**摘 要**

随着科技的进步与现代产业的飞速发展，对控制系统的发展也提出了越来越高的要求，非接触控制、中远程通信正在扮演这越来越重要的角色，所以单片机的中远程通信的意义也愈发重要。作为一名工科生，加强对这方面的学习是很有必要的。基于AT89C51单片机与HC-08蓝牙模块通信的基础，我们设计了能够实现在手机模拟串口APP的客户端上进行温度监视与控制的系统。该系统主要由蓝牙通信模块，灯光模拟加热电路，单片机控制电路，基于DS18B20的温度监视电路等部分组成。画出了系统电路原理图，进行了软件设计，给出了系统流程图，并编写了系统程序。最后在进行系统仿真的基础上进行了实物制作，实物调试结果表明，所设计的系统能够满足要求。本系统具有成本低，安全实用，80米左右通信等特点。

**关键词：**AT89C51；HC-08蓝牙；DS18B20；LCD显示屏；

[一、概述 4](#_Toc468203959)

[1.1 课程考核目的 4](#_Toc468203960)

[1.2 设计任务及要求 4](#_Toc468203961)

[1.3设计需要的相关知识 4](#_Toc468203962)

[二、总体设计方案与说明 5](#_Toc468203963)

[2.1系统总体设计方案 5](#_Toc468203964)

[2.2系统的技术指标 5](#_Toc468203965)

[2.3 AT89C51单片机的串口 5](#_Toc468203966)

[2.3.1 概念 5](#_Toc468203967)

[2.3.2 串行口结构 6](#_Toc468203968)

[2.3.3 特殊功能寄存器PCON 7](#_Toc468203969)

[2.3.4串行口的4种工作方式 8](#_Toc468203970)

[三、系统硬件部分设计 10](#_Toc468203971)

[3.1 Protel DXP电路原理图 10](#_Toc468203972)

[3.2 LCD显示电路 11](#_Toc468203973)

[3.2.1 LCD 1602引脚 11](#_Toc468203974)

[3.2.2．LCD1602字符的显示及命令 11](#_Toc468203975)

[3.3 HC-08蓝牙模块电路 12](#_Toc468203976)

[3.3.1 模块简介 13](#_Toc468203977)

[3.3.2 HC-08蓝牙引脚定义 13](#_Toc468203978)

[3.4 温度检测电路 14](#_Toc468203979)

[3.4.1 DS18B20模块简介 14](#_Toc468203980)

[3.4.2 引脚功能 14](#_Toc468203981)

[3.4.3 编程方式 14](#_Toc468203982)

[3.5 模拟加热电路（本设计中以LED灯和继电器模拟加热电路） 15](#_Toc468203983)

[四、系统软件部分设计 16](#_Toc468203984)

[4.1系统软件流程图 16](#_Toc468203985)

[4.2 程序清单 16](#_Toc468203986)

[五、系统仿真及实物制作 17](#_Toc468203987)

[5.1仿真软件 17](#_Toc468203988)

[5.2程序编译软件 18](#_Toc468203989)

[5.3编译过程 18](#_Toc468203990)

[5-4 Proteus仿真原理图 19](#_Toc468203991)

[5.5仿真过程 19](#_Toc468203992)

[5.6实物制作及功能演示 23](#_Toc468203993)

[六、总结 25](#_Toc468203994)

[七、参考文献 25](#_Toc468203995)

[附录一 Protel DXP原理 26](#_Toc468203996)

[附录二 程序清单 27](#_Toc468203997)

[附录三 Proteus 仿真原理图 30](#_Toc468203998)

##### 一、概述

1.1 课程考核目的

通过本课程学习，使学生掌握控制系统设计的一般步骤，掌握系统总体控制方案的设计方法。使学生进一步掌握微型计算机应用系统的硬、软件开发方法，输入/输出(I/O)接口技术，单片机中远程通信的方法，应用程序设计技术，并能结合专业设计简单实用的单片机应用系统。针对课堂重点讲授内容使学生加深对单片机硬件原理的理解及提高C51语言程序设计的能力，为以后的毕业设计搭建了单片机系统应用平台，提高学生的开发创新能力。

