InvenSense	**InvenSense Inc.** 1197 Borregas Ave, Sunnyvale, CA 94089 U.S.A. Tel: +1 (408) 988-7339 Fax: +1 (408) 988-8104 Website: www.invensense.com	Document Number: RM-MPU-3000A-00 Revision: 2.1 Release Date: 05/19/2011

MPU-3000 & MPU-3050 Register Map and Register Descriptions

Revision 2.1

CONTENTS

1 Revision History

Revision Date	Revision	Description
12/23/2010	1.0	First revision of document
03/01/2011	2.0	Section 4.1: Added clarification for bit [0] Section 4.5: Fixed logic polarity of AUX_IF_EN flag in User Control Register (61)
05/19/2011	2.1	Section 2.2: Added InvenSense software solutions section Section 4.13: Corrected TEMP_XOUT as TEMP_OUT in diagram

2 Purpose and Scope

This document provides information regarding the register map and register descriptions for the MPU-3000™ and MPU-3050™, collectively called the MPU-30X0™ or MPU™.

2.1 Product Overview

The MPU-30X0 Motion Processing Unit (MPU™) is the world's first MotionProcessing™ solution with integrated 6-axis sensor fusion using its field-proven and proprietary MotionFusion™ engine for smart phone applications. The MPU-30X0 has an embedded 3-axis gyroscope and Digital Motion Processor™ (DMP) hardware accelerator engine with a secondary I^2C port that interfaces to third party digital accelerometers to deliver a complete 6-axis sensor fusion output to its primary I^2C port. This combines both linear and rotational motion into a single data stream for the application. This breakthrough in gyroscope technology provides a dramatic 68% smaller footprint, 40% thinner package, consumes 55% less power, and has inherent cost advantages compared to the latest competitive gyro solutions to uniquely address the fast-growing demand for 6-axis MotionProcessing in mobile handsets. The primary interface also supports SPI protocol on the MPU-3000 and can be used to read/write to all the registers on the part. The MPU's memory and FIFO are not accessible via the SPI interface.

The MPU-30X0 significantly extends and transforms motion sensing features provided by accelerometers beyond portrait and landscape orientation, to MotionProcessing functionality. The MPU measures and processes both linear and rotational movements, creating a higher degree of 1:1 motion interactivity between the user and their handset. Similar to the proliferation of Bluetooth, camera phone image sensors and Wi-Fi, MotionProcessing is becoming a "must-have" function in mobile handsets benefitting wireless carriers, mobile handset OEMs, application developers and end-users. By providing an integrated sensor fusion output, the DMP in the MPU-30X0 offloads the intensive MotionProcessing computation requirements from the applications processor, reducing the need for frequent polling of the motion sensor output and enabling use of low cost, low power application processors thereby increasing overall battery life of handsets. Since handsets today are of multi-function nature, MPU-30X0 not only provides accurate 1:1 motion tracking for some of the more common applications such as still/video image stabilization, gaming and dead reckoning, the 32-bit DMP can be programmed to deliver advanced UI, e.g. multiple kinds of gestures and character recognition leading to applications such as *Airsign*™, *TouchAnywhere*™, *MotionCommand*™.

By leveraging its patented and volume-proven Nasiri-Fabrication platform, which integrates MEMS wafers with companion CMOS electronics through wafer-level bonding, InvenSense has driven the MPU-30X0 package size down to a revolutionary footprint of 4x4x0.9mm (QFN), while providing the highest performance, lowest noise, and the lowest cost semiconductor packaging to address a wide range of handheld consumer electronic devices.

The MPU-30X0 integrates 16-bit analog-to-digital converters (ADCs), selectable low-pass filters, FIFO, embedded temperature sensor, and Fast Mode I^2C or SPI (MPU-3000 only) interfaces. Performance features include programmable full-scale range from ±250 degrees-per-second up to ±2000 degrees-per-second (º/s or dps), and low-noise of 0.01º/s/√Hz, while providing the highest robustness supporting 10,000*g* shock in operation. The highest cross-axis isolation is achieved by design from its single silicon integration. Factory-calibrated initial sensitivity reduces production-line calibration requirements. The part's on-chip FIFO and dedicated I^2C-master accelerometer sensor

bus simplify system timing and lower system power consumption. The sensor bus allows the MPU-30X0 to directly acquire data from the off-chip accelerometer without intervention from an external processor. Other industry-leading features include a small 4mmx4mmx0.9mm plastic QFN package, an embedded temperature sensor, programmable interrupts, and a low 13mW power consumption. Parts are available with I^2C and SPI serial interfaces, a VDD operating range of 2.1 to 3.6V, and a VLOGIC interface voltage from 1.71V to 3.6V.

For more detailed information regarding the MPU-30X0 devices, please refer to the "MPU-3000 and MPU-3050 Product Specification".

