

LSM6DSV16X: 具有嵌入式机器学习内核和 Qvar 检测功能的始终开启 (Always-On) 3 轴加速度计和 3 轴陀螺仪

简介

本文档提供 ST LSM6DSV16X iNEMO 惯性模块相关的使用信息和应用提示。

LSM6DSV16X 是系统级封装的 3 轴数字加速度计和 3 轴数字陀螺仪，具有数字 I²C、SPI 和 MIPI I3C[®] 串行接口标准输出，组合工作在高性能模式下电流消耗只要 0.65 mA。由于陀螺仪和加速度计均具有超低噪声性能，始终具有低功耗特性，并结合了高传感精度，因此能够为客户提供最佳运动体验。此外，加速度计具有智能的休眠到唤醒（活动）和返回休眠（不活动）功能，具备先进的节能能力。

该器件具有动态的用户可选择的满量程加速度范围： $\pm 2/\pm 4/\pm 8/\pm 16$ g，且角速率范围为 $\pm 125/\pm 250/\pm 500/\pm 1000/\pm 2000/\pm 4000$ dps。它能够使能最多三个不同的内核来进行 UI、EIS 和 OIS 数据处理。

经过配置，LSM6DSV16X 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，例如传感器集线器 (sensor hub)、辅助 SPI 和模拟集线器 (analog hub)。

LSM6DSV16X 可兼容主要操作系统的要求，提供真实、虚拟以及批处理模式传感器。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 8 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。

LSM6DSV16X 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。

LSM6DSV16X 嵌入了模拟集线器 (analog hub) 传感功能，能够连接模拟输入并将其转换为数字信号进行嵌入式处理。此外，可以使用嵌入式 Qvar（电荷变化检测）通道进行人体接近和运动检测、触摸检测以及用户界面 (UI) 应用。

LSM6DSV16X 集成有一个 4.5 KB 的智能先进先出 (FIFO) 缓冲器，支持对有效数据（即，外部传感器、计步器、时间戳和温度、MLC 导出的滤波器和特征）进行动态批处理。

LSM6DSV16X 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 到 +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、可穿戴设备以及任何其他需要减小封装尺寸和重量的应用。

1 引脚说明

图 1. 引脚连接

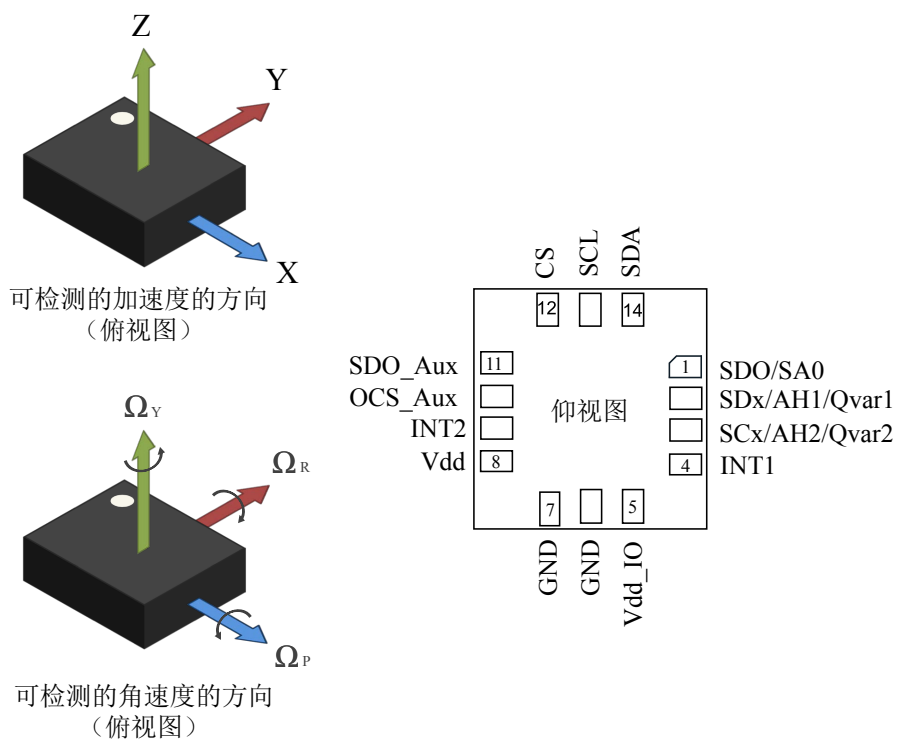


表 1. 内部引脚状态

引脚编号	名称	模式 1 功能	模式 2 功能	模式 3 功能	引脚状态模式 1	引脚状态模式 2	引脚状态模式 3 ⁽¹⁾
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: 输入无上拉	默认值: 输入无上拉	默认值: 输入无上拉
	SA0	I ² C 设备地址的最低有效位 (SA0) 静态地址 (SA0) 的 MIPI I3C [®] 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 [®] 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 [®] 最低有效位	如果寄存器 PIN_CTRL (02h) 中的位 SDO_PU_EN = 1, 则上拉使能。	如果寄存器 PIN_CTRL (02h) 中的位 SDO_PU_EN = 1, 则上拉使能。	如果寄存器 PIN_CTRL (02h) 中的位 SDO_PU_EN = 1, 则上拉使能。
2	SDx/AH1/Qvar1	如果模拟集线器 (analog hub) 和/或 Qvar 禁用, 则连接到 Vdd_IO 或 GND。 如果 Qvar 功能使能, 则连接到模拟输入或 Qvar 电极 1。 ⁽²⁾	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: 输入无上拉 如果寄存器 IF_CFG (03h) 中的位 SHUB_PU_EN = 1, 则上拉使能。	默认值: 输入无上拉 如果寄存器 IF_CFG (03h) 中的位 SHUB_PU_EN = 1, 则上拉使能。	默认值: 输入无上拉 如果寄存器 IF_CFG (03h) 中的位 SHUB_PU_EN = 1, 则上拉使能。
3	SCx/AH2/Qvar2	如果模拟集线器 (analog hub) 和/或 Qvar 禁用, 则连接到 Vdd_IO 或 GND。 如果 Qvar 功能使能, 则连接到模拟输入或 Qvar 电极 2。 ⁽²⁾	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: 输入无上拉 如果寄存器 IF_CFG (03h) 中的位 SHUB_PU_EN = 1, 则上拉使能。	默认值: 输入无上拉 如果寄存器 IF_CFG (03h) 中的位 SHUB_PU_EN = 1, 则上拉使能。	默认值: 输入无上拉 如果寄存器 IF_CFG (03h) 中的位 SHUB_PU_EN = 1, 则上拉使能。
4	INT1	可编程中断 1	可编程中断 1	可编程中断 1	默认值: 输出强制接地	默认值: 输出强制接地	默认值: 输出强制接地
5	Vdd_IO	I/O 引脚的供电	I/O 引脚的供电	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 (Data enable, 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 Vdd_IO 或保持断开	连接到 Vdd_IO 或保持断开	辅助 SPI 3/4 线接口使能	默认值: 输入带上拉 如果寄存器 PIN_CTRL (02h) 中的位 OIS_PU_DIS = 1, 则上拉禁用。	默认值: 输入带上拉 如果寄存器 PIN_CTRL (02h) 中的位 OIS_PU_DIS = 1, 则上拉禁用。	默认值: 输入无上拉 (与寄存器 PIN_CTRL (02h) 中的位 OIS_PU_DIS 的值无关)
11	SDO_Aux	连接到 Vdd_IO 或保持断开	连接到 Vdd_IO 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: 输入带上拉 如果寄存器 PIN_CTRL (02h) 中的位 OIS_PU_DIS = 1, 则上拉禁用。	默认值: 输入带上拉 如果寄存器 PIN_CTRL (02h) 中的位 OIS_PU_DIS = 1, 则上拉禁用。	默认值: 输入无上拉 如果寄存器 70h 中的位 SIM_OIS = 1 (Aux_SPI 3 线) 且寄存器 PIN_CTRL (02h) 中的位 OIS_PU_DIS = 0, 则上拉使能。
12	CS	I ² C/SPI 模式选择 (1: SPI 空闲模式/ I ² C 通信使能; 0: SPI 通信模式/ I ² C 禁用)	I ² C/SPI 模式选择 (1: SPI 空闲模式/ I ² C 通信使能; 0: SPI 通信模式/ I ² C 禁用)	I ² C/SPI 模式选择 (1: SPI 空闲模式/ I ² C 通信使能; 0: SPI 通信模式/ I ² C 禁用)	默认值: 输入带上拉 如果寄存器 IF_CFG (03h) 中的位 I2C_I3C_disable = 1, 则上拉禁用。	默认值: 输入带上拉 如果寄存器 IF_CFG (03h) 中的位 I2C_I3C_disable = 1, 则上拉禁用。	默认值: 输入带上拉 如果寄存器 IF_CFG (03h) 中的位 I2C_I3C_disable = 1, 则上拉禁用。
13	SCL	I ² C/MIPI I3C [®] 串行时钟 (SCL)/SPI 串口时钟 (SPC)	I ² C/MIPI I3C [®] 串行时钟 (SCL)/SPI 串口时钟 (SPC)	I ² C/MIPI I3C [®] 串行时钟 (SCL)/SPI 串口时钟 (SPC)	默认值: 输入无上拉	默认值: 输入无上拉	默认值: 输入无上拉

引脚编号	名称	模式 1 功能	模式 2 功能	模式 3 功能	引脚状态模式 1	引脚状态模式 2	引脚状态模式 3 ⁽¹⁾
14	SDA	I ² C/MIPI I3C [®] 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 [®] 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 [®] 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：输入无上拉 如果寄存器 IF_CFG (03h) 中的位 SDA_PU_EN = 1，则上拉使能。	默认值：输入无上拉 如果寄存器 IF_CFG (03h) 中的位 SDA_PU_EN = 1，则上拉使能。	默认值：输入无上拉 如果寄存器 IF_CFG (03h) 中的位 SDA_PU_EN = 1，则上拉使能。

- 1. 当 UI_CTRL1_OIS (70h)/SPI2_CTRL1_OIS (70h) 寄存器中的 OIS_XL_EN 位或 OIS_G_EN 位置为 1 时，模式 3 使能。
- 2. 通过将 CTRL7 (16h) 中的 AH_QVAR_EN 位置为 1 来使能模拟集线器 (analog hub) 和 Qvar 功能。

内部上拉值范围从 30 kΩ 到 50 kΩ，取决于 Vdd_IO。



下表给出的所有寄存器都只能从主 SPI/I²C/MIPI I3C[®] 接口访问。

表 2. 寄存器

寄存器名称	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
FUNC_CFG_ACCESS	01h	EMB_FUNC_REG_ACCESS	SHUB_REG_ACCESS	0	0	FSM_WR_CTRL_EN	SW_POR	SPI2_RESET	OIS_CTRL_FROM_UI
PIN_CTRL	02h	OIS_PU_DIS	SDO_PU_EN	IBHR_POR_EN	0	0	0	1	1
IF_CFG	03h	SDA_PU_EN	SHUB_PU_EN	ASF_CTRL	H_LACTIVE	PP_OD	SIM	0	I2C_I3C_DISABLE
S4S_TPH_L	04h	TPH_H_SEL	TPH_L_6	TPH_L_5	TPH_L_4	TPH_L_3	TPH_L_2	TPH_L_1	TPH_L_0
S4S_TPH_H	05h	TPH_H_7	TPH_H_6	TPH_H_5	TPH_H_4	TPH_H_3	TPH_H_2	TPH_H_1	TPH_H_0
S4S_RR	06h	0	0	0	0	0	0	RR_1	RR_0
FIFO_CTRL1	07h	WTM_7	WTM_6	WTM_5	WTM_4	WTM_3	WTM_2	WTM_1	WTM_0
FIFO_CTRL2	08h	STOP_ON_WTM	FIFO_COMPRT_EN	0	ODR_CHG_EN	0	UNCOMPR_RATE_1	UNCOMPR_RATE_0	XL_DualC_BATCH_FROM_FSM
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	G_EIS_FIFO_EN	FIFO_MODE_2	FIFO_MODE_1	FIFO_MODE_0
COUNTER_BDR_REG1	0Bh	0	TRIG_COUNTER_BDR_1	TRIG_COUNTER_BDR_0	0	0	0	CNT_BDR_TH_9	CNT_BDR_TH_8
COUNTER_BDR_REG2	0Ch	CNT_BDR_TH_7	CNT_BDR_TH_6	CNT_BDR_TH_5	CNT_BDR_TH_4	CNT_BDR_TH_3	CNT_BDR_TH_2	CNT_BDR_TH_1	CNT_BDR_TH_0
INT1_CTRL	0Dh	0	INT1_CNT_BDR	INT1_FIFO_FULL	INT1_FIFO_OVR	INT1_FIFO_TH	0	INT1_DRDY_G	INT1_DRDY_XL
INT2_CTRL	0Eh	INT2_EMB_FUNC_ENDOP	INT2_CNT_BDR	INT2_FIFO_FULL	INT2_FIFO_OVR	INT2_FIFO_TH	INT2_DRDY_G_EIS	INT2_DRDY_G	INT2_DRDY_XL
WHO_AM_I	0Fh	0	1	1	1	0	0	0	0
CTRL1	10h	0	OP_MODE_XL_2	OP_MODE_XL_1	OP_MODE_XL_0	ODR_XL_3	ODR_XL_2	ODR_XL_1	ODR_XL_0
CTRL2	11h	0	OP_MODE_G_2	OP_MODE_G_1	OP_MODE_G_0	ODR_G_3	ODR_G_2	ODR_G_1	ODR_G_0
CTRL3	12h	BOOT	BDU	0	0	0	IF_INC	0	SW_RESET
CTRL4	13h	0	0	0	INT2_on_INT1	DRDY_MASK	INT2_DRDY_TEMP	DRDY_PULSED	INT2_IN_LH
CTRL5	14h	0	0	0	0	0	BUS_ACT_SEL_1	BUS_ACT_SEL_0	INT_EN_I3C
CTRL6	15h	0	LPF1_G_BW_2	LPF1_G_BW_1	LPF1_G_BW_0	FS_G_3	FS_G_2	FS_G_1	FS_G_0
CTRL7	16h	AH_QVAR_EN	INT2_DRDY_AH_QVAR	AH_QVAR_C_ZIN_1	AH_QVAR_C_ZIN_0	0	0	0	LPF1_G_EN
CTRL8	17h	HP_LPF2_XL_BW_2	HP_LPF2_XL_BW_1	HP_LPF2_XL_BW_0	0	XL_DualC_EN	0	FS_XL_1	FS_XL_0
CTRL9	18h	0	HP_REF_MODE_XL	XL_FASTSETTL_MODE	HP_SLOPE_XL_EN	LPF2_XL_EN	0	USR_OFF_W	USR_OFF_ON_OUT





寄存器名称	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
CTRL10	19h	0	EMB_FUNC_DEBUG	0	0	ST_G_1	ST_G_0	ST_XL_1	ST_XL_0
CTRL_STATUS	1Ah	0	0	0	0	0	FSM_WR_CTRL_STATUS	-	0
FIFO_STATUS1	1Bh	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	1Ch	FIFO_WTM_IA	FIFO_OVR_IA	FIFO_FULL_IA	COUNTER_BDR_IA	FIFO_OVR_LATCHED	0	0	DIFF_FIFO_8
ALL_INT_SRC	1Dh	EMB_FUNC_IA	SHUB_IA	SLEEP_CHANGE_IA	D6D_IA	0	TAP_IA	WU_IA	FF_IA
STATUS_REG	1Eh	TIMESTAMP_ENDCOUNT	0	OIS_DRDY	GDA_EIS	AH_QVARDA	TDA	GDA	XLDA
OUT_TEMP_L	20h	Temp7	Temp6	Temp5	Temp4	Temp3	Temp2	Temp1	Temp0
OUT_TEMP_H	21h	Temp15	Temp14	Temp13	Temp12	Temp11	Temp10	Temp9	Temp8
OUTX_L_G	22h	D7	D6	D5	D4	D3	D2	D1	D0
OUTX_H_G	23h	D15	D14	D13	D12	D11	D10	D9	D8
OUTY_L_G	24h	D7	D6	D5	D4	D3	D2	D1	D0
OUTY_H_G	25h	D15	D14	D13	D12	D11	D10	D9	D8
OUTZ_L_G	26h	D7	D6	D5	D4	D3	D2	D1	D0
OUTZ_H_G	27h	D15	D14	D13	D12	D11	D10	D9	D8
OUTX_L_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
UI_OUTX_L_G_OIS_EIS	2Eh	D7	D6	D5	D4	D3	D2	D1	D0
UI_OUTX_H_G_OIS_EIS	2Fh	D15	D14	D13	D12	D11	D10	D9	D8
UI_OUTY_L_G_OIS_EIS	30h	D7	D6	D5	D4	D3	D2	D1	D0
UI_OUTY_H_G_OIS_EIS	31h	D15	D14	D13	D12	D11	D10	D9	D8
UI_OUTZ_L_G_OIS_EIS	32h	D7	D6	D5	D4	D3	D2	D1	D0
UI_OUTZ_H_G_OIS_EIS	33h	D15	D14	D13	D12	D11	D10	D9	D8
UI_OUTX_L_A_OIS_DualC	34h	D7	D6	D5	D4	D3	D2	D1	D0
UI_OUTX_H_A_OIS_DualC	35h	D15	D14	D13	D12	D11	D10	D9	D8
UI_OUTY_L_A_OIS_DualC	36h	D7	D6	D5	D4	D3	D2	D1	D0
UI_OUTY_H_A_OIS_DualC	37h	D15	D14	D13	D12	D11	D10	D9	D8
UI_OUTZ_L_A_OIS_DualC	38h	D7	D6	D5	D4	D3	D2	D1	D0
UI_OUTZ_H_A_OIS_DualC	39h	D15	D14	D13	D12	D11	D10	D9	D8



寄存器名称	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
AH_QVAR_OUT_L	3Ah	AH_Qvar_7	AH_Qvar_6	AH_Qvar_5	AH_Qvar_4	AH_Qvar_3	AH_Qvar_2	AH_Qvar_1	AH_Qvar_0
AH_QVAR_OUT_H	3Bh	AH_Qvar_15	AH_Qvar_14	AH_Qvar_13	AH_Qvar_12	AH_Qvar_11	AH_Qvar_10	AH_Qvar_9	AH_Qvar_8
TIMESTAMP0	40h	D7	D6	D5	D4	D3	D2	D1	D0
TIMESTAMP1	41h	D15	D14	D13	D12	D11	D10	D9	D8
TIMESTAMP2	42h	D23	D22	D21	D20	D19	D18	D17	D16
TIMESTAMP3	43h	D31	D30	D29	D28	D27	D26	D25	D24
UI_STATUS_REG_OIS	44h	0	0	0	0	0	GYRO_SETTLING	GDA_OIS	XLDA_OIS
WAKE_UP_SRC	45h	0	SLEEP_CHANGE_IA	FF_IA	SLEEP_STATE	WU_IA	X_WU	Y_WU	Z_WU
TAP_SRC	46h	0	TAP_IA	SINGLE_TAP	DOUBLE_TAP	TAP_SIGN	X_TAP	Y_TAP	Z_TAP
D6D_SRC	47h	0	D6D_IA	ZH	ZL	YH	YL	XH	XL
STATUS_MASTER_MAINPAGE	48h	WR_ONCE_DONE	SLAVE3_NACK	SLAVE2_NACK	SLAVE1_NACK	SLAVE0_NACK	0	0	SENS_HUB_ENDOP
EMB_FUNC_STATUS_MAINPAGE	49h	IS_FSM_LC	0	IS_SIGMOT	IS_TILT	IS_STEP_DET	0	0	0
FSM_STATUS_MAINPAGE	4Ah	IS_FSM8	IS_FSM7	IS_FSM6	IS_FSM5	IS_FSM4	IS_FSM3	IS_FSM2	IS_FSM1
MLC_STATUS_MAINPAGE	4Bh	0	0	0	0	IS_MLC4	IS_MLC3	IS_MLC2	IS_MLC1
INTERNAL_FREQ_FINE	4Fh	FREQ_FINE_7	FREQ_FINE_6	FREQ_FINE_5	FREQ_FINE_4	FREQ_FINE_3	FREQ_FINE_2	FREQ_FINE_1	FREQ_FINE_0
FUNCTIONS_ENABLE	50h	INTERRUPTS_ENABLE	TIMESTAMP_EN	0	0	DIS_RST_LIR_ALL_INT	0	INACT_EN_1	INACT_EN_0
DEN	51h	0	LVL1_EN	LVL2_EN	DEN_XL_EN	DEN_X	DEN_Y	DEN_Z	DEN_XL_G
INACTIVITY_DUR	54h	SLEEP_STATUS_ON_INT	WU_INACT_THS_W_2	WU_INACT_THS_W_1	WU_INACT_THS_W_0	XL_INACT_ODR_1	XL_INACT_ODR_0	INACT_DUR_1	INACT_DUR_0
INACTIVITY_THS	55h	0	0	INACT_THS_5	INACT_THS_4	INACT_THS_3	INACT_THS_2	INACT_THS_1	INACT_THS_0
TAP_CFG0	56h	0	LOW_PASS_ON_6D	HW_FUNC_MASK_XL_SETTL	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	0	0	0	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_THS_1	SIXD_THS_0	TAP_THS_Z_4	TAP_THS_Z_3	TAP_THS_Z_2	TAP_THS_Z_1	TAP_THS_Z_0
TAP_DUR	5Ah	DUR_3	DUR_2	DUR_1	DUR_0	QUIET_1	QUIET_0	SHOCK_1	SHOCK_0
WAKE_UP_THS	5Bh	SINGLE_DOUBLE_TAP	USR_OFF_ON_WU	WK_THS_5	WK_THS_4	WK_THS_3	WK_THS_2	WK_THS_1	WK_THS_0
WAKE_UP_DUR	5Ch	FF_DUR_5	WAKE_DUR_1	WAKE_DUR_0	0	SLEEP_DUR_3	SLEEP_DUR_2	SLEEP_DUR_1	SLEEP_DUR_0
FREE_FALL	5Dh	FF_DUR_4	FF_DUR_3	FF_DUR_2	FF_DUR_1	FF_DUR_0	FF_THS_2	FF_THS_1	FF_THS_0
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
S4S_ST_CMD_CODE	60h	ST_CMD_CODE_7	ST_CMD_CODE_6	ST_CMD_CODE_5	ST_CMD_CODE_4	ST_CMD_CODE_3	ST_CMD_CODE_2	ST_CMD_CODE_1	ST_CMD_CODE_0



寄存器名称	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
S4S_DT_REG	61h	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
EMB_FUNC_CFG	63h	0	0	EMB_FUNC_IRQ_MASK_G_SETTL	EMB_FUNC_IRQ_MASK_XL_SETTL	EMB_FUNC_DISABLE	0	0	0
UI_HANDSHAKE_CTRL	64h	0	0	0	0	0	0	UI_SHARED_ACK	UI_SHARED_REQ
UI_SPI2_SHARED_0	65h	D7	D6	D5	D4	D3	D2	D1	D0
UI_SPI2_SHARED_1	66h	D7	D6	D5	D4	D3	D2	D1	D0
UI_SPI2_SHARED_2	67h	D7	D6	D5	D4	D3	D2	D1	D0
UI_SPI2_SHARED_3	68h	D7	D6	D5	D4	D3	D2	D1	D0
UI_SPI2_SHARED_4	69h	D7	D6	D5	D4	D3	D2	D1	D0
UI_SPI2_SHARED_5	6Ah	D7	D6	D5	D4	D3	D2	D1	D0
CTRL_EIS	6Bh	ODR_G_EIS_1	ODR_G_EIS_0	0	LPF_G_EIS_BW	G_EIS_ON_G_OIS_OUT_REG	FS_G_EIS_2	FS_G_EIS_1	FS_G_EIS_0
UI_INT_OIS	6Fh	INT2_DRDY_OIS	DRDY_MASK_OIS	0	ST_OIS_CLAMPDIS	0	0	0	0
UI_CTRL1_OIS	70h	0	0	SIM_OIS	0	0	OIS_XL_EN	OIS_G_EN	SPI2_READ_EN
UI_CTRL2_OIS	71h	0	0	0	LPF1_G_OIS_BW_1	LPF1_G_OIS_BW_0	FS_G_OIS_2	FS_G_OIS_1	FS_G_OIS_0
UI_CTRL3_OIS	72h	0	0	LPF_XL_OIS_BW_2	LPF_XL_OIS_BW_1	LPF_XL_OIS_BW_0	0	FS_XL_OIS_1	FS_XL_OIS_0
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	-
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

下表给出的所有寄存器都只能从辅助 SPI 接口访问。

表 3. SPI 寄存器

寄存器名称	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
SPI2_WHO_AM_I	0Fh	0	1	1	1	0	0	0	0
SPI2_STATUS_REG_OIS	1Eh	0	0	0	0	0	GYRO_SETTLING	GDA	XLDA
SPI2_OUT_TEMP_L	20h	Temp7	Temp6	Temp5	Temp4	Temp3	Temp2	Temp1	Temp0
SPI2_OUT_TEMP_H	21h	Temp15	Temp14	Temp13	Temp12	Temp11	Temp10	Temp9	Temp8
SPI2_OUTX_L_G_OIS	22h	D7	D6	D5	D4	D3	D2	D1	D0
SPI2_OUTX_H_G_OIS	23h	D15	D14	D13	D12	D11	D10	D9	D8
SPI2_OUTY_L_G_OIS	24h	D7	D6	D5	D4	D3	D2	D1	D0
SPI2_OUTY_H_G_OIS	25h	D15	D14	D13	D12	D11	D10	D9	D8
SPI2_OUTZ_L_G_OIS	26h	D7	D6	D5	D4	D3	D2	D1	D0
SPI2_OUTZ_H_G_OIS	27h	D15	D14	D13	D12	D11	D10	D9	D8
SPI2_OUTX_L_A_OIS	28h	D7	D6	D5	D4	D3	D2	D1	D0
SPI2_OUTX_H_A_OIS	29h	D15	D14	D13	D12	D11	D10	D9	D8
SPI2_OUTY_L_A_OIS	2Ah	D7	D6	D5	D4	D3	D2	D1	D0
SPI2_OUTY_H_A_OIS	2Bh	D15	D14	D13	D12	D11	D10	D9	D8
SPI2_OUTZ_L_A_OIS	2Ch	D7	D6	D5	D4	D3	D2	D1	D0
SPI2_OUTZ_H_A_OIS	2Dh	D15	D14	D13	D12	D11	D10	D9	D8
SPI2_HANDSHAKE_CTRL	6Eh	0	0	0	0	0	0	SPI2_SHARED_REQ	SPI2_SHARED_ACK
SPI2_INT_OIS	6Fh	INT2_DRDY_OIS	DRDY_MASK_OIS	0	ST_OIS_CLAMPDIS	ST_G_OIS_1	ST_G_OIS_0	ST_XL_OIS_1	ST_XL_OIS_0
SPI2_CTRL1_OIS	70h	0	0	SIM_OIS	0	0	OIS_XL_EN	OIS_G_EN	SPI2_READ_EN
SPI2_CTRL2_OIS	71h	0	0	0	LPF1_G_OIS_BW_1	LPF1_G_OIS_BW_0	FS_G_OIS_2	FS_G_OIS_1	FS_G_OIS_0
SPI2_CTRL3_OIS	72h	0	0	LPF_XL_OIS_BW_2	LPF_XL_OIS_BW_1	LPF_XL_OIS_BW_0	0	FS_XL_OIS_1	FS_XL_OIS_0

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 EMB_FUNC_REG_ACCESS 位置为 1 时，嵌入功能寄存器可访问。

表 4. 嵌入功能寄存器

寄存器名称	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	MLC_BEFORE_FSM_EN	0	SIGN_MOTION_EN	TILT_EN	PEDO_EN	0	SFLP_GAME_EN	0
EMB_FUNC_EN_B	05h	0	0	0	MLC_EN	FIFO_COMPR_EN	0	0	FSM_EN
EMB_FUNC_EXEC_STAT_US	07h	0	0	0	0	0	0	EMB_FUNC_EXEC_OVR	EMB_FUNC_ENDOP
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	0Bh	INT1_FSM8	INT1_FSM7	INT1_FSM6	INT1_FSM5	INT1_FSM4	INT1_FSM3	INT1_FSM2	INT1_FSM1
MLC_INT1	0Dh	0	0	0	0	INT1_MLC4	INT1_MLC3	INT1_MLC2	INT1_MLC1
EMB_FUNC_INT2	0Eh	INT2_FSM_LC	0	INT2_SIG_MOT	INT2_TILT	INT2_STEP_DETECTOR	0	0	0
FSM_INT2	0Fh	INT2_FSM8	INT2_FSM7	INT2_FSM6	INT2_FSM5	INT2_FSM4	INT2_FSM3	INT2_FSM2	INT2_FSM1
MLC_INT2	11h	0	0	0	0	INT2_MLC4	INT2_MLC3	INT2_MLC2	INT2_MLC1
EMB_FUNC_STATUS	12h	IS_FSM_LC	0	IS_SIGMOT	IS_TILT	IS_STEP_DET	0	0	0
FSM_STATUS	13h	IS_FSM8	IS_FSM7	IS_FSM6	IS_FSM5	IS_FSM4	IS_FSM3	IS_FSM2	IS_FSM1
MLC_STATUS	15h	0	0	0	0	IS_MLC4	IS_MLC3	IS_MLC2	IS_MLC1
PAGE_RW	17h	EMB_FUNC_LIR	PAGE_WRITE	PAGE_READ	0	0	0	0	0
EMB_FUNC_FIFO_EN_A	44h	MLC_FIFO_EN	STEP_COUNTER_FIFO_EN	SFLP_GBIAS_FIFO_EN	SFLP_GRAVITY_FIFO_EN	0	0	SFLP_GAME_FIFO_EN	0
EMB_FUNC_FIFO_EN_B	45h	0	0	0	0	0	0	MLC_FILTER_FEATURE_FIFO_EN	0
FSM_ENABLE	46h	FSM8_EN	FSM7_EN	FSM6_EN	FSM5_EN	FSM4_EN	FSM3_EN	FSM2_EN	FSM1_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
INT_ACK_MASK	4Bh	IACK_MASK7	IACK_MASK6	IACK_MASK5	IACK_MASK4	IACK_MASK3	IACK_MASK2	IACK_MASK1	IACK_MASK0
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



寄存器名称	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
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
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
SFLP_ODR	5Eh	0	1	SFLP_GAME_ODR_2	SFLP_GAME_ODR_1	SFLP_GAME_ODR_0	0	1	1
FSM_ODR	5Fh	0	1	FSM_ODR_2	FSM_ODR_1	FSM_ODR_0	0	1	1
MLC_ODR	60h	0	MLC_ODR_2	MLC_ODR_1	MLC_ODR_0	0	1	0	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_COUNTEER_BIT_SET	0	0
EMB_FUNC_INIT_A	66h	MLC_BEFORE_FSM_INIT	0	SIG_MOT_INIT	TILT_INIT	STEP_DET_INIT	0	SFLP_GAME_INIT	0
EMB_FUNC_INIT_B	67h	0	0	0	MLC_INIT	FIFO_COMPR_INIT	0	0	FSM_INIT
MLC1_SRC	70h	MLC1_SRC_7	MLC1_SRC_6	MLC1_SRC_5	MLC1_SRC_4	MLC1_SRC_3	MLC1_SRC_2	MLC1_SRC_1	MLC1_SRC_0
MLC2_SRC	71h	MLC2_SRC_7	MLC2_SRC_6	MLC2_SRC_5	MLC2_SRC_4	MLC2_SRC_3	MLC2_SRC_2	MLC2_SRC_1	MLC2_SRC_0
MLC3_SRC	72h	MLC3_SRC_7	MLC3_SRC_6	MLC3_SRC_5	MLC3_SRC_4	MLC3_SRC_3	MLC3_SRC_2	MLC3_SRC_1	MLC3_SRC_0
MLC4_SRC	73h	MLC4_SRC_7	MLC4_SRC_6	MLC4_SRC_5	MLC4_SRC_4	MLC4_SRC_3	MLC4_SRC_2	MLC4_SRC_1	MLC4_SRC_0



