Introduction

This application note will help developers quickly implement proof-of-concept designs using the KXCNL tri-axis accelerometer. Please refer to the KXCNL data sheet for additional implementation guidelines. The KXCNL provides the capability to define two independent finite state machines with up to 16 states, along with programmable actions initiated at state transitions. This capability allows users to implement a wide range of recognition algorithms such as wake up, free fall, screen orientation, tap/double tap, step recognition, etc.. This application note discusses the implementation of free-fall and motion detection algorithm utilizing one of the state machines. Required theory, equations, and sample event signature are provided with this note as guidelines for characterizing free-fall and motion models.

Note: Examples discussed in this application note pertain to State Program 2.

Circuit Schematic

Recommended wiring for the KXCNL are based on proven operation of the part. Specific applications may require modifications from these recommendations. Please refer to the KXCNL Data Sheet for all pin descriptions.

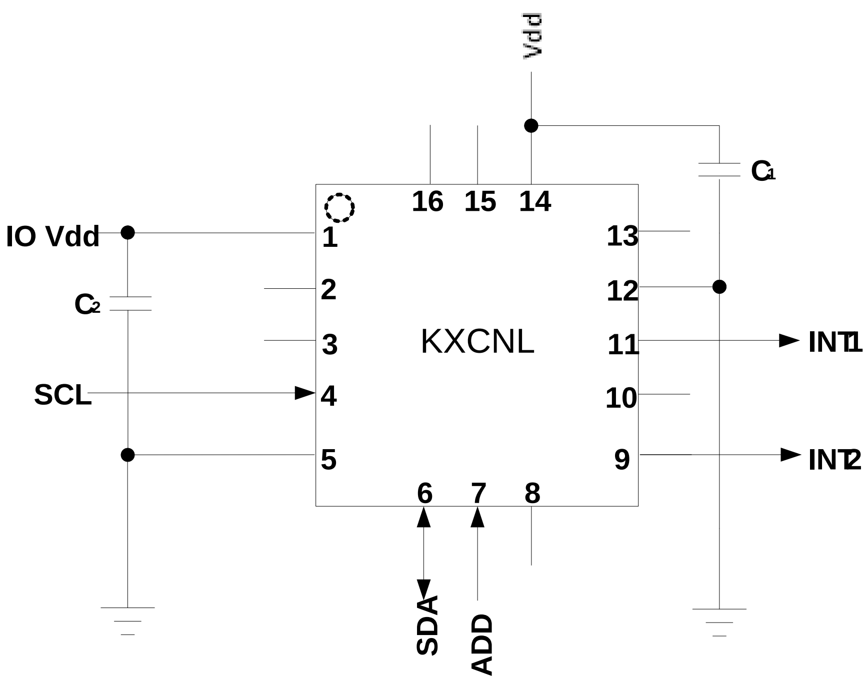

Figure 1. Application Schematic

Table 1: KXCNL Pin Descriptions

36 Thornwood Dr. – Ithaca, NY 14850
tel: 607-257-1080 – fax: 607-257-1146
www.kionix.com - info@kionix.com

Pin	Name	Description
1	V_{IO}	The power supply input for the digital logic and communication bus. Decouple this pin to ground with a 0.001 - 0.01uF ceramic capacitor.
2	NC	Not Connected Internally.
3	NC	Not Connected Internally.
4	SCL	I^2C Serial Clock
5	GND	Ground
6	SDA	I^2C Serial Data
7	ADDR	I^2C Address selection. Connect to **V_{IO}** or **GND** to select I^2C slave address.
8	NC	Not Connected Internally.
9	INT2	Physical Interrupt 2
10	NC	Not Connected Internally.
11	INT1	Physical Interrupt 1 / Data Ready
12	GND	Ground
13	NC	Not Connected Internally.
14	Vdd	The main power supply input. Decouple this pin to ground with a 0.1 - 0.47uF ceramic capacitor.
15	NC	Not Connected Internally.
16	NC	Not Connected Internally.

Quick Start Implementation

Two basic ways to initialize the part are presented. These methods can vary based on desired operation, but generally the initial operations a developer wants to do are: 1) read back acceleration data, 2) use one of the state machines. 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- Read Back Acceleration Data

- Write 0x95 to Control Register 1 (CTRL1) to assert PC1 (Power Control bit), set the G-range to +/-2g, and set the ODR to 100 Hz.
-

Register Name	Address		Value	
	Hex	Binary	Hex	Binary
CTRL_REG1	**0x1B**	**0001 1011**	**0x95**	**1001 0101**

- Acceleration data can now be read from the OUTX_L, OUTX_H, OUTY_L, OUTY_H, OUTZ_L, and OUTZ_H registers.

