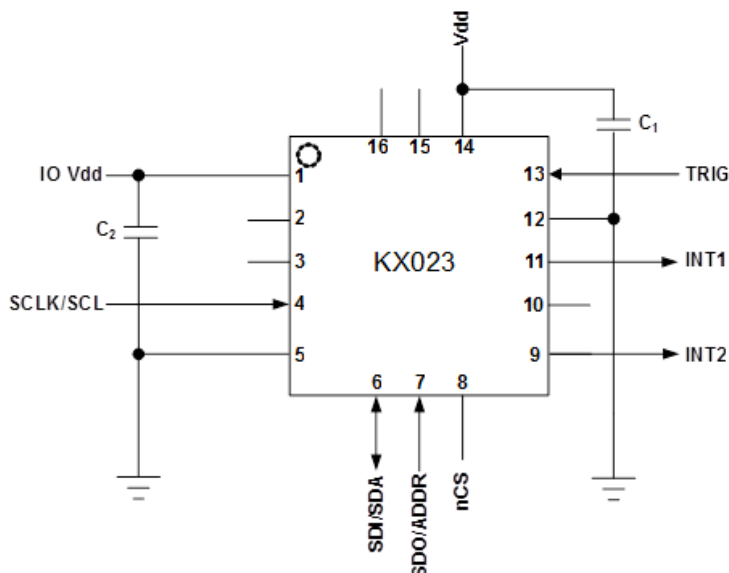


## Introduction

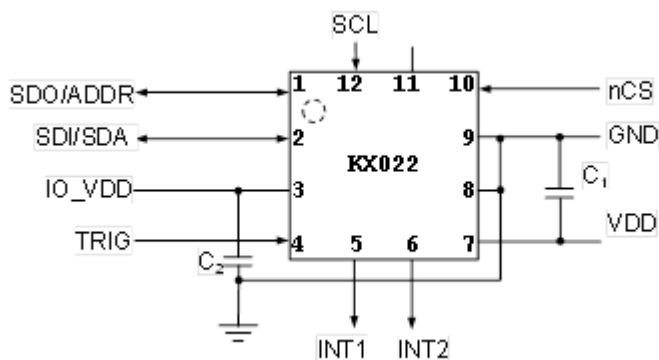
This application note will help developers quickly implement proof-of-concept designs using the KX023 and KX022 tri-axis accelerometer. Please refer to the KX023 and KX022 data sheet for additional implementation guidelines. Kionix strives to ensure that our accelerometers will meet design expectations by default, but it is not possible to provide default settings to work in every environment. Depending on the intended application, it is very likely that some customization will be required in order to optimize performance. The information provided here will help the developer get the most out of the KX023 and KX022.

## Circuit Schematic

This section shows recommended wiring for the KX023 and KX022, based on proven operation of the part. Specific applications may require modifications from these recommendations. Please refer to the KX023 and KX022 Data Sheet for all pin descriptions.



**Figure 1. KX023 Application Schematic**



**Figure 2: KX022 Application Schematic**

## Quick Start Implementation

Here we present several basic ways to initialize the part. These can vary based on desired operation, but generally the initial operations a developer wants to do are: 1) read back acceleration data asynchronously, 2) read back acceleration data when next data is ready via interrupt, 3) use the Wake Up function, 4) activate the tilt position function, and 5) activate the tap/double-tap function. These cursory solutions are provided as a means for configuring the part to a known operational state. Note that these conditions just provide a starting point, and the values may vary as developers refine their application requirements.

### 1. Asynchronous Read Back Acceleration Data (Setting G-Range and ODR)

- Write 0x40 to Control Register 1 (CNTL1) to set the accelerometer in stand-by mode, to set the performance mode to 16 bit resolution and G-range to +/-2g

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>CNTL1</b>	<b>0x18</b>	<b>0001 1000</b>	<b>0x40</b>	<b>0100 0000</b>

- Write 0x02 to Data Control Register (ODCNTL) to set the Output Data Rate (ODR) of the accelerometer to 50 Hz. (Note: This is also the default value.)

