June 28, 2007

Solutions Based in Accelerometers
AC317

Oscar Camacho
Systems and Application Engineer
Presentation

Vibration

Positioning

Shock

Fall

Movement

Tilt
Objective

✓ Understand **basic functions** that can be easily integrated into an application that involves acceleration sensors.

✓ Understand the **basic operation** of Freescale Acceleration Sensors.

✓ Basic functions include **fall, motion, positioning, shock, tilt** and **vibration** detection.

![Image of MMA6271 and MMA6281 Acceleration Sensors]
Introduction

- Accelerometers Present & Future
- Analog Output Accelerometers
- Digital Output Accelerometers
- Types of Basic Applications
- Theory & Algorithms for:
  - Tilt
  - Movement & Shock
  - Fall
  - Positioning
  - Vibration
Introduction

Accelerometers Present & Future
Analog Output Accelerometers
Digital Output Accelerometers
Types of Basic Applications

Theory & Algorithms for:
- Tilt
- Movement & Shock
- Fall
- Positioning
- Vibration

Agenda
Introduction to Accelerometers Applications

Automotive applications
- Roll over
- Airbags control
- Motion sensing
- Navigation
- GPS w/ E-Compass

Industrial applications
- Motor stability
- Seismometers

Consumer
- Camera stabilization
- Text Scroll
- Motion Dialing
- Tilt and Motion Sensing
- Pedometers
- Hard disk protection
- 3D gaming
- Freefall Detection
- Image Stability
- Motion Sensing

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. 2006-2008.
Accelerometer Six Sensing Functions

Six Functions!!!

- Movement
- Vibration
- Fall
- Tilt
- Positioning
- Shock
Gravity Measurements

1 g = 9.8 m/s²

Tilt/Inclinometer: 0-1 g
PDA, Cell phone

Vibration: 8-10 g
Motor stability

Crash Detection:
Front: 20-250 g
Side: 40-250 g

Game Controller: 1-2 g
Virtual Reality, Joysticks

Inertial Navigation: 500 mg-1 g
Avionics, Military, GPS

Pedometer: 20-30 g
Pace, Physiology

Freefall Detection: 1-2g
Mobile HDD, Cell phone

Seismometry: 0.002-2 g
Geophones, Seismic Switches

Roll Over: 2-8 g
Axial, Skew

Bullet: >5000 g

1 g = 9.8 m/s²

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. 2006-2008.
What is Acceleration?

It is a measure of how fast the velocity of an object is changing.

\[
a = \frac{dV}{dt}
\]

**Acceleration** is the change in velocity divided by time.

**Units:**
- meters per second per second [m/s²]
- miles per second per second [miles/s²]
- gravities, multiples of 9.8 m/s² [g]

**Example:** if a car take 10 seconds to accelerate from 0 miles/s to a speed of 85 miles/s, then its acceleration is 8.5 miles/s²
Dynamic Acceleration

When an object is moving or falling, the effect of gravity is called **dynamic acceleration**.

The arrows indicate the direction of the mass movement.
Static Acceleration

When an object is not moving, the effect of gravity is called **static acceleration**

9.8 m/s²

Direction of **earth’s gravity** field.

When positioned as shown, the earth’s gravity will result in a positive 1g output.
The sensor’s position determines the static acceleration in each axis

- $X_{OUT}@0g$
- $Y_{OUT}@-1g$
- $Z_{OUT}@0g$

- $X_{OUT}@-1g$
- $Y_{OUT}@0g$
- $Z_{OUT}@0g$

- $X_{OUT}@0g$
- $Y_{OUT}@0g$
- $Z_{OUT}@-1g$
X-Lateral g-Cell Structure

Direction of Motion

Movable Mass

Direction of Motion

Movable Mass
X-Lateral g-Cell SEM Photo

- Movable Mass
- Doubly Fixed Finger
- Sensing Finger
- Stop
Z Axis G-Cell Principle Structure Overview

Cross Section View
The Z Axis G-Cell

- Moving middle plate
- Fixed top plate
- Z-stops
Introduction

Accelerometers Present & Future

- Analog Output Accelerometers
- Digital Output Accelerometers
- Types of Basic Applications
- Theory & Algorithms for:
  - Tilt
  - Movement & Shock
  - Fall
  - Positioning
  - Vibration
NEW Products!