2- State Program 2 as a Free Fall Detection Engine or Motion Detection Engine

Overview

Many applications require some sort of processing of the sensor readings, in this application note we will discuss free fall and motion detection. In free fall detection, generally one wants to know when total acceleration (Equation 1) has stayed below a certain threshold for a certain amount of time, where in motion detection, generally one wants to know when acceleration on one or more axis is above a threshold for a certain amount of time. However, total acceleration in KXCNL is calculated with an approximation formula (Equation 2). The calculated total acceleration vector length result is filtered (if enabled) with an adjustable Band Pass filter (Please refer to the Product Data Sheet under Vector Filter Coefficients section). Free-Fall and Motion algorithms and many others can be described as a finite state machine. To support this type of decoding without CPU intervention, KXCNL includes two highly configurable state machines with up to 16 states. The behavior of each state can be individually configured. Please refer to State Program OP Codes under the Appendix Section and End Programmers Topics for State Program Execution document for a complete list and proper usage of conditions and commands.

$$a_{total} = \sqrt{x^2 + y^2 + z^2}$$

Equation 1: Total Acceleration

$$a_1 = |x| + |y| + |z|$$

$$a_2 = \max(|x|, |y|, |z|)$$

$$v_{raw} = (45 * a_1 + 77 * a_2) / 256$$

Equation 2: Approximation for Total Acceleration

Free-Fall Algorithm

When a tri-axis accelerometer is stationary, its total acceleration it measures is 1g (9.8 m/s^2), regardless of orientation. When a tri-axis accelerometer is dropped in any orientation, it is in free-fall and the measured acceleration on all three axis is 0g. Therefore, the total vector is zero as well. Total vector can be monitored by the state programs to determine if the accelerometer has been dropped. Throughout this application note we will be looking at linear free-fall and will not be discussing scenarios where rotation and or projection is introduced. Table 1 below describes the implementation of free-fall in State Program 2 with following algorithm parameters (Please refer to State Program Appendix document for detailed description of Conditions and Commands usage). Table 2 describes the necessary control register settings (Refer to Product Datasheet for control register bit descriptions).

Algorithm Parameters:
- Data = raw, no decimation
- Threshold = 0.250 G, unsigned
- Stability Timer = 100 ms
- G-range = +/-2g
- ODR = 100 Hz
- Interrupt = interrupt is latched and routed to INT2 pin
- Mask = Vector length, unfiltered

Register Name	Address		Mnemonic		Value		Description
	Hex	Binary	Reset	Next	Hex	Binary	
/ST1_2	**0x60**	**0110 0000**	**NOP**	**LLTH2**	**0x0A**	**0000 1010**	wait for vector length to be less than threshold 2
/ST2_2	**0x61**	**0110 0001**	**GNTH2**	**TI2**	**0x62**	**0110 0010**	make sure vector length is less than threshold 2 for 100ms, reset immediately if vector length is greater than threshold 2
/ST3_2	**0x62**	**0110 0010**		**OUTC**	**0x88**	**1000 1000**	Output source information to /OUTS2 register and continue
/ST4_2	**0x63**	**0110 0011**		**CONT**	**0x11**	**0001 0001**	Continue execution from reset point, reset point is first address of state program 2
/ST5_2	**0x64**	**0110 0100**			**0x00**	**0000 0000**	Not Used
/ST6_2	**0x65**	**0110 0101**			**0x00**	**0000 0000**	Not Used
/ST7_2	**0x66**	**0110 0110**			**0x00**	**0000 0000**	Not Used
/ST8_2	**0x67**	**0110 0111**			**0x00**	**0000 0000**	Not Used
/ST9_2	**0x68**	**0110 1000**			**0x00**	**0000 0000**	Not Used
/ST10_2	**0x69**	**0110 1001**			**0x00**	**0000 0000**	Not Used
/ST11_2	**0x6A**	**0110 1010**			**0x00**	**0000 0000**	Not Used

