Software for Tilt-Compensated eCompass with Magnetic Calibration (v3 Release)

User Guide

Contents

1 Introduction

This user guide is part of the documentation for the tilt-compensated eCompass functions in Freescale's Xtrinsic eCompass and magnetic calibration software provided under license at www.freescale.com/ecompass. Its use and distribution are controlled by the license agreement. This license agreement restricts the use of this software to platforms using Freescale accelerometer and magnetometer but does not require that the software run on a Freescale processor.

1.1 Enhancements in version 3

The main changes in version 3 of the software relative to version 2 are:

- Consolidation of variables and arrays into a smaller number of data structures
- Reduction of RAM, flash memory and floating point calculations
- Addition of 9 new Application Notes providing further visibility into the algorithms and software

Users who are happy with the performance of version 2 need not upgrade. New users and those for whom memory and processing overhead is a particular issue should use version 3.

1.2 Technical support

Questions on the software or bug reports should be addressed to sfusion@freescale.com.

Freescale has found that the most common source of errors on embedded systems is improper alignment of the physical sensors to the coordinate system of the final product. If you are observing roll, pitch or yaw angles that are inverted or 90° different from that expected, or if the compass heading varies wildly as the circuit board is rotated in pitch and roll, please recheck your sensor alignment using the procedures documented in AN4696.

1.3 Description

This software is intended for use on embedded systems, such as those found in smartphones or tablets comprising a processor connected to a three-axis accelerometer and three-axis magnetometer to implement a tilt-compensated eCompass. This version of the software does not support a gyro angular rate sensor, although that may be added in future releases.

The software is provided as standard ANSI C software suitable for any embedded processor. It does perform floating point calculations, but any modern 32-bit μC with access to a floating point emulation library should be able to run the software.

In order to demonstrate the operation of the algorithms, the software is delivered:

- With a simulation sensor driver which simulates the outputs of accelerometer and magnetometer sensors at different orientations and in different levels of magnetic hard and soft iron interference
- As a pre-built Windows console application

The purpose of the simulation sensor driver and the Windows application is to provide confidence to users that the software is functioning correctly before the step of moving to an embedded processor and real physical sensors. The Windows application is purely for demonstration purposes and it is expected that customers will almost immediately build the reference software on their target development system.

1.4 Sensor drivers

The supplied software does not include reference drivers for the accelerometer and magnetometer sensors and instead contains a simulated sensor driver. Experience has shown that example drivers for physical sensors are of limited use and actually cause more confusion than they remove since:

- Each customer hardware platform will be different
- The eCompass and magnetic calibration software is sensor fusion software abstracted from the low level sensor drivers and
- Writing the I^2C drivers for the accelerometer and magnetometer sensors is a trivial task involving I^2C writes to configure the sensors and then I^2C reads of the internal sensor registers to access the sensor data

1.5 Candidate processors

Customers looking at candidate processors for new designs should investigate Freescale's range of Kinetis ARM® Cortex™ devices at Kinetis ARM Cortex Microcontrollers.

The software executes comfortably on Kinetis L series devices using the ARM® Cortex™ M0+ architecture Kinetis L Series MCUs.

1.6 Contents

The software and documentation comprises the following files:

- This User Guide MAGCALSWUG
- Nine Application Notes
- A prebuilt Windows console application main.exe
- ANSI C source code files as follows:
    - main.c: the main control loop with consoler input and output
    - orientation.c: orientation functions
    - magnetic.c: magnetic calibration functions
    - matrix.c: matrix algebra functions
    - simulation.c: the simulated sensor driver
    - include.h: compile time constants

1.7 Customer steps

It is expected that users of the software will take the following sequence of steps:

1. Run the prebuilt application `main.exe` on a Microsoft Windows PC to familiarize themselves with the concept and operation of the software
2. Build and run the software on their target processor platform to duplicate the results seen with the Windows PC application main.exe
3. Replace the sensor simulation driver with their own drivers to physical accelerometer and magnetometer sensors

Depending on the configuration of their Windows computer, users may see the error message below when running the file `main.exe`:

"This application has failed to start because `MSVCR100.dll` was not found. Re-installing the application may fix this problem".

If this message appears, download the Microsoft package from one of the two sites below:

- For 32 bit Windows: www.microsoft.com/download/en/details.aspx?id=5555
- For 64 bit Windows: www.microsoft.com/download/en/details.aspx?id=14632

1.8 Magnetic calibration solvers

The 4-element calibration algorithm can fit the three components of the hard iron vector plus the geomagnetic field strength. The soft iron matrix is the identity matrix.

The 7-element calibration algorithm can fit the parameters of the 4-element algorithm plus the three diagonal elements of the soft iron matrix. The off-diagonal elements of the soft iron matrix remain zero.

The 10-element calibration algorithm can fit the parameters of the 7-element algorithm plus the three off-diagonal elements of a symmetric soft iron matrix.

The file `magnetic.c` contains the source code for 4-element and 7-element magnetic calibration functions. The highest performing 10-element calibration solver is provided as part of the prebuilt application `main.exe` but source code is not provided by Freescale. Users interested in receiving object code for the 10 element calibration solver compiled for ARM Thumb2 processors should email their request to sfusion@freescale.com.

2 Windows PC Demonstration

2.1 Summary

The Windows PC executable `main.exe` is provided to allow customers to quickly understand the eCompass and magnetic calibration software. It is intended as a brief stepping stone to the next step where the supplied ANSI C source code is built on the customer's embedded system.

No attempt has been made to provide a sophisticated user interface, because the target application is an embedded system and user interface software simply complicates the understanding of the underlying software.

Because Windows PCs do not typically have embedded sensors, although this is changing with Windows 8 notebooks and tablets, the software includes a simulated sensor driver to simulate the output of the accelerometer and magnetometer sensors at random orientation angles and with user-specified levels of hard and soft iron interference.

Finally, users should note that the prebuilt executable `main.exe` does include the high-performing 10 element calibration software, but this is not provided in the supplied source code. This is to protect Freescale's intellectual property. Customers interested in the 10 element calibration software should follow the procedure in Magnetic calibration solvers.

On launch, the application `main.exe` displays the command line interface shown in Figure 1.

