

ST加速度计使用基础篇 二 --Free fall, wake up, 6D功能

[Log in](#) or [register](#) to post comments 热度： 416

给大家介绍下sensor的一些嵌入的中断功能，教大家如何实现wake up, free fall, 6D功能，这些功能本身都是由sensor实现的，所以主控只需要配置好sensor，然后等待中断的触发即可，并不需要主控的实时姿态运算，从而节省大量系统资源。这里还是以lis3dh为基础给大家介绍。**1. INT1中断**Lis3dh提供了一个INT1 (inertial interrupt)的信号处理中断，它包括下面三个寄存器。

INT1_CFG (30h)

Table 46. INT1_CFG register

AOI	6D	ZHIE/ ZUPE	ZLIE/ ZDOWNE	YHIE/ YUPE	YLIE/ YDOWNE	XHIE/ XUPE	XLIE/ XDOWNE
-----	----	---------------	-----------------	---------------	-----------------	---------------	-----------------

INT1_THS (32h)

Table 51. INT1_THS register

0	THS6	THS5	THS4	THS3	THS2	THS1	THS0
---	------	------	------	------	------	------	------

INT1_DURATION (33h)

Table 53. INT1_DURATION register

0	D6	D5	D4	D3	D2	D1	D0
---	----	----	----	----	----	----	----

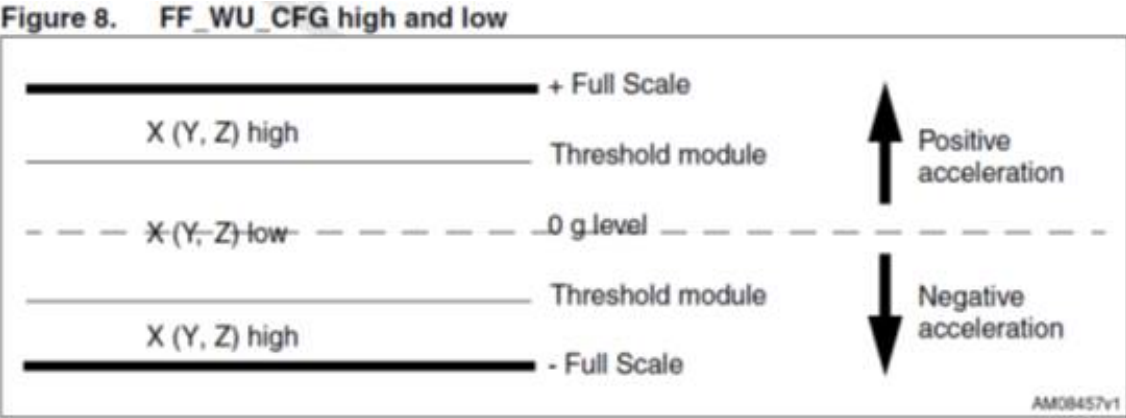
简单的说这个中断的功能就是检测sensor三个轴的加速度的值大于或小于设定阈值的信号，在寄存器INT_CFG里，你可以选择哪个轴的值大于或小于这个阈值，这里可以组合你要检测的轴，亦即方向（大于或小于）。INT1_THS就是设定的阈值，这个值是一个绝对的值，假如设定这个值是200mg, 然后enable XLIE，这意味你要检测到X轴的值小于200mg，大于-200mg的信号。如果enable XHIE, 这表示你要检测到x轴大于200mg，或小于-200mg的信号，INT1_DURATION 定义检测到设定信号后持续的时间，当达到这个时间后就可以触发中断，当然这里还要设置下如下寄存器的I1_AOI1 把实际的中断信号enable到硬件的中断管脚1。

CTRL_REG3 (22h)

Table 30. CTRL_REG3 register

I1_CLICK	I1_AOI1	I1_AOI2	I1_DRDY1	I1_DRDY2	I1_WTM	I1_OVERRUN	--
----------	---------	---------	----------	----------	--------	------------	----

信号检测的模型：



2. 参数设置 1) THS设置 首先阈值的设置是和sensor设定的测量范围是相关的，如下表，如sensor的测量范围设为2g时， 阈值寄存器的每个逻辑数对应的是16mg。举个例子，我们要设定一个256mg的 wakeup的阈值，我们就应该在INT1_THS写入16。因为16mg*16=256mg

