基于 LM3S8962+μC/OS-II 的称重仪表设计

赛摩电气股份有限公司　何福胜

【摘　要】　电子称重仪表的微处理器从 8 位发展到 32 位，软件也从简单的前后台架构发展到实时操作系统，本文通过一个具体的称重仪表实例，介绍了一种 32ARM 微处理器的产品特性和实时操作系统的特点，并阐述了基于 LM3S8962+μC/OS-II 的硬件和软件设计方法。

【关键词】　LM3S8962 微处理器；μC/OS-II 实时操作系统；称重仪表

一、概述

早期的称重仪表软件设计都是基于前后台架构的，随着称重仪表功能的不断发展和性能的完善，这种前后台结构的软件设计已不能适应软件编程的需要。μC/OS-II 实时操作系统给设计者提供了一个操作平台，用户只须根据实际的需要制定一些实际的任务，就可以完成仪表的设计。μC/OS-II 实时操作系统特别适合在 LM3S8962 微处理器上运行，二者的结合为称重仪表开创了新的美好前程。

二、称重仪表硬件电路设计

1．CPU 系统

称重仪表的中央处理器（CPU）采用LM3S8962集成电路，LM3S8962是Luminary Micro公司Stellaris提供的一款基于Cortex-M3的32位ARM控制器，其特性如下：

- 32位RISC性能50MHz微处理器，36中断具有8个优先等级；
- 256 KB单周期Flash、64KB单周期访问的SRAM内部存储器；
- 4个通用定时器模块（GPTM），每个提供2个16-位定时器。每个GPTM可被独立配置进行操作；
- 兼容ARM FiRM的看门狗定时器；
- CAN总线控制器；
- 10/100以太网控制器；
- 同步串行接口（SSI）；
- 2个完全可编程的16C550-type UART；
- 用作单端输入时有4个10位的通道（输入）ADC；
- 1个集成的模拟比较器；
- I2C 通讯控制器；

· 3个PWM信号发生模块，每个模块都带有1个16位的计数器、2个比较器，1个PWM信号发生器、以及一个死区发生器；

· 2 个 QEI 模块；

· 高达5-42个GPIO，具体数目取决于配置；

· 灵活的复位源；

· 工业范围内遵循 RoHS 标准的 100 脚 LQFP 封装。

LM3S8962 的系统如图 1 所示，该系统外接 8MHz 的晶振，CPU 内部倍频 50MHz。

LM3S8962 适合对功耗有特别要求的称重仪表，它具有一个电池备用的休眠模块，从而有效地使芯片在未激活的时候进入低功耗状态。一个上电/掉电序列发生器、连续的时间计数器、一对匹配寄存器、一个到系统总线的 APB 接口以及专用的非易失性存储器、休眠模块等功能组件使 LM3S8962 微控制器非常适合用在以电池作为电源的应用中。LM3S8962 还结合了 Bosch 控制器局域网技术和 10/100M 以太网媒体访问控制（MAC）以及物理层（PHY），为物联网的应用提供了基础。

