## Introduction

This application note will help developers quickly implement proof-of-concept designs using the KXG07/KXG08 tri-axis gyroscope and tri-axis accelerometer. Please refer to the KXG07/KXG08 data sheet for additional implementation guidelines. Kionix strives to ensure that our sensor offerings 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 KXG07/KXG08 tri-axis gyroscope and tri-axis accelerometer.

## Circuit Schematic

This section shows recommended wiring for the KXG07/KXG08, based on proven operation of the part. Specific applications may require modifications from these recommendations. Please refer to the KXG07/KXG08 Data Sheets for all pin descriptions.

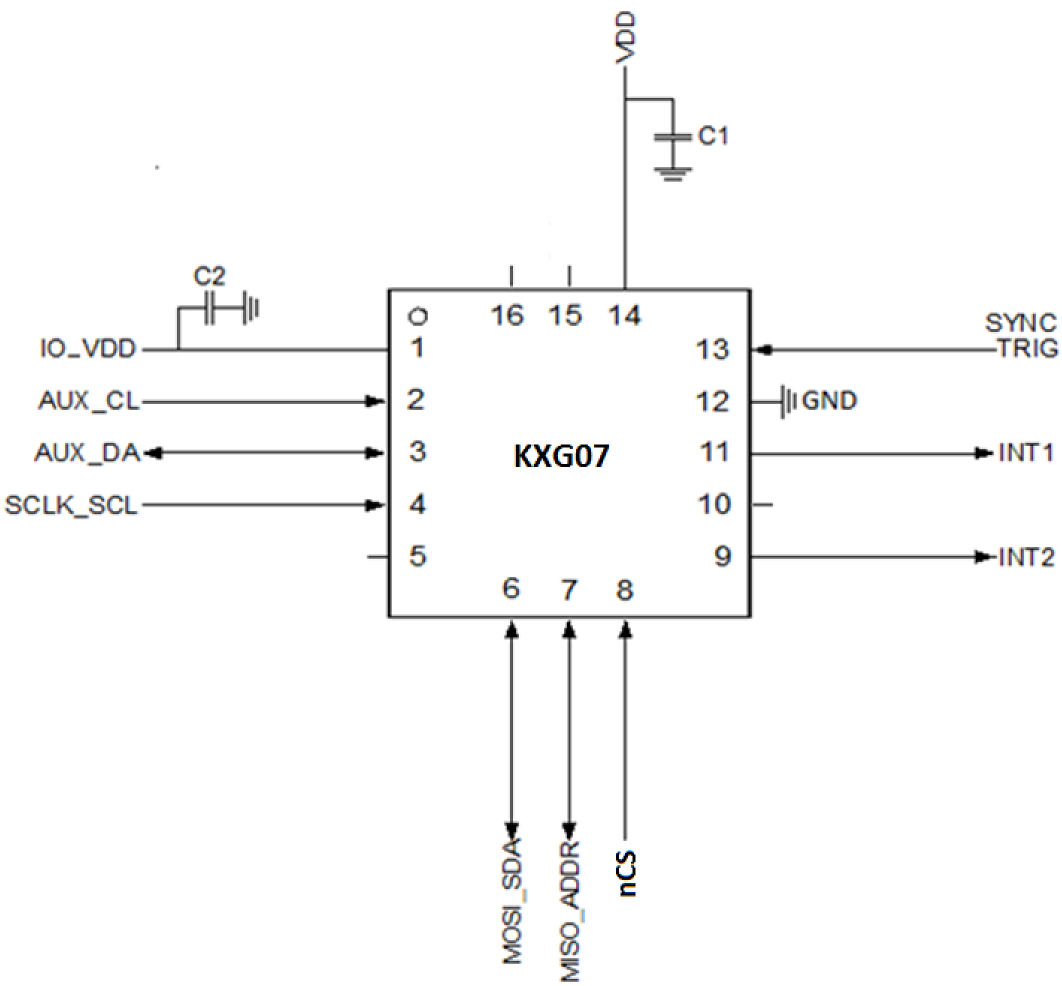


**Figure 1:** KXG07 Application Schematic

36 Thornwood Dr. — Ithaca, NY 14850 USA
Tel: 607-257-1080 — Fax: 607-257-1146
www.kionix.com — info@kionix.com

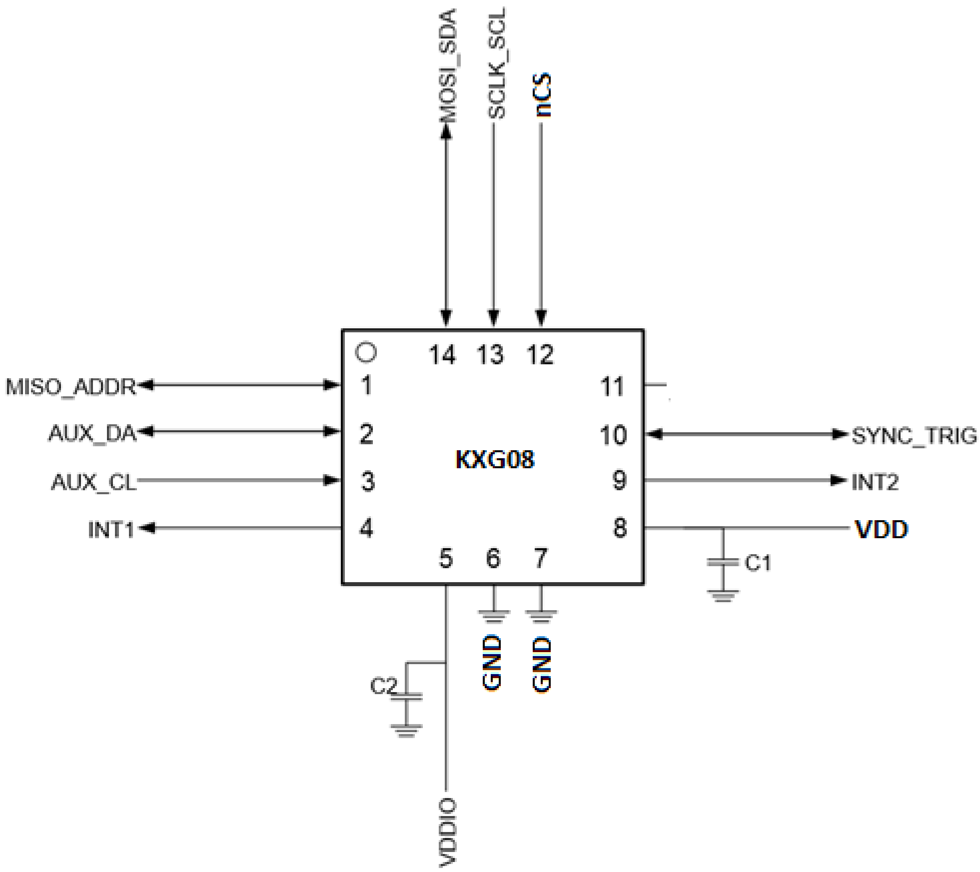


**Figure 2:** KXG08 Application Schematic

## Quick Start Implementation

Here we present several basic ways to initialize the part's accelerometer and gyroscope, as well as, additional features that are offered. 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 set is ready via interrupt
3) read back gyroscope data asynchronously,
4) read back gyroscope data when next data set is ready via interrupt
5) gyroscope sample buffer full interrupt
6) wake up from sleep
7) activate the tilt position function
8) activate the tap/double-tap function
9) activate the free-fall 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. The $I^2C$ 7-bit Slave Address of this part is 0x4E if ADDR=GND or 0x4F if ADDR=IO_VDD.

### 1. Asynchronous Read Back of Acceleration Data

This example enables the accelerometer to start outputting sensor data that can be read from the output registers.

a) Write 0x3F (Reset Value – All Sensors in Standby) to Standby Register (STDBY) to set the accelerometer and gyroscope in stand-by mode (this step must be taken prior to making any register changes).

| Register Name | Address | Value |
|---|---|---|
| STDBY | 0x6E | 0x3F |

b) Write 0xD6 (LPMODEA[8]=1, NAVGA[6:4]=5, ODRA [2:0]=6) to Accelerometer Control Register (ACCEL_ODR) to set the Output Data Rate (ODR) of the accelerometer to 50 Hz, Number of Averages (NAV) to 32, and Low Power Mode (LPMODE) enabled (Note: This is also the default value).

| Register Name | Address | Value |
|---|---|---|
| ACCEL_ODR | 0x3C | 0xD6 |

c) Write 0x00 (ACC_ST_POL[5]=0, ACC_ST[4]=0, ACC_BW[3]=0, ACCFS[1:0]=0) to Accelerometer Range Control Register (ACCEL_CTL) to set the accelerometer range to ±2g.

