EMBEDDED DEVELOPMENT

Electronic Compass: Tilt Compensation & Calibration

Authored by Mark Pedley
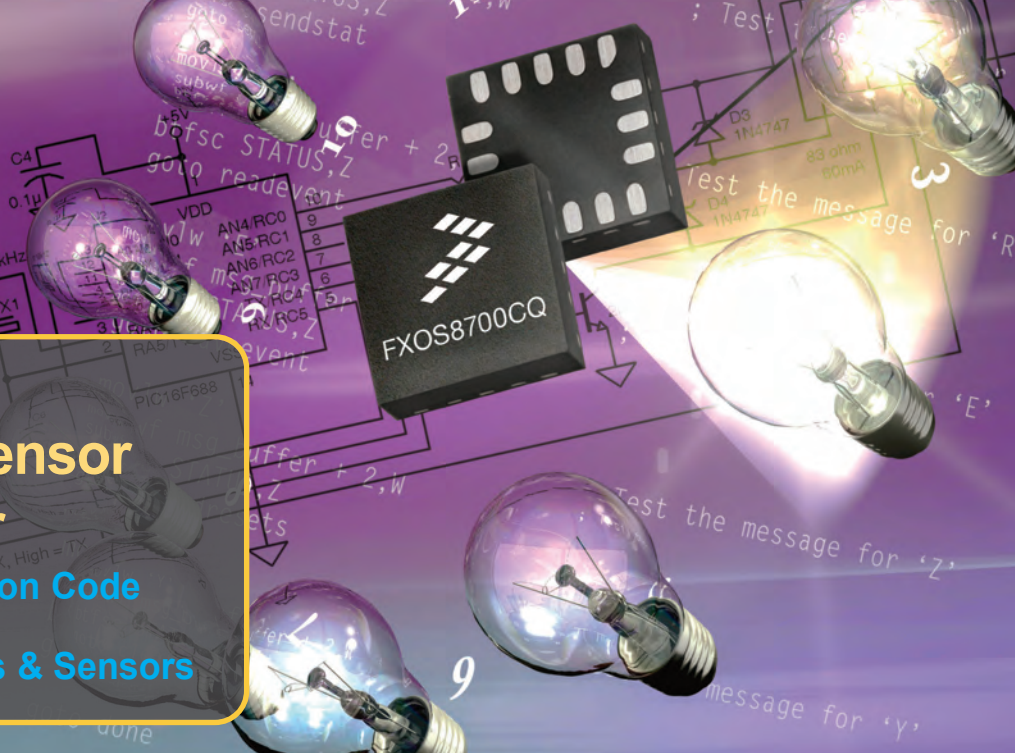
Reprint sponsored by:

freescale™

FXOS8700CQ

PLUS
The Mind of a Sensor Fusion Engineer

• **Innovative Sensor Fusion Code**
• **Inside the Design Lab**
• **Intelligent Use of MCUs & Sensors**

eCompass

Build and Calibrate a Tilt-Compensating Electronic Compass

A modern smartphone contains a built-in electronic compass (eCompass). How does the tilt compensation work, and how is the eCompass calibrated for the magnetic interference from the circuit board? This article describes how you can use the high-performance consumer accelerometers and magnetometers developed for the smartphone market to add a tilt-compensated eCompass to your own microcontroller project for less than $5.

Whenever a new high-volume consumer electronics market develops, the semiconductor companies are never far behind, providing excellent components at surprisingly low prices. The market for sensors in consumer products is a recent example. It all started with an accelerometer used to select between portrait and landscape display orientations and then, with the addition of a magnetometer, evolved into the electronic compass (eCompass) used to align street maps to the smartphone's geographic heading or to enable augmented reality overlays. As a result, high-volume pricing for smartphone accelerometer and magnetometer sensors is now less than $1 each.

A magnetometer sensor alone cannot provide an accurate compass heading for two reasons. First, the magnetic field measured at the magnetometer varies significantly with tilt angle. Second, the magnetometer requires calibrating not only for its own offset but also against spurious magnetic fields resulting from any nearby ferromagnetic components on the circuit board. This article describes how the accelerometer is used to compensate the magnetometer for tilt and includes a simple technique for calibrating the magnetometer.

