

LSM6DSO：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 LSM6DSO iNEMO** 六轴惯性传感器模块相关的使用信息和应用提示。

LSM6DSO 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 和 MIPI I3CSM 串口标准输出，组合工作在高性能模式下功耗只要 0.55 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，具备先进的节能能力。

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，**LSM6DSO** 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。

可使用不同连接方式与外部传感器相连，从而实现额外的功能，例如传感器集合（sensor hub）、辅助 SPI 等。

LSM6DSO 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、计步功能和时间戳，并提供强大的定制功能：可单独设定最多 16 个嵌入式有限状态机，用于运动检测或手势识别，例如查看、绝对手腕倾斜、摇晃、连续两次摇晃或拿起。

LSM6DSO 还嵌入了机器学习内核逻辑，可以识别数据模式与用户定义类别集是否匹配。应用的典型示例有跑步、行走和驾车等活动检测。

LSM6DSO 集成有一个 9Kbyte 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。

LSM6DSO 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

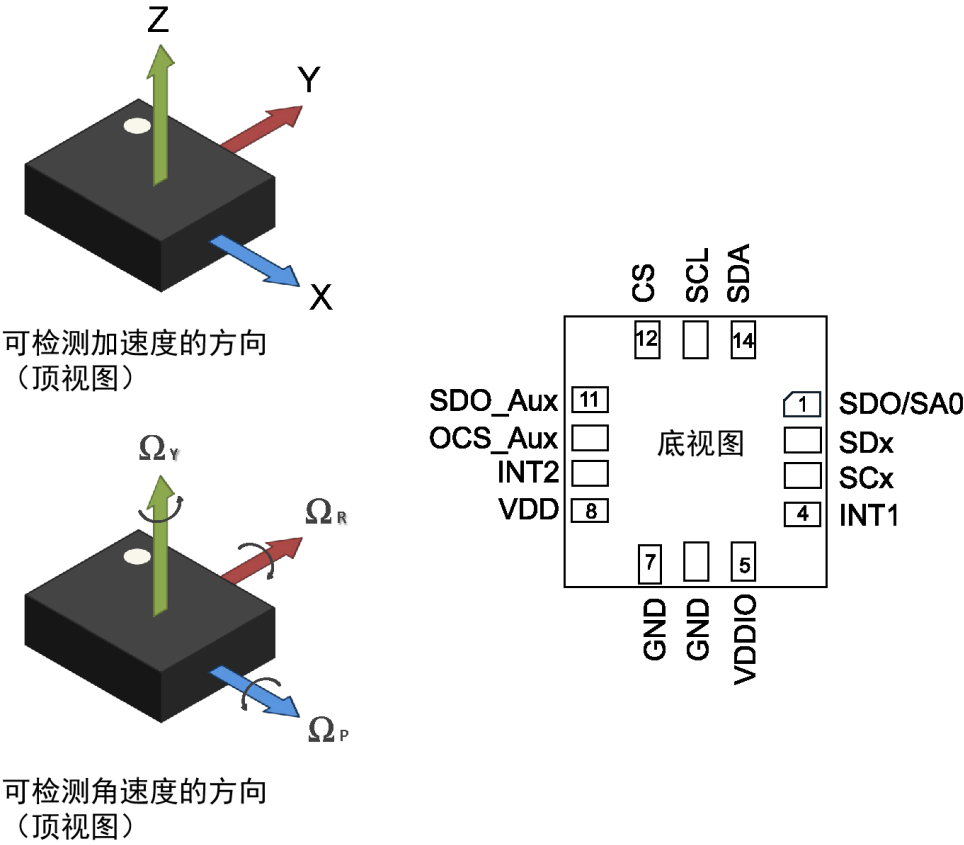


表 1. 引脚状态

引脚#	名称	模式 1 功能 ⁽¹⁾	模式 2 功能 ⁽¹⁾	模式 3/4 功能 ⁽¹⁾	引脚状态模式 1	引脚状态模式 2	引脚状态模式 3/4
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)	SPI 4 线接口串行数据输出 (serial data output, SDO)	默认值：无上拉的输入。 如果 PIN_CTRL 寄存器中的位 SDO_PU_EN = 1，则上拉使能。	默认值：无上拉的输入。 如果 PIN_CTRL 寄存器中的位 SDO_PU_EN = 1，则上拉使能。	默认值：无上拉的输入。 如果 PIN_CTRL 寄存器中的位 SDO_PU_EN = 1，则上拉使能。
	SA0	I ² C 设备地址的最低有效位 (SA0) 静态地址 (SA0) 的 MIPI I3C SM 最低有效位	I ² C 设备地址的最低有效位 (SA0) 静态地址 (SA0) 的 MIPI I3C SM 最低有效位	I ² C 设备地址的最低有效位 (SA0) 静态地址 (SA0) 的 MIPI I3C SM 最低有效位			
2	SDx	连接到 VDDIO 或 GND	I ² C 串行数据主机 (MSDA)	辅助 SPI 3/4 线接口串行数据输入 (SDI) 和 SPI 3 线串行数据输出 (SDO)	默认值：无上拉的输入。 如果 MASTER_CONFIG 寄存器中的位 SHUB_UP_EN = 1，则上拉使能。	默认值：无上拉的输入。 如果 MASTER_CONFIG 寄存器中的位 SHUB_UP_EN = 1，则上拉使能。	默认值：无上拉的输入。 如果 MASTER_CONFIG 寄存器中的位 SHUB_UP_EN = 1，则上拉使能。
3	SCx	连接到 VDDIO 或 GND	I ² C 串行时钟主机 (MSCL)	辅助 SPI 3/4 线接口串行端口时钟 (SPC_Aux)	默认值：无上拉的输入。 如果 MASTER_CONFIG 寄存器中的位 SHUB_UP_EN = 1，则上拉使能。	默认值：无上拉的输入。 如果 MASTER_CONFIG 寄存器中的位 SHUB_UP_EN = 1，则上拉使能。	默认值：无上拉的输入。 如果 MASTER_CONFIG 寄存器中的位 SHUB_UP_EN = 1，则上拉使能。
4	INT1	可编程中断 1 如果器件纯粹用作 MIPI I3C SM 从器件，则此引脚必须置为 1。	可编程中断 1	可编程中断 1	默认值：带下拉的输入。 如果 I3C_BUS_AVB 寄存器中的位 PD_DIS_INT1 = 1，则下拉禁用。	默认值：带下拉的输入。 如果 I3C_BUS_AVB 寄存器中的位 PD_DIS_INT1 = 1，则下拉禁用。	默认值：带下拉的输入。 如果 I3C_BUS_AVB 寄存器中的位 PD_DIS_INT1 = 1，则下拉禁用。
5	Vdd_IO	I/O 引脚的供电	I/O 引脚的供电	I/O 引脚的供电			
6	GND	0 V 电源	0 V 电源	0 V 电源			
7	GND	0 V 电源	0 V 电源	0 V 电源			
8	Vdd	电源	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 (DEN) / I ² C 主线外部同步信号 (MDRDY)	可编程中断 2 (INT2) / 数据使能 (Data enable, DEN)	默认值：输出强制接地。	默认值：输出强制接地。	默认值：输出强制接地。
10	OCS_Aux	连接到 VDDIO 或保持断开	连接到 VDDIO 或保持断开	辅助 SPI 3/4 线接口使能	默认值：带上拉的输入。 如果 PIN_CTRL 寄存器中的位 OIS_PU_DIS = 1，则上拉禁用。	默认值：带上拉的输入。 如果 PIN_CTRL 寄存器中的位 OIS_PU_DIS = 1，则上拉禁用。	输入无上拉。 (无论 PIN_CTRL 寄存器中的 OIS_PU_DIS 位的值是多少)
11	SDO_Aux	连接到 VDDIO 或保持断开	连接到 VDDIO 或保持断开	辅助 SPI 3 线接口：保持断开 辅助 SPI 4 线接口：串行数据输出 (SDO_Aux)	默认值：带上拉的输入。 如果 PIN_CTRL 寄存器中的位 OIS_PU_DIS = 1，则上拉禁用。	默认值：带上拉的输入。 如果 PIN_CTRL 寄存器中的位 OIS_PU_DIS = 1，则上拉禁用。	默认值：无上拉的输入。 如果 CTRL1_OIS 寄存器中的 SIM_OIS 位 = 1 (Aux SPI 3 线) 且 PIN_CTRL 寄存器中的 OIS_PU_DIS 位 = 0，则上拉使能。
12	CS	I ² C 和 MIPI I3C SM /SPI 模式选择 (1: SPI 空闲模式/I ² C 和 MIPI I3C SM 通信使能; 0: SPI 通信模式/I ² C 和 MIPI I3C SM 禁用)	I ² C 和 MIPI I3C SM /SPI 模式选择 (1: SPI 空闲模式/I ² C 和 MIPI I3C SM 通信使能; 0: SPI 通信模式/I ² C 和 MIPI I3C SM 禁用)	I ² C 和 MIPI I3C SM /SPI 模式选择 (1: SPI 空闲模式/I ² C 和 MIPI I3C SM 通信使能; 0: SPI 通信模式/I ² C 和 MIPI I3C SM 禁用)	默认值：带上拉的输入。 如果 CTRL4_C 寄存器中的 I2C_disable 位 = 1 且 CTRL9_XL 寄存器中的 I3C_disable 位 = 1，则上拉禁用。	默认值：带上拉的输入。 如果 CTRL4_C 寄存器中的 I2C_disable 位 = 1 且 CTRL9_XL 寄存器中的 I3C_disable 位 = 1，则上拉禁用。	默认值：带上拉的输入。 如果 CTRL4_C 寄存器中的 I2C_disable 位 = 1 且 CTRL9_XL 寄存器中的 I3C_disable 位 = 1，则上拉禁用。
13	SCL	I ² C/MIPI I3C SM 串行时钟 (SCL) / SPI 串口时钟 (SPC)	I ² C/MIPI I3C SM 串行时钟 (SCL) / SPI 串口时钟 (SPC)	I ² C/MIPI I3C SM 串行时钟 (SCL) / SPI 串口时钟 (SPC)	默认值：无上拉的输入。	默认值：无上拉的输入。	默认值：无上拉的输入。

根据保密协议，不可复制

引脚#	名称	模式 1 功能 ⁽¹⁾	模式 2 功能 ⁽¹⁾	模式 3/4 功能 ⁽¹⁾	引脚状态模式 1	引脚状态模式 2	引脚状态模式 3/4
14	SDA	I ² C/MIPI I3C SM 串行数据 (SDA) /SPI 串行数据输入 (SDI) /3 线接口串行数据输出 (SDO)	I ² C/MIPI I3C SM 串行数据 (SDA) /SPI 串行数据输入 (SDI) /3 线接口串行数据输出 (SDO)	I ² C/MIPI I3C SM 串行数据 (SDA) /SPI 串行数据输入 (SDI) /3 线接口串行数据输出 (SDO)	默认值：无上拉的输入。	默认值：无上拉的输入。	默认值：无上拉的输入。

1. 请参考第 3.7 节 连接模式中的描述。

内部上拉值范围从 30 kΩ 至 50 kΩ，取决于 VDDIO。





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_ACCESS	SHUB_REG_ACCESS	0	0	0	0	0	0
PIN_CTRL	02h	OIS_PU_DIS	SDO_PU_EN	1	1	1	1	1	1
FIFO_CTRL1	07h	WTM7	WTM6	WTM5	WTM4	WTM3	WTM2	WTM1	WTM0
FIFO_CTRL2	08h	STOP_ON_WTM	FIFO_COMPR_RT_RN	0	ODRCHG_EN	0	UNCOPTR_RATE_1	UNCOPTR_RATE_0	WTM8
FIFO_CTRL3	09h	BDR_GY_3	BDR_GY_2	BDR_GY_1	BDR_GY_0	BDR_XL_3	BDR_XL_2	BDR_XL_1	BDR_XL_0
FIFO_CTRL4	0Ah	DEC_TS_BATCH_1	DEC_TS_BATCH_0	ODR_T_BATCH_1	ODR_T_BATCH_0	0	FIFO_MODE2	FIFO_MODE1	FIFO_MODE0
COUNTER_BDR_REG1	0Bh	dataready_pulsed	RST_COUNTER_BDR	TRIG_COUNTER_BDR	0	0	CNT_BDR_TH_10	CNT_BDR_TH_9	CNT_BDR_TH_8
COUNTER_BDR_REG2	0Ch	CNT_BDR_TH_7	CNT_BDR_TH_6	CNT_BDR_TH_5	CNT_BDR_TH_4	CNT_BDR_TH_3	CNT_BDR_TH_2	CNT_BDR_TH_1	CNT_BDR_TH_0
INT1_CTRL	0Dh	DEN_DRDY_flag	INT1_CNT_BDR	INT1_FIFO_FULL	INT1_FIFO_OVR	INT1_FIFO_TH	INT1_BOOT	INT1_DRDY_G	INT1_DRDY_XL
INT2_CTRL	0Eh	0	INT2_CNT_BDR	INT2_FIFO_FULL	INT2_FIFO_OVR	INT2_FIFO_TH	INT2_DRDY_TEMP	INT2_DRDY_G	INT2_DRDY_XL
WHO_AM_I	0Fh	0	1	1	0	1	1	0	0
CTRL1_XL	10h	ODR_XL3	ODR_XL2	ODR_XL1	ODR_XL0	FS1_XL	FS0_XL	LPF2_XL_EN	0
CTRL2_G	11h	ODR_G3	ODR_G2	ODR_G1	ODR_G0	FS1_G	FS0_G	FS_125	0
CTRL3_C	12h	BOOT	BDU	H_LACTIVE	PP_OD	SIM	IF_INC	0	SW_RESET
CTRL4_C	13h	0	SLEEP_G	INT2_on_INT1	0	DRDY_MASK	I2C_disable	LPF1_SEL_G	0
CTRL5_C	14h	XL_ULP_EN	ROUNDING1	ROUNDING0	0	ST1_G	ST0_G	ST1_XL	ST0_XL
CTRL6_C	15h	TRIG_EN	LVL1_EN	LVL2_EN	XL_HM_MODE	USR_OFF_W	FTYPE_2	FTYPE_1	FTYPE_0
CTRL7_G	16h	G_HM_MODE	HP_G_EN	HPM1_G	HPM0_G	0	OIS_ON_EN	USR_OFF_ON_OUT	OIS_ON
CTRL8_XL	17h	HPCF_XL2	HPCF_XL1	HPCF_XL0	HP_REF_MODE_XL	FASTSETTL_MODE_XL	HP_SLOPE_XL_EN	XL_FS_MODE	LOW_PASS_ON_6D
CTRL9_XL	18h	DEN_X	DEN_Y	DEN_Z	DEN_XL_G	DEN_XL_EN	DEN_LH	I3C_disable	0
CTRL10_C	19h	0	0	TIMESTAMP_EN	0	0	0	0	0
ALL_INT_SRC	1Ah	TIMESTAMP_ENDCOUNT	0	SLEEP_CHANGE_IA	D6D_IA	DOUBLE_TAP	SINGLE_TAP	WU_IA	FF_IA
WAKE_UP_SRC	1Bh	0	SLEEP_CHANGE_IA	FF_IA	SLEEP_STATE	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 / STATUS_SPIAux	1Eh	0	0	0	0	0	TDA / GYRO_SETTLING	GDA / GDA	XLDA / XLDA

根据保密协议，不可复制



寄存器名	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
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_A	28h	D7	D6	D5	D4	D3	D2	D1	D0
OUTX_H_A	29h	D15	D14	D13	D12	D11	D10	D9	D8
OUTY_L_A	2Ah	D7	D6	D5	D4	D3	D2	D1	D0
OUTY_H_A	2Bh	D15	D14	D13	D12	D11	D10	D9	D8
OUTZ_L_A	2Ch	D7	D6	D5	D4	D3	D2	D1	D0
OUTZ_H_A	2Dh	D15	D14	D13	D12	D11	D10	D9	D8
EMB_FUNC_STATUS_MAINPAGE	35h	IS_FSM_LC	0	IS_SIGMOT	IS_TILT	IS_STEP_DET	0	0	0
FSM_STATUS_A_MAINPAGE	36h	IS_FSM8	IS_FSM7	IS_FSM6	IS_FSM5	IS_FSM4	IS_FSM3	IS_FSM2	IS_FSM1
FSM_STATUS_B_MAINPAGE	37h	IS_FSM16	IS_FSM15	IS_FSM14	IS_FSM13	IS_FSM12	IS_FSM11	IS_FSM10	IS_FSM9
STATUS_MASTER_MAINPAGE	39h	WR_ONCE_DONE	SLAVE3_NACK	SLAVE2_NACK	SLAVE1_NACK	SLAVE0_NACK	0	0	SENS_HUB_ENDOP
FIFO_STATUS1	3Ah	DIFF_FIFO_7	DIFF_FIFO_6	DIFF_FIFO_5	DIFF_FIFO_4	DIFF_FIFO_3	DIFF_FIFO_2	DIFF_FIFO_1	DIFF_FIFO_0
FIFO_STATUS2	3Bh	FIFO_WTM_IA	FIFO_OVR_IA	FIFO_FULL_IA	COUNTER_BDR_IA	FIFO_OVR_LATCHED	0	DIFF_FIFO_9	DIFF_FIFO_8
TIMESTAMP0	40h	T7	T6	T5	T4	T3	T2	T1	T0
TIMESTAMP1	41h	T15	T14	T13	T12	T11	T10	T9	T8
TIMESTAMP2	42h	T23	T22	T21	T20	T19	T18	T17	T16
TIMESTAMP3	43h	T31	T30	T29	T28	T27	T26	T25	T24
TAP_CFG0	56h	0	INT_CLR_ON_READ	SLEEP_STATUS_ON_INT	SLOPE_FDS	TAP_X_EN	TAP_Y_EN	TAP_Z_EN	LIR
TAP_CFG1	57h	TAP_PRIORITY_2	TAP_PRIORITY_1	TAP_PRIORITY_0	TAP_THS_X_4	TAP_THS_X_3	TAP_THS_X_2	TAP_THS_X_1	TAP_THS_X_0
TAP_CFG2	58h	INTERRRUPTS_ENABLE	INACT_EN1	INACT_EN0	TAP_THS_Y_4	TAP_THS_Y_3	TAP_THS_Y_2	TAP_THS_Y_1	TAP_THS_Y_0
TAP_THS_6D	59h	D4D_EN	SIXD_THS1	SIXD_THS0	TAP_THS_Z_4	TAP_THS_Z_3	TAP_THS_Z_2	TAP_THS_Z_1	TAP_THS_Z_0
INT_DUR2	5Ah	DUR3	DUR2	DUR1	DUR0	QUIET1	QUIET0	SHOCK1	SHOCK0
WAKE_UP_THS	5Bh	SINGLE_DOUBLE_TAP	USR_OFF_ON_WU	WK_THS5	WK_THS4	WK_THS3	WK_THS2	WK_THS1	WK_THS0
WAKE_UP_DUR	5Ch	FF_DUR5	WAKE_DUR1	WAKE_DUR0	WAKE_THS_W	SLEEP_DUR3	SLEEP_DUR2	SLEEP_DUR1	SLEEP_DUR0



根据保密协议，不可复制

寄存器名	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
FREE_FALL	5Dh	FF_DUR4	FF_DUR3	FF_DUR2	FF_DUR1	FF_DUR0	FF_THS2	FF_THS1	FF_THS0
MD1_CFG	5Eh	INT1_SLEEP_CHANGE	INT1_SINGLE_TAP	INT1_WU	INT1_FF	INT1_DOUBLE_TAP	INT1_6D	INT1_EMB_FUNC	INT1_SHUB
MD2_CFG	5Fh	INT2_SLEEP_CHANGE	INT2_SINGLE_TAP	INT2_WU	INT2_FF	INT2_DOUBLE_TAP	INT2_6D	INT2_EMB_FUNC	INT2_TIMESTAMP
I3C_BUS_AVB	62h	0	0	0	I3C_Bus_Avb_Sel1	I3C_Bus_Avb_Sel0	0	0	PD_DIS_INT1
INTERNAL_FREQ_FINE	63h	FREQ_FINE7	FREQ_FINE6	FREQ_FINE5	FREQ_FINE4	FREQ_FINE3	FREQ_FINE2	FREQ_FINE1	FREQ_FINE0
INT_OIS	6Fh	INT2_DRDY_OIS	LVL2_OIS	DEN_LH_OIS	-	-	0	ST1_XL_OIS	ST0_XL_OIS
CTRL1_OIS	70h	0	LVL1_OIS	SIM_OIS	Mode4_EN	FS1_G_OIS	FS0_G_OIS	FS_125_OIS	OIS_EN_SPI2
CTRL2_OIS	71h	-	-	HPM1_OIS	HPM0_OIS	0	FTYPE_1_OIS	FTYPE_0_OIS	HP_EN_OIS
CTRL3_OIS	72h	FS1_XL_OIS	FS0_XL_OIS	FILTER_XL_CONF_OIS_2	FILTER_XL_CONF_OIS_1	FILTER_XL_CONF_OIS_0	ST1_OIS	ST0_OIS	ST_OIS_CLAMPDIS
X_OFS_USR	73h	X_OFS_USR_7	X_OFS_USR_6	X_OFS_USR_5	X_OFS_USR_4	X_OFS_USR_3	X_OFS_USR_2	X_OFS_USR_1	X_OFS_USR_0
Y_OFS_USR	74h	Y_OFS_USR_7	Y_OFS_USR_6	Y_OFS_USR_5	Y_OFS_USR_4	Y_OFS_USR_3	Y_OFS_USR_2	Y_OFS_USR_1	Y_OFS_USR_0
Z_OFS_USR	75h	Z_OFS_USR_7	Z_OFS_USR_6	Z_OFS_USR_5	Z_OFS_USR_4	Z_OFS_USR_3	Z_OFS_USR_2	Z_OFS_USR_1	Z_OFS_USR_0
FIFO_DATA_OUT_TAG	78h	TAG_SENSOR_4	TAG_SENSOR_3	TAG_SENSOR_2	TAG_SENSOR_1	TAG_SENSOR_0	TAG_CNT_1	TAG_CNT_0	TAG_PARITY
FIFO_DATA_OUT_X_L	79h	D7	D6	D5	D4	D3	D2	D1	D0
FIFO_DATA_OUT_X_H	7Ah	D15	D14	D13	D12	D11	D10	D9	D8
FIFO_DATA_OUT_Y_L	7Bh	D7	D6	D5	D4	D3	D2	D1	D0
FIFO_DATA_OUT_Y_H	7Ch	D15	D14	D13	D12	D11	D10	D9	D8
FIFO_DATA_OUT_Z_L	7Dh	D7	D6	D5	D4	D3	D2	D1	D0
FIFO_DATA_OUT_Z_H	7Eh	D15	D14	D13	D12	D11	D10	D9	D8



2.1 嵌入功能寄存器

下表提供了器件中可用的嵌入功能的寄存器和相应地址的列表。当 FUNC_CFG_ACCESS 寄存器中的 FUNC_CFG_ACCESS 位置为 1 时，嵌入功能寄存器可供访问。

表 3. 嵌入功能寄存器

寄存器名	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
PAGE_SEL	02h	PAGE_SEL3	PAGE_SEL2	PAGE_SEL1	PAGE_SEL0	0	0	0	1
EMB_FUNC_EN_A	04h	0	0	SIGN_MOTION_EN	TILT_EN	PEDO_EN	0	0	0
EMB_FUNC_EN_B	05h	0	0	0	PEDO_ADV_EN	FIFO_COMPR_EN	0	0	FSM_EN
PAGE_ADDRESS	08h	PAGE_ADDR7	PAGE_ADDR6	PAGE_ADDR5	PAGE_ADDR4	PAGE_ADDR3	PAGE_ADDR2	PAGE_ADDR1	PAGE_ADDR0
PAGE_VALUE	09h	PAGE_VALUE7	PAGE_VALUE6	PAGE_VALUE5	PAGE_VALUE4	PAGE_VALUE3	PAGE_VALUE2	PAGE_VALUE1	PAGE_VALUE0
EMB_FUNC_INT1	0Ah	INT1_FSM_LC	0	INT1_SIG_MOT	INT1_TILT	INT1_STEP_DETECTOR	0	0	0
FSM_INT1_A	0Bh	INT1_FSM8	INT1_FSM7	INT1_FSM6	INT1_FSM5	INT1_FSM4	INT1_FSM3	INT1_FSM2	INT1_FSM1
FSM_INT1_B	0Ch	INT1_FSM16	INT1_FSM15	INT1_FSM14	INT1_FSM13	INT1_FSM12	INT1_FSM11	INT1_FSM10	INT1_FSM9
EMB_FUNC_INT2	0Eh	INT2_FSM_LC	0	INT2_SIG_MOT	INT2_TILT	INT2_STEP_DETECTOR	0	0	0
FSM_INT2_A	0Fh	INT2_FSM8	INT2_FSM7	INT2_FSM6	INT2_FSM5	INT2_FSM4	INT2_FSM3	INT2_FSM2	INT2_FSM1
FSM_INT2_B	10h	INT2_FSM16	INT2_FSM15	INT2_FSM14	INT2_FSM13	INT2_FSM12	INT2_FSM11	INT2_FSM10	INT2_FSM9
EMB_FUNC_STATUS	12h	IS_FSM_LC	0	IS_SIGMOT	IS_TILT	IS_STEP_DET	0	0	0
FSM_STATUS_A	13h	IS_FSM8	IS_FSM7	IS_FSM6	IS_FSM5	IS_FSM4	IS_FSM3	IS_FSM2	IS_FSM1
FSM_STATUS_B	14h	IS_FSM16	IS_FSM15	IS_FSM14	IS_FSM13	IS_FSM12	IS_FSM11	IS_FSM10	IS_FSM9
PAGE_RW	17h	EMB_FUNC_LIR	PAGE_WRITE	PAGE_READ	0	0	0	0	0
EMB_FUNC_FIFO_CFG	44h	0	PEDO_FIFO_EN	0	0	0	0	0	0
FSM_ENABLE_A	46h	FSM8_EN	FSM7_EN	FSM6_EN	FSM5_EN	FSM4_EN	FSM3_EN	FSM2_EN	FSM1_EN
FSM_ENABLE_B	47h	FSM16_EN	FSM15_EN	FSM14_EN	FSM13_EN	FSM12_EN	FSM11_EN	FSM10_EN	FSM9_EN
FSM_LONG_COUNTER_L	48h	FSM_LC_7	FSM_LC_6	FSM_LC_5	FSM_LC_4	FSM_LC_3	FSM_LC_2	FSM_LC_1	FSM_LC_0
FSM_LONG_COUNTER_H	49h	FSM_LC_15	FSM_LC_14	FSM_LC_13	FSM_LC_12	FSM_LC_11	FSM_LC_10	FSM_LC_9	FSM_LC_8
FSM_LONG_COUNTER_CLEAR	4Ah	0	0	0	0	0	0	FSM_LC_CLEARED	FSM_LC_CLEAR
FSM_OUTS1	4Ch	P_X	N_X	P_Y	N_Y	P_Z	N_Z	P_V	N_V
FSM_OUTS2	4Dh	P_X	N_X	P_Y	N_Y	P_Z	N_Z	P_V	N_V
FSM_OUTS3	4Eh	P_X	N_X	P_Y	N_Y	P_Z	N_Z	P_V	N_V
FSM_OUTS4	4Fh	P_X	N_X	P_Y	N_Y	P_Z	N_Z	P_V	N_V
FSM_OUTS5	50h	P_X	N_X	P_Y	N_Y	P_Z	N_Z	P_V	N_V
FSM_OUTS6	51h	P_X	N_X	P_Y	N_Y	P_Z	N_Z	P_V	N_V



寄存器名	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
FSM_OUTS7	52h	P_X	N_X	P_Y	N_Y	P_Z	N_Z	P_V	N_V
FSM_OUTS8	53h	P_X	N_X	P_Y	N_Y	P_Z	N_Z	P_V	N_V
FSM_OUTS9	54h	P_X	N_X	P_Y	N_Y	P_Z	N_Z	P_V	N_V
FSM_OUTS10	55h	P_X	N_X	P_Y	N_Y	P_Z	N_Z	P_V	N_V
FSM_OUTS11	56h	P_X	N_X	P_Y	N_Y	P_Z	N_Z	P_V	N_V
FSM_OUTS12	57h	P_X	N_X	P_Y	N_Y	P_Z	N_Z	P_V	N_V
FSM_OUTS13	58h	P_X	N_X	P_Y	N_Y	P_Z	N_Z	P_V	N_V
FSM_OUTS14	59h	P_X	N_X	P_Y	N_Y	P_Z	N_Z	P_V	N_V
FSM_OUTS15	5Ah	P_X	N_X	P_Y	N_Y	P_Z	N_Z	P_V	N_V
FSM_OUTS16	5Bh	P_X	N_X	P_Y	N_Y	P_Z	N_Z	P_V	N_V
EMB_FUNC_ODR_CFG_B	5Fh	0	1	0	FSM_ODR1	FSM_ODR0	0	1	1
STEP_COUNTER_L	62h	STEP_7	STEP_6	STEP_5	STEP_4	STEP_3	STEP_2	STEP_1	STEP_0
STEP_COUNTER_H	63h	STEP_15	STEP_14	STEP_13	STEP_12	STEP_11	STEP_10	STEP_9	STEP_8
EMB_FUNC_SRC	64h	PEDO_RST_STEP	0	STEP_DETECTED	STEP_COUNT_DELTA_IA	STEP_OVERFLOW	STEP_COUNTER_BIT_SET	0	0
EMB_FUNC_INIT_A	66h	0	0	SIG_MOT_INIT	TILT_INIT	STEP_DET_INIT	0	0	0
EMB_FUNC_INIT_B	67h	0	0	0	0	FIFO_COMPR_INIT	0	0	FSM_INIT

2.2 嵌入高级功能页

下表提供了嵌入高级功能 page 0 的寄存器列表。当 PAGE_SEL 寄存器中的 PAGE_SEL[3:0]位被置为 0000b 时，这些寄存器可访问。

表 4. 嵌入高级功能寄存器 - page 0

寄存器名	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
MAG_SENSITIVITY_L	BAh	MAG_SENS7	MAG_SENS6	MAG_SENS5	MAG_SENS4	MAG_SENS3	MAG_SENS2	MAG_SENS1	MAG_SENS0
MAG_SENSITIVITY_H	BBh	MAG_SENS15	MAG_SENS14	MAG_SENS13	MAG_SENS12	MAG_SENS11	MAG_SENS10	MAG_SENS9	MAG_SENS8
MAG_OFFX_L	C0h	MAG_OFFX_7	MAG_OFFX_6	MAG_OFFX_5	MAG_OFFX_4	MAG_OFFX_3	MAG_OFFX_2	MAG_OFFX_1	MAG_OFFX_0
MAG_OFFX_H	C1h	MAG_OFFX_15	MAG_OFFX_14	MAG_OFFX_13	MAG_OFFX_12	MAG_OFFX_11	MAG_OFFX_10	MAG_OFFX_9	MAG_OFFX_8
MAG_OFFY_L	C2h	MAG_OFFY_7	MAG_OFFY_6	MAG_OFFY_5	MAG_OFFY_4	MAG_OFFY_3	MAG_OFFY_2	MAG_OFFY_1	MAG_OFFY_0
MAG_OFFY_H	C3h	MAG_OFFY_15	MAG_OFFY_14	MAG_OFFY_13	MAG_OFFY_12	MAG_OFFY_11	MAG_OFFY_10	MAG_OFFY_9	MAG_OFFY_8
MAG_OFFZ_L	C4h	MAG_OFFZ_7	MAG_OFFZ_6	MAG_OFFZ_5	MAG_OFFZ_4	MAG_OFFZ_3	MAG_OFFZ_2	MAG_OFFZ_1	MAG_OFFZ_0
MAG_OFFZ_H	C5h	MAG_OFFZ_15	MAG_OFFZ_14	MAG_OFFZ_13	MAG_OFFZ_12	MAG_OFFZ_11	MAG_OFFZ_10	MAG_OFFZ_9	MAG_OFFZ_8
MAG_SI_XX_L	C6h	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_XX_H	C7h	MAG_SI_XX_15	MAG_SI_XX_14	MAG_SI_XX_13	MAG_SI_XX_12	MAG_SI_XX_11	MAG_SI_XX_10	MAG_SI_XX_9	MAG_SI_XX_8
MAG_SI_XY_L	C8h	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_XY_H	C9h	MAG_SI_XY_15	MAG_SI_XY_14	MAG_SI_XY_13	MAG_SI_XY_12	MAG_SI_XY_11	MAG_SI_XY_10	MAG_SI_XY_9	MAG_SI_XY_8
MAG_SI_XZ_L	CAh	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_XZ_H	CBh	MAG_SI_XZ_15	MAG_SI_XZ_14	MAG_SI_XZ_13	MAG_SI_XZ_12	MAG_SI_XZ_11	MAG_SI_XZ_10	MAG_SI_XZ_9	MAG_SI_XZ_8
MAG_SI_YY_L	CCh	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_YY_H	CDh	MAG_SI_YY_15	MAG_SI_YY_14	MAG_SI_YY_13	MAG_SI_YY_12	MAG_SI_YY_11	MAG_SI_YY_10	MAG_SI_YY_9	MAG_SI_YY_8
MAG_SI_YZ_L	CEh	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_YZ_H	CFh	MAG_SI_YZ_15	MAG_SI_YZ_14	MAG_SI_YZ_13	MAG_SI_YZ_12	MAG_SI_YZ_11	MAG_SI_YZ_10	MAG_SI_YZ_9	MAG_SI_YZ_8
MAG_SI_ZZ_L	D0h	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_SI_ZZ_H	D1h	MAG_SI_ZZ_15	MAG_SI_ZZ_14	MAG_SI_ZZ_13	MAG_SI_ZZ_12	MAG_SI_ZZ_11	MAG_SI_ZZ_10	MAG_SI_ZZ_9	MAG_SI_ZZ_8
MAG_CFG_A	D4h	0	MAG_Y_AXIS2	MAG_Y_AXIS1	MAG_Y_AXIS0	0	MAG_Z_AXIS2	MAG_Z_AXIS1	MAG_Z_AXIS0
MAG_CFG_B	D5h	0	0	0	0	0	MAG_X_AXIS2	MAG_X_AXIS1	MAG_X_AXIS0

下表提供了嵌入高级功能 page 1 的寄存器列表。当 PAGE_SEL 寄存器中的 PAGE_SEL[3:0]位被置为 0001b 时，这些寄存器可访问。

表 5. 嵌入高级功能寄存器 - page 1

寄存器名	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
FSM_LC_TIMEOUT_L	7Ah	FSM_LC_TIMEOUT7	FSM_LC_TIMEOUT6	FSM_LC_TIMEOUT5	FSM_LC_TIMEOUT4	FSM_LC_TIMEOUT3	FSM_LC_TIMEOUT2	FSM_LC_TIMEOUT1	FSM_LC_TIMEOUT0



根据保密协议，不可复制



寄存器名	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
FSM_LC_TIMEOUT_H	7Bh	FSM_LC_TIMEOUT15	FSM_LC_TIMEOUT14	FSM_LC_TIMEOUT13	FSM_LC_TIMEOUT12	FSM_LC_TIMEOUT11	FSM_LC_TIMEOUT10	FSM_LC_TIMEOUT9	FSM_LC_TIMEOUT8
FSM_PROGRAMS	7Ch	FSM_N_PROG7	FSM_N_PROG6	FSM_N_PROG5	FSM_N_PROG4	FSM_N_PROG3	FSM_N_PROG2	FSM_N_PROG1	FSM_N_PROG0
FSM_START_ADD_L	7Eh	FSM_START7	FSM_START6	FSM_START5	FSM_START4	FSM_START3	FSM_START2	FSM_START1	FSM_START0
FSM_START_ADD_H	7Fh	FSM_START15	FSM_START14	FSM_START13	FSM_START12	FSM_START11	FSM_START10	FSM_START9	FSM_START8
PEDO_CMD_REG	83h	0	0	0	0	CARRY_COUNT_EN	FP_REJECTION_EN	0	AD_DET_EN
PEDO_DEB_STEPS_CONF	84h	DEB_STEP7	DEB_STEP6	DEB_STEP5	DEB_STEP4	DEB_STEP3	DEB_STEP2	DEB_STEP1	DEB_STEP0
PEDO_SC_DELTAT_L	D0h	PD_SC_7	PD_SC_6	PD_SC_5	PD_SC_4	PD_SC_3	PD_SC_2	PD_SC_1	PD_SC_0
PEDO_SC_DELTAT_H	D1h	PD_SC_15	PD_SC_14	PD_SC_13	PD_SC_12	PD_SC_11	PD_SC_10	PD_SC_9	PD_SC_8

2.3 传感器集合寄存器

下表提供了器件中可用的传感器集合功能的寄存器和相应地址的列表。当 FUNC_CFG_ACCESS 寄存器中的 SHUB_REG_ACCESS 位被置为 1 时，传感器集合寄存器可访问。