/ST12_2	**0x6B**	**0110 1011**			**0x00**	**0000 0000**	Not Used
/ST13_2	**0x6C**	**0110 1100**			**0x00**	**0000 0000**	Not Used
/ST14_2	**0x6D**	**0110 1101**			**0x00**	**0000 0000**	Not Used
/ST15_2	**0x6E**	**0110 1110**			**0x00**	**0000 0000**	Not Used
/ST16_2	**0x6F**	**0110 1111**			**0x00**	**0000 0000**	Not Used
/TIM4_2	**0x70**	**0111 0000**			**0x00**	**0000 0000**	
/TIM3_2	**0x71**	**0111 0001**			**0x00**	**0000 0000**	
/TIM2_2 (LSB)	**0x72**	**0111 0010**			**0x0A**	**0000 1010**	Timer 2 = 10 (100 ms @ 100 Hz ODR)
/TIM2_2 (MSB)	**0x73**	**0111 0011**			**0x00**	**0000 0000**	
/TIM1_2 (LSB)	**0x74**	**0111 0100**			**0x00**	**0000 0000**	
/TIM1_2 (MSB)	**0x75**	**0111 0101**			**0x00**	**0000 0000**	
/THRS2_2	**0x76**	**0111 0110**			**0x0F**	**0000 1111**	Threshold 2 = 15 (0.250 mg @ +/-2g range)
/THRS1_2	**0x77**	**0111 0111**			**0x00**	**0000 0000**	
/DES2	**0x78**	**0111 1000**			**0x00**	**0000 0000**	
/SA2	**0x79**	**0111 1001**			**0x00**	**0000 0000**	
/MA2	**0x7A**	**0111 1010**			**0x02**	**0000 0010**	+Vector length unmasked
/SETT2	**0x7B**	**0111 1011**			**0x01**	**0000 0001**	Continue command proceeds (continue execution from reset point)

Table 2: Implementation of Free-Fall in State Program 2

Register Name	**Address**		**Value**		**Description**
	Hex	Binary	Hex	Binary	
/CTRL1	**0x1B**	**0001 1011**	**0x95**	**1001 0101**	Active mode, +/-2g range, 100 Hz ODR, and physical interrupt enabled
/CTRL3	**0x1D**	**0001 1101**	**0x09**	**0000 1001**	State Program 2 interrupt routed to INT2 pin, and State Program 2 enabled
/CTRL4	**0x1E**	**0001 1110**	**0x50**	**0101 0000**	Interrupt signal active high, INT2 signal is enabled,

Table 3: Control Register Settings

Motion Detection Algorithm

When a tri-axis accelerometer is stationary, its total acceleration it measures is 1g (9.8 m/s^2), regardless of orientation. In order for all three axes to be equally sensitive when triggering a motion interrupt we need to take out the gravitational component. This is why we will be looking at differential acceleration. In this example, differential acceleration will be configured for current x, y, z sample minus the previous x, y, z sample. When a tri-axis accelerometer is moved in any direction that yields a stimulus greater than a pre-defined threshold, it is deemed to be in motion and an interrupt will be sent. Table 3 below describes the implementation of motion detection in State Program 2 with following algorithm parameters (Please refer to State Program Appendix document for detailed description of Conditions and Commands usage). Table 4 describes the necessary control register settings (Refer to Product Datasheet for control register bit descriptions).

Algorithm Parameters:
- Data = differential, no decimation
- Threshold = 0.080 G, unsigned
- Stability Timer = 200 ms
- G-range = +/-2g
- ODR = 25 Hz
- Interrupt = interrupt is latched and routed to INT2 pin
- Mask = unfiltered

Register Name	Address		Mnemonic		Value		Description
	Hex	Binary	Reset	Next	Hex	Binary	
/ST1_2	**0x60**	**0110 0000**	**NOP**	**GNTH2**	**0x06**	**0000 0110**	Any/triggered axis greater than threshold 2
/ST2_2	**0x61**	**0110 0001**	**LLTH2**	**TI2**	**0xA2**	**1010 0010**	Make sure that differential acceleration is greater than threshold for 200 ms or 5 samples (25 Hz ODR)
/ST3_2	**0x62**	**0110 0010**		**OUTC**	**0x88**	**1000 1000**	Output source information to /OUTS2 register and continue
/ST4_2	**0x63**	**0110 0011**		**CONT**	**0x11**	**0001 0001**	Continue execution from reset point, reset point is first address of state program 2
/ST5_2	**0x64**	**0110 0100**			**0x00**	**0000 0000**	Not Used
/ST6_2	**0x65**	**0110 0101**			**0x00**	**0000 0000**	Not Used
/ST7_2	**0x66**	**0110 0110**			**0x00**	**0000 0000**	Not Used
/ST8_2	**0x67**	**0110 0111**			**0x00**	**0000 0000**	Not Used
/ST9_2	**0x68**	**0110 1000**			**0x00**	**0000 0000**	Not Used
/ST10_2	**0x69**	**0110 1001**			**0x00**	**0000 0000**	Not Used

