

AN-012

mCube Application Note on SNIFF Mode for MC3600, 3-Axis Accelerometer Series

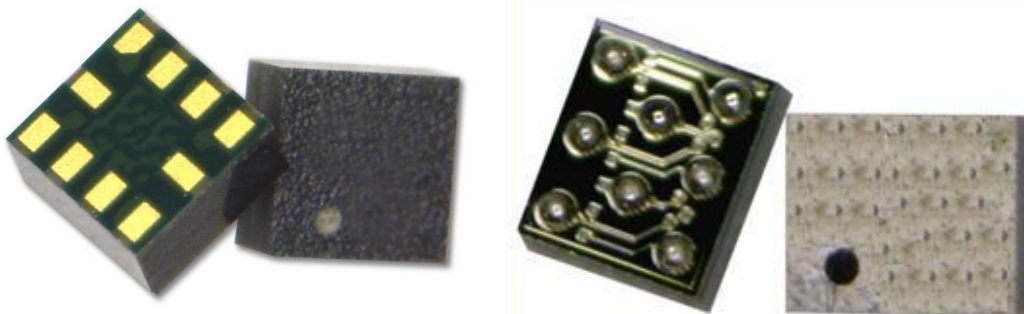
GENERAL DESCRIPTION

SNIFF is an ultra-low power mode for any activity detection which only consumes 400nA at 6Hz and 300nA at 1Hz; In this mode, the sniff circuitry and sniff-only sampling are enabled. There are no FIFO operations, and hardware will automatically transition to CWAKE mode upon activity detection.

MC3600 Series also support SLEEP mode (100nA) and three WAKE modes: Ultra Low Power Mode (900nA at 25Hz), Low Power Mode and Precision Mode.

FEATURES

SNIFF mode is ideal for being implemented as power switch. The rest of the parts could remain in sleep until certain motion exceeds the pre-set threshold in the SNIFF mode.



**Figure 1. The MC3635 (left) and the MC3672 (right),
World’s Smallest and Lowest Power 3-axis Accelerometers.**



TABLE OF CONTENTS

1	Mode State Machine Flow.....	3
2	Initialising Sequence	4
3	Formula for Delta g to Threshold	6
3.1	High Precision Mode	6
4	Sample Code	7
5	Revision History	8
6	Legal	9



1 MODE STATE MACHINE FLOW

The flow for SNIFF mode is demonstrated in the area marked in red in Figure 2.

