Technical Document

How to use BMA400 FIFO

Bosch Sensortec

BOSCH

Invented for life

Table of contents

1 FIFO DESCRIPTION

BMA400 is an ultra-low current 3-axis 12-bit accelerometer. After power on BMA400 enters sleep mode by default that consumes about 160nA. Users need to write value of 0x02 to register 0x19 to bring BMA400 to normal mode. Then it consumes about 3.5uA at normal mode from 12.5Hz to 800Hz output data rate (ODR). During normal mode BMA400 continuously measures the acceleration without duty-cycling and can still achieve such low current consumption. Therefore, there is no aliasing issue for BMA400 at normal mode.

BMA400 is integrated with a 1024-byte data FIFO. The FIFO captures data registers in frame and each frame contains only one sample of a sensor. The FIFO is written only in normal mode. The FIFO has two modes of operation i.e. Stream mode and FIFO mode. When FIFO_CONFIG0.fifo_stop_on_full = '0', the device is in stream mode this means that if the FIFO is full it overwrites the oldest data. When FIFO_CONFIG0.fifo_stop_on_full = '1', the device is in FIFO mode this means that if the FIFO is full it discards newest data.

The data collected is defined through fifo_data_src, fifo_x_en, fifo_y_en and fifo_z_en bits. The FIFO is disabled when fifo_x_en, fifo_y_en and fifo_z_en bits are set to '0'. BMA400 also generates 2 interrupts. It generates a FIFO full interrupt if the filling level of FIFO is equal or greater than 1016. It generates a FIFO watermark interrupt when the filling level of the FIFO is greater or equal to the watermark level.

The FIFO supports two modes for acceleration data storage in FIFO: 12 bits stored as two bytes into FIFO and 8 bits stored as single byte into FIFO per acceleration axis. By default the data is stored as 12 bits per acceleration axis. The 8-bit mode can be activated by setting FIFO_CONFIG0.fifo_8bit_en = '1'.

FIFO Length
FIFO length can be calculated from registers FIFO_LENGTH0 (0x12) and FIFO_LENGTH2 (0x13). FIFO byte count registers are updated only after a full frame has been written or read from the FIFO.

$$\text{FIFO Length (bytes)} = \underbrace{\text{fifo\_bytes\_cnt\_7\_0}}_{\text{(FIFO\_LENGTH0)}} + 256 * \underbrace{\text{fifo\_bytes\_cnt\_10\_8}}_{\text{(FIFO\_LENGTH1)}}$$

2 FIFO REGISTERS

The FIFO can be configured using 3 register viz. FIFO_CONFIG0 (0x26), FIFO_CONFIG1 (0x27) and FIFO_CONFIG2 (0x28). FIFO_CONFIG0 configures all the necessary settings for data collection in FIFO. The registers FIFO_CONFIG1 & FIFO_CONFIG2 configure the threshold for FIFO watermark.

The definition of register FIFO_CONFIG0 (0x26) is as shown below.

Bit	Bit-7	Bit-6	Bit-5	Bit-4
Access	RW	RW	RW	RW
Reset value	0	0	0	0
Content	fifo_z_en	fifo_y_en	fifo_x_en	fifo_8bit_en

Bit	Bit-3	Bit-2	Bit-1	Bit-0
Access	RW	RW	RW	RW
Reset value	0	0	0	0
Content	fifo_data_src	fifo_time_en	fifo_stop_on_full	auto_flush

Description of the parameters for this register:

Data	Bit No.	Description
fifo_z_en <1>	7	z-channel data storage control: '0' – do not store; '1' – store data.
fifo_y_en <1>	6	y-channel data storage control: '0' – do not store; '1' – store data.
fifo_x_en <1>	5	x-channel data storage control: '0' – do not store; '1' – store data.
fifo_8bit_en <1>	4	enables 8 bit FIFO mode: '0' – store data in 12bit format (default); '1' – store data in 8bit format.
fifo_data_src <1>	3	Define the data source selection for the acceleration data 0 acc_filt1 1 acc_filt2 The data source can be selectable between acceleration from accfilt1 or from acc_filt2. acc_filt1: Filter 1 has a variable ODR (output data rate) which can be configured between 800Hz and 12.5Hz. acc_filt2: Filter 2 has a fixed ODR of 100 Hz.
fifo_time_en <1>	2	enable sending of sensor time frame when reading burst from FIFO: '0' – disable sensor time; '1' – enable sensor time.
fifo_stop_on_full <1>	1	FIFO writing mode – stream mode / FIFO full mode. value mode 0 streaming 1 fifo-stop-on-full In streaming mode, the oldest data in FIFO is overwritten if the FIFO is full. In fifo-stop-on-full mode, no data is written in the FIFO when it is full.
auto_flush <1>	0	auto flush FIFO when changing power mode: '0' –no FIFO flush on changing power mode; '1' – FIFO flush on changing power mode.

