DT0100
Design tip

Setting up free-fall recognition with ST's MEMS accelerometers

By Vladimir JANOUSEK, Zuzana JIRANKOVA and Petr STUKJUNGER

Main components	
LIS2DW12	MEMS digital output motion sensor: high-performance ultra-low-power 3-axis "femto" accelerometer
LIS2DH12	MEMS digital output motion sensor: ultra-low-power high-performance 3-axis "femto" accelerometer

Purpose and benefits

This design tip explains how to enable and personalize free-fall recognition feature of MEMS accelerometers from STMicroelectronics.

First we explain this embedded feature, what it does and how it can be parameterized. Then we discuss the impact of its parameters on detection results. Finally we show using the two most frequently used ST accelerometers, LIS2DW12 and LIS2DH12, exact settings and example source codes for implementing the motion-detection feature in applications.

Description

By definition free-fall is the motion of an object when gravity is the only force acting upon it. In reality it is a situation when the object is usually falling.

Free-fall detection was one of the first applications of MEMS accelerometers which was utilized in computer hard drives.

MEMS accelerometers can recognize free-fall, because it is in the end a kind of dynamic motion with acceleration. When a device is laying on a table for example, the MEMS accelerometer senses a specific force that has the opposite direction of the Earth's gravity and keeps the device in a still position. In this situation the accelerometer outputs an acceleration value of 1 *g*. When free-fall occurs, the above-mentioned specific force is removed and the accelerometer will sense no acceleration on any axis. This fact is utilized by the free-fall recognition feature embedded in ST's MEMS accelerometers.

Parameterization

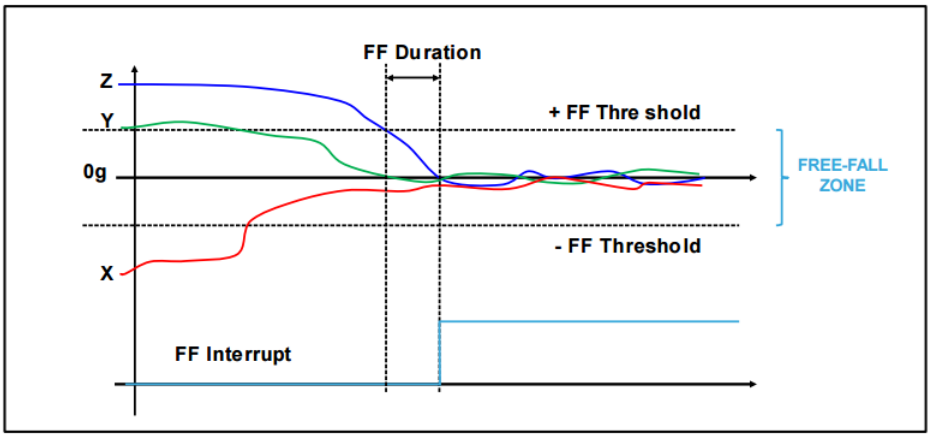
When a device is in free-fall, acceleration measured along all the axes of the accelerometer goes to zero.

For event recognition there are two parameters to set up – threshold and duration.

Threshold defines an amplitude (free-fall zone - see figure below). Accelerations on all three axes need to be smaller than the threshold to generate an interrupt.

Duration defines a minimum time period for which accelerations measured on all axes must be smaller than the threshold. Duration is dependent on the ODR selected – 1LSB equals 1/ODR. It can be also interpreted as the number of consecutive samples that have to be smaller than the threshold.

Figure 1. Free-fall recognition: threshold and duration parameters



Threshold can be used to adjust the sensitivity of free-fall recognition. A lower threshold sets a higher requirement on an event being defined as free-fall. It seems to be wise to put it low, however it could result in missing some free-falls, especially short ones. Duration can be used to reject false-positive detection. A higher duration will avoid detection of free-fall in case there are only random movements.
It is useful to fine tune case-by-case both threshold and duration. This way one can achieve the most accurate results.

When utilizing free-fall recognition, the full scale of the accelerometer should be set to the lowest available option.

In some cases it will also be necessary to change the ODR (output data rate) of the sensor – a higher ODR means higher sensitivity in event recognition and also the possibility to detect shorter free-fall events. The length of a free-fall event can be calculated by this formula:

$$y(t) = \frac{1}{2} g t^2$$

where *y(t)* is altitude travelled, *t* is time and *g* is gravity acceleration.

Table 1. shows the altitude difference of a fall and the time that it takes an object to fall. Assuming that the first sample fulfills the condition set by the threshold parameter, a free-fall event of 1/ODR duration can be detected.

