ANALOG DEVICES

AN-602
应用笔记

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com

ADXL202在计步器和个人导航设备中的应用

作者：Harvey Weinberg

简介

iMEMS®加速度计激发起了许多设计人员构建精确计步器的兴趣。个人导航系统是计步器的延伸，它将电子罗盘集成到计步器中，以便用户能够确定其相对于某个起点的位置。本应用笔记讨论设计人员在这些应用中遇到的问题，并且介绍一些用于实现个人导航系统的策略。

经典实现方法

加速度计用作惯性导航系统中的位置传感器已有多年的历史。惯性导航系统综合运用加速度计和陀螺仪并通过航位推算法来确定位置，位置相对于已知参考点（或起点）的偏差通过各轴的加速度对时间的积分来确定。其数学计算公式非常直观：

$$Position = Starting\ Position + \frac{A \times t^2}{2}$$

然而，对于低速运动，这种系统在任何合理长度时间内的精度都很差，因为小的直流误差不断积累，最终导致非常大的误差。通过下面的例子可以非常清楚地说明这一点。假设一个人以5 km/h (1.39 m/s)的速度步行5分钟；在他行走的416 m距离中，平均加速度为：

$$A_{avg} = \frac{2 \times Displacement}{t^2} = \frac{833}{300^2} =$$

$$0.00926\ m/s^2 = 0.944\ mg$$

ADXL202的温度系数约为2 mg/°C，因此即使在这5分钟只有0.5°C的温度偏差，也会带来1 mg的误差—大于目标信号本身！事实上，加速度计的倾斜度仅仅改变0.06°C就会大于1 mg。

为将误差降至最低，我们必须知道加速度计的方位，并且有某种方法可以将积分器频繁地“重置”到已知参考位置。许多系统使用GPS接收机或位置开关来定期提供此参考位置信息。如果能够相当频繁地提供此绝对位置信息(例如每10秒提供一次)，我们将能大大降低误差。

在10秒内，平均加速度为28.4mg。假设我们能将10秒内的所有直流误差限制在1mg，并且能够固定加速度计的方位，那么位置误差将只有大约0.5 m，远远小于仅使用GPS系统时的误差。因此，将航位推算法配合现有定位系统使用可能非常有利，而单独使用则不太精确。

下面以电梯为例说明航位推算法在何种情况下应用效果良好。电梯轨道每隔1米的位置上都放有磁性位置开关。但是，我们希望电梯的定位精度能够达到10 mm。传统解决方案是在与轨道耦合的轮子上使用一个光学编码器，作为精密位置传感器。由于机械传感器易于磨损，因此我们希望用加速度计取代编码器轮，以提高长期可靠性。

假设我们能使几秒内的直流误差稳定在1 mg，电梯以1 m/s的速度运行，那么位置误差为：

$$E_{pos} = \frac{A \times t^2}{2} = \frac{1\ mg \times 9.8\ m/s \times 1}{2} = 4.9\ mm$$

该误差远远小于目标误差。

计步器

当试图确定一个人步行了多远时，我们还有其它信息可供使用。人们在步行时，每走一步身体都会发生Z轴（垂直）运动。一种简单但不够精确的步行距离测量方法是利用此Z轴运动来确定走了多少步，然后将步数乘以平均跨步长度。

常用的计步算法采用某种形式的峰值检测。一般以10 Hz至20 Hz的频率进行采样，然后平均到2 Hz至3 Hz以消除噪声。跨步检测程序检测Z轴加速度的斜率有无变化。如果斜率发生变化，则表明已跨出一步。

速度计激发起了许多设计人员构建精确计步器的兴趣。个人导航系统是计步器的延伸，它将电子罗盘集成到计步器中，以便用户能够确定其相对于某个起点的位置。本应用笔记讨论设计人员在这些应用中遇到的问题，并且介绍一些用于实现个人导航系统的策略。

提高精度

如果使用固定不变的跨步长度，系统精度必然很低。(给定步速下的)跨步长度主要取决于腿长，不同人之间可能相差高达±40%。一些计步器要求用户设置其跨步长度，以消除大部分这种误差。然而，每个个体的跨步长度也会随着其步行速度的快慢而不同，变化幅度最高可达±50%(低速步行时跨步较短，高速步行时则长得多)。知道腿长并不能消除这种误差。但如果仔细观察应用情况，我们可以找到改善之道。