2.2 Software Solutions

This section describes the MotionApps™ software solutions included with the InvenSense MPU (Motion Processing Unit) and IMU (Inertial Measurement Unit) product families. Please note that the products within the IDG, IXZ, and ITG families do not include these software solutions.

The MotionApps Platform is a complete software solution that in combination with the InvenSense IMU and MPU MotionProcessor families delivers robust, well-calibrated 6-axis and/or 9-axis sensor fusion data using its field proven and proprietary MotionFusion™ engine. Solution packages are available for smartphones and tablets as well as for embedded microcontroller-based devices.

The MotionApps Platform provides a turn-key solution for developers and accelerates time-to-market. It consists of complex 6/9-axis sensor fusion algorithms, robust multi-sensor calibration, a proven software architecture for Android and other leading operating systems, and a flexible power management scheme.

The MotionApps Platform is integrated within the middleware of the target OS (the sensor framework), and also provides a kernel device driver to interface with the physical device. This directly benefits application developers by providing a cohesive set of APIs and a well-defined sensor data path in the user-space.

The table below describes the MotionApps software solutions included with the InvenSense MPU and IMU product families.

InvenSense MotionProcessor Devices and Included MotionApps Software

Feature	Included Software				Notes
	MotionApps	Embedded MotionApps	MotionApps Lite	Embedded MotionApps Lite	
Part Number	MPU-3050 MPU-6050		IMU-3000		
Processor Type	Mobile Application Processor	8/16/32-bit Microcontroller	Mobile Application Processor	8/16/32-bit Microcontroller	
Applications	Smartphones, tablets	TV remotes, health/fitness, toys, other embedded	Smartphones, tablets	TV remotes, health/fitness, toys, other embedded	
6-Axis MotionFusion	Yes		Yes		< 2% Application Processor load using on-chip Digital Motion Processor (DMP). Reduces processing requirements for embedded applications
9-Axis MotionFusion	Yes		No		
Gyro Bias Calibration	Yes		Yes		No-Motion calibration and temperature calibration
3rd Party Compass Cal API	Yes		No		Integrates 3rd party compass libraries
Gyro-Assisted Compass Calibration (Fast Heading)	Yes		No		Quick compass calibration using gyroscope
Magnetic Anomaly Rejection (Improved Heading)	Yes		No		Uses gyro heading data when magnetic anomaly is detected

The table below lists recommended documentation for the MotionApps software solutions.

Software Documentation

Platform	MotionApps and MotionApps Lite	Embedded MotionApps and Embedded MotionApps Lite
Software Documentation	• Installation Guide for Linux and Android MotionApps Platform, v1.9 or later • MPL Functional Specifications	• Embedded MotionApps Platform User Guide, v3.0 or later • Embedded MPL Functional Specifications

For more information about the InvenSense MotionApps Platform, please visit the Developer's Corner or consult your local InvenSense Sales Representative.

3 Register Map

The register map for the MPU-30X0 is listed below.

Addr (Hex)	Addr (Decimal)	Register Name	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	0	WHO_AM_I	R/W	I2C_IF_DIS	ID						-
1	1	PRODUCT_ID	R/W	PART_NUM				VERSION			
C	12	X_OFFS_USRH	R/W	X_OFF_H							
D	13	X_OFFS_USRL	R/W	X_OFF_L							
E	14	Y_OFFS_USRH	R/W	Y_OFFS_H							
F	15	Y_OFFS_USRL	R/W	Y_OFFS_L							
10	16	Z_OFFS_USRH	R/W	Z_OFFS_H							
11	17	Z_OFFS_USRL	R/W	Z_OFFS_L							
12	18	FIFO_EN	R/W	TEMP_OUT	GYRO_XOUT	GYRO_YOUT	GYRO_ZOUT	AUX_XOUT	AUX_YOUT	AUX_ZOUT	FIFO_FOOTER
13	19	AUX_VDDIO	R/W	0	0	0	0	0	AUX_VDDIO	0	0
14	20	AUX_SLV_ADDR	R/W	CLKOUT EN	AUX_ID						
15	21	SMPLRT_DIV	R/W	SMPLRT_DIV							
16	22	DLPF_FS_SYNC	R/W	EXT_SYNC_SET			FS_SEL		DLPF_CFG		
17	23	INT_CFG	R/W	ACTL	OPEN	LATCH_INT_EN	INT_ANYRD_2CLEAR	-	MPU_RDY_EN	DMP_DONE_EN	RAW_RDY_ EN
18	24	AUX_ADDR	R/W	BURST_ADDR							
1A	26	INT_STATUS	R	-	-	-	-	-	MPU_RDY	DMP_DONE	RAW_DATA_RDY
1B	27	TEMP_OUT_H	R	TEMP_OUT_H							
1C	28	TEMP_OUT_L	R	TEMP_OUT_L							
1D	29	GYRO_XOUT_H	R	GYRO_XOUT_H							
1E	30	GYRO_XOUT_L	R	GYRO_XOUT_L							
1F	31	GYRO_YOUT_H	R	GYRO_YOUT_H							
20	32	GYRO_YOUT_L	R	GYRO_YOUT_L							
21	33	GYRO_ZOUT_H	R	GYRO_ZOUT_H							
22	34	GYRO_ZOUT_L	R	GYRO_ZOUT_L							
23	35	AUX_XOUT_H	R	AUX_XOUT_H							
24	36	AUX_XOUT_L	R	AUX_XOUT_L							
25	37	AUX_YOUT_H	R	AUX_YOUT_H							
26	38	AUX_YOUT_L	R	AUX_YOUT_L							
27	39	AUX_ZOUT_H	R	AUX_ZOUT_H							
28	40	AUX_ZOUT_L	R	AUX_ZOUT_L							
3A	58	FIFO_COUNTH	R	-	-	-	-	-	-	FIFO_COUNT_H	
3B	59	FIFO_COUNTL	R	FIFO_COUNT_L							
3C	60	FIFO_R	R	FIFO_DATA							
3D	61	USER_CTRL	R/W	-	FIFO_ EN	AUX_IF_EN	-	AUX_IF_RST	-	FIFO_RST	GYRO_RST
3E	62	PWR_MGM	R/W	H_RESET	SLEEP	STBY_XG	STBY_YG	STBY_ZG	CLK_SEL		

