MAX21105 USER GUIDE

Revision 1.4

July 2015

maxim integrated™

©2015 Maxim Integrated Products, Inc.
All rights reserved.

No part of this documentation may be reproduced nor distributed in any form or by any means, graphic, electronic, or mechanical, including but not limited to photocopying, scanning, recording, taping, e-mailing, or storing in information storage and retrieval systems without the written permission of Maxim Integrated Products, Inc. (hereafter, “Maxim”). Products that are referenced in this document such as Microsoft Windows® may be trademarks and/or registered trademarks of their respective owners. Maxim makes no claim to these trademarks. While every precaution has been taken in the preparation of this document, individually, as a series, in whole, or in part, Maxim, the publisher, and the author assume no responsibility for errors or omissions, including any damages resulting from the express or implied application of information contained in this document or from the use of products, services, or programs that may accompany it. In no event shall Maxim, publishers, authors, or editors of this guide be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Rev. 1.4, Jul 2015

CONTENTS

1 Introduction 5

2 Nomenclature 6

3 MAX21105 Description 7

4 Pin Description 8

5 I^2C Interface 9

1.1 I^2C Protocol 10
1.2 Slave Address 10
1.3 Acknowledge 10
1.4 Register Address 11
1.5 I^2C Operations 11
5.1.1 Write One Register 11
5.1.2 Write Multiple Registers 12
5.1.3 Read One Register 12
5.1.4 Read Multiple Registers 13
5.2 Configuration Registers 14

6 SPI Interface 15

6.1 SPI Protocol 15
6.2 Register Address 16
6.3 Configuration Registers 17

7 Interrupts 18

7.1 Interrupt Flags 18
7.2 Interrupt Lines 18
7.3 Rate Interrupts 19
7.4 Configuration Registers 21

8 Reading Data from MAX21105 22

8.1 Synchronous Reading 22
8.2 Asynchronous Reading 22

9 FIFO 23

9.1 FIFO Data Storage Selection 23
9.2 FIFO Modes Description 24
9.2.1 FIFO OFF Mode 24
9.2.2 Normal Mode 24
9.2.3 Interrupt Mode 26
9.2.4 Snapshot Mode 28
9.3 Example of FIFO Read/Write Pointers Evolution 30
9.4 Configuration Registers 31

10 Programming Example 32

10.1 Simple Read-Out Sequence, No FIFO, No Interrupts 33
10.2 Simple Read-Out Sequence, FIFO Normal Mode, No Interrupts 34
10.3 Simple Read-Out Sequence, Normal Mode, Data Ready Interrupts 35

11 Register Map 36
11.1 Register Overview 36
11.1.1 Common 36
11.1.2 Bank 00 37
11.1.3 Bank 01 38
11.1.4 Bank 02 39
11.2 Registers Description 40
11.2.1 Common 40
11.2.2 Bank 00 63
11.2.3 Bank 01 83
11.2.4 Bank 02 106

TABLES

Table 1: Pin Description 8
Table 2: I^2C External Component Properties 9
Table 3: I^2C Device Addresses 10
Table 4: SPI External Component Properties 15
Table 5: Power Mode Transition Token Codes 23

FIGURES

Figure 1: Block Diagram 7
Figure 2: I2C Interface Connection to an Application Processor 9
Figure 3: START (S), STOP (P), and Repeated START (Sr) Conditions 10
Figure 4: I^2C Write One Byte 11
Figure 5: I^2C Write a Burst of Data 12
Figure 6: I^2C Read One Byte 13
Figure 7: I^2C Read a Burst of Data 14
Figure 8: SPI Interface Connection to an Application Processor 15
Figure 9: SPI Protocol 16
Figure 10: Rate Interrupt ranges 19
Figure 11: FIFO Power Mode Transition Example 24
Figure 12: FIFO Normal Mode, Stop on Full 25
Figure 13: FIFO Normal Mode, Overwrite 26
Figure 14: FIFO Interrupt Mode, Stop on Full 27
Figure 15: FIFO Interrupt Mode, Overwrite 28
Figure 16: FIFO Snapshot Mode 29
Figure 17: FIFO Read/Write Pointer Evolution 30
Figure 18: Simple Read-Out Sequence, No FIFO, No Interrupts 33
Figure 19: Simple Read-Out Sequence, FIFO Normal, No Interrupts 34
Figure 20: Simple Read-Out Sequence, Normal Mode, w/ Data Ready Interrupts and No FIFO 35

1 Introduction

MEMS sensors are revolutionizing the way people interact with everyday technology, making it easier and more user-friendly. Maxim can leverage its analog integration expertise to develop and manufacture new breakthrough MEMS sensors being smaller, lower power and more accurate than ever.

Owning the entire supply chain, Maxim brings its customers complete, reliable and cost-effective solutions, ensuring prompt time-to-volume and time-to-market to effectively address high-volume applications in consumer and industrial market segments.

Thanks to its leadership in analog integration and its manufacturing experience in MEMS, Maxim is capable of high-volume production to meet the market's demands. Maxim’s manufacturing expertise and highest quality standards also guarantee high performance and product reliability.

Every MEMS sensor is tested and trimmed in factory so that for most consumer applications, no additional sensor calibrations are required. The end user can quickly verify the sensor’s operation without physically tilting or rotating the sensor thanks to the built-in self-test feature, which allows accelerating the time-to-market for mass production.

This User Guide will provide a clear picture of the guidelines for its use in consumer applications and a comprehensive description of his unique features. The final section of this guide will present the structure of the register file and the purpose of each field or every register.

2 Nomenclature

ODR	Output Data Rate
BW	Bandwidth
FS	Fulllscale
UI	User Interface
OIS	Optical Image Stabilization
MSB	Most Significant Byte
MSb	Most Significant Bit
LSB	Least Significant Byte
LSb	Least Significant Bit
HPF	Highpass Filter
LPF	Lowpass Filter
dps	Degrees per seconds
RFU	Reserved for future uses

3 MAX21105 Description

The MAX21105 is a monolithic 3-axis gyroscope plus 3-axis accelerometer inertial measurement unit (IMU) that provides unprecedented accuracy and stability over temperature and time.

The MAX21105 is the industry's most accurate six DoF inertial measurement units capable of working with a supply voltage as low as 1.71V designed to serve applications such as drone/helicopter toys, handsets and tablets, game controllers, motion remote controls, and other consumer devices. In particular, the MAX21105 features low gyroscope zero-rate level error (GZRLE), low and linear gyroscope zero-rate level drift over temperature (GZRLDT) and low gyroscope phase delay (GPD) that makes the MAX21105 ideally suited for both flight and camera platforms stabilization on drone applications.

A large 512-byte FIFO extends the time during which the application processor can stay in a power-saving state.

The MAX21105 is available in a 3mm x 3mm x 0.83mm package 16-lead plastic land grid array (LGA) package and can operate within a temperature range of -40°C to +85°C.

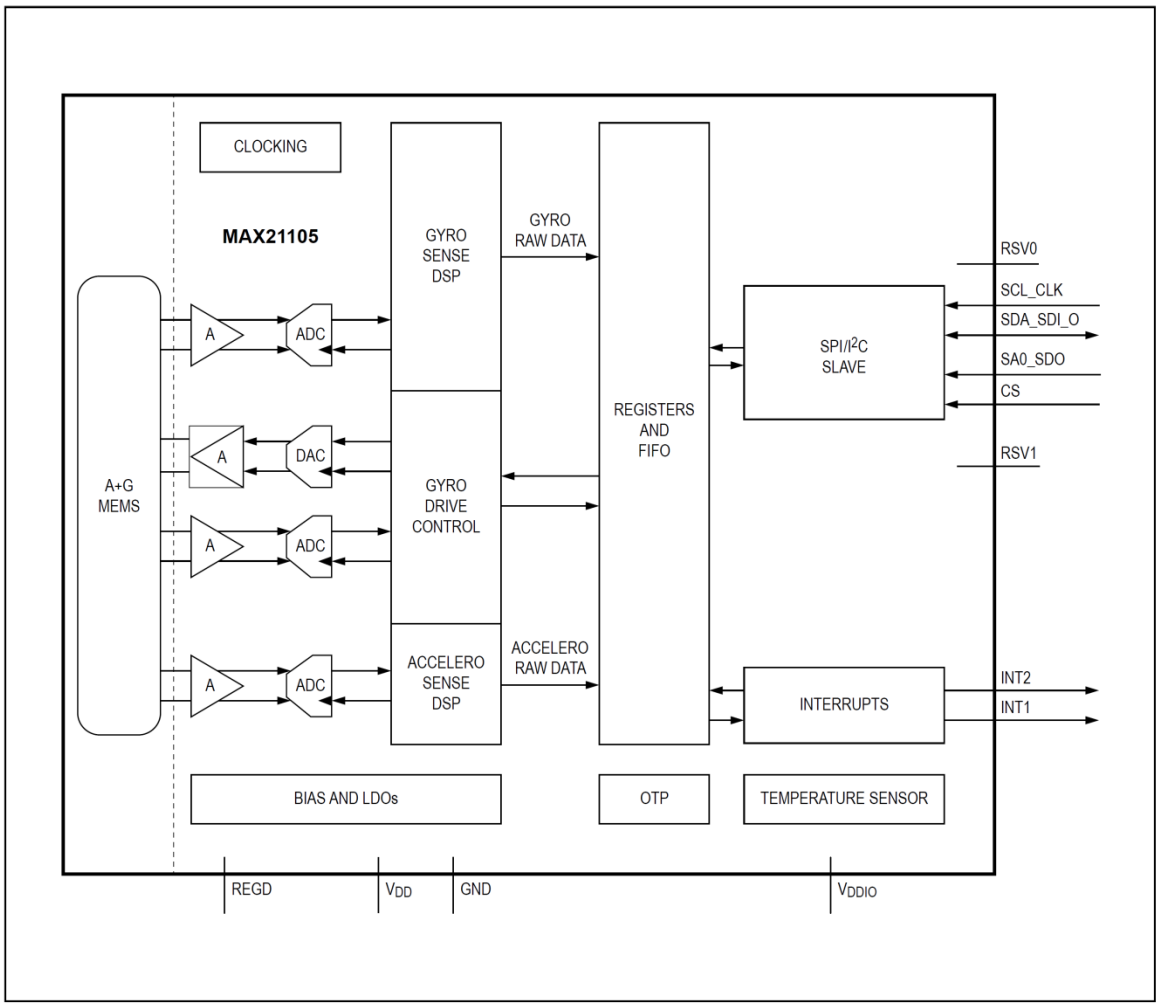


Figure 1: Block Diagram

4 Pin Description

Table 1: Pin Description

Pin	Name	Function
1	V_{DDIO}	Interface and Interrupt Pad Supply Voltage
2	N.C.	Not Internally Connected
3	N.C.	Not Internally Connected
4	SCL_CLK	SPI and I^2C Clock. When in I^2C mode, the IO has selectable anti-spike filter and delay to ensure correct hold time
5	GND	Power-Supply Ground
6	SDA_SDI_O	SPI In/Out Pin and I^2C Serial Data. When in I^2C mode, the IO has selectable anti-spike filter and delay to ensure correct hold time
7	SA0_SDO	SPI Serial Data Out or I^2C Slave Address LSB
8	CS	SPI Chip Select/Serial Interface Selection
9	INT2	Interrupt Line #2
10	RSV0	Reserved. Must be connected to GND
11	INT1	Interrupt Line #1
12	RSV1	Reserved. Must be left unconnected or connected to GND
13	REGD	Internal regulator output 2.2V max. A 100nF capacitor has to be connected to this pin as close as possible to ensure a proper device operation
14	V_{DD}	Analog Power Supply. Bypass to GND with a 0.1µF capacitor
15	N.C.	Not Internally Connected
16	N.C.	Not Internally Connected

5 I²C Interface

The MAX21105 can operate as an I²C slave device when communicating to the system processor, which thus acts as the master. The maximum bus speed is 3.4MHz (I²C HS); this reduces the amount of time the system processor is kept busy in supporting the exchange of data.

To connect a MAX21105 device to an I²C master, the SDA_SDI_O pin of the MAX21105 device must be connected to the SDA pin of the I²C master and the SCL_CLK pin of the MAX21105 device must be connected to the SCL pin of the I²C master. Both SDA and SCL lines must be connected to a pullup resistor. The SA0_SDO pin must be connected to VDD or GND to configure the MAX21105 I²C slave address (see *Table 3: I²C Device Addresses*).

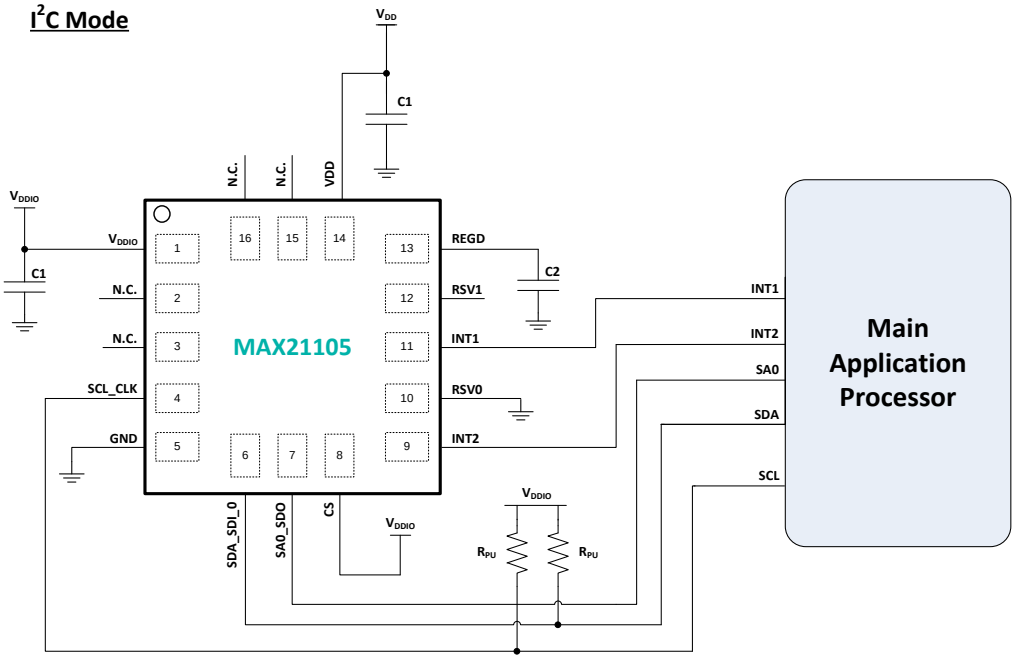


Figure 2: I²C Interface Connection to an Application Processor

Table 2: I²C External Component Properties

Component	Label	Specification	Quantity
V_{DDIO} /V_{DD} Bypass Capacitor	C1	Ceramic, X7R, 100nF ±10%, 4V	2
V_{DD} Bypass Capacitor	C2	Ceramic, X7R, 100nF ±10%, 2V	1
Pullup Resistor (I²C Mode only)	R_{PU}	1.1kΩ < R_{PU} < 10kΩ	2

1.1 I²C Protocol

To start an I²C request, the master sends a START condition (S), followed by the MAX21105's I²C address. Then, the master sends the address of the register to be programmed. Finally, the master terminates the communication by issuing a STOP condition (P) to relinquish the control of the bus, or a repeated START condition (Sr) to keep controlling it.

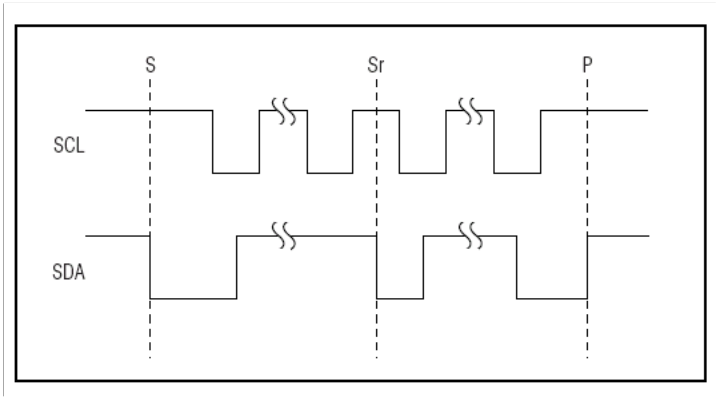


Figure 3: START (S), STOP (P), and Repeated START (Sr) Conditions

1.2 Slave Address

The slave address is used to identify the MAX21105 device in I²C communications. The address is defined as the seven most significant bits followed by the read/write bit. Set the read/write bit to 1 to request a read operation, or 0 to request a write operation (see the table below).

Table 3: I²C Device Addresses

I²C Base Address	SA0_SDO pin	R/W bit	Resulting Address
0x2C (6bit)	0	0	0xB0
0x2C	0	1	0xB1
0x2C	1	0	0xB2
0x2C	1	1	0xB3

1.3 Acknowledge

The acknowledge bit is sent after each byte. This bit allows the receiver to notify the transmitter that the byte has been received correctly and another byte may be sent. The master generates all clock pulses, including the ninth clock pulse of the acknowledge bit (8 bits of data + acknowledge bit).

To allow the receiver to send the acknowledge, the transmitter releases the SDA line during the acknowledge pulse, so that the receiver can pull the SDA line LOW during the ninth clock pulse to signal an Acknowledge (ACK), or release the SDA line (HIGH) to signal a Not Acknowledge (NACK).

The NACK is sent if the device is busy or a system fault occurs. It is also used by the master to signal the end of the transfer during a read operation.

