AN4650
Application note

LSM6DS3: always-on 3D accelerometer and 3D gyroscope

Introduction

This document is intended to provide usage information and application hints related to ST's LSM6DS3 iNEMO inertial module.

The LSM6DS3 is a 3D digital accelerometer and 3D digital gyroscope system-in-package with a digital I^2C/SPI serial interface standard output, performing at 0.9 mA in combo Normal mode and 1.25 mA (up to 1.6 kHz) in combo High-Performance mode. Thanks to the ultra-low noise performance of both the gyroscope and the accelerometer, the device combines always-on low-power features with superior sensing precision for an optimal motion experience for the consumer. Furthermore, the accelerometer features smart sleep-to-wake-up (Activity) and return-to-sleep (Inactivity) functions that allow advanced power saving.

The device has a dynamic user-selectable full-scale acceleration range of ±2/±4/±8/±16 *g* and an angular rate range of ±125/±245/±500/±1000/±2000 dps.

The LSM6DS3 can be configured to generate interrupt signals by using hardware recognition of free-fall events, 6D orientation, tap and double-tap sensing, activity or inactivity, wake-up events.

The availability of a dedicated connection mode to external sensors allows the implementation of the sensor hub functionality.

The LSM6DS3 is compatible with the requirements of the leading OSs, offering real, virtual and batch-mode sensors. It has been designed to implement in hardware significant motion, tilt, pedometer functions, timestamp and to support the data acquisition of an external magnetometer with ironing correction (hard, soft).

The LSM6DS3 has an integrated smart first-in first-out (FIFO) buffer of up to 8 kbyte size, allowing dynamic batching of significant data (i.e. external sensors, step counter, timestamp and temperature).

The LSM6DS3 is available in a small plastic land grid array package (LGA-14L) and it is guaranteed to operate over an extended temperature range from -40 °C to +85 °C.

The ultra-small size and weight of the SMD package make it an ideal choice for handheld portable applications such as smartphones, IoT connected devices, and wearables or any other application where reduced package size and weight are required.

Contents

天猫店铺网址:https://telesky.tmall.com/

天猫店铺网址:https://telesky.tmall.com/

List of tables

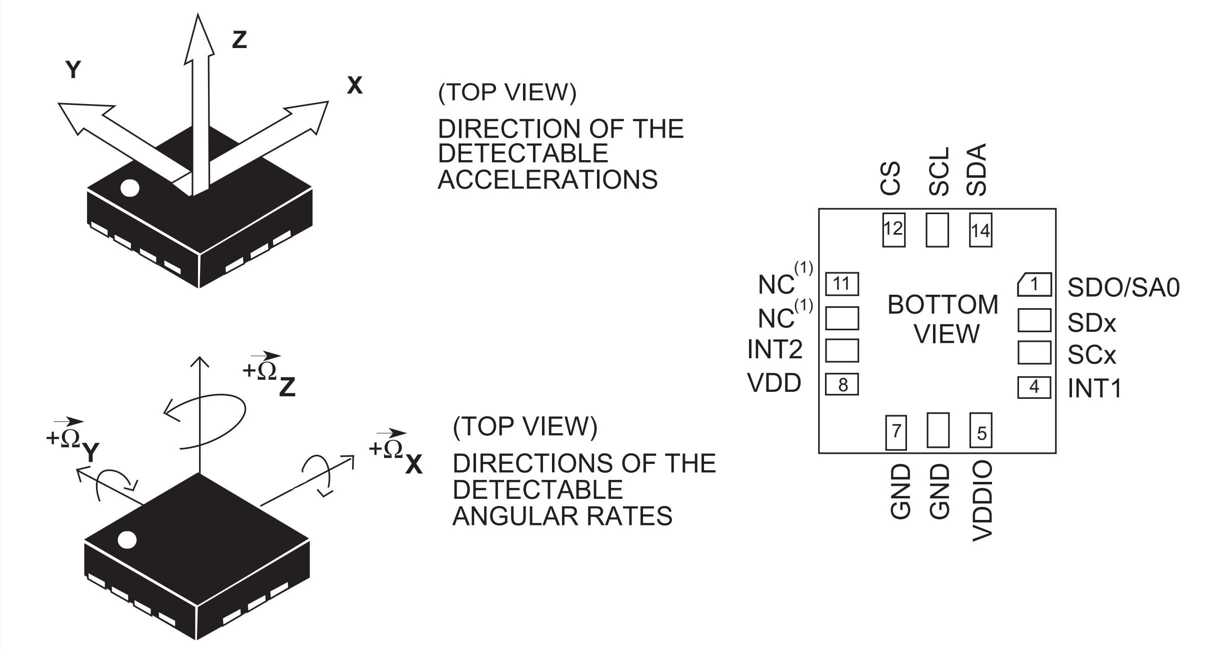
天猫店铺网址:https://telesky.tmall.com/

天猫店铺网址:https://telesky.tmall.com/

List of figures

1 Pin description

Figure 1. Pin connections



1. Leave pin electrically unconnected and soldered to PCB.

Table 1. Pin status

Pin#	Name	Mode 1 function	Mode 2 function	Pin status Mode 1	Pin status Mode 2
1	SDO/SA0	SPI 4-wire interface serial data output (SDO) I^2C least significant bit of the device address (SA0)	SPI 4-wire interface serial data output (SDO) I^2C least significant bit of the device address (SA0)	Default: Input without pull-up. Pull-up is enabled if bit SIM = 1 (SPI 3-wire) in reg 12h.	Default: Input without pull-up. Pull-up is enabled if bit SIM = 1 (SPI 3-wire) in reg 12h.
2	SDx	Connect to VDDIO or GND	I^2C serial data master (MSDA)	Default: input without pull-up. Pull-up is enabled if bit PULL_UP_EN = 1 in reg 1Ah.	Default: input without pull-up. Pull-up is enabled if bit PULL_UP_EN = 1 in reg 1Ah.
3	SCx	Connect to VDDIO or GND	I^2C serial clock master (MSCL)	Default: input without pull-up. Pull-up is enabled if bit PULL_UP_EN = 1 in reg 1Ah.	Default: input without pull-up. Pull-up is enabled if bit PULL_UP_EN = 1 in reg 1Ah.
4	INT1	Programmable interrupt 1		Default: Output forced to ground	Default: Output forced to ground
5	VDDIO	Power supply for I/O pins			
6	GND	0 V supply			
7	GND	0 V supply			
8	VDD	Power supply			
9	INT2	Programmable interrupt 2 (INT2)/ Data enable (DEN)	Programmable interrupt 2 (INT2)/ Data enable (DEN)/ I^2C master external synchronization signal (MDRDY)	Default: Output forced to ground	Default: Output forced to ground
10	NC	Leave unconnected		Internally pulled up	Internally pulled up
11	NC	Leave unconnected		Internally connected to GND	Internally connected to GND
12	CS	I^2C/SPI mode selection (1: SPI idle mode / I^2C communication enabled; 0: SPI communication mode / I^2C disabled)	I^2C/SPI mode selection (1: SPI idle mode / I^2C communication enabled; 0: SPI communication mode / I^2C disabled)	Default: Input with pull-up. Pull-up is disabled if bit I2C_disable = 1 in reg 13h.	Default: Input with pull-up. Pull-up is disabled if bit I2C_disable = 1 in reg 13h.

天猫店铺网址:https://telesky.tmall.com/

Table 1. Pin status (continued)

Pin#	Name	Mode 1 function	Mode 2 function	Pin status Mode 1	Pin status Mode 2
13	SCL	I^2C serial clock (SCL) SPI serial port clock (SPC)	I^2C serial clock (SCL) SPI serial port clock (SPC)	Input without pull-up	Input without pull-up
14	SDA	I^2C serial data (SDA) SPI serial data input (SDI) 3-wire interface serial data output (SDO)	I^2C serial data (SDA) SPI serial data input (SDI) 3-wire interface serial data output (SDO)	Input without pull-up	Input without pull-up

Internal pull-up value is from 30 kΩ to 50 kΩ, depending on VDDIO.

天猫店铺网址:https://telesky.tmall.com/

2 Registers

Table 2. Registers

Register name	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FUNC_CFG_ACCESS	01h	FUNC_CFG_EN	0	0	0	0	0	0	0
SENSOR_SYNC_TIME_FRAME	04h	TPH_7	TPH_6	TPH_5	TPH_4	TPH_3	TPH_2	TPH_1	TPH_0
FIFO_CTRL1	06h	FTH_7	FTH_6	FTH_5	FTH_4	FTH_3	FTH_2	FTH_1	FTH_0
FIFO_CTRL2	07h	TIMER_PEDO_FIFO_EN	TIMER_PEDO_FIFO_DRDY	0	0	FTH_11	FTH_10	FTH_9	FTH_8
FIFO_CTRL3	08h	0	0	DEC_FIFO_GYRO2	DEC_FIFO_GYRO1	DEC_FIFO_GYRO0	DEC_FIFO_XL2	DEC_FIFO_XL1	DEC_FIFO_XL0
FIFO_CTRL4	09h	0	ONLY_HIGH_DATA	DEC_DS4_FIFO2	DEC_DS4_FIFO1	DEC_DS4_FIFO0	DEC_DS3_FIFO2	DEC_DS3_FIFO1	DEC_DS3_FIFO0
FIFO_CTRL5	0Ah	0	ODR_FIFO_3	ODR_FIFO_2	ODR_FIFO_1	ODR_FIFO_0	FIFO_MODE_2	FIFO_MODE_1	FIFO_MODE_0
ORIENT_CFG_G	0Bh	0	0	SignX_G	SignY_G	SignZ_G	Orient_2	Orient_1	Orient_0
INT1_CTRL	0Dh	INT1_STEP_DETECTOR	INT1_SIG_MOT	INT1_FULL_FLAG	INT1_FIFO_OVR	INT1_FTH	INT1_BOOT	INT1_DRDY_G	INT1_DRDY_XL
INT2_CTRL	0Eh	INT2_STEP_DELTA	INT2_STEP_COUNT_OV	INT2_FULL_FLAG	INT2_FIFO_OVR	INT2_FTH	INT2_DRDY_TEMP	INT2_DRDY_G	INT2_DRDY_XL
WHO_AM_I	0Fh	0	1	1	0	1	0	0	1
CTRL1_XL	10h	ODR_XL3	ODR_XL2	ODR_XL1	ODR_XL0	FS_XL1	FS_XL0	BW_XL1	BW_XL0
CTRL2_G	11h	ODR_G3	ODR_G2	ODR_G1	ODR_G0	FS_G1	FS_G0	FS_125	0
CTRL3_C	12h	BOOT	BDU	H_LACTIVE	PP_OD	SIM	IF_INC	BLE	SW_RESET
CTRL4_C	13h	XL_BW_SCAL_ODR	SLEEP_G	INT2_on_INT1	FIFO_TEMP_EN	DRDY_MASK	I2C_disable	0	STOP_ON_FTH
CTRL5_C	14h	ROUNDING2	ROUNDING1	ROUNDING0	0	ST1_G	ST0_G	ST1_XL	ST0_XL

Table 2. Registers (continued)

Register name	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CTRL6_C	15h	TRIG_EN	LVLen	LVL2_EN	XL_HM _MODE	0	0	0	0
CTRL7_G	16h	G_HM_MODE	HP_G_EN	HPCF_G1	HPCF_G0	HP_G_RST	ROUNDING _STATUS	0	0
CTRL8_XL	17h	LPF2_XL_EN	HPCF_XL1	HPCF_XL0	0	0	HP_SLOPE_ XL_EN	0	LOW_PASS _ON_6D
CTRL9_XL	18h	0	0	Zen_XL	Yen_XL	Xen_XL	SOFT_EN	0	0
CTRL10_C	19h	0	0	Zen_G	Yen_G	Xen_G	FUNC_EN	PEDO_RST _STEP	SIGN_MOTI ON_EN
MASTER_CONFIG	1Ah	DRDY_ON _INT1	DATA_VALID _SEL_FIFO	0	START _CONFIG	PULL_UP _EN	PASS _THROUGH _MODE	IRON_EN	MASTER _ON
WAKE_UP_SRC	1Bh	0	0	FF_IA	SLEEP _STATE_IA	WU_IA	X_WU	Y_WU	Z_WU
TAP_SRC	1Ch	0	TAP_IA	SINGLE _TAP	DOUBLE _TAP	TAP_SIGN	X_TAP	Y_TAP	Z_TAP
D6D_SRC	1Dh	0	D6D_IA	ZH	ZL	YH	YL	XH	XL
STATUS_REG	1Eh	-	-	-	-	-	TDA	GDA	XLDA
OUT_TEMP_L	20h	Temp7	Temp6	Temp5	Temp4	Temp3	Temp2	Temp1	Temp0
OUT_TEMP_H	21h	Temp15	Temp14	Temp13	Temp12	Temp11	Temp10	Temp9	Temp8
OUTX_L_G	22h	D7	D6	D5	D4	D3	D2	D1	D0
OUTX_H_G	23h	D15	D14	D13	D12	D11	D10	D9	D8
OUTY_L_G	24h	D7	D6	D5	D4	D3	D2	D1	D0
OUTY_H_G	25h	D15	D14	D13	D12	D11	D10	D9	D8
OUTZ_L_G	26h	D7	D6	D5	D4	D3	D2	D1	D0
OUTZ_H_G	27h	D15	D14	D13	D12	D11	D10	D9	D8
OUTX_L_XL	28h	D7	D6	D5	D4	D3	D2	D1	D0

Table 2. Registers (continued)

Register name	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OUTX_H_XL	29h	D15	D14	D13	D12	D11	D10	D9	D8
OUTY_L_XL	2Ah	D7	D6	D5	D4	D3	D2	D1	D0
OUTY_H_XL	2Bh	D15	D14	D13	D12	D11	D10	D9	D8
OUTZ_L_XL	2Ch	D7	D6	D5	D4	D3	D2	D1	D0
OUTZ_H_XL	2Dh	D15	D14	D13	D12	D11	D10	D9	D8
SENSORHUB1_REG	2Eh	SHub1_7	SHub1_6	SHub1_5	SHub1_4	SHub1_3	SHub1_2	SHub1_1	SHub1_0
SENSORHUB2_REG	2Fh	SHub2_7	SHub2_6	SHub2_5	SHub2_4	SHub2_3	SHub2_2	SHub2_1	SHub2_0
SENSORHUB3_REG	30h	SHub3_7	SHub3_6	SHub3_5	SHub3_4	SHub3_3	SHub3_2	SHub3_1	SHub3_0
SENSORHUB4_REG	31h	SHub4_7	SHub4_6	SHub4_5	SHub4_4	SHub4_3	SHub4_2	SHub4_1	SHub4_0
SENSORHUB5_REG	32h	SHub5_7	SHub5_6	SHub5_5	SHub5_4	SHub5_3	SHub5_2	SHub5_1	SHub5_0
SENSORHUB6_REG	33h	SHub6_7	SHub6_6	SHub6_5	SHub6_4	SHub6_3	SHub6_2	SHub6_1	SHub6_0
SENSORHUB7_REG	34h	SHub7_7	SHub7_6	SHub7_5	SHub7_4	SHub7_3	SHub7_2	SHub7_1	SHub7_0
SENSORHUB8_REG	35h	SHub8_7	SHub8_6	SHub8_5	SHub8_4	SHub8_3	SHub8_2	SHub8_1	SHub8_0
SENSORHUB9_REG	36h	SHub9_7	SHub9_6	SHub9_5	SHub9_4	SHub9_3	SHub9_2	SHub9_1	SHub9_0
SENSORHUB10_REG	37h	SHub10_7	SHub10_6	SHub10_5	SHub10_4	SHub10_3	SHub10_2	SHub10_1	SHub10_0
SENSORHUB11_REG	38h	SHub11_7	SHub11_6	SHub11_5	SHub11_4	SHub11_3	SHub11_2	SHub11_1	SHub11_0
SENSORHUB12_REG	39h	SHub12_7	SHub12_6	SHub12_5	SHub12_4	SHub12_3	SHub12_2	SHub12_1	SHub12_0
FIFO_STATUS1	3Ah	DIFF_FIFO_7	DIFF_FIFO_6	DIFF_FIFO_5	DIFF_FIFO_4	DIFF_FIFO_3	DIFF_FIFO_2	DIFF_FIFO_1	DIFF_FIFO_0
FIFO_STATUS2	3Bh	FTH	FIFO_OVER_RUN	FIFO_FULL	FIFO_EMPTY	DIFF_FIFO_11	DIFF_FIFO_10	DIFF_FIFO_9	DIFF_FIFO_8
FIFO_STATUS3	3Ch	FIFO_PATTERN_7	FIFO_PATTERN_6	FIFO_PATTERN_5	FIFO_PATTERN_4	FIFO_PATTERN_3	FIFO_PATTERN_2	FIFO_PATTERN_1	FIFO_PATTERN_0
FIFO_STATUS4	3Dh	0	0	0	0	0	0	FIFO_PATTERN_9	FIFO_PATTERN_8

Table 2. Registers (continued)

Register name	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FIFO_DATA_OUT_L	3Eh	DATA_OUT _FIFO_L_7	DATA_OUT _FIFO_L_6	DATA_OUT _FIFO_L_5	DATA_OUT _FIFO_L_4	DATA_OUT _FIFO_L_3	DATA_OUT _FIFO_L_2	DATA_OUT _FIFO_L_1	DATA_OUT _FIFO_L_0
FIFO_DATA_OUT_H	3Fh	DATA_OUT _FIFO_H_7	DATA_OUT _FIFO_H_6	DATA_OUT _FIFO_H_5	DATA_OUT _FIFO_H_4	DATA_OUT _FIFO_H_3	DATA_OUT _FIFO_H_2	DATA_OUT _FIFO_H_1	DATA_OUT _FIFO_H_0
TIMESTAMP0_REG	40h	TIMESTAMP 0_7	TIMESTAMP 0_6	TIMESTAMP 0_5	TIMESTAMP 0_4	TIMESTAMP 0_3	TIMESTAMP 0_2	TIMESTAMP 0_1	TIMESTAMP 0_0
TIMESTAMP1_REG	41h	TIMESTAMP 1_7	TIMESTAMP 1_6	TIMESTAMP 1_5	TIMESTAMP 1_4	TIMESTAMP 1_3	TIMESTAMP 1_2	TIMESTAMP 1_1	TIMESTAMP 1_0
TIMESTAMP2_REG	42h	TIMESTAMP 2_7	TIMESTAMP 2_6	TIMESTAMP 2_5	TIMESTAMP 2_4	TIMESTAMP 2_3	TIMESTAMP 2_2	TIMESTAMP 2_1	TIMESTAMP 2_0
STEP_TIMESTAMP_L	49h	STEP_TIMES TAMP_L_7	STEP_TIMES TAMP_L_6	STEP_TIMES TAMP_L_5	STEP_TIME STAMP_L_4	STEP_TIME STAMP_L_3	STEP_TIME STAMP_L_2	STEP_TIME STAMP_L_1	STEP_TIME STAMP_L_0
STEP_TIMESTAMP_H	4Ah	STEP_TIMES TAMP_H_7	STEP_TIMES TAMP_H_6	STEP_TIMES TAMP_H_5	STEP_TIME STAMP_H_4	STEP_TIME STAMP_H_3	STEP_TIME STAMP_H_2	STEP_TIME STAMP_H_1	STEP_TIME STAMP_H_0
STEP_COUNTER_L	4Bh	STEP_COUN TER_L_7	STEP_COUNT ER_L_6	STEP_COUN TER_L_5	STEP_COU NTER_L_4	STEP_COU NTER_L_3	STEP_COU NTER_L_2	STEP_COU NTER_L_1	STEP_COU NTER_L_0
STEP_COUNTER_H	4Ch	STEP_COUN TER_H_7	STEP_COUNT ER_H_6	STEP_COUN TER_H_5	STEP_COU NTER_H_4	STEP_COU NTER_H_3	STEP_COU NTER_H_2	STEP_COU NTER_H_1	STEP_COU NTER_H_0
SENSORHUB13_REG	4Dh	SHub13_7	SHub13_6	SHub13_5	SHub13_4	SHub13_3	SHub13_2	SHub13_1	SHub13_0
SENSORHUB14_REG	4Eh	SHub14_7	SHub14_6	SHub14_5	SHub14_4	SHub14_3	SHub14_2	SHub14_1	SHub14_0
SENSORHUB15_REG	4Fh	SHub15_7	SHub15_6	SHub15_5	SHub15_4	SHub15_3	SHub15_2	SHub15_1	SHub15_0
SENSORHUB16_REG	50h	SHub16_7	SHub16_6	SHub16_5	SHub16_4	SHub16_3	SHub16_2	SHub16_1	SHub16_0
SENSORHUB17_REG	51h	SHub17_7	SHub17_6	SHub17_5	SHub17_4	SHub17_3	SHub17_2	SHub17_1	SHub17_0
SENSORHUB18_REG	52h	SHub18_7	SHub18_6	SHub18_5	SHub18_4	SHub18_3	SHub18_2	SHub18_1	SHub18_0
FUNC_SRC	53h	STEP_COUN T_DELTA_IA	SIGN_ MOTION_IA	TILT_IA	STEP_ DETECTED	STEP_OVE RFLOW	0	SI_END_OP	SENSORHU B_END_OP
TAP_CFG	58h	TIMER_EN	PEDO_EN	TILT_EN	SLOPE _FDS	TAP_X_EN	TAP_Y_EN	TAP_Z_EN	LIR

天猫店铺网址:https://telesky.tmall.com/

Table 2. Registers (continued)

Register name	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TAP_THS_6D	59h	D4D_EN	SIXD_THS1	SIXD_THS0	TAP_THS4	TAP_THS3	TAP_THS2	TAP_THS1	TAP_THS0
INT_DUR2	5Ah	DUR3	DUR2	DUR1	DUR0	QUIET1	QUIET0	SHOCK1	SHOCK0
WAKE_UP_THS	5Bh	SINGLE_DOUBLE_TAP	INACTIVITY	WK_THS5	WK_THS4	WK_THS3	WK_THS2	WK_THS1	WK_THS0
WAKE_UP_DUR	5Ch	FF_DUR5	WAKE_DUR1	WAKE_DUR0	TIMER_HR	SLEEP_DUR3	SLEEP_DUR2	SLEEP_DUR1	SLEEP_DUR0
FREE_FALL	5Dh	FF_DUR4	FF_DUR3	FF_DUR2	FF_DUR1	FF_DUR0	FF_THS2	FF_THS1	FF_THS0
MD1_CFG	5Eh	INT1_INACT_STATE	INT1_SINGLE_TAP	INT1_WU	INT1_FF	INT1_DOUBLE_TAP	INT1_6D	INT1_TILT	INT1_TIMER
MD2_CFG	5Fh	INT2_INACT_STATE	INT2_SINGLE_TAP	INT2_WU	INT2_FF	INT2_DOUBLE_TAP	INT2_6D	INT2_TILT	INT2 _IRON
OUT_MAG_RAW_X_L	66h	D7	D6	D5	D4	D3	D2	D1	D0
OUT_MAG_RAW_X_H	67h	D15	D14	D13	D12	D11	D10	D9	D8
OUT_MAG_RAW_Y_L	68h	D7	D6	D5	D4	D3	D2	D1	D0
OUT_MAG_RAW_Y_H	69h	D15	D14	D13	D12	D11	D10	D9	D8
OUT_MAG_RAW_Z_L	6Ah	D7	D6	D5	D4	D3	D2	D1	D0
OUT_MAG_RAW_Z_H	6Bh	D15	D14	D13	D12	D11	D10	D9	D8

2.1 Embedded functions registers

The list of the registers for embedded functions available in the device is given in Table 3.

Embedded functions registers are accessible when the FUNC_CFG_EN bit is set to '1' in the FUNC_CFG_ACCESS register.

Note: All modifications to the content of the embedded functions registers have to be performed with both the accelerometer and the gyroscope sensor in Power-Down mode.

Table 3. Embedded functions registers

Register name	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SLV0_ADD	02h	Slave0 _add6	Slave0 _add5	Slave0 _add4	Slave0 _add3	Slave0 _add2	Slave0 _add1	Slave0 _add0	rw_0
SLV0_SUBADD	03h	Slave0 _reg7	Slave0 _reg6	Slave0 _reg5	Slave0 _reg4	Slave0 _reg3	Slave0 _reg2	Slave0 _reg1	Slave0 _reg0
SLAVE0_CONFIG	04h	Slave0 _rate1	Slave0 _rate0	Aux_sens _on1	Aux_sens _on0	Src_mode	Slave0 _numop2	Slave0 _numop1	Slave0 _numop0
SLV1_ADD	05h	Slave1 _add6	Slave1 _add5	Slave1 _add4	Slave1 _add3	Slave1 _add2	Slave1 _add1	Slave1 _add0	r_1
SLV1_SUBADD	06h	Slave1 _reg7	Slave1 _reg6	Slave1 _reg5	Slave1 _reg4	Slave1 _reg3	Slave1 _reg2	Slave1 _reg1	Slave1 _reg0
SLAVE1_CONFIG	07h	Slave1 _rate1	Slave1 _rate0	0	0	0	Slave1 _numop2	Slave1 _numop1	Slave1 _numop0
SLV2_ADD	08h	Slave2 _add6	Slave2 _add5	Slave2 _add4	Slave2 _add3	Slave2 _add2	Slave2 _add1	Slave2 _add0	r_2
SLV2_SUBADD	09h	Slave2 _reg7	Slave2 _reg6	Slave2 _reg5	Slave2 _reg4	Slave2 _reg3	Slave2 _reg2	Slave2 _reg1	Slave2 _reg0
SLAVE2_CONFIG	0Ah	Slave2 _rate1	Slave2 _rate0	0	0	0	Slave2 _numop2	Slave2 _numop1	Slave2 _numop0
SLV3_ADD	0Bh	Slave3 _add6	Slave3 _add5	Slave3 _add4	Slave3 _add3	Slave3 _add2	Slave3 _add1	Slave3 _add0	r_3

天猫店铺网址:https://telesky.tmall.com/

Table 3. Embedded functions registers (continued)

Register name	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SLV3_SUBADD	0Ch	Slave3 _reg7	Slave3 _reg6	Slave3 _reg5	Slave3 _reg4	Slave3 _reg3	Slave3 _reg2	Slave3 _reg1	Slave3 _reg0
SLAVE3_CONFIG	0Dh	Slave3 _rate1	Slave3 _rate0	0	0	0	Slave3 _numop2	Slave3 _numop1	Slave3 _numop0
DATAWRITE_SRC _MODE_SUB_SLV0	0Eh	Slave_dataw 7	Slave_dataw 6	Slave_dataw 5	Slave_dataw 4	Slave_dataw 3	Slave_dataw 2	Slave_dataw 1	Slave_dataw 0
PEDO_THS_REG	0Fh	PEDO_4G	-	-	THS_MIN_4	THS_MIN_3	THS_MIN_2	THS_MIN_1	THS_MIN_0
SM_THS	13h	SM_THS_7	SM_THS_6	SM_THS_5	SM_THS_4	SM_THS_3	SM_THS_2	SM_THS_1	SM_THS_0
PEDO_DEB_REG	14h	DEB _TIME_4	DEB _TIME_3	DEB _TIME_2	DEB _TIME_1	DEB _TIME_0	DEB _STEP_2	DEB _STEP_1	DEB _STEP_0
STEP_COUNT_DELTA	15h	SC_DELTA _7	SC_DELTA _6	SC_DELTA _5	SC_DELTA _4	SC_DELTA _3	SC_DELTA _2	SC_DELTA _1	SC_DELTA _0
MAG_SI_XX	24h	MAG_SI _XX_7	MAG_SI _XX_6	MAG_SI _XX_5	MAG_SI _XX_4	MAG_SI _XX_3	MAG_SI _XX_2	MAG_SI _XX_1	MAG_SI _XX_0
MAG_SI_XY	25h	MAG_SI _XY_7	MAG_SI _XY_6	MAG_SI _XY_5	MAG_SI _XY_4	MAG_SI _XY_3	MAG_SI _XY_2	MAG_SI _XY_1	MAG_SI _XY_0
MAG_SI_XZ	26h	MAG_SI _XZ_7	MAG_SI _XZ_6	MAG_SI _XZ_5	MAG_SI _XZ_4	MAG_SI _XZ_3	MAG_SI _XZ_2	MAG_SI _XZ_1	MAG_SI _XZ_0
MAG_SI_YX	27h	MAG_SI _YX_7	MAG_SI _YX_6	MAG_SI _YX_5	MAG_SI _YX_4	MAG_SI _YX_3	MAG_SI _YX_2	MAG_SI _YX_1	MAG_SI _YX_0
MAG_SI_YY	28h	MAG_SI _YY_7	MAG_SI _YY_6	MAG_SI _YY_5	MAG_SI _YY_4	MAG_SI _YY_3	MAG_SI _YY_2	MAG_SI _YY_1	MAG_SI _YY_0
MAG_SI_YZ	29h	MAG_SI _YZ_7	MAG_SI _YZ_6	MAG_SI _YZ_5	MAG_SI _YZ_4	MAG_SI _YZ_3	MAG_SI _YZ_2	MAG_SI _YZ_1	MAG_SI _YZ_0
MAG_SI_ZX	2Ah	MAG_SI _ZX_7	MAG_SI _ZX_6	MAG_SI _ZX_5	MAG_SI _ZX_4	MAG_SI _ZX_3	MAG_SI _ZX_2	MAG_SI _ZX_1	MAG_SI _ZX_0
MAG_SI_ZY	2Bh	MAG_SI _ZY_7	MAG_SI _ZY_6	MAG_SI _ZY_5	MAG_SI _ZY_4	MAG_SI _ZY_3	MAG_SI _ZY_2	MAG_SI _ZY_1	MAG_SI _ZY_0

天猫店铺网址:https://telesky.tmall.com/

Table 3. Embedded functions registers (continued)

Register name	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MAG_SI_ZZ	2Ch	MAG_SI_ZZ_7	MAG_SI_ZZ_6	MAG_SI_ZZ_5	MAG_SI_ZZ_4	MAG_SI_ZZ_3	MAG_SI_ZZ_2	MAG_SI_ZZ_1	MAG_SI_ZZ_0
MAG_OFFX_L	2Dh	MAG_OFFX_L_7	MAG_OFFX_L_6	MAG_OFFX_L_5	MAG_OFFX_L_4	MAG_OFFX_L_3	MAG_OFFX_L_2	MAG_OFFX_L_1	MAG_OFFX_L_0
MAG_OFFX_H	2Eh	MAG_OFFX_H_7	MAG_OFFX_H_6	MAG_OFFX_H_5	MAG_OFFX_H_4	MAG_OFFX_H_3	MAG_OFFX_H_2	MAG_OFFX_H_1	MAG_OFFX_H_0
MAG_OFFY_L	2Fh	MAG_OFFY_L_7	MAG_OFFY_L_6	MAG_OFFY_L_5	MAG_OFFY_L_4	MAG_OFFY_L_3	MAG_OFFY_L_2	MAG_OFFY_L_1	MAG_OFFY_L_0
MAG_OFFY_H	30h	MAG_OFFY_H_7	MAG_OFFY_H_6	MAG_OFFY_H_5	MAG_OFFY_H_4	MAG_OFFY_H_3	MAG_OFFY_H_2	MAG_OFFY_H_1	MAG_OFFY_H_0
MAG_OFFZ_L	31h	MAG_OFFZ_L_7	MAG_OFFZ_L_6	MAG_OFFZ_L_5	MAG_OFFZ_L_4	MAG_OFFZ_L_3	MAG_OFFZ_L_2	MAG_OFFZ_L_1	MAG_OFFZ_L_0
MAG_OFFZ_H	32h	MAG_OFFZ_H_7	MAG_OFFZ_H_6	MAG_OFFZ_H_5	MAG_OFFZ_H_4	MAG_OFFZ_H_3	MAG_OFFZ_H_2	MAG_OFFZ_H_1	MAG_OFFZ_H_0

3 Operating modes

The LSM6DS3 provides three possible operating configurations:

- only accelerometer active and gyroscope in Power-Down;
- only gyroscope active and accelerometer in Power-Down;
- both accelerometer and gyroscope active with independent ODR.

After the power supply is applied, the LSM6DS3 performs a 20 ms boot procedure to load the trimming parameters. After the boot is completed, both the accelerometer and the gyroscope are automatically configured in Power-Down mode.

The accelerometer and the gyroscope can be independently configured in four different power modes: Power-Down, Low-Power, Normal and High-Performance mode. They are allowed to have different data rates without any limit. The gyroscope sensor can also be set in Sleep mode to reduce its power consumption.

When both the accelerometer and gyroscope are on, the accelerometer is synchronized with the gyroscope and the data rates of the two sensors are integer multiples of each other. If the accelerometer and the gyroscope have been configured with the same output data rate, the gyroscope data-ready signal (DRDY_G) is always subsequent to the accelerometer data-ready signal (DRDY_XL); in this case, for synchronous reading of the two sensors, it is convenient to use the gyroscope data-ready signal.

Referring to the LSM6DS3 datasheet, the output data rate (ODR_XL) bits of CTRL1_XL register and the High-Performance disable (XL_HM_MODE) bit of CTRL6_C register are used to select the power mode and the output data rate of the accelerometer sensor (*Table 4*).

Note: When the LSM6DS3 is configured in accelerometer-only mode (the gyroscope is in Power-Down mode) and the accelerometer is set in Low-Power mode, the FUNC_EN bit of the CTRL10_C register must be set to 1.

Table 4. Accelerometer ODR and power mode selection

ODR_XL [3:0]	ODR [Hz] when XL_HM_MODE = 1	ODR [Hz] when XL_HM_MODE = 0
0000	Power Down	Power Down
0001	12.5 Hz (Low Power)	12.5 Hz (High Performance)
0010	26 Hz (Low Power)	26 Hz (High Performance)
0011	52 Hz (Low Power)	52 Hz (High Performance)
0100	104 Hz (Normal mode)	104 Hz (High Performance)
0101	208 Hz (Normal mode)	208 Hz (High Performance)
0110	416 Hz (High Performance)	416 Hz (High Performance)
0111	833 Hz (High Performance)	833 Hz (High Performance)
1000	1.66 kHz (High Performance)	1.66 kHz (High Performance)
1001	3.33 kHz (High Performance)	3.33 kHz (High Performance)
1010	6.66 kHz (High Performance)	6.66 kHz (High Performance)

The output data rate (ODR_G) bits of CTRL2_G register and the High-Performance disable (G_HM_MODE) bit of CTRL7_G register are used to select the power mode and output data rate of the gyroscope sensor (*Table 5*).

Table 5. Gyroscope ODR and power mode selection

ODR_G [3:0]	ODR [Hz] when G_HM_MODE = 1	ODR [Hz] when G_HM_MODE = 0
0000	Power Down	Power Down
0001	12.5 Hz (Low Power)	12.5 Hz (High Performance)
0010	26 Hz (Low Power)	26 Hz (High Performance)
0011	52 Hz (Low Power)	52 Hz (High Performance)
0100	104 Hz (Normal mode)	104 Hz (High Performance)
0101	208 Hz (Normal mode)	208 Hz (High Performance)
0110	416 Hz (High Performance)	416 Hz (High Performance)
0111	833 Hz (High Performance)	833 Hz (High Performance)
1000	1.66 kHz (High Performance)	1.66 kHz (High Performance)

Table 6 shows the typical values of power consumption for the different operating modes.

Table 6. Power consumption

ODR [Hz]	Accelerometer only (at Vdd = 1.8 V)	Gyroscope only (at Vdd = 1.8 V)	Combo [Acc + Gyro] (at Vdd = 1.8 V)
Power Down	-	-	6 µA
12.5 Hz (Low Power)	24 µA	470 µA	425 µA
26 Hz (Low Power)	31 µA	500 µA	450 µA
52 Hz (Low Power)	45 µA	540 µA	500 µA
104 Hz (Normal mode)	70 µA	625 µA	600 µA
208 Hz (Normal mode)	120 µA	880 µA	900 µA
12.5 Hz (High Perf.)	240 µA	1.15 mA	1.25 mA
26 Hz (High Perf.)	240 µA	1.15 mA	1.25 mA
52 Hz (High Perf.)	240 µA	1.15 mA	1.25 mA
104 Hz (High Perf.)	240 µA	1.15 mA	1.25 mA
208 Hz (High Perf.)	240 µA	1.15 mA	1.25 mA
416 Hz (High Perf.)	240 µA	1.15 mA	1.25 mA
833 Hz (High Perf.)	240 µA	1.15 mA	1.25 mA
1.66 kHz (High Perf.)	240 µA	1.15 mA	1.25 mA
3.33 kHz (High Perf.)	325 µA	N.A.	N.A.
6.66 kHz (High Perf.)	325 µA	N.A.	N.A.

