± 2g Tri-Axis Digital Accelerometer Specifications

PART NUMBER: KXP84-2050

Rev 2
Mar 07

36 Thornwood Drive
Ithaca, New York 14850
Tel: 607-257-1080
Fax: 607-257-1146
www.kionix.com

APPROVED BY		DATE
PROD. MGR.	S. Miller	3/19/07
TECH MGR.	K. Foust	3/19/07
TEST MGR.	J. Chong	3/19/07
VP ENG.	T. Davis	3/19/07

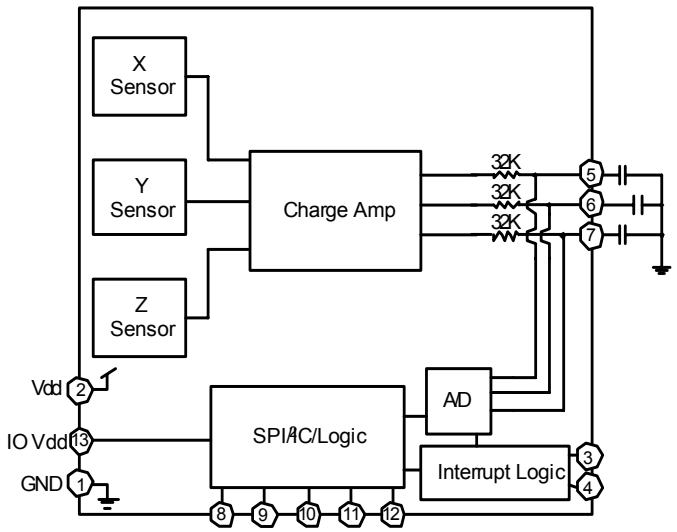
ECN #	REV	PG #	REVISION DESCRIPTION	DATE	REV'D BY
	-		Initial release	2/14/06	S. Miller
06-004	1		Updated specification to new format	3/23/06	A. Bergstrom, S. Miller
	1.1		Fixed FFMOTI description error	9/28/2006	K. Foust
07-007	2	6	Revised application schematic and pin description	3/19/07	I. Cemer

Kionix®	**± 2g Tri-Axis Digital Accelerometer Specifications**	**PART NUMBER: KXP84-2050** **Rev 2** **Mar 07**

Product Description

The KXP84-2050 is a tri-axis silicon micromachined accelerometer with a full-scale output range of ±2g (19.6m/s^2). The sense element is fabricated using Kionix's proprietary plasma micromachining process technology. Acceleration sensing is based on the principle of a differential capacitance arising from acceleration-induced motion of the sense element, which further utilizes common mode cancellation to decrease errors from process variation, temperature, and environmental stress. The sense element is hermetically sealed at the wafer level by bonding a second silicon lid wafer to the device using a glass frit. A separate ASIC device packaged with the sense element provides signal conditioning and self-test. The accelerometer is delivered in a 5 x 5 x 1.2mm Dual Flat No-lead (DFN) plastic package operating from a 2.7 - 5V DC supply. The digital I/O pads are powered from a separate power pin, and will interface to 1.8V logic. The ASIC will trigger interrupt signals if an acceleration threshold is exceeded in any axis (motion interrupt), or if the total acceleration falls below a threshold (free-fall interrupt). The threshold levels are set by the customer. Either I^{2}C or SPI interfaces can be used to communicate to the chip to trigger A/D conversions, set thresholds or threshold delays, or manage power consumption.

Functional Diagram



Kionix®	**± 2g Tri-Axis Digital Accelerometer Specifications**	**PART NUMBER: KXP84-2050** **Rev 2** **Mar 07**

Product Specifications

Table 1. Mechanical

(specifications are for 12-bit operation at V_{dd} = 3.3V and T = 25ºC unless stated otherwise)

Parameters	Units	Min	Target	Max
Operating Temperature Range	ºC	-40	-	85
Zero-g Offset	counts	1946	2048	2150
Zero-g Offset Variation from RT over Temp.	mg	-150	-	150
Sensitivity	counts/g	794	819	844
Sensitivity Variation from RT over Temp.	%	-2	0	+2
Offset Ratiometric Error (V_{dd} = 3.3V ± 5%)	%	-	0.4	1.5
Expanded Offset Ratiometric Error (V_{dd} = 2.7 – 3.6V)	% mg	-3.5 -87.5	0 0	3.5 87.5
Sensitivity Ratiometric Error (V_{dd} = 3.3V ± 5%)	%	-	0.4	1.5
Expanded Sensitivity Ratiometric Error (V_{dd} = 2.7 – 3.6V)	%	-3.5	0	3.5
Resolution	mg	-	1.22	-
Non-Linearity	% of FS	-	0.1	0.5
Cross Axis Sensitivity	%	-	2.0	3.0
Self Test Output change on Activation	g	2.0 (xy) 0.4 (z)	2.2 (xy) 0.8 (z)	2.4 (xy) 1.2 (z)
Bandwidth (-3dB)[1]	Hz	-	-	3500 (xy) 1750 (z)
Noise Density (on filter pins)	µg / √Hz	-	175	250

Notes:

1. User definable with external capacitors. Maximum defined by the frequency response of the sensors.

Table 2. Electrical

(specifications are for 12-bit operation at V_{dd} = 3.3V and T = 25ºC unless stated otherwise)

Parameters		Units	Min	Target	Max
Supply Voltage (V_{dd})	Operating	V	2.7	3.3	5.25
I/O Pads Supply Voltage (V_{IO})		V	1.7	-	V_{dd}
Current Consumption	Operating	mA	0.6	1.0	1.7
	Standby	µA	-	-	10
Input Low Voltage		V	-	-	0.2 * V_{IO}
Input High Voltage		V	0.8 * V_{IO}	-	-
Input Pull-down Current		µA		0	
Analog Output Resistance(R_{out})		kΩ	24	32	40
Power Up Time[1]		ms		5*R_{out}*C	
A/D Conversion time		µs		200	
SPI Communication Rate[2]		MHz		1	
I^2C Communication Rate		KHz		400	

Notes:

1. Power up time is determined by 5 times the RC time constant of the user defined low pass filter.
2. SPI Communication Rate can be optimized for faster communication per the SPI timing diagram below.

