

蔬菜大棚温湿度采集系统的设计毕业论文

目 录

引 言.....	错误!未定义书签。.....
1 系统总体设计	2.....
1.1 系统工作原理	
1.2 系统组成.....	
1.3 系统性能指标	
2 硬件电路设计	5.....
2.1 设计原则.....	
2.2 系统硬件总体结构	
2.3 单片机的最小系统设计	
2.4 DHT11 温湿度传感器	
2.5 串口的简介及作用	
2.6 单片机与 TQ2440 开发板的串口通信	
3 软件设计.....	11.....
3.1 设计原则.....	
3.2 单片机软件设计	
3.3 TQ2440 开发板的软件设计.....	
3.3.1 Qt 的介绍	
3.3.2 串口软件设计	
3.3.3 TQ2440 开发板网络连接的设计	
3.4 网络服务器端软件设计	
3.5 数据库设计.....	
4 系统连接和测试	20.....
5 设计总结与分析	23.....
致谢.....	错误!未定义书签。.....
参考文献.....	24.....
科技外文文献.....	25.....
附录 A: 源程序代码	35.....
附录 B: 系统使用说明书	42.....

1 系统总体设计

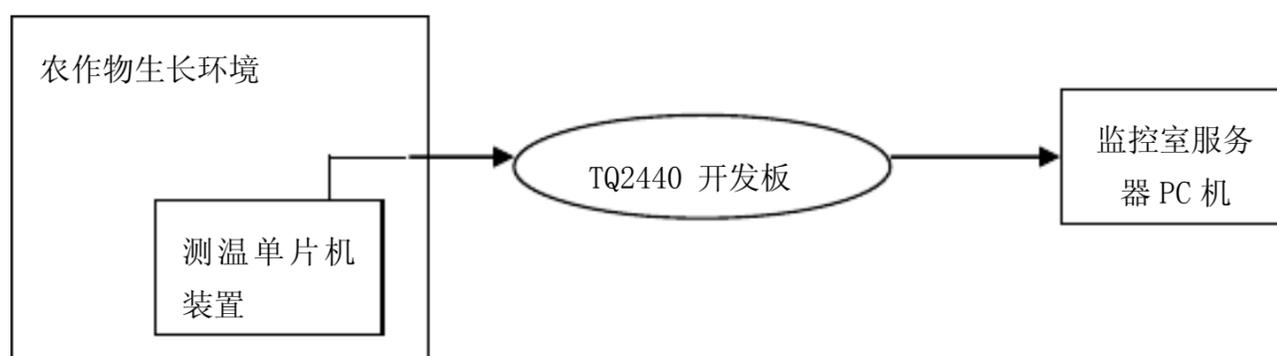
本系统主要针对农作物生长环境,设计了以 PC 机为服务器,天嵌 TQ2440 开发版为远程终端,单片机为下位机的农作物环境的网络监控系统。综合考虑系统的精度、效率以及经济性要求这三个方面之后,最终确定下位机以 AT89C52 单片机为采集端,选用性价比比较高的 DHT11 温湿度传感器模块,实现对温度精确测量与准确控制。当单片机检测到温度和湿度值时,则通过 RS-232 串口发送至 TQ2440 开发版。TQ2440 通过实现串口与单片机通讯,通过实现 TCP/IP 协议实现与远程服务器通讯。为了便于系统的软件设计、调试、移植、修改,单片机采集端软件以 C 语言为基础,采用模块化设计,主要包括单片机的最小系统、数据采集模块、串口发送模块以及串行通讯模块。TQ2440 开发板使用 Linux 内核并使用 QT 文件系统,以 C++语言为基础,采用面向对象方法设计,使用 TCP/IP 协议传送数据给服务器。服务器使用 Qt4.7 编写温湿度监控界面,使用 MySQL 数据库存储温湿度信息。

1.1 系统工作原理

单片机首先通过传感器 DHT11 采集农作物生长环境的温湿度信息,再利用单片机的串口进行编程,单片机模块通过 RS232 接口与 TQ2440 开发版相连,将传感器测得的温室度数据传送到 TQ2440 开发板;TQ2440 开发板使用 TCP/IP 协议传送数据给服务器。在 PC 机中,利用面向对象编程,让 PC 机和 TQ2440 开发板通过 Internet 进行数据传输,同时 PC 机界面显示作物生长环境的实时信息,实现对大棚温湿度实时的测控,保证了人类在远程可以得到农作物的生长环境信息。

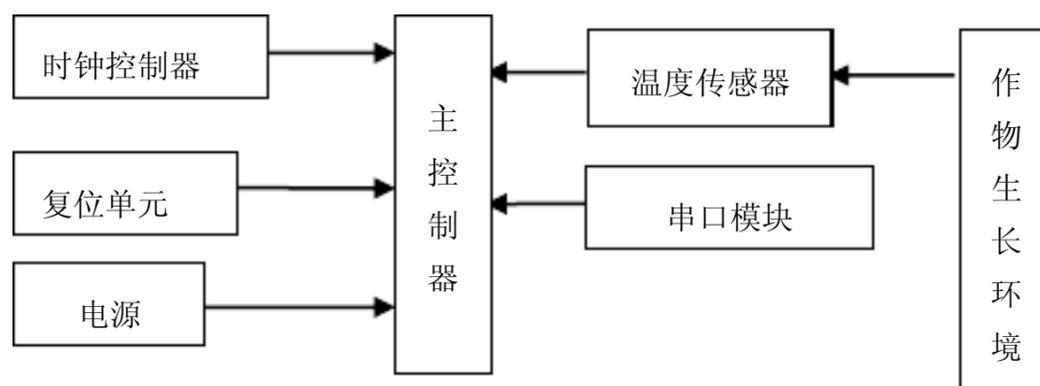
1.2 系统组成

整个无线监测系统主要分为三部分:即温度检测、无线传输和 PC 机对温度的监测环节。如图 1.1 所示。



(1) 温湿度检测模块的组成

在温度检测中,由单片机 AT89C52 主控制器所组成的最小系统以及外部接口模块主要有温湿度传感器(DHT11),各模块连接如下图 1.2 所示。



(2) TQ2440 开发板接入 Internet 组成模块

TQ2440 开发版硬件主要由 ARM9 处理器、RAM、Flash 等组成。系统软件采用 linux 内核,文件系统采用 Qt4.0。在本设计中,TQ2440 开发版接收到串口数据,然后与服务器通过 TCP/IP 协议建立网络连

接，接着向服务器传输温湿度数据，数据传输完毕，断开网络连接。模块的连接框图如图1.3所示。

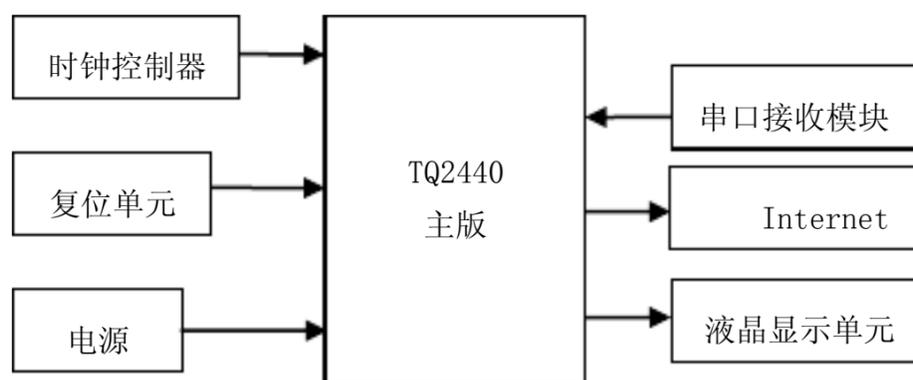


图 1.3 TQ2440 开发版连接模块框图

(3) PC 机服务器端温湿度的实时显示与历史数据查询

PC 机服务器端接收到 TQ2440 开发版传输过来的数据，将环境实时数据存入数据库。并以友好界面将农作物生长环境温湿度的实时信息显示，PC 上还可以查询作物生长环境历史数据，可以选择查询时间范围和显示方式，可以以二维表方式或者是温湿度曲线变化图方式。PC 机服务器端温湿度的实时显示与历史数据查询模块如图 1.4 所示。

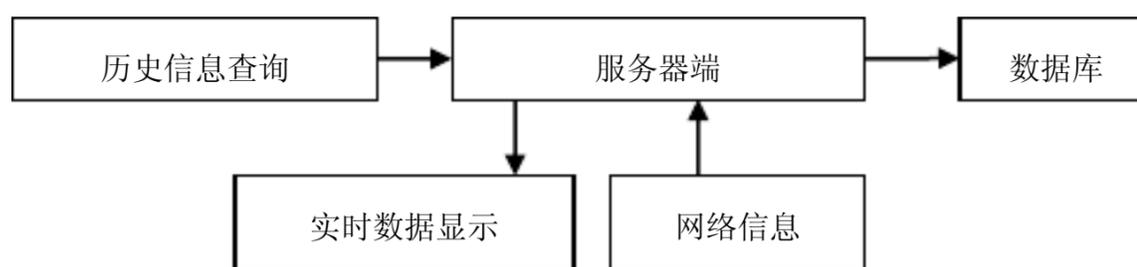


图 1.4 PC 机服务器端监测模块图

(4) 系统设计流程

系统的总体设计分为硬件和软件设计两方面，首先确定系统实现的功能，然后对硬件、软件分别进行规划，完成这些准备工作之后，就可以开始制作硬件电路，编写软件程序，在模块化调试结束后，进行软硬件联调，针对出现的问题对软硬件进行相应的修改，直到调试成功为止。系统的总体设计流程图如图 1.5 所示。