1.4 Register Address

The I²C register address for the MAX21105 is composed by 6 bits of address and 1 bit whose meaning can be configured as:

- **Auto-increment:** if 0, in case of burst operation the initial register address is auto-incremented after every data byte; if 1, the operation is executed always on the same register;
- **Even parity:** this bit represents the even parity computed on the 6 bits of the register address;
- **Odd parity:** this bit represents the odd parity computed on the 6 bits of the register address;

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
N/A	MS/Parity	Address of the register you want to read or write					

1.5 I²C Operations

5.1.1 Write One Register

To write one register, the following steps must be executed:

1. The master sends a START condition;
2. The master sends the slave address (7 bits) plus a write bit (LOW);
3. If the slave address matches, the MAX21105 asserts an ACK on the data line;
4. The master sends the address (8 bits) of the register to be written;
5. The MAX21105 asserts an ACK on the data line ***only if the address is valid (NACK if not)****;*
6. The master sends 8 bits of data;
7. The MAX21105 asserts an ACK on the data line;
8. The master generates a STOP condition.

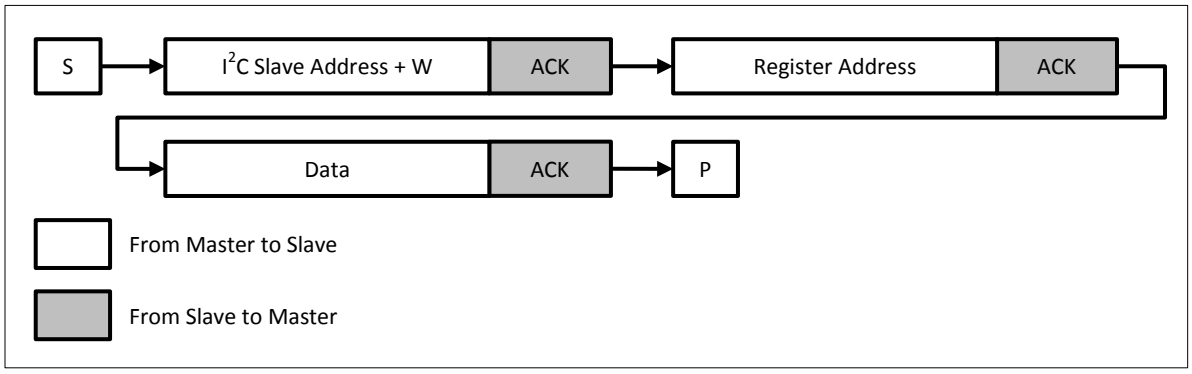


Figure 4: I²C Write One Byte

5.1.2 Write Multiple Registers

To execute a write of *n* consecutive registers, the following steps must be executed:

1. The master sends a START condition;
2. The master sends the slave address (7 bits) plus a write bit (LOW);
3. If the slave address matches, the MAX21105 asserts an ACK on the data line;
4. The master sends the Register Address of the first register to be written, setting the MS/Parity bit to 0 if it is configured as auto-increment bit (see *Register Address*);
5. The MAX21105 asserts an ACK on the data line ***only if the address is valid (NACK if not)***;
6. The master sends 8 bits of data;
7. The MAX21105 asserts an ACK on the data line;
8. Repeat 6 and 7 *n* times;
9. The master generates a STOP condition.

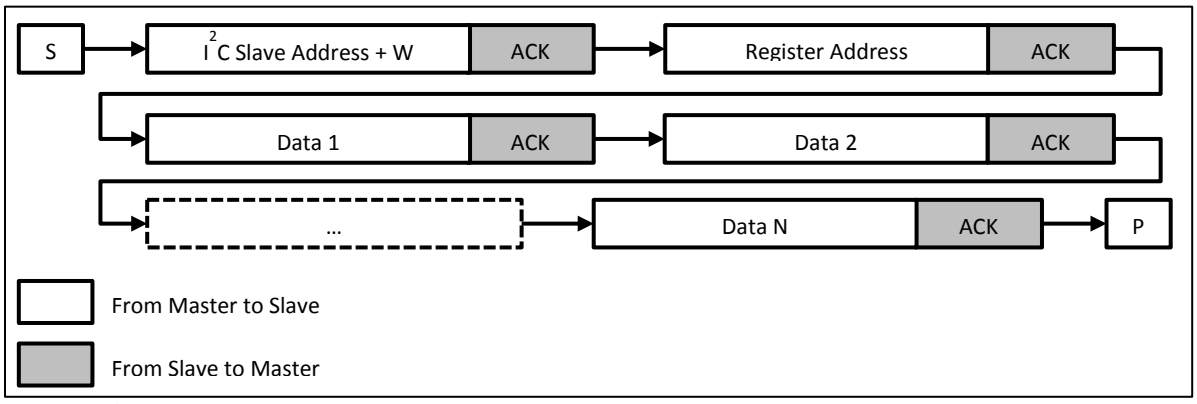


Figure 5: I²C Write a Burst of Data

5.1.3 Read One Register

To read one register, the following steps must be executed:

1. The master sends a START condition;
2. The master sends the slave address (7 bits) plus a write bit (LOW);
3. If the slave address matches, the MAX21105 asserts an ACK on the data line;
4. The master sends the address (8 bits) of the register to be read;
5. The MAX21105 asserts an ACK on the data line ***only if the address is valid (NACK if not)***;
6. The master sends a restart condition;
7. The master sends the slave address (7 bits) plus a read bit (HIGH);
8. If the slave address matches, the MAX21105 asserts an ACK on the data line;
9. The MAX21105 sends the content of the requested register (8 bits);
10. The master asserts a NACK on the data line;
11. The master generates a STOP condition.

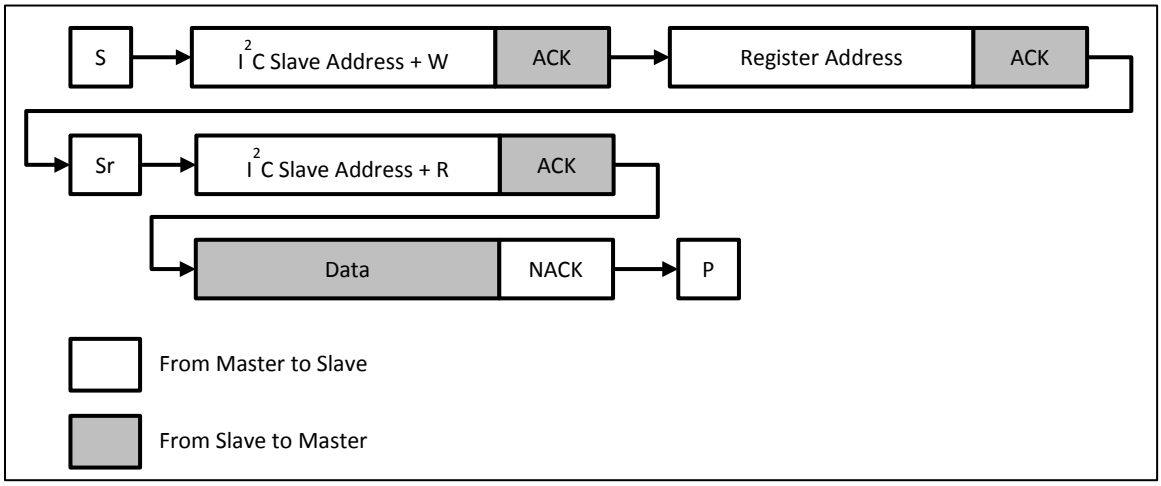


Figure 6: I²C Read One Byte

5.1.4 Read Multiple Registers

To execute a read of *n* consecutive registers, the following steps must be executed:

1. The master sends a START condition;
2. The master sends the slave address (7 bits) plus a write bit (LOW);
3. If the slave address matches, the MAX21105 asserts an ACK on the data line;
4. The master sends the Register Address of the first register to be read, setting the MS/Parity bit to 0 if it is configured as auto-increment bit (see *Register Address*);
5. The MAX21105 asserts an ACK on the data line ***only if the address is valid (NACK if not)***;
6. The master sends a restart condition;
7. The master sends the slave address (7 bits) plus a read bit (HIGH);
8. If the slave address matches, the MAX21105 asserts an ACK on the data line;
9. The MAX21105 sends the content of the requested register (8 bits);
10. The master asserts a ACK on the data line;
11. Repeat 9 and 10 *n-1* times;
12. The MAX21105 sends the content of the requested register (8 bits);
13. The master asserts a NACK on the data line;
14. The master generates a STOP condition.

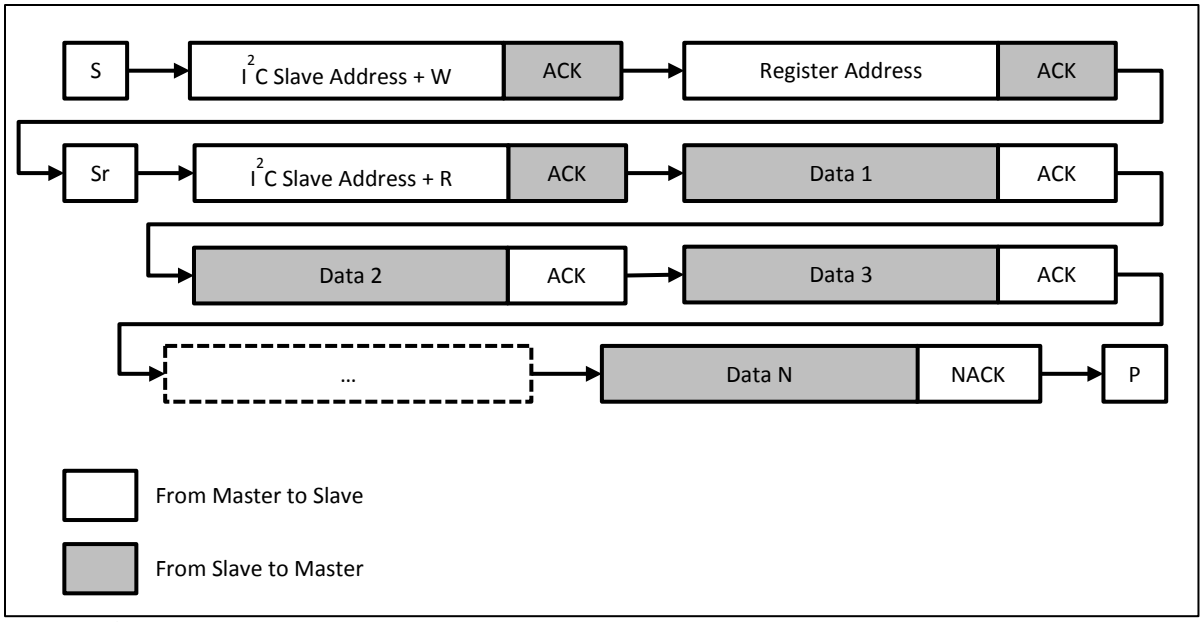


Figure 7: I²C Read a Burst of Data

5.2 Configuration Registers

Field	Description
slv_pu_dis	Enable/Disable the internal pullups of the slave pads.
strong_slave	Increase the I²C slave pads speed
if_setting	Select the master interface type
i2c_off	Turn on/off the I²C interface
if_parity	Sets the meaning of the bit 6 of the address during the communication from the master to the slave
parity_error	Indicate if parity error occurred
parity_rst	Trigger a reset of the parity

6 SPI Interface

The MAX21105 SPI interface can operate up to 10MHz, in both 3-wires (half duplex) and 4-wires mode (full duplex). It is recommended to set the I2C_OFF bit at address 0x16 if the MAX21105 is used together with other SPI devices to avoid the possibility to switch inadvertently into I^2C mode when traffic is detected with the CS un-asserted.

To connect a MAX21105 device to an SPI master, the CS pin of the MAX21105 device must be connected to the CSn pin of the SPI master, the SDA_SDI_O pin of the MAX21105 device must be connected to the MOSI pin of the SPI master, the SA0_SDO pin of the MAX21105 device must be connected to the MISO pin of the SPI master and the SCL_CLK pin of the MAX21105 device must be connected to the SCLK pin of the SPI master.

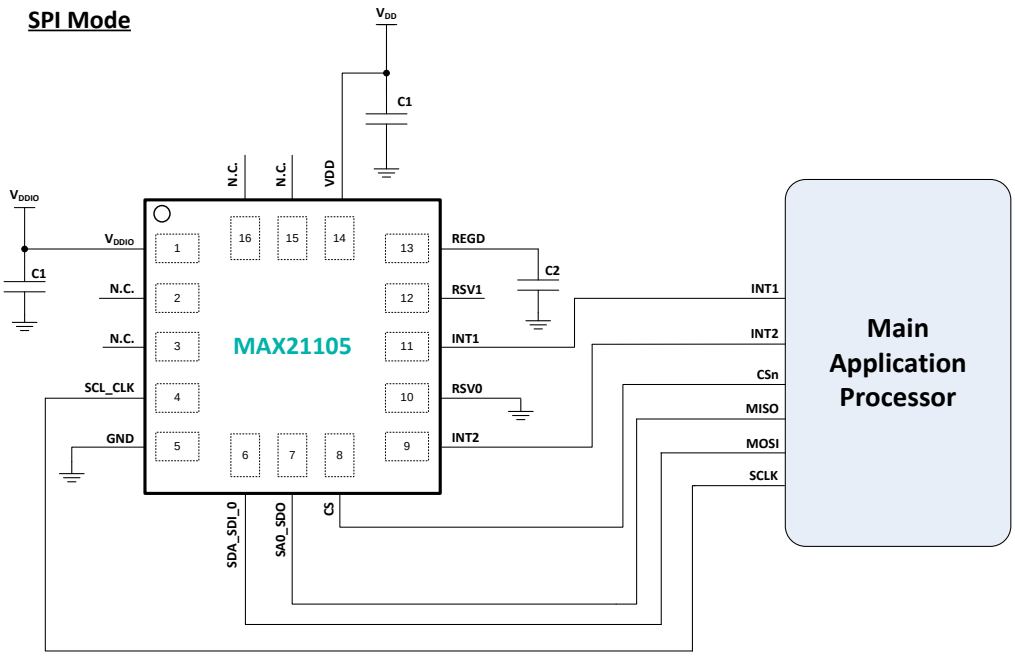


Figure 8: SPI Interface Connection to an Application Processor

Table 4: SPI External Component Properties

Component	Label	Specification	Quantity
V_{DDIO} /V_{DD} Bypass Capacitor	C1	Ceramic, X7R, 100nF ±10%, 4V	2
V_{DD} Bypass Capacitor	C2	Ceramic, X7R, 100nF ±10%, 2V	1

6.1 SPI Protocol

CSn is the serial port enable pin and is controlled by the SPI master. It goes LOW at the start of the transmission and returns to HIGH at the end.

SCLK is the serial port clock and is controlled by the SPI master. It is kept high when CSn is HIGH (no transmission). MISO and MOSI are, respectively, the serial port master data input and output. These lines are driven at the falling edge of SCLK and are sampled at the rising edge of SCLK.
Both the read register and write register commands are completed in 16 clock pulses, or in multiples of 8 in case of multiple read/write bytes.

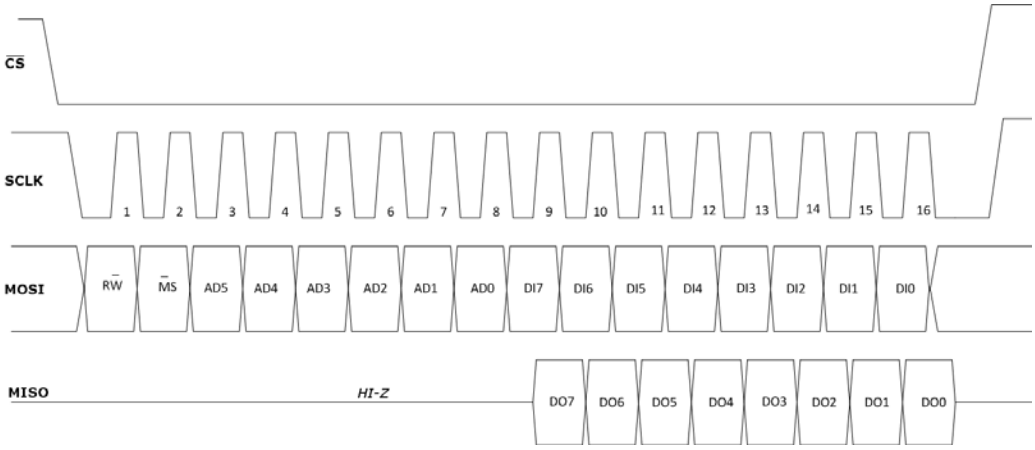


Figure 9: SPI Protocol

6.2 Register Address

The SPI register address for the MAX21105 is composed by 6 bits of address, 1 bit to select the direction of the operation ('1' to perform a read operation, '0' to perform a write operation) and 1 bit whose meaning can be configured as:

- **Auto-increment:** if 0, in case of burst operation the initial register address is auto-incremented after every data byte; if 1, the operation is executed always on the same register;
- **Even parity:** this bit represents the even parity computed on the 6 bits of the register address;
- **Odd parity:** this bit represents the odd parity computed on the 6 bits of the register address;

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R/W	MS/Parity	Address of the register you want to read or write					

Here are shown some examples:

Address	count of 1 bits	8 bits including parity	
		Even	Odd
000000 (0x00)	0	*x**0***000000	*x**1***000000
100000 (0x20)	1	*x**1***100000	*x**0***100000
100011 (0x23)	3	*x**1***100011	*x**0***100011
111111 (0x3F)	6	*x**0***111111	*x**1***111111

Bit 7, indicated with *x*, can be 0 or 1, depending on if you want to write or read the correspondent register. For further explanation:

- to **write** register **0x00**, using **odd parity,** you have to send from your MCU the byte '**01000000**' (0x40);

- to **read** register **0x20**, using **even parity,** you have to send from your MCU the byte '**11100000**' (0xE0);
- to **read** register **0x23**, using **odd parity,** you have to send from your MCU the byte '**10100011**' (0xA3);
- to **read** register **0x3F**, using **even parity,** you have to send from your MCU the byte '**10111111**' (0xBF).