| Register Name | Address | Value |
|---|---|---|
| ACCEL_CTL | 0x3D | 0x00 |

d) Write 0x3E (ACC_STDBY[0]=0) to Standby Register (STDBY) to set the accelerometer in run mode.

| Register Name | Address | Value |
|---|---|---|
| STDBY | 0x6E | 0x3E |

e) Acceleration data can now be read asynchronously from the ACCEL_XOUT_L, ACCEL_XOUT_H, ACCEL_YOUT_L, ACCEL_YOUT_H, ACCEL_ZOUT_L, and ACCEL_ZOUT_H registers in 2's complement format.

## 2. Synchronous Hardware Interrupt Read Back of Acceleration Data

This example configures the accelerometer to start outputting sensor data synchronously with the data ready interrupt on INT1. When data is ready, data can be read from the output registers.

a) Write 0x3F (Reset Value – All Sensors in Standby) to Standby Register (STDBY) to set the accelerometer and gyroscope in stand-by mode (this step must be taken prior to making any register changes).

| Register Name | Address | Value |
|---|---|---|
| STDBY | 0x6E | 0x3F |

b) Read from the Interrupt Latch Release registers (INT1_L and INT2_L) to clear any outstanding interrupts.

| Register Name | Address | Value |
|---|---|---|
| INT1_L | 0x35 | n/a |
| INT2_L | 0x3B | n/a |

c) Write 0x02 (DRDY_ACC[1]=1) to Interrupt Mask Register 1 (INT_MASK1) to enable the Data Ready Accelerometer Interrupt.

| Register Name | Address | Value |
|---|---|---|
| INT_MASK1 | 0x44 | 0x02 |

d) Write 0x02 (DRDY_ACC_P1[1]=1) to Physical Interrupt Pin INT1 Select Register (INT_PIN_SEL1) to route the Data Ready Accelerometer Interrupt to INT1 pin.

| Register Name | Address | Value |
|---|---|---|
| INT_PIN_SEL1 | 0x41 | 0x02 |

e) Write 0x0C (IEN1[3]=1, IEA1[2]=1, IEL1[1:0]=0) to Interrupt Pin Control Register (INT_PIN_CTL) to configure the physical interrupt pin (INT1). For this example: INT1 pin is enabled, active high, and mode of operation is latched.

| Register Name | Address | Value |
|---|---|---|
| INT_PIN_CTL | 0x40 | 0x0C |

f) Write 0xD6 (LPMODEA[8]=1, NAVGA[6:4]=5, ODRA [2:0]=6) to Accelerometer Control Register (ACCEL_ODR) to set the Output Data Rate (ODR) of the accelerometer to 50 Hz, Number of Averages (NAV) to 32, and Low Power Mode (LPMODE) enabled (Note: This is also the default value).

| Register Name | Address | Value |
|---|---|---|
| ACCEL_ODR | 0x3C | 0xD6 |

g) Write 0x00 (ACC_ST_POL[5]=0, ACC_ST[4]=0, ACC_BW[3]=0, ACCFS[1:0]=0) to Accelerometer Range Control Register (ACCEL_CTL) to set the accelerometer range to ±2g.

