DT0101
Design tip

Setting up single-tap and double-tap 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 the single-tap and double-tap recognition feature of MEMS accelerometers from STMicroelectronics.

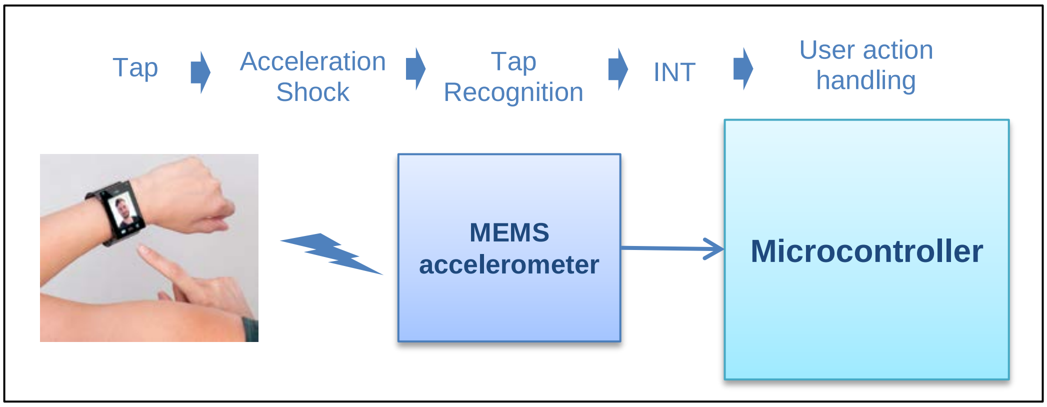
First we explain this embedded feature, what it does and how it can be parameterized. Then we discuss the impact of its parameters on detection results. Finally, we show using the two most frequently used ST accelerometers, LIS2DW12 and LIS2DH12, exact settings and example source codes for implementing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

Description

The single-tap and double-tap recognition feature allows detecting actions similar to the single click and double click of a mouse. The difference in using a MEMS accelerometer is that there is no need for a mechanical button to be pressed. This feature allows easy implementation of user interfaces of wearable, portable and other devices. Replacing the mechanical button brings more robust, user-friendly and cheaper design as well as the possibility of a higher degree of system integration.

Single-tap is the action when the user taps with his finger on the device casing. It is detected by the accelerometer as an acceleration shock. The accelerometer then informs the host microcontroller of this finger tap by an interrupt signal. Figure 1. depicts this flow.
Double-tap is basically a combination of two consecutive single-tap actions. Recognizing two taps as a different action and linking different application behavior to single-tap and double-tap events allow even more flexibility for controlling the application. Using an MP3 player as an example, single-tap can be used to start/stop playback, while double-tap can be used to move to the next song.

Figure 1. Tap event recognition flow

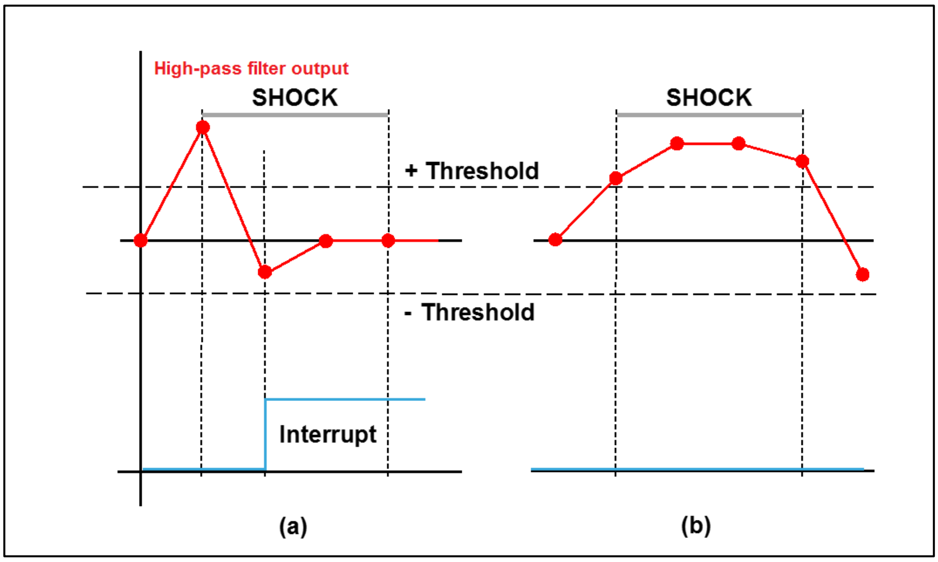


Parameterization

For **single-tap** recognition there are two parameters to set up – threshold and shock time window.