天猫店铺网址:https://telesky.tmall.com/

3.1 Power-Down mode

When the accelerometer/gyroscope is in Power-Down mode, almost all internal blocks of the device are switched off to minimize power consumption. Digital interfaces (I^2C and SPI) are still active to allow communication with the device. The content of the configuration registers is preserved and the output data registers are not updated, keeping the last data sampled in memory before going into Power-Down mode.

3.2 High-Performance mode

In High-Performance mode, all accelerometer/gyroscope circuitry is always on and data are generated at the data rate selected through the ODR_XL/ODR_G bits.

Data interrupt generation is active.

3.3 Normal mode

While High-Performance mode guarantees the best performance in terms of noise, Normal mode further reduces the current consumption. The accelerometer/gyroscope data reading chain is automatically turned on and off to save power. In the gyroscope device, only the driving circuitry is always on.

Data interrupt generation is active.

Note: When the LSM6DS3 is configured in accelerometer-only mode (the gyroscope is in Power-Down mode) and the accelerometer is set in Low-Power mode, the FUNC_EN bit of the CTRL10_C register must be set to 1.

3.4 Low-Power mode

Low-Power mode differs from Normal mode in the available output data rates. In Low-Power mode low-speed ODRs are enabled; three low-speed ODRs can be chosen through the ODR_XL/ODR_G bits.

Data interrupt generation is active.

Note: When the LSM6DS3 is configured in accelerometer-only mode (the gyroscope is in Power-Down mode) and the accelerometer is set in Low-Power mode, the FUNC_EN bit of the CTRL10_C register must be set to 1.

3.5 Gyroscope Sleep mode

While the gyroscope is in Sleep mode the circuitry that drives the oscillation of the gyroscope mass is kept active. Compared to gyroscope Power-Down, turn-on time from Sleep mode to Low-Power/Normal/High-Performance mode is drastically reduced.

If the gyroscope is not configured in Power-Down mode, it enters in Sleep mode when the Sleep mode enable (SLEEP_G) bit of CTRL4_C register is set to 1, regardless of the selected gyroscope ODR.

3.6 Changing the power mode in accelerometer-only mode

In the LSM6DS3 different power modes and ODR are implemented. When the power mode is changed from High-Performance mode to Low-Power / Normal mode or vice versa, the internal reading chain needs to be reset in order to guarantee correct behavior of the device in the new selected power mode.

In accelerometer-only mode, a reading chain reset is executed by design when the ODR value is changed or when the Power-Down mode is set. If the power mode is changed without changing the ODR or without passing through Power-Down mode (e.g. when directly passing in accelerometer-only mode from 100 Hz High-Performance mode to 100 Hz Normal mode), the reading chain is not reset and the proper functionality of the device in the new selected power mode cannot be guaranteed.

In this case, there are two possible methods that allow always performing a correct reset of the reading chain during a power mode change:

1. If no ODR change is needed, Power-Down mode must be set before the new power mode change (refer to *Figure 2*).
2. If an ODR change is needed, the ODR must be changed after the power mode change (refer to *Figure 3*).

Note: No specific power mode procedure has to be applied in Gyroscope-only mode or in Accelerometer/Gyroscope Combo mode.

The following example refers to the first method above (no ODR change needed during power mode change); the procedure in accelerometer-only mode to change the power mode from 100 Hz High-Performance mode to 100 Hz Normal mode is:

a.	Write CTRL6_C = 00h	// Accelerometer in High-Performance mode (initial configuration)
b.	Write CTRL1_XL = 40h	// Accelerometer-only, 100 Hz ODR (initial configuration)
...		
n.	Write CTRL1_XL = 00h	// Accelerometer in Power-Down mode
n+1.	Write CTRL6_C = 10h	// Accelerometer in Normal mode
n+2.	Write CTRL1_XL = 40h	// Accelerometer-only, 100 Hz ODR

Note: Step n. is mandatory.

The following example refers to the second method above (ODR change needed during power mode change); the procedure in accelerometer-only mode to change the power mode from 100 Hz High-Performance mode to 200 Hz Normal mode is:

a.	Write CTRL6_C = 00h	// Accelerometer in High-Performance mode (initial configuration)
b.	Write CTRL1_XL = 40h	// Accelerometer-only, 100 Hz ODR (initial configuration)
...		
n.	Write CTRL6_C = 10h	// Accelerometer in Normal mode
n+1.	Write CTRL1_XL = 50h	// Accelerometer-only, 200 Hz ODR

Note: Steps n. and n+1. cannot be inverted.

天猫店铺网址:https://telesky.tmall.com/

Figure 2. Switching power modes (no change in ODR)

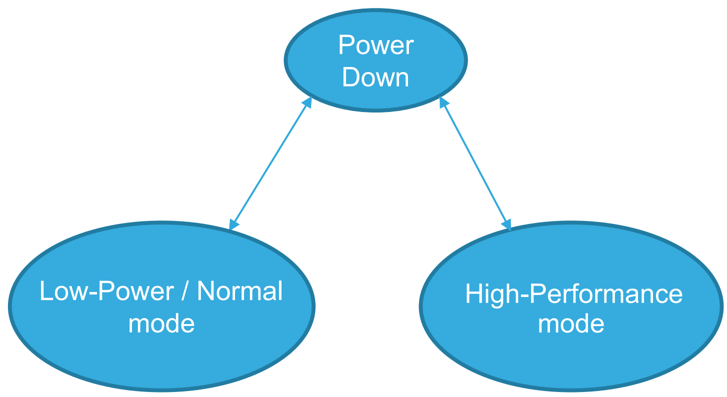


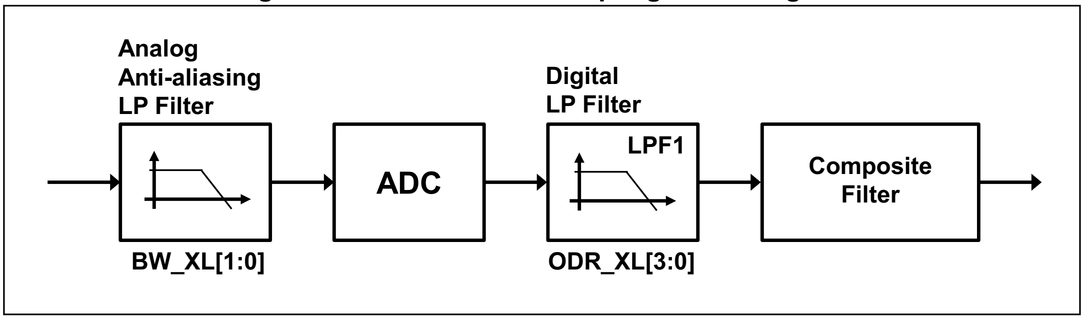
Figure 3. Switching power modes (with subsequent change in ODR)

(*) This procedure has to be performed also when passing from High-Performance mode to Low-Power / Normal mode.

3.7 Accelerometer bandwidth

The accelerometer sampling chain (*Figure 4*) is represented by a cascade of four blocks: an analog low-pass filter, an ADC converter, a digital low-pass filter and the composite group of digital filters described in *Figure 5*.

Figure 4. Accelerometer sampling chain diagram



The analog signal coming from the mechanical parts is filtered by a low-pass anti-aliasing filter before being converted by the ADC. The anti-aliasing filter is enabled in High-Performance mode only.

If the XL_BW_SCAL_ODR bit in CTRL4_C register is set to 1, the bandwidth of this analog filter is determined by setting the BW_XL bits of CTRL1_XL register; relative filter cutoff frequency values are given in *Table 7*. If the XL_BW_SCAL_ODR bit is set to 0, the bandwidth of the analog filter is determined by the ODR_XL selection (*Table 8*).

Table 7. Accelerometer anti-aliasing filter bandwidth selection (XL_BW_SCAL_ODR=1)

BW_XL[1:0]	Bandwidth [Hz]
00	400
01	200
10	100
11	50

Table 8. Accelerometer anti-aliasing bandwidth options (High-Performance mode)

Accelerometer ODR [Hz]	Analog filter cutoff [Hz] XL_BW_SCAL_ODR = 0	Analog filter cutoff [Hz] XL_BW_SCAL_ODR = 1
12.5 Hz (High Performance)	50	BW_XL[1:0]
26 Hz (High Performance)	50	BW_XL[1:0]
52 Hz (High Performance)	50	BW_XL[1:0]
104 Hz (High Performance)	50	BW_XL[1:0]
208 Hz (High Performance)	100	BW_XL[1:0]
416 Hz (High Performance)	200	BW_XL[1:0]
833 Hz (High Performance)	400	BW_XL[1:0]
1.66 kHz (High Performance)	400	BW_XL[1:0]
3.33 kHz (High Performance)	FILTER NOT USED	BW_XL[1:0]
6.66 kHz (High Performance)	FILTER NOT USED	BW_XL[1:0]

The digital signal is then filtered by a low-pass digital filter (LPF1) whose cutoff frequency depends on the selected accelerometer ODR, as shown in *Table 9*.

Table 9. Accelerometer LPF1 cutoff frequency

Accelerometer ODR [Hz]	LPF1 digital Filter cutoff frequency [Hz]
12.5 Hz (Low Power)	742
26 Hz (Low Power)	742
52 Hz (Low Power)	742
104 Hz (Normal mode)	742
208 Hz (Normal mode)	742
12.5 Hz (High Performance)	23
26 Hz (High Performance)	46
52 Hz (High Performance)	92
104 Hz (High Performance)	184
208 Hz (High Performance)	369
416 Hz (High Performance)	742
833 Hz (High Performance)	1517
1.66 kHz (High Performance)	3320
3.33 kHz (High Performance)	1517
6.66 kHz (High Performance)	3320

The total cutoff frequency resulting from the combination of the anti-aliasing filter and digital LPF1 filter is indicated in *Table 10*: it's basically determined by the lowest cutoff frequency of these two filters.

Table 10. Accelerometer anti-aliasing + LPF1 overall cutoff frequency

Accelerometer ODR [Hz]	Analog filter BW [Hz]	Digital filter BW [Hz]	Total BW [Hz]
12.5 Hz to 208 Hz (Low Power / Normal mode)	-	742	740
12.5 Hz (High Performance)	400	23	23
	200	23	23
	100	23	22
	50	23	20
26 Hz (High Performance)	400	46	45.5
	200	46	44.5
	100	46	41
	50	46	32.5

天猫店铺网址:https://telesky.tmall.com/

Table 10. Accelerometer anti-aliasing + LPF1 overall cutoff frequency (continued)

Accelerometer ODR [Hz]	Analog filter BW [Hz]	Digital filter BW [Hz]	Total BW [Hz]
52 Hz (High Performance)	400	92	89
	200	92	82
	100	92	65
	50	92	43
104 Hz (High Performance)	400	184	165
	200	184	131
	100	184	86
	50	184	48
208 Hz (High Performance)	400	369	265
	200	369	172
	100	369	96
	50	369	49
416 Hz (High Performance)	400	742	350
	200	742	192
	100	742	99
	50	742	50
833 Hz (High Performance)	400	1517	395
	200	1517	199
	100	1517	100
	50	1517	50
1.66 kHz (High Performance)	400	3320	400
	200	3320	200
	100	3320	100
	50	3320	50
3.33 kHz (High Performance)	-	1517	1500
	400	1517	390
	200	1517	198
	100	1517	100
	50	1517	50
6.66 kHz (High Performance)	-	3320	3160
	400	3320	400
	200	3320	200
	100	3320	100
	50	3320	50

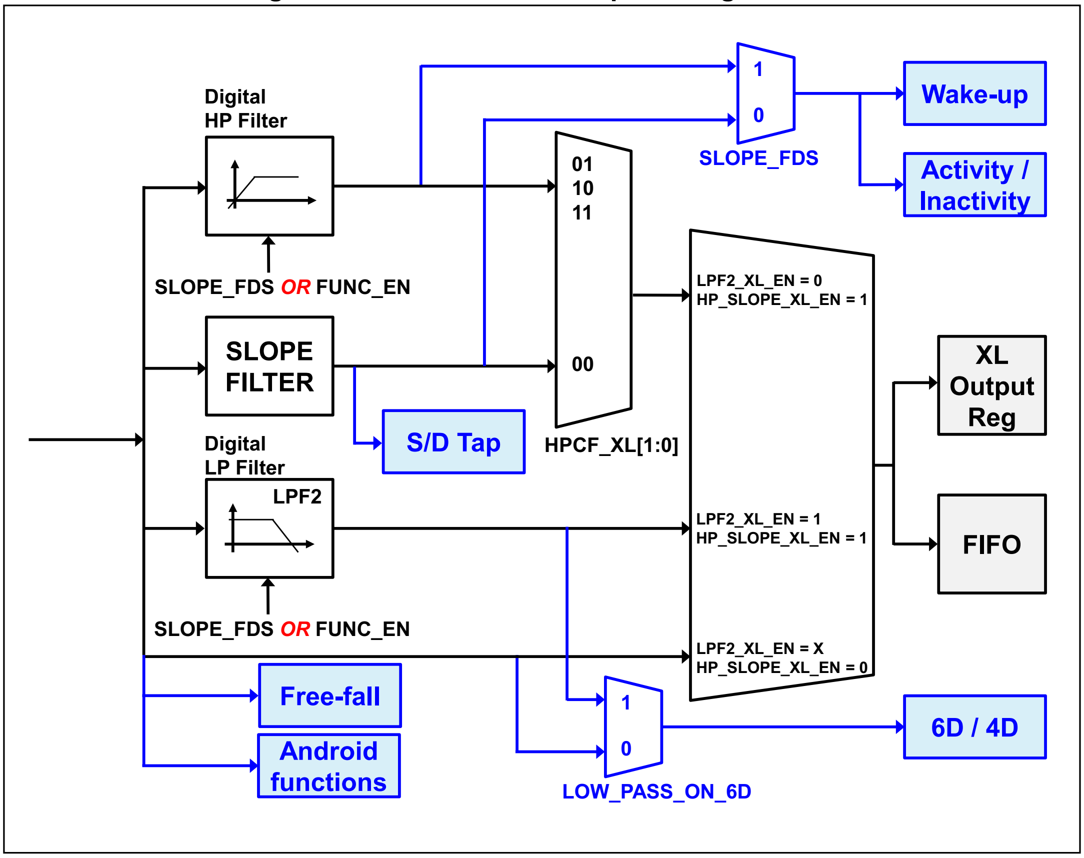
Finally, the digital signal is processed by the composite group of filters composed of a low-pass digital filter (LPF2), a high-pass digital filter and a slope filter. As shown in *Figure 5*, it is possible to independently apply these filters to the accelerometer output data (and to the FIFO data) and/or to the interrupt generators.

The enable signal of these high-pass and low-pass digital filters is the logic "OR" of the SLOPE_FDS bit of the TAP_CFG register and the FUNC_EN bit of the CTRL10_C register.

The SLOPE_FDS bit is also used to select the filter (high-pass or slope) used for the wake-up interrupt functionality. For this reason, if the wake-up functionality is implemented using the slope filter and also the LPF2 filter is required, the latter has to be enabled by setting the FUNC_EN bit to 1.

In all other cases, to enable the high-pass and low-pass digital filters it's recommended to set to 1 the SLOPE_FDS bit and set to 0 the FUNC_EN bit (if the embedded functions, such as the Android functions and the sensor hub, are not used).

Figure 5. Accelerometer composite digital filter



The bits LPF2_XL_EN, HP_SLOPE_XL_EN and HPCF_XL [1:0] of CTRL8_XL are used to select the filter applied to the accelerometer output data and to the FIFO data:

- if the HP_SLOPE_XL_EN bit is set to 0, no filter is applied, regardless of the LPF2_XL_EN bit configuration;
- if both the LPF2_XL_EN bit and the HP_SLOPE_XL_EN bit are set to 1, the LP digital filter (LPF2) is applied;
- if the LPF2_XL_EN bit is set to 0 and the HP_SLOPE_XL_EN bit is set to 1, the applied filter depends on the configuration of the HPCF_XL [1:0] bits, as shown in *Table 11*.

Table 11. Accelerometer slope and high-pass filter selection and cutoff frequency

HPCF_XL[1:0]	Applied filter	HP digital filter cutoff frequency [Hz]
00	Slope	ODR_XL / 4
01	High-Pass	ODR_XL / 100
10	High-Pass	ODR_XL / 9
11	High-Pass	ODR_XL / 400

The HPCF_XL [1:0] bits of CTRL8_XL are also used to select the cutoff frequency of the LPF2 filter, as shown in *Table 12*. This low-pass filter can also be used in the 6D/4D functionality by setting the LOW_PASS_ON_6D bit of CTRL8_XL register to 1.

Table 12. Accelerometer LPF2 cutoff frequency

HPCF_XL[1:0]	LPF2 digital filter cutoff frequency [Hz]
00	ODR_XL / 50
01	ODR_XL / 100
10	ODR_XL / 9
11	ODR_XL / 400

3.7.1 Accelerometer slope filter

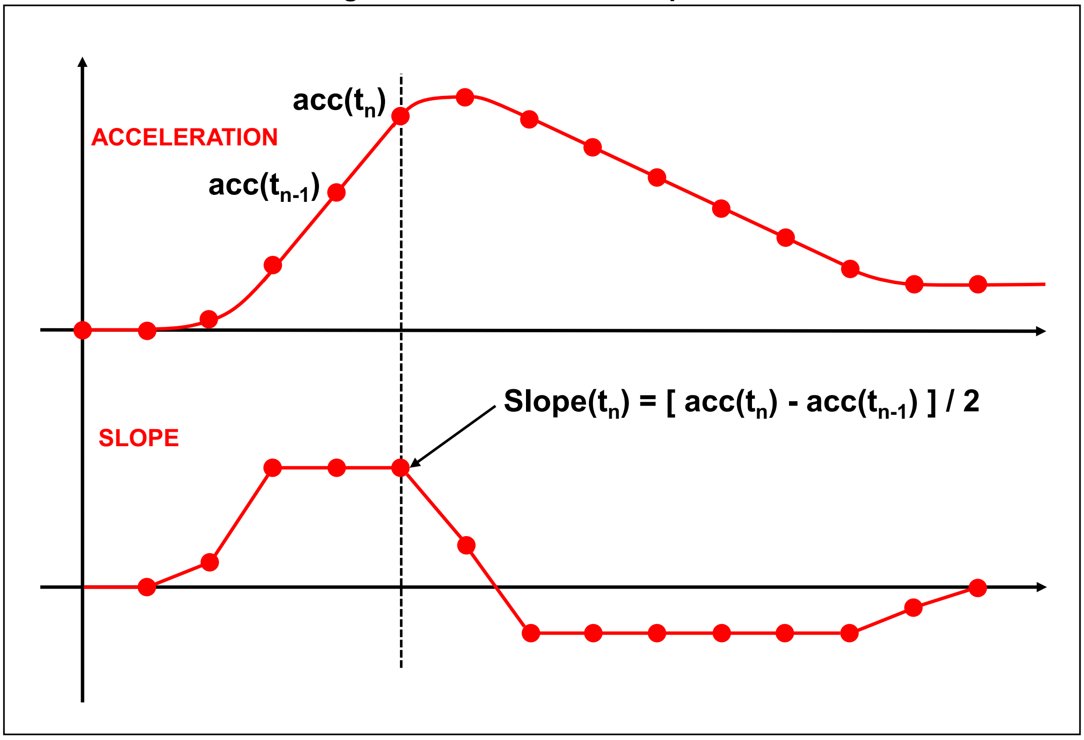
As shown in *Figure 5*, the LSM6DS3 device embeds a digital slope filter which is used for single/double-tap features; it can also be used for wake-up detection and for activity/inactivity functionality when the SLOPE_FDS bit of the TAP_CFG register is set to 0.

The slope filter output data is computed using the following formula:

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

An example of a slope data signal is illustrated in *Figure 6*.

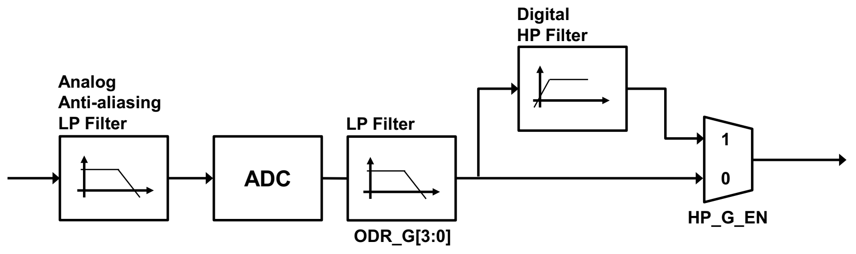
Figure 6. Accelerometer slope filter



3.8 Gyroscope bandwidth

The gyroscope sampling chain is represented by a cascade of four blocks: analog low-pass anti-aliasing filter, ADC converter, digital low-pass filter and a selectable high-pass filter (*Figure 7*).

Figure 7. Gyroscope sampling chain diagram



The analog signal coming from the mechanical parts is filtered by a low-pass anti-aliasing filter (having a constant bandwidth) before being converted by the ADC. The digital signal is then filtered by a low-pass digital filter whose cutoff frequency depends on the selected gyroscope ODR: the cutoff values in Low-Power and Normal mode are shown in *Table 13*; the cutoff values related to High-Performance mode are indicated in *Table 14*.

Table 13. Gyroscope digital low-pass filter cutoff in Low-Power / Normal mode

Gyroscope ODR [Hz]	Cutoff [Hz]
12.5 Hz (Low Power)	3.9
26 Hz (Low Power)	7.9
52 Hz (Low Power)	15.8
104 Hz (Normal mode)	31.4
208 Hz (Normal mode)	60.2

Table 14. Gyroscope digital low-pass filter cutoff in High-Performance mode

Gyroscope ODR [Hz]	Cutoff [Hz]
12.5 Hz (High Perf.)	4.2
26 Hz (High Perf.)	8.3
52 Hz (High Perf.)	16.6
104 Hz (High Perf.)	33.4
208 Hz (High Perf.)	66.7
416 Hz (High Perf.)	135.9
833 Hz (High Perf.)	295.4
1.66 kHz (High Perf.)	1057.0

The LSM6DS3 gyroscope provides embedded high-pass filtering capability to easily delete the DC component of the measured angular rate. As shown in *Figure 7*, through the HP_G_EN bit of the CTRL7_G register, it is possible to apply the filter on the gyroscope output data and on FIFO stored data.

Note: The embedded high-pass filter is available in High-Performance mode only. If the gyroscope is configured in Low-Power / Normal mode, the high-pass filter is bypassed regardless of the configuration of the HP_G_EN bit in the CTRL7_G register.

The bandwidth of the high-pass filter depends on the settings of the HPCF_G[1:0] bits of the CTRL7_G register. The high-pass filter cutoff frequencies are shown in *Table 15*.

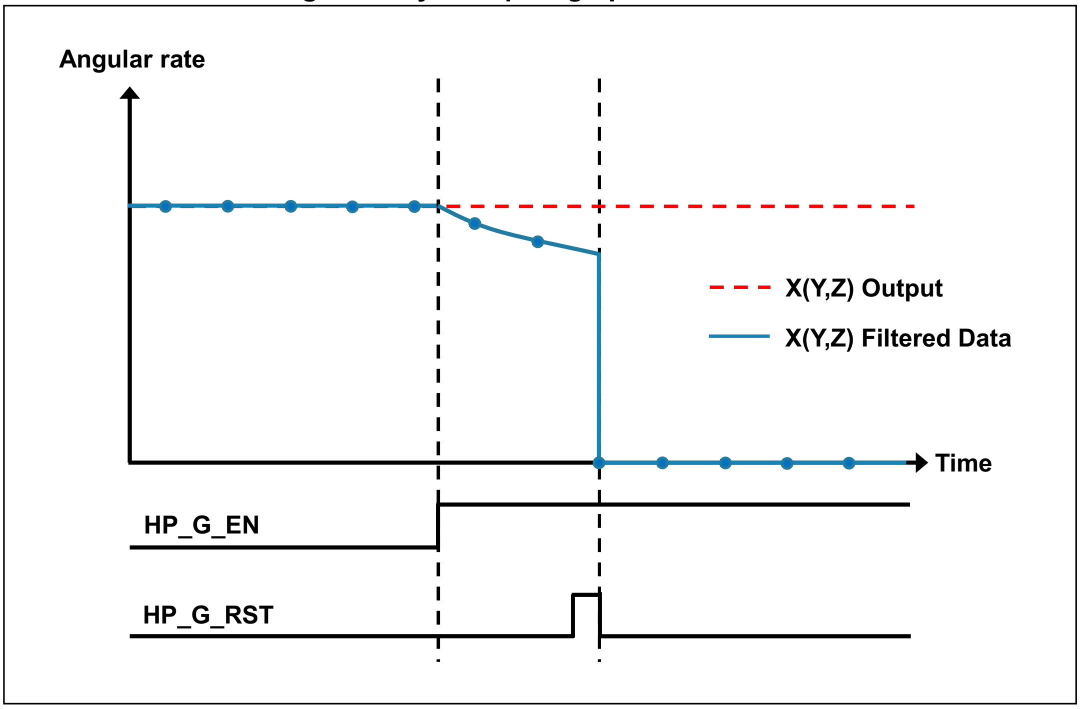
Table 15. Gyroscope high-pass filter cutoff frequency [Hz]

HPCF_G1	HPCF_G0	High-pass filter cutoff frequency [Hz]
0	0	0.0081
0	1	0.0324
1	0	2.07
1	1	16.32

The High-Pass filter can be reset by setting the HP_G_RST bit of the CTRL7_G register to 1. The reset operation instantly deletes the DC component of the angular rate from the next generated X, Y, Z output value (*Figure 8*).

After the filter resets, the HP_G_RST bit is automatically set back to 0.

Figure 8. Gyroscope high-pass filter reset



3.9 Accelerometer and gyroscope turn-on/off time

The accelerometer reading chain contains low-pass filtering to improve signal-to-noise performance and to reduce aliasing effects. For this reason, it is needed to take into account the settling time of the filter when the accelerometer / gyroscope power mode is switched or when the accelerometer / gyroscope ODR is changed.

Turn-on/off time has to be considered also for the gyroscope sensor when switching its modes or when the accelerometer / gyroscope ODR is changed.

The list of expected accelerometer / gyroscope turn-on/off times when the operating modes are changed is indicated in *Table 16*; the starting condition must be stable for at least 200 ms.

Table 17 clarifies how many accelerometer samples have to be discarded in High-Performance mode depending on the internal filter bandwidth and output data rate selection. Bandwidth value selection is described in *Table 8*.

Table 18 shows how many gyroscope samples have to be discarded when switching from gyroscope Sleep mode to Low-Power / Normal / High-Performance mode or when the accelerometer / gyroscope ODR is changed, depending on the output data rate selection.

Setting the DRDY_MASK bit of the CTRL4_C register to 1, the accelerometer and gyroscope data-ready signals are masked until the settling of the sensor filters is completed: this feature allows automatically ignoring the samples to be discarded.

Note: The DRDY_MASK bit masks the samples to be discarded only when switching from Power-Down to an active mode, it doesn't mask them on ODR changes.

Table 16. Accelerometer/gyroscope turn-on/off time

Starting mode	Operating mode change	Accelerometer Max turn-on/off time	Gyroscope Max turn-on/off time
Acc: Power Down Gyro: Power Down	Acc: Low Power / Normal mode	First sample correct	-
Acc: Power Down Gyro: Power Down	Acc: High Performance mode	see *Table 17*	-
Acc: Power Down Gyro: Power Down	Gyro: Sleep / Low Power / Normal / High Performance mode	-	80 ms
Acc: LP / NM / HP mode Gyro: Power Down	Gyro: Sleep / Low Power / Normal / High Performance mode	Acc in LP/NM mode: 20ms+1/ODR Acc in HP mode: max(20ms+1/ODR, number of sample to be discarded from *Table 17*)	80 ms
Acc: Power Down Gyro: LP / NM / HP mode	Acc: Low Power / Normal mode	First sample correct	see *Table 18*
Acc: Power Down Gyro: LP / NM / HP mode	Acc: High Performance mode	see *Table 17*	see *Table 18*

Table 16. Accelerometer/gyroscope turn-on/off time (continued)

Starting mode	Operating mode change	Accelerometer Max turn-on/off time	Gyroscope Max turn-on/off time
Acc: Power Down Gyro: Sleep mode	Acc: Low Power / Normal mode	First sample correct	No impact on current Sleep mode
Acc: Power Down Gyro: Sleep mode	Acc: High Performance mode	see *Table 17*	No impact on current Sleep mode
Acc: LP / NM / HP mode Gyro: LP / NM / HP mode	Acc: Power Down	1us	First sample correct
Acc: LP / NM / HP mode Gyro: Sleep mode	Acc: Power Down	1us	No impact on current Sleep mode
Acc: LP / NM / HP mode Gyro: LP / NM / HP mode	Gyro: Power Down / Sleep mode	First sample correct	1us
Acc: LP / NM / HP mode Gyro: LP / NM / HP mode	Acc: change ODR	Acc in LP/NM mode: first sample correct Acc in HP mode: see *Table 17*	see *Table 18*
Acc: LP / NM / HP mode Gyro: Sleep mode	Acc: change ODR	Acc in LP/NM mode: first sample correct Acc in HP mode: see *Table 17*	No impact on current Sleep mode
Acc: LP / NM / HP mode Gyro: Sleep mode	Gyro: Low Power / Normal / High Performance mode	Acc in LP/NM mode: first sample correct Acc in HP mode: see *Table 17*	see *Table 18*
Acc: LP / NM / HP mode Gyro: LP / NM / HP mode	Gyro: change ODR	Acc in LP/NM mode: first sample correct Acc in HP mode: see *Table 17*	see *Table 18*
Acc: LP / NM / HP mode Gyro: LP / NM / HP mode	Acc: change Analog Anti-aliasing filter BW through BW_XL bits of CTRL1_XL (with XL_BW_SCAL_ODR =1 in CTRL4_C register)	Acc in LP/NM mode: first sample correct Acc in HP mode: see *Table 17*	First sample correct

Table 17. Accelerometer number of samples to be discarded

Accelerometer ODR [Hz]	BW = 400 Hz	BW = 200 Hz	BW = 100 Hz	BW = 50 Hz	No filter
12.5 Hz (High Perf.)	1	1	1	1	N.A.
26 Hz (High Perf.)	1	1	1	1	N.A.
52 Hz (High Perf.)	1	1	1	1	N.A.
104 Hz (High Perf.)	1	1	1	2	N.A.
208 Hz (High Perf.)	1	1	2	4	N.A.
416 Hz (High Perf.)	1	2	3	7	N.A.
833 Hz (High Perf.)	2	4	7	14	N.A.
1.66 kHz (High Perf.)	4	8	14	28	N.A.
3.33 kHz (High Perf.)	8	16	28	56	2
6.66 kHz (High Perf.)	16	32	56	112	4

Table 18. Gyroscope number of samples to be discarded

Gyroscope ODR [Hz]	Number of samples
12.5 Hz	2
26 Hz	2
52 Hz	2
104 Hz	2
208 Hz	2
416 Hz	2
833 Hz	3
1.66 kHz	4

4 Reading output data

4.1 Startup sequence

Once the device is powered up, it automatically downloads the calibration coefficients from the embedded flash to the internal registers. When the boot procedure is completed, i.e. after approximately 20 milliseconds, the accelerometer and gyroscope automatically enter Power-Down mode.

To turn on the accelerometer and gather acceleration data, it is necessary to select one of the operating modes through the CTRL1_XL register and to enable at least one of the axes through CTRL9_XL.

The following general-purpose sequence can be used to configure the accelerometer:

1	Write CTRL9_XL = 38h	// Acc X, Y, Z axes enabled
2	Write CTRL1_XL = 60h	// Acc = 416Hz (High-Performance mode)
3	Write INT1_CTRL = 01h	// Acc Data Ready interrupt on INT1

To turn on the gyroscope and gather angular rate data, it is necessary to select one of the operating modes through the CTRL2_G register and to enable at least one of the axes through CTRL10_C.

The following general-purpose sequence can be used to configure the gyroscope:

1	Write CTRL10_C = 38h	// Gyro X, Y, Z axes enabled
2	Write CTRL2_G = 60h	// Gyro = 416Hz (High-Performance mode)
3	Write INT1_CTRL = 02h	// Gyro Data Ready interrupt on INT1

4.2 Using the status register

The device is provided with a STATUS_REG register which should be polled to check when a new set of data is available. The XLDA bit is set to 1 when a new set of data is available at accelerometer output; the GDA bit is set to 1 when a new set of data is available at gyroscope output.

For the accelerometer (the gyroscope is similar), the reads should be performed as follows:

1. Read STATUS
2. If XLDA = 0, then go to 1
3. Read OUTX_L_XL
4. Read OUTX_H_XL
5. Read OUTY_L_XL
6. Read OUTY_H_XL
7. Read OUTZ_L_XL
8. Read OUTZ_H_XL
9. Data processing
10. Go to 1

天猫店铺网址:https://telesky.tmall.com/

4.3 Using the data-ready signal

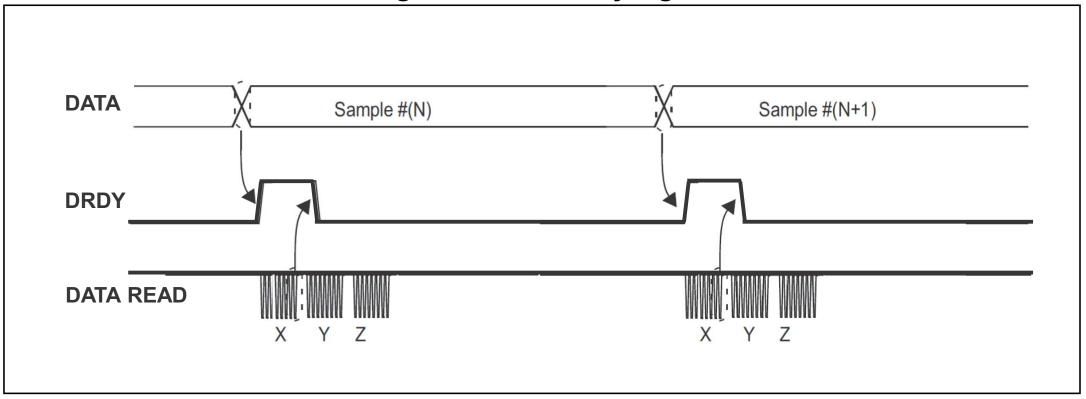
The device can be configured to have one HW signal to determine when a new set of measurement data is available for reading.

For the accelerometer sensor, the data-ready signal is represented by the XLDA bit of the STATUS_REG register. The signal can be driven to the INT1 pin by setting to 1 the INT1_DRDY_XL bit of the INT1_CTRL register and to the INT2 pin by setting to 1 the INT2_DRDY_XL bit of the INT2_CTRL register.

For the gyroscope sensor, the data-ready signal is represented by the GDA bit of the STATUS_REG register. The signal can be driven to the INT1 pin by setting to 1 the INT1_DRDY_G bit of the INT1_CTRL register and to the INT2 pin by setting to 1 the INT2_DRDY_G bit of the INT2_CTRL register.

The data-ready signal rises to 1 when a new set of data has been generated and it is available for reading. The interrupt is reset when the higher part of one of the enabled channels has been read (29h, 2Bh, 2Dh for the accelerometer; 23h, 25h, 27h for the gyroscope).

Figure 9. Data-ready signal



4.3.1 DRDY mask functionality

Setting the DRDY_MASK bit of the CTRL4_C register to 1, the accelerometer and gyroscope data-ready signals are masked until the settling of the sensor filters is completed.

The DRDY_MASK bit masks the samples to be discarded only when switching from Power-Down to an active mode, it doesn't mask them on ODR changes. Furthermore, only the data-ready signals are masked, whereas the output registers are continuously updated also during the settling period of the sensor filters.

When FIFO is active and the DRDY_MASK bit is set to 1, accelerometer invalid samples stored in FIFO are always equal to 7FFFh; gyroscope invalid samples stored in FIFO can be equal to 7FFFh, 7FFEh or 7FFDh. In this way, a tag is applied to the invalid samples stored in the FIFO buffer, so that they can be easily identified and discarded during data post-processing.

