Introduction

This application note will help developers quickly implement proof-of-concept designs using the KXG03 tri-axis gyroscope and tri-axis accelerometer. Please refer to the KXG03 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 KXG03 tri-axis gyroscope and tri-axis accelerometer.

Circuit Schematic

This section shows recommended wiring for the KXG03, based on proven operation of the part. Specific applications may require modifications from these recommendations. Please refer to the KXG03 Data Sheet for all pin descriptions.

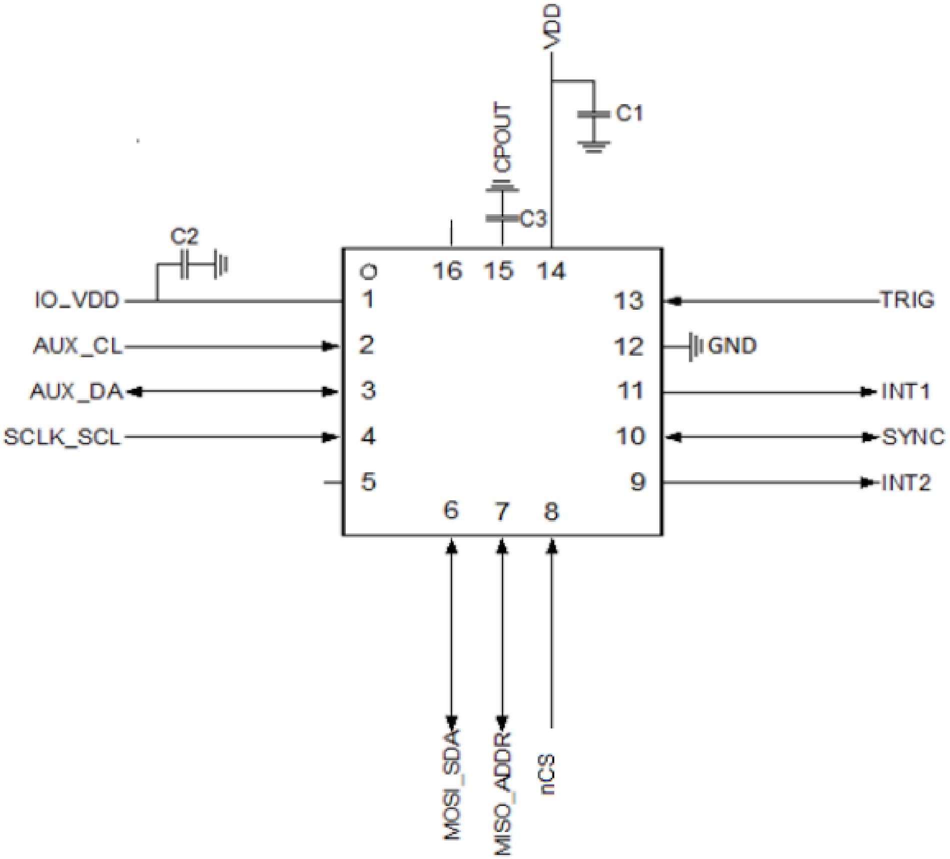


Figure 1: KXG03 Application Schematic

Quick Start Implementation

Here we present several basic ways to initialize the part's accelerometer and gyroscope. 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. 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

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

a) Write 0xEF (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	
	Hex	Binary	Hex	Binary
STDBY	0x43	0100 0011	0xEF	1110 1111

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

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
ACCEL_ODR_WAKE	0x3E	0011 1110	0xD6	1101 0110

c) Write 0x00 (ACC_FS_S[7:6]=0, ACC_FS_W[3:2]=0) to Accelerometer Range Control Register (ACCEL_CTL) to set the accelerometer range to ±2g.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
ACCEL_CTL	0x40	0100 0000	0x00	0000 0000

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

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
STDBY	0x43	0100 0011	0xEE	1110 1110

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 Acceleration Data

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

a) Write 0xEF (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	
	Hex	Binary	Hex	Binary
STDBY	0x43	0100 0011	0xEF	1110 1111

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

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
INT1_L	0x39	0011 1001	n/a	n/a
INT2_L	0x3D	0011 1101	n/a	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	
	Hex	Binary	Hex	Binary
INT_MASK1	0x48	0100 1000	0x02	0000 0010

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	
	Hex	Binary	Hex	Binary
INT_PIN_CTL	0x45	0100 0101	0x0C	0000 1100

e) Write 0x02 (DRDY_ACCTEMP_P1[2]=1) to Interrupt Pin 1 Select Register (INT_PIN1_SEL) to route the corresponding interrupt to the INT1 pin.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
INT_PIN1_SEL	0x46	0100 0110	0x02	0000 0010

