课程设计报告

**姓名：杨秀佳**

**学号：1328403002**

**专业：电子信息工程**

**指导老师：邓晶**

苏州大学 电子信息学院

2016年4月

**摘要**

本次课程设计主要包含基于51单片机的4项实验设计：采用单总线数据传输时序的基于DS18B20的数字温度计设计；采用I2C总线时序的基于AT24C02的电子密码锁设计；SPI传输时序的基于DS1302电子日历的设计；基于nRF24L01无线数据传输

**关键词：**51单片机 DS18B20AT24C02 DS1302 nRF24L01

**目录**

摘要...............................................................1

目录...............................................................2

1. 基于DS18B20数字温度计设计..................................3

1.1 系统功能.......................................................3

1.2 系统组成.......................................................3

1.3 系统设计.......................................................3

1.3.1 总体设计.................................................3

1.3.2 硬件设计.................................................3

1.3.3 软件设计.................................................4

1.4 调试与结果.....................................................6

1. 基于AT24C02电子密码锁设计..................................7

2.1 系统功能.......................................................8

2.2 系统组成.......................................................8

2.3 系统设计.......................................................9

2.3.1 总体设计.................................................9

2.3.2 硬件设计.................................................10

2.3.3 软件设计.................................................10

2.4 调试与结果.....................................................12

1. 基于DS1302电子日历的设计...................................12

3.1 系统功能.......................................................13

3.2 系统组成.......................................................13

3.3 系统设计.......................................................13

3.3.1 总体设计.................................................13

3.3.2 硬件设计.................................................14

3.3.3 软件设计.................................................15

3.4 调试与结果.....................................................15

1. 基于NRF24L01无线数据传输...................................17

4.1 系统功能.......................................................18

4.2 系统组成.......................................................18

4.3 系统设计.......................................................18

4.3.1 总体设计.................................................18

4.3.2 硬件设计.................................................18

4.3.3 软件设计.................................................18

4.4 调试与结果.....................................................20

总结...............................................................20

1. **基于DS18B20数字温度计的设计**

**1.1系统功能**

1、用液晶屏实时显示当前环境的温度，分辨率为0.0625℃。

2、直接显示数字温度值。

3、可以任意设定温度的上下限报警功能。

4、系统的测量范围为 -55 ℃ ~+ 125 ℃ ； 在 10~+ 85°C 范围内，精度为 ± 0.5°C 。

**1.2系统组成**

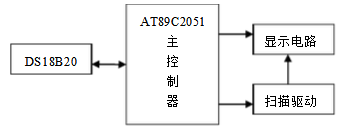
****

图1-1系统组成图

本设计由单片机主控芯片STC89C52和DS18B20芯片，液晶显示模块LCD1602组成。

DS18B20是单总线数字式温度传感器，采用单总线协议，即与单片机接口仅需占用一个I/O端口，无需任何外围器件，直接将温度转化为数字信号，以数字码形式串行输出。可由一根I/O数据线既供电又传输数据。

LCD1602可以显示两行字符，每行16个字符，只能显示ASCII码字符。用来显示读取的序列码和温度值。

**1.3系统设计**

**1.3.1总体设计**

将18B20的单总线DQ与单片机接口P2.2相连，通过时序控制首先对其进行初始化，然后发送读写和温度转换命令，使DS18B20内部温度传感器开始工作，最后从温度寄存器中读取两字节二进制码，转换为温度值后显示在LCD上。

同理，对于序列码的读取，也要在18B20初始化成功的基础上，发送读ROM命令（该命令只适用于总线上存在单只DS18B20），将读取的字符显示在液晶屏上。

**1.3.2硬件设计**

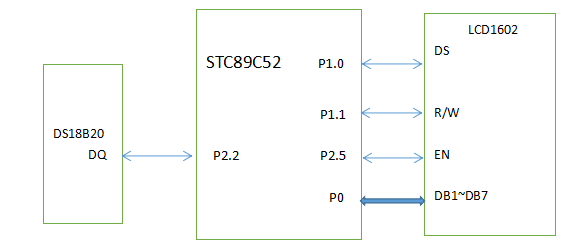
****

图2-2 系统连接图

**1.3.3软件设计**

读出温度子程序

读出温度子程序的主要功能是读出RAM中的9字节。在读出时必须进行CRC校验，校验有错时不能进行温度数据的改写。读出温度子程序流程图如下图所示：

调用显示子程序

1s到?

初次上电

读出温度值

温度计酸处理

显示数据刷新

发温度转换开始命令

图1-3 读出温度子程序流程图

读出温度子程序

读出温度子程序的主要功能是读出RAM中的9字节。在读出时须进行CRC校验，校验有错时不进行温度数据的改写。得出温度子程序流程图如下图所示。

发DS18B20复位指令

CRC校验正确？

发跳过ROM指令

移入温度暂存器

发读取温度指令

读取操作，CRC校验

9字节完？

图1-4 得出温度子程序流程图

温度转换命令子程序

温度转换命令子程序主要是发温度转换开始命令。当采用12位分辨率时，转换时间约为750ms。在本程序设计中，采用1s显示程序延时法等待转换的完成。温度转换命令子程序流程图如下图所示。

发DS18B20复位命令

发跳过ROM命令

发温度转换开始命令

图1-5 温度转换命令子程序流程图

计算温度子程序

计算温度子程序将RAM中读取值进行BCD码的转换运算，并进行温度值正负的判定。计算温度子程序流程图如下左图所示.