SENSOR SELECTION

The accelerometer should be three axis and capable of operating in the ±2-g range with a minimum of 10 bits of resolution. The output of a 10-bit accelerometer operating in the ±2-g range will change by 512 counts as the accelerometer is rotated 180° from pointing downward to upward, giving an average sensitivity of one count per 0.35° change in tilt. This is more than adequate for tilt-compensation purposes.

It is important to check the accelerometer datasheet for the "0-g offset accuracy" which is the output when the accelerometer is in 0-g freefall. Since this value is a constant additive error on each accelerometer channel, it adds a bias in the calculated tilt angles, so look for accelerometers where this parameter does not exceed 50 mg.

The magnitude of the earth's geomagnetic field is typically about 50 µT with a horizontal component that varies over the earth's surface, from a maximum of about 40 µT down to 0 at the geomagnetic poles. If an eCompass is required to operate in horizontal geomagnetic fields down to 10 µT (in arctic Canada, for example) with a noise jitter of ±3°, then a back-of-the-envelope calculation indicates that a magnetometer with a maximum noise level of 0.5 µT is needed.

Most of my projects have used Freescale's MMA8451Q Xtrinsic three-axis, 14-bit accelerometer and MAG3110 three-axis magnetometer. The MMA8451Q is supplied in a 3-mm × 3-mm × 1-mm, 16-pin QFN package and provides a 14-bit data output with ±30-mg, 0-g offset accuracy. The MAG3110 magnetometer is supplied in a 2-mm × 2-mm × 0.85 mm, 10-pin DFN package and provides a measurement range of ±1,000 µT with 0.1-µT resolution and a noise level down to 0.25 µT. Both parts operate with a supply voltage between 1.95 V and 3.6 V.

Similar sensors are supplied by Asahi Kasei (AKM), Kionix, STMicroelectronics, and other manufacturers. Your best strategy is to go to the manufacturers' websites and make a list of those that provide samples in single units or low-volume

isting 1—C source code for the tilt-compensated eCompass

```
/* multiplicative conversion constants */
#define DegToRad 0.017453292F
#define RadToDeg 57.295779F

/* global variables */
float Psi, The, Phi;               /* yaw, pitch, roll angles in deg */
float Vx, Vy, Vz;                  /* hard iron calibration coefficients */

/* NED tilt-compensated e-Compass function */
void eCompass(float Bx, float By, float Bz, float Gx, float Gy, float Gz)
{
float sinAngle, cosAngle;/* sine and cosine */
float fBfx, fBfy, fBfz;  /* calibrated mag data in uT after tilt correction */

/* subtract off the hard iron interference computed using equation 9 */
Bx -= Vx;
By -= Vy;
Bz -= Vz;

/* calculate roll angle Phi (-180deg, 180deg) and sin, cos */
Phi = atan2(Gy, Gz) * RadToDeg;          /* Equation 2 */
sinAngle = sin(Phi * DegToRad);          /* sin(Phi) */
cosAngle = cos(Phi * DegToRad);          /* cos(Phi) */
/* de-rotate by roll angle Phi */
Bfy = By * cosAngle - Bz * sinAngle;     /* Equation 5 y component */
Bz = By * sinAngle + Bz * cosAngle;      /* Bz=(By-Vy).sin(Phi)+(Bz-Vz).cos(Phi) */
Gz = Gy * sinAngle + Gz * cosAngle;      /* Gz=Gy.sin(Phi)+Gz.cos(Phi) */

/* calculate pitch angle Theta (-90deg, 90deg) and sin, cos*/
The = atan(-Gx / Gz) * RadToDeg;         /* Equation 3 */
sinAngle = sin(The * DegToRad);          /* sin(Theta) */
cosAngle = cos(The * DegToRad);          /* cos(Theta) */
/* de-rotate by pitch angle Theta */
Bfx = Bx * cosAngle + Bz * sinAngle;     /* Equation 5 x component */
Bfz = -Bx * sinAngle + Bz * cosAngle;    /* Equation 5 z component */

/* calculate yaw = ecompass angle psi (-180deg, 180deg) */
Psi = atan2(-Bfy, Bfx) * RadToDeg;       /* Equation 7 */

return;
}
```

packs of up to five devices. With a bit of luck, you may be able to get both the accelerometer and magnetometer sensors for free. Add a handful of decoupling capacitors and pull-up resistors and you should be well within the $5 component cost.

