

# Using the Xtrinsic FXLS8471Q Accelerometer Vector-Magnitude Function

by: Talat Ozyagcilar  
Applications Engineer

## 1 Introduction

This application note demonstrates the FXLS8471Q vector-magnitude function with use-case examples. Use cases where this application note is useful include:

- Detection of linear acceleration exceeding a threshold
- Detection of change of tilt angle exceeding a threshold.

### 1.1 Keywords

Vector Magnitude, Motion Detection, Tilt Detection, Linear-Acceleration Detection, Tilt-Angle Detection, Inertial, Accelerometer.

## Contents

1	Introduction . . . . .	1
1.1	Keywords . . . . .	1
1.2	Related Documentation . . . . .	2
2	Accelerometer Vector-Magnitude Function . . . . .	2
3	Use Cases for Vector-Magnitude Function . . . . .	4
3.1	Use Case 1: Detection of linear acceleration exceeding a threshold . . . . .	4
3.2	Use Case 2: Detection of change of tilt angle exceeding a threshold . . . . .	4

## 1.2 Related Documentation

The FXLS8471Q device features and operations are described in a variety of reference manuals, user guides, and application notes. To find the most-current versions of these documents:

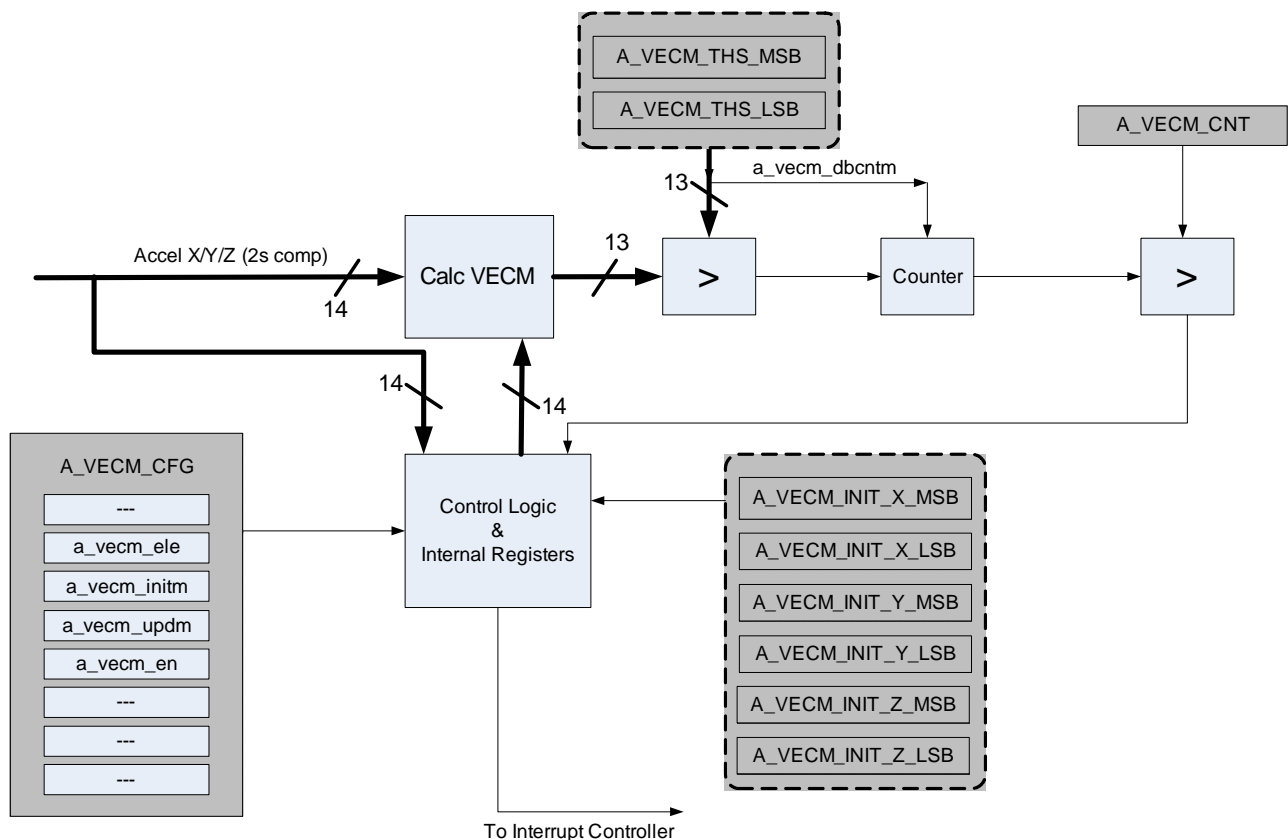
1. Go to the Freescale homepage at:
 

<http://www.freescale.com/>
2. In the Keyword search box at the top of the page, enter the device number FXLS8471Q.
3. In the Refine Your Result pane on the left, click on the Documentation link.

