

LSM6DSL：始终开启的 3D 加速度计和 3D 陀螺仪

引言

本文档旨在提供 ST **LSM6DSL** iNEMO 六轴惯性传感器模块相关的使用信息和应用提示。

LSM6DSL 是系统级封装的 3D 数字加速度计和 3D 数字陀螺仪，具有数字 I²C/SPI 串口标准输出，在组合高性能工作模式下 6 轴功耗 0.65 mA。由于陀螺仪和加速度计均具有超低噪声性能，始终具有低功耗特性，并结合了高传感精度，因此能够为客户提供最佳运动体验。此外，加速度计具有智能的休眠到唤醒（活动）和返回休眠（不活动）功能，具备先进的节电能力。

该器件具有动态的用户可选择的满量程加速度范围： $\pm 2/\pm 4/\pm 8/\pm 16\text{ g}$ ，且角速率范围为 $\pm 125/\pm 250/\pm 500/\pm 1000/\pm 2000\text{ dps}$ 。

经过配置，**LSM6DSL** 可利用硬件识别出的自由落体事件、6D 方向、单击和双击感应、活动或不活动、唤醒事件，来生成中断信号。

可使用专用连接模式与外部传感器相连，因此能够实现传感器集合（**sensor hub**）功能。

LSM6DSL 可兼容主要操作系统的要求，提供真实、虚拟以及批量模式传感器。它经过专门设计，可在硬件上实现大幅运动检测、相对倾斜度、绝对倾斜度、硬件计步功能、时间戳，并支持对外部磁力计的数据采集，且支持铁磁校准（硬磁校准，软磁校准）。

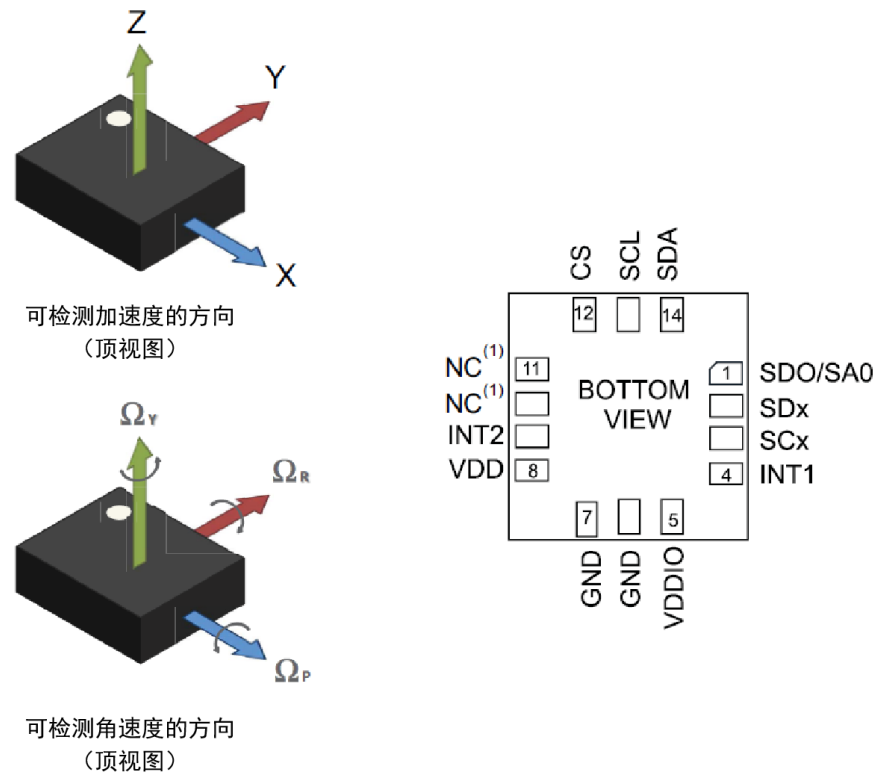
LSM6DSL 集成有一个 4 Kbyte 的智能先进先出（FIFO）缓冲器，支持对有效数据（包括外部传感器、计步器、时间戳和温度数据）进行动态批处理。

LSM6DSL 采用小型塑料焊盘网格阵列封装（LGA-14L），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手持便携式应用的理想选择，如智能手机、物联网（IoT）连接设备，穿戴，以及需要减小封装尺寸和重量的其他应用。

1 引脚说明

图 1. 引脚连接



1. Leave pin electrically unconnected and soldered to PCB.

表 1. 引脚状态

引脚 #	名称	模式 1 功能	模式 2 功能	引脚状态模式 1	引脚状态模式 2
1	SDO	SPI 4 线接口串行数据输出 (serial data output, SDO)	SPI 4 线接口串行数据输出 (serial data output, SDO)	默认值: 无上拉的输入。 上拉使能	默认值: 无上拉的输入。 上拉使能
	SA0	器件地址的 I ² C 最低有效位 (SA0)	器件地址的 I ² C 最低有效位 (SA0)	如果寄存器 12h 中的位 SIM = 1 (SPI 3 线)	如果寄存器 12h 中的位 SIM = 1 (SPI 3 线)
2	SDx	连接到 VDDIO 或 GND	I ² C 串行数据主机 (MSDA)	默认值: 无上拉的输入。 上拉使能 如果寄存器 1Ah 中的位 PULL_UP_EN = 1。	默认值: 无上拉的输入。 上拉使能 如果寄存器 1Ah 中的位 PULL_UP_EN = 1。
3	SCx	连接到 VDDIO 或 GND	I ² C 串行时钟主机 (MSCL)	默认值: 无上拉的输入。 上拉使能 如果寄存器 1Ah 中的位 PULL_UP_EN = 1。	默认值: 无上拉的输入。 上拉使能 如果寄存器 1Ah 中的位 PULL_UP_EN = 1。
4	INT1	可编程中断 1	可编程中断 1	默认值: 输出强制接地	默认值: 输出强制接地
5	Vdd_IO	I/O 引脚的供电	I/O 引脚的供电		
6	GND	0 V 电源	0 V 电源		
7	GND	0 V 电源	0 V 电源		
8	Vdd	电源	电源		
9	INT2	可编程中断 2 (INT2) / 数据使能 (data enable, DEN)	可编程中断 2 (INT2) / 数据使能 (data enable, DEN) / I ² C 主线的同步信号 (MDRDY)	默认值: 输出强制接地	默认值: 输出强制接地
10	NC	(1)	保持断开 ⁽¹⁾	默认值: 带上拉的输入。 (请参见下方的注释来禁用上拉)	默认值: 带上拉的输入。 (请参见下方的注释来禁用上拉)
11	NC	保持断开 ⁽¹⁾	保持断开 ⁽¹⁾	默认值: 带上拉的输入。 (请参见下方的注释来禁用上拉)	默认值: 带上拉的输入。 (请参见下方的注释来禁用上拉)
12	CS	I ² C/SPI 模式选择 (1: SPI 空闲模式/ I ² C 通信使能; 0: SPI 通信模式/ I ² C 禁用)	I ² C/SPI 模式选择 (1: SPI 空闲模式/ I ² C 通信使能; 0: SPI 通信模式/ I ² C 禁用)	默认值: 带上拉的输入。 上拉禁用 如果寄存器 13h 中的位 I2C_disable = 1。	默认值: 带上拉的输入。 上拉禁用 如果寄存器 13h 中的位 I2C_disable = 1。
13	SCL	I ² C 串行时钟 (SCL) / SPI 串行时钟 (serial port clock, SPC)	I ² C 串行时钟 (SCL) / SPI 串行时钟 (serial port clock, SPC)	输入无上拉	输入无上拉
14	SDA	I ² C 串行数据 (SDA) / SPI 串行数据输入 (SDI) / 3 线 SPI 接口串行数据输出 (serial data output, SDO)	I ² C 串行数据 (SDA) / SPI 串行数据输入 (SDI) / 3 线 SPI 接口串行数据输出 (serial data output, SDO)	输入无上拉	输入无上拉

1. 悬空的引脚也保证焊接到 PCB 上。

内部上拉值范围从 30 kΩ 至 50 kΩ, 取决于 VDDIO。

注: 禁用引脚 10-11 上上拉的步骤

- 通过主 I²C/SPI 接口: 在地址 00h 的寄存器中写入 80h
- 通过主 I²C/SPI 接口: 在地址 05h 的寄存器中写入 01h (禁用引脚 10 & 11 上的上拉)
- 通过主 I²C/SPI 接口: 在地址 00h 的寄存器中写入 00h

表 2. 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
FUNC_CFG_ACCESS	01h	FUNC_CFG_EN	0	FUNC_CFG_EN_B	0	0	0	0	0
SENSOR_SYNC_TIME_FRAME	04h	0	0	0	0	TPH_3	TPH_2	TPH_1	TPH_0
SENSOR_SYNC_RES_RATIO	05h	0	0	0	0	0	0	RR_1	RR_0
FIFO_CTRL1	06h	FTH_7	FTH_6	FTH_5	FTH_4	FTH_3	FTH_2	FTH_1	FTH_0
FIFO_CTRL2	07h	TIMER_PEDO_FIFO_EN	TIMER_PEDO_FIFO_DRDY	0	0	FIFO_TEMP_EN	FTH_10	FTH_9	FTH_8
FIFO_CTRL3	08h	0	0	DEC_FIFO_GYRO2	DEC_FIFO_GYRO1	DEC_FIFO_GYRO0	DEC_FIFO_XL2	DEC_FIFO_XL1	DEC_FIFO_XL0
FIFO_CTRL4	09h	STOP_ON_FTH	ONLY_HIGH_DATA	DEC_DS4_FIFO2	DEC_DS4_FIFO1	DEC_DS4_FIFO0	DEC_DS3_FIFO2	DEC_DS3_FIFO1	DEC_DS3_FIFO0
FIFO_CTRL5	0Ah	0	ODR_FIFO_3	ODR_FIFO_2	ODR_FIFO_1	ODR_FIFO_0	FIFO_MODE_2	FIFO_MODE_1	FIFO_MODE_0
DRDY_PULSE_CFG	0Bh	DRDY_PULSED	0	0	0	0	0	0	INT2_WRIST_TILT
INT1_CTRL	0Dh	INT1_STEP_DETECTOR	INT1_SIGN_MOT	INT1_FULL_FLAG	INT1_FIFO_OVR	INT1_FTH	INT1_BOOT	INT1_DRDY_G	INT1_DRDY_XL
INT2_CTRL	0Eh	INT2_STEP_DELTA	INT2_STEP_COUNT_OV	INT2_FULL_FLAG	INT2_FIFO_OVR	INT2_FTH	INT2_DRDY_TEMP	INT2_DRDY_G	INT2_DRDY_XL
WHO_AM_I	0Fh	0	1	1	0	1	0	1	0
CTRL1_XL	10h	ODR_XL3	ODR_XL2	ODR_XL1	ODR_XL0	FS_XL1	FS_XL0	LPF1_BW_SEL	BW0_XL
CTRL2_G	11h	ODR_G3	ODR_G2	ODR_G1	ODR_G0	FS_G1	FS_G0	FS_125	0
CTRL3_C	12h	BOOT	BDU	H_LACTIVE	PP_OD	SIM	IF_INC	BLE	SW_RESET
CTRL4_C	13h	DEN_XL_EN	SLEEP	INT2_on_INT1	DEN_DRDY_INT1	DRDY_MASK	I2C_disable	LPF1_SEL_G	0
CTRL5_C	14h	ROUNDING2	ROUNDING1	ROUNDING0	DEN_LH	ST1_G	ST0_G	ST1_XL	ST0_XL
CTRL6_C	15h	TRIG_EN	LVL1_EN	LVL2_EN	XL_HM_MODE	USR_OFF_W	0	FTYPE_1	FTYPE_0
CTRL7_G	16h	G_HM_MODE	HP_G_EN	HPM1_G	HPM0_G	0	ROUNDING_STATUS	0	0



寄存器名	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
CTRL8_XL	17h	LPF2_XL_EN	HPCF_XL1	HPCF_XL0	HP_REF_MODE	INPUT_COMPOSITE	HP_SLOPE_XL_EN	0	LOW_PASS_ON_6D
CTRL9_XL	18h	DEN_X	DEN_Y	DEN_Z	DEN_XL_G	0	SOFT_EN	0	0
CTRL10_C	19h	WRIST_TILT_EN	0	TIMER_EN	PEDO_EN	TILT_EN	FUNC_EN	PEDO_RST_STEP	SIGN_MOTION_EN
MASTER_CONFIG	1Ah	DRDY_ON_INT1	DATA_VALID_SEL_FIFO	0	START_CONFIG	PULL_UP_EN	PASS_THROUGH_MODE	IRON_EN	MASTER_ON
WAKE_UP_SRC	1Bh	0	0	FF_IA	SLEEP_STATE_IA	WU_IA	X_WU	Y_WU	Z_WU
TAP_SRC	1Ch	0	TAP_IA	SINGLE_TAP	DOUBLE_TAP	TAP_SIGN	X_TAP	Y_TAP	Z_TAP
D6D_SRC	1Dh	DEN_DRDY	D6D_IA	ZH	ZL	YH	YL	XH	XL
STATUS_REG	1Eh	0	0	0	0	0	TDA	GDA	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_G	22h	D7	D6	D5	D4	D3	D2	D1	D0
OUTX_H_G	23h	D15	D14	D13	D12	D11	D10	D9	D8
OUTY_L_G	24h	D7	D6	D5	D4	D3	D2	D1	D0
OUTY_H_G	25h	D15	D14	D13	D12	D11	D10	D9	D8
OUTZ_L_G	26h	D7	D6	D5	D4	D3	D2	D1	D0
OUTZ_H_G	27h	D15	D14	D13	D12	D11	D10	D9	D8
OUTX_L_XL	28h	D7	D6	D5	D4	D3	D2	D1	D0
OUTX_H_XL	29h	D15	D14	D13	D12	D11	D10	D9	D8
OUTY_L_XL	2Ah	D7	D6	D5	D4	D3	D2	D1	D0
OUTY_H_XL	2Bh	D15	D14	D13	D12	D11	D10	D9	D8
OUTZ_L_XL	2Ch	D7	D6	D5	D4	D3	D2	D1	D0
OUTZ_H_XL	2Dh	D15	D14	D13	D12	D11	D10	D9	D8
SENSORHUB1_REG	2Eh	SHub1_7	SHub1_6	SHub1_5	SHub1_4	SHub1_3	SHub1_2	SHub1_1	SHub1_0
SENSORHUB2_REG	2Fh	SHub2_7	SHub2_6	SHub2_5	SHub2_4	SHub2_3	SHub2_2	SHub2_1	SHub2_0
SENSORHUB3_REG	30h	SHub3_7	SHub3_6	SHub3_5	SHub3_4	SHub3_3	SHub3_2	SHub3_1	SHub3_0
SENSORHUB4_REG	31h	SHub4_7	SHub4_6	SHub4_5	SHub4_4	SHub4_3	SHub4_2	SHub4_1	SHub4_0
SENSORHUB5_REG	32h	SHub5_7	SHub5_6	SHub5_5	SHub5_4	SHub5_3	SHub5_2	SHub5_1	SHub5_0
SENSORHUB6_REG	33h	SHub6_7	SHub6_6	SHub6_5	SHub6_4	SHub6_3	SHub6_2	SHub6_1	SHub6_0
SENSORHUB7_REG	34h	SHub7_7	SHub7_6	SHub7_5	SHub7_4	SHub7_3	SHub7_2	SHub7_1	SHub7_0
SENSORHUB8_REG	35h	SHub8_7	SHub8_6	SHub8_5	SHub8_4	SHub8_3	SHub8_2	SHub8_1	SHub8_0
SENSORHUB9_REG	36h	SHub9_7	SHub9_6	SHub9_5	SHub9_4	SHub9_3	SHub9_2	SHub9_1	SHub9_0
SENSORHUB10_REG	37h	SHub10_7	SHub10_6	SHub10_5	SHub10_4	SHub10_3	SHub10_2	SHub10_1	SHub10_0
SENSORHUB11_REG	38h	SHub11_7	SHub11_6	SHub11_5	SHub11_4	SHub11_3	SHub11_2	SHub11_1	SHub11_0



根据保密协议，不可复制

AN5040
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
SENSORHUB12_REG	39h	SHub12_7	SHub12_6	SHub12_5	SHub12_4	SHub12_3	SHub12_2	SHub12_1	SHub12_0
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	WaterM	OVER_RUN	FIFO_FULL_SMART	FIFO_EMPTY	0	DIFF_FIFO_10	DIFF_FIFO_9	DIFF_FIFO_8
FIFO_STATUS3	3Ch	FIFO_PATTERN_7	FIFO_PATTERN_6	FIFO_PATTERN_5	FIFO_PATTERN_4	FIFO_PATTERN_3	FIFO_PATTERN_2	FIFO_PATTERN_1	FIFO_PATTERN_0
FIFO_STATUS4	3Dh	0	0	0	0	0	0	FIFO_PATTERN_9	FIFO_PATTERN_8
FIFO_DATA_OUT_L	3Eh	DATA_OUT_FIFO_L_7	DATA_OUT_FIFO_L_6	DATA_OUT_FIFO_L_5	DATA_OUT_FIFO_L_4	DATA_OUT_FIFO_L_3	DATA_OUT_FIFO_L_2	DATA_OUT_FIFO_L_1	DATA_OUT_FIFO_L_0
FIFO_DATA_OUT_H	3Fh	DATA_OUT_FIFO_H_7	DATA_OUT_FIFO_H_6	DATA_OUT_FIFO_H_5	DATA_OUT_FIFO_H_4	DATA_OUT_FIFO_H_3	DATA_OUT_FIFO_H_2	DATA_OUT_FIFO_H_1	DATA_OUT_FIFO_H_0
TIMESTAMP0_REG	40h	TIMESTAMP_0_7	TIMESTAMP_0_6	TIMESTAMP_0_5	TIMESTAMP_0_4	TIMESTAMP_0_3	TIMESTAMP_0_2	TIMESTAMP_0_1	TIMESTAMP_0_0
TIMESTAMP1_REG	41h	TIMESTAMP_1_7	TIMESTAMP_1_6	TIMESTAMP_1_5	TIMESTAMP_1_4	TIMESTAMP_1_3	TIMESTAMP_1_2	TIMESTAMP_1_1	TIMESTAMP_1_0
TIMESTAMP2_REG	42h	TIMESTAMP_2_7	TIMESTAMP_2_6	TIMESTAMP_2_5	TIMESTAMP_2_4	TIMESTAMP_2_3	TIMESTAMP_2_2	TIMESTAMP_2_1	TIMESTAMP_2_0
STEP_TIMESTAMP_L	49h	STEP_TIME_STAMP_L_7	STEP_TIME_STAMP_L_6	STEP_TIME_STAMP_L_5	STEP_TIME_STAMP_L_4	STEP_TIME_STAMP_L_3	STEP_TIME_STAMP_L_2	STEP_TIME_STAMP_L_1	STEP_TIME_STAMP_L_0
STEP_TIMESTAMP_H	4Ah	STEP_TIME_STAMP_H_7	STEP_TIME_STAMP_H_6	STEP_TIME_STAMP_H_5	STEP_TIME_STAMP_H_4	STEP_TIME_STAMP_H_3	STEP_TIME_STAMP_H_2	STEP_TIME_STAMP_H_1	STEP_TIME_STAMP_H_0
STEP_COUNTER_L	4Bh	STEP_COUNTER_L_7	STEP_COUNTER_L_6	STEP_COUNTER_L_5	STEP_COUNTER_L_4	STEP_COUNTER_L_3	STEP_COUNTER_L_2	STEP_COUNTER_L_1	STEP_COUNTER_L_0
STEP_COUNTER_H	4Ch	STEP_COUNTER_H_7	STEP_COUNTER_H_6	STEP_COUNTER_H_5	STEP_COUNTER_H_4	STEP_COUNTER_H_3	STEP_COUNTER_H_2	STEP_COUNTER_H_1	STEP_COUNTER_H_0
SENSORHUB13_REG	4Dh	SHub13_7	SHub13_6	SHub13_5	SHub13_4	SHub13_3	SHub13_2	SHub13_1	SHub13_0
SENSORHUB14_REG	4Eh	SHub14_7	SHub14_6	SHub14_5	SHub14_4	SHub14_3	SHub14_2	SHub14_1	SHub14_0
SENSORHUB15_REG	4Fh	SHub15_7	SHub15_6	SHub15_5	SHub15_4	SHub15_3	SHub15_2	SHub15_1	SHub15_0
SENSORHUB16_REG	50h	SHub16_7	SHub16_6	SHub16_5	SHub16_4	SHub16_3	SHub16_2	SHub16_1	SHub16_0
SENSORHUB17_REG	51h	SHub17_7	SHub17_6	SHub17_5	SHub17_4	SHub17_3	SHub17_2	SHub17_1	SHub17_0
SENSORHUB18_REG	52h	SHub18_7	SHub18_6	SHub18_5	SHub18_4	SHub18_3	SHub18_2	SHub18_1	SHub18_0
FUNC_SRC1	53h	STEP_COUNT_DELTA_IA	SIGN_MOTION_IA	TILT_IA	STEP_DETECTED	STEP_OVERFLOW	HI_FAIL	SI_END_OP	SENSORHUB_END_OP
FUNC_SRC2	54h	0	SLAVE3_NACK	SLAVE2_NACK	SLAVE1_NACK	SLAVE0_NACK	0	0	WRIST_TILT_IA



寄存器名	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
WRIST_TILT_IA	55h	WRIST_TILT_IA_Xpos	WRIST_TILT_IA_Xneg	WRIST_TILT_IA_Ypos	WRIST_TILT_IA_Yneg	WRIST_TILT_IA_Zpos	WRIST_TILT_IA_Zneg	0	0
TAP_CFG	58h	INTERRUPTS_ENABLE	INACT_EN1	INACT_EN0	SLOPE_FDS	TAP_X_EN	TAP_Y_EN	TAP_Z_EN	LIR
TAP_THS_6D	59h	D4D_EN	SIXD_THS1	SIXD_THS0	TAP_THS4	TAP_THS3	TAP_THS2	TAP_THS1	TAP_THS0
INT_DUR2	5Ah	DUR3	DUR2	DUR1	DUR0	QUIET1	QUIET0	SHOCK1	SHOCK0
WAKE_UP_THS	5Bh	SINGLE_DOUBLE_TAP	0	WK_THS5	WK_THS4	WK_THS3	WK_THS2	WK_THS1	WK_THS0
WAKE_UP_DUR	5Ch	FF_DUR5	WAKE_DUR1	WAKE_DUR0	TIMER_HR	SLEEP_DUR3	SLEEP_DUR2	SLEEP_DUR1	SLEEP_DUR0
FREE_FALL	5Dh	FF_DUR4	FF_DUR3	FF_DUR2	FF_DUR1	FF_DUR0	FF_THS2	FF_THS1	FF_THS0
MD1_CFG	5Eh	INT1_INACT_STATE	INT1_SINGLE_TAP	INT1_WU	INT1_FF	INT1_DOUBLE_TAP	INT1_6D	INT1_TILT	INT1_TIMER
MD2_CFG	5Fh	INT2_INACT_STATE	INT2_SINGLE_TAP	INT2_WU	INT2_FF	INT2_DOUBLE_TAP	INT2_6D	INT2_TILT	INT2_IRON
MASTER_CMD_CODE	60h	MASTER_CMD_CODE7	MASTER_CMD_CODE6	MASTER_CMD_CODE5	MASTER_CMD_CODE4	MASTER_CMD_CODE3	MASTER_CMD_CODE2	MASTER_CMD_CODE1	MASTER_CMD_CODE0
SENS_SYNC_SPI_ERROR_CODE	61h	ERROR_CODE7	ERROR_CODE6	ERROR_CODE5	ERROR_CODE4	ERROR_CODE3	ERROR_CODE2	ERROR_CODE1	ERROR_CODE0
OUT_MAG_RAW_X_L	66h	D7	D6	D5	D4	D3	D2	D1	D0
OUT_MAG_RAW_X_H	67h	D15	D14	D13	D12	D11	D10	D9	D8
OUT_MAG_RAW_Y_L	68h	D7	D6	D5	D4	D3	D2	D1	D0
OUT_MAG_RAW_Y_H	69h	D15	D14	D13	D12	D11	D10	D9	D8
OUT_MAG_RAW_Z_L	6Ah	D7	D6	D5	D4	D3	D2	D1	D0
OUT_MAG_RAW_Z_H	6Bh	D15	D14	D13	D12	D11	D10	D9	D8
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

2.1 嵌入功能寄存器

该器件可用的嵌入功能寄存器列表在表 3. 嵌入功能寄存器（A 区）和表 4. 嵌入功能寄存器（B 区）中给出。

当 FUNC_CFG_EN 位被置为“1”并且 FUNC_CFG_ACCESS 寄存器中的 FUNC_CFG_EN_B 位被置为 0 时，第一（A）区的嵌入功能寄存器可以被访问。

当 FUNC_CFG_EN 和 FUNC_CFG_EN_B 位在 FUNC_CFG_ACCESS 寄存器中都被置为“1”时，第二（B）区的嵌入功能寄存器可以被访问。

注意：所有对嵌入功能寄存器内容的修改必须在加速度计和陀螺仪传感器处于下电模式时进行。

表 3. 嵌入功能寄存器（A 区）

寄存器名	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
SLV0_ADD	02h	Slave0 _add6	Slave0 _add5	Slave0 _add4	Slave0 _add3	Slave0 _add2	Slave0 _add1	Slave0 _add0	rw_0
SLV0_SUBADD	03h	Slave0 _reg7	Slave0 _reg6	Slave0 _reg5	Slave0 _reg4	Slave0 _reg3	Slave0 _reg2	Slave0 _reg1	Slave0 _reg0
SLAVE0_CONFIG	04h	Slave0 _rate1	Slave0 _rate0	Aux_sens _on1	Aux_sens _on0	Src_mode	Slave0_numop2	Slave0_numop1	Slave0_numop0
SLV1_ADD	05h	Slave1 _add6	Slave1 _add5	Slave1 _add4	Slave1 _add3	Slave1 _add2	Slave1 _add1	Slave1 _add0	r_1
SLV1_SUBADD	06h	Slave1 _reg7	Slave1 _reg6	Slave1 _reg5	Slave1 _reg4	Slave1 _reg3	Slave1 _reg2	Slave1 _reg1	Slave1 _reg0
SLAVE1_CONFIG	07h	Slave1 _rate1	Slave1 _rate0	write_once	0	0	Slave1_numop2	Slave1_numop1	Slave1_numop0
SLV2_ADD	08h	Slave2 _add6	Slave2 _add5	Slave2 _add4	Slave2 _add3	Slave2 _add2	Slave2 _add1	Slave2 _add0	r_2
SLV2_SUBADD	09h	Slave2 _reg7	Slave2 _reg6	Slave2 _reg5	Slave2 _reg4	Slave2 _reg3	Slave2 _reg2	Slave2 _reg1	Slave2 _reg0
SLAVE2_CONFIG	0Ah	Slave2 _rate1	Slave2 _rate0	0	0	0	Slave2_numop2	Slave2_numop1	Slave2_numop0
SLV3_ADD	0Bh	Slave3 _add6	Slave3 _add5	Slave3 _add4	Slave3 _add3	Slave3 _add2	Slave3 _add1	Slave3 _add0	r_3
SLV3_SUBADD	0Ch	Slave3 _reg7	Slave3 _reg6	Slave3 _reg5	Slave3 _reg4	Slave3 _reg3	Slave3 _reg2	Slave3 _reg1	Slave3 _reg0
SLAVE3_CONFIG	0Dh	Slave3 _rate1	Slave3 _rate0	0	0	0	Slave3_numop2	Slave3_numop1	Slave3_numop0
DATAWRITE_SRC _MODE_SUB_SLV0	0Eh	Slave_ dataw7	Slave_ dataw6	Slave_ dataw5	Slave_ dataw4	Slave_ dataw3	Slave_ dataw2	Slave_ dataw1	Slave_ dataw0
CONFIG_PEDO	0Fh	PEDO_FS	0	0	ths_min_4	ths_min_3	ths_min_2	ths_min_1	ths_min_0

寄存器名	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
_THS_MIN									
SM_THS	13h	SM_THS_7	SM_THS_6	SM_THS_5	SM_THS_4	SM_THS_3	SM_THS_2	SM_THS_1	SM_THS_0
PEDO_DEB_REG	14h	DEB _TIME_4	DEB _TIME_3	DEB _TIME_2	DEB _TIME_1	DEB _TIME_0	DEB _STEP_2	DEB _STEP_1	DEB _STEP_0
STEP_COUNT_ DELTA	15h	SC_DELTA _7	SC_DELTA _6	SC_DELTA _5	SC_DELTA _4	SC_DELTA _3	SC_DELTA _2	SC_DELTA _1	SC_DELTA _0
MAG_SI_XX	24h	MAG_SI_XX_7	MAG_SI_XX_6	MAG_SI_XX_5	MAG_SI_XX_4	MAG_SI_XX_3	MAG_SI_XX_2	MAG_SI_XX_1	MAG_SI_XX_0
MAG_SI_XY	25h	MAG_SI_XY_7	MAG_SI_XY_6	MAG_SI_XY_5	MAG_SI_XY_4	MAG_SI_XY_3	MAG_SI_XY_2	MAG_SI_XY_1	MAG_SI_XY_0
MAG_SI_XZ	26h	MAG_SI_XZ_7	MAG_SI_XZ_6	MAG_SI_XZ_5	MAG_SI_XZ_4	MAG_SI_XZ_3	MAG_SI_XZ_2	MAG_SI_XZ_1	MAG_SI_XZ_0
MAG_SI_YX	27h	MAG_SI_YX_7	MAG_SI_YX_6	MAG_SI_YX_5	MAG_SI_YX_4	MAG_SI_YX_3	MAG_SI_YX_2	MAG_SI_YX_1	MAG_SI_YX_0
MAG_SI_YY	28h	MAG_SI_YY_7	MAG_SI_YY_6	MAG_SI_YY_5	MAG_SI_YY_4	MAG_SI_YY_3	MAG_SI_YY_2	MAG_SI_YY_1	MAG_SI_YY_0
MAG_SI_YZ	29h	MAG_SI_YZ_7	MAG_SI_YZ_6	MAG_SI_YZ_5	MAG_SI_YZ_4	MAG_SI_YZ_3	MAG_SI_YZ_2	MAG_SI_YZ_1	MAG_SI_YZ_0
MAG_SI_ZX	2Ah	MAG_SI_ZX_7	MAG_SI_ZX_6	MAG_SI_ZX_5	MAG_SI_ZX_4	MAG_SI_ZX_3	MAG_SI_ZX_2	MAG_SI_ZX_1	MAG_SI_ZX_0
MAG_SI_ZY	2Bh	MAG_SI_ZY_7	MAG_SI_ZY_6	MAG_SI_ZY_5	MAG_SI_ZY_4	MAG_SI_ZY_3	MAG_SI_ZY_2	MAG_SI_ZY_1	MAG_SI_ZY_0
MAG_SI_ZZ	2Ch	MAG_SI_ZZ_7	MAG_SI_ZZ_6	MAG_SI_ZZ_5	MAG_SI_ZZ_4	MAG_SI_ZZ_3	MAG_SI_ZZ_2	MAG_SI_ZZ_1	MAG_SI_ZZ_0
MAG_OFFX_L	2Dh	MAG_OFFX_L_7	MAG_OFFX_L_6	MAG_OFFX_L_5	MAG_OFFX_L_4	MAG_OFFX_L_3	MAG_OFFX_L_2	MAG_OFFX_L_1	MAG_OFFX_L_0
MAG_OFFX_H	2Eh	MAG_OFFX_H_7	MAG_OFFX_H_6	MAG_OFFX_H_5	MAG_OFFX_H_4	MAG_OFFX_H_3	MAG_OFFX_H_2	MAG_OFFX_H_1	MAG_OFFX_H_0
MAG_OFFY_L	2Fh	MAG_OFFY_L_7	MAG_OFFY_L_6	MAG_OFFY_L_5	MAG_OFFY_L_4	MAG_OFFY_L_3	MAG_OFFY_L_2	MAG_OFFY_L_1	MAG_OFFY_L_0
MAG_OFFY_H	30h	MAG_OFFY_H_7	MAG_OFFY_H_6	MAG_OFFY_H_5	MAG_OFFY_H_4	MAG_OFFY_H_3	MAG_OFFY_H_2	MAG_OFFY_H_1	MAG_OFFY_H_0
MAG_OFFZ_L	31h	MAG_OFFZ_L_7	MAG_OFFZ_L_6	MAG_OFFZ_L_5	MAG_OFFZ_L_4	MAG_OFFZ_L_3	MAG_OFFZ_L_2	MAG_OFFZ_L_1	MAG_OFFZ_L_0
MAG_OFFZ_H	32h	MAG_OFFZ_H_7	MAG_OFFZ_H_6	MAG_OFFZ_H_5	MAG_OFFZ_H_4	MAG_OFFZ_H_3	MAG_OFFZ_H_2	MAG_OFFZ_H_1	MAG_OFFZ_H_0

表 4. 嵌入功能寄存器（B 区）

寄存器名	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
A_WRIST_TILT_LAT	50h	WRIST_TILT _TIMER7	WRIST_TILT _TIMER6	WRIST_TILT _TIMER5	WRIST_TILT _TIMER4	WRIST_TILT _TIMER3	WRIST_TILT _TIMER2	WRIST_TILT _TIMER1	WRIST_TILT _TIMER0
A_WRIST_TILT_THS	54h	WRIST_TILT _THS7	WRIST_TILT _THS6	WRIST_TILT _THS5	WRIST_TILT _THS4	WRIST_TILT _THS3	WRIST_TILT _THS2	WRIST_TILT _THS1	WRIST_TILT _THS0
A_WRIST_TILT_Mask	59h	WRIST_TILT _MASK_Xpos	WRIST_TILT _MASK_Xneg	WRIST_TILT _MASK_Ypos	WRIST_TILT _MASK_Yneg	WRIST_TILT _MASK_Zpos	WRIST_TILT _MASK_Zneg	0	0



根据保密协议，不可复制

3 工作模式

LSM6DSL 提供了 3 种可能的操作配置：

- 只有加速度计工作，陀螺仪下电；
- 只有陀螺仪工作，加速度计下电；
- 加速度计和陀螺仪均工作，且具有独立的 ODR。

该器件提供了较宽的 VDD 电压范围（从 1.71 V 至 3.6 V）和 1.62 V 至 3.6 V 的 VDDIO 范围。为避免潜在冲突，在上电时序期间，建议将连接至器件 IO 引脚的线路设置为主机侧的高阻抗状态。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 100µs。

施加电源后，LSM6DSL 执行一段 15 ms 的启动程序来加载启动参数。启动完成后，加速度计和陀螺仪均自动配置为下电模式。

加速度计和陀螺仪可分别配置为四种不同的功耗模式：下电，低功耗，正常和高性能模式。它们可以具有不同的数据率而不受任何限制。陀螺仪传感器还能够设置为睡眠模式，以降低其功耗。

当加速度计和陀螺仪均工作时，加速度计与陀螺仪同步，两个传感器的数据率是彼此的整数倍。

参考 LSM6DSL 数据手册，可以利用 CTRL1_XL 寄存器的输出数据率（ODR_XL）位和 CTRL6_C 寄存器的高性能禁止（XL_HM_MODE）位，来选择功耗模式和加速度计的输出数据率（表 5. 加速度计 ODR 和功耗模式选择）。

表 5. 加速度计 ODR 和功耗模式选择

ODR_XL [3:0]	ODR [Hz], 当 XL_HM_MODE = 1	ODR [Hz], 当 XL_HM_MODE = 0
0000	下电	下电
1011	1.6 Hz（仅低功耗）	12.5 Hz（高性能）
0001	12.5 Hz（低功耗）	12.5 Hz（高性能）
0010	26 Hz（低功耗）	26 Hz（高性能）
0011	52 Hz（低功耗）	52 Hz（高性能）
0100	104 Hz（正常模式）	104 Hz（高性能）
0101	208 Hz（正常模式）	208 Hz（高性能）
0110	416 Hz（高性能）	416 Hz（高性能）
0111	833 Hz（高性能）	833 Hz（高性能）
1000	1.66 kHz（高性能）	1.66 kHz（高性能）
1001	3.33 kHz（高性能）	3.33 kHz（高性能）
1010	6.66 kHz（高性能）	6.66 kHz（高性能）

