S	S	S	S	2^6	2^5	2^4

二、DSl820 工作过程及时序

DSl820 工作过程中的协议如下：

初始化：RoM 操作命令；存储器操作命令；处理数据。

1．初始化

单总线上的所有处理均从初始化开始。

2．ROM 操作品令

总线主机检测到 DSl820 的存在，便可以发出 ROM 操作命令之一，这些命令如

指令	代码
Read ROM(读 ROM)	[33H]
Match ROM(匹配 ROM)	[55H]
Skip ROM(跳过 ROM]	[CCH]
Search ROM(搜索 ROM)	[F0H]
Alarm search(告警搜索)	[ECH]

3．存储器操作命令

指令	代码
Write Scratchpad(写暂存存储器)	[4EH]
Read Scratchpad(读暂存存储器)	[BEH]
Copy Scratchpad(复制暂存存储器)	[48H]
Convert Temperature(温度变换)	[44H]
Recall EPROM(重新调出)	[B8H]
Read Power supply(读电源)	[B4H]

4．时　序

主机使用时间隙(time slots)来读写 DSl820 的数据位和写命令字的位

(1)初始化

时序见图 2.25-2。主机总线 to 时刻发送一复位脉冲(最短为 480us 的低电平信号)，接着在 tl 时刻释放总线并进入接收状态，DSl820 在检测到总线的上升沿之后，等待 15-60us，接着 DS1820 在 t2 时刻发出存在脉冲(低电平，持续 60-240 us)，如图中虚线所示。

以下子程序在 MCS51 仿真机上通过，其晶振为 12M.。初始化子程序：

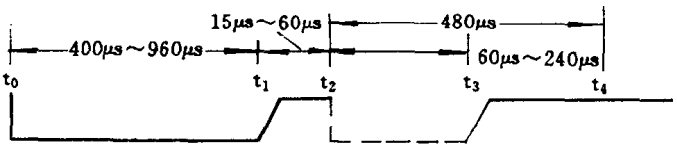

图 2.25-2　初始化时序

RESET：

```
        PUSH B          ;保存 B 寄存器
        PUSH A          保存 A 寄存器
        MOV A,#4        ;设置循环次数
        CLR P1.0        ;发出复位脉冲
        MOV B,#250      ;计数 250 次
        DJNZ B,$        ;保持低电平 500us
        SETB Pl.0       ;释放总线
        MOV B,#6        ;设置时间常数
        CLR C           ;清存在信号标志
WAITL:  JB Pl.0,WH      ;若总线释放，跳出循环
        DJNZ B,WAITL    ;总线低，等待
        DJNZ ACC,WAITL;释放总线等待一段时间
        SJMP SHORT
WH:     MOV B,#111
WH1:    ORL C,P1.0
        DJNZ B,WH1      ;存在时间等待
SHORT:  POP A
        POP B
        RET
```

(2)写时间隙

当主机总线 t o 时刻从高拉至低电平时，就产生写时间隙，见图 2．25—3、图 2．25—4，从 to 时刻开始 15us 之内应将所需写的位送到总线上，DSl820 在 t。后 15-60us 间对总线采样。若低电平，写入的位是 0，见图 2．25—3；若高电平，写入的位是 1，见图 2．25—4。连续写 2 位间的间隙应大于 1us。

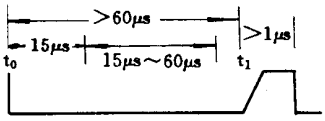

图 2.25-3　写 0 时序

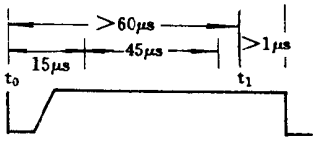

图 2.25-4　写 1 时序

写位子程序(待写位的内容在 C 中)：

```
WRBIT:
        PUSH B          ;保存 B
        MOV B,#28       ;设置时间常数
        CLR P1.0        ;写开始
        NOP             ;1us
        NOP             ;1us
        NOP             ;1us
        NOP             ;1us
        N0P             ;1us
        MOVPl.0,C       ;C 内容到总线
WDLT:   DJNZ B,WDLT;等待 56Us
        POP B
```

```
        SETB Pl.0      ;释放总线
        RET            ;返回
```

写字节子程序(待写内容在 A 中):

```
WRBYTB:
        PUSH B         :保存 B
        MOV B，#8H     ;设置写位个数
WLOP:   RRC A          ;把写的位放到 C
        ACALL WRBIT    ;调写 1 位子程序
        DJNZ B，WLOP   ;8 位全写完?
        POP B
        RET
```

(3)读时间隙

见图 2．25—5，主机总线 to 时刻从高拉至低电平时，总线只须保持低电平 l 7ts。之后在 t1 时刻将总线拉高，产生读时间隙，读时间隙在 t1 时刻后 t 2 时刻前有效。t z 距 to 为 15 捍 s，也就是说，t z 时刻前主机必须完成读位，并在 t o 后的 60 尸 s 一 120 fzs 内释放总线。读位子程序(读得的位到 C 中)：