1.2 设计任务及要求

设计要求：

设计一个手机蓝牙通过HC-08蓝牙模块联系AT89C51单片机，对其发出指令进行温度控制，并且能够将温度回馈到手机的模拟串口的客户端界面上。

主要任务：

（1）进行系统方案总体设计，画出系统设计的框图。

（2）硬件部分设计。将整个硬件系统划分为若干功能单元电路，使用Protel DXP画出每个单元电路原理图，绘出整个系统电路原理图。

（3）软件部分设计。画出程序流程图，编写系统源程序(有必要的注释)并调试。

（4）购买实物，进行实物连接。

（5）画出Proteus仿真图，载入程序，进行仿真。

（6）软硬件联调，完成系统工作调试。

在以上工作基础上完成课程设计报告，包括设计任务与要求，总体方案说明，电路原理图与说明，软件流程图和源程序清单，问题分析与解决方案，结论与体会，参考资料等。

1.3设计需要的相关知识

通信基础下的温度控制系统设计主要涉及单片机的串口通信。基础知识包括：单片机的串口方式，HC-08蓝牙的串口通信知识，LCD显示，DS18B20的温度显示等等。

##### 二、总体设计方案与说明

2.1系统总体设计方案

特别说明：本次设计中，使用一个与继电器连接LED灯模拟加热模块。



图2-1系统工作框图

本设计基本工作方式如图2.1-1所示，在手机模拟串口发出指令，通过HC-08蓝牙从机传递给单片机，单片机启动模拟加热模块，测温模块测得实时温度，然后将温度反馈给单片机，单片机再发送到LCD显示屏上显示或是回馈给手机客户端。

2.2系统的技术指标

利用51单片机接收从手机发出的指令，控制LCD显示电路、加热电路、测温电路，系统功能：

指令“0”LED灯亮，指令“1”LED灯灭，指令“2”，单片机将温度回馈给手机。

2.3 AT89C51单片机的串口

2.3.1 概念

AT89S51集成一个全双工通用异步收发（UART）串行口。

全双工：两个单片机之间串行数据可同时双向传输。

异步通信：收、发双方使用各自时钟控制发送和接收，省去收、发双方的1条同步时钟信号线，使异步串行通信连接更简单且易实现。

2.3.2 串行口结构

AT89S51串行口内部结构见图2-2。有两个物理上独立的接收、发送缓冲器SBUF（特殊功能寄存器），可同时收发数据。发送缓冲器只写不读，接收缓冲器只读不写，两个缓冲器共用一个特殊功能寄存器字节地址（99H）。

