

## LIS2HH12: MEMS 数字输出运动传感器超低功耗高性能 3 轴“pico”加速度计

### 引言

本文档介绍了以 LGA 封装提供的低压 3 轴数字量输出线性 MEMS 加速度计。

LIS2HH12 是属于“pico”系列的超低功耗高性能 3 轴线性加速度计，具有数字 I<sup>2</sup>C、SPI 串行接口标准输出。

器件具有超低功耗工作模式，可实现高级节能、智能睡眠唤醒（活动）以及恢复睡眠（不活动）功能。

LIS2HH12 具有  $\pm 2g/\pm 4g/\pm 8g$  的动态用户可选满量程，并能通过 10 Hz 到 800 Hz 的输出数据速率测量加速度。

器件可配置为通过检测惯性唤醒/自由落体事件以及通过器件自身的位置生成两个独立中断信号。两个中断发生器的阈值和时序可由终端用户动态设定。

LIS2HH12 内置 32 级先进先出（FIFO）缓冲器提供给用户存储数据，可以减少主控的干预。

LIS2HH12 采用纤薄的小型塑料平面网格阵列封装(LGA)，可确保在更大的温度范围（-40 °C 至+85 °C）内正常工作。

SMD 封装的超小尺寸和重量使其成为智能手机和可穿戴设备以及计步器、倾斜监测和活动识别等应用的理想选择。

## 1 寄存器

表 1. 寄存器

寄存器名	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
TEMP_L	0Bh	TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0
TEMP_H	0Ch	TD15	TD14	TD13	TD12	TD11	TD10	TD9	TD8
RESERVED	00-0Eh								
WHO_AM_I	0Fh	0	1	0	0	0	0	0	1
ACT_THS	1Eh	ATHS7	ATHS6	ATHS5	ATHS4	ATHS3	ATHS2	ATHS1	ATHS0
ACT_DUR	1Fh	ADUR7	ADUR6	ADUR5	ADUR4	ADUR3	ADUR2	ADUR1	ADUR0
CTRL1	20h	HR	ODR2	ODR1	ODR0	BDU	Zen	Yen	Xen
CTRL2	21h	-	DFC1	DFC0	HPM1	HPM0	FDS	HPIS1	HPIS2
CTRL3	22h	FIFO_EN	STOP_FTH	INT1_INACT	INT1_IG2	INT1_IG1	INT1_OVR	INT1_FTH	INT1_DRDY
CTRL4	23h	BW2	BW1	FS1	FS0	BW_SCALE_ODR	IF_ADD_INC	I2C_DISABLE	SIM
CTRL5	24h	DEBUG	SOFT_RESET	DEC1	DEC0	ST2	ST1	H_LActive	PP_OD
CTRL6	25h	BOOT	-	INT2_BOOT	INT2_IG2	INT2_IG1	INT2_Empty	INT2_FTH	INT2_DRDY
CTRL7	26h	-	-	DCRM2	DCRM1	LIR2	LIR1	4D_IG2	4D_IG1
STATUS	27h	ZYXOR	ZOR	YOR	XOR	ZYXDA	ZDA	YDA	XDA
OUT_X_L	28h	XD7	XD6	XD5	XD4	XD3	XD2	XD1	XD0
OUT_X_H	29h	XD15	XD14	XD13	XD12	XD11	XD10	XD9	XD8
OUT_Y_L	2Ah	YD7	YD6	YD5	YD4	YD3	YD2	YD1	YD0
OUT_Y_H	2Bh	YD15	YD14	YD13	YD12	YD11	YD10	YD9	YD8
OUT_Z_L	2Ch	ZD7	ZD6	ZD5	ZD4	ZD3	ZD2	ZD1	ZD0
OUT_Z_H	2Dh	ZD15	ZD14	ZD13	ZD12	ZD11	ZD10	ZD9	ZD8
FIFO_CTRL	2E	FMODE2	FMODE1	FMODE0	FTH4	FTH3	FTH2	FTH1	FTH0
FIFO_SRC	2F	FTH	OVR	空	FSS4	FSS3	FSS2	FSS1	FSS0
IG_CFG1	30h	AOI	6D	ZHIE	ZLIE	YHIE	YLIE	XHIE	XLIE
IG_SRC1	31h	-	IA	ZH	ZL	YH	YL	XH	XL
IG_THS_X1	32h	XTH1_7	XTH1_6	XTH1_5	XTH1_4	XTH1_3	XTH1_2	XTH1_1	XTH1_0



寄存器名	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
IG_THS_Y1	33h	YTH1_7	YTH1_6	YTH1_5	YTH1_4	YTH1_3	YTH1_2	YTH1_1	YTH1_0
IG_THS_Z1	34h	ZTH1_7	ZTH1_6	ZTH1_5	ZTH1_4	ZTH1_3	ZTH1_2	ZTH1_1	ZTH1_0
IG_DUR1	35h	WAIT1	DUR1_6	DUR1_5	DUR1_4	DUR1_3	DUR1_2	DUR1_1	DUR1_0
IG_CFG2	36h	AOI	6D	ZHIE	ZLIE	YHIE	YLIE	XHIE	XLIE
IG_SRC2	37h	-	IA	ZH	ZL	YH	YL	XH	XL
IG_THS2	38h	TH2_7	TH2_6	TH2_5	TH2_4	TH2_3	TH2_2	TH2_1	TH2_0
IG_DUR2	39h	WAIT2	DUR2_6	DUR2_5	DUR2_4	DUR2_3	DUR2_2	DUR2_1	DUR2_0
XL_REFERENCE	3Ah	XREF7	XREF6	XREF5	XREF4	XREF3	XREF2	XREF1	XREF0
XH_REFERENCE	3Bh	XREF15	XREF14	XREF13	XREF12	XREF11	XREF10	XREF9	XREF8
YL_REFERENCE	3Ch	YREF7	YREF6	YREF5	YREF4	YREF3	YREF2	YREF1	YREF0
YH_REFERENCE	3Dh	YREF15	YREF14	YREF13	YREF12	YREF11	YREF10	YREF9	YREF8
ZL_REFERENCE	3Eh	ZREF7	ZREF6	ZREF5	ZREF4	ZREF3	ZREF2	ZREF1	ZREF0
ZH_REFERENCE	3Fh	ZREF15	ZREF14	ZREF13	ZREF12	ZREF11	ZREF10	ZREF9	ZREF8

## 2 工作模式

LIS2HH12 提供两种不同的工作模式：下电模式和激活模式。在激活模式中，可选择六种不同的输出数据速率 (ODR)：800 Hz、400 Hz、200 Hz、100 Hz、50 Hz 和 10 Hz。从 800 Hz 到 100 Hz，每个 ODR 的电流消耗都是恒定的，而 50 Hz 和 10 Hz 进一步降低了电流消耗，具体取决于所选的 ODR。

施加电源后，LIS2HH12 执行一段 20 ms 的启动程序来从内部存储器加载校准参数。在启动阶段结束时，将器件自动配置为下电模式。

通过参考 LIS2HH12 数据表，CTRL1 寄存器的输出数据速率 ODR[2:0]位用于选择操作模式（下电模式、激活模式）和相关输出数据速率（表 2. 数据速率配置）。

表 2. 数据速率配置

ODR2	ODR1	ODR0	功率模式选择
0	0	0	下电模式
0	0	1	激活模式 10 Hz
0	1	0	激活模式 50 Hz
0	1	1	激活模式 100 Hz
1	0	0	激活模式 200 Hz
1	0	1	激活模式 400 Hz
1	1	0	激活模式 800 Hz
1	1	1	N. A.

表 3. 功耗 ( $\mu\text{A}$ ) 显示了不同工作模式下功耗典型值。

表 3. 功耗 ( $\mu\text{A}$ )

ODR[Hz]	电流消耗 [ $\mu\text{A}$ ] @ Vdd = 2.5 V [典型值]
下电	5
10	50
50	110
100	180
200	180
400	180
800	180

### 2.1 下电模式

器件处于下电模式时，器件内部几乎所有模块都会被关闭，以最大限度地降低功耗。数字接口 (I<sup>2</sup>C 和 SPI) 仍然在工作，以便能够与器件进行通信。保留配置寄存器的内容而不更新输出数据寄存器，因此可保持进入下电模式前存储器中采样的最后数据。

在下电模式期间，中断引脚继续在最后一个逻辑状态下驱动。

### 2.2 激活模式

在激活模式中，以通过 CTRL1 寄存器的 ODR[2:0]位选择的输出数据速率 (ODR) 生成数据。可选择六种不同的 ODR，以保证所需的应用带宽：800 Hz、400 Hz、200 Hz、100 Hz、50 Hz 和 10 Hz。

可通过 CTRL1 寄存器的 Zen、Yen 和 Xen 位单独启用三个加速度轴。可采用单轴、双轴或三轴的所有组合，与禁用的轴相关的输出寄存器被强制为 0000h。

数据中断生成和 FIFO 缓冲区有效，并可以分别通过 IG\_CFGx (30h, 36h)和 FIFO\_CTRL (2Eh)寄存器激活和配置。

可启用活动/不活动识别功能，以降低系统功耗。

### 2.3 数据稳定时间

LIS2HH12 读取链路包含了低通滤波，能够提高信噪比性能并降低混叠效应。因此，当器件从下电模式切换到激活模式或更改 ODR 时，必须考虑滤波器的建立时间。

表 4. 要丢弃的样本数 阐明了根据内部滤波器带宽和输出数据率选择，必须丢弃的样本数。通过设置 CTRL4 寄存器中的 BW [2:1]位可改变带宽，通过设置 CTRL1 中的 ODR [2:0]位可改变输出数据速率。

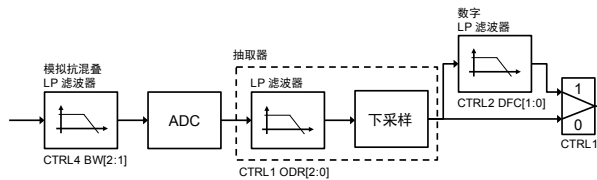
表 4. 要丢弃的样本数

ODR [Hz]	BW = 400 Hz	BW = 200 Hz	BW = 100 Hz	BW = 50 Hz
10	1	-	-	-
50	1	-	-	-
100	1	1	1	1
200	1	1	1	4
400	1	1	4	7
800	1	4	7	14