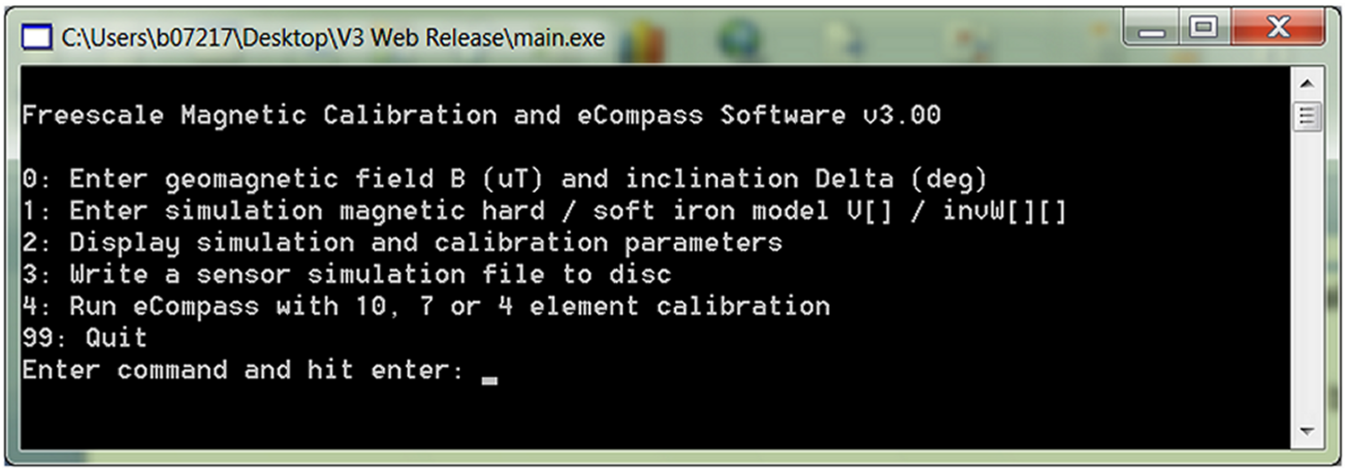


Figure 1. Application on launch

2.2 Option 0: Enter geomagnetic field

Type "0" to enter the local geomagnetic field strength B (in μT) and the magnetic inclination angle delta (in degrees). Separate these entries with a space. These values are only used for simulation of the magnetic sensor output and are not required when the sensor simulation functions are replaced with the real sensor drivers. Neither the eCompass nor magnetic calibration algorithms require a prior knowledge of the local geomagnetic field. Also, the algorithms will operate anywhere in the world where a normal needle compass will operate, which is anywhere except in the immediate vicinity of the north and south geomagnetic poles.

In Figure 2, the values 51 μT for the geomagnetic field strength and 55° for the inclination angle have been entered.

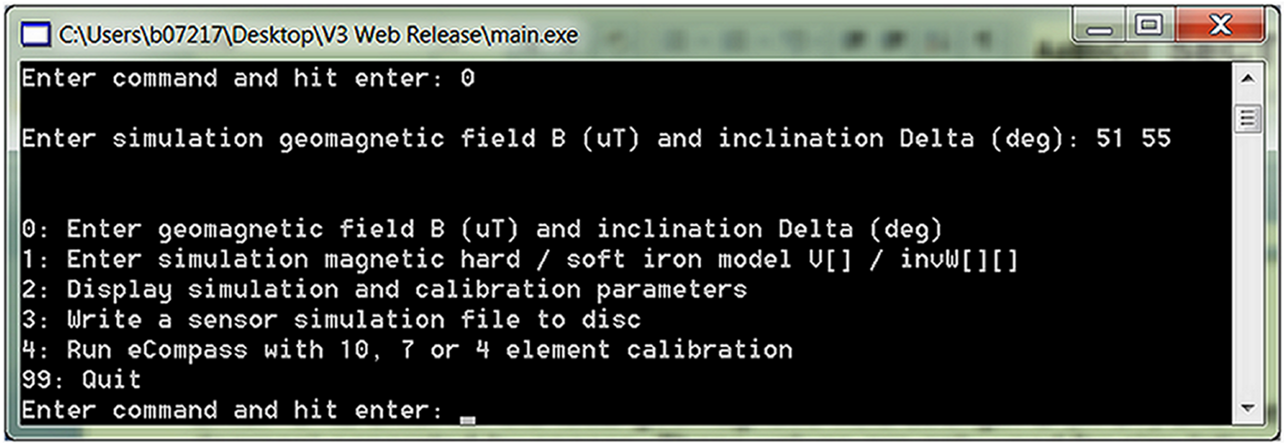


Figure 2. Option 0

The precise values entered here are not particularly important for verification of the software. A value of 0 μT field strength or an inclination angle of +90° or –90° should, however, be avoided since these correspond to the magnetometer sensor being completely shielded from the geomagnetic field and the sensor being at the north and south geomagnetic poles respectively.

Typical values for the industrialized northern hemisphere are 50 μT magnitude with a 50° (downward) inclination angle. If option 0 is never selected, the software defaults to simulating the magnetometer sensor using a geomagnetic field strength of 50 μT and an inclination angle of 50°.

2.3 Option 1: Enter hard and soft iron model parameters

Type "1" to enter the components of the hard and soft iron model to be used to simulate the magnetic distortions measured by the magnetometer sensor. In the final product, these distortions are imposed naturally on the magnetometer measurements by ferromagnetic components on the circuit board. The role of the magnetic calibration algorithms is to determine the parameters of this interference. The PC demonstration shows how the magnetic calibration algorithms can determine these parameters from magnetometer measurements and then remove their effects to give an accurate compass heading. In the simplest possible terms, the values entered are arbitrary and for demonstration purposes only.

```
C:\Users\b07217\Desktop\V3 Web Release\main.exe
Enter command and hit enter: 1

Enter simulation hard iron (-1000uT to 1000uT): Vx, Vy, Vz: 212 -321 34

Diagonal elements of inverse soft iron (range typically 0.5 to 2)
Enter simulation invW[0][0] [1][1] [2][2]: 1.2 0.9 1.1

Off-diagonal elements of inverse soft iron (range typically -0.5 to 0.5)
Enter simulation inv soft iron invW[0][1] [0][2] [1][2]: 0 0 0

Matrix is diagonal but not identity: 7 element model or above should be used
Determinant of inverse soft iron matrix is     1.188

Simulation inverse soft iron matrix invW (normalized)
Row 0     1.13303     0.00000     0.00000
Row 1     0.00000     0.84977     0.00000
Row 2     0.00000     0.00000     1.03861

Simulation forward soft iron matrix W (normalized)
Row 0     0.88259     0.00000     0.00000
Row 1     0.00000     1.17678     0.00000
Row 2     0.00000     0.00000     0.96282

Simulation ellipsoid matrix A = invW^T * invW (normalized)
Row 0     1.28376     0.00000     0.00000
Row 1     0.00000     0.72212     0.00000
Row 2     0.00000     0.00000     1.07872
```

Figure 3. Option 1

First, enter the hard iron offset vector V in μT separated by spaces. In Figure 3, the values 212 μT, –321 μT and 34 μT have been specified. The recommended range is –1000 μT to +1000 μT which corresponds to the typical range of a magnetometer sensor and to the typical maximum level of hard iron interference observed in a smartphone. In passing, it is worth noting that 1000 μT is 20x the value of the geomagnetic field strength and would result in complete jamming of the smartphone electronic compass, if it were not characterized and removed by the magnetic calibration software.

