

Using the Xtrinsic FXOS8700CQ Auto-Calibration Function

by: Talat Ozyagcilar
Applications Engineer

1 Introduction

This application note demonstrates the auto-calibration function of the FXOS8700CQ.

Auto calibration is intended for mobile platforms with limited processing power and memory. It provides automatic hard-iron calibration and compensation using the min/max averaging method to calculate a hard-iron estimate.

This application note makes references to the FXOS8700CQ data sheet and Freescale application notes AN4246, AN4248 and AN4249.

1.1 Keywords

Hard-iron estimate, Auto calibration, Minimum, Maximum, Magnetic field, eCompass.

Contents

1	Introduction	1
1.1	Keywords	1
1.2	Related Documentation	2
2	Auto-Calibration Function	2
3	Using Auto-Calibration to Calculate Compass Angle ...	4
3.1	Calibrate-then-Compensate	4
3.2	Calibrate-and-Compensate	5
4	Investigation of Program Output	8
5	Effects of Avoiding Soft-Iron and Tilt Compensation ...	11

1.2 Related Documentation

The FXOS8700CQ device features and operations are described in a variety of reference manuals, user guides, and application notes. To find the most-current versions of these documents:

1. Go to the Freescale homepage at:

<http://www.freescale.com/>
2. In the Keyword search box at the top of the page, enter the device number FXOS8700CQ.
3. In the Refine Your Result pane on the left, click on the Documentation link.