表 6. 传感器集合寄存器

寄存器名	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
SENSOR_HUB_1	02h	SensorHub1_7	SensorHub1_6	SensorHub1_5	SensorHub1_4	SensorHub1_3	SensorHub1_2	SensorHub1_1	SensorHub1_0
SENSOR_HUB_2	03h	SensorHub2_7	SensorHub2_6	SensorHub2_5	SensorHub2_4	SensorHub2_3	SensorHub2_2	SensorHub2_1	SensorHub2_0
SENSOR_HUB_3	04h	SensorHub3_7	SensorHub3_6	SensorHub3_5	SensorHub3_4	SensorHub3_3	SensorHub3_2	SensorHub3_1	SensorHub3_0
SENSOR_HUB_4	05h	SensorHub4_7	SensorHub4_6	SensorHub4_5	SensorHub4_4	SensorHub4_3	SensorHub4_2	SensorHub4_1	SensorHub4_0
SENSOR_HUB_5	06h	SensorHub5_7	SensorHub5_6	SensorHub5_5	SensorHub5_4	SensorHub5_3	SensorHub5_2	SensorHub5_1	SensorHub5_0
SENSOR_HUB_6	07h	SensorHub6_7	SensorHub6_6	SensorHub6_5	SensorHub6_4	SensorHub6_3	SensorHub6_2	SensorHub6_1	SensorHub6_0
SENSOR_HUB_7	08h	SensorHub7_7	SensorHub7_6	SensorHub7_5	SensorHub7_4	SensorHub7_3	SensorHub7_2	SensorHub7_1	SensorHub7_0
SENSOR_HUB_8	09h	SensorHub8_7	SensorHub8_6	SensorHub8_5	SensorHub8_4	SensorHub8_3	SensorHub8_2	SensorHub8_1	SensorHub8_0
SENSOR_HUB_9	0Ah	SensorHub9_7	SensorHub9_6	SensorHub9_5	SensorHub9_4	SensorHub9_3	SensorHub9_2	SensorHub9_1	SensorHub9_0
SENSOR_HUB_10	0Bh	SensorHub10_7	SensorHub10_6	SensorHub10_5	SensorHub10_4	SensorHub10_3	SensorHub10_2	SensorHub10_1	SensorHub10_0
SENSOR_HUB_11	0Ch	SensorHub11_7	SensorHub11_6	SensorHub11_5	SensorHub11_4	SensorHub11_3	SensorHub11_2	SensorHub11_1	SensorHub11_0
SENSOR_HUB_12	0Dh	SensorHub12_7	SensorHub12_6	SensorHub12_5	SensorHub12_4	SensorHub12_3	SensorHub12_2	SensorHub12_1	SensorHub12_0
SENSOR_HUB_13	0Eh	SensorHub13_7	SensorHub13_6	SensorHub13_5	SensorHub13_4	SensorHub13_3	SensorHub13_2	SensorHub13_1	SensorHub13_0
SENSOR_HUB_14	0Fh	SensorHub14_7	SensorHub14_6	SensorHub14_5	SensorHub14_4	SensorHub14_3	SensorHub14_2	SensorHub14_1	SensorHub14_0
SENSOR_HUB_15	10h	SensorHub15_7	SensorHub15_6	SensorHub15_5	SensorHub15_4	SensorHub15_3	SensorHub15_2	SensorHub15_1	SensorHub15_0
SENSOR_HUB_16	11h	SensorHub16_7	SensorHub16_6	SensorHub16_5	SensorHub16_4	SensorHub16_3	SensorHub16_2	SensorHub16_1	SensorHub16_0
SENSOR_HUB_17	12h	SensorHub17_7	SensorHub17_6	SensorHub17_5	SensorHub17_4	SensorHub17_3	SensorHub17_2	SensorHub17_1	SensorHub17_0
SENSOR_HUB_18	13h	SensorHub18_7	SensorHub18_6	SensorHub18_5	SensorHub18_4	SensorHub18_3	SensorHub18_2	SensorHub18_1	SensorHub18_0
MASTER_CONFIG	14h	RST_MASTER_REGS	WRITE_ONCE	START_CONFIG	PASS_THROUGH_MODE	SHUB_PU_EN	MASTER_ON	AUX_SENS_ON1	AUX_SENS_ON0
SLV0_ADD	15h	slave0_add6	slave0_add5	slave0_add4	slave0_add3	slave0_add2	slave0_add1	slave0_add0	rw_0
SLV0_SUBADD	16h	slave0_reg7	slave0_reg6	slave0_reg5	slave0_reg4	slave0_reg3	slave0_reg2	slave0_reg1	slave0_reg0
SLAVE0_CONFIG	17h	SHUB_ODR1	SHUB_ODR0	0	0	BATCH_EXT_SENS_0_EN	Slave0_numop2	Slave0_numop1	Slave0_numop0
SLV1_ADD	18h	slave1_add6	slave1_add5	slave1_add4	slave1_add3	slave1_add2	slave1_add1	slave1_add0	r_1
SLV1_SUBADD	19h	slave1_reg7	slave1_reg6	slave1_reg5	slave1_reg4	slave1_reg3	slave1_reg2	slave1_reg1	slave1_reg0
SLAVE1_CONFIG	1Ah	0	0	0	0	BATCH_EXT_SENS_1_EN	Slave1_numop2	Slave1_numop1	Slave1_numop0
SLV2_ADD	1Bh	slave2_add6	slave1_add5	slave1_add4	slave1_add3	slave1_add2	slave1_add1	slave1_add0	r_2
SLV2_SUBADD	1Ch	slave2_reg7	slave2_reg6	slave2_reg5	slave2_reg4	slave2_reg3	slave2_reg2	slave2_reg1	slave2_reg0



根据保密协议，不可复制

寄存器名	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
SLAVE2_CONFIG	1Dh	0	0	0	0	BATCH_EXT_SENS_2_EN	Slave2_numop2	Slave2_numop1	Slave2_numop0
SLV3_ADD	1Eh	slave3_add6	slave3_add5	slave3_add4	slave3_add3	slave3_add2	slave3_add1	slave3_add0	r_3
SLV3_SUBADD	1Fh	slave3_reg7	slave3_reg6	slave3_reg5	slave3_reg4	slave3_reg3	slave3_reg2	slave3_reg1	slave3_reg0
SLAVE3_CONFIG	20h	0	0	0	0	BATCH_EXT_SENS_3_EN	Slave3_numop2	Slave3_numop1	Slave3_numop0
DATAWRITE_SLV0	21h	Slave0_dataw7	Slave0_dataw6	Slave0_dataw5	Slave0_dataw4	Slave0_dataw3	Slave0_dataw2	Slave0_dataw1	Slave0_dataw0
STATUS_MASTER	22h	WR_ONCE_DONE	SLAVE3_NACK	SLAVE2_NACK	SLAVE1_NACK	SLAVE0_NACK	0	0	SENS_HUB_ENDOP

3 工作模式

LSM6DSO 提供了 3 种可能的操作配置：

- 只有加速度计工作，陀螺仪处于下电或睡眠模式；
- 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，建议在主机侧将连接至器件 IO 引脚的管脚设置成高阻态。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，器件执行一段 10 ms（最长）的启动程序来加载修整参数。启动完成后，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独立配置。加速度计可配置为五种不同的功耗模式：下电、超低功耗、低功耗、正常和高性能模式。陀螺仪可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。

参考数据手册，可以利用 CTRL1_XL 寄存器的输出数据率（ODR_XL）位、CTRL5_C 寄存器的超低功耗使能（XL_ULP_EN）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 7. 加速度计 ODR 和功耗模式选择）。

表 7. 加速度计 ODR 和功耗模式选择

ODR_XL [3:0]	ODR [Hz], 当 XL_ULP_EN = 1 且 XL_HM_MODE = 0 (陀螺仪必须处于下电模式)	ODR [Hz], 当 XL_ULP_EN = 0 且 XL_HM_MODE = 1	ODR [Hz], 当 XL_ULP_EN = 0 且 XL_HM_MODE = 0
0000	省电	省电	省电
1011	1.6 Hz (超低功耗)	1.6 Hz (低功耗)	12.5 Hz (高性能)
0001	12.5 Hz (超低功耗)	12.5 Hz (低功耗)	12.5 Hz (高性能)
0010	26 Hz (超低功耗)	26 Hz (低功耗)	26 Hz (高性能)
0011	52 Hz (超低功耗)	52 Hz (低功耗)	52 Hz (高性能)
0100	104 Hz (超低功耗)	104 Hz (正常模式)	104 Hz (高性能)
0101	208 Hz (超低功耗)	208 Hz (正常模式)	208 Hz (高性能)
0110	不可用	417 Hz (高性能)	417 Hz (高性能)
0111	不可用	833 Hz (高性能)	833 Hz (高性能)
1000	不可用	1.66 kHz (高性能)	1.66 kHz (高性能)
1001	不可用	3.33 kHz (高性能)	3.33 kHz (高性能)
1010	不可用	6.66 kHz (高性能)	6.66 kHz (高性能)

注：在仅加速度计模式下可以选择超低功耗模式。在使能或禁用超低功耗模式前，必须将加速度计置为下电模式。在使用模式 3/4 连接模式时，不能将加速度计配置为超低功耗模式。

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 8. 陀螺仪 ODR 和功耗模式选择）。

表 8. 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	省电	省电
0001	12.5 Hz (低功耗)	12.5 Hz (高性能)
0010	26 Hz (低功耗)	26 Hz (高性能)
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	417 Hz (高性能)	417 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 9. 功耗 (典型值) 显示了不同工作模式下功耗典型值。

表 9. 功耗 (典型值)

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)
省电	-	-	3 µA
睡眠	-	245 µA	-
1.6 Hz (超低功耗)	4.4 µA	-	-
12.5 Hz (超低功耗)	5.5 µA	-	-
26 Hz (超低功耗)	7.0 µA	-	-
52 Hz (超低功耗)	9.5 µA	-	-
104 Hz (超低功耗)	14.5 µA	-	-
208 Hz (超低功耗)	24.5 µA	-	-
1.6 Hz (低功耗)	4.5 µA	-	-
12.5 Hz (低功耗)	9.0 µA	255 µA	265 µA
26 Hz (低功耗)	15 µA	265 µA	280 µA
52 Hz (低功耗)	26 µA	280 µA	300 µA
104 Hz (正常)	45 µA	310 µA	350 µA
208 Hz (正常)	85 µA	375 µA	430 µA
12.5 Hz (高性能)	170 µA	450 µA	550 µA
26 Hz (高性能)	170 µA	450 µA	550 µA
52 Hz (高性能)	170 µA	450 µA	550 µA
104 Hz (高性能)	170 µA	450 µA	550 µA
208 Hz (高性能)	170 µA	450 µA	550 µA
417 Hz (高性能)	170 µA	450 µA	550 µA
833 Hz (高性能)	170 µA	450 µA	550 µA

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)
1.66 kHz (高性能)	170 μ A	450 μ A	550 μ A
3.33 kHz (高性能)	170 μ A	450 μ A	550 μ A
6.66 kHz (高性能)	170 μ A	450 μ A	550 μ A

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、MIPI I3CSM 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 加速度计超低功耗模式

超低功耗模式侧重于器件功耗。超低功耗模式下，低速 ODR 使能。可以通过 ODR_XL 位为加速度计选择六个低速 ODR：1.6 Hz、12.5 Hz、26 Hz、52 Hz、104 Hz 和 208 Hz。陀螺仪必须设置为下电模式。数据中断产生是激活的。



3.6 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_G) 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。

3.7 连接模式

器件提供了四种不同的连接方式，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、MIPI I3CSM 从接口或 SPI (3-/4-线) 串口可用。
- 模式 2：它是传感器集合模式；I²C 从接口、MIPI I3CSM 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。
- 模式 3：除了主 I²C 从接口、MIPI I3CSM 从接口或 SPI (3/4 线) 串口，还有一个用于外部器件连接 (即相机模块) 的辅助 SPI (3/4 线) 串口只对陀螺仪可用。第 8 节 模式 3 和模式 4 - 辅助 SPI 模式中描述了此连接模式。
- 模式 4：除了主 I²C 从接口、MIPI I3CSM 从接口或 SPI (3/4 线) 串口，还有一个用于外部器件连接的辅助 SPI (3/4 线) 串口对陀螺仪和加速度计均可用。第 8 节 模式 3 和模式 4 - 辅助 SPI 模式中描述了此连接模式。

3.8 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 (LPF1) 和复合数字滤波器组。

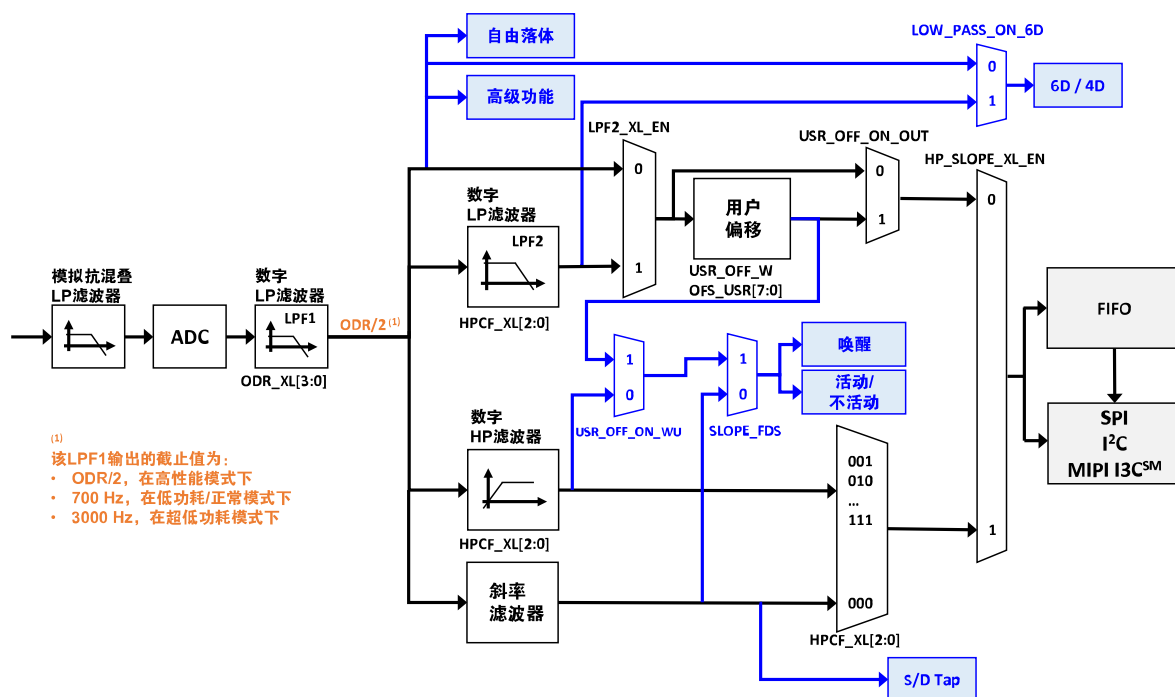
图 2. 加速度计滤波链 (UI 路径) 显示了 UI 路径上的加速度计采样链；第 8 节 模式 3 和模式 4 - 辅助 SPI 模式描述了 OIS 路径上的激活加速度计采样链 (使用模式 4 时)。

来自机械部件的模拟信号在被 ADC 转换前，由模拟抗混叠低通滤波器进行滤波。该抗混叠滤波器仅在高性能模式下使能。

数字 LPF1 滤波器基于选择的加速度计模式提供不同截止值：

- ODR / 2，当加速度计配置为高性能模式时；
- 700 Hz，当加速度计配置为低功耗模式时；
- 3000 Hz，当加速度计配置为超低功耗模式时。

图 2. 加速度计滤波链 (UI 路径)



上图中的“高级功能”块指计步器、步伐检测器、步伐计数器、大幅运动和倾斜功能（如第 6 节 嵌入功能所述），还包括有限状态机。

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。

CTRL1_XL 寄存器的 LPF2_XL_EN 位和 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 10. 模式 1/2/3 下的加速度计带宽选择中所示。参考该表，在低通路径侧，如果 LPF2_XL_EN = 0，则带宽是指 LPF1 带宽；如果 LPF2_XL_EN = 1，则带宽指 LPF2 带宽。在高通路径侧，如果 HPCF_XL[2:0] = 000b，则带宽列表示斜率滤波器带宽；对于所有其他配置，则表示 HP 滤波器带宽。

表 10. 模式 1/2/3 下的加速度计带宽选择 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。

表 10. 模式 1/2/3 下的加速度计带宽选择

HP_SLOPE_XL_EN	LPF2_XL_EN	HPCF_XL[2:0]	带宽 HP	带宽 LP	带宽 ULP	最大总稳定时间 ⁽¹⁾ （要丢弃的采样）
0 (低通路径)	0	-	ODR / 2	700 Hz	3000 Hz	See 表 12
		000	ODR / 4			See 表 12
	1	001	ODR / 10			20
		010	ODR / 20			20
		011	ODR / 45			40
		100	ODR / 100			76
		101	ODR / 200			150
		110	ODR / 400			305
		111	ODR / 800			605
		000	ODR / 4（斜率滤波器）			See 表 12
1 (高通路径)	-	001	ODR / 10			33
		010	ODR / 20			33
		011	ODR / 45			40
		100	ODR / 100			76
		101	ODR / 200			150
		110	ODR / 400			305
		111	ODR / 800			605

1. 最终值的 99% 时的稳定时间，考虑所有输出数据率和所有操作模式切换

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 [2:0] 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 [2:0] 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 [2:0] = 000b 时）或数字高通滤波器（其他 HPCF_XL [2:0] 配置）。HPCF_XL [2:0] 字段也可用来选择 HP 滤波器的截止频率。

高通滤波器参考模式功能可用于加速度计传感器：启用此功能后，当前的 X、Y、Z 加速度计采样将在内部存储，并从所有后续输出值中减去。为了使能参考模式，必须将 CTRL8_XL 寄存器的 HP_REF_MODE_XL 位和 HP_SLOPE_XL_EN 位置为 1，并且 HPCF_XL [2:0] 字段的值必须不等于 000b。当启用 Reference 模式功能时，LPF2 滤波器和 HP 滤波器都不可用。启用 Reference 模式之后的第一个加速度计输出数据必须被丢弃。

CTRL8_XL 寄存器的 FASTSETTL_MODE_XL 位使能加速度计 LPF2 或 HPF 快速建立模式：选择的滤波器在写入此位后设置第二个采样。仅在器件退出下电模式时应用此功能。



3.8.1 加速度计斜率滤波器

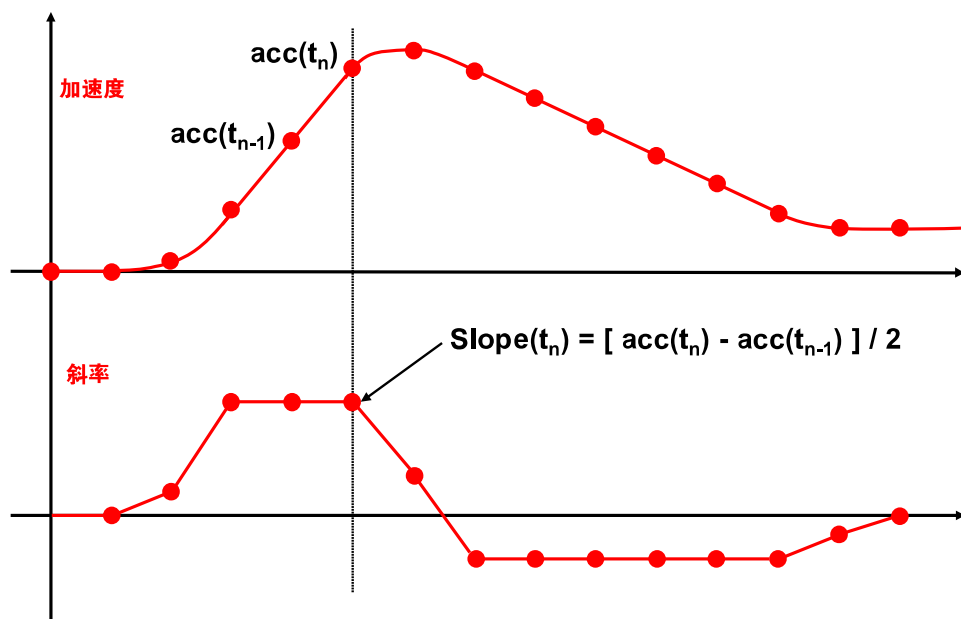
如图 2. 加速度计滤波链 (UI 路径) 中所示, 器件嵌入了一个数字斜率滤波器, 该滤波器还可用于某些嵌入式功能, 如单/双击识别、唤醒检测和活动/不活动。

该斜率滤波器输出数据利用以下公式进行计算:

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

下图举例说明了斜率数据信号的示例。

图 3. 加速度计斜率滤波器



3.9 加速度计开启/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。

加速度计链稳定时间取决于为以下配置选择的功耗模式和输出数据率：

- LPF2 和 HP 滤波器禁用；
- LPF2 或 HP 滤波器使能且带宽选择了 ODR/4。

对于这两种可能的配置，下面的表 11. 加速度计开启/关闭时间（LPF2 和 HP 禁用）和 显示了切换加速度计功耗模式或加速度计 ODR 所需的最大总开启/关闭时间。表 12. 要丢弃的加速度计样本

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 11. 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	超低功耗/低功耗/正常	See 表 12
省电	高性能	See 表 12
低功耗/正常	高性能	见表 12 + 丢弃 1 个额外样本
超低功耗/低功耗/正常	超低功耗/低功耗/正常（ODR 改变）	See 表 12
高性能	低功耗/正常	见表 12 + 丢弃 1 个额外样本
高性能	高性能 @ ODR < 6.66 kHz	丢弃 3 个样本
高性能	高性能 @ ODR = 6.66 kHz	丢弃 7 个样本
超低功耗/低功耗/正常/高性能	省电	1 μ s

1. @ 最终值的 99% 时的稳定时间

表 12. 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 (LPF2 和 HP 滤波器禁用)	要丢弃的采样数 (LPF2 或 HP 滤波器使能且带宽为 ODR/4)
1.6 (超低功耗/低功耗)	0 (第一个采样正确)	1
12.5 (超低功耗/低功耗)	0 (第一个采样正确)	1
26 (超低功耗/低功耗)	0 (第一个采样正确)	1
52 (超低功耗/低功耗)	0 (第一个采样正确)	1
104 (超低功耗/正常)	0 (第一个采样正确)	1
208 (超低功耗/正常)	0 (第一个采样正确)	1
12.5 (高性能)	1	2
26 (高性能)	1	2
52 (高性能)	1	2
104 (高性能)	1	2
208 (高性能)	1	2
417 (高性能)	1	2
833 (高性能)	1	2
1667 (高性能)	4	5
3333 (高性能)	10	10
6667 (高性能)	32	30



表 10. 模式 1/2/3 下的加速度计带宽选择中已经列出了 LPF2 或 HP 数字滤波器使能且带宽不是 ODR/4 时的总稳定时间。

当器件配置为模式 4 时，加速度计 UI 路径滤波链不受加速度计/陀螺仪 OIS 路径滤波链使能/禁用的影响。

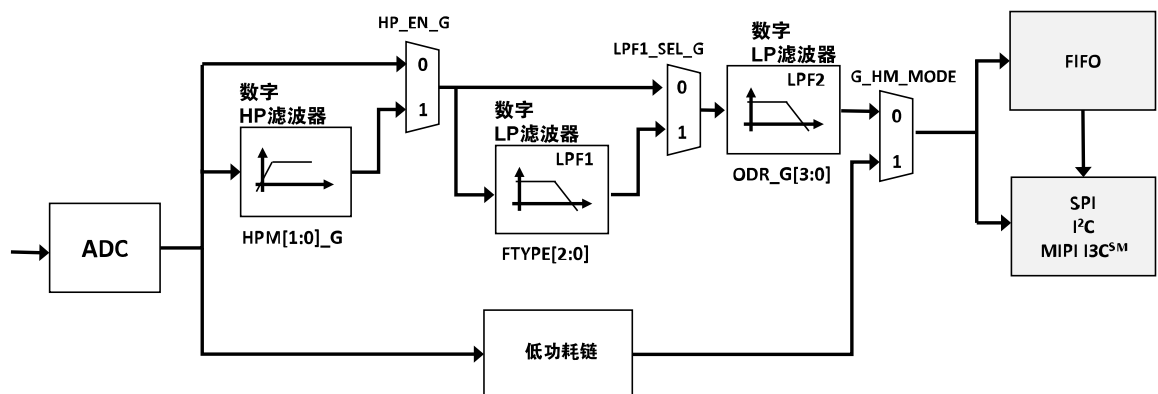
3.10 陀螺仪带宽

陀螺仪滤波链依赖于所使用的连接方式。

当选择模式 1 或模式 2 时，陀螺仪滤波链配置如图 4. 陀螺仪数字链 - 模式 1 和模式 2 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）。

这三个嵌入式滤波器只有在高性能模式下才可用；当器件被配置为低功耗/正常模式时，它们会被绕过。

图 4. 陀螺仪数字链 - 模式 1 和模式 2



在高性能模式下，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。

表 13. 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]	总稳定时间最大值 [s] ⁽¹⁾
00	0.016	45
01	0.065	11
10	0.260	3
11	1.040	0.7

1. 最终值的 99% 时的稳定时间

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_ [2:0] 来选择。

数字 LPF2 滤波器不能由用户配置，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 滤波器被烧开。

下表汇总了在 CTRL4_C 寄存器的 LPF1_SEL_G 位和 CTRL6_C 寄存器的 FTYPE_ [2:0] 字段的不同配置下，不同陀螺仪 ODR 值的总陀螺仪带宽。

表 14. 模式 1/2 下的陀螺仪总带宽选择

陀螺仪 ODR [Hz]	LPF1_SEL_G	FTYPE[2:0]	带宽[Hz] (@ 20 Hz 时的相位延迟)
12.5	0	-	4.2 (-36° @ 1.3 Hz)
	1	0xx	4.2 (-36° @ 1.3 Hz)
	1	100	4.2 (-36° @ 1.3 Hz)
	1	101	4.2 (-36° @ 1.3 Hz)
	1	110	4.1 (-36° @ 1.3 Hz)
	1	111	3.9 (-36° @ 1.3 Hz)
26	0	-	8.3 (-36° @ 2.5 Hz)
	1	0xx	8.3 (-36° @ 2.5 Hz)
	1	100	8.3 (-36° @ 2.5 Hz)
	1	101	8.3 (-36° @ 2.5 Hz)
	1	110	7.8 (-36° @ 2.5 Hz)
	1	111	6.7 (-36° @ 2.5 Hz)
52	0	-	16.6 (-36° @ 5 Hz)
	1	0xx	16.6 (-36° @ 5 Hz)
	1	100	16.7 (-39° @ 5 Hz)
	1	101	16.8 (-43° @ 5 Hz)
	1	110	13.4 (-45° @ 5 Hz)
	1	111	9.7 (-49° @ 5 Hz)
104	0	-	33 (-36° @ 10 Hz)
	1	0xx	33 (-38° @ 10 Hz)
	1	100	33 (-43° @ 10 Hz)
	1	101	31 (-52° @ 10 Hz)
	1	110	19 (-55° @ 10 Hz)
	1	111	11.5 (-64° @ 10 Hz)
208	0	-	66.80 (-35.09°)



陀螺仪 ODR [Hz]	LPF1_SEL_G	FTYPE[2:0]	带宽[Hz] (@ 20 Hz 时的相位延迟)
(Continued) 208	1	0xx	67.00 (-39°)
	1	100	62.40 (-51.03°)
	1	101	43.20 (-68.67°)
	1	110	23.10 (-74.11°)
	1	111	12.20 (-93.61°)
417	0	-	135.90 (-17.81°)
	1	000	136.60 (-22.89°)
	1	001	130.50 (-24.98°)
	1	010	120.30 (-27.38°)
	1	011	137.10 (-21.11°)
	1	100	86.70 (-33.75°)
	1	101	48.00 (-51.39°)
	1	110	24.60 (-56.83°)
	1	111	12.40 (-76.33°)
833	0	-	295.50 (-9.17°)
	1	000	239.20 (-14.25°)
	1	001	192.40 (-16.34°)
	1	010	154.20 (-18.74°)
	1	011	281.80 (-12.47°)
	1	100	96.60 (-25.11°)
	1	101	49.40 (-42.75°)
	1	110	25.00 (-48.19°)
	1	111	12.50 (-67.69°)
1667	0	-	1108.10 (-4.85°)
	1	000	304.20 (-9.93°)
	1	001	220.70 (-12.02°)
	1	010	166.60 (-14.42°)
	1	011	453.20 (-8.15°)
	1	100	99.60 (-20.79°)
	1	101	49.80 (-38.43°)
	1	110	25.10 (-43.87°)
	1	111	12.50 (-63.37°)
3333	0	-	1320.70 (-2.69°)
	1	000	328.50 (-7.77°)
	1	001	229.60 (-9.86°)
	1	010	170.10 (-12.26°)
	1	011	559.20 (-5.99°)
	1	1xx	不可用
6667	0	-	1441.80 (-1.61°)
	1	000	335.50 (-6.69°)
	1	001	232.00 (-8.78°)
	1	010	171.10 (-11.18°)
	1	011	609.00 (-4.91°)



陀螺仪 ODR [Hz]	LPF1_SEL_G	FTYPE[2:0]	带宽[Hz] (@ 20 Hz 时的相位延迟)
(Continued) 6667	1	1xx	不可用

如果陀螺仪配置为低功耗/正常模式，则上述陀螺仪滤波链会被绕开。低功耗/正常模式下的带宽如下表所示。

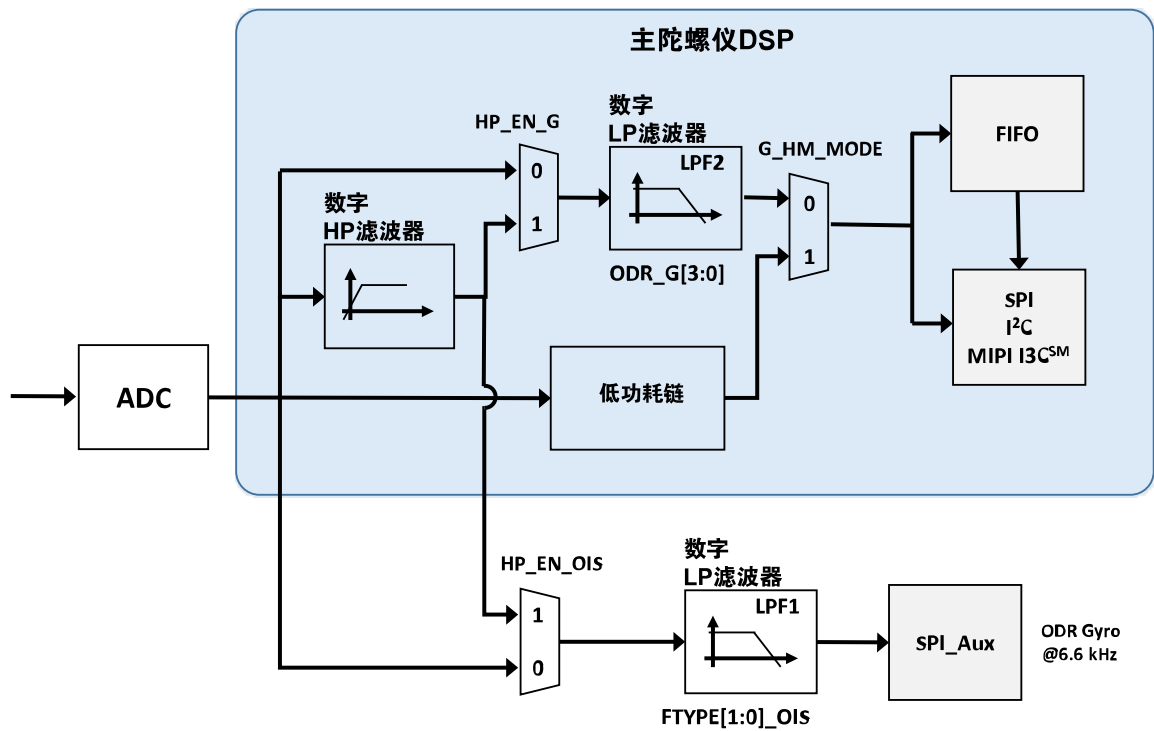
表 15. 陀螺仪低功耗/正常模式下的带宽

陀螺仪 ODR [Hz]	带宽 [Hz]
12.5	4
26	8
52	15.8
104	31.5
208	61.6

当模式 3 或模式 4 使能时，陀螺仪数字链如图 5. 陀螺仪数字链 - 模式 3 和模式 4 所示。在该配置下，有两个不同的数据链：

- 用户接口（UI）链，将陀螺仪数据提供给主 I²C / MIPI I3CSM / SPI，可选 ODR 范围为 12.5 Hz 至 6.66 kHz。
- 光学防抖（OIS）链，将陀螺仪数据提供给辅助 SPI，ODR 固定在 6.66 kHz。

图 5. 陀螺仪数字链 - 模式 3 和模式 4



在模式 3/4 下，LPF1 滤波器专用于 OIS 链；在 UI 侧，如果陀螺仪配置为高性能模式，则总带宽取决于陀螺仪 ODR 值，如表 16. UI 链 - 模式 3/4 下的陀螺仪总带宽选择所示。如果陀螺仪配置为低功耗/正常模式，则陀螺仪链带宽仍如表 15. 陀螺仪低功耗/正常模式下的带宽所示。

表 16. UI 链 - 模式 3/4 下的陀螺仪总带宽选择

陀螺仪 ODR [Hz]	带宽 [Hz]
12.5	4.2
26	8.3
52	16.6
104	33
208	66.8
417	135.9
833	295.5
1667	1108.1
3333	1320.7
6667	1441.8

UI 和 OIS 链共享数字 HP 滤波器，但该滤波器一次只能应用于一条链：

- 如果 CTRL7_G 寄存器的 HP_EN_G 位置为 1，则 HP 滤波器只应用于 UI 链，无论 CTRL2_OIS 寄存器的 HP_EN_OIS 位的值是多少；
- 如果 HP_EN_G 位置为 0 且 HP_EN_OIS 位置为 1，则 HP 滤波器应用于 OIS 链。

注：数字 LPF1 滤波器只能在模式 3/4 使能时在陀螺仪 UI 链上使用。如欲使用模式 3/4，建议避免使用 LPF1 滤波器。

第 8 节 模式 3 和模式 4 - 辅助 SPI 模式中提供了模式 3/4 连接模式和陀螺仪 OIS 链的详细描述。

3.11 陀螺仪开启/关闭时间

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。

当器件配置为模式 1/2 时，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）如表 17. 模式 1/2 下的陀螺仪开启/关闭时间（HP 禁用）所示。

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 17. 模式 1/2 下的陀螺仪开启/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 18 或 表 19
睡眠	低功耗/正常	丢弃 1 个样本
睡眠	高性能	见表 18 或 表 19
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 μ s，如果 XL 和 Gyro 在下电模式中 300 μ s，如果 XL 不在下电模式中

1. @ 最终值的 99% 时的稳定时间

表 18. 模式 1/2 下要丢弃的陀螺仪样本 (LPF1 禁用)

陀螺仪 ODR [Hz]	要丢弃的样本数 ⁽¹⁾
12.5 Hz	2
26 Hz	3
52 Hz	3
104 Hz	3
208 Hz	3
417 Hz	3
833 Hz	3
1.66 kHz	135
3.33 kHz	270
6.66 kHz	540

1. 最终值的 99% 时的建立时间

表 19. 模式 1/2 下的陀螺仪链稳定时间 (LPF1 使能)

FTYPE[2:0]	每种 ODR 的最长稳定时间[ms] ⁽¹⁾
000	3.5
001	4.8
010	6.6
011	2.3
100	11
101	22
110	30
111	60

1. 最终值的 99% 时的稳定时间

当模式变为高性能模式且 HP 滤波器使能或 HP 滤波器开启时，必须将 HP 滤波器稳定时间添加到表 17. 模式 1/2 下的陀螺仪开启/关闭时间 (HP 禁用)。HP 滤波器稳定时间如表 13. 陀螺仪数字 HP 滤波器截止频率选择所示，它与 ODR 无关。

当器件配置为模式 3 或 4 时，陀螺仪 UI 路径滤波链不受陀螺仪 OIS 路径滤波链使能/禁用的影响。



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 10 ms 后，加速度计和陀螺仪自动进入下电模式。

要启用加速度计并通过主 I²C / MIPI I3CSM / SPI 接口采集加速度数据，需要通过 CTRL1_XL 寄存器选择某一种工作模式。

以下通用线序可用来配置加速度计：

1. 写 INT1_CTRL = 01h // INT1 上，Acc 数据准备就绪中断
2. 写 CTRL1_XL = 60h // Acc = 417 Hz（高性能模式）

要启用陀螺仪并通过主 I²C / MIPI I3CSM / SPI 接口采集角速度数据，需要通过 CTRL2_G 来选择某一种工作模式。以下通用序列可用来配置陀螺仪：

1. 写 INT1_CTRL = 02h // INT1 上，Gyro 数据准备就绪中断
2. 写 CTRL2_G = 60h // Gyro = 417 Hz（高性能模式）

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. 读取 STATUS_REG
2. 如果 XLDA = 0，则进入 1
3. 读取 OUTX_L_A
4. 读取 OUTX_H_A
5. 读取 OUTY_L_A
6. 读取 OUTY_H_A
7. 读取 OUTZ_L_A
8. 读取 OUTZ_H_A
9. 数据处理
10. 跳到步骤 1



4.3 使用数据准备就绪信号

该器件可配置为具有一个硬件信号，以确定新的一组测量数据何时可以读取。

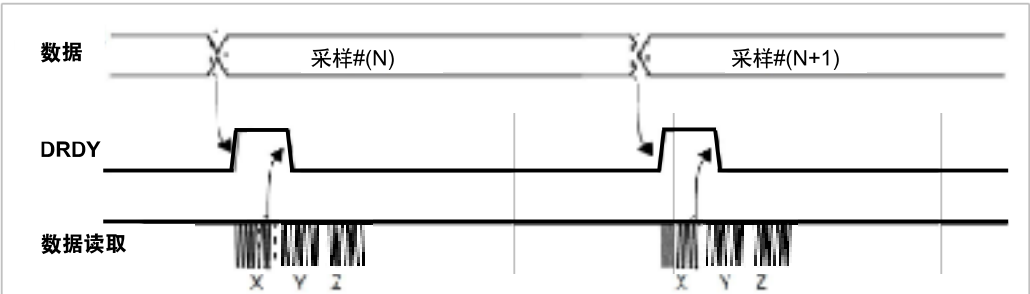
对于加速度计传感器，数据准备就绪信号由 **STATUS_REG** 寄存器的 **XLDA** 位表示。通过将 **INT1_CTRL** 寄存器的 **INT1_DRDY_XL** 位置为 1，可将该信号驱动至 **INT1** 引脚，通过将 **INT2_CTRL** 寄存器的 **INT2_DRDY_XL** 位置为 1，将其驱动至 **INT2** 引脚。