MMA73x0L: Analog Output

► Features

- 3-axis Analog Output with g-Select
  - MMA7360L (1.6g, 6g)
  - MMA7340L (3g, 12g)
  - MMA7330L (4g, 16g)
- Low current consumption: 400μA
- 3μA in Sleep mode
- Low voltage operation: 2.2 V – 3.6 V
- Linear 0g freefall detect logic output
- Z-axis self test for freefall function check

► Package

- 3 x 5 x 1mm LGA-14

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. 2006-2008.
MMA7450L: Digital Output

► Features
  • 8-bit I2C or SPI Digital Output
  • 2.4 – 3.6V Vdd Operation
  • 1.8V Compatible I/Os
  • 450μA $I_{DD}$, 5μA at Sleep mode
  • Selectable full scale range (2g, 8g)
  • Programmable Threshold Interrupt
  • Programmable Pulse Interrupt

► Package
  • 3 x 5 x 0.8mm LGA-14
Introduction

Accelerometers Present & Future

**Analog Output Accelerometers**

Digital Output Accelerometers

Types of Basic Applications

Theory & Algorithms for:
- Tilt
- Movement & Shock
- Fall
- Positioning
- Vibration

Agenda
Analog Output Accelerometers (MMA7260Q)

- Selectable Sensitivity (1.5g / 2g / 4g / 6g)
- Low Current Consumption: 500 µA
- Sleep Mode: 3 µA
- Low Voltage Operation: 2.2 V to 3.6 V
- 6mm x 6mm x 1.45mm QFN package

<table>
<thead>
<tr>
<th>Sensitivity</th>
<th>g-Range</th>
<th>g-Select2</th>
<th>g-Select1</th>
</tr>
</thead>
<tbody>
<tr>
<td>800mV/g</td>
<td>1.5g</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>600mV/g</td>
<td>2g</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>300mV/g</td>
<td>4g</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>200mV/g</td>
<td>6g</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Typical MMA7260Q Output Response

MMA7260Q typical output response at 1.5 g-range and 3V supply

Typical value @ 0g = Vdd/2
Transfer Function for the MMA7260Q

\[ V_{OUT} = \frac{V_{DD}}{2} + S \cdot a \]

Where:

- **Vout** is the output voltage for any axis [V]
- **Vdd** is the device supply voltage [V]
- **S** (sensitivity) is the rate of change in voltage due to acceleration [V/g]

<table>
<thead>
<tr>
<th>Sensitivity</th>
<th>g-Range</th>
<th>g-Select2</th>
<th>g-Select1</th>
</tr>
</thead>
<tbody>
<tr>
<td>800mV/g</td>
<td>1.5g</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>600mV/g</td>
<td>2g</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>300mV/g</td>
<td>4g</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>200mV/g</td>
<td>6g</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- **a** is the acceleration [g]
Introduction

Accelerometers Present & Future

Analog Output Accelerometers

Digital Output Accelerometers

Types of Basic Applications

Theory & Algorithms for:

- Tilt
- Movement & Shock
- Fall
- Positioning
- Vibration
Digital Output Accelerometers

- Digital output with I2C/SPI
- Selectable Sensitivity ($\pm 2g$, $\pm 4g$, $\pm 8g$)
- Low Current Consumption: 400 $\mu$A; Sleep Mode: 5 $\mu$A
- 3mm x 5mm x 0.8mm LGA-14 Package
- Programmable threshold interrupt output
- Freefall interrupt output
- Low external component count
MMA7450L Block Diagram

[Block Diagram Image]
Basic MMA7450L connection

**I2C Connectivity**

- Two external Components
- I2C SCL and SDA
- 2 Interrupts

**SPI connectivity**

- Two external Components
- SPI SCK, MISO, MOSI, CS
- 2 Interrupts
Introduction
- Accelerometers Present & Future
- Analog Output Accelerometers
- Digital Output Accelerometers