可以利用 CTRL2_G 寄存器的输出数据率（ODR_G）位和 CTRL7_G 寄存器的高性能禁止（G_HM_MODE）位，来选择功耗模式和陀螺仪传感器的输出数据率（表 6. 陀螺仪 ODR 和功耗模式选择）。

表 6. 陀螺仪 ODR 和功耗模式选择

ODR_G [3:0]	ODR [Hz], 当 G_HM_MODE = 1	ODR [Hz], 当 G_HM_MODE = 0
0000	下电	下电
0001	12.5 Hz（低功耗）	12.5 Hz（高性能）
0010	26 Hz（低功耗）	26 Hz（高性能）

ODR_G [3:0]	ODR [Hz], 当 G_HM_MODE = 1	ODR [Hz], 当 G_HM_MODE = 0
0011	52 Hz (低功耗)	52 Hz (高性能)
0100	104 Hz (正常模式)	104 Hz (高性能)
0101	208 Hz (正常模式)	208 Hz (高性能)
0110	416 Hz (高性能)	416 Hz (高性能)
0111	833 Hz (高性能)	833 Hz (高性能)
1000	1.66 kHz (高性能)	1.66 kHz (高性能)
1001	3.33 kHz (高性能)	3.33 kHz (高性能)
1010	6.66 kHz (高性能)	6.66 kHz (高性能)

表 7. 功耗 显示了不同工作模式下功耗典型值。

表 7. 功耗

ODR [Hz]	只有加速度计 (在 Vdd = 1.8 V 时)	只有陀螺仪 (在 Vdd = 1.8 V 时)	组合 [Acc + Gyro] (在 Vdd = 1.8 V 时)
下电	-	-	3 µA
1.6 Hz (低功耗)	4.5 µA	-	-
12.5 Hz (低功耗)	9 µA	232 µA	240 µA
26 Hz (低功耗)	14 µA	245 µA	260 µA
52 Hz (低功耗)	25 µA	270 µA	290 µA
104 Hz (正常模式)	44 µA	325 µA	360 µA
208 Hz (正常模式)	85 µA	430 µA	450 µA
12.5 Hz (高性能)	150 µA	555 µA	650 µA
26 Hz (高性能)	150 µA	555 µA	650 µA
52 Hz (高性能)	150 µA	555 µA	650 µA
104 Hz (高性能)	150 µA	555 µA	650 µA
208 Hz (高性能)	150 µA	555 µA	650 µA
416 Hz (高性能)	150 µA	555 µA	650 µA
833 Hz (高性能)	150 µA	555 µA	650 µA
1.66 kHz (高性能)	160 µA	555 µA	650 µA
3.33 kHz (高性能)	160 µA	555 µA	650 µA
6.66 kHz (高性能)	160 µA	555 µA	650 µA

3.1 下电模式

当加速度计/陀螺仪处于下电模式时，该器件几乎所有的内部模块都会断开，以减小功耗。数字接口（I²C 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，可保存进入下电模式前存储器中采样的最后数据。

3.2 高性能模式

高性能模式下，所有的加速器/陀螺仪电路始终接通，并通过 ODR_XL/ODR_G 位来选择生成数据的数据率。数据中断产生是激活的。

3.3 正常模式

高性能模式能够确保其在噪声方面具有最佳性能，而正常模式可进一步降低电流消耗。加速度计/陀螺仪数据读取链自动打开和关闭，以便节能。在陀螺仪模块中，只有驱动电路是始终工作的。数据中断产生是激活的。

3.4 低功耗模式

低功耗模式与正常模式可使用的输出数据率不同。低功耗模式下，低速 ODR 使能。可以通过 ODR_XL 位为加速度计选择四个低速 ODR：1.6 Hz、12.5 Hz、26 Hz 和 52 Hz。可以通过 ODR_G 位为陀螺仪选择三个低速 ODR：12.5 Hz、26 Hz 和 52 Hz。

数据中断产生是激活的。

3.5 陀螺仪睡眠模式

当陀螺仪处于睡眠模式时，陀螺仪模块的振荡驱动电路保持激活。相比于陀螺仪下电模式，从睡眠模式到低功耗/正常/高性能模式的打开时间大幅度减少。

如果陀螺仪未被配置为下电模式，则当 CTRL4_C 寄存器的睡眠模式使能（SLEEP）位被置为 1 时，不论所选陀螺仪 ODR 为何，它都将进入睡眠模式。

3.6 连接模式

LSM6DSL 提供了两种不同的连接模式，本文档中作了详细描述：

- 模式 1：这是默认启用的连接模式；I²C 从接口或 SPI（3-/4-线）串口可用。当器件配置为连接模式 1 时，SCx/SDx 引脚不能浮空。建议将二者连接到 VDDIO，以便优化器件启动过程中的功耗。
- 模式 2：它是传感器集合模式；I²C 从接口或 SPI（3-/4-线）串口和用于外部传感器连接的 I²C 接口主机可用。第 7 节 模式 2 - 传感器集合（sensor hub）模式中描述了此连接模式。

3.7 加速度计带宽

加速度计采样链路由 4 个级联主模块表示：一个模拟抗混叠低通滤波器，一个 ADC 转换器，一个数字低通滤波器和复合数字滤波器组。

如图 2. 加速度计滤波链中所示，来自机械部件的模拟信号在被 ADC 转换之前，会由模拟低通抗混叠滤波器进行滤波。抗混叠滤波器仅在高性能模式下启用，其带宽取决于所选加速度计 ODR，如下表所示。

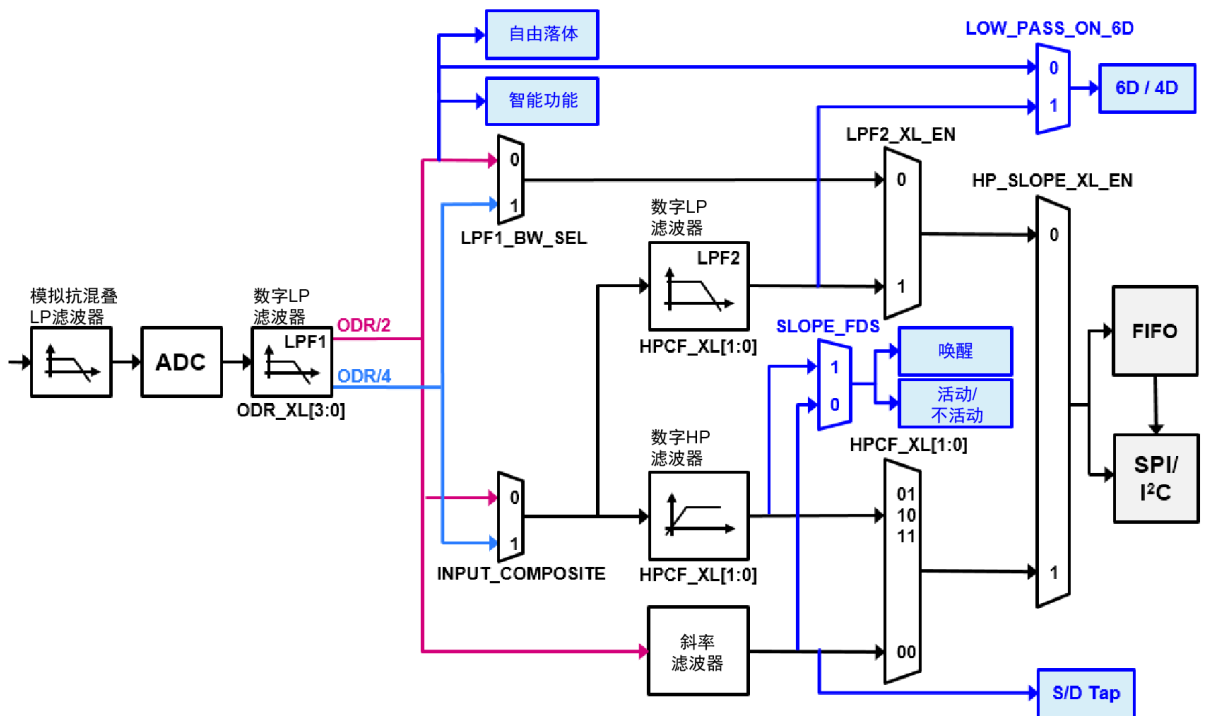
表 8. 加速度计模拟滤波器带宽

加速度计 ODR [Hz]	模拟滤波器 BW [Hz]
≥ 1666	1500
< 1666	400

通过将 CTRL1_XL 寄存器中的 BW0_XL 位置为 1，在加速度计 $ODR \geq 1666$ Hz 的情况下，模拟滤波器带宽也可以设置为 400 Hz。

数字 LPF1 滤波器可提供两个具有不同截止频率的输出；可以通过 CTRL1_XL 寄存器的 LPF1_BW_SEL 位和 CTRL8_XL 寄存器的 INPUT_COMPOSITE 位选择所需的 LPF1 输出。

图 2. 加速度计滤波链



参照图 2. 加速度计滤波链，LPF1 滤波器的“ODR/2”输出的截止频率在高性能模式下等于 ODR/2，在低功耗/正常模式下等于 740Hz。无论选择何种功耗模式，“ODR/4”输出的截止频率始终等于 ODR/4。这些图中的智能功能模块是指计步器、步进检测器和步进计数器、大幅运动和倾斜功能，如第 6 节 嵌入功能中所述。

最后，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。

CTRL8_XL 寄存器可用于配置复合滤波器组和加速度计滤波链的总带宽，如表 9. 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 [1:0] = 00b，则带宽表示斜率滤波器带宽；如果 HPCF_XL [1:0] = 01b / 10b / 11b，则带宽表示 HP 滤波器带宽。

表 9. 加速度计带宽选择 对于加速度计滤波链的各种配置，针对要丢弃的样本还提供了最大（最坏情况）的稳定时间。第 3.9 节 加速度计和陀螺仪开启/关断时间中有进一步的细节描述。

表 9. 加速度计带宽选择

HP_SLOPE_XL_EN	LPF2_XL_EN	LPF1_BW_SEL	HPCF_XL[1:0]	INPUT_COMPOSITE	带宽	稳定时间的最大值 ⁽¹⁾ (要丢弃的采样数)
0 (低通路径)	0	0	-	-	ODR/2	14
		1	-	-	ODR/4	14
	1	-	00	1 (低噪声)	ODR/50	40
			01		ODR/100	80
			10	0 (低延迟)	ODR/9	15
			11		ODR/400	320
			00	0	ODR/4	14
1 (高通路径)	-	-	01		ODR/100	80
			10		ODR/9	15
			11		ODR/400	320

1. 最终值的 99% 时的稳定时间

通过将 HP_SLOPE_XL_EN 位置为 0，可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 [1:0] 字段来设置加速度计链的总带宽。

通过将 CTRL8_XL 寄存器的 LOW_PASS_ON_6D 位置为 1，LPF2 低通滤波器还可用于 6D/4D 功能。

通过将 HP_SLOPE_XL_EN 位置为 1，可以选择复合滤波器模块的高通路径：HPCF_XL [1:0] 字段除了能用于使能 LPF1 滤波器之外，还可以使能斜率滤波器（HPCF_XL [1:0] = 00b 时）或数字高通滤波器（其他 HPCF_XL [1:0] 配置）。HPCF_XL [1:0] 字段也可用来选择 HP 滤波器的截止频率。

Reference 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 **Reference 模式**，必须将 CTRL8_XL 寄存器的 HP_REF_MODE 位和 HP_SLOPE_XL_EN 位置为 1，并且 HPCF_XL [1:0] 字段的值必须不等于 00b。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。

3.7.1 加速度计斜率滤波器

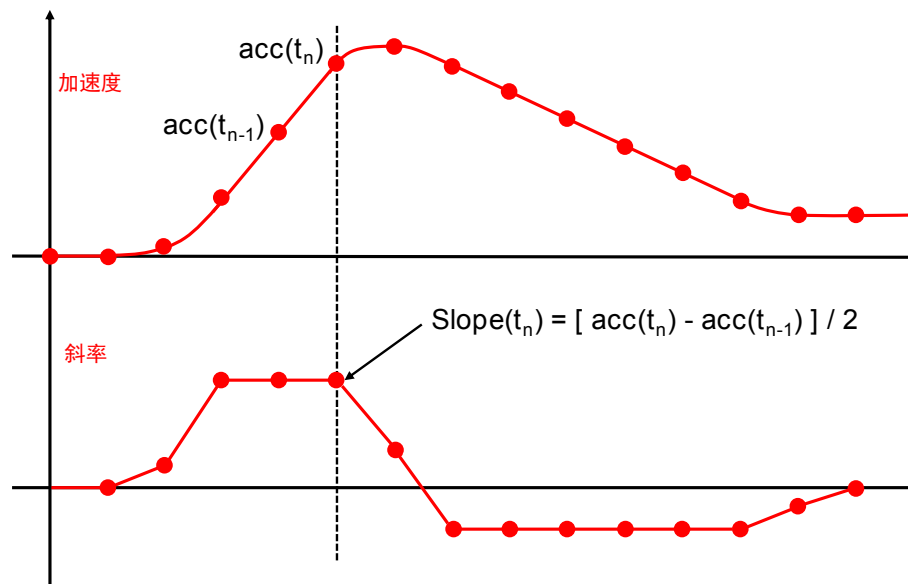
如图 2. 加速度计滤波链中所示，LSM6DSL 器件嵌入了一个数字斜率滤波器，该滤波器还可用于某些嵌入式功能，如单/双击识别、唤醒检测和活动/不活动。

该斜率滤波器输出数据利用以下公式进行计算：

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

图 3. 加速度计斜率滤波器中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

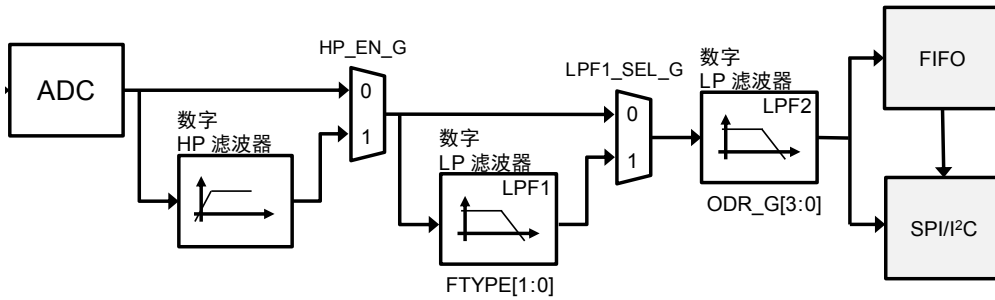


3.8 陀螺仪带宽

在 LSM6DSL 器件中，陀螺仪滤波链依赖于所使用的连接模式。

陀螺仪滤波链配置如图 4. 陀螺仪数字链中所示。它是三个滤波器的级联：可选数字高通滤波器（HPF）、可选数字低通滤波器（LPF1）和数字低通滤波器（LPF2）。

图 4. 陀螺仪数字链



可通过将 CTRL7_G 寄存器的 HP_EN_G 位置为 1，来使能数字高通滤波器。根据下表，可以通过 CTRL7_G 寄存器的 HPM_G [1:0] 字段，来选择数字 HP 滤波器的截止频率。

注：嵌入式 HP 滤波器仅能在高性能模式下使用。如果陀螺仪配置为低功耗/正常模式，则无论 CTRL7_G 寄存器的 HP_G_EN 位配置为何，该高通滤波器都要被 bypass。

表 10. 陀螺仪数字 HP 滤波器截止频率选择

HPM_G[1:0]	高通滤波器截止频率 [Hz]
00	0.016
01	0.065
10	0.260
11	1.040

数字 LPF1 滤波器可以通过将 CTRL4_C 寄存器的 LPF1_SEL_G 位置为 1 来使能，其带宽可以通过 CTRL6_C 寄存器的字段 FTYPE_ [1:0] 来选择。

注：数字 LPF1 滤波器仅能在高性能模式下使用。如果陀螺仪配置为低功耗/正常模式，则无论 CTRL4_C 寄存器中的 LPF1_SEL_G 位配置为何，LPF1 滤波器都要被 bypass。

数字 LPF2 滤波器不能由用户配置（不管选择的功耗模式如何），其截止频率取决于所选的陀螺仪 ODR。当陀螺仪 ODR 等于 6.66kHz 时，LPF2 滤波器被 bypass。

下表汇总了在 CTRL4_C 寄存器的 LPF1_SEL_G 位和 CTRL6_C 寄存器的 FTYPE_ [1:0] 的不同配置下，总的陀螺仪带宽。

表 11. 陀螺仪总带宽选择

陀螺仪 ODR [Hz]	LPF1_SEL_G	FTYPE[1:0]	截止频率[Hz]（@ 20 Hz 时的相位延迟）
12.5	0	-	4
	1	00	4
	1	01	4
	1	10	4
	1	11	4
26	0	-	8



陀螺仪 ODR [Hz]	LPF1_SEL_G	FTYPE[1:0]	截止频率[Hz] (@ 20 Hz 时的相位延迟)
26	1	00	8
	1	01	8
	1	10	8
	1	11	8
52	0	-	17
	1	00	17 (144°)
	1	01	17 (146°)
	1	10	17 (149°)
	1	11	17 (142°)
104	0	-	33
	1	00	33 (75°)
	1	01	33 (77°)
	1	10	33 (79°)
	1	11	33 (73°)
208	0	-	67
	1	00	67 (40°)
	1	01	67 (42°)
	1	10	67 (45°)
	1	11	67 (39°)
416	0	-	137
	1	00	138 (23°)
	1	01	131 (25°)
	1	10	121 (28°)
	1	11	138 (21°)
833	0	-	312
	1	00	245 (14°)
	1	01	195 (17°)
	1	10	155 (19°)
	1	11	293 (13°)
1666	0	-	988
	1	00	315 (10°)
	1	01	224 (12°)
	1	10	168 (15°)
	1	11	505 (8°)
3333	0	-	1161
	1	00	343 (8°)
	1	01	234 (10°)
	1	10	172 (12°)
	1	11	925 (6°)
6666	0	-	1250
	1	00	351 (7°)
	1	01	237 (9°)
	1	10	173 (11°)

陀螺仪 ODR [Hz]	LPF1_SEL_G	FTYPE[1:0]	截止频率[Hz] (@ 20 Hz 时的相位延迟)
6666	1	11	937 (5°)

3.9 加速度计和陀螺仪开启/关断时间

加速度计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，切换加速度计的功耗模式时，或改变加速度计的 ODR 时，必须考虑滤波器的稳定时间。

用于切换加速度计功耗模式或加速度计 ODR 的最大总开启/关闭时间（LPF2 和 HP 滤波器禁用）显示于表 12. 加速度计开启/关闭时间（LPF2 和 HP 禁用）

注：加速度计 ODR 时序不受功耗模式更改的影响（新的配置在当前周期完成后生效）。

表 12. 加速度计开启/关闭时间（LPF2 和 HP 禁用）

起始模式	目标模式	最大开启/关闭时间 ⁽¹⁾
省电	低功耗/正常	See 表 13. 要丢弃的加速度计样本
省电	高性能	See 表 13. 要丢弃的加速度计样本
低功耗/正常	高性能	见表 13. 要丢弃的加速度计样本 + 丢弃 1 个额外样本
低功耗/正常	低功耗/正常（ODR 改变）	See 表 13. 要丢弃的加速度计样本
高性能	低功耗/正常	见表 13. 要丢弃的加速度计样本 + 丢弃 1 个额外样本
高性能 @ ODR ≤ 833 Hz	高性能 @ ODR ≤ 833 Hz	见表 13. 要丢弃的加速度计样本 + 丢弃 1 个额外样本
高性能 @ ODR ≤ 833 Hz	高性能 @ ODR > 833 Hz	见表 13. 要丢弃的加速度计样本 + 丢弃 1 个额外样本
高性能 @ ODR > 833 Hz	高性能 @ ODR ≤ 833 Hz	见表 13. 要丢弃的加速度计样本 + 丢弃 1 个额外样本
高性能 @ ODR > 833 Hz	高性能 @ ODR > 833 Hz	丢弃 5 个样本
低功耗/正常/高性能	省电	1 μs

1. @ 最终值的 99% 时的稳定时间

表 13. 要丢弃的加速度计样本

目标模式 加速度计 ODR [Hz]	要丢弃的采样数 (LPF1_BW_SEL = 0 和 LPF2_XL_EN = 0 和 HP_SLOPE_XL_EN = 0)	要丢弃的采样数 (LPF1_BW_SEL = 1 和 LPF2_XL_EN = 0 和 HP_SLOPE_XL_EN = 0) OR (HPCF_XL = 00 和 HP_SLOPE_XL_EN = 1)
1.6 (低功耗)	0 (第一个采样正确)	1
12.5 (低功耗)	0 (第一个采样正确)	1
26 (低功耗)	0 (第一个采样正确)	1
52 (低功耗)	0 (第一个采样正确)	1
104 (正常)	0 (第一个采样正确)	1
208 (正常)	0 (第一个采样正确)	1
12.5 (高性能)	0 (第一个采样正确)	1

目标模式 加速度计 ODR [Hz]	要丢弃的采样数 (LPF1_BW_SEL = 0 和 LPF2_XL_EN = 0 和 HP_SLOPE_XL_EN = 0)	要丢弃的采样数 (LPF1_BW_SEL = 1 和 LPF2_XL_EN = 0 和 HP_SLOPE_XL_EN = 0) OR (HPCF_XL = 00 和 HP_SLOPE_XL_EN = 1)
26 (高性能)	0 (第一个采样正确)	1
52 (高性能)	1	1
104 (高性能)	1	2
208 (高性能)	1	2
416 (高性能)	1	2
833 (高性能)	1	2
1666 (高性能)	2	2
3333 (高性能)	3	4
6666 (高性能)	13	13

切换其模式或改变陀螺仪 ODR 时，还必须考虑陀螺仪传感器的开启/关断时间。

用于切换陀螺仪功耗模式或陀螺仪 ODR 的最大总开启/关闭时间（HP 滤波器禁用）显示于表 14. 陀螺仪开启/关闭时间（HP 禁用）。

注：陀螺仪 ODR 时序不受功耗模式更改的影响（新的配置在当前周期完成后生效）。

表 14. 陀螺仪开启/关闭时间（HP 禁用）

起始模式	目标模式	最大开启/关闭时间 ⁽¹⁾
省电	睡眠	70 ms
省电	低功耗/正常	70 ms + 丢弃 1 个样本
省电	高性能	70 ms + 见表 15. 要丢弃的陀螺仪样本（LPF1 禁用）或 表 16. 对于所有 ODR，要丢弃的陀螺仪样本（LPF1 使能）
睡眠	低功耗/正常	丢弃 1 个样本
睡眠	高性能	见表 15. 要丢弃的陀螺仪样本（LPF1 禁用）或 表 16. 对于所有 ODR，要丢弃的陀螺仪样本（LPF1 使能）
低功耗/正常	高性能	丢弃 2 个样本
低功耗/正常	低功耗/正常 (ODR 改变)	丢弃 1 个样本
高性能	低功耗/正常	丢弃 1 个样本
高性能	高性能 (ODR 改变)	丢弃 2 个样本
低功耗/正常/高性能	省电	1 μs, 如果 XL 和 Gyro 在下电模式中 300 μs, 如果 XL 不在下电模式中

1. @ 最终值的 99% 时的稳定时间

表 15. 要丢弃的陀螺仪样本（LPF1 禁用）

陀螺仪 ODR [Hz]	要丢弃的采样数
12.5 Hz	2
26 Hz	3
52 Hz	3
104 Hz	3
208 Hz	3
416 Hz	3
833 Hz	3
1.66 kHz	135
3.33 kHz	270
6.66 kHz	540

表 16. 对于所有 ODR，要丢弃的陀螺仪样本（LPF1 使能）

陀螺仪 ODR [Hz]	FTYPE[1:0]	要丢弃的采样数
12.5 Hz	00	2
	01	2
	10	2
	11	2
26 Hz	00	3
	01	3
	10	3
	11	3
52 Hz	00	3
	01	3
	10	3
	11	3
104 Hz	00	4
	01	4
	10	4
	11	4
208 Hz	00	4
	01	4
	10	5
	11	4
416 Hz	00	5
	01	6
	10	6
	11	5
833 Hz	00	7
	01	8
	10	9



陀螺仪 ODR [Hz]	FTYPE[1:0]	要丢弃的采样数
833 Hz	11	6
1.66 kHz	00	135
	01	135
	10	135
	11	135
3.33 kHz	00	270
	01	270
	10	270
	11	270
6.66 kHz	00	540
	01	540
	10	540
	11	540

4 模式 1 - 读取输出数据

4.1 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。当启动程序完成时，即，约 15 ms 后，加速度计和陀螺仪自动进入下电模式。

要启用加速度计并采集加速度数据，需要通过 CTRL1_XL 寄存器选择某一种工作模式。

以下通用线序可用于配置加速度计：

1. 写 CTRL1_XL = 60h // Acc = 416Hz（高性能模式）
2. 写 INT1_CTRL = 01h // INT1 上，Acc 数据准备就绪中断

要启用陀螺仪并采集角速度数据，需要通过 CTRL2_G 来选择某一种工作模式。

以下通用序列可用于配置陀螺仪：

1. 写 CTRL2_G = 60h // Gyro = 416Hz（高性能模式）
2. 写 INT1_CTRL = 02h // INT1 上，Gyro 数据准备就绪中断

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。当一组新数据在加速度计输出上可用时，XLDA 位被置为 1；当一组新数据在陀螺仪输出上可用时，GDA 位被置为 1。

对于加速度计（陀螺仪也是类似的），应当按照如下步骤对输出寄存器进行读取：

1. 读 STATUS
2. 如果 XLDA = 0，则进入 1
3. 读 OUTX_L_XL
4. 读 OUTX_H_XL
5. 读 OUTY_L_XL
6. 读 OUTY_H_XL
7. 读 OUTZ_L_XL
8. 读 OUTZ_H_XL
9. 数据处理
10. 跳到步骤 1

4.3 使用数据准备就绪信号

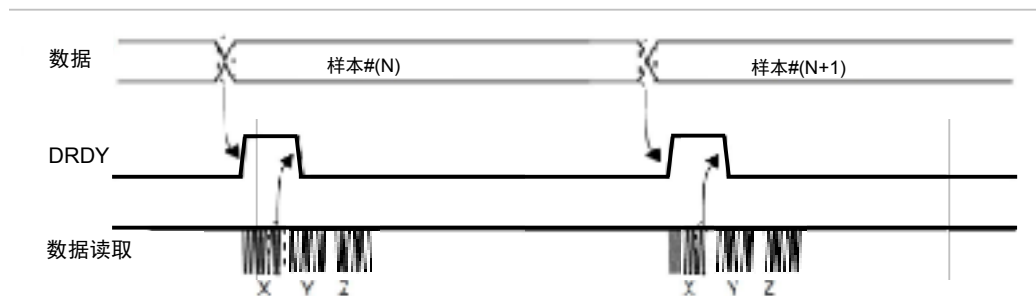
该器件可配置为具有一个 HW 信号，以确定新的一组测量数据何时可以读取。

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。

对于陀螺仪传感器，数据准备就绪信号由 STATUS_REG 寄存器的 GDA 位表示。通过将 INT1_CTRL 寄存器的 INT1_DRDY_G 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 DRDY_PULSE_CFG 寄存器的 DRDY_PULSED 位被置为 0（默认值），则数据就绪信号被锁存，并且当其中某一个的较高部分（对于加速度计，为 29h、2Bh、2Dh；对于陀螺仪，为 23h、25h、27h）被读取时，中断复位。如果 DRDY_PULSE_CFG 寄存器的 DRDY_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 75μs。脉冲模式不适用于总是锁存的 XLDA 和 GDA 位。

图 5. 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，则加速度计和陀螺仪数据准备就绪信号会被屏蔽，直至完成传感器滤波器稳定稳定。

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/陀螺仪无效采样等于 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 和陀螺仪 LPF2 数字滤波器稳定时间起作用。

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

如果读取加速度计/陀螺仪数据特别慢，并且不能（或者不需要）与 STATUS_REG 寄存器中的 XLDA/GDA 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）。特别地，当 BDU 被激活时，每个数据寄存器中始终包含器件产生的最新输出数据，但是，如果发起了对给定寄存器的读取（即 OUTX_H_XL(G) 和 OUTX_L_XL(G)，OUTY_H_XL(G) 和 OUTY_L_XL(G)，OUTZ_H_XL(G) 和 OUTZ_L_XL(G)），这时读数不能更新，直至数据的 MSB 和 LSB 部分均被读取。

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。

4.5 认识输出数据

测得的加速度数据被发送到 OUTX_H_XL、OUTX_L_XL、OUTY_H_XL、OUTY_L_XL、OUTZ_H_XL 和 OUTZ_L_XL 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。

测得的角速率数据被发送到 OUTX_H_G、OUTX_L_G、OUTY_H_G、OUTY_L_G、OUTZ_H_G 和 OUTZ_L_G 寄存器。这些寄存器分别容纳角速率信号在 X、Y 和 Z 轴上的最高有效部分和最低有效部分。

X、Y、Z 轴的完整输出数据由 OUTX_H_XL(G) & OUTX_L_XL(G)，OUTY_H_XL(G) & OUTY_L_XL(G)，OUTZ_H_XL(G) & OUTZ_L_XL(G) 合并提供，表示为 2 的补码。

加速度数据和角速率数据均表示为 16 bit 的数字。

4.5.1 大小端选择

LSM6DSL 允许输出数据寄存器的高低部分（即，OUTX_H_XL(G) 与 OUTX_L_XL(G)，以及 OUT_TEMP_H 与 OUT_TEMP_L）进行交换，以便兼容小端和大端数据表示。

“小端模式”表示数字的低位字节存储在存储器的最低地址中，高位字节存储在最高地址中。这种模式对应于 TCTRL3_C 寄存器的 BLE 位置为 0（默认配置）。

相反，“大端模式”表示数字的高位字节存储在存储器的最低地址中，低位字节存储在最高地址中。这种模式对应于 TCTRL3_C 寄存器的 BLE 位置为 1。

4.5.2 输出数据示例

表 17. 输出数据寄存器内容 vs. 加速度 ($FS_{XL} = 2\text{ g}$) 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。

表 18. 输出数据寄存器内容 vs. 角速率 ($FS_G = \pm 250\text{ dps}$) 提供了陀螺仪数据的一些基本示例, 当器件施加了给定的角速率时, 在数据寄存器中读取这些数据。

下表所列的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,), 实际显示受 BLE 位的影响。

表 17. 输出数据寄存器内容 vs. 加速度 ($FS_{XL} = 2\text{ g}$)

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

表 18. 输出数据寄存器内容 vs. 角速率 ($FS_G = \pm 250\text{ dps}$)

角速率值	BLE = 0		BLE = 1	
	寄存器地址			
	OUTX_H_G (23h)	OUTX_L_G (22h)	OUTX_H_G (23h)	OUTX_L_G (22h)
0 dps	00h	00h	00h	00h
100 dps	2Ch	A4h	A4h	2Ch
200 dps	59h	49h	49h	59h
-100 dps	D3h	5Ch	5Ch	D3h
-200 dps	A6h	B7h	B7h	A6h

4.6 加速度计偏移寄存器

LSM6DSL 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 g 偏移校正, 或者用来将偏移量应用于加速度计输出数据。

在偏移寄存器中设置的偏移值从测得的 Z 轴加速度值内部减去, 并内部添加到测得的 X 轴和 Y 轴加速度值上; 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 LSM6DSL 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。

4.7.1 FIFO 输出寄存器环行

对 FIFO 输出寄存器 FIFO_DATA_OUT_L (3Eh) 和 FIFO_DATA_OUT_H (3Fh) 执行多字节读取操作时, 环行功能自动使能。

4.7.2 状态寄存器环行

LSM6DSL 器件的状态寄存器也可以实现环行功能，能够识别在一次多字节读取中是否产生了新数据或是否检测到新的中断事件。

通过将 CTRL7_G 寄存器的 ROUNDING_STATUS 位置为 1，可以使能状态寄存器上的环形功能。当该功能使能时，通过多字节读取操作，正在读取的寄存器地址在 WAKE_UP_SRC (1Bh)、TAP_SRC (1Ch)、D6D_SRC (1Dh)、STATUS_REG (1Eh) 和 FUNC_SRC1 (53h) 上自动循环，并返回到 WAKE_UP_SRC (1Bh)。

4.7.3 传感器输出寄存器环行

以下输出寄存器组也可以使能环行功能：

- 陀螺仪输出寄存器，从 OUTX_L_G (22h) 到 OUTZ_H_G (27h)；
- 加速度计输出寄存器，从 OUTX_L_XL (28h) 到 OUTZ_H_XL (2Dh)；
- 第一组传感器集合 (sensor hub) 输出寄存器，从 SENSORHUB1_REG (2Eh) 到 SENSORHUB6_REG (33h)；
- 第二组传感器集合 (sensor hub) 输出寄存器，从 SENSORHUB7_REG (34h) 到 SENSORHUB12_REG (39h)。

输出寄存器环行模式可利用 CTRL5_C 寄存器的 ROUNDING[2:0] 位进行配置，如表 19. 输出寄存器环行模式所示。

表 19. 输出寄存器环行模式

ROUNDING[2:0]	环行模式
000	无环行
001	只有加速度计
010	只有陀螺仪
011	陀螺仪 + 加速度计
100	只有第 1 组传感器集合 (sensor hub)
101	加速度计 + 第 1 组传感器集合 (sensor hub)
110	陀螺仪 + 加速度计 + 第 1 组传感器集合 (sensor hub) + 第 2 组传感器集合 (sensor hub)
111	陀螺仪 + 加速度计 + 第 1 组传感器集合 (sensor hub)

4.8 边沿感应和电平感应数据使能 (DEN)

通过使能 CTRL6_C 中的 TRIG_EN、LVL1_EN 和 LVL2_EN 位, LSM6DSL 可允许外部触发电平识别。有四种不同模式可供选择 (表 20. DEN 配置):

- 边沿感应触发模式
- 电平感应触发模式
- 电平感应锁存模式
- 电平感应 FIFO 使能模式

数据使能 (DEN) 输入信号在 INT2 引脚上被驱动, 当其中一个触发模式使能时, INT2 被配置为输入引脚。

DEN 功能仅在陀螺仪数据上被默认激活。要将此功能扩展到加速度计数据, CTRL4_C 中的 DEN_XL_EN 位必须置为 1。

DEN 有效电平默认为低电平。通过将 CTRL5_C 中的 DEN_LH 位置为 1, 可以将其更改为高电平有效。

表 20. DEN 配置

TRIG_EN	LVL1_EN	LVL2_EN	功能
1	0	0	边沿感应触发模式
0	1	0	电平感应触发模式
0	1	1	电平感应锁存模式
1	1	0	电平感应 FIFO 使能模式

4.8.1 边沿感应触发模式

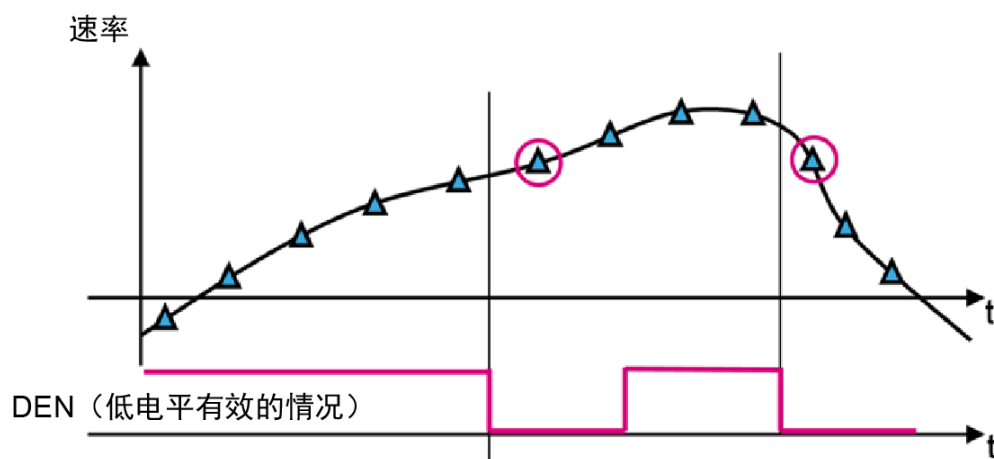
可以通过将 CTRL6_C 中的 TRIG_EN 位置为 1, 并将 CTRL6_C 中的 LVL1_EN、LVL2_EN 位置为 0, 来启用边沿感应触发模式。