The parameters of the soft iron interference are then entered on the next two lines. The diagonal elements are entered on the first line and the off-diagonal elements on the second line. Here, diagonal soft iron values of 1.2, 0.9 and 1.1 and 0 off-diagonal elements have been entered. Again, it must be emphasized that these values are naturally imposed on the magnetometer measurements in the final embedded product by ferromagnetic components on the circuit board and they are computed from the magnetometer measurements by the calibration algorithms and removed from the magnetometer measurements without user involvement. The operation of the eCompass and magnetic calibration algorithms are entirely automated in the final product.

The software identifies that the off-diagonal elements of the simulated soft iron matrix are zero and advises that the 7-element or 10-element calibration algorithm should be used. In reality, the developer of an embedded implementation of the eCompass and magnetic calibration will include just one of the 4-, 7- or 10-element magnetic calibration solvers with the decision made on a performance versus processing-overhead tradeoff. The demonstration software provides this advice to avoid any confusion resulting from the user entering a more sophisticated soft iron matrix than the selected calibration algorithm can solve.

The un-normalized inverse soft iron matrix as entered by the user in the screenshot in Figure 3 has the value:

$$\boldsymbol{W}^{-1} \sim \begin{pmatrix} 1.2 & 0.0 & 0.0 \\ 0.0 & 0.9 & 0.0 \\ 0.0 & 0.0 & 1.1 \end{pmatrix} \quad (1)$$

The determinant of the simulation inverse soft iron matrix is then computed (here 1.188). The normalized inverse soft iron matrix is then computed and displayed:

$$\boldsymbol{W}^{-1} = \begin{pmatrix} 1.1333 & 0.0 & 0.0 \\ 0.0 & 0.84977 & 0.0 \\ 0.0 & 0.0 & 1.03861 \end{pmatrix} \quad (2)$$

The forward soft iron matrix $\boldsymbol{W}$ is then computed by inverting the normalized inverse soft iron matrix and the result displayed. The forward soft iron matrix distortion is applied to the simulation magnetometer measurements and then removed by the inverse soft iron matrix computed by the calibration algorithms.

$$\boldsymbol{W} = \begin{pmatrix} 0.88259 & 0.0 & 0.0 \\ 0.0 & 1.17678 & 0.0 \\ 0.0 & 0.0 & 0.96282 \end{pmatrix} \quad (3)$$

The simulation ellipsoid matrix $\boldsymbol{A} = (\boldsymbol{W}^{-1})^T(\boldsymbol{W}^{-1})$ is then computed and displayed. The ellipsoid matrix A models the distribution of magnetometer measurements and is discussed further in application note AN4684. Here, this matrix has value:

$$\boldsymbol{A} = \begin{pmatrix} 1.28376 & 0.0 & 0.0 \\ 0.0 & 0.72212 & 0.0 \\ 0.0 & 0.0 & 1.07872 \end{pmatrix} \quad (4)$$

2.4 Option 2: Display simulation and calibrated parameters

Enter "2" to display a summary of the input simulation and the output computed calibration values, as shown in Figure 4.

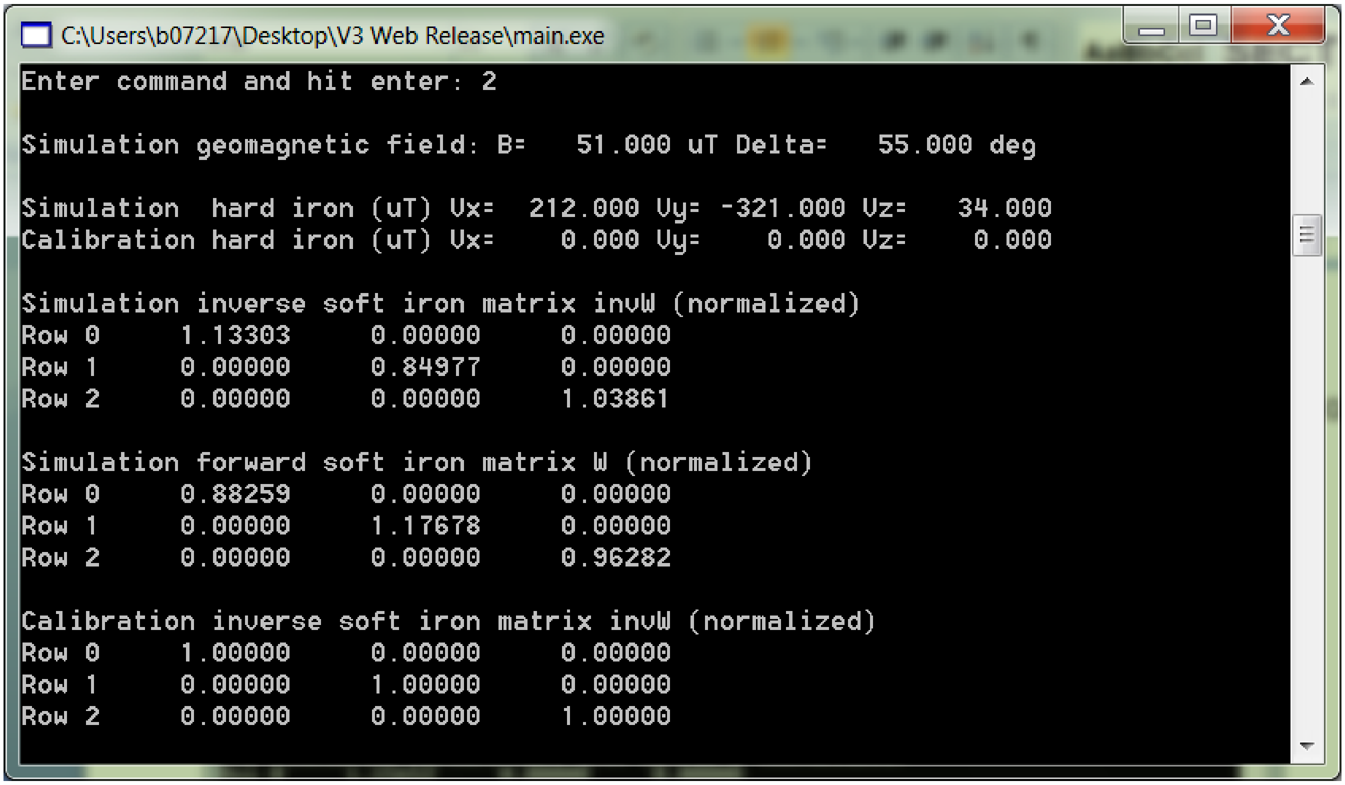


Figure 4. Option 2

The simulation geomagnetic field and declination angle shows the values of 51 μT and 55° just entered.