Note: Register Names ending in _H and _L contain the high and low bytes, respectively of an internal register value. In the detailed register tables that follow, register names are in capital letters, while register values are in capital letters and italicized. For example, the AUX_XOUT_H register (Register 35) contains the 8 most significant bits, *AUX_XOUT*[15:8], of the 16-bit X-Axis auxiliary measurement, *AUX_XOUT*.

4 Register Descriptions

This section details each register within the InvenSense MPU-30X0 gyroscope. Note that any bit that is not defined should be set to zero in order to be compatible with future InvenSense devices.

The register space allows single-byte reads and writes, as well as burst reads and writes. When performing burst reads or writes, the memory pointer will increment until either (1) reading or writing is terminated by the master, or (2) the memory pointer reaches an indirect-read or indirect read/write register (registers 57 and 60).

4.1 Register 0 – Who Am I

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
0	0	I2C_IF_ DIS	ID						-	68h or 69h

Description:

This register is used to verify the identity of the device, and to enable/disable the I^2C interface.

Parameters:

I2C_IF_DIS Setting this bit disables I^2C access mode.

ID Contains the 6-bit I^2C address of the device. The Power-On-Reset value of Bit6: Bit1 is 110 100.

Bit0 is reserved. (May be 0 or 1)

4.2 Registers 12 to 17 – Gyro Offsets

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
C	12	X_OFFS_H							
D	13	X_OFFS_L							
E	14	Y_OFFS_H							
F	15	Y_OFFS_L							
10	16	Z_OFFS_H							
11	17	Z_OFFS_L							

Description:

These registers are used to remove DC bias from the sensor outputs. The values in these registers are subtracted from the gyro sensor values before going into the sensor registers (see registers 27 to 34).

Parameters:

X_OFFS_H/L 16-bit offset (high and low bytes) of X gyro offset (2’s complement)

Y_OFFS_H/L 16-bit offset (high and low bytes) of Y gyro offset (2’s complement)

Z_OFFS_H/L 16-bit offset (high and low bytes) of Z gyro offset (2’s complement)

4.3 Registers 18 – FIFO Enable

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
12	18	TEMP_ OUT	GYRO_ XOUT	GYRO_ YOUT	GYRO_ ZOUT	AUX_ XOUT	AUX_ YOUT	AUX_ ZOUT	FIFO_ FOOTER	00h

Description:

These registers determines what data goes into the MPU-3000/3050 FIFO, which is a 512 byte First-In-First-Out buffer (see register 60). Sensor data is automatically placed into the FIFO after each ADC sampling period is complete. The ADC sample rate is controlled by register 21.

The order at which the data is put into the FIFO is from MSB to LSB, which means that it will match the order shown in the parameter detail below. Two bytes are used for each reading. For example, if Gyro X, Gyro Y, Gyro Z, and FIFO_FOOTER are configured to go into the FIFO, then each sample period the following 8 bytes would be inserted into the FIFO, as shown below:

Gyro X high byte	Gyro X low byte	Gyro Y high byte	Gyro Y low byte	Gyro Z high byte	Gyro Z low byte	FIFO_FOOTER High byte	FIFO_FOOTER Low byte

Parameters:

TEMP_OUT Setting this inserts the Temperature reading into FIFO

GYRO_XOUT Setting this inserts the X Gyro reading into FIFO

GYRO_YOUT Setting this inserts the Y Gyro reading into FIFO

GYRO_ZOUT Setting this inserts the Z Gyro reading into FIFO

AUX_XOUT Setting this inserts the X Accelerometer reading into FIFO

AUX_YOUT Setting this inserts the Y Accelerometer reading into FIFO

AUX_ZOUT Setting this inserts the Z Accelerometer reading into FIFO

FIFO_FOOTER Last word (2 bytes) for FIFO read. Described in more detail in register 60

4.4 Registers 19 – AUX (Accel) VDDIO

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
13	19	0	0	0	0	0	AUX_ VDDIO	0	0	00h

Description:

This register determines the I/O logic levels for the secondary I^2C bus clock and data lines (AUX_CL, AUX_DA). 1=VDD, 0=VLOGIC.