### 2.4 系统带宽

采样链可由 4 个级联模块表示：模拟低通滤波器、ADC 转换器、抽取滤波器和数字低通滤波器。

图 1. 采样链框图



来自机械模块的模拟信号在经 ADC 转换前通过抗混叠滤波器进行滤波，在激活模式中，对于大于 50 Hz 的 ODR，如果将 CTRL4 寄存器中的 BW\_SCALE\_ODR 位设为 1，则可以使用 CTRL4 寄存器中的 BW[2:1]位配置滤波器截止频率，否则（默认条件），自动将滤波器带宽设为 ODR/2。

表 5. 抗混叠滤波器带宽

CTRL4 BW[2:1]	带宽 [Hz]
00	400
01	200
10	100
11	50

表 6. 抗混叠带宽选项

ODR [Hz]	模拟滤波器截止频率 BW_SCALE_ODR = 0	模拟滤波器截止频率 BW_SCALE_ODR = 1
10/50	400 [Hz]	400 [Hz]
100	50 [Hz]	BW[2:1]
200	100 [Hz]	BW[2:1]
400	200 [Hz]	BW[2:1]
800	400 [Hz]	BW[2:1]

ADC 模块是一个模数转换器，可以 3.2 kHz 的恒定采样率转换输入模拟信号。在激活模式期间，ADC 周期性地接通/关闭，其占空比取决于所选数据速率。

抽取滤波器是执行信号滤波和下采样的数字模块。抽取器平均执行 N 个采样，以将采样率从

3.2 kHz 降至选定 ODR。低通抽取滤波器的截止频率取决于所选的 ODR，如表 7. 抽取滤波器带宽中所述。

表 7. 抽取滤波器带宽

ODR[Hz]	抽取滤波器截止频率[Hz]
10/50	66
100	133
200	266
400	534
800	1090

对于等于或大于 100 Hz 的 ODR，当 BW\_SCALE\_ODR = 0 时，可通过模拟滤波器保证抗混叠带宽。

通过将 CTRL1 寄存器中的 HR 位设为 1，可以使能数字低通滤波器。选择高分辨率模式时，将对来自抽取滤波器的数字信号进行额外的低通滤波，并通过 CTRL2 寄存器中的 DFC[1:0]位选择带宽。

表 8. 数字低通滤波器带宽

HR 位	CTRL2 DFC[1:0]	LP 截止频率 [Hz]
1	00	ODR/50
1	01	ODR/100
1	10	ODR/9
1	11	ODR/400

## 2.5 活动/不活动功能

活动/不活动识别功能能够减少系统功耗，可支持开发新型智能应用。

当活动/不活动识别功能激活时，LIS2HH12 能够自动将采样率降低至 10 Hz，当检测到唤醒中断事件时会立即增加 ODR 和带宽。

利用此功能，根据用户所选的加速事件，系统可以高效地从低功耗模式转换成全性能模式，反之亦然，因此可以保证节能和灵活性。

通过在 ACT\_THS (1Eh)中写入所需阈值来激活活动/不活动识别功能；阈值由 7 个位表示，并应用于正负方向。高通滤波器自动启用。

表 9. 活动/不活动控制寄存器

寄存器	分辨率[LSB]
ACT_THS	满量程/ 128 [mg]
ACT_DUR	8/ODR [s]

当加速度低于阈值至少 $(8 \text{ ACT\_DUR} + 1)/\text{ODR}$  的时间时，CTRL1 的 ODR [2:0]位被旁路（不活动），ODR 被内部设置位 10 Hz，尽管 CTRL1 的内容保持不变。当加速度超过阈值时，会立即恢复（活动）CTRL1 寄存器设置。当活动/不活动检测功能使能时，通过将 CTRL3 寄存器中的 INT1\_INACT 位设为“1”，可将状态驱动到 INT1 引脚。通过将 ACT\_THS 寄存器设为 00h，可自动禁用活动/不活动功能。

## 3 启动序列

当器件上电时，器件会自动从嵌入的内存中加载校准系数到内部寄存器中。启动程序完成后，也就是大约 20 毫秒后，器件会自动进入下电模式。要导通器件并采集加速度数据，必须选择其中一个激活模式（表 2. 数据速率配置），并通过 CTRL1 寄存器至少使能一个轴。

可使用下列通用序列对器件进行配置：

1. 写入 CTRL1 = 3Fh // X、Y、Z 使能，ODR = 100 Hz，BDU 使能
2. 写入 CTRL3 = 01h // INT1 上的数据准备就绪中断

### 3.1 读取加速度数据

#### 3.1.1 使用状态寄存器

该器件提供了一个 STATUS 寄存器，可用于轮询检查新的一组数据什么时候可以用。读取数据的步骤如下：

1. 读 STATUS
2. 如果 STATUS(ZYXDA) == 0，则转至 1
3. 如果 STATUS(ZYXOR) == 1，说明一些数据已被重写
4. 读 OUTX\_L
5. 读 OUTX\_H
6. 读 OUTY\_L
7. 读 OUTY\_H
8. 读 OUTZ\_L
9. 读 OUTZ\_H
10. 数据处理
11. 调到步骤 1

第 3 步中进行的检查可了解读取速率是否适合数据生产速率。如果一个或多个加速度样本已被新数据覆盖，由于读取速率较慢，STATUS 寄存器的 ZYXOR 位会置 1。

to 1.

如果输出寄存器内存在的所有数据均已被读取，同时尚未生成新数据，上溢位会自动清零。

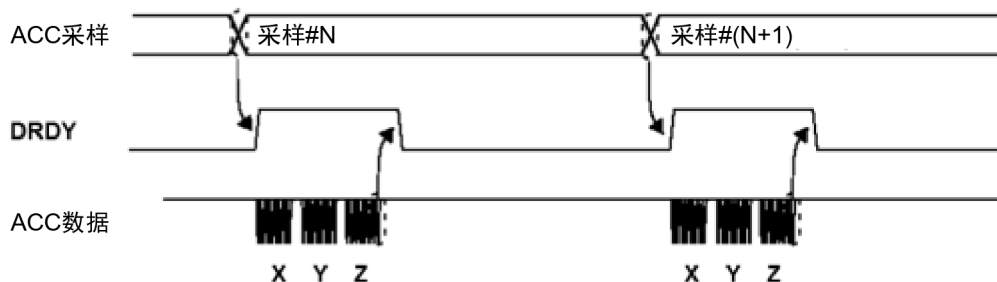
#### 3.1.2 使用数据就绪（DRDY）信号

该器件可配置为具有一个 HW 信号，以确定新的一组测量数据何时可以读取。此信号与 STATUS 寄存器的 XYZDA 位相关。通过将 CTRL3 寄存器中的 INT1\_DRDY 位设为 1，该信号可驱动至 INT1 引脚，或通过将 CTRL6 寄存器中的 INT2\_DRDY 位设为 1，该信号可驱动至 INT2 引脚。通过 CTRL5 寄存器的 H\_LACTIVE 位可将中断极性设置为低电平有效或高电平有效。

新的加速度数据集已生成并可供读取时，数据就绪信号会上升为 1。所有已使能通道的数据高位部分（29h、2Bh、2Dh）均读取完毕后，中断会复位。如果长时间未读取输出寄存器，则必须执行虚拟读取以清除数据准备就绪信号。



图 2. 数据准备就绪信号



### 3.1.3 使用块数据更新 (block data update, BDU) 功能

如果加速度数据的读取速度特别慢并且不能通过 STATUS 寄存器中存在的 XYZDA 位或通过 DRDY 信号进行同步 (或者不需要同步), 则强烈建议将 CTRL1 寄存器中的 BDU (块数据更新) 位置 1。

此功能可避免读取与其它样本相关联的加速度数据的最高有效部分和最低有效部分。特别是在 BDU 被激活的情况下, 与每条通道相关联的数据寄存器始终会包含由器件生成的最新加速度数据, 但如果开始了对给定寄存器对 (即 OUT\_X\_H 和 OUT\_X\_L、OUT\_Y\_H 和 OUT\_Y\_L、OUT\_Z\_H 和 OUT\_Z\_L) 的读取, 读取数据的 MSB 和 LSB 部分之前, 都会禁止刷新该寄存器对。

*注意: BDU 仅会确保已同时对 OUT\_X(Y, Z)\_L 和 OUT\_X(Y, Z)\_H 进行采样。但是, 如果读取速度过慢, 可能会读取在 T1 采样的 X、在 T2 采样的 Y, 以及在 T3 采样的 Z。*

## 3.2 了解加速度数据

测得的加速度数据会发送至 OUT\_X\_H、OUT\_X\_L、OUT\_Y\_H、OUT\_Y\_L、OUT\_Z\_H 和 OUT\_Z\_L 寄存器。这些寄存器分别包含作用于 X、Y 和 Z 轴的加速度信号的最高有效部分和最低有效部分。

X (Y、Z) 通道的完整加速度数据是由 OUT\_X\_H & OUT\_X\_L (OUT\_Y\_H & OUT\_Y\_L、OUT\_Z\_H & OUT\_Z\_L) 共同提供的, 表示为 2 的补码。

### 3.2.1 数据对齐

加速度数据表示为 16 位 2 的补码, 向左对齐。

### 3.2.2 加速度数据示例

表 10. 输出数据寄存器内容与加速度对比 (FS = 2 g, 灵敏度 0.061 mg/digit) 提供的几个基本示例中, 会在器件受给定加速度影响的情况下读取数据寄存器中的数据。本表中所列值是在理想器件校准的假设下给出的 (即, 无偏移, 无增益误差, ……)。

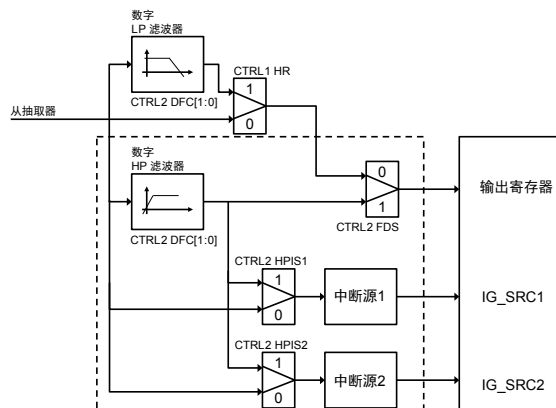
表 10. 输出数据寄存器内容与加速度对比 (FS = 2 g, 灵敏度 0.061 mg/digit)

加速度输出	寄存器地址	
	OUT_X_H (29h)	OUT_X_L (28h)
0 g	00h	00h
350 mg	16	69
1 g	40h	09h
-350 mg	E9	97
-1 g	BFh	F7h

## 4 高通滤波器

LIS2HH12 提供的嵌入式高通滤波功能可轻松去除测得加速度的 DC 分量。如图 3. 高通滤波器连接框图中所示，通过配置 CTRL2 寄存器的 FDS、HPIS[2:1]位，可以将滤波器独立应用于输出数据和/或中断数据。这意味着可以在中断生成作用于未滤波数据的同时获得已进行滤波的数据，反之亦然。

图 3. 高通滤波器连接框图



### 4.1 滤波器配置

参照表 11. 高通滤波器模式配置，高通滤波器可具有两种工作模式。

表 11. 高通滤波器模式配置

HPM1	HPM0	高通滤波器模式
0	0	正常模式
0	1	滤波参考信号
1	0	正常模式
1	1	N. A.