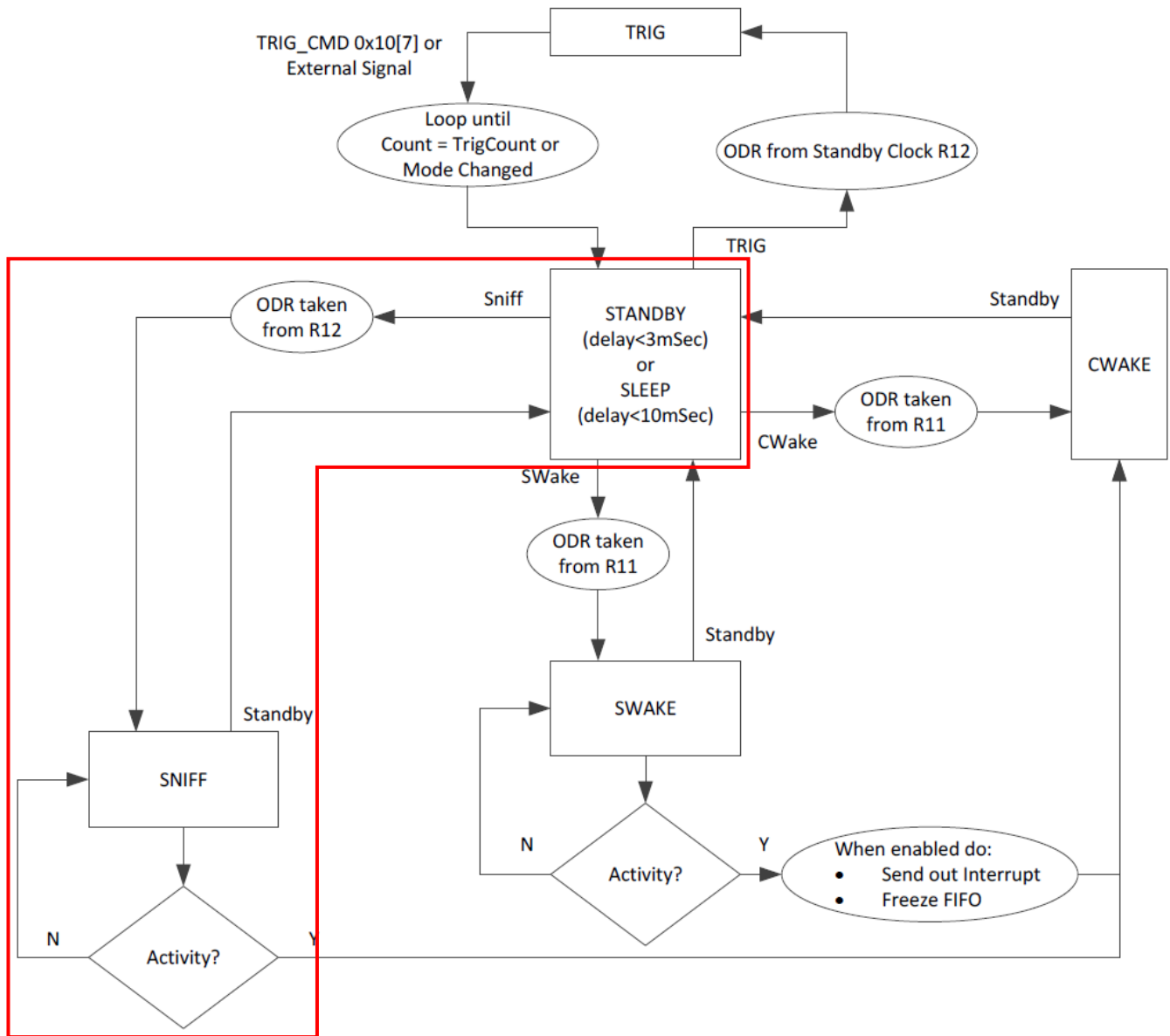


Figure 2. Mode Operational Flow.



2 INITIALISING SEQUENCE

To initialise the part for SNIFF mode, please follow the sequence below.

1. Switch to STANDBY mode
 - a. Initialise
Write 0x42 to the Initialisation Register 1 (0x0F).
 - b. Switch mode
Write 0b001 (STANDBY) to the Mode Control Register (0x10) [2:0].
2. Set SNIFF threshold
(Please refer to Section 3 “Formula for Delta g to Threshold” for threshold value)
 - a. Set threshold for X axis
Write 0b001 (Threshold, X-axis) to the SNIFF Config Register (0x14) [2:0].
Write the threshold (0 to 63, independent from Y and Z) to the SNIFF Threshold Control Register (0x13) [5:0].
 - b. Set threshold for Y axis
Write 0b010 (Threshold, Y-axis) to the SNIFF Config Register (0x14) [2:0].
Write the threshold (0 to 63, independent from X and Z) to the SNIFF Threshold Control Register (0x13) [5:0].
 - c. Set threshold for Z axis
Write 0b011 (Threshold, Z-axis) to the SNIFF Config Register (0x14) [2:0].
Write the threshold (0 to 63, independent from X and Y) to the SNIFF Threshold Control Register (0x13) [5:0].
3. Set SNIFF detect count
 - a. Set detect count for X axis
Write 0b101 (Detect Count, X-axis) to the SNIFF Config Register (0x14) [2:0].
Write count - 1 (0 to 61, indicating 1 to 62 counts, independent from Y and Z) to the SNIFF Threshold Ctrl Register (0x13) [5:0], e.g. for 1 event, write 0b00000.
 - b. Set detect count for Y axis
Write 0b110 (Detect Count, Y-axis) to the SNIFF Config Register (0x14) [2:0].
Write count - 1 (0 to 61, indicating 1 to 62 counts, independent from X and Z) to the SNIFF Threshold Ctrl Register (0x13) [5:0], e.g. for 2 events, write 0b00001.
 - c. Set detect count for Z axis
Write 0b111 (Detect Count, Z-axis) to the SNIFF Config Register (0x14) [2:0].
Write count - 1 (0 to 61, indicating 1 to 62 counts, independent from X and Y) to the SNIFF Threshold Ctrl Register (0x13) [5:0], e.g. for 3 events, write 0b00010.
4. Set the logical mode for combining of X/Y/Z SNIFF wakeup events.
 - a. OR – Wakeup is triggered when any of the channels have met threshold & count.
Write 0b0 (OR) to the SNIFF Threshold Ctrl Register (0x13) [6].
 - b. AND – Wakeup is triggered when all active channels have met threshold & count.
Write 0b1 (AND) to the SNIFF Threshold Ctrl Register (0x13) [6].



-
5. Set SNIFF delta mode
 - a. C2P (current to previous) mode – The delta count between current and previous samples is a moving window. The SNIFF logic uses the current sample and the immediate previous sample to compute a delta.
Write 0b0 (C2P) to the SNIFF Threshold Ctrl Register (0x13) [7].
 - b. C2B (current to baseline) mode – The delta count is computed from subtracting the current sample from the first sample stored when entering SNIFF mode.
Write 0b1 (C2B) to the SNIFF Threshold Ctrl Register (0x13) [7].
 6. Set INT control
 - a. Generate INT when activity is detected in SNIFF mode and the device auto-transits to CWAKE mode – Write 0b0 to the Interrupt Control Register (0x17) [2].
 - b. Disable other INTs – Write 0b0000 to the Interrupt Control Register (0x17) [6:3].
 7. Switch to SNIFF mode
 - a. Initialise
Write a 0x42 to the Initialisation Register 1 (0x0F).
 - b. Switch mode
Write 0b010 (SNIFF) to the Mode Control Register (0x10) [2:0].

NOTE: For detailed register description, please refer the Register section in the datasheet.