KXP84 SPI Timing Diagram

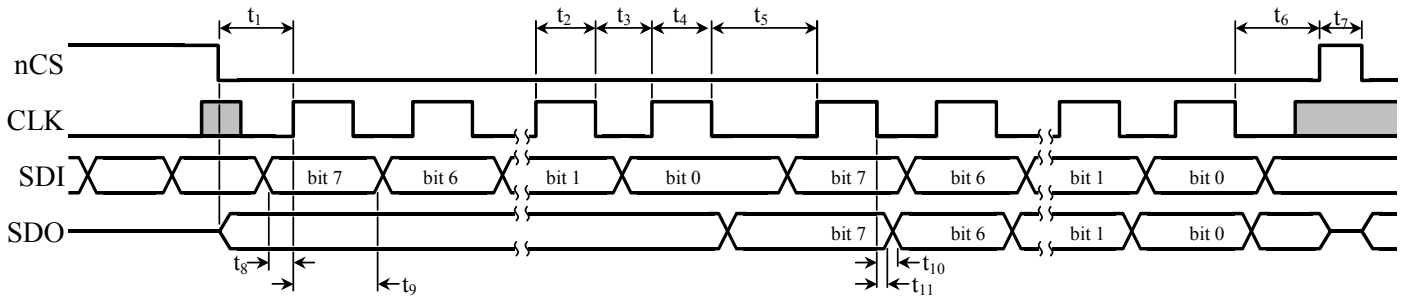


Table 3. SPI Timing

Number	Description	MIN	MAX	Units
t_1	nCS low to first CLK setup time	130	-	ns
t_2	CLK pulse width: high (Does not apply to the last bit of a byte.)	130	-	ns
t_3	CLK pulse width: low (Does not apply to the last bit of a byte.)	130	-	ns
t_4	CLK pulse width: high (Only on last bit of a byte.)	200	-	ns
t_5	CLK pulse width: low (Only on last bit of a byte.)	350	-	ns
t_6	nCS low after the final CLK falling edge	350	-	ns
t_7	nCS pulse width: high	130	-	ns
t_8	SDI valid to CLK rising edge	10	-	ns
t_9	CLK rising edge to SDI invalid	100	-	ns
t_{10}	CLK falling edge to SDO valid	-	130	ns
t_{11}	CLK falling edge to SDO invalid	0	-	ns
Notes	Recommended SPI CLK	1	-	us
	A/D conversion CLK hold (t_5)	200	-	us

Table 4. Environmental

Parameters		Units	Min	Target	Max
Supply Voltage (V_{dd})	Absolute Limits	V	-0.3	-	7.0
Operating Temperature Range		ºC	-40	-	85
Storage Temperature Range		ºC	-55	-	150
Mech. Shock (powered and unpowered)		g	-	-	5000 for 0.5ms
ESD	HBM	V	-	-	3000

Caution: ESD Sensitive and Mechanical Shock Sensitive Component, improper handling can cause permanent damage to the device.

The 14-pin DFN package conforms to European Union Directive 2002/95/EC on the restriction of the use of certain hazardous substances in electrical and electronic equipment (RoHS).

Soldering

Soldering recommendations available upon request or from www.kionix.com.

Application Schematic

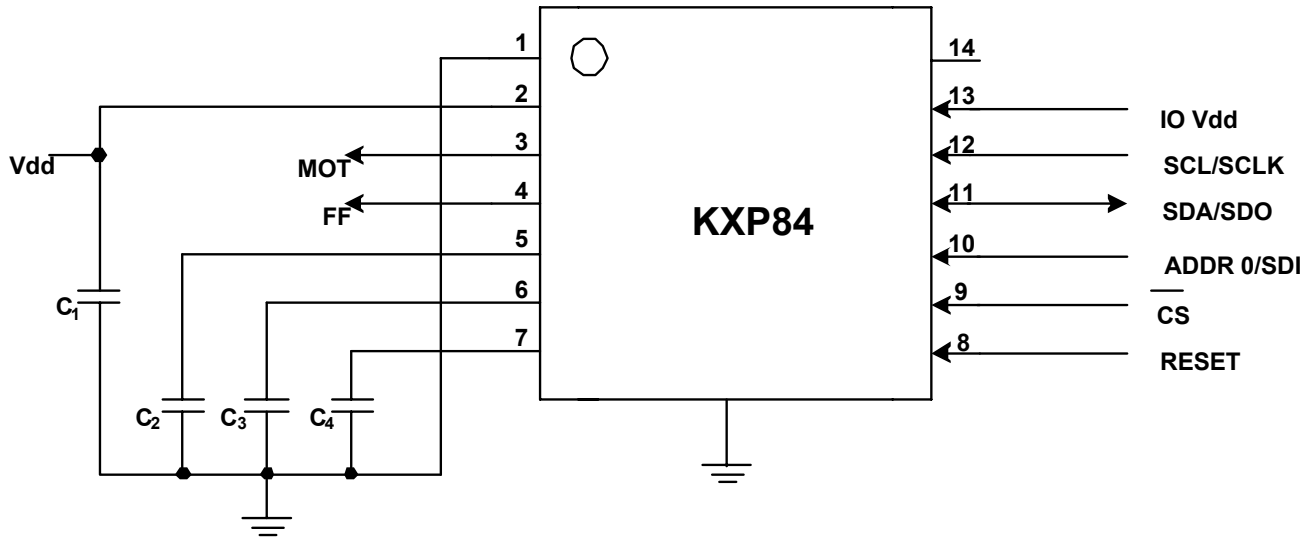


Table 5. KXP84 Pin Descriptions

Pin	Name	Description
1	GND	Ground
2	Vdd	The power supply input. Decouple this pin to ground with a 0.1uF ceramic capacitor (C_1).
3	MOT	Motion interrupt
4	FF	Free-fall interrupt
5	X Output	Analog output of the x-channel. Optionally, a capacitor (C_2) placed between this pin and ground will form a low pass filter.
6	Y Output	Analog output of y-channel. Optionally, a capacitor (C_3) placed between this pin and ground will form a low pass filter.
7	Z Output	Analog output of z-channel. Optionally, a capacitor (C_4)placed between this pin and ground will form a low pass filter.
8	Reset	Reset clears all KXP84 registers
9	nCS	SPI Chip Select and I^2C/SPI mode selection: (1 = I^2C mode, 0 = SPI mode)
10	ADDR0/SDI	I^2C programmable address bit/SPI Serial Data Input
11	SDA/SD0	I^2C Serial Data/SPI Serial Data Output
12	SCL/SCLK	I^2C Serial Clock/SPI Serial Clock
13	IO Vdd	The power supply input for the I/O pads
14	DNC	Do Not Connect
	Center Pad	Ground

Application Design Equations

The bandwidth is determined by the filter capacitors connected from pins 5, 6 and 7 to ground. The response is single pole. Given a desired bandwidth, f_{BW}, the filter capacitors are determined by:

$$C_2 = C_3 = C_4 = \frac{4.97x10^{-6}}{f_{BW}}$$

Note:
Self test and standby modes are enabled through the control registers.

KXP84 Interrupt Features

As shown in the application schematic, the KXP84 features a high-g motion interrupt (MOT) and a free-fall interrupt (FF). Each interrupt features independent, user-definable thresholds, debounce times, and latch/unlatch capabilities that are customized through the KXP84's embedded 8-bit registers.

High-g Motion Interrupt - The high-g motion interrupt goes high when a high-g event is detected. A high-g event occurs when the acceleration on any axis exceeds the high acceleration threshold for a certain amount of time. The high acceleration threshold and debounce time is set by the user during power up through the embedded 8-bit registers.

Free-fall Detection Interrupt - The free-fall interrupt goes high when a free-fall event is detected. A free-fall event occurs when the acceleration on all three accelerometer axes simultaneously falls below the low acceleration threshold for a certain amount of time. The low acceleration threshold and debounce time is set by the user during power up through the embedded 8-bit registers.

