



# High Pass Filtered Data and Transient Detection Using the MMA8450Q

by: Kimberly Tuck  
Applications Engineer

## 1.0 Introduction

The acceleration readings from the sensor couple both static and dynamic accelerations. Many tilt based applications such as orientation detection, scrolling and tilt based gaming apps use the accelerometer analyzing the change in static acceleration due to gravity only. These functions are often best when they've been low pass filtered because the high frequency data is considered noise. There are many functions where the accelerometer must analyze the dynamic acceleration. Functions such as taps, flicks, shakes and step counting are based on the analysis of the change in the acceleration with no regard for the static acceleration. These functions interpret the acceleration threshold change and direction of motion. It is simpler to interpret the dynamic acceleration data when the static component has been removed.

The MMA8450Q has a built in high pass filter. The output data goes through the high pass filter, eliminating the offset (DC) and low frequencies (well below the cut-off). The high pass filter cut-off frequency can be set by the user to four different frequencies which are dependent on the Output Data Rate (ODR). A higher cut-off frequency ensures the DC data or slower moving data will be filtered out, allowing only the higher frequencies to pass. The MMA8450Q also contains an embedded transient function which analyzes the high pass filtered data allowing the user to set a preset threshold with an added debounce counter to ensure the event happens for a set duration. The threshold represents a change in acceleration to be exceeded. This embedded function can be routed to an interrupt pin.

### 1.1 Key Words

Accelerometer, Static Acceleration, Image Stabilization, High Pass Filter, Transient, Delta Data, 3 Axis-Accelerometer, Embedded Motion Detection, Embedded Transient Detection, Vibration, Shake, Directional Shake, Displacement, Velocity, Sensor, Interrupt, DBCNTM Bit

### TABLE OF CONTENTS

<b>1.0 Introduction</b> .....	<b>1</b>
1.1 Key Words .....	1
1.2 Summary .....	1
<b>2.0 MMA8450Q Consumer 3-axis Accelerometer 3 x 3 x 1 mm</b> .....	<b>2</b>
2.1 Key Features of the MMA8450Q .....	2
2.2 Two (2) Programmable Interrupt Pins for 8 Interrupt Sources .....	3
2.3 Application Notes for the MMA8450Q .....	3
<b>3.0 Comparison of Motion Detection and Transient Detection</b> .....	<b>3</b>
3.1 Use Case 1 .....	3
3.2 Use Case 2 .....	4
<b>4.0 Transient Settings and Interpretation of the Data</b> .....	<b>5</b>
4.1 High Pass Filter Values .....	5
<b>5.0 Using the Delta Data to Write Algorithms such as Directional Shake</b> .....	<b>7</b>
<b>6.0 Configuring the Embedded Transient Detection</b> .....	<b>9</b>
6.1 Register 0x2B: TRANSIENT_CFG Transient Configuration Register .....	9
6.2 Register 0x2D: TRANSIENT_THS Transient Threshold Register .....	10
6.3 Register 0x2E: TRANSIENT_COUNT Transient Debounce Register .....	10
6.4 Register 0x2C: TRANSIENT_SRC Transient Source Register .....	10
6.5 Configuring the Transient Detection to an Interrupt Pin .....	11
6.6 Reading the System Interrupt Status Source Register .....	11
<b>7.0 Details for Configuring the MMA8450Q for Transient Detection</b> .....	<b>12</b>
7.1 Example Steps for Configuring Transient Detection .....	12

## 1.2 Summary

- A. The high pass filter is used to eliminate the static acceleration reading the delta X, Y, Z data.
- B. Different cut off frequencies from the high pass filter affect the output delta data.
- C. The high pass filtered data is useful for writing algorithms which require an analysis of the change in acceleration and simplifies the data analysis by eliminating the static acceleration.
- D. The transient function is interrupt driven and saves on current consumption and unnecessary processing because the logic is internal to the sensor.
- E. An algorithm for directional shake can be reliably detected by analyzing the (high pass filter) delta data at 400 Hz with the HPF cut off frequency set to 4 Hz.
- F. The delta data can be used for image stability applications, compensating for the jitter of the user's hands and/or from the environment.
- G. The transient function can be used to detect a change in acceleration that exceeds a set threshold. This is slightly different than the motion detection function.
- H. The latch will hold the EA bit until the status register is read to indicate an event is active, but the other bits in the status register are never latched. The status must be read immediately to determine the condition of the axes when an event occurs.
- I. The transient threshold and debounce counter can be changed in either standby or active mode to allow for adjustments after the part has transitioned from wake to sleep mode.

## 2.0 MMA8450Q Consumer 3-axis Accelerometer 3 x 3 x 1 mm

The MMA8450Q has a selectable dynamic range of  $\pm 2g$ ,  $\pm 4g$  and  $\pm 8g$  with sensitivities of 1024 counts/g, 512 counts/g and 256 counts/g respectively. The device offers either 8-bit or 12-bit XYZ output data for algorithm development. The chip shot and pinout are shown in [Figure 1](#).

