

基于 MCU 与 DSP 的 双机压电捷联惯导系统

■ 重庆大学 王家兵 陈林 陈国洪

摘要

以压电陀螺及压电加速度计作为惯性器件的惯性导航/制导系统,具有成本低、质量轻、抗冲击、可靠性高等优点,是惯性导航一直研究的解决方案。文章介绍基于 TI 公司 TMS320C5410 实现的压电捷联惯导系统的平台,论述系统的硬件与系统组成和设计,详细说明 DSP 系统中 HPI 接口、串口发送数据以及程序加载自举等的使用与实现方法。

关键词 压电捷联惯导系统 DSP 捷联姿态计算

引言

近年来,广大科研工作者研究了各种减小压电捷联惯导系统的误差方法,使压电惯性器件的精度得到了极大的提高^[1]。本文介绍了一种实用的基于 DSP 实现的压电捷联惯导系统方案。

1 系统的硬件设计

整个压电捷联惯导系统分为三个部分:压电惯性组合部分;由 ADS1251 与 ADuC834 组成的信号接口与模数转换单元;由 TMS320C5410 等构成的数据处理单元。系统框图如图 1 所示。

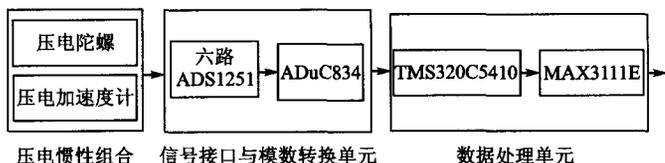


图 1

压电惯性组合采用专用压电陀螺及压电加速度计。由 TI 公司 24 位,20 kHz 的 A/D 转换器 ADS1251 完成六路压电陀螺及压电加速度计的信号精确采样,实际采样速率为 500 Hz。采用美国模拟器件公司的 8 位 51 MCU 微处理器 ADuC834 作六路采样的主控制器。ADuC834 集成了温度传感器、62 KB 的可编程序 EEPROM、定时器,以及 I²C 兼容的 SPI 和标准的串行 I/O 等。通过 SPI 方式读入六路采样转换后的信号,同时完成温度的采样,所有采样后得到的数字信号通过 HPI 接口写入到 TMS320C5410

的数据单元。采用 ADuC834 的口 0 与口 2 实现与 TMS320C5410 的 HPI 接口相连,接口电路如图 2 所示。

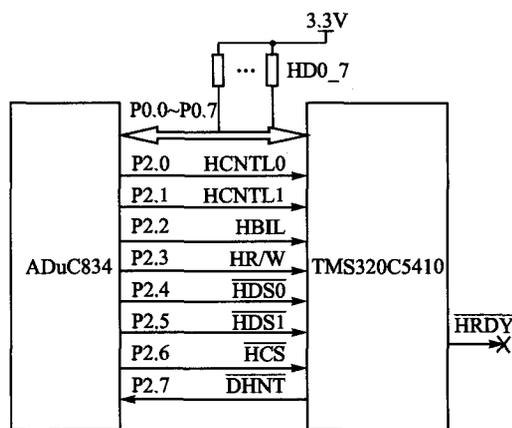


图 2

数据处理单元由 TMS320C5410、SST39VF200B 及 MAX3111E 组成。TMS320C5410 是 TI 公司 54 系列 DSP 处理器,外接 10 MHz 晶振,通过设置 PLL,工作频率在 100 MHz,处理能力可达到 100 MIPS。它采用微计算机工作方式(MP/MC 引脚接地),外接 SST39VF200B 作为外接存储器。系统启动时,由固化在 TMS320C5410 片内 ROM 的自举引导程序加载 SST39VF200B 中的应用程序。TMS320C5410 与 ST39VF200B 接口如图 3 所示。

TMS320C5410 的 McBSP0 与 MAX3111E 相连,完成串口数据的输出。设置 McBSP0 工作在 SPI 主动模式,与 MAX3111E 进行通信。电路接口如图 4 所示。