Types of Basic Applications
- Theory & Algorithms for:
  - Tilt
  - Movement & Shock
  - Fall
  - Positioning
  - Vibration
## Accelerometer Applications

<table>
<thead>
<tr>
<th>Tilt</th>
<th>Inclinometer, Gaming, TextScrolling/User Interfacing, Image Rotating, LCD projection, Physical Therapy, Camcorder Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movement</td>
<td>Motion Control, Pedometers, General Movement Detection</td>
</tr>
<tr>
<td>Positioning</td>
<td>Personal navigation, Car navigation, Back-up GPS, Anti-theft Devices, Map Tracking</td>
</tr>
<tr>
<td>Shock</td>
<td>Fall log, Black Boxes/Event Recorders, HDD Protection, Shipping and Handling Monitor</td>
</tr>
<tr>
<td>Vibration</td>
<td>Seismic Activity Monitors, Smart Motor Maintenance, Appliance Balance &amp; Monitoring, Acoustics</td>
</tr>
<tr>
<td>Fall</td>
<td>Free-fall Protection, HDD Protection, Fall Log, Fall Detection, Motion Control &amp; Awareness</td>
</tr>
</tbody>
</table>
Introduction

Accelerometers Present & Future

Analog Output Accelerometers

Digital Output Accelerometers

Types of Basic Applications

Theory & Algorithms for:

- **Tilt**
- Movement & Shock
- Fall
- Positioning
- Vibration
Basics About Tilt

Application
- 3D Gaming
- Text Scrolling
- Digital Camera Stability

Things to consider
- What is the angle of reference?
- How is your accelerometer mounted?
- Inclination range?
- It is based on static acceleration

Output will vary from –1.0g to +1.0g when the angle is tilted from -90° to +90°

Mount accelerometer so axis of sensitivity is parallel to the earth’s surface
Vectors Decomposition

\[ a = r \times \cos(\theta) \]
\[ b = r \times \sin(\theta) \]

\[ \theta = \arcsin \left( \frac{V_{out} - V_{offset}}{\Delta V} \frac{\Delta V}{\Delta G} \right) \]

- \( V_{OUT} \) = Output of Accelerometer
- \( V_{OFF} \) = Zero G Acceleration
- \( \Delta V / \Delta G \) = Sensitivity
- 1.0G = Earth's gravity
- \( \Theta \) = Tilt angle
Output Response Vs Inclination

Output Voltage ($V_{out}$) - Offset Voltage ($V_{off}$)

Angle (degree)

$1g$
Calculate Angle with 8-Bit Lookup Table

Table II. Typical Sensor Outputs using 8-bit ADC (for any axis)

<table>
<thead>
<tr>
<th>ADC</th>
<th>Voltage</th>
<th>g</th>
<th>Angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>66</td>
<td>-0.80</td>
<td>-1.00</td>
<td>-87.47</td>
</tr>
<tr>
<td>77</td>
<td>-0.66</td>
<td>-0.82</td>
<td>-55.26</td>
</tr>
<tr>
<td>88</td>
<td>-0.52</td>
<td>-0.64</td>
<td>-40.13</td>
</tr>
<tr>
<td>99</td>
<td>-0.37</td>
<td>-0.47</td>
<td>-27.86</td>
</tr>
<tr>
<td>110</td>
<td>-0.23</td>
<td>-0.29</td>
<td>-16.86</td>
</tr>
<tr>
<td>121</td>
<td>-0.09</td>
<td>-0.11</td>
<td>-6.48</td>
</tr>
<tr>
<td>132</td>
<td>0.05</td>
<td>0.06</td>
<td>3.70</td>
</tr>
<tr>
<td>143</td>
<td>0.19</td>
<td>0.24</td>
<td>13.99</td>
</tr>
<tr>
<td>154</td>
<td>0.34</td>
<td>0.42</td>
<td>24.77</td>
</tr>
<tr>
<td>165</td>
<td>0.48</td>
<td>0.60</td>
<td>36.60</td>
</tr>
<tr>
<td>176</td>
<td>0.62</td>
<td>0.77</td>
<td>50.66</td>
</tr>
<tr>
<td>187</td>
<td>0.76</td>
<td>0.95</td>
<td>71.93</td>
</tr>
</tbody>
</table>

