毕业教学环节成果

（2011 届）

题　　目　基于 51 单片机的家用水流量设计

学　　院　信息工程学院

专　　业　应用电子技术

班　　级　应电 082

学　　号　**********

姓　　名　*******

指导教师　******

2011 年 5 月 10 日

金华职业技术学院毕业教学成果

目　　录

摘要.....1
引言.....2
1　任务设计.....3
2　系统硬件电路的设计.....4
2.1　主芯片 STC89C52.....4
2.2　时钟电路.....6
2.3　复位电路.....7
2.4　电源电路.....8
2.5　液晶显示电路.....8
2.6　状态显示电路.....1　5
2.7　温度测量电路.....1　5
2.8　水流量测量电路.....2　2
2.9　按键控制电路.....2　2
2.10　小结.....2　3
3　软件系统的设计.....2　4
3.1　软件设计总流程.....2　4
3.2　温度程序模块.....2　5
3.3　水流量程序模块.....3　7
3.4　显示程序.....3　9
3.5　小结.....2　9
4　调试结果记录.....3　0
4.1　温度测量.....3　0
4.2　流量测量.....3　0
5　总结.....3　0
6　成果展示.....3　1
结论与谢辞.....3　2
参考文献.....3　3
附件 1 电路原理图.....3　4
附件 2　PCB 图.....3　5
附件 3　仿真图.....3　6

附件 4 元件清单.....3 7
附件 5 程序清单.....3 8

基于 51 单片机的家用水流量设计

信息工程学院应用电子技术　　刘海清

摘要:文以智能家居系统中的水流量模块为课题，以水流量计传感器和数字温度传感器 DS18B20 实时采集水流量状态和水温的数据,并根据主控器 STC89C52 的程序指令处理后计算出水费价格，用液晶屏 LCD1602 分当前温度、水价和水流总量三种状态动态显示。另外，本课题设计了上下限报警，使查看更为直观，使用更加方便。

关键词:　单片机　DS18B20 温度传感器　水流量传感器

Household Water-Flow Detection Circuit Design Based on 51 MCU

(Major of Applied Electronic Technology,Information and Engineering college, JinHua College of Vocation And Technology, LIU Hai-qing)

Abstract: Taking the smart home system in the water flow module is subject to water flow sensor and digital temperature sensor DS18B20 real-time acquisition state water flow and water temperature data and program instructions according to master STC89C52 calculated after the price of water, with LCD TX-1602 points the current temperature, water flow volume and dynamic display of three states. In addition, the project design of the upper and lower alarm, so view is more intuitive, easier to use.

Keyword: microprocessor　DS18B20temperature　sensor water flow sensor

引言

随着现代社会的进步，经济的发展，人们对精神领域的追求更高，对生活水平的要求更高。现代的家居生活是一种高品位、高质量、个性化、智能化的方式。本系统就是基于 STC89C52 单片机控制的智能家居系统，可以实际监控室内各种不同的家电设备，并能通过液晶屏动态显示当前工作状态。该系统与传统的智能家居系统相比，具有功能多样化、成本造价低等优点，且符合当今社会智能、节能、环保的发展观念，并在人们享受高品位、高质量、个性化、智能化生活的同时提高人们的节约意识。由于智能家居系统有众多模块，本课题只采取其中的水流量模块进行单独设计。

1　任务设计

当打开水龙头时，根据单片机 STC89C52 的指令、水流量计传感器和数字温度传感器 DS18B20 实时采集水流量状态和水温的数据。当单片机 STC89C52 扫描到水流量计传感器的脉冲数，经过单片机 STC89C52 处理，计算出所采集的水流量后，通过液晶屏 LCD1602 能动态显示当前水流量、水费及水温。

根据设计过程，可以将的本课题划分为 8 个电路模块如图 1 所所示：

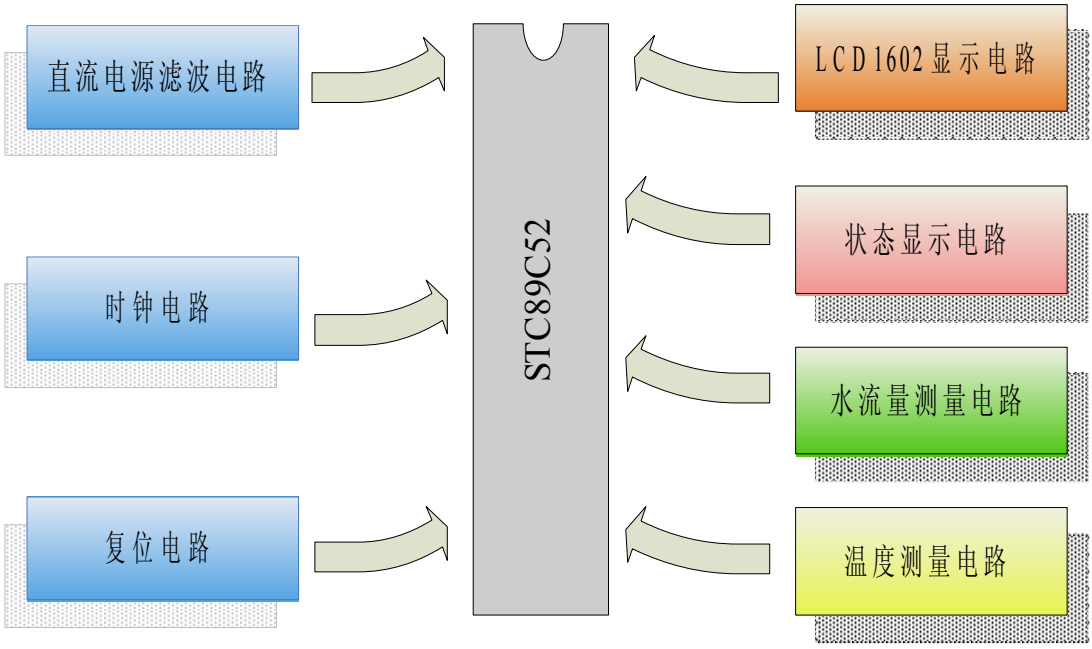


图 1-1　电路总框图

2　系统硬件电路的设计

2.1　主芯片 STC89C52

2.1.1　主要性能

① 与 **MCS-52** 单片机产品兼容 、**8K** 字节在系统可编程 **Flash** 存储器

② **1000** 次擦写周期

③ 全静态操作：**0Hz～33Hz**

④ 三级加密程序存储器

⑤ **32** 个可编程 **I/O** 口线

⑥ 三个 **16** 位定时器/计数器八个中断源

⑦ 全双工 **UART** 串行通道

⑧ 低功耗空闲和掉电模式

⑨ 掉电后中断可唤醒

⑩ 看门狗定时器

⑪ 双数据指针

⑫ 掉电标识符

2.1.2　芯片功能特性简述：

STC89C52 是一种低功耗、高性能 **CMOS8** 位微控制器，具有 **8K** 在系统可编程 **Flash** 存储器。使用高密度非易失性存储器技术制造，与工业 **80C51** 产品指令和引脚完全兼容。片上 **Flash** 允许程序存储器在系统可编程，亦适于常规编程器。在单芯片上，拥有灵巧的 **8** 位 **CPU** 和在线系统可编程 **Flash**，使得 **STC89C52** 为众多嵌入式控制应用系统提供高灵活、超有效的解决方案。 **STC89C52** 具有以下标准功能： **8k** 字节 **Flash**，**256** 字节 **RAM**， **32** 位 **I/O** 口线，看门狗定时器，**2** 个数据指针，三个 **16** 位 定时器/计数器，一个 **6** 向量 **2** 级中断结构，全双工串行口，片内晶振及时钟电路。另外，**STC89C52** 可降至 **0Hz** 静态逻辑操作，支持 **2** 种软件可选择节电模式。空闲模式下，**CPU** 停止工作，允许 **RAM**、定时器/计数器、串口、中断继续工作。掉电保护方式下，**RAM** 内容被保存，振荡器被冻结，单片机一切工作停止，直到下一个中断或硬件复位为止。**8** 位微控制器 **8K** 字节在系统可编程 **Flash**。

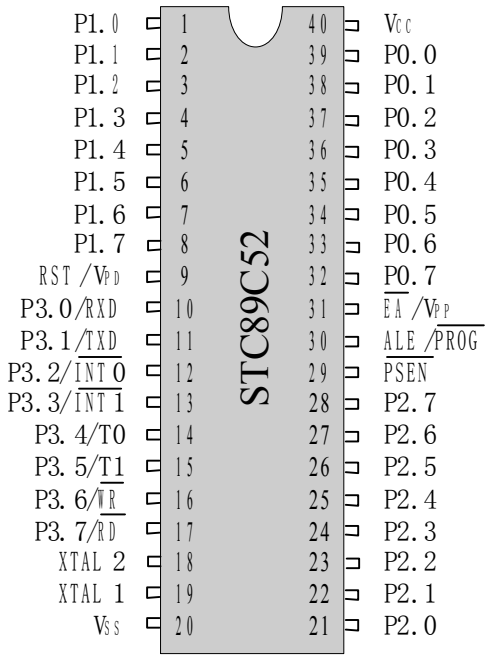


图 2-1　单片机引脚

2.1.3　引脚功能

表 2-1　STC89C52 引脚介绍说明

引脚	功能介绍
VCC	+5V 电源电压
VSS	电路接地端
P0.0～P0.7	8 位漏极开路的双向 I/O 通道
P2.0～P2.7	8 位拟双向 I/O 通道
P3.0	RXD，串行输入口
P3.1	TXD，串行输出口
P3.2	INT0，外部中断 0 输入口
P3.3	INT1，外部中断 1 输入口
P3.4	定时器/计数器 0 外部事件脉冲输入端
P3.5	定时器/计数器 1 外部事件脉冲输入端
P3.6	外部数据存贮器写脉冲
P3.7	外部数据存贮器读脉冲
RST/VpD	复位输入信号
ALE/PROG	地址锁存有效信号
PSEN	程序选通有效信号
EA/VPP	当保持 TTL 高电平，执行内部 EPROM 的指令，当使 TTL 为低电平， 从外部程序存贮器取出所有指令，在内的 EPROM 编程时，此端为 21Ⅴ编程电源输入端
XTAL1	内部振荡器外接晶振的一个输入端
XTAL2	内部振荡器外接晶振的一个输入端

2.2　时钟电路

单片机的最小系统有三部分组成，即电源，时钟电路和复位电路。其中单片机的电源引脚与 **5V** 电源连通即可，而时钟电路和复位电路还需接口扩展，这也是单片机的基本电路操作。

时钟电路用于产生单片机工作所需的时钟信号，时序是指令执行中各信号之间的相互关系。单片机本身就如同一个复杂的同步时序电路，为了保证同步工作方式的实现，电路应在唯一的时钟信号控制下严格地按时序进行工作。在 **STC89C52** 单片机内部带有时钟电路，因此，只需要在片外通过 **XTAL1** 和 **XTAL2** 引脚接入定时控制元件(晶体振荡器和电容)，即可构成一个稳定的自激振荡器。在 **STC89C52** 芯片内部有一个高增益反相放大器，而在芯片的外部，**XTAL1** 和 **XTAL2** 之间跨接晶体振荡器和微调电容。在单片机的 **XTAL1** 脚和 **XTAL2** 脚之间并接一个晶体振荡器就构成了内部振荡方式。**STC89C52** 单片机内部有一个高增益的反相放大器，**XTAL1** 为内部反相放大器的输入端，**XTAL2** 为内部反相放大器的输出端，在其两端接上晶振后，就构成了自激振荡电路，并产生振荡脉冲，振荡电路输出的脉冲信号的频率就是晶振的固有频率。在实际应用中通常还需要在晶振的两端和地之间各并上一个小电容。

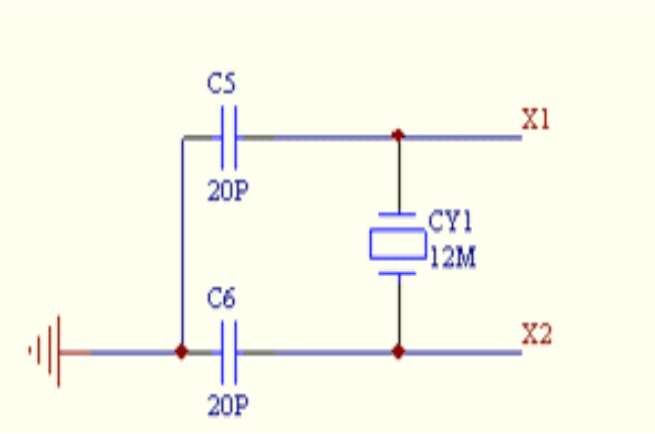


图 2-2　时钟电路

用晶振和电容构成谐振电路。电容大小与晶振频率和工作电压有关。但电容的大小影响振荡器的稳定性和起振的快速性，为了提高精度，本实验板采用 **20pF** 的电容作为微调电容。在设计电路板时，晶振、电容等均应尽可能靠近芯片，减小分布电容，以保证振荡器振荡的稳定性。