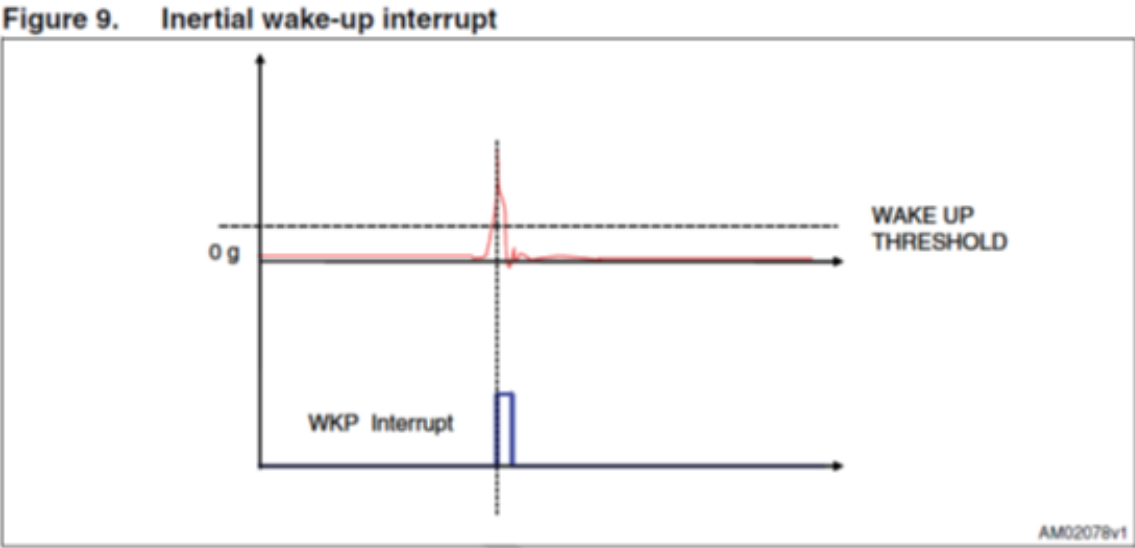
Table 16. Threshold LSB value

Full scale	Threshold LSB value (mg)
2	~16
4	~31
8	~63
16	~125

2) Duration设置 Duration时间的设置是与sensor的ODR(output datarate)相关的，如下表，所以 duration= 1/ODR * LSB， LSB写入寄存器的值。

ODR (Hz)	Duration LSB value (ms)
1	1000
10	100
25	40
50	20
100	10
200	5
400	2.5
1000	1
1344	0.744
1620	0.617

3. 基于int1实现的功能 1) Wake up 当传感器静止放在水平面时， x轴和y轴是一个接近0mg的值，这时可以设置x轴和y轴中任意一个值大于设定的阈值，当触发中断时，表明这个器件姿态发生了变化，导致 x轴和y轴大于阈值，从而唤醒主控。但有个问题是当器件不是放在水平面时怎么办， x和y 可能是一个任意的-1g到1g 的值，如何设定这个阈值， 还有在水平的情况下， 又如何检测z 轴， 因为此时z轴是个1 g或-1g的值。最好的情况是当器件静止但在任意姿态时， 最好三个轴的输出都接近0mg， 我只要检测到任意一个轴大于一个小的阈值， 就表明器件姿态发生了改变， 相比前面只检测x和y轴， 这样灵敏度会大大提高。要达到这个目的就要用到highpass filter 功能。

