

基于 MSP430F449 单片机的超温报警系统

王巍 蒋大明 北京交通大学电子信息工程学院 100044

摘要: 本文讲述了 MSP430F449 系列单片机的性能特点, 并介绍了温度传感器 AD590 进行温度测量的工作原理。基于 MSP430F449 单片机给出了超温报警系统的硬件设计方案和软件主程序流程图, 并进一步讨论了有关系统优化的问题。

关键词: 单片机 MSP430; 温度传感器 AD590; 报警系统

引言

目前, 在许多情况下需要对环境的温度进行限定, 其中包括人的生活工作环境、仪器设备的工作环境以及动植物的生长环境等。随着人民生活的改善和工业的发展, 安全问题变的就更加的。环境温度一旦超过限定值, 势必会对所处环境的人或设备造成影响, 更甚者将会给个人和社会造成巨大的损失和社会影响。因此, 在某些特定环境下安装温度报警器进行监控也成为一种必然的需要。本系统充分利用了温度传感器测量温度快速、使用简便的特点, 同时结合了 MSP430 单片机强大的功能和超低的功耗, 来对温度进行实时监控并做到超温报警。

1. 系统设计方案

1.1 硬件设计

系统硬件框图如图1所示, 该温度报警系统包括的功能模块主要有: 传感器 AD590, 信号调理电路, MSP430F449 单片机, 用于向单片机内设置温度限值的键盘输入电路, 降温电路, 报警电路, 和用于显示的 uPD16682A 液晶显示器。

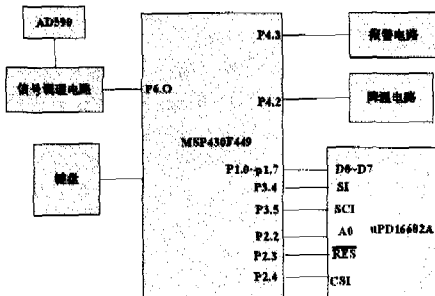


图1 硬件结构设计框图

1.1.1 AD590性能特点及工作原理

AD590是美国 Analog Devices 公司生产的二端式集成温度-电流传感器, 具有体积小、重量轻、性能稳定等一系列优点。测量温度范围为 -50°C ~ $+150^{\circ}\text{C}$, 完全适用于对环境进行测量, 由于 AD590 是温度-电流传感器, 适宜于远距离图1、硬件结构设计框图离测量和传输。AD590 的主要性能有: 线性电流输出是 $1\mu\text{A}/^{\circ}\text{K}$; 线性度好, 满刻度范围为 $\pm 0.3\%$; 电源电压范围 $4\sim 30\text{V}$, 当电流电压在 $5\sim 10\text{V}$ 之间, 电压稳定度为 1% 时, 所产生的误差只有 $\pm 0.1\%$ 。

AD590的基本工作原理是: PN 结伏安特性可用下式表示: $I = I_s [\exp(qU/kT) - 1]$

式中: I —pn 结正向电流; U —pn 结正向电压; I_s —pn 结反向饱和电流; q —电子电荷量; T —绝对温度; K —波尔兹曼常数。当 $\exp(qU/pT) \gg 1$ 时, 上式可表示为

$$I = I_s \exp(qU/pT) \text{ 则 } U = (kT/q) \ln(I/I_s)$$

可以看出通过 pn 结上的正向电流恒定的时候, pn 结的正向电压 U 与温度的线性关系就只受饱和电流 I_s 的影响。 I_s 是温度的缓变函数, 只要选择合适的掺杂浓度, 就可以认为在不太宽的温度范围内 I_s 近似为常数。因此, 就有了正向压

降与温度成线性关系。

1.1.2 MSP430F449单片机

MSP430 系列单片机是由美国 TI 公司生产的一种超低功耗的单片机。该系列单片机为 16 位 RISC 结构, 具有丰富的寻址方式 (7 种源操作数寻址、4 种目的操作数寻址)、简洁的 27 条内核指令以及大量的模拟指令; 大量的寄存器以及片内数据存储单元都可参加多种运算; 有高效的查表处理方法; 有较高的处理速度, 并且系统工作稳定。考虑到环境因素的影响在某些情况下可能不具备对系统提供有线电源的条件, 而采用 MSP430 系列单片机可大幅度的降低系统功耗, 从而为系统采用电池供电提供了可能。同时又综合考虑了存储空间是否满足程序及数据的需要、是否具有 FLASH 存储器以及是否具有 AD 转换功能等因素, 从 MSP430 系列的众多产品中选择了 MSP430F449 单片机。