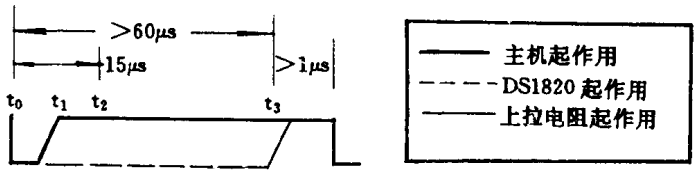

图 2.25-5　读时序

```
RDBIT:
        PUSH B         ;保存 B
        PUSH A         ;保存 A
        MOV B,#23      ;设置时间常数
        CLR P1.0       ;读开始，图 2．25—5 的 t0 时刻
        NOP            ;1us
        NOP            ;1us
        NOP            ;1us
        NOP            ;1us
        SETB Pl.0      ;释放总线
        MOV A,P1       ;P1 口读到 A
        MOV C,EOH      ;P1.0 内容 C
        NOP            ;1us
        NOP            ;1us
        NOP            ;1us
        NOP            ;1us
RDDLT:  DJNZ B,RDDLT   ;等待 46us
        SETB P1.0
        POP A
```

```
        POP B
        RET
```
读字节子程序(读到内容放到 A 中)；
```
RDBYTE:
        PUSH B          ;保存 B
RLOP:   MOV B,#8H       ;设置读位数
        ACALL RDBIT     ;调读 1 位子程序
        RRC A           ;把读到位在 C 中并依次送给 A
        DJNZ B,RLOP     ;8 位读完?
        POP B           ;恢复 B
        RET
```

三、多路测量

每一片 DSl820 在其 ROM 中都存有其唯一的 48 位序列号，在出厂前已写入片内 ROM 中，主机在进入操作程序前必须逐一接入 1820 用读 ROM(33H)命令将该 l 820 的序列号读出并登录。

当主机需要对众多在线 1820 的某一个进行操作时，首先要发出匹配 ROM 命令(55H)，紧接着主机提供 64 位序列(包括该 1820 的 48 位序列号)，之后的操作就是针对该 1820 的。而所谓跳过 ROM 命令即为：之后的操作是对所有 1820 的。框图中先有跳过 ROM，即是启动所有 1820 进行温度变换，之后，通过匹配 ROM，再逐一地读回每个 1820 的温度数据。

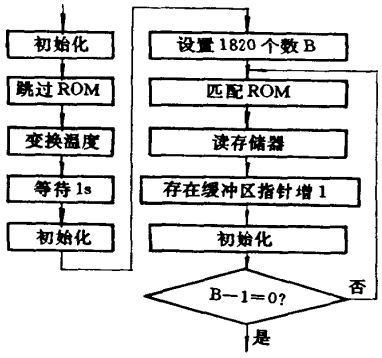

图 2. 25-6　多路测温程序框图

在 1820 组成的测温系统中，主机在发出跳过 ROM 命令之后，再发出统一的温度转换启动码 44H，就可以实现所有 1820 的统一转换，再经过 1s 后，就可以用很少的时间去逐一读取。这种方式使其 T 值往往小于传统方式（由于采取公用的放大电路和 A／D 转换器，只能逐一转换。)，显然通道数越多，这种省时效应就越明显。

四、实际应用

1、ds1820 序列号获得
```
;|-------------------------------------------------------------------------|
;|          读出 ds1820 序列号应用程序,P1.6 接 ds1820                      |
;|-------------------------------------------------------------------------|
        ORG 0000H
        AJMP MAIN
        ORG 0020H
MAIN:
        MOV SP,#60H
        CLR EA              ;使用 ds1820 一定要禁止任何中断产生
        LCALL INT           ;初始化 ds1820
        MOV A,#33H
```

```
        LCALL WRITE       ;送入读 ds1820 的 ROM 命令
        LCALL READ        ;开始读出当前 ds1820 序列号
        MOV 40H,A
        LCALL READ
        MOV 41H,A
        LCALL READ
        MOV 42H,A
        LCALL READ
        MOV 43H,A
        LCALL READ
        MOV 44H,A
        LCALL READ
        MOV 45H,A
        LCALL READ
        MOV 46H,A
        LCALL READ
        MOV 47H,A
        SETB EA
        SJMP $

INT:                      ;初始化 ds1820 子程序
        CLR EA
    L0:CLR P1.6           ;ds1820 总线为低复位电平
        MOV R2,#200
    L1:CLR P1.6
        DJNZ R2,L1        ;总线复位电平保持 400us
        SETB P1.6         ;释放 ds1820 总线
        MOV R2,#30
    L4:DJNZ R2,L4         ;释放 ds1820 总线保持 60us
        CLR C             ;清存在信号
        ORL C,P1.6
        JC L0             ;存在吗?不存在则重新来
        MOV R6,#80
    L5:ORL C,P1.6
        JC L3
        DJNZ R6,L5
        SJMP L0
    L3:MOV R2,#240
    L2:DJNZ R2,L2
        RET

WRITE:                    ;向 ds1820 写操作命令子程序
        CLR EA
        MOV R3,#8         ;写入 ds1820 的 bit 数,一个字节 8 个 bit
```

```
    WR1:SETB P1.6
        MOV R4,#8
        RRC A                ;把一个字节 data(A)分成 8 个 bit 环移给  C
        CLR P1.6             ;开始写入 ds1820 总线要处于复位(低)状态
    WR2:DJNZ R4,WR2          ;ds1820 总线复位保持 16us
        MOV P1.6,C           ;写入一个 bit
        MOV R4,#20
    WR3:DJNZ R4,WR3          ;等待 40us
        DJNZ R3,WR1          ;写入下一个 bit
        SETB P1.6            ;重新释放 ds1820 总线
        RET

READ:
        CLR EA
        MOV R6,#8            ;连续读 8 个 bit
    RE1:CLR P1.6             ;读前总线保持为低
        MOV R4,#4
        NOP
        SETB P1.6            ;开始读，总线释放
    RE2:DJNZ R4,RE2          ;持续 8us
        MOV C,P1.6           ;从 ds1820 总线读得一个 bit
        RRC A                ;把读得的位值环移给  A
        MOV R5,#30
    RE3:DJNZ R5,RE3          ;持续 60us
        DJNZ R6,RE1          ;读下一个 bit
        SETB P1.6            ;重新释放 ds1820 总线
        RET

        END
```

