Application Note

Programming Sequence for DMP Hardware Functions

CONTENTS

1 Revision History

Revision Date	Revision	Description
10/17/2012	0.1	Initial Release

2 Overview

InvenSense MotionTracking Devices include four advanced hardware features that can be enabled and disabled through simple hardware register settings described in this document. These advanced hardware features enable the following motion-based functions without using an external microprocessor: Android orientation, gyroscope data integration ("quaternion"), tap gesture recognition, and pedometer.

This document covers the following areas:

1. MPU registers used to set up the DMP
2. DMP registers to set up and configure the hardware features
3. Set up and configuration procedures for each of the advanced hardware features
    - Prepare the chip settings for use of advanced hardware features
    - Setup and configure the advanced hardware features
    - Enable operation of advanced hardware features
    - Read and interpret output data

The hardware features considered in this application note are as follows:

- Android Orientation
- Gyroscope data integration ("quaternion")
- Tap gesture
- Pedometer

Note: Android Orientation is compliant to the Ice Cream Sandwich definition of the function.

In addition to these advanced hardware features, the InvenSense embedded Digital Motion Processor (DMP) can output the following data to the FIFO:

- Gyroscope raw data
- Accelerometer raw data
- 3-Axis Quaternion data
- 6-Axis Quaternion data
- Android Orientation data
- Tap Gesture data

Please note that the pedometer data is directly read from the MotionTracking device advanced hardware feature registers.

References:

[1] MPU-6500 Register Map & Descriptions (RM-MPU-6500A-00)

[2] MPU-6000 and MPU-6050 Register Map & Descriptions (RM-MPU-6000A-00)

3 MPU Register Map

The following partial list of MPU registers are used to set up the MotionTracking device for using the advanced hardware features. For the complete set of registers, please refer to the Register Map and Register Descriptions documents for the MPU-6000, MPU-6050, and MPU-6500 devices [1][2].

Addr (Hex)	Addr (Dec.)	Register Name	Serial I/F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
19	25	SMPLRT_DIV	R/W	SMPLRT_DIV[7:0]							
1A	26	CONFIG	R/W	-	-	EXT_SYNC_SET[2:0]			DLPF_CFG[2:0]		
1B	27	GYRO_CONFIG	R/W	-	-	-	FS_SEL [1:0]		-	-	-
1C	28	ACCEL_CONFIG	R/W	XA_ST	YA_ST	ZA_ST	AFS_SEL[1:0]		ACCEL_HPF[2:0]		
23	35	FIFO_EN	R/W	TEMP _FIFO_EN	XG _FIFO_EN	YG _FIFO_EN	ZG _FIFO_EN	ACCEL _FIFO_EN	SLV2 _FIFO_EN	SLV1 _FIFO_EN	SLV0 _FIFO_EN
38	56	INT_ENABLE	R/W	FF_EN	MOT_EN	ZMOT_EN	FIFO _OFLOW _EN	I2C_MST _INT_EN	-	DMP_INT_ EN	DATA _RDY_EN
3A	58	INT_STATUS	R	FF_INT	MOT_INT	ZMOT _INT	FIFO _OFLOW _INT	I2C_MST _INT	-	DMP_INT	DATA _RDY_INT
6A	106	USER_CTRL	R/W	DMP_EN	FIFO_EN	I2C_MST _EN	I2C_IF _DIS	DMP_RST	FIFO _RESET	I2C_MST _RESET	SIG_COND _RESET
6B	107	PWR_MGMT_1	R/W	DEVICE _RESET	SLEEP	CYCLE	-	TEMP_DIS	CLKSEL[2:0]		
6C	108	PWR_MGMT_2	R/W	LP_WAKE_CTRL[1:0]		STBY_XA	STBY_YA	STBY_ZA	STBY_XG	STBY_YG	STBY_ZG
6D	109	DMP_CTRL_1	R/W	DMP_CTRL_1							
6E	110	DMP _CTRL_2	R/W	DMP_CTRL_2							
6F	111	DMP _CTRL_3	R/W	DMP_CTRL_3							
70	112	FW_START_H	R/W	FW_START_H							
71	113	FW_START_L	R/W	FW_START_L							

4 Advanced Hardware Register Map

The registers below are used to set up and configure the advanced hardware features. Please refer to Section 6.1 on how to write to these registers. Detailed usage of these registers is provided in Section 6.

4.1 Tap and Android Orientation

Addr (Hex)	Register Name	Serial I/F	Bits[7:0]
6AA	ORIENT_EN	R/W	Enable Android Orientation output. ORIENT_TAP_EN must be enabled as well
81E	TAP_EN	R/W	Enable Tap Gesture output. ORIENT_TAP_EN must be enabled as well
AB9	ORIENT_TAP_EN	R/W	Enable Android Orientation and Tap Gesture Output. ORIENT_EN and TAP_EN must be enabled as well
148	TAP_AXES_EN	R/W	Enables Tap Gesture for all 3 Axes
124 – 125 & 1D4 – 1D5	TAP_THR_X	R/W	Sets X-Axis Tap Threshold
128 – 129 & 1D8 – 1D9	TAP_THR_Y	R/W	Sets Y-Axis Tap Threshold
12C – 12D & 1DC – 1DD	TAP_THR_Z	R/W	Sets Z-Axis Tap Threshold
1DE	TAP_TIME_THR	R/W	Sets the “wait time” between two consecutive tap detections
14F	MULTI_TAP_SET	R/W	Sets Tap Gesture for Multi-Tap detection
1DA	MULTI_TAP_THR	R/W	Sets the time range under which all the taps for a “Multi-Tap” must occur
158	SHAKE_REJECT_TIME_THR	R/W	Sets the duration above which gyro data must remain above the shake reject threshold for taps to be rejected
15A	SHAKE_REJECT_TIMEOUT_THR	R/W	Sets the duration above which a gyro axis must remain below the shake reject before tap events will be detected again
15C	SHAKE_REJECT_THR	R/W	Sets the gyro data threshold above which tap events will be rejected

4.2 Pedometer

Addr (Hex)	Register Name	Serial I/F	Bits[7:0]
328 - 329	PEDO_MIN_STEP_BUFFER_THR	R/W	Minimum number of steps that are needed to start incrementing the pedometer counter (prevents false starts)
32A – 32B	PEDO_MIN_STEP_TIME	R/W	Minimum time to constitute a valid step
32C – 32D	PEDO_MAX_STEP_BUFFER_TIME	R/W	Step Buffer is reset to 0 if the “next step” does not occur for longer than this amount of time
32E – 32F	PEDO_MAX_STEP_TIME	R/W	Maximum time to constitute a valid step
360 - 363	PEDO_STEP_COUNT	R	Pedometer Step Count
398 – 39B	PEDO_PEAK_THR	R/W	Pedometer Peak Threshold
3C4 – 3C7	PEDO_WALK_TIME	R	Pedometer Walk Time

4.3 Data Output Configuration and Low Power Quaternion

Addr (Hex)	Register Name	Serial I/F	Bits[7:0]
426	GYRO_MOUNT_MATRIX_CONFIG	R/W	Configures Gyro Mounting Matrix
42A	ACCEL_MOUNT_MATRIX_CONFIG	R/W	Configures Accel Mounting Matrix
456	GYRO_MOUNT_MATRIX_CONFIG_SIGN	R/W	Configures Gyro Mounting Matrix
434	ACCEL_MOUNT_MATRIX_CONFIG_SIGN	R/W	Configures Accel Mounting Matrix
A9D – AA0	3A_LPQ_EN	R/W	Enable 3-Axis Low Power Quaternion output
AA3 – AA6	6A_LPQ_EN	R/W	Enable 6-Axis Low Power Quaternion output
AAB - AB4	RAW_DATA_EN	R/W	Enable raw gyro and raw accel data output
216	FIFO_RATE_DIV_H	R/W	Divider for reducing data output rate
217	FIFO_RATE_DIV_L	R/W	Divider for reducing data output rate
AC4 – ACF	FIFO_RATE_DIV_EN	R/W	Enables Divider for reducing data output rate

5 Register Descriptions

5.1 Register 0x19 – Sample Rate Divider

SMPRT_DIV

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
19	25	SMPLRT_DIV[7:0]							

Description:

This register specifies the divider from the gyroscope output rate used to generate the Sample Rate for the MPU-60X0.