The definition of register FIFO_CONFIG1 (0x27) is as shown below.

Bit	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset value	0	0	0	0	0	0	0	0
Content	fifo_watermark_7_0							

fifo_watermark_7_0: LSB of FIFO watermark threshold configuration

The definition of register FIFO_CONFIG2 (0x28) is as shown below.

Bit	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset value	0	0	0	0	0	0	0	0
Content	reserved					fifo_watermark_10_8		

fifo_watermark_10_8: MSB of FIFO watermark threshold configuration

These register are used to define the FIFO watermark threshold.
fifo_watermark_threshold <10:0> = fifo_watermark_7_0 + 256*fifo_watermark_10_8

3 FIFO FRAMES

FIFO captures data in frame which consist of a header and a payload. Each data frame consists of a 1 byte header which describes the properties of the frame and the data itself.

The header has a length of 8 bit and the following format.

Bit	7	6	5	4	3	2	1	0
Content	fh_mode		fh_param					0

fh_mode and fh_param indicate whether the frame is a data frame, a sensortime frame a control frame or an empty frame (all data 0).

Table: fh_mode and fh_param examples for different scenarios

fh_mode		fh_param						Header Value	Description
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0		
1	0	0	0	0	0	1	0	0x82	a sensor data frame operating in 8bit mode with only x axis enabled
1	0	0	0	0	1	0	0	0x84	a sensor data frame operating in 8bit mode with only y axis enabled
1	0	0	0	1	0	0	0	0x88	a sensor data frame operating in 8bit mode with only z axis enabled
1	0	0	0	0	1	1	0	0x86	a sensor data frame operating in 8bit mode with y & x axis enabled
1	0	0	0	1	0	1	0	0x8A	a sensor data frame operating in 8bit mode with z & x axis enabled
1	0	0	0	1	1	0	0	0x8C	a sensor data frame operating in 8bit mode with z & y axis enabled
1	0	0	0	1	1	1	0	0x8E	a sensor data frame operating in 8bit mode with z, y & x axis enabled
1	0	0	1	0	0	1	0	0x92	a sensor data frame operating in 12bit mode with only x axis enabled

1	0	0	1	0	1	0	0	0x94	a sensor data frame operating in 12bit mode with only y axis enabled
1	0	0	1	1	0	0	0	0x98	a sensor data frame operating in 12bit mode with only z axis enabled
1	0	0	1	0	1	1	0	0x96	a sensor data frame operating in 12bit mode with y & x axis enabled
1	0	0	1	1	0	1	0	0x9A	a sensor data frame operating in 12bit mode with z & x axis enabled
1	0	0	1	1	1	0	0	0x9C	a sensor data frame operating in 12bit mode with z & y axis enabled
1	0	0	1	1	1	1	0	0x9E	a sensor data frame operating in 12bit mode with z, y & x axis enabled
1	0	0	0	0	0	0	0	0x80	an empty frame. this header is followed by 0x00 which signifies no data (empty frame)
1	0	1	0	0	0	0	0	0xA0	a sensor-time frame. this frame is usually followed by 3 bytes of data
0	1	0	0	1	0	0	0	0x48	a control frame. this frame is followed by 1 byte of data which is inserted if there is any change on the configuration of the FIFO

Sensor data frame

In a data frame, fh_param<2:0> defines which sensors axes are included in the data part of the frame. fh_param<2/1/0> indicate whether z,y or x axis data are stored. fh_param<3> defines in which resolution – 8 or 12bit – the data are stored. Thus fh_param<3:0> allows to calculate the amount of data payload following the header.
As an example, data frames with 12bit and 8bit resolution are shown below, all axes enabled.