Test Specifications

! *Special Characteristics:*

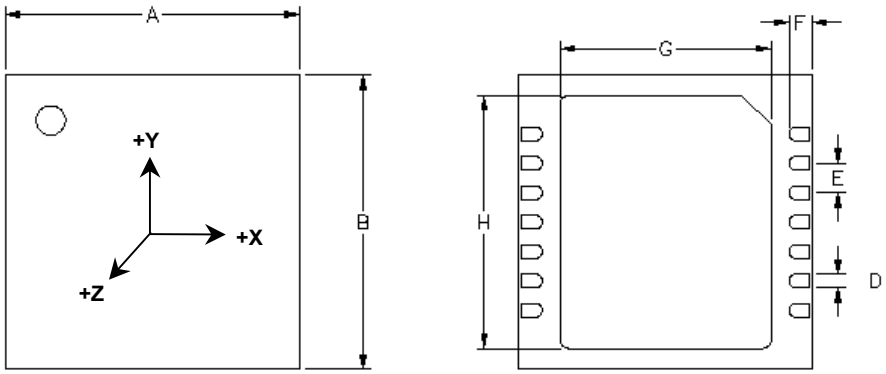
These characteristics have been identified as being critical to the customer. Every part is tested to verify its conformance to specification prior to shipment.

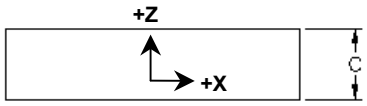
Table 6. Test Specifications

Parameter		Specification	Test Conditions
Zero-g Offset @ RT		2048 ± 102	25ºC, V_{dd} = 3.3V
Sensitivity @ RT		819 ± 25	25ºC, V_{dd} = 3.3V
Cross Axis Sensitivity		<3%	25ºC, V_{dd} = 3.3V
Current Consumption	Operating	0.6 <= I_{dd} <= 1.7mA	25ºC, V_{dd} = 3.3V

Package Dimensions and Orientation

5 x 5 x 1.2 mm DFN





Dimension	mm			inch		
	Min	Nom	Max	Min	Nom	Max
A		5.00			0.197	
B		5.00			0.197	
C	1.10	1.20	1.30	0.043	0.047	0.051
D	0.18	0.23	0.28	0.007	0.009	0.011
E		0.50			0.020	
F	0.35	0.40	0.45	0.014	0.016	0.018
G	3.50	3.60	3.70	0.138	0.142	0.146
H	4.20	4.30	4.40	0.165	0.169	0.173

All dimensions and tolerances conform to ASME Y14.5M-1994

When device is accelerated in +X, +Y or +Z direction, the corresponding output will increase.

KXP84 Digital Interfaces

The Kionix KXP84 digital accelerometer has the ability to communicate on both I^2C and SPI digital serial interface busses. This flexibility allows for easy system integration by eliminating analog-to-digital converter requirements and by providing direct communication with system micro-controllers. In doing so, all of the digital communication pins have shared responsibilities.

The serial interface terms and descriptions as indicated in Table 7 below will be observed throughout this document.

Term	Description
Transmitter	The device that transmits data to the bus.
Receiver	The device that receives data from the bus.
Master	The device that initiates a transfer, generates clock signals and terminates a transfer.
Slave	The device addressed by the Master.

Table 7. Serial Interface Terminologies

I^2C Serial Interface

The KXP84 has the ability to communicate on an I^2C bus. I^2C is primarily used for synchronous serial communication between a Master device and one or more Slave devices. The Master, typically a micro controller, provides the serial clock signal and addresses Slave devices on the bus. The KXP84 always operates as a Slave device during standard Master-Slave I^2C operation as shown in Figure 1 on the following page.

I^2C is a two-wire serial interface that contains a Serial Clock (SCL) line and a Serial Data (SDA) line. SCL is a serial clock that is provided by the Master, but can be held low by any Slave device, putting the Master into a wait condition. SDA is a bi-directional line used to transmit and receive data to and from the interface. Data is transmitted MSB (Most Significant Bit) first in 8-bit per byte format, and the number of bytes transmitted per transfer is unlimited. The I^2C bus is considered free when both lines are high.

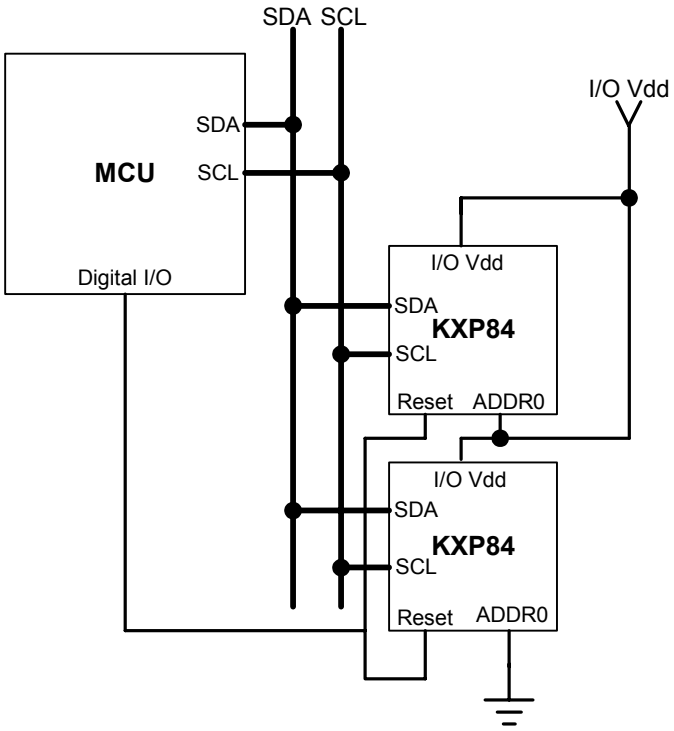


Figure 1 Multiple KXP84 I²C Connection

I²C Operation

Transactions on the I²C bus begin after the Master transmits a start condition (S), which is defined as a high-to-low transition on the data line while the SCL line is held high. The bus is considered busy after this condition. The next byte of data transmitted after the start condition contains the Slave Address (SAD) in the seven MSBs (Most Significant Bits), and the LSB (Least Significant Bit) tells whether the Master will be receiving data '1' from the Slave or transmitting data '0' to the Slave. When a Slave Address is sent, each device on the bus compares the seven MSBs with its internally-stored address. If they match, the device considers itself addressed by the Master. The KXP84's Slave Address is comprised of a programmable part and a fixed part, which allows for connection of multiple KXP84's to the same I²C bus. The Slave Address associated with the KXP84 is 001100X, where the programmable bit, X, is determined by the assignment of ADDR0 (pin 10) to GND or I/0 Vdd. Figure 1 above shows how two KXP84's would be implemented on an I²C bus.

It is mandatory that receiving devices acknowledge (ACK) each transaction. Therefore, the transmitter must release the SDA line during this ACK pulse. The receiver then pulls the data line low so that it remains stable low during the high period of the ACK clock pulse. A receiver that has been addressed, whether it is Master or Slave, is obliged to generate an ACK after each byte of data has been received. To conclude a transaction, the Master must transmit a stop condition (P) by transitioning the SDA line from low to high while SCL is high. The I²C bus is now free.