4.4 Using the block data update (BDU) feature

If reading the accelerometer/gyroscope data is particularly slow and cannot be synchronized (or it is not required) with either the XLDA/GDA bits in the STATUS_REG register or with the DRDY signal driven to the INT1/INT2 pins, it is strongly recommended to set the BDU (block data update) bit to 1 in the CTRL3_C register.

This feature avoids reading values (most significant and least significant parts of output data) related to different samples. In particular, when the BDU is activated, the data registers related to each channel always contain the most recent output data produced by the device, but, in case the read of a given pair (i.e. OUTX_H_XL(G) and OUTX_L_XL(G), OUTY_H_XL(G) and OUTY_L_XL(G), OUTZ_H_XL(G) and OUTZ_L_XL(G)) is initiated, the refresh for that pair is blocked until both MSB and LSB parts of the data are read.

Note: BDU only guarantees that the LSB part and MSB part have been sampled at the same moment. For example, if the reading speed is too slow, X and Y can be read at T1 and Z sampled at T2.

4.5 Understanding output data

The measured acceleration data are sent to the OUTX_H_XL, OUTX_L_XL, OUTY_H_XL, OUTY_L_XL, OUTZ_H_XL, and OUTZ_L_XL registers. These registers contain, respectively, the most significant part and the least significant part of the acceleration signals acting on the X, Y, and Z axes.

The measured angular rate data are sent to the OUTX_H_G, OUTX_L_G, OUTY_H_G, OUTY_L_G, OUTZ_H_G, and OUTZ_L_G registers. These registers contain, respectively, the most significant part and the least significant part of the angular rate signals acting on the X, Y, and Z axes.

The complete output data for the X, Y, Z channels is given by the concatenation OUTX_H_XL(G) & OUTX_L_XL(G), OUTY_H_XL(G) & OUTY_L_XL(G) , OUTZ_H_XL(G) & OUTZ_L_XL(G) and it is expressed as a two’s complement number.

Both acceleration data and angular rate data are represented as 16-bit numbers.

4.5.1 Big-little endian selection

The LSM6DS3 allows swapping the content of the lower and the upper part of the output data registers (i.e. OUTX_H_XL(G) with OUTX_L_XL(G), and OUT_TEMP_H with OUT_TEMP_L) in order to be compliant with both little-endian and big-endian data representations.

“Little Endian” means that the low-order byte of the number is stored in memory at the lowest address, and the high-order byte at the highest address. This mode corresponds to the BLE bit of the CTRL3_C register set to 0 (default configuration).

On the contrary, “Big Endian” means that the high-order byte of the number is stored in memory at the lowest address, and the low-order byte at the highest address. This mode corresponds to the BLE bit of the CTRL3_C register set to 1.

天猫店铺网址:https://telesky.tmall.com/

4.5.2 Examples of output data

Table 19 provides a few basic examples of the accelerometer data that is read in the data registers when the device is subject to a given acceleration.

Table 20 provides a few basic examples of the gyroscope data that is read in the data registers when the device is subject to a given angular rate.

The values listed in the following tables are given under the hypothesis of perfect device calibration (i.e. no offset, no gain error,....) and practically show the effect of the BLE bit.

Table 19. Output data registers content vs. acceleration (FS_XL = 2 *g*)

Acceleration values	BLE = 0		BLE = 1	
	Register address			
	OUTX_H_XL (29h)	OUTX_L_XL (28h)	OUTX_H_XL (29h)	OUTX_L_XL (28h)
0 *g*	00h	00h	00h	00h
350 m*g*	16h	69h	69h	16h
1 *g*	40h	09h	09h	40h
-350 m*g*	E9h	97h	97h	E9h
-1 *g*	BFh	F7h	F7h	BFh

Table 20. Output data registers content vs. angular rate (FS_G = 245 dps)

Angular rate values	BLE = 0		BLE = 1	
	Register address			
	OUTX_H_G (23h)	OUTX_L_G (22h)	OUTX_H_G (23h)	OUTX_L_G (22h)
0 dps	00h	00h	00h	00h
100 dps	2Ch	A4h	A4h	2Ch
200 dps	59h	49h	49h	59h
-100 dps	D3h	5Ch	5Ch	D3h
-200 dps	A6h	B7h	B7h	A6h

4.6 Rounding functions

The rounding function can be used to auto address the LSM6DS3 registers for a circular burst-mode read. Basically, with a multiple read operation the address of the register that is being read goes automatically from the first register to the last register of the pattern and then goes back to the first one.

4.6.1 Rounding of FIFO output registers

The rounding function is automatically enabled when performing a multiple read operation of the FIFO output registers FIFO_DATA_OUT_L (3Eh) and FIFO_DATA_OUT_H (3Fh).

4.6.2 Rounding of source registers

It's possible to apply the rounding function also to the source registers of the LSM6DS3 device, in order to verify with one multiple read whether new data was generated or a new interrupt event was detected.

The rounding function on the source registers can be enabled by setting to 1 the ROUNDING_STATUS bit of the CTRL7_G register: when this function is enabled, with a multiple read operation the address of the register that is being read goes automatically from STATUS_REG (1Eh) to FUNC_SRC (53h) and goes back to WAKE_UP_SRC (1Bh).

4.6.3 Rounding of sensor output registers

The rounding function can also be enabled for the following groups of output registers:

- Gyroscope output registers, from OUTX_L_G (22h) to OUTZ_H_G (27h);
- Accelerometer output registers, from OUTX_L_XL (28h) to OUTZ_H_XL (2Dh);
- First group of sensor hub output registers, from SENSORHUB1_REG (2Eh) to SENSORHUB6_REG (33h);
- Second group of sensor hub output registers, from SENSORHUB7_REG (34h) to SENSORHUB12_REG (39h).

The output registers rounding pattern can be configured using the bits ROUNDING[2:0] of the CTRL5_C register, as indicated in *Table 21*.

Table 21. Output registers rounding pattern

ROUNDING[2:0]	Rounding pattern
000	No rounding
001	Accelerometer only
010	Gyroscope only
011	Gyroscope + Accelerometer
100	1st group of Sensor Hub only
101	Accelerometer + 1st group of Sensor Hub
110	Gyroscope + Accelerometer + 1st group of Sensor Hub + 2nd group of Sensor Hub
111	Gyroscope + Accelerometer + 1st group of Sensor Hub

天猫店铺网址:https://telesky.tmall.com/

4.7 Gyroscope edge-sensitive/level-sensitive/impulse-sensitive data enable (DEN)

The LSM6DS3 allows external trigger level recognition by configuring the TRIG_EN, LVLen and LVL2_EN bits of the CTRL6_C register (*Table 22*). The default value for these three bits is 0 (external trigger is disabled). Three different trigger modes can be used: edge-, level-, or impulse-sensitive trigger; the Data Enable (DEN) input signal is driven on the INT2 pin, which is configured as an input pin when one of the gyroscope trigger modes is enabled.

Table 22. DEN configurations

TRIG_EN	LVLen	LVL2_EN	Function
1	0	0	Edge-sensitive
0	1	0	Level-sensitive
0	1	1	Impulse-sensitive

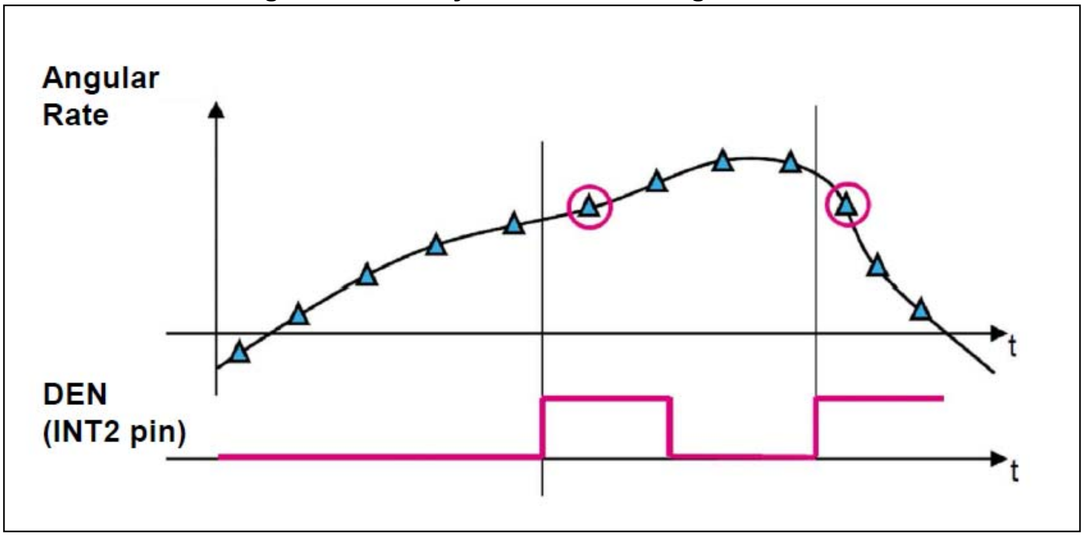
Edge-sensitive and impulse-sensitive triggers need both accelerometer and gyroscope sensors to be active (not in Power-Down mode) and configured with the same output data rate value. The level-sensitive trigger can also work with only the gyroscope in active mode, regardless of the selected gyroscope ODR.

4.7.1 Edge-sensitive trigger

The edge-sensitive trigger is enabled when the TRIG_EN bit of CTRL6_C register is set to 1 and the LVLen and LVL2_EN bits are set to 0. If the FIFO is not used, both accelerometer and gyroscope sensors have to be in active mode and configured with the same ODR value.

Once enabled, the gyroscope output registers are updated with the next generated X, Y, Z gyroscope data at the rising edge of the DEN (INT2 pin) input signal. If no rising edge occurs, the gyroscope output registers are not updated.

Figure 10. Data synchronization: edge-sensitive



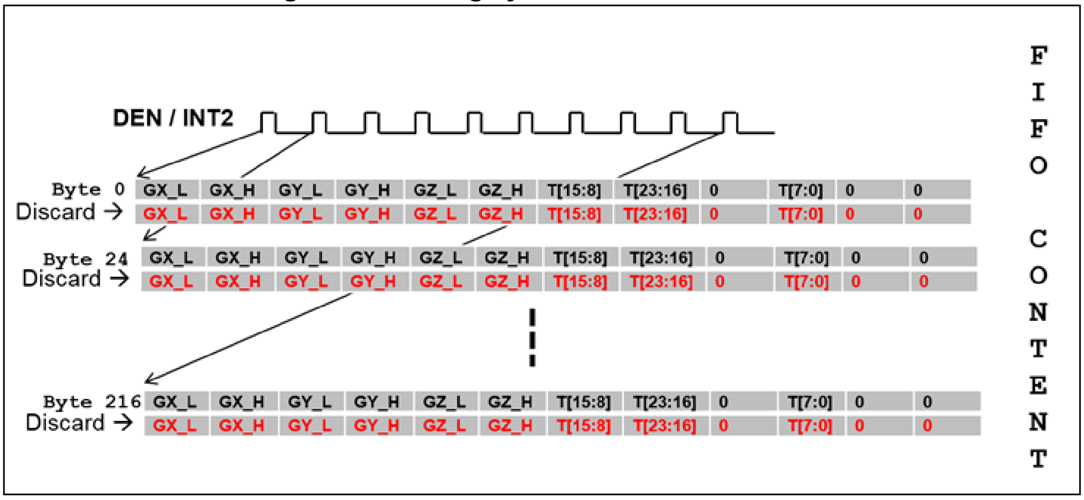
If both the edge-sensitive trigger and FIFO (see *Section 8: First-in first-out (FIFO) buffer*) are enabled, the DEN (INT2 pin) signal works as trigger signal for writing data in the FIFO buffer: FIFO patterns are stored in the FIFO on the rising edge of the DEN signal. In order to have this feature enabled, both the DATA_VALID_SEL_FIFO bit of the MASTER_CONFIG register and the TIMER_PEDO_FIFO_DRDY bit of the FIFO_CTRL2 register have to be set to 0. Furthermore, both accelerometer and gyroscope sensors have to be in active mode and the accelerometer ODR value has to be half of the gyroscope ODR value.

Note: Each pattern is stored in the FIFO buffer twice, so the second stored pattern has to be discarded.

This features allows, for example, the synchronization of the camera frames with the samples coming from the gyroscope for Electrical Image Stabilization (EIS) applications. The synchronization signal from the camera module must be connected to the INT2 pin.

In the example shown in *Figure 11* the FIFO has been configured to store in the FIFO buffer both the gyroscope data and the timestamp data; when the DEN signal toggles, the data are written to FIFO twice on the rising edge. The second set of data must be discarded.

Figure 11. Storing synchronized data in FIFO



The settings required for this example are as follows:

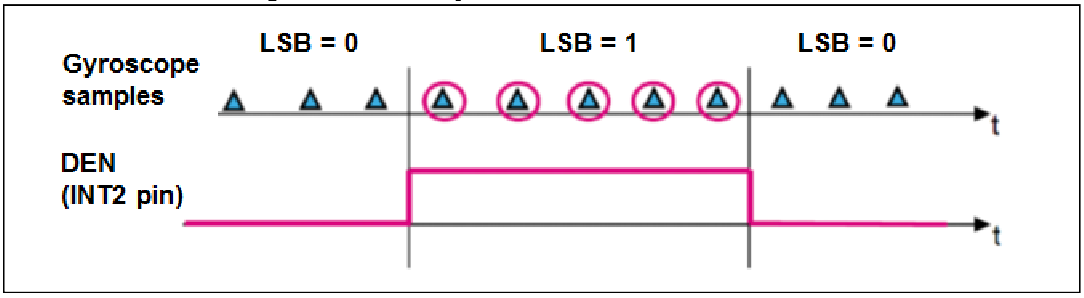
1	Write 08h into FIFO_CTRL3	// Enable gyroscope data in FIFO (no decimation)
2	Write 80h into FIFO_CTRL2	// Enable step counter and timestamp data as 4th FIFO data set
3	Write 08h into FIFO_CTRL4	// Enable step counter and timestamp in FIFO (no decimation)
4	Write 26h into FIFO_CTRL5	// Set FIFO in Continuous mode, FIFO ODR = 104Hz
5	Write 80h into CTRL6_G	// Enable the edge-sensitive trigger // INT2 pin is switched to input mode (DEN signal)
6	Write 30h into CTRL1_XL	// Turn on the accelerometer: ODR_XL = 52Hz, FS_XL = 2*g*
7	Write 4Ch into CTRL2_G	// Turn on the gyroscope // ODR_G = 104Hz, FS_G = 2000dps
8	Write 80h into TAP_CFG	// Enable timestamp counter

4.7.2 Level-sensitive trigger stamping

The level-sensitive trigger is enabled when the LVLen bit of the CTRL6_C register is set to 1 and the TRIG_EN and LVL2_EN bits are set to 0.

Once enabled, the LSB of the generated gyroscope X, Y, Z output data is replaced with the current DEN (INT2 pin) level. This replacement applies also to the data stored in FIFO.

Figure 12. Data synchronization: level-sensitive



4.7.3 Impulse-sensitive trigger stamping

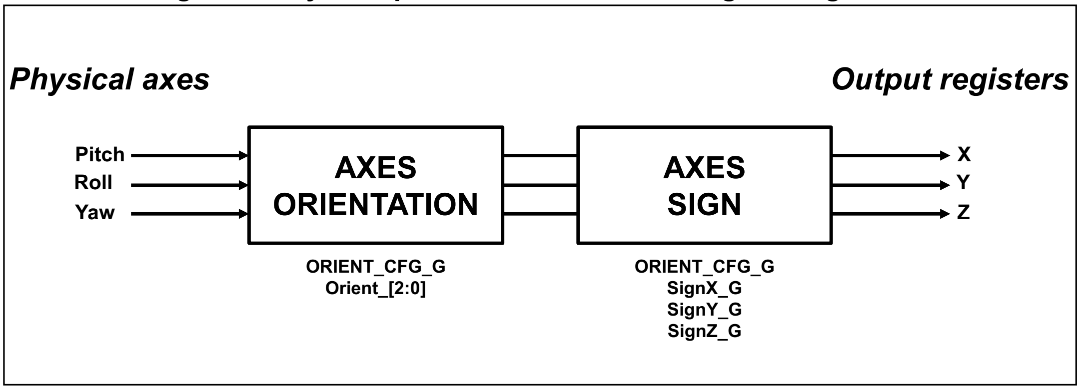
The impulse-sensitive trigger is enabled when the LVLen and LVL2_EN bits of the CTRL6_C register are set to 1 and the TRIG_EN bit is set to 0. Furthermore, both accelerometer and gyroscope sensors have to be in active mode and configured with the same ODR value.

The impulse-sensitive trigger is similar to the level-sensitive trigger and has to be used if the duration of the DEN positive pulse is shorter than the selected gyroscope ODR. Once enabled, the LSB bit of the gyroscope X, Y, Z output data generated after the pulse is set to 1. If no pulse occurs, the LSB bit of the next generated gyroscope X, Y, Z output data is set to 0. If the pulse occurs during the update of the data, it could happen that two consecutive gyroscope data are tagged instead of just the first one, and it's also possible that the tag of the LSB bit to 1 is not applied to all three axes of each X, Y, Z output data: in this case, one of the two consecutively tagged data has to be discarded at sw level.

4.8 Gyroscope axes orientation

Axes orientation and sign of the gyroscope sensor can be changed by software using the ORIENT_CFG_G register, as illustrated in *Figure 13*.

Figure 13. Gyroscope axes orientation and sign configuration



天猫店铺网址:https://telesky.tmall.com/

The Orient_[2:0] bits of the ORIENT_CFG_G register allows driving the pitch, roll and yaw physical axes to the X, Y and Z output registers as indicated in *Table 24*.

Table 23. ORIENT_CFG_G register

b7	b6	b5	b4	b3	b2	b1	b0
0	0	SignX_G	SignY_G	SignZ_G	Orient_2	Orient_1	Orient_0

Table 24. Settings for gyroscope axes orientation

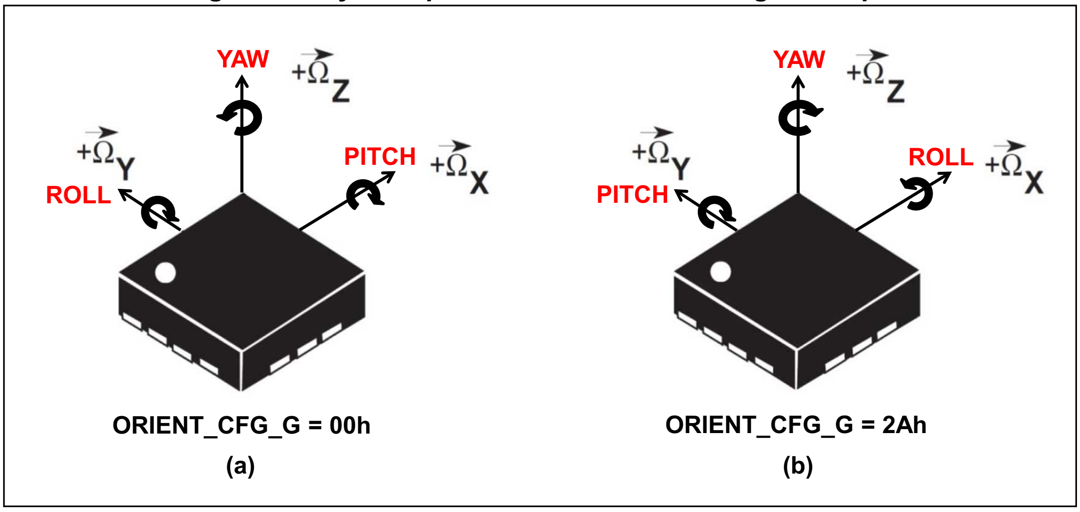
Orient_[2:0]	Pitch	Roll	Yaw
000	X	Y	Z
001	X	Z	Y
010	Y	X	Z
011	Y	Z	X
100	Z	X	Y
101	Z	Y	X

The SignX_G, SignY_G and SignZ_G bits of the ORIENT_CFG_G register allow respectively changing (setting the bit to 1) the sign of the X, Y and Z output registers data.

Case (a) of example in *Figure 14* corresponds to the default case for axes orientation and sign, with all Orient_[2:0], SignX_G, SignY_G and SignZ_G bits of the ORIENT_CFG_G register set to 0.

In case (b) the Orient_[2:0] bits have been set to 010b, driving Pitch and Roll data on different axes than the default case; furthermore the sign on X and Z data has been changed by setting the SignX_G and SignZ_G bits to 1.

Figure 14. Gyroscope axes orientation and sign example



天猫店铺网址:https://telesky.tmall.com/

5 Interrupt generation

In the LSM6DS3 device the interrupt generation is based on accelerometer data only, so, for interrupt generation purposes, the accelerometer sensor has to be set in an active operating mode (not in Power-Down); the gyroscope sensor can be configured in Power-Down mode since it's not involved in interrupt generation.

The interrupt generator can be configured to detect:

- Free-fall;
- Wake-up;
- 6D/4D orientation detection;
- Single-tap and double-tap sensing;
- Activity/Inactivity recognition.

In addition, the LSM6DS3 can efficiently run the sensor-related features specified in Android, saving power and enabling faster reaction time. In particular, it has been designed to implement in hardware:

- Significant motion;
- Tilt;
- Pedometer functions;
- Timestamp.

All these interrupt signals, together with FIFO interrupt signals, can be independently driven to the INT1 and INT2 interrupt pins or checked by reading the dedicated source register bits.

The H_LACTIVE bit of the CTRL3_C register must be used to select the polarity of the interrupt pins. If this bit is set to 0 (default value), the interrupt pins are active high and they change from low to high level when the related interrupt condition is verified. Otherwise, if the H_LACTIVE bit is set to 1 (active low), the interrupt pins are normally at high level and they change from high to low when interrupt condition is reached.

The PP_OD bit of CTR3_C allows changing the behavior of the interrupt pins from push-pull to open drain. If the PP_OD bit is set to 0, the interrupt pins are in push-pull configuration (low-impedance output for both high and low level). When the PP_OD bit is set to 1, only the interrupt active state is a low-impedance output.

The LIR bit of TAP_CFG allows applying the latched mode to the interrupt signals. When the LIR bit is set to 1, once the interrupt pin is asserted, it must be reset by reading the related interrupt source register. If the LIR bit is set to 0, the interrupt signal is automatically reset when the interrupt condition is no longer verified or after a certain amount of time.

5.1 Interrupt pin configuration

The device is provided with two pins that can be activated to generate either Data Ready or interrupt signals. The functionality of these pins is selected through the MD1_CFG and INT1_CTRL registers for the INT1 pin, and through the MD2_CFG and INT2_CTRL registers for the INT2 pin.

A brief description of these interrupt control registers is given in the following summary; the default value of their bits is equal to 0, which corresponds to 'disable'. In order to enable the routing of a specific interrupt signal on the pin, the related bit has to be set to 1.

Table 25. INT1_CTRL register

b7	b6	b5	b4	b3	b2	b1	b0
INT1_STEP_ DETECTOR	INT1_ SIGN_MOT	INT1_ FULL_FLAG	INT1_ FIFO_OVR	INT1_ FTH	INT1_ BOOT	INT1_ DRDY_G	INT1_ DRDY_XL

- INT1_STEP_DETECTOR: Pedometer step recognition interrupt on INT1.
- INT1_SIGN_MOT: Significant motion interrupt on INT1.
- INT1_FULL_FLAG: FIFO full flag interrupt on INT1.
- INT1_FIFO_OVR: FIFO overrun flag interrupt on INT1.
- INT1_FTH: FIFO threshold interrupt on INT1.
- INT1_BOOT: Boot interrupt on INT1.
- INT1_DRDY_G: Gyroscope Data-Ready on INT1.
- INT1_DRDY_XL: Accelerometer Data-Ready on INT1.

Table 26. MD1_CFG register

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

- INT1_INACT_STATE: Inactivity interrupt on INT1.
- INT1_SINGLE_TAP: Single-tap interrupt on INT1.
- INT1_WU: Wake-up interrupt on INT1.
- INT1_FF: Free-fall interrupt on INT1.
- INT1_DOUBLE_TAP: Double-tap interrupt on INT1.
- INT1_6D: 6D detection interrupt on INT1.
- INT1_TILT: Tilt interrupt on INT1.
- INT1_TIMER: Timer interrupt on INT1.

Table 27. INT2_CTRL register

b7	b6	b5	b4	b3	b2	b1	b0
INT2_STEP_ DELTA	INT2_STEP_ COUNT_OV	INT2_FUL L_FLAG	INT2_FIF O_OVR	INT2_ FTH	INT2_DRD Y_TEMP	INT2_ DRDY_G	INT2_ DRDY_XL

- INT2_STEP_DELTA: Pedometer step recognition on delta time interrupt on INT2.
- INT2_STEP_COUNT_OV: Step counter overflow interrupt on INT2.
- INT2_FULL_FLAG: FIFO full flag interrupt on INT2.
- INT2_FIFO_OVR: FIFO overrun flag interrupt on INT2.
- INT2_FTH: FIFO threshold interrupt on INT2.
- INT2_DRDY_TEMP: Temperature Data-Ready on INT2.
- INT2_DRDY_G: Gyroscope Data-Ready on INT2.
- INT2_DRDY_XL: Accelerometer Data-Ready on INT2.

Table 28. MD2_CFG register

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

- INT2_INACT_STATE: Inactivity interrupt on INT2.
- INT2_SINGLE_TAP: Single-tap interrupt on INT2.
- INT2_WU: Wake-up interrupt on INT2.
- INT2_FF: Free-fall interrupt on INT2.
- INT2_DOUBLE_TAP: Double-tap interrupt on INT2.
- INT2_6D: 6D detection interrupt on INT2.
- INT2_TILT: Tilt interrupt on INT2.
- INT2_IRON: Soft-iron / hard-iron interrupt on INT2.

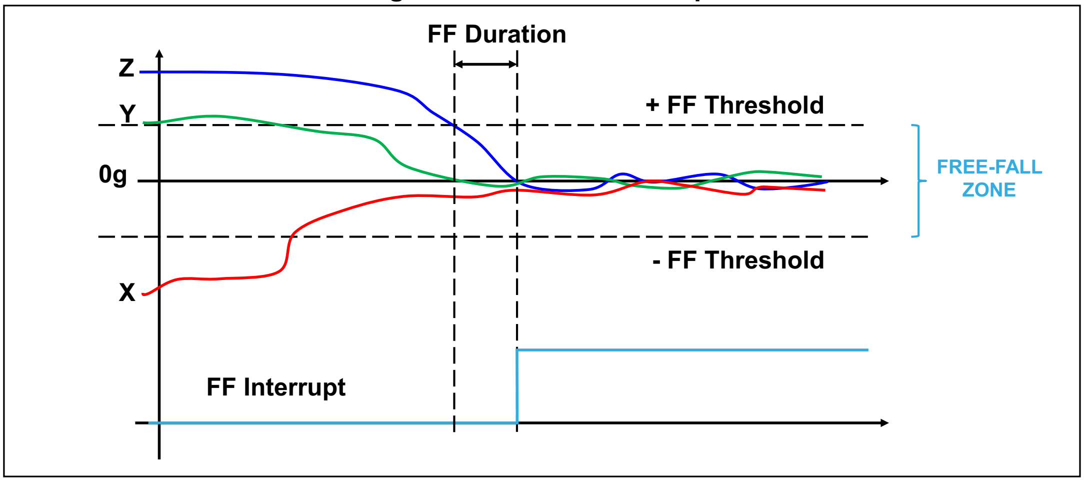
If multiple interrupt signals are routed on the same pin (INTx), the logic level of this pin is the "OR" combination of the selected interrupt signals. In order to know which event has generated the interrupt condition, the related source registers have to be read: WAKE_UP_SRC, D6D_SRC, TAP_SRC and FUNC_SRC.

The INT2_on_INT1 pin of CTRL4_C register allows driving all the enabled interrupt signals in logic "OR" on the INT1 pin (by setting this bit to 1). When this bit is set to 0, the interrupt signals are divided between the INT1 and INT2 pins.

5.2 Free-fall interrupt

Free-fall detection refers to a specific register configuration that allows recognizing when the device is in free-fall: the acceleration measured along all the axes goes to zero. In a real case a "free-fall zone" is defined around the zero-*g* level where all the accelerations are small enough to generate the interrupt. Configurable threshold and duration parameters are associated to free-fall event detection: the threshold parameter defines the free-fall zone amplitude; the duration parameter defines the minimum duration of the free-fall interrupt event to be recognized (*Figure 15*).

Figure 15. Free-fall interrupt



天猫店铺网址:https://telesky.tmall.com/

The free-fall interrupt signal can be driven to the two interrupt pins by setting to 1 the INT1_FF bit of the MD1_CFG register or the INT2_FF bit of the MD2_CFG register; it can also be checked by reading the FF_IA bit of the WAKE_UP_SRC register.

If latch mode is disabled (LIR bit of TAP_CFG is set to 0), the interrupt signal is automatically reset when the free-fall condition is no longer verified. If latch mode is enabled and the free-fall interrupt signal is driven to the interrupt pins, once a free-fall event has occurred and the interrupt pin is asserted, it must be reset by reading the WAKE_UP_SRC register. If the latch mode is enabled but the interrupt signal is not driven to the interrupt pins, the latch feature does not take effect.

The register used to configure the threshold parameter is named FREE_FALL; the unsigned threshold value is related to the value of the FF_THS[2:0] field value as indicated in *Table 29*. The values given in this table are valid for each accelerometer full-scale value.

Table 29. Free-fall threshold LSB value

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

Duration time is measured in N/ODR_XL, where N is the content of the FF_DUR[5:0] field of the FREE_FALL / WAKE_UP_DUR registers and ODR_XL is the accelerometer data rate.

A basic SW routine for the free-fall event recognition is given below.

1	Write 60h into CTRL1_XL	// Turn on the accelerometer // ODR_XL = 416 Hz, FS_XL = 2g
2	Write 00h into WAKE_UP_DUR	// Set event duration (FF_DUR5 bit)
3	Write 33h into FREE_FALL	// Set FF threshold (FF_THS[2:0] = 011b) // Set six samples event duration (FF_DUR[5:0] = 000110b)
4	Write 10h into MD1_CFG	// FF interrupt driven to INT1 pin
5	Write 01h into TAP_CFG	// Latch interrupt

The sample code exploits a threshold set to 312 m*g* for free-fall recognition and the event is notified by hardware through the INT1 pin. The FF_DUR[5:0] field of FREE_FALL / WAKE_UP_DUR registers is configured like this to ignore events that are shorter than 6/ODR_XL = 6/412 Hz ~= 15 msec in order to avoid false detections.

5.3 Wake-up interrupt

In the LSM6DS3 device the wake-up feature can be implemented using either the slope filter (see *Section 3.7.1* for more details) or the high-pass digital filter, as illustrated in *Figure 5*. The filter to be applied can be selected using the SLOPE_FDS bit of the TAP_CFG register: if this bit is set to 0 (default value), the slope filter is used; if it's set to 1, the high-pass digital filter is used.

The wake-up interrupt signal is generated if a certain number of consecutive filtered data exceed the configured threshold (*Figure 16*).

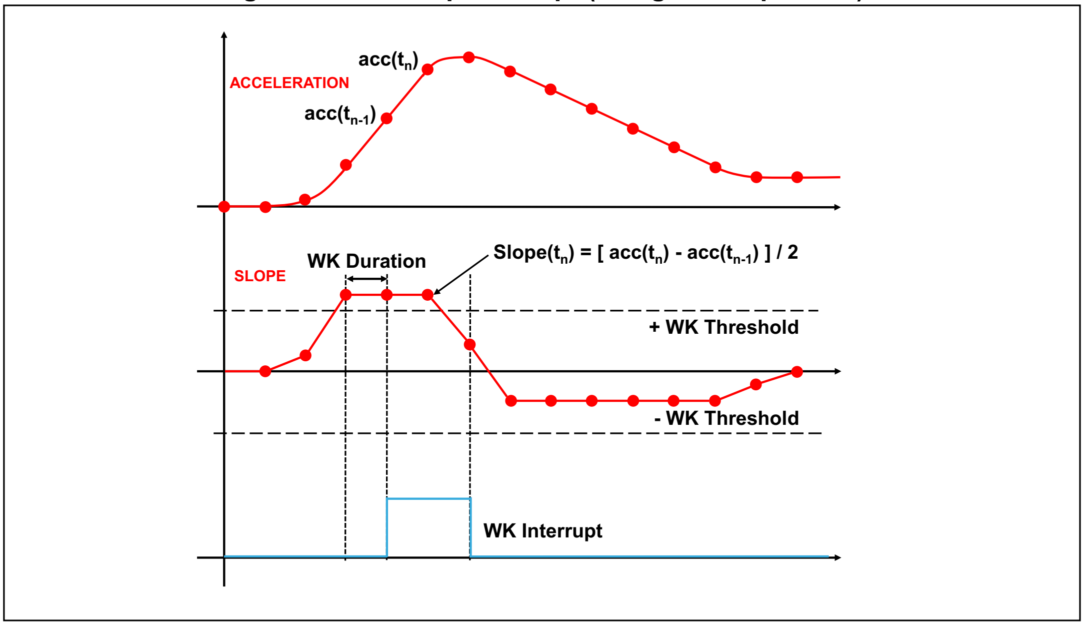
The unsigned threshold value is defined using the WK_THS[5:0] bits of the WAKE_UP_THS register; the value of 1 LSB of these 6 bits depends on the selected accelerometer full scale: 1 LSB = (FS_XL)/(2^6). The threshold is applied to both positive and negative data: for a wake-up interrupt generation, the module of the filtered data must be bigger than the threshold.

The duration parameter defines the minimum duration of the wake-up event to be recognized; its value is set using the WAKE_DUR[1:0] bits of the WAKE_UP_DUR register: 1 LSB corresponds to 1*ODR_XL time, where ODR_XL is the accelerometer output data rate. It is important to appropriately define the duration parameter to avoid unwanted wake-up interrupts due to spurious spikes of the input signal.

This interrupt signal can be driven to the two interrupt pins setting to 1 the INT1_WU bit of the MD1_CFG register or the INT2_WU bit of the MD2_CFG register; it can also be checked by reading the WU_IA bit of the WAKE_UP_SRC register. The X_WU, Y_WU, Z_WU bits of the WAKE_UP_SRC register indicate which axis has triggered the wake-up event.

If latch mode is disabled (LIR bit of TAP_CFG is set to 0), the interrupt signal is automatically reset when the filtered data falls below the threshold. If latch mode is enabled and the wake-up interrupt signal is driven to the interrupt pins, once a wake-up event has occurred and the interrupt pin is asserted, it must be reset by reading the WAKE_UP_SRC register. If the latch mode is enabled but the interrupt signal is not driven to the interrupt pins, the latch feature does not take effect.

Figure 16. Wake-up interrupt (using the slope filter)



天猫店铺网址:https://telesky.tmall.com/

The example code which implements the SW routine for the wake-up event recognition using the high-pass digital filter is given below.

1	Write 60h into CTRL1_XL	// Turn on the accelerometer // ODR_XL = 416Hz, FS_XL = 2g
2	Write 10h into TAP_CFG	// Apply high-pass digital filter; latch mode disabled
3	Write 00h into WAKE_UP_DUR	// No duration
4	Write 02h into WAKE_UP_THS	// Set wake-up threshold
5	Write 20h into MD1_CFG	// Wake-up interrupt driven to INT1 pin

Since the duration time is set to zero, the wake-up interrupt signal is generated for each X,Y,Z filtered data exceeding the configured threshold. The WK_THS field of the WAKE_UP_THS register is set to 000010b, therefore the wake-up threshold is 62.5 m*g* (= 2 * FS_XL / 2^6).

Since the wake-up functionality is implemented using the slope/high-pass digital filter, it's necessary to consider the settling time of the filter just after this functionality is enabled. For example, when using the slope filter (but a similar consideration can be done for the high-pass digital filter usage) the wake-up functionality is based on the comparison of the threshold value with half of the difference of the acceleration of the current (x,y,z) sample and the previous one (refer to *Section 3.7.1: Accelerometer slope filter*).

At the very first sample, the slope filter output is calculated as half of the difference of the current sample [e.g. (x,y,z) = (0,0,1*g*)] with the previous one which is (x,y,z)=(0,0,0) since it doesn't exist. For this reason, on the z-axis the first output value of the slope filter is (1g - 0)/2=500m*g* and it could be higher than the threshold value in which case a spurious interrupt event is generated. The interrupt signal is kept high for 1 ODR then it goes low.

