The embedded self-test feature in MEMS inertial sensors

Jay Esfandyari, Gang Xu, Marco Capovilla, Paolo Bendiscioli, Marco Bianco - July 22, 2012
Early MEMS sensors did not have built-in self-test features. Engineers were left to their own devices and had to physically tilt (in an accelerometer case) or rotate (in a gyroscope case) the Printed Circuit Board (PCB) with the mounted sensor and measure the sensor’s output to determine if the sensor was working according to its specifications. If the output changed according to the motion applied, this indicated that the sensor was working properly. Otherwise, the sensor was considered “failed”. Obviously, this ponderous procedure was not suitable for mass production.

Recently, the embedded self-test has become an indispensable feature in MEMS sensors. The concept of the self-test is to apply an electrostatic force to the mechanical sensing element of the sensor and simulate the situation as if an external motion or rotation was applied to the device. If the change of sensor output, with or without self-test feature enabled, is within the specified range, this then means that the sensor operates correctly according to its specifications.

Engineers are familiar with MEMS self-test features. However, sometimes it is not clear how to use the self-test correctly. This paper describes the methods of how to implement the self-test feature in digital MEMS accelerometers and gyroscopes with block diagrams and sample codes included.

1. Self-test procedures for accelerometers and gyroscopes

1.1. Accelerometer

When the embedded self-test feature is enabled in an accelerometer, an electrostatic test force is applied to the mechanical sensing element and causes the moving part to move away from its original position, emulating a definite input acceleration. The output in this self-test mode will then be compared with the output data of the device in normal mode. It is important to keep in mind that normal mode is when the self-test is disabled. If the absolute value of the output difference is within the minimum and maximum range of the preselected full scale range, the accelerometer is working properly.

To better understand the accelerometer self-test mode, let’s clarify the following points:

- The raw data needs to be collected with the accelerometer at the same arbitrary stationary position when the self-test is enabled and disabled.
- When switching the accelerometer from power down mode to normal mode, there is a certain amount of time \(^1\) required to allow the accelerometer data to stabilize before data acquisition.
The accelerometer self-test can be performed at different Full Scale (FS) ranges.

During the self-test procedure, 5 or 10 samples can be collected at different modes and then averaged to calculate the final self-test results.

When the accelerometer is in normal mode, “Data_no_ST” (see Figure 1) appears, which means

\[ A_x = x_0; A_y = y_0; A_z = z_0 \]

At the same stationary position, when the accelerometer is in self-test mode, “Data_ST” means

\[ A'_x = x_0 + \Delta_x; A'_y = y_0 + \Delta_y; A'_z = z_0 + \Delta_z \]

Then, the self-test results are

\[ \Delta_x = \text{abs}(A'_x - A_x); \Delta_y = \text{abs}(A'_y - A_y); \Delta_z = \text{abs}(A'_z - A_z) \]

If\{\min_x < \Delta_x < \max_x\) and \(\min_y < \Delta_y < \max_y\) and \(\min_z < \Delta_z < \max_z\)\}

Then, the self-test has been successful. Otherwise, the sensor has failed.

Where, \(\min_{x,y,z}\) and \(\max_{x,y,z}\) are related to the full scale range in the unit of integer LSBs (Least Significant Bit).

There are usually two self-test modes -- self-test0 and self-test1. The difference between self-test0 and self-test1 is that the proof mass moves into opposite directions with the same magnitude. That means that the self-test0 results should be close to the self-test1 results with the opposite sign. Therefore, you only need to pick either self-test0 or self-test1 mode for self-test at any stationary position.

The accelerometer raw data in normal mode (self-test disabled) is equal to half of the sum of the raw data in self-test0 and the raw data in self-test1, because the \(\Delta\) values in self-test0 mode and self-test1 mode get cancelled out. That is,

\[ A_{x,y,z} \text{ (normal mode) } = \frac{[A'_{x,y,z} \text{ (self-test0)} + A'_{x,y,z} \text{ (self-test1)}]}{2} \]
2.2. Gyroscope

Now, when the self-test feature is enabled for a gyroscope, an electrostatic test force is applied to the seismic mass to move away from its original position, simulating a definite Coriolis force. Therefore, the output will change compared to the output values when the gyroscope is in normal mode. If the output change is within the minimum and maximum range that is related to the preselected full scale range, then the gyroscope is functioning properly.