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>ODCNTL</b>	<b>0x1B</b>	<b>0001 1011</b>	<b>0x02</b>	<b>0000 0010</b>

- Write 0xC0 to Control Register 1 (CNTL1) to set the accelerometer into operating mode (PC1 = 1)

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>CNTL1</b>	<b>0x18</b>	<b>0001 1000</b>	<b>0xC0</b>	<b>1100 0000</b>

- Acceleration data can now be read from the XOUT\_L, XOUT\_H, YOUT\_L, YOUT\_H, ZOUT\_L, and ZOUT\_H registers in 2's complement format.

## 2. Synchronous Hardware Interrupt Read Back Acceleration Data (Setting G-Range and ODR)

- Write 0x60 to Control Register 1 (CNTL1) to set the accelerometer in stand-by mode, to set the performance mode of the KX023 and KX022 to 16 bit resolution, G-range to +/-2g, and enable the availability of new data as an interrupt.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>CNTL1</b>	<b>0x18</b>	<b>0001 1000</b>	<b>0x60</b>	<b>0110 0000</b>

- Write 0x38 to Interrupt Control Register (INC1) to enable physical interrupt pin INT1, to set the polarity of the physical interrupt to active high and to transmit interrupt pulses with a period of 0.03 ms to 0.05 ms in pin(5).

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>INC1</b>	<b>0x1C</b>	<b>0001 1100</b>	<b>0x38</b>	<b>0011 1000</b>

- Write 0x10 to Interrupt Control Register 4 (INC4) to set the Data Ready interrupt to be reported on physical interrupt pin INT1.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>INC4</b>	<b>0x1F</b>	<b>0001 1111</b>	<b>0x10</b>	<b>0001 0000</b>

- Write 0x02 to Data Control Register (ODCNTL) to set the Output Data Rate (ODR) of the accelerometer to 50 Hz. (Note: This is also the default value.)

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>OSCNTL</b>	<b>0x1B</b>	<b>0001 1011</b>	<b>0x02</b>	<b>0000 0010</b>

- Write 0xE0 to Control Register 1 (CNTL1) to set the accelerometer into operating mode (PC1 = 1)

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>CNTL1</b>	<b>0x18</b>	<b>0001 1000</b>	<b>0xE0</b>	<b>1110 0000</b>

- Acceleration data can now be read from the XOUT\_L, XOUT\_H, YOUT\_L, YOUT\_H, ZOUT\_L, and ZOUT\_H registers in 2's complement format.

### 3. Sample Buffer-Full Interrupt via physical hardware interrupt

- Write 0x00 to Control Register 1 (CNTL1) to set the accelerometer in stand-by mode, to set the performance mode of the KX023 and KX022 to 8 bit resolution, and G-range to +/-2g.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>CNTL1</b>	<b>0x18</b>	<b>0001 1000</b>	<b>0x00</b>	<b>0000 0000</b>

- Write 0x38 to Interrupt Control Register (INC1) to enable physical interrupt pin INT1, to set the polarity of the physical interrupt to active high and to transmit interrupt pulses with a period of 0.03 ms to 0.05 ms in pin(5).

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>INC1</b>	<b>0x1C</b>	<b>0001 1100</b>	<b>0x38</b>	<b>0011 1000</b>

- Write 0x40 to Interrupt Control Register 4 (INC4) to set the Buffer Full interrupt to be reported on physical interrupt pin INT1.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>INC4</b>	<b>0x1F</b>	<b>0001 1111</b>	<b>0x40</b>	<b>0100 0000</b>

- Write 0x02 to Data Control Register (ODCNTL) to set the Output Data Rate (ODR) of the accelerometer to 50 Hz. (Note: This is also the default value.)

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>ODCNTL</b>	<b>0x1B</b>	<b>0001 1011</b>	<b>0x02</b>	<b>0000 0010</b>

- Write 0xA0 to Buffer Control Register 2 (BUF\_CTL2) to activate the sample buffer, to set the resolution of the acceleration data samples collected to 8 bits, to enable report of the interrupt status in INS2, and set the operating mode of the sample buffer to FIFO.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>BUF_CTL2</b>	<b>0x3B</b>	<b>0011 1011</b>	<b>0xA0</b>	<b>1010 0000</b>

- Write 0x80 to Control Register 1 (CNTL1) to set the accelerometer into operating mode (PC1 = 1)

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>CNTL1</b>	<b>0x18</b>	<b>0001 1000</b>	<b>0x80</b>	<b>1000 0000</b>

- Once Buffer-Full Interrupt is issued on INT1 pin, acceleration data can then be read from the Buffer Read (BUF\_READ) register at address 0x3F in 2's complement format. Since the resolution of the samples data was set to 8-bit, only the most significant 8 bits of each sample will be stored in the buffer, and recorded in the following order: X\_HIGH, Y\_HIGH, Z\_HIGH with the oldest data point read first as it is a FIFO buffer. The full buffer contains 252 X, Y, Z elements, which corresponds to 84 unique acceleration data samples.

