Introduction

This application note will help developers quickly implement proof-of-concept designs using the KXTJ2 tri-axis accelerometer. Please refer to the KXTJ2 data sheet for additional implementation guidelines. While Kionix strives to ensure that our accelerometers will meet design expectations by default, it has ever so closed the gap by allowing users to autonomously analyze sensor outputs. KXTJ2 is an extension of the KXTJ9 product to a 2 x 2 mm 12-pin. The KXTJ2 is significantly lower in current than our previous offerings. The KXTJ2 provides an I2C bus interface and small 2x2 footprint with 12 pads compatible with Bosch 222/250 series products. The KXTJ2 also provides a low power motion wake up feature which is demonstrated in this application note.

Circuit Schematic

This section shows recommended wiring for the KXTJ2, based on proven operation of the part. Specific applications may require modifications from these recommendations. Please refer to the KXTJ2 Data Sheet for all pin descriptions.

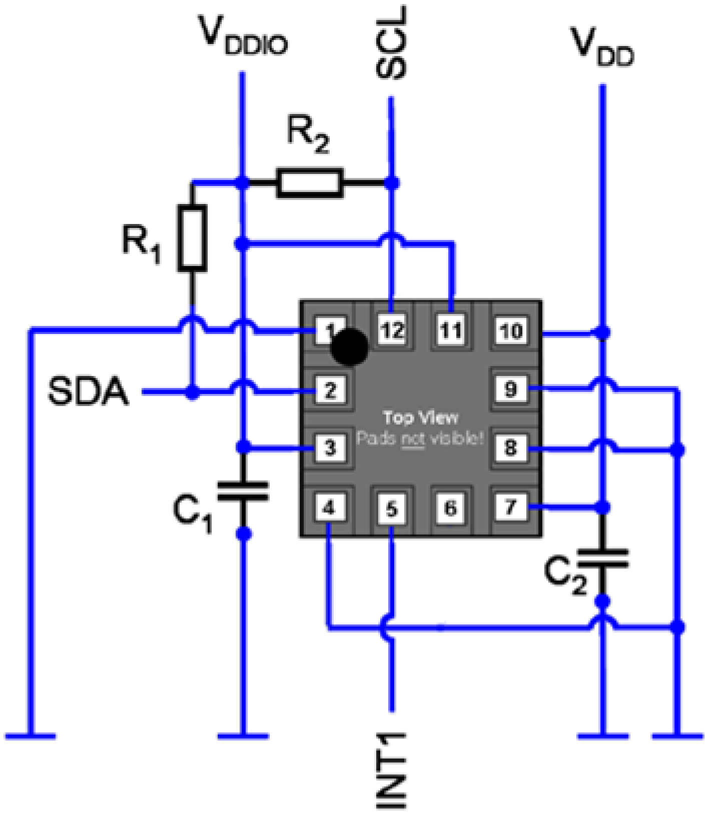


Figure 1. KXTJ2 Application Schematic

Quick Start Implementation

Here we present two basic ways to initialize the part and explore the Wake-Up (motion detect) functionality. These can vary based on desired operation, but generally the initial operations a developer wants to do are: 1) Asynchronous read back acceleration data, 2) Synchronous hardware interrupt_read back acceleration data, 3) Wake-Up function via latched physical hardware interrupt. There are other ways that the interrupt sources can be read via status registers (software interrupt) in the device that can be used to check if the interrupt event has occurred and these are not explored here.

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

- Write 0x00 to Control Register 1 (CTRL_REG1) to initialize accelerometer in stand-by mode. (PC1 = 0)

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
CTRL_REG1	**0x1B**	**0001 1011**	**0x00**	**0000 0000**

- Write 0x40 to Control Register 1 (CTRL_REG1) to keep the accelerometer in stand-by mode, to set the performance mode of the KXTJ2 to high current 12 bit resolution since we are setting the G-range to +/-2g. (Note: 14 bit resolution is only available when the G-range is +/-8g)

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
CTRL_REG1	**0x1B**	**0001 1011**	**0x40**	**0100 0000**