2 Auto-Calibration Function

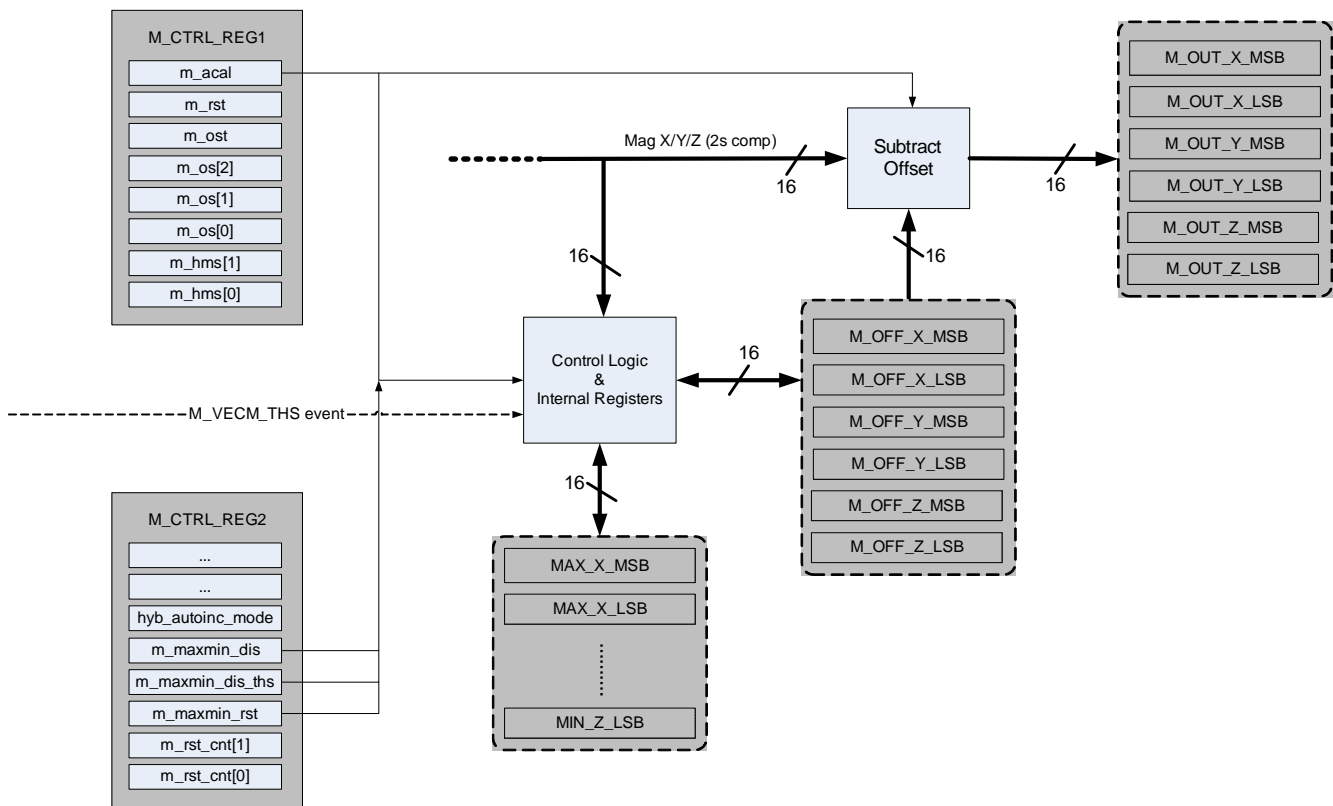


Figure 1. Auto-calibration function block diagram

On target platforms with limited processing power (wrist watches for example) it may not be possible to run a compensation engine for an eCompass as depicted in Freescale application notes AN4246 and AN4248. For such platforms, FXOS8700CQ provides an auto-calibration function that calculates the hard-iron estimate (calibration) and automatically subtracts it (compensation) from the current magnetic-field readings. This way FXOS8700CQ provides magnetic X/Y/Z data with hard-iron compensation already applied to it. For platforms where hard-iron interference is the primary source of error, and soft-iron distortion is minimal or non-existent, the auto-calibration function can serve as the magnetic-calibration algorithm for the system.

The hard-iron estimate used by this function is a simple average as shown in [Equation 1](#).

$$\mathbf{V} = \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} = \frac{1}{2} \begin{pmatrix} m_{x_out_max} + m_{x_out_min} \\ m_{y_out_max} + m_{y_out_min} \\ m_{z_out_max} + m_{z_out_min} \end{pmatrix} \quad \text{Eqn. 1}$$

where the $m_{x/y/z_out_max}$ and $m_{x/y/z_out_min}$ values correspond to the maximum and minimum magnetic field readings measured on the corresponding axis. It is possible to observe these values in registers MAX_X_MSB (0x45)...MIN_Z_LSB (0x50) with a resolution of 0.1 $\mu\text{T}/\text{LSB}$.

The model for expressing the magnetic field experienced by a mobile platform under arbitrary rotation is depicted in [Equation 2](#) (see AN4246).

$$\mathbf{B}_p = \mathbf{W}\mathbf{R}_x(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(\psi)\mathbf{B} \begin{pmatrix} \cos \delta \\ 0 \\ \sin \delta \end{pmatrix} + \mathbf{V} \quad \text{Eqn. 2}$$

Where $\mathbf{B} \begin{pmatrix} \cos \delta \\ 0 \\ \sin \delta \end{pmatrix}$ is the reference magnetic-field vector (x-component aligned to magnetic north), $\mathbf{R}_x(\phi)$,

$\mathbf{R}_y(\theta)$ and $\mathbf{R}_z(\psi)$ are the rotation matrices around each axis, \mathbf{W} is a 3 x 3 matrix representing soft-iron interference and \mathbf{V} is a 3 x 1 matrix representing hard-iron interference.

The magnetic field reported by FXOS8700CQ can be represented using [Equation 3](#), provided that the device and the platform share the same reference frame.

$$\mathbf{B}_{dev} = \mathbf{B}_p - \mathbf{V} = \mathbf{W}\mathbf{R}_x(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(\psi)\mathbf{B} \begin{pmatrix} \cos \delta \\ 0 \\ \sin \delta \end{pmatrix} \quad \text{Eqn. 3}$$

If the soft-iron interference is negligible [Equation 3](#) simplifies to [Equation 4](#). In this case, the host processor on the target platform will need to be utilized for applying the tilt-compensation ([Equation 6](#)) and finally calculating the compass angle ([Equation 7](#)). For further details, please see application note AN4246.

$$\mathbf{B}_{dev} = \mathbf{R}_x(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(\psi)\mathbf{B} \begin{pmatrix} \cos \delta \\ 0 \\ \sin \delta \end{pmatrix} \quad \text{Eqn. 4}$$

$$\mathbf{R}_y(-\theta)\mathbf{R}_x(-\phi)\mathbf{B}_{dev} = \mathbf{R}_z(\psi)\mathbf{B} \begin{pmatrix} \cos \delta \\ 0 \\ \sin \delta \end{pmatrix} = \begin{pmatrix} \mathbf{B}_{fx} \\ \mathbf{B}_{fy} \\ \mathbf{B}_{fz} \end{pmatrix} \quad \text{Eqn. 5}$$

It is possible to avoid the need to perform tilt compensation if the end-user is required to hold the mobile platform horizontally, similar to using a conventional analog compass. Equation 4 then simplifies to Equation 7.