The sensor register output, FIFO output, DMP sampling, Motion detection, Zero Motion detection, and Free Fall detection are all based on the Sample Rate.

The Sample Rate is generated by dividing the gyroscope output rate by *SMPLRT_DIV*:

Sample Rate = Gyroscope Output Rate / (1 + *SMPLRT_DIV*)

where Gyroscope Output Rate = 8kHz when the DLPF is disabled (*DLPF_CFG* = 0 or 7), and 1kHz when the DLPF is enabled (see Register 26).

Note: The accelerometer output rate is 1kHz. This means that for a Sample Rate greater than 1kHz, the same accelerometer sample may be output to the FIFO, DMP, and sensor registers more than once.

For a diagram of the gyroscope and accelerometer signal paths, see Section 8 of the MPU-6000/MPU-6050 Product Specification document.

Parameters:

SMPLRT_DIV 8-bit unsigned value. The Sample Rate is determined by dividing the gyroscope output rate by this value.

5.2 Register 0x1A – Configuration

CONFIG

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1A	26	-	-	EXT_SYNC_SET[2:0]			DLPF_CFG[2:0]		

Description:

This register configures the external Frame Synchronization (FSYNC) pin sampling and the Digital Low Pass Filter (DLPF) setting for both the gyroscopes and accelerometers.

An external signal connected to the FSYNC pin can be sampled by configuring *EXT_SYNC_SET*.

Signal changes to the FSYNC pin are latched so that short strobes may be captured. The latched FSYNC signal will be sampled at the Sampling Rate, as defined in register 25. After sampling, the latch will reset to the current FSYNC signal state.

The sampled value will be reported in place of the least significant bit in a sensor data register determined by the value of *EXT_SYNC_SET* according to the following table.

EXT_SYNC_SET	FSYNC Bit Location
0	Input disabled
1	TEMP_OUT_L[0]
2	GYRO_XOUT_L[0]
3	GYRO_YOUT_L[0]
4	GYRO_ZOUT_L[0]
5	ACCEL_XOUT_L[0]
6	ACCEL_YOUT_L[0]
7	ACCEL_ZOUT_L[0]

The DLPF is configured by *DLPF_CFG*. The accelerometer and gyroscope are filtered according to the value of *DLPF_CFG* as shown in the table below.

DLPF_CFG	Accelerometer (F_s = 1kHz)		Gyroscope		
	Bandwidth (Hz)	Delay (ms)	Bandwidth (Hz)	Delay (ms)	Fs (kHz)
0	260	0	256	0.98	8
1	184	2.0	188	1.9	1
2	94	3.0	98	2.8	1
3	44	4.9	42	4.8	1
4	21	8.5	20	8.3	1
5	10	13.8	10	13.4	1
6	5	19.0	5	18.6	1
7	RESERVED		RESERVED		8

Bit 7 and bit 6 are reserved.

Parameters:

EXT_SYNC_SET 3-bit unsigned value. Configures the FSYNC pin sampling.
DLPF_CFG 3-bit unsigned value. Configures the DLPF setting.

5.3 Register 0x1B – Gyroscope Configuration

GYRO_CONFIG

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1B	27	XG_ST	YG_ST	ZG_ST	FS_SEL[1:0]		-	-	-

Description:

This register is used to trigger gyroscope self-test and configure the gyroscopes' full scale range.

Gyroscope self-test permits users to test the mechanical and electrical portions of the gyroscope. The self-test for each gyroscope axis can be activated by controlling the *XG_ST*, *YG_ST*, and *ZG_ST* bits of this register. Self-test for each axis may be performed independently or all at the same time.

When self-test is activated, the on-board electronics will actuate the appropriate sensor. This actuation will move the sensor's proof masses over a distance equivalent to a pre-defined Coriolis force. This proof mass displacement results in a change in the sensor output, which is reflected in the output signal. The output signal is used to observe the self-test response.

The self-test response is defined as follows:

Self-test response = Sensor output with self-test enabled – Sensor output without self-test enabled

The self-test limits for each gyroscope axis is provided in the electrical characteristics tables of the MPU-6000/MPU-6050 Product Specification document. When the value of the self-test response is within the min/max limits of the product specification, the part has passed self test. When the self-test response exceeds the min/max values specified in the document, the part is deemed to have failed self-test.

FS_SEL selects the full scale range of the gyroscope outputs according to the following table.

FS_SEL	Full Scale Range
0	± 250 °/s
1	± 500 °/s
2	± 1000 °/s
3	± 2000 °/s

Bits 2 through 0 are reserved.

Parameters:

XG_ST	Setting this bit causes the X axis gyroscope to perform self test.
YG_ST	Setting this bit causes the Y axis gyroscope to perform self test.
ZG_ST	Setting this bit causes the Z axis gyroscope to perform self test.
FS_SEL	2-bit unsigned value. Selects the full scale range of gyroscopes.

5.4 Register 0x1C – Accelerometer Configuration

ACCEL_CONFIG

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1C	28	XA_ST	YA_ST	ZA_ST	AFS_SEL[1:0]		ACCEL_HPF[2:0]		

Description:

This register is used to trigger accelerometer self test and configure the accelerometer full scale range. This register also configures the Digital High Pass Filter (DHPF).

Accelerometer self-test permits users to test the mechanical and electrical portions of the accelerometer. The self-test for each accelerometer axis can be activated by controlling the *XA_ST*, *YA_ST*, and *ZA_ST* bits of this register. Self-test for each axis may be performed independently or all at the same time.

When self-test is activated, the on-board electronics will actuate the appropriate sensor. This actuation simulates an external force. The actuated sensor, in turn, will produce a corresponding output signal. The output signal is used to observe the self-test response.

The self-test response is defined as follows:

Self-test response = Sensor output with self-test enabled – Sensor output without self-test enabled

The self-test limits for each accelerometer axis is provided in the electrical characteristics tables of the MPU-6000/MPU-6050 Product Specification document. When the value of the self-test response is within the min/max limits of the product specification, the part has passed self test. When the self-test response exceeds the min/max values specified in the document, the part is deemed to have failed self-test.

AFS_SEL selects the full scale range of the accelerometer outputs according to the following table.

AFS_SEL	Full Scale Range
0	± 2*g*
1	± 4*g*
2	± 8*g*
3	± 16*g*

ACCEL_HPF configures the DHPF available in the path leading to motion detectors (Free Fall, Motion threshold, and Zero Motion). The high pass filter output is not available to the data registers (see Figure in Section 8 of the MPU-6000/MPU-6050 Product Specification document).

The high pass filter has three modes:

- Reset: The filter output settles to zero within one sample. This effectively disables the high pass filter. This mode may be toggled to quickly settle the filter.
- On: The high pass filter will pass signals above the cut off frequency.
- Hold: When triggered, the filter holds the present sample. The filter output will be the difference between the input sample and the held sample.

