DT0077
Design tip

LIS3DSH State Machine quick guide

By Andrea Vitali

Main components	
LIS3DSH	MEMS digital output motion sensor: ultra low-power high performance 3-axes nano accelerometer

Purpose and benefits

This design tip is a quick guide to help you exploit the state machines embedded in the linear accelerometer LIS3DSH and provides the following:

- Quick configuration of the acquisition chain from analog-to-digital converter output to state machines' input
- Fundamentals of state machines programming; one example is discussed in detail: double-tap pattern detection
- State machine block diagram and features: basic blocks, shared blocks, peak detection, mask sets and configuration details.
- Quick reference: operation codes, registers overview.

Acquisition chain: from ADC (Analog-to-Digital) to SM (State Machine)

The LIS3DSH does embed two state machines. Figure 1 shows the acquisition chain from ADC to each State Machine. The chain is the following

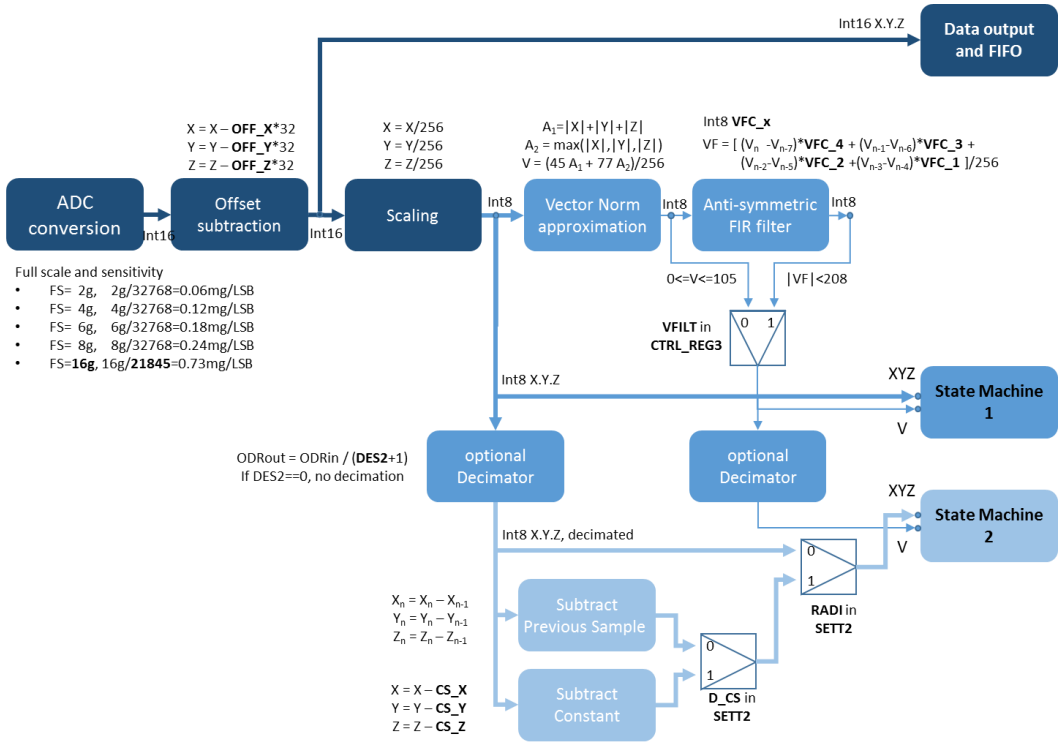
- ADC conversion: output is a signed 16bits integer (Int16, +/-32767); acceleration in mg is computed by multiplying the digital value by the sensitivity which depends on selected Full Scale (2, 4, 6, 8 or 16g).
- Offset subtraction: registers OFF_X (11h), OFF_Y (12h) and OFF_Z (13h) are multiplied by 32 and subtracted from data; offset-corrected data is then fed to output registers and the FIFO.
- Scaling: offset-corrected data is divided by 256; output is a signed 8-bit integer (Int8, +/-127); scaled data is fed to State Machine 1.
- Vector norm approximation: scaled data is used to compute an approximation of vector norm (0…+105); optionally the vector norm is filtered by an anti-symmetric

FIR filter with signed 8-bit coefficients VFC_1 (1Bh), VFC_2 (1Ch), VFC_3 (1Dh), VFC_4 (1Eh); the vector norm or the filter output is fed to both state machines.