$$\mathbf{B}_{dev} = \mathbf{R}_z(\psi)\mathbf{B} \begin{pmatrix} \cos \delta \\ 0 \\ \sin \delta \end{pmatrix} = \begin{pmatrix} \mathbf{B}_{fx} \\ \mathbf{B}_{fy} \\ \mathbf{B}_{fz} \end{pmatrix} \quad \text{Eqn. 6}$$

$$\psi = \tan^{-1} \left(\frac{-\mathbf{B}_{fy}}{\mathbf{B}_{fx}} \right) \quad \text{Eqn. 7}$$

3 Using Auto-Calibration to Calculate Compass Angle

The compensation can be accomplished using one of the following two methods:

- Calibrate-then-Compensate
- Calibrate-and-Compensate

3.1 Calibrate-then-Compensate

- is robust against unexpected hard-iron interference
- will burden the end-user with rotating the final platform in a “figure of 8” whenever calibration is no longer valid. Invalid calibration can be detected by a persistent (longer than period specified by *autoCalResetCtr* in Example 1) jamming condition. Ideally, when the end-user moves far enough away from the source of magnetic interference, the jamming condition should go away. However, exposure to magnetic fields greater than 10000 μT may permanently shift the offset⁽¹⁾. In this case, recalibration is required and the offset shift will be absorbed in the new hard-iron estimation.

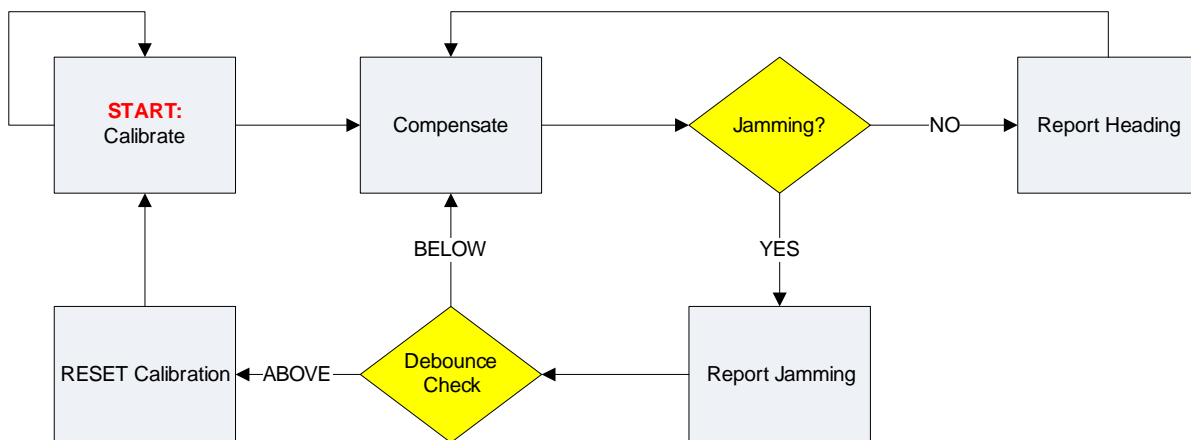


Figure 2. Calibrate-then-Compensate

1. “Maximum exposed magnetic field without perming” as expressed in the FXOS8700CQ data sheet. If the applied magnetic field goes above 0.1T, the part may be permanently damaged.

3.2 Calibrate-and-Compensate

- will spare the end-user from having to explicitly perform calibration, but will be more susceptible to external (external to the PCB) magnetic interference contaminating the hard-iron estimate.
- is recommended when time-varying magnetic interference from a component on the PCB is expected (i.e. interference due to voice coil currents in a speaker on the PCB).

