

多路风速测量方法

敖振浪 吕玉嫦 陈武框

(广东省大气探测技术中心, 广州 510080)

摘要 自动站测量风速通常只有 1 路采集处理电路, 为了实现多路风速同时测量, 针对多路风速测量的特点, 设计了基于中断控制器的硬件电路, 利用中断方式进行软件计数的方法实现了 8 路风速同时测量。分析了系统的单片机运行速度和中断程序代码的相互关系, 计算中断例程可容许的最长执行时间。这种方法不仅避免了各路中断计数丢失, 而且起到了很好抗干扰作用。在本设计基础上, 适当增加硬件和软件优化, 可以扩展到同时 16 路风速测量的能力。给出实际的硬件原理图和部分源代码。该测量方法成功地应用于广东省大气探测技术中心开发的多路风速观测系统, 并在全国气象行业推广应用, 实验数据结果证明, 该测量方法实用可行, 应用效果良好。

关键词 中断控制器 风速测量 抗干扰

引言

风速测量的方法通常有两种, 一种是频率计数, 另一种是电压测量。目前普遍使用的是频率计数方法^[1], 而且一般采用单路或者多路硬件计数器。实践证明, 硬件计数器方法很容易受到电磁干扰^[2], 可能出现“飞车”现象, 产生错误计数, 而且计数器每次置位还可能漏掉一些脉冲计数, 造成测量误差。对于多路测量来说, 由多个独立的单路测量组合而成存在许多不足, 其中最主要的是系统一致性不好的问题。为了解决这些问题, 必须在同一个采集处理电路上同时接收各路风速信号。本文针对这种需求, 设计出利用中断控制器(8259A)构成的硬件电路, 利用中断方式进行软件计数的方法实现了 8~16 路风速同时测量。这种方法能够保持数据采集同步, 保持系统误差一致性, 而且系统具有性能稳定, 检测精度高, 抗干扰能力强等特点。

1 风速与脉冲信号

风速传感器信号大多数采取脉冲输出, 风速脉冲使用霍尔元件产生, 通常其频率在 0~1000 Hz 之间变化, 风速传感器输出脉冲与风速对应关系为: 风

速 $V = a + bf$, 其中: a 为常数, 即为起动风速; b 为系数, f 为每秒的脉冲数。对于特定型号传感器来说, 比如长春气象仪器研究所生产的 EC9-1 型风传感器, $b = 0.1$, $a = 0.5$ 。自然界风速是随机变化的, 所以 f 也是随机变化的。设备安装环境条件一般比较恶劣, 容易受到雷电产生的电磁脉冲以及其它毛刺信号的严重干扰, 采用硬件计数器的方案较易受到干扰使计数出错^[3]。而采用硬件中断软件计数法可以实现对各路输入脉冲宽度的甄别, 以尽量避免干扰信号, 甄别宽度的大小由中断程序延时决定, 这样能够最大限度地减少计数差错。

2 硬件电路构成

图 1 是 8 路风速脉冲通过中断获取的实际电路原理图。对于普遍使用的脉冲量风速传感器, 其输出脉冲频率范围一般是 0~1000 Hz。8 路风速脉冲信号从 P1 输入, 分别送到 OP0~OP7 光电耦合器, 经过隔离送到 U2 和 U3 非门电路进行整形, 再输入 8259A 中断控制器^[4]的中断输入脚 IR0~IR7。光电耦合器的作用是将传感器与采集电路物理隔离开来, 防止电磁感应(雷击)的直接冲击, 同时可以过滤毛刺干扰。PW1 是隔离式 DC/DC 电源转

广东省科技计划项目(2005B60401024)资助

作者简介: 敖振浪, 男, 1962 年生, 学士, 高级工程师, 长期从事大气探测方法研究、技术开发、技术支持、维修保障、建设规划和装备管理等, Email: zlao@grmc.gov.cn

收稿日期: 2007 年 10 月 30 日; 定稿日期: 2008 年 3 月 3 日

换器,独立供给风速传感器。8259A 中断控制器的控制线 CS、WR、RD、A0 和数据总线 D0~D7 等与单片机 MCU 对应相连,INT 经反相接单片机的外部中断 INT0。如果 IR_x 引脚加上有效电平,会使中断请求寄存器相应的位置位,优先级较高的位就

会通过 INT 向 MCU 发出中断请求信号。当 MCU 发出下一个 INTA 信号时,中断控制器 8259A 将中断号送到数据线上,被 MCU 读取,触发相应的中断事件。