高通滤波器的带宽取决于所选 ODR 以及 CTRL2 寄存器的 DCF[1:0]位的设置。高通滤波器截止频率 ( $f_t$ ) 如表 12. 高通滤波器截止频率 [Hz] 中所示。

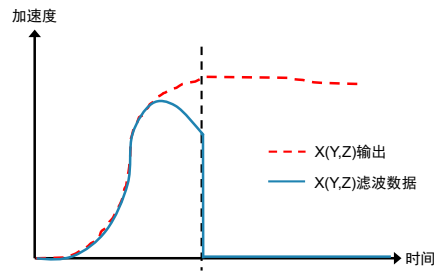
表 12. 高通滤波器截止频率 [Hz]

CTRL2 DCF [1:0]	HP 截止频率 [Hz]
00	ODR/50
01	ODR/100
10	ODR/9
11	ODR/400

#### 4.1.1 正常模式

在该配置中，可通过读取其中一个相关参考寄存器（XL\_REFERENCE 或 XH\_REFERENCE、YL\_REFERENCE 或 YH\_REFERENCE、ZL\_REFERENCE 或 ZH\_REFERENCE）为每个轴重置高通滤波器。复位操作会立即删除加速度的 DC 分量，并将相应的轴置于零。

**图 4. 读取 X(Y,Z)L\_REFERENCE 或 X(Y,Z)H\_REFERENCE**



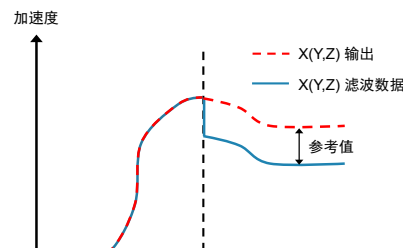
#### 4.1.2 滤波参考信号

在该配置中，输出数据被计算为输入加速度（X、Y、Z）与相关参考寄存器：XL(H)\_REFERENCE、YL(H)\_REFERENCE、ZL(H)\_REFERENCE 内容之间的偏差。每个轴的参考值是允许定义 16 位值的两个 8 位寄存器的组合。该值表示为 2 的补码形式，该 16 位值的 1 LSB 的值取决于所选满量程： $LSB = (FS)/(2^{15})$ （表 13. 参考模式 LSB 值）。

**表 13. 参考模式 LSB 值**

满量程	参考模式 LSB 值(mg)
2	~0.061
4	~0.122
8	~0.244

**图 5. 参考模式**



当传感器方向未知时，可在静态条件下使用参考模式。此模式允许在不应用任何滤波器架构的情况下移除传感器输出上存在的 DC 分量。该方法适用于需要删除 DC 分量，并受低于 HP 滤波器截止频率的加速度变化（缓慢运动）影响的应用。用作参考的程序应如下：

1. 将设备放置在指定位置
2. 读取 OUTX\_L(H)、OUTY\_L(H)和 OUTZ\_L(H)

3. 设置 X(Y, Z)L(H)\_REFERENCE = OUTX(Y, Z)\_L(H)
4. 启用参考模式: CTRL2 HPM[1:0] = "01"
5. 将参考数据驱动到输出寄存器: CTRL2 [FDS] = 1

## 5 中断生成

LIS2HH 集成了两个独立的中断发生器（IG1 和 IG2），可发出自由落体、唤醒、6D 和 4D 方向检测中断信号。这些信号可以独立驱动到两个中断引脚（INT1 和 INT2），或通过读取 IG\_SRC1(2)的 IA 位来检查。

### 5.1 中断引脚配置

器件提供的两个引脚可激活为生成数据就绪信号或中断信号。这些引脚功能是通过 CTRL3(22h)和 CTRL6(25h)选择的。请参考表 14. CTRL3 寄存器、表 15. CTRL3 描述、表 16. CTRL6 寄存器、表 17. CTRL6 描述以及图 6. 中断信号和中断引脚中提供的框图。

**表 14. CTRL3 寄存器**

-	-	INT1 _INACT	INT1 _IG2	INT1 _IG1	INT1 _OVR	INT1 _FTH	INT1 _DRDY
---	---	----------------	--------------	--------------	--------------	--------------	---------------

**表 15. CTRL3 描述**

INT1_INACT	INT1 上非活动中断。默认值 0。（0：禁用；1：启用）
INT1_IG2	INT1 上的中断发生器 2。默认值 0。（0：禁用；1：启用）
INT1_IG1	INT1 上的中断发生器 1。默认值 0。（0：禁用；1：启用）
INT1_OVR	INT1 上的 FIFO 上溢信号。默认值 0。（0：禁用；1：启用）
INT1_FTH	INT1 上的 FIFO 阈值信号。默认值 0。（0：禁用；1：启用）
INT1_DRDY	INT1 上的数据准备就绪信号。默认值 0。（0：禁用；1：启用）

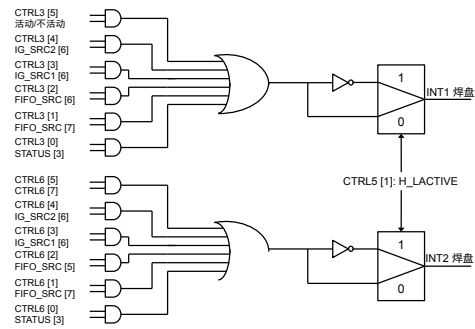
**表 16. CTRL6 寄存器**

BOOT	-	INT2 _BOOT	INT2 _IG2	INT2 _IG1	INT2 _EMPTY	INT2 _FTH	INT2 _DRDY
------	---	---------------	--------------	--------------	----------------	--------------	---------------

**表 17. CTRL6 描述**

INT2_BOOT	INT2 上的 BOOT 中断。默认值 0。（0：禁用；1：启用）
INT2_IG2	INT2 上的中断发生器 2。默认值 0。（0：禁用；1：启用）
INT2_IG1	INT2 上的中断发生器 1。默认值 0。（0：禁用；1：启用）
INT2_EMPTY	INT2 上的 FIFO 空标志。默认值 0。（0：禁用；1：启用）
INT2_FTH	INT2 上 FIFO 阈值信号。默认值 0。（0：禁用；1：启用）
INT2_DRDY	INT2 上的数据准备就绪信号。默认值 0。（0：禁用；1：启用）

图 6. 中断信号和中断引脚



如果多个中断信号发送到同一个引脚上 (INTx)，则该引脚的逻辑电平为所选中断信号组合的“或”。要识别出是什么信号产生了中断条件，则须读取相关源寄存器：IG\_SRC1(2)、FIFO\_SRC、STATUS。

## 6 惯性中断

LIS2HH12 可提供两个惯性中断信号，并可通过多种方式对这两个信号进行定制化。中断生成行为涉及的寄存器有 IG\_CFG1(2)、IG\_SRC1(2)、IG\_THS\_X(Y,Z)1、IG\_THS2 和 IG\_DUR1(2)。

可通过 IG\_CFG1(2) 寄存器配置中断功能，两个 MSB 用于配置操作模式，而六个 LSB 用于定义相对于所定义阈值的轴行为。

**表 18. IG\_CFG1(2) 寄存器**

AOI	6D	ZHIE	ZLIE	YHIE	YLIE	XHIE	XLIE
-----	----	------	------	------	------	------	------

**表 19. 中断模式配置**

AOI 位	6D 位	中断模式
0	0	轴请求的“或”组合
0	1	6 方向运动识别
1	0	轴请求的“与”组合
1	1	6 方向位置识别

**表 20. 中断行为配置**

位	中断行为
ZHIE	当 Z 输出高于阈值时使能中断请求。 默认值：0 (0: 禁用中断请求; 1: 使能中断请求)
ZLIE	当 Z 输出低于阈值时使能中断请求。 默认值：0 (0: 禁用中断请求; 1: 使能中断请求)
YHIE	当 Y 输出高于阈值时使能中断请求。 默认值：0 (0: 禁用中断请求; 1: 使能中断请求)
YLIE	Y 输出低于阈值时使能中断请求。 默认值：0 (0: 禁用中断请求; 1: 使能中断请求)
XHIE	X 输出高于阈值时使能中断请求。 默认值：0 (0: 禁用中断请求; 1: 使能中断请求)
XLIE	X 输出低于阈值时使能中断请求。 默认值：0 (0: 禁用中断请求; 1: 使能中断请求)

必须根据目标应用（自由落体、唤醒等）选择 X(Y,Z)HIE 和 X(Y,Z)LIE 配置。请注意，使能相同轴的 X(Y,Z)HIE 和 X(Y,Z)LIE 时，始终满足该轴的中断请求。来自所有三个轴的中断请求根据所选模式进行组合，以产生中断条件。满足中断条件时，会生成中断信号，通过读取 IG\_SRC1(2) 寄存器，可以了解哪个轴产生了请求。