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] 位置为 0000 时，这些寄存器可访问。

表 5. 嵌入高级功能寄存器——page 0

寄存器名称	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
SFLP_GAME_GBIASX_L	6Eh	GAME_GBIASX_7	GAME_GBIASX_6	GAME_GBIASX_5	GAME_GBIASX_4	GAME_GBIASX_3	GAME_GBIASX_2	GAME_GBIASX_1	GAME_GBIASX_0
SFLP_GAME_GBIASX_H	6Fh	GAME_GBIASX_15	GAME_GBIASX_14	GAME_GBIASX_13	GAME_GBIASX_12	GAME_GBIASX_11	GAME_GBIASX_10	GAME_GBIASX_9	GAME_GBIASX_8
SFLP_GAME_GBIASY_L	70h	GAME_GBIASY_7	GAME_GBIASY_6	GAME_GBIASY_5	GAME_GBIASY_4	GAME_GBIASY_3	GAME_GBIASY_2	GAME_GBIASY_1	GAME_GBIASY_0
SFLP_GAME_GBIASY_H	71h	GAME_GBIASY_15	GAME_GBIASY_14	GAME_GBIASY_13	GAME_GBIASY_12	GAME_GBIASY_11	GAME_GBIASY_10	GAME_GBIASY_9	GAME_GBIASY_8
SFLP_GAME_GBIASZ_L	72h	GAME_GBIASZ_7	GAME_GBIASZ_6	GAME_GBIASZ_5	GAME_GBIASZ_4	GAME_GBIASZ_3	GAME_GBIASZ_2	GAME_GBIASZ_1	GAME_GBIASZ_0
SFLP_GAME_GBIASZ_H	73h	GAME_GBIASZ_15	GAME_GBIASZ_14	GAME_GBIASZ_13	GAME_GBIASZ_12	GAME_GBIASZ_11	GAME_GBIASZ_10	GAME_GBIASZ_9	GAME_GBIASZ_8
FSM_EXT_SENSITIVITY_L	BAh	FSM_EXT_S_7	FSM_EXT_S_6	FSM_EXT_S_5	FSM_EXT_S_4	FSM_EXT_S_3	FSM_EXT_S_2	FSM_EXT_S_1	FSM_EXT_S_0
FSM_EXT_SENSITIVITY_H	BBh	FSM_EXT_S_15	FSM_EXT_S_14	FSM_EXT_S_13	FSM_EXT_S_12	FSM_EXT_S_11	FSM_EXT_S_10	FSM_EXT_S_9	FSM_EXT_S_8
FSM_EXT_OFFX_L	C0h	FSM_EXT_OFFX_7	FSM_EXT_OFFX_6	FSM_EXT_OFFX_5	FSM_EXT_OFFX_4	FSM_EXT_OFFX_3	FSM_EXT_OFFX_2	FSM_EXT_OFFX_1	FSM_EXT_OFFX_0
FSM_EXT_OFFX_H	C1h	FSM_EXT_OFFX_15	FSM_EXT_OFFX_14	FSM_EXT_OFFX_13	FSM_EXT_OFFX_12	FSM_EXT_OFFX_11	FSM_EXT_OFFX_10	FSM_EXT_OFFX_9	FSM_EXT_OFFX_8
FSM_EXT_OFFY_L	C2h	FSM_EXT_OFFY_7	FSM_EXT_OFFY_6	FSM_EXT_OFFY_5	FSM_EXT_OFFY_4	FSM_EXT_OFFY_3	FSM_EXT_OFFY_2	FSM_EXT_OFFY_1	FSM_EXT_OFFY_0
FSM_EXT_OFFY_H	C3h	FSM_EXT_OFFY_15	FSM_EXT_OFFY_14	FSM_EXT_OFFY_13	FSM_EXT_OFFY_12	FSM_EXT_OFFY_11	FSM_EXT_OFFY_10	FSM_EXT_OFFY_9	FSM_EXT_OFFY_8
FSM_EXT_OFFZ_L	C4h	FSM_EXT_OFFZ_7	FSM_EXT_OFFZ_6	FSM_EXT_OFFZ_5	FSM_EXT_OFFZ_4	FSM_EXT_OFFZ_3	FSM_EXT_OFFZ_2	FSM_EXT_OFFZ_1	FSM_EXT_OFFZ_0
FSM_EXT_OFFZ_H	C5h	FSM_EXT_OFFZ_15	FSM_EXT_OFFZ_14	FSM_EXT_OFFZ_13	FSM_EXT_OFFZ_12	FSM_EXT_OFFZ_11	FSM_EXT_OFFZ_10	FSM_EXT_OFFZ_9	FSM_EXT_OFFZ_8
FSM_EXT_MATRIX_XX_L	C6h	FSM_EXT_MAT_XX_7	FSM_EXT_MAT_XX_6	FSM_EXT_MAT_XX_5	FSM_EXT_MAT_XX_4	FSM_EXT_MAT_XX_3	FSM_EXT_MAT_XX_2	FSM_EXT_MAT_XX_1	FSM_EXT_MAT_XX_0
FSM_EXT_MATRIX_XX_H	C7h	FSM_EXT_MAT_XX_15	FSM_EXT_MAT_XX_14	FSM_EXT_MAT_XX_13	FSM_EXT_MAT_XX_12	FSM_EXT_MAT_XX_11	FSM_EXT_MAT_XX_10	FSM_EXT_MAT_XX_9	FSM_EXT_MAT_XX_8
FSM_EXT_MATRIX_XY_L	C8h	FSM_EXT_MAT_XY_7	FSM_EXT_MAT_XY_6	FSM_EXT_MAT_XY_5	FSM_EXT_MAT_XY_4	FSM_EXT_MAT_XY_3	FSM_EXT_MAT_XY_2	FSM_EXT_MAT_XY_1	FSM_EXT_MAT_XY_0
FSM_EXT_MATRIX_XY_H	C9h	FSM_EXT_MAT_XY_15	FSM_EXT_MAT_XY_14	FSM_EXT_MAT_XY_13	FSM_EXT_MAT_XY_12	FSM_EXT_MAT_XY_11	FSM_EXT_MAT_XY_10	FSM_EXT_MAT_XY_9	FSM_EXT_MAT_XY_8
FSM_EXT_MATRIX_XZ_L	CAh	FSM_EXT_MAT_XZ_7	FSM_EXT_MAT_XZ_6	FSM_EXT_MAT_XZ_5	FSM_EXT_MAT_XZ_4	FSM_EXT_MAT_XZ_3	MAG_SI_XZ_2	FSM_EXT_MAT_XZ_1	FSM_EXT_MAT_XZ_0
FSM_EXT_MATRIX_XZ_H	CBh	FSM_EXT_MAT_XZ_15	FSM_EXT_MAT_XZ_14	FSM_EXT_MAT_XZ_13	FSM_EXT_MAT_XZ_12	FSM_EXT_MAT_XZ_11	FSM_EXT_MAT_XZ_10	FSM_EXT_MAT_XZ_9	FSM_EXT_MAT_XZ_8
FSM_EXT_MATRIX_YY_L	CCh	FSM_EXT_MAT_YY_7	FSM_EXT_MAT_YY_6	FSM_EXT_MAT_YY_5	FSM_EXT_MAT_YY_4	FSM_EXT_MAT_YY_3	FSM_EXT_MAT_YY_2	FSM_EXT_MAT_YY_1	FSM_EXT_MAT_YY_0
FSM_EXT_MATRIX_YY_H	CDh	FSM_EXT_MAT_YY_15	FSM_EXT_MAT_YY_14	FSM_EXT_MAT_YY_13	FSM_EXT_MAT_YY_12	FSM_EXT_MAT_YY_11	FSM_EXT_MAT_YY_10	FSM_EXT_MAT_YY_9	FSM_EXT_MAT_YY_8
FSM_EXT_MATRIX_YZ_L	CEh	FSM_EXT_MAT_YZ_7	FSM_EXT_MAT_YZ_6	FSM_EXT_MAT_YZ_5	FSM_EXT_MAT_YZ_4	FSM_EXT_MAT_YZ_3	FSM_EXT_MAT_YZ_2	FSM_EXT_MAT_YZ_1	FSM_EXT_MAT_YZ_0
FSM_EXT_MATRIX_YZ_H	CFh	FSM_EXT_MAT_YZ_15	FSM_EXT_MAT_YZ_14	FSM_EXT_MAT_YZ_13	FSM_EXT_MAT_YZ_12	FSM_EXT_MAT_YZ_11	FSM_EXT_MAT_YZ_10	FSM_EXT_MAT_YZ_9	FSM_EXT_MAT_YZ_8
FSM_EXT_MATRIX_ZZ_L	D0h	FSM_EXT_MAT_ZZ_7	FSM_EXT_MAT_ZZ_6	FSM_EXT_MAT_ZZ_5	FSM_EXT_MAT_ZZ_4	FSM_EXT_MAT_ZZ_3	FSM_EXT_MAT_ZZ_2	FSM_EXT_MAT_ZZ_1	FSM_EXT_MAT_ZZ_0
FSM_EXT_MATRIX_ZZ_H	D1h	FSM_EXT_MAT_ZZ_15	FSM_EXT_MAT_ZZ_14	FSM_EXT_MAT_ZZ_13	FSM_EXT_MAT_ZZ_12	FSM_EXT_MAT_ZZ_11	FSM_EXT_MAT_ZZ_10	FSM_EXT_MAT_ZZ_9	FSM_EXT_MAT_ZZ_8
EXT_CFG_A	D4h	0	EXT_Y_AXIS2	EXT_Y_AXIS1	EXT_Y_AXIS0	0	EXT_Z_AXIS2	EXT_Z_AXIS1	EXT_Z_AXIS0
EXT_CFG_B	D5h	0	0	0	0	0	EXT_X_AXIS2	EXT_X_AXIS1	EXT_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] 位置为 0001 时，这些寄存器可访问。

表 6. 嵌入高级功能寄存器——page 1

寄存器名称	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
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	0
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
MLC_EXT_SENSITIVITY_L	E8h	MLC_EXT_S_7	MLC_EXT_S_6	MLC_EXT_S_5	MLC_EXT_S_4	MLC_EXT_S_3	MLC_EXT_S_2	MLC_EXT_S_1	MLC_EXT_S_0
MLC_EXT_SENSITIVITY_H	E9h	MLC_EXT_S_15	MLC_EXT_S_14	MLC_EXT_S_13	MLC_EXT_S_12	MLC_EXT_S_11	MLC_EXT_S_10	MLC_EXT_S_9	MLC_EXT_S_8

下表提供了嵌入高级功能 page 2 的寄存器列表。当 PAGE_SEL 寄存器中的 PAGE_SEL[3:0] 位置为 0010 时，这些寄存器可访问。

表 7. 嵌入高级功能寄存器——page 2

寄存器名称	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
EXT_FORMAT	00h	0	0	0	0	0	EXT_FORMAT_SEL	0	0
EXT_3BYTE_SENSITIVITY_L	02h	EXT_3BYTE_S_7	EXT_3BYTE_S_6	EXT_3BYTE_S_5	EXT_3BYTE_S_4	EXT_3BYTE_S_3	EXT_3BYTE_S_2	EXT_3BYTE_S_1	EXT_3BYTE_S_0
EXT_3BYTE_SENSITIVITY_H	03h	EXT_3BYTE_S_15	EXT_3BYTE_S_14	EXT_3BYTE_S_13	EXT_3BYTE_S_12	EXT_3BYTE_S_11	EXT_3BYTE_S_10	EXT_3BYTE_S_9	EXT_3BYTE_S_8
EXT_3BYTE_OFFSET_XL	06h	EXT_3BYTE_OFF_7	EXT_3BYTE_OFF_6	EXT_3BYTE_OFF_5	EXT_3BYTE_OFF_4	EXT_3BYTE_OFF_3	EXT_3BYTE_OFF_2	EXT_3BYTE_OFF_1	EXT_3BYTE_OFF_0
EXT_3BYTE_OFFSET_L	07h	EXT_3BYTE_OFF_15	EXT_3BYTE_OFF_14	EXT_3BYTE_OFF_13	EXT_3BYTE_OFF_12	EXT_3BYTE_OFF_11	EXT_3BYTE_OFF_10	EXT_3BYTE_OFF_9	EXT_3BYTE_OFF_8
EXT_3BYTE_OFFSET_H	08h	EXT_3BYTE_OFF_23	EXT_3BYTE_OFF_22	EXT_3BYTE_OFF_21	EXT_3BYTE_OFF_20	EXT_3BYTE_OFF_19	EXT_3BYTE_OFF_18	EXT_3BYTE_OFF_17	EXT_3BYTE_OFF_16



2.3

传感器集线器寄存器

下表提供了器件中可用的传感器集线器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 位置为 1 时，传感器集线器寄存器可访问。

表 8. 传感器集线器寄存器

寄存器名称	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	0	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
SLV0_CONFIG	17h	SHUB_ODR_2	SHUB_ODR_1	SHUB_ODR_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
SLV1_CONFIG	1Ah	0	0	0	1	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





寄存器名称	地址	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
SLV2_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
SLV3_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

LSM6DSV16X 提供了 3 种可能的操作配置：

- 只有加速度计工作，陀螺仪处于掉电模式或睡眠模式
- 只有陀螺仪工作，加速度计处于掉电模式
- 加速度计和陀螺仪均工作，且具有独立的 ODR 和功耗模式

器件提供从 1.71 V 到 3.6 V 的宽 Vdd 电压范围和从 1.08 V 到 3.6 V 的 Vdd_IO 范围。上电序列不受限制。Vdd/Vdd_IO 引脚可设置为电源电平或接地电平（不得悬空），无需特定序列即可上电。

为了避免潜在冲突，在上电过程期间，建议将连接到器件 IO 引脚的线路（在主机侧）设置成悬空状态或接地，直到设置 Vdd_IO 为止。设置 Vdd_IO 后，连接到 IO 引脚的线路必须根据内部引脚状态中列出的默认状态进行配置。为避免电流消耗意外增加，主机必须对未上拉/下拉的输入引脚进行极化。

施加 Vdd 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掉电模式。为保证器件能正确关断，建议将 Vdd 线接地的持续时间保持至少 100 μ s。

加速度计和陀螺仪可独立配置。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。

加速度计可配置为以下功耗模式之一：

- 掉电模式
- 低功耗模式（根据用于求平均值的测量结果数量，可以分为三种不同的模式）
- 正常模式
- 高性能模式
- 高精度 ODR 模式
- ODR 触发模式

陀螺仪可配置为以下功耗模式之一：

- 掉电模式
- 睡眠模式
- 低功耗模式
- 高性能模式
- 高精度 ODR 模式
- ODR 触发模式

3.1 加速度计功耗模式和输出数据率

可使用 CTRL1 寄存器选择加速度计的功耗模式和输出数据率。

当加速度计配置为掉电模式时，几乎所有的内部模块都会关闭，以最大限度地降低功耗。数字接口（I²C、MIPI I3C[®] 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，可保存进入掉电模式前存储器中采样的最后数据。

当加速度计配置为低功耗模式时，其读取链自动接通和断开，以便优化电流消耗。生成采样时需要对多个测量结果求平均值，而根据测量结果的数量可以分为以下三种不同的低功耗模式：

- 低功耗模式 1 (LPM1)，对两个测量结果求平均值
- 低功耗模式 2 (LPM2)，对四个测量结果求平均值
- 低功耗模式 3 (LPM3)，对八个测量结果求平均值

增加用于求平均值的测量结果数量可以降低噪声，而减少用于求平均值的测量结果数量可以降低电流消耗。

在这些低功耗模式下，将禁用抗混叠滤波器，加速度的 ODR 最高可设置 240 Hz。

当加速度计配置为正常模式时，其读取链始终接通。此时，将使能抗混叠滤波器，加速度的 ODR 最高可设置 1920 Hz。正常模式可在噪声与电流消耗之间达到平衡。

当加速度计配置为高性能模式时，其读取链始终接通。此时，将使能抗混叠滤波器，加速度的 ODR 最高可设置 7680 Hz。高性能模式在噪声方面具有最佳性能。

当加速度计配置为高精度 ODR (HAODR) 模式时，其读取链始终接通。此时，将使能抗混叠滤波器，加速度计 ODR 是可选项（15 Hz 到 7680 Hz）。高精度 ODR 模式在噪声方面具有最佳性能（与高性能模式相同），通常可降低不同器件之间的 ODR 差异。

如果要使用高精度 ODR 模式，必须考虑以下限制：

- 当加速度计和陀螺仪中有一个传感器要使用 HAODR 模式时，另一个传感器也必须配置为 HAODR 模式。

- HAODR 模式不适用于加速度计和陀螺仪 OIS 通道。
- 使能或禁用 HAODR 模式前，加速度计和陀螺仪必须设置为掉电模式（CTRL1 寄存器中的 ODR_XL = 0000 且 CTRL2 寄存器中的 ODR_G = 0000）。
- HAODR 模式与 Qvar 功能不兼容。
- HAODR 模式与活动/不活动功能不兼容（只能使用运动/静止）。

表 9 汇总了可用的功耗模式（基于 CTRL1 寄存器的 OP_MODE_XL 位）。如果 ODR_XL = 0000，则选择掉电模式，而与 OP_MODE_XL 位的配置无关。

表 9. 加速度计功耗模式

OP_MODE_XL [2:0]	功耗模式
000	高性能（默认）
001	高精度 ODR
100	低功耗模式 1
101	低功耗模式 2
110	低功耗模式 3
111	正常

表 10 汇总了可用的 ODR 值（基于 CTRL1 寄存器的 ODR_XL 位）。

表 10. 加速度计 ODR

ODR_XL [3:0]	选定的 ODR [Hz]	高性能	高精度 ODR	正常	低功耗模式 1/2/3
0000	掉电（默认）	•	•	•	•
0001	1.875				•
0010	7.5	•		•	
0011	15	•	•	•	•
0100	30	•	•	•	•
0101	60	•	•	•	•
0110	120	•	•	•	•
0111	240	•	•	•	•
1000	480	•	•	•	
1001	960	•	•	•	
1010	1920	•	•	•	
1011	3840	•	•		
1100	7680	•	•		

3.2 陀螺仪功耗模式和输出数据率

可使用 **CTRL2** 寄存器选择陀螺仪的功耗模式和输出数据率。

当陀螺仪配置为掉电模式时，器件几乎所有的内部模块都会关闭，以最大限度地降低功耗。数字接口（**I²C**、**MIPI I3C[®]** 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，可保存进入掉电模式前存储器中采样的最后数据。

当陀螺仪处于睡眠模式时，陀螺仪块的振荡驱动电路工作，但读取链断开。相比于掉电模式，从睡眠模式到任何工作模式的开启时间大幅度减少。

当陀螺仪配置为低功耗模式时，驱动电路始终开启，但读取链自动接通和断开，以便优化电流消耗。陀螺仪 **ODR** 最高可设置 **240 Hz**。

当陀螺仪配置为高性能模式时，其读取链始终接通。陀螺仪 **ODR** 最高可设置 **7680 Hz**。高性能模式在噪声方面具有最佳性能。

当陀螺仪配置为高精度 **ODR (HAODR)** 模式时，其读取链始终接通。陀螺仪 **ODR** 可选（**15 Hz** 到 **7680 Hz**）。高精度 **ODR** 模式在噪声方面具有最佳性能（与高性能模式相同），通常可降低不同器件之间的 **ODR** 差异。

如果要使用高精度 **ODR** 模式，必须考虑以下限制：

- 当加速度计和陀螺仪中有一个传感器要使用 **HAODR** 模式时，另一个传感器必须也配置为 **HAODR** 模式。
- **HAODR** 模式不适用于加速度计和陀螺仪 **OIS** 通道。
- 使能或禁用 **HAODR** 模式前，加速度计和陀螺仪必须设置为掉电模式（**CTRL1** 寄存器中的 **ODR_XL = 0000** 且 **CTRL2** 寄存器中的 **ODR_G = 0000**）。
- **HAODR** 模式与 **Qvar** 功能不兼容。
- **HAODR** 模式与活动/不活动功能不兼容（只能使用运动/静止）。

表 11 汇总了可用的功耗模式（基于 **CTRL2** 寄存器的 **OP_MODE_G** 位）。如果 **ODR_G = 0000**，则选择掉电模式，而与 **OP_MODE_G** 位的配置无关。

表 11. 陀螺仪功耗模式

OP_MODE_G [2:0]	功耗模式
000	高性能
001	高精度 ODR
100	睡眠
101	低功耗

表 12 汇总了可用的 **ODR**（基于 **CTRL2** 寄存器的 **ODR_G** 位）。

表 12. 陀螺仪 ODR

ODR_G [3:0]	选定的 ODR [Hz]	高性能	高精度 ODR	低功耗模式
0000	掉电（默认）	•	•	•
0010	7.5	•		•
0011	15	•	•	•
0100	30	•	•	•
0101	60	•	•	•
0110	120	•	•	•
0111	240	•	•	•
1000	480	•	•	
1001	960	•	•	
1010	1920	•	•	
1011	3840	•	•	
1100	7680	•	•	

3.3 电流消耗

表 13 显示了不同工作模式下的功耗典型值。

表 13. 功耗 (@ Vdd = 1.8 V, T = 25 °C)

ODR [Hz]	只有加速度计	只有陀螺仪	[加速度计 + 陀螺仪] 组合
	[μA]	[μA]	[μA]
掉电	2.6	2.6	2.6
1.875 Hz (低功耗模式 1)	4.0	-	-
15 Hz (低功耗模式 1)	7.1	-	-
30 Hz (低功耗模式 1)	10.5	-	-
60 Hz (低功耗模式 1)	17.0	-	-
120 Hz (低功耗模式 1)	29.5	-	-
240 Hz (低功耗模式 1)	54.5	-	-
1.875 Hz (低功耗模式 2)	4.2	-	280
7.5 Hz (低功耗模式 2)	-	257	
15 Hz (低功耗模式 2)	7.8	262	285
30 Hz (低功耗模式 2)	12.0	272	295
60 Hz (低功耗模式 2)	20.0	289	316
120 Hz (低功耗模式 2)	35.1	325	358
240 Hz (低功耗模式 2)	66.1	397	441
1.875 Hz (低功耗模式 3)	4.4	-	-
15 Hz (低功耗模式 3)	9.2	-	-
30 Hz (低功耗模式 3)	14.6	-	-
60 Hz (低功耗模式 3)	25.6	-	-
120 Hz (低功耗模式 3)	46.6	-	-
240 Hz (低功耗模式 3)	88.6	-	-
所有 ODR (正常模式)	100.0	-	-
所有 ODR (高性能模式)	190.0	535	650
所有 ODR (高精度 ODR 模式)	210.0	555	670

3.4 连接模式

器件提供了三种不同的连接模式，本文档中作了详细描述：

- 模式 1：默认使能的连接模式。I²C 从接口、MIPI I3C[®] 从接口或 SPI (3/4 线) 串行接口可用。
- 模式 2：传感器集线器 (sensor hub) 模式。I²C 从接口、MIPI I3C[®] 从接口或 SPI (3/4 线) 串行接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——传感器集线器 (sensor hub) 模式中描述了此连接模式。
- 模式 3：除了主 I²C 从接口、MIPI I3C[®] 从接口或 SPI (3/4 线) 串行接口，还有一个用于外部器件连接（例如，照相机模块）的辅助 SPI (3/4 线) 串行接口可用。第 8 节 模式 3——OIS 功能中描述了此连接模式。

3.5 加速度计带宽

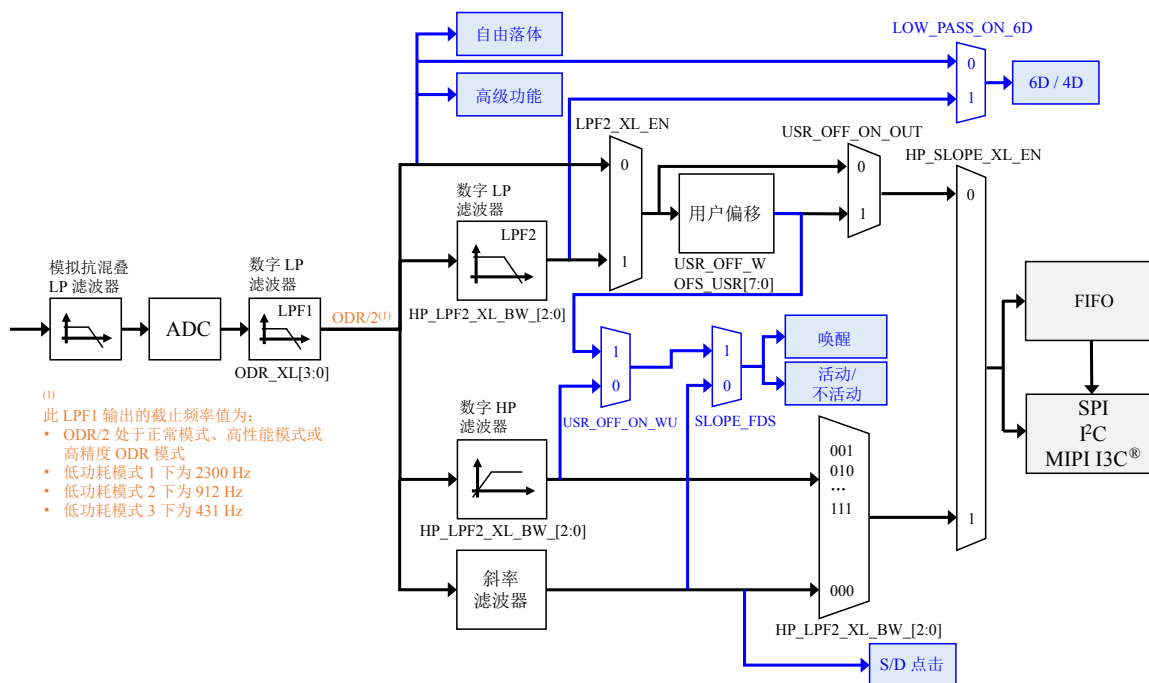
加速度计采样链由 4 个级联主模块表示：一个模拟抗混叠低通滤波器、一个 ADC 转换器、一个数字低通滤波器 (LPF1) 和复合数字滤波器组。

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——OIS 功能描述了 OIS 路径上的激活加速度计采样链 (使用模式 3 配置时)。

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。该抗混叠滤波器无法在低功耗模式下使用。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 (当加速度计配置为正常模式、高性能模式或高精度 ODR 模式时)
- 2300 Hz (当加速度计配置为低功耗模式 1 时)
- 912 Hz (当加速度计配置为低功耗模式 2 时)
- 431 Hz (当加速度计配置为低功耗模式 3 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、步伐检测、步数计算、大幅运动检测和倾斜检测功能 (如第 6 节 嵌入功能所述)，还包括有限状态机和机器学习内核。

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

CTRL8 寄存器和 CTRL9 寄存器的 HP_LPF2_XL_BW_[2:0] 位可用来配置复合滤波器组和加速度计滤波链的总带宽，如表 14. 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。在高通路侧，如果 HP_LPF2_XL_BW_[2:0] = 000，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 14. 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，针对要丢弃的采样还提供了最大 (最坏情况) 的稳定时间。

表 14. 模式 1/2/3 下的加速度计带宽选择

HP_SLOPE_XL_EN	LPF2_XL_EN	HP_LPF2_XL_BW [2:0]	带宽	最大总稳定时间 ⁽¹⁾ (要丢弃的采样数)
0 (低通路径)	0	-	ODR / 2 ⁽²⁾	见 表 16
	1	000	ODR / 4	见 表 16
		001	ODR / 10	21
		010	ODR / 20	21
		011	ODR / 45	39
		100	ODR / 100	78
		101	ODR / 200	156
		110	ODR / 400	313
		111	ODR / 800	626
		1 (高通路径)	-	000
001	ODR / 10			27
010	ODR / 20			27
011	ODR / 45			39
100	ODR / 100			78
101	ODR / 200			156
110	ODR / 400			313
111	ODR / 800			626

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
2. 当加速度计处于高性能模式、高精度 ODR 模式和正常模式时，该值为 ODR / 2。当加速度计处于低功耗模式 1 时（对 2 个测量结果求平均值），该值等于 2300 Hz；当加速度计处于低功耗模式 2 时（对 4 个测量结果求平均值），该值等于 912 Hz；当加速度计处于低功耗模式 3 时（对 8 个测量结果求平均值），该值等于 431 Hz。

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 寄存器的 HP_LPF2_XL_BW [2:0] 位域来设置加速度计链的总带宽。

通过将 TAP_CFG0 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，可选择复合滤波器模块的高通路径。HP_LPF2_XL_BW [2:0] 位域除了能用于使能 LPF1 滤波器之外，还可以使能斜率滤波器（HP_LPF2_XL_BW [2:0] = 000 时）或数字高通滤波器（其他 HP_LPF2_XL_BW [2:0] 配置）。HP_LPF2_XL_BW [2:0] 位域也可用来选择 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：使能此功能后，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 CTRL9 寄存器的 HP_REF_MODE_XL 位和 HP_SLOPE_XL_EN 位置为 1，并且 HP_LPF2_XL_BW [2:0] 位域的值必须不等于 000。当使能参考模式功能时，LPF2 滤波器和 HP 滤波器都不可用。使能参考模式之后的第一个加速度计输出数据必须被丢弃。

CTRL9 寄存器的 XL_FASTSETTL_MODE 位使能加速度计 LPF2 或 HPF 快速稳定模式：选择的滤波器在写入此位后设置第一个采样。仅在器件退出掉电模式时应用此功能。

3.5.1 加速度计斜率滤波器

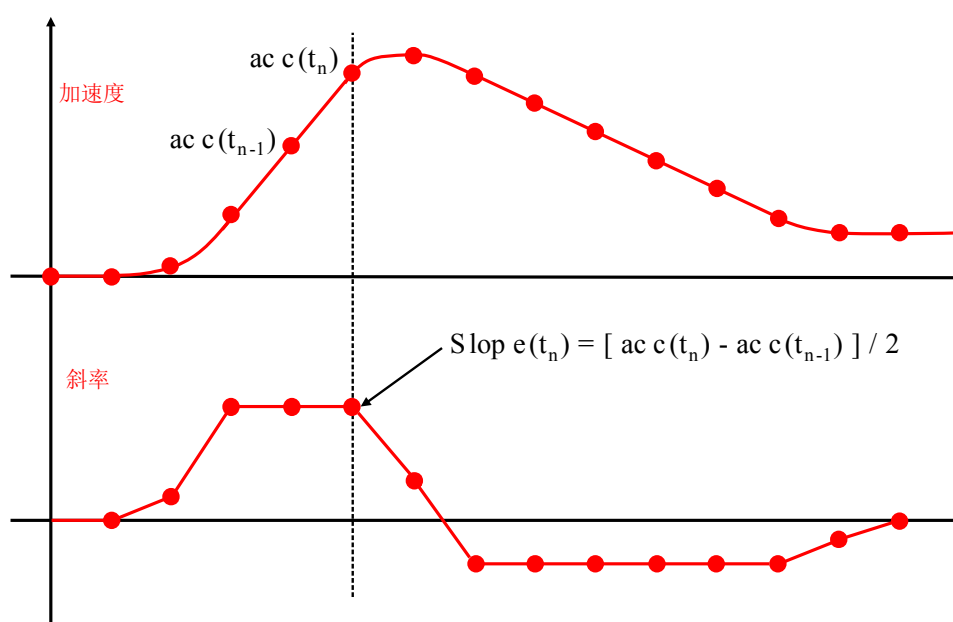
如图 2. 加速度计滤波链（UI 路径）中所示，器件嵌入了一个数字斜率滤波器，该滤波器还可用于某些嵌入功能，如单/双击识别、唤醒检测和活动/不活动。

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



3.6 加速度计开启/关闭时间

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 15. 加速度计开启/关闭时间（LPF2 和 HP 禁用）和表 16. 要丢弃的加速度计采样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。

提示

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 15. 加速度计开启/关闭时间（LPF2 和 HP 禁用）

起始模式	目标模式	(1)最终值的 99% 时的稳定时间
掉电	低功耗模式	见 表 16
掉电	正常模式	见 表 16
掉电	高性能模式	见 表 16
掉电	高精度 ODR 模式	见 表 16
正常模式/低功耗模式	高性能模式	见表 16 + 丢弃 1 个额外采样
高性能模式/低功耗模式	正常模式	见表 16 + 丢弃 1 个额外采样
高性能模式/正常模式	低功耗模式	见表 16 + 丢弃 1 个额外采样
低功耗模式	低功耗模式（ODR 改变）	见 表 16
正常模式	正常模式（ODR 改变）	丢弃 3 个采样
高性能模式	高性能模式（ODR 改变）	丢弃 4 个采样
低功耗模式/正常模式/高性能模式/高精度 ODR 模式	掉电	1 μ s

1. 最大开启/关闭时间
表 16. 要丢弃的加速度计采样

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)			
	低功耗模式	正常模式	高性能模式	高精度 ODR 模式	低功耗模式	正常模式	高性能模式	高精度 ODR 模式
1.875 Hz	0 (第一个采样正确)	-	-	-	1	-	-	-
7.5 Hz	-	1	1	-	-	2	2	-
15 Hz	0 (第一个采样正确)	1	1	4	1	2	2	5
30 Hz	0 (第一个采样正确)	1	1	4	1	2	2	5
60 Hz	0 (第一个采样正确)	1	1	4	1	2	2	5
120 Hz	0 (第一个采样正确)	1	1	4	1	2	2	5
240 Hz	0 (第一个采样正确)	1	1	4	1	2	2	5
480 Hz	-	1	1	4	-	2	2	5
960 Hz	-	2	1	4	-	3	2	5
1920 Hz	-	3	4	8	-	4	5	9
3840 Hz	-	-	12	16	-	-	13	17
7680 Hz	-	-	27	32	-	-	27	32

表 14. 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。

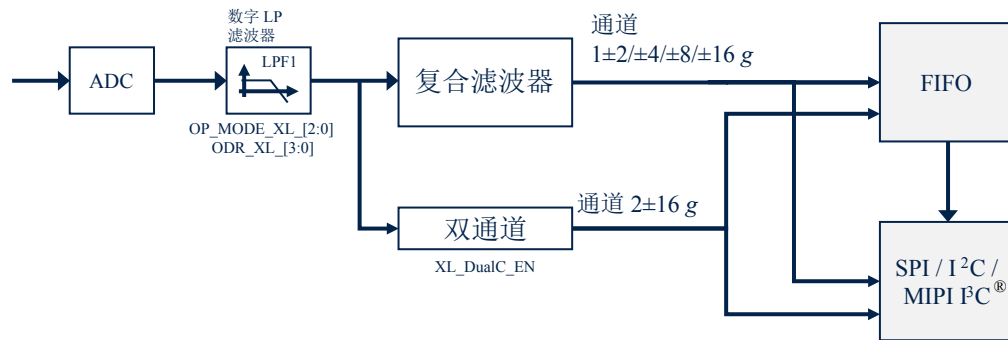
当器件配置为模式 3 时，加速度计 UI 路径滤波链不受加速度计/陀螺仪 OIS 路径滤波链使能/禁用的影响。

3.7 加速度计双通道模式

LSM6DSV16X 加速度计内部采用双通道架构，能够同时提供两组满量程值不同的加速度数据。默认情况下，器件在单通道模式下工作，支持从 $\pm 2\text{ g}$ 到 $\pm 16\text{ g}$ 的满量程值和多种功耗模式。通过将 CTRL8 寄存器中的 XL_DualC_EN 位配置为 1（使能）或 0（禁用），可以使能/禁用双通道功能。该功能可用于加速度计的所有工作模式。

使能/禁用双通道模式不会影响加速度计通道 1，该通道一直以通过 CTRL8 寄存器中的 FS_XL_[1:0] 位设置的满量程工作。

图 4. 双通道模式



如图 4 所示，使能双通道模式时，加速度计链的一条额外路径（通道 2）以 $\pm 16\text{ g}$ 满量程和通过 CTRL1_XL 寄存器的 ODR_XL 位域配置的输出数据率生成数据。加速度计通道 2 数据位于输出寄存器 UI_OUTX_L_A_OIS_DualC 到 UI_OUTZ_H_A_OIS_DualC（34h 到 39h）中，用二进制补码表示。通道 2 的带宽和稳定时间受 LPF1 数字低通滤波器影响。更多详细信息，请参见第 3.6 节 加速度计开启/关闭时间。

此外，通过使用 LSM6DSV16X 器件中嵌入的有限状态机 (FSM) 的一些专用命令，还可以将采用 $\pm 16\text{ g}$ 满量程的加速度计通道 2 数据存储到 FIFO 中。可通过将 FIFO_CTRL2 寄存器中的 XL_DualC_BATCH_FROM_FSM 位置为 1 来使能该功能。更多详细信息，请参见第 9 节 先进先出 (FIFO) 缓冲器。