- Scaled data and (filtered) vector norm is fed to State Machine 1; State Machine 2 is fed with the same data but decimated according to register DES2 (78h): ODR2 = ODR/(DES2+1)

- State Machine 2 has also the option to work on differential data: difference with respect to previous sample or difference with respect to constant shift registers CS_X (13h), CS_Y (14h), CS_Z (15h).

Figure 1. Acquisition chain: from ADC to State Machines



Fundamentals of State Machine programming

Each state machine can execute a 16-byte program. Program for State Machine 1 is stored in registers 40h-4Fh. Program for State Machine 2 is stored in registers 60h-6Fh. Program bytes are used as follows:

- State: 4-bit “RESET” condition code and 4bits “NEXT” condition code; conditions are checked when a new sample is available

- Command: 8-bit command followed by parameters (if any, 8 or 16-bit); commands are immediately executed, the state machine does not wait for a new sample; two exceptions: conditional jump JMP and synchronization SSYNC.

The state machine reads the byte. If it is a command, the state machine executes it immediately (unless there is a condition to check, as for conditional jumps). If it is a state,

the state machine waits for the data sample to be available. The sample is made of X, Y, Z triplet and the (filtered) vector norm approximation V. The state machine checks the “RESET” condition first and then the “NEXT” condition. Figure 2 shows the execution flow.

Figure 2. Execution flow: RESET and NEXT condition.

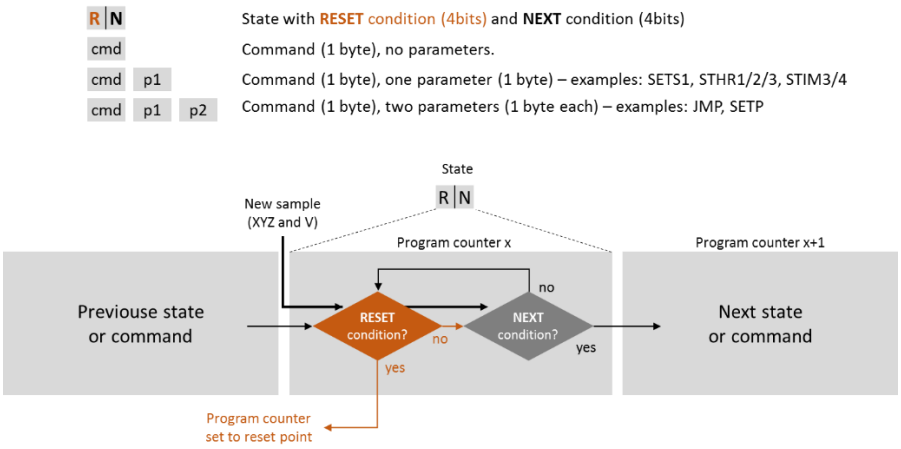


Figure 3 shows an example program for double-tap recognition. A tap is defined as a short duration peak of acceleration (10ms), which can be followed by some bouncing (20ms), and is concluded by a still period (70ms). Double-tap is two taps in 500ms.

Figure 3. Example program: double-tap recognition.

Register	Address	Value	Description
CTRL_REG4	20h	67h	ODR=100Hz (10msec per sample), no BDU, ZYX enabled
CTRL_REG3	23h	48h	Interrupt active high, INT1 pin enabled
CTRL_REG5	24h	00h	Bandwidth=800Hz, Full Scale=2g, No self test, 4-wire SPI
TIM4_1	50h	02h	2 x 10msec: 20msec (de-bounce time interval)
TIM3_1	51h	01h	1 x 10msec: 10msec (duration of peak acceleration caused by tapping)
TIM2_1_L	52h	32h	50 x 10msec: 500msec (time interval between two taps when double-tapping)
TIM1_1_L	54h	07h	7 x 10msec: 70msec (still time interval after tap and de-bounce)
THRS2_1	56h	55h	85 x 2g/127: 1.338g (acceleration threshold to identify tapping)
THRS1_1	57h	55h	85 x 2g/127: 1.338g (acceleration threshold to identify stillness)
MASK1_A	5Ah	FCh	Positive and negative XYZ axis enabled
SETT1	5Bh	A1h	Peak detection enabled, Comparisons use sign, STOP/CONT can generate interrupt
ST1_1	40h	51h	Reset **GNTH1**, Next **TI1** : peak axis must be below THRS1 for 70msec
ST1_2	41h	51h	Reset **GNTH1**, Next **TI1** : peak axis must be below THRS1 for 70msec
ST1_3	42h	06h	Reset **NOP**, Next **GNTH2** : wait for peak axis to be greater than THRS2
ST1_4	43h	38h	Reset **TI3**, Next **LNTH2** : peak axis must return below THRS2 in 10msec
ST1_5	44h	04h	Reset **NOP**, Next **TI4** : wait for 20msec (de-bounce)
ST1_6	45h	91h	Reset **GTTH1**, Next **TI1** : all axis must be below THRS1 for 70msec
ST1_7	46h	26h	Reset **TI2**, Next **GNTH2**: peak axis must go again above THRS2 within 500msec
ST1_8	47h	38h	Reset **TI3**, Next **LNTH2** : peak axis must return below THRS2 in 10msec
ST1_9	48h	04h	Reset **NOP**, Next **TI4** : wait for 20msec (de-bounce)
ST1_10	49h	91h	Reset **GTTH1**, Next **TI1** : all axis must be below THRS1 for 70msec
ST1_11	4Ah	11h	**CONT** command : generate interrupt and restart from beginning
CTRL_REG1	21h	01h	Hysteresis=0, state machine interrupt to INT1, **state machine enable (start)**

