

Introduction

This application note will help developers quickly implement proof-of-concept designs using the KXTF9 tri-axis accelerometer. Please refer to the KXTF9 data sheet for additional implementation guidelines. While Kionix strives to ensure that our accelerometers will meet design expectations by default, 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. We hope the information provided here will help the developer get the most out of the KXTF9

Circuit Schematic

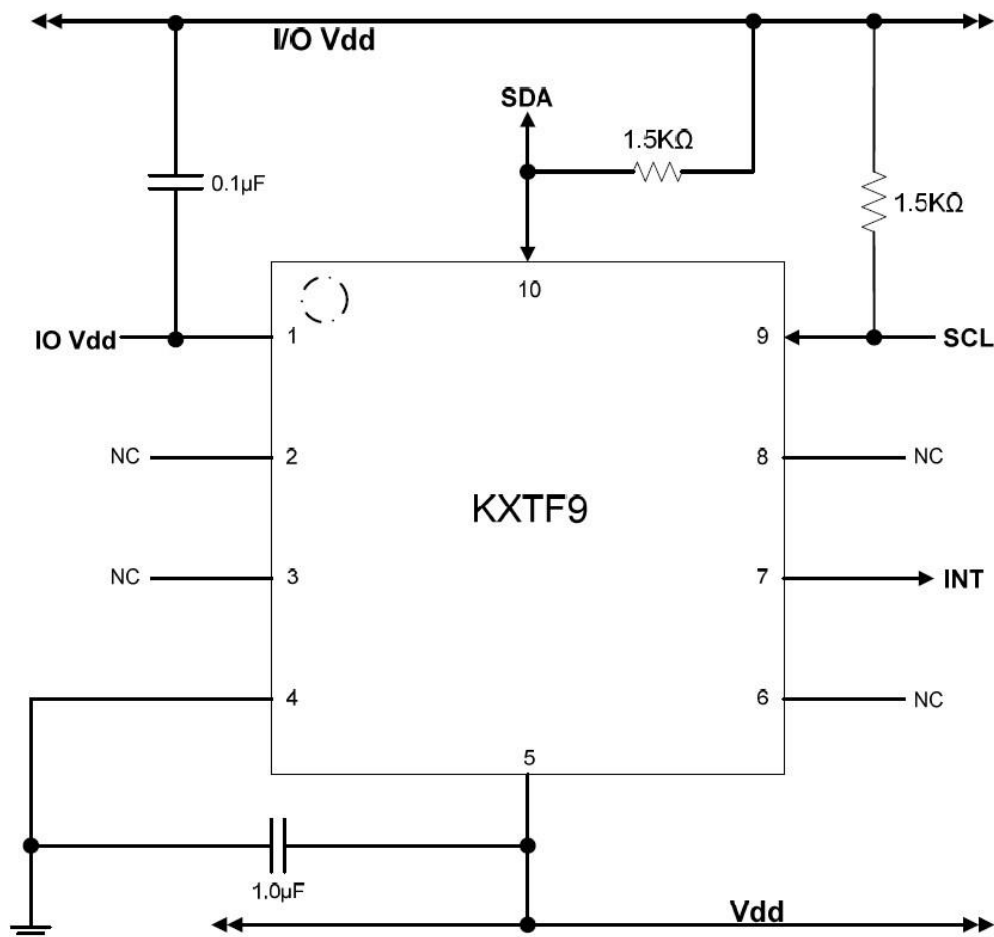


Figure 1: Application Schematic

Quick Start Implementation

Here we present three basic ways to initialize the part. These can vary based on desired operation, but generally the initial operations a developer wants to do are: read back acceleration data, activate the tilt position function, and 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.

Read Back Acceleration Data

- Write 0x80 to Control Register 1 to assert PC1 (Power Control bit), set the G-range to $\pm 2g$, and set the resolution to 8 bits.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
CTRL_REG1	0x1B	0001 1011	0x80	1000 0000

- Acceleration data can now be read from the XOUT, YOUT, and ZOUT registers.