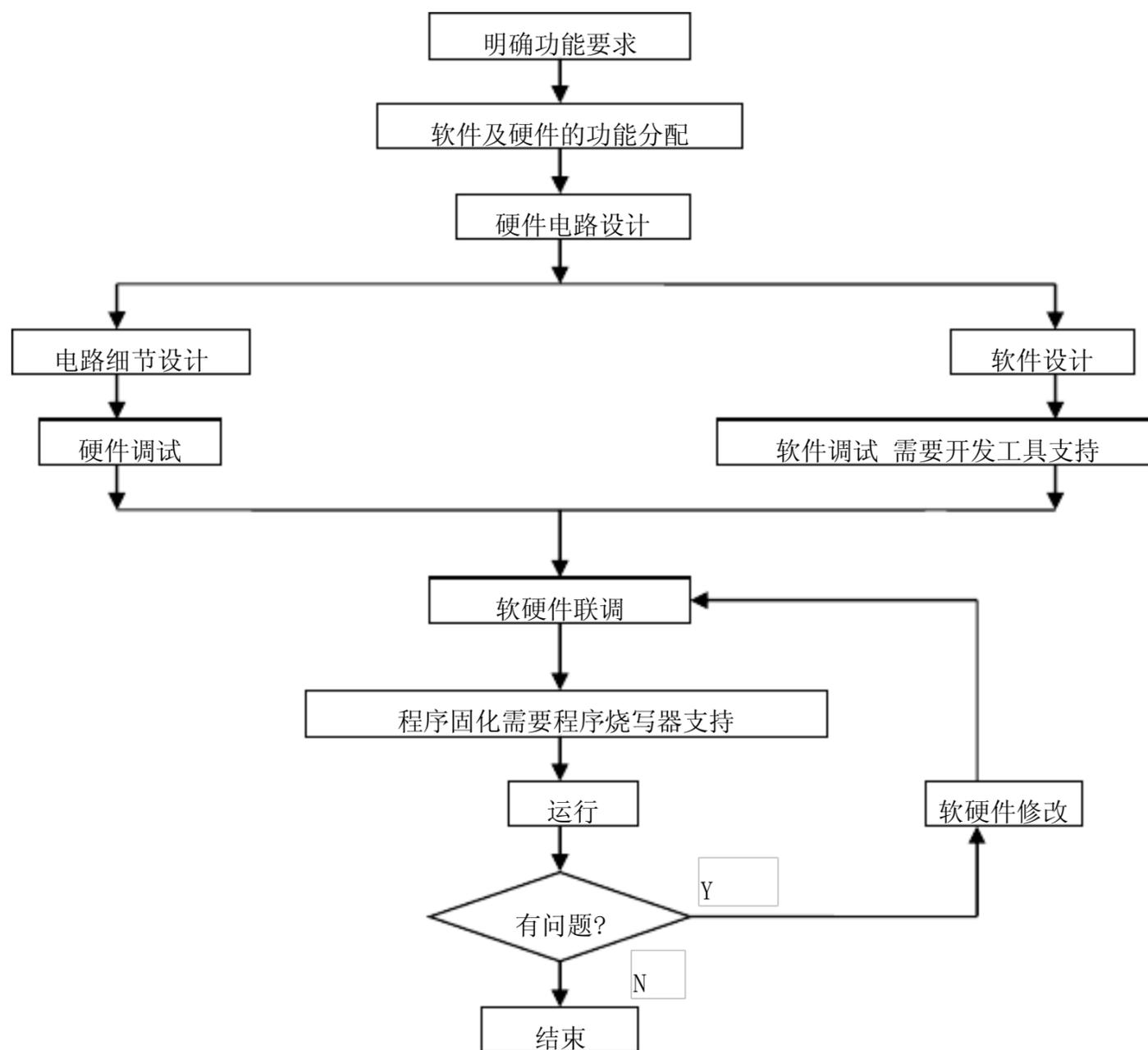


图 1.5 系统总体设计流程图

1.3 系统性能指标

本系统具有良好的可靠性和经济性，能够实现对作物生长环境准确测量和控制，在实际应用中有一定价值。具体性能指标分述如下。

- (1) 温度范围：0℃~+50℃；
- (2) 温度精度：2℃；
- (3) 湿度范围：20-90%RH；
- (4) 湿度精度：5%RH；

2 硬件电路设计

2.1 设计原则

- (1) 尽可能选择典型电路，并符合单片机的常规用法。为硬件电路的标准化、模块化打下良好基础。可靠性和抗干扰设计是硬件设计必不可少的一部分，它包括芯片和器件的选择、去耦电容、滤波电容、电路板的布线等。
- (2) 尽量朝单片方向设计硬件。硬件器件越多，器件之间相互干扰越强，功耗也会越大，就会不可避免的降低系统的稳定性。
- (3) 在速度允许的情况下，尽量使用串行为主的扩展方式。串行扩展具有方便、灵活、电路简单、占用 I/O 资源少等特点。
- (4) 留下一些指示灯或通信口以方便调试和判别系统问题。

2.2 系统硬件总体结构

监控系统的硬件部分按功能划分为三个部分，主控制部分，信号采集部分和通信部分。硬件结构图如图 2.1 所示。

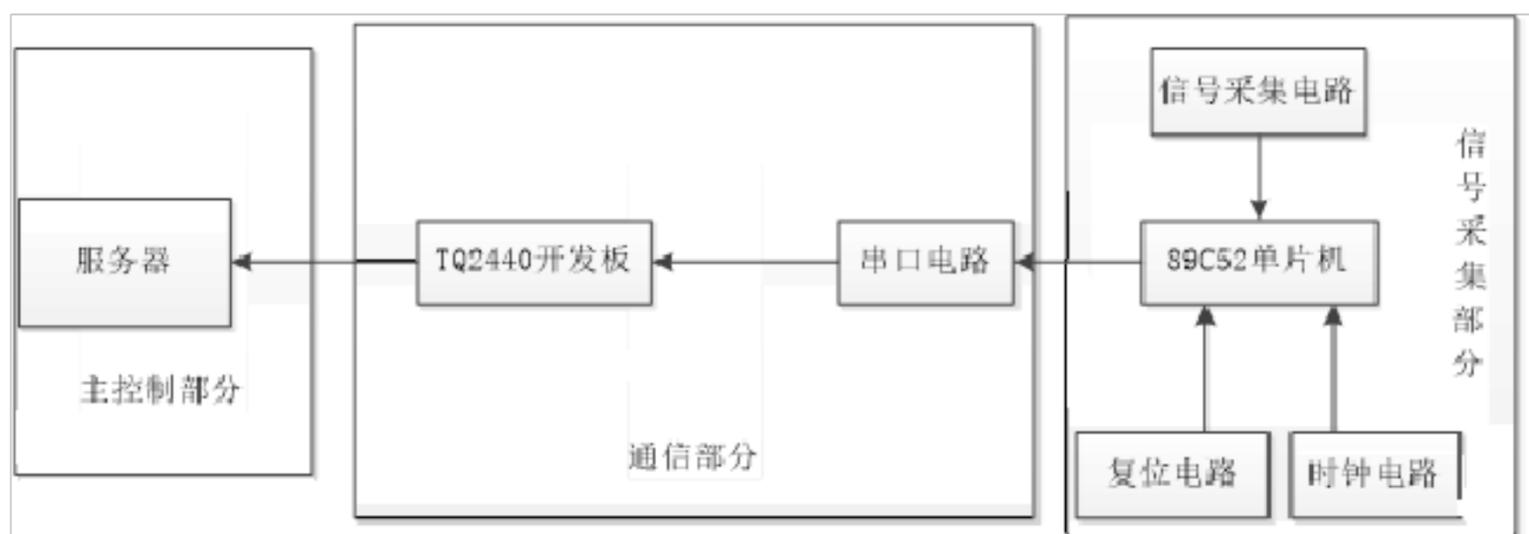


图 2.1 硬件结构图

2.3 单片机的最小系统设计

在本设计中采用了 AT89C52 单片机作为核心处理器，因此在电路中首先设计的是 AT89C52 的最小系统。

AT89C52 单片机的最小系统包含以下几部分。

(1) 单片机供电电路：AT89C52 需要可靠的 5V 供电，在电路图中的 VCC 和 GND 为供电网络标识符；

振荡电路：AT89C52 需要一个稳定的振荡电路才能够正常工作，单片机的时钟信号是用来提供单片机内各种微操作的基准。在该电路中采用了 12MHz 的晶振作为 AT89C52 的时钟源；这里采用的是内部振荡方式，在引脚 XTAL1 和 XTAL2 外接晶振，通过内部振荡得到的时钟信号比较稳定，在电路中使用较多。在下面的电路图中可以看到在晶振两侧连了两个电容 C2, C3，它们是起稳定振荡频率、快速起振的作用，电容值一般为 5~30pF。本设计中用的是 30pF 的电容。

