

## IIS3DWB: 超高带宽、低噪声 3 轴数字振动传感器

### 引言

本文档旨在提供意法半导体 IIS3DWB 振动传感器相关的使用信息和应用提示。

IIS3DWB 是一款高性能三轴 MEMS 振动传感器，具有超宽带宽、平坦的频率响应、低噪声和数字 SPI 接口标准输出。

该器件具有  $\pm 2/\pm 4/\pm 8/\pm 16$  g 的动态用户可选满量程加速度范围（16 位数据输出，数据速率 26.667 kHz），并能以 6.3 kHz 的信号带宽测量加速度。

IIS3DWB 具有超宽带宽、高达 6.3 kHz 的平坦频率响应、消除频率混叠和低噪声的嵌入式滤波功能，可在工业应用中实现开箱即用、高质量和具有成本效益的振动监测。

IIS3DWB 采用纤薄的小型塑料焊盘栅格阵列封装（LGA），可确保在更大的温度范围（-40 °C 至 +105 °C）内正常工作。



**表 1. 默认引脚状态**

引脚#	名称	功能	默认状态	建议的连接
1	SDO/SA0	SPI 4 线接口串行数据输出 (serial data output, SDO) I <sup>2</sup> C 设备地址的最低有效位 (SA0)	输入无上拉 如果寄存器 02h 中的位 SDO_PU_EN = 1, 则启用上拉	专用
2	RES	保留	输入无上拉	连接到 VDD_IO 或 GND
3	RES	保留	输入无上拉	连接到 VDD_IO 或 GND
4	INT1	可编程中断 1	带下拉的输入	设备通电期间必须设置为 0 或保持未连接状态。 设备通电后, 根据特定应用进行连接。
5	VDD_IO	I/O 引脚的供电	-	
6	GND	接地	-	
7	GND	接地	-	
8	VDD	电源	-	
9	INT2	可编程中断 2	输出强制接地	专用
10	RES	保留	带上拉的输入	连接到 VDD <sub>IO</sub> 或保持引脚电气断开并焊接到 PCB
11	RES	保留	带上拉的输入	连接到 VDD <sub>IO</sub> 或保持引脚电气断开并焊接到 PCB
12	CS	I <sup>2</sup> C/SPI 模式选择 (1: SPI 空闲模式 / 启用了 I <sup>2</sup> C 通讯; 0: SPI 通信模式 / 禁用 I <sup>2</sup> C)	带上拉的输入 如果寄存器 13h 中的位 I2C_DISABLE = 1, 则上拉禁用。	专用
13	SPC/SCL	SPI 串口时钟 (serial port clock, SPC) I <sup>2</sup> C 串行时钟 (SCL)	输入无上拉	专用
14	SDI/SDO/SDA	SPI 串行数据输入 (serial data input, SDI) 3 线接口串行数据输出 (serial data output, SDO) I <sup>2</sup> C 串行数据 (SDA)	输入无上拉	专用

## 2 寄存器



表 2. 寄存器

寄存器名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PIN_CTRL	02h	0	SDO_PU_EN	1	1	1	1	1	1
FIFO_CTRL1	07h	WTM7	WTM6	WTM5	WTM4	WTM3	WTM2	WTM1	WTM0
FIFO_CTRL2	08h	STOP_ON_WTM	0	0	0	0	0	0	WTM8
FIFO_CTRL3	09h	0	0	0	0	BDR_XL_3	BDR_XL_2	BDR_XL_1	BDR_XL_0
FIFO_CTRL4	0Ah	DEC_TS_BATCH_1	DEC_TS_BATCH_0	ODR_T_BATCH_1	ODR_T_BATCH_0	0	FIFO_MODE2	FIFO_MODE1	FIFO_MODE0
COUNTER_BDR_REG1	0Bh	dataready_pulsed	RST_COUNTER_BDR	0	0	0	CNT_BDR_TH_10	CNT_BDR_TH_9	CNT_BDR_TH_8
COUNTER_BDR_REG2	0Ch	CNT_BDR_TH_7	CNT_BDR_TH_6	CNT_BDR_TH_5	CNT_BDR_TH_4	CNT_BDR_TH_3	CNT_BDR_TH_2	CNT_BDR_TH_1	CNT_BDR_TH_0
INT1_CTRL	0Dh	0	INT1_CNT_BDR	INT1_FIFO_FULL	INT1_FIFO_OVR	INT1_FIFO_TH	INT1_BOOT	0	INT1_DRDY_XL
INT2_CTRL	0Eh	0	INT2_CNT_BDR	INT2_FIFO_FULL	INT2_FIFO_OVR	INT2_FIFO_TH	INT2_DRDY_TEMP	0	INT2_DRDY_XL
WHO_AM_I	0Fh	0	1	1	1	1	0	1	1
CTRL1_XL	10h	XL_EN_2	XL_EN_1	XL_EN_0	0	FS1_XL	FS0_XL	LPF2_XL_EN	0
CTRL3_C	12h	BOOT	BDU	H_LACTIVE	PP_OD	SIM	IF_INC	0	SW_RESET
CTRL4_C	13h	0	0	INT2_on_INT1	0	DRDY_MASK	I2C_disable	0	1AX_TO_3REGOUT
CTRL5_C	14h	0	ROUNDING1	ROUNDING0	0	0	0	ST1_XL	ST0_XL
CTRL6_C	15h	0	0	0	0	USR_OFF_W	0	XL_AXIS_SEL_1	XL_AXIS_SEL_0
CTRL7_C	16h	0	0	0	0	0	0	USR_OFF_ON_OUT	0
CTRL8_XL	17h	HPCF_XL2	HPCF_XL1	HPCF_XL0	HP_REF_MODE_XL	FASTSETTL_MODE_XL	FDS	0	0
CTRL10_C	19h	0	0	TIMESTAMP_EN	0	0	0	0	0
ALL_INT_SRC	1Ah	TIMESTAMP_ENDCOUNT	0	SLEEP_CHANGE_IA	0	0	0	WU_IA	0
WAKE_UP_SRC	1Bh	0	SLEEP_CHANGE_IA	0	SLEEP_STATE_IA	WU_IA	X_WU	Y_WU	Z_WU
STATUS_REG	1Eh	0	0	0	0	0	TDA	0	XLDA
OUT_TEMP_L	20h	Temp7	Temp6	Temp5	Temp4	Temp3	Temp2	Temp1	Temp0
OUT_TEMP_H	21h	Temp15	Temp14	Temp13	Temp12	Temp11	Temp10	Temp9	Temp8
OUTX_L_A	28h	D7	D6	D5	D4	D3	D2	D1	D0
OUTX_H_A	29h	D15	D14	D13	D12	D11	D10	D9	D8
OUTY_L_A	2Ah	D7	D6	D5	D4	D3	D2	D1	D0
OUTY_H_A	2Bh	D15	D14	D13	D12	D11	D10	D9	D8

寄存器名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OUTZ_L_A	2Ch	D7	D6	D5	D4	D3	D2	D1	D0
OUTZ_H_A	2Dh	D15	D14	D13	D12	D11	D10	D9	D8
FIFO_STATUS1	3Ah	DIFF_FIFO_7	DIFF_FIFO_6	DIFF_FIFO_5	DIFF_FIFO_4	DIFF_FIFO_3	DIFF_FIFO_2	DIFF_FIFO_1	DIFF_FIFO_0
FIFO_STATUS2	3Bh	FIFO_WTM_IA	FIFO_OVR_IA	FIFO_FULL_IA	COUNTER_BDR_IA	FIFO_OVR_LATCHED	0	DIFF_FIFO_9	DIFF_FIFO_8
TIMESTAMP0	40h	T7	T6	T5	T4	T3	T2	T1	T0
TIMESTAMP1	41h	T15	T14	T13	T12	T11	T10	T9	T8
TIMESTAMP2	42h	T23	T22	T21	T20	T19	T18	T17	T16
TIMESTAMP3	43h	T31	T30	T29	T28	T27	T26	T25	T24
SLOPE_EN	56h	0	0	SLEEP_STATUS_ON_INT	SLOPE_FDS	0	0	0	LIR
INTERRUPTS_EN	58h	INTERRUPTS_ENABLE	0	0	0	0	0	0	0
WAKE_UP_THS	5Bh	0	USR_OFF_ON_WU	WK_THS5	WK_THS4	WK_THS3	WK_THS2	WK_THS1	WK_THS0
WAKE_UP_DUR	5Ch	0	WAKE_DUR1	WAKE_DUR0	WAKE_THS_W	SLEEP_DUR3	SLEEP_DUR2	SLEEP_DUR1	SLEEP_DUR0
MD1_CFG	5Eh	INT1_SLEEP_CHANGE	0	INT1_WU	0	0	0	0	0
MD2_CFG	5Fh	INT2_SLEEP_CHANGE	0	INT2_WU	0	0	0	0	INT2_TIMESTAMP
INTERNAL_FREQ_FINE	63h	FREQ_FINE7	FREQ_FINE6	FREQ_FINE5	FREQ_FINE4	FREQ_FINE3	FREQ_FINE2	FREQ_FINE1	FREQ_FINE0
X_OFS_USR	73h	X_OFS_USR_7	X_OFS_USR_6	X_OFS_USR_5	X_OFS_USR_4	X_OFS_USR_3	X_OFS_USR_2	X_OFS_USR_1	X_OFS_USR_0
Y_OFS_USR	74h	Y_OFS_USR_7	Y_OFS_USR_6	Y_OFS_USR_5	Y_OFS_USR_4	Y_OFS_USR_3	Y_OFS_USR_2	Y_OFS_USR_1	Y_OFS_USR_0
Z_OFS_USR	75h	Z_OFS_USR_7	Z_OFS_USR_6	Z_OFS_USR_5	Z_OFS_USR_4	Z_OFS_USR_3	Z_OFS_USR_2	Z_OFS_USR_1	Z_OFS_USR_0
FIFO_DATA_OUT_TAG	78h	TAG_SENSOR_4	TAG_SENSOR_3	TAG_SENSOR_2	TAG_SENSOR_1	TAG_SENSOR_0	TAG_CNT_1	TAG_CNT_0	TAG_PARITY
FIFO_DATA_OUT_X_L	79h	D7	D6	D5	D4	D3	D2	D1	D0
FIFO_DATA_OUT_X_H	7Ah	D15	D14	D13	D12	D11	D10	D9	D8
FIFO_DATA_OUT_Y_L	7Bh	D7	D6	D5	D4	D3	D2	D1	D0
FIFO_DATA_OUT_Y_H	7Ch	D15	D14	D13	D12	D11	D10	D9	D8
FIFO_DATA_OUT_Z_L	7Dh	D7	D6	D5	D4	D3	D2	D1	D0
FIFO_DATA_OUT_Z_H	7Eh	D15	D14	D13	D12	D11	D10	D9	D8



### 3 工作模式

IIS3DWB 提供两种可能的操作配置：

- 掉电模式；
- 正常模式。

该设备提供 2.1 V 到 3.6 V 的宽 VDD 电压范围和 1.62 V 到 VDD+0.1V 的 VDDIO 范围。上电时序不受限制：VDD/VDDIO 引脚可以设置为电源电平或接地电平（禁止设置为悬空状态），并且不需要特定的上电时序。

为了避免潜在的冲突，在上电时序期间，建议将连接到设备 IO 引脚的线路（主机侧）设置为悬空或接地，直到设置了 VDDIO。设置 VDDIO 后，必须根据表 1. 默认引脚状态中描述的默认状态对 IO 引脚进行配置。为了避免电流消耗的意外增加，未上拉/下拉的输入引脚必须由主机极化。

施加电源后，器件执行一段 10 ms（最长）的启动程序来加载修整参数。启动完成后，器件会自动配置为掉电模式。此外，为保证器件能正确关断，建议将 VDD 线接地的持续时间保持至少 100 $\mu$ s。

当传感器处于掉电模式时，器件的几乎所有内部模块都会关闭。SPI / I<sup>2</sup>C 数字接口保持激活，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，可保存进入下电模式前存储器中采样的最后数据。

在正常模式下，可通过 CTRL1\_XL 寄存器 XL\_EN[2:0] 位的位置配置 IIS3DWB：传感器以 26.667 kHz 的输出数据率提供加速度数据。该器件的可选满量程加速度范围为  $\pm 2$  g、 $\pm 4$  g、 $\pm 8$  g 或  $\pm 16$  g。

可使用 CTRL6\_C 寄存器的 XL\_AXIS\_SEL [1:0] 位来选择加速度计的活动轴，如下表所示。有两种工作模式可用：

- 三轴模式：所有三个轴（X、Y、Z）同时使能；
- 单轴模式：仅一个轴活动（其他两个轴的输出值设置为 0）。

表 3. 加速度计使能轴

XL_AXIS_SEL[1:0]	使能轴
00（默认）	3 轴（XYZ）
01	X 轴
10	Y 轴
11	Z 轴

在上述两种情况下，都可以使用加速度计输出寄存器（参见第 4 节 读取输出数据）或 FIFO 寄存器（参见第 6 节 先进先出（FIFO）缓冲区）从传感器同时读取活动轴的加速度数据