只有当低通滤波器 LPF2 禁用 (CTRL8_XL 寄存器中的 LPF2_XL_EN = 0) 时, 边沿感应触发才工作。

一旦使能了边沿感应触发模式, FIFO 缓冲器和输出寄存器就会被填充上 DEN 输入信号每个上升沿 (如果 DEN_LH 位等于 1) 或下降沿 (如果 DEN_LH 位等于 0) 之后所获取的第一个采样。

图 6. 边沿感应触发模式, DEN 低电平有效 用红色圆圈表示在下降沿之后 (DEN 低电平有效) 采集的采样。

图 6. 边沿感应触发模式, DEN 低电平有效



边沿感应触发模式启用后, 仅对陀螺仪输出寄存器起作用。DRDY_G 仅与降低采样数据相关, 而加速计输出寄存器和 DRDY_XL 则根据 ODR_XL 进行更新。如果 DEN_XL_EN 位置为 1, 则加速计传感器也会被降低采样。这种情况下, 陀螺仪和加速度计必须设置为 ODR 相同的组合模式。通过将陀螺仪设置为下电模式, 可以使用加速度计独立模式。这种情况下, DRDY_XL 只与降低采样数据相关。

请注意, 在更新数据寄存器之前, DEN 触发器会内部锁存: 如果在此事件之后发生触发事件, 那么 DEN 将在下一个 ODR 中进行确认。

对于 FIFO 中的边沿感应触发器，有三种可能的配置，描述如下：

1. 只有陀螺仪处于触发模式，不能保存在 FIFO 中：这种情况下，FIFO 只与加速度计相关，并且正常工作。
2. 只有陀螺仪处于触发模式，存储于 FIFO 中：这种情况下，FIFO_CTRL3 寄存器的陀螺仪抽取位 DEC_FIFO_GYRO [2:0] 必须置为 001 (FIFO 中的陀螺仪传感器不带抽取)。要实现它，FIFO 需由外部触发器驱动。采用这种配置时，由于在触发发生时也写入了加速度计数据，所以可能发生加速度计数据重复或丢失。
3. 陀螺仪和加速度计处于触发模式，并保存在 FIFO 中：通过将 DEN_XL_EN 置为 1，并将 FIFO_CTRL3 寄存器的陀螺仪和加速度计抽取位 DEC_FIFO_GYRO [2:0] 和 DEC_FIFO_XL [2:0] 置为 001 (FIFO 中的陀螺仪和加速度计无抽取)。这种情况下，当触发发生时，两个传感器的数据都会写入 FIFO 中。

边沿感应触发模式可支持相机帧与陀螺仪采样同步，用于电子图像稳定 (Electrical Image Stabilization, EIS) 应用。来自相机模块的同步信号必须连接到 INT2 引脚。

在下面所示的例子中，FIFO 被配置为将陀螺仪数据和加速度计数据都存储在 FIFO 缓冲器中；当 DEN 信号切换时，数据在上升沿写入 FIFO 中。

- | | |
|------------------------|---|
| 1. 将 09h 写入 FIFO_CTRL3 | // 使能 FIFO 中的陀螺仪和加速度计 (无抽取) |
| 2. 将 26h 写入 FIFO_CTRL5 | // 将 FIFO 设置为 Continue 模式，FIFO ODR = 104 Hz |
| 3. 将 80h 写入 CTRL6_C | // 使能边沿感应触发 |
| | // INT2 引脚切换为输入模式 (DEN 信号) |
| 4. 将 80h 写入 CTRL4_C | // 将 DEN 功能扩展到加速计传感器 |
| 5. 将 40h 写入 CTRL1_XL | // 开启加速度计：ODR_XL = 104 Hz, FS_XL = $\pm 2g$ |
| 6. 将 4Ch 写入 CTRL2_G | // 开启陀螺仪 |
| | // ODR_G = 104 Hz, FS_G = ± 2000 dps |

4.8.2

电平感应触发模式

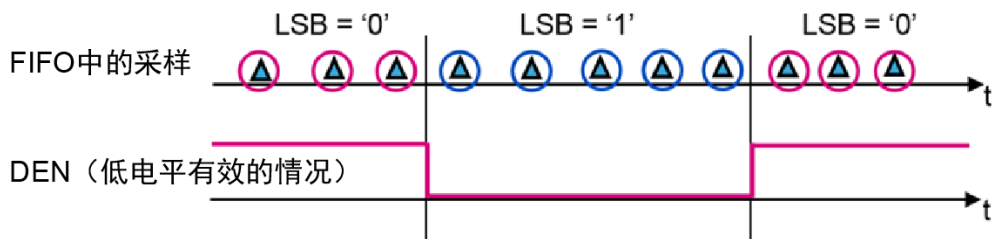
可以通过将 CTRL6_C 中的 LVL1_EN 位置为 1，并将 CTRL6_C 中的 TRIG_EN、LVL2_EN 位置为 0，来使能电平感应触发模式。

一旦使能电平感应触发模式，那么如果 DEN 电平有效，则所选数据 (在输出寄存器和 FIFO 中) 的 LSB 位将被替换为 1；如果 DEN 电平未被激活，则所选数据的 LSB 位会被替换为 0。所选数据可以是加速度计或陀螺仪传感器的 X、Y、Z 轴，可通过 CTRL9_XL 中的 DEN_X、DEN_Y、DEN_Z 和 DEN_XL_G 位来定义。

所有数据都可以根据 FIFO 设置存储在 FIFO 中。

图 7. 电平感应触发模式，DEN 低电平有效 以红色圆圈显示 LSB = 0 (DEN 未激活) 时存储在 FIFO 中的样本，蓝色圆圈表示 LSB = 1 (DEN 激活) 时存储在 FIFO 中的样本。

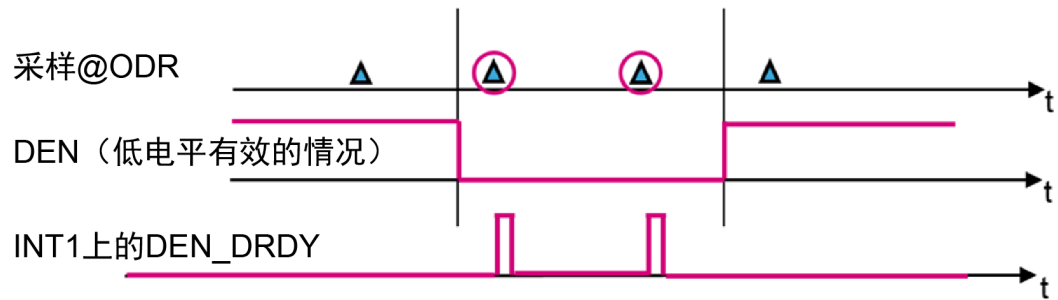
图 7. 电平感应触发模式，DEN 低电平有效



当电平感应触发模式使能时，DEN 信号也可用于过滤 INT1 引脚上的数据就绪信号。只有在 DEN 引脚处于激活状态时，INT1 才会显示数据就绪信息。为此，CTRL4_C 寄存器的 DEN_DRDY_INT1 位必须置为 1。中断信号可以根据 DRDY_PULSE_CFG 寄存器的 DRDY_PULSED 位进行锁存或脉冲。

图 8. 电平感应触发模式，DEN 低电平有效，DEN_DRDY 在 INT1 上 显示了当 DEN 电平低 (有效状态) 时 INT1 上的数据就绪示例。

图 8. 电平感应触发模式，DEN 低电平有效，DEN_DRDY 在 INT1 上



4.8.3

电平感应锁存模式

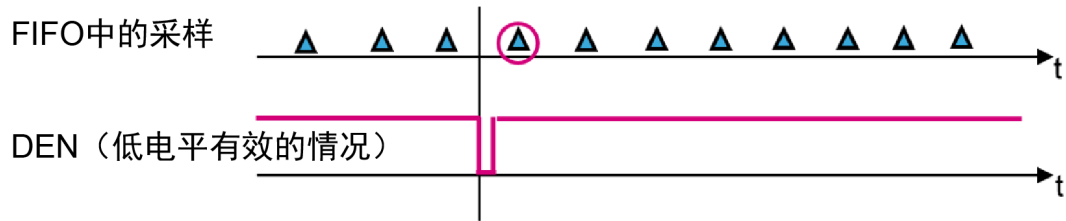
可以通过将 CTRL6_C 中的 LVL1_EN 和 LVL2_EN 位设置为 1，并将 CTRL6_C 中的 TRIG_EN 位设置为 0，来使能电平感应锁存模式。

当使能电平感应锁存模式时，所选数据（在输出寄存器和 FIFO 中）的 LSB 位通常设置为 0，并且仅在 DEN 引脚上一个脉冲之后的第一个采样时变为 1。

数据可以通过 CTRL9_XL 中的 DEN_X、DEN_Y、DEN_Z 和 DEN_XL_G 位来选择。

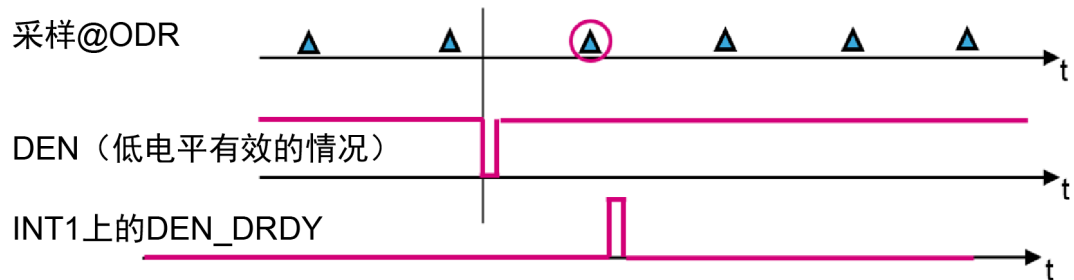
图 9. 电平感应锁存模式，DEN 低电平有效 显示了 DEN 低电平有效时电平感应锁存模式的示例。在 DEN 引脚上的脉冲之后，带红色圆圈的采样在 LSB 位上的值为 1。所有其他样本的 LSB 位都为 0。

图 9. 电平感应锁存模式，DEN 低电平有效



当使能电平感应锁存模式，并且 CTRL4_C 寄存器的 DEN_DRDY_INT1 位设置为 1 时，INT1 引脚上会产生一个脉冲，对应于 DEN 脉冲出现后产生的第一个采样可用（见图 10. 电平感应锁存模式，DEN 低电平有效，DEN_DRDY 在 INT1 上）。

图 10. 电平感应锁存模式，DEN 低电平有效，DEN_DRDY 在 INT1 上



4.8.4 启用电平感应 FIFO

可通过将 CTRL6_C 中的 TRIG_EN 和 LVL1_EN 位设置为 1，并将 CTRL6_C 中的 LVL2_EN 位设置为 0，来启用电平感应 FIFO 使能模式。

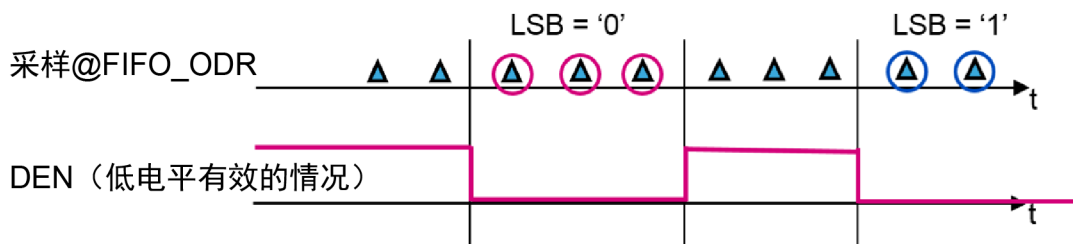
一旦启用了电平感应 FIFO 使能模式，只有当 DEN 引脚为有效状态时，才会将数据存储到 FIFO 中。

此模式下，所选数据的 LSB 位（在输出寄存器和 FIFO 中），对于奇数 DEN 事件，会替代为 0，对于偶数 DEN 事件，会替代为 1。该功能可将当前 DEN 激活窗口期间存储在 FIFO 中的数据与下一个 DEN 激活窗口期间存储在 FIFO 中的数据区分开来。

所选数据可以是加速度计或陀螺仪传感器的 X、Y、Z 轴。数据可以通过 CTRL9_XL 中的 DEN_X、DEN_Y、DEN_Z 和 DEN_XL_G 位来选择。

图 11. 电平感应 FIFO 使能模式，DEN 低电平有效中显示了电平感应 FIFO 使能模式的示例，红圈表示存储在 FIFO 中、LSB 位为 0 的采样，而蓝圈表示 LSB 位为 1 的采样。

图 11. 电平感应 FIFO 使能模式，DEN 低电平有效



4.8.5 用于 DEN 冲压的 LSB 选择

当使用电平感应模式（触发或锁存）时，可以选择哪个 LSB 必须包含与 DEN 引脚特性有关的信息。根据 CTRL9_XL 寄存器的 DEN_X、DEN_Y、DEN_Z 和 DEN_XL_G 位，可以在加速度计或陀螺仪轴上标记此信息。将 DEN_X、DEN_Y、DEN_Z 位置为 1 时，DEN 信息会被标记在由 DEN_XL_G 位选择的传感器对应轴的 LSB 中。通过将 DEN_XL_G 设置为 0，可将 DEN 信息标记在选定的陀螺仪轴中，而将 DEN_XL_G 设置为 1，可将 DEN 信息标记在所选的加速度计轴中。

默认情况下，这些位被配置为包含所有陀螺仪轴上的信息。

5 中断生成

LSM6DSL 器件中，中断产生仅以加速度计数据为基础，因此要产生中断，加速度计传感器必须设置为活动工作模式（不能处于下电模式）；陀螺仪传感器可配置为下电模式，因为它与中断产生无关。

可对中断发生器进行配置，来检测：

- 自由落体；
- 唤醒；
- 6D/4D 方向检测；
- 单击和双击感测；
- 活动/不活动识别。

此外，LSM6DSL 能够高效运行 Android 中特定的传感器相关功能，可节能并具有更快的反应速度。以下功能仅在使用加速度计的硬件中实现：

- 大幅运动检测；
- 相对倾斜；
- 绝对手腕倾斜；
- 计步功能；
- 时间戳。

所有这些中断信号，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。

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。

TAP_CFG 的 LIR 位可支持中断信号应用锁存模式。LIR 位置为 1 时，中断引脚一旦被产生，就必须通过读取相关中断源寄存器才能将其复位。如果 LIR 位置为 0，则当不再检测到中断条件或一定时间后，中断信号可自动复位。

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 21. INT1_CTRL 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
INT1_ STEP_ DETECTOR	INT1_ SIGN_ MOT	INT1_ FULL_ FLAG	INT1_ FIFO_ OVR	INT1_ FTH	INT1_ BOOT	INT1_ DRDY_G	INT1_ DRDY_ XL

- INT1_STEP_DETECTOR: INT1 上计数器步伐检测中断
- INT1_SIGN_MOT: INT1 上大幅运动中断
- INT1_FULL_FLAG: INT1 上 FIFO 全满标志中断
- INT1_FIFO_OVR: INT1 上 FIFO 溢出标志中断
- INT1_FTH: INT1 上 FIFO 阈值中断
- INT1_BOOT: INT1 上启动中断
- INT1_DRDY_G: INT1 上陀螺仪数据准备就绪
- INT1_DRDY_XL: INT1 上加速度计数据准备就绪

表 22. MD1_CFG 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
INT1_ INACT_ STATE	INT1_ SINGLE_ TAP	INT1_ WU	INT1_ FF	INT1_ DOUBLE _TAP	INT1_ 6D	INT1_ TILT	INT1_ TIMER

- INT1_INACT_STATE: INT1 上非活动中断
- INT1_SINGLE_TAP: INT1 上单击中断
- INT1_WU: INT1 上唤醒中断
- INT1_FF: INT1 上自由落体中断
- INT1_DOUBLE_TAP: INT1 上双击中断
- INT1_6D: INT1 上 6D 检测中断
- INT1_TILT: INT1 上倾斜中断
- INT1_TIMER: INT1 上定时器中断

表 23. INT2_CTRL 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
INT2_ STEP_ DELTA	INT2_ STEP_ COUNT_OV	INT2_ FULL_ FLAG	INT2_ FIFO_ OVR	INT2_ FTH	INT2_ DRDY_ TEMP	INT2_ DRDY_G	INT2_ DRDY_ XL

- INT2_STEP_DELTA: INT2 上计步器步数识别时间变化量中断
- INT2_STEP_COUNT_OV: INT2 上步数计数器溢出中断
- INT2_FULL_FLAG: INT2 上 FIFO 全满标志中断
- INT2_FIFO_OVR: INT2 上 FIFO 溢出标志中断
- INT2_FTH: INT2 上 FIFO 阈值中断
- INT2_DRDY_TEMP: INT2 上温度数据准备就绪
- INT2_DRDY_G: INT2 上陀螺仪数据准备就绪
- INT2_DRDY_XL: INT2 上加速度计数据准备就绪

表 24. MD2_CFG 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
INT2_ INACT_ STATE	INT2_ SINGLE_ TAP	INT2_ WU	INT2_ FF	INT2_ DOUBLE _TAP	INT2_ 6D	INT2_ TILT	INT2_ IRON

- INT2_INACT_STATE: INT2 上非活动中断
- INT2_SINGLE_TAP: INT2 上单击中断
- INT2_WU: INT2 上唤醒中断
- INT2_FF: INT2 上自由落体中断
- INT2_DOUBLE_TAP: INT2 上双击中断
- INT2_6D: INT2 上 6D 检测中断
- INT2_TILT: INT2 上倾斜中断
- INT2_IRON: INT2 上软铁/硬铁中断

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，则须读取相关状态寄存器：WAKE_UP_SRC、D6D_SRC、TAP_SRC、FUNC_SRC1 和 FUNC_SRC2。

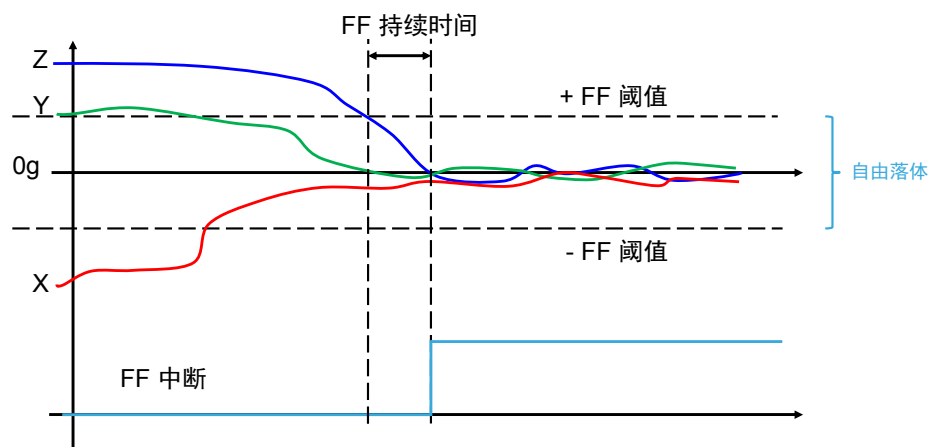
CTRL4_C 寄存器的 INT2_on_INT1 引脚能够将所有已使能的中断信号进行逻辑“与”后驱动到 INT1 引脚上（通过将该位置为 1）。当该位置为 0 时，中断信号分配到 INT1 和 INT2 引脚上。

必须通过设置 TAP_CFG 寄存器中的 INTERRUPTS_ENABLE 位来使能基本中断（6D/4D、自由落体、唤醒、点击、不活动）。

5.2 自由落体中断

自由落体检测涉及特定的寄存器配置，可以识别器件何时处于自由落体：各轴所测得的加速度均为 **0**。真实情境下，一个“自由落体”定义为大约零-**g** 水平，其中所有加速度均足够小，可以产生中断。自由落体事件检测相关的可配置的阈值和持续时间参数：阈值参数定义了自由落体的幅度；持续时间参数定义了可识别的自由落体中断事件的最小持续时间（图 12. 自由落体中断）。

图 12. 自由落体中断



通过将 TAP_CFG 寄存器中的 INTERRUPTS_ENABLE 位置为 1，可启用该自由落体中断信号，将 MD1_CFG 寄存器的 INT1_FF 位或 MD2_CFG 寄存器的 INT2_FF 位置为 1，可将该中断信号驱动至两个中断引脚上；还可通过读取 WAKE_UP_SRC 寄存器的 FF_IA 位对其进行检查。

如果锁存模式禁用（TAP_CFG 的 LIR 位置为 0），则当检测不到自由落体条件时，中断信号会自动复位。如果锁存模式启用且自由落体中断信号被驱动至中断引脚，那么当发生自由落体事件且产生了中断引脚时，必须通过读取 WAKE_UP_SRC 寄存器来将其复位。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。

FREE_FALL 寄存器用来配置阈值参数；无符号阈值与 FF_THS[2:0] 的值相关，如表 25. 自由落体阈值 LSB 值所示。此表中给出的值对于每个加速度计满量程值均有效。

表 25. 自由落体阈值 LSB 值

FREE_FALL - FF_THS[2:0]	阈值 LSB 值 [mg]
000	156
001	219
010	250
011	312
100	344
101	406
110	469
111	500

持续时间在 N/ODR_XL 中测得，其中 N 为 FREE_FALL / WAKE_UP_DUR 寄存器 FF_DUR[5:0] 字段的内容，ODR_XL 为加速度计数据率。

下面给出了自由落体事件识别的基本软件程序。

1. 将 60h 写入 CTRL1_XL // 启动加速度计
// ODR_XL = 416 Hz, FS_XL = ±2 g
2. 将 81h 写入 TAP_CFG // 使能中断并锁存中断
3. 将 00h 写入 WAKE_UP_DUR // 设置事件持续时间（FF_DUR5 位）
4. 将 33h 写入 FREE_FALL // 设置 FF 阈值（FF_THS[2:0] = 011b）
// 设置六个采样事件持续时间（FF_DUR[5:0] = 000110b）
5. 将 10h 写入 MD1_CFG // FF 中断驱动至 INT1 引脚

示例代码中利用设置为 312mg 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。FREE_FALL / WAKE_UP_DUR 寄存器的 FF_DUR[5:0] 字段像这样配置：忽略短于 $6/ODR_{XL} = 6/412 \text{ Hz} \approx 15 \text{ ms}$ 的事件，以避免错误检测。

5.3 唤醒中断

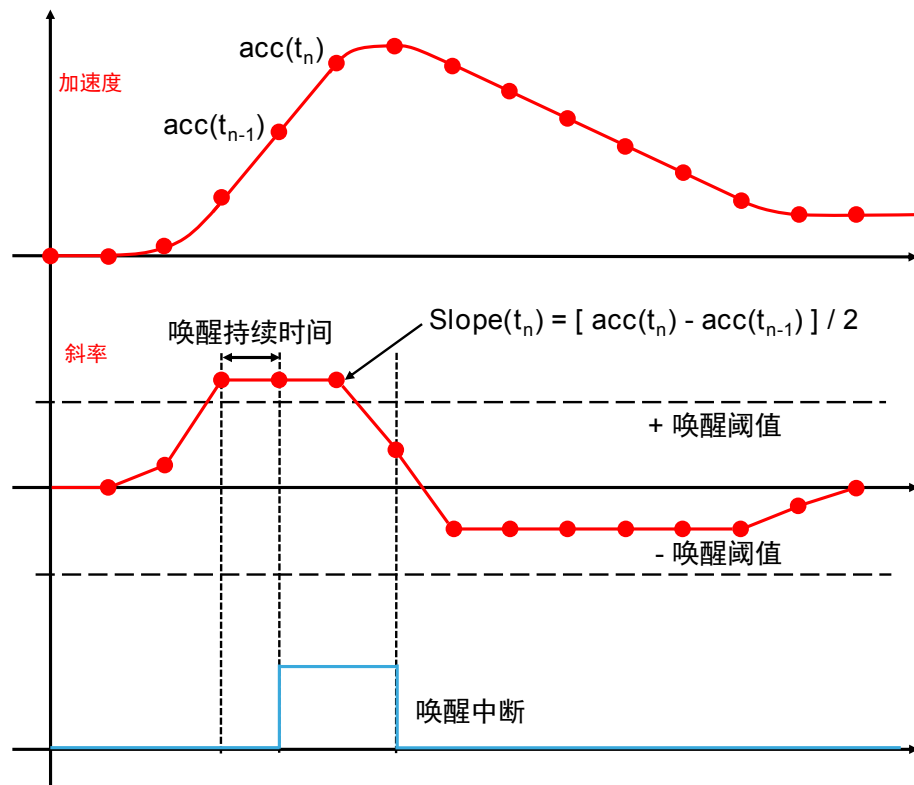
LSM6DSL 器件中，唤醒功能可利用斜率滤波器（更多详细信息参见第 3.7.1 节 加速度计斜率滤波器）或高通数字滤波器来实现，如图 2. 加速度计滤波链所示。所用滤波器可通过 TAP_CFG 寄存器的 SLOPE_FDS 位来选择：如果该位被置为 0（默认值），则使用斜率滤波器；如果被置为 1，则使用 HPF 数字滤波器。

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 13. 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 6 bit 的 1 LSB 值取决于所选加速度计满量程： $1 \text{ LSB} = (FS_{XL})/(2^6)$ 。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 $1/\text{ODR_XL}$ 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 TAP_CFG 寄存器中的 INTERRUPTS_ENABLE 位置为 1，可启用该中断信号，将 MD1_CFG 寄存器的 INT1_WU 位或 MD2_CFG 寄存器的 INT2_WU 位置为 1，可将该中断信号驱动至两个中断引脚上；还可通过读取 WAKE_UP_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 13. 唤醒中断（利用斜率滤波器）



如果锁存模式禁用（TAP_CFG 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产生了中断引脚时，必须通过读取 WAKE_UP_SRC 寄存器来将其复位。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 60h 写入 CTRL1_XL // 启动加速度计
// ODR_XL = 416 Hz, FS_XL = ±2 g
2. 将 90h 写入 TAP_CFG // 使能中断并应用高通数字滤波器；锁存
// 模式禁用
3. 将 00h 写入 WAKE_UP_DUR // 无持续
4. 将 02h 写入 WAKE_UP_THS // 设置唤醒阈值
5. 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 ($= 2 * FS_XL / 2^6$)。由于唤醒功能利用斜率/高通数字滤波器实现，因此有必要考虑此功能使能后滤波器的稳定时间。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 3.7.1 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。下面三个备选解决方案（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；
- c. 初始时设置一个较高的 ODR（833 Hz），这样会在较短时间内产生最初的 2 个采样，然后按下述步骤设置所需 ODR（例如 12.5 Hz）并驱动中断信号至引脚：

- | | |
|-------------------------|---------------------------------------|
| 1. 将 00h 写入 WAKE_UP_DUR | // 无持续 |
| 2. 将 02h 写入 WAKE_UP_THS | // 设置唤醒阈值 |
| 3. 将 80h 写入 TAP_CFG | // 使能中断并应用斜率滤波器；锁存模式禁用 |
| 4. 将 70h 写入 CTRL1_XL | // 启动加速度计 |
| | // ODR_XL = 833 Hz, FS_XL = $\pm 2 g$ |
| 5. 等待 4 ms | // 插入（减少）等待时间 |
| 6. 将 10h 写入 CTRL1_XL | // ODR_XL = 12.5 Hz |
| 7. 将 20h 写入 MD1_CFG | // 唤醒中断驱动至 INT1 引脚 |

5.4 6D/4D 定向检测

LSM6DSL 器件具有检测空间中器件方向的能力，能够容易地实现移动设备的节能程序和自动图像旋转。

5.4.1 6D 定向检测

可以检测器件在空间中的六个方向；当器件从一个方向转向另一个方向时，中断信号产生。只要保持其位置，中断就不会重新产生。

对于两个连续采样，当只有一个轴超出所选阈值，其他两轴上测得的加速度值低于阈值时，会产生 6D 中断：

D6D_SRC 寄存器的 ZH, ZL, YH, YL, XH, XL 位表示出哪个轴触发了 6D 事件。

更具体地说：

表 26. D6D_SRC 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
DEN_DRDY	D6D_IA	ZH	ZL	YH	YL	XH	XL

- 当器件从一个方向转向另一个方向时，D6D_IA 被置为高电平。
- 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绝对值大于阈值时，ZH (YH, XH) 被置为高电平。
- 当垂直于(Y,X)轴的面几乎是平面，Z(Y,X) 轴上测得的加速度为负且绝对值大于阈值时，ZL (YL, XL)被置为高电平。

TAP_THS_6D 寄存器的 SIXD_THS[1:0]位用来选择阈值，该阈值用于检测器件方向变化。表 27. 4D/6D 功能阈值中给出的阈值对于每种加速度计的量程均有效。

表 27. 4D/6D 功能阈值

SIXD_THS[1:0]	阈值[degrees]
00	80
01	70
10	60
11	50

通过将 CTRL8_XL 寄存器的 LOW_PASS_ON_6D 位置为 1，此低通滤波器 LPF2 还可用于 6D 功能。

通过将 TAP_CFG 寄存器中的 INTERRUPTS_ENABLE 位置为 1，可使能该中断信号，将 MD1_CFG 寄存器的 INT1_6D 位或 MD2_CFG 寄存器的 INT2_6D 位置为 1，可将该中断信号驱动至两个中断引脚上；还可通过读取 D6D_SRC 寄存器的 D6D_IA 位对其进行检查。

如果锁存模式禁用（TAP_CFG 的 LIR 位置为 0），则中断信号仅激活 1/ODR_XL[s]，然后自动取消（ODR_XL 为加速度计输出数据率）。如果锁存模式使能，并且 6D 中断信号被驱动至中断引脚，那么当方向发生了改变且中断引脚被产生时，对 D6D_SRC 寄存器的读取会清除中断信号，器件将识别另一个不同的方向。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。

参考图 14. 6D 识别方向中所示的六种可能情形，表 28. 6D 定位下的 D6D_SRC 寄存器中显示了每个位置对应的 D6D_SRC 寄存器内容所示。

图 14. 6D 识别方向

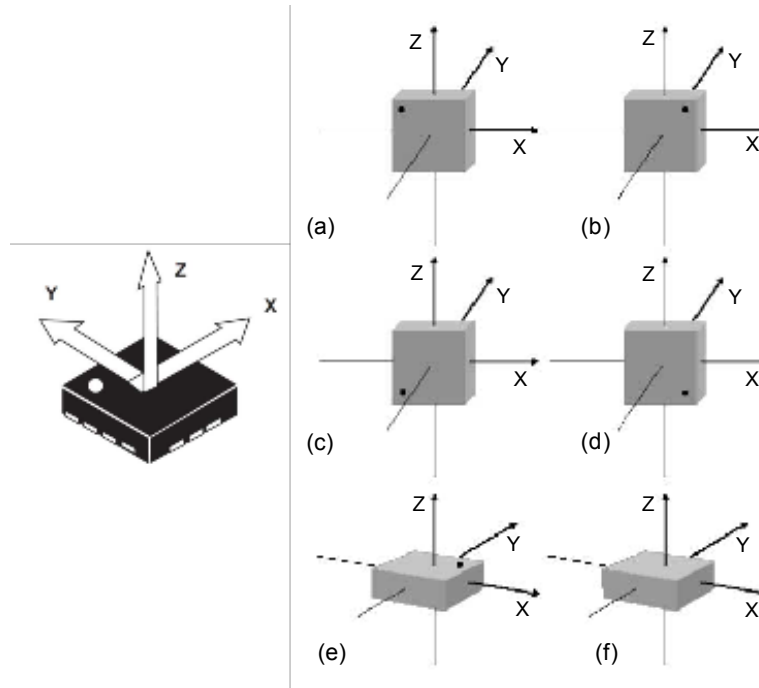


表 28. 6D 定位下的 D6D_SRC 寄存器

用例	D6D_IA	ZH	ZH	YH	YL	XH	XL
(a)	1	0	0	1	0	0	0
(b)	1	0	0	0	0	0	1
(c)	1	0	0	0	0	1	0
(d)	1	0	0	0	1	0	0
(e)	1	1	0	0	0	0	0
(f)	1	0	1	0	0	0	0

6D 方向检测的基本软件例程如下。

1. 将 60h 写入 CTRL1_XL // 启动加速度计
// ODR_XL = 416 Hz, FS_XL = $\pm 2g$
2. 将 80h 写入 TAP_CFG // 使能中断；锁存模式禁用
3. 将 40h 写入 TAP_THS_6D // 设置 6D 阈值 (SIXD_THS[1:0] = 10b = 60 degrees)
4. 将 01h 写入 CTRL8_XL // 将 LPF2 滤波器用于 6D 功能
5. 将 04h 写入 MD1_CFG // 6D 中断驱动至 INT1 引脚

5.4.2 4D 方向检测

4D 方向功能是 6D 功能的子集，它被专门定义来进行移动设备中的纵向和横向计算。它可通过将 TAP_THS_6D 寄存器的 D4D_EN 位置为 1 来使能。这种配置下，Z 轴位置检测被禁用，因此位置识别减少为表 28. 6D 定位下的 D6D_SRC 寄存器的(a)、(b)、(c)和(d)的情形。

5.5 单击和双击识别

LSM6DSL 具有单击和双击识别功能，能够在极少加载软件的情况下帮助创建人机界面。器件可配置为沿任意方向敲击时在专用引脚上输出中断信号。

如果传感器施加单个输入激励，那么它会在中断引脚 INT1 和/或 INT2 上产生中断请求。更先进的功能可在识别到两次输入激励（两个事件的间隔时间可通过程序设定）时生成中断请求，从而可实现类似鼠标按键的功能。

LSM6DSL 器件中，单击和双击识别功能利用两个连续加速度采样之间的斜率来检测点击事件；斜率数据利用以下公式计算：

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

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斜率数据幅度和时序进行编程。

单击和双击识别独立于所选输出数据率而工作。对于这些功能，建议的加速度计 ODR 为 416 Hz 和 833 Hz。

要启用单击和双击识别功能，必须将 TAP_CFG 寄存器中的 INTERRUPTS_ENABLE 位置为 1。

5.5.1

单击

如果器件配置为单击事件检测，那么当所选通道的斜率数据超出了所编程阈值时，会产生一个中断，并在 Shock 时间窗口内返回低电平。

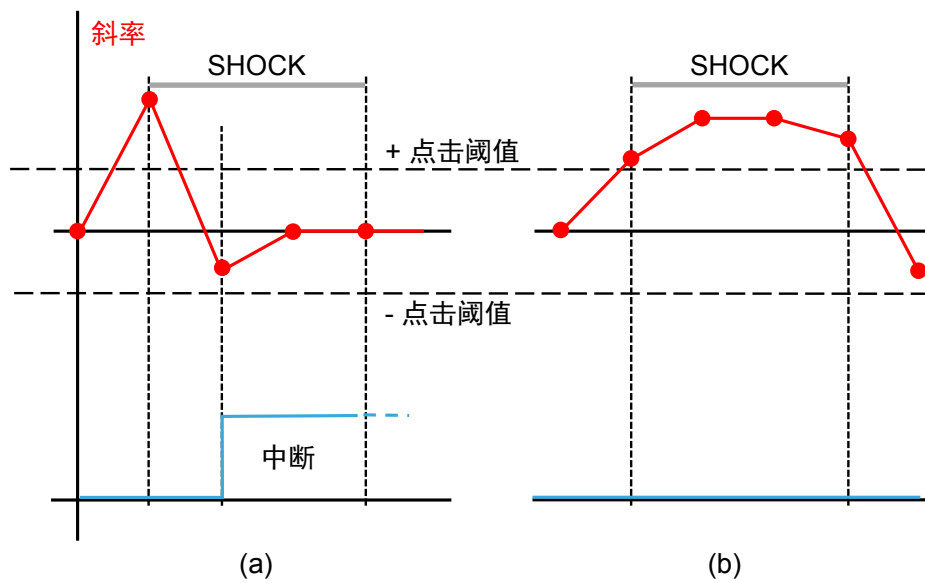
在单击情况下，如果 TAP_CFG 寄存器的 LIR 位被置为 0，则中断在 Quiet 窗口持续时间内保持激活。