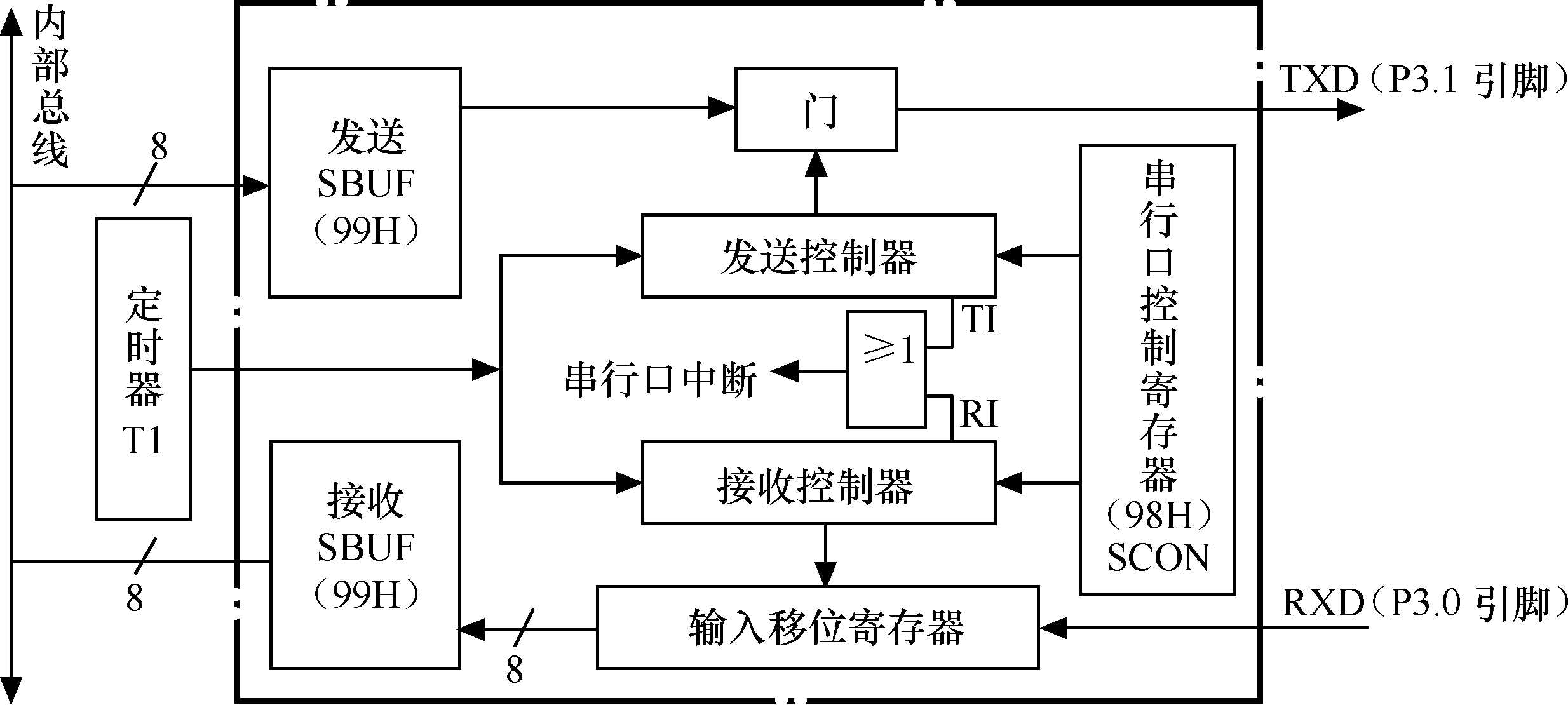


图2-2 串行口结构图

寄存器SCON各位功能：

1. SM0、SM1—串口4种工作方式选择，如表2-1



表2-1

(2）SM2—多机通信控制位

多机通信是在方式2和方式3下进行，因此SM2位主要用于方式2或方式3。

当串口以方式2或方式3接收时，如SM2=1，则只有当接收到的第9位数据（RB8）为“1”时，才使RI置“1”，产生中断请求，并将收到的前8位数据送入SBUF；当收到的第9位数据（RB8）为“0”时，则将收到的前8位数据丢弃。

当SM2=0时，则不论第9位数据是“1”还是“0”，都将接收的前8位数据送入SBUF中，并使RI置“1”，产生中断请求。

方式1时，如果SM2=1，则只有收到有效的停止位时才会激活RI。

方式0时，SM2必须为0。

(3）REN—允许串行接收位，由软件置“1”或清“0”。

REN=1，允许串行口接收数据。

REN=0，禁止串行口接收数据。

（4）TB8—发送的第9位数据

　在方式2和方式3时，TB8是要发送的第9位数据，其值由软件置“1”或清“0”。

在双机串行通信时，TB8一般作为奇偶校验位使用；也可在多机串行通信中表示主机发送的是地址帧还是数据帧，TB8=1为地址帧，TB8=0为数据帧。

（5）RB8—接收的第9位数据

在方式2和方式3时，RB8存放接收到的第9位数据。在方式1，如果SM2=0，RB8是接收到的停止位。在方式0，不使用RB8。

（6）TI—发送中断标志位

方式0时，串行发送的第8位数据结束时，TI由硬件置“1”，在其他工作方式中，串行口发送停止位的开始时，置TI为“1”。TI=1，表示1帧数据发送结束。TI位状态可供软件查询，也可申请中断。CPU响应中断后，在中断服务程序向SBUF写入要发送的下一帧数据。注意：TI必须由软件清“0”。