ACCEL_HPF	Filter Mode	Cut-off Frequency
0	Reset	None
1	On	5Hz
2	On	2.5Hz
3	On	1.25Hz

4	On	0.63Hz
7	Hold	None

Parameters:

XA_ST When set to 1, the X- Axis accelerometer performs self test.

YA_ST When set to 1, the Y- Axis accelerometer performs self test.

ZA_ST When set to 1, the Z- Axis accelerometer performs self test.

ACCEL_FS_SEL 2-bit unsigned value. Selects the full scale range of accelerometers.

ACCEL_HPF 3-bit unsigned value. Selects the Digital High Pass Filter configuration.

5.5 Register 0x23 – FIFO Enable

FIFO_EN

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
23	35	TEMP_ FIFO_EN	XG_ FIFO_EN	YG_ FIFO_EN	ZG_ FIFO_EN	ACCEL _FIFO_EN	SLV2 _FIFO_EN	SLV1 _FIFO_EN	SLV0 _FIFO_EN

Description:

This register determines which sensor measurements are loaded into the FIFO buffer.

Data stored inside the sensor data registers (Registers 59 to 96) will be loaded into the FIFO buffer if a sensor's respective FIFO_EN bit is set to 1 in this register.

When a sensor's FIFO_EN bit is enabled in this register, data from the sensor data registers will be loaded into the FIFO buffer. The sensors are sampled at the Sample Rate as defined in Register 25. For further information regarding sensor data registers, please refer to Registers 59 to 96

When an external Slave's corresponding FIFO_EN bit (*SLVx_FIFO_EN,* where x=0, 1, or 2) is set to 1, the data stored in its corresponding data registers (EXT_SENS_DATA registers, Registers 73 to 96) will be written into the FIFO buffer at the Sample Rate. EXT_SENS_DATA register association with I^2C Slaves is determined by the I2C_SLVx_CTRL registers (where x=0, 1, or 2; Registers 39, 42, and 45). For information regarding EXT_SENS_DATA registers, please refer to Registers 73 to 96.

Note that the corresponding FIFO_EN bit (*SLV3_FIFO_EN)* is found in I2C_MST_CTRL (Register 36). Also note that Slave 4 behaves in a different manner compared to Slaves 0-3. Please refer to Registers 49 to 53 for further information regarding Slave 4 usage.

Parameters:

TEMP_FIFO_EN When set to 1, this bit enables TEMP_OUT_H and TEMP_OUT_L (Registers 65 and 66) to be written into the FIFO buffer.

XG_ FIFO_EN When set to 1, this bit enables GYRO_XOUT_H and GYRO_XOUT_L (Registers 67 and 68) to be written into the FIFO buffer.

YG_ FIFO_EN When set to 1, this bit enables GYRO_YOUT_H and GYRO_YOUT_L (Registers 69 and 70) to be written into the FIFO buffer.

ZG_ FIFO_EN When set to 1, this bit enables GYRO_ZOUT_H and GYRO_ZOUT_L (Registers 71 and 72) to be written into the FIFO buffer.

ACCEL_ FIFO_EN When set to 1, this bit enables ACCEL_XOUT_H, ACCEL_XOUT_L, ACCEL_YOUT_H, ACCEL_YOUT_L, ACCEL_ZOUT_H, and ACCEL_ZOUT_L (Registers 59 to 64) to be written into the FIFO buffer.

SLV2_ FIFO_EN	When set to 1, this bit enables EXT_SENS_DATA registers (Registers 73 to 96) associated with Slave 2 to be written into the FIFO buffer.
SLV1_ FIFO_EN	When set to 1, this bit enables EXT_SENS_DATA registers (Registers 73 to 96) associated with Slave 1 to be written into the FIFO buffer.
SLV0_ FIFO_EN	When set to 1, this bit enables EXT_SENS_DATA registers (Registers 73 to 96) associated with Slave 0 to be written into the FIFO buffer.

Note: For further information regarding the association of EXT_SENS_DATA registers to particular slave devices, please refer to Registers 73 to 96.

5.6 Register 0x38 – Interrupt Enable

INT_ENABLE

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
38	56	FF_EN	MOT_EN	ZMOT_EN	FIFO _OFLOW _EN	I2C_MST _INT_EN	-	DMP_INT_ EN	DATA _RDY_EN

Description:

This register enables interrupt generation by interrupt sources.

For information regarding Free Fall detection, Motion detection, and Zero Motion detection, please refer to Registers 29 to 34

For information regarding the interrupt status for of each interrupt generation source, please refer to Register 58. Further information regarding I^2C Master interrupt generation can be found in Register 54.

Bits 2 and 1 are reserved.

Parameters:

FF_EN	When set to 1, this bit enables Free Fall detection to generate an interrupt.
MOT_EN	When set to 1, this bit enables Motion detection to generate an interrupt.
ZMOT_EN	When set to 1, this bit enables Zero Motion detection to generate an interrupt.
FIFO_OFLOW_EN	When set to 1, this bit enables a FIFO buffer overflow to generate an interrupt.
I2C_MST_INT_EN	When set to 1, this bit enables any of the I^2C Master interrupt sources to generate an interrupt.
DMP_INT_EN	When set to 1, this bit enables the DMP interrupt, which occurs each time a set of data from the DMP has been written to the FIFO.
DATA_RDY_EN	When set to 1, this bit enables the Data Ready interrupt, which occurs each time a write operation to all of the sensor registers has been completed.

5.7 Register 0x3A – Interrupt Status INT_STATUS

Type: Read Only

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
3A	58	FF_INT	MOT_INT	ZMOT _INT	FIFO _OFLOW _INT	I2C_MST _INT	-	DMP_INT	DATA _RDY_INT

Description:

This register shows the interrupt status of each interrupt generation source. Each bit will clear after the register is read.

For information regarding the corresponding interrupt enable bits, please refer to Register 56.

For a list of I^2C Master interrupts, please refer to Register 54.

Bits 2 and 1 are reserved.

Parameters:

FF_INT This bit automatically sets to 1 when a Free Fall interrupt has been generated.

The bit clears to 0 after the register has been read.

MOT_INT This bit automatically sets to 1 when a Motion Detection interrupt has been generated.

The bit clears to 0 after the register has been read.

ZMOT_INT This bit automatically sets to 1 when a Zero Motion Detection interrupt has been generated.

The bit clears to 0 after the register has been read.

FIFO_OFLOW_INT This bit automatically sets to 1 when a FIFO buffer overflow interrupt has been generated.

The bit clears to 0 after the register has been read.

I2C_MST_INT This bit automatically sets to 1 when an I^2C Master interrupt has been generated. For a list of I^2C Master interrupts, please refer to Register 54.

The bit clears to 0 after the register has been read.

DMP_INT This bit automatically sets to 1 when the DMP interrupt has been generated.

DATA_RDY_INT This bit automatically sets to 1 when a Data Ready interrupt is generated.

The bit clears to 0 after the register has been read.

5.8 Register 0x6A – User Control

USER_CTRL

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
6A	106	DMP_EN	FIFO_EN	I2C_MST_EN	I2C_IF_DIS	DMP_RST	FIFO_RESET	I2C_MST_RESET	SIG_COND_RESET

Description:

This register allows the user to enable and disable the DMP, FIFO buffer, I^2C Master Mode**,** and primary I^2C interface. The DMP, FIFO buffer, I^2C Master, sensor signal paths and sensor registers can also be reset using this register.