Supply Voltage at 3.3V and a 8-Bit resolution ADC

Please refer to Application Note AN3107 & AN3461 for more information
Tilt Considerations

Resolution problem close to 90 degrees

<table>
<thead>
<tr>
<th>ADC</th>
<th>8-bit</th>
<th>10-bit</th>
<th>12-bit</th>
<th>16-bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of steps</td>
<td>255</td>
<td>1023</td>
<td>4095</td>
<td>65535</td>
</tr>
<tr>
<td>step value (mV)</td>
<td>12.941</td>
<td>3.226</td>
<td>0.806</td>
<td>0.050</td>
</tr>
<tr>
<td>resolution @ 0°</td>
<td>-0.927</td>
<td>-0.231</td>
<td>-0.058</td>
<td>-0.004</td>
</tr>
<tr>
<td>resolution @ 24°</td>
<td>-1.011</td>
<td>-0.253</td>
<td>-0.063</td>
<td>-0.004</td>
</tr>
<tr>
<td>resolution @ 45°</td>
<td>-1.296</td>
<td>-0.326</td>
<td>-0.082</td>
<td>-0.005</td>
</tr>
<tr>
<td>resolution @ 90°</td>
<td>-10.320</td>
<td>-5.147</td>
<td>-2.572</td>
<td>-0.643</td>
</tr>
</tbody>
</table>

Ouch!

If 2 axes are available:
- use X from 0 to 45 degrees
- use Y from X’s 45 to 90 degrees (Y’s 0 to 45 degrees)
Tilt Flow Diagram

• Description
  - Start the ADC
  - Take a sample from ADC
  - Compare the sample with Table
  - Show angle

• Tilt tables:
  - 8-bit Angle lookup table
  - 10-bit Angle lookup table

Begin

ADC Init

Take Sample From ADC

Angle= Table(Sample)

Show angle
Suggested Tilt Code

```c
5, 11, 15, 19, 21, 24, 26, 28, 30, 31, 33,
35, 36, 38, 40, 41, 42, 44, 45, 46, 47, 49, 50, 51, 52, 53, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67,
68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94,
117, 118, 120, 121, 122, 123, 124, 125, 126, 127, 129, 130, 131, 132, 134, 135, 136, 138, 139, 140, 142, 144,
145, 147, 149, 160, 162, 165, 157, 159, 162, 166, 171.

};

void tilt(void)
{
    delay=0xFF;
    /*ADC_GetAllAxis();*/
    Sample_Z = ADC_GetSingleAxis(Z_AXIS_CHANNEL);
    while(--delay);

    Sensor_Data[1] = angle0bit[Sample_Z <= 67];
    if(Sample_Z <= 67)
    {
        Sensor_Data[1] = 0;
    }
    else if (Sample_Z >= 180)
    {
        Sensor_Data[1] = 180;
    }
    Sensor_Data[0] = 0x01;
    Sensor_Data[2] = 0x41;
    Sensor_Data[3] = 0x11;
    Sensor_Data[4] = 0x11;
    SCITxKsq(Sensor_Data);
}

void main(void)
{
    init();
    do
    {
        tilt();
        /* Wait for Tx Complete */
        while (SCIC2 & 0x06);
    }while(1);

}
```

8-Bit tilt table

Transmit angle

Used for the GUI

Remove table offset
Introduction
Accelerometers Present & Future
Analog Output Accelerometers
Digital Output Accelerometers
Types of Basic Applications
Theory & Algorithms for:
- Tilt
- Movement & Shock
  - Fall
  - Positioning
  - Vibration
Basics about Movement & Shock

Application:
- Pedometers
- General Movement Detection
- HDD Protection
- Shipping and Handling Monitor

Things to consider:
- What is the acceleration range?
- What is the sampling frequency?