| Register Name | Address | Value |
|---|---|---|
| ACCEL_CTL | 0x3D | 0x00 |

h) Write 0x3E (ACC_STDBY[0]=0) to Standby Register (STDBY) to set the accelerometer in run mode.

| Register Name | Address | Value |
|---|---|---|
| STDBY | 0x6E | 0x3E |

i) Acceleration data can now be read synchronously (when INT1 becomes active) from the ACCEL_XOUT_L, ACCEL_XOUT_H, ACCEL_YOUT_L, ACCEL_YOUT_H, ACCEL_ZOUT_L, and ACCEL_ZOUT_H registers in 2's complement format

## 3. Asynchronous Read Back of Gyroscope Data

This example enables the gyroscope to start outputting sensor data that can be read from the output registers.

a) Write 0x3F (Reset Value – All Sensors in Standby) to Standby Register (STDBY) to set the accelerometer and gyroscope in stand-by mode (this step must be taken prior to making any register changes).

| Register Name | Address | Value |
|---|---|---|
| STDBY | 0x6E | 0x3F |

b) Write 0xD6 (LPMODEG[7]=1, NAVGG[6:4]=5, ODRG[3:0]=6) to Gyroscope Control Register (GYRO_ODR) to set the Output Data Rate (ODR) of the gyroscope to 50 Hz, averaging to 32, and low power mode enabled. (Note: This is also the default value.)

| Register Name | Address | Value |
|---|---|---|
| GYRO_ODR | 0x3E | 0xD6 |

c) Write 0x00 (GYRO_BW[3]=0, GYRO_FS[2:0]=0) to Gyroscope Range Control Register (GYRO_CTL) to set the gyroscope velocity range to ±64 degrees/second and bandwidth to ODR/8. (Note: This is also the default value.)

| Register Name | Address | Value |
|---|---|---|
| GYRO_CTL | 0x3F | 0x00 |

d) Write 0x39 (GYRO_FSTART[2]=0, GYRO_STDBY[1]=0) to Standby Register (STDBY) to set the gyroscope in run mode. (Note: Enables gyro, including outputs)

| Register Name | Address | Value |
|---|---|---|
| STDBY | 0x6E | 0x39 |

e) Gyroscope data can now be read from the GYRO_XOUT_L, GYRO_XOUT_H, GYRO_YOUT_L, GYRO_YOUT_H, GYRO_ZOUT_L, and GYRO_ZOUT_H registers in 2's complement format.

## 4. Synchronous Hardware Interrupt Read Back of Gyroscope Data

This example configures the gyroscope to start outputting sensor data synchronously with the data ready interrupt on INT1. When data is ready, data can be read from the output registers.

a) Write 0x3F (Reset Value – All Sensors in Standby) to Standby Register (STDBY) to set the accelerometer and gyroscope in stand-by mode (this step must be taken prior to making any register changes).

| Register Name | Address | Value |
|---|---|---|
| STDBY | 0x6E | 0x3F |

b) Read from the Interrupt Latch Release registers (INT1_L and INT2_L) to clear any outstanding interrupts.

| Register Name | Address | Value |
|---|---|---|
| INT1_L | 0x35 | n/a |
| INT2_L | 0x3B | n/a |

c) Write 0x01 (DRDY_GYRO[0]=1) to Interrupt Mask Register 1 (INT_MASK1) to enable the Data Ready Accelerometer Interrupt.

| Register Name | Address | Value |
|---|---|---|
| INT_MASK1 | 0x44 | 0x01 |

d) Write 0x01 (DRDY_GYRO_P1[0]=1) to Physical Interrupt Pin INT1 Select Register (INT_PIN_SEL1) to route the Data Ready Gyroscope Interrupt to INT1 pin.

| Register Name | Address | Value |
|---|---|---|
| INT_PIN_SEL1 | 0x41 | 0x01 |

e) Write 0x0C (IEN1[3]=1, IEA1[2]=1, IEL1[1:0]=0) to Interrupt Pin Control Register (INT_PIN_CTL) to configure the physical interrupt pin (INT1). For this example: INT1 pin is enabled, active high, and mode of operation is latched.

| Register Name | Address | Value |
|---|---|---|
| INT_PIN_CTL | 0x40 | 0x0C |

f) Write 0xD6 (LPMODEG[7]=1, NAVGG[6:4]=5, ODRG[3:0]=6) to Gyroscope Control Register (GYRO_ODR) to set the Output Data Rate (ODR) of the gyroscope to 50 Hz, averaging to 32, and low power mode enabled. (Note: This is also the default value.)

| Register Name | Address | Value |
|---|---|---|
| GYRO_ODR | 0x3E | 0xD6 |

g) Write 0x00 (GYRO_BW[3]=0, GYRO_FS[2:0]=0) to Gyroscope Range Control Register (GYRO_CTL) to set the gyroscope velocity range to ±64 degrees/second and bandwidth to ODR/8. (Note: This is also the default value.)

| Register Name | Address | Value |
|---|---|---|
| GYRO_CTL | 0x3F | 0x00 |

h) Write 0x39 (GYRO_FSTART[2]=0, GYRO_STDBY[1]=0) to Standby Register (STDBY) to set the gyroscope in run mode. (Note: Enables gyro, including outputs)

| Register Name | Address | Value |
|---|---|---|
| STDBY | 0x6E | 0x39 |

i) Gyroscope data can now be read synchronously (when INT1 becomes active) from the GYRO_XOUT_L, GYRO_XOUT_H, GYRO_YOUT_L, GYRO_YOUT_H, GYRO_ZOUT_L, and GYRO_ZOUT_H registers in 2's complement format.

