Application Note

AN-013 Accelerometer with FIFO for Low Power Applications

Introduction

The application note explains how the implementation of FIFO benefits low power applications using accelerometer.

It also introduces the optimal size of FIFO for reducing current consumption with the diminishing return of increasing FIFO size.

Document valid for	MC3600 Series, MC3451
Document number	AN-013
Document revision	0.1
Document release date	2020/4/22

CONTENTS

1 RELEVANT MCUBE DEVICES

The concepts and examples in this application note are applicable to all of the following mCube devices:

- **MC3600 Series** Accelerometers
- **MC3451** Accelerometer

2 INTRODUCTION AND BACKGROUND

The Internet Of Things (IOT) has spawned a variety of sub categories of connected devices such as the Industrial Internet Of Things (IIOT) and the Internet Of Moving Things (IOMT) [1] to name a few. The latter consists of sensors capable of measuring motion. This is chiefly accomplished by the use of accelerometers or Gyros or magnetometers or a combination of all three of these types of sensors. Another characteristic of IOMT devices is that they are generally battery powered. Smart phones, tablets and wearables are some examples. Since these devices are battery powered, it is critical that the system power consumption is kept to a minimum.

3 METHODS OF MINIMIZING IOMT SYSTEM POWER CONSUMPTION

“Power consumption arises as a third axis in the optimization space in addition to the traditional speed (performance) and area (cost) dimensions.” [2]

4 FIFO IMPLEMENTATION FOR LOW POWER

The general idea behind the use of a FIFO to reduce the power consumption of a system is quite simple. During the time when the FIFO is being filled, the components of the system that are not filling the FIFO are in low power mode such as standby or sleep mode. Once the FIFO is filled and data is available, the necessary component of the system awaken to receive the data and batch process it. This cycle simply repeats again and again.

The interesting point to consider here is the relationship between the FIFO size and the real power savings of the system. The assumption might be that the greater the FIFO size the greater the power savings. While this might be true mathematically, in a practical sense, increasing the sample depth of the FIFO may be an exercise in the law of diminishing returns.

Let's consider a typical example of a system that uses an accelerometer to count steps. A very simple, high-level block diagram of a pedometer is illustrated in Figure 1 below.

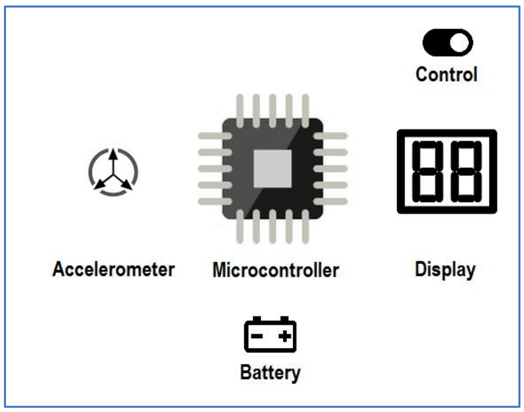

Figure 1. Pedometer Block Diagram.

While pedometer algorithms that use accelerometer data to count steps can be extremely complex, the hardware components used for these systems are so highly integrated that they tend to be very small and minimal in terms of parts count. In most cases the interconnections between the blocks above tend to be digital: I2C or SPI.

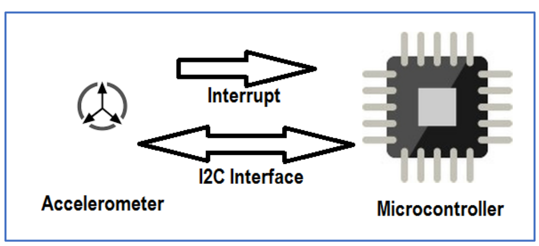

Figure 2. Accelerometer/Micro Interface.

Figure 2 shows a common connection scheme between the accelerometer and the microcontroller. The I2C interface is used to setup and control the accelerometer as well as to get the acceleration data from the sensor. The Interrupt line from the accelerometer to the microcontroller is used to inform the microcontroller that the FIFO is full. This is a key feature of a FIFO-based sensor system.

As described above, the microcontroller is placed in a low power mode such as sleep while the accelerometer collects acceleration data and fills its FIFO buffer. Once the FIFO is full, the accelerometer wakes up the microcontroller by use of the interrupt line. The microcontroller then collects the data from the accelerometer via the I2C interface for processing. Following the processing of the data and updating the display, the microcontroller instructs that accelerometer to begin filling the FIFO again and then returns to a low power state.

Based on some common microcontroller datasheet data, we can approximate a few electrical characteristics of the microcontroller:

- Wake current = 5mA
- Sleep current = 5uA

Likewise, we can accurately approximate the current consumption of the accelerometer by consulting the mCube datasheet [3]:

- Current = .9uA @ 25Hz ODR

Let's assume that the microcontroller takes an amount of time, 250 uS, to process each data point stored in the FIFO. We are going to use this process time to calculate the on time of the microcontroller. We'll then use this to essentially calculate the duty cycle of the microcontroller in terms of current consumed.