The simulation hard iron offset has the values 212.0 μT, -321.0 μT, 34.0 μT just entered but the calibration hard iron is still zero since the calibration algorithms have not yet run.

The simulation inverse and forward soft iron matrices are shown with the normalized values computed from the inverse soft iron matrix just entered. The calibration inverse soft iron matrix remains the identity matrix since the calibration algorithms have not yet run.

2.5 Option 3: Write a sensor simulation file to disc

Enter "3" to write a sensor test file to disc with magnetometer and accelerometer readings computed for random simulated eCompass orientation angles. The software prompts for the disc filename, the coordinate system (NED, Android or Windows 8) and the number of records to be written. In Figure 5 we have specified an output file called test1.csv to contain 45 simulated measurements computed for the Android coordinate system.

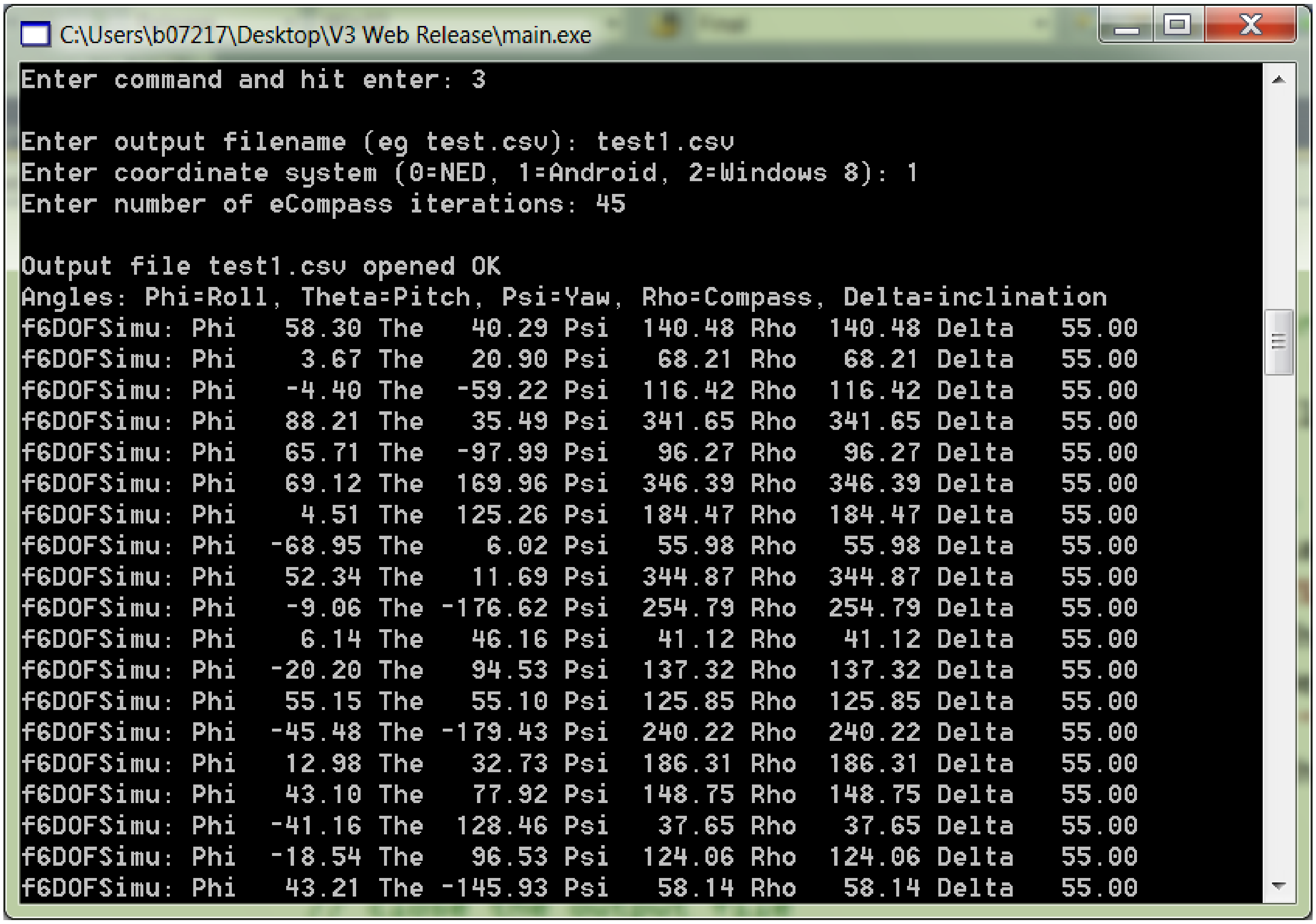


Figure 5. Option 3

The main use of this function is to provide a template file for formatting measured sensor data from real sensors that the user may wish to pass through the PC software before the code is ported to an embedded processor. The columns Gpx, Gpy and Gpz are the x, y and z accelerometer outputs (in units of g) aligned to the selected coordinate system (NED, Android or Windows 8). Similarly, Bpx, Bpy and Bpz are the x, y and z magnetometer outputs (in units of uT) aligned to the selected coordinate system.

The first five lines of the output file are listed below showing the comma separated integer values.

Gpx (g)	Gpy (g)	Gpz (g)	Bpx (μT)	Bpy (μT)	Bpz (μT)
0.851	–0.340	0.401	171.996	–336.606	15.462
0.064	–0.356	0.932	185.716	–292.295	1.791
–0.077	0.857	0.510	191.776	–372.983	23.251
1.000	–0.018	0.025	175.400	–287.216	41.280
0.912	0.407	–0.057	167.836	–309.614	35.801

2.6 Option 4: Run the eCompass with calibration (disc file)

Enter "4" to run the eCompass and calibration algorithms. The software allows the user to select:

- the source of the sensor data (from sensor simulation or from a disc file of measurements)
- the coordinate system to be used (NED, Android or Windows 8)
- the magnetic calibration model to be used
- the number of recorded or simulated measurements to be processed.