The g-Force can range from +/-1g from freefall detection to +/-250g for a car crash.
What is Movement or Shock?

**Shock** is a sudden acceleration or deceleration caused, for example, by impact. Shock is measured in the same unit as acceleration. i.e. meter per squared second (m/s²)

**Movement** is an event that involves a change in position or location of something

The difference between Shock and Movement is: **The magnitude of the force applied to the object**
What is Movement?

![Graph showing Movement over time](image)

- ADC Value (8-Bit)
- Time (mS)

- X
- Y
- Z
What is Shock?
How Do We Measure Movement?

• Previous Signal Sample

• Take the current Sample

• Compare current sample with previous Sample, if difference is greater than predefined threshold then you have a Movement condition
How Do We Measure Shock?

- Previous Signal Sample
- Take the current Sample
- Compare current sample with previous Sample, if difference is greater than predefined threshold then you have a Shock condition
Movement & Shock Flow Diagram

• Description
  ▪ Start the ADC
  ▪ Define Delta
  ▪ Take Sample from Z-Axis
  ▪ Delay
  ▪ Calculate absolute value of difference between sample1 and sample2
  ▪ If difference is greater than delta, enable Buzzer
  ▪ If difference is less than delta, go to take new sample

•Configure for Shock or Movement
  ▪ Set Delta to a higher value for Shock
  ▪ Set Delta to a lower value for Movement
Another method for detecting movement

- **Planetary Model**: check to see if \( X^2 + Y^2 + Z^2 = 1 \)

**No previous history!**

when the accelerometer is not moving
the 3D representation of the X,Y,Z outputs
is a dot on the surface of the planet

Allows for decision based on one reading, without previous history
→ Ideal for low-battery applications, e.g.
  MCU wakes up for 5ms every 1s
#include <stdint.h> /* for EnableInterrupts macro */
#include "derivative.h" /* include peripheral declarations */
#include "adc.h"
#include "buzzer.h"
#include "SC100.h"

#define DELTA 10

unsigned char frequency;
unsigned char Sample_X;
unsigned char Sample_Y;
unsigned char Sample_Z;
unsigned char Sensor_Data[8];
unsigned int delay;

void init(void);
void shock(void);

void shock(void)
{
  static char ADC_PrevConversion;
  delay = 0x01FF;
  while(--delay):
    /*ADC_GetAllAxis():*/
    Sample_Z = ADC_GetSingleAxis(Z_AXIS_CHANNEL);
    frequency = 0x0FF;
    if (((Sample_Z - ADC_PrevConversion) >= DELTA) || ((ADC_PrevConversion - Sample_Z) >= DELTA))
      buzzer();
    ADC_PrevConversion = Sample_Z;
}

void main(void)
{
  init();
  do
    { shock();
  }while(1);
}

Suggestion:
Possible Delta Values for Movement are between 3 and 6
and for Shock are between 7 and 20
Suggested Shock & Movement Code

```c
#include <stdio.h> /* for EnableInterruption macro */
#include "derivative.h" /* include peripheral declarations */
#include "adc.h"
#include "buzzer.h"
#include "SCLtx.h"

#define DELTA 25

unsigned char frequency;
unsigned char Sample_X;
unsigned char Sample_Y;
unsigned char Sample_Z;
unsigned char Sensor_Data[8];
unsigned int delay;

void init(void);
void shock(void);

/***************************************************************
void shock(void)
{
    static char ADC_PrevConversion;
    delay = 0x01FF;
    while(--delay);

    /*ADC_GetAllAxis()*/
    Sample_Z = ADC_GetSingleAxis(Z_AXIS_CHANNEL);
    frequency = 0xFF;

    if (((Sample_Z - ADC_PrevConversion) >= DELTA) ||
        ((ADC_PrevConversion - Sample_Z) >= DELTA))
    {
        buzzer();
    }
    ADC_PrevConversion = Sample_Z;
}
/***************************************************************

void main(void)
{
    init();
    do
    {
        shock();
    }while(1);
}
```