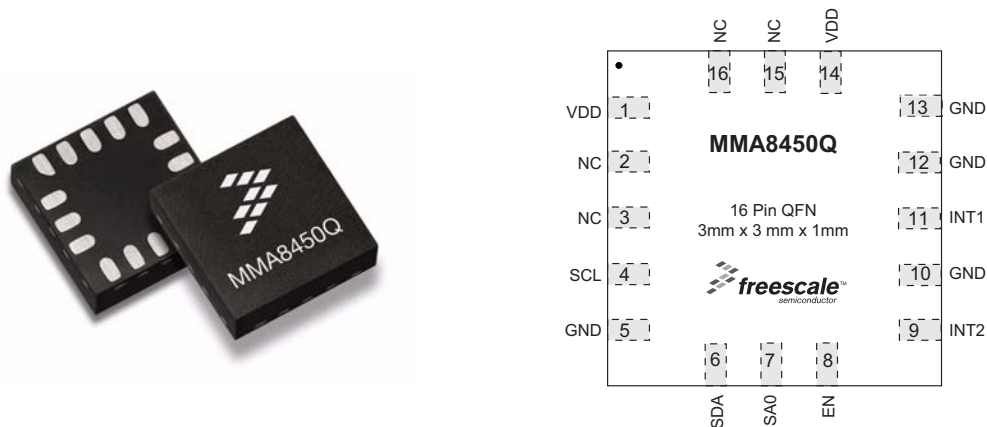


Figure 1. MMA8450Q Consumer 3-axis Accelerometer 3 x 3 x 1 mm

### 2.1 Key Features of the MMA8450Q

- 1. Shutdown Mode: Typical  $< 1 \mu A$ , Standby Mode  $3 \mu A$
- 2. Low Power Mode current consumption ranges from  $27 \mu A$  (1.56 - 50 Hz) to  $120 \mu A$  (400 Hz)
- 3. Normal Mode current consumption ranges from  $42 \mu A$  (1.56 - 50 Hz) to  $225 \mu A$  (400 Hz)
- 4. I<sup>2</sup>C digital output interface (operates up to 400 kHz Fast Mode)
- 5. 12-bit and 8-bit data output, 8-bit high pass filtered data output
- 6. Post Board Mount Offset  $< \pm 50$  mg typical
- 7. Self Test X, Y and Z axes

## 2.2 Two (2) Programmable Interrupt Pins for 8 Interrupt Sources

1. Embedded 4 channels of Motion detection
  - a. Freefall or Motion detection: 2 channels
  - b. Tap detection: 1 channel
  - c. Transient detection: 1 channel
2. Embedded orientation (Portrait/Landscape) detection with hysteresis compensation
3. Embedded automatic ODR change for auto-wake-up and return-to-sleep
4. Embedded 32 sample FIFO
5. Data Ready Interrupt

## 2.3 Application Notes for the MMA8450Q

The following is a list of Freescale Application Notes written for the MMA8450Q:

- **AN3915**, *Embedded Orientation Detection Using the MMA8450Q*
- **AN3916**, *Offset Calibration of the MMA8450Q*
- **AN3917**, *Motion and Freefall Detection Using the MMA8450Q*
- **AN3918**, *High Pass Filtered Data and Transient Detection Using the MMA8450Q*
- **AN3919**, *MMA8450Q Single/Double and Directional Tap Detection*
- **AN3920**, *Using the 32 Sample First In First Out (FIFO) in the MMA8450Q*
- **AN3921**, *Low Power Modes and Auto-Wake/Sleep Using the MMA8450Q*
- **AN3922**, *Data Manipulation and Basic Settings of the MMA8450Q*
- **AN3923**, *MMA8450Q Design Checklist and Board Mounting Guidelines*

## 3.0 Comparison of Motion Detection and Transient Detection

The transient function is different from the motion detection embedded function in MMA8450Q. Refer to AN3917 for details on the motion detection. The intended use of these functions is different.

- A. The embedded motion function triggers an interrupt when any of the enabled axes has exceeded a set acceleration threshold.
  - Motion detection is useful when the device may be tilted in an orientation and the interest is to know if the acceleration in one axis has crossed a specified threshold amount measured from 0g. The motion detection analyzes change in threshold considering both static and dynamic acceleration.
- B. The transient detection embedded function compares whether any of the enabled axes has exceeded a set change in acceleration disregarding the static acceleration. The data that is compared to the threshold to be exceeded is high pass filtered data.
  - Transient detection is useful when the device may be tilted in an orientation (i.e., there is some non-zero reading on one or more of X, Y and Z) and the interest is more on the dynamic acceleration crossing a specified threshold amount in a short amount of time (frequency being the reciprocal of time) regardless of what the current value is.

### Example

If a change in acceleration of 0.5g in any axes was the target for an application, then it would be challenging to use the embedded motion detection function for detecting this change. The reason for this can be better understood with an example: Assume that the device was oriented such that  $X = 0.24g$ ,  $Y = 0.44g$  and  $Z = 0.82g$ ; it would be challenging to detect a 0.5g change in acceleration in any of the axes using the motion detect function since the new readings have to be outside the range  $X = -0.34g$  to  $0.74g$ ,  $Y = -0.04g$  to  $0.94g$  and  $Z = 0.3g$  to  $1.32g$ . Because only a single threshold setting can be chosen for each axis when using the motion detect embedded function and because of the need to detect acceleration outside a range, it would, therefore, be seen as impossible to choose a threshold that would detect a 0.5 change in acceleration for any of the axes.

### 3.1 Use Case 1

#### Use Case 1 Transient: Unknown Orientation with Specific Acceleration Change for Detection

A use case example for transient detection may be when the device containing the accelerometer is in an environment of an unknown orientation or changing orientation and a small amplitude, but quick disturbance must be detected, e.g., when the device is accelerating through 0g to +1g to -1g and a subtle impact force (change in acceleration of 0.5g) is used as an indication to wake the device or as a motion signature of an event. The accelerometer reading combines the gravity and the linear acceleration reading. It is much simpler to decouple the two by eliminating the effect of gravity. Using the example above, assume that the device was oriented such that at given time T,  $X = 0.24g$ ,  $Y = 0.44g$  and  $Z = 0.82g$ . If transient detection is enabled on all 3 axes with a threshold of 0.5g and an ODR of 100 Hz, any time that X goes outside of the range -0.34g to 0.74g within T + 10 ms,

a transient detect interrupt would be generated. Eliminating the static acceleration on the device simplifies the condition to focus on the change in acceleration only, since the orientation is of no consequence in this situation. Figure 2 demonstrates that the sensor detects both the static and the dynamic acceleration. Decoupling the static from the dynamic acceleration can simplify the analysis.