为了在单击中断信号上使能锁存功能，LIR 位和 MD1_CFG (MD2_CFG) 的 INT1_DOUBLE_TAP (或 INT2_DOUBLE_TAP) 位必须置为 1：中断保持激活，直至 TAP_SRC 寄存器被读取。

要实现仅使能单击识别，则 WAKE_UP_THS 的 SINGLE_DOUBLE_TAP 位必须置为 0。

图 15. 单击事件识别的情况(a)中识别出了单击事件，而在情况(b)中，由于在经过 Shock 时间窗口之后斜率数据低于阈值，因此未识别出点击。

图 15. 单击事件识别



5.5.2 双击

如果器件配置为双击事件检测，那么在第一次点击后、识别出第二次点击时，生成中断。只有当事件满足 **Shock**、**Latency** 和 **Duration** 时间窗口所定义的规则时，才进行第二次点击识别。

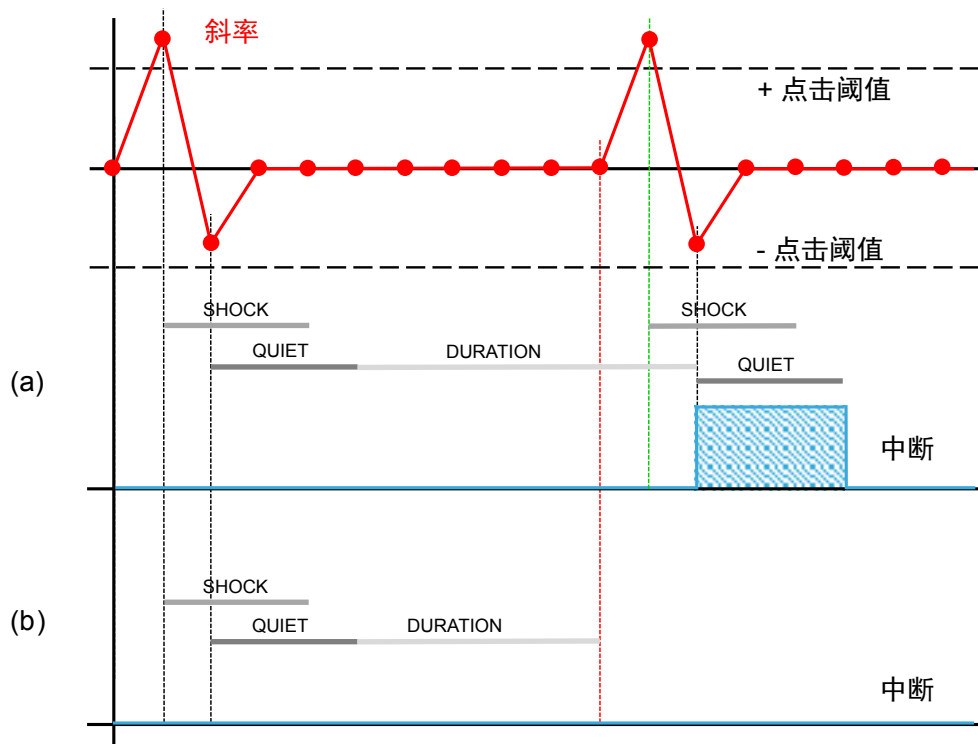
特别是，识别出第一次敲击后，第二次敲击检测过程会延迟 **Quiet** 时间所定义的时间间隔。这意味着，识别出第一次敲击后，只有在 **Quiet** 窗口之后、**Duration** 窗口结束前，斜率数据超过阈值时，才开始第二次敲击识别过程。
图 16. 双击事件识别 (**LIR 位 = 0**) 的情况(a)中，正确识别出了双击事件，而在情况(b)中，由于在经过窗口间隔之后斜率数据超出了阈值，因此未产生中断。

一旦第二次点击检测过程开始，则会按照与第一次相同的规则来识别第二次点击：在 **Shock** 窗口结束之前，斜率数据必须返回到低于阈值之下。

要避免因输入信号伪突变而产生不期望的点击，适当定义 **Quiet** 窗口是非常重要的。

在双击情况下，如果 **TAP_CFG** 寄存器的 **LIR** 位被置为 0，则中断在 **Quiet** 窗口持续时间内保持激活。如果 **LIR** 位被置为 1，则中断保持激活直至 **TAP_SRC** 寄存器被读取。

图 16. 双击事件识别 (**LIR 位 = 0**)



5.5.3 单击和双击识别配置

可对 LSM6DSL 器件进行配置，使其在任一方向发生点击（一次或两次）时均输出中断信号：TAP_CFG 寄存器的 TAP_X_EN、TAP_Y_EN 和 TAP_Z_EN 位必须置为 1，分别使能 X、Y、Z 方向上的点击识别。此外，TAP_CFG 寄存器的 INTERRUPTS_ENABLE 位必须设置为 1。

点击识别功能的可配置参数为点击阈值和 Shock、Quiet 及 Duration 时间窗。

TAP_THS_6D 寄存器的 TAP_THS[4:0]位用来选择用于检测点击事件的无符号阈值。这 5 bit 的 1 LSB 值取决于所选加速度计满量程：1 LSB = (FS_XL)/(2⁵)。无符号阈值可应用于正负斜率数据上。

注：通过 TAP_THS_6D 寄存器的 TAP_THS [4:0]位设置的点击阈值（以 mg 为单位）必须高于通过 WAKE_UP_THS 寄存器的 WK_THS [5:0]位设置的唤醒阈值（以 mg 为单位）。

Shock 时间窗口定义了超阈值事件的最大持续时间：在 Shock 窗口结束前，加速度必须返回到低于阈值之下，否则不能检测到该点击事件。INT_DUR2 寄存器的 SHOCK[1:0]位用来设置 Shock 时间窗口值：这几个位的默认值为 00b，对应于 4/ODR_XL 的时间，这里 ODR_XL 为加速度计输出数据率。如果 SHOCK[1:0]位被置为其他不同的值，那么 1 LSB 对应于 8、ODR_XL 的时间。

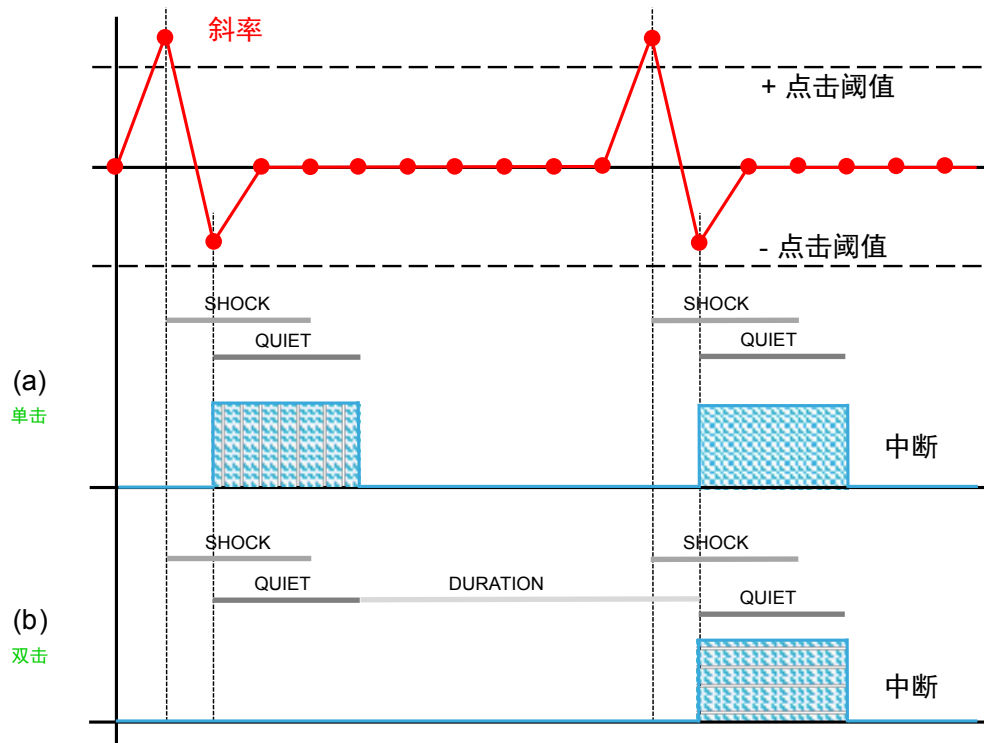
双击情况下，Quiet 时间窗口定义了第一次点击识别后的时间，期间不能发生超阈值。当锁存模式禁用（TAP_CFG 的 LIR 位置为 0）时，Quiet 时间还定义了中断脉冲的长度（单击和双击情况下均如此）。INT_DUR2 寄存器的 QUIET[1:0]位用来设置 Quiet 时间窗口值：这几个位的默认值为 00b，对应于 2/ODR_XL 的时间，这里 ODR_XL 为加速度计输出数据率。如果 QUIET[1:0]位被置为其他不同的值，那么 1 LSB 对应于 4/ODR_XL 的时间。

双击情况下，Duration 时间窗口定义了连续两次检测到点击之间的最大时间。Duration 时间周期在第一次点击的 Quiet 时间结束后开始。INT_DUR2 寄存器的 DUR[3:0]位用来设置 Duration 时间窗口值：这几个位的默认值为 0000b，对应于 16/ODR_XL 的时间，这里 ODR_XL 为加速度计输出数据率。如果 DUR[3:0]位被置为其他不同的值，那么 1 LSB 对应于 32/ODR_XL 的时间。

图 17. 单击和双击识别（LIR 位 = 0）显示了单击事件（a）和双击事件（b）。这些中断信号可被驱动至两个中断引脚，单击情况下通过将 MD1_CFG 寄存器的 INT1_SINGLE_TAP 位或 MD2_CFG 寄存器的 INT2_SINGLE_TAP 位置为 1 来实现，双击情况下通过将 MD1_CFG 寄存器的 INT1_DOUBLE_TAP 位或 MD2_CFG 寄存器的 INT2_DOUBLE_TAP 位置为 1 来实现。

如果加速度计处于下电状态，则不产生单击/双击中断（更多详细信息见第 5.6 节 活动/不活动识别）。

图 17. 单击和双击识别（LIR 位 = 0）



还可通过读取 TAP_SRC (1Ch) 寄存器来检查点击中断信号，如表 29. TAP_SRC 寄存器所述。

表 29. TAP_SRC 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
0	TAP_IA	SINGLE_TAP	DOUBLE_TAP	TAP_SIGN	X_TAP	Y_TAP	Z_TAP

- 当检测到单击或双击事件时，TAP_IA 置为高电平。
- 当检测到单击时，SINGLE_TAP 置为高电平。
- 当检测到双击时，DOUBLE_TAP 置为高电平。
- TAP_SIGN 指示检测到点击事件时的加速度符号。符号为正时它为低电平，符号为负时它为高电平。
- 当在 X (Y、Z) 轴上检测到点击事件时，X_TAP (Y_TAP、Z_TAP) 置为高电平。

单击和双击识别独立工作。将 WAKE_UP_THS 寄存器的 SINGLE_DOUBLE_TAP 位置为 0，则仅使能单击识别；双击识别被禁用，不能被检测到。当 SINGLE_DOUBLE_TAP 置为 1 时，单击和双击识别均使能。

如果锁存模式使能，且中断信号被驱动至中断引脚，则指定到 SINGLE_DOUBLE_TAP 的值还会影响中断信号的特性：当它被置为 0 时，单击中断信号可应用于锁存模式；当它被置为 1 时，只有双击中断信号可应用于锁存模式。锁存的中断信号保持激活，直至 TAP_SRC 寄存器被读取。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。

5.5.4

单击示例

下面给出了单击检测的基本软件程序。

1. 将 60h 写入 CTRL1_XL // 启动加速度计
// ODR_XL = 416 Hz, FS_XL = $\pm 2 g$
2. 将 8Eh 写入 TAP_CFG // 使能 X、Y、Z 轴上的中断和点击检测
3. 将 89h 写入 TAP_THS_6D // 设置点击阈值
4. 将 06h 写入 INT_DUR2 // 设置 Quiet 和 Shock 时间窗口
5. 将 00h 写入 WAKE_UP_THS // 只使能单击 (SINGLE_DOUBLE_TAP = 0)
6. 将 40h 写入 MD1_CFG // 单击中断驱动至 INT1 引脚

本例中，TAP_THS_6D 寄存器的 TAP_THS 字段被置为 01001b，因此点击阈值为 562.5 mg ($= 9 * FS_{XL} / 2^5$)。

INT_DUR2 寄存器的 SHOCK 字段被置为 10b：当斜率数据超出所编程阈值时，产生中断，并在 38.5 ms ($= 2 * 8 / ODR_{XL}$) 内返回到低于该阈值，这段时间对应于 Shock 时间窗口。

INT_DUR2 寄存器的 QUIET 字段被置为 01b：由于锁存模式禁用，中断会保持高电平并持续 Quiet 窗口的时间，因此为 9.6 ms ($= 1 * 4 / ODR_{XL}$)。

5.5.5

双击示例

下面给出了双击检测的基本软件程序。

1. 将 60h 写入 CTRL1_XL // 启动加速度计
// ODR_XL = 416 Hz, FS_XL = $\pm 2 g$
2. 将 8Eh 写入 TAP_CFG // 使能 X、Y、Z 轴上的点击检测
3. 将 8Ch 写入 TAP_THS_6D // 设置点击阈值
4. 将 7Fh 写入 INT_DUR2 // 设置 Duration、Quiet 和 Shock 时间窗口

5. 将 80h 写入 WAKE_UP_THS // 使能单击 & 双击 (SINGLE_DOUBLE_TAP = 1)
6. 将 08h 写入 MD1_CFG // 双击中断驱动至 INT1 引脚

本例中, TAP_THS_6D 寄存器的 TAP_THS 字段被置为 01100b, 因此点击阈值为 750 mg ($= 12 * FS_XL / 2^5$)。要实现中断生成, 在第一次和第二次点击过程中, Shock 结束前, 斜率数据必须返回到低于阈值。INT_DUR2 寄存器的 SHOCK 字段被置为 11b, 因此 Shock 时间为 57.7 ms ($= 3 * 8 / ODR_XL$)。

对于中断生成, 第一次点击识别后, 在 Quiet 时间窗口内斜率数据不能超阈值。而且, 由于锁存模式禁用, 因此中断会保持高电平, 并持续 Quiet 窗口的时间。INT_DUR2 寄存器的 QUIET 字段被置为 11b, 因此 Quiet 时间为 28.8 ms ($= 3 * 4 / ODR_XL$)。

要使连续两次检测到的点击之间时间达到最大, INT_DUR2 寄存器的 DUR 字段被置为 0111b, 因此 Duration 时间为 538.5 ms ($= 7 * 32 / ODR_XL$)。

5.6 活动/不活动识别

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, LSM6DSL 器件能够自动将加速度计采样率降低至 12.5 Hz, 当检测到唤醒中断事件时增加加速度计 ODR 和带宽。在 LSM6DSL 中这个功能可以扩展到陀螺仪, 有三种可能的选择:

- 陀螺仪配置不变;
- 陀螺仪进入睡眠模式;
- 陀螺仪进入下电模式。

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。

使用活动/不活动功能所允许的最大加速度计 ODR (可通过 CTRL1_XL 寄存器的 ODR_XL [3:0] 位来配置) 为 3.3 kHz。

通过将 INTERRUPTS_ENABLE 位置为 1, 并配置 TAP_CFG 寄存器的 INACT_EN 位, 可使能活动/不活动识别功能。表 30. 不活动事件配置中总结了不活动事件的可能配置。

表 30. 不活动事件配置

INACT_EN[1:0]	加速度计	陀螺仪
00	不活动事件禁用	不活动事件禁用
01	XL ODR = 12.5 Hz (低功耗模式)	陀螺仪配置不变
10	XL ODR = 12.5 Hz (低功耗模式)	Gyro 处于睡眠模式
11	XL ODR = 12.5 Hz (低功耗模式)	陀螺仪处于下电模式

LSM6DSL 器件中, 可以利用斜率滤波器 (更多详细信息见第 3.7.1 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 TAP_CFG 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 18. 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 6 bit 的 1 LSB 值取决于所选加速度计测量程: $1 \text{ LSB} = (FS_XL) / (2^6)$ 。该阈值可适用于正负滤波数据。

当一定数量的连续 X、Y、Z 滤波数据小于所配置阈值时, 忽略 CTRL1_XL 寄存器的 ODR_XL [3:0] 位 (不活动), 加速度计被内部地设置为 12.5 Hz, 尽管 CTRL1_XL 内容保持不变。陀螺仪的特性随着 TAP_CFG 寄存器 INACT_EN 位的配置而变化。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 $512 / ODR_XL$ 的时间, 这里 ODR_XL 为加速度计输出数据率。

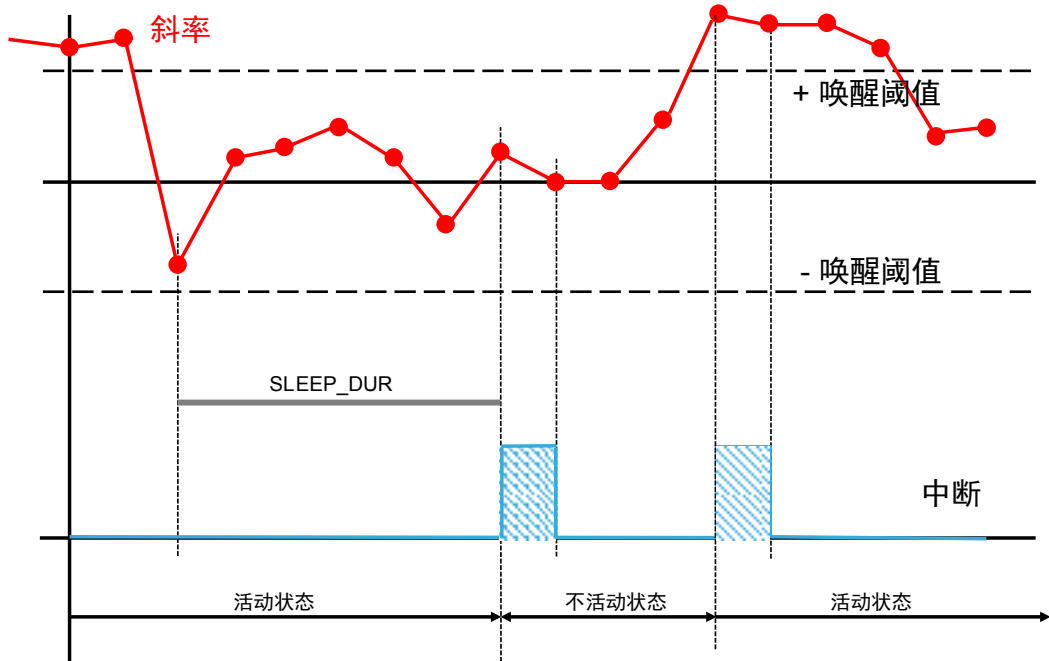
当检测到不活动状态时, 中断被置为高电平并持续 $1 / ODR_XL[s]$ 的时间周期, 然后自动取消。

当一个轴上有一个采样的 X、Y、Z 滤波数据大于阈值时, 会立即恢复 CTRL1_XL 寄存器设置 (活动), 并且陀螺仪恢复到先前的状态。

当检测到活动状态时，中断被置为高电平并持续 $1/\text{ODR_XL}[\text{s}]$ 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_INACT_STATE 位或 MD2_CFG 寄存器的 INT2_INACT_STATE 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 寄存器的 SLEEP_STATE_IA 位来对其进行检查。

图 18. 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 50h 写入 CTRL1_XL // 启动加速度计
// ODR_XL = 208 Hz, FS_XL = $\pm 2\text{ g}$
2. 将 40h 写入 CTRL2_G // 开启陀螺仪
// ODR_G = 104 Hz, FS_G = $\pm 250\text{ dps}$
3. 将 02h 写入 WAKE_UP_DUR // 设置不活动检测的持续时间
4. 将 02h 写入 WAKE_UP_THS // 设置活动/不活动阈值
5. 将 E0h 写入 TAP_CFG // 使能中断
// 不活动配置：加速度计配置为 12.5 LP，陀螺仪配置为下电
// 使能斜率滤波器
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 $62.5\text{ mg} (= 2 * \text{FS_XL} / 2^6)$ 。

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 $4.92\text{ s} (= 2 * 512 / \text{ODR_XL})$ 。经过这段时间之后，加速度计 ODR 内部设置为 12.5 Hz，陀螺仪内部设置为下电模式。



如果（至少）一个轴的斜率数据大于阈值，则会检测到活动状态并立即恢复 CTRL1_XL 寄存器设置和开启陀螺仪。

5.7 启动状态

器件上电后，LSM6DSL 执行一段 15 ms 的启动程序来加载配置参数。启动完成后，加速度计和陀螺仪均自动配置为下电模式。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。SW_RESET 过程会花费 50 μ 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. 将陀螺仪设置为下电模式；
2. 将加速度计设置为高性能模式；
3. 将 CTRL3_C 寄存器的 BOOT 位置为 1；
4. 等待 15 ms。

复位流程如下：

1. 将陀螺仪设置为下电模式；
2. 将加速度计设置为高性能模式；
3. 将 CTRL3_C 寄存器的 SW_RESET 位置为 1；
4. 等待 50 μ s（或等待至 CTRL3_C 寄存器的 SW_RESET 位返回 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。

6 嵌入功能

LSM6DSL 器件在硬件中实现许多嵌入功能；功耗可忽略且高性能的专用 IP 模块仅用加速度计即可实现以下功能：

- 计步功能（步伐侦测和步数计算）；
- 大幅运动检测；
- 相对倾斜；
- 绝对手腕倾斜；
- 时间戳。

所有这些功能工作于 26 Hz，因此加速度计 ODR 必须设置为 26 Hz 或更高的值。

6.1 计步功能：步伐侦测和步数计算

LSM6DSL 器件的专用 IP 模块来专门实现计步功能：步伐侦测和步数计算。

计步功能工作于 26 Hz，因此加速度计 ODR 必须设置为 26 Hz 或更高的值。

要使能计步功能，需要将 CTRL10_C 寄存器的 FUNC_EN 位和 PEDO_EN 位置为 1。

使能计步功能后，计步器显示算法检测到的步数。步数由 STEP_COUNTER_H 和 STEP_COUNTER_L 寄存器输出，表示为一个 16 位无符号数字。

当加速度计被配置为下电或计步器禁用时，步数不会复位至 0；可通过将 CTRL10_C 寄存器的 PEDO_RST_STEP 位置为 1 来将其复位至 0。计数器复位后，PEDO_RST_STEP 位不会自动重置回 0。

步伐侦测功能会在每次识别出一步时生成一个中断。在随机行走事件中，须检测到连续 7 步（去抖动步数）才能生成第一个中断，以避免出现错误步数检测（去抖动功能）。

去抖动步数可通过 PEDO_DEB_REG 寄存器的 DEB_STEP 字段进行修改：基本上，它对应于在第一次计步器增加前，要检测到的最小步数。该字段的 1 LSB 对应于 1 步，默认值为 6 步。

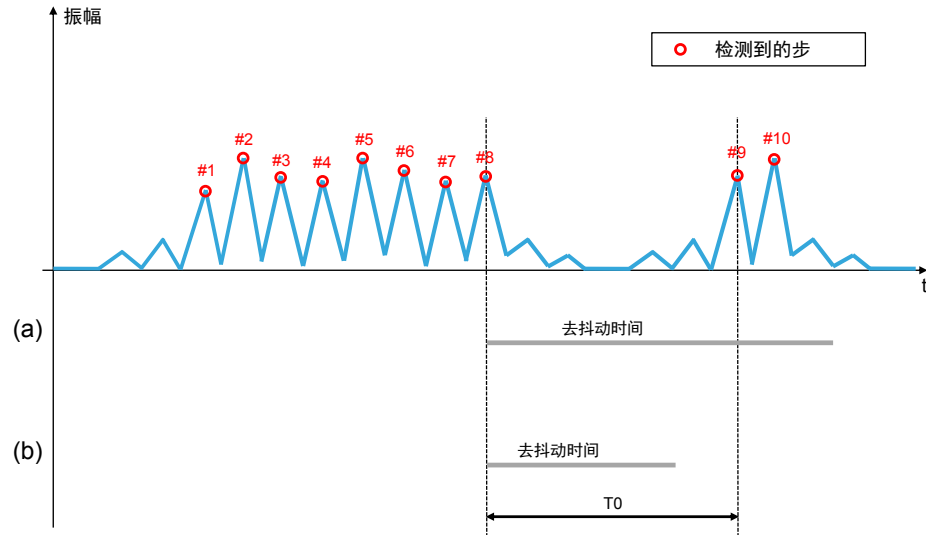
器件不活动约 1 秒后，会重新启动去抖动功能。这段时间（去抖动时间）可通过 PEDO_DEB_REG 寄存器的 DEB_TIME 字段进行修改。1LSB 对应于 80 ms，默认值为 13 (13 * 80 ms = 1040 ms)。此值必须大于 0。

图 19. 计步器去抖动中示例说明了改变去抖动时间，计步器特性会如何变化。本例中，计步算法检测出彼此接近的 7 步，并在一定时间后检测到单独的另外两步；假设 PEDO_DEB_REG 寄存器的 DEB_STEP 字段值设置为 6 LSB (= 6 个去抖动步，默认值)，且 STEP_COUNTER_H/L 寄存器中初始计步器值为零（先前未检测到步数）：

a.情形(a)中，第七步之后步数开始增加，第一次检测到八步后，STEP_COUNTER_H/L 寄存器的值将为 8。由于 PEDO_DEB_REG 寄存器的 DEB_TIME 字段中设置的去抖动时间大于步#8 和步#9 之间的时长，因此步#9 和步#10 还是会引引起计步器增加：STEP_COUNTER_H/L 寄存器中最后的步数值将为 10。

b.情形(b)中，也是第七步后步数开始增加，第一次检测到八步后，STEP_COUNTER_H/L 寄存器的值将为 8，但是由于 PEDO_DEB_REG 寄存器 DEB_TIME 字段中设置的去抖动时间小于步#8 和步#9 之间的时长，因此步#9 和步#10 不会引起计步器增加：STEP_COUNTER_H/L 寄存器中最后的步数值将为 8。而且，如果步#10 和下一步之间经过的这段时间大于去抖动时间，那么检测到的步#9 和步#10 将明确会丢弃且不再考虑。

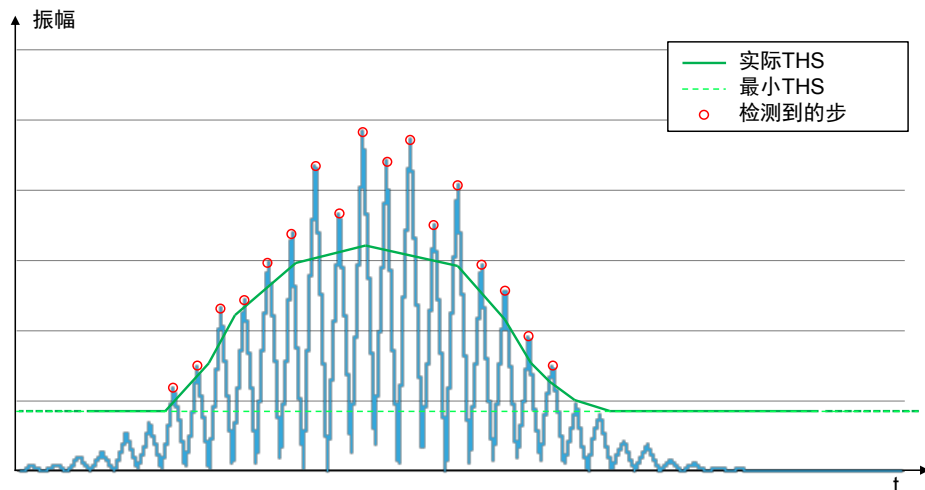
图 19. 计步器去抖动



默认情况下，计步器工作于 $\pm 2g$ 满量程，与所配置的器件满量程无关，但是它也可被配置为工作于 $\pm 4g$ 满量程，这样可以帮助避免加速度饱和（例如，快速行走时）。要为计步器设置 $\pm 4g$ 满量程，CONFIG_PEDO_THS_MIN寄存器的PEDO_FS位必须置为1，并且CTRL1_XL寄存器中配置的加速度计满量程必须 $\geq \pm 4g$ 。

还可以设置“最小阈值”，它是未检测到步数时步数识别阈值渐进趋向的值，并且不能下降到比它更小的值（见图20. 计步器最小阈值）。可在CONFIG_PEDO_THS_MIN寄存器的ths_min字段实现此配置。这5位的1 LSB值取决于所选计步器满量程：如果PEDO_FS位为0，1 LSB = 32 mg；如果PEDO_FS位为1，1 LSB = 62.5 mg。

图 20. 计步器最小阈值



通过将INT1_CTRL寄存器的INT1_STEP_DETECTOR位置为1，可将步数检测器中断信号驱动至INT1中断引脚上；还可通过读取FUNC_SRC1寄存器的STEP_DETECTED位对其进行检查。



如果在一定时间段内检测到了至少一步，即可生成中断，而不是每次识别出一步才生成中断。这段时间可通过在 STEP_COUNT_DELTA 寄存器中设置一个高于 00h 的值来定义。使用此功能时，需要将 CTRL10_C 寄存器的 TIMER_EN 位置为 1（来使能定时器），并将 WAKE_UP_DUR 寄存器的 TIMER_HR 位置为 0；这样，STEP_COUNT_DELTA 寄存器值的 1 LSB 对应 1.6384 秒。通过将 INT2_CTRL 寄存器的 INT2_STEP_DELTA 位置为 1，可将该中断信号驱动至 INT2 中断引脚上；还可通过读取 FUNC_SRC1 寄存器的 STEP_COUNT_DELTA_IA 位对其进行检查。

可通过 INT2_CTRL 将寄存器的 INT2_STEP_COUNT_OV 位置为 1，将计步器溢出信号驱动至 INT2 中断引脚：这种情况下，当步数达到 2^{16} 的值时，INT2 引脚上会产生一个中断信号，步数自动复位至零，而无需通过将 PEDO_RST_STEP 位置为 1 来将其复位。

如果锁存模式禁用（TAP_CFG 的 LIR 位置为 0），计步功能所产生的中断信号为脉冲的：中断引脚上观测到的脉冲持续时间约为 150 μ s；FUNC_SRC1 寄存器的 STEP_COUNT_DELTA_IA、STEP_DETECTED 和 STEP_OVERFLOW 位上观测到的脉冲持续时间为 1/26 Hz。

如果锁存模式使能（TAP_CFG 的 LIR 位置为 1），且中断信号被驱动至中断引脚，则当进行一步时，对 FUNC_SRC1 寄存器的读取会将两个引脚上的请求以及 FUNC_SRC1 寄存器的 STEP_COUNT_DELTA_IA、STEP_DETECTED 和 STEP_OVERFLOW 位清除，器件准备好识别下一步。如果锁存模式使能但是中断信号未驱动至中断引脚，那么 FUNC_SRC1 寄存器位上观测到的中断信号是脉冲，且持续时间固定为 1/26 Hz。

计步器时间戳信息可在 STEP_TIMESTAMP_H 和 STEP_TIMESTAMP_L 寄存器中获取：当检测到一步时，TIMESTAMP_REG2 寄存器的值复制到 STEP_TIMESTAMP_H 中，TIMESTAMP_REG1 寄存器的值复制到 STEP_TIMESTAMP_L 中，提供该步的时间戳信息。关于 LSM6DSL 时间戳计数器和 TIMESTAMP_REG2/TIMESTAMP_REG1 的更多详细信息，见第 6.5 节 时间戳。

计步器时间戳的分辨率取决于 WAKE_UP_DUR 寄存器 TIMER_HR 位的值：当该位被置为 0 时，步数的 1 LSB 对应的时间为 1638.4 ms；当该位被置为 1 时，步数的 1 LSB 对应的时间为 6.4 ms。

计步器数据可连同时间戳数据一起作为第四数据集存储在 FIFO 中（更多详细信息见第 8.8 节 FIFO 中的计步器和时间戳数据）。

以下是一个基本软件程序，显示如何使能计步功能：

- | | |
|---------------------------------|--------------------------------------|
| 1. 将 80h 写入 FUNC_CFG_ACCESS | // 使能对嵌入功能寄存器的访问（A 区） |
| 2. 将 8Eh 写入 CONFIG_PEDO_THS_MIN | // PEDO_FS = ± 4 g 并配置计步器最小阈值 |
| 3. 将 00h 写入 FUNC_CFG_ACCESS | // 禁用对嵌入功能寄存器的访问（A 区） |
| 4. 将 28h 写入 CTRL1_XL | // 启动加速度计 |
| | // ODR_XL = 26 Hz, FS_XL = ± 4 g |
| 5. 将 14h 写入 CTRL10_C | // 启用嵌入功能和计步器算法 |
| 6. 将 80h 写入 INT1_CTRL | // 步数探测器中断驱动至 INT1 引脚 |

当识别出一步时，产生该中断信号，可通过读取 STEP_COUNTER_H / STEP_COUNTER_L 寄存器来获取步数。

6.2 大幅运动检测

检测到“大幅运动”（该运动可能是由于用户位置发生变化）时，大幅运动检测功能会产生一个中断：LSM6DSL 器件中，仅利用加速度计，已经在硬件中实现了此功能。

大幅运动检测功能可用于基于位置的应用，用来接收指示用户何时改变位置的通知。

大幅运动检测功能工作于 26 Hz，因此加速度计 ODR 必须设置为 26 Hz 或更高的值。

要使能大幅运动检测检测，需要将 CTRL10_C 寄存器的 FUNC_EN 位和 SIGN_MOTION_EN 位置为 1。

通过将 INT1_CTRL 寄存器的 INT1_SIGN_MOT 位置为 1，可将大幅运动中中断信号驱动至 INT1 中断引脚上；还可通过读取 FUNC_SRC1 寄存器的 SIGN_MOTION_IA 位对其进行检查。

如果锁存模式禁用（TAP_CFG 的 LIR 位置为 0），大幅运动检测功能所产生的中断信号为脉冲的：中断引脚上观测到的脉冲持续时间约为 150 μ s；FUNC_SRC1 寄存器的 SIGN_MOTION_IA 位上观测到的脉冲持续时间为 1/26 Hz。

如果锁存模式使能（TAP_CFG 的 LIR 位置为 1），且中断信号被驱动至中断引脚，那么当检测到倾斜时，对 FUNC_SRC1 寄存器的读取会将两个引脚上的请求以及 FUNC_SRC1 寄存器的 TILT_IA 位清零，器件准备好识别下一次倾斜事件。如果锁存模式使能但是中断信号未驱动至中断引脚，那么 FUNC_SRC1 寄存器 SIGN_MOTION_IA 位上观测到的中断信号是脉冲，且持续时间固定为 1/26 Hz。

用来配置大幅运动阈值参数的嵌入功能寄存器（可通过将 `FUNC_CFG_ACCESS` 的 `FUNC_CFG_EN` 位置 1 来访问）为 `SM_THS` 寄存器。此寄存器的 `SM_THS_[7:0]` 位定义了阈值：它对应于产生大幅运动中中断前，位置改变时用户所行走的步数。它表示为 8 位无符号值：该字段默认值等于 6 (= 00000110b)。

当计步器的去抖动功能激活时（详细信息见第 6.1 节 **计步功能：步伐侦测和步数计算**），仅当其值（对应 `SM_THS` 寄存器 `SM_THS_[7:0]` 位的值）大于等于计步器去抖动阈值（对应 `PEDO_DEB_REG` 寄存器 `DEB_STEP[2:0]` 位的值）时，大幅运动阈值有效。

基本上，对于大幅运动阈值，可以有三种不同的情形：

- 如果计步器去抖动功能未激活，则大幅运动阈值由 `SM_THS_[7:0]` 位定义；
- 如果计步器去抖动功能激活，且大幅运动阈值大于等于计步器去抖动阈值，则有效的大幅运动阈值由 `SM_THS_[7:0]` 位定义；
- 如果计步器去抖动功能激活，且大幅运动阈值小于计步器去抖动阈值，则有效的大幅运动阈值由 `DEB_STEP[2:0]` 位定义。

注意：情形 c) 中，如果所需大幅运动阈值低于默认值，则 `PEDO_DEB_REG` 寄存器 `DEB_STEP[2:0]` 位的值必须随之降低。**注意，过度减少计步器去抖动阈值可能导致计步器步数检测报错！**

