1. Introduction

This application note will help developers quickly implement proof-of-concept designs using the **KX222,** and **KX224** tri-axis accelerometers. Please refer to the corresponding Product Specifications document for additional implementation guidelines. Kionix strives to ensure that our accelerometers will meet design expectations by default, but it is not possible to provide default setting to work in every environment. Depending on the intended application, it is very likely that some customization will be required to optimize performance. The information provided here will help the developer get the most out of these tri-axis accelerometers.

2. Circuit Schematic

This section shows recommended wiring for this family of accelerometers, based on proven operation of the part. Specific applications may require modifications from these recommendations. Please refer to the corresponding Product Specifications document for all pin descriptions.

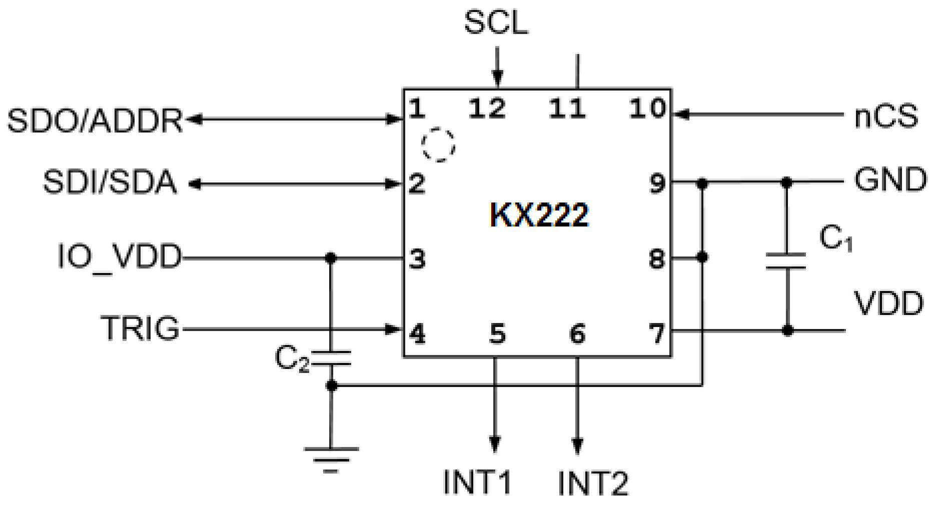

Figure 1: KX222 Application Schematic

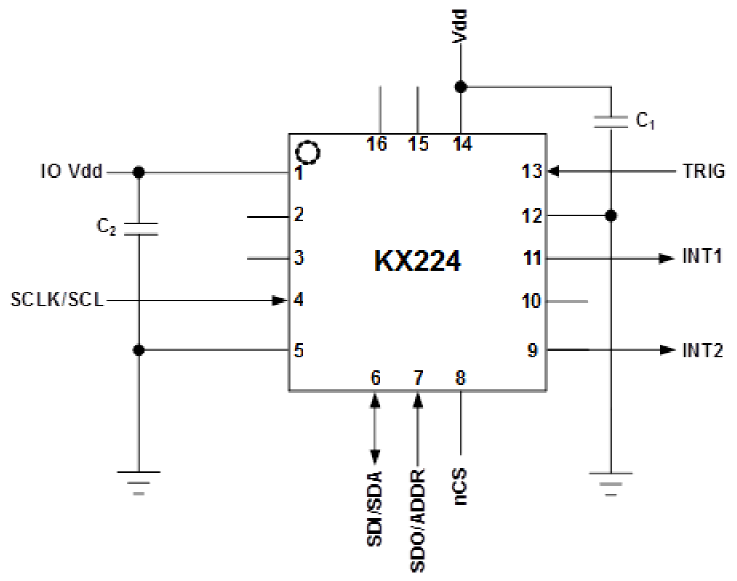

Figure 2: KX224 Application Schematic

3. Quick Start Implementations

Here we present several basic ways to initialize the part. These can vary based on desired operation, but generally the initial operations a developer wants to do are: 1) read back acceleration data asynchronously, 2) read back acceleration data when next data is ready via interrupt (synchronous data reading), 3) use of the sample buffer, 4) use the Wake-Up function, 5) activate the tilt position function, 6) activate the tap/double-tap function, and 7) 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.