/ST11_2	**0x6A**	**0110 1010**			**0x00**	**0000 0000**	Not Used
/ST12_2	**0x6B**	**0110 1011**			**0x00**	**0000 0000**	Not Used
/ST13_2	**0x6C**	**0110 1100**			**0x00**	**0000 0000**	Not Used
/ST14_2	**0x6D**	**0110 1101**			**0x00**	**0000 0000**	Not Used
/ST15_2	**0x6E**	**0110 1110**			**0x00**	**0000 0000**	Not Used
/ST16_2	**0x6F**	**0110 1111**			**0x00**	**0000 0000**	Not Used
/TIM4_2	**0x70**	**0111 0000**			**0x00**	**0000 0000**	
/TIM3_2	**0x71**	**0111 0001**			**0x00**	**0000 0000**	
/TIM2_2 (LSB)	**0x72**	**0111 0010**			**0x05**	**0000 1010**	Timer 2 = 5 (200 ms @ 25 Hz ODR)
/TIM2_2 (MSB)	**0x73**	**0111 0011**			**0x00**	**0000 0000**	
/TIM1_2 (LSB)	**0x74**	**0111 0100**			**0x00**	**0000 0000**	
/TIM1_2 (MSB)	**0x75**	**0111 0101**			**0x00**	**0000 0000**	
/THRS2_2	**0x76**	**0111 0110**			**0x0F**	**0000 0101**	Threshold 2 = 5 (0.080 mg @ +/-2g range)
/THRS1_2	**0x77**	**0111 0111**			**0x00**	**0000 0000**	
/DES2	**0x78**	**0111 1000**			**0x00**	**0000 0000**	
/SA2	**0x79**	**0111 1001**			**0x00**	**0000 0000**	
/MA2	**0x7A**	**0111 1010**			**0xFC**	**1111 1100**	+/-Vector length masked
/SETT2	**0x7B**	**0111 1011**			**0x11**	**0001 0001**	Use difference data and continue command proceeds (continue execution from reset point)

Table 2: Implementation of Free-Fall in State Program 2

Register Name	**Address**		**Value**		**Description**
	Hex	Binary	Hex	Binary	
/CTRL1	**0x1B**	**0001 1011**	**0x8D**	**1000 1101**	Active mode, +/-2g range, 25 Hz ODR, and physical interrupt enabled
/CTRL3	**0x1D**	**0001 1101**	**0x09**	**0000 1001**	State Program 2 interrupt routed to INT2 pin, and State Program 2 enabled
/CTRL4	**0x1E**	**0001 1110**	**0x50**	**0101 0000**	Interrupt signal active high, INT2 signal is enabled,

Table 3: Control Register Settings

Free-Fall Test

Kionix's USB Development Kit along with KXCNL accelerometer was used for collecting Free-Fall data. The USB Development Kit was dropped onto a table top from approximately 0.5 m height. This data was later used in debug mode to validate the free-fall algorithm coded in State Program 2 as shown in Figure 2 below.

