实时时钟电路 DS1302 的原理及应用

摘 要：介绍美国 DALLAS 公司推出的具有涓细电流充电能力的低功耗实时时钟电路 DS1302 的结构、工作原理及其在实时显示时间中的应用。它可以对年、月、日、周日、时、分、秒进行计时，且具有闰年补偿等多种功能。给出 DS1302 在读写中的 C51 程序及流程图，以及在调试过程中的注意事项。

关键词：时钟电路；实时时钟；单片机；应用

1 引言　现在流行的串行时钟电路很多，如 DS1302、 DS1307、PCF8485 等。这些电路的接口简单、价格低廉、使用方便，被广泛地采用。本文介绍的实时时钟电路 DS1302 是 DALLAS 公司的一种具有涓细电流充电能力的电路，主要特点是采用串行数据传输，可为掉电保护电源提供可编程的充电功能，并且可以关闭充电功能。采用普通 32.768kHz 晶振。

2 DS1302 的结构及工作原理

DS1302 是美国 DALLAS 公司推出的一种高性能、低功耗、带 RAM 的实时时钟电路，它可以对年、月、日、周日、时、分、秒进行计时，具有闰年补偿功能，工作电压为 2.5V～5.5V。采用三线接口与 CPU 进行同步通信，并可采用突发方式一次传送多个字节的时钟信号或 RAM 数据。DS1302 内部有一个 31×8 的用于临时性存放数据的 RAM 寄存器。DS1302 是 DS1202 的升级产品，与 DS1202 兼容，但增加了主电源/后背电源双电源引脚，同时提供了对后背电源进行涓细电流充电的能力。

2.1 引脚功能及结构

图 1 示出 DS1302 的引脚排列,其中 V_{cc1} 为后备电源，V_{CC2} 为主电源。在主电源关闭的情况下，也能保持时钟的连续运行。DS1302 由 V_{cc1} 或 V_{cc2} 两者中的较大者供电。当 V_{cc2} 大于 V_{cc1}＋0.2V 时，V_{cc2} 给 DS1302 供电。当 V_{cc2} 小于 V_{cc1} 时，DS1302 由 V_{cc1} 供电。X1 和 X2 是振荡源，外接 32.768kHz 晶振。RST 是复位/片选线，通过把 RST 输入驱动置高电平来启动所有的数据传送。RST 输入有两种功能：首先，RST 接通控制逻辑，允许地址/命令序列送入移位寄存器；其次，RST 提供终止单字节或多字节数据的传送手段。当 RST 为高电平时，所有的数据传送被初始化，允许对 DS1302 进行操作。如果在传送过程中 RST 置为低电平，则会终止此次数据传送，I/O 引脚变为高阻态。上电运行时，在 V_{cc}≥2.5V 之前，RST 必须保持低电平。只有在 SCLK 为低电平时，才能将 RST 置为高电平。I/O 为串行数据输入输出端(双向)，后面有详细说明。SCLK 始终是输入端。

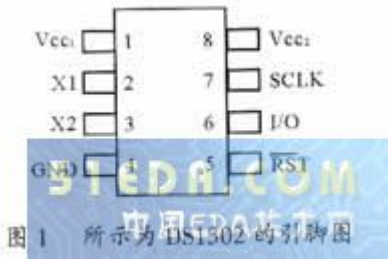

图 1　所示为 DS1302 的引脚图

2.2 DS1302 的控制字节

DS1302 的控制字如图 2 所示。控制字节的最高有效位(位 7)必须是逻辑 1，如果它为 0，则不能把数据写入 DS1302 中，位 6 如果为 0，则表示存取日历时钟数据，为 1 表示存取 RAM 数据;位 5 至位 1 指示操作单元的地址;最低有效位(位 0)如为 0 表示要进行写操作，为 1 表示进行读操作，控制字节总是从最低位开始输出。