步行时，只有脚离地，膝盖才会弯曲。因此，当脚未离地时，我们可以将腿部视作一个定长的杠杆。图1显示了步行时臀部连带着上半身如何垂直运动。通过相似角几何原理可知：

$$\alpha = \theta$$

因此可以得到：

$$Stride \approx \frac{2 \times Bounce}{\alpha}$$

其中弹跳长度(Bounce)指臀部(或上半身)的垂直位移(Z轴)。

弹跳长度(Z轴位移)可以通过Z轴加速度的二次积分来计算。α是一个很小的角，难以测量，因为步行时所有轴上都存在大量冲击。经验证明，可以简单地认为α是一个常数，这对精度的影响不大。事实上，我们可以通过以下公式近似计算步行距离：

$$Distance \approx \sqrt[4]{A_{max} - A_{min}} \times n \times K$$

其中：

- A_{min}表示单次跨步中测得的Z轴最小加速度。
- A_{max}表示单次跨步中测得的Z轴最大加速度。
- n为行走的步数。
- K为单位转换(即用英尺或米来计量)所用的常数。

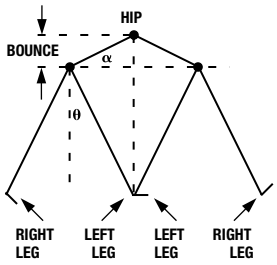

图1.步行时的臀部垂直运动

通过对不同腿长的各类对象进行测试表明，这种技术的测距精度在±8%以内。加速度计与身体的紧密耦合对于保持精度非常重要。采用能够"学习"用户跨步特征的自适应性算法可以显著提高精度。

本应用笔记的附录提供了一个适用于Parallax BASIC Stamp® (BS2)处理器的BASIC程序，它能执行步伐计数和距离计算功能，并能将行走的距离和步数显示在一个标准16×2 LCD显示器上。

增加方向检测功能

为了实现一个完整的个人导航系统，需要提供某种方向检测方法。通常由电子罗盘来处理此类任务。霍尼韦尔和飞利浦等制造商可提供适合个人导航应用的低成本电子罗盘传感器元件和模块。微控制器使用来自加速度计的距离信息和来自电子罗盘的方向信息，通过矢量加法来确定用户身在何处(相对于起始位置)。

加速度计和微控制器也可以用来实现一个罗盘倾斜校正算法(请参考电子罗盘制造商关于倾斜校正技术的应用笔记)，从而提高电子罗盘的精度。

结论

对于航位推算时间较短的系统，航位推算法可以用来提高其位置分辨率，但对于长时间位置测量应用，此方法不太有效。对应用情况进行仔细研究，往往能够发现异常简单的解决方案。在本例中，一个简单的数学公式加上一个简单的计步程序就胜过传统的航位推算技术。