2.3　复位电路

复位是单片机的初始化操作，其目的是使 **CPU** 和系统中各部分处于一个确定的状态，并从这一状态开始工作。系统上电路或死机后都要进行复位操作。单片机的 **RST** 引脚为复位引脚，振荡电路正常工作后，**RST** 端加上持续两个机器周期的高电平后，单片机就被复位。复位电路有 **3** 种基本方式：上电复位，开关复位和看门狗复位。

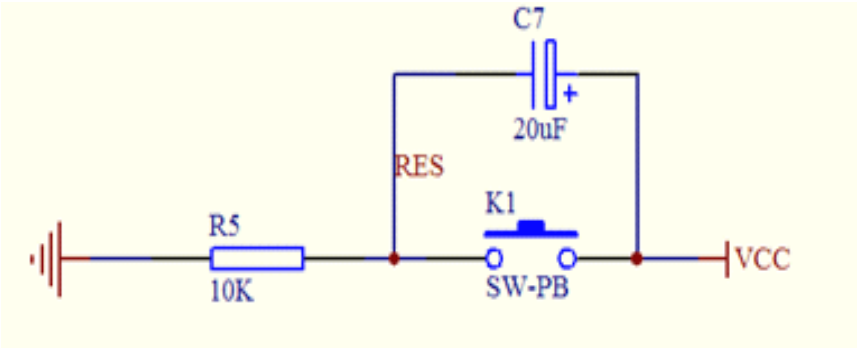


图 2-3　复位电路

本课题采用按键开关复位是指通过接通按钮开关，使单片机进入复位状态。开关复位电路一般不单独使用。在应用系统设计中，若需使用开关复位电路，一般的做法是将开关复位与上电复位组合在一起形成组合复位电路，上电复位电路完成上电复位功能，开关复位电路完成人工复位。

图 **2-3** 中 **C7** 与 **R1** 构成了上电复位电路。上电复位后，电源经 **R1** 对 **C7** 充满电源，**C7** 等效于开路，**RST** 端为低电平；单片机正常工作。按开关 **K1** 后，**C7** 两端电荷经 **R1** 迅速放电，**K1** 断开后，由 **C7**、**R1** 及电源完成对单片机的复位操作。在上述电路中 **C7**、**R1** 按上电复位电路的设计而取值。

复位电路的作用非常重要，能否成功复位关系但单片机系统能否正常运行的问题。如果振荡电路正常而单片机系统不能正常运行，其主要原因是单片机没有完成正常复位，程序计数器的值没有回 **0**，特殊功能寄存器没有回到初始状态。这时可以适当地调整上电复位电路的阻容值，增加其充电时间常数来解决问题。

2.4 直流电源滤波电路

单片机对于直流电源电压非常敏感，但是一般直流电源都存在一些杂波，通常是直流电压中的高频交流成分，消除电源中的高频交流成分对增强电路的性能具有较大作用。因而，我们设计了滤波电路，起到滤波的作用，从而更好的避免不必要的故障发生。滤波电路的基本原理是利用电容或电感的滤波特性，图 **2-4** 电源电路采用电容滤波。图中 **LED** 是用来指示电源接通的情况。

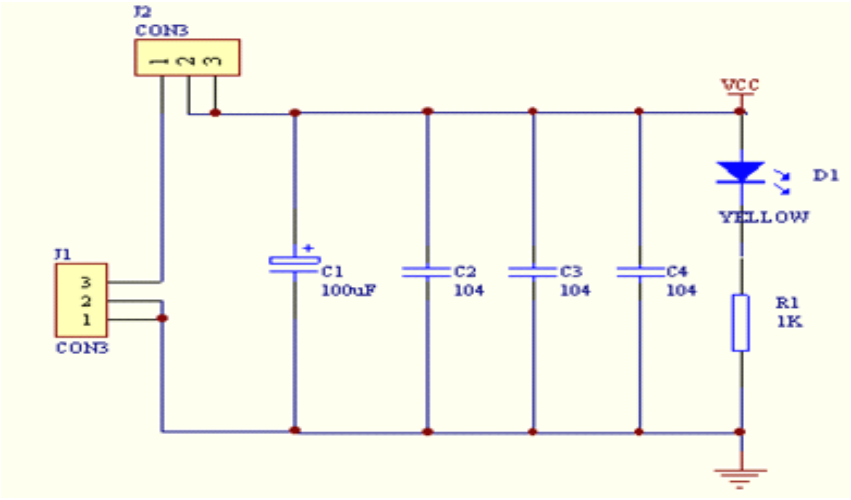


图 2-4 电源电路

2.5 液晶显示电路

课题任务要求以 **LCD1602** 芯片显示单片机处理后的温度、水费和水流量，在此有必要详尽的介绍 **LCD1602** 的特性和用法。

2.5.1 显示特性

① 只需 **5V** 电源电压，低功耗、长寿命、高可靠性

② 内置 **192** 种字符（**160** 个 **5×7** 点阵字符和 **32** 个 **5×10** 点阵字符）

③ 具有 **64** 个字节的自定义字符 **RAM**

④ 显示方式：**STN**、半透、正显

⑤ 驱动方式：**1/16DUTY**，**1/5BIAS**

⑥ 视角方向：**6** 点

⑦ 背光方式：底部 **LED**

⑧ 通讯方式：**4** 位或 **8** 位并口可选

⑨ 标准的接口特性：适配 **MC51** 和 **M6800** 系列 **MPU** 的操作时序。

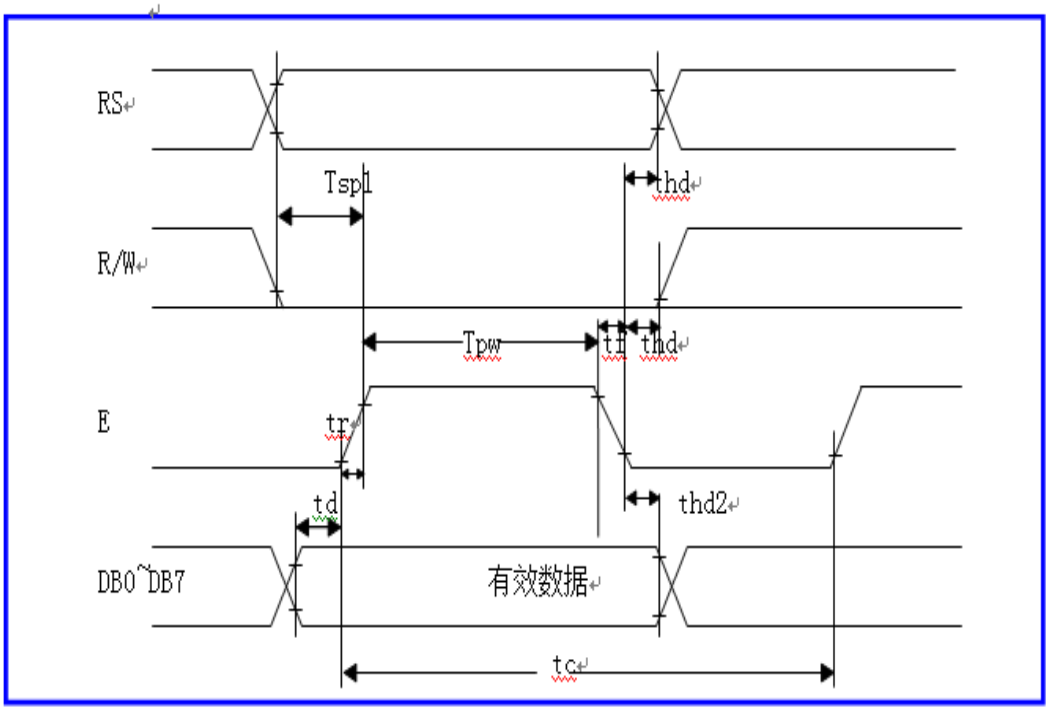
2.5.2 引脚说明

表 2-2 液晶 1602 引脚说明

管脚号	符号	功能	
1	Vss	电源地（GND）	
2	Vdd	电源电压(+5V)	
3	V0	LCD 驱动电压(可调) 寄存器选择输入端,输入 MPU 选择模块内部寄存器类型信号：RS=0，当 MPU 进行写模块操作，指向指令寄存器；	
4	RS	当 MPU 进行读模块操作，指向地址计数器；RS=1，无论 MPU 读操作还是写操作，均指向数据寄存器	
5	R/W	R/W=0 读操作；R/W=1 写操作	
6	E	使能信号输入端，输入 MPU 读/写模块操作使能信号：	4 位方式通讯时,不使用 DB0-DB3
7	DB0	数据输入/输出口，MPU 与模块之间的数据传送通道	
8	DB1	数据输入/输出口，MPU 与模块之间的数据传送通道	
9	DB2	数据输入/输出口，MPU 与模块之间的数据传送通道	
10	DB3	数据输入/输出口，MPU 与模块之间的数据传送通道	
11	DB4	数据输入/输出口，MPU 与模块之间的数据传送通道	
12	DB5	数据输入/输出口，MPU 与模块之间的数据传送通道	
13	DB6	数据输入/输出口，MPU 与模块之间的数据传送通道	
14	DB7	数据输入/输出口，MPU 与模块之间的数据传送通道	
15	A	背光的正端+5V	
16	K	背光的负端 0V	

2.5.3 接口时序

读操作时序：



写操作时序：

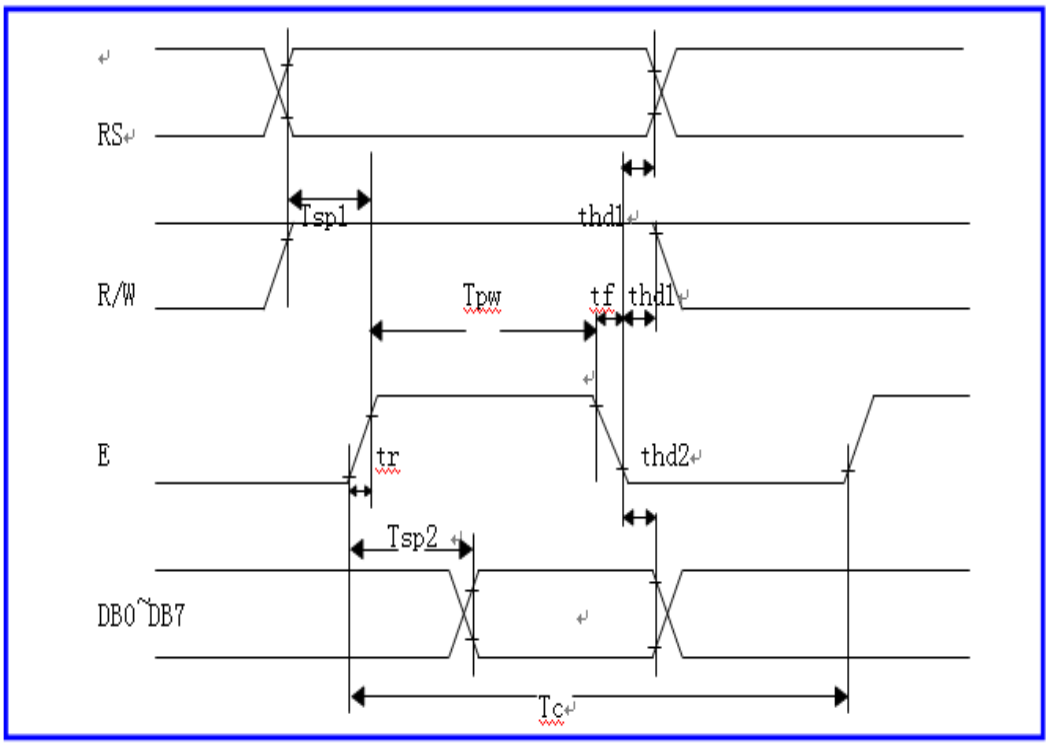


图 2-5　时序图

表 2-3　液晶 1602 时序图标号说明

时序参数	符号	极限值			单位	测试条件
		最小值	典型值	最大值		
E 信号周期	tc	400			ns	引脚 E
E 脉冲宽度	Tpm	150			ns	
E 上升沿/下降沿时间	Tr,tf			25	ns	
地址建立时间	Tsp1	30			ns	引脚 E、RS、R\W
地址保持时间	Thd1	10			ns	
数据建立时间（读操作）	Td			100	ns	引脚 DB0~DB7
数据保持时间（读操作）	Thd2	20			ns	
数据建立时间（写操作）	Tsp2	40			ns	
数据保持时间（写操作）	Thd2	10			ns	