(2) 复位电路：复位电路是单片机正常运行的一个必要部分。复位操作一般有两种基本形式：上电复位和开关复位。在本设计中采用的是第二种。复位电路应该保证单片机在上电的瞬间进行一次有效的复位，在单片机正常工作时将 RST 引脚置低。此外通过一个按键进行手动复位，在单片机运行不正常时使用。上电后，由于电容充电，是 RST 持续一段高电平时间。当单片机已经在运行时，按下复位键也能使 RST 持续一段高电平，从而实现上电且开关复位的操作。通常我们选择的复位电容为 10~50μF，电阻为 1~10k。在本设计中复位电容选的是 47μF 的，电阻选的是 10k 的。

AT89C52 的最小系统电路如图 2.1 所示。

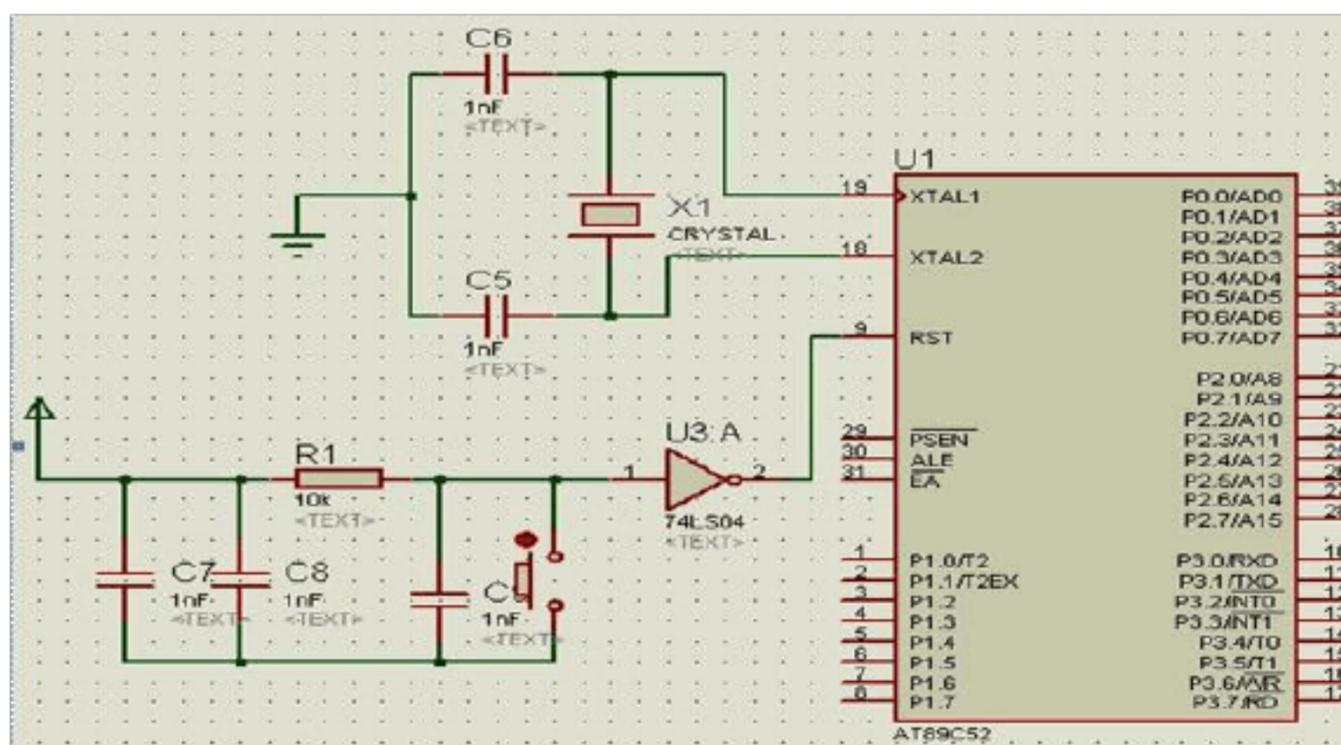


图2.1 AT89C52的最小系统电路

2.4 DHT11 温湿度传感器

在选择温湿度传感器时，应考虑的主要因素有温湿度的测量范围、精度、测温时间、稳定性、灵敏度和经济性。

(1) DHT11 产品概述

DHT11 数字温湿度传感器是一款含有已校准数字信号输出的温湿度复合传感器。它应用专用的数字模块采集技术和温湿度传感技术，确保产品具有极高的可靠性与卓越的长期稳定性。传感器包括一个电阻式感湿元件和一个 NTC 测温元件，并与一个高性能 8 位单片机相连接。因此该产品具有品质卓越、超快响应、抗干扰能力强、性价比极高等优点。每个 DHT11 传感器都在极为精确的湿度校验室中进行校准。校准系数以程序的形式储存在 OTP 内存中，传感器内部在检测信号的处理过程中要调用这些校准系数。单线制串行接口，使系统集成变得简易快捷。超小的体积、极低的功耗，信号传输距离可达 20 米以上，使其成为各类应用甚至最为苛刻的应用场合的最佳选则。产品为 4 针单排引脚封装。连接方便，特殊封装形式可根据用户需求而提供。

(2) 传感器性能说明如表2.1所示。

表2.1 传感器性能说明

参数	条件	Min	Typ	Max	单位
湿度	分辨率	1	1	1	%RH
			8		Bit
重复性	精度	25℃	±1	±5	%RH
			0—50℃		%RH
互换性		可完全互换			
量程范围	0℃	30		90	%RH
	25℃	20		90	%RH
	50℃	20		80	%RH
响应时间	1/e (63%) 25℃, 1m/s 空气	6	10	15	S
迟滞			±1		%RH

长期稳定性	典型值	±1	%RH/yr
-------	-----	----	--------

续表 2.1

参数	条件	Min	Typ	Max	单位
温度					
分辨率		1	1	1	°C
		8	8	8	Bit
重复性			±1		°C
精度		±1		±2	°C
量程范围		0		50	°C
响应时间	1/e (63%)	6		30	S

(3) 接口说明

建议连接线长度短于 20 米时用 5K 上拉电阻, 大于 20 米时根据实际情况使用合适的上拉电阻, 接口如图 2.2 所示。

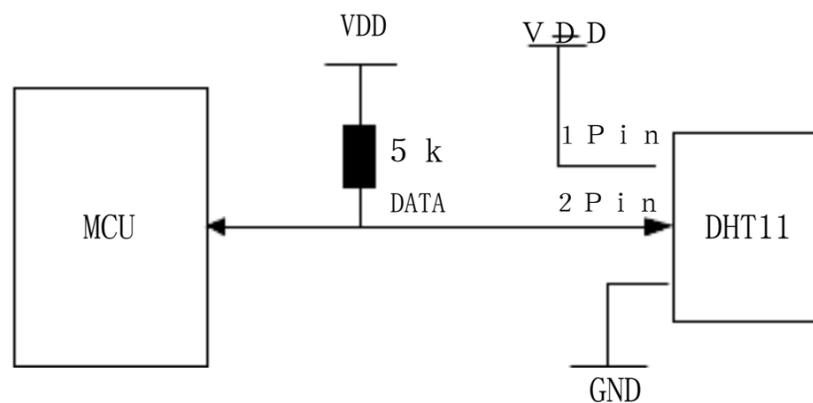


图2.2 DHT11接口图

(4) 电源引脚

DHT11 的供电电压为 3—5.5V。传感器上电后, 要等待 1s 以越过不稳定状态在此期间无需发送任何指令。电源引脚 (VDD, GND) 之间可增加一个 100nF 的电容, 用以去耦滤波。

(5) 串行接口 (单线双向)

DATA 用于微处理器与 DHT11 之间的通讯和同步, 采用单总线数据格式, 一次通讯时间 4ms 左右, 数据分小数部分和整数部分, 具体格式在下面说明, 当前小数部分用于以后扩展, 现读出为零。操作流程如下:

一次完整的数据传输为 40bit, 高位先出。

数据格式: 8bit 湿度整数数据+8bit 湿度小数数据+8bit 温度整数数据+8bit 温度小数数据+8bit 校验和

数据传送正确时校验和数据等于“8bit 湿度整数数据+8bit 湿度小数数据+8bit 温度整数数据+8bit 温度小数数据”所得结果的末 8 位。

用户 MCU 发送一次开始信号后, DHT11 从低功耗模式转换到高速模式, 等待主机开始信号结束后, DHT11 发送响应信号, 送出 40bit 的数据, 并触发一次信号采集, 用户可选择读取部分数据。从模式下, DHT11 接收到开始信号触发一次温湿度采集, 如果没有接收到主机发送开始信号, DHT11 不会主动进行温湿度采集。采集数据后转换到低速模式。通讯过程如图 2.3 所示。