When *I2C_MST_EN* is set to 1, I^2C Master Mode is enabled. In this mode, the MPU-60X0 acts as the I^2C Master to the external sensor slave devices on the auxiliary I^2C bus. When this bit is cleared to 0, the auxiliary I^2C bus lines (AUX_DA and AUX_CL) are logically driven by the primary I^2C bus (SDA and SCL). This is a precondition to enabling Bypass Mode. For further information regarding Bypass Mode, please refer to Register 55.

MPU-6000: The primary SPI interface will be enabled in place of the disabled primary I^2C interface when *I2C_IF_DIS* is set to 1.

MPU-6050: Always write 0 to *I2C_IF_DIS*.

When the reset bits (*DMP_RST*, *FIFO_RESET, I2C_MST_RESET, and SIG_COND_RESET*) are set to 1, these reset bits will trigger a reset and then clear to 0.

Bits 7 and 3 are reserved.

Parameters:

DMP_EN — When set to 1, this bit enables the DMP features.

When this bit is cleared to 0, DMP features are disabled after the current processing round has completed.

FIFO_EN — When set to 1, this bit enables FIFO operations.

When this bit is cleared to 0, the FIFO buffer is disabled. The FIFO buffer cannot be written to or read from while disabled.

The FIFO buffer's state does not change unless the MPU-60X0 is power cycled.

I2C_MST_EN — When set to 1, this bit enables I^2C Master Mode.

When this bit is cleared to 0, the auxiliary I^2C bus lines (AUX_DA and AUX_CL) are logically driven by the primary I^2C bus (SDA and SCL).

I2C_IF_DIS — MPU-6000: When set to 1, this bit disables the primary I^2C interface and enables the SPI interface instead.

MPU-6050: Always write this bit as zero.

DMP_RST — This bit resets the DMP when set to 1 while DMP_EN equals 0. This bit automatically clears to 0 after the reset has been triggered.

FIFO_RESET — This bit resets the FIFO buffer *w*hen set to 1 while *FIFO_EN* equals 0. This bit automatically clears to 0 after the reset has been triggered.

I2C_MST_RESET This bit resets the I^2C Master when set to 1 while *I2C_MST_EN* equals 0. This bit automatically clears to 0 after the reset has been triggered.

SIG_COND_RESET When set to 1, this bit resets the signal paths for all sensors (gyroscopes, accelerometers, and temperature sensor). This operation will also clear the sensor registers. This bit automatically clears to 0 after the reset has been triggered.

When resetting only the signal path (and not the sensor registers), please use Register 104, SIGNAL_PATH_RESET.

5.9 Register 0x6B – Power Management 1

PWR_MGMT_1

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
6B	107	DEVICE _RESET	SLEEP	CYCLE	-	TEMP_DIS	CLKSEL[2:0]		

Description:

This register allows the user to configure the power mode and clock source. It also provides a bit for resetting the entire device, and a bit for disabling the temperature sensor.

By setting *SLEEP* to 1, the MPU-60X0 can be put into low power sleep mode. When *CYCLE* is set to 1 while *SLEEP* is disabled, the MPU-60X0 will be put into Cycle Mode. In Cycle Mode, the device cycles between sleep mode and waking up to take a single sample of data from accelerometer at a rate determined by *LP_WAKE_CTRL* (register 108). To configure the wake frequency, use *LP_WAKE_CTRL* within the Power Management 2 register (Register 108).

An internal 8MHz oscillator, gyroscope based clock, or external sources can be selected as the MPU-60X0 clock source. When the internal 8 MHz oscillator or an external source is chosen as the clock source, the MPU-60X0 can operate in low power modes with the gyroscopes disabled.

Upon power up, the MPU-60X0 clock source defaults to the internal oscillator. However, it is highly recommended that the device be configured to use one of the gyroscopes (or an external clock source) as the clock reference for improved stability. The clock source can be selected according to the following table.

CLKSEL	Clock Source
0	Internal 8MHz oscillator
1	PLL with X axis gyroscope reference
2	PLL with Y axis gyroscope reference
3	PLL with Z axis gyroscope reference
4	PLL with external 32.768kHz reference
5	PLL with external 19.2MHz reference
6	Reserved
7	Stops the clock and keeps the timing generator in reset

For further information regarding the MPU-60X0 clock source, please refer to the MPU-6000/MPU-6050 Product Specification document.

Bit 4 is reserved.

Parameters:

DEVICE_RESET	When set to 1, this bit resets all internal registers to their default values. The bit automatically clears to 0 once the reset is done. The default values for each register can be found in Section 3.
SLEEP	When set to 1, this bit puts the MPU-60X0 into sleep mode.
CYCLE	When this bit is set to 1 and *SLEEP* is disabled, the MPU-60X0 will cycle between sleep mode and waking up to take a single sample of data from active sensors at a rate determined by *LP_WAKE_CTRL* (register 108).
TEMP_DIS	When set to 1, this bit disables the temperature sensor.
CLKSEL	3-bit unsigned value. Specifies the clock source of the device.

5.10 Register 0x6C – Power Management 2

PWR_MGMT_2

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
6C	108	LP_WAKE_CTRL[1:0]		STBY_XA	STBY_YA	STBY_ZA	STBY_XG	STBY_YG	STBY_ZG

Description:

This register allows the user to configure the frequency of wake-ups in Accelerometer Only Low Power Mode. This register also allows the user to put individual axes of the accelerometer and gyroscope into standby mode.

The MPU-60X0 can be put into Accelerometer Only Low Power Mode using the following steps:

(i) Set CYCLE bit to 1
(ii) Set SLEEP bit to 0
(iii) Set TEMP_DIS bit to 1
(iv) Set STBY_XG, STBY_YG, STBY_ZG bits to 1

All of the above bits can be found in Power Management 1 register (Register 107).

In this mode, the device will power off all devices except for the primary I^2C interface, waking only the accelerometer at fixed intervals to take a single measurement. The frequency of wake-ups can be configured with *LP_WAKE_CTRL* as shown below.

LP_WAKE_CTRL	Wake-up Frequency
0	1.25 Hz
1	5 Hz
2	20 Hz
3	40 Hz

For further information regarding the MPU-6050's power modes, please refer to Register 107.

The user can put individual accelerometer and gyroscopes axes into standby mode by using this register. If the device is using a gyroscope axis as the clock source and this axis is put into standby mode, the clock source will automatically be changed to the internal 8MHz oscillator.

Parameters:

LP_WAKE_CTRL 2-bit unsigned value.
Specifies the frequency of wake-ups during Accelerometer Only Low Power Mode.

STBY_XA When set to 1, this bit puts the X axis accelerometer into standby mode.

STBY_YA When set to 1, this bit puts the Y axis accelerometer into standby mode.

STBY_ZA When set to 1, this bit puts the Z axis accelerometer into standby mode.

STBY_XG When set to 1, this bit puts the X axis gyroscope into standby mode.

STBY_YG When set to 1, this bit puts the Y axis gyroscope into standby mode.

STBY_ZG When set to 1, this bit puts the Z axis gyroscope into standby mode.

5.11 Registers 0x6D – 0x6F – DMP_CTRL

DMP_CTRL

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
6D	109	DMP_CTRL_1							
6E	110	DMP_CTRL_2							
6F	111	DMP_CTRL_3							

Description:

These registers are used to configure the DMP hardware features. Detailed usage guidelines are provided in Section 7.7.