If the parity mode is enabled, when the device receives the above bytes from the MCU, it will try to calculate the parity bit. The result of this computation is then reported to the user through a register field of the register map.

6.3 Configuration Registers

Field	Description
slv_pu_dis	Enable/Disable the internal pullups of the slave pads.
strong_slave	Increase the I^2C slave pads speed
if_setting	Select the master interface type
spi_3_wire	Set the interface type of the SPI
if_parity	Sets the meaning of the bit 6 of the address during the communication from the master to the slave
parity_error	Indicate if parity error occurred
parity_rst	Trigger a reset of the parity

7 Interrupts

The MAX21105 is equipped with an interrupt module to control a set of interrupt flags and two interrupt lines (INT1 and INT2).
This module allows to:

1. Configure the behavior of the interrupt lines (INT1 and INT2);
2. Map each interrupt flag to one of both the interrupt lines;
3. Create a conditional interrupt (Rate Interrupt) based on four different thresholds.

7.1 Interrupt Flags

The interrupt module provides several interrupt flags:

- **DATA_READY:** it is triggered when a new data on the active channels is available.
  It can be configured to work in one of the following modalities:
  - **ALL**: it is cleared after all the active channels are read. Data are not updated until the clear operation is accomplished.
  - **ANY**: it is cleared when at least a byte of one of the active channels is read. Data are updated independently from the clear operation.
  - **STATUS**: it is cleared when status register is read. Data are not updated until the clear operation is accomplished.
- **FIFO_EMPTY:** it is triggered when no new data on the FIFO queue is available;
- **FIFO_OVERRUN:** it is triggered when older unread FIFO data are lost;
- **FIFO_THRESHOLD:** it is triggered when the number of available data in FIFO exceeds a configured threshold (see the *FIFO* paragraph);
- **INT_AND:** See the *Rate Interrupts* paragraph
- **INT_OR:** See the *Rate Interrupts* paragraph
- **OTP_DOWNLOADING:** it is triggered when the OTP is being downloaded.

7.2 Interrupt Lines

An interrupt line is a dedicated pin where a notification to an external application processor can be provided. The MAX21105 is equipped with two interrupt lines that can be configured independently.
For each interrupt line, it is possible to:

- Enable/disable them;
- Set the active level to low;
- Set the output type to push-pull or open drain configuration.

To map an interrupt flag to an interrupt line it is necessary to enable it through the mask registers: by setting its corresponding bit to 1 on the bit-mask, the interrupt flag is mapped to the related interrupt line. The output of the two INT1 and INT2 interrupt lines is then computed by applying the OR operator to all the enabled interrupt flags contained in latched interrupt status register. Please note that the enable is not applied to the register, so it contains also the value of the interrupt flags that are not enabled.

An interrupt line can be configured in order to keep the status until the master requests to clear it by reading or writing the flag (latched). The lines can be also configured to auto-clear its status after a period of time where the duration to clear the interrupt is programmable (timeout). The only exception is the **DATA_READY** flag that is always unlatched regardless of the selected clearing mode.

7.3 Rate Interrupts

The *Rate Interrupt* allows generating an interrupt event when the gyroscope/accelerometer data falls inside a range of values defined by a set of thresholds. By setting the absolute value of the threshold (|TH|) for each axis, four ranges are defined:

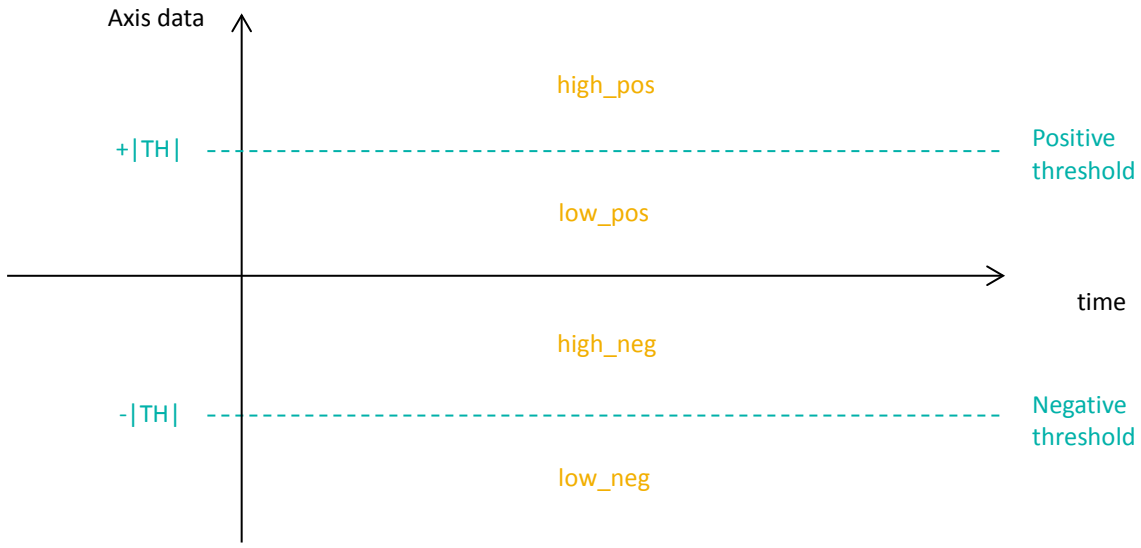


Figure 10: Rate Interrupt ranges

For each range, it is possible to select if it contributes to the generation of the *Rate Interrupt* for each axis. Then, the rate interrupts are computed as follow:

- **INT_AND:** Active if the defined conditions are satisfied for all the axes at the same time;
- **INT_OR:** Active if the defined conditions are satisfied for at least one of the axes.

The threshold absolute value can be set by writing the *INT_REF_X*, *INT_REF_Y* and *INT_REF_Z* registers. Those registers represent the most significant byte of the real threshold; so, to compute the actual absolute value of the threshold, the register value must be multiplied by 256. If the application requires a better resolution, it is possible to specify a 16-bit threshold by setting the *int_single_ref* register field; in this case, all the axes share the same threshold that is defined as the combination of *INT_REF_X* and *INT_REF_Y* where the first one is the most significant byte and the last one is the least significant byte of the threshold.
Once the thresholds are defined, the *INT_THS_X*, *INT_THS_Y* and *INT_THS_Z* permits to select which range contributes to the generation of the rate interrupts for each *axis* as follow:

- ***int_axis_high_pos_en*:** if enabled, the condition is satisfied if the data of the *axis* falls in the high_pos range;
- ***int_axis_high_neg_en*:** if enabled, the condition is satisfied if the data of the *axis* falls in the high_neg range;
- ***int_axis_low_pos_en*:** if enabled, the condition is satisfied if the data of the *axis* falls in the low_pos range;

- **int_*axis*_low_neg_en:** if enabled, the condition is satisfied if the data of the *axis* falls in the low_neg range.

Then, the desired axes must be enabled by setting to 1 the corresponding bit of the *int_mask_xyz_and* for the **INT_AND** and *int_mask_xyz_or* for the **INT_OR**.

Through the *INT_THS_X*, *INT_THS_Y* and *INT_THS_Z* registers it is also possible to read the current value of the conditions, regardless of the content of the *int_mask_xyz_or* and *int_mask_xyz_and* register fields. Each of these values can be configured as latched by setting the *int_freeze* register field. For the rate interrupts, it is possible also to define a debounce value by defining the number of samples the axis data has to satisfy the condition before asserting the corresponding interrupt. Those values can be set in the *INT_DEB_X*, *INT_DEB_Y* and *INT_DEB_Z* registers. If the required value is greater than 255, it is possible to define a 16-bit debounce value by setting *int_single_deb* register field; in this case, all the conditions share the same debounce value that is defined as the combination of *INT_DEB_X* and *INT_DEB_Y*, where the first one is the most significant byte and the last one is the least significant byte of the debounce value.

7.4 Configuration Registers

Function	Register/Field	Description
Pads	*int1_pd_en, int1_pu_en*	Enable/disable the internal pullup/down of the INT1 pad
	int2_pd_en, int2_pu_en	Enable/disable the internal pullup/down of the INT2 pad
	strong_int_aux	Increase the interrupt pads speed
Functional	*dr_rst_mode*	Set the DATA_READY interrupt reset and data-out update mode
	INT_CFG_2	INT1/INT2 generic configuration register
	INT_TMO_CFG	Interrupts latch mode and duration (timeout) configuration
	INT_STATUS_UL , INT_STS	Interrupt status (unlatched and latched)
	INT_MSK	Interrupt enables
	acc_intp_sel	Set the filtering mode for the accelerometer interrupt
	msk_acc_int_d_rdy, msk_gyr_int_d_rdy	Enable the gyroscope/accelerometer contribution to DATA_READY interrupt assertion
Rate Interrupt	*INT_REF_X, INT_REF_Y, INT_REF_Z*	Rate Interrupt thresholds for each axis
	INT_DEB_X, INT_DEB_Y, INT_DEB_Z	Rate Interrupt debounce value for each axis
	INT_THS_X, INT_THS_Y, INT_THS_Z	Rate Interrupt thresholds ranges enables and status registers for each axis
	int_freeze	Rate Interrupt latch configuration
	int_mask_xyz_and, int_mask_xyz_or	Axis enables for **INT_AND** and **INT_OR** respectively
	sns_intp_fsc	Gyroscope full-scale used by the Rate Interrupt
	sns_intp_hpf	Gyroscope HPF used by the Rate Interrupt
	int_single_ref	Enable a single 16-bits Rate Interrupt threshold for all the axes
	int_single_deb	Enable a single 16-bits Rate Interrupt debounce value for all the axes
	int_src_cfg	Select the Rate Interrupt source

8 Reading Data from MAX21105

MAX21105 sensor output data can be read by the host processor in two different mechanisms: synchronous (polling) and asynchronous (interrupt) reading methods.

8.1 Synchronous Reading

In synchronous reading, the host actively samples the status of the *gyro_dr* and *acc_dr* flags to check the status of the data output. When one (or both) of them is set a new data is ready to be read, so the host can read the new data related to the asserted flag from the output registers.

8.2 Asynchronous Reading

In asynchronous reading, the host does not need to continuously monitoring the status of the data output, because the data availability is notified through the interrupt line.

Only after the **DATA_READY** interrupt is asserted, the host can read the *gyro_dr* and *acc_dr* flags to verify which data are available and then read the corresponding output registers.

It is possible to configure which sensor (gyroscope or accelerometer) contributes to the **DATA_READY** interrupt assertion. If both the gyroscope and accelerometer are mapped to the **DATA_READY** signal, the interrupt is asserted when a new data of at least one sensor is available.

9 FIFO

The MAX21105 embeds a 512-byte data FIFO. This allows a consistent power saving for the system since the host processor does not need to continuously extract the data from the sensor, but it can wake up only when needed and burst the data out from the FIFO.

The data order in FIFO depends on the endian setting:

- **Big Endian:** *GYRO_X_H, GYRO_X_L, GYRO_Y_H, GYRO_Y_L, GYRO_Z_H, GYRO_Z_L, ACC_X_H, ACC_X_L, ACC_Y_H, ACC_Y_L, ACC_Z_H, ACC_Z_L*
- **Little Endian:** *GYRO_X_L, GYRO_X_H, GYRO_Y_L, GYRO_Y_H, GYRO_Z_L, GYRO_Z_H ACC_X_L, ACC_X_H, ACC_Y_L, ACC_Y_H, ACC_Z_L, ACC_Z_H*

9.1 FIFO Data Storage Selection

The selection of data to be stored in the FIFO can be configured in two ways: statically and power-mode dependent.

In static mode, the FIFO data storage enable must be coherent with the power-mode and in this configuration the power-mode cannot be changed when the FIFO is running. Changing the power-mode during the FIFO operation or selecting a power-mode that is not coherent with the FIFO data storage enable could break the FIFO functionality.

In power-mode dependent configuration, the actual FIFO data that will be put in FIFO depends on both the power-mode and the FIFO data storage enable. In particular, the FIFO stores a sensor only if this one is powered-on and its data storage in FIFO is enabled. In order to distinguish FIFO packets that are stored in different power-modes, a 3-words reserved token is inserted in FIFO after a power-mode transition. A token notifies which power-mode transition occurred, according to the following table:

Table 5: Power Mode Transition Token Codes

Power mode transition		Reserved Token Codes		
from	to	1st word	2nd word	3rd word
Gyroscope + Accelerometer	Accelerometer only	0x8000	0x0000	0x0001
Gyroscope + Accelerometer	Gyroscope only	0x8000	0x0000	0x0002
Accelerometer only	Gyroscope + Accelerometer	0x8000	0x0000	0x0003
Gyroscope only	Gyroscope + Accelerometer	0x8000	0x0000	0x0004
Accelerometer only	Gyroscope only	0x8000	0x0000	0x0005
Gyroscope only	Accelerometer only	0x8000	0x0000	0x0006

In order to avoid ambiguities, the special value 0x8000 is kept reserved, so it is never produces as sensor data value.

Figure 11 shows an example of transition from *Acc Low-Power* to *Acc Low-Noise + Gyro Low-Power*:

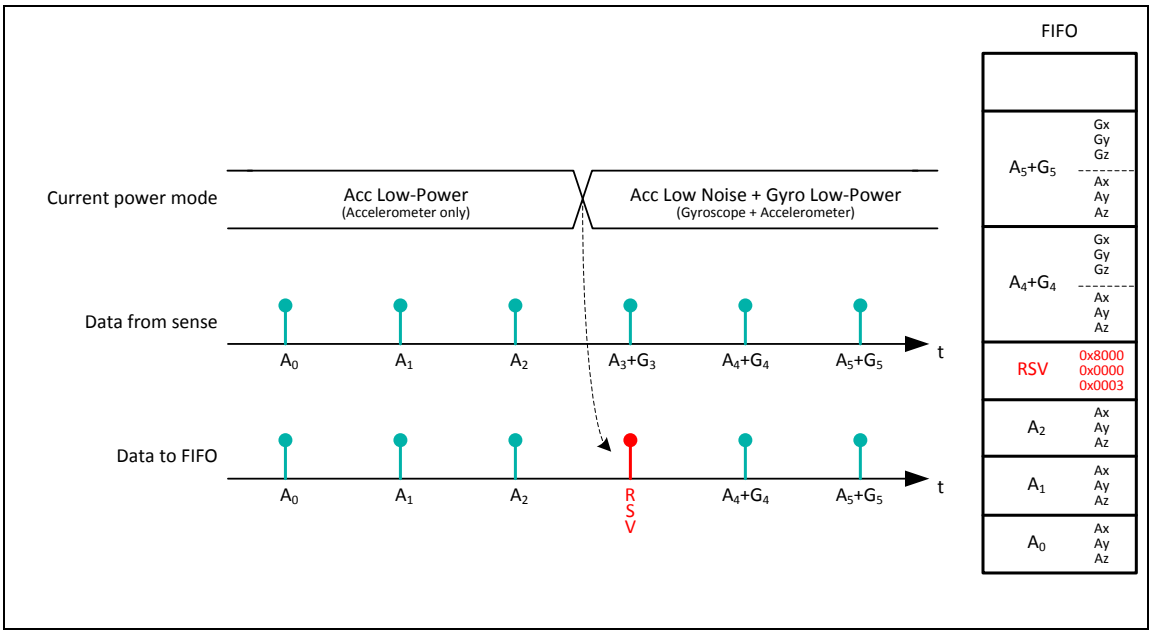


Figure 11: FIFO Power Mode Transition Example

9.2 FIFO Modes Description

The FIFO buffer can work according to four main different modes: off, normal, interrupt and snapshot. When configured in Snapshot mode, it offers the ideal mechanism to capture the data following a Rate Interrupt event.
Both normal and Interrupt modes can be optionally configured to operate in overrun Mode, depending on whether, in case of buffer under-run, newer or older data are lost.
Various FIFO status flags can be enabled to generate interrupt events on INT1/INT2 pins.

9.2.1 FIFO OFF Mode

In this mode, the FIFO is turned off; data are stored only in the data registers. No data are available from FIFO if read. When the FIFO is turned OFF, there are two options to use the device: synchronous or asynchronous reading.

9.2.2 Normal Mode

The behavior of the FIFO in Normal mode varies depending on the **Overrun** setting.
The following paragraphs show a description of the behavior for both settings of the overrun. For the next sections, the following descriptions are useful for the user's understanding of FIFO:

- **Write Pointer (WP):** Write pointer is the address number where the next data will be written in FIFO, and whenever there is a new data is written, the write pointer is incremented by 1;
- **Read Pointer (RP):** Read pointer is the address number from where the first data in FIFO is read, and whenever there is a new data is read, the read pointer is incremented by 1;
- **Level:** Level tells the difference between Write pointer and Read pointer, which also gives you the total number of data available in FIFO.

9.2.2.1 Stop on Full

1. FIFO is turned on;
2. FIFO is filled with the data at the selected Output Data Rate (ODR);
3. When FIFO is full, an interrupt can be generated;
4. When FIFO is full, all the new incoming data will be discharged. Reading only a subset of the data already stored into the FIFO keeps locked the possibility for new data to be written;
5. Only if all the data are read the FIFO will restart saving data. If the communication speed is high, data loss can be prevented;
6. In order to prevent a FIFO full condition, the required condition is to complete the reading of the data set before the next **DATA_READY** occurs;
7. If this condition is not guaranteed, data can be lost.