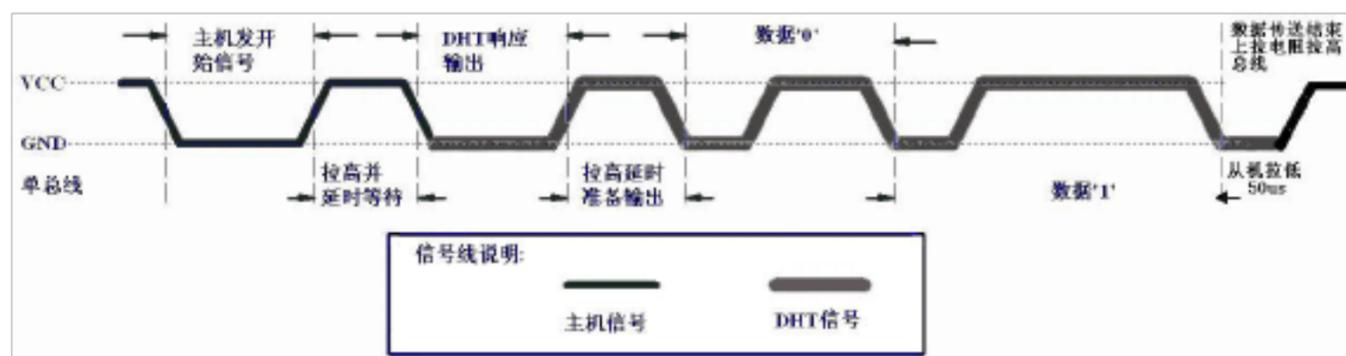


图2.3 通讯过程时序图

总线空闲状态为高电平，主机把总线拉低等待 DHT11 响应，主机把总线拉低必须大于 18 毫秒，保证 DHT11 能检测到起始信号。DHT11 接收到主机的开始信号后，等待主机开始信号结束，然后发送 80us 低电平响应信号。主机发送开始信号结束后，延时等待 20-40us 后，读取 DHT11 的响应信号，主机发送开始信号后，可以切换到输入模式，或者输出高电平均可，总线由上拉电阻拉高，如图 2.4 所示。

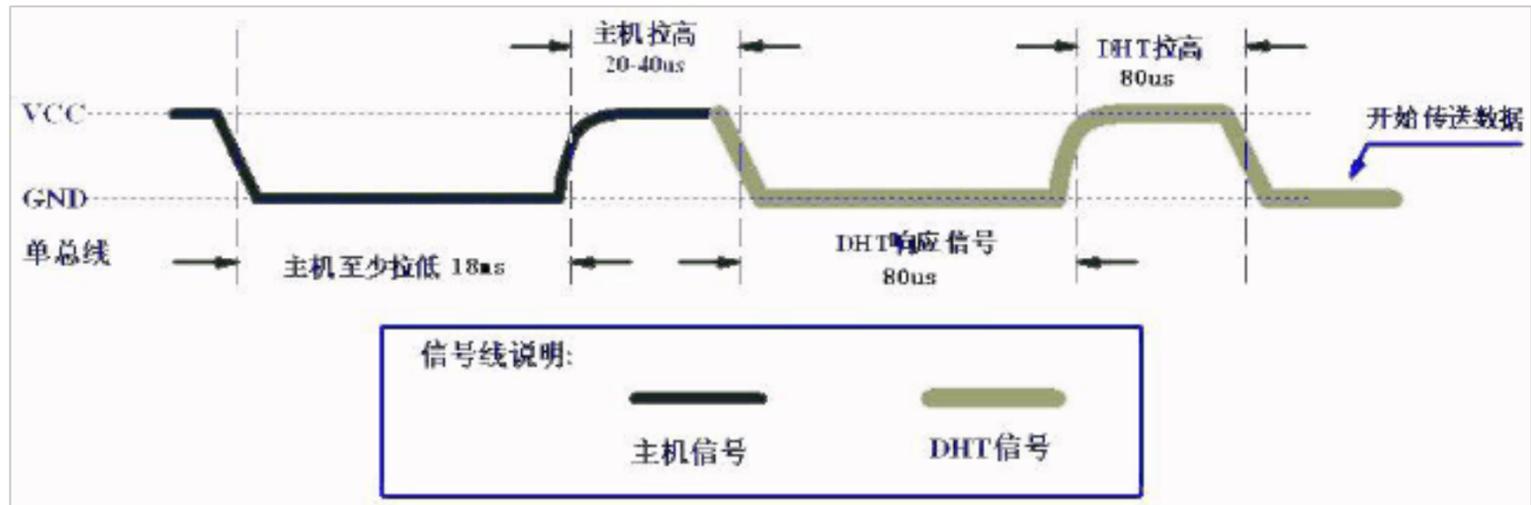


图2.4 DHT11响应主机时序图

总线为低电平，说明 DHT11 发送响应信号，DHT11 发送响应信号后，再把总线拉高 80us，准备发送数据，每一 bit 数据都以 50us 低电平时隙开始，高电平的长短定了数据位是 0 还是 1。格式见下面图示。如果读取响应信号为高电平，则 DHT11 没有响应，请检查线路是否连接正常。当最后一 bit 数据传送完毕后，DHT11 拉低总线 50us，随后总线由上拉电阻拉高进入空闲状态。

数字 0 信号表示方法如图 2.5 所示。

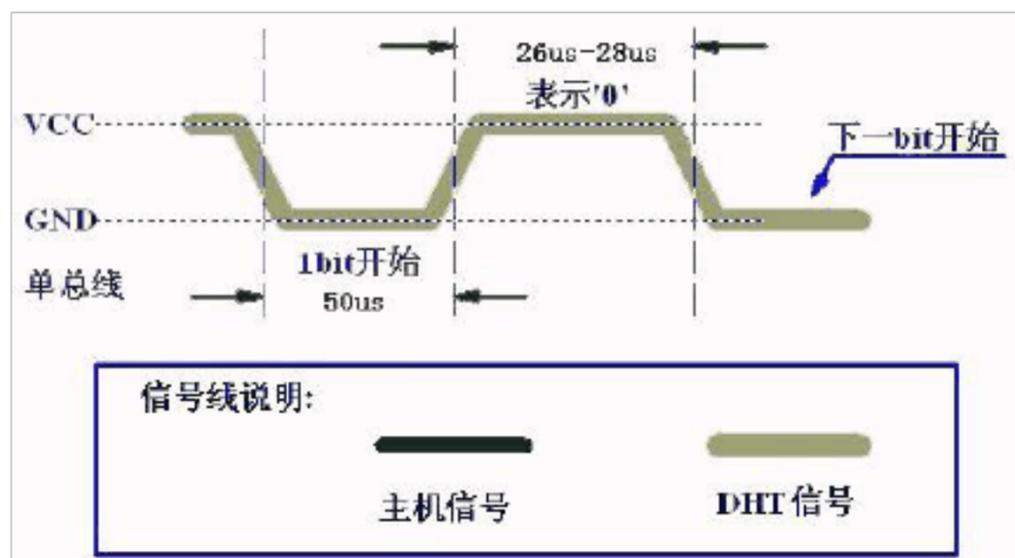


图 2.5 DHT11 发送 0 时序图

数字 1 信号表示方法，如图 2.6 所示。

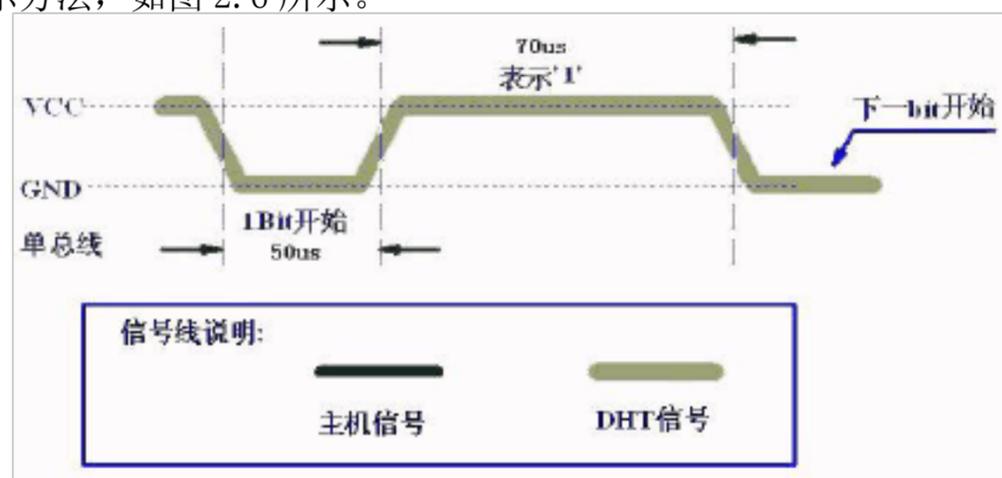


图2.6 DHT11发送1时序图

(5) 测量分辨率

测量分辨率分别为 8bit (温度)、8bit (湿度)。

2.5 串口的简介及作用

串行通信是在一根传输线上一位一位的传送信息，所用的传输线少，并且可以借助现成的电话网进行信息传送，因此，特别适合于远距离传输。对于那些与计算机相距不远的人—机交换设备和串行存储的外部设备如终端、打印机、逻辑分析仪、磁盘等，采用串行方式交换数据也很普遍。所以串行接口是微机应用系统常用的接口。许多外设和计算机按串行方式进行通信，这里所说的串行方式，是指外设与接口电路之间的信息传送方式，实际上，CPU 与接口之间仍按并行方式工作。