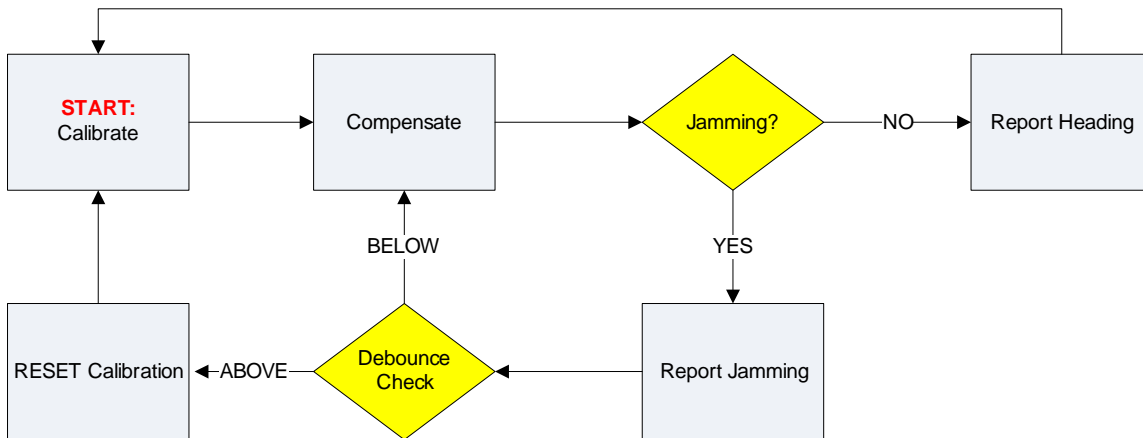


Figure 3. Calibrate-and-Compensate

The two methods mentioned above are by no means comprehensive. Naturally, the method of calibration/compensation should be dictated by the requirements of the final application; software flows implemented by the industry today primarily follow one of the two methods above or a mix of the two.

Example 1 implements Calibrate-and-Compensate approach as shown in [Figure 2](#) since it is simpler to implement; no interaction with the end-user is necessary. However, implementing Calibrate-then-Compensate method does not require a change in setting up of any device registers except, once the calibration period is over, the updating of MIN/MAX registers (0x45...0x50) must be suspended via M_CTRL_REG2 (0x5C), bit *m_maxmin_dis* (bit 4).

In [Example 1](#), observe the following key points:

1. Choosing of the *magThreshold*: This is the threshold that will be used to detect magnetic jamming. In this example, 100 μ T is used as the detection threshold.
2. Choosing of the *autoCalResetCtr* threshold: This controls how many ODR cycles the jamming must persist for before the calibration is reset.
 - a) A short time period may throw away good calibration if the interference disappears within a reasonable time and risk new calibration to be contaminated by varying (unsettled) hard-iron interference (Ex. end-user walking past a source of magnetism).
 - b) A long time period may fail to accommodate persistent PCB hard-iron interference and inconvenience the end-user with a prolonged jamming indication.

3. Disabling the updating of the MIN/MAX registers via M_CTRL_REG2, bit *m_maxmin_dis_ths* once a magnetic-vector magnitude event (jamming) is detected. This will prevent magnetic interference from contaminating the hard-iron estimate; hard-iron estimate will not be updated once jamming is detected.

Example 1 makes the assumption that soft-iron interference is negligible and PCB orientation is near horizontal when being used, similar to an analog compass. Therefore the output from FXOS8700CQ equals B_{dev} in [Equation 6](#) and the compass angle is computed using [Equation 7](#).

Example 1. Python Code Snippet for Auto-Calibration function

```
# Initialize debounce counter for reset

autoCalResetCtr = 0

# Issue soft reset

write_byte( [0x2B, 0x40] )
time.sleep(0.1)

# Set threshold. 1000 counts = 100.0uT

magThreshold = 1000 # counts
magThresholdHi = (magThreshold & 0xFF00) >> 8
magThresholdLo = magThreshold & 0xFF

write_byte( [0x6A, 0x80 | magThresholdHi] )
write_byte( [0x6B, magThresholdLo] )

# M_VECM_CNT = 1 * 20ms = 20ms
# ! - steps double in hybrid mode

write_byte( [0x6C, 0x01] )

# M_VECM_CFG
# m_vecm_ele = 1 => event latching enabled
# m_vecm_initm = 1 => use M_VECM_INITX/Y/Z as initial reference
# m_vecm_updm = 1 => do not update initial reference
# m_vecm_en = 1 => enable magnetometer vector magnitude detection feature

write_byte( [0x69, 0x7B] )

# enable interrupts for DRDY using CTRL_REG4

write_byte( [0x2D, 0x01] )

# route interrupts to INT1 pin using CTRL_REG5

write_byte( [0x2E, 0x01] )

# Setup device
# via M_CTRL_REG1 (0x5B): Hybrid mode, OS = 32, Auto Cal
# via M_CTRL_REG2 (0x5C): Hybrid auto increment, maxmin disable
```

```

# threshold
# via CTRL_REG1 (0x2A): ODR = 50Hz, ACTIVE mode

write_byte( [0x5B, 0x9F] )
write_byte( [0x5C, 0x20] )
write_byte( [0x2A, 0x31] )

# Wait for INT1 to assert and clear interrupt by reading register INT_SOURCE (0x0C)
# or reading sensor data (0x01...0x06, 0x33...0x38) depending on the function that
# generated the interrupt

while( True ):

    # Check INT1 pin for interrupts

    transition = aa_gpio_change( handle, 100 )
    if (transition & INT1_PIN ) == INT1_PIN:
        # print "No interrupt..."
        continue

    print "\nODR cycle =====>\n"

    # Check if interrupt is due to Magnetometer Vector Magnitude

    (count, dataIn) = read_dev( 0x5E, 1 )
    if dataIn[0] & 0x02 == 0x02:

        print "Interrupt due to Magnetometer Vector Magnitude feature"
        print "Magnetic Jamming detected"

        # Start debounce ctr

        autoCalResetCtr = autoCalResetCtr + 1

    else:

        autoCalResetCtr = 0 # reset counter

    # Check if interrupt is due to Data Ready (DRDY)

    (count, dataIn) = read_dev( 0x0C, 1 )
    if dataIn[0] & 0x01 == 0x01:

        print "Interrupt due to data ready"

        (magX, magY, magZ, accX, accY, accZ) = dev_display_sens_data()

        # Apply HAL to match device coordinate system with NED
        # X matches, Z irrelevant

        magY = -magY

        print "Magnetic Vector Magnitude = %4.1f uT" % (math.sqrt(