In order to avoid this spurious interrupt generation, multiple solutions are possible. Hereafter three alternative solutions (for slope filter case):

a) Ignore the first generated wake-up signal;

b) Add a wait time higher than 1 ODR before driving the interrupt signal to the INT1/2 pin;

c) Initially set a higher ODR (833 Hz) so the first 2 samples are generated in a shorter period of time, reducing the slope filter latency time, then set the desired ODR (e.g. 12.5 Hz) and drive the interrupt signal on the pin as indicated in the following procedure:

1	Write 00h into WAKE_UP_DUR	// No duration
2	Write 02h into WAKE_UP_THS	// Set wake-up threshold
3	Write 00h into TAP_CFG	// Apply slope filter; latch mode disabled
4	Write 70h into CTRL1_XL	// Turn on the accelerometer // ODR_XL = 833Hz, FS_XL = 2*g*
5	Wait 4ms	// Insert (reduced) wait time
6	Write 10h into CTRL1_XL	// ODR_XL = 12.5Hz
7	Write 20h into MD1_CFG	// Wake-up interrupt driven to INT1 pin

5.4 6D/4D orientation detection

The LSM6DS3 device provides the capability to detect the orientation of the device in space, enabling easy implementation of energy-saving procedures and automatic image rotation for mobile devices.

5.4.1 6D orientation detection

Six orientations of the device in space can be detected; the interrupt signal is asserted when the device switches from one orientation to another. The interrupt is not re-asserted as long as the position is maintained.

6D interrupt is generated when, for two consecutive samples, only one axis exceeds a selected threshold and the acceleration values measured from the other two axes are lower than the threshold: the ZH, ZL, YH, YL, XH, XL bits of the D6D_SRC (1Dh) register indicate which axis has triggered the 6D event.

In more detail:

Table 30. D6D_SRC register

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

- D6D_IA is set high when the device switches from one orientation to another.
- ZH (YH, XH) is set high when the face perpendicular to the Z(Y,X) axis is almost flat and the acceleration measured on the Z(Y,X) axis is positive and in the module bigger than the threshold.
- ZL (YL, XL) is set high when the face perpendicular to the Z(Y,X) axis is almost flat and the acceleration measured on the Z(Y,X) axis is negative and in the module bigger than the threshold.

The SIXD_THS[1:0] bits of the TAP_THS_6D register are used to select the threshold value used to detect the change in device orientation. The threshold values given in *Table 31* are valid for each accelerometer full-scale value.

Table 31. Threshold for 4D/6D function

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

The low-pass filter LPF2 can also be used in 6D functionality by setting the LOW_PASS_ON_6D bit of the CTRL8_XL register to 1. The LPF2 filter has to be enabled as described in *Section 3.7*.

This interrupt signal can be driven to the two interrupt pins by setting to 1 the INT1_6D bit of the MD1_CFG register or the INT2_6D bit of the MD2_CFG register; it can also be checked by reading the D6D_IA bit of the D6D_SRC register.

If latch mode is disabled (LIR bit of TAP_CFG is set to 0), the interrupt signal is active only for 1/ODR_XL[s] then it is automatically deasserted (ODR_XL is the accelerometer output data rate). If latch mode is enabled and the 6D interrupt signal is driven to the interrupt pins, once an orientation change has occurred and the interrupt pin is asserted, a reading of the D6D_SRC register clears the request and the device is ready to recognize a different orientation. If the latch mode is enabled but the interrupt signal is not driven to the interrupt pins, the latch feature does not take effect.

Referring to the six possible cases illustrated in *Figure 17*, the content of the D6D_SRC register for each position is shown in *Table 32*.

Figure 17. 6D recognized orientations

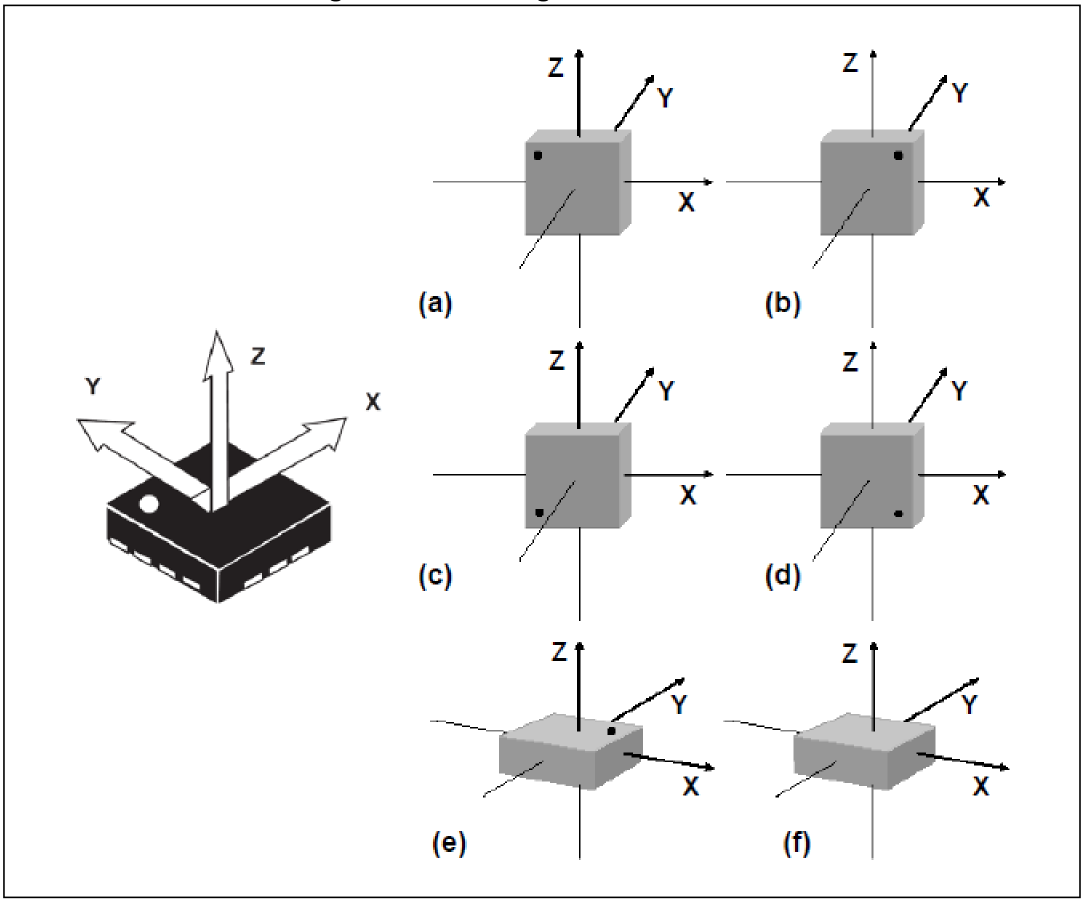


Table 32. D6D_SRC register in 6D positions

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

Hereafter an example which implements the SW routine for 6D orientation detection.

1	Write 60h into CTRL1_XL	// Turn on the accelerometer // ODR_XL = 416 Hz, FS_XL = 2g
2	Write 40h into TAP_THS_6D	// Set 6D threshold (SIXD_THS[1:0] = 10b = 60 degrees)
3	Write 10h into TAP_CFG	// Enable LPF2 filter
4	Write 01h into CTRL8_XL	// Apply LPF2 filter to 6D functionality
5	Write 04h into MD1_CFG	// 6D interrupt driven to INT1 pin

5.4.2 4D orientation detection

The 4D direction function is a subset of the 6D function especially defined to be implemented in mobile devices for portrait and landscape computation. It can be enabled by setting the D4D_EN bit of the TAP_THS_6D register to 1. In this configuration, the Z-axis position detection is disabled, therefore reducing position recognition to cases (a), (b), (c), and (d) of *Table 32*.

5.5 Single-tap and double-tap recognition

The single-tap and double-tap recognition functions featured in the LSM6DS3 help to create a man-machine interface with little software loading. The device can be configured to output an interrupt signal on a dedicated pin when tapped in any direction.

If the sensor is exposed to a single input stimulus, it generates an interrupt request on the inertial interrupt pin INT1 and/or INT2. A more advanced feature allows the generation of an interrupt request when a double input stimulus with programmable time between the two events is recognized, enabling a mouse button-like function.

In the LSM6DS3 device the single-tap and double-tap recognition functions use the slope between two consecutive acceleration samples to detect the tap events; the slope data is computed using the following formula:

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

This function can be fully programmed by the user in terms of expected amplitude and timing of the slope data by means of a dedicated set of registers.

Single and double-tap recognition work independently of the selected output data rate. Recommended accelerometer ODRs for these functions are 416 Hz and 833 Hz.

5.5.1 Single tap

If the device is configured for single-tap event detection, an interrupt is generated when the slope data of the selected channel exceeds the programmed threshold, and returns below it within the Shock time window.

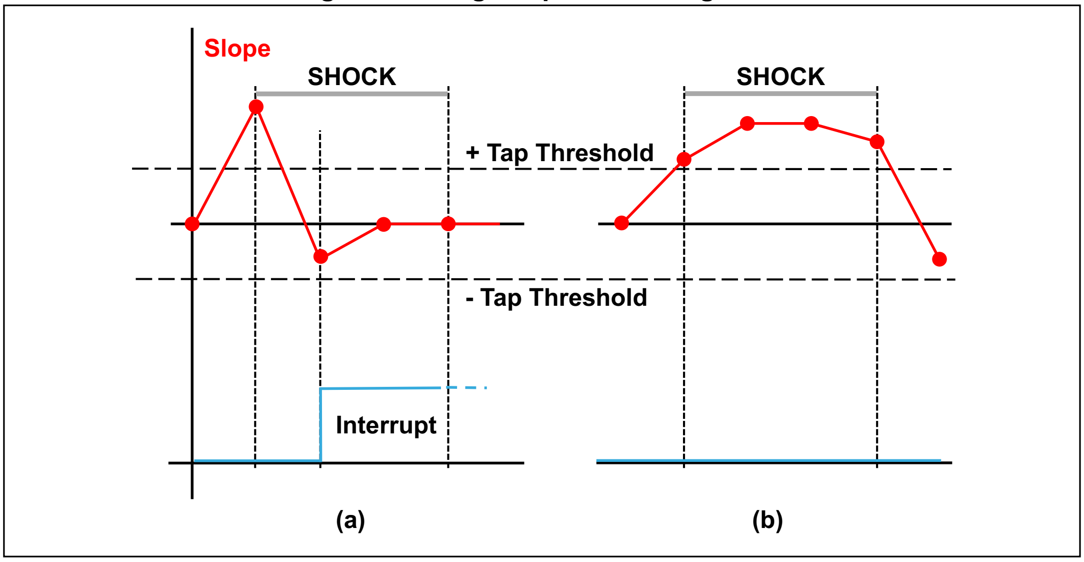
In the single-tap case, if the LIR bit of the TAP_CFG register is set to 0, the interrupt is kept high for the duration of the Quiet window.

In order to enable the latch feature on the single-tap interrupt signal, both the LIR bit and the INT1_DOUBLE_TAP (or INT2_DOUBLE_TAP) bit of MD1_CFG (MD2_CFG) have to be set to 1: the interrupt is kept high until the TAP_SRC register is read.

The SINGLE_DOUBLE_TAP bit of WAKE_UP_THS has to be set to 0 in order to enable single-tap recognition only.

In case (a) of *Figure 18* the single-tap event has been recognized, while in case (b) the tap has not been recognized because the slope data falls below the threshold after the Shock time window has expired.

Figure 18. Single-tap event recognition



5.5.2 Double tap

If the device is configured for double-tap event detection, an interrupt is generated when, after a first tap, a second tap is recognized. The recognition of the second tap occurs only if the event satisfies the rules defined by the Shock, the Latency and the Duration time windows.

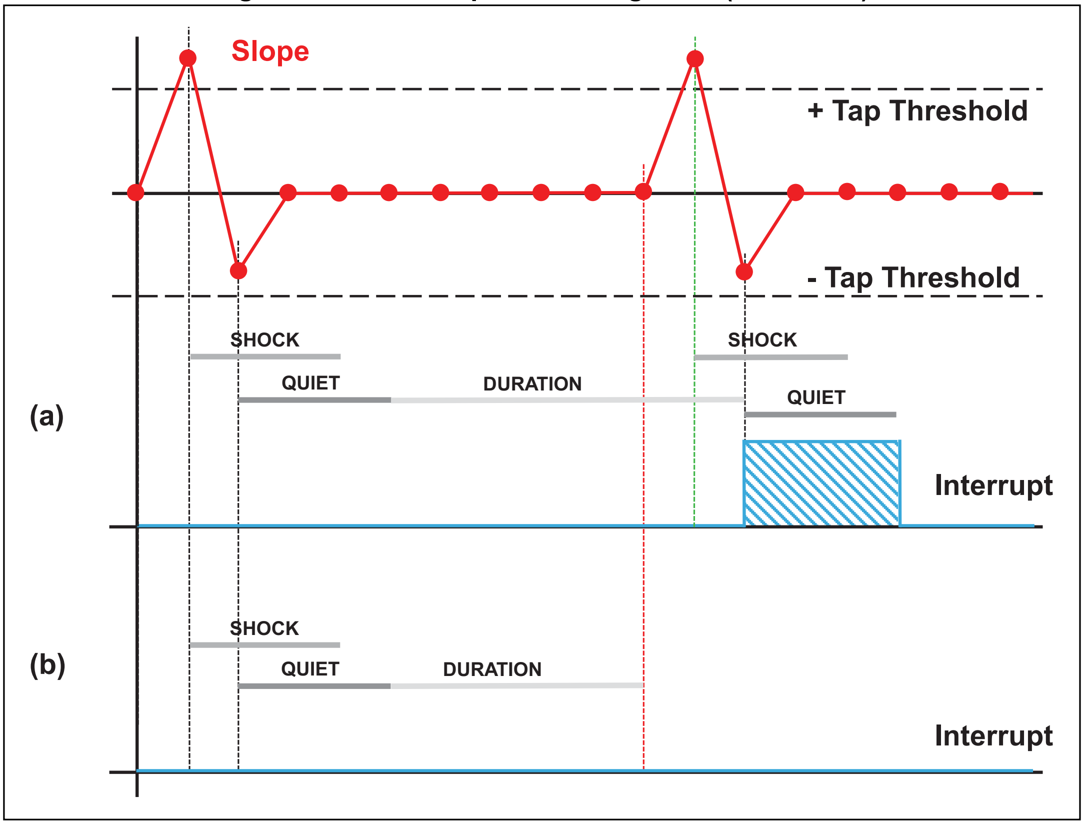
In particular, after the first tap has been recognized, the second tap detection procedure is delayed for an interval defined by the Quiet time. This means that after the first tap has been recognized, the second tap detection procedure starts only if the slope data exceed the threshold after the Quiet window but before the Duration window has expired. In case (a) of *Figure 19*, a double-tap event has been correctly recognized, while in case (b) the interrupt has not been generated because the slope data exceeds the threshold after the window interval has expired.

Once the second tap detection procedure is initiated, the second tap is recognized with the same rule as the first: the slope data must return below the threshold before the Shock window has expired.

It is important to appropriately define the Quiet window to avoid unwanted taps due to spurious bouncing of the input signal.

In the double-tap case, if the LIR bit of the TAP_CFG register is set to 0, the interrupt is kept high for the duration of the Quiet window. If the LIR bit is set to 1, the interrupt is kept high until the TAP_SRC register is read.

Figure 19. Double-tap event recognition (LIR bit = 0)



5.5.3 Single-tap and double-tap recognition configuration

The LSM6DS3 device can be configured to output an interrupt signal when tapped (once or twice) in any direction: the TAP_X_EN, TAP_Y_EN and TAP_Z_EN bits of the TAP_CFG register must be set to 1 to enable the tap recognition on X, Y, Z directions, respectively.

Configurable parameters for tap recognition functionality are the tap threshold and the Shock, Quiet and Duration time windows.

The TAP_THS[4:0] bits of the TAP_THS_6D register are used to select the unsigned threshold value used to detect the tap event. The value of 1 LSB of these 5 bits depends on the selected accelerometer full scale: 1 LSB = (FS_XL)/(2^5). The unsigned threshold is applied to both positive and negative slope data.

The Shock time window defines the maximum duration of the overthreshold event: the acceleration must return below the threshold before the Shock window has expired, otherwise the tap event is not detected. The SHOCK[1:0] bits of the INT_DUR2 register are used to set the Shock time window value: the default value of these bits is 00b and corresponds to 4*ODR_XL time, where ODR_XL is the accelerometer output data rate. If the SHOCK[1:0] bits are set to a different value, 1 LSB corresponds to 8*ODR_XL time.

In the double-tap case, the Quiet time window defines the time after the first tap recognition in which there must not be any overthreshold. When the latch mode is disabled (LIR bit of TAP_CFG is set to 0), the Quiet time also defines the length of the interrupt pulse (in both single and double-tap case). The QUIET[1:0] bits of the INT_DUR2 register are used to set the Quiet time window value: the default value of these bits is 00b and corresponds to

天猫店铺网址:https://telesky.tmall.com/

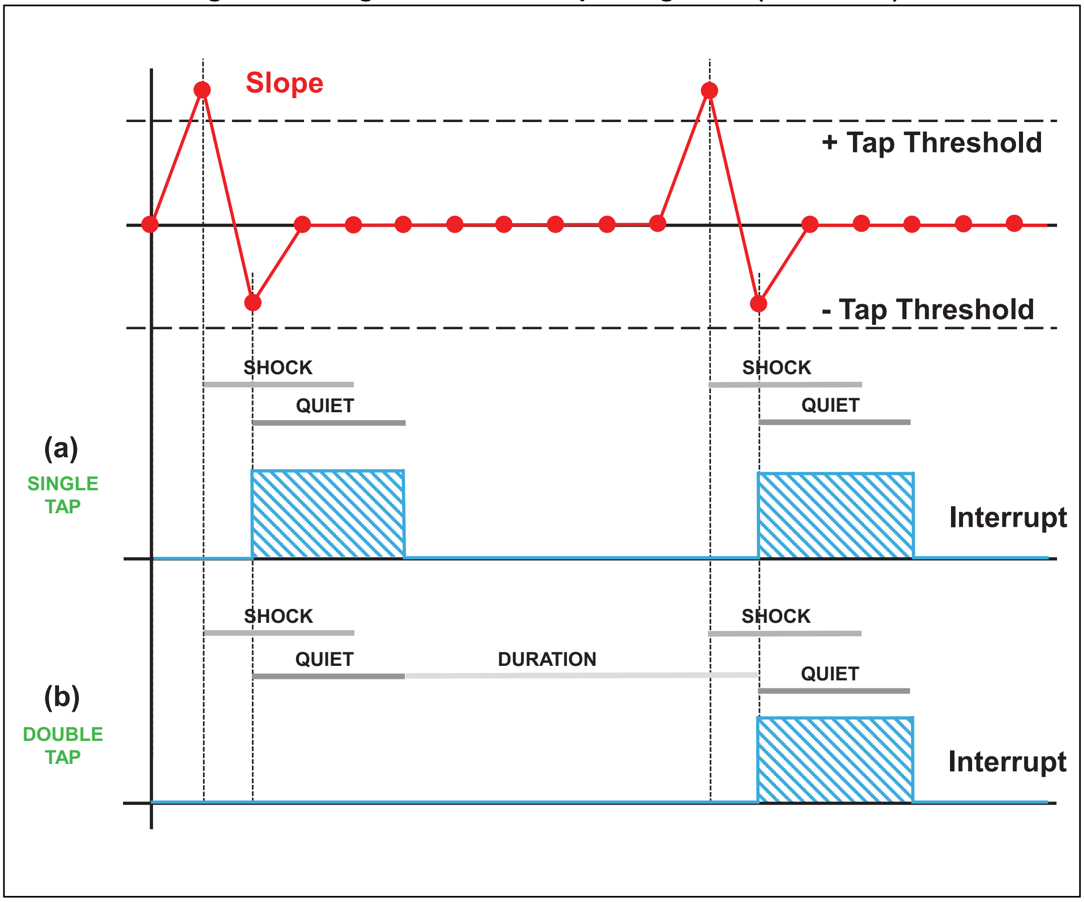
2*ODR_XL time, where ODR_XL is the accelerometer output data rate. If the QUIET[1:0] bits are set to a different value, 1 LSB corresponds to 4*ODR_XL time.

In the double-tap case, the Duration time window defines the maximum time between two consecutive detected taps. The Duration time period starts just after the completion of the Quiet time of the first tap. The DUR[3:0] bits of the INT_DUR2 register are used to set the Duration time window value: the default value of these bits is 0000b and corresponds to 16*ODR_XL time, where ODR_XL is the accelerometer output data rate. If the DUR[3:0] bits are set to a different value, 1 LSB corresponds to 32*ODR_XL time.

Figure 20 illustrates a single-tap event (a) and a double-tap event (b). These interrupt signals can be driven to the two interrupt pins by setting to 1 the INT1_SINGLE_TAP bit of the MD1_CFG register or the INT2_SINGLE_TAP bit of the MD2_CFG register for the single-tap case, and setting to 1 the INT1_DOUBLE_TAP bit of the MD1_CFG register or the INT2_DOUBLE_TAP bit of the MD2_CFG register for the double-tap case.

No single/double tap interrupt is generated if the accelerometer is in Inactivity status (see *Section 5.6* for more details).

Figure 20. Single and double-tap recognition (LIR bit = 0)



Tap interrupt signals can also be checked by reading the TAP_SRC (1Ch) register, described in *Table 33*.

Table 33. TAP_SRC register

b7	b6	b5	b4	b3	b2	b1	b0
0	TAP_IA	SINGLE _TAP	DOUBLE _TAP	TAP_SIGN	X_TAP	Y_TAP	Z_TAP

- TAP_IA is set high when a single-tap or double-tap event has been detected.
- SINGLE_TAP is set high when a single tap has been detected.
- DOUBLE_TAP is set high when a double tap has been detected.
- TAP_SIGN indicates the acceleration sign when the tap event is detected. It is set low in case of positive sign and it is set high in case of negative sign.
- X_TAP (Y_TAP, Z_TAP) is set high when the tap event has been detected on the X (Y, Z) axis.

Single and double-tap recognition works independently. Setting the SINGLE_DOUBLE_TAP bit of WAKE_UP_THS to 0, only the single-tap recognition is enabled: double-tap recognition is disabled and cannot be detected. When the SINGLE_DOUBLE_TAP is set to 1, both single and double-tap recognition are enabled.

If the latch mode is enabled and the interrupt signal is driven to the interrupt pins, the value assigned to SINGLE_DOUBLE_TAP also affects the behavior of the interrupt signal: when it is set to 0, the latch mode is applied to the single-tap interrupt signal; when it is set to 1, the latch mode is applied to the double-tap interrupt signal only. The latched interrupt signal is kept high until the TAP_SRC register is read. If the latch mode is enabled but the interrupt signal is not driven to the interrupt pins, the latch feature does not take effect.

5.5.4 Single-tap example

A basic SW routine for single-tap detection is given below.

1	Write 60h into CTRL1_XL	// Turn on the accelerometer // ODR_XL = 416 Hz, FS_XL = 2g
2	Write 0Eh into TAP_CFG	// Enable tap detection on X, Y, Z axis
3	Write 09h into TAP_THS_6D	// Set tap threshold
4	Write 06h into INT_DUR2	// Set Quiet and Shock time windows
5	Write 00h into WAKE_UP_THS	// Only single tap enabled (SINGLE_DOUBLE_TAP = 0)
6	Write 40h into MD1_CFG	// Single tap interrupt driven to INT1 pin

In this example the TAP_THS field of the TAP_THS_6D register is set to 01001b, therefore the tap threshold is 562.5 mg (= 9 * FS_XL / 2^5).

The SHOCK field of the INT_DUR2 register is set to 10b: an interrupt is generated when the slope data exceeds the programmed threshold, and returns below it within 38.5 ms (= 2 * 8 / ODR_XL) corresponding to the Shock time window.

The QUIET field of the INT_DUR2 register is set to 01b: since the latch mode is disabled, the interrupt is kept high for the duration of the Quiet window, therefore 9.6 ms (= 1 * 8 / ODR_XL.)

5.5.5 Double-tap example

A basic SW routine for double-tap detection is given below.

1	Write 60h into CTRL1_XL	// Turn on the accelerometer // ODR_XL = 416 Hz, FS_XL = 2g
2	Write 0Eh into TAP_CFG	// Enable tap detection on X, Y, Z axis
3	Write 0Ch into TAP_THS_6D	// Set tap threshold
4	Write 7Fh into INT_DUR2	// Set Duration, Quiet and Shock time windows
5	Write 80h into WAKE_UP_THS	// Single & Double tap enabled (SINGLE_DOUBLE_TAP = 1)
6	Write 08h into MD1_CFG	// Double tap interrupt driven to INT1 pin

In this example the TAP_THS field of the TAP_THS_6D register is set to 01100b, therefore the tap threshold is 750 mg (= 12 * FS_XL / 2^5).

For interrupt generation, during the first and the second tap the slope data must return below the threshold before the Shock window has expired. The SHOCK field of the INT_DUR2 register is set to 11b, therefore the Shock time is 57.7 ms (= 3 * 8 / ODR_XL).

For interrupt generation, after the first tap recognition there must not be any slope data overthreshold during the Quiet time window. Furthermore, since the latch mode is disabled, the interrupt is kept high for the duration of the Quiet window. The QUIET field of the INT_DUR2 register is set to 11b, therefore the Quiet time is 28.8 ms (= 3 * 4/ ODR_XL).

For the maximum time between two consecutive detected taps, the DUR field of the INT_DUR2 register is set to 0111b, therefore the Duration time is 538.5 ms (= 7 * 32 / ODR_XL).

5.6 Activity/Inactivity recognition

The Activity/Inactivity recognition function allows reducing system power consumption and developing new smart applications.

When the Activity/Inactivity recognition function is activated, the LSM6DS3 device is able to automatically decrease the accelerometer sampling rate to 12.5 Hz, increasing the accelerometer ODR and bandwidth as soon as the wake-up interrupt event has been detected. This feature is applied to the accelerometer sensor only, regardless of the selected gyroscope power mode and ODR.

With this feature the system may be efficiently switched from low-power consumption to full performance and vice-versa depending on user-selectable acceleration events, thus ensuring power saving and flexibility.

The Activity/Inactivity recognition function is enabled by setting to 1 the INACTIVITY bit of the WAKE_UP_THS register.

In the LSM6DS3 device the Activity/Inactivity recognition function can be implemented using either the slope filter (see *Section 3.7.1* for more details) or the high-pass digital filter, as illustrated in *Figure 5*. The filter to be applied can be selected using the SLOPE_FDS bit of the TAP_CFG register: if this bit is set to 0 (default value), the slope filter is used; if it's set to 1, the high-pass digital filter is used.

This function can be fully programmed by the user in terms of expected amplitude and timing of the slope data by means of a dedicated set of registers (*Figure 21*).

The unsigned threshold value is defined using the WK_THS[5:0] bits of dedicated set of registers WAKE_UP_THS register; the value of 1 LSB of these 6 bits depends on the selected accelerometer full scale: 1 LSB = (FS_XL)/(2^6). The threshold is applied to both positive and negative slope data.

When a certain number of consecutive X,Y,Z slope data is smaller than the configured threshold, the ODR_XL [3:0] bits of the CTRL1_XL register are bypassed (Inactivity) and the accelerometer is internally set to 12.5 Hz although the content of CTRL1_XL is left untouched. The duration of the Inactivity status to be recognized is defined by the SLEEP_DUR[3:0] bits of the WAKE_UP_DUR register: 1 LSB corresponds to 512*ODR_XL time, where ODR_XL is the accelerometer output data rate.

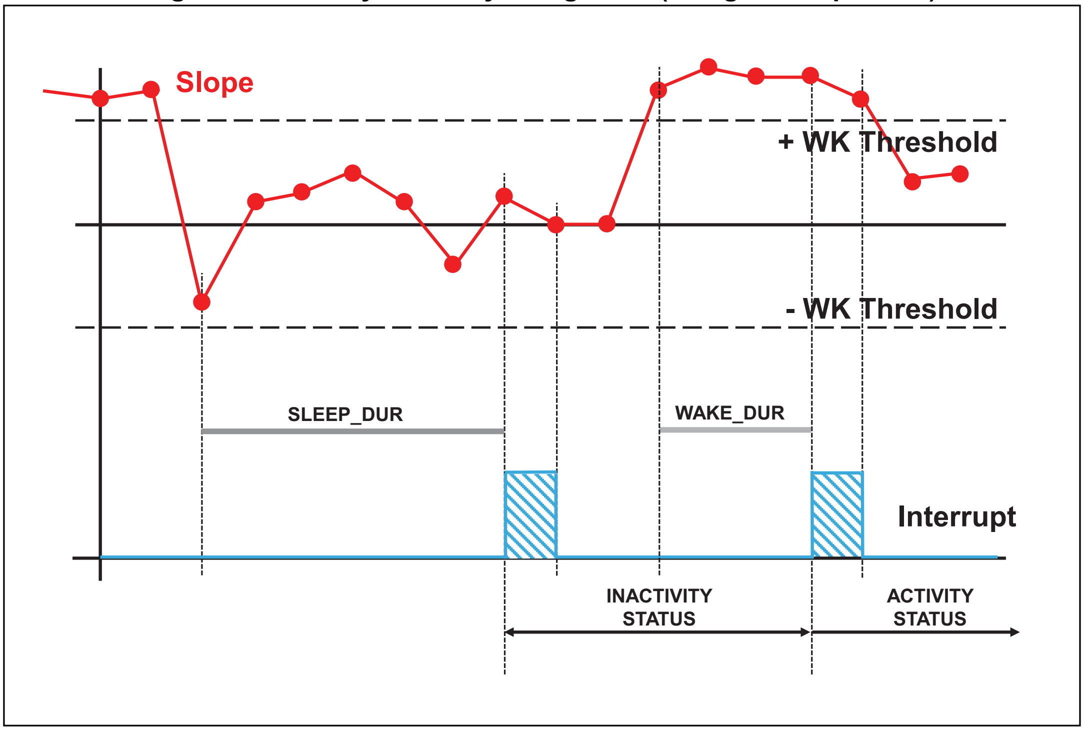
When the Inactivity status is detected, the interrupt is set high for 1/ODR_XL[s] period then it is automatically deasserted.

When a certain number of consecutive slope data on one axis become bigger than the threshold, the CTRL1_XL register settings are immediately restored (Activity). The duration of the Activity status to be recognized is defined by the WAKE_DUR[1:0] bits of the WAKE_UP_DUR register: 1 LSB corresponds to 1*ODR_XL time, where ODR_XL is the accelerometer output data rate.

When the Activity status is detected, the interrupt is set high for 1/ODR_XL[s] period then it is automatically deasserted.

Once the Activity/Inactivity detection function is enabled, the status can be driven to the two interrupt pins by setting to 1 the INT1_INACT_STATE bit of the MD1_CFG register or the INT1_INACT_STATE bit of the MD2_CFG register; it can also be checked by reading the SLEEP_STATE_IA bit of the WAKE_UP_SRC register.

Figure 21. Activity/Inactivity recognition (using the slope filter)



The code provided below is a basic routine for Activity/Inactivity detection implementation.

1	Write 50h into CTRL1_XL	// Turn on the accelerometer // ODR_XL = 208 Hz, FS_XL = 2*g*
2	Write 42h into WAKE_UP_DUR	// Set duration for Inactivity detection // Set duration for Activity detection
3	Write 42h into WAKE_UP_THS	// Set Activity/Inactivity threshold // Enable Activity/Inactivity detection
4	Write 80h into MD1_CFG	// Activity/Inactivity interrupt driven to INT1 pin

In this example the WK_THS field of the WAKE_UP_THS register is set to 000010b, therefore the Activity/Inactivity threshold is 62.5 mg (= 2 * FS_XL / 2^6).

Before Inactivity detection, the X,Y,Z slope data must be smaller than the configured threshold for a period of time defined by the SLEEP_DUR field of the WAKE_UP_DUR register: this field is set to 0010b, corresponding to 4.92 s (= 2 * 512 / ODR_XL). After this period of time has elapsed, the accelerometer ODR is internally set to 12.5 Hz.

The Activity status is detected and the CTRL1_XL register settings immediately restored if the slope data of (at least) one axis are bigger than the threshold for a period of time defined by the WAKE_DUR field of the WAKE_UP_DUR register: this field is set to 10b, corresponding to 9.62 ms (= 2 * 1 / ODR_XL).

5.7 Boot status

After the device is powered up, the LSM6DS3 performs a 20 ms boot procedure to load the trimming parameters. After the boot is completed, both the accelerometer and the gyroscope are automatically configured in Power-Down mode. During the boot time the registers are not accessible.

After power up, the trimming parameters can be re-loaded by setting to 1 the BOOT bit of the CTRL3_C register.

No toggle of the device power lines is required and the content of the device control registers is not modified, so the device operating mode doesn’t change after boot. If the reset to the default value of the control registers is required, it can be performed by setting to 1 the SW_RESET bit of the CTRL3_C register. The SW_RESET procedure can take 50 µs; the status of reset is signaled by the status of SW_RESET bit of the CTRL3_C register: once the reset is completed, this bit is automatically set low.

The boot status signal is driven to the INT1 interrupt pin by setting to 1 the INT1_BOOT bit of the INT1_CTRL register: this signal is set high while the boot is running and it is set low again at the end of the boot procedure.

Reboot flow is as follows:

1. Set the Gyroscope in Power-Down mode;
2. Set the Accelerometer in High-Performance mode;
3. Set to 1 the BOOT bit of the CTRL3_C register;
4. Wait 20 ms.

Reset flow is as follows:

1. Set the Gyroscope in Power-Down mode;
2. Set the Accelerometer in High-Performance mode;
3. Set to 1 the SW_RESET bit of the CTRL3_C register;
4. Wait 50 µs (or wait until the SW_RESET bit of the CTRL3_C register returns to 0).

In order to avoid conflicts, the reboot and the sw reset must not be executed at the same time (do not set to 1 at the same time both the BOOT bit and SW_RESET bit of CTRL3_C register). The above flows must be performed serially.

天猫店铺网址:https://telesky.tmall.com/

6 Android embedded functions

The LSM6DS3 device implements in hardware the sensor-related functions specified in Android L; specific IP blocks with negligible power consumption and high-level performance implement the following functions using only the accelerometer:

- Pedometer functions (step detector and step counter);
- Significant motion;
- Tilt;
- Timestamp.

All these functions work at 26 Hz, so the accelerometer ODR must be set at 26 Hz or higher values.

6.1 Pedometer functions: step detector and step counter

A specific IP block of the LSM6DS3 device is dedicated to pedometer functions: the step detector and the step counter.

Pedometer functions work at 26 Hz, so the accelerometer ODR must be set at 26 Hz or higher values.

In order to enable the pedometer functions it is necessary to set to 1 both the FUNC_EN bit of the CTRL10_C register and the PEDO_EN bit of the TAP_CFG register.

The step counter indicates the number of steps detected by the step detector algorithm after the pedometer function has been enabled. The step count is given by the concatenation of the STEP_COUNTER_H and STEP_COUNTER_L registers and it is represented as a 16-bit unsigned number.

The step count is not reset to zero when the accelerometer is configured in Power-Down or the pedometer is disabled; it can be reset to zero by setting the PEDO_RST_STEP bit of the CTRL10_C register to 1. After the counter resets, the PEDO_RST_STEP bit is not automatically set back to 0.

The step detector functionality generates an interrupt every time a step is recognized. In case of interspersed step sessions, 7 consecutive steps (debounce steps) have to be detected before the first interrupt generation in order to avoid false step detections (debounce functionality).

The number of debounce steps can be modified through the DEB_STEP field of the PEDO_DEB_REG register: basically, it corresponds to the minimum number of steps to be detected before the first step counter increment. 1 LSB of this field corresponds to 1 step, the default value is 6 steps.