使能双通道模式时，除非已使能由 FSM 触发的 FIFO 中对加速度计通道 2 数据的批处理，否则无法通过主接口访问 OIS 加速度计数据（模式 3），只能通过辅助 SPI 接口访问这些数据。

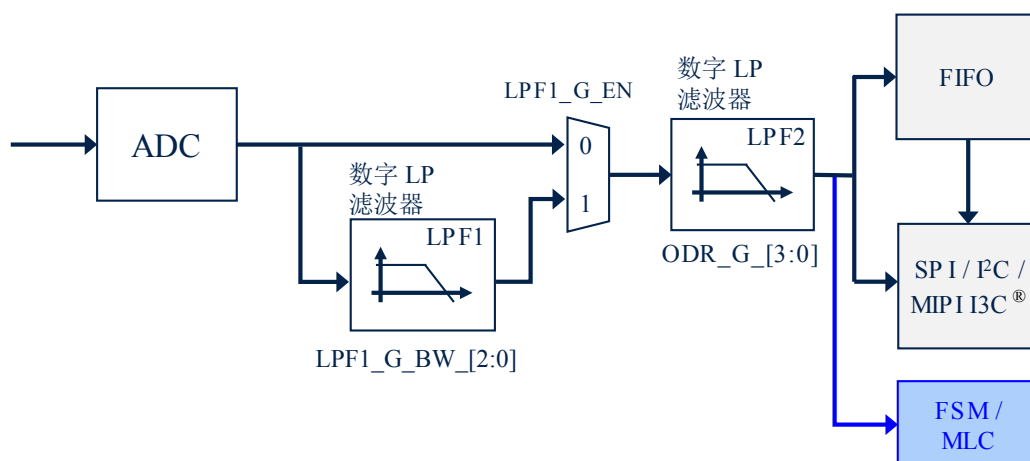
3.8 陀螺仪带宽

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 5 所示。它由两个滤波器级联而成：一个可选数字低通滤波器 (LPF1) 和一个数字低通滤波器 (LPF2)。

LPF1 滤波器支持高性能模式和高精度 ODR 模式。如果陀螺仪配置为低功耗模式，则 LPF1 滤波器会被 bypass。

图 5. 陀螺仪数字链——模式 1 (UI/EIS) 和模式 2



提示

当陀螺仪 OIS 或 EIS 链使能时，LPF1 滤波器在陀螺仪 UI 链上不可用。如欲使用陀螺仪 OIS 或 EIS，建议避免在陀螺仪 UI 链中使用 LPF1 滤波器。

数字 LPF1 滤波器可以通过将 CTRL7 寄存器的 LPF1_G_EN 位置为 1 来使能，其带宽可以通过 CTRL6 寄存器的 LPF1_G_BW[2:0] 位域来选择。

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 7680 Hz 时，LPF2 滤波器被 bypass。

下表汇总了在 CTRL7 寄存器的 LPF1_G_EN 位和 CTRL6 寄存器的 LPF1_G_BW[2:0] 位域的不同配置下，不同陀螺仪 ODR 值的总陀螺仪带宽。

表 17. 模式 1/2 下的陀螺仪总带宽选择

陀螺仪 ODR [Hz]	LPF1	LPF2	总 LPF 截止频率 [Hz] (相位 @ 20 Hz)
	LPF1_G_BW[2:0]		
7.5 Hz	Bypass	使能	3.4 (-62.7° @ 2.5 Hz)
	0xx		3.4 (-63.4° @ 2.5 Hz)
	100		3.4 (-64.6° @ 2.5 Hz)
	101		3.3 (-66.1° @ 2.5 Hz)
	110		3.3 (-67.7° @ 2.5 Hz)
	111		3.2 (-72.6° @ 2.5 Hz)
15 Hz	Bypass	使能	6.6 (-65.3° @ 5 Hz)
	0xx		6.6 (-66.7° @ 5 Hz)
	100		6.6 (-69.2° @ 5 Hz)
	101		6.6 (-72.1° @ 5 Hz)
	110		6.4 (-75.2° @ 5 Hz)
	111		5.9 (-84.6° @ 5 Hz)
30 Hz	Bypass	使能	13.0 (-70.4° @ 10 Hz)
	0xx		13.0 (-73.2° @ 10 Hz)

陀螺仪 ODR [Hz]	LPF1	LPF2	总 LPF 截止频率 [Hz] (相位 @ 20 Hz)
	LPF1_G_BW [2:0]		
30 Hz	100	使能	13.0 (-78.3° @ 10 Hz)
	101		13.0 (-84.2° @ 10 Hz)
	110		11.6 (-89.7° @ 10 Hz)
	111		9.3 (-105° @ 10 Hz)
60 Hz	Bypass	使能	24.6 (-80.6°)
	0xx		24.6 (-86.1°)
	100		24.6 (-96.5°)
	101		24.6 (-109°)
	110		18.0 (-116°)
	111		12.1 (-135°)
120 Hz	Bypass	使能	49.4 (-42.8°)
	0xx		49.4 (-48.4°)
	100		49.4 (-58.7°)
	101		42.6 (-71.5°)
	110		24.2 (-77.9°)
	111		13.6 (-97.5°)
240 Hz	Bypass	使能	96 (-24.8°)
	0xx		96 (-30.4°)
	100		78.4 (-40.7°)
	101		53 (-53.6°)
	110		27.3 (-59.9°)
	111		14.2 (-79.5°)
480 Hz	Bypass	使能	187 (-15.6°)
	000		175 (-21.1°)
	001		157 (-23.0°)
	010		131 (-25.9°)
	011		188 (-19.1°)
	100		94 (-31.5°)
	101		56.7 (-44.3°)
	110		28.4 (-50.7°)
	111		14.3 (-70.3°)
960 Hz	Bypass	使能	342 (-10.9°)
	000		241 (-16.4°)
	001		195 (-18.3°)
	010		149 (-21.1°)
	011		310 (-14.4°)
	100		100 (-26.8°)
	101		57.9 (-39.6°)
	110		28.7 (-46.0°)
	111		14.4 (-65.6°)
1920 Hz	Bypass	使能	491 (-8.3°)
	000		273 (-13.9°)

陀螺仪 ODR [Hz]	LPF1	LPF2	总 LPF 截止频率 [Hz] (相位 @ 20 Hz)
	LPF1_G_BW_[2:0]		
1920 Hz	001	使能	210 (-15.8°)
	010		155 (-18.6°)
	011		387 (-11.8°)
	100		101 (-24.3°)
	101		58.2 (-37.1°)
	110		28.8 (-43.5°)
	111		14.4 (-63.1°)
3840 Hz	Bypass	使能	528 (-7.4°)
	000		280 (-13.0°)
	001		213 (-14.9°)
	010		156 (-17.7°)
	011		403 (-10.9°)
	100		102 (-23.3°)
	101		58.3 (-36.1°)
	110 ⁽¹⁾		28.8 (-42.5°)
	111 ⁽¹⁾		14.4 (-62.1°)
7680 Hz	Bypass	禁用	537 (-6.9°)
	000		281 (-12.5°)
	001		213 (-14.4°)
	010		156 (-17.2°)
	011		407 (-10.4°)
	100		102 (-22.9°)
	101		58 (-35.7°)
	110 ⁽¹⁾		28.8 (-42.1°)
	111 ⁽¹⁾		14.4 (-61.7°)

1. ODR ≥ 3840 Hz 时, 由于高频时会出现低 LPF1 衰减, 因此应避免 LPF1_G_BW_[2:0] = 11x 的情况。

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

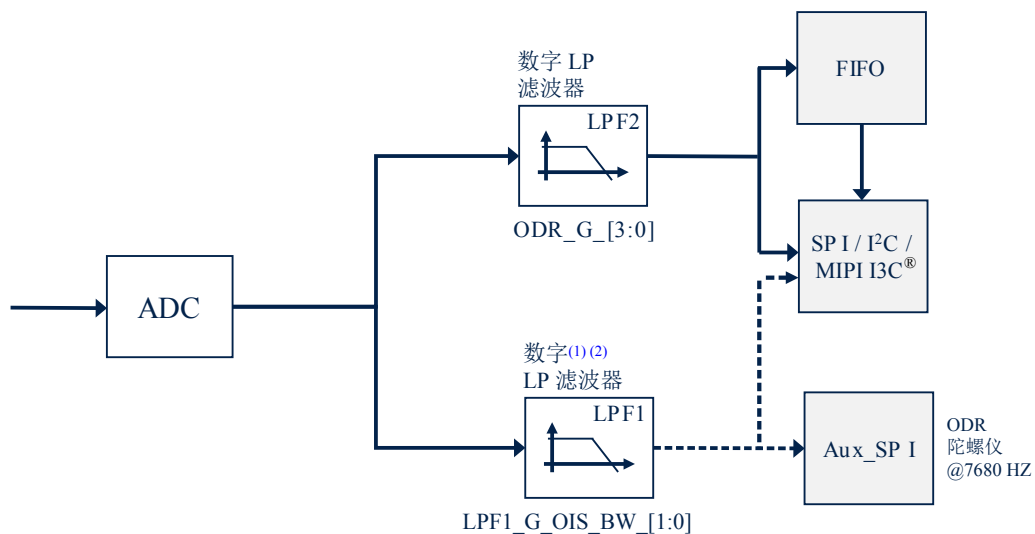
表 18. 陀螺仪低功耗模式下的带宽

陀螺仪 ODR [Hz]	带宽 [Hz]
7.5	2.3
15	4.6
30	9.1
60	18
120	36
240	71

当模式 3 使能时, 陀螺仪数字链如图 6 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 I3C[®]/SPI, 可选 ODR 范围为 7.5 Hz 到 7680 Hz。
- 光学防抖 (OIS) 链, 将陀螺仪数据提供给辅助 SPI, ODR 固定在 7680 Hz。

图 6. 陀螺仪数字链——模式 3 (OIS)



1. 当陀螺仪 OIS 或 EIS 链使能时，LPF1 滤波器在陀螺仪 UI 链上不可用。
2. 如欲使用陀螺仪 OIS 或 EIS 链，建议避免在模式 1/2 下使用 LPF1 滤波器。

在模式 3 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 19 所示。如果陀螺仪配置为低功耗模式，则陀螺仪链带宽仍如表 18 所示。

表 19. UI 链——模式 3 下的陀螺仪总带宽选择

陀螺仪 ODR [Hz]	带宽 [Hz]
7.5	3.4
15	6.6
30	13.0
60	24.6
120	49
240	96
480	185
960	353
1920	491
3840	528
7680	537

第 8 节 模式 3——OIS 功能中提供了模式 3 连接模式和陀螺仪 OIS 链的详细描述。

3.9 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/关闭时间如表 20. 模式 1/2 下的陀螺仪开启/关闭时间 所示。

提示 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 20. 模式 1/2 下的陀螺仪开启/关闭时间

起始模式	目标模式	最大开启/关闭时间 ⁽¹⁾
掉电	睡眠	70 ms
掉电	低功耗模式	70 ms + 丢弃 1 个采样
掉电	高性能模式	70 ms + 表 21 + 表 22 ⁽²⁾
掉电	高精度 ODR 模式	70 ms + 表 21 + 表 22 ⁽²⁾
睡眠	低功耗模式	丢弃 1 个采样
睡眠	高性能模式	表 21 + 表 22 ⁽²⁾
低功耗模式	高性能模式	表 21 + 表 22 ⁽²⁾
低功耗模式	低功耗模式（ODR 改变）	丢弃 1 个采样
高性能模式	低功耗模式	丢弃 1 个采样
高性能模式	高性能模式（ODR 改变）	丢弃 2 个采样
低功耗/高性能/高精度 ODR 模式	掉电	5 ms

1. @ 最终值的 99% 时的稳定时间

2. 仅当 LPF1 使能时

表 21. 模式 1/2 下要丢弃的陀螺仪采样（LPF1 禁用）

陀螺仪 ODR [Hz]	要丢弃的采样数 ⁽¹⁾ (高性能模式)	要丢弃的采样数 ⁽¹⁾ (高精度 ODR 模式)
7.5 Hz	2	-
15 Hz	2	5
30 Hz	2	5
60 Hz	3	6
120 Hz	3	6
240 Hz	4	7
480 Hz	5	8
960 Hz	6	9
1920 Hz	10	13
3840 Hz	18	21
7680 Hz	33	36

1. 最终值的 99% 时的稳定时间

表 22. 模式 1/2 下的陀螺仪链稳定时间 (LPF1 使能)

LPF1_G_BW_[2:0]	每种 ODR 的最长稳定时间 [ms] ⁽¹⁾
000	3.7
001	5.0
010	6.9
011	2.4
100	10.7
101	18.3
110	25.9
111	51.8

1. 最终值的 99% 时的稳定时间

当器件配置为模式 3 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。

3.10 陀螺仪 EIS 通道

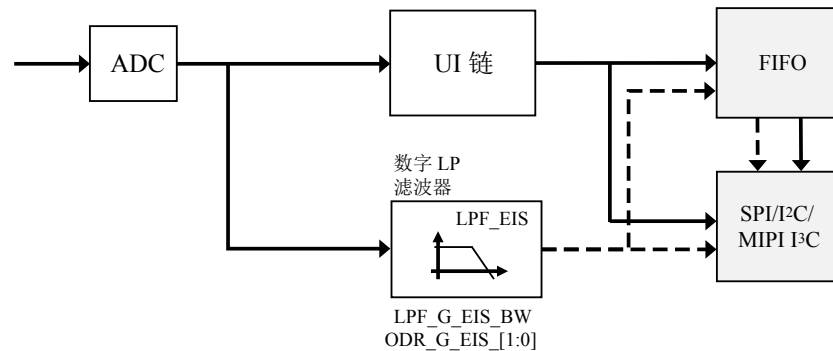
LSM6DSV16X 集成了一个专为 EIS（电子图像稳定）应用而设计的独立陀螺仪通道。从根本上说，就是提供了一个额外的专用陀螺仪通道，可通过主 I²C/MIPI I³C[®]/SPI 接口访问，并且满量程和滤波链可单独配置。

可通过配置 CTRL_EIS 寄存器的 ODR_G_EIS_[1:0] 位域来使能/禁用陀螺仪 EIS 通道。默认情况下，EIS 通道关闭 (ODR_G_EIS_[1:0] = 00)。当这两位设置为 01 或 10 时，EIS 通道使能，并且 EIS 陀螺仪数据分别以 1920 Hz 或 960 Hz 的速率生成。只有这两种数据率适用于 EIS 通道。

提示

当 EIS 通道使能时，陀螺仪 UI 只能配置为掉电模式、高性能模式或高精度 ODR 模式。

图 7. 陀螺仪 EIS 通道



可使用 CTRL_EIS 寄存器的 FS_G_EIS_[2:0] 位域设置 EIS 陀螺仪数据的满量程。

提示

如果 CTRL6 中的 FS_G_[3:0] 位等于 1100 (± 4000 dps)，则 FS_G_EIS_[2:0] 必须设置为 100 以在 UI 和 EIS 通道上选择 ± 4000 dps 满量程。

如图 7 所示，EIS 通道可使用专用滤波链。LPF_EIS 数字低通滤波器的带宽取决于 CTRL_EIS 寄存器中的 ODR_G_EIS_[1:0] 位域和 LPF_G_EIS_BW 位所应用的配置，如表 23 所示，表中还列出了陀螺仪退出掉电模式时，在 70 ms 等待时间后要丢弃的采样数。

表 23. 陀螺仪 LPF_EIS 滤波器带宽选择

ODR_G_EIS_[1:0]	陀螺仪 EIS ODR	LPF_G_EIS_BW	截止频率	相位 @ 20 Hz	要丢弃的采样数 ⁽¹⁾ 高性能模式	要丢弃的采样数 ⁽¹⁾ 高精度 ODR 模式
01	1920 Hz	0	153 Hz	-13.5°	15	17
		1	203 Hz	-10.8°	12	14
10	960 Hz	0	148 Hz	-15.4°	8	10
		1	193 Hz	-12.7°	6	8

1. 最终值的 99% 时的稳定时间

使能 EIS 通道后，通过将 CTRL_EIS 寄存器中的 G_EIS_ON_G_OIS_OUT_REG 位置为 1，可使用输出数据寄存器 UI_OUTX_L_G_OIS_EIS 到 UI_OUTZ_H_G_OIS_EIS (2Eh 到 33h) 中的 EIS 陀螺仪数据。EIS 通道与模式 3 选项兼容，但当 G_EIS_ON_G_OIS_OUT_REG 位置为 1 时，无法通过主接口读取陀螺仪 OIS 数据（只能通过辅助 SPI 接口访问这些数据）。

对于 EIS 陀螺仪通道，数据准备就绪信号由 STATUS_REG 寄存器的 GDA_EIS 位表示。通过将 INT2_CTRL 寄存器的 INT2_DRDY_G_EIS 位置为 1，可将该信号驱动至 INT2 引脚。

还可通过将 FIFO_CTRL4 寄存器中的 G_EIS_FIFO_EN 位置为 1，将 EIS 陀螺仪数据存储到 FIFO 中。请参见第 9 节 先进先出 (FIFO) 缓冲器以获得更多信息。

4 模式 1 - 读取输出数据

4.1 启动序列

当器件上电时，器件会自动从内嵌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，10 ms（最大值）后，加速度计和陀螺仪自动进入掉电模式。

要开启加速度计并通过主 I²C/MIPI I3C[®]/SPI 接口采集加速度数据，需要通过 CTRL1 寄存器选择某一种工作模式。以下通用序列可用来配置加速度计：

1. 写入 INT1_CTRL = 01h // INT1 上加速度计数据准备就绪中断
2. 写入 CTRL1 = 08h // ODR_XL = 480 Hz（高性能模式）

要开启陀螺仪并通过主 I²C/MIPI I3C[®]/SPI 接口采集角速率数据，需要通过 CTRL2 寄存器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. 写入 INT1_CTRL = 02h // INT1 上陀螺仪数据准备就绪中断
2. 写入 CTRL2 = 08h // ODR_G = 480 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，可供轮询以检查新的一组数据何时可用。当加速度计输出寄存器中有一组新的数据可用时，XLDA 位置为 1。当陀螺仪输出寄存器中有一组新的数据可用时，GDA 位置为 1。

对于加速度计（陀螺仪也是类似的），可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，则跳到步骤 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

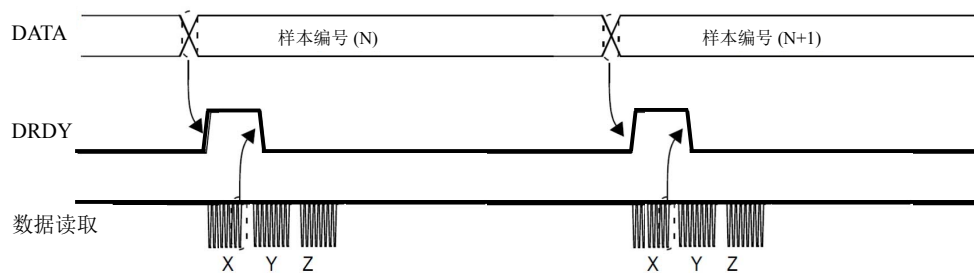
该器件可配置为具有一个硬件信号，以确定新的一组测量数据何时可以读取。

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_G 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。数据准备就绪信号可以是锁存的或脉冲的。如果 CTRL4 寄存器的 DRDY_PULSED 位置为 0（默认值），则数据准备就绪信号被锁存，并且当其中某一个轴的高字节（对于加速度计，为 29h、2Bh 和 2Dh 寄存器；对于陀螺仪，为 23h、25h 和 27h 寄存器）被读取时，中断复位。如果 CTRL4 寄存器的 DRDY_PULSED 位置为 1，则数据准备就绪信号是脉冲的，并且在中断引脚上观察到的脉冲持续时间为 65 μ s。如果加速度计或陀螺仪配置为 HAODR 模式，则在中断引脚上观察到的脉冲持续时间为 43 μ s。脉冲模式不适用于总是锁存模式的 XLDA 和 GDA 位。

图 8. 数据准备就绪信号



4.3.1 DRDY 屏蔽功能

如果将 CTRL4 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，直到传感器滤波器完全稳定下来为止。

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

对于加速度计 UI 链，DRDY 屏蔽功能适用于所有功耗模式、满量程和 ODR（还需考虑运行时间改变）。该屏蔽功能涵盖 HP 或 LPF2 滤波器配置（高达 ODR / 20），并且可以与 CTRL9 寄存器的 XL_FASTSETTL_MODE 位结合使用来管理所有其他滤波器配置。如果 DRDY_MASK 和 XL_FASTSETTL_MODE 位置为 1，所有数据准备就绪信号均被屏蔽，直到内部滤波器稳定为止。

对于陀螺仪 UI 链，DRDY 屏蔽功能适用于所有功耗模式、满量程和 ODR（还需考虑运行时间改变）。

对于陀螺仪 EIS 链，DRDY 屏蔽功能仅适用于驱动电路稳定期间。

DRDY 屏蔽功能不适用于 S4S 数据率同步和 HAODR 模式。

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 引脚的数据准备就绪信号同步，那么强烈建议将 CTRL3 寄存器中的 BDU（块数据更新）位置为 1。

此功能可以避免读取不同采样相关的值（输出数据的最高有效部分和最低有效部分）。特别地，当 BDU 被激活时，每个轴相关的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) 以及 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。

提示

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) 和 OUTZ_H_A(G) & OUTZ_L_A(G) 合并提供，并用二进制补码表示。

加速度数据和角速率数据均用 16 位数字表示。要将它们转换为对应的物理表示，必须应用灵敏度参数。该灵敏度值取决于所选满量程范围（参见数据手册）。具体而言：

- 每个加速度采样必须乘以适当的灵敏度参数 LA_So（线性加速度灵敏度用 mg/LSB 表示），以获得单位为 mg 的相应值。
- 每个角速率采样必须乘以适当的灵敏度参数 G_So（角速率灵敏度用 mdps/LSB 表示），以获得单位为 mdps 的相应值。

4.5.1 输出数据示例

表 24. 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 25. 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。

以下表格中所列值是在理想器件校准的假设下给出的（即，无偏移、无增益误差等）。

表 24. 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 25. 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

角速率值	寄存器地址	
	OUTX_H_G (23h)	OUTX_L_G (22h)
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。

可通过将 CTRL9 寄存器的 USR_OFF_ON_OUT 位置为 1 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各轴加速度值内部减去。内部处理的数据随后被发送到加速度计输出寄存器和 FIFO（如果使能）。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满量程，并可利用 CTRL9 寄存器的 USR_OFF_W 位进行配置：

- 2^{-10} g/LSB（USR_OFF_W 位置为 0 时）
- 2^{-6} g/LSB（USR_OFF_W 位置为 1 时）

4.7 DEN（数据使能）

通过配置 DEN 寄存器中的 LVL1_EN 和 LVL2_EN 位，器件可允许外部触发电平识别。

有两种不同模式可供选择（参见表 26. DEN 配置）：

- 电平感应触发模式
- 电平感应锁存模式

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，DEN 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 CTRL4 寄存器中的 INT2_IN_LH 位置为 1，可以将其更改为高电平有效。

表 26. DEN 配置

LVL1_EN	LVL2_EN	功能	触发类型	动作
0	0	数据使能关闭	-	-
1	0	电平感应触发模式	电平	数据加戳
1	1	电平感应锁存模式	边沿	数据加戳

4.7.1 电平感应触发模式

可以通过将 DEN 寄存器中的 LVL1_EN 位置为 1，并将 DEN 寄存器中的 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 4.7.3 节 用于 DEN 冲压的 LSB 选择）。

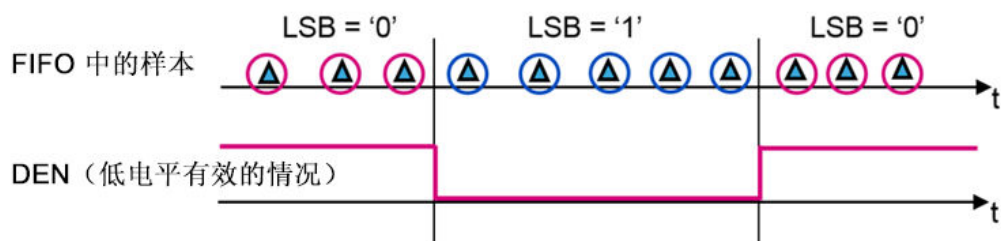
所有数据都可以根据 FIFO 设置存储在 FIFO 中。

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 DEN 寄存器的 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掉电模式。

图 9 以品红色圆圈显示 LSB = 0（DEN 未激活）时存储在 FIFO 中的采样，以蓝色圆圈表示 LSB = 1（DEN 激活）时存储在 FIFO 中的采样。

图 9. 电平感应触发模式，DEN 低电平有效



4.7.2 电平感应锁存模式

可以通过将 DEN 寄存器中的 LVL1_EN 和 LVL2_EN 位置为 1 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。

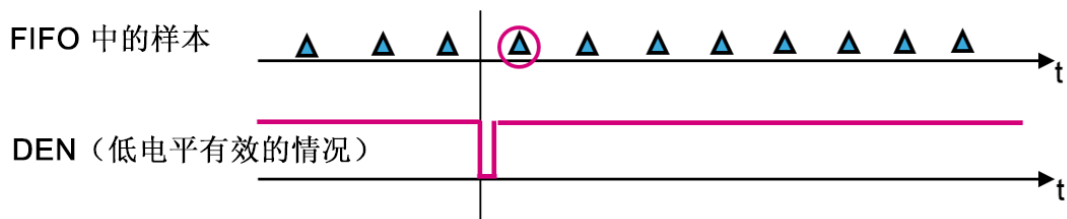
请注意，在更新数据寄存器前会对 DEN 电平进行内部读取：如果在读取后发生电平变化，那么将在下一个 ODR 中确认 DEN。

如果通过 DEN 寄存器的 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掉电模式。

数据可以通过 DEN 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 4.7.3 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，带品红色圆圈的采样在 LSB 位上的值为 1。所有其他采样的 LSB 位都为 0。

图 10. 电平感应锁存模式，DEN 低电平有效



4.7.3 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 输入信号特性有关的信息。根据 DEN 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 和 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

中断生成仅基于加速度计数据，因此，为了生成中断，加速度计传感器必须设置为活动工作模式（不是掉电）。陀螺仪传感器可配置为掉电模式，因为它与中断生成无关。

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。以下功能在硬件中实现：

- 大幅运动检测
- 相对倾斜
- 计步功能
- 时间戳
- 传感器融合功能（游戏旋转向量、重力向量和陀螺仪偏差）

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 **8** 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。此外，可以在机器学习内核逻辑中同步并独立地运行最多 **4** 个决策树。

嵌入式有限状态机和机器学习内核功能提供强大的从头开始或从导入 **ST** 直接提供的活动/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和机器学习内核应用笔记（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 I3C**® 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（默认行为）。作为一种额外的功能（将 **CTRL5** 寄存器的 **INT_EN_I3C** 位置为 **1**），即使使用 **MIPI I3C**® 接口，中断引脚也会激活。

必须使用 **IF_CFG** 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。

IF_CFG 寄存器的 **PP_OD** 位允许将中断引脚逻辑电路从推挽更改为开漏。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。

下面汇总了这些中断控制寄存器的简要说明。寄存器位的默认值等于 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 27. INT1_CTRL 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
0	INT1_CNT_BDR	INT1_FIFO_FULL	INT1_FIFO_OVR	INT1_FIFO_TH	0	INT1_DRDY_G	INT1_DRDY_XL

- INT1_CNT_BDR: INT1 上 FIFO COUNTER_BDR_IA 中断
- INT1_FIFO_FULL: INT1 上 FIFO 全满标志中断
- INT1_FIFO_OVR: INT1 上 FIFO 溢出标志中断
- INT1_FIFO_TH: INT1 上 FIFO 阈值中断
- INT1_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 28. 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 上的传感器集线器端操作中断

表 29. INT2_CTRL 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
INT2_EMB_FUNC_ENDOP	INT2_CNT_BDR	INT2_FIFO_FULL	INT2_FIFO_OVR	INT2_FIFO_TH	INT2_DRDY_G_EIS	INT2_DRDY_G	INT2_DRDY_XL

- INT2_EMB_FUNC_ENDOP: INT2 上嵌入功能操作结束中断。该引脚旨在用于调试目的。因此，建议不要在其他中断信号要路由到 INT2 引脚时使能该引脚。使能 INT2 引脚时，如果未在运行任何嵌入功能，则该引脚设置为高电平；如果正在运行任何嵌入功能，则该引脚设置为低电平。因此，可以使用该引脚来测量嵌入功能的执行时间。
- 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_G_EIS: INT2 上陀螺仪 EIS 数据准备就绪中断
- 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

CTRL4 寄存器的 INT2_DRDY_TEMP 位用于使能 INT2 引脚上的温度数据准备就绪中断。

CTRL7 寄存器的 INT2_DRDY_AH_QVAR 位用于使能 INT2 引脚上的模拟集线器（或 Qvar）数据准备就绪中断。

表 30. 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_TIMESTAMP

- INT2_SLEEP_CHANGE: INT2 上活动/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

如果多个中断信号路由到同一个中断引脚上，则该引脚的逻辑电平为所选中断信号组合的逻辑“或”结果。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）
- EMB_FUNC_STATUS_MAINPAGE/EMB_FUNC_STATUS（用于嵌入功能）
- FSM_STATUS_MAINPAGE/FSM_STATUS（用于有限状态机）
- MLC_STATUS_MAINPAGE/MLC_STATUS（用于机器学习内核）
- STATUS_MASTER_MAINPAGE / STATUS_MASTER（用于传感器集线器）
- FIFO_STATUS2（用于 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、单击、活动/不活动）、嵌入功能和传感器集线器中断状态。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 寄存器的 INT2_on_INT1 引脚能够将一些特定的中断信号进行逻辑“或”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。当该位置为 1 时，可移动中断为 INT2_DRDY_G_EIS 和 INT2_EMB_FUNC_ENDOP（通过 INT2_CTRL 寄存器使能）、INT2_TIMESTAMP（通过 MD2_CFG 寄存器使能）、INT2_DRDY_TEMP（通过 CTRL4 寄存器使能）和 INT2_DRDY_AH_QVAR（通过 CTRL7 寄存器使能）。

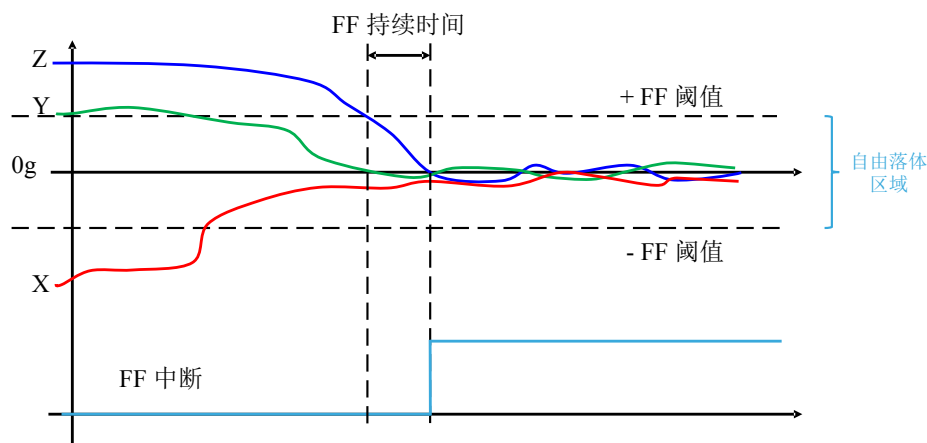
必须通过将 FUNCTIONS_ENABLE 寄存器中的 INTERRUPTS_ENABLE 位置为 1 来使能基本中断。

TAP_CFG0 寄存器的 LIR 位用于使能基本中断功能的锁存中断：当此位置为 1 且中断标记发送到 INT1 引脚和/或 INT2 引脚时，中断保持激活状态，直到 ALL_INT_SRC 寄存器或相应的源寄存器被读取。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。可将 FUNCTIONS_ENABLE 寄存器的 DIS_RST_LIR_ALL_INT 位置为 1，以避免通过读取 ALL_INT_SRC 寄存器来复位锁存的中断信号。该功能可用于使一些状态标志在相应的状态寄存器被读取之前不发生复位。

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 $0g$ 水平，其中所有加速度均足够小以产生中断。自由落体事件检测关联了可配置的阈值和持续时间参数。阈值参数定义了自由落体区幅度；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 11）。

图 11. 自由落体中断



通过将 `FUNCTIONS_ENABLE` 寄存器中的 `INTERRUPTS_ENABLE` 位置为 1，可启用自由落体中断信号，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` 或 `ALL_INT_SRC` 寄存器的 `FF_IA` 位对其进行检查。

如果锁存模式禁用（`TAP_CFG0` 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中断引脚置为有效时，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]` 位域值相关，如表 31. 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 31. 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` 寄存器的 `FF_DUR_[5:0]` 位域的内容，`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 08h 写入 `CTRL1` // 开启加速度计 (ODR = 480 Hz)

2. 将 01h 写入 TAP_CFG0 // 使能锁存模式
3. 将 80h 写入 FUNCTIONS_ENABLE // 使能中断功能
4. 将 00h 写入 WAKE_UP_DUR // 设置事件持续时间 (FF_DUR_5 位)
5. 将 33h 写入 FREE_FALL // 设置 FF 阈值 (FF_THS_[2:0] = 011)
// 设置六个采样事件持续时间 (FF_DUR_[5:0] = 000110)
6. 将 10h 写入 MD1_CFG // FF 中断驱动至 INT1 引脚

示例代码中将阈值设置为 312 mg，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] 位域配置如下：忽略短于 $6/ODR_{XL} = 6/480 \text{ Hz} \approx 12.5 \text{ ms}$ 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 3.5.1 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。如果该位置为 0（默认值），则使用斜率滤波器；如果置为 1，则使用 HPF 高通数字滤波器。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 12. 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] 位定义。这 6 位的 1 LSB 值取决于 INACTIVITY_DUR 寄存器的 WU_INACT_THS_W_[2:0] 位的值，如下表所示。