（7）RI—接收中断标志位

串口在方式0时，接收完第8位数据时，RI由硬件置“1”。在其他工作方式中，串行接收到停止位时，该位置“1”。RI=1，表示一帧数据接收完毕，并申请中断，要求CPU从接收SBUF取走数据。该位状态也可供软件查询。

注意：RI必须由软件清“0”。

2.3.3 特殊功能寄存器PCON

　 字节地址为87H，不能位寻址。格式见图2-3

仅最高位SMOD与串口有关，低4位功能在第2章中已介绍。

SMOD位：波特率选择位。

0703

图2-3 特殊功能寄存器PCON格式

2.3.4串行口的4种工作方式

4种工作方式由特殊功能寄存器SCON中SM0、SM1位定义，编码见表2.3-1。

**方式0**

方式0为同步移位寄存器输入/输出方式。该方式并不用于两个AT89S51单片机间的异步串行通信，而是用于外接移位寄存器，用来扩展并行I/O口。

方式0以8位数据为1帧，没有起始位和停止位，先发送或接收最低位。波特率是固定的，为fosc/12。

1．方式0输出

（1）方式0输出的工作原理

当单片机执行将数据写入发送缓冲器SBUF指令时，产生一个正脉冲，串口把8位数据以fosc/12固定波特率从RXD脚串行输出，低位在先，TXD脚输出同步移位脉冲，当8位数据发送完，中断标志位TI置“1”。

2．方式0输入

(1）方式0输入的工作原理:方式0输入时，REN为串行口允许接收控制位，REN=0，禁止接收；REN=1，允许接收。

当CPU向串行口SCON寄存器写入控制字（设置为方式0，并使REN位置“1”，同时RI=0）时，产生一正脉冲，串口开始接收数据。引脚RXD为数据输入端，TXD为移位脉冲信号输出端，接收器以fosc/12固定波特率采样RXD引脚数据信息，当接收器接收完8位数据时，中断标志RI置“1”，表示一帧接收完毕，可进行下一帧接收。