现实数据刷新子程序 现实数据刷新子程序主要是对显示缓冲器中的显示数据进行刷新操作，当最高数据显示位为0时，将符号显示位移入下一位。现实数据刷新子程序流程图如下右图所示.

温度数据移入显示寄存器

计算小数位温 度BCD值

十位数0？

温度零下？

计算整数位温 度BCD值

百位数0？

置+标志

十位数显示符号百位数不显示

温度值取补码置‘一’标志

百位数显示数据（不显示符号）

# 图1-6 计算温度子程序流程图

# 图1-7 现实数据刷新子程序流程图

**1.4 调试与结果**

如图1-8，液晶屏直接显示数字温度值，在液晶1602的第一行显示的即为序列码，第二行为实时温度。液晶屏实时显示当前环境的温度，分辨率为0.0625℃。当手轻轻触碰，温度升高时发出报警声。

图1-8液晶显示结果

**第二章 基于AT24C02电子密码锁的设计**

**2.1 系统功能**

1、设置初始密码为“000000”

2、进入密码锁后，按键S11修改密码，并在液晶显示密码

S12为确定键，修改成功

S13为重新设置键

S14为退出密码锁

3、在未进入密码锁时，S11,S12,S13,S14无效

**2.2 系统组成**

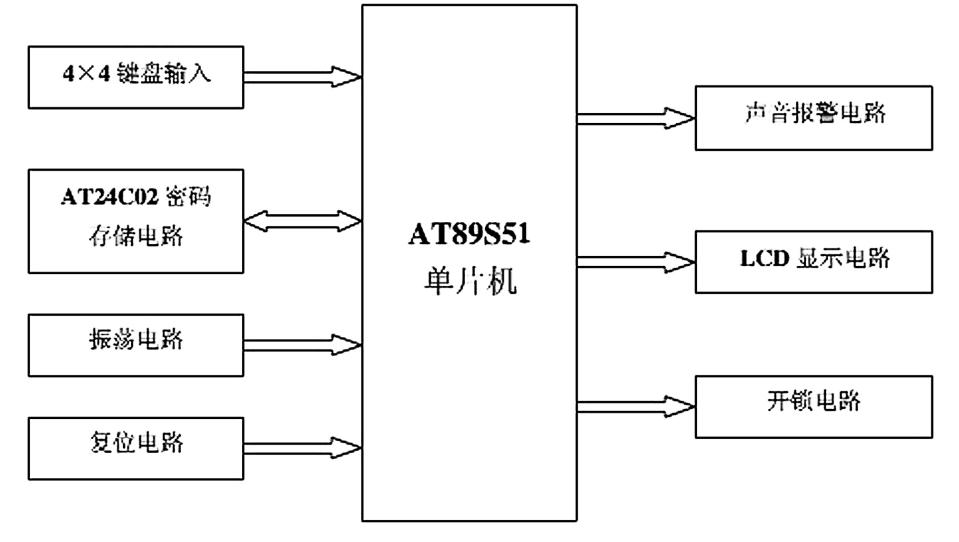
****

图2-1 系统组成图

本设计由STC89C52单片机的主芯片和具有I2C总线的接口的串行E2PROM AT24C02以及显示液晶LCD。

I2C要求有两条信号线，一条串行数据线SDA，一条串行时钟线SCL。每根连接到总线的器件都可以通过唯一的地址进行寻址。串行的8位双向数据传输，其位速率在标准模式下可达100Kbit/s,在快速模式下可达400Kbit/s,在高速模式下可达3.4Mbit/s。只有在总线空闲时才允许启动数据传送，在数据传送过程中当时钟线为高电平时数据线必须保持稳定状态不允许有跳变，时钟线为高电平时数据线的任何电平变化将被看作总线的起始或停止信号。