Enter "test1.csv" to use the measurements file created previously although normally this option will be used to process sensor data recorded from a separate platform. Then enter "1" to specify that the measurements should be processed according to the Android coordinate system. Enter "7" to run the 7 element calibration software and then enter "45" to process all 45 measurement records in the file test1.csv. The display of the first few processed records is shown in Figure 6.

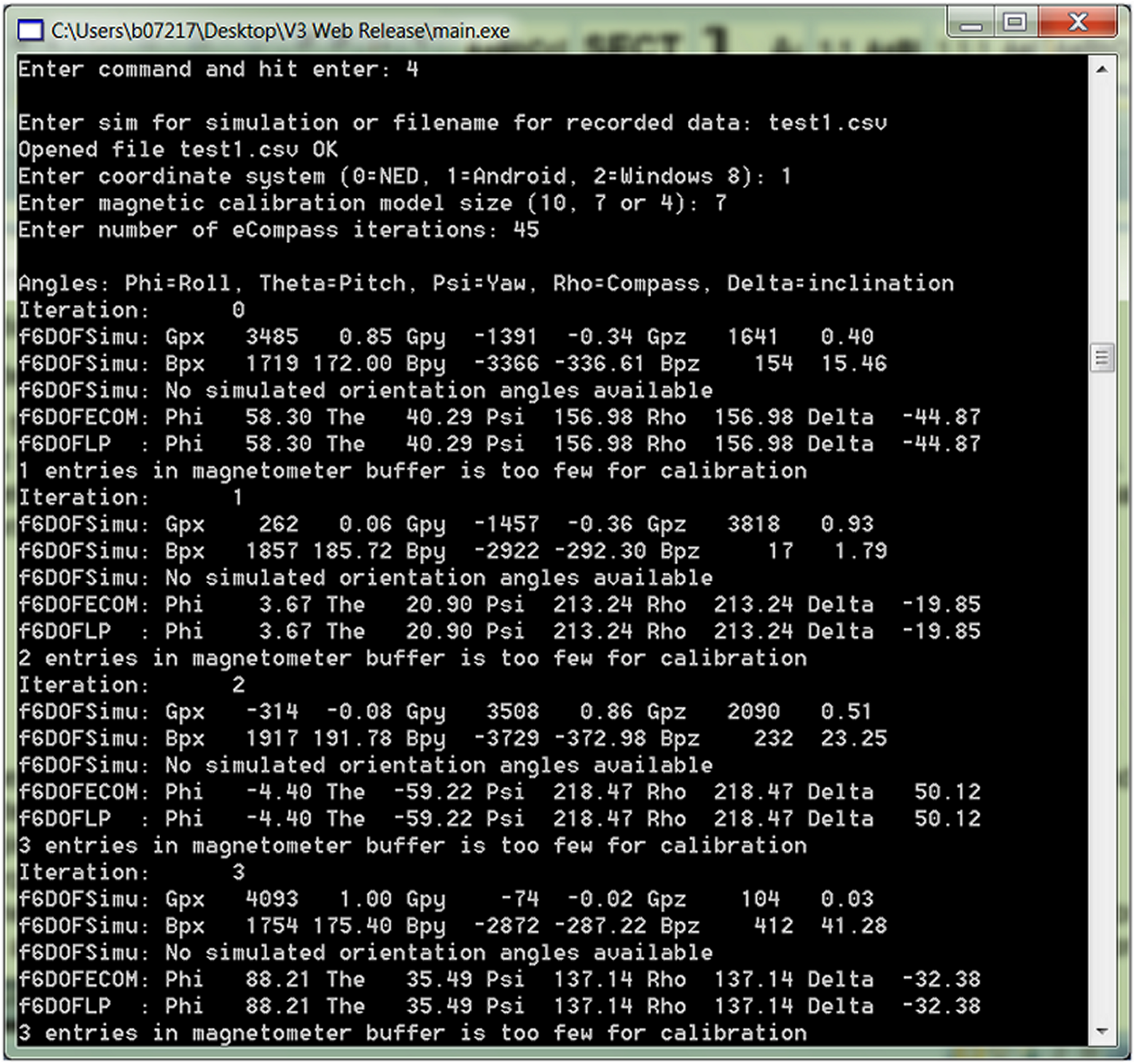

```
C:\Users\b07217\Desktop\V3 Web Release\main.exe
Enter command and hit enter: 4

Enter sim for simulation or filename for recorded data: test1.csv
Opened file test1.csv OK
Enter coordinate system (0=NED, 1=Android, 2=Windows 8): 1
Enter magnetic calibration model size (10, 7 or 4): 7
Enter number of eCompass iterations: 45

Angles: Phi=Roll, Theta=Pitch, Psi=Yaw, Rho=Compass, Delta=inclination
Iteration:      0
f6DOFSimu: Gpx   3485   0.85 Gpy  -1391  -0.34 Gpz   1641   0.40
f6DOFSimu: Bpx   1719 172.00 Bpy  -3366 -336.61 Bpz    154  15.46
f6DOFSimu: No simulated orientation angles available
f6DOFECOM: Phi   58.30 The   40.29 Psi  156.98 Rho  156.98 Delta  -44.87
f6DOFLP  : Phi   58.30 The   40.29 Psi  156.98 Rho  156.98 Delta  -44.87
1 entries in magnetometer buffer is too few for calibration
Iteration:      1
f6DOFSimu: Gpx    262   0.06 Gpy  -1457  -0.36 Gpz   3818   0.93
f6DOFSimu: Bpx   1857 185.72 Bpy  -2922 -292.30 Bpz     17   1.79
f6DOFSimu: No simulated orientation angles available
f6DOFECOM: Phi    3.67 The   20.90 Psi  213.24 Rho  213.24 Delta  -19.85
f6DOFLP  : Phi    3.67 The   20.90 Psi  213.24 Rho  213.24 Delta  -19.85
2 entries in magnetometer buffer is too few for calibration
Iteration:      2
f6DOFSimu: Gpx   -314  -0.08 Gpy   3508   0.86 Gpz   2090   0.51
f6DOFSimu: Bpx   1917 191.78 Bpy  -3729 -372.98 Bpz    232  23.25
f6DOFSimu: No simulated orientation angles available
f6DOFECOM: Phi   -4.40 The  -59.22 Psi  218.47 Rho  218.47 Delta   50.12
f6DOFLP  : Phi   -4.40 The  -59.22 Psi  218.47 Rho  218.47 Delta   50.12
3 entries in magnetometer buffer is too few for calibration
Iteration:      3
f6DOFSimu: Gpx   4093   1.00 Gpy    -74  -0.02 Gpz    104   0.03
f6DOFSimu: Bpx   1754 175.40 Bpy  -2872 -287.22 Bpz    412  41.28
f6DOFSimu: No simulated orientation angles available
f6DOFECOM: Phi   88.21 The   35.49 Psi  137.14 Rho  137.14 Delta  -32.38
f6DOFLP  : Phi   88.21 The   35.49 Psi  137.14 Rho  137.14 Delta  -32.38
3 entries in magnetometer buffer is too few for calibration
```