Writing to a KXP84 8-bit Register

Upon power up, the Master must write to the KXP84's control registers to set its operational mode. Therefore, when writing to a control register on the I^2C bus, as shown Sequence 1 on the following page, the following protocol must be observed: After a start condition, SAD+W transmission, and the KXP84 ACK has been returned, an 8-bit Register Address (RA) command is transmitted by the Master. This command is telling the KXP84 to which 8-bit register the Master will be writing the data. Since this is I^2C mode, the MSB of the RA command should always be zero (0). The KXP84 acknowledges the RA and the Master transmits the data to be stored in the 8-bit register. The KXP84 acknowledges that it has received the data and the Master transmits a stop condition (P) to end the data transfer. The data sent to the KXP84 is now stored in the appropriate register. The KXP84 automatically increments the received RA commands and, therefore, multiple bytes of data can be written to sequential registers after each Slave ACK as shown in Sequence 2 on the following page.

Reading from a KXP84 8-bit Register

When reading data from a KXP84 8-bit register on the I^2C bus, as shown in Sequence 3 on the next page, the following protocol must be observed: The Master first transmits a start condition (S) and the appropriate Slave Address (SAD) with the LSB set at '0' to write. The KXP84 acknowledges and the Master transmits the 8-bit RA of the register it wants to read. The KXP84 again acknowledges, and the Master transmits a repeated start condition (Sr). After the repeated start condition, the Master addresses the KXP84 with a '1' in the LSB (SAD+R) to read from the previously selected register. The Slave then acknowledges and transmits the data from the requested register. The Master does not acknowledge (NACK) it received the transmitted data, but transmits a stop condition to end the data transfer. Note that the KXP84 automatically increments through its sequential registers, allowing data reads from multiple registers following a single SAD+R command as shown below in Sequence 4 on the following page.

If a receiver cannot transmit or receive another complete byte of data until it has performed some other function, it can hold SCL low to force the transmitter into a wait state. Data transfer only continues when the receiver is ready for another byte and releases SCL. For instance, after the Master has requested to read acceleration data from the KXP84, the KXP84 can hold SCL low to force the Master into a wait state while it completes the A/D conversion. After the A/D conversion, the KXP84 will release SCL and transmit the acceleration data to the Master. Note that the KXP84 will hold for A/D conversions only if the CLKhld bit is set in CTRL_REGB.

Data Transfer Sequences

The following information clearly illustrates the variety of data transfers that can occur on the I^2C bus and how the Master and Slave interact during these transfers. Table 8 on the following page defines the I^2C terms used during the data transfers.

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

Table 8. I^2C Terms

Sequence 1. The Master is writing one byte to the Slave.

Master	S	SAD + W		RA		DATA		P
Slave			ACK		ACK		ACK	

Sequence 2. The Master is writing multiple bytes to the Slave.

Master	S	SAD + W		RA		DATA		DATA		P
Slave			ACK		ACK		ACK		ACK	

Sequence 3. The Master is receiving one byte of data from the Slave.

Master	S	SAD + W		RA		Sr	SAD + R			NACK	P
Slave			ACK		ACK			ACK	DATA		

Sequence 4. The Master is receiving multiple bytes of data from the Slave.

Master	S	SAD + W		RA		Sr	SAD + R			ACK		NACK	P
Slave			ACK		ACK			ACK	DATA		DATA		

SPI Interface

The KXP84 also utilizes an integrated Serial Peripheral Interface (SPI) for digital communication. The SPI interface is primarily used for synchronous serial communication between one Master device and one or more Slave devices. The Master, typically a micro controller, provides the SPI clock signal (SCLK) and

determines the state of Chip Select (nCS). The KXP84 always operates as a Slave device during standard Master-Slave SPI operation.

SPI is a 4-wire synchronous serial interface that uses two control and two data lines. With respect to the Master, the Serial Clock output (SCLK), the Data Output (MOSI) and the Data Input (MISO) are shared among the Slave devices. The Master generates an independent Chip Select (nCS) for each Slave device that goes low at the start of transmission and goes back high at the end. The Slave Data Output (SDO) line, remains in a high-impedance (hi-z) state when the device is not selected, so it does not interfere with any active devices. This allows multiple Slave devices to share a master SPI port as shown in Figure 2 below.

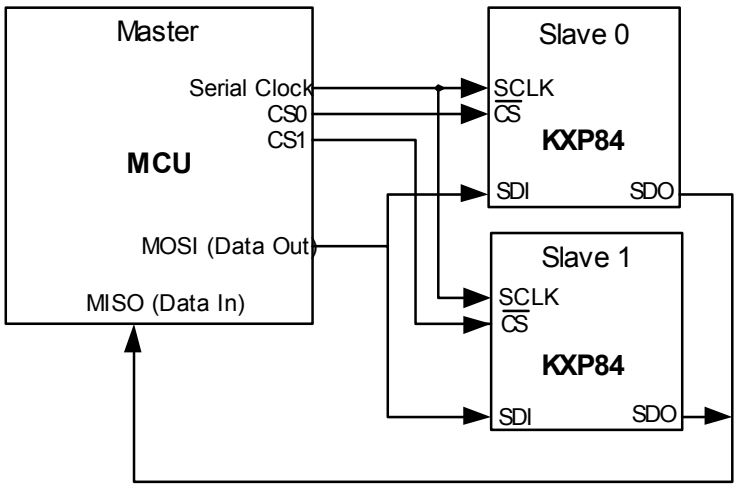


Figure 2 KXP84 SPI Connections

Read and Write Control Registers

The control registers embedded in the KXP84 have 8-bit addresses. Upon power up, the Master must write to the accelerometer's control registers to set its operational mode. On the falling edge of nCS, a 2-byte command is written to the appropriate control register. The first byte initiates the write to the appropriate register, and is followed by the user-defined, operational-mode byte. The MSB (Most Significant Bit) of the control register address byte will indicate "0" when writing to the register and "1" when reading from the register. This operation occurs over 16 clock cycles. All commands are sent MSB first, and the host must return nCS high for at least 130nS before the next data request. Figure 3 below shows the timing diagram for carrying out the 8-bit control register write operation.

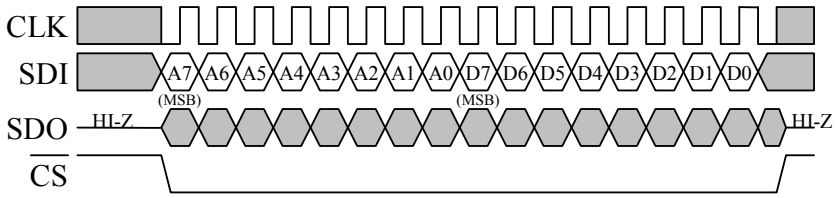


Figure 3 Timing Diagram for 8-Bit Control Register Write Operation

In order to read an 8-bit control register, an 8-bit read command must be written to the accelerometer to initiate the read. The MSB of this control register address byte will indicate "0" when writing to the register and "1" when reading from the register. Upon receiving the command, the accelerometer returns the 8-bit operational-mode data stored in the appropriate control register. This operation also occurs over 16 clock cycles. All returned data is sent MSB first, and the host must return nCS high for at least 130nS before the next data request. Figure 4 shows the timing diagram for an 8-bit control register read operation.