•起始信号

时钟线保持高电平期间数据线电平从高到低的跳变作为I2C 总线起始信号。

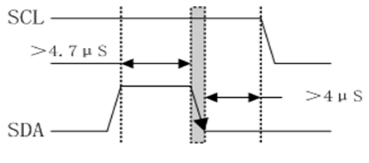


图2-2 起始信号

•停止信号

时钟线保持高电平期间数据线电平从低到高的跳变作为I2C 总线停止信号。

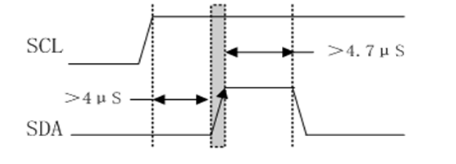


图2-3 停止信号

I2C 总线数据传送时每成功地传送一个字节数据后接收器都必须产生一个应答信号应答的器件在第9 个时钟周期时将SDA 线拉低表示其已收到一个8 位

数据。

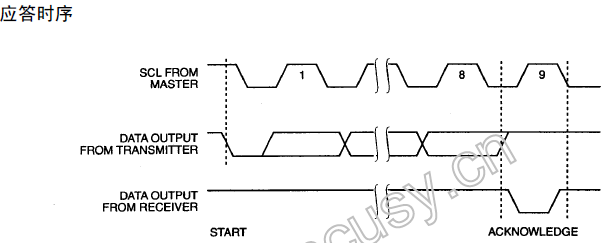


图2-4应答信号

•字节写

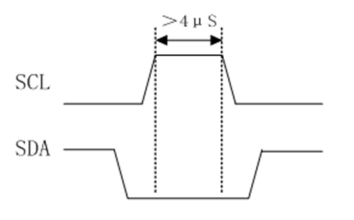
在字节写模式下主器件发送起始命令和写命令（字节为0XA0）给从器件,在从器件产生应答信号后，再发送数据到被寻址的存储单元。

图2-5逻辑“0”

•读操作的初始化方式和写操作时一样仅把R/W 位置为1 。

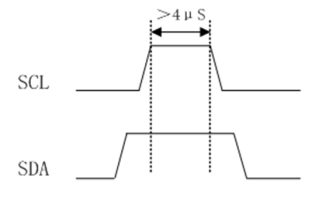


图2-5逻辑“1”

**2.3 系统设计**

**2.3.1 总体设计**

本设计使用STC89C52的两个I/O口，P2.0和P2.1,来模仿SDA和SCL,由于此实验只用一块AT24C02,写地址和读地址固定为0X80和0X81。通过时钟线SCL配合SDA的高低转换来进行启动和停止控制，然后发送从器件地址信息，从器件送回应答信号后，主机开始进行对从器件的读或写，这里用来做密码锁，所以对从器件的操作就是给相应的存储单元写密码（修改新密码），或者对某个存储单元读密码，然后与按键的值进行对比，验证密码是否正确。

**2.3.2 硬件设计**

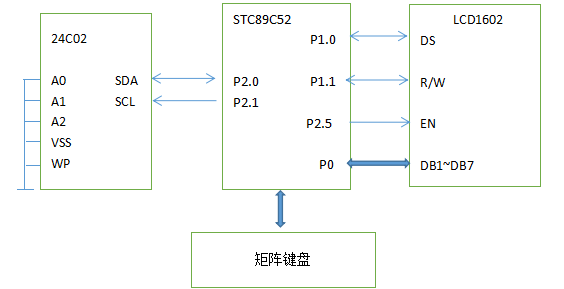


图2-6系统连接图

**2.3.3 软件设计**

主程序流程图：

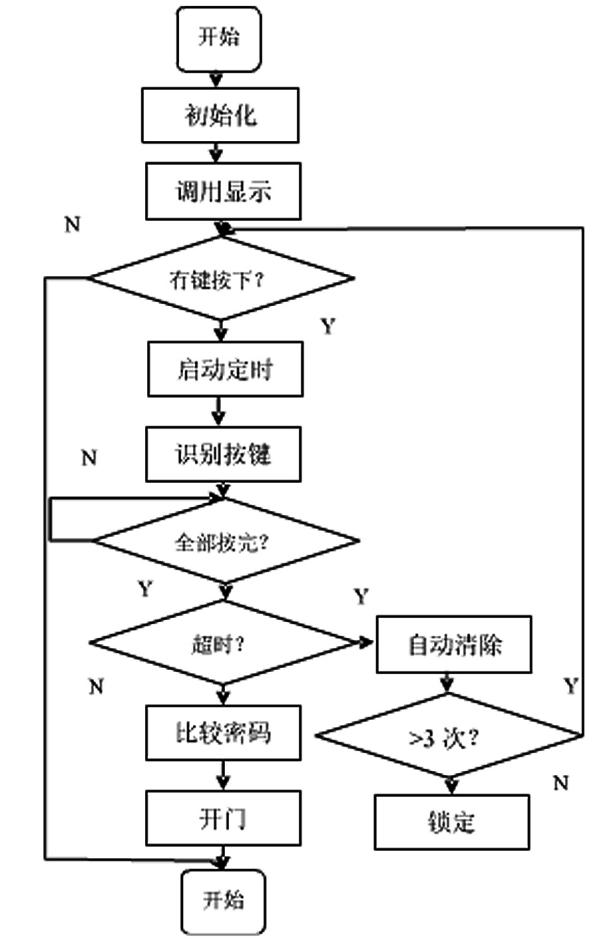


图2-7主程序流程图

修改密码流程图：

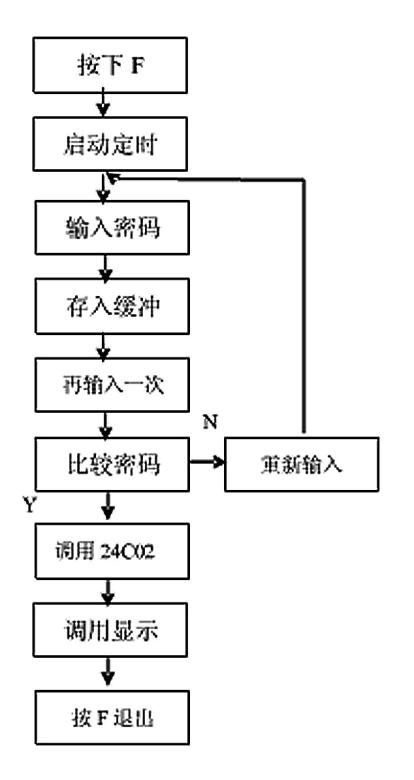


图2-8修改密码流程图

**2.4 调试与结果**

1、 设置初始密码为“000000”