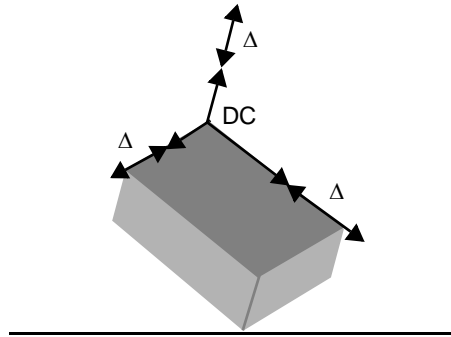


Figure 2. Device in an Unknown Orientation While Detecting Change in Acceleration

### 3.2 Use Case 2

#### Use Case 2: Image Stabilization

Another use case example for using data processed by the high pass filter is for users viewing their PDA while in a car/bus or on a train to provide image stabilization in a shaky environment. Small motions/vibrations can be compensated for with the change in X, Y and Z movement through a stabilization routine, while a large change in motion could change the mode of the device.

As shown in Figure 3, an image on a PDA that appears to be shaking can dynamically be compensated by shifting the screen content in the opposite direction of the shake. The data must first be filtered to eliminate the gravity component to analyze the change in acceleration of the axes. A mass spring damper model is a reasonable way to analyze the system to reduce display shaking since the spring allows the mass to move in the opposite direction of the shake and dampers would prevent oscillation of the mass. This type of model only requires the acceleration to compensate for the shaking. The parameters of the model can be tuned based on the amplitudes and frequency of the oscillations in the environment since different environments may affect the usability of the stabilization capabilities. For more information on this type of image stabilization technique there have been various publications written for example, Rahmati et al<sup>1</sup>.

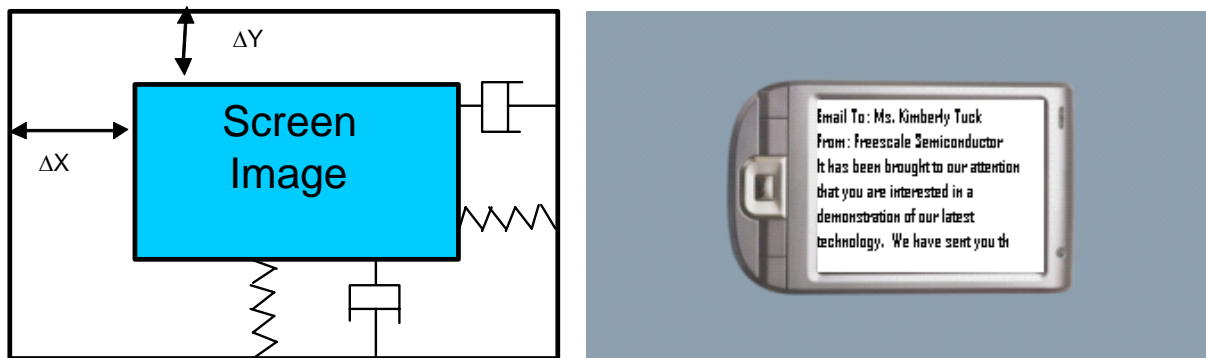


Figure 3. PDA Image Stabilization in a Train, Bus, Car, or Vibrating Environment

1. A. Rahmati, C. Shepard, L. Zhong, "NoShake: Content Stabilization for Shaking Screens of Mobile Devices", in *Proc. IEEE Int. Conf. Pervasive Computing and Communication (PerCom)*, March 2009.

## 4.0 Transient Settings and Interpretation of the Data

Figure 4 is a display of the 8-bit delta data (high pass filtered data registers 0xC-0x0E) in 8g mode. The high pass filter is set to 4 Hz with the sampling rate at 400 Hz. The delta data is 8-bit data, and therefore the sensitivity in 8g mode is 16LSB/g. The embedded block is set to detect when the delta data exceeds 0.5g. This shows that the device can be vibrating, or bouncing at a low acceleration level and the event will not be detected until the change in acceleration exceeds 0.5g.

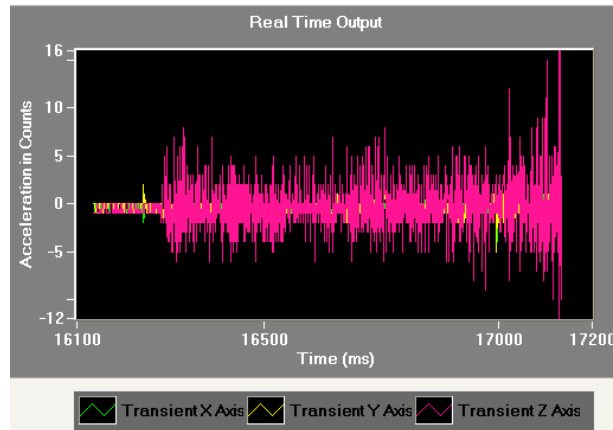


Figure 4. Delta Data in the Embedded Transient Function with Threshold  $\pm 0.5g$

### 4.1 High Pass Filter Values

The high pass filter values available scale with the output data rate in the MMA8450Q. Table 1 and Table 2, list the output high pass filter register where the cut off frequency values are stored for every corresponding data rate. FC in the table header refers to the cut-off frequency on the high pass filter.