附录

计步和距离计算源代码

与Parallax BASIC Stamp (BS2)一起使用。欲了解更多信息，请参见应用笔记“采用ADXL202/210和PARALLAX BASIC STAMP模块加速算法开发”(www.analog.com/library/applicationNotes/mems/StampXL202.pdf)。

```
DATAO  VAR    OUTH
RS     VAR    OUT1
RW     VAR    OUT2
E              CON 3
STEPS  VAR    WORD
DIST   VAR    WORD
T1     VAR    WORD
T2     VAR    WORD
TEMP2  VAR    WORD
ACCEL  VAR    WORD
Tn     VAR    WORD
Tn1    VAR    WORD
XLMIN  VAR    WORD
XLMAX  VAR    WORD
STRIDE VAR    WORD

DELTA          CON    190     'ACCELERATION DELTA BETWEEN SAMPLES. ADJUST
                              'THIS CONSTANT TO CHANGE NOISE IMMUNITY
FUDGEMULT      CON    7       'FUDGE FACTORS. STRIDE IS MULTIPLIED BY
FUDGEDIV       CON    48      '(FUDGEMULT/FUDGEDIV)THESE FUDGE FACTORS
                              'DETERMINE DISTANCE UNITS (FEET, m, etc.)

DIRH=%11111111                'INITIALIZE I/O FOR LCD
DIR2=1
DIR1=1
STEPS=0                       'ZERO THE STEPS AND DISTANCE COUNTERS
DIST=0
HIGH 5                        'TURN ON ACCELEROMETER
COUNT 7,500,TEMP2             'READ T2 PERIOD ONCE
T2=25000/(TEMP2/20)           'T2 IS THE PERIOD IN µs


MAIN
  GOSUB LCDSTART              'DISPLAY THE TOP LINE
                              'ONLY THE BOTTOM LINE GETS REFRESHED
  XLMIN=10000                 'INITIAL DUMMY VALUES
  XLMAX=10000
LOOP
  GOSUB SAMPLE                'TAKE AN ACCELERATION SAMPLE
  GOSUB FINDPEAK              'LOOK FOR A PEAK
  STRIDE=XLMAX-XLMIN          'NORMALIZE STRIDE ACCELERATION
  STRIDE=SQR STRIDE
  STRIDE=STRIDE*16
  STRIDE=SQR STRIDE
  STRIDE=STRIDE/3
  STRIDE=(STRIDE*FUDGEMULT)/FUDGEDIV
  IF STRIDE>0 THEN MAIN1      'IF MATH UNDERFLOWS, THEN
  STRIDE=1                    'ANY STRIDE IS > 0
```

```
MAIN1
  FREQOUT 4,35,2800         'BEEP WHEN A STEP IS DETECTED
  GOSUB INCSTEPS            'INCREMENT STEP AND DISTANCE COUNTERS
  GOSUB SHOWSTEPS           'UPDATE STEP AND DISTANCE DISPLAYS
GOTO LOOP


'********* SUBROUTINES ************
SEND                        'USED TO PULSE THE ENABLE PIN ON LCD DISPLAY
                            'NO REGISTERS USED OR MODIFIED
  PULSOUT E,30
  PAUSE 1
RETURN


SAMPLE                      'TAKES 2 SAMPLES OF T1X AND CONVERTS TO mg × 2
                            'REGISTERS MODIFIED: T1, TEMP2, ACCEL
                            'INPUTS: T2
                            'OUTPUTS: ACCEL
  T1=0
  PULSIN 7,1,TEMP2          'ACCUMULATE 4 PULSES
  T1=T1+TEMP2
  PULSIN 7,1,TEMP2
  T1=T1+TEMP2
  PULSIN 7,1,TEMP2
  T1=T1+TEMP2
  PULSIN 7,1,TEMP2
  T1=T1+TEMP2
  T1=T1×80
  TEMP2=T2/50
  ACCEL=T1/TEMP2            'ACCEL=ACCELERATION IN mg
RETURN


FINDPEAK                    'FINDS THE ACCELERATION PEAK
                            'INPUTS: ACCEL
                            'REGISTERS MODIFIED: Tn, Tn1
                            'OUTPUTS: XLMAX, XLMIN
  Tn=(XLMIN/2)+(XLMAX/2)    'THIS FORCES THIS ROUTINE TO FIND A NEW
  Tn1=Tn                    'MIN AND MAX EVERY TIME
  GOSUB SAMPLE              'READ ACCELERATION (Nth SAMPLE)
  Tn=ACCEL
  PAUSE 50
P1
  GOSUB SAMPLE
  PAUSE 50
  Tn1=ACCEL                 '(Nth + 1 SAMPLE)