Each reader has a preferred microcontroller to read the raw data from the two sensors and implement the eCompass. This article assumes the microcontroller provides an I²C bus to interface to the sensors, supports floating-point operations whether natively or through software emulation libraries, and has a few spare bytes of program and data memory to add the 15 executable lines of eCompass code in Listing 1.

LAYOUT & BOARD BRING-UP

Once you've selected your sensors, the next step is to design the accelerometer and magnetometer daughterboard with I²C bus connection to the microcontroller. Reference schematics for the MMA8451Q and MAG3110 are provided in the sensor datasheets and reproduced in Figure 1.

Don't waste any time rotating the accelerometer or magnetometer packages to align their x-, y-, and z-sensing directions to each other since this will be fixed later in software. But do ensure the sensor board will not be mounted in the immediate vicinity of any ferromagnetic materials since these will produce a constant additive magnetic field termed the "hard-iron field." The most

common ferromagnetic materials are iron, steel, ferrite, nickel, and cobalt. Non-ferromagnetic materials are all safe to use (e.g., aluminum, copper, brass, tin, silver, and gold).

The calibration process described later enables the estimation and software subtraction of any hard-iron field, but it's good practice to minimize hard iron interference at the design stage. Remember, a current trace will create a cylindrical magnetic field that falls off relatively slowly with the inverse of distance, so place the magnetometer as far away from high current traces as possible. A 0.1-A current trace at 10-mm distance will produce a 2-µT magnetic field, four times our 0.5-µT error budget, only reducing to 0.5 µT at a 40-mm distance. More detailed layout guidance is provided in Freescale Semiconductor's application note AN4247: "Layout Recommendations for PCBs Using a Magnetometer Sensor."

You'll be surprised at the number of features implemented in the latest consumer sensors (i.e., freefall detection, high- and low-pass filtering options, automatic portrait and landscape detection, etc.), but disable all these since you simply want the raw accelerometer and magnetometer data. Configure the accelerometer into a 2-g range and check that you can read the x, y, and z accelerometer and magnetometer data (in units of bit counts) from the sensors' internal registers at a sampling rate of between 10 Hz and 50 Hz. Smartphones commonly use IDH3 to minimize power consumption while anything above 50 Hz is overkill. Check the accelerometer datasheet for the conversion factor between counts and g (4,096 counts per g for the MMA8451Q in ±2-g mode) and use this to scale the x, y, z accelerometer readings into units of g. Do the same for the x, y, z magnetometer data again taking the conversion factor from the magnetometer datasheet (10 counts per µT for the MAG3110).
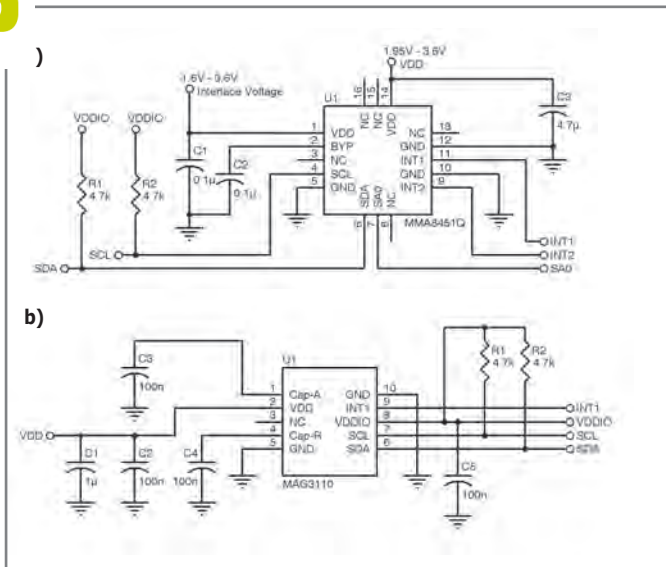
Figure 1—Schematics for **(a)** MMA8451Q and **(b)** MAG3110 sensors

COORDINATE SYSTEM

The equations and C software in Listing 1 use the "aerospace," or "x-North y-East z-Down," coordinate system depicted in Photo 1. This defines the initial eCompass orientation to be where the x-axis points north, the y-axis points east, and the z-axis points downwards. The three orientation angles, roll (ϕ), pitch (θ), and compass heading, or yaw (ψ)—are defined as clockwise rotations about the positive x, y, and z axes— respectively. Photo 1 also shows the earth's gravitational vector which points downward with magnitude of 1 g or 9.81 ms^{-2} and the earth's geomagnetic field vector, which points downward from horizontal (in the northern hemisphere) by the inclination angle δ to give a horizontal component $B_0\cos\delta$ and a downward component $B_0\sin\delta$. Based on how your eCompass housing will be held, you should be able to assign the compass-pointing direction or x-axis, the downward or z-axis, and the y-axis, which should point to the right to complete a right-handed coordinate system.