以下是一个基本软件程序，显示如何使能大幅运动检测功能：

- | | |
|--|--|
| 1. 将 80h 写入 <code>FUNC_CFG_ACCESS</code> | // 使能对嵌入功能寄存器的访问（A 区） |
| 2. 将 08h 写入 <code>SM_THS</code> | // 设置大幅运动阈值 |
| 3. 将 00h 写入 <code>FUNC_CFG_ACCESS</code> | // 禁用对嵌入功能寄存器的访问（A 区） |
| 4. 将 20h 写入 <code>CTRL1_XL</code> | // 启动加速度计 |
| | // <code>ODR_XL = 26 Hz, FS_XL = ±2 g</code> |
| 5. 将 05h 写入 <code>CTRL10_C</code> | // 使能嵌入功能 |
| | // 使能大幅运动检测 |
| 6. 将 40h 写入 <code>INT1_CTRL</code> | // 大幅运动中中断驱动至 INT1 引脚 |

本例中 `SM_THS` 寄存器的 `SM_THS_[7:0]` 位被置为 00001000b，因此大幅运动阈值等于 8。

6.3 相对倾斜

倾斜度检测功能支持检测何时发生活动改变（例如，当电话在前口袋中用户从坐到站或从站到坐时）：在 `LSM6DSL` 器件中，仅使用加速度计即可在硬件中实现它。

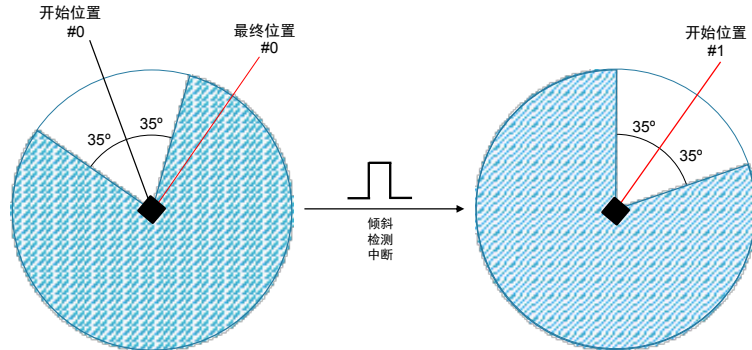
要使能倾斜探测器，需要将 `CTRL10_C` 寄存器的 `FUNC_EN` 位和 `TILT_EN` 位置为 1。

如果器件配置为用于倾斜度检测，那么当器件距起始位置的倾斜角度大于 35 度时，会产生一个中断。起始位置定义为倾斜检测使能时器件的位置，或上一次倾斜中断产生时的器件位置。

使能此功能后，要产生第一次倾斜中断，器件距起始位置的倾斜角度应大于 35 度并持续 2 秒的时间。产生第一次倾斜中断后，当器件距离上次中断检测时的器件位置的倾斜角度大于 35 度时（不需要等待 2 秒），倾斜中断信号即被置为高电平。

在图 21. 倾斜度检测中所示的示例中，当器件方向对应“起始位置#0”时，倾斜检测使能：如果器件从起始位置开始旋转角度大于 35 度，并且保持在蓝色区域中至少 2 秒的时间，则会产生第一次中断。第一次倾斜检测中断产生后，新的起始位置（#1）对应前次中断产生的器件位置（结束位置#0），当器件倾斜角度大于 35 度时，即产生下一个中断信号，进入起始位置#1 周围的蓝色区域。

图 21. 倾斜度检测



通过将 MD1_CFG 寄存器的 INT1_TILT 位或 MD2_CFG 寄存器的 INT2_TILT 位置为 1，可将此中断信号驱动至两个中断引脚；还可通过读取 FUNC_SRC1 寄存器的 TILT_IA 位来对其进行检查。

如果锁存模式禁用（TAP_CFG 的 LIR 位置为 0），则倾斜功能所产生的中断信号为脉冲的：中断引脚上观测到的脉冲持续时间约为 150 μ s；FUNC_SRC1 寄存器 TILT_IA 位上观测到的脉冲持续时间为 1/26 Hz。

如果锁存模式使能（TAP_CFG 的 LIR 位置为 1），且中断信号被驱动至中断引脚，那么当检测到倾斜时，对 FUNC_SRC1 寄存器的读取会将两个引脚上的请求以及 FUNC_SRC1 寄存器的 TILT_IA 位清零，器件准备识别下一次倾斜事件。如果锁存模式使能但是中断信号未驱动至中断引脚，那么 FUNC_SRC1 寄存器 TILT_IA 位上观测到的中断信号是脉冲，且持续时间固定为 1/26 Hz。

倾斜度检测功能工作于 26 Hz，因此加速度计 ODR 必须设置为 26 Hz 或更高的值。

下面是显示如何使能倾斜度检测功能的基本软件程序：

1. 将 20h 写入 CTRL1_XL // 启动加速度计
- // ODR_XL = 26 Hz, FS_XL = ± 2 g
2. 将 0Ch 写入 CTRL10_C // 使能嵌入功能
- // 使能倾斜度检测
3. 将 02h 写入 MD1_CFG // 倾斜探测器中断驱动至 INT1 引脚

6.4 绝对手腕倾斜

LSM6DSL 设备在硬件中实现了绝对手腕倾斜（AWT）功能：它能够检测可选加速度计半轴与水平面之间的角度何时高于特定的用户可选值。

AWT 功能仅基于加速度计传感器，工作于 26 Hz：要使用它，加速度计 ODR 必须设置为 26 Hz 或更高的值。

可以通过将 CTRL10_C 寄存器的 FUNC_EN 和 WRIST_TILT_EN 位置为 1，来启用 AWT 功能。

如果器件配置为进行绝对手腕倾斜事件检测，则当器件倾斜角度大于可配置阈值并持续倾斜了最小可配置时间时，会产生一个中断：如果倾斜角度大于阈值角度并且持续倾斜时间大于等于延迟时间，则会产生 AWT 中断信号。

默认情况下，AWT 中断应用于正 X 轴。通过将 DRDY_PULSE_CFG 寄存器的 INT2_WRIST_TILT 位置为 1，可将其驱动到 INT2 中断引脚，通过读取 FUNC_SRC2 寄存器的 WRIST_TILT_IA 位，还可以检查该位（如果通过设置寄存器 TAP_CFG 的 LIR 位为 1 将其锁存，则此读取会清除中断信号）。

如果锁存模式禁用（TAP_CFG 的 LIR 位置为 0），AWT 功能所产生的中断信号为脉冲的：中断引脚上观测到的脉冲持续时间约为 150 μ s；FUNC_SRC2 寄存器的 WRIST_TILT_IA 位上观测到的脉冲持续时间为 1/26 Hz。

如果锁存模式使能（TAP_CFG 的 LIR 位置为 1），且中断信号被驱动至中断引脚，那么当检测到手腕倾斜时，对 FUNC_SRC2 寄存器的读取会将两个引脚上的请求以及 FUNC_SRC2 寄存器的 WRIST_TILT_IA 位清零，器件准备识别下一次手腕倾斜事件。如果锁存模式使能但是中断信号未驱动至中断引脚，那么 FUNC_SRC2 寄存器 WRIST_TILT_IA 位上观测到的中断信号是脉冲，且持续时间固定为 1/26 Hz。

启用 AWT 功能默认配置的基本例程如下：

1. 将 20h 写入 CTRL1_XL // 启动加速度计
// ODR_XL = 26 Hz, FS_XL = ±2 g
2. 将 84h 写入 CTRL10_C // 使能嵌入功能
// 使能 AWT 检测
3. 将 01h 写入 DRDY_PULSE_CFG // AWT 中断驱动至 INT1 引脚

通过三个相关的嵌入功能寄存器（B 区）：A_WRIST_TILT_LAT、A_WRIST_TILT_THS 和 A_WRIST_TILT_Mask，可以调整延迟值、阈值和轴掩码。

通过设置嵌入寄存器 A_WRIST_TILT_LAT 的 WRIST_TILT_TIMER 字段，可以修改延迟参数：1 LSB = 40 ms，默认值为 0Fh，即 600 ms。

阈值参数可通过设置嵌入功能寄存器 A_WRIST_TILT_THS 的 WRIST_TILT_THS 字段进行配置。它是完全独立的。1 LSB 对应于 15.625 mg，WRIST_TILT_THS 字段必须 <40h（64d）。相对于水平面的倾斜角度可以如下计算：

$$angle[deg] = \frac{180}{\pi} * \arcsin \frac{WRIST_TILT_THS}{64}$$

寄存器 54h 的默认值是 20h = 32d = 500 mg，相当于 30 度的倾斜角度。

通过配置嵌入寄存器 A_WRIST_TILT_Mask 的位 WRIST_TILT_MASK_Xpos、WRIST_TILT_MASK_Xneg、WRIST_TILT_MASK_Ypos、WRIST_TILT_MASK_Yneg、WRIST_TILT_MASK_Zpos、WRIST_TILT_MASK_Zneg，用户可以选择 AWT 算法所考虑的轴；默认配置是 X 正轴 AWT 检测。可以选择另一个或另外的半轴来更改/修改 AWT 检测：应用这些位的“或”组合。

注：每次加速计退出下电模式时，这些嵌入功能寄存器（B 区）都会复位为其默认值，因此每次将电源模式从下电模式切换到活动模式时，都必须对它们进行重新配置。

当加速器退出下电模式时，要重复进行的完整 AWT 配置过程如下：

1. 将 20h 写入 CTRL1_XL // 启动加速度计
// ODR_XL = 26 Hz, FS_XL = ±2 g
2. 将 04h 写入 CTRL10_C // 使能嵌入功能
3. 等待 50 ms
4. 将 00h 写入 CTRL10_C // 禁用嵌入功能
5. 将 A0h 写入 FUNC_CFG_ACCESS // 使能对嵌入寄存器的访问（B 区）
6. 在 A_WRIST_TILT_LAT 中设置新的延迟
7. 在 A_WRIST_TILT_THS 中设置新的阈值
8. 在 A_WRIST_TILT_Mask 中设置新的掩码
9. 将 00h 写入 FUNC_CFG_ACCESS // 禁用对嵌入寄存器的访问（B 区）
10. 将 84h 写入 CTRL10_C // 使能嵌入功能
// 使能 AWT 检测
11. 将 01h 写入 DRDY_PULSE_CFG // AWT 中断驱动至 INT1 引脚



6.5 时间戳

同时利用传感器数据，LSM6DSL 器件能够给出时间戳信息。

如果加速度计和陀螺仪均处于下电模式，则时间戳计数器不工作。

要使用此功能，CTRL10_C 寄存器的 TIMER_EN 位必须置位 1：时间步数由 TIMESTAMP_REG2 & TIMESTAMP_REG1 & TIMESTAMP_REG0 寄存器输出，并表示为一个 24 位的无符号数。

时间戳精度可利用 WAKE_UP_DUR 寄存器的 WAKE_UP_DUR 位进行配置：该位被置为 0 时，1 LSB 的时间步数对应 6.4 ms（低精度模式）；该位被置为 1 时，1 LSB 的时间步数对应 25 μ s（高精度模式）。

当达到最大值 16777215 LSB（对应 FFFFFFFh）时，计数器自动复位至 000000h 并继续计数。可通过向 TIMESTAMP_REG2 寄存器中写入值 AAh，来随时将定时器计数复位至零。

在高精度模式（当定时器步数达到值 FF0000h 时）和低精度模式（当定时器步数达到值 FFFF00h）中，在定时器饱和和前约 1.638 秒时均会产生中断产生。通过将 MD1_CFG 寄存器的 INT1_TIMER 位置为 1，可将此中断信号驱动至 INT1 引脚。当中断引脚被产生，则它必须通过向 TIMESTAMP_REG2 寄存器中写入 AAh 来复位至 0（定时器步数也将复位）。

时间戳计数可与计步器数据一起作为第四数据集存储在 FIFO 里（详细信息见第 8.8 节 FIFO 中的计步器和时间戳数据）。

时间戳精度必须在使能时间戳功能前设置；基本程序如下：

1. 将 50h 写入 CTRL1_XL // 启动加速度计
// ODR_XL = 208 Hz, FS_XL = ± 2 g
2. 将 10h 写入 WAKE_UP_DUR // 时间戳精度 = 25 μ s
3. 将 20h 写入 CTRL10_C // 使能时间戳计数
4. 将 01h 写入 MD1_CFG // 计数器结束中断驱动至 INT1 引脚

当从低时间戳精度切换到高时间戳精度时，定时器计数必须复位，如以下示例所示：

1. 将 50h 写入 CTRL1_XL // 启动加速度计
// ODR_XL = 208 Hz, FS_XL = ± 2 g
2. 将 00h 写入 WAKE_UP_DUR // 时间戳精度 = 6.4 ms
3. 将 20h 写入 CTRL10_C // 使能时间戳计数
- ...
- N 将 10h 写入 WAKE_UP_DUR // 时间戳精度 = 25 μ s
- N+1 将 AAh 写入 TIMESTAMP_REG2 // 复位定时器计数器

7 模式 2 - 传感器集合 (sensor hub) 模式

LSM6DSL 的硬件灵活性允许通过不同的模式连接将引脚连接至外部传感器以扩展功能，如增加传感器集合 (sensor hub)。使能传感器集合模式 (模式 2) 时，用来连接外部传感器的主 I²C/SPI (3 线和 4 线) 从接口和 I²C 主接口都可用。模式 2 连接模式在下面的段落中详细描述。

7.1 传感器集合 (sensor hub) 模式说明

在传感器集合 (sensor hub) 模式 (模式 2) 下，可有 4 个外部传感器连接至 LSM6DSL 器件的 I²C 主接口。传感器集合触发信号可以与加速度计数据就绪信号 (最高 104Hz) 同步；或者，连接到 INT2 引脚的外部信号可用作传感器集合触发器。在第二种情况下，外部传感器支持的最大 ODR 取决于两个连续触发信号之间可以执行的读/写操作数。

在传感器集合触发信号上，通过寄存器 SLVx_ADD、SLVx_SUBADD、SLAVEx_CONFIG 和 DATAWRITE_SRC_MODE_SUB_SLV0 配置的所有写和读 I²C 操作都是从外部传感器 0 到外部传感器 3 顺序执行 (取决于由 SLAVE0_CONFIG 寄存器中 Aux_sens_on[1:0] 字段所使能的外部传感器)。

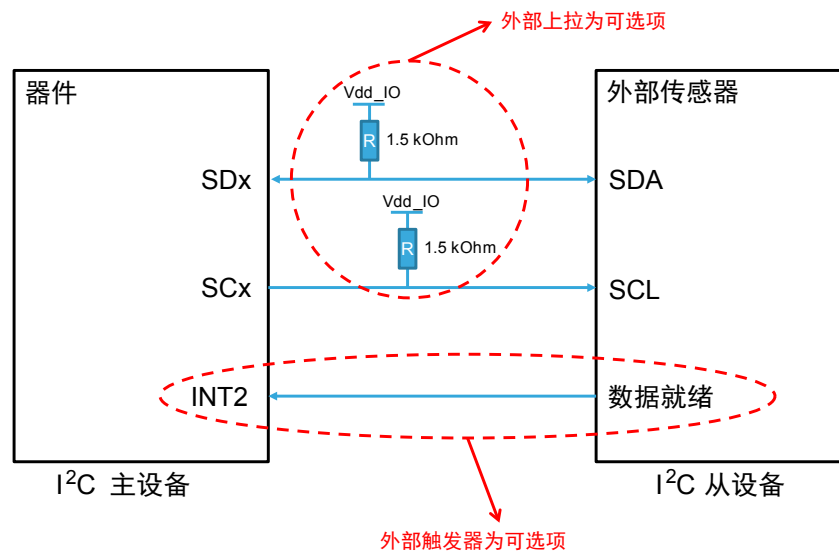
利用可配置抽取因子，外部传感器数据也可存储在 FIFO 中 (详细信息见第 8 节 先进先出 (FIFO) 缓冲区)。

如果加速度计和陀螺仪均处于下电模式，则传感器集合 (sensor hub) 不工作。

所有外部传感器必须并联到器件的 SDx/SCx 引脚，对于单个外部传感器，如图 22. 模式 2 下外部传感器连接所示。

外部上拉电阻和外部触发器信号连接是可选的，取决于寄存器配置。

图 22. 模式 2 下外部传感器连接



7.2 传感器集合 (sensor hub) 模式寄存器

要使能 LSM6DSL 嵌入功能，CTRL10_C 寄存器 FUNC_EN 位必须置为 1；使能嵌入功能后，MASTER_CONFIG 寄存器必须用来配置 I²C 主接口。

一组寄存器 SLVx_ADD、SLVx_SUBADD、SLAVEx_CONFIG 专门用来配置到 4 个可连接外部传感器的 4 个从接口。一个额外的寄存器 DATAWRITE_SRC_MODE_SUB_SLV0 只关联到从属#0：它可对从属#0 所关联外部传感器的寄存器实现写入以及源模式条件读取。

最后，有 18 个寄存器 (从 SENSORHUB1_REG 到 SENSORHUB18_REG) 可用来存储从外部传感器读取的数据。

7.2.1 CTRL10_C (19h)

表 31. CTRL10_C 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
X	0	X	X	X	FUNC _EN	X	X

- FUNC_EN 必须置为 1，以启用 LSM6DSL 的嵌入功能（计步器，倾斜，大幅运动，铁磁校正）。

7.2.2 MASTER_CONFIG (1Ah)

此寄存器用于配置 I²C 主控制器。

表 32. MASTER_CONFIG 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
DRDY_ ON_INT1	X	0	START_ CONFIG	PULL_ UP_EN	PASS_ THROUGH _MODE	X	主 _ON

- DRDY_ON_INT1 位必须置为 1，以驱动 INT1 引脚上的 I²C 主数据准备就绪信号（对应 FUNC_SRC1 寄存器的 SENSORHUB_END_OP 位特性）。关于 SENSORHUB_END_OP 位的更多详细信息，请参考第 7.2.3 节 FUNC_SRC1 (53h)。如果 DRDY_PULSE_CFG 寄存器的 DRDY_PULSED 位置为 1，则 I²C 主数据就绪信号会是脉冲的，脉冲持续时间为 150 μs。

START_CONFIG 位对传感器集合 (sensor hub) 触发信号进行选择。

- 当此位被置为 0 时，加速度计传感器必须激活（不能处于下电模式），传感器集合 (sensor hub) 触发信号为加速度计数据准备就绪信号，其频率对应加速度计 ODR，可达 104 Hz。
- 当此位被置为 1 时，加速度计和陀螺仪中必须激活至少一个传感器，并且传感器集合 (sensor hub) 触发信号到 INT2 引脚；实际上，当 MASTER_ON 位和 START_CONFIG 位被置为 1 时，INT2 引脚被配置为输入信号。这种情况下，INT2 引脚必须连接到外部传感器的数据准备就绪信号（图 22. 模式 2 下外部传感器连接），以触发对外部传感器寄存器的读取/写入操作。来自 INT2 的传感器集合中断是“高电平触发的”（不可编程）。

注意：使用外部触发信号的情况下（START_CONFIG=1），如果 INT2 引脚连接到外部传感器的数据准备就绪（图 22. 模式 2 下外部传感器连接），并且后者处于下电模式，那么外部传感器可以生成数据准备就绪信号。因此，必须使用内部触发信号来实现外部传感器寄存器的初始配置（START_CONFIG=0）。外部传感器激活且数据准备就绪信号可用之后，通过将 START_CONFIG 位切换为 1，可以使用外部触发信号。

- PULL_UP_EN 位使能/禁用辅 I²C 线路上的内部上拉。当此位被置为 0 时，内部上拉禁用，SDx/SCx 引脚上需要外部上拉电阻，如图 22. 模式 2 下外部传感器连接所示。当此位被置为 1 时，内部上拉使能，SDx/SCx 引脚上不需要外部上拉电阻。
- PASS_THROUGH_MODE 位用来使能/禁用 I²C 接口直通。当此位被置为 1，主 I²C 线（例如，连接到外部微控制器）短接到辅线，以实现对外部传感器寄存器的直接访问。详细信息，请参见第 7.3 节 传感器集合 (sensor hub) 直连功能。
- MASTER_ON 位必须置为 1，以启用 LSM6DSL 器件的辅助 I²C 主线（传感器集合 (sensor hub) 模式）。

7.2.3 FUNC_SRC1 (53h)

表 33. FUNC_SRC1 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
X	X	X	X	X	X	X	SENSOR HUB_ END_OP

- SENSORHUB_END_OP** 位报告 I²C 主器件的状态：当 I²C 主器件处于空闲状态时，此位等于 1；当 I²C 主器件处于读/写操作状态时，此位变为 0。
 当传感器集合例程完成时，此位自动变为 1，并且可以从 **SENSORHUBx_REG** 寄存器读取外部传感器数据（取决于 **SLVx_ADD**、**SLVx_SUBADD** 和 **SLAVEx_CONFIG** 寄存器的配置）。
注：如果 TAP_CFG 寄存器的 LIR 位置为 1，则 SENSORHUB_END_OP 位通过读取而被清零，否则该位仅在 I²C 主线读或写操作期间清零。
 通过将 **MASTER_CONFIG** 寄存器的 **DRDY_ON_INT1** 位置为 1，可将 I²C 主器件的状态信号驱动至 **INT1** 中断引脚：如果 **TAP_CFG** 寄存器的 **LIR** 位置为 0，则在 **SENSORHUB_END_OP** 信号的上升沿生成脉冲中断信号（典型脉冲持续时间为约 150 μs）。如果锁存模式使能（**LIR** 位置为 1）且中断信号被驱动至中断引脚 **INT1**，则通过读取 **FUNC_SRC1** 寄存器清除中断信号。

7.2.4 FUNC_SRC2 (54h)

表 34. FUNC_SRC2 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
0	SLAVE3_ _NACK	SLAVE2_ _NACK	SLAVE1_ _NACK	SLAVE0_ _NACK	X	0	X

- 如果在与相应从线 x 的通信过程中发生了“未确认”事件，则 **SLAVEx_NACK** 位会被置为 1。

7.2.5 SLV0_ADD (02h), SLV0_SUBADD (03h), SLAVE0_CONFIG (04h)

嵌入功能寄存器（当 **FUNC_CFG_ACCESS** 寄存器的 **FUNC_CFG_EN** 位被置为 1，**FUNC_CFG_EN_B** 位被置为 0 时可访问）用来配置关联到第一外部传感器的 I²C 从接口，下面对此进行介绍。

表 35. SLV0_ADD 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
Slave0_add6	Slave0_add5	Slave0_add4	Slave0_add3	Slave0_add2	Slave0_add1	Slave0_add0	rw_0

- Slave0_add[6:0]** 位用来指示第一个外部传感器的 I²C 从线地址。
- rw_0** 位对第一外部传感器进行读取/写入操作（0：写操作；1：读操作）。当发生下一个传感器集合 (sensor hub) 触发事件时，执行读取/写入操作。当 **rw_0** 位设置为 0（选择写入操作）时，**SENSORHUBx_REG** 寄存器的内容不会更新。

表 36. SLV0_SUBADD 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
Slave0_reg7	Slave0_reg6	Slave0_reg5	Slave0_reg4	Slave0_reg3	Slave0_reg2	Slave0_reg1	Slave0_reg0

- Slave0_reg[7:0]位用来指示要写入的第一个外部传感器寄存器的地址（如果 SLV0_ADD 寄存器的 rw_0 位置为 0）或要读取的第一个寄存器的地址（如果 rw_0 位被置为 1）。

表 37. SLAVE0_CONFIG 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
Slave0_rate1	Slave0_rate0	Aux_sens_on1	Aux_sens_on0	Src_mode	Slave0_numop2	Slave0_numop1	Slave0_numop0

- Slave0_rate[1:0]位用来定义从传感器集合 (sensor hub) 触发的第一个外部传感器上读操作所应用的抽取因子：
 - 00: 无抽取
 - 01: 每隔 2 次传感器集合 (sensor hub) 触发事件更新一次
 - 10: 每隔 4 次传感器集合 (sensor hub) 触发事件更新一次
 - 11: 每隔 8 次传感器集合 (sensor hub) 触发事件更新一次
- 必须用 Aux_sens_on[1:0]位来指示传感器集合 (sensor hub) 所管理外部传感器数：
 - 00: 无外部传感器
 - 01: 两个外部传感器
 - 10: 三个外部传感器
 - 11: 四个外部传感器
- Src_mode 位使能/禁用源模式条件读取。当此位被置为 1 时，源模式条件读取使能；继续读取 SLV0_SUBADD 寄存器中所示的寄存器地址之前，检查 DATAWRITE_SRC_MODE_SUB_SLV0 所指定地址上寄存器的内容：如果其内容非零，则操作继续，否则读取操作中断。源模式条件读取仅在从线 0 上可用。
- Slave0_numop[2:0]位专门用来定义从 SLV0_SUBADD 寄存器所示的寄存器地址开始的第一个外部传感器上执行的连续读取操作数。

7.2.6

SLV1_ADD (05h), SLV1_SUBADD (06h), SLAVE1_CONFIG (07h)

嵌入功能寄存器（当 FUNC_CFG_ACCESS 寄存器的 FUNC_CFG_EN 位被置为 1，FUNC_CFG_EN_B 位被置为 0 时可访问）用来配置关联到第二外部传感器的 I²C 从接口，下面对此进行介绍。

表 38. SLV1_ADD 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
Slave1_add6	Slave1_add5	Slave1_add4	Slave1_add3	Slave1_add2	Slave1_add1	Slave1_add0	r_1

- Slave1_add[6:0]位用来指示第二个外部传感器的 I²C 从线地址。
- r_1 位使能/禁用第二个外部传感器上执行的读取操作（0：读操作禁用；1：读操作使能）。当发生下一个传感器集合 (sensor hub) 触发事件时，执行读取操作。

表 39. SLV1_SUBADD 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
Slave1_reg7	Slave1_reg6	Slave1_reg5	Slave1_reg4	Slave1_reg3	Slave1_reg2	Slave1_reg1	Slave1_reg0

- Slave1_reg[7:0]位用来表示，当 SLV1_ADD 寄存器的 r_1 位被置为 1 时，要读取的第二个外部传感器寄存器的地址。

表 40. SLAVE1_CONFIG 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
Slave1_rate1	Slave1_rate0	写 _once	0	0	Slave1_numop2	Slave1_numop1	Slave1_numop0

- Slave1_rate[1:0]位用来定义从传感器集合（sensor hub）触发的第二个外部传感器上读操作所应用的抽取因子：
 - 00: 无抽取
 - 01: 每隔 2 次传感器集合（sensor hub）触发事件更新一次
 - 10: 每隔 4 次传感器集合（sensor hub）触发事件更新一次
 - 11: 每隔 8 次传感器集合（sensor hub）触发事件更新一次
- write_once 位用来限制从线 0 的写操作只发生一次（避免多次重复相同的写操作）。如果该位未被产生，那么每个 ODR 都会触发写操作。

注：要启用 write_once 功能，SLAVE0_CONFIG 寄存器中的 Aux_sens_on 字段必须不为 00b（即使只使用从 0）。

- Slave3_numop[2:0]位专门用来定义从 SLV3_SUBADD 寄存器所示的寄存器地址开始的第四个外部传感器上执行的连续读取操作数。

7.2.7

SLV2_ADD (08h), SLV2_SUBADD (09h), SLAVE2_CONFIG (0Ah)

嵌入功能寄存器（当 FUNC_CFG_ACCESS 寄存器的 FUNC_CFG_EN 位被置为 1，FUNC_CFG_EN_B 位被置为 0 时可访问）用来配置关联到第三外部传感器的 I²C 从接口，下面对此进行介绍。

表 41. SLV2_ADD 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
Slave2_add6	Slave2_add5	Slave2_add4	Slave2_add3	Slave2_add2	Slave2_add1	Slave2_add0	r_2

- Slave2_add[6:0]位用来指示第三个外部传感器的 I²C 从线地址。
- r_2 位使能/禁用第三个外部传感器上执行的读取操作（0：读操作禁用；1：读操作使能）。当发生下一个传感器集合（sensor hub）触发事件时，执行读取操作。

表 42. SLV2_SUBADD 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
Slave2_reg7	Slave2_reg6	Slave2_reg5	Slave2_reg4	Slave2_reg3	Slave2_reg2	Slave2_reg1	Slave2_reg0

- Slave3_reg[7:0]位用来表示，当 SLV3_ADD 寄存器的 r_3 位被置为 1 时，要读取的第四个外部传感器寄存器的地址。

表 43. SLAVE2_CONFIG 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
Slave2_rate1	Slave2_rate0	0	0	0	Slave2_numop2	Slave2_numop1	Slave2_numop0

- Slave2_rate[1:0]位用来定义从传感器集合（sensor hub）触发的第三个外部传感器上读操作所应用的抽取因子：
 - 00：无抽取
 - 01：每隔 2 次传感器集合（sensor hub）触发事件更新一次
 - 10：每隔 4 次传感器集合（sensor hub）触发事件更新一次
 - 11：每隔 8 次传感器集合（sensor hub）触发事件更新一次
- Slave2_numop[2:0]位专门用来定义从 SLV2_SUBADD 寄存器所示的寄存器地址开始的第三个外部传感器上执行的连续读取操作数。

7.2.8

SLV3_ADD (0Bh), SLV3_SUBADD (0Ch), SLAVE3_CONFIG (0Dh)

嵌入功能寄存器（当 FUNC_CFG_ACCESS 寄存器的 FUNC_CFG_EN 位被置为 1，FUNC_CFG_EN_B 位被置为 0 时可访问）用来配置关联到第四外部传感器的 I²C 从接口，下面对此进行介绍。

表 44. SLV3_ADD 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
Slave3_add6	Slave3_add5	Slave3_add4	Slave3_add3	Slave3_add2	Slave3_add1	Slave3_add0	r_3

- Slave3_add[6:0]位用来指示第四个外部传感器的 I²C 从线地址。
- r_3 位使能/禁用第四个外部传感器上执行的读取操作（0：读操作禁用；1：读操作使能）。当发生下一个传感器集合 (sensor hub) 触发事件时，执行读取操作。

表 45. SLV3_SUBADD 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
Slave3_reg7	Slave3_reg6	Slave3_reg5	Slave3_reg4	Slave3_reg3	Slave3_reg2	Slave3_reg1	Slave3_reg0

- Slave3_reg[7:0]位用来表示，当 SLV3_ADD 寄存器的 r_3 位被置为 1 时，要读取的第四个外部传感器寄存器的地址。

表 46. SLAVE3_CONFIG 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
Slave3_rate1	Slave3_rate0	0	0	0	Slave3_numop2	Slave3_numop1	Slave3_numop0

- Slave3_rate[1:0]位用来定义从传感器集合 (sensor hub) 触发的第四个外部传感器上读操作所应用的抽取因子：
 - 00：无抽取
 - 01：每隔 2 次传感器集合 (sensor hub) 触发事件更新一次
 - 10：每隔 4 次传感器集合 (sensor hub) 触发事件更新一次
 - 11：每隔 8 次传感器集合 (sensor hub) 触发事件更新一次
- Slave3_numop[2:0]位专门用来定义从 SLV3_SUBADD 寄存器所示的寄存器地址开始的第四个外部传感器上执行的连续读取操作数。

7.2.9 DATAWRITE_SRC_MODE_SUB_SLV0 (0Eh)

表 47. DATAWRITE_SRC_MODE_SUB_SLV0 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
Slave_dataw7	Slave_dataw6	Slave_dataw5	Slave_dataw4	Slave_dataw3	Slave_dataw2	Slave_dataw1	Slave_dataw0

- 当 SLV0_ADD 寄存器的 rw_0 位被置为 0 (写操作) 时, Slave_dataw[7:0] 位专门用来表示要写入到第一个外部传感器 (其地址在 SLV0_SUBADD 寄存器中指定) 的数据。读操作过程中 (rw_0 = 1), 如果源模式条件读取使能, 则使用此寄存器 (SLAVE0_CONFIG 寄存器中 Src_mode bit = 1), 它表示在继续进行读操作前所要检查的外部传感器寄存器地址。

7.2.10 SENSORHUBx_REG 寄存器

当辅助 I²C 主线使能时, 每个外部传感器读取若干寄存器, 该数等于 Slavex_numop (x = 0, 1, 2, 3) 字段的值, 从 SLVx_SUBADD (x = 0, 1, 2, 3) 寄存器指定的寄存器地址开始。要处理的外部传感器数在 SLAVE0_CONFIG 寄存器的 Aux_sens_on 位中指定。

读取的数据连续存储在 (按照与读取它们相同的顺序) LSM6DSL 寄存器中, 从 SENSORHUB1_REG 寄存器开始, 如图 23. SENSORHUBx_REG 配置示例中示例所示; 18 个寄存器, 从 SENSORHUB1_REG 到 SENSORHUB18_REG, 可以用来存储从外部传感器中读取的数据。

从 SENSORHUB1_REG 到 SENSORHUB6_REG 寄存器的值可作为第三数据集保存在 FIFO 缓冲器中; 从 SENSORHUB7_REG 到 SENSORHUB12_REG 寄存器的值可作为第四数据集保存在 FIFO 缓冲器中 (详细信息见第 8 节 先进先出 (FIFO) 缓冲区)。

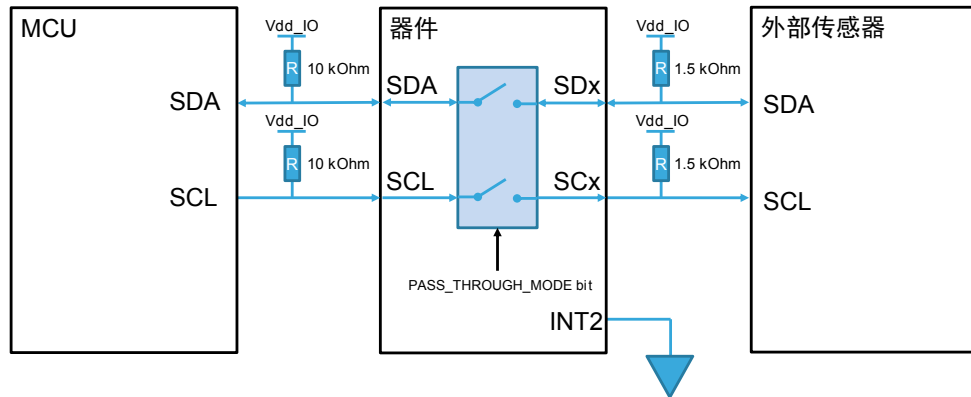
图 23. SENSORHUBx_REG 配置示例

传感器 #1	$\left\{ \begin{array}{l} \text{SLV0_SUBADD}(03\text{h}) = 28\text{h} \\ \text{SLAVE0_CONFIG}(04\text{h}) - \text{Slave0_numop}[2:0] = 3 \end{array} \right.$	SENSORHUB1_REG	Value of reg 28h	传感器 #1
		SENSORHUB2_REG	Value of reg 29h	
		SENSORHUB3_REG	Value of reg 2Ah	
传感器 #2	$\left\{ \begin{array}{l} \text{SLV1_SUBADD}(06\text{h}) = 00\text{h} \\ \text{SLAVE1_CONFIG}(07\text{h}) - \text{Slave1_numop}[2:0] = 6 \end{array} \right.$	SENSORHUB4_REG	Value of reg 00h	传感器 #2
		SENSORHUB5_REG	Value of reg 01h	
		SENSORHUB6_REG	Value of reg 02h	
传感器 #3	$\left\{ \begin{array}{l} \text{SLV2_SUBADD}(09\text{h}) = 20\text{h} \\ \text{SLAVE2_CONFIG}(0Ah) - \text{Slave2_numop}[2:0] = 4 \end{array} \right.$	SENSORHUB7_REG	Value of reg 03h	传感器 #3
		SENSORHUB8_REG	Value of reg 04h	
		SENSORHUB9_REG	Value of reg 05h	
传感器 #4	$\left\{ \begin{array}{l} \text{SLV3_SUBADD}(0Ch) = 40\text{h} \\ \text{SLAVE3_CONFIG}(0Dh) - \text{Slave3_numop}[2:0] = 5 \end{array} \right.$	SENSORHUB10_REG	Value of reg 20h	传感器 #3
		SENSORHUB11_REG	Value of reg 21h	
		SENSORHUB12_REG	Value of reg 22h	
传感器 #4		SENSORHUB13_REG	Value of reg 23h	传感器 #4
		SENSORHUB14_REG	Value of reg 40h	
		SENSORHUB15_REG	Value of reg 41h	
传感器 #4		SENSORHUB16_REG	Value of reg 42h	传感器 #4
		SENSORHUB17_REG	Value of reg 43h	
		SENSORHUB18_REG	Value of reg 44h	