Sensor data frame (12bit resolution)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Header	1	0	0	1: 12bit	1:Z	1:Y	1:X	0
data	unused				acc_x<3:0>			
	acc_x<11:4>							
	unused				acc_y<3:0>			
	acc_y<11:4>							
	unused				acc_z<3:0>			
	acc_z<11:4>							

A Sensor data frame operating in 12bit mode will have a header frame as 0x9E if all the axis are enabled. Such frame will have a total payload of 7 bytes, where 1st byte is the header and remaining 6 bytes are values of x, y & z axis

Sensor data frame (8bit resolution)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Header	1	0	0	0: 8bit	1:Z	1:Y	1:X	0
data	acc_x<11:4>							
	acc_y<11:4>							
	acc_z<11:4>							

A Sensor data frame operating in 8bit mode will have a header frame as 0x8E if all the axis are enabled. Such frame will have a total payload of 4 bytes, where 1st byte is the header and remaining 3 bytes are values of x, y & z axis

Empty frame

An empty frame is delivered if the last frame in the FIFO was already read out or if the FIFO is empty. The header for an empty frame is 0x80 which is followed by 1 data byte of value 0x00. The FIFO empty frame format is shown below.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Header	1	0	0	0	0	0	0	0
Data	0	0	0	0	0	0	0	0

Sensor time frame

A sensor time frame is sent after all the data frame have been transmitted and the burst read carries on requesting data. A sensor time frame header is 0xA0 which is followed by 3 bytes of sensor time.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Header	1	0	1	0	0	0	0	0
Data	sensor_time<7:0>							
	sensor_time<15:8>							
	sensor_time<23:16>							

Control frame

A control frame is inserted in the FIFO when there is some change in configuration of the FIFO. Bit 0-2 are the ones which indicate what changes were made to the FIFO configurations. The FIFO control frame format is shown below.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Header	0	1	0	0	1	0	0	0
Opcode	0	1	1	0	0	acc_config1_chg	acc_config0_chg	fifo_config0_chg

Description of the parameters for this frame:

Opcode	Bit No.	Description
fifo_config0_chg	0	this bit is set to ‘1’ if there is a change in fifo data src. (fifo_config0.fifo_data_src)
acc_config0_chg	1	this bit is set to ‘1’ if the change in filt1_bw is valid for data stored in FIFO (ACC_CONFIG0.filt1_bw)

acc_config1_chg	2	this bit is set to '1' if the change in acc_odr or acc_osr or acc_range is valid for data stored in FIFO (ACC_CONFIG1.acc_odr, ACC_CONFIG1.acc_osr, ACC_CONFIG1.acc_range)

If more changes become active at one acceleration sample just one control frame will be inserted, with more than one of the three config_chg bits set.

4 FIFO INTERRUPTS

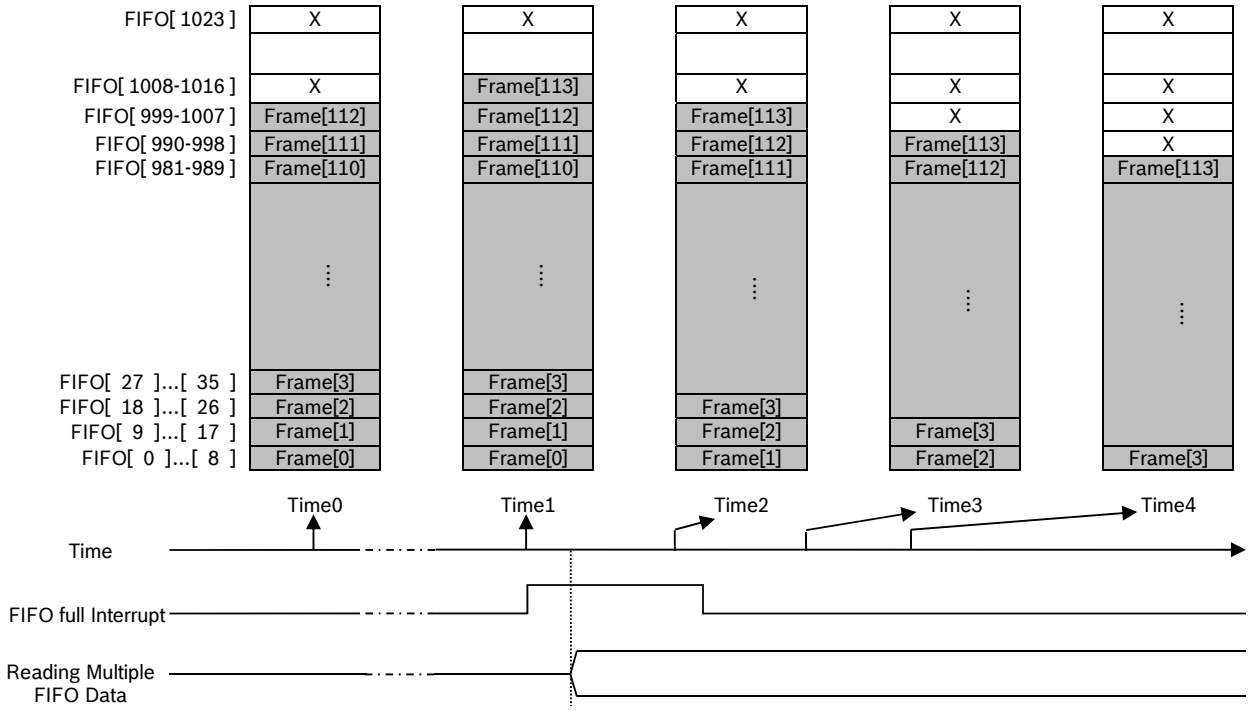
The FIFO supports the full and watermark interrupts. FIFO full interrupt and watermark interrupt can be enabled or disabled through the Register INT_CONFIG0 (0x1F) and can be mapped or unmapped to interrupt pin 1 or 2 through the Registers INT1_MAP (0x21) AND INT2_MAP (0X22). Detail information can be found in the BMA400 datasheet.