Single-tap recognition interrupt is generated if the acceleration exceeds the preset threshold and falls below within the shock time window - see Figure 2. below.

Figure 2. Single-tap recognition



(a) – Single-tap event recognized, (b) – Single-tap event not recognized, shock lasted too long

Threshold defines the intensity of the shock needed to generate a single-tap interrupt. The amplitude of the shock is proportional to the force that the user applies in either a positive or negative direction.

Shock time window sets the maximum allowed duration of a shock. During this window the acceleration has to fall below the pre-selected threshold. Figure 2. depicts functionality of the shock window parameter.

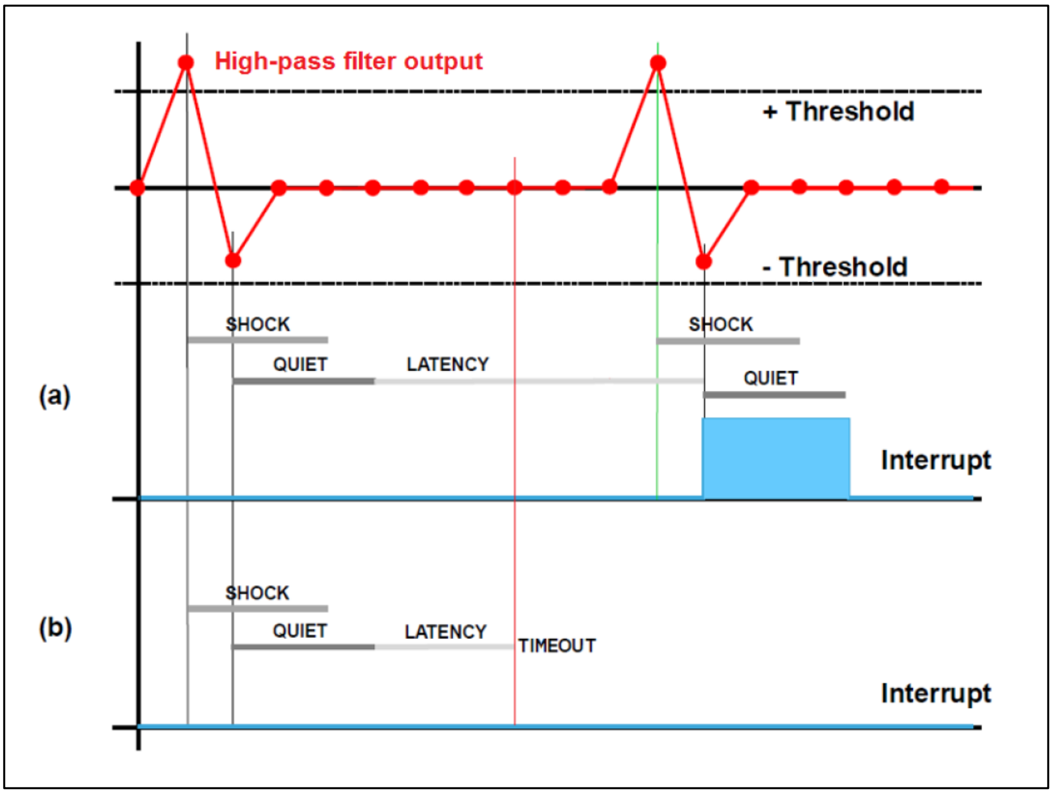
For **double-tap** recognition there are four parameters to set up. Besides the threshold and shock time window which are the same as for single-tap recognition, there are also the quiet time window and latency time window parameters.

Double-tap interrupt is generated if there are two consecutive single taps recognized. The time between two single taps shall not be shorter than the quiet time window and longer than latency time window - see Figure 3. below.

Quiet time window defines the period of time after the first detected single tap where there should not be any other shock detected.

Latency time window defines the time period starting after the quiet time window where the second single tap should happen in order to recognize this event as a double tap.

Figure 3. Double-tap recognition



(a) – Double-tap event recognized, (b) – Double-tap event not recognized, Latency time window shorter

The sensitivity of tap recognition can be modified using the threshold. A lower threshold means higher sensitivity and vice versa. However a threshold that is too low will lead to too many false positives.
The shock time window helps to limit the number of false positives by avoiding long shocks to be detected as a tap. A very short shock time window will decrease tap detection sensitivity. Nevertheless a shock time window that is too long will increase the number of false positives.
Quiet time and latency time windows used for double tap are usually not sensitive to device placement and are rather subject to personalization.