7.3 传感器集合 (sensor hub) 直连功能

MASTER_CONFIG 寄存器的 PASS_THROUGH_MODE 位用来使能/禁用 I²C 接口直连功能：当此位被置为 1，主 I²C 线（例如，连接到外部微控制器）短接到辅线，以实现对外部传感器寄存器的直接访问。建议在配置外部传感器时使用此功能。

图 24. 直连功能



使用传感器集合 (sensor hub) 和直连功能时必须考虑一些限制。可能有如下三种情形：

1. 传感器集合 (sensor hub) 在 MASTER_CONFIG 寄存器的 START_CONFIG 位被置为 0 时使用（内部触发），且直连功能不使用：在使用 INT2 引脚时无限制。
2. 传感器集合 (sensor hub) 在 MASTER_CONFIG 寄存器的 START_CONFIG 位被置为 0 时使用（内部触发），且直连功能使用：INT2 引脚必须接地；不能切换到外部触发配置（通过将 START_CONFIG 位置为 1），并且 INT2 引脚上不能使用数字中断。必须使用特定的步骤来使能/禁用直连功能：在第 7.3.1 节 直连功能使能和第 7.3.2 节 直连功能禁用中有相应描述。
3. 传感器集合 (sensor hub) 在 MASTER_CONFIG 寄存器的 START_CONFIG 位被置为 1 时使用（外部触发）：直连功能不能使用；INT2 引脚必须连接到外部传感器的数据准备就绪引脚（触发信号），并且必须执行以下步骤来避免与 INT2 线的冲突：
 - a. 将 CTRL6_C 寄存器的 TRIG_EN 或 LVL1_EN 或 LVL2_EN 位置为 1（将 INT2 引脚配置为输入引脚）；
 - b. 配置外部传感器（不使用直连功能）；
 - c. 配置传感器集合 (sensor hub) SLAVEx 寄存器；
 - d. 将 MASTER_CONFIG 寄存器的 START_CONFIG 位置 1；
 - e. 将 MASTER_CONFIG 寄存器的 MASTER_ON 位置 1；
 - f. 在步骤 a 中将 CTRL6_C 寄存器中的位复位为 0。

不使用直连功能的外部传感器配置示例在第 7.4 节 传感器集合 (sensor hub) 模式示例和第 7.5.4 节 铁磁校准示例中给出。



7.3.1 直连功能使能

当嵌入传感器集合 (sensor hub) 功能禁用时, 可通过将 MASTER_CONFIG 寄存器的 PASS_THROUGH_MODE 位置为 1, 来随时使能直连功能。

当嵌入传感器集合 (sensor hub) 功能使能时, 必须按照特定步骤来使能直连功能, 以防止 I²C 总线仲裁丢失:

1. 将 MASTER_CONFIG 寄存器的 START_CONFIG 位置为 1, 以禁用传感器集合 (sensor hub) 触发 (外部触发使能, 但是在 INT2 引脚上不能接收到触发, 因为该引脚连接到 GND);
2. 至少等待 5 ms (将会完成正在运行的 I²C 操作);
3. 将 MASTER_CONFIG 寄存器的 MASTER_ON 位置为 0, 以禁用嵌入传感器集合 (sensor hub);
4. 将 MASTER_CONFIG 寄存器的 START_CONFIG 位置为 0, 以恢复传感器集合 (sensor hub) 触发;
5. 将 MASTER_CONFIG 寄存器的 PULL_UP_EN 位置为 0, 以禁用 I²C 主线上拉;
6. 将 MASTER_CONFIG 寄存器的 PASS_THROUGH_MODE 位置为 1, 以使能直连功能。

7.3.2 直连功能禁用

必须使用以下步骤来禁用直连:

1. 等待外部微控制器连接到主 I²C 线完成所有正在运行的 I²C 操作。在一个 I²C 事务过程中, 直连功能无法禁用;
2. 将 MASTER_CONFIG 寄存器的 PASS_THROUGH_MODE 位置 0。

这时, 可通过将 MASTER_CONFIG 寄存器的 PULL_UP_EN 位置为 1, 来恢复内部 I²C 主线上拉, 可通过将 MASTER_CONFIG 寄存器的 MASTER_ON 位置为 1, 来使能辅助 I²C 主线。

7.4 传感器集合 (sensor hub) 模式示例

应利用直连功能来实现外部传感器配置: 此功能可通过将 MASTER_CONFIG 寄存器的 PASS_THROUGH_MODE 位置为 1 来使能, 并且能够实现对外部传感器寄存器的直接访问, 允许进行快速配置。

下面提供的代码给出了将 LSM6DSL 配置为传感器集合 (sensor hub) 模式的基本程序。而且, 该程序将 LIS2MDL 外部磁力计传感器 (额外细节参考数据手册) 配置为 Continue 模式, 并读取磁力计输出寄存器, 将其值保存到 SENSORHUB1_REG 到 SENSORHUB6_REG 寄存器中。本例中不使用直连功能。

- | | |
|---|---------------------------------------|
| 1. 将 80h 写入 FUNC_CFG_ACCESS | // 使能对嵌入功能寄存器的访问 (A 区) |
| 2. 将 3Ch 写入 SLV0_ADD | // LIS2MDL 从设备地址 = 0011110b |
| | // 使能写操作 (rw_0=0) |
| 3. 将 60h 写入 SLV0_SUBADD | // 60h 为待写入的 LIS2MDL 寄存器 |
| 4. 将 8Ch 写入 DATAWRITE_SRC_MODE_SUB_SLV0 | // 8Ch 为写入到 LIS2MDL 的寄存器 60h 的值 |
| | // 以将其配置为 Continue 模式, |
| | // ODR = 100 Hz, 温度补偿使能 |
| 5. 将 10h 写入 SLAVE0_CONFIG | // 将 Aux_sens_on 位设置为不等于 00b |
| 6. 将 20h 写入 SLAVE1_CONFIG | // 使能 write_once 位 |
| 7. 将 00h 写入 FUNC_CFG_ACCESS | // 禁用对嵌入功能寄存器的访问 (A 区) |
| 8. 将 04h 写入 CTRL10_C | // 使能嵌入功能 |
| 9. 将 09h 写入 MASTER_CONFIG | // 使能 SDx/SCx 线上的内部上拉 |
| | // 传感器集合 (sensor hub) 触发信号为 XL 数据准备就绪 |
| | // 使能辅助 I ² C 主线 |
| 10. 将 80h 写入 CTRL1_XL | // 开启加速度计 (以获取触发信号) |
| 11. 读取 FUNC_SRC1 | // 等待传感器集合通信结束 |
| 12. 如果 SENSORHUB_END_OP = 0, 转到 9 | |
| 13. 将 00h 写入 CTRL10_C | // 禁用嵌入功能 |
| 14. 将 00h 写入 MASTER_CONFIG | // 禁用辅助 I ² C 主线 |



- | | |
|------------------------------|---------------------------------------|
| 15. 将 00h 写入 CTRL1_XL | // 关闭加速度计 |
| 16. 将 80h 写入 FUNC_CFG_ACCESS | // 使能对嵌入功能寄存器的访问 (A 区) |
| 17. 将 3Dh 写入 SLV0_ADD | // LIS2MDL 从设备地址 = 0011110b |
| | // 使能读操作 (rw_0=1) |
| 18. 将 68h 写入 SLV0_SUBADD | // 68h 为待读取的第一个 LIS2MDL 输出寄存器 |
| 19. 将 06h 写入 SLAVE0_CONFIG | // 无抽取 |
| | // 连接了 1 个外部传感器 |
| | // 读取的寄存器数 = 6 |
| 20. 将 00h 写入 FUNC_CFG_ACCESS | // 禁用对嵌入功能寄存器的访问 (A 区) |
| 21. 将 04h 写入 CTRL10_C | // 使能嵌入功能 |
| 22. 将 09h 写入 MASTER_CONFIG | // 使能 SDx/SCx 线上的内部上拉 |
| | // 传感器集合 (sensor hub) 触发信号为 XL 数据准备就绪 |
| | // 使能辅助 I ² C 主线 |
| 23. 将 80h 写入 CTRL1_XL | // 开启加速度计 (以获取触发信号) |

7.5 磁力计硬磁/软磁校准

LSM6DSL 器件支持外部磁力计数据采集，并具有软磁和硬磁校准功能。为此，需要将 MASTER_CONFIG 寄存器的 MASTER_ON 位置为 1 来使能传感器集合（sensor hub）模式，将外部磁力计关联到从 0 寄存器（SLV0_ADD, SLV0_SUBADD 和 SLAVE0_CONFIG），并将 SLAVE0_CONFIG 的 Slave0_numop 字段置为 6。

CTRL10_C 寄存器的 FUNC_EN 位必须置位 1，以使能嵌入铁磁校准功能。然后，可使能校准算法，如表 48. 铁磁校准配置中所述：MASTER_CONFIG 的 IRON_EN 位和 CTRL9_XL 的 SOFT_EN 位可用来单独使能硬磁校准，也可同时使能硬磁和软磁校准。后一种情况下，校准（硬磁 & 软磁）和非校准（仅限软磁）磁力计数据均可用。

表 48. 铁磁校准配置

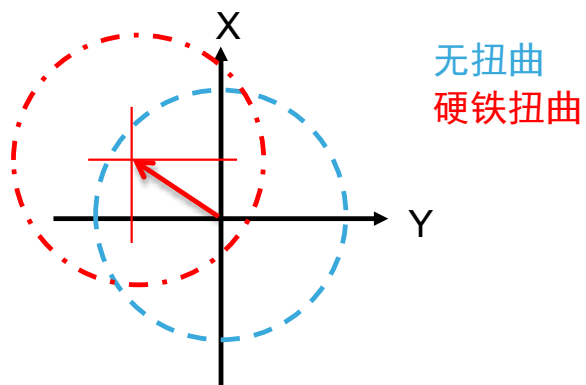
CTRL9_XL SOFT_EN 位	MASTER_CONFIG IRON_EN 位	铁磁校准 配置
0	0	不使用校准
0	1	仅用硬磁校准
1	1	硬磁 + 软磁校准

7.5.1 硬磁校准

硬磁失真通常由具有永久磁场的铁磁材料产生，该永久磁场是所用物体（例如，平板）的一部分；这些材料可以是永磁体或磁化铁或钢。它们是时不变的，并且使局部磁场发生扭曲，在不同方向上产生不同偏移。

通常，如果用户在理想环境中对对象执行多次 3D 旋转（无硬磁/软磁失真），并画出所采集的磁传感器原始数据，其结果将是无偏移的完美球体。硬磁失真效应是使球体沿着 X、Y 和 Z 轴偏移；在 X-Y 平面上，硬磁失真利用理想圆原点到(0, 0)的偏移来识别，如图 25. 硬磁效应（X-Y 2D 散点图）中所示。

图 25. 硬磁效应 (X-Y 2D 散点图)



在 LSM6DSL 器件中，3x1 硬磁化矢量包含用户计算出的 X、Y、Z 磁偏移值，必须由专门的寄存器来显示：MAG_OFFX_L 和 MAG_OFFX_H 寄存器专门用于 X-轴偏移，MAG_OFFY_L 和 MAG_OFFY_H 寄存器专门用于 Y-轴偏移，MAG_OFFZ_L 和 MAG_OFFZ_H 寄存器专门用于 Z-轴偏移。这些寄存器值以二进制补码的形式表示为 16 位字；计算硬磁寄存器值所应用的灵敏度[LSB/Gauss]对应于外部磁力计的灵敏度。

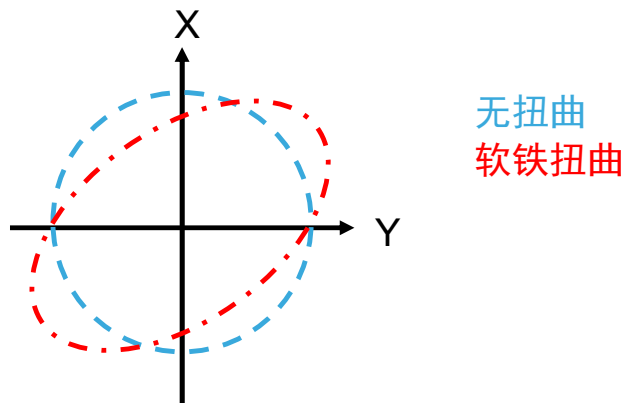
当 FUNC_CFG_ACCESS 寄存器的 FUNC_CFG_EN 位置为 1 时，可访问硬磁寄存器。要使能硬磁校准算法，需要将 CTRL10_C 寄存器的 FUNC_EN 位和 MASTER_CONFIG 寄存器的 IRON_EN 位均置为 1（表 48. 铁磁校准配置）。

7.5.2 软磁校准

软磁失真由软磁材料或载流 PCB 走迹线产生。硬磁失真是恒定的，与方向无关，而软磁失真会随着对象在地球场中的方向而改变。基本上，局部地磁场会发生变形，在不同方向上有不同增益。

软磁失真的效果是使理想的全圆球体变成倾斜的椭球体；在 X-Y 平面上，软磁失真利用原点在(0, 0)的倾斜椭球体来识别，如图 25. 硬磁效应 (X-Y 2D 散点图) 中所示。

图 26. 软磁效应 (X-Y 2D 散点图)





LSM6DSL 器件中，用户计算出的 3x3 软磁变换矩阵必须由 9 个专门的寄存器来显示：MAG_SI_XX，MAG_SI_XY，MAG_SI_XZ，MAG_SI_YX，MAG_SI_YY，MAG_SI_YZ，MAG_SI_ZX，MAG_SI_ZY，MAG_SI_ZZ。这些寄存器值以有符号数据格式表示为 8 位字；对于这些寄存器，1 LSB 对应 1/8，因此在将用户计算出的矩阵参数写入软磁寄存器之前，它们必须乘以 8。

当 FUNC_CFG_ACCESS 寄存器的 FUNC_CFG_EN 位置为 1 时，可访问软磁寄存器。要使能软磁校准算法，需要将 CTRL10_C 寄存器的 FUNC_EN 位、MASTER_CONFIG 寄存器的 IRON_EN 位和 CTRL9_XL 寄存器的 SOFT_EN 位均置为 1（表 48. 铁磁校准配置）。

7.5.3 得到磁力计补偿数据

可利用 FUNC_SRC1 寄存器对磁力计数据采集和硬磁/软磁校准的状态进行检查：

- 当传感器集合（sensor hub）程序完成时，SENSORHUB_END_OP 位被置为高电平。采集的磁力计原始数据可在地址从 66h（OUT_MAG_RAW_X_L）至 6Bh（OUT_MAG_RAW_Z_H）的寄存器中获取。
- 当已使能的硬磁和软磁算法执行完成时，SI_END_OP 位被置为高电平。如果软磁校准使能，磁力计未校准数据（仅应用软磁）可在地址从 4Dh（SENSORHUB13_REG）至 52h（SENSORHUB18_REG）的寄存器中获取。磁力计校准数据，使用了硬磁（如果使能）和软磁（如果使能）校准，可在地址从 2Eh（SENSORHUB1_REG）至 33h（SENSORHUB6_REG）的寄存器中获取。

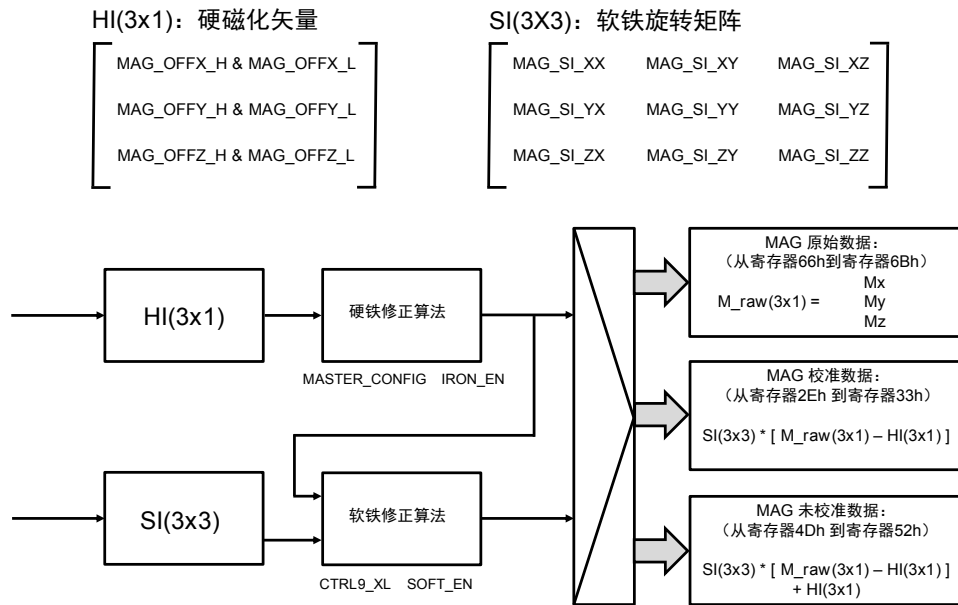
如果锁存模式禁用（TAP_CFG 的 LIR 位置为 0），则 SENSORHUB_END_OP 和 SI_END_OP 位仅在 1/100 Hz 下激活，然后它们自动取消。如果锁存模式使能，通过读取 FUNC_SRC1 寄存器，可使这两位清零。

通过将 MASTER_CONFIG 寄存器的 DRDY_ON_INT1 位置为 1，可将 SENSORHUB_END_OP 信号驱动至 INT1 引脚。通过将 MD2_CFG 寄存器的 INT2_IRON 位置为 1，可将 SI_END_OP 信号驱动至 INT1 引脚。

硬磁和软磁校准特性图示如下面的图 27. 硬磁/软磁校准框图所示。

如果软磁校准使能，且软磁寄存器仍然具有默认零值，那么磁力计校准数据和磁力计非标准数据也将等于零。因此，当软磁校准使能，软磁变换矩阵必须至少初始化为单位矩阵乘以 8，将 MAG_SI_XX、MAG_SI_YY 和 MAG_SI_ZZ 寄存器的值置为 08h。

图 27. 硬磁/软磁校准框图



7.5.4 铁磁校准示例

以下示例演示了如何从计算出的硬磁矢量和软磁旋转矩阵开始，定义分配到硬磁和软磁校准寄存器的值。本例中涉及 LIS2MDL 磁力计传感器的使用。

- 硬磁(X,Y,Z)偏移值矢量（高斯）：

$$HI(3x1) = \begin{bmatrix} -0.335605 \\ 0.126487 \\ -0.114722 \end{bmatrix}$$

这三个偏移值必须除以 LIS2MDL 灵敏度值（0.0015 高斯/LSB），以获得要写入到硬磁校准寄存器的 LSB 值（表 49. 硬磁寄存器值）。

表 49. 硬磁寄存器值

	偏移值[LSB]	寄存器值
X	-224 (FF20h)	MAG_OFFX_H = FFh MAG_OFFX_L = 20h
是	84 (0054h)	MAG_OFFY_H = 00h MAG_OFFY_L = 54h
Z	-76 (FFB4h)	MAG_OFFZ_H = FFh MAG_OFFZ_L = B4h

- 软磁旋转矩阵:

$$SI(3 \times 3) = \begin{bmatrix} 1.229006 & 0.173917 & 0.052327 \\ 0.173917 & 1.033307 & -0.130089 \\ 0.052327 & -0.130089 & 1.243645 \end{bmatrix}$$

这些软磁矩阵元素必须乘以 8，以获得要写入到软磁校准寄存器的 LSB 值（表 50. 软磁寄存器值）。LSB 值表示为有符号数据格式。

表 50. 软磁寄存器值

	软磁矩阵元素	寄存器值
XX	+1.229006	MAG_SI_XX = 0Ah
XY	+0.173917	MAG_SI_XY = 01h
XZ	+0.052327	MAG_SI_XZ = 00h
YX	+0.173917	MAG_SI_YX = 01h
YY	+1.033307	MAG_SI_YY = 08h
YZ	-0.130089	MAG_SI_YZ = 81h
ZX	+0.052327	MAG_SI_ZX = 00h
ZY	-0.130089	MAG_SI_ZY = 81h
ZZ	+1.243645	MAG_SI_ZZ = 0Ah

下面提供的代码给出了将 LIS2MDL 外部磁力计传感器（更多详细信息请参考数据手册）配置为 **Continue** 模式，初始化硬磁和软磁校准寄存器，以及读取磁力计输出寄存器的基本程序。本实例中，磁力计配置中不使用直连功能。

- 将 80h 写入 FUNC_CFG_ACCESS // 使能对嵌入功能寄存器的访问（A 区）
- 将 3Ch 写入 SLV0_ADD // LIS2MDL 从设备地址 = 0011110b
// 使能写操作（rw_0=0）
- 将 60h 写入 SLV0_SUBADD // 60h 为待写入的 LIS2MDL 寄存器
- 将 8Ch 写入 DATAWRITE_SRC_MODE_SUB_SLV0 // 8Ch 为写入到 LIS2MDL 的寄存器 60h 的值
// 以将其配置为 **Continue** 模式，
// ODR = 100 Hz，温度补偿使能
- 将 10h 写入 SLAVE0_CONFIG // 将 Aux_sens_on 位设置为不等于 00b
- 将 20h 写入 SLAVE1_CONFIG // 使能 write_once 位
- 将 00h 写入 FUNC_CFG_ACCESS // 禁用对嵌入功能寄存器的访问（A 区）
- 将 04h 写入 CTRL10_C // 使能嵌入功能



- | | |
|-----------------------------------|---------------------------------------|
| 9. 将 09h 写入 MASTER_CONFIG | // 使能 SDx/SCx 线上的内部上拉 |
| | // 传感器集合 (sensor hub) 触发信号为 XL 数据准备就绪 |
| | // 使能辅助 I ² C 主线 |
| 10. 将 80h 写入 CTRL1_XL | // 开启加速度计 (以获取触发信号) |
| 11. 读取 FUNC_SRC1 | // 等待传感器集合通信结束 |
| 12. 如果 SENSORHUB_END_OP = 0, 转到 9 | |
| 13. 将 00h 写入 CTRL10_C | // 禁用嵌入功能 |
| 14. 将 00h 写入 MASTER_CONFIG | // 禁用辅助 I ² C 主线 |
| 15. 将 00h 写入 CTRL1_XL | // 关闭加速度计 |
| 16. 将 80h 写入 FUNC_CFG_ACCESS | // 使能对嵌入功能寄存器的访问 (A 区) |
| 17. 将 3Dh 写入 SLV0_ADD | // LIS2MDL 从设备地址 = 0011110b |
| | // 使能读操作 (rw_0=1) |
| 18. 将 68h 写入 SLV0_SUBADD | // 68h 为待读取的第一个 LIS2MDL 输出寄存器 |
| 19. 将 06h 写入 SLAVE0_CONFIG | // 无抽取 |
| | // 连接了 1 个外部传感器 |
| | // 读取的寄存器数 = 6 |
| 20. 将 FFh 写入 MAG_OFFX_H | // X 偏移值初始化 |
| 21. 将 20h 写入 MAG_OFFX_L | // X 偏移值初始化 |
| 22. 将 00h 写入 MAG_OFFY_H | // Y 偏移值初始化 |
| 23. 将 54h 写入 MAG_OFFY_L | // Y 偏移值初始化 |
| 24. 将 FFh 写入 MAG_OFFZ_H | // Z 偏移值初始化 |
| 25. 将 B4h 写入 MAG_OFFZ_L | // Z 偏移值初始化 |
| 26. 将 0Ah 写入 MAG_SI_XX | // XX 软磁参数 |
| 27. 将 01h 写入 MAG_SI_XY | // XY 软磁参数 |
| 28. 将 00h 写入 MAG_SI_XZ | // XZ 软磁参数 |
| 29. 将 01h 写入 MAG_SI_YX | // YX 软磁参数 |
| 30. 将 08h 写入 MAG_SI_YY | // YY 软磁参数 |
| 31. 将 81h 写入 MAG_SI_YZ | // YZ 软磁参数 |
| 32. 将 00h 写入 MAG_SI_ZX | // ZX 软磁参数 |
| 33. 将 81h 写入 MAG_SI_ZY | // ZY 软磁参数 |
| 34. 将 0Ah 写入 MAG_SI_ZZ | // ZZ 软磁参数 |
| 35. 将 00h 写入 FUNC_CFG_ACCESS | // 禁用对嵌入功能寄存器的访问 (A 区) |
| 36. 将 04h 写入 CTRL10_C | // 使能嵌入功能 |
| 37. 将 0Bh 写入 MASTER_CONFIG | // 使能 SDx/SCx 线上的内部上拉 |
| | // 传感器集合 (sensor hub) 触发信号为 XL 数据准备就绪 |
| | // 使能硬磁校准 |
| | // 使能辅助 I ² C 主线 |
| 38. 将 04h 写入 CTRL9_XL | // 使能软磁校准 |
| 39. 将 80h 写入 CTRL1_XL | // 开启加速度计 (以获取触发信号) |

采集的磁力计原始数据可在地址从 66h (OUT_MAG_RAW_X_L) 至 6Bh (OUT_MAG_RAW_Z_L) 的寄存器中获取。



磁力计非校准数据（仅使用软磁）可在地址从 4Dh（SENSORHUB13_REG）至 52h（SENSORHUB18_REG）的寄存器中获取。

磁力计校准数据（使用了硬磁和软磁校准）可在地址从 2Eh（SENSORHUB1_REG）至 33h（SENSORHUB6_REG）的寄存器中获取。

8 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，LSM6DSL 嵌入了一个 4 kB 的先进先出缓冲器 (FIFO)。FIFO 可配置为存储以下数据：

- 陀螺仪传感器数据；
- 加速度计传感器数据；
- 外部传感器（连接到传感器集合 (sensor hub) 接口）数据；
- 计步器和时间戳数据；
- 温度传感器数据。

将数据保存在 FIFO 缓冲器是基于四个“FIFO 数据集”，每个数据集由 6 个字节构成：

- 第 1 FIFO 数据集预留给陀螺仪数据；
- 第 2 FIFO 数据集预留给加速度计数据；
- 第 3 FIFO 数据集预留给存储在从 SENSORHUB1_REG 至 SENSORHUB6_REG 的寄存器中外部传感器数据（关于 SENSORHUBx_REG 的详细信息，请参考第 7.2.10 节 SENSORHUBx_REG 寄存器）；
- 第 4 FIFO 数据集可交替关联到外部传感器数据（存储在从 SENSORHUB7_REG 至 SENSORHUB12_REG 的寄存器中）、计步器和时间戳信息，或温度传感器数据。

通过设置 FIFO_CTRL3 和 FIFO_CTRL4 寄存器的抽取因子，所有这些数据集能够以不同的 ODR 存储在 FIFO 中。抽取因子也可用来选择哪个 FIFO 数据集必须存储在 FIFO 中。

利用 FIFO_CTRL5 寄存器的 FIFO_MODE_[2:0] 位，有五种不同的 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。

注意：当使用 FIFO 时，CTRL3_C 寄存器的 IF_INC 和 BDU 位必须等于 1。

通过两个专用寄存器从 FIFO 中恢复数据：FIFO_DATA_OUT_L 和 FIFO_DATA_OUT_H。这样，数据可从 FIFO（以较低 ODR）或从器件输出寄存器（以正常 ODR）进行读取。

要监控 FIFO 状态（满，空，存储的采样数，等等），可以使用四个专用寄存器：FIFO_STATUS1，FIFO_STATUS2，FIFO_STATUS3，FIFO_STATUS4。

可编程 FIFO 阈值可以利用 FTH_[10:0] 位在 FIFO_CTRL1 和 FIFO_CTRL2 中进行设置。

通过 INT1_CTRL 寄存器的 INT1_FULL_FLAG、INT1_FTH 和 INT1_FIFO_OVR 位，以及 INT2_CTRL 寄存器的 INT2_FULL_FLAG、INT2_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。

要增加可存储在 FIFO 中的采样数，还可以通过将 FIFO_CTRL4 寄存器的 ONLY_HIGH_DATA 置为 1，只存储（作为第 1 个 FIFO 数据集）加速度计和陀螺仪数据的 8 个最高有效位。

向 FIFO 中写入数据可由加速度计/陀螺仪数据准备就绪来触发；还可由传感器集合 (sensor hub) 数据准备就绪（对应 FUNC_SRC1 寄存器 SENSORHUB_END_OP 位特性）：这种情况下，MASTER_CONFIG 寄存器的 DATA_VALID_SEL_FIFO 位必须置为 1。而且，如果 DATA_VALID_SEL_FIFO 被置为 0，且 FIFO_CTRL2 寄存器的 TIMER_PEDO_FIFO_DRDY 位被置为 1，则每次检测到一步时，数据存储到 FIFO 中。

8.1 FIFO 寄存器

FIFO 缓冲器由以下几项管理：

- 五个控制寄存器（从 FIFO_CTRL1 至 FIFO_CTRL5）；
- 四个状态寄存器（从 FIFO_STATUS1 至 FIFO_STATUS4）；
- 两个数据输出寄存器（FIFO_DATA_OUT_L 和 FIFO_DATA_OUT_H）；
- 一些附加位，用来使能阈值使用（STOP_ON_FTH）和将 FIFO 满、阈值或溢出事件发送至两个中断引脚（位：INT1_FULL_FLAG、INT2_FULL_FLAG、INT1_FTH、INT2_FTH、INT1_FIFO_OVR、INT2_FIFO_OVR）。

8.1.1 FIFO_CTRL1 (06h)

FIFO_CTRL1 寄存器包含 11 位 FIFO 阈值的低位部分。对于完整的阈值配置，还要考虑 FIFO_CTRL2 寄存器的 FTH_[10:8] 位。FIFO 阈值的值是指 16 位格式的数据。

当 FIFO 中存储的字节数大于或等于阈值水平时，FIFO 水印标记（FIFO_STATUS2 寄存器的 WaterM 位）升高。要将 FIFO 深度限制为水印级别，FIFO_CTRL4 寄存器中 STOP_ON_FTH 位必须设置为 1。

表 51. FIFO_CTRL1 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
FTH_7	FTH_6	FTH_5	FTH_4	FTH_3	FTH_2	FTH_1	FTH_0

8.1.2 FIFO_CTRL2 (07h)

表 52. FIFO_CTRL2 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
TIMER_ PEDO_ FIFO_EN	TIMER_ PEDO_ FIFO_ DRDY	0	0	FIFO_ TEMP_ EN	FTH_10	FTH_9	FTH_8

- **TIMER_PEDO_FIFO_EN** 将计数器和时间戳数据作为第 4 FIFO 数据集存储起来。当此位置为 1 时，FIFO 中存储的 6 个字节内容如第 8.8 节 FIFO 中的计数器和时间戳数据中所示。
- **TIMER_PEDO_FIFO_DRDY**。当此位置为 1，并且 MASTER_CONFIG 寄存器的 DATA_VALID_SEL_FIFO 位被置为 0 时，每次计数器检测到新的一步时，所有数据均存储到 FIFO 中。详细信息，请参见第 8.3 节设置 FIFO 触发、FIFO ODR 和抽取因子。
- **FIFO_TEMP_EN** 位能够将温度数据作为第四个 FIFO 数据集存储起来。当此位置为 1 时，FIFO 中存储的 6 个字节内容如第 8.9 节 FIFO 中的温度数据中所示。
- **FTH_[10:8]** 包含了 FIFO 阈值的高位部分。对于完整的阈值配置，还要考虑 FIFO_CTRL1 寄存器的 FTH_[7:0] 位。

8.1.3

FIFO_CTRL3 (08h)

FIFO_CTRL3 寄存器包含了加速度计和陀螺仪 FIFO 抽取因子，用于选择这些传感器的数据是否必须存储到 FIFO 中，以及以什么速率进行存储。

当 DEC_FIFO_GYRO[2:0] 位被置为 000b，第 1 FIFO 数据集（预留给陀螺仪数据）不存储在 FIFO 中。当 DEC_FIFO_XL[2:0] 位被置为 000b 时，第 2 FIFO 数据集（预留给加速度计数据）不存储在 FIFO 中。

表 53. FIFO_CTRL3 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
0	0	DEC_ FIFO_ GYRO2	DEC_ FIFO_ GYRO1	DEC_ FIFO_ GYRO0	DEC_ FIFO_ XL2	DEC_ FIFO_ XL1	DEC_ FIFO_ XL0

表 54. 陀螺仪 FIFO 抽取设置

DEC_FIFO_GYRO [2:0]	配置
000	陀螺仪传感器不在 FIFO 中
001	// 无抽取
010	抽取因子为 2
011	抽取因子为 3
100	抽取因子为 4
101	抽取因子为 8
110	抽取因子为 16
111	抽取因子为 32

表 55. 加速度计 FIFO 抽取设置

DEC_FIFO_XL [2:0]	配置
000	加速度计传感器不在 FIFO 中
001	// 无抽取
010	抽取因子为 2
011	抽取因子为 3
100	抽取因子为 4
101	抽取因子为 8
110	抽取因子为 16
111	抽取因子为 32

8.1.4 FIFO_CTRL4 (09h)

FIFO_CTRL4 寄存器含有抽取因子，用来定义第 3 FIFO 和第 4 FIFO 数据集所关联的数据以多少数据率存储到 FIFO 中。

当 DEC_DS3_FIFO[2:0] 位置为 000b 时，第 3 个 FIFO 数据集不存储在 FIFO 中。当 DEC_DS4_FIFO[2:0] 位被置为 000b 时，第 4 个 FIFO 数据集不存储在 FIFO 中。

FIFO_CTRL4 寄存器还还有位 ONLY_HIGH_DATA，该位允许只将加速度计和陀螺仪数据的高位部分（最高有效字节）存储到 FIFO 中，以增加 FIFO 中加速度计和陀螺仪的最大采样数。关于此功能的更多详细信息，参见第 8.7 节 陀螺仪和加速度计数据的高位部分。

FIFO_CTRL4 寄存器包含了位 STOP_ON_FTH，可将 FIFO 深度限制在水印级别。

表 56. FIFO_CTRL4 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
STOP_ ON_FTH	ONLY_ HIGH _DATA	DEC_ DS4 _FIFO2	DEC_ DS4 _FIFO1	DEC_ DS4 _FIFO0	DEC_ DS3 _FIFO2	DEC_ DS3 _FIFO1	DEC_ DS3 _FIFO0

表 57. 第 3 FIFO 数据集抽取设置

DEC_DS3_FIFO [2:0]	配置
000	第 3 FIFO 数据集不在 FIFO 中
001	// 无抽取
010	抽取因子为 2
011	抽取因子为 3
100	抽取因子为 4
101	抽取因子为 8
110	抽取因子为 16
111	抽取因子为 32

表 58. 第 4 FIFO 数据集抽取设置

DEC_DS4_FIFO [2:0]	配置
000	第 4 FIFO 数据集不在 FIFO 中
001	// 无抽取
010	抽取因子为 2
011	抽取因子为 3
100	抽取因子为 4
101	抽取因子为 8
110	抽取因子为 16
111	抽取因子为 32

8.1.5 FIFO_CTRL5 (0Ah)

FIFO_CTRL5 寄存器包含 FIFO 操作模式位（FIFO_MODE_[2:0]）和 FIFO 输出数据率位（ODR_FIFO_[3:0]）。

FIFO 操作模式如第 8.2 节 FIFO 模式中所示。

当使用内部触发（加速度计/陀螺仪数据准备就绪）时，ODR_FIFO [3:0]位定义数据存储到 FIFO 中的最大数据率。利用 FIFO 抽取因子，数据能够以较低速率存储到 FIFO 中。关于 FIFO 触发和 FIFO ODR 配置的更多信息，参见第 8.3 节 设置 FIFO 触发、FIFO ODR 和抽取因子。

注意：当使用 FIFO 时，CTRL3_C 寄存器的 IF_INC 位必须等于 1。

表 59. FIFO_CTRL5 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
0	ODR_ FIFO_3	ODR_ FIFO_2	ODR_ FIFO_1	ODR_ FIFO_0	FIFO_ MODE_2	FIFO_ MODE_1	FIFO_ MODE_0