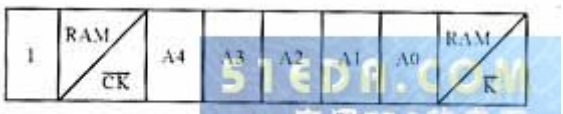

图 2 DS1302 的控制字节

2.3 数据输入输出(I/O)

在控制指令字输入后的下一个 SCLK 时钟的上升沿时，数据被写入 DS1302，数据输入从低位即位 0 开始。同样，在紧跟 8 位的控制指令字后的下一个 SCLK 脉冲的下降沿读出 DS1302 的数据，读出数据时从低位 0 位到高位 7。

2.4 DS1302 的寄存器

DS1302 有 12 个寄存器，其中有 7 个寄存器与日历、时钟相关，存放的数据位为 BCD 码形式,其日历、时间寄存器及其控制字见表 1。

表 1 日历、时间寄存器及其控制字

寄存器名称	命令字		取值范围	各位内容							
	写操作	读操作		7	6	5	4	3	2	1	0
秒寄存器	80H	81H	00～59	CH	10SEC			SEC			
分寄存器	82H	83H	00～59	0	10MIN			MIN			
时寄存器	84H	85H	01～12 或 00～23	12/24	0	10 HR		HR			
日寄存器	86H	87H	01～28,29, 30, 31	0	0	10DATE		DATE			
月寄存器	88H	89H	01～12	0	0	0	10M	MONTH			
周寄存器	8AH	8BH	01～07	0	0	0	0	0	DAY		
年寄存器	8CH	8DH	00～99	10YEAR				YEAR			

此外，DS1302 还有年份寄存器、控制寄存器、充电寄存器、时钟突发寄存器及与 RAM 相关的寄存器等。时钟突发寄存器可一次性顺序读写除充电寄存器外的所有寄存器内容。 DS1302 与 RAM 相关的寄存器分为两类：一类是单个 RAM 单元，共 31 个，每个单元组态为一个 8 位的字节，其命令控制字为 C0H～FDH，其中奇数为读操作，偶数为写操作；另一类为突发方式下的 RAM 寄存器，此方式下可一次性读写所有的 RAM 的 31 个字节，命令控制字为 FEH(写)、FFH(读)。

3 DS1302 实时显示时间的软硬件

DS1302 与 CPU 的连接需要三条线，即 SCLK(7)、I/O(6)、RST(5)。图 3 示出 DS1302 与 89C2051 的连接图，其中，时钟的显示用 LCD。

3.1 DS1302 与 CPU 的连接

实际上，在调试程序时可以不加电容器，只加一个 32.768kHz 的晶振即可。只是选择晶振时，不同的晶振，误差也较大。另外，还可以在上面的电路中加入 DS18B20，同时显示实时温度。只要占用 CPU 一个口线即可。 LCD 还可以换成 LED，还可以使用北京卫信杰科技发展有限公司生产的 10 位多功能 8 段液晶显示模块 LCM101，内含看门狗(WDT)/时钟发生器及两种频率的蜂鸣器驱动电路，并有内置显示 RAM，可显示任意字段笔划，具有 3－4 线串行接口，可与任何单片机、IC 接口。功耗低，显示状态时电流为 2μA (典型值)，省电模式时小于 1μA，工作电压为 2.4V～3.3V，显示清晰。