Photo 1—The aerospace north-east-down coordinate system

AXIS ALIGNMENT & MAGNETIC CALIBRATION

You now need to align the sensor data to the aerospace coordinate system. As with all work with magnetometers, this should be performed on a wooden table well away from any laboratory power supplies or steel furniture. Place the eCompass flat and upright so the z-axis points downward and is aligned with gravity. Check that the accelerometer z-axis reads approximately 1 g and the x- and y-axes are near 0. Invert the eCompass so its z-axis points upward and check that the z-axis now reads approximately −1 g. Repeat with the x- and y-axes pointing downward and then upward and check that the x- and y-axis accelerometer readings are near 1 g and −1 g, respectively. It's not important if the accelerometer readings are a few tens of mg away from the required reading since all you're doing here is correcting for gross rotations of the sensor packages and the sensor daughterboard in multiples of 90°. Any needed correction will be unique for your board layout and mounting orientation but will be no more complicated than "swap the x- and y-accelerometer channels and negate the z-channel reading." Code this accelerometer axis mapping into your software and don't touch it again.

Figure 2 may help explain this visually. The accelerometer sensor measures both gravity and linear acceleration and, in the absence of any linear acceleration (as is the case when sitting on a desk), the magnitude of the accelerometer reading will always equal 1 g, and therefore, lie on the surface of a 1-g sphere, irrespective of the orientation. The six measurements lie on the vertices of an octahedron inscribed within the 1-g sphere and the axis mapping simply rotates and reflects the octahedron as needed until the accelerometer channels are correctly aligned.

The magnetometer axis alignment is similar to that of the accelerometer, but makes use of the geomagnetic field vector. Place the eCompass flat, upright, and pointing northward and then rotate in yaw angle by 270° to the east, south, and finally west. The x-channel magnetometer reading should be a maximum when the eCompass is pointed north and a minimum when pointed south. The y-channel magnetometer reading should be a minimum when the eCompass is pointed east and a maximum when pointed west. The z-channel reading should be approximately constant since the vertical component of the geomagnetic field is constant irrespective of rotation in yaw.

Then invert the eCompass on the desk and repeat the process. As before, the magnetometer x-axis reading should be a maximum when the eCompass is pointed north and a minimum when pointed south. But now, because of the inverted position, the magnetometer y-axis should be a maximum when the eCompass is pointed east and a minimum when pointed west. The magnetometer z-axis reading should still be constant but, in the northern hemisphere, lower than the previous upright readings since the magnetometer z-axis is now aligned against the downward component of the geomagnetic field vector.