Configuration
- Peak detection is enabled but not essential
- Comparisons use sign to improve selectivity but absolute comparison can also be used

Still for 140msec
(TIM1 is used twice)

First tap
Defined by this pattern:
- Short peak (<10msec)
- Bouncing (20msec)
- Still period (>70msec)

Generate interrupt

Run program

State machines block diagram and features

Basic blocks

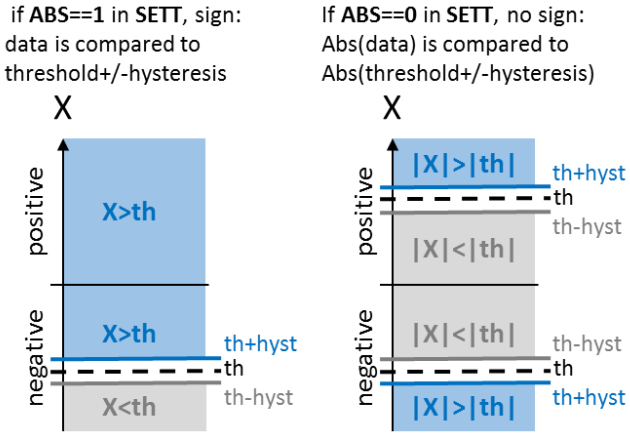
Input data is made by XYZ triplets and the (filtered) approximation of vector norm V. Each state machine has two thresholds (THRS1 and THRS2). An 8-bit mask (MASK or TMASK) enables/disables comparisons: positive X (first comparison), negative X, positive Y, negative Y, positive Z, negative Z, positive V and negative V (last comparison). The first comparison which is found to be true, is stored in the temporary mask (TMASK).

TMASK is copied to the output register OUTS when appropriate: OUTC/OUTW commands, STOP/CONT commands (if SITR bit asserted n SETT1/SETT2), LC long counter reaches 0. INT_SM1/INT_SM2 bit is also asserted in STAT register. OUTS is cleared when read, INT_SM is also cleared.

MASK usually has many bits set to 1. TMASK has only one bit set to 1 (unless REL command is used to copy MASK into TMASK). The user can select MASK or TMASK to control comparisons (R_TAM bit in SETT1/SETT2 register).

Sign may/may not be used for comparisons: ABS=1/0 in CTRL_REG1/CTRL_REG2; in the latter case the absolute threshold is compared to absolute data.

The threshold is modified by the hysteresis value, HYST in CTRL_REG1/CTRL_REG2. THRS+HYST is used for “greater than” comparison. THRS-HYST is used for “less than or equal” comparison. HYST goes from 0 to 7.



Each state machine has also a timer counter TC. The TIx command initialize the timer with the value stored in the corresponding TIMx register. The timer is then decremented when a new sample is available. If the timer reaches 0, the condition for the timer is true.

Shared blocks

Both state machines share the LC long counter register and the THRS3 register.

The DEC command decrements LC immediately. If the timer reaches 0, the temporary mask is copied to the output register OUTS. LONG and INT_SM1/INT_SM2 bits are asserted in STAT register.