**表 21. IG\_SRC1(2) 寄存器**

-	IA	ZH	ZL	YH	YL	XH	XL
---	----	----	----	----	----	----	----

表 22. IG\_SRC1(2)描述

IA	中断有效。默认值：0（0：未生成中断；1：已生成一个或多个中断请求）
ZH	Z 高。默认值：0（0：无请求；1：ZH 中断请求已发生）
ZL	Z 低。默认值：0（0：无请求；1：ZL 中断请求已发生）
YH	Y 高。默认值：0（0：无请求；1：YH 中断请求已发生）
YL	Y 低。默认值：0（0：无请求；1：YL 中断请求已发生）
XH	X 高。默认值：0（0：无请求；1：XH 中断请求已发生）
XL	X 低。默认值：0（0：无请求；1：XL 中断请求已发生）

## 6.1 持续时间

IG\_DUR1(2)寄存器的内容会设置要识别的中断时间的最短持续时间。持续时间也可以表示为在生成中断信号之前必须满足中断条件的连续样本的数量。在满足中断条件时，持续时间计数器开始递增，仅在达到 IG\_DUR1(2)内容值时允许产生中断；步骤和最大时间值取决于所选 ODR。在正常情况下，只要不再满足中断条件，就将持续时间计数器设为零，除非启用递减功能。

持续时间的测量公式为  $N/ODR$  [s]，其中 N 为持续时间寄存器内容，ODR 可假定为以下值：10、50、100、200、400 和 800 Hz。

表 23. 正常模式下的持续时间 LSB 值

ODR (Hz)	持续时间 LSB 值 (ms)
10	100
50	20
100	10
200	5
400	2.5
800	1.25

持续时间计数器允许在出现低于所选阈值的故障时防止中断生成。

### 6.1.1 递减函数

通过将 CTRL7(26h)寄存器中的 DCRM1(2)位设为“1”来使能递减功能。当该功能使能时，只要不再满足中断条件，就将中断引脚设为无效状态，但不会立即将持续时间计数器设为零：从 IG\_DUR1(2)寄存器值开始递减。如果在持续时间计数器达到零之前满足新的中断条件，则将再次从实际值开始增加，并在持续时间计数器再次达到 IG\_DUR1(2)寄存器值时产生新的中断。

此功能可在连续中断条件下提供更高的响应速度。



图 7. DCRM1(2) = 0

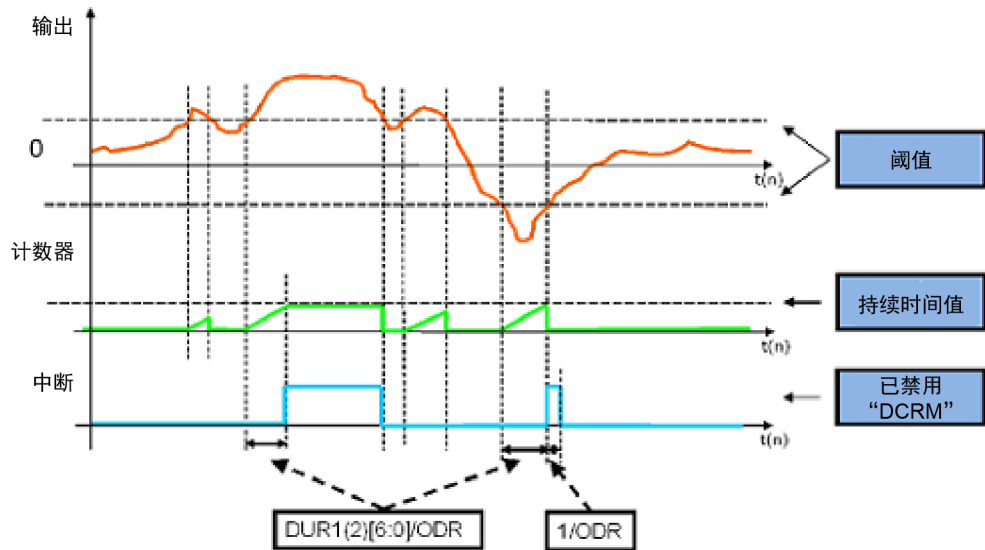
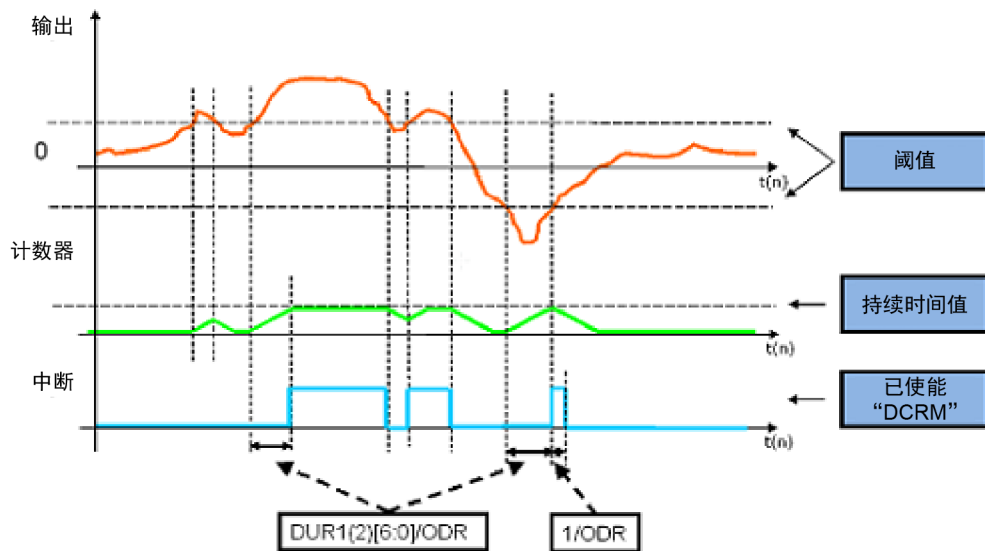


图 8. DCRM1(2) = 1

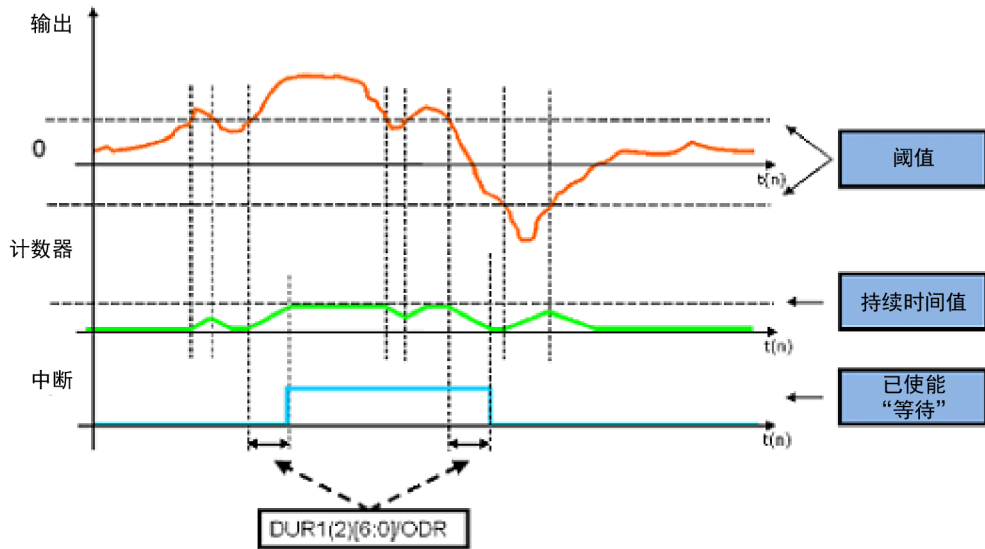


### 6.1.2 等待函数

通过将 `IG_DUR1(2)` 寄存器中的 `WAIT1(2)` 位设为“1”来使能等待功能。当不再满足中断条件时，持续时间计数器按每个样本递减 1。中断引脚信号保持有效，直到持续时间计数器不为零（至少：`DUR1(2)[6:0] / ODR [时间]`）。如果满足新的中断条件，并且持续时间计数器大于零，则持续时间计数器再次开始递增，并且中断引脚信号保持有效状态。

等待功能允许在出现低于所选阈值的故障时保持中断。

图 9. WAIT1(2) = 1



## 6.2 阈值

阈值寄存器定义了中断生成电路使用的参考加速度。LIS2HH12 中断发生器#1 允许为每个轴选择不同的阈值，而中断发生器#2 支持 X、Y 和 Z 轴的不同阈值等级。

对于中断发生器#1，用于配置该参数的寄存器名为 IG\_THS\_X(Y,Z)1，而对于中断发生器#2 则为 IG\_THS2，阈值 LSB 值与所选的满量程相关，并且等于： $FS/(2^8)$ （表 24. 阈值 LSB 值）。

表 24. 阈值 LSB 值

满量程	阈值 LSB 值(mg)
2	~8
4	~16
8	~31

## 6.3 自由落体和唤醒中断

LIS2HH12 中断信号可当做自由落体和唤醒中断。所有三个器件输出都被驱动到阈值比较器中，但仅评估启用的条件。IG\_CFG1(2)寄存器的 ZHIE、ZLIE、YHIE、YLIE、XHIE 和 XLIE 位可决定必须对哪一条轴执行中断决策、并可决定必须沿哪一方向传递阈值才能生成中断请求。只要满足选定中断条件，就会产生中断信号。必须读取 IG\_SRC1(2)寄存器，以了解发生了哪种情况。

对于中断发生器#1，自由落体信号(FF)和唤醒信号(WU)中断生成块在图 10. 自由落体、唤醒中断发生器#1 框图中表示，对于中断发生器#2，在图 11. 自由落体、唤醒中断发生器#2 框图中表示。

FF 或 WU 中断生成可通过 IG\_CFG1(2)寄存器中的 AOI 位选择。当 AOI 位被置为“0”时，选定的中断条件以逻辑“或”组合。在这种情况下，当至少一个使能轴超过写入 IG\_THS\_X(Y,Z)1（中断发生器#1）和 IG\_THS2（中断发生器#2）的阈值时，将产生中断。否则，当 AOI 位被设为“1”时，选定的中断信号被置于逻辑与的输入中，当所有使能轴超过阈值时，会产生中断信号。

CTRL7 寄存器的 LIR1(2)位决定是否必须锁存中断请求。如果 LIR1(2)位置为“0”，当中断条件满足时，中断信号会变为高电平，如果中断条件不再满足，中断信号会立即恢复低电平。否则，如果 LIR1(2)位置为“1”，如果中断条件适用，即使条件恢复为非中断状态，中断信号也会保持高电平，直至对 IG\_SRC1(2)寄存器执行读取操作。