Buffer reading tips:

- The acceleration data can be read from a buffer using multiple-byte read as shown in the Figure 3 below. The register auto-increment feature is disabled when data is read from the Buffer Read register.
- If data is read using single-byte read, it should be read in increments of 3 bytes in 8-bit resolution mode and 6 bytes in 16-bit resolution mode.
- It is very important to follow proper I2C Write-Read sequence as specified in the product specifications. More specifically, the Master should avoid sending the Stop (P) bit at the end of the I2C Write command, and should issue a Repeat Start bit (Sr) at the start of the I2C Read command as show in the Figure 3. Failure of following this sequence may result in reading the same value from the Read Buffer.

Master	S	SAD + W		RA		Sr	SAD + R			ACK		NACK	P
Slave			ACK		ACK			ACK	DATA		DATA		

Term	Definition
S	Start Condition
Sr	Repeated Start Condition
SAD	Slave Address
W	Write Bit
R	Read Bit
ACK	Acknowledge
NACK	Not Acknowledge
RA	Register Address
Data	Transmitted/Received Data
P	Stop Condition

Figure 3: Proper I2C Sequence to Receive Data from the Slave

#### 4. Wake-Up Function via latched physical hardware interrupt

- Write 0x42 to Control Register 1 (CNTL1) to set the accelerometer in stand-by mode, to set the performance mode of the KX023 and KX022 to 16 bit resolution, G-range to +/-2g, and enable the Wake Up (motion detect) function.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>CNTL1</b>	<b>0x18</b>	<b>0001 1000</b>	<b>0x42</b>	<b>0100 0010</b>

- Write 0x06 to Control Register 3 (CNTL3) to set the Output Data Rate of the Wake Up function (motion detection) (OWUF) to 50 Hz.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>CNTL3</b>	<b>0x1A</b>	<b>0001 1010</b>	<b>0x06</b>	<b>0000 0110</b>

- Write 0x7F to Interrupt Control Register 2 (INC2) to define the direction of detected motion for all positive and negative directions: x positive (x+), x negative (x-), y positive (y+), y negative (y-), z positive (z+), z negative (z-) directions.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>INC2</b>	<b>0x1D</b>	<b>0001 1101</b>	<b>0x7F</b>	<b>0111 1111</b>

- Write 0x05 to Interrupt Wake-Up Timer (WUFC) to set the time motion must be present before a wake-up interrupt is set to 0.1 second. The following formula is used:

$$WUFC \text{ (counts)} = \text{Desired Delay Time (sec)} \times OWUF \text{ (Hz)}$$

$$WUFC \text{ (counts)} = 0.1 \times 50 = 5 \text{ counts}$$

Since the desired delay time is 0.1 second and the OWUF was previously set to 50 Hz, then the Wake-Up Timer is 5 counts (0000 0101).

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>WUFC</b>	<b>0x23</b>	<b>0010 0011</b>	<b>0x05</b>	<b>0000 0101</b>

- Write 0x08 to Interrupt Wake-Up Threshold (ATH) to set the level to 0.5g. The following formula is used:

$$WAKEUP\_THRESHOLD \text{ (counts)} = \text{Desired Threshold (g)} \times 16 \text{ (counts/g)}$$

$$WAKEUP\_THRESHOLD \text{ (counts)} = 0.5 \times 16 = 8 \text{ counts}$$

Note that this threshold is differential with respect to the previous reading.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>ATH</b>	<b>0x30</b>	<b>0010 0000</b>	<b>0x08</b>	<b>0000 1000</b>