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

- Write 0x40 to Control Register 1 (CNTL1) to set the accelerometer in stand-by mode, to set the performance mode to High Resolution (full power) and G-range to ±8g.

Register Name	Address	Value
CNTL1	0x18	0x40

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

Register Name	Address	Value
ODCNTL	0x1B	0x02

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

Register Name	Address	Value
CNTL1	0x18	0xC0

- 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.

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

- Write 0x60 to Control Register 1 (CNTL1) to set the accelerometer in stand-by mode, to set the performance mode to High Resolution (full power), G-range to ±8g and enable the availability of new data as an interrupt.

Register Name	Address	Value
CNTL1	0x18	0x60

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

Register Name	Address	Value
INC1	0x1C	0x38

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

Register Name	Address	Value
INC4	0x1F	0x10

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

Register Name	Address	Value
ODCNTL	0x1B	0x02

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

Register Name	Address	Value
CNTL1	0x18	0xE0

- 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. Also, the interrupt would be reflected in Bit4 (INT) of STATUS_REG and Bit4 (DRDY) in INS2 register.

6. Sample Buffer Operation

6.1. Sample Buffer-Full Interrupt via Physical Hardware Interrupt

- Write 0x00 to Control Register 1 (CNTL1) to set the accelerometer in stand-by mode, to set the performance mode of the accelerometer to Low Power mode, and G-range to ±8g.

Register Name	Address	Value
CNTL1	0x18	0x00

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

Register Name	Address	Value
INC1	0x1C	0x38

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

Register Name	Address	Value
INC4	0x1F	0x40

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

Register Name	Address	Value
ODCNTL	0x1B	0x02

- Write 0xA0 to Buffer Control Register 2 (BUF_CNTL2) to activate the sample buffer, to set the resolution of the acceleration data samples collected to 8 bits, to enable report of the interrupt status in INS2, and set the operating mode of the sample buffer to FIFO.

Register Name	Address	Value
BUF_CNTL2	0x3B	0xA0

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

Register Name	Address	Value
CNTL1	0x18	0x80

- Once Buffer-Full Interrupt is issued on INT1 pin, acceleration data can then be read from the Buffer Read (BUF_READ) register at address 0x3F in 2's complement format. Also, the interrupt would be reflected in Bit4 (INT) of STATUS_REG register and Bit6 (BFI) of INS2 register. Since the resolution of the samples data was set to 8-bit, only the most significant 8 bits of each sample will be stored in the buffer, and recorded in the following order: X_HIGH, Y_HIGH, Z_HIGH with the oldest data point read first as it is a FIFO buffer. The full buffer contains 2043 X, Y, Z elements, which corresponds to 681 unique acceleration data samples.

6.2. Trigger Mode Buffer Operation

- Ensure that the TRIG pin is can be controlled via hardware or software. The pin should be held low (inactive state) and be driven high to activate the internal trigger event.

- Write 0x00 to Control Register 1 (CNTL1) to set the accelerometer in stand-by mode, to set the performance mode of the accelerometer to Low Power mode, and G-range to ±8g.

Register Name	Address	Value
CNTL1	0x18	0x00

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

Register Name	Address	Value
INC1	0x1C	0x38

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

Register Name	Address	Value
INC4	0x1F	0x40

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

Register Name	Address	Value
ODCNTL	0x1B	0x02

- Choose desired number of samples to collect before and after a trigger event. To begin, we need to know the capacity of the buffer. The buffer has the capability of storing 2048 bytes of data. Design constraints limit the number of samples in the buffer to *rounddown(2048/dataset_in_bytes)-1*. Consider 16-bit data being stored into the buffer and a complete *dataset_in_bytes=6* (*3 axes * 2 bytes/axis*). The resulting number of samples that can be stored in the buffer is *rounddown(2048/6)-1=340*. For this example, we will save time by only collecting a single sample after a trigger event. This means we need to set a threshold of 339 samples (340-339=1 sample of data after a trigger event). The sample threshold is stored in register space across 2 different registers BUF_CNTL1 (lower 8 bits) and BUF_CNTL2 (bits 9 and 10). The following is an illustration of what values need to be written to the buffer threshold low and high bytes:

```
int smp_th = 339;
char smp_th_l = (smp_th & 0xFF);          // 0x53
char smp_th_h = ((smp_th >> 8) & 0xFF);   // 0x01
```