For single-tap and double-tap recognition it is usually enough to utilize the lowest full scale (±2 *g*), because of the high sensitivity required. In some cases where shocks are bigger, full scale (±4 *g*) could be recommended.
Output data rate (ODR) selection also affects tap recognition. As acceleration shocks generated by tap events are usually short, ODRs of 400 Hz and higher are recommended. For low-power applications an ODR of 200 Hz could be used, carefully considering the configuration of the tap parameters.

Single-tap and double-tap recognition has to be always fine-tuned carefully for each application case. The reason is that each application uses a different mounting for the sensor – it makes a significant difference whether the device is on a wrist, in a shirt pocket, on one's head (headphones) or lying on a table. Each position will change the acceleration profile generated by a tap action. This has to be considered case-by-case. A GUI application for evaluation of sensor functionality (e.g. ST's Unicleo-GUI or Unico) can greatly help with the process of fine-tuning the parameters of tap recognition.

Single-tap and double-tap in the LIS2DW12 and LIS2DH12

What was described so far is valid for both the LIS2DW12 and LIS2DH12. However there are some differences between the two sensors in functionality and naming when using the single-tap and double-tap recognition feature (see Table 1.).

Table 1. Differences in functionality of single-tap and double-tap recognition

Functionality	LIS2DW12	LIS2DH12
Input data filter	HP filter	No filter / HP filter
Thresholds	3 (one per axis)	1 (common to all)
Tap priorities	Yes	No
Simultaneous detection of single and double-tap	Yes	No

As it can be seen above, the LIS2DW12 offers more configuration options and flexibility. Nevertheless in the majority of cases the resulting performance of single-tap and double-tap recognition implemented in either the LIS2DW12 or LIS2DH12 is comparable. Table 2. shows differences in naming related to the single-tap and double-tap recognition features of the LIS2DW12 and LIS2DH12.

Table 2. Differences in naming related to single-tap and double-tap recognition

LIS2DW12	LIS2DH12
Single-tap	Single-click
Double-tap	Double-click
Shock time window	Time limit
Quiet time window	Time latency
Latency time window	Time window

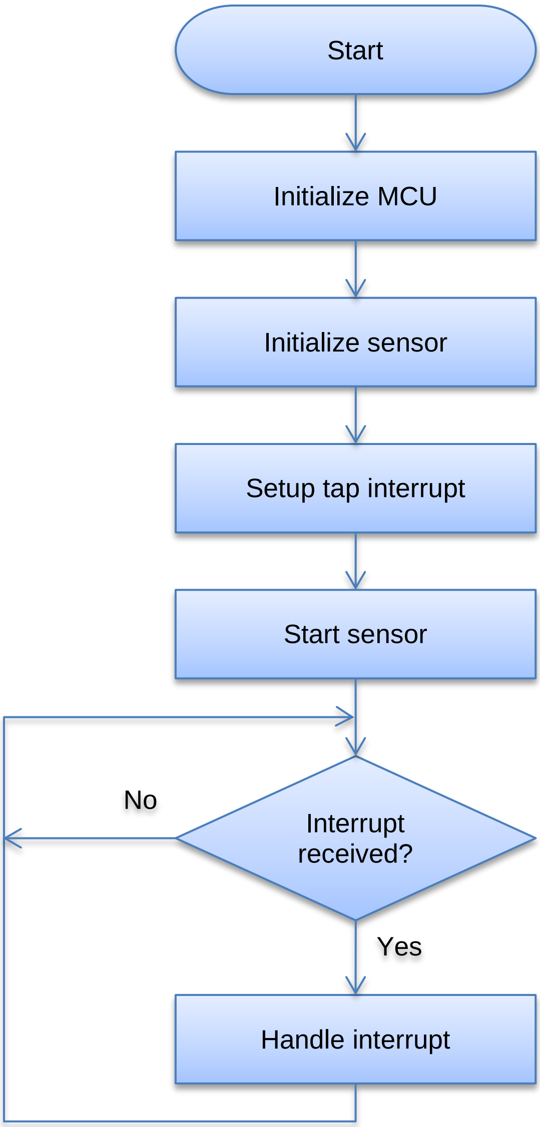
Enabling single-tap simultaneously with double-tap in the LISDW12

In the LIS2DW12 it is possible to enable both features at the same time. Use the setting for double-tap and in the CTRL4_INT1_PAD_CTRL register (23h) set bit INT1_SINGLE_TAP. Now an interrupt will be generated both for single-tap and double-tap. To recognize which tap was detected, use the TAP_SRC register (39h) or STATUS register (27h) and read bits SINGLE_TAP and DOUBLE_TAP.