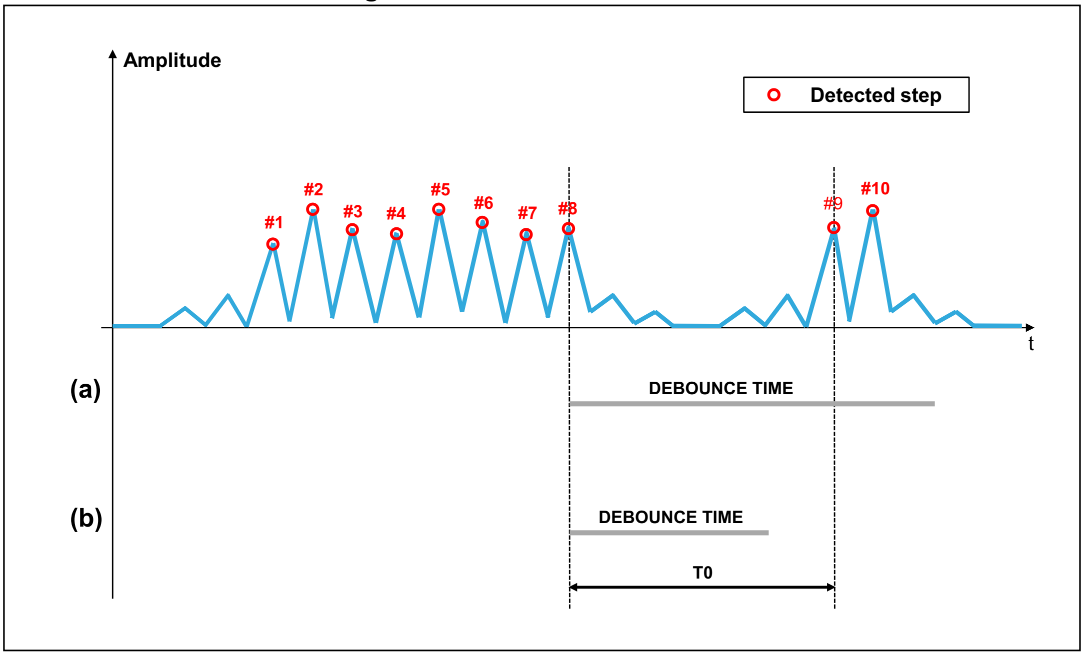
The debounce functionality restarts after around 1 second of device inactivity. This period of time (debouncing time) can be modified through the DEB_TIME field of the PEDO_DEB_REG register. 1LSB corresponds to 80 ms, the default value is 13 (13 *80ms = 1040 ms). This value must be greater than 0.

The example in *Figure 22* explains how the step counter behavior changes by changing the debounce time. In this example, the pedometer algorithm detects 7 steps close to each other and then two more isolated steps after a certain period of time; assuming that the value of DEB_STEP field of the PEDO_DEB_REG register is set to 6 LSB (= 6 debounce steps, default value) and the initial step counter value in STEP_COUNTER_H/L registers is zero (no steps previously detected):

a) in case (a), the step count starts increasing after the seventh step and after the first eight detected steps, the value of STEP_COUNTER_H/L registers will be 8. Since the debounce time set in the DEB_TIME field of the PEDO_DEB_REG register is greater than the period of time between the step #8 and the step #9, also the steps #9 and #10 will cause the step counter to increase: the final step count value in STEP_COUNTER_H/L registers will be 10.

b) also in case (b) the step count starts increasing after the seventh step and after the first eight detected steps, the value of STEP_COUNTER_H/L registers will be 8, but since the debounce time set in the DEB_TIME field of the PEDO_DEB_REG register is lower than the period of time between the step #8 and the step #9, the steps #9 and #10 will not cause the step counter to increase: the final step count value in STEP_COUNTER_H/L registers will be 8. Furthermore, if between the step #10 and the following step elapses a period of time greater than the debounce time, the detected steps #9 and #10 will be definitively discarded and no longer considered.

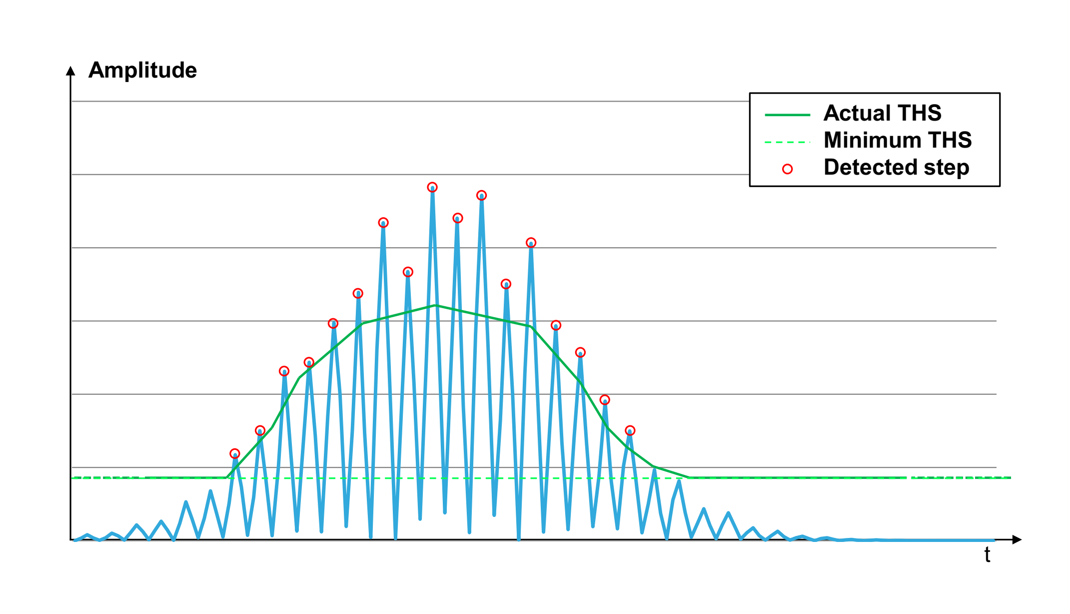
Figure 22. Pedometer debounce



By default, the step counter works at 2*g* full scale, independently of the configured device full scale, but it can be configured to work at 4*g* full scale which can help to avoid acceleration saturation (e.g. in fast walk). In order to set the 4*g* full scale for the step counter, the PEDO_4G bit of the PEDO_THS_REG register has to be set to 1 and the accelerometer full scale configured in CTRL1_XL register must be ≥ 4*g*.

It is also possible to set the “Minimum Threshold”, that is the value at which the threshold for step recognition asymptotically tends if no steps are detected and below which it cannot descend (see *Figure 23*). This configuration is available in the THS_MIN field of the PEDO_THS_REG register. The value of 1 LSB of these 6 bits depends on the selected step counter full scale: 1 LSB = 16 m*g* if the PEDO_4G bit is 0; 1 LSB = 32 m*g* if the PEDO_4G bit is 1.

Figure 23. Pedometer minimum threshold



The step detector interrupt signal can be driven to the INT1 interrupt pin by setting to 1 the INT1_STEP_DETECTOR bit of the INT1_CTRL register; it can also be checked by reading the STEP_DETECTED bit of the FUNC_SRC register.

Instead of generating an interrupt every time a step is recognized, it is possible to generate it if at least one step is detected within a certain time period. This time period is defined by setting a value higher than 00h in the STEP_COUNT_DELTA register. It is necessary to set the TIMER_EN bit of the TAP_CFG register to 1 (to enable the timer) and the TIMER_HR bit of the WAKE_UP_DUR register to 0 when using this feature: in this case, 1 LSB of the value of the STEP_COUNT_DELTA register corresponds to 1.6384 seconds. This interrupt signal can be driven to the INT2 interrupt pin by setting to 1 the INT2_STEP_DELTA bit of the INT2_CTRL register; it can also be checked by reading the STEP_COUNT_DELTA_IA bit of the FUNC_SRC register.

The Step Counter overflow signal can be driven to the INT2 interrupt pin by setting to 1 the INT2_STEP_COUNT_OV bit of the INT2_CTRL register: in this case, when the step count reaches the 2^{16} value, an interrupt signal is generated on the INT2 pin and the step count is automatically reset to zero, no need to reset it by setting to 1 the PEDO_RST_STEP bit.

If latch mode is disabled (LIR bit of TAP_CFG is set to 0), the interrupt signal generated by the pedometer functions is pulsed: the duration of the pulse observed on the interrupt pins is about 60 µs; the duration of the pulse observed on the bits STEP_COUNT_DELTA_IA, STEP_DETECTED_IA and STEP_OVERFLOW of the FUNC_SRC register is 1/26 Hz.

If latch mode is enabled (LIR bit of TAP_CFG is set to 1) and the interrupt signal is driven to the interrupt pins, once a step has occurred, a reading of the FUNC_SRC register clears the request on both the pins and the STEP_COUNT_DELTA_IA, STEP_DETECTED_IA and STEP_OVERFLOW bits of the FUNC_SRC register, and the device is ready to recognize the next step. If latch mode is enabled but the interrupt signal is not driven to the interrupt pins, the interrupt signal observed on the bits of the FUNC_SRC register is pulsed, with a fixed duration of 1/26 Hz.

Step counter timestamp information is available in the STEP_TIMESTAMP_H and STEP_TIMESTAMP_L registers: when a step is detected, the value of the

TIMESTAMP_REG2 register is copied in STEP_TIMESTAMP_H, and the value of the TIMESTAMP_REG1 register is copied in STEP_TIMESTAMP_L, providing the timestamp information of this step. For more details about LSM6DS3 timestamp counter and TIMESTAMP_REG2/TIMESTAMP_REG1, see *Section 6.4*.

The step counter timestamp resolution depends on the value of the TIMER_HR bit of the WAKE_UP_DUR register: when this bit is set to 0, 1 LSB of the time step count corresponds to 1638.4 ms; when this bit is set to 1, 1 LSB of the time step count corresponds to 6.4 ms.

Step counter data can be stored in FIFO as a fourth data set along with timestamp data (see *Section 8.8* for more details).

Hereafter a basic SW routine which shows how to enable the pedometer functions:

1	Write 20h into CTRL1_XL	// Turn on the accelerometer // ODR_XL = 26 Hz, FS_XL = 2g
2	Write 3Ch into CTRL10_C	// Enable embedded functions
3	Write 40h into TAP_CFG	// Enable pedometer algorithm
4	Write 80h into INT1_CTRL	// Step Detector interrupt driven to INT1 pin

The interrupt signal is generated when a step is recognized and the step count is available by reading the STEP_COUNTER_H / STEP_COUNTER_L registers.

6.2 Significant motion

The Significant Motion function generates an interrupt when a ‘significant motion’, that could be due to a change in user location, is detected: in the LSM6DS3 device this function has been implemented in hardware using only the accelerometer.

Significant Motion functionality can be used in location-based applications in order to receive a notification indicating when the user is changing location.

The Significant Motion function works at 26 Hz, so the accelerometer ODR must be set at a value of 26 Hz or higher.

In order to enable Significant Motion detection it is necessary to set to 1 both the FUNC_EN bit and the SIGN_MOTION_EN bit of the CTRL10_C register.

Note: In order to ensure that the Significant Motion function works properly, the step detector/step counter has to be on.

The Significant Motion interrupt signal is driven to the INT1 interrupt pin by setting to 1 both the INT1_SIGN_MOTION bit and the INT1_STEP_DETECTOR bit of the INT1_CTRL register; it can also be checked by reading the SIGN_MOTION_IA bit of the FUNC_SRC register.

If latch mode is disabled (LIR bit of TAP_CFG is set to 0), the interrupt signal generated by the Significant Motion function is pulsed: the duration of the pulse observed on the interrupt pins is about 60 µs; the duration of the pulse observed on the SIGN_MOTION_IA bit of the FUNC_SRC register is 1/26 Hz.

If latch mode is enabled (LIR bit of TAP_CFG is set to 1) and the interrupt signal is driven to the interrupt pins, once a ‘significant motion’ is detected, a reading of the FUNC_SRC register clears the request on both the pins and the SIGN_MOTION_IA bit of the

FUNC_SRC register, and the device is ready to recognize the next event. If latch mode is enabled but the interrupt signal is not driven to the interrupt pins, the interrupt signal observed on the SIGN_MOTION_IA bit of the FUNC_SRC register is pulsed, with a fixed duration of 1/26 Hz.

The embedded function register (accessible by setting to 1 the FUNC_CFG_EN bit of FUNC_CFG_ACCESS) used to configure the Significant Motion threshold parameter is the SM_THS register. The SM_THS_[7:0] bits of this register define the threshold value: it corresponds to the number of steps to be performed by the user upon a change of location before the Significant Motion interrupt is generated. It is expressed as an 8-bit unsigned value: the default value of this field is equal to 6 (= 00000110b).

When the debounce functionality of the pedometer is active (see *Section 6.1* for details), the Significant Motion threshold is effective only if its value, corresponding to the value of the SM_THS_[7:0] bits of the SM_THS register, is equal to or greater than the pedometer debounce threshold (corresponding to the value of the DEB_STEP[2:0] bits of the PEDO_DEB_REG (2Bh) register).

Basically, three different scenarios are possible for the Significant Motion threshold value:

a) If the pedometer debounce functionality is not active, the Significant Motion threshold value is defined by the SM_THS_[7:0] bits;

b) if the pedometer debounce functionality is active and the Significant Motion threshold value is equal to or greater than the pedometer debounce value, the effective Significant Motion threshold value is defined by the SM_THS_[7:0] bits;

c) if the pedometer debounce functionality is active and the Significant Motion threshold value is lower than the pedometer debounce value, the effective Significant Motion threshold value is defined by the DEB_STEP[2:0] bits.

Note: In case c), if the desired Significant Motion threshold is lower than the default value, the value of the DEB_STEP[2:0] bits of the PEDO_DEB_REG (2Bh) register has to be decreased accordingly. Note that an excessive reduction of the pedometer debounce threshold can cause the Pedometer to report false step detections!

Hereafter a basic SW routine which shows how to enable the significant motion detection function:

1	Write 80h into FUNC_CFG_ADDRESS	// Enable access to embedded functions registers
2	Write 08h into SM_THS	// Set Significant Motion threshold
3	Write 00h into FUNC_CFG_ADDRESS	// Disable access to embedded functions registers
4	Write 20h into CTRL1_XL	// Turn on the accelerometer // ODR_XL = 26 Hz, FS_XL = 2g
5	Write 00h into TAP_CFG	// Disable pedometer
6	Write 3Dh into CTRL10_C	// Enable embedded functions // Enable Significant Motion detection
7	Write 40h into TAP_CFG	// Enable pedometer algorithm
8	Write C0h into INT1_CTRL	// Significant motion interrupt driven to INT1 pin

In this example the SM_THS_[7:0] bits of the SM_THS register are set to 00001000b, therefore the Significant Motion threshold is equal to 8.

6.3 Tilt

The Tilt function allows detecting when an activity change occurs (e.g. when phone is in a front pocket and the user goes from sitting to standing or standing to sitting): in the LSM6DS3 device it has been implemented in hardware using only the accelerometer.

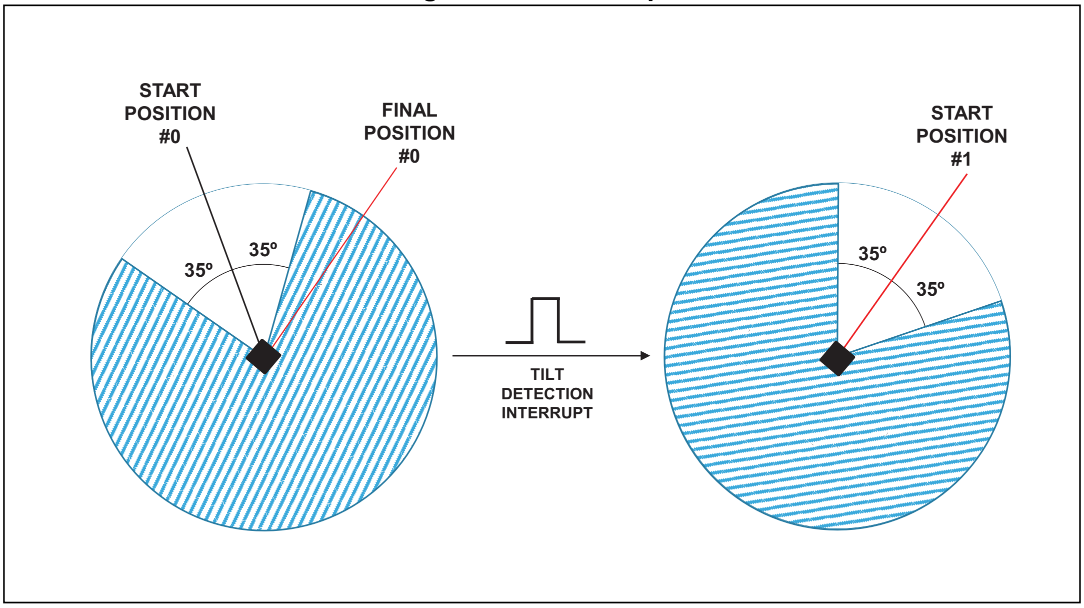
In order to enable the tilt detector it is necessary to set to 1 both the FUNC_EN bit of the CTRL10_C register and the TILT_EN bit of the TAP_CFG register.

If the device is configured for tilt event detection, an interrupt is generated when the device is tilted by an angle greater than 35 degrees from the start position. The start position is defined as the position of the device when the tilt detection is enabled or the position of the device when the last Tilt interrupt was generated.

After this function is enabled, for the generation of the first Tilt interrupt the device should be continuously tilted by an angle greater than 35 degrees since start position for a period of time of 2 seconds. After the first Tilt interrupt is generated, the Tilt interrupt signal is set high as soon as the device is tilted by an angle greater than 35 degrees from the position of the device corresponding to the last interrupt detection (no need to wait 2 seconds).

In the example shown in *Figure 24* tilt detection is enabled when the device orientation corresponds to “start position #0”: the first interrupt is generated if the device is rotated by an angle greater than 35 degrees from the start position and remains in the blue zone for a period of time of at least 2 seconds. After the first tilt detection interrupt is generated, the new start position (#1) corresponds to the position of the device when the previous interrupt was generated (final position #0), and the next interrupt signal will be generated as soon as the device is tilted by an angle greater than 35 degrees, entering the blue zone surrounding the start position #1.

Figure 24. Tilt example



This interrupt signal can be driven to the two interrupt pins by setting to 1 the INT1_TILT bit of the MD1_CFG register or the INT2_TILT bit of the MD2_CFG register; it can also be checked by reading the TILT_IA bit of the FUNC_SRC register.

If latch mode is disabled (LIR bit of TAP_CFG is set to 0), the interrupt signal generated by the Tilt function is pulsed: the duration of the pulse observed on the interrupt pins is about 60 µs; the duration of the pulse observed on the TILT_IA bit of FUNC_SRC register is 1/26 Hz.

If latch mode is enabled (LIR bit of TAP_CFG is set to 1) and the interrupt signal is driven to the interrupt pins, once a tilt is detected, a reading of the FUNC_SRC register clears the request on both the pins and the TILT_IA bit of FUNC_SRC register, and the device is ready to recognize the next tilt event. If latch mode is enabled but the interrupt signal is not driven to the interrupt pins, the interrupt signal observed on the TILT_IA bit of the FUNC_SRC register is pulsed, with a fixed duration of 1/26 Hz.

The tilt function works at 26 Hz, so the accelerometer ODR must be set at 26 Hz or higher values.

Hereafter a basic SW routine which shows how to enable the tilt detection function:

1	Write 20h into CTRL1_XL	// Turn on the accelerometer // ODR_XL = 26 Hz, FS_XL = 2g
2	Write 3Ch into CTRL10_C	// Enable embedded functions
3	Write 20h into TAP_CFG	// Enable tilt detection
4	Write 02h into MD1_CFG	// Tilt detector interrupt driven to INT1 pin

6.4 Timestamp

Together with sensor data the LSM6DS3 device can provide timestamp information.

If both the accelerometer and the gyroscope are in Power-Down mode, the timestamp counter doesn’t work.

To enable this functionality the TIMER_EN bit of the TAP_CFG register has to be set to 1: the time step count is given by the concatenation of the TIMESTAMP_REG2 & TIMESTAMP_REG1 & TIMESTAMP_REG0 registers and is represented as a 24-bit unsigned number.

The timestamp resolution can be configured using the TIMER_HR bit of the WAKE_UP_DUR register: when this bit is set to 0, 1 LSB of time step count corresponds to 6.4 ms (low-resolution mode); when this bit is set to 1, 1 LSB of time step count corresponds to 25 µs (high-resolution mode).

When the maximum value 16777215 LSB (corresponding to FFFFFFh) is reached and low resolution (TIMER_HR = 0) is used, the counter is automatically reset to 000000h and continues to count. When the maximum value is reached and high resolution (TIMER_HR = 1) is used, the counter is not automatically reset to 0 and freezes at FFFFFFh. In any case, the timer count can be reset to zero at any time by writing the reset value AAh in the TIMESTAMP_REG2 register.

An interrupt is generated around 1.638 seconds before timer saturation in both high-resolution mode (when the timer step count reaches the value FF0000h) and low-resolution mode (when the timer step count reaches the value FFFF00h). This interrupt signal can be driven to the INT1 pin by setting to 1 the INT1_TIMER bit of the MD1_CFG register. Once the interrupt pin is asserted, it must be reset to zero by writing AAh in the TIMESTAMP_REG2 register (the timer step count will be reset also).

The timestamp count can be stored in FIFO as a fourth data set along with the step counter data (see *Section 8.8* for details).

The timestamp resolution has to be set before enabling the timestamp functionality; a basic SW routine is as follows:

1	Write 50h into CTRL1_XL	// Turn on the accelerometer // ODR_XL = 208 Hz, FS_XL = 2g
2	Write 10h into WAKE_UP_DUR	// Timestamp resolution = 25µs
3	Write 80h into TAP_CFG	// Enable timestamp count
4	Write 01h into MD1_CFG	// End counter interrupt driven to INT1 pin

When switching from a low timestamp resolution to a high resolution the timer count must be reset, as indicated in the example below:

1	Write 50h into CTRL1_XL	// Turn on the accelerometer // ODR_XL = 208 Hz, FS_XL = 2g
2	Write 00h into WAKE_UP_DUR	// Timestamp resolution = 6.4ms
3	Write 80h into TAP_CFG	// Enable timestamp count
	...	
N	Write 10h into WAKE_UP_DUR	// Timestamp resolution = 25µs
N+1	Write AAh into TIMESTAMP_REG2	// Reset timer counter

7 Mode 2 - sensor hub mode

The hardware flexibility of the LSM6DS3 allows connecting the pins to external sensors to expand functionalities such as adding a sensor hub.

Two different mode connections are available:

- **Mode 1 - Slave-only mode**: I^2C/SPI (3- and 4-wire) slave interface is available;
- **Mode 2 - Sensor Hub mode**: I^2C/SPI (3- and 4-wire) slave interface and I^2C master interface for the connection of external sensors are available.

The pin connections and function for each mode are described in the LSM6DS3 datasheet; Mode 2 connection mode is described in detail in the following paragraphs.

7.1 Sensor hub mode description

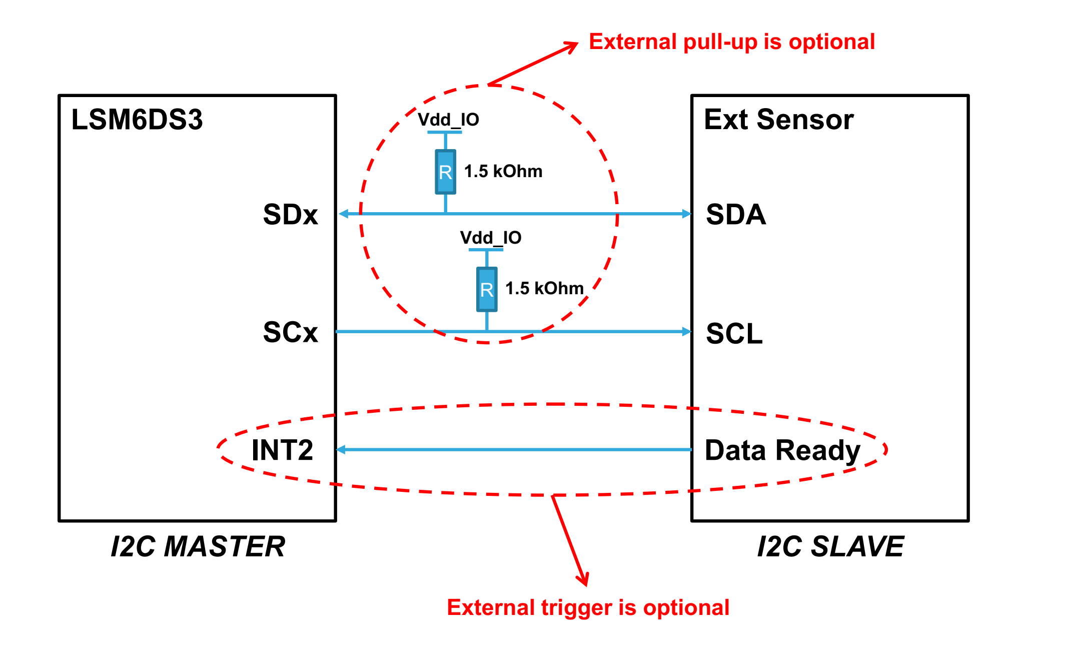
In sensor hub mode (Mode 2) up to 4 external sensors can be connected to the I^2C master interface of LSM6DS3 device.

External sensor data can also be stored in FIFO with a configurable decimation factor (see *Section 8: First-in first-out (FIFO) buffer* for details).

If both the accelerometer and the gyroscope are in Power-Down mode, the sensor hub doesn't work.

All external sensors have to be connected in parallel to the SDx/SCx pins of the device, as illustrated in *Figure 25* for a single external sensor. External pull-up resistors and the external trigger signal connection are optional and depend on the configuration of the registers.

Figure 25. External sensor connections in Mode 2



7.2 Sensor hub mode registers

In order to enable the embedded functionalities of registers LSM6DS3, the FUNC_EN bit of the CTRL10_C register has to be set to 1; after enabling the sensor hub functionality, the MASTER_CONFIG register has to be used for the configuration of the I^{2}C master interface.

A set of registers SLVx_ADD, SLVx_SUBADD, SLAVEx_CONFIG is dedicated to the configuration of the 4 slave interfaces associated to the 4 connectable external sensors. An additional register, DATAWRITE_SRC_MODE_SUB_SLV0, is associated to slave #0 only: it can be used to implement the writing and the source mode conditioned reading of the registers of the external sensor associated to slave #0.

Finally, 18 registers (from SENSORHUB1_REG to SENSORHUB18_REG) are available to store the data read from the external sensors.

7.2.1 CTRL10_C (19h)

Table 34. CTRL10_C register

b7	b6	b5	b4	b3	b2	b1	b0
0	0	X	X	X	FUNC_EN	X	X

- FUNC_EN must be set to 1 in order to enable the embedded functionalities of the LSM6DS3 (pedometer, tilt, significant motion, ironing).

7.2.2 MASTER_CONFIG (1Ah)

This register is used to configure the I^{2}C master interface.

Table 35. MASTER_CONFIG register

b7	b6	b5	b4	b3	b2	b1	b0
DRDY_ON_INT1	X	0	START_CONFIG	PULL_UP_EN	PASS_THROUGH_MODE	X	MASTER_ON

- DRDY_ON_INT1 bit has to be set to 1 to drive on the INT1 pin the I^{2}C master Data-Ready signal (corresponding to the behavior of the SENSORHUB_END_OP bit of FUNC_SRC register). The I^{2}C master Data-Ready signal indicates when the sensor hub routine is complete and the external sensor data are available to be read on the SENSORHUBx_REG registers (depending on the configuration of the SLVx_ADD, SLVx_SUBADD, SLAVEx_CONFIG registers).
- START_CONFIG bit selects the sensor hub trigger signal.
  - When this bit is set to 0, the accelerometer sensor has to be active (not in Power Down mode) and the sensor hub trigger signal is the accelerometer data-ready signal, with a frequency corresponding to the accelerometer ODR up to 100 Hz.
  - When this bit is set to 1, at least one sensor between the accelerometer and the gyroscope has to be active and the sensor hub trigger signal is the INT2 pin; in fact, when both the MASTER_ON bit and START_CONFIG bit are set to 1, the INT2 pin is configured as an input signal. In this case, the INT2 pin has to be connected to the Data-Ready pin of the external sensor (*Figure 25*) in order to trigger the reading/writing operations on the external sensor registers.

Note: In case of external trigger signal usage (START_CONFIG=1), if the INT2 pin is connected to the Data-Ready pin of the external sensor (Figure 25) and the latter is in Power-Down mode, then no Data-Ready signal can be generated by the external sensor. For this reason, the initial configuration of the external sensor's register has to be performed using the internal trigger signal (START_CONFIG=0). After the external sensor is activated and the Data-Ready signal is available, the external trigger signal can be used by switching the START_CONFIG bit to 1.

- PULL_UP_EN bit enables/disables the internal pull-up on the auxiliary I²C line. When this bit is set to 0, the internal pull-up is disabled and the external pull-up resistors on the SDx/SCx pins are required, as shown in *Figure 25*. When this bit is set to 1, the internal pull-up is enabled and the external pull-up resistors on the SDx/SCx pins are not required.
- PASS_THROUGH_MODE bit is used to enable/disable the I²C interface pass-through. When this bit is set to 1, the main I²C line (e.g. connected to an external microcontroller) is short-circuited with the auxiliary one, in order to implement a direct access to the external sensor registers. See *Section 7.3* for details.
- MASTER_ON bit has to be set to 1 to enable the auxiliary I²C master of the LSM6DS3 device (sensor hub mode).

7.2.3 FUNC_SRC (53h)

Table 36. FUNC_SRC register

b7	b6	b5	b4	b3	b2	b1	b0
0	X	X	X	0	0	X	SENSOR HUB _END_OP

- SENSORHUB_END_OP bit is set high when the sensor hub routine is completed and the external sensor data are available to be read from the SENSORHUBx_REG registers (depending on the configuration of the SLVx_ADD, SLVx_SUBADD, SLAVEx_CONFIG registers). This signal can be driven to the INT1 interrupt pin by setting to 1 the DRDY_ON_INT1 bit of the MASTER_CONFIG register. The SENSORHUB_END_OP bit is cleared by reading the FUNC_SRC register.

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

The embedded function registers (accessible when the FUNC_CFG_EN bit of the FUNC_CFG_ACCESS register is set to 1) used to configure the I^{2}C slave interface associated to the first external sensor are described hereafter.

Table 37. SLV0_ADD register

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] bits are used to indicate the I^{2}C slave address of the first external sensor.
- rw_0 bit configures the read/write operation to be performed on the first external sensor (0: write operation; 1: read operation). The read/write operation is executed when the next sensor hub trigger event occurs.

Table 38. SLV0_SUBADD register

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] bits are used to indicate the address of the register of the first external sensor to be written (if the rw_0 bit of the SLV0_ADD register is set to 0) or the address of the first register to be read (if the rw_0 bit of the SLV0_ADD register is set to 1).

Table 39. SLAVE0_CONFIG register

b7	b6	b5	b4	b3	b2	b1	b0
Slave0 _rate1	Slave0 _rate0	Aux_sens_ on1	Aux_sens_ on0	Src_mode	Slave0 _numop2	Slave0 _numop1	Slave0 _numop0

- Slave0_rate[1:0] bits are used to define the decimation factor applied to read operations on the first external sensor starting from the sensor hub trigger:
  - 00: no decimation
  - 01: update every 2 sensor hub trigger events
  - 10: update every 4 sensor hub trigger events
  - 11: update every 8 sensor hub trigger events
- Aux_sens_on[1:0] bits have to be used to indicate the number of external sensors to be managed by the sensor hub:
  - 00: one external sensor
  - 01: two external sensors
  - 10: three external sensors
  - 11: four external sensors
- Src_mode bit enables/disabled Source Mode conditioned reading. When this bit is set to 1, Source Mode conditioned reading is enabled; before proceeding with the reading of the register address indicated in the SLV0_SUBADD register, the content of the

register at the address specified in DATAWRITE_SRC_MODE_SUB_SLV0 is checked: if the content is non-zero the operation continues, else the reading operation is interrupted. Source Mode conditioned reading is available on slave 0 only.

- Slave0_numop[2:0] bits are dedicated to define the number of consecutive read operations to be performed on the first external sensor starting from the register address indicated in the SLV0_SUBADD register.

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

The embedded function registers (accessible when the FUNC_CFG_EN bit of the FUNC_CFG_ACCESS register is set to 1) used to configure the I^2C slave interface associated to the second external sensor are described hereafter.

Table 40. SLV1_ADD register

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] bits are used to indicate the I^2C slave address of the second external sensor.
- r_1 bit enables/disables the read operation to be performed on the second external sensor (0: read operation disabled; 1: read operation enabled). The read operation is executed when the next sensor hub trigger event occurs.

Table 41. SLV1_SUBADD register

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] bits are used to indicate the address of the register of the second external sensor to be read when the r_1 bit of SLV1_ADD register is set to 1.

Table 42. SLAVE1_CONFIG register

b7	b6	b5	b4	b3	b2	b1	b0
Slave1 _rate1	Slave1 _rate0	0	0	0	Slave1 _numop2	Slave1 _numop1	Slave1 _numop0

- Slave1_rate[1:0] bits are used to define the decimation factor applied to read operations on the second external sensor starting from the sensor hub trigger:
  - 00: no decimation
  - 01: update every 2 sensor hub trigger events
  - 10: update every 4 sensor hub trigger events
  - 11: update every 8 sensor hub trigger events
- Slave1_numop[2:0] bits are dedicated to define the number of consecutive read operations to be performed on the second external sensor starting from the register address indicated in SLV1_SUBADD register.

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

The embedded function registers (accessible when the FUNC_CFG_EN bit of the FUNC_CFG_ACCESS register is set to 1) used to configure the I^2C slave interface associated to the third external sensor are described hereafter.

Table 43. SLV2_ADD register

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] bits are used to indicate the I^2C slave address of the third external sensor.
- r_2 bit enables/disables the read operation to be performed on the third external sensor (0: read operation disabled; 1: read operation enabled). The read operation is executed when the next sensor hub trigger event occurs.

Table 44. SLV2_SUBADD register

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] bits are used to indicate the address of the register of the third external sensor to be read when the r_2 bit of SLV2_ADD register is set to 1.

Table 45. SLAVE2_CONFIG register

b7	b6	b5	b4	b3	b2	b1	b0
Slave2 _rate1	Slave2 _rate0	0	0	0	Slave2 _numop2	Slave2 _numop1	Slave2 _numop0

- Slave2_rate[1:0] bits are used to define the decimation factor applied to read operations on the third external sensor starting from the sensor hub trigger:
  - 00: no decimation
  - 01: update every 2 sensor hub trigger events
  - 10: update every 4 sensor hub trigger events
  - 11: update every 8 sensor hub trigger events
- Slave2_numop[2:0] bits are dedicated to define the number of consecutive read operations to be performed on the third external sensor starting from the register address indicated in the SLV2_SUBADD register.

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

The embedded function registers (accessible when the FUNC_CFG_EN bit of FUNC_CFG_ACCESS register is set to 1) used to configure the I²C slave interface associated to the fourth external sensor are described hereafter.

Table 46. SLV3_ADD register

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] bits are used to indicate the I²C slave address of the fourth external sensor.
- r_3 bit enables/disables the read operation to be performed on the fourth external sensor (0: read operation disabled; 1: read operation enabled). The read operation is executed when the next sensor hub trigger event occurs.

Table 47. SLV3_SUBADD register

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] bits are used to indicate the address of the register of the fourth external sensor to be read when the r_3 bit of the SLV3_ADD register is set to 1.

Table 48. SLAVE3_CONFIG register

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

- Slave3_rate[1:0] bits are used to define the decimation factor applied to read operations on the fourth external sensor starting from the sensor hub trigger:
  - 00: no decimation
  - 01: update every 2 sensor hub trigger events
  - 10: update every 4 sensor hub trigger events
  - 11: update every 8 sensor hub trigger events
- Slave3_numop[2:0] bits are dedicated to define the number of consecutive read operations to be performed on the fourth external sensor starting from the register address indicated in the SLV3_SUBADD register.