程序实现如下：

```
/************************写指令程序************************/
void wr_com(unsigned char com)   //写指令
 {
    delay(1);       //延时 1ms
    RS=0;           //写命令设置
    RW=0;               //并行数据的读写
    EN=0;           //使能为 0
    P2=com;      //输入命令
    delay(1);      //延时 1ms
    EN=1;           //使能为 1
    delay(1);      //延时 1ms
    EN=0;           //使能为 0
  }
/************************写数据程序************************/
void wr_dat(unsigned char dat)//  写数据
{
    delay(1);      //延时 1ms
    RS=1;       //写数据设置
```

```
    RW=0;          //并行数据的读写
    EN=0;      //使能为 0
    P2=dat;          //输入数据
    delay(1);     //延时 1ms
    EN=1;      //使能为 1
    delay(1);     //延时 1ms
    EN=0;      //使能为 0
}
```

2.5.4 初始化指令：

表 2-4 清屏指令

指令功能	指令编码										执行时间 /ms
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
清屏	0	0	0	0	0	0	0	0	0	1	1.64

功能：

1. 清除液晶显示器，即将 DDRAM 的内容全部填入"空白"的 ASCII 码 20H;
2. 光标归位，即将光标撤回液晶显示屏的左上方;
3. 将地址计数器(AC)的值设为 0。

表 2-5 光标归位指令

指令功能	指令编码										执行时间 /ms
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
光标归位	0	0	0	0	0	0	0	0	1	X	1.64

功能：

1. 把光标撤回到显示器的左上方;
2. 把地址计数器(AC)的值设置为 0;
3. 保持 DDRAM 的内容不变

表 2-6 进入模式设置指令

指令功能	指令编码										执行时间 /ms
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
进入模式设置	0	0	0	0	0	0	0	1	I/D	S	1.64

功能：设定每次定入 1 位数据后光标的移位方向，并且设定每次写入的一个字符是否移动。

表 2-7　显示开关控制指令

指令功能	指令编码										执行时间/us
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
显示开关控制	0	0	0	0	0	0	1	D	C	B	40

功能：控制显示器开/关、光标显示/关闭以及光标是否闪烁。

表 2-8　设定显示屏或光标移动方向指令

指令功能	指令编码										执行时间/us
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
设定显示屏或光标移动方向	0	0	0	0	0	1	S/C	R/L	X	X	40

功能：使光标移位或使整个显示屏幕移位。

表 2-9　功能设定指令

指令功能	指令编码										执行时间/us
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
功能设定	0	0	0	0	1	DL	S/C	R/L	X	X	40

功能：设定数据总线位数、显示的行数及字型。参数设定的情况如下：

表 2-10　设定 CGRAM 地址指令

指令功能	指令编码										执行时间/us
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
设定 CGRAM 地址	0	0	0	1	CGRAM 的地址（6 位）						

功能：设定下一个要存入数据的 CGRAM 的地址。

表 2-11　设定 DDRAM 地址指令

指令功能	指令编码										执行时间/us
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
设定 DD RAM 地址	0	0	1	CGRAM 的地址（7 位）							40

功能：设定下一个要存入数据的 CGRAM 的地址。

(注意这里我们送地址的时候应该是 0x80+Address，这也是前面说到写地址命令的时候要加上 0x80 的原因)

表 2-12　读取忙信号或 AC 地址指令

指令功能	指令编码										执行时间/u s
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
读取忙碌信号或 AC 地址	0	1	FB	AC 内容（7 位）							40

功能：

1. 读取忙碌信号 BF 的内容，BF=1 表示液晶显示器忙，暂时无法接收单片机送来的数据或指令;当 BF=0 时，液晶显示器可以接收单片机送来的数据或指令;
2. 读取地址计数器(AC)的内容。

表 2-13　数据写入 DDRAM 或 CGRAM 指令一览

指令功能	指令编码										执行时间/u s
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
写数据到 DDRAM 或 CGRAM	1	0	要写的数据 D7~D0								40

功能：

1. 将字符码写入 DDRAM，以使液晶显示屏显示出相对应的字符;
2. 将使用者自己设计的图形存入 CGRAM。

表 2-14　从 CGRAM 或 DDRAM 读出数据的指令一览

指令功能	指令编码										执行时间/u s
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
从 CGRAM 或 DDRAM 中读数据	1	1	要读的数据 D7~D0								40

功能：读取 DDRAM 或 CGRAM 中的内容。

单片机和 LCD 液晶显示器的连接

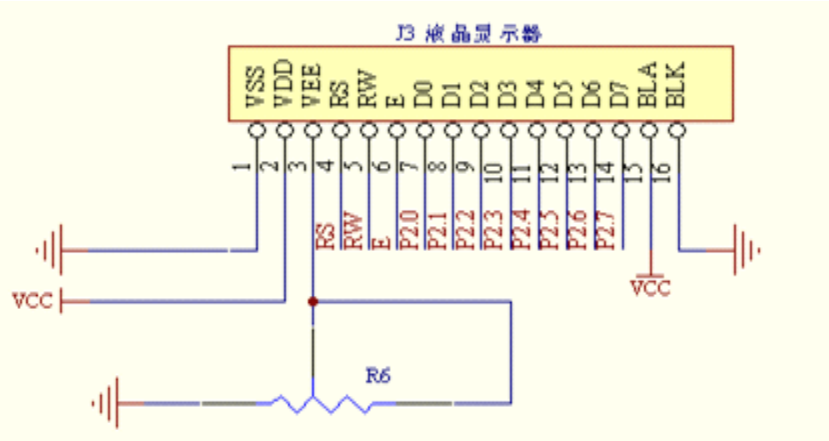


图 2-6　液晶显示电路

2.6　状态显示电路

电路设计若水流量开始技术则绿灯亮，水流量计数停止则绿灯灭。温度超过 **40** 度或小于 **0** 度红灯报警。

由于发光二极管的需求电流为 **3mA~10mA,**电压为 **3V** 所以计算

R=3V/（3~10）mA=(330~1000)欧

故我们在这采用 **R=330**，以使经过发光二极管电流较大，发光更亮。

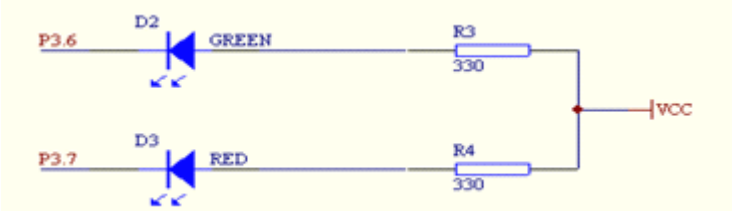


图 2-7　状态显示电路

2.7　温度测量电路

课题任务中需要测量水温，故先用温度传感器 **DS18B02** 的数据采集，再通过单片机数据处理，最后在液晶屏显示出来。

2.7.1　DS18B20 简介

DS18B20 是 **DALLAS** 半导体公司推出的第一片支持“一线总线”接口的温度传感器，他它具有微型化、低功耗、高性能、抗干扰能力强、易配微处理器等优点，可直接将温度转化成串行数字信号供处理器处理。**DS18B20** 具有以下优点：

适应电压范围宽，电压范围在 **3.0V~5.5V**，在寄生电源方式下可由数据线供电。

独特的单线接口方式，与微处理器连接时只需要一条口线即可实现微处理器与 **DS18B20** 的双向通信。

① 支持多点组网功能，多个 DS18B20 可以并联在唯一的三线上，实现组网多点测温。

② 在使用中不需要任何外围元件，全部传感元件以及转换电路集成在形如一直三极管的集成电路内。

③ 测温范围-55℃+125℃，在-10℃~+85℃时进度为±0.5℃，可编程分辨率为 9~12 位，对应的可分辨温度分别为 0.5℃、0.25℃、0.125℃和 0.0625℃，可实现高精度测温。

④ 负压特性。电源极性接反时，芯片不会因为过热而烧毁，但不能正常工作。

2.7.2 DS18B20 结构及其工作原理

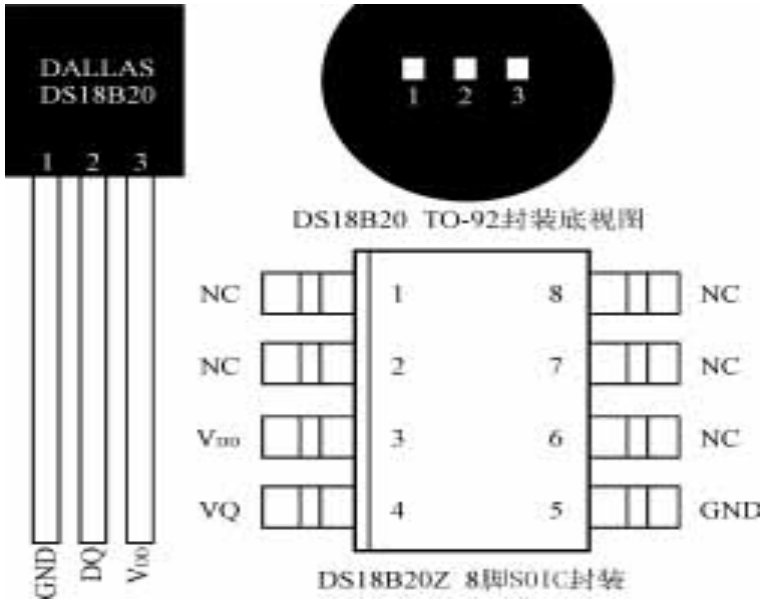


图 2-8 DS18B20 的引脚和封装

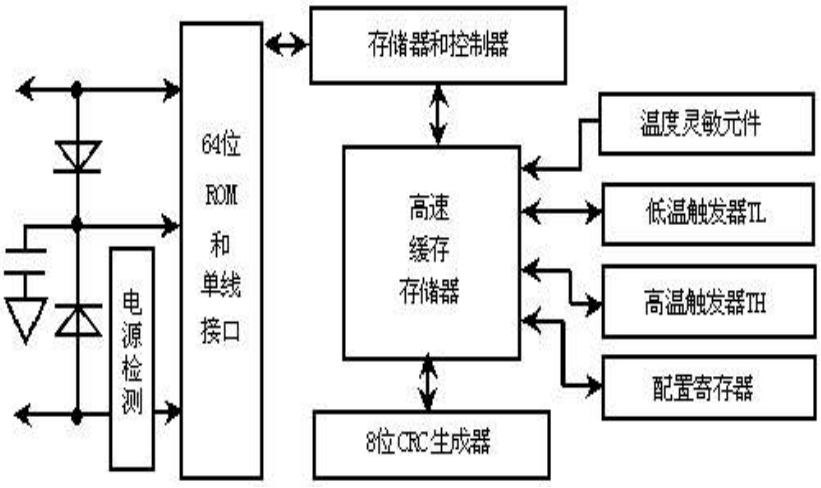


图 2-9 DS18B20 的内部结构

DS18B20 组成：64 位 ROM、温度传感器、非挥发的温度报警触发器 TH 和 TL、

配置寄存器。

DS18B20 的管脚排列： **DQ** 为数字信号输入／输出端

GND 为电源地

VDD 为外接供电电源输入端，在寄生电源接线方式时接地。

DS18B20 内部带有共 **9** 个字节的高速暂存器 **RAM** 和电可擦除 **EEPROM**，起结构如表 **2-15** 所示。

表 2-15　DS18B20 高速暂存器结构

寄存器内容	字节地址
温度值低位（LSB）	0
温度值高位（MSB）	1
高温限值（TH）	2
低温限值（TL）	3
配置寄存器	4
保留	5
保留	6
保留	7
CRC 校验值	8