If any of enabled absolute data exceeds absolute THRS3, then the state machine is restarted. THRS3_MA or THRS3_SA must be set in SETT1/SETT2 registers. The first selects MASKx_A, the second selects MASKx_B (see below for mask sets).

Peak detection

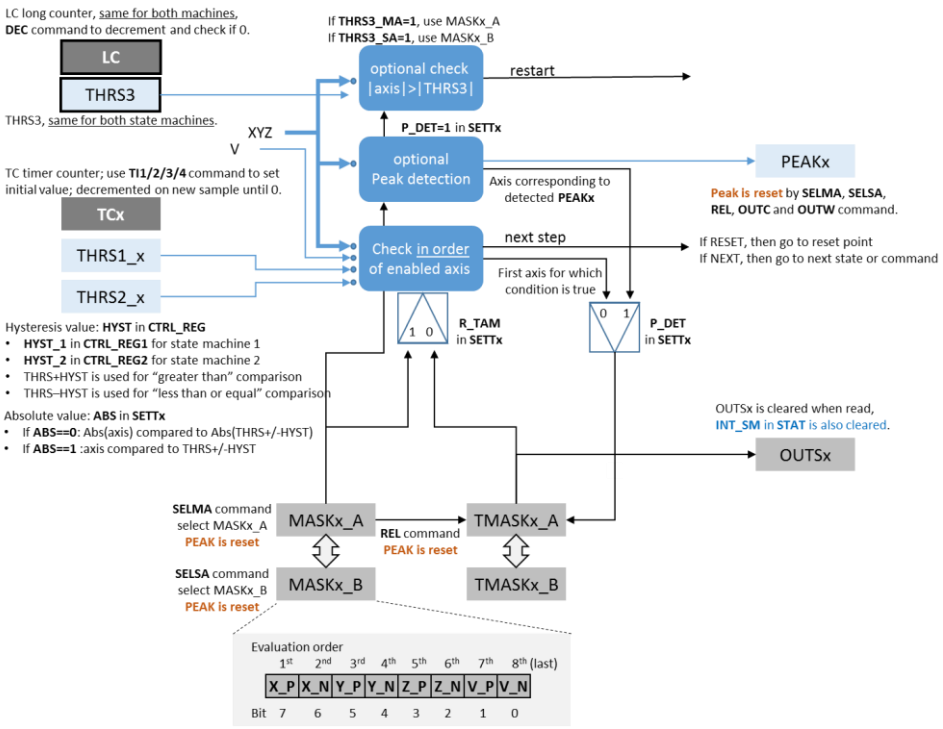
Peak detection is enabled by P_DET in SETT1/SETT2 registers. When enabled, the temporary mask holds the bit corresponding to the axis where the peak has been found. If the temporary mask is used to control comparisons (R_TAM=0 in SETT1/SETT2) then the comparison is done on the axis where peak has been found. The largest value is stored in the PEAK1/PEAK2 register.

Peak detection is reset when output is generated (OUTC/OUTW commands) or when masks are changed (REL, SELMA and SELSA command, see next paragraph).

Mask sets and configuration details

Each state machine has two mask sets: (T)MASKx_A, and (T)MASKx_B. Execution starts with MASKx_A by default. SELMA command selects set A. SELSA command selects set B. REL command copies MASK to TMASK. Note that peak detection is reset whenever one of these commands is executed. Figure 4 summarize the configuration details of the state machine.

Figure 4. State Machine configuration details.



Quick reference

Figure 5 summarizes 4-bit operation codes for NEXT and RESET conditions. Figure 6 and 7 summarize 8-bit operation codes for commands with their respective parameters (if any).

- RESET and NEXT conditions must be different. If they were the same, execution would be unpredictable: the state machine would go back and forth from RESET to NEXT while the condition is false; and when the condition becomes true, it could cause both a RESET or a transition to NEXT.

- RESET and NEXT condition cannot be TIx at the same time: if both RESET and NEXT condition were TIx, then the shortest delay would win and it would cause a reset or a transition to the next state/command.

Figure 8 summarizes the device registers. Figure 9 summarizes registers for both state machines.

Figure 5. Operation codes for RESET and NEXT condition in a state.