MSP430F449 单片机主要有以下特点: 低工作电压 $1.8\text{V}\sim 3.6\text{V}$; 超低功耗 活动模式下工作电流为 $280\mu\text{A}$, 待机模式下是 $1.1\mu\text{A}$, RAM 关闭模式下工作电流为 $0.1\mu\text{A}$, 5 种节电方式; 从待机到唤醒的响应时间不超过 $6\mu\text{s}$; 16 位精简指令结构, 150ns 指令周期; 12 位 A/D 转换器带有内部参考源、采样保持、自动扫描特性; 带有捕获/比较寄存器的 16 位定时器; 定时器 A 与定时器 B; 串行通信可软件选择 UART/SPI 模式; 片内比较器配合其他器件可以后成单斜边 A/D 转换器; 可编程电压监测器; 可在线串行编程, 不需要外部编程电压; 可编程的保险熔丝保护设计者代码; FLASH 存储器多达 60KB, RAM 多达 2KB。

1.1.3 信号调理电路

AD590 的输出量是电流, 线性电流输出是 $1\mu\text{A}/^{\circ}\text{K}$ 。为把电流转换成电压, 在信号调理电路中通常将 AD590 的输出电流通过一个 $1\text{K}\Omega$ 电阻, 从而获得 $1\text{mV}/^{\circ}\text{K}$ 的输出电压。AD590 测量的是绝对温度, 绝对温度和摄氏度之间的关系为 $K = ^{\circ}\text{C} + 273.2$ 因此必须将 AD590 的输出偏置为 273.2mV , 才能将测量的绝对温度转换成摄氏温度。并再连一个增益可调数据放大器, 获得 MSP430F449 单片机内部 AD 转换器可以采用的输出电压。

1.1.4 液晶显示 uPD16682A

uPD16682A 是 NEC 公司 2001 年初推出的液晶显示驱动器, 该产品内置大容量显示 RAM 内存, 并能够提供 132×65 点阵的全点显示, 特别适用于 16×16 或 12×12 点阵的中、日文字符显示。该产品采用 $+3\text{V}$ 单电源供电, 内置升压电路并具有 3 倍压和 4 倍压两种工作模式, 支持 8 位串行或并行数据的输入, 内置时钟发生电路和程序可编程控制的偏压电路。

MSP430F449 单片机作为系统的核心部分, 主要用于对数据的处理和根据数据的处理结果做出相应的动作判断。报警电路由 0.5W 的音频功率放大器 LM386、扬声器和若干电容电阻组成, 由 MSP430F449 单片机的 P4.3 口驱动。降温电路主要是通过 P4.2 口控制继电器, 通过继电器控制风扇工作。

1.2 软件设计

主程序的流程图如图 2 所示。通过与 MSP430F449 连接的键盘可以自主设置超温报警系统中降温系统的启动温度和报警温度, 并能在液晶显示器上显示整个设置过程。键盘由 4 个按键组成, 分别是 E, S, O, Δ 。uPD16682A 在每个串行时钟的上升沿读取控制信号线 A0 上的电平, 并且根据电平来决定是显示数据还是执行控制指令。在 S 没有按下时, P2.2 为低电平, 显示器实时显示采集的温度; 当按下 S 时, P2.2 口输出改为高电平, 使 A0 口也变为高电平, 从而停止温度显示, 进入温度限值设定界面。设定温度时, 按 O 键来选择是要设置降温系统的启动温度还是报警温度, 选择好后按 Δ 键来选择要设置的温度限值, 并按 E 确定并返回数据显示界面。但要保证 (降温系统的启动温度) $<$ (报警温度), 并且不能超出本系统能测量的温度的范围。在测量温度值中采用了连续测 10 次采样值, 去掉最大值和最小值, 然后求平均作为最后测量结果的方法。这个方法可以有效的减小 AD 采样过程中显著的随机误差对测量精度造成的影响。

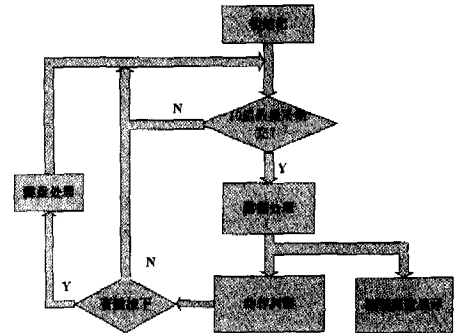


图2 主程序软件流程图

再用这个结果与设定的温度限值进行比较来进行动作判断可以避免单次测量的误差引起的降温系统的误启动或“虚报警”。在该系统中设定启动到停止风扇的延迟时间为 $0\sim 90\text{s}$ 。

2. 设计优化

该系统的测量温度范围可以通过对 MSP430F449 内部的 AD 转换器提供适当的外部参考电压作为基准电压, 从而达到调节 AD 的测量范围满足温度测量的需要, 这对于 MSP430F449 内部的 AD 转换模块是很容易的, 并且使用外部电压作为 AD 转换器的参考电压可以避免因 AD 转换模块内部参考电压不稳定造成的测量误差。并且在调节 AD 转换器参考电压的同时, 信号调理电路也要进行相应的调整, 但一定要保证调节的测量范围不能超出传感器 AD590 所能测量的范围。

系统中 AD590 的 $\Delta T-T$ 关系曲线如图 3 所示, 由图可知当 AD590 测量的温度小于 $+104^{\circ}\text{C}$ 时, 可近似将 $\Delta T-T$ 的关系看作是线性的, 测量误差与温度值基本满足

$$\Delta T = (2.8/104) * T$$

(下转第 12 页)

(void), void OSTimeTickHook(void).