当器件处于断电模式时，必须通过 XL\_AXIS\_SEL [1:0] 位选择活动轴，然后才能将器件设置为正常模式。

在单轴模式下，虽然 IIS3DWB 的功耗与 3 轴模式（1.1 mA typ.）相同，但活动轴的分辨率（噪声密度）显著提高。

**注意：**只有 SPI 接口支持所有设备特性和功能。I<sup>2</sup>C 接口只能在单轴模式下使用，不建议使用。

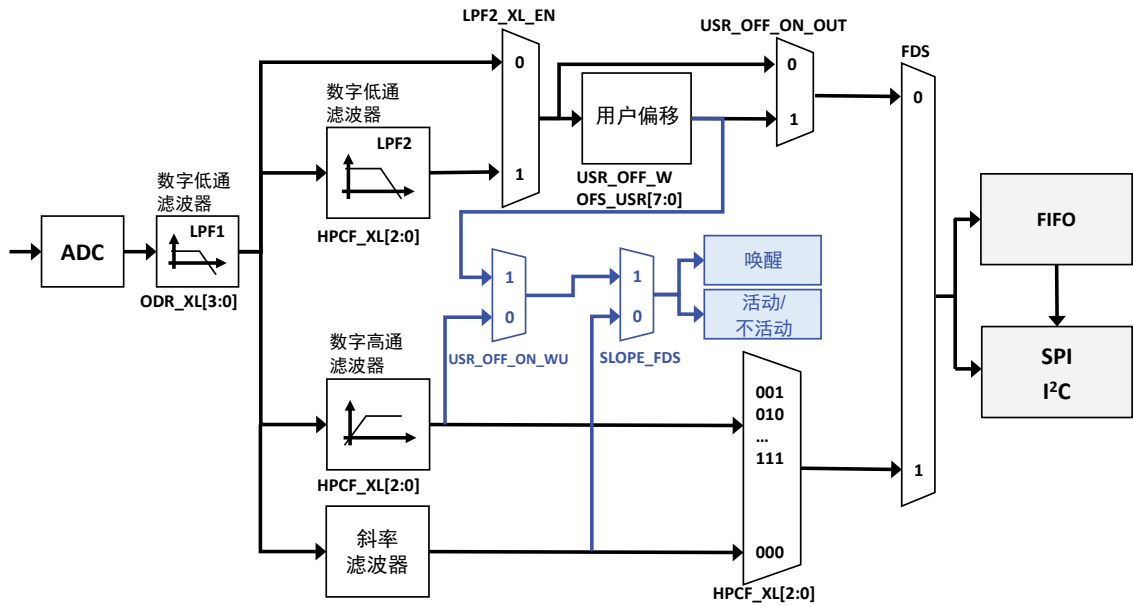
### 3.1 加速度计滤波链

IIS3DWB 提供了宽带宽，通带频率响应非常平坦，在停止带中衰减很高；IIS3DWB 的频率响应细节见器件数据手册。

加速度计采样链路由 3 个级联主模块表示：一个 ADC 转换器，一个数字低通滤波器（LPF1）和复合数字滤波器组，如图 2. 加速度计滤波链所示。

来自机械部件的模拟信号由 ADC 转换；然后，数字信号由截止频率等于 6.3 kHz 的数字低通滤波器 LPF1 过滤。

图 2. 加速度计滤波链



由一个低通数字滤波器（LPF2）、一个高通数字滤波器和一个斜率滤波器组成的组合滤波器组处理数字信号。

CTRL1\_XL 寄存器的 LPF2\_XL\_EN 位和 CTRL8\_XL 寄存器可用来配置复合滤波器组和加速度计滤波链的总带宽，如表 4. 加速度计带宽选择中所示。参考该表，在低通路径侧，如果 LPF2\_XL\_EN = 0，则带宽是指 LPF1 带宽；如果 LPF2\_XL\_EN = 1，则带宽指 LPF2 带宽。在高通路径侧，如果 HPCF\_XL[2:0] = 000b，则带宽列表表示斜率滤波器带宽；对于所有其他配置，则表示 HP 滤波器带宽。

**表 4. 加速度计带宽选择**

FDS	LPF2_XL_EN	HPCF_XL[2:0]	带宽	总稳定时间的最大值 <sup>(1)</sup> (要丢弃的采样数)	
0 (低通路径)	0	-	6.3 kHz	12	
		000	ODR/4	12	
		001	ODR/10	12	
	1	1	010	ODR/20	19
			011	ODR/45	38
			100	ODR/100	75
			101	ODR/200	150
			110	ODR/400	295
			111	ODR/800	595
1 (高通路径)	-	000	ODR / 4 (斜率滤波器)	13	
		001	ODR/10	13	
		010	ODR/20	19	
		011	ODR/45	38	
		100	ODR/100	75	
		101	ODR/200	150	
		110	ODR/400	295	
		111	ODR/800	595	

1. 最终值的 99% 时的稳定时间。

通过将 CTRL8\_XL 寄存器的 FDS 位设置为 0，选择复合滤波器模块的低通路径。如果 LPF2\_XL\_EN 位置为 0，则无需应用额外的滤波器；如果 LPF2\_XL\_EN 位置为 1，那么除 LPF1 外还需应用 LPF2 滤波器，并可通过配置 CTRL8\_XL 寄存器的 HPCF\_XL [2:0] 字段来设置加速度计链的总带宽。

通过将 FDS 位设置为 1，选择复合滤波器模块的高通路径：HPCF\_XL [2:0] 字段除了能用于启用 LPF1 滤波器之外，还可使能斜率滤波器（HPCF\_XL[2:0] = 000b 时）或数字高通滤波器（其他 HPCF\_XL [2:0] 配置）。HPCF\_XL [2:0] 字段也可用来选择 HP 滤波器的截止频率。

高通滤波器参考模式功能可用于加速度计传感器：启用此功能后，当前的 X、Y、Z 加速度计采样将在内部存储，并从所有后续输出值中减去。为了使能参考模式，必须将 CTRL8\_XL 寄存器的 HP\_REF\_MODE\_XL 位和 FDS 位设置为 1，并且 HPCF\_XL [2:0] 字段的值必须等于 111b。当启用 Reference 模式功能时，LPF2 滤波器和 HP 滤波器都不可用。启用 Reference 模式之后的第一个加速度计输出数据必须被丢弃。

CTRL8\_XL 寄存器的 FASTSETTL\_MODE\_XL 位使能加速度计 LPF2 或 HPF 快速建立模式：选择的滤波器在写入此位后设置第二个样本。仅在器件退出下电模式时应用此功能。



### 3.1.1 加速度计斜率滤波器

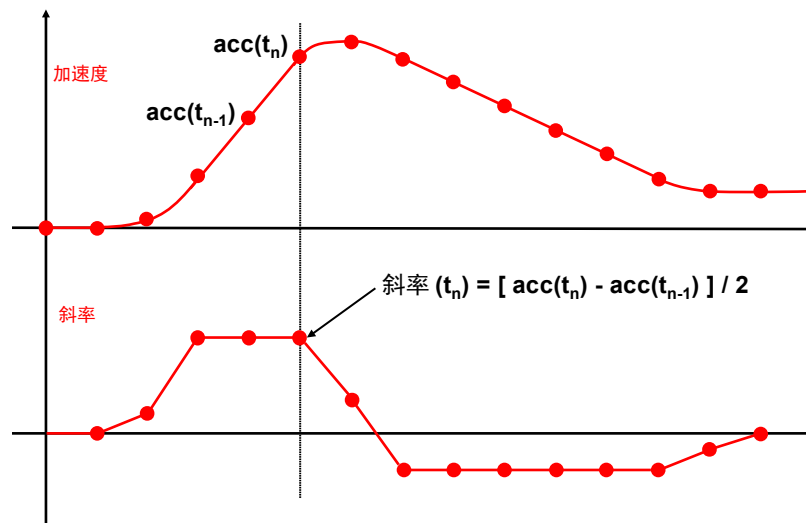
如图 2. 加速度计滤波链中所示，器件嵌入了一个数字斜率滤波器，该滤波器还可用于嵌入功能（唤醒检测和活动/不活动）。

该斜率滤波器输出数据利用以下公式进行计算：

$$\text{slope}(t_n) = [ \text{acc}(t_n) - \text{acc}(t_{n-1}) ] / 2$$

下图举例说明了斜率数据信号的示例。

图 3. 加速度计斜率滤波器



### 3.2 加速度计开启/关断时间

加速度计读取链包含低通滤波，以改善信噪比性能。因此，在切换加速度计工作模式时，必须考虑滤波器的建立时间。

表 4. 加速度计带宽选择 针对加速度计滤波链各种配置将丢弃的样本，提供了最大(最坏情况)导通时间(从断电模式切换到正常模式时)。

从正常模式切换到断电模式时的最大关断时间为 1  $\mu\text{s}$ 。

## 4 读取输出数据

### 4.1 启动序列

当器件上电时，器件会自动从内嵌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 10 毫秒后，加速度计会自动进入掉电模式。

要打开加速度计并通过 SPI /I<sup>2</sup>C 接口收集加速度数据，必须将 CTRL1\_XL 寄存器的 XL\_EN [2:0]位配置为 101b。

以下通用线序可用于配置加速度计：

1. 写 INT1\_CTRL = 01h // INT1 上，Acc 数据准备就绪中断
2. 写入 CTRL6\_C = 00h // 使能三轴模式
3. 写 CTRL1\_XL = A0h // 使能加速度计 (ODR = 26.667 kHz; FS = ±2 g)

### 4.2 使用状态寄存器

该器件具有一个 STATUS\_REG 寄存器，应当对该寄存器进行轮询以检查一组新数据何时可用。当加速度计输出中有一组新的数据可用时，XLDA 位被置为 1。

加速度计输出寄存器（三轴模式下）的读取应按以下步骤进行：

1. 读取 STATUS\_REG
2. 如果 XLDA = 0，则进入 1
3. 读取 OUTX\_L\_A
4. 读取 OUTX\_H\_A
5. 读取 OUTY\_L\_A
6. 读取 OUTY\_H\_A
7. 读取 OUTZ\_L\_A
8. 读取 OUTZ\_H\_A
9. 数据处理
10. 跳到步骤 1

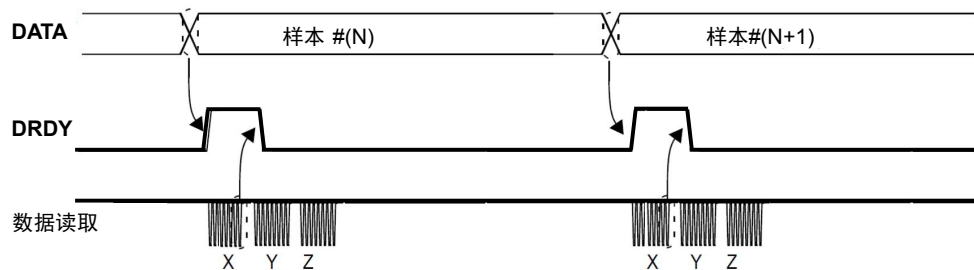
### 4.3 使用数据准备就绪信号

该器件可配置为具有一个硬件信号，以确定新的一组测量数据何时可以读取。

对于加速度计传感器，数据准备就绪信号由 STATUS\_REG 寄存器的 XLDA 位表示。通过将 INT1\_CTRL 寄存器的 INT1\_DRDY\_XL 位置为 1，可将该信号驱动至 INT1 引脚，通过将 INT2\_CTRL 寄存器的 INT2\_DRDY\_XL 位置为 1，将其驱动至 INT2 引脚。

当一组新数据生成并可读取时，数据准备就绪信号升高为 1。数据就绪信号可以是锁存的或脉冲的：如果 COUNTER\_BDR\_REG1 寄存器的 dataready\_pulsed 位被置为 0（默认值），则数据就绪信号被锁存，并且当其中一个通道的较高部分（29h、2Bh、2Dh 寄存器）被读取时，中断复位。如果 COUNTER\_BDR\_REG1 寄存器的 dataready\_pulsed 位置为 1，则数据就绪信号是脉冲的，并且在中断引脚上观察到的脉冲持续时间为 18.75  $\mu\text{s}$ （典型值）。脉冲模式不适用于总是锁存的 XLDA 位。

图 4. 数据准备就绪信号



#### 4.3.1 DRDY 屏蔽功能

如果将 CTRL4\_C 寄存器的 DRDY\_MASK 位置为 1，则加速度计数据准备就绪信号会被屏蔽，直至滤波器配置稳定。

当 FIFO 处于激活状态且 DRDY\_MASK 位置为 1 时，存储在 FIFO 中的加速计无效的数值可能为 7FFFh、7FFEh 或 7FFDh。这样，存储在 FIFO 缓冲器中的无效采样被加上了一个标签，因此在数据后处理过程中，可以容易地识别出它们并将其丢弃。

注：DRDY\_MASK 位仅对加速度计 LPF1 数字滤波器的建立时间起作用。

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

如果读取加速度计数据特别慢，并且不能（或者不需要）与 STATUS\_REG 寄存器中的 XLDA 位或驱动到 INT1/INT2 引脚的 DRDY 信号同步，那么强烈建议将 CTRL3\_C 寄存器中的 BDU（块数据更新）位置为 1。

此功能可以避免读取到不同采样的值（输出数据的高字节和低字节部分）。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输出数据，但如果发起了对给定寄存器对（即 OUTX\_H\_A 和 OUTX\_L\_A、OUTY\_H\_A 和 OUTY\_L\_A、OUTZ\_H\_A 和 OUTZ\_L\_A）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。

请注意：BDU 只能确保 LSB 部分和 MSB 部分同一时刻被采样。例如，如果读取速度非常慢，则 X 和 Y 可在 T1 读取，Z 在 T2 采样。

BDU 功能还作用于 FIFO\_STATUS1 和 FIFO\_STATUS2 寄存器。当 BDU 位置为 1 时，必须首先读取 FIFO\_STATUS1，然后读取 FIFO\_STATUS2。

## 4.5 理解输出数据

测得的加速度数据被发送到 OUTX\_H\_A, OUTX\_L\_A, OUTY\_H\_A, OUTY\_L\_A, OUTZ\_H\_A 和 OUTZ\_L\_A 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 通道的完整输出数据是由 OUTX\_H\_A & OUTX\_L\_A、OUTY\_H\_A & OUTY\_L\_A、OUTZ\_H\_A & OUTZ\_L\_A 共同提供的，表示为二进制补码。

加速度数据表示为 16 比特的数字。

### 4.5.1 输出数据示例

表 5. 输出数据寄存器内容 vs. 加速度 (FS\_XL = ±2 g) 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。

下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 5. 输出数据寄存器内容 vs. 加速度 (FS\_XL = ±2 g)

加速度值	寄存器地址	
	OUTX_H_A (29h)	OUTX_L_A (28h)
0 g	00h	00h
350 mg	16h	69h
1 g	40h	09h
-350 mg	E9h	97h
-1 g	BFh	F7h

## 4.6 加速度计偏移寄存器

器件提供了加速度计偏移寄存器 (X\_OFS\_USR、Y\_OFS\_USR、Z\_OFS\_USR)，可用于零偏校正，或者用来将偏移量应用于加速度计输出数据。

可通过 CTRL7\_C 寄存器 USR\_OFF\_ON\_OUT 位的置位来使能加速度计偏移功能块。在偏移寄存器中设置的偏移值从测得的相应轴加速度值内部减去；内部处理的数据随后被发送到加速计输出寄存器和 FIFO（如果启用）。这些寄存器值以二进制补码的形式表示为 8 位数据，并且必须在[-127,127]的范围内。

应用于偏移寄存器值的权重[g/LSB]独立于加速度计所选的量程，并可利用 CTRL6\_C 寄存器的 USR\_OFF\_W 位进行配置：

- $2^{-10}$ g/LSB，如果 USR\_OFF\_W 位置为 0；
- $2^{-6}$ g/LSB，如果 USR\_OFF\_W 位置为 1。

## 4.7 环行功能

环行功能可用来自动寻址器件寄存器，以进行多字节读取。基本上，伴随多字节读取操作，所读取的寄存器地址会自动从第一个寄存器转到最后一个寄存器，然后返回第一个寄存器。

为了启用加速度计输出寄存器（从 OUTX\_L\_A (28h) 到 OUTZ\_H\_A (2Dh)）的环行功能，必须将 CTRL5\_C 寄存器的 ROUNDING [1:0]位设置为 01b。

## 5 中断生成

中断生成基于加速度计数据，因此，出于中断生成的目的，必须将加速度计传感器设置为“正常”模式。

中断生成器可以配置为检测唤醒事件以及活动/不活动事件。

这些中断信号，以及 FIFO 中断信号，可被独立地驱动至 INT1 和 INT2 中断引脚，或通过读取特定源寄存器位分别对其进行检测。

必须使用 CTRL3\_C 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有效，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（低电平有效），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。