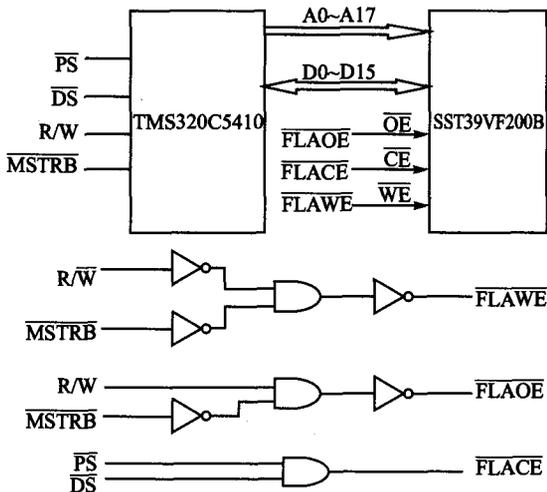


图 3

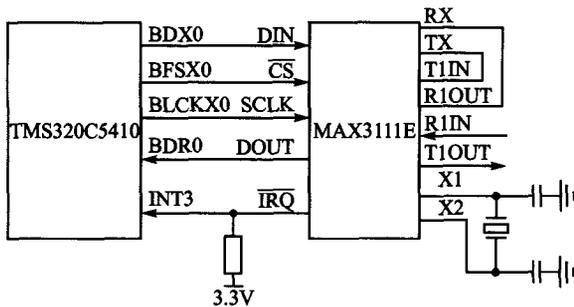


图 4

2 系统软件设计

2.1 HPI 接口

系统软件包括 ADuC834 的软件设计与 DSP 的软件设计。ADuC834 软件部分采用汇编语言编写,完成 HPI 的初始化、温度信号的采集、通过相应引脚的控制完成六路信号采集及接收、HPI 数据的发送等。采用了 $\overline{\text{HINT}}$ 引脚信号来完成双方数据的同步。DSP 通过向 HPIC 的 HINT 位写 1,使 $\overline{\text{HINT}}$ 引脚变为低电平,指示 ADuC834 发送新的数据帧。ADuC834 从引脚 P2.7 读到此低电平信号,写完一帧数据到设定的 DSP 数据区域,再写 HPIC 的 HINT 位,恢复 $\overline{\text{HINT}}$ 引脚为高电平。然后向 HPIC 中的 DSPINT 位写入 1,通知 DSP 进入 HPI 中断接收数据。DSP 接收完数据后,再向 HPIC 的 HINT 位写 1,指示新的数据传输过程。

2.2 主程序

DSP 部分的软件采用 C 语言设计,包括主程序、HPI 中断服务子程序及定时中断服务子程序。主程序完成系统的初始化,包括 DSP 工作模式的设置、堆栈的设置、初

始化 McBSP0、初始化 MAX3111E、设置定时中断等。框图如图 5 所示。

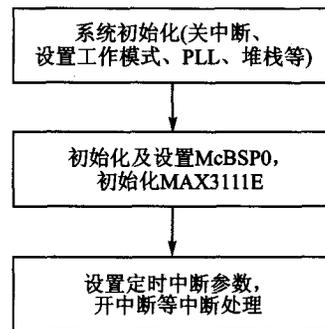


图 5

2.3 捷联姿态计算

HPI 中断部分完成数据接收的同时,完成数据的计算处理,ADuC834 的数据已存放在设定的几个数据单元,对读出数据采用四元数算法进行捷联姿态计算。算法部分功能模块如图 6 所示。

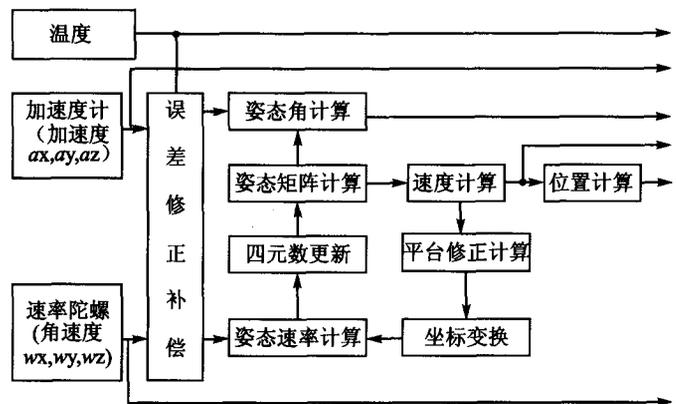


图 6

2.4 串行数据发送

通过配置 McBSP0 及 MAX3111E,实现在定时中断部分完成计算出的角度、位置、速度等数据的发送。设定了每 10 ms 发送一次数据,采用查询方式完成数据的发送。