表 2-16　DS18B20 操作指令

ROM 操作指令		
指令	约定代码	功能
读 ROM	33H	读 DS18B20 温度传感器 ROM 中的编码（即 64 位地址）
匹配 ROM	55H	发出命令后接着发出 64 位 ROM 编码，访问总线上与该编码对应的芯片
搜索 ROM	F0H	用于确定挂接在同一总线上 DS18B20 的个数
跳过 ROM	CCH	忽略 64 位 ROM 地址，直接向 DS18B20 发温度变换命令
告警搜索	ECH	执行后只有问多超过上限或下限的芯片才响应
RAM 操作指令		
指令	约定代码	功能
温度转换	44H	启动 DS18B20 温度转换。12 位转换时长典型值 750ms
读暂存器	BEH	读内部 RAM 中 9 字节的数据。
写暂存器	4EH	向 RAM 第 2、3 字节写上、下限温度数据，紧跟命令之后传送 2 字节数据
复制暂存器	48H	将 RAM 中第 2、3 字节的内容复制到内部 EEPROM 中
重调 EEPROM	B8H	将 EEPROM 中内容恢复到 RAM 中第 3、4 字节

温度数据在高速暂存器 RAM 的第 0 和第 1 个字节中的存储格式如下表 2-17 所示。

表 2-17　DS18B20 温度数据存储格式

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
位 15	位 14	位 13	位 12	位 11	位 10	位 9	位 8
S	S	S	S	S	2^6	2^5	2^4

DS18B20 在出厂是默认配置为 **12** 位，其中最高位为符号位，即温度值共 **11** 位，单片机在读取数据时，一次会读 **2** 字节共 **16** 位，读完后将低 **11** 位的二进制数转化为十进制数后再乘以 **0.0625** 便为所测的实际温度值。另外，还需要判断温度的正负。前 **5** 个数字为符号位，这 **5** 位同时变化，我们只需要判断 **11** 位就可以了。前 **5** 位为 **1** 时，读取的温度为负值，且测到的数值需要取反加 **1** 再乘以 **0.0625** 才可得到实际温度值。前 **5** 位为 **0** 时，读取的温度为正值，且温度为正值时，只要将测得的数值乘以 **0.0625** 即可得到实际温度值。考虑到实际使用的需要，在这里我们只使用一个 **DS18B20**，故每次操作前只需复位后发出 **Skip ROM** 指令（即跳过 **ROM** 指令）再读出温度的正值、并精确到小数点后一位，即可满足设计需求。

每颗 **DS18B20** 在出厂前都有一个 **64** 位光刻 **ROM**，它可以看作该 **DS18B20** 的地址序列码。其各位排列顺序是：开始 **8** 位为产品类型标号，接下来 **48** 位是该 **DS18B20** 自身的序列号，最后 **8** 位是前面 **56** 位的 **CRC** 循环冗余校验码（**CRC=X^8+X^5+X^4+1**）。光刻 **ROM** 的作用是使每一个 **DS18B20** 都各不相同，这样就可以实现一条总线 挂接多个 **DS18B20** 的目的。

由于 **DS18B20** 是在一根 **I/O** 线上读写数据，因此，对读写的数据位有着严格的时序要求。**DS18B20** 有严格的通信协议来保证各位数据传输的正确性和完整性。该协议定义了几种信号的时序：复位时序、读时序、写时序。所有时序都是将主机作为主设备，单总线器件作为从设备。而每一次命令和数据的传输都是从主机主动启动写时序开始，如果要求单总线器件回送数据，在进行写命令后，主机需启动读时序完成数据接收。数据和命令的传输都是低位在先。

（1） DS18B20 的复位时序：

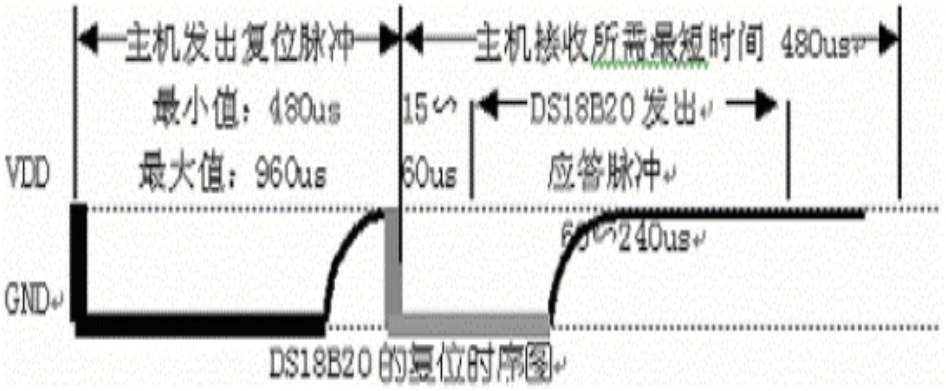


图 2-10 DS18B20 的复位时序图

```
/*****************ds1820 复位子程序******************/
void ds1820rst()
 {
  unsigned char x=0;
  DQ = 1;                  //DQ 复位
  delay_18B20(4);          //延时
  DQ = 0;                  //DQ 拉低
  delay_18B20(100);        //精确延时大于 480us
  DQ = 1;                  //拉高
  delay_18B20(40);         //延时
   }
```

（2） **DS18B20** 的读时序：

对于 **DS18B20** 的读时序分为读 **0** 时序和读 **1** 时序两个过程。

对于 **DS18B20** 的读时隙是从主机把单总线拉低之后，在 **15uS** 之内释放单总线，以让 **DS18B20** 把数据传输到单总线上。**DS18B20** 在完成一个读时序过程，至少需要 **60us** 才能完成。

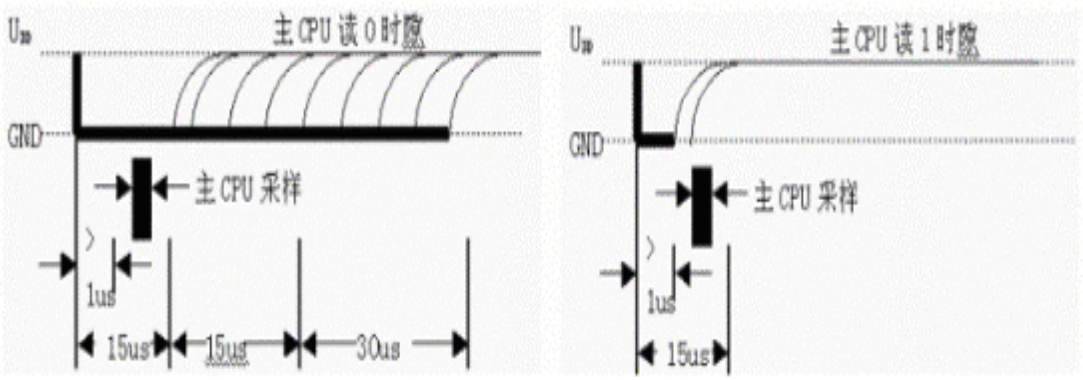


图 2-11　DS18B20 的读时序图

```
/*************ds1820 读数据子程序*******************/
uchar ds1820rd()                    //读数据
  {
    unsigned char i=0;
    unsigned char dat=0;
   for (i=8;i>0;i--)                //读温度 2 进制 8 次
   {
           DQ = 0;                  //给脉冲信号
            dat>>=1;                //将温度数据转移到 dat
            DQ = 1;                 //给脉冲信号
      if(DQ)                       //数据转换
          dat|=0x80;
      delay_18B20(10);
     }
    return(dat);
  }
```

（3）DS18B20 的写时序：

对于 DS18B20 的写时序仍然分为写 0 时序和写 1 时序两个过程。

对于 DS18B20 写 0 时序和写 1 时序的要求不同，当要写 0 时序时，单总线要被拉低至少 60us，保证 DS18B20 能够在 15us 到 45us 之间能够正确地采样 IO 总线上的“0”电平，当要写 1 时序时，单总线被拉低之后，在 15us 之内就得释放单总线。

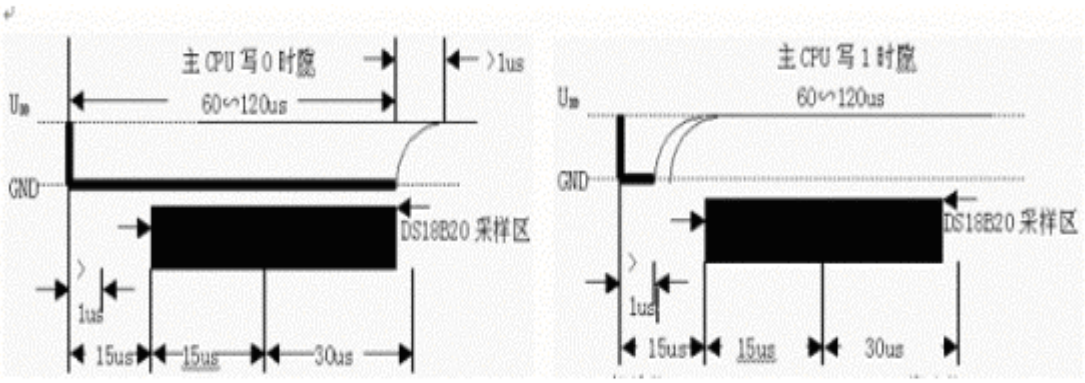


图 2-12　DS18B20 的写时序图

```
/*******************ds1820 写数据子程序******************/
void ds1820wr(uchar wdata)
  {
    unsigned char i=0;
    for (i=8; i>0; i--)              //写数据 2 进制 8 次
   {
     DQ = 0;                         //给脉冲信号
     DQ = wdata&0x01;                //数据传送
     delay_18B20(10);                //延时
     DQ = 1;                         //给脉冲信号
     wdata>>=1;                      //数据移位
   }
  }
```

2.7.3　温度测量电路接口展示

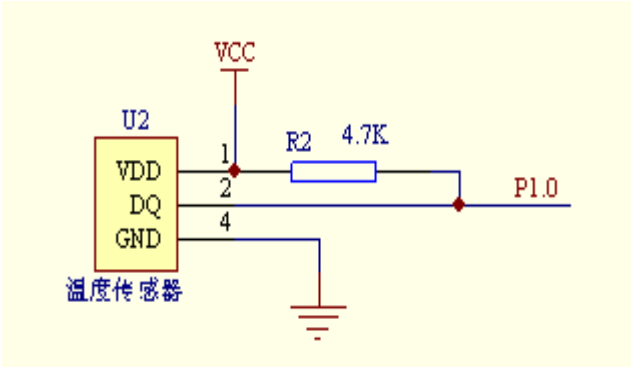


图 2-13　温度测量电路

2.8　水流量测量电路

课题任务中最重要的是水流量的测量。通过水流量传感器的数据采集，根据商家的水流量传感器的参数可以得出单片机在运算时的数据。

水流量传感器参数：

环境温度：**-10~55**℃

流量计算在流量为**:0.2~0.4L/Min** 时，**1L=2100** 次;**0.5~0.8L/Min** 时，**1L=2280** 次；**0.9~1.2L/Min** 时,**1L=2350** 次；**1.2~2.5L/Min** 时，**1L=2460** 次；　（脉冲次数在流量变化时有一定程度的变动）

接线方法　白线：信号输出；黑线：电源负（也可按客户要求定做）

测量精度：±**5%**（在流量稳定的系统，精度可达±**2%**）

工作电压：**DC0~~24V**

工作压力：≤**100PSI**（**7kg/cm)**

耐湿性能：在环境湿度为 **90%**以下时性能保持稳定

寿命测试：本产品用进口干簧管作感应元件，在负荷小于 **24V 1mA** 前提下，开关寿命大于 **3** 亿次。

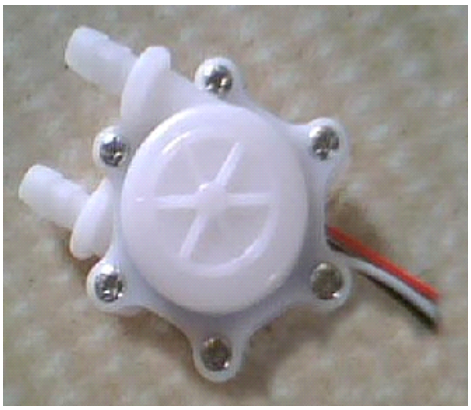


图 2-14　水流量计

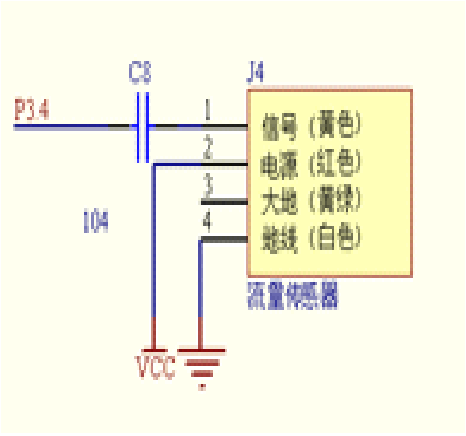


图 2-15　水流量测量电路

2.9　按键控制电路

课题设计之初，要求基本实现的功能有水流量的控制、水价的改变。由于液晶显示屏不能够完全实现其显示，故又增加了换页功能，共三个控制键。

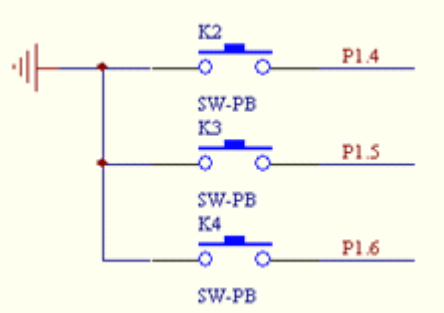


图 2-16　按键电路

2.10　小结

电路的设计，首先是把预定功能设定好，再看运行这些需要什么样的硬件，然后实施硬件的操作。各个模块都要有合理的设计。仔细认真是设计硬件电路的基本，一步走错，整个设计就毁于一旦。在设计本课题时，电路设计规则一定要注意，还有多多借鉴网络上的众多设计者分享的经验，益于自己的设计，总之一切为课题的成功做准备。