To better understand the gyroscope self-test, let’s clarify these points:

- Gyroscope raw data needs to be collected at the same arbitrary stationary position when the self-test is enabled and disabled.

- When switching the gyroscope from power down mode to normal mode, a certain amount of time \(^{[2]}\) is required for the gyroscope data to be stable before data acquisition.

- The gyroscope can be set to higher Output Data Rate (ODR) to save total self-test time.

- The gyroscope self-test can be performed at different FS ranges. Then, the min and max values in the unit of degrees per second (dps) need to be converted to in the unit of integer LSBs according to the sensitivity parameters at those FS ranges.

When the gyroscope is in normal mode, “Data_no_ST” (see Figure 1) means

\[
G_x = x_0; \quad G_y = y_0; \quad G_z = z_0
\]

At the same stationary position, when the gyroscope is in self-test mode, “Data_ST” means,

\[
G_x' = x_0 + \Delta_x; \quad G_y' = y_0 + \Delta_y; \quad G_z' = z_0 + \Delta_z
\]

Then, the self-test results are,

\[
\Delta_x = \text{abs}(G_x' - G_x); \quad \Delta_y = \text{abs}(G_y' - G_y); \quad \Delta_z = \text{abs}(G_z' - G_z)
\]

If \{(\text{min}_x < \Delta_x < \text{max}_x) \text{ and } (\text{min}_y < \Delta_y < \text{max}_y) \text{ and } (\text{min}_z < \Delta_z < \text{max}_z)\}
Then the sensor has passed the test. Otherwise, the sensor fails.

Where, \( \min_{x,y,z} \) and \( \max_{x,y,z} \) are related to the full scale range in the unit of integer LSBs.

The self-test results when self-test0 is selected should be close to the self-test1 results when self-test1 is selected. You only need to pick either self-test0 or self-test1 mode for self-test at any random stationary position. Figure 1 shows the self-test procedure diagram as an example for a digital accelerometer or a digital gyroscope.
Figure 1: Self-test procedure for an Accelerometer or a Gyroscope

Accelerometer

2. Self-test sample codes

2.1. Accelerometer:
We’ve chosen the LIS3DH accelerometer from STMicroelectronics [1] for the sample code below:

Void ACC_self_test (void)
{
    // ACC is in power down mode after power up

    // set the control register (23h) to ±2g FS, normal mode with BDU (Block Data Update) and HR (High Resolution) bits enabled
    Write(0x23, 0x88);

    // set the control register (20h) to 50Hz ODR (Output Data Rate) with X/Y/Z axis enabled.
    Write(0x20, 0x47);

    Wait for (1/ODR+1ms) = 21mS for the accelerometer to be stabilized [1]

    // read the accelerometer data 5 times in normal mode
    for (i = 0; i < 5; i++)
    {
        Ax[i] = Read(ACC_X);
        Ay[i] = Read(ACC_Y);
        Az[i] = Read(ACC_Z);
    }

    // average the raw data
    Ax_no_st = (Ax[0] + Ax[1] + ... + Ax[4]) / 5;
    Ay_no_st = (Ay[0] + Ay[1] + ... + Ay[4]) / 5;
Az_no_st = (Az[0] + Az[1] + ... + Az[4]) / 5;

// enable self-test 0 at +/-2g FS with BDU and HR bits enabled Wait for (1/ODR + 1) = 21mS for the accelerometer to be stabilized
Write(0x23, 0x8A);

// read the accelerometer data 5 times in self-test0 mode
for (i = 0; i < 5; i++)
{
    Ax'[i] = Read(ACC_X);
    Ay'[i] = Read(ACC_Y);
    Az'[i] = Read(ACC_Z);
}