Table 1. High Pass Filter Register 0x17 (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	SEL1	SEL0

Table 2. HP\_FILTER\_CUTOFF Setting Options

SEL1	SEL0	FC @ ODR = 400 Hz	FC (Hz) @ ODR = 200 Hz	FC (Hz) @ ODR = 100 Hz	FC (Hz) @ ODR = 50 Hz	FC (Hz) @ ODR = 12.5 Hz	FC (Hz) @ ODR = 1.563 Hz
0	0	4	2	1	0.5	0.125	0.01
0	1	2	1	0.5	0.25	0.063	0.007
1	0	1	0.5	0.25	0.125	0.031	0.004
1	1	0.5	0.25	0.125	0.062	0.016	0.002

Figure 5 shows a 0.8 Hz -1.5 Hz Z-axis only motion at 400 Hz ODR with the filter set at 4 Hz. Much of the motion at 1 Hz is filtered out and the accelerometer reading remains between 0 to -1 counts.

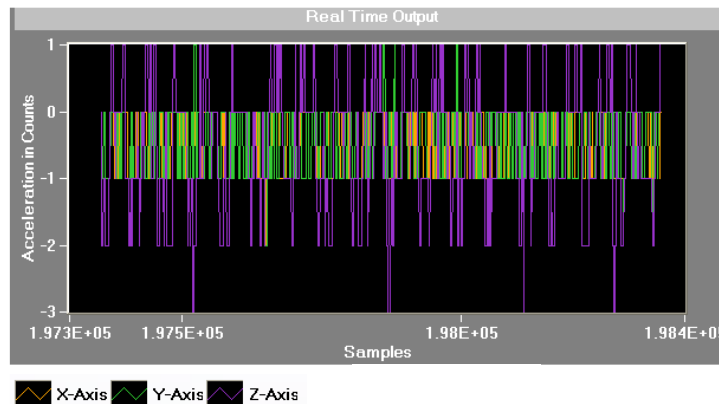


Figure 5. 4 Hz High Pass Filtered Transient Data with approximately 1 Hz Motion

Figure 6 shows a controlled Z-axis motion of 0.8 - 1.2 Hz frequency with 50 Hz ODR with the high pass filter set at 0.063 Hz. Now the motion of the Z-axis matches very closely with the true motion since the high pass filter cut off is extremely low. The static acceleration data is filtered out centering the delta data at 0 counts. Note the Z-axis motion matches very closely to the XYZ unfiltered output shown in Figure 7.

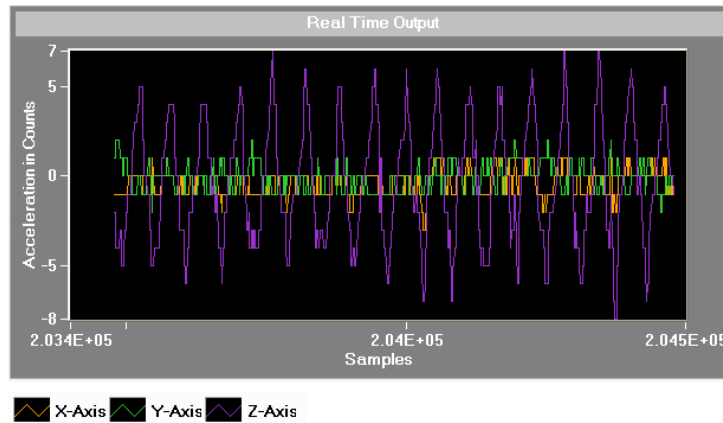


Figure 6. 0.063 Hz High Pass Filtered Transient Data with Approximately 1 Hz Motion

Figure 7 shows 1 Hz of motion, 50 Hz ODR with no high pass filter, with fairly controlled Z-axis only motion. The part is held in the X = 0g, Y = g, Z = -1g configuration since the Z-axis is bouncing with the center around -16 counts (16 counts/g), while X and Y are close to 0 counts.

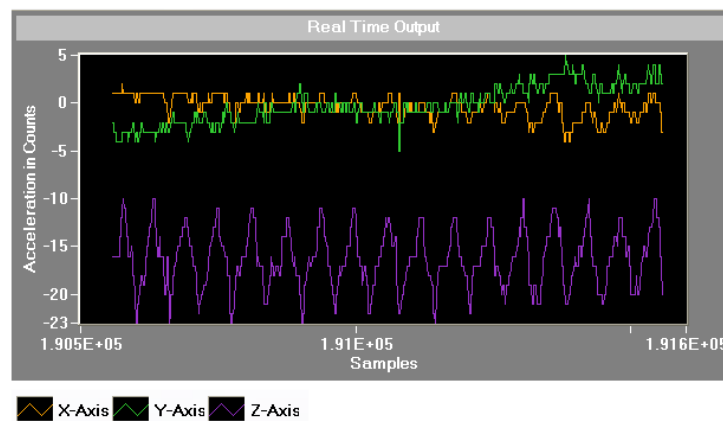


Figure 7. XYZ Output Data with no High Pass Filter Showing Approximately 1 Hz Motion

## 5.0 Using the Delta Data to Write Algorithms such as Directional Shake

Refer to [Figure 8](#), “Directional Shake Showing Position, Velocity, Acceleration”. The delta data can be used to write algorithms such as for directional shake. This application could be used in a motion based game on a cell phone for detecting the direction of the motion. Directional shake is an application that requires a change in acceleration with the direction of the acceleration change indicated by the initial change sign, followed by a “rebound”, which is a change in acceleration in the opposite direction. This occurs because the proof mass moves in the opposite direction of the package. Therefore as the hand accelerates in the positive direction, the proof mass accelerates in the negative direction. Next, as the hand decelerates and stops (Velocity = 0), the proof mass accelerates in the opposite (positive) direction. By analyzing the output acceleration from the directional shake signature, one can use the delta data to write a very simple algorithm.