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 **COUNTER_BDR_REG1** 寄存器的 **dataready_pulsed** 位被置为 0（默认值），则数据就绪信号被锁存，并且当其中某一个的高字节寄存器（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 **COUNTER_BDR_REG1** 寄存器的 **dataready_pulsed** 位置为 1，则数据就绪信号是脉冲的，并且在中断引脚上观察到的脉冲持续时间为 75 μ s。脉冲模式不适用于总是锁存模式的 **XLDA** 和 **GDA** 位。



图 6. 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，直至滤波器配置稳定。

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

注：DRDY_MASK 位仅作用于所有加速度计 ODR 的加速度计 LPF1 数字滤波器稳定时间和陀螺仪 ODR ≤ 833 Hz 时的陀螺仪 LPF2 数字滤波器稳定时间。

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_A(G) 和 OUTX_L_A(G)，OUTY_H_A(G) 和 OUTY_L_A(G)，OUTZ_H_A(G) 和 OUTZ_L_A(G)），这对读数不能更新，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。

BDU 功能还作用于 FIFO_STATUS1 和 FIFO_STATUS2 寄存器。当 BDU 位置为 1 时，必须首先读取 FIFO_STATUS1，然后读取 FIFO_STATUS2。

4.5 理解输出数据

测得的加速度数据被发送到 OUTX_H_A、OUTX_L_A、OUTY_H_A、OUTY_L_A、OUTZ_H_A 和 OUTZ_L_A 寄存器。这些寄存器分别包含作用于 X、Y 和 Z 轴的加速度信号的最高有效部分和最低有效部分。

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_A(G) & OUTX_L_A(G)，OUTY_H_A(G) & OUTY_L_A(G)，OUTZ_H_A(G) & OUTZ_L_A(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 输出数据示例

表 20. 输出数据寄存器内容 vs. 加速度 (FS_XL = ±2 g) 提供的几个基本示例中，会在器件受给定加速度影响的情况下读取数据寄存器中的数据。

表 21. 输出数据寄存器内容 vs. 角速度 (FS_G = ±250 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 20. 输出数据寄存器内容 vs. 加速度 (FS_XL = ±2 g)

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



表 21. 输出数据寄存器内容 vs. 角速度 (FS_G = ±250 dps)

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

4.6 加速度计偏移寄存器

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

可通过 CTRL7_G 寄存器 USR_OFF_ON_OUT 位的置位来使能加速度计偏移功能块。在偏移寄存器中设置的偏移值从测得的相应轴加速度值内部减去；内部处理的数据随后被发送到加速计输出寄存器和 FIFO（如果启用）。这些寄存器值以二进制补码的形式表示为 8 位数据，并且必须在[-127,127]的范围内。

应用于偏移寄存器值的权重[g/LSB]独立于加速度计所选的量程，并可利用 CTRL6_C 寄存器的 USR_OFF_W 位进行配置：

- $2^{-10}g/LSB$ ，如果 USR_OFF_W 位置为 0；
- $2^{-6}g/LSB$ ，如果 USR_OFF_W 位置为 1。

4.7 环行功能

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

4.7.1 FIFO 输出寄存器环行

在执行 FIFO 输出寄存器的多字节读取操作时，环行功能自动使能：在读取 FIFO_DATA_OUT_Z_H (7Eh)后，将要读取的下一个寄存器的地址将自动回到 FIFO_DATA_OUT_TAG (78h)，从而允许用户通过一次多字节读取读取许多数据。

4.7.2 传感器输出寄存器环行

可以对其他输出寄存器应用环行功能。

以下输出寄存器组也可以使能环行功能：

- 加速度计输出寄存器，从 OUTX_L_A (28h)到 OUTZ_H_A (2Dh)；
- 陀螺仪输出寄存器，从 OUTX_L_G (22h)到 OUTZ_H_G (27h)；
- 陀螺仪和加速度计输出寄存器，从 OUTX_L_G (22h)到 OUTZ_H_A (2Dh)；

输出寄存器环行模式可利用 CTRL5_C 寄存器的 ROUNDING[1:0]位进行配置，如下表所示。

表 22. 输出寄存器环行模式

ROUNDING[1:0]	环行模式
00	无环行
01	只有加速度计
10	只有陀螺仪
11	陀螺仪 + 加速度计



4.8 DEN（数据使能）

通过使能 CTRL6_C 寄存器中的 TRIG_EN、LVL1_EN 和 LVL2_EN 位，器件可允许外部触发电平识别。有四种不同模式可供选择（参见表 23. 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）输入信号必须在 INT2 引脚上被驱动，当其中一种模式使能时，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 寄存器中的 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 寄存器中的 DEN_LH 位置为 1，可以将其更改为高电平有效。

表 23. DEN 配置

TRIG_EN	LVL1_EN	LVL2_EN	功能	触发类型 (Trigger Type)	动作
0	0	0	数据使能关闭	-	-
1	0	0	边沿感应触发模式	边沿	数据生成
0	1	0	电平感应触发模式	级别	数据加戳
0	1	1	电平感应锁存模式	边沿	数据加戳
1	1	0	电平感应 FIFO 使能模式	级别	FIFO 中的数据生成和加戳

4.8.1

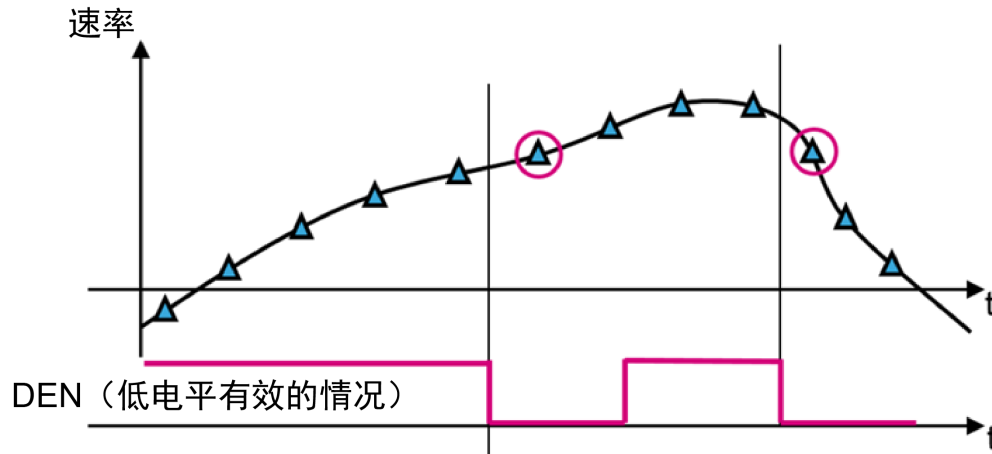
边沿感应触发模式

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

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 7 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 7. 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。GDA 仅与降低采样数据相关，而加速度计输出寄存器和 XLDA 则根据 ODR_XL 进行更新。如果 DEN_XL_EN 位置为 1，则加速度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。

请注意，在更新数据寄存器前会对 DEN 电平进行内部读取：如果在读取后发生电平变化，那么将在下一个 ODR 中确认 DEN。

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并保存在 FIFO 中：在这种配置中，FIFO 存在以下限制：
 - 陀螺仪批处理数据率（FIFO_CTRL3 寄存器的 BDR_GY_[3:0] 位）和陀螺仪输出数据率（CTRL2_G 寄存器的 ODR_G[3:0] 位）必须置为相同值；
 - 禁用配置变化传感器（CFG-Change）（FIFO_CTRL2 寄存器的 ODRCHG_EN 位必须置为 0）；
 - 禁用 FIFO 中的时间戳抽取（FIFO_CTRL4 寄存器的 DEC_TS_BATCH_[1:0] 位必须置为 00b）。
3. 陀螺仪和加速度计处于触发模式并保存在 FIFO 中：在这种配置中，FIFO 存在以下限制：
 - 陀螺仪批处理数据率（FIFO_CTRL3 寄存器的 BDR_GY_[3:0] 位）和陀螺仪输出数据率（CTRL2_G 寄存器的 ODR_G[3:0] 位）必须置为相同值；
 - 加速度计批处理数据率（FIFO_CTRL3 寄存器的 BDR_XL_[3:0] 位）和加速度计输出数据率（CTRL1_XL 寄存器的 ODR_XL[3:0] 位）必须置为相同值；
 - 陀螺仪和加速度计必须设置为相同输出数据率，否则必须将陀螺仪配置为下电模式；
 - 禁用配置变化传感器（CFG-Change）（FIFO_CTRL2 寄存器的 ODRCHG_EN 位必须置为 0）；
 - 禁用 FIFO 中的时间戳抽取（FIFO_CTRL4 寄存器的 DEC_TS_BATCH_[1:0] 位必须置为 00b）。

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，数据在下降沿写入 FIFO 中。

1. 将 44h 写入 FIFO_CTRL3 // 使能 FIFO 中的加速度计和陀螺仪 @ 104 Hz



2. 将 06h 写入 FIFO_CTRL4 // 将 FIFO 设置为 Continuous 模式
// 使能边沿感应触发
3. 将 80h 写入 CTRL6_C // INT2 引脚切换为输入模式 (DEN 信号)
4. 将 E8h 写入 CTRL9_XL // 将 DEN 功能扩展到加速计传感器
// 选择 DEN 有效电平 (低电平有效)
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 寄存器中的 LVL1_EN 位置为 1，并将 CTRL6_C 寄存器中的 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（详情请参见第 4.8.5 节用于 DEN 冲压的 LSB 选择）。

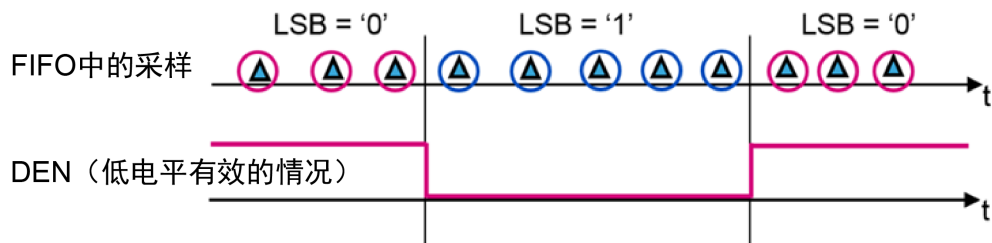
所有数据都可以根据 FIFO 设置存储在 FIFO 中。

请注意，在更新数据寄存器前会对 DEN 电平进行内部读取：如果在读取后发生电平变化，那么将在下一个 ODR 中确认 DEN。

如果通过 CTRL9_XL 寄存器 DEN_XL_EN 位的置位使能了加速度计传感器的 DEN 功能，则加速度计和陀螺仪传感器必须配置为相同 ODR，否则必须将陀螺仪设置为下电模式。

图 8 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。

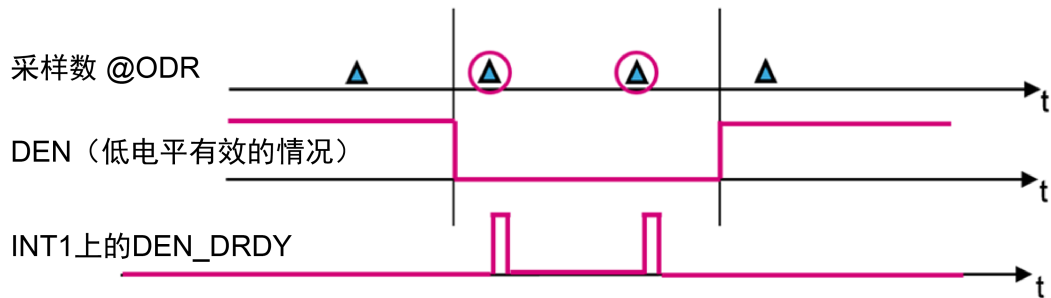
图 8. 电平感应触发模式，DEN 低电平有效



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，INT1_CTRL 寄存器的 DEN_DRDY_flag 位必须置为 1。中断信号可以根据 COUNTER_BDR_REG1 寄存器的 dataready_pulsed 位进行锁存或脉冲。

图 9 显示了当 DEN 电平低（有效状态）时 INT1 上的数据就绪示例。

图 9. 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 寄存器中的 LVL1_EN 和 LVL2_EN 位置为 1，并将 CTRL6_C 寄存器中的 TRIG_EN 位置为 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。

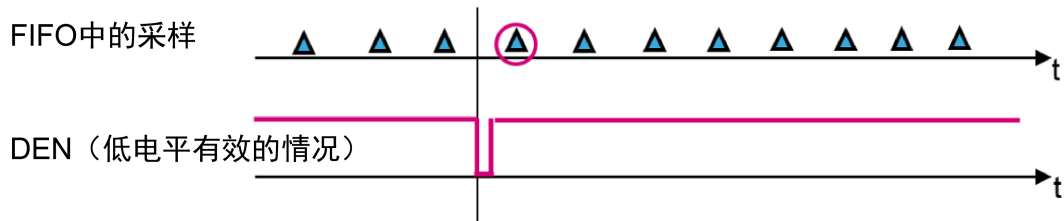
请注意，在更新数据寄存器前会对 DEN 电平进行内部读取：如果在读取后发生电平变化，那么将在下一个 ODR 中确认 DEN。

如果通过 CTRL9_XL 寄存器 DEN_XL_EN 位的置位使能了加速度计传感器的 DEN 功能，则加速度计和陀螺仪传感器必须配置为相同 ODR，否则必须将陀螺仪设置为下电模式。

数据可以通过 CTRL9_XL 中的 DEN_X、DEN_Y、DEN_Z 和 DEN_XL_G 位来选择（详情请参见第 4.8.5 节 用于 DEN 冲压的 LSB 选择）。

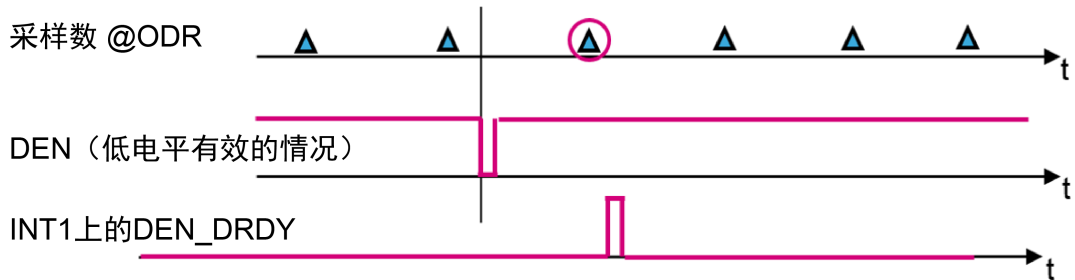
图 10 显示了 DEN 低电平有效时电平感应锁存模式的示例。在 DEN 引脚上的脉冲之后，带红色圆圈的采样在 LSB 位上的值为 1。所有其他样本的 LSB 位都为 0。

图 10. 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 INT1_CTRL 寄存器的 DEN_DRDY_flag 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 11）。

图 11. 电平感应锁存模式，DEN 低电平有效，DEN_DRDY 在 INT1 上



4.8.4 电平感应 FIFO 使能模式

可通过将 CTRL6_C 寄存器中的 TRIG_EN 和 LVL1_EN 位设置为 1，并将 CTRL6_C 寄存器中的 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。

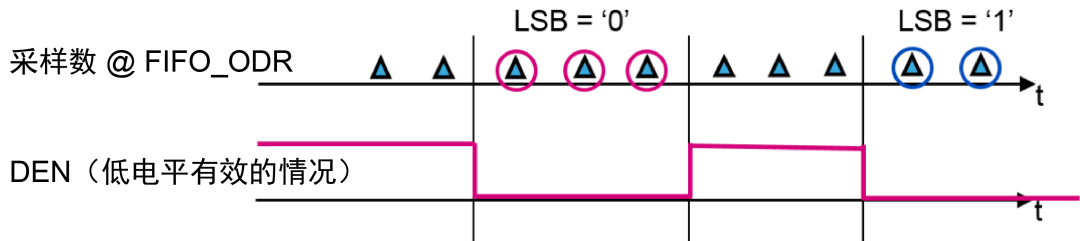
请注意，在更新数据寄存器前会对 DEN 电平进行内部读取：如果在读取后发生电平变化，那么将在下一个 ODR 中确认 DEN。

如果通过 CTRL9_XL 寄存器 DEN_XL_EN 位的置位使能了加速度计传感器的 DEN 功能，则加速度计和陀螺仪传感器必须配置为相同 ODR，否则必须将陀螺仪设置为下电模式。

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 寄存器中的 DEN_X、DEN_Y、DEN_Z 和 DEN_XL_G 位来选择（详情请参见第 4.8.5 节 用于 DEN 冲压的 LSB 选择）。

图 12 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 12. 电平感应 FIFO 使能模式，DEN 低电平有效



在使用电平感应 FIFO 使能模式时，在 FIFO 配置中必须考虑一些限制：

- 陀螺仪批处理数据率（FIFO_CTRL3 寄存器的 BDR_GY_[3:0] 位）和陀螺仪输出数据率（CTRL2_G 寄存器的 ODR_G[3:0] 位）必须置为相同值；
- 如果 CTRL9_XL 寄存器的 DEN_XL_EN 位置为 1，则加速度计批处理数据率（FIFO_CTRL3 寄存器的 BDR_XL_[3:0] 位）和加速度计输出数据率（CTRL1_XL 寄存器的 ODR_XL[3:0] 位）必须置为相同值；
- 禁用配置变化传感器（CFG-Change）（FIFO_CTRL2 寄存器的 ODRCHG_EN 位必须置为 0）；
- 禁用 FIFO 中的时间戳抽取（FIFO_CTRL4 寄存器的 DEC_TS_BATCH_[1:0] 位必须置为 00b）。

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

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/不活动和运动/静止识别。

器件还能够高效运行 **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；
- 计步功能；
- 时间戳。

此外，可以配置器件，以便生成按用户定义的运动模式激活的中断信号。为此，可以单独设定最多 **16** 个嵌入式有限状态机，用于运动检测或手势识别，例如查看、绝对手腕倾斜、摇晃、连续两次摇晃、拿起等。

嵌入式有限状态机功能提供强大的从头开始或从导入 **ST** 直接提供的手势识别程序开始进行定制的能力。请参考有限状态机应用笔记（可从 **ST** 网站 www.st.com 上获取）。

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。

根据数据手册中的描述，在使用 **MIPI I3CSM** 接口时，与触发中断事件的功能相关的信息包含在带内中断（**IBI**）帧中。

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。

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 24. INT1_CTRL 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
DEN_ DRDY_ 标志	INT1_ CNT_ BDR	INT1_ FIFO_ FULL	INT1_ FIFO_ OVR	INT1_ FIFO_TH	INT1_ BOOT	INT1_ DRDY_G	INT1_ DRDY_ XL

- **DEN_DRDY_flag**: **INT1** 上的 **DEN_DRDY** 标记中断
- **INT1_CNT_BDR**: **INT1** 上 **FIFO COUNTER_BDR_IA** 中断
- **INT1_FIFO_FULL**: **INT1** 上 **FIFO** 全满标志中断
- **INT1_FIFO_OVR**: **INT1** 上 **FIFO** 溢出标志中断

- INT1_FIFO_TH: INT1 上 FIFO 阈值中断
- INT1_BOOT: INT1 上启动中断
- INT1_DRDY_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 25. MD1_CFG 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
INT1_ SLEEP_ CHANGE	INT1_ SINGLE_ TAP	INT1_ WU	INT1_ FF	INT1_ DOUBLE_ TAP	INT1_ 6D	INT1_ EMB_ FUNC	INT1_ SHUB

- INT1_SLEEP_CHANGE: INT1 上的活动/不活动识别事件中断
- 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_EMB_FUNC: INT1 上的嵌入功能中断（更多详细信息见第 6 节 嵌入功能）。
- INT1_SHUB: INT1 上的传感器集合端操作中断

表 26. INT2_CTRL 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
0	INT2_ CNT_ BDR	INT2_ FIFO_ FULL	INT2_ FIFO_ OVR	INT2_ FIFO_ TH	INT2_ DRDY_ TEMP	INT2_ DRDY_G	INT2_ DRDY_ XL

- INT2_CNT_BDR: INT2 上 FIFO COUNTER_BDR_IA 中断
- INT2_FIFO_FULL: INT2 上 FIFO 全满标志中断
- INT2_FIFO_OVR: INT2 上 FIFO 溢出标志中断
- INT2_FIFO_TH: INT2 上 FIFO 阈值中断
- INT2_DRDY_TEMP: INT2 上温度数据准备就绪
- INT2_DRDY_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 27. MD2_CFG 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
INT2_ SLEEP_ CHANGE	INT2_ SINGLE_ TAP	INT2_ WU	INT2_ FF	INT2_ DOUBLE_ TAP	INT2_ 6D	INT2_ EMB_ FUNC	INT2_ TIME STAMP

- INT2_SLEEP_CHANGE: INT2 上的活动/不活动识别事件中断
- 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_EMB_FUNC: INT2 上的嵌入功能中断（更多详细信息见第 6 节 嵌入功能）。
- INT2_TIMESTAMP: INT2 上的时间戳溢出报警中断

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

- WAKE_UP_SRC、TAP_SRC 和 D6D_SRC（基本中断功能）
- STATUS_REG（用于数据准备就绪信号）
- EMBD_FUNC_STATUS_MAINPAGE / EMB_FUNC_SRC（用于嵌入功能）
- FSM_STATUS_A_MAINPAGE / FSM_STATUS_A 和 FSM_STATUS_B_MAINPAGE / FSM_STATUS_B（用于有限状态机）
- STATUS_MASTER_MAINPAGE / STATUS_MASTER（用于传感器集合）
- FIFO_STATUS2（用户 FIFO）。

ALL_INT_SRC 寄存器在一个寄存器中汇集了基本中断功能事件状态（6D/4D、自由落体、唤醒、点击、活动/不活动）：可以读取该寄存器，以便为后续特定的源寄存器读取寻址。

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_CFG2 寄存器中的 INTERRUPTS_ENABLE 位使能基本中断。

TAP_CFG0 寄存器的 LIR 位使能基本中断功能的锁存中断：当此位置为 1 且中断标记发送至 INT1 引脚和/或 INT2 引脚时，中断保持激活状态，直至 ALL_INT_SRC 寄存器或相应的源寄存器被读取，它在下一个 ODR 周期复位。仅当功能路由至 INT1 或 INT2 引脚时才对功能使能锁存中断：如果使能了锁存模式，但是中断信号未驱动至中断引脚，那么锁存功能不起作用。

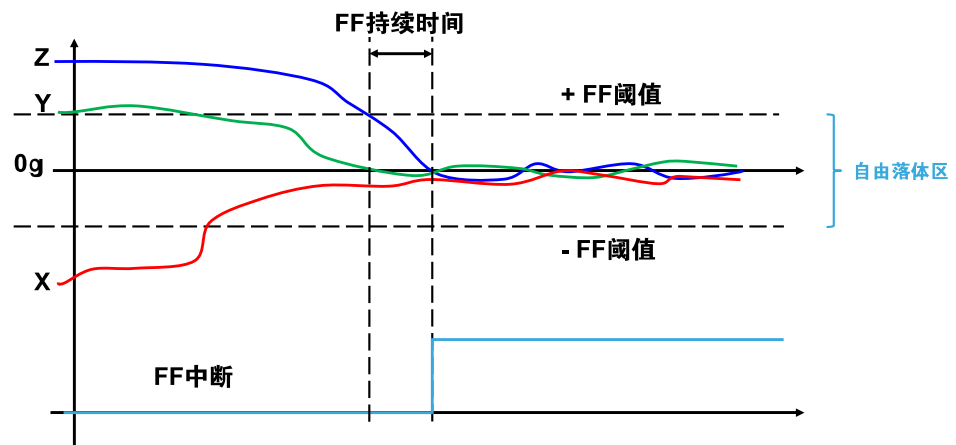
注：如果使能了锁存模式（LIR = 1），不建议连续轮询 ALL_INT_SRC 或专用源寄存器，因为通过读取它们，嵌入功能将实现内部复位；本例中建议同步（有中断事件）读取源寄存器。

在使能锁存模式（LIR=1）后，可以在读取 ALL_INT_SRC（或相关源寄存器）时强制路由至 INT1 或 INT2 引脚的中断信号及其相应中断状态位立即复位。为了执行立即复位，TAP_CFG0 寄存器的 INT_CLR_ON_READ 位必须置为 1。当 INT_CLR_ON_READ 位等于 0 时，在下一个 ODR 周期发生复位。

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 13）。

图 13. 自由落体中断



通过将 TAP_CFG2 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声明了中断引脚时，必须通过读取 WAKE_UP_SRC 或 ALL_INT_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 28. 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 28. 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 = 417 Hz, FS_XL = ±2 g
2. 将 41h 写入 TAP_CFG0 // 使能锁存模式并在读取时复位
3. 将 80h 写入 TAP_CFG2 // 使能中断功能
4. 将 00h 写入 WAKE_UP_DUR // 设置事件持续时间（FF_DUR5 位）
5. 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）
6. 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

唤醒功能可利用斜率滤波器（更多详细信息参见第 3.8.1 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（UI 路径）所示。所用滤波器可通过 TAP_CFG0 寄存器的 SLOPE_FDS 位来选择：如果该位被置为 0（默认值），则使用斜率滤波器；如果被置为 1，则使用 HPF 数字滤波器。此外，可以将唤醒功能配置为在可编程位置执行绝对唤醒。这可以通过将 TAP_CFG0 寄存器的 SLOPE_FDS 位或 WAKE_UP_THS 寄存器的 USR_OFF_ON_WU 位置为 1 来实现。使用此配置时，唤醒功能的输入数据来自于低通滤波器路径，而可编程位置则作为偏移量减去。可编程位置可通过 X_OFS_USR、Y_OFS_USR 和 Z_OFS_USR 寄存器进行配置（更多详细信息见第 4.6 节 加速度计偏移寄存器）。

如果一定数量的连续滤波数据超出了所配置阈值，则产生唤醒中断信号（图 14. 唤醒中断（利用斜率滤波器））。该无符号阈值由 WAKE_UP_THS 寄存器的 WK_THS[5:0] 位来定义；这些 6 位的 1 LSB 值取决于所选加速度计满量程和 WAKE_UP_DUR 寄存器的 WAKE_THS_W 位的值：

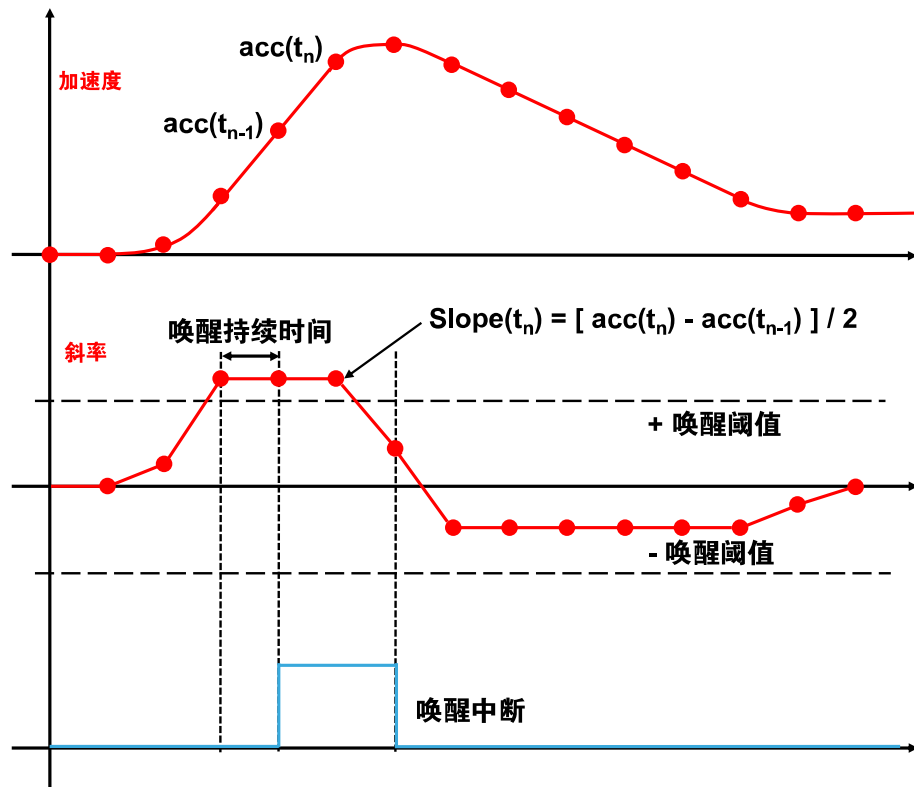
- 如果 WAKE_THS_W = 0，则 $1 \text{ LSB} = \text{FS}_{\text{XL}} / 2^6$ ；
- 如果 WAKE_THS_W = 1，则 $1 \text{ LSB} = \text{FS}_{\text{XL}} / 2^8$ 。

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 $1/\text{ODR}_{\text{XL}}$ 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_CFG2 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 或 ALL_INT_SRC 寄存器的 WU_IA 位对其进行检查。WAKE_UP_SRC 寄存器的 X_WU、Y_WU、Z_WU 位指示哪个轴触发了唤醒事件。

图 14. 唤醒中断（利用斜率滤波器）





如果锁存模式禁用（TAP_CFG0 的 LIR 位置为 0），则当滤波数据低于阈值时，中断信号会自动复位。如果锁存模式使能且唤醒中断信号被驱动至中断引脚，那么当发生唤醒事件且声明了中断引脚时，必须通过读取 WAKE_UP_SRC 寄存器或 ALL_INT_SRC 寄存器来将其复位。X_WU、Y_WU 和 Z_WU 位在执行读取前维持生成中断的状态，并在下一个 ODR 周期释放。如果除了 WU_IA 位还必须评估 WU_X、WU_Y 和 WU_Z 位，建议直接读取 WAKE_UP_SRC 寄存器（此特定情况下不使用 ALL_INT_SRC 寄存器）。如果使能了锁存模式，但是中断信号未驱动至中断引脚，那么锁存功能不起作用。

以下给出使用高通数字滤波器进行唤醒事件识别的基本软件程序。

1. 将 60h 写入 CTRL1_XL // 启动加速度计
// ODR_XL = 417 Hz, FS_XL = $\pm 2 g$
2. 将 51h 写入 TAP_CFG0 // 使能锁存模式（在读取时复位）和数字高通滤波器
3. 将 80h 写入 TAP_CFG2 // 使能中断功能
4. 将 00h 写入 WAKE_UP_DUR // 无持续时间和唤醒阈值权重选择（ $1 \text{ LSB} = \text{FS_XL} / 2^6$ ）
5. 将 02h 写入 WAKE_UP_THS // 设置唤醒阈值
6. 将 20h 写入 MD1_CFG // 唤醒中断驱动至 INT1 引脚

由于持续时间被置为 0，因此每个 X、Y、Z 滤波数据超出所配置阈值时，会生成唤醒中断信号。本例中，WAKE_UP_THS 寄存器的 WK_THS 字段被置为 000010b，因此活动/不活动阈值为 $62.5 \text{ mg} (= 2 * \text{FS_XL} / 2^6)$ 。

如果唤醒功能利用斜率/高通数字滤波器实现，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 3.8.1 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 $(1g - 0)/2 = 500 \text{ mg}$ ，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 // 无持续时间和唤醒阈值权重选择（ $1 \text{ LSB} = \text{FS_XL} / 2^6$ ）
2. 将 02h 写入 WAKE_UP_THS // 设置唤醒阈值
3. 将 51h 写入 TAP_CFG0 // 使能中断并应用斜率滤波器；锁存模式禁用
4. 将 80h 写入 TAP_CFG2 // 使能中断功能
5. 将 70h 写入 CTRL1_XL // 启动加速度计
// ODR_XL = 833 Hz, FS_XL = $\pm 2 g$
6. 等待 4 ms // 插入（减少）等待时间
7. 将 10h 写入 CTRL1_XL // ODR_XL = 12.5 Hz
8. 将 20h 写入 MD1_CFG // 唤醒中断驱动至 INT1 引脚

5.4 6D/4D 定向检测

器件能够检测空间中器件的方向，可以很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 29. 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。下表中给出的阈值对于每个加速度计满量程值均有效。

表 30. 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_CFG2 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声明时，对 D6D_SRC 或 ALL_INT_SRC 寄存器的读取会清除请求，器件将识别另一个不同的方向。XL、XH、YL、YH、ZL 和 ZH 位不受 LIR 配置的影响：在读取 D6D_SRC 寄存器时，它们对应于器件的当前状态。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 15. 6D 识别方向中所示的六种可能情形，表 31. 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 15. 6D 识别方向

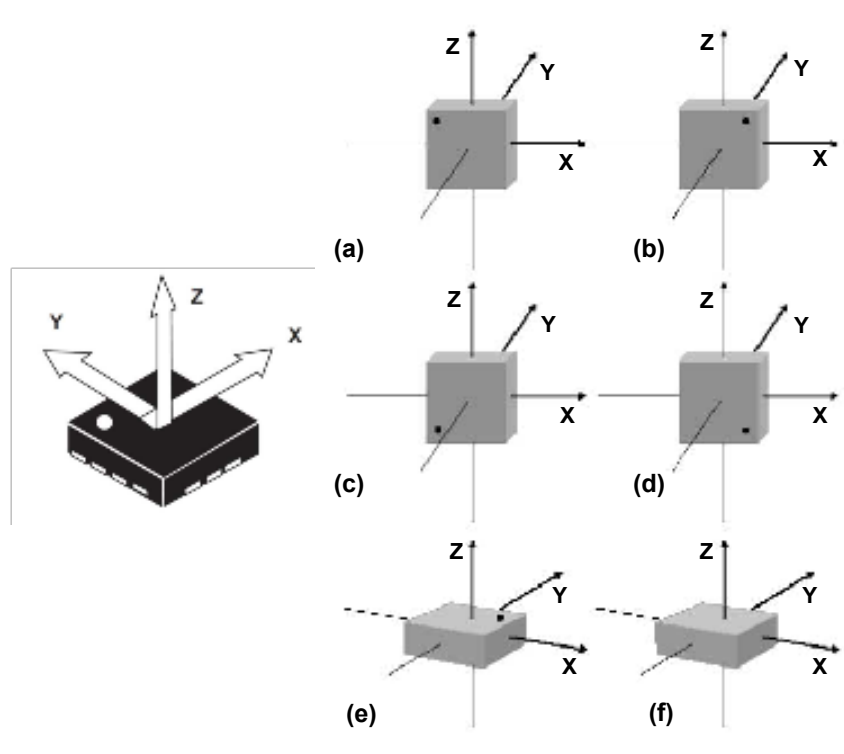


表 31. 6D 定位下的 D6D_SRC 寄存器

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 = 417 Hz, FS_XL = ±2 g
2. 将 41h 写入 TAP_CFG0 // 使能锁存模式并在读取时复位
3. 将 80h 写入 TAP_CFG2 // 使能中断功能
4. 将 40h 写入 TAP_THS_6D // 设置 6D 阈值 (SIXD_THS[1:0] = 10b = 60 degrees)
5. 将 01h 写入 CTRL8_XL // 将 LPF2 滤波器用于 6D 功能
6. 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 31. 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

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。

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。这些功能的推荐最小加速度计 ODR 为 417 Hz。

要启用单击和双击识别功能，必须将 TAP_CFG2 寄存器中的 INTERRUPTS_ENABLE 位置为 1。

5.5.1 单击

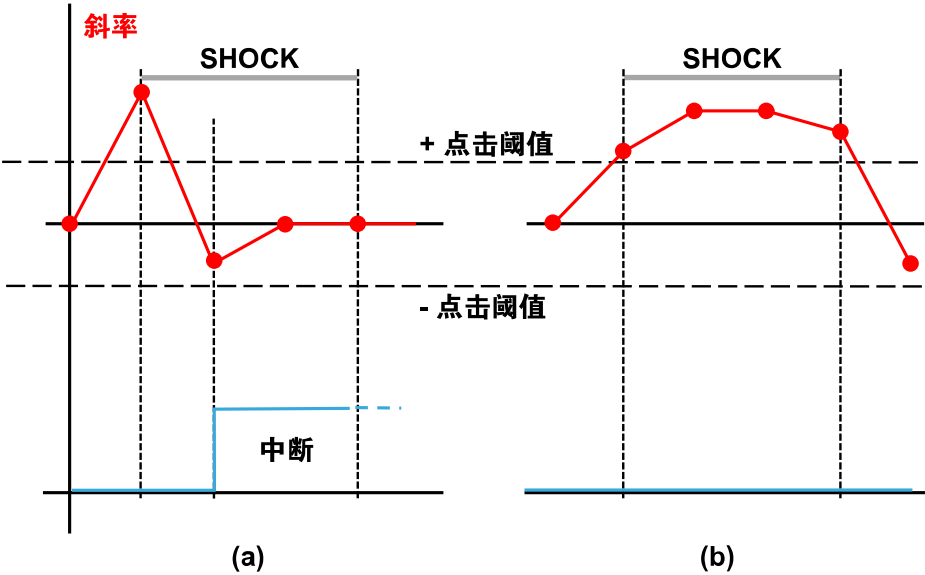
如果器件配置为单击事件检测，那么当所选通道的斜率数据超出了所编程阈值时，会产生一个中断，并在 **Shock** 时间窗口内返回低电平。

在单击情况下，如果 **TAP_CFG0** 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** 或 **ALL_INT_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 16. 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 16. 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**、**Quiet** 和 **Duration** 时间窗口所定义的规则时，才进行第二次点击识别。