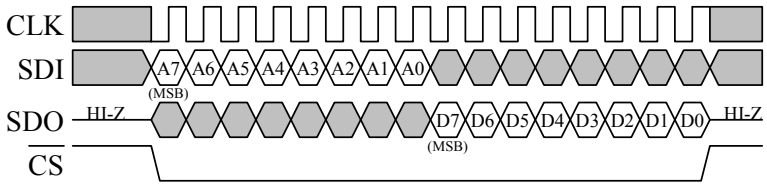


Figure 4 Timing Diagram for 8-Bit Control Register Read Operation

Accelerometer Read Back Operation

The KXP84 has an onboard 12-bit ADC that can sample, convert and read back sensor data at any time. Transmission of an 8-bit axis-conversion command (see Table 10) begins on the falling edge of nCS. The MSB of this command indicates if you are writing to (0) or reading from (1) the register. After the eight clock cycles used to send the command, the host must hold SCLK low for at least 200µs during the A/D conversion time. Note that all returned data is sent MSB first. Once the data is received, nCS must be returned high for at least 130nS before the next data request. Figure 5 on the following page shows the timing diagram for the accelerometer 12-bit ADC read operation.

The Read Back Operation is a 3-byte SPI command. The first byte of SDI contains the command to convert one of the axes. The second and third bytes of SDO contain the 12 bits of the A/D result plus four bits of padding in the LSB to make a total of 16 bits. See Figure 6 below.

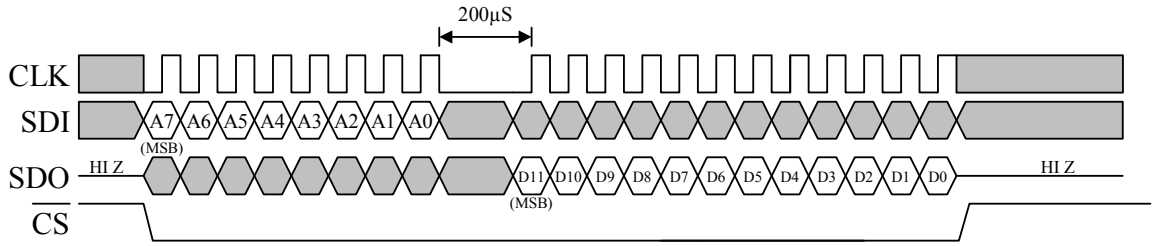


Figure 5 Timing Diagram for an A/D conversion and 12-Bit data read operation.

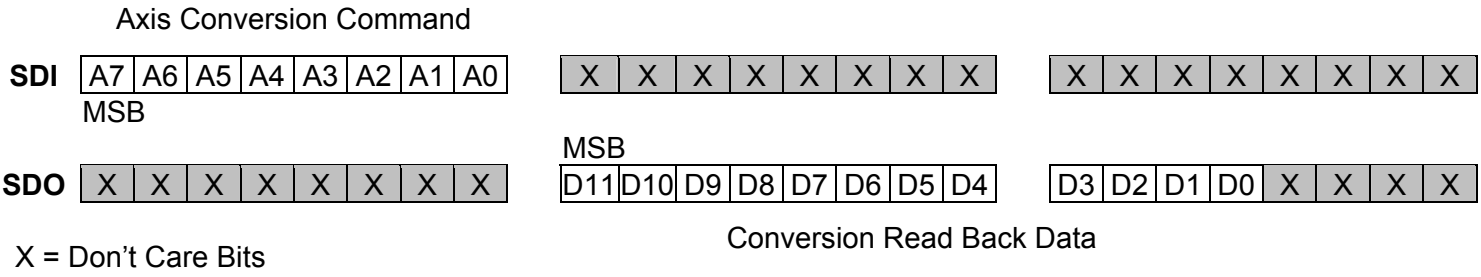


Figure 6 Register Diagram for 12-Bit ADC Read Operation

Digital Accelerometer SPI Sequence

An example of a SPI sequence for reading sensor data is as follows:

- Power up digital accelerometer
- nCS low to select
- Write operational mode commands to the 8-bit control registers
  CTRL_REGB and CTRL_REGC
- nCS high for at least 130nS
- nCS low to select
- Send convert axis command
  There should be a minimum of 200µs between the first and second bytes in order to give the A/D conversion adequate time to complete.
- The 12-bit A/D data is read to the second and third SDO bytes.
  The KXP84 auto-increments register transmits on SDO. Therefore, Y-axis, Z-axis, CTRL_REGA, CTRL_REGB, and CTRL_REGC will follow the two X-axis bytes automatically.
- After receiving the last byte of required data, return nCS high for at least 130nS to reset the auto-increment.
- Repeat data read cycle
- Recommend reading X-axis, Y-axis, Z-axis, and the three Control Registers for each read cycle to verify the mode selections and status

KXP84 Embedded Registers

The KXP84 has 13 embedded 8-bit registers that are accessible by the user. This section contains the addresses for all embedded registers and also describes bit functions of each register. Tables 9 and 10 below provide a listing of the accessible 8-bit registers and their addresses when in I^2C mode and SPI Mode.

Register Name	Type Read/Write	Address Hex	Address Binary
XOUT_H	R	0x00	0000 0000
XOUT_L	R	0x01	0000 0001
YOUT_H	R	0x02	0000 0010
YOUT_L	R	0x03	0000 0011
ZOUT_H	R	0x04	0000 0100
ZOUT_L	R	0x05	0000 0101
FF_INT	R/W	0x06	0000 0110
FF_DELAY	R/W	0x07	0000 0111
MOT_INT	R/W	0x08	0000 1000
MOT_DELAY	R/W	0x09	0000 1001
CTRL_REGC	R/W	0x0A	0000 1010
CTRL_REGB	R/W	0x0B	0000 1011
CTRL_REGA	R	0x0C	0000 1100

Table 9. I^2C Mode Register Map

Register Name	Type Read/Write	Read Address Hex	Read Address Binary	Write Address Hex	Write Address Binary
XOUT_H	R	0x80	1000 0000	xxxx	xxxx xxxx
XOUT_L	R	0x81	1000 0001	xxxx	xxxx xxxx
YOUT_H	R	0x82	1000 0010	xxxx	xxxx xxxx
YOUT_L	R	0x83	1000 0011	xxxx	xxxx xxxx
ZOUT_H	R	0x84	1000 0100	xxxx	xxxx xxxx
ZOUT_L	R	0x85	1000 0101	xxxx	xxxx xxxx
FF_INT	R/W	0x86	1000 0110	0x06	0000 0110
FF_DELAY	R/W	0x87	1000 0111	0x07	0000 0111
MOT_INT	R/W	0x88	1000 1000	0x08	0000 1000
MOT_DELAY	R/W	0x89	1000 1001	0x09	0000 1001
CTRL_REGC	R/W	0x8A	1000 1010	0x0A	0000 1010
CTRL_REGB	R/W	0x8B	1000 1011	0x0B	0000 1011
CTRL_REGA	R	0x8C	1000 1100	xxxx	xxxx xxxx