4.1 FIFO FULL INTERRUPT

The FIFO full interrupt is issued when the FIFO is full and the next full data sample would cause a FIFO overflow, which may lead to samples being deleted. For BMA400, FIFO full interrupt is triggered if FIFO length is equal to or more than 1016 Bytes. In BMA400, largest frame length is 7 bytes. But each time a frame is written, 9 bytes can be written to the FIFO in total, consisting 2 frames: one with the measurement results (7 bytes) and configuration change frame consisting of 2 bytes. So a FIFO full interrupt is generated if the available space in FIFO is less than 9 bytes. (i.e equal to or higher than 1016 bytes)

Mode	8 bit mode			12 bit mode		
Enabled axis	Single (X, Y or Z)	Any Two (XY, YZ or XZ)	All Three (XYZ)	Single (X, Y or Z)	Any Two (XY, YZ or XZ)	All Three (XYZ)
Frame Size	2 Bytes	3 Bytes	4 Bytes	3 Bytes	5 Bytes	7 Bytes
FIFO Byte Count when full interrupt generated	1016 Bytes (508 Frames)	1017 Bytes (339 Frames)	1016 Bytes (254 Frames)	1017 Bytes (339 Frames)	1020 Bytes (204 Frames)	1022 Bytes (146 Frames)

FIFO Operation with full interrupt
An example is show in the figure below. Here we consider all the frame are of 9 bytes (7 bytes of measured results and 2 bytes of configuration change frame). At Time0 the FIFO fill level in fifo_bytes_cnt is 1008 bytes. At Time1, the FIFO fill level is updated to 1016 bytes as Frame[113] is pushed into the FIFO. Then the FIFO full interrupt is fired. After some latency, the interrupt is serviced and a burst read that reads all the frames in the FIFO is issued. At Time2, a complete frame Frame[0] is read, fifo_bytes_cnt is updated which is below the full interrupt level and the full interrupt is cleared.