必须使用 CTRL5 寄存器的 H\_LACTIVE 位来选择中断引脚极性。如果该位置为 0（默认值），则中断引脚为高电平激活，当检测到相关中断条件时，这些引脚从低电平变为高电平。否则，如果 H\_LACTIVE 位置为 1（低电平激活），则中断引脚正常为高电平，当达到中断条件时，从高电平变为低电平。

CTRL5 的 PP\_OD 位允许将中断引脚性质从推挽更改为开漏。如果 PP\_OD 位置为 0，则中断引脚处于推挽配置：对于高电平和低电平均为低阻抗输出。当 PP\_OD 位置为“1”时，只有中断活动状态是低阻抗输出。

图 10. 自由落体、唤醒中断发生器#1 框图

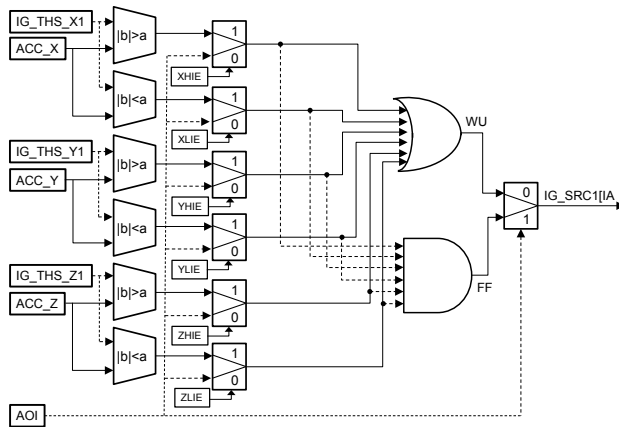
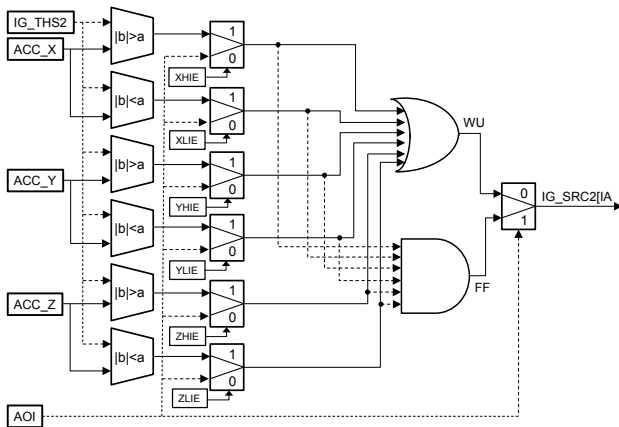


图 11. 自由落体、唤醒中断发生器#2 框图



系统检测任何自由落体或惯性唤醒事件所使用的阈值模块是由 IG\_THS\_X(Y,Z)1 和 IG\_THS2 寄存器定义的。阈值表示为 8 位无符号数字，并且绕零重力水平对称。

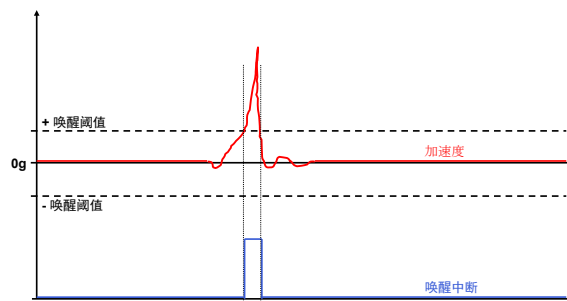
如果 X (Y, Z) 通道的无符号加速度值大于 IG\_THS\_X(Y,Z)1 (中断发生器#1) 和 IG\_THS2 (中断发生器#2)，IG\_SRC1(2) 寄存器的 XH(YH, ZH) 位为真。同理，如果 X(Y, Z) 通道的无符号加速度值低于所选阈值，XL(YL, ZL) 位为真。更多详细信息，请参见图 12. FF\_WU\_CFG 高电平和低电平。

图 12. FF\_WU\_CFG 高电平和低电平

### 6.3.1 惯性唤醒

唤醒中断是指 IG\_SRC1(2) 寄存器的特定配置，该配置允许在一个已配置轴的加速度超过定义的阈值(图 13. 惯性唤醒中断)时生成中断。

图 13. 惯性唤醒中断



### 6.3.2 不使用高通滤波器

本段介绍的基本算法演示了惯性唤醒功能在器件水平时的实际应用。下列代码会将器件配置为可识别出沿 X 或 Y 轴方向的绝对加速度超过预设阈值（本例中使用的阈值为 250 mg）的情况。触发中断的事件会锁存在设备内，会使用 INT1 引脚指示发生该事件。

```

0   - // 将器件置于水平位置
1.   写入 CTRL1 = 3Fh // X、Y、Z 使能，ODR = 100 Hz，BDU
    // 使能
2.   写入 CTRL2 = 00h // 关闭高通滤波器
3.   写入 CTRL3 = 08h // INT1 引脚上的中断发生器 1
    // FS = 2g，寄存器地址在
4.   写入 CTRL4 = 04h // 通过串行接口进行多字节访问期间
    // 自动递增
5.   写入 CTRL5 = 00h // 中断高电平有效；中断引脚推挽
    // 配置
6.   写入 CTRL7 = 04h // 中断 1 已锁存
7.   写入 IG_THS_X(Y)1 = 20h // 阈值 = 250 mg [ (2/256)*32 = 250 mg]
8.   写入 IG_DUR1 = 00h // 无持续
9.   写入 IG_CFG1 = 0Ah // 使能 XHIE 和 YHIE 中断生成
10.  如果(INT1 pin == 1)，转至 11，
    否则转至 10 // 等待唤醒事件
11.  读取 IG_SRC1 // 返回触发了中断的事件
    // 并清除中断引脚（锁存）
12.  （发生了唤醒事件；在此插入您的代码） // 事件处理
13.  进入 10

```

### 6.3.3 使用高通滤波器

以下代码中的基本例程展示了对已进行高通滤波的数据执行的惯性唤醒功能的实际应用。器件被配置为可识别出施加到 X、Y 或 Z 轴的加速度的高频分量超过预设阈值（本例中使用的阈值为 250 mg）的情况。触发中断的事件会锁存在设备内，会使用 INT1 引脚指示发生该事件。

```

0   - // 将设备放在任何方向

```

```

1.  写入 CTRL1 = 3Fh           // X、Y、Z 使能，ODR = 100 Hz，BDU
                                   // 使能
2.  写入 CTRL2 = 02h           // 为中断发生器 1 使能 HP 滤波器
3.  写入 CTRL3 = 08h           // INT1 引脚上的中断发生器 1
                                   // FS = 2g，寄存器地址在
4.  写入 CTRL4 = 04h           // 通过串行接口进行多字节访问期间
                                   // 自动递增
5.  写入 CTRL5 = 00h           // 中断高电平有效；中断引脚推挽
                                   // 配置
6.  写入 CTRL7 = 04h           // 中断 1 已锁存
7.  写入 IG_THS_X(Y,Z)1 = 20h // 阈值 = 250 mg [ (2/256)*32 = 250 mg]
8.  写入 IG_DUR1 = 00h         // 无持续
                                   // 进行虚拟读取，以立即将 HP 滤波器输出强制为
9.  读取寄存器：X(Y,Z)L_REFERENCE 或 // 当前加速度值。
    X(Y,Z)H_REFERENCE           // 允许保存滤波器建立时间
10. 写入 IG_CFG1 = 2Ah         // 使能 ZHIE、YHIE 和 XHIE 中断生成
11. 如果(INT1 pin == 1)，转至 12， // 等待唤醒事件
    否则转至 11
12. 读取 IG_SRC1               // 返回触发了中断的事件
                                   // 并清除中断引脚（锁存）
13. （发生了唤醒事件；在此插入您的代码） // 事件处理
14. 进入 11

```

在第 9 步中立即对 X(Y,Z)L\_REFERENCE 或 X(Y,Z)H\_REFERENCE 寄存器进行虚拟读取，以将 HP 滤波器输出立即强制为器件执行阈值比较时所参照的当前加速度/倾斜状态。

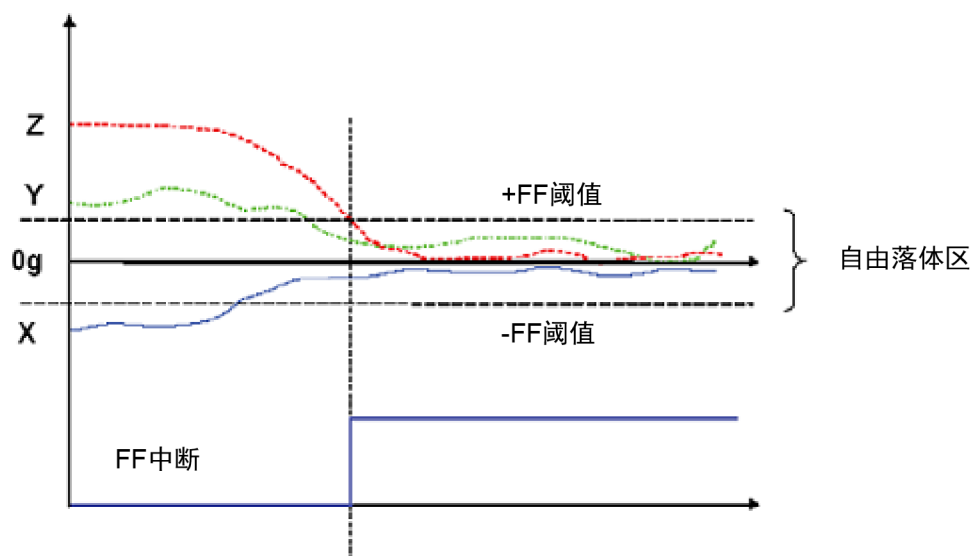
该读取可在需要删除加速度 DC 分量的任何时候执行，而无需等待滤波器的建立时间。

HP 滤波器用于器件方向优先级未知，并且必须删除 X、Y 和 Z 上的加速度重力分量时。

## 6.4 自由落体检测

自由落体检测涉及 IG\_SRC1(2)寄存器的特定配置，可以识别器件何时处于自由落体：沿各轴所测得的加速度均为 0。在实际情况下“自由落体区”定义为零重力水平附近，在该区域中，所有加速度都足够小，可生成中断(图 14. 自由落体中断)。