## 5. Gyro Sample Buffer-Full Interrupt via physical hardware interrupt

This example configures the sensor to utilize the internal FIFO buffer to store gyroscope data, but can be configured to include a combination of gyroscope, acceleration and temperature data. Buffers can be used to stream data, collect data before an event, and/or collect data after an event; without putting strain on a host system to continuously monitor/read sensor data. When the buffer is full, data can be read from the output buffer registers (BUF_READ). The interrupt will be generated on INT1 pin.

a) Write 0x3F (Reset Value – All Sensors in Standby) to Standby Register (STDBY) to set the accelerometer and gyroscope in stand-by mode (this step must be taken prior to making any register changes).

| Register Name | Address | Value |
|---|---|---|
| STDBY | 0x6E | 0x3F |

b) Read from the Interrupt Latch Release registers (INT1_L and INT2_L) to clear any outstanding interrupts.

| Register Name | Address | Value |
|---|---|---|
| INT1_L | 0x35 | n/a |
| INT2_L | 0x3B | n/a |

c) Write 0x80 (BFIE[7]=1) to Interrupt Mask Register 1 (INT_MASK1) to enable the Buffer Full Interrupt.

| Register Name | Address | Value |
|---|---|---|
| INT_MASK1 | 0x44 | 0x80 |

d) Write 0x80 (BFI_P1[7]=1) to Physical Interrupt Pin INT1 Select Register (INT_PIN_SEL1) to route the Buffer Full Interrupt to INT1 pin.

| Register Name | Address | Value |
|---|---|---|
| INT_PIN_SEL1 | 0x41 | 0x80 |

e) Write 0x0C (IEN1[3]=1, IEA1[2]=1, IEL1[1:0]=0) to Interrupt Pin Control Register (INT_PIN_CTL) to configure the physical interrupt pin (INT1). For this example: INT1 pin is enabled, active high, and mode of operation is latched.

| Register Name | Address | Value |
|---|---|---|
| INT_PIN_CTL | 0x40 | 0x0C |

f) Write 0xD6 (LPMODEG[7]=1, NAVGG[6:4]=5, ODRG[3:0]=6) to Gyroscope Control Register (GYRO_ODR) to set the Output Data Rate (ODR) of the gyroscope to 50 Hz, averaging to 32, and low power mode enabled. (Note: This is also the default value.)

| Register Name | Address | Value |
|---|---|---|
| GYRO_ODR | 0x3E | 0xD6 |

g) Write 0x00 (GYRO_BW[3]=0, GYRO_FS[2:0]=0) to Gyroscope Range Control Register (GYRO_CTL) to set the gyroscope velocity range to ±64 degrees/second and bandwidth to ODR/8. (Note: This is also the default value.)

| Register Name | Address | Value |
|---|---|---|
| GYRO_CTL | 0x3F | 0x00 |

h) Write 0x07 (BUF_GYR_X[2]=1, BUF_GYR_Y[1]=1, BUF_GYR_Z[0]=1) to the Buffer Control Register 1 (BUF_CTL1). This will place the gyroscope data into the sample buffer in wake mode.

| Register Name | Address | Value |
|---|---|---|
| BUF_CTL1 | 0x76 | 0x07 |

i) Write 0x80 (BUFE[7]=1, BUF_M[1:0]=0) to the Buffer Enable Register (BUF_EN) to enable buffering of data in FIFO mode. For this example, the buffer is being enabled and using FIFO mode.

| Register Name | Address | Value |
|---|---|---|
| BUF_EN | 0x78 | 0x80 |

j) Write 0x39 (GYRO_FSTART[2]=0, GYRO_STDBY[1]=0) to Standby Register (STDBY) to set the gyroscope in run mode. (Note: Enables gyro, including outputs)

| Register Name | Address | Value |
|---|---|---|
| STDBY | 0x6E | 0x39 |

k) Once Buffer-Full Interrupt is issued on INT1 pin, gyroscope data can then be read from the Buffer Read (BUF_READ) register at address 0x7B in 2's complement format. Since the resolution of the samples data is 16-bit, data will be stored in the buffer, and recorded in the following order: X_LOW, X_HIGH, Y_LOW, Y_HIGH, Z_LOW, Z_HIGH with the oldest data point read first as it is a FIFO buffer. The full buffer contains 4104 X, Y, Z elements, which corresponds to 684 unique gyroscope data samples.

| Register Name | Address | Value |
|---|---|---|
| BUF_READ | 0x7B | n/a |

Buffer reading tips:

a) The gyroscope data can be read from a buffer using multiple-byte read as shown in Figure 3 the below. The register auto-increment feature is disabled when data is read from the Buffer Read register.
b) If data is read using single-byte read, it should be read in increments of 6 bytes in 16-bit resolution mode.
c) 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

## 6. Wakeup from Sleep Interrupt via physical hardware interrupt

This example configures the sensor to utilize the embedded Wake-up from Sleep feature and configured to generate the interrupt on INT1 pin. The interrupt engine can be configured by the user to report when qualified changes detected by the acceleration occur, using the accelerometer. Optionally, the user has the ability to enable/disable specific accelerometer axes and specific directions, as well as to specify the delay time. An example use cause for the engine would be to detect motion on any axis to signal an event to wake up or put back to sleep the KXG07/KXG08 or other devices.

a) Write 0x3F (Reset Value – All Sensors in Standby) to Standby Register (STDBY) to set the accelerometer and gyroscope in stand-by mode (this step must be taken prior to making any register changes).

| Register Name | Address | Value |
|---|---|---|
| STDBY | 0x6E | 0x3F |

b) Read from the Interrupt Latch Release registers (INT1_L and INT2_L) to clear any outstanding interrupts.

| Register Name | Address | Value |
|---|---|---|
| INT1_L | 0x35 | n/a |
| INT2_L | 0x3B | n/a |

c) Write 0x20 (WUF_P1[5]=1) to Physical Interrupt Pin INT1 Select Register (INT_PIN_SEL1) to route the Wake-up Function Interrupt to INT1 pin.

| Register Name | Address | Value |
|---|---|---|
| INT_PIN_SEL1 | 0x41 | 0x20 |

d) Write 0x0C (IEN1[3]=1, IEA1[2]=1, IEL1[1:0]=0) to Interrupt Pin Control Register (INT_PIN_CTL) to configure the physical interrupt pin (INT1). For this example: INT1 pin is enabled, active high, and mode of operation is latched.