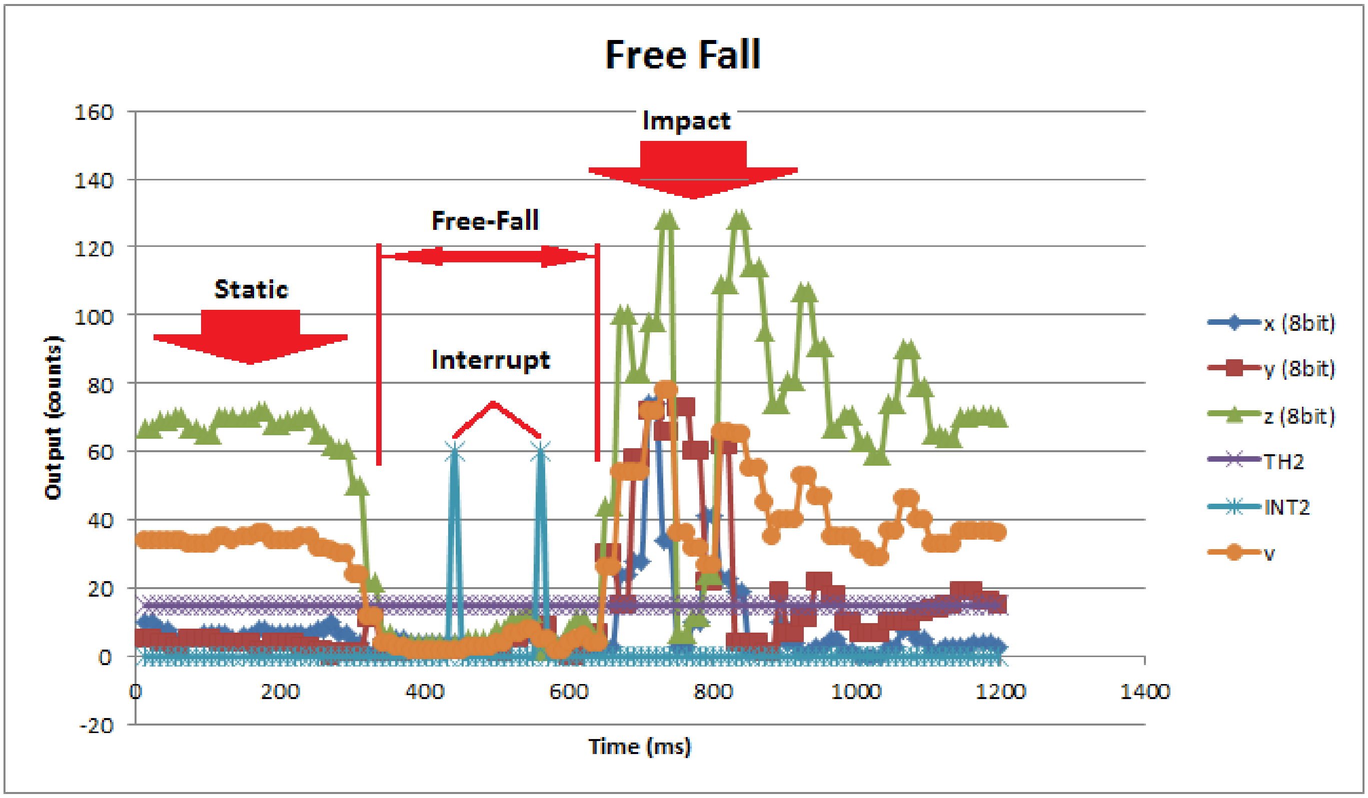

Figure 2: Free-Fall Signature

Motion Detection Test

Kionix's USB Development Kit along with KXCNL accelerometer was used for collecting Motion data. This data was later used in debug mode to validate the motion detection algorithm coded in State Program 2 as shown in Figure 3 below.

