

## 1 DS1302 简介:

DS1302是美国 DALLAS 公司推出的一种高性能、低功耗的实时时钟芯片，附加31字节静态 RAM，采用 SPI 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和 RAM 数据。实时时钟可提供秒、分、时、日、星期、月和年，一个月小与31天时可以自动调整，且具有闰年补偿功能。工作电压宽达2.5~5.5V。采用双电源供电（主电源和备用电源），可设置备用电源充电方式，提供了对后背电源进行涓细电流充电的能力。DS1302的外部引脚分配如图1所示及内部结构如图2所示。DS1302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上，能实现数据与出现该数据的时间同时记录，因此广泛应用于测量系统中。

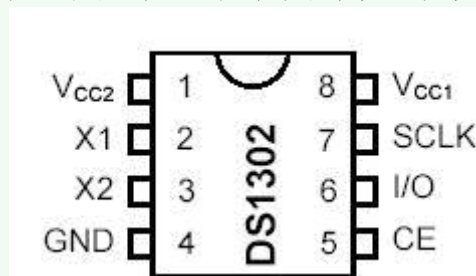


图1 DS1302的外部引脚分配

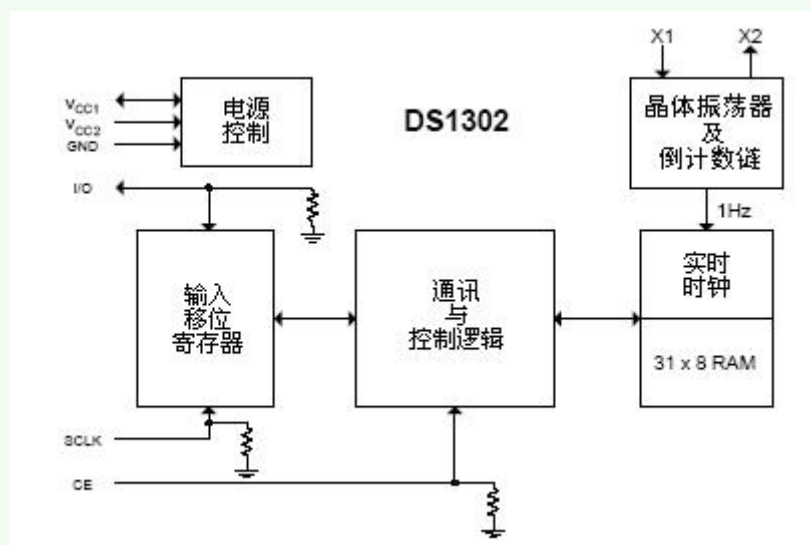


图2 DS1302的内部结构

各引脚的功能为:

**Vcc1:** 主电源; **Vcc2:** 备份电源。当  $V_{cc2} > V_{cc1} + 0.2V$  时, 由  $V_{cc2}$  向 DS1302 供电, 当  $V_{cc2} < V_{cc1}$  时, 由  $V_{cc1}$  向 DS1302 供电。

**SCLK:** 串行时钟, 输入, 控制数据的输入与输出;

**I/O:** 三线接口时的双向数据线;

**CE:** 输入信号, 在读、写数据期间, 必须为高。该引脚有两个功能: 第一, CE 开始控制字访问移位寄存器的控制逻辑; 其次,

CE 提供结束单字节或多字节数据传输的方法。

DS1302有下列几组寄存器：

① DS1302有关日历、时间的寄存器共有12个，其中有7个寄存器（读时81h～8Dh，写时80h～8Ch），存放的数据格式为BCD码形式，如图3所示。

读寄存器	写寄存器	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	范围
81h	80h	CH	10 秒			秒				00-59
83h	82h		10分			分				00-59
85h	84h	12/24	0	10 AM/PM	时	时				1-12/0-23
87h	86h	0	0	10 日		日				1-31
89h	88h	0	0	0	10 月	月				1-12
8Bh	8Ah	0	0	0	0	0	周日			1-7
8Dh	8Ch	10 年				年				00-99
8Fh	8Eh	WP	0	0	0	0	0	0	0	—

图 3 DS1302有关日历、时间的寄存器

小时寄存器（85h、84h）的位7用于定义DS1302是运行于12小时模式还是24小时模式。当为高时，选择12小时模式。在12小时模式时，位5是，当为1时，表示PM。在24小时模式时，位5是第二个10小时位。

秒寄存器（81h、80h）的位7定义为时钟暂停标志（CH）。当该位置为1时，时钟振荡器停止，DS1302处于低功耗状态；当该位置为0时，时钟开始运行。

控制寄存器（8Fh、8Eh）的位7是写保护位（WP），其它7位均置为0。在任何的对时钟和RAM的写操作之前，WP位必须为0。当WP位为1时，写保护位防止对任一寄存器的写操作。