Figure 6. Processed records

For each record, the console display shows:

- f6DOFSIMU: the accelerometer reading read from file in counts and g
- f6DOFSIMU: the magnetometer reading read from file in counts and μT

- f6DOFSIMU: No simulated orientation angles available. In this case, this information is not available since a file of real world accelerometer and magnetometer sensor measurements will not contain this information
- f6DOFECOM: the computed Euler angles (roll Phi, pitch Theta and yaw Psi) plus the compass heading angle (Rho) and computed magnetic inclination angle (Delta)
- f6DOFLP: the low pass filtered computed Euler angles (roll Phi, pitch Theta and yaw Psi) plus the compass heading angle (Rho) and computed magnetic inclination angle (Delta). Since the simulation sensor driver generates sensor readings for random orientation angles, the low pass filter defaults to all pass.

Initially there are too few measurements read from file to compute an accurate magnetic calibration. The yaw angle (Psi), compass angle (Rho) and magnetic inclination angle (delta) will therefore be in error and this immediately apparent from the various inclination angles which differ markedly from the value of 55° previously entered for the simulation.

After a small number of iterations, here 29 but the precise number will depend on the exact random orientation angles generated in the simulation, the minimum number 24 of magnetometer measurements will be present in the magnetometer buffer. The 7 element calibration algorithm then executes showing:

- A first estimate of the hard iron offset of 211.950 μT, -320.923 and 33.991 μT, very close to the simulation values entered earlier of 212 μT, -321 μT and 34 μT
- A first estimate of the geomagnetic field strength to be 50.995 μT, very close to the simulation value of 51 μT entered earlier
- A normalized computed inverse soft iron matrix with values very close to the normalized version of the inverse soft iron matrix values entered earlier. For convenience of checking the result, the original normalized simulation inverse soft iron matrix is displayed after the inverse soft iron matrix computed by the calibration algorithms.

The first record after the calibration run, Figure 7,shows an estimated magnetic inclination angle of 54.92°, very close to the value of 55° used for simulation.

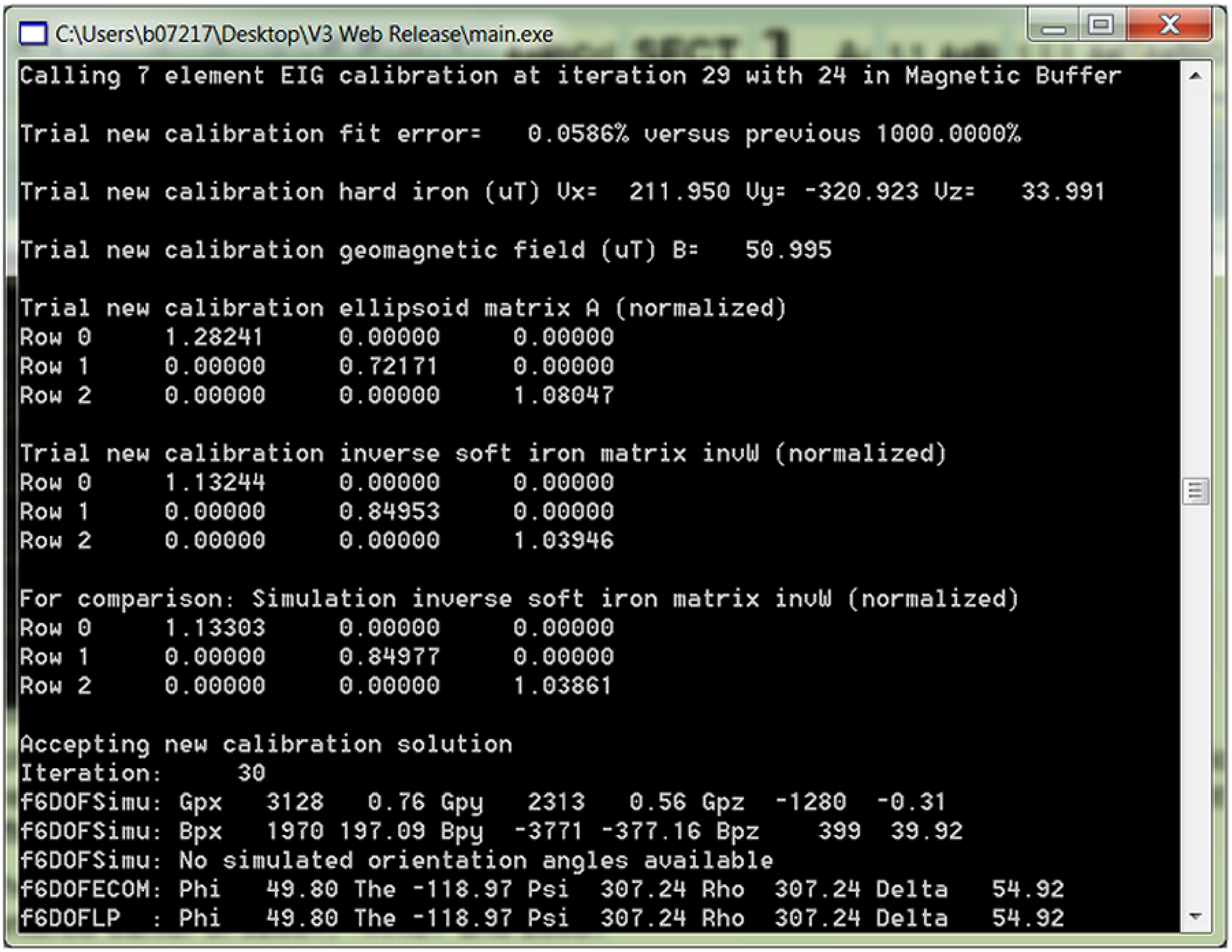
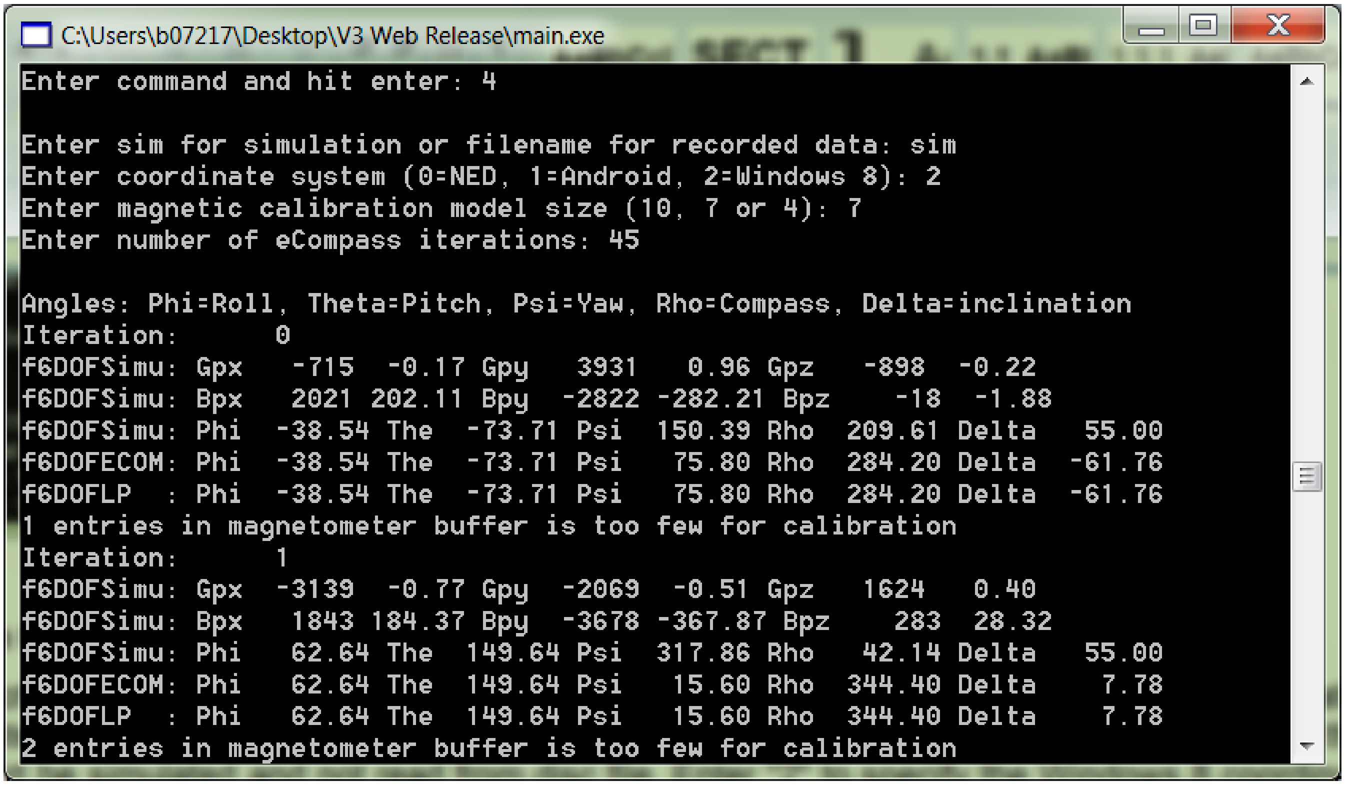