2、温度转换和读取

```
;|-------------------------------------------------------------------------------------|
;|          获取单个 ds1820 转化的温度值的应用程序,P1.6 接 ds1820                      |
;|-------------------------------------------------------------------------------------|
        ORG 0000H
        AJMP MAIN
        ORG 0020H
MAIN:
        MOV SP,#60H
        LCALL GET_TEMP
        SJMP $

GET_TEMP:
        CLR PSW.4
```

```
        SETB PSW.3          ;设置工作寄存器当前所在的区域
        CLR EA              ;使用 ds1820 一定要禁止任何中断产生
        LCALL INT           ;调用初使化子程序
        MOV A,#0CCH
        LCALL WRITE         ;送入跳过 ROM 命令
        MOV A, #44H
        LCALL WRITE         ;送入温度转换命令
        LCALL INT           ;温度转换完全,再次初使化 ds1820
        MOV A,#0CCH
        LCALL WRITE         ;送入跳过 ROM 命令
        MOV A,#0BEH
        LCALL WRITE         ;送入读温度暂存器命令
        LCALL READ
        MOV R7,A            ;读出温度值低字节存入 R7
        LCALL READ
        MOV R6,A            ;读出谩度值高字节存入 R6
        SETB EA
        RET

INT:                        ;初始化 ds1820 子程序
        CLR EA
     L0:CLR P1.6            ;ds1820 总线为低复位电平
        MOV R2,#200
     L1:CLR P1.6
        DJNZ R2,L1          ;总线复位电平保持 400us
        SETB P1.6           ;释放 ds1820 总线
        MOV R2,#30
     L4:DJNZ R2,L4          ;释放 ds1820 总线保持 60us
        CLR C               ;清存在信号
        ORL C,P1.6
        JC L0               ;存在吗?不存在则重新来
        MOV R6,#80
     L5:ORL C,P1.6
        JC L3
        DJNZ R6,L5
        SJMP L0
     L3:MOV R2,#240
     L2:DJNZ R2,L2
        RET

WRITE:                      ;向 ds1820 写操作命令子程序
        CLR EA
        MOV R3,#8           ;写入 ds1820 的 bit 数,一个字节 8 个 bit
    WR1:SETB P1.6
```

```
        MOV R4,#8
        RRC A               ;把一个字节 data(A)分成 8 个 bit 环移给  C
        CLR P1.6            ;开始写入 ds1820 总线要处于复位(低)状态
    WR2:DJNZ R4,WR2         ;ds1820 总线复位保持 16us
        MOV P1.6,C          ;写入一个 bit
        MOV R4,#20
    WR3:DJNZ R4,WR3         ;等待 40us
        DJNZ R3,WR1         ;写入下一个 bit
        SETB P1.6           ;重新释放 ds1820 总线
        RET

READ:
        CLR EA
        MOV R6,#8           ;连续读 8 个 bit
    RE1:CLR P1.6            ;读前总线保持为低
        MOV R4,#4
        NOP
        SETB P1.6           ;开始读，总线释放
    RE2:DJNZ R4,RE2         ;持续 8us
        MOV C,P1.6          ;从 ds1820 总线读得一个 bit
        RRC A               ;把读得的位值环移给  A
        MOV R5,#30
    RE3:DJNZ R5,RE3         ;持续 60us
        DJNZ R6,RE1         ;读下一个 bit
        SETB P1.6           ;重新释放 ds1820 总线
        RET

        END
```