Xtrinic Sensor Fusion: Alignment to Sensor Framework of Win8, Android™ and Others

AMF-ENT-T0747

Mark Pedley
Systems and Algorithm Engineering
Sensor Solutions Division
Tempe, Arizona

October 2013

Contents

- Schematics
- Physics
- Accelerometer, Magnetometer and Gyroscope Sensors
- New Freescale Sensors
- Sensor Fusion Mathematics
- Software Drivers
- Hands On Demonstrations
- Aerospace (iPhone), Android and Windows 8 Standards
- Application Notes and C Source Code

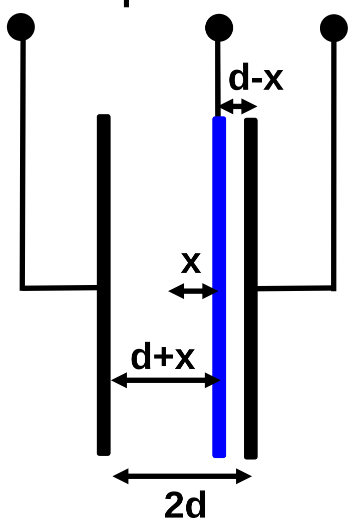
freescale™

Accelerometer

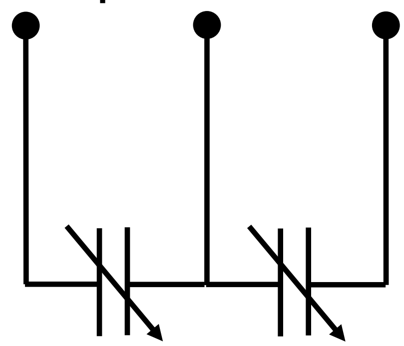
Accelerometer Sensors: Electrical Model

- Accelerometer are single package, two die, devices:
  - g-cell: Surface micromachined (MEMS) capacitive sensing cells modeled as a set of beams attached to a central mass that moves between fixed beams.
  - ASIC: Performs capacitance to voltage (C to V), internal calibration, temperature compensation and, on digital devices, ADC conversion

Simplified Model



Equivalent Circuit

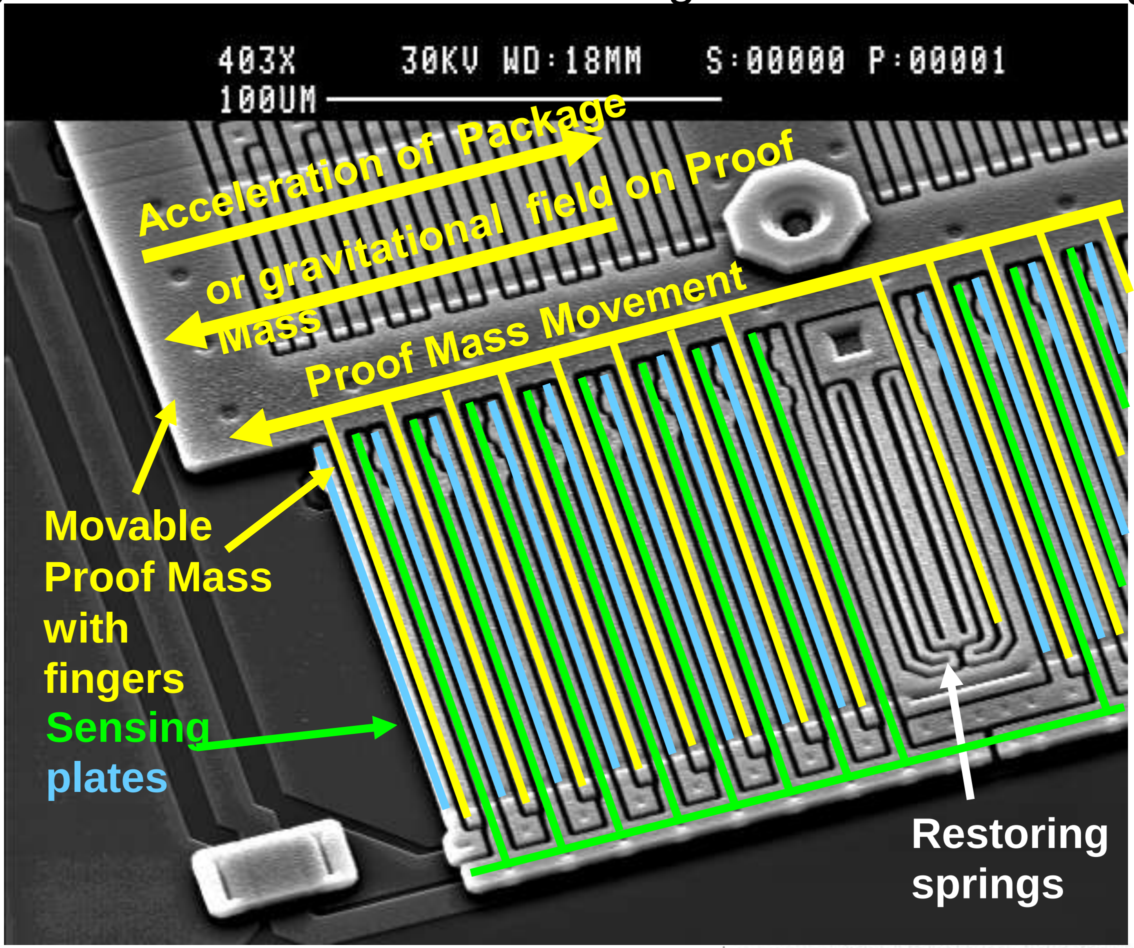


$$\Delta C = \left(\frac{\varepsilon_0 A}{d - x}\right) - \left(\frac{\varepsilon_0 A}{d + x}\right) \approx \frac{2\varepsilon_0 A x}{d^2}$$

- Capacitance change ΔC is proportional to displacement x of the MEMS structure which is proportional to applied force from acceleration and gravity

Accelerometer Sensors: MEMS

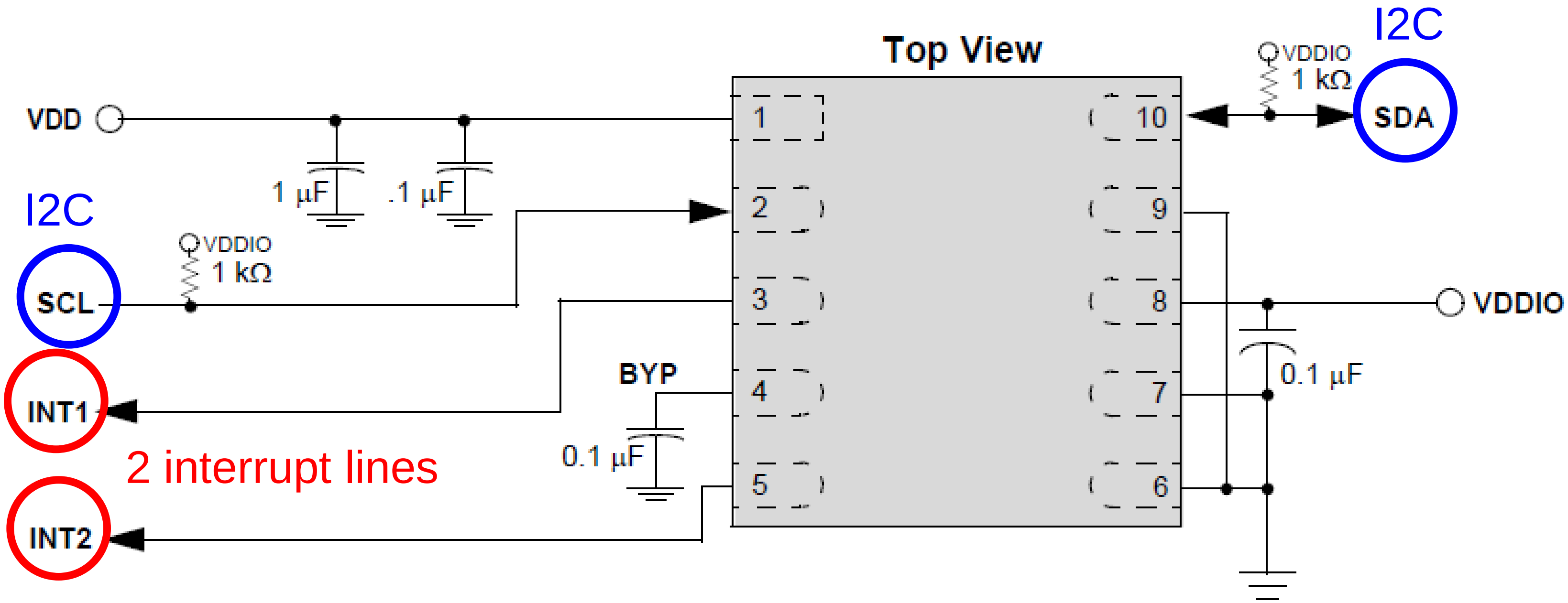
- Electron Micrograph of one axis of a 3-axis MEMS accelerometer
- By inspection, it's clear that the proof mass moves in response to both package acceleration and when being held in the earth's gravity



MMA8652 Accelerometer: Schematic

- Typical three-axis consumer MEMS accelerometer
- Range up to 8g with 12 bit output
- Digital output read over I2C serial bus from host uC

10-PIN DFN
2 mm x 2 mm x 1 mm



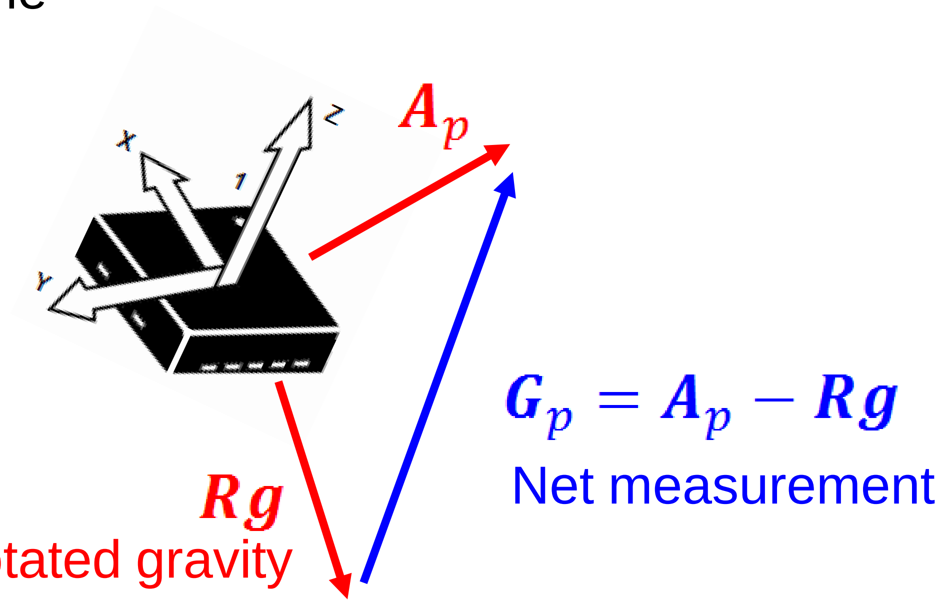
Accelerometer Sensors: Physics 101

- All accelerometers are sensitive to both linear acceleration **and** gravity. The accelerometer beams deflect when the package is accelerated or if held in the earth's gravitational field
- An accelerometer **on its own** cannot distinguish between gravity and acceleration (but it can combined with a gyro sensor)
- This is actually a fundamental limitation of Physics termed "Einstein's Equivalence Principle". Not possible to distinguish between a conference room in 1g **downwards** earth gravity and being in outer space and accelerating **upwards** at 1g by rocket
- **KEY**: The accelerometer measurement $\boldsymbol{G}_p$ is equal to its acceleration $\boldsymbol{A}_p$ **minus** the downwards pointing gravity $\boldsymbol{g}$ rotated by rotation matrix $\boldsymbol{R}$

$$\boldsymbol{G}_p = \boldsymbol{A}_p - \boldsymbol{R}\boldsymbol{g}$$

Accelerometer Sensors: Physics 101 (cont)

- Vector diagram of the forces that the accelerometer sees in its rotated reference frame