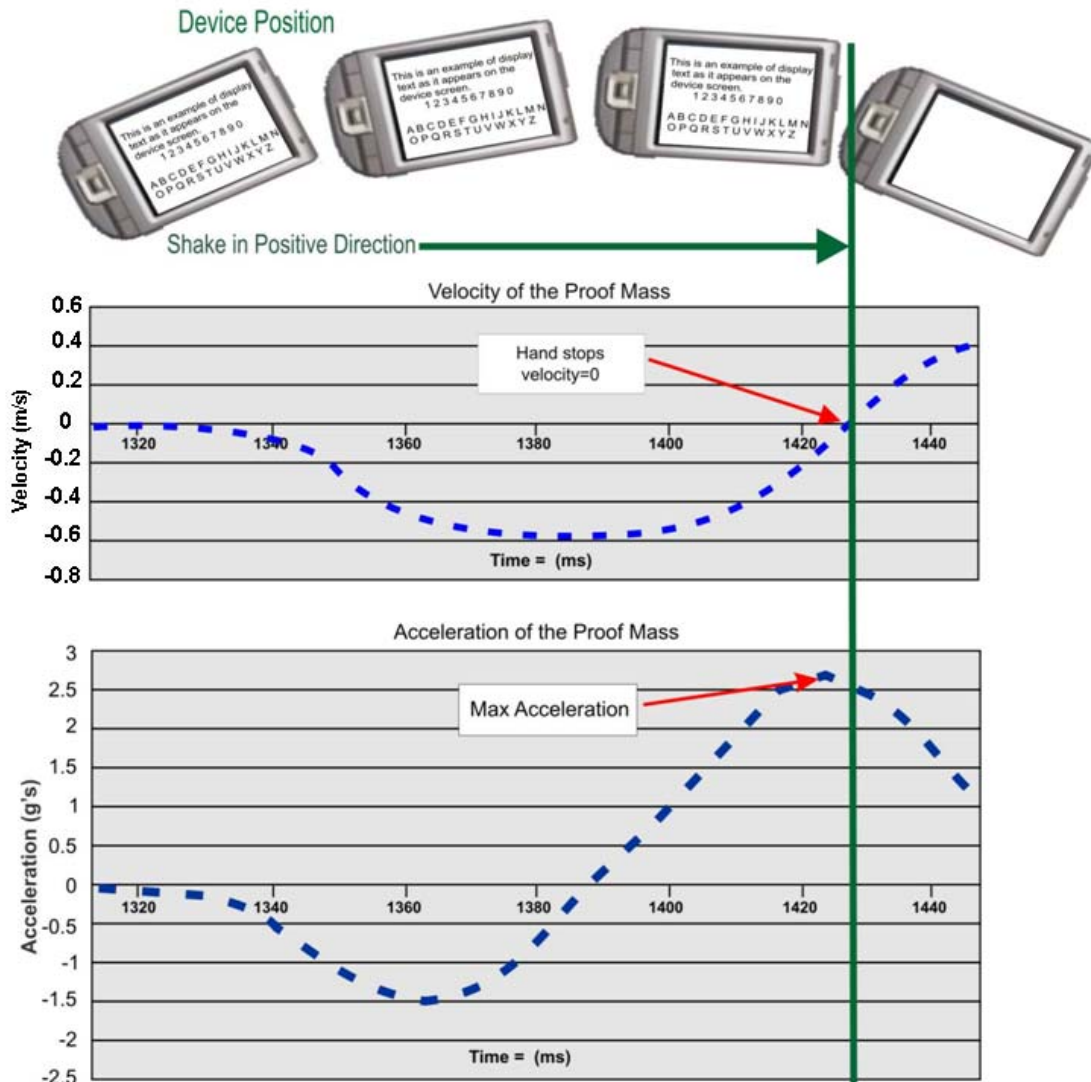


Figure 8. Directional Shake Showing Position, Velocity, Acceleration

Figure 9 shows the Z-axis while shaking in the downward direction. The device was shaken 8 times, consecutively, in the direction of gravity. Note that the deceleration was followed by a larger positive acceleration each time. This is very repeatable. Repeatability of a motion with distinct characteristics is another important factor that can be leveraged while writing an algorithm to detect a condition such as directional shake. This algorithm would look for a change in acceleration above a threshold. Next, it would check the sign of the acceleration, followed by another change in acceleration above a threshold of opposite sign. The algorithm would have to choose appropriate thresholds based on sensor placement and based on how the user is expected to flick the device. Sometimes these algorithms require a bit of usability testing under various conditions with different users to analyze the range of thresholds, and the timing required.

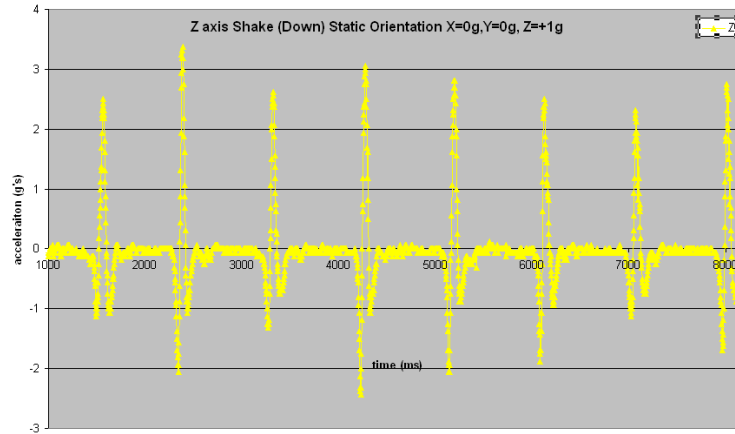


Figure 9. Z-Axis Directional Shake with Downward Motion

The graph in Figure 10 shows the accelerometer shaken upward producing the opposite results of the graph in Figure 9. Acceleration initially increases (positive) and then rebounds with a larger negative acceleration each time. Directional shake has a very distinct signature which is very repeatable. This is an ideal application using the transient data that is obtained from the accelerometer.