Parameters:

AUX_VDDIO I/O logic levels for the secondary I^2C bus clock and data lines (AUX_CL, AUX_DA). 1=VDD, 0=VLOGIC.

0 Load zeros into Bits 0, 1, 3-7.

4.5 Register 20 – AUX (Accel) Slave Address

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
14	20	CLKOUT_EN	AUX_ID							00h

Description:

This register contains the 7-bit slave address of the external accelerometer device. This address is used to access the accel device so that its sensor reading can be automatically read during each sample period at the same time as the gyro sensors.

When reading the accel sensor registers, the MPU-3000/3050 takes over the secondary I^2C bus, as a master to the accel device, performing a burst read of the sensor registers. For this interface to be active, the *AUX_IF_EN* flag in the User Control register (61) must be set (set to 1).

Whenever changing this register, the accel interface must be reset to take effect. Refer to the User Control register (61).

Parameters:

AUX_ID Contains the I^2C address of the device, which can also be changed by writing to this register.

CLKOUT_EN 1 – reference clock output is provided at CLKOUT pin

0 – function is disabled.

4.6 Register 21 – Sample Rate Divider

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
15	21	SMPLRT_DIV								00h

Description:

This register determines the sample rate of the MPU-3000/3050 gyros. The analog gyros are sampled internally at either 1kHz or 8kHz, determined by the *DLPF_CFG* setting (see register 22). This sampling is then filtered digitally and delivered into the sensor registers after the number of cycles determined by this register. The sample rate is given by the following formula:

$F_{sample} = F_{internal}$ / (divider+1), where $F_{internal}$ is either 1kHz or 8kHz

As an example, if the internal sampling is at 1kHz, then setting this register to 7 would give the following:

F_{sample} = 1kHz / (7 + 1) = 125Hz, or 8ms per sample

Parameters:

SMPLRT_DIV Sample rate divider: 0 to 255

4.7 Register 22 – DLPF, Full Scale, External Sync

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
16	22	EXT_SYNC_SET			FS_SEL		DLPF_CFG			00h

Description:

This register configures several parameters related to the sensor acquisition.

The *EXT_SYNC_SET* parameter allows capturing the state of the external frame synchronization input pin (FSYNC, pin 11). The value of this input can be inserted into the LSB of one of the sensor registers. The register chosen is as follows:

EXT_SYNC_SET

EXT_SYNC_SET	Register
0	No sync (default)
1	TEMP_OUT_L[0]
2	GYRO_XOUT_L[0]
3	GYRO_YOUT_L[0]
4	GYRO_ZOUT_L[0]
5	AUX_XOUT_L[0]
6	AUX_YOUT_L[0]
7	AUX_ZOUT_L[0]

The *FS_SEL* parameter allows setting the full-scale range of the gyro sensors, as described in the table below.

FS_SEL

FS_SEL	Gyro Full-Scale Range
0	±250°/sec
1	±500°/sec
2	±1000°/sec
3	±2000°/sec

The *DLPF_CFG* parameter sets the digital low pass filter configuration. It also determines the internal analog sampling rate used by the device as shown in the table below.

DLPF_CFG

DLPF_CFG	Low Pass Filter Bandwidth	Analog Sample Rate
0	256Hz	8kHz
1	188Hz	1kHz
2	98Hz	1kHz
3	42Hz	1kHz
4	20Hz	1kHz
5	10Hz	1kHz
6	5Hz	1kHz

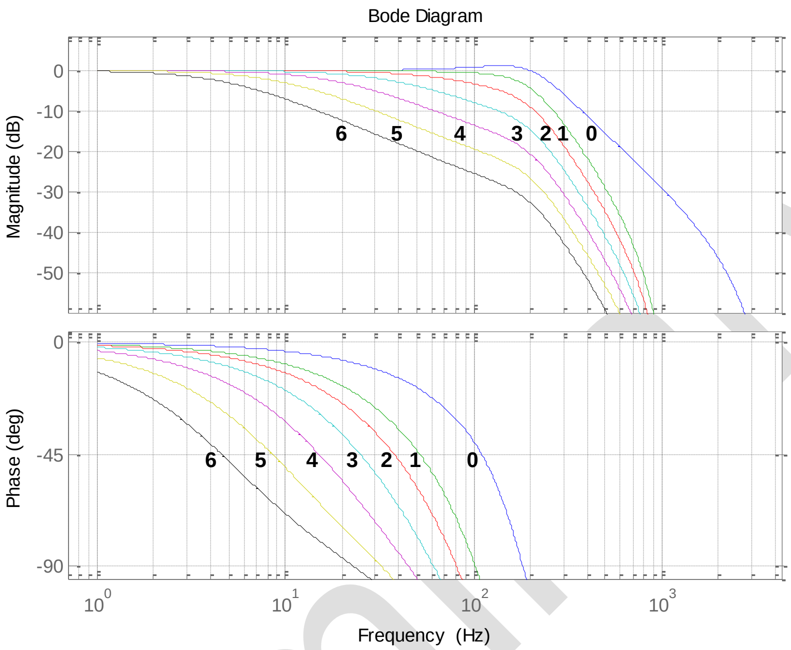
Parameters:

EXT_SYNC_SET Routing for the external frame synchronization input bit

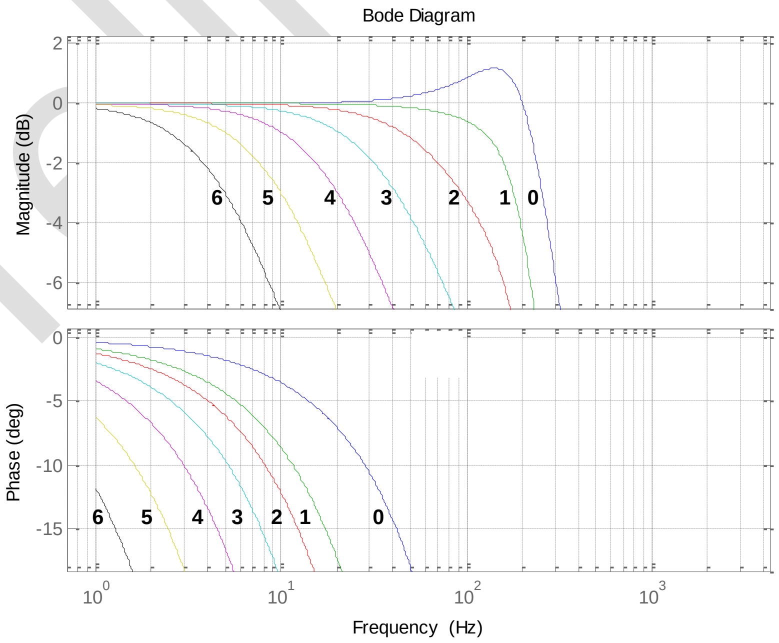
FS_SEL Full scale selection for gyro sensor data

DLPF_CFG Digital low pass filter configuration

DLPF Characteristics: The gain and phase responses of the digital low pass filter settings (*DLPF_CFG*) are shown below:



Gain and Phase vs. Digital Filter Setting



Gain and Phase vs. Digital Filter Setting, Showing Passband Details

4.8 Register 23 – Interrupt Configuration

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
17	23	ACTL	OPEN	LATCH_INT_EN	INT_ANYRD_2CLEAR	-	MPU_RDY_EN	DMP_DONE_EN	RAW_RDY_EN	00h

Description:

This register configures the interrupt operation of the MPU-3000/3050. The interrupt output pin (INT) configuration can be set, the interrupt latching/clearing method can be set, and the triggers for the interrupt can be set. If LATCH_INT_EN = 1, the INT pin is held active until the interrupt status register is cleared.

Note that if the application requires reading every sample of data from the MPU-3000/3050, it is best to enable the raw data ready interrupt (*RAW_RDY_EN*). This allows the application to know when new sample data is available.

Parameters:

ACTL Logic level for INT output pin – 1=active low, 0=active high

OPEN Drive type for INT output pin – 1=open drain, 0=push-pull

LATCH_INT_EN Latch mode – 1=latch until interrupt is cleared, 0=50us pulse

INT_ANYRD_2CLEAR Interrupt status register clear method – 1=clear by reading any register, 0=clear by reading interrupt status register (26) only

MPU_RDY_EN Enable interrupt when device is ready (PLL ready after changing clock source)

DMP_DONE_EN Enable interrupt when DMP is done (programmable functionality)

RAW_RDY_EN Enable interrupt when data is available

4.9 Register 24 – AUX (Accel) Burst Read Address

Type: Read only

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
18	24	BURST_ADDR								00h

Description:

This register configures the burst-mode-read starting address for an accelerometer attached to the secondary I2C bus of the MPU-3000/3050

Parameters:

BURST_ADDR Burst-mode read starting address for external accelerometer attached to secondary I2C bus of the MPU-3000/3050. This is the starting address of the accelerometer which the MPU could use to read from.

4.10 Register 26 – Interrupt Status

Type: Read only

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
1A	26	-	-	-	-	-	MPU_RDY	DMP_DONE	RAW_DATA_RDY	00h

Description:

This register is used to determine the status of the MPU-3000/3050 interrupt. Whenever one of the interrupt sources is triggered, the corresponding bit will be set. The polarity of the interrupt pin (active high/low) and the latch type (pulse or latch) has no effect on these status bits.