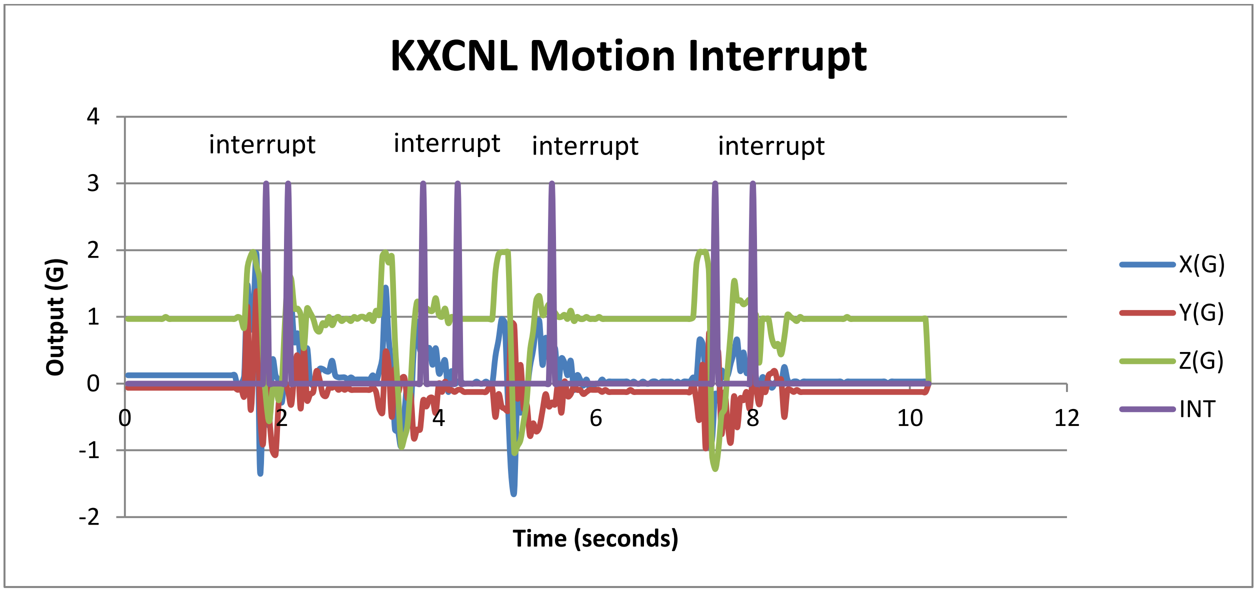

Figure 3: Motion Detection

Reliability of Algorithm

Bias offset will affect the reliability of free-fall algorithm (Please refer to the qualification report for variation in 0g offset). This is why it is very important to perform 0g calibration whenever possible. 0g offset correction values can be directly applied to raw data by storing them into OFF_X, OFF_Y, and OFF_Z registers. Please refer to Application Note AN012 for ways to measure offset bias error.

Placement

It is important to note that the placement of the accelerometer within the target device can have a significant effect on free-fall and/or other algorithms running within the state machine. If reliable free-fall detection is desired, the ideal location of an accelerometer should be at the target device's center of mass to minimize the effect of spin during free-fall event.

Interrupt Release

In latched mode, if the output source information is loaded into /OUTS2 register using OUTC command as shown in Table 1, interrupt can be released by simply reading the /OUTS2 register after an interrupt has fired. This will also clear the /OUTS2 register. If physical interrupts are not used, a polling mechanism can be devised, which checks the accelerometer output status bits in /STAT register. In this application note we are using State Program 2, therefore monitoring INT_SM2 bit will tell us whether or not State Program 2 has triggered a free-fall event. Again, by reading /OUTS2 register, interrupt information will be released/reset in /STAT register.

Masking

Each of the 4 axes (X, Y, Z and V) along with direction can be masked using /MA2 register. Note that to mask a particular axis along with direction, the bit associated with the particular axis and direction needs to be set to 0. To mask an entire axis, both possible directions for that axis will have to be set to 0.

Timing Requirements

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

I²C Clock - The I²C Clock can be up to **3.4 MHz**.

Power Up to Communication - After the part is powered up, it takes **50ms** 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 **0.5ms** before the acceleration outputs are valid.

Software Reset/Power On Reset Delay - After a Software or Power On Reset, the part takes **50ms** before it is ready for I²C communication.

Troubleshooting

All Interrupt Issues

- Make sure the KXCNL 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 main interrupt enable switch (IEN) bit in /CTRL1 register, Double check the routing of State Program 2 to INT2 pin (SM2_PIN) bit in Control Register 3, the total acceleration (vector) mask bit in /MA2 register, the interrupt enable for State Program 2 (INT2_EN) bit in /CTRL4 register.
- The 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 KXCNL product specification).

State Program not Working

- Make sure that State Program 2 is enabled, SM2_EN bit in /CTRL3 register.

- Try increasing the threshold value in /THRS2_2 register and/or decreasing the stability timer value in /TIM2_2 register to ensure that the algorithm is working and that it is in fact the sensor data that is causing the State Program 2 not to fire an interrupt.

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 KXCNL. 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 users 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.

