

DS18B20 在粮情监控系统中的应用

陈涛, 王仲东

(华中科技大学 自动控制系, 湖北 武汉 430074)

摘要: 介绍了一种基于数字式温度传感器 DS18B20 的粮库温度监控系统, 作者针对 DS18B20 对 I/O 时序要求严格, 大多数编程人员不得不用汇编语言编写接口程序的特点, 利用 `_nop()` 函数实现了 C 语言接口程序的设计, 为整个系统软件设计带来了方便。

关键词: DS18B20; 监控系统; 单总线

中图分类号: TP29 **文献标识码:** A **文章编号:** 1007 - 855X(2003)05 - 0093 - 04

Application of DS18B20 to Grain Monitor System

CHEN Tao, WANG Zhong-dong

(Dept. of Control Science & Engineering, HUST, Wuhan 430074, China)

Abstract: The application of DS18B20 to the monitor system of a major grain depot is explicitly introduced. DS18B20 system strictly requires the correct sequence and the most programmers to write program with assembly language. By using `_nop()` function, the fetch program design of C is realized, which brings much convenience to the whole program design.

Key words: DS18B20; monitor system; single bus

0 引言

对于中国这样的人口大国, 粮食的生产和储备显得至关重要, 要完好的储备粮食, 必须对粮库中的温度、湿度、二氧化氮浓度等因素进行实时监控, 而监控系统能否准确工作与传感器的精度及稳定性息息相关的, 因此传感器的选型成为关键问题。

在众多应用于温室环境监测的温敏元件中, 温敏电阻虽然成本低, 但后续电路复杂, 且需进行温度标定; 电流型集成温度传感器 AD590 也因其输出为模拟信号, 且输出信号较弱, 需后续放大及 A/D 转换电路, 若采用普通运放则精度难以保证, 而测量放大器价格偏高, 这就使系统的成本升高。本系统采用美国 DALLAS 公司的产品——可编程单总线数字式温度传感器 DS18B20 进行温度采集, 通过 485 总线与上位机进行通讯, 上位机通过对数据的分析下达相应的控制命令, 从而达到对粮情进行监控。

1 系统结构

本系统是为某国家储备粮库设计的, 由于库存量大, 每栋库房需采集 500 个温度点, 系统采用图 1 所示的方案, 从机负责采集温度和湿度并向上微机发送数据, 上微机通过对温度和湿度值进行分析, 下达相应控制命令。从机接到命令后执行命令, 如自动开关门窗, 打开通风设备等。

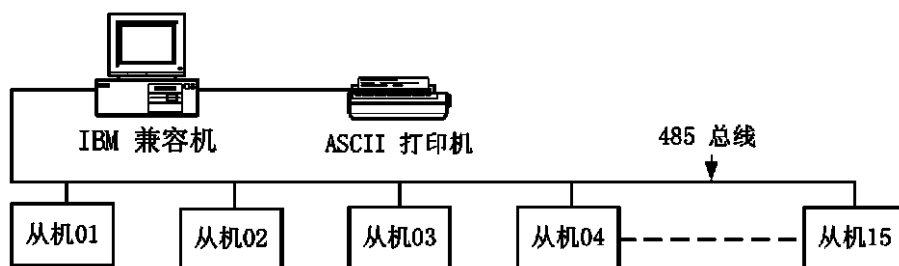


图 1 系统结构图

2 从机硬件电路设计

由于要采集的温度点较多,本系统选用 DS18B20 作为温度传感器,DS18B20 具有如下优点.

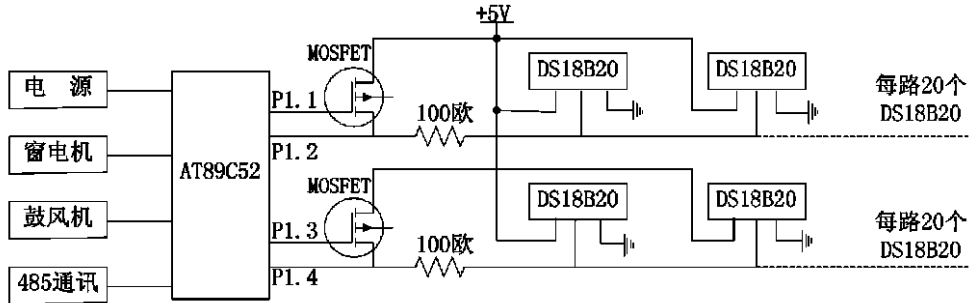


图2 硬件接口电路

1) 精度高. 12 位二进制转换结果,确保正负 0.5 温差和 0.062 5 度的分辨率.

2) 全数字化. 直接将数字信号传给 CPU,传输可靠,避免了模拟方式的干扰问题.

3) 连线少. 仅有 3 根连线: +5 V 电源,地线和 1 根数字 I/O 总线. 如采用寄生电源方式,DS18B20 会从数字 I/O 总线获取寄生电源,则仅连接 I/O 线和地线即可.