- Write 0x53 to Buffer Control Register 1 (BUF_CNTL1) as this is the value for BUF_TH[7:0].

Register Name	Address	Value
BUF_CNTL1	0x3A	0x53

- Read the contents of the Buffer Control Register 2

```
char buf_cntl2;
```

Register Name	Address	Value (READ)
BUF_CNTL2	0x3B	buf_cntl2

- Bit mask clear SMP_TH[9:8] from the value returned from the previous step.

```
BITMASK_CLEAR(buf_cntl2, (0x3 << 2)); // #define BITMASK_CLEAR(x,y) ((x) &= (~(y)))
```

- Logically OR and Bit Shift the new bit settings (smp_th_h) with the current value of BUF_CNTL2.

```
buf_cntl2 |= (smp_th_h << 2);
```

- Write the new value of Buffer Control Register 2 (BUF_CNTL2). If starting with default values the value of buf_cntl2 should be 0x04.

Register Name	Address	Value (WRITE)
BUF_CNTL2	0x3B	buf_cntl2

- Read the new contents of the Buffer Control Register 2

```
char buf_cntl2;
```

Register Name	Address	Value (READ)
BUF_CNTL2	0x3B	buf_cntl2

- Update and write the new bit settings in Buffer Control Register 2 (BUF_CTL2) to activate the sample buffer, to set the resolution of the acceleration data samples collected to 16 bits, to enable report of the buffer full interrupt status in INS2, and set the operating mode of the sample buffer to Trigger Mode. If starting with default values the value of buf_cntl2 should be 0xE6.

```
buf_cntl2 |=
            BUF_CNTL2_BUFE | // sample buffer active (1 << 7)
            BUF_CNTL2_BRES | // 16-bit samples in buffer (1 << 6)
            BUF_CNTL2_BFIE | // buffer full int enable (1 << 5)
            BUF_CNTL2_BUF_M_BTRIGGER // trigger mode (2 << 0)
;
```

Register Name	Address	Value (WRITE)
BUF_CNTL2	0x3B	buf_cntl2

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

Register Name	Address	Value
CNTL1	0x18	0x80

- Provide some time for the buffer to fill to the configured threshold. Assuming the default ODR was used, it should take approximately 6.8 seconds. After this time, externally activate the

Trigger pin (Trig). Next, wait for the Buffer-Full Interrupt. Once Buffer-Full Interrupt is issued on INT1 pin, acceleration data can then be read from the Buffer Read (BUF_READ) register at address 0x3F in 2's complement format. Also, the interrupt would be reflected in Bit4 (INT) of STATUS_REG register and Bit6 (BFI) of INS2 register. Since the resolution of the samples data was set to 16-bit, both high and low bytes of each sample were stored in the buffer, and recorded in the following order: X_L, X_H, Y_L, Y_H, Z_L, Z_H with the oldest data point read first as it is a FIFO buffer. The full buffer contains 2040 X, Y, Z elements, which corresponds to 340 unique acceleration data samples. The data set will include all the data prior to the trigger event, plus 1 sample after the trigger event.

Buffer reading tips:

a) The acceleration data can be read from a buffer using multiple-byte read as shown in the Figure 3 below. The register auto-increment feature is disabled when data is read from the Buffer Read register.
b) If data is read using single-byte read, it should be read in increments of 3 bytes in 8-bit resolution mode and 6 bytes in 16-bit resolution mode.
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

7. Wake Up Function via latched physical hardware interrupt

- Write 0x42 to Control Register 1 (CNTL1) to set the accelerometer in stand-by mode, to set the performance mode of the accelerometer to High Resolution (full power), G-range to ±8g and enable the Wake Up (motion detect) function.