图 14. 自由落体中断



本段介绍了使用自由落体检测的基础知识。还特别介绍了以下将器件配置为检测自由落体事件并发信号指示此类事件的软件例程：

- |     |                                     |  |
|-----|-------------------------------------|--|
| 1.  | 写入 CTRL1 = 3Fh                      | // X、Y、Z 使能，ODR = 100 Hz，BDU<br>// 使能              |
| 2.  | 写入 CTRL2 = 00h                      | // 关闭高通滤波器   |
| 3.  | 写入 CTRL3 = 08h                      | // INT1 引脚上的中断发生器 1                                |
| 4.  | 写入 CTRL4 = 04h                      | // FS = 2g，寄存器地址在<br>// 通过串行接口进行多字节访问期间<br>// 自动递增 |
| 5.  | 写入 CTRL5 = 00h                      | // 中断高电平有效；中断引脚推挽<br>// 配置                         |
| 6.  | 写入 CTRL7 = 04h                      | // 中断 1 已锁存  |
| 7.  | 写入 IG_THS_X(Y)1 = 2Dh               | // 阈值 = 352 mg [ (2/256)*45 = 352 mg]              |
| 8.  | 写入 IG_DUR1 = 03h                    | // 设置三个样本事件持续时间                                    |
| 9.  | 写入 IG_CFG1 = 95h                    | // 为 XLIE、YLIE 和 ZLIE：自由落体识别<br>// 启用逻辑与比较         |
| 10. | 如果(INT1 pin == 1)，转至 11，<br>否则转至 10 | // 等待自由落体事件  |
| 11. | 读取 IG_SRC1                          | // 返回触发了中断的事件<br>// 并清除中断引脚（锁存）                    |
| 12. | （发生了自由落体事件；在此插入您的代码）                | // 事件处理  |
| 13. | 进入 10                               |  |

示例代码中利用设置为 352 mg 的阈值，用于自由落体识别，该事件由硬件通过 INT1 引脚进行通知。在第 8 步，IG\_DUR1 寄存器像这样进行了配置，从而可忽略短于  $3/ODR = 3/100 \text{ Hz} \approx 30 \text{ ms}$  的事件，以避免假检测现象的发生。

发生自由落体事件后，对 IG\_SRC1 寄存器的读取操作会清空请求，器件会准备好识别其他自由落体事件。

## 7 6D/4D 定向检测

LIS2HH12 的高级功能可检测器件在空间中的方向，从而可轻松为手持设备实施节能程序，并可实现图像自动旋转。

### 7.1 6D 定向检测

当器件的一面在已知方向稳定时（几乎水平），允许产生中断信号：可以检测到器件在空间中的六个方向。6D 定向功能可通过 IG\_CFG1(2)寄存器的 AOI 和 6D 位使能，具体来讲，必须将 ZHIE、ZLIE、YHIE、YLIE、XHIE 和 XLIE 设为“1”，以便在所有六个可能的方向产生中断。当只有一个轴超过所选阈值并且通过其他两个轴测量的加速度低于阈值时，6D 中断被置位，当多个轴测量的加速度超过所选阈值时，中断被取消置位。

当为 6D 功能配置时，IG\_SRC1(2)的 ZH、ZL、YH、YL、XH 和 XL 位提供有关设备方向的信息，指示哪个方向几乎水平以及朝向（即朝上或朝下）。当使能 6D 方向时：ZH 或 ZL 或 YH 或 YL 或 XH 或 XL 时，只能将 IG\_SRC1(2)的六个 LSB 的其中一个设为“1”。

更具体地说：

- 当垂直于 Z(Y,X)轴的面几乎是平面，Z(Y,X)轴上测得的加速度为正且模块中的加速度大于阈值时，ZH(YH, XH)等于“1”。
- 当垂直于 Z(Y,X)轴的面几乎是平面，Z(Y,X)轴上测得的加速度为负且模块中的加速度大于阈值时，ZL(YL, XL)等于“1”。

6D 功能有两种可行的配置：

- 6D 运动识别：在该配置下，当器件从一个方向（已知或未知方向）移动到另一已知方向时，会生成中断。中断信号仅对 1/ODR [s]有效，然后自动解除置位。通过在 IG\_CFG1(2)寄存器中设置 AOI 位 = 0 和 6D 位 = 1 来激活该功能。为在全部六个方向上产生 6D 移动中断，必须将 IG\_CFG1(2)寄存器设为 0x7F。
- 6D 位置识别：在该配置中，器件在六个已知位置之一稳定时产生中断；只要维持位置不变，中断就会保持有效。通过在 IG\_CFG1(2)寄存器中设置 AOI 位 = 1 和
- 6D-bit = 1 来激活该功能。为在全部六个位置产生中断，必须将 IG\_CFG1(2)寄存器设为 0xFF。

参考图 15. 6D 运动与 6D 位置中断对比，6D 运动行显示了当器件配置为在 X 和 Y 正轴上实现 6D 运动识别 (IG\_CFG1(2) = 0x4A)时中断的行为，而 6D 位置行则显示了当器件配置为在 X 和 Y 正轴上实现 6D 位置识别 (IG\_CFG1(2) = 0xCAh)时中断的行为。

参考图 16. 6D 识别位置，器件配置为在 X、Y 和 Z 轴上实现 6D 位置功能。表 25. 6D 位置中的 IG\_SRC1(2)寄存器显示的是每个位置的 IG\_SRC1(2)寄存器的内容。

图 15. 6D 运动与 6D 位置中断对比

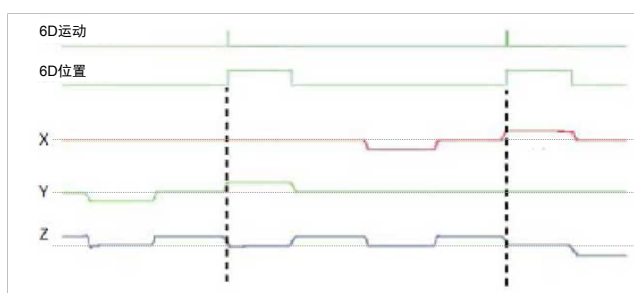


图 16. 6D 识别位置

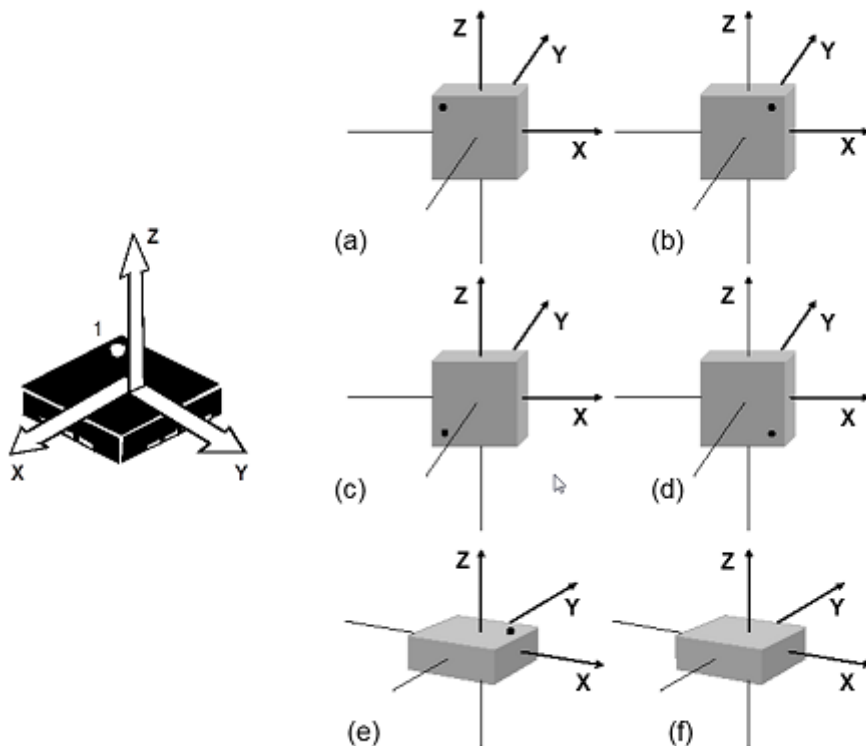


表 25. 6D 位置中的 IG\_SRC1(2)寄存器

用例	IA	ZH	ZL	YH	YL	XH	XL
(a)	1	0	0	0	0	0	1
(b)	1	0	0	0	1	0	0
(c)	1	0	0	1	0	0	0
(d)	1	0	0	0	0	1	0
(e)	1	1	0	0	0	0	0
(f)	1	0	1	0	0	0	0

## 7.2 4D 方向

当设备几乎水平时，也可以识别 X 和 Y 上的纵向和横向条件。启用 4D 方向时，纵向和横向计算不考虑 Z 输出：IG\_SRC1(2)的 YH、YL、XH、XL 位与 Z 轴无关。

当 IG\_CFG1(2)的 6D 位也设为 1 时，可通过将 CTRL7 的 4D\_IG1(2)位置 1 来启用此模式。选择 4D 功能时，必须将 IG\_CFG1(2)寄存器的 ZHIE 和 ZLIE 位设为零，否则可能会产生不需要的中断。这种模式下，Z 轴位置检测被禁用，因此位置识别减少为表 25. 6D 位置中的 IG\_SRC1(2)寄存器的 (a)、(b)、(c) 和 (d) 的情形。

1.	写入 CTRL1 = 3Fh	// X、Y、Z 使能，ODR = 100 Hz，BDU // 使能
2.	写入 CTRL2 = 00h	// 关闭高通滤波器
3.	写入 CTRL3 = 08h	// INT1 引脚上的中断发生器 1



4.	写入 CTRL4 = 34h	// FS = 8g, 寄存器地址在 // 通过串行接口进行多字节访问期间 // 自动递增
5.	写入 CTRL5 = 00h	// 中断高电平有效; 中断引脚推挽 // 配置
6.	写入 CTRL7 = 01h	// 中断 1 未锁存, INT1 上的 4D
7.	写入 IG_THS_X(Y)1 = 0Ah	// X、Y 阈值 = 10 * 31 mg = 310 mg
8.	写入 IG_DUR1 = 00h	// 1°样本超过阈值会产生中断
9.	写入 IG_CFG1 = CFh	// 6D 位置, X 和 Y 大于和小于 // 阈值, ZHIE = ZLIE = 0