5.12 Registers 0x70 – 0x71 – FW_START

DMP_CTRL

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
70	109	FW_START_H							
71	110	FW_START_L							

Description:

These registers store the Firmware Start value for DMP features. The Firmware Start value is provided by InvenSense. If you do not have this value, please contact your field representative.

6 Advanced Hardware Feature Register Descriptions

The advanced hardware features are not initially enabled after device power up. These features must be individually enabled and configured. There are three advanced hardware features available at this time:

- Low Power Quaternion (3-Axis Gyro & 6-Axis Gyro + Accel)
- Android Orientation (A low-power implementation of Android's screen rotation algorithm)
- Tap (detects the tap gesture)
- Pedometer

6.1 Accessing DMP Registers

To write to advanced hardware register 0xABC with data {0x1, 0x2, 0x3, 0x4}, please follow the procedure below:

1. Write 0xA to DMP_CTRL_1 (0x6D)
2. Write 0xBC to DMP_CTRL_2 (0x6E)
3. Write {0x1, 0x2, 0x3, 0x4} to DMP_CTRL_3 (0x6F)

Please note that when we say "write {0x1, 0x2, 0x3, 0x4} to 0xABC", we imply writing {0x1, 0x2, 0x3, 0x4} to registers 0xABC – 0xABF. This can be achieved by writing the registers one at a time or by burst writing, as described in Section 7.7.

6.2 Registers 0x6AA, 0x81E, 0xAB9 – Android Orientation and Tap Gesture Enable

ORIENT_EN, TAP_EN, ORIENT_TAP_EN

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x6AA	1706	ORIENT_EN	0xD9
0x81E	2078	TAP_EN	0xF8
0xAB9	2745	ORIENT_TAP_EN	0x20

Description:

To enable Android Orientation, write the values above to both ORIENT_EN and ORIENT_TAP_EN.

To enable the Tap Gesture, write the values above to both TAP_EN and ORIENT_TAP_EN

6.3 Register 0x148 – Tap Axes Enable

TAP_AXES_EN

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x148	328	TAP_AXES_EN	0x3F

Description:

Writing the value above to this register will enable all three axes of the Tap Gesture.

6.4 Registers 0x124, 0x125, 0x1D4, 0x1D5 – Tap Threshold X-Axis

TAP_THR_X

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x124	292	TAP_THR_X_1	(T_2 >> 8)
0x125	293	TAP_THR_X_2	(T_2 & 0xFF)
0x1D4	468	TAP_THR_X_3	(T_1 >> 8)
0x1D5	469	TAP_THR_X_4	(T_1 & 0xFF)

Description:

The Tap Threshold determines the sensitivity of tap detection. This threshold can be independently configured for the X, Y, and Z axes. The threshold values can be set to any of the following 7 discrete values: {0, 0.025, 0.050, 0.1, 0.2, 0.4, 0.8}. The threshold values are in *g*/millisecond. In other words, the threshold is equivalent to the slope of the accelerometer data vs time. A tap is detected when this slope exceeds the chosen threshold. A typical value for T is 0.1 *g*/ms.

Given a threshold T:

- T_1 = T * (2^14), T is a 2 byte value
- T_2 = 0.75 * T_1

6.5 Registers 0x128, 0x129, 0x1D8, 0x1D8 – Tap Threshold Y-Axis

TAP_THR_Y

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x128	296	TAP_THR_Y_1	(T_2 >> 8)
0x129	297	TAP_THR_Y_2	(T_2 & 0xFF)
0x1D8	472	TAP_THR_Y_3	(T_1 >> 8)
0x1D9	473	TAP_THR_Y_4	(T_1 & 0xFF)

Description:

The Tap Threshold determines the sensitivity of tap detection. This threshold can be independently configured for the X, Y, and Z axes. The threshold values can be set to any of the following 7 discrete values: {0, 0.025, 0.050, 0.1, 0.2, 0.4, 0.8}. The threshold values are in *g*/millisecond. In other words, the threshold is equivalent to the slope of the accelerometer data vs time. A tap is detected when this slope exceeds the chosen threshold. A typical value for T is 0.1 *g*/ms.

Given a threshold T:

- T_1 = T * (2^14), T is a 2 byte value
- T_2 = 0.75 * T_1

6.6 Registers 0x12C, 0x12D, 0x1DC, 0x1DD – Tap Threshold Z-Axis

TAP_THR_Z

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x12C	300	TAP_THR_Z_1	(T_2 >> 8)
0x12D	301	TAP_THR_Z_2	(T_2 & 0xFF)
0x1DC	476	TAP_THR_Z_3	(T_1 >> 8)
0x1DD	477	TAP_THR_Z_4	(T_1 & 0xFF)

Description:

The Tap Threshold determines the sensitivity of tap detection. This threshold can be independently configured for the X, Y, and Z axes. The threshold values can be set to any of the following 7 discrete values: {0, 0.025, 0.050, 0.1, 0.2, 0.4, 0.8}. The threshold values are in *g*/millisecond. In other words, the threshold is equivalent to the slope of the accelerometer data vs time. A tap is detected when this slope exceeds the chosen threshold. A typical value for T is 0.1 *g*/ms.

Given a threshold T:

- T_1 = T * (2^14), T is a 2 byte value
- T_2 = 0.75 * T_1

6.7 Registers 0x1DE & 0x1DF – Tap Time Threshold

TAP_TIME_THR

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x1DE	478	TAP_TIME_THR_1	(T >>8)
0x1DF	479	TAP_TIME_THR_2	(T & 0xFF)

Description:

This register configures the tap time threshold. The Tap Time Threshold sets the “wait time” that must pass between a detected tap and the next detected tap.

Given a desired time of x milliseconds: T = x/5.

A typical Tap Time Threshold is 100ms.

6.8 Register 0x14F – Set Multi Tap

MULTI_TAP_SET

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x14F	335	MULTI_TAP_SET	N-1

Description:

This register configures the multi-tap setting of the Tap Gesture. The Tap Gesture can be configured to detect multi-tap events instead of single tap events (double tap, triple tap, quad tap.).

The possible options are N = {1,2,3,4}.

6.9 Registers 0x1DA & 0x1DB – Multi-Tap Time Threshold

MULTI_TAP_THR

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x1DA	474	MULTI_TAP_THR_1	(T >>8)
0x1DB	475	MULTI_TAP_THR_2	(T & 0xFF)

Description:

This register configures the multi-tap time threshold. This threshold sets the time range within which all of the “single taps” that constitute a “multi-tap event” must occur.

Given a desired time of x milliseconds: T = x/5.

A typical Multi-Tap Time Threshold is 500ms.

6.10 Registers 0x158 & 0x159 – Shake Reject Time Threshold

SHAKE_REJECT_TIME_THR

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x158	344	SHAKE_REJECT_TIME_THR_1	(T >>8)
0x159	345	SHAKE_REJECT_TIME_THR_2	(T & 0xFF)

Description:

This register configures the Shake Reject Time Threshold. This threshold determines the duration which the data from a gyro axis must be above the shake reject threshold before taps are rejected.

Given a desired time of x milliseconds: T = x/5.

A typical Shake Reject Time Threshold is 10ms.

6.11 Registers 0x15A & 0x15B – Shake Reject Timeout Threshold

SHAKE_REJECT_TIMEOUT_THR

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x15A	346	SHAKE_REJECT_TIMEOUT_THR_1	(T >>8)
0x15B	347	SHAKE_REJECT_TIMEOUT_THR_2	(T & 0xFF)

Description:

This register configures the Shake Reject Timeout Threshold. This threshold determines the duration which a gyro axis must remain below the shake reject threshold, after taps have been rejected, before Tap events will be detected again.