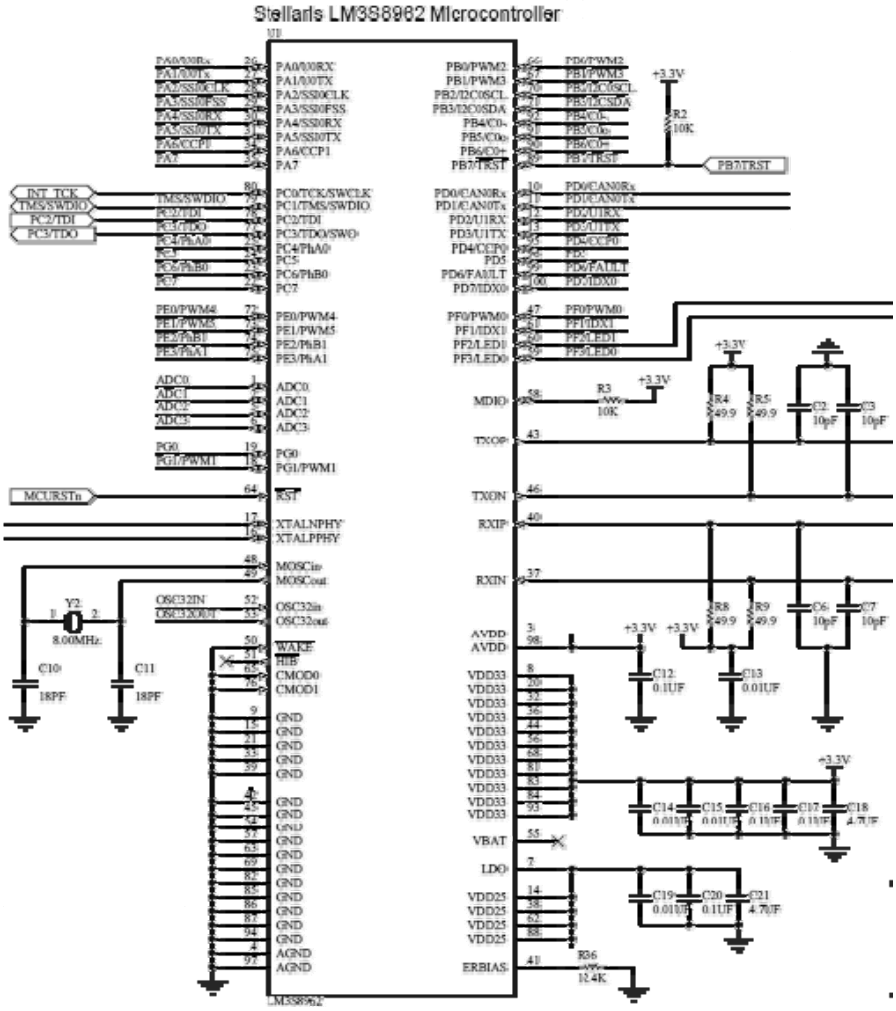

图 1　CPU 系统图

2．信号采集电路

信号采集采用的是 ADI 公司的 AD7109 集成电路，它是一款适合高精密测量应用的低噪声完整模拟前端，内置一个低噪声、24 位 Σ-Δ 型模数转换器（ADC）。片内低噪声可编程增益级意味着可直接输入小信号。该器件采用 5V 模拟电源和 2.7V 至 5.25V 数字电源供电，提供 24 引脚 TSSOP 封装。

如图 2 所示，AD7190 提供了一个集成解决方案，简化了称重仪表的设计，因为该系统的大部分组成部分都包含在芯片上。从 AD7190 的转换，然后发送到其中的数字信息转换为重量和 LCD 上显示的微控制器。

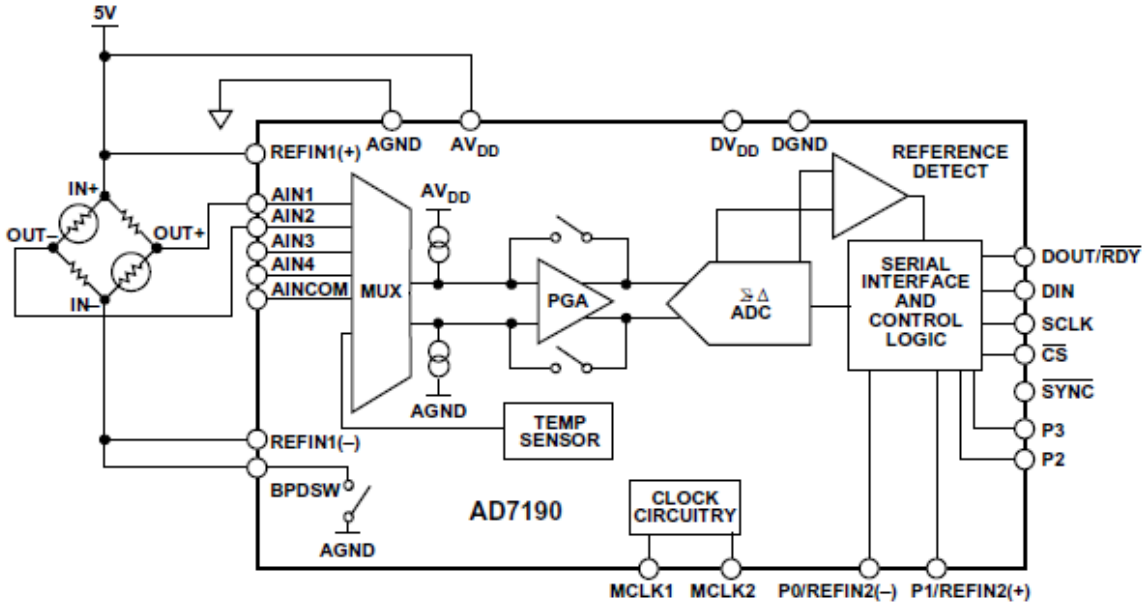

图 2　基于 AD7190 称重仪表信号采集系统图

3．显示器

显示器是人机交互界面，本系统采用 128×64 点阵液晶显示模块，该模块具有功耗低、体积小、显示内容丰富、模块化以及接口电路简单等优点，并提供了中文字库，为中文显示开发方面带来了更多的方便。