Activate Tilt Position Function

- Write 0x01 to Tilt Timer. Here we assume a 80ms timer will be sufficient (this is based on the default Tilt Output Data Rate of 12.5Hz).
- Write 0x81 to Control Register 1 to assert PC1 and TPE (Tilt Position Enable bit), set the G-range to $\pm 2g$, and set the resolution to 8 bits.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
TILT_TIMER	0x28	0010 1000	0x01	0000 0001
CTRL_REG1	0x1B	0001 1011	0x81	1000 0001

- Changes to tilt position state will now be reflected in bit 4 of STATUS_REG (INT bit), bit 0 of INT_SRC_REG2 (TPS bit), TILT_POS_CUR, and TILT_POS_PRE registers, and also on the physical interrupt pin.

Activate Tap/Double Tap Function

- Write 0x84 to Control Register 1 to assert PC1 (Power Control bit) and TDTE (Tap/Double-Tap Enable bit), set the G-range to +/-2g, and set the resolution to 8 bits.

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
CTRL_REG1	0x1B	0001 1011	0x84	1000 0100

- Tap/double-tap events will now be reflected in bit 4 of STATUS_REG (INT bit), bits 2 and 3 of INT_SRC_REG2 (TDTS1 and TDTS0 bits), and INT_SRC_REG1 (tap direction) registers, and also on the physical interrupt pin.

Timing Requirements

There are several timing requirements that developers should keep in mind when working with the KXTF9.

- I²C Clock - The I²C Clock can be up to 400 KHz.
- Power Up to Communication - After the part is powered up, it takes 110ms 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 5ms before the acceleration outputs are valid.
- Software Reset/Power On Reset Delay - After a Software or Power-On Reset, the part takes 100 ms before it is ready for I²C communication.

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.

Interrupt Configuration

The physical interrupt has 8 possible configurations, based on two states for each of the three customizable variables located in Interrupt Control Register 1:

Alternate Unlatched Response (IEU – bit 2 – 0x04)

- 0 – Normal mode – Latched or pulse response is dictated by the IEL bit.
- 1 – Alternate mode – If the motion interrupt feature is enabled, response will always be unlatched, regardless of the value of the IEL bit. This means that when a motion event is detected, the interrupt will transition. When the motion event has ended, the interrupt will return. The transition state and return state are determined by the polarity bit IEA.

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 ($\sim 50\mu\text{s}$) 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 (1 and 2) and the INT bit (0x10) in the Status Register.

Microcontroller/GPIO Interrupt Handling

GPIO configuration is based solely on the connected hardware. The KXTF9 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 I2C 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.

Digital Engine Operation

Tilt Position Function

Position Nomenclature and Reporting

- Tilt Position Current – Indicates the current tilt position of the device. Defaults to 0x20 (left).
- Tilt Position Previous – Indicates the previous tilt position of the device. Defaults to 0x20 (left).

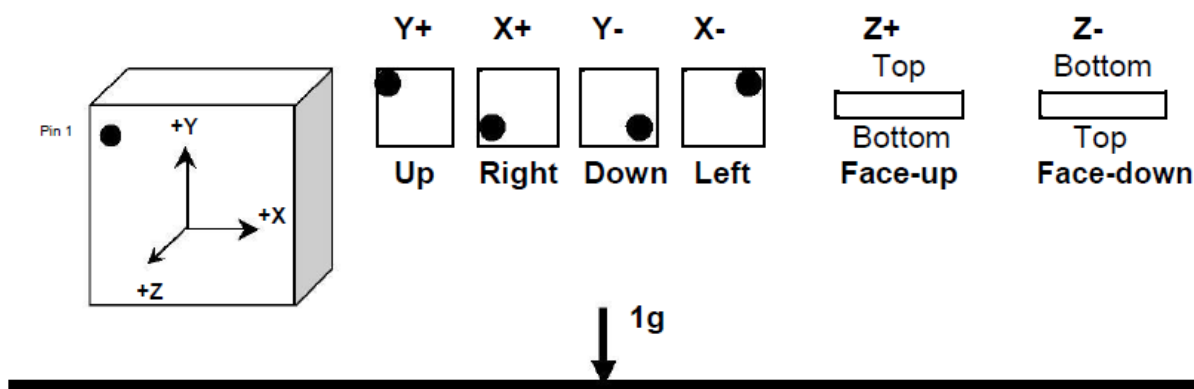
Register Name	Address		Value								
	Hex	Binary	Bit #	7	6	5	4	3	2	1	0
TILT_POS_CUR	0x10	0001 0000	Value	X	X	LE	RI	DO	UP	FD	FU
TILT_POS_PRE	0x11	0001 0001	Value	X	X	LE	RI	DO	UP	FD	FU

Bit	Description
LE	Left State (X-)
RI	Right State (X+)
DO	Down State (Y-)
UP	Up State (Y+)
FD	Face-Down State (Z-)
FU	Face-Up State (Z+)

Note that all tilt position states are with respect to the accelerometer itself. If the accelerometer's orientation does not coincide with that of the project device, some conversion must be done to the tilt position state outputs.