表 60. FIFO ODR 选择设置

ODR_FIFO [3:0]	配置
0000	FIFO 禁用
0001	FIFO ODR 被置为 12.5 Hz
0010	FIFO ODR 被置为 26 Hz
0011	FIFO ODR 被置为 52 Hz
0100	FIFO ODR 被置为 104 Hz
0101	FIFO ODR 被置为 208 Hz
0110	FIFO ODR 被置为 416 Hz
0111	FIFO ODR 被置为 833 Hz
1000	FIFO ODR 被置为 1.66 kHz
1001	FIFO ODR 被置为 3.33 kHz
1010	FIFO ODR 被置为 6.66 kHz

表 61. FIFO 模式选择

FIFO_MODE [2:0]	配置
000	Bypass 模式。FIFO 禁用。
001	FIFO 模式。当 FIFO 满时，停止采集数据。
010	Reserved
011	直至触发条件满足前均为 Continue 模式，条件满足后进入 FIFO 模式。
100	直至触发条件满足前均为 Bypass 模式，条件满足后进入 Continue 模式。
101	Reserved
110	Continue 模式。如果 FIFO 已满，则新的采样将会覆盖原有值。
111	Reserved

8.1.6 FIFO_STATUS1 (3Ah)

FIFO_STATUS1 寄存器，连同 FIFO_STATUS2 寄存器一起，可提供 FIFO 中存储的采样数相关信息。每个采样被表示为 16 位数据。

表 62. FIFO_STATUS1 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
DIFF_	DIFF_	DIFF_	DIFF_	DIFF_	DIFF_	DIFF_	DIFF_

b7	b6	b5	b4	b3	b2	b1	b0
FIFO_7	FIFO_6	FIFO_5	FIFO_4	FIFO_3	FIFO_2	FIFO_1	FIFO_0

8.1.7

FIFO_STATUS2 (3Bh)

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，空）。

表 63. FIFO_STATUS2 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
WaterM	OVER_RUN	FIFO_FULL_SMART	FIFO_空	0	DIFF_FIF0_10	DIFF_FIF0_9	DIFF_FIF0_8

- **WaterM** 表示水印状态。当 FIFO 中已存储的字节数大于等于水印水平，则此位被置为高电平（每个采样表示为 16 位数据）。通过将 INT1_CTRL 寄存器的 INT1_FTH 位或 INT2_CTRL 寄存器的 INT2_FTH 位置为 1，可将水印状态驱动至两个中断引脚上。
- 当 FIFO 溢出，至少一个采样已经被覆盖掉以存储新数据时，OVER_RUN 置为高电平。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 中的下一数据集将要使 FIFO 满时，FIFO_FULL_SMART 被置为高电平。通过将 INT1_CTRL 寄存器的 INT1_FULL_FLAG 位或 INT2_CTRL 寄存器的 INT2_FULL_FLAG 位置为 1，可将此信号驱动至两个中断引脚上。
- 当 FIFO 空时，FIFO_EMPTY 置为高电平。
- DIFF_FIF0_[10:8] 包含 FIFO 中存储的未读字（16 位数据）数的高位部分。其低位部分由 FIFO_STATUS1 中的 DIFF_FIF0_[7:0] 位表示。DIFF_FIF0_[10:0] 字段的值对应 FIFO 中的采样数（每个采样表示为 16 位数据）。当发生 FIFO 溢出事件时（OVER_RUN 位置为高电平），DIFF_FIF0_[10:0] 字段的值为 0。

寄存器内容与 FIFO 读写操作同步更新，如表 64. FIFO_STATUS2 特性（一个传感器处于 FIFO 模式，且 STOP_ON_FTH = 0 的情况下）所示。

表 64. FIFO_STATUS2 特性（一个传感器处于 FIFO 模式，且 STOP_ON_FTH = 0 的情况下）

FIFO_OVER_RUN	FIFO_FULL	FIFO_EMPTY	DIFF_FIF0_[10:0]	FIFO 采样数	FIFO 触发时序
0	0	1	0	0	t0
0	0	0	3	3	t1
0	0	0	6	6	t2
...	...	...	...	...	...
0	0	0	2044	2044	t_full - 2
0	1	0	2047	2047	t_full - 1
1	1	0	0	2048 (旧的采样被覆盖)	t_full

8.1.8

FIFO_STATUS3 (3Ch)

FIFO_STATUS3 寄存器，连同 FIFO_STATUS4 寄存器一起，指明了下次读取时读取哪个传感器哪个轴上的数据。关于如何从 FIFO 中恢复数据，更多信息参见第 8.5 节 FIFO 标识。

表 65. FIFO_STATUS3 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
FIFO_PATTERN	FIFO_PATTERN	FIFO_PATTERN	FIFO_PATTERN	FIFO_PATTERN	FIFO_PATTERN	FIFO_PATTERN	FIFO_PATTERN

b7	b6	b5	b4	b3	b2	b1	b0
_7	_6	_5	_4	_3	_2	_1	_0

8.1.9 FIFO_STATUS4 (3Dh)

FIFO_STATUS4 寄存器，连同 FIFO_STATUS3 寄存器一起，指明了下次读取时读取哪个传感器哪个轴上的数据。关于如何从 FIFO 中恢复数据，更多信息参见第 8.5 节 [FIFO 标识](#)。

表 66. FIFO_STATUS4 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	FIFO_ PATTERN _9	FIFO_ PATTERN _8

8.1.10 FIFO_DATA_OUT_L (3Eh)

FIFO_DATA_OUT_L 寄存器为 FIFO 输出数据的最低有效字节。最高有效字节存储在 FIFO_DATA_OUT_H 寄存器中。关于如何从 FIFO 中恢复数据，更多信息参见第 8.4 节 [从 FIFO 恢复数据](#)。

表 67. FIFO_DATA_OUT_L 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
DATA_ OUT_FIFO _L_7	DATA_ OUT_FIFO _L_6	DATA_ OUT_FIFO _L_5	DATA_ OUT_FIFO _L_4	DATA_ OUT_FIFO _L_3	DATA_ OUT_FIFO _L_2	DATA_ OUT_FIFO _L_1	DATA_ OUT_FIFO _L_0

8.1.11 FIFO_DATA_OUT_H (3Fh)

FIFO_DATA_OUT_H 寄存器为 FIFO 输出数据的最高有效字节。最低有效字节存储在 FIFO_DATA_OUT_L 寄存器中。关于如何从 FIFO 中恢复数据，更多信息参见第 8.4 节 [从 FIFO 恢复数据](#)。

表 68. FIFO_DATA_OUT_H 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
DATA_ OUT_FIFO _H_7	DATA_ OUT_FIFO _H_6	DATA_ OUT_FIFO _H_5	DATA_ OUT_FIFO _H_4	DATA_ OUT_FIFO _H_3	DATA_ OUT_FIFO _H_2	DATA_ OUT_FIFO _H_1	DATA_ OUT_FIFO _H_0

8.2 FIFO 模式

通过 FIFO_CTRL5 寄存器的 FIFO_MODE_[2:0] 字段，LSM6DSL FIFO 缓冲器可配置为五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、Continue、Continue-FIFO 和 Bypass-Continue 模式。

注意：当使用 FIFO 时，CTRL3_C 寄存器的 IF_INC 位必须等于 1。

8.2.1 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。当此模式使能，FIFO_STATUS2 寄存器含有值 10h（FIFO 空）。

在不同模式下工作时，要停止和复位 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。

8.2.2 FIFO 模式

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

按照这些步骤进行 FIFO 模式配置（如果加速度计/陀螺仪数据准备就绪被用作 FIFO 触发）：

1. 通过 FIFO_CTRL3 和 FIFO_CTRL4 寄存器中的抽取位来为每个传感器选择抽取因子（详细信息参见第 8.3 节 设置 FIFO 触发、FIFO ODR 和抽取因子）；
2. 通过 FIFO_CTRL5 寄存器中的 ODR_FIFO_[3:0] 位选择 FIFO ODR；
3. 将 FIFO_CTRL5 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。

当下一个要存储的数据集将会使 FIFO 满时，FIFO_STATUS2 寄存器的 FIFO_FULL_SMART 位被置为 1，不再有数据存储到 FIFO 缓冲器中。通过读取 FIFO_DATA_OUT_L 和 FIFO_DATA_OUT_H 寄存器获取 FIFO_STATUS1 和 FIFO_STATUS2 寄存器的 DIFF_FIFO_[10:0] 位所指定的次数，可在 FIFO_FULL_SMART 事件后恢复。

利用 FIFO_STATUS2 寄存器的 WaterM 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 FTH_[10:0]）时恢复。

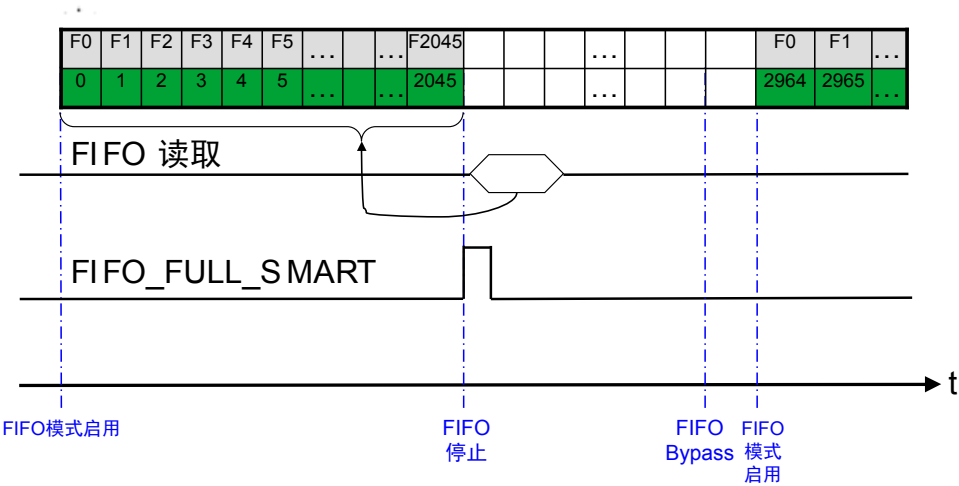
如果 FIFO_CTRL4 寄存器的 STOP_ON_FTH 位置为 1，则 FIFO 空间大小被限制为 FIFO_CTRL1 和 FIFO_CTRL2 寄存器中 FTH_[10:0] 位的值：这种情况下，当下一次 FIFO 写操作时 FIFO 中采样数达到或超过 FTH_[11:0] 值，那么 FIFO_STATUS2 寄存器的 FIFO_FULL_SMART 位会被置为高电平。

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 28. FIFO 模式（STOP_ON_FTH = 0）显示了 FIFO 模式使用的示例。示例中，X-Y-Z 数据（绿色单元表示采样数）只来自一个传感器，存储到 FIFO 中。这些条件下，FIFO 中可存储 2046 个 16 位采样值。



图 28. FIFO 模式 (STOP_ON_FTH = 0)



8.2.3 Continue 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。

按照这些步骤进行 Continue 模式配置（如果加速度计/陀螺仪数据准备就绪被用作 FIFO 触发）：

1. 通过 FIFO_CTRL3 和 FIFO_CTRL4 寄存器中的抽取位来为每个传感器选择抽取因子（详细信息参见第 8.3 节 设置 FIFO 触发、FIFO ODR 和抽取因子）；
2. 通过 FIFO_CTRL5 寄存器中的 ODR_FIFO_[3:0] 位选择 FIFO ODR；
3. 将 FIFO_CTRL5 寄存器中 FIFO_MODE_[2:0] 位置为 110b 来使能 FIFO Continue 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。

当下一个要存储的数据集将会使 FIFO 满时，FIFO_STATUS2 寄存器的 FIFO_FULL_SMART 位被置为 1。

FIFO_STATUS2 寄存器的 OVER_RUN 位表示至少有一个采样被覆盖以存储新数据。

通过读取 FIFO_DATA_OUT_L 和 FIFO_DATA_OUT_H 寄存器获取 FIFO_STATUS1 和 FIFO_STATUS2 寄存器的 DIFF_FIFO_[10:0] 位所指定的次数，可在 FIFO_FULL 事件后恢复。

利用 FIFO_STATUS2 寄存器的 WaterM 位，数据还可在达到阈值（FIFO_CTRL1 和 FIFO_CTRL2 寄存器中的 FTH_[10:0]）时恢复。

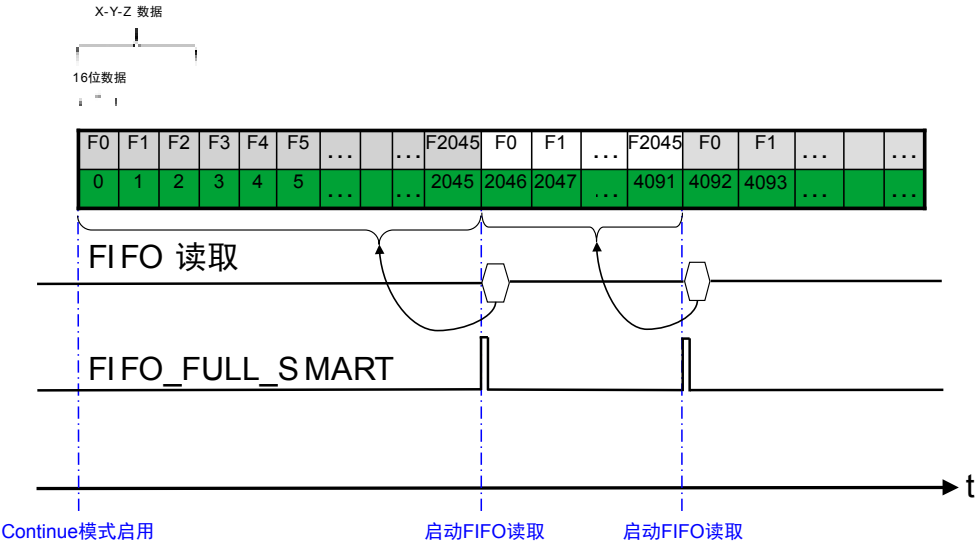
如果 FIFO_CTRL4 寄存器的 STOP_ON_FTH 位置为 1，则 FIFO 空间大小被限制为 FIFO_CTRL1 和 FIFO_CTRL2 寄存器中 FTH_[10:0] 位的值：这种情况下，当下一次 FIFO 写操作时 FIFO 中采样数达到 FTH_[10:0] 值，那么 FIFO_STATUS2 寄存器的 FIFO_FULL_SMART 位会被置为高电平。

要释放 FIFO 空间用于新数据，建议读取速度比 $1 \times \text{ODR}$ 至少快已使能 FIFO 数据数的三倍：这能够避免丢失数据。

图 29. Continue 模式 显示了 Continue 模式使用的示例。示例中，X-Y-Z 数据（绿色单元表示采样数）仅来自一个传感器，存储到 FIFO 中，并且读取速度快于 $1 \times \text{ODR}$ ，因此无数据丢失。这些条件下，所存储的 16 位采样数为 2046。



图 29. Continue 模式



8.2.4 Continue-FIFO 模式

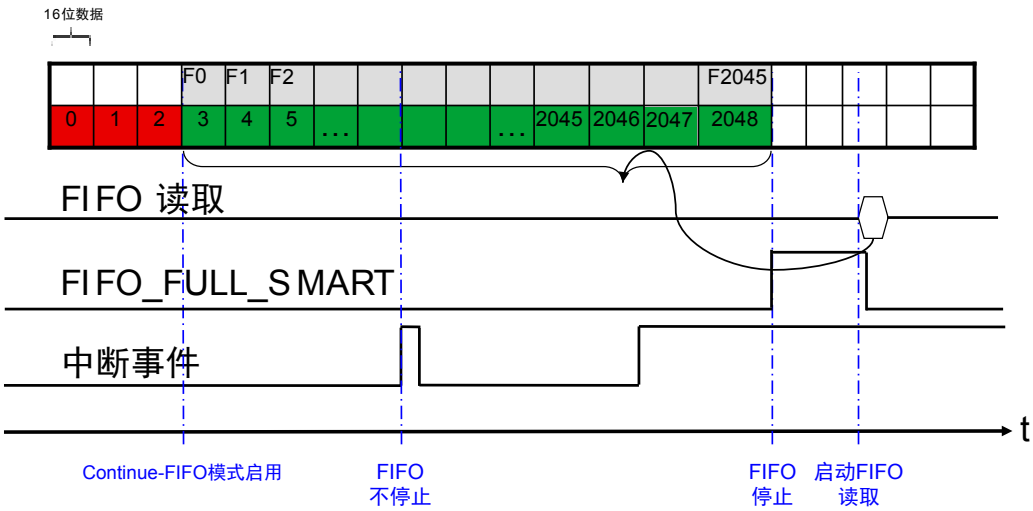
此模式是先前所述的 Continue 和 FIFO 模式的组合。在 Continue-FIFO 模式中，FIFO 缓冲器开始工作于 Continue 模式，当发生事件条件时切换为 FIFO 模式。

事件条件可为以下之一：

- 双击：必须配置事件检测，并且 MD2_CFG 寄存器的 INT2_DOUBLE_TAP 位必须被置为 1；
- 倾斜：必须配置事件检测，并且 MD2_CFG 寄存器的 INT2_TILT 位必须被置为 1；
- 计步运动：必须配置事件检测，并且 INT1_CTRL 寄存器的 INT1_STEP_DETECTOR 位必须被置为 1；
- 单击：必须配置事件检测，并且 MD2_CFG 寄存器的 INT2_SINGLE_TAP 位必须被置为 1；
- 双击：必须配置事件检测，并且 MD2_CFG 寄存器的 INT2_DOUBLE_TAP 位必须被置为 1；
- 自由落体：必须配置事件检测，并且 MD2_CFG 寄存器的 INT2_FF 位必须被置为 1；
- 唤醒：必须配置事件检测，并且 MD2_CFG 寄存器的 INT2_WU 位必须被置为 1；
- 6D：必须配置事件检测，并且 MD2_CFG 寄存器的 INT2_6D 位必须被置为 1。

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

图 30. Continue-FIFO 模式



按照这些步骤进行 Continue-FIFO 模式配置（如果加速度计/陀螺仪数据准备就绪被用作 FIFO 触发）：

1. 配置先前所述的其中一个事件；
2. 通过 FIFO_CTRL3 和 FIFO_CTRL4 寄存器中的抽取位来为每个传感器选择抽取因子（详细信息参见第 8.3 节 设置 FIFO 触发、FIFO ODR 和抽取因子）；
3. 通过 FIFO_CTRL5 寄存器中的 ODR_FIFO_[3:0] 位选择 FIFO ODR；
4. 将 FIFO_CTRL5 寄存器中 FIFO_MODE_[2:0] 位置为 011b 来使能 Continue-FIFO 模式。

在 Continue-FIFO 模式中，FIFO 缓冲器连续填充；当下一次存储的数据集会填满 FIFO 时，FIFO_FULL_SMART 位置为高电平。

如果 FIFO_CTRL4 寄存器的 STOP_ON_FTH 位置为 1，则 FIFO 空间大小被限制为 FIFO_CTRL1 和 FIFO_CTRL2 寄存器中 FTH_[10:0] 位的值：这种情况下，当下一次 FIFO 写操作时 FIFO 中采样数达到或超过 FTH_[11:0] 值，那么 FIFO_STATUS2 寄存器的 FIFO_FULL_SMART 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（FIFO_FULL_SMART = 1），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（初始瞬态），则继续填充直至填满（FIFO_FULL_SMART = 1），然后如果触发仍然存在，则停止采集数据。

Continue-FIFO 可用来分析产生中断的采样历史；当触发 FIFO 模式，且 FIFO 缓冲器已满、停止填充时，标准操作是读取 FIFO 内容。

8.2.5 Bypass-Continue 模式

此模式是先前所述的 Bypass 和 Continue 模式的组合。在 Bypass-Continue 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 Continue 模式。

事件条件可为以下之一：

- 双击：必须配置事件检测，并且 MD2_CFG 寄存器的 INT2_DOUBLE_TAP 位必须被置为 1；
- 倾斜：必须配置事件检测，并且 MD2_CFG 寄存器的 INT2_TILT 位必须被置为 1；
- 计步运动：必须配置事件检测，并且 INT1_CTRL 寄存器的 INT1_STEP_DETECTOR 位必须被置为 1；
- 单击：必须配置事件检测，并且 MD2_CFG 寄存器的 INT2_SINGLE_TAP 位必须被置为 1；
- 双击：必须配置事件检测，并且 MD2_CFG 寄存器的 INT2_DOUBLE_TAP 位必须被置为 1；
- 自由落体：必须配置事件检测，并且 MD2_CFG 寄存器的 INT2_FF 位必须被置为 1；
- 唤醒：必须配置事件检测，并且 MD2_CFG 寄存器的 INT2_WU 位必须被置为 1；
- 6D：必须配置事件检测，并且 MD2_CFG 寄存器的 INT2_6D 位必须被置为 1。

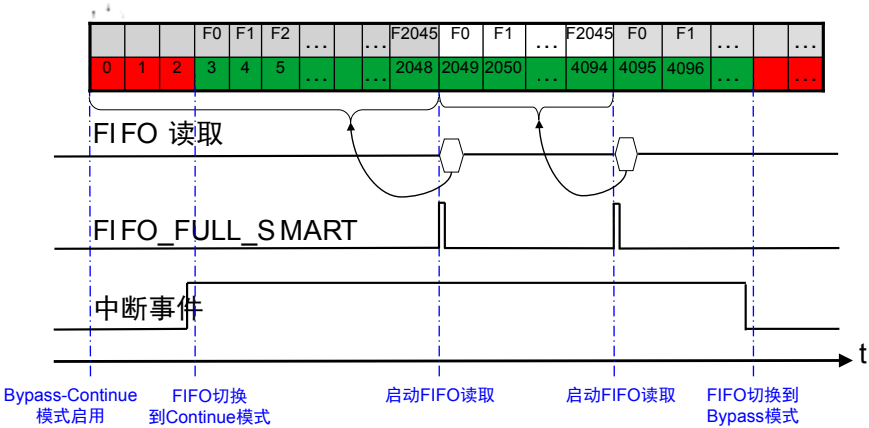
Bypass-Continue 模式对中断信号的边沿感应：在第一次中断事件时，FIFO 从 Bypass 模式切换到 Continue 模式，并维持该模式，直到 Bypass 模式被置位。

按照这些步骤进行 Bypass-Continue 模式配置（如果加速度计/陀螺仪数据准备就绪被用作 FIFO 触发）：

1. 配置先前所述的其中一个事件；
2. 通过 FIFO_CTRL3 和 FIFO_CTRL4 寄存器中的抽取位来为每个传感器选择抽取因子（详细信息参见第 8.3 节 设置 FIFO 触发、FIFO ODR 和抽取因子）；
3. 通过 FIFO_CTRL5 寄存器中的 ODR_FIFO_[3:0] 位选择 FIFO ODR。
4. 将 FIFO_CTRL5 寄存器中 FIFO_MODE_[2:0] 位置为 100b 来使能 Bypass-Continue 模式。



图 31. Bypass-Continue 模式





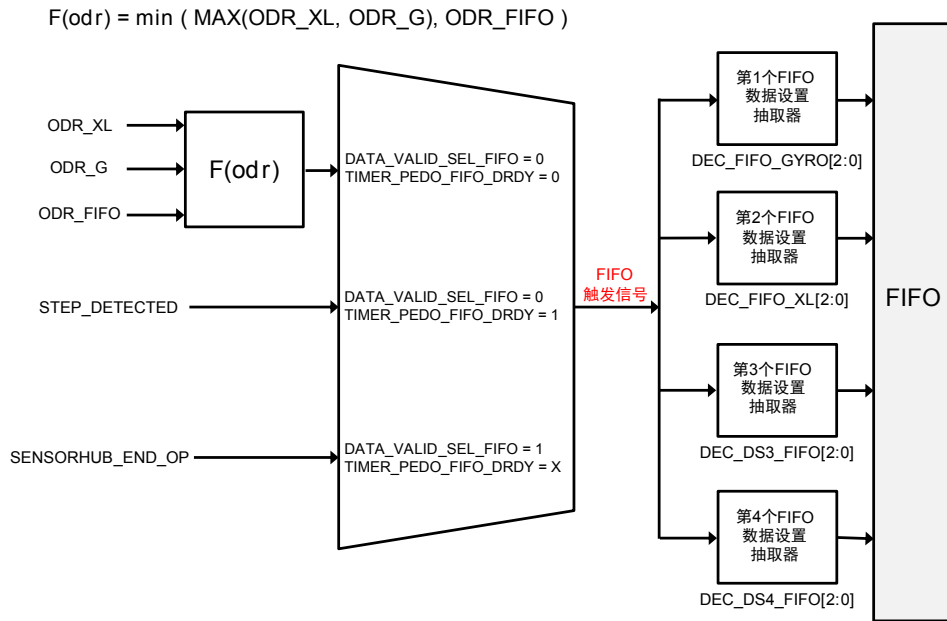
当出现触发条件且缓冲器切换至 **Continue** 模式时，FIFO 缓冲器继续填充。当下一个要存储的数据集将会使 FIFO 满时，FIFO_FULL_SMART 位被置为高电平。

Bypass-Continue 可用来在产生所配置中断时启动采集。

8.3 设置 FIFO 触发、FIFO ODR 和抽取因子

向 FIFO 中写入数据可配置为由三种不同的来源触发。

图 32. FIFO 触发信号选择



如图 32. FIFO 触发信号选择中所述，MASTER_CONFIG 寄存器的 DATA_VALID_SEL_FIFO 位和 FIFO_CTRL2 寄存器的 TIMER_PEDO_FIFO_DRDY 位可用来实现此目的：

- 如果 DATA_VALID_SEL_FIFO 位和 TIMER_PEDO_FIFO_DRDY 位均被置为 0，则向 FIFO 中写入数据由加速度计/陀螺仪数据准备就绪来触发。FIFO_CTRL5 的 ODR_FIFO_[3:0] 位定义了向 FIFO 中存储数据的最大数据率；后者受限于加速度计 ODR（由 CTRL1_XL 寄存器的 ODR_XL[3:0] 位定义）和陀螺仪 ODR（由 CTRL2_G 寄存器的 ODR_G[3:0] 位定义）之间的最大值；
- 如果 DATA_VALID_SEL_FIFO 位被置为 0，且 TIMER_PEDO_FIFO_DRDY 位被置为 1，则向 FIFO 中写入数据由计步来触发（对应于 STEP_DETECTED 寄存器 STEP_DETECTED 位的特性）：每次检测到一步时，数据存储到 FIFO 中；
- 如果 DATA_VALID_SEL_FIFO 位被置为 1，则向 FIFO 中写入数据由传感器集合（sensor hub）来触发（对应于 FUNC_SRC1 寄存器 SENSORHUB_END_OP 位的特性），而无论 TIMER_PEDO_FIFO_DRDY 位是何配置：当传感器集合（sensor hub）程序完成时，数据存储到 FIFO 中。

使用 FIFO 抽取因子，数据能够以低于 FIFO 触发信号速率的速率存储到 FIFO 中。可以配置四种抽取因子，每个 FIFO 数据集一个：

- FIFO_CTRL3 寄存器的 DEC_FIFO_G[2:0] 位定义了陀螺仪数据（与第 1 个 FIFO 数据集相关）是否存储在 FIFO 中及其速率；
- FIFO_CTRL3 寄存器的 DEC_FIFO_XL[2:0] 位定义了加速度计数据（与第 2 个 FIFO 数据集相关）是否存储在 FIFO 中及其速率；
- FIFO_CTRL4 寄存器的 DEC_DS3_FIFO[2:0] 位定义了加速度计数据（与第 3 个 FIFO 数据集相关）是否存储在 FIFO 中及其速率；
- FIFO_CTRL4 寄存器的 DEC_DS4_FIFO[2:0] 位定义了加速度计数据（与第 4 个 FIFO 数据集相关）是否存储在 FIFO 中及其速率。

8.3.1 使用 FIFO 时改变 ODR 或 FIFO 配置步骤

当必须改变加速度计/陀螺仪 ODR 或 FIFO 配置时，使用以下步骤：

1. 读取 FIFO 中存储的所有数据，直至将其清空（详细信息见第 8.4 节 从 FIFO 恢复数据）；



2. 将 FIFO 置为 Bypass 模式（将 FIFO_CTRL5 寄存器的 FIFO_MODE 位置为 000b）；
3. 分别通过 CTRL1_XL 寄存器的 ODR_XL 位和 CTRL2_G 寄存器的 ODR_G 位，为加速度计和陀螺仪设置目标 ODR；
4. 通过 FIFO_CTRL5 寄存器的 ODR_FIFO 位，为 FIFO 设置 ODR；
5. 如下在 FIFO_CTRL3 寄存器的 DEC_FIFO_G[2:0] 位中设置陀螺仪抽取因子，在 FIFO_CTRL3 寄存器的 DEC_FIFO_XL[2:0] 位中设置加速度计抽取因子（参见表 54. 陀螺仪 FIFO 抽取设置和表 55. 加速度计 FIFO 抽取设置，得到 FIFO_CTRL3 的 DEC_FIFO_G[2:0] 位和 DEC_FIFO_XL[2:0] 位中要设置的值）。
6. 设置所需的 FIFO 工作模式（详细信息见第 8.3 节 设置 FIFO 触发、FIFO ODR 和抽取因子）。

8.4 从 FIFO 恢复数据

注意：当数据存储到 FIFO 中时，不能改变配置，以便正确恢复数据。

当 FIFO 使能且其模式不是 Bypass 模式时，读取 FIFO 输出寄存器（FIFO_DATA_OUT_L 和 FIFO_DATA_OUT_H）会返回原 FIFO 采样集。无论何时读取这些数据，其内容均会移动到 SPI/I²C 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（16 位数据）。此信息在 DIFF_FIFO_[10:0] 位中。
2. 读取 FIFO_STATUS3 和 FIFO_STATUS4 寄存器。FIFO_PATTERN_[9:0] 位可以识别哪个传感器以及哪些字节被读取（更多详细信息见第 8.5 节 FIFO 标识）。
3. 读取 FIFO_DATA_OUT_L 和 FIFO_DATA_OUT_H 寄存器来恢复 FIFO 中最旧的采样（16 位格式）。它们分别是该最旧采样的低位部分和高位部分。

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_STATUS2 寄存器的 FIFO_EMPTY 位置为高电平），可以恢复全部 FIFO 内容。

注：一旦 FIFO 为空，则数据不能从 FIFO_DATA_OUT_L 和 FIFO_DATA_OUT_H 寄存器中取回。

要释放 FIFO 空间用于新数据，建议读取速度比 $1 \times \text{ODR}$ 至少快已使能 FIFO 数据集的三倍：这能够避免丢失数据。

对 FIFO 输出寄存器 FIFO_DATA_OUT_L 和 FIFO_DATA_OUT_H 应用多次读取操作时，环形函数（详细信息见第 4.7 节 环形功能）自动使能。

8.5 FIFO 标识

存储在 FIFO 中的数据没有任何标签，以便能够最大化存储采样数。要识别将要读取哪些数据以及哪个 FIFO 数据集，需要检查 FIFO_STATUS3 和 FIFO_STATUS4 寄存器的 FIFO_PATTERN_[9:0] 位的内容。

数据以特定模式写入 FIFO（例如，GyroX, GyroY, GyroZ, AccX, AccY, AccZ）。此模式根据四个 FIFO 数据集所分配的 ODR 和抽取因子而改变。FIFO_PATTERN_[9:0] 位含有一个数字，此数字从 0 至其模式上次采样的索引，然后对所有 FIFO 内容重复该模式。

FIFO 缓冲器中存储的第一个数据序列包含所有使能 FIFO 数据集（从第一个至第四个）中的数据。然后，根据每个 FIFO 数据集所设置的抽取因子，重复数据。

下节中的示例说明了如何使用 FIFO_PATTERN_[9:0] 位中包含的信息。

8.5.1 示例 1

假设 FIFO 正在存储来自陀螺仪和加速度计（二者具有相同的 ODR）的数据：

- 陀螺仪 ODR = 104 Hz，加速度计 ODR = 104 Hz。

如果使用了内部触发（加速度计/陀螺仪数据准备就绪），建议将 FIFO_CTRL5 寄存器的 ODR_FIFO_[3:0] 位置为 0100b，以便将 FIFO 触发 ODR 设置为 104 Hz。

FIFO_CTRL3 寄存器的 DEC_FIFO_GYRO[2:0] 和 DEC_FIFO_XL[2:0] 字段必须置为 001b（无抽取）。

每 6 次采样后重复以下数据模式（每个采样表示为 16 位数据）：

- Gx Gy Gz XLx XLy XLz（陀螺仪和加速度计数据）

FIFO_PATTERN_[9:0] 位会含有一个从 0 至 5 的数字，如表 69. 示例 1: FIFO_PATTERN_[9:0] 位和下一个读数所示。

表 69. 示例 1: FIFO_PATTERN_[9:0]位和下一个读数

时间	FIFO_PATTERN_[9:0]	下次读取 FIFO 输出寄存器
t0	0	Gx
t0	1	Gy
t0	2	Gz
t0	3	XLx
t0	4	XLy
t0	5	XLz

8.5.2

示例 2

假设 FIFO 正在存储来自陀螺仪和加速度计（二者 ODR 不同）的数据：

- 陀螺仪 ODR = 208 Hz，加速度计 ODR = 104 Hz。

如果使用了内部触发（加速度计/陀螺仪数据准备就绪），建议将 FIFO_CTRL5 寄存器的 ODR_FIFO_[3:0]位置为 0101b，以便将 FIFO 触发 ODR 设置为 208 Hz。

FIFO_CTRL3 寄存器的 DEC_FIFO_GYRO[2:0]字段必须置为 001b（陀螺仪数据不使用抽取），并且 DEC_FIFO_XL[2:0]字段必须置为 010b（加速度计数据所用抽取因子为 2）。

由于陀螺仪 ODR 是加速度计 ODR 的两倍，因此每 9 次采样后重复以下数据模式（每个采样表示为 16 位数据）：

- Gx Gy Gz XLx XLy XLz Gx Gy Gz

FIFO_PATTERN_[9:0]位会含有一个从 0 至 8 的数字，如表 70. 示例 2: FIFO_PATTERN_[9:0]位和下一个读数所示。

表 70. 示例 2: FIFO_PATTERN_[9:0]位和下一个读数

时间	FIFO_PATTERN_[9:0]	下次读取 FIFO 输出寄存器
t0	0	Gx
t0	1	Gy
t0	2	Gz
t0	3	XLx
t0	4	XLy
t0	5	XLz
t1	6	Gx
t1	7	Gy
t1	8	Gz

8.5.3

示例 3

假设 FIFO 正在存储来自陀螺仪、加速度计和磁力计（它们具有不同的 ODR）的数据：

- 陀螺仪 ODR = 104 Hz，加速度计 ODR = 208 Hz，磁力计 ODR = 52 Hz。

如果使用了内部触发（加速度计/陀螺仪数据准备就绪），建议将 FIFO_CTRL5 寄存器的 ODR_FIFO_[3:0]位置为 0101b，以便将 FIFO 触发 ODR 设置为 208 Hz。

FIFO_CTRL3 寄存器的 DEC_FIFO_GYRO[2:0]字段必须置为 010b（陀螺仪数据所用抽取因子为 2），并且 DEC_FIFO_XL[2:0]字段必须置为 001b（加速度计数据不使用抽取）。假设磁力计关联到第 3 FIFO 数据集，FIFO_CTRL4 寄存器的 DEC_DS3_FIFO[2:0]字段必须置为 100b（磁力计数据所用抽取因子为 4）。

每 21 个采样后，重复以下数据模式：

- Gx Gy Gz XLx XLy XLz Mx My Mz（陀螺仪、加速度计、磁力计数据 - 9 个采样）
- XLx XLy XLz（加速度计数据 - 3 个采样）
- Gx Gy Gz XLx XLy XLz（陀螺仪和加速度计数据 - 6 个采样）
- XLx XLy XLz（加速度计数据 - 3 个采样）

FIFO_PATTERN_[9:0]位会含有一个从 0 至 20 的数字，如表 71. 示例 3: FIFO_PATTERN_[9:0]位和下一个读数所示。

表 71. 示例 3: FIFO_PATTERN_[9:0]位和下一个读数

时间	FIFO_PATTERN_[9:0]	下次读取 FIFO 输出寄存器
t0	0	Gx
t0	1	Gy
t0	2	Gz
t0	3	XLx
t0	4	XLy
t0	5	XLz
t0	6	Mx
t0	7	My
t0	8	Mz
t1	9	XLx
t1	10	XLy
t1	11	XLz
t2	12	Gx
t2	13	Gy
t2	14	Gz
t2	15	XLx
t2	16	XLy
t2	17	XLz
t3	18	XLx
t3	19	XLy
t3	20	XLz