**方式1**

　方式1为双机串行通信方式，如图2-4。

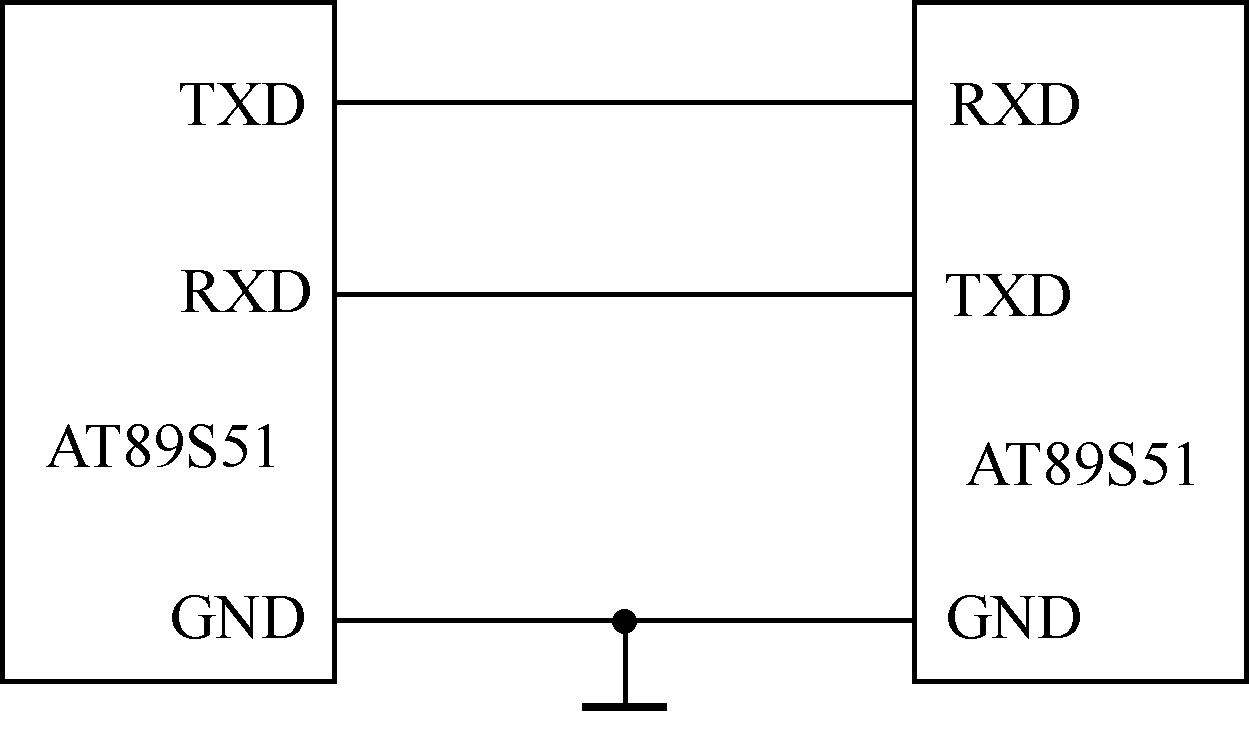


图2.-4 方式1双机串行通信方式

当SM0、SM1=01时，串行口设为方式1双机串行通信。TXD脚和RXD脚分别用于发送和接收数据。

　方式1收发一帧数据为10位，1个起始位（0），8个数据位，1个停止位（1），先发送或接收最低位。

1．方式1发送

　 串口以方式1输出，数据位由TXD端输出，发送一帧信息为10位，1位起始位0，8位数据位（先低位）和1位停止位1，当CPU执行写数据到发送缓冲器SBUF的命令后，就启动发送。

发送时钟TX时钟频率就是发送波特率。发送开始时，内部逻辑将起始位向TXD脚（P3.1）输出，此后每经1个TX时钟周期，便产生1个移位脉冲，并由TXD脚输出1个数据位。8位全发送完后，中断标志位TI置“1”。

2．方式1接收

　串行口以方式1（SM0、SM1=01）接收时（REN=1），数据从RXD（P3.0）脚输入。当检测到起始位负跳变时，则开始接收。

接收时，定时控制信号有两种，一种是接收移位时钟（RX时钟），频率和传送的波特率相同，另一种是位检测器采样脉冲，它的频率是RX时钟的16倍。也就是在1位数据期间，有16个采样脉冲，以波特率的16倍速率采样RXD引脚状态。

当采样到RXD端从1到0的负跳变（有可能是起始位）时，就启动接收检测器。接收的值是3次连续采样（第7、8、9个脉冲时采样），取其中两次相同的值，以确认是否是真正起始位（负跳变）开始，这样能较好消除干扰引起的影响，以保证可靠无误地开始接收数据。

当确认起始位有效时，开始接收一帧信息。接收每一位数据时，也都进行3次连续采样（第7、8、9个脉冲时采样），接收的值是3次采样中至少两次相同的值，以保证接收到的数据位的准确性。当一帧数据接收完毕后，必须同时满足以下两个条件，这次接收才真正有效。

（1）RI=0，即上一帧数据接收完成时，RI=1发出的中断请求已被响应，SBUF中的数据已被取走，说明“接收SBUF”已空。

（2）SM2=0或收到的停止位=1（方式1时，停止位已进入RB8），则将接收到的数据装入SBUF和RB8（装入的是停止位），且中断标志RI置“1”。

　 若不同时满足这两个条件，收到的数据不能装入SBUF，这意味着该帧数据将丢失。

**方式2**

串口工作于方式2和方式3时，为9位异步通信接口。每帧数据均为11位，1位起始位0，8位数据位（先低位），1位可程控为1或0的第9位数据及1位停止位。

1．方式2发送

　 发送前，先由通信协议由软件设置TB8（如奇偶校验位或多机通信的地址/数据的标志位），然后将要发送的数据写入SBUF，即可启动发送过程。串行口能自动把TB8取出，并装入到第9位数据位的位置，再逐一发送出去。发送完毕，则使TI位置“1”。