✎ *The first tap of a double-tap will be recognized as a single-tap in this setting.*

✎ *It is not possible to enable both single-tap and double-tap recognition in the LIS2DH12.*

Flowchart


Start
Initialize MCU
Initialize sensor
Setup tap interrupt
Start sensor
No
Interrupt received?
Yes
Handle interrupt

Setting up single-tap recognition in the LIS2DW12

To enable single–tap recognition you need to:

- Initialize the MCU
- Set **full scale to ±2 *g*** using bits **FS[1:0]** and enable low-noise **using** bit **LOW_NOISE** in **CTRL6** register (25h)
- Set bit **INT1_SINGLE_TAP** in **CTRL4_INT1_PAD_CTRL** register (23h)
- Set desired X-axis threshold to bits **TAP_THSX_[4:0]** in **TAP_THS_X** register (30h)
- Set desired Y-axis threshold to bits **TAP_THSY_[4:0]** and axis priority to bits **TAP_PRIOR_[2:0]** in **TAP_THS_Y** register (31h)
- Set desired Z-axis threshold to bits **TAP_THSZ_[4:0]** and set bits **TAP_X_EN**, **TAP_Y_EN** and **TAP_Z_EN** in **TAP_THS_Z** register (32h)
- Set desired shock time window to bits **SHOCK[1:0]** in **INT_DUR** register (33h)
- Set sensor's **ODR to 400 Hz** (recommended) using **ODR[3:0]** bits and operating mode to high-performance mode (14-bit) using **MODE[1:0]** in **CTRL1** register (20h)
- Set bit **INTERRUPTS_ENABLE** in **CTRL7** register (3Fh)

✎ *Recommended value for the axis threshold is TAP_THS(X/Y/Z) = 0b1001 corresponding to 9 × FS / 32 = 562.5 mg where FS = ±2 g; details of choosing the right threshold are described in the chapter "Parameterization".*

✎ *Recommended value for tap priority is TAP_PRIOR = 0b000 corresponding to equal priorities for all axes.*

✎ *Recommended value for the shock time window is SHOCK = 0b10 corresponding to 2 × 8 / ODR = 40 ms where ODR is 400 Hz; details of choosing the right shock time window are described in the chapter "Parameterization".*

Pseudocode – single-tap in the LIS2DW12

```
void LIS2DW12_INT1_handler(void)
{
  print("Single-tap recognized\r\n");
  /* … */
}

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

  /* Initialization of sensor */
  write_reg(0x25, 0x04);   /* CTRL6(25h): Set Full-scale to +/-2g, Low-noise 
enabled */

  /* Single-tap recognition enable */
```

```
  write_reg(0x23, 0x40);   /* CTRL4_INT1_PAD_CTRL(23h): Enable Single-tap
interrupt */
  write_reg(0x30, 0x09);   /* TAP_THS_X (30h): Threshold on X */
  write_reg(0x31, 0x09);   /* TAP_THS_Y (31h): Threshold on Y and default
priority*/
  write_reg(0x32, 0xe9);   /* TAP_THS_Z (32h): Threshold on Z and enable all axes
in tap recognition */
  write_reg(0x33, 0x02);   /* INT_DUR (33h): Shock duration */

  /* Start sensor */
  write_reg(0x20, 0x74);   /* CTRL1(20h): Set ODR 400Hz, High performance mode
(14 bit) */
  delay(3);                /* Settling time ( 1 sample, i.e. 1/ODR ) */
  write_reg(0x3f, 0x20);   /* CTRL7 (3Fh): Enable interrupts */

  while (1)
  {
    /* … */
  }
}
```

Setting up double-tap recognition in the LIS2DW12

To enable double–tap recognition you need to:

- Initialize the MCU
- Set **full scale to ±2 *g*** bits using bits **FS[1:0]** and enable low-noise **using** bit **LOW_NOISE** in **CTRL6** register (25h)
- Set bit **INT1 _TAP** in **CTRL4_INT1_PAD_CTRL** register (23h)
- Set desired X-axis threshold to bits **TAP_THSX_[4:0]** in **TAP_THS_X** register (30h)
- Set desired Y-axis threshold to bits **TAP_THSY_[4:0]** and axis priority to bits **TAP_PRIOR_[2:0]** in **TAP_THS_Y** register (31h)
- Set desired Z-axis threshold to bits **TAP_THSZ_[4:0]** and set bits **TAP_X_EN**, **TAP_X_EN** and **TAP_X_EN** in **TAP_THS_Z** register (32h)
- Set desired latency time window to bits **LATENCY[3:0]**, shock time window to bits **SHOCK[1:0]** and quiet time window to bits **QUIET[1:0]** in **INT_DUR** register (33h)
- Set bit **SINGLE_DOUBLE_TAP** in **WAKE_UP_THS** register (34h) to enable double-tap recognition
- Set sensor's **ODR to 400 Hz** (recommended) using **ODR[3:0]** bits and operating mode to high-performance mode (14-bit) using **MODE[1:0]** in **CTRL1** register (20h)
- Set bit **INTERRUPTS_ENABLE** in **CTRL7** register (3Fh)

Recommended value for the axis threshold is TAP_THS(X/Y/Z) = 0b1100 corresponding to 12 × FS / 32 = 750 mg where FS = ±2 g; details of choosing the right threshold are described in the chapter "Parameterization".

- ✎ *Recommended value for tap priority is TAP_PRIOR = 0b000 corresponding to equal priorities for all axes.*

- ✎ *Recommended value for the shock time window is SHOCK = 0b11 corresponding to 3 × 8 / ODR = 60 ms, for quiet time window QUIET = 0b11 corresponding to 3 × 4 / ODR = 30 ms and latency time window LATENCY = 0b0111 corresponding to 7 × 32 / ODR = 560 ms where ODR is 400 Hz; details of choosing the right time windows are described in chapter "Parameterization".*

Pseudocode – double-tap in the LIS2DW12

```
void LIS2DW12_INT1_handler(void)
{
  print("Double-tap recognized\r\n");
  /* … */
}
int main(void)
{
  init_MCU();              /* Initialize MCU clock and pins */
  print("Starting program\r\n");

  /* Initialization of sensor */
  write_reg(0x25, 0x04);   /* CTRL6(25h): Set Full-scale to +/-2g, Low-noise
enabled */

  /* Double-tap recognition enable */
  write_reg(0x23, 0x08);   /* CTRL4_INT1_PAD_CTRL(23h): Enable Double-tap
interrupt */
  write_reg(0x30, 0x0c);   /* TAP_THS_X (30h): Threshold on X */
  write_reg(0x31, 0x0c);   /* TAP_THS_Y (31h): Threshold on Y and default
priority*/
  write_reg(0x32, 0xec);   /* TAP_THS_Z (32h): Threshold on Z and enable all axes
in tap recognition */
  write_reg(0x33, 0x7f);   /* INT_DUR (33h): Shock, latency and quiet duration */
  write_reg(0x34, 0x80);   /* WAKE_UP_THS (34h): Enable Double-tap recognition */

  /* Start sensor */
  write_reg(0x20, 0x74);   /* CTRL1(20h): Set ODR 400Hz, High performance mode
(14 bit) */
  delay(3);                /* Settling time ( 1 sample, i.e. 1/ODR ) */
  write_reg(0x3f, 0x20);   /* CTRL7 (3Fh): Enable interrupts */

  while (1)
  {
    /* … */
  }
}
```

Setting up single-tap recognition in the LIS2DH12

To enable single–tap recognition you need to:

- Initialize the MCU
- Set bit HPCLICK in **CTRL_REG2** register (21h) to enable the high-pass filter on tap detection
- Set bit **I1_CLICK** in **CTRL_REG3** register (22h)
- Set bits **ZS**, **YS** and **XS** in **CLICK_CFG** register (38h)
- Set desired threshold to bits **THS[6:0]** in **CLICK_THS** register (3Ah)
- Set desired time limit to bits **TLI[6:0]** in **TIME_LIMIT** register (3Bh)
- Set time latency to bits **TLA[7:0]** in **TIME_LATENCY** register (3Ch) to define the interrupt signal duration
- Start sensor with **ODR 400 Hz, high-resolution mode** (recommended) - bits **ODR[3:0]** in **CTRL_REG1** register (20h) and bit **HR** in **CTRL_REG4** register (23h)

✎ *Recommended value for the axis threshold is THS = 0b0110000 corresponding to 48 × FS / 128 = 750 mg where FS is ±2 g; details of choosing the right threshold are described in the chapter "Parameterization".*

✎ *Recommended value for the time limit is TLI = 0b00011000 corresponding to 24 × 1 / ODR = 60 ms where ODR is 400 Hz; details of choosing the right values are described in the chapter "Parameterization".*