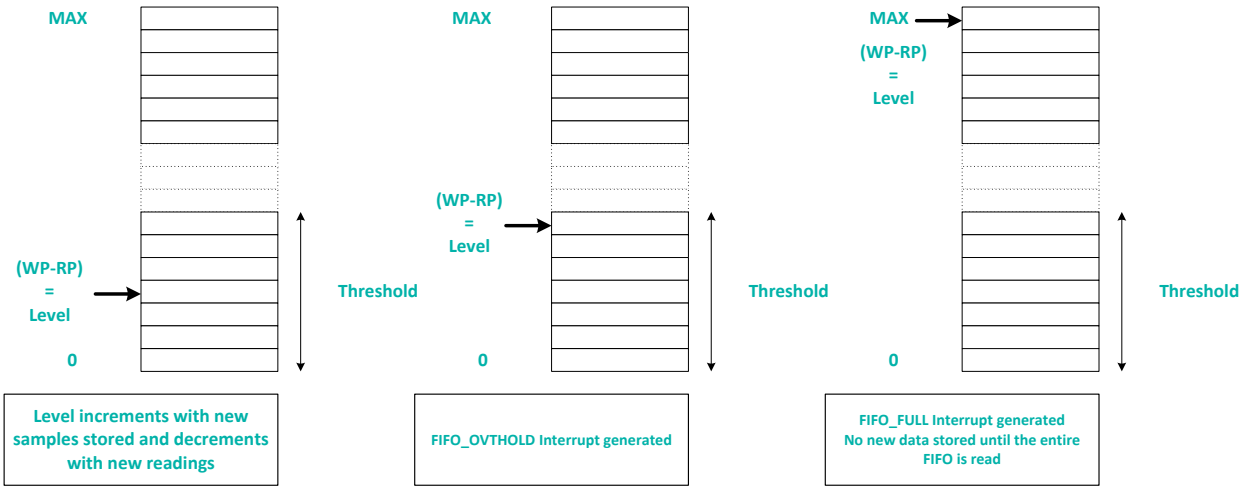


Figure 12: FIFO Normal Mode, Stop on Full

9.2.2.2 Overwrite

1. FIFO is turned on;
2. FIFO is filled with the data at the selected ODR;
3. When FIFO is full, an interrupt can be generated;
4. When FIFO is full, the oldest data will be overwritten with the new ones;
5. If communication speed is high, data integrity can be preserved;
6. In order to prevent a data lost condition, the requirement is to complete the reading of the data set before the next **DATA_READY** occurs;
7. If this condition is not guaranteed, data can be overwritten;
8. When an overrun condition occurs the Reading pointer is forced to Writing Pointer -1, to make sure only older data are discarded and newer data have a chance to be read.

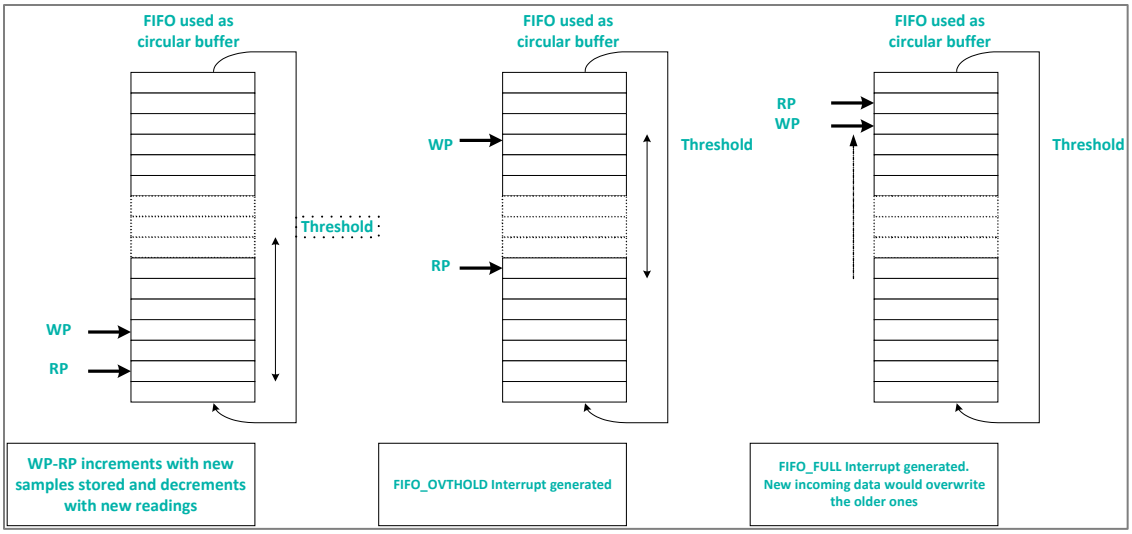


Figure 13: FIFO Normal Mode, Overwrite

9.2.3 Interrupt Mode

The behavior of the FIFO in Interrupt mode varies depending on the **Overrun** setting. The following paragraphs show a description of the behavior for both settings of the overrun.

9.2.3.1 Stop on Full

1. FIFO is initially disabled. Data is stored only in the data registers;
2. When a Rate Interrupt (either **INT_OR** or **INT_AND**) is generated, the FIFO is turned on automatically. It stores the data at the selected ODR.
3. When FIFO is full, all the new incoming data will be discharged. Reading only a subset of the data already stored into the FIFO keeps locked the possibility for new data to be written;
4. Only if all the data are read the FIFO will restart saving data;
5. If communication speed is high, data loss can be prevented;
6. In order to prevent a FIFO full condition, the required condition is to complete the reading of the data set before the next **DATA_READY** occurs;
7. If this condition is not guaranteed, data can be lost.

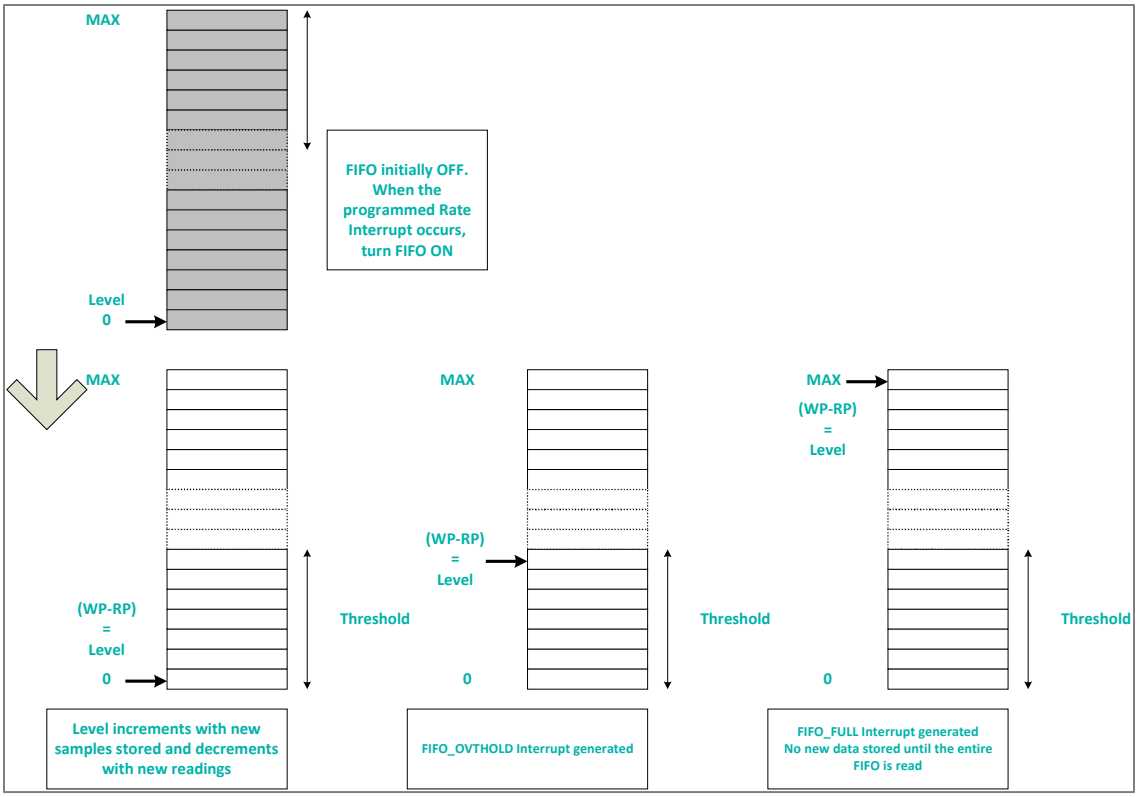


Figure 14: FIFO Interrupt Mode, Stop on Full

9.2.3.2 Overwrite

1. FIFO is initially disabled. Data is stored only in the data registers;
2. When a Rate Interrupt (either **INT_OR** or **INT_AND**) is generated, the FIFO is turned on automatically. It stores the data at the selected ODR;
3. When FIFO is full, an interrupt can be generated;
4. When FIFO is full, the oldest data will be overwritten with the new ones;
5. If communication speed is high, data integrity can be preserved;
6. In order to prevent a data lost condition, the required condition is to complete the reading of the data set before the next **DATA_READY** occurs;
7. If this condition is not guaranteed, data can be overwritten;
8. When an overrun condition occurs the Reading pointer is forced to Writing Pointer -1, to make sure only older data are discarded and newer data have a chance to be read.

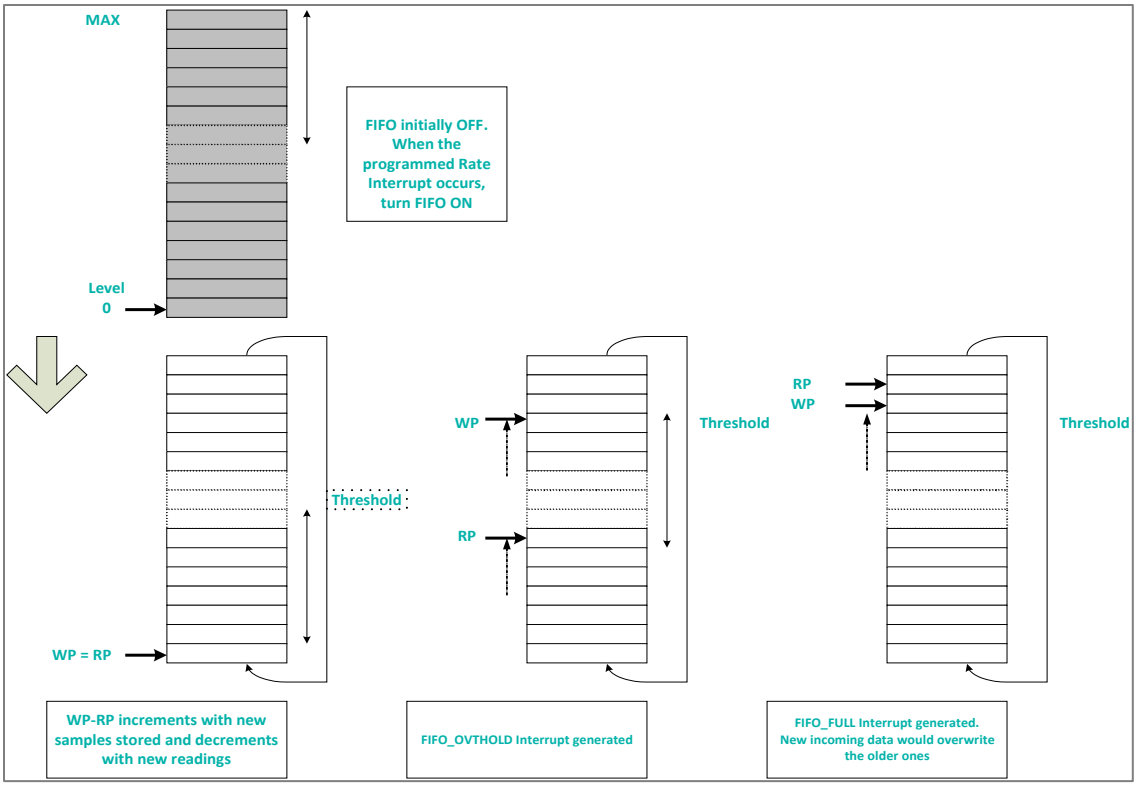


Figure 15: FIFO Interrupt Mode, Overwrite

9.2.4 Snapshot Mode

1. FIFO is initially in normal mode with overrun enabled;
2. When a Rate Interrupt (either **INT_OR** or **INT_AND**) is generated, the FIFO switches automatically to not-overrun mode. It stores the data at the selected ODR until the FIFO becomes full;
3. When FIFO is full, an interrupt can be generated;
4. When FIFO is full, all the new incoming data will be discharged. Reading only a subset of the data already stored into the FIFO keeps locked the possibility for new data to be written;
5. Only if all the data are read the FIFO will restart saving data;
6. If communication speed is high, data loss can be prevented;
7. In order to prevent a FIFO full condition, the required condition is to complete the reading of the data set before the next **DATA_READY** occurs;
8. If this condition is not guaranteed, data can be lost.

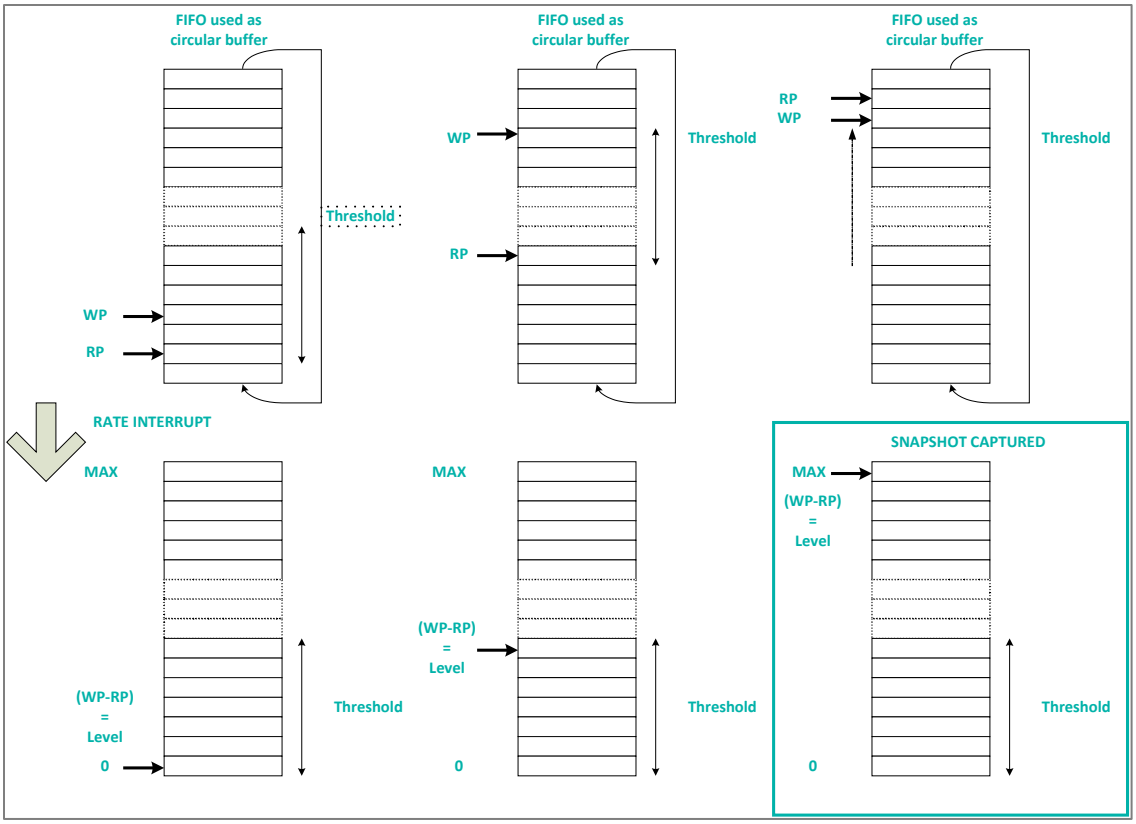


Figure 16: FIFO Snapshot Mode

9.3 Example of FIFO Read/Write Pointers Evolution

The following drawing assumes:

- Reading frequency roughly twice the writing frequency (ODR);
- FIFO threshold = *Threshold*;
- FIFO is in normal mode.