Table 10. SPI Mode Register Map

Register Descriptions

XOUT_H

X-axis accelerometer output most significant byte

XOUTD11	XOUTD10	XOUTD9	XOUTD8	XOUTD7	XOUTD6	XOUTD5	XOUTD4

XOUT_L

X-axis accelerometer output least significant byte

XOUTD3	XOUTD2	XOUTD1	XOUTD0	X	X	X	X

YOUT_H

Y-axis accelerometer output most significant byte

YOUTD11	YOUTD10	YOUTD9	YOUTD8	YOUTD7	YOUTD6	YOUTD5	YOUTD4

YOUT_L

Y-axis accelerometer output least significant byte

YOUTD3	YOUTD2	YOUTD1	YOUTD0	X	X	X	X

ZOUT_H

Z-axis accelerometer output most significant byte

ZOUTD11	ZOUTD10	ZOUTD9	ZOUTD8	ZOUTD7	ZOUTD6	ZOUTD5	ZOUTD4

ZOUT_L

Z-axis accelerometer output least significant byte

ZOUTD3	ZOUTD2	ZOUTD1	ZOUTD0	X	X	X	X

CTRL_REGA

Read-only status register

X	X	X	X	X	Parity	MOTI	FFI

FFI *reflects the status of the free-fall interrupt. When FFI = 1, the free-fall interrupt pin is high. When FFI = 0, the free-fall interrupt pin is low. The free-fall interrupt is reset by setting FFI = 0.*

MOTI *reflects the status of the motion interrupt. When MOTI = 1, the motion- interrupt pin is high. When MOTI = 0, the motion-interrupt pin is low. The motion interrupt is reset by setting MOTI = 0.*

Parity *reports on even (0) or odd (1) EEPROM parity. A properly functioning KXP84 will return even (0) EEPROM parity. This bit is read-only and operates independently of the other modes.*

CTRL_REGB

Read/write control register

CLKhld	nENABLE	ST	0	0	MOTIen	FFIen	FFMOTI

FFMOTI *determines the pin configuration of the free-fall and motion interrupt.*

FFMOTI = 1 – puts (logical OR) both the motion interrupt and the free-fall interrupt flags on pin 4

FFMOTI = 0 - puts the motion interrupt flag on pin 3 and the free-fall interrupt flag on pin 4

FFIen *enables the freefall interrupt.*

FFIen = 1 - an interrupt will be generated when the KXP84 is in a predetermined free-fall state

FFIen = 0 – a free-fall interrupt is never generated

FF_Int and FF_Delay must be set prior to enabling the freefall interrupt.

MOTIen *enables the motion interrupt.*

MOTIen = 1 - an interrupt will be generated when the KXP84 is in a predetermined motion state

MOTIen = 0 – a motion interrupt is never generated

MOT_Int and MOT_Delay must be set prior to enabling the freefall interrupt.

Self test *activates the self-test function for the sensor elements on all three axes. A correctly functioning KXP84 will increase all channel outputs when Self test = 1 and nEnable = 0. This bit can be read or written.*

nEnable *powers up the KXP84 for operation.*

Enable = 1 – low-power standby

Enable = 0 – normal operation

CLKhld *allows the KXP84 to hold the serial clock, SCL, low in I^2C mode to force the transmitter into a wait state during A/D conversions.*

CLKhld = 1 – SCL held low during A/D conversions

CLKhld = 0 – SCL unaffected

CLKhld should be set to 0 when nEnable is set to 1 (disabled) to prevent potential holding of the CLK line.

CTRL_REGC

Read/write control register

X	X	X	FFLatch	MOTLatch	0	IntSpd1	IntSpd0

IntSpd0 *is the first of two bits used to select the rate at which the accelerometer is sampled when debouncing a potential interrupt event. See Table 11 below.*

IntSpd1 *is the second of two bits used to select the rate at which the accelerometer is sampled when debouncing a potential interrupt event. See Table 11 below.*

IntSpd1	IntSpd0	Interrupt Frequency
0	0	250Hz
0	1	1KHz
1	0	4KHz
1	1	16KHZ

Table 11. Sampling Frequencies

MOTLatch *switches the motion interrupt function between latching and non-latching as shown in Figures 7 and 8*

MOTLatch = 0 - The motion interrupt output will go high whenever the criterion for motion detection is met. The output will return low when the criterion is not met.

MOTLatch = 1 - The motion interrupt output will go high whenever the criterion for motion detection is met. The output will remain high until the MOTIen bit in CTRL_REGB is cycled low.

Typical Motion Interrupt Example (nonLatching, count up/reset)

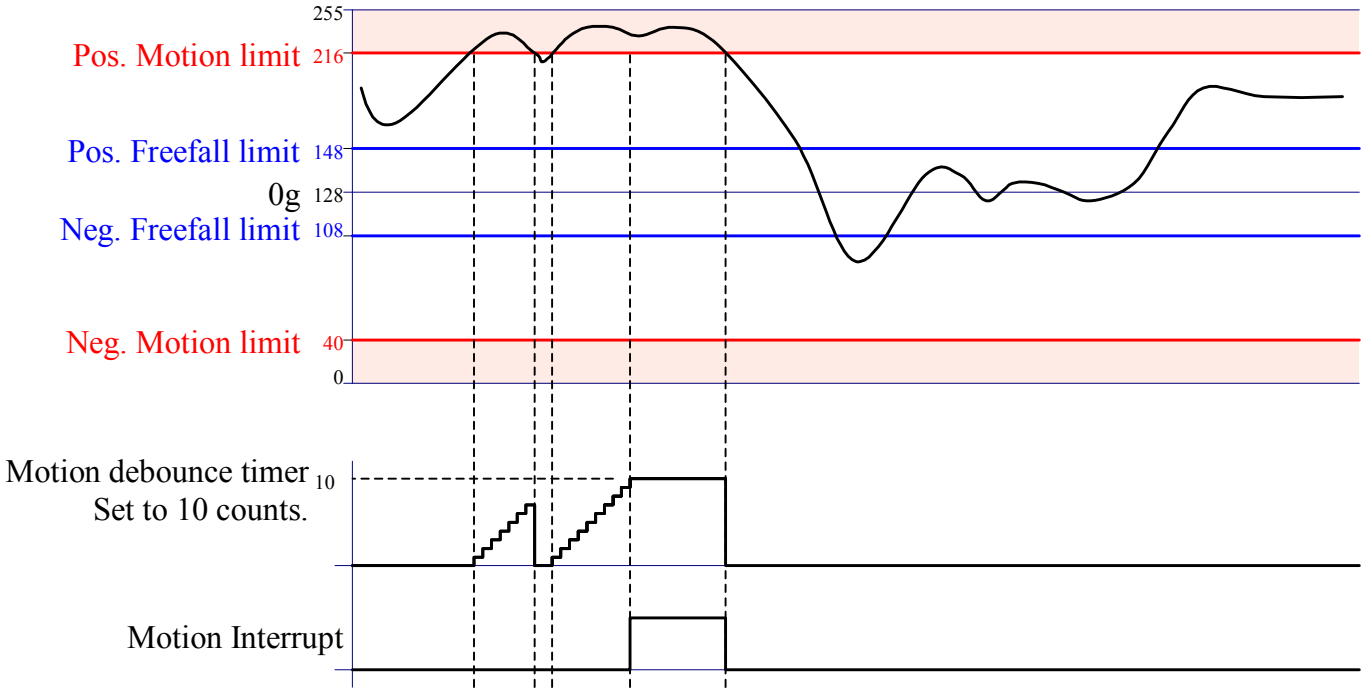


Figure 7. Typical Motion Interrupt Example (MOTLatch = 0)