7.2.8 DATAWRITE_SRC_MODE_SUB_SLV0 (0Eh)

Table 49. DATAWRITE_SRC_MODE_SUB_SLV0 register

b7	b6	b5	b4	b3	b2	b1	b0
Slave _dataw7	Slave _dataw6	Slave _dataw5	Slave _dataw4	Slave _dataw3	Slave _dataw2	Slave _dataw1	Slave _dataw0

- Slave_dataw[7:0] bits are dedicated, when the rw_0 bit of SLV0_ADD register is set to 0 (write operation), to indicate the data to be written to the first external sensor at the address specified in the SLV0_SUBADD register. During read operations (rw_0 = 1), this register is used if the Source Mode conditioned reading is enabled (Src_mode bit = 1 in the SLAVE0_CONFIG register) and it indicates the address of the external sensor register to be checked before proceeding with the reading operation.

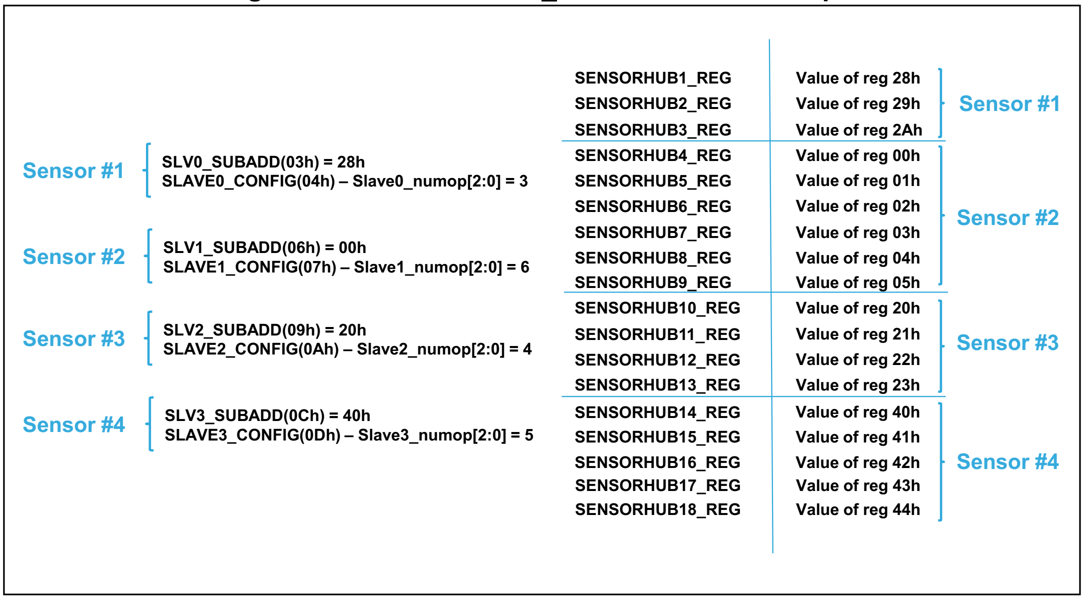
7.2.9 SENSORHUBx_REG registers

Once the auxiliary I^{2}C master is enabled, for each of the external sensors it reads a number of registers equal to the value of the Slavex_numop (x = 0, 1, 2, 3) field, starting from the register address specified in SLVx_SUBADD (x = 0, 1, 2, 3) register. The number of external sensors to be managed is specified in the Aux_sens_on bits of the SLAVE0_CONFIG register.

Read data are consecutively stored (in the same order they are read) in the LSM6DS3 registers starting from the SENSORHUB1_REG register, as in the example in *Figure 26*; 18 registers, from SENSORHUB1_REG to SENSORHUB18_REG, are available to store the data read from the external sensors.

The values of the registers from SENSORHUB1_REG to SENSORHUB6_REG can be saved in the FIFO buffer as a third data set; the values of the registers from SENSORHUB7_REG to SENSORHUB12_REG can be saved in the FIFO buffer as a fourth data set (see *Chapter 8* for details).

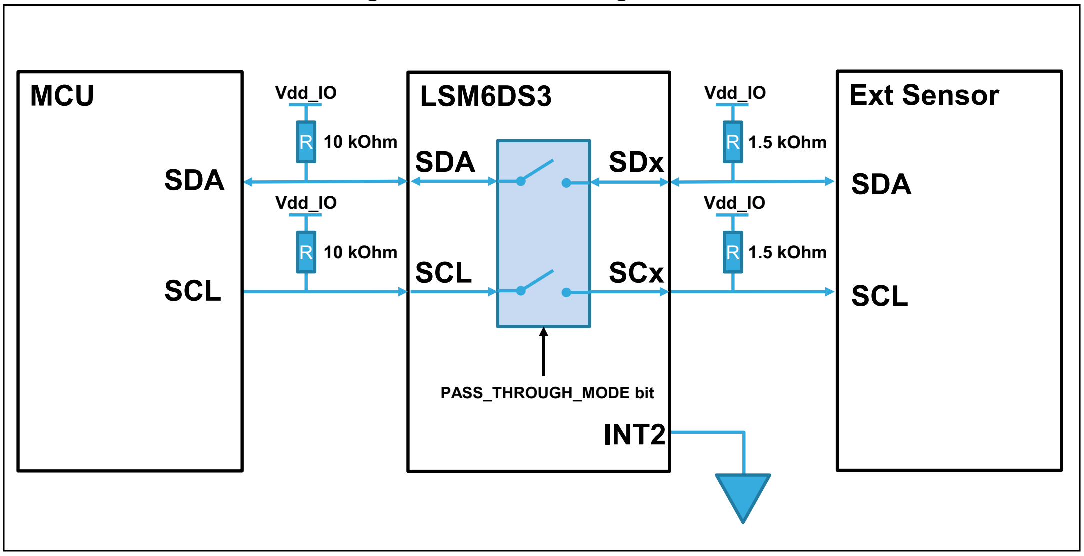
Figure 26. SENSORHUBx_REG allocation example



7.3 Sensor hub pass-through feature

The PASS_THROUGH_MODE bit of the MASTER_CONFIG register is used to enable/disable the I^{2}C interface pass-through: when it is set to 1, the main I^{2}C line (e.g. connected to an external microcontroller) is short-circuited with the auxiliary one, in order to implement a direct access to the external sensor registers. It is recommended to use this feature when configuring the external sensors.

Figure 27. Pass-through feature



Some limitations must be considered when using the sensor hub and the pass-through feature. Three different scenarios are possible:

1. The sensor hub is used with the START_CONFIG bit of the MASTER_CONFIG register set to 0 (internal trigger) and the pass-through feature is not used: there is no limitation on INT2 pin usage.
2. The sensor hub is used with the START_CONFIG bit of the MASTER_CONFIG register set to 0 (internal trigger) and the pass-through feature is used: the INT2 pin must be connected to GND; it's not possible to switch to external trigger config. (by setting the START_CONFIG bit to 1) and the INT2 pin cannot be used for the digital interrupts. Specific procedures have to be applied to enable/disable the pass-through feature: they are described in *Section 7.3.1* and in *Section 7.3.2*.
3. The sensor hub is used with the START_CONFIG bit of the MASTER_CONFIG register set to 1 (external trigger): the pass-through feature cannot be used; the INT2 pin has to be connected to the data-ready pin of the external sensor (trigger signal) and the procedure below has to be executed to avoid conflicts with the INT2 line:
   a) Set the LVLen bit of CTRL6_C register to 1 (to configure the INT2 pin as input pin);
   b) Configure the external sensors (do not use the pass-through);
   c) Configure the sensor hub SLAVEx registers;
   d) Set the START_CONFIG bit of the MASTER_CONFIG register to 1;
   e) Set the MASTER_ON bit of the MASTER_CONFIG register to 1;
   f) Set the LVLen bit of CTRL6_C register to 0.

Examples of external sensors configuration without using the pass-through are given in *Section 7.4* and *Section 7.5.4*.

7.3.1 Pass-through feature enable

When the embedded sensor hub functionality is disabled, the pass-through feature can be enabled at any time by setting the PASS_THROUGH_MODE bit of the MASTER_CONFIG register to 1.

When the embedded sensor hub functionality is enabled, a specific procedure has to be followed to enable the pass-through feature in order to prevent I^2C bus arbitration loss:

a) Set the START_CONFIG bit of the MASTER_CONFIG register to 1 in order to disable the sensor hub trigger (external trigger is enabled, but no trigger can be received on the INT2 pin since it's connected to GND);

b) Wait at least 5 ms (running I^2C operations will be completed);

c) Set the MASTER_ON bit of the MASTER_CONFIG register to 0 in order to disable the embedded sensor hub;

d) Set the START_CONFIG bit of the MASTER_CONFIG register to 0 in order to restore the sensor hub trigger;

e) Set the PULL_UP_EN bit of the MASTER_CONFIG register to 0 in order to disable the I^2C master pull-up;

f) Set the PASS_THROUGH_MODE bit of the MASTER_CONFIG register to 1 in order to enable the pass-through feature.

7.3.2 Pass-through feature disable

The procedure below has to be used in order to disable the pass-through:

a) Wait for the external microcontroller connected to the main I^2C line to complete all running I^2C operations. The pass-through must not be disabled in the middle of an I^2C transaction;

b) Set the PASS_THROUGH_MODE bit of the MASTER_CONFIG register to 0.

At this point the internal I^2C master pull-up can be restored by setting to 1 the PULL_UP_EN bit of the MASTER_CONFIG register, and the auxiliary I^2C master can be enabled by setting to 1 the MASTER_ON bit of the MASTER_CONFIG register.

7.4 Sensor hub mode example

The configuration of the external sensors should be performed using the pass-through feature: this feature can be enabled by setting the PASS_THROUGH_MODE bit of the MASTER_CONFIG register to 1 and implements a direct access to the external sensor registers, allowing quick configuration.

The code provided below gives a basic routine to configure the LSM6DS3 in sensor hub mode. Furthermore, this sequence configures the LIS3MDL external magnetometer sensor (refer to the datasheet for additional details) in continuous-conversion mode and reads the magnetometer output registers, saving their values in the SENSORHUB1_REG to SENSORHUB6_REG registers. The pass-through feature is not used in this example.

1	Write 80h into FUNC_CFG_ADDRESS	// Enable access to embedded functions registers
2	Write 38h into SLV0_ADD	// LIS3MDL slave address = 0011100b (if SDO=0) // Enable write operation (rw_0=0)

3	Write 22h into SLV0_SUBADD	// 22h is the LIS3MDL register to be written

4	Write 00h into DATAWRITE_SRC_MODE_SUB_SLV0	// 00h is the value to be written in register 22h of
		// LIS3MDL to configure it in continuous
		// conversion mode

5	Write 00h into FUNC_CFG_ADDRESS	// Disable access to embedded functions registers
6	Write 3Ch into CTRL10_C	// Enable embedded functions

7	Write 09h into MASTER_CONFIG	// Enable internal pull-up on SDx/SCx lines
		// Sensor hub trigger signal is XL Data Ready
		// Enable auxiliary I^2C master

8	Write 80h into CTRL1_XL	// Turn on the accelerometer (for trigger signal)
9	Write 38h into CTRL10_C	// Disable embedded functions
10	Write 00h into MASTER_CONFIG	// Disable auxiliary I^2C master
11	Write 00h into CTRL1_XL	// Turn off the accelerometer
12	Write 80h into FUNC_CFG_ADDRESS	// Enable access to embedded functions registers
13	Write 39h into SLV0_ADD	// LIS3MDL slave address = 0011100b (if SDO=0) // Enable read operation (rw_0=1)
14	Write 28h into SLV0_SUBADD	// 28h is the first LIS3MDL output register to be // read

15	Write 06h into SLAVE0_CONFIG	// No decimation
		// 1 external sensor connected
		// Number of registers to read = 6

16	Write 00h into FUNC_CFG_ADDRESS	// Disable access to embedded functions registers
17	Write 3Ch into CTRL10_C	// Enable embedded functions

18	Write 09h into MASTER_CONFIG	// Enable internal pull-up on SDx/SCx lines
		// Sensor hub trigger signal is XL Data Ready
		// Enable auxiliary I^2C master

19	Write 80h into CTRL1_XL	// Turn on the accelerometer (for trigger signal)

7.5 Magnetometer hard-iron / soft-iron correction

The LSM6DS3 device supports the data acquisition of an external magnetometer with soft-iron and hard-iron correction features. For this purpose, it is required to set the MASTER_ON bit of the MASTER_CONFIG register to 1 to enable the sensor hub mode, to associate the external magnetometer to slave 0 registers (SLV0_ADD, SLV0_SUBADD and SLAVE0_CONFIG) and to set the Slave0_numop field of SLAVE0_CONFIG to 6.

The FUNC_EN bit of CTRL10_C register has to be set to 1 in order to enable the embedded ironing functionalities. Then, distortion correction algorithms can be enabled as described in *Table 50*: the IRON_EN bit of MASTER_CONFIG and the SOFT_EN bit of CTRL9_XL are used to enable hard-iron correction only or both hard-iron and soft-iron corrections. In the latter case, both calibrated (hard-iron & soft-iron) and uncalibrated (soft-iron only) magnetometer data are available.

Table 50. Ironing configuration

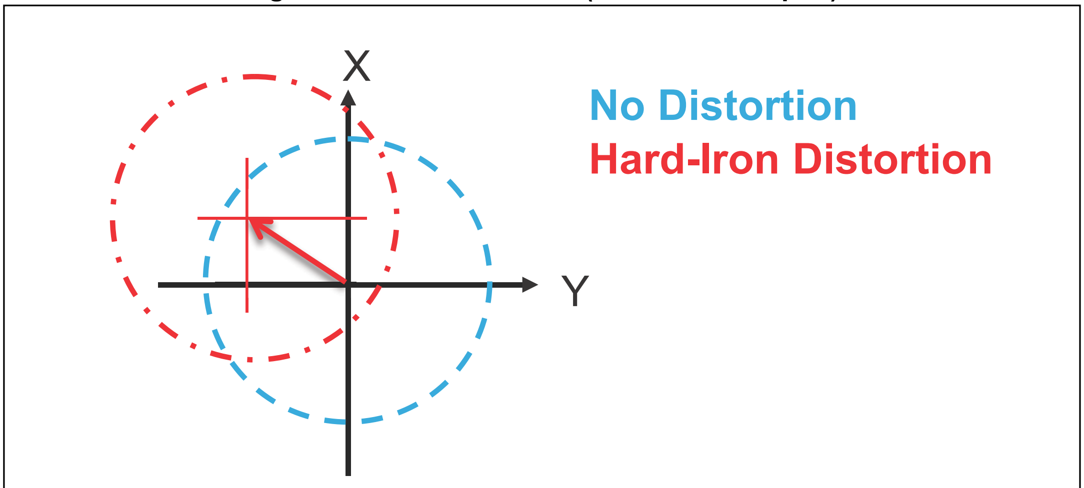
CTRL9_XL SOFT bit	MASTER_CONFIG IRON_EN bit	Ironing configuration
0	0	No correction applied
0	1	Hard-iron only
1	1	Hard-iron + soft-iron corrections

7.5.1 Hard-iron correction

Hard-iron distortion is normally generated by ferromagnetic material with permanent magnetic fields that are part of the object (e.g. a tablet) in use; these materials could be permanent magnets or magnetized iron or steel. They are time invariant and deform the local geomagnetic field with different offset on different directions.

Generally, if the user performs many 3D rotations of the object in an ideal environment (no hard-iron/soft-iron distortion) and plots the collected magnetic sensor raw data, the result will be a perfect sphere with no offset. The hard-iron distortion effect is to offset the sphere along the x, y and z axes; in the x-y plane, the hard-iron distortion is identified by an offset of the origin of the ideal circle from (0, 0), as shown in *Figure 28*.

Figure 28. Hard-iron effect (X-Y 2D scatter plot)



In the LSM6DS3 device, the 3x1 hard-iron vector containing the X, Y, Z magnetic offset values calculated by the user have to be indicated in dedicated registers: the MAG_OFFX_L and MAG_OFFX_H registers are dedicated to the X-axis offset, the MAG_OFFY_L and MAG_OFFY_H registers are dedicated to the Y-axis offset, the MAG_OFFZ_L and MAG_OFFZ_H registers are dedicated to the Z-axis offset. These registers values are expressed as a 16-bit word in two's complement; the sensitivity [LSB/Gauss] to be applied to calculate the hard-iron register values corresponds to that of the external magnetometer.

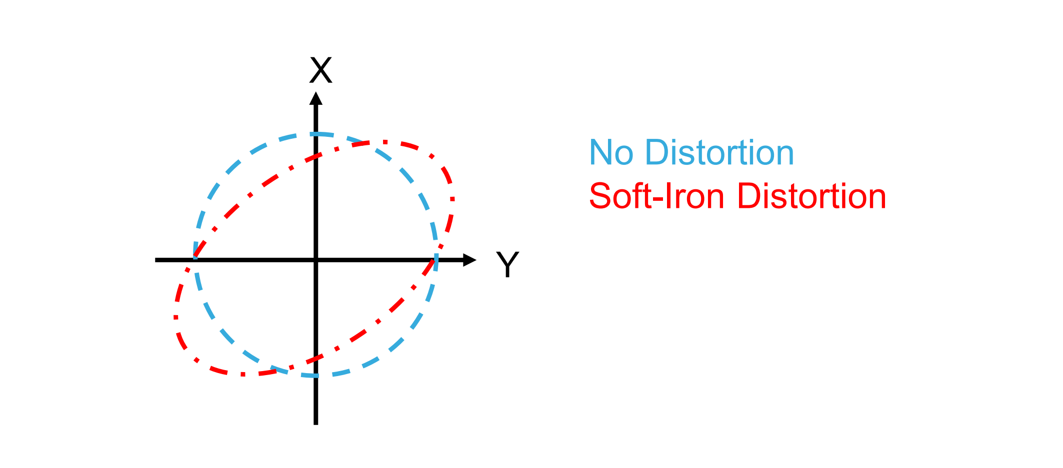
The hard-iron registers are accessible when the FUNC_CFG_EN bit of the FUNC_CFG_ACCESS register is set to 1. In order to enable the hard-iron correction algorithm, it is necessary to set to 1 both the FUNC_EN bit of the CTRL10_C register and the IRON_EN bit of the MASTER_CONFIG register (*Table 50*).

7.5.2 Soft-iron correction

Soft-iron distortion is generated by magnetically soft materials or current carrying PCB traces. While the hard-iron distortion is constant regardless of the orientation, the soft-iron distortion changes with the orientation of the object in the Earth's field. Basically, the local geomagnetic field is deformed with different gain on different directions.

The effect of the soft-iron distortion is to make the ideal full round sphere become a tilted ellipsoid; in the x-y plane, the soft-iron distortion is identified by a tilted ellipse with the origin in (0, 0), as shown in *Figure 29*.

Figure 29. Soft-iron effect (X-Y 2D scatter plot)



In the LSM6DS3 device, the 3x3 soft-iron rotation matrix calculated by the user has to be indicated in 9 dedicated registers: MAG_SI_XX, MAG_SI_XY, MAG_SI_XZ, MAG_SI_YX, MAG_SI_YY, MAG_SI_YZ, MAG_SI_ZX, MAG_SI_ZY, MAG_SI_ZZ. These register values are expressed as an 8-bit word in sign-module format; for these registers 1 LSB corresponds to 1/8, so the matrix parameters calculated by the user must be multiplied by 8 before writing them in the soft-iron registers.

The soft-iron registers are accessible when the FUNC_CFG_EN bit of the FUNC_CFG_ACCESS register is set to 1. In order to enable the soft-iron correction algorithm it is necessary to set to 1 the FUNC_EN bit of the CTRL10_C register, the IRON_EN bit of the MASTER_CONFIG register and the SOFT_EN bit of the CTRL9_XL register (*Table 50*).

7.5.3 Getting compensated magnetometer data

The status of magnetometer data acquisition and hard-iron/soft correction can be checked using the FUNC_SRC register:

- SENSORHUB_END_OP bit is set high when the sensor hub routine is completed. The acquired magnetometer raw data are available in registers from address 66h (OUT_MAG_RAW_X_L) to 6Bh (OUT_MAG_RAW_Z_H).
- SI_END_OP bit is set high when the execution of the enabled hard-iron and soft-iron algorithms has completed. If the soft-iron correction is enabled, the magnetometer uncalibrated data (with soft-iron only applied) are available in registers from address 4Dh (SENSORHUB13_REG) to 52h (SENSORHUB18_REG). The magnetometer calibrated data, with both hard-iron (if enabled) and soft-iron (if enabled) correction are

available in registers from address 2Eh (SENSORHUB1_REG) to 33h (SENSORHUB6_REG).

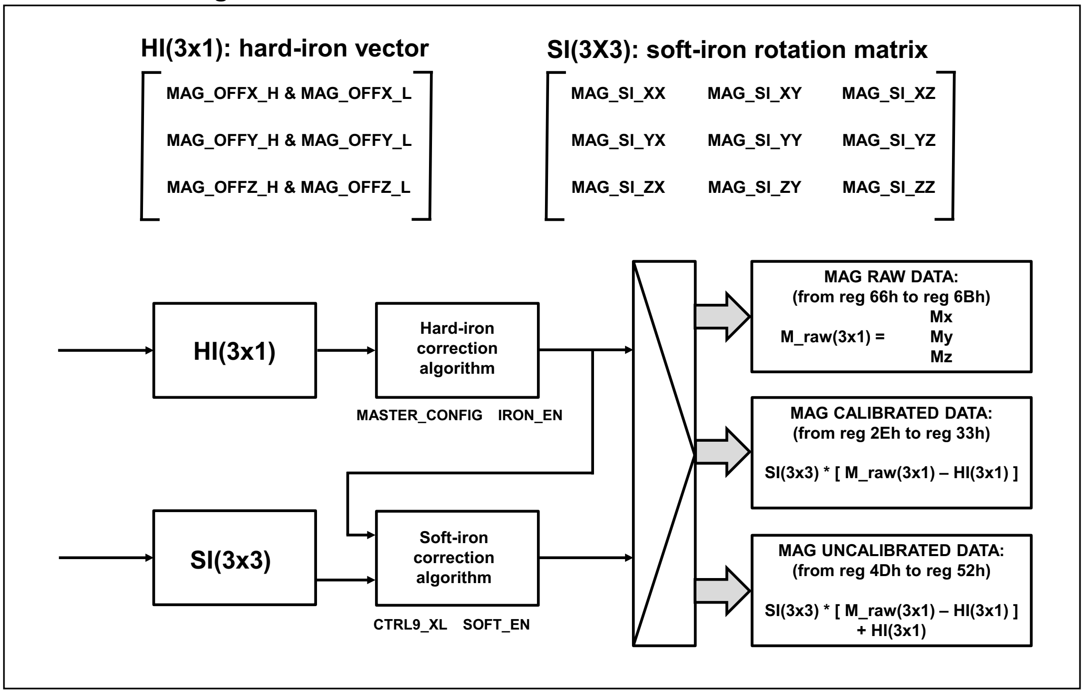
If latch mode is disabled (LIR bit of TAP_CFG is set to 0), the SENSORHUB_END_OP and SI_END_OP bits are active only for 1/100 Hz, then they are automatically deasserted. If latch mode is enabled, these two bits are cleared by reading the FUNC_SRC register.

The SENSORHUB_END_OP signal can be driven to the INT1 interrupt pin by setting to 1 the DRDY_ON_INT1 bit of the MASTER_CONFIG register. The SI_END_OP signal can be driven to the INT2 interrupt pin by setting to 1 the INT2_IRON bit of the MD2_CFG register.

A schematic representation of hard-iron and soft-iron correction feature is illustrated in *Figure 30* below.

If the soft-iron correction is enabled and the soft-iron registers still have the default zero value, then the magnetometer calibrated data and the magnetometer uncalibrated data will be equal to zero too. As a consequence, when the soft-iron correction is enabled, the soft-iron rotation matrix must be at least initialized to the identity matrix multiplied by 8, setting the value of the MAG_SI_XX, MAG_SI_YY and MAG_SI_ZZ registers to 08h.

Figure 30. Hard-iron / soft-iron correction block scheme



7.5.4 Ironing example

The following example demonstrates how to define the values to be assigned to hard-iron and soft-iron correction registers starting from the calculated hard-iron vector and soft-iron rotation matrix. This example refers to the usage of the LIS3MDL magnetometer sensor with 4 gauss of configured full-scale (sensitivity = 6842 LSB/gauss).

- Hard-iron (X,Y,Z) offset values vector (gauss):

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

These three offset values must be multiplied by the LIS3MDL sensitivity value (6842 LSB/gauss for full-scale = 4 gauss) in order to get the LSB values to be written in the hard-iron correction registers (*Table 51*).

Table 51. Hard-iron register values

	Offset values [LSB]	Register values
X	-2296 (F708h)	MAG_OFFX_H = F7h MAG_OFFX_L = 08h
Y	+865 (0361h)	MAG_OFFY_H = 03h MAG_OFFY_L = 61h
Z	-785 (FCEFh)	MAG_OFFZ_H = FCh MAG_OFFZ_L = EFh

- Soft-iron rotation matrix:

$$SI(3x3) = \begin{bmatrix} 1.229006 & 0.173917 & 0.052327 \\ 0.173917 & 1.033307 & -0.130089 \\ 0.052327 & -0.130089 & 1.243645 \end{bmatrix}$$

These soft-iron matrix elements must be multiplied by 8 in order to get the LSB values to be written in the soft-iron correction registers (*Table 52*).

Table 52. Soft-iron register values

	Soft-iron matrix elements	Register values
XX	+1.229006	MAG_SI_XX = 0Ah
XY	+0.173917	MAG_SI_XY = 01h
XZ	+0.052327	MAG_SI_XZ = 00h
YX	+0.173917	MAG_SI_YX = 01h
YY	+1.033307	MAG_SI_YY = 08h
YZ	-0.130089	MAG_SI_YZ = 81h
ZX	+0.052327	MAG_SI_ZX = 00h
ZY	-0.130089	MAG_SI_ZY = 81h
ZZ	+1.243645	MAG_SI_ZZ = 0Ah

天猫店铺网址:https://telesky.tmall.com/

The code provided below gives a basic routine to configure the LIS3MDL external magnetometer sensor (refer to the datasheet for additional details) in continuous-conversion mode, initialize the hard-iron and soft-iron correction registers and read the magnetometer output registers. In this case, the pass-through feature is not used for the magnetometer configuration.

1	Write 80h into FUNC_CFG_ADDRESS	// Enable access to embedded functions registers
2	Write 38h into SLV0_ADD	// LIS3MDL slave address = 0011100b (if SDO=0) // Enable write operation (rw_0=0)
3	Write 22h into SLV0_SUBADD	// 22h is the LIS3MDL register to be written

4	Write 00h into DATAWRITE_SRC_MODE_SUB_SLV0	// 00h is the value to be written in register 22h of
		// LIS3MDL to configure it in continuous
		// conversion mode

5	Write 00h into FUNC_CFG_ADDRESS	// Disable access to embedded functions registers
6	Write 3Ch into CTRL10_C	// Enable embedded functions

7	Write 09h into MASTER_CONFIG	// Enable internal pull-up on SDx/SCx lines
		// Sensor hub trigger signal is XL Data Ready
		// Enable auxiliary I^{2}C master

8	Write 80h into CTRL1_XL	// Turn on the accelerometer (for trigger signal)
9	Write 38h into CTRL10_C	// Disable embedded functions
10	Write 00h into MASTER_CONFIG	// Disable auxiliary I^{2}C master
11	Write 00h into CTRL1_XL	// Turn off the accelerometer
12	Write 80h into FUNC_CFG_ADDRESS	// Enable access to embedded functions registers
13	Write 39h into SLV0_ADD	// LIS3MDL slave address = 0011100b (if SDO=0) // Enable read operation (rw_0=1)
14	Write 28h into SLV0_SUBADD	// 28h is the first LIS3MDL output register to be // read

15	Write 06h into SLAVE0_CONFIG	// No decimation
		// 1 external sensor connected
		// Number of registers to read = 6

16	Write F7h into MAG_OFFX_H	// X offset value initialization
17	Write 08h into MAG_OFFX_L	// X offset value initialization
18	Write 03h into MAG_OFFY_H	// Y offset value initialization
19	Write 61h into MAG_OFFY_L	// Y offset value initialization
20	Write FCh into MAG_OFFZ_H	// Z offset value initialization
21	Write EFh into MAG_OFFZ_L	// Z offset value initialization
22	Write 0Ah into MAG_SI_XX	// XX soft-iron element
23	Write 01h into MAG_SI_XY	// XY soft-iron element
24	Write 00h into MAG_SI_XZ	// XZ soft-iron element

天猫店铺网址:https://telesky.tmall.com/

25	Write 01h into MAG_SI_YX	// YX soft-iron element
26	Write 08h into MAG_SI_YY	// YY soft-iron element
27	Write 81h into MAG_SI_YZ	// YZ soft-iron element
28	Write 00h into MAG_SI_ZX	// ZX soft-iron element
29	Write 81h into MAG_SI_ZY	// ZY soft-iron element
30	Write 0Ah into MAG_SI_ZZ	// ZZ soft-iron element
31	Write 00h into FUNC_CFG_ADDRESS	// Disable access to embedded functions registers
32	Write 3Ch into CTRL10_C	// Enable embedded functions

33	Write 0Bh into MASTER_CONFIG	// Enable internal pull-up on SDx/SCx lines
		// Sensor hub trigger signal is XL data-ready
		// Enable hard-iron correction
		// Enable auxiliary I^{2}C master

34	Write 3Ch into CTRL9_XL	// Enable soft-iron correction
35	Write 80h into CTRL1_XL	// Turn on the accelerometer (for trigger signal)

The acquired magnetometer raw data are available in registers from address 66h (OUT_MAG_RAW_X_L) to 6Bh (OUT_MAG_RAW_Z_L).

The magnetometer uncalibrated data (with soft-iron only applied) are available in registers from address 4Dh (SENSORHUB13_REG) to 52h (SENSORHUB18_REG).

The magnetometer calibrated data, with both hard-iron and soft-iron correction are available in registers from address 2Eh (SENSORHUB1_REG) to 33h (SENSORHUB6_REG).

8 First-in first-out (FIFO) buffer

In order to limit intervention by the host processor and facilitate post-processing data for event recognition, the LSM6DS3 embeds an 8 kbyte first-in first-out buffer (FIFO).

The FIFO can be configured to store the following data:

- gyroscope sensor data;
- accelerometer sensor data;
- external sensors (connected to sensor hub interface) data;
- step counter and timestamp data;
- temperature sensor data.

Saving data in the FIFO buffer is based on four 'FIFO data set' consisting of 6 bytes each:

- The 1st FIFO data set is reserved for gyroscope data;
- The 2nd FIFO data set is reserved for accelerometer data;
- The 3rd FIFO data set is reserved for the external sensor data stored in the registers from SENSORHUB1_REG to SENSORHUB6_REG (see *Section 7.2.9* for details on the SENSORHUBx_REG);
- The 4th FIFO data set can be alternately associated to the external sensor data stored in the registers from SENSORHUB7_REG to SENSORHUB12_REG, to the step counter and timestamp info, or to the temperature sensor data.

All these data sets can be stored in FIFO at different ODRs, by setting the decimation factors in the FIFO_CTRL3 and FIFO_CTRL4 registers. Decimation factors are also used to select which FIFO data sets have to be stored in FIFO.

Five different FIFO operating modes can be chosen through the FIFO_MODE_[2:0] bits of the FIFO_CTRL5 register:

- Bypass mode;
- FIFO mode;
- Continuous mode;
- Continuous-to-FIFO mode;
- Bypass-to-Continuous mode.

Note: When the FIFO is used, the IF_INC bit of the CTRL3_C register must be equal to 1.

Data are retrieved from the FIFO through two dedicated registers: FIFO_DATA_OUT_L and FIFO_DATA_OUT_H. In this way, data can be read either from the FIFO (at a slower ODR) or from the device output registers (at the normal ODR).

To monitor the FIFO status (full, empty, number of sample stored, etc), four dedicated registers are available: FIFO_STATUS1, FIFO_STATUS2, FIFO_STATUS3, FIFO_STATUS4.

Programmable FIFO thresholds can be set in FIFO_CTRL1 and FIFO_CTRL2 using the FTH [11:0] bits.

FIFO full, FIFO threshold and FIFO overrun events can be enabled to generate dedicated interrupts on the two interrupt pins (INT1 and INT2) through the INT1_FULL_FLAG, INT1_FTH and INT1_OVR bits of the INT1_CTRL register, and through the INT2_FULL_FLAG, INT2_FTH and INT2_OVR bits of the INT2_CTRL register.

天猫店铺网址:https://telesky.tmall.com/

In order to increase the number of samples which can be stored in the FIFO, it is also possible to store (as 1st FIFO data set) only the 8 most significant bits of the accelerometer and gyroscope data by setting the bit ONLY_HIGH_DATA in the FIFO_CTRL4 register.

Writing data in the FIFO can be triggered by the accelerometer/gyroscope data-ready; it can also be triggered by the sensor hub data-ready (corresponding to the behavior of the SENSORHUB_END_OP bit of FUNC_SRC register): in this case the DATA_VALID_SEL_FIFO bit of the MASTER_CONFIG register must be set to 1. Moreover, if DATA_VALID_SEL_FIFO is set to 0 and the TIMER_PEDO_FIFO_DRDY bit of the FIFO_CTRL2 register is set to 1, the data are stored in FIFO every time a step is detected.

8.1 FIFO registers

The FIFO buffer is managed by:

- five control registers (from FIFO_CTRL1 to FIFO_CTRL5);
- four status registers (from FIFO_STATUS1 to FIFO_STATUS4);
- two data output registers (FIFO_DATA_OUT_L and FIFO_DATA_OUT_H);
- some additional bits to enable threshold usage (STOP_ON_FTH) and route FIFO full, threshold or overrun events to the two interrupt lines (bits: INT1_FULL_FLAG, INT2_FULL_FLAG, INT1_FTH, INT2_FTH, INT1_FIFO_OVR, INT2_FIFO_OVR).

8.1.1 FIFO_CTRL1 (06h)

The FIFO_CTRL1 register contains the lower part of the 12-bit FIFO threshold level. For the complete threshold level configuration, consider also the FTH_[11:8] bits of the FIFO_CTRL2 register. The value of the FIFO threshold level is referred to data having 16-bit format.

The FIFO watermark flag (FTH bit in FIFO_STATUS2 register) rises when the number of bytes stored in the FIFO is equal to or higher than the threshold level.

In order to limit the FIFO depth to the watermark level, the STOP_ON_FTH bit must be set to 1 in the CTRL4_C register.

Table 53. FIFO_CTRL1 register

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

天猫店铺网址:https://telesky.tmall.com/

8.1.2 FIFO_CTRL2 (07h)

Table 54. FIFO_CTRL2 register

b7	b6	b5	b4	b3	b2	b1	b0
TIMER_PEDO_FIFO_EN	TIMER_PEDO_FIFO_DRDY	0	0	FTH_11	FTH_10	FTH_9	FTH_8

- TIMER_PEDO_FIFO_EN enables step counter and timestamp data to be stored as the 4th FIFO data set. The content of the 6 bytes stored in the FIFO when this bit is set to 1 is described in *Section 8.8*.
- TIMER_PEDO_FIFO_DRDY. When this bit is set to 1 and the DATA_VALID_SEL_FIFO bit in the MASTER_CONFIG register is set to 0, all the data are stored in the FIFO every time a new step has been detected by the step counter. See *Section 8.3* for details.
- FTH_[11:8] contains the upper part of the FIFO threshold level. For the complete threshold level configuration, consider also the FTH_[7:0] bits in the FIFO_CTRL1 register.

8.1.3 FIFO_CTRL3 (08h)

The FIFO_CTRL3 register contains the accelerometer and gyroscope FIFO decimation factors, used to choose if the data of these sensors have to be stored in the FIFO and at which rate they are stored.

When the DEC_FIFO_GYRO[2:0] bits are set to 000b, the 1st FIFO data set (reserved for gyroscope data) is not stored in the FIFO. When the DEC_FIFO_XL[2:0] bits are set to 000b, the 2nd FIFO data set (reserved for accelerometer data) is not stored in the FIFO.

Note: It's required to set at least one of the four decimation factors to 1 (no decimation).