Figure 7. Processed records

2.7 Option 4: Run the eCompass with calibration (simulated data)

This demonstration software typically operates on simulated sensor data generated by the sensor simulated driver in real time. Enter "4" to run the eCompass and calibration algorithms but now enter "sim" to denote that the sensor data should be simulated and not read from disc file. Enter "2" to specify the Windows 8 coordinate system, "7" for the 7 element calibration model and "45" for 45 iterations of the eCompass.

The output from the first two iterations is shown in Figure 8. As in the previous example, initially there are too few magnetometer measurements for the magnetic calibration algorithms to run and the computed compass heading (Rho) and the magnetic inclination angle (Delta) differ from those used in the simulation. The roll (Phi) and pitch (Theta) angles are correct since these are derived from the accelerometer reading. The low pass filtered Euler angles match the instantaneous (and random) Euler angles used in the simulation since the filter defaults to all pass in the software.

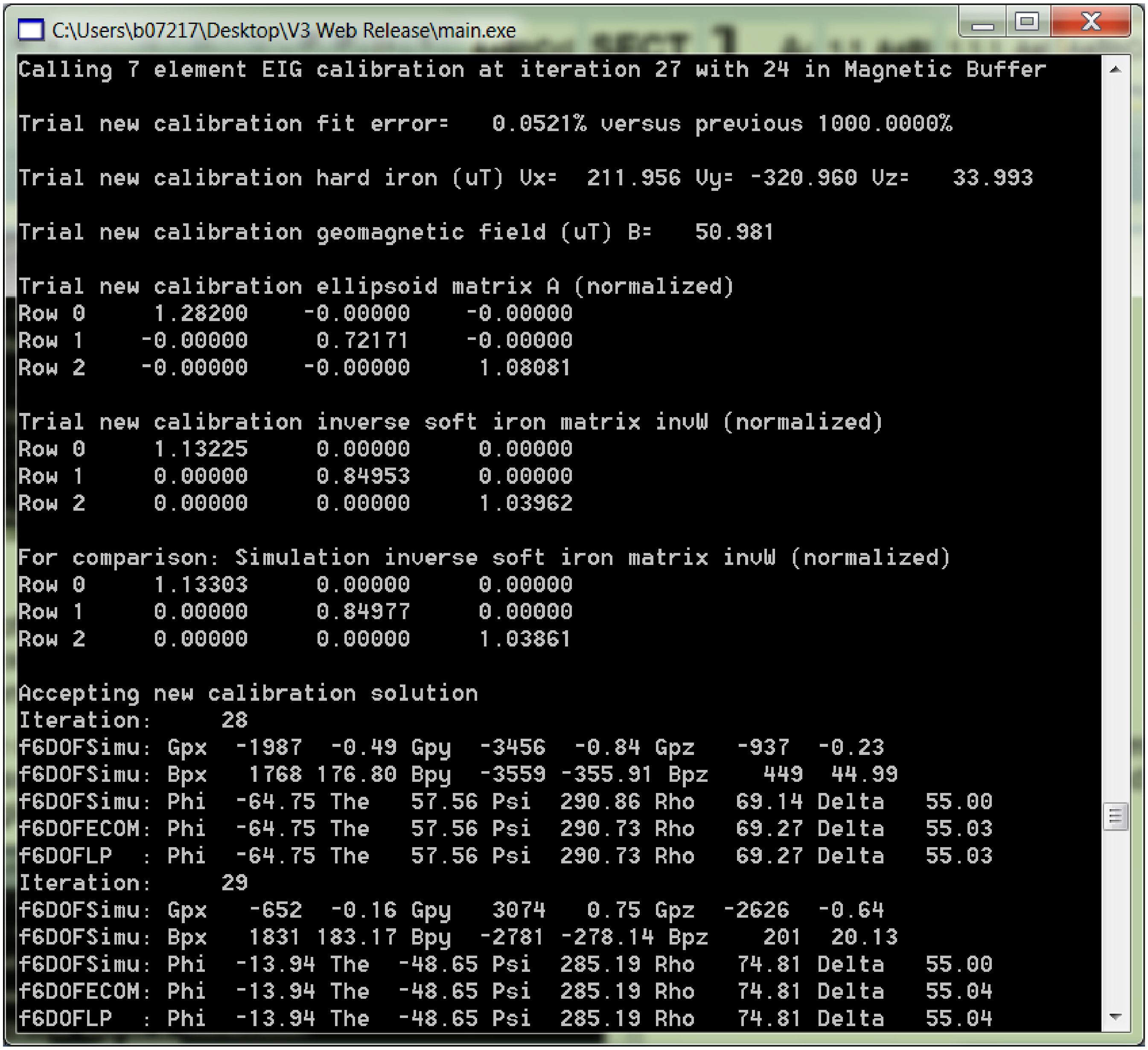


```
C:\Users\b07217\Desktop\V3 Web Release\main.exe
Enter command and hit enter: 4

Enter sim for simulation or filename for recorded data: sim
Enter coordinate system (0=NED, 1=Android, 2=Windows 8): 2
Enter magnetic calibration model size (10, 7 or 4): 7
Enter number of eCompass iterations: 45

Angles: Phi=Roll, Theta=Pitch, Psi=Yaw, Rho=Compass, Delta=inclination
Iteration:      0
f6DOFSimu: Gpx   -715  -0.17 Gpy   3931   0.96 Gpz   -898  -0.22
f6DOFSimu: Bpx   2021 202.11 Bpy  -2822 -282.21 Bpz    -18  -1.88
f6DOFSimu: Phi  -38.54 The  -73.71 Psi  150.39 Rho  209.61 Delta   55.00
f6DOFECOM: Phi  -38.54 The  -73.71 Psi   75.80 Rho  284.20 Delta  -61.76
f6DOFLP  : Phi  -38.54 The  -73.71 Psi   75.80 Rho  284.20 Delta  -61.76
1 entries in magnetometer buffer is too few for calibration
Iteration:      1
f6DOFSimu: Gpx  -3139  -0.77 Gpy  -2069  -0.51 Gpz   1624   0.40
f6DOFSimu: Bpx   1843 184.37 Bpy  -3678 -367.87 Bpz    283  28.32
f6DOFSimu: Phi   62.64 The  149.64 Psi  317.86 Rho   42.14 Delta   55.00
f6DOFECOM: Phi   62.64 The  149.64 Psi   15.60 Rho  344.40 Delta    7.78
f6DOFLP  : Phi   62.64 The  149.64 Psi   15.60 Rho  344.40 Delta    7.78
2 entries in magnetometer buffer is too few for calibration
```

Figure 8. Processed records

After a small number of iterations, 27 in Figure 9 but the precise number will vary from run to run, the 7 element calibration algorithm executes and again produces excellent estimates of i) the hard iron offset ii) the geomagnetic field strength and iii) the inverse soft iron matrix which match the values entered earlier.