## 8 先进先出 (FIFO) 缓冲区

为了减少主控的干预并便于对事件识别数据进行后处理，LIS2HH12 为三条输出通道 X、Y 和 Z 分别嵌入了先进先出 (FIFO) 缓冲区。

由于主机处理器不需要连续轮询来自传感器的数据，FIFO 使用可以使系统保持连续的节约功耗，只在需要时唤醒并从 FIFO 批量输出重要数据。

FIFO 缓冲区可在六种不同模式下工作，各个模式可确保在应用开发过程中实现高度灵活性：**Bypass 模式**、**FIFO 模式**、**Stream 模式**、**Stream-to-FIFO 模式**、**Bypass-to-Stream 模式**和 **Bypass-to-FIFO 模式**。通过 FIFO\_CTRL 寄存器中的 FMODE[2:0]位选择每种模式。

可编程 FIFO 阈值等级、FIFO 空或 FIFO 上溢事件在 FIFO\_SRC 寄存器中可用，可通过设置在 INT1 和 INT2 引脚上产生专用中断。

### 8.1 FIFO 描述

FIFO 缓冲区最多能为每条通道存储 32 个 16 位加速度样本；数据采用左对齐 16 位 2 的补码形式存储。

数据样本集合由 6 个字节 (Xl、Xh、Yl、Yh、Zl 和 Zh) 和组成，它们会以选定的输出数据速率 (ODR) 释放到 FIFO 中。

新样本集合会放在第一个空闲的 FIFO 位置中，缓冲区被占满后，新样本集合会覆盖最早的值。

**表 26. FIFO 缓冲区填满示例 (存储第 32 个样本集)**

输出寄存器	0x28h	0x29h	0x2Ah	0x2Bh	0x2Ch	0x2Dh
	Xl (0)	Xh (0)	Yl (0)	Yh (0)	Zl (0)	Zh (0)
FIFO 索引	FIFO 样本集合					
FIFO (0)	Xl (0)	Xh (0)	Yl (0)	Yh (0)	Zl (0)	Zh (0)
FIFO (1)	Xl (1)	Xh (1)	Yl (1)	Yh (1)	Zl (1)	Zh (1)
FIFO (2)	Xl (2)	Xh (2)	Yl (2)	Yh (2)	Zl (2)	Zh (2)
FIFO (3)	Xl (3)	Xh (3)	Yl (3)	Yh (3)	Zl (3)	Zh (3)
...	...	...	...	...	...	...
...	...	...	...	...	...	...
FIFO (30)	Xl (30)	Xh (30)	Yl (30)	Yh (30)	Zl (30)	Zh (30)
FIFO (31)	Xl (31)	Xh (31)	Yl (31)	Yh (31)	Zl (31)	Zh (31)

**表 27. FIFO 溢出示例 (存储第 33 个样本集同时丢弃第 1 个样本)**

输出寄存器	0x28h	0x29h	0x2Ah	0x2Bh	0x2Ch	0x2Dh
	Xl (1)	Xh (1)	Yl (1)	Yh (1)	Zl (1)	Zh (1)
FIFO 索引	FIFO 样本集合					
FIFO (0)	Xl (1)	Xh (1)	Yl (1)	Yh (1)	Zl (1)	Zh (1)
FIFO (1)	Xl (2)	Xh (2)	Yl (2)	Yh (2)	Zl (2)	Zh (2)
FIFO (2)	Xl (3)	Xh (3)	Yl (3)	Yh (3)	Zl (3)	Zh (3)
FIFO (3)	Xl (4)	Xh (4)	Yl (4)	Yh (4)	Zl (4)	Zh (4)
...	...	...	...	...	...	...
...	...	...	...	...	...	...

输出 寄存器	0x28h	0x29h	0x2Ah	0x2Bh	0x2Ch	0x2Dh
	XI (1)	Xh (1)	YI (1)	Yh (1)	ZI (1)	Zh (1)
FIFO 索引	FIFO 样本集合					
FIFO (30)	XI (31)	Xh (31)	YI (31)	Yh (31)	ZI (31)	Zh (31)
FIFO (31)	XI (32)	Xh (32)	YI (32)	Yh (32)	ZI (32)	Zh (32)

表 26. FIFO 缓冲区填满示例（存储第 32 个样本集） 表 27. FIFO 溢出示例（存储第 33 个样本集同时丢弃第 1 个样本）表示的是存储了 32 个样本时 FIFO 已满的状态，而表示下一步，当第 33 个采样插入到 FIFO 中，同时第 1 个采样被覆盖。新的最早样本集合在输出寄存器中可用。

如果 FIFO 已使能，并且所处模式不是 Bypass 模式，LIS2HH12 输出寄存器（28h 到 2Dh）始终会包含最早的 FIFO 样本集合。

## 8.2 FIFO 寄存器

FIFO 缓冲区由三个不同的加速度计寄存器进行管理，其中两个寄存器可使能并配置 FIFO 特性，第三个寄存器会提供关于缓冲区状态的信息。

### 8.2.1 CTRL3 寄存器（0x22）

CTRL3 寄存器中的 FIFO\_EN 位必须设为 1 才能使能内部的先进先出缓冲区；如果该位置 1，加速度计输出寄存器（28h 到 2Dh）不会包含当前加速度计值，但会包含 FIFO 中存储的最早值。

表 28. CTRL3 寄存器

FIFO_EN	STOP_FTH						
---------	----------	--	--	--	--	--	--

STOP\_FTH 位用于激活 FIFO 阈值电平功能，使能该位时，物理 FIFO 尺寸将被限制为 FIFO\_CTRL 寄存器的 FTH[4:0]值。如果将 STOP\_FTH 位设为“1”，FTH 中断将开始作为 OVR 中断执行。

### 8.2.2 FIFO\_CTRL 寄存器（0x2E）

该寄存器专用于 FIFO 模式选择和阈值配置。

表 29. FIFO\_CTRL 寄存器

FMODE2	FMODE1	FMODE0	FTH4	FTH3	FTH2	FTH1	FTH0
--------	--------	--------	------	------	------	------	------

FMODE[2:0]位专用于定义 FIFO 缓冲区特性选择：

FMODE[2:0] = (0,0,0): Bypass 模式

FMODE[2:0] = (0,0,1): FIFO 模式

FMODE[2:0] = (0,1,0): Stream 模式

FMODE[2:0] = (0,1,1): Stream-to-FIFO 模式

FMODE[2:0] = (1,0,0): Bypass-to-Stream 模式

FMODE[2:0] = (1,0,1): 不使用

FMODE[2:0] = (1,1,0): 不使用

FMODE[2:0] = (1,1,1): Bypass-to-FIFO 模式

FTH[4:0]位用于定义阈值等级；当存储到 FIFO 中的样本数超过该值时，将 FIFO\_SRC 寄存器中的 FTH 位设为“1”。

如果将 CTRL3 寄存器的 STOP\_FTH 位设为“1”，则 FTH[4:0]值也可以定义 FIFO 物理深度，在这种情况下，FTH 中断开始作为 OVR 中断执行。

### 8.2.3 FIFO\_SRC 寄存器 (0x2F)

该只读寄存器每个 ODR 会更新一次，会提供关于 FIFO 缓冲区状态的信息。

**表 30. FIFO\_SRC 寄存器**

FTH	OVR	空	FSS4	FSS3	FSS2	FSS1	FSS0
-----	-----	---	------	------	------	------	------

当 FIFO 内容超过 FTH[4:0] 阈值等级时 (FSS > FTH), FTH 位被设为“1”。

当 FIFO 缓冲区已满时, OVR 位被设为“1”, 这意味着 FIFO 缓冲区包含 32 个未读取的新样本, 并且可以读取。下一 ODR 时, 新样本集会替换最早的 FIFO 值。第一个样本集合已被读取时, OVR 位会复位至“0”。

当已读取所有 FIFO 样本且缓冲区中没有未读样本时, EMPTY 标志被置为高电平。当 EMPTY 位等于“1”, 并且 FIFO 已有最后一个样本集合时, 其他读操作继续返回相同的值。

FSS[4:0] 字段始终包含 FIFO 缓冲区中存储的当前未读样本数。FIFO 使能后, 该值会以 ODR 频率增加, 直至缓冲区已满, 随后, 每次从 FIFO 读取一个样本集合, 该值都会减小。

FIFO\_SRC 寄存器内容会与 FIFO 写操作和读操作同步更新。

**表 31. FIFO\_SRC\_REG 启动特性 (假定 FTH[4:0] = 15)**

FTH	OVRN	空	FSS[4:0]	未读 FIFO 样本	时序
0	0	1	00000	0	t0
0	0	0	00000	1	t0 + 1/ODR
0	0	0	00001	2	t0 + 2/ODR
0	0	0	00010	3	t0 + 3/ODR
...	...	...	...	...	...
0	0	0	01110	15	t0 + 15/ODR
1	0	0	01111	16	t0 + 16/ODR
...	...	...	...	...	...
1	0	0	11110	31	t0 + 31/ODR
1	1	0	11111	32	t0 + 32/ODR
1	1	0	11111	32 (丢弃最早的数据)	t0 + 33/ODR

**表 32. FIFO\_SRC\_REG 运行特性 (假定 FTH[4:0] = 15)**

FTH	OVRN	空	FSS[4:0]	未读 FIFO 样本	时序
0	0	1	00000	(#1 读取样本)	t0
0	0	0	00001	2	t0 + 1/ODR
0	0	0	00010	3	t0 + 2/ODR
...	...	...	...	...	...
0	0	0	01110	15	t0 + 14/ODR
1	0	0	01111	16	t0 + 15/ODR
...	...	...	...	...	...
1	0	0	11110	31	t0 + 30/ODR
1	0	0	11111	32	t0 + 31/ODR

FTH	OVRN	空	FSS[4:0]	未读 FIFO 样本	时序
1	1	0	11111	32 (丢弃读取样本)	$t_0 + 32/ODR$

仅在启动时满足条件  $OVR = 0$  和  $FSS = 0$ ，因为 FIFO 实际为空，在运行模式中，最后一个读取样本仍保留在 FIFO 缓冲区中。

通过配置 CTRL3 寄存器中的 INT1\_OVR 和 INT1\_FTH 位以及 CTRL6 寄存器中的 INT2\_EMPTY 和 INT2\_FTH 位，可以使能 WTM、OVR 和 EMPTY 标志，以在 INT1(2)引脚上产生专用中断。