②DS1302有关RAM的地址

DS1302中附加31字节静态RAM的地址如图4所示。

读地址	写地址	数据范围
C1h	C0h	00-FFh
C3h	C2h	00-FFh
C5h	C4h	00-FFh
·	·	·
·	·	·
·	·	·
FDh	FCh	00-FFh

图4

③ DS1302的工作模式寄存器

所谓突发模式是指一次传送多个字节的时钟信号和RAM数据。突发模式寄存器如图5所示。

工作模式寄存器		读寄存器	写寄存器
时钟突发模式寄存器	CLOCK BURST	BFh	BEh
RAM 突发模式寄存器	RAM BURST	FFh	FEh

图5

④此外，DS1302还有充电寄存器等。

## 2 读写时序说明

DS1302是 SPI 总线驱动方式。它不仅要向寄存器写入控制字，还需要读取相应寄存器的数据。

要想与 DS1302通信，首先要先了解 DS1302的控制字。DS1302的控制字如图6。

7	6	5	4	3	2	1	0
1	RAM $\overline{CK}$	A4	A3	A2	A1	A0	RD $\overline{WR}$

图6 控制字（即地址及命令字节）

控制字的最高有效位（位7）必须是逻辑1，如果它为0，则不能把数据写入到 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（A4~A0）：指示操作单元的地址；

位0（最低有效位）：如为0，表示要进行写操作，为1表示进行读操作。

控制字总是从**最低位开始输出**。在控制字指令输入后的下一个 SCLK 时钟的**上升沿**时，数据被**写入** DS1302，数据输入从最低位（0位）开始。同样，在紧跟8位的控制字指令后的下一个 SCLK 脉冲的**下降沿**，**读出** DS1302的数据，读出的数据也是从最低位到最高位。数据读写时序如图7。