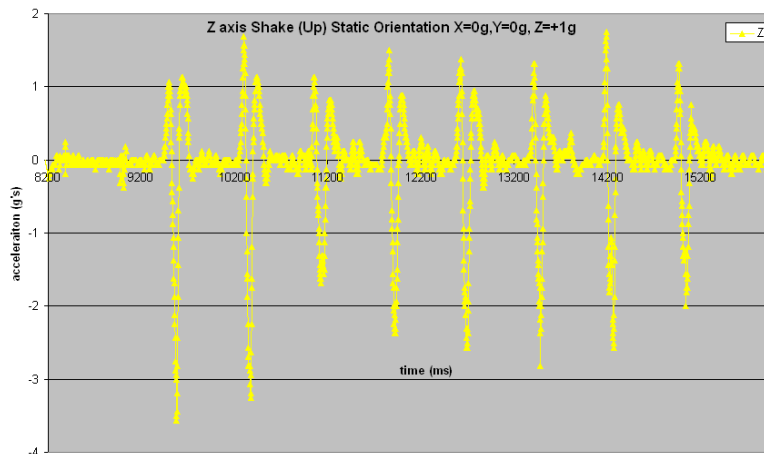


Figure 10. Z-Axis Directional Shake with Upward Motion

As seen in Figure 11, the same applies in the X and Y direction. Below are the graphed results of a shake on the Y-axis in the positive direction. In this case the Y-axis is at zero g's to start and then is accelerated toward the positive direction. This produces similar results to the Z-axis, with a negative initial acceleration followed by positive rebound acceleration.

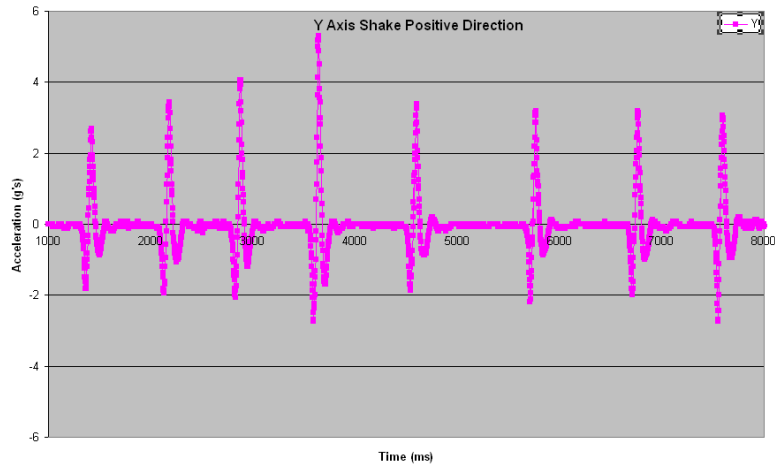


Figure 11. Y-Axis Positive Directional Shake Acceleration Output

The high pass filtered XYZ data is available by reading out registers 0x0B-0x0E. Register 0x0B is the status register and Registers 0x0C, 0x0D and 0x0E are the X, Y and Z Output data registers.

## 6.0 Configuring the Embedded Transient Detection

There are four (4) registers associated with the embedded transient detection function. These are registers 0x2B to 0x2E.

1. Register 0x2B **TRANSIENT\_CFG** - Transient Configuration
2. Register 0x2D **TRANSIENT\_THS** - Setting the Threshold
3. Register 0x2E **TRANSIENT\_COUNT** - Setting the Debounce Counter
4. Register 0x2C **TRANSIENT\_SRC** - Transient Source Detection

This embedded function is used to detect a change in acceleration greater than a set threshold for a determined period of time. The function can be routed to one of the two interrupt pins on the accelerometer.

### 6.1 Register 0x2B:TRANSIENT\_CFG Transient Configuration Register

The transient detection mechanism can be configured to raise an interrupt when the magnitude of the high pass filtered acceleration is greater than the set threshold. The **TRANSIENT\_CFG** register is used to enable the transient interrupt generation mechanism for the 3 axes (X, Y, Z) of acceleration. The transient configuration register and descriptions are shown below in Table 3.

Table 3. Register 0x2B TRANSIENT\_CFG Register (Read/Write) and Description

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
—	—	—	—	ELE	ZTEFE	YTEFE	XTEFE

Note that if **ELE** = 1, the latch is enabled and the “EA”, bit 3 in register 0x2C will remain asserted if an event occurs until the status register (0x2C) is read. The other flags in the status register (0x2C) do not latch. Therefore the register must be read immediately upon the assertion of the interrupt to determine the axis asserted.

**Example: X or Y > 6g with Latch for EA bit enabled for time period of 60 ms at 50 Hz ODR**

**Code Example: IIC\_RegWrite(0x2B, 0x0B); //This enables the X-axis and the Z-axis only with the Latch**

## 6.2 Register 0x2D: TRANSIENT\_THS Transient Threshold Register

The TRANSIENT\_THS register shown in Table 4 sets the threshold limit for the detection of the transient acceleration. The value in the TRANSIENT\_THS register corresponds to a g value which is compared against the values of OUT\_X\_DELTA, OUT\_Y\_DELTA, and OUT\_Z\_DELTA. If the high pass filtered acceleration exceeds the threshold limit, an event flag is raised and an interrupt is generated assuming the interrupts have been configured and enabled.