CTR3\_C 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。

### 5.1 中断引脚配置

该器件具有两个引脚，可激活引脚来生成数据准备就绪或中断信号。这些引脚的功能，对于 INT1 引脚是通过 MD1\_CFG 和 INT1\_CTRL 寄存器来进行选择，对于 INT2 引脚是通过 MD2\_CFG 和 INT2\_CTRL 寄存器来进行选择。

以下概述给出了这些中断控制寄存器的简要描述：这些位的默认值等于 0，对应于“禁用”。要使能引脚上特定中断信号的线路，须将有关位置为 1。

**表 6. INT1\_CTRL 寄存器**

b7	b6	b5	b4	b3	b2	b1	b0
0	INT1_CNT_BDR	INT1_FIFO_FULL	INT1_FIFO_OVR	INT1_FIFO_TH	INT1_BOOT	0	INT1_DRDY_XL

- INT1\_CNT\_BDR: INT1 上 FIFO COUNTER\_BDR\_IA 中断
- INT1\_FIFO\_FULL: INT1 上 FIFO 全满标志中断
- INT1\_FIFO\_OVR: INT1 上 FIFO 溢出标志中断
- INT1\_FIFO\_TH: INT1 上 FIFO 阈值中断
- INT1\_BOOT: INT1 上启动中断
- INT1\_DRDY\_XL: INT1 上加速度计数据准备就绪

**表 7. MD1\_CFG 寄存器**

b7	b6	b5	b4	b3	b2	b1	b0
INT1_SLEEP_CHANGE	0	INT1_WU	0	0	0	0	0

- INT1\_SLEEP\_CHANGE: INT1 上的活动/不活动识别事件中断
- INT1\_WU: INT1 上唤醒中断

**表 8. INT2\_CTRL 寄存器**

b7	b6	b5	b4	b3	b2	b1	b0
0	INT2_CNT_BDR	INT2_FIFO_FULL	INT2_FIFO_OVR	INT2_FIFO_TH	INT2_DRDY_TEMP	0	INT2_DRDY_XL

- INT2\_CNT\_BDR: INT2 上 FIFO COUNTER\_BDR\_IA 中断
- INT2\_FIFO\_FULL: INT2 上 FIFO 全满标志中断
- INT2\_FIFO\_OVR: INT2 上 FIFO 溢出标志中断
- INT2\_FIFO\_TH: INT2 上 FIFO 阈值中断

- INT2\_DRDY\_TEMP: INT2 上温度数据准备就绪
- INT2\_DRDY\_XL: INT2 上加速度计数据准备就绪

**表 9. MD2\_CFG 寄存器**

b7	b6	b5	b4	b3	b2	b1	b0
INT2_SLEEP_CHANGE	0	INT2_WU	0	0	0	0	INT2_TIMESTAMP

- INT2\_SLEEP\_CHANGE: INT2 上的活动/不活动识别事件中断
- INT2\_WU: INT2 上唤醒中断
- INT2\_TIMESTAMP: INT2 上的时间戳溢出报警中断

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”。要识别出是什么事件产生了中断条件，则须读取相关源寄存器：

- WAKE\_UP\_SRC, ALL\_INT\_SRC (用于唤醒和活动/不活动功能)
- STATUS\_REG (用于数据准备就绪信号)
- FIFO\_STATUS2 (用户 FIFO)

CTRL4\_C 寄存器的 INT2\_on\_INT1 引脚能够将所有已使能的中断信号进行逻辑“与”后驱动到 INT1 引脚上 (通过将该位置为 1)。当该位置为 0 时，中断信号分配到 INT1 和 INT2 引脚上。

必须将 INTERRUPTS\_EN 寄存器中的 INTERRUPTS\_ENABLE 位设置为 1 来启用唤醒和活动/不活动中断。

SLOPE\_EN 寄存器的 LIR 位使能基本中断功能的锁存中断：当此位置为 1 且中断标记发送至 INT1 引脚和/或 INT2 引脚时，中断保持激活状态，直至 ALL\_INT\_SRC 寄存器或相应的源寄存器被读取，它在下一个 ODR 周期复位。仅当功能路由至 INT1 或 INT2 引脚时才对功能使能锁存中断：如果使能了锁存模式，但是中断信号未驱动至中断引脚，那么锁存功能不起作用。

*注：如果使能了锁存模式 (LIR = 1)，不建议连续轮询 ALL\_INT\_SRC 或专用源寄存器，因为通过读取它们，嵌入功能将实现内部复位；本例中建议同步 (有中断事件) 读取源寄存器。*

## 5.2 唤醒中断

唤醒功能可利用斜率滤波器 (更多详细信息参见第 3.1.1 节 加速度计斜率滤波器) 或高通数字滤波器来实现，如图 2. 加速度计滤波链所示。所用滤波器可通过 SLOPE\_EN 寄存器的 SLOPE\_FDS 位来选择：如果该位被置为 0 (默认值)，则使用斜率滤波器；如果被置为 1，则使用 HPF 数字滤波器。此外，可以将唤醒功能配置为在可编程位置执行绝对唤醒。这可以通过将 SLOPE\_EN 寄存器的 SLOPE\_FDS 位或 WAKE\_UP\_THS 寄存器的 USR\_OFF\_ON\_WU 位置为 1 来实现。使用此配置时，唤醒功能的输入数据来自于低通滤波器路径，而可编程位置则作为偏移量减去。可编程位置可通过 X\_OFS\_USR、Y\_OFS\_USR 和 Z\_OFS\_USR 寄存器进行配置 (更多详细信息见第 4.6 节 加速度计偏移寄存器)。

如果一定数量的连续滤波数据超出了所配置阈值，则产生唤醒中断信号 (图 5. 唤醒中断 (利用斜率滤波器))。

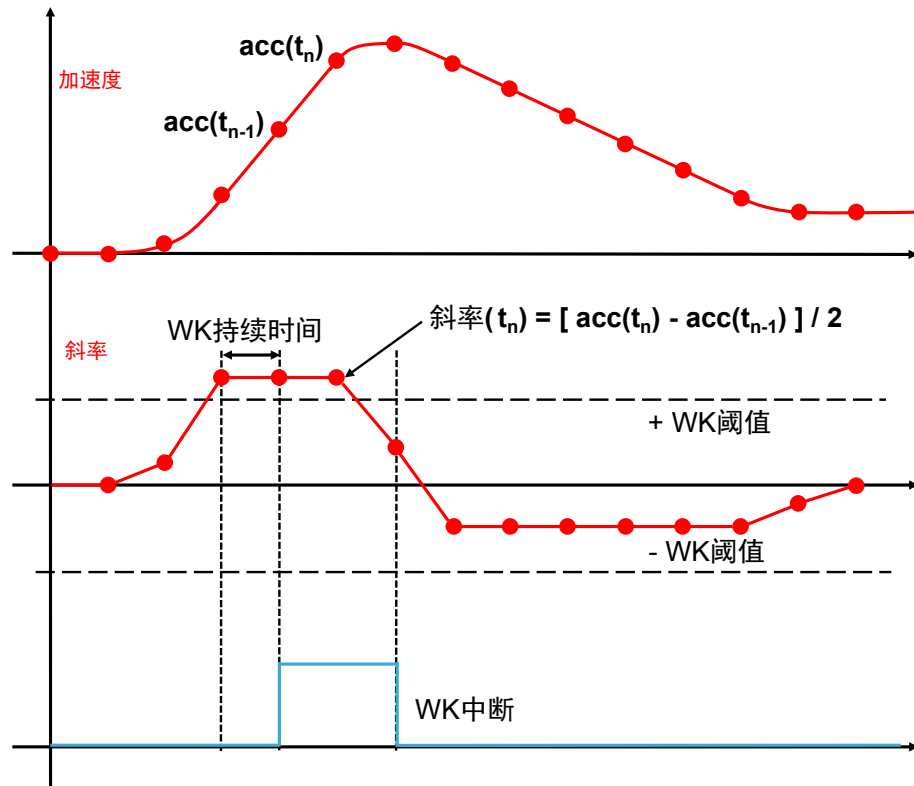
该无符号阈值由 WAKE\_UP\_THS 寄存器的 WK\_THS[5:0] 位来定义；这些 6 位的 1 LSB 值取决于所选加速度计满量程和 WAKE\_UP\_DUR 寄存器的 WAKE\_THS\_W 位的值：

- 如果 WAKE\_THS\_W = 0，则 1 LSB = FS\_XL / 2<sup>6</sup>；
- 如果 WAKE\_THS\_W = 1，则 1 LSB = FS\_XL / 2<sup>8</sup>。

阈值可应用于正负数据：对于唤醒中断生成，滤波数据的绝对值必须大于阈值。

持续时间参数定义了所识别的唤醒事件的最小持续时间；其值由 WAKE\_UP\_DUR 寄存器的 WAKE\_DUR[1:0] 位来设置：1 LSB 对应于 1/ODR\_XL 时间，这里 ODR\_XL 为加速度计输出数据率。要避免因输入信号寄生尖峰而产生不期望的唤醒中断，适当定义持续时间参数是非常重要的。

通过将 INTERRUPTS\_EN 寄存器中的 INTERRUPTS\_ENABLE 位设置为 1，可使能该中断信号，而将 MD1\_CFG 寄存器的 INT1\_WU 位或 MD2\_CFG 寄存器的 INT2\_WU 位设置为 1，可将该中断信号驱动至两个中断引脚上；此外还可通过读取 WAKE\_UP\_SRC 或 ALL\_INT\_SRC 寄存器的 WU\_IA 位对其进行检查。WAKE\_UP\_SRC 寄存器的 X\_WU、Y\_WU、Z\_WU 位指示哪个轴触发了唤醒事件。

**图 5. 唤醒中断（利用斜率滤波器）**


如果锁存模式禁用（SLOPE\_EN 的 LIR 位置为 0），则当滤波数据低于阈值时，中断信号会自动复位。如果锁存模式使能且唤醒中断信号被驱动至中断引脚，那么当发生唤醒事件且声明了中断引脚时，必须通过读取 WAKE\_UP\_SRC 寄存器或 ALL\_INT\_SRC 寄存器来将其复位。X\_WU、Y\_WU 和 Z\_WU 位在执行读取前维持生成中断的状态，并在下一个 ODR 周期释放。如果除了 WU\_IA 位还必须评估 WU\_X、WU\_Y 和 WU\_Z 位，建议直接读取 WAKE\_UP\_SRC 寄存器（此特定情况下不使用 ALL\_INT\_SRC 寄存器）。如果使能了锁存模式，但是中断信号未驱动至中断引脚，那么锁存功能不起作用。

以下给出使用高通数字滤波器进行唤醒事件识别的基本软件程序。

1. 将 A0h 写入 CTRL1\_XL // 使能加速度计（ODR = 26.667 kHz；FS = ±2 g）
2. 将 11h 写入 SLOPE\_EN // 使能锁存模式和数字高通滤波器
3. 将 80h 写入 INTERRUPTS\_EN // 使能中断功能
4. 将 00h 写入 WAKE\_UP\_DUR // 无持续时间和唤醒阈值权重选择（1 LSB = FS\_XL / 2<sup>6</sup>）
5. 将 02h 写入 WAKE\_UP\_THS // 设置唤醒阈值
6. 将 20h 写入 MD1\_CFG // 唤醒中断驱动至 INT1 引脚

由于持续时间被置为 0，因此每个 X、Y、Z 滤波数据超出所配置阈值时，会生成唤醒中断信号。本例中，WAKE\_UP\_THS 寄存器的 WK\_THS 字段被置为 000010b，因此活动/不活动阈值为 62.5 mg (= 2 \* FS\_XL / 2<sup>6</sup>)。如果利用斜率/高通数字滤波器实现唤醒功能，则有必要考虑此功能使能后滤波器的建立时间。例如，当使用斜率滤波器时（不过对于使用高通数字滤波器要做类似考虑），唤醒功能基于阈值与两次(x,y,z)采样（当前和前次）加速度差的一半相比较（参考第 3.1.1 节 加速度计斜率滤波器）。

在第一个采样处，斜率滤波器输出计算为当前采样[例如，(x,y,z) = (0,0,1g)]与前次采样(x,y,z)=(0,0,0)（因为前次采样不存在）之差的一半。因此，在 z 轴上，斜率滤波器的第一个输出值为(1g - 0)/2=500 mg，发生伪中断事件的情况下，该值可能会大于阈值。中断信号保持为高电平，并持续 1 个 ODR，然后变为低电平。

要避免产生这种伪中断，可以有多种解决方案。下面两个备选解决方案（对于斜率滤波器的情形）：

- a. 忽略第一个产生的唤醒信号；
- b. 将中断信号驱动到 INT1/2 引脚前，增加一段高于 1 个 ODR 的等待时间。



### 5.3 活动/不活动识别

活动/不活动嵌入式功能的工作原理类似于唤醒。如果在一段可编程的时间内未检测到运动状态，则生成不活动/静止状态事件；否则，在加速度计数据超过可配置的阈值时，生成活动/运动状态事件。

当 `INTERRUPTS_EN` 寄存器的 `INTERRUPTS_ENABLE` 位设置为 1 时，将使能活动/不活动识别功能。

可以利用斜率滤波器（更多详细信息见第 3.1.1 节 加速度计斜率滤波器）或高通数字滤波器来实现活动/不活动识别功能，如图 2. 加速度计滤波链中所示。所用滤波器可通过 `SLOPE_EN` 寄存器的 `SLOPE_FDS` 位来选择：如果该位被置为 0（默认值），则使用斜率滤波器；如果被置为 1，则使用高通数字滤波器。

此功能可完全由用户编程，利用专门的寄存器组对所期望的滤波数据幅度和时序进行编程（图 6. 活动/不活动识别（利用斜率滤波器））。

该无符号阈值由 `WAKE_UP_THS` 寄存器的 `WK_THS[5:0]` 位来定义；这些 6 位的 1 LSB 值取决于所选加速度计量程和 `WAKE_UP_DUR` 寄存器的 `WAKE_THS_W` 位的值：

- 如果 `WAKE_THS_W = 0`，则  $1 \text{ LSB} = \text{FS}_{\text{XL}} / 2^6$ ；
- 如果 `WAKE_THS_W = 1`，则  $1 \text{ LSB} = \text{FS}_{\text{XL}} / 2^8$ 。

该阈值可适用于正负滤波数据。

待识别的不活动状态的持续时间由 `WAKE_UP_DUR` 寄存器的 `SLEEP_DUR[3:0]` 位来定义：1 LSB 对应于  $512 / \text{ODR}_{\text{XL}}$  的时间，这里 `ODR_{\text{XL}}` 为加速度计输出数据率。如果 `SLEEP_DUR[3:0]` 位被置为 0000b，则待识别不活动状态的持续时间等于  $16 / \text{ODR}_{\text{XL}}$  的时间。