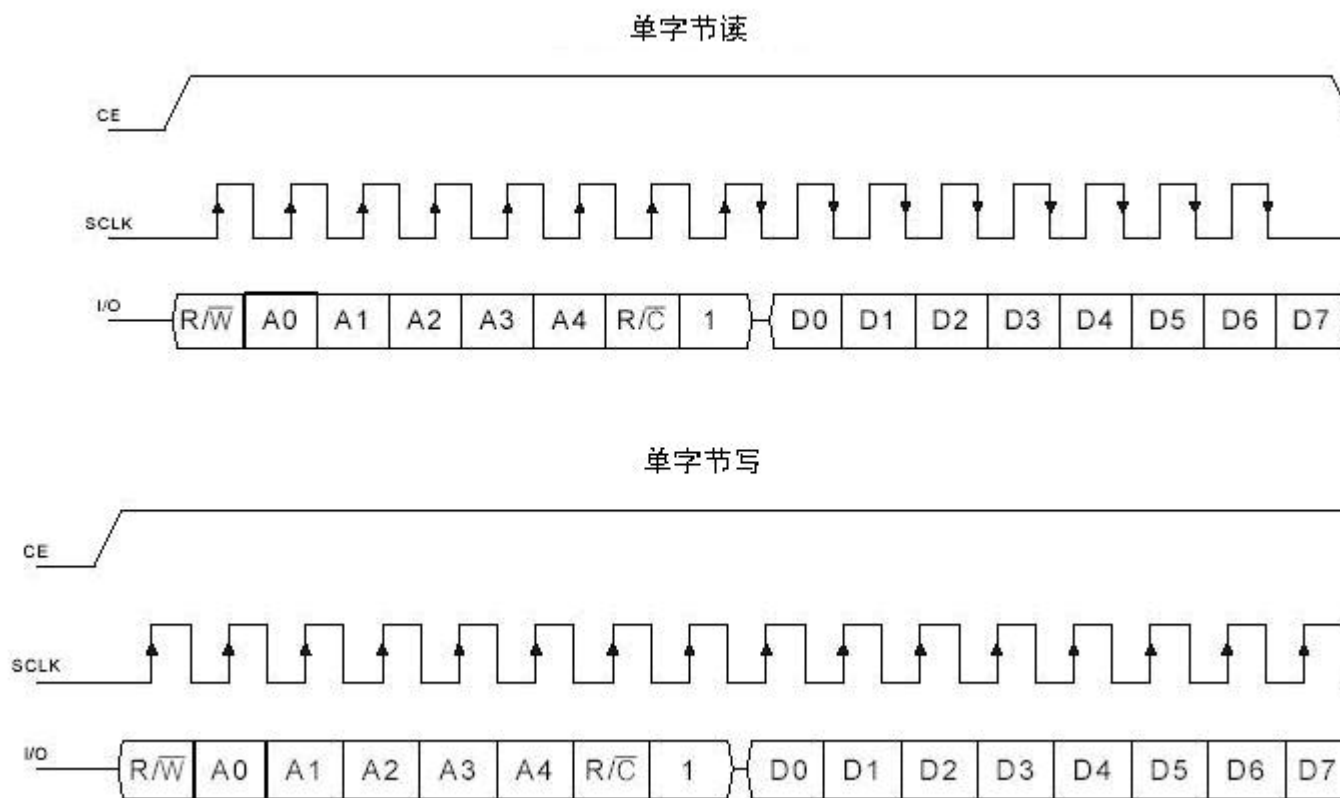


图7 数据读写时序

具体操作见驱动程序。

### 3 电路原理图:

电路原理图如图8, DS1302与单片机的连接也仅需要3条线: CE 引脚、SCLK 串行时钟引脚、I/O 串行数据引脚,  $V_{cc2}$ 为备用电源, 外接 32.768kHz 晶振, 为芯片提供计时脉冲。

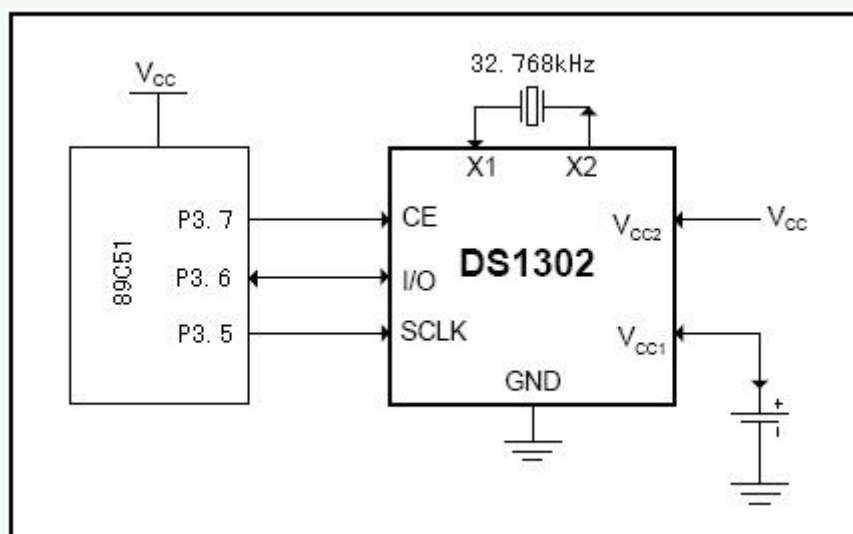


图8

## 4 驱动程序

//寄存器宏定义

```
#define WRITE_SECOND          0x80
#define WRITE_MINUTE          0x82
#define WRITE_HOUR            0x84
#define READ_SECOND           0x81
#define READ_MINUTE           0x83
#define READ_HOUR             0x85
#define WRITE_PROTECT         0x8E
```

//位寻址寄存器定义

sbit ACC\_7 = ACC^7;

//管脚定义

sbit SCLK = P3^5; // DS1302时钟信号

7脚

sbit DIO= P3^6; // DS1302数据信号

6脚

sbit CE = P3^7; // DS1302片选

5脚

//地址、数据发送子程序

void Write1302 ( unsigned char addr,dat )

{

unsigned char i,temp;

CE=0; //CE 引脚为低,数据传送

中止

SCLK=0; //清零时钟总线

CE = 1; //CE 引脚为高,逻辑控制

有效

//发送地址

for ( i=8; i>0; i-- ) //循环8次移位

{

SCLK = 0;

temp = addr;

DIO = (bit)(temp&0x01); //每次传输低字

节

addr >>= 1; //右移一位

SCLK = 1;

}

```

//发送数据
    for ( i=8; i>0; i-- )
    {
        SCLK = 0;
        temp = dat;
        DIO = (bit)(temp&0x01);
        dat >>= 1;
        SCLK = 1;
    }
    CE = 0;
}

//数据读取子程序
unsigned char Read1302 ( unsigned char addr )
{
    unsigned char i,temp,dat1,dat2;
    CE=0;
    SCLK=0;
    CE = 1;
    //发送地址
    for ( i=8; i>0; i-- )                                //循环8次移位
    {
        SCLK = 0;
        temp = addr;
        DIO = (bit)(temp&0x01);                            //每次传输低字
        addr >>= 1;                                          //右
        SCLK = 1;
    }
    //读取数据
    for ( i=8; i>0; i-- )
    {
        ACC_7=DIO;
        SCLK = 1;
        ACC>>=1;
        SCLK = 0;
    }
    CE=0;
    dat1=ACC;
    dat2=dat1/16;                                          //数据进制转换
}