虽然 DS18B20 有诸多优点,但使用起来并不容易. 由于采用单总线数据传输方式,DS18B20 的数据 I/O 均由同一条线完成,因此,对读写的操作时序要求严格. 一般情况下需要用汇编语言编写接口程序,而如今单片机编程已广泛采用 C 语言. 这样为整个系统软件设计带来很多不便,作者在对 C51 编译生成的程序代码分析的基础上,依据 C51 的编译特点,采用 C 语言编写接口程序,从而达到统一用 C 语言完成系统软件设计的目的. 系统采用美国 ATMEL 公司的 AT89C52 单片机. 由于所要监测的粮库储量大,要监测的点比较多(一共有 500 个点)的特点,采用外部供电方式,理论上采用外部供电可以在一根数据总线上挂 256 个 DS18B20,但实际应用中发现,如果挂接 25 个以上的 DS18B20 仍旧有可能产生功耗问题. 另外单总线的长度不宜超过 80 m,否则也会影响数据的传输. 在这种情况下我们采用分组的方式,用单片机的多个 I/O 口来驱动多路 DS18B20. 在实际应用中还可以如图 2 中所示的那样,使用一个 MOSFET 将 I/O 线直接和电源相连,起到上拉作用. 图中采用 P1 口做 I/O 口,共接了四路,每路接 20 个 DS18B20,一共可测 80 个点,其中中间两路在图 2 中省略.

由于 AT89C52 内部含有 4 K 可重复的 FLASH 存储器,可进行 1 000 次的擦写操作,故在开发过程中可以十分容易地进行程序修改,大大缩短了开发周期. 采集的数据通过 485 总线传送到上层计算机进行计算分析,然后根据分析结果向单片机发出控制命令,再由单片机控制直流 24 V 电机实现对粮仓门窗的开关. 系统还可以进行监测方式的选择,主要有循环显示各路测量值、单独显示某一路或某个点的测量值、给每一路设定上下限报警值等. 由于篇幅的原因,本文仅对温度采集部分给予详细说明.

3 软件设计

系统软件采用模块化程序设计,主模块为基于 Modbus 协议的通讯程序,Modbus 协议采用主从结构的通讯方式,规定总线上有一个主机,多个从机,每个从机有一个唯一的地址. 工作时采用命令/应答的通讯方式,每一种命令帧都对应着一种应答帧. 主机向要访问的从机发出命令帧,地址匹配的从机进行响应,并向主机发出与命令帧对应的应答帧. 温度采集模块负责温度采集并向上位机发数据,基本过程如下:当上位机发温度转换命令时,所有从机执行温度转换并等待上位机发读数据命令,当上位机发读数据命令时,从机从各个 DS18B20 中读出温度值依次发向上位机,这里详细说一下使用 DS18B20 采集温度的过程.

对 DS18B20 处理的顺序如下:先初始化,然后发送 ROM 命令,最后发送功能命令. 初始化包括主机发出复位脉冲(通过将总线拉低至少 480 μ s 来实现)随即主机等待 DS18B20 发回的存在脉冲. DS18B20 则从检测到复位脉冲的上升沿开始等待 15 ~ 16 μ s 后通过将单总线拉低 60 ~ 240 μ s 实现存在脉冲的发送. 初始化完成后即可发送 ROM 命令,包括搜索 ROM 命令 (F0H) 读 ROM 命令 (33H) 符合 ROM 命令 (55H)

跳过 ROM 命令 (CCH) 及报警搜索命令 (ECH). 随后即可发送功能命令, 包括温度变换命令 (44H) 写暂存命令 (4EH) 和读暂存命令 (BEH) 等. 命令的传送是通过写时序实现的, 而主机读取 DS18B20 传送的数据是通过读时序实现的. 本测温子系统主要使用的命令为匹配 ROM 命令 温度变换命令及读暂存器命令. 其软件流程图见图 3, 主要子程序代码如下:

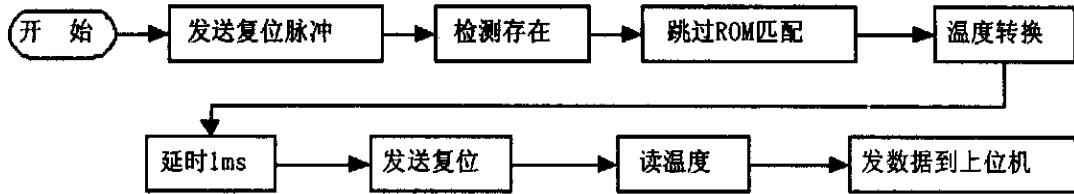


图3 温度采集流程图

有关精确延时问题, 为保证 DS18B20 严格的 I/O 时序要求, 需要作较精确的延时. 在 DS18B20 的操作中, 延时分两种: 短时间延时和较长时间延时. 短时间延时是指 $10\mu\text{s}$ 以下的延时, 在汇编语言下采用若干 NOP 指令即可. 因 C51 提供了若干内部函数 `_nop_()` 函数为其中之一, 其编译结果就是在对应位置嵌入一个 `nop` 汇编指令, 因此, 短时间延时可利用 `_nop_()` 函数实现. 较长延时指 $10\mu\text{s}$ 以上的延时. 在 DS18B20 操作中, 用到较长时间延时有 $15\mu\text{s}$, $90\mu\text{s}$, $270\mu\text{s}$, $540\mu\text{s}$ 等. 因这些延时均为 $15\mu\text{s}$ 的整数倍, 因此可以编写一个 `Delay15(n)` 函数, 用该函数进行大约 $15\mu\text{s} * n$ 的延时, 有了比较精确的延时保证, 就可以对 DS18B20 进行底层基本操作了, DS18B20 的底层基本操作有四个. 源码如下:

延时子程序:

```

void Delay15(n) // delay 15us
uchar n;
{
    do{
        _nop_(); _nop_();
        _nop_(); _nop_();
        _nop_(); _nop_();
        _nop_(); _nop_();
        _nop_(); _nop_();
        _nop_(); _nop_();
        _nop_(); _nop_();
    }while(n);
}
  
```

初始化子程序:

```

void RST18B20(void) // 初始化
{
    DS18B20 = 0;
    Delay15(32); // delay 510s
    DS18B20 = 1;
    Delay15(4); // delay 60s
    PSW_5 = 0;
    if (!DS18B20)
    {
        PSW_5 = 1;
        Delay15(28); // delay 420s
    }
}
  
```

初始化是 DS18B20 的底层基本操作之一, 通过单线总线进行的所有操作都从一个初始化序列开始. 初始化序列包括一个由 CPU 发出的复位脉冲及其后由 DS18B20 发出的存在脉冲. 存在脉冲让 CPU 知道 DS18B20 在总线上且已做好操作准备.

读数据子程序:

```

uchar RD18B20(void) // 数据读
{
    uchar q;
    ACC = 0;
    for(q = 8; q > 0; q - -)
    {
  
```

写数据子程序:

```

void WR18B20(d) // 写数据
{
    uchar d;
    ACC = d;
    for(p = 8; p > 0; p - -)
    {
  
```

```

ACC = ACC >> 1;
DS18B20 = 1;
_nop_();
DS18B20 = 0;
_nop_();
DS18B20 = 1;
Delay15(1);
BIT7 = DS18B20;
Delay15(4);
}
return (ACC);
}

```

写数据是 DS18B20 的底层基本操作之一,所有的指令数据发送均由该操作完成。DS18B20 的写操作都是逐位进行的,因此,采用 C51 中的位右移操作来实现。读数据也是 DS18B20 的底层基本操作之一,温度值和其它状态信息的传回均由该操作完成。起初打算采用与逐位写相同的方式编制逐位读的函数,将数据读入 CY 寄存器后再利用位右移操作将数据逐位送入 ACC;但实际写出的代码却不能正常工作。经分析 C51 所生成的汇编总是先清 CY 寄存器,再进行右移,这样数据在被送入 ACC 前就已经被清掉了。为了实现数据的逐位读,利用 ACC 的位寻址功能,在 C51 中将 ACC 的最高位定义为 BIT7,然后利用它来实现逐位数据读功能。

总体实现:有了上述几个基本子函数,就可以对 DS18B20 进行操作了。为了操作简便,可编写两个操作函数,源码如下:

<pre> . 启动温度转换 void ConvertT(void) // 启动温度转换 { RST18B20(); WR18B20(0xcc); WR18B20(0x44); } </pre>	<pre> . 读取温度值 int ReadT(void) // 读取温度值 { RST18B20(); WR18B20(0xbe); DPL = RD18B20(); DPH = RD18B20(); return (DPTR); } </pre>
---	---

以上程序采用 Keil uVision 编译通过,在采用 12MHZ 晶振的 AT89C52 单片机上试运行,并在中央储备粮库赤壁直属库中投入使用,取得了令人满意的效果。

4 结束语

由于 DS18B20 采用数字单总线技术,使得系统电路简单,易于扩展,加上总线数字化,使得系统的抗干扰性能好,可靠性高,测温范围比较广(-55~125),因而被广泛地应用于各个领域,与传统的采集系统相比还具有较高的性能价格比。通过用 C 语言来完成 DS18B20 的底层基本操作,我们发现,许多时候并不是非要用汇编不可,通过对 C51 编译的汇编代码分析,完全可以编制出准确而高效的 C 程序代码,其效率比汇编差不了多少。

参考文献:

- [1] 马忠梅. 单片机的 C 语言应用程序设计[M]. 北京:北京航空航天大学出版社,1997.
- [2] DALLAS 公司. DS18B20 数据手册[Z].
- [3] 周云波. 由 DS18B20 单线数字温度计构成的单线多点温度测量系统[J]. 电子技术应用,1996,(2):15~20.