3　软件系统的设计

3.1　软件设计总流程

程序要求液晶显示有水温、水价、水费和当前水流量。水流量的测量由开关控制。由于 **LCD1602** 的显示屏幕有限，所以又把程序设计成可以翻页的形式。翻页也用开关控制。由于在两个界面里面都显示水温，故在显示程序里加入测量水温程序也未尝不可。整个程序中结构较为简单，但其中也有几个重要而且比较费脑筋的子程序，包括水测量程序、水流量数据的计算与转换、温度数据转换程序。

在程序中可以分为 **3** 个主要模块：水流量模块，温度模块，显示模块

如图 **3-1** 所示：

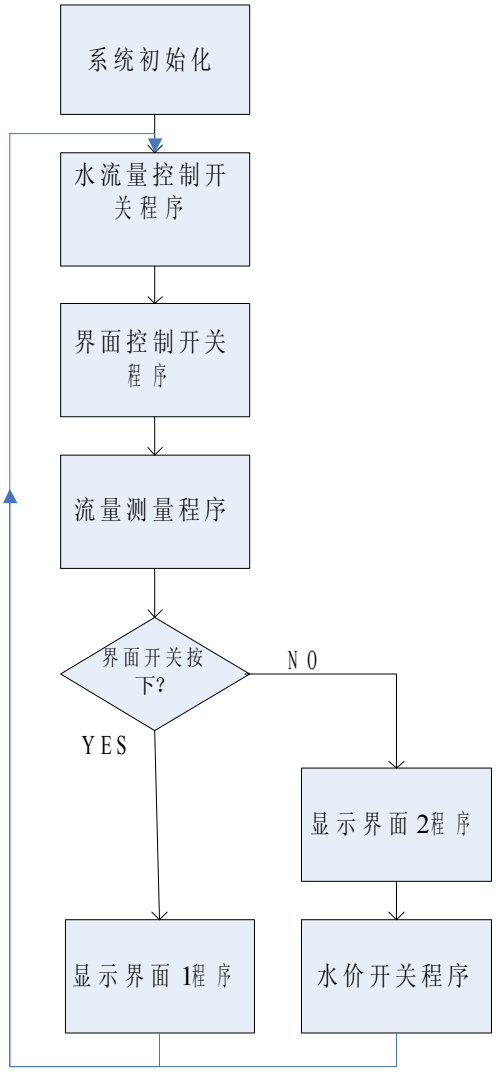


图 3-1　总流程图

3.2　温度程序模块

3.2.1　温度数据转换程序

由温度传感器 **DS18B20** 采集的温度数据读取后温度的低位和高位分别存在主芯片 **SCT89C52** 存储器中。其中依定传感器的设计，读出的数据最高位为 **0** 时温度为正，温度为 **1** 时，温度是负数。是以对温度数据处理，将温度数据高位和低位整合在一起，在判断温度的正负即可。

如图 **3-2** 所示：

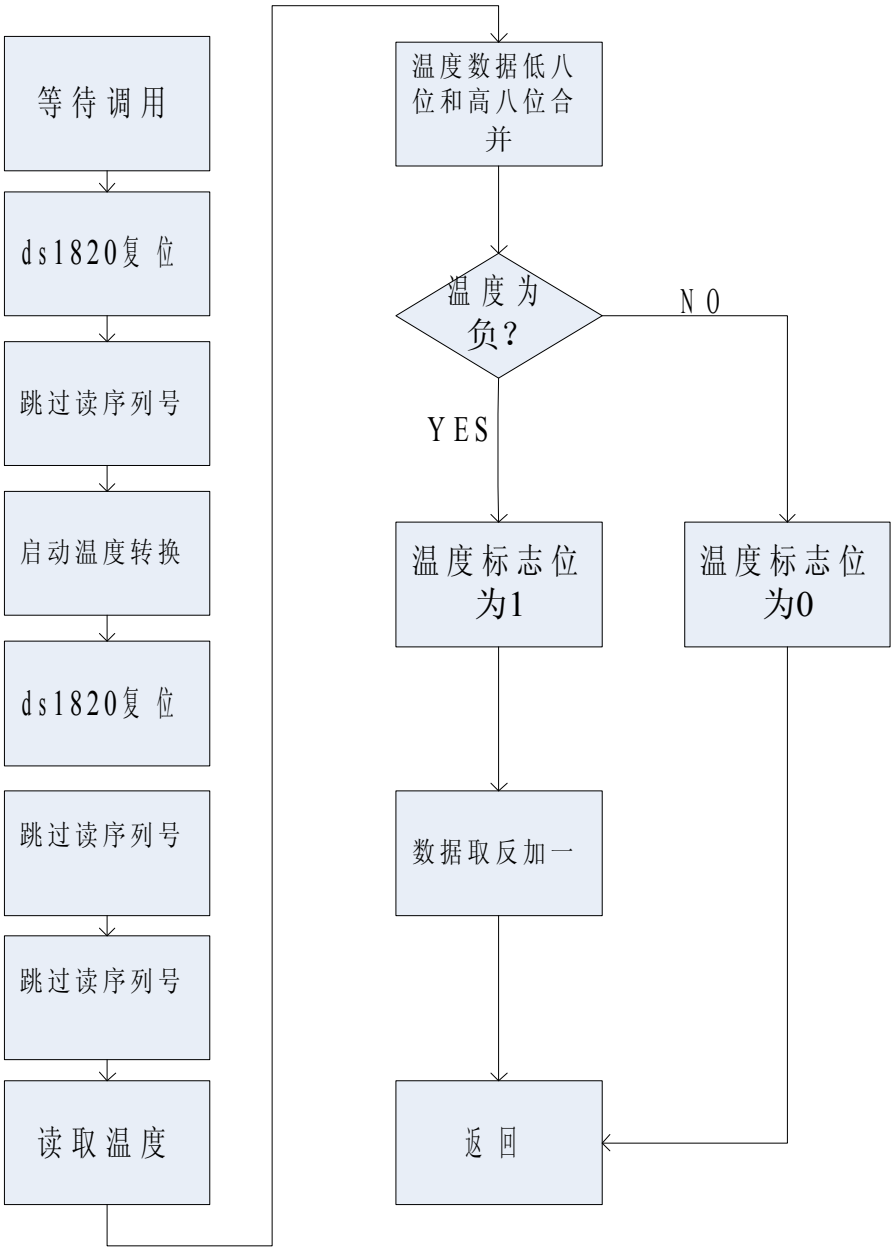


图 3-2　温度测量流程图

3.2.2　温度数据显示前处理程序

将已处理好的温度数据的首先判断它的正负，然后再去运行其他的代码。在这里，程序还设定了温度报警。温度报警本来可以在 **DS18B20** 中对芯片进行设置，但考虑到对芯片的熟悉度不够，容易出错，而在程序中设定比较容易理解，写起来也不会太难。还有实际水的温度不可能超过 **100** 度，所以测得 **100** 度以上的温度值就会显得多余，在程序中只要设定最高 **99.9** 度即可。再就是美观上的设定，测得的温度在为个位时，十位为 **0** 就会看起来不太美观，只要把十位设定看不见即可。

如图 **3-3** 所示：

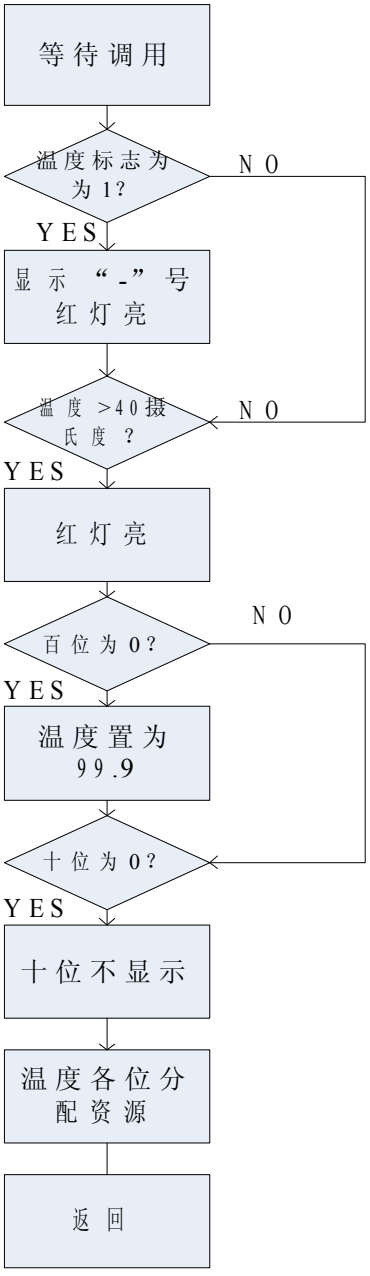


图 3-3　数字不显示流程图

3.3　水流量程序模块

水流量的测量主要依靠对得到的频率处理，由流量计在一段时间下产生高电平的个数决定，即 **Q**（流量）**=F**（频率）**/R**（商家设定值）所以只要在单片机中设定一定的时间，并在该时间之下计算出得到的高电平即可。在本程序中 **STC89C52** 的两个定时器 **T0** 为计数状态，**T1** 为计时状态，这样方可测量流量。

3.3.1　水流量的读取程序

中断程序运行的时间到，就可以读取计数器中的数值，将下数值读出后把计数器赋值为 **0**，等待下一次的取出，然后进入对读出数据的处理程序。

如图 **3-4** 所示：

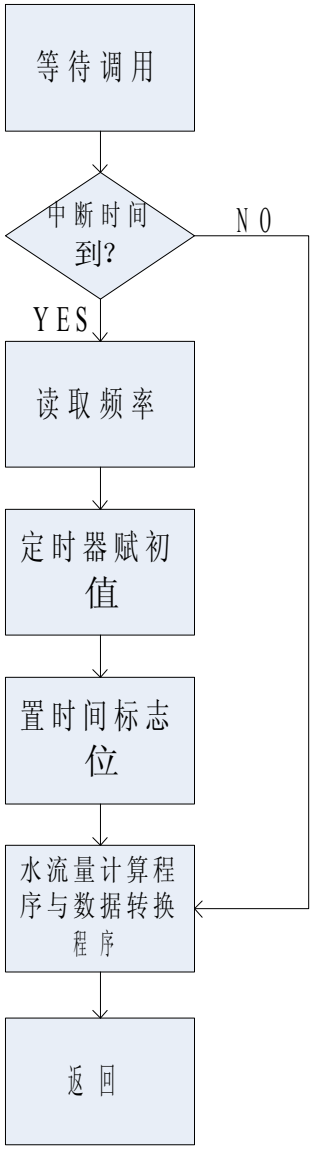


图 3-4　水流量测量流程图

3.3.2 水流量数据的处理程序

在预定时间到，即已经取出了定时器的数值。我们计算的水流量最大是以每吨来计算并显示的，故一个整形或长整型的数不够，故先把它放在了一个 **int** 变量的存储空间内，做水流量的前三位，在定义一个长整型的数，把它作为水流量的后六位，这样计算起来也比较容易，数据也不会起冲突，也是为将要计算水价做准备。设定该水流量的最大计数为 **250** 吨，超过了定值，则会从 **0** 开始。