```

```

        dat1=dat1%16;                                //十六进制转十
进制
        dat1=dat1+dat2*10;
        return (dat1);
    }

//初始化 DS1302
void Initial(void)
{
    Write1302 (WRITE_PROTECT,0X00);                //禁止写保
护
    Write1302 (WRITE_SECOND,0x56);                //秒位初始
化
    Write1302 (WRITE_MINUTE,0x34);                //分钟初始化
    Write1302 (WRITE_HOUR,0x12);                //
小时初始化
    Write1302 (WRITE_PROTECT,0x80);                //
允许写保护
}

```

## 51单片机综合学习系统之 DS1302时钟应用篇 《电子制作》2008年10月 站长原创，如需引用请注明出处

大家好，通过以前的学习，我们已经对51单片机综合学习系统的使用方法及学习方式有所了解与熟悉，学会了使用 AD 模数转换的基本知识，体会到了综合学习系统的易用性与易学性，这一期我们将一起学习 DS1302时钟的基本原理与应用实例。

先看一下我们将要使用的51单片机综合学习系统能完成哪些实验与产品开发工作：分别有流水灯，数码管显示，液晶显示，按键开关，蜂鸣器奏乐，继电器控制，IIC 总线，SPI 总线，PS/2实验，AD 模数转换，光耦实验，串口通信，红外线遥控，无线遥控，温度传感，步进电机控制等等。主体系统如图1所示，其配套书本教程《单片机快速入门》如图2所示。



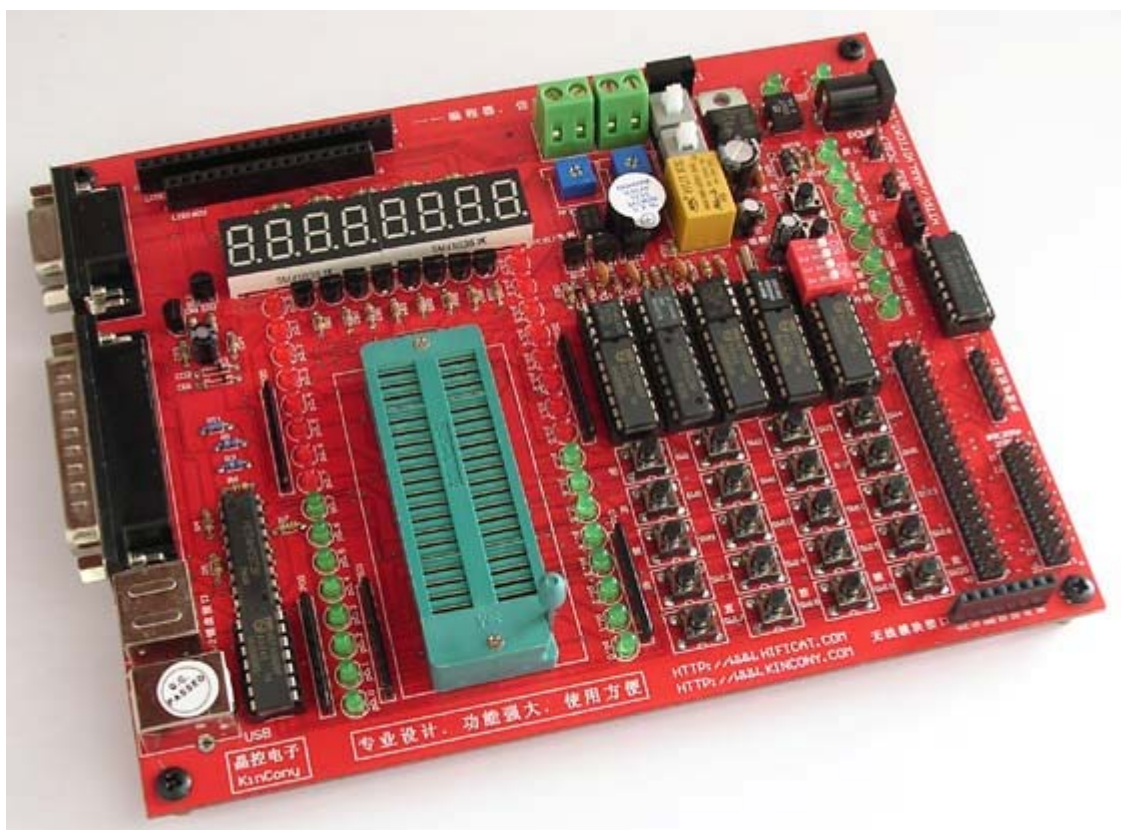


图1 51单片机综合学习系统主机部分图片





图2 51单片机综合学习系统配套书本教程——《单片机快速入门》

上图是我们将要使用的51单片机综合学习系统硬件平台，如图1所示，本期实验我们用到了综合系统主机、板载的 DS1302时钟芯片，综合系统其它功能模块原理与使用详见前几期《电子制作》杂志及后期连载教程介绍。