In normal use, the *RAW_DATA_RDY* interrupt is used to determine when new sensor data is available in either the sensor registers (27 to 34) or in the FIFO (60).

Interrupt Status bits get cleared as determined by INT_ANYRD_2CLEAR in the interrupt configuration register (23).

Parameters:

MPU_RDY PLL ready

DMP_DONE Digital Motion Processor (DMP) is done

RAW_DATA_RDY Raw data or FIFO data is ready

4.11 Registers 27 to 40 – Sensor Registers

Type: Read only

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value*
1B	27	TEMP_OUT_H								00h
1C	28	TEMP_OUT_L								00h
1D	29	GYRO_XOUT_H								00h
1E	30	GYRO_XOUT_L								00h
1F	31	GYRO_YOUT_H								00h
20	32	GYRO_YOUT_L								00h
21	33	GYRO_ZOUT_H								00h
22	34	GYRO_ZOUT_L								00h
23	35	AUX_XOUT_H								00h
24	36	AUX_XOUT_L								00h
25	37	AUX_YOUT_H								00h
26	38	AUX_YOUT_L								00h
27	39	AUX_ZOUT_H								00h
28	40	AUX_ZOUT_L								00h

*Default Value applies if sensor is disabled.

Description:

These registers contain the gyro, temperature and auxiliary (accel) sensor data for the MPU-3000/3050. At any time, these values can be read from the device; however it is best to use the interrupt function to determine when new data is available.

Before being placed into these registers, the sensor data are first manipulated by the full scale setting (register 22) and the offset settings (registers 12 to 17).

Parameters:

TEMP_OUT_H/L 16-bit temperature data (2’s complement data format)

GYRO_XOUT_H/L 16-bit X gyro output data (2’s complement data format)

GYRO_YOUT_H/L 16-bit Y gyro output data (2’s complement data format)

GYRO_ZOUT_H/L 16-bit Z gyro output data (2’s complement data format)

AUX_XOUT_H/L 16-bit X aux (accel) output data (as available from aux)

AUX_YOUT_H/L 16-bit Y aux (accel) output data (as available from aux)

AUX_ZOUT_H/L 16-bit Z aux (accel) output data (as available from aux)

4.12 Registers 58 to 59 – FIFO Count

Type: Read only

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
3A	58	-	-	-	-	-	-	FIFO_COUNT_H		00h
3B	59	FIFO_COUNT_L								00h

Description:

This register indicates how many bytes of valid data are contained in the FIFO. The FIFO can contain up to 512 bytes of data

If the FIFO gets filled up completely, the length will read 512. In this state, the MPU-3000/3050 continues to put new sensor data into the FIFO, thus overwriting old FIFO data. Note, however, that the alignment of sensor data can change in this overflow condition. InvenSense recommends resetting the FIFO if an overflow condition occurs (use register 61), which will clear out the FIFO.

Parameters:

FIFO_COUNT_H/L Number of bytes currently in FIFO

4.13 Register 60 – FIFO Data

Type: Read only

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
3C	60	FIFO_DATA								00h

Parameters:

FIFO_DATA Contains the FIFO data

Description:

This is the output register of the FIFO. Each read of this register gets the oldest contents of the MPU-3000/3050 FIFO buffer; thus the data is read out in the same order that the DMP put the data in. If the FIFO operation is enabled, the DMP puts new data into the FIFO at each sample interval. The data that goes in is determined by the FIFO enable registers (18 and 19).

A burst read is required for reading *multiple* bytes from this register, since any read on this register causes an auto increment and a prefetch to occur.

Proper operation of the FIFO requires that at least one word (2 bytes) of data be left in the FIFO during any read operation. To implement this, it is recommended that one extra word be added to the end of the FIFO data so that all desired data can be read at each cycle, leaving the extra word remaining in the FIFO. This extra word will be read out (first) during the next read operation on the FIFO.