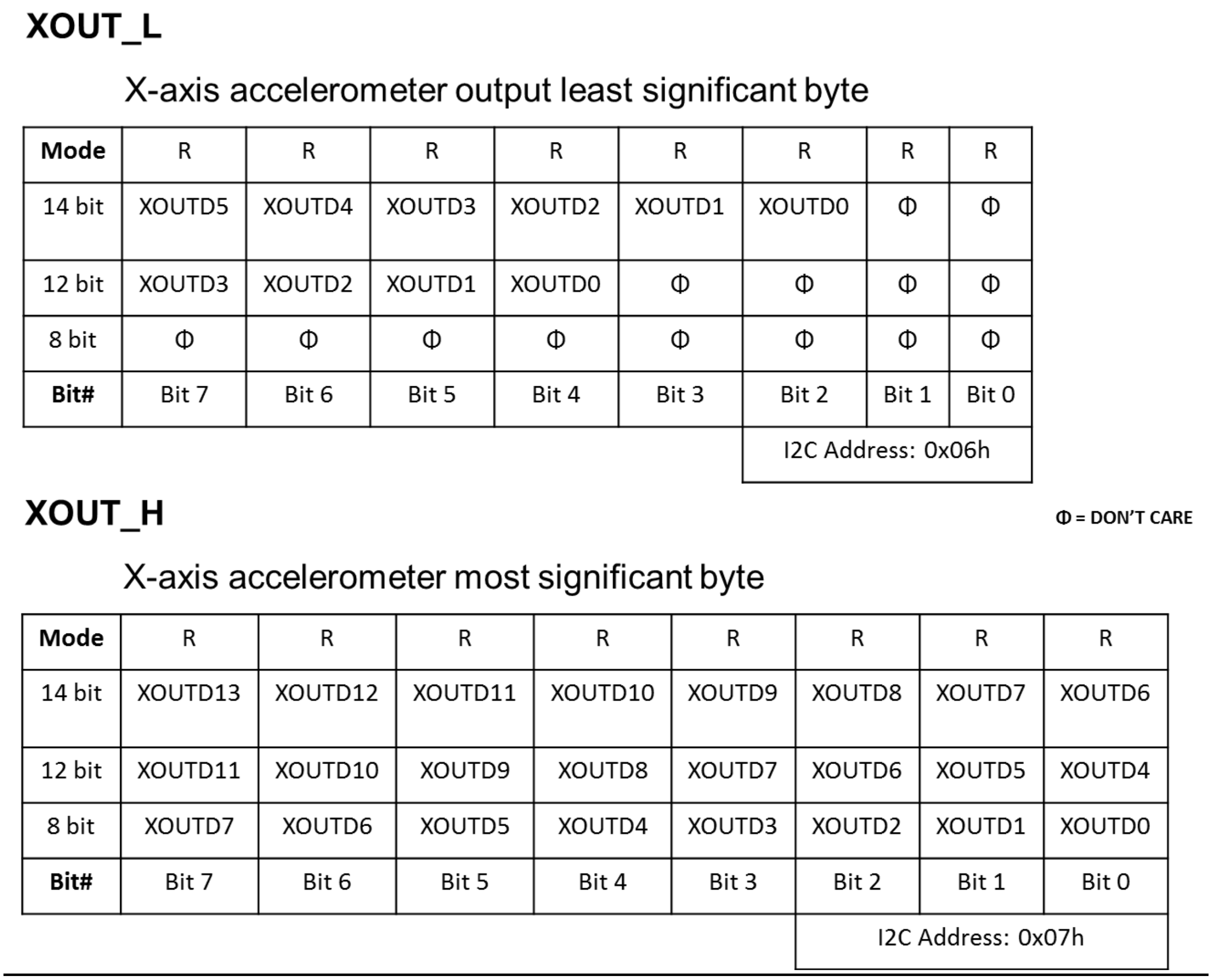
- Write 0x02 to Data Control Register (DATA_CTRL_REG) to set the Output Data Rate (ODR) of the accelerometer to 50 Hz. (Note: This is the default value. Output Data Rates >= 400Hz will force device into Full Power mode (PC1 = 1, RES = 1))

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
DATA_CTRL_REG	**0x21**	**0010 0001**	**0x02**	**0000 0010**

- Write 0xC0 to Control Register 1 (CTRL_REG1) to set the accelerometer into operating mode (PC1 = 1)

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
CTRL_REG1	**0x1B**	**0001 1011**	**0xC0**	**1100 0000**

- 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 as shown in the next tables depending on the performance mode (resolution) selected before. For example, to read a 12 bit value, XOUTD11 to XOUTD0, then XOUTD11 to XOUTD4 bit positions are located in XOUT_H register and the remaining 4 bits XOUTD3 to XOUTD0 are located in the upper nibble of the XOUT_L. The previous statement also applies to the Y and Z axis.