表 32. 唤醒阈值分辨率

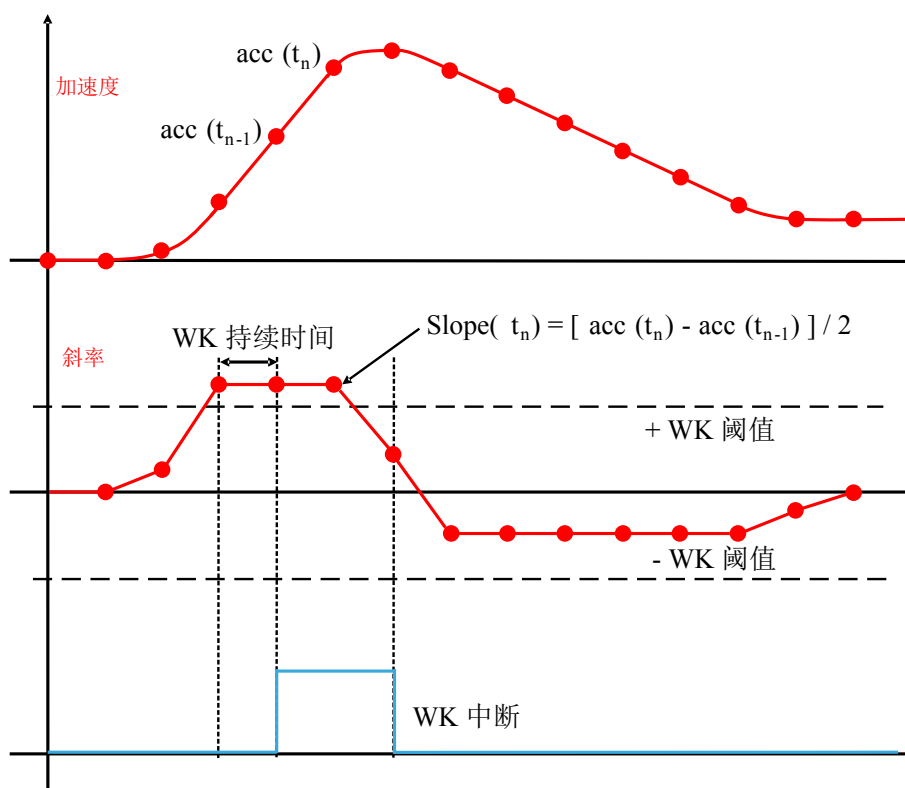
WU_INACT_THS_W_[2:0]	1 LSB 分辨率
000	7.8125 mg
001	15.625 mg
010	31.25 mg
011	62.5 mg
100	125 mg
101	250 mg
110	250 mg
111	250 mg

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要识别的唤醒事件的最短持续时间。其值由 WAKE_UP_DUR 寄存器的 WAKE_DUR[1:0] 位来设置：1 LSB 对应于 $1/ODR_{XL}$ 时间，这里 ODR_{XL} 为加速度计输出数据率。要避免因输入信号寄生尖峰而产生不期望的唤醒中断，适当定义持续时间参数是非常重要的。

通过将 FUNCTIONS_ENABLE 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_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 12. 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中断引脚置为有效时，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。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。

以下给出了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 34h 写入 INACTIVITY_DUR | // 将唤醒阈值分辨率设置为 62.5 mg |
| 2. 将 11h 写入 TAP_CFG0 | // 选择 HPF 路径并使能锁存模式 |
| 3. 将 01h 写入 WAKE_UP_THS | // 设置唤醒阈值 |
| 4. 将 00h 写入 WAKE_UP_DUR | // 将持续时间设置为 0 |
| 5. 将 20h 写入 MD1_CFG | // 唤醒中断驱动至 INT1 引脚 |
| 6. 将 80h 写入 FUNCTIONS_ENABLE | // 使能中断功能 |
| 7. 将 08h 写入 CTRL1 | // 开启加速度计 (ODR = 480 Hz) |

由于持续时间置为 0，因此每个 X、Y 和 Z 滤波数据超出所配置阈值时，会生成唤醒中断信号。WAKE_UP_THS 寄存器的 WK_THS 位域置为 000001 且 1 LSB 分辨率置为 62.5 mg（INACTIVITY_DUR 寄存器的 WU_INACT_THS_W [2:0] 位置为 011），因此唤醒阈值为 62.5 mg。

由于唤醒功能利用斜率/高通数字滤波器实现，因此有必要考虑此功能使能后滤波器的稳定时间。例如，当使用斜率滤波器时（不过对于使用高通数字滤波器要做类似考虑），唤醒功能基于阈值与两次采样（当前 (X, Y, Z) 采样和前次采样）加速度差的一半相比较（参考第 3.5.1 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， $(X, Y, Z) = (0, 0, 1) g$ ）与前次采样 $(X, Y, Z) = (0, 0, 0) g$ （因为尚未生成采样）之差的一半。因此，在 Z 轴上，斜率滤波器的第一个输出值为 $(1 - 0) / 2 = 0.5 g = 500 mg$ ，发生伪中断事件的情况下，该值可能会大于阈值。中断信号保持为高电平，并持续 1 个 ODR，然后变为低电平。

为了避免数字斜率/高通滤波器稳定阶段引起伪中断，可以在此阶段期间将 CTRL9 寄存器的 XL_FASTSETTL_MODE 位和 TAP_CFG0 寄存器的 HW_FUNC_MASK_XL_SETTL 位均配置为 1，以屏蔽基本中断功能的执行触发。

可基于上述唤醒配置步骤简单修改为如下形式：

- | | |
|------------------------------|---|
| 1. 将 20h 写入 CTRL9 | // 将 XL_FASTSETTL_MODE 设置为 1 |
| 2. 将 34h 写入 INACTIVITY_DUR | // 将唤醒阈值分辨率设置为 62.5 mg |
| 3. 将 31h 写入 TAP_CFG0 | // 将 HW_FUNC_MASK_XL_SETTL 设置为 1，选择 HPF 路径并使能锁存模式 |
| 4. 将 01h 写入 WAKE_UP_THS | // 设置唤醒阈值 |
| 5. 将 00h 写入 WAKE_UP_DUR | // 将持续时间设置为 0 |
| 6. 将 20h 写入 MD1_CFG | // 唤醒中断驱动至 INT1 引脚 |
| 7. 将 80h 写入 FUNCTIONS_ENABLE | // 使能中断功能 |
| 8. 将 08h 写入 CTRL1 | // 开启加速度计 (ODR = 480 Hz) |

5.4 6D/4D 方向检测

加速度计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屏幕旋转。

5.4.1 6D 方向检测

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，中断信号会置为有效。只要保持其位置，中断就不会重新置为有效。

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 33. D6D_SRC 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
0	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 置为高电平。
- 当垂直于 Z(Y, X) 轴的面几乎水平时，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) 置为高电平。
- 当垂直于 Z(Y,X) 轴的面几乎水平时，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) 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 34. 4D/6D 功能阈值

SIXD_THS_[1:0]	阈值 [度]
00	80
01	70
10	60
11	50

通过将 TAP_CFG0 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 FUNCTIONS_ENABLE 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 或 ALL_INT_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，然后自动失效（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置为有效时，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 配置影响。在读取 D6D_SRC 寄存器时，这些位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 13. 6D 识别方向中所示的六种可能情形，表 35. 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 13. 6D 识别方向

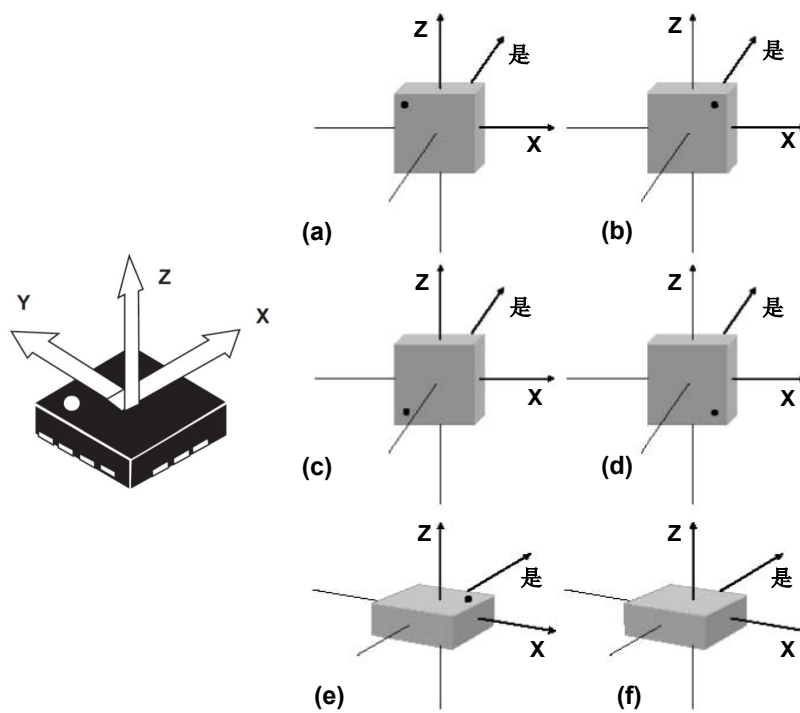


表 35. 6D 定位下的 D6D_SRC 寄存器

情形	D6D_IA	ZH	ZL	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程序如下。

1. 将 41h 写入 TAP_CFG0 // 为 6D 功能和锁存模式使能 LPF2 滤波器
2. 将 40h 写入 TAP_THS_6D // 设置 6D 阈值 (SIXD_THS_[1:0] = 10 = 60 度)
3. 将 04h 写入 MD1_CFG // 6D 中断驱动至 INT1 引脚
4. 将 80h 写入 FUNCTIONS_ENABLE // 使能中断功能
5. 将 08h 写入 CTRL1 // 开启加速度计 (ODR = 480 Hz)

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检测。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 35. 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 上产生中断信号。其高级功能可支持在识别出一个双输入激励（它在两个事件之间具有可编程时间）时产生中断信号，从而使能类似于鼠标按钮的功能。

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。该斜率数据利用以下公式进行计算：

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

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。

单击和双击识别基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 480 Hz。

要使能单击和双击识别功能，必须将 FUNCTIONS_ENABLE 寄存器中的 INTERRUPTS_ENABLE 位置为 1。

5.5.1 单击

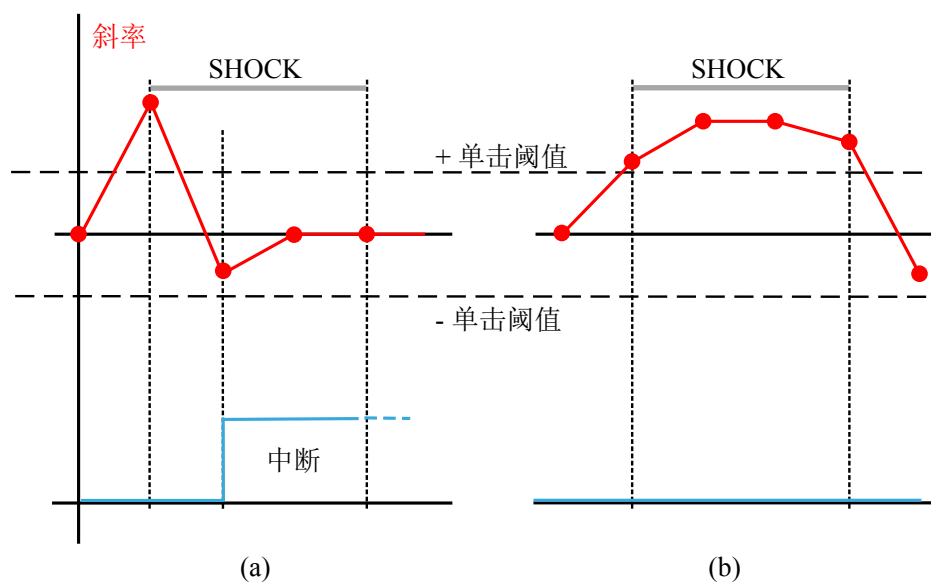
如果器件配置为单击事件检测，那么当所选轴的斜率数据超出了所编程阈值时，会产生一个中断，并在 **shock** 时间窗口内返回到低于阈值。

在单击情况下，如果 **TAP_CFG0** 寄存器的 **LIR** 位置为 0，则中断在 **quiet** 窗口持续时间内保持激活。如果 **LIR** 位置为 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 14. 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 14. 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。

特别地，识别出第一次点击后，第二次点击检测过程会延迟 **quiet** 时间所定义的时间间隔。这意味着，识别出第一次点击后，只有在 **quiet** 窗口之后、**duration** 窗口结束前，斜率数据超过阈值时，才开始第二次点击检测过程。

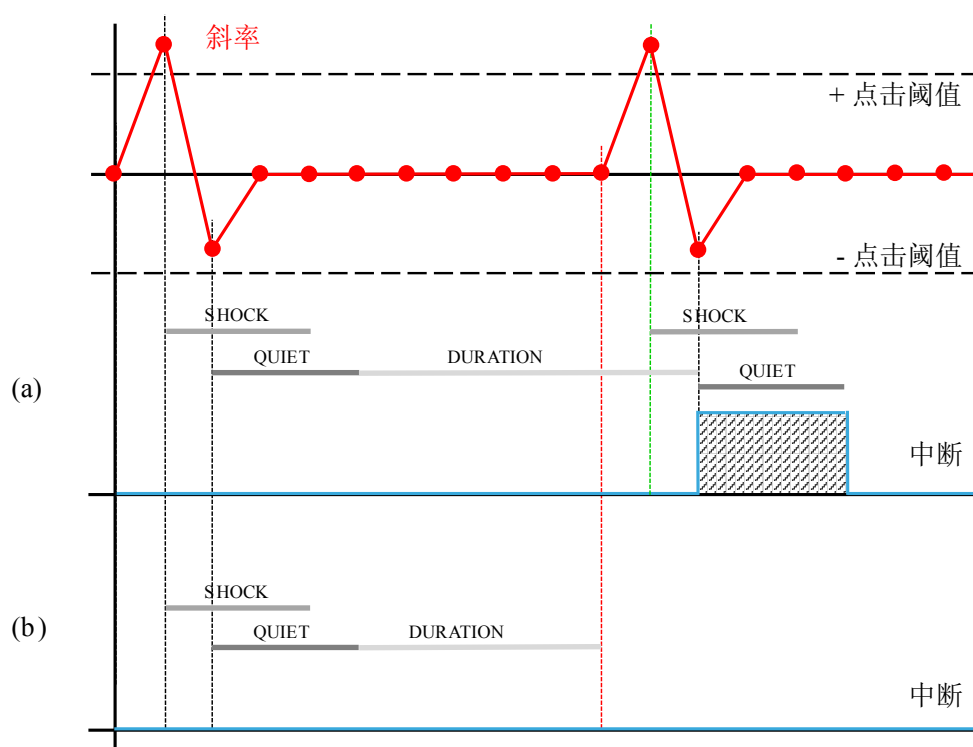
图 15 的情况 (a) 中，正确识别出了双击事件，而在情况 (b) 中，由于在经过窗口间隔之后斜率数据超出了阈值，因此未产生中断。

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** 位置为 0，则中断在 **quiet** 窗口持续时间内保持激活。如果 **LIR** 位置为 1，则中断保持激活直至 **TAP_SRC** 或 **ALL_INT_SRC** 寄存器被读取。

图 15. 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，FUNCTIONS_ENABLE 寄存器的 INTERRUPTS_ENABLE 位必须置为 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 时间窗口。

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 位的 1 LSB 值取决于所选加速度计满量程：1 LSB = $FS_{XL} / 2^5$ 。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 36. 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

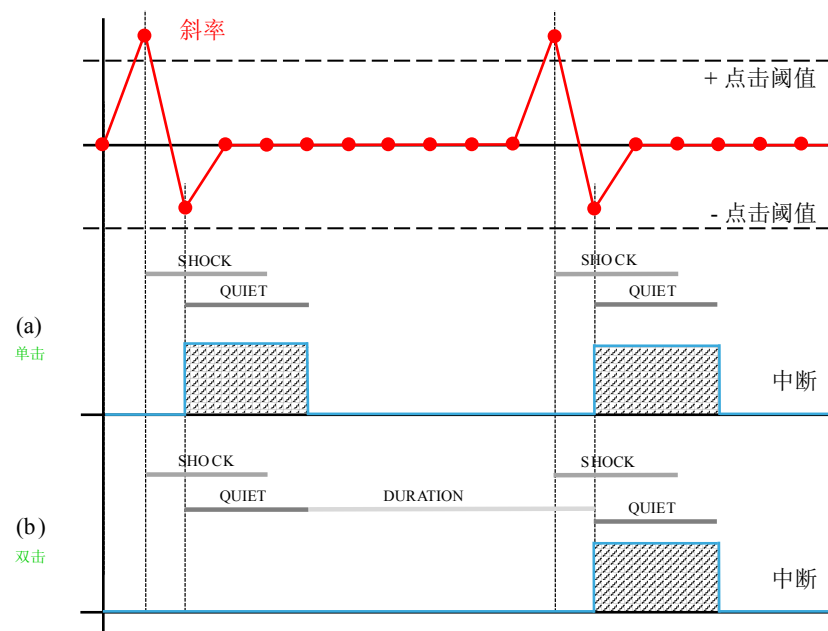
shock 时间窗口定义了超阈值事件的最大持续时间：在 shock 窗口结束前，加速度必须返回到低于阈值之下，否则不能检测到该点击事件。TAP_DUR 寄存器的 SHOCK[1:0] 位用来设置 shock 时间窗口值：这几个位的默认值为 00，对应于 $4/ODR_{XL}$ 的时间，这里 ODR_{XL} 为加速度计输出数据率。如果 SHOCK[1:0] 位置为其他不同的值，那么 1 LSB 对应于 $8/ODR_{XL}$ 的时间。

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事件。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）。TAP_DUR 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 00，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。TAP_DUR 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 0000，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] 位置为其他不同的值，那么 1 LSB 对应于 $32/ODR_{XL}$ 的时间。

图 16. 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。

图 16. 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 37. 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。当它置为 0 时，单击中断信号可应用于锁存模式；当它置为 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。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 02h 写入 TAP_CFG0	// 使能 Z 轴上的点击检测
2.	将 00h 写入 TAP_CFG1	// 设置 X 轴阈值和轴优先级
3.	将 00h 写入 TAP_CFG2	// 设置 Y 轴阈值
4.	将 02h 写入 TAP_THS_6D	// 设置 Z 轴阈值
5.	将 06h 写入 TAP_DUR	// 设置 quiet 和 shock 时间窗口
6.	将 00h 写入 WAKE_UP_THS	// 只使能单击 (SINGLE_DOUBLE_TAP = 0)
7.	将 80h 写入 FUNCTIONS_ENABLE	// 使能硬件功能
8.	将 40h 写入 MD1_CFG	// 单击中断驱动至 INT1 引脚
9.	将 02h 写入 CTRL8	// FS_XL = ± 8 g
10.	将 08h 写入 CTRL1	// 开启加速度计 (480 Hz)

在本例中，TAP_THS_Z[4:0] 位置为 00010，因此 Z 轴的点击阈值为 500 mg ($= 2 * FS_XL / 2^5$)。

TAP_DUR 寄存器的 SHOCK 位域置为 10。当斜率数据超出所编程阈值时，产生中断，并在 33.3 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。

TAP_DUR 寄存器的 QUIET 位域置为 01。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，即 8.3 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 02h 写入 TAP_CFG0	// 使能 Z 轴上的点击检测
2.	将 00h 写入 TAP_CFG1	// 设置 X 轴阈值和轴优先级
3.	将 00h 写入 TAP_CFG2	// 设置 Y 轴阈值
4.	将 03h 写入 TAP_THS_6D	// 设置 Z 轴阈值
5.	将 7Fh 写入 TAP_DUR	// 设置 quiet 和 shock 时间窗口
6.	将 80h 写入 WAKE_UP_THS	// 使能单击和双击 (SINGLE_DOUBLE_TAP = 1)
7.	将 80h 写入 FUNCTIONS_ENABLE	// 使能硬件功能
8.	将 08h 写入 MD1_CFG	// 单击中断驱动至 INT1 引脚
9.	将 02h 写入 CTRL8	// FS_XL = ± 8 g
10.	将 08h 写入 CTRL1	// 开启加速度计 (480 Hz)

在本例中，TAP_THS_Z[4:0] 位置为 00011，因此点击阈值为 750 mg ($3 * FS_XL / 2^5$)。

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。TAP_DUR 寄存器的 SHOCK 位域置为 11，因此 shock 时间为 50 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。TAP_DUR 寄存器的 QUIET 位域置为 11，因此 quiet 时间为 25 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，TAP_DUR 寄存器的 DUR 位域置为 0111，因此 duration 时间为 533.3 ms ($= 8 * 32 / ODR_XL$)。

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，如果检测到不活动状态，器件能够自动将加速度计功耗模式切换为低功耗模式 1 并将采样率更改为可配置的低 ODR（提供的可选 ODR 为 1.875 Hz、15 Hz、30 Hz 和 60 Hz）；如果检测到活动状态，器件能够自动切换回通过 CTRL1 寄存器的 OP_MODE_XL[2:0] 位和 ODR_XL[3:0] 位选择的功耗模式和采样率。

不活动状态下的目标加速度计 ODR 可通过 INACTIVITY_DUR 寄存器的 XL_INACT_ODR[1:0] 位选择，具体的值如下表所示。

表 38. 不活动事件时的目标加速度计 ODR 配置

XL_INACT_ODR[1:0]	ODR [Hz]
00	1.875
01	15
10	30
11	60

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。
- 陀螺仪进入掉电模式。

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

通过将 INTERRUPTS_ENABLE 位置为 1，并配置 FUNCTIONS_ENABLE 寄存器的 INACT_EN[1:0] 位，可使能活动/不活动识别功能。如果 FUNCTIONS_ENABLE 寄存器的 INACT_EN[1:0] 位等于 00，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 39. 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	加速度计 ODR 通过 XL_INACT_ODR[1:0] 位设置（低功耗模式 1）	陀螺仪配置不变
10	加速度计 ODR 通过 XL_INACT_ODR[1:0] 位设置（低功耗模式 1）	陀螺仪进入睡眠模式
11	加速度计 ODR 通过 XL_INACT_ODR[1:0] 位设置（低功耗模式 1）	陀螺仪进入掉电模式

可以利用斜率滤波器（更多详细信息见第 3.5.1 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。如果该位置为 0（默认值），则使用斜率滤波器；如果该位置为 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 17. 活动/不活动识别（利用斜率滤波器））。

该无符号阈值使用 INACTIVITY_THS 寄存器的 INACT_THS[5:0] 位定义。这 6 位的 1 LSB 值取决于 INACTIVITY_DUR 寄存器的 WU_INACT_THS_W[2:0] 位的值，如下表所示。

表 40. 活动/不活动阈值分辨率

WU_INACT_THS_W_[2:0]	1 LSB 分辨率
000	7.8125 mg
001	15.625 mg
010	31.25 mg
111	62.5 mg
100	125 mg
101	250 mg
110	250 mg
111	250 mg

该阈值可适用于正负滤波数据。

当一定数量的连续 X、Y 和 Z 滤波数据小于所配置阈值时，忽略 CTRL1 寄存器的 OP_MODE_XL_[2:0] 和 ODR_XL_[3:0] 位（不活动），加速度计被内部设置为低功耗模式 1 且采样率通过 INACTIVITY_DUR 寄存器的 XL_INACT_ODR_[1:0] 位配置，尽管 CTRL1 寄存器的内容保持不变。陀螺仪的特性随着 FUNCTIONS_ENABLE 寄存器的 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] 位置为 0000，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，中断置为高电平并持续 $1 / \text{ODR_XL}[s]$ 的时间，然后自动失效。

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 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 INACTIVITY_DUR 寄存器的 INACT_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，中断置为高电平并持续 $1 / \text{ODR_XL}[s]$ 的时间，然后自动失效。

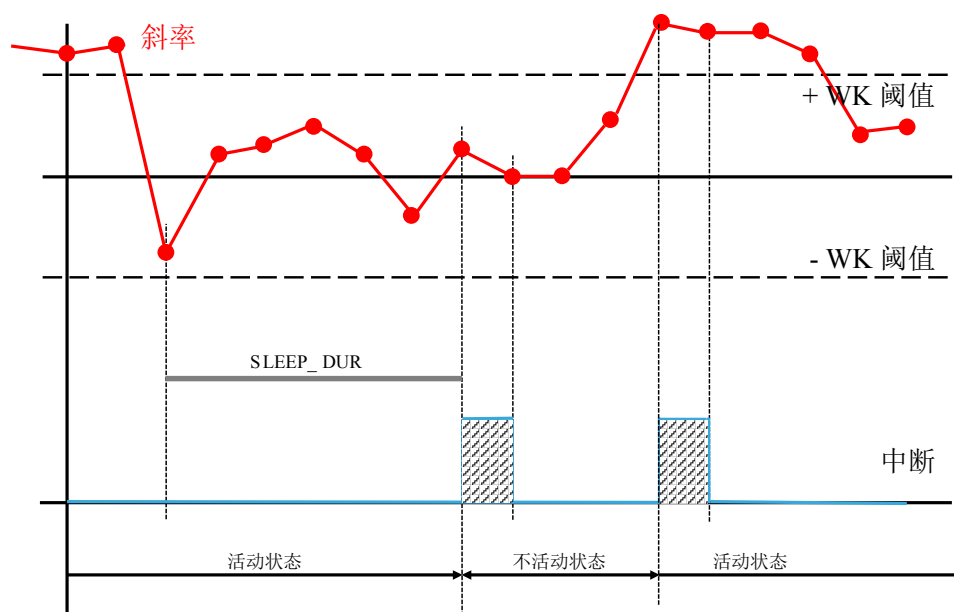
当使能活动/不活动检测功能时，通过将 MD1_CFG 寄存器的 INT1_SLEEP_CHANGE 位或 MD2_CFG 寄存器的 INT2_SLEEP_CHANGE 位置为 1，可将活动/不活动事件驱动至两个中断引脚。还可通过读取 WAKE_UP_SRC 或 ALL_INT_SRC 寄存器的 SLEEP_CHANGE_IA 位来检查活动/不活动事件。

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，该位对应于器件的当前状态。

通过将 INACTIVITY_DUR 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 17. 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程序如下所示：

- | | |
|------------------------------|-------------------------------|
| 1. 将 83h 写入 FUNCTIONS_ENABLE | // 使能中断功能 |
| | // 处于不活动状态时，将陀螺仪设置为掉电模式 |
| 2. 将 34h 写入 INACTIVITY_DUR | // 将阈值分辨率设置为 62.5 mg |
| | // 将加速度计不活动 ODR 设置为 15 Hz |
| | // 设置活动持续时间 |
| 3. 将 01h 写入 INACTIVITY_THS | // 将阈值设置为 000001 |
| 4. 将 05h 写入 WAKE_UP_DUR | // 将睡眠持续时间设置为 0101 |
| 5. 将 80h 写入 MD1_CFG | // 活动/不活动中断驱动至 INT1 引脚 |
| 6. 将 04h 写入 CTRL6 | // 将陀螺仪 FS 设置为 ± 2000 dps |
| 7. 将 02h 写入 CTRL8 | // 将加速度计 FS 设置为 ± 8 g |
| 8. 将 08h 写入 CTRL1 | // 开启加速度计 (ODR = 480 Hz) |
| 9. 将 08h 写入 CTRL2 | // 开启陀螺仪 (ODR = 480 Hz) |

在本例中，INACTIVITY_THS 寄存器的 INACT_THS [5:0] 位域置为 000001 且 1 LSB 分辨率为 62.5 mg (INACTIVITY_DUR 寄存器的 WU_INACT_THS_W [2:0] 位置为 011)，因此活动/不活动阈值为 62.5 mg。

进行不活动检测前，X、Y 和 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 位域定义：该位域置为 0101，对应 5.33 s ($= 5 * 512 / ODR_{XL}$)。经过这段时间之后，加速度计 ODR 内部设置为 15 Hz (XL_INACT_ODR [1:0] = 01)，陀螺仪内部设置为掉电模式。

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 寄存器设置和开启陀螺仪，因为 INACTIVITY_DUR 寄存器的 INACT_DUR [1:0] 位被配置为 00。

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 FUNCTIONS_ENABLE 寄存器的 INACT_EN [1:0] 位置为 00 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掉电模式。启动时间内，寄存器不可访问。

提示

如果需要通过 Vdd 断电再上电来强制启动，并且 Vdd 断电与再上电之间相隔不到 20 ms，则最大启动时间增至 30 ms。

上电后，可通过将 CTRL3 寄存器的 BOOT 位置为 1 来重载修整参数。在这种情况下，必须等待 30 ms 才能完成内部重启程序。CTRL3 寄存器的 BOOT 位自动返回 0。

如果需要复位至控制寄存器的默认值，可通过将 CTRL3 寄存器的 SW_RESET 位置为 1 来执行。当此位置为 1 时，以下寄存器复位至其默认值：

- FUNC_CFG_ACCESS (01h)
- S4S_TPH_L (04h) 到 ALL_INT_SRC (1Dh)
- TIMESTAMP0 (40h) 到 TIMESTAMP3 (43h)
- WAKE_UP_SRC (45h) 到 D6D_SRC (47h)
- FUNCTIONS_ENABLE (50h) 到 UI_HANDSHAKE_CTRL (64h)
- CTRL_EIS (6Bh) 到 Z_OFS_USR (75h)
- FIFO_DATA_OUT_TAG (78h)

软件复位程序最多耗时 150 μ s。复位状态通过 CTRL3 寄存器的 SW_RESET 位的状态指示。当复位完成时，该位自动置为 0。

重启流程如下：

1. 将加速度计和陀螺仪设置为掉电模式。
2. 将 CTRL3 寄存器的 BOOT 位置为 1。

3. 等待 30 ms。

软件复位流程如下：

1. 将加速度计和陀螺仪设置为掉电模式。
2. 将 CTRL3 寄存器的 SW_RESET 位置为 1。
3. 监视软件复位状态。存在两种可能：
 - a. 等待 150 μ s。
 - b. 轮询 CTRL3 寄存器的 SW_RESET 位，直至其自动返回 0。

为了避免冲突，重启和软件复位不能同时执行（不要同时将 CTRL3 寄存器的 BOOT 位和 SW_RESET 位置为 1）。上述流程必须顺序执行。

如果需要完整复位（包括启动、软件复位以及嵌入功能和内部滤波器复位），可通过将 FUNC_CFG_ACCESS 寄存器的 SW_POR 位置为 1 来执行。当该位置为 1 时，器件将触发自身完整复位，类似于上电复位。在这种情况下，必须等待 30 ms 才能完成器件复位。FUNC_CFG_ACCESS 寄存器的 SW_POR 位自动返回 0。完整复位流程如下：

1. 将 FUNC_CFG_ACCESS 寄存器的 SW_POR 位置为 1。
2. 等待 30 ms。

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（步伐检测和步数计算）
- 大幅运动检测
- 相对倾斜
- 时间戳
- 传感器融合功能（游戏旋转向量、重力向量和陀螺仪偏差）

6.1 计步功能：步伐检测和步数计算

专用 IP 模块专门实现计步功能：步伐检测和步数计算。

计步功能的工作频率为 30 Hz 且只基于加速度计传感器。因此，使用计步功能时，必须将加速度计 ODR 设置为 30 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掉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 来将步数复位至 0。计数器复位后，PEDO_RST_STEP 位自动重置回 0。

步伐检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，以避免出现错误步伐检测（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。该位域的 1 LSB 对应于 1 步，默认值为 10 步。器件不活动约 1 秒后，会重新启动去抖动功能。

可以使能一个额外的假阳性拒绝 (FPR) 块，以便根据统计数据来实时识别行走活动（包括跑步）以及在未检测到行走活动时禁用计步器。具体可以按照以下操作来激活：

- 将 PEDO_CMD_REG 嵌入高级功能寄存器的 FP_REJECTION_EN 位置为 1。
- 将 EMB_FUNC_EN_B 寄存器的 MLC_EN 位或 EMB_FUNC_EN_A 寄存器的 MLC_BEFORE_FSM_EN 位置为 1。

在 LSM6DSV16X 器件中，用户可以定制 FPR 块。在这种情况下，必须对 MLC 进行编程才能使用第一个决策树来识别两个类别：无行走（代码为 0x04 的类别）和行走（代码为 0x08 的类别）。具体来说，如果 MLC 检测到以下类别组，将禁用计步器：

- 代码为 0x4 到 0x7 的类别
- 代码为 0xC 到 0xE 的类别

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 41. 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 FUNCTIONS_ENABLE 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 5.6 ms。当在设定时间段内计数到至少一步（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。
- **STEP_COUNTER_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 PEDO_SC_DELTAT 位域（PEDO_SC_DELTAT_H 和 PEDO_SC_DELTAT_L 嵌入高级功能寄存器联合）以及 PEDO_CMD_REG 嵌入高级功能寄存器的 CARRY_COUNT_EN 位的值。

表 42. IS_STEP_DET 配置

PEDO_SC_DELTAT	CARRY_COUNT_EN	IS_STEP_DET
PEDO_SC_DELTAT = 0	0	STEP_COUNTER_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，还必须通过将 MD1_CFG/MD2_CFG 寄存器 INT1_EMB_FUNC/INT2_EMB_FUNC 位置为 1 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 $1 / \text{MAX_RATE}$ 秒，其中 MAX_RATE 表示已使能的嵌入功能的最大速率。如果只使能计步功能，则脉冲持续时间等于 1 / 30 秒。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) 缓冲器）。

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 // 使能带计步器的假阳性拒绝功能块链路 (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 // 使能假阳性拒绝功能块 (MLC_EN = 1)
9. 将 08h 写入 EMB_FUNC_INT1 // 步伐检测中断驱动至 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 // 使能嵌入功能中断的路由
12. 将 02h 写入 CTRL8 // FS_XL = ±8 g
13. 将 04h 写入 CTRL1 // 开启加速度计 (ODR_XL = 30 Hz)

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 30 Hz，因此加速度计 ODR 必须设置为 30 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。

提示

注：大幅运动检测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，可将大幅运动中中断信号驱动至 INT1/INT2 中断引脚。这种情况下，还必须通过将 MD1_CFG/MD2_CFG 寄存器 INT1_EMB_FUNC/INT2_EMB_FUNC 位置为 1 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中断信号进行检查。

大幅运动中中断信号的行为默认为脉冲。脉冲持续时间等于 $1 / \text{MAX_RATE}$ 秒，其中 MAX_RATE 表示已使能的嵌入功能的最大速率。如果只使能大幅运动检测功能，则脉冲持续时间等于 1 / 30 秒。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_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 | // 使能嵌入功能中断的路由 |
| 7. 将 04h 写入 CTRL1 | // 开启加速度计 |
| | // ODR_XL = 30 Hz |