在很多单片机系统中都要求带有实时时钟电路，如最常见的数字钟、钟控设备、数据记录仪，这些仪表往往需要采集带时标的数据，同时一般它们也会有一些需要保存起来的重要数据，有了这些数据，便于用户后期对数据进行观察、分析。本小节就介绍市面上常见的时钟芯片 DS1302的应用。DS1302是美国 DALLAS 公司推出的一款高性能、低功耗、带内部 RAM 的实时时钟芯片(RTC)，也就是一种能够为单片机系统提供日期和时间的芯片。通过本小节的学习，我们将会把 RTC 相关的一些技术粗略介绍一下，然后介绍 DS1302与单片机之间的软硬件应用。

**DS1302时钟芯片简介**

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单片机进行通信。实时时钟/日历电路提供秒、分、时、日、星期、月、年的信息，每个月的天数和闰年的天数可自动调整，时钟操作可通过 AM/PM 标志位决定采用24或12小时时间格式。DS1302与单片机之间能简单地采用同步串行的方式进行通信，仅需三根 I/O 线：复位（RST）、I/O 数据线、串行时钟（SCLK）。时钟/RAM 的读/写数据以一字节或多达31字节的字符组方式通信。DS1302工作时功耗很低，保持数据和时钟信息时，功耗小于1mW。