**Table 4. Register 0x2D TRANSIENT\_THS Register (Read/Write) and Description**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBCNTM	THS6	THS5	THS4	THS3	THS2	THS1	THS0

The threshold THS[6:0] is a 7-bit unsigned number. The minimum threshold resolution is dependent on the selected acceleration g range and the threshold register has a range of 0 to 127 counts.

Therefore:

- If the selected acceleration g range is 8g mode (FS = 11), the minimum threshold resolution is 0.063g/LSB. The maximum threshold is 8g.
- If the selected acceleration g range is 4g mode (FS = 10), the minimum threshold resolution is 0.0315g/LSB. The maximum threshold is 4g.
- If the selected acceleration g range is 2g mode (FS = 01), the minimum threshold resolution is 0.01575g/LSB. The maximum threshold is 2g.

**Example: X or Y > 6g with Latch for EA bit enabled for time period of 60 ms at 50 Hz ODR**

For 6g the device must be in 8g mode to accommodate the threshold. Therefore the minimum time step is 0.063g per LSB.

$6g / 0.063g = 95.2$  counts, round to 96 counts

**Code Example: IIC\_RegWrite(0x2D, 0x60); // Threshold set to 96 counts (6.048g)**

## 6.3 Register 0x2E: TRANSIENT\_COUNT Transient Debounce Register

The TRANSIENT\_COUNT shown in Table 5 sets the number of debounce counts (timing) required for the condition to persist.

**Table 5. Register 0x2E TRANSIENT\_COUNT Register (Read/Write) and Description**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
D7	D6	D5	D4	D3	D2	D1	D0

The time step for the debounce counter is set by the value of the system ODR as shown in Table 6.

**Table 6. TRANSIENT\_COUNT Relationship with the ODR**

Output Data Rate (Hz)	Step	Duration Range
400	2.5 ms	2.5 ms – 0.637s
200	5 ms	5 ms – 1.275s
100	10 ms	10 ms – 2.55s
50	20 ms	20 ms – 5.1s
12.5	80 ms	80 ms – 20.4s
1.56	640 ms	640 ms – 163s

**Example: X or Y > 6g with Latch for EA bit enabled for time period of 60 ms at 50 Hz ODR**

60 ms total time/ 20 ms (at 50 Hz) = 3 counts

**Code Example: IIC\_RegWrite(0x2E, 0x03);**

## 6.4 Register 0x2C: TRANSIENT\_SRC Transient Source Register

The transient event source flag register shown in Table 7 contains bit fields which can be used to determine which event has occurred when a bit is set. Reading this register will clear the “EA” bit if the latch has been enabled (ELE bit) in register 0x2B. This register should be read immediately after the event is detected to determine if the event occurred on the X, Y or Z-axis. If only the X and Z axes are enabled then the event can never show up on the Y-axis.

**Table 7. Register 0x2C TRANSIENT\_SRC Register (Read Only) and Description**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	-	EA	ZTRANSE	YTRANSE	XTRANSE

## 6.5 Configuring the Transient Detection to an Interrupt Pin

In order to set up the system to route to a hardware interrupt pin, the System Interrupt (bit 5 in Reg 0x3B) must be enabled in [Table 8](#). The MMA8450Q allows for eight (8) separate types of interrupts. One (1) of these is reserved for transient detection. To configure the Transient function, the following two step procedure should be followed.

**Step 1:** Enable the Interrupt in Register 0x3B shown in [Table 8](#).

**Table 8. Register 0x3B CTRL\_REG4 Register (Read/Write)**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INT_EN_ASLP	INT_EN_FIFO	INT_EN_TRANS	INT_EN_LNDPRT	INT_EN_PULSE	INT_EN_FF_MT_1	INT_EN_FF_MT_2	INT_EN_DRDY

The corresponding interrupt enable bit allows the function to route its event detection flag to the interrupt controller of the system. The interrupt controller routes the enabled function to the INT1 or INT2 pin.

To enable the Transient function, set bit 5 in register 0x3B as follows:

**Code Example:** `IIC_RegWrite(0x3B, 0x20);`

**Step 2:** Route the interrupt to INT1 or to INT2 shown in [Table 9](#).

**Table 9. 0x3C CTRL\_REG5 Register (Read/Write) and Description**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INT_CFG_ASLP	INT_CFG_FIFO	INT_CFG_TRANS	INT_CFG_LNDPRT	INT_CFG_PULSE	INT_CFG_FF_MT_1	INT_CFG_FF_MT_2	INT_CFG_DRDY

To set the Transient function to INT 1, set bit 5 in register 0x3C.

**Code Example:** `IIC_RegWrite(0x3C,0x20);`

## 6.6 Reading the System Interrupt Status Source Register

In the interrupt source register shown in [Table 10](#), the status of the various embedded functions can be determined.

- The bits that are set (logic '1') indicate which function has asserted an interrupt.
- Conversely, the bits that are cleared (logic '0') indicate which function has not asserted or has de-asserted an interrupt.

The interrupts are rising edge sensitive. The bits are set by a low to high transition and are cleared by reading the appropriate interrupt source register.

**Table 10. 0x15 INT\_SOURCE: System Interrupt Status Register (Read Only)**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SRC_ASLP	SRC_FIFO	SRC_TRANS	SRC_LNDPRT	SRC_PULSE	SRC_FF_MT_1	SRC_FF_MT_2	SRC_DRDY

## 7.0 Details for Configuring the MMA8450Q for Transient Detection

The registers of importance for configuring the MMA8450Q for transient detection are listed below in [Table 11](#).