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 30 Hz，因此加速度计 ODR 必须设置为 30 Hz 或更高的值。

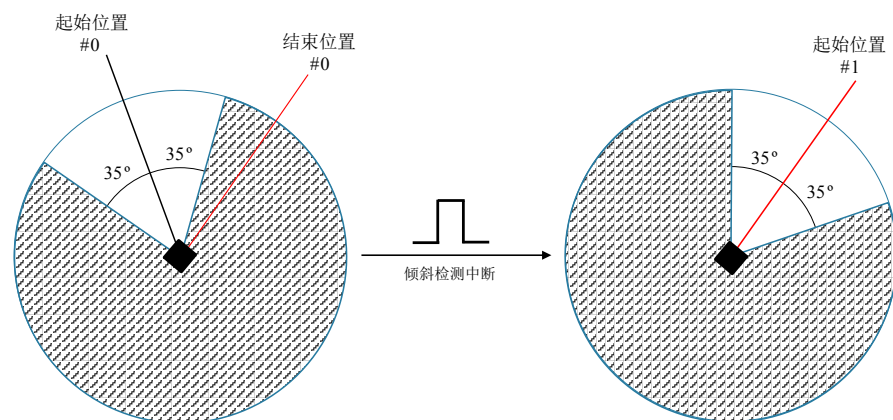
为了使能相对倾斜度检测功能，必须将 EMB_FUNC_EN_A 嵌入功能寄存器的 TILT_EN 位置为 1。通过将 EMB_FUNC_INIT_A 嵌入功能寄存器的 TILT_INIT 位置位，可以重新初始化算法。

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 18. 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 18. 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，还必须通过将 MD1_CFG/MD2_CFG 寄存器 INT1_EMB_FUNC/INT2_EMB_FUNC 位置为 1 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 $1 / \text{MAX_RATE}$ 秒，其中 MAX_RATE 表示已使能的嵌入功能的最大速率。如果只使能倾斜度检测功能，则脉冲持续时间等于 1 / 30 秒。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 | // 使能嵌入功能中断的路由 |
| 7. 将 04h 写入 CTRL1 | // 开启加速度计 |
| | // ODR_XL = 30 Hz |

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 `FUNCTIONS_ENABLE` 寄存器的 `TIMESTAMP_EN` 位置为 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 21.75 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。

$$t_{\text{actual}}[\text{s}] = \frac{1}{46080 \cdot (1 + 0.0013 \cdot \text{FREQ_FINE})}$$

同样地，可以使用以下公式获取实际输出数据率：

$$\text{ODR}_{\text{actual}}[\text{Hz}] = \frac{7680 \cdot (1 + 0.0013 \cdot \text{FREQ_FINE})}{\text{ODR}_{\text{coeff}}}$$

其中的 $\text{ODR}_{\text{coeff}}$ 值如下表所示。

表 43. $\text{ODR}_{\text{coeff}}$ 值

选择的 ODR [Hz]	$\text{ODR}_{\text{coeff}}$
7.5	1024
15	512
30	256
60	128
120	64
240	32
480	16
960	8
1920	4
3840	2
7680	1

如果加速度计和陀螺仪均处于掉电模式，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 26 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 5.6 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（65 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) 缓冲器）。

6.5 传感器融合功能

专用传感器融合块 SFLP（传感器融合低功耗）旨在基于加速度计和陀螺仪数据处理来生成以下数据：

- 游戏旋转向量，提供表示设备姿态的四元数
- 重力向量，提供表示重力方向的三维向量
- 陀螺仪偏差，提供表示陀螺仪偏差的三维向量

通过将 `EMB_FUNC_EN_A` 嵌入功能寄存器的 `SFLP_GAME_EN` 位置为 1 来使能 SFLP 块。

可以通过将 `EMB_FUNC_INIT_A` 嵌入功能寄存器的 `SFLP_GAME_INIT` 位置为 1 来重新初始化 SFLP 块。

SFLP 块以可配置的输出数据率（必须等于或小于加速度计和陀螺仪的选定输出数据率）工作，该数据率可通过 `SFLP_ODR` 嵌入功能寄存器的 `SFLP_GAME_ODR_[2:0]` 位域配置为以下值：

- 000: 15 Hz
- 001: 30 Hz
- 010: 60 Hz

- 011: 120 Hz (默认值)
- 100: 240 Hz
- 101: 480 Hz

SFLP 生成的数据只能从 FIFO 读取, 详细信息请参见第 9 节 先进先出 (FIFO) 缓冲器。

6.5.1 陀螺仪偏差初始值设置

SFLP 嵌入了一个陀螺仪偏差校准程序, 该程序在器件稳定时自动执行。在某些应用中, 无法保证陀螺仪偏差校准所需的稳定条件, 此时需要按照特定流程来设置先前在 SFLP 块中计算得出的偏差。该流程将强制复位 SFLP 算法, 必须按如下步骤实现:

1. 将 HFP 格式的 **gbias** 转换为 [rad/s], 然后根据表 44 将其除以相应的 **k** 系数。
2. 保存当前传感器配置 (CTRL1 和 CTRL2 寄存器) 并设置高性能模式 (如果加速度计和陀螺仪均处于掉电模式, 则开启加速度计并等待第一个有效采样)。
3. [可选] 如果传感器集线器使能, 则关闭 I²C 主器件, 如第 7.2.1 节 MASTER_CONFIG (14h) 所述。
4. 禁用嵌入功能 (保存 EMB_FUNC_EN_A 和 EMB_FUNC_EN_B 寄存器的当前值并将其设置为 00h)。
5. 等待 EMB_FUNC_ENDOP = 1。
6. 将 CTRL10 寄存器的 EMB_FUNC_DEBUG 位置为 1。
7. 将 EMB_FUNC_EN_A 寄存器的 SFLP_GAME_EN 位置为 1。
8. 将加速度计数据写入传感器集线器寄存器 SENSOR_HUB_1 到 SENSOR_HUB_9。每个轴必须以 24 位有符号数 (二进制补码) 形式写入传感器集线器寄存器, 按当前加速度计满量程设置左移 (FS_XL = 00, 不左移; FS_XL = 01, 左移一位; FS_XL = 10, 左移两位; FS_XL = 11, 左移三位)。传感器集线器寄存器 SENSOR_HUB_10 到 SENSOR_HUB_18 必须设置为 00h。
9. 等待 30 μ s, 然后等待至 EMB_FUNC_ENDOP = 1。
10. 将在步骤 1 中计算出的 **gbias** 值写入嵌入高级特征 page 0 寄存器 SFLP_GAME_GBIASX_L 到 SFLP_GAME_GBIASZ_H。
11. 重载在步骤 2 和步骤 4 中保存的传感器配置 (不要将 EMB_FUNC_EN_A 寄存器的 SFLP_GAME_EN 位再次置为 0)。
12. 将 CTRL10 寄存器的 EMB_FUNC_DEBUG 位置为 0。
13. [可选] 如果传感器集线器使能, 开启 I²C 主器件。

表 44. k 系数

SFLP 游戏 ODR [Hz]	k 系数
15	0.04
30	0.02
60	0.01
120	0.005
240	0.0025
480	0.00125

6.6 嵌入功能额外配置和监视

器件可以根据需要通过 `EMB_FUNC_CFG` 寄存器使能一些额外的配置。

具体以下三项额外的功能：

- 通过将 `EMB_FUNC_IRQ_MASK_XL_SETTL` 位置为 1，可以使能在加速度计数据处于稳定阶段时屏蔽嵌入功能的执行触发，从而避免在稳定阶段处理加速度计数据。
- 通过将 `EMB_FUNC_IRQ_MASK_G_SETTL` 位置为 1，可以使能在陀螺仪数据处于稳定阶段时屏蔽嵌入功能的执行触发，从而避免在稳定阶段处理陀螺仪数据。
- 通过将 `EMB_FUNC_DISABLE` 位置为 1，可以停止嵌入功能的执行触发。当该位再次置为 0 时，将强制执行所有初始化程序并再次使能执行触发。

器件可以通过 `EMB_FUNC_EXEC_STATUS` 嵌入功能寄存器监视嵌入功能的执行。

具体包含以下信息：

- 执行超时：该信息位于 `EMB_FUNC_EXEC_OVR` 位中。如果已使能嵌入功能的执行时间超过最大时间（即，在嵌入功能执行结束之前，生成了一组要用作输入的新的传感器数据），则该位置为有效。
- 正在执行：该信息位于 `EMB_FUNC_ENDOP` 位中。当该位置为 1 时，嵌入功能未在运行；当该位置为 0 时，嵌入功能正在运行。通过将 `INT2_CTRL` 寄存器的 `INT2_EMB_FUNC_ENDOP` 位置为 1，可以将该信息路由到 `INT2` 引脚。

7 模式 2——传感器集线器 (sensor hub) 模式

LSM6DSV16X 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，如增加传感器集线器 (sensor hub)。使能传感器集线器模式（模式 2）时，用来连接外部传感器的主 I²C/MIPI I3C[®]/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 传感器集线器 (sensor hub) 模式说明

在传感器集线器 (sensor hub) 模式（模式 2）下，最多可有 4 个外部传感器连接至器件的 I²C 主接口。传感器集线器触发信号可以与加速度计/陀螺仪数据就绪信号（最高 480 Hz）同步。使用这种配置时，可通过 SLV0_CONFIG 寄存器的 SHUB_ODR_[2:0] 位配置传感器集线器 ODR。或者，可以将连接到 INT2 引脚的外部信号用作传感器集线器触发器。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。

在传感器集线器触发信号上，通过寄存器 SLVx_ADD、SLVx_SUBADD、SLVx_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] 位域所使能的外部传感器）。

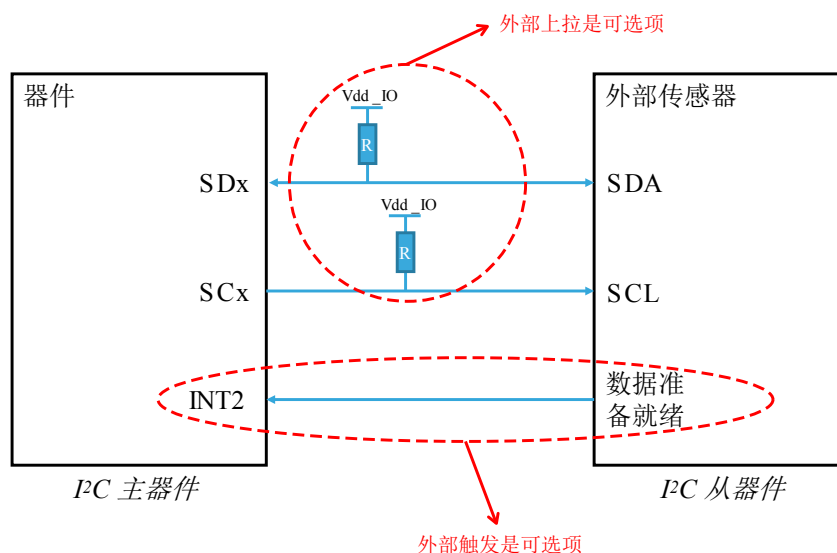
外部传感器数据也可存储在 FIFO 中（详细信息见第 9 节 先进先出 (FIFO) 缓冲器）。

如果加速度计和陀螺仪均处于掉电模式，则传感器集线器 (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 19. 模式 2 下的外部传感器连接所示。外部上拉电阻和外部触发信号连接是可选的，取决于寄存器配置。

IF_CFG 寄存器的 SHUB_PU_EN 位可用于使能或禁用 I²C 总线上的内部上拉。当此位置为 0 时，内部上拉禁用，SDx/SCx 引脚上需要外部上拉电阻，如图 19. 模式 2 下的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。

图 19. 模式 2 下的外部传感器连接



7.2 传感器集线器 (sensor hub) 模式寄存器

当 `FUNC_CFG_ACCESS` 寄存器的 `SHUB_REG_ACCESS` 位置为 1 时，可以访问传感器集线器配置寄存器和输出寄存器。在 `SHUB_REG_ACCESS` 位置为 1 后，只有传感器集线器寄存器可供使用。为了保证其他操作的正确寄存器映射，在传感器集线器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` 和 `SLVx_CONFIG` 专门用来配置与 4 个可连接外部传感器相关的 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 主接口。

表 45. MASTER_CONFIG 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
RST_MASTER_REGS	WRITE_ONCE	START_CONFIG	PASS_THROUGH_MODE	0	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。

提示

如果 *slave 0* 用于读取事务，则 `WRITE_ONCE` 位必须置为 1。

- `START_CONFIG` 位对传感器集线器触发信号进行选择。
 - 当此位置为 0 时，加速度计/陀螺仪传感器必须激活（不能处于掉电模式），传感器集线器触发信号为加速度计/陀螺仪数据准备就绪信号，其频率由 `SLV0_CONFIG` 寄存器的 `SHUB_ODR[2:0]` 位定义（最高 480 Hz）。
 - 当此位置为 1 时，加速度计和陀螺仪中必须激活至少一个传感器，并且传感器集线器触发信号路由至 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 19. 模式 2 下的外部传感器连接），以触发对外部传感器寄存器的读取/写入操作。INT2 极性引起的传感器集线器中断可通过 `CTRL4` 寄存器的 `INT2_IN_LH` 位选择：如果该位置为 0，则引脚为低电平有效，否则为高电平有效。

提示

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 19. 模式 2 下的外部传感器连接），并且后者处于掉电模式，那么外部传感器无法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。当该位置为 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 传感器集线器 (sensor hub) 直通功能。
- `MASTER_ON` 位必须置为 1，以使能器件的辅助 I²C 主线（传感器集线器 (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掉电模式时或在应用软件复位程序时修改传感器集线器配置，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. 修改传感器集线器寄存器的配置或将加速度计/陀螺仪设置为掉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：
 - 00：一个从器件
 - 01：两个从器件
 - 10：三个从器件
 - 11：四个从器件

7.2.2 STATUS_MASTER (22h)

类似于其他传感器集线器配置和输出寄存器，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线器寄存器的访问。

表 46. 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 位会置为 1。
- SENS_HUB_ENDOP 位报告 I²C 主事务结束。当事务结束时，该位置为 1；当读取 STATUS_MASTER/STATUS_MASTER_MAINPAGE 寄存器时，该位复位为 0。

当传感器集线器程序完成时，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 SLV_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到 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) 和 SLV0_CONFIG (17h)

传感器集线器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 47. 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线器 (sensor hub) 触发事件时，执行读取/写入操作。

表 48. 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 位置为 1）。

表 49. SLV0_CONFIG 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
SHUB_ODR_2	SHUB_ODR_1	SHUB_ODR_0	0	BATCH_EXT_SENS_0_EN	Slave0_numop2	Slave0_numop1	Slave0_numop0

- SHUB_ODR_[2:0] 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）时的传感器集线器输出数据率。可以将传感器集线器输出数据率配置为六个可能值（受加速度计和陀螺仪传感器 ODR 的限制）：
 - 000: 1.875 Hz
 - 001: 15 Hz
 - 010: 30 Hz
 - 011: 60 Hz
 - 100: 120 Hz（默认值）
 - 101: 240 Hz
 - 110: 480 Hz

SHUB_ODR_[2:0] 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 FIFO 中与 slave0 相关的外部传感器数据的批处理。
- 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) 和 SLV1_CONFIG (1Ah)

传感器集线器寄存器用于配置关联到第二外部传感器的 I²C 从接口，下面对此进行介绍。

表 50. 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线器 (sensor hub) 触发事件时，执行读取操作。

表 51. 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 位置为 1 时，要读取的第二个外部传感器寄存器的地址。

表 52. SLV1_CONFIG 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	1	BATCH_EXT_SENS_1_EN	Slave1_numop2	Slave1_numop1	Slave1_numop0

- BATCH_EXT_SENS_1_EN 位用于使能 FIFO 中与 slave1 相关的外部传感器数据的批处理。

Slave1_numop[2:0] 位专门用来定义从 SLV1_SUBADD 寄存器所示的寄存器地址开始的第二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) 和 SLV2_CONFIG (1Dh)

传感器集线器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 53. 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线器 (sensor hub) 触发事件时，执行读取操作。

表 54. 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

- Slave2_reg[7:0] 位用来表示，当 SLV2_ADD 寄存器的 r_2 位置为 1 时，要读取的第三个外部传感器寄存器的地址。

表 55. SLV2_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 FIFO 中与 slave2 相关的外部传感器数据的批处理。
- 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) 和 SLV3_CONFIG (20h)

传感器集线器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 56. 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线器 (sensor hub) 触发事件时，执行读取操作。

表 57. 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 位置为 1 时，要读取的第四个外部传感器寄存器的地址。

表 58. SLV3_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 FIFO 中与 slave3 相关的外部传感器数据的批处理。

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 (21h)

表 59. 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) 位域的值，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 20. 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。

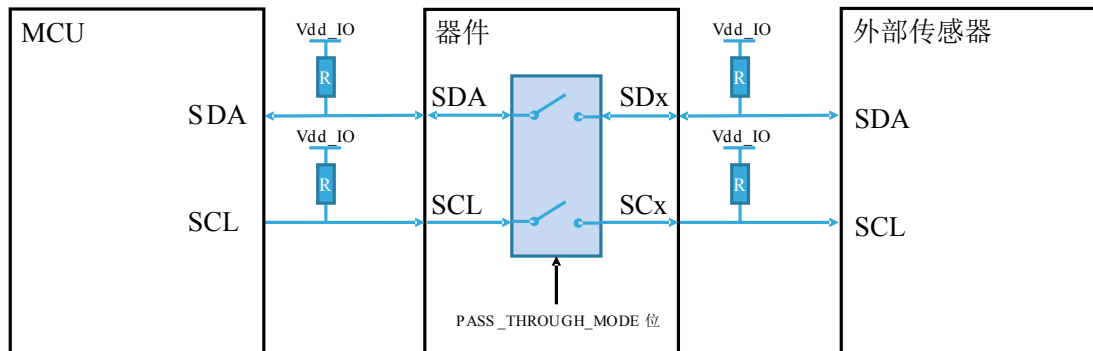
图 20. SENSOR_HUB_X 配置示例

传感器 1	<div> <div>SLV0_SUBADD(16h)=28h</div> <div>SLV0_CONFIG(17h) - Slave0_numop[2:0]=3</div> </div>	SENSOR_HUB_1	寄存器 28h 的值	传感器 1
		SENSOR_HUB_2	寄存器 29h 的值	
		SENSOR_HUB_3	寄存器 2Ah 的值	
传感器 2	<div> <div>SLV1_SUBADD(19h)=00h</div> <div>SLV1_CONFIG(1Ah) - Slave1_numop[2:0]=6</div> </div>	SENSOR_HUB_4	寄存器 00h 的值	传感器 2
		SENSOR_HUB_5	寄存器 01h 的值	
		SENSOR_HUB_6	寄存器 02h 的值	
		SENSOR_HUB_7	寄存器 03h 的值	
		SENSOR_HUB_8	寄存器 04h 的值	
		SENSOR_HUB_9	寄存器 05h 的值	
传感器 3	<div> <div>SLV2_SUBADD(1Ch)=20h</div> <div>SLV2_CONFIG(1Dh) - Slave2_numop[2:0]=4</div> </div>	SENSOR_HUB_10	寄存器 20h 的值	传感器 3
		SENSOR_HUB_11	寄存器 21h 的值	
		SENSOR_HUB_12	寄存器 22h 的值	
		SENSOR_HUB_13	寄存器 23h 的值	
传感器 4	<div> <div>SLV3_SUBADD(1Fh)=40h</div> <div>SLV3_CONFIG(20h) - Slave3_numop[2:0]=5</div> </div>	SENSOR_HUB_14	寄存器 40h 的值	传感器 4
		SENSOR_HUB_15	寄存器 41h 的值	
		SENSOR_HUB_16	寄存器 42h 的值	
		SENSOR_HUB_17	寄存器 43h 的值	
		SENSOR_HUB_18	寄存器 44h 的值	

7.3 传感器集线器 (sensor hub) 直通功能

MASTER_CONFIG 寄存器的 PASS_THROUGH_MODE 位用来使能/禁用 I²C 接口直通。当该位置为 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通功能。此功能可用于配置外部传感器。

图 21. 直通功能



可按照以下步骤来使能直通模式：

1. 如果 I²C 主器件使能 (MASTER_ON = 1)，则将其关闭（将 MASTER_ON 位置为 0）并等待 300 μ s。
2. 如果 I²C 主线上的上拉使能，则将其禁用（将 IF_CFG 寄存器的 SHUB_PU_EN 位置为 0）。
3. 通过将 PASS_THROUGH_MODE 位置为 1 来使能直通模式。

7.4 传感器集线器 (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线器 (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 120 Hz 的加速度计，I²C 辅助总线上带外部上拉电阻。

- | | |
|--|---------------------------------------|
| 1. 将 40h 写入 FUNC_CFG_ACCESS | // 使能对传感器集线器寄存器的访问 |
| 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 81h 写入 SLV0_CONFIG | // 读取一个字节, SHUB_ODR = 120 Hz |
| 5. 将 44h 写入 MASTER_CONFIG | // 强制读取 WRITE_ONCE |
| | // I ² C 主器件使能，只使用 slave 0 |
| 6. 将 00h 写入 FUNC_CFG_ACCESS | // 禁用对传感器集线器寄存器的访问 |
| 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 | // 等待传感器集线器触发信号 |
| 9. 轮询 STATUS_MASTER_MAINPAGE，直至 SENS_HUB_ENDOP = 1 | // 等待传感器集线器读事务 |
| 10. 将 40h 写入 FUNC_CFG_ACCESS | // 使能对传感器集线器寄存器的访问 |

- | | |
|------------------------------|---------------------------|
| 11. 将 00h 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 | // 禁用对传感器集线器寄存器的访问 |

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 | // 使能对传感器集线器寄存器的访问 |
| 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 8xh 写入 SLV0_CONFIG | // 读取 x 个字节（最多 6 个），SHUB_ODR = 120 Hz |
| 5. 将 44h 写入 MASTER_CONFIG | // 强制读取 WRITE_ONCE |
| | // I ² C 主器件使能，只使用 slave 0 |
| 6. 将 00h 写入 FUNC_CFG_ACCESS | // 禁用对传感器集线器寄存器的访问 |

在执行步骤 6 后，传感器集线器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 120 Hz 的加速度计，I²C 辅助总线上带外部上拉电阻。

- | | |
|---|---------------------------------------|
| 1. 将 40h 写入 FUNC_CFG_ACCESS | // 使能对传感器集线器寄存器的访问 |
| 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 80h 写入 SLV0_CONFIG | // SHUB_ODR = 120 Hz |
| 5. 将 VAL 写入 DATAWRITE_SLV0 | // 配置要在 REG 中写入的值 (VAL) |
| 6. 将 44h 写入 MASTER_CONFIG | // 为单次写入使能 WRITE_ONCE |
| | // I ² C 主器件使能，只使用 slave 0 |
| 7. 轮询 STATUS_MASTER，
直至 WR_ONCE_DONE = 1 | // 等待传感器集线器写事务 |
| 8. 将 00h 写入 MASTER_CONFIG | // I ² C 主器件禁用 |
| 9. 等待 300 μs | |
| 10. 将 00h 写入 FUNC_CFG_ACCESS | // 禁用对传感器集线器寄存器的访问 |

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 到 SENSOR_HUB_6 中。

- | | |
|-------------------------|---|
| 1. 将 06h 写入 CTRL1 | // 开启加速度计（以获取触发信号），频率为 120 Hz |
| 2. 执行一次性读取，其中： | // 检查 LIS2MDL WHO_AM_I 寄存器 |
| SLV0_ADD = 3Dh | // LIS2MDL 从器件地址为 3Ch 且 rw_0=1 |
| SLV0_SUBADD = 4Fh | // WHO_AM_I 寄存器地址为 4Fh |
| 3. 执行一次性写入，其中： | // 写入 LIS2MDL 寄存器 CFG_REG_A (60h) = 8Ch |
| SLV0_ADD = 3Ch | // LIS2MDL 从器件地址为 3Ch 且 rw_0=0 |
| SLV0_SUBADD = 60h | // 使能温度补偿 |
| DATAWRITE_SLV0 = 8Ch | // 使能磁力计，100 Hz ODR，Continue 模式 |
| 4. 执行一次性写入，其中： | // 写入 LIS2MDL 寄存器 CFG_REG_B (61h) = 02h |
| SLV0_ADD = 3Ch | // LIS2MDL 从器件地址为 3Ch 且 rw_0=0 |
| SLV0_SUBADD = 61h | // 使能磁力计偏移消除 |
| DATAWRITE_SLV0 = 02h | |
| 5. 执行一次性写入，其中： | // 写入 LIS2MDL 寄存器 CFG_REG_B (62h) = 10h |
| 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 | |
| 6. 设置连续读取，其中： | // LIS2MDL 从器件地址为 3Ch 且 rw_0=1 |
| SLV0_ADD = 3Dh | // 磁力计输出寄存器，从 68h 开始 |
| SLV0_SUBADD = 68h | // 设置从 I ² C 主接口进行连续 6 字节读取 |
| SLV0_CONFIG = 80h 06h | |

8 模式 3——OIS 功能

LSM6DSV16X 内置专门面向 OIS 应用的陀螺仪和加速度计 DSP。通过使能 OIS 功能（模式 3），可以从多个外部器件访问该器件。I²C/SPI（3/4 线）/MIPI I3C[®] 从接口和辅助 SPI（3/4 线）从接口可用于连接外部主器件。举例来说，在光学防抖 (OIS) 应用中，可以使用它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。

有两种不同的方法用于使能和配置 OIS 功能，具体可使用 FUNC_CFG_ACCESS 寄存器中的 OIS_CTRL_FROM_UI 位来选择（通过主接口进行配置）：

- 辅助 SPI 完全控制 (OIS_CTRL_FROM_UI = 0)：通过辅助 SPI 进行使能和配置（参见第 8.1 节 辅助 SPI 全控制）
- 主接口完全控制 (OIS_CTRL_FROM_UI = 1)：通过主接口进行使能和配置（参见第 8.5 节 主接口全控制）

提示

OIS_CTRL_FROM_UI 位通过软件复位程序进行复位。

这两种模式均能够完全灵活地管理 OIS 链，并且为主接口和 SPI 辅助接口提供相同的功能和配置选项。

8.1 辅助 SPI 全控制

辅助 SPI 全控制针对连接到 SPI 辅助接口的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。可通过从辅助 SPI 访问和设置寄存器 SPI2_INT_OIS、SPI2_CTRL1_OIS、SPI2_CTRL2_OIS 和 SPI2_CTRL3_OIS 来配置辅助 SPI 模式。

将 SPI2_CTRL1_OIS 寄存器的 OIS_G_EN 位置为 1 可使能陀螺仪 OIS 链。当陀螺仪 OIS 链使能时，可通过辅助 SPI 接口（3/4 线）获取陀螺仪输出值，满量程通过 SPI2_CTRL2_OIS 寄存器的 FS_G_OIS_[2:0] 位进行选择，ODR 为 7680 Hz。

将 SPI2_CTRL1_OIS 寄存器的 OIS_XL_EN 位置为 1 可使能加速度计 OIS 链。当加速度计 OIS 链使能时，可通过辅助 SPI 接口（3/4 线）获取加速度计输出值，满量程通过 SPI2_CTRL3_OIS 寄存器的 FS_XL_OIS_[1:0] 位进行选择，ODR 为 7680 Hz。

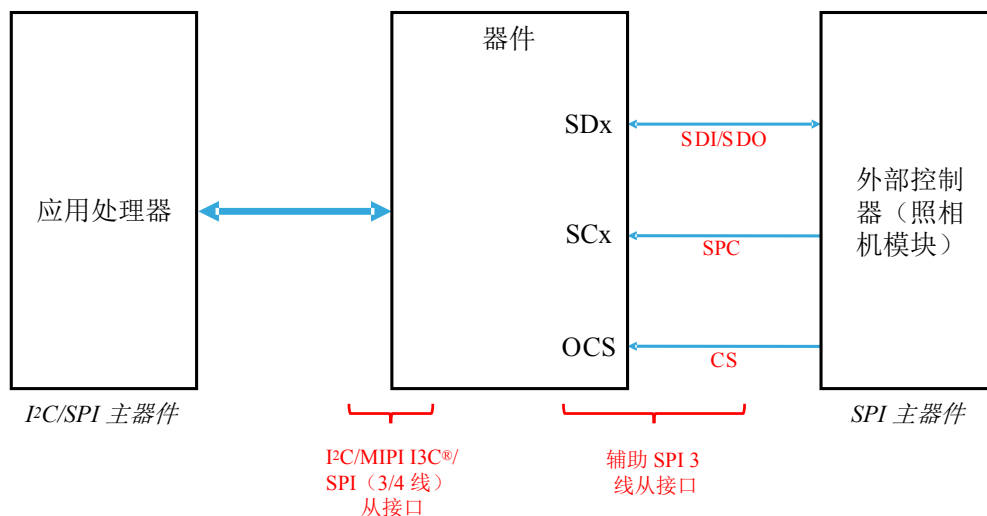
模式 3 使能后器件引脚的功能如下表所示。

表 60. 模式 3 引脚说明

引脚	模式 3 功能
SDx/AH1/Qvar1	辅助 SPI 3/4 线串行数据输入 (SDI) 和 SPI 3 线串行数据输出 (SDO)
SCx/AH2/Qvar2	辅助 SPI 3/4 线串行端口时钟 (SPC)
OCS_Aux	辅助 SPI 3/4 线芯片选择 (CS)
SDO_Aux	辅助 SPI 4 线串行数据输出 (SDO)

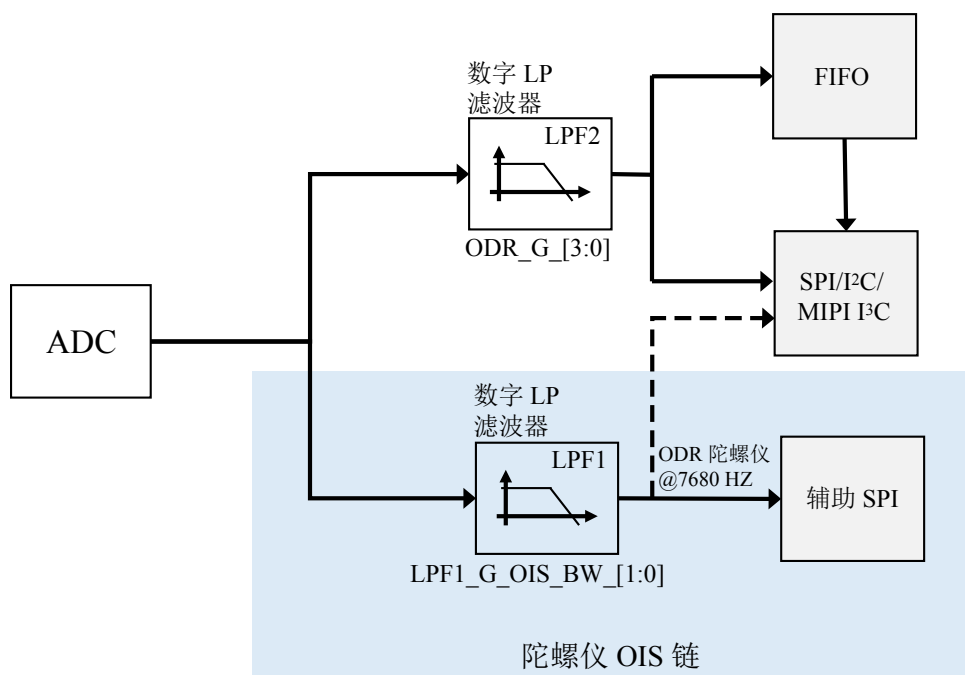
如果使用 SPI 3 线接口（SPI2_CTRL1_OIS 中的 SIM_OIS 位 = 1），外部器件必须按照图 22 所示连接到 LSM6DSV16X。在使用 SPI 4 线接口（还连接 SDO_Aux 引脚）时，必须相应地更改设置。

图 22. 模式 3 下的外部控制器连接 (SPI 3 线)



下图显示了陀螺仪滤波链。数字低通滤波器 LPF1 专用于 OIS 链，可通过 SPI2_CTRL2_OIS 寄存器的 LPF1_G_OIS_BW_[1:0] 位配置陀螺仪 OIS 链的带宽。

图 23. 模式 3 下的陀螺仪滤波链

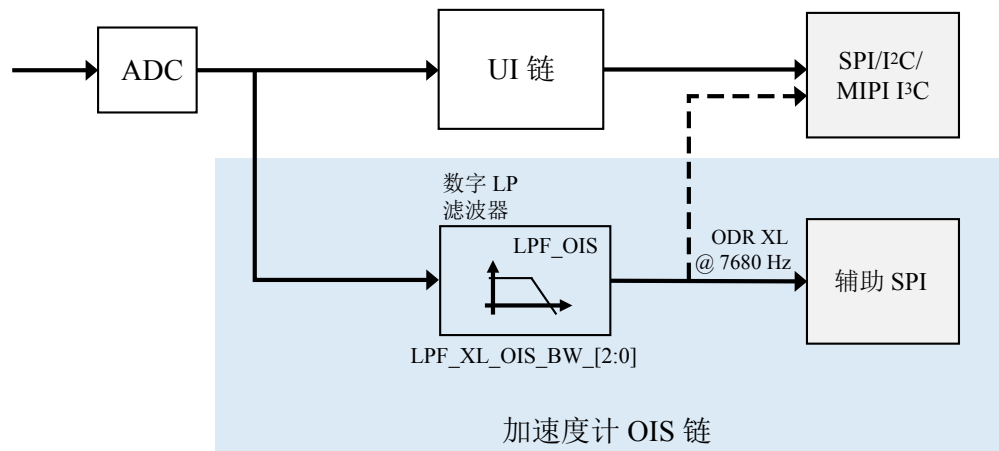


提示

± 4000 dps 陀螺仪满量程配置不兼容模式 3。当选择 ± 4000 dps 满量程时，陀螺仪 OIS 链必须禁用 (UI_CTRL1_OIS/SPI2_CTRL1_OIS 寄存器的 OIS_G_EN 位必须置为 0)。

下图显示了加速度计滤波链。数字低通滤波器 LPF_OIS 专用于 OIS 链，可通过 SPI2_CTRL3_OIS 寄存器的 LPF_XL_OIS_BW_[2:0] 位配置加速度计 OIS 链的带宽。