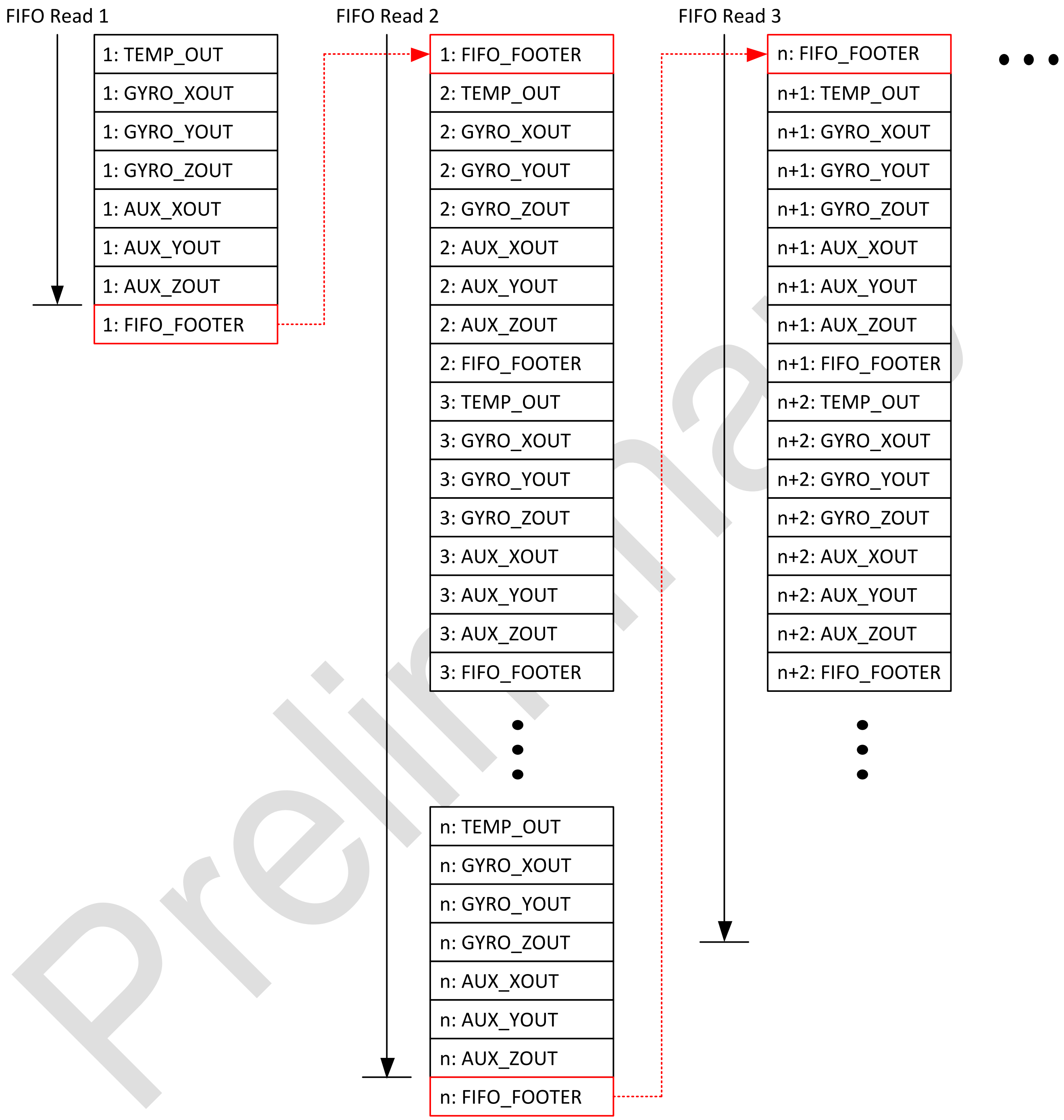
Data is read into the FIFO in the following order:

TEMP_OUT	Temperature
GYRO_XOUT	X Gyro
GYRO_YOUT	Y Gyro
GYRO_ZOUT	Z Gyro
AUX_XOUT	*X Accelerometer high and low bytes (2 bytes)*
AUX_YOUT	*Y Accelerometer high and low bytes (2 bytes)*
AUX_ZOUT	*Z Accelerometer high and low bytes (2 bytes)*
FIFO_FOOTER	*Last word for FIFO read (2 bytes)*

For example, if it is desired to obtain temp, gyro, and accel data from the FIFO, then one should also add one of the aux ADC readings (the required extra word) into the FIFO enable registers (18 or 19) in addition to the desired data. As shown in the figure below, the first time data is written to the FIFO, the FIFO will contain: *TEMP_OUT, GYRO_XOUT, GYRO_YOUT, GYRO_ZOUT, AUX_XOUT, AUX_YOUT, AUX_ZOUT,* and *FIFO_FOOTER*. The first FIFO read will read all but the *FIFO_FOOTER* data, which will be read in the 2nd FIFO read. In the 2nd FIFO read, the *FIFO_FOOTER* data that was left over from the previous read is read out first, followed by all but the last *FIFO_FOOTER* data in the FIFO. This pattern of reading is continued, as shown in the figure below.

Note that the first FIFO read is similar to the subsequent reads in that one word of data is always left in the FIFO. It differs, though, in that in subsequent reads the leftover data from the previous read is read first; however, for the first read there is no leftover data from a previous read.

If the FIFO is allowed to overflow, it operates as a circular buffer in which at any time it contains the most recent 512 bytes. Recommended operation in this mode is to disable data going into the FIFO prior to reading the FIFO to avoid pointer conflicts. After halting the FIFO input, the 512 bytes in the FIFO should be read out in a single burst read. The first byte read will not be valid.