Opcode	Mnemonic	Action executed when new sample is available for the state RESET and NEXT condition must be different (commands opcode are XXh).
0h	NOP	No operation; condition is false: in RESET go to NEXT, in NEXT go back to RESET.
1h/2h/3h/4h	TI1/TI2/TI3/TI4	Wait for TIM1/2/3/4 samples: set counter TC, decrement at each sample, check if 0. RESET and NEXT condition cannot be both TIx (commands opcode are 1xh/2xh/3xh/4xh).
5h/6h	GNTH1/GNTH2	Check if any of enabled axis > threshold1/2 + hysteresis value.
7h/8h	LNTH1/LNTH2	Check if any of enabled axis <= threshold1/2 – hysteresis value.
9h	GTTH1	Check if any of enabled axis > thresold1 + hysteresis; **use mask of state machine 1**.
Ah	LLTH2	Check if **ALL** axis enabled <= threshold2 – hysteresis value.
Bh/Dh	GRTH1/GRTH2	Check if any of enabled axis > negated threshold1/2 + hysteresis value.
Ch/Eh	LRTH1/LRTH2	Check if any of enabled axis <= negated threshold1/2 + hysteresis value.
Fh	NZERO	Check if any of enabled axis changes sign (zero crossing detection).

Figure 6. Operation codes for commands.

Opcode	Mnemonic	Action executed immediately (does not wait for new sample) except where noted
00h	STOP	Stop execution (SM_EN=0 and SYNC=0 in SETT); if SITR=1 in SETT, copy temporary mask to output (OUTS) and assert interrupt (INT_SM in STAT)
11h	CONT	Continue from beginning (program pointer set to 0h in PR); if SITR=1 in SETT, copy temporary mask to output (OUTS) and assert interrupt (INT_SM in STAT)
22h XYh ZWh	JMP	Conditional jump: **wait for new sample** and evaluate condition; if condition Xh is true, then program pointer = Zh; else if condition Yh is true, then program pointer = Wh; else wait for another sample and repeat.
33h/44h	SRP/CRP	Set/clear the reset pointer: set to current program pointer + 1 (next state or command) / clear by setting it to 0
55h XYh ZWh	SETP	Set parameter: register XYh is set to ZWh; register must be writeable.
66h XYh	SETS1	Set register SETTx to XYh
77h/AAh/EEh XYh	STHR1/STHR2/STHR3	Set threshold1/2/3 to XYh
88h/99h	OUTC/OUTW	Output/output-and-wait: copy temporary mask to output (OUTS) and assert state interrupt (INT_SM in STAT); OUTW also assert SYNCW and wait for host to read output; OUTS, SYNCW and INT_SM are cleared upon read; PEAK is reset
BBh	DEC	Decrement and check if long counter LC is 0; if 0, copy temporary mask to output (OUTS) and assert interrupts (INT_SM and LONG in STAT); LC is reset to -1 (inactive).
CCh	SISW	Negate temporary mask: for each axis, if positive enabled, then enable negative and vice versa; no change if none or both enabled.
DDh	REL	Release temporary mask by setting it to default mask (A or B), PEAK is reset
FFh	SSYNC	Sync with other state machine: **wait for new sample**, SYNCx bit is asserted in STAT, the other state machine is executed until another SSYNC command is encountered. Both state machines must be enabled.

Opcode	Mnemonic	Action executed immediately but setting effective for next state or command
12h/13h	SABS0/SABS1	Set ABS=0 in SETTx (no sign, abs(data) compared to abs(th+/-hyst)) / ABS=1 (sign kept, data compared to th+/hyst)
14h/24h	SELMA/SELSA	Select mask A / mask B; PEAK is reset
21h/23h	SRADI0/SRADI1	Set RADI=0 in SETT2 (use raw data) / RADI=1 (use diff data); only state machine 2, halts state machine 1
31h/32h	SCS0/SCS1	Set D_CS=0 in SETT2 (diff with prev sample) / D_CS=1 (diff with CS_X/Y/Z), only state machine 2, halts state machine 1
34h/43h	SRTAM0/SRTAM1	Set R_TAM=0 in SETTx (preserve and use temporary mask) / R_TAM=1 (always use default mask)
41h/42h XYh	STIM3/STIM4	Set TIM3/TIM4 reload value to XYh

Figure 7. Common registers.