2．方式2接收

　当SCON寄存器SM0、SM1=10，且REN=1时，允许串行口以方式2接收数据。接收时，数据由RXD端输入，接收11位信息。当位检测逻辑采样到RXD引脚从1到0的负跳变，并判断起始位有效后，便开始接收一帧信息。在接收完第9位数据后，需满足以下两个条件，才将接收到的数据送入接收缓冲器SBUF。

　（1）RI=0，意味着接收缓冲器为空。

　（2）SM2=0或接收到的第9位数据位RB8=1。

　当满足上述两个条件时，接收到的数据送入SBUF（接收缓冲器），第9位数据送入RB8，且RI置“1”。若不满足这两个条件，接收的信息将被丢弃。

**方式3**

　 当SM0、SM1两位为11时，串行口被定义工作在方式3。方式3为波特率可变的9位异步通信方式，除了波特率外，方式3和方式2相同。

##### 三、系统硬件部分设计

3.1 Protel DXP电路原理图

见附录一

3.2 LCD显示电路

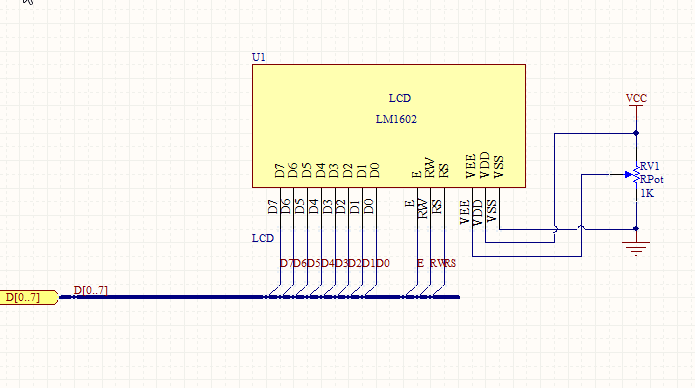


图3-1 LCD显示电路

3.2.1 LCD 1602引脚

LCD 1602工作电压4.5~5.5V，典型5V，工作电流2mA。标准的14引脚（无背光）或16个引脚（有背光）的外形及引脚如图3-2所示。

图3-2 LCD 1602引脚功能图

3.2.2．LCD1602字符的显示及命令

（1）显示字符

首先要解决待显示字符的ASCII码产生。用户只需在C51程序中写入欲显示的字符常量或字符串常量，C51程序在编译后会自动生成其标准的ASCII码，然后将生成的ASCII码送入LCD，内部控制电路就会自动将该ASCII码对应的字符在LCD1602显示出来。

让液晶显示器显示字符，首先对其进行初始化设置，还必须对有、无光标、光标移动方向、光标是否闪烁及字符移动方向等进行工作方式设置，才能获得所需显示效果。

（2）命令

通过控制引脚进行命令控制，四种状态，每种状态都是在E的脉冲下按完成。如表3-1所示

|  |  |  |
| --- | --- | --- |
| RS | R/ | 操作 |
| 0 | 0 | 写命令（初始化、光标位置设置等） |
| 1 | 0 | 写数据 |
| 0 | 1 | 读状态（液晶忙状态） |
| 1 | 1 | 读数据（把显示的数据反读出来） |

表3-1 LCD 1602状态

3.3 HC-08蓝牙模块电路

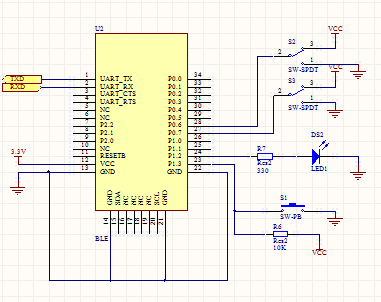


图3-3 HC-08蓝牙模块电路

3.3.1 模块简介

HC-08 蓝牙串口通信模块是新一代的基于 Bluetooth Specification V4.0 BLE 蓝牙协议的数传模块。无线工作频段为 2.4GHz ISM，调制方式是 GFSK。模块最大发射功率为4dBm，接收灵敏度-93dBm，空旷环境下和 iphone4s 可以实现 80 米超远距离通信。