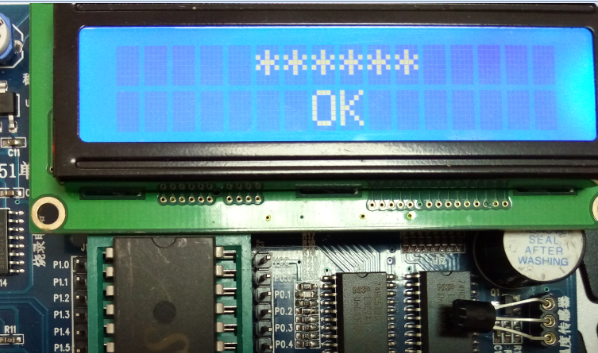


图2-8第一次进入密码锁

2、进入密码锁后，按键S11修改密码，并在液晶显示密码

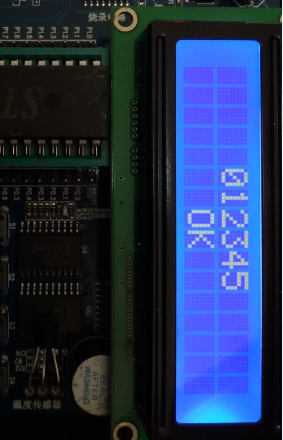


图2-9 修改密码并显示

3、S12为确定键，修改成功

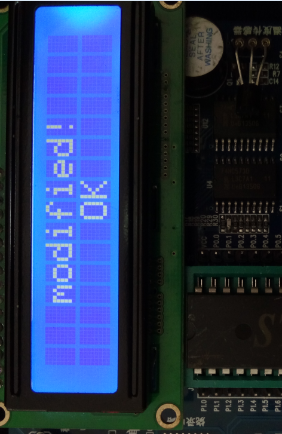


图2-10确定修改成功，屏幕显示“modified!”

4、S13为重新设置键

S14为退出密码锁

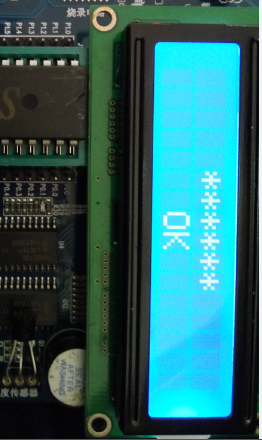


图2-11再次用新密码进入密码锁

**第三章 基于DS1302电子日历的设计**

**3.1 系统功能**

1、在LCD上显示当前的时间信息，即年、月、日、星期、时、分、秒

2、具有时间的调校功能

3、显示当前调节对象

4、日期时间加调整与保存。

**3.2 系统组成**

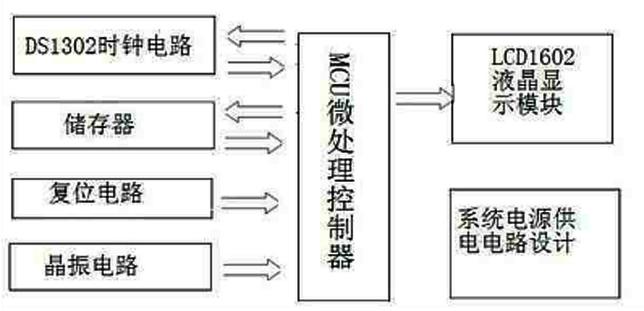
****

图3-1系统组成图

本设计由STC89C52单片机的主芯时钟日历DS1302芯片以及显示液晶LCD。DS1302与单片机之间可简单地采用SPI串行方式进行通讯，仅用到三根信号线：RST（复位），I/O(数据线)，SCLK(同步串行时钟)。实时时钟具有计算2100年之前的秒，分，时，日，月，星期，年的能力，还具有闰年调整能力。LCD第一行用于显示年月日及星期，第二行显示时分秒。

**3.3 系统设计**

**3.3.1 总体设计**

DS1302与51单片机的连接只需要利用单片机的三个I/O引脚对DS1302的SCLK、I/O、和RST进行控制，根据系统功能要求，基于DS1302的电子日历原理图如图3-3所示，51单片机的P2.0,P2.1和P2.4分别与DS1302 的SCLK,I/O和RST相连，DS1302的主电源VCC2与系统电源VCC相连，DS1302的晶振选用32.768KHz，LCD用于显示当前的时间信息，按键S4,S8,S12用于时间的调整。DS1302共有10个寄存器，其中要控制操作的有年、月、日、星期、时、分、秒寄存器，通过对DS1302单字节读/写操作或者多字节读/写操作，可对其内部的时钟日历寄存器，控制寄存器，RAM存储器进行访问，数据是以BCD码的格式存放。