Address	R/W	Mnemonic	Register documentation
0Ch	R	OUT_T	Output temperature
0Dh/0Eh	R	INFO1/2	
11h/11h/12h	R W	OFF_X/Y/Z	Offset correction: axis = axis – offset*32, INT8
13h/14h/15h	**R W**	**CS_X/Y/Z**	Constant shift for state machine 2: axis/256 = axis/256 – constant, INT8
16h/17h	**R W**	**LC_L/H**	Long counter, decremented by **DEC** command, generate **LONG** interrupt when 0, -1 when inactive, UINT16
18h	**R**	**STAT**	Status of state machines: **LONG, SYNCW, SYNC1, SYNC2, INT_SM1, INT_SM2, DOR**, DRDY
1Bh/1Ch/1Dh/1Eh	**R W**	**VFC_1/2/3/4**	Filter coefficients for vector norm approximation V, enabled by **VFILT** in **CTRL_REG3**, INT8
1Fh	**R W**	**THRS3**	Threshold 3, reset to mask A/B if **THRS3_MA/SA==1** in **SETT** and abs(axis)>THRS3, INT8
20h	R W	CTRL_REG4	ODR (4bits, power down, 3.125Hz to 1.6kHz), BDU (lock OUT_L/H until both read), Z enable, Y enable, Z enable
23h	R W	**CTRL_REG3**	Data ready to INT1, interrupt polarity (0=low) and latch (0=latched), INT2 and INT1 enable, **VFILT enable**, Soft reset
24h	R W	CTRL_REG5	Bandwidth (2bits, 50/200/400/800Hz), Full scale (3bits, 2/4/6/8/16g), Self test (2bits), SPI mode (4/3wire)
25h	R W	CTRL_REG6	Boot, FIFO/watermark enable, auto SPI/I2C addr increment, FIFO empty/watermark/overrun on INT1, boot on INT2
27h	R	STATUS	Status of device: ZYX/Z/Y/X overrun, ZYX/Z/Y/X data available
28h/29h	R	OUT_X_L/H	Output X acceleration, INT16
2Ah/2Bh	R	OUT_Y_L/H	Output Y acceleration, INT16
2Ch/2Dh	R	OUT_Z_L/H	Output Z acceleration, INT16
2Eh	R W	FIFO_CTRL	FIFO setting: mode (3bits), watermark level (5bits)
2Fh	R	FIFO_SRC	FIFO status: watermark reached, overrun, empty, data level (5bits)

Status in **STAT**:
- Bit 7: **LONG**: long counter is 0
- Bit 6: **SYNCW**: **OUTW** executed, waiting for host to read output (OUTS1 or OUTS2)
- Bit 5: **SYNC1**: **SSYNC** executed by S.M.1, now in stand-by, waiting for SSYNC of S.M.2
- Bit 4: **SYNC2**: **SSYNC** executed by S.M.2, now in stand-by, waiting for SSYNC of S.M.1
- Bit 3: **INT_SM1**: interrupt status for state machine 1
- Bit 2: **INT_SM2**: interrupt status for state machine 2
- Bit 1: **DOR**: data overrun, unread output (OUTS1 or OUTS2) when new sample available
- Bit 0: DRDY: data ready, new sample available (OUT_X/Y/Z_L/H)

CTRL_REG4:
- ODR: 0000b = power down, 0001b = 3.125Hz, 0010b = 6.25Hz, 0011b = 12.5Hz, 0100b = 25Hz, 0101b = 50Hz, 0110b = 100Hz, 0111b = 400Hz, 1000b = 800Hz, 1001b = 1600Hz

CTRL_REG5:
- Bandwidth: 00b=800Hz 01b=400Hz, 10b=200Hz, 11b=50Hz
- Full Scale: 000b=2, 001b=4g, 010b=6g, 011b=8g, 100b=16g

Figure 8. State Machine registers.

State Machine **1**

Address	R/W	Mnemonic	Register documentation
19h	R	PEAK**1**	Peak value, INT8
21h	R W	CTRL_REG**1**	Hysteresis (3bits), INT1/2 pin select, **enable**
40h-4Fh	W	ST**1**_x	Program (state or command)
50h	W	TIM4_**1**	Timer reload value 4, 1LSB=1/ODR, UINT8
51h	W	TIM3_**1**	Timer reload value 3, 1LSB=1/ODR, UINT8
52h/53h	W	TIM2_**1** L/H	Timer reload value 2, 1LSB=1/ODR, UINT16
54h/55h	W	TIM1_**1** L/H	Timer reload value 1, 1LSB=1/ODR, UINT16
56h	W	THRS2_**1**	Threshold 2, 1LSB=FS/127, INT8
57h	W	THRS1_**1**	Threshold 1, 1LSB=FS/127, INT8
59h	W	MASK**1**_B	Mask B, see **SELSA** command
5Ah	W	MASK**1**_A	Mask A, see **SELMA** command
5Bh	W	SETT**1**	Setting (see below)
5Ch	R	PR**1**	Program pointer (4bits), reset point (4bits)
5Dh/5Eh	R	TC**1**	Timer counter, see **TI1/2/3/4** command
5Fh	R	OUTS**1**	Output