XOUT_L

X-axis accelerometer output least significant byte

Mode	R	R	R	R	R	R	R	R
14 bit	XOUTD5	XOUTD4	XOUTD3	XOUTD2	XOUTD1	XOUTD0	Φ	Φ
12 bit	XOUTD3	XOUTD2	XOUTD1	XOUTD0	Φ	Φ	Φ	Φ
8 bit	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ
Bit#	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

I2C Address: 0x06h

XOUT_H

Φ = DON'T CARE

X-axis accelerometer most significant byte

Mode	R	R	R	R	R	R	R	R
14 bit	XOUTD13	XOUTD12	XOUTD11	XOUTD10	XOUTD9	XOUTD8	XOUTD7	XOUTD6
12 bit	XOUTD11	XOUTD10	XOUTD9	XOUTD8	XOUTD7	XOUTD6	XOUTD5	XOUTD4
8 bit	XOUTD7	XOUTD6	XOUTD5	XOUTD4	XOUTD3	XOUTD2	XOUTD1	XOUTD0
Bit#	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

I2C Address: 0x07h

YOUT_L

Y-axis accelerometer output least significant byte

Mode	R	R	R	R	R	R	R	R
14 bit	YOUTD5	YOUTD4	YOUTD3	YOUTD2	YOUTD1	YOUTD0	Φ	Φ
12 bit	YOUTD3	YOUTD2	YOUTD1	YOUTD0	Φ	Φ	Φ	Φ
8 bit	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ
Bit#	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						I2C Address: 0x08h		

YOUT_H

Φ = DON'T CARE

Y-axis accelerometer most significant byte

Mode	R	R	R	R	R	R	R	R
14 bit	YOUTD13	YOUTD12	YOUTD11	YOUTD10	YOUTD9	YOUTD8	YOUTD7	YOUTD6
12 bit	YOUTD11	YOUTD10	YOUTD9	YOUTD8	YOUTD7	YOUTD6	YOUTD5	YOUTD4
8 bit	YOUTD7	YOUTD6	YOUTD5	YOUTD4	YOUTD3	YOUTD2	YOUTD1	YOUTD0
Bit#	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						I2C Address: 0x09h		

ZOUT_L

Z-axis accelerometer output least significant byte

Mode	R	R	R	R	R	R	R	R
14 bit	ZOUTD5	ZOUTD4	ZOUTD3	ZOUTD2	ZOUTD1	ZOUTD0	Φ	Φ
12 bit	ZOUTD3	ZOUTD2	ZOUTD1	ZOUTD0	Φ	Φ	Φ	Φ
8 bit	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ
Bit#	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						I2C Address: 0x0Ah		

ZOUT_H

Φ = DON'T CARE

Z-axis accelerometer most significant byte

Mode	R	R	R	R	R	R	R	R
14 bit	ZOUTD13	ZOUTD12	ZOUTD11	ZOUTD10	ZOUTD9	ZOUTD8	ZOUTD7	ZOUTD6
12 bit	ZOUTD11	ZOUTD10	ZOUTD9	ZOUTD8	ZOUTD7	ZOUTD6	ZOUTD5	ZOUTD4
8 bit	ZOUTD7	ZOUTD6	ZOUTD5	ZOUTD4	ZOUTD3	ZOUTD2	ZOUTD1	ZOUTD0
Bit#	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						I2C Address: 0x0Bh		

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

- Write 0x00 to Control Register 1 (CTRL_REG1) for the purpose to initialize accelerometer in stand-by mode. (PC1 = 0)

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
CTRL_REG1	**0x1B**	**0001 1011**	**0x00**	**0000 0000**

- Write 0x60 to Control Register 1 (CTRL_REG1) to keep the accelerometer in stand-by mode, to set the performance mode of the KXTJ2 to high current 12 bit resolution since we are setting the G-range to +/-2g. (Note: 14 bit resolution is only available when the G-range is +/-8g). Also enable the availability of new data as an interrupt.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
CTRL_REG1	**0x1B**	**0001 1011**	**0x60**	**0110 0000**

- Write 0x38 to Interrupt Control Register (INT_CTRL_REG1) to enable physical interrupt pin(5), 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
INT_CTRL_REG1	**0x1E**	**0001 1110**	**0x38**	**0011 1000**