Table 55. FIFO_CTRL3 register

b7	b6	b5	b4	b3	b2	b1	b0
0	0	DEC_FIFO_GYRO2	DEC_FIFO_GYRO1	DEC_FIFO_GYRO0	DEC_FIFO_XL2	DEC_FIFO_XL1	DEC_FIFO_XL0

Table 56. Gyroscope FIFO decimation setting

DEC_FIFO_GYRO [2:0]	Configuration
000	Gyroscope sensor not in FIFO
001	No decimation
010	Decimation with factor 2
011	Decimation with factor 3
100	Decimation with factor 4
101	Decimation with factor 8
110	Decimation with factor 16
111	Decimation with factor 32

Table 57. Accelerometer FIFO decimation setting

DEC_FIFO_XL [2:0]	Configuration
000	Accelerometer sensor not in FIFO
001	No decimation
010	Decimation with factor 2
011	Decimation with factor 3
100	Decimation with factor 4
101	Decimation with factor 8
110	Decimation with factor 16
111	Decimation with factor 32

8.1.4 FIFO_CTRL4 (09h)

The FIFO_CTRL4 register contains the decimation factors used to define at which data rate the data associated to the 3rd FIFO and the 4th FIFO data sets are stored in the FIFO.

When the DEC_DS3_FIFO[2:0] bits are set to 000b, the 3rd FIFO data set is not stored in the FIFO. When the DEC_DS4_FIFO[2:0] bits are set to 000b, the 4th FIFO data set is not stored in the FIFO.

Note: It's required to set at least one of the four decimation factors to 1 (no decimation).

The FIFO_CTRL4 register also contains the bit ONLY_HIGH_DATA, which allows storing in the FIFO only the upper part (Most Significant Byte) of accelerometer and gyroscope data, in order to increase the maximum number of accelerometer and gyroscope samples in the FIFO. See *Section 8.7* for more details about this functionality.

Table 58. FIFO_CTRL4 register

b7	b6	b5	b4	b3	b2	b1	b0
0	ONLY_HIGH _DATA	DEC_DS4 _FIFO2	DEC_DS4 _FIFO1	DEC_DS4 _FIFO0	DEC_DS3 _FIFO2	DEC_DS3 _FIFO1	DEC_DS3 _FIFO0

Table 59. 3rd FIFO data set decimation setting

DEC_DS3_FIFO [2:0]	Configuration
000	3rd FIFO data set not in FIFO
001	No decimation
010	Decimation with factor 2
011	Decimation with factor 3
100	Decimation with factor 4
101	Decimation with factor 8
110	Decimation with factor 16
111	Decimation with factor 32

Table 60. 4th FIFO data set decimation setting

DEC_DS4_FIFO [2:0]	Configuration
000	4th FIFO data set not in FIFO
001	No decimation
010	Decimation with factor 2
011	Decimation with factor 3
100	Decimation with factor 4
101	Decimation with factor 8
110	Decimation with factor 16
111	Decimation with factor 32

8.1.5 FIFO_CTRL5 (0Ah)

The FIFO_CTRL5 register contains the FIFO operating mode bits (FIFO_MODE_[2:0]) and the FIFO output data rate bits (ODR_FIFO_[3:0]).

FIFO operating modes (*Table 63*) are described in *Section 8.2*.

When the internal trigger (accelerometer/gyroscope data-ready) is used, the ODR_FIFO_[3:0] bits define the maximum data rate at which data are stored in FIFO. Data can be stored in FIFO at a lower data rate using the FIFO decimation factors. For more information about FIFO trigger and FIFO ODR configuration see *Section 8.3*.

Note: When the FIFO is used, the IF_INC bit of the CTRL3_C register must be equal to 1.

Table 61. FIFO_CTRL5 register

b7	b6	b5	b4	b3	b2	b1	b0
0	ODR _FIFO_3	ODR _FIFO_2	ODR _FIFO_1	ODR _FIFO_0	FIFO _MODE_2	FIFO _MODE_1	FIFO _MODE_0

Table 62. FIFO ODR selection setting

ODR_FIFO [3:0]	Configuration
0000	FIFO disabled
0001	FIFO ODR is set to 12.5 Hz
0010	FIFO ODR is set to 26 Hz
0011	FIFO ODR is set to 52 Hz
0100	FIFO ODR is set to 104 Hz
0101	FIFO ODR is set to 208 Hz
0110	FIFO ODR is set to 416 Hz
0111	FIFO ODR is set to 833 Hz
1000	FIFO ODR is set to 1.66 kHz
1001	FIFO ODR is set to 3.33 kHz
1010	FIFO ODR is set to 6.66 kHz

Table 63. FIFO mode selection

FIFO_MODE [2:0]	Configuration
000	Bypass mode. FIFO disabled.
001	FIFO mode. Stops collecting data when FIFO is full.
010	Reserved
011	Continuous mode until trigger is deasserted, then FIFO mode.
100	Bypass mode until trigger is deasserted, then Continuous mode.
101	Reserved
110	Continuous mode. If the FIFO is full, the new sample overwrites the older one.
111	Reserved

8.1.6 FIFO_STATUS1 (3Ah)

The FIFO_STATUS1 register, together with the FIFO_STATUS2 register, provides information about the number of samples stored in the FIFO. Each sample is represented as 16-bit data.

Table 64. FIFO_STATUS1 register

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

8.1.7 FIFO_STATUS2 (3Bh)

The FIFO_STATUS2 register, together with the FIFO_STATUS1 register, provides information about the number of samples stored in the FIFO and about the current status (threshold, overrun, full, empty) of the FIFO buffer.

Table 65. FIFO_STATUS2 register

b7	b6	b5	b4	b3	b2	b1	b0
FTH	FIFO_ OVER_RUN	FIFO_ FULL	FIFO_ EMPTY	DIFF_ FIFO_11	DIFF_ FIFO_10	DIFF_ FIFO_9	DIFF_ FIFO_8

- FTH represents the watermark status. This bit is set high when the number of bytes already stored in the FIFO is equal to or higher than the watermark level (each sample is represented as 16-bit data). The watermark status can be driven to the two interrupt pins by setting to 1 the INT1_FTH bit of the INT1_CTRL register or the INT2_FTH bit of the INT2_CTRL register.
- FIFO_OVER_RUN is set high when the FIFO is completely filled and at least one sample has already been overwritten to store the new data. This signal can be driven to the two interrupt pins by setting to 1 the INT1_FIFO_OVR bit of the INT1_CTRL register or the INT2_FIFO_OVR bit of the INT2_CTRL register.
- FIFO_FULL is set high when the next set of data that will be stored in FIFO will make the FIFO full. This signal can be driven to the two interrupt pins by setting to 1 the

INT1_FULL_FLAG bit of the INT1_CTRL register or the INT2_FULL_FLAG bit of the INT2_CTRL register.

- FIFO_EMPTY is set high when the FIFO is empty.
- DIFF_FIFO_[11:8] contains the upper part of the number of unread words (16-bit data) stored in the FIFO. The lower part is represented by the DIFF_FIFO_[7:0] bits in FIFO_STATUS1. The value of DIFF_FIFO_[11:0] field corresponds to the number of samples in the FIFO (each sample is represented as 16-bit data). When a FIFO overrun event occurs (FIFO_OVER_RUN bit is set high), the value of the DIFF_FIFO_[11:0] field is set to 0.

Register content is updated synchronously to the FIFO write and read operation, as illustrated in *Table 66*.

Table 66. FIFO_STATUS2 behavior (case with one sensor in FIFO, STOP_ON_FTH = 0)

FIFO_OVER_RUN	FIFO_FULL	FIFO_EMPTY	DIFF_FIFO_[11:0]	Number of FIFO samples	FIFO trigger timing
0	0	1	0	0	t0
0	0	0	3	3	t1
0	0	0	6	6	t2
...	...	...	...	...	...
0	0	0	4092	4092	t_full - 2
0	1	0	4095	4095	t_full - 1
1	1	0	0	4096 (old sample overwritten)	t_full

8.1.8 FIFO_STATUS3 (3Ch)

The FIFO_STATUS3 register, together with FIFO_STATUS4 register, specifies which axis of which sensor data will be read at the next reading. For more information on how to retrieve data from the FIFO see *Section 8.5*.

Table 67. FIFO_STATUS3 register

b7	b6	b5	b4	b3	b2	b1	b0
FIFO_PATTERN_7	FIFO_PATTERN_6	FIFO_PATTERN_5	FIFO_PATTERN_4	FIFO_PATTERN_3	FIFO_PATTERN_2	FIFO_PATTERN_1	FIFO_PATTERN_0

8.1.9 FIFO_STATUS4 (3Dh)

The FIFO_STATUS4 register, together with the FIFO_STATUS3 register, specifies which axis of which sensor data will be read at the next reading. For more information on how to retrieve data from the FIFO see Section 8.5.

Table 68. FIFO_STATUS4 register

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	FIFO_ PATTERN_9	FIFO_ PATTERN_8

8.1.10 FIFO_DATA_OUT_L (3Eh)

The FIFO_DATA_OUT_L register is the least significant byte of the FIFO output data. The most significant byte is stored in the FIFO_DATA_OUT_H register. For more information on how to retrieve data from the FIFO, see Section 8.4.

.

Table 69. FIFO_DATA_OUT_L register

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

8.1.11 FIFO_DATA_OUT_H (3Fh)

The FIFO_DATA_OUT_H register is the most significant byte of the FIFO output data. The least significant byte is stored in the FIFO_DATA_OUT_L register. For more information on how to retrieve data from the FIFO, see Section 8.4.

.

Table 70. FIFO_DATA_OUT_H register

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

8.2 FIFO modes

The LSM6DS3 FIFO buffer can be configured to operate in five different modes selectable through the FIFO_MODE_[2:0] field of the FIFO_CTRL5 register. The available configurations ensure a high level of flexibility and extend the number of functions usable in application development.

Bypass, FIFO, Continuous, Continuous-to-FIFO and Bypass-to-Continuous modes are described in the following paragraphs.

Note: When the FIFO is used, the IF_INC bit of the CTRL3_C register must be equal to 1.

天猫店铺网址:https://telesky.tmall.com/

8.2.1 Bypass mode

When Bypass mode is enabled, the FIFO is not used, the buffer content is cleared, and it remains empty until another mode is selected.

Bypass mode is selected when the FIFO_MODE_[2:0] bits are set to 000b. When this mode is enabled, the FIFO_STATUS2 register contains the value 10h (FIFO empty).

Bypass mode must be used in order to stop and reset the FIFO buffer when a different mode is operating. Note that placing the FIFO buffer into Bypass mode, the whole buffer content is cleared.

After Bypass mode is set, it's necessary to wait at least 30 μs before setting a different FIFO operating mode.

8.2.2 FIFO mode

In FIFO mode, the buffer continues filling until it becomes full. Then it stops collecting data and the FIFO content remains unchanged until a different mode is selected.

Follow these steps for FIFO mode configuration (if accelerometer/gyroscope data-ready is used as FIFO trigger):

1. Choose the decimation factor for each sensor through the decimation bits in the FIFO_CTRL3 and FIFO_CTRL4 registers (see *Section 8.3* for details);
2. Choose the FIFO ODR through the ODR_FIFO_[3:0] bits in the FIFO_CTRL5 register;
3. Set the FIFO_MODE_[2:0] bits in the FIFO_CTRL5 register to 001b to enable the FIFO mode.

When this mode is selected, the FIFO starts collecting data. The FIFO_STATUS1 and FIFO_STATUS2 registers are updated according to the number of samples stored.

When the next stored set of data will make the FIFO full, the FIFO_FULL bit of the FIFO_STATUS2 register is set to 1 and no more data are stored in the FIFO buffer. Data can be retrieved after the FIFO_FULL event, by reading the FIFO_DATA_OUT_L and FIFO_DATA_OUT_H registers for the number of times specified by the DIFF_FIFO_[11:0] bits of the FIFO_STATUS1 and FIFO_STATUS2 registers.

Using the FTH bit of the FIFO_STATUS2 register, data can also be retrieved when a threshold level (FTH_[11:0] in FIFO_CTRL1 and FIFO_CTRL2 registers) is reached, if the application requires a lower number of samples in the FIFO.

If the STOP_ON_FTH bit of the CTRL4_C register is set to 1, the FIFO size is limited to the value of the FTH_[11:0] bits in the FIFO_CTRL1 and FIFO_CTRL2 registers: in this case, the FIFO_FULL bit of the FIFO_STATUS2 register is set high when the number of samples in FIFO will reach or exceed the FTH_[11:0] value on the next FIFO write operation.

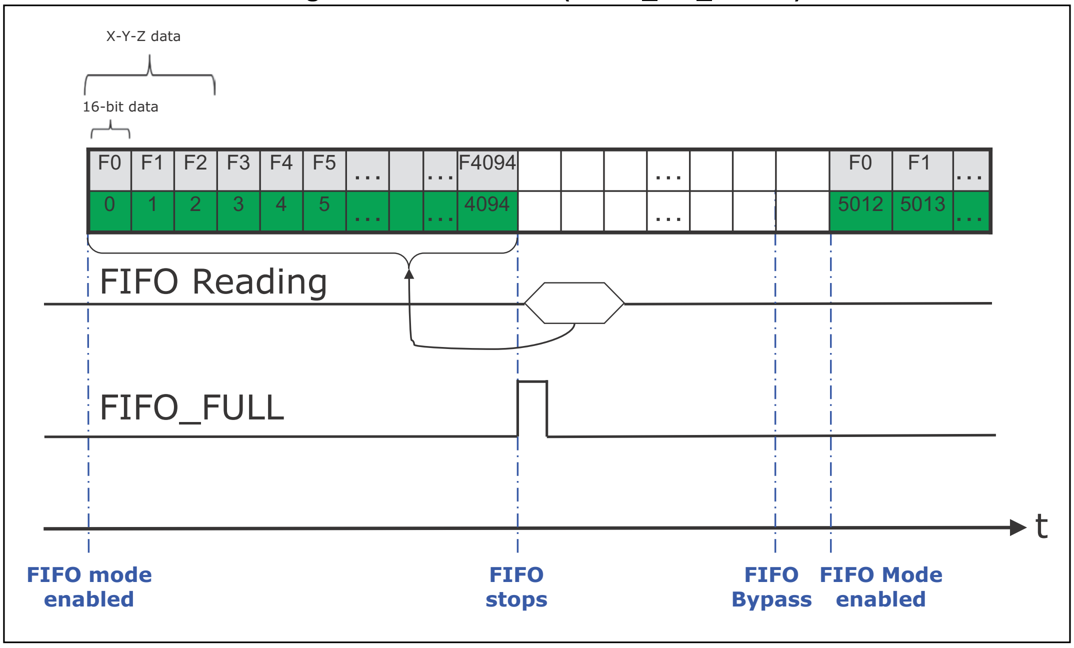
In case the FIFO is read before the FIFO_FULL event, it must not be completely emptied in order to avoid the misalignment of the data read from it. At least one complete FIFO pattern has to be left in the FIFO buffer (do not read it); it will be the first data read in the next read operation.

In case FIFO gets emptied after the FIFO_OUT registers are read, a FIFO reset (through Bypass mode setting) is needed.

Communication speed is not very important in FIFO mode because the data collection is stopped and there is no risk of overwriting data already acquired. Before restarting the FIFO mode, it is necessary to set to Bypass mode first, in order to completely clear the FIFO content.

Figure 31 shows an example of FIFO mode usage. In the example X-Y-Z data (green cells indicate the sample number) from just one sensor are stored in the FIFO. In these conditions, the number of samples that can be stored in the FIFO buffer is 4095: when the FIFO buffer is completely filled, the FIFO_FULL bit of the FIFO_STATUS2 register is set high.

Figure 31. FIFO mode (STOP_ON_FTH=0)



8.2.3 Continuous mode

In Continuous mode, the FIFO continues filling. When the buffer is full, the FIFO index restarts from the beginning, and older data are replaced by the new data. The oldest values continue to be overwritten until a read operation frees FIFO slots. The host processor's reading speed is important in order to free slots faster than new data is made available. To stop this configuration, the Bypass mode must be selected.

Follow these steps for Continuous mode configuration (if accelerometer/gyroscope data-ready is used as FIFO trigger):

1. Choose the decimation factor for each sensor through the decimation bits in the FIFO_CTRL3 and FIFO_CTRL4 registers (see *Section 8.3* for details);
2. Choose the FIFO ODR through the ODR_FIFO_[3:0] bits in the FIFO_CTRL5 register;
3. Set the FIFO_MODE_[2:0] bits in the FIFO_CTRL5 register to 110b to enable FIFO Continuous mode.

When this mode is selected, the FIFO collects data continuously. The FIFO_STATUS1 and FIFO_STATUS2 registers are updated according to the number of samples stored.

When the next stored set of data will make the FIFO full, the FIFO_FULL bit of the FIFO_STATUS2 register is set to 1. The FIFO_OVER_RUN bit in the FIFO_STATUS2 register indicates when at least one sample has been overwritten to store the new data.

Data can be retrieved after the FIFO_FULL event, by reading the FIFO_DATA_OUT_L and FIFO_DATA_OUT_H registers for a number of times specified by the DIFF_FIFO_[11:0] bits in the FIFO_STATUS1 and FIFO_STATUS2 registers.

Using the FTH bit of the FIFO_STATUS2 register, data can also be retrieved when a threshold level (FTH_[11:0] in FIFO_CTRL1 and FIFO_CTRL2 registers) is reached.

If the STOP_ON_FTH bit of CTRL4_C register is set to 1, the FIFO size is limited to the value of the FTH_[11:0] bits in the FIFO_CTRL1 and FIFO_CTRL2 registers: in this case, the FIFO_FULL bit of the FIFO_STATUS2 register is set high when the number of samples in FIFO will reach the FTH_[11:0] value on the next FIFO write operation.

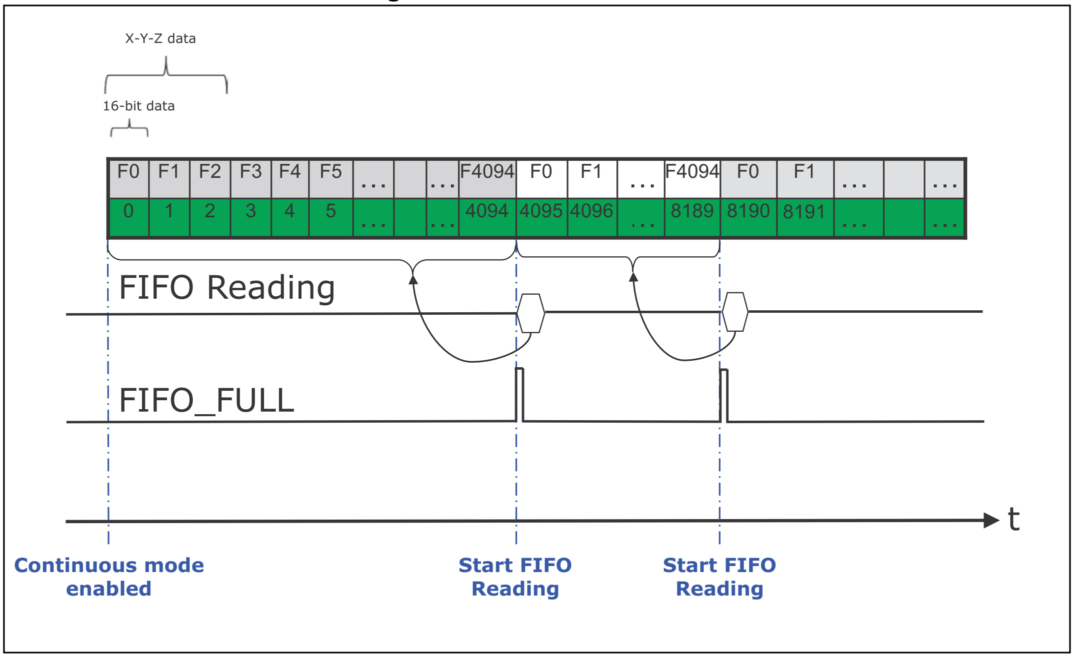
In case the FIFO is read before the FIFO_FULL event, it must not be completely emptied in order to avoid the misalignment of the data read from it. At least one complete FIFO pattern has to be left in the FIFO buffer (do not read it); it will be the first data read in the next read operation.

In case FIFO gets emptied after the FIFO_OUT registers are read, a FIFO reset (through Bypass mode setting) is needed.

It is recommended to read faster than 1*ODR at least three times the number of the enabled FIFO data set, in order to free FIFO slots for the new data: this allows avoiding loss of data.

Figure 32 shows an example of the Continuous mode usage. In the example, X-Y-Z data (green cells indicate the sample number) from just one sensor are stored in the FIFO and the FIFO samples are read faster than 1 * ODR, so that no data is lost. In these conditions, the number of samples stored is 4095.

Figure 32. Continuous mode



天猫店铺网址:https://telesky.tmall.com/

8.2.4 Continuous-to-FIFO mode

This mode is a combination of the Continuous and FIFO modes previously described. In Continuous-to-FIFO mode, the FIFO buffer starts operating in Continuous mode and switches to FIFO mode when an event condition occurs.

The event condition can be one of the following:

- Significant Motion: event detection has to be configured and the INT1_SIG_MOT bit of the INT1_CTRL register has to be set to 1;
- Tilt: event detection has to be configured and the INT2_TILT bit of the MD2_CFG register has to be set to 1;
- Step detection: event detection has to be configured and the INT1_STEP_DETECTOR bit of the INT1_CTRL register has to be set to 1;
- Single tap: event detection has to be configured and the INT2_SINGLE_TAP bit of the MD2_CFG register has to be set to 1;
- Double tap: event detection has to be configured and the INT2_DOUBLE_TAP bit of the MD2_CFG register has to be set to 1;
- Free-fall: event detection has to be configured and the INT2_FF bit of the MD2_CFG register has to be set to 1;
- Wake-up: event detection has to be configured and the INT2_WU bit of the MD2_CFG register has to be set to 1;
- 6D: event detection has to be configured and the INT2_6D bit of the MD2_CFG register has to be set to 1.

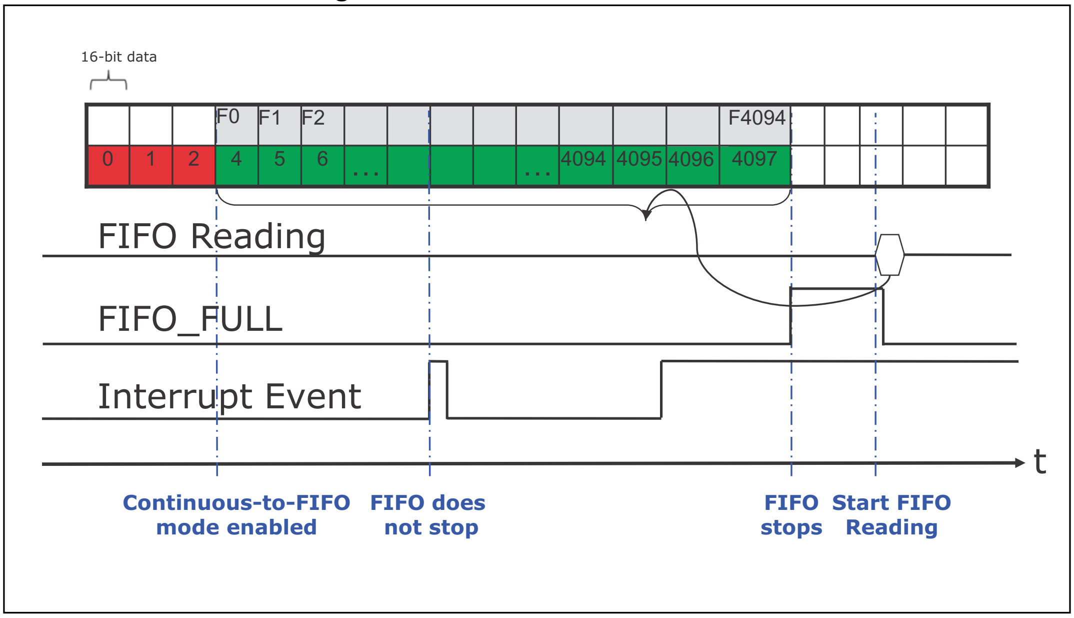
Continuous-to-FIFO mode is sensitive to the level of the interrupt signal and not to the edge, which means that if Continuous-to-FIFO is in FIFO mode and the interrupt condition disappears, the FIFO buffer returns to Continuous mode. It is recommended to latch the interrupt signal used as the FIFO event in order to avoid losing interrupt events (the interrupt signal has to be driven to the interrupt pin so that the latch function takes effect).

In case the FIFO is read before the FIFO_FULL event, it must not be completely emptied in order to avoid the misalignment of the data read from it. At least one complete FIFO pattern has to be left in the FIFO buffer (do not read it); it will be the first data read in next read operation.

In case FIFO gets emptied after FIFO_OUT registers are read, a FIFO reset (through Bypass mode setting) is needed.

天猫店铺网址:https://telesky.tmall.com/

Figure 33. Continuous-to-FIFO mode



Follow these steps for Continuous-to-FIFO mode configuration (if the accelerometer/gyroscope data-ready is used as the FIFO trigger):

1. Configure one of the events as previously described;
2. Choose the decimation factor for each sensor through the decimation bits in the FIFO_CTRL3 and FIFO_CTRL4 registers (see *Section 8.3* for details);
3. Choose the FIFO ODR through the ODR_FIFO_[3:0] bits in the FIFO_CTRL5 register;
4. Set the FIFO_MODE_[2:0] bits in the FIFO_CTRL5 register to 011b to enable FIFO Continuous-to-FIFO mode.

In Continuous-to-FIFO mode the FIFO buffer continues filling; when the next stored set of data will make the FIFO full, the FIFO_FULL bit is set high.

If the STOP_ON_FTH bit of the CTRL4_C register is set to 1, the FIFO size is limited to the value of the FTH_[11:0] bits in the FIFO_CTRL1 and FIFO_CTRL2 registers: in this case, the FIFO_FULL bit of the FIFO_STATUS2 register is set high when the number of samples in FIFO will reach or exceed the FTH_[11:0] value on the next FIFO write operation.

When the trigger event occurs, two different cases can be observed:

1. If the FIFO buffer is already full (FIFO_FULL = 1), it stops collecting data at the first sample after the event trigger. The FIFO content is composed of the samples collected before the event.
2. If FIFO buffer is not full yet (initial transient), it continues filling until it becomes full (FIFO_FULL = 1) and then, if the trigger is still present, it stops collecting data.

Continuous-to-FIFO can be used in order to analyze the history of the samples which have generated an interrupt; the standard operation is to read the FIFO content when the FIFO mode is triggered and the FIFO buffer is full and stopped.

8.2.5 Bypass-to-Continuous mode

This mode is a combination of the Bypass and Continuous modes previously described. In Bypass-to-Continuous mode, the FIFO buffer starts operating in Bypass mode and switches to Continuous mode when a trigger condition occurs.

The event condition can be one of the following:

- Significant Motion: event detection has to be configured and the INT1_SIG_MOT bit of the INT1_CTRL register has to be set to 1;
- Tilt: event detection has to be configured and the INT2_TILT bit of the MD2_CFG register has to be set to 1;
- Step detection: event detection has to be configured and the INT1_STEP_DETECTOR bit of the INT1_CTRL register has to be set to 1;
- Single tap: event detection has to be configured and the INT2_SINGLE_TAP bit of MD2_CFG register has to be set to 1;
- Double tap: event detection has to be configured and the INT2_DOUBLE_TAP bit of the MD2_CFG register has to be set to 1;
- Free-fall: event detection has to be configured and the INT2_FF bit of the MD2_CFG register has to be set to 1;
- Wake-up: event detection has to be configured and the INT2_WU bit of the MD2_CFG register has to be set to 1;
- 6D: event detection has to be configured and the INT2_6D bit of the MD2_CFG register has to be set to 1.

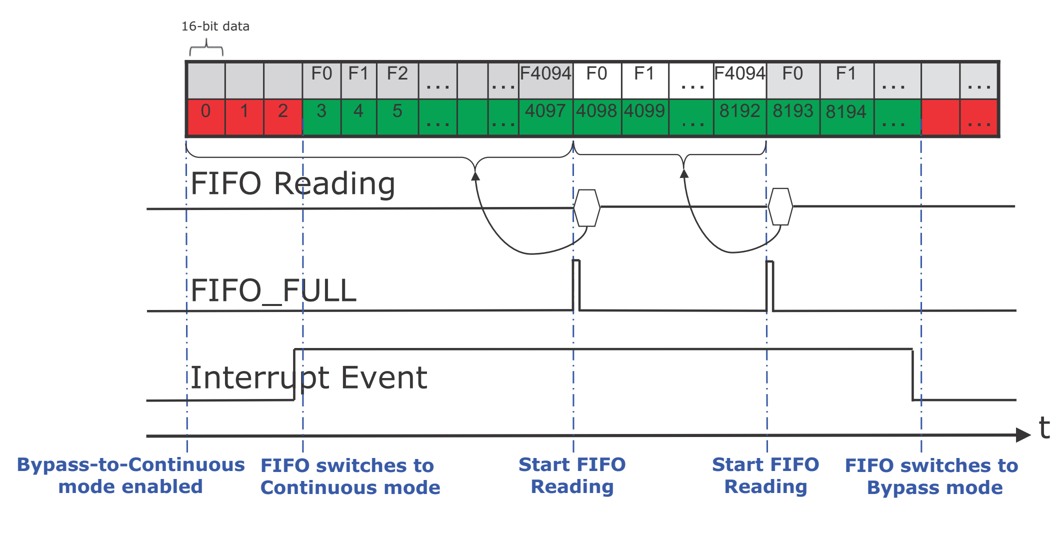
Bypass-to-Continuous mode is sensitive to the level of the interrupt signal and not to the edge, which means that if Bypass-to-Continuous is in Continuous mode and the interrupt condition disappears, the FIFO buffer returns to Bypass mode. It is recommended to latch the interrupt signal used as the FIFO event in order to avoid losing data (the interrupt signal has to be driven to the interrupt pin so that the latch function takes effect).

Follow these steps for Bypass-to-Continuous mode configuration (if the accelerometer / gyroscope data-ready is used as the FIFO trigger):

1. Configure one of the events as previously described;
2. Choose the decimation factor for each sensor through the decimation bits in the FIFO_CTRL3 and FIFO_CTRL4 registers (see *Section 8.3* for details);
3. Choose the FIFO ODR through the ODR_FIFO_[3:0] bits in the FIFO_CTRL5 register.
4. Set the FIFO_MODE_[2:0] bits in the FIFO_CTRL5 register to 100b to enable FIFO Bypass-to-Continuous mode.

天猫店铺网址:https://telesky.tmall.com/

Figure 34. Bypass-to-Continuous mode



Once the trigger condition appears and the buffer switches to Continuous mode, the FIFO buffer continues filling. When the next stored set of data will make the FIFO full, the FIFO_FULL bit is set high.

In case the FIFO is read before the FIFO_FULL event, it must not be completely emptied in order to avoid the misalignment of the data read from it. At least one complete FIFO pattern has to be left in the FIFO buffer (do not read it); it will be the first data read in the next read operation.

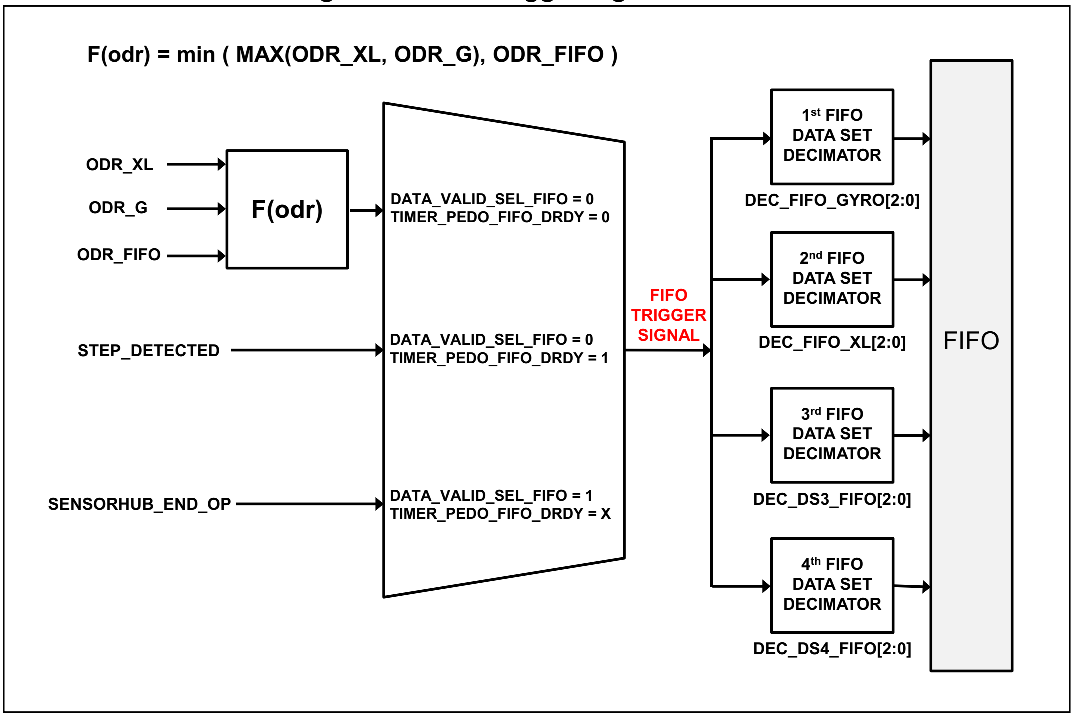
In case FIFO gets emptied after the FIFO_OUT registers are read, a FIFO reset (through Bypass mode setting) is needed.

Bypass-to-Continuous can be used in order to start the acquisition when the configured interrupt is generated.

8.3 Setting the FIFO trigger, FIFO ODR and decimation factors

Writing data in the FIFO can be configured to be triggered by three different sources.

Figure 35. FIFO trigger signal selection



As described in *Figure 35*, the DATA_VALID_SEL_FIFO bit of the MASTER_CONFIG register and the TIMER_PEDO_FIFO_DRDY bit of the FIFO_CTRL2 register are used for this purpose:

- if both the DATA_VALID_SEL_FIFO bit and the TIMER_PEDO_FIFO_DRDY bit are set to 0, writing data in the FIFO is triggered by the accelerometer/gyroscope data-ready. The ODR_FIFO_[3:0] bits of FIFO_CTRL5 define the maximum data rate at which data are stored in FIFO; the latter is limited to the maximum value between the accelerometer ODR (defined by the ODR_XL[3:0] bits of the CTRL1_XL register) and the gyroscope ODR (defined by the ODR_G[3:0] bits of the CTRL2_G register);
- if the DATA_VALID_SEL_FIFO bit is set to 0 and the TIMER_PEDO_FIFO_DRDY bit is set to 1, writing data in the FIFO is triggered by step detection (corresponding to the behavior of the STEP_DETECTED bit of the FUNC_SRC register): the data are stored in FIFO every time a step is detected;
- if the DATA_VALID_SEL_FIFO bit is set to 1, writing data in the FIFO is triggered by the sensor hub (corresponding to the behavior of the SENSORHUB_END_OP bit of the FUNC_SRC register), regardless of the configuration of the TIMER_PEDO_FIFO_DRDY bit: the data are stored in FIFO when the sensor hub routine is complete.

Using the FIFO decimation factors, data can be stored in FIFO at a rate lower than the rate of the FIFO trigger signal. Four decimation factors can be configured, one for each FIFO data set:

- the DEC_FIFO_G[2:0] bits of the FIFO_CTRL3 register define if the gyroscope data (associated to the 1st FIFO data set) are stored in FIFO and the relative rate;
- the DEC_FIFO_XL[2:0] bits of the FIFO_CTRL3 register define if the accelerometer data (associated to the 2nd FIFO data set) are stored in FIFO and the relative rate;
- the DEC_DS3_FIFO[2:0] bits of the FIFO_CTRL4 register define if the data associated to the 3rd FIFO data set are stored in FIFO and the relative rate;
- the DEC_DS4_FIFO[2:0] bits of the FIFO_CTRL4 register define if the data associated to the 4th FIFO data set are stored in FIFO and the relative rate.

Note: It's required to set at least one of the four decimation factors to 1 (no decimation).

When using the internal trigger (accelerometer/gyroscope data-ready), the recommended procedure to configure the FIFO trigger is the following:

1. Set the ODR_FIFO bits of the FIFO_CTRL5 register to the value corresponding to the maximum ODR between the gyroscope and accelerometer;
2. Set to 1 the decimation factor of the FIFO data set associated to the sensor having the maximum ODR.