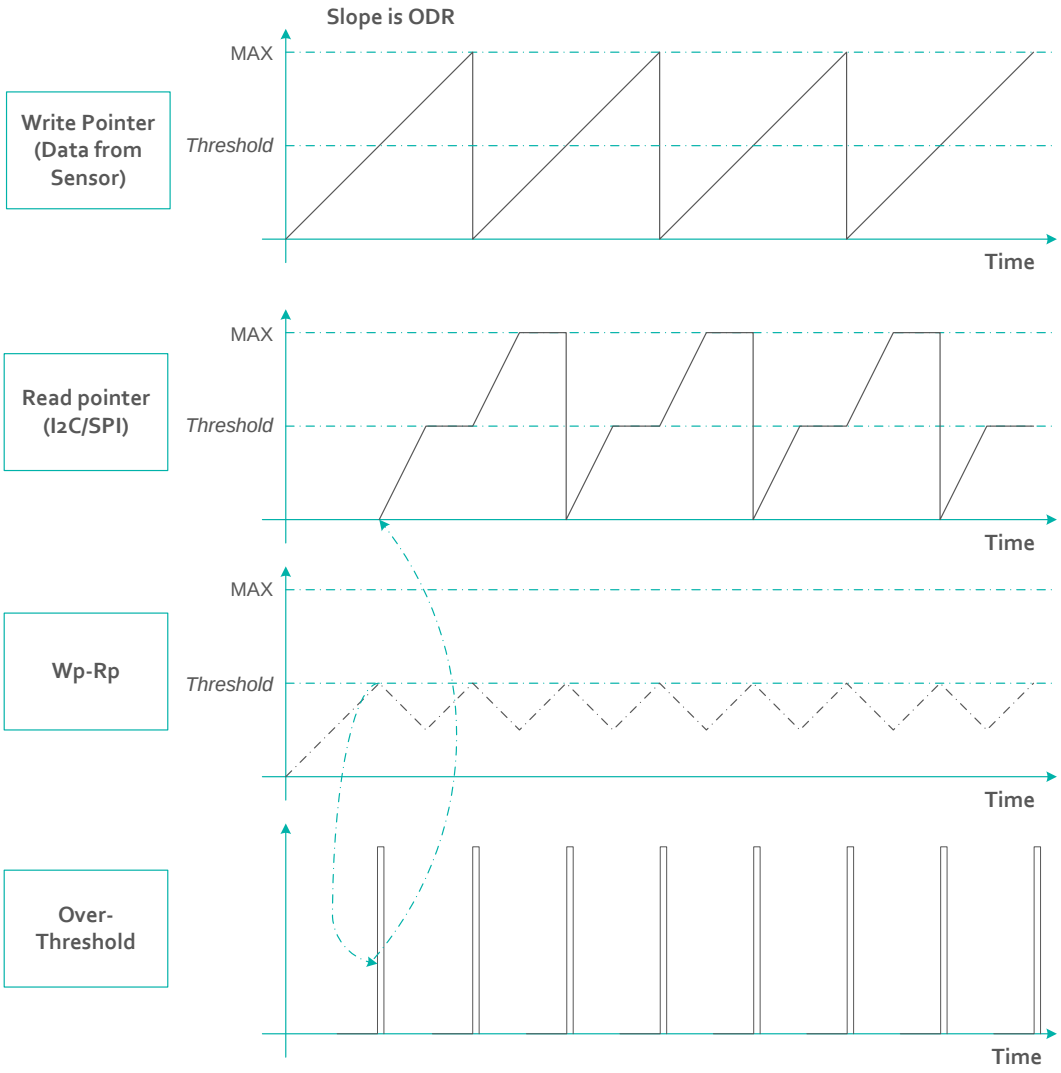


Figure 17: FIFO Read/Write Pointer Evolution

9.4 Configuration Registers

Register/Field	Description
fifo_auto_store_pwr_en	Enable the power-dependent FIFO data selection
fifo_store_gyr, fifo_store_acc	Enable the data storage of gyroscope/accelerometer data in FIFO
FIFO_THS	FIFO threshold
FIFO_CFG	FIFO configuration
sts_ul_fifo_empty, sts_ul_fifo_overrun, sts_ul_fifo_ths, sts_i1_fifo_empty, sts_i1_fifo_overrun, sts_i1_fifo_ths	FIFO status interrupt
msk_i1_fifo_empty, msk_i1_fifo_overrun, msk_i1_fifo_ths	FIFO interrupt enables
FIFO_COUNT	Number of FIFO words available on FIFO
FIFO_STATUS	FIFO status register
FIFO_DATA	FIFO data register

10 Programming Example

This chapter shows some examples on how to execute some operations on the device.
Three functions are defined to abstract the SPI/I^2C communication:

- **Write(<Register Address>, <Value>):**
  Writes the <Value> to the register at the <Register Address>;
- ***Read(<Register Address>)*:**
  Returns the value of the register at the *<Register Address>*;
- ***ReadBurstNoInc(<Register Address>, <count>)*:**
  Returns an array of *count* elements with the result of a burst read with no auto-increment on *<Register Address>* register;
- ***ReadBurstInc(<Start Address>, <count>)*:**
  Returns an array of *count* elements with the result of a burst read with auto-increment, starting from *<Start Address>* register address;

10.1 Simple Read-Out Sequence, No FIFO, No Interrupts

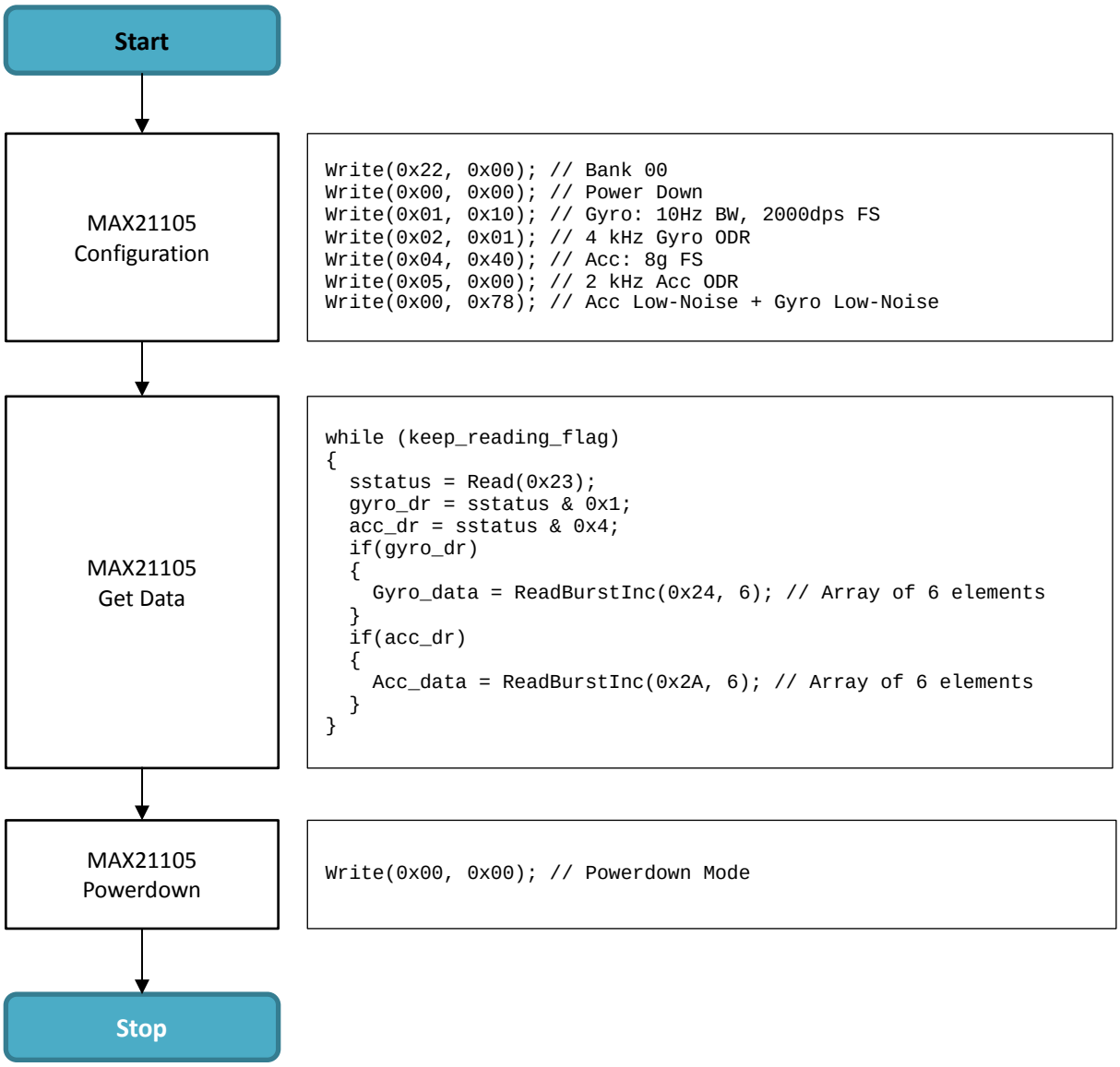


Figure 18: Simple Read-Out Sequence, No FIFO, No Interrupts

10.2 Simple Read-Out Sequence, FIFO Normal Mode, No Interrupts

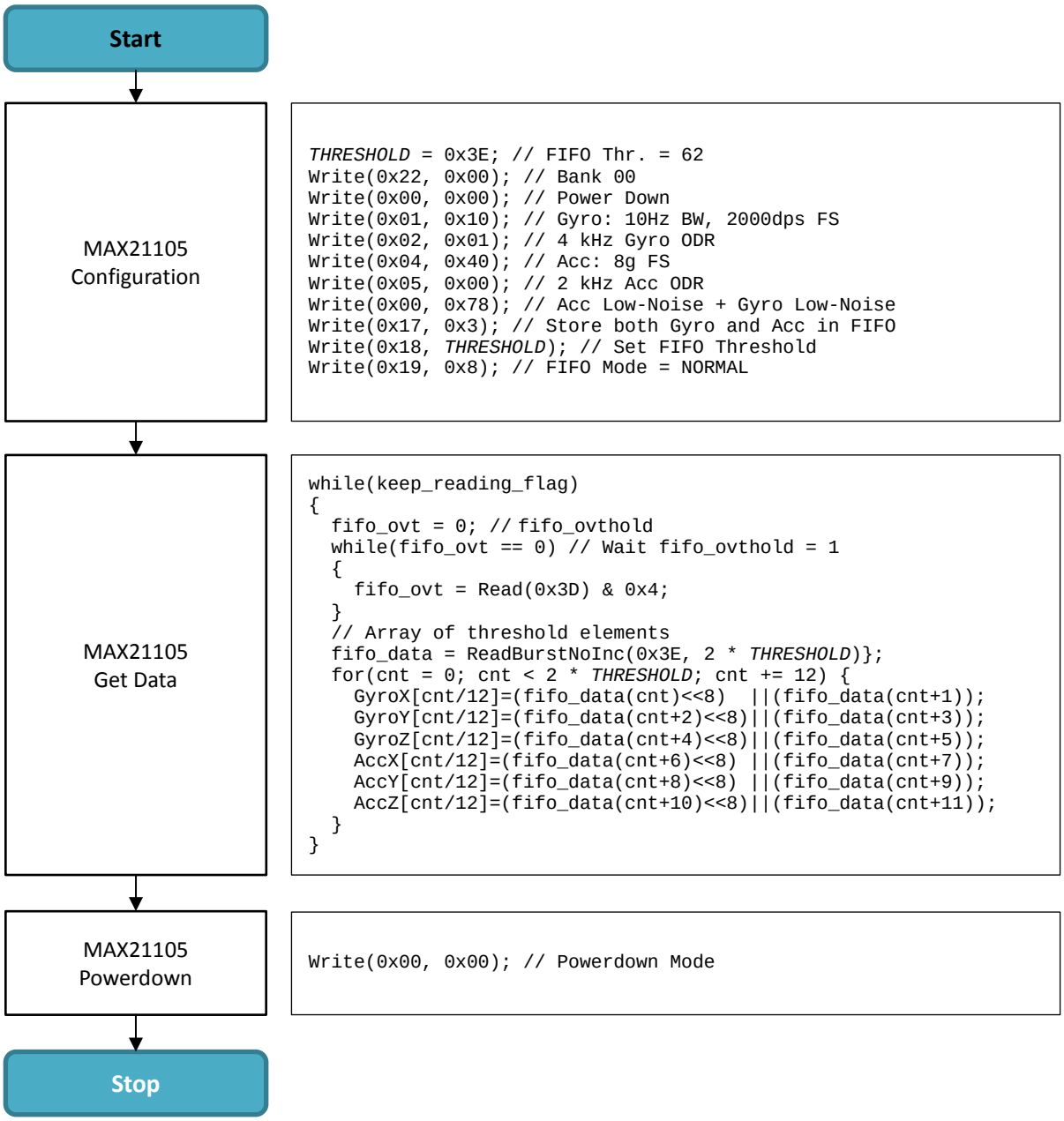


Figure 19: Simple Read-Out Sequence, FIFO Normal, No Interrupts

10.3 Simple Read-Out Sequence, Normal Mode, Data Ready Interrupts

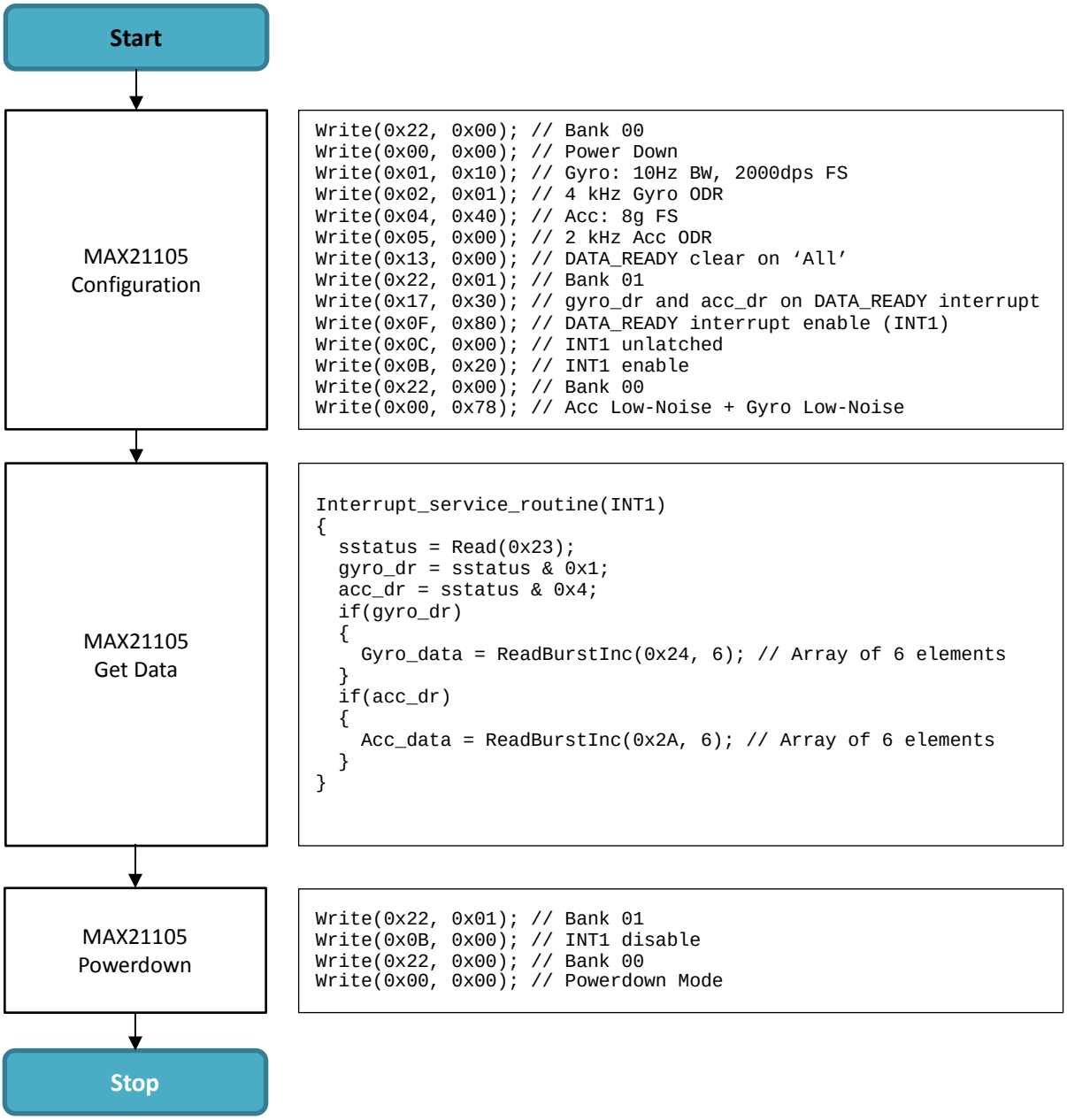


Figure 20: Simple Read-Out Sequence, Normal Mode, w/ Data Ready Interrupts and No FIFO

11 Register Map

The following paragraphs will illustrate the register map of the device. When not specified differently, register/field values are expressed in binary format.

11.1 Register Overview

11.1.1 Common

Name	Address	Access	Default	Description
WHO_AM_I	0x20	R	10110100	Identifier of the product family.
SILICON_REV_OTP	0x21	R	00000101	Identifier of the silicon revision.
EXT_STATUS	0x22	R/W	00000000	Temperature data ready and error flags and bank selection.
SYSTEM_STATUS	0x23	R	00000000	Gyroscope and accelerometer data ready and error flags.
GYRO_X_H	0x24	R	00000000	Gyroscope X MSB.
GYRO_X_L	0x25	R	00000000	Gyroscope X LSB.
GYRO_Y_H	0x26	R	00000000	Gyroscope Y MSB.
GYRO_Y_L	0x27	R	00000000	Gyroscope Y LSB.
GYRO_Z_H	0x28	R	00000000	Gyroscope Z MSB.
GYRO_Z_L	0x29	R	00000000	Gyroscope Z LSB.
ACCE_X_H	0x2A	R	00000000	Accelerometer X MSB.
ACCE_X_L	0x2B	R	00000000	Accelerometer X LSB.
ACCE_Y_H	0x2C	R	00000000	Accelerometer Y MSB.
ACCE_Y_L	0x2D	R	00000000	Accelerometer Y LSB.
ACCE_Z_H	0x2E	R	00000000	Accelerometer Z MSB.
ACCE_Z_L	0x2F	R	00000000	Accelerometer Z LSB.
TEMP_H	0x30	R	00000000	Temperature MSB.
TEMP_L	0x31	R	00000000	Temperature LSB.
TRM_BNK_REG	0x38	R/W	00000000	Trimming banks enable.
FIFO_COUNT	0x3C	R	00000000	Available FIFO data.
FIFO_STATUS	0x3D	R	00000000	FIFO status.
FIFO_DATA	0x3E	R/W	00000000	FIFO data register.
RST_REG	0x3F	R/W	00000000	Reset register.

11.1.2 Bank 00

Name	Address	Access	Default	Description
SET_PWR	0x00	R/W	00000000	Power Modes.
SNS_CFG_1	0x01	R/W	00101000	Gyroscope self-test, full-scale and LPF frequency.
SNS_CFG_2	0x02	R/W	00000100	Gyroscope OIS enable, ODR and HPF enable.
SNS_CFG_3	0x03	R/W	00000000	Gyroscope HPF frequency and wait condition before generating the data.
SET_ACC_PWR	0x04	R/W	11000000	Accelerometer self-test and full-scale.
ACC_CFG_1	0x05	R/W	00000010	Accelerometer ODR and LPF/HPF frequencies.
ACC_CFG_2	0x06	R/W	01000000	Accelerometer output filter, signal averages in Low-Power mode and HPF input's filter.
SET_TEMP_DR	0x13	R/W	00000001	DATA_READY reset and temperature sensor settings.
SET_PU_PD_PAD	0x14	R/W	00000000	Pullup/down connections.
SET_I2C_PAD	0x15	R/W	00000100	PAD speed.
MIF_CFG	0x16	R/W	00000000	Master communication interface settings and status.
FIFO_STORE	0x17	R/W	00000000	FIFO configuration.
FIFO_THS	0x18	R/W	00000000	FIFO threshold.
FIFO_CFG	0x19	R/W	00000000	FIFO configuration.
OTP_STS_CFG	0x1C	R/W	10000000	Interface parity setting, general status and errors.