- Write 0x02 to Data Control Register (DATA_CTRL_REG) to set the Output Data Rate (ODR) of the accelerometer to 50 Hz. (Note: This is the default value. Output Data Rates >= 400Hz will force device into Full Power mode (PC1 = 1, RES = 1))

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
DATA_CTRL_REG	**0x21**	**0010 0001**	**0x02**	**0000 0010**

- Write 0xE0 to Control Register 1 (CTRL_REG1) to set the accelerometer into operating mode (PC1 = 1)

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
CTRL_REG1	**0x1B**	**0001 1011**	**0xE0**	**1110 0000**

- 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 as shown in the tables shown previously depending on the performance mode (resolution) selected before. For example, to read a 12 bit value, XOUTD11 to XOUTD0, then XOUTD11 to XOUTD4 bit positions are located in XOUT_H register and the remaining 4 bits XOUTD3 to XOUTD0 are located in the upper nibble of the XOUT_L. The previous statement also applies to the Y and Z axis.

3- Wake Up Function via latched physical hardware interrupt

- Write 0x00 to Control Register 1 (CTRL_REG1) for the purpose to initialize accelerometer in stand-by mode. (PC1 = 0)

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
CTRL_REG1	**0x1B**	**0001 1011**	**0x00**	**0000 0000**

- Write 0x42 to Control Register 1 (CTRL_REG1) to keep the accelerometer in stand-by mode, to set the performance mode of the KXTJ2 to high current 12 bit resolution since we are setting the G-range to +/-2g. (Note: 14 bit resolution is only available when the G-range is +/-8g). Also enable the Wake Up (motion detect) function.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
CTRL_REG1	**0x1B**	**0001 1011**	**0x42**	**0100 0010**

- Write 0x06 to Control Register 2 (CTRL_REG2) 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
CTRL_REG2	**0x1D**	**0001 1101**	**0x06**	**0000 0110**

- Write 0x7F to Interrupt Control Register 2 (INT_CTRL_REG2) 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
INT_CTRL_REG2	**0x1F**	**0001 1111**	**0x7F**	**0111 1111**

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

  WAKEUP_TIMER (counts) = Desired Delay Time (sec) x OWUF (Hz)

  WAKEUP_TIMER (counts) == 0.1 x 50 = 5 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
WAKEUP_TIMER	**0x29**	**0010 1001**	**0x05**	**0000 0101**

- Write 0x18 to Interrupt Wake-Up Threshold (WAKEUP_THRESHOLD) to set the level to 0.5g. The following formula is used:

  WAKEUP_THREHOLD (counts) = Desired Threshold (g) x 16 (counts/g)

  WAKEUP_THREHOLD (counts) =0.5 x 16 =8 counts

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

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
WAKEUP_THRESHOLD	**0x6A**	**0110 1010**	**0x08**	**0000 1000**

- Write 0x30 to Interrupt Control Register (INT_CTRL_REG1) 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
INT_CTRL_REG1	**0x1E**	**0001 1110**	**0x30**	**0011 0000**

- Write 0xC2 to Control Register 1 (CTRL_REG1) to set the accelerometer in operating mode with the previously defined settings.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
CTRL_REG1	**0x1B**	**0001 1011**	**0xC2**	**1100 0010**

- Move the KXTJ2 and monitor the physical interrupt (INT pin 5) 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
INT REL	**0x1A**	**0001 1100**

- Monitor the physical interrupt (INT pin 5) 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.

Timing Requirements

There are several timing requirements that developers should keep in mind when working with the KXTJ2.

I²C Clock - The I²C Clock can support Fast Mode up to **400 KHz** and High Speed mode up to **3.4 MHz**.

Power Up to Communication - After the part is powered up, it takes **10ms** before it is ready for I²C communication.

Enable to Valid Outputs - After the part is enabled (PC1 bit in Control Register 1 is asserted), it takes from **2.1 ms** to **80 ms** depending on the ODR before the acceleration outputs are valid. (See KXTJ2 Product Specification for details)

Software Reset/Power On Reset Delay - After a Software or Power On Reset, the part takes **10ms** before it is ready for I²C communication.

Troubleshooting

All Interrupt Issues

- Make sure the KXTJ2 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.

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 KXTJ2. 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 while providing the ability to autonomously analyze sensor data 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
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 sheets at http://www.kionix.com/sensors/accelerometer-products.php.