当检测到不活动状态时，中断被置为高电平并持续  $1 / \text{ODR}_{\text{XL}}[\text{s}]$  的时间周期，然后自动取消。

当一个轴上的过滤数据在一个可配置时间内大于阈值时，将检测到活动状态。待识别活动状态的持续时间由 `WAKE_UP_DUR` 寄存器的 `WAKE_DUR[1:0]` 位来定义。1 LSB 对应于  $1 / \text{ODR}_{\text{XL}}$  的时间，这里 `ODR_{\text{XL}}` 为加速度计输出数据率。

当检测到活动状态时，中断被置为高电平并持续  $1 / \text{ODR}_{\text{XL}}[\text{s}]$  的时间周期，然后自动取消。

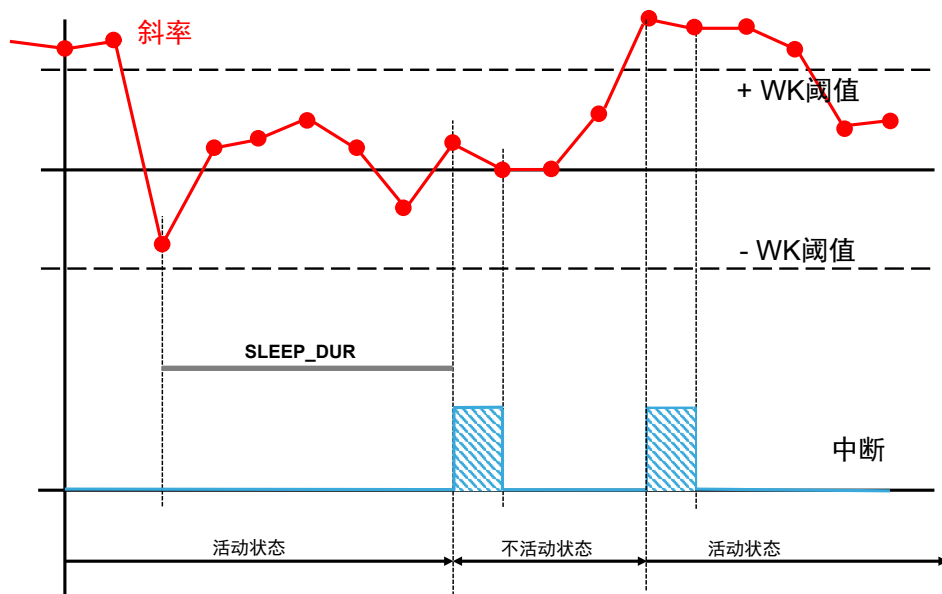
当使能活动/不活动检测功能时，通过将 `MD1_CFG` 寄存器的 `INT1_SLEEP_CHANGE` 位或 `MD2_CFG` 寄存器的 `INT2_SLEEP_CHANGE` 位置为 1，可将其状态驱动至两个中断引脚；还可通过读取 `WAKE_UP_SRC` 或 `ALL_INT_SRC` 寄存器的 `SLEEP_CHANGE_IA` 位来对其进行检查。

`SLEEP_CHANGE_IA` 位默认为脉冲模式。可通过将 `SLOPE_EN` 寄存器的 `LIR` 位置为 1 和将 `MD1_CFG` 寄存器的 `INT1_SLEEP_CHANGE` 位或 `MD2_CFG` 寄存器的 `INT2_SLEEP_CHANGE` 位置为 1 来选择锁存模式。

`WAKE_UP_SRC` 寄存器的 `SLEEP_STATE_IA` 位不受 `LIR` 配置的影响：在读取 `WAKE_UP_SRC` 寄存器时，它对应于器件的当前状态。

通过将 `SLOPE_EN` 寄存器的 `SLEEP_STATUS_ON_INT` 位置为 1，将路由至 `INT1` 或 `INT2` 引脚的信号配置为活动/不活动状态（`WAKE_UP_SRC` 寄存器的 `SLEEP_STATE` 位）而不是睡眠-变化信号：它在不活动状态时为高电平，在活动状态时为低电平。此配置不支持锁存模式。

图 6. 活动/不活动识别（利用斜率滤波器）



活动/不活动检测的基本软件例程如下所示：

1. 将 A0h 写入 CTRL1\_XL // 使能加速度计 (ODR = 26.667 kHz; FS =  $\pm 2 g$ )
2. 将 02h 写入 WAKE\_UP\_DUR // 设置不活动检测的持续时间  
// 选择活动/不活动阈值分辨率和持续时间
3. 将 02h 写入 WAKE\_UP\_THS // 设置活动/不活动阈值
4. 将 00h 写入 SLOPE\_EN // 选择睡眠-变化通知  
// 选择斜率滤波器
5. 将 80h 写入 INTERRUPTS\_EN // 使能中断
6. 将 80h 写入 MD1\_CFG // 活动/不活动中断驱动至 INT1 引脚

本例中，WAKE\_UP\_THS 寄存器的 WK\_THS 字段被置为 000010b，因此活动/不活动阈值为 62.5 mg ( $= 2 * FS_{XL} / 2^6$ ，因为 WAKE\_UP\_DUR 寄存器的 WAKE\_THS\_W 位被置为 0)。

进行不活动检测前，X、Y、Z 斜率数据必须小于所配置阈值并持续一段时间，该时间由 WAKE\_UP\_DUR 寄存器的 SLEEP\_DUR 字段定义：该字段被置为 0010b，对应 0.038 s ( $= 2 * 512 / ODR_{XL}$ )。在此之后，将检测到“不活动”状态。

一旦（至少）一个轴的斜率数据大于一个样本的阈值，就会检测到活动状态，因为 WAKE\_UP\_DUR 寄存器的 WAKE\_DUR [1:0]位配置为 00b。

## 5.4 启动状态

器件上电后，执行一段 10 ms（最长）的启动程序来加载修整参数。启动完成后，器件会自动配置为掉电模式。启动时间内，寄存器不可访问。

上电后，可通过将 CTRL3\_C 寄存器的 BOOT 位置为 1，来重载配置参数。

不需要切换设备电源线，器件控制寄存器内容不被修改，因此启动后器件工作模式不变。如果需要复位至控制寄存器的默认值，可通过将 CTRL3\_C 寄存器的 SW\_RESET 位置为 1 来实现。当此位置为 1 时，以下寄存器复位至其默认值：

- PIN\_CTRL (02h);
- FIFO\_CTRL1 (07h) 至 FIFO\_CTRL4 (0Ah) ;
- COUNTER\_BDR\_REG1 (0Bh) 和 COUNTER\_BDR\_REG2 (0Ch);
- INT1\_CTRL (0Dh) 和 INT2\_CTRL (0Eh);
- CTRL1\_XL (10h) 至 CTRL10\_C (19h) ;
- FIFO\_STATUS1 (3Ah) 和 FIFO\_STATUS2 (3Bh);
- SLOPE\_EN (56h);
- WAKE\_UP\_THS (5Bh) 和 WAKE\_UP\_DUR (5Ch) ;
- MD1\_CFG (5Eh) 和 MD2\_CFG (5Fh);
- X\_OFS\_USR (73h), Y\_OFS\_USR (74h) 和 Z\_OFS\_USR (75h)。

SW\_RESET 过程会花费 50  $\mu$ s；复位状态由 CTRL3\_C 寄存器 SW\_RESET 位的状态为信号：当复位完成时，该位自动置为低电平。

启动状态信号可通过将 INT1\_CTRL 寄存器的 INT1\_BOOT 位置为 1，来把启动状态信号驱动至 INT1 中断引脚：当正在运行启动时此信号被置为高电平，启动过程结束时此信号被重新置为低电平。

重启流程如下：

1. 将加速度计设置为下电模式；
2. 将 INT1\_CTRL 寄存器的 INT1\_BOOT 位置为 1[可选]；
3. 将 CTRL3\_C 寄存器的 BOOT 位置 1；
4. 监控重启状态，有三种可能：
  - a.等待 10 ms；
  - b.监控 INT1 引脚，直至其重新变为 0（本例中必须执行步骤 2）；
  - c.轮询 CTRL3\_C 寄存器的 BOOT 位，直至其重新变为 0。

复位流程如下：

1. 将加速度计设置为下电模式；
2. 将 CTRL3\_C 寄存器的 SW\_RESET 位设置为 1；
3. 监控软件复位状态，有两种可能：
  - a.等待 50  $\mu$ s
  - b.轮询 CTRL3\_C 寄存器的 SW\_RESET 位，直至其重新变为 0。

为了避免冲突，重启和 sw 复位不能同时执行（不要同时将 CTRL3\_C 寄存器的 BOOT 位和 SW\_RESET 位同时置为 1）。上述流程必须顺序执行。

## 5.5 时间戳

器件能够提供与传感器数据对应的时间戳信息。

为了使能该功能，必须将 CTRL10\_C 寄存器的 `TIMESTAMP_EN` 位置 1。时间戳计数器由 `TIMESTAMP3 & TIMESTAMP2 & TIMESTAMP1 & TIMESTAMP0` 寄存器输出，表示为一个 32 位无符号数字。

标称时间戳精度为 12.5  $\mu\text{s}$ 。可通过 `INTERNAL_FREQ_FINE` 寄存器的 `FREQ_FINE[7:0]` 位获取实际时间戳精度值，其中包含实际 ODR（和时间戳速率）相对于标称值的百分比差值。

$$t_{actual}[s] = \frac{1}{80000 \cdot (1 + 0.0015 \cdot FREQ\_FINE)}$$

同样地，可以使用以下公式获取实际输出数据率：

$$ODR_{actual}[Hz] = 26667 * (1 + (0.0015 * INTERNAL\_FREQ\_FINE))$$

如果加速度计处于下电模式，则时间戳计数器不工作且时间戳值冻结为最后的值。

当达到相当于约 30 小时的最大值 4294967295 LSB（等于 FFFFFFFFh）时，计数器自动复位至 00000000h 并继续计数。可通过向 `TIMESTAMP2` 寄存器中写入值 AAh，来随时将定时器计数复位至零。

在时间戳溢出条件发生前 3.2 ms，`ALL_INT_SRC` 的 `TIMESTAMP_ENDCOUNT` 位变为高电平。在读取 `ALL_INT_SRC` 寄存器时，此标记复位。还可以通过将 `MD2_CFG` 寄存器的 `INT2_TIMESTAMP` 位置为 1 将此信号路由至 INT2 引脚（37.5  $\mu\text{s}$  持续时间脉冲）。

时间戳可以在 FIFO 中进行批处理（详细信息见第 6 节 先进先出（FIFO）缓冲区）。

## 6 先进先出 (FIFO) 缓冲区

为了限制主处理器干预并简化事件识别的后处理数据，IIS3DWB 嵌入了一个 3 kB 的先进先出缓冲器 (FIFO)。

FIFO 可配置为存储以下数据：

- 加速度计传感器数据；
- 时间戳数据；
- 温度传感器数据。

FIFO 分成 512 个字，每个字 7 字节。1 个 FIFO 字包含 1 个带 TAG 信息的字节和 6 个数据字节：整个 FIFO 缓冲区大小等于 3584 字节，可包含 3072 字节的数据。TAG 字节包含表明数据保存在 FIFO 数据字段中的信息和其他有用信息。

将数据写入 FIFO 由加速度计数据就绪信号触发。可通过六个专用寄存器（从地址 79h 至 7Eh）从 FIFO 检索数据：FIFO\_DATA\_OUT\_X\_L、FIFO\_DATA\_OUT\_X\_H、FIFO\_DATA\_OUT\_Y\_L、FIFO\_DATA\_OUT\_Y\_H、FIFO\_DATA\_OUT\_Z\_L 和 FIFO\_DATA\_OUT\_Z\_H。

FIFO 流的重建是一项简单的任务，这得益于 FIFO\_DATA\_OUT\_TAG 寄存器的 FIFO\_TAG 字节，可识别 FIFO 字的含义。

利用 FIFO\_CTRL4 寄存器的 FIFO\_MODE[2:0]位，有六种不同的 FIFO 操作模式可供选择：

- Bypass 模式；
- FIFO 模式；
- Continue 模式；
- Continue-FIFO 模式；
- Bypass-Continue 模式；
- Bypass-FIFO 模式。

要监控 FIFO 状态（满，溢出，存储的采样数，等等...），可以使用两个专用寄存器：FIFO\_STATUS1 和 FIFO\_STATUS2。

可编程 FIFO 阈值可以利用 WTM[8:0]位在 FIFO\_CTRL1 和 FIFO\_CTRL2 中进行设置。

通过 INT1\_CTRL 寄存器的 INT1\_FIFO\_FULL、INT1\_FIFO\_FTH 和 INT1\_FIFO\_OVR 位，以及 INT2\_CTRL 寄存器的 INT2\_FIFO\_FULL、INT2\_FIFO\_FTH 和 INT2\_FIFO\_OVR 位，可以使能 FIFO 满、FIFO 阈值和 FIFO 溢出事件，在两个中断引脚（INT1 和 INT2）上产生专门的中断。

### 6.1 FIFO 寄存器

FIFO 缓冲器由以下几项管理：

- 六个控制寄存器：FIFO\_CTRL1、FIFO\_CTRL2、FIFO\_CTRL3、FIFO\_CTRL4、COUNTER\_BDR\_REG1 和 COUNTER\_BDR\_REG2；
- 两个状态寄存器：FIFO\_STATUS1 和 FIFO\_STATUS2；
- 七个输出寄存器（标签 + 数据）：FIFO\_DATA\_OUT\_TAG、FIFO\_DATA\_OUT\_X\_L、FIFO\_DATA\_OUT\_X\_H、FIFO\_DATA\_OUT\_Y\_L、FIFO\_DATA\_OUT\_Y\_H、FIFO\_DATA\_OUT\_Z\_L 和 FIFO\_DATA\_OUT\_Z\_H；
- 一些额外的位，将 FIFO 事件路由至两条中断线路：INT1\_CTRL 寄存器的 INT1\_CNT\_BDR、INT1\_FIFO\_FULL、INT1\_FIFO\_OVR 和 INT1\_FIFO\_TH 位，以及 INT2\_CTRL 寄存器的 INT2\_CNT\_BDR、INT2\_FIFO\_FULL、INT2\_FIFO\_OVR 和 INT2\_FIFO\_TH 位。

### 6.1.1 FIFO\_CTRL1

FIFO\_CTRL1 寄存器包含 9 位 FIFO 水印阈值的低位部分。对于完整的水印阈值配置，还要考虑 FIFO\_CTRL2 寄存器的 WTM8 位。FIFO 阈值的 1 LSB 值称为一个 FIFO 字（7 字节）。