Typical Motion Interrupt Example (Latching, count up/reset)

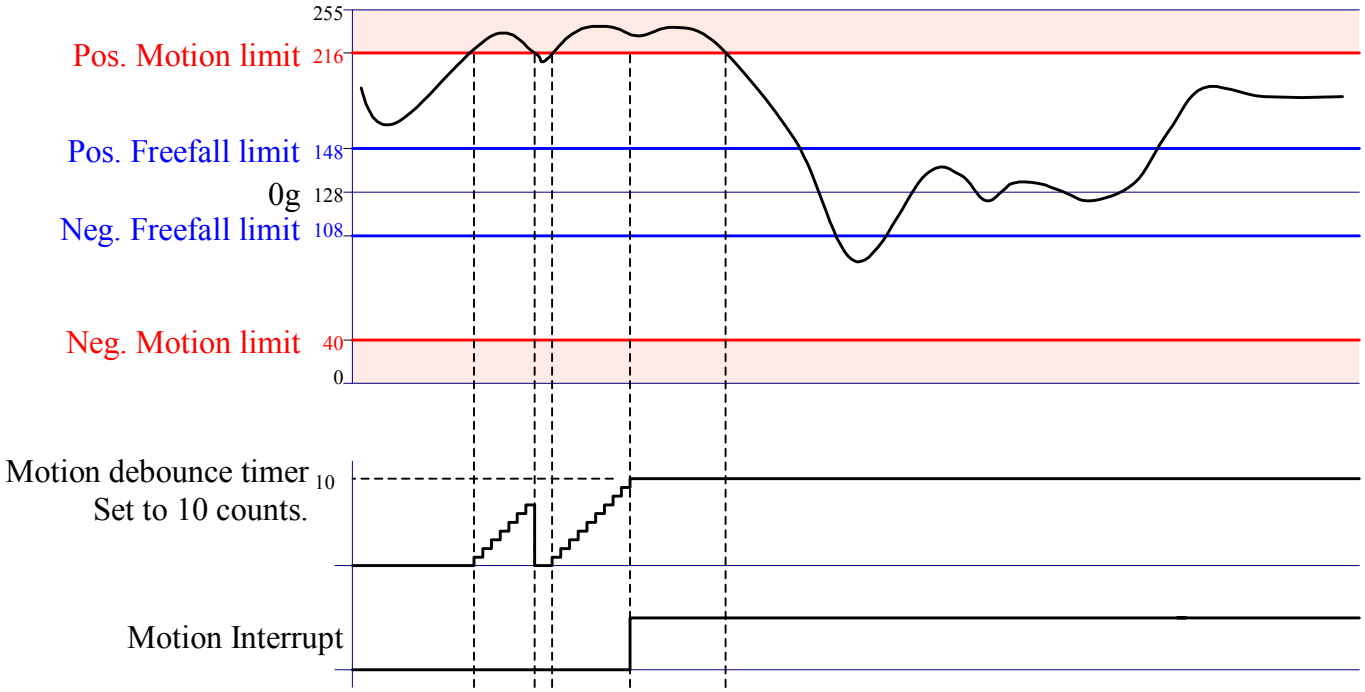


Figure 8. Typical Motion Interrupt Example (MOTLatch = 1)

FFLatch *switches the free-fall interrupt function between latching and non-latching as shown in Figures 9 and 10.*

FFLatch = 0 - The free-fall interrupt output will go high whenever the criterion for free-fall detection is met. The output will return low when the criterion is not met.

FFLatch = 1 - The free-fall interrupt output will go high whenever the criterion for free-fall detection is met. The output will remain high until FFIen bit in CTRL_REGB is cycled low.

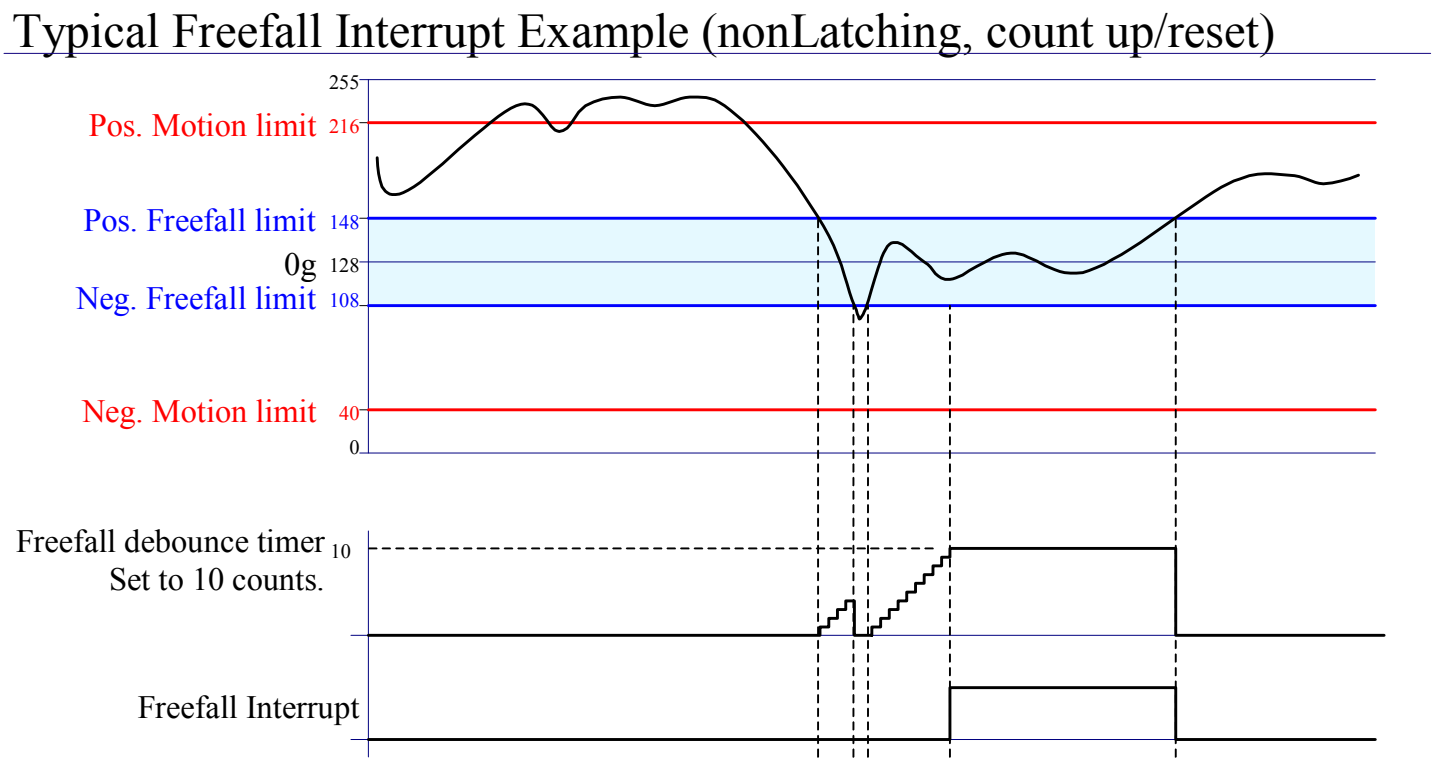


Figure 9. Typical Free-fall Interrupt Example (FFLatch = 0)