## 8.3 FIFO 模式

通过 FIFO\_CTRL 寄存器的 FMODE[2:0] 字段，LIS2HH12 FIFO 缓冲器可配置为六种不同的可选工作模式。可用配置确保了高度灵活性，并扩展了可用于应用开发的功能数量。

以下段落介绍了 Bypass 模式、FIFO 模式、Stream 模式、Stream-to-FIFO 模式、Bypass-to-Stream 模式和 Bypass-to-FIFO 模式。

CTRL3 寄存器中的 FIFO\_EN 位必须设为 1 才能使能内部的先进先出缓冲区。

### 8.3.1 Bypass 模式

Bypass 模式下 (FIFO\_CTRL (FMODE [2:0])= 000)，FIFO 不可操作，保持为空。

在 FIFO 模式下，Bypass 模式也用于重置 FIFO。

### 8.3.2 FIFO 模式

在 FIFO 模式中 (FIFO\_CTRL (FMODE [2:0])= 001)，X、Y 和 Z 通道的数据存储存储在 FIFO 中，直至存满。可通过设置 CTRL3 (INT1\_OVR) = "1" 使能上溢中断。发生上溢中断时，FIFO 已满并停止收集数据。要重置 FIFO 内容，必须在 FIFO\_CTRL (FMODE [2:0]) 中选择 Bypass 模式。在重置后，通过设置 FIFO\_CTRL (FMODE [2:0]) = 001，可重新启动 FIFO 模式。

FIFO 缓冲区可保存 32 级 X、Y 和 Z 数据，但可以使用 CTRL3 (STOP\_FTH) 位降低 FIFO 的深度。设置 STOP\_FTH bit = 1 时，FIFO 深度被限制为 FIFO\_CTRL (FTH [4:0])。

### 8.3.3 Stream 模式

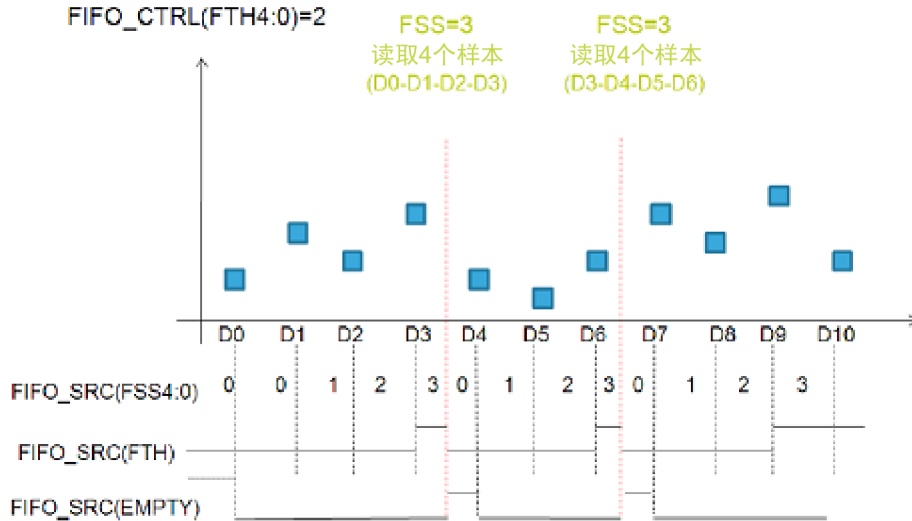
Stream 模式 (FIFO\_CTRL (FMODE [2:0]) = 010) 提供连续的 FIFO 更新；若 FIFO 已满，当新数据到来时，旧数据会被丢弃。

可以使能溢出中断，CTRL3 (INT1\_OVR) = "1"，以便立即读取全部 FIFO 内容。为了清空 FIFO，必须在产生溢出中断时读取 FIFO\_SRC (FSS [4:0]) + 1 个样本。

如果数据不能丢失并且无法在一个 ODR 周期内读取至少一个数据样本集，则可以使能水印中断，以部分读取 FIFO，并为传入数据留下空闲内存插槽。为了刷新 FIFO，必须在水印中断上升时读取 FIFO\_SRC (FSS [4:0]) 样本。在最后一种情况下，如果在水印中断上检索到 FIFO\_SRC (FSS [4:0]) + 1 个样本，则第一个读取样本将为上一个突发的最后一个样本 (图 17. Stream 模式中的 D3)。

图 17. Stream 模式

## Stream模式



Stream 模式用于在收到上溢信号后读取 ODR 中的全部 32 个 FIFO 样本。但是，也可以启用水印中断 CTRL3 (INT1\_FTH)、CTRL6 (INT2\_FTH)，以读取 FIFO 中的数据，并为传入数据留下空闲内存插槽。

### 8.3.4 Stream-to-FIFO 模式

在 Stream-to-FIFO 模式中(FIFO\_CTRL(FMODE2:0) = 011)，FIFO 特性根据 IG\_SRC1 (IA)位改变。当 IG\_SRC1(IA)位等于“1”时，FIFO 以 FIFO 模式运行，当 IG\_SRC1 (IA)位等于“0”时，FIFO 以 Stream 模式运行。必须通过 IG\_CFG1、IGTHS\_X1、IGTHS\_Y1 和 IGTHS\_Z1 寄存器正确设置中断发生器 1。

应将 CTRL7 (LIR1)位设为“1”，以锁存中断状态并保留 FIFO 内容。如果中断未锁存，并且中断条件消失，FIFO 特性将切换至 Stream 模式，以覆盖 FIFO 内容。

### 8.3.5 Bypass-to-Stream 模式

在 Bypass-to-Stream 模式中(FIFO\_CTRL (FMODE [2:0]) = 100)，FIFO 缓冲区复位

(Bypass 模式)，当 G\_SRC1 (IA)等于“0”时，开始收集数据，并在 G\_SRC1 (IA)等于“1”时，以 Stream 模式运行。

必须通过 IG\_CFG1、IGTHS\_X1、IGTHS\_Y1 和 IGTHS\_Z1 寄存器正确设置中断发生器 1。

应将 CTRL7 (LIR1)位设为“1”，以锁存中断状态并保留 FIFO 内容。如果中断未锁存，并且中断条件消失，则 FIFO 特性切换至 Bypass，并重置内容。

### 8.3.6 Bypass-FIFO 模式

在 Bypass-to-FIFO 模式(FIFO\_CTRL (FMODE [2:0]) = 111)中，FIFO 特性根据 IG\_SRC1(IA)位改变。当 IG\_SRC1(IA)位等于“1”时，FIFO 以 FIFO 模式运行，当 IG\_SRC1(IA)位等于“0”时，FIFO 以 Bypass 模式运行 (FIFO 内容复位)。如果产生中断，FIFO 开始收集数据，直至存满。

必须通过 IG\_CFG1、IGTHS\_X1、IGTHS\_Y1 和 IGTHS\_Z1 寄存器正确设置中断发生器 1。

应将 CTRL7 (LIR1)位设为“1”，以锁存中断状态并保留 FIFO 内容。如果中断未锁存，并且中断条件消失，则 FIFO 特性切换至 Bypass，并重置内容。

## 8.4 从 FIFO 读取数据

如果 FIFO 已使能，并且所处模式不是 Bypass 模式，读取输出寄存器 (28h 到 2Dh) 会返回最早的 FIFO 样本集合。

读取输出寄存器时，其内容会移至 SPI/I<sup>2</sup>C 输出缓冲区。理想地，FIFO 插槽会向上移动一格，以便释放空间接收新的采样，并且输出寄存器载入 FIFO 缓冲器中存储的实际最旧的值。

执行数据读取的最佳方法是在将 OVR 位设为“1”时清空 FIFO 缓冲区，但可以随时执行读取操作。要读取的 FIFO 样本数取决于读取方法：

- 如果 OVR 等于“1”，则必须读取#FSS + 1 个样本。空位必须变成“1”。
- 如果 OVR 等于“0”，则必须读取#FSS + 个样本。空位必须保持为“0”。

整个 FIFO 内容是通过通过对加速度计输出寄存器执行 32 次读取操作读取的，FIFO 缓冲区有新样本集合之前，所有其他的读取都会返回相同的最后值。

为了提高应用的灵活性，可通过读取任意字节组合从 FIFO 读取数据（例如：196 次单字节读取，32 次 6 字节读取，1 次 196 字节的多字节读取等）。

建议以快于 1 ODR 的速度通过 196 字节多字节读取的方式来读取所有 FIFO 位置（6 个输出寄存器乘以 32 个存储位置）。为了最大限度地减少主器件与从器件之间的通信，器件会自动更新读取地址；到达寄存器 0x2D 时，会回滚到 0x28。

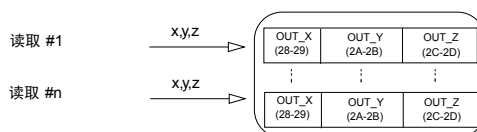
为了避免数据丢失，必须按照可用的串行通信速率选择正确的 ODR。如果使用标准 I<sup>2</sup>C 模式（最大速率 100 kHz），单次样本集合读取会用时 830 μs，而整体 FIFO 下载约用时

17.57 ms。I<sup>2</sup>C 速度低于 SPI，它需要大约 29 个时钟脉冲才能开始通信（开始、从地址，器件地址+写入、重新启动、器件地址+读取），并且每个字节读取都需要额外 9 个时钟脉冲。完整 FIFO 读取的执行速度会快于 1\*ODR，这意味着如果使用标准 I<sup>2</sup>C，可选 ODR 必须低于 57 Hz。如果使用快速 I<sup>2</sup>C 模式（最大速率 400 kHz），可选 ODR 必须低于 228 Hz。

#### 8.4.1 FIFO 多次读取（突发）

从地址 0x28 开始，可能执行多次读取。一旦读到地址 0x2D，系统将自动从地址 0x28 重新启动。

图 18. FIFO 多次读取



## 版本历史

表 33. 文档版本历史

日期	版本	变更
2015 年 4 月 13 日	1	初始版本。