Setting in **SETT1**:
- **P_DET**: peak detection enable
- **THRS3_SA**: enable reset when abs(axis)>THRS3, mask B selected
- **ABS**: 0=no sign, compare abs(data) to abs(threshold+/-hyst); 1=sign
- 0
- 0
- **THRS3_MA**: enable reset when abs(axis)>THRS3, mask A selected
- **R_TAM**: 0=use temporary mask, 1=use mask
- **SITR**: enable output and interrupt generation for **STOP** and **CONT** command

CTRL_REG1:
- Bit 7-5: hysteresis value (+/-7, 1LSB=FS/127)
- Bit 3: 0=interrupt to INT1 pin, 1=interrupt to INT2 pin
- Bit 0: enable state machine (start program execution)

State Machine **2**

Address	R/W	Mnemonic	Register documentation
1Ah	R	PEAK**2**	Peak value, INT8
22h	R W	CTRL_REG**2**	Hysteresis (3bits), INT1/2 pin select, **enable**
60h-6Fh	W	ST**2**_x	Program (state or command)
70h	W	TIM4_**2**	Timer reload value 4, 1LSB=1/ODR, UINT8
71h	W	TIM3_**2**	Timer reload value 3, 1LSB=1/ODR, UINT8
72h/73h	W	TIM2_**2** L/H	Timer reload value 2, 1LSB=1/ODR, UINT16
74h/75h	W	TIM1_**2** L/H	Timer reload value 1, 1LSB=1/ODR, UINT16
76h	W	THRS2_**2**	Threshold 2, 1LSB=FS/127, INT8
77h	W	THRS1_**2**	Threshold 1, 1LSB=FS/127, INT8
78h	W	DES**2**	Decimation factor, UINT8
79h	W	MASK**2**_B	Mask B, see **SELSA** command
7Ah	W	MASK**2**_A	Mask A, see **SELMA** command
7Bh	W	SETT**2**	Setting (see below)
7Ch	R	PR**2**	Program pointer (4bits), reset point (4bits)
7Dh/7Eh	R	TC**2**	Timer counter, see **TI1/2/3/4** command
7Fh	R	OUTS**2**	Output

Setting in **SETT2**:
- **P_DET**: peak detection enable
- **THRS3_SA**: enable reset when abs(axis)>THRS3, mask B selected
- **ABS**: 0=no sign, compare abs(data) to abs(threshold+/-hyst); 1=sign
- **RADI**: 0=raw data, 1=diff data, see **D_CS**
- **D_CS**: 0=diff with previous sample, 1=diff with constant, see CS_X/Y/Z
- **THRS3_MA**: enable reset when abs(axis)>THRS3, mask A selected
- **R_TAM**: 0=use temporary mask, 1=use mask
- **SITR**: enable output and interrupt generation for **STOP** and **CONT**

CTRL_REG2:
- Bit 7-5: hysteresis value (+/-7, 1LSB=FS/127)
- Bit 3: 0=interrupt to INT1 pin, 1=interrupt to INT2 pin
- Bit 0: enable state machine (start program execution)

Support material

Related design support material
Product/ system Evaluation board – list references and links
Development kit – list reference and link
Evaluation tool, STEVAL-MKI109V2, eMotion: ST MEMS adapters motherboard based on STM32F103, compatible with all ST MEMS adapter boards
Evaluation tool, STEVAL-MKI134V1, LIS3DSH adapter board for standard DIL24 socket
Documentation
Datasheet, LIS3DSH, MEMS digital output motion sensor: ultra-low-power high-performance three-axis "nano" accelerometer
User manual, UM0979, STEVAL-MKI109V2 - eMotion motherboard for MEMS adapter boards.
User manual, UM1049, Unico GUI (paragraph 2.12).
Application note, AN3393, LIS3DSH: 3-axis digital output accelerometer

Revision history

Date	Version	Changes
3-Feb-2017	1	Initial release