| Register Name | Address | Value |
|---|---|---|
| INT_PIN_CTL | 0x40 | 0x0C |

e) Write 0x06 (TH_MODE[5]=0, OWUF[2:0]=6) to the Wake Sleep Control Register 1 (WAKE_SLEEP_CTL1) to configure part of the wake/sleep control. For this example: The WUF threshold mode will be “absolute” (where the ASIC compares the current output and previous outputs to the threshold) and ODR for wake up (motion detection) will be set to 50 Hz.

| Register Name | Address | Value |
|---|---|---|
| WAKE_SLEEP_CTL1 | 0x4E | 0x06 |

f) Write 0x66 (WUF_EN[6]=1, MAN_SLEEP[5]=1, OBTS[2:0]=6) to the Wake Sleep Control Register 2 (WAKE_SLEEP_CTL2) to configure the wake/sleep control. For this example: Wake-up function will be enabled, the sensor will be forced to sleep, and ODR for back to sleep (BTS) will be set to 50 Hz (BTS ODR unused in this example).

| Register Name | Address | Value |
|---|---|---|
| WAKE_SLEEP_CTL2 | 0x4F | 0x66 |

g) Write 0x01 (ATH[7:0] = 1) to the Wake Up Function Active Threshold Register (WUF_TH) to configure the threshold value that will be used for motion detection. For this example: Since Resolution = 62.5 mg/LSB for FS < +/- 16 g the formula to calculate the threshold would be: 0.0625g / 0.0625g/count = 1 (0x01).

| Register Name | Address | Value |
|---|---|---|
| WUF_TH | 0x4B | 0x01 |

h) Write 0x01 (WUTH[7:0] = 1) to the Wake Up Function Counter Register (WUF_COUNTER) to configure the time motion value that must be present to detect motion. For this example: Every count is 1/OWUF, Since OWUF = 50Hz, the formula to time period per count would be 0.20 ms / count. We will use 20 ms for this example so we set this counter to 1.

| Register Name | Address | Value |
|---|---|---|
| WUF_COUNTER | 0x4C | 0x01 |

i) Write 0x3E (ACC_STDBY[0]=0) to Standby Register (STDBY) to set the accelerometer in run mode.

| Register Name | Address | Value |
|---|---|---|
| STDBY | 0x6E | 0x3E |

j) Since the device is forced to “Sleep”, motion (i.e. shake the sensor) can be now be detected by monitoring INT1.

## 7. Activate Tilt Position Function

a) Write 0x3F (Reset Value – All Sensors in Standby) to Standby Register (STDBY) to set the accelerometer and gyroscope in stand-by mode (this step must be taken prior to making any register changes).

| Register Name | Address | Value |
|---|---|---|
| STDBY | 0x6E | 0x3F |

b) Read from the Interrupt Latch Release registers (INT1_L and INT2_L) to clear any outstanding interrupts.

| Register Name | Address | Value |
|---|---|---|
| INT1_L | 0x35 | n/a |
| INT2_L | 0x3B | n/a |

c) Write 0xD6 (LPMODEA[8]=1, NAVGA[6:4]=5, ODRA [2:0]=6) to Accelerometer Control Register (ACCEL_ODR) to set the Output Data Rate (ODR) of the accelerometer to 50 Hz, Number of Averages (NAV) to 32, and Low Power Mode (LPMODE) enabled (Note: This is also the default value).

| Register Name | Address | Value |
|---|---|---|
| ACCEL_ODR | 0x3C | 0xD6 |

d) Write 0x00 (ACC_ST_POL[5]=0, ACC_ST[4]=0, ACC_BW[3]=0, ACCFS[1:0]=0) to Accelerometer Range Control Register (ACCEL_CTL) to set the accelerometer range to ±2g. (Note: This is also the default value)

| Register Name | Address | Value |
|---|---|---|
| ACCEL_CTL | 0x3D | 0x00 |

e) Write 0x08 (TPS_P1[3]=1) to Physical Interrupt Pin INT1 Select Register (INT_PIN_SEL3) to route the Tilt Position Interrupt to INT1 pin.

| Register Name | Address | Value |
|---|---|---|
| INT_PIN_SEL3 | 0x43 | 0x08 |

f) Write 0x0C (IEN1[3]=1, IEA1[2]=1, IEL1[1:0]=0) to Interrupt Pin Control Register (INT_PIN_CTL) to configure the physical interrupt pin (INT1). For this example: INT1 pin is enabled, active high, and mode of operation is latched.

| Register Name | Address | Value |
|---|---|---|
| INT_PIN_CTL | 0x40 | 0x0C |

g) Write 0x3F to the Interrupt Mask 4 Register (INT_MASK4) to enable all axis and directions that can generate an interrupt.

| Register Name | Address | Value |
|---|---|---|
| INT_MASK4 | 0x47 | 0x3F |

h) Write 0x14 to the Hysteresis Register (HYST_SET) to set the hysteresis that is placed in between Screen Rotation states. (Kionix ships from the factory with HYST_SET set to ±15° of hysteresis).

| Register Name | Address | Value |
|---|---|---|
| HYST_SET | 0x5B | 0x14 |

i) Write 0x2A to the Tilt Angle High Limit (TILT_ANGLE_HL).

| Register Name | Address | Value |
|---|---|---|
| TILT_ANGLE_HL | 0x5C | 0x2A |

j) Write 0x0C to the Tilt Angle High Limit (TILT_ANGLE_HL).

| Register Name | Address | Value |
|---|---|---|
| TILT_ANGLE_LL | 0x5D | 0x0C |

k) Write 0x03 to the Tilt/Tap ODR Register (TILT_ANGLE_HL) to set the Tilt ODR to 100Hz (optional).

| Register Name | Address | Value |
|---|---|---|
| TILT_TAP_ODR | 0x5F | 0x03 |

l) Write 0x01 (TSC[7:0]=0) 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 TILT_TAP_ODR).

| Register Name | Address | Value |
|---|---|---|
| TILT_TIMER | 0x5E | 0x01 |

m) Write 0x08 (TILT_EN[3]=1) to Control Register 1 (CTL_REG_1) to enable the Tile Position Engine.

| Register Name | Address | Value |
|---|---|---|
| CTL_REG_1 | 0x6B | 0x08 |

n) Write 0x3E (ACC_STDBY[0]=0) to Standby Register (STDBY) to set the accelerometer in run mode.

| Register Name | Address | Value |
|---|---|---|
| STDBY | 0x6E | 0x3E |

o) Change to the tilt position state will now be reflected via the physical interrupt pin INT1, INT1[7] of Status Register 1 (STATUS1), INT1_TPS[3] of Interrupt 1 Source Register 4 (INT1_SRC4), Current Tilt Position Register (TSCP), and Previous Tilt Position Register (TSPP).