3 FORMULA FOR DELTA g TO THRESHOLD

The formula varies from power mode to power mode.

Please refer to the power mode in use.

3.1 HIGH PRECISION MODE

$$\text{Threshold} = 16,672 * \Delta g / (0.85 * (\text{Gain} + 40))$$

Please find the corresponded value for threshold (0 to 63) in the following table.

(e.g. for gain = 110 and Δg = 0.2, please set threshold to 26)

gain Δg	90	110	130	150	170	190	210	230	250
0.1	15	13	12	10	9	9	8	7	7
0.2	30	26	23	21	19	17	16	15	14
0.3	45	39	35	31	28	26	24	22	20
0.4	60	52	46	41	37	34	31	29	27
0.5	-	-	58	52	47	43	39	36	34
0.6	-	-	-	62	56	51	47	44	41
0.7	-	-	-	-	-	60	55	51	47
0.8	-	-	-	-	-	-	63	58	54
0.9	-	-	-	-	-	-	-	-	61

Table 1. Look Up Table for Threshold (High Precision Mode)

NOTE: The value of gain usually falls in the range of 88 to 264.



4 SAMPLE CODE

Please refer to mCube GitHub Repository:

https://github.com/mcubemems/mCube_mc36xx_arduino_driver.

Function `sensorsniff()` in `MC36XX_demo.ino` demonstrates the initialising sequence for SNIFF.

```
void sensorsniff()
{
    // Switch to STANDBY mode
    MC36XX_acc.stop();
    // Set SNIFF threshold
    MC36XX_acc.SetSniffThreshold(MC36XX_AXIS_X,5);
    MC36XX_acc.SetSniffThreshold(MC36XX_AXIS_Y,5);
    MC36XX_acc.SetSniffThreshold(MC36XX_AXIS_Z,5);
    // Set SNIFF detect count
    MC36XX_acc.SetSniffDetectCount(MC36XX_AXIS_X,3);
    MC36XX_acc.SetSniffDetectCount(MC36XX_AXIS_Y,3);
    MC36XX_acc.SetSniffDetectCount(MC36XX_AXIS_Z,3);
    // Set the logical mode for combining of X/Y/Z SNIFF wakeup events
    MC36XX_acc.SetSniffAndOrN(MC36XX_ANDORN_OR);
    // Set SNIFF delta mode
    MC36XX_acc.SetSniffDeltaMode(MC36XX_DELTA_MODE_C2P);
    // Set INT control
    MC36XX_acc.SetINTCtrl(0,0,0,0,1);
    // Switch to SNIFF mode
    MC36XX_acc.sniff();
}
```



5 REVISION HISTORY

Date	Revision	Description
2020-03	1.0	First release.
2020-10	1.1	Revised values for register 0x14 [2:0] setting to be in binary.



6 LEGAL

1. M-CUBE reserves the right to make corrections, modifications, enhancements, improvements and other changes to its products and to this document at any time and discontinue any product without notice. The information contained in this document has been carefully checked and is believed to be accurate. However, M-CUBE shall assume no responsibilities for inaccuracies and make no commitment to update or to keep current the information contained in this document.
2. M-CUBE products are designed only for commercial and normal industrial applications and are not suitable for other purposes, such as: medical life support equipment; nuclear facilities; critical care equipment; military / aerospace; automotive; security or any other applications, the failure of which could lead to death, personal injury or environmental or property damage. Use of the products in unsuitable applications are at the customer’s own risk and expense.
3. M-CUBE shall assume no liability for incidental, consequential or special damages or injury that may result from misapplication or improper use of operation of the product.
4. No license, express or implied, by estoppel or otherwise, to any intellectual property rights of M-CUBE or any third party is granted under this document.
5. M-CUBE makes no warranty or representation of non-infringement of intellectual property rights of any third party with respect to the products. M-CUBE specifically excludes any liability to the customers or any third party regarding infringement of any intellectual property rights, including the patent, copyright, trademark or trade secret rights of any third party, relating to any combination, machine, or process in which the M-CUBE products are used.
6. Examples of use described herein are provided solely to guide use of M-CUBE products and merely indicate targeted characteristics, performance and applications of products. M-CUBE shall assume no responsibility for any intellectual property claims or other problems that may result from applications based on the examples described herein.
7. Information described in this document including parameters, application circuits and its constants and calculation formulas, programs and control procedures are provided for the purpose of explaining typical operation and usage. “Typical” parameters that may be provided in M-CUBE data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters including “Typicals,” must be validated for each customer application by customer’s technical experts. In no event shall the information described be regarded as a guarantee of conditions or characteristics of the products. Therefore, the customer should evaluate the design sufficiently as whole system under the consideration of various external or environmental conditions and determine their application at the customer’s own risk. M-CUBE shall assume no responsibility or liability for claims, damages, costs and expenses caused by the customer or any third party, owing to the use of the above information.



M-CUBE and the M-CUBE logo are trademarks of M-CUBE, Inc.,

All other product or service names are the property of their respective owners.

© M-CUBE, Inc. 2020. All rights reserved.