**DS1302的内部结构**

DS1302的外部引脚功能说明如图3所示：

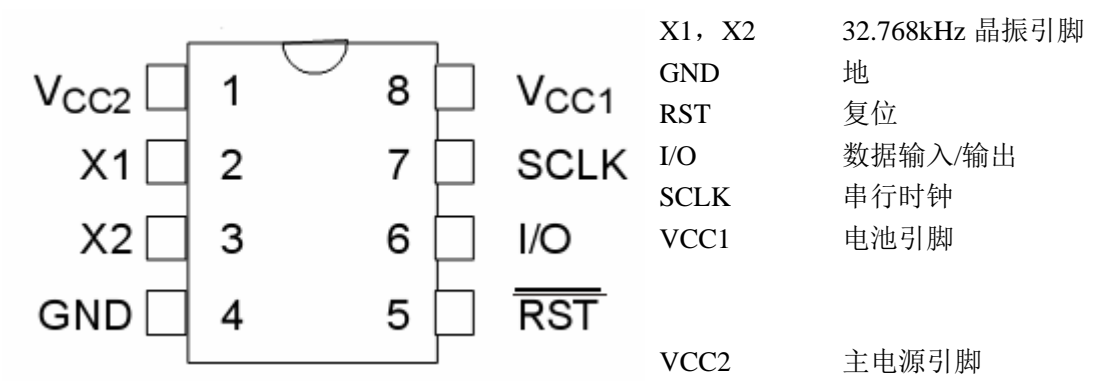


图3 DS1302封装图

DS1302的内部结构如图4所示，主要组成部分为：移位寄存器、控制逻辑、振荡器、实

时时钟以及 RAM。虽然数据分成两种，但是对单片机的程序而言，其实是一样的，就是对特定的地址进行读写操作。

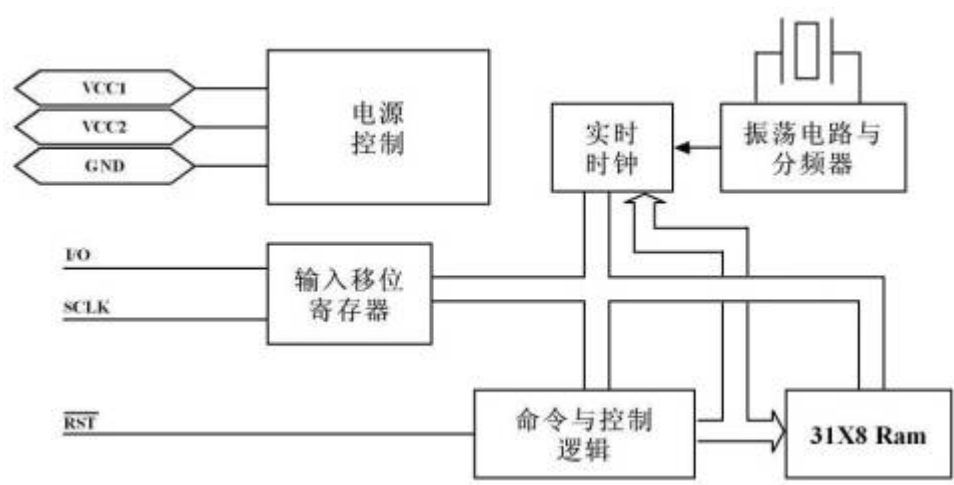


图4 DS1302的内部结构图

DS1302含充电电路，可以对作为后备电源的可充电电池充电，并可选择充电使能和串入的二极管数目，以调节电池充电电压。不过对我们目前而言，最需要熟悉的是和时钟相关部分的功能，对于其它参数请参阅数据手册。

**DS1302的工作原理**

DS1302工作时为了对任何数据传送进行初始化，需要将复位脚（RST）置为高电平且将8位地址和命令信息装入移位寄存器。数据在时钟（SCLK）的上升沿串行输入，前8位指定访问地址，命令字装入移位寄存器后，在之后的时钟周期，读操作时输出数据，写操作时输出数据。时钟脉冲的个数在单字节方式下为8+8（8位地址+8位数据），在多字节方式下为8加最多可达248的数据。

**DS1302的寄存器和控制命令**

对 DS1302的操作就是对其内部寄存器的操作，DS1302内部共有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。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以外的寄存器。日历、时间寄存器及控制字如表1所示：

	7	6	5	4	3	2	1	0
寄存器名称	1	RAM/C K	A4	A3	A2	A1	A0	RD/W
秒寄存器	1	0	0	0	0	0	0	
分寄存器	1	0	0	0	0	0	1	
小时寄存器	1	0	0	0	0	1	0	
日寄存器	1	0	0	0	0	1	1	
月寄存器	1	0	0	0	1	0	0	

星期寄存器	1	0	0	0	1	0	1
年寄存器	1	0	0	0	1	1	0
写保护寄存器	1	0	0	0	1	1	1
慢充电寄存器	1	0	0	1	0	0	0
时钟突发寄存器	1	0	1	1	1	1	1

表1：日历、时钟寄存器与控制字对照表

最后一位 RD/W 为“0”时表示进行写操作，为“1”时表示读操作。

DS1302内部寄存器列表如表2所示：

寄存器名称	命令字		取值范围	各位内容							
	写	读		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		
小时寄存器	84H	85H	01-12或00-23	12/2	0	A	HR		HR		
				4							
日期寄存器	86H	87H	01-28,29,30,31	0	0	10DATE			DATE		
月份寄存器	88H	89H	01-12	0	0	0	10M		MONTH		
周寄存器	8AH	8BH	01-07	0	0	0	0	0	DAY		
年份寄存器	8CH	8DH	00-99		10YEAR				YEAR		

表2：DS14302内部主要寄存器分布表

DS1302内部的 RAM 分为两类，一类是单个 RAM 单元，共31个，每个单元为一个8位的字节，其命令控制字为 COH~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，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）。

我们现在已经知道了控制寄存器和 RAM 的逻辑地址，接着就需要知道如何通过外部接口来访问这些资源。单片机是通过简单的同步串行通讯与 DS1302通讯的，每次通讯都必须由单片机发起，无论是读还是写操作，单片机都必须先向 DS1302写入一个命令帧，这个帧的格式如表1所示，最高位 BIT7固定为1，BIT6决定操作是针对 RAM 还是时钟寄存器，接着的5个 BIT 是 RAM 或时钟寄存器在 DS1302的内部地址，最后一个 BIT 表示这次操作是读操作抑或是写操作。

物理上，DS1302的通讯接口由3个口线组成，即 RST，SCLK，I/O。其中 RST 从低电平变成高电平启动一次数据传输过程，SCLK 是时钟线，I/O 是数据线。具体的读写时序参考图5，但是请注意，无论是哪种同步通讯类型的串行接口，都是对时钟信号敏感的，而且一般数据写入有效是在上升沿，读出有效是在下降沿（DS1302正是如此的，但是在芯片手册里没有明确说明），如果不是特别确定，则把程序设计成这样：平时 SCLK 保持低电平，在时钟变动前设置数据，在时钟变动后读取数据，即数据操作总是在 SCLK 保持为低电平的时候，相邻的操作之间间隔有一个上升沿和一个下降沿。