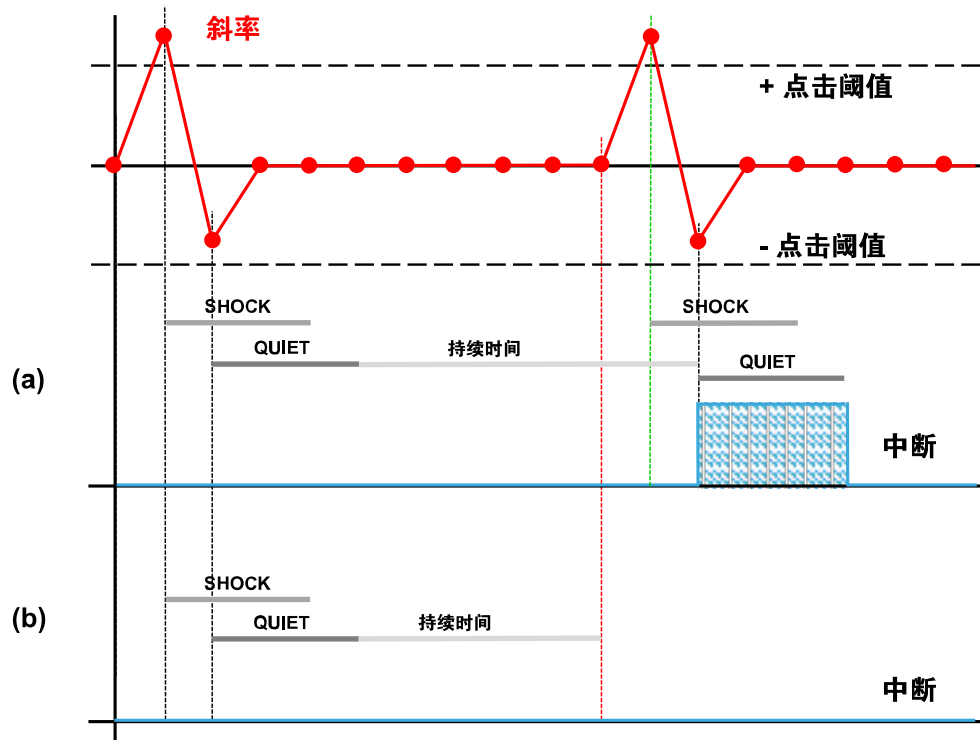
特别的是，识别出第一次敲击后，第二次敲击检测过程会延迟 **Quiet** 时间所定义的时间间隔。这意味着，识别出第一次敲击后，只有在 **Quiet** 窗口之后、**Duration** 窗口结束前，斜率数据超过阈值时，才开始第二次敲击识别过程。**图 17** 的情况(a)中，正确识别出了双击事件，而在情况(b)中，由于在经过窗口间隔之后斜率数据超出了阈值，因此未产生中断。

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_CFG0** 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** 或 **ALL_INT_SRC** 寄存器被读取。

图 17. 双击事件识别 (LIR 位 = 0)



5.5.3 单击和双击识别配置

可对器件进行配置，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_CFG0 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_CFG2 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（每个轴有专用的阈值）和 Shock、Quiet 及 Duration 时间窗。

使用 TAP_CFG1 寄存器的 TAP_THS_X[4:0]位、TAP_CFG2 寄存器的 TAP_THS_Y[4:0]位和 TAP_THS_6D 寄存器的 TAP_THS_Z[4:0]位来选择用于检测相应轴上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。

单击和双击识别功能都只应用于一个轴。如果使能了不止 1 个轴且它们具有各自的阈值，则算法只继续评估具有最高优先级的轴。优先级可通过 TAP_CFG1 寄存器的 TAP_PRIORITY_[2:0]位进行配置。下表显示了所有可能的配置。

表 32. TAP_PRIORITY_[2:0]位的配置

TAP_PRIORITY_[2:0]	最高优先级	中等优先级	最低优先级
000	X	Y	Z
001	Y	X	Z
010	X	Z	Y
011	Z	Y	X
100	X	Y	Z
101	Y	Z	X
110	Z	X	Y
111	Z	Y	X

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。

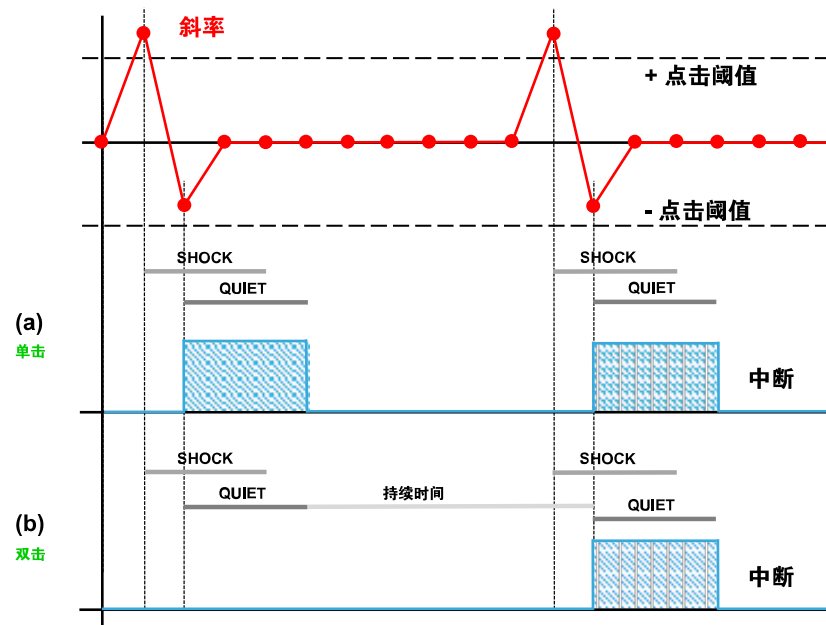
双击情况下，Quiet 时间窗口定义了第一次点击识别后的时间，期间不能发生超阈值。当锁存模式禁用（TAP_CFG 的 LIR 位置为 0）时，Quiet 时间还定义了中断脉冲的长度（单击和双击情况下均如此）。INT_DUR2 寄存器的 QUIET[1:0]位用来设置 Quiet 时间窗口值：这几个位的默认值为 00b，对应于 2/ODR_XL 的时间，这里 ODR_XL 为加速度计输出数据率。如果 QUIET[1:0]位被置为其他不同的值，那么 1 LSB 对应于 4/ODR_XL 的时间。

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 18. 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/不活动和运动/静止识别）。

图 18. 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，如下表所述。

表 33. 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 或 ALL_INT_SRC 寄存器被读取。TAP_SIGN、X_TAP、Y_TAP 和 Z_TAP 位在执行读取前维持生成中断的状态，并在下一个 ODR 周期释放。如果除了 TAP_IA 位还必须评估 TAP_SIGN、X_TAP、Y_TAP 和 Z_TAP 位，建议直接读取 TAP_SRC 寄存器（此特定情况下不使用 ALL_INT_SRC 寄存器）。如果使能了锁存模式，但是中断信号未驱动至中断引脚，那么锁存功能不起作用。

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 = 417 Hz, FS_XL = $\pm 2 g$
2. 将 0Eh 写入 TAP_CFG0 // 使能 X、Y、Z 轴上的点击检测
3. 将 09h 写入 TAP_CFG1 // 设置 X 轴阈值和轴优先级
4. 将 89h 写入 TAP_CFG2 // 设置 Y 轴阈值并使能中断
5. 将 09h 写入 TAP_THS_6D // 设置 Z 轴阈值
6. 将 06h 写入 INT_DUR2 // 设置 Quiet 和 Shock 时间窗口
7. 将 00h 写入 WAKE_UP_THS // 只使能单击 (SINGLE_DOUBLE_TAP = 0)
8. 将 40h 写入 MD1_CFG // 单击中断驱动至 INT1 引脚

在本例中，TAP_THS_X[4:0]、TAP_THS_Y[4:0] 和 TAP_THS_Z[4:0] 位被置为 01001b，因此每个轴的点击阈值为 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 = 417 Hz, FS_XL = $\pm 2 g$

2. 将 0Eh 写入 TAP_CFG0 // 使能 X、Y、Z 轴上的点击检测
3. 将 0Ch 写入 TAP_CFG1 // 设置 X 轴阈值和轴优先级
4. 将 8Ch 写入 TAP_CFG2 // 设置 Y 轴阈值并使能中断
5. 将 0Ch 写入 TAP_THS_6D // 设置 Z 轴阈值
6. 将 7Fh 写入 INT_DUR2 // 设置 Duration、Quiet 和 Shock 时间窗口
7. 将 80h 写入 WAKE_UP_THS // 使能单击和双击 (SINGLE_DOUBLE_TAP = 1)
8. 将 08h 写入 MD1_CFG // 双击中断驱动至 INT1 引脚

在本例中，TAP_THS_X[4:0]、TAP_THS_Y[4:0]和 TAP_THS_Z[4:0]位被置为 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/不活动和运动/静止识别

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

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，器件能够自动将加速度计采样率降低至 12.5 Hz（低功耗模式），并在检测到唤醒中断事件时自动增加加速度计 ODR 和带宽。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_CFG2 寄存器的 INACT_EN[1:0]位，可使能活动/不活动识别功能。如果 TAP_CFG2 寄存器的 INACT_EN[1:0]位等于 00b，则运动/静止嵌入功能使能。下表总结了不活动事件的可能配置。

表 34. 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

可以利用斜率滤波器（更多详细信息见第 3.8.1 节 加速度计斜率滤波器）或高通数字滤波器来实现活动/不活动和运动/静止识别功能，如图 2. 加速度计滤波链（UI 路径）中所示。所用滤波器可通过 TAP_CFG0 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 19. 活动/不活动识别（利用斜率滤波器））。

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

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

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_CFG2 寄存器 INACT_EN[1:0] 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 $512/\text{ODR_XL}$ 的时间，这里 ODR_XL 为加速度计输出数据率。如果 SLEEP_DUR[3:0] 位被置为 0000b，则待识别不活动状态的持续时间等于 $16 / \text{ODR_XL}$ 的时间。

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

当一个轴的滤波数据大于可配置时间的阈值时，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。待识别活动状态的持续时间由 WAKE_UP_DUR 寄存器的 WAKE_DUR[1:0] 位来定义。1 LSB 对应于 $1/\text{ODR_XL}$ 的时间，这里 ODR_XL 为加速度计输出数据率。

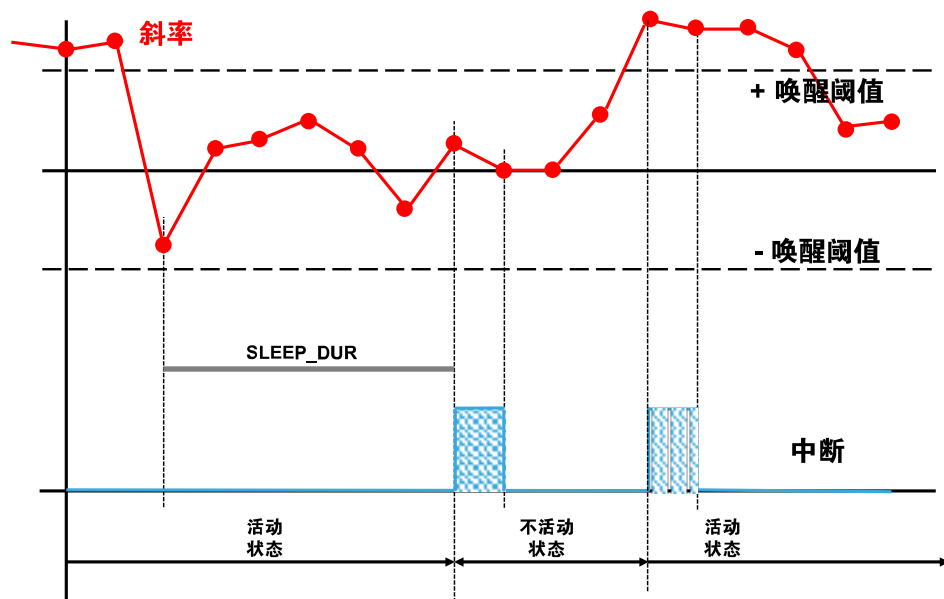
当检测到活动状态时，中断被置为高电平并持续 $1/\text{ODR_XL}[\text{s}]$ 的时间周期，然后自动取消。

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

SLEEP_CHANGE_IA 位默认为脉冲模式。可通过将 TAP_CFG0 寄存器的 LIR 位置为 1 和将 MD1_CFG 寄存器的 INT1_SLEEP_CHANGE 位或 MD2_CFG 寄存器的 INT2_SLEEP_CHANGE 位置为 1 来选择锁存模式。WAKE_UP_SRC 寄存器的 SLEEP_STATE 位不受 LIR 配置的影响：在读取 WAKE_UP_SRC 寄存器时，它对应于器件的当前状态。

通过将 TAP_CFG0 寄存器的 SLEEP_STATUS_ON_INT 位置为 1，将路由至 INT1 或 INT2 引脚的信号配置为活动/不活动状态（WAKE_UP_SRC 寄存器的 SLEEP_STATE 位）而不是睡眠-变化信号：它在不活动状态时为高电平，在活动状态时为低电平。此配置不支持锁存模式。

图 19. 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 = ± 2 g
2. 将 40h 写入 CTRL2_G // 开启陀螺仪
// ODR_G = 104 Hz, FS_G = ± 250 dps
3. 将 02h 写入 WAKE_UP_DUR // 设置不活动检测的持续时间
// 选择活动/不活动阈值分辨率和持续时间
4. 将 02h 写入 WAKE_UP_THS // 设置活动/不活动阈值
5. 将 00h 写入 TAP_CFG0 // 选择睡眠-变化通知
// 选择斜率滤波器
6. 将 E0h 写入 TAP_CFG2 // 使能中断
// 不活动配置：加速度计至 12.5 Hz（LP 模式），
// 陀螺仪至下电模式
7. 将 80h 写入 MD1_CFG // 活动/不活动中断驱动至 INT1 引脚

本例中，WAKE_UP_THS 寄存器的 WK_THS 字段被置为 000010b，因此活动/不活动阈值为 62.5 mg ($= 2 * FS_XL / 2^6$ ，因为 WAKE_UP_DUR 寄存器的 WAKE_THS_W 位被置为 0)。

进行不活动检测前，X、Y、Z 斜率数据必须小于所配置阈值并持续一段时间，该时间由 WAKE_UP_DUR 寄存器的 SLEEP_DUR 字段定义：该字段被置为 0010b，对应 4.92 s ($= 2 * 512 / ODR_XL$)。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。

只要一个采样的（至少）一个轴的斜率数据大于阈值，就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，因为 WAKE_UP_DUR 寄存器的 WAKE_DUR[1:0] 位被配置为 00b。

5.6.1 静止/运动检测

静止/运动检测是“活动/不活动”功能的特殊情况，其中，当检测到睡眠条件（相当于静止条件）时，ODR/电源模式不改变。通过将 TAP_CFG2 寄存器的 INACT_EN[1:0] 位置为 00b 激活静止/运动检测。

5.7 启动状态

器件上电后，执行一段 10 ms（最长）的启动程序来加载修整参数。启动完成后，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。当此位置为 1 时，以下寄存器复位至其默认值：

- FUNC_CFG_ACCESS (01h);
- PIN_CTRL (02h);
- FIFO_CTRL1 (07h) 至 FIFO_CTRL4 (0Ah) ;
- COUNTER_BDR_REG1 (0Bh) 和 COUNTER_BDR_REG2 (0Ch);
- INT1_CTRL (0Dh) 和 INT2_CTRL (0Eh);
- CTRL1_XL (10h) 至 CTRL10_C (19h) ;
- FIFO_STATUS1 (3Ah) 和 FIFO_STATUS2 (3Bh);
- TAP_CFG0 (56h) 至 MD2_CFG (5Fh) ;
- I3C_BUS_AVB (62h);
- X_OFS_USR (73h), Y_OFS_USR (74h) 和 Z_OFS_USR (75h)。

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. 将加速度计和陀螺仪设置为下电模式；
2. 将 INT1_CTRL 寄存器的 INT1_BOOT 位置为 1[可选]；
3. 将 CTRL3_C 寄存器的 BOOT 位置 1；
4. 监控重启状态，有三种可能：
 - a.等待 10 ms；
 - b.监控 INT1 引脚，直至其重新变为 0（本例中必须执行步骤 2）；
 - c.轮询 CTRL3_C 寄存器的 BOOT 位，直至其重新变为 0。

复位流程如下：

1. 将加速度计和陀螺仪设置为下电模式；
2. 将 CTRL3_C 寄存器的 SW_RESET 位设置为 1；
3. 监控软件复位状态，有两种可能：
 - a.等待 50 μ s
 - b.轮询 CTRL3_C 寄存器的 SW_RESET 位，直至其重新变为 0。

为了避免冲突，重启和 sw 复位不能同时执行（不要同时将 CTRL3_C 寄存器的 BOOT 位和 SW_RESET 位同时置为 1）。上述流程必须顺序执行。

6 嵌入功能

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；
- 时间戳。

6.1 计步功能：步伐侦测和步数计算

专用 IP 模块专门实现计步功能：步伐侦测和步数计算。

计步功能的工作频率为 26 Hz 且只基于加速度计传感器；因此，在使用它们时，加速度计 ODR 必须设置为 26 Hz 或更高的值。

为了使能计步功能，必须将 EMB_FUNC_EN_A 嵌入功能寄存器的 PEDO_EN 位置为 1。通过将 EMB_FUNC_INIT_A 嵌入功能寄存器的 STEP_DET_INIT 位置位，可以重新初始化算法内部状态。

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 嵌入功能寄存器输出，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或重新初始化时，步数不会复位至 0；可通过将 EMB_FUNC_SRC 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 10 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）。

去抖动步数可通过嵌入高级功能寄存器中的 PEDO_DEB_STEPS_CONF 寄存器的 DEB_STEP[7:0] 位进行修改：基本上，它对应于在第一次计步器值递增前，要检测到的最小步数。该字段的 1 LSB 对应于 1 步，默认值为 10 步。器件不活动约 1.2 秒后，会重新启动去抖动功能。

可以激活两个额外的功能块以便执行特定条件的实时识别，从而调整计步器操作。可通过将 EMB_FUNC_EN_B 寄存器的 PEDO_ADV_EN 位置为 1 将其激活：

- 假阳性拒绝功能块，可通过将 PEDO_CMD_REG 嵌入高级功能寄存器的 FP_REJECTION_EN 位置为 1 来使能。此功能块基于统计数据实时识别行走活动，并在没有检测到行走活动时禁用计步器。
- 高级检测功能块，可通过将 PEDO_CMD_REG 嵌入高级功能寄存器的 AD_DET_EN 位置为 1 来使能。此功能块可识别低能耗步态（例如，当设备握在手中和步伐起落极小时），并自动切换内部配置，以便更好地针对此情况调整计步器算法。此功能块与假阳性拒绝功能块一起工作，因此 FP_REJECTION_EN 位也必须置为 1。

ST 提供了可从一组具有参考步数的数据记录开始生成特定计步器配置的工具（st.com 上的 Unico GUI）。

EMB_FUNC_SRC 嵌入功能寄存器包含一些与计步器功能状态相关的只读位。

表 35. EMB_FUNC_SRC 嵌入功能寄存器

b7	b6	b5	b4	b3	b2	b1	b0
PEDO_RST_STEP	0	STEP_DETECTED	STEP_COUNT_DELTA_IA	STEP_OVERFLOW	STEP_COUNTER_BIT_SET	0	0

- PEDO_RST_STEP：计步器步伐计数器复位。可以将它置为 1，以将计得的步数复位。在计数器复位后，它自动重置为 0。
- STEP_DETECTED：步伐检测器事件状态。它发出检测到步伐（去抖后）的信号。
- STEP_COUNT_DELTA_IA：如果在特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；通过在 PEDO_SC_DELTAT_H 和 PEDO_SC_DELTAT_L 嵌入高级功能（page 1）寄存器中设置非 00h 的值来定义此时间段。必须将 CTRL10_C 寄存器的 TIMESTAMP_EN 位置为 1（以便使能定时器）。时间段由 PEDO_SC_DELTAT_H 和 PEDO_SC_DELTAT_L 联合定义，表示为 16 位无符号值，精度为 6.4 ms。当在设定时间段内检测到至少一步（去抖后）时，STEP_COUNT_DELTA_IA 变为高电平（在每个时间段结束时）。如果未设定时间段（PEDO_SC_DELTAT = 0），此位维持 0 不变。

- **STEP_OVERFLOW**: 溢出信号，在计步器值达到 2^{16} 时变为高电平。
- **STEPCOUNTER_BIT_SET**: 计步器事件状态。它发出计步器值递增的信号（去抖后）。如果在 **PEDO_SC_DELTAT_H** 和 **PEDO_SC_DELTAT_L** 嵌入高级功能（**page 1**）寄存器中设定了时间段，此位维持 0 不变。

通过读取 **EMB_FUNC_STATUS** 嵌入功能寄存器的 **IS_STEP_DET** 位或 **EMB_FUNC_STATUS_MAINPAGE** 寄存器的 **IS_STEP_DET** 位，还可以对步伐检测中断信号进行检查。

如下表所示，**IS_STEP_DET** 位可以有不同行为，具体取决于 **EMB_FUNC_SRC** 嵌入功能寄存器的 **PEDO_SC_DELTAT** 位和 **PEDO_CMD_REG** 嵌入高级功能寄存器的 **CARRY_COUNT_EN** 位的值。

表 36. **IS_STEP_DET** 配置

PEDO_SC_DELTAT	CARRY_COUNT_EN	IS_STEP_DET
PEDO_SC_DELTAT = 0	0	STEPCOUNTER_BIT_SET
PEDO_SC_DELTAT > 0	0	STEP_COUNT_DELTA_IA
PEDO_SC_DELTAT ≥ 0	1	STEP_OVERFLOW

通过将 **EMB_FUNC_INT1/EMB_FUNC_INT2** 寄存器的 **INT1_STEP_DETECTOR/INT2_STEP_DETECTOR** 位置为 1，可将 **IS_STEP_DET** 中断信号驱动至 **INT1/INT2** 中断引脚。这种情况下，还必须通过 **MD1_CFG/MD2_CFG** 寄存器 **INT1_EMB_FUNC/INT2_EMB_FUNC** 位的置位来使能嵌入功能事件至 **INT1/INT2** 中断引脚的路由。

中断信号的行为默认为脉冲。脉冲持续时间等于 1/26 Hz。可通过将 **PAGE_RW** 嵌入功能寄存器的 **EMB_FUNC_LIR** 位置为 1 使能锁存模式。在这种情况下，通过读取 **EMB_FUNC_STATUS** 嵌入功能寄存器的 **IS_STEP_DET** 位或 **EMB_FUNC_STATUS_MAINPAGE** 寄存器的 **IS_STEP_DET** 位复位中断信号。

计步器可以在 **FIFO** 中进行批处理（详细信息见第 9 节 先进先出（**FIFO**）缓冲区）。

以下是一个基本软件程序，显示如何使能计步器检测功能：

1. 将 80h 写入 **FUNC_CFG_ACCESS** // 使能对嵌入功能寄存器的访问
2. 将 40h 写入 **PAGE_RW** // 选择写操作模式
3. 将 11h 写入 **PAGE_SEL** // 选择 **page 1**
4. 将 83h 写入 **PAGE_ADDR** // 设置要写入的嵌入高级功能寄存器（**PEDO_CMD_REG**）
5. 将 04h 写入 **PAGE_VALUE** // 使能假阳性拒绝功能块（**FP_REJECTION_EN** = 1）
6. 将 00h 写入 **PAGE_RW** // 写操作模式已禁用
7. 将 08h 写入 **EMB_FUNC_EN_A** // 使能计步器
8. 将 10h 写入 **EMB_FUNC_EN_B** // 使能计步器假阳性拒绝功能块和高级检测功能块（**PEDO_ADV_EN** = 1）
9. 将 08h 写入 **EMB_FUNC_INT1** // 步数探测中断驱动至 **INT1** 引脚
10. 将 00h 写入 **FUNC_CFG_ACCESS** // 禁用对嵌入功能寄存器的访问
11. 将 02h 写入 **MD1_CFG** // 使能嵌入功能中断路由
12. 将 28h 写入 **CTRL1_XL** // 启动加速度计
// **ODR_XL** = 26 Hz, **FS_XL** = ±4 g



6.2 大幅运动检测

当检测到可能是由于用户位置变化引起的“大幅运动”时，大幅运动功能会产生一个中断。在设备中，在只使用加速度的硬件中实现此功能：

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。它在初始化/复位后计得的步数超过 10 步时生成中断。在生成中断后，算法内部状态复位。

为了使能大幅运动检测功能，必须将 EMB_FUNC_EN_A 嵌入功能寄存器的 SIGN_MOTION_EN 位置为 1。通过将 EMB_FUNC_INIT_A 嵌入功能寄存器的 SIG_MOT_INIT 位置位，可以重新初始化算法。

注：大幅运动功能自动使能内部计步器算法。

通过将 EMB_FUNC_INT1/EMB_FUNC_INT2 寄存器的 INT1_SIG_MOT/INT2_SIG_MOT 位置为 1，可将大幅运动中断信号驱动至 INT1/INT2 中断引脚。这种情况下，还必须通过 MD1_CFG/MD2_CFG 寄存器 INT1_EMB_FUNC/INT2_EMB_FUNC 位的置位来使能嵌入功能事件至 INT1/INT2 中断引脚的路由。

通过读取 EMB_FUNC_STATUS 嵌入功能寄存器的 IS_SIGMOT 位或 EMB_FUNC_STATUS_MAINPAGE 寄存器的 IS_SIGMOT 位，还可以对大幅运动中断信号进行检查。

大幅运动中断信号的行为默认为脉冲。脉冲持续时间等于 1/26 Hz。锁存模式可通过将 PAGE_RW 嵌入功能寄存器的 EMB_FUNC_LIR 位置为 1 来使能：这种情况下，通过读取 EMB_FUNC_STATUS 嵌入功能寄存器的 IS_SIGMOT 位或 EMB_FUNC_STATUS_MAINPAGE 寄存器的 IS_SIGMOT 位复位中断信号。

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 FUNC_CFG_ACCESS | // 使能对嵌入功能寄存器的访问 |
| 2. 将 20h 写入 EMB_FUNC_EN_A | // 使能大幅运动检测 |
| 3. 将 20h 写入 EMB_FUNC_INT1 | // 大幅运动中断驱动至 INT1 引脚 |
| 4. 将 80h 写入 PAGE_RW | // 为嵌入功能使能锁存模式 |
| 5. 将 00h 写入 FUNC_CFG_ACCESS | // 禁用对嵌入功能寄存器的访问 |
| 6. 将 02h 写入 MD1_CFG | // 使能嵌入功能中断路由 |
| 7. 将 20h 写入 CTRL1_XL | // 启动加速度计 |
| | // ODR_XL = 26 Hz, FS_XL = ±2 g |

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）。在设备中，在只使用加速度计的硬件中实现此功能。

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。

为了使能相对倾斜度检测功能，必须将 `EMB_FUNC_EN_A` 嵌入功能寄存器的 `TILT_EN` 位置为 1。通过将 `EMB_FUNC_INIT_A` 嵌入功能寄存器的 `TILT_INIT` 位置位，可以重新初始化算法。

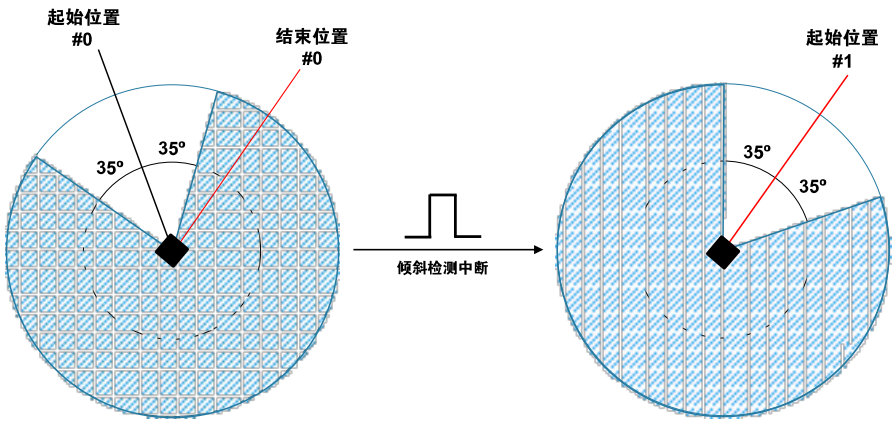
如果器件配置为用于倾斜度检测，那么当器件距起始位置的倾斜角度大于 35 度时，会产生一个中断。起始位置定义为倾斜检测使能/重新初始化时器件的位置，或上一次倾斜中断产生时的器件位置。

在使能或重新初始化此功能后，倾斜逻辑通常需要 2 秒钟的稳定时间，然后才能生成第一个中断。

在图 20. 倾斜度检测所示的示例中，在器件方向对应于“起始位置#0”时使能倾斜度检测功能。如果器件偏离起始位置的角度大于 35 度，将产生第一个中断。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 20. 倾斜度检测





通过将 EMB_FUNC_INT1/EMB_FUNC_INT2 寄存器的 INT1_TILT/INT2_TILT 位置为 1，可将倾斜中断信号驱动至 INT1/INT2 中断引脚。这种情况下，还必须通过 MD1_CFG/MD2_CFG 寄存器 INT1_EMB_FUNC/INT2_EMB_FUNC 位的置位来使能嵌入功能事件至 INT1/INT2 中断引脚的路由。

通过读取 EMB_FUNC_STATUS 嵌入功能寄存器的 IS_TILT 位或 EMB_FUNC_STATUS_MAINPAGE 寄存器的 IS_TILT 位，还可以对倾斜中断信号进行检查。

倾斜中断信号的行为默认为脉冲。脉冲持续时间等于 1/26 Hz。可通过将 PAGE_RW 嵌入功能寄存器的 EMB_FUNC_LIR 位置为 1 使能锁存模式。在这种情况下，通过读取 EMB_FUNC_STATUS 嵌入功能寄存器的 IS_TILT 位或 EMB_FUNC_STATUS_MAINPAGE 寄存器的 IS_TILT 位复位中断信号。

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 80h 写入 FUNC_CFG_ACCESS | // 使能对嵌入功能寄存器的访问 |
| 2. 将 10h 写入 EMB_FUNC_EN_A | // 使能倾斜度检测 |
| 3. 将 10h 写入 EMB_FUNC_INT1 | // 倾斜中断驱动至 INT1 引脚 |
| 4. 将 80h 写入 PAGE_RW | // 为嵌入功能使能锁存模式 |
| 5. 将 00h 写入 FUNC_CFG_ACCESS | // 禁用对嵌入功能寄存器的访问 |
| 6. 将 02h 写入 MD1_CFG | // 使能嵌入功能中断路由 |
| 7. 将 20h 写入 CTRL1_XL | // 启动加速度计 |
| | // ODR_XL = 26 Hz, FS_XL = ±2 g |

6.4 时间戳

器件能够提供与传感器数据对应的的时间戳信息。

为了使能该功能，必须将 CTRL10_C 寄存器的 TIMESTAMP_EN 位置 1。时间步数由 TIMESTAMP3 & TIMESTAMP2 & TIMESTAMP1 & TIMESTAMP0 寄存器输出，表示为一个 32 位无符号数字。

标称时间戳精度为 25 μs。可通过 INTERNAL_FREQ_FINE 寄存器的 FREQ_FINE[7:0] 位获取实际时间戳精度值，其中包含实际 ODR（和时间戳速率）相对于标称值的百分比差值。

$$t_{actual}[\mu s] = \frac{1}{40000 \cdot (1 + 0.0015 \cdot FREQ_FINE)}$$

同样地，可以使用以下公式获取实际输出数据率：

$$ODR_{actual}[Hz] = \frac{6667 + 0.0015 \cdot FREQ_FINE \cdot 6667}{ODR_{coeff}}$$

其中的 ODR_{coeff} 值如下表所示。

表 37. ODR_{coeff} 值

选择的 ODR [Hz]	ODR _{coeff}
12.5	512
26	256
52	128
104	64
208	32
417	16
833	8
1667	4
3333	2
6667	1

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且时间戳值冻结为最后的值。

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

在时间戳溢出条件发生前 6.4 ms，ALL_INT_SRC 的 TIMESTAMP_ENDCOUNT 位变为高电平。在读取 ALL_INT_SRC 寄存器时，此标记复位。还可以通过将 MD2_CFG 寄存器的 INT2_TIMESTAMP 位置为 1 将此信号路由至 INT2 引脚（75 μs 持续时间脉冲）。

时间戳可以在 FIFO 中进行批处理（详细信息见第 9 节 先进先出（FIFO）缓冲区）。

7 模式 2 - 传感器集合 (sensor hub) 模式

LSM6DSO 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/MIPI I3CSM/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器件的 I²C 主接口。传感器集合触发信号可以与加速度计/陀螺仪数据就绪信号 (最高 104 Hz) 同步。使用这种配置时，可通过 SLAVE0_CONFIG 寄存器的 SHUB_ODR_[1:0] 位配置传感器集合 ODR。或者，可以将连接到 INT2 引脚的外部信号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。

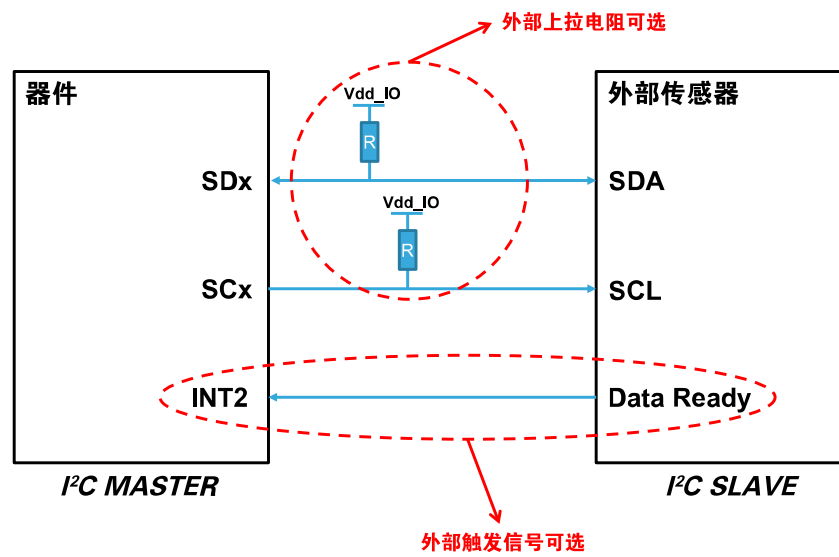
在传感器集合触发信号上，通过寄存器 SLVx_ADD、SLVx_SUBADD、SLAVEx_CONFIG 和 DATAWRITE_SLV0 配置的所有写和读 I²C 操作都是从外部传感器 0 到外部传感器 3 顺序执行 (取决于由 MASTER_CONFIG 寄存器中 AUX_SENS_ON[1:0] 字段所使能的外部传感器)。

外部传感器数据也可存储在 FIFO 中 (详细信息见第 9 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 21. 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 21. 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

当 FUNC_CFG_ACCESS 寄存器的 SHUB_REG_ACCESS 位置为 1 时，可以访问传感器集合配置寄存器和输出寄存器。在 SHUB_REG_ACCESS 位置为 1 后，只有传感器集合寄存器可供使用。为了保证其他操作的正确寄存器映射，在传感器集合配置或输出数据读取后，必须将 FUNC_CFG_ACCESS 寄存器的 SHUB_REG_ACCESS 位置为 0。

必须使用 MASTER_CONFIG 寄存器进行 I²C 主接口的配置。

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。额外的寄存器 DATAWRITE_SLV0 只与从器件#0 有关。必须使用它实现写操作。

最后，有 18 个寄存器（从 SENSOR_HUB_1 到 SENSOR_HUB_18）可用来存储从外部传感器读取的数据。

7.2.1 MASTER_CONFIG (14h)

此寄存器用于配置 I²C 主控制器。

表 38. MASTER_CONFIG 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
RST_MASTER_REGS	WRITE_ONCE	START_CONFIG	PASS_THROUGH_MODE	SHUB_PU_EN	MASTER_ON	AUX_SENS_ON1	AUX_SENS_ON0

- RST_MASTER_REGS 位用于复位 I²C 主接口、配置和输出寄存器。它必须手动置位和取消置位。
- 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：如果 slave 0 用于读取，则 WRITE_ONCE 位必须置为 1。

- 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/陀螺仪传感器必须激活（不能处于下电模式），传感器集合 (sensor hub) 触发信号为加速度计/陀螺仪数据准备就绪信号，其频率由 SLAVE0_CONFIG 寄存器的 SHUB_ODR[1:0] 位定义（最高 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，并且传感器集合触发信号路由至 INT2 引脚。实际上，当 MASTER_ON 位和 START_CONFIG 位均置为 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引脚（图 21. 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引脚（图 21. 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。
- 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。
 - SHUB_PU_EN 位使能/禁用 I²C 主线上的内部上拉。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 21. 模式 2 下外部传感器连接所示。当此位置为 1 时，内部上拉使能（无论 MASTER_ON 位的配置是什么），SDx/SCx 引脚上不需要外部上拉电阻。
 - MASTER_ON 位必须置为 1，以使能器件的辅助 I²C 主线（传感器集合 (sensor hub) 模式）。为了在运行时间或在加速度计和陀螺仪传感器设置为下电模式时或在应用软件复位程序时修改传感器集合配置，必须禁用 I²C 主器件，然后等待 300 μs。必须执行以下程序：
 1. 通过将 MASTER_ON 位置为 0 关闭 I²C 主器件。
 2. 等待 300 μs。
 3. 修改传感器集合寄存器的配置或将加速度计/陀螺仪设置为下电模式或应用软件复位程序。

- 必须根据要使用的从器件数量设置 **AUX_SENS_ON[1:0]** 位。按顺序从 slave 0 到 slave 3 执行 I²C 事务。可能的值为：
 - 00b: 一个从器件；
 - 01b: 两个从器件；
 - 10b: 三个从器件；
 - 11b: 四个从器件。

7.2.2

STATUS_MASTER (22h)

类似于其他传感器集合配置和输出寄存器，**STATUS_MASTER** 寄存器只能在 **FUNC_CFG_ACCESS** 寄存器的 **SHUB_REG_ACCESS** 位置为 1 后读取。**STATUS_MASTER** 寄存器还映射到 **STATUS_MASTER_MAINPAGE** 寄存器，该寄存器可直接读取，无需使能对传感器集合寄存器的访问。

表 39. STATUS_MASTER / STATUS_MASTER_MAINPAGE 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
WR_ONCE_DONE	SLAVE3_NACK	SLAVE2_NACK	SLAVE1_NACK	SLAVE0_NACK	0	0	SENS_HUB_ENDOP