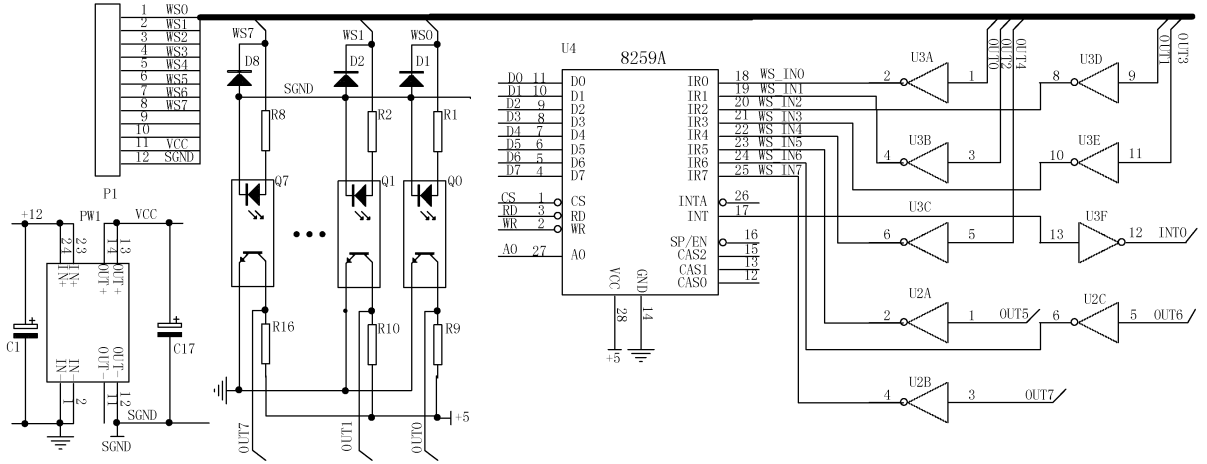


图1 多路风速测量原理图

3 软件编程方法

利用中断请求技术是提高系统实时性最有效方法^[5],主要模块是第1部分中断处理程序,专门负责对8路风速脉冲进行计数,每来一个脉冲相应通道计数加1;第2部分是利用单片机定时器中断例程对各路脉冲单位时间(1 s)内脉冲数的处理;第3部分是生成气象数据3 s瞬时风速、1 min平均风速和10 min平均风速。要考虑单片机中断优先级问题,从实际出发,本设计优先级设置顺序最高是外部中断 INT0,其次是定时器 T0,其他是低级别。采用单片机高级语言 Keil C51 编程,简明易读,限于篇幅,这里只给出部分源代码。

```
#include "define.c"
void main(void)
{Init8031Status(); Init_8259();
while(1){ ProcessDataTask(); Watchdog();
}
```

由于8路风速脉冲随机输入,都能够触发外部中断,因此要求中断服务程序代码非常短,执行时间足够快,完成任务后立刻退出才不会影响下一个其它中断。这里外部中断服务程序 Ext0() 代码执行时间长度 L_i 就等于对干扰信号甄别宽度,也就是

说周期小于宽度 L_i 的干扰脉冲信号被剔除掉,从而起到抗干扰作用。

```
void Ext( ) interrupt 0
{data unsigned char addressL;
BYTE[ADD_8259_A0_0]=0x0c;
addressL= BYTE[ADD_8259_A0_0];
WindSpeedCount[addressL&0x07]++;
BYTE[ADD_8259_A0_0]=0x20;
}
```

关键的技术是,为了在响应中断处理期间不丢失下一个中断,可容许中断例程最长处理时间是这样计算的:风速脉冲频率范围是0~1000 Hz,假设取最大1000 Hz,那么周期 $T=1/1000=1(\text{ms})$,也就是说,定时器 T0 和外部中断 INT0(有8路脉冲中断)总的执行处理时间不得大于1 ms。假设定时中断服务程序 Timer() 的代码执行时间长度为 L_d ,那么必须满足 $L_d+8L_i \leq 1 \text{ ms}$ 。实际运行表明,如果单片机使用35.2512 MHz晶振,外部中断服务程序代码执行时间长度大约是 $L_i=47 \mu\text{s}$,定时中断服务程序代码执行时间长度大约是 $L_d=178 \mu\text{s}$,那么 $L_d+8L_i=178+8 \times 47=0.554(\text{ms})$,远小于可容许中断例程最长处理时间1 ms的限制,完全不会影响下一个中断的处理。利用定时器0中断作为

秒时钟处理单元,设置每10 ms中断一次,中断100次(1 s)处理一次风速的计数值,转存到缓冲区等待数据处理例程处理。这里必须注意的是,因为定时器T0例程执行计数值WindSpeedCoun处理过程中可能会被外部中断INT0而中断,所以读走计数值WindSpeedCoun后,不能只置初值0,而要考虑处理过程中可能产生的计数增量,如下代码所示。

```
void Timer( ) interrupt 1
{ unsigned char loop;
  if(++T0Count >= 100)
    { for (loop = 0; loop < WIND_LAYS; loop++)
      { WindSpeedCount_Tmp [loop] = WindSpeedCount
        [loop];
        WindSpeedCount [loop] = WindSpeedCount
        [loop] - WindSpeedCount_Tmp[loop]; }
      T0Count = 0;
    }
}
```