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

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
ACCEL_ODR_WAKE	0x3E	0011 1110	0xD6	1101 0110

g) Write 0x00 (ACC_FS_S[7:6]=0, ACC_FS_W[3:2]=0) to Accelerometer Range Control Register (ACCEL_CTL) to set the accelerometer range to ±2g.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
STDBY	0x40	0100 0000	0x00	0000 0000

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

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
STDBY	0x43	0100 0011	0xEE	1110 1110

i) Acceleration data can now be read synchronously (when INT1 becomes active or by monitoring INT1_DRDY_ACCTEMP[5] bit of the INT1_SRC1 register) 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 Gyroscope Data

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

a) Write 0xEF (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	
	Hex	Binary	Hex	Binary
STDBY	0x43	0100 0011	0xEF	1110 1111

b) Write 0x06 (GYRO_FS_W[7]=0, ODRG_BW[6:5]=0, ODRG_W[3:0]=6) to Gyroscope Wake Mode Control Register (GYRO_ODR_WAKE) to set the Output Data Rate (ODR) of the gyroscope to 50 Hz, bandwidth to 10Hz, and velocity range to ±256. (Note: This is also the default value.)

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
GYRO_ODR_WAKE	0x41	0100 0001	0x06	0000 0110

c) Write 0xED to Standby Register (STDBY) to set the gyroscope in run mode.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
STDBY	0x43	0100 0011	0xED	1110 1101

d) Gyroscope data can now be read asynchronously 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 Gyroscope Data

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

a) Write 0xEF (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	
	Hex	Binary	Hex	Binary
STDBY	0x43	0100 0011	0xEF	1110 1111

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

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
INT1_L	0x39	0011 1001	n/a	n/a
INT2_L	0x3D	0011 1101	n/a	n/a

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

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
INT_MASK1	0x48	0100 1000	0x01	0000 0001

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	
	Hex	Binary	Hex	Binary
INT_PIN_CTL	0x45	0100 0101	0x0C	0000 1100

e) Write 0x01 (DRDY_GYRO_P1[0]=1) to Interrupt Pin 1 Select Register (INT_PIN1_SEL) to route the corresponding interrupt to the INT1 pin.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
INT_PIN1_SEL	0x46	0100 0110	0x01	0000 0001

f) Write 0x06 (GYRO_FS_W[7]=0, ODRG_BW[6:5]=0, ODRG_W[3:0]=6) to Gyroscope Wake Mode Control Register (GYRO_ODR_WAKE) to set the Output Data Rate (ODR) of the gyroscope to 50 Hz, bandwidth to 10Hz, and velocity range to ±256. (Note: This is also the default value.)

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
GYRO_ODR_WAKE	0x41	0100 0001	0x06	0000 0110

g) Write 0xED (GYRO_STDBY_W[1]=0) to Standby Register (STDBY) to set the gyroscope in run mode.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
STDBY	0x43	0100 0011	0xED	1110 1101

h) Gyroscope data can now be read synchronously (when INT1 becomes active or by monitoring INT1_DRDY_GYRO[0] bit of the INT1_SRC1 register) 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).

a) Write 0xEF (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	
	Hex	Binary	Hex	Binary
STDBY	0x43	0100 0011	0xEF	1110 1111

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

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
INT1_L	0x39	0011 1001	n/a	n/a
INT2_L	0x3D	0011 1101	n/a	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	
	Hex	Binary	Hex	Binary
INT_MASK1	0x48	0100 1000	0x80	1000 0000

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	
	Hex	Binary	Hex	Binary
INT_PIN_CTL	0x45	0100 0101	0x0C	0000 1100

e) Write 0x80 (BFI_P1[7]=1) to Interrupt Pin 1 Select Register (INT_PIN1_SEL) to route the corresponding interrupt to the INT1 pin.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
INT_PIN1_SEL	0x46	0100 0110	0x80	1000 0000

f) Write 0x06 (GYRO_FS_W[7]=0, ODRG_BW[6:5]=0, ODRG_W[3:0]=6) to Gyroscope Wake Mode Control Register (GYRO_ODR_WAKE) to set the Output Data Rate (ODR) of the gyroscope to 50 Hz, bandwidth to 10Hz, and velocity range to ±256. (Note: This is also the default value.)

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
GYRO_ODR_WAKE	0x41	0100 0001	0x06	0000 0110

g) Write 0x07 (BUF_GYR_X[2]=1, BUF_GYR_Y[1]=1, BUF_GYR_Z[0]=1) to the Buffer Control Register 2 (BUF_CTL2). This will place the gyroscope data into the sample buffer in wake mode.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
BUF_CTL2	0x79	0111 1001	0x07	0000 0111

h) Write 0x80 (BUFE[7]=1, BUF_SYM[3:2]=0, BUF_M[1:0]=0) to the Buffer Enable Register (BUF_EN) to enable buffering of data. For this example, the buffer is being enabled, no symbols are placed between data sets, and using FIFO mode.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
BUF_CTL2	0x7C	0111 1100	0x80	1000 0000

i) Write 0xED (GYRO_STDBY_W[1]=0) to Standby Register (STDBY) to set the gyroscope in run mode.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
STDBY	0x43	0100 0011	0xED	1110 1101