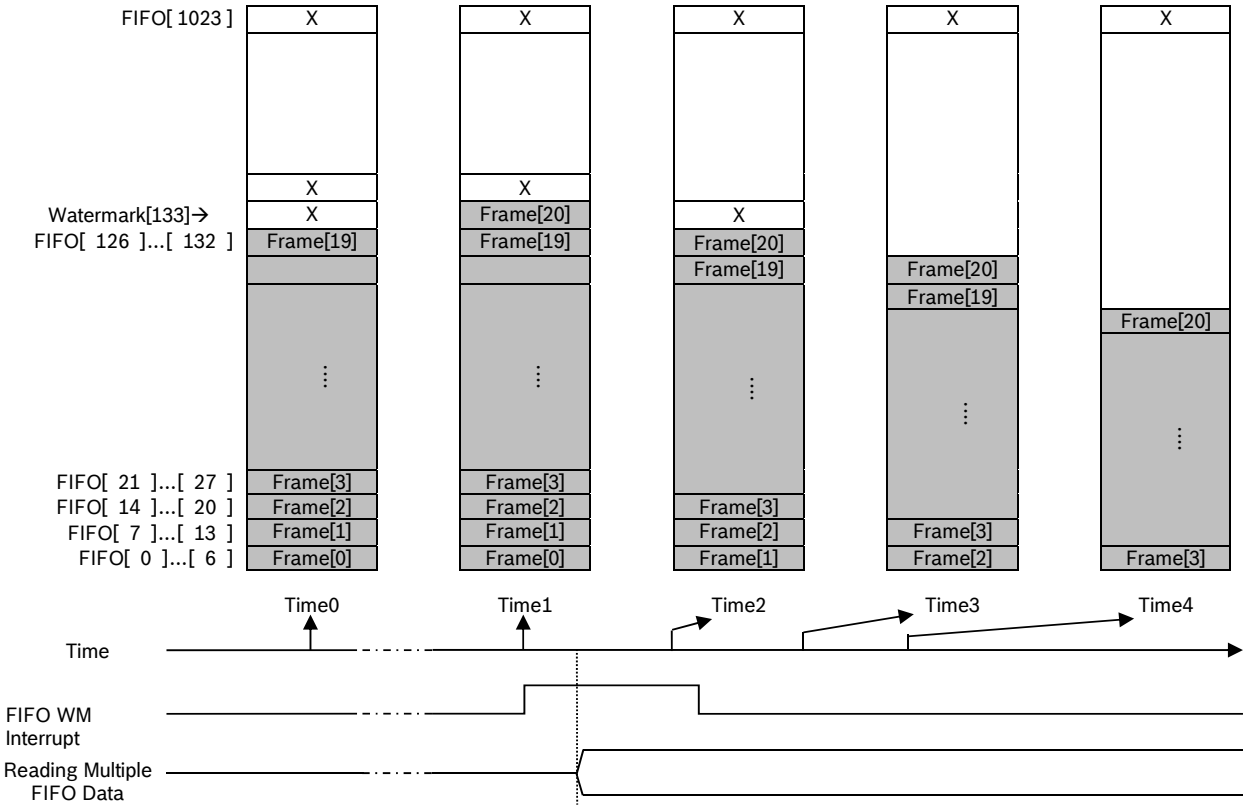


4.2 FIFO WATERMARK INTERRUPT

The watermark interrupt is triggered when the FIFO fill level in fifo_byte_cnt in FIFO Length Registers (0x12-0x13) is above a pre-configured fifo_watermark in FIFO Configuration Registers (0x27-0x28) and it remains set until the condition causing it is eliminated. To clear the watermark interrupt, read the FIFO until the fill level in fifo_byte_counter is lower than the value stored in fifo_watermark; however, reading more than the valid frames in FIFO is recommended for two reasons:

- It prevents the watermark interrupt from triggering too often, thus eliminating the benefit of using FIFO.
- A sensortime frame is returned after the last valid frame when more data are read than valid frames are in FIFO.

FIFO Operation with watermark interrupt
An example is show in the figure below how the FIFO operates with watermark interrupt. The pre-configured watermark is 133 bytes. At Time0 the FIFO fill level in fifo_byte_cnt is 132 bytes. At Time1, the FIFO fill level is above the watermark level as Frame[20] is pushed into the FIFO. The FIFO watermark interrupt is fired. At Time2, a complete frame Frame[0] is read, fifo_byte_cnt is updated, the fifo_bye_cnt is below the watermark level. So the FIFO watermark interrupt is cleared. Until at Time3 another complete frame Frame[1] is read.



5 SAMPLE CODE

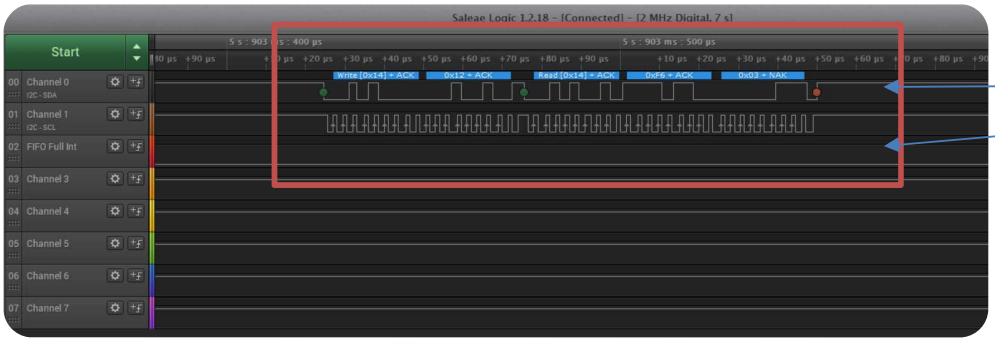
Below given are some pseudo codes for initializing BMA400 with FIFO Full and FIFO Watermark interrupts.