## 8. Activate Tap/Double Tap Function

a) Write 0x3F (Reset Value – All Sensors in Standby) to Standby Register (STDBY) to set the accelerometer and gyroscope in stand-by mode (this step must be taken prior to making any register changes).

| Register Name | Address | Value |
|---|---|---|
| STDBY | 0x6E | 0x3F |

b) Read from the Interrupt Latch Release registers (INT1_L and INT2_L) to clear any outstanding interrupts.

| Register Name | Address | Value |
|---|---|---|
| INT1_L | 0x35 | n/a |
| INT2_L | 0x3B | n/a |

c) Write 0xD6 (LPMODEA[8]=1, NAVGA[6:4]=5, ODRA [2:0]=6) to Accelerometer Control Register (ACCEL_ODR) to set the Output Data Rate (ODR) of the accelerometer to 50 Hz, Number of Averages (NAV) to 32, and Low Power Mode (LPMODE) enabled (Note: This is also the default value).

| Register Name | Address | Value |
|---|---|---|
| ACCEL_ODR | 0x3C | 0xD6 |

d) Write 0x00 (ACC_ST_POL[5]=0, ACC_ST[4]=0, ACC_BW[3]=0, ACCFS[1:0]=0) to Accelerometer Range Control Register (ACCEL_CTL) to set the accelerometer range to ±2g. (Note: This is also the default value)

| Register Name | Address | Value |
|---|---|---|
| ACCEL_CTL | 0x3D | 0x00 |

e) Write 0x02 (TDS_P1[1]=1) to Physical Interrupt Pin INT1 Select Register (INT_PIN_SEL3) to route the Tap/Double Tap Interrupt to INT1 pin.

| Register Name | Address | Value |
|---|---|---|
| INT_PIN_SEL3 | 0x43 | 0x02 |

f) Write 0x0C (IEN1[3]=1, IEA1[2]=1, IEL1[1:0]=0) to Interrupt Pin Control Register (INT_PIN_CTL) to configure the physical interrupt pin (INT1). For this example: INT1 pin is enabled, active high, and mode of operation is latched.

| Register Name | Address | Value |
|---|---|---|
| INT_PIN_CTL | 0x40 | 0x0C |

g) Write 0x3F to the Interrupt Mask 3 Register (INT_MASK3) to enable all axis and direction of tap/double tap that can cause an interrupt.

| Register Name | Address | Value |
|---|---|---|
| INT_MASK3 | 0x46 | 0x3F |

h) Write 0x18 to the Tilt/Tap ODR Control Register to set the "*Tap*" ODR to 100Hz.

| Register Name | Address | Value |
|---|---|---|
| TILT_TAP_ODR | 0x5F | 0x18 |

i) Write 0x03 to the Tap/Double Tap Reporting Control Register (TDTRC) to update interrupts on both single and double tap events.

| Register Name | Address | Value |
|---|---|---|
| TDTRC | 0x60 | 0x03 |

j) Write 0x78 to the Tap/Double Tap Counter Register (TDTC) to set the counter to 0.3 seconds (Kionix recommended, see specification for more information).

| Register Name | Address | Value |
|---|---|---|
| TDTC | 0x61 | 0x78 |

k) Write 0xCB to the Tap/Jerk Threshold High Register (Kionix recommended, see specification for more information).

| Register Name | Address | Value |
|---|---|---|
| TTH | 0x62 | 0xCB |

l) Write 0x1A to the Tap/Jerk Threshold Low Register (Kionix recommended, see specification for more information).

| Register Name | Address | Value |
|---|---|---|
| TTL | 0x63 | 0x1A |

m) Write 0xA2 to the FTD Register (Kionix recommended, see specification for more information).

| Register Name | Address | Value |
|---|---|---|
| FTD | 0x64 | 0xA2 |

n) Write 0x24 to the STD Register (Kionix recommended, see specification for more information).

| Register Name | Address | Value |
|---|---|---|
| STD | 0x65 | 0x24 |

o) Write 0x28 to the TLT Register (Kionix recommended, see specification for more information).

| Register Name | Address | Value |
|---|---|---|
| TLT | 0x66 | 0x28 |

p) Write 0xA0 to the TWS Register (Kionix recommended, see specification for more information).

| Register Name | Address | Value |
|---|---|---|
| TWS | 0x67 | 0xA0 |

q) Write 0x04 (TAP_EN[2]=1) to Control Register 1 (CTL_REG_1) to enable the Tap/Double Tap Engine.

| Register Name | Address | Value |
|---|---|---|
| CTL_REG_1 | 0x6B | 0x04 |

r) Write 0x3E (ACC_STDBY[0]=0) to Standby Register (STDBY) to set the accelerometer in run mode.

| Register Name | Address | Value |
|---|---|---|
| STDBY | 0x6E | 0x3E |

s) Tap/double-tap events will now be reflected via the physical interrupt pin INT1, INT1[7] of Status Register 1 (STATUS1), or INT1_TDTS[2:1] of Interrupt 1 Source Register 4 (INT1_SRC4).