Figure 3 shows upright and inverted magnetometer measurements taken in the northern hemisphere with a 270° compass rotation. The maximum and minimum of the x- and y-axis magnetometer measurements occur at the expected angles and the z-axis measurement is less when inverted than when upright. These magnetometer axes are therefore
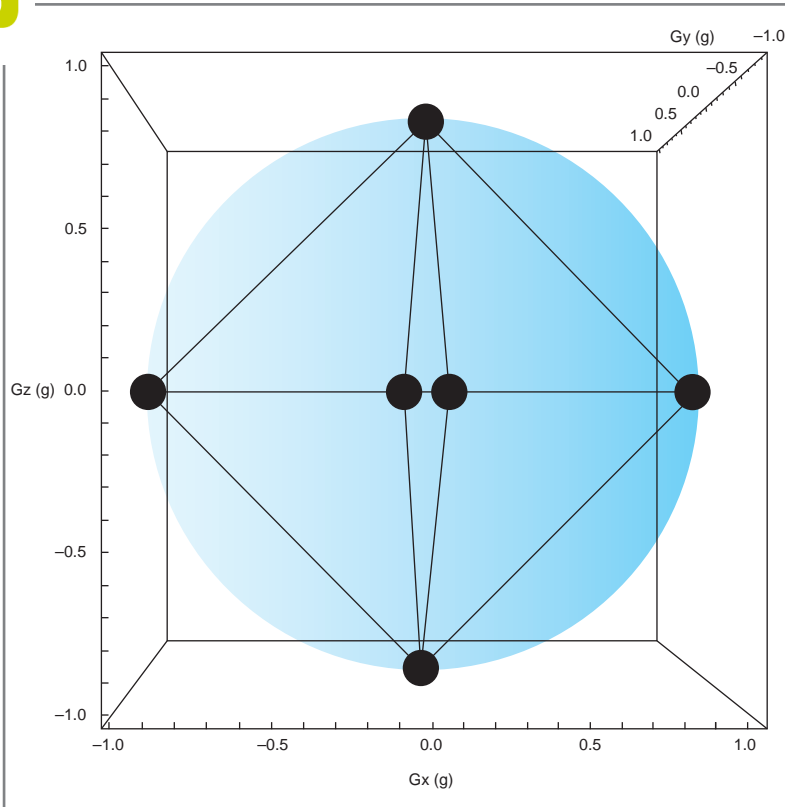
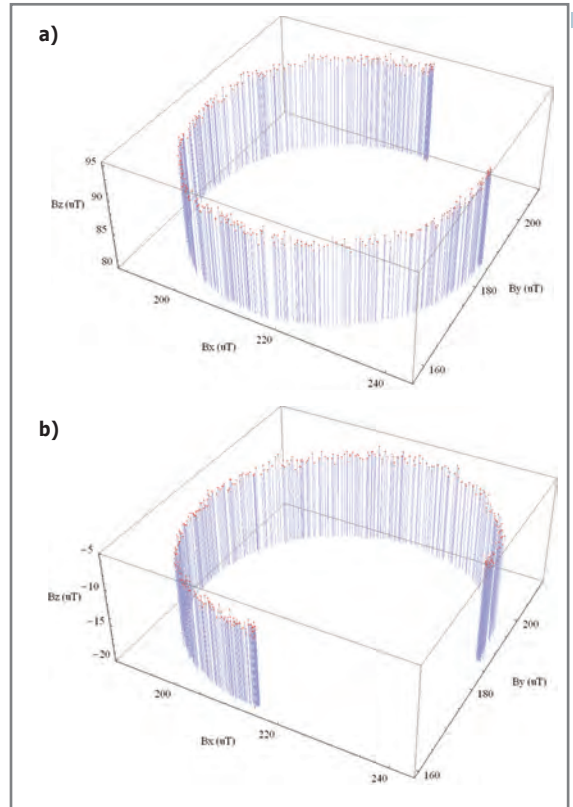**Figure 2**—Accelerometer axis alignment points

Figure 3—The upright (a) and inverted (b) magnetometer measurements

correctly aligned but, as with the accelerometer correction, swap and negate the measurements from your three magnetometer channels as needed until correctly aligned and then lock down this part of your code.

A lot can be learned by closely looking at the measurements in Figure 3. The x- and y-magnetometer measurements lie on a circle with radius of approximately 25 µT enabling us to deduce that the horizontal geomagnetic field is approximately 25 µT. But the measurements are offset from zero by the magnetic "hard iron" interfering field, which results from both permanently magnetized ferromagnetic materials on the circuit board and from a zero-field offset in the magnetometer sensor itself. Consumer sensor manufacturers long ago realized it was pointless to accurately calibrate their magnetometers when their target market is smartphones, each with a different hard-iron interfering field. The magnetometer sensor offset is, therefore, calibrated together with the circuit board hard-iron magnetic field. For now, simply note that the x and y components of the hard iron offset have values of approximately 215 µT and 185 µT. A simple method to determine all three hard-iron components is described later.

CALCULATION OF THE ROLL & PITCH ANGLES

In the absence of linear acceleration, the accelerometer reading g after arbitrary rotation in yaw (ψ), then pitch (θ), and finally roll (ϕ) from the smartphone starting position (flat, upright, and pointing to magnetic north) is simply the earth's gravitational field vector (1 g downward in the positive z direction) after application of roll, yaw, and pitch rotation

matrices $R_X (\phi)$, $R_Y (\theta)$, and $R_Z (\psi)$. The resulting accelerometer reading is:

$$
\begin{aligned}
G = \begin{pmatrix} G_X \\ G_Y \\ G_Z \end{pmatrix} &= R_X (\phi)\, R_Y (\theta)\, R_Z (\psi) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \\
&\quad \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -\sin\theta \\ \cos\theta\,\sin\phi \\ \cos\theta\,\cos\phi \end{pmatrix}
\end{aligned}
\qquad [1]
$$

It's worth emphasizing the point that the mathematics of the tilt-compensated eCompass assume that the accelerometer reading is entirely the rotated gravitational field without any linear acceleration. This is a perfectly reasonable assumption for the smartphone market, where users will naturally hold the phone steady to see the compass display, but readers should be aware that they will get erroneous compass headings from an eCompass located in a radio-controlled airplane pulling high g aerobatics or even when shaking the eCompass.