We'll also need to make an assumption about the microcontroller wakeup time and the time it takes it to process and display the data. The wakeup time and process time is listed below.

With no FIFO employed and the microcontroller on 100% of the time (Duty multiplier = 1), the total system current consumption is simply the microcontroller current plus the accelerometer current: 5000 + .9 = 5000.9 uA.

				Current [uA]		
FIFO Size [Samples]	FIFO Fill Time [uS]	Data Point Process Time by uP [uS]	uP duty	Accel	uP	Total Current [uA]
0	0	0	1	0.9	5000	5000.9

Table 1.

Let's employ a 2-sample FIFO. That is to say that the microcontroller will sleep until the FIFO is full at which time the accelerometer will awaken the microcontroller to receive the data and process it.

				Current [uA]		
FIFO Size [Samples]	FIFO Fill Time [uS]	Data Point Process Time by uP [uS]	uP duty	Accel	uP	Total Current [uA]
0	0	0	1	0.9	5000	5000.9
2	80000	5500	0.064327485	0.9	321.637427	322.5374

Table 2.

Table 2 includes the current consumed by the pedometer system using the 2-sample FIFO.

Where:

FIFO Fill Time [uS] = 1/data rate[Hz] * FIFO size

Data rate = 25 Hz; 1/25[Hz] = 40000 [uS]
40000 [S] * 2 FIFO samples = 80000 [uS]

Data Point Processing Time by uP [uS] =
FIFO size * uP Data Point Processing Time [uS]
+ uP Wake & Processing Time [uS]

This is the total time that the microprocessor takes after it is awakened by the accelerometer's interrupt.

uP duty = uP on time / total time on

Where the "total time on" is the time needed to fill the FIFO plus the time it takes for the microcontroller to process the data.

Finally, the last three columns in the table show the Current [uA] consumed by the accelerometer, the microprocessor (uP) and the total system current consumed. The current consumed by the microcontroller is simply the maximum current times the uP duty. The total system current consumed is simply the current consumed by the microprocessor plus the current consumed by the accelerometer.

It is expected that the current consumed will be dramatically reduced when the FIFO is used because the microcontroller is off for a vast majority of the time. In this case, it's off for nearly 94% of the time.

Let's add to the depth of the FIFO:

				Current [uA]		
FIFO Size [Samples]	FIFO Fill Time [uS]	Data Point Process Time by uP [uS]	uP duty	Accel	uP	Total Current [uA]
0	0	0	1	0.9	5000	5000.9
2	80000	5500	0.064327485	0.9	321.637427	322.5374
4	160000	6000	0.036144578	0.9	180.722892	181.6229
8	320000	7000	0.021406728	0.9	107.033639	107.9336
16	640000	9000	0.013867488	0.9	69.3374422	70.23744
32	1280000	13000	0.010054138	0.9	50.2706883	51.17069
64	2560000	21000	0.008136381	0.9	40.6819062	41.58191
128	5120000	37000	0.007174714	0.9	35.8735699	36.77357
170	6800000	47500	0.006936838	0.9	34.6841913	35.58419

Table 3.

Table 3 shows exactly what we expect to see. The total current consumed does decrease with an increased number of FIFO samples. However, the amount that the total current decreases begins to get smaller as the FIFO size grows. Figure 3 illustrates this graphically. Most of the efficiency gain is realized with a FIFO sample size of only 16 or 32.