在单片机系统中，串口是一个非常重要的组成部分。通常使用单片机串口通过 RS232 接口和电平转换芯片 MAX232 与上位机连接，以进行上位机与下位机的数据交换、参数设置、组成网络以及各种外部设备的连接等。RS232 串行接口总线具有成本低、简单可靠、容易使用等特点，加上其历史悠久，所以目前应用仍然非常广泛；特别对于数据量不是很大的场合，串口通信仍然是很好的选择，有着广阔的使用前景。在单片机编程中，串口占了很重要的地位。

2.6 单片机与 TQ2440 开发板的串口通信

通信接口的选择：

为了便于 QT2440 开发板和各种外围设备的串行通信连接，更广义地来讲是为了各种数据终端设备 (DTE) 和数据通信设备 (DCE) 之间的连接，制定了若干种串行通信接口标准。只要是符合某种标准的设备之间就可以直接互相连接、互相通信。

串行通信接口按电气标准及协议来分包括 RS-232、RS-422、RS485、USB 等。RS-232、RS-422 与 RS-485 标准只对接口的电气特性做出规定，不涉及接插件、电缆或协议。USB 是近几年发展起来的新型接口标准，主要应用于高速数据传输领域。

在本设计中，选择 RS-232 接口就可以满足通信需求了。

目前 RS-232 是 PC 机与通信工业中应用最广泛的一种串行接口。RS-232 被定义为一种在低速率串行通信中增加通信距离的单端标准。RS-232 采取不平衡传输方式，即所谓单端通信。典型的 RS-232 信号在正负电平之间摆动，在发送数据时，发送端驱动器输出正电平在 +5~+15V，负电平在 -5~-15V 电平。当无数据传输时，线上为 TTL 电平，从开始传送数据到结束，线上电平从 TTL 电平到 RS-232 电平再返回 TTL 电平。完整的 RS-232 接口有 25 根线，采用一种 25 芯 (针) 的插头座，彼此连接十分方便。现在经常采用一种 9 针的插座来互相连接，因为 25 条线中最经常使用的只有 9 条线。由于一般的微机中都有 RS-232 接口，利用 RS-232 通信进行测量，连接、携带、运输方便。在那些临时、快速测量而测量的通道数又不多的场合下，利用 RS-232 接口的测量模块十分方便。其传送距离最大为约 15m，最高速率为 20kb/s。基本的数据传送引脚：

TXD：数据发送引脚；

RXD：数据接收引脚；

GND：信号地线；

在单片机通信中最简单的通信只需连这三根线。

由于单片机的串行发送线 TXD 和接收线 RXD 是 TTL 电平，而 ARM 开发板的 tq2440_serial1 的 RS-232C 连接器 (D 型 9 针插座) 是 EIA 电平，因此单片机需加接 MAX232 芯片，通过串行电缆线和 ARM 开发板相连接。单片机的串口电路图如图 2.7 所示。

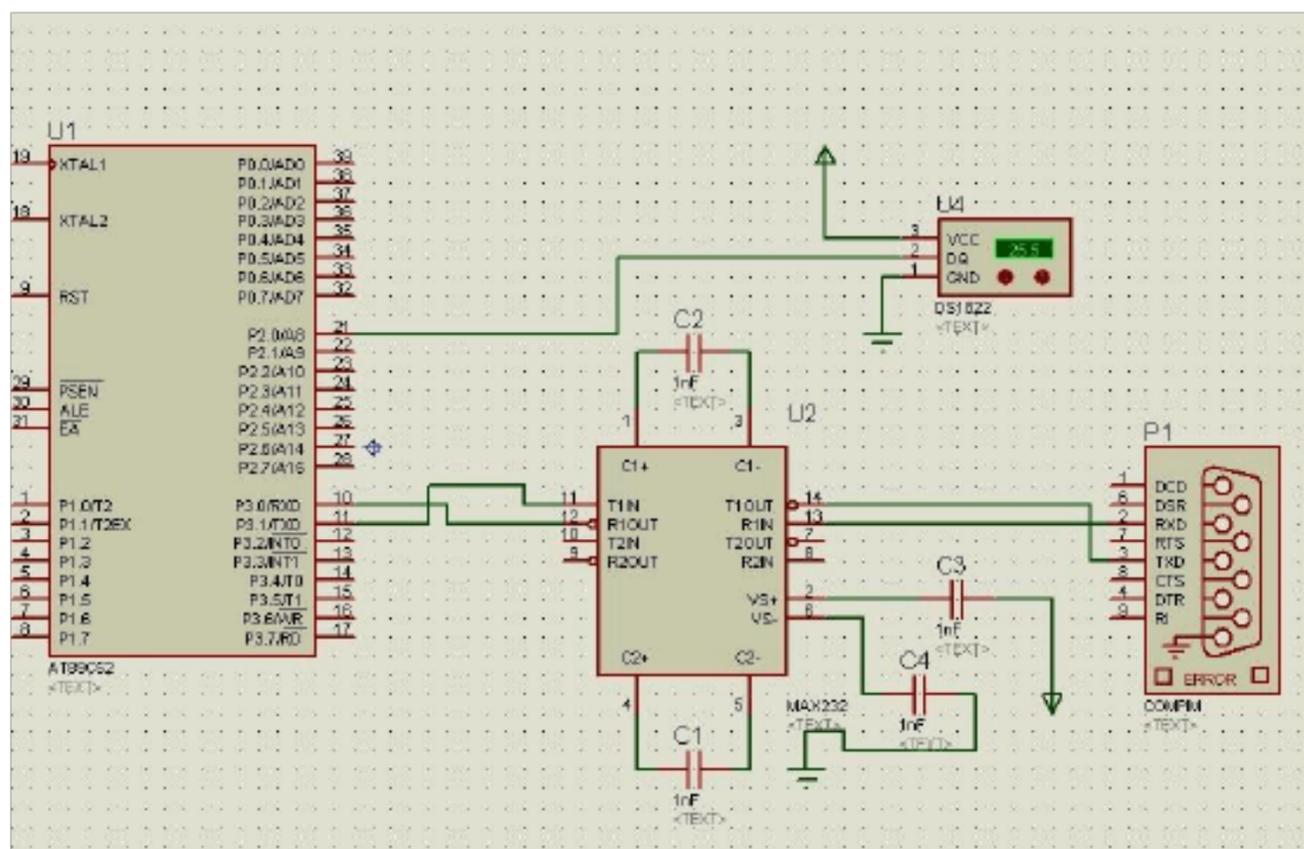


图 2.7 单片机串口电路图

RS-232 的 EIA 标准

它是以正负电压来表示逻辑状态，而 TTL 以高低电平来表示器逻辑状态。目前较广泛使用的转换芯片很多，但很多需要正负 12V 两种电源，使用不方便。而 MAXIM 公司的 MAX232 芯片之需要+5V 电源就可以实现 TTL 和 EIA 的双向电平转换，因此得到了广泛应用。

(2) 通信协议

在进行数据通信时，必须解决好两个方面的问题：一是可靠性，二是速度。可靠性是第一位的，速度只是在可靠的基础上的追求。可靠快速传输的实现，需要上、下位机软件以及通信协议等各个环节的可靠和相互配合。

在串行通信的硬件设计完成后，通信双方（在本系统中指单片机和上位机）必须约定通信协议，否则将无法保证通信数据的可靠性，从而失去通信的意义。协议一方面要规定通信的基本参数，如通信波特率、数据位数、停止位数及奇偶校验的方式等，更重要的一方面是要规定双方传输数据的格式，以及传输数据时控制数据流的方式。现约定系统的通信协议如下：

串行通信波特率为 4800bps；

数据传送格式为 1 个起始位，8 个数据位，1 个停止位；

无奇偶校验；

串行通信方式采用查询方式；

3 软件设计

3.1 设计原则

(1) 明确任务，弄清软件所承担的任务细节。

(2) 软件结构设计，合理的软件结构是设计出一个性能优良的单片机应用系统软件的基础。

模块化程序设计，是单片机应用中最常用的程序设计技术。将一个完整的程序分解成若干个功能相对独立的较小的程序模块，对各个程序模块分别进行设计、编制和调试，最后将各个调试好的程序模块进行联调。而面向对象程序设计的数据抽象可以在保持外部接口不变的情况下改变内部实现，从而减少甚至避免对外界的干扰；通过继承大幅减少冗余的代码，并可以方便地扩展现有代码，提高编码效率，也减低了出错概率，降低软件维护的难度；结合面向对象分析、面向对象设计，允许将问题域中的对象直接映射到程序中，减少软件开发过程中中间环节的转换过程。

(3) 编写程序。根据系统功能和操作过程，列出程序的功能流程图。在完成流程图的设计之后，便可编写程序了。

3.2 单片机软件设计

单片机采用 C52 在 keil uvision4 的开发环境进行编程，在仿真软件中调试成功后，再把生成的 HEX 文件烧到单片机中，在真实的硬件环境下进行测试。