11.1.3 Bank 01

Name	Address	Access	Default	Description
INT_REF_X	0x00	R/W	00000000	MSB of the reference for X-axis interrupt.
INT_REF_Y	0x01	R/W	00000000	MSB of the reference for Y-axis interrupt.
INT_REF_Z	0x02	R/W	00000000	MSB of the reference for Z-axis interrupt.
INT_DEB_X	0x03	R/W	00000000	Rate Interrupt debounce (X-axis).
INT_DEB_Y	0x04	R/W	00000000	Rate Interrupt debounce (Y-axis).
INT_DEB_Z	0x05	R/W	00000000	Rate Interrupt debounce (Z-axis).
INT_THS_X	0x06	R/W	00000000	Signal threshold condition and mask for X-axis.
INT_THS_Y	0x07	R/W	00000000	Signal threshold condition and mask for Y-axis.
INT_THS_Z	0x08	R/W	00000000	Signal threshold condition and mask for Z-axis.
INT_COND	0x09	R/W	00000000	Rate Interrupt settings.
INT_CFG_1	0x0A	R/W	00000000	Interrupt configuration register #1.
INT_CFG_2	0x0B	R/W	00000000	Interrupt configuration register #2.
INT_TMO_CFG	0x0C	R/W	00000000	Interrupts latch mode and duration.
INT_STATUS_UL	0x0D	R	00000000	Unlatched interrupts status bits.
INT_STS	0x0E	R	00000000	Latched status bits.
INT_MSK	0x0F	R/W	00000010	Enable the interrupt status.
INT_SRC_CFG	0x17	R/W	00100000	DATA_READY interrupt configuration, Rate Interrupt source selection and accelerometer filters.
SERIAL_0	0x1A	R	00000000	Serial number #0.
SERIAL_1	0x1B	R	00000000	Serial number #1.
SERIAL_2	0x1C	R	00000000	Serial number #2.
SERIAL_3	0x1D	R	00000000	Serial number #3.
SERIAL_4	0x1E	R	00000000	Serial number #4.
SERIAL_5	0x1F	R	00000000	Serial number #5.

11.1.4 Bank 02

Name	Address	Access	Default	Description
BIAS_COMP_GX_MSB	0x13	R/W	00000000	Bias compensation gyroscope X (MSB).
BIAS_COMP_GX_LSB	0x14	R/W	00000000	Bias compensation gyroscope X (LSB).
BIAS_COMP_GY_MSB	0x15	R/W	00000000	Bias compensation gyroscope Y (MSB).
BIAS_COMP_GY_LSB	0x16	R/W	00000000	Bias compensation gyroscope Y (LSB).
BIAS_COMP_GZ_MSB	0x17	R/W	00000000	Bias compensation gyroscope Z (MSB).
BIAS_COMP_GZ_LSB	0x18	R/W	00000000	Bias compensation gyroscope Z (LSB).
BIAS_COMP_AX	0x19	R/W	00000000	Bias compensation accelerometer X.
BIAS_COMP_AY	0x1A	R/W	00000000	Bias compensation accelerometer Y.
BIAS_COMP_AZ	0x1B	R/W	00000000	Bias compensation accelerometer Z.
GYR_ODR_TRIM	0x1F	R	00000000	Sensors status.

11.2 Registers Description

11.2.1 Common

WHO_AM_I

Address	Common - 0x20 (Hex) - 32 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	10110100							

Description

Identifier of the product family.

Back to Register Overview

SILICON_REV_OTP

Address	Common - 0x21 (Hex) - 33 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000101							

Description

Identifier of the silicon revision.

Back to Register Overview

EXT_STATUS

Address	Common - 0x22 (Hex) - 34 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU		***temp_err***	***temp_dr***	***bank_sel***			
Access	R		R	R	R/W			
Default	-		0	0	0000			

Description

Temperature data ready and error flags and bank selection.

Fields

temp_err: Temperature output error: it is set when a new temperature data is generated before or during data reading.

temp_dr: Temperature output available: it is set when a new temperature data is available.

bank_sel: Bank selection: it allows addressing 16 different pages of registers, other than the common bank. Pages are 32 bytes sized.

0000: Bank 0
0001: Bank 1
0010: Bank 2

Back to Register Overview

SYSTEM_STATUS

Address	Common - 0x23 (Hex) - 35 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU				*acc_err*	*acc_dr*	*gyro_err*	*gyro_dr*
Access	R				R	R	R	R
Default	-				0	0	0	0

Description

Gyroscope and accelerometer data ready and error flags.

Fields

acc_err: Accelerometer output error: it goes high when a new accelerometer data is generated before or during data reading.

acc_dr: Accelerometer output available: it goes high when a new set of accelerometer data is available.

gyro_err: Gyroscope output error: it goes high when a new gyroscope data is generated before or during data reading.

gyro_dr: Gyroscope output available: it goes high when a new set of gyroscope data is available.

Back to Register Overview

GYRO_X_H

Address	Common - 0x24 (Hex) - 36 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000000							

Description

Gyroscope X MSB.
Most significant byte of the X-axis gyroscope data.

Back to Register Overview

GYRO_X_L

Address	Common - 0x25 (Hex) - 37 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000000							

Description

Gyroscope X LSB.
Least significant byte of the X-axis gyroscope data.

Back to Register Overview

GYRO_Y_H

Address	Common - 0x26 (Hex) - 38 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000000							

Description

Gyroscope Y MSB.
Most significant byte of the Y-axis gyroscope data.

Back to Register Overview

GYRO_Y_L

Address	Common - 0x27 (Hex) - 39 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000000							

Description

Gyroscope Y LSB.
Least significant byte of the Y-axis gyroscope data.

Back to Register Overview

GYRO_Z_H

Address	Common - 0x28 (Hex) - 40 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000000							

Description

Gyroscope Z MSB.
Most significant byte of the Z-axis gyroscope data.

Back to Register Overview

GYRO_Z_L

Address	Common - 0x29 (Hex) - 41 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000000							

Description

Gyroscope Z LSB.
Least significant byte of the Z-axis gyroscope data.

Back to Register Overview

ACCE_X_H

Address	Common - 0x2A (Hex) - 42 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000000							

Description

Accelerometer X MSB.
Most significant byte of the X-axis accelerometer data.

Back to Register Overview

ACCE_X_L

Address	Common - 0x2B (Hex) - 43 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000000							

Description

Accelerometer X LSB.
Least significant byte of the X-axis accelerometer data.

Back to Register Overview

ACCE_Y_H

Address	Common - 0x2C (Hex) - 44 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000000							

Description

Accelerometer Y MSB.
Most significant byte of the Y-axis accelerometer data.

Back to Register Overview

ACCE_Y_L

Address	Common - 0x2D (Hex) - 45 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000000							

Description

Accelerometer Y LSB.
Least significant byte of the Y-axis accelerometer data.

Back to Register Overview

ACCE_Z_H

Address	Common - 0x2E (Hex) - 46 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000000							

Description

Accelerometer Z MSB.
Most significant byte of the Z-axis accelerometer data.

Back to Register Overview

ACCE_Z_L

Address	Common - 0x2F (Hex) - 47 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000000							

Description

Accelerometer Z LSB.
Least significant byte of the Z-axis accelerometer data.

Back to Register Overview

TEMP_H

Address	Common - 0x30 (Hex) - 48 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000000							

Description

Temperature MSB.
Most significant byte of the temperature data.

Back to Register Overview

TEMP_L

Address	Common - 0x31 (Hex) - 49 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000000							

Description

Temperature LSB.
Least significant byte of the temperature data.

Back to Register Overview

TRM_BNK_REG

Address	Common - 0x38 (Hex) - 56 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU						***trim_bank_en***	RFU
Access	R						R	R
Default	-						0	-

Description

Trimming banks enable.

Fields

trim_bank_en: Enable read/write operation on banks greater than two. If TM pad is set, this bit is set automatically.

0: Disable
1: Enable

Back to Register Overview

FIFO_COUNT

Address	Common - 0x3C (Hex) - 60 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000000							

Description

Available FIFO data.
This register contains the number of FIFO words available on FIFO.

Back to Register Overview

FIFO_STATUS

Address	Common - 0x3D (Hex) - 61 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU			*fifo_data_lost*	*fifo_read_empty*	*fifo_ovthold*	*fifo_full*	*fifo_empty*
Access	R			R	R	R	R	R
Default	-			0	0	0	0	0

Description

FIFO status.
This register contains information about the FIFO.

Fields

fifo_data_lost: Indicates if at least one data has been lost when the FIFO was full.

fifo_read_empty: Indicates if a read occurred when the FIFO was empty.

fifo_ovthold: Indicates if the number of data in FIFO exceeds the threshold

fifo_full: Indicates if the FIFO is full.

fifo_empty: Indicates if the FIFO is empty.

Back to Register Overview

FIFO_DATA

Address	Common - 0x3E (Hex) - 62 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R/W							
Default	00000000							

Description

FIFO data register.
Burst reading auto-increment the FIFO read pointer.

Back to Register Overview

RST_REG

Address	Common - 0x3F (Hex) - 63 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU				***sw_rst***	***parity_rst***	***hpf_acc_rst***	***hpf_gyro_rst***
Access	R				R/W	R/W	R/W	R/W
Default	-				0	0	0	0

Description

Reset register.

Fields

sw_rst: Trigger a reset of the user registers.

0: Idle
1: Trigger

parity_rst: Trigger a reset of the parity.

0: Idle
1: Trigger

hpf_acc_rst: Set the accelerometer HP filter and restore the output to baseline.

0: Idle
1: Trigger

hpf_gyro_rst: Set the gyroscope HP filter and restore the output to baseline.

0: Idle
1: Trigger

Back to Register Overview

11.2.2 Bank 00

SET_PWR

Address	Bank 00 - 0x00 (Hex) - 0 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU	*pwr_mode*				RFU		
Access	R	R/W				R		
Default	-	0000				-		

Description

Power Modes.

Fields

pwr_mode: The MAX21105 features nine power modes, allowing selecting the appropriate tradeoff between power consumption, noise level, accuracy and turn-on time. The transition between different power modes can be controlled with the software by explicitly setting a power mode in the Configuration register.

Value	Mnemonic	Description
0000	Power-Down	In Power-Down Mode, the IC is configured to minimize the power consumption. In Power-Down Mode, registers can still be read and written, but neither sensor can generate new data. Compared to Standby Mode, it takes longer to activate the IC and start collecting data from the sensors.
0001	Gyro Standby	To reduce power consumption and have a shorter turn-on time, the IC features a standby mode for the gyroscope. In standby mode, the MAX21105 gyroscope does not generate data because a significant portion of the signal processing resources is turned off to save power. Still, this mode enables a much quicker turn-on time.
0010	Gyro Low-Power	In this power mode, only the gyroscope is switched on and it is operating in Low-Power mode. The Low-Power mode reduces power consumption with the same sensor accuracy at the price of a higher rate noise density. This feature can be activated for the gyroscope with different ODR from 5Hz to 200Hz.
0011	Gyro Low-Noise	In this power mode, only the gyroscope is switched on and it is operational with minimum noise level.
1000	Acc Low-Power	In this power mode, only the accelerometer is switched on, and it is operating in Low-Power mode. The Low-Power mode reduces power consumption with the same sensor accuracy at the price of a higher accelerometer noise density. This feature can be activated for accelerometer with different ODR from 5Hz to 400Hz.

1100	Acc Low-Noise	In this power mode, only the accelerometer is switched on. It is operational with minimum noise level.
1101	Acc Low-Noise + Gyro Standby	This power mode is the combination between the Acc Low-Noise and the Gyro Standby power modes.
1110	Acc Low-Noise + Gyro Low-Power	This power mode is the combination between the Acc Low-Noise and the Gyro Low-Power power modes.
1111	Acc Low-Noise + Gyro Low-Noise	This power mode is the combination between the Acc Low-Noise and the Gyro Low-Noise power modes.

Back to Register Overview

SNS_CFG_1

Address	Bank 00 - 0x01 (Hex) - 1 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	*self_test*		*sns_lpf_bnd*				*sns_dout_fsc*	
Access	R/W		R/W				R/W	
Default	00		1010				00	

Description

Gyroscope self-test, full-scale and LPF frequency.

Fields

self_test: Set the gyroscope self-test mode.
This gyroscope embedded self-test feature can be used to verify if the gyroscope is working properly without physically rotating the device. That may be used either before or after it is assembled on a PCB. If the gyroscope's outputs are within the specified self-test values in the data sheet, then the gyroscope is working properly.

Value	Mnemonic	Description
00	Disabled	Self-test mode is disabled.
01	Positive sign	In this self-test mode, the expected values are {+X, -Y, +Z}, where X, Y and Z are a percentage of the full-scale.
10	Negative sign	In this self-test mode, the expected values are {-X, +Y, -Z}, where X, Y and Z are a percentage of the full-scale.

sns_lpf_bnd: Set the gyroscope low-pass filter cut-off frequency.

SNS_CFG_2[sns_gyr_ois_lpf]	Value	Mnemonic
0	0000	2 Hz
0	0001	4 Hz
0	0010	6 Hz
0	0011	8 Hz
0	0100	10 Hz
0	0101	14 Hz
0	0110	22 Hz
0	0111	32 Hz
0	1000	50 Hz
0	1001	75 Hz
0	1010	100 Hz
0	1011	150 Hz
0	1100	200 Hz

0	1101	250 Hz
0	1110	300 Hz
0	1111	400 Hz
1	0XXX	1 kHz
1	1XXX	2 kHz

sns_dout_fsc: Set the gyroscope full-scale.

SNS_CFG_2[sns_gyr_ois_lpf]	Value	Mnemonic
0	00	2000 dps
0	01	1000 dps
0	10	500 dps
0	11	250 dps
1	00	1000 dps
1	01	500 dps
1	10	250 dps
1	11	125 dps

Back to Register Overview

SNS_CFG_2

Address	Bank 00 - 0x02 (Hex) - 2 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU		***sns_gyr_ois_lpf***	***sns_gyr_hpf_en***	***sns_odr***			
Access	R		R/W	R/W	R/W			
Default	-		0	0	0100			

Description

Gyroscope OIS enable, ODR and HPF enable.

Fields

sns_gyr_ois_lpf: Enable the gyroscope full-scale for OIS applications.

Value	Mnemonic	Description
0	Normal full-scale	The available full-scale settings are: 2000dps, 1000dps, 500dps and 250dps.
1	OIS full-scale	For optical image stabilization (OIS) applications the main requirement is the resolution. So, in this mode, the available full scales are: 1000, 500dps, 250dps and 125dps.

sns_gyr_hpf_en: Enable the gyroscope high-pass filter (this option is not available in Low-Power mode).

0: Disabled
1: Enabled

sns_odr: Set the gyroscope output data rate (ODR).

If ***pwr_mode*** is Gyro Low-Power or Acc Low-Noise + Gyro Low-Power:

Value	Mnemonic
00XX	200 Hz
010X	200 Hz
0110	100 Hz
0111	50 Hz
1000	25 Hz
1001	10 Hz
101X	5 Hz
1100	5 Hz
1101	Don't care
111X	Don't care

Otherwise:

Value	Mnemonic
0000	8 kHz
0001	4 kHz
0010	2 kHz
0011	800 Hz
0100	400 Hz
0101	200 Hz
0110	100 Hz
0111	50 Hz
1000	25 Hz
1001	10 Hz

Back to Register Overview

SNS_CFG_3

Address	Bank 00 - 0x03 (Hex) - 3 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU			*wait_data_mode*		*sns_hpf_co*		
Access	R			R/W		R/W		
Default	-			00		000		

Description

Gyroscope HPF frequency and wait condition before generating the data.

Fields

wait_data_mode: To start the data generation, two triggers are defined:
- Trig1: from driveON to senseON
- Trig2: from senseON to NormalMode

This register allows specifying the conditions that have to be satisfied on both the triggers before starting to generate the data.

Value	Mnemonic	Description
00	None/None	Data generation starts immediately.
01	Freq Lock/Freq Lock	Data generation starts when the frequency is locked on both the Trig1 and Trig2 paths.
10	Freq Lock/Amp Lock	Data generation starts when the frequency is locked on the Trig1 path and the amplitude is locked on the Trig2 path.
11	Amp Lock/Amp Lock	Data generation starts when the amplitude is locked on both the Trig1 and Trig2 paths.

sns_hpf_co: Set the gyroscope high-pass filter cut-off frequency.

000: 0.08 Hz
001: 0.24 Hz
010: 0.8 Hz
011: 2 Hz
100: 5 Hz
101: 10 Hz
110: 20 Hz
111: 50 Hz

Back to Register Overview

SET_ACC_PWR

Address	Bank 00 - 0x04 (Hex) - 4 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	*sns_acc_fsc*		*acc_self_test*			RFU		
Access	R/W		R/W			R		
Default	11		000			-		

Description

Accelerometer self-test and full-scale.

Fields

sns_acc_fsc: Set the accelerometer full-scale.