如图 **3-5** 所示：

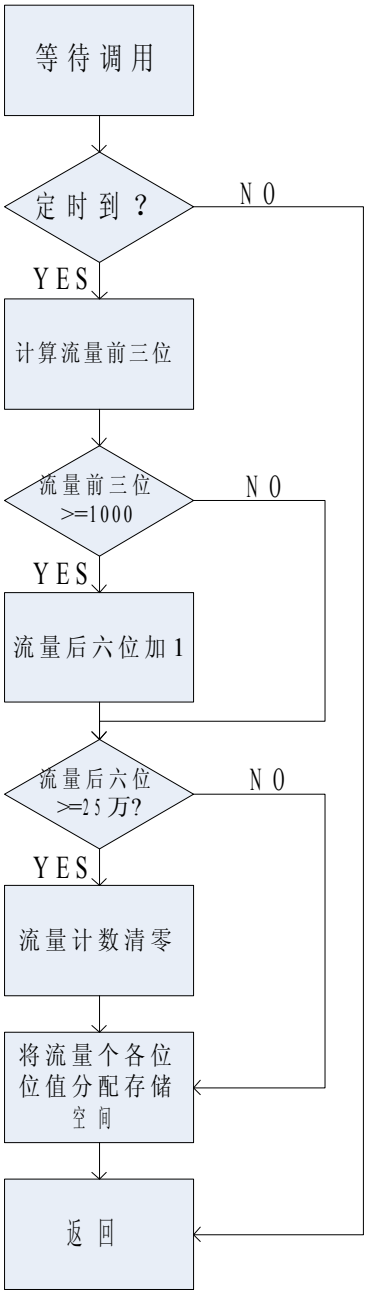


图 3-5　水流量数据处理流程图

3.4 显示程序

在程序中只是计算出流量值，温度值以及和资源分配的情况，一些字母的显示，都要另加，这样才能尽善尽美的表达出显示出来的意思。在显示程序中加入了温度的测量，所以在主程序中不必再测量温度。在第一页主要显示的是温度、水费和水流量的后六位，最后以 **t** 结尾。第二页主要显示温度、单位水价和水流量的前三位，最后以 **kg** 结尾。

如图 **3-6**、图 **3-7** 所示：

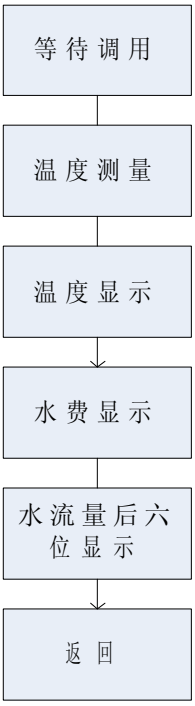


图 3-6 第一页显示流程图

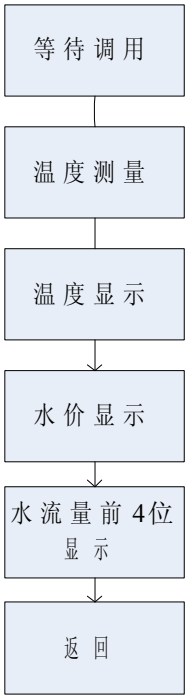


图 3-7 第二页显示程序流程图

3.5 小结：

本课题的程序全部用 **C** 语言来编写。在编写过程中，会借鉴网络的编程资料，会将整个程序进行整体的规划，会对错误的程序反复修改，尽善尽美的达到课题要求。其中主要的问题在于怎样节省存储空间，怎样最大限度的利用芯片，怎样的程序在现实实物中能够很好的运行成功。这就主要归结于对芯片的熟悉程度和对程序的应用程度。经过一番努力后，最终把程序做了出来。在做程序的过程中，遇到自己解决不了的难题就和同学们一起商量探讨。这次的课题充分的锻炼了自己。

4　调试结果记录

4.1　温度测量

表 4-1　温度测量

调试环境	温度调试
自来水	7℃
洗澡水	52℃

4.2　流量测量

由于自来水水流不够均匀，水流量传感器较小，测量不会标准，故测试改水流量测量用气流代替，每次结果不一。因原本设计要求已达到，本设计硬件部分和软件部分结束。

5　总结

这次设计的制作，充分的考验了自己的综合能力。从硬件到软件的制作过程中，问题多多，但经过一番努力后，所有的问题都解决。

硬件的问题主要是设计电路这一块。因为这次设计中有第一次触摸的芯片和电路，这都不得不自己找资料去查找其芯片的结构和用法。当电路设计好了以后，元器件不多，自己就用万能板做电路。由于焊接工艺不好，使得电路有虚焊、漏焊，甚至焊错的接口，短路也时有发生。查找错误也很慢。让自己看到了在动手操作能力上的不足。

软件部分的问题就是查找资料太多，但整理出来有用的很少，还有液晶屏 **1602** 和 **DS18B20** 温度传感器时序图看起来始终很费劲。网上那个有很多它们的资料，看懂网上的程序也很艰难，发现自己对程序上面也只是略懂皮毛。

毕业设计是一次对学校对学生专业技能的验收，通过这次的实际操作，让我从硬件到软件，从思考到动手，从理论到实践，让我对电子专业的更加深刻的了解，学习的专业知识更加的巩固，更大的好处是让我知道自己哪些的不足，哪些是自己的强项，失败和成功都是我的财富。也让我知道在关键时刻要充分的发挥自己的能力，在平时要多劳动，多学习，多看多想多实践。马上就要走上社会的我们更加体会到专业对自己重要性。

在这次毕业设计中，我重要的收获就是其中许多次的失败。由失败走向成功是很让人兴奋的一件事，其中妙味无穷。由于有问题，所以我找过同学一起探讨，上网找过资料来搜集，找过老师来辅导。使我感觉到过的很充实，希望在以后工作中也能持续保持自己这样的积极性。所以这次非常感谢老师同学们的帮助。