- Write 0x30 to Interrupt Control Register (INC1) to output the physical interrupt of the previously defined Wake-Up detect function. This value will create an active high and latched interrupt.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>INC1</b>	<b>0x1C</b>	<b>0001 1100</b>	<b>0x30</b>	<b>0011 0000</b>

- Write 0x02 to Interrupt Control Register 4 (INC4) to set the WUF interrupt to be reported on physical interrupt pin INT1.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>INC4</b>	<b>0x1F</b>	<b>0001 1111</b>	<b>0x02</b>	<b>0000 0010</b>

- Write 0xC2 to Control Register 1 (CNTL1) to set the accelerometer in operating mode with the previously defined settings.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>CNTL1</b>	<b>0x18</b>	<b>0001 1000</b>	<b>0xC2</b>	<b>1100 0010</b>

- Monitor the physical interrupt INT1 of the accelerometer, if the acceleration input profile satisfies the criteria previously established for the 0.5g motion detect threshold level in both positive and negative directions of the X, Y, Z axis for more than 0.1 second, then there should be positive latched interrupt present.
- Read Interrupt Release (INT\_REL) register to unlatch (clear) the output interrupt created by the motion detection function.

Register Name	Address	
	Hex	Binary
<b>INT REL</b>	<b>0x17</b>	<b>0001 0111</b>

- Monitor the physical interrupt INT1 of the accelerometer, the positive latched interrupt should not be present. Repeat the last two steps of moving the accelerometer and clearing the interrupt as needed to test this motion detect functionality based on the previous settings.

## 5. Activate Tilt Position Function

- Write 0x41 to Control Register 1 (CNTL1) to set the accelerometer in stand-by mode, to set the performance mode of the KX023 and KX022 to 16 bit resolution, G-range to +/-2g, and enable the Tilt Position function.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>CNTL1</b>	<b>0x18</b>	<b>0001 1000</b>	<b>0x41</b>	<b>0100 0001</b>

- Write 0x01 to Tilt Timer (TILT\_TIMER). Here we assume a 80ms timer will be sufficient (the default Tilt Output Data Rate is 12.5Hz in register CNTL3).

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>TILT_TIMER</b>	<b>0x22</b>	<b>0010 0010</b>	<b>0x01</b>	<b>0000 0001</b>

- Write 0xC1 to Control Register 1 (CNTL1) to set the accelerometer in operating mode with the previously defined settings.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>CNTL1</b>	<b>0x18</b>	<b>0001 1000</b>	<b>0xC1</b>	<b>1100 0001</b>

- Changes to tilt position state will now be reflected in bit 4 of STATUS\_REG (INT bit), bit 0 of INS2 (TPS bit), Current Tilt Position register TSCP, Previous Tilt Position register TSPP, and also on the physical interrupt pin (if enabled).



## 6. Activate Tap/Double Tap Function

- Write 0x44 to Control Register 1 (CNTL1) to set the accelerometer in stand-by mode, to set the performance mode of the KX023 and KX022 to 16 bit resolution, G-range to +/-2g, and enable the Directional Tap function.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>CNTL1</b>	<b>0x18</b>	<b>0001 1000</b>	<b>0x44</b>	<b>0100 0100</b>

- 
- Write 0xC4 to Control Register 1 (CNTL1) to set the accelerometer in operating mode with the previously defined settings.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
<b>CNTL1</b>	<b>0x18</b>	<b>0001 1000</b>	<b>0xC4</b>	<b>1100 0100</b>

- Tap/double-tap events will now be reflected in bit 4 of STATUS\_REG (INT bit), bits 2 and 3 of INS2 (TDTS1 and TDTS0 bits), and INS1 (tap direction) registers, and also on the physical interrupt pin (if enabled).

## Timing Requirements

There are several timing requirements that developers should keep in mind when working with the KX023 and KX022.

I<sup>2</sup>C Clock - The I<sup>2</sup>C Clock can support Fast Mode up to **400 KHz** and High Speed mode up to **3.4 MHz**.

SPI Clock - The SPI Clock can support up to **10MHz**.

Power Up to Communication - After the part is powered up, it takes **10ms** before it is ready for I<sup>2</sup>C communication.

Enable to Valid Outputs - After the part is enabled (PC1 bit in Control Register 1 is asserted), it takes from **3.4 ms** to **80 ms** depending on the ODR before the acceleration outputs are valid. (See KX023 and KX022 Product Specification for details)

Power On Reset Delay - After a Power-On Reset, the part takes **10ms** before it is ready for I<sup>2</sup>C communication.