```

```

    magX * magX + magY * magY + magZ * magZ ) / 10)
print "Heading ==> %.2f degrees from NORTH" % (math.atan2(
    -magY, magX ) * (180 / math.pi))

# Reset Auto Cal based on debounce count

if autoCalResetCtr > 10:

    print "Resetting Hard Iron Estimation..."

    write_dev( 0x5C, [0x24] )
    autoCalResetCtr = 0 # reset counter

```

4 Investigation of Program Output

Figure 4 is a snapshot of program output shortly after startup.

```

X MAG: -4, Y MAG: -6, Z MAG: 1, X ACC: -265, Y ACC: -411, Z ACC: 4098
Magnetometer Vector Magnitude = 0.7 uT
Heading ==> -123.69 degrees from NORTH

ODR cycle ==>

Interrupt due to data ready
X MAG: 0, Y MAG: -7, Z MAG: 0, X ACC: -267, Y ACC: -408, Z ACC: 4101
Magnetometer Vector Magnitude = 0.7 uT
Heading ==> -90.00 degrees from NORTH

ODR cycle ==>

Interrupt due to data ready
X MAG: 1, Y MAG: -6, Z MAG: 0, X ACC: -266, Y ACC: -405, Z ACC: 4102
Magnetometer Vector Magnitude = 0.6 uT
Heading ==> -80.54 degrees from NORTH

ODR cycle ==>

Interrupt due to data ready
X MAG: 1, Y MAG: -4, Z MAG: -1, X ACC: -261, Y ACC: -408, Z ACC: 4097
Magnetometer Vector Magnitude = 0.4 uT
Heading ==> -75.96 degrees from NORTH

```

Figure 4. Sample code output upon startup

Upon startup, the magnetic-vector magnitude is near zero because auto calibration has only the initial magnetic data to go by and this initial data becomes the hard-iron estimate per [Equation 1](#). The compass angle is unstable since it is mainly dominated by noise.

As the PCB is rotated, the hard-iron estimate gets better and the vector magnitude gradually approaches the earth's magnetic-field vector magnitude and the compass angle becomes more accurate, as seen in [Figure 5](#).

```

X MAG: -102, Y MAG: 290, Z MAG: -262, X ACC: 724, Y ACC: -496, Z ACC: 4045
Magnetometer Vector Magnitude = 40.4 uT
Heading ==> 109.38 degrees from NORTH

ODR cycle ==>

Interrupt due to data ready
X MAG: -102, Y MAG: 291, Z MAG: -260, X ACC: 718, Y ACC: -506, Z ACC: 4039
Magnetometer Vector Magnitude = 40.3 uT
Heading ==> 109.32 degrees from NORTH

ODR cycle ==>

Interrupt due to data ready
X MAG: -103, Y MAG: 293, Z MAG: -260, X ACC: 722, Y ACC: -506, Z ACC: 4046
Magnetometer Vector Magnitude = 40.5 uT
Heading ==> 109.37 degrees from NORTH

ODR cycle ==>

Interrupt due to data ready
X MAG: -105, Y MAG: 292, Z MAG: -259, X ACC: 741, Y ACC: -505, Z ACC: 4033
Magnetometer Vector Magnitude = 40.4 uT
Heading ==> 109.78 degrees from NORTH