Register Name	Address	Value
CNTL1	0x18	0x42

- Write 0x06 to Control Register 3 (CNTL3) to set the Output Data Rate of the Wake-Up function (motion detection) as set but OWUF<2:0> bits to 50 Hz.

Register Name	Address	Value
CNTL3	0x1A	0x06

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

Register Name	Address	Value
INC2	0x1D	0x7F

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

  WUFC (counts) = Desired Delay Time (sec) x Wake-Up Function ODR (Hz)

  WUFC (counts) = 0.1 sec x 50 Hz = 5 counts

  Since the desired delay time is 0.1 second and the Wake-Up Function ODR was previously set to 50 Hz, then the Wake-Up Timer is 5 counts.

Register Name	Address	Value
WUFC	0x23	0x05

- Write 0x02 to Interrupt Wake-Up Threshold (ATH) to set the level to 0.5g. The following formula is used:

  - ***WAKEUP_THRESHOLD (counts) = Desired Threshold (g) x 4 (counts/g)***
  - ***WAKEUP_THRESHOLD (counts) = 0.5 g x 4 counts/g = 2 counts***

Register Name	Address	Value
ATH	0x30	0x02

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

Register Name	Address	Value
INC1	0x1C	0x30

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

Register Name	Address	Value
INC4	0x1F	0x02

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

Register Name	Address	Value
CNTL1	0x18	0xC2

- Monitor the physical interrupt INT1 of the accelerometer, if the acceleration input profile satisfies the criteria previously established for the 0.5g motion detect threshold level in both positive and negative directions of the X, Y, Z axis for more than 0.1 second, then there should be positive latched interrupt present. Also, the interrupt would be reflected in Bit4 (INT) of STATUS_REG register and Bit1 (WUFS) in the INS2 register.

- Read Interrupt Release (INT_REL) register to unlatch (clear) the output interrupt created by the motion detection function.

Register Name	Address	Value
INT_REL	0x17	n/a

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

8. Activate Tilt Position Function with Face Detect

- Write 0x41 to Control Register 1 (CNTL1) to set the accelerometer in stand-by mode, to set the performance mode of the accelerometer to High Resolution (full power), G-range to ±8g and enable the Tilt Position function.

Register Name	Address	Value
CNTL1	0x18	0x41

- Write 0x3F to Control 2 register (CNTL2) to enable Tilt detection from positive and negative directions of all three axes (+x, -x, +y, -y, +z, -z). *This step is optional as this is also a default setting.*

Register Name	Address	Value
CNTL2	0x19	0x3F

- Write 0x98 to Control 3 register (CNTL3) to set the output data rate for the Tilt Position function to 12.5Hz. *This step is optional as this is also a default setting*.

Register Name	Address	Value
CNTL3	0x1A	0x98

- Write 0x01 to TILT_TIMER register. Here we assume an 80 msec timer will be sufficient. Note that each count value written to this register is calculated as 1/(Tilt Position ODR) = 1/12.5Hz = 80 msec.

Register Name	Address	Value
TILT_TIMER	0x22	0x01

- Write 0x03 to Tilt Angle Low Limit (TILT_ANGLE_LL) register to set the low threshold to 22º from Horizontal. Note the minimum suggested tilt angle is 10°. This value is calculated using the following equations: TILT_ANGLE_LL (counts) = sin θ * (8 (counts/g)).
-

Register Name	Address	Value
TILT_ANGLE_LL	0x32	0x03

- Write 0x0B to Tilt Angle High Limit (TILT_ANGLE_HL) register to set the High threshold. This corresponds to setting the high-level threshold above 1g plus some margin of error to account for offset variation due to external factors (e.g. device mounting).