Table 1. Free-fall time with respect to altitude

Altitude difference [mm]	Free-fall time [ms]
10	45.2
20	63.9
100	142.8
500	319.3
1000	451.5

Table 2. shows the shortest theoretical free-fall events to be detected using different ODRs and different duration settings. **Real resolution depends on the sensor noise level, sensor resolution and other factors.**

Table 2. Minimum theoretical free-fall time and altitude at different ODRs

ODR [Hz]	Minimum free-fall time [ms]				Minimum altitude difference [mm]			
	D = 2	D = 5	D = 10	D = 50	D = 2	D = 5	D = 10	D = 50
25	80	200	400	2000	31.4	196.2	784.8	19620
50	40	100	200	1000	7.8	49.1	196.2	4905
100	20	50	100	500	2	12.3	49.1	1226.3
200	10	25	50	250	0.5	3.1	12.3	306.6
400	5	12.5	25	125	0.1	0.8	3.1	76.6

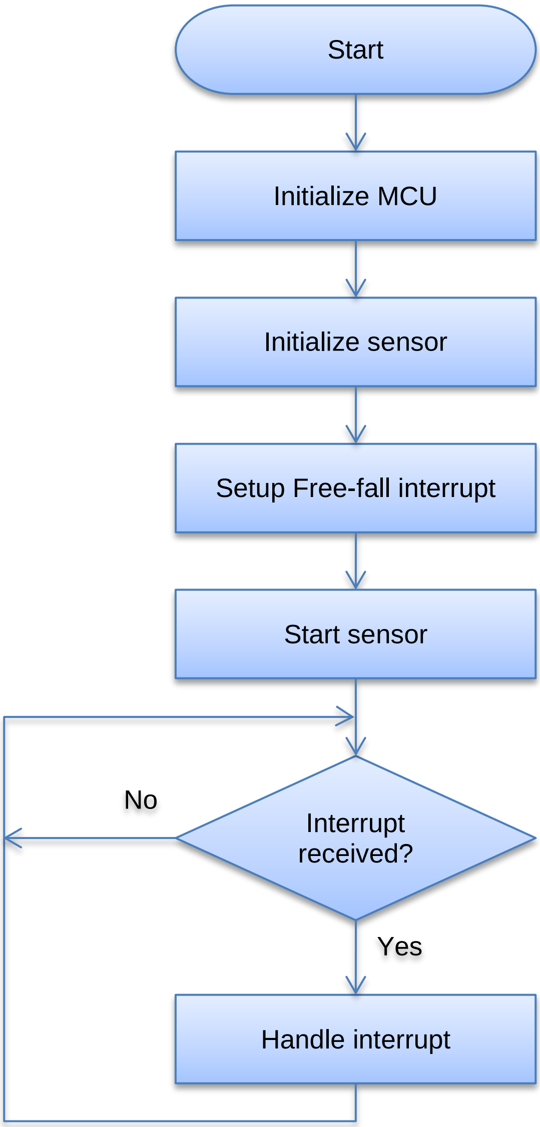
Note 1: D means Duration i.e. number of samples.
Note 2: Shaded values are not realistic in terms of practical use of a sensor.

Table 3. shows how to set up an accelerometer sensor in order to detect free-fall of a certain altitude difference. Based on altitude difference and ODR, the configuration of the duration parameter to safely detect a free-fall event can be determined.

Table 3. Minimum duration with respect to altitude of free-fall

Altitude difference [mm]	Duration [-]				
	ODR = 25 Hz	ODR = 50 Hz	ODR = 100 Hz	ODR = 200 Hz	ODR = 400 Hz
30	2	4	8	16	32
50	3	6	11	21	41
100	4	8	15	29	58
200	6	11	21	41	81
500	8	16	32	64	128

Flowchart



Setting up free-fall recognition with the LIS2DW12

To enable free-fall recognition in the LIS2DW12 you need to:

- Initialize the MCU
- Set bit **INT1_FF** in **CTRL4_INT1_PAD_CTRL** register (23h)
- Set desired threshold to bits **FF_THS [2:0]** in **FREE_FALL** register (36h)
- Set desired duration to bits **FF_DUR [5:0]** in **FREE_FALL** and **WAKE_UP_DUR** registers (35h and 36h)
- Set sensor **ODR to 100 Hz** (recommended) using **ODR[3:0]** bits and operating mode to Low-Power mode 1 using **MODE[1:0]** and **LP_MODE[1:0]** bits in **CTRL1** register (20h)

- Set bit **INTERRUPTS_ENABLE** in **CTRL7** register (3Fh)