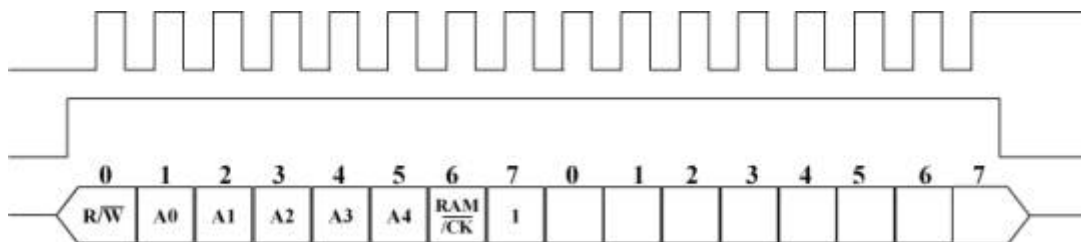
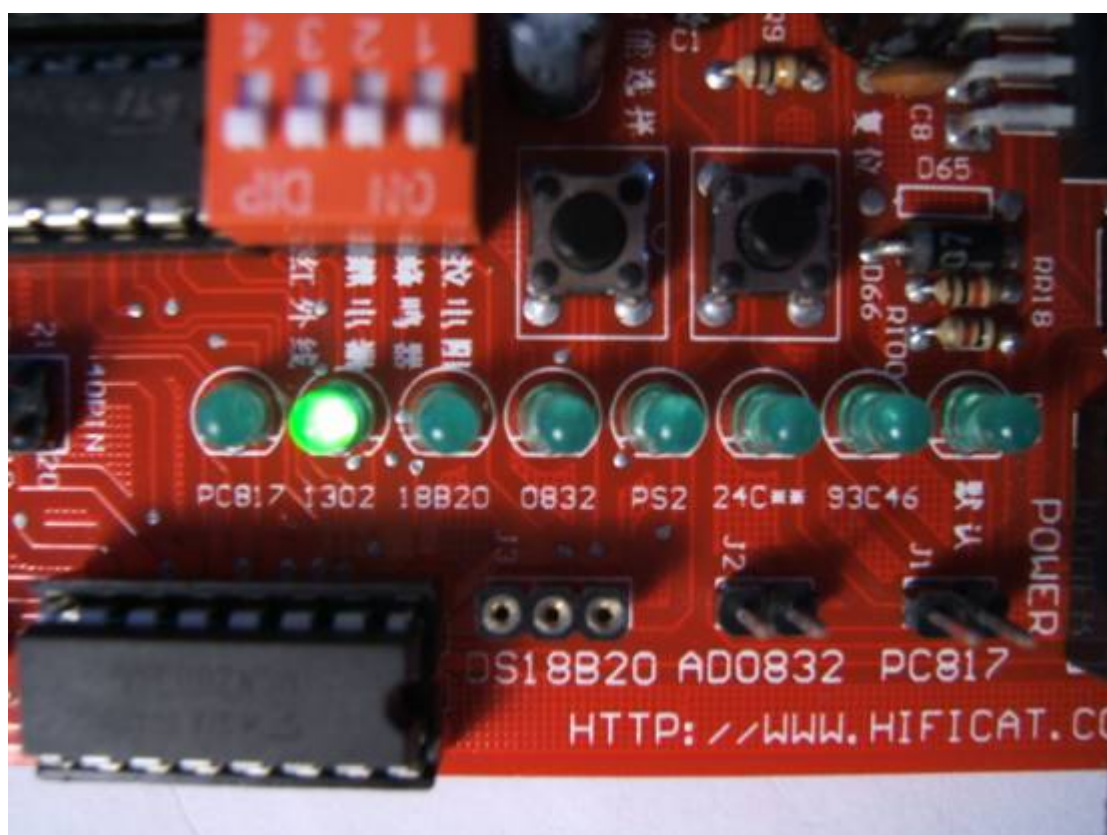