Pseudocode – single-tap in the LIS2DH12

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

  /* Initialization of sensor */
  write_reg(0x21, 0x04);   /* CTRL_REG2 (21h): Enable HP filter on tap detection
*/
  write_reg(0x22, 0x80);   /* CTRL_REG3 (22h): TAP interrupt on INT1 pin */
  write_reg(0x23, 0x08);   /* CTRL_REG4 (23h): Set Full-scale to +/-2g, set HR
bit */

  /* Single-tap recognition enable */
  write_reg(0x38, 0x15);   /* CLICK_CFG (38h): Tap config */
  write_reg(0x3a, 0x30);   /* CLICK_THS (3Ah): Tap threshold set */
  write_reg(0x3b, 0x18);   /* TIME_LIMIT (3Bh): Tap time limit set */
```

```
  write_reg(0x3c, 0x02);   /* TIME_LATENCY (3Ch): Tap time latency set */

  /* Start sensor */
  write_reg(0x20, 0x77);   /* CTRL1(20h): Set ODR 400Hz */
  delay(18);               /* Settling time (7/ODR) */

  while (1)
  {
    /* … */
  }
}
```

Setting up double-tap recognition in the LIS2DH12

To enable double–tap recognition you need to:

- Initialize the MCU
- Set bit HPCLICK in **CTRL_REG2** register (21h) to enable the high-pass filter on tap detection
- Set bit **I1_CLICK** in **CTRL_REG3** register (22h)
- Set bits **ZD**, **YD** and **XD** in **CLICK_CFG** register (38h)
- Set desired threshold to bits **THS[6:0]** in **CLICK_THS** register (3Ah)
- Set desired time limit to bits **TLI[6:0]** in **TIME_LIMIT** register (3Bh)
- Set desired time latency to bits **TLA[7:0]** in **TIME_LATENCY** register (3Ch)
- Set desired time window to bits **TW[7:0]** in **TIME_WINDOW** register (3Dh)
- Start sensor with **ODR 400 Hz, high-resolution mode** (recommended) - bits **ODR[3:0]** in **CTRL_REG1** register (20h) and bit **HR** in **CTRL_REG4** register (23h)

✎ *Recommended value for the axis threshold is THS = 0b0110000 corresponding to 48 × FS / 128 = 750 mg where FS is ±2 g; details of choosing the right threshold are described in the chapter "Parameterization".*

✎ *Recommended value for the time limit is TLI = 0b00011000 corresponding to 24 × 1 / ODR = 60 ms, for time latency TLA = 0b00001100 corresponding to 12 × 1 / ODR = 30 ms and time window TW = 0b11100000 corresponding to 24 × 1 / ODR = 560 ms where ODR is 400 Hz; details of choosing the right time windows are described in the chapter "Parameterization".*

Pseudocode – double-tap in the LIS2DH12

```
void LIS2DH12_INT1_handler(void)
{
  print("Double-tap recognized\r\n");
  /* … */
}
int main(void)
```

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

  /* Initialization of sensor */
  write_reg(0x21, 0x04);   /* CTRL_REG2 (21h): Enable HP filter on tap detection
*/
  write_reg(0x22, 0x80);   /* CTRL_REG3 (22h): TAP interrupt on INT1 pin */
  write_reg(0x23, 0x08);   /* CTRL_REG4 (23h): Set Full-scale to +/-2g, set HR
bit */

  /* Double-tap recognition enable */
  write_reg(0x38, 0x2a);   /* CLICK_CFG (38h): Tap config */
  write_reg(0x3a, 0x30);   /* CLICK_THS (3Ah): Tap threshold set */
  write_reg(0x3b, 0x18);   /* TIME_LIMIT (3Bh): Tap time limit set */
  write_reg(0x3c, 0x0c);   /* TIME_LATENCY (3Ch): Tap time latency set */
  write_reg(0x3d, 0xe0);   /* TIME_WINDOW (3Dh): Tap time window set */

  /* Start sensor */
  write_reg(0x20, 0x77);   /* CTRL1(20h): Set ODR 400Hz */
  delay(18);               /* Settling time (7/ODR) */

  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: MEMS digital output motion sensor ultra-low-power high-performance 3-axis "nano" accelerometer
Application note AN5005, LIS2DH12: MEMS digital output motion sensor ultra-low-power high-performance 3-axis "nano" accelerometer

Revision history

Date	Version	Changes
23-May-2018	1	Initial release