Register Name	Address	Value
TILT_ANGLE_HL	0x33	0x0B

- Write 0x14 to HYST_SET register to set the hysteresis that is placed between the screen rotation states to ±15º. *This step is optional as this is also a default setting.*

Register Name	Address	Value
HYST_SET	0x34	0x14

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

Register Name	Address	Value
INC1	0x1C	0x30

- Write 0x01 to Interrupt Control Register 4 (INC4) to set the Tilt Position interrupt (TPI1) to be reported on physical interrupt pin INT1.

Register Name	Address	Value
INC4	0x1F	0x01

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

Register Name	Address	Value
CNTL1	0x18	0xC1

- Monitor the physical interrupt INT1 of the accelerometer. If changes in the tilt position satisfies the criteria previously established, then there should be a positive latched interrupt present. Also, the interrupt would be reflected in Bit4 (INT) of STATUS_REG, Bit0 (TPS) of INS2 register, and Current Tilt Position register TSCP as well as Previous Tilt Position register TSPP.

- Read Interrupt Release (INT_REL) register to unlatch (clear) the output interrupt created by the Tilt Position function. The read value is dummy.

Register Name	Address	Value
INT_REL	0x17	N/A

8.1. Face Detect Position

- As was noted above, the current orientation of the sensor can be read from TSCP register at any time. Assuming tilt detection was not changed from its default setting (detect positive and negative directions of all three axes (+x, -x, +y, -y, +z, -z)), one of the bits in TSCP register will be set to 1 to indicate the current orientation of the sensor.

9. Activate Tap/Double Tap Function

- Write 0x44 to Control Register 1 (CNTL1) to set the accelerometer in stand-by mode, to set the performance mode of the accelerometer to High Resolution (full power), G-range to ±8g and enable the Directional Tap function.

Register Name	Address	Value
CNTL1	0x18	0x44

- Write 0x3F to Interrupt Control 3 register (INC3) to enable tap/double tap from positive and negative directions of all three axes (+x, -x, +y, -y, +z, -z). *This step is optional as this is also a default setting.*

Register Name	Address	Value
INC3	0x1E	0x3F

- Write 0x98 to Control 3 register (CNTL3) to set the output data rate for the Directional Tap function to 400Hz. *This step is optional as this is also a default setting.*

Register Name	Address	Value
CNTL3	0x1A	0x98

- Write 0x03 to Tap / Double-Tap Register Control register (TDTRC) to enable interrupt on single tap and double tap. This step is optional as this is also a default setting. *This setting can be adjusted as needed.*

Register Name	Address	Value
TDTRC	0x24	0x03

- Write 0x78 to Tap / Double-Tap Counter register (TDTC) to set the counter to 0.3 sec. The TDTC counts starts at the beginning of the fist tap and it represents the minimum time separation between the first tap and the second tap in a double tap event. More specifically, the second tap event must end outside of the TDTC. *This step is optional as this is also a default setting. This setting can be adjusted as needed.*

Register Name	Address	Value
TDTC	0x25	0x78

- This register represents the 8-bit jerk high threshold to determine if a tap is detected. The value is compared against the upper 8 bits of the 16g output value (independent of the actual g-range setting of the device). Write 0x33 (51d) to Tap Threshold High register (TTH). This register represents the 8-bit jerk high threshold to determine if a tap is detected. Though this is an 8-bit register, the register value is internally multiplied by two to set the high threshold. This multiplication results in a range of 0d to 510d with a resolution of two counts. The Performance Index (PI) is the jerk signal that is expected to be less than this threshold, but greater than the TTL threshold during single and double tap events.

Register Name	Address	Value
TTH	0x26	0x33

- This register represents the 8-bit (0–255) jerk low threshold to determine if a tap is detected. The value is compared against the upper 8 bits of the 16g output value (independent of the actual g-range setting of the device). Write 0x07 (7d) to Tap Threshold Low register (TTL). This register represents the 8-bit (0d–255d) jerk low threshold to determine if a tap is detected. The Performance Index (PI) is the jerk signal that is expected to be greater than this threshold and less than the TTH threshold during single and double tap events.