// average the raw data
Ax_st = (Ax'[0] + Ax'[1] + ... + Ax'[4]) / 5;
Ay_st = (Ay'[0] + Ay'[1] + ... + Ay'[4]) / 5;
Az_st = (Az'[0] + Az'[1] + ... + Az'[4]) / 5;

// calculate the absolute self-test values of with self-test minus without self-test
ST_x = abs(Ax_st - Ax_no_st);
ST_y = abs(Ay_st - Ay_no_st);
ST_z = abs(Az_st - Az_no_st);

// determine if the accelerometer passes or fails the self-test
if (((min_x <= ST_x <= max_x) && (min_y <= ST_y <= max_y) && (min_z <= ST_z <= max_z))
{ 
    The accelerometer passes;
}
else 
{
    The accelerometer fails;
}

2.2. Gyroscope

We’ve based the sample code below on the L3GD4200D/L3GD20 gyroscope from STMicroelectronics [2].

Void GYRO_self_test (void)
{
    // GYRO is in power down mode after power up
    // set the control register (24h) to Cut-Off Frequency = 50Hz
    Write(0x24, 0x02);

    // set the control register (23h) to ±2000dps FS, self-test0 mode with BDU bit enabled
    Write(0x23, 0xA0);

    // set the control register (20h) to 200Hz ODR with X/Y/Z axis enabled.
    Write(0x20, 0x6F);
Wait for 800mS for the gyroscope to be stabilized. 

// read the gyroscope data 5 times in self-test0 mode 
for (i = 0; i < 5; i++) 
{
    Gx[i] = Read(GYRO_X);
    Gy[i] = Read(GYRO_Y);
    Gz[i] = Read(GYRO_Z);
}

// average the raw data

Gx_no_st = (Gx[0] + Gx[1] + ... + Gx[4]) / 5;
Gy_no_st = (Gy[0] + Gy[1] + ... + Gy[4]) / 5;
Gz_no_st = (Gz[0] + Gz[1] + ... + Gz[4]) / 5;

// enable self-test 0 at +/-2000dps FS with BDU enabled, Wait for 60mS for the gyroscope to be stabilized
Write(0x23, 0xA2);

// read the gyroscope data 5 times in self-test0 mode 
for (i = 0; i < 5; i++) 
{
    Gx'[i] = Read(GYRO_X);
    Gy'[i] = Read(GYRO_Y);
    Gz'[i] = Read(GYRO_Z);
// average the raw data
Gx_st = (Gx'[0] + Gx'[1] + ... + Gx'[4]) / 5;
Gy_st = (Gy'[0] + Gy'[1] + ... + Gy'[4]) / 5;
Gz_st = (Gz'[0] + Gz'[1] + ... + Gz'[4]) / 5;

// calculate the absolute self-test values of with self-test minus without self-test
ST_x = \text{abs}(Gx_st - Gx_no_st);
ST_y = \text{abs}(Gy_st - Gy_no_st);
ST_z = \text{abs}(Gz_st - Gz_no_st);

if ((\text{min}_x <= ST_x <= \text{max}_x) && (\text{min}_y <= ST_y <= \text{max}_y) && (\text{min}_z <= ST_z <= \text{max}_z))
{
    The gyroscope passes;
}
Else
{
    The gyroscope fails;
}

3. Conclusion

The best and easiest way to determine if MEMS accelerometers and gyroscopes are
functioning after PCB assembly is with the embedded self-test feature. With the high volume production numbers necessary for consumer and other mass market applications, this feature has become mandatory as it eliminates the need for mechanical tests on each product’s PCB.

4. References

1. STMicroelectronics, Inc.

   AN3308: LIS3DH: MEMS digital output motion sensor ultra-low-power high performance 3-axis “nano” accelerometer

   [Link](http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/APPLICATION_NOTE/CD00290365.pdf)

2. STMicroelectronics, Inc.

   AN3393: L3G4200D: three axis digital output gyroscope

   [Link](http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/APPLICATION_NOTE/DM00026768.pdf)