(3) 用汇编语言编写 4 个与处理器相关的函数(OS_CPU.ASM); OSStartHighRdy(), 运行优先级最高的就绪任务; OSCtxsw(), 任务就绪的任务切换函数; OSIntCtxSw(), 中断就绪的任务切换函数; OSTickISR(), 中断服务函数。

使用 $\mu\text{C}/\text{OS-II}$ 系统应注意的问题:

$\mu\text{C}/\text{OS-II}$ 和 Linux 等分时操作系统不同, 不支持时间片轮转法。它是一个基于优先级的实时操作系统。每一个任务的优先级必须不同。进入就绪态的优先级最高的任务首先得到 CPU 的使用权, 只有等它交出 CPU 的使用权后, 其它任务才可以被执行。

$\mu\text{C}/\text{OS-II}$ 对其资源共享提供了保护的机制。 $\mu\text{C}/\text{OS-II}$ 是一个支持多任务的操作系统。我们可以把一个完整的程序划分成几个任务, 不同的任务执行不同的功能。对于共享资源(比如串口), $\mu\text{C}/\text{OS-II}$ 也提供了很好的解决办法。一般情况下使用的是信号量方法。我们创建一个信号量并对它进行初始化, 当一个任务需要使用一个共享资源时, 它必须先申请得到这个信号量。在这个过程中即使有优先级更高的任务进入了就绪态, 因为无法得到信号量, 也不能使用该资源。在 $\mu\text{C}/\text{OS-II}$ 中称为优先级反转。简单地说, 就是高优先级任务必须等待低优先级任务的完成。在上述情况下, 在两个任务之间发生优先级反转是无法避免的, 所以不在使用 $\mu\text{C}/\text{OS-II}$ 时, 必须对所开发的系统了解清楚才能选择对于某种共享资源是否使用信号量。

$\mu\text{C}/\text{OS-II}$ 内存管理不够完善。在分析许多 $\mu\text{C}/\text{OS-II}$ 的应用实例中发现, 任务栈空间和内存分区区的创建采用了定义全局数组的方法, 这样实现起来固然简单, 但不够灵活有效。数组的大小是固定的, 生成映像后不可能在使用过程中动态地改变。现在 $\mu\text{C}/\text{OS-II}$ 只支持固定大小的内存分区, 容易造成内存浪费。

4.2 驱动程序

驱动程序是连接硬件和 API 函数的部分, 执行硬件的修改、添加、删除操作只需要改变相应驱动程序, 不需要改动系统的 API 函数和实现功能的应用程序。由于有了 $\mu\text{C}/\text{OS-II}$ 中的信号量和邮箱等机制, 多任务之间的同步性就可以控制驱动程序来实现, 从而保证了硬件访问的唯一性。

这个系统中我们需要的是 JTAG 驱动、RS232 驱动、LCD 驱动、存储器驱动、键盘驱动, 这里简要介绍 LCD 驱动。我们选用 TOPRO-LCDD-MG 的 320*240 的液晶显示控制器, 将它以总线的方式连接到 S3C44BOX 上, 需要在系统内开辟一块内存作为它的后台缓冲区。程序如下:

```
#define LCD_H 240,
#define LCD_W 320,
static unsigned char LCDBuf[LCD_H][LCD_W],
```

其驱动需要初始化, 更新显示, 启动背光。接口函数如下:

```
void LCD_Cls(),
void LCD_Cls(), //文本模式下输出字符串
void LCD_Init(), //初始化LCD, 系统启动时调用
void LCD_Print(char*A, ...), //文本模式下输出字符串, 屏幕自动滚动, A 是所
```

要输出的字符串, 由于只用显示或者右, 因此只需要两个不同的输出就可以了, 如 "L", "R"

```
void LCD_Refresh(), //LCD 显示刷新
void LCDBkLight(INT8U IsOp), //打开或关闭 LCD 背光
void LCDDispOpen(INT8U IsOp), //打开或关闭 LCD 显示
```

4.3 应用程序

传统的程序设计基本是顺序、分支、循环结构, 效率比较低。在 S3C44BOX 芯片中移植的 $\mu\text{C}/\text{OS-II}$ 实时操作系统, 支持信号量、消息、邮箱等资源共享的方式, 有效避免的竞争资源。程序利用多任务的优势来设计可以大大提高资源的利用率, 创建几个任务来完成所需的功能。