Register Name	**Address**	**Value**
TTL	0x27	0x07

- Write 0xA2 (162d) to set the FTD counter register to 0.005 seconds. This register contains counter information for the detection of *any* tap event. A tap event must be above the performance index threshold for at least the low limit (FTDL0 – FTDL2) and no more than the high limit (FTDH0 – FTDH4). *This step is optional as this is also a default setting. This setting can be adjusted as needed.*

Register Name	**Address**	**Value**
FTD	0x28	0xA2

- Write 0x24 (36d) to set the STD counter register to 0.09 seconds. This register contains counter information for the detection of a *double* tap event. This register sets the total amount of time that the two taps in a double tap event can be above the PI threshold (TTL). *This step is optional as this is also a default setting.* This setting can be adjusted as needed.

Register Name	**Address**	**Value**
STD	0x29	0x24

- Write 0x28 (40d) to set the TLT counter register to 0.1 seconds. This register contains counter information for the detection of a tap event. This register sets the total amount of time that the tap algorithm will count samples that are above the PI threshold (TTL) during a potential tap event. It is used during both single and double tap events. However, reporting of single taps on the physical interrupt pin INT1 or INT2 will occur at the end of the TWS. *This step is optional as this is also a default setting*. This setting can be adjusted as needed.

Register Name	**Address**	**Value**
TLT	0x2A	0x28

- Write 0xA0 (160d) to set the TWS counter register to 0.4 seconds. This register contains counter information for the detection of single and double taps. This counter defines the time window for the entire tap event, single or double, to occur. Reporting of single taps on the physical interrupt pin INT1 or INT2 will occur at the end of this tap window. *This step is optional as this is also a default setting.* This setting can be adjusted as needed.

Register Name	**Address**	**Value**
TWS	0x2B	0xA0

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

Register Name	Address	Value
INC1	0x1C	0x30

- Write 0x04 to Interrupt Control Register 4 (INC4) to set the Tap/Double-Tap interrupt (TDTI) to be reported on physical interrupt pin INT1.

Register Name	Address	Value
INC4	0x1F	0x04

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

Register Name	Address	Value
CNTL1	0x18	0xC4

- Monitor the physical interrupt INT1 of the accelerometer, if the acceleration input profile satisfies the criteria previously established for Tap/Double-Tap, then there should be a positive latched interrupt present. Also, the interrupt would be reflected in Bit4 (INT) of STATUS_REG, and Bit3 and Bit2 (TDTS<1:0>) of INS2 registers. To distinguish between a single and double-tap events, monitor INS2 register bits <TDTS1:TDTS0>. TDTS1 bit3 would be set for Double-tap event, and TDTS0 bit2 for Single-tap event. Also, INS1 register can be monitored to identify the direction the tap came from.

- Read Interrupt Release (INT_REL) register to unlatch (clear) the output interrupt created by the Tap/Double-Tap detection function. The read value is dummy.

Register Name	Address	Value
INT_REL	0x17	N/A

10. Activate Free-fall Function

- Write 0x40 to Control Register 1 (CNTL1) to set the accelerometer in stand-by mode, to set the performance mode of the accelerometer to 16-bit resolution, and G-range to ±8g.

Register Name	Address	Value
CNTL1	0x18	0x40

- Write 0x80 to Free-Fall Control Register (FFCNTL) to enable Free fall engine, to set Free-fall interrupt latch control, count up/down debounce methodology, and the Output Data Rate (ODR) to 12.5Hz (Default).