## 2 Accelerometer Vector-Magnitude Function

The accelerometer vector-magnitude function raises an event flag<sup>(1)</sup> when the magnitude of the current acceleration vector measurement minus a programmable reference acceleration vector reading exceeds a programmable threshold for a programmable period of time:

$$\sqrt{(a_{x\_out} - a_{x\_ref})^2 + (a_{y\_out} - a_{y\_ref})^2 + (a_{z\_out} - a_{z\_ref})^2} > A\_VECM\_THS \quad \text{Eqn. 1}$$



**Figure 1. Accelerometer vector-magnitude function block diagram**

1. An event flag sets src\_a\_vecm bit in the INT\_SOURCE register, which can be mapped to either the INT1 and INT2 pins, with programmable logic polarity.

Equation 1 is a test of the form:

$$|\mathbf{p} - \mathbf{q}| = \sqrt{(\mathbf{p} - \mathbf{q}) \cdot (\mathbf{p} - \mathbf{q})} > r \quad \text{Eqn. 2}$$

where  $\mathbf{p}$  is the accelerometer reading,  $\mathbf{q}$  is the programmable offset and  $r$  is the event threshold.

Equation 2 can be rewritten as:

$$\sqrt{\mathbf{p} \cdot \mathbf{p} + \mathbf{q} \cdot \mathbf{q} - 2\mathbf{p} \cdot \mathbf{q}} = \sqrt{p^2 + q^2 - 2pq \cos \alpha} > r \quad \text{Eqn. 3}$$

where  $p$  and  $q$  are the magnitudes of the vectors  $\mathbf{p}$  and  $\mathbf{q}$ , and  $\alpha$  is the angle between the vectors  $\mathbf{p}$  and  $\mathbf{q}$ .

For the special case where  $\mathbf{p}$  and  $\mathbf{q}$  have the same magnitude,  $|\mathbf{p}| = |\mathbf{q}|$ , Equation 3 can be written as:

$$\sqrt{2p^2 - 2p^2 \cos \alpha} = \sqrt{4p^2 \sin^2\left(\frac{\alpha}{2}\right)} = \left|2p \sin\left(\frac{\alpha}{2}\right)\right| > r \quad \text{Eqn. 4}$$

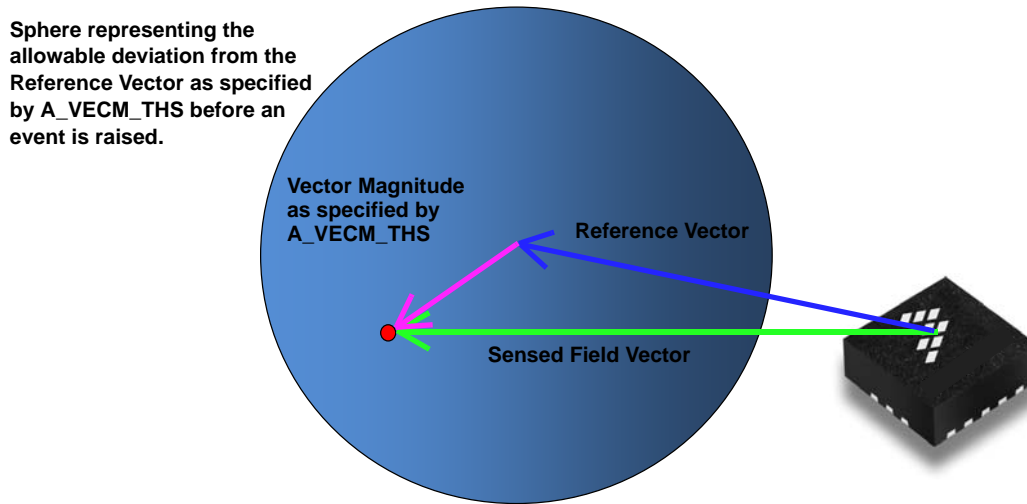


Figure 2. Operating principle of the vector-magnitude feature

The operating principle of this function is depicted in Figure 2. The vector-magnitude function simply compares the magnitude of the difference between the sensed acceleration vector and the reference acceleration vector against the magnitude specified by `A_VECM_THS`. If the former is greater than the latter for a minimum duration of time specified by `A_VECM_CNT`, then an event flag is raised.