模块采用 TI 的 CC2540 芯片，配置 256K Byte 空间，支持 AT 指令，用户可根据需要更改角色（主、从模式）以及串口波特率、设备名称等参数，使用灵活。

3.3.2 HC-08蓝牙引脚定义

通常情况下，只需要用到1脚、2脚、12脚、13脚，引脚定义如表3-2。

|  |  |  |  |
| --- | --- | --- | --- |
| 引脚 | 定义 | I/O 方向 | 说明 |
| 1 | TXD | 输出 | URAT 输出口，3.3V TTL 电平 |
| 2 | RXD | 输入 | URAT 输入口，3.3V TTL 电平 |
| 12 | VCC | 电源脚 | 电源脚，要求直流 3.3V 电源，供电电流不小于 100mA |
| 13 | GND |  | 模块公共地 |

表3-2 HC-08引脚定义

##### 3.4 温度检测电路

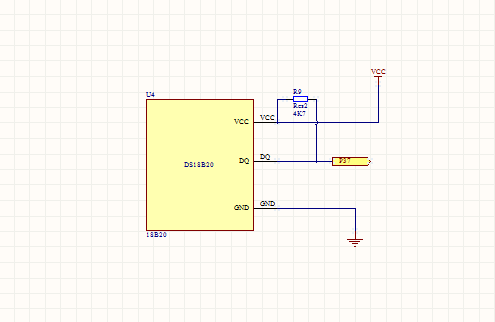


图3-4 温度检测电路

3.4.1 DS18B20模块简介

DS18B20是美国DALLAS公司推出的数字温度传感器，将温度传感器、数字转换电路集成到了一起。直接将温度转化成数字信号传送给单片机处理，因而可省去传统的信号放大、A/D转换等外围电路。

3.4.2 引脚功能

引脚功能定义如下：

DQ：数据输入输出，可直接与单片机的I/O口相连。

VDD：+5V电源电压。

GND：电源地。

3.4.3 编程方式

DS18B20内部有 9个字节的暂存器。第2、3、4个字节数据为存放于EEPROM数据的镜像。

开始两个暂存器存放当前测到的温度值，以16位补码形式表示12位温度读数。16位温度转换值的存放格式，高5位是温度值的符号扩展，中间7位是温度值的整数部分，最低4位为小数部分。

如果测得的温度大于0，高5位为0，只要将测到的数值乘以0.0625（1/16）即可得到实际温度；如果温度小于0，高5位为1，测到的数值需要取反加1再乘以0.0625即可得到实际温度。温度与转换后的数字量的对应关系见下表所示。

|  |  |  |
| --- | --- | --- |
| **度/℃** | **16位二进制编码** | **十六进制表示** |
| **＋125** | **0000 0111 1101 0000** | **07D0H** |
| **＋85** | **0000 0101 0101 0000** | **0550H** |
| **+25.0625** | **0000 0001 1001 0001** | **0191H** |
| **+10.125** | **0000 0000 1010 0010** | **00A2H** |
| **+0.5** | **0000 0000 0000 1000** | **0008H** |
| **0** | **0000 0000 0000 0000** | **0000H** |
| **-0.5** | **1111 1111 1111 1000** | **FFF8H** |
| **-10.125** | **1111 1111 0101 1110** | **FF5EH** |
| **-25.0625** | **1111 1110 0110 1111** | **FE6FH** |
| **-55** | **1111 1100 1001 0000** | **FC90H** |