图 24. 模式 3 下的加速度计滤波链


提示

加速度计 UI 链的配置不受加速度计 OIS 链使能/禁用的影响，但有一种情况例外：加速度计 OIS 链使能时，无法使用加速度计正常工作模式（CTRL1 (10h) 寄存器中的 OP_MODE_XL_[2:0] = 111）。

8.2 SPI2 寄存器

主接口始终可用，可读取寄存器 `OUTX_L_G` 到 `OUTZ_H_G`（22h 到 27h）中的陀螺仪输出值，并可通过 `CTRL2` 寄存器选择满量程和 `ODR`。同样地，可通过主接口读取寄存器 `OUTX_L_A` 到 `OUTZ_H_A`（28h 到 2Dh）中的加速度计输出值，并可通过 `CTRL1` 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。

`SPI2_INT_OIS`、`SPI2_CTRL1_OIS`、`SPI2_CTRL2_OIS` 和 `SPI2_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。此外，主接口可在读模式下访问 `OIS` 控制寄存器（`UI_INT_OIS`、`UI_CTRL1_OIS`、`UI_CTRL2_OIS` 和 `UI_CTRL3_OIS`）。

只有 `SPI2_CTRL1_OIS` 寄存器中的 `SPI2_READ_EN` 位置为 1 时，才能使能通过辅助 SPI 读取 `OIS` 数据。

当陀螺仪 `OIS` 链使能时，可通过辅助 SPI 接口从寄存器 `SPI2_OUTX_L_G_OIS` 到 `SPI2_OUTZ_H_G_OIS`（22h 到 27h）中读取陀螺仪输出值。当 `OIS` 链上有新的陀螺仪数据可用时，`SPI2_STATUS_REG_OIS` 寄存器的 `GDA` 位置为 1。在读取输出数据寄存器（23h、25h 和 27h）的高部分时，`GDA` 位复位。当陀螺仪 `OIS` 链处于稳定阶段时，`SPI2_STATUS_REG_OIS` 寄存器中的 `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。

当加速度计 `OIS` 链使能时，也可通过辅助 SPI 接口从寄存器 `SPI2_OUTX_L_A_OIS` 到 `SPI2_OUTZ_H_A_OIS`（28h 到 2Dh）中读取加速度计输出值。当 `OIS` 链上有新的加速度计数据可用时，`SPI2_STATUS_REG_OIS` 寄存器的 `XLDA` 位置为 1。在读取输出数据寄存器（29h、2Bh 和 2Dh）的高部分时，`XLDA` 位复位。

在模式 3 下，还可通过主接口读取 `ODR` 为 7680 Hz 时的陀螺仪 `OIS` 数据和加速度计 `OIS` 数据。如果陀螺仪 `OIS` 使能且 `CTRL_EIS` 寄存器的 `G_EIS_ON_G_OIS_OUT_REG` 位置为 0，则可通过主接口从寄存器 `UI_OUTX_L_G_OIS_EIS` 到 `UI_OUTZ_H_G_OIS_EIS`（2Eh 到 33h）读取陀螺仪输出值。如果加速度计 `OIS` 使能且加速度计双通道模式禁用（`CTRL8` 寄存器的 `XL_DualC_EN` 位置为 0），则可通过主接口从寄存器 `UI_OUTX_L_A_OIS_DualC` 到 `UI_OUTZ_H_A_OIS_DualC`（34h 到 39h）读取加速度计输出值。当 `OIS` 链上有新的陀螺仪/加速度计数据可用时，`STATUS_REG` 寄存器的 `OIS_DRDY` 位（可通过主接口访问）置为 1。

还可以通过辅助 SPI 接口读取 `SPI2_OUT_TEMP_L` 和 `SPI2_OUT_TEMP_H` 寄存器，从而读取温度传感器输出数据。

可同时从两个外部主器件读取器件的所有寄存器。

8.2.1 SPI2_INT_OIS (6Fh)

表 61. SPI2_INT_OIS 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
INT2_DRDY_OIS	DRDY_MASK_OIS	0	ST_OIS_CLAMPDIS	ST_G_OIS_1	ST_G_OIS_0	ST_XL_OIS_1	ST_XL_OIS_0

- 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。
- DRDY_MASK_OIS 可用于屏蔽路由到 INT2 引脚的中断信号，直到 OIS 加速度计/陀螺仪滤波器稳定状态结束为止。OIS 加速度计和 OIS 陀螺仪单独屏蔽。
- 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 OIS 链读取的输出值将显示在从 UI 链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 OIS 链读取的输出值始终钳位至值 8000h。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。根据设计，输出值的有效范围介于 8004h (-32764 LSB) 和 7FFCh (+32764 LSB) 之间，其中 8000h (-32768 LSB) 是一个特殊值，可通过辅助 SPI 使用该值来检测是否已从 UI 侧使能自检功能。
- 可以将 ST_G_OIS_[1:0] 位置位，以便选择陀螺仪 OIS 链上的自检功能（参见第 12 节 自检功能了解详细信息）。
- 可以将 ST_XL_OIS_[1:0] 位置位，以便选择加速度计 OIS 链上的自检功能（参见第 12 节 自检功能了解详细信息）。

8.2.2 SPI2_CTRL1_OIS (70h)

表 62. SPI2_CTRL1_OIS 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
0	0	SIM_OIS	0	0	OIS_XL_EN	OIS_G_EN	SPI2_READ_EN

- 为了使用 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。
- OIS_XL_EN 位可用于使能加速度计 OIS 链。
- OIS_G_EN 位可用于使能陀螺仪 OIS 链。
- 要使能辅助 SPI 以从寄存器 SPI2_OUTX_L_G_OIS (22h) 到 SPI2_OUTZ_H_A_OIS (2Dh) 读取 OIS 输出数据，SPI2_READ_EN 位必须置为 1。

8.2.3 SPI2_CTRL2_OIS (71h)

表 63. SPI2_CTRL2_OIS 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	LPF1_G_OIS_BW_1	LPF1_G_OIS_BW_0	FS_G_OIS_2	FS_G_OIS_1	FS_G_OIS_0

- LPF1_G_OIS_BW_[1:0] 位可用于选择陀螺仪数字 LPF1 滤波器带宽。表 64 所示为使用所有配置获得的截止频率和相位延迟值。
- FS_G_OIS_[2:0] 位可用于选择陀螺仪 OIS 满量程（000: ± 125 dps（默认值）；001: ± 250 dps；010: ± 500 dps；011: ± 1000 dps；100: ± 2000 dps）。

表 64. LPF1 滤波器配置

LPF1_G_OIS_BW_[1:0]	截止频率 [Hz]	相位 [°] @ 20 Hz	UI 陀螺仪链接通时的总体最大稳定时间 [要丢弃的采样数] ⁽¹⁾⁽²⁾
00	293	-7.1	33
01	217	-9.1	43
10	158	-11.9	58
11	476	-5.1	23

1. 最终值的 99% 时的稳定时间
2. 如果 UI 陀螺仪处于掉电模式，则等待 70 ms 后才能丢弃本列指示的采样数。

8.2.4 SPI2_CTRL3_OIS (72h)

表 65. SPI2_CTRL3_OIS 寄存器

b7	b6	b5	b4	b3	b2	b1	b0
0	0	LPF_XL_OIS_BW_2	LPF_XL_OIS_BW_1	LPF_XL_OIS_BW_0	0	FS_XL_OIS_1	FS_XL_OIS_0

- LPF_XL_OIS_BW_[2:0] 位可用于选择加速度计数字 LPF_OIS 滤波器带宽。表 66 所示为使用所有配置获得的截止频率和相位延迟值。
- FS_XL_OIS_[1:0] 位可用于选择加速度计 OIS 满量程（00: ± 2 g（默认值）；01: ± 4 g；10: ± 8 g；11: ± 16 g）

表 66. LPF_OIS 滤波器配置

LPF_XL_OIS_BW_[2:0]	截止频率 [Hz]	相位 [°]	UI 加速度计处于掉电模式或 LP 模式 1/2/3 时的总体最大稳定时间 [要丢弃的采样数] ⁽¹⁾	UI 加速度计处于 HP 模式时的总体最大稳定时间 [要丢弃的采样数] ⁽¹⁾
000	749	-3.41 @ 20 Hz	21	10
001	539	-4.04 @ 20 Hz	21	13
010	342	-5.31 @ 20 Hz	21	19
011	162	-9.08 @ 20 Hz	39	39
100	78.5	-16.4 @ 20 Hz	78	78
101	38.6	-29.6 @ 20 Hz	156	156
110	19.3	-28.8 @ 10 Hz	313	313
111	9.8	-29.1 @ 5 Hz	626	626

1. 最终值的 99% 时的稳定时间

8.2.5 SPI2_STATUS_REG_OIS (1Eh)

表 67. SPI2_STATUS_REG_OIS 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。
- 当 OIS 链上的寄存器 SPI2_OUTX_L_G_OIS 到 SPI2_OUTZ_H_G_OIS (22h 到 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 位复位。
- 当 OIS 链上的寄存器 SPI2_OUTX_L_A_OIS 到 SPI2_OUTZ_H_A_OIS (28h 到 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 位复位。

8.3 通过辅助 SPI 读取 OIS 陀螺仪数据

器件上电后通过辅助 SPI 4 线接口读取 OIS 陀螺仪输出数据的流程如下：

- 等待 10 ms // 启动时间
// 该时间段结束后器件处于掉电模式
- 将 03h 写入 SPI2_CTRL1_OIS // 接通陀螺仪 OIS 链并使能辅助 SPI 来读取 OIS 数据
// (OIS 陀螺仪: FS = ± 125 dps / ODR = 7680 Hz)
- 等待 75 ms // 陀螺仪最大导通时间为 70 ms
// 选择的 LPF1 (00) 稳定时间为 4.3 ms
// (33 个采样 @ 7680 Hz)
- 读取输出寄存器 22h 到 27h // 通过辅助 SPI 读取陀螺仪输出数据

8.4 通过辅助 SPI 读取 OIS 加速度计数据

器件上电后通过辅助 SPI 4 线接口读取 OIS 加速度计输出数据的流程如下：

- 等待 10 ms // 启动时间
// 该时间段结束后器件处于掉电模式
- 将 05h 写入 SPI2_CTRL1_OIS // 接通加速度计 OIS 链并使能辅助 SPI 来读取 OIS 数据
// (OIS 加速度计: FS = ± 2 g / ODR = 7680 Hz)
- 等待 2 ms // 选择的 LPF_OIS (000) 稳定时间为 1.3 ms
// (10 个采样 @ 7680 Hz)
- 读取输出寄存器 28h 到 2Dh // 通过辅助 SPI 读取加速度计输出数据

8.5 主接口全控制

除了第 8.1 节 辅助 SPI 全控制中描述的辅助 SPI 全控制，LSM6DSV16X 还提供了另一种管理 OIS 数据链的方式：主接口全控制。它允许使能 OIS 链和直接通过连接到应用处理器的主接口获取 UI 和 OIS 数据。通过从主接口将 `FUNC_CFG_ACCESS` 寄存器中的 `OIS_CTRL_FROM_UI` 位置为 1 使能主接口全控制。在将此位置 1 后，可以从主接口使用 `UI_INT_OIS`、`UI_CTRL1_OIS`、`UI_CTRL2_OIS` 和 `UI_CTRL3_OIS` 寄存器直接配置 OIS 功能。与这些寄存器的位相关的功能与通过相应 `SPI2_xxx` 寄存器中包含的位实现的功能相同（参见表 3. SPI 寄存器）。

使能主接口全控制后，寄存器 `UI_OUTX_L_G_OIS_EIS` 到 `UI_OUTZ_H_G_OIS_EIS`（2Eh 到 33h）中的 OIS 陀螺仪数据（频率为 7680 Hz）可用，寄存器 `UI_OUTX_L_A_OIS_DualC` 到 `UI_OUTZ_H_A_OIS_DualC`（34h 到 39h）中的 OIS 加速度计数据（频率为 7680 Hz）可用。`UI_STATUS_REG_OIS` 寄存器（地址 44h）充当此类数据的状态寄存器。

此外，通过将 `UI_CTRL1_OIS` 寄存器中的 `SPI2_READ_EN` 位置为 1，也可以通过辅助 SPI 接口读取 OIS 链数据。辅助 SPI 还可以访问 `SPI2_INT_OIS`、`SPI2_CTRL1_OIS`、`SPI2_CTRL2_OIS` 和 `SPI2_CTRL3_OIS` 寄存器（只读模式）。

8.6 UI/SPI2 共享寄存器

LSM6DSV16X 提供了 `UI_SPI2_SHARED_0` (65h) 到 `UI_SPI2_SHARED_5` (6Ah) 共六个由主接口和辅助 SPI 接口共享的寄存器，可用作主接口主机与辅助接口主机之间的接触点。这些共享寄存器一次只能由一个接口访问，而具体由哪个接口访问通过 `UI_HANDSHAKE_CTRL` 寄存器的 `UI_SHARED_REQ` 和 `UI_SHARED_ACK` 位以及 `SPI2_HANDSHAKE_CTRL` 寄存器的 `SPI2_SHARED_REQ` 和 `SPI2_SHARED_ACK` 位来管理。

当一个接口（UI 为主接口，SPI2 为辅助接口）要写入共享寄存器时，它必须将相应的请求位置为 1（对于主接口，为 `UI_SHARED_REQ`；对于辅助接口，为 `SPI2_SHARED_REQ`）并轮询相应的应答位（对于主接口，为 `UI_SHARED_ACK`；对于辅助接口，为 `SPI2_SHARED_ACK`），直到该应答位变为 1 为止。当相应的应答位置为 1 时，该接口可对共享寄存器进行写访问。该接口完成对共享寄存器的写访问后，它必须将相应的请求位再置为 0，并轮询相应的应答位，直到该应答位再次变为 0 为止。两个接口始终可对共享寄存器进行读访问。

9 先进先出 (FIFO) 缓存器

为了限制主处理器干预并简化事件识别的后处理数据，LSM6DSV16X 嵌入了一个 1.5 KB（最多 4.5 KB，压缩功能使能）的先进先出缓存器 (FIFO)。

FIFO 可配置为存储以下数据：

- 陀螺仪传感器数据（UI 或 EIS 通道）
- 加速度计传感器数据（任一通道）
- 时间戳数据
- 温度传感器数据
- 外部传感器（连接到传感器集线器接口）数据
- 计步器（和相关时间戳）数据
- SFLP 游戏旋转向量、重力向量和陀螺仪偏差
- 机器学习内核滤波器、特征和结果

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 到 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 位域可用于识别 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，有七种不同的 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
- ContinueWTM-Full 模式

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

可编程 FIFO 阈值可以使用 FIFO_CTRL1 中的 WTM_[7:0] 位来设置。

通过 INT1_CTRL 寄存器的 INT1_FIFO_FULL、INT1_FIFO_TH 和 INT1_FIFO_OVR 位，以及 INT2_CTRL 寄存器的 INT2_FIFO_FULL、INT2_FIFO_TH 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 4.5 KB 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 256 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 缓冲器大小等于 1792 字节，可包含 1536 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：可使能元信息标签，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（任一通道）
 - b. 陀螺仪传感器（UI 或 EIS 通道）
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. 从传感器集线器接口读取的外部传感器
 - b. 计步器传感器
 - c. SFLP 游戏旋转向量、重力向量和陀螺仪偏差
 - d. 机器学习内核滤波器、特征和结果

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。

可通过以下不同事件触发 FIFO 写入：

- 内部数据准备就绪信号（加速度计与陀螺仪之间的最快速传感器）
- 传感器集线器数据准备就绪
- 步伐检测事件
- 虚拟传感器新数据可用

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，用于将 FIFO 事件路由至两条中断线：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_EN_A 寄存器的 STEP_COUNTER_FIFO_EN 位，用于使能在 FIFO 中对计步器数据进行批处理
 - EMB_FUNC_FIFO_EN_A 寄存器的 MLC_FIFO_EN 位，用于使能在 FIFO 中对机器学习内核结果进行批处理
 - EMB_FUNC_FIFO_EN_B 寄存器的 MLC_FILTER_FEATURE_FIFO_EN 位，用于使能在 FIFO 中对机器学习内核滤波器和特征进行批处理
 - EMB_FUNC_FIFO_EN_B 寄存器的 SFLP_GBIAS_FIFO_EN 位，用于使能在 FIFO 中对陀螺仪偏差数据进行批处理（必须使能 SFLP 嵌入功能）
 - EMB_FUNC_FIFO_EN_B 寄存器的 SFLP_GRAVITY_FIFO_EN 位，用于使能在 FIFO 中对重力向量数据进行批处理（必须使能 SFLP 嵌入功能）
 - EMB_FUNC_FIFO_EN_B 寄存器的 SFLP_GAME_FIFO_EN 位，用于使能在 FIFO 中对游戏旋转向量数据进行批处理（必须使能 SFLP 嵌入功能）
 - 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
 - SLV0_CONFIG、SLV1_CONFIG、SLV2_CONFIG 和 SLV3_CONFIG 传感器集线器寄存器的 BATCH_EXT_SENS_0_EN、BATCH_EXT_SENS_1_EN、BATCH_EXT_SENS_2_EN 和 BATCH_EXT_SENS_3_EN 位，用于使能在 FIFO 中对相关外部传感器数据进行批处理

9.2.1 FIFO_CTRL1

FIFO_CTRL1 寄存器包含 FIFO 水印阈值。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 68. FIFO_CTRL1 寄存器

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
WTM_7	WTM_6	WTM_5	WTM_4	WTM_3	WTM_2	WTM_1	WTM_0

9.2.2 FIFO_CTRL2

表 69. 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	ODR_CHG_EN	0	UNCOMPR_RATE_1	UNCOMPR_RATE_0	XL_DualC_BATCH_FROM_FSM

FIFO_CTRL2 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。
- UNCOMPR_RATE_[1:0] 用于配置压缩算法，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 70. 强制未压缩数据写入配置

UNCOMPR_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 寄存器还包含 ODR_CHG_EN 位和 XL_DualC_BATCH_FROM_FSM 位，ODR_CHG_EN 位置为 1 可使能 CFG-Change 辅助传感器在 FIFO 中进行批处理（接下来的章节中将进行介绍），而 XL_DualC_BATCH_FROM_FSM 位置为 1（与特定 FSM 配置结合使用）可使能加速度计通道 2 在 FIFO 缓冲器中进行批处理。在这种情况下，必须通过将 CTRL8 寄存器的 XL_DualC_EN 位置为 1 来使能加速度计双通道模式，并配置一个 FSM 来实际使能批处理操作。请参考有限状态机应用笔记（可从 ST 网站 www.st.com 上获取）。

9.2.3 FIFO_CTRL3

表 71. 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位域。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 位域配置的输出数据率。

下面的表格显示了所有可选择的批处理数据率。

表 72. 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
0001	1.875
0010	7.5
0011	15
0100	30
0101	60
0110	120
0111	240
1000	480
1001	960
1010	1920
1011	3840
1100	7680

表 73. 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	1.875
0010	7.5
0011	15
0100	30
0101	60
0110	120
0111	240
1000	480
1001	960
1010	1920
1011	3840
1100	7680

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位域。

时间戳写入速率被配置为最大批处理数据率 (BDR_MAX) 除以 DEC_TS_BATCH_[1:0] 位域指定的抽取因子。BDR_MAX 为以下批处理数据率中的最大批处理数据率：

- 加速度计批处理数据率 (BDR_XL)
- 加速度计通道 2 批处理数据率，等于 ODR_XL（使能在 FIFO 中对加速度计通道 2 数据进行批处理时）
- 陀螺仪批处理数据率 (BDR_GY)
- 陀螺仪 EIS 批处理数据率，等于 ODR_G_EIS（使能在 FIFO 中对陀螺仪 EIS 通道数据进行批处理时）
- 传感器集线器批处理数据率 (BDR_SHUB)

下表显示了可编程抽取因子。

表 74. 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	BDR_MAX
10	BDR_MAX / 8
11	BDR_MAX / 32

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

表 75. 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	1.875
10	15
11	60

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。此外，FIFO_CTRL4 寄存器还包含 G_EIS_FIFO_EN 位，该位置为 1 时可使能陀螺仪 EIS 通道数据的批处理。

表 76. 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	G_EIS_FIFO_EN	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 77. COUNTER_BDR_REG1 寄存器

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
0	TRIG_COUNTER_BDR _1	TRIG_COUNTER_BDR _0	0	0	0	CNT_BDR_TH_9	CNT_BDR_TH_8

TRIG_COUNTER_BDR_[1:0] 位域选择 BDR 计数器的触发信号：

- 00：选择加速度计传感器作为触发信号
- 01：选择陀螺仪传感器（UI 通道）作为触发信号
- 10 或 11：选择陀螺仪传感器（EIS 通道）作为触发信号

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_[9:0] 位进行配置。阈值的上部包含在寄存器 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）。

当 FIFO 为空时，BDR 计数器自动复位。

9.2.6 COUNTER_BDR_REG2

COUNTER_BDR_REG2 寄存器包含 BDR 计数器阈值的下部。

表 78. 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 79. 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 80. 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	0	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。
- 当存储在 FIFO 中的下一组数据会使 FIFO 全满（即，DIFF_FIFO_8 = 1）或生成 FIFO 溢出时，FIFO_FULL_IA 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 [9:0] 位配置的 BDR 计数器阈值时，此位置为高电平。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_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 [8:0] 位域的值对应 FIFO 中的 7 字节字数量。

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 81. 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_SENSOR [4:0] 位域标识 6 个数据字节中保存的传感器（表 82）。
- TAG_CNT [1:0] 位域标识 FIFO 时隙（如后面的章节所述）。

下表包含了 TAG_SENSOR [4:0] 位域的所有可能值和相关传感器类型。

表 82. TAG_SENSOR 位域和相关传感器

TAG_SENSOR [4:0]	传感器名称	传感器类别	说明
0x00	空	-	FIFO 为空条件
0x01	陀螺仪 NC	主要	陀螺仪未压缩数据
0x02	加速度计 NC	主要	加速度计未压缩数据

TAG_SENSOR_[4:0]	传感器名称	传感器类别	说明
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_1	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 2xC	主要	加速度计 2x 压缩数据
0x09	加速度计 3xC	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	传感器集线器 Slave 0	虚拟	来自 slave 0 的传感器集线器数据
0x0F	传感器集线器 Slave 1	虚拟	来自 slave 1 的传感器集线器数据
0x10	传感器集线器 Slave 2	虚拟	来自 slave 2 的传感器集线器数据
0x11	传感器集线器 Slave 3	虚拟	来自 slave 3 的传感器集线器数据
0x12	计步器	虚拟	计步器数据
0x13	游戏旋转向量	虚拟	SFLP 生成的游戏旋转向量
0x16	陀螺仪偏差	虚拟	SFLP 生成的陀螺仪偏差
0x17	重力向量	虚拟	SFLP 生成的重力向量
0x19	传感器集线器 Nack	虚拟	来自 slave 0/1/2/3 的传感器集线器 nack
0x1A	MLC 结果	虚拟	机器学习内核生成的结果
0x1B	MLC 滤波器	虚拟	机器学习内核生成的滤波器
0x1C	MLC 特征	虚拟	机器学习内核生成的特征
0x1D	加速度计 DualC	主要	加速度计通道 2 数据
0x1E	陀螺仪 EIS	主要	陀螺仪 EIS 通道数据

9.2.10

FIFO_DATA_OUT

可通过六个专用寄存器（从地址 79h 到地址 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 LSM6DSV16X 器件的物理传感器：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] 位域进行配置。批处理数据率必须小于等于通过 CTRL1 和 CTRL2 寄存器的 ODR_XL_[3:0] 和 ODR_G_[3:0] 位域配置的相关传感器输出数据率。

必须通过 FSM 触发加速度计通道 2 数据的批处理（XL_DualC_BATCH_FROM_FSM 位必须置为 1 且一个 FSM 必须配置为实际使能批处理操作）。加速度计通道 2 数据根据 CTRL1 寄存器的 ODR_XL_[3:0] 位域配置存储到 FIFO 中。

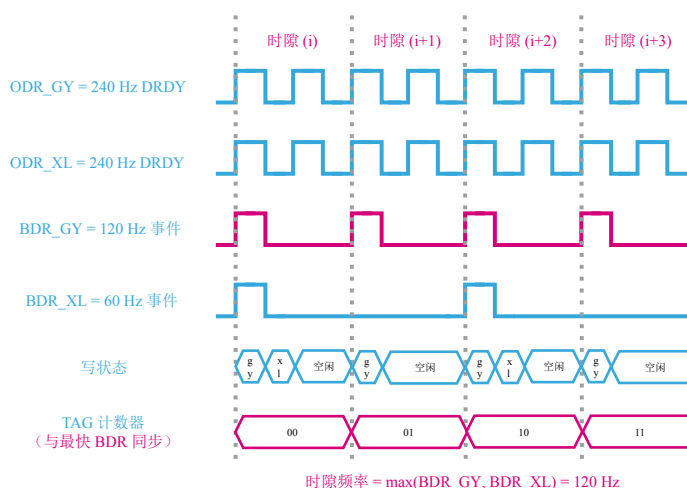
通过将 FIFO_CTRL4 寄存器的 G_EIS_FIFO_EN 位置为 1 来使能陀螺仪 EIS 通道数据的批处理。陀螺仪 EIS 通道数据根据 CTRL_EIS 寄存器的 ODR_G_EIS_[1:0] 位域配置存储到 FIFO 中。

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 位域）递增。此计数器由两个位组成，其值连续递增（从 00 到 11）以标识不同时隙。

图 25. 主要传感器和时隙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隙，其频率等于 BDR_XL 和 BDR_GY 中的最大值，因为加速度计通道 2 和陀螺仪 EIS 通道的数据在本例中不进行批处理。

通常情况下，TAG 计数器的频率等于加速度计或陀螺仪的最大批处理数据率（以较快者为准）（还需考虑加速度计通道 2 和陀螺仪 EIS 通道）。

图 25. 主要传感器和时隙的定义



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

表 83. FIFO 中的主要传感器输出数据格式

TAG	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（必须将 FUNCTIONS_ENABLE 寄存器的 TIMESTAMP_EN 位置为 1 并正确地配置 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 ODR_CHG_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 84. 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 85. 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]
G_EIS_FIFO_EN	FIFO_DATA_OUT_Z_L[4]
XL_DualC_BATCH_EN ⁽¹⁾	FIFO_DATA_OUT_Z_L[5]
0	FIFO_DATA_OUT_Z_L[7:6]
BDR_XL	FIFO_DATA_OUT_Z_H[3:0]
BDR_GY	FIFO_DATA_OUT_Z_H[7:4]

1. 在加速度计通道 2 存储到 FIFO 中时置为 1 的内部信号。

如表 85 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线器的批处理数据率。BDR_SHUB 无法通过专用寄存器进行配置。它与通过 SLV0_CONFIG 传感器集线器寄存器的 SHUB_ODR_[2:0] 位配置的传感器集线器 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。此外，时间戳数据还包含有关加速度计通道 2 和陀螺仪 EIS 通道的当前批处理状态的元信息。

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 86. FIFO 中的 CFG-Change 输出数据格式

数据	FIFO_DATA_OUT 寄存器
OP_MODE_XL	FIFO_DATA_OUT_X_L[2:0]
OP_MODE_G	FIFO_DATA_OUT_X_L[6:4]
LPF1_G_EN	FIFO_DATA_OUT_X_L[7]
LPF1_G_BW	FIFO_DATA_OUT_X_H[2:0]
FS_G[2:0]	FIFO_DATA_OUT_X_H[7:5]
LPF2_XL_EN	FIFO_DATA_OUT_Y_L[0]
HP_LPF2_XL_BW	FIFO_DATA_OUT_Y_L[3:1]
FS_XL	FIFO_DATA_OUT_Y_L[7:6]
BDR_SHUB	FIFO_DATA_OUT_Y_H[3:0]
G_EIS_FIFO_EN	FIFO_DATA_OUT_Y_H[4]
XL_DualC_BATCH_EN ⁽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_G	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. 在加速度计通道 2 存储到 FIFO 中时置为 1 的内部信号。
2. 在陀螺仪启动阶段（最长启动时间为 70 ms）结束时置为 0 的内部信号。

9.6 虚拟传感器

虚拟传感器分为以下四类：

1. 外部传感器，从传感器集线器接口读取
2. 计步器传感器
3. SFLP 生成的传感器
4. MLC 生成的传感器

9.6.1 外部传感器和 NACK 传感器

从传感器集线器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线器读取），则批处理数据率 (BDR_SHUB) 对应于 SLV0_CONFIG 寄存器中 SHUB_ODR_[2:0] 位域的当前值。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 位域的可能值。

表 87. BDR_SHUB

BDR_SHUB	BDR [Hz]
0000	未批处理或使用了外部触发信号
0001	1.875
0010	7.5 ⁽¹⁾
0011	15
0100	30
0101	60
0110	120
0111	240
1000	480

1. 可通过选择 000 之外的 SHUB_ODR_[2:0] 值以及使用 ODR 等于 7.5 Hz 的内部触发信号（加速度计或陀螺仪）来获取该值。

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 SLV0_CONFIG、SLV1_CONFIG、SLV2_CONFIG 和 SLV3_CONFIG 传感器集线器寄存器的 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数据批处理。

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 的顺序与传感器集线器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，传感器集线器将写入 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 到 3），并具有以下输出数据格式。

表 88. 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. 使能批处理计步器数据（将 EMB_FUNC_FIFO_EN_A 嵌入功能寄存器的 STEP_COUNTER_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 89. 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

9.6.3 SFLP 生成的传感器

专用传感器融合块 (SFLP) 旨在基于加速度计和陀螺仪数据处理来生成以下虚拟传感器：

- 游戏旋转向量，提供表示设备姿态的四元数
- 重力向量，提供表示重力方向的三维向量
- 陀螺仪偏差，提供表示陀螺仪偏差的三维向量

SFLP 生成的传感器只从 FIFO 读取，并有选择地进行使能：

- 通过将 EMB_FUNC_FIFO_EN_A 寄存器的 SFLP_GAME_FIFO_EN 位置为 1 来对游戏旋转向量进行批处理。
- 通过将 EMB_FUNC_FIFO_EN_A 寄存器的 SFLP_GRAVITY_FIFO_EN 位置为 1 来对重力向量进行批处理。
- 通过将 EMB_FUNC_FIFO_EN_A 寄存器的 SFLP_GBIAS_FIFO_EN 位置为 1 来对陀螺仪偏差进行批处理。

如果使能 FIFO 中的批处理，则 SFLP 生成的传感器将按照 SFLP 输出数据率存储到 FIFO 中。

下面列出了 FIFO 中 SFLP 生成的传感器的格式：

- 游戏旋转向量：X、Y 和 Z 轴（四元数的向量部分）以半精度浮点格式存储，其中 w（四元数的标量部分）在读取 FIFO 中的数据后用软件计算，因为游戏旋转向量为单位四元数。
- 重力向量：X、Y 和 Z 轴以 16 位二进制补码存储，灵敏度为 $\pm 2 g$ 。
- 陀螺仪偏差：X、Y 和 Z 轴以 16 位二进制补码存储，灵敏度为 $\pm 125 dps$ 。

9.6.4 MLC 生成的传感器

以下机器学习内核（由 MLC 生成）虚拟传感器可存储到 FIFO 中：

- 结果
- 滤波器
- 特征，包括窗口特征和递归特征

要将 MLC 生成的传感器存储到 FIFO 中，必须通过将 EMB_FUNC_EN_A 寄存器的 MLC_BEFORE_FSM_EN 位或 EMB_FUNC_EN_B 寄存器的 MLC_EN 位置为 1 来使能 MLC 块。

通过将 EMB_FUNC_FIFO_EN_A 寄存器的 MLC_FIFO_EN 位置为 1 来使能 MLC 结果的批处理。

MLC 结果包含相应 MLCx_SRC 寄存器的信息，并在相应 MLCx_SRC 发生更改时存储到 FIFO 中。

使用 ST 提供的 MLC 配置工具之一，有选择地使能 MLC 滤波器的批处理。此外，必须将 EMB_FUNC_FIFO_EN_B 寄存器的 MLC_FILTER_FEATURE_FIFO_EN 位置为 1，以全局使能将 MLC 滤波器或特征存储到 FIFO 中。

MLC 滤波器以与 MLC 输出数据率（MLC_ODR 位）相等的速率存储到 FIFO 中。如果滤波器应用到所需传感器的 X、Y 和 Z 轴，则每个轴有一个字存储到 FIFO 中。如果滤波器应用到所需传感器的范数，则有一个字存储到 FIFO 中。

使用 ST 提供的 MLC 配置工具之一，有选择地使能 MLC 特征的批处理。此外，必须将 EMB_FUNC_FIFO_EN_B 寄存器的 MLC_FILTER_FEATURE_FIFO_EN 位置为 1，以全局使能将 MLC 滤波器或特征存储到 FIFO 中。

MLC 窗口特征在每个窗口结束时存储到 FIFO 中。

MLC 递归特征（如 MLC 滤波器）以与 MLC 输出数据率 (MLC_ODR) 相等的速率存储到 FIFO 中。

以下表格列出了 FIFO 中的 MLC 结果、特征和滤波器的格式。

表 90. FIFO 中的 MLC 结果

数据	FIFO_DATA_OUT 寄存器
MLCx_SRC	FIFO_DATA_OUT_X_L
MLC_SRC 的索引 ⁽¹⁾	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

1. MLCx_SRC 寄存器的索引为 0 到 3（例如，MLC1_SRC 的索引为 0）。

表 91. FIFO 中的 MLC 滤波器或特征

数据	FIFO_DATA_OUT 寄存器
VALUE[7:0] ⁽¹⁾	FIFO_DATA_OUT_X_L
VALUE[15:8] ⁽¹⁾	FIFO_DATA_OUT_X_H
IDENTIFIER[7:0] ⁽²⁾	FIFO_DATA_OUT_Y_L
IDENTIFIER[15:8] ⁽²⁾	FIFO_DATA_OUT_Y_H
保留	FIFO_DATA_OUT_Z_L
保留	FIFO_DATA_OUT_Z_H