### 3 Use Cases for Vector-Magnitude Function

The accelerometer vector-magnitude function use cases below are discussed:

- Detection of linear acceleration exceeding a threshold
- Detection of change of tilt angle exceeding a threshold.

#### 3.1 Use Case 1: Detection of linear acceleration exceeding a threshold

An accelerometer sensor measures the difference between linear acceleration and the local gravitational field as measured in its frame of reference. Since the accelerometer may be mounted at an arbitrary orientation in the device or object being monitored, it is convenient to introduce a rotation matrix  $\mathbf{R}_0$  defining the accelerometer mounting orientation relative to the fixed earth frame of reference. The accelerometer reading  $\mathbf{G}$  is then:

$$\mathbf{G} = \mathbf{R}_0(\mathbf{a} - \mathbf{g}) \quad \text{Eqn. 5}$$

where  $\mathbf{g}$  is the earth's gravitational field vector, which has constant magnitude and constant downwards direction in the earth's reference frame, and  $\mathbf{a}$  is the device or object's acceleration in the earth's reference frame.

When the device or object is stationary, there can be no linear acceleration and the accelerometer reading  $\mathbf{G}_{ref}$  will be:

$$\mathbf{G}_{ref} = -\mathbf{R}_0\mathbf{g} \quad \text{Eqn. 6}$$

If the reading  $\mathbf{G}_{ref}$  is recorded and stored as the reference accelerometer reading, the vector magnitude threshold test is:

$$|\mathbf{G} - \mathbf{G}_{ref}| = |\mathbf{R}_0(\mathbf{a} - \mathbf{g}) + \mathbf{R}_0\mathbf{g}| = |\mathbf{R}_0\mathbf{a}| = |\mathbf{a}| > r \quad \text{Eqn. 7}$$

The mounting orientation rotation matrix  $\mathbf{R}_0$  cancels from [Equation 7](#) since the magnitude of the acceleration vector  $\mathbf{a}$  is independent of any rotation of the sensor. The threshold test is therefore completely independent of mounting orientation and is triggered whenever the acceleration magnitude exceeds the user-defined threshold  $r$  which is stored in the `A_VECM_THS` register.

#### 3.2 Use Case 2: Detection of change of tilt angle exceeding a threshold

Accelerometers are widely used for determining the orientation angles of a device held in the earth's gravitational field. If there is negligible linear acceleration, the accelerometer reading  $\mathbf{G}_{ref}$  with the device in its default orientation, represented by rotation matrix  $\mathbf{R}_0$ , will again be:

$$\mathbf{G}_{ref} = -\mathbf{R}_0\mathbf{g} \quad \text{Eqn. 8}$$

If the device orientation changes, the accelerometer reading due to gravity alone will be:

$$\mathbf{G} = -\mathbf{R}\mathbf{g} \tag{Eqn. 9}$$

where  $\mathbf{R}$  is the rotation matrix representing the new orientation.

If  $\mathbf{G}_{ref}$  is stored as the reference accelerometer reading for the modulus vector difference calculation, the threshold test becomes:

$$|\mathbf{G} - \mathbf{G}_{ref}| = |-\mathbf{R}\mathbf{g} + \mathbf{R}_0\mathbf{g}| > r \tag{Eqn. 10}$$

Substituting Equation 4, which is applicable because the rotations have no effect on the magnitude of the measured earth's gravitational field, gives:

$$|\mathbf{G} - \mathbf{G}_{ref}| = \left| 2g \sin\left(\frac{\alpha}{2}\right) \right| < r \tag{Eqn. 11}$$