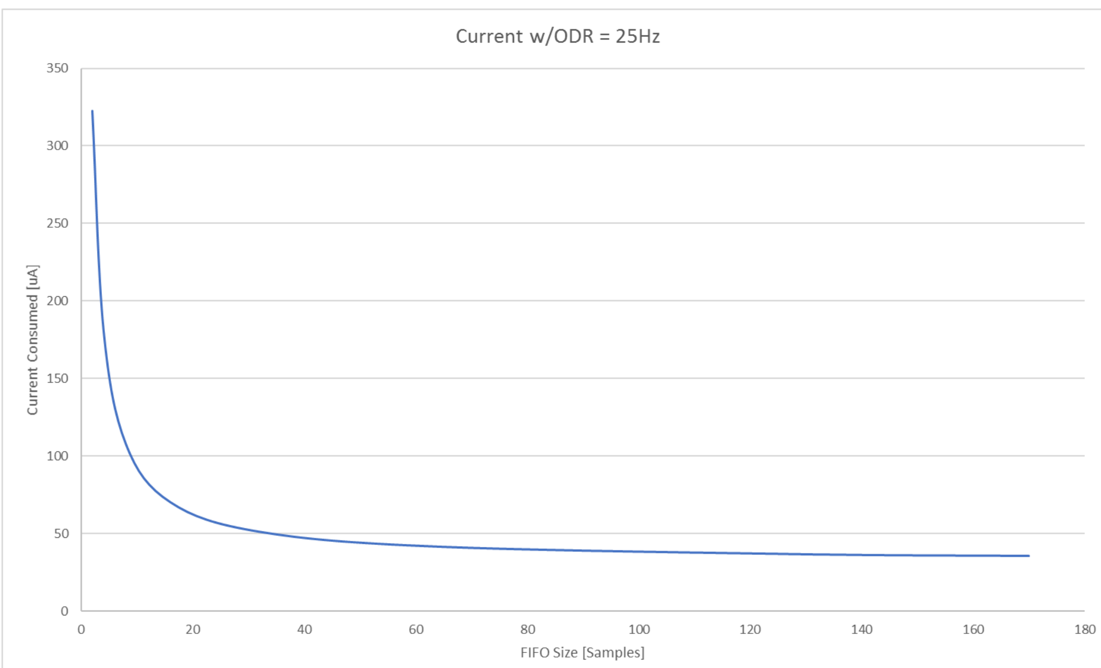

Figure 3.

Another way to look at the data is to look at the percent decrease of the current when compared to the NO FIFO scenario.

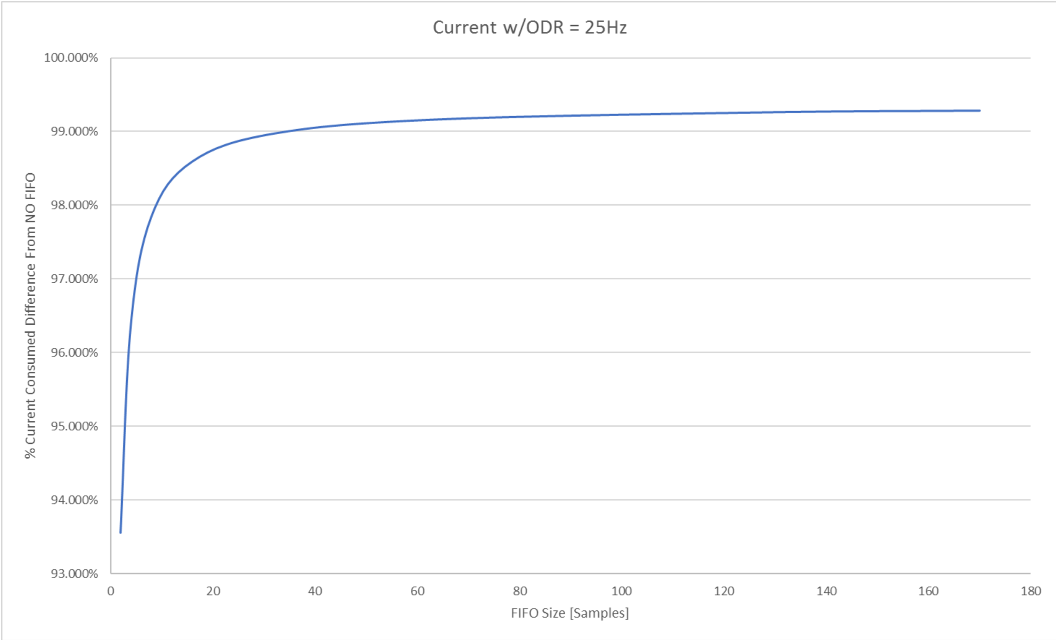

Figure 4.

With a FIFO sample size of 32, the current consumption has decreased by nearly 99% when compared to the case of NO FIFO.

5 CONCLUSION

The use of a FIFO in an application where current consumption is critical may prove to be extremely advantageous. In fact, system current consumption can be reduced by 99% if used properly and batch processing is practical. *However, there is a diminishing return in terms of the current consumption decrease with the increased FIFO size.* Further, there is a cost tradeoff that needs to be considered when large buffer sizes are used. Remember FIFOs take up considerable die size and impact the device cost. The substantial cost increase needs to be weighed against the diminishing benefit in terms of the current consumed by the system in which the device is used.

REFERENCES

1. http://www.mcubemems.com/markets/internet-moving-things/
2. Power Reduction Techniques for Embedded Systems, 2008
   http://shodhganga.inflibnet.ac.in/bitstream/10603/32821/12/12_chapter%203.pdf
3. https://mcubemems.com/wp-content/uploads/2019/11/MC3635-Datasheet-APS-048-0044v1.7.pdf

REVISION HISTORY

Date	Revision	Description
2020-04	1.0	First release.

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.