It should be no surprise that Equation 1 has no dependence on the yaw (or compass heading) angle ψ. All accelerometers are unaffected by rotations about a vertical axis. Of course, if accelerometers could measure rotation about the vertical gravitational field, we would not need a magnetometer to compute the compass heading.

Equation 1 can easily be solved for the roll angle ϕ and pitch angle θ from the x, y, and z accelerometer channel readings G_X,
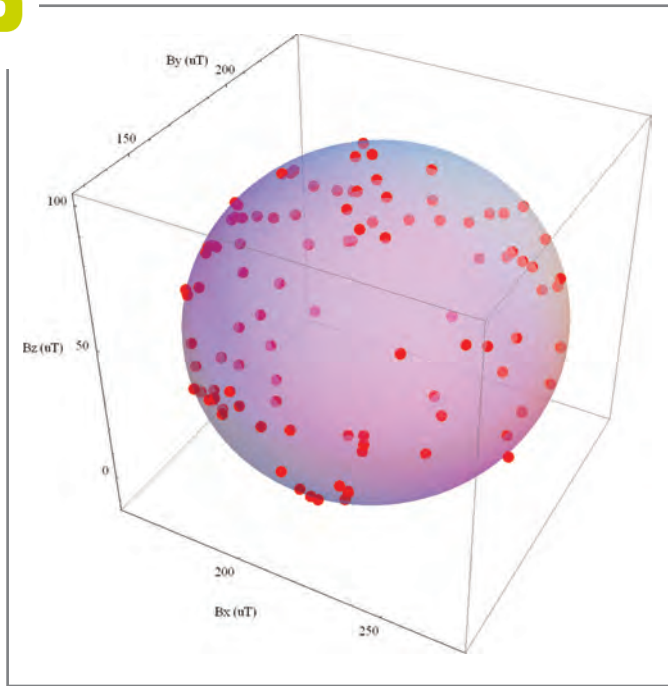
Figure 4—Distribution of magnetometer measurements under arbitrary rotation

G_Y, and G_Z. Simple algebra gives:

$$\tan\phi = \frac{G_Y}{G_Z} \quad [2]$$

$$\tan\theta = \left(\frac{-G_X}{G_Y\sin\phi + G_Z\cos\phi}\right) \quad [3]$$

The range of the roll angle ϕ is $-180°$ to $180°$ so the C atan2 function is used in Listing 1 to calculate θ. The pitch angle θ only varies between $-90°$ and $90°$, so the standard C atan function is used in Listing 1 to calculate θ. Don't try to simplify the algebra by noting that $G_X = -\sin\theta$ in Equation 1 and computing $\theta = \sin^{-1}(-G_X)$. Even without linear acceleration, sensor noise will make G_X occasionally go outside the range of -1 g to 1 g, making the arcsine calculation nonsensical. The arctangent calculations in Listing 1 are stable in comparison since the atan and atan2 functions can take any argument from $-\infty$ to ∞.

The next step is to program up Equations 2 and 3, remembering that most programming languages return angles in radians requiring a multiplication by $180/\pi$, or approximately 57.3, to convert to degrees. Place the eCompass flat on the table and check that the calculated roll and pitch angles are approximately $0°$. Rotate the eCompass upward and check that the computed pitch angle is positive, reaches $90°$ when pointing vertically upward, and then decreases to $0°$ when the eCompass is inverted. Repeat with the eCompass pointed slightly downward and check that the computed pitch is negative. Then apply a positive roll rotation by raising the left side of the eCompass and check that the computed roll angle is positive, passes through $90°$ when the eCompass is on its right edge, and finally reaches $180°$ when the eCompass is flat but inverted. Repeat the roll test by lowering the left side of the eCompass and check that the compute roll angle is negative. If you find

the angles are swapped or negated from what is expected, you'll need to return to the accelerometer axis alignment procedure.