1、DS1302读写时序：

（1）初始化：先把SCLK置0接着把RST端置1，最后才能输出SCLK脉冲

（2）读：DS1302是通过SPI串行总线跟单片机通信的，当进行一次读写操作时最少得读写两个字节，第一个字节是控制字节，就是一个命令，告诉DS1302是读还是写操作，是对RAM还是对CLOK寄存器操作。第二个字节就是要读或写的数据了。单字节读：只有在SCLK为低电平时，才能将CE置为高电平。所以在进行操作之前先将SCLK置低电平，然后将CE置为高电平，接着开始在IO上面放入要传送的电平信号，然后跳变SCLK。数据在SCLK上升沿时，DS1302读取数据，在SCLK下降沿时，DS1302放置数据到IO上

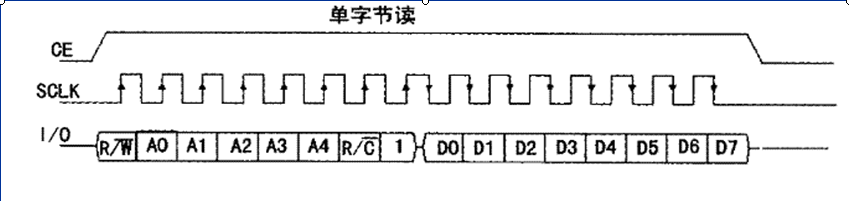


图3-2单字节读

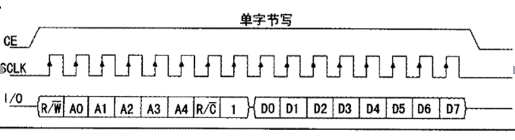
（3）写：DS1302是通过SPI串行总线跟单片机通信的，当进行一次读写操作时最少得读写两个字节，第一个字节是控制字节，就是一个命令，高速DS1302是读还是写操作，是对RAM还是对CLOK寄存器操作。第二个字节就是要读或写的数据了。单字节写：只有在SCLK为低电平时，才能将RST置为高电平。说以在进行操作之前先将SCLK置低电平，然后将RST置为高电平，开始在IO上面放入要传送的电平信号，然后跳变SCLK。数据在SCLK上升沿时，DS1302读取数据，在SCLK下降沿时，DS1302放置数据到IO上 

图3-3单字节写

**3.3.2 硬件设计**

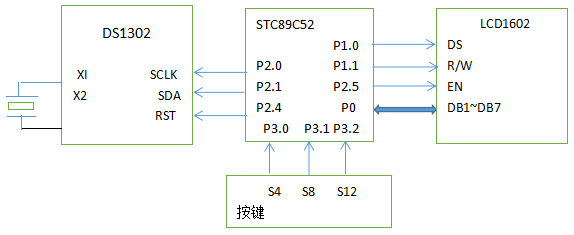


图3-4硬件连接图

**3.3.3 软件设计**

时间程序的设计：

因为使用了时钟芯片DS1302，时间程序只需从DS1302各寄存器中读出年、周、月、日、［小］时、分、秒等数据，再处理即可。在首次对DS1302进行操作之前，必须对它进行初始化，然后从DS1302中读出数据，再经过处理后，送给显示缓冲单元。时间程序流程图见图所示。

时间调整程序设计

调整时间用三个调整按钮，分别定义为模式选择，减少，增加按钮。在调整时间过程中，首先要调节模式设置按钮，在显示屏上会显示对应的要调试的英文名称。例如，要调节月份，先调节模式设置按钮，直到屏幕上test行上显示：month，此时就可以调节减少和增加按钮，对其进行调试。时间调整程序流程图如图所示。