1. 该值用半精度浮点数表示。

2. 滤波器和特征标识符在由 ST 的 MLC 配置工具生成的配置文件中指示。

9.7 FIFO 模式

通过 FIFO_CTRL4 寄存器的 FIFO_MODE_[2:0] 位域, LSM6DSV16X FIFO 缓冲器可配置为七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 模式、FIFO 模式、Continue 模式、Continue-FIFO 模式、Bypass-Continue 模式、Bypass-FIFO 模式和 ContinueWTM-Full 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] 位置为 000 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 001 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_8 位置为 1, 不再有数据存储到 FIFO 缓冲器中。可以通过读取所有 FIFO_DATA_OUT 寄存器 (78h 到 7Eh) 并达到 FIFO_STATUS1 和 FIFO_STATUS2 寄存器 DIFF_FIFO_[8:0] 位指定的次数来恢复数据。

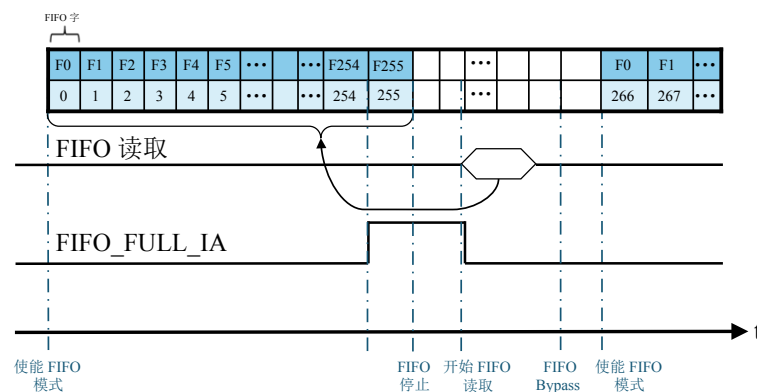
利用 FIFO_STATUS2 寄存器的 FIFO_WTM_IA 位, 如果应用要求 FIFO 中采样数较低, 则数据还可在达到阈值 (FIFO_CTRL1 寄存器中的 WTM[7:0]) 时恢复。

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 寄存器中的 WTM_[7:0] 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_[7:0] 值, 那么 FIFO_STATUS2 寄存器的 FIFO_FULL_IA 位会置为高电平。

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 26. 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在 FIFO 中。这些条件下, FIFO 缓冲器中可存储 256 个采样 (压缩算法禁用)。在标记为 254 后, FIFO_STATUS2 寄存器的 FIFO_FULL_IA 位置为高电平, 以通知 FIFO 缓冲器将在下一次 FIFO 写操作时达到全满。在 FIFO 已满 (FIFO_DIFF_8 = 1) 后, 数据采集停止。

图 26. FIFO 模式 (STOP_ON_WTM = 0)



9.7.3 Continue 模式

在 Continue 模式下，FIFO 连续填充。当缓冲器已满时，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。

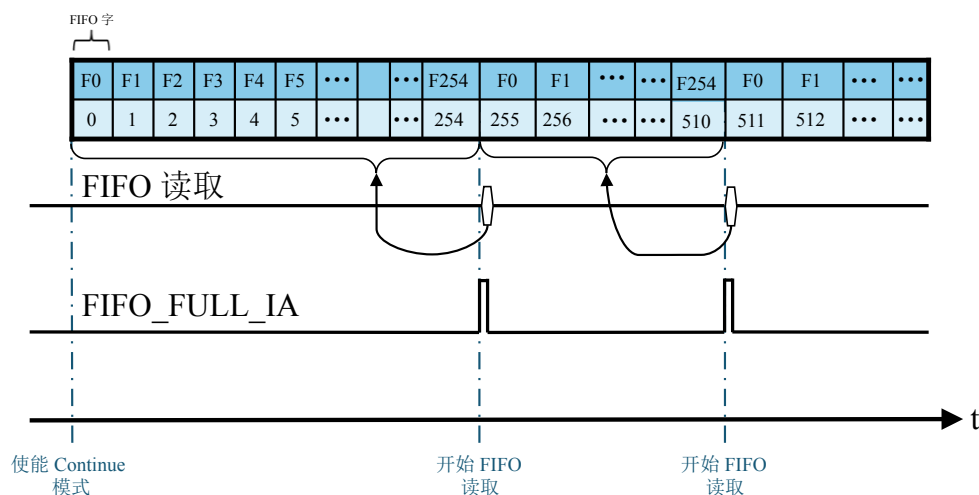
按照这些步骤进行 Continue 模式配置（如果加速度计/陀螺仪数据准备就绪被用作 FIFO 触发）：

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] 位域置为 110 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 到 7Eh）寄存器获取 FIFO_STATUS1 和 FIFO_STATUS2 寄存器的 DIFF_FIFO_[8:0] 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 寄存器中的 WTM_[7:0]）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 寄存器中的 WTM_[7:0] 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_[7:0] 值，那么 FIFO_STATUS2 寄存器的 FIFO_FULL_IA 位会置为高电平。

图 27. 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于 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 255。

图 27. Continue 模式



9.7.4 Continue-FIFO 模式

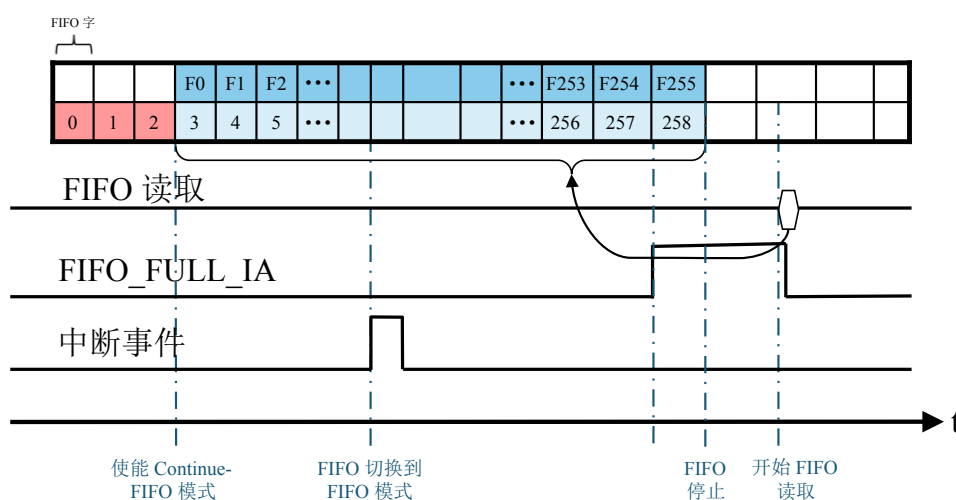
此模式是先前所述的 Continue 模式和 FIFO 模式的组合。在 Continue-FIFO 模式中，FIFO 缓冲器开始工作于 Continue 模式，当发生事件条件时切换为 FIFO 模式。

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 位必须置为 1。

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

图 28. 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 011 来使能 FIFO Continue-FIFO 模式。

在 Continue-FIFO 模式下，FIFO 缓冲器会继续填入数据。当 FIFO 在下次 FIFO 写操作时已满或溢出时，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 寄存器中的 WTM_[7:0] 位的值。这种情况下，如果下次 FIFO 写操作时 FIFO 中的采样数达到或超过 WTM_[7:0] 值，那么 FIFO_STATUS2 寄存器的 FIFO_FULL_IA 位会置为高电平。

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。

Continue-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 缓冲器已满并停止时读取 FIFO 内容。

9.7.5 Bypass-Continue 模式

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

事件条件可为以下之一：

- 单击：必须配置事件检测，并且 MD2_CFG 寄存器的 INT2_SINGLE_TAP 位必须置为 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 模式变为 Continue 模式，并维持该模式，直到 Bypass 模式被置位。

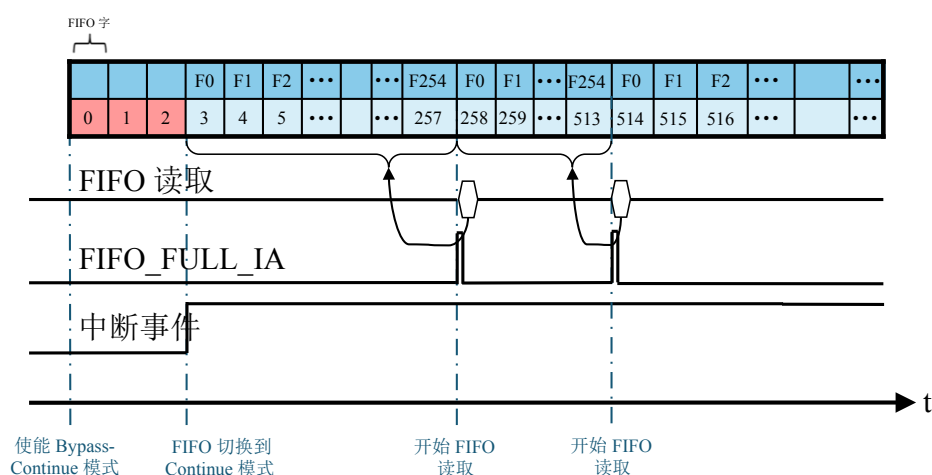
按照以下步骤进行 Bypass-Continue 模式配置（如果加速度计/陀螺仪数据准备就绪被用作 FIFO 触发）：

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 100 来使能 FIFO Bypass-Continue 模式。

当出现触发条件且缓冲器切换至 Continue 模式时，FIFO 缓冲器继续填充。当下一个要存储的数据集会使 FIFO 全满或溢出时，FIFO_FULL_IA 位置为高电平。

Bypass-Continue 可用在产生所配置中断时启动采集。

图 29. Bypass-Continue 模式



9.7.6 Bypass-FIFO 模式

此模式是先前所述的 **Bypass** 模式和 **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 位必须置为 1。

Bypass-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。

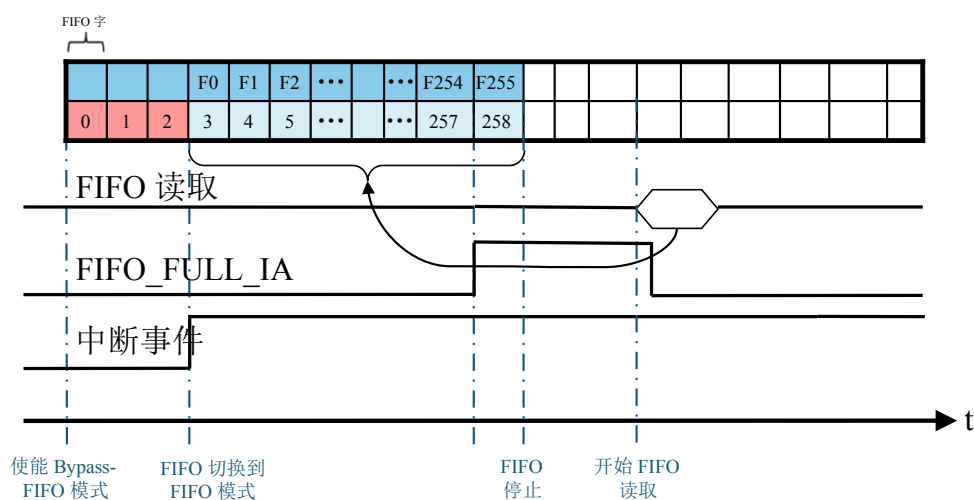
按照以下步骤进行 **Bypass-FIFO** 模式配置（如果加速度计/陀螺仪数据准备就绪被用作 FIFO 触发）：

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 111 来使能 FIFO **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会使 FIFO 全满或溢出时，FIFO_FULL_IA 位置为高电平且 FIFO 停止。

Bypass-FIFO 可用来分析生成中断的采样历史。

图 30. Bypass-FIFO 模式



9.7.7 ContinueWTM-Full 模式

此模式与前面所述的 Continue-FIFO 模式类似，但增加了以下特性：

- 在 Continue 模式下，FIFO 空间大小自动受所选 FIFO 阈值（FIFO_CTRL1 寄存器的 WTM_[7:0] 位域）限制，因此称为“ContinueWTM”模式。在此模式下，FIFO 已满事件被内部屏蔽。
- 在 FIFO 模式下，FIFO 空间大小不再受所选 FIFO 阈值限制，因此称为“Full”模式。在此模式下，FIFO 已满事件不再被内部屏蔽。

在 ContinueWTM-Full 模式下，FIFO 缓冲器开始工作于 ContinueWTM 模式，当发生事件条件时切换为 Full 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 位必须置为 1。

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

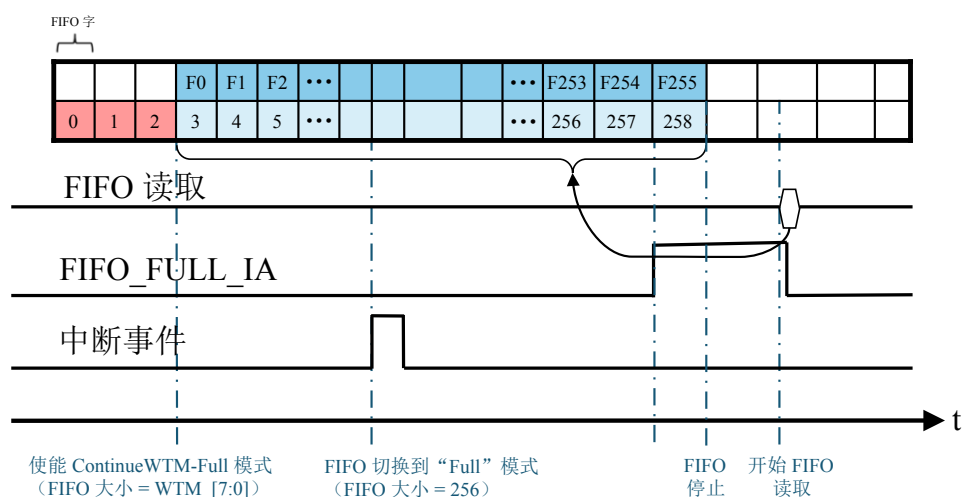
按照以下步骤进行 ContinueWTM-Full 模式配置（如果加速度计/陀螺仪数据准备就绪被用作 FIFO 触发）：

- 配置先前所述的其中一个事件。
- 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）。
- 将 FIFO_CTRL4 寄存器中的 FIFO_MODE_[2:0] 位域置为 010 来使能 FIFO ContinueWTM-Full 模式。

在 ContinueWTM-Full 模式下，FIFO 缓冲器会继续填入数据。当 FIFO 在下次 FIFO 写操作时已满或溢出时（如上文所述，FIFO 空间大小自动受 FIFO_CTRL1 寄存器中的 WTM_[7:0] 位域的值限制），FIFO_FULL_IA 位不会变为高电平，因为其被内部屏蔽。发生触发事件时，FIFO 缓冲器的大小不再受 FIFO_CTRL1 寄存器中 WTM_[7:0] 位域的值限制，继续填入数据至已满状态，然后停止收集数据。

ContinueWTM-Full 模式可用于分析已生成中断的采样和生成中断后的采样的历史。标准操作是在 FIFO 模式已触发、FIFO 缓冲器已满并停止时读取 FIFO 内容。

图 31. ContinueWTM-Full 模式



9.8 从 FIFO 获取数据

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 中获取数据的一种方法如下：

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_[8:0] 位域中。
2. 对于 FIFO 中的每个字，读取 FIFO 字（标签和输出数据）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_[8:0] 位等于 0），可以恢复全部 FIFO 内容。

FIFO 为空时可对其进行读取。在这种情况下，FIFO 字用特定空标签来标记。

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 到 FIFO_DATA_OUT_TAG 的环行功能。在这种情况下，建议按以下步骤从 FIFO 中恢复数据：

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_[8:0] 位域中。
2. 通过多重运算（即 $(\text{DIFF_FIFO} + N) * 7$ 个字节）读取 DIFF_FIFO + N 个字，选择 N 时应确保 FIFO 已为空。
3. 如果从 FIFO 读取的数据不包含用空标签标记的数据，则额外读取 N 个采样以清空 FIFO。

9.9 FIFO 水印阈值

FIFO 阈值是 LSM6DSV16X 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 寄存器中的 WTM_[7:0] 位包含水印阈值。WTM_[7:0] 位域的分辨率为 7 字节，相当于一个完整的 FIFO 字。因此，用户能够在 0 到 255 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 32. FIFO 阈值 (STOP_ON_WTM = 0)

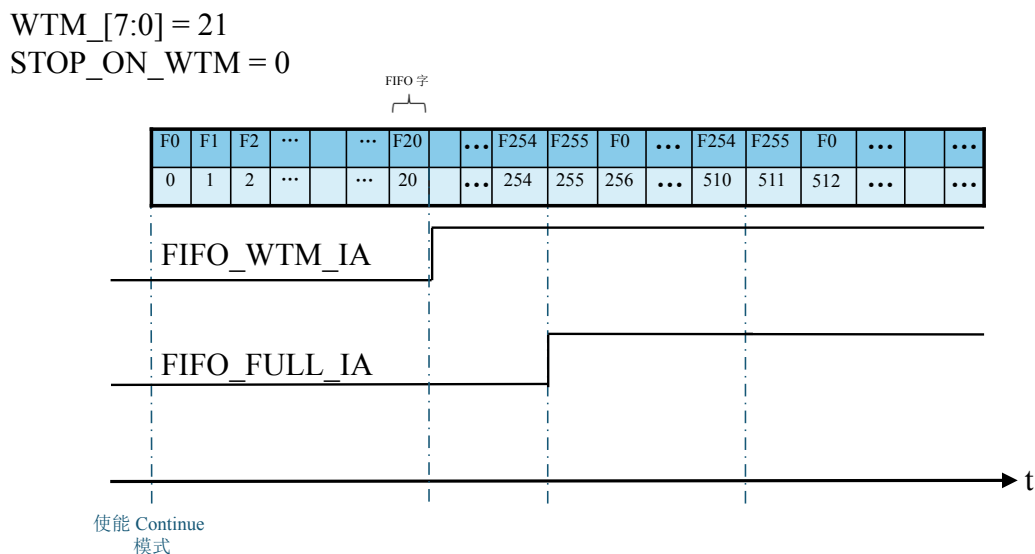


图 32. FIFO 阈值 (STOP_ON_WTM = 0) 显示了只存储加速度计（或陀螺仪）数据时，FIFO 阈值水平使用的示例。FIFO_CTRL2 寄存器中 STOP_ON_WTM 位置为 0。利用 WTM_[7:0] 位，阈值设置为 21。达到 21 后（FIFO 中有 21 个字），FIFO_STATUS2 寄存器的 FIFO_WTM_IA 位升高。由于 STOP_ON_WTM 位置为 0，FIFO 不会在第 21 个数据组时停止，而是会继续存储数据，直至 FIFO_FULL_IA 标记置为高电平。

图 33. FIFO 模式下的 FIFO 阈值 (STOP_ON_WTM = 1)

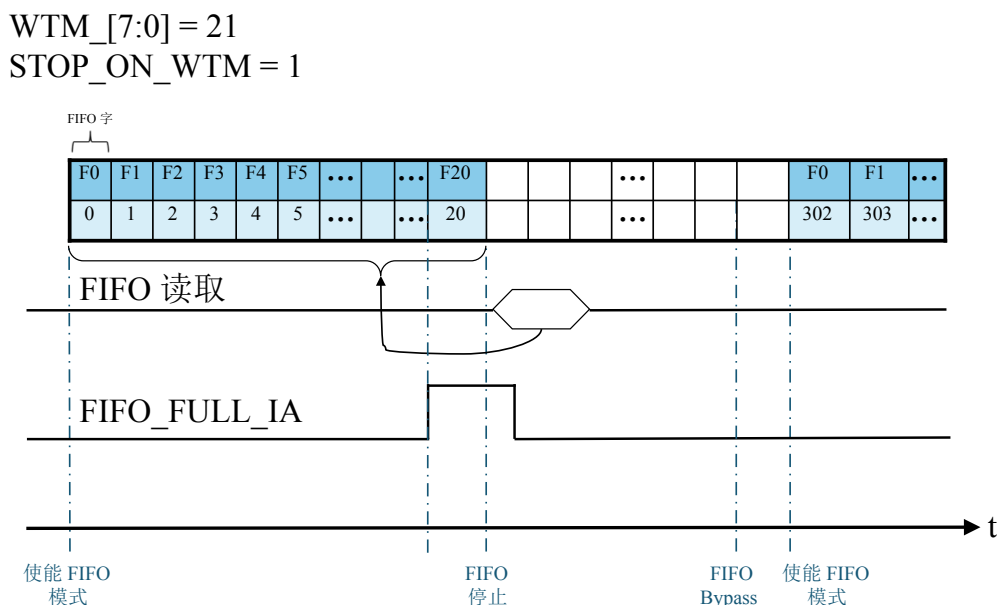


图 33. FIFO 模式下的 FIFO 阈值 (STOP_ON_WTM = 1) 显示了 FIFO 模式下 FIFO 阈值使用的示例，其中 FIFO_CTRL2 寄存器中 STOP_ON_WTM 位置为 1。本例中只存储了加速度计（或陀螺仪）数据。利用 WTM_[7:0] 位，阈值设置为 21，并定义当前 FIFO 空间大小。在 FIFO 模式下，数据保存在 FIFO 缓冲器，直至 FIFO 已满。在 FIFO 中保存的下一个数据生成 FIFO 已满或溢出条件时，FIFO_STATUS2 寄存器的 FIFO_FULL_IA 位升高。当 FIFO 已满时，FIFO_STATUS2 寄存器的 FIFO_WTM_IA 位升高。

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

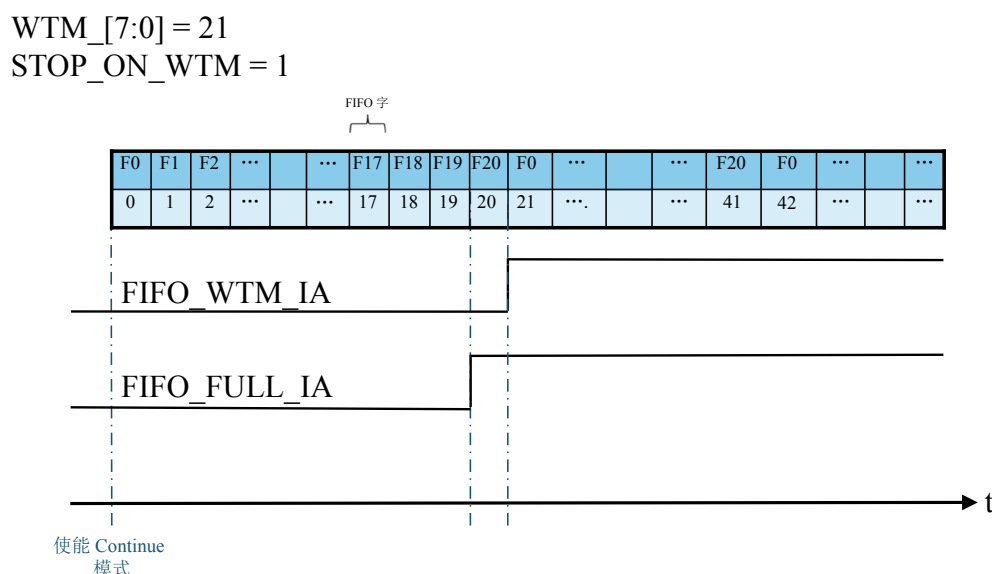


图 34. Continue 模式下的 FIFO 阈值 (STOP_ON_WTM = 1) 显示了 Continue 模式下 FIFO 阈值使用的示例，其中 FIFO_CTRL2 寄存器的 STOP_ON_WTM 位置为 1。本例中只存储了加速度计（或陀螺仪）数据。利用 WTM_[7:0] 位，阈值设置为 21。FIFO 中存储的下一个数据使 FIFO 已满时，FIFO_STATUS2 寄存器的 FIFO_FULL_IA 位升高。当 FIFO 已满时，FIFO_STATUS2 寄存器的 FIFO_WTM_IA 位升高。如果不从 FIFO 恢复数据，新数据（标记为采样 21）将覆盖 FIFO 中的旧数据（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。

可通过将 EMB_FUNC_EN_B 嵌入功能寄存器中的 FIFO_COMPR_EN 位和 FIFO_CTRL2 寄存器中的 FIFO_COMPR_RT_EN 位置 1 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) 可单独配置，但如果加速度计和/或陀螺仪以大于 1920 Hz 的速率进行批处理，则不支持压缩算法。

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 LSB 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 标签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 $\text{diff}(i)$ 、 $\text{diff}(i - 1)$ 和 $\text{diff}(i - 2)$
- 2xC 数据写入，其中包含 $\text{diff}(i - 1)$ 和 $\text{diff}(i - 2)$
- NC_T_2 数据写入，其中包含 $\text{data}(i - 2)$

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 位的复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 92. FIFO 压缩标签和相关数据

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

如表 92 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 93. 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 94. FIFO 中的 3xC 压缩数据输出数据格式

数据	公式
diffx(i - 2)	5bit_signed(FIFO_DATA_OUT_X[4:0])
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)	5bit_signed(FIFO_DATA_OUT_Y[4:0])
diffy(i - 1)	5bit_signed(FIFO_DATA_OUT_Y[9:5])
diffz(i - 1)	5bit_signed(FIFO_DATA_OUT_Y[14:10])
diffx(i)	5bit_signed(FIFO_DATA_OUT_Z[4:0])
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。切换按器件配置修改进行管理。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 95. 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 启用 FIFO 压缩的 CFG-Change 传感器

在将配置修改应用于器件时，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_COMPRT_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 96. 使能了 FIFO 压缩的器件配置修改示例

时间	FIFO_COMPRT_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 UNCOMPR_RATE_[1:0] 位域保证以特定周期（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。UNCOMPR_RATE_[1:0] 位域配置压缩算法，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 97. UNCOMPR_RATE 配置

UNCOMPR_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_COMPRT_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 98. 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，没有任何延迟。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 标签的未压缩字，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器：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 位域
3. 混合法，基于 TAG_CNT 位域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：

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

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掉电模式，则温度传感器关闭。

温度传感器的最大输出数据率为 60 Hz，其值取决于加速度计和陀螺仪传感器如何配置：

- 如果陀螺仪处于掉电模式：
 - 如果加速度计配置为低功耗模式且其 ODR 低于 60 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，温度数据率均等于 60 Hz。
- 如果陀螺仪不是处于掉电模式，则无论加速度计和陀螺仪配置是什么，温度数据率均等于 60 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 CTRL4 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) 缓冲器）。

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。表中所列值是在理想器件校准的假设下给出的（即，无偏移、无增益误差等）。

表 99. 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

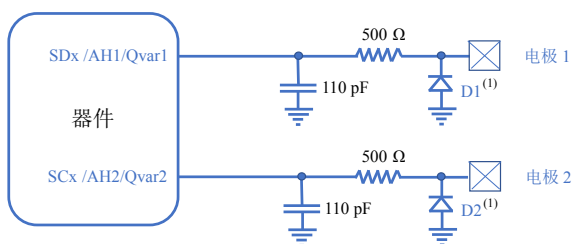
11 模拟集线器和 Qvar 功能

LSM6DSV16X 嵌入了模拟集线器 (AH) 链，能够连接外部模拟输入并将其转换为数字信号进行处理。该模拟链可用于连接外部电极和电子信号调节电路，从而实现 Qvar（电荷变化检测）功能。外部电极必须连接到引脚 2 (SDx/AH1/Qvar1) 和/或引脚 3 (SCx/AH2/Qvar2)，因此使用 Qvar 时，I²C 主接口（模式 2）和辅助 SPI（模式 3）不可用。图 35 提供了测试电路示例。

提示

Qvar 功能仅在 I²C 接口的模式 1 连接模式下可用。

图 35. Qvar 外部连接



(1) 这里使用的是 ST 的产品 ST ESDALCL5-1BM2，但也可以使用其他具有类似功能的 ESD 二极管来代替。

在 LSM6DSV16X 中，模拟集线器/Qvar 具有专用通道，可通过将 CTRL7 寄存器的 AH_QVAR_EN 位置为 1 来激活。

提示

当模拟集线器/Qvar 通道使能时，加速度计传感器必须设置为高性能模式或正常模式。

模拟集线器/Qvar 数据准备就绪信号由 STATUS_REG 寄存器的 AH_QVARDA 位表示。通过将 CTRL7 寄存器的 INT2_DRDY_AH_QVAR 位置为 1，可将该信号驱动至 INT2 引脚。

模拟集线器/Qvar 数据在 AH_QVAR_OUT_L 和 AH_QVAR_OUT_H 寄存器中以 16 位字（二进制补码）的形式提供，频率固定为 240 Hz（典型值）。它们还可通过 MLC/FSM 逻辑处理。

可通过正确地设置 CTRL7 寄存器中的 AH_QVAR_C_ZIN_[1:0] 位来选择模拟集线器/Qvar 缓冲器的等效输入阻抗（00: 2.4 GΩ（默认值）；01: 730 MΩ；10: 300 MΩ；11: 235 MΩ）。

有关 Qvar 的更多详细信息，请参见 www.st.com 上的应用笔记 AN5755。

12 自检功能

嵌入式自检功能允许在不移动设备的情况下对其自身功能进行检查。

12.1 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，模拟一定的输入加速度。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满量程。

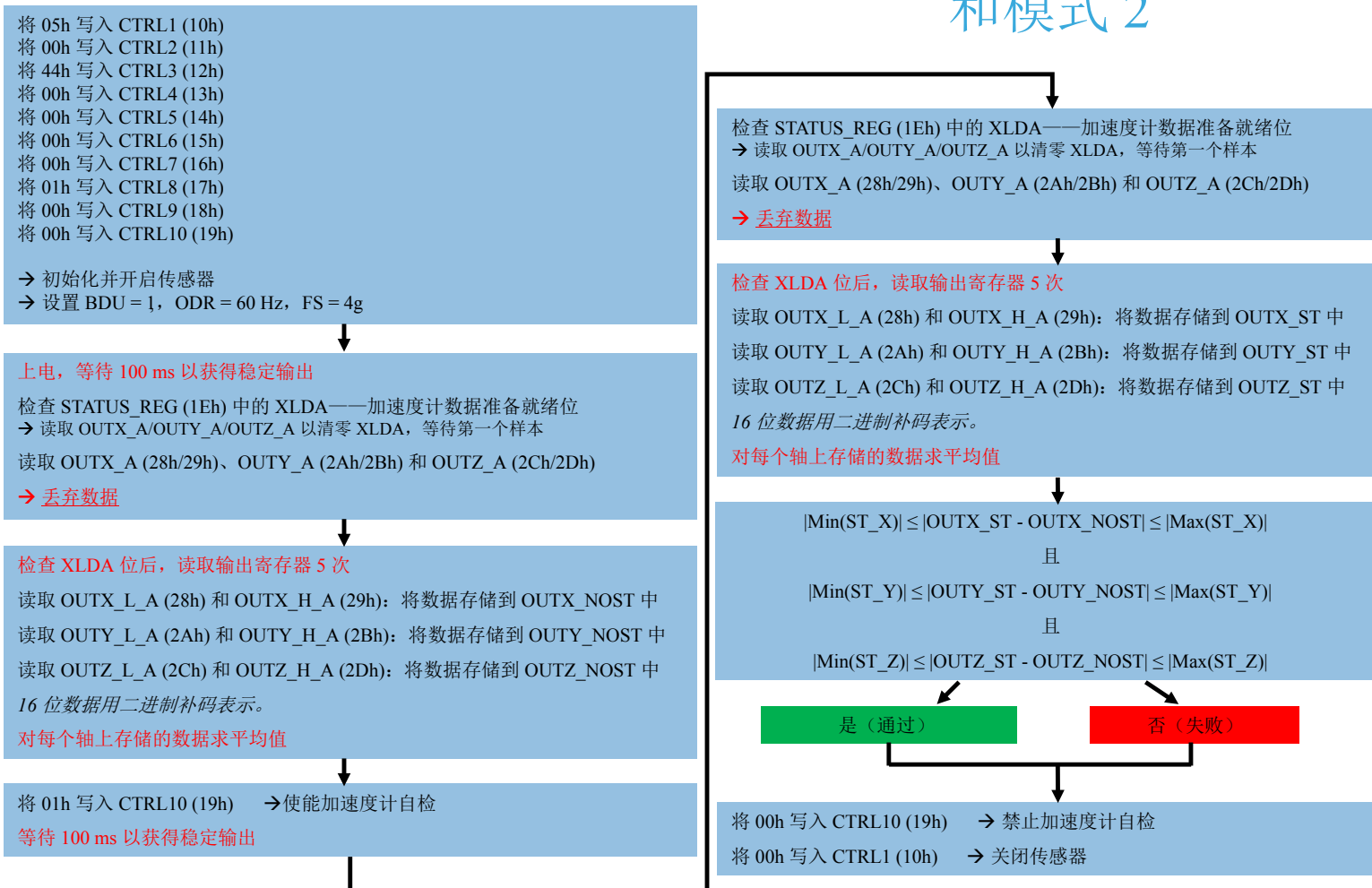
当器件配置为模式 1 或模式 2 时，只能从主接口配置加速度计自检功能。当 CTRL10 寄存器的 ST_XL_[1:0] 位置为 00 时，自检功能关闭。当 ST_XL_[1:0] 位置为 01（正符号自检）或 10（负符号自检）时，自检功能使能。

当加速度计自检功能激活时，传感器输出电平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流程如图 36. 加速度计自检流程 (UI) 中所示。

图 36. 加速度计自检流程 (UI)

注意：必须通过主 I²C/SPI/I³C 接口执行
该流程中的所有读/写操作



12.2 OIS 链接通后的加速度计自检 (UI) – 模式 3

如果使用了辅助 SPI 且 UI 链和 OIS 链接通，则必须从主接口通过 CTRL10 寄存器的 ST_XL_[1:0] 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）。