Given a desired time of x milliseconds: T = x/5.

A typical Shake Reject Timeout Threshold is 10ms.

6.12 Registers 0x15C – 0x15F – Shake Reject Threshold

SHAKE_REJECT_THR

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x15C	348	SHAKE_REJECT_THR_1	((T >>24) & 0xFF)
0x15D	349	SHAKE_REJECT_THR_2	((T >> 16) & 0xFF)
0x15E	350	SHAKE_REJECT_THR_3	((T >> 8) & 0xFF)
0x15F	351	SHAKE_REJECT_THR_4	(T & 0xFF)

Description:

This register configures the Shake Reject Threshold. When gyro sensor data is above this threshold, tap events will be rejected.

Given a desired threshold of x dps: T = round_to_32_bit_integer(x * 46850.825)

6.13 Registers 0x328 – 0x329 – Pedometer Minimum Step Buffer Threshold

PEDO_MIN_STEP_BUFFER_THR

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x328	808	PEDO_MIN_STEP_BUFFER_THR_1	(T >> 8)
0x329	809	PEDO_MIN_STEP_BUFFER_THR_2	(T & 0xFF)

Description:

These registers set the minimum number of steps that must be detected before the pedometer step count begins incrementing. This is used to prevent false starts of the pedometer when the user only takes a small number of steps but stops again quickly. Once the threshold is exceeded, the pedometer step count increases by T. The pedometer increments regularly thereafter.

Minimum Step Buffer Threshold = T

6.14 Registers 0x32A – 0x32B – Pedometer Minimum Step Time

PEDO_MIN_STEP_TIME

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x32A	810	PEDO_MIN_STEP_TIME_1	(N >> 8)
0x32B	811	PEDO_MIN_STEP_TIME_2	(N & 0xFF)

Description:

These registers configure the minimum elapsed time required to constitute a valid step. The default value is 320ms.

Minimum Step Time (ms)= N * 20

6.15 Registers 0x32C – 0x32D – Pedometer Maximum Step Buffer Time

PEDO_MAX_STEP_BUFFER_TIME

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x32C	812	PEDO_MAX_STEP_BUFFER_TIME_1	(T >> 8)
0x32D	813	PEDO_MAX_STEP_BUFFER_TIME_2	(T & 0xFF)

Description:

These registers set the pedometer maximum step buffer time. While in the step buffer state, the step buffer count resets to 0 if a new step isn't detected for this amount of time (user is considered to have "stopped walking").

Maximum Step Buffer Time = T

6.16 Registers 0x32E – 0x32F – Pedometer Maximum Step Time

PEDO_MAX_STEP_TIME

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x32E	814	PEDO_MAX_STEP_TIME_1	(N >> 8)
0x32F	815	PEDO_MAX_STEP_TIME_2	(N & 0xFF)

Description:

These registers configure the maximum elapsed time required to constitute a valid step. The default value is 1200ms.

Minimum Step Time (ms)= N * 20

6.17 Registers 0x360 – 0x363 – Pedometer Step Count

PEDO_STEP_COUNT

Type: Read

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x360	864	PEDO_STEP_COUNT_1	Data[3]
0x361	865	PEDO_STEP_COUNT_2	Data[2]
0x362	866	PEDO_STEP_COUNT_3	Data[1]
0x363	867	PEDO_STEP_COUNT_4	Data[0]

Description:

These registers store the Pedometer's step count. The step count is found as follows:

Step Count = `data[3]*(2^24) + data[2]*(2^16) + data[1]*(2^8) + data[0]`

Please note that the Pedometer requires the accelerometer to be enabled and FSR set to 2*g*.

6.18 Registers 0x398 – 0x39B – Pedometer Peak Threshold

PEDO_PEAK_THR

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x398	920	PEDO_PEAK_THR_1	((T >> 24) & 0xFF)
0x399	921	PEDO_PEAK_THR_2	((T >> 16) & 0xFF)
0x39A	922	PEDO_PEAK_THR_3	((T >> 8) & 0xFF)
0x39B	923	PEDO_PEAK_THR_4	(T & 0xFF)

Description:

These registers configure the Pedometer's peak threshold. The peak threshold is the absolute value of the minimum accelerometer data that is considered a valid step.

Peak Threshold (*g*)= (T / (2^15)) / (2^14)

6.19 Registers 0x3C4 – 0x3C7 – Pedometer Walk Time

PEDO_WALK_TIME

Type: Read

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x3C4	964	PEDO_WALK_TIME_1	Data[3]
0x3C5	965	PEDO_WALK_TIME_2	Data[2]
0x3C6	966	PEDO_WALK_TIME_3	Data[1]
0x3C7	967	PEDO_WALK_TIME_4	Data[0]

Description:

These registers store the Pedometer's walk time. The walk time (in seconds) is found as follows:

Walk Time = 20 * `(data[3]*(2^24) + data[2]*(2^16) + data[1]*(2^8) + data[0])`

6.20 Registers 0x426 – 0x428 – Gyro Mounting Matrix Configuration

GYRO_MOUNT_MATRIX_CONFIG

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x426	1062	GYRO_MOUNT_MATRIX_CONFIG_1	Refer to Section 8.2
0x427	1063	GYRO_MOUNT_MATRIX_CONFIG_2	Refer to Section 8.2
0x428	1064	GYRO_MOUNT_MATRIX_CONFIG_3	Refer to Section 8.2

Description:

These registers configure the Gyro mounting matrix. The appropriate values that should be written depend on your particular system. Please refer to Section 8.2 on how to find the appropriate values for your system.

6.21 Registers 0x42A – 0x42C – Accel Mounting Matrix Configuration

ACCEL_MOUNT_MATRIX_CONFIG

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x42A	1066	ACCEL_MOUNT_MATRIX_CONFIG_1	Refer to Section 8.3
0x42B	1067	ACCEL_MOUNT_MATRIX_CONFIG_2	Refer to Section 8.3
0x42C	1068	ACCEL_MOUNT_MATRIX_CONFIG_3	Refer to Section 8.3

Description:

These registers configure the Accel mounting matrix. The appropriate values that should be written depend on your particular system. Please refer to Section 8.3 on how to find the appropriate values for your system.

6.22 Registers 0x434 – 0x436 – Accel Mounting Matrix Sign Configuration

GYRO_MOUNT_MATRIX_CONFIG_SIGN

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x434	1076	ACCEL_MOUNT_MATRIX_CONFIG_SIGN_1	Refer to Section 8.3
0x435	1077	ACCEL_MOUNT_MATRIX_CONFIG_SIGN_2	Refer to Section 8.3
0x436	1078	ACCEL_MOUNT_MATRIX_CONFIG_SIGN_3	Refer to Section 8.3

Description:

These registers configure the signs of the non-zero values in the Accel mounting matrix. The appropriate values that should be written depend on your particular system. Please refer to Section 8.3 on how to find the appropriate values for your system.

6.23 Registers 0x456 – 0x458 – Gyro Mounting Matrix Sign Configuration

GYRO_MOUNT_MATRIX_CONFIG_SIGN

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x456	1110	GYRO_MOUNT_MATRIX_CONFIG_SIGN_1	Refer to Section 8.2
0x457	1111	GYRO_MOUNT_MATRIX_CONFIG_SIGN_2	Refer to Section 8.2
0x458	1112	GYRO_MOUNT_MATRIX_CONFIG_SIGN_3	Refer to Section 8.2

Description:

These registers configure the signs of the non-zero values in the Gyro mounting matrix. The appropriate values that should be written depend on your particular system. Please refer to Section 8.2 on how to find the appropriate values for your system.