Orientation

If the part is set on its edge with the part number facing the user, and the pin 1 indicator is oriented so that it is in the top-left corner, the part is in the Y+ or "UP" position. If the user then turns the part 90° counter-clockwise so that pin 1 is in the bottom left corner, it will be in the X+ or "RIGHT" position. The continued rotation of the part counter-clockwise results in the Y- or "DOWN" position when pin 1 is in the bottom right corner, and X- or "LEFT" when pin 1 is in the top right corner. Setting the part on a flat surface with the part number and pin 1 indicator facing the sky results in the Z+ or "FACE-UP" state. Likewise, flipping the part over to expose the physical pins to the sky would be the Z- or "FACE-DOWN" state.



State Masking

Each of the 6 possible tilt position states can be masked using Control Register 2. To mask an entire axis, both possible states for that axis will have to be masked. Note that to mask a state, the bit associated with that state needs to be set to 0.

Timers and Thresholds

- **Tilt Timer** – This timer establishes the number of Tilt ODR cycles that a new tilt state must be active before an interrupt is triggered and the position registers are updated. The Tilt ODR is dictated by the OTPA and OTPB bits in Control Register 3.
- **Tilt Angle** – The tilt engine uses acceleration due to gravity to determine tilt position state. As such, it is unable to determine the rotational position of the part when the Z axis is in parallel with the direction of gravitational acceleration. This state (for the purposes of the Tilt Angle variable) is referred to as zero degrees. The Z axis must be a certain angle away from the direction of gravity in order for tilt changes to be registered on the X and Y axes. The Tilt Angle register is used to tell the engine at what angle it should start looking at X and Y axis acceleration instead of Z.

Motion Detection

The Wake up Function generates an interrupt when the part transitions from an inactive to an active state, as determined by the WUF_TIMER and WUF_THRESH register values. If the interrupt is configured in unlatched mode, it will be de-asserted when the part transitions from an active to an inactive state.

Axis Masking

- It is possible to mask all wake-up events which occur on a particular axis (or axes). This is done with the three bits: XBW, YBW, and ZBW in Interrupt Control Register 2.

XBW (bit 7 – 0x80)

- 0 – Events in the X axis are masked and will not generate an interrupt.
- 1 – Events in the X axis are unmasked and will generate an interrupt.

YBW (bit 6 – 0x40)

- 0 – Events in the Y axis are masked and will not generate an interrupt.
- 1 – Events in the Y axis are unmasked and will generate an interrupt.

ZBW (bit 5 – 0x20)

- 0 – Events in the Z axis are masked and will not generate an interrupt.
- 1 – Events in the Z axis are unmasked and will generate an interrupt.

Timers and Thresholds

- WUF (Wake up Function) Timer - This timer establishes the number of ODR cycles that the acceleration on an unmasked axis must be above the WUF threshold before a wake up interrupt is triggered. Each count in this register equals one Motion Detection ODR cycle, as dictated by the OWUFA and OWUFB bits in Control Register 3.
- WUF Threshold - This threshold determines how much acceleration is required in an un-masked axis in order to trigger a wake up interrupt that causes the part to transition from inactivity to activity.

Tap/Double-Tap Function

Tap Direction Resolution and Reporting

Interrupt Source Register 1 – Indicates the direction of a tap or double-tap event.

Register Name	Address		Value								
	Hex	Binary	Bit #	7	6	5	4	3	2	1	0
INT_SRC_REG1	0x15	0001 0101	Value	X	X	TLE	TRI	TDO	TUP	TFD	TFU

Bit	Description
TLE	X Negative (X-) Reported
TRI	X Positive (X+) Reported
TDO	Y Negative (Y-) Reported
TUP	Y Positive (Y+) Reported
TFD	Z Negative (Z-) Reported
TFU	Z Positive (Z+) Reported

Note that all tap/double-tap directions are with respect to the accelerometer itself. If the accelerometer's orientation does not coincide with that of the project device, some conversion must be done to the tap direction output.