## 9. Activate Free-fall Function

a) Write 0x3F (Reset Value – All Sensors in Standby) to Standby Register (STDBY) to set the accelerometer and gyroscope in stand-by mode (this step must be taken prior to making any register changes).

| Register Name | Address | Value |
|---|---|---|
| STDBY | 0x6E | 0x3F |

b) Read from the Interrupt Latch Release registers (INT1_L and INT2_L) to clear any outstanding interrupts.

| Register Name | Address | Value |
|---|---|---|
| INT1_L | 0x35 | n/a |
| INT2_L | 0x3B | n/a |

c) Write 0xD6 (LPMODEA[8]=1, NAVGA[6:4]=5, ODRA [2:0]=6) to Accelerometer Control Register (ACCEL_ODR) to set the Output Data Rate (ODR) of the accelerometer to 50 Hz, Number of Averages (NAV) to 32, and Low Power Mode (LPMODE) enabled (Note: This is also the default value).

| Register Name | Address | Value |
|---|---|---|
| ACCEL_ODR | 0x3C | 0xD6 |

d) Write 0x00 (ACC_ST_POL[5]=0, ACC_ST[4]=0, ACC_BW[3]=0, ACCFS[1:0]=0) to Accelerometer Range Control Register (ACCEL_CTL) to set the accelerometer range to ±2g. (Note: This is also the default value)

| Register Name | Address | Value |
|---|---|---|
| ACCEL_CTL | 0x3D | 0x00 |

e) Write 0x01 (FFS_P1[0]=1) to Physical Interrupt Pin INT1 Select Register (INT_PIN_SEL3) to route the Free Fall Interrupt to INT1 pin.

| Register Name | Address | Value |
|---|---|---|
| INT_PIN_SEL3 | 0x43 | 0x01 |

f) Write 0x0C (IEN1[3]=1, IEA1[2]=1, IEL1[1:0]=0) to Interrupt Pin Control Register (INT_PIN_CTL) to configure the physical interrupt pin (INT1). For this example: INT1 pin is enabled, active high, and mode of operation is latched.

| Register Name | Address | Value |
|---|---|---|
| INT_PIN_CTL | 0x40 | 0x0C |

g) Write 0x09 to the Free Fall Threshold (FFTH). This gives the option to define the acceleration threshold value where 256 counts cover the g range of the accelerometer. This value is compared to the top 8 bits of the accelerometer 8g output.

| Register Name | Address | Value |
|---|---|---|
| FFTH | 0x68 | 0x09 |

h) Write 0x05 to the Free Fall Counter Register (FFC). Every count is calculated as 1/ODR delay period. Per the next step, with 100Hz ODR for the Free Fall Engine and desiring a delay of 0.05 seconds: 0.05 seconds * 1/100Hz = 5.

| Register Name | Address | Value |
|---|---|---|
| FFC | 0x69 | 0x05 |

i) Write 0x83 (FFIE[7]=1, ULMODE[6]=0, DCRM[3]=0, OFFI[2:0]=3) to the Free-Fall Control Register (FFCNTL), which will enable the Free fall engine, set Free-fall interrupt latch control, count up/down debounce methodology, and the Output Data Rate (ODR) to 100Hz (optional).

| Register Name | Address | Value |
|---|---|---|
| FFCTL | 0x6A | 0x83 |

j) Write 0x3E (ACC_STDBY[0]=0) to Standby Register (STDBY) to set the accelerometer in run mode.

| Register Name | Address | Value |
|---|---|---|
| STDBY | 0x3E | 0x3E |

k) Free-fall events will now be reflected via the physical interrupt pin INT1, INT1[7] of Status Register 1 (STATUS1), or INT1_FFS[0] of Interrupt 1 Source Register 4 (INT1_SRC4).

## Timing Requirements

There are several timing requirements that developers should keep in mind when working with the KXG07/KXG08.

I²C Clock - The I²C 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**.

## Interrupt Configuration

There are two (2) available physical interrupts. Each has eight (8) possible configurations, based on two (2) states for both enable/disable and polarity and four (4) states for latched/pulsed configuration for the Interrupt Pin Control Register (INT_PIN_CTL):

Enable/Disable (IEN2[7], IEN1[3])

- 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.

Polarity (IEA2[6], IEA1[2])

- 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.

Latched/Pulsed (IEL2[5:4], IEL1[1:0])

- 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.
- 2 – Pulse mode – When an interrupt is triggered, it will cause a short (~200μs) pulse on the pin and clear itself.
- 3 – Real time mode – When an interrupt is triggered, it will only remain asserted as long as the underlying interrupt conditions exist.

There are also three (3) physical interrupt pin select registers (INT_PIN_SEL1, INT_PIN_SEL2, and INT_PIN_SEL3).

INT_PIN_SEL1 and INT_PIN_SEL2 controls the physical interrupt pins INT1 and INT2, respectively. These select registers route the corresponding interrupt to the pins:

| | |
|---|---|
| BFI_P[7] | – Buffer Full |
| WMI_P[6] | – Watermark |
| WUF_P[5] | – Wake Up Function |
| BTS_P[4] | – Back To Sleep |

DRDY_AUX2[3] – Data Ready Aux2
DRDY_AUX1[2] – Data Ready Aux1
DRDY_ACC[1] – Data Ready Accelerometer
DRDY_GYRO[0] – Data Ready Gyroscope

[By Default: All interrupts (noted above) are routed only to INT1]

- 0 – Disabled – Associated interrupt is not routed to INT1 / INT2 pin
- 1 – Enabled – Associated interrupt is routed to INT1 / INT2 pin

INT_PIN_SEL3 controls both physical interrupt pins INT1 and INT2, but for the Digital Engine Interrupts (Tilt, Tap/Double Tap, and Free-Fall). These select registers route the corresponding interrupt to the pins:

TPS_P2[7] – Tile Position Interrupt for INT2 pin
TDS_P2[5] – Tap/Double Tap Interrupt for INT2 pin
FFS_PS[4] – Free Fall Interrupt for INT2 pin
TPS_P1[3] – Tile Position Interrupt for INT1 pin
TDS_P1[1] – Tap/Double Tap Interrupt for INT1 pin
FFS_P1[0] – Free Fall Interrupt for INT1 pin

[By Default: All interrupts (noted above) are not routed (disabled)]

- 0 – Disabled – Associated interrupt is not routed to INT1 / INT2 pin
- 1 – Enabled – Associated interrupt is routed to INT1 / INT2 pin

## A Few Interrupt Tips

Read the Interrupt Release Register to Clear

In latched mode, the INT1_REL/INT2_REL registers must be read in order to clear the physical interrupt pin. This will also clear the Interrupt Source Registers and the particular INT bit in the Interrupt Source Register.