CALCULATING THE COMPASS HEADING ANGLE

In a similar manner as Equation 1, the magnetometer reading B can be modeled as the sum of the rotated geomagnetic field plus the hard-iron interfering field vector V to give:

$$B = \begin{pmatrix} B_X \\ B_Y \\ B_Z \end{pmatrix} = R_X\,(\phi)\,R_Y\,(\theta)\,R_Z\,(\psi)\,B_0 \begin{pmatrix} \cos\delta \\ 0 \\ \sin\delta \end{pmatrix} + V \quad [4]$$

If you subtract off the hard-iron interference V from both sides of Equation 4 and premultiply by the inverse roll and then inverse pitch rotation matrices, you obtain:

$$R_Z\,(\psi)\,B_0 \begin{pmatrix} \cos\delta \\ 0 \\ \sin\delta \end{pmatrix} = B_0 \begin{pmatrix} \cos\psi\,\cos\delta \\ -\sin\psi\,\cos\delta \\ \sin\delta \end{pmatrix} = R_Y\,(-\theta)\,R_X\,(-\phi) \begin{pmatrix} B_X - V_X \\ B_Y - V_Y \\ B_Z - V_Z \end{pmatrix}$$

$$= \begin{pmatrix} (B_X - V_X)\cos\theta + (B_Y - V_Y)\sin\theta\sin\phi + (B_Z - V_Z)\sin\theta\cos\phi \\ (B_Y - V_Y)\cos\phi - (B_Z - V_Z)\sin\phi \\ -(B_X - V_X)\sin\theta + (B_Y - V_Y)\cos\theta\sin\phi + (B_Z - V_Z)\cos\theta\cos\phi \end{pmatrix} \quad [5]$$

The x and y components of Equation 5 are shown below in Equation 6:

$$B_0\cos\psi\cos\delta = (B_X - V_X)\cos\theta + (B_Y - V_Y)\sin\theta\sin\phi + (B_Z - V_Z) \quad [6]$$

$$\sin\theta\cos\phi - B_0\sin\psi\cos\delta = (B_Y - V_Y)\cos\phi - (B_Z - V_Z)\sin\phi$$

Taking the ratio of the y and x components in Equation 6 gives the tilt-compensated compass heading angle ψ as:

$$\tan\psi = \left(\frac{-(B_Y - V_Y)\cos\phi + (B_Z - V_Z)\sin\phi}{(B_X - V_X)\cos\theta + (B_Y - V_Y)\sin\theta\sin\phi + (B_Z - V_Z)\sin\theta\cos\phi}\right) \quad [7]$$

We know the roll ϕ and pitch θ angles from the accelerometer so all that remains to compute the compass heading angle ψ from Equation 7 is to find a simple way to calculate the components of the interfering hard-iron magnetic field V. But, if you subtract V from both sides of Equation 4 and then calculate the resulting magnitude squared, you obtain Equation 8. The left-hand side is expanded using Pythagoras's theorem and the right-hand side is simplified by observing that the geomagnetic field magnitude remains B_0 irrespective of the rotations applied.

$$\left\Vert \begin{matrix} B_X - V_X \\ B_Y - V_Y \\ B_Z - V_Z \end{matrix} \right\Vert^2 = \left\Vert R_X\,(\phi)\,R_Y\,(\theta)\,R_Z\,(\psi)\,B_0 \begin{pmatrix} \cos\delta \\ 0 \\ \sin\delta \end{pmatrix} \right\Vert^2$$

$$\Rightarrow (B_X - V_X)^2 + (B_Y - V_Y)^2 + (B_Z - V_Z)^2 = B_0{}^2 \quad [8]$$

Equation 8 simply states that under arbitrary rotations in roll,

ch, and yaw, the magnetometer readings lie on the surface a sphere with radius B_0 centered at the hard-iron interference V_X, V_Y, and V_Z. Figure 4 shows actual magnetometer measurements taken at random orientations together with a sphere fitted to the measurements with radius 57.1 µT and centered at V_X = 216.4 µT, V_Y = 183.6 µT, and V_Z = 44.0 µT. To obtain your own hard-iron calibration, simply record and plot the magnetometer readings under random orientations and estimate your own hard-iron correction V_X, V_Y, V_Z from the center of the resulting sphere. A simple but effective technique is to rotate the eCompass in a figure of eight twisting motions for a few seconds, record the minimum and maximum magnetometer readings, and compute the hard-iron calibration from their average using Equation 9.