当 FIFO 中存储的字节数大于或等于水印阈值时，FIFO 水印标记（FIFO\_STATUS2 寄存器的 FIFO\_WTM\_IA 位）升高。

要将 FIFO 深度限制为水印级别，FIFO\_CTRL2 寄存器中 STOP\_ON\_WTM 位必须设置为 1。

**表 10. FIFO\_CTRL1 寄存器**

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
WTM7	WTM6	WTM5	WTM4	WTM3	WTM2	WTM1	WTM0

### 6.1.2 FIFO\_CTRL2

**表 11. FIFO\_CTRL2 寄存器**

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
STOP_ON_WTM	0	0	0	0	0	0	WTM8

FIFO\_CTRL2 寄存器包含 9 位 FIFO 水印阈值的高位部分（WTM8 位）。对于完整的水印阈值配置，还要考虑 FIFO\_CTRL1 寄存器的 WTM[7:0]位。寄存器包含了位 STOP\_ON\_WTM，可将 FIFO 深度限制在水印级别。

### 6.1.3 FIFO\_CTRL3

**表 12. FIFO\_CTRL3 寄存器**

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
0	0	0	0	BDR_XL_3	BDR_XL_2	BDR_XL_1	BDR_XL_0

FIFO\_CTRL3 寄存器包含在 FIFO 中选择写入频率的字段。

下表列出了 BDR\_XL [3:0]位的允许配置；其他配置值不得使用。

**表 13. 加速度计批处理数据率**

BDR_XL[3:0]	批处理数据率[Hz]
0000	FIFO 中不批处理
1010	26667

### 6.1.4 FIFO\_CTRL4

FIFO\_CTRL4 寄存器包含用于选择 FIFO 中时间戳批处理抽取因子和温度传感器批处理数据率的字段。

时间戳写入速率对应加速度计批处理数据速率除以 DEC\_TS\_BATCH\_[1:0] 字段中指定的抽取因子。下表显示了可编程抽取因子。

**表 14. 时间戳批处理数据率**

DEC_TS_BATCH[1:0]	时间戳批处理数据率[Hz]
00	FIFO 中不批处理（默认）
01	BDR_XL[Hz]（不抽取）
10	BDR_XL[Hz] / 8
11	BDR_XL[Hz] / 32

温度批处理数据率可通过 ODR\_T\_BATCH\_[1:0] 字段配置，如下表所示。

**表 15. 温度传感器批处理数据率**

ODR_T_BATCH[1:0]	温度批处理数据率[Hz]
00	FIFO 中不批处理（默认）
11	104

FIFO\_CTRL4 寄存器还包含 FIFO 操作模式位。FIFO 操作模式如第 6.3 节 FIFO 模式中所示。

**表 16. FIFO\_CTRL4 寄存器**

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DEC_TS_BATCH_1	DEC_TS_BATCH_0	ODR_T_BATCH_1	ODR_T_BATCH_0	0	FIFO_MODE2	FIFO_MODE1	FIFO_MODE0

## 6.1.5 COUNTER\_BDR\_REG1

由于 FIFO 可能包含加速度计、时间戳和温度数据，因此 FIFO 提供了一种方法，可根据 FIFO 中存储的加速度计实际采样数（BDR 计数器）来同步 FIFO 读数。

BDR 计数器可通过 COUNTER\_BDR\_REG1 和 COUNTER\_BDR\_REG2 寄存器进行配置。

表 17. COUNTER\_BDR\_REG1 寄存器

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
-	RST_COUNTER_BDR	0	0	0	CNT_BDR_TH_10	CNT_BDR_TH_9	CNT_BDR_TH_8

可以将 RST\_COUNTER\_BDR 置位以便复位 BDR 计数器：它将自动复位至零。

用户可以在 FIFO\_STATUS2 寄存器中选择生成 COUNTER\_BDR\_IA 事件的阈值。在内部 BDR 计数器达到阈值后，将 COUNTER\_BDR\_IA 位置为 1。阈值可通过 CNT\_BDR\_TH\_[10:0] 位进行配置。字段上部包含在寄存器 COUNTER\_BDR\_REG1 中。CNT\_BDR\_TH 阈值的 1 LSB 值称为一个加速度计采样（X、Y 和 Z 数据）。

## 6.1.6 COUNTER\_BDR\_REG2

COUNTER\_BDR\_REG2 寄存器包含 BDR 计数器阈值的下部。

表 18. COUNTER\_BDR\_REG2 寄存器

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
CNT_BDR_TH_7	CNT_BDR_TH_6	CNT_BDR_TH_5	CNT_BDR_TH_4	CNT_BDR_TH_3	CNT_BDR_TH_2	CNT_BDR_TH_1	CNT_BDR_TH_0

## 6.1.7 FIFO\_STATUS1

FIFO\_STATUS1 寄存器，连同 FIFO\_STATUS2 寄存器一起，提供 FIFO 中存储的采样数相关信息。DIFF\_FIFO 的 1 LSB 值称为一个 FIFO 字（7 字节）。

表 19. FIFO\_STATUS1 寄存器

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DIFF_FIFO_7	DIFF_FIFO_6	DIFF_FIFO_5	DIFF_FIFO_4	DIFF_FIFO_3	DIFF_FIFO_2	DIFF_FIFO_1	DIFF_FIFO_0



### 6.1.8 FIFO\_STATUS2

FIFO\_STATUS2 寄存器，连同 FIFO\_STATUS1 寄存器一起，可提供 FIFO 中存储的采样数相关信息和 FIFO 缓冲器当前状态信息（水印，溢出，满，BDR 计数器）。

**表 20. FIFO\_STATUS2 寄存器**

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
FIFO_WTM_IA	FIFO_OVR_IA	FIFO_FULL_IA	COUNTER_BDR_IA	FIFO_OVR_LATCHED	0	DIFF_FIFO_9	DIFF_FIFO_8

- FIFO\_WTM\_IA 表示水印状态。当 FIFO 中已存储的 FIFO 字（每个 7 字节）数量大于等于水印阈值水平时，此位变为高电平。通过将 INT1\_CTRL 寄存器的 INT1\_FIFO\_TH 位或 INT2\_CTRL 寄存器的 INT2\_FIFO\_TH 位置为 1，可将水印状态信号驱动至两个中断引脚上。
- 当 FIFO 完全填满，至少一个采样已经被覆盖掉以存储新数据时，FIFO\_OVR\_IA 变为高电平。通过将 INT1\_CTRL 寄存器的 INT1\_FIFO\_OVR 位或 INT2\_CTRL 寄存器的 INT2\_FIFO\_OVR 位置为 1，可将此信号驱动至两个中断引脚上。
- 当要存储在 FIFO 中的下一组数据会使 FIFO 全满（即，DIFF\_FIFO\_9 = 1）或生成 FIFO 溢出时，FIFO\_FULL\_IA 被置为高电平。通过将 INT1\_CTRL 寄存器的 INT1\_FIFO\_FULL 位或 INT2\_CTRL 寄存器的 INT2\_FIFO\_FULL 位置为 1，可将此信号驱动至两个中断引脚上。
- COUNTER\_BDR\_IA 表示 BDR 计数器状态。当加速度计批处理采样数达到通过 COUNTER\_BDR\_REG1 和 COUNTER\_BDR\_REG2 寄存器的 CNT\_BDR\_TH [10:0] 位配置的 BDR 计数器阈值时，此位被置为高电平。在读取 FIFO\_STATUS2 寄存器时 COUNTER\_BDR\_IA 位自动复位。通过将 INT1\_CTRL 寄存器的 INT1\_CNT\_BDR 位或 INT2\_CTRL 寄存器的 INT2\_CNT\_BDR 位置为 1，可将 BDR 计数器状态驱动至两个中断引脚上。
- 当 FIFO 完全填满，至少一个采样已经被覆盖掉以存储新数据时，FIFO\_OVR\_LATCHED（如同 FIFO\_OVR\_IA 一样）变为高电平。两个标记之间的差异在于，FIFO\_OVR\_LATCHED 在读取 FIFO\_STATUS2 寄存器时复位，而 FIFO\_OVR\_IA 在读取至少一个 FIFO 字时复位。这样就可以在从 FIFO 读取数据时检测 FIFO 溢出条件。
- DIFF\_FIFO [9:8] 包含 FIFO 中存储的未读字数的高位部分。其低位部分由 FIFO\_STATUS1 中的 DIFF\_FIFO [7:0] 位表示。DIFF\_FIFO [9:0] 字段的值对应 FIFO 中的 7 字节字的数量。

寄存器内容会与 FIFO 写操作和读操作同步更新。

*注：BDU 功能还作用于 FIFO\_STATUS1 和 FIFO\_STATUS2 寄存器。当 BDU 位置为 1 时，必须首先读取 FIFO\_STATUS1，然后读取 FIFO\_STATUS2。*

## 6.1.9 FIFO\_DATA\_OUT\_TAG

通过读取 FIFO\_DATA\_OUT\_TAG 寄存器，可以知道当前读取的数据属于哪个传感器并检查数据是否一致。

表 21. FIFO\_DATA\_OUT\_TAG 寄存器

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
TAG_SENSOR_4	TAG_SENSOR_3	TAG_SENSOR_2	TAG_SENSOR_1	TAG_SENSOR_0	TAG_CNT_1	TAG_CNT_0	TAG_PARITY

- TAG\_SENSOR\_[4:0] 字段标识 6 个数据字节中保存的传感器（表 22）；
- TAG\_CNT\_[1:0] 字段标识 FIFO 时隙（如后面的章节所述）；
- TAG\_PARITY 位标识 FIFO\_DATA\_OUT\_TAG 寄存器的内容是否损坏。

下表包含了 TAG\_SENSOR\_[4:0] 字段的所有可能值和相关传感器类型。

表 22. TAG\_SENSOR 字段和相关传感器

TAG_SENSOR_[4:0]	传感器名称
0x02	加速度计
0x03	温度
0x04	时间戳

TAG\_PARITY 位可用于检查 FIFO\_DATA\_OUT\_TAG 寄存器的内容。为此，用户可以执行以下程序：

1. 读取 FIFO\_DATA\_OUT\_TAG 寄存器；
2. 对等于 1 的位进行计数；
3. 如果等于 1 的位数为偶数，则 FIFO\_DATA\_OUT\_TAG 内容可靠，否则不可靠。

## 6.1.10 FIFO\_DATA\_OUT

可通过六个专用寄存器（从地址 79h 至地址 7Eh）从 FIFO 检索数据：FIFO\_DATA\_OUT\_X\_L、FIFO\_DATA\_OUT\_X\_H、FIFO\_DATA\_OUT\_Y\_L、FIFO\_DATA\_OUT\_Y\_H、FIFO\_DATA\_OUT\_Z\_L 和 FIFO\_DATA\_OUT\_Z\_H。

FIFO 输出寄存器的内容取决于传感器类别和类型，如下一节所述。

## 6.2 FIFO 批处理传感器

如前所述，FIFO 可配置为存储以下数据：

- 加速度计传感器数据；
- 时间戳数据；
- 温度传感器数据。

本节将详细介绍批处理传感器。

### 6.2.1 FIFO 中的加速度计数据

可以通过 FIFO\_CTRL3 寄存器的 BDR\_XL\_[3: 0]位来配置加速度计批处理数据速率。

加速度传感器定义 FIFO 时基。这意味着可以将存储在 FIFO 中的各项其他数据（时间戳、温度）与加速度传感器定义的时基时隙相关联。加速度传感器的批处理事件会使 TAG 计数器（FIFO\_DATA\_OUT\_TAG 寄存器的 TAG\_CNT 字段）递增：该计数器由两位组成，其值连续递增（从 00b 到 11b），以标识不同的时隙。

下表显示了加速度计传感器的 FIFO 字格式，代表了从 78h 至 7Eh 的设备地址。

**表 23. FIFO 中的主要传感器输出数据格式**

标记	X_L	X_H	Y_L	Y_H	Z_L	Z_H
----	-----	-----	-----	-----	-----	-----

### 6.2.2 FIFO 中时间戳和温度数据

除加速度计数据外，还可将以下传感器的数据存储在 FIFO 中：

- 温度传感器（必须正确地配置 FIFO\_CTRL4 寄存器的 ODR\_T\_BATCH\_[1:0]位）；
- 时间戳传感器：它保存对应于 FIFO 时隙的时间戳（必须正确地配置 CTRL10\_C 寄存器的 TIMESTAMP\_EN 位和 FIFO\_CTRL4 寄存器的 DEC\_TS\_BATCH\_[1:0]位）。

温度和时间戳传感器无法触发 FIFO 写入。当加速度计数据就绪信号事件发生时，其寄存器被写入 FIFO。

下表显示了 FIFO 中的温度输出数据格式。

**表 24. FIFO 中的温度输出数据格式**

数据	FIFO_DATA_OUT 寄存器
TEMPERATURE[7:0]	FIFO_DATA_OUT_X_L
TEMPERATURE[15:8]	FIFO_DATA_OUT_X_H
0	FIFO_DATA_OUT_Y_L
0	FIFO_DATA_OUT_Y_H
0	FIFO_DATA_OUT_Z_L
0	FIFO_DATA_OUT_Z_H

下表显示了 FIFO 中的时间戳输出数据格式。

**表 25. FIFO 中的时间戳输出数据格式**

数据	FIFO_DATA_OUT 寄存器
TIMESTAMP[7:0]	FIFO_DATA_OUT_X_L
TIMESTAMP[15:8]	FIFO_DATA_OUT_X_H
TIMESTAMP[23:16]	FIFO_DATA_OUT_Y_L
TIMESTAMP[31:24]	FIFO_DATA_OUT_Y_H
0	FIFO_DATA_OUT_Z_L[3:0]
0	FIFO_DATA_OUT_Z_L[7:4]
BDR_XL	FIFO_DATA_OUT_Z_H[3:0]
0	FIFO_DATA_OUT_Z_H[7:4]

## 6.3 FIFO 模式

通过 FIFO\_CTRL4 寄存器的 FIFO\_MODE\_[2:0] 字段，IIS3DWB 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。

以下段落描述了 Bypass、FIFO、Continuous、Continuous-FIFO、Bypass-Continuous 和 Bypass-FIFO 模式。

### 6.3.1 Bypass 模式

使能 Bypass 模式时，FIFO 不使用，缓冲器内容被清零，并保持为空，直至选择了另一种模式。当 FIFO\_MODE\_[2:0] 位被置为 000b 时，选用 Bypass 模式。当准备使用其他模式时，为了停止和复位 FIFO 缓冲器，必须使用 Bypass 模式。注意，将 FIFO 缓冲器置于 Bypass 模式时，整个缓冲器内容会被清零。

### 6.3.2 FIFO 模式

FIFO 模式中，缓冲器继续填充直至填满。然后停止刷新数据，FIFO 内容保持不变，直至选用不同模式。

请按照以下步骤配置 FIFO 模式：

1. 使传感器数据以相应的批处理数据速率存储在 FIFO 中；
2. 将 FIFO\_CTRL4 寄存器中的 FIFO\_MODE\_[2:0] 位置为 001b 来使能 FIFO 模式。