  IF Tn1>Tn THEN P2         'IF ACCELERATION IS INCREASING THEN JUMP
  Tn=Tn1
  XLMIN=Tn
  GOTO P1
P2
  Tn=Tn1
  GOSUB SAMPLE
  PAUSE 50
  Tn1=ACCEL
  IF Tn1+DELTA>Tn THEN P2
  XLMAX=Tn1                 'PEAK FOUND
  PAUSE 50
RETURN
```

```
LCDSTART                    'INITIALIZES THE LCD AND DISPLAYS TOP LINE
                            'NO INPUT OR OUTPUT REGISTERS
                            'REGISTERS MODIFIED: DATAO
  PAUSE 200                 'WAIT FOR POWER TO STABILIZE
  RS=0                      'LCD RESET ROUTINE
  RW=0
  DATAO=%00110000
  GOSUB SEND
  GOSUB SEND
  GOSUB SEND
  DATAO=%00111000
  GOSUB SEND
  DATAO=%00001000
  GOSUB SEND
  DATAO=%00000001
  GOSUB SEND
  DATAO=%00000111
  GOSUB SEND
  DATAO=%00001111
  GOSUB SEND
  DATAO=%00000110
  GOSUB SEND
  RS=1
  DATAO=%01010011           'SEND S
  GOSUB SEND
  DATAO=%01010100           'T
  GOSUB SEND
  DATAO=%01000101           'E
  GOSUB SEND
  DATAO=%01010000           'P
  GOSUB SEND
  DATAO=%01010011           'S
  GOSUB SEND
  DATAO=%10100000           'SPACE
  GOSUB SEND
  DATAO=%10100000           'SPACE
  GOSUB SEND
  DATAO=%10100000           'SPACE
  GOSUB SEND
  DATAO=%01000100           'D
  GOSUB SEND
  DATAO=%01001001           'I
  GOSUB SEND
  DATAO=%01010011           'S
  GOSUB SEND
  DATAO=%01010100           'T
  GOSUB SEND
  DATAO=%01000001           'A
  GOSUB SEND
  DATAO=%01001110           'N
  GOSUB SEND
  DATAO=%01000011           'C
  GOSUB SEND
  DATAO=%01000101           'E
  GOSUB SEND
RETURN

SHOWSTEPS                   'DISPLAYS NUMBER OF STEPS
                            'INPUT: STEPS, DIST
                            'REGISTERS MODIFIED: DATAO
```

```
  RS=0
  RW=0                      'CURSOR LOCATION START OF 2nd LINE
  DATAO=%11000000
  GOSUB SEND
  RS=1                      'SEND STEPS
  DATAO=%00110000
  DATAO.LOWNIB=STEPS.HIGHBYTE.HIGHNIB
  GOSUB SEND
  DATAO.LOWNIB=STEPS.HIGHBYTE.LOWNIB
  GOSUB SEND
  DATAO.LOWNIB=STEPS.LOWBYTE.HIGHNIB
  GOSUB SEND
  DATAO.LOWNIB=STEPS.LOWBYTE.LOWNIB
  GOSUB SEND
  RS=0                      'CURSOR LOCATION 9th SPACE OF 2nd
  DATAO=%11001000
  GOSUB SEND
  RS=1                      'SEND DISTANCE
  DATAO=%00110000
  DATAO.LOWNIB=DIST.HIGHBYTE.HIGHNIB
  GOSUB SEND
  DATAO.LOWNIB=DIST.HIGHBYTE.LOWNIB
  GOSUB SEND
  DATAO.LOWNIB=DIST.LOWBYTE.HIGHNIB
  GOSUB SEND
  DATAO.LOWNIB=DIST.LOWBYTE.LOWNIB
  GOSUB SEND
RETURN


INCSTEPS                    'INCREMENTS STEPS AND DISTANCE. ALSO CONVERTS HEX
                            'INPUT TO DECIMAL
                            'INPUTS: STEPS, DIST, STRIDE
                            'REGISTERS MODIFIED: STEPS, DIST
  STEPS=STEPS+1
  IF STEPS.LOWBYTE.LOWNIB <= 9 THEN M1
  STEPS=STEPS+6
M1
  IF STEPS.LOWBYTE.HIGHNIB <= 9 THEN M2
  STEPS=STEPS+96
M2
  IF STEPS.HIGHBYTE.LOWNIB <= 9 THEN M3
  STEPS=STEPS+1536
M3
  DIST=DIST+STRIDE
  IF DIST.LOWBYTE.LOWNIB <= 9 THEN M4
  DIST=DIST+6
M4
  IF DIST.LOWBYTE.HIGHNIB <= 9 THEN M5
  DIST=DIST+96
M5
  IF DIST.HIGHBYTE.LOWNIB <= 9 THEN M6
  DIST=DIST+1536
M6
RETURN
```

注释

注释