Direction Masking

Each of the 6 possible tap/double-tap directions can be masked using Interrupt Control Register 3. To mask an entire axis, both possible directions for that axis will have to be masked. Note that to mask a direction, the bit associated with that direction needs to be set to 0.

Timers and Thresholds

Tap ODR – The output data rate for the tap/double-tap engine is variable between 50Hz and 400Hz. The Tap ODR is dictated by the OTDTA and OTDTB bits in Control Register 3. All tap timers are based on this ODR, which defaults to 400Hz. Each count in every tap timer equals one cycle of this Tap ODR.

TDT (Tap/Double-Tap) Timer – This timer establishes the minimum time separation between the first tap and the second tap in a double tap event. This timer defaults to 0.3 seconds.

TDT_H THRESH – This is the 9-bit upper jerk threshold for all tap events. If the performance index (PI) of a potential tap event is greater

than this threshold, the event cannot be classified as a tap event. This threshold is defaulted to 7j (jerk). Performance index is calculated as follows:

$$X' = X \text{ (current)} - X \text{ (previous)}$$

$$Y' = Y \text{ (current)} - Y \text{ (previous)}$$

$$Z' = Z \text{ (current)} - Z \text{ (previous)}$$

$$PI = |X'| + |Y'| + |Z'|$$

TDT_L_THRESH – This is the 7-bit lower jerk threshold for all tap events. If the performance index of a potential tap event is less than this threshold, the event cannot be classified as a tap event. This threshold is defaulted to 1j. Note that this register also contains bit 0 of the TDT_H_THRESH in its MSB.

TDT_TAP_TIMER – This register contains two timers which dictate the minimum and maximum times that a potential tap event must be above the performance index low limit (TDT_L_THRESH) and below the performance index high limit (TDT_H_THRESH). Bits 0 through 2 represent the low limit, which defaults to 0.005 seconds, while bits 3 through 7 represent the high limit, which defaults to 0.05 seconds.

TDT_TOTAL_TIMER – This register sets the total amount of time that the two taps in a double tap event can be above the performance index threshold (TDT_L_THRESH). This timer is defaulted to 0.09 seconds.

TDT_LATENCY_TIMER – This timer is triggered on a tap event and represents the time period that a tap event will only be characterized as a single tap. The second tap of a double tap event must occur outside of this timer. This timer is defaulted to 0.1 seconds.

TDT_WINDOW_TIMER – This timer represents the total time for the entire tap event (single or double) to occur. Reporting of single-tap events on the physical interrupt pin will occur at the expiration of this timer, while reporting of double-tap events will occur at the expiration of the second latency timer. This timer is defaulted to 0.4 seconds.

General Tips

- If the Window Timer is one more than the TDT Timer, the engine will only report single-tap events. This can result in more responsiveness for demanding applications.
- Masking directions that are unused by the application can reduce the occurrence of anomalous tap direction resolutions.
- For more information on the tap/double-tap engine, please see the KXTF9 Product Specification.

Troubleshooting

All Interrupt Issues

- Make sure the KXTF9 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 KXTF9 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 (TDT_L_THRESH) and/or reducing the performance index high threshold (TDT_H_THRESH). 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 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 KXTF9. 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's 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.

Motion

This utility allows the user to test whether the accelerometer has detected motion on any of the axes. Options are available to change the resolution (8-bit or 12-bit), output data rate, high-pass filter settings, WUF_THRESH and WUF_TIMER register values, or mask any of the axes.

Rotation

This utility allows the user to monitor the orientation of the accelerometer. It can be used to determine whether the accelerometer is in the face-up, face-down, left, right, up or down orientations. Options are available to change the resolution (8-bit or 12-bit), output data rate, TILT_TIMER, TILT_ANGLE, or HYST_SET (hysteresis) register values, or mask any of the axes.

Tap Detection

This utility allows the user to detect tap/double-tap events and see the axis and direction in which the events occur. Options are available to change the resolution (8-bit or 12-bit), output data rate, TDT (Tap/Double-Tap) Timer, TDT_H_THRESH, TDT_L_THRESH, TDT_TAP_TIMER, TDT_TOTAL_TIMER, TDT_LATENCY_TIMER, or TDT_WINDOW_TIMER register values, or mask any of the axes.

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 sheets at <http://www.kionix.com/parametric/Accelerometers>