具体设计主要分两部分，一是温湿度传感器的驱动和测温程序，另一个是利用单片机串口编写的无线传输程序。下面将对程序中的关键部分进行阐述。

温湿度传感器接口软件设计

单片机从 DHT11 完成温湿度数据读取要经过以下步骤：单片机发送一次开始信号后，DHT11 从低功耗模式转换到高速模式，等待主机开始信号结束后，DHT11 发送响应信号，送出 40bit 的数据，并触发一次信号采集，用户可选择读取部分数据。从模式下，DHT11 接收到开始信号触发一次温湿度采集，如果没有接收到主机发送开始信号，DHT11 不会主动进行温湿度采集。采集数据后转换到低速模式。流程图如图 3.1 所示。

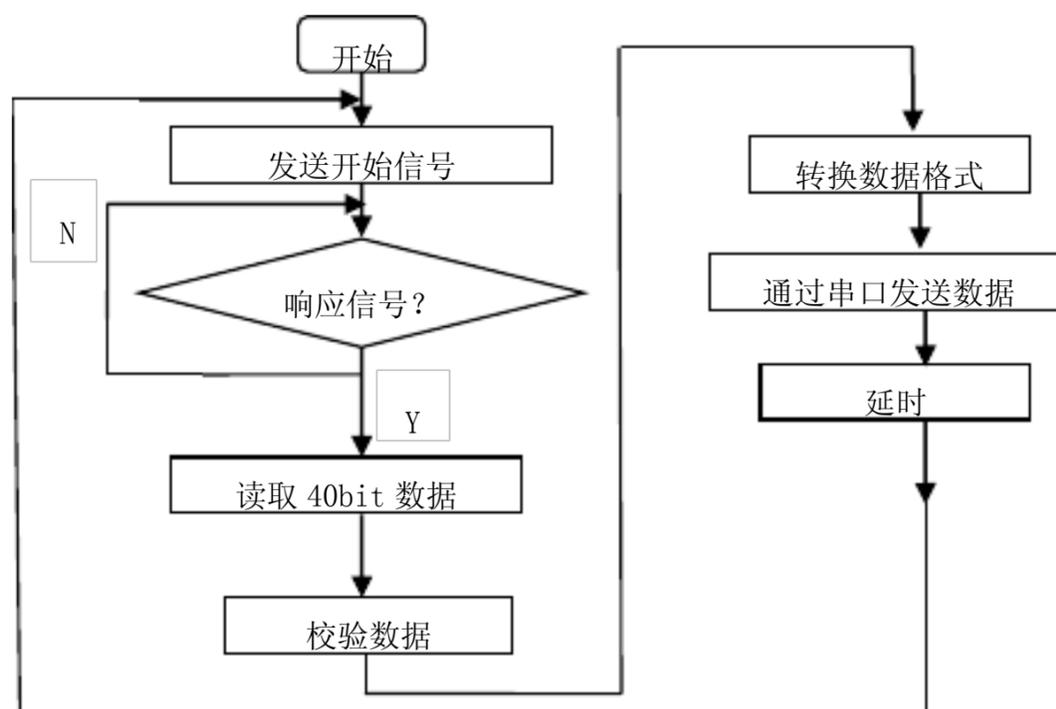


图 3.1 单片机软件设计的流程图

具体源码如下：

主函数如下：

```

void main()
{
    SCON = 0x50;

```

```

TMOD|= 0x20;
PCON|= 0x80;
TH1 = 0xF3;
TL1 = 0xF3;
TR1  = 1;
ES   = 1;
EA   = 1;
SendData(str) ;
Delay(1);
while(1)
{
    RH();
    str[0]=(U8RH_data_H/10)+48;
    str[1]=(U8RH_data_H%10)+48;
    str[2]=(U8T_data_H/10)+48;
    str[3]=(U8T_data_H%10)+48;
    str[4]=(U8RH_data_H/10+U8RH_data_H%10+U8T_data_H/10+U8T_data_H%10)%10+48;
    //读取模块数据周期不易小于 8S
    Delay(20000);
    Delay(20000);
    Delay(20000);
    Delay(20000);
    SendData(str);
} //elihw
} // main

```

延时函数

```

void Delay(U16 j) {
    U8 i;
    for(;j>0;j--){
        for(i=0;i<27;i++);
    }
}

```

读取传感器数据函数

```

void RH(void) {
    P2_0=0;
    Delay(180);
    P2_0=1;
    Delay_10us();
    Delay_10us();
    Delay_10us();
    Delay_10us();
    //主机设为输入 判断从机响应信号
    P2_0=1;
    //判断从机是否有低电平响应信号 如不响应则跳出，响应则向下运行
    if(!P2_0)    //T !
    {

```


和设计，开发版的软件采用 crosstools_3.4.5 交叉编译工具编译，服务器端采用 Qt 编译器编译。

3.3.1 Qt 的介绍

Qt 是一个 1991 年由奇趣科技开发的跨平台 C++ 图形用户界面应用程序开发框架。它既可以开发 GUI 程式，也可用于开发非 GUI 程式，比如控制台工具和服务器。Qt 是面向对象语言，易于扩展，并且允许组件编程。

3.3.2 串口软件设计

用 C++ 具体实现串口的通信，必须掌握 Qt 中对串口操作的方法，每种语言都提供了对串口读写操作。在这里我们使用 Qt 提供的 API 通信函数实现步骤是，它是先打开串口，在得到串口句柄后，再进行串口参数的配置。步骤如图 3.2 所示。

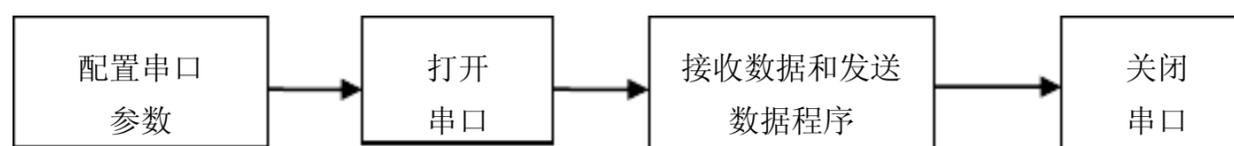


图 3.2 串口通信程序实现步骤

源码如下：

```

void Widget::on_openMyComBtn_clicked()
{
    myCom->open(QIODevice::ReadWrite);
    myCom->setBaudRate(BAUD4800);
    myCom->setDataBits(DATA_8);
    myCom->setParity(PAR_NONE);
    myCom->setStopBits(STOP_1);
    myCom->setFlowControl(FLOW_OFF);
    myCom->setTimeout(10);
    readTimer = new QTimer(this);
    readTimer->start(100);
    connect(readTimer, SIGNAL(timeout()), this, SLOT(readMyCom()));
    //信号和槽函数关联，延时一段时间，进行读串口操作
    ui->openMyComBtn->setEnabled(false);
    ui->closeMyComBtn->setEnabled(true);
    ui->sendMsgBtn->setEnabled(true);
}
  
```

3.3.3 TQ2440 开发板网络连接的设计

TCP 即 Transmission Control Protocol，传输控制协议。与 UDP 不同，它是面向连接和数据流的可靠传输协议。也就是说，它能使一台计算机上的数据无差错的发往网络上的其他计算机，所以当要传输大量数据时，我们选用 TCP 协议。TCP 协议的程序使用的是客户端/服务器模式，在 Qt 中提供了 QTcpSocket 类来编写客户端程序，使用 QTcpServer 类编写服务器端程序。我们在服务器端进行端口的监听，一旦发现客户端的连接请求，就会发出 newConnection() 信号，我们可以关联这个信号到我们自己的槽函数，进行数据的发送。而在客户端，一旦有数据到来就会发出 readyRead() 信号，我们可以关联此信号，进行数据的接收。面向连接的客户/服务器程序工作流程如图 3.4 所示。