6　成果展示

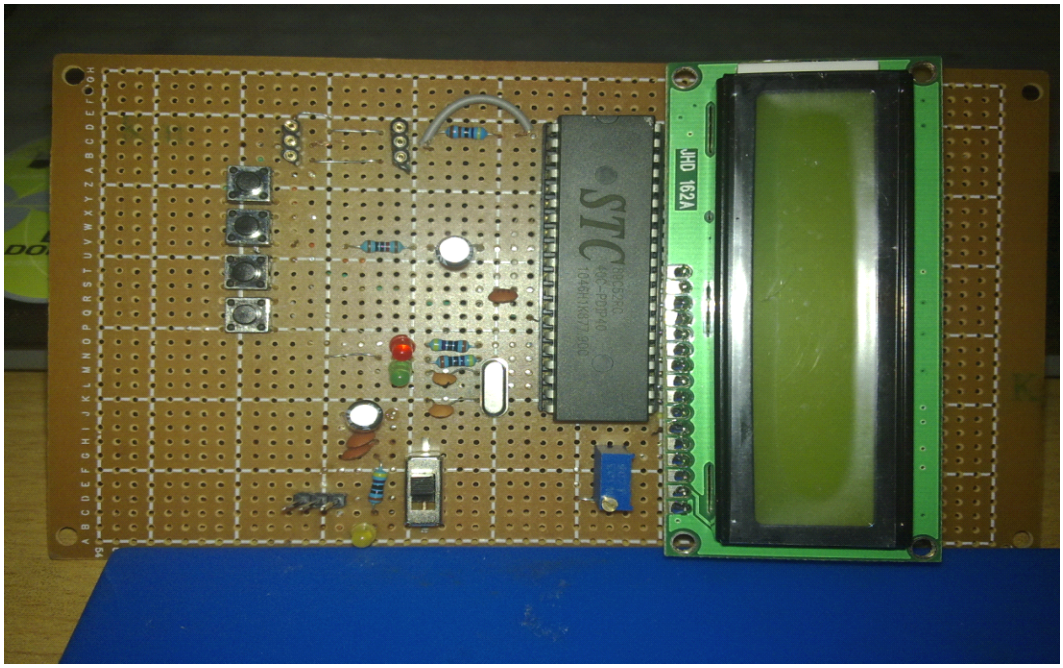


图 6-1　实物

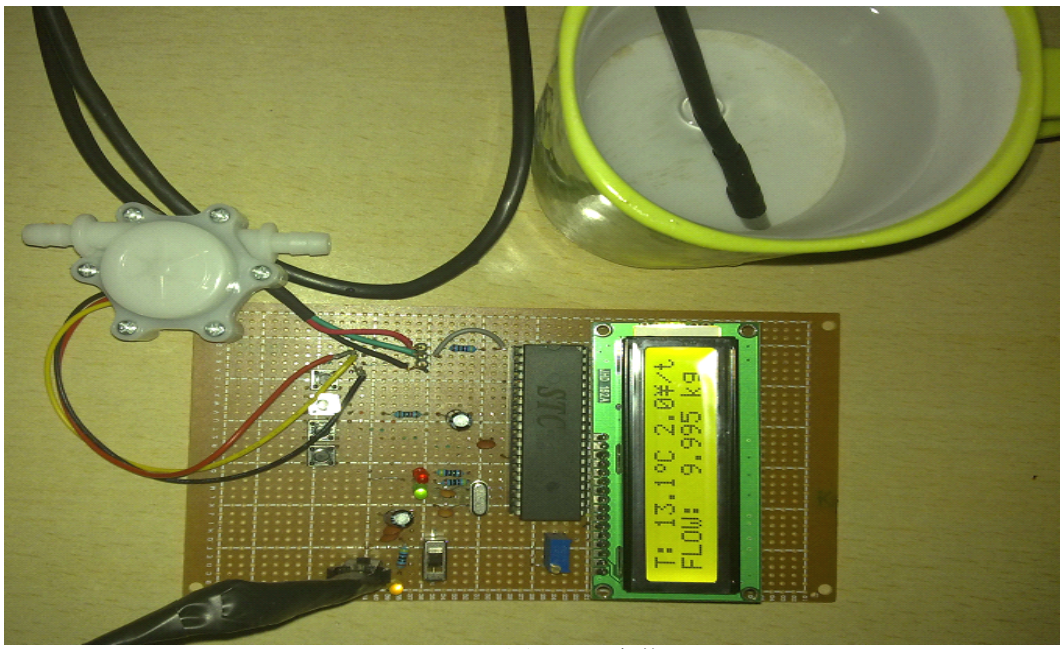


图 6-2　实物

结论与谢辞

本次毕业设计通过指导老师的指导，以及自己的努力，我才能完成任务。不但让我增长了见识也让我真正感受到理论与实际相结合的成果，受益匪浅。

感谢我的指导老师，他给予我许多的帮助和鼓励。我还要感谢在设计过程中所有给予我真诚帮助的其他老师和同学。

参考文献

[1] 张菁，基于单片机温度控制系统方案的研究。2007（9）。

[2] 楼然苗，51 系列单片机设计实例. 北京航空航天大学出版社，2003。

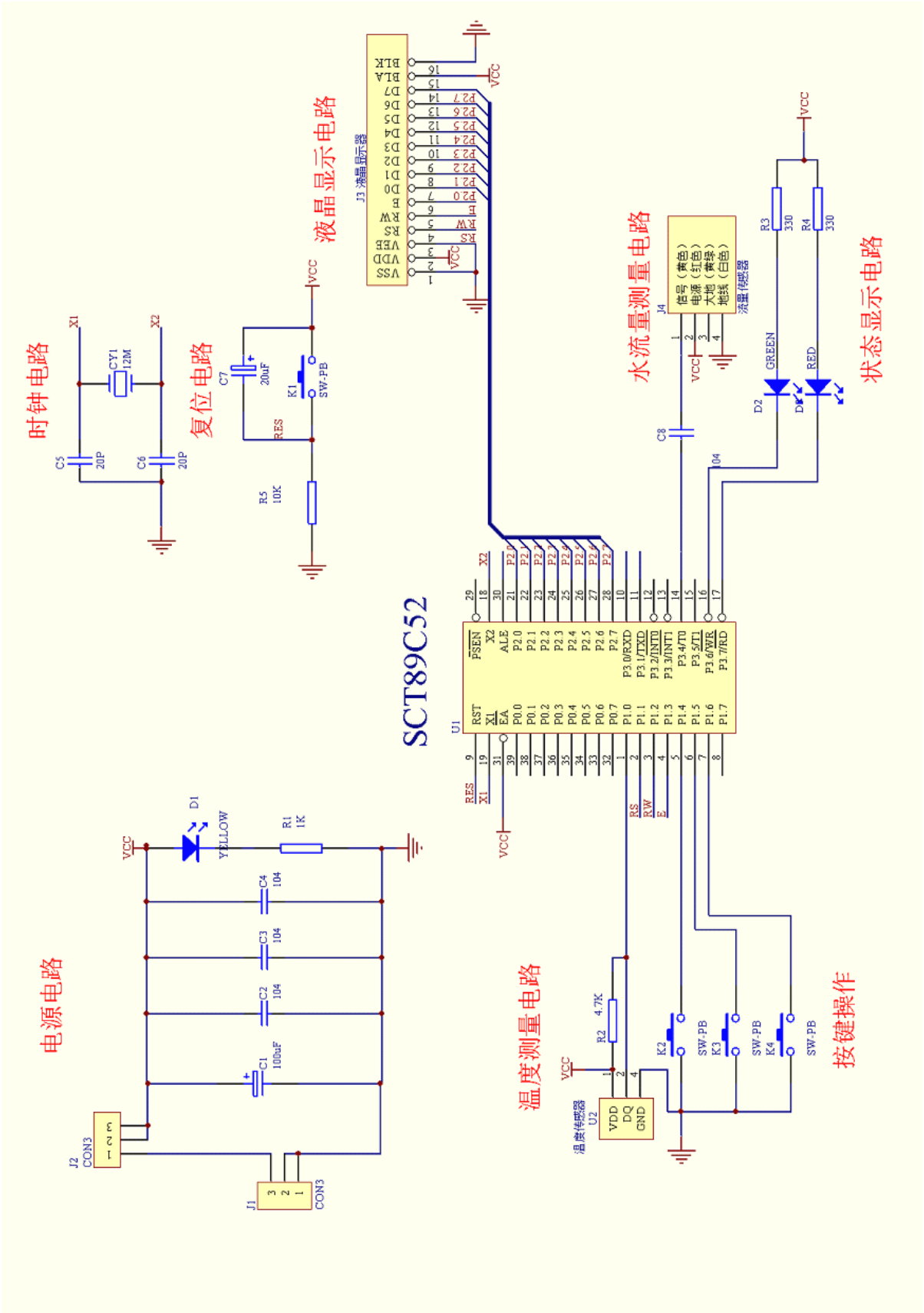
[3] 苏铁力、关振海等.传感器及其接口技术[M].中国石化出版社,1998.

[4] 宋亚伟、李恒宗，基于 DS18B20 的温度采集控制电路，2008,3,7（9）。.

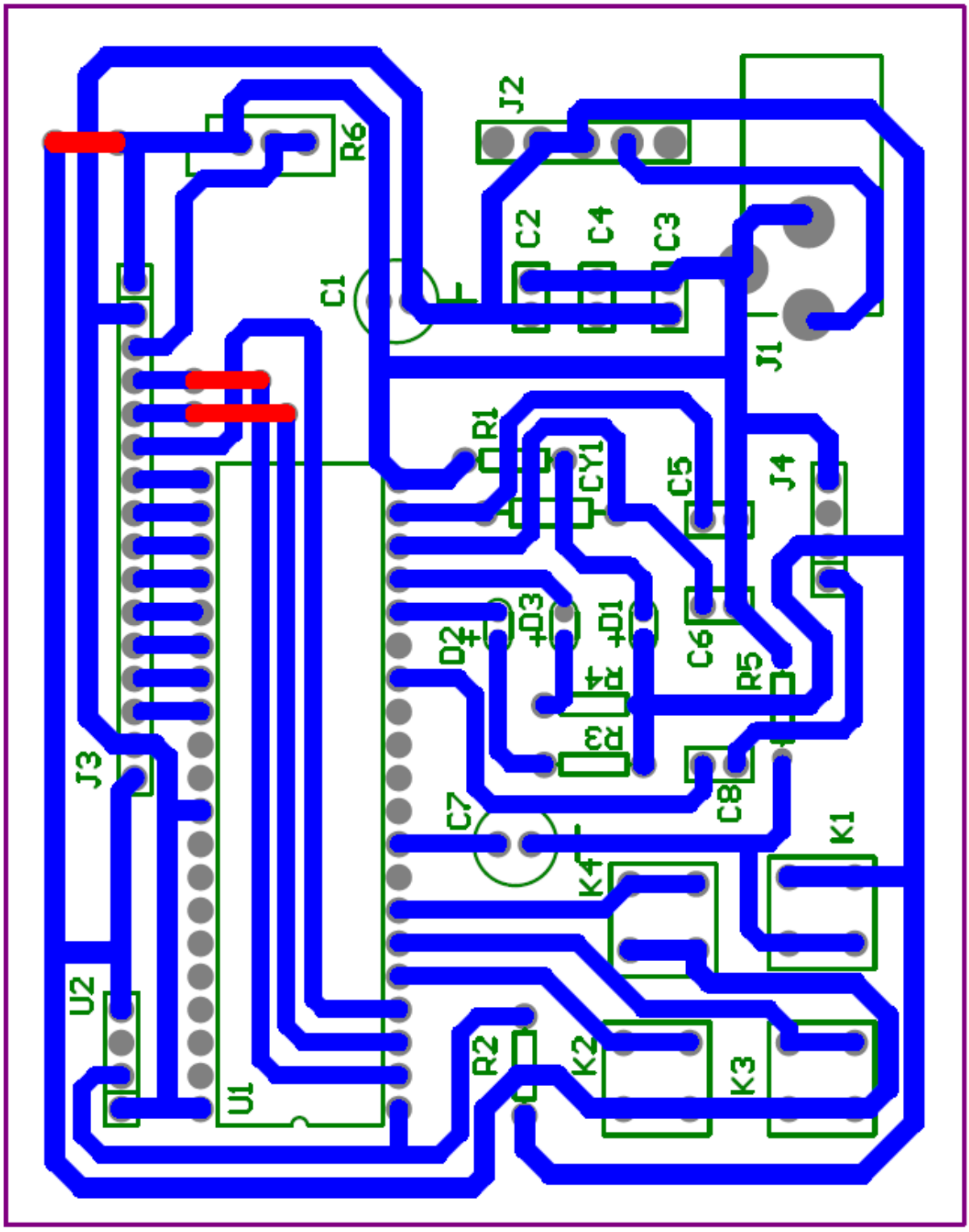
[5] 肖晴，液晶显示温度的控制，2005（2）。

[6] 谭浩强，C 程序设计(第二版). 清华大学出版社. 1999 年 12 月第 2 版

附件 1 电路原理图


液晶显示电路
时钟电路
复位电路
水流量测量电路
状态显示电路
电源电路
温度测量电路
按键操作
SCT89C52
J3 液晶显示器
VSS
VDD
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7
BLA
BLK
X1
X2
CY1 12M
C5 20P
C6 20P
C7 20uF
K1 SW-PB
R5 10K
RES
VCC
J4
信号（黄色）
电源（红色）
大地（黄绿）
地线（白色）
流量传感器
R3 330
R4 330
GREEN
RED
D2
D4
C8
U1
P0.0
P0.1
P0.2
P0.3
P0.4
P0.5
P0.6
P0.7
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P2.0
P2.1
P2.2
P2.3
P2.4
P2.5
P2.6
P2.7
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
PSEN
ALE
EA
D1 YELLOW
R1 1K
C4 104
C3 104
C2 104
C1 100uF
J2 CON3
J1 CON3
R2 4.7K
U2
VDD
DQ
GND
温度传感器
K2 SW-PB
K3 SW-PB
K4 SW-PB

附件 2　PCB 图


J2
R6
C2
C4
C3
C1
J1
R1
CY1
C5
J4
D3
D1
D2
C6
R4
R5
R3
J3
C7
C8
K1
K4
U2
R2
K2
K3
U1

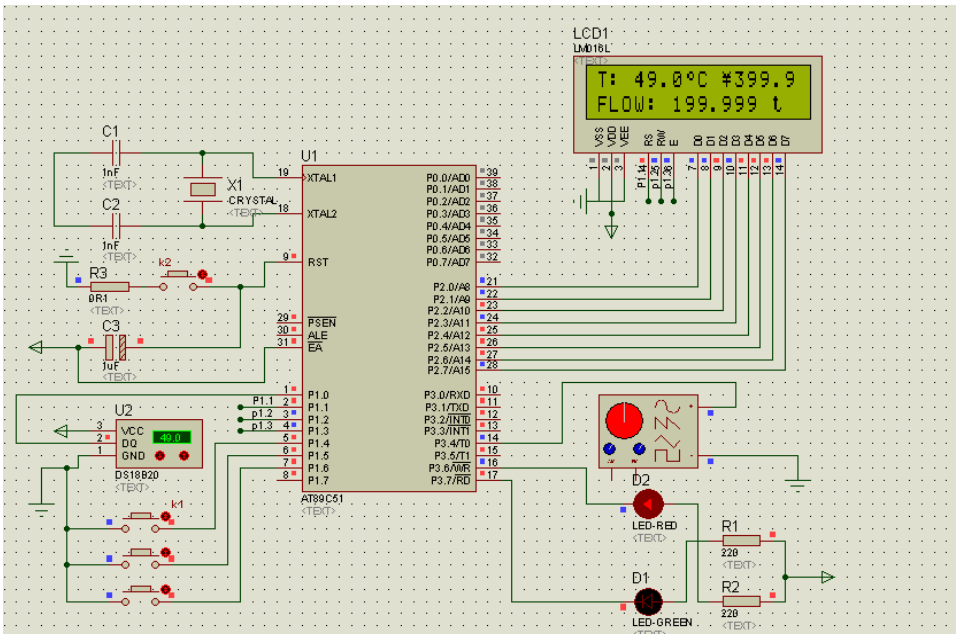
附件 3　仿真图

液晶显示第一页面：

① 温度 49℃>40℃，红灯亮报警

② 总用水费 399.9 元

③ 总用水量的前六位 199.999 吨

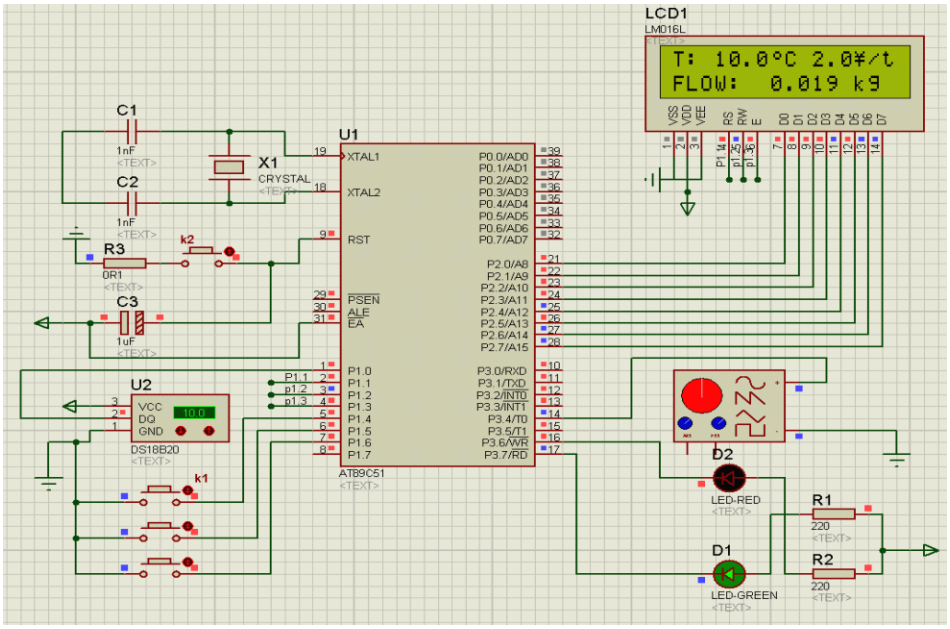


液晶显示第二页面：

① 温度 10℃<40℃

② 水价/吨 2 元

③ 总用水量的后 4 位 0.019kg



附件 4 元器件清单

元器件名称	型号	数量（个）
单片机	STC89C51	1
温度传感器探头	DS18B20	1
液晶显示器	LCD1602	1
晶振	11.0596MHZ	1
电阻	500 欧	2
电阻	1K	1
电阻	4.7K	1
电阻	10K	1
瓷片电容	22pF	2
瓷片电容	0.1uF	2
电解电容	20uF	2
电解电容	100uF	1
发光二极管	红色	1
发光二极管	黄色	2
发光二极管	绿色	1
电源借口	5V	1
拨头开关		1
按键		3
总计		25