Example 1:

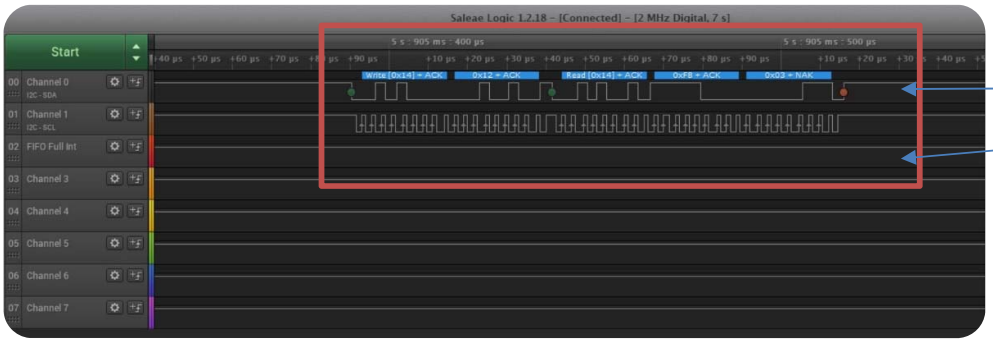
This is a pseudo code to initialize the BMA400 to record data in FIFO. Here we initialize the FIFO to capture 8Bit Data for only Z axis at 100 Hz. Also enable & map FIFO Full Interrupt to interrupt pin 1. The FIFO in this example is configured to operate in stop on full mode. Auto_flush when changing mode is deactivated.

```
void init_BMA400(void)
{
        // configure common control registers

        Write 0x02 to register 0x19;        // bring BMA400 to normal mode form sleep mode
        Write 0x48 to register 0x1A;        // set BMA400 to 100Hz ODR, +/-4g full scale range
                                            and 0 over sampling which means one single
                                            measurement without averaging
        Write 0x00 to register 0x1B;        // default value for acc_filt1 for variable ODR filter
```

```
// FIFO configuration

Write 0x96 to register 0x26          // fifo configuration. Z axis data enabled, 8 bit data
                                     mode. Source is acc_filt1, fifo timestamp enabled, fifo
                                     operating in stop on full mode. And auto flush is not
                                     activated.

// configure interrupt registers

Write 0x20 to register 0x1F;         // enable FIFO Full interrupt
Write 0x20 to register 0x21;         // route FIFO Full interrupt to INT1 pin
Write 0x22 to register 0x24;         // set INT1 pin and INT2 pin both to push-pull and
                                     active-high
}
```



fifo_bye_cnt(0x12) = 0x3F7 (1015 bytes) which is less than 1016 bytes. Therefore fifo full interrupt is '0' (low)