Proposed Delta Value for Shock
Delay Between Samples
Compare Present Sample with Previous Sample
Turn Buzzer On
Introduction
Accelerometers Present & Future
Analog Output Accelerometers
Digital Output Accelerometers
Types of Basic Applications
Theory & Algorithms for:
  ✷ Tilt
  ✷ Movement & Shock
  ✷ **Fall**
  ✷ Positioning
  ✷ Vibration
Basics About Free Fall

Application:
- Portable Media HDD protection
- People Fall detection
- Shipment mishandling protection

Things to consider:
- Linear free fall requires a 3-axis accelerometer
- Rotational and Projectile free fall require a more complex algorithm
When a freefall condition exists, all of the Axis are at Zero-g.
Determining the Height of the Fall

\[ v = at \]

\[ d = \frac{at^2}{2} \]

1 ms = 4.9 \mu m!

10 ms = 0.49 mm!

100 ms = 49 mm

1000 ms = 4.9 m
• Description:
  ▪ ADC Initialization
  ▪ Define Threshold High and Low
  ▪ Take Sample
  ▪ If Sample X, Sample Y, and Sample Z is between Threshold Hi and Low, then
  ▪ Devices is Falling
  ▪ Else, devices is not falling

We need the threshold to ensure that all the falling conditions are detected, since at different altitudes different G’s are detected and the offset of the accelerometer could vary
Suggested Free Fall Code

```c
#include <stdio.h>
#include "derivative.h"
#include "adc.h"
#include "buzzer.h"

#define THRESHOLD_HIGH 150
#define THRESHOLD_LOW 104

unsigned char Sample_X;
unsigned char Sample_Y;
unsigned char Sample_Z;
unsigned char frequency;

void init(void);

/***********************************************************/
void freefall (void)
{
    ADC_GetAllAxes();

    if ((Sample_X <= THRESHOLD_HIGH)&(Sample_X >= THRESHOLD_LOW))
    {
        if ((Sample_Y <= THRESHOLD_HIGH)&(Sample_Y >= THRESHOLD_LOW))
        {
            if ((Sample_Z <= THRESHOLD_HIGH)&(Sample_Z >= THRESHOLD_LOW))
            {
                buzzer();
            }
        }
    }
}

/***********************************************************/
void main(void)
{
    init();
    do
    {
        freefall();
    }while(1);
}
```

Suggested Threshold Values

Compress if Sample is between Threshold Values

Turn Buzzer On
Introduction

Accelerometers Present & Future

Analog Output Accelerometers

Digital Output Accelerometers

Types of Basic Applications

Theory & Algorithms for:
- Tilt
- Movement & Shock
- Fall
- **Positioning**
- Vibration
Basics About Position

Application:
- GPS Compensation
- 3D Gaming
- Map tracking

Things to consider:
- What is the acceleration range?
- How is the accelerometer mounted?
- Integration Algorithms accuracy

Position data is obtained when double integration is performed on the acceleration data.

\[ x(t) = \int \int a(t) \, dt \]
Integrating Acceleration to Determine Velocity and Position

- Remove the offset from the signal

- **Velocity** = Previous Velocity + Current Acceleration

- **Position** = Previous Position + Current Velocity
Performing Accurate Signal Integration

Integration = Area under the curve

$$\text{Area}_n = \text{Sample}_n \times T$$

Please refer to Application Note AN3397 for more information
Calibration Procedure

X and Y Offset
- Hold the board on a flat surface so the accelerometer is facing up

The typical value for the offset is 1.65V when the device is powered from 3.3V

Please refer to Application Note AN3447 for more information
Positioning Flow Diagram

• Steps:
  ▪ Start the ADC
  ▪ Calibrate (Get signal offset)
  ▪ Take sample from ADC
  ▪ Remove offset from sample
  ▪ Velocity = Previous Velocity + Current Acceleration
  ▪ Position = Previous Position + Current Velocity
  ▪ Show Position