当选用此模式时，FIFO 开始采集数据。FIFO\_STATUS1 和 FIFO\_STATUS2 寄存器根据所存储的采样数来更新。

当 FIFO 已满时，FIFO\_STATUS2 寄存器的 DIFF\_FIFO\_9 位被置为 1，不再有数据存储到 FIFO 缓冲区中。可以通过读取所有 FIFO\_DATA\_OUT 寄存器（78h 至 7Eh）并达到 FIFO\_STATUS1 和 FIFO\_STATUS2 寄存器 DIFF\_FIFO\_[9:0] 位指定的次数来获取数据。

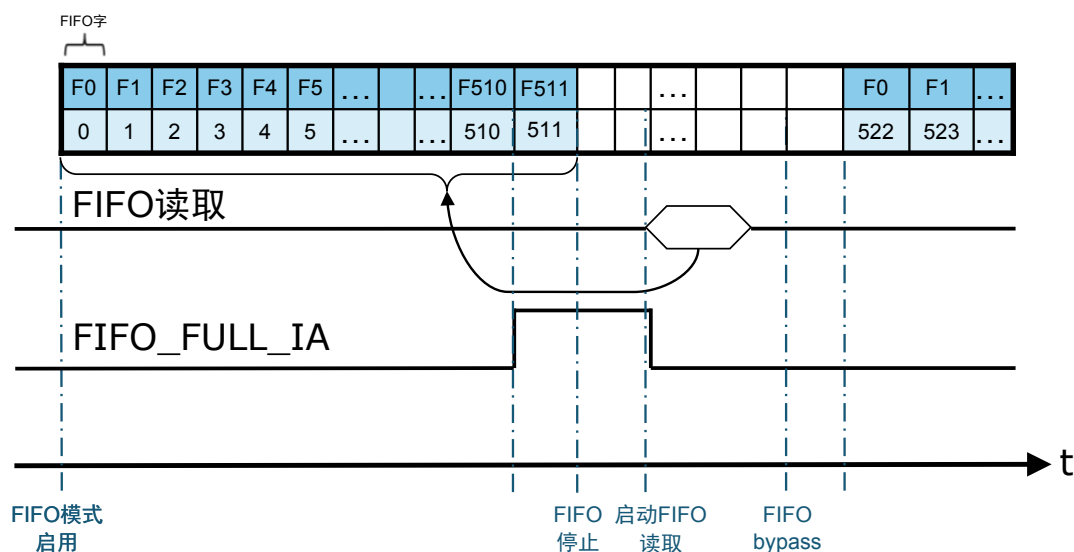
利用 FIFO\_STATUS2 寄存器的 FIFO\_WTM\_IA 位，如果应用要求 FIFO 中采样数较低，则数据还可在达到阈值（FIFO\_CTRL1 和 FIFO\_CTRL2 寄存器中的 WTM[8:0]）时恢复。

如果 FIFO\_CTRL2 寄存器的 STOP\_ON\_WTM 位置为 1，则 FIFO 空间大小被限制为 FIFO\_CTRL1 和 FIFO\_CTRL2 寄存器中 WTM[8:0] 位的值。这种情况下，如果下一次 FIFO 写操作时 FIFO 中的采样数达到或超过 WTM[8:0] 值，那么 FIFO\_STATUS2 寄存器的 FIFO\_FULL\_IA 位会被置为高电平。

FIFO 模式下通信速度不是很重要，因为数据采集已停止，不存在已采集数据被覆盖的风险。重启 FIFO 模式之前，需要首先设置为 Bypass 模式，以完全清空 FIFO 内容。

图 7. FIFO 模式（STOP\_ON\_WTM = 0）显示了 FIFO 模式使用示例；在示例中，只有加速度计的数据存储在 FIFO 中。这些条件下，FIFO 缓冲区中可存储 512 个加速度计采样。在标记为 510 后，FIFO\_STATUS2 寄存器的 FIFO\_FULL\_IA 位升高，以通知 FIFO 缓冲区将在下一次 FIFO 写操作时达到全满。在 FIFO 已满（FIFO\_DIFF\_9 = 1）后，数据采集停止。

图 7. FIFO 模式（STOP\_ON\_WTM = 0）



### 6.3.3 Continuous 模式

Continue 模式中，FIFO 连续填充。当缓冲器满时，FIFO 索引重新从头开始，原有数据被新数据替代。最早先的数据继续被覆盖，直至读取操作释放了 FIFO 空间。要实现空间释放速度快于新数据产生速度，主处理器读取速度很重要。要停止此配置，必须选用 Bypass 模式。

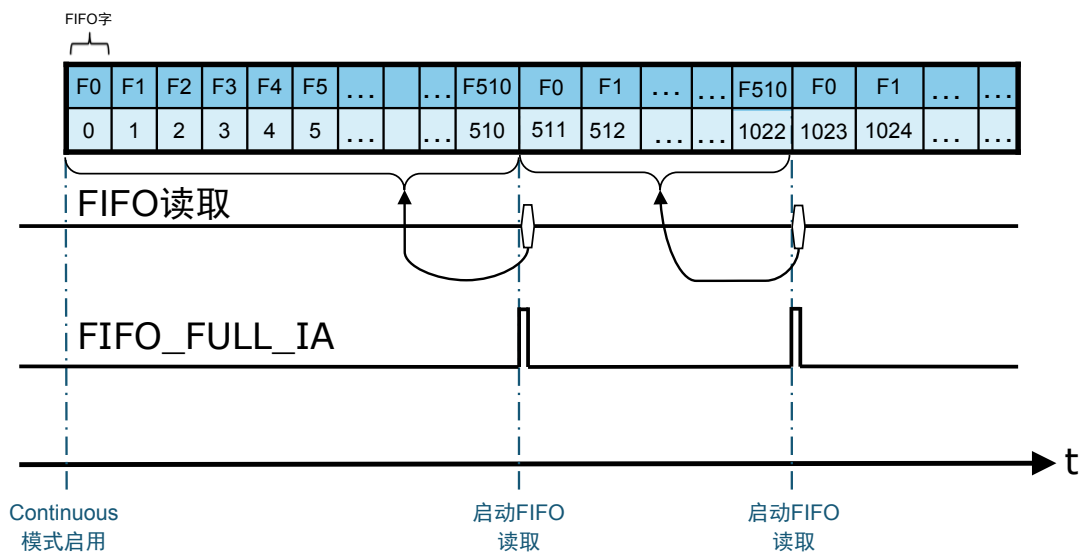
请按照以下步骤配置 Continuous 模式：

1. 使传感器数据以相应的批处理数据速率存储在 FIFO 中；
2. 将 FIFO\_CTRL4 寄存器中的 FIFO\_MODE\_[2:0] 位置为 110b 来使能 FIFO 模式。

当选用此模式时，FIFO 连续采集数据。FIFO\_STATUS1 和 FIFO\_STATUS2 寄存器根据所存储的采样数来更新。当下一个 FIFO 写操作会使 FIFO 全满或生成 FIFO 溢出时，FIFO\_STATUS2 寄存器的 FIFO\_FULL\_IA 位变为 1。FIFO\_STATUS2 寄存器的 FIFO\_OVR\_IA 和 FIFO\_OVR\_LATCHED 位表示至少有一个 FIFO 字被覆盖以存储新数据。通过读取 FIFO\_DATA\_OUT (78h 至 7Eh) 寄存器获取 FIFO\_STATUS1 和 FIFO\_STATUS2 寄存器的 DIFF\_FIFO\_[9:0] 位所指定的次数，数据可在 FIFO\_FULL\_IA 事件后恢复。利用 FIFO\_STATUS2 寄存器的 FIFO\_WTM\_IA 位，数据还可在达到阈值 (FIFO\_CTRL1 和 FIFO\_CTRL2 寄存器中的 WTM[8:0]) 时恢复。如果 FIFO\_CTRL2 寄存器的 STOP\_ON\_WTM 位置为 1，则 FIFO 空间大小被限制为 FIFO\_CTRL1 和 FIFO\_CTRL2 寄存器中 WTM[8:0] 位的值。这种情况下，如果下一次 FIFO 写操作时 FIFO 中的采样数达到或超过 WTM[8:0] 值，那么 FIFO\_STATUS2 寄存器的 FIFO\_FULL\_IA 位会被置为高电平。

图 8. Continuous 模式 显示了 Continue 模式使用的示例。示例中，仅加速度计传感器的数据存储在 FIFO 中，在发生 FIFO\_FULL\_IA 事件时读取 FIFO 采样且读取速度快于  $1 * ODR$ ，因此无数据丢失。这些条件下，所存储的采样数为 511。

图 8. Continuous 模式



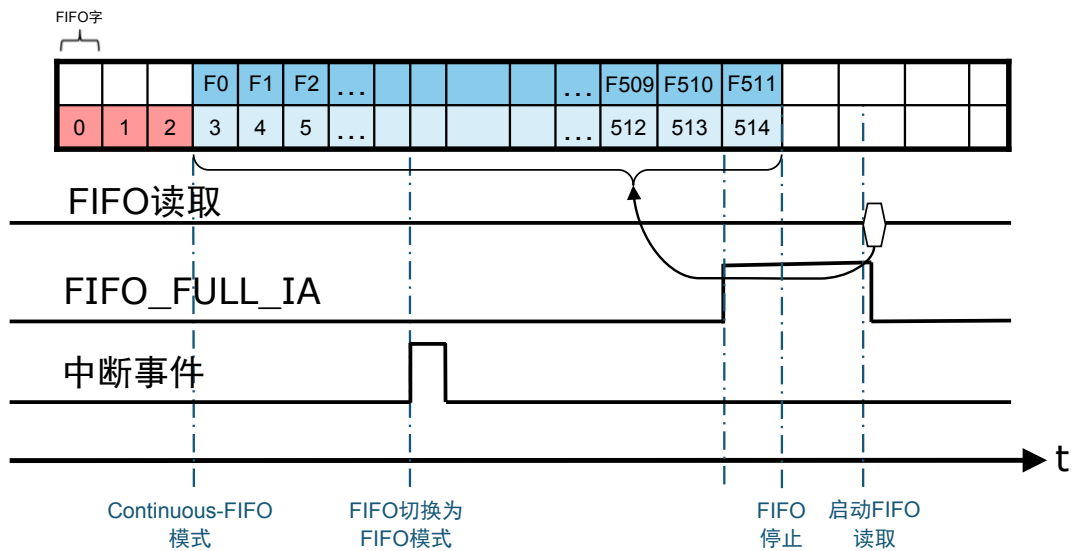
### 6.3.4 Continue-FIFO 模式

此模式是先前所述的 Continue 和 FIFO 模式的组合。在 Continue-FIFO 模式中，FIFO 缓冲器开始工作于 Continue 模式，当发生事件条件时切换为 FIFO 模式。必须正确配置唤醒事件检测，并且 MD2\_CFG 寄存器的 INT2\_WU 位必须被置为 1。

事件条件可为以下之一：

Continuous-to-FIFO 模式对中断信号的边沿感应。在第一次中断事件时，FIFO 从 Continuous 模式变为 FIFO 模式，并维持该模式，直到被设置成 bypass 模式。

图 9. Continue-FIFO 模式



请按照以下步骤配置 Continue-FIFO 模式：

1. 如前所述配置唤醒事件检测；
2. 使传感器数据以相应的批处理数据速率存储在 FIFO 中；
3. 将 FIFO\_CTRL4 寄存器中 FIFO\_MODE\_[2:0] 位置为 011b 来使能 Continue-FIFO 模式。

在 Continuous-FIFO 模式下，FIFO 缓冲区会继续填入数据。当 FIFO 将在下一次 FIFO 写操作时全满或溢出时，FIFO\_FULL\_IA 位变成‘1’。

如果 FIFO\_CTRL2 寄存器的 STOP\_ON\_WTM 位置为 1，则 FIFO 空间大小被限制为 FIFO\_CTRL1 和 FIFO\_CTRL2 寄存器中 WTM[8:0] 位的值。这种情况下，如果下一次 FIFO 写操作时 FIFO 中的采样数达到或超过 WTM[8:0] 值，那么 FIFO\_STATUS2 寄存器的 FIFO\_FULL\_IA 位会被置为高电平。

发生触发事件时，可观察到两种不同的情况：

1. 如果 FIFO 缓冲器已满，则事件触发后第一次采样时即停止采集数据。FIFO 内容由该事件之前所采集的采样组成。
2. 如果 FIFO 尚未满，则继续填充直至填满，然后停止采集数据。

Continuous-FIFO 用来分析生成唤醒中断的采样历史。标准操作是在 FIFO 模式已触发、FIFO 缓冲区已满并停止时读取 FIFO 内容。

### 6.3.5 Bypass-Continue 模式

此模式是先前所述的 **Bypass** 和 **Continue** 模式的组合。在 **Bypass-Continuous** 模式中，FIFO 缓冲器开始工作于 **Bypass** 模式，当发生唤醒事件条件时切换为 **Continuous** 模式。必须正确配置唤醒事件检测，并且 MD2\_CFG 寄存器的 INT2\_WU 位必须被置为 1。

**Bypass-Continue** 模式对中断信号的边沿感应：在第一次中断事件时，FIFO 从 **Bypass** 模式切换到 **Continuous** 模式，并维持该模式，直到 **Bypass** 模式被置位。

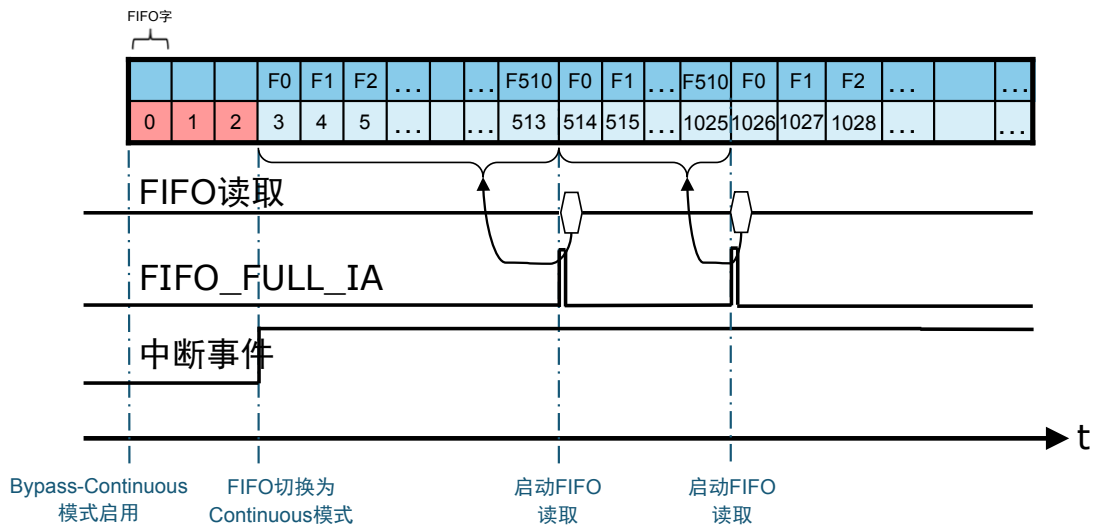
请按照以下步骤配置 **Bypass-Continue** 模式：

1. 配置唤醒事件检测如前所述；
2. 使能以相应批处理数据率将传感器数据保存到 FIFO 中；
3. 将 FIFO\_CTRL4 寄存器中 FIFO\_MODE[2:0]位置为 100b 来使能 FIFO 旁路-连续模式。