- 在 **MASTER_CONFIG** 寄存器的 **WRITE_ONCE** 位配置为 1 的情况下执行写操作后，**WR_ONCE_DONE** 位被置为 1。为了检查单个写事务是否已经完成，可以轮询此位。
- 如果在与相应从线 x 的通信过程中发生了“未确认”事件，则 **SLAVE_x_NACK** 位会被置为 1。
- **SENS_HUB_ENDOP** 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 **SENSOR_HUB_x** 寄存器读取外部传感器数据（取决于 **SLV_x_ADD**、**SLV_x_SUBADD** 和 **SLAVE_x_CONFIG** 寄存器的配置）。

通过将 **MD1_CFG** 寄存器的 **INT1_SHUB** 位置为 1，可将 I²C 主器件的状态信号驱动至 **INT1** 中断引脚。此信号在 **SENS_HUB_ENDOP** 信号的上升沿变为高电平，并且只有在 **STATUS_MASTER / STATUS_MASTER_MAINPAGE** 寄存器被读取时才清零。

7.2.3

SLV0_ADD (15h), SLV0_SUBADD (16h), SLAVE0_CONFIG (17h)

传感器集合寄存器用于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 40. 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。

表 41. 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 42. SLAVE0_CONFIG 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
SHUB_ODR_1	SHUB_ODR_0	0	0	BATCH_EXT_SENS_0_EN	Slave0_numop2	Slave0_numop1	Slave0_numop0

- SHUB_ODR_[1:0]位用于配置使用内部触发信号（加速度计/陀螺仪数据准备就绪信号）时的传感器集合输出数据率。可以将传感器集合输出数据率配置为四个可能值（受加速度计和陀螺仪传感器 ODR 的限制）：
 - 00b: 104 Hz;
 - 01b: 52 Hz;
 - 10b: 26 Hz;
 - 11b: 12.5 Hz。

SHUB_ODR_[1:0]位的最大允许值对应于加速度计和陀螺仪传感器中的最大 ODR。

- BATCH_EXT_SENS_0_EN 位用于使能与 slave0 相关的外部传感器 FIFO 中的批处理。
- 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。

7.2.4

SLV1_ADD (18h), SLV1_SUBADD (19h), SLAVE1_CONFIG (1Ah)

传感器集合寄存器用于配置关联到第二个外部传感器的 I²C 从接口，下面对此进行介绍。

表 43. 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 44. 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 45. SLAVE1_CONFIG 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	BATCH_EXT_SENS_1_EN	Slave1_numop2	Slave1_numop1	Slave1_numop0

- BATCH_EXT_SENS_1_EN 位用于使能与 slave1 相关的外部传感器 FIFO 中的批处理。

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。

7.2.5

SLV2_ADD (1Bh), SLV2_SUBADD (1Ch), SLAVE2_CONFIG (1Dh)

传感器集合寄存器用于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 46. 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 47. 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 48. SLAVE2_CONFIG 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	BATCH_EXT_SENS_2_EN	Slave2_numop2	Slave2_numop1	Slave2_numop0

- BATCH_EXT_SENS_2_EN 位用于使能与 slave2 相关的外部传感器 FIFO 中的批处理。
- 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。

7.2.6

SLV3_ADD (1Eh), SLV3_SUBADD (1Fh), SLAVE3_CONFIG (20h)

传感器集合寄存器用于配置关联到第四个外部传感器的 I²C 从接口，下面对此进行介绍。

表 49. 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 50. 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 51. SLAVE3_CONFIG 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	BATCH_EXT_SENS_3_EN	Slave3_numop2	Slave3_numop1	Slave3_numop0

- BATCH_EXT_SENS_3_EN 位用于使能与 slave3 相关的外部传感器 FIFO 中的批处理。

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

DATAWRITE_SLV0 (0Eh)

表 52. DATAWRITE_SLV0 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
Slave0_dataw7	Slave0_dataw6	Slave0_dataw5	Slave0_dataw4	Slave0_dataw3	Slave0_dataw2	Slave0_dataw1	Slave0_dataw0

- 当 SLV0_ADD 寄存器的 rw_0 位被置为 0（写操作）时，Slave0_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。

7.2.8

SENSOR_HUB_x 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 MASTER_CONFIG 寄存器的 AUX_SENS_ON[1:0]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）器件寄存器中，从 SENSOR_HUB_1 寄存器开始，如图 22. SENSOR_HUB_X 配置示例中示例所示；18 个寄存器，从 SENSOR_HUB_1 到 SENSOR_HUB_18，可以用来存储从外部传感器中读取的数据。

图 22. SENSOR_HUB_X 配置示例



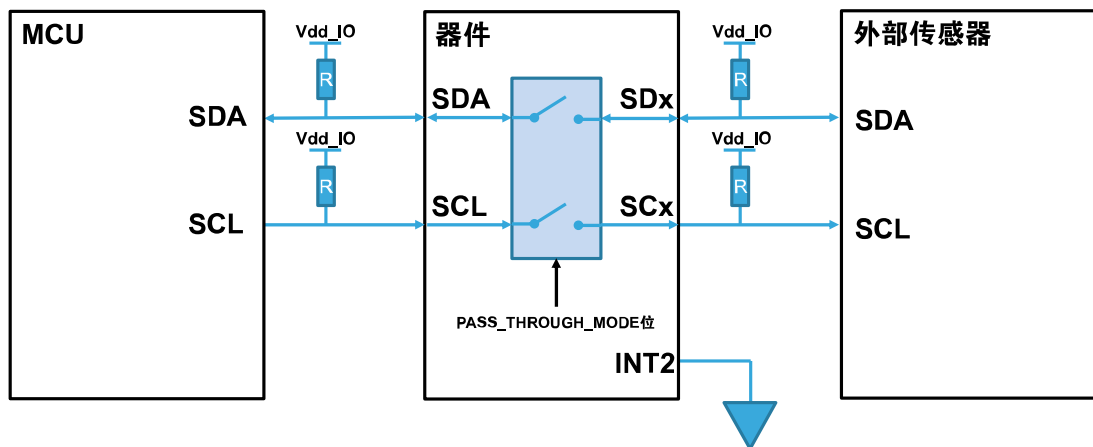
传感器#1	{ SLV0_SUBADD (16h) = 28h SLAVE0_CONFIG (17h) – Slave0_numop[2:0] = 3	SENSOR_HUB_1	Value of reg 28h	传感器#1
		SENSOR_HUB_2	Value of reg 29h	
		SENSOR_HUB_3	Value of reg 2Ah	
传感器#2	{ SLV1_SUBADD (19h) = 00h SLAVE1_CONFIG (1Ah) – Slave1_numop[2:0] = 6	SENSOR_HUB_4	Value of reg 00h	传感器#2
		SENSOR_HUB_5	Value of reg 01h	
		SENSOR_HUB_6	Value of reg 02h	
传感器#3	{ SLV2_SUBADD (1Ch) = 20h SLAVE2_CONFIG (1Dh) – Slave2_numop[2:0] = 4	SENSOR_HUB_7	Value of reg 03h	传感器#3
		SENSOR_HUB_8	Value of reg 04h	
		SENSOR_HUB_9	Value of reg 05h	
传感器#4	{ SLV3_SUBADD (1Fh) = 40h SLAVE3_CONFIG (20h) – Slave3_numop[2:0] = 5	SENSOR_HUB_10	Value of reg 20h	传感器#3
		SENSOR_HUB_11	Value of reg 21h	
		SENSOR_HUB_12	Value of reg 22h	
		SENSOR_HUB_13	Value of reg 23h	传感器#4
		SENSOR_HUB_14	Value of reg 40h	
		SENSOR_HUB_15	Value of reg 41h	
		SENSOR_HUB_16	Value of reg 42h	传感器#4
		SENSOR_HUB_17	Value of reg 43h	
		SENSOR_HUB_18	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。只有在主接口上使用 I²C 协议时，才能使用用于外部器件配置的直连功能。此功能可用于配置外部传感器。

图 23. 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)，且全通功能使用。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，这些功能在第 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 (内部触发)。不能使用直连功能。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 SHUB_PU_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 SHUB_PU_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

可以使用直连功能执行外部传感器的配置。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器件配置为传感器集合 (sensor hub) 模式的基本程序。下面提供了三个不同的代码片段, 目的是展示如何简单地使用 slave 0 执行一次性写或读操作, 以及如何设置 slave 0 以便连续地读取外部传感器数据。

为了尽可能通用, 所有这些程序中都禁用了 PASS_THROUGH_MODE 位。

下面描述的是一次性读取程序 (使用内部触发信号)。为简单起见, 该程序使用了频率为 104 Hz 的加速度计, I²C 辅助总线上无外部上拉电阻。

- | | |
|--------------------------------------|---|
| 1. 将 40h 写入 FUNC_CFG_ACCESS | // 使能对传感器集合寄存器的访问 |
| 2. 将 EXT_SENS_ADDR 01h 写入 SLV0_ADD | // 配置外部器件地址 (EXT_SENS_ADDR) |
| | // 使能读操作 (rw_0 = 1) |
| 3. 将 REG 写入 SLV0_SUBADD | // 配置要读取的寄存器的地址 (REG) |
| 4. 将 01h 写入 SLAVE0_CONFIG | // 读取一个字节, SHUB_ODR = 104 Hz |
| 5. 将 4Ch 写入 MASTER_CONFIG | // 强制读取 WRITE_ONCE |
| | // I ² C 主器件使能, 只使用 slave 0 |
| | // SDx 和 SCx 上的 I ² C 上拉电阻使能 |
| 6. 将 00h 写入 FUNC_CFG_ACCESS | // 禁用对传感器集合寄存器的访问 |
| 7. 读取 OUTX_H_A 寄存器 | // 将加速度计数据准备就绪 XLDA 清零 |
| 8. 轮询 STATUS_REG, 直至 XLDA = 1 | // 等待传感器集合触发信号 |
| 9. 轮询 STATUS_MASTER_MAINPAGE, | // 等待传感器集合读事务 |
| 直至 SENS_HUB_ENDOP = 1 | |
| 10. 将 40h 写入 FUNC_CFG_ACCESS | // 使能对传感器集合寄存器的访问 |
| 11. 将 08h 写入 MASTER_CONFIG | // I ² C 主器件禁用 |
| 12. 等待 300 μs | |
| 13. 读取 SENSOR_HUB_1 寄存器 | // 检索读操作的输出 |

14. 将 00h 写入 FUNC_CFG_ACCESS // 禁用对传感器集合寄存器的访问

可以简单地修改一次性程序，以便将器件设置为连续读取外部传感器数据：

1. 将 40h 写入 FUNC_CFG_ACCESS // 使能对传感器集合寄存器的访问
2. 将 EXT_SENS_ADDR | 01h 写入 SLV0_ADD // 配置外部器件地址 (EXT_SENS_ADDR)
// 使能读操作 (rw_0 = 1)
3. 将 REG 写入 SLV0_SUBADD // 配置要读取的寄存器的地址 (REG)
4. 将 0xh 写入 SLAVE0_CONFIG // 读取 x 个字节 (最多 6 个), SHUB_ODR = 104 Hz
5. 将 4Ch 写入 MASTER_CONFIG // 强制读取 WRITE_ONCE
// I²C 主器件使能, 只使用 slave 0
// SDx 和 SCx 上的 I²C 上拉电阻使能
6. 将 00h 写入 FUNC_CFG_ACCESS // 禁用对传感器集合寄存器的访问

在执行步骤 6 后, 传感器集合输出寄存器中的外部传感器数据可供读取。

下面描述的是一次性写入程序 (使用内部触发信号)。为简单起见, 该程序使用了频率为 104 Hz 的加速度计, I²C 辅助总线上无外部上拉电阻。

1. 将 40h 写入 FUNC_CFG_ACCESS // 使能对传感器集合寄存器的访问
2. 将 EXT_SENS_ADDR 写入 SLV0_ADD // 配置外部器件地址 (EXT_SENS_ADDR)
// 使能写操作 (rw_0 = 0)
3. 将 REG 写入 SLV0_SUBADD // 配置要写入的寄存器的地址 (REG)
4. 将 00h 写入 SLAVE0_CONFIG // SHUB_ODR = 104 Hz
5. 将 VAL 写入 DATAWRITE_SLV0 // 配置要在 REG 中写入的值 (VAL)
6. 将 4Ch 写入 MASTER_CONFIG // 为单次写入使能 WRITE_ONCE
// I²C 主器件使能, 只使用 slave 0
// SDx 和 SCx 上的 I²C 上拉电阻使能
7. 轮询 STATUS_MASTER,
直至 WR_ONCE_DONE = 1 // 等待传感器集合写事务
8. 将 08h 写入 MASTER_CONFIG // I²C 主器件禁用
9. 等待 300 μs
10. 将 00h 写入 FUNC_CFG_ACCESS // 禁用对传感器集合寄存器的访问

下面的程序将 LIS2MDL 外部磁力计传感器 (更多详情见数据手册) 配置为连续转换模式 (使能温度补偿、BDU 和偏移消除功能), 频率为 100 Hz, 并读取磁力计输出寄存器, 将其值保存到寄存器 SENSOR_HUB_1 至 SENSOR_HUB_6 中。

1. 将 40h 写入 CTRL1_XL // 开启加速度计 (以获取触发信号), 频率为 104 Hz



2. 执行一次性读取, 其中:
 - SLV0_ADD = 3Dh // 检查 LIS2MDL WHO_AM_I 寄存器
 - SLV0_SUBADD = 4Fh // LIS2MDL 从器件地址为 3Ch 且 rw_0=1
3. 执行一次性写入, 其中:
 - SLV0_ADD = 3Ch // WHO_AM_I 寄存器地址为 4Fh
 - SLV0_SUBADD = 60h // 写入 LIS2MDL 寄存器 CFG_REG_A (60h) = 8Ch
 - DATAWRITE_SLV0 = 8Ch // LIS2MDL 从器件地址为 3Ch 且 rw_0=0
4. 执行一次性写入, 其中:
 - SLV0_ADD = 3Ch // 使能温度补偿
 - SLV0_SUBADD = 61h // 使能磁力计, 100 Hz ODR, 连续模式
 - DATAWRITE_SLV0 = 02h // 写入 LIS2MDL 寄存器 CFG_REG_B (61h) = 02h
5. 执行一次性写入, 其中:
 - SLV0_ADD = 3Ch // LIS2MDL 从器件地址为 3Ch 且 rw_0=0
 - SLV0_SUBADD = 62h // 使能磁力计 BDU
 - DATAWRITE_SLV0 = 10h // 写入 LIS2MDL 寄存器 CFG_REG_B (62h) = 10h
6. 设置连续读取, 其中:
 - SLV0_ADD = 3Dh // LIS2MDL 从器件地址为 3Ch 且 rw_0=1
 - SLV0_SUBADD = 68h // 磁力计输出寄存器, 从 68h 开始
 - SLAVE0_CONFIG = 06h // 设置从 I²C 主接口进行连续 6 字节读取

8 模式 3 和模式 4 - 辅助 SPI 模式

辅助 SPI 模式（模式 3 和模式 4）允许从多个外部器件访问该器件：当这些模式之一使能时，I²C/SPI（3/4 线）/ MIPI I3CSM 从接口和辅助 SPI（3/4 线）从接口均可用于连接外部器件。

当模式 3 使能时，陀螺仪 OIS 链激活。当模式 4 使能时，加速度计 OIS 链和陀螺仪 OIS 链均激活。

举例来说，在光学防抖（OIS）应用中，可以使用它们同时从应用处理器和照相机模块访问器件。照相机模块可连续高速获取传感器数据用于其图像稳定算法。

8.1 辅助 SPI 模式说明

可用两种不同方式使能辅助 SPI 模式：

- 辅助 SPI 全控制：通过辅助 SPI 执行辅助接口的使能/禁用和模式 3/4 的配置；
- 主接口使能：通过主接口（I²C/SPI（3/4 线）/MIPI I3CSM 从接口）执行辅助接口的使能/禁用，通过辅助 SPI 执行模式 3/4 的配置。

辅助 SPI 全控制为照相机模块完全独立于应用处理器（应用处理器通过主接口与器件相连）的情况而设计。可通过 CTRL1_OIS 寄存器 OIS_EN_SPI2 位的置位来使能辅助 SPI 模式。此操作自动使能陀螺仪 OIS 链（模式 3）。为了同时使能加速度计 OIS 链（模式 4），可以将 CTRL1_OIS 寄存器的 Mode4_EN 位置为 1。

主接口使能为可通过应用处理器控制照相机模块的情况而设计。这种情况下，可以通过主接口使能/禁用辅助 SPI 接口。可通过将 CTRL7_G 寄存器的 OIS_ON_EN 位置为 1 来选择主接口使能模式。这种情况下，可通过将 CTRL7_G 寄存器的 OIS_ON 位置为 1 来使能辅助 SPI 模式。CTRL1_OIS 寄存器的 OIS_EN_SPI2 位变为只读且其值维持 0 不变。使能辅助 SPI 模式将自动使能陀螺仪 OIS 链（模式 3）。为了同时使能加速度计 OIS 链（模式 4），可以将 CTRL1_OIS 寄存器的 Mode4_EN 位置为 1。

注：当模式 3 或模式 4 使能时，不能将加速度计配置为超低功耗模式。此外，如果在需要使能 OIS（模式 3 或模式 4）时将加速度计配置为超低功耗模式，则必须在使能 OIS 链之前将器件配置为下电模式。出于这个原因，如果准备使用加速度计超低功耗模式，建议使用主接口使能模式，以便在模式 3/4 使能时正确地管理加速度计功耗模式。

当模式 3 使能时，可通过选择的辅助 SPI 接口（3/4 线）获取陀螺仪输出值，满量程通过 CTRL1_OIS 寄存器的 FS[1:0]_G_OIS 和 FS_125_OIS 位进行选择，ODR 为 6.66 kHz。

如果使能了模式 4（通过将 CTRL1_OIS 寄存器的 Mode4_EN 位置为 1 并使能陀螺仪 OIS 链），则除了陀螺仪值，还可通过辅助 SPI 接口获取加速度计输出值（ODR 为 6.66 kHz）。OIS 链上的加速度计满量程可通过 CTRL3_OIS 寄存器的 FS[1:0]_XL_OIS 位进行配置。有两种可能的配置：

- OIS 链满量程的选择取决于 UI 链：可通过将 CTRL8_XL 寄存器的 XL_FS_MODE 位置为 0 来选择。在这种情况下，可以有两种情况：
 - 加速度计 UI 链未处于下电模式：通过 CTRL1_XL 寄存器的 FS[1:0]_XL 位选择 OIS 链满量程。
 - 加速度计 UI 链处于下电模式：通过 CTRL3_OIS 寄存器的 FS[1:0]_XL_OIS 位选择 OIS 链满量程。
- OIS 链满量程的选择与 UI 链无关：可通过将 CTRL8_XL 寄存器的 XL_FS_MODE 位置为 1 来选择。这种情况下，通过 CTRL3_OIS 寄存器的 FS[1:0]_XL_OIS 位选择配置，最高 8 g。

下表总结了加速度计 OIS 链满量程选择的所有可能配置。

表 53. 加速度计 OIS 链满量程的选择

FS[1:0]_XL_OIS	XL_FS_MODE = 0		XL_FS_MODE = 1
	加速度计（UI）未处于下电模式	加速度计（UI）处于下电模式	-
00	通过 CTRL1_XL 寄存器的 FS[1:0]_XL 位选择满量程	2 g	2 g
01		16 g	2 g
10		4 g	4 g
11		8 g	8 g

模式 3/模式 4 使能后器件引脚的功能如下表所示。

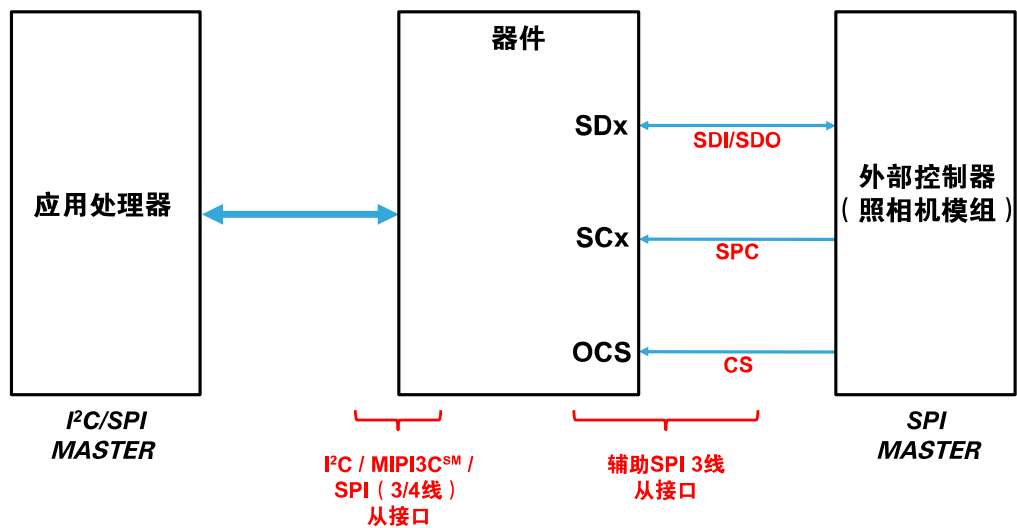


表 54. 模式 3/4 引脚说明

引脚	模式 3/4 功能
SDx	辅助 SPI 3/4 线接口串行数据输入 (SDI) 和 SPI 3 线串行数据输出 (SDO)
SCx	辅助 SPI 3/4 线串行端口时钟 (SPC)
OCS_Aux	辅助 SPI 3/4 线芯片使能
SDO_Aux	辅助 SPI 4 线串行数据输出 (SDO)

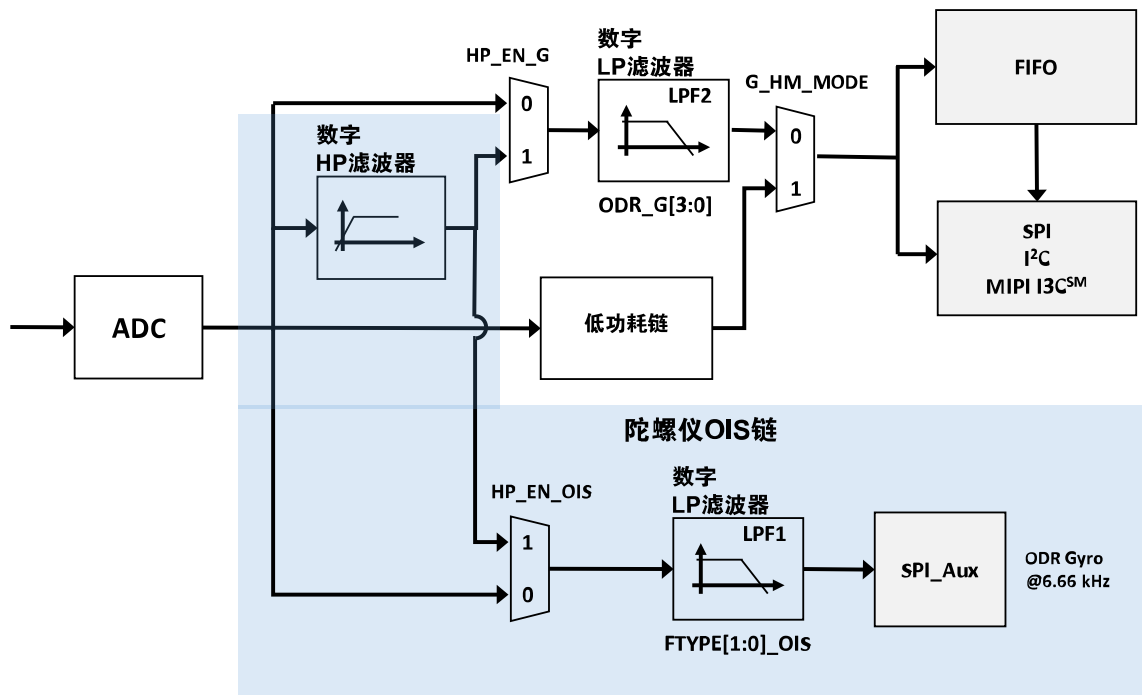
如果使用 SPI 3 线接口 (CTRL1_OIS 中的 SIM_OIS 位 = 1)，外部器件必须按照图 24. 模式 3/4 下的外部控制器连接 (SPI 3 线) 所示连接到 LSM6DSO。在使用 SPI 4 线接口 (还连接 SDO_Aux 引脚) 时，必须相应地更改设置。

图 24. 模式 3/4 下的外部控制器连接 (SPI 3 线)



陀螺仪滤波链如图 25. 陀螺仪滤波链中所示。数字低通滤波器 LPF1 专用于 OIS 链，可以配置四种不同的截止带宽。UI 和 OIS 链共享数字高通滤波器。不能同时对 UI 链和 OIS 链使能高通滤波器。如果 CTRL7_G 寄存器的 HP_EN_G 位和 CTRL2_OIS 寄存器的 HP_EN_OIS 位均置为 1，则高通滤波器仅应用于 UI 链。

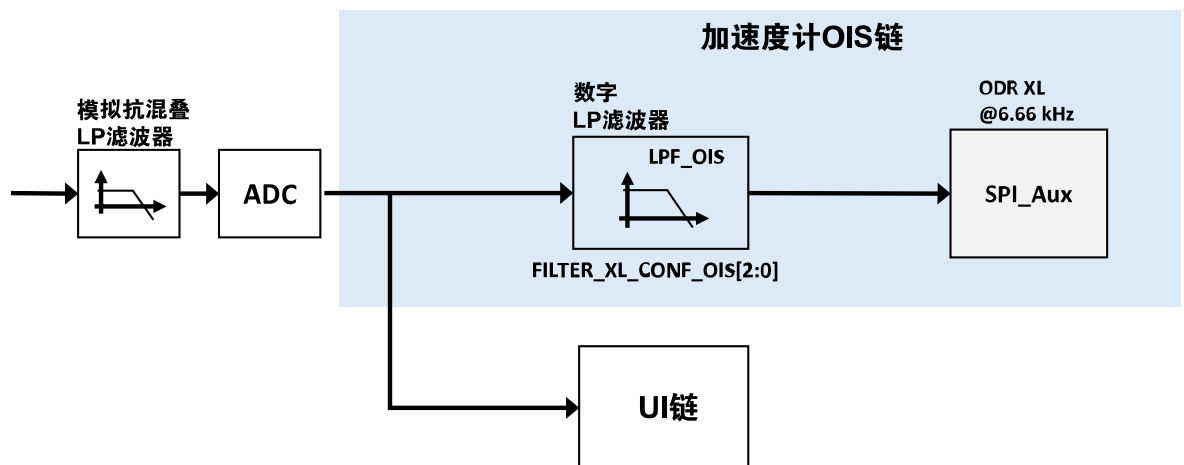
图 25. 陀螺仪滤波链





加速度计滤波链如图 26. 加速度计滤波链中所示。数字低通滤波器 LPF_OIS 专用于 OIS 链，可以配置八种不同的截止带宽。

图 26. 加速度计滤波链



8.2 辅助 SPI 模式寄存器

主接口始终可用，可读取寄存器 22h 至 27h 中的陀螺仪输出值，并可通过 CTRL2_G 寄存器选择满量程和 ODR。同样地，可通过主接口读取寄存器 28h 至 2Dh 中的加速度计输出值，并可通过 CTRL1_XL 寄存器选择满量程和 ODR。

FIFO 中保存的加速度计/陀螺仪数据只能通过主接口访问。

INT_OIS、CTRL1_OIS、CTRL2_OIS 和 CTRL3_OIS 寄存器的位值只能通过辅助 SPI 接口修改（当通过主接口访问时，这些寄存器为只读寄存器）。只有这些寄存器可以通过辅助 SPI 接口写入；所有其他读/写寄存器均只能通过主接口写入，并且只能通过辅助 SPI 读取。

当模式 3 使能时，可通过辅助 SPI 接口读取寄存器 22h 至 27h 中的陀螺仪输出值。当 OIS 链上有新的陀螺仪数据可用时，STATUS_SPIAux 寄存器的 GDA 位置为 1；在读取输出数据寄存器（23h、25h、27h）的一个高部分时，GDA 位复位。当陀螺仪 OIS 链处于建立阶段时，STATUS_SPIAux 寄存器的 GYRO_SETTLING 位等于 1。在该建立阶段读取的数据无效。建议检查此位的状态以了解何时有效数据可用。

当模式 4 使能时，除了陀螺仪输出值（寄存器 22h 至 27h 中），还可以通过辅助 SPI 接口读取寄存器 28h 至 2Dh 中的加速度计输出值。当 OIS 链上有新的加速度计数据可用时，STATUS_SPIAux 寄存器的 XLDA 位置为 1；在读取输出数据寄存器（29h、2Bh、2Dh）的一个高部分时，XLDA 位复位。

当从主接口和辅助 SPI 接口读取数据时，加速度计/陀螺仪输出数据寄存器（22h 至 2Dh）和 STATUS_REG 寄存器（1Eh）包含不同数据。所有其他寄存器包含相同值。

可同时从两个外部主器件读取器件的所有寄存器。

8.2.1 INT_OIS (6Fh)

表 55. INT_OIS 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
INT2_DRDY_OIS	LVL2_OIS	DEN_LH_OIS	-	-	0	ST1_XL_OIS	ST0_XL_OIS

- INT2_DRDY_OIS 位可用于将 OIS 链的 DRDY 信号驱动至 INT2 引脚。OIS 链的 DRDY 信号始终是脉冲；锁存模式不可用。屏蔽路由至 INT2 引脚的中断信号，直至 STATUS_SPIAux 寄存器的 GYRO_SETTLING 位被置为 1。
- LVL2_OIS 使能（与 CTRL1_OIS 寄存器的 LVL1_OIS 位一起）OIS 链上的电平感应触发/锁存模式；请参考第 8.2.2 节 CTRL1_OIS (70h) 获取详细信息。
- DEN_LH_OIS 位可用于选择 OIS 链上 DEN 信号的极性：如果置为 0，则 DEN 引脚为低电平有效，否则为高电平有效。
- 可以将 ST[1:0]_XL_OIS 位置位，以便选择加速度计 OIS 链上的自检功能（参见第 11 节 自检功能了解详细信息）。



8.2.2 CTRL1_OIS (70h)

表 56. CTRL1_OIS 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
0	LVL1_OIS	SIM_OIS	Mode4_EN	FS1_G_OIS	FS0_G_OIS	FS_125_OIS	OIS_EN_SPI2

- LVL1_OIS 可与 INT_OIS 寄存器的 LVL2_OIS 位一起，用于使能 OIS 上的电平感应触发模式（表 57. DEN 模式选择）。
- 为了使能 3 线辅助 SPI 接口，SIM_OIS 位必须置为 1，否则使用 4 线辅助 SPI 接口。
- Mode4_EN 位使能加速度计 OIS 链（模式 4）；还必须使能陀螺仪 OIS 链。
- FS[1:0]_G_OIS 位可用于选择陀螺仪 OIS 满量程（当 FS_125_OIS 位置为 0 时），类似于 CTRL2_G 寄存器的 FS[1:0]_G 位。
- FS_125_OIS 位使能陀螺仪 OIS 链上的 ± 125 dps 满量程。如果等于 0，则通过 FS[1:0]_G_OIS 位选择满量程。
- 为了在使用辅助 SPI 全控制模式时使能通过辅助 SPI 接口进行陀螺仪的 OIS 链数据处理（模式 3），OIS_EN_SPI2 位必须置为 1。

OIS 侧的 DEN 模式可使用 CTRL1_OIS 寄存器的 LVL1_OIS 位和 INT_OIS 寄存器的 LVL2_OIS 位使能。

表 57. DEN 模式选择

LVL1_OIS, LVL2_OIS	DEN 模式（OIS 链）
00	OIS 路径上的 DEN 模式禁用
10	已选择电平感应触发模式
11	已选择电平感应锁存模式

在 OIS 路径上，仅陀螺仪传感器的 DEN 模式激活。一旦使能了两种 OIS DEN 模式之一，所有三个轴的 LSB 位会如第 4.8 节 DEN（数据使能）所述发生变化。在这种情况下，不可能只选择一个或两个轴。

8.2.3 CTRL2_OIS (71h)

表 58. CTRL2_OIS 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
-	-	HPM1_OIS	HPM0_OIS	0	FTYPE_1_OIS	FTYPE_0_OIS	HP_EN_OIS

- HPM[1:0]_OIS 位可用于选择陀螺仪 OIS 侧的数字 HP 滤波器截止频率。下表所示为可用配置。

表 59. 陀螺仪 OIS 链 HP 滤波器截止频率选择

HPM[1:0]_OIS	截止[Hz]	建立时间[s]
00	0.016	45
01	0.065	11
10	0.26	3
11	1.04	0.7

- FTYPE_[1:0]_OIS 位可用于选择数字 LPF1 滤波器带宽。下表所示为使用所有配置获得的截止频率和相位延迟值。

表 60. LPF1 滤波器配置

FTYPE_[1:0]_OIS	截止[Hz]	相位 @ 20 Hz [°]	稳定时间[要丢弃的采样数]
00	335.50	-6.69	27
01	232.00	-8.78	36
10	171.10	-11.18	48
11	609.00	-4.91	19

- HP_EN_OIS 位可用于使能陀螺仪 OIS 链上的 HP 滤波器。陀螺仪 UI 和 OIS 链共享数字高通滤波器。仅当 CTRL7_G 寄存器中的 HP_EN_OIS 位置为 1 且 HP_EN_G 位置为 0 时，OIS 侧的 HP 滤波器才可用。

8.2.4 CTRL3_OIS (72h)

表 61. CTRL3_OIS 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
FS1_XL_OIS	FS0_XL_OIS	FILTER_XL_CONF_OIS_2	FILTER_XL_CONF_OIS_1	FILTER_XL_CONF_OIS_0	ST1_OIS	ST0_OIS	ST_OIS_CLAMPDIS

- FS[1:0]_XL_OIS 位可用于选择加速度计 OIS 满量程，如第 8.1 节 辅助 SPI 模式说明所述。
- FILTER_XL_CONF_OIS_[2:0]位可用于选择数字 LPF_OIS 带宽。下表所示为使用所有配置获得的截止频率和相位延迟值。

表 62. LPF_OIS 滤波器配置

FILTER_XL_CONF_OIS [2:0]	截止[Hz]	相位[°]	稳定时间[要丢弃的采样数]
000	289	-5.72 @ 20 Hz	19
001	258	-6.80 @ 20 Hz	21
010	120	-13.20 @ 20 Hz	42
011	65.1	-21.50 @ 20 Hz	80
100	33.2	-19.1 @ 10 Hz	155
101	16.6	-33.5 @ 10 Hz	305
110	8.30	-26.7 @ 4 Hz	600
111	4.14	-26.2 @ 2 Hz	1200

- 可以将 ST[1:0]_OIS 位置位，以便选择陀螺仪 OIS 链上的自检功能（参见第 11 节 自检功能了解详细信息）。
 - ST_OIS_CLAMPDIS 位可用于在陀螺仪和加速度计自检中使能/禁用 OIS 链钳位。如果 ST_OIS_CLAMPDIS 位置为 1，则一旦陀螺仪/加速度计自检功能使能，从辅助 SPI 接口读取的输出值将显示在从主接口读取数据时观测到的相同变化。如果 ST_OIS_CLAMPDIS 位置为 0，则当陀螺仪/加速度计自检功能使能时，从辅助 SPI 接口读取的输出值始终钳位至 8000h 值。例如，此功能允许连接到辅助接口的主机设备检测从 UI 侧使能自检功能的时间。设计的最大输出值比 8000h 小一个 LSB，因此如果从辅助 SPI 读取了 8000h，则表示从 UI 侧使能了自检功能。

8.2.5 STATUS_SPIAux (1Eh)

表 63. STATUS_SPIAux 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	GYRO_SETTLING	GDA	XLDA

- 陀螺仪输出的初始设置阶段，GYRO_SETTLING 位置为 1。在此位等于 1 时生成的陀螺仪输出数据必须丢弃。
 - 注：GYRO_SETTLING 位不考虑陀螺仪 HP 滤波器。如果使能 OIS 链上的 HP 滤波器，用户应考虑其稳定时间。
- 当 OIS 链上的寄存器 22h 至 27h 中有新的陀螺仪数据可用时，GDA 位置为 1。在读取输出数据寄存器的一个高部分时，GDA 位置复位。
- 当 OIS 链上的寄存器 28h 至 2Dh 中有新的加速度计数据可用时，XLDA 位置为 1。在读取输出数据寄存器的一个高部分时，XLDA 位置复位。

8.3 OIS 链稳定时间

陀螺仪和加速度计传感器读取链包含低通和高通滤波功能。陀螺仪还需要最长 **70 ms** 的开启时间。出于这些原因，在功耗模式改变后开始采集传感器数据前，必须考虑滤波器稳定时间和开启时间。