Software Reset Delay - After a Software Reset, the part takes **2ms** before it is ready for I<sup>2</sup>C communication.

## **Interrupt Configuration**

The physical interrupt has 6 possible configurations, based on two states for each of the three customizable variables located in Interrupt Control Register 1:

### Latched/Pulsed (IEL – bit 3 – 0x08)

- 0 – Latched mode – When an interrupt is triggered, it will remain active on the pin until cleared.
- 1 – Pulse mode – When an interrupt is triggered, it will cause a short (~50µs) pulse on the pin and clear itself.

### Polarity (IEA – bit 4 – 0x10)

- 0 – Active Low – The interrupt pin will normally be HIGH, but will transition to LOW when an interrupt is triggered.
- 1 – Active High – The interrupt pin will normally be LOW, but will transition to HIGH when an interrupt is triggered.

### Enable/Disable (IEN – bit 5 – 0x20)

- 0 – Disabled – Interrupt conditions will not be reflected on the physical interrupt pin.
- 1 – Enabled – Interrupt conditions will be reflected on the physical interrupt pin.

## **A Few Interrupt Tips**

### Read the Interrupt Release Register to Clear

In latched mode, the INT\_REL register must be read in order to clear the physical interrupt pin. This will also clear the Interrupt Source Registers and the INT bit (0x10) in the Status Register.

### Microcontroller/GPIO Interrupt Handling –

GPIO configuration is based solely on the connected hardware. The KX023 and KX022 can be configured to issue interrupts depending on how the GPIO is programmed to catch them (if this is not the case, please contact your Kionix Sales Representative). Generally, when an interrupt is triggered, the developer should take the following steps:

- 1- Disable GPIO interrupt
- 2- Clear GPIO interrupt and generate desired functionality
- 3- Enable GPIO interrupt

These steps should be taken without calling any digital communication transactions if done in an interrupt context, because the operating system or kernel will not allow busy-waiting on an I/O operation during an interrupt service routine.

Interrupt Polling - If physical interrupts are not used, a polling mechanism can be devised, which checks the INT bit in STATUS\_REG. This mechanism should then look at INT\_SRC\_REG2 to determine which engine caused the interrupt and what steps should be taken before clearing the interrupt source information by reading the INT\_REL register.

## Troubleshooting

### All Interrupt Issues

- Make sure the KX023 and KX022 is configured to issue interrupt signals in the way that your GPIO is programmed to handle them.
- An oscilloscope on the physical interrupt pin can be a valuable tool to confirm physical interrupt operation.
- Double check the Tilt Position State Mask bits in Control Register 2 (Tilt Position Function), the axis mask bits in Interrupt Control Register 2 (Wake-up Function), and/or the Tap/Double-Tap Mask bits in Interrupt Control Register 3 (Tap/Double-Tap Function).
- The Tilt Timer, WUF Timer, and TDT Timer(s) are based on their respective Output Data Rates, so make sure the correct cycle time is used when calculating the expected timer length (please refer to the KX023 and KX022 product specification).

### Tilt Interrupt Not Working

- Make sure that the Tilt Position engine is enabled (TPE bit in Control Register 1).
- Try shortening the timer requirements and make sure the next state transition does not occur until after the expiration of the Tilt Timer.
- Try increasing the Tilt Angle to ensure that the engine can see the transition between the X and Y axes and the Z axis (this should not be necessary if using the default value for Tilt Angle, but it's worth looking into if problems continue).

### WUF (Wake Up Function) Interrupt Not Working

- Make sure that the WUF engine is enabled (WUFE bit in Control Register 1).
- Try altering the threshold requirements to achieve desired operation. If the part is waking up too easily, try increasing the threshold. If the interrupt is not firing at all, the threshold may be set too high.
- Try shortening the timer requirements, and make sure the acceleration on an unmasked axis is above the threshold until the expiration of the WUF Timer.