The readings after the calibration algorithms execute for the first time now show that all the orientation angles including yaw, compass heading and inclination angle are computed correctly by the eCompass function and match the random angles selected for the sensor simulation angles.



```
C:\Users\b07217\Desktop\V3 Web Release\main.exe
Calling 7 element EIG calibration at iteration 27 with 24 in Magnetic Buffer

Trial new calibration fit error=   0.0521% versus previous 1000.0000%

Trial new calibration hard iron (uT) Vx=  211.956 Vy= -320.960 Vz=   33.993

Trial new calibration geomagnetic field (uT) B=   50.981

Trial new calibration ellipsoid matrix A (normalized)
Row 0     1.28200    -0.00000    -0.00000
Row 1    -0.00000     0.72171    -0.00000
Row 2    -0.00000    -0.00000     1.08081

Trial new calibration inverse soft iron matrix invW (normalized)
Row 0     1.13225     0.00000     0.00000
Row 1     0.00000     0.84953     0.00000
Row 2     0.00000     0.00000     1.03962

For comparison: Simulation inverse soft iron matrix invW (normalized)
Row 0     1.13303     0.00000     0.00000
Row 1     0.00000     0.84977     0.00000
Row 2     0.00000     0.00000     1.03861

Accepting new calibration solution
Iteration:      28
f6DOFSimu: Gpx  -1987  -0.49 Gpy  -3456  -0.84 Gpz   -937  -0.23
f6DOFSimu: Bpx   1768 176.80 Bpy  -3559 -355.91 Bpz    449  44.99
f6DOFSimu: Phi  -64.75 The   57.56 Psi  290.86 Rho   69.14 Delta   55.00
f6DOFECOM: Phi  -64.75 The   57.56 Psi  290.73 Rho   69.27 Delta   55.03
f6DOFLP  : Phi  -64.75 The   57.56 Psi  290.73 Rho   69.27 Delta   55.03
Iteration:      29
f6DOFSimu: Gpx   -652  -0.16 Gpy   3074   0.75 Gpz  -2626  -0.64
f6DOFSimu: Bpx   1831 183.17 Bpy  -2781 -278.14 Bpz    201  20.13
f6DOFSimu: Phi  -13.94 The  -48.65 Psi  285.19 Rho   74.81 Delta   55.00
f6DOFECOM: Phi  -13.94 The  -48.65 Psi  285.19 Rho   74.81 Delta   55.04
f6DOFLP  : Phi  -13.94 The  -48.65 Psi  285.19 Rho   74.81 Delta   55.04
```

Figure 9. Processed records

2.8 Option 99: Quit

Enter "99" to quit the simulation. At this point there should be good confidence that the algorithms are executing correctly with a full end to end simulation of sensors, magnetic calibration and the eCompass functions. The next steps are to i) build the same application on the target embedded system and then ii) to replace the sensor simulation functions with the drivers for the physical accelerometer and magnetometer sensors being used.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

Document Number: MAGCALSWUG
Rev. 1.0, 4/2013