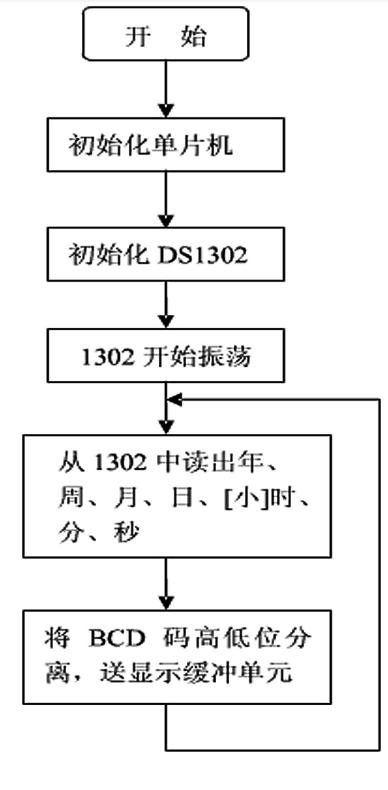


图3-4 时间程序的设计图

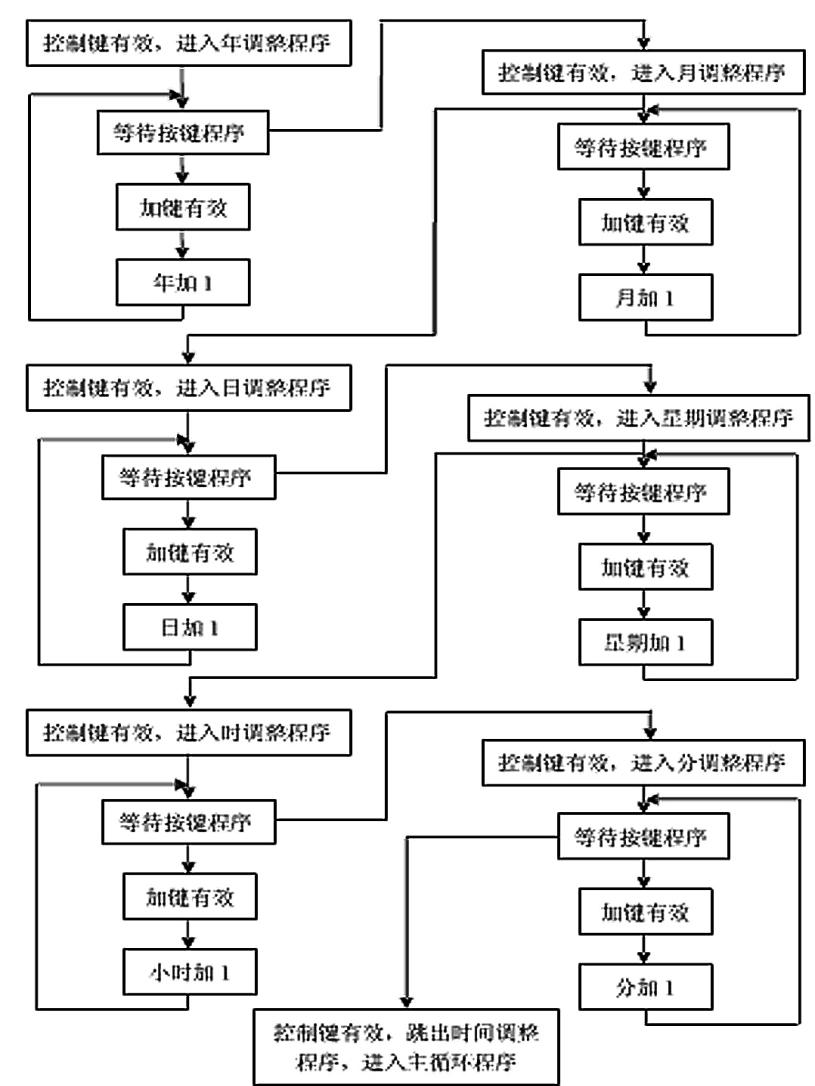


图3-6时间调整程序设计图

**3.4 调试与结果.**

1、在LCD上显示当前的时间信息，即年、月、日、星期、时、分、秒



图3-7上电后显示的日期时间，并开始计时

2、具有时间的调校功能

3、显示当前调节对象



图3-8按下S12键后停止计时，按下S4键后，光标停在个位并闪烁。

4、日期时间加调整与保存

 图3-9按S8加键，修改好的时间日期

**第四章 基于nRF24L01无线数据传输**

**4.1 系统功能**

1、实现两个单片机之间的点对点通讯,最大通讯距离为100米。

2、主机按“S1~12”,发送“0~9，a，b”,从机会接收到相应的字符并用数码管显示。

3、主从机可以随时调换角色。

**4.2 系统组成**

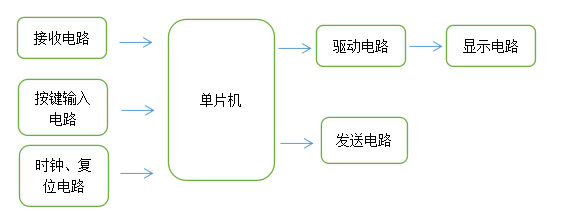
****

图4-1系统组成图

硬件系统由STC89C52芯片、NRF24L01模块、矩阵键盘和数码管组成，51芯片的I/O接口模拟SPI串行口，通过时钟线SCK来控制对NRF24L01的读写操作。通过按键控制模式状态转换，由于矩阵键盘第四行用P3.3口扫描，而芯片的片选线SCN也用P3.3，为避免冲突，所以只扫面前三行。发送或接受到的数据用数码管显示。

**4.3 系统设计**

**4.3.1 总体设计**

本设计由51单片机芯片的两个I/O接口P1.7、P1.6模拟SPI时序（MOSI和MISO），CSN为芯片片选线，SCK为芯片控制的时钟线。CE为芯片的模式控制线。在 CSN 为低的情况下，CE 协同NRF24L01 的CONFIG 寄存器共同决定NRF24L01 的状态。当主机设置为发送模式时，通过SPI时序给既定的发送缓冲区写入数据。当从机设置为接受模式时，通过SPI从相应的接收缓冲区读出数据，并在数码管上显示。

nRF24l01 工作模式：

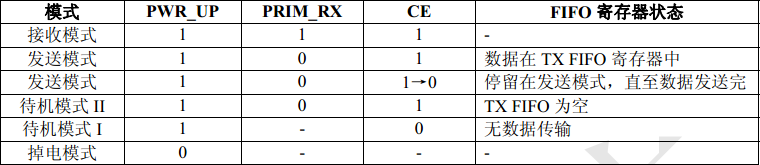


图4-2系统工作模式图

①待机模式：待机模式I在保证快速启动的同时减少系统平均消耗电流。在待机模式I下，晶振正常工作。在待机模式II下部分时钟缓冲器处在工作模式。当发送端TX FIFO寄存器为空并且CE为高电平时进入待机模式II。在待机模式期间，寄存器配置字内容保持不变。