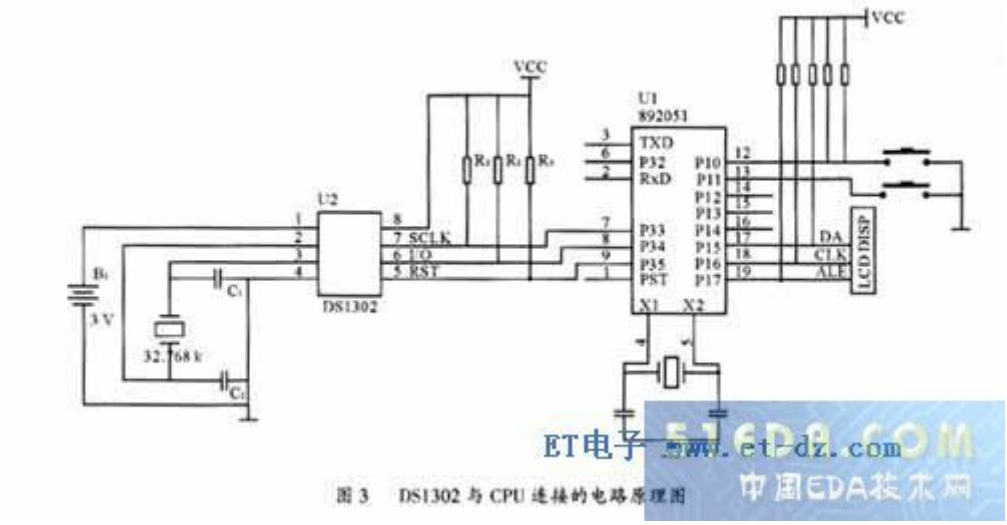

图 3 DS1302 与 CPU 连接的电路原理图

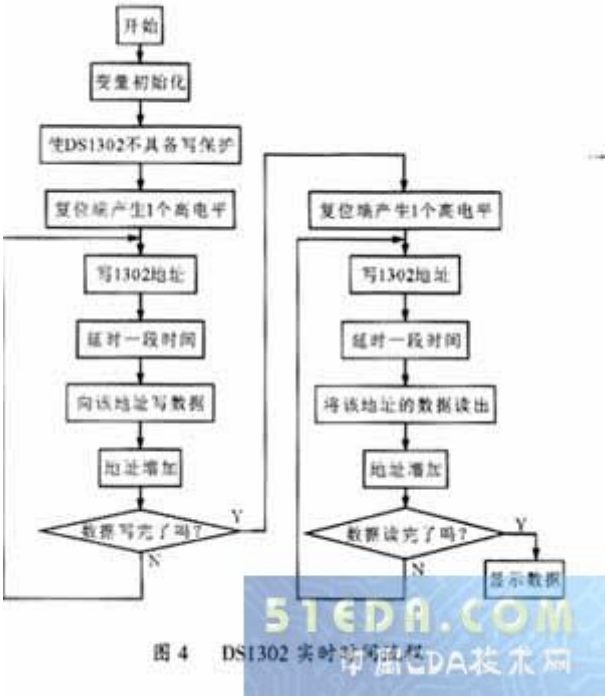

图 4 DS1302 实时时间流程

3.2 DS1302 实时时间流程

图 4 示出 DS1302 的实时时间流程。根据此流程框图，不难采集实时时间。下面结合流程图对 DS1302 的基本操作进行编程：

```
#include "Intrins.h"
sbit t_clk = P3^3
sbit t_jo = P3^4
sbit t_rst = P3^5
sbit   BIT7 = ACC^7
sbit BIT0 = ACC^0
void inputbyte(unsigned char ucda)//8 位数据写
入函数
{unsigned char i;
ACC = ucda; 将要写入的数放入 ACC
t_rst = 1; 启动数据传送
for(i = 8; i> 0; i--);循环 8 次,写入 8 位数据,
                    ;从低位到高位
{t_jo = BIT0; 将 ACC^0 的值赋给时钟数据线
t_clk = 0
t_clk = 1; 在时钟线的上升沿写入 1 位数据
ACC = ACC≫1; 将高 1 位数据移至 ACC^0
}
}
unsigned char outputbyte(void)//8 位数据读出
函数
{unsigned char i;
t_rst = 1; 启动数据传送
for(i = 8; i> 0; i--); 读出 8 位数据,从低位到高位
{ACC = ACC≫1; 将前一下降沿读出的数据右移
1 位,从而该次读出的数放入 ACC^7
t_jo = 1; P1 口输入之前置 1
t_clk = 1
t_clk = 0; 时钟线下降沿读出 1 位数据
BIT7 = t_jo; can not use P1^7 = t_jo for P1^7 not
            ;a varient
}
return(ACC)
}
```

```
void wr_1302(unsigned char add, unsigned char
ucda)//将指令或数据写入对应寄存器
{t_rst = 0
t_clk = 0
t_rst = 1
inputbyte(add)
//delay15(1)
inputbyte(ucda)
t_rst = 0
t_io = 1
}
unsigned char re_1302(unsigned char add)//读出
对应寄存器内容
{unsigned char ucda
t_rst = 0
t_clk = 0
t_rst = 1
inputbyte(add)
//delay15(1)
ucda = outputbyte()
t_rst = 0
return(ucda)
}
void set1302(unsigned char *pda)//设置时间
初值
```

```
{unsigned char i
unsigned char add = 0x80
wr_1302(0x8e, 0x00); 将控制寄存器值设为零,
                     ;最高位 WP = 0 允许写
for(i = 7; i> 0; i - - ); 将七个时间初值写入对