00: 16 g
01: 8 g
10: 4 g
11: 2 g

acc_self_test: Set the accelerometer self-test mode. The accelerometer embedded self-test feature is used to verify the sensor functionality without physically moving the device. When this feature is enabled, an electrostatic test force is applied to the mechanical sensing element and causes the moving part to move away from its original position, emulating a definite input acceleration. In this case the sensor outputs exhibit a change in their DC levels which is related to the selected full scale through the device sensitivity. The output in this self-test mode is then compared with the output data of the device when the self-test is disabled. If the absolute value of the output difference is within the minimum and maximum range of the preselected full-scale range, the accelerometer is working properly.

Value	Mnemonic	Description
000	Disabled	Self-test mode is disabled.
001	Positive Force / X-Axis self-test	Self-test is enabled on X-axis and a positive force is applied.
010	Positive Force / Y-Axis self-test	Self-test is enabled on Y-axis and a positive force is applied.
011	Positive Force / Z-Axis self-test	Self-test is enabled on Z-axis and a positive force is applied.
101	Negative Force / X-Axis self-test	Self-test is enabled on X-axis and a negative force is applied.
110	Negative Force / Y-Axis self-test	Self-test is enabled on Y-axis and a negative force is applied.
111	Negative Force / Z-Axis self-test	Self-test is enabled on Z-axis and a negative force is applied.

Back to Register Overview

ACC_CFG_1

Address	Bank 00 - 0x05 (Hex) - 5 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	*sns_acc_hpf_co*		*sns_acc_lpf_co*		*sns_acc_odr*			
Access	R/W		R/W		R/W			
Default	00		00		0010			

Description

Accelerometer ODR and LPF/HPF frequencies.

Fields

sns_acc_hpf_co: Set the accelerometer highpass filter cut-off frequency.
For ODR = 33.33 Hz, the bandwidth computation is set as fraction of ODR* = 100 Hz.
For ODR = 16.67 Hz, the bandwidth computation is set as fraction of ODR* = 50 Hz.
For ODR = 10 Hz, the bandwidth computation is set as fraction of ODR* = 50 Hz.
For ODR = 5 Hz, the bandwidth computation is set as fraction of ODR* = 25 Hz.
For all other ODR settings, ODR* = ODR.

00: ODR/400
01: ODR/200
10: ODR/100
11: ODR/50

sns_acc_lpf_co: Set the accelerometer low-pass filter cut-off frequency.
For ODR = 33.33 Hz, the bandwidth computation is set as fraction of ODR* = 100 Hz.
For ODR = 16.67 Hz, the bandwidth computation is set as fraction of ODR* = 50 Hz.
For ODR = 10 Hz, the bandwidth computation is set as fraction of ODR* = 50 Hz.
For ODR = 5 Hz, the bandwidth computation is set as fraction of ODR* = 25 Hz.
For all other ODR settings, ODR* = ODR.

00: ODR/48
01: ODR/22
10: ODR/9
11: ODR/3

sns_acc_odr: Set the accelerometer output data rate (ODR).

If ***pwr_mode*** is Acc Low-Power:

Value	Mnemonic
000X	400 Hz
0010	400 Hz
0011	200 Hz
0100	100 Hz
0101	50 Hz
0110	25 Hz
0111	10 Hz
1XXX	5 Hz

Otherwise:

Value	Mnemonic
0000	2000 Hz
0001	800 Hz
0010	400 Hz
0011	200 Hz
0100	100 Hz
0101	50 Hz
0110	25 Hz
0111	10 Hz
1XXX	5 Hz

Back to Register Overview

ACC_CFG_2

Address	Bank 00 - 0x06 (Hex) - 6 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	***acc_dout_sel***		***acc_lowp_avg***		RFU			***acc_hpf_sel***
Access	R/W		R/W		R			R/W
Default	01		00		-			0

Description

Accelerometer output filter, signal averages in Low-Power mode and HPF input's filter.

Fields

acc_dout_sel: Set the filtering mode for the accelerometer data output.

00: Unfiltered
01: Low-pass filter
10: High-pass filter
11: High-pass filter

acc_lowp_avg: Set the number of high frequency signal averages to be computed in accelerometer Low-Power mode.
Limitations:
- 32 and 16 not achievable at 400Hz; the number of averages is set to 8;
- 32 is not achievable at 200Hz; the number of averages is set to 16.

00: 4 averages
01: 8 averages
10: 16 averages
11: 32 averages

acc_hpf_sel: Set the filtering mode to be applied to the accelerometer high-pass input.

0: Low-pass filter
1: Unfiltered

Back to Register Overview

SET_TEMP_DR

Address	Bank 00 - 0x13 (Hex) - 19 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU				***dr_rst_mode***		***coarse_temp***	***temp_en***
Access	R				R/W		R/W	R/W
Default	-				00		0	1

Description

DATA_READY reset and temperature sensor settings.

Fields

dr_rst_mode: Set the DATA_READY interrupt reset and data-out update mode.

Value	Mnemonic	Description
00	All	DATA_READY is cleared after all the active channels are read. Data are not updated until the clear operation is accomplished.
01	Any	DATA_READY is cleared when at least a byte of one of the active channels is read. Data are updated independently from the clear operation.
10	Status	DATA_READY is cleared when status register is read. Data are not updated until the clear operation is accomplished.

coarse_temp: Set the temperature data update granularity.

Value	Mnemonic	Description
0	Fine	Temperature data is updated only when both the temperature data registers are read (TEMP_H, TEMP_L).
1	Coarse	Temperature data is updated only when the most significant byte of the temperature data is read (TEMP_H).

temp_en: Enable the temperature sensor.

0: Disabled
1: Enabled

Back to Register Overview

SET_PU_PD_PAD

Address	Bank 00 - 0x14 (Hex) - 20 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU		***int1_pd_en***	***int1_pu_en***	***int2_pd_en***	***int2_pu_en***	RFU	***slv_pu_dis***
Access	R		R/W	R/W	R/W	R/W	R	R/W
Default	-		0	0	0	0	-	0

Description

Pullup/down connections.

Fields

int1_pd_en: Connect the internal pulldown of the INT1 pad.

0: Disconnected
1: Connected

int1_pu_en: Connect the internal pullup of the INT1 pad.

0: Disconnected
1: Connected

int2_pd_en: Connect the internal pulldown of the INT2 pad.

0: Disconnected
1: Connected

int2_pu_en: Connect the internal pullup of the INT2 pad.

0: Disconnected
1: Connected

slv_pu_dis: Disconnect the internal pullups of the I^2C slave pads.

0: Connected
1: Disconnected

Back to Register Overview

SET_I2C_PAD

Address	Bank 00 - 0x15 (Hex) - 21 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU					***strong_slave***	RFU	***strong_int_aux***
Access	R					R/W	R	R/W
Default	-					1	-	0

Description

PAD speed.

Fields

strong_slave: Increase the I^2C slave pads speed. This mode absorbs more current.

0: Normal (Low-Power)
1: Fast

strong_int_aux: Increase the interrupt pads speed. This mode absorbs more current.

0: Normal (Low-Power)
1: Fast

Back to Register Overview

MIF_CFG

Address	Bank 00 - 0x16 (Hex) - 22 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU		*if_setting*			*spi_3_wire*	*endian*	*i2c_off*
Access	R		R/W			R/W	R/W	R/W
Default	-		000			0	0	0

Description

Master communication interface settings and status.

Fields

if_setting: Select the master interface type.

000: I^2C Fast Mode standard configuration
001: I^2C Fast Mode without anti-spike filter
010: I^2C High Speed standard configuration
011: I^2C High Speed without anti-spike filter
100: I^2C mode without filters and delays
101: SPI interface

spi_3_wire: Set the interface type of the SPI.

Value	Mnemonic	Description
0	SPI Mode 4	In SPI Mode 4, data in/out have dedicated lines (MISO and MOSI, respectively), so the SPI bus has four lines.
1	SPI Mode 3	In SPI Mode 3, data in/out are shared, so the SPI bus has three lines.

endian: Set the endian of the data output.

0: Big endian
1: Little endian

i2c_off: Turn on/off the I^2C interface.

0: Active
1: Off

Back to Register Overview

FIFO_STORE

Address	Bank 00 - 0x17 (Hex) - 23 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU			*fifo_auto_store_pwr_en*	RFU		*fifo_store_acc*	*fifo_store_gyr*
Access	R			R/W	R		R/W	R/W
Default	-			0	-		0	0

Description

FIFO configuration.

Fields

fifo_auto_store_pwr_en: Enable the power-dependent FIFO data selection.
When the Power-Mode based data selection is active, which sensors are to be saved in FIFO is selected according to the power mode. For example, if the 'Gyro Low-Power' power mode is selected, only the gyroscope is stored in FIFO; then, if the power mode is switched to 'Acc Low-Noise + Gyro Low-Power' power mode, both the gyroscope and accelerometer data are stored in FIFO.
Both the Low-Power and Low-Noise mode are allowed for all the sensors.

Value	Mnemonic	Description
0	Register based data selection.	The data to be stored in FIFO are selected according to the fifo_store_acc and fifo_store_gyr values.
1	Power-Mode based data selection.	The data to be stored in FIFO are selected according to the selected power mode. The allowed transitions are: - G+A → A - G+A → G - A → G+A - G → G+A

fifo_store_acc: Store the accelerometer data in FIFO.

0: No Accelerometer in FIFO
1: Accelerometer in FIFO

fifo_store_gyr: Store the gyroscope data in FIFO.

0: No Gyroscope in FIFO
1: Gyroscope in FIFO

Back to Register Overview

FIFO_THS

Address	Bank 00 - 0x18 (Hex) - 24 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R/W							
Default	00000000							

Description

FIFO threshold.
When the number of 16-bits samples stored in FIFO is above the threshold, the fifo_ovthold interrupt is generated.
In this case, a sample is the entire set of axes; for example, if the threshold is set to 5 and all the axes are stored in FIFO, the interrupt is generated when the FIFO contains 5 sets of {X, Y, Z} gyroscope data.
This value must be different from 0.

Back to Register Overview

FIFO_CFG

Address	Bank 00 - 0x19 (Hex) - 25 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU			*fifo_mode*		*fifo_double_size*	*fifo_int_mode*	*fifo_overrun*
Access	R			R/W		R/W	R/W	R/W
Default	-			00		0	0	0

Description

FIFO configuration.

Fields

fifo_mode: Set the FIFO mode.

Value	Mnemonic	Description
00	Off	FIFO is disabled and the data are exposed through the registers only.
01	Normal	FIFO starts collecting the data immediately.
10	Interrupt	FIFO starts collecting the data right after a Rate Interrupt event (either OR or AND) is generated.
11	Snapshot	FIFO starts collecting the data immediately, overwriting the oldest data in case of FIFO full. When a Rate Interrupt event (either OR or AND) is generated, data are collected until the FIFO is full, then the data collection is stopped without overwriting the oldest values. This mechanism is useful in case a post-processing of the data that generated the Rate Interrupt event is requested to better classify the event itself.

fifo_double_size: Enable the FIFO to use also the MCS RAM as memory storage (the FIFO size becomes 2x the size of the standard memory). In this mode, the MCS cannot be used.

0: Single
1: Double

fifo_int_mode: When an interrupt mode is selected, this bit defines which kind of mask must be used.

0: OR Mask
1: AND Mask

fifo_overrun: Set the FIFO overrun mode.

Value	Mnemonic	Description
0	Stop on Full	When the FIFO is full, no more data are collected and newer data are discarded.
1	Overwrite	When the FIFO is full, oldest data are replaced by newer ones.

Back to Register Overview

OTP_STS_CFG

Address	Bank 00 - 0x1C (Hex) - 28 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	*pwr_mode_rdy*	*parity_error*	*if_parity*		*otp_ecc_stat*		*chrp_in_prgs*	*otp_dld_restart*
Access	R	R	R/W		R		R	R/W
Default	1	0	00		00		0	0

Description

Interface parity setting, general status and errors.

Fields

pwr_mode_rdy: Indicate if the current power mode is equal to the selected one in the pwr_mode field or the device is still switching it.

0: Ongoing
1: Completed

parity_error: Indicate if a parity error occurred.

0: No Error
1: Error

if_parity: Sets the meaning of the bit 6 of the address during the communication from the master to the slave.

Value	Mnemonic	Description
00	Auto-increment	Bit 6 indicates if the register address has to be incremented after each data byte when a burst operation is requested. If bit 6 is LOW, the register address is auto-incremented; if bit 6 is HIGH, the register address is kept unchanged.
01	Even parity bit	Bit 6 represents the even parity bit. When this mode is selected and a burst operation is requested, the register address is incremented automatically after each data byte of the burst.
10	Odd parity bit	Bit 6 represents the odd parity bit. When this mode is selected and a burst operation is requested, the register address is incremented automatically after each data byte of the burst.

otp_ecc_stat: OTP download status.

Value	Mnemonic	Description
00	Program OK (1)	OTP download completed successfully.
01	Program OK (2)	OTP download completed successfully, 1 bit corrected.
10	Program OK (3)	OTP download completed successfully, after some attempts.
11	Program FAIL	OTP download completed with errors.

chrp_in_prgs: Status of the stiction recovery operation.

0: Idle
1: In progress

otp_dld_restart: Trigger the OTP download.

0: Idle
1: Trigger

Back to Register Overview

11.2.3 Bank 01

INT_REF_X

Address	Bank 01 - 0x00 (Hex) - 0 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R/W							
Default	00000000							

Description

MSB of the reference for X-axis interrupt.
Set the most significant byte of the reference for interrupt of X-axis.
When int_single_ref is set, the reference will be {INT_REF_X, INT_REF_Y}, where INT_REF_X is the most significant byte and INT_REF_Y is the least significant byte.

Back to Register Overview

INT_REF_Y

Address	Bank 01 - 0x01 (Hex) - 1 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R/W							
Default	00000000							

Description

MSB of the reference for Y-axis interrupt.
Set the most significant byte of the reference for interrupt of Y-axis.
When int_single_ref is set, the reference will be {INT_REF_X, INT_REF_Y}, where INT_REF_X is the most significant byte and INT_REF_Y is the least significant byte.

Back to Register Overview

INT_REF_Z

Address	Bank 01 - 0x02 (Hex) - 2 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R/W							
Default	00000000							

Description

MSB of the reference for Z-axis interrupt.
Set the most significant byte of the reference for interrupt of Z-axis.
When int_single_ref is set, the reference will be {INT_REF_X, INT_REF_Y}, where INT_REF_X is the most significant byte and INT_REF_Y is the least significant byte.

Back to Register Overview

INT_DEB_X

Address	Bank 01 - 0x03 (Hex) - 3 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU				***int_deb_x***			
Access	R				R/W			
Default	-				0000			

Description

Rate Interrupt debounce (X-axis).

Fields

int_deb_x: This register determines how long (measured in number of samples) the selected AND/OR interrupt configuration has to stay asserted before the corresponding interrupt source bit is set and the interrupt on either INT1 or INT2 is eventually generated.

When the selected AND/OR interrupt configuration deasserts (goes to 0) the corresponding interrupt source bit de-asserts immediately, without debounce.

The actual value is 2*int_deb_x + 1.

Back to Register Overview

INT_DEB_Y

Address	Bank 01 - 0x04 (Hex) - 4 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU				***int_deb_y***			
Access	R				R/W			
Default	-				0000			

Description

Rate Interrupt debounce (Y-axis).

Fields

int_deb_y: This register determines how long (measured in number of samples) the selected AND/OR interrupt configuration has to stay asserted before the corresponding interrupt source bit is set and the interrupt on either INT1 or INT2 is eventually generated.

When the selected AND/OR interrupt configuration de-asserts (goes to 0) the corresponding interrupt source bit de-asserts immediately, without debounce.

The actual value is 2*int_deb_y + 1.

Back to Register Overview

INT_DEB_Z

Address	Bank 01 - 0x05 (Hex) - 5 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU				***int_deb_z***			
Access	R				R/W			
Default	-				0000			

Description

Rate Interrupt debounce (Z-axis).

Fields

int_deb_z: This register determines how long (measured in number of samples) the selected AND/OR interrupt configuration has to stay asserted before the corresponding interrupt source bit is set and the interrupt on either INT1 or INT2 is eventually generated.

When the selected AND/OR interrupt configuration de-asserts (goes to 0) the corresponding interrupt source bit deasserts immediately, without debounce.

The actual value is 2*int_deb_z + 1.

Back to Register Overview

INT_THS_X

Address	Bank 01 - 0x06 (Hex) - 6 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	***int_x_high_pos_en***	***int_x_low_pos_en***	***int_x_high_neg_en***	***int_x_low_neg_en***	***int_x_high_pos***	***int_x_low_pos***	***int_x_high_neg***	***int_x_low_neg***
Access	R/W	R/W	R/W	R/W	R	R	R	R
Default	0	0	0	0	0	0	0	0

Description

Signal threshold condition and mask for X-axis.

Fields

int_x_high_pos_en: Enable the int_x_high_pos interrupt generation for threshold event detection on X-axis.

0: Disabled
1: Enabled

int_x_low_pos_en: Enable the int_x_low_pos interrupt generation for threshold event detection on X-axis.

0: Disabled
1: Enabled

int_x_high_neg_en: Enable the int_x_high_neg interrupt generation for threshold event detection on X-axis.

0: Disabled
1: Enabled

int_x_low_neg_en: Enable the int_x_low_neg interrupt generation for threshold event detection on X-axis.

0: Disabled
1: Enabled

int_x_high_pos: Signal is positive, above threshold.

0: False
1: True

int_x_low_pos: Signal is positive, below threshold.

0: False
1: True

int_x_high_neg: Signal is negative, above threshold.

0: False
1: True

int_x_low_neg: Signal is negative below threshold.

0: False
1: True

Back to Register Overview

INT_THS_Y

Address	Bank 01 - 0x07 (Hex) - 7 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	***int_y_high_pos_en***	***int_y_low_pos_en***	***int_y_high_neg_en***	***int_y_low_neg_en***	***int_y_high_pos***	***int_y_low_pos***	***int_y_high_neg***	***int_y_low_neg***
Access	R/W	R/W	R/W	R/W	R	R	R	R
Default	0	0	0	0	0	0	0	0

Description

Signal threshold condition and mask for Y-axis.