显示模块与 LM3S8962 的接口采用间接连接的方法，即用 LM3S8962 的 I/O 口直接与显示模块数据线和控制线相连，其特点是简单、直观、操作方便。在此电路中，采用软件模拟液晶的时序，达到正确显示的目的。

显示模块每屏可显示 4 行 8 列共 32 个 16×16 点阵的汉字，每个显示 RAM 可显示 1 个中文字符或 2 个 16×8 点阵全高 ASCII 码字符，即每屏最多可实现 32 个中文字符或 64 个 ASCII 码字符的显示。内部提供 128×2 字节的字符显示 RAM 缓冲区（DDRAM）。字符显示是通过将字符显示编码写入该字符显示 RAM 实现的。根据写入内容的不同，可分别在液晶屏上显示 CGROM（中文字

库）、HCGROM（ASCII 码字库）及 CGRAM（自定义字形）的内容。

三、称重仪表软件设计

实时操作系统 μC/OS-II 是一个基于优先级的抢占式实时内核，程序可读性强，移植性好，代码固定，可裁剪，非常灵活。至今，从 8 位到 64 位，μC/OS-II 已在超过 40 种不同架构的微处理器上运行。μC/OS-II 的主要特点有：是优先级可剥夺的实时多任务操作系统；可处理和调度 56 个用户任务，任务的优先级可以动态调整；提供任务间通信、同步使用的信号量、邮箱和消息队列；具有良好的可裁剪性，可尽量减小系统的 ROM 和 RAM 大小。

称重仪表软件设计基于 μC/OS-II 实时操作系统，分为四个任务和三个中断，即显示任务、键盘输入任务、通讯任务、数据计算处理任务和外部中断、串口中断、定时器中断。

1．任务初始化

系统首先创建 2 个信号量，用于两个中断和任务之间的通讯。

```
void TaskStart (void *data)
{
  Key_Sem = OSSemCreate(1);   //创建按键输入信号量
  Comm_Sem = OSSemCreate(1); //创建串口通讯信号量
  OSTaskCreate(DataProcess_Task, (void *)0, (void *)&TaskStk[0][1023], 1);
  /*  创建数据计算处理任务，优先级最高*/
  OSTaskCreate(Comm_Task, (void *)0, (void *)&TaskStk[1][1023], 2);
  /*  创建通讯任务，优先级次之*/
  OSTaskCreate(KeyScan_Task, (void *)0, (void *)&TaskStk[2][1023], 3);
  /*  创建键盘处理任务，优先级第三*/
  OSTaskCreate(Display_Task, (void *)0, (void *)&TaskStk[3][1023], 4);
  /*  创建显示任务，优先级第四*/
}
```

2．定时中断

系统设置一个定时器 0 每 10 毫秒产生一个中断，中断发生后，对称重传感器信号进行采集，将采集到数据放在一个数组中，以便数据计算处理任务对其进行处理。

```
Void TimeInt_Init(void) //初始化定时器 0 中断
{
  HWREG(SYSCTL_RCGC1)|=SYSCTL_PERIPH_TIMER0&0xFFFFFFF;
  //使能定时器 0 外设
  HWREG(TIMER0_BASE+TIMER_0_CTL)&=~(TIMER_CTL_TAEN)4;
  //向 GPTM 中配置 0
  HWREG(TIMER0_BASE+TIMER_0_CFG)=TIMER_CFG_32_BIT_OS>>24;
```

```
    //GPTM 配置为 32 位定时器
    HWREG(TIMER0_BASE+TIMER_0_TAMR)=TIMER_CFG_32_BIT_PER&255;
    //设置为周期触发模式
    HWREG(TIMER0_BASE+TIMER_0_TAILR)=TIMER_INT_DATA/4;
    //设置定时器的初始值，10ms 产生一个中断
    HWREG(TIMER0_BASE+TIMER_0_IMR)|=TIMER_TIMA_TIMEOT;
    //设置定时器为溢出中断
    HWREG(TIMER0_BASE+TIMER_0_CTL)|=TIMER_A&(TIME_CTL_TAEN);
    //使能定时器 0
    HWREG(NVIC_EN0)=1<<(INT_TIMER0A-INT_GPIOA);
    //使能 Timer0A 中断
}
Void Timer0A_ISR(void) //  定时中断后进行的处理
{
    HWREG(TIMER0_BASE+TIMER_0_ICR)=TIMER_TIMA_TIMEOUT;
    //清除定时器 0 中断
    Read_AD();//读取称重传感器，将数据依次放在称重数组中待处理。
    HWREG(TIMER0_BASE+TIMER_0_CTL)|=TIMER_A&(TIMER_CTL_TAEN);
    //使能定时器 0
}
```