图5 DS1302的命令字结构

### DS1302的软硬件设计实例

本例将实现对 DS1302的读写操作，将时钟数据在 LED 数码管上显示出来。调试时将功能选择开关调到 DS1302的状态上。



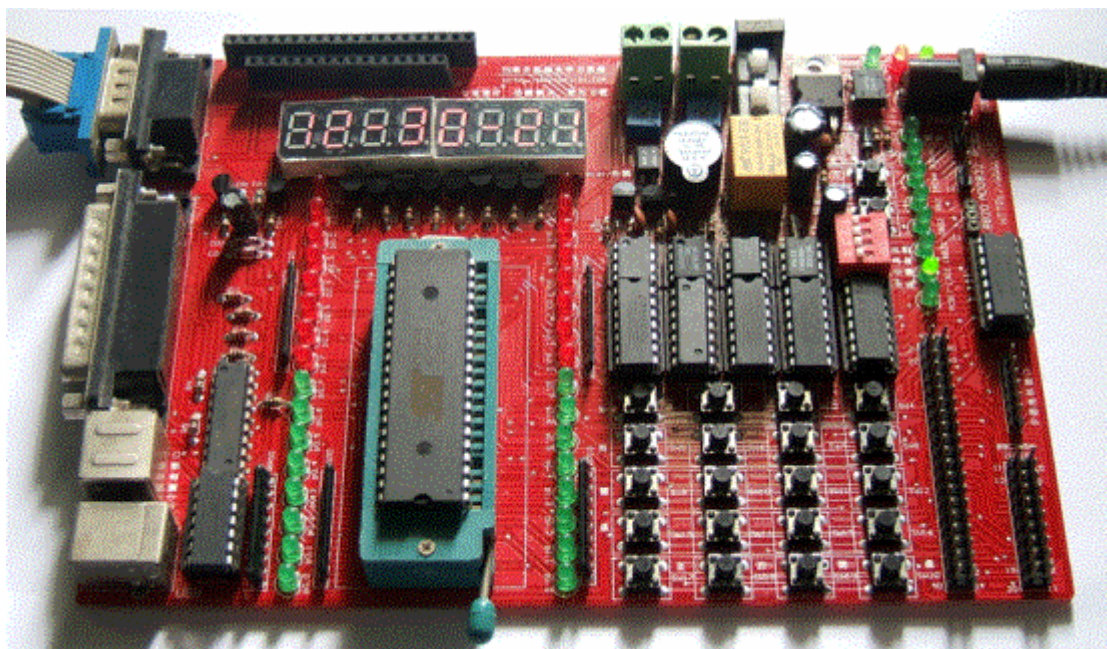


图6 DS1302实验演示图

## 硬件原理图

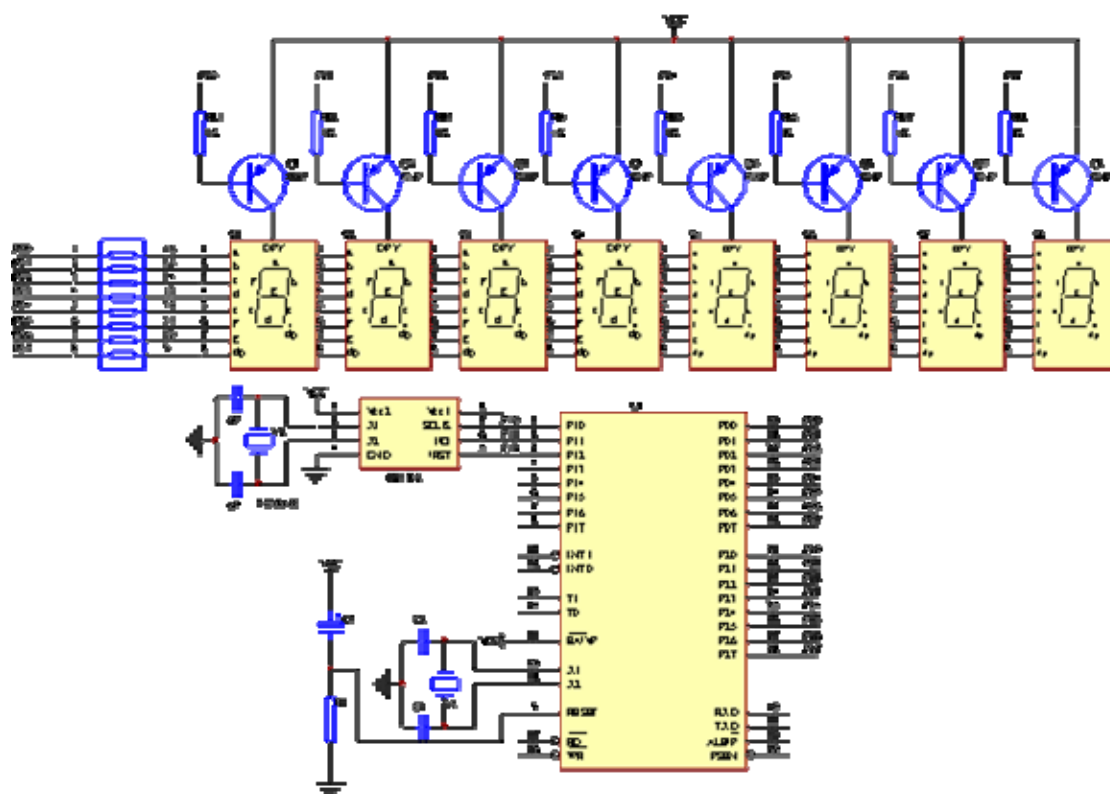


图10-46 硬件原理图

### 程序流程图



图10-47 软件流程图