表3-3 温度与转换后的数字量的对应关系

3.5 模拟加热电路（本设计中以LED灯和继电器模拟加热电路）

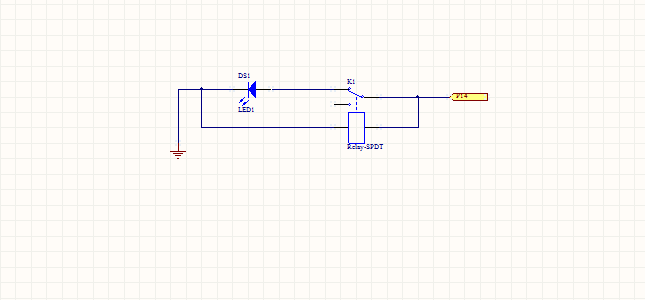


图3-5 加热电路图

当单片机接收到命令0时，继电器吸合，加热电路开始工作；当单片机接收命令1时，继电器断开，加热电路停止工作

##### 四、系统软件部分设计

4.1系统软件流程图

主函数设置定时器为方式2，然后运行头lcd1602文件中的LCD初始化函数，串口初始化函数，以及ds18b20头文件中的读取温度函数，再通过数组显示温度，见图4-1。

定时器中断函数采用if选择结构，当sbuf接收到ASCII码0x30，则点亮LED，0x31熄灭LED，0x32就把温度以数组的依次送回手机界面。见图4-2



图4-1主函数流程图 图4-2 定时器中断函数流程图

4.2 程序清单

见附录二

##### 五、系统仿真及实物制作

##### 5-4 Proteus仿真原理图

见附录三

##### 附录一 Protel DXP原理

##### 附录二 程序清单

#include <reg52.h>

#include"LCD1602.h"

#include"ds1820.h"

#define PINLV 11059200UL //使用11.0592M晶体

#define BAUD\_9600 9600UL //波特率定义为9600

sbit LED = P1^4;

#define uchar unsigned char

#define uint unsigned int

uchar fh,bw,sw;

int i;

uint ty1,temp,tempx,tempi;

uchar displaytemp[16] = "temp ";//第一行

void UART\_INIT(void); //函数声明

void delays (unsigned int us)

{

while(us--);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 串口初始化子函数

\* T1工作在方式2，波特率9600，开串口中断

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void UART\_INIT(void)

{

TMOD=0x20; //定时器1工作在模式2，自动重装模式

SCON=0x50; //串口工作在模式1

TH1=256-PINLV/(BAUD\_9600\*12\*16); //计算定时器重装值

TL1=256-PINLV/(BAUD\_9600\*12\*16);

PCON|=0x80; //串口波特率加倍

ES=1; //串行中断允许

TR1=1; //启动定时器1

EA=1; //允许中断

}

void DISPLAY()

{

displaytemp[4]=':';

displaytemp[5]=fh;

bw=tempi/100;

if(bw==0)

{displaytemp[6]=' ';}

else

{displaytemp[6]=bw+0x30;}

sw=(tempi%100)/10;

if(sw==0)

{displaytemp[7]=' ';}

else

{displaytemp[7]=sw+0x30;}

displaytemp[8]= (tempi%100)%10+0x30;

displaytemp[9]='.';

displaytemp[10]=tempx/10+0x30;

displaytemp[11]= 0xdf;

displaytemp[12]= 0x43;

LCD\_Write\_String(0,1,displaytemp);//显示第二行

}

//主程序

void main()

{ P1=0x00;

LCD\_initial();

UART\_INIT(); //串口初始化

while(1)

{

temp=retemp();

ty1=temp;

if(ty1>0xf000)

{

fh='-';

temp=~temp;

temp+=1;

}

else

{

fh='+';

}

tempi =temp>>4;

tempx=(temp&0x000f)\*6.25;

DISPLAY();

delays(10);//显示 //原地踏步

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 中断服务子函数

\* 清除RI，同时判断接到的数据

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void uart(void) interrupt 4

{

unsigned char n; //接收数据寄存器

if(RI) //是否收到数据

{

RI=0; //清中断请求

n=SBUF; //读入缓冲区的值

/\* switch(n)

{

case 0X30:LED=1;break;

case 0X31:LED=0;break;

case 0X32:for(i=0;i<11;i++)

{n=displaytemp[i];

while(TI==0);

TI=0;

delays(100000);}break;

} \*/

if(n==0x30)

LED=1;

if(n==0x31)

LED=0;

if(SBUF==0x32)

{

for(i=0;i<11;i++)

{

SBUF=displaytemp[i];

while(TI==0);

TI=0;

delays(100000);

}

}

}

}

##### 附录三 Proteus 仿真原理图