                     ;应寄存器
{wr_1302(add, * pda); 写对应时钟寄存器的值
pda + +
add + = 2;
}
wr_1302(0x8e, 0x80); 写保护,防止干扰影响时
                     ;间值
}
void get_1302(unsigned char curtime[ ]) //读取
当前时间值
{unsigned char i, j
unsigned char add = 0x81
bdata unsigned char sec
for(i = 0; i <7; i + + )
  {curtime[i] = re_1302(add); 读对应时钟寄
                              存器的值
  sec = curtime[i]
  j = sec≫4; 将 BCD 码转化成对应十进制数
  j* = 10
  sec = sec&0x0f
  sec + = j
  curtime[i] = sec
  add + = 2
  }
```


根据本人在调试中遇到的问题，特作如下说明：

DS1302 与微处理器进行数据交换时，首先由微处理器向电路发送命令字节，命令字节最高位 MSB(D7)必须为逻辑 1，如果 D7=0，则禁止写 DS1302，即写保护；D6=0，指定时钟数据，D6=1，指定 RAM 数据；D5～D1 指定输入或输出的特定寄存器；最低位 LSB(D0)为逻辑 0，指定写操作(输入)， D0=1，指定读操作(输出)。

在 DS1302 的时钟日历或 RAM 进行数据传送时，DS1302 必须首先发送命令字节。若进行单字节传送，8 位命令字节传送结束之后，在下 2 个 SCLK 周期的上升沿输入数据字节，或在下 8 个 SCLK 周期的下降沿输出数据字节。

DS1302 与 RAM 相关的寄存器分为两类:一类是单个 RAM 单元，共 31 个，每个单元组态为一个 8 位的

字节，其命令控制字为 C0H～FDH，其中奇数为读操作，偶数为写操作；再一类为突发方式下的 RAM 寄存器，在此方式下可一次性读、写所有的 RAM 的 31 个字节。

要特别说明的是备用电源 B1，可以用电池或者超级电容器(0.1F 以上)。虽然 DS1302 在主电源掉电后的耗电很小，但是，如果要长时间保证时钟正常，最好选用小型充电电池。可以用老式电脑主板上的 3.6V 充电电池。如果断电时间较短(几小时或几天)时，就可以用漏电较小的普通电解电容器代替。100 μF 就可以保证 1 小时的正常走时。DS1302 在第一次加电后，必须进行初始化操作。初始化后就可以按正常方法调整时间。

4 结论

DS1302 存在时钟精度不高，易受环境影响，出现时钟混乱等缺点。DS1302 可以用于数据记录，特别是对某些具有特殊意义的数据点的记录，能实现数据与出现该数据的时间同时记录。这种记录对长时间的连续测控系统结果的分析及对异常数据出现的原因的查找具有重要意义。传统的数据记录方式是隔时采样或定时采样，没有具体的时间记录，因此，只能记录数据而无法准确记录其出现的时间；若采用单片机计时，一方面需要采用计数器，占用硬件资源，另一方面需要设置中断、查询等，同样耗费单片机的资源，而且，某些测控系统可能不允许。但是，如果在系统中采用时钟芯片 DS1302，则能很好地解决这个问题。