- If the orientation is known (using a gyroscope sensor for example) then the rotation matrix ***R*** can be computed, ***g*** is always 1 gravity downwards and it's possible to solve for A_p :

$$A_p = G_p + Rg$$

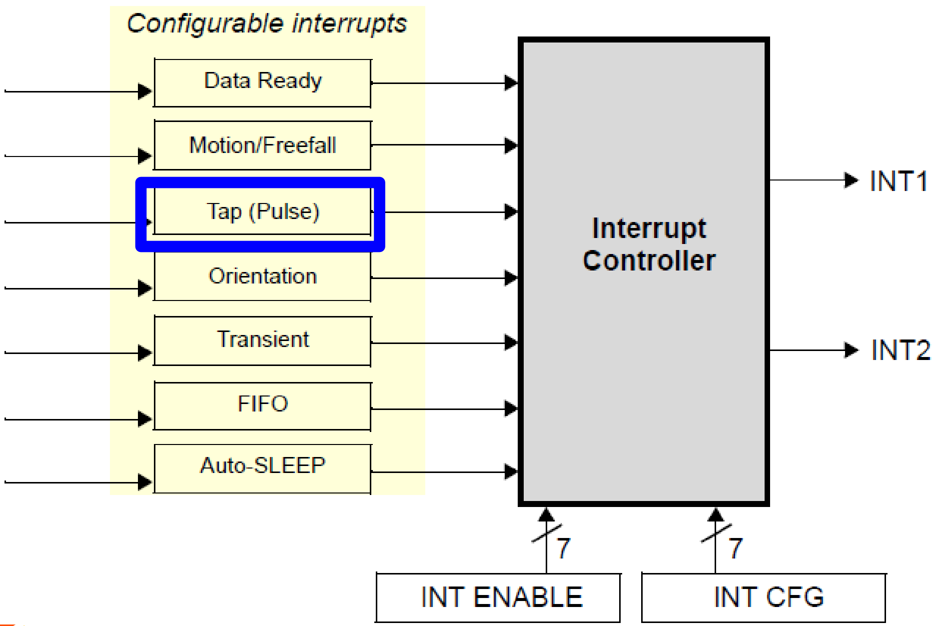
- But an accelerometer alone cannot separate acceleration from gravity and that's just the way Physics works

Accelerometer Sensors: Freefall

- Newton's law of gravity: If the accelerometer MEMS has mass m then the gravitational force $\boldsymbol{F}$ on it is $\boldsymbol{F}=m\boldsymbol{Rg}$
- Newton's second law: When dropped, the acceleration (in the accelerometer frame) $\boldsymbol{A}_p=\boldsymbol{F}/m=\boldsymbol{Rg}$
- The accelerometer reading $\boldsymbol{G}_p$ is then **zero** until impact on the ground:

$$\boldsymbol{G}_p = \boldsymbol{A}_p - \boldsymbol{Rg} = \boldsymbol{0}$$

- Freefall detection logic is commonly provided as an interrupt source



Note: Rotating freefall creates an additional centripetal acceleration which can prevent the freefall threshold being triggered. Sidestep this problem by placing the accelerometer at the product's centre of mass.

Demonstration Session (i)

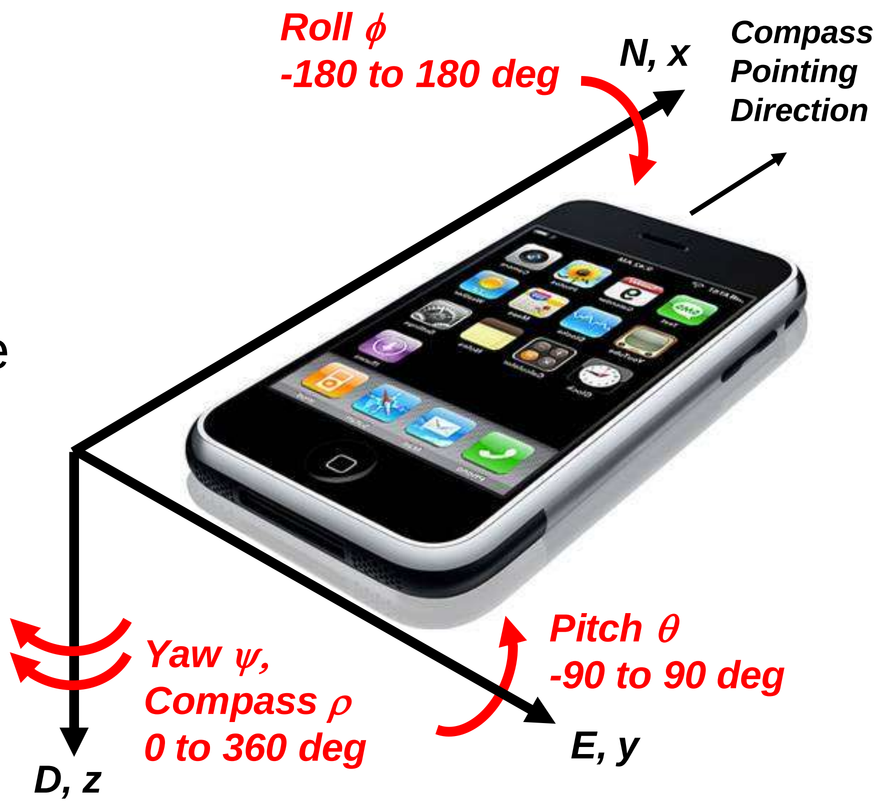
- Measurement on the 1g gravity sphere in the absence of acceleration

- Sensitivity of the accelerometer sensor to both gravity linear acceleration

- Freefall conditions

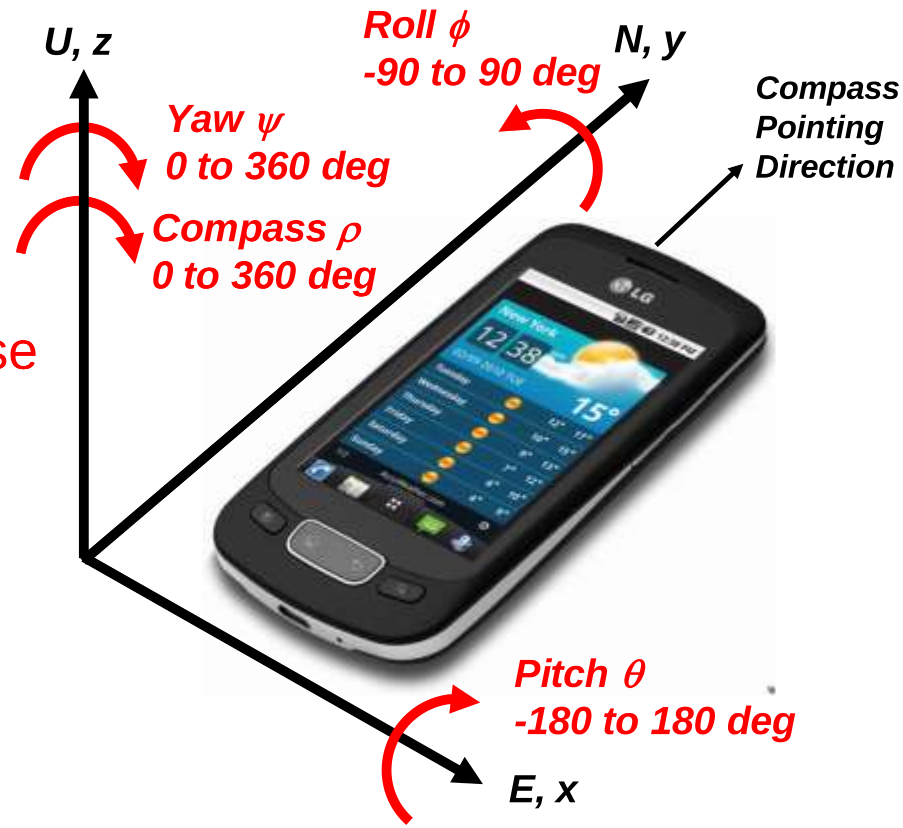
Aerospace (iPhone) Coordinate System

- Termed "NED" system: x=North, y=East, z=Down
- Gravity positive, acceleration negative
- Roll range -180 to 180 deg
- Pitch range -90 to 90 deg
- Yaw range 0 to 360 deg
- Compass heading = Yaw
- Angle convention: Clockwise
- Sequence: Yaw, Pitch, Roll



Android Coordinate System

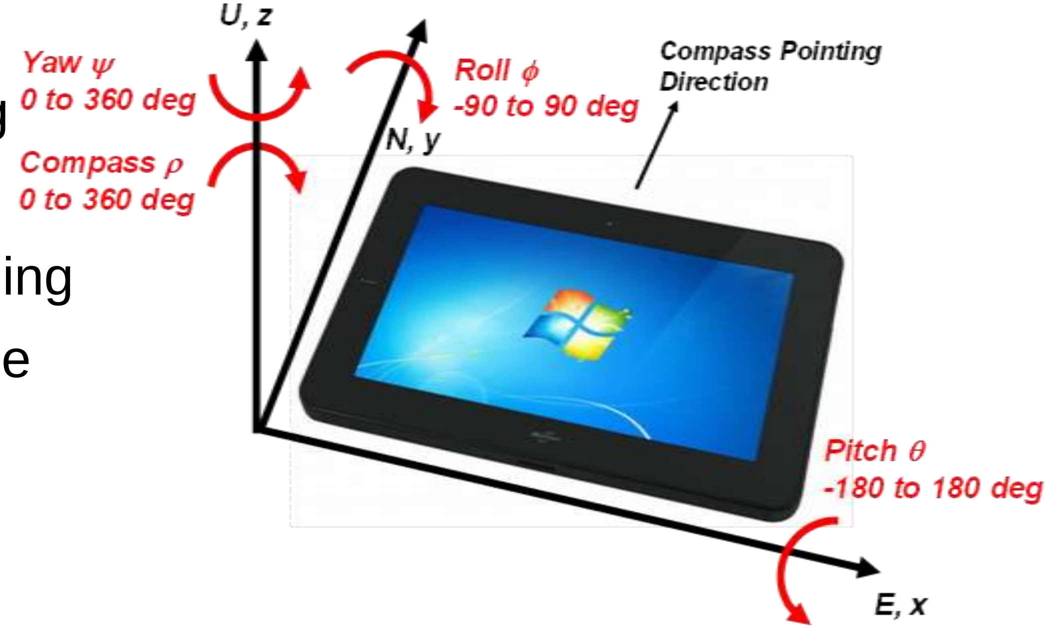
- Termed “ENU” system: x=East, y=North, z=Up
- Acceleration positive, gravity negative
- Roll range -90 to 90 deg
- Pitch range -180 to 180 deg
- Yaw range 0 to 360 deg
- Compass heading = Yaw
- Angle convention: Anti-Clockwise
- Sequence: Yaw, Roll, Pitch



freescale™

Windows 8 Coordinate System

- Termed “ENU” system: x=East, y=North, z=Up
- Gravity positive, acceleration negative
- Roll range -90 to 90 deg
- Pitch range -180 to 180 deg
- Yaw range 0 to 360 deg
- Yaw=Minus Compass heading
- Angle convention: Clockwise
- Sequence: Yaw, Pitch, Roll



Tilt-Sensing (Windows 8 Coordinate System)

- Previously we introduced the equation defining the accelerometer output

$$\boldsymbol{G}_p = \boldsymbol{A}_p - \boldsymbol{R}\boldsymbol{g}$$

- Windows 8 is a gravity +ve standard so we need to flip the sign to map the raw accelerometer signal from the package to Windows 8 coordinates

$$\boldsymbol{G}_p = -\boldsymbol{A}_p + \boldsymbol{R}\boldsymbol{g}$$

- Cannot separate acceleration from gravity so assume zero acceleration

$$\boldsymbol{G}_p = \boldsymbol{R}\boldsymbol{g}$$

- Apply a rotation ***R*** in yaw ψ, then pitch θ and finally roll ϕ:

$$\boldsymbol{G}_p = \begin{pmatrix} G_{px} \\ G_{py} \\ G_{pz} \end{pmatrix} = \boldsymbol{R}\boldsymbol{g} = \boldsymbol{R}_y(\phi)\boldsymbol{R}_x(\theta)\boldsymbol{R}_z(\psi)\begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} = \begin{pmatrix} \cos\theta \sin\phi \\ -\sin\theta \\ -\cos\theta \cos\phi \end{pmatrix}$$

Independent of yaw angle ψ

- Simple algebra gives the solution for Windows 8 roll and pitch angles:

$$\tan\phi = \frac{-G_{px}}{G_{pz}} \qquad \tan\theta = \left(\frac{G_{py}}{G_{pz}\cos\phi - G_{px}\sin\phi}\right)$$

- No solution is possible for the yaw angle ψ since an accelerometer is insensitive to rotation about gravity and that's just the way Physics works

Demonstration Session (ii)

- Roll and Pitch angles for Aerospace (iPhone), Android and Windows 8 coordinate systems

- Insensitivity of accelerometer to rotation about the gravity vector

- Corruption of accelerometer-derived tilt angles in the presence of linear acceleration

How Accurately Can We Determine Tilt?

- **Restriction**: An accelerometer sensor alone cannot separate acceleration from gravity so the first requirement is that there is no acceleration. That's reasonable and the same applies to a spirit level
- **Sensor Noise**: This might be 2mg RMS but averaging 100 measurements (1sec at 100Hz) takes that down to 0.2mg: Negligible
- **Zero g Offset**: Typical (1 stdev) values in each channel are 25mg as shipped which increases to 33.5mg as a result of soldering thermal stresses on the MEMS structure. This is the dominant term.