TaskMain()接收 TaskAD()给出的信号量, 完成控制任务; TaskAD()用来完成六路模拟信号数据的采集; TaskFresh()完成各个控制 I 过程中数据的记录并向 TaskKeon()完成数据的运算并向 TaskDA()完成数字信号向模拟信号的转换。各个任务的优先级不一样, 但是都必须低于 $\mu\text{C}/\text{OS-II}$ 的内核任务。

整个应用程序分为系统初始化和主控程序, 系统初始化又包含了硬件初始化和操作系统初始化, 主控程序对整个进程进行控制。下面给出主控程序的部分代码结构:

```
#include "main.h"
#include "includes.h"
..... //需要的各种头文件
void main(void)
{
.....
HardwareInit(), //硬件初始化
OSInit(), //μC/OS-II 初始化
ActiveEvenQ=OSQCreate(&ActiveEvenMsg[0],10), //消息初始化放在此处
DataFlashSem=OSSemCreate(1), //信号量初始化
OSTaskCreate(TaskStart, (void *)0,
&TaskStartStk[STK_STK_SIZE-1],0),
OSTart(), //多任务开始
.....
}
```

5、结语

S3C44BOX 是一款数控中常用的嵌入式芯片, 其稳定性得到公认, $\mu\text{C}/\text{OS-II}$ 的嵌入式系统更好的进行多任务处理, 并保证了系统的实时性。模拟道指道指固车系统中的一个部分, 要实现固车全部系统的数字化还有相当的工作要做。

参考文献:

1. 何立民主编. 单片机应用技术选编. 北京: 北京航空航天大学出版社
2. 胡大可. MSP430 系列超低功耗 16 位单片机原理与应用. 北京: 北京航空航天大学出版社
3. 宋奇兵, 李毅. MSP430 与液晶显示器的串行接口方案. 《单片机与嵌入式的应用》2001.2 P46-52
4. 魏小龙. MSP430 系列单片机接口技术及系统设计实例
5. 作者简介: 王巍 (1981.1), 男, 硕士研究生, 研究方向是智能控制。

(上接第 10 页)

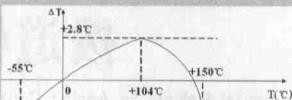


图 3 AD590 $\Delta T-T$ 关系曲线

并利用这个公式通过软件设计对测量的温度进行线性补偿, 可基本消除温度传感器带来的线性误差。

但当测量的范围超过 AD590 的 $\Delta T-T$ 关系曲线的线性部分的时候, 即测量的温度上限超出 +104°C 时, 这就要求在补偿传感器的测量误差的时候不能再采用线性补偿的方法, 而采用传统的方法对传感器进行标定。标定的方法是将若干个实际温度值以及与其相对应的信号调理后的电压值存入 MSP430P449 中, 然后采用最小二乘的方法拟合出与实际接近的特性曲线, 再将 AD 的采样值根据特性曲线还原为温度值, 这样就可以得到精度很高的测量结果了。

系统中的键盘功能部分和显示部分在实际当中可以用计算机代替, 可以通过 MSP430 单片机的串行同步步通信接口 USART 将系统与计算机网连接, 通过计算机设定温度限值并写入 FLASH 中, 并利用计算机的显示器实时显示测量的温度值。并且什么时候开始对温度进行测量和是否开启温度监控都是通过计算机的界面进行设置的, 使用者可以根据需要灵活的进行设置。并且在为了降低系统的功耗, 可以是单片机在处于等待状态的时候使其进入低功耗状态。

抗干扰的能力对于一个系统来说也是十分重要的性能指标。在本系统的电路板的设计中将 "接地线" 与 "数字地" 分开连接, 从而减小了接地网络对系统造成的干扰。

3、结束语

该系统具有结构简单, 容易实现, 测量范围大, 测量精度高并且价格低廉等优点。而且还可以给该系统添加一些功能进去, 比如: 自动控制功能, 电话报警功能等等。将该系统与计算机网络连接起来以后, 可以实现远程监控, 这样就方便了对系统的监测。

参考文献:

1. 何立民主编. 单片机应用技术选编. 北京: 北京航空航天大学出版社
2. 胡大可. MSP430 系列超低功耗 16 位单片机原理与应用. 北京: 北京航空航天大学出版社
3. 宋奇兵, 李毅. MSP430 与液晶显示器的串行接口方案. 《单片机与嵌入式的应用》2001.2 P46-52
4. 魏小龙. MSP430 系列单片机接口技术及系统设计实例
5. 作者简介: 王巍 (1981.1), 男, 硕士研究生, 研究方向是智能控制。