✎ *Recommended value for threshold is FF_THS = 0b011 corresponding to ~ 10 x 31.25 mg = 312 mg; details of choosing the right threshold are described in the chapter "Parameterization"*

✎ *Recommended value for duration is FF_DUR=0b000011 corresponding to 3 x 1/ODR = 30 ms where ODR is 100 Hz; details of choosing the right duration are described in the chapter "Parameterization"*

Pseudocode for the LIS2DW12

```
void LIS2DW12_INT1_handler(void)
{
  print("Free-fall detected\r\n");
  /* … */
}

int main(void)
{
  init_MCU();                /* Initialize MCU clock and pins */
  print("Starting program\r\n");

  /* Initialization of sensor */
  write_reg(0x25, 0x00);    /* CTRL6(25h): Set Full-scale to +/-2g */

  /* Free-fall recognition enable */
  write_reg(0x23, 0x10);    /* CTRL4_INT1_PAD_CTRL(23h): Enable Free-fall on INT1
pin */
  write_reg(0x35, 0x00);    /* WAKE_UP_DUR(35h): Bit 7 is MSB of FF duration */
  write_reg(0x36, 0x1B);    /* FREE_FALL(36h): Set Free-fall duration and
threshold */

  /* Start sensor */
  write_reg(0x20, 0x50);    /* CTRL1(20h): Set ODR 100Hz,Low-Power mode 1 */
  write_reg(0x3f, 0x20);    /* ABS_INT_CFG(3Fh): Enable interrupts */

  while (1)
  {
    /* … */
  }
}
```

Setting up free-fall recognition with the LIS2DH12

To enable free-fall recognition in LIS2DH12 you need to:

- Initialize the MCU
- Set bit **I1_IA1** in CTRL_REG3 register (22h)
- Set bits **AOI**, **ZLIE**, **YLIE**, **XLIE** in **INT1_CFG** register (30h)
- Set desired threshold to bits **THS [6:0]** in **INT1_THS** register (32h)
- Set desired duration to bits **D [6:0]** in **INT1_DURATION** register (33h)
- Start sensor with **ODR low-power 100 Hz** (recommended) - bits **ODR[3:0]** and **LPen** bit in **CTRL_REG1** register (20h)

✎ *Recommended value for threshold is THS = 0b0010110 corresponding to 22 × 16 mg = 352 mg; details of choosing the right threshold are described in the chapter "Parameterization"*

✎ *Recommended value for duration is D = 0b0000011 corresponding to 3 x 1/ODR = 30 ms where ODR = 100 Hz; details of choosing the right duration are described in the chapter "Parameterization"*

Pseudocode for the LIS2DH12

```
void LIS2DH12_INT1_handler(void)
{
  print("Free-fall detected\r\n");
  /* … */
}
int main(void)
{
  init_MCU();                /* Initialize MCU clock and pins */
  print("Starting program\r\n");

  /* Initialization of sensor */
  write_reg(0x22, 0x40);    /* CTRL_REG3(22h): IA1 interrupt on INT1 pin */
  write_reg(0x23, 0x00);    /* CTRL_REG4(23h): Set Full-scale to +/-2g */

  /* Free-fall recognition enable */
  write_reg(0x30, 0x95);    /* INT1_CFG(30h): INT1 Configuration */
  write_reg(0x32, 0x16);    /* INT1_THS(32h): INT1 Threshold set */
  write_reg(0x33, 0x03);    /* INT1_DURATION(33h): INT1 Duration set */

  /* Start sensor */
  write_reg(0x20, 0x5f);    /* CTRL_REG1(20h): Start sensor at ODR 100Hz Low-
power mode */
  delay(1);                 /* Settling time 1 ms */
```

```
  while (1)
  {
    /* … */
  }
}
```

Support material

Related design support material
Product evaluation board – X-NUCLEO-IKS01A2, Motion MEMS and environmental sensor expansion board for STM32 Nucleo
Product evaluation board – STEVAL-MKI179V1, LIS2DW12 adapter board for a standard DIL 24 socket
Product evaluation board – STEVAL-MKI151V1, LIS2DH12 3-axis accelerometer adapter board for standard DIL 24 socket, compatible with STEVAL-MKI109V2
Documentation
Datasheet LIS2DW12, High-performance ultra-low-power 3-axis "femto" accelerometer
Datasheet LIS2DH12, High-performance ultra-low-power 3-axis "femto" accelerometer
Application note AN5038, LIS2DW12: always-on 3D accelerometer
Application note AN5005, LIS2DH12: MEMS digital output motion sensor ultra-low-power high-performance 3-axis "nano" accelerometer

Revision history

Date	Version	Changes
04-May-2018	1	Initial release