附件 5 程序清单

```
#include<reg51.h>
#define uchar unsigned char
#define uint unsigned int
sbit DQ=P1^0;     //ds18b20 与单片机连接口
sbit RS=P1^1;
sbit RW=P1^2;
sbit EN=P1^3;
sbit p37=P3^6;       //流量正常运行接口
sbit p36=P3^7;       //温度报警接口
sbit p14=P1^4;     //功能按键接口
sbit p15=P1^5;
sbit p16=P1^6;
uchar code table[]={0x30,0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x39,0x20}; //0-9
数字和显示无
unsigned long int a;
float     f,zong;
uint       tvalue;
uchar     n1,n2,n3,n4,n5,n6,n7,n8,n9;
uchar     tflag,d,g,kai,j,j1,j2,b;
/************************延时子程序**************************/
void delay   (uint z)
  {  int x,y;
     for(x=z;x>0;x--)
     for(y=120;y>0;y--) ;
  }
/***************延时 1us 子程序**********************/
  void delay_18B20(unsigned int i)//延时 1 微秒
{
    while(i--);
```

```
}
 /***************液晶显示写指令*******************************/
void wr_com(unsigned char com)    //写指令
 {
    delay(1);     //延时 1ms
    RS=0;         //写命令设置
    RW=0;         //并行数据的读写
    EN=0;         //使能为 0
    P2=com;           //输入命令
    delay(1);     //延时 1ms
    EN=1;         //使能为 1
    delay(1);     //延时 1ms
    EN=0;         //使能为 0
  }
  /*************液晶显示写数据************************/
void wr_dat(unsigned char dat)//写数据
{
    delay(1);     //延时 1ms
    RS=1;         //写数据设置
    RW=0;         //并行数据的读写
    EN=0;         //使能为 0
    P2=dat;       //输入数据
    delay(1);     //延时 1ms
    EN=1;         //使能为 1
    delay(1);     //延时 1ms
    EN=0;         //使能为 0
}

/***********水流量计算程序与数据转换程序*****************/
void loop ()
```

```
{
   uint       zong1;
    if(g==0)   //判断时间是否到
        {
        zong=(b/100.0)+zong; //得出总流量的后三位
    }

   g=1; //将标志为为 1
   if(zong>=1000) //判断总流量后三位是否大于 1000
   {
        a++;     //总流量前六位加 1
        zong=0; //后三位清零
   }
   if(a>=1000000) //总流量前六位大于 1 百万
   {
        a=0; //总流量前六位清零
   }
   zong1=(int)zong ; //总流量后三位转换为整形
   n9=a/100000;           //总流量 9 位全部分配
   n8=a%100000/10000;
   n7=a%10000/1000;
   n6=a%1000/100;
   n5=a%100/10;
   n4=a%10;
   n3=zong1%1000/100;
   n2=zong1%100/10;
   n1=zong1%10;
}
/**********************水费计算与数据转换************************/
void    shuifei()
```

```
  {
     uchar h1,h2,h3,h4;
     h4=(j*a)/1000000;         //水费显示
     h3=(j*a)%1000000/100000;
     h2=(j*a)%100000/10000;
     h1=(j*a)%10000/1000;
      wr_com(0x80+10);        //水费显示
        wr_dat(0x5c);
        wr_dat(table[h4]);
        wr_dat(table[h3]);
        wr_dat(table[h2]);
        wr_dat(0x2e);            //显示小数点
        wr_dat(table[h1]);
  }
/**********************ds1820 复位子程序*****************/
void ds1820rst()
  {
   unsigned char x=0;
   DQ = 1;                       //DQ 复位
   delay_18B20(4);              //延时
   DQ = 0;                       //DQ 拉低
   delay_18B20(100);          //精确延时大于 480us
   DQ = 1;                       //拉高
   delay_18B20(40);           //延时
     }
/***********ds1820 读数据子程序*******************/
uchar ds1820rd()          //读数据
   {
      unsigned char i=0;
      unsigned char dat=0;
```

```
    for (i=8;i>0;i--)       //读温度 2 进制 8 次
    {
            DQ = 0;                    //给脉冲信号
             dat>>=1;       //将温度数据转移到 dat
             DQ = 1;        //给脉冲信号
        if(DQ)                      //数据转换
            dat|=0x80;
        delay_18B20(10);
      }
     return(dat);
  }
/**************ds1820 写数据子程序********************/
void ds1820wr(uchar wdata)
  {
     unsigned char i=0;
     for (i=8; i>0; i--)        //写数据 2 进制 8 次
    {
      DQ = 0;                    //给脉冲信号
      DQ = wdata&0x01;           //数据传送
      delay_18B20(10);           //延时
      DQ = 1;                    //给脉冲信号
      wdata>>=1;                 //数据移位
    }
 }
 /*****************ds1820 温度转换程序*****************************/
uint read_temp()
{
  uchar a,b;
  ds1820rst();                 //ds1820 复位
  ds1820wr(0xcc);              //跳过读序列号
```

```
   ds1820wr(0x44);              //启动温度转换
   ds1820rst();                 //ds1820 复位
   ds1820wr(0xcc);              //跳过读序列号
   ds1820wr(0xbe);              //读取温度
   a=ds1820rd();                //将温度数据给 a
   b=ds1820rd();                //将温度数据给 b
   tvalue=b;                    //将温度数据转移
   tvalue<<=8;
   tvalue=tvalue|a;
      if(tvalue<0x0fff)     //是正温度吗
    tflag=0;                    //正温度标志
      else
     {
        tvalue=~tvalue+1;   //负温度，取反加 1
        tflag=1;                //负温度标志
     }
      tvalue=tvalue*(0.625);//温度值扩大 10 倍，精确到 1 位小数(0.0625)
   return(tvalue);
   }
/****************ds1820 温度显示子程序************************/
void ds1820disp( )
  {
     uchar t1,t2,t3,t4;
        t3=tvalue/1000;         //百位数  (0-9 是 0x30-0x39)   温度值放大了十倍
        t2=tvalue%1000/100; //十位数
        t1=tvalue%100/10;     //个位数
        t4=tvalue%10;           //小数位
        if(tflag==1)
       {
           wr_dat(0x2d);//负温度显示负号:-
```

```
}
   else
        p37=1;
 if(t3!=0)
 {
    t1=9;
    t2=9;
    t3=10;
    t4=9;
    p37=0;
 }
 if(t2>=4)
 p37=0;
 else
 p37=1;
  if(tflag==1)
 {
       p37=0;     //负温度显示负号:-
}
   if(t3==0)
 {
       t3=10;              //如果百位为 0，不显示
    if(t2==0)
    {
    t2=10;              //如果百位为 0，十位为 0 也不显示
    }
 }
 wr_dat(table[t3]);//显示百位
 wr_com(0x80+3);
 wr_dat(table[t2]);//显示十位
```

```
      wr_dat(table[t1]);//显示个位
      wr_dat(0x2e);        //显示小数点
      wr_dat(table[t4]);//显示小数位
}
   /**********************流量程序*******************************/
   void liuliang()
  {
   while(d>=2)             //时间到?
    {
      d=0;                  //d 清零
      b=TH0*256+TL0;                //读取频率
      TH0=0x00;             //计数器清零
      TL0=0x00;
      g=0;                  //时间标志位
     }
     loop();          //水流量计算程序与数据转换程序
  }
/**************第一页显示子程序*************************/
void     display1()
{
      wr_com(0x80);         //温度显示
       wr_dat(0x54);         //显示 T
       wr_dat(0x3a);       //显示：
       read_temp();          //读取温度
       ds1820disp( );       //温度显示
       wr_dat(0x00);         //显示°
       wr_dat(0x43);       //显示 C
     shuifei();
     wr_com(0x80+0x40);   //水流量（t）显示
      wr_dat(0x46);
```

```
    wr_dat(0x4c);
    wr_dat(0x4f);
    wr_dat(0x57);
    wr_dat(0x3a);
    wr_dat(0x20);
    wr_dat(table[n9]);
    wr_dat(table[n8]);
    wr_dat(table[n7]);
    wr_dat(0x2e);          //显示小数点
    wr_dat(table[n6]);
    wr_dat(table[n5]);
    wr_dat(table[n4]);
    wr_dat(0x20);          //显示小数点
    wr_dat(0x74);
    wr_dat(0x20);
}
 /**************第二页显示子程序**********************/
 void display2()
 {
   wr_com(0x80);           //温度显示
    wr_dat(0x54);          //显示 T
    wr_dat(0x3a);         //显示：
    read_temp();           //读取温度
    ds1820disp( );        //温度显示
    wr_dat(0x00);          //显示°
    wr_dat(0x43);         //显示 C
   wr_com(0x80+10);    //水价显示
   wr_dat(table[j2]);
   wr_dat(0x2e);
   wr_dat(table[j1]);
```

```
    wr_dat(0x5c);
    wr_dat(0x2f);
    wr_dat(0x74);
    wr_dat(0x20);
    wr_com(0x80+0x40+5);//水流量（kg）显示
    wr_dat(0x20);
    wr_dat(0x20);
    wr_dat(table[n4]);
    wr_dat(0x2e);          //显示小数点
    wr_dat(table[n3]);
    wr_dat(table[n2]);
    wr_dat(table[n1]);
    wr_dat(0x20);
    wr_dat(0x6b);
    wr_dat(0x67);
  }
/*****************控制流量测量的开关程序**********************/
  void kaiguan1()
  {
    if(p14==0)              //按键按下？
    {delay(2);
    if(p14==0)
        { delay(2);

          p36=~p36;       //P3.6 取反
          TR0=~TR0;       //TR0 取反
          while(p14==0);//按键松开
          }
      }
  }
```

```
/**************界面切换开关程序********************/
void kaiguan2()
{
   if(p15==0)              //按键按下？
   {
            delay(2);
      if(p15==0)
       { delay(2);
          kai=~kai;        //P3.6 取反
          while(p15==0);//按键松开？
       }
   }
}
/*****************水的单价控制开关程序************************/
void kaiguan3()
{
   if(p16==0)              //按键按下？
   {delay(2);
   if(p16==0)
       { delay(2);
         j++;          //水价加一
         if(j>40)    //水价大于 40
         j=0;          //水价清零
         j2=j/10;    //水价分位
         j1=j%10;
        while(p16==0);//按键松开？
        }
    }
}
/**********************程序初始化*****************************/
```

```
void lcd_init ()
{
    TMOD=0x15;                  //设定定时器 0 为计数功能，定时器为 1 为定时功能
    TH1=(65536-50000)/256;//定时器 1 赋出值 50ms
    TL1=(65536-50000)%256;
    TH0=0x00;                   //定时器 0 赋初值 0 次
    TL0=0x00;
    EN=0;                       //液晶使能端为 0
    wr_com(0x38);               //液晶初始设置
    wr_com(0x0c);
    wr_com(0x06);
    wr_com(0x01);
    wr_com(0x40);
    wr_dat(0x06);               //写 CGRAM 写字模
    wr_dat(0x09);
    wr_dat(0x09);
     wr_dat(0x06);
    wr_dat(0x00);
    wr_dat(0x00);
    wr_dat(0x00);
    wr_dat(0x00);
    zong=0;                     //总流量（带小数点）
    EA=1;                       //开启总中断
    ET1=1;                      //开启定时器 1 中断
    TR1=1;                      //开启定时器 1
    TR0=0;                      //开启计数器 0
     p16=1;                     //p16 为 1
    p15=1 ;                     //p15 为 1
     p14=1;                     //p14 为 1
    zong=995;                   //总流量的后三位赋初值
```

```
    a=199999;                          //总流量的前六位赋初值
    j=20;                              //水价赋初值
    j2=2;                              //水价个位赋初值
    j1=0;                             //水价小数位赋初值
    kai=0;
}
/********************主程序*****************************/
void main()
 {
    lcd_init();       //初始化显示
    while(1)
     {
    kaiguan1();     //控制水流量的测量
    kaiguan2();     //控制显示界面
    liuliang();     //流量测量
    if(kai==0)     //判断界面开关是否按下
    display1();    //显示第一页界面
    else
         {
          kaiguan3(); //判断水价开关是否按下
          display2(); //显示第二页界面
         }
     }   }
 /********************定时器 1 中断**********************/
 void time1() interrupt    3
 {
    TH1=(65536-50000)/256;//定时器 1 初值定时 50ms
    TL1=(65536-50000)%256;
    d++;                      //20ms 后 d 加 1
 }
```