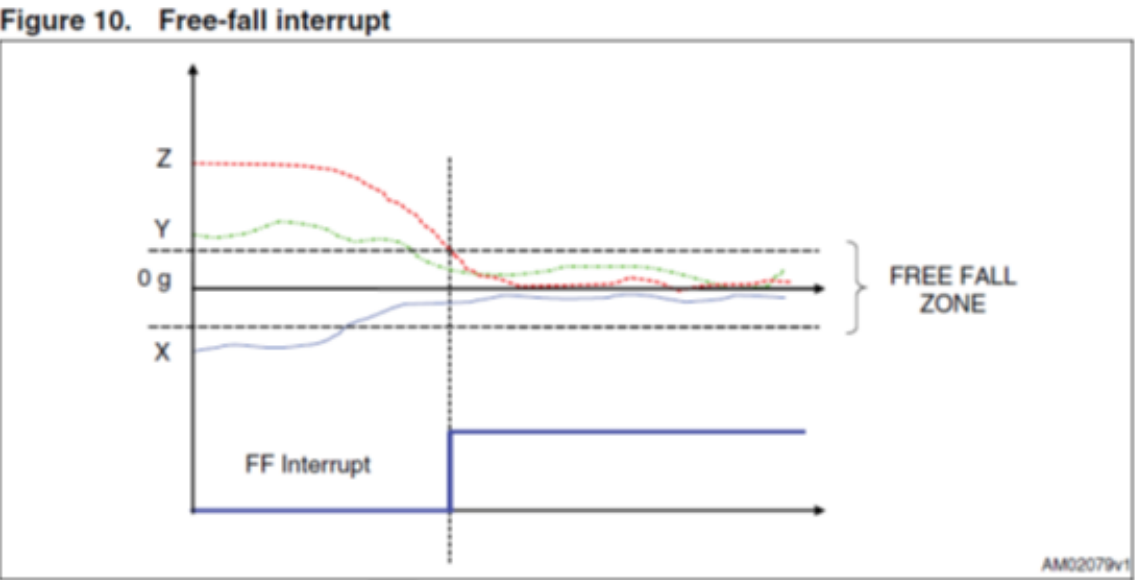


CTRL_REG2 (21h)