8.3.1 Procedure for ODR changes when using FIFO

In combo mode configuration and if the application under development expects to store the data of at least one sensor (accelerometer, gyroscope or both) in the FIFO buffer and to modify the accelerometer/gyroscope output data rate (including Power-Down), the following rules have to be respected:

1. Both the accelerometer data and the gyroscope data must be stored in the FIFO;
2. Set the ODR_FIFO bits of the FIFO_CTRL5 register to 1010b (FIFO ODR is set to 6.66 kHz);
3. Apply the following procedure when an accelerometer/gyroscope ODR change has to be performed:
   a) read all the data stored in the FIFO to empty it (see *Section 8.4* for details);
   b) set the FIFO in Bypass mode (set the FIFO_MODE bits of the FIFO_CTRL5 register to 000b);
   c) Set the target ODR for the accelerometer and gyroscope through the ODR_XL bits of the CTRL1_XL register and the ODR_G bits of the CTRL2_G register respectively;
   d) Set the gyroscope decimation factor in the DEC_FIFO_G[2:0] bits of the FIFO_CTRL3 register and the accelerometer decimation factor in the DEC_FIFO_XL[2:0] bits of the FIFO_CTRL3 register as follows:
   – Accelerometer decimation factor = max(ODR_XL[Hz], ODR_G[Hz])/ODR_XL[Hz]
   – Gyroscope decimation factor = max(ODR_XL[Hz], ODR_G[Hz])/ODR_G[Hz]

   See *Table 56* and *Table 57* for the values to be set in the DEC_FIFO_G[2:0] bits and the DEC_FIFO_XL[2:0] bits of FIFO_CTRL3. One of the two decimation factors will be equal to 1, as required.
   e) Set the desired FIFO operating mode (see *Section 8.3* for details).

If ODR_XL and ODR_G are always equal to each other and the ODR_FIFO setting is always constant, no need to follow the above rules.

8.4 Retrieving data from the FIFO

Note: When data are stored in the FIFO, the configuration must not be changed in order to be able to retrieve data correctly.

When FIFO is enabled and the mode is different from Bypass, reading the FIFO output registers (FIFO_DATA_OUT_L and FIFO_DATA_OUT_H) returns the oldest FIFO sample set. Whenever these registers are read, their content is moved to the SPI/I^2C output buffer. FIFO slots are ideally shifted up one level in order to release room for a new sample, and the FIFO output registers load the current oldest value stored in the FIFO buffer.

The recommended way to retrieve data from the FIFO is the following:

1. Read the FIFO_STATUS1 and FIFO_STATUS2 registers to check how many words (16-bit data) are stored in the FIFO. This information is contained in the DIFF_FIFO_[11:0] bits.
2. Read the FIFO_STATUS3 and FIFO_STATUS4 registers. The FIFO_PATTERN_[9:0] bits allows understanding which sensor and which couple of bytes is being read (see *Section 8.5* for more details).
3. Read the FIFO_DATA_OUT_L and FIFO_DATA_OUT_H registers to retrieve the oldest sample (16-bits format) in the FIFO. They are respectively the lower and the upper part of the oldest sample.

The entire FIFO content is retrieved by performing a certain number of read operations from the FIFO output registers until the buffer becomes empty (FIFO_EMPTY bit of FIFO_STATUS2 register is set high). Once the FIFO is empty, every other read operation returns the same value (the latest sample).

It is recommended to read faster than 1*ODR at least three times the number of the enabled FIFO data set, in order to free FIFO slots for the new data: this allows avoiding loss of data.

The rounding function (see *Section 4.6* for details) is automatically enabled when applying a multiple read operation to the FIFO output registers FIFO_DATA_OUT_L and FIFO_DATA_OUT_H.

8.5 FIFO pattern

Data are stored in the FIFO without any tag in order to maximize the number of samples stored. To understand which couple of data and which FIFO data set is going to be read, it is necessary to check the content of the FIFO_PATTERN_[9:0] bits in the FIFO_STATUS3 and FIFO_STATUS4 registers.

Data are written to the FIFO with a specific pattern (for example GyroX, GyroY, GyroZ, AccX, AccY, AccZ). This pattern changes depending on the ODRs and decimation factors assigned to the four FIFO data sets. The FIFO_PATTERN_[9:0] bits contain a number from 0 to the index of the last sample of the pattern, then the pattern is repeated in all FIFO content.

The first sequence of data stored in FIFO buffer contains the data of all the enabled FIFO data sets, from the first one to the fourth one. Then, data are repeated depending on the value of the decimation factor set for each FIFO data set.

The examples in the next sections explain how to use the information contained in the FIFO_PATTERN_[9:0] bits.

8.5.1 Example 1

Supposing the FIFO is storing data from the gyroscope and accelerometer at the same ODR:

- Gyroscope ODR = 104 Hz, Accelerometer ODR = 104 Hz.

If the internal trigger (accelerometer/gyroscope data-ready) is used, it's recommended to set the ODR_FIFO_[3:0] bits of the FIFO_CTRL5 register to 0100b in order to set the FIFO trigger ODR to 104 Hz.

Both the DEC_FIFO_GYRO[2:0] and the DEC_FIFO_XL[2:0] fields of the FIFO_CTRL3 register have to be set to 001b (no decimation).

The following data pattern is repeated every 6 samples (each sample is represented as 16-bit data):

- Gx Gy Gz XLx XLy XLz (Gyroscope and Accelerometer data)

The FIFO_PATTERN_[9:0] bits will contain a number from 0 to 5, as shown in *Table 71*.

Table 71. Example 1: FIFO_PATTERN_[9:0] bits and next reading

Time	FIFO_PATTERN_[9:0]	Next reading from FIFO output registers
t0	0	Gx
t0	1	Gy
t0	2	Gz
t0	3	XLx
t0	4	XLy
t0	5	XLz

8.5.2 Example 2

Supposing the FIFO is storing data from the gyroscope and accelerometer at different ODRs:

- Gyroscope ODR = 208 Hz, Accelerometer ODR = 104 Hz.

If the internal trigger (accelerometer/gyroscope data-ready) is used, it's recommended to set the ODR_FIFO_[3:0] bits of the FIFO_CTRL5 register to 0101b in order to set the FIFO trigger ODR to 208 Hz.

The DEC_FIFO_GYRO[2:0] field of theFIFO_CTRL3 register has to be set to 001b (no decimation applied to gyroscope data) and the DEC_FIFO_XL[2:0] field has to be set to 010b (decimation with factor 2 applied to accelerometer data).

Since the gyroscope ODR is twice the accelerometer ODR, the following data pattern is repeated every 9 samples (each sample is represented as 16-bit data):

- Gx Gy Gz XLx XLy XLz Gx Gy Gz

The FIFO_PATTERN_[9:0] bits will contain a number from 0 to 8, as shown in *Table 72*.

Table 72. Example 2: FIFO_PATTERN_[9:0] bits and next reading

Time	FIFO_PATTERN_[9:0]	Next reading from FIFO output registers
t0	0	Gx
t0	1	Gy
t0	2	Gz
t0	3	XLx
t0	4	XLy
t0	5	XLz
t1	6	Gx
t1	7	Gy
t1	8	Gz

8.5.3 Example 3

Supposing the FIFO is storing data from the gyroscope, accelerometer and magnetometer at different ODRs:

Gyroscope ODR = 104 Hz, Accelerometer ODR = 208 Hz, Magnetometer ODR = 52 Hz.

If the internal trigger (accelerometer/gyroscope data-ready) is used, it’s recommended to set the ODR_FIFO_[3:0] bits of the FIFO_CTRL5 register to 0101b in order to set the FIFO trigger ODR to 208 Hz.

The DEC_FIFO_GYRO[2:0] field of FIFO_CTRL3 register has to be set to 010b (decimation with factor 2 applied to gyroscope data) and the DEC_FIFO_XL[2:0] field has to be set to 001b (no decimation applied to accelerometer data). Assuming that the magnetometer is associated to the 3rd FIFO data set, the DEC_DS3_FIFO[2:0] field of theFIFO_CTRL4 register has to be set to 100b (decimation with factor 4 applied to magnetometer data).

The following data pattern is repeated every 21 samples:

- Gx Gy Gz XLx XLy XLz Mx My Mz (gyroscope, accelerometer, magnetometer data - 9 samples)
- XLx XLy XLz (accelerometer data - 3 samples)
- Gx Gy Gz XLx XLy XLz (gyroscope and accelerometer data - 6 samples)
- XLx XLy XLz (accelerometer data - 3 samples)

The FIFO_PATTERN_[9:0] bits will contain a number from 0 to 20, as shown in *Table 73*.

Table 73. Example 3: FIFO_PATTERN_[9:0] bits and next reading

Time	FIFO_PATTERN_[9:0]	Next reading from FIFO output registers
t0	0	Gx
t0	1	Gy
t0	2	Gz
t0	3	XLx
t0	4	XLy
t0	5	XLz
t0	6	Mx
t0	7	My
t0	8	Mz
t1	9	XLx
t1	10	XLy
t1	11	XLz
t2	12	Gx
t2	13	Gy
t2	14	Gz
t2	15	XLx
t2	16	XLy
t2	17	XLz
t3	18	XLx
t3	19	XLy
t3	20	XLz

8.6 FIFO threshold

The FIFO threshold is a functionality of the LSM6DS3 FIFO which can be used to check when the number of samples in the FIFO reaches a defined threshold level.

The bits FTH_[11:0] in the FIFO_CTRL1 and FIFO_CTRL2 registers contain the threshold level. The resolution of the FTH_[11:0] field is two bytes (1 LSB = 2 Bytes, each sample is represented as 16-bit data). So, the user can select the desired level in a range between 0 and 4095.

The bit FTH in the FIFO_STATUS2 register represents the watermark status. This bit is set high if the number of samples in the FIFO reaches or exceeds the watermark level (each sample is represented as 16-bit data).

FIFO size can be limited to the threshold level by setting the STOP_ON_FTH bit in the CTRL4_C register to 1.

Figure 36. FIFO threshold (STOP_ON_FTH = 0)

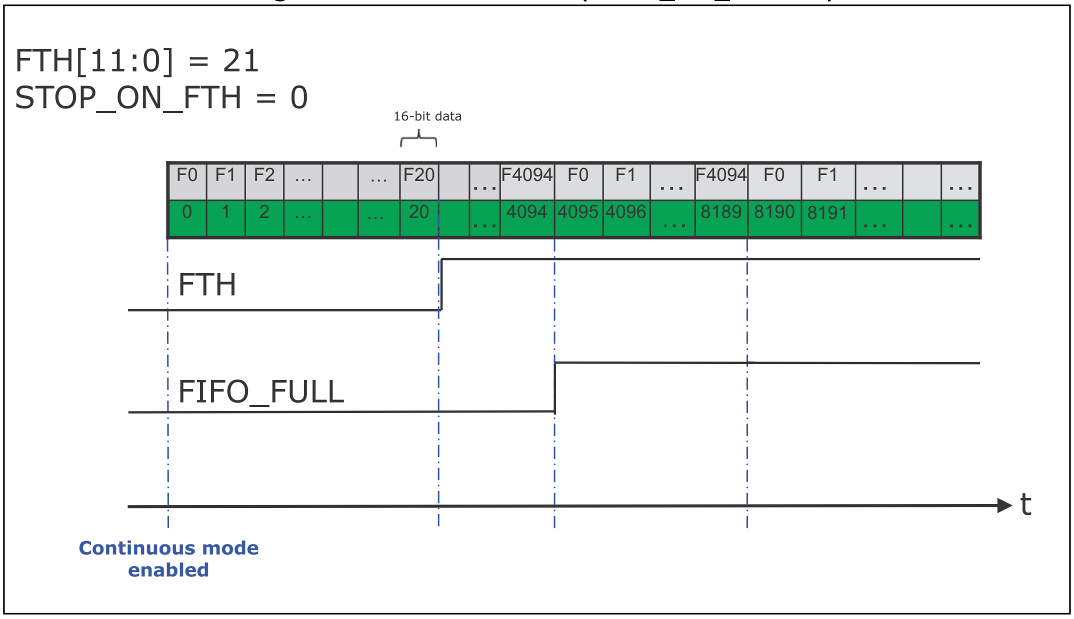


Figure 36 shows an example of FIFO threshold level usage when just accelerometer (or gyroscope) data are stored. The STOP_ON_FTH bit set to 0 in the CTRL4_C register. The threshold level is set to 21 through the FTH[11:0] bits. The FTH bit of the FIFO_STATUS2 register rises after the level 21 has been reached (21 samples in the FIFO). Since, the STOP_ON_FTH bit is set to 0, the FIFO will not stop at the 21st sample, but will keep storing data until the FIFO_FULL flag is set high.

Figure 37. FIFO threshold (STOP_ON_FTH = 1) in FIFO mode

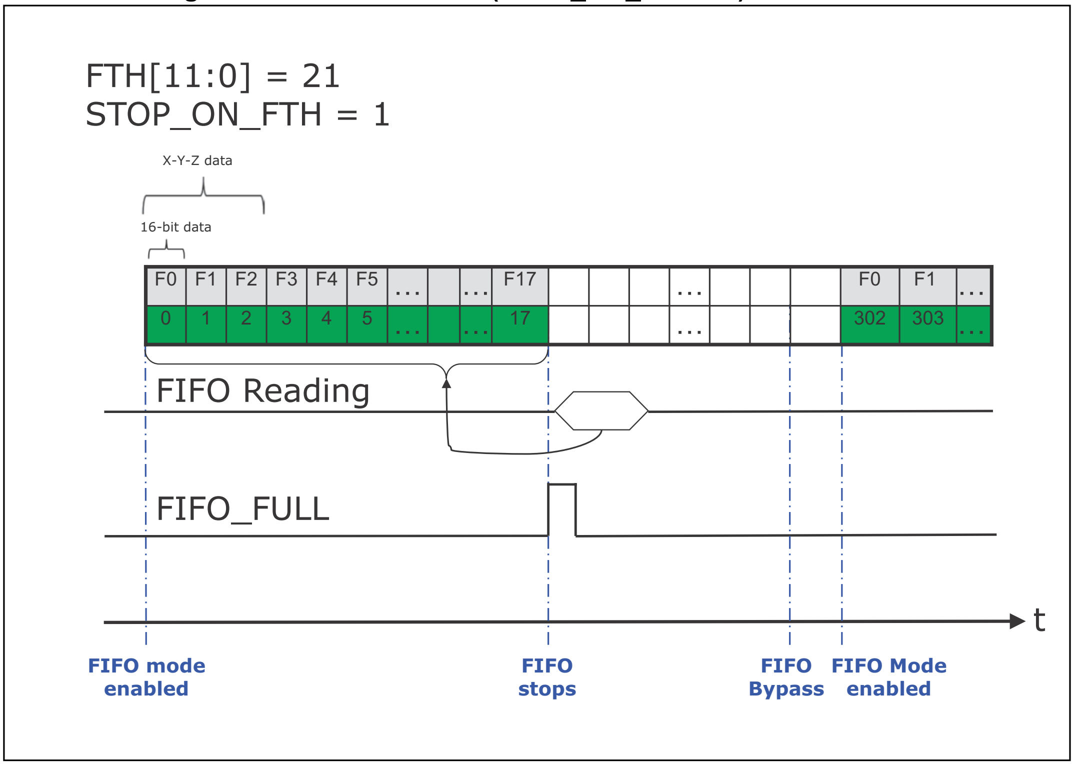


Figure 37 shows an example of FIFO threshold level usage in FIFO mode with the STOP_ON_FTH bit set to 1 in the CTRL4_C register; just accelerometer (or gyroscope) data are stored in this example. The threshold level is set to 21 through the FTH[11:0] bits and defines the current FIFO size. In FIFO mode, data are stored in the FIFO buffer until the FIFO_FULL signal rises; the FIFO_FULL bit of the FIFO_STATUS2 register rises when the next data stored in the FIFO will make the FIFO full, so in this example it rises after the first 18 data (16-bit each) are stored in FIFO. The FTH bit of the FIFO_STATUS2 register cannot go to 1 since the FTH threshold level is never reached (data are no longer stored in FIFO after the FIFO is full).

Figure 38. FIFO threshold (STOP_ON_FTH = 1) in Continuous mode

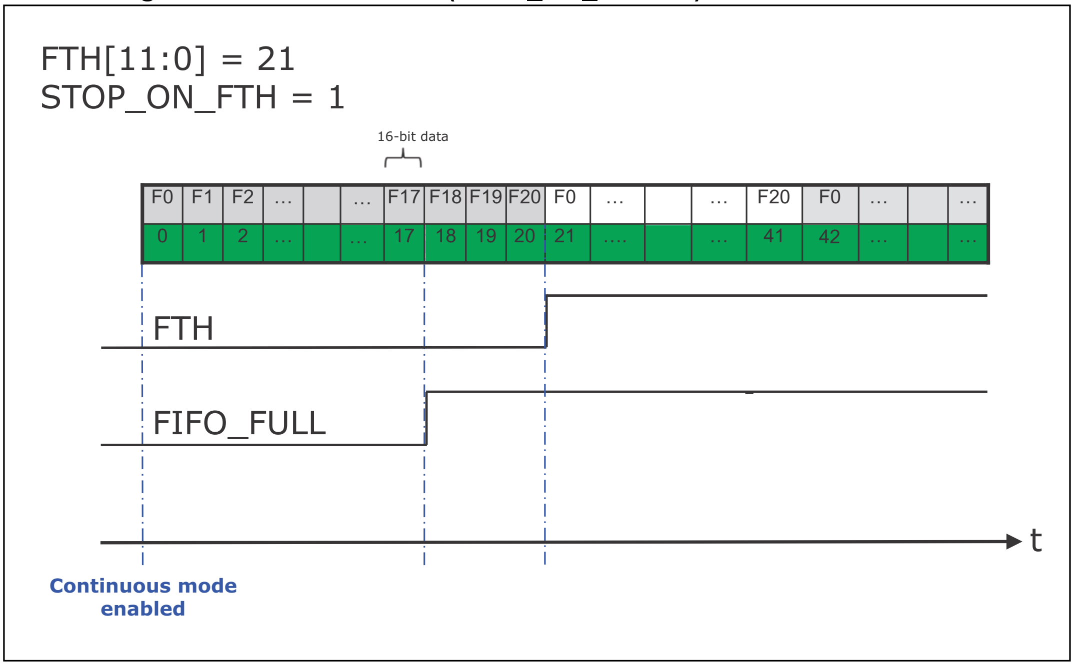


Figure 38 shows an example of FIFO threshold level usage in Continuous mode with the STOP_ON_FTH bit set to 1 in the CTRL4_C register; just accelerometer (or gyroscope) data are stored in this example. The threshold level is set to 21 through the FTH[11:0] bits. The FIFO_FULL bit of the FIFO_STATUS2 register rises when the next data stored in the FIFO will make the FIFO full, so in this example it rises after the first 18 data (16-bit each) are stored in FIFO. The FTH bit of the FIFO_STATUS2 register rises after the level 21 has been reached (21 samples in the FIFO).

8.7 High part of gyroscope and accelerometer data

It is possible to increase the number of samples stored in the FIFO by storing just the high part (8 bits) of gyroscope and accelerometer data. This feature is not valid for the other (external) sensors.

To the enable this feature, the bit ONLY_HIGH_DATA must be set to 1 in the FIFO_CTRL4 register. Gyroscope and accelerometer data will be written in the FIFO at the same ODR, in the order shown in *Table 74*.

Table 74. High part of gyroscope and accelerometer data in FIFO

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

When this feature is enabled, the 6 bytes containing the high part (8 bits) of gyroscope and accelerometer data are associated to the 1st FIFO data set and the 2nd FIFO data set is not used.

The DEC_FIFO_G[2:0] field of the FIFO_CTRL3 register has to be set to a value different from 000b (1st FIFO data set stored in FIFO).

The DEC_FIFO_XL[2:0] field of FIFO_CTRL3 register has to be set to 000b (2nd FIFO data set not in FIFO).

8.8 Step counter and timestamp data in FIFO

It is possible to store timestamp and step counter data in the FIFO. These data are stored as a 4th FIFO data set in the 6-byte data format shown in *Table 75*.

- 3 bytes for the timestamp;
- 1 byte is not used;
- 2 bytes for the number of steps.

Table 75. Timestamp and pedometer data in FIFO

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

To enable this feature, the bit TIMER_PEDO_FIFO_EN must be set to 1 in the FIFO_CTRL2 register.

When this feature is enabled, the 6 bytes containing the timestamp and step counter data are associated to the 4th FIFO data set: the DEC_DS4_FIFO[2:0] field of FIFO_CTRL4 register has to be used to define the decimation factor.

When this feature is enabled and the DATA_VALID_SEL_FIFO bit of the MASTER_CONFIG register is set to 0, data can be stored in the FIFO in two ways, depending on the configuration of the TIMER_PEDO_FIFO_DRDY bit in FIFO_CTRL2:

- When the TIMER_PEDO_FIFO_DRDY bit is set to 0, data are written to the FIFO at the ODR_FIFO rate set in the FIFO_CTRL5 register.
- When the TIMER_PEDO_FIFO_DRDY bit is set to 1, data are stored in the FIFO every time a new step is detected.

Follow these steps to store timestamp and pedometer data in the FIFO using either the internal trigger (accelerometer/gyroscope data ready) or the 'step detected' method:

1. Turn on the accelerometer;
2. Enable the timestamp and pedometer (see *Section 6.1* and *Section 6.4*);
3. Choose the decimation factor for the 4th FIFO data set through the DEC_DS4_FIFO[2:0] bits of the FIFO_CTRL4 register;
4. Set to 1 the TIMER_PEDO_FIFO_EN bit in the FIFO_CTRL2 register;
5. Configure the bit TIMER_PEDO_FIFO_DRDY in the FIFO_CTRL2 register, in order to choose the method of storing data in the FIFO (internal trigger or every step detected);
6. If an internal trigger is used, choose the FIFO ODR through the ODR_FIFO_[3:0] bits of the FIFO_CTRL5 register. If 'step detected' trigger is used, no need to set the ODR_FIFO_[3:0] bits;
7. Configure the FIFO operating mode through the FIFO_MODE_[2:0] field of the FIFO_CTRL5 register.

8.9 Temperature data in FIFO

It is possible to store only temperature data as the 4th FIFO data set.

To enable this feature:

- the bit TIMER_PEDO_FIFO_EN of the FIFO_CTRL2 register has to be set to 0;
- the bit FIFO_TEMP_EN of the CTRL4_C register has to be set to 1.

Temperature samples (16-bit) are stored in FIFO in the 6-byte data format shown in *Table 76*.

Table 76. Temperature data in FIFO

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

Follow these steps to store 16-bit temperature data in the FIFO using the internal trigger (accelerometer/gyroscope data ready):

1. Turn on the accelerometer or the gyroscope;
2. Choose the decimation factor (different from 000b) for the 4th FIFO data set through the DEC_DS4_FIFO[2:0] bits in the FIFO_CTRL4 register;
3. Set to 1 the FIFO_TEMP_EN bit in the CTRL4_C register and to 0 the bit TIMER_PEDO_FIFO_EN of the FIFO_CTRL2 register;
4. Choose the FIFO ODR through the ODR_FIFO_[3:0] bits of the FIFO_CTRL5 register;
5. Configure the FIFO operating mode through the FIFO_MODE_[2:0] field of the FIFO_CTRL5 register.

天猫店铺网址:https://telesky.tmall.com/

9 Temperature sensor

The LSM6DS3 is provided with an internal temperature sensor that is suitable for ambient temperature measurement.

If both the accelerometer and the gyroscope sensors are in Power-Down mode, the temperature sensor is off.

The maximum output data rate of temperature sensor is 52 Hz and its value depends on how the accelerometer and gyroscope sensors are configured:

- If the gyroscope is in Power-Down mode:
  - the temperature data rate is equal to 12.5 Hz if the accelerometer ODR is equal to 12.5 Hz (in both Low-Power and High-Performance mode);
  - the temperature data rate is equal to 26 Hz if the accelerometer configuration is 26 Hz Low-Power mode;
  - the temperature data rate is equal to 52H z for all other accelerometer configurations.
- If the accelerometer is in Power-Down mode:
  - the temperature data rate is equal to 12.5 Hz if the gyroscope configuration is 12.5 Hz Low-Power mode;
  - the temperature data rate is equal to 26 Hz if the gyroscope configuration is 26 Hz Low-Power mode;
  - the temperature data rate is equal to 52 Hz for all other gyroscope configurations.
- In combo mode:
  - if the gyroscope is configured in High-Performance mode, the temperature data rate is equal to 52 Hz regardless of the gyroscope ODR and the accelerometer configuration;
  - if the gyroscope is configured in Low-Power / Normal mode, the temperature data rate is equal to the maximum value between the accelerometer ODR and gyroscope ODR, while remaining below the 52 Hz value.

For the temperature sensor, the data-ready signal is represented by the TDA bit of the STATUS_REG register. The signal can be driven to the INT2 pin by setting to 1 the INT2_DRDY_TEMP bit of the INT2_CTRL register.

The temperature data is given by the concatenation of the OUT_TEMP_H and OUT_TEMP_L registers and it is represented as a number of 16 bits in two's complement format, with a sensitivity of +16 LSB/°C. The output zero level corresponds to 25 °C.

The LSM6DS3 allows swapping, by setting the BLE bit of the CTRL3_C register set to 1, the content of the lower and the upper part of the temperature output data registers (i.e. OUT_TEMP_H with OUT_TEMP_L).

Temperature sensor data can also be stored in FIFO with a configurable decimation factor (see *Section 8.9* for details).

9.1 Example of temperature data calculation

Table 77 provides a few basic examples of the data that is read in the temperature data registers at different ambient temperature values. The values listed in this table are given under the hypothesis of perfect device calibration (i.e. no offset, no gain error,....).

Table 77. Output data registers content vs. temperature

Temperature values	BLE = 0		BLE = 1	
	Register address			
	OUT_TEMP_H (21h)	OUT_TEMP_L (20h)	OUT_TEMP_H (21h)	OUT_TEMP_L (20h)
0°C	FEh	70h	70h	FEh
25°C	00h	00h	00h	00h
50°C	01h	90h	90h	01h

天猫店铺网址:https://telesky.tmall.com/

10 Self-test

The embedded self-test functions allows checking the device functionality without moving it.

10.1 Accelerometer self-test

When the accelerometer self-test is enabled, an actuation force is applied to the sensor, simulating a definite input acceleration. In this case the sensor outputs exhibit a change in their DC levels which are related to the selected full scale through the sensitivity value.

The accelerometer self-test function is off when the ST_XL[1:0] bits of the CTRL5_C register are programmed to 00b; it is enabled when the ST_XL bits are set to 01b (positive sign self-test) or 10b (negative sign self-test).

When the accelerometer self-test is activated, the sensor output level is given by the algebraic sum of the signals produced by the acceleration acting on the sensor and by the electrostatic test-force. The complete accelerometer self-test procedure is indicated in *Figure 39*.

10.2 Gyroscope self-test

The gyroscope self-test allows testing of the mechanical and electrical part of the gyroscope sensor: when it is activated, an actuation force is applied to the sensor, emulating a definite Coriolis force, and the seismic mass is moved by means of this electrostatic test-force. In this case the sensor output exhibits an output change.

The gyroscope self-test function is off when the ST_G[1:0] bits of the CTRL5_C register are programmed to 00b; it is enabled when the ST_G bits are set to 01b (positive sign self-test) or 11b (negative sign self-test).

When the gyroscope self-test is active, the sensor output level is given by the algebraic sum of the signals produced by the velocity acting on the sensor and by the electrostatic test-force. The complete gyroscope self-test procedure is indicated in *Figure 40*.

Figure 39. Accelerometer self-test procedure

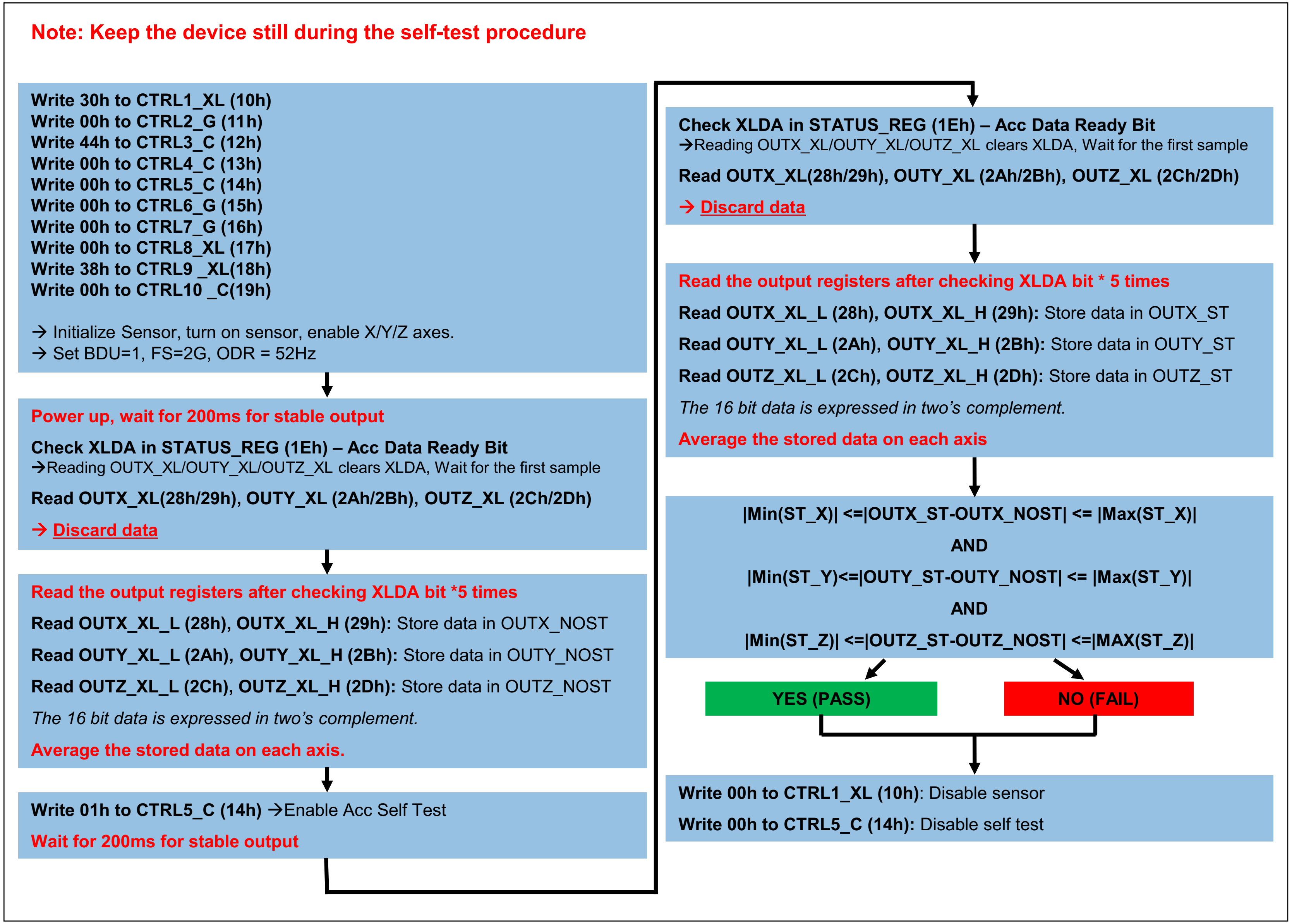
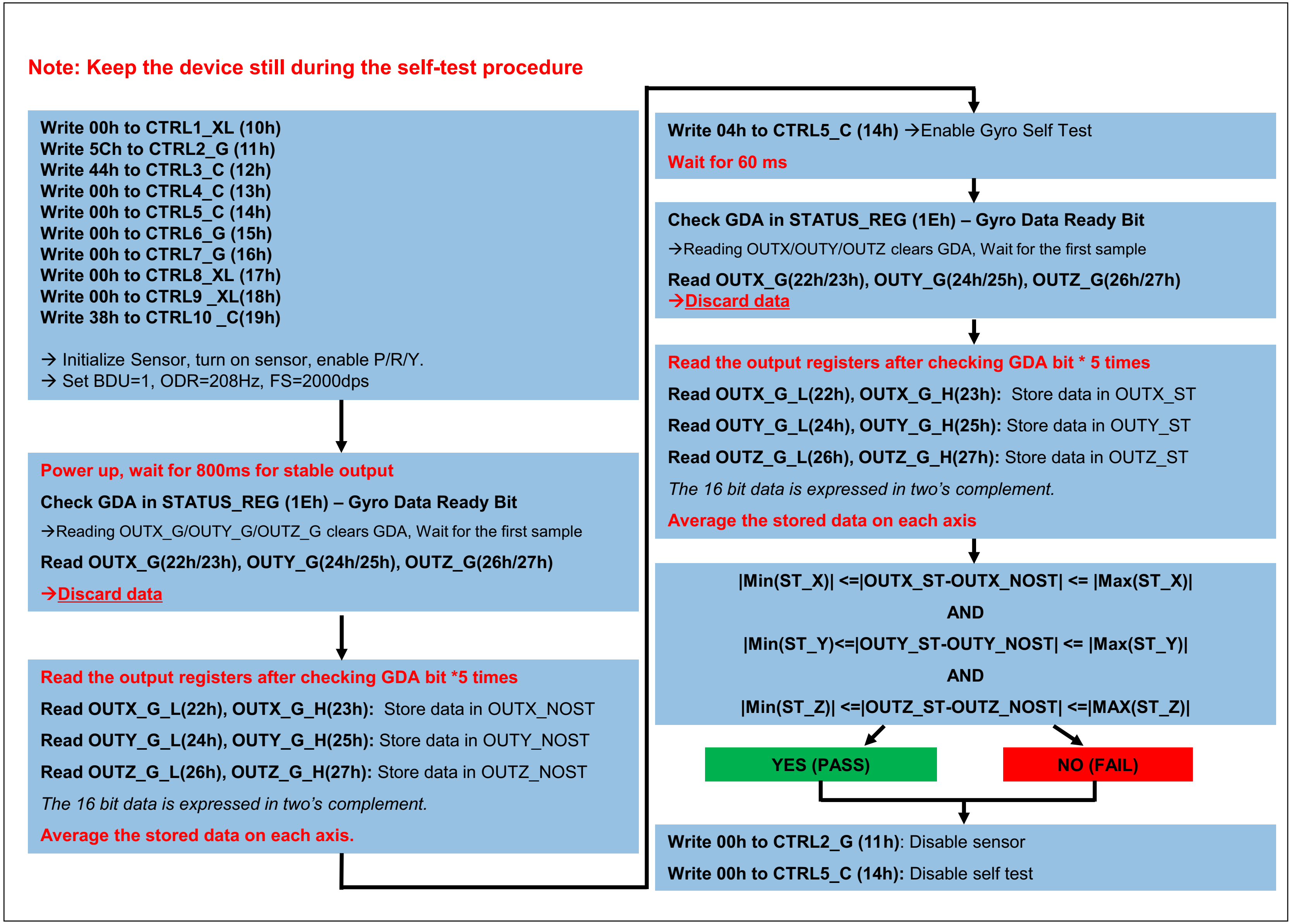


Figure 40. Gyroscope self-test procedure



11 Revision history

Table 78. Document revision history

Date	Revision	Changes
13-Feb-2015	1	Initial release.
10-Feb-2016	2	Updated *Introduction*, *Section 2.1: Embedded functions registers*, Added *Table 10: Accelerometer anti-aliasing + LPF1 overall cutoff frequency* Updated *Figure 5: Accelerometer composite digital filter* Added *Section 3.9: Accelerometer and gyroscope turn-on/off time* Updated *Figure 14: Gyroscope axes orientation and sign example* Updated *Section 5.3: Wake-up interrupt* Updated *Section 5.7: Boot status* Added *Figure 22: Pedometer debounce* and *Figure 23: Pedometer minimum threshold* Updated *Section 6.2: Significant motion* Added *Figure 24: Tilt example* Updated *Section 6.4: Timestamp* Added *Section 7.3: Sensor hub pass-through feature*, *Figure 27* Added *Section 8.3.1: Procedure for ODR changes when using FIFO* Updated *Table 74: High part of gyroscope and accelerometer data in FIFO*
14-Oct-2016	3	Updated *Introduction* Updated *Table 2: Registers* and *Table 3: Embedded functions registers* Added *Section 1: Pin description* Added *Section 3.6: Changing the power mode in accelerometer-only mode* Updated *Section 5.7: Boot status* Updated *Section 6.4: Timestamp* Updated *Section 8.3.1: Procedure for ODR changes when using FIFO* Minor textual updates
10-Feb-2017	4	Updated *Section 3.3: Normal mode* and *Section 3.4: Low-Power mode*