*One time calibration
void position(void) // this function transforms acceleration to a proportional movement
{
    unsigned char count2 = 0;
    do {
        ADC_GetAllAxis();
        count2++;
    } while (count2 < 10);
    accelerationx[2] = accelerationx[1] - (int)sqrt(ex);
    if ((accelerationx[1] < 0) && (accelerationx[1] > 0))
    {accelerationx[1] = 0;
    }
    velocityx[1] = velocityx[0] + accelerationx[0] + ((accelerationx[1] - accelerationx[0]) >> 1);
    positionx[1] = positionx[0] + velocityx[0] + ((velocityx[1] - velocityx[0]) >> 1);
    accelerationy[1] = accelerationy[1] - (int)sqrt(ey);
    if ((accelerationy[1] < 0) && (accelerationy[1] > 0))
    {accelerationy[1] = 0;
    }
    velocityy[1] = velocityy[0] + accelerationy[0] + ((accelerationy[1] - accelerationy[0]) >> 1);
    positiony[1] = positiony[0] + velocityy[0] + ((velocityy[1] - velocityy[0]) >> 1);
    accelerationx[0] = accelerationx[1];
    accelerationy[0] = accelerationy[1];
    velocityx[0] = velocityx[1];
    velocityy[0] = velocityy[1];
    data_management_and_transfer();
    movement_end_check();
    positionx[0] = positionx[1];
    positiony[0] = positiony[1];
    direction = 0;
}
Introduction
Accelerometers Present & Future
Analog Output Accelerometers
Digital Output Accelerometers
Types of Basic Applications
Theory & Algorithms for:
  ◆ Tilt
  ◆ Movement & Shock
  ◆ Fall
  ◆ Positioning
  ◆ Vibration
Basics About Vibration

Application:
- Seismic Activity Monitors
- Smart Motor Maintenance
- Acoustics

Things to consider:
- What is the frequency of the vibration?
- Where is the Accelerometer mounted?
- What is the acceleration range?

The time it takes between peaks in a periodic signal determines the fundamental frequency
Vibration Plot Using a 3 Axis Accelerometer
How is Vibration Determined?

For this example we just use the Z-Axis
How is Vibration Determined?

1. Remove the offset from the signal.
2. Set the threshold values.
How is Vibration Determined?

Detect positive to negative transitions

The transition is valid only if it passes the threshold value

A period equals to a negative to positive then again to negative transition

Count the number of periods until 1 second has past

\[ F = nHz \]

Where:
- \( F \) = Frequency
- \( n \) = Number of cycles
**Vibration Flow Diagram**

- **Description:**
  - ADC Initialization
  - Calibrate (Get signal offset)
  - Take Sample
  - Remove Offset
  - If the Sample is greater than positive threshold you are in the positive side of the signal
  - If the Sample is lower than negative threshold you are in the negative side of the signal
  - Each transition is stored into a counter
  - The number of transitions from negative to positive within one second will determine the frequency
#include <stdio.h> /* for EnableInterrupts macro */
#include "derivative.h" /* include peripheral declarations */
#include "adc.h"
#include "buzzer.h"
#include "SCITx.h"

#define THRESHOLD_HIGH 137
#define THRESHOLD_LOW 117

unsigned char frequency;
unsigned char Sample_H;
unsigned char Sample_L;
unsigned char Sample_Z;
unsigned char Sensor_Data[8];

void init(void);

void vibration(void)
{
    static unsigned char ThresholdHighFlag;
    // SRTISC = 0x37;
    /* The RTI runs from internal clock */
    SRTISC = 0x17;
    /* ADC_GetAllAxis(); */
    Sample_Z = ADC_GetSingleAxis(Z_AXIS_CHANNEL);
    if (Sample_Z >= THRESHOLD_HIGH)
    {
        ThresholdHighFlag = 1;
    }
    if ((Sample_Z <= THRESHOLD_LOW) && (ThresholdHighFlag))
    {
        ThresholdHighFlag = 0;
        frequency++;
    }
}

void main(void)
{
    init();
    do
    {
        vibration();
    }while(1);
}
Introduction

Accelerometers Present & Future

Analog Output Accelerometers

Digital Output Accelerometers

Types of Basic Applications

Theory & Algorithms for:

- Tilt
- Movement & Shock
- Fall
- Positioning
- Vibration