Register Name	Address	Value
FFCNTL	0x2E	0x80

- Write 0x02 to Free-Fall Threshold Register (FFTH) to set the Free-fall threshold to 0.5g based on the following equation:

FFTH (counts) = Free fall Threshold (g) x 4 (counts/g)

FFTH [counts] =0.5 g x 4 counts/g = 2 counts

Register Name	Address	Value
FFTH	0x2C	0x02

- Write 0x04 to Free-Fall Counter Register (FFC) to set the Free-fall delay detection to 0.320 sec. Note that the period of the free-fall counter is a function of Free-fall ODR set by <OFFI2:OFF0> bits in FFCTNL register. The following equation can be used:

FFC (counts) = Free fall delay (sec) x Free fall ODR (Hz)

FFC [counts] = 0.320 sec x 12.5Hz = 4 counts

Register Name	Address	Value
FFC	0x2D	0x04

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

Register Name	Address	Value
INC1	0x1C	0x30

- Write 0x80 to Interrupt Control Register 4 (INC4) to set the Free-fall interrupt (FFI) to be reported on physical interrupt pin INT1.

Register Name	Address	Value
INC4	0x1F	0x80

- Write 0xC0 to Control Register 1 (CNTL1) to set the accelerometer in operating mode with the previously defined settings.

Register Name	Address	Value
CNTL1	0x18	0xC0

- Monitor the physical interrupt INT1 of the accelerometer, if the acceleration input profile satisfies the criteria previously established for the 0.5g free-fall detect threshold level in both positive and negative directions of the X, Y, Z axis for more than 0.320 second, then there should be a positive latched interrupt present. Also, Free-fall events will now be reflected in Bit4 (INT) of STATUS_REG register, and Bit7 (FFS) of INS2 register.

- Read Interrupt Release (INT_REL) register to unlatch (clear) the output interrupt created by the free-fall detection function. The read value is dummy.

Register Name	Address	Value
INT_REL	0x17	N/A

11. Timing Requirements

There are several timing requirements that developers should keep in mind when working with the KX222/KX224 accelerometer. Please refer to Technical Note *TN004 Power On Procedure* for more information.

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 **10 MHz**.

Enable to Valid Outputs (Start-Up Time) - After the part is enabled (PC1 bit in Control Register 1 is asserted), it takes from **2 ms** to **1300 ms** depending on the ODR and Power Mode setting before the acceleration outputs are valid. (See the relevant Product Specification for details)

Power-Up Time (Time from VDD and IO VDD valid to device boot completion) - After a Power-up, the part takes between **20 ms** to **50 ms** before it is ready for communication.

Software Reset Delay - After a Software Reset, the part takes **2 ms** before it is ready for communication.

Standby to Operation Delay - Please allow 2/ODR delay time when transitioning from stand-by PC1 = 0 to operating mode PC1 = 1 to allow new settings to load.

12. Interrupt Configuration

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

Latched/Pulsed (IEL – bit 3 – 0x08)

0 – Latched mode – When an interrupt is triggered, it will remain active on the pin until cleared.

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

Polarity (IEA – bit 4 – 0x10)

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

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

Enable/Disable (IEN – bit 5 – 0x20)

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

1 – Enabled – Interrupt conditions will be reflected on the physical interrupt pin.

12.1. A Few Interrupt Tips

Read the Interrupt Release Register to Clear

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

Microcontroller/GPIO Interrupt Handling –

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

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

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

Interrupt Polling - If physical interrupts are not used, a polling mechanism can be devised, which checks the INT bit in STATUS_REG. This mechanism should then look at INT_SRC_REG2 to determine which engine caused the interrupt and what steps should be taken before clearing the interrupt source information by reading the INT_REL register.

13. Troubleshooting

All Interrupt Issues

- Make sure the accelerometer 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), and Free Fall Threshold value in Free Fall Threshold Register (Free Fall Detect 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 specific 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 considering if problems continue).

WUF (Wake Up Function) Interrupt Not Working

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

TDT (Tap/Double-Tap) Interrupt Not Working

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

Free fall Detect Interrupt Not Working

- Make sure the Free fall engine is enabled (FFIE bit in Free Fall Control Register FFCNTL)
- Try adjusting the threshold requirements to achieve desired operation by adjusting the threshold value in the FFTH register. If the part is generating interrupts two often, try increasing the delay/debounce time in the FFC register. If the interrupt is not firing at all, perhaps the threshold is set too high.

14. 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.

15. 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

16. Theory of Operation

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

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