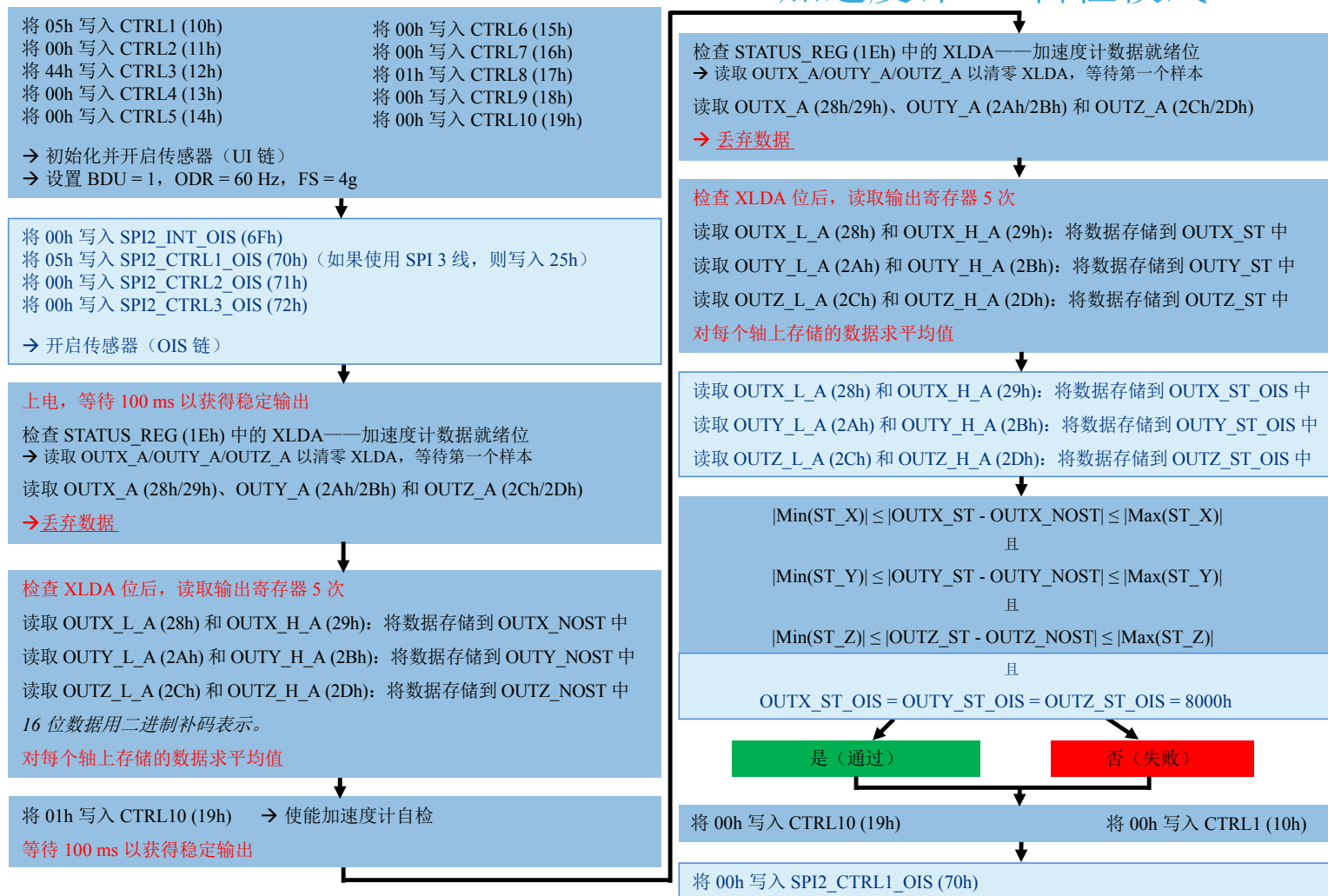
当 OIS 链连通时，UI 链上的推荐加速度计自检流程如图 37. OIS 链接通时的加速度计自检流程 (UI) 中所示。

图 37. OIS 链接通时的加速度计自检流程 (UI)

注意:

- 必须通过主 I²C/SPI 接口执行所有黑色字体的读/写操作
- 必须通过辅助 SPI 接口执行所有蓝色字体的读写操作

UI 和 OIS 链均接通时的 加速度计 UI 自检模式 3



12.3 加速度计自检 (OIS) – 模式 3

如果使用辅助 SPI 且 UI 链断开，可通过辅助 SPI 接口将 SPI2_INT_OIS 寄存器的 ST_XL_OIS_[1:0] 位置位，从而使能 OIS 链上的加速度计自检功能。当 ST_XL_OIS_[1:0] 位置为 00 时，自检功能关闭。当 ST_XL_OIS_[1:0] 位置为 01（正符号自检）或 10（负符号自检）时，自检功能使能。

当 UI 链断开时，OIS 链上的完整加速度计自检流程如图 38. 加速度计自检流程 (OIS) 中所示。仅当 UI 链断开（CTRL1 寄存器中的 ODR_XL_[3:0] = 0000 且 CTRL2 寄存器中的 ODR_G_[3:0] = 0000）时，才能执行此流程。

图 38. 加速度计自检流程 (OIS)

注意:

- 必须通过辅助 SPI 接口执行该流程中的所有读/写操作
- 只能在 UI 链断开 (CTRL1 中的 ODR_XL_[3:0] = 0000) 时执行该流程

将 00h 写入 SPI2_INT_OIS (6Fh)
将 05h 写入 SPI2_CTRL1_OIS (70h) (如果使用 SPI 3 线, 则写入 25h)
将 00h 写入 SPI2_CTRL2_OIS (71h)
将 01h 写入 SPI2_CTRL3_OIS (72h)

→ 初始化并开启传感器
→ FS = 4g (ODR 为 7680 Hz)

上电, 等待 100 ms 以获得稳定输出

检查 SPI2_STATUS_REG_OIS (1Eh) 中的 XLDA——加速度计数据准备就绪位
→ 读取 OUTX_A/OUTY_A/OUTZ_A 以清零 XLDA, 等待第一个样本
读取 OUTX_A (28h/29h)、OUTY_A (2Ah/2Bh) 和 OUTZ_A (2Ch/2Dh)

→ 丢弃数据

检查 XLDA 位后, 读取输出寄存器 5 次

读取 OUTX_L_A (28h) 和 OUTX_H_A (29h): 将数据存储到 OUTX_NOST_OIS 中
读取 OUTY_L_A (2Ah) 和 OUTY_H_A (2Bh): 将数据存储到 OUTY_NOST_OIS 中
读取 OUTZ_L_A (2Ch) 和 OUTZ_H_A (2Dh): 将数据存储到 OUTZ_NOST_OIS 中
16 位数据用二进制补码表示。

对每个轴上存储的数据求平均值

OIS链接通且 UI 链断开时的 加速度计 OIS 自检模式 3

将 01h 写入 INT_OIS (6Fh) → 使能加速度计自检
等待 100 ms

检查 SPI2_STATUS_REG_OIS (1Eh) 中的 XLDA——加速度计数据准备就绪位
→ 读取 OUTX_A/OUTY_A/OUTZ_A 以清零 XLDA, 等待第一个样本
读取 OUTX_A (28h/29h)、OUTY_A (2Ah/2Bh) 和 OUTZ_A (2Ch/2Dh)
→ 丢弃数据

检查 XLDA 位后, 读取输出寄存器 5 次

读取 OUTX_L_A (28h) 和 OUTX_H_A (29h): 将数据存储到 OUTX_ST_OIS 中
读取 OUTY_L_A (2Ah) 和 OUTY_H_A (2Bh): 将数据存储到 OUTY_ST_OIS 中
读取 OUTZ_L_A (2Ch) 和 OUTZ_H_A (2Dh): 将数据存储到 OUTZ_ST_OIS 中
16 位数据用二进制补码表示。

对每个轴上存储的数据求平均值

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

是 (通过)

否 (失败)

将 00h 写入 SPI2_INT_OIS (6Fh) → 禁止加速度计自检
将 00h 写入 SPI2_CTRL1_OIS (70h) → 关闭传感器

12.4 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。当自检功能激活时，会在 ASIC 前端的输入上仿真等效 Coriolis 信号，之后传感器输出会发生变化。

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 CTRL10 寄存器的 ST_G_[1:0] 位置为 00 时，自检功能关闭。当 ST_G_[1:0] 位置为 01（正符号自检）或 10（负符号自检）时，自检功能使能。

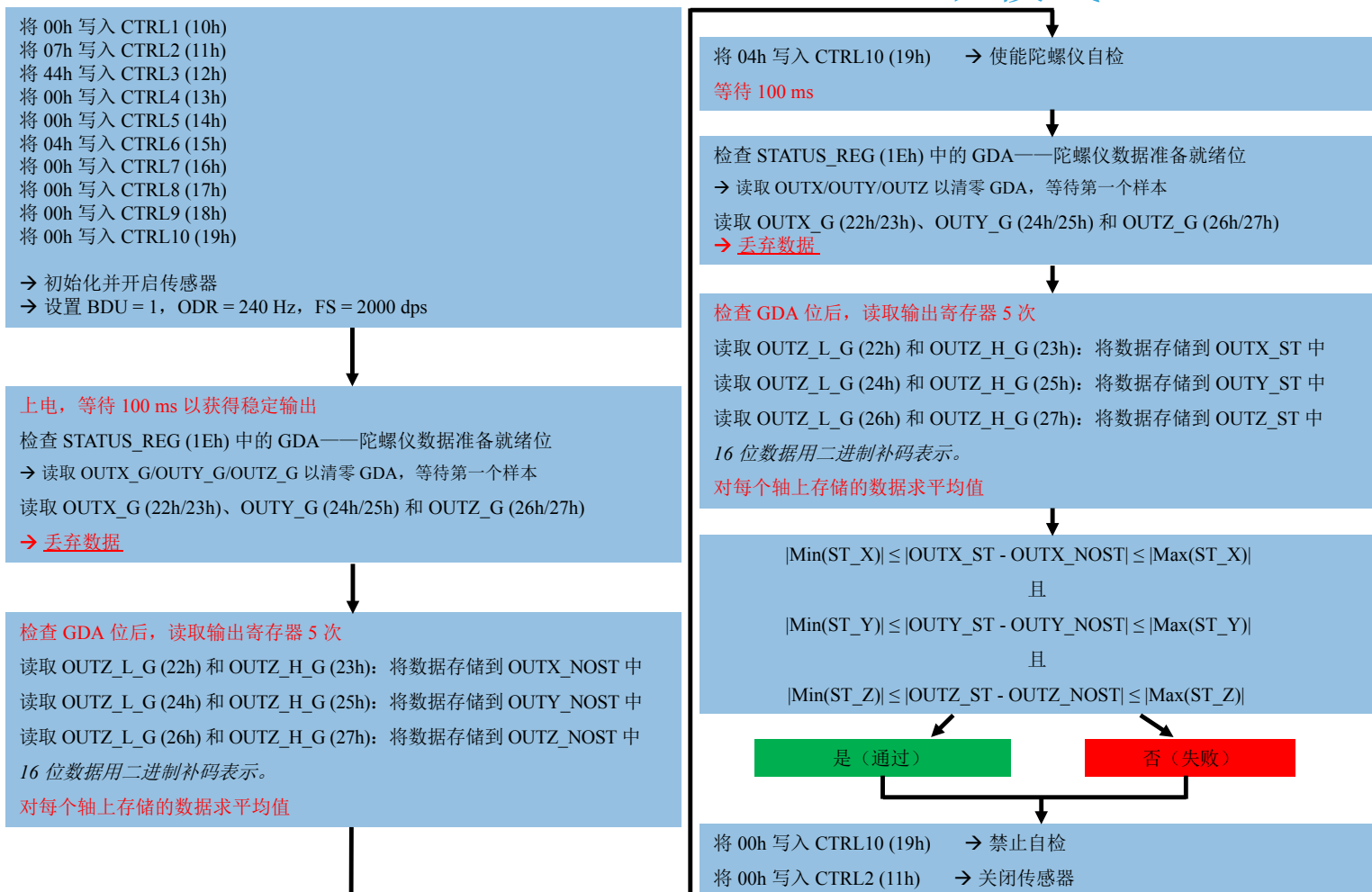
当陀螺仪自检功能激活时，传感器输出电平由作用在传感器上的角速度和静电测试力的代数和给出。

模式 1 或模式 2 下的完整陀螺仪自检流程如图 39. 陀螺仪自检流程 (UI) 中所示。

图 39. 陀螺仪自检流程 (UI)

注意：必须通过主 I²C/SPI/I³C 接口执行
该流程中的所有读/写操作

陀螺仪 UI 自检模式 1 和模式 2



12.5 OIS 链接通后的陀螺仪自检 (UI) – 模式 3

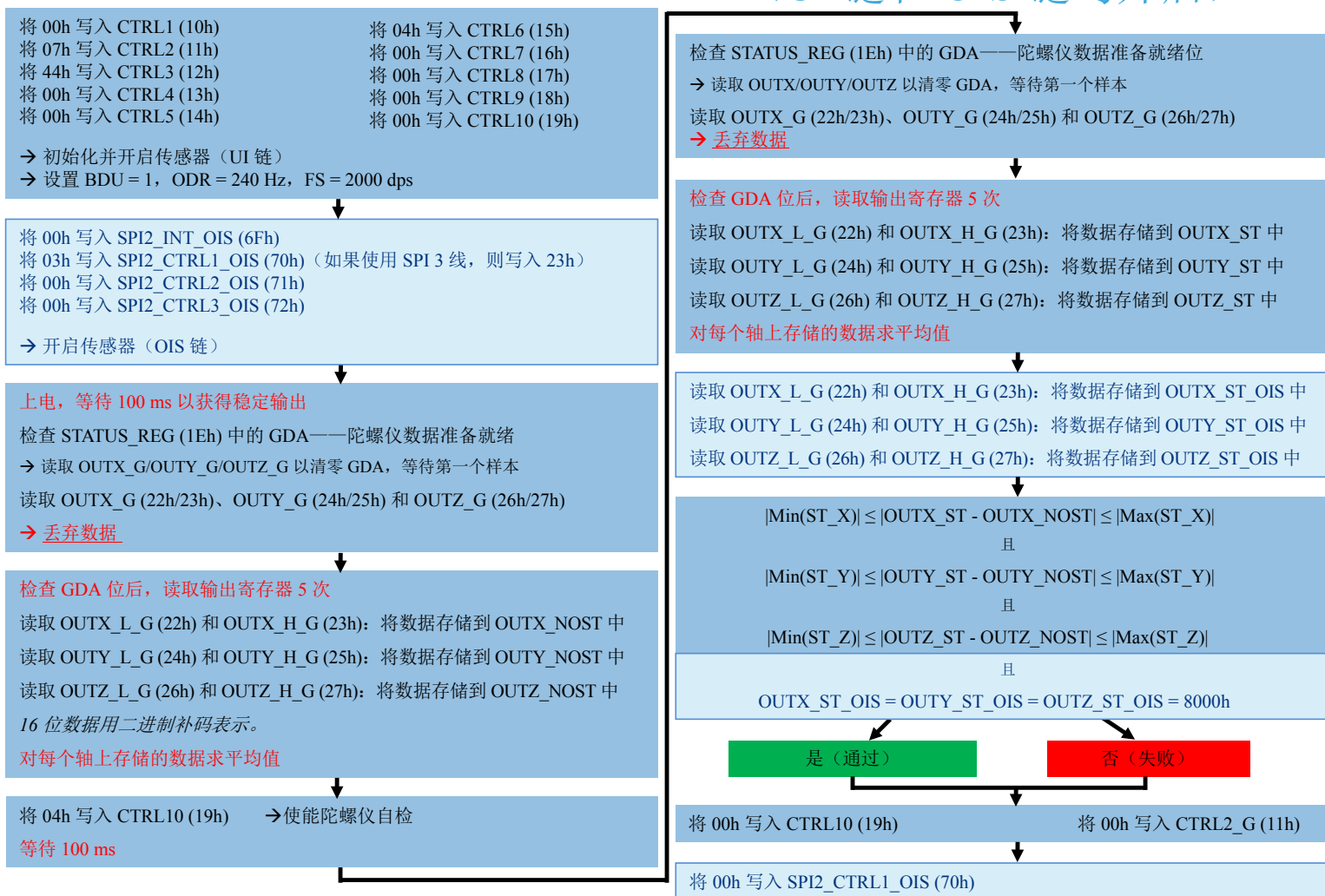
如果使用了辅助 SPI 且 UI 链和 OIS 链接通，则必须从主接口通过 CTRL10 寄存器的 ST_G_[1:0] 位使能陀螺仪自检功能。不能同时从两个接口使能（禁止条件）。

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)

注意:

- 必须通过主 I²C/SPI 接口执行所有黑色字体的读/写操作
- 必须通过辅助 SPI 接口执行所有蓝色字体的读/写操作



12.6 陀螺仪自检 (OIS) – 模式 3

如果使用辅助 SPI 且 UI 链断开，可通过辅助 SPI 接口将 SPI2_INT_OIS 寄存器的 ST_G_OIS_[1:0] 位置位，从而使能 OIS 链上的陀螺仪自检功能。当 ST_G_OIS_[1:0] 位置为 00 时，自检功能关闭。当 ST_G_OIS_[1:0] 位置为 01（正符号自检）或 11（负符号自检）时，自检功能使能。

当 UI 链断开时，OIS 链上的完整陀螺仪自检流程如图 41. 陀螺仪自检流程 (OIS) 中所示。仅当 UI 链断开（CTRL1 寄存器中的 ODR_XL_[3:0] = 0000 且 CTRL2 寄存器中的 ODR_G_[3:0] = 0000）时，才能执行此流程。

图 41. 陀螺仪自检流程 (OIS)

注意:

- 必须通过辅助 SPI 接口执行该流程中的所有读/写操作
- 只能在 UI 链断开 (CTRL2 中的 ODR_G[3:0] = 0000) 时执行该流程

将 00h 写入 SPI2_INT_OIS (6Fh)
将 03h 写入 SPI2_CTRL1_OIS (70h) (如果使用 SPI 3 线, 则写入 23h)
将 04h 写入 SPI2_CTRL2_OIS (71h)
将 00h 写入 SPI2_CTRL3_OIS (72h)

→ 初始化并开启传感器
→ FS = 2000 dps (ODR 为 7680 Hz)

上电, 等待 100 ms 以获得稳定输出

检查 SPI2_STATUS_REG_OIS (1Eh) 中的 GDA——陀螺仪数据准备就绪位
→ 读取 OUTX_G/OUTY_G/OUTZ_G 以清零 GDA, 等待第一个样本
读取 OUTX_G (22h/23h)、OUTY_G (24h/25h) 和 OUTZ_G (26h/27h)

→ 丢弃数据

检查 GDA 位后, 读取输出寄存器 5 次

读取 OUTX_L_G (22h) 和 OUTX_H_G (23h): 将数据存储到 OUTX_NOST_OIS 中
读取 OUTY_L_G (24h) 和 OUTY_H_G (25h): 将数据存储到 OUTY_NOST_OIS 中
读取 OUTZ_L_G (26h) 和 OUTZ_H_G (27h): 将数据存储到 OUTZ_NOST_OIS 中
16 位数据用二进制补码表示。

对每个轴上存储的数据求平均值

OIS 链接通且 UI 链断开时的 陀螺仪 OIS 自检模式 3

将 04h 写入 SPI2_INT_OIS (6Fh) → 使能陀螺仪自检
等待 100 ms

检查 SPI2_STATUS_REG_OIS (1Eh) 中的 GDA——陀螺仪数据准备就绪位
→ 读取 OUTX/OUTY/OUTZ 以清零 GDA, 等待第一个样本
读取 OUTX_G (22h/23h)、OUTY_G (24h/25h) 和 OUTZ_G (26h/27h)
→ 丢弃数据

检查 XLDA 位后, 读取输出寄存器 5 次

读取 OUTX_L_G (22h) 和 OUTX_H_G (23h): 将数据存储到 OUTX_ST_OIS 中
读取 OUTY_L_G (24h) 和 OUTY_H_G (25h): 将数据存储到 OUTY_ST_OIS 中
读取 OUTZ_L_G (26h) 和 OUTZ_H_G (27h): 将数据存储到 OUTZ_ST_OIS 中
16 位数据用二进制补码表示。

对每个轴上存储的数据求平均值

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

是 (通过)

否 (失败)

将 00h 写入 SPI2_INT_OIS (6Fh) → 禁止自检
将 00h 写入 SPI2_CTRL1_OIS (70h) → 关闭传感器

版本历史

表 100. 文档版本历史

日期	版本	变更
2022 年 8 月 4 日	1	初始版本

目录

1	引脚说明.....	2
2	寄存器.....	5
2.1	嵌入功能寄存器	10
2.2	嵌入高级功能页	12
2.3	传感器集线器寄存器	14
3	工作模式.....	16
3.1	加速度计功耗模式和输出数据率.....	16
3.2	陀螺仪功耗模式和输出数据率.....	18
3.3	正常模式	19
3.4	连接模式	19
3.5	加速度计带宽	20
3.5.1	加速度计斜率滤波器.....	22
3.6	加速度计开启/关闭时间	22
3.7	加速度计双通道模式.....	24
3.8	陀螺仪带宽	25
3.9	陀螺仪开启/关闭时间	29
3.10	陀螺仪 EIS 通道	31
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	33
4.4	使用块数据更新（block data update, BDU）功能	34
4.5	理解输出数据	34
4.5.1	输出数据示例	34
4.6	加速度计偏移寄存器.....	35
4.7	DEN（数据使能）.....	35
4.7.1	电平感应触发模式	36
4.7.2	电平感应锁存模式	36
4.7.3	用于 DEN 冲压的 LSB 选择	36

5	中断生成	37
5.1	中断引脚配置	38
5.2	自由落体中断	40
5.3	唤醒中断	41
5.4	6D/4D 方向检测	44
5.4.1	6D 方向检测	44
5.4.2	4D 方向检测	46
5.5	单击和双击识别	46
5.5.1	单击	47
5.5.2	双击	48
5.5.3	单击和双击识别配置	50
5.5.4	单击示例	53
5.5.5	双击示例	53
5.6	活动/不活动和运动/静止识别	53
5.6.1	静止/运动检测	57
5.7	启动状态	57
6	嵌入功能	59
6.1	计步功能：步伐检测和步数计算	59
6.2	大幅运动检测	61
6.3	相对倾斜	62
6.4	时间戳	65
6.5	传感器融合功能	65
6.5.1	陀螺仪偏差初始值设置	66
6.6	嵌入功能额外配置和监视	67
7	模式 2——传感器集线器 (sensor hub) 模式	68
7.1	传感器集线器 (sensor hub) 模式说明	68
7.2	传感器集线器 (sensor hub) 模式寄存器	69
7.2.1	MASTER_CONFIG (14h)	69
7.2.2	STATUS_MASTER (22h)	70
7.2.3	SLV0_ADD (15h)、SLV0_SUBADD (16h) 和 SLV0_CONFIG (17h)	71
7.2.4	SLV1_ADD (18h)、SLV1_SUBADD (19h) 和 SLV1_CONFIG (1Ah)	72

7.2.5	SLV2_ADD (1Bh)、SLV2_SUBADD (1Ch) 和 SLV2_CONFIG (1Dh)	73
7.2.6	SLV3_ADD (1Eh)、SLV3_SUBADD (1Fh) 和 SLV3_CONFIG (20h)	74
7.2.7	DATAWRITE_SLV0 (0Eh)	74
7.2.8	SENSOR_HUB_x 寄存器	75
7.3	传感器集线器 (sensor hub) 直通功能	76
7.4	传感器集线器 (sensor hub) 模式示例	76
8	模式 3——OIS 功能	79
8.1	辅助 SPI 全控制	79
8.2	SPI2 寄存器	82
8.2.1	SPI2_INT_OIS (6Fh)	83
8.2.2	SPI2_CTRL1_OIS (70h)	83
8.2.3	SPI2_CTRL2_OIS (71h)	84
8.2.4	SPI2_CTRL3_OIS (72h)	84
8.2.5	SPI2_STATUS_REG_OIS (1Eh)	85
8.3	通过辅助 SPI 读取 OIS 陀螺仪数据	85
8.4	通过辅助 SPI 读取 OIS 加速度计数据	85
8.5	主接口全控制	86
8.6	UI/SPI2 共享寄存器	86
9	先进先出 (FIFO) 缓冲器	87
9.1	FIFO 说明和批处理传感器	88
9.2	FIFO 寄存器	89
9.2.1	FIFO_CTRL1	90
9.2.2	FIFO_CTRL2	90
9.2.3	FIFO_CTRL3	91
9.2.4	FIFO_CTRL4	92
9.2.5	COUNTER_BDR_REG1	93
9.2.6	COUNTER_BDR_REG2	93
9.2.7	FIFO_STATUS1	93
9.2.8	FIFO_STATUS2	94
9.2.9	FIFO_DATA_OUT_TAG	94
9.2.10	FIFO_DATA_OUT	95
9.3	FIFO 批处理传感器	96

9.4	主要传感器	96
9.5	辅助传感器	97
9.6	虚拟传感器	99
9.6.1	外部传感器和 NACK 传感器	99
9.6.2	计步器传感器	100
9.6.3	SFLP 生成的传感器	100
9.6.4	MLC 生成的传感器	101
9.7	FIFO 模式	102
9.7.1	Bypass 模式	102
9.7.2	FIFO 模式	102
9.7.3	Continue 模式	103
9.7.4	Continue-FIFO 模式	104
9.7.5	Bypass-Continue 模式	104
9.7.6	Bypass-FIFO 模式	106
9.7.7	ContinueWTM-Full 模式	107
9.8	从 FIFO 恢复数据	108
9.9	FIFO 水印阈值	109
9.10	FIFO 压缩	110
9.10.1	时间相关性	111
9.10.2	数据格式	112
9.10.3	在运行时间禁用 FIFO 压缩	113
9.10.4	使能了 FIFO 压缩的 CFG-Change 传感器	114
9.10.5	未压缩数据率	114
9.10.6	FIFO 压缩初始化	114
9.10.7	FIFO 压缩示例	114
9.11	时间戳相关性	116
10	温度传感器	117
10.1	温度数据计算示例	117
11	模拟集线器和 Qvar 功能	118
12	自检功能	119
12.1	加速度计自检 (UI) – 模式 1、2	119
12.2	OIS 链接通后的加速度计自检 (UI) – 模式 3	121

12.3	加速度计自检 (OIS) – 模式 3.....	123
12.4	陀螺仪自检 (UI) – 模式 1、2.....	125
12.5	OIS 链接通后的陀螺仪自检 (UI) – 模式 3.....	127
12.6	陀螺仪自检 (OIS) – 模式 3	129
版本历史		131

表一览

表 1.	内部引脚状态	3
表 2.	寄存器	5
表 3.	SPI 寄存器	9
表 4.	嵌入功能寄存器	10
表 5.	嵌入高级功能寄存器——page 0	12
表 6.	嵌入高级功能寄存器——page 1	13
表 7.	嵌入高级功能寄存器——page 2	13
表 8.	传感器集线器寄存器	14
表 9.	加速度计功耗模式	17
表 10.	加速度计 ODR	17
表 11.	陀螺仪功耗模式	18
表 12.	陀螺仪 ODR	18
表 13.	功耗 (@ V _{dd} = 1.8 V, T = 25 °C)	19
表 14.	模式 1/2/3 下的加速度计带宽选择	21
表 15.	加速度计开启/关闭时间 (LPF2 和 HP 禁用)	23
表 16.	要丢弃的加速度计采样	23
表 17.	模式 1/2 下的陀螺仪总带宽选择	25
表 18.	陀螺仪低功耗模式下的带宽	27
表 19.	UI 链——模式 3 下的陀螺仪总带宽选择	28
表 20.	模式 1/2 下的陀螺仪开启/关闭时间	29
表 21.	模式 1/2 下要丢弃的陀螺仪采样 (LPF1 禁用)	29
表 22.	模式 1/2 下的陀螺仪链稳定时间 (LPF1 使能)	30
表 23.	陀螺仪 LPF_EIS 滤波器带宽选择	31
表 24.	输出数据寄存器内容 vs. 加速度 (FS _{XL} = ±2 g)	34
表 25.	输出数据寄存器内容 vs. 角速度 (FS _G = ±250 dps)	34
表 26.	DEN 配置	35
表 27.	INT1_CTRL 寄存器	38
表 28.	MD1_CFG 寄存器	38
表 29.	INT2_CTRL 寄存器	38
表 30.	MD2_CFG 寄存器	39
表 31.	自由落体阈值 LSB 值	40
表 32.	唤醒阈值分辨率	41
表 33.	D6D_SRC 寄存器	44
表 34.	4D/6D 功能阈值	44
表 35.	6D 定位下的 D6D_SRC 寄存器	45
表 36.	TAP_PRIORITY_[2:0] 位的配置	50
表 37.	TAP_SRC 寄存器	52
表 38.	不活动事件时的目标加速度计 ODR 配置	54
表 39.	不活动事件配置	54
表 40.	活动/不活动阈值分辨率	55
表 41.	EMB_FUNC_SRC 嵌入功能寄存器	59
表 42.	IS_STEP_DET 配置	60
表 43.	ODR _{coeff} 值	65
表 44.	k 系数	66
表 45.	MASTER_CONFIG 寄存器	69
表 46.	STATUS_MASTER/STATUS_MASTER_MAINPAGE 寄存器	70
表 47.	SLV0_ADD 寄存器	71
表 48.	SLV0_SUBADD 寄存器	71
表 49.	SLV0_CONFIG 寄存器	71
表 50.	SLV1_ADD 寄存器	72
表 51.	SLV1_SUBADD 寄存器	72
表 52.	SLV1_CONFIG 寄存器	72

表 53.	SLV2_ADD 寄存器	73
表 54.	SLV2_SUBADD 寄存器	73
表 55.	SLV2_CONFIG 寄存器	73
表 56.	SLV3_ADD 寄存器	74
表 57.	SLV3_SUBADD 寄存器	74
表 58.	SLV3_CONFIG 寄存器	74
表 59.	DATAWRITE_SLV0 寄存器	74
表 60.	模式 3 引脚说明	79
表 61.	SPI2_INT_OIS 寄存器	83
表 62.	SPI2_CTRL1_OIS 寄存器	83
表 63.	SPI2_CTRL2_OIS 寄存器	84
表 64.	LPF1 滤波器配置	84
表 65.	SPI2_CTRL3_OIS 寄存器	84
表 66.	LPF_OIS 滤波器配置	84
表 67.	SPI2_STATUS_REG_OIS 寄存器	85
表 68.	FIFO_CTRL1 寄存器	90
表 69.	FIFO_CTRL2 寄存器	90
表 70.	强制未压缩数据写入配置	90
表 71.	FIFO_CTRL3 寄存器	91
表 72.	加速度计批处理数据率	91
表 73.	陀螺仪批处理数据率	91
表 74.	时间戳批处理数据率	92
表 75.	温度传感器批处理数据率	92
表 76.	FIFO_CTRL4 寄存器	92
表 77.	COUNTER_BDR_REG1 寄存器	93
表 78.	COUNTER_BDR_REG2 寄存器	93
表 79.	FIFO_STATUS1 寄存器	93
表 80.	FIFO_STATUS2 寄存器	94
表 81.	FIFO_DATA_OUT_TAG 寄存器	94
表 82.	TAG_SENSOR 位域和相关传感器	94
表 83.	FIFO 中的主要传感器输出数据格式	96
表 84.	FIFO 中的温度输出数据格式	97
表 85.	FIFO 中的时间戳输出数据格式	97
表 86.	FIFO 中的 CFG-Change 输出数据格式	98
表 87.	BDR_SHUB	99
表 88.	FIFO 中的 Nack 传感器输出数据格式	99
表 89.	FIFO 中的计步器输出数据格式	100
表 90.	FIFO 中的 MLC 结果	101
表 91.	FIFO 中的 MLC 滤波器或特征	101
表 92.	FIFO 压缩标签和相关数据	111
表 93.	FIFO 中的 2xC 压缩数据输出数据格式	112
表 94.	FIFO 中的 3xC 压缩数据输出数据格式	112
表 95.	禁用运行时间压缩的示例	113
表 96.	使能了 FIFO 压缩的器件配置修改示例	114
表 97.	UNCOPTR_RATE 配置	114
表 98.	FIFO 压缩示例	115
表 99.	输出数据寄存器内容 vs. 温度	117
表 100.	文档版本历史	131

图一览

图 1.	引脚连接	2
图 2.	加速度计滤波链 (UI 路径)	20
图 3.	加速度计斜率滤波器	22
图 4.	双通道模式	24
图 5.	陀螺仪数字链——模式 1 (UI/EIS) 和模式 2	25
图 6.	陀螺仪数字链——模式 3 (OIS)	28
图 7.	陀螺仪 EIS 通道	31
图 8.	数据准备就绪信号	33
图 9.	电平感应触发模式, DEN 低电平有效	36
图 10.	电平感应锁存模式, DEN 低电平有效	36
图 11.	自由落体中断	40
图 12.	唤醒中断 (利用斜率滤波器)	42
图 13.	6D 识别方向	45
图 14.	单击事件识别	47
图 15.	双击事件识别 (LIR 位 = 0)	49
图 16.	单击和双击识别 (LIR 位 = 0)	51
图 17.	活动/不活动识别 (利用斜率滤波器)	56
图 18.	倾斜度检测	63
图 19.	模式 2 下的外部传感器连接	68
图 20.	SENSOR_HUB_X 配置示例	75
图 21.	直通功能	76
图 22.	模式 3 下的外部控制器连接 (SPI 3 线)	80
图 23.	模式 3 下的陀螺仪滤波链	80
图 24.	模式 3 下的加速度计滤波链	81
图 25.	主要传感器和时隙的定义	96
图 26.	FIFO 模式 (STOP_ON_WTM = 0)	102
图 27.	Continue 模式	103
图 28.	Continue-FIFO 模式	104
图 29.	Bypass-Continue 模式	105
图 30.	Bypass-FIFO 模式	106
图 31.	ContinueWTM-Full 模式	107
图 32.	FIFO 阈值 (STOP_ON_WTM = 0)	109
图 33.	FIFO 模式下的 FIFO 阈值 (STOP_ON_WTM = 1)	109
图 34.	Continue 模式下的 FIFO 阈值 (STOP_ON_WTM = 1)	110
图 35.	Qvar 外部连接	118
图 36.	加速度计自检流程 (UI)	120
图 37.	OIS 链接通时的加速度计自检流程 (UI)	122
图 38.	加速度计自检流程 (OIS)	124
图 39.	陀螺仪自检流程 (UI)	126
图 40.	OIS 链接通时的陀螺仪自检流程 (UI)	128
图 41.	陀螺仪自检流程 (OIS)	130

重要通知 - 请仔细阅读

意法半导体公司及其子公司（“意法半导体”）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意法半导体产品的最新信息。意法半导体产品的销售依照订单确认时的相关意法半导体销售条款。

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

意法半导体不对任何知识产权进行任何明示或默示的授权或许可。

转售的意法半导体产品如有不同于此处提供的信息的规定，将导致意法半导体针对该产品授予的任何保证失效。

ST 和 ST 标志是意法半导体的商标。关于意法半导体商标的其他信息，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。

© 2021 STMicroelectronics - 保留所有权利