**Table 11. Registers of Importance for Setting up the Transient Detection and Using the HPF**

Reg	Name	Definition	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0B	STATUS	Data Status R	ZYXOW	ZOW	YOW	XOW	ZYXDR	ZDR	YDR	XDR
0C	OUT_X_DELTA	8 bit Transient X Data R	XD7	XD6	XD5	XD4	XD3	XD2	XD1	XD0
0D	OUT_Y_DELTA	8 bit Transient Y Data R	YD7	YD6	YD5	YD4	YD3	YD2	YD1	YD0
0E	OUT_Z_DELTA	8 bit Transient Z Data R	ZD7	ZD6	ZD5	ZD4	ZD3	ZD2	ZD1	ZD0
15	INT_SOURCE	Interrupt Status	SRC_ASLP	SRC_FIFO	SRC_TRANS	SRC_LNDPRT	SRC_PULSE	SRC_FF_MT_1	SRC_FF_MT_2	SRC_DRDY
17	HP_FILTER_CUTOFF	HP Filter Setting R/W	0	0	0	0	0	0	SEL1	SEL0
2B	TRANSIENT_CFG	Transient Config R/W	-	-	-	-	ELE	ZTEFE	YTEFE	XTEFE
2C	TRANSIENT_SRC	Transient Source R	-	-	-	-	EA	ZTRANSE	YTRANSE	XTRANSE
2D	TRANSIENT_THS	Transient Threshold R/W	DBCNTM	THS6	THS5	THS4	THS3	THS2	THS1	THS0
2E	TRANSIENT_COUNT	Transient Debounce R/W	D7	D6	D5	D4	D3	D2	D1	D0
3B	CTRL_REG4	Control Reg4 R/W (Interrupt Enable Map)	INT_EN_ASLP	INT_EN_FIFO	INT_EN_TRANS	INT_EN_LNDPRT	INT_EN_PULSE	INT_EN_FF_MT_1	INT_EN_FF_MT_2	INT_EN_DRDY

### 7.1 Example Steps for Configuring Transient Detection

**Change in X or Y > 0.5g for 50 ms at 100 Hz ODR, in 2g mode**

**Step 1:** Put the device in Standby Mode: Register 0x38 CtrlReg1

**IIC\_RegWrite(0x38, 0x08); //Set device in 100 Hz ODR, Standby**

**Step 2:** Enable X and Y Axes and enable the latch: Register 0x2B Configuration Register

**IIC\_RegWrite (0x2B, 0x0B); Enable Latch, Enable X and Enable Y**

**Step 3:** Set the Threshold: Register 0x2D

**Note: 2g mode therefore resolution is 15.75mg per count.**

•  $0.5g / 0.01575g = 31.75$ . Therefore set the threshold to 32 counts

**IIC\_RegWrite(0x2D, 0x20);**

**Step 4:** Set the Debounce Counter for 50 ms: Register 0x2E

**Note: 100 Hz ODR, therefore 10 ms step sizes**

**IIC\_RegWrite(0x2E, 0x05);**

**Step 5:** Enable Transient Detection Interrupt in the System (Ctrl Reg4)

**IIC\_RegWrite(0x3B, 0x20);**

**Step 6:** Route the Transient Interrupt to INT 1 hardware pin (Ctrl Reg5)

**IIC\_RegWrite(0x3C, 0x20);**

**Step 7:** Put the device in 2g Active Mode: Register 0x38 Ctrl Reg1

**IIC\_RegWrite(0x38, 0x09); //ODR at 100 Hz, 2g Mode**

**Step 8:** Write Interrupt Service Routine Reading the System Interrupt Status and the Transient Status

```
Interrupt void isr_KBI (void)
{
    //clear the interrupt flag
    CLEAR_KBI_INTERRUPT;
    //Determine source of interrupt by reading the system interrupt register
    Int_SourceSystem = IIC_RegRead(0x15);
    // Set up Case statement here to service all of the possible interrupts
    if ((Int_SourceSystem&0x20)==0x20)
    {
        //Perform an Action since Transient Flag has been set
        //Read the Transient to clear system interrupt and Transient
        Int_SourceTrans = IIC_RegRead(0x2C);
        //Can parse out the data to perform a specific action based on the
        //axes that made the condition true
    }
}
```

## **How to Reach Us:**

### **Home Page:**

[www.freescale.com](http://www.freescale.com)

### **Web Support:**

<http://www.freescale.com/support>

### **USA/Europe or Locations Not Listed:**

Freescale Semiconductor, Inc.  
Technical Information Center, EL516  
2100 East Elliot Road  
Tempe, Arizona 85284  
1-800-521-6274 or +1-480-768-2130  
[www.freescale.com/support](http://www.freescale.com/support)

### **Europe, Middle East, and Africa:**

Freescale Halbleiter Deutschland GmbH  
Technical Information Center  
Schatzbogen 7  
81829 Muenchen, Germany  
+44 1296 380 456 (English)  
+46 8 52200080 (English)  
+49 89 92103 559 (German)  
+33 1 69 35 48 48 (French)  
[www.freescale.com/support](http://www.freescale.com/support)

### **Japan:**

Freescale Semiconductor Japan Ltd.  
Headquarters  
ARCO Tower 15F  
1-8-1, Shimo-Meguro, Meguro-ku,  
Tokyo 153-0064  
Japan  
0120 191014 or +81 3 5437 9125  
[support.japan@freescale.com](mailto:support.japan@freescale.com)

### **Asia/Pacific:**

Freescale Semiconductor China Ltd.  
Exchange Building 23F  
No. 118 Jianguo Road  
Chaoyang District  
Beijing 100022  
China  
+86 10 5879 8000  
[support.asia@freescale.com](mailto:support.asia@freescale.com)

### **For Literature Requests Only:**

1-800-441-2447 or +1-303-675-2140  
Fax: +1-303-675-2150  
[LDCForFreescaleSemiconductor@hibbertgroup.com](mailto:LDCForFreescaleSemiconductor@hibbertgroup.com)

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2010. All rights reserved.