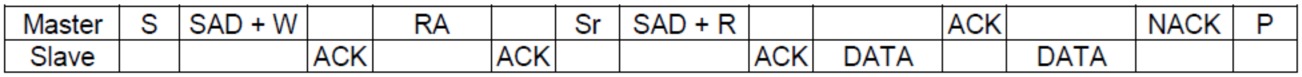
j) Once Buffer-Full Interrupt is issued on INT1 pin or by monitoring the INT1_BFI[7] bit of the INT1_SRC1 register, gyroscope data can then be read from the Buffer Read (BUF_READ) register at address 0x7F in 2's complement format. Since the resolution of the samples data is 16-bit, data will be stored 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 510 X, Y, Z high and low byte elements, which corresponds to 170 unique gyroscope data samples.

Buffer reading tips:

a) The gyroscope data can be read from a buffer using multiple-byte read as shown in Figure 2 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 2. 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 2: Proper I2C Sequence to Receive Data from the Slave

6. Wakeup from Sleep Interrupt via physical hardware interrupt (WUF)

a) Write 0xEF (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	
	Hex	Binary	Hex	Binary
STDBY	0x43	0100 0011	0xEF	1110 1111

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

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
INT1_L	0x39	0011 1001	n/a	n/a
INT2_L	0x3D	0011 1101	n/a	n/a

c) Write 0x20 (WUFE[5]=1) to Interrupt Mask Register 1 (INT_MASK1) to enable the Wake Up Function Interrupt.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
INT_MASK1	0x48	0100 1000	0x20	0010 0000

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	
	Hex	Binary	Hex	Binary
INT_PIN_CTL	0x45	0100 0101	0x0C	0000 1100

e) Write 0x20 (WUF_P1[5]=1) to Interrupt Pin 1 Select Register (INT_PIN1_SEL) to route the corresponding interrupt to the INT1 pin.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
INT_PIN1_SEL	0x46	0100 0110	0x20	0010 0000

f) Write 0x66 (WUF_EN[6]=1, MAN_SLEEP[5]=1, OWUF[2:0]=6) to the Wake Sleep Control Register 1 (WAKE_SLEEP_CTL1) 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 wake up (motion detection) will be set to 50 Hz.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
WAKE_SLEEP_CTL1	0x4B	0100 1011	0x66	0110 0110

g) Write 0x01 (ATH[7:0] = 1) to the Wake Up Function 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: 0.5g / 0.0625g/count = 8 (0x08).

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
WUF_TH	0x4D	0100 1101	0x08	0000 1000

h) Write 0x01 (WUFC[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	
	Hex	Binary	Hex	Binary
WUF_COUNTER	0x4E	0100 1110	0x01	0000 0001

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

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
STDBY	0x43	0100 0011	0xEE	1110 1110

j) Since the device is forced to “Sleep”, motion (i.e. shake the sensor) can be now be detected by monitoring INT1 and reflected in INT1_WUFS[5] bit of the INT1_SRC1 register. Additionally, the axis and direction of the detected motion is reflected in INT1_SRC2 (0x38) register.

k) Since the device is now in wake mode and back to sleep mode is disabled, it must now be forced back to sleep in order to be trigger again. This can be done by writing to the MAN_SLEEP[5] bit in the WAKE_SLEEP_CTL1 register. For advanced usage: The back-to-sleep function can be enabled by setting BTS_EN[7] in the WAKE_SLEEP_CTL1 register. The threshold and counter that will trigger a back to sleep condition is controlled through the BTS_TH and BTS_COUNTER registers, respectively.

Timing Requirements

There are several timing requirements that developers should keep in mind when working with the KXG03.

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 two (2) physical interrupt pin select registers (INT_PIN1_SEL and INT_PIN2_SEL), one for each physical interrupt pin INT1 and INT2. These select registers route the corresponding interrupt to the designated pin (INT1 and/or INT2):

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 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

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 KXG03 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 KXG03 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 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).
   - Ensure the accelerometer is in “run mode”.

3. **Gyroscope Data Ready Interrupt Not Working**
   - Make sure that the Gyroscope Data Ready interrupt is enabled (DRDY_GYRO) 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).
   - Ensure the gyroscope is in “run mode”.

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 2 (BUF_CTL2).
   - Make sure the Buffer Enable Register (BUF_EN) is enabled to buffer data.
   - Ensure the gyroscope is in “run mode”.

5. **WUF (Wake-Up Function) Interrupt Not Working**
   - Make sure that the WUF interrupt is enabled (WUFE) in the Interrupt Mask Register 1 (INT_MASK1).
   - 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.
   - 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.

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 KXG03. 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.
  - Real-time 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 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