下表显示了所有可能配置的稳定时间。

表 64. OIS 链稳定时间

起始模式 UI	起始模式 OIS	目标模式 OIS	开启 + 滤波器稳定
XL: 下电 Gyro: 下电	XL: 下电 Gyro: 下电	XL: @6.66 kHz 陀螺仪: @6.66 kHz (模式 4)	加速度计: 滤波器稳定 ⁽¹⁾ 陀螺仪: 70 ms + 滤波器稳定 ⁽²⁾
XL: 下电 Gyro: 下电	XL: 下电 陀螺仪: @6.66 kHz (模式 3)	XL: @6.66 kHz 陀螺仪: @6.66 kHz (模式 4)	加速度计: 滤波器稳定 ⁽¹⁾ 陀螺仪: 第一个采样正确
加速度计: 每次 ODR 和功耗模式改变 Gyro: 下电	XL: 下电 Gyro: 下电	XL: @6.66 kHz 陀螺仪: @6.66 kHz (模式 4)	加速度计: 滤波器稳定 ⁽¹⁾ 陀螺仪: 70 ms + 滤波器稳定 ⁽²⁾
加速度计: 每次 ODR 和功耗模式改变 Gyro: 下电	XL: 下电 陀螺仪: @6.66 kHz (模式 3)	XL: @6.66 kHz 陀螺仪: @6.66 kHz (模式 4)	加速度计: 滤波器稳定 ⁽¹⁾ 陀螺仪: 第一个采样正确
XL: 下电 陀螺仪: 每次 ODR 和功耗模式改变	XL: 下电 Gyro: 下电	XL: @6.66 kHz 陀螺仪: @6.66 kHz (模式 4)	加速度计: 滤波器稳定 ⁽¹⁾ 陀螺仪: 滤波器稳定 ⁽²⁾
XL: 下电 陀螺仪: 每次 ODR 和功耗模式改变	XL: 下电 陀螺仪: @6.66 kHz (模式 3)	XL: @6.66 kHz 陀螺仪: @6.66 kHz (模式 4)	加速度计: 滤波器稳定 ⁽¹⁾ 陀螺仪: 第一个采样正确
加速度计: 每次 ODR 和功耗模式改变 陀螺仪: 每次 ODR 和功耗模式改变	XL: 下电 Gyro: 下电	XL: @6.66 kHz 陀螺仪: @6.66 kHz (模式 4)	加速度计: 滤波器稳定 ⁽¹⁾ 陀螺仪: 滤波器稳定 ⁽²⁾
加速度计: 每次 ODR 和功耗模式改变 陀螺仪: 每次 ODR 和功耗模式改变	XL: 下电 陀螺仪: @6.66 kHz (模式 3)	XL: @6.66 kHz 陀螺仪: @6.66 kHz (模式 4)	加速度计: 滤波器稳定 ⁽¹⁾ 陀螺仪: 第一个采样正确

1. **LPF_OIS** 滤波器稳定时间，如表 62. **LPF_OIS** 滤波器配置所示。
2. **HP**（如已配置）和 **LPF1** 滤波器稳定时间中的最大值，如表 59. 陀螺仪 OIS 链 **HP** 滤波器截止频率选择和表 60. **LPF1** 滤波器配置所示。



8.4 模式 3 - 通过辅助 SPI 读取陀螺仪数据

器件上电后通过辅助 SPI 3 线接口读取陀螺仪输出数据的流程如下：

1. 等待 10 ms // 启动时间
// 该时间段结束后器件处于下电模式
2. 将 21h 写入 CTRL1_OIS // 通过辅助 SPI 3 线接口启用陀螺仪
// (OIS 陀螺仪: FS = ± 250 dps / ODR = 6.66 kHz)
3. 等待 74 ms // 陀螺仪最大导通时间为 70 ms
// 选择的 LPF1 (00b) 稳定时间为 4.05 ms
// (27 个采样 @ 6.66 kHz)
4. 读取输出寄存器 22h 至 27h // 通过辅助 SPI 读取陀螺仪输出数据

8.5 模式 4 - 通过辅助 SPI 读取陀螺仪和加速度计数据

器件上电后通过辅助 SPI 3 线接口读取陀螺仪和加速度计输出数据的流程如下：

1. 等待 10 ms // 启动时间
// 该时间段结束后器件处于下电模式
2. 将 31h 写入 CTRL1_OIS // 通过辅助 SPI 3 线接口启用陀螺仪
// (OIS 陀螺仪: FS = ± 250 dps / ODR = 6.66 kHz)
// 使能模式 4 (Mode4_EN = 1)
3. 将 00h 写入 CTRL3_OIS // 通过辅助 SPI 3 线接口设置 XL
// (OIS XL: FS = ± 2 g / ODR = 6.66 kHz)
4. 等待 74 ms // 陀螺仪最大导通时间为 70 ms
// 选择的 LPF1 (00b) 稳定时间为 4.05 ms
// (27 个采样 @ 6.66 kHz)
// 选择的 LPF OIS (000b) 稳定时间为 2.85 ms
// (19 个采样 @ 6.66 kHz)
5. 读取输出寄存器 22h 至 27h // 通过辅助 SPI 读取陀螺仪输出数据
6. 读取输出寄存器 28h 至 2Dh // 通过辅助 SPI 读取加速度计输出数据



9 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，LSM6DSO 嵌入了一个 3 KB（最多 9 KB，压缩功能使能）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；
- 时间戳数据；
- 温度传感器数据；
- 外部传感器（连接到传感器集合 (sensor hub) 接口）数据；
- 计步器（和相关时间戳）数据。

基于 FIFO 字将数据保存在 FIFO 中。一个 FIFO 字包含：

- 标签，1 字节
- 数据，6 字节

可通过六个专用寄存器（从地址 79h 至 7Eh）从 FIFO 检索数据：FIFO_DATA_OUT_X_L、FIFO_DATA_OUT_X_H、FIFO_DATA_OUT_Y_L、FIFO_DATA_OUT_Y_H、FIFO_DATA_OUT_Z_L 和 FIFO_DATA_OUT_Z_H。

FIFO_DATA_OUT_TAG 寄存器的 FIFO_TAG 字段可用于识别 FIFO 中字的含义，使得 FIFO 流的重建成为一项简单的任务。在使用专用 FIFO 配置的情况下选择传感器批处理数据率时，应用具有最大的灵活性。

利用 FIFO_CTRL4 寄存器的 FIFO_MODE[2:0] 位，有六种不同的 FIFO 操作模式可供选择：

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

要监控 FIFO 状态（满，溢出，存储的采样数，等等...），可以使用两个专用寄存器：FIFO_STATUS1 和 FIFO_STATUS2。

可编程 FIFO 阈值可以利用 WTM[8:0] 位在 FIFO_CTRL1 和 FIFO_CTRL2 中进行设置。

通过 INT1_CTRL 寄存器的 INT1_FIFO_FULL、INT1_FIFO_FTH 和 INT1_FIFO_OVR 位，以及 INT2_CTRL 寄存器的 INT2_FIFO_FULL、INT2_FIFO_FTH 和 INT2_FIFO_OVR 位，可以使能 FIFO 满、FIFO 阈值和 FIFO 溢出事件，在两个中断引脚（INT1 和 INT2）上产生专门的中断。

最后，FIFO 嵌入了压缩算法，用户可以使能压缩算法以便在 FIFO 中保存最多 9 字节的数据，并从 FIFO 清空和通信功耗方面利用接口通信长度的优势。



9.1 FIFO 说明和批处理传感器

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

FIFO 运行时间可配置：可使能元信息标记，以便在批处理传感器配置更改后通知用户。

此外，为了增加其容量，FIFO 内置了加速度计和陀螺仪数据的压缩算法（请参考第 9.10 节 FIFO 压缩了解详细信息）。

批处理传感器可分为三个不同类别：

1. 主要传感器，即物理传感器：
 - a. 加速度计传感器；
 - b. 陀螺仪传感器；
2. 辅助传感器，其中包含设备状态信息：
 - a. 时间戳传感器；
 - b. 配置变化传感器（CFG-Change）；
 - c. 温度传感器；
3. 虚拟传感器：
 - a. 从传感器集合接口读取的外部传感器；
 - b. 计步器传感器。

可通过六个专用寄存器从 FIFO 检索数据：FIFO_DATA_OUT_X_L、FIFO_DATA_OUT_X_H、FIFO_DATA_OUT_Y_L、FIFO_DATA_OUT_Y_H、FIFO_DATA_OUT_Z_L 和 FIFO_DATA_OUT_Z_H。

可通过三个不同事件触发 FIFO 写入：

- 内部数据就绪信号（加速度计与陀螺仪之间的最快速传感器）；
- 传感器集合数据就绪；
- 步伐检测事件。

9.2 FIFO 寄存器

FIFO 缓冲器由以下几项管理：

- 六个控制寄存器：FIFO_CTRL1、FIFO_CTRL2、FIFO_CTRL3、FIFO_CTRL4、COUNTER_BDR_REG1 和 COUNTER_BDR_REG2；
- 两个状态寄存器：FIFO_STATUS1 和 FIFO_STATUS2；
- 七个输出寄存器（标签 + 数据）：FIFO_DATA_OUT_TAG、FIFO_DATA_OUT_X_L、FIFO_DATA_OUT_X_H、FIFO_DATA_OUT_Y_L、FIFO_DATA_OUT_Y_H、FIFO_DATA_OUT_Z_L 和 FIFO_DATA_OUT_Z_H；
- 一些额外的位，将 FIFO 事件路由至两条中断线路：INT1_CTRL 寄存器的 INT1_CNT_BDR、INT1_FIFO_FULL、INT1_FIFO_OVR 和 INT1_FIFO_TH 位，以及 INT2_CTRL 寄存器的 INT2_CNT_BDR、INT2_FIFO_FULL、INT2_FIFO_OVR 和 INT2_FIFO_TH 位；
- 用于其他功能的一些额外的位：
 - EMB_FUNC_EN_B 嵌入功能寄存器的 FIFO_COMPR_EN 位，用于使能 FIFO 压缩算法；
 - EMB_FUNC_FIFO_CFG 寄存器的 PEDO_FIFO_EN 位，用于使能 FIFO 中的计步器批处理；
 - EMB_FUNC_INIT_B 嵌入功能寄存器的 FIFO_COMPR_INIT 位，用于请求 FIFO 压缩算法重新初始化；
 - SLAVE0_CONFIG、SLAVE1_CONFIG、SLAVE2_CONFIG 和 SLAVE3_CONFIG 传感器集合寄存器的 BATCH_EXT_SENS_0_EN、BATCH_EXT_SENS_1_EN、BATCH_EXT_SENS_2_EN 和 BATCH_EXT_SENS_3_EN 位，使能 FIFO 中相关外部传感器的批处理。

9.2.1 FIFO_CTRL1

FIFO_CTRL1 寄存器包含 9 位 FIFO 水印阈值的低位部分。对于完整的水印阈值配置，还要考虑 FIFO_CTRL2 寄存器的 WTM8 位。FIFO 阈值的 1 LSB 值称为 FIFO 字（7 字节）。

当 FIFO 中存储的字节数大于或等于水印阈值时，FIFO 水印标记（FIFO_STATUS2 寄存器的 FIFO_WTM_IA 位）升高。

要将 FIFO 深度限制为水印级别，FIFO_CTRL2 寄存器中 STOP_ON_WTM 位必须设置为 1。

表 65. FIFO_CTRL1 寄存器

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

9.2.2 FIFO_CTRL2

表 66. FIFO_CTRL2 寄存器

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
STOP_ON_WTM	FIFO_COMPR_RT_EN	0	ODRCHG_EN	0	UNCOPTR_RATE_1	UNCOPTR_RATE_0	WTM8

FIFO_CTRL2 寄存器包含 9 位 FIFO 水印阈值的高位部分（WTM8 位）。对于完整的水印阈值配置，还要考虑 FIFO_CTRL1 寄存器的 WTM[7:0] 位。寄存器包含了位 STOP_ON_WTM，可将 FIFO 深度限制在水印级别。

FIFO_CTRL2 寄存器还包含用于管理加速度计和陀螺仪传感器的 FIFO 压缩算法的位：

- FIFO_COMPR_RT_EN 位允许压缩算法的运行时间使能/禁用：如果此位置 1，则使能压缩，否则禁用；
- UNCOPTR_RATE_[1:0] 配置压缩算法，以便以特定速率写入未压缩数据。下表总结了可能的配置。

表 67. 强制未压缩数据写入配置

UNCOPTR_RATE[1:0]	强制未压缩数据写入
00	从不
01	每 8 BDR
10	每 16 BDR
11	每 32 BDR

此外，FIFO_CTRL2 寄存器包含 ODRCHG_EN 位，可将此位置为 1，以使能 CFG-Change 辅助传感器在 FIFO 中的批处理（描述见下文）。

9.2.3 FIFO_CTRL3

表 68. FIFO_CTRL3 寄存器

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
BDR_GY_3	BDR_GY_2	BDR_GY_1	BDR_GY_0	BDR_XL_3	BDR_XL_2	BDR_XL_1	BDR_XL_0

FIFO_CTRL3 寄存器包含用于选择 FIFO 中加速度计和陀螺仪传感器数据写入频率的字段。选择的批处理数据率必须小于等于通过 CTRL1_XL 和 CTRL2_G 寄存器的 ODR_XL 和 ODR_G 字段配置的输出数据率。

下面的表格显示了所有可选择的批处理数据率。

表 69. 加速度计批处理数据率

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

BDR_XL[3:0]	批处理数据率[Hz]
0001	12.5
0010	26
0011	52
0100	104
0101	208
0110	417
0111	833
1000	1667
1001	3333
1010	6667
1011	1.6

表 70. 陀螺仪批处理数据率

BDR_GY[3:0]	批处理数据率[Hz]
0000	FIFO 中不批处理
0001	12.5
0010	26
0011	52
0100	104
0101	208
0110	417
0111	833
1000	1667
1001	3333
1010	6667
1011	6.5

9.2.4 FIFO_CTRL4

FIFO_CTRL4 寄存器包含用于选择 FIFO 中时间戳批处理抽取因子和温度传感器批处理数据率的字段。

时间戳写入速率被配置为加速度计和陀螺仪批处理数据率中的最大速率除以 DEC_TS_BATCH_[1:0] 字段指定的抽取因子。下表显示了可编程抽取因子。

表 71. 时间戳批处理数据率

DEC_TS_BATCH[1:0]	时间戳批处理数据率[Hz]
00	FIFO 中不批处理
01	$\max(\text{BDR_GY}[\text{Hz}], \text{BDR_XL}[\text{Hz}], \text{BDR_SHUB}[\text{Hz}])$
10	$\max(\text{BDR_GY}[\text{Hz}], \text{BDR_XL}[\text{Hz}], \text{BDR_SHUB}[\text{Hz}]) / 8$
11	$\max(\text{BDR_GY}[\text{Hz}], \text{BDR_XL}[\text{Hz}], \text{BDR_SHUB}[\text{Hz}]) / 32$

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

表 72. 温度传感器批处理数据率

ODR_T_BATCH[1:0]	温度批处理数据率[Hz]
00	FIFO 中不批处理
01	1.6
10	12.5
11	52

FIFO_CTRL4 寄存器还包含 FIFO 操作模式位。FIFO 操作模式如第 9.7 节 FIFO 模式中所示。

表 73. FIFO_CTRL4 寄存器

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

9.2.5 COUNTER_BDR_REG1

由于 FIFO 可能包含元信息（即 CFG-Change 传感器）且加速度计和陀螺仪数据可能被压缩，FIFO 提供了一种在 FIFO 中保存的加速度计或陀螺仪实际采样数的基础上同步 FIFO 读取的方式：BDR 计数器。

BDR 计数器可通过 COUNTER_BDR_REG1 和 COUNTER_BDR_REG2 寄存器进行配置。

表 74. COUNTER_BDR_REG1 寄存器

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
-	RST_ COUNTER _BDR	TRIG_ COUNTER _BDR	0	0	CNT_BDR _TH_10	CNT_BDR _TH_9	CNT_BDR _TH_8

可以将 RST_COUNTER_BDR 置位以便复位 BDR 计数器：它将自动复位至零。

TRIG_COUNTER_BDR 选择 BDR 计数器的触发信号：如果配置为 0，则选择加速度计传感器，否则选择陀螺仪传感器。

用户可以在 FIFO_STATUS2 寄存器中选择生成 COUNTER_BDR_IA 事件的阈值。在内部 BDR 计数器达到阈值后，将 COUNTER_BDR_IA 位置为 1。阈值可通过 CNT_BDR_TH_[10:0] 位进行配置。字段上部包含在寄存器 COUNTER_BDR_REG1 中。CNT_BDR_TH 阈值的 1 LSB 值称为一个加速度计/陀螺仪采样（X、Y 和 Z 数据）。

9.2.6 COUNTER_BDR_REG2

COUNTER_BDR_REG2 寄存器包含 BDR 计数器阈值的下部。

表 75. COUNTER_BDR_REG2 寄存器

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
CNT_BDR _TH_7	CNT_BDR _TH_6	CNT_BDR _TH_5	CNT_BDR _TH_4	CNT_BDR _TH_3	CNT_BDR _TH_2	CNT_BDR _TH_1	CNT_BDR _TH_0

9.2.7 FIFO_STATUS1

FIFO_STATUS1 寄存器，连同 FIFO_STATUS2 寄存器一起，提供 FIFO 中存储的采样数相关信息。DIFF_FIFO 的 1 LSB 值称为一个 FIFO 字（7 字节）。

表 76. FIFO_STATUS1 寄存器

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DIFF_ FIFO_7	DIFF_ FIFO_6	DIFF_ FIFO_5	DIFF_ FIFO_4	DIFF_ FIFO_3	DIFF_ FIFO_2	DIFF_ FIFO_1	DIFF_ FIFO_0

9.2.8 FIFO_STATUS2

FIFO_STATUS2 寄存器，连同 FIFO_STATUS1 寄存器一起，可提供 FIFO 中存储的采样数相关信息和 FIFO 缓冲器当前状态信息（水印，溢出，满，BDR 计数器）。

表 77. FIFO_STATUS2 寄存器

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

- FIFO_WTM_IA 表示水印状态。当 FIFO 中已存储的 FIFO 字（每个 7 字节）数量大于等于水印阈值水平时，此位变为高电平。通过将 INT1_CTRL 寄存器的 INT1_FIFO_TH 位或 INT2_CTRL 寄存器的 INT2_FIFO_TH 位置为 1，可将水印状态信号驱动至两个中断引脚上。
- 当 FIFO 完全填满，至少一个采样已经被覆盖掉以存储新数据时，FIFO_OVR_IA 变为高电平。通过将 INT1_CTRL 寄存器的 INT1_FIFO_OVR 位或 INT2_CTRL 寄存器的 INT2_FIFO_OVR 位置为 1，可将此信号驱动至两个中断引脚上。
- 当要存储在 FIFO 中的下一组数据会使 FIFO 全满（即，DIFF_FIFO_9 = 1）或生成 FIFO 溢出时，FIFO_FULL_IA 被置为高电平。通过将 INT1_CTRL 寄存器的 INT1_FIFO_FULL 位或 INT2_CTRL 寄存器的 INT2_FIFO_FULL 位置为 1，可将此信号驱动至两个中断引脚上。
- COUNTER_BDR_IA 表示 BDR 计数器状态。当加速度计或陀螺仪批处理采样数（基于选择的传感器触发信号）达到通过 COUNTER_BDR_REG1 和 COUNTER_BDR_REG2 寄存器的 CNT_BDR_TH[10:0]位配置的 BDR 计数器阈值时，此位被置为高电平。在读取 FIFO_STATUS2 寄存器时 COUNTER_BDR_IA 位自动复位。通过将 INT1_CTRL 寄存器的 INT1_CNT_BDR 位或 INT2_CTRL 寄存器的 INT2_CNT_BDR 位置为 1，可将 BDR 计数器状态驱动至两个中断引脚上。
- 当 FIFO 完全填满，至少一个采样已经被覆盖掉以存储新数据时，FIFO_OVR_LATCHED（如同 FIFO_OVR_IA 一样）变为高电平。两个标记之间的差异在于，FIFO_OVR_LATCHED 在读取 FIFO_STATUS2 寄存器时复位，而 FIFO_OVR_IA 在读取至少一个 FIFO 字时复位。这样就可以在从 FIFO 读取数据时检测 FIFO 溢出条件。
- DIFF_FIFO_[9:8]包含 FIFO 中存储的未读字数的高位部分。其低位部分由 FIFO_STATUS1 中的 DIFF_FIFO_[7:0]位表示。DIFF_FIFO_[9:0]字段的值对应 FIFO 中的 7 字节字的数量。

寄存器内容会与 FIFO 写操作和读操作同步更新。

注：BDU 功能还作用于 FIFO_STATUS1 和 FIFO_STATUS2 寄存器。当 BDU 位置为 1 时，必须首先读取 FIFO_STATUS1，然后读取 FIFO_STATUS2。

9.2.9 FIFO_DATA_OUT_TAG

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

表 78. FIFO_DATA_OUT_TAG 寄存器

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

- TAG_SENSOR_[4:0]字段标识 6 个数据字节中保存的传感器（表 79）；
- TAG_CNT_[1:0]字段标识 FIFO 时隙（如后面的章节所述）；
- TAG_PARITY 位标识 FIFO_DATA_OUT_TAG 寄存器的内容是否损坏。

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

表 79. TAG_SENSOR 字段和相关传感器

TAG_SENSOR_[4:0]	传感器名称	传感器类别	说明
0x01	Gyroscope NC	主要	陀螺仪未压缩数据
0x02	Accelerometer NC	主要	加速度计未压缩数据
0x03	温度	辅助	温度数据
0x04	时间戳	辅助	时间戳数据
0x05	CFG_Change	辅助	元信息数据
0x06	加速度计 NC_T_2	主要	在两倍于先前时隙的时隙批处理的加速度计未压缩数据
0x07	加速度计 NC_T_2	主要	在先前时隙批处理的加速度计未压缩数据
0x08	加速度计 NC_T_2	主要	加速度计 2x 压缩数据
0x09	加速度计 NC_T_2	主要	加速度计 3x 压缩数据
0x0A	陀螺仪 NC_T_2	主要	在两倍于先前时隙的时隙批处理的陀螺仪未压缩数据
0x0B	陀螺仪 NC_T_1	主要	在先前时隙批处理的陀螺仪未压缩数据
0x0C	陀螺仪 2xC	主要	陀螺仪 2x 压缩数据
0x0D	陀螺仪 3xC	主要	陀螺仪 3x 压缩数据
0x0E	传感器集合 Slave 0	虚拟	来自 slave 0 的传感器集合数据
0x0F	传感器集合 Slave 1	虚拟	来自 slave 1 的传感器集合数据
0x10	传感器集合 Slave 2	虚拟	来自 slave 2 的传感器集合数据
0x11	传感器集合 Slave 3	虚拟	来自 slave 3 的传感器集合数据
0x12	步进计数器	虚拟	计数器数据
0x19	传感器集合 Nack	虚拟	来自 slave 0/1/2/3 的传感器集合 nack

TAG_PARITY 位可用于检查 FIFO_DATA_OUT_TAG 寄存器的内容。为此，用户可以执行以下程序：

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

9.2.10

FIFO_DATA_OUT

可通过六个专用寄存器（从地址 79h 至地址 7Eh）从 FIFO 检索数据：FIFO_DATA_OUT_X_L、FIFO_DATA_OUT_X_H、FIFO_DATA_OUT_Y_L、FIFO_DATA_OUT_Y_H、FIFO_DATA_OUT_Z_L 和 FIFO_DATA_OUT_Z_H。

FIFO 输出寄存器的内容取决于传感器类别和类型，如下一节所述。



9.3 FIFO 批处理传感器

如前文所述，批处理传感器可分为三个不同类别：

1. 主要传感器；
2. 辅助传感器；
3. 虚拟传感器。

本部分将提供每个类别的所有详细信息。

9.4 主要传感器

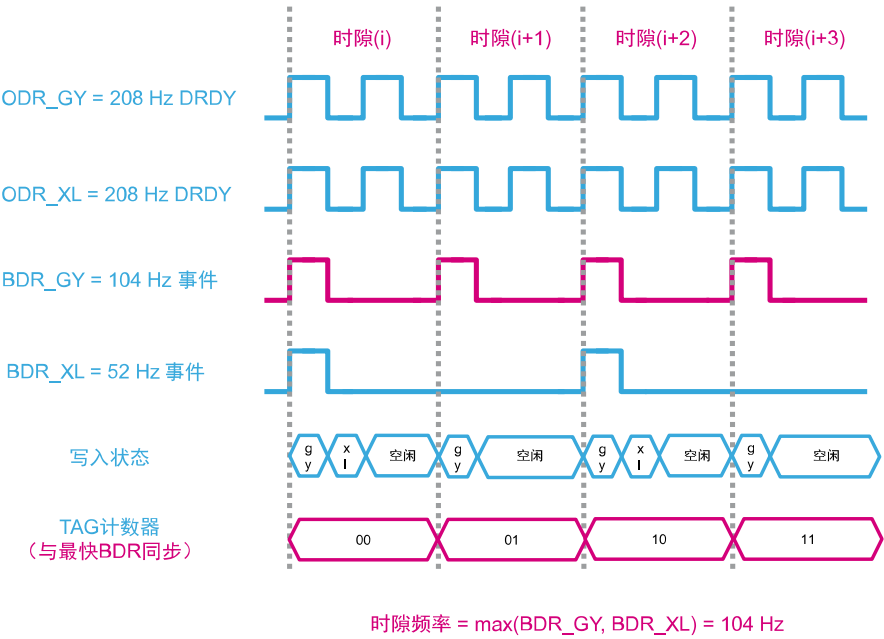
主要传感器是 LSM6DSO 设备物理传感器：加速度计和陀螺仪。批处理数据率可通过 FIFO_CTRL3 寄存器的 BDR_XL[3:0] 和 BDR_GY[3:0] 字段进行配置。批处理数据率必须小于等于通过 CTRL1_XL 和 CTRL2_G 寄存器的 ODR_XL[3:0] 和 ODR_G[3:0] 字段配置的相对传感器输出数据率。

主要传感器定义了 FIFO 时基。这意味着其他传感器中的每一个都与主要传感器定义的时基时隙相关。最快主要传感器的批处理事件也会使 TAG 计数器值（FIFO_DATA_OUT_TAG 寄存器的 TAG_CNT 字段）递增。此计数器由两个比特位组成，其值连续递增（从 00b 至 11b）以标识不同时序。

图 27. 主要传感器和时序的定义显示了批处理数据率事件的示例。BDR_GY 事件和 BDR_XL 事件标识相应传感器数据写入 FIFO 的时间。TAG 计数器的递增标识不同时间隙，其频率等于 BDR_XL 和 BDR_GY 中的最大值。



图 27. 主要传感器和时序的定义





下表显示了主要传感器的 FIFO 字格式，代表了从 78h 至 7Eh 的设备地址。

表 80. FIFO 中的主要传感器输出数据格式

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

9.5 辅助传感器

辅助传感器被视为主要传感器的服务传感器。辅助传感器包括：

- 温度传感器（必须正确地配置 **FIFO_CTRL4** 寄存器的 **ODR_T_BATCH_[1:0]** 位）；
- 时间戳传感器：它保存对应于 **FIFO** 时隙的时间戳（必须正确地配置 **CTRL10_C** 寄存器的 **TIMESTAMP_EN** 位和 **FIFO_CTRL4** 寄存器的 **DEC_TS_BATCH_[1:0]** 位）；
- **CFG-Change** 传感器：它识别器件的一些配置中的变化（**FIFO_CTRL2** 寄存器的 **ODRCHG_EN** 位必须置为 1）。

辅助传感器无法触发 **FIFO** 中的写操作。在发生第一个主要传感器或外部传感器事件时写入它们的寄存器（即使为它们配置了更高的批处理数据率）。

下表显示了 **FIFO** 中的温度输出数据格式。

表 81. FIFO 中的温度输出数据格式

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

下表显示了 **FIFO** 中的时间戳输出数据格式。

表 82. FIFO 中的时间戳输出数据格式

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

如表 82 所示，时间戳数据还包含一些元信息，如果 **CFG-Change** 传感器在 **FIFO** 中不是批处理，则可以使用这些元信息检测 **BDR** 变化：主要传感器和传感器集合的批处理数据率。**BDR_SHUB** 无法通过专用寄存器进行配置。它与通过 **SLAVE0_CONFIG** 传感器集合寄存器的 **SHUB_ODR_[1:0]** 位配置的传感器集合 **ODR** 和有效触发传感器输出数据率（如果使用了内部触发信号，则为加速度计和陀螺仪中的最快数据率）有关。关于 **BDR_SHUB** 的完整说明，请参考下一节中与虚拟传感器相关的内容。

CFG-Change 识别主要或虚拟传感器的输出数据率、批处理数据率或其他配置中的运行时间变更。在应用支持的运行时间更改时，在第一个新的主要传感器或虚拟传感器事件时写入该传感器，然后是时间戳传感器（当时时间戳传感器不是批处理时）。

该传感器可用于将传感器数据关联到器件时间戳，无需每次保存时间戳。它还可用于在嵌入式滤波器稳定时间或其他配置变更（即切换模式、输出数据率等）的情况下，通知用户丢弃数据。

下表显示了 **FIFO** 中的 **CFG-Change** 输出数据格式。

表 83. FIFO 中的 CFG-change 输出数据格式

数据	FIFO_DATA_OUT 寄存器
LPF1_SEL_G	FIFO_DATA_OUT_X_H[0]
FTYPE[2:0]	FIFO_DATA_OUT_X_H[3:1]
G_HM_MODE	FIFO_DATA_OUT_X_H[4]
FS_125	FIFO_DATA_OUT_X_H[5]
FS[1:0]_G	FIFO_DATA_OUT_X_H[7:6]
LPF2_XL_EN	FIFO_DATA_OUT_Y_L[0]
HPCF_XL_[2:0]	FIFO_DATA_OUT_Y_L[3:1]
XL_HM_MODE	FIFO_DATA_OUT_Y_L[4]
XL_ULP_EN	FIFO_DATA_OUT_Y_L[5]
FS[1:0]_XL	FIFO_DATA_OUT_Y_L[7:6]
BDR_SHUB	FIFO_DATA_OUT_Y_H[3:0]
OIS 使能 ⁽¹⁾	FIFO_DATA_OUT_Y_H[5]
陀螺仪启动 ⁽²⁾	FIFO_DATA_OUT_Y_H[6]
FIFO_COMPR_RT_EN	FIFO_DATA_OUT_Y_H[7]
ODR_XL	FIFO_DATA_OUT_Z_L[3:0]
ODR_GY	FIFO_DATA_OUT_Z_L[7:4]
BDR_XL	FIFO_DATA_OUT_Z_H[3:0]
BDR_GY	FIFO_DATA_OUT_Z_H[7:4]

1. 在两种情况下断言 OIS 使能:

1. OIS_ON_EN = 1 且 OIS_EN = 1 (从 UI 接口使能 OIS)
2. OIS_ON_EN = 0 且 OIS_EN_SPI2 = 1 (从 OIS 接口使能 OIS)

2. 在陀螺仪启动阶段 (最长启动时间为 70 ms) 结束时生成的内部信号。

9.6 虚拟传感器

虚拟传感器分为两类：

1. 外部传感器，从传感器集合接口读取；
2. 计步器传感器。

9.6.1 外部传感器和 NACK 传感器

从传感器集合读取的四个外部传感器的数据（最多 18 字节）可以保存在 FIFO 中。

它们是连续的虚拟传感器，如果使用内部触发信号（加速度计或陀螺仪数据准备就绪信号触发传感器集合读取），则批处理数据率（BDR_SHUB）对应于 SLAVE0_CONFIG 寄存器中 SHUB_ODR_[1:0]字段的当前值。此值受限于有效触发传感器输出数据率（加速度计和陀螺仪中的最快数据率）。如果外部传感器不是批处理或使用了外部触发信号，则 BDR_SHUB 置为 0。

下表显示了 BDR_SHUB 字段的可能值。

表 84. BDR_SHUB

BDR_SHUB	BDR [Hz]
0000	未批处理或使用了外部触发信号
0001	12.5
0010	26
0011	52
0100	104

作为主要传感器，外部传感器定义了 FIFO 时基，它们可以触发 FIFO 中辅助传感器的写入（仅当采用批处理且未使用外部触发信号时）。

可使用 SLAVE0_CONFIG、SLAVE1_CONFIG、SLAVE2_CONFIG 和 SLAVE3_CONFIG 传感器集合寄存器的 BATCH_EXT_SENS_0_EN、BATCH_EXT_SENS_1_EN、BATCH_EXT_SENS_2_EN 和 BATCH_EXT_SENS_3_EN 位选择性地使能不同外部传感器的批处理。

每个外部传感器都有专用的 TAG 值和为数据预留的 6 个字节。外部传感器写入 FIFO 的顺序与传感器集合输出寄存器的顺序相同，并且如果从外部传感器读取的字节数小于 6 字节，则用零填充空闲字节。

如果与在 FIFO 中批处理的一个外部传感器通信失败，传感器集合将写入 NACK 传感器而不是在 FIFO 中写入相应的传感器数据。NACK 传感器包含失败从器件的索引（编号 0 至 3），并具有以下输出数据格式。

表 85. FIFO 中的 Nack 传感器输出数据格式

数据	FIFO_DATA_OUT 寄存器
失败从器件索引	FIFO_DATA_OUT_X_L[1:0]
0	FIFO_DATA_OUT_X_L[7:2]
0	FIFO_DATA_OUT_X_H
0	FIFO_DATA_OUT_Y_L
0	FIFO_DATA_OUT_Y_H
0	FIFO_DATA_OUT_Z_L
0	FIFO_DATA_OUT_Z_H

9.6.2 计步器传感器

计步器数据和相关的时间戳可以保存在 FIFO 中。它不是连续频率传感器：步伐检测事件触发其在 FIFO 中的写入。

为了在 FIFO 中使能计步器传感器，用户应：

1. 使能计步器传感器（将 EMB_FUNC_EN_A 嵌入功能寄存器的 PEDO_EN 位置为 1）；



2. 使能计步器批处理（将 EMB_FUNC_FIFO_CFG 嵌入功能寄存器的 PEDO_FIFO_EN 位置为 1）。
下表显示了从 FIFO 读取的计步器数据的格式。

表 86. FIFO 中的计步器输出数据格式

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

9.7 FIFO 模式

通过 FIFO_CTRL4 寄存器的 FIFO_MODE_[2:0] 字段，LSM6DSO FIFO 缓冲器可配置为六种不同的可选工作模式。可用配置确保了高度灵活性，并扩展了可用于应用开发的功能数量。
以下段落描述了 Bypass、FIFO、Continuous、Continuous-FIFO、Bypass-Continuous 和 Bypass-FIFO 模式。

9.7.1 Bypass 模式

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

9.7.2

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。

请按照以下步骤配置 FIFO 模式：

1. 使能以相应批处理数据率将传感器数据保存到 FIFO 中（如果可以配置）；
2. 将 FIFO_CTRL4 寄存器中的 FIFO_MODE_[2:0] 位置为 001b 来使能 FIFO 模式。

当选用此模式时，FIFO 开始采集数据。FIFO_STATUS1 和 FIFO_STATUS2 寄存器根据所存储的采样数来更新。

当 FIFO 已满时，FIFO_STATUS2 寄存器的 DIFF_FIFO_9 位被置为 1，不再有数据存储到 FIFO 缓冲区中。可以通过读取所有 FIFO_DATA_OUT 寄存器（78h 至 7Eh）并达到 FIFO_STATUS1 和 FIFO_STATUS2 寄存器 DIFF_FIFO_[9:0] 位指定的次数来获取数据。

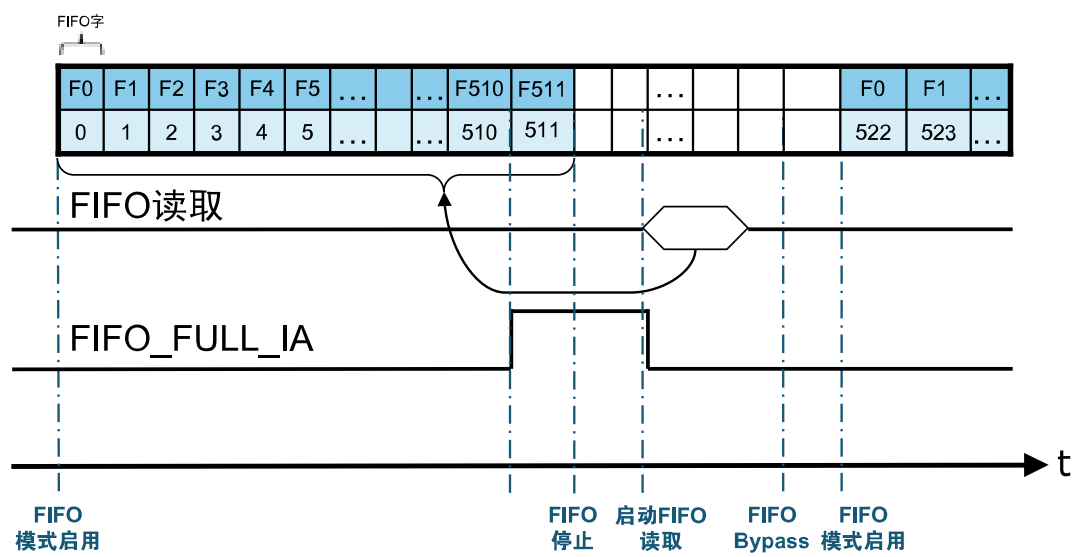
利用 FIFO_STATUS2 寄存器的 FIFO_WTM_IA 位，如果应用要求 FIFO 中采样数较低，则数据还可在达到阈值（FIFO_CTRL1 和 FIFO_CTRL2 寄存器中的 WTM[8:0]）时恢复。

如果 FIFO_CTRL2 寄存器的 STOP_ON_WTM 位置为 1，则 FIFO 空间大小被限制为 FIFO_CTRL1 和 FIFO_CTRL2 寄存器中 WTM[8:0] 位的值。这种情况下，如果下一次 FIFO 写操作时 FIFO 中的采样数达到或超过 WTM[8:0] 值，那么 FIFO_STATUS2 寄存器的 FIFO_FULL_IA 位会被置为高电平。

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

图 28. FIFO 模式（STOP_ON_WTM = 0）显示了 FIFO 模式使用示例：只有一个传感器的数据存储到 FIFO 中。这些条件下，FIFO 缓冲区中可存储 512 个采样（压缩算法禁用）。在标记为 510 后，FIFO_STATUS2 寄存器的 FIFO_FULL_IA 位升高，以通知 FIFO 缓冲区将在下一次 FIFO 写操作时达到全满。在 FIFO 已满（FIFO_DIFF_9 = 1）后，数据采集停止。

图 28. FIFO 模式 (STOP_ON_WTM = 0)