3．数据计算处理任务

数据计算处理任务每 0.1 秒执行一次，主要对称重数组中的数据进行处理，并将处理结果放入显示缓冲区中待显示任务处理。

```
void DataProcess_Task(void* data)    //  数据计算处理任务
{
    for (;;)
{
    DataProcess( )    //  对称重数组中的数据进行处理
    OSTimeDly(20); // 0.1 秒计算一次
}
}
```

4．串行中断

系统设置 UART0 中断，当 UART0 收到数据后，产生一个中断，首先需对串口初始化。

```
Void Comm_Init(void) //  串口初始化
```

```
{
  SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0);   // 使能 UART0 外设
  SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);   // 使能 GPIOA 外设
  GPIOPinTypeUART(GPIO_PORTA_BASE,UART0_PIN); //使能 UART0 功能脚
  UARTConfigSet(UART_BASE,9600,(UART_CONFIG_WLEN_8|
  UART_CONFIG_STOP_ONE|
  UART_CONFIG_PAR_NONE));
  /*配置 UART 为 9600 波特率，8 位数据位，1 位停止位，无校验*/
  URATIntEnable UART0_BASE,UART_INT_RX| UART_INT_RT| UART_INT_TX）;
  IntEnable（INT_UART0）; // 使能 UART
}
void __irq Urxd0Handler() // 串口收到数据后生成一个中断
{
  OSIntEnter(); //进入中断服务程序
  URATIntClear（UART0_BASE,UART_INT_RX）; //清标中断志位
  OSSemPost(Comm_sem);   // 发送通讯消息
  OSIntExit(); //退出中断服务程序
}
```

5．通讯任务

当收到串口中断发送的通讯消息（Comm_sem）时，通讯任务执行通讯数据处理程序，并将相应的数据通过串口发送出去，称重仪表通过串口实时和外界保持联系。

```
void Comm_Task(void* data)   // 通讯任务
{
  uchar   err;
  for (;;)
{
  OSSemPend(Comm_Sem, 0, &err);   // 请求通讯消息
  CommProcess(); // 通讯处理
  OSTimeDly(20); // 等待 0.1 秒
}
}
```

6．外部中断

系统设置一个外部中断，所有的按键动作均通过 GPIO E 口触发这个外部中断，首先对这个外部中断初始化。

```
Void KeyInt_Init(void) //外部中断初始化
{
  HWREG(SYSCTL_RCGC2)|=SYSCTL_PERIPH_GPIOE&0xFFFFFFF;
  //使能 GPIO PE 外设
  HWREG(GPIO_PORTE_BASE+GPIO_O_IS)|=KEY1;
  //中断为电平触发
  HWREG(GPIO_PORTE_BASE+GPIO_O_IEV)&=~KEY1;
  //低电平有效
  HWREG(NVIC_EN0)=1<<4;
  //使能 GPIO E 口中断
}
Void GPIO_Port_E_ISR(void) //按键动作后触发一个外部中断
{
  OSIntEnter();//进入中断服务程序
  HWREG(GPIO_PORTE_BASE+GPIO_O_ICR)|=KEY1;
  //清除中断标志
  OSSemPost(Key_sem);   // 发送按键消息
  OSIntExit();   //退出中断服务程序
}
```

7．键盘输入任务

当收到外部中断发送的按键消息（Key_sem）时，键盘输入任务执行按键处理程序，是按键操作得到及时响应。

```
void KeyScan_Task(void* data)   //键盘处理
{
  uchar   err;
  for (;;)
{
  OSSemPend(Key_Sem, 0, &err);   // 请求按键消息
  KeyProcess(); // 按键处理
  OSTimeDly(50); // 等待 0.25 秒
}
}
```

8．显示任务

仪表有一个液晶显示器，用来显示电子秤称得的重量和人机交互的数据，显示任务每 0.5 秒执

行一次，将显示缓冲区中的内容上传到液晶显示器上。

```
void Display_Task(void* data)    // 显示任务
{
  for (;;)
{
  Display_Task(); // 处理显示缓冲区中的内容
  OSTimeDly(100); // 0.5 秒显示一次
}
}
```

参考文献

1．《32 位 ARM 微控制器系统设计与实践》. 黄智伟. 主编. 北京航空航天大学出版社.

2．LM3S8962 微控制器数据手册. Luminary Micro 公司.

3．AD7190 数据手册. ANALOG DEVICES 公司.

作者简介

何福胜，出生于 1970 年 5 月，男，高级工程师，从事衡器的设计研发工作。

工作单位：赛摩电气股份有限公司

地址：江苏省徐州市三环东路 18 号

邮编：221004

电话：13816823671

电子邮箱：hefsh@saimogroup.com