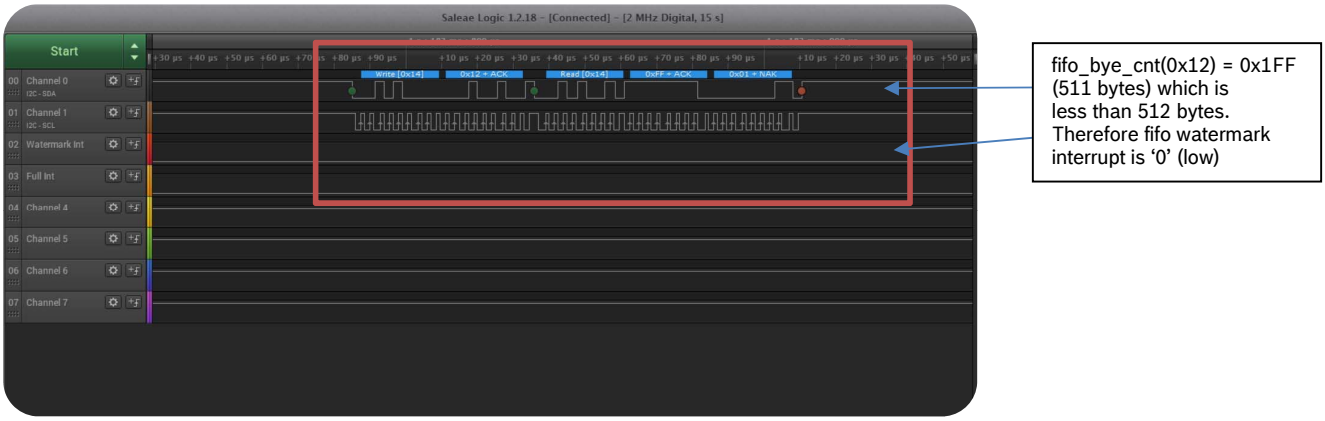


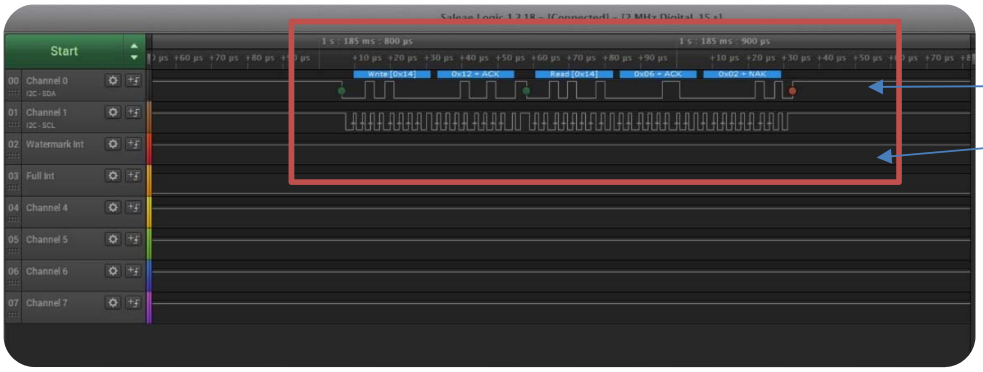
fifo_bye_cnt(0x12) = 0x3F7 (1015 bytes) which is less than 1016 bytes. Therefore fifo full interrupt is '0' (low)

Example 2:

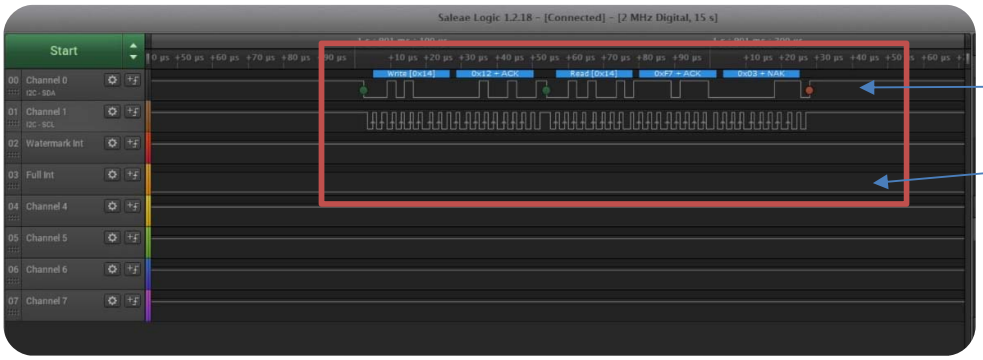
In this example we make use of FIFO Watermark Interrupt and FIFO Full Interrupt both at once. We initialize the FIFO to capture 12Bit Data for X, Y, Z axis at 25 Hz. Also enable & map FIFO Full Interrupt to interrupt pin 1 and FIFO Watermark Interrupt to interrupt pin 2. The FIFO in this example is configured to operate in stop on full mode. Auto_flush when changing mode is deactivated.

```
void init_BMA400(void)
{
        // configure common control registers

        Write 0x02 to register 0x19;        // bring BMA400 to normal mode form sleep mode
        Write 0x46 to register 0x1A;        // set BMA400 to 25Hz ODR, +/-4g full scale range and
                                            0 over sampling which means one single
                                            measurement without averaging
        Write 0x00 to register 0x1B;        // default value for acc_filt1 for variable ODR filter

        // FIFO configuration

        Write 0xE6 to register 0x26         // FIFO configuration. X, Y, Z axis data enabled, 12bit
                                            data mode. Source is acc_filt1, FIFO timestamp
                                            enabled, FIFO operating in stop on full mode. And
                                            auto flush is not activated.
        Write 0x00 to register 0x27         // FIFO watermark LSB is set to 00
        Write 0x02 to register 0x28         // FIFO watermark MSB is set to 2
                                            // FIFO watermark will be 0x200 = 512 bytes.

        // configure interrupt registers

        Write 0x60 to register 0x1F;        // enable FIFO Full & Watermark interrupt
        Write 0x20 to register 0x21;        // route FIFO Full interrupt to INT1 pin
        Write 0x40 to register 0x22;        // route FIFO Watermark interrupt to INT2 pin
        Write 0x22 to register 0x24;        // set INT1 pin and INT2 pin both to push-pull and
                                            active-high
}
```