$$\begin{pmatrix} V_X \\ V_Y \\ V_Z \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \max(B_X) + \min(B_X) \\ \max(B_Y) + \min(B_Y) \\ \max(B_Z) + \min(B_Z) \end{pmatrix} \qquad [9]$$

Equation 9 completes the information needed in Equation 7 to compute the compass heading ψ. Readers interested in a more sophisticated algorithm for least squares fit to the hard-iron calibration should refer to Freescale Semiconductor's Application Note AN4246: "Calibrating an eCompass in the Presence of Hard and Soft-Iron Interference" and visit Freescale's eCompass website for additional C source code and details.

The entire eCompass with tilt-compensation is provided as C code with just 15 executable lines in Listing 1. Readers wishing to add a digital low-pass filter to reduce jitter on the computed angles should take care to use modulo 360° arithmetic to avoid averaging angles jittering between −179° and 179° (pointing south) to 0° (pointing north). Freescale's Application Note AN4248 "Implementing a Tilt-Compensated eCompass Using Accelerometer and Magnetometer Sensors," contains further details and C source code for those interested. 🖼

Mark Pedley (mark.pedley@freescale. com) has a Physics degree from Oxford University and now works on sensor fusion algorithms for Freescale Semiconductor in Tempe, Arizona.

RESOURCES
Asahi Kasei Corp., www.asahi-kasei.co.jp/asahi/en.

Freescale Semiconductor, Xtrinsic eCompass Software, www.freescale.com/ecompass.

———, Application Note AN4247: "Layout Recommendations for PCBs Using a Magnetometer Sensor," 2012, www.freescale.com/files/sensors/doc/app_note/AN4247.pdf.

———, Application Note AN4246: "Calibrating an eCompass in the Presence of Hard and Soft-Iron Interference," 2012, http://cache.freescale.com/files/sensors/doc/app_note/AN4246.pdf.

———, Application Note AN4248: "Implementing a Tilt-Compensated eCompass using Accelerometer and Magnetometer Sensors," 2012, http://cache.freescale.com/files/sensors/doc/app_note/AN4248.pdf.

Kionix, Inc., www.kionix.com.

ST Microelectronics, www.st.com.

SOURCES
MMA8451Q Xtrinsic accelerometer and MAG3110 magnetometer
Freescale, Inc. | www.freescale.com

NEED-TO-KNOW INFO

Knowledge is power. In the computer applications industry, informed engineers and programmers don't just survive, they *thrive* and *excel*. For more need-to-know information about some of the topics covered in this article, the *Circuit Cellar* editorial staff recommends the following content:

—

Mini Rover 7
Electronic Compassing for Mobile Robotics
by Joseph Miller
Circuit Cellar **165, 2004**
Electronic compassing is an excellent way to provide absolute heading information for a mobile robot. Joseph details how it factors in his Mini Rover 7 robot design, which he modeled after the NASA/JPL Rocky 7 Mars rover. Topics: Electronic Compassing, Robot, Heading, Navigation

—

Digital Compassing
Build a Tilt-Compensated Digital Magnetic Compass
by Vadym Grygorenko
Circuit Cellar **188, 2006**
Vadym built a PSoC-based digital compass with a three-axis magneto-resistive sensor featuring tilt compensation.The system stores calibration data in the PSoC's internal EEPROM. Topics: Tilt Compensation, Compass, Calibration

—

Find a New Direction
A Low-Power Digital Compass
by Jess Marroquin
Circuit Cellar **253, 2011**
You can use a low-power microcontroller to build a digital compass. This article details the planning, design, and programming processes. Topic: Digital Compass, Magnetic Sensor, Low-Power Consumption

—

Launching the "Roboat"
Navigation by GPS and Digital Compass
by Riccardo Rocca
Circuit Cellar **115, 2000**
Riccardo wanted to test the feasibility of driving an object using a GPS and a digital compass. In this case, the object is the "Roboat" model boat that can sail autonomously along a course planned in advance. Topic: Digital Compass, GPS, Navigation, Proportional Servo

Go to *Circuit Cellar*'s webshop to find these articles and more: www.cc-webshop.com