6.24 Registers 0xA9D – 0xAA0 – 3-Axis Low Power Quaternion Enable

3A_LPQ_EN

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
A9D	2717	3A_LPQ_EN_1	0xC0
A9E	2718	3A_LPQ_EN_2	0xC2
A9F	2719	3A_LPQ_EN_3	0xC4
AA0	2720	3A_LPQ_EN_4	0xC6

Description:

Writing the values above to these registers will enable 3-Axis Low Power Quaternion output from the DMP to the FIFO.

6.25 Registers 0xAA3 – 0xAA6 – 6-Axis Low Power Quaternion Enable

6A_LPQ_EN

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
AA3	2723	6A_LPQ_EN_1	0x20
AA4	2724	6A_LPQ_EN_2	0x28
AA5	2725	6A_LPQ_EN_3	0x30
AA6	2726	6A_LPQ_EN_4	0x30

Description:

Writing the values above to these registers will enable 6-Axis Low Power Quaternion output from the DMP to the FIFO.

6.26 Registers 0xAAB – 0xAB4 – Raw Data Enable

RAW_DATA_EN

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
AAB	2731	RAW_DATA_EN_1	0xA3
AAC	2732	RAW_DATA_EN_2	0xC0
AAD	2733	RAW_DATA_EN_3	0xC8
AAE	2734	RAW_DATA_EN_4	0xC2
AAF	2735	RAW_DATA_EN_5	0xC4
AB0	2736	RAW_DATA_EN_6	0xCC
AB1	2737	RAW_DATA_EN_7	0xC6
AB2	2738	RAW_DATA_EN_8	0xA3
AB3	2739	RAW_DATA_EN_9	0xA3
AB4	2740	RAW_DATA_EN_10	0xA3

Description:

Writing these values to these registers will enable gyro raw data and accel raw data output from the DMP to the FIFO.

Writing the values above to these registers will enable 6-Axis Low Power Quaternion output from the DMP to the FIFO.

6.27 Registers 0x216 & 0x217 – FIFO Rate Divider

FIFO_RATE_DIV_H, FIFO_RATE_DIV_L

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0x216	534	FIFO_RATE_DIV_H	(N >> 8)
0x217	535	FIFO_RATE_DIV_L	(N & 0xFF)

Description:

By default, data will be written from the DMP to the FIFO at 200Hz. These registers set a N+1 integer divider to reduce the data output rate.

- Ex1: for N=0 (the default) the output data is 200/(0 + 1) = 200 Hz.
- Ex2: for N=7, the rate at which FIFO receives data is 200/ (7+1) = 25 Hz.

To enable the FIFO Rate Divider, FIFO_RATE_DIV_EN (0xAC4) must be enabled as well.

6.28 Registers 0xAC4 & 0xACE – FIFO Rate Divider Enable

FIFO_RATE_DIV_EN

Type: Read/Write

Register (Hex)	Register (Decimal)	Register Name	Bits[7:0]
0xAC4	2756	FIFO_RATE_DIV_EN_1	0xFE
0xAC5	2757	FIFO_RATE_DIV_EN_2	0xF2
0xAC6	2758	FIFO_RATE_DIV_EN_3	0xAB
0xAC7	2759	FIFO_RATE_DIV_EN_4	0xC4
0xAC8	2760	FIFO_RATE_DIV_EN_5	0xAA
0xAC9	2761	FIFO_RATE_DIV_EN_6	0xF1
0xACA	2762	FIFO_RATE_DIV_EN_7	0xDF
0xACB	2763	FIFO_RATE_DIV_EN_8	0xDF
0xACC	2764	FIFO_RATE_DIV_EN_9	0xBB
0xACD	2765	FIFO_RATE_DIV_EN_10	0xAF
0xACE	2766	FIFO_RATE_DIV_EN_11	0xDF
0xACF	2767	FIFO_RATE_DIV_EN_12	0xDF

Description:

Writing these values to these registers will enable the FIFO Rate Divider.

7 MPU Setup for Enabling Advanced Hardware Features

Please set up the MPU as shown below before using the advanced hardware features.

7.1 Configure Power Management Registers

- Write 0x00 to PWR_MGMT_1 (0x6B).
- Write 0x00 to PWR_MGMT_2 (0x6C).

Note: This is also the hardware reset value for these registers.

7.2 Configure Gyroscope Parameters

- Write 0x03 to CONFIG (0x1A).
- Write 0x18 to GYRO_CONFIG (0x1B).

Sets the cut-off frequency of the Digital Low-Pass Frequency (DLPF) filter to 42Hz.

Sets the Full Scale Range (FSR) of the gyroscope to 2000dps.

7.3 Configure Accelerometer Parameters

- Write 0x00 to ACCEL_CONFIG (0x1C)

Sets the Accelerometer FSR to 2*g*.

7.4 Configure FIFO and Interrupts

- Write 0x00 to FIFO_EN (0x23).
- Write 0x00 to INT_ENABLE (0x38).

Defers control of the FIFO and the interrupts from the MPU to the DMP.

7.5 Reset the FIFO

- Write 0x04 to USER_CTRL (0x6A)

7.6 Configure Sensor Sample Rate

- Write 0x04 to SMPLRT_DIV (0x19)

Sets sample rate to 200Hz.

7.7 Load DMP Firmware

The DMP firmware is a few kbytes in size. The exact size depends on the particular hardware feature set.

7.7.1 Load Firmware One Byte at a Time

Starting from the byte[0] of data of the firmware image, perform the following for each byte[N] of data:

- Write the value (N >> 8) to DMP_CTRL_1 (0x6D)
- Write the value (N & 0xFF) to DMP_CTRL_2 (0x6E)
- Write the Nth byte of data to DMP_CTRL_3 (0x6F)

Repeat the steps above for each byte of the DMP firmware image.

7.7.2 Load Firmware up to 256 Bytes at a Time

Up to 256 bytes can be burst written. Please follow the steps outlined in Section 7.7.1. However, the maximum number of bytes that can be written at once is not always 256, but is given by (256 – (N & 0xFF)).

7.7.3 Load Firmware Start Value

Please perform the following once the DMP firmware has been loaded.

- Write the 2 byte Firmware Start Value to FW_START (0x70 & 0x71)

The Firmware Start Value is provided by InvenSense. If you do not have this value, please contact your field representative.

8 Mounting Matrix Configuration

Definitions:

- Body Frame = orientation of the end OEM device
- Chip Frame = orientation of the gyro and accelerometer chips
- Mounting Matrix = a 3x3 permutation matrix that is applied to a 3x1 sample of (x,y,z) sensor data to convert its orientation from Chip Frame to Body Frame

A proper mounting matrix is required to align the chip's output data orientation with that of the end OEM device.

8.1 Mounting Matrix Examples

The identity mounting matrix is used when the chip and end device are aligned for all 3 axes. This matrix is represented as [1,0,0; 0,1,0; 0,0,1]. The semi-colons separate the rows.