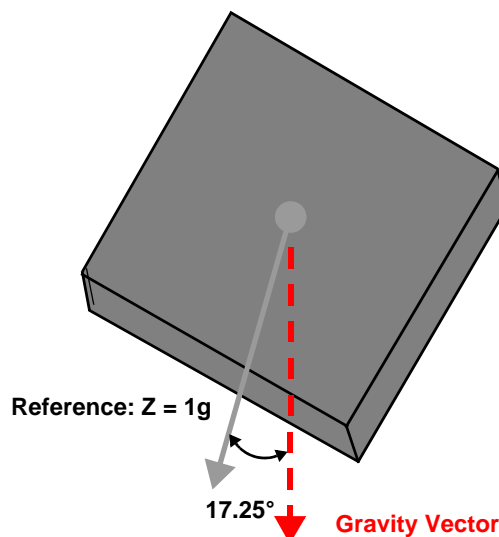
where  $\alpha$  is the change in tilt angle between the reference measurement  $\mathbf{G}_{ref}$  and the current measurement  $\mathbf{G}$ .

For a given threshold  $r$ , the event will trigger for a change in tilt angle  $\alpha$  given by:

$$|\alpha| > 2 \sin^{-1}\left(\frac{r}{2g}\right) \tag{Eqn. 12}$$

If  $r$  is set to 0.3g, as shown in the code Example 1 below, the event will trigger for a change in tilt angle  $\alpha$  exceeding 17.25°.

In Example 1, the Python code snippet sets up the device for detection of a change in tilt angle exceeding 17.25° from the horizontal plane. The device is set up so that once an event is triggered, an interrupt will be generated on the INT1 pin. The user application will read register INT\_SOURCE (0x0C) in an interrupt service routine, thereby clearing the interrupt. This example uses polling for checking the state of the INT1 pin by reading the INT\_SOURCE interrupt status register.



**Figure 3. Calculations reveal that tilting the PCB about ±17° with respect to the horizontal plane will trigger a vector-magnitude event with  $r = 0.3\text{ g}$**

### Example 1. Python code snippet for accelerometer vector-magnitude function

---

```

# 0.3G
# A_VECM_THS_MSB, LSB = 0x84CC => 1228 *0.224mg = ~300mg
# 0x8000 designates the debounce behavior

write_byte( [0x60, 0x84] ) # write 0x84 to reg 0x60
write_byte( [0x61, 0xCC] )

# A_VECM_CNT = 1 * 80ms = 80ms

write_byte( [0x62, 0x01] )

# A_VECM_INITX/Y/Z
# Set Z only as 1g

write_byte( [0x63, 0x00] )
write_byte( [0x64, 0x00] )

write_byte( [0x65, 0x00] )
write_byte( [0x66, 0x00] )

write_byte( [0x67, 0x10] )
write_byte( [0x68, 0x00] )

# A_VECM_CFG
# a_vecm_ele = 1 => event latching enabled
# a_vecm_initm = 1 => use A_VECM_INITX/Y/Z as initial reference
# a_vecm_updm = 1 => do not update initial reference
# a_vecm_en = 1 => enable acceleration vector magnitude detection feature

write_byte( [0x5F, 0x78] )

# enable interrupts for the feature using CTRL_REG4

write_byte( [0x2D, 0x02] )

# route interrupts to INT1 pin using CTRL_REG5

write_byte( [0x2E, 0x02] )

# Enable device with ODR = 12.5Hz using CTRL_REG1,
# OSR=NORMAL (def)

write_byte( [0x2A, 0x29] )

# Wait for INT1 to assert and clear interrupt by reading register INT_SOURCE (0x0C)

while( True ):

    transition = aa_gpio_change( handle, 100 )

    if (transition & INT1_PIN ) == INT1_PIN:

```

```
print "No interrupt..."
continue

print "Interrupt occurred!!!"
(count, dataIn) = read_dev( 0x0C, 1 )
print "0x0C =0x%X" % dataIn[0]
```

---

**How to Reach Us:**

**Home Page:**  
www.freescale.com

**Web Support:**  
<http://www.freescale.com/support>

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

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale 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 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 does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: <http://www.reg.net/v2/webservices/Freescale/Docs/TermsandConditions.htm>.

Freescale and the Freescale logo are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Xtrinsic is a trademark of Freescale Semiconductor, Inc.

All other product or service names are the property of their respective owners.

© 2013 Freescale Semiconductor, Inc. All rights reserved.