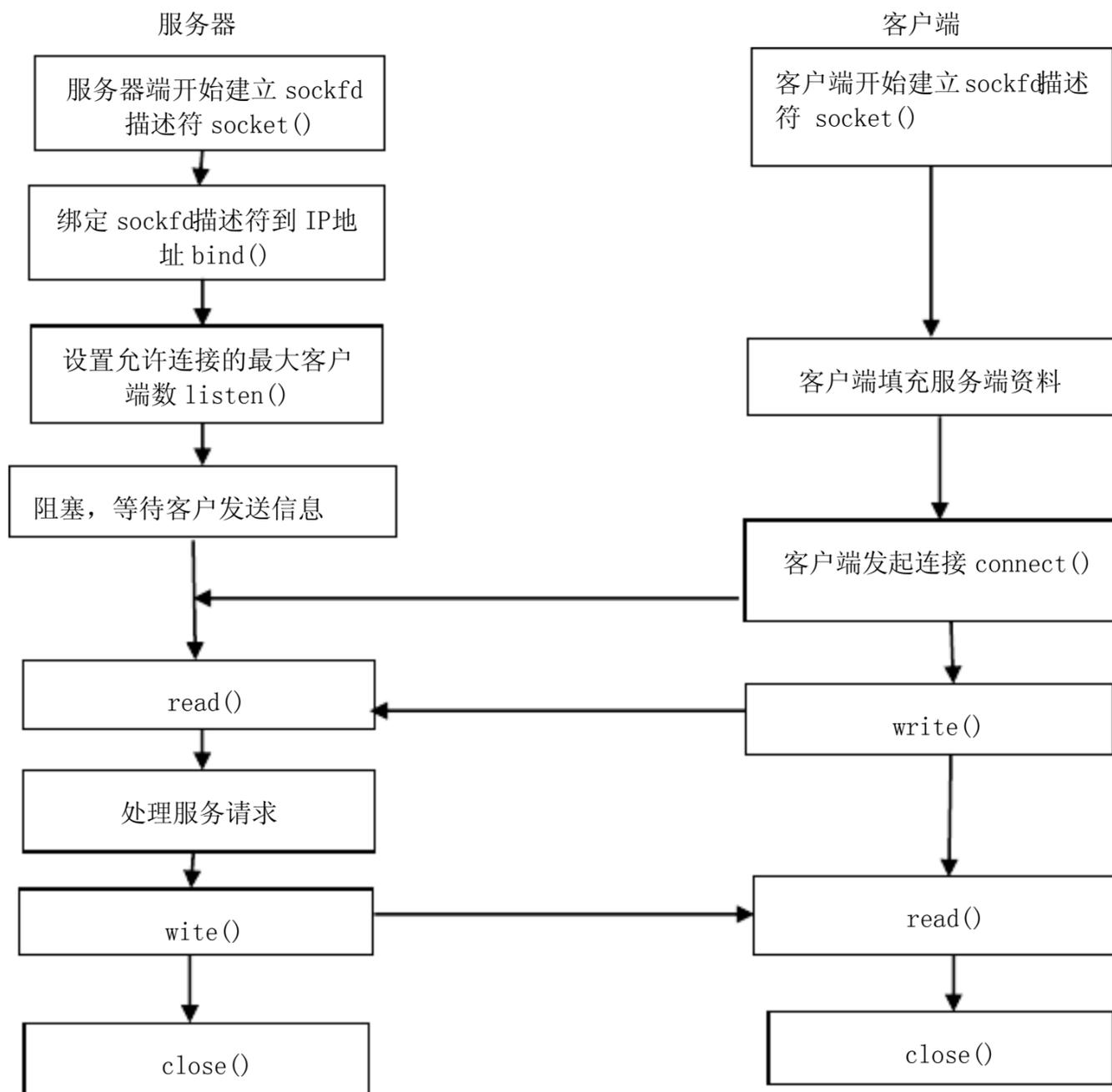


图 3.4 面向连接的客户/服务器程序工作流程

TQ2440 开发板网络客户端源程序

```

void Widget::Connect_to_Server()
{
    QString port = "6666";
    tcpClient->connectToHost(ui->hostLineEdit->text(), port.toInt());
}
void Widget::SendMsg()
{
    outBlock = temp;
    qDebug() << "outBlock:" << outBlock;
    temp.resize(0);
    tcpClient->write(outBlock);
    outBlock.resize(0);
    tcpClient->close();
}
  
```

3.4 网络服务器端软件设计

创建网络服务器端步骤:

- (1) 创建 socket。
- (2) 绑定 IP 地址, 端口等信息到 socket。

- (3) 设置最大连接数。
- (4) 接收客户端上来的连接。
- (5) 接收数据。
- (6) 关闭网络连接。

网络服务器模块代码如下。

```
void Widget::start()
{
    if(!tcpServer.listen(QHostAddress::Any, 6666))
    {
        qDebug() << tcpServer.errorString();
        close();
        return;
    }
}

void Widget::acceptConnection()
{
    tcpServerConnection = tcpServer.nextPendingConnection();
    connect(tcpServerConnection, SIGNAL(readyRead()), this, SLOT(Receive()));
    connect(tcpServerConnection, SIGNAL(error(QAbstractSocket::SocketError)), this, SLOT(displayError(QAbstractSocket::SocketError)));
    tcpServer.close();
}
```

服务器端接收到网络上的数据后首先要对数据进行解析，然后对解析后的数据进行校验，如果校验正确，将数据插入数据库，如果不正确，对数据作丢弃处理。源码如下。

```
void Widget::Receive()
{
    inBlock = tcpServerConnection->readAll();
    size=inBlock.size();
    qDebug() <<"size=" <<size;
    QString hum1=inBlock.data()[0];
    QString hum2=inBlock.data()[1];
    QString tem1=inBlock.data()[2];
    QString tem2=inBlock.data()[3];
    QString sum=inBlock.data()[4];
    qDebug() <<"inBlock:" <<inBlock;
    qDebug() <<"sum" <<inBlock;
    bool ok=true;
    qDebug() <<"hum1=" <<hum1;
    qDebug() <<"hum2=" <<hum2;
    qDebug() <<"tem1=" <<tem1;
    qDebug() <<"tem2=" <<tem2;
    qDebug() <<"sum=" <<sum;
    if(sum.toInt(&ok, 16)==(tem1.toInt(&ok, 16)+tem2.toInt(&ok, 16)+hum1.toInt(&ok, 16)+hum2.toInt(&ok, 16))%10)
    {
        ui->label_3->setText(QString::number(tem1.toInt(&ok, 16)*10+tem2.toInt(&ok, 16)));
        ui->label_5->setText(QString::number(hum1.toInt(&ok, 16)*10+hum2.toInt(&ok, 16)));
    }
}
```

```

QString date = QDateTime::currentDateTime().toString("yyyy-MM-dd hh:mm:ss");
 QSqlQuery query;
 query.prepare("INSERT INTO tem_hum VALUES(?, ?, ?)");
 query.addBindValue(date);
 query.addBindValue(tem1.toInt(&ok, 16)*10+tem2.toInt(&ok, 16));
 query.addBindValue(hum1.toInt(&ok, 16)*10+tem2.toInt(&ok, 16));
 bool flag=query.exec();
 }//end if
 tcpServerConnection->close();
 inBlock.resize(0);
 start();
 }

```

3.5 数据库设计

MySQL 是一个小型关系型数据库管理系统，MySQL 是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内。这样就增加了速度并提高了灵活性。MySQL 的 SQL “结构化查询语言”。SQL 是用于访问数据库的最常用标准化语言。MySQL 软件采用了 GPL（GNU 通用公共许可证）。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，许多中小型企业为了降低企业总体拥有成本而选择了 MySQL 作为网络数据库。

(1) 根据需求分析得出的个体实体间的关系画 E-R 图如图 3.5 所示。

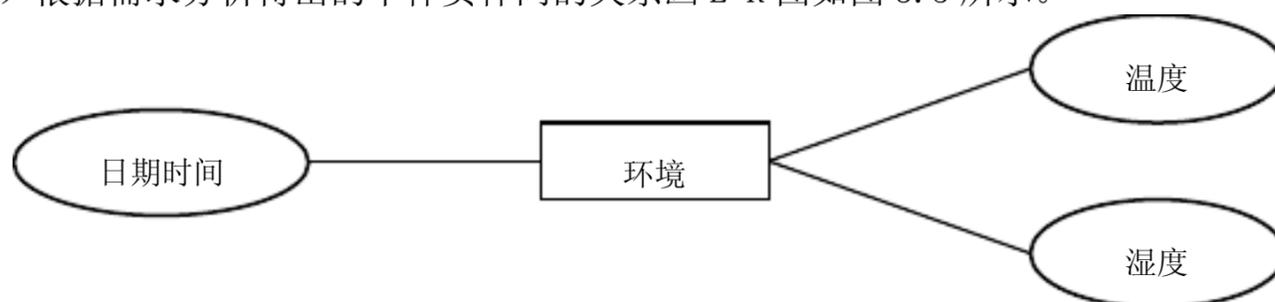


图 3.5 E-图

(2) 根据上述 E-R 图，表的关系模式结构如下所示：

环境（日期时间，温度，湿度）

(3) 建立数据库

```

CREATE TABLE `tem_hum` ( `date` datetime NOT NULL,
 `temperature` char(20) NOT NULL,
 `humidity` char(20) DEFAULT NULL,
 PRIMARY KEY (`date`));

```

(4) 数据库连接源码

```

bool Widget::createDataBase()
{
    QSqlDatabase db=QSqlDatabase::addDatabase("QMYSQL")连接数据库类型
    db.setHostName("localhost"主机名字
    db.setDatabaseName("vegetable"数据库名字
    db.setUserName("root");
    db.setPassword("123");
    if(!db.open())
    {
        QMessageBox::critical(0, tr("Database Error"),
        db.lastError().text());
    }
}

```

```

        return false;
    }
    qDebug("database open succeeded");
    return true;
}

```

服务器查询源码设计如下。

void select::sel()查询某一天的温湿度记录用曲线显示出

```

{
    QString num_1;
    QString num_2;
    num_1.clear();
    num_1.operator +=(ui->yearEdit->text());
    num_1.operator +=("-");
    num_1+=(ui->monthEdit->text());
    num_1+=("-");
    num_1+=(ui->dayEdit->text());
    num_1+=(" ");
    for(int i=0;i<24;i++)
    {
        num_2=num_1;
        if(i<10)
            num_2+="0";
        num_2+=(QString::number(i));
        num_2+=("%");
        query.exec("select * from tem_hum where date like '"+num_2+"'");
        if(query.size()>=1)
        {
            query.next();
            tem_hum[0][i]=query.value(1).toInt();
            tem_hum[1][i]=query.value(2).toInt();
        }
    }
    paintEvent();
}

void select::sel1()
{
    tem=0;
    hum=0;
    te_tab=1;
    hu_tab=0;
    te_hu=0;
    ui->label_3->hide();
    ui->label_4->hide();
    ui->label_5->hide();
    ui->pushButton->hide();
    ui->yearEdit->hide();
}