Appendix

1.0 State Program OP Codes

#	Mnemonic	Explanation	Notes
0h	NOP	No operation	Execution moved to next or resetconditions in state
1h	TI1	Timer 1 valid	Data samples are not evaluated
2h	TI2	Timer 2 valid	Data samples are not evaluated
3h	TI3	Timer 3 valid	Data samples are not evaluated
4h	TI4	Timer 4 valid	Data samples are not evaluated
5h	GNTH1	Any/triggered axis greater than threshold 1	First axis triggers
6h	GNTH2	Any/triggered axis greater than threshold 2	First axis triggers
7h	LNTH1	Any/triggered axis less than or equal to threshold 1	First axis triggers
8h	LNTH2	Any/triggered axis less than or equal to threshold 2	First axis triggers
9h	GTTH1	Any/triggered axis greater than threshold 1	First axis triggers
Ah	LLTH2	All axis less than or equal to threshold 2	First masked axis triggers
Bh	GRTH1	Any/triggered axis greater than to reversed threshold 1	First axis triggers
Ch	LRTH1	Any/triggered axis less than or equal to reversed threshold 1	First axis triggers
Dh	GRTH2	Any/triggered axis greater than to reversed threshold 2	First axis triggers
Eh	LRTH2	Any/triggered axis less than or equal to reversed threshold 2	First axis triggers
Fh	NZERO	Any axis zero crossed	Uses previous data samples sign First axis triggers

Table 1. Conditions

#	Mnemonic	Explanation	Run Scope	Notes
00h	STOP	Stop execution, and resets reset-point to start	Immediately	Output also if enabled
11h	CONT	Continues execution from reset-point	Immediately	Output also if enabled
22h	JMP	Jump address for two Next conditions - 1st parameter is conditions - 2nd parameter are addresses for valid conditions	Immediately for command & Sample for conditions	Special (command and conditions)
33h	SRP	Set reset-point to next address / state	Immediately	
44h	CRP	Clear reset-point to start position (to 1st address)	Immediately	
55h	SETP	Set parameter in register memory -1st is address of parameter - 2nd parameter is new parameter set to address	Immediately	Address parameter is direct absolute pointer to register memory
66h	SETS1	Set new setting to Settings 1 register - 1st is new settings byte	Immediately	
77h	STHR1	Set new value to /THRS1_y register - 1st is new settings byte	Immediately	
88h	OUTC	Set outputs to output registers	Immediately output	
99h	OUTW	Set outputs to output registers and wait for latch reset from host	Immediately output and Wait (host)	Host driven event
AAh	STHR2	Set new value to /THRS2_y register - 1st is new settings byte	Immediately	
BBh	DEC	Decrease long counter -1 and validate counter	Immediately	
CCh	SISW	Swaps sign information to opposite in mask and trigger	Immediately	
DDh	REL	Releases temporary output information	Immediately	
EEh	STHR3	Set new value to /THRS3 register - 1st is new settings byte	Immediately	
FFh	SSYNC	Set synchronization point to other State program	Immediately and Wait (sync)	Affects both State Programs

Table 2. Commands

#	Mnemonic	Explanation	Run Scope	Notes
12h	SABS0	Set /SETTy, bit ABS = 0. Select unsigned filter	Immediately	
13h	SABS1	Set /SETTy, bit ABS = 1. Select signed filter ON	Immediately	
14h	SELMA	Set /MASAy pointer to MAy (set MASAy = 0)	Immediately	
21h	SRADI0	Set /SETT2, bit RADI = 0. Select raw data mode	Immediately	Only for State Program 2*
23h	SRADI1	Set /SETT2, bit RADI = 1. Select difference data mode	Immediately	Only for State Program 2*
24h	SELSA	Set /MASAy pointer to SAy (set MASAy = 1)	Immediately	
31h	SCS0	Set /SETT2, bit D_CS = 0. Select DIFF data mode	Immediately	Only for State Program 2*
32h	SCS1	Set /SETT2, bit D_CS = 1. Select Constant Shift data mode	Immediately	Only for State Program 2*
34h	STRAM0	Set /SETTy, bit R_TAM = 0. Temporary Axis Mask /TAMxAy is kept intact	Immediately	
41h	STIM3	Set new value to /TIM3_y register - 1st is new settings byte	Immediately	
42h	STIM4	Set new value to /TIM4_y register - 1st is new settings byte	Immediately	
43h	SRTAM1	Set /SETTy, bit R_TAM = 1. Temporary Axis Mask /TAMxAy is released to default after every valid condition	Immediately	

Table 3. Commands (extended set)

***Note:** 21h, 23h, 31h, and 32h are forbidden with State Program 1. When a forbidden OP code exists in State Program y, it will immediately stop/halt (F_SMy_EM = 0).

Kionix®