Zero-g level offset accuracy (1)	TyOff			±25		mg
Zero-g level offset accuracy, post-board mount (2)	TyOffPBM			±33.5		mg

- Geometry gives typical tilt error (roll or pitch) 1.91 deg with 33.5mg:

$$Typical\ error = \arcsin\left(\frac{\mathrm{TyOffPBM(mg)}}{1000}\right)$$

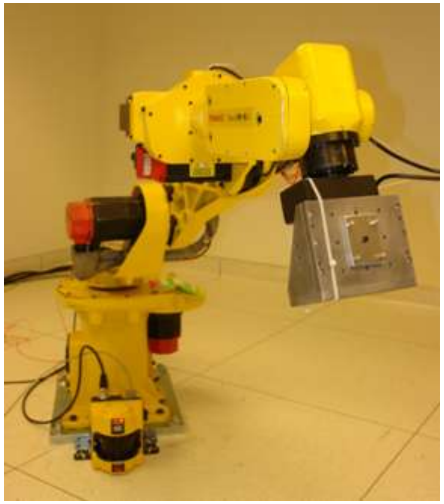
$$\approx \frac{\mathrm{TyOffPBM(mg)}}{17.5}\,\mathrm{(deg)} = 1.91\,\mathrm{deg\ for\ TyOffPBM} = 33.5\mathrm{mg}$$

freescale™

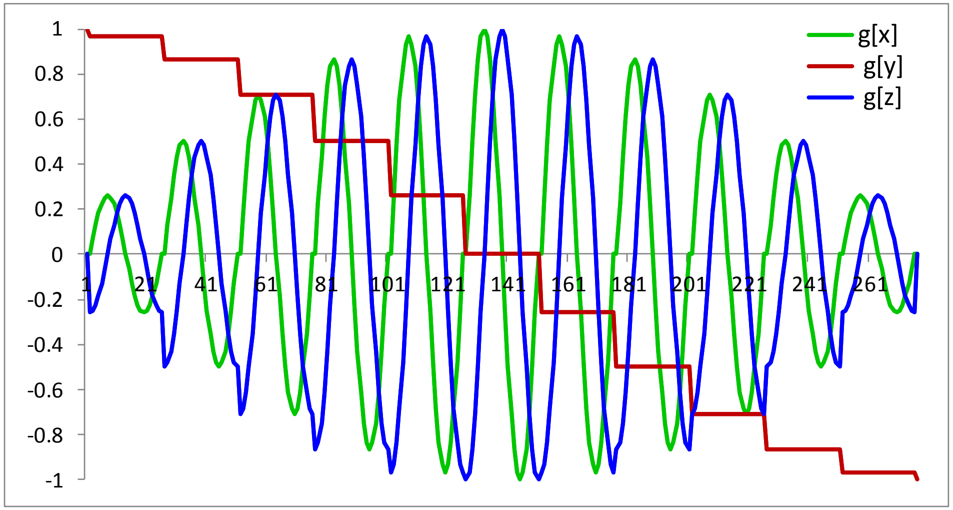
Precision Re-Calibration after Board Mount (i)

- Robot used to rotate accelerometers to 277 positions on the 1g sphere

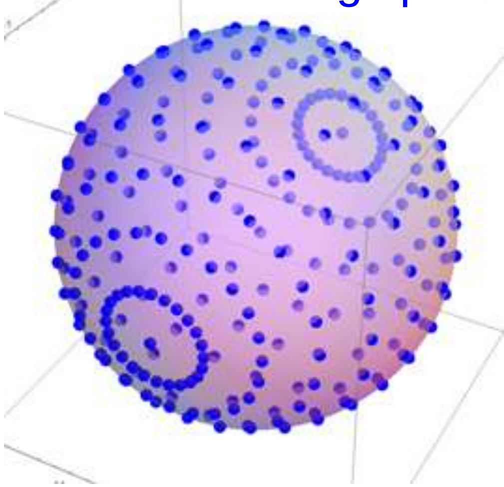
Robot



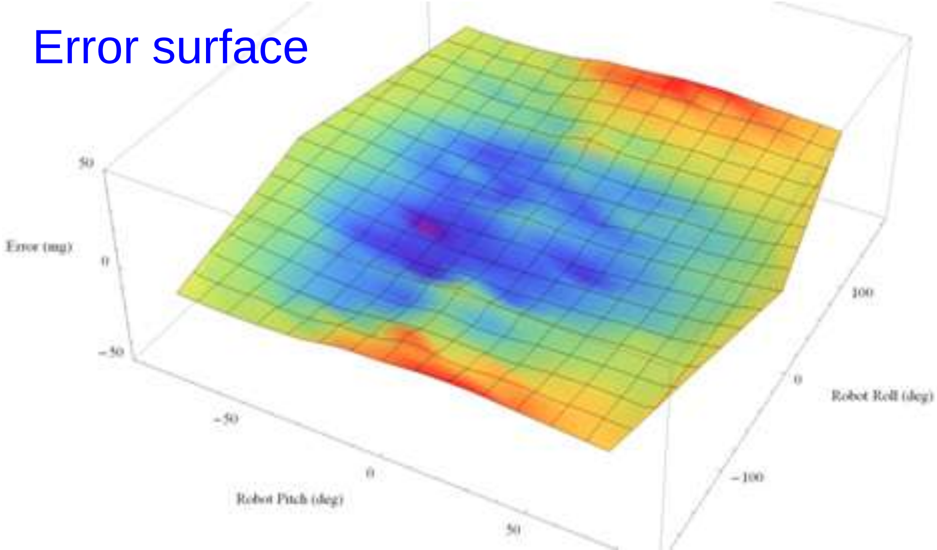
x, y, z gravitational fields



Locations on 1g sphere



Error surface



Precision Re-Calibration after Board Mount (ii)

- With zero acceleration, a perfect accelerometer measures the rotated gravity field:

$$\boldsymbol{G_p} = -\boldsymbol{Rg}$$

- Under a general linear model of accelerometer distortion, the accelerometer measures:

$$\boldsymbol{G_p} = -\boldsymbol{WRg} - \boldsymbol{V}$$

- ***W*** is a 3x3 matrix modeling gain correction, cross axis interference and rotation of the accelerometer package on the PCB
- ***V*** is a 3x1 matrix correcting for the post board mount offsets

Rotation and cross-axis terms in blue

Zero g Offset Corrections

$$\boldsymbol{W} = \begin{pmatrix} W_{xx} & W_{xy} & W_{xz} \\ W_{yx} & W_{yy} & W_{yz} \\ W_{zx} & W_{zy} & W_{zz} \end{pmatrix} \qquad \boldsymbol{V} = \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}$$

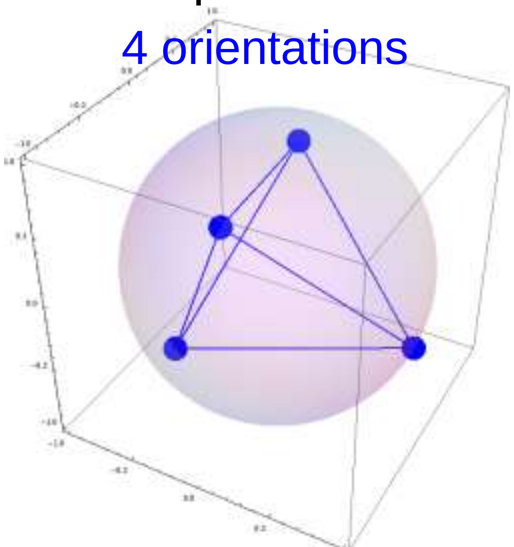
Gain correction terms in red

freescale™

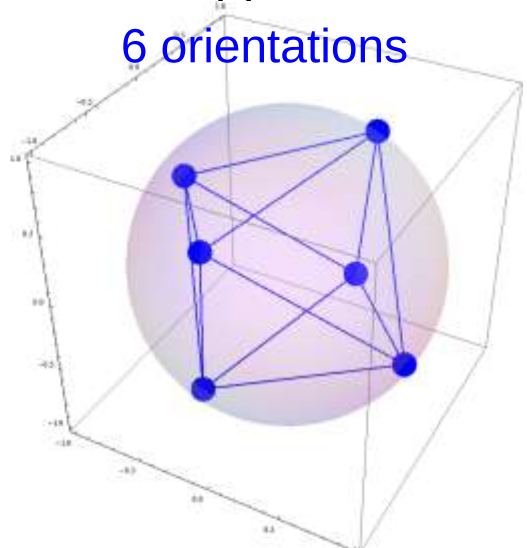
Precision Re-Calibration after Board Mount (iii)

- 12 unknowns in total to be determined in ***W*** (3x3=9) and ***V*** (3x1=3)
- Use a jig to place the PCB at a minimum of 4 orientations (4x3=12 measurements) or 6 to be safe (6x3=18 measurements)
- Least squares fit mathematics documented in Application Note **AN4399**

4 orientations



6 orientations



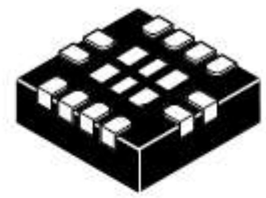
- Typical result from MMA8652 showing ~30mg residual reduced to 2.7mg
- And 2.7mg corresponds to about 0.15 deg angular error

Initial RMS error **29.869mg**	Re-calibrated RMS error **2.686mg**	Wxx=0.98159 Wyy=1.01213 Wzz=1.00546	Wxy=-0.00095 Wxz=0.00829 Wyz=0.00071	Vx=16.87mg Vy=-14.41mg Vz=-10.77mg

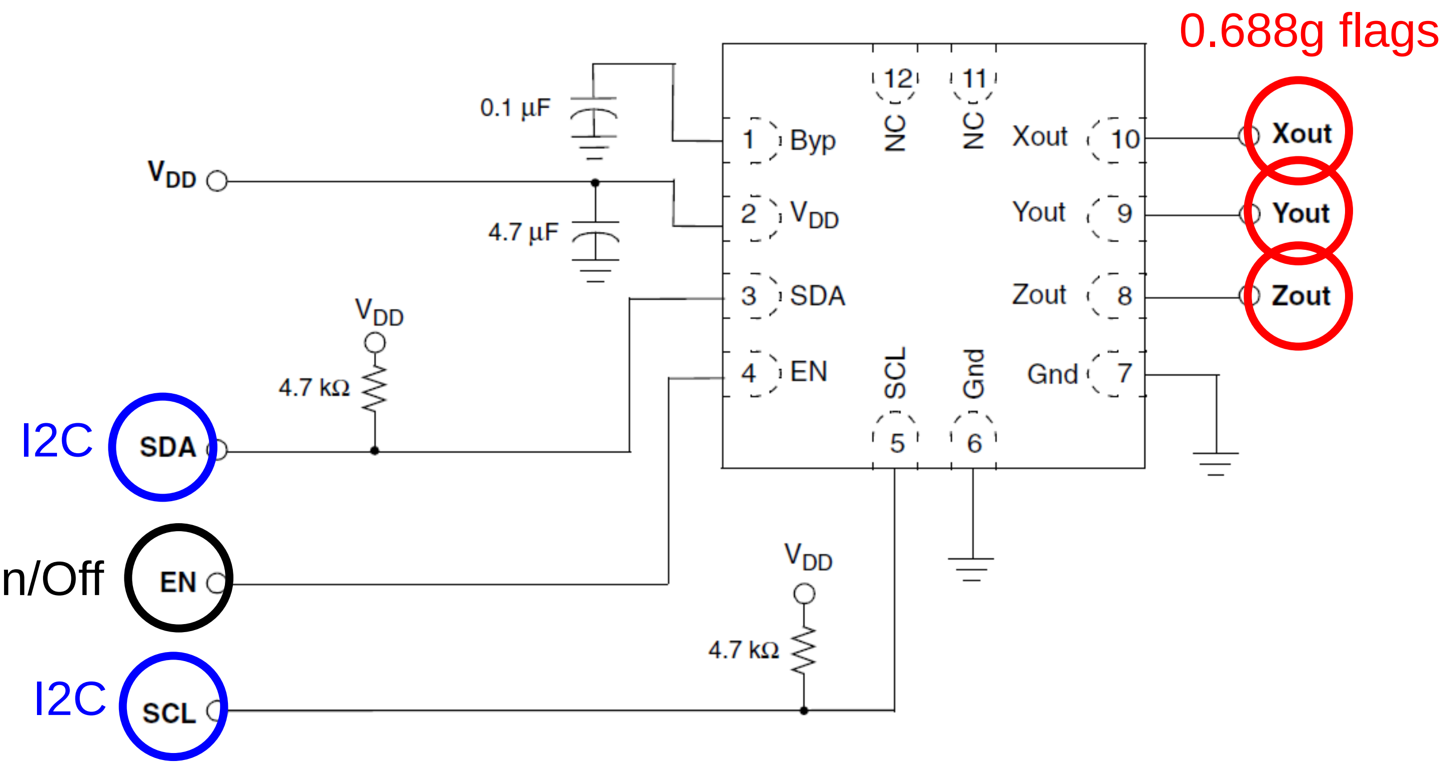
freescale™

MMA8491 Accelerometer: Schematic

- Range up to 8g, 14 bit resolution
- Extremely low power: 400nA per Hz sampling rate
- Three output lines flagging |acceleration| above or below 0.688g in each axis (~46.5deg)