```

```
    ui->monthEdit->hide();
    ui->dayEdit->hide();
    ui->label->show();
    ui->label_2->show();
    ui->lineEdit->show();
    ui->lineEdit_2->show();
    ui->pushButton_2->show();
    QString num_1 = ui->lineEdit->text();
    num_1.operator +=(" 00:00:00");
    QString num_2 = ui->lineEdit_2->text();
    num_2.operator +=(" 23:59:59");
    QSqlQueryModel *model=new QSqlQueryModel;
    model->setQuery("SELECT date,temperature FROM tem_hum WHERE date >= '"+num_1+"' and date
<=' "+num_2+"' ");
    model->setHeaderData(0, Qt::Horizontal, tr("date"));
    model->setHeaderData(1, Qt::Horizontal, tr("temperature"));
    ui->tableView->setModel(model);
    ui->tableView->show();
    update();
}
```

4 系统连接和测试

系统各模块的连接是先通过单片机采集数据后由串口线连接到 QT2440 开发板，QT2440 开发板通过网线接入网络。模块连接图如图 4.1 所示。



图 4.1 各模块连接图

打开单片机和 QT2440 开发板，并在 ARM 版启动串口程序，测试是否能够接收到采集的数据，测试如图 4.2 所示。

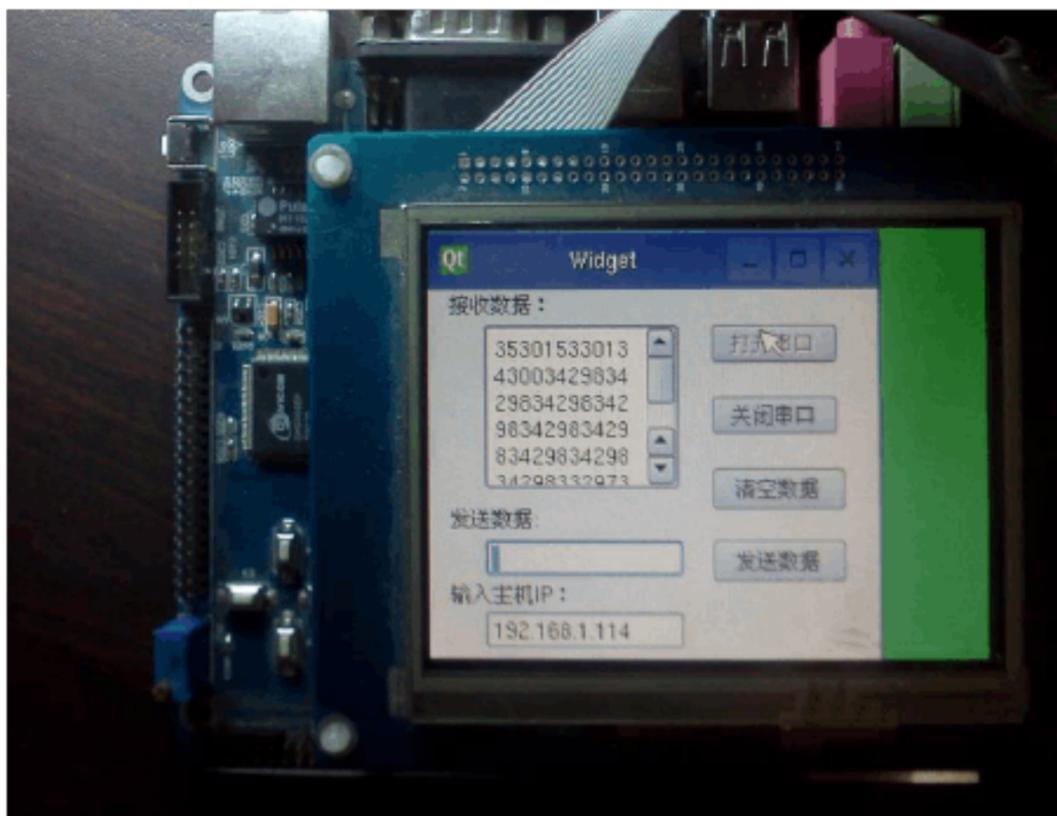


图 4.2 ARM 板接收到的串口数据

在服务器端启动服务器程序，然后在 QT2440 开发版输入服务器 IP，开发板通过网络服务将串口数据传输到服务器，测试结果如图 4.3 所示。



图 4.3 服务器接收到的当前数据

当用户打开服务器后，我们可以选择查询了历史信息，点击“查询历史信息”按钮后，软件会弹出历史信息查询界面，如图 4.4 所示。

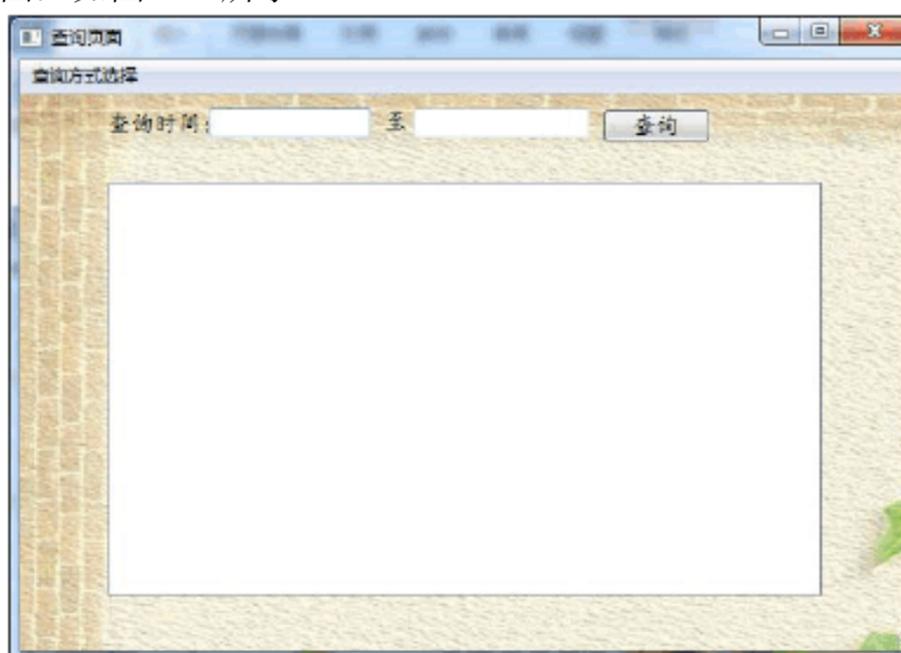


图 4.4 历史信息查询界面

在查询界面下，用户可以根据查询要求选择所需要的信息，也可以选着显示方式，如图 4.5 示。

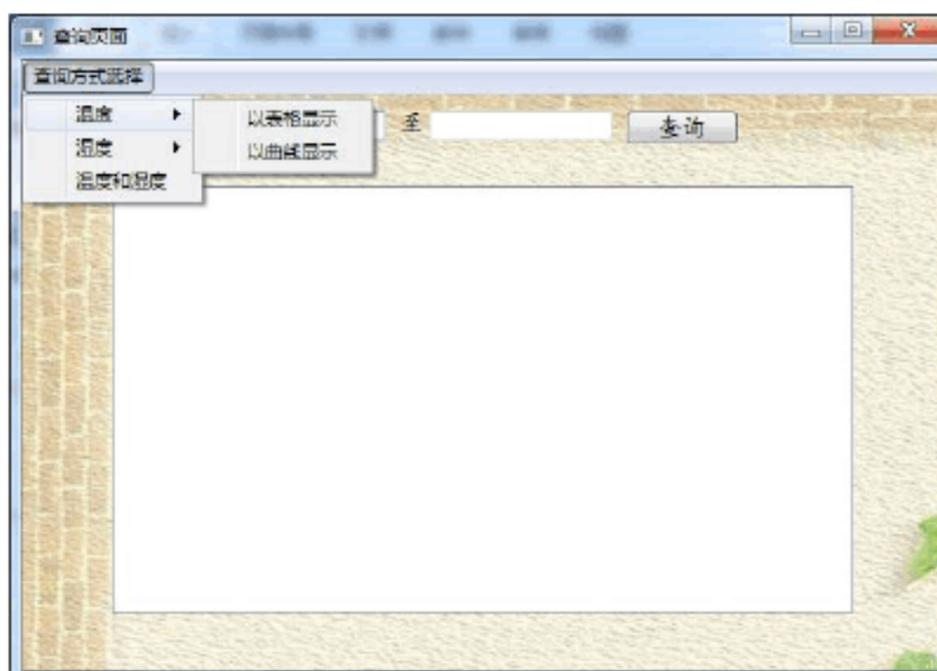


图 4.5 选择查询方式界面

选择查询 2012 年 4 月 20 日温湿度信息以图表显示界面如图 4.6 所示。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/227123052101010005>