在主程序的循环里,数据处理例程负责计算风速的统计值,通过系统的实时时钟控制定时,读取中断服务例程放在缓冲区里的数据进行风速瞬时、1 min和10 min平均值的统计。

```
void ProcessDataTask(void)
{ int i;
  if(Read_Min_Sec_Clock())//1s tast
    { Read_WindSpeed();
      for(i = 0; i < WIND_LAYS; i++) { Calculate_
        3sWind(i); Calculate_1MinWind(i); }
      if(++Wind1minCount > 61) Wind1minCount =
        60;
      if (CLK_BUF. MinutesBCD! = OLD_CLK.
        MinutesBCD) //per 1 minute
        { OLD_CLK. MinutesBCD = CLK_BUF. MinutesB-
          CD;
          for(i = 0; i < WIND_LAYS; i++) { Calculate_
            2MinWind(i); Calculate_10MinWind(i); }
          Wind1minCount = 1;
          if(((BCDToInt (CLK_BUF. MinutesBCD))%
            Howmany_Minute) == 0)//per 10 minute
            { Watchdog(); Read_NewClock();//save 10
              minutes data
              memcpy((char *) &.Data10Min_Buf, (char
                *) &.Data_Buf, OneTextLength);
              if(CLK_BUF. MinutesBCD == 0) { Save-
                Text(); } //save 1 hour data
```

```
    }
  }
}
void Calculate_3sWind(int windport)
{ float Temp;
  Temp = (WindTmp[windport]. W3sD[0] + WindTmp
    [windport]. W3sD[1] + Ws_Input[windport]);
  Data_Buf. WindBuf [windport]. Ws3sMax =
    Temp/3.0 + 0.5;
  WindTmp [windport]. W3sD [0] = WindTmp
    [windport]. W3sD[1];
  WindTmp [windport]. W3sD [1] = Ws_Input
    [windport];
}
void Calculate_1MinWind(int windport)
{ float Temp;
  Temp = WindTmp [windport]. Ws_1min [10] +
    (float)(Ws_Input[windport] - WindTmp[windport].
      Ws_1min[10])/Wind1minCount;
  WindTmp[windport]. Ws_1min[10] = Temp;
}
void Calculate_10MinWind(int windport)
{ float Temp, Temp1; char loop;
  Temp1 = WindTmp[windport]. Ws_1min[0];
  for(loop = 1; loop <= 10; loop++)
    { Temp = Temp1 + (WindTmp[windport]. Ws_1min
      [loop] - Temp1)/(loop + 1); Temp1 = Temp; }
  Data_Buf. WindBuf[windport]. Ws10min = Temp1;
  for(loop = 0; loop < 10; loop++)
    WindTmp [windport]. Ws_1min [loop] =
    WindTmp[windport]. Ws_1min[loop + 1];
}
```

4 使用情况与结论

对自然界的垂直分布梯度风速的测量必须采用具有同一系统误差的多路(多通道)测量仪器完成,测量的数据才有可比性。针对多路风速测量的特点,采用中断控制器的硬件中断软件计数的设计方法不仅可靠实用、成本低,而且具有抗干扰作用,还可以用于各种脉冲量数据采集项目中。但设计中要注意中断例程的合理优化,保证不影响后续中断的响应。通过增加一片8259A和优化软件代码,可以进一步扩展到16路风速采集的能力。本设计给出的电路和参数,成功地应用于广东省大气探测技术

中心开发的梯度风速观测系统中,经过计量部门使用风洞检测,检定结果准确、合格,在全国气象行业推广应用,实验数据结果证明,设计方法完全可行,应用效果良好。

参考文献

[1] 刘志刚. 测量风速的一种新方法[J]. 气象科技, 1983, (4): 78 - 80.

[2] 王接枝. 浅析一个实验电路的错误[J]. 上饶师范学院学报, 2002, (6): 25 - 26.

[3] 尹延凯, 乐秀番. 发电机转速数字测量技术[J]. 河海大学学报(自然科学版), 1999, 27(3): 69.

[4] 郭福田, 李从信, 刘贤梅. 微机打印口中断方式编程技术及应用[J]. 计算机应用研究, 2000, 17(1): 111 - 113.

[5] 王军安. 浅析嵌入式系统的软件优化设计[J]. 计算机工程与应用, 2004, (3): 102 - 103.

Multi-Channel Method for Measuring Wind Speed

Ao Zhenlang Lu Yuchang Chen Wukuang

(Guangdong Atmospheric Observation Technology Center, Guangzhou 510080)

Abstract: The wind speed measurement at automatic weather stations usually adopts only the single-channel acquiring and processing circuit. In order to achieve the multi-channel simultaneous wind speed measurement, the multi-channel measurement of wind speed is studied, and the characteristics of the hardware circuit design based on the interrupt controller with the software for interrupt counting are described. The 8-channel simultaneous wind speed measurement can be realized with the method. The relationship between system MCU (Microprocessor Control Unit) operating speed and interruption codes is analyzed, and the longest executive time allowable for an interrupt routine is calculated. This method not only can be used to avoid the interruption count lose, and but also play a very good role of anti-interference. On the basis of the appropriate additional hardware and software optimization, it can be extended to 16-channel simultaneous wind speed measurement. The hardware schematic chart and some source codes are given. This measurement method is successfully put to use in the Multi-Channel Wind Observing System developed by the Guangdong Atmospheric Detection Technology Center, as well as in other institutions across China, and the trial application shows that it is feasible and practical.

Key words: interrupt controller, wind speed measurement, anti-interference