当出现触发条件且缓冲器切换至 **Continuous** 模式时，FIFO 缓冲器继续填充。当下一个要存储的数据集将会使 FIFO 满或溢出时，FIFO\_FULL\_IA 位被置为‘1’。

**Bypass-Continue** 可用在产生所配置中断时启动采集。

**图 10. Bypass-Continue 模式**





### 6.3.6 Bypass-FIFO 模式

此模式是先前所述的 Bypass 和 FIFO 模式的组合。在 Bypass-FIFO 模式中，FIFO 缓冲器开始工作于 Bypass 模式，当发生事件条件时切换为 FIFO 模式。必须正确配置唤醒事件检测，并且 MD2\_CFG 寄存器的 INT2\_WU 位必须被置为 1。

Bypass-to-FIFO 模式对中断信号的边沿感应。在第一次中断事件时，FIFO 从 Bypass 模式变为 FIFO 模式，并维持该模式，直到 Bypass 模式被置位。

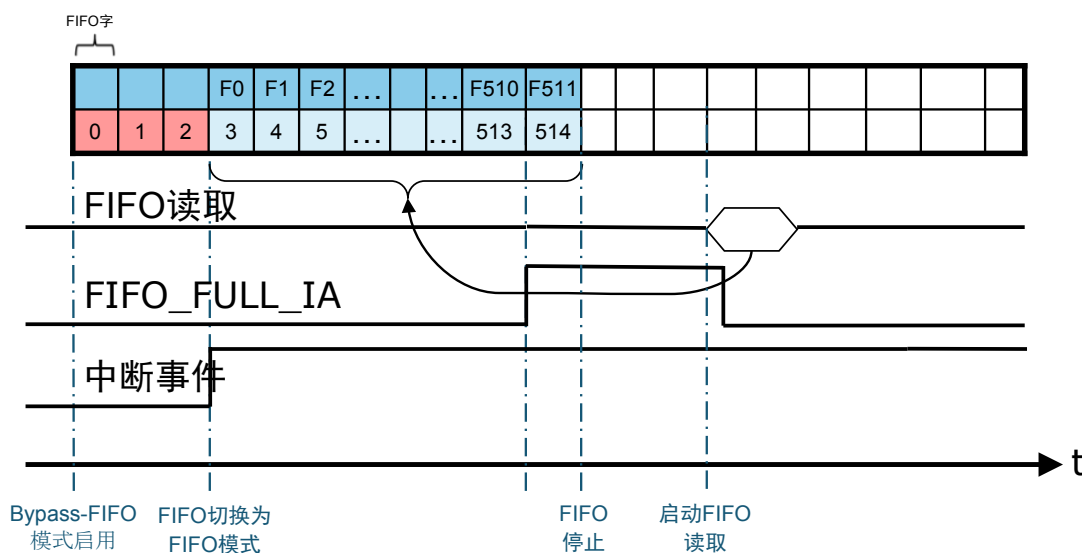
请按照以下步骤配置 Bypass-to-FIFO 模式：

1. 配置唤醒事件检测如前所述；
2. 使能以相应批处理数据率将传感器数据保存到 FIFO 中；
3. 将 FIFO\_CTRL4 寄存器中 FIFO\_MODE\_[2:0]位置为 111b 来使能 Bypass-FIFO 模式。

当出现触发条件且缓冲器切换至 FIFO 模式时，FIFO 缓冲器开始填充。当下一个要存储的数据集将会使 FIFO 满或溢出时，FIFO\_FULL\_IA 位被置为高电平且 FIFO 停止。

Bypass-to-FIFO 可用来分析生成中断的采样历史。

图 11. Bypass-FIFO 模式



## 6.4 从 FIFO 检索数据

当 FIFO 得到使能且其模式不是 Bypass 模式时，读取 FIFO 输出寄存器会返回原 FIFO 采样集。无论何时读取这些寄存器，其内容均会移动到 SPI/I<sup>2</sup>C 输出缓冲器。

理想地，FIFO 插槽会向上移动一格，以便释放空间给新的采样，并且 FIFO 输出寄存器载入 FIFO 缓冲器中存储的当前最旧的值。

从 FIFO 中恢复数据的建议方法如下：

1. 读取 FIFO\_STATUS1 和 FIFO\_STATUS2 寄存器来检查 FIFO 中存储了多少字。此信息在 DIFF\_FIFO\_[9:0]位中。
2. 对于 FIFO 中的每个字，读取 FIFO 字（标记和输出数据）并基于 FIFO 标记解读它。
3. 转至步骤 1。

通过从 FIFO 输出寄存器执行一定次数的读操作，直至缓冲器为空（FIFO\_STATUS1 和 FIFO\_STATUS2 寄存器的 DIFF\_FIFO\_[9:0]位等于 0），可以恢复全部 FIFO 内容。

建议避免在 FIFO 为空时读取它。

必须按 7 字节的倍数读取 FIFO 输出数据，从 FIFO\_DATA\_OUT\_TAG 寄存器开始读取。为了通过一次多字节读取操作读取许多字，在器件中自动执行从地址 FIFO\_DATA\_OUT\_Z\_H 至 FIFO\_DATA\_OUT\_TAG 的环行功能。

## 6.5 FIFO 水印阈值

FIFO 阈值是 IIS3DWB FIFO 的功能，可用于检查 FIFO 中的采样数何时达到定义的水印阈值水平。

FIFO\_CTRL1 和 FIFO\_CTRL2 寄存器的 WTM[8:0] 中有水印阈值水平。WTM[8:0] 字段的分辨率为 7 字节，相当于一个完整的 FIFO 字。因此，用户能够在 0 至 511 之间选择所需的值。

FIFO\_STATUS2 寄存器的位 FIFO\_WTM\_IA 表示水印状态。如果 FIFO 中的字数达到或超过水印水平，则此位被置为高电平。通过将 FIFO\_CTRL2 寄存器的 STOP\_ON\_WTM 位置为 1，FIFO 空间大小可由阈值水平来限制。

图 12. FIFO 阈值 (STOP\_ON\_WTM = 0)

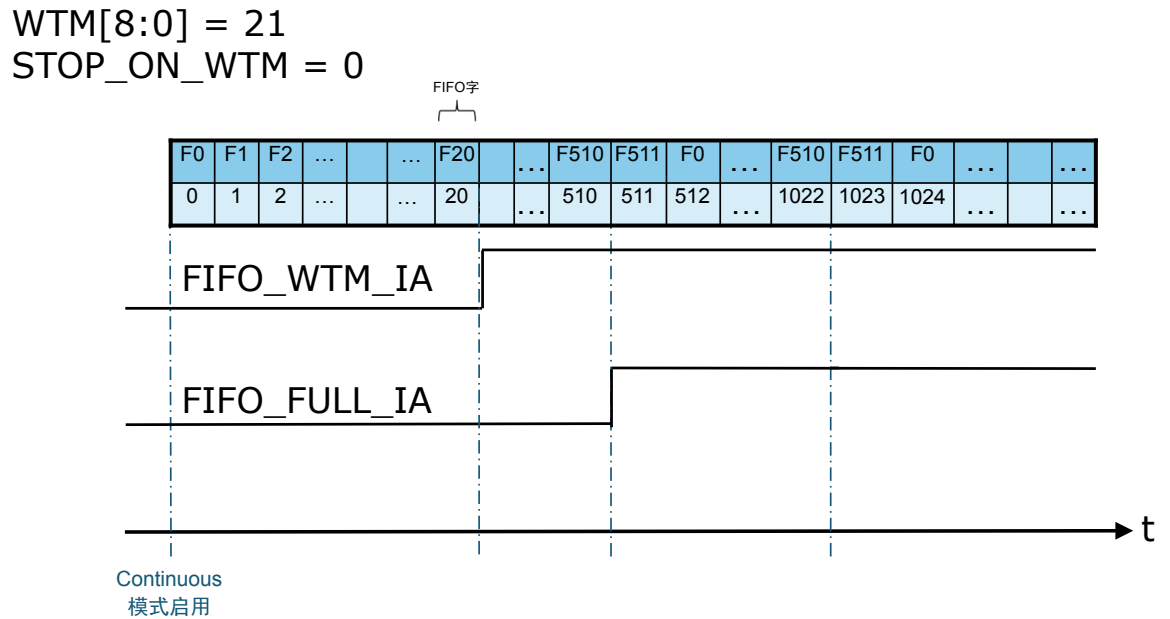


图 12. FIFO 阈值 (STOP\_ON\_WTM = 0) 显示了只存储加速度计数据时，FIFO 阈值水平使用的示例。

FIFO\_CTRL2 寄存器中 STOP\_ON\_WTM 位置为 0。利用 WTM[8:0] 位，阈值水平设置为 21。达到 21 后 (FIFO 中有 21 个字)，FIFO\_STATUS2 寄存器的 FIFO\_WTM\_IA 位升高。因此，STOP\_ON\_WTM 位置 0，FIFO 不会在第 21 个数据组时停止，而是会继续存储数据，直至 FIFO\_FULL\_IA 标记被置为 '1'。

图 13. FIFO 模式下的 FIFO 阈值 (STOP\_ON\_WTM = 1)

WTM[8:0] = 21  
STOP\_ON\_WTM = 1

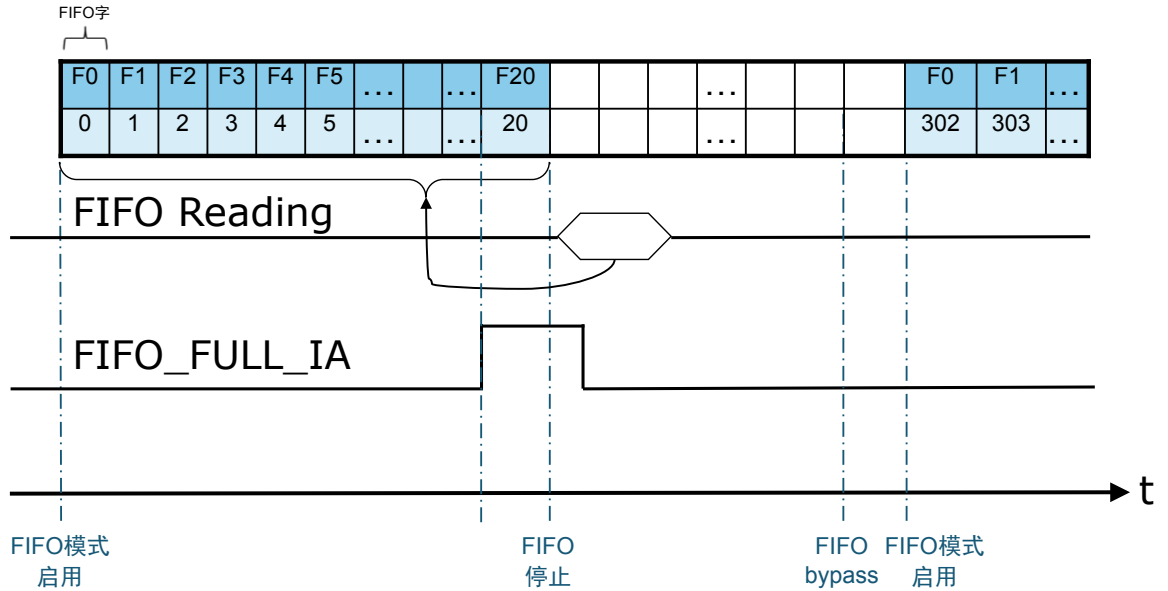


图 13. FIFO 模式下的 FIFO 阈值 (STOP\_ON\_WTM = 1) 显示了 FIFO 模式下 FIFO 阈值使用的示例，其中 FIFO\_CTRL2 寄存器中 STOP\_ON\_WTM 位被置为 1。本例中只存储了加速度计数据。利用 WTM[8:0]位，阈值水平设置为 21，并定义当前 FIFO 空间大小。在 FIFO 模式下，数据保存在 FIFO 缓冲区，直至 FIFO 已满。在 FIFO 中保存下一个数据时，FIFO\_STATUS2 寄存器的 FIFO\_FULL\_IA 位升高，将生成 FIFO 已满或溢出条件。当 FIFO 已满时，FIFO\_STATUS2 寄存器的 FIFO\_WTM\_IA 位升高。

图 14. Continue 模式下的 FIFO 阈值 (STOP\_ON\_WTM = 1)

WTM[8:0] = 21  
STOP\_ON\_WTM = 1

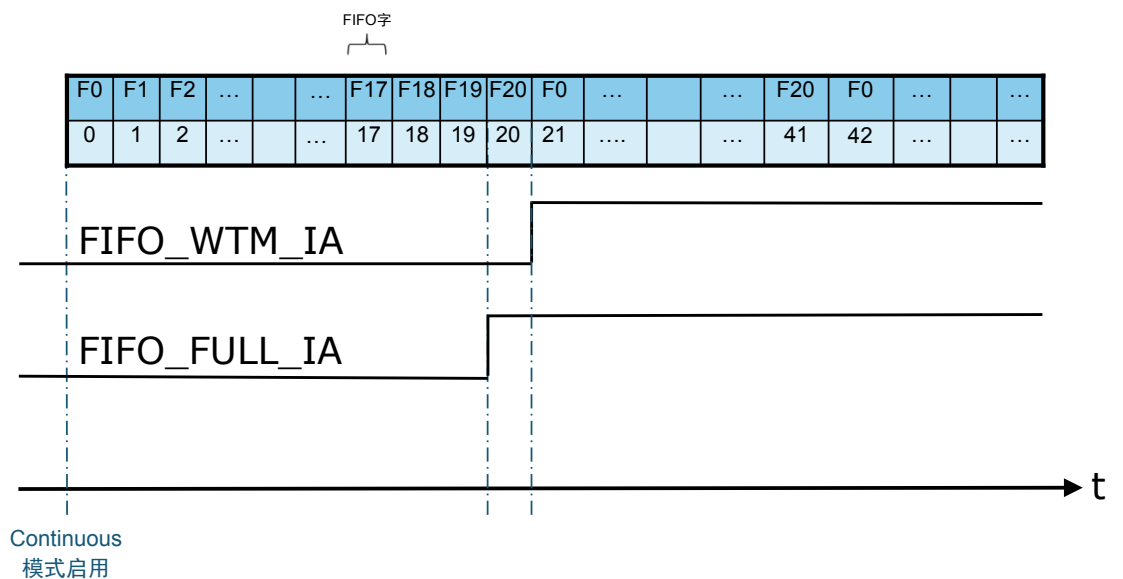


图 14. Continue 模式下的 FIFO 阈值 (STOP\_ON\_WTM = 1) 显示了 Continuous 模式下 FIFO 阈值使用的示例，其中 FIFO\_CTRL2 寄存器的 STOP\_ON\_WTM 位被置为 1。本例中只存储了加速度计数据。利用 WTM[8:0]位，阈值水平设置为 21。在 FIFO 中保存下一个数据时，FIFO\_STATUS2 寄存器的 FIFO\_FULL\_IA 位升高，FIFO 已满。当 FIFO 已满时，FIFO\_STATUS2 寄存器的 FIFO\_WTM\_IA 位升高。如果不从 FIFO 检索数据，新数据 (标记为采样 21) 将覆盖 FIFO 中的旧数据 (标记为采样 F0)。