②掉电模式：在掉电模式下,nRF24L01各功能关闭，保持电流消耗最小。进入掉电模式后，nRF24L01停止工作，但寄存器内容保持不变。掉电模式由寄存器中PWR\_UP位来控制

③ShockBurst模式：nRF24L01可以与成本较低的低速MCU相连。高速信号处理是由芯片内部的射频协议处理的，nRF24L01提供SPI接口，数据率取决于单片机本身接口速度。SockBurst模式通过允许与单片机低速通信而无线部分高速通信，减小了通信的平均消耗电流。在ShockBurstTM接收模式下，当接收到有效的地址和数据时IRQ通知MCU，随后MCU可将接收到的数据从RX FIFO寄存器中读出。发送模式下，自动生成前导码和CRC校验，数据发送完毕，IRQ通知MCU。

④增强型 ShockBurst模式：数据通道0是唯一的一个可以配置为40位自身地址的数据通道。1~5数据通道都为8位自身地址和32位公用地址。所有的数据通道都可以设置为增强型ShockBurst模式。 nRF24L01在确认收到数据后记录地址，并以此地址为目标地址发送应答信号。在发送端，数据通道0被用做接收应答信号，因此，数据通道0的接收地址要与发送端地址相等以确保接收到正确的应答信号。

⑤增强型 ShockBurst发送模式：当MCU有数据要发送时，接收节点地（TX\_ADDR）和有效数据(TX\_PLD)通过SPI接口写入nRF24L01。发送数据的长度以字节计数从MCU写入TX FIFO。当CSN为低时数据被不断的写入。发送端发送完数据后，将通道0设置为接收模式来接收应答信号，其接收地址(RX\_ADDR\_P0)与接收端地址(TX\_ADDR)相同。

⑥增强型 ShockBurst接收模式： ShockBurst接收模式是通过设置寄存器中PRIM\_RX位为高来选择的。准备接收数据的通道必须被使能（EN\_RXADDR寄存器），所有工作在增强型ShockBurst模式下的数据通道的自动应答功能是由(EN\_AA寄存器)来使能的，有效数据宽度是由RX\_PW\_Px寄存器来设置的。

**配置NRF24L01模块：**

NRF2401的所有配置工作都是通过SPI完成，共有30字节的配置。

①数据宽度：声明射频数据包中数据占用的位数。这使得NRF24L01能够区分接收数据包中的数据和CRC校验码；

②地址宽度：声明射频数据包中地址占用的位数。这使得NRF24L01能够区分地址和数据；

地址：接收数据的地址，有通道0到通道5的地址；

③CRC：使NRF24L01能够生成CRC校验码和解码。

当使用NRF24L01片内的CRC技术时，要确保在配置字(CONFIG的EN\_CRC)中CRC校验被使能，并且发送和接收使用相同的协议。

SPI读时序：

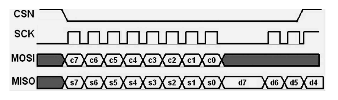


图4-3 SPI读时序图

SPI写时序：

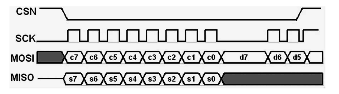


图4-4 SPI写时序图

**4.3.2 硬件设计**

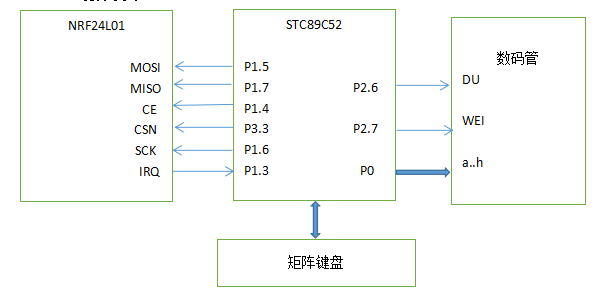


图4-5 系统硬件连接图

**4.3.3 软件设计**

软件实现的流程图:

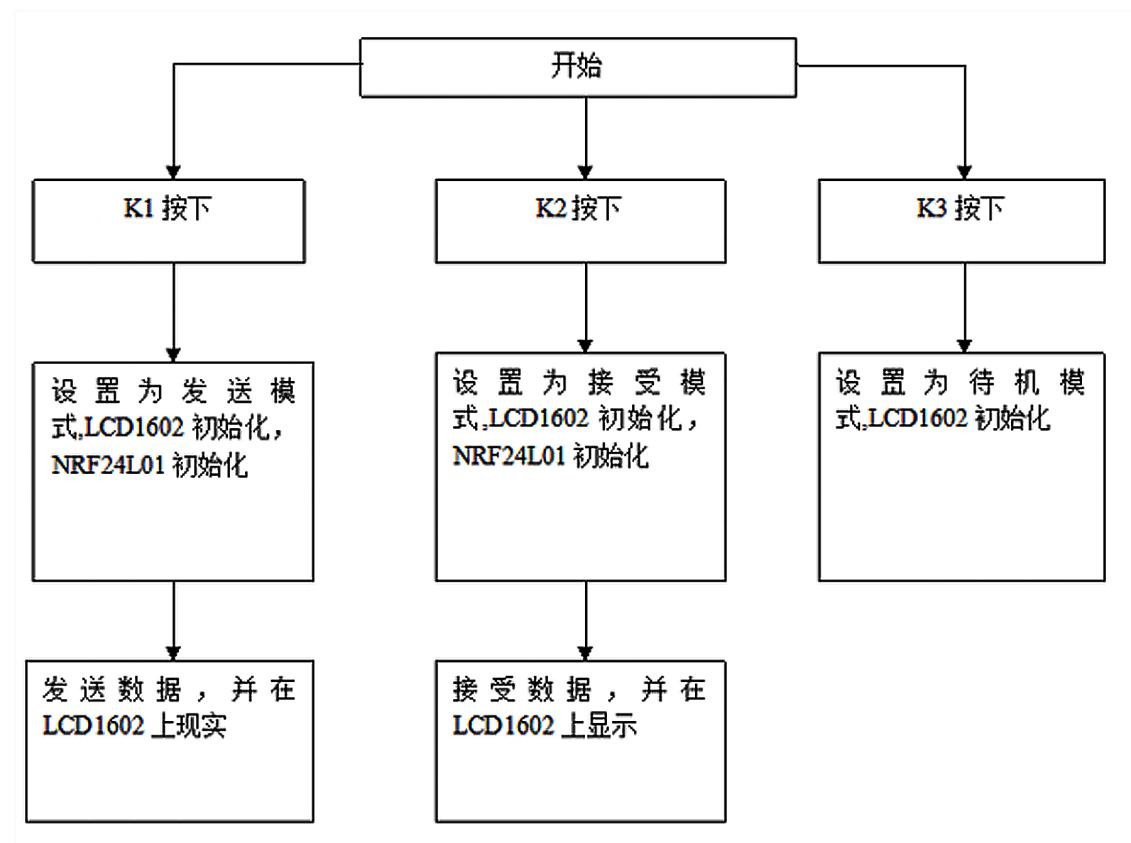
****

图4-6 软件实现的流程图:

**4.4 调试与结果**

1、实现两个单片机之间的点对点通讯,最大通讯距离为100米。

2、主机按“S1~12”,发送“0~9，a，b”,从机会接收到相应的字符并用数码管显示。

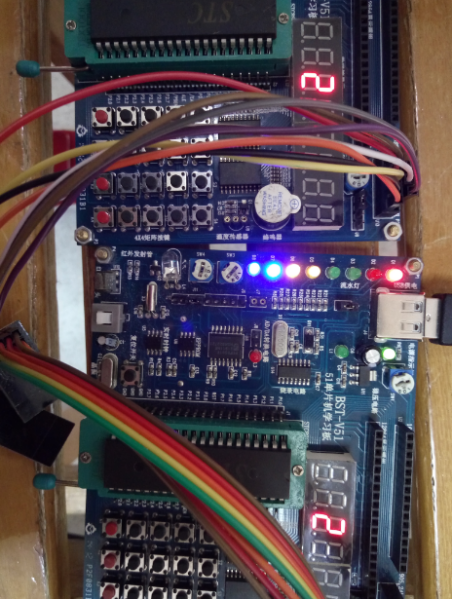


图4-7 主机发送一个字符“2”，并在数码管显示，从机接收到“2”，同样在数码管显示。

**总结**

历时几个月的单片机课程设计已经结束了，在这几个月的时间里，我们在老师的指导下完成了：采用单总线数据传输时序的基于DS18B20的数字温度计设计；采用I2C总线时序的基于AT24C02的电子密码锁设计；SPI传输时序的基于DS1302电子日历的设计；基于nRF24L01无线数据传输设计。在进行实验的过程中，我们了解并熟悉DS18B20、AT89C51、AT24C02、DS1302以及nRF24L01的工作原理和性能。并且通过这些设计，我们将单片机课上学到的知识进行运用，并在实际操作中发现问题，解决问题，更加增加对知识的认识和理解。

在课程设计的过程中，也遇到了一些问题。在将程序烧到单片机之后，接上电源，发现数码管没有出现正常的“000.0”，而是出现了“666.6”，在检查线路无误之后，将8个输入输出端口进行掉序。于是，我便将8个端口进行调换，之后再次接通电源，数码管显示“000.0”，调试成功。解决这些问题让我又学习到了许多知识。总之，通过实验不仅将课本的知识与实践相结合，而且在实践中更加深入了解书中原本抽象的知识。这也是整个课程设计中最有收获的地方。