Microcontroller/GPIO Interrupt Handling –

GPIO configuration is based solely on the connected hardware. The KXG07/KXG08 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 INT1/INT2 bit in INT1_SRC1/INT2_SRC1 registers. If reading acceleration or gyroscope data, the status bit is also cleared if data is read from the respective output registers. If using WUF, this mechanism should then look at INT1_SRC2/INT2_SRC2 to determine which direction caused the interrupt and what steps should be taken before clearing the interrupt source information by reading the INT1_L/INT2_L registers.

## Troubleshooting

1. **All Interrupt Issues**
   - Make sure the KXG07/KXG08 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 INT_MASK1, INT_MASK2, INT_MASK3, and INT_MASK4 bits in the Interrupt Mask Registers

2. **Accelerometer Data Ready Interrupt Not Working**
   - Make sure that the Accelerometer Data Ready interrupt is enabled (DRDY_ACC) in the Interrupt Mask Register 1 (INT_MASK1).
   - Make sure that the Accelerometer Data Ready interrupt signal is routed to the appropriate physical interrupt pin INT1/INT2 (INT_PIN1_SEL/ INT_PIN2_SEL).

3. **Gyroscope Data Ready Interrupt Not Working**
   - Make sure that the Gyroscope Data Ready interrupt is enabled (DRDY_GYR0) in the Interrupt Mask Register 1 (INT_MASK1).
   - Make sure that the Gyroscope Data Ready interrupt signal is routed to the appropriate physical interrupt pin INT1/INT2 (INT_PIN1_SEL/ INT_PIN2_SEL).

4. **Buffer Full Interrupt Not Working**
   - Make sure that the Buffer Full interrupt is enabled (BFIE) in the Interrupt Mask Register 1 (INT_MASK1).
   - Make sure that the Buffer Full interrupt signal is routed to the appropriate physical interrupt pin INT1/INT2 (INT_PIN1_SEL/ INT_PIN2_SEL).
   - Make sure gyroscope data is being placed into the sample buffer by configuring Buffer Control Register 1 (BUF_CTL1).
   - Make sure the Buffer Enable Register (BUF_EN) is enabled to buffer data.

5. **Active Tilt Position Interrupt Not Working**
   - Make sure the correct axis and directional settings are enabled in the Interrupt Mask Register 4 (INT_MASK4).
   - Make sure Tilt Position Interrupt (TPS_P1) is routed to INT1 in the INT_PIN_SEL3 register.
   - Make sure that the Tile Position Engine (TILT_EN) is enabled in the CTL_REG_1 register.
   - Try altering the threshold requirements to achieve desired operation. See HYST_SET, TILT_ANGLE_HL, TILT_ANGLE_LL, and TILT_TIMER for more information.
   - Ensure the accelerometer is in “run mode”.

6. **Active Tap/Double Tap Interrupt Not Working**
    - Make sure the correct axis and directional settings are enabled in the Interrupt Mask Register 3 (INT_MASK3).
    - Make sure Tap/Double Tap Interrupt (TDS_P1) is routed to INT1 in the INT_PIN_SEL3 register.
    - Make sure that the Tile Position Engine (TAP_EN) is enabled in the CTL_REG_1 register. See TTH and TTL for more information.
    - Ensure the accelerometer is in “run mode”.

7. **Active Free-fall Interrupt Not Working**
    - Make sure Free-Fall Interrupt (FFS_P1) is routed to INT1 in the INT_PIN_SEL3 register.
    - Make sure the Free-Fall Engine is enabled in the FFCNTL register.
    - Try altering the threshold/counter requirements to achieve desired operation. See FFC and FFTH for more information.
    - Ensure the accelerometer is in “run mode”.

## Placement and Orientation

Placement – It is important to note that the placement of the accelerometer/gyroscope within the target device can have a significant effect on determining orientation/direction (accelerometer) and angular velocity (gyroscope). If orientation/motion detection is desired (accelerometer), 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. If angular velocity is desired (gyroscope), the ideal location would be the target’s device center of mass and in a location with the least vibration (noise).

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. In regards to the gyroscope, orientation does not matter, due to the fact that it is measuring angular velocity.

## The Kionix Advantage

Kionix technology provides 6 Degrees-of-Freedom inertial sensor system on a single, silicon chip, which is designed to strike a balance between current consumption and noise performance with excellent bias stability over temperature. A gyroscope 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
Remote controls
Toys

## Theory of Operation

During operation, the gyroscope sensor elements are forced into vibration. When angular velocities are applied about the sensing axes, vibration is transferred to sensing elements, causing capacitance changes at the sensor electrodes. Acceleration sensing is based on the principle of a differential capacitance arising from acceleration-induced motion of the sense element, which utilizes common mode cancellation to decrease errors from process variation, temperature, and environmental stress. Capacitance changes are amplified and converted into digital signals which are processed by a dedicated digital signal processing unit. The digital signal processor applies filtering, bias and sensitivity adjustment, as well as temperature compensation. The DSP also feeds back the driving signal to ensure the proper sensor excitation.

For product summaries, specifications, and schematics, please refer to the Kionix MEMS accelerometer product catalog at [http://www.kionix.com/parametric/6-Axis Combo Parts And 9-Axis Solutions](http://www.kionix.com/parametric/6-Axis Combo Parts And 9-Axis Solutions)



Kionix