## 6.6 时间戳相关性

可通过三种不同方法重建 FIFO 流的时间戳：

1. 基本法，只使用时间戳传感器信息；
2. 节省存储法，基于 FIFO\_DATA\_OUT\_TAG 中的 TAG\_CNT 字段
3. 混合法，基于 TAG\_CNT 字段和抽取时间戳传感器的组合使用

基本法保证时间戳重建的最高精度，但会浪费 FIFO 中的大量存储空间。在每个时隙将时间戳传感器写入 FIFO。如果发生溢出条件，从 FIFO 检索数据的正确程序是丢弃新的时间戳传感器之前的每次数据读取。

节省存储法只使用 TAG\_CNT 信息，并且当 TAG\_CNT 值增加时，应按如下方式更新在软件层保存的时间戳：

$$timestamp = timestamp(i - 1) + \frac{1}{BDR_{XL}}$$

节省存储法允许用户在 FIFO 中保存尽可能多的数据。通过这种方式，将所有时间戳相关性转发到应用处理器。

当可能发生溢出条件时，不建议使用此方法。

混合法是一种折衷方法，是前面两种解决方案的组合。时间戳被配置为在抽取后写入 FIFO。当 TAG\_CNT 值增加时，应以存储器保存法更新在软件层保存的时间戳，而在读取时间戳传感器时，应使用来自传感器的正确值重新校准在软件层保存的时间戳。

## 7 温度传感器

器件具有内部温度传感器，适用于环境温度测量。

如果加速度计传感器处于掉电模式，则温度传感器关闭。

启用加速度计后，温度传感器的输出数据速率为 104 Hz。

对于温度传感器，数据准备就绪信号由 STATUS\_REG 寄存器的 TDA 位表示。通过将 INT2\_CTRL 寄存器的 INT2\_DRDY\_TEMP 位置为 1，可将该信号驱动至 INT2 引脚。

温度数据通过 OUT\_TEMP\_H 和 OUT\_TEMP\_L 寄存器级联得出，并以 16 位数的二进制补码形式表示，灵敏度为 256 LSB / °C。输出零值对应 25°C。通过在室温（25°C）下执行 OPC（单点校准），可提高绝对温度精度，减少温度偏移的影响。

温度传感器数据也可存储在 FIFO 中（详细信息见第 6 节 先进先出（FIFO）缓冲区）。

### 7.1 温度数据计算示例

下表提供了在不同环境温度值下从温度数据寄存器中读取数据的几个基本示例。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 26. 输出数据寄存器内容 vs. 温度**

温度值	寄存器地址	
	OUT_TEMP_H (21h)	OUT_TEMP_L (20h)
0 °C	E7h	00h
25 °C	00h	00h
50 °C	19h	00h

## 8 自检功能

嵌入式自检功能可支持无需移动器件而对其功能进行检查。

当加速度计自检使能时，传感器上会施加一个驱动力，模拟一定的加速度。这种情况下，传感器输出会在其 DC 电平上表现出变化，该电平通过灵敏度值关联到所选量程。

当 CTRL5\_C 寄存器的 ST[1:0]\_XL 位被设定为 00b 时，加速度计自检功能关闭；当 ST[1:0]\_XL 位被置为 01b（正符号自检）或 10b（负符号自检）时，该功能使能。

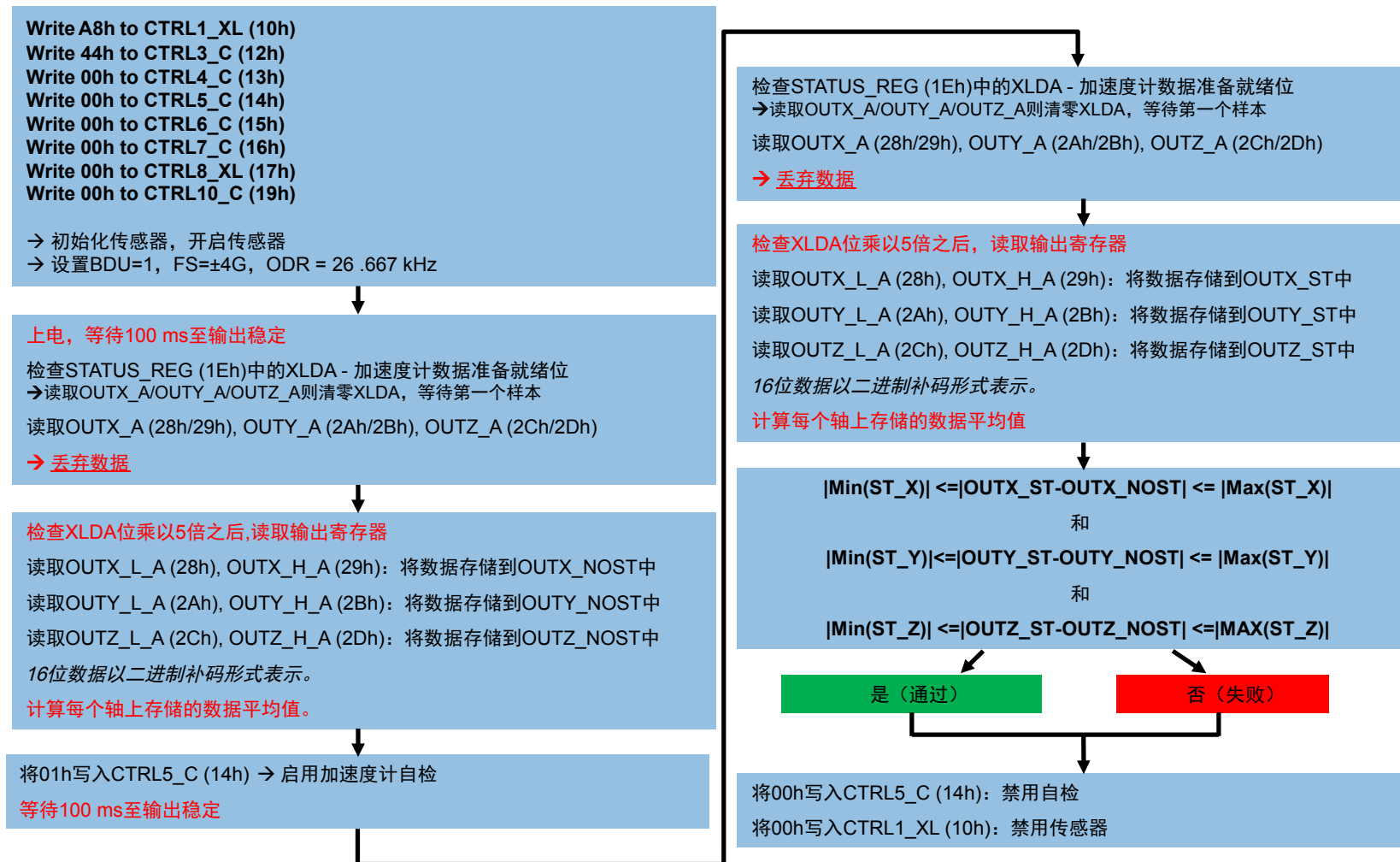
当加速度计自检功能激活时，传感器的输出由作用在传感器上的加速度和静电测试力的代数和给出。

完整的加速度计自检过程如图 15. 加速度计自检步骤中所示。



图 15. 加速度计自检步骤

注意：自检过程中保持器件静止。



## 版本历史

表 27. 文档版本历史

日期	版本	变更
2020 年 3 月 3 日	1	初始版本
2020 年 8 月 4 日	2	更新了 第 3 节 工作模式 更新了 第 5.5 节 时间戳

## 目录

<b>1</b>	引脚说明.....	<b>2</b>
<b>2</b>	寄存器.....	<b>4</b>
<b>3</b>	工作模式.....	<b>6</b>
<b>3.1</b>	加速度计滤波链 .....	<b>7</b>
<b>3.1.1</b>	加速度计斜率滤波器.....	<b>9</b>
<b>3.2</b>	加速度计开启/关断时间 .....	<b>9</b>
<b>4</b>	读取输出数据 .....	<b>10</b>
<b>4.1</b>	启动序列 .....	<b>10</b>
<b>4.2</b>	使用状态寄存器 .....	<b>10</b>
<b>4.3</b>	使用数据准备就绪信号.....	<b>11</b>
<b>4.3.1</b>	DRDY 屏蔽功能.....	<b>11</b>
<b>4.4</b>	使用块数据更新（block data update, BDU）功能 .....	<b>11</b>
<b>4.5</b>	理解输出数据 .....	<b>12</b>
<b>4.5.1</b>	输出数据示例 .....	<b>12</b>
<b>4.6</b>	加速度计偏移寄存器.....	<b>12</b>
<b>4.7</b>	环行功能 .....	<b>12</b>
<b>5</b>	中断生成.....	<b>13</b>
<b>5.1</b>	中断引脚配置 .....	<b>13</b>
<b>5.2</b>	唤醒中断 .....	<b>14</b>
<b>5.3</b>	活动/不活动识别 .....	<b>17</b>
<b>5.4</b>	启动状态 .....	<b>19</b>
<b>5.5</b>	时间戳 .....	<b>20</b>
<b>6</b>	先进先出（FIFO）缓冲区 .....	<b>21</b>
<b>6.1</b>	FIFO 寄存器.....	<b>21</b>
<b>6.1.1</b>	FIFO_CTRL1 .....	<b>22</b>
<b>6.1.2</b>	FIFO_CTRL2 .....	<b>22</b>
<b>6.1.3</b>	FIFO_CTRL3 .....	<b>22</b>
<b>6.1.4</b>	FIFO_CTRL4 .....	<b>23</b>
<b>6.1.5</b>	COUNTER_BDR_REG1.....	<b>24</b>
<b>6.1.6</b>	COUNTER_BDR_REG2.....	<b>24</b>

6.1.7	FIFO_STATUS1 .....	24
6.1.8	FIFO_STATUS2 .....	25
6.1.9	FIFO_DATA_OUT_TAG .....	26
6.1.10	FIFO_DATA_OUT .....	26
<b>6.2</b>	<b>FIFO 批处理传感器 .....</b>	<b>27</b>
6.2.1	FIFO 中的加速度计数据 .....	27
6.2.2	FIFO 中时间戳和温度数据 .....	27
<b>6.3</b>	<b>FIFO 模式 .....</b>	<b>28</b>
6.3.1	Bypass 模式 .....	28
6.3.2	FIFO 模式 .....	29
6.3.3	Continuous 模式 .....	30
6.3.4	Continue-FIFO 模式 .....	31
6.3.5	Bypass-Continue 模式 .....	32
6.3.6	Bypass-FIFO 模式 .....	33
<b>6.4</b>	<b>从 FIFO 检索数据 .....</b>	<b>34</b>
<b>6.5</b>	<b>FIFO 水印阈值 .....</b>	<b>35</b>
<b>6.6</b>	<b>时间戳相关性 .....</b>	<b>38</b>
<b>7</b>	<b>温度传感器 .....</b>	<b>39</b>
7.1	温度数据计算示例 .....	39
<b>8</b>	<b>自检功能 .....</b>	<b>40</b>
	版本历史 .....	42
	目录 .....	43
	表一览 .....	45
	图一览 .....	46

## 表一览

表 1.	默认引脚状态 .....	3
表 2.	寄存器 .....	4
表 3.	加速度计使能轴 .....	6
表 4.	加速度计带宽选择 .....	8
表 5.	输出数据寄存器内容 vs. 加速度 (FS_XL = ±2 g) .....	12
表 6.	INT1_CTRL 寄存器 .....	13
表 7.	MD1_CFG 寄存器 .....	13
表 8.	INT2_CTRL 寄存器 .....	13
表 9.	MD2_CFG 寄存器 .....	14
表 10.	FIFO_CTRL1 寄存器 .....	22
表 11.	FIFO_CTRL2 寄存器 .....	22
表 12.	FIFO_CTRL3 寄存器 .....	22
表 13.	加速度计批处理数据率 .....	22
表 14.	时间戳批处理数据率 .....	23
表 15.	温度传感器批处理数据率 .....	23
表 16.	FIFO_CTRL4 寄存器 .....	23
表 17.	COUNTER_BDR_REG1 寄存器 .....	24
表 18.	COUNTER_BDR_REG2 寄存器 .....	24
表 19.	FIFO_STATUS1 寄存器 .....	24
表 20.	FIFO_STATUS2 寄存器 .....	25
表 21.	FIFO_DATA_OUT_TAG 寄存器 .....	26
表 22.	TAG_SENSOR 字段和相关传感器 .....	26
表 23.	FIFO 中的主要传感器输出数据格式 .....	27
表 24.	FIFO 中的温度输出数据格式 .....	27
表 25.	FIFO 中的时间戳输出数据格式 .....	28
表 26.	输出数据寄存器内容 vs. 温度 .....	39
表 27.	文档版本历史 .....	42

## 图一览

图 1.	引脚连接 .....	2
图 2.	加速度计滤波链 .....	7
图 3.	加速度计斜率滤波器 .....	9
图 4.	数据准备就绪信号 .....	11
图 5.	唤醒中断（利用斜率滤波器） .....	15
图 6.	活动/不活动识别（利用斜率滤波器） .....	17
图 7.	FIFO 模式（STOP_ON_WTM = 0） .....	29
图 8.	Continuous 模式 .....	30
图 9.	Continue-FIFO 模式 .....	31
图 10.	Bypass-Continue 模式 .....	32
图 11.	Bypass-FIFO 模式 .....	33
图 12.	FIFO 阈值（STOP_ON_WTM = 0） .....	35
图 13.	FIFO 模式下的 FIFO 阈值（STOP_ON_WTM = 1） .....	36
图 14.	Continue 模式下的 FIFO 阈值（STOP_ON_WTM = 1） .....	36
图 15.	加速度计自检步骤 .....	41

重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对 ST 产品和/或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。

© 2018 STMicroelectronics - 保留所有权利