### TDT (Tap/Double-Tap) Interrupt Not Working

- Make sure that the TDT engine is enabled (TDTE bit in Control Register 1).
- Try altering the threshold requirements to achieve desired operation. If the part is generating interrupts too often, perhaps due to a large noise floor created by excessive environmental vibrations, try increasing the performance index low threshold (TTL) and/or reducing the performance index high threshold (TTH). If the interrupt is not firing at all, perhaps the low threshold may be set too high, or the high threshold may be set too low.
- There are many timers in this engine which have to work together closely, so for standard operation if one timer is changed the other timers may need to be changed proportionally.

## **Accelerometer Placement and Orientation**

Placement – It is important to note that the placement of the accelerometer within the target device can have a significant effect on tap/double-tap direction resolution. If tap detection is desired, the part should be placed as far away from the edges of the device housing as possible, with the ideal location being at the target device's center of mass.

Orientation – While it is recommended to align the accelerometer's axes with those of the target device, it will sometimes be desirable or necessary to alter the part's orientation with respect to the device housing. Rotating about the Z axis at intervals of 90 degrees or about the X or Y axes at intervals of 180 degrees should not impact functionality. However, it is highly recommended that the device is not rotated 90 or 270 degrees about the X or Y axes. Due to the asymmetrical nature of the tilt position function, altering the orientation of the Z axis in this manner can cause incorrect screen rotation direction resolution.

## **Accelerometer USB Development Kit**

Kionix offers an Accelerometer USB Development Kit that can be used to quickly begin the development of applications and firmware that incorporate Kionix accelerometers including the KX023 and KX022. The Development Kit provides a common interface to Kionix evaluation boards. For additional information regarding the development kit please refer to Kionix Application Firmware Development Kit user manual. Here is a brief description of the applications and utilities supported by the development kit.

### **SensorScope**

This application allows the user to monitor data coming from the attached sensor. This data can be saved to a file or viewed in real time. With only two verification steps, the application will display a series of graphs representing acceleration with respect to time for each axis. This data can be used to measure the noise of the accelerometer by using the following steps:

- Place the evaluation board on a flat surface in the desired orientation.
- To change the application settings, select *Settings* from the *Edit* menu. On this menu the following settings can be changed:
  - Sampling Rate - The rate at which the software queries the accelerometer for axis data.
  - Realtime Interval - The amount of data the software will buffer and display in real time.
- Select the capture button. The application will begin to capture data immediately. Captured data is written to a file, and will not be viewable until after the capture has finished. The status bar is used to notify the user of a capture in progress.
- The application will continue to collect data until the user clicks the *Stop* button, or the resulting capture file has exceeded the file size limits (~1Gigabyte). We recommend collecting the data for at least 120 seconds.
- Captured data will be saved as a list of comma-separated values (.csv). Each entry in the list is comprised of a time, followed by the raw count for each axis (x, y, and z respectively).
- Select *Save* or *Save As* from the *File* menu to save the file.
- Open the saved file using Excel. Calculate the average of the samples. This gives the noise of the accelerometer in raw counts.

### **SensorCalc**

This application allows the user to test and calculate the zero-g offset and sensitivity parameters of the accelerometer. Once the accelerometer is properly placed relative to the Earth's gravity, simple mouse clicks initiate a series of test sequences that result in the display of raw-count data.

### **SensorMap**

This application allows the user to read and write to specific registers of the accelerometer. The registers and their values are all displayed simultaneously on one color-coded grid.

## **The Kionix Advantage**

Kionix technology provides for X, Y, and Z-axis sensing on a single, silicon chip. One accelerometer can be used to enable a variety of simultaneous features including, but not limited to:

- Hard Disk Drive protection
- Vibration analysis
- Tilt screen navigation
- Sports modeling
- Theft, man-down, accident alarm
- Image stability, screen orientation & scrolling
- Computer pointer
- Navigation, mapping
- Game playing
- Automatic sleep mode

## **Theory of Operation**

Kionix MEMS linear tri-axis accelerometers function on the principle of differential capacitance. Acceleration causes displacement of a silicon structure resulting in a change in capacitance. A signal-conditioning CMOS technology ASIC detects and transforms changes in capacitance into an analog output voltage, which is proportional to acceleration. These outputs can then be sent to a micro-controller for integration into various applications.

For product summaries, specifications, and schematics, please refer to the Kionix MEMS accelerometer product catalog at <http://www.kionix.com/parametric/Accelerometers>.