9.7.3 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）：

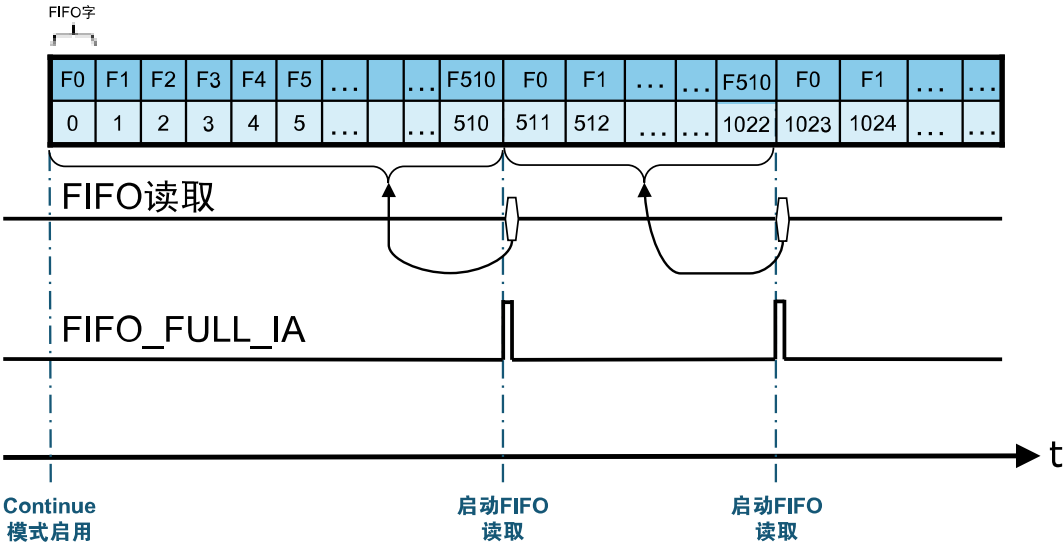
1. 使能以相应批处理数据率将传感器数据保存到 FIFO 中（如果可以配置）；
2. 将 FIFO_CTRL4 寄存器中的 FIFO_MODE_[2:0] 位置为 110b 来使能 FIFO 模式。

当选用此模式时，FIFO 连续采集数据。FIFO_STATUS1 和 FIFO_STATUS2 寄存器根据所存储的采样数来更新。当下一个 FIFO 写操作会使 FIFO 全满或生成 FIFO 溢出时，FIFO_STATUS2 寄存器的 FIFO_FULL_IA 位变为 1。FIFO_STATUS2 寄存器的 FIFO_OVR_IA 和 FIFO_OVR_LATCHED 位表示至少有一个 FIFO 字被覆盖以存储新数据。通过读取 FIFO_DATA_OUT（78h 至 7Eh）寄存器获取 FIFO_STATUS1 和 FIFO_STATUS2 寄存器的 DIFF_FIFO_[9:0] 位所指定的次数，数据可在 FIFO_FULL_IA 事件后恢复。利用 FIFO_STATUS2 寄存器的 FIFO_WTM_IA 位，数据还可在达到阈值（FIFO_CTRL1 和 FIFO_CTRL2 寄存器中的 WTM[8:0]）时恢复。如果 FIFO_CTRL2 寄存器的 STOP_ON_WTM 位置为 1，则 FIFO 空间大小被限制为 FIFO_CTRL1 和 FIFO_CTRL2 寄存器中 WTM[8:0] 位的值。这种情况下，如果下一次 FIFO 写操作时 FIFO 中的采样数达到或超过 WTM[8:0] 值，那么 FIFO_STATUS2 寄存器的 FIFO_FULL_IA 位会被置为高电平。

图 29. Continue 模式 显示了 Continue 模式使用的示例。示例中，仅一个传感器的数据存储在 FIFO 中，在发生 FIFO_FULL_IA 事件时读取 FIFO 采样且读取速度快于 $1 * ODR$ ，因此无数据丢失。这些条件下，所存储的采样数为 511。



图 29. Continue 模式





9.7.4

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。

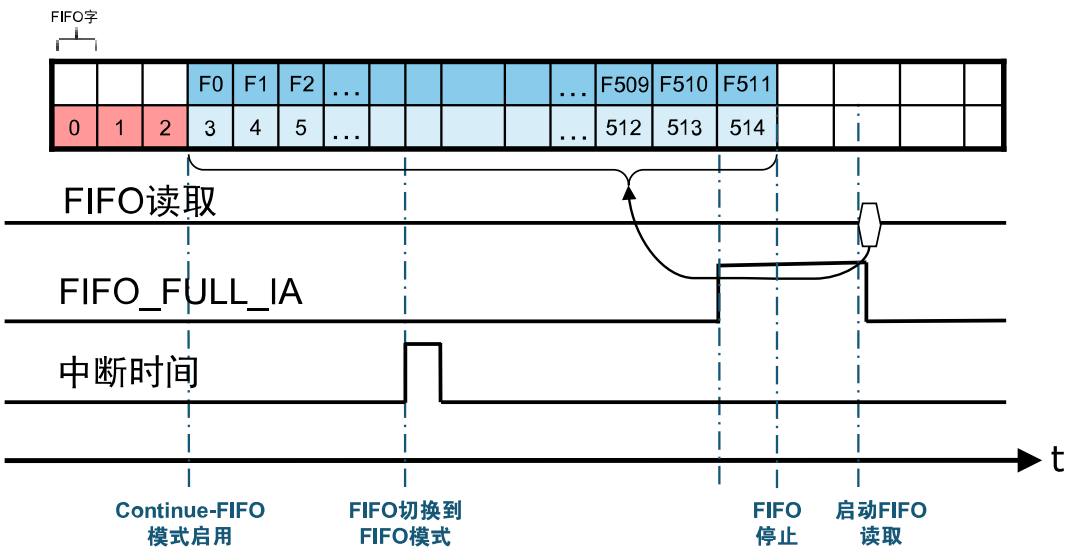
事件条件可为以下之一：

- 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。

Continuous-to-FIFO 模式对中断信号的边沿感应。在第一次中断事件时，FIFO 从 Continuous 模式变为 FIFO 模式，并维持该模式，直到 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. 使能以相应批处理数据率将传感器数据保存到 FIFO 中（如果可以配置）；
3. 将 FIFO_CTRL4 寄存器中 FIFO_MODE_[2:0] 位置为 011b 来使能 **Continue-FIFO** 模式。

在 **Continuous-FIFO** 模式下，FIFO 缓冲区会继续填入数据。当 FIFO 将在下一次 FIFO 写操作时全满或溢出时，FIFO_FULL_IA 位变为高电平。

如果 FIFO_CTRL2 寄存器的 STOP_ON_WTM 位置为 1，则 FIFO 空间大小被限制为 FIFO_CTRL1 和 FIFO_CTRL2 寄存器中 WTM[8:0] 位的值。这种情况下，如果下一次 FIFO 写操作时 FIFO 中的采样数达到或超过 WTM[8:0] 值，那么 FIFO_STATUS2 寄存器的 FIFO_FULL_IA 位会被置为高电平。

发生触发事件时，可观察到两种不同的情况：

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

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

9.7.5 Bypass-Continue 模式

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

事件条件可为以下之一：

- 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）：

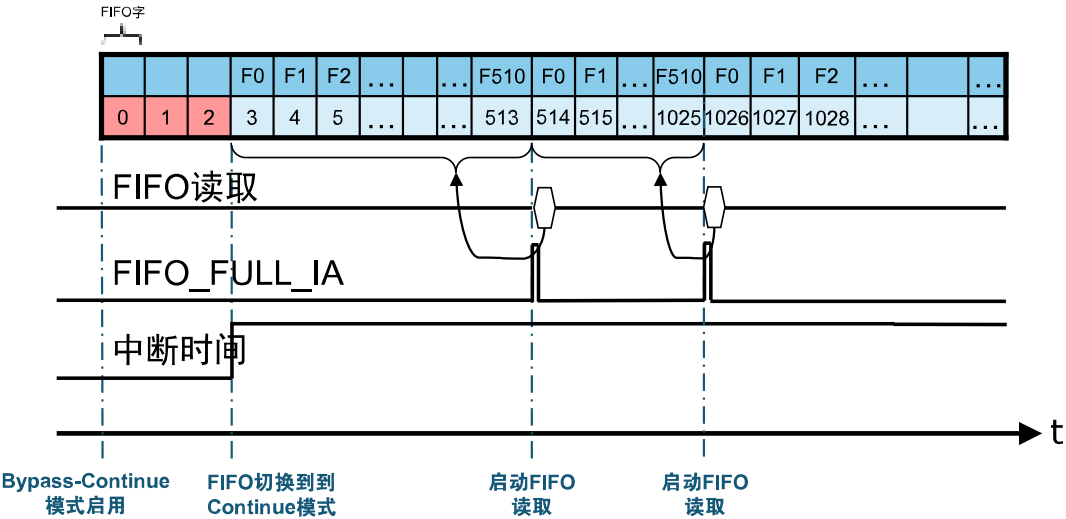
1. 配置先前所述的其中一个事件；
2. 使能以相应批处理数据率将传感器数据保存到 FIFO 中（如果可以配置）；
3. 将 FIFO_CTRL4 寄存器中 FIFO_MODE[2:0] 位置为 100b 来使能 FIFO 旁路-连续模式。

当出现触发条件且缓冲器切换至 **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 满或溢出时，FIFO_FULL_IA 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。



图 31. Bypass-Continue 模式



9.7.6 Bypass-FIFO 模式

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

事件条件可为以下之一：

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

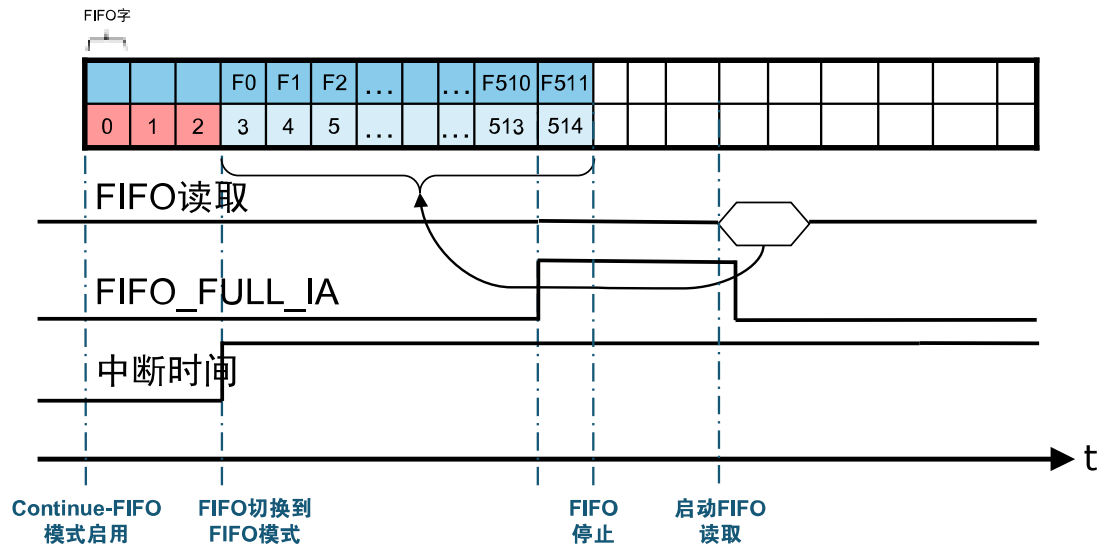
按照这些步骤进行 Bypass-FIFO 模式配置（如果加速度计/陀螺仪数据准备就绪被用作 FIFO 触发）：

1. 配置先前所述的其中一个事件；
2. 使能以相应批处理数据率将传感器数据保存到 FIFO 中（如果可以配置）；
3. 将 FIFO_CTRL4 寄存器中 FIFO_MODE_[2:0]位置为 111b 来使能 Bypass-FIFO 模式。

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

Bypass-to-FIFO 可用来分析生成中断的采样历史。

图 32. Bypass-FIFO 模式





9.8 从 FIFO 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/MIPI I3CSM 输出缓冲器。

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

从 FIFO 中恢复数据的建议方法如下：

1. 读取 FIFO_STATUS1 和 FIFO_STATUS2 寄存器来检查 FIFO 中存储了多少字。此信息在 DIFF_FIFO_[9:0]位中。
2. 对于 FIFO 中的每个字，读取 FIFO 字（标记和输出数据）并基于 FIFO 标记解读它。
3. 转至步骤 1。

通过从 FIFO 输出寄存器执行一定次数的读操作，直至缓冲器为空（FIFO_STATUS1 和 FIFO_STATUS2 寄存器的 DIFF_FIFO_[9:0]位等于 0），可以恢复全部 FIFO 内容。

建议避免在 FIFO 为空时读取它。

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



9.9 FIFO 水印阈值

FIFO 阈值是 LSM6DSO FIFO 的功能，可用于检查 FIFO 中的采样数何时达到定义的水印阈值水平。

FIFO_CTRL1 和 FIFO_CTRL2 寄存器的 WTM[8:0]中有水印阈值水平。WTM[8:0]字段的分辨率为 7 字节，相当于一个完整的 FIFO 字。因此，用户能够在 0 至 511 之间选择所需的值。

FIFO_STATUS2 寄存器的位 FIFO_WTM_IA 表示水印状态。如果 FIFO 中的字数达到或超过水印水平，则此位被置为高电平。通过将 FIFO_CTRL2 寄存器的 STOP_ON_WTM 位置为 1，FIFO 空间大小可由阈值水平来限制。



图 33. FIFO 阈值 (STOP_ON_WTM = 0)

WTM[8:0] = 21
STOP_ON_WTM = 0

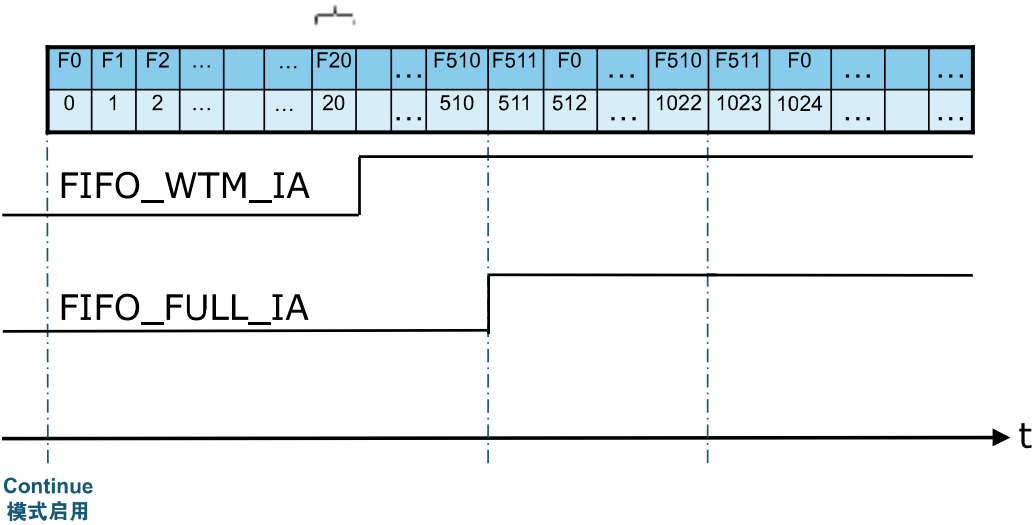


图 33. FIFO 阈值 ($STOP_ON_WTM = 0$) 显示了只存储加速度计 (或陀螺仪) 数据时, FIFO 阈值水平使用的示例。FIFO_CTRL2 寄存器中 $STOP_ON_WTM$ 位置为 0。利用 $WTM[8:0]$ 位, 阈值水平设置为 21。达到 21 后 (FIFO 中有 21 个字), FIFO_STATUS2 寄存器的 $FIFO_WTM_IA$ 位升高。因此, $STOP_ON_WTM$ 位置 0, FIFO 不会在第 21 个数据组时停止, 而是会继续存储数据, 直至 $FIFO_FULL_IA$ 标记被置为高电平。



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

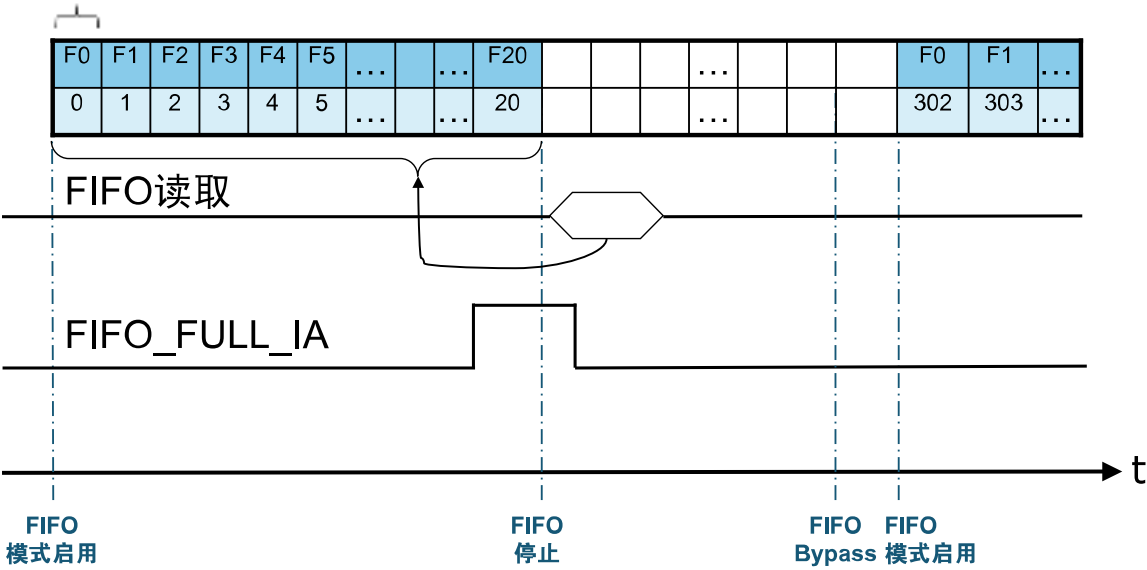


图 34. FIFO 模式下的 FIFO 阈值 ($STOP_ON_WTM = 1$) 显示了 FIFO 模式下 FIFO 阈值使用的示例，其中 FIFO_CTRL2 寄存器中 STOP_ON_WTM 位被置为 1。本例中只存储了加速度计（或陀螺仪）数据。利用 WTM[8:0] 位，阈值水平设置为 21，并定义当前 FIFO 空间大小。在 FIFO 模式下，数据保存在 FIFO 缓冲区，直至 FIFO 已满。在 FIFO 中保存下一个数据时，FIFO_STATUS2 寄存器的 FIFO_FULL_IA 位升高，将生成 FIFO 已满或溢出条件。当 FIFO 已满时，FIFO_STATUS2 寄存器的 FIFO_WTM_IA 位升高。



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

WTM[8:0] = 21
STOP_ON_WTM = 1

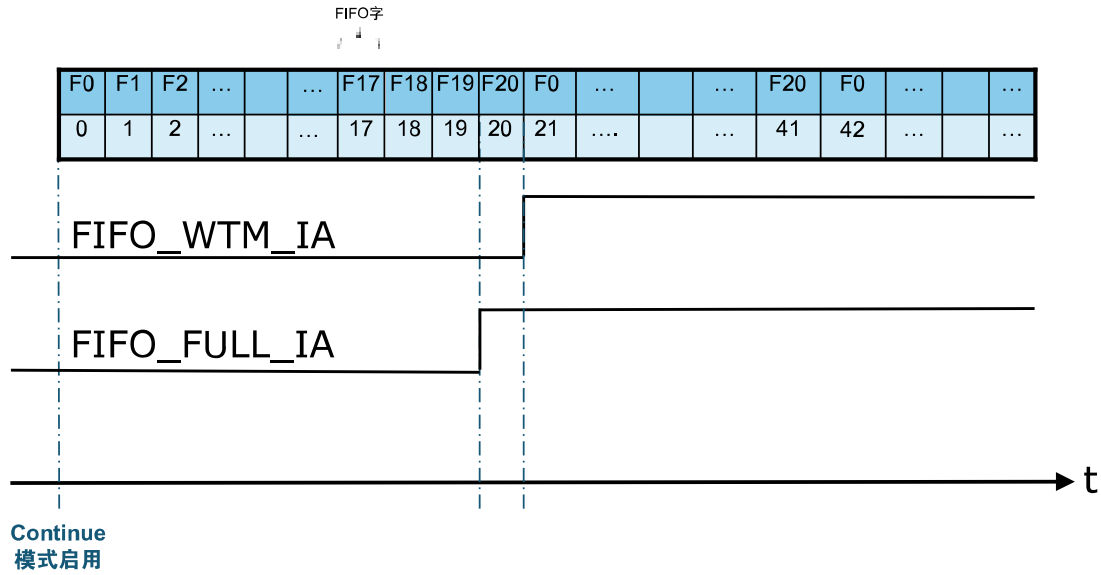


图 35. Continue 模式下的 FIFO 阈值 ($STOP_ON_WTM = 1$) 显示了 Continuous 模式下 FIFO 阈值使用的示例，其中 FIFO_CTRL2 寄存器的 STOP_ON_WTM 位被置为 1。本例中只存储了加速度计（或陀螺仪）数据。利用 WTM[8:0] 位，阈值水平设置为 21。在 FIFO 中保存下一个数据时，FIFO_STATUS2 寄存器的 FIFO_FULL_IA 位升高，FIFO 已满。当 FIFO 已满时，FIFO_STATUS2 寄存器的 FIFO_WTM_IA 位升高。如果不从 FIFO 检索数据，新数据（标记为采样 21）将覆盖 FIFO 中的旧数据（标记为采样 F0）。

9.10 FIFO 压缩

FIFO 压缩是一种嵌入式算法，允许在 FIFO 中保存最多 3 倍数量的加速度计和陀螺仪数据。压缩算法自动分析传感器波形的斜率（两个连续采样之间的差值），并基于斜率在 FIFO 中应用数据压缩。

可通过 EMB_FUNC_EN_B 嵌入功能寄存器中 FIFO_COMPR_EN 位和 FIFO_CTRL2 寄存器中 FIFO_COMPR_RT_EN 位的置位使能 FIFO 中加速度计和陀螺仪数据的 FIFO 压缩。当激活时，压缩作用于加速度计和陀螺仪数据，压缩程度互不相干。

加速度计和陀螺仪批处理数据率（BDR）可以独立配置，但以下配置不支持压缩算法：

1. 在 FIFO 中批处理加速度计和陀螺仪数据且 $\max(ODR_XL, ODR_G) \geq 1.66 \text{ kHz}$;
2. 在 FIFO 中只批处理加速度计或陀螺仪数据且 $\max(ODR_XL, ODR_G) \geq 3.33 \text{ kHz}$ 。

FIFO 压缩支持三种不同压缩水平：

- **NC**，不压缩，如果实际数据与先前数据之间的差异超过 128 LSB：在一个 FIFO 字中保存一个传感器采样；
- **2xC**，低压缩率，如果实际数据与先前数据之间的差异介于 16 与 128 LSB 之间：在一个 FIFO 字中保存两个传感器采样；
- **3xC**，高压缩率，如果实际数据与先前数据之间的差异少于 16 LSB：在一个 FIFO 字中保存三个传感器采样。

9.10.1

时间相关性

根据压缩程度，有五种不同的标签（适用于每种主要传感器）：

- **NC**，不压缩，关联到实际时隙；
- **NC_T_2**，不压缩，关联到两倍于先前时隙的时隙；
- **NC_T_1**，不压缩，关联到先前时隙；
- **2xC**，低压缩率；
- **3xC**，高压缩率。

所有 **NC** 标记均有助于理解时隙相关性。通过解码传感器标签，可以了解生成数据的时间帧。

在第一个批处理事件发生时，压缩算法将未压缩字（**NC**）写入 **FIFO**。此后，算法分析波形斜率，有三种可能的 **FIFO** 条目：

- **3xC** 数据写入，其中包含 **diff(i)**、**diff(i - 1)** 和 **diff(i - 2)**；
- **2xC** 数据写入，其中包含 **diff(i - 1)** 和 **diff(i - 2)**；
- **NC_T_2** 数据写入，其中包含 **data(i - 2)**。

可以在配置发生变化时或用户需要临时禁用运行时间 **FIFO** 压缩时，通过 **FIFO_CTRL2** 寄存器中 **FIFO_COMPRT_RT_EN** 位的置位写入不压缩标签传感器 **NC_T_1**。

下表总结了每个标签的相关数据和时隙。

表 87. FIFO 压缩标签和相关数据

标签传感器	时隙数据
NC	data(i)
NC_T_1	data(i - 1)
NC_T_2	data(i - 2)
2xC	diff(i - 2), diff(i - 1)
3xC	diff(i - 2), diff(i - 1), diff(i)

如表 87 所示，使用 **FIFO** 压缩会产生 $2 / \text{BDR}$ 的延迟，因为压缩将作用于一个 $3 \times \text{BDR}$ 的窗口。

9.10.2 数据格式

压缩数据的 FIFO 字包含其相对于之前数据的斜率信息：

$$data(i) = diff(i) + data(i - 1)$$

因此，在执行解压缩任务时，必须保存最后一个解码数据，即以上公式中的 **data(i-1)**。

下表总结了 2xC 压缩数据在 FIFO 中的输出数据格式。

表 88. FIFO 中的 2xC 压缩数据输出数据格式

数据	公式
diffx(i - 2)	8bit_signed(FIFO_DATA_OUT_X_L)
diffy(i - 2)	8bit_signed(FIFO_DATA_OUT_X_H)
diffz(i - 2)	8bit_signed(FIFO_DATA_OUT_Y_L)
diffx(i - 1)	8bit_signed(FIFO_DATA_OUT_Y_H)
diffy(i - 1)	8bit_signed(FIFO_DATA_OUT_Z_L)
diffz(i - 1)	8bit_signed(FIFO_DATA_OUT_Z_H)

下表总结了 3xC 压缩数据在 FIFO 中的输出数据格式。

表 89. FIFO 中的 3xC 压缩数据输出数据格式

数据	公式
diffx(i - 2)	0bit_signed(FIFO_DATA_OUT_X[4:5])
diffy(i - 2)	5bit_signed(FIFO_DATA_OUT_X[9:5])
diffz(i - 2)	5bit_signed(FIFO_DATA_OUT_X[14:10])
diffx(i - 1)	0bit_signed(FIFO_DATA_OUT_Y[4:5])
diffy(i - 1)	5bit_signed(FIFO_DATA_OUT_Y[9:5])
diffz(i - 1)	5bit_signed(FIFO_DATA_OUT_Y[14:10])
diffx(i)	0bit_signed(FIFO_DATA_OUT_Z[4:5])
diffy(i)	5bit_signed(FIFO_DATA_OUT_Z[9:5])
diffz(i)	5bit_signed(FIFO_DATA_OUT_Z[14:10])

在上面的表格中：

- FIFO_DATA_OUT_X[15:0] = FIFO_DATA_OUT_X_L + FIFO_DATA_OUT_X_H << 8
- FIFO_DATA_OUT_Y[15:0] = FIFO_DATA_OUT_Y_L + FIFO_DATA_OUT_Y_H << 8
- FIFO_DATA_OUT_Z[15:0] = FIFO_DATA_OUT_Z_L + FIFO_DATA_OUT_Z_H << 8

9.10.3 在运行时间禁用 FIFO 压缩

FIFO 压缩会在 FIFO 中写入传感器数据时产生 $2 / \text{BDR}$ 的延迟。当用户想要以低延迟清空 FIFO 时，不建议使用 FIFO 压缩。

如果高延迟和低延迟均可使用，则可以在运行时间以更方便的方式配置 FIFO。

为了从使能的压缩算法切换至禁用的压缩算法（无延迟），可以在运行时间修改 FIFO_COMPR_RT_EN 位。切换按设备配置修改进行管理。FIFO 在修改后的第一个 BDR 事件时写入 CFG-Change 传感器。这种情况下，使用标签 NC、NC_T_2 或 NC_T_1 在同一时隙写入尚未保存的所有数据。

下表显示了运行时间禁用的压缩算法示例。这种情况下，应在 FIFO 中批处理主要传感器、CFG-Change 传感器和时间戳传感器。在时刻 $t(i-1)$ 与时刻 $t(i)$ 之间，FIFO 压缩的运行时间禁用。如上文所述，所有尚未保存的数据被写入到 CFG-Change 和时间戳传感器之后的同一时隙。

表 90. 禁用运行时间压缩的示例

时间	FIFO_COMPR_RT_EN	传感器	FIFO_DATA_OUT
...	1	...	...
$t(i-3)$	1	3xC	diff(i-5), diff(i-4), diff(i-3)
$t(i-2)$	1	-	-
$t(i-1)$	1	-	-
异步事件	0	-	-
$t(i)$	0	CFG_Change	CFG-change 数据
		时间戳	时间戳数据
		NC_T_2	data(i-2)
		NC_T_1	data(i-1)
		NC	data(i)
$t(i+1)$	0	NC	data(i+1)
$t(i+2)$	0	NC	data(i+2)

9.10.4 使能了 FIFO 压缩的 CFG-Change 传感器

在将配置修改应用于设备时，应用处理器必须区分原配置的数据和新配置的数据。对该任务应用了与 FIFO_COMPR_RT_EN 修改相同的方法，如下表所示。这种情况下，应在 FIFO 中批处理主要传感器、CFG-Change 传感器和时间戳传感器。在时刻 $t(i-1)$ 与时刻 $t(i)$ 之间应用新的器件配置。如上文所述，所有尚未保存的数据被写入到 CFG-Change 和时间戳传感器之后的同一时隙。此后，FIFO 压缩算法按照预期重新开始操作。

表 91. 使能了 FIFO 压缩的设备配置修改示例

时间	FIFO_COMPR_RT_EN	传感器	FIFO_DATA_OUT
...	1	...	...
$t(i-3)$	1	3xC	diff(i-5), diff(i-4), diff(i-3)
$t(i-2)$	1	-	-
$t(i-1)$	1	-	-
异步事件 (CFG-Change)	1	-	-
$t(i)$	1	CFG_Change	CFG-change 数据
		时间戳	时间戳数据
		NC_T_2	data(i-2)
		NC_T_1	data(i-1)
		NC	data(i)
$t(i+1)$	1	-	-
$t(i+2)$	1	-	-
$t(i+3)$	1	3xC	diff(i+1), diff(i+2), diff(i+3)

9.10.5 未压缩数据率

可以配置压缩算法，以便通过 FIFO_CTRL2 寄存器的 UNCOPTR_RATE_[1:0] 字段保证以特定周期（8、16 和 32 BDR 事件）写入未压缩数据。

当有可能发生 FIFO 溢出事件时，FIFO 中未压缩数据率的使用对于数据重建十分有用：如果发生溢出且参考未压缩数据被覆盖，则在新的未压缩数据写入 FIFO 前不可能重建当前数据。UNCOPTR_RATE_[1:0] 配置压缩算法，以特定速率写入未压缩数据，从而确保每 8、16 或 32 个采样有至少一个未压缩数据。

表 92. UNCOPTR_RATE 配置

UNCOPTR_RATE_[1:0]	NC 数据写入
00	不强制 NC 数据
01	每 8 BDR 有至少一个 NC 数据
10	每 16 BDR 有至少一个 NC 数据
11	每 32 BDR 有至少一个 NC 数据

9.10.6 FIFO 压缩初始化

当 FIFO 设置为 Bypass 模式时，必须通过 EMB_FUNC_INIT_B 嵌入功能寄存器中 FIFO_COMPR_INIT 位的置位重新初始化压缩算法。

9.10.7 FIFO 压缩示例

下表提供了压缩功能使能时可从 FIFO 读取的数据的基本数值示例。在这个示例中，只有加速度计传感器保存在 FIFO 中，为其配置了满量程 $\pm 2g$ 。



表 93. FIFO 压缩示例

时间 [n/ODR]	FIFO_DATA_OUT 寄存器							数据分析				
	TAG_ SENSOR_[4:0]	X_L	X_H	Y_L	Y_H	Z_L	Z_H	压缩	加速度 X [LSB]	加速度 Y [LSB]	加速度 Z [LSB]	延迟 [n/ODR]
0	0x02	0x4F	0x01	0x84	0x00	0x85	0x3C	NC	335	132	15493	0
3	0x06	0x61	0x01	0x96	0x00	0x86	0x40	NC_T_2	353	150	16518	2
4	0x09	0x5C	0x0B	0x43	0x0D	0x33	0xF8	3xC	349	144	16520	2
									352	154	16523	1
									339	155	16521	0
7	0x09	0x9E	0x04	0x03	0xEC	0xC2	0x03	3xC	337	159	16522	2
									340	159	16517	1
									342	157	16517	0
10	0x08	0xFB	0x0A	0x15	0x0E	0xEE	0xF0	2xC	337	167	16538	2
									351	149	16522	1
12	0x09	0x80	0xD8	0x64	0x20	0x97	0x2B	3xC	351	153	16512	2
									355	156	16520	1
									346	152	16530	0

在第一个批处理事件发生时，压缩算法将未压缩字（NC）写入 FIFO，无延迟。此后，算法分析波形斜率，有三种可能的 FIFO 条目：3xC、2xC 和 NC_T_2。本例中没有带 NC_T_1 标记的未压缩字，因为运行时间配置没有变化。

保存在 FIFO 中的第二个采样是具有 2 个采样的延迟的未压缩字（NC_T_2）：此 FIFO 条目包含完整加速度计数据（未经任何压缩）。

然后，由于加速度计数据斜率较低，压缩算法开始压缩加速度计数据：应从当前采样之前的最后一个采样开始重建加速度计数据（第一个压缩数据表达为与 NC_T_2 数据的差异，第二个压缩数据表达为与第一个压缩数据的差异，以此类推）。

如例中所示，压缩算法使用三级深度缓冲区：如果在 FIFO 中以 2xC 压缩级别写入，则只有上一个数据（延迟 1）和两倍于上一个数据（延迟 2）的数据保存在 FIFO 字中。

该示例还显示了 FIFO 压缩的优势：以交错的 ODR 在 FIFO 中写入采样，通过主机处理器比正常 FIFO 使用更好地限制干扰。



9.11 时间戳相关性

可通过三种不同方法重建 FIFO 流的时间戳：

1. 基本法，只使用时间戳传感器信息；
2. 存储器保存法，基于 FIFO_DATA_OUT_TAG 中的 TAG_CNT 字段
3. 混合法，基于 TAG_CNT 字段和抽取时间戳传感器的组合使用

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

存储器保存法只使用 TAG_CNT 信息，并且当 TAG_CNT 值增加时，应按如下方式更新在软件层保存的时间戳：

$$timestamp = timestamp(i-1) + \frac{1}{\max(BDR_{XL}, BDR_{GY}, BDR_{SHUB})}$$

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

当可能发生溢出条件时，不建议使用此方法。

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

10 温度传感器

器件具有内部温度传感器，适用于环境温度测量。

如果加速度计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：
 - 如果加速度计配置为超低功耗或低功耗模式且其 ODR 低于 52 Hz，则温度数据率等于配置的加速度计 ODR；
 - 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。

温度传感器数据也可以采用可配置的批处理数据率存储在 FIFO 中（详细信息见第 9 节 先进先出（FIFO）缓冲区）。

10.1 温度数据计算示例

下表提供了在不同环境温度值下从温度数据寄存器中读取数据的几个基本示例。本表中所列值是在理想器件校准的假设下给出的（即，无偏移，无增益误差，……）。

表 94. 输出数据寄存器内容 vs. 温度

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



11 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。

11.1 加速度计自检 (UI) – 模式 1、2

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。

当器件配置为模式 1 或模式 2 时，只能从主接口配置加速度计自检功能。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。

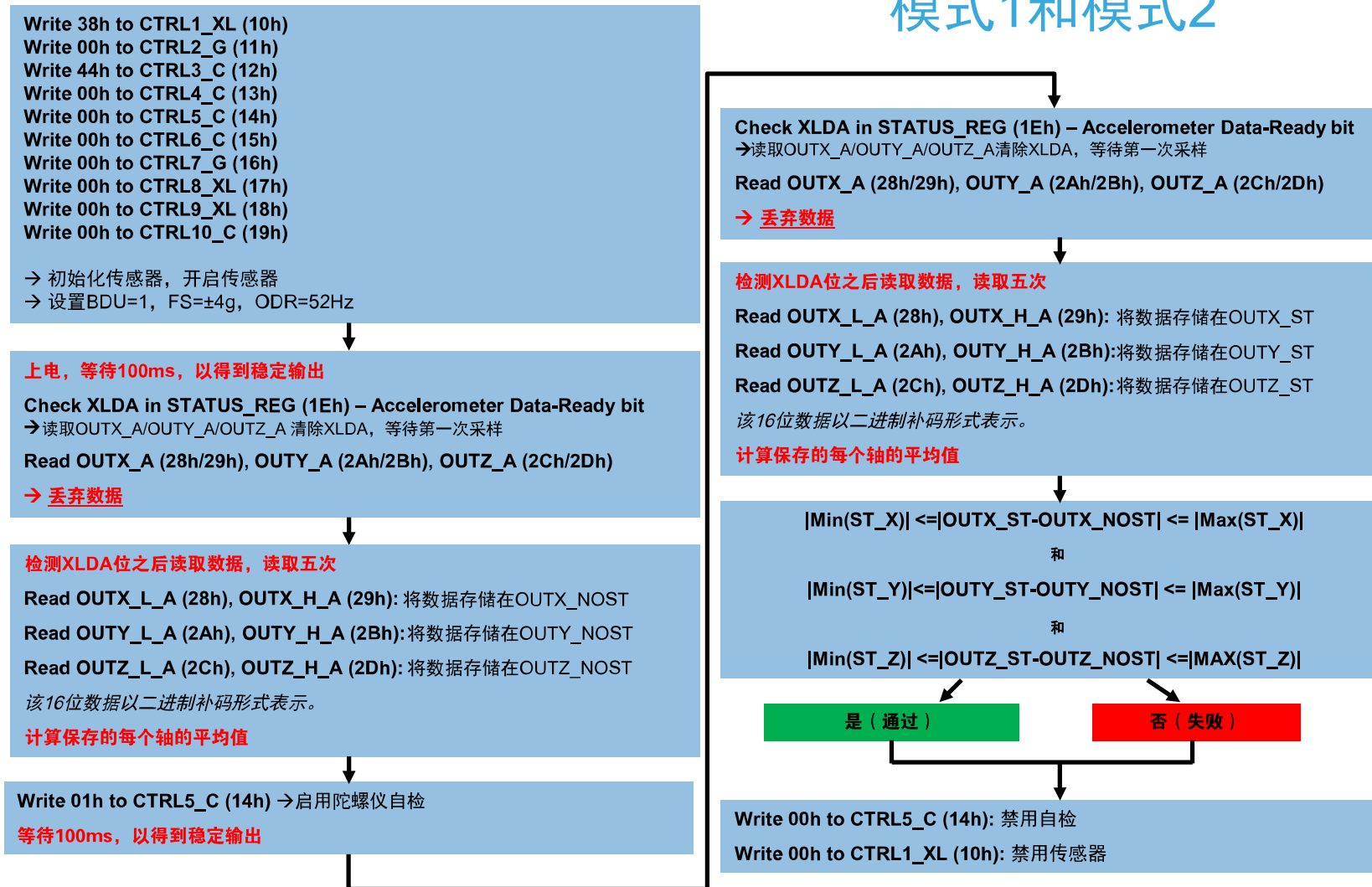
当加速度计自检功能激活时，传感器的输出由作用在传感器上的加速度和静电测试力的代数和给出。