Fields

int_y_high_pos_en: Enable the int_y_high_pos interrupt generation for threshold event detection on Y-axis.
0: Disabled
1: Enabled

int_y_low_pos_en: Enable the int_y_low_pos interrupt generation for threshold event detection on Y-axis.
0: Disabled
1: Enabled

int_y_high_neg_en: Enable the int_y_high_neg interrupt generation for threshold event detection on Y-axis.
0: Disabled
1: Enabled

int_y_low_neg_en: Enable the int_y_low_neg interrupt generation for threshold event detection on Y-axis.
0: Disabled
1: Enabled

int_y_high_pos: Signal is positive, above threshold.
0: False
1: True

int_y_low_pos: Signal is positive, below threshold.
0: False
1: True

int_y_high_neg: Signal is negative, above threshold.
0: False
1: True

int_y_low_neg: Signal is negative below threshold.
0: False
1: True

Back to Register Overview

INT_THS_Z

Address	Bank 01 - 0x08 (Hex) - 8 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	***int_z_high_pos_en***	***int_z_low_pos_en***	***int_z_high_neg_en***	***int_z_low_neg_en***	***int_z_high_pos***	***int_z_low_pos***	***int_z_high_neg***	***int_z_low_neg***
Access	R/W	R/W	R/W	R/W	R	R	R	R
Default	0	0	0	0	0	0	0	0

Description

Signal threshold condition and mask for Z-axis.

Fields

int_z_high_pos_en: Enable the int_z_high_pos interrupt generation for threshold event detection on Z-axis.

0: Disabled
1: Enabled

int_z_low_pos_en: Enable the int_z_low_pos interrupt generation for threshold event detection on Z-axis.

0: Disabled
1: Enabled

int_z_high_neg_en: Enable the int_z_high_neg interrupt generation for threshold event detection on Z-axis.

0: Disabled
1: Enabled

int_z_low_neg_en: Enable the int_z_low_neg interrupt generation for threshold event detection on Z-axis.

0: Disabled
1: Enabled

int_z_high_pos: Signal is positive, above threshold.

0: False
1: True

int_z_low_pos: Signal is positive, below threshold.

0: False
1: True

int_z_high_neg: Signal is negative, above threshold.

0: False
1: True

int_z_low_neg: Signal is negative below threshold.

0: False
1: True

Back to Register Overview

INT_COND

Address	Bank 01 - 0x09 (Hex) - 9 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU	***int_freeze***	***int_mask_xyz_and***			***int_mask_xyz_or***		
Access	R	R/W	R/W			R/W		
Default	-	0	000			000		

Description

Rate Interrupt settings.

Fields

int_freeze: Set the interrupt on threshold as latched.
When enabled, all the Rate Interrupt flags are latched. When triggered, the interrupt is latched until the INT_MSK_{X,Y,Z} register is read.

0: Unlatched
1: Latched

int_mask_xyz_and: Each bit activates an axis. The active axes are ANDed together to generate the AND interrupt.

int_mask_xyz_or: Each bit activates an axis. The active axes are ORed together to generate the OR interrupt.

Back to Register Overview

INT_CFG_1

Address	Bank 01 - 0x0A (Hex) - 10 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	*sns_intp_fsc*		*int1_clk_out*	*int2_clk_out*	*int_single_deb*	*int_single_ref*	*sns_intp_hpf*	RFU
Access	R/W		R/W	R/W	R/W	R/W	R/W	R
Default	00		0	0	0	0	0	-

Description

Interrupt configuration register #1.

Fields

sns_intp_fsc: Set the full-scale applied to the gyroscope data used to issue the AND/OR interrupts.

SNS_CFG_2[sns_gyr_ois_lpf]	Value	Mnemonic
0	00	2000 dps
0	01	1000 dps
0	10	500 dps
0	11	250 dps
1	00	1000 dps
1	01	500 dps
1	10	250 dps
1	11	125 dps

int1_clk_out/ int2_clk_out: INT1/INT2 pad drives out the internal clock (8.8MHz).

0: Off
1: On

int_single_deb: When "On", the same duration {INT_DEB_X, INT_DEB_Y} is used for all the axes, where INT_DEB_Y is the LSB.

0: Off
1: On

int_single_ref: When "On", the same threshold {INT_REF_X, INT_REF_Y} is used for all the axes.

0: Off
1: On

sns_intp_hpf: Enable the high-pass filtering to the gyroscope data out used to issue the AND/OR interrupts.

0: Without high-pass filtering
1: With high-pass filtering

Back to Register Overview

INT_CFG_2

Address	Bank 01 - 0x0B (Hex) - 11 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU	***merge_int2_to_int1***	***int1_enable***	***int1_active_level***	***int1_out_mode***	***int2_enable***	***int2_active_level***	***int2_out_mode***
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	-	0	0	0	0	0	0	0

Description

Interrupt configuration register #2.

Fields

merge_int2_to_int1: Enable to merge the INT1 and INT2. When set, the interrupt INT1 changes its output depending on both INT1 and INT2.

0: INT1 and INT2 separated
1: INT2 merged with INT1

int1_enable: Enable interrupts on INT1.

0: Disabled
1: Enabled

int1_active_level: Set the INT1 active level.

0: INT1 active high
1: INT1 active low

int1_out_mode: Set the INT1 output configuration.

0: Push-pull
1: Open drain

int2_enable: Enable interrupts on INT2.

0: Disabled
1: Enabled

int2_active_level: Set the INT2 active level.

0: INT2 active high
1: INT2 active low

int2_out_mode: Set the INT2 output configuration.

0: Push-pull
1: Open drain

Back to Register Overview

INT_TMO_CFG

Address	Bank 01 - 0x0C (Hex) - 12 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	*int1_latch_mode*		*int2_latch_mode*		*int_timeout*			
Access	R/W		R/W		R/W			
Default	00		00		0000			

Description

Interrupts latch mode and duration.

Fields

int1_latch_mode: Set the INT1 latch mode.

00: Not latched
01: Latched - Clear on read
10: Latched - Clear on write
11: Timed

int2_latch_mode: Set the INT2 latch mode.

00: Not latched
01: Latched - Clear on read
10: Latched - Clear on write
11: Timed

int_timeout: Set how long the interrupt must be kept raised. This is shared between INT1 and INT2.

0000: 125 us
0001: 250 us
0010: 500 us
0011: 1 ms
0100: 2 ms
0101: 5 ms
0110: 10 ms
0111: 20 ms
1000: 50 ms
1001: 100 ms
1010: 200 ms
1011: 500 ms

Back to Register Overview

INT_STATUS_UL

Address	Bank 01 - 0x0D (Hex) - 13 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	***sts_ul_data_ready***	***sts_ul_fifo_empty***	***sts_ul_fifo_overrun***	***sts_ul_fifo_ths***	***sts_ul_int_and***	***sts_ul_int_or***	***sts_ul_otp_downloading***	RFU
Access	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	-

Description

Unlatched interrupts status bits.

Fields

sts_ul_data_ready: DATA_READY status.

sts_ul_fifo_empty: FIFO_EMPTY status.

sts_ul_fifo_overrun: FIFO_OVERRUN status.

sts_ul_fifo_ths: FIFO_THRESHOLD status.

sts_ul_int_and: INT_AND status.

sts_ul_int_or: INT_OR status.

sts_ul_otp_downloading: OTP_DOWNLOADING status.

Back to Register Overview

INT_STS

Address	Bank 01 - 0x0E (Hex) - 14 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	***sts_i1_data_ready***	***sts_i1_fifo_empty***	***sts_i1_fifo_overrun***	***sts_i1_fifo_ths***	***sts_i2_int_and***	***sts_i2_int_or***	***sts_i2_otp_downloading***	RFU
Access	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	-

Description

Latched status bits.

Fields

sts_i1_data_ready: DATA_READY status on INT1.
Added for user convenience, it is the same as sts_ul_data_ready in INT_STATUS_UL, so it is unlatched.

sts_i1_fifo_empty: FIFO_EMPTY status on INT1. During latched_mode = timed this is unlatched.

sts_i1_fifo_overrun: FIFO_OVERRUN status on INT1.

sts_i1_fifo_ths: FIFO_THRESHOLD status on INT1.

sts_i2_int_and: INT_AND status on INT2.

sts_i2_int_or: INT_OR status on INT2.

sts_i2_otp_downloading: OTP_DOWNLOADING status on INT2.

Back to Register Overview

INT_MSK

Address	Bank 01 - 0x0F (Hex) - 15 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	***msk_i1_data_ready***	***msk_i1_fifo_empty***	***msk_i1_fifo_overrun***	***msk_i1_fifo_ths***	***msk_i2_int_and***	***msk_i2_int_or***	***msk_i2_otp_downloading***	RFU
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Default	0	0	0	0	0	0	1	-

Description

Enable the interrupt status.

Fields

msk_i1_data_ready: DATA_READY status on INT1.

0: Disabled
1: Enabled

msk_i1_fifo_empty: FIFO_EMPTY status on INT1.

0: Disabled
1: Enabled

msk_i1_fifo_overrun: FIFO_OVERRUN status on INT1.

0: Disabled
1: Enabled

msk_i1_fifo_ths: FIFO_THRESHOLD status on INT1.

0: Disabled
1: Enabled

msk_i2_int_and: INT_AND status on INT2.

0: Disabled
1: Enabled

msk_i2_int_or: INT_OR status on INT2.

0: Disabled
1: Enabled

msk_i2_otp_downloading: OTP_DOWNLOADING status on INT2.

0: Disabled
1: Enabled

Back to Register Overview

INT_SRC_CFG

Address	Bank 01 - 0x17 (Hex) - 23 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	*acc_intp_sel*		*msk_gyr_int_d_rdy*	*msk_acc_int_d_rdy*	RFU	*acc_norm_unfilt*	*int_src_cfg*	
Access	R/W		R/W	R/W	R	R/W	R/W	
Default	00		1	0	-	0	00	

Description

DATA_READY interrupt configuration, Rate Interrupt source selection and accelerometer filters.

Fields

acc_intp_sel: Set the filtering mode for the accelerometer interrupt.

00: Unfiltered
01: Low-pass filter
10: High-pass filter
11: High-pass filter

msk_gyr_int_d_rdy: If set, a new gyroscope data contributes to DATA_READY interrupt assertion.

0: None
1: New Gyro data to data ready

msk_acc_int_d_rdy: If set, a new accelerometer data contributes to DATA_READY interrupt assertion.

0: None
1: New Acc data to data ready

acc_norm_unfilt: If set, the accelerometer norm low-pass filter is enabled.

0: Norm before LPF
1: Norm after LPF

int_src_cfg: Configure the Rate Interrupt source.

00: Gyroscope
10: Accelerometer HPF norm
11: Accelerometer before HPF norm

Back to Register Overview

SERIAL_0

Address	Bank 01 - 0x1A (Hex) - 26 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000000							

Description

Serial number #0.

Back to Register Overview

SERIAL_1

Address	Bank 01 - 0x1B (Hex) - 27 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000000							

Description

Serial number #1.

Back to Register Overview

SERIAL_2

Address	Bank 01 - 0x1C (Hex) - 28 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000000							

Description

Serial number #2.

Back to Register Overview

SERIAL_3

Address	Bank 01 - 0x1D (Hex) - 29 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000000							

Description

Serial number #3.

Back to Register Overview

SERIAL_4

Address	Bank 01 - 0x1E (Hex) - 30 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000000							

Description

Serial number #4.

Back to Register Overview

SERIAL_5

Address	Bank 01 - 0x1F (Hex) - 31 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	-							
Access	R							
Default	00000000							

Description

Serial number #5.

Back to Register Overview

11.2.4 Bank 02

BIAS_COMP_GX_MSB

Address	Bank 02 - 0x13 (Hex) - 19 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU			***bias_comp_gyr_x_msb***				
Access	R			R/W				
Default	-			00000				

Description

Bias compensation gyroscope X (MSB).
All the bias compensation registers for the gyroscope must be written in sequence for the changes to take effect.

Fields

bias_comp_gyr_x_msb: Most significant bits of the 13 bits gyroscope offset compensation register for the X-axis.

The real value is computed as: bias_comp_gyr * 8.33 mdps.

The real value range is: [-34.13; +34.13] dps.

Back to Register Overview

BIAS_COMP_GX_LSB

Address	Bank 02 - 0x14 (Hex) - 20 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	***bias_comp_gyr_x_lsb***							
Access	R/W							
Default	00000000							

Description

Bias compensation gyroscope X (LSB).
All the bias compensation registers for the gyroscope must be written in sequence for the changes to take effect.

Fields

bias_comp_gyr_x_lsb: Least significant byte of the 13 bits gyroscope offset compensation register for the X-axis.

The real value is computed as: bias_comp_gyr * 8.33 mdps.

The real value range is: [-34.13; +34.13] dps.

Back to Register Overview

BIAS_COMP_GY_MSB

Address	Bank 02 - 0x15 (Hex) - 21 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU			***bias_comp_gyr_y_msb***				
Access	R			R/W				
Default	-			00000				

Description

Bias compensation gyroscope Y (MSB).
All the bias compensation registers for the gyroscope must be written in sequence for the changes to take effect.

Fields

bias_comp_gyr_y_msb: Most significant bits of the 13 bits gyroscope offset compensation register for the Y-axis.

The real value is computed as: bias_comp_gyr * 8.33 mdps.

The real value range is: [-34.13; +34.13] dps.

Back to Register Overview

BIAS_COMP_GY_LSB

Address	Bank 02 - 0x16 (Hex) - 22 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	***bias_comp_gyr_y_lsb***							
Access	R/W							
Default	00000000							

Description

Bias compensation gyroscope Y (LSB).
All the bias compensation registers for the gyroscope must be written in sequence for the changes to take effect.

Fields

bias_comp_gyr_y_lsb: Least significant byte of the 13 bits gyroscope offset compensation register for the Y-axis.

The real value is computed as: bias_comp_gyr * 8.33 mdps.

The real value range is: [-34.13; +34.13] dps.

Back to Register Overview

BIAS_COMP_GZ_MSB

Address	Bank 02 - 0x17 (Hex) - 23 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU			***bias_comp_gyr_z_msb***				
Access	R			R/W				
Default	-			00000				

Description

Bias compensation gyroscope Z (MSB).
All the bias compensation registers for the gyroscope must be written in sequence for the changes to take effect.

Fields

bias_comp_gyr_z_msb: Most significant bits of the 13 bits gyroscope offset compensation register for the Z-axis.

The real value is computed as: bias_comp_gyr * 8.33 mdps.

The real value range is: [-34.13; +34.13] dps.

Back to Register Overview

BIAS_COMP_GZ_LSB

Address	Bank 02 - 0x18 (Hex) - 24 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	***bias_comp_gyr_z_lsb***							
Access	R/W							
Default	00000000							

Description

Bias compensation gyroscope Z (LSB).
All the bias compensation registers for the gyroscope must be written in sequence for the changes to take effect.

Fields

bias_comp_gyr_z_lsb: Least significant byte of the 13 bits gyroscope offset compensation register for the Z-axis.

The real value is computed as: bias_comp_gyr * 8.33 mdps.

The real value range is: [-34.13; +34.13] dps.

Back to Register Overview

BIAS_COMP_AX

Address	Bank 02 - 0x19 (Hex) - 25 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU	***bias_comp_acc_x***						
Access	R	R/W						
Default	-	0000000						

Description

Bias compensation accelerometer X.
All the bias compensation registers for the accelerometer must be written in sequence for the changes to take effect.

Fields

bias_comp_acc_x: Accelerometer offset compensation register for the X-axis.

The real value is computed as: bias_comp_acc * 6.4 mg.

The real value range is: [-0.41; +0.41] g.

Back to Register Overview

BIAS_COMP_AY

Address	Bank 02 - 0x1A (Hex) - 26 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU	***bias_comp_acc_y***						
Access	R	R/W						
Default	-	0000000						

Description

Bias compensation accelerometer Y.
All the bias compensation registers for the gyroscope must be written in sequence for the changes to take effect.

Fields

bias_comp_acc_y: Accelerometer offset compensation register for the Y-axis.

The real value is computed as: bias_comp_acc * 6.4 mg.

The real value range is: [-0.41; +0.41] g.

Back to Register Overview

BIAS_COMP_AZ

Address	Bank 02 - 0x1B (Hex) - 27 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU	***bias_comp_acc_z***						
Access	R	R/W						
Default	-	0000000						

Description

Bias compensation accelerometer Z.
All the bias compensation registers for the gyroscope must be written in sequence for the changes to take effect.

Fields

bias_comp_acc_z: Accelerometer offset compensation register for the Z-axis.

The real value is computed as: bias_comp_acc * 6.4 mg.

The real value range is: [-0.41; +0.41] g.

Back to Register Overview

GYR_ODR_TRIM

Address	Bank 02 - 0x1F (Hex) - 31 (Dec)							
Bit #	7	6	5	4	3	2	1	0
Fields	RFU				*gyr_ok*	*acc_ok*	RFU	
Access	R				R	R	R	
Default	-				0	0	-	

Description

Sensors status.

Fields

gyr_ok: Gyroscope drive loop is locked in amplitude and gyro sense chain is active.

0: Not OK
1: OK

acc_ok: Accelerometer sense chain is active.

0: Not OK
1: OK

Back to Register Overview

REVISION HISTORY

Revision, Date	Change	Page(s)
1.0, 06/19/2015	Creation	-
1.1, 07/03/2015	Review	-
1.2, 07/07/2015	Fixed corrupted bookmarks	-
1.3, 07/09/2015	Fixed register map	-
1.4, 07/10/2015	Fixed FIFO images and added register map comments	-