Typical Freefall Interrupt Example (Latching, count up/reset)

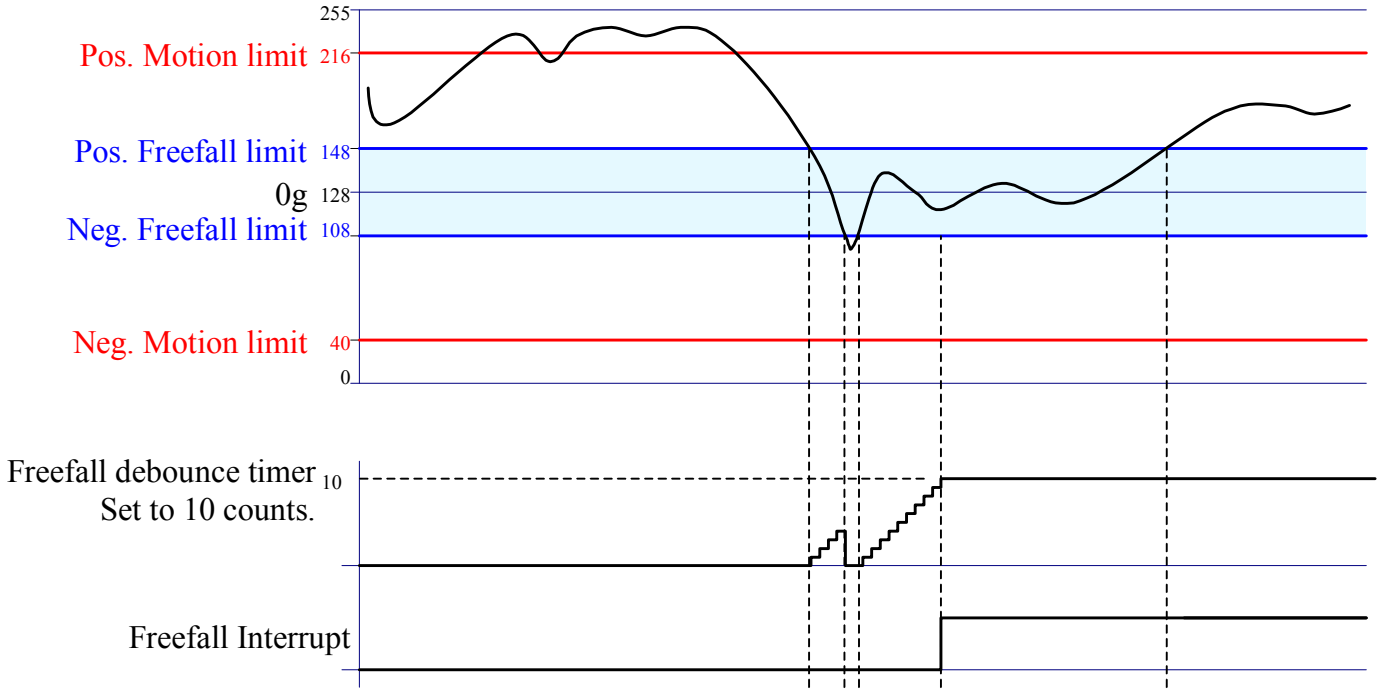


Figure 10. Typical Free-fall Interrupt Example (FFLatch = 1)

FF_INT

Sets the free-fall interrupt threshold to this value

FI7	FI6	FI5	FI4	FI3	FI2	FI1	FI0

FF_DELAY

Sets the free-fall delay/debounce time to this value

FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

Free-fall Detect

The KXP84 features a free-fall interrupt that sends a flag through pin 4 when the accelerometer senses a free-fall event. A free-fall event is evident when all three accelerometer axes simultaneously fall below a certain acceleration threshold for a set amount of time. The KXP84 gives the user the option to define the acceleration threshold value through the FF_INT 8-bit register where 256 counts cover the g range of the accelerometer. Equation 1 below shows how to calculate the acceleration threshold based on FF_INT and Sensitivity.

Threshold (g) = 16*FF_INT (counts)*Sensitivity (g/count)

Equation 1. Acceleration Threshold Calculation

Through the FF_DELAY 8-bit register, the user can set the amount of time all three accelerometer axes must simultaneously remain below the FF_INT acceleration threshold before the free-fall interrupt flag is sent through pin 4. This delay/debounce time is defined by the available 0 to 255 counts, which represent accelerometer samples taken at the rate defined by IntSpd0 and IntSpd1. Equation 2 below shows how to calculate the debounce time based on FF_DELAY, IntSpd0 and IntSpd1.

Delay (sec) = FF_DELAY (counts)/Interrupt Sampling Rate (Hz)

Equation 2. Debounce Calculation

When the Freefall interrupt is enabled the part must not be in a physical state that would trigger the freefall interrupt or the delay will not be correct for the present freefall.

MOT_INT

Sets the motion activated interrupt acceleration threshold

MI7	MI6	MI5	MI4	MI3	MI2	MI1	MI0

MOT_DELAY

Sets the motion activated delay/debounce time to this value

MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0

Motion Detect

The KXP84 also features a high-g motion interrupt that sends a flag through pin 3 when the accelerometer senses a high-g acceleration. A high-g acceleration is evident when any of the three accelerometer axes sense acceleration above a certain threshold for a set amount of time. The KXP84 gives the user the option to define the acceleration threshold value through the MOT_INT 8-bit register where 256 counts cover the g range of the accelerometer. Equation 3 shows how to calculate the acceleration threshold based on MOT_INT and Sensitivity.

Threshold (g) = 16*MOT_INT (counts)*Sensitivity (g/count)

Equation 3. Acceleration Threshold Calculation

Through the MOT_DELAY 8-bit register, the user can set the amount of time that any of the three accelerometer axes has to sense acceleration above a certain threshold before the motion interrupt flag is sent through pin 3. This delay/debounce time is defined by the available 0 to 255 counts, which represent accelerometer samples taken at the rate defined by IntSpd0 and IntSpd1. Equation 4 shows how to calculate the debounce time based on MOT_DELAY, IntSpd0 and IntSpd1.

Delay (sec) = MOT_DELAY (counts)/Interrupt Sampling Rate (Hz)

Equation 4. Debounce Calculation

When the Motion interrupt is enabled the part must not be in a physical state that would trigger the motion interrupt or the delay will not be correct for the present motion.

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Kionix:
KXP84-2050