```

Figure 5. Sample code output upon startup

In [Figure 6](#) jamming is detected when hard-iron interference is introduced by placing a permanent magnet on the PCB (simulating a speaker on the PCB turning on). As the interference persists for longer than the debounce threshold, the calibration is reset and the interference is absorbed into the new hard-iron estimate, bringing the magnetic-vector magnitude back to near zero. As long as the permanent magnet orientation stays fixed with respect to the PCB, compensation for the interference introduced will occur.

```

Heading ==> 40.00 degrees from NORTH

ODR cycle =====>

Interrupt due to Magnetometer Vector Magnitude feature
Magnetic Jamming detected
Interrupt due to data ready
X MAG: 383, Y MAG: 324, Z MAG: -3065, X ACC: 61, Y ACC: -415, Z ACC: 4107
Magnetometer Vector Magnitude = 310.6 uT
Heading ==> 40.23 degrees from NORTH

ODR cycle =====>

Interrupt due to Magnetometer Vector Magnitude feature
Magnetic Jamming detected
Interrupt due to data ready
X MAG: 399, Y MAG: 350, Z MAG: -3072, X ACC: 64, Y ACC: -412, Z ACC: 4113
Magnetometer Vector Magnitude = 311.8 uT
Heading ==> 41.26 degrees from NORTH

ODR cycle =====>

Interrupt due to Magnetometer Vector Magnitude feature
Magnetic Jamming detected
Interrupt due to data ready
X MAG: 410, Y MAG: 368, Z MAG: -3082, X ACC: 66, Y ACC: -417, Z ACC: 4106
Magnetometer Vector Magnitude = 313.1 uT
Heading ==> 41.91 degrees from NORTH

ODR cycle =====>

Interrupt due to Magnetometer Vector Magnitude feature
Magnetic Jamming detected
Interrupt due to data ready
X MAG: 421, Y MAG: 352, Z MAG: -3084, X ACC: 66, Y ACC: -415, Z ACC: 4114
Magnetometer Vector Magnitude = 313.2 uT
Heading ==> 39.90 degrees from NORTH
Resetting Hard Iron Estimation... ←

ODR cycle =====>

Interrupt due to data ready
X MAG: 1, Y MAG: 1, Z MAG: 0, X ACC: 71, Y ACC: -411, Z ACC: 4111
Magnetometer Vector Magnitude = 0.1 uT
Heading ==> 45.00 degrees from NORTH

ODR cycle =====>

Interrupt due to data ready
X MAG: 8, Y MAG: 14, Z MAG: -1, X ACC: 75, Y ACC: -414, Z ACC: 4113
Magnetometer Vector Magnitude = 1.6 uT
Heading ==> 60.26 degrees from NORTH

ODR cycle =====>

Interrupt due to data ready
X MAG: 8, Y MAG: -1, Z MAG: 0, X ACC: 77, Y ACC: -416, Z ACC: 4105
Magnetometer Vector Magnitude = 0.8 uT
Heading ==> -7.13 degrees from NORTH

```

Figure 6. Sample code output during and after the introduction of hard-iron interference

5 Effects of Avoiding Soft-Iron and Tilt Compensation

It is important to understand the effects of avoiding soft-iron and tilt compensation since both can significantly affect the compass heading angle error if not kept to a minimum.

In Freescale application note AN4249, Section 7 discusses the induced errors in compass heading angle due to an imperfect soft-iron calibration. This is analogous to errors in compass angle due to avoiding soft-iron compensation altogether. A one-time investigation of the soft-iron characteristics of the final platform can provide insight to the possible compass angle errors induced by it.

Section 5 of AN4249 discusses the induced errors in compass heading angle due to errors in the calculated roll and pitch angles. This is analogous to errors in compass angle due to avoiding tilt compensation altogether. Figure 5 in AN4249 shows the error in compass angle up to $\Delta\phi_{\tan\delta} = \Delta\theta_{\tan\delta} = 5$ degrees of error in tilt compensation, as an example. Equation 22 in AN4249 can provide more insights on how tilt-compensation errors contribute to compass heading angle errors.

How to Reach Us:

Home Page:
www.freescale.com

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

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: <http://www.reg.net/v2/webservices/Freescale/Docs/TermsandConditions.htm>.

Freescale and the Freescale logo are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Xtrinsic is a trademark of Freescale Semiconductor, Inc.

All other product or service names are the property of their respective owners.

© 2012 Freescale Semiconductor, Inc. All rights reserved.