模式 1 或模式 2 下的完整加速度计自检过程如图 36. 加速度计自检步骤 (UI) 中所示。

图 36. 加速度计自检步骤 (UI)

注：本程序中的所有读/写操作都必须通过
主要I²C/SPI/I3C接口执行

加速度计UI自检 模式1和模式2



11.2 OIS 链接通后的加速度计自检 (UI) – 模式 4

如果 UI 链和 OIS 链连通，则必须从主接口通过 CTRL5_C 寄存器的 ST[1:0]_XL 位使能加速度计自检功能。不能同时从两个接口使能（禁止条件）。

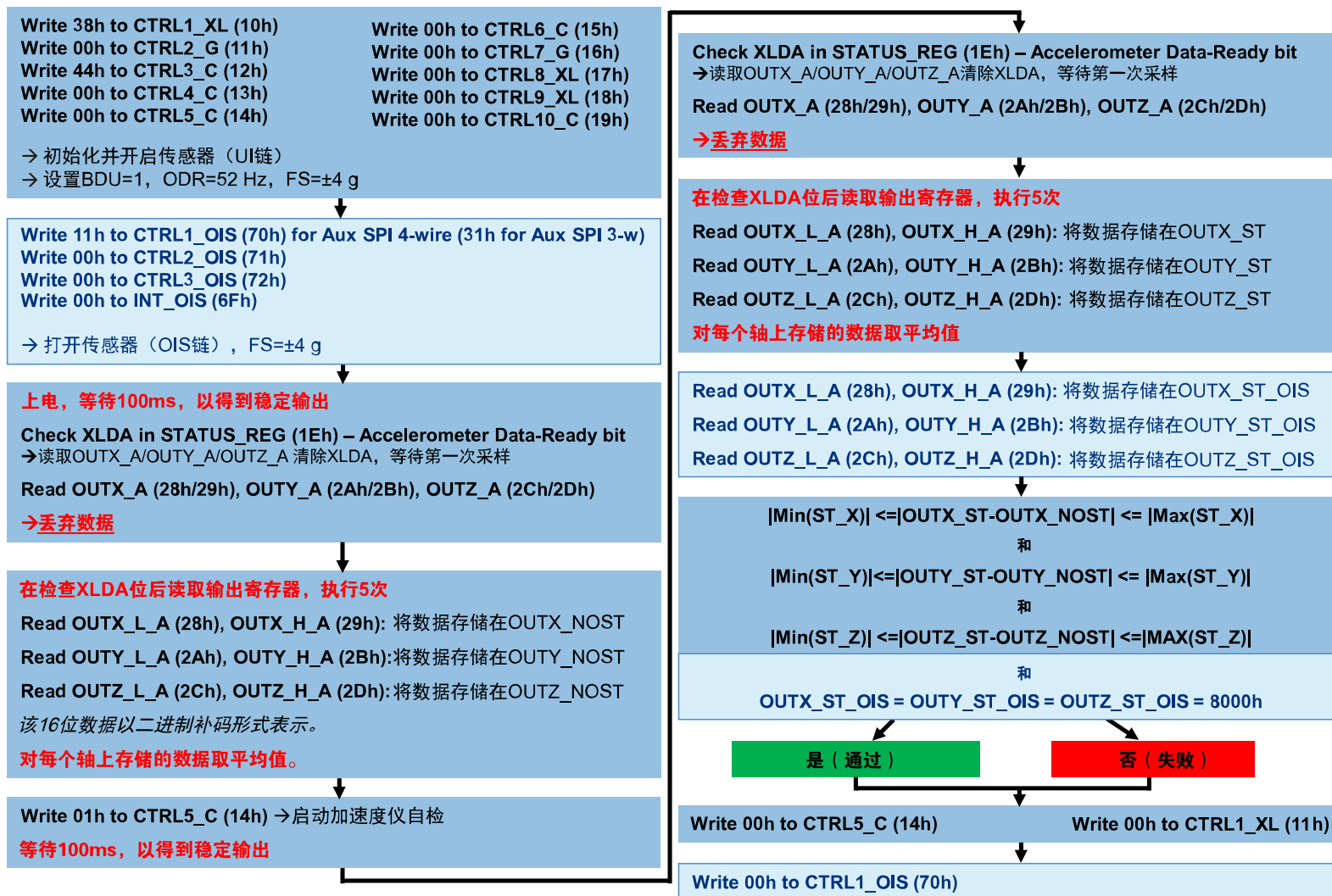
当 OIS 链连通时，UI 链上的推荐加速度计自检流程如图 37. OIS 链连通时的加速度计自检流程 (UI) 中所示。

图 37. OIS 链连通时的加速度计自检流程 (UI)

注:

- 所有黑色读/写操作都必须通过主要PC/SPI接口执行
- 所有蓝色读/写操作都必须通过辅助SPI接口执行

加速度计UI自检模式4, UI和OIS链连通



根据保密协议 - 不可复制



11.3 加速度计自检 (OIS) – 模式 4

当 UI 链断开时, 可通过辅助 SPI 接口, 将 INT_OIS 寄存器的 ST[1:0]_XL_OIS 位置位, 从而使能 OIS 链上的加速度计自检功能。当 ST[1:0]_XL_OIS 位被设定为 00b 时, 自检功能关闭; 当 ST[1:0]_XL_OIS 位被置为 01b (正符号自检) 或 11b (负符号自检) 时, 该功能使能。

当 UI 链断开时, OIS 链上的完整加速度计自检流程如图 38. 加速度计自检步骤 (OIS) 中所示。仅当 UI 读出链断开 (CTRL1_XL 寄存器中的 ODR_XL[1:0] = 0000b, CTRL2_G 寄存器中的 ODR_G[1:0] = 0000b) 时, 才能执行此流程。

图 38. 加速度计自检步骤 (OIS)

加速度计OIS自检模式4, OIS链连通且UI链断开

注:

- 本程序中的所有读/写操作都必须通过**辅助SPI接口执行**
- 仅当**UI读出链断开** (CTRL1_XL中的ODR_XL[3:0] = 0000b) 时, 才能执行此程序。

Write 11h to CTRL1_OIS (70h) for Aux SPI 4-wire (31h for Aux SPI 3-w)
Write 00h to CTRL2_OIS (71h)
Write 80h to CTRL3_OIS (72h)
Write 00h to INT_OIS (6Fh)

→ 初始化并启动传感器
→ FS=±4g (ODR默认6.66kHz)

上电, 等待100ms, 以得到稳定输出

Check XLDA in STATUS_SPIAux (1Eh) – Accelerometer Data-Ready Bit
→ 读取OUTX_A/OUTY_A/OUTZ_A 清除XLDA, 等待第一次采样
Read OUTX_A (28h/29h), OUTY_A (2Ah/2Bh), OUTZ_A (2Ch/2Dh)
→ **丢弃数据**

检测XLDA位之后读取数据, 读取五次

Read OUTX_L_A (28h), OUTX_H_A (29h): 将数据存储存储在OUTX_NOST_OIS
Read OUTY_L_A (2Ah), OUTY_H_A (2Bh): 将数据存储存储在OUTY_NOST_OIS
Read OUTZ_L_A (2Ch), OUTZ_H_A (2Dh): 将数据存储存储在OUTZ_NOST_OIS
该16位数据以二进制补码形式表示。

计算保存的每个轴的平均值

Write 01h to INT_OIS (6Fh) → 启动加速度计自检
等待 100 ms

Check XLDA in STATUS_SPIAux (1Eh) – XL Data-Ready bit
→ 读取OUTX_A/OUTY_A/OUTZ_A清除XLDA, 等待第一次采样
Read OUTX_A (28h/29h), OUTY_A (2Ah/2Bh), OUTZ_A (2Ch/2Dh)
→ **丢弃数据**

检测XLDA位之后读取数据, 读取五次

Read OUTX_L_A (28h), OUTX_H_A (29h): 将数据存储存储在OUTX_ST_OIS
Read OUTY_L_A (2Ah), OUTY_H_A (2Bh): 将数据存储存储在OUTY_ST_OIS
Read OUTZ_L_A (2Ch), OUTZ_H_A (2Dh): 将数据存储存储在OUTZ_ST_OIS
该16位数据以二进制补码形式表示。
计算保存的每个轴的平均值

$|\text{Min}(\text{ST_X})| \leq |\text{OUTX_ST_OIS} - \text{OUTX_NOST_OIS}| \leq |\text{Max}(\text{ST_X})|$
和
 $|\text{Min}(\text{ST_Y})| \leq |\text{OUTY_ST_OIS} - \text{OUTY_NOST_OIS}| \leq |\text{Max}(\text{ST_Y})|$
和
 $|\text{Min}(\text{ST_Z})| \leq |\text{OUTZ_ST_OIS} - \text{OUTZ_NOST_OIS}| \leq |\text{Max}(\text{ST_Z})|$

是 (通过)

否 (失败)

Write 00h to INT_OIS (6Fh): 禁用自检
Write 00h to CTRL1_OIS (70h): 禁用传感器



11.4 陀螺仪自检 (UI) – 模式 1、2

陀螺仪自检可以测试陀螺仪传感器的机械和电气部件。当此功能激活时，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。

当器件配置为模式 1 或模式 2 时，只能从主接口配置陀螺仪自检功能。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。

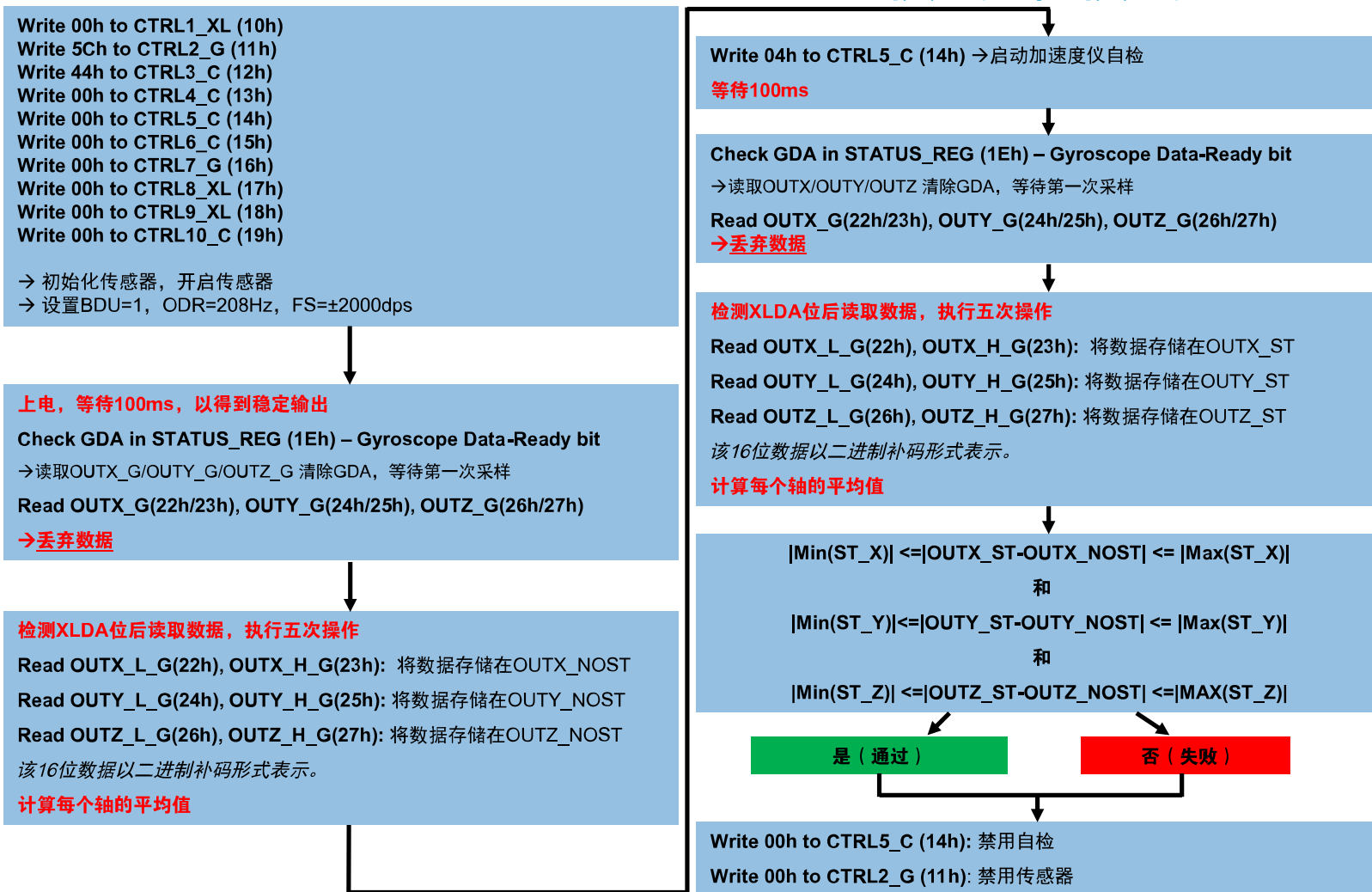
当陀螺仪自检功能激活时，传感器的输出由作用在传感器上的角速度和静电测试力的代数和给出。

模式 1 或模式 2 下的完整陀螺仪自检过程如图 39. 陀螺仪自检步骤 (UI) 中所示。

图 39. 陀螺仪自检步骤 (UI)

本程序中的所有读/写操作都必须通过
主要I²C/SPI/I³C接口执行

陀螺仪UI自检 模式1和模式2



11.5 OIS 链接通后的陀螺仪自检 (UI) – 模式 3、4

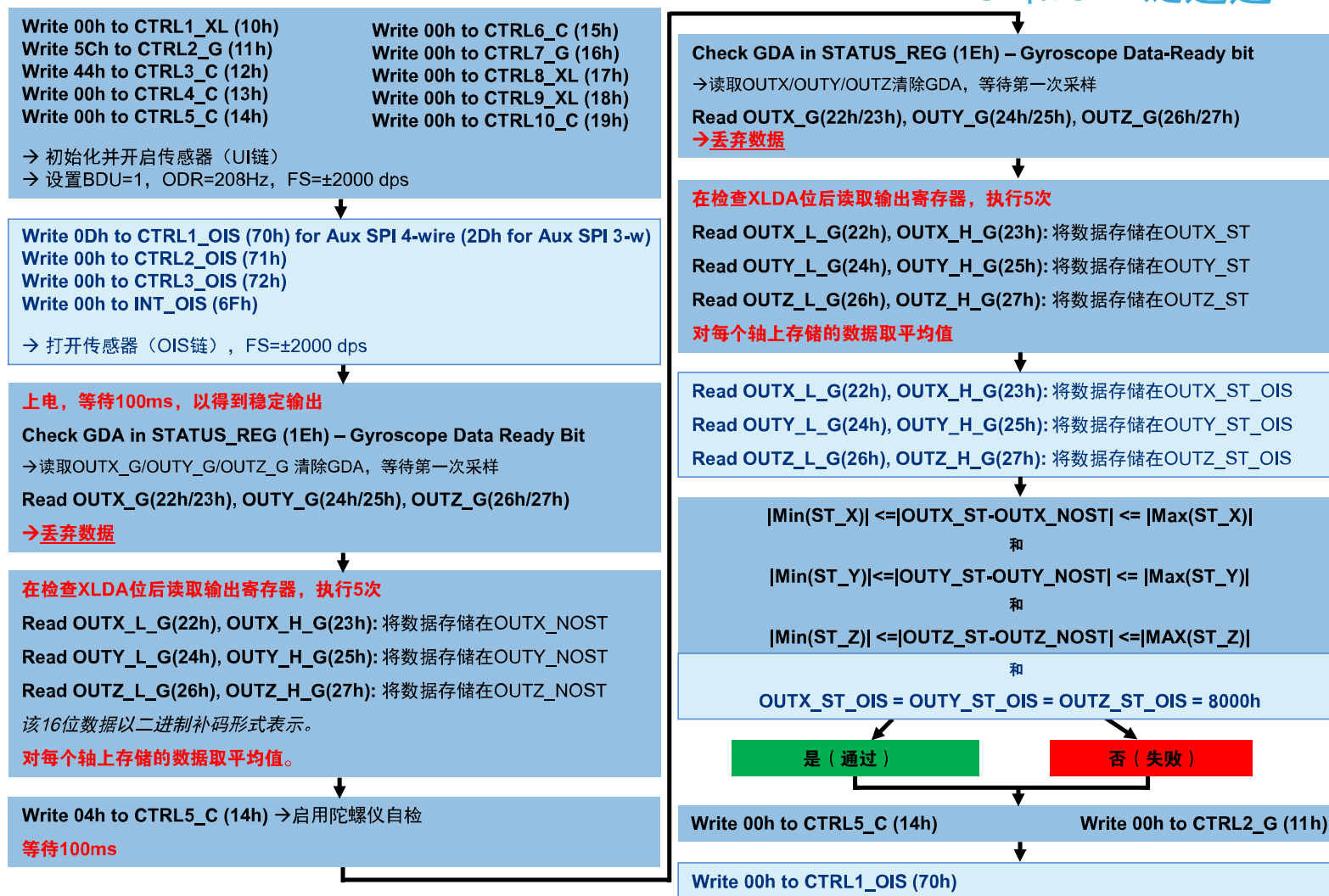
如果 UI 链和 OIS 链连通，则必须从主接口通过 CTRL5_C 寄存器的 ST[1:0]_G 位使能陀螺仪自检功能。不能同时从两个接口使能（禁止条件）。

当 OIS 链连通时，UI 链上的推荐陀螺仪自检流程如图 40. OIS 链连通时的陀螺仪自检流程 (UI) 中所示。

图 40. OIS 链连通时的陀螺仪自检流程 (UI)

注:

- 所有黑色读/写操作都必须通过主要I²C/SPI接口执行
- 所有蓝色读/写操作都必须通过辅助SPI接口执行





11.6 陀螺仪自检(OIS) – 模式 3、4

当 UI 链断开时，可通过辅助 SPI 接口，将 CTRL3_OIS 寄存器的 ST[1:0]_OIS 位置位，从而使能 OIS 链上的陀螺仪自检功能。当 ST[1:0]_OIS 位被设定为 00b 时，自检功能关闭；当 ST[1:0]_OIS 位被置为 01b（正符号自检）或 11b（负符号自检）时，该功能使能。

当 UI 链断开时，OIS 链上的完整陀螺仪自检流程如图 41. 陀螺仪自检步骤（OIS）中所示。仅当 UI 读出链断开（CTRL1_XL 寄存器中的 ODR_XL[1:0] = 0000b，CTRL2_G 寄存器中的 ODR_G[1:0] = 0000b）时，才能执行此流程。

图 41. 陀螺仪自检步骤 (OIS)

陀螺仪OIS自检模式3和模式4， OIS链连通且UI链断开

注：

- 本程序中的所有读/写操作都必须通过**辅助SPI接口执行**
- 仅当**UI读出链断开**（CTRL2_G中的ODR_G[3:0] = 0000b）时，才能执行此程序。

Write 0Dh to CTRL1_OIS (70h) for Aux SPI 4-wire (2Dh for Aux SPI 3-w)
Write 00h to CTRL2_OIS (71h)
Write 00h to CTRL3_OIS (72h)
Write 00h to INT_OIS (6Fh)

→ 初始化并启动传感器
→ FS=±2000 dps (ODR默认6.66kHz)

上电，等待100ms，以得到稳定输出

Check GDA in STATUS_SPIAux (1Eh) – Gyroscope Data-Ready bit
→ 读取OUTX_G/OUTY_G/OUTZ_G 清除GDA，等待第一次采样
Read OUTX_G(22h/23h), OUTY_G(24h/25h), OUTZ_G(26h/27h)
→ **丢弃数据**

检测XLDA位后读取数据，执行五次操作

Read OUTX_L_G(22h), OUTX_H_G(23h): 将数据存储到OUTX_NOST_OIS
Read OUTY_L_G(24h), OUTY_H_G(25h): 将数据存储到OUTY_NOST_OIS
Read OUTZ_L_G(26h), OUTZ_H_G(27h): 将数据存储到OUTZ_NOST_OIS
该16位数据以二进制补码形式表示。

计算每个轴的平均值

Write 03h to CTRL3_OIS (72h) → 启用陀螺仪自检
等待100 ms

Check GDA in STATUS_SPIAux (1Eh) – Gyroscope Data-Ready bit
→ 读取OUTX/OUTY/OUTZ清除GDA，等待第一次采样
Read OUTX_G(22h/23h), OUTY_G(24h/25h), OUTZ_G(26h/27h)
→ **丢弃数据**

检测XLDA位后读取数据，执行五次操作

Read OUTX_L_G(22h), OUTX_H_G(23h): 将数据存储到OUTX_ST_OIS
Read OUTY_L_G(24h), OUTY_H_G(25h): 将数据存储到OUTY_ST_OIS
Read OUTZ_L_G(26h), OUTZ_H_G(27h): 将数据存储到OUTZ_ST_OIS
该16位数据以二进制补码形式表示。
计算每个轴的平均值

$|\text{Min}(\text{ST}_X)| \leq |\text{OUTX_ST_OIS} - \text{OUTX_NOST_OIS}| \leq |\text{Max}(\text{ST}_X)|$
和
 $|\text{Min}(\text{ST}_Y)| \leq |\text{OUTY_ST_OIS} - \text{OUTY_NOST_OIS}| \leq |\text{Max}(\text{ST}_Y)|$
和
 $|\text{Min}(\text{ST}_Z)| \leq |\text{OUTZ_ST_OIS} - \text{OUTZ_NOST_OIS}| \leq |\text{Max}(\text{ST}_Z)|$

是 (通过)

否 (失败)

Write 00h to CTRL3_OIS (72h): 禁用自检
Write 00h to CTRL1_OIS (70h): 禁用传感器

版本历史

表 95. 文档版本历史

日期	版本	变更
2018 年 8 月 30 日	1	初始版本
2019 年 1 月 28 日	2	更新了表 1。引脚状态 更新了表 14。模式 1/2 下的陀螺仪总带宽选择 更新了 4.4 节 使用块数据 (BDU) 功能 更新了 5.7 节“启动状态” 更新了图 21。模式 2 下外部传感器连接 更新了 7.2.1 节 MASTER_CONFIG (14h) 更新了图 23。直连功能 更新了 9.2.8 节“FIFO_STATUS2” 更新了 9.10.5 节：“未压缩数据率” 增加了 9.10.7 节“FIFO 压缩示例”
2019 年 4 月 16 日	3	更新了 3 节“工作模式” 更新了 3.8 节“加速度计带宽” 更新了 5.5.3 节“单击和双击识别配置” 更新了 6.4 节“时间戳” 更新了表 64。OIS 链稳定时间 更新了 9.2.2 节“FIFO_CTRL2” 更新了 9.8 节“从 FIFO 恢复数据”
2019 年 7 月 10 日	4	更新了“表 1. 引脚状态”中的引脚 10 和 11 表 1. 引脚状态 更新了“6.1 计步功能：步伐检测 and 步数计算”中的软件例程 第 6.1 节 计步功能：步伐检测 and 步数计算

目录

1	引脚说明.....	2
2	寄存器.....	5
2.1	嵌入功能寄存器	8
2.2	嵌入高级功能页	10
2.3	传感器集合寄存器	12
3	工作模式.....	14
3.1	下电模式	16
3.2	高性能模式	16
3.3	正常模式	16
3.4	低功耗模式	16
3.5	加速度计超低功耗模式.....	16
3.6	陀螺仪睡眠模式	17
3.7	连接模式	17
3.8	加速度计带宽	17
3.8.1	加速度计斜率滤波器.....	20
3.9	加速度计开启/关断时间	22
3.10	陀螺仪带宽	24
3.11	陀螺仪开启/关闭时间	30
4	模式 1 - 读取输出数据	32
4.1	启动序列	32
4.2	使用状态寄存器	32
4.3	使用数据准备就绪信号.....	33
4.3.1	DRDY 屏蔽功能.....	35
4.4	使用块数据更新 (block data update, BDU) 功能	35
4.5	理解输出数据	35
4.5.1	输出数据示例	35
4.6	加速度计偏移寄存器.....	37
4.7	环行功能	37
4.7.1	FIFO 输出寄存器环行	37

4.7.2	传感器输出寄存器环行	37
4.8	DEN (数据使能)	38
4.8.1	边沿感应触发模式	39
4.8.2	电平感应触发模式	41
4.8.3	电平感应锁存模式	42
4.8.4	启用电平感应 FIFO	43
4.8.5	用于 DEN 冲压的 LSB 选择	43
5	中断生成	44
5.1	中断引脚配置	44
5.2	自由落体中断	47
5.3	唤醒中断	50
5.4	6D/4D 定向检测	52
5.4.1	6D 定向检测	52
5.4.2	4D 方向检测	55
5.5	单击和双击识别	55
5.5.1	单击	56
5.5.2	双击	58
5.5.3	单击和双击识别配置	60
5.5.4	单击示例	62
5.5.5	双击示例	62
5.6	活动/不活动和运动/静止识别	63
5.6.1	静止/运动检测	66
5.7	启动状态	67
6	嵌入功能	68
6.1	计步功能: 步伐侦测和步数计算	68
6.2	大幅运动检测	70
6.3	相对倾斜	71
6.4	时间戳	74
7	模式 2 - 传感器集合 (sensor hub) 模式	75
7.1	传感器集合 (sensor hub) 模式说明	75
7.2	传感器集合 (sensor hub) 模式寄存器	77

7.2.1	MASTER_CONFIG (14h)	77
7.2.2	STATUS_MASTER (22h)	78
7.2.3	SLV0_ADD (15h), SLV0_SUBADD (16h), SLAVE0_CONFIG (17h)	79
7.2.4	SLV1_ADD (18h), SLV1_SUBADD (19h), SLAVE1_CONFIG (1Ah)	80
7.2.5	SLV2_ADD (1Bh), SLV2_SUBADD (1Ch), SLAVE2_CONFIG (1Dh)	81
7.2.6	SLV3_ADD (1Eh), SLV3_SUBADD (1Fh), SLAVE3_CONFIG (20h)	82
7.2.7	DATAWRITE_SLV0 (0Eh)	82
7.2.8	SENSOR_HUB_x 寄存器	82
7.3	传感器集合 (sensor hub) 直连功能	85
7.3.1	直连功能使能	88
7.3.2	直连功能禁用	88
7.4	传感器集合 (sensor hub) 模式示例	88
8	模式 3 和模式 4 - 辅助 SPI 模式	91
8.1	辅助 SPI 模式说明	91
8.2	辅助 SPI 模式寄存器	98
8.2.1	INT_OIS (6Fh)	98
8.2.2	CTRL1_OIS (70h)	99
8.2.3	CTRL2_OIS (71h)	100
8.2.4	CTRL3_OIS (72h)	100
8.2.5	STATUS_SPIAux (1Eh)	101
8.3	OIS 链稳定时间	102
8.4	模式 3 - 通过辅助 SPI 读取陀螺仪数据	103
8.5	模式 4 - 通过辅助 SPI 读取陀螺仪和加速度计数据	103
9	先进先出 (FIFO) 缓冲区	104
9.1	FIFO 说明和批处理传感器	105
9.2	FIFO 寄存器	105
9.2.1	FIFO_CTRL1	105
9.2.2	FIFO_CTRL2	106
9.2.3	FIFO_CTRL3	106
9.2.4	FIFO_CTRL4	107
9.2.5	COUNTER_BDR_REG1	109
9.2.6	COUNTER_BDR_REG2	109

9.2.7	FIFO_STATUS1	109
9.2.8	FIFO_STATUS2	110
9.2.9	FIFO_DATA_OUT_TAG	110
9.2.10	FIFO_DATA_OUT	111
9.3	FIFO 批处理传感器	112
9.4	主要传感器	112
9.5	辅助传感器	115
9.6	虚拟传感器	117
9.6.1	外部传感器和 NACK 传感器	117
9.6.2	计步器传感器	117
9.7	FIFO 模式	118
9.7.1	Bypass 模式	118
9.7.2	FIFO 模式	119
9.7.3	Continue 模式	121
9.7.4	Continue-FIFO 模式	123
9.7.5	Bypass-Continue 模式	125
9.7.6	Bypass-FIFO 模式	127
9.8	从 FIFO 恢复数据	129
9.9	FIFO 水印阈值	130
9.10	FIFO 压缩	136
9.10.1	时间相关性	137
9.10.2	数据格式	138
9.10.3	在运行时间禁用 FIFO 压缩	139
9.10.4	使能了 FIFO 压缩的 CFG-Change 传感器	140
9.10.5	未压缩数据率	140
9.10.6	FIFO 压缩初始化	140
9.10.7	FIFO 压缩示例	140
9.11	时间戳相关性	142
10	温度传感器	143
10.1	温度数据计算示例	143
11	自检功能	144



11.1	加速度计自检 (UI) ◆C 模式 1、2	144
11.2	OIS 链接通后的加速度计自检 (UI) ◆C 模式 4	146
11.3	加速度计自检 (OIS) – 模式 4	148
11.4	陀螺仪自检 (UI) – 模式 1、2	150
11.5	OIS 链接通后的陀螺仪自检 (UI) – 模式 3、4	152
11.6	陀螺仪自检(OIS) – 模式 3、4	154
版本历史		156



表一览

表 1.	引脚状态	3
表 2.	寄存器	5
表 3.	嵌入功能寄存器	8
表 4.	嵌入高级功能寄存器 - page 0	10
表 5.	嵌入高级功能寄存器 - page 1	10
表 6.	传感器集合寄存器	12
表 7.	加速度计 ODR 和功耗模式选择	14
表 8.	陀螺仪 ODR 和功耗模式选择	15
表 9.	功耗 (典型值)	15
表 10.	模式 1/2/3 下的加速度计带宽选择	19
表 11.	加速度计开启/关闭时间 (LPF2 和 HP 禁用)	22
表 12.	要丢弃的加速度计样本	22
表 13.	陀螺仪数字 HP 滤波器截止频率选择	26
表 14.	模式 1/2 下的陀螺仪总带宽选择	26
表 15.	陀螺仪低功耗/正常模式下的带宽	28
表 16.	UI 链 - 模式 3/4 下的陀螺仪总带宽选择	30
表 17.	模式 1/2 下的陀螺仪开启/关闭时间 (HP 禁用)	30
表 18.	模式 1/2 下要丢弃的陀螺仪样本 (LPF1 禁用)	31
表 19.	模式 1/2 下的陀螺仪链稳定时间 (LPF1 使能)	31
表 20.	输出数据寄存器内容 vs. 加速度 ($FS_{XL} = \pm 2 g$)	35
表 21.	输出数据寄存器内容 vs. 角速度 ($FS_G = \pm 250 dps$)	36
表 22.	输出寄存器环行模式	37
表 23.	DEN 配置	38
表 24.	INT1_CTRL 寄存器	44
表 25.	MD1_CFG 寄存器	45
表 26.	INT2_CTRL 寄存器	45
表 27.	MD2_CFG 寄存器	45
表 28.	自由落体阈值 LSB 值	49
表 29.	D6D_SRC 寄存器	53
表 30.	4D/6D 功能阈值	53
表 31.	6D 定位下的 D6D_SRC 寄存器	55
表 32.	TAP_PRIORITY_[2:0]位的配置	60
表 33.	TAP_SRC 寄存器	62
表 34.	不活动事件配置	63
表 35.	EMB_FUNC_SRC 嵌入功能寄存器	68
表 36.	IS_STEP_DET 配置	69
表 37.	ODR _{coeff} 值	74
表 38.	MASTER_CONFIG 寄存器	77
表 39.	STATUS_MASTER / STATUS_MASTER_MAINPAGE 寄存器	78
表 40.	SLV0_ADD 寄存器	79
表 41.	SLV0_SUBADD 寄存器	79
表 42.	SLAVE0_CONFIG 寄存器	79
表 43.	SLV1_ADD 寄存器	80
表 44.	SLV1_SUBADD 寄存器	80
表 45.	SLAVE1_CONFIG 寄存器	80
表 46.	SLV2_ADD 寄存器	81
表 47.	SLV2_SUBADD 寄存器	81
表 48.	SLAVE2_CONFIG 寄存器	81
表 49.	SLV3_ADD 寄存器	82
表 50.	SLV3_SUBADD 寄存器	82
表 51.	SLAVE3_CONFIG 寄存器	82



表 52.	DATAWRITE_SLV0 寄存器	82
表 53.	加速度计 OIS 链满量程的选择	91
表 54.	模式 3/4 引脚说明	92
表 55.	INT_OIS 寄存器	98
表 56.	CTRL1_OIS 寄存器	99
表 57.	DEN 模式选择	99
表 58.	CTRL2_OIS 寄存器	100
表 59.	陀螺仪 OIS 链 HP 滤波器截止频率选择	100
表 60.	LPF1 滤波器配置	100
表 61.	CTRL3_OIS 寄存器	100
表 62.	LPF_OIS 滤波器配置	101
表 63.	STATUS_SPIAux 寄存器	101
表 64.	OIS 链稳定时间	102
表 65.	FIFO_CTRL1 寄存器	106
表 66.	FIFO_CTRL2 寄存器	106
表 67.	强制未压缩数据写入配置	106
表 68.	FIFO_CTRL3 寄存器	106
表 69.	加速度计批处理数据率	106
表 70.	陀螺仪批处理数据率	107
表 71.	时间戳批处理数据率	107
表 72.	温度传感器批处理数据率	108
表 73.	FIFO_CTRL4 寄存器	108
表 74.	COUNTER_BDR_REG1 寄存器	109
表 75.	COUNTER_BDR_REG2 寄存器	109
表 76.	FIFO_STATUS1 寄存器	109
表 77.	FIFO_STATUS2 寄存器	110
表 78.	FIFO_DATA_OUT_TAG 寄存器	110
表 79.	TAG_SENSOR 字段和相关传感器	111
表 80.	FIFO 中的主要传感器输出数据格式	114
表 81.	FIFO 中的温度输出数据格式	115
表 82.	FIFO 中的时间戳输出数据格式	115
表 83.	FIFO 中的 CFG-change 输出数据格式	116
表 84.	BDR_SHUB	117
表 85.	FIFO 中的 Nack 传感器输出数据格式	117
表 86.	FIFO 中的计步器输出数据格式	118
表 87.	FIFO 压缩标签和相关数据	137
表 88.	FIFO 中的 2xC 压缩数据输出数据格式	138
表 89.	FIFO 中的 3xC 压缩数据输出数据格式	138
表 90.	禁用运行时间压缩的示例	139
表 91.	使能了 FIFO 压缩的设备配置修改示例	140
表 92.	UNCOPTR_RATE 配置	140
表 93.	FIFO 压缩示例	141
表 94.	输出数据寄存器内容 vs. 温度	143
表 95.	文档版本历史	156



图一览

图 1.	引脚连接	2
图 2.	加速度计滤波链 (UI 路径)	18
图 3.	加速度计斜率滤波器	21
图 4.	陀螺仪数字链 - 模式 1 和模式 2	25
图 5.	陀螺仪数字链 - 模式 3 和模式 4	29
图 6.	数据准备就绪信号	34
图 7.	边沿感应触发模式, DEN 低电平有效	39
图 8.	电平感应触发模式, DEN 低电平有效	41
图 9.	电平感应触发模式, DEN 低电平有效, DEN_DRDY 在 INT1 上	41
图 10.	电平感应锁存模式, DEN 低电平有效	42
图 11.	电平感应锁存模式, DEN 低电平有效, DEN_DRDY 在 INT1 上	42
图 12.	电平感应 FIFO 使能模式, DEN 低电平有效	43
图 13.	自由落体中断	48
图 14.	唤醒中断 (利用斜率滤波器)	51
图 15.	6D 识别方向	54
图 16.	单击事件识别	56
图 17.	双击事件识别 (LIR 位 = 0)	59
图 18.	单击和双击识别 (LIR 位 = 0)	61
图 19.	活动/不活动识别 (利用斜率滤波器)	65
图 20.	倾斜度检测	72
图 21.	模式 2 下外部传感器连接	76
图 22.	SENSOR_HUB_X 配置示例	83
图 23.	直连功能	86
图 24.	模式 3/4 下的外部控制器连接 (SPI 3 线)	93
图 25.	陀螺仪滤波链	95
图 26.	加速度计滤波链	97
图 27.	主要传感器和时序的定义	113
图 28.	FIFO 模式 (STOP_ON_WTM = 0)	120
图 29.	Continue 模式	122
图 30.	Continue-FIFO 模式	124
图 31.	Bypass-Continue 模式	126
图 32.	Bypass-FIFO 模式	128
图 33.	FIFO 阈值 (STOP_ON_WTM = 0)	131
图 34.	FIFO 模式下的 FIFO 阈值 (STOP_ON_WTM = 1)	133
图 35.	Continue 模式下的 FIFO 阈值 (STOP_ON_WTM = 1)	135
图 36.	加速度计自检步骤 (UI)	145
图 37.	OIS 链连通时的加速度计自检流程 (UI)	147
图 38.	加速度计自检步骤 (OIS)	149
图 39.	陀螺仪自检步骤 (UI)	151
图 40.	OIS 链连通时的陀螺仪自检流程 (UI)	153
图 41.	陀螺仪自检步骤 (OIS)	155



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