8.6 FIFO 阈值

FIFO 阈值是 LSM6DSL FIFO 的功能，可用于检查 FIFO 中的采样数何时达到定义的阈值水平。

FIFO_CTRL1 和 FIFO_CTRL2 寄存器的 FTH_[10:0] 中有阈值水平。FTH_[10:0] 字段的精度为两字节（1 LSB = 2 字节，每个采样表示为 16 位数据）。因此，用户能够在 0 至 2047 之间选择所需的值。

FIFO_STATUS2 寄存器的位 WaterM 表示水印状态。如果 FIFO 中采样数达到或超出了水印水平，则此位被置为高电平（每个采样表示为 16 位数据）。

通过将 FIFO_CTRL4 寄存器的 STOP_ON_FTH 位置为 1，FIFO 空间大小可由阈值水平来限制。



图 33. FIFO 阈值 (STOP_ON_FTH=0)

FTH_[10:0] = 21
STOP_ON_FTH = 0

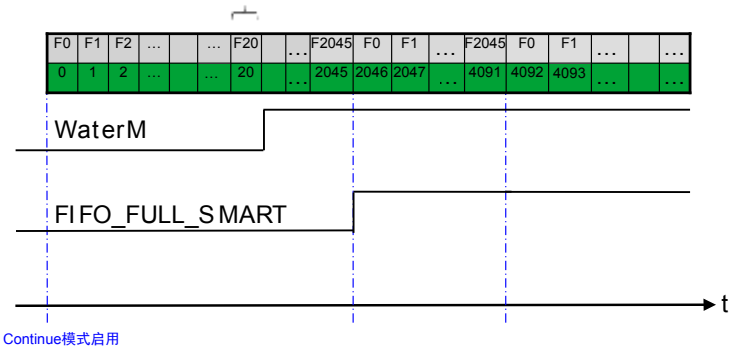


图 33. FIFO 阈值 ($STOP_ON_FTH=0$) 显示了只存储加速度计 (或陀螺仪) 数据时, FIFO 阈值水平使用的示例。FIFO_CTRL4 寄存器中 STOP_ON_FTH 位置为 0。利用 FTH_[10:0] 位, 阈值水平设置为 21。达到 21 后 (FIFO 中有 21 个采样), FIFO_STATUS2 寄存器的 WaterM 位升高。因此, STOP_ON_FTH 位置为 0, FIFO 不会在第 21 个采样时停止, 而是会继续存储数据, 直至 FIFO_FULL_SMART 标志被置为高电平。



图 34. FIFO 模式下的 FIFO 阈值 (STOP_ON_FTH=1)

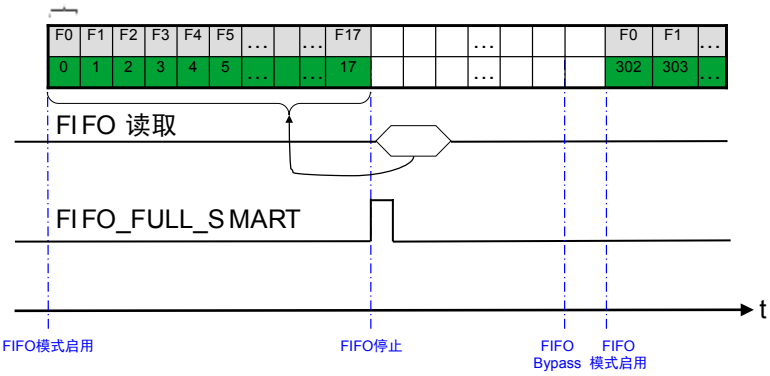


图 34. FIFO 模式下的 FIFO 阈值 (STOP_ON_FTH=1) 显示了 FIFO 模式下 FIFO 阈值使用的示例，其中 FIFO_CTRL4 寄存器中 STOP_ON_FTH 位被置为 1；本例中只存储了加速度计（或陀螺仪）数据。利用 FTH_[10:0] 位，阈值水平设置为 21，并定义当前 FIFO 空间大小。FIFO 模式下，数据存储在 FIFO 缓冲器中，直至 FIFO_FULL_SMART 信号升高；当 FIFO 中存储的下一个数据会使得 FIFO 满时，FIFO_STATUS2 寄存器的 FIFO_FULL_SMART 位升高，因此本例中，第一次有 18 个数据（每个 16 位）存储到 FIFO 中后，该位升高。因为永远不会达到 FTH 阈值，所以 FIFO_STATUS2 寄存器的 WaterM 位不会变为 1（FIFO 满后，数据不再存储到 FIFO 中）。



图 35. Continue 模式下的 FIFO 阈值 (STOP_ON_FTH=1)

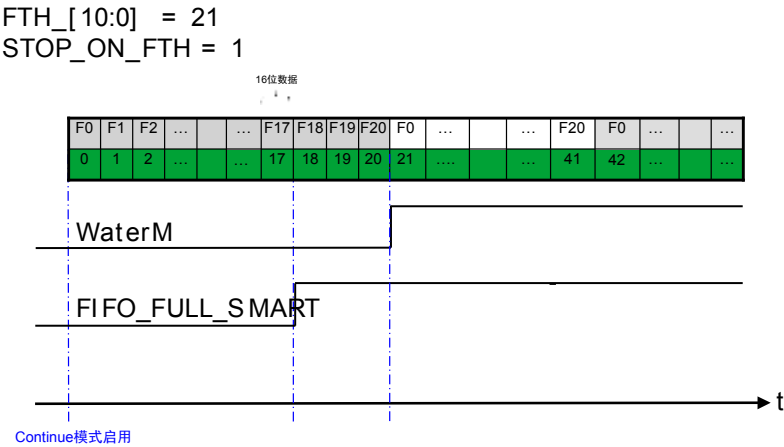


图 35. *Continue* 模式下的 FIFO 阈值 ($STOP_ON_FTH=1$) 显示了 *Continue* 模式下 FIFO 阈值使用的示例，其中 FIFO_CTRL4 寄存器中 STOP_ON_FTH 位被置为 1；本例中只存储了加速度计（或陀螺仪）数据。利用 FTH_[10:0] 位，阈值水平设置为 21。当 FIFO 中存储的下一个数据会使得 FIFO 满时，FIFO_STATUS2 寄存器的 FIFO_FULL_SMART 位升高，因此本例中，第一次有 18 个数据（每个 16 位）存储到 FIFO 中后，该位升高。达到 21 后（FIFO 中有 21 个采样），FIFO_STATUS2 寄存器的 WaterM 位升高。

8.7 陀螺仪和加速度计数据的高位部分

只存储陀螺仪和加速度计数据的高位部分（8 位），可以增加 FIFO 中存储的采样数。此功能对其他（外部）传感器无效。

要启用此功能，FIFO_CTRL4 寄存器中位 ONLY_HIGH_DATA 必须置为 1。陀螺仪和加速度计数据将以相同的 ODR 写入 FIFO，顺序如表 72. FIFO 中陀螺仪和加速度计数据的高位部分所示。

表 72. FIFO 中陀螺仪和加速度计数据的高位部分

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Accel_X_H	Gyro_X_H	Accel_Y_H	Gyro_Y_H	Accel_Z_H	Gyro_Z_H

当此功能使能时，包含陀螺仪和加速度计数据的高位部分（8 位）的 6 个字节关联到第 1 个 FIFO 数据集，而第 2 个 FIFO 数据集不使用。

FIFO_CTRL3 寄存器的 DEC_FIFO_G[2:0] 字段必须置为非 000b 的值（FIFO 中存储的第 1 个 FIFO 数据集）。

FIFO_CTRL3 寄存器的 DEC_FIFO_XL[2:0] 字段必须置为 000b（FIFO 中存储的第 2 个 FIFO 数据集）。

8.8 FIFO 中的计步器和时间戳数据

可以将时间戳和计步器数据存储到 FIFO 中。这些数据按 6 字节数据的格式存储为第 4 FIFO 数据集，如表 73. FIFO 中时间戳和计步器数据所示。

- 3 个字节用于时间戳；
- 1 字节不用；
- 2 个字节用于步数。

表 73. FIFO 中时间戳和计步器数据

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
TIMESTAMP	TIMESTAMP	-	TIMESTAMP	STEPS	STEPS
[15:8]	[23:16]		[7:0]	[7:0]	[15:8]

要使用此功能，FIFO_CTRL2 寄存器中位 TIMER_PEDO_FIFO_EN 必须置为 1。

当此功能使能时，包含时间戳和计步器数据的 6 个字节关联到第 4 FIFO 数据集：FIFO_CTRL4 寄存器的 DEC_DS4_FIFO[2:0] 字段必须用来定义抽取因子。

当此功能使能，且 MASTER_CONFIG 寄存器的 DATA_VALID_SEL_FIFO 位置为 0，则数据能够以两种方式存入 FIFO 中，取决于 FIFO_CTRL2 中 TIMER_PEDO_FIFO_DRDY 位的配置：

- 当 TIMER_PEDO_FIFO_DRDY 位置为 0 时，数据以 FIFO_CTRL5 寄存器中设置的 ODR_FIFO 速率写入 FIFO 中。
- 当 TIMER_PEDO_FIFO_DRDY 位置为 1 时，每次检测到新的一步时数据存储到 FIFO 中。

按照以下步骤，利用内部触发（加速度计/陀螺仪数据准备就绪）或“检测到步数”的方式，将时间戳和计步器数据存储到 FIFO 中：

1. 启动加速度计；
2. 使能时间戳和计步器（参见第 6.1 节 计步功能：步伐检测和步数计算和第 6.5 节 时间戳）；
3. 通过 FIFO_CTRL4 寄存器的 DEC_DS4_FIFO[2:0] 位选择第 4 个 FIFO 数据集的抽取因子；
4. 将 FIFO_CTRL2 寄存器中的 TIMER_PEDO_FIFO_EN 位置 1；
5. 配置 FIFO_CTRL2 寄存器中的位 TIMER_PEDO_FIFO_DRDY，以选择将数据存储到 FIFO 中的方式（内部触发或每检测到一步）；
6. 如果使用了内部触发，则通过 FIFO_CTRL5 寄存器中的 ODR_FIFO_[3:0] 位选择 FIFO ODR。如果使用“检测到一步”触发，则无需设置 ODR_FIFO_[3:0] 位；
7. 利用 FIFO_CTRL5 寄存器的 FIFO_MODE_[2:0] 字段，配置 FIFO 工作模式。

8.9 FIFO 中的温度数据

可以只将温度数据作为第 4 个 FIFO 数据集存储。

要使能此功能：

- FIFO_CTRL2 寄存器的位 TIMER_PEDO_FIFO_EN 必须置为 0；
- FIFO_CTRL2 寄存器的位 FIFO_TEMP_EN 必须置为 1。

温度采样（16 位）以 6 字节数据格式存储到 FIFO 中，如表 74. FIFO 中的温度数据中所示。

表 74. FIFO 中的温度数据

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
-	-	TEMP [7:0]	TEMP [15:8]	-	-

按照以下步骤，利用内部触发（加速度计/陀螺仪数据准备就绪），将 16 位温度数据存储到 FIFO 中：

1. 开启加速度计或陀螺仪；
2. 通过 FIFO_CTRL4 寄存器的 DEC_DS4_FIFO[2:0] 位选择第 4 个 FIFO 数据集的抽取因子（非 000b）；
3. 将 FIFO_CTRL2 寄存器的 FIFO_TEMP_EN 位置为 1，并将 FIFO_CTRL2 寄存器的位 TIMER_PEDO_FIFO_EN 置为 0；
4. 通过 FIFO_CTRL5 寄存器中的 ODR_FIFO_[3:0] 位选择 FIFO ODR；
5. 利用 FIFO_CTRL5 寄存器的 FIFO_MODE_[2:0] 字段，配置 FIFO 工作模式。

9 温度传感器

LSM6DSL 具有内部温度传感器，适用于环境温度测量。

如果加速度计和陀螺仪传感器均处于下电模式，则温度传感器关闭。

温度传感器的最大输出数据率为 **52 Hz**，其值取决于加速度计和陀螺仪传感器如何配置：

- 如果陀螺仪处于下电模式：
 - 如果加速度计 ODR 等于 **12.5 Hz** 低功耗模式，则温度数据速率等于 **12.5 Hz**；
 - 如果加速度计配置为 **26 Hz** 低功耗模式，则温度数据速率等于 **26 Hz**；
 - 对于所有其他的加速度计配置，温度数据速率均等于 **52 Hz**。
- 如果陀螺仪不是处于下电模式，则无论加速度计和陀螺仪配置是什么，温度数据速率均为 **52 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。

温度数据由 OUT_TEMP_H 和 OUT_TEMP_L 寄存器联合给出，以二进制补码的格式表示为一个 16 位的数字，其灵敏度为 +256 LSB/°C。输出零值对应于 25 °C。

将 CTRL3_C 寄存器的 BLE 位置为 1，LSM6DSL 可支持温度输出数据寄存器低位部分和高位部分的内容进行交换（即，OUT_TEMP_H 和 OUT_TEMP_L）。

温度传感器数据也可以采用可配置的抽取因子而存储在 FIFO 中（详细信息见第 8.9 节 FIFO 中的温度数据）。

9.1 温度数据计算示例

表 75. 输出数据寄存器内容 vs. 温度 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 75. 输出数据寄存器内容 vs. 温度

温度值	BLE = 0		BLE = 1	
	寄存器地址			
	OUT_TEMP_H (21h)	OUT_TEMP_L (20h)	OUT_TEMP_H (21h)	OUT_TEMP_L (20h)
0°C	E7h	00h	00h	E7
25°C	00h	00h	00h	00h
50°C	19h	00h	00h	19h

10 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。

10.1 加速度计自检

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。

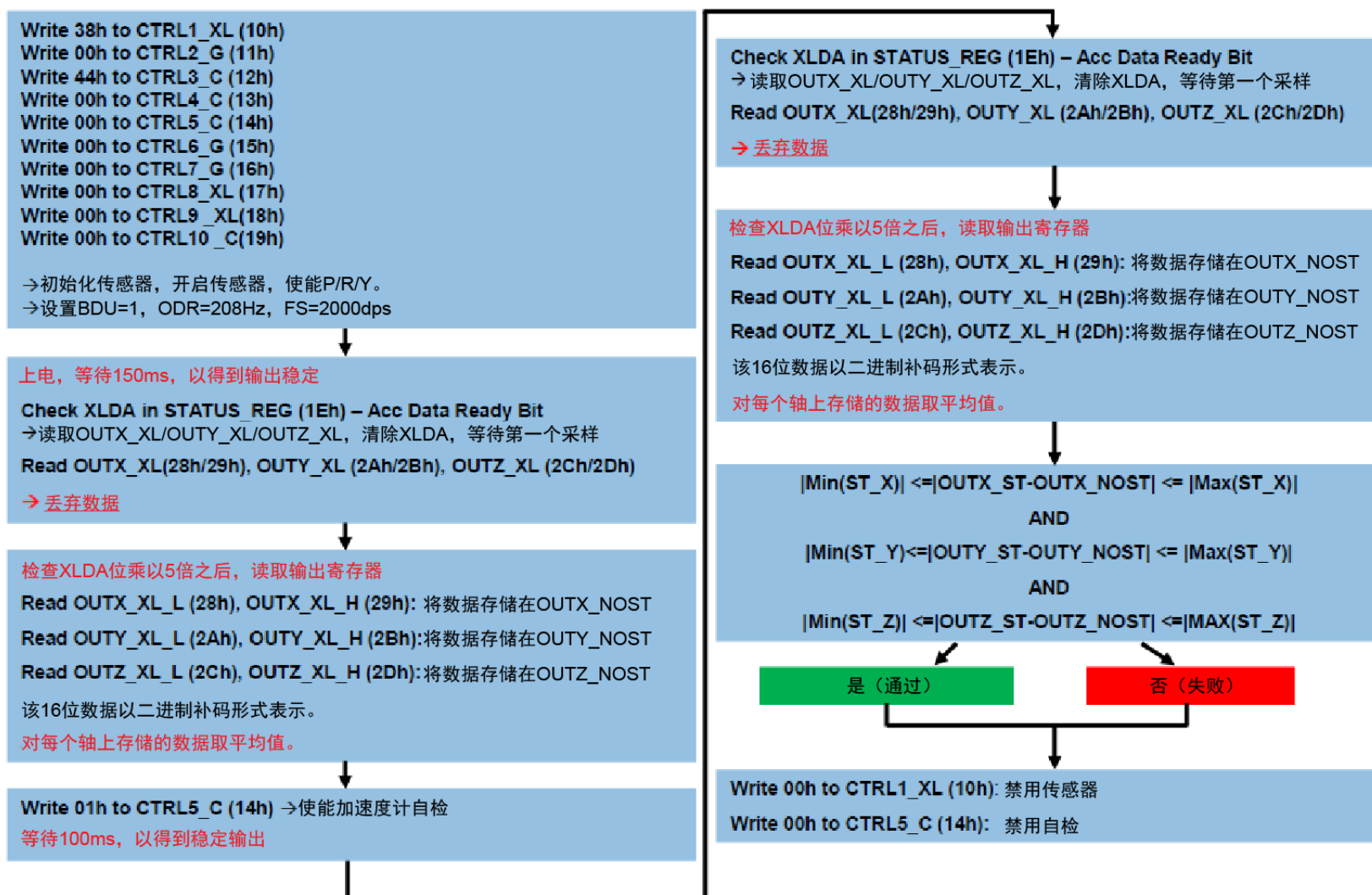
当 **CTRL5_C** 寄存器的 **ST[1:0]_XL** 位被设定为 **00b** 时，加速度计自检功能关闭；当 **ST[1:0]_XL** 位被置为 **01b**（正符号自检）或 **10b**（负符号自检）时，该功能使能。

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 36. 加速度计自检步骤中所示。

图 36. 加速度计自检步骤

注：自检过程中请保持器件稳定。



根据保密协议，不可复制

10.2 陀螺仪自检

陀螺仪自检能够支持对陀螺仪传感器的机械和电气部分进行测试：当它激活时，驱动力施加到传感器上，模拟一定的 Coriolis 力，并利用该静电测试力移动质量块。这种情况下，传感器输出会显示出一个输出变化。

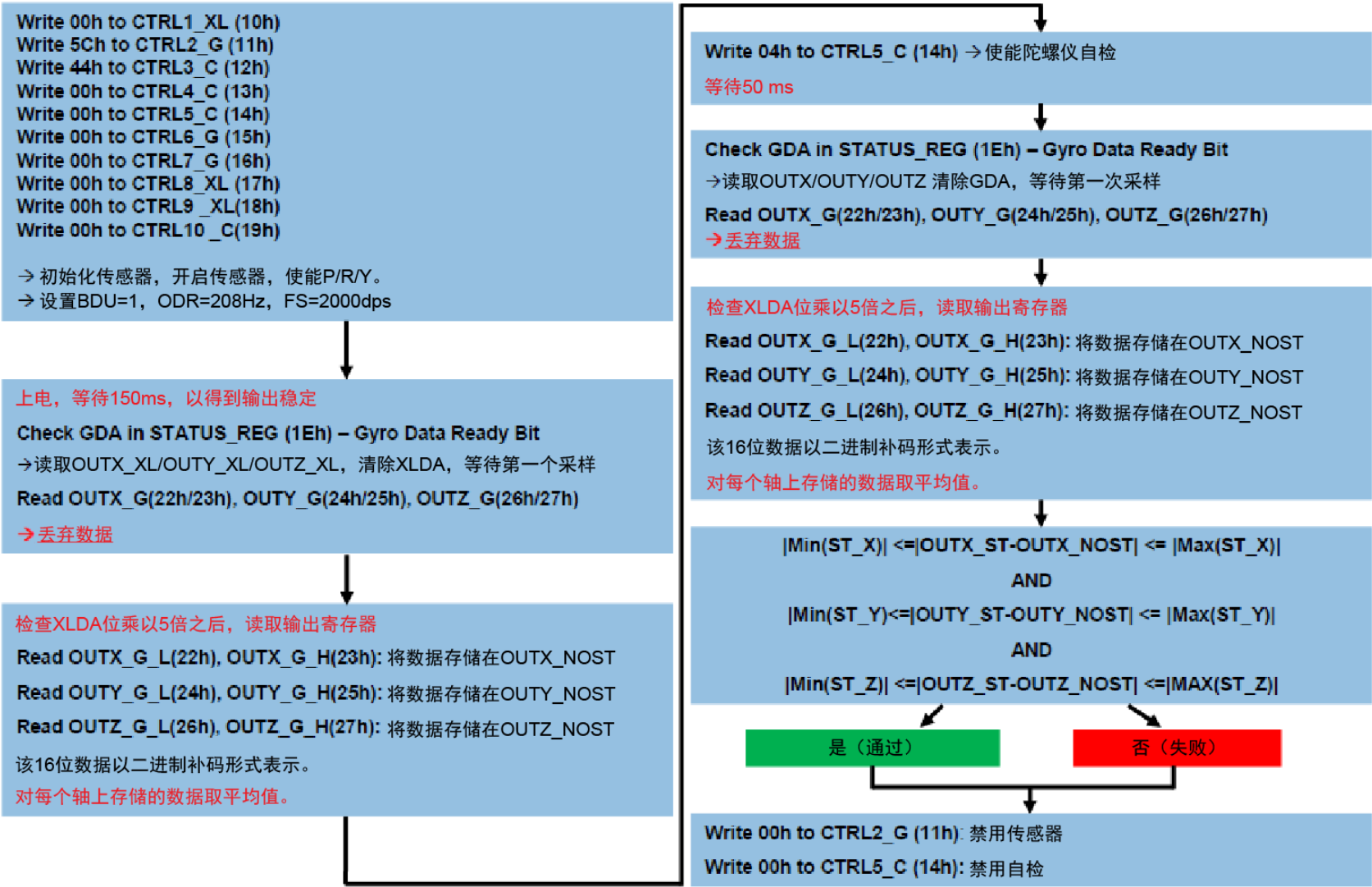
当 CTRL5_C 寄存器的 ST[1:0]_G 位被设定为 00b 时，陀螺仪自检功能关闭；当 ST[1:0]_G 位被置为 01b（正符号自检）或 11b（负符号自检）时，该功能使能。

当陀螺仪自检功能激活时，传感器的输出由作用在传感器上的角速度和静电测试力的代数和给出。

完整的陀螺仪自检过程如图 37. 陀螺仪自检步骤中所示。

图 37. 陀螺仪自检步骤

注：自检过程中请保持器件稳定。



根据保密协议，不可复制



版本历史

表 76. 文档版本历史

日期	版本	变更
2017 年 6 月 28	1	初始版本
2018 年 1 月 26 日	2	更新了简介 更新了表 1 下方的注释。引脚状态 更新了表 7。功耗 更新了 3.6 节“Continue 模式” 更新了表 12。加速度计开启/关闭时间（LPF2 和 HP 禁用） 更新了 5.3 节“唤醒中断”和 5.5.3 节“单击和双击识别配置” 更新了 5.6 节“活动/不活动识别”和图 18。活动/不活动识别（利用斜率滤波器） 更新了 7.2.5 节“SLV0_ADD（02h）、SLV0_SUBADD（03h）和 SLAVE0_CONFIG（04h）”以及第 8.1.2 节“FIFO_CTRL2（07h）” 更新了 7.4 节“传感器集合模式示例” 更新了 7.5.4 节“铁磁校正示例”
2018 年 8 月 14 日	3	更新了 第 6.1 节 计步功能：步伐侦测和步数计算 更新了 第 6.2 节 大幅运动检测 更新了 第 6.3 节 相对倾斜 更新了 第 6.4 节 绝对手腕倾斜 更新了 第 7.2.3 节 FUNC_SRC1（53h） 更新了 第 8.2.2 节 FIFO 模式 更新了 第 8.2.3 节 Continue 模式 更新了 图 30. Continue-FIFO 模式 更新了 图 33. FIFO 阈值（STOP_ON_FTH=0） 更新了 图 34. FIFO 模式下的 FIFO 阈值（STOP_ON_FTH=1） 更新了 图 35. Continue 模式下的 FIFO 阈值（STOP_ON_FTH=1） 更新了 第 10.2 节 陀螺仪自检

目录

1	引脚说明.....	2
2	寄存器.....	4
2.1	嵌入功能寄存器	8
3	工作模式.....	10
3.1	下电模式	12
3.2	高性能模式	12
3.3	正常模式	12
3.4	低功耗模式	12
3.5	陀螺仪睡眠模式	12
3.6	连接模式	12
3.7	加速度计带宽	13
3.7.1	加速度计斜率滤波器.....	15
3.8	陀螺仪带宽	16
3.9	加速度计和陀螺仪开启/关断时间	18
4	模式 1 - 读取输出数据	22
4.1	启动序列	22
4.2	使用状态寄存器	22
4.3	使用数据准备就绪信号.....	23
4.3.1	DRDY 屏蔽功能.....	23
4.4	使用块数据更新 (block data update, BDU) 功能	24
4.5	认识输出数据	24
4.5.1	大小端选择.....	24
4.5.2	输出数据示例	25
4.6	加速度计偏移寄存器.....	25
4.7	环行功能	25
4.7.1	FIFO 输出寄存器环行	25
4.7.2	状态寄存器环行	25
4.7.3	传感器输出寄存器环行	26
4.8	边沿感应和电平感应数据使能 (DEN)	27

4.8.1	边沿感应触发模式	27
4.8.2	电平感应触发模式	28
4.8.3	电平感应锁存模式	30
4.8.4	启用电平感应 FIFO	31
4.8.5	用于 DEN 冲压的 LSB 选择	31
5	中断生成	32
5.1	中断引脚配置	33
5.2	自由落体中断	35
5.3	唤醒中断	37
5.4	6D/4D 定向检测	40
5.4.1	6D 定向检测	40
5.4.2	4D 方向检测	41
5.5	单击和双击识别	42
5.5.1	单击	43
5.5.2	双击	44
5.5.3	单击和双击识别配置	45
5.5.4	单击示例	46
5.5.5	双击示例	46
5.6	活动/不活动识别	47
5.7	启动状态	49
6	嵌入功能	50
6.1	计步功能：步伐侦测和步数计算	50
6.2	大幅运动检测	52
6.3	相对倾斜	53
6.4	绝对手腕倾斜	54
6.5	时间戳	56
7	模式 2 - 传感器集合 (sensor hub) 模式	57
7.1	传感器集合 (sensor hub) 模式说明	57
7.2	传感器集合 (sensor hub) 模式寄存器	57
7.2.1	CTRL10_C (19h)	57
7.2.2	MASTER_CONFIG (1Ah)	58

7.2.3	FUNC_SRC1 (53h)	58
7.2.4	FUNC_SRC2 (54h)	59
7.2.5	SLV0_ADD (02h), SLV0_SUBADD (03h), SLAVE0_CONFIG (04h)	59
7.2.6	SLV1_ADD (05h), SLV1_SUBADD (06h), SLAVE1_CONFIG (07h)	60
7.2.7	SLV2_ADD (08h), SLV2_SUBADD (09h), SLAVE2_CONFIG (0Ah)	62
7.2.8	SLV3_ADD (0Bh), SLV3_SUBADD (0Ch), SLAVE3_CONFIG (0Dh)	63
7.2.9	DATAWRITE_SRC_MODE_SUB_SLV0 (0Eh)	64
7.2.10	SENSORHUBx_REG 寄存器	64
7.3	传感器集合 (sensor hub) 直连功能	65
7.3.1	直连功能使能	66
7.3.2	直连功能禁用	66
7.4	传感器集合 (sensor hub) 模式示例	66
7.5	磁力计硬磁/软磁校准	68
7.5.1	硬磁校准	68
7.5.2	软磁校准	70
7.5.3	得到磁力计补偿数据	73
7.5.4	铁磁校准示例	73
8	先进先出 (FIFO) 缓冲区	77
8.1	FIFO 寄存器	77
8.1.1	FIFO_CTRL1 (06h)	77
8.1.2	FIFO_CTRL2 (07h)	78
8.1.3	FIFO_CTRL3 (08h)	79
8.1.4	FIFO_CTRL4 (09h)	80
8.1.5	FIFO_CTRL5 (0Ah)	80
8.1.6	FIFO_STATUS1 (3Ah)	81
8.1.7	FIFO_STATUS2 (3Bh)	82
8.1.8	FIFO_STATUS3 (3Ch)	82
8.1.9	FIFO_STATUS4 (3Dh)	83
8.1.10	FIFO_DATA_OUT_L (3Eh)	83
8.1.11	FIFO_DATA_OUT_H (3Fh)	83
8.2	FIFO 模式	83
8.2.1	Bypass 模式	83

8.2.2	FIFO 模式	83
8.2.3	Continue 模式	86
8.2.4	Continue-FIFO 模式	88
8.2.5	Bypass-Continue 模式	91
8.3	设置 FIFO 触发、FIFO ODR 和抽取因子	94
8.3.1	使用 FIFO 时改变 ODR 或 FIFO 配置步骤	94
8.4	从 FIFO 恢复数据	95
8.5	FIFO 标识	95
8.5.1	示例 1	95
8.5.2	示例 2	96
8.5.3	示例 3	96
8.6	FIFO 阈值	98
8.7	陀螺仪和加速度计数据的高位部分	104
8.8	FIFO 中的计步器和时间戳数据	105
8.9	FIFO 中的温度数据	106
9	温度传感器	107
9.1	温度数据计算示例	107
10	自检功能	108
10.1	加速度计自检	108
10.2	陀螺仪自检	110
	版本历史	112



表一览

表 1.	引脚状态	3
表 2.	寄存器	4
表 3.	嵌入功能寄存器 (A 区)	8
表 4.	嵌入功能寄存器 (B 区)	9
表 5.	加速度计 ODR 和功耗模式选择	10
表 6.	陀螺仪 ODR 和功耗模式选择	10
表 7.	功耗	11
表 8.	加速度计模拟滤波器带宽	13
表 9.	加速度计带宽选择	14
表 10.	陀螺仪数字 HP 滤波器截止频率选择	16
表 11.	陀螺仪总带宽选择	16
表 12.	加速度计开启/关闭时间 (LPF2 和 HP 禁用)	18
表 13.	要丢弃的加速度计样本	18
表 14.	陀螺仪开启/关闭时间 (HP 禁用)	19
表 15.	要丢弃的陀螺仪样本 (LPF1 禁用)	20
表 16.	对于所有 ODR, 要丢弃的陀螺仪样本 (LPF1 使能)	20
表 17.	输出数据寄存器内容 vs. 加速度 ($FS_{XL} = 2g$)	25
表 18.	输出数据寄存器内容 vs. 角速率 ($FS_G = \pm 250 dps$)	25
表 19.	输出寄存器环行模式	26
表 20.	DEN 配置	27
表 21.	INT1_CTRL 寄存器	33
表 22.	MD1_CFG 寄存器	33
表 23.	INT2_CTRL 寄存器	33
表 24.	MD2_CFG 寄存器	34
表 25.	自由落体阈值 LSB 值	37
表 26.	D6D_SRC 寄存器	40
表 27.	4D/6D 功能阈值	40
表 28.	6D 定位下的 D6D_SRC 寄存器	41
表 29.	TAP_SRC 寄存器	46
表 30.	不活动事件配置	47
表 31.	CTRL10_C 寄存器	58
表 32.	MASTER_CONFIG 寄存器	58
表 33.	FUNC_SRC1 寄存器	59
表 34.	FUNC_SRC2 寄存器	59
表 35.	SLV0_ADD 寄存器	59
表 36.	SLV0_SUBADD 寄存器	59
表 37.	SLAVE0_CONFIG 寄存器	60
表 38.	SLV1_ADD 寄存器	60
表 39.	SLV1_SUBADD 寄存器	60
表 40.	SLAVE1_CONFIG 寄存器	61
表 41.	SLV2_ADD 寄存器	62
表 42.	SLV2_SUBADD 寄存器	62
表 43.	SLAVE2_CONFIG 寄存器	62
表 44.	SLV3_ADD 寄存器	63
表 45.	SLV3_SUBADD 寄存器	63
表 46.	SLAVE3_CONFIG 寄存器	63
表 47.	DATAWRITE_SRC_MODE_SUB_SLV0 寄存器	64
表 48.	铁磁校准配置	68
表 49.	硬磁寄存器值	74
表 50.	软磁寄存器值	74
表 51.	FIFO_CTRL1 寄存器	78
表 52.	FIFO_CTRL2 寄存器	78



表 53.	FIFO_CTRL3 寄存器	79
表 54.	陀螺仪 FIFO 抽取设置	79
表 55.	加速度计 FIFO 抽取设置	79
表 56.	FIFO_CTRL4 寄存器	80
表 57.	第 3 FIFO 数据集抽取设置	80
表 58.	第 4 FIFO 数据集抽取设置	80
表 59.	FIFO_CTRL5 寄存器	81
表 60.	FIFO ODR 选择设置	81
表 61.	FIFO 模式选择	81
表 62.	FIFO_STATUS1 寄存器	81
表 63.	FIFO_STATUS2 寄存器	82
表 64.	FIFO_STATUS2 特性（一个传感器处于 FIFO 模式，且 STOP_ON_FTH = 0 的情况下）	82
表 65.	FIFO_STATUS3 寄存器	82
表 66.	FIFO_STATUS4 寄存器	83
表 67.	FIFO_DATA_OUT_L 寄存器	83
表 68.	FIFO_DATA_OUT_H 寄存器	83
表 69.	示例 1: FIFO_PATTERN[9:0]位和下一个读数	96
表 70.	示例 2: FIFO_PATTERN[9:0]位和下一个读数	96
表 71.	示例 3: FIFO_PATTERN[9:0]位和下一个读数	97
表 72.	FIFO 中陀螺仪和加速度计数据的高位部分	104
表 73.	FIFO 中时间戳和计数器数据	105
表 74.	FIFO 中的温度数据	106
表 75.	输出数据寄存器内容 vs. 温度	107
表 76.	文档版本历史	112

图一览

图 1.	引脚连接	2
图 2.	加速度计滤波链	13
图 3.	加速度计斜率滤波器	15
图 4.	陀螺仪数字链	16
图 5.	数据准备就绪信号	23
图 6.	边沿感应触发模式, DEN 低电平有效	27
图 7.	电平感应触发模式, DEN 低电平有效	28
图 8.	电平感应触发模式, DEN 低电平有效, DEN_DRDY 在 INT1 上	29
图 9.	电平感应锁存模式, DEN 低电平有效	30
图 10.	电平感应锁存模式, DEN 低电平有效, DEN_DRDY 在 INT1 上	30
图 11.	电平感应 FIFO 使能模式, DEN 低电平有效	31
图 12.	自由落体中断	36
图 13.	唤醒中断 (利用斜率滤波器)	38
图 14.	6D 识别方向	41
图 15.	单击事件识别	43
图 16.	双击事件识别 (LIR 位 = 0)	44
图 17.	单击和双击识别 (LIR 位 = 0)	45
图 18.	活动/不活动识别 (利用斜率滤波器)	48
图 19.	计步器去抖动	51
图 20.	计步器最小阈值	51
图 21.	倾斜度检测	54
图 22.	模式 2 下外部传感器连接	57
图 23.	SENSORHUBx_REG 配置示例	64
图 24.	直连功能	65
图 25.	硬磁效应 (X-Y 2D 散点图)	69
图 26.	软磁效应 (X-Y 2D 散点图)	71
图 27.	硬磁/软磁校准框图	73
图 28.	FIFO 模式 (STOP_ON_FTH = 0)	85
图 29.	Continue 模式	87
图 30.	Continue-FIFO 模式	89
图 31.	Bypass-Continue 模式	92
图 32.	FIFO 触发信号选择	94
图 33.	FIFO 阈值 (STOP_ON_FTH=0)	99
图 34.	FIFO 模式下的 FIFO 阈值 (STOP_ON_FTH=1)	101
图 35.	Continue 模式下的 FIFO 阈值 (STOP_ON_FTH=1)	103
图 36.	加速度计自检步骤	109
图 37.	陀螺仪自检步骤	111



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 标志是 ST 的商标。关于 ST 商标的其他信息，请访问 www.st.com/trademarks。其他所有产品或服务名称是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。

© 2020 STMicroelectronics - 保留所有权利