Each row has 3 elements, or columns. The column number of the non-zero element of each row is the crucial one. For this mounting matrix the mapping from chip frame to body is straightforward:

X_Body_Frame = X_Chip_Frame

Y_Body_Frame = Y_Chip_Frame

Z_Body_Frame = Z_Chip_Frame

Another mounting matrix is [0,1,0; -1,0,0; 0,0,1]. The result of the Chip Frame to Body Frame transformation is:

X_Body_Frame = Y_Chip_Frame

Y_Body_Frame = -1 * X_Chip_Frame

Z_Body_Frame = Z_Chip_Frame

8.2 Configure the Gyro Mounting Matrix

For a given Mounting Matrix M, first perform the following:

- "A" is a variable with 3 elements
- Initialize A as {0x4C, 0xCD, 0x6C}
- For i = {1,2,3} and M such that $\begin{bmatrix} M11 & M12 & M13 \\ M21 & M22 & M23 \\ M31 & M32 & M33 \end{bmatrix}$:
  - if Mi1 != 0 then Ai = 0x4C
  - else if Mi2 != 0 then Ai = 0xCD
  - else if Mi3 != 0 then Ai = 0x6C

As a result of this procedure, the final value of A is a permutation of the initial value.

- Write the 3 bytes of the final value of A to GYRO_MOUNT_MATRIX_CONFIG (DMP 0x426)

Next, we compensate for the sign of the non-zero elements of Mounting Matrix M.

- "B" is a variable with 3 elements
- Initialize B as {0x36,0x56,0x76}

- If the non-zero element of Row 1 of M is Negative, then B1 = (B1 | 0x01). Otherwise, B1 is unchanged.
- If the non-zero element of Row 2 of M is Negative, then B2 = (B2 | 0x01). Otherwise, B2 is unchanged.
- If the non-zero element of Row 3 of M is Negative, then B3 = (B3 | 0x01). Otherwise, B3 is unchanged.

- Write the 3 bytes of the final value of B into GYRO_MOUNT_MATRIX_CONFIG_SIGN (DMP 0x456)

8.3 Configure the Accel Mounting Matrix

The procedure to configure the Accelerometer mounting matrix is analogous to that of the gyro mounting matrix.

- Initilize A as {0x0C, 0xC9, 0x2C}, and repeat the permutation steps listed in the gyro mounting matrix section above.
- Write the 3 bytes of the final value of A into ACCEL_MOUNT_MATRIX_CONFIG (DMP 0x42A)

- Initialize B as {0x26, 0x46, 0x66}, and repeat the sign compensation steps listen in the gyro mounting matrix section above.
- Write the 3 bytes of the final value of B into ACCEL_MOUNT_MATRIX_CONFIG_SIGN (DMP 0x434)

9 Start Using DMP Hardware Features

Once the MPU and DMP have been set up and configured to use DMP hardware features, the DMP is ready to process data and output the results into the FIFO.

To start using the DMP hardware features, please perform the following steps:

9.1 Reset and Enable FIFO & DMP:

1. Write {0x40} to USER_CTRL (0x6A)
2. Write {0x04} to USER_CTRL (0x6A)
3. Write {0x80} to USER_CTRL (0x6A)
4. Write {0x08} to USER_CTRL (0x6A)

9.2 Enable DMP Interrupt

5. Write {0x02} to INT_ENABLE (0x38)

Interrupts will be sent from the DMP at the configured output rate.

10 Reading and Interpreting DMP Output

The procedures described in the previous sections outlined how to setup, configure and start the DMP hardware mode of operation. At this point, the hardware should be receiving samples at a rate of 200 Hz, and writing sets of outputs to the FIFO at the configured output rate.

10.1 DMP Interrupt

Each time a set of data is written to the FIFO, it raises an interrupt. Since the interrupt can come from multiple sources, the user should confirm that it is related to one of the advanced hardware features described in this document by reading the Interrupt Status register (0x3A). If the interrupt was raised by an advanced hardware feature, then the value in this register should be 0x02.

The contents of the FIFO can be read at this point. To learn the procedure to read the MPU FIFO, refer to InvenSense Product Specification and Register Map documents.

10.2 Content of DMP Output to FIFO

The exact contents of the data output to the FIFO depend on the features that were enabled in Section 6.

When all features are enabled, a single set of FIFO data consists of 48 bytes. The data is ordered as shown below:

- Low Power 3-Axis Quaternion (16 bytes)
- Low Power 6-Axis Quaternion (16 bytes)
- Raw Sensor Data (12 bytes)
- Gesture Word (Android Orientation + Tap outputs) (4 bytes)

Note: Pedometer output is read directly from the advanced hardware registers. Please refer to Section 6.17 for details.

If a particular feature is not enabled, its output will not be present in the FIFO data. However, the above order is maintained.

As an example, if 3 Axis Quaternion, Android Orientation, Tap and Raw sensor data outputs are enabled, a total of 16 + 4 + 12 = 32 bytes of data are written to the FIFO.

10.3 Quaternion Data

The 3-Axis and 6-Axis Quaternion outputs each consist of 16 bytes of data. These 16 bytes in turn consists of four 4-byte elements. For a given quaternion Q, the ordering of its elements is {Q0, Q1, Q2, Q3}. Each element is represented using Big Endian byte order.

10.4 Raw Sensor Data

The Raw sensor data consists of 12 bytes. The data is ordered as follows:

- Accel_x_MSB
- Accel_x_LSB
- Accel_y_MSB
- Accel_y_LSB
- Accel_z_MSB
- Accel_z_LSB
- Gyro_x_MSB
- Gyro_x_LSB
- Gyro_y_MSB
- Gyro_y_LSB

- Gyro_z_MSB
- Gyro_z_LSB

10.5 Gesture Word

The 4 byte Gesture Word holds the outputs from the Android Orientation and Tap Gesture features.

Register Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
GESTURE_WORD_3	-	-	-	-	-	-	-	-
GESTURE_WORD_2	-	-	-	-	DISPLAY_ORIENT_EVENT	-	-	TAP_EVENT
GESTURE_WORD_1	-	-	-	-	-	-	-	-
GESTURE_WORD_0	ANDROID_ORIENTATION		TAP_DIRECTION			TAP_NUMBER		

10.5.1 Tap Information

TAP_EVENT indicates a tap event. If TAP_EVENT is 1, then a tap has been detected by the device.

TAP_NUMBER indicates the number of consecutive taps that were detected. The possible number of taps range from 1 to 8, corresponding to TAP_NUMBER + 1. When TAP_EVENT is 1 and TAP_NUMBER is 0, the number of total taps detected is 1. If MULTI_TAP_SET is set for multi-taps, then TAP_NUMBER counts the number of multi-taps that have been detected.

TAP_DIRECTION indicates the direction of the detected tap event. The values are interpreted as shown below:

TAP_DIRECTION Value	Direction
1	Positive X-axis tap
2	Negative X-axis tap
3	Positive Y-axis tap
4	Negative Y-axis tap
5	Positive Z-axis tap
6	Negative Z-axis tap

10.5.2 Android Orientation Information

ANDROID_ORIENT_EVENT indicates an Android orientation event. If ANDROID_ORIENT_EVENT is 1, then a change in the Android orientation was detected.

ANDROID_ORIENTATION indicates the Android orientation value as follows:

ANDROID_ORIENTATION Value	Direction
0	Portrait mode
1	Landscape mode
2	Reverse Portrait mode
3	Reverse Landscape mode

11 Power Optimization

For minimum power consumption of the device, please set up the sensors as follows for each DMP hardware function.

DMP Feature	Accel Rate	Gyro Rate	Compass Rate
Tap	200Hz	(optional@200Hz)*	Off
Android Orientation	200Hz	Off	Off
Pedometer	200Hz	(optional@200Hz)	Off
3-Axis Quaternion	Off	200Hz	Off
6-Axis Quaternion	200Hz	200Hz	Off

*Note: Running Tap Gesture without Gyroscope enabled will disable the Shake rejection feature of the gesture.