在系统引导时完成 McBSP0 及 MAX3111E 的初始化,McBSP0 始初始化程序如下:

```
void init_mcbbsp () {
    int i, j;
    int a[14] = {0x1800, 0x0, 0x40, 0x41, 0x40, 0x41, 0x63, 0x2000,
                0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0xf0c};
    for (i=0; i<14; i++)
        { *SPSA0=i;
```

```

* SPSD0=a[i];
//对各控制寄存器写入,设置 McBSP0 工作模式
for (j=0;j<100;j++){ //延时
};
* SPSA0=0x01; // SPCR20
* SPSD0=*SPSD0&0xFFFE; //复位
* SPSA0=0x01; // SPCR20
* SPSD0=*SPSD0|0X40; //使能采样率发生器
* SPSA0=0x00; // SPCR10
* SPSD0=*SPSD0|0X81; //使能 dx 串行接口接收器
* SPSA0=0x01; // SPCR20
* SPSD0=*SPSD0|0X80; //帧同步发生器,串行接口发送器
* SPSA0=0x01; // SPCR20
* SPSD0=*SPSD0|0X01; //使能串口发送
};

```

通过查询 SPCR2 的 XRDY 位与 XEMPTY 位可知是否可向 McBSP 写数据,查询程序如下:

```

void mcbsp_ready (){
unsigned int flag=0x0;
while (flag!=0x02){
* SPSA0=0x01;
flag=*SPSD0&0x02;
//查询 SPCR2 的 XRDY 位,1 为准备好
};
flag=0x00;
while (flag?!=0xfffb){
* SPSA0=0x01;
flag=*SPSD0|0xfffb;
//查询 SPCR2 的 XEMPTY 位,0 为空
};
};

```

通过发送数据写配置寄存器完成 MAX3111E 的初始化,程序如下:

```

void init_max3111e (){
unsigned int flag1=0x0;
mcbsp_ready();
* DXR10=0xc801;
//写配置寄存器,允许发送缓冲区空中断,fosc=3.6864 MHz
//波特率为 115 200 b/s,8 位数据位,1 位停止位
mcbsp_ready();
};

```

在发送数据时,为保证 MAX311E 不丢失数据,需用中断方式或查询方式,在 MAX3111E 的缓冲区空时再发送数据,查询方式发送数据程序如下:

```

void send(unsigned char data){
unsigned char flag=0x0, datahi, datalow;
while (flag!=0x4801){ //第 14 位为 1,表示发送缓冲区为空
mcbsp_ready()

```

```

* DXR10=0x4000; //发送查询字
mcbsp_ready();
flag=*DRR10;
};
* DRX10=(data&0x00ff)|0x8000; //发送低 8 位
while (flag!=0x4801){
mcbsp_ready()
* DXR10=0x4000; //发送查询字
mcbsp_ready();
flag=*DRR10;
};
* DRX10=(data>>8)|0x8000; //发送高 8 位
};

```

3 程序的编译与自举

通过 JTAG 接口,由仿真器可方便的对平台进行调试,同时完成应用程序的写入。可在 CCS 集成开发环境中建立相应的工程,导入 .cmd 文件、vector.asm 文件、库文件、源程序等。在编译选项中加入 -v548,编译后生成相应的 .out 文件。整个程序小于 32 KB,使用 C54xx 通用 Flash 烧写工具 C54xx Flash Tool 2.01^[2],生成相应的 16 位 hex 文件及 Flash 烧写的 flashburn.out 文件。在 CCS 中导入 flashburn.out,设置 CPU 寄存器 DROM 位为 0,然后运行,即完成了对 SST39VF200B 中程序的烧写。

结 语

基于 TMS320C5410 实现的压电捷联惯导系统平台,电路体积小,系统稳定性高。经测试,整体性能满足误差校正、姿态角及速度、加速度的计算要求,并提供了捷联惯导系统实现各种算法的基础平台。

参 考 文 献

- 1 赵忠. 压电捷联惯导系统研究. 中国惯性技术学报, 2000(4)
- 2 张健军. C54xx Flash Tool 2.01. <http://skywolf.8866.org/FlashTool/FlashTool.htm>
- 3 赵忠, 马戎, 等. 一种超低成本捷联惯性制导系统的研制. 弹箭与制导学报, 2000(2)
- 4 刘勇, 李泽莉, 等. 一种基于压电陀螺的捷联惯性测量与制导系统. 压电与声光, 2003(6)
- 5 张雄伟, 陈亮, 等. DSP 芯片的原理与开发应用. 北京: 电子工业出版社, 2003
- 6 戴明祯, 周建江. TMS320C54X DSP 结构、原理及应用. 北京: 北京航空航天大学出版社, 2001
- 7 Texas Instruments. TMS320C54x DSP Reference Set. 2001

(收稿日期: 2005-07-11)