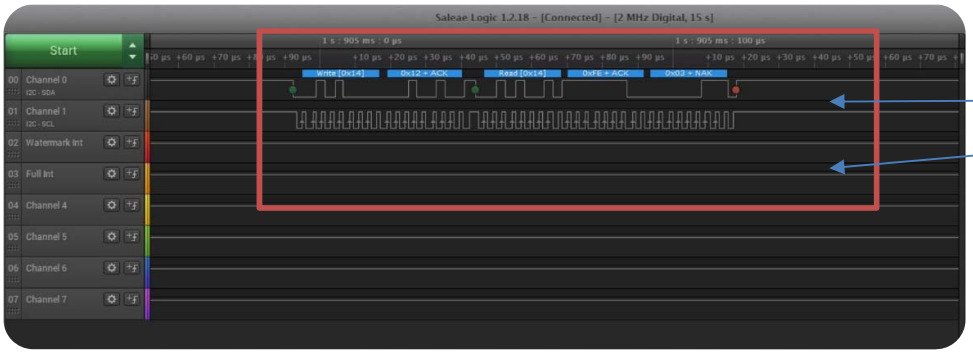




fifo_bye_cnt(0x12) = 0x206 (518 bytes) which is more than 512 bytes. Therefore fifo watermark interrupt is '1' (high)



fifo_bye_cnt(0x12) = 0x3F7 (1015 bytes) which is less than 1016 bytes. Therefore fifo full interrupt is '0' (low)



fifo_bye_cnt(0x12) = 0x3FE (1022 bytes) which is more than 1016 bytes. Therefore fifo full interrupt is '1' (high)

6 LEGAL DISCLAIMER

6.1 ENGINEERING SAMPLES

Engineering Samples are marked with an asterisk (*) or (e) or (E). Samples may vary from the valid technical specifications of the product series contained in this data sheet. They are therefore not intended or fit for resale to third parties or for use in end products. Their sole purpose is internal client testing. The testing of an engineering sample may in no way replace the testing of a product series. Bosch Sensortec assumes no liability for the use of engineering samples. The Purchaser shall indemnify Bosch Sensortec from all claims arising from the use of engineering samples.

6.2 PRODUCT USE

Bosch Sensortec products are developed for the consumer goods industry. They may only be used within the parameters of this product data sheet. They are not fit for use in life-sustaining or security sensitive systems. Security sensitive systems are those for which a malfunction is expected to lead to bodily harm or significant property damage. In addition, they are not fit for use in products which interact with motor vehicle systems.

The resale and/or use of products are at the purchaser’s own risk and his own responsibility. The examination of fitness for the intended use is the sole responsibility of the Purchaser.

The purchaser shall indemnify Bosch Sensortec from all third party claims arising from any product use not covered by the parameters of this product data sheet or not approved by Bosch Sensortec and reimburse Bosch Sensortec for all costs in connection with such claims.

The purchaser must monitor the market for the purchased products, particularly with regard to product safety, and inform Bosch Sensortec without delay of all security relevant incidents.

6.3 APPLICATION EXAMPLES AND HINTS

With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Bosch Sensortec hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights or copyrights of any third party. The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics. They are provided for illustrative purposes only and no evaluation regarding infringement of intellectual property rights or copyrights or regarding functionality, performance or error has been made.

7 DOCUMENT HISTORY AND MODIFICATION

Rev. No	Chapter	Description of modification/changes	Date
1.0		Document creation	May 22nd, 2019

Bosch Sensortec GmbH
Gerhard-Kindler-Strasse 8
72770 Reutlingen / Germany

Contact@bosch-sensortec.com
www.bosch-sensortec.com