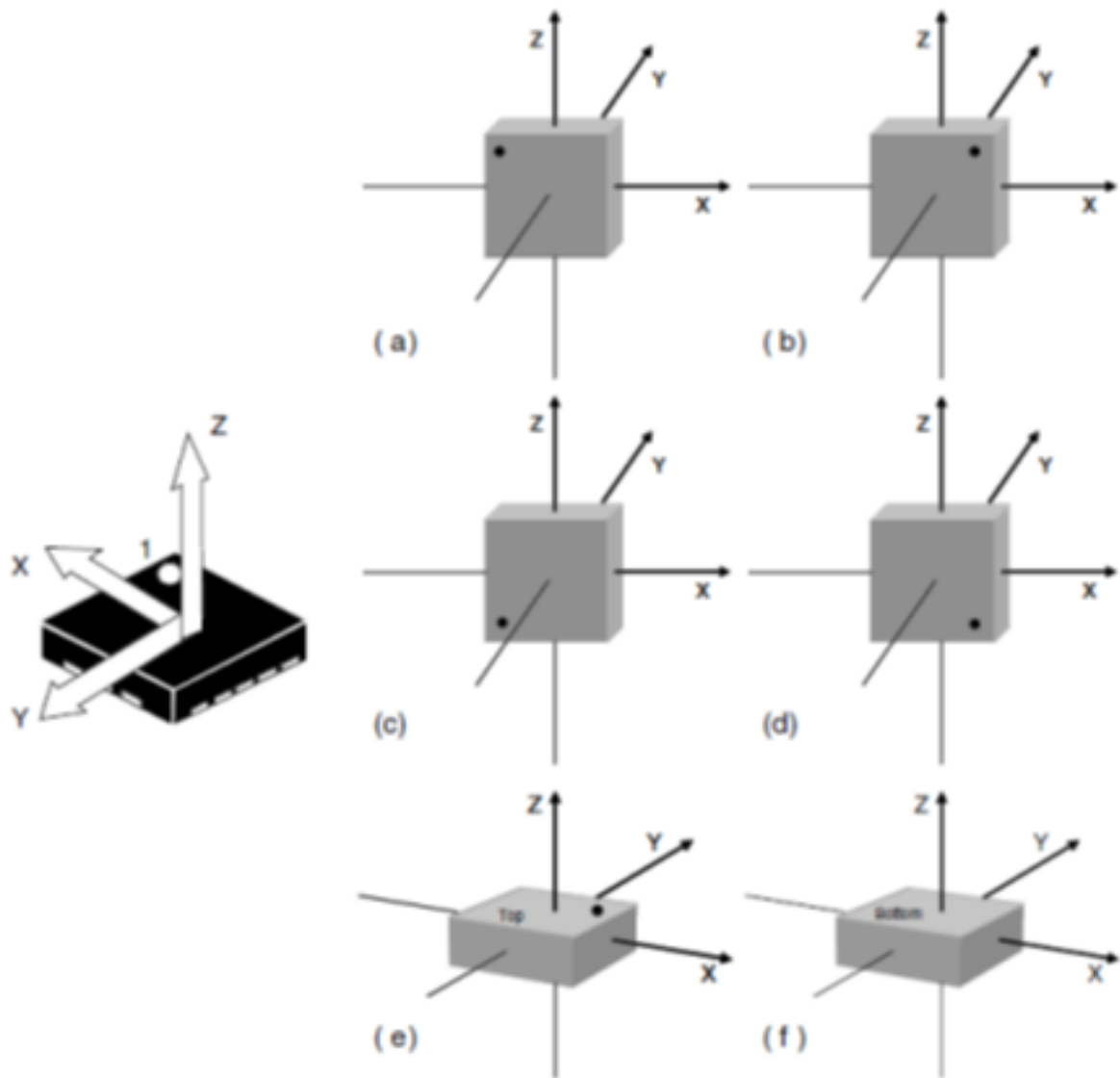
Table 27. CTRL_REG2 register

HPM1	HPM0	HPCF2	HPCF1	FDS	HPCLICK	HPIS2	HPIS1
------	------	-------	-------	-----	---------	-------	-------

当设置了HPIS1后，highpass filter可以滤掉数据的直流分量，只要器件静止，任何一个轴的数据都是0mg，当器件姿态变化时，这个变化的量仍能体现在数据的输出上，所以可以检测相应变化。Note: 这里的highpass filter 只作用在int1上，而不会影响sensor的数据输出。2) free fall 这个功能刚好是与wake up相反的，它是检测到三个轴都小于一个阈值，另一个不同的是触发条件是三个轴同时满足，和wakeup 任意一个轴触发是不同的，体现在寄存器INT1_CFG 的AOIbit，AOI 决定了这些信号是与关系还是或关系，显然对于wake up 是或的关系，对于freefall是与的关系。



3) 6D 如下图，你可以看到器件的6个position，6D功能就是检测当前是在那个或移动到哪个position的一种状态识别。



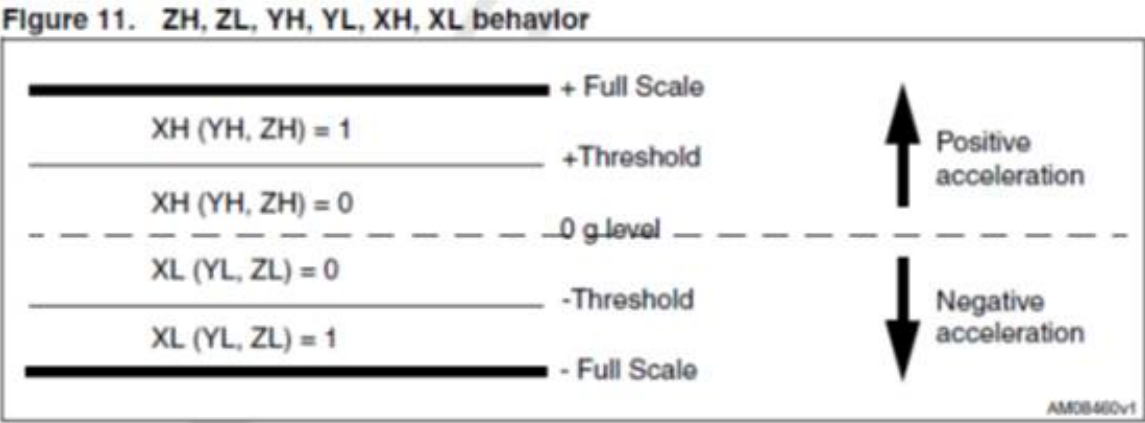
这个功能要用到INT1的另一个source寄存器，实际上就是通过source寄存器来确定6D position。

INT1_SRC (31h)

Table 49. INT1_SRC register

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

XH, YH, ZH, 分别表示x, y, z轴在正方向大于或小于阈值的状态 XL, YL, ZL, 分别表示x, y, z轴在负方向上大于或小于阈值的状态



如下表source的组合对应的就是上图的6个positions。

Table 16. INT1_SRC register in 6D position

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

这里6d有两种工作模式，一个是movement识别，另一个是position识别。 Movement: 就是当物体从一个已知的或未知的方向转动到另一个已知的但不同原方向的方向，并且产生一个1/ODR(数据输出平率) 周期的中断。 Position: 当物体稳定在一个已知的方向， 将产生一个持续的中断信号直到方向改变。 **通过INT_CFG1的AO1, 6D bit 配置这两种模。** Movement: AOI-6D = 01 Position: AOI-6D = 11 对于这六个面的识别，很关键的是阈值的设置，在相对静态的转动时，每个轴最大是1g 或-1g，考虑到sensor的 offset，和一些其他的误差，所以阈值要适当小于1g，但不能太小，不能出现两个轴同时大于阈值的情况。