12-Lead Industrial QFN
3 mm x 3 mm x 1.05 mm
0.65 mm Pitch



Accelerometer

Key Products 2013

	Sample	Prod	SRP$	Applications
MMA8451/52/53 3-axis ±2, ±4, ±8 g 10/12/14-bit **Digital I²C**				
• Embedded function and interrupt : (FIFO, High pass filter, P/L,…) • Ultra low noise (99 µg/√Hz), low TCO (0.15mg/°C) • High performance Consumer & Industrial • Down to 0.25mg/LSB sensitivity • 1.95...3.6 Volt, 3 x 3 x 1 mm QFN	Web	ok	0.86 0.68	Tilt Measurement Pedometer Power Management eCompass Asset Tracking Activity Monitor Sports Watch Fleet Management Remote Controls Appliance
FXLS8471 3-axis ±2, ±4, ±8 g 14-bit **Digital SPI**	Now	Nov-13	TBD	
• Embedded functions and interrupts (all + Vector magnitude) • High performance industrial grade • 1.95...3.6 Volt, 3 x 3 x 1 mm QFN				
MMA8652/53 3-axis ±2, ±4, ±8 g 10/12-bit **Digital I²C**	Web	Now	0.53	
• Embedded functions and interrupts (8652 same than MMA8451) • Software compatible with the MMA845x family • Low cost • 1.95...3.6 Volt, 2 x 2 mm DFN				
MMA8491 3-axis Tilt Sensor 14-bit **Digital I²C + 3 Logic Out**	Now	Now	0.77	Tamper Sensor Rolling Ball Switch Alarm/Security Freefall Detect Remote Control Low Power Wake-up
• Ultra low power down to 400 nA/Hz, • 3 logic outputs to flag tilt on the 3 axis • I²C interface to read raw acceleration data • 1.95...3.6 Volt, 3 x 3 x 1 mm DFN				

Drivers Available on request

freescale™

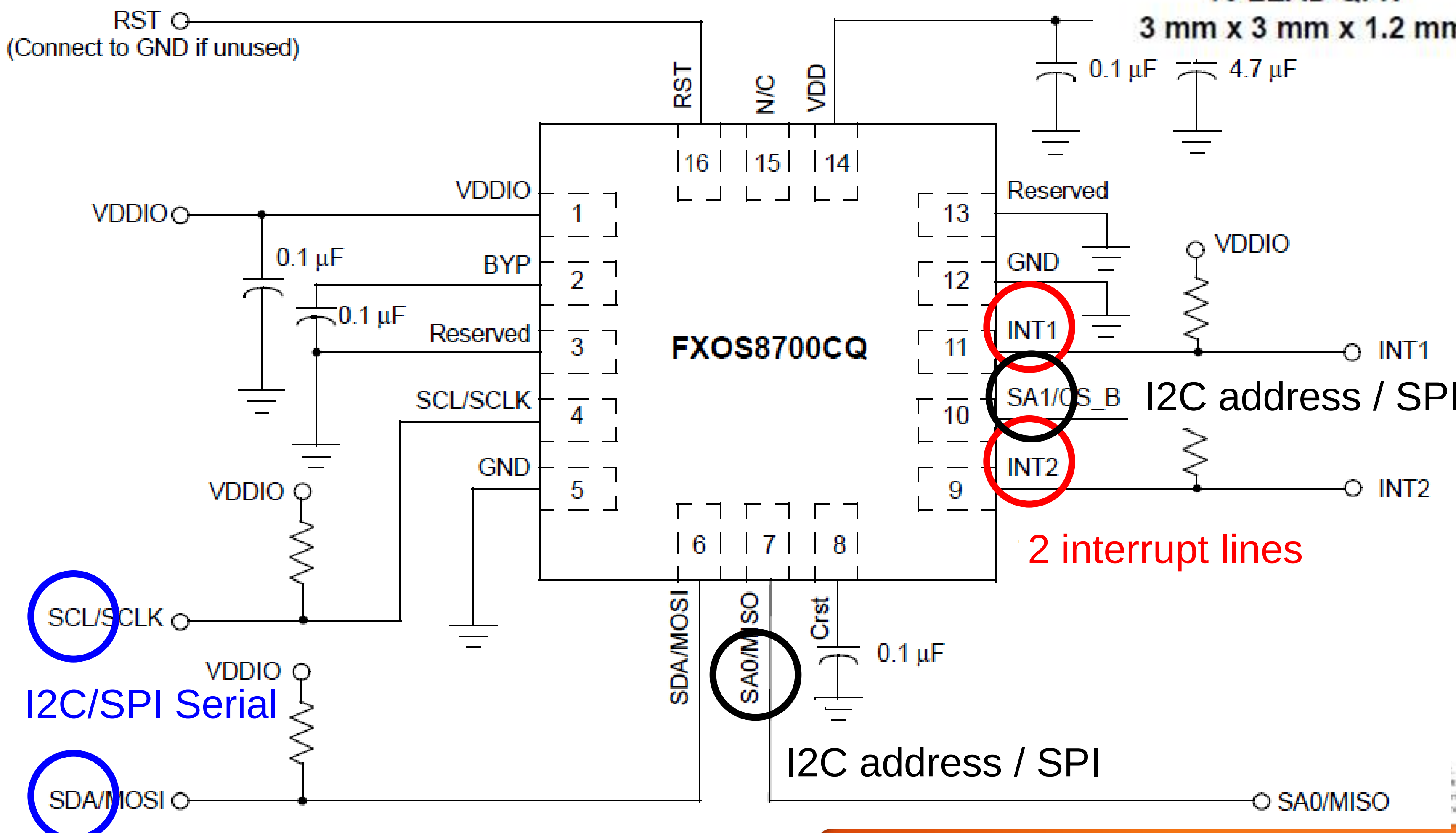
Magnetometer

Magnetometer

- Measures the three components of the magnetic field at the sensor
- Very commonly combined with an accelerometer to implement a tilt-compensated eCompass.
- Aligns map displays to the direction that the handset is pointing.
- **KEY**: Magnetometer sensors must be used with calibration software that models and subtracts circuit board interference

FXOS8700 Combination Accelerometer and Magnetometer

16 LEAD QFN
3 mm x 3 mm x 1.2 mm



FXOS8700 Magnetometer: Configuration

- Five I2C calls to configure the FXOS8700

```
// write 0x00 to accelerometer control register 1 to place FXOS8700 into standby
I2C2_Buf[0] = A_CTRL_REG1; I2C2_Buf[1] = 0x00;
I2C2_MasterSendBlock(DeviceDataPtr, I2C2_Buf, 2, LDD_I2C_SEND_STOP);

// write 0x1F to magnetometer control register 1
// [4-2]: m_os=111: 8x oversampling (for 200Hz) to reduce magnetometer noise
// [1-0]: m_hms=11: select hybrid mode with accel and magnetometer active
I2C2_Buf[0] = M_CTRL_REG1; I2C2_Buf[1] = 0x1F;
I2C2_MasterSendBlock(DeviceDataPtr, I2C2_Buf, 2, LDD_I2C_SEND_STOP);

// write 0x20 to magnetometer control register 2
// [5]: hyb_autoinc_mode=1 to map the magnetometer registers to follow accelerometer
// [1-0]: m_rst_cnt=00 to enable magnetic reset each cycle
I2C2_Buf[0] = M_CTRL_REG2; I2C2_Buf[1] = 0x20;
I2C2_MasterSendBlock(DeviceDataPtr, I2C2_Buf, 2, LDD_I2C_SEND_STOP);

// write 0x01 to XYZ_DATA_CFG register
// [1-0]: fs=01 for accelerometer range of +/-4g range with 0.488mg/LSB
I2C2_Buf[0] = A_XYZ_DATA_CFG; I2C2_Buf[1] = 0x01;
I2C2_MasterSendBlock(DeviceDataPtr, I2C2_Buf, 2, LDD_I2C_SEND_STOP);

// write 0x0D to accelerometer control register 1
// [5-3]: dr=001=1 for 200Hz data rate (when in hybrid mode)
// [2]: lnoise=1 for low noise mode
// [0]: active=1 to take the part out of standby and enable sampling
I2C2_Buf[0] = A_CTRL_REG1; I2C2_Buf[1] = 0x0D;
I2C2_MasterSendBlock(DeviceDataPtr, I2C2_Buf, 2, LDD_I2C_SEND_STOP);
```

FXOS8700 Magnetometer: Data Read

- Two I2C calls to read the three accelerometer and three magnetometer channels (each 16 bit, totaling 12 bytes) into the byte buffer I2C2_Buff[]

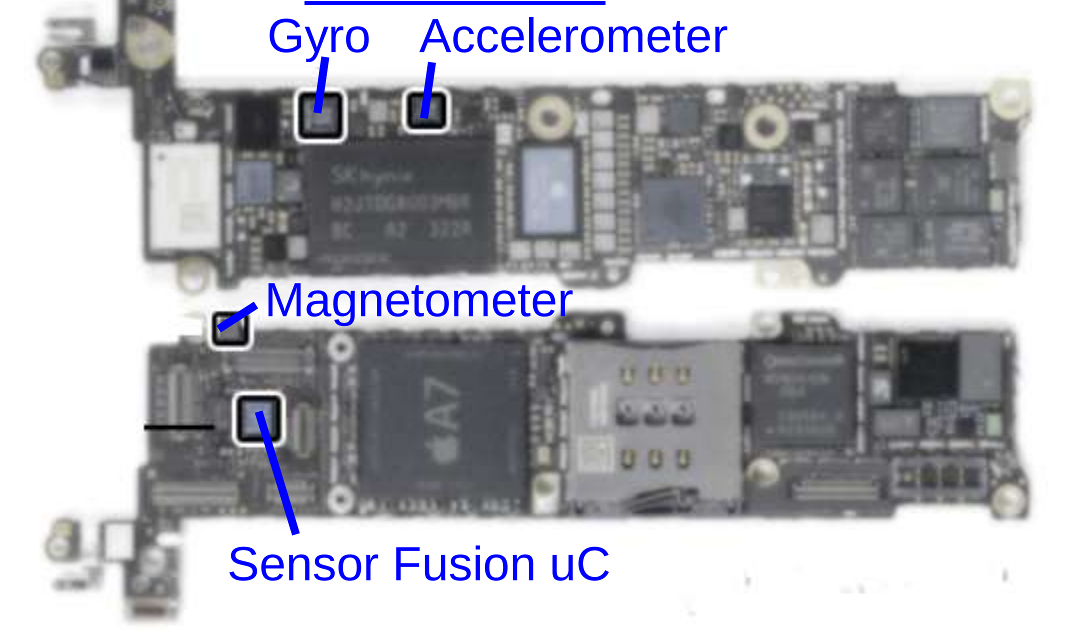
```
// read 12 bytes of measurement data into I2C2_Buff
I2C2_Buf[0] = A_DATA_REG;
I2C2_MasterSendBlock(DeviceDataPtr, I2C2_Buf, 1, LDD_I2C_NO_SEND_STOP);
I2C2_MasterReceiveBlock(DeviceDataPtr, I2C2_Buf, 12, LDD_I2C_SEND_STOP);
```

```
// place the 12 bytes read into the 16 bit accelerometer and magnetometer structures
pthisAccel->iData[X] = (I2C2_Buf[0] << 8) | I2C2_Buf[1];
pthisAccel->iData[Y] = (I2C2_Buf[2] << 8) | I2C2_Buf[3];
pthisAccel->iData[Z] = (I2C2_Buf[4] << 8) | I2C2_Buf[5];
pthisMag->iData[X] = (I2C2_Buf[6] << 8) | I2C2_Buf[7];
pthisMag->iData[Y] = (I2C2_Buf[8] << 8) | I2C2_Buf[9];
pthisMag->iData[Z] = (I2C2_Buf[10] << 8) | I2C2_Buf[11];
```