Reading from the FIFO

4.14 Register 61 – User Control

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
3D	61	-	FIFO_ EN	AUX_IF_ EN	-	AUX_IF_ RST	-	FIFO_ RST	GYRO_ RST	00h

Description:

This register is used to enable various modes on the MPU-3000/3050, as well as reset these functions.

For each of the functions that can be enabled, the function should be reset at the same time to assure it works properly. Note that the reset bits in the register are automatically cleared after the function is reset.

Parameters:

FIFO_EN	Enable FIFO operation for sensor data
AUX_IF_EN	Enable third-party accelerometer interface via I2C (clear bit to pass through I2C bus)
AUX_IF_RST	Reset third-party accelerometer interface function; set this only after changing AUX_IF_EN to 0.
FIFO_RST	Reset FIFO function; set this to clear FIFO or when changing FIFO_EN
GYRO_RST	Reset gyro analog and digital functions

4.15 Register 62 – Power Management

Type: Read/Write

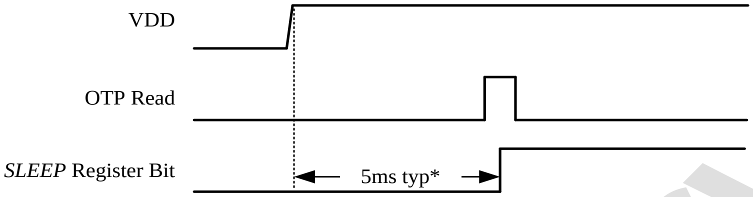
Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
3E	62	H_RESET	SLEEP	STBY _XG	STBY _YG	STBY_Z G	CLK_SEL			00h

Description:

This register is used to manage the power control, select the clock source, and to issue a master reset to the device.

Setting the *SLEEP* bit in the register puts the device into a low power sleep mode. In this mode, only the serial interface and internal registers remain active, allowing for a very low standby current. Clearing this bit puts the device back into normal mode. The individual standby selections for each of the gyros should be used if any of them are not used by the application.

The power-up sequence of the *SLEEP* register bit is shown in the figure below. After VDD is applied to the part, SLEEP is initially low (part in normal operating mode). A short while afterwards, the internal charge pumps are brought up, and the part's OTP memory is read, and *SLEEP* is set high, thus putting the part into its low-power sleep mode. The part stays in this mode until the register bit is cleared.



Power-Up Sequence of *SLEEP* Register Bit

*Note: characterization data for this timing spec will be available upon characterization of Rev F devices.

The *CLK_SEL* setting determines the device clock source as follows:

CLK_SEL

CLK_SEL	Clock Source
0	Internal oscillator
1	PLL with X Gyro reference
2	PLL with Y Gyro reference
3	PLL with Z Gyro reference
4	PLL with external 32.768kHz reference
5	PLL with external 19.2MHz reference
6	Reserved
7	Stop clock and synchronous reset clock state

On power up, the MPU-3000/3050 defaults to the internal oscillator. It is highly recommended that the device is configured to use one of the gyros (or an external clock) as the clock reference, due to the improved stability.

Parameters:

H_RESET	Reset device and internal registers to the power-up-default settings
SLEEP	Enable low power sleep mode
STBY_XG	Put gyro X in standby mode (1=standby, 0=normal)
STBY_YG	Put gyro Y in standby mode (1=standby, 0=normal)
STBY_ZG	Put gyro Z in standby mode (1=standby, 0=normal)
CLK_SEL	Select device clock source

This information furnished by InvenSense is believed to be accurate and reliable. However, no responsibility is assumed by InvenSense for its use, or for any infringements of patents or other rights of third parties that may result from its use. Specifications are subject to change without notice. InvenSense reserves the right to make changes to this product, including its circuits and software, in order to improve its design and/or performance, without prior notice. InvenSense makes no warranties, neither expressed nor implied, regarding the information and specifications contained in this document. InvenSense assumes no responsibility for any claims or damages arising from information contained in this document, or from the use of products and services detailed therein. This includes, but is not limited to, claims or damages based on the infringement of patents, copyrights, mask work and/or other intellectual property rights.

Certain intellectual property owned by InvenSense and described in this document is patent protected. No license is granted by implication or otherwise under any patent or patent rights of InvenSense. This publication supersedes and replaces all information previously supplied. Trademarks that are registered trademarks are the property of their respective companies. InvenSense sensors should not be used or sold in the development, storage, production or utilization of any conventional or mass-destructive weapons or for any other weapons or life threatening applications, as well as in any other life critical applications such as medical equipment, transportation, aerospace and nuclear instruments, undersea equipment, power plant equipment, disaster prevention and crime prevention equipment.

InvenSense®, MotionCommand®, TouchAnywhere®, and AirSign® are registered trademarks of InvenSense, Inc. MPU^{TM}, MPU-3000^{TM}, MPU-3050^{TM}, MPU-30X0^{TM}, Digital Motion Processor™, DMP ™, Motion Processing Unit™, MotionApps™, MotionProcessing™, and MotionFusion™ are trademarks of InvenSense, Inc.