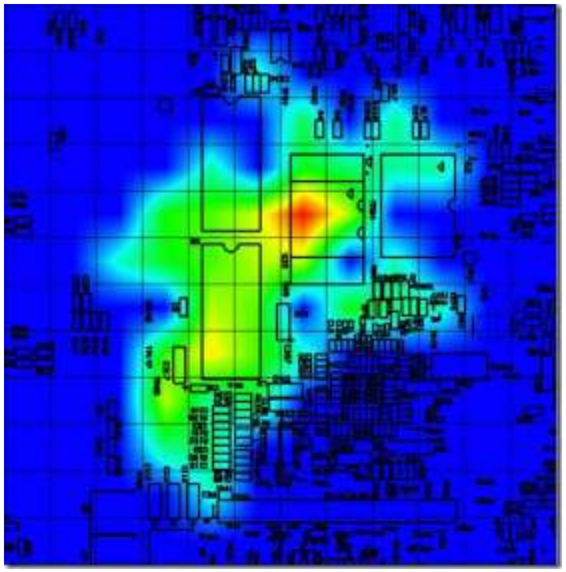
Magnetic Environment inside PCB

- Speaker magnets, high power current traces, steel RF shields combine to create interfering magnetic fields up to 1000uT
- The earth's geomagnetic field which provides the compass heading is only 40uT or so and is swamped by the interfering fields
- The magnetic calibration software must track and subtract the interfering fields to an accuracy of 0.25uT
- Magnetometer must be carefully placed - typically at the PCB edge

iPhone 5 PCB



Typical magnetic field scan



Tilt-Compensated eCompass in Brief

- The mathematical model for the magnetometer measurement $\boldsymbol{B}_p$ is:
  $$\boldsymbol{B}_p = \boldsymbol{W}\boldsymbol{R}_\phi \boldsymbol{R}_\theta \boldsymbol{R}_\psi \boldsymbol{B}_r + \boldsymbol{V}$$
- Which in words means:
  - "Take the earth's geomagnetic field $\boldsymbol{B}_r$ which we know points northwards and downwards (in the northern hemisphere)
  - "Rotate it by yaw ψ (compass), pitch θ and roll ϕ to match the PCB's orientation
  - "Stretch the result with the 3x3 "soft iron" matrix ***W*** which models unmagnetized ferromagnetic components on the PCB"
  - "Add on the effects of permanently magnetized ferromagnetic components on the PCB modeled by the "hard iron" vector ***V***"
- Determining the hard and soft iron calibration ***W*** and ***V*** is a complex non-linear optimization but, as always, refer customers to the click-through license software at www.freescale.com/ecompass

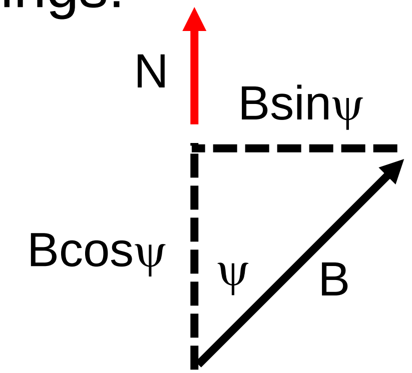
Tilt-Compensated eCompass in Brief

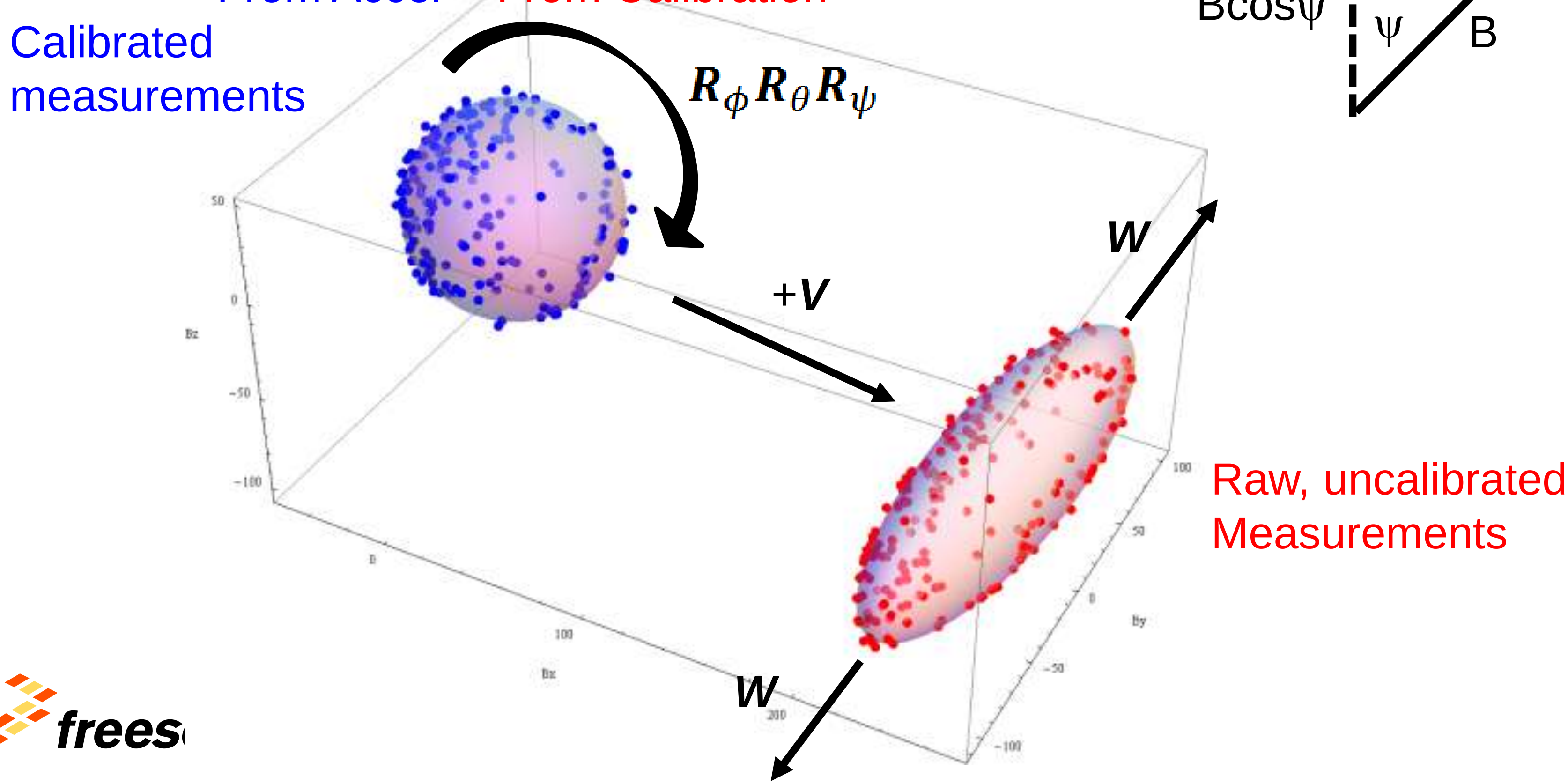
- Simple algebra gives the yaw (compass) angle ψ as the arctan of the horizontal de-rotated and calibrated readings:

$$R_\psi B_r = R_\theta^{-1} R_\phi^{-1} W^{-1}(B_p - V)$$

From Accel

From Calibration





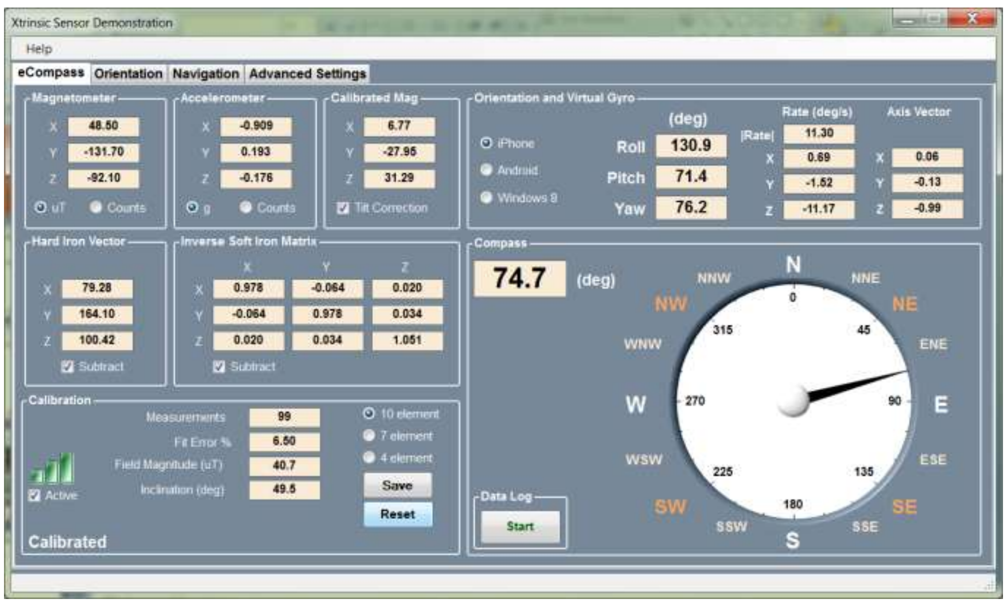
freesc

Demonstration Session (iii)

- Calculation of magnetic calibration coefficients
- Determination of yaw (compass) angle
- Sensitivity of eCompass to linear accelerations which corrupt tilt-compensation
- Different behaviors of Aerospace, Android and Windows 8 compass heading angles when compass is inverted

RD4247FXOS8700 eCompass (PC Hosted)

- RD4247FXOS8700 eCompass
  - PC executable implements eCompass algorithms and user interface
  - Available from www.freescale.com/sensortoolbox



Magnetometer

Key Products 2013

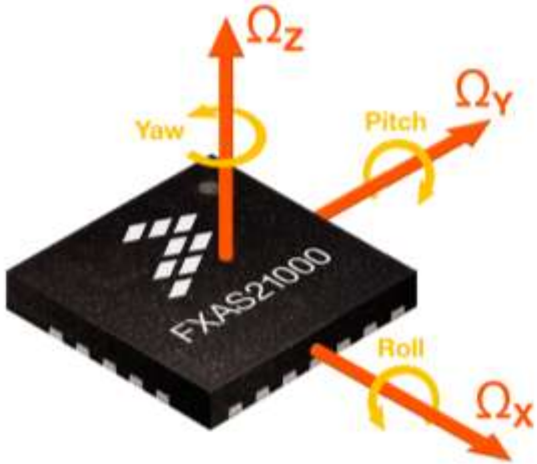
	Sample	Prod	SRP$	Applications
MAG3110G 3-axis Digital **Magnetometer**	Web	Now	0.56	Industrial Compass Current Sensing Presence Detection Car Detect Industrial Safety Magnetic Tamper Sports Watch Diving Watch
Capable of measuring geomagnetic fields • Wide dynamic range +/- 1000 µT (10 Gauss) • Low power in measurement mode 8.6 µA . • ODR Output data rate up to 80 Hz • Interrupt pin trigger when new data available • Tilt compensation and Soft/Hard Iron calibration SW available • 1.95...3.6 Volt, 2 x 2 x 0.85 mm DFN				

	Sample	Prod	SRP$	Applications
FXOS8700CQ COMBO 6-axis **Magnetometer** and **Accelerometer**	Now	Now	1.25	Industrial Compass Current Sensing Presence Detection Car Detect Industrial Safety Magnetic Tamper Sports Watch Diving Watch
• Capable of measuring geomagnetic fields with Tilt compensation • Wider dynamic range +/- 1200 µT • ODR up to 800 Hz by sensors, or 400Hz in Hybrid mode • Embedded interrupts and pre-programmed functions • Low power 80 µA in Hybrid mode @ 25 Hz • 1.95...3.6 Volt, 3 x 3 x 1.2 mm QFN				

Gyroscope

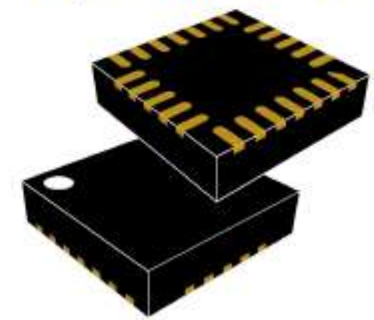
Gyroscope Sensor

- Gyroscopes measure angular rotation rates about three axes
- **KEY**: Gyroscope sensors on their own cannot determine absolute orientation (roll, pitch, yaw angles) but only **changes** in orientation
- **KEY**: Commonly used with accelerometers and magnetic sensors which provide the absolute orientation
- **KEY**: Gyroscopes are (almost) insensitive to linear acceleration and completely insensitive to magnetic fields
- Gyroscopes are active devices containing MEMS structures vibrating in x, y, z planes (10kHz to 20kHz) which, like a bicycle wheel, prefer to stay vibrating in that plane while the package rotates
- The deflection of the vibrating structures is proportional to angular velocity and detected, like an accelerometer, by a C to V circuit
- Like accelerometers, the gyro combines a MEMS die and ASIC die in same package

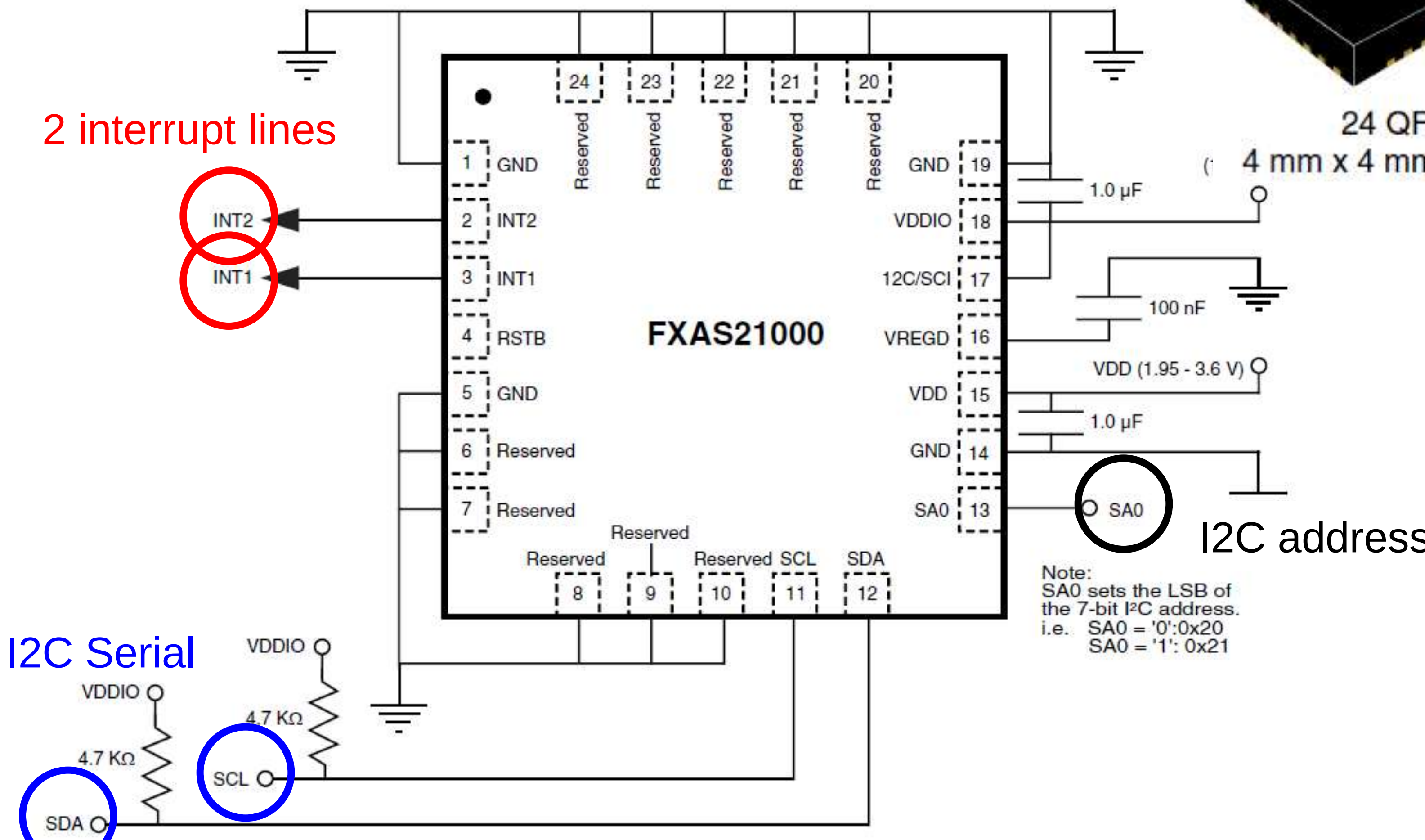


FXAS21000 Gyroscope (Electrical)

Top and bottom view



24 QFN
4 mm x 4 mm x 1 mm



freescale™

FXAS21000 Gyroscope: Configuration and Read

- Two I2C calls to configure the FXAS21000

```
// write 0x00 to control register 1 to place FXOS21000 in Standby
I2C2_Buf[0] = G_CTRL_REG1; I2C2_Buf[1] = 0x00;
I2C2_MasterSendBlock(DeviceDataPtr, I2C2_Buf, 2, LDD_I2C_SEND_STOP);

// write 0x02 to control register 1 to configure 200Hz ODR and enter Active mode
// [7]: ZR_cond=0: do not initiate a zero rotation rate condition
// [6]: RST=0: do not trigger a software reset
// [5]: ST=0 do not trigger a self test
// [4-2]: DR[2-0]=000 for 200Hz ODR
// [1-0]: Active=1, Ready=0 for Active mode
I2C2_Buf[0] = G_CTRL_REG1; I2C2_Buf[1] = 0x02;
I2C2_MasterSendBlock(DeviceDataPtr, I2C2_Buf, 2, LDD_I2C_SEND_STOP);
```

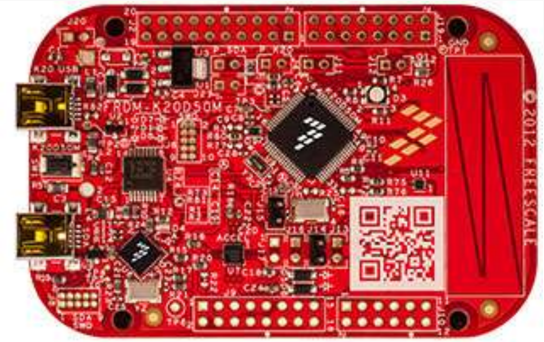
- And two I2C calls to read the three channels of 16 bit measurements (six bytes total) into the byte buffer I2C2_Buff[]

```
// read six bytes (three channels of 16 bit data)
I2C2_Buf[0] = G_DATA_REG;
I2C2_MasterSendBlock(DeviceDataPtr, I2C2_Buf, 1, LDD_I2C_NO_SEND_STOP);
I2C2_MasterReceiveBlock(DeviceDataPtr, I2C2_Buf, 6, LDD_I2C_SEND_STOP);

// place the six bytes into the 16 bit gyro measurement structure
pthisGyro->iData[X] = (I2C2_Buf[0] << 8) | I2C2_Buf[1];
pthisGyro->iData[Y] = (I2C2_Buf[2] << 8) | I2C2_Buf[3];
pthisGyro->iData[Z] = (I2C2_Buf[4] << 8) | I2C2_Buf[5];
```

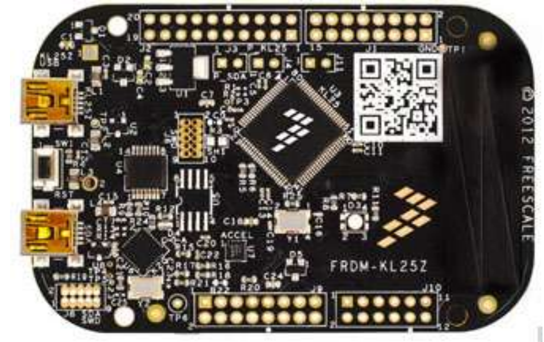
Kinetis Sensor Fusion Platforms

Board: Freedom FRDM-K20D50M
- MCU = MK20DX128VLH5
- Core clock: 48MHz
- Bus clock: 24MHz
- OpenSDA programming interface
- Capacitive touch slider
- Ambient light sensor
- MMA8451Q accelerometer



Available NOW

Board: Freedom FRDM-KL25Z
- MCU = KL25Z128VLK4
- Core clock: 48MHz
- Bus clock: 24MHz
- OpenSDA programming interface
- Capacitive touch slider
- MMA8451Q accelerometer



Available NOW

Board: FRDM-FXS-MULTI shield board
- Freescale MAG3110 3-axis magnetometer
- MMA8652QR1 3-axis accelerometer
- Freescale MMA9553LR1 pedometer
- **Freescale FXOS8700CQ 3-axis magnetometer + 3 axis accelerometer**
- **Freescale FXAS21000 3-axis gyroscope**
- Freescale MPL3115A2 Pressure Sensor / Altimeter
- 3rd party ambient light sensor

Available Nov '13

freescale™

Alternative Hardware

element 14 Xtrinsic Sensors Board
Freedom compatible shield Supports:
MPL3115A2 pressure sensor/altimeter
MAG3110 magnetometer
MMA8491Q accelerometer
No wireless
The fusion library has not been tested on this board (9-axis is not applicable).
Details: http://www.element14.com/community/docs/DOC-54360?ICID=knode-xtrinsic-evk

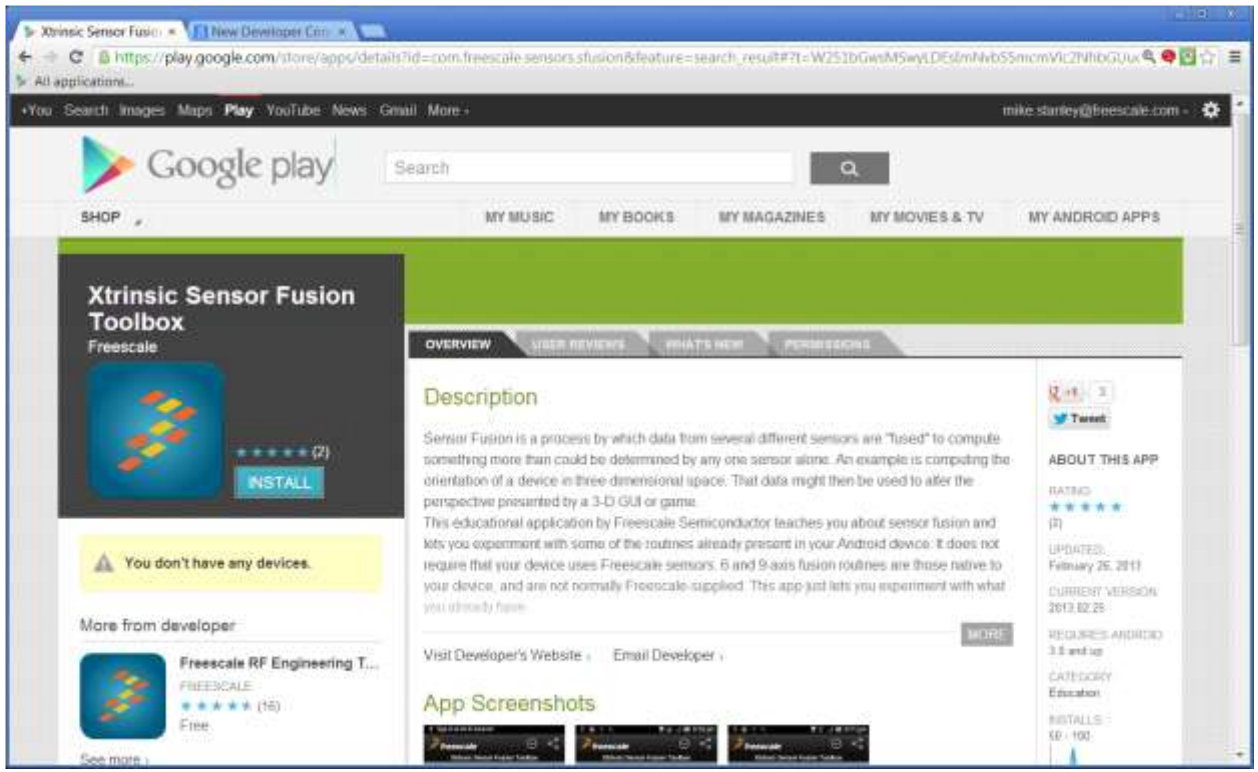
Avnet Wi-Go System
KL25Z Freedom board
MMA8451Q accelerometer
MAG3110 magnetometer
MPL33115A2 pressure sensor/altimeter
ambient light sensor
Wi-Fi communications
Details: http://www.em.avnet.com/en-us/design/drc/Pages/Avnet-Wi-Go-Module.aspx

WIN8_MK20DX256Z
Algorithms team development board
NOT available outside Freescale
FXOS87000CQ mag/accel
FXAS21000 gyroscope
Bluetooth interface

Both element14 and Avnet boards are capable of running 6-axis/eCompass portions of the fusion library. That port has not been made to date. The Wi-Go DOES run a prior version of Freescale's MagCal/eCompass code, embedded within the Avnet distributed sample application.

UI runs on Android phones and Tablets:
All sensor fusion is performed on Kinetis uC
Orientation transmitted by Bluetooth

http://play.google.com/store/apps/details?id=com.freescale.sensors.sfusion



requires Android 3.0 or higher

Xtrinsic Sensor Fusion Software

- Supplied under development license from Freescale
- Implemented as pure C-code sitting on top of device driver and MQX-lite implementations created via Processor Expert
- Shipped in the form of CodeWarrior projects compatible with the Android demo
- Shipped as a mix of precompiled libraries (containing the fusion routines) and C source code.
- Execution is “locked” to the use of specific Kinetis-brand MCUs

The Classic Applications for Sensor Fusion: Demonstration Session (v)

- Accelerometer:
  - Inclinometer provides roll and pitch (tilt) estimation (eg camera horizon)
  - But cannot determine yaw (compass) angle and the performance degrades when linear accelerations is present
- Accelerometer + Magnetometer (eCompass)
  - Tilt compensated eCompass provides complete orientation solution including yaw (compass) heading
  - But tilt-compensation algorithm uses the accelerometer and so degrades when linear acceleration is present
  - Compass is perturbed by external magnetic interference
- Accelerometer + Gyroscope (Gaming consoles)
  - Gyroscope separates gravity from linear acceleration in accelerometer signal
  - Robust estimates of rotation rate and pitch / roll (from gravity component)
  - But cannot determine absolute yaw (compass) angle since accelerometer is insensitive to rotation about the vertical gravity axis
- Accelerometer + Magnetometer + Gyroscope (Gyro stabilized eCompass)
  - Provides accurate orientation even under high accelerations and external magnetic interference
  - No significant drawbacks

Gyroscope

Key Products 2013

	Sample	Prod	SRP$	Applications
FXAS21000 3-axis Digital **Gyroscope**	Jul-13	Oct-13	2.09	Inertial Navigation Gaming Remote Control Smart Phones Stabilization
• Full scale range +/-1600°/sec • Angular speed resolution better than 0.2°/sec • Current consumption in run mode : 5.5mA • 1.95V-3.6V voltage supply, 4 x 4 x 1 mm QFN 				

Other Resources

Xtrinsic Sensor Fusion Software

- V3 release with C source code and documentation under click-through license at www.freescale.com/ecompass
- Awarded “Product of the Year” by ‘Electronic Products’ Dec 2012

Reference C Code and Documentation

- No charge for use with Freescale Accelerometers and Magnetometers

Current Updates and Releases

ECOMPASS_SW: New eCompass Software Version 3

By clicking the "I AGREE" button below, you will confirm your agreement with these terms, which will be legally binding. If you do not agree with these terms, click the "I DO NOT AGREE" button below, which will cancel the transaction. These terms were most recently revised on 2013-08-20. You can print and/or download these terms by clicking the "PRINT" or "DOWNLOAD" buttons below.

I Agree | I Do Not Agree

- Processor agnostic ANSI C code with pre-built PC demonstration and 12 detailed application notes

- include.h
- magnetic.c
- main.c
- main.exe
- matrix.c
- orientation.c
- simulation.c

- AN4676.pdf
- AN4681.pdf
- AN4684.pdf
- AN4685.pdf
- AN4696.pdf
- AN4697.pdf
- AN4698.pdf
- AN4699.pdf
- AN4700.pdf
- AN4706.pdf
- AN4709.pdf
- MAGCALSWUG.pdf

Documentation Available Under License (Behind the Click-Through License Firewall)

AN4676

Euler Angle, Rotation Matrix and Quaternion Representations of Orientation

in Aerospace, Android and Windows 8 Coordinates

AN4681

Implementing a Virtual Gyro Using Accelerometer and Magnetometer Sensors

AN4684

Magnetic Calibration of Hard and Soft Iron Interference

AN4685

Tilt-Compensated eCompass in Aerospace, Android and Windows 8 Coordinate Systems

AN4696

Accelerometer and Magnetometer Sensor Simulation for Tilt-Compensated eCompass

AN4697

Low Pass Filtering of Orientation Estimates

AN4706

Accelerometer and Magnetometer Selection and Configuration

AN4709

Tilt Estimation Using an Accelerometer

in Aerospace, Android and Windows 8 Coordinate Systems

freescale™

Detailed Derivations and Reference C Functions

- This is a powerful marketing tool and unique to Freescale
- Support emails come directly to the author with same day response

The Android compass heading angle ρ always equals the yaw angle ψ:

$$\rho = \psi \tag{153}$$

The function `fAndroidAnglesDegFromRotationMatrix` computes the roll ϕ, pitch θ, yaw ψ and compass heading ρ in units of degrees from the NED rotation matrix $\boldsymbol{R}$. The matrix $\boldsymbol{R}$ is assumed to be a valid rotation matrix but bounds checking is performed on R_{02} before taking the inverse sine to prevent run-time errors.

```
// extract the Android angles in degrees from the Android rotation matrix
void fAndroidAnglesDegFromRotationMatrix(float **R, float *pfPhiDeg, float *pfTheDeg,
    float *pfPsiDeg, float *pfRhoDeg)
{
    // calculate the roll angle -90.0 <= Phi <= 90.0 deg
    if (R[0][2] >= 1.0F)
        *pfPhiDeg = 90.0F;
    else if (R[0][2] <= -1.0F)
        *pfPhiDeg = -90.0F;
    else
        *pfPhiDeg = (float) asin(R[0][2]) * FRADTODEG;

    // calculate the pitch angle -180.0 <= The < 180.0 deg
    *pfTheDeg = (float)atan2(-R[1][2], R[2][2]) * FRADTODEG;
    // map +180 pitch onto the functionally equivalent -180 deg pitch
    if (*pfTheDeg == 180.0F)
        *pfTheDeg = -180.0F;

    // calculate the yaw-compass angle 0.0 <= Psi < 360.0 deg
    *pfPsiDeg = (float)atan2(-R[0][1], R[0][0]) * FRADTODEG;
    if (*pfPsiDeg < 0.0F)
        *pfPsiDeg += 360.0F;
    // check for rounding errors mapping small negative angle to 360 deg
    if (*pfPsiDeg >= 360.0F)
        *pfPsiDeg = 0.0F;
    // the compass heading angle Rho equals the yaw angle Psi
    // this definition is compliant with Motorola Xoom tablet behavior
    *pfRhoDeg = *pfPsiDeg;

    return;
}
```

5.4 Windows 8 Euler angles from rotation matrix

The Windows 8 rotation matrix derived in equation (132) has the form:

$$\begin{pmatrix} R_{00} & R_{01} & R_{02} \\ R_{10} & R_{11} & R_{12} \\ R_{20} & R_{21} & R_{22} \end{pmatrix} = \begin{pmatrix} \cos\phi\cos\psi - \sin\phi\sin\theta\sin\psi & \cos\phi\sin\psi + \cos\psi\sin\phi\sin\theta & -\sin\phi\cos\theta \\ -\cos\theta\sin\psi & \cos\theta\cos\psi & \sin\theta \\ \cos\psi\sin\phi + \cos\phi\sin\psi\sin\theta & \sin\phi\sin\psi - \cos\phi\cos\psi\sin\theta & \cos\phi\cos\theta \end{pmatrix} \tag{154}$$

The solution for the three Windows 8 Euler angles is:

$$\phi = \tan^{-1}\left(\frac{-R_{02}}{R_{22}}\right), -90 \le \phi < 90\ deg \tag{155}$$

The pitch angle θ has a range of $-180 \le \theta < 180\ deg$, but is first computed in the range $-90°$ to $90°$.

$$\theta = \sin^{-1}(R_{12}), -90 \le \theta < 90\ deg \tag{156}$$

Since ϕ is in the range $-90°$ to $90°$, it follows that $\cos\phi$ is non-negative and that $R_{22} = \cos\phi\cos\theta$ has the same sign as $\cos\theta$ and can be used to correct θ into the range $-180° \le \theta$

$$\theta \leftarrow \pi - \theta \ if\ R_{22} < 0 \ and\ \theta > 0 \tag{157}$$

$$\theta \leftarrow -\pi - \theta \ if\ R_{22} < 0 \ and\ \theta \le 0 \tag{158}$$

The general solution for the yaw angle ψ for non-zero $\cos\theta$ is:

$$\psi = \tan^{-1}\left(\frac{-\cos\theta R_{10}}{\cos\theta R_{11}}\right), 0 \le \psi < 360\ deg, \theta \ne -90deg, , \theta \ne 90deg \tag{159}$$

At $\theta = 90deg$, equation (154) simplifies to:

$$\begin{pmatrix} R_{00} & R_{01} & R_{02} \\ R_{10} & R_{11} & R_{12} \\ R_{20} & R_{21} & R_{22} \end{pmatrix} = \begin{pmatrix} \cos\phi\cos\psi - \sin\phi\sin\psi & \cos\phi\sin\psi + \cos\psi\sin\phi & 0 \\ 0 & 0 & 1 \\ \cos\psi\sin\phi + \cos\phi\sin\psi & \sin\phi\sin\psi - \cos\phi\cos\psi & 0 \end{pmatrix} for\ \theta = 90deg \tag{160}$$

$$= \begin{pmatrix} \cos(\psi+\phi) & \sin(\psi+\phi) & 0 \\ 0 & 0 & 1 \\ \sin(\psi+\phi) & -\cos(\psi+\phi) & 0 \end{pmatrix} for\ \theta = 90deg \tag{161}$$

$$\Rightarrow \tan(\psi+\phi) = \left(\frac{R_{01}}{R_{00}}\right) for\ \theta = 90deg \tag{162}$$

At $\theta = -90deg$, equation 5.4.1 simplifies to:

$$\begin{pmatrix} R_{00} & R_{01} & R_{02} \\ R_{10} & R_{11} & R_{12} \\ R_{20} & R_{21} & R_{22} \end{pmatrix} = \begin{pmatrix} \cos\phi\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\psi - \cos\psi\sin\phi & 0 \\ 0 & 0 & -1 \\ \cos\psi\sin\phi - \cos\phi\sin\psi & \sin\phi\sin\psi + \cos\phi\cos\psi & 0 \end{pmatrix} for\ \theta = -90deg \tag{163}$$

$$= \begin{pmatrix} \cos(\psi-\phi) & \sin(\psi-\phi) & 0 \\ 0 & 0 & -1 \\ -\sin(\psi-\phi) & \cos(\psi-\phi) & 0 \end{pmatrix} for\ \theta = -90deg \tag{164}$$

$$\Rightarrow \tan(\psi-\phi) = \left(\frac{R_{01}}{R_{00}}\right) for\ \theta = -90deg \tag{165}$$

The Windows 8 compass heading angle ρ is the negative (modulo 360°) of the yaw angle ψ:

$$\rho = -\psi \tag{166}$$

The function `fWin8AnglesDegFromRotationMatrix` computes the roll ϕ, pitch θ, yaw ψ and compass heading ρ in units of degrees from the Windows 8 rotation matrix $\boldsymbol{R}$.

Documentation Freely Available

AN4399

High Precision Calibration of a Three-Axis Accelerometer

AN4246

Calibrating an eCompass in the Presence of Hard and Soft-Iron Interference

AN4248

Implementing a Tilt-Compensated eCompass using Accelerometer and Magnetometer Sensors

AN4247

Layout Recommendations for PCBs Using a Magnetometer Sensor

AN3461

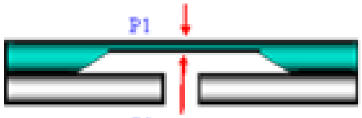
Tilt Sensing Using a Three-Axis Accelerometer

Pressure

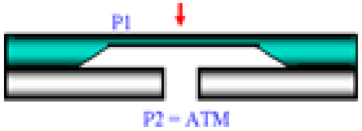
Pressure Sensors

Devices are composed of single silicon, piezoresistive devices capable of monitoring processes in the industrial, medical, automotive, and avionics industries. Pressure sensors are composed of the following reference designs:

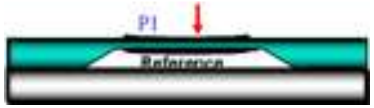
- Differential: Difference in pressure between two points is measured as pressure is applied to both sides of the die/sensing element.



- Gauge: Bottom side is exposed to the atmosphere, while pressure is applied to top of die.

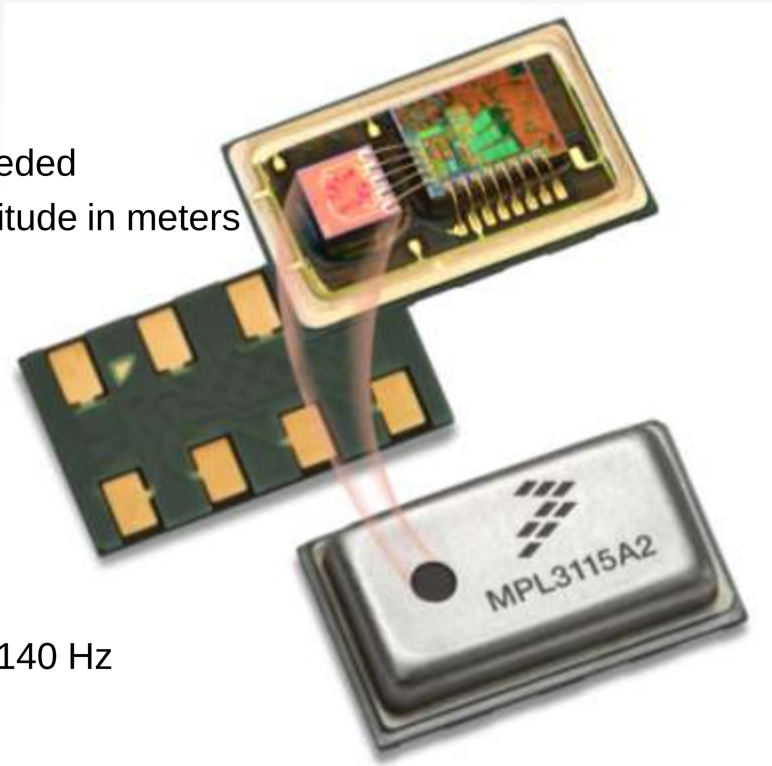


- Absolute: Pressure is applied to the top of the die while the bottom of the die is a vacuum-sealed reference.



Xtrinsic MPL3115A2
Precision Digital Altimeter

- **Differentiating Points**
  - Internally compensated, software is not needed
  - Direct reading pressure in Pascals and altitude in meters
  - On-board intelligence
- **Product Features**
  - Altitude resolution : < 1 foot / 0.3 m
  - Pressure resolution: 1.5 Pa
  - Pressure range: 20 – 110 kPa
  - Calibrated pressure range: 50 – 110 kPa
  - 1.95V to 3.6V supply voltage
  - Variable output sampling rate (OST) up to 140 Hz
  - Current Consumption:
    - Standby mode: 2 µA
    - Low-power mode: 8.5 µA at 1 Hz
  - I²C digital interface
- **Typical Applications**
  - High Accuracy Altimeter
  - Smartphones / Tablets
  - GPS Enhancement for Location Based Services



Package

3 x 5 x 1.1 mm LGA

In Production

freescale™

Additional Accelerometers: SPI and Analog

Xtrinsic MMA955xL

3-Axis Accelerometer plus 8MHz 32-bit ColdFire MCU

- **Differentiating Points**
  - Intelligent Motion Platform with embedded libraries
  - Sensor hub capability
  - Power management features and low power modes

- **Product Features**
  - 1.71 to1.89V supply voltage
  - Output data rate (ODR) 488 Hz
  - +/-2g, 4g, 8g configurable dynamic ranges available
  - 14/12/10/8 bit resolution available
  - 16K Flash, 2K RAM

- **Typical Applications**
  - Sensor Hub
  - Real-time orientation detection
  - Static orientation detection

Package

3x3x1mm LGA, 0.5mm pitch

Variations

MMA9550L

Infrastructure only functions
User Flash: 6.5 Kbytes
User RAM: 576 bytes
10k $1.61

MMA9551L

Infrastructure plus gestures
User Flash: 4.5 Kbytes
User RAM: 452 bytes
10k $1.61

MMA9553L

Infrastructure plus pedometer
User Flash: 1.5 Kbytes
User RAM: 200 bytes
10k $1.72

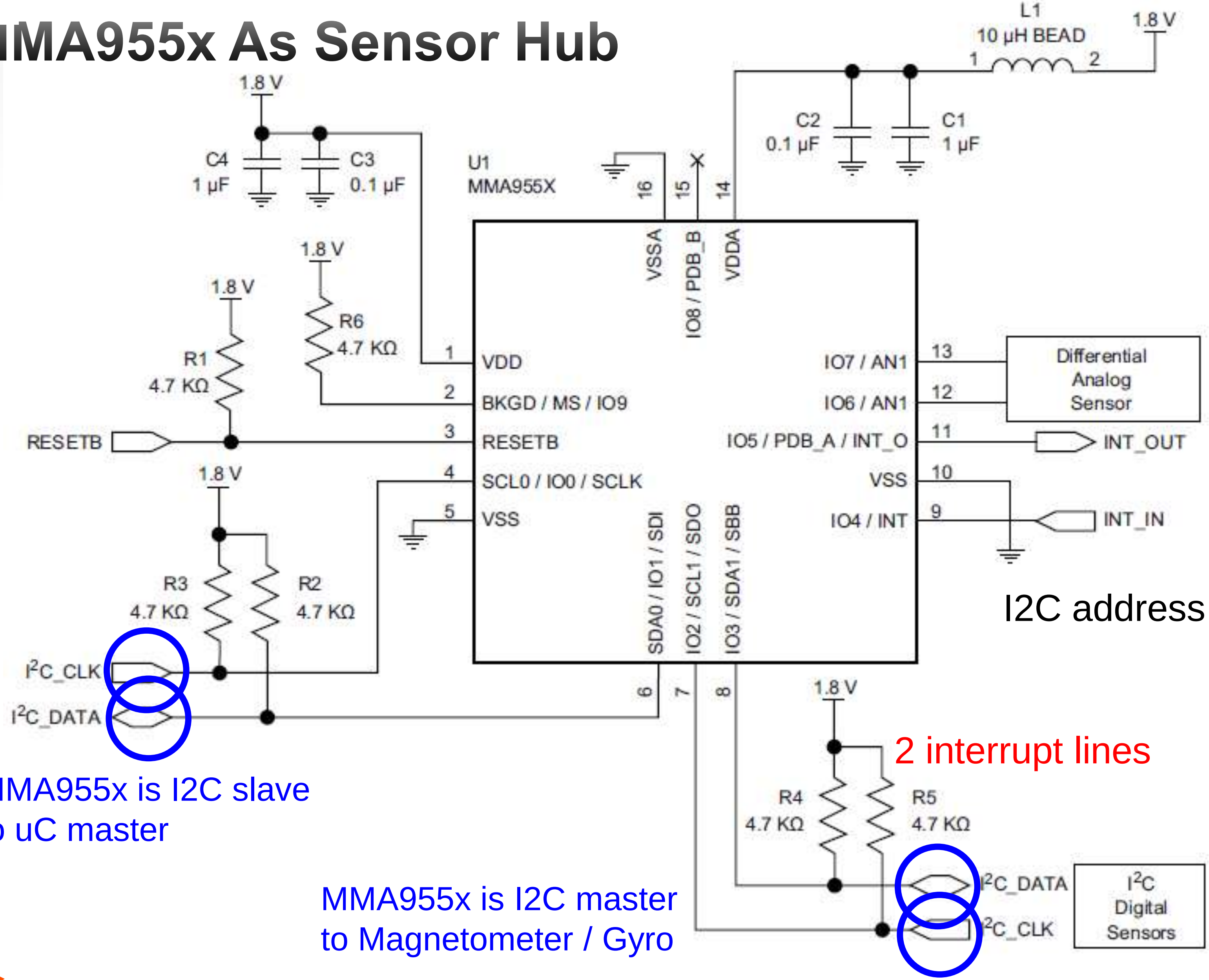
MMA9559L

Lightweight Infrastructure
User Flash: 14 Kbytes
User RAM: 1.5 bytes
10k $1.53

Availability

Samples: Now
Production: Now

MMA955x As Sensor Hub



I2C address / SPI

2 interrupt lines

MMA955x is I2C slave to uC master

MMA955x is I2C master to Magnetometer / Gyro

freescale™

Xtrinsic FXLC95000L
3-Axis Accelerometer plus 16MHz 32-bit ColdFire MCU

- **Differentiating Points**
  - Intelligent Motion Platform with embedded libraries
  - Sensor hub capability
  - Power management features and low power modes

- **Product Features**
  - 1.72 to 3.6V I/O supply voltage
  - Output data rate (ODR) 488 Hz
  - +/-2g, 4g, 8g configurable dynamic ranges available
  - 14/12/10/8 bit resolution available
  - 128K Flash, 16K RAM

Package
3x5x1mm LGA, 0.5mm pitch

- **Typical Applications**
  - Sensor Hub
  - Real-time orientation detection
  - Static orientation detection

Availability
Samples: Now
Production: August 2013

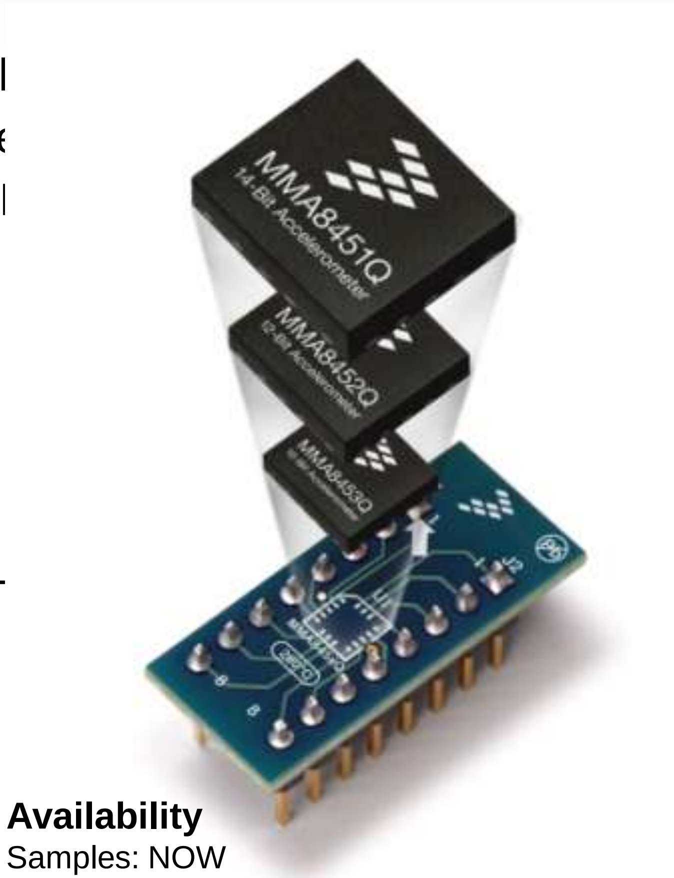
Pricing
10k $2.04

freescale™

FXLS8471 3-axis Low-G Accel with SPI

Digital Accelerometer providing capability for industrial applications requiring a SPI interface with best in class offset, temperature performance and digital feature set

- **Differentiating Points**
  - Interface compatibility – first sub $2 SPI accel
  - Benchmark Performance – Best in class Offse
  - Complete Integrated Feature Set – Including I
- **Product Features**
  - SPI/I²C digital output interface w/interrupts
  - 1.95V to 3.6V supply voltage
  - ±2g/±4g/±8g dynamically selectable
  - Output data rate (ODR) from 800Hz to 1.563H
  - Benchmark Output Noise (<125ug/rHz)
  - Resolution up to 0.25mg/count
  - QFN 3x3x1.05mm 0.5mm pitch package
- **Typical Application**
  - Industrial – SPI interface

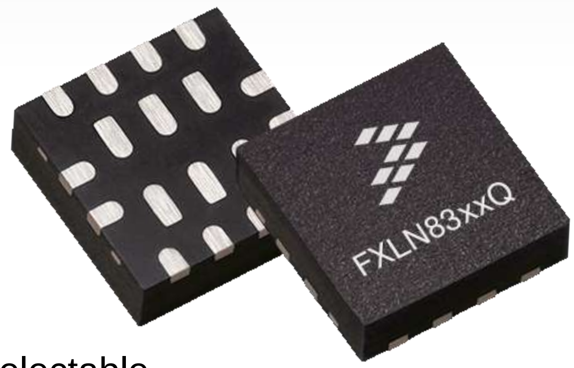


Availability
Samples: NOW
Production: September '13

Pricing
10k $0.95

High-Performance Low g Analog Accelerometers

- **Differentiating Points**
  - Zero-g Level Offset Accuracy
  - Board Mount Offset Shift
  - Output Noise



- **Product Features**
  - 1.71V to 3.6V supply voltage
  - ±2g/±4g/±8g/±16g dynamically selectable
  - 150µg/√Hz noise
  - Non-ratiometric Analog Parallel Outputs
  - High Bandwidth: Up to 4.9kHz Capable
  - Visible Solder Inspection Leads
  - QFN 3x3x1.05mm 0.65mm pitch package

- **Typical Applications**
  - Vibration, shock, motion detect
  - Factory Automation
  - Real-time orientation detection
  - Static orientation detection

Package

3x3x1mm QFN, 0.65mm pitch

Availability

Samples: Now

Production: November 2013

Industrial

FXLN8371Q

4.9kHz bandwidth
±2g, 228mV/g sensitivity
±8g, 57mV/g sensitivity
10k $1.61

FXLN8372Q

4.9kHz bandwidth
±4g, 115mV/g sensitivity
±16g, 29mV/g sensitivity
10k $1.69

FXLN8361Q

±2g, 115mV/g sensitivity
±8g, 29mV/g sensitivity
10k $1.21

FXLN8362Q

±4g, 115mV/g sensitivity
±16g, 29mV/g sensitivity
10k $1.27

Consumer

FXLN8351Q

4.9kHz bandwidth
±2g, 115mV/g sensitivity
±8g, 29mV/g sensitivity

FXLN8352Q